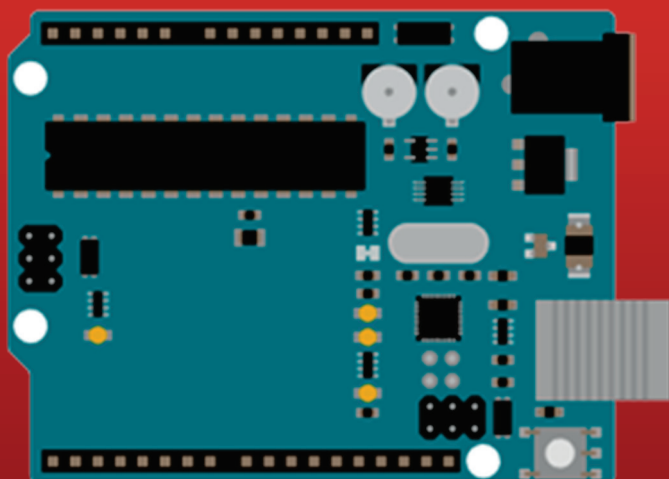


Τοπάλης Ευάγγελος

Δρ. Ηλεκτρολόγος Μηχανικός & Τεχνολογίας Υπολογιστών

Συστήματα αυτοματισμού και
ελέγχου σε κτιριακό περιβάλλον
βασισμένα στις πλατφόρμες
Arduino και ESP32

2024



ISBN 978-618-87290-0-1

ΠΕΡΙΕΧΟΜΕΝΑ

Table of Contents

ΚΕΦΑΛΑΙΟ 1 ΕΙΣΑΓΩΓΗ	13
1.1 ΓΕΝΙΚΑ	13
1.2 ΕΞΥΠΝΟ ΣΠΙΤΙ ΚΑΙ ΟΙΚΙΑΚΟΙ (ΚΤΙΡΙΑΚΟΙ) ΑΥΤΟΜΑΤΙΣΜΟΙ	14
1.3 ΑΝΤΙΚΕΙΜΕΝΟ	16
ΚΕΦΑΛΑΙΟ 2 ΠΛΑΤΦΟΡΜΑ ARDUINO – ΥΛΙΚΟ & ΛΟΓΙΣΜΙΚΟ.....	17
2.1 ΕΙΣΑΓΩΓΗ	17
2.2 ΥΛΙΚΟ (HARDWARE)	18
2.2.1 Γενικά.....	18
2.2.2 Επεξήγηση Βασικών Ακροδεκτών Arduino.....	18
2.2.2.1 Γενικά.....	18
2.2.2.2 Ακροδέκτες Γείωσης (GND pins)	19
2.2.2.3 Ακροδέκτες Τροφοδοσίας (Power Supply pins)	20
2.2.2.4 Ψηφιακοί Ακροδέκτες (Digital pins).....	21
2.2.2.5 PWM (Pulse Width Modulation) - Διαμόρφωση πλάτους παλμού.....	23
2.2.2.6 Ακροδέκτες Διακοπών (Interrupt pins)	23
2.2.2.7 Αναλογικοί Ακροδέκτες (Analog pins).....	25
2.2.3 Επεξήγηση Ειδικών Ακροδεκτών - Πρωτόκολλα Επικοινωνίας.....	26
2.2.3.1 Γενικά.....	26
2.2.3.2 Ακροδέκτες για UART-Serial (UART-Serial pins).....	26
2.2.3.3 Ακροδέκτες για I2C (I2C pins)	27
2.2.3.4 Ακροδέκτες για SPI (SPI pins)	28
2.2.4 Εκδόσεις Arduino	29
2.3 ARDUINO SHIELDS	33
2.4 ΛΟΓΙΣΜΙΚΟ (SOFTWARE) – ARDUINO IDE.....	36
2.4.1 Γενικά.....	36
2.4.2 Λογισμικό ARDUINO IDE.....	37
2.4.3 Εγκατάσταση Λογισμικού ARDUINO IDE	38
2.4.4 Περιβάλλον ARDUINO IDE	40
2.4.5 Λογισμικό ARDUINO IDE – Πρώτος Κώδικας.....	41
2.4.6 Περιβάλλον ARDUINO IDE – Φόρτωμα κώδικα στην πλακέτα	42
2.5 ΛΟΓΙΣΜΙΚΟ (SOFTWARE) - ARDUBLOCK	50
2.5.1 Γενικά.....	50
2.5.2 Εγκατάσταση	50
2.5.3 Περιβάλλον και Χρήση του ArduBlock	52
2.5.4 Λογισμικό Ardublock – Πρώτος Κώδικας	53
ΚΕΦΑΛΑΙΟ 3 ΠΛΑΤΦΟΡΜΑ ESP32 – ΥΛΙΚΟ & ΛΟΓΙΣΜΙΚΟ	59
3.1 ΕΙΣΑΓΩΓΗ	59
3.2 ΒΑΣΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ESP32	60
3.3 ΒΑΣΙΚΕΣ ΕΚΔΟΣΕΙΣ ESP32.....	61
3.4 ΛΟΓΙΣΜΙΚΟ (SOFTWARE).....	64
3.4.1 Γενικά.....	64
ΚΕΦΑΛΑΙΟ 4 ΥΛΙΚΟ ΜΕΡΟΣ ΣΥΣΤΗΜΑΤΟΣ	65
4.1 ΑΙΣΘΗΤΗΡΕΣ, ΕΝΕΡΓΟΠΟΙΗΤΕΣ	65
4.1.1 Γενικά.....	65
4.1.2 Αισθητήρας κίνησης (HC-SR501 PIR).....	65
4.1.3 Αισθητήρας μέτρησης αερίων (MQ-6 Gas Sensor Module).....	67

4.1.4	Αισθητήρας Θερμοκρασίας-Υγρασίας DHT-11 (KY-015 Temperature and Humidity Sensor Module)	69
4.1.5	Αισθητήρας φωτιάς, KY-026 Flame Sensor Module IR Sensor Detector ...	72
4.1.6	Ρελέ, 5V Relay TTL Signal 1 Channel Module High Level Expansion Board For Arduino (KY-019).....	74
4.1.7	Ρολόι πραγματικού χρόνου RTC (Real Time Clock) DS1302	75
4.1.8	Ψηφιακός αισθητήρας υπερήχων, HC-SR04 Ultrasonic Module Distance Measuring Transducer.....	77
4.1.9	RGB LED, KY-016 Color LED Module	79
4.1.10	Φωτοαντίσταση, LDR KY-018 Photoresistor module.....	81
4.1.11	Αισθητήρας υγρασίας εδάφους, Soil moisture detector module	83
4.1.12	Διακόπτης κλίσης υδραργύρου, KY-017 Mercury Tilt Switch Module	84
4.1.13	Αισθητήρας νερού βροχής, Water Level Sensor	86
4.1.14	Ενεργός πιεζοηλεκτρικός βομβητής, KY-012 Active Buzzer Module.....	88
4.1.15	Χωρητικός αισθητήρας αφής, DFRobot Capacitive Touch Sensor	90
4.1.16	Αισθητήρας ανίχνευσης βροχής FC-37	91
4.1.17	KIT επικοινωνίας υπέρυθρων, Infrared IR receiver module wireless remote control kit94	
4.1.18	Ηλεκτροβάνα.....	96
4.1.19	Αισθητήρας ανάκλασης TCRT5000	99
4.1.20	Αισθητήρας ρεύματος, ACS712 Current sensor	101
4.1.21	Ανεμόμετρο	103
4.1.22	Ανεμιστήρας DC Fan.....	105
4.1.23	Πληκτρολόγιο μεμβράνης, 4x4 Membrane Keypad.....	107
4.1.24	Μαγνητικός διακόπτης πόρτας, Magnetic sensor switch.....	109
4.1.25	Αισθητήρας ανίχνευσης ήχου, KY-038 Sound Detection Sensor.....	111
4.2	SHIELDS & ΑΛΛΕΣ ΠΛΑΚΕΤΕΣ ΕΠΙΚΟΙΝΩΝΙΑΣ	113
4.2.1	Γενικά.....	113
4.2.2	Ethernet shield.....	113
4.2.3	GSM/GPRS/GPS (SIM808) shield	114
4.2.3.1	Γενικά.....	114
4.2.4	Bluetooth HC-06 Module	119
4.3	XBEE S2	121
4.3.1	4.4.1 Εκδόσεις.....	121
4.4	ΠΛΑΤΦΟΡΜΑ RASPBERRY-PI.....	122
4.4.1	Βασικές Εκδόσεις.....	122
4.4.2	Λειτουργικό σύστημα Raspbian	125

ΚΕΦΑΛΑΙΟ 5 ΕΝΔΕΙΚΤΙΚΑ ΣΕΝΑΡΙΑ ΑΥΤΟΜΑΤΙΣΜΟΥ & ΕΛΕΓΧΟΥ ΣΕ ΚΤΙΡΙΑΚΟ ΠΕΡΙΒΑΛΛΟΝ

5.1	ΓΕΝΙΚΑ	127
5.2	ΣΕΝΑΡΙΟ ΕΛΕΓΧΟΥ ΦΩΤΙΣΜΟΥ	128
5.3	ΣΕΝΑΡΙΟ ΕΛΕΓΧΟΥ ΟΙΚΙΑΚΩΝ ΣΥΣΚΕΥΩΝ.....	130
5.4	ΣΕΝΑΡΙΟ ΑΥΤΟΜΑΤΗΣ ΑΝΤΛΗΣΗΣ ΝΕΡΩΝ	131
5.5	ΣΕΝΑΡΙΟ ΑΝΙΧΝΕΥΣΗΣ ΦΩΤΙΑΣ Ή ΠΛΗΜΜΥΡΑΣ	133
5.6	ΣΕΝΑΡΙΟ ΕΛΕΓΧΟΥ ΠΟΡΤΑΣ ΓΚΑΡΑΖ (Η ΑΛΛΗΣ ΠΟΡΤΑΣ) – ΗΧΗΤΙΚΗ Ή ΟΠΤΙΚΗ ΕΝΔΕΙΞΗ ΚΑΤΑΣΤΑΣΗΣ ΠΟΡΤΑΣ	135
5.7	ΣΕΝΑΡΙΟ ΑΝΙΧΝΕΥΣΗΣ ΠΟΣΟΣΤΟΥ ΒΡΟΧΗΣ	137

5.8	ΣΕΝΑΡΙΟ ΕΛΕΓΧΟΥ ΘΕΡΜΟΚΡΑΣΙΑΣ ΜΕΣΩ ΚΛΙΜΑΤΙΣΜΟΥ ΚΑΙ ΕΛΕΓΧΟΥ ΠΟΙΟΤΗΤΑΣ ΑΕΡΑ	139
5.9	ΣΕΝΑΡΙΟ ΑΝΟΙΓΜΑΤΟΣ ΕΞΩΤΕΡΙΚΗΣ ΠΟΡΤΑΣ ΜΕ ΔΥΟ ΒΑΘΜΙΔΕΣ ΑΣΦΑΛΕΙΑΣ (ΑΙΣΘ. ΑΦΗΣ ΚΑΙ ΚΩΔΙΚΟΣ).....	142
5.10	ΣΕΝΑΡΙΟ ΑΥΤΟΜΑΤΟΥ ΠΟΤΙΣΜΑΤΟΣ ΓΛΑΣΤΡΩΝ ΣΕ ΜΠΑΛΚΟΝΙ	145
5.11	ΣΕΝΑΡΙΟ ΑΥΤΟΜΑΤΙΣΜΟΥ ΚΗΠΟΥ.....	147
5.12	ΣΕΝΑΡΙΟ ΕΛΕΓΧΟΥ ΕΞΩΤΕΡΙΚΩΝ ΡΟΛΩΝ ΚΑΙ ΕΛΕΓΧΟΥ ΦΩΤΙΣΜΟΥ ΕΞΩΤΕΡΙΚΟΥ ΧΩΡΟΥ	155
5.13	ΣΕΝΑΡΙΟ ΕΛΕΓΧΟΥ ΜΕ ΦΩΝΗΤΙΚΕΣ ΕΝΤΟΛΕΣ(ΜΕΣΩ BLUETOOTH)	157
5.14	ΣΕΝΑΡΙΟ ΜΕΤΡΗΣΗΣ ΚΑΤΑΝΑΛΩΣΗΣ (ΜΕ ΜΕΤΡΗΤΗ ΤΑΣΗΣ ΚΑΙ ΡΕΥΜΑΤΟΣ).....	160
5.15	ΑΝΙΧΝΕΥΣΗ ΤΟΠΟΘΕΣΙΑΣ ΚΑΤΟΙΚΙΔΙΟΥ Ή ΑΤΟΜΟΥ ΜΕ ΕΙΔΙΚΕΣ ΑΝΑΓΚΕΣ, ΚΟΥΜΠΙΟΥ ΕΚΤΑΚΤΟΥ ΑΝΑΓΚΗΣ ΚΑΙ ΣΥΝΘΗΚΩΝ ΠΕΡΙΒΑΛΛΟΝΤΟΣ ΧΩΡΟΥ	166
5.16	ΣΕΝΑΡΙΟ ΚΑΤΑΣΚΕΥΗΣ ΣΤΑΘΜΟΥ ΚΑΙΡΙΚΩΝ ΣΥΝΘΗΚΩΝ (WEATHER STATION).....	171
ΚΕΦΑΛΑΙΟ 6 ΔΙΑΣΥΝΔΕΣΗ ΜΕ ΤΟ ΔΙΑΔΙΚΤΥΟ ΚΑΙ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ.....		179
6.1	ΓΕΝΙΚΑ	179
6.2	ΔΙΑΣΥΝΔΕΣΗ ΜΕ ΤΟ ΔΙΑΔΙΚΤΥΟ	179
6.3	ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ	181
6.4	ΣΥΝΔΕΣΗ MYSQL ΜΕ PHP	184
ΚΕΦΑΛΑΙΟ 7 ΑΠΟΜΑΚΡΥΣΜΕΝΟΣ ΕΛΕΓΧΟΣ, ΙΣΤΟΣΕΛΙΔΕΣ, ΓΡΑΦΗΜΑΤΑ.....		187
7.1	ΓΕΝΙΚΑ	187
7.2	SCRIPTS.....	187
7.2.1	GOOGLE GAUGE	187
7.2.2	STATISTICS.....	191
7.2.3	WEEKLY STATISTICS	195
7.2.4	MONTHLY STATISTICS.....	200
7.3	ΕΝΔΕΙΚΤΙΚΕΣ ΙΣΤΟΣΕΛΙΔΕΣ - ΓΡΑΦΗΜΑΤΑ.....	205
7.3.1	ΚΑΤΑΣΤΑΣΗ, ΑΥΤΟΜΑΤΑ ΣΕΝΑΡΙΑ (SD ΚΑΡΤΑ)	205
7.3.2	ΚΑΤΑΣΤΑΣΗ (WEB SERVER ΣΕ RASPBERRY PI)	213
7.3.3	ΕΝΔΕΙΚΤΙΚΑ ΣΤΑΤΙΣΤΙΚΑ (WEB SERVER ΣΕ RASPBERRY PI)	222
7.3.4	WEEKLY STATISTICS (WEB SERVER ΣΕ RASPBERRY PI).....	228
7.4	ΔΙΑΔΙΚΑΣΙΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΣΕΝΑΡΙΟΥ ΚΗΠΟΥ/ΘΕΡΜΟΚΗΠΙΟΥ ΜΕ THINKSPEAK ΚΑΙ IFTTT	233
7.4.1	THINKSPEAK.....	233
7.4.2	THINKSPEAK &-IFTTT	250
7.5	ΟΛΟΚΛΗΡΩΜΕΝΟ ΣΕΝΑΡΙΟ – ΚΑΤΑΣΤΑΣΗ/ΑΥΤΟΜΑΤΑ ΣΕΝΑΡΙΑ ΜΕ THINKSPEAK/IFTTT	253
7.5.1	THINKSPEAK.....	253
7.5.2	THINKSPEAK - IFTTT	264
ΚΕΦΑΛΑΙΟ 8 ΒΙΒΛΙΟΓΡΑΦΙΑ		267
8.1	ΕΛΛΗΝΟΓΛΩΣΣΗ	267
8.2	ΞΕΝΟΓΛΩΣΣΗ	268

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 2-1 Πλακέτα Arduino v.R3 (2 πλευρές)	17
Εικόνα 2-2 Ακροδέκτες Arduino	19
Εικόνα 2-3 Διάγραμμα Ακροδεκτών Arduino	19
Εικόνα 2-4 Ακροδέκτες Γείωσης Arduino.....	20
Εικόνα 2-5 Ακροδέκτες Γείωσης Arduino.....	20
Εικόνα 2-6 Ακροδέκτες Τροφοδοσίας Arduino.....	20
Εικόνα 2-7 Ακροδέκτες Τροφοδοσίας Arduino.....	20
Εικόνα 2-8 Ακροδέκτες Τροφοδοσίας Arduino.....	21
Εικόνα 2-9 Διαφορετικές Τύπου Τροφοδοσίες Arduino.....	21
Εικόνα 2-10 Ψηφιακοί Ακροδέκτες Arduino.....	22
Εικόνα 2-11 Ψηφιακοί Ακροδέκτες Arduino.....	22
Εικόνα 2-12 Ακροδέκτες Διακοπών	24
Εικόνα 2-13 Αναλογικοί Ακροδέκτες Arduino.....	25
Εικόνα 2-14 Αναλογικοί Ακροδέκτες Arduino.....	25
Εικόνα 2-15 Ακροδέκτες για UART-Serial.....	26
Εικόνα 2-16 Ακροδέκτες για UART-Serial.....	26
Εικόνα 2-17 Ακροδέκτες για I2C.....	28
Εικόνα 2-18 Ακροδέκτες για I2C.....	28
Εικόνα 2-19 Ακροδέκτες για SPI.....	29
Εικόνα 2-20 Ακροδέκτες για SPI.....	29
Εικόνα 2-21 Arduino Uno R3	30
Εικόνα 2-22 Arduino Due	30
Εικόνα 2-23 Arduino Micro	30
Εικόνα 2-24 Arduino Mega 2560 Rev3.....	31
Εικόνα 2-25 Arduino Shields	34
Εικόνα 2-26 Ethernet Shield.....	34
Εικόνα 2-27 Relay Shield	35
Εικόνα 2-28 Motor Shield.....	35
Εικόνα 2-29 LCD Shield.....	36
Εικόνα 2-30 GSM/GPRS Shield	36
Εικόνα 2-31 Περιβάλλον Arduino IDE	37
Εικόνα 2-32 Διαθέσιμες εκδόσεις Arduino IDE.....	38
Εικόνα 2-33 Επιλογή έκδοσης Arduino IDE για windows	38
Εικόνα 2-34 Στάδια εγκατάστασης Arduino IDE σε περιβάλλον Windows	39
Εικόνα 2-35 Επιλογή έκδοσης Arduino IDE για Linux	39
Εικόνα 2-36 Επιλογή έκδοσης Arduino IDE για Linux	40
Εικόνα 2-37 Κυκλωματικό σχέδιο για τον έλεγχο του led	42
Εικόνα 2-38 Επιλογή πλακέτας Arduino	43
Εικόνα 2-39 Επιλογή θύρας usb την οποία είναι συνδεδεμένη η πλακέτα Arduino	44
Εικόνα 2-40 Έλεγχος του προγράμματος για συντακτικά λάθη	44
Εικόνα 2-41 Ανέβασμα του κώδικα στην πλακέτα Arduino	45
Εικόνα 2-42 Σειριακή οθόνη	45
Εικόνα 2-43 Ένδειξη σφαλμάτων	46
Εικόνα 2-44 Εγκατάσταση των βιβλιοθηκών	49
Εικόνα 2-45 Εγκατάσταση Ardublock	50
Εικόνα 2-46 Θέση του Sketchbook (Sketchbook Location)	51
Εικόνα 2-47 Δομή φακέλων	51
Εικόνα 2-48 Επιλογή ArduBlock από το μενού του Arduino IDE	51

Εικόνα 2-49 Περιβάλλον ArduBlock.....	52
Εικόνα 2-50 Περιβάλλον ArduBlock.....	52
Εικόνα 2-51 Περιβάλλον ArduBlock.....	53
Εικόνα 2-52 Κυκλωματικό σχέδιο για τον έλεγχο του led	54
Εικόνα 2-53 Επιλογή Upload to Arduino.....	56
Εικόνα 2-54 Γραφικό περιβάλλον και κώδικας.....	57
Εικόνα 3-1 Πλακέτα ESP32 (2 πλευρές)	59
Εικόνα 3-2 Λειτουργικό μπλοκ διάγραμμα ESP32.....	61
Εικόνα 4-1 Αισθητήρας κίνησης (HC-SR501 PIR)	66
Εικόνα 4-2 Αισθητήρας κίνησης (HC-SR501 PIR)	66
Εικόνα 4-3 Αισθητήρας αερίων (MQ-6 Gas Sensor Module).....	67
Εικόνα 4-4 Αισθητήρες αερίων	68
Εικόνα 4-5 Σύνδεση αισθητήρα αερίων (MQ-6 Gas Sensor Module)	69
Εικόνα 4-6 Αισθητήρας θερμοκρασίας-υγρασίας DHT-11	70
Εικόνα 4-7 Σύνδεση αισθητήρα θερμοκρασίας-υγρασίας DHT-11	71
Εικόνα 4-8 Αισθητήρας φωτιάς, KY-026 Flame Sensor Module IR Sensor Detector	72
Εικόνα 4-9 Σύνδεση αισθητήρα φωτιάς, KY-026 Flame Sensor Module IR Sensor Detector.	73
Εικόνα 4-10 Πλακέτα Ρελέ, 5V Relay TTL Signal 1 Channel Module High Level Expansion Board For Arduino (KY-019)	74
Εικόνα 4-11 Σύνδεση πλακέτας Ρελέ, 5V Relay TTL Signal 1 Channel Module High Level Expansion Board For Arduino (KY-019)	75
Εικόνα 4-12 Πλακέτα ρολογιού πραγματικού χρόνου RTC (Real Time Clock) DS1302	76
Εικόνα 4-13 Σύνδεση πλακέτας ρολογιού πραγματικού χρόνου RTC (Real Time Clock) DS1302	76
Εικόνα 4-14 Πλακέτα ψηφιακού αισθητήρα υπερήχων, HC-SR04 Ultrasonic Module Distance Measuring Transducer	78
Εικόνα 4-15 Σύνδεση πλακέτας ψηφιακού αισθητήρα υπερήχων, HC-SR04 Ultrasonic Module Distance Measuring Transducer	78
Εικόνα 4-16 Πλακέτα RGB LED, KY-016 Color LED Module.....	79
Εικόνα 4-17 Σύνδεση πλακέτας RGB LED, KY-016 Color LED Module	80
Εικόνα 4-18 Πλακέτα φωτοαντίσταση, LDR KY-018 Photoresistor module	81
Εικόνα 4-19 Κύκλωμα φωτοαντίστασης, LDR KY-018 Photoresistor module	82
Εικόνα 4-20 Σύνδεση πλακέτας φωτοαντίστασης, LDR KY-018 Photoresistor module	82
Εικόνα 4-21 Πλακέτα αισθητήρα υγρασίας εδάφους, Soil moisture detector module.....	83
Εικόνα 4-22 Σύνδεση αισθητήρα υγρασίας εδάφους, Soil moisture detector module.....	84
Εικόνα 4-23 Πλακέτα διακόπτη κλίσης υδραργύρου, KY-017 Mercury Tilt Switch Module... ..	85
Εικόνα 4-24 Σύνδεση πλακέτας διακόπτη κλίσης υδραργύρου, KY-017 Mercury Tilt Switch Module	85
Εικόνα 4-25 Πλακέτα αισθητήρα νερού βροχής, Water Level Sensor	86
Εικόνα 4-26 Σύνδεση αισθητήρα νερού βροχής, Water Level Sensor	87
Εικόνα 4-27 Πλακέτα ενεργού πιεζοηλεκτρικού βομβητή, KY-012 Active Buzzer Module	89
Εικόνα 4-28 Σύνδεση ενεργού πιεζοηλεκτρικού βομβητή, KY-012 Active Buzzer Module.....	89
Εικόνα 4-29 Πλακέτα χωρητικού αισθητήρα αφής, DFRobot Capacitive Touch Sensor.....	90
Εικόνα 4-30 Σύνδεση χωρητικού αισθητήρα αφής, DFRobot Capacitive Touch Sensor.....	91
Εικόνα 4-31 Πλακέτα αισθητήρα ανίχνευσης βροχής FC-37.....	92
Εικόνα 4-32 Σύνδεση αισθητήρα ανίχνευσης βροχής FC-37.....	93
Εικόνα 4-33 KIT επικοινωνίας υπέρυθρων, Infrared IR receiver module wireless remote control kit	94

Εικόνα 4-34 Σύνδεση KIT επικοινωνίας υπέρυθρων, Infrared IR receiver module wireless remote control kit	95
Εικόνα 4-35 Ηλεκτροβάνα.....	97
Εικόνα 4-36 Τρανζίστορ TIP120	97
Εικόνα 4-37 Σύνδεση Ηλεκτροβάνας	98
Εικόνα 4-38 Πλακέτα αισθητήρα ανάκλασης TCRT5000.....	99
Εικόνα 4-39 Σύνδεση αισθητήρα ανάκλασης TCRT5000.....	100
Εικόνα 4-40 Πλακέτα αισθητήρα ρεύματος, ACS712 Current sensor	101
Εικόνα 4-41 Σύνδεση αισθητήρα ρεύματος, ACS712 Current sensor	101
Εικόνα 4-42 Ανεμόμετρο.....	103
Εικόνα 4-43 Σύνδεση ανεμόμετρου.....	103
Εικόνα 4-44 Ανεμιστήρας DC	105
Εικόνα 4-45 Σύνδεση ανεμιστήρα DC Fan	106
Εικόνα 4-46 Πληκτρολόγιο μεμβράνης, 4x4 Membrane Keypad	108
Εικόνα 4-47 Σύνδεση πληκτρολογίου μεμβράνης, 4x4 Membrane Keypad	108
Εικόνα 4-48 Μαγνητικός διακόπτης πόρτας, Magnetic sensor switch	110
Εικόνα 4-49 Σύνδεση μαγνητικού διακόπτη πόρτας, Magnetic sensor switch	110
Εικόνα 4-50 Πλακέτα αισθητήρα ανίχνευσης ήχου, KY-038 Sound Detection Sensor	111
Εικόνα 4-51 Σύνδεση αισθητήρα ανίχνευσης ήχου, KY-038 Sound Detection Sensor	112
Εικόνα 4-52 Πλακέτα Ethernet shield (2 πλευρές)	113
Εικόνα 4-53 Σύνδεση shield με arduino.....	114
Εικόνα 4-54 Πλακέτα με τον Sim808	115
Εικόνα 4-55 Πλακέτα με τον Sim808	117
Εικόνα 4-56 Σύνδεση Bluetooth HC-06 Module	120
Εικόνα 5-1 Σενάριο ελέγχου φωτισμού	129
Εικόνα 5-2 Σενάριο ελέγχου οικιακής συσκευής.....	130
Εικόνα 5-3 Σενάριο αυτόματης άντλησης νερών	132
Εικόνα 5-4 Σενάριο ανίχνευσης φωτιάς και πλημμύρας.....	134
Εικόνα 5-5 Σχεδίαση ελέγχου γκαραζόπορτας	136
Εικόνα 5-6 Κύκλωμα σύνδεσης του σεναρίου ανίχνευσης βροχής.....	138
Εικόνα 5-7 Συνδεσμολογία σεναρίου ελέγχου θερμοκρασίας μέσω κλιματισμού και ελέγχου ποιότητας αέρα με Arduino	140
Εικόνα 5-8 Σενάριο ανοίγματος εξωτερικής πόρτας με δύο βαθμίδες ασφάλειας (αισθ. αφής και κωδικός)	143
Εικόνα 5-9 Σενάριο αυτόματου ποτίσματος γλαστρών σε μπαλκόνι.....	146
Εικόνα 5-10 Σενάριο αυτοματισμού κήπου.....	148
Εικόνα 5-11 Σενάριο ελέγχου εξωτερικών ρολών και φωτισμού εξωτερικού χώρου	156
Εικόνα 5-12 Εφαρμογή AMR Voice	158
Εικόνα 5-13 Συνδεσμολογία Bluetooth HC 06 με Arduino	159
Εικόνα 5-14 Αναλυτικό κύκλωμα για τη μέτρηση ρεύματος από CT αισθητήρα.....	161
Εικόνα 5-15 Αναλυτικό κύκλωμα για τη μέτρηση ρεύματος από το Arduino.....	162
Εικόνα 5-16 Συνδεσμολογία σεναρίου μέτρησης κατανάλωσης (με μετρητή τάσης και ρεύματος) με Arduino	163
Εικόνα 5-17 Σενάριο ανίχνευσης τοποθεσίας κατοικίδιου ή ατόμου με ειδικές ανάγκες, κουμπιού έκτακτου ανάγκης και συνθηκών περιβάλλοντος χώρου.....	167
Εικόνα 5-18 Σενάριο κατασκευής σταθμού καιρικών συνθηκών (weather station)	172
Εικόνα 7-1 Ο μετρητής με την τελευταία μέτρηση της θερμοκρασίας.....	191
Εικόνα 7-2 Η γραφική της θερμοκρασίας συναρτήσει του χρόνου	195
Εικόνα 7-3 Η γραφική της υγρασίας συναρτήσει του χρόνου	195

Εικόνα 7-4 Απεικόνιση της ελάχιστης, μέγιστης και μέσης τιμής της θερμοκρασίας	199
Εικόνα 7-5 Απεικόνιση της ελάχιστης, μέγιστης και μέσης τιμής της υγρασίας.....	200
Εικόνα 7-6 Απεικόνιση της ελάχιστης, μέγιστης και μέσης τιμής της θερμοκρασίας	204
Εικόνα 7-7 Απεικόνιση της ελάχιστης, μέγιστης και μέσης τιμής της υγρασίας.....	204
Εικόνα 7-8 Τρέχουσες τιμές	205
Εικόνα 7-9 Αυτόματα σενάρια	205
Εικόνα 7-10 Πίνακας ελέγχου	206
Εικόνα 7-11 Ολοκληρωμένη Ιστοσελίδα.....	206
Εικόνα 7-12 Τρέχουσες τιμές (web server).....	214
Εικόνα 7-13 Τρέχουσες τιμές (web server).....	223
Εικόνα 7-14 Weekly Statistics (web server).....	229
Εικόνα 7-15 Δημιουργία λογαριασμού.....	234
Εικόνα 7-16 Δημιουργία καναλιών	234
Εικόνα 7-17 WriteAPIKey	236
Εικόνα 7-18 Επεξεργασία Fields.....	236
Εικόνα 7-19 Ιδιότητες της απεικόνισης των fields.....	237
Εικόνα 7-20 Παράδειγμα παραμετροποιημένου field	238
Εικόνα 7-21 Επεξεργασία τιμών	238
Εικόνα 7-22 Παρουσίαση τιμών σε ένα πλαίσιο	239
Εικόνα 7-23 Time Control.....	240
Εικόνα 7-24 Matlab Visualizations 1	241
Εικόνα 7-25 Matlab Visualizations 2	242
Εικόνα 7-26 Matlab Visualizations 3	242
Εικόνα 7-27 Public View/Sharing	243
Εικόνα 7-28 Plugins	244
Εικόνα 7-29 Matlab Visualizations 4	246
Εικόνα 7-30 Matlab Visualizations 5	246
Εικόνα 7-31 Gauge	246
Εικόνα 7-32 Ιδιότητες του γραφήματος (Configure Gauge).....	247
Εικόνα 7-33 Gauge	247
Εικόνα 7-34 Γράφημα την ελάχιστη, μέγιστη και μέση τιμή της θερμοκρασίας για την προηγούμενη ώρα	250
Εικόνα 7-35 IFTTT 1	251
Εικόνα 7-36 IFTTT 2	251
Εικόνα 7-37 IFTTT 3	251
Εικόνα 7-38 IFTTT 4	252
Εικόνα 7-39 IFTTT Key	252
Εικόνα 7-40 ThinkSpeak Ιστοσελίδα	253
Εικόνα 7-41 ThinkSpeak Ιστοσελίδες	256
Εικόνα 7-42 ThinkSpeak Ιστοσελίδες	257
Εικόνα 7-43 ThinkSpeak Ιστοσελίδες	258
Εικόνα 7-44 ThinkSpeak Ιστοσελίδες	259
Εικόνα 7-45 ThinkSpeak Ιστοσελίδες	260
Εικόνα 7-46 ThinkSpeak-IFTTT Ιστοσελίδες	264
Εικόνα 7-47 ThinkSpeak-IFTTT Ιστοσελίδες	265

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 2-1 Πλακέτες Arduino της οικογένειας MKR	32
Πίνακας 2-2 Πλακέτες Arduino της οικογένειας Nano	32
Πίνακας 2-3 Πλακέτες Arduino της οικογένειας Portenta.....	33
Πίνακας 2-4 Συναρτήσεις	46
Πίνακας 2-5 Τιμές (μεταβλητές, σταθερές)	47
Πίνακας 2-6 Αριθμητικοί τελεστές.....	47
Πίνακας 2-7 Δομή Ελέγχου	48
Πίνακας 2-8 Further Syntax.....	48
Πίνακας 2-9 Ενδεικτικές Βιβλιοθήκες	49
Πίνακας 2-10 Βήματα δημιουργίας του πρώτου προγράμματος	56
Πίνακας 3-1 Εκδόσεις ESP32	64
Πίνακας 4-1 PINS OUT.....	71
Πίνακας 4-2 Components.....	118
Πίνακας 4-3 AT-Commands κάρτας sim.....	118
Πίνακας 4-4 AT-Commands SMS.....	119
Πίνακας 4-5 AT-Commands GPRS	119
Πίνακας 4-6 AT-Commands GPS	119
Πίνακας 4-7. Υλοποιήσεις XBee S2	122
Πίνακας 4-8. Εκδόσεις Raspberry PI	124

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

1.1 ΓΕΝΙΚΑ

Στη σύγχρονη εποχή, παρατηρούμε μια σαφή τάση προς την αυτοματοποίηση και την ενίσχυση της νοημοσύνης σε κάθε πτυχή της ζωής μας. Περιτριγυριζόμαστε από έξυπνες συσκευές τόσο το σπίτι μας όσο και οι εργασιακό μας χώρο, ενώ οι έξυπνες τεχνολογίες γίνονται ολοένα και πιο ολοκληρωμένο μέρος της καθημερινής μας ρουτίνας.

Μια αυξανόμενη ποικιλία ηλεκτρικών και ηλεκτρονικών προϊόντων συμβάλουν καθημερινά στην άνεση και στην οικονομία των σημερινών σπιτιών/κτιρίων ελέγχοντας τις οικιακές/κτιριακές συσκευές, όπως οι “άσπρες” λεγόμενα συσκευές (white goods), (πλυντήρια, ψυγεία,...), οι “μαύρες” λεγόμενα συσκευές (brown goods) (τηλεοράσεις, video, στερεοφωνικά), οι συσκευές επικοινωνίας, οι συσκευές θέρμανσης και αερισμού, οι συσκευές φωτισμού, οι συσκευές ασφάλειας κ.λ.π. Αυτά τα προϊόντα προσφέρουν ολοένα και έναν μεγαλύτερο αριθμό λειτουργιών, ενώ την ίδια στιγμή είναι ανοικτά σε καινούργιες εφαρμογές.

Σήμερα υπάρχει μια αυξανόμενη ανάγκη για καινούργιες και περισσότερο αυτοματοποιημένες λειτουργίες - υπηρεσίες σε ένα κτιριακό ή οικιακό περιβάλλον. Μερικές από αυτές είναι: διαχείριση της κατανάλωσης της ενέργειας, έλεγχος εσωτερικού ή εξωτερικού φωτισμού, ασφάλεια - επιτήρηση, πυρασφάλεια, έλεγχος της εσωτερικής θερμοκρασίας – θέρμανση/ψύξη, υπηρεσίες για άτομα ειδικής ανάγκης ή τρίτης ηλικίας και πολλές άλλες. Η ολοκλήρωση όμως τέτοιων λειτουργιών απαιτεί από μια ποικιλία συσκευών που υπάρχουν στο κτιριακό περιβάλλον να συνεργάζονται μεταξύ τους χωρίς την επέμβαση του ανθρώπου.

Η τεχνολογία των υπολογιστών καθημερινά γίνεται ολοένα και πιο φτηνή, αυτό έχει σαν αποτέλεσμα όλο και περισσότερες συσκευές της καθημερινότητας, οικιακές/κτιριακές συσκευές να ενσωματώνουν υπολογιστικά συστήματα, με συνέπεια να αποκτούν περισσότερη εξυπνά-

δα, ευκολίες και δυνατότητες. Μέσα από τη διασύνδεση αυτών των συσκευών, μπορεί να φανταστεί κανείς, τι το μπορεί να προσφερθεί στο μέσο άνθρωπο. Όλες αυτές οι λειτουργίες – υπηρεσίες μπορούν να γίνουν πραγματικότητα, μόνο μέσα από ένα ολοκληρωμένο σύστημα οικιακής/κτιριακής αυτοματοποίησης.

Το "έξυπνο σπίτι" αποτελεί έναν όρο που περιγράφει μια οικία εξοπλισμένη με τεχνολογία που επιτρέπει τη ρύθμιση και τον έλεγχο διαφόρων ηλεκτρομηχανολογικών εγκαταστάσεων. Επιπλέον, ένα έξυπνο σπίτι λαμβάνει αποφάσεις και δρα βάση των καθημερινών αναγκών και συνηθειών των κατοίκων του. Αυτό σημαίνει ότι παρέχει πλήρη έλεγχο σε συστήματα ασφαλείας, θέρμανσης, φωτισμού, ηλεκτρικών συσκευών, πολυμέσων και άλλα. Οι λειτουργίες αυτές μπορούν να ενεργοποιηθούν ή να απενεργοποιηθούν με ένα απλό πάτημα κουμπιού, είτε ο ιδιοκτήτης βρίσκεται εντός του σπιτιού είτε σε απόσταση, χάρη στην απομακρυσμένη δυνατότητα ελέγχου.

1.2 ΕΞΥΠΝΟ ΣΠΙΤΙ ΚΑΙ ΟΙΚΙΑΚΟΙ (ΚΤΙΡΙΑΚΟΙ) ΑΥΤΟΜΑΤΙΣΜΟΙ

Η λέξη "έξυπνο" χρησιμοποιήθηκε αρχικά κατά τη δεκαετία του '70 για να περιγράψει τεχνολογικές εξελίξεις, κυρίως στον τομέα των στρατιωτικών εφαρμογών, όπως έξυπνα όπλα που μπορούσαν να καθοδηγούνται προς τον στόχο τους. Τη δεκαετία του '80, η λέξη απέκτησε επιπλέον έννοιες, περιγράφοντας συσκευές που ενσωμάτωναν ολοκληρωμένα κυκλώματα, όπως οι υπολογιστές και οι προηγμένες οικιακές συσκευές. Τον όρο "έξυπνο σπίτι" επινόησε η αμερικανική ομοσπονδία κατασκευαστών σπιτιών το 1984.

Σύμφωνα με τον Aldrich (2003), ένα "έξυπνο σπίτι" μπορεί να οριστεί ως μια κατοικία που είναι εξοπλισμένη με υπολογιστές και τεχνολογία πληροφοριών, η οποία προβλέπει και ανταποκρίνεται στις ανάγκες των ενοίκων. Αυτό το σπίτι "εργάζεται" για να παρέχει ανέσεις, ευκολίες και ασφάλεια μέσω της διαχείρισης της τεχνολογίας του σπιτιού και της σύνδεσης με τον έξω κόσμο. Ουσιαστικά, αυτό το σπίτι "σκέπτεται" και "ενεργεί" βάση των καθημερινών αναγκών και συνηθειών των χρηστών. Ένα βασικό στοιχείο του έξυπνου σπιτιού είναι η δυνατότητα διαχείρισης του, είτε ο ιδιοκτήτης βρίσκεται μέσα στο κτίριο, είτε μακριά από αυτό, μέσω κινητού τηλεφώνου ή υπολογιστή, και η εποπτεία και έλεγχος συγκεκριμένων λειτουργιών του. Ένα «έξυπνο σπίτι» είναι μια κατοικία που ενσωματώνει ένα δίκτυο επικοινωνίας το οποίο συνδέει ηλεκτρικές συσκευές και υπηρεσίες, παρέχοντας τη δυνατότητα απομακρυσμένου ελέγχου, παρακολούθησης και πρόσβασης σε αυτές. Όταν αναφερόμαστε στον απομακρυσμένο έλεγχο σε ένα έξυπνο σπίτι, εννοούμε τη δυνατότητα να ελέγχουμε τις συσκευές και τις υπηρεσίες του σπιτιού από μακριά, είτε βρισκόμαστε εντός του σπιτιού είτε όχι. Αυτή η δυνατότητα είναι κεντρική στην έννοια των έξυπνων σπιτιών, καθώς πολλές υλοποιήσεις περιλαμβάνουν δικτύωση, αλληλεπίδραση συσκευών και αυτοματισμό. Επομένως, όταν μιλάμε για ένα έξυπνο σπίτι, αναφερόμαστε στο συνολικό αποτέλεσμα των υπηρεσιών που προσφέρει, παρά στα μεμονωμένα στοιχεία του, όπως οι αισθητήρες ή οι διακόπτες.

Οι σύγχρονες τεχνολογίες αυτοματισμού προσφέρουν πολλαπλά οφέλη σε ένα έξυπνο σπίτι σε σύγκριση με τους παραδοσιακούς προγραμματιζόμενους χρονοδιακόπτες. Ένα πραγματικά έξυπνο αυτοματοποιημένο σπίτι διακρίνεται από τη δυνατότητά του να δημιουργεί πολύπλοκες συνθήκες φωτισμού, να παρέχει εξελιγμένο έλεγχο ασφαλείας, να διανέμει εικόνα και ήχο, καθώς και να διαχειρίζεται αποτελεσματικά την ενέργεια και πολλά άλλα.

Οι αυτοματισμοί ενός έξυπνου είναι πάρα πολλοί. Επιγραμματικά μερικοί από αυτούς αναφέρονται παρακάτω:

- Ασφάλεια - Προστασία

Αισθητήρες για τον έλεγχο της κατάστασης των θυρών & παραθύρων μπορεί να υπάρχουν σε κάθε θύρα ή παράθυρο, ανιχνευτές κίνησης, κάμερες, συγχρόνως μπορεί να υπάρχουν αισθητήρες καπνού, πυρκαγιάς, απότομης αλλαγής της θερμοκρασίας, πλημμύρας, προσφέροντας προς τον καταναλωτή και αυτόματη ειδοποίηση μέσω του τηλεφωνικού δικτύου, σε περίπτωση συναγερμού, των ιδιοκτητών ή της αστυνομίας.

- Θέρμανση – Κλιματισμός

Αισθητήρες θερμοκρασίας μπορεί να ελέγχουν 24-ώρες το 24-ώρο τη θερμοκρασία σε κάθε χώρο ξεχωριστά, ανάλογα με τις καιρικές συνθήκες ή τις απαιτήσεις του καταναλωτή, ανοίγοντας και κλείνοντας θερμαντικά και κλιματιστικά σώματα.

- Συσκευές Εικόνας & Ήχου – Ψυχαγωγία

Έλεγχος συσκευών ήχου & εικόνας ελεγχόμενες από οποιοδήποτε δωμάτιο του σπιτιού, πρόσβαση σε βάσεις δεδομένων, επικοινωνίες με άλλα άτομα και παιχνίδια με άτομα που βρίσκονται σε άλλο δωμάτιο του σπιτιού, ή σε άλλο απομακρυσμένο σπίτι.

- Φωτισμός

Έλεγχος του φωτισμού σε οποιοδήποτε χώρο του σπιτιού/κτιρίου ανάλογα με τις καιρικές συνθήκες, την ώρα της μέρας, την εποχή, ή τις απαιτήσεις του καταναλωτή.

- Άσπρες Συσκευές

Έλεγχος των άσπρων συσκευών του σπιτιού, όπως η ηλεκτρική κουζίνα, το ψυγείο, το πλυντήριο πιάτων & ρούχων και άλλες.

- Περσίδες – Μηχανοκίνητες Συσκευές

Έλεγχος των περσίδων ή οποιαδήποτε μηχανοκίνητης συσκευής, η οποία βρίσκεται σε οποιοδήποτε χώρο του σπιτιού/κτιρίου. Π.χ. οι περσίδες των παραθύρων να κλείνουν και να ανοίγουν αυτόματα ανάλογα με φως του ήλιου.

- Συστήματα Νερού

Έλεγχος θερμοκρασίας νερού, στάθμης νερού, ποιότητας νερού, αυτόματο πότισμα φυτών ανάλογα με τις καιρικές συνθήκες και την υγρασία του χώματος και γενικά ότι έχει να κάνει με αυτοματισμούς που έχουν σχέση με το νερό.

- Φροντίδα Οικιακών Κατοικίδιων

Αυτόματο τάισμα κατοικίδιων ζώων, προσδιορισμός της θέσης ενός κατοικίδιου, και γενικά ότι έχει σχέση με τη φροντίδα των κατοικίδιων ζώων.

- Επικοινωνίες

Οι τηλεφωνικές συσκευές είναι μεταξύ τους διασυνδεδεμένες, ενδοεπικοινωνία.

- Χρήση - Διαχείριση της ενέργειας

Καταμέτρηση και βελτιστοποίηση της κατανάλωσης της ενέργειας ανάλογα με τις ανάγκες του πελάτη και με το κόστος της ενέργειας ώστε να υπάρχει αποτελεσματικότερη θέρμανση χώρου, θέρμανση νερού και χρήση των οικιακών συσκευών.

- Δίκτυα Υπολογιστών

Μεταφορά και ανταλλαγή αρχείων μεταξύ των υπολογιστικών συσκευών του σπιτιού.

- Ιατρικές Εφαρμογές

Έλεγχος της κατάστασης των ιατρικών συσκευών, συλλογή μετρήσεων & δεδομένων από ιατρικές συσκευές, αυτόματη παραγγελία φαρμάκων μέσω της τηλεόρασης, αυτόματη ειδοποίηση συγγενικού προσώπου ή του γιατρού σε περίπτωση ιατρικής ανάγκης – με τη βοήθεια – πατώντας ένα κουμπί, αυτόματη καταμέτρηση ιατρικών μεγεθών (π.χ πίεση) και απόφαση για δοσοληψία του κατάλληλου φαρμάκου. Ενώ μπορεί να υπάρχουν και εφαρμογές για άτομα με

ειδικές ανάγκες, αισθητήρες κίνησης στο μπάνιο, ασύρματα κουμπιά πρώτης βοήθειας και άλλα πολλά

1.3 ΑΝΤΙΚΕΙΜΕΝΟ

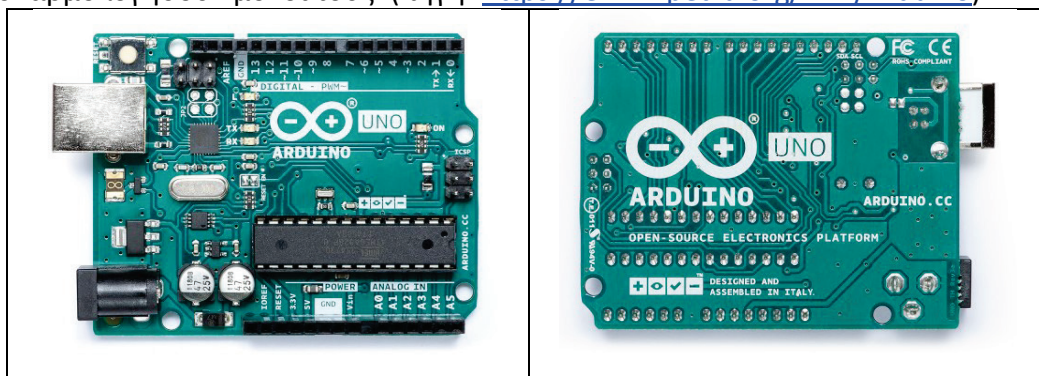
Αντικείμενο αυτού του βιβλίου είναι παρουσίαση διαφόρων σεναρίων αυτοματισμού και ελέγχου σε ένα κτιριακό περιβάλλον βασισμένα στις πλατφόρμες Arduino και ESP32. Παρουσιάζονται οι πλατφόρμες (ARDUINO/ ESP32), το απαραίτητο υλικό και λογισμικό για την υλοποίηση διαφόρων οικιακών σεναρίων αυτοματισμού και ελέγχου, το απαραίτητο λογισμικό για τον απομακρυσμένο έλεγχο, εποπτεία και τηλεχειρισμό διαφόρων οικιακών συσκευών/οικιακών σεναρίων αυτοματισμού και ελέγχου, αναλυτικά παραδείγματα διασύνδεσης υλικού και υλοποίησης λογισμικού.

ΚΕΦΑΛΑΙΟ 2

ΠΛΑΤΦΟΡΜΑ ARDUINO – ΥΛΙΚΟ & ΛΟΓΙΣΜΙΚΟ

2.1 ΕΙΣΑΓΩΓΗ

Το Arduino είναι ένας μικροελεγκτής μονής πλακέτας, που αποτελείται από μια απλή μητρική πλακέτα με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους. Μπορεί να προγραμματιστεί με τη γλώσσα προγραμματισμού Wiring. Το Arduino χρησιμοποιείται για τη δημιουργία ανεξάρτητων διαδραστικών αντικειμένων και μπορεί να συνδεθεί με υπολογιστή μέσω προγραμμάτων. Οι περισσότερες εκδόσεις του Arduino είναι διαθέσιμες προσαρμοσμένες, ενώ πληροφορίες για το υλικό και διαγράμματα είναι ελεύθερα προσβάσιμα για αυτούς που επιθυμούν να το συναρμολογήσουν μόνοι τους. (πηγή: <https://el.wikipedia.org/wiki/Arduino>)



Εικόνα 2-1 Πλακέτα Arduino v.R3 (2 πλευρές)

(Arduino Docs | Arduino Documentation, n.d.)

Το 2005 ξεκίνησε ένα σχέδιο για τη δημιουργία μιας συσκευής ελέγχου προγραμμάτων διαδραστικών σχεδίων για μαθητές, η οποία θα ήταν πιο οικονομική από άλλα πρωτότυπα συστήματα που ήταν διαθέσιμα εκείνη την περίοδο. Οι ιδρυτές Massimo Banzi και David Cueartielles επέλεξαν το όνομα "Αρντουίνιο" από το χωριό Arduin της Ιταλίας και ξεκίνησαν την παραγωγή πλακετών σε ένα μικρό εργοστάσιο στην περιοχή Pedemonti της βορειοδυτικής Ιτα-

λίας, κοντά στην Τορίνο. Το σχέδιο Arduino αποτελεί μια παραλλαγή της πλατφόρμας Wiring για λογισμικό ανοικτού κώδικα και προγραμματίζεται χρησιμοποιώντας μια γλώσσα βασισμένη στο Wiring, με παρόμοια σύνταξη και βιβλιοθήκες με την C++, αλλά με απλοποιήσεις και αλλαγές. Επίσης, παρέχει ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE). (πηγή: <https://el.wikipedia.org/wiki/Arduino> και <https://www.arduino.cc>)

2.2 ΥΛΙΚΟ (HARDWARE)

2.2.1 Γενικά

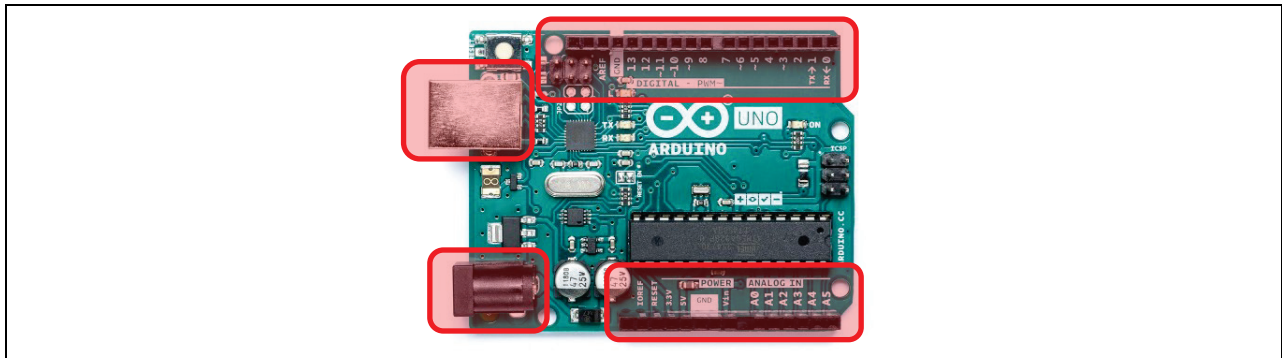
Η πλακέτα Arduino αποτελείται από έναν μικροελεγκτή Atmel AVR. Στις πιο πρόσφατες εκδόσεις, χρησιμοποιούνται τα μοντέλα ATmega328 και ATmega168, ενώ στις παλαιότερες εκδόσεις χρησιμοποιείται το μοντέλο ATmega8. Η πλακέτα περιλαμβάνει επιπλέον εξαρτήματα που βοηθούν τον χρήστη στον προγραμματισμό και την ενσωμάτωσή της σε άλλα κυκλώματα. Κάθε πλακέτα περιλαμβάνει ένα γραμμικό ρυθμιστή τάσης 5V και έναν κρυσταλλικό ταλαντωτή 16MHz. Σε ορισμένες παραλλαγές χρησιμοποιείται επίσης κεραμικός αντηχητής. Ο μικροελεγκτής έχει εγκατασταθεί με έναν bootloader, που επιτρέπει τον προγραμματισμό του χωρίς την ανάγκη εξωτερικού προγραμματιστή.

Οι πλακέτες Arduino μπορούν να προγραμματιστούν μέσω μιας σειριακής σύνδεσης, όπως η RS-232. Ωστόσο, ο τρόπος με τον οποίο επιτυγχάνεται αυτή η σύνδεση διαφέρει ανάλογα με την εκάστοτε υλοποίηση της υποδομής του Arduino. Οι πλακέτες Arduino που χρησιμοποιούν σειριακή σύνδεση περιλαμβάνουν ένα απλό κύκλωμα επιπέδου (level shifter) που μετατρέπει το σήμα RS-232 σε σήμα TTL. Ωστόσο, σήμερα η προγραμματιστική σύνδεση των Arduino γίνεται κυρίως μέσω USB. Αυτό επιτυγχάνεται με τη χρήση προσαρμογέων USB-to-Serial chip, όπως το FTDI FT232, που επιτρέπουν τη μετατροπή του σήματος USB σε σήμα σειριακής επικοινωνίας που κατανοείται από το μικροελεγκτή του Arduino. (πηγή: <https://el.wikipedia.org/wiki/Arduino> και <https://www.arduino.cc>)

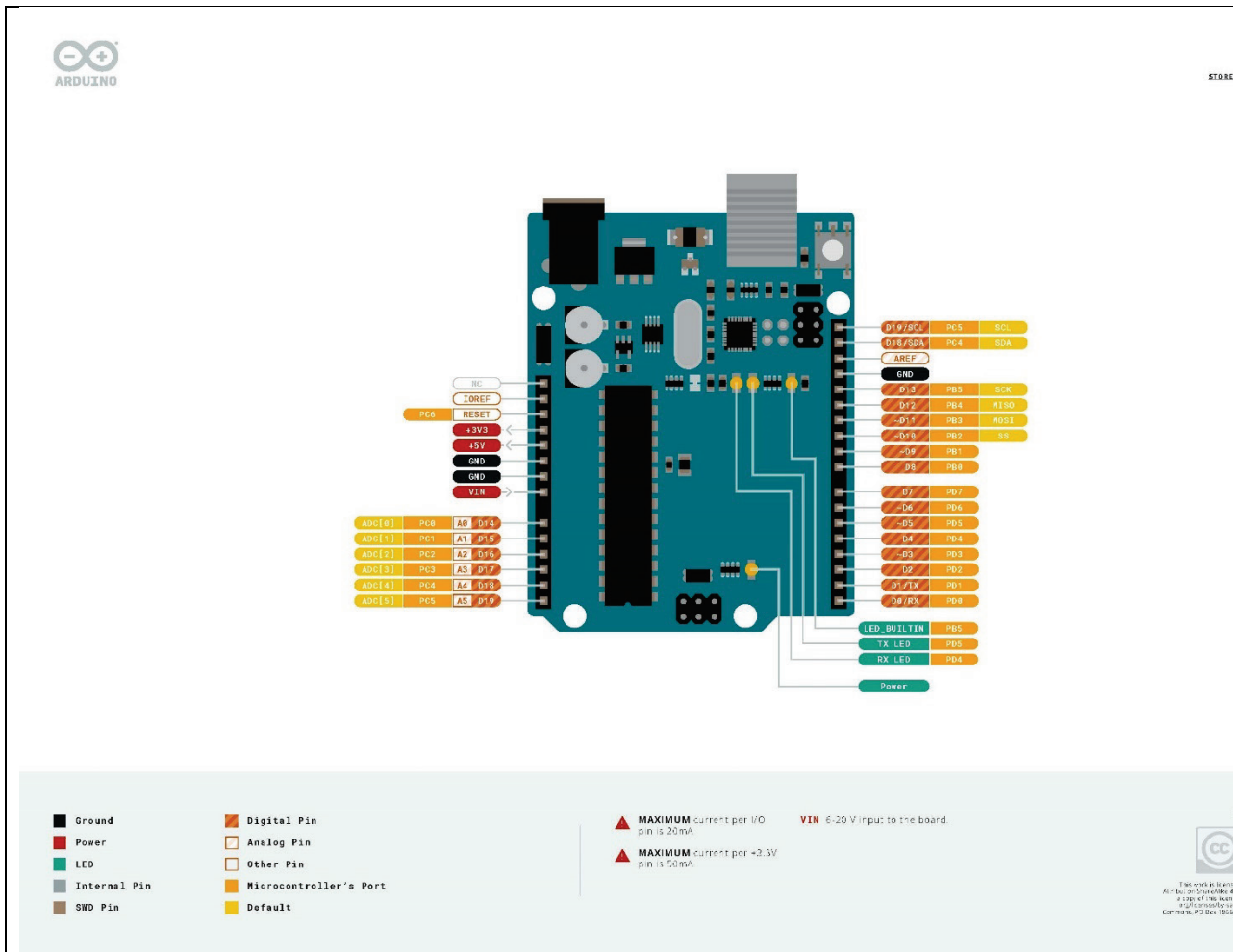
2.2.2 Επεξήγηση Βασικών Ακροδεκτών Arduino

2.2.2.1 Γενικά

Σε αυτήν την ενότητα θα αναλυθούν οι ακροδέκτες που βρίσκονται στην πάνω πλευρά του Arduino.



Εικόνα 2-2 Ακροδέκτες Arduino

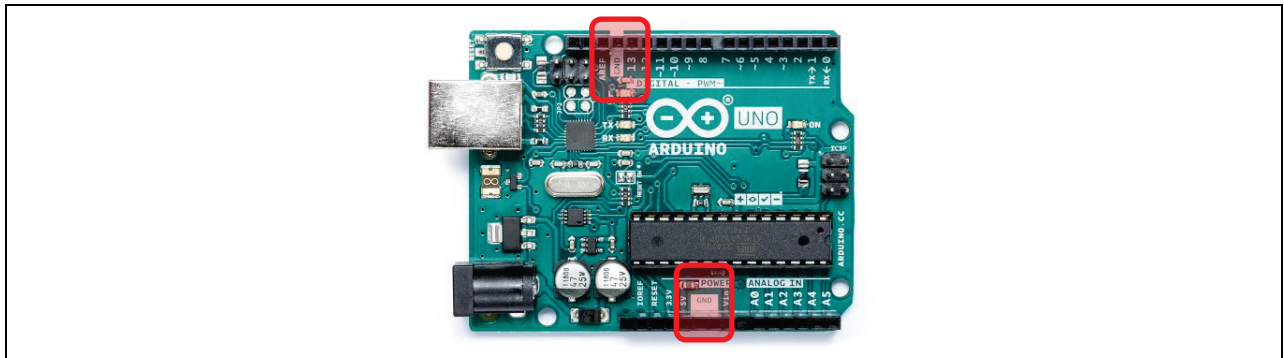


Εικόνα 2-3 Διάγραμμα Ακροδεκτών Arduino

(Arduino Docs | Arduino Documentation, n.d.)

2.2.2.2 Ακροδέκτες Γείωσης (GND pins)

Υπάρχουν αρκετοί ακροδέκτες γείωσης (GND) στο Arduino, οι οποίοι μπορούν να χρησιμοποιηθούν για τη γείωση του κυκλώματος.



Εικόνα 2-4 Ακροδέκτες Γείωσης Arduino

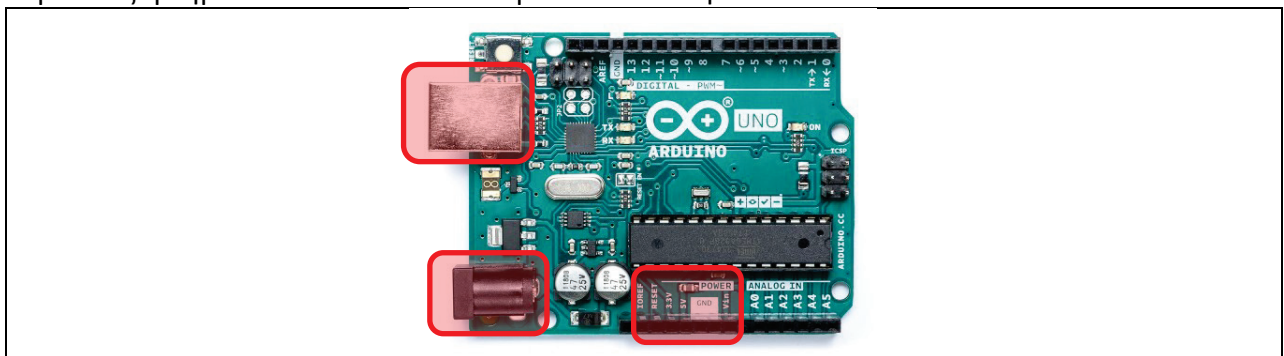


Εικόνα 2-5 Ακροδέκτες Γείωσης Arduino

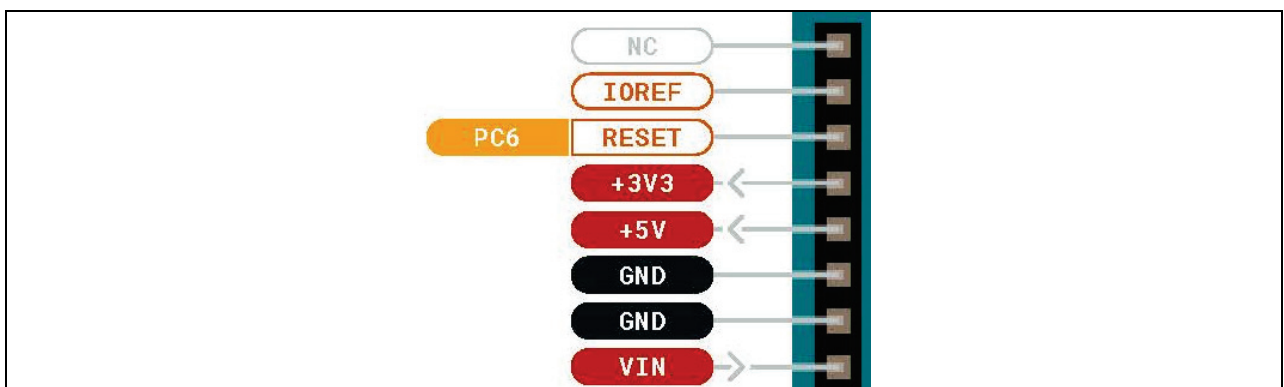
(Arduino Docs | Arduino Documentation, n.d.)

2.2.2.3 Ακροδέκτες Τροφοδοσίας (Power Supply pins)

Η πλακέτα Arduino Uno τροφοδοτείται από εξωτερική πηγή. Μπορείτε να τροφοδοτήσετε ορισμένα εξαρτήματα που είναι συνδεδεμένα πάνω στην πλακέτα.



Εικόνα 2-6 Ακροδέκτες Τροφοδοσίας Arduino



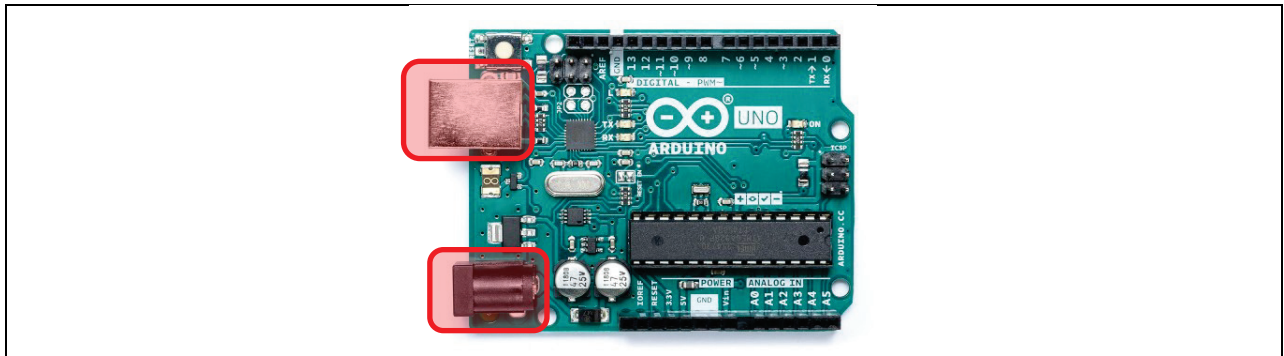
Εικόνα 2-7 Ακροδέκτες Τροφοδοσίας Arduino

(Arduino Docs | Arduino Documentation, n.d.)

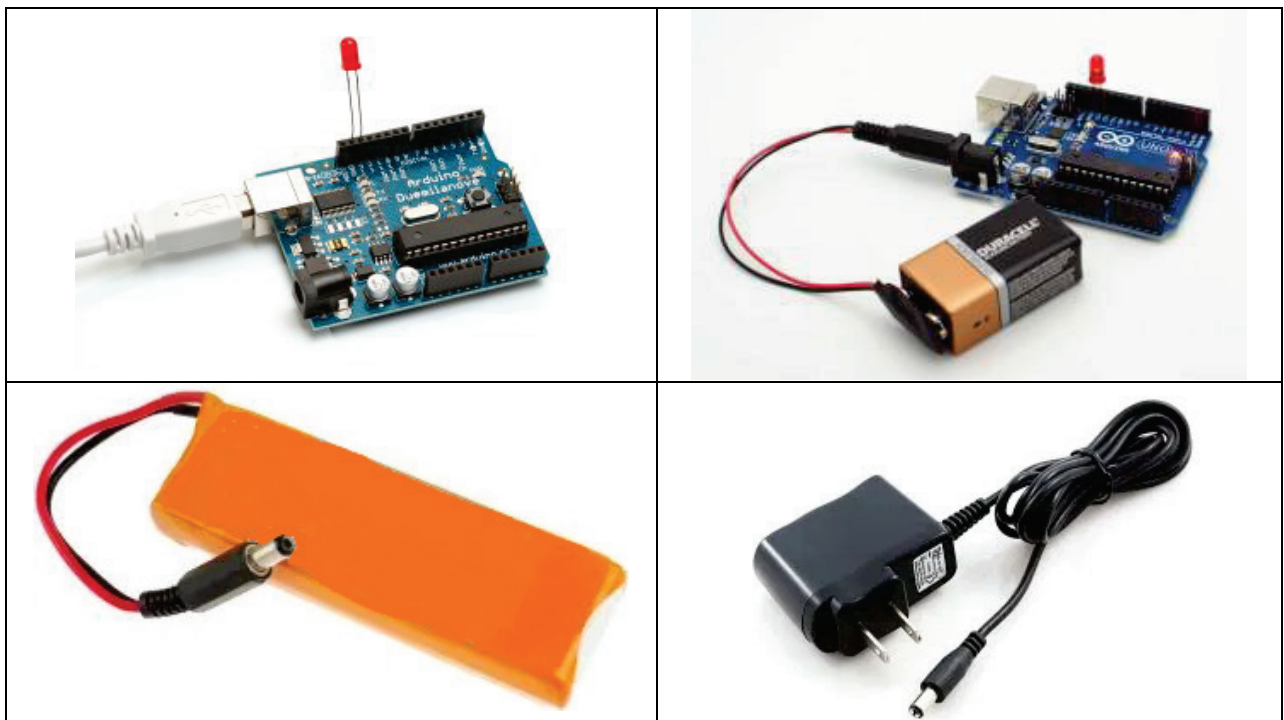
Για την τροφοδοσία του πίνακα Arduino Uno, υπάρχουν διαφορετικές επιλογές:

- Μπορείτε να συνδέσετε την πλακέτα Arduino στον υπολογιστή σας χρησιμοποιώντας ένα καλώδιο USB.

- Μπορείτε να χρησιμοποιήσετε μία μπαταρία ή και επαναφορτιζόμενη μπαταρία.
- Μπορείτε να χρησιμοποιήσετε ένα τροφοδοτικό με έξοδο DC 7-12V.



Εικόνα 2-8 Ακροδέκτες Τροφοδοσίας Arduino



Εικόνα 2-9 Διαφορετικές Τύπου Τροφοδοσίες Arduino

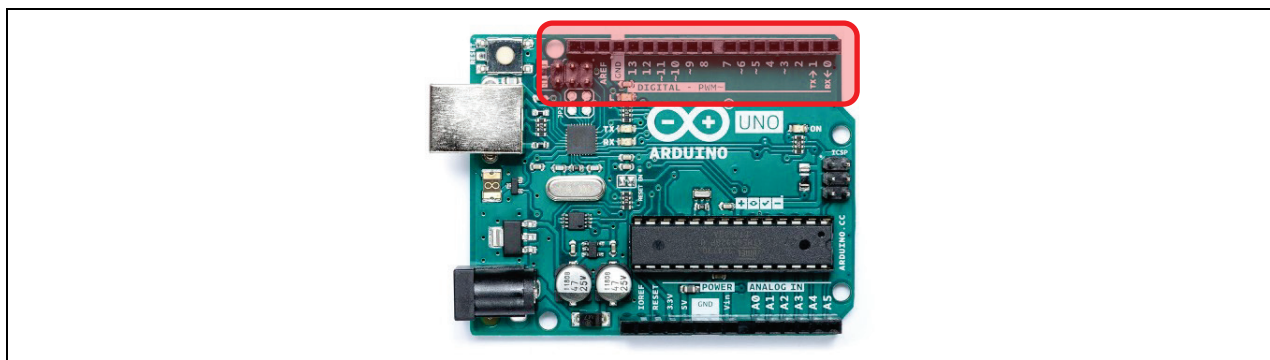
(<https://randomnerdtutorials.com>, n.d.)

2.2.2.4 Ψηφιακοί Ακροδέκτες (Digital pins)

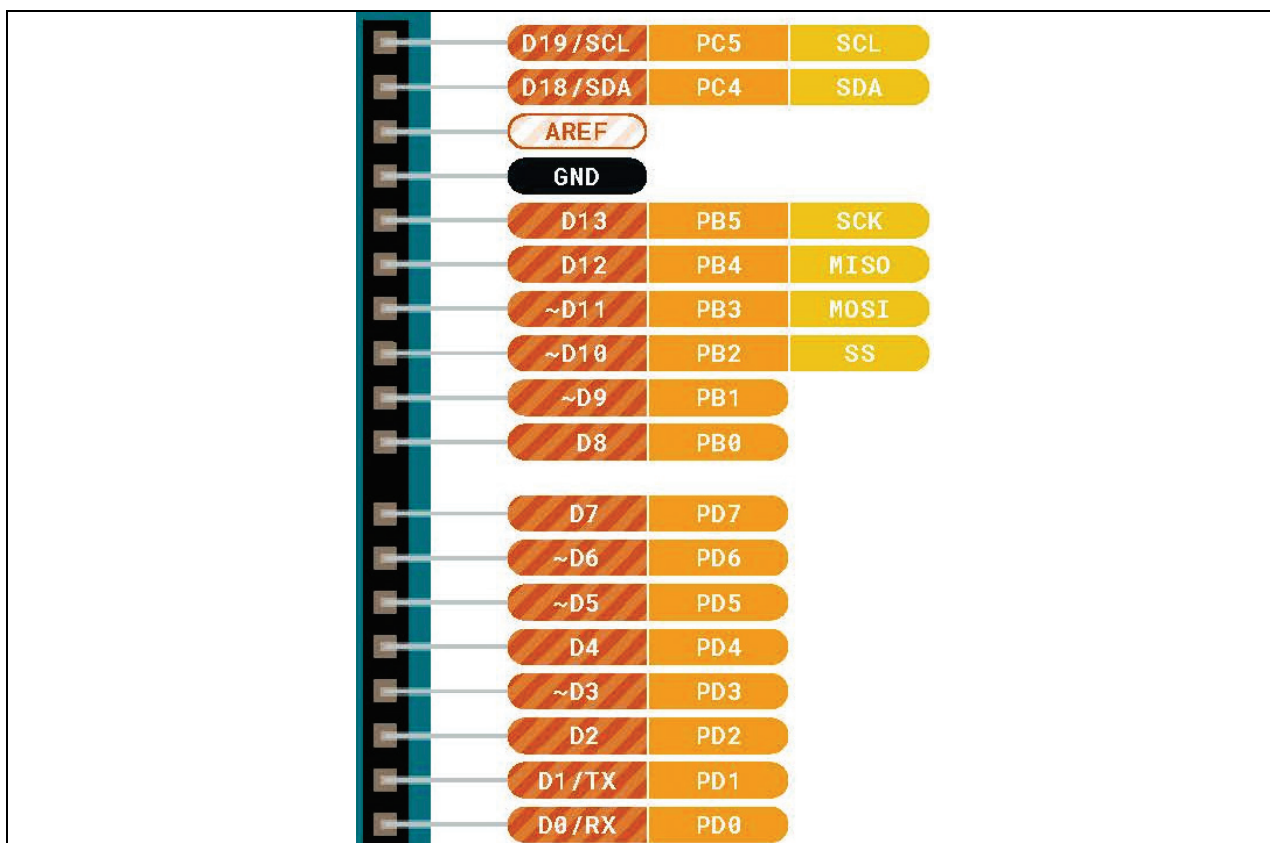
Η πλακέτα Arduino Uno έχει 14 ψηφιακούς ακροδέκτες (digital pins). Οι ψηφιακοί ακροδέκτες χρησιμοποιούνται για να διαβάσουμε δεδομένα από ορισμένα στοιχεία, όπως αισθητήρες και για να γράψουμε δεδομένα σε άλλα στοιχεία, όπως ενεργοποιητές.

Ένας ψηφιακός ακροδέκτης μπορεί να έχει μόνο 2 καταστάσεις, υψηλή (high) ή χαμηλή (low).

- χαμηλή (low).σημαίνει ότι η τάση στον πείρο είναι 0V.
- υψηλή (high) σημαίνει Vcc, που είναι 5V εδώ για το Arduino Uno.



Εικόνα 2-10 Ψηφιακοί Ακροδέκτες Arduino



Εικόνα 2-11 Ψηφιακοί Ακροδέκτες Arduino

(Arduino Docs | Arduino Documentation, n.d.)

Για να χρησιμοποιηθεί ένας ψηφιακός ακροδέκτης (pin) σε μια πλακέτα Arduino, πρέπει να διαμορφωθεί η λειτουργία του. Ο ψηφιακός ακροδέκτης μπορεί να είναι είτε σε λειτουργία εισόδου (input) είτε σε λειτουργία εξόδου (output). Όταν ο ακροδέκτης βρίσκεται σε λειτουργία εισόδου, μπορεί να διαβάζει δεδομένα από εξωτερικές πηγές. Αντίστοιχα, όταν ο ακροδέκτης βρίσκεται σε λειτουργία εξόδου, μπορεί να γράφει δεδομένα σε εξωτερικές συσκευές. Ο ορισμός της λειτουργίας του ακροδέκτη γίνεται συνήθως στη συνάρτηση "setup()" του προγράμματος Arduino, χρησιμοποιώντας τη συνάρτηση "pinMode()". Για την ανάγνωση της κατάστασης του ακροδέκτη (pin), χρησιμοποιείται η συνάρτηση "digitalRead()", ενώ για την εγγραφή δεδομένων στον ακροδέκτη (pin) χρησιμοποιείται η συνάρτηση "digitalWrite()".

Όταν ένας ακροδέκτης έχει ρυθμιστεί σε λειτουργία εισόδου (input), μπορείτε να διαβάσετε την κατάστασή του χρησιμοποιώντας τη συνάρτηση "digitalRead()". Η κατάσταση του ακροδέκτη μπορεί να είναι είτε χαμηλή (low) είτε υψηλή (high). Στην Arduino, οποιαδήποτε τάση που

μετράται κάτω από τα 0,8V θεωρείται ως χαμηλή κατάσταση (low), ενώ οποιαδήποτε τάση μεγαλύτερη από 2V θεωρείται ως υψηλή κατάσταση (high). Τιμές μεταξύ αυτών των ορίων θεωρούνται ανεπαρκείς για να καθορίσουν με βεβαιότητα την κατάσταση του ακροδέκτη και μπορεί να ερμηνευθούν ως χαμηλή ή υψηλή, ανάλογα με τον ακριβή μικροελεγκτή ή τις συνθήκες περιβάλλοντος. (πηγή: <https://el.wikipedia.org/wiki/Arduino> και <https://www.arduino.cc>)

2.2.2.5 PWM (Pulse Width Modulation) - Διαμόρφωση πλάτους παλμού

Ορισμένοι από τους ψηφιακούς ακροδέκτες μπορούν να χρησιμοποιηθούν για να δημιουργηθεί ένας καθοδηγούμενος πλάτους παλμός (PWM). Η τεχνική του καθοδηγούμενου πλάτους παλμού (Pulse Width Modulation - PWM) ουσιαστικά αναφέρεται στον τρόπο με τον οποίο μπορούμε να παράγουμε μια επιθυμητή τάση χρησιμοποιώντας μόνο δύο καταστάσεις, χαμηλή και υψηλή (0V/5V). Το πλάτος του παλμού ελέγχεται με τη συχνότητα (500Hz για το Arduino Uno). Στη συνέχεια, η παράμετρος του κύκλου εργασίας (duty cycle) υποδεικνύει το ποσοστό του χρόνου που ο παλμός παραμένει σε κατάσταση χαμηλής τάσης ή υψηλής τάσης. Μια απλουστευμένη ερμηνεία του PWM στο Arduino είναι ότι η συχνή εναλλαγή μεταξύ των καταστάσεων χαμηλής και υψηλής τάσης παράγει μια μέση τιμή τάσης στην έξοδο. Για παράδειγμα, σε έναν κύκλο εργασίας 50% (50% του χρόνου υψηλή τάση, 50% του χρόνου χαμηλή τάση), η τάση εξόδου θα ήταν 2,5V.

Η χρήση του PWM είναι δυνατή μόνο σε συγκεκριμένους ακροδέκτες που συνοδεύονται από το σύμβολο "~" δίπλα στον αριθμό τους. Στο Arduino Uno, οι ακροδέκτες που υποστηρίζουν την τεχνική PWM είναι οι ακροδέκτες 3, 5, 6, 9, 10 και 11. Συνεπώς, έχουμε έξι ακροδέκτες στους οποίους μπορούμε να χρησιμοποιήσουμε τη λειτουργία `analogWrite()` για να δημιουργήσουμε PWM. Αυτό αποτελεί ιδιαίτερα χρήσιμο εργαλείο για τον έλεγχο ορισμένων ενεργοποιητών, οι οποίοι απαιτούν απόλυτο συντονισμό και δεν μπορούν να ενεργοποιηθούν ή απενεργοποιηθούν απλά. Για παράδειγμα, μπορούμε να χρησιμοποιήσουμε τη λειτουργία `analogWrite()` για να προσαρμόσουμε τη φωτεινότητα ενός LED. (πηγή: <https://el.wikipedia.org/wiki/Arduino> και <https://www.arduino.cc>)

2.2.2.6 Ακροδέκτες Διακοπών (Interrupt pins)

Στο Arduino Uno μόνο οι ψηφιακοί ακροδέκτες 2 και 3 μπορούν να χρησιμοποιηθούν ως ακροδέκτες διακοπών (interrupt pins). Οι ακροδέκτες διακοπών επιτρέπουν τη δημιουργία συναρτήσεων που εκτελούνται αυτόματα όταν συμβαίνει μια συγκεκριμένη συμβάντος, όπως η ενέργεια πατήματος ενός κουμπιού. Όταν δημιουργείτε ένα πρόγραμμα Arduino, ο κώδικας εκτελείται γραμμή προς γραμμή και δεν υπάρχει πολυνηματισμός (multithreading). Αυτό σημαίνει ότι η εκτέλεση του προγράμματος σας διακόπτεται όταν η εκτέλεση μεταβαίνει σε μια λειτουργία που καλείται από μια διακοπή. Κατά τη διάρκεια της λειτουργίας διακοπής, η εκτέλεση του κυρίως προγράμματος παύει και επανέρχεται μόνο όταν η λειτουργία διακοπής ολοκληρωθεί. Αυτό σημαίνει ότι μπορείτε να χρησιμοποιήσετε τη λειτουργία διακοπής για να ανταποκριθείτε αμέσως σε μια ενέργεια όπως το πάτημα ενός κουμπιού, χωρίς να χρειάζεται να διαβάζετε συνεχώς την κατάσταση του κουμπιού. Ωστόσο, αυτό δεν επιλύει το ζήτημα του πολυνηματισμού ή του ταυτόχρονου εκτελεστικού κώδικα. Όταν η εκτέλεση μεταβαίνει σε μια

λειτουργία διακοπής, η εκτέλεση του κυρίως προγράμματος παύει και επανέρχεται μόνο όταν η λειτουργία διακοπής ολοκληρωθεί.

Για την αποφυγή προβλημάτων και για να διασφαλίσετε τη σωστή λειτουργία του προγράμματος, είναι σημαντικό να κρατάτε τη διάρκεια της λειτουργίας διακοπής όσο το δυνατόν συντομότερη. Αποφύγετε την εκτέλεση μακροχρόνιων διαδικασιών μέσα στη λειτουργία διακοπής, καθώς αυτό μπορεί να προκαλέσει προβλήματα στην απόκριση του συστήματος.

Για να χρησιμοποιήσετε τους ακροδέκτες διακοπών στο Arduino Uno, μπορείτε να χρησιμοποιήσετε τη συνάρτηση `attachInterrupt()`. Αυτή η συνάρτηση δέχεται τρία ορίσματα: τον αριθμό του ακροδέκτη διακοπής, τη συνάρτηση που θα κληθεί όταν συμβεί η διακοπή και τον τρόπο λειτουργίας της διακοπής (ανόδου, φθίνουσας ή και των δύο φορών). (πηγή:

<https://el.wikipedia.org/wiki/Arduino> και <https://www.arduino.cc>)

Για παράδειγμα, αν έχετε συνδέσει ένα κουμπί στον ψηφιακό ακροδέκτη 2, μπορείτε να χρησιμοποιήσετε την `attachInterrupt()` ως εξής:

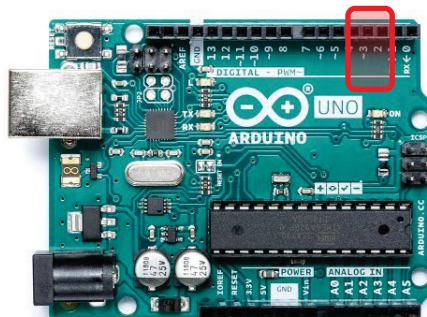
```
const int interruptPin = 2;

void setup() {
  pinMode(interruptPin, INPUT_PULLUP); // Ορίζει τον ακροδέκτη ως είσοδο με ενσωματωμένη
  // αντίσταση pull-up
  attachInterrupt(digitalPinToInterrupt(interruptPin), myInterruptFunction, CHANGE);
}

void loop() {
  // Κύριος κώδικας που εκτελείται συνεχώς
}

void myInterruptFunction() {
  // Κώδικας που εκτελείται όταν συμβαίνει η διακοπή
  // Προσέξτε να κρατήσετε αυτήν τη λειτουργία σύντομη και γρήγορη
}
```

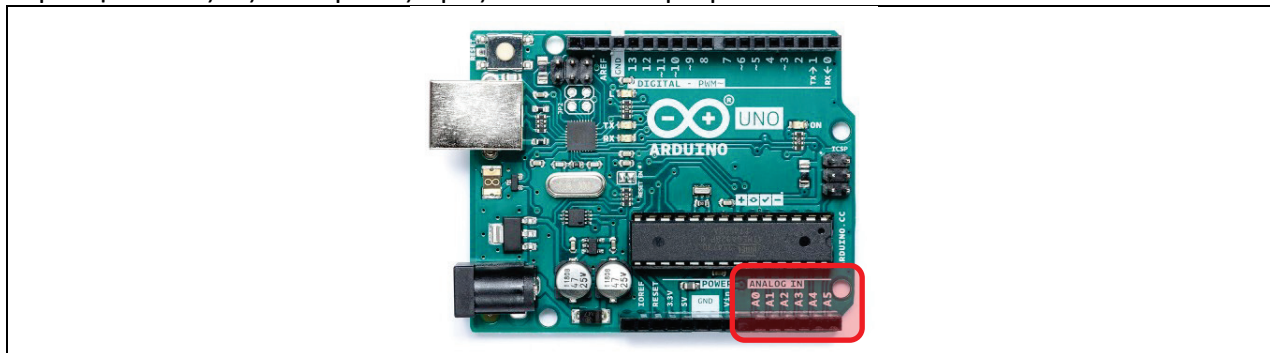
Κώδικας 2-1 Παράδειγμα Κώδικα `attachInterrupt()`



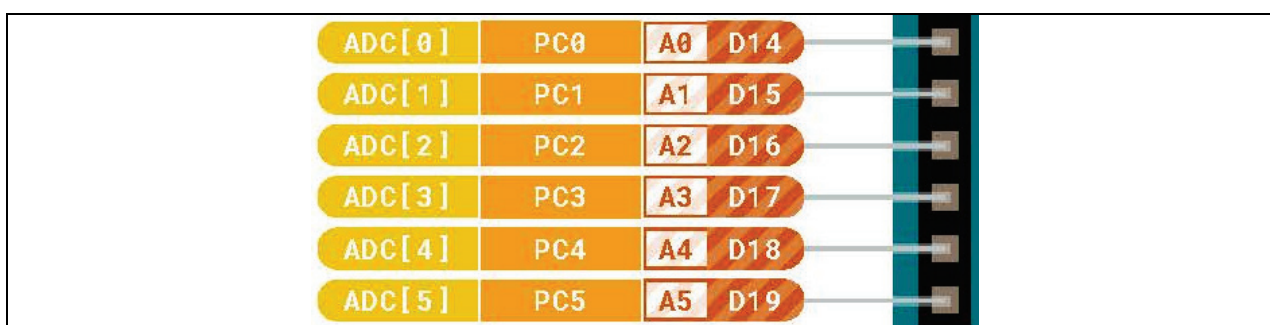
Εικόνα 2-12 Ακροδέκτες Διακοπών

2.2.2.7 Αναλογικοί Ακροδέκτες (Analog pins)

Η πλακέτα Arduino Uno διαθέτει 6 αναλογικούς ακροδέκτες (A0 έως A5), οι οποίοι είναι χρήσιμοι για την ανάγνωση τιμών που δεν μπορούν να εκφραστούν μόνο με τα δυαδικά 0 ή 1. Ας υποθέσουμε ότι έχετε ένα ποτενσιόμετρο και θέλετε να αναγνωρίσετε το ποσοστό της τρέχουσας τιμής του ποτενσιόμετρου. Με έναν ψηφιακό ακροδέκτη, θα μπορούσατε να γνωρίζετε μόνο αν το ποτενσιόμετρο βρίσκεται στην ελάχιστη ή τη μέγιστη θέση, αλλά δεν θα είχατε πληροφορίες για τις ενδιάμεσες τιμές. Αντίθετα, με τους αναλογικούς ακροδέκτες, έχετε πρόσβαση σε όλες τις ενδιάμεσες τιμές του ποτενσιόμετρου.



Εικόνα 2-13 Αναλογικοί Ακροδέκτες Arduino



Εικόνα 2-14 Αναλογικοί Ακροδέκτες Arduino

(Arduino Docs | Arduino Documentation, n.d.)

Οι αναλογικοί ακροδέκτες στην πλακέτα Arduino Uno χρησιμοποιούνται για ανάγνωση αναλογικών τιμών. Η λειτουργία τους είναι η εξής: λαμβάνουν μια τάση εισόδου και τη διαβάζουν. Ας υποθέσουμε ότι ένας αναλογικός ακροδέκτης διαβάζει μια τάση των 3.0V. Στη συνέχεια, ένας μετατροπέας Αναλογικού σε Ψηφιακό (ADC - Analog to Digital Converter) μετατρέπει αυτήν την αναλογική τιμή σε μια τιμή που το πρόγραμμα της Arduino μπορεί να κατανοήσει, δηλαδή μια ψηφιακή τιμή.

Η πλακέτα Arduino Uno διαθέτει έναν ADC 10 bit. Αυτό σημαίνει ότι η ανάλυση του ADC είναι $2^{10} = 1024$. Έτσι, η τιμή που λαμβάνετε όταν διαβάζετε δεδομένα από έναν αναλογικό ακροδέκτη είναι από το 0 έως το 1023. Συνεπώς, για μια τάση εισόδου 3.0V, η αναγνωσθείσα τιμή θα είναι 614. Ανάλογα με την τιμή που λαμβάνει ο ακροδέκτης, αυτή μετατρέπεται σε μια τιμή από 0 έως 1023 (συνολικά 1024 διαφορετικές τιμές), παρέχοντάς μας την ακρίβεια που μπορούμε να έχουμε. Είναι επίσης σημαντικό να σημειωθεί ότι η λειτουργία `analogWrite()` για την παραγωγή PWM είναι διαθέσιμη μόνο για ορισμένους από τους ψηφιακούς ακροδέκτες και όχι για τους αναλογικούς ακροδέκτες.

Τέλος, ένας αναλογικός ακροδέκτης μπορεί να χρησιμοποιηθεί και ως ψηφιακός ακροδέκτης (αντίθετα, όμως, δεν ισχύει). Αφού ο αναλογικός ακροδέκτης μπορεί να διαβάσει οποιαδήποτε τιμή μεταξύ 0V και 5V. Ωστόσο, για να χρησιμοποιηθεί ως ψηφιακός ακροδέκτης, μπορούμε

να θέσουμε ένα κατώφλι τάσης για τον προσδιορισμό των ψηφιακών επιπέδων. Για παράδειγμα, μπορούμε να θεωρήσουμε ότι τιμές κάτω από 0.8V ερμηνεύονται ως χαμηλή κατάσταση (low), ενώ τιμές άνω των 2V ερμηνεύονται ως υψηλή κατάσταση (high). (πηγή: <https://el.wikipedia.org/wiki/Arduino> και <https://www.arduino.cc>)

2.2.3 Επεξήγηση Ειδικών Ακροδεκτών - Πρωτόκολλα Επικοινωνίας

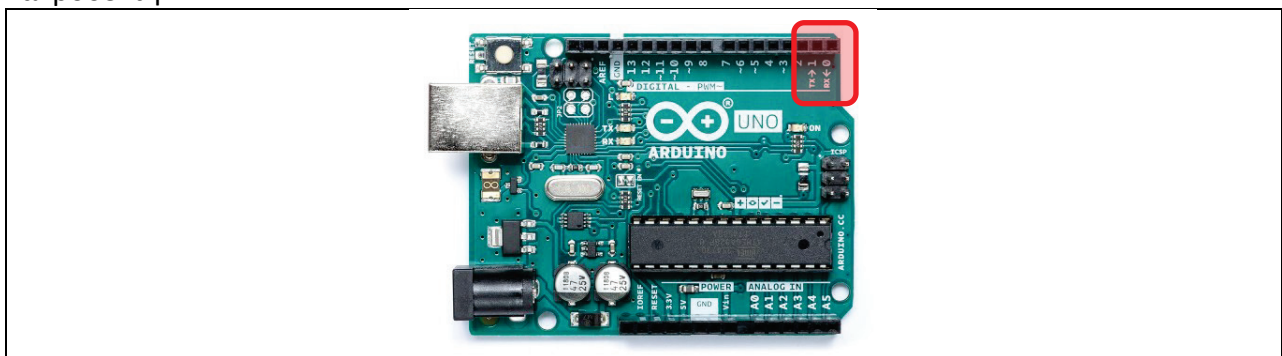
2.2.3.1 Γενικά

Στην πλακέτα Arduino Uno, μπορούν να χρησιμοποιηθούν τρία κύρια πρωτόκολλα επικοινωνίας μέσω των ακροδεκτών του κυκλώματος. Αυτά τα πρωτόκολλα είναι το UART, το I2C και το SPI. Με τη χρήση αυτών των πρωτοκόλλων, μπορείτε να επικοινωνήσετε με διάφορες συσκευές όπως αισθητήρες και ενεργοποιητές, και να δημιουργήσετε πιο πολύπλοκες εφαρμογές με το Arduino Uno. Παρακάτω αναλύεται το κάθε ένα ξεχωριστά.

2.2.3.2 Ακροδέκτες για UART-Serial (UART-Serial pins)

Το πιο χρησιμοποιούμενο πρωτόκολλο για το Arduino Uno είναι το UART. Οι δύο απαιτούμενες ακίδες για την UART επικοινωνία είναι οι ακίδες 0 και 1, γνωστές ως RX και TX αντίστοιχα. Το RX αντιπροσωπεύει τη λειτουργία λήψης (receive), ενώ το TX αντιπροσωπεύει τη λειτουργία μετάδοσης (transmit). Αυτή είναι μια αμφίδρομη επικοινωνία, που σημαίνει ότι μπορείτε να μεταδώσετε και να λάβετε δεδομένα μέσω αυτών των ακίδων.

Για να χρησιμοποιήσετε έναν αναλογικό ακροδέκτη ως ψηφιακό ακροδέκτη, απλά πρέπει να ορίσετε τη λειτουργία του ακροδέκτη, όπως θα κάνατε και για τους ψηφιακούς ακροδέκτες, μέσα στην συνάρτηση `setup()` του προγράμματος Arduino. Στη συνέχεια, μπορείτε να χρησιμοποιήσετε τις συναρτήσεις `digitalWrite()` και `digitalRead()` για να ελέγξετε την κατάσταση του ακροδέκτη.



Εικόνα 2-15 Ακροδέκτες για UART-Serial



Εικόνα 2-16 Ακροδέκτες για UART-Serial

(Arduino Docs | Arduino Documentation, n.d.)

Το Serial που χρησιμοποιείται μέσω της θύρας USB είναι το ίδιο με αυτό που χρησιμοποιείται με τους ακροδέκτες 0 και 1 της πλακέτας Arduino Uno. Επομένως, εάν επιθυμείτε να συνδέσετε μια άλλη συσκευή μέσω των ακροδεκτών RX/TX της πλακέτας σας, θυμηθείτε να μην χρησιμοποιείτε το Serial μέσω USB (Serial through USB). Σε ορισμένες άλλες πλακέτες Arduino, όπως η Arduino Mega, υπάρχουν πολλά διαφορετικά UART διαθέσιμα. Ωστόσο, για το Arduino Uno, υπάρχει μόνο ένα UART διαθέσιμο.

Αν θέλετε να χρησιμοποιήσετε περισσότερες διεπαφές UART, μπορείτε να το κάνετε χρησιμοποιώντας τη βιβλιοθήκη SoftwareSerial. Αυτή η βιβλιοθήκη επιτρέπει τη χρήση οποιασδήποτε άλλης ψηφιακής ακίδας για την επικοινωνία μέσω UART. Ωστόσο, υπάρχει μια σημαντική διαφορά εδώ: το "αληθές" (true) Serial χρησιμοποιεί την λειτουργία υλικού επικοινωνίας της πλακέτας Arduino Uno, η οποία είναι γρήγορη και δεν απαιτεί μεγάλη υπολογιστική ισχύ. Η SoftwareSerial, από την άλλη πλευρά, λειτουργεί με χρήση υπολογιστικής ισχύος αντί για υλικού επιπέδου επικοινωνία. Επομένως, είναι προτιμότερο να χρησιμοποιείτε το "true" Serial όταν είναι δυνατόν και να επιλέγετε την SoftwareSerial μόνο όταν χρειάζεστε πρόσθετες διεπαφές UART. Φυσικά, υπάρχουν και άλλες εκδόσεις των πλακετών Arduino με περισσότερες υλικές διεπαφές UART, όπως η έκδοση Arduino Mega.

Για να συνδέσετε ένα εξάρτημα στους ακροδέκτες Arduino Uno και να χρησιμοποιήσετε τη σειριακή επικοινωνία, θα χρειαστείτε 4 καλώδια:

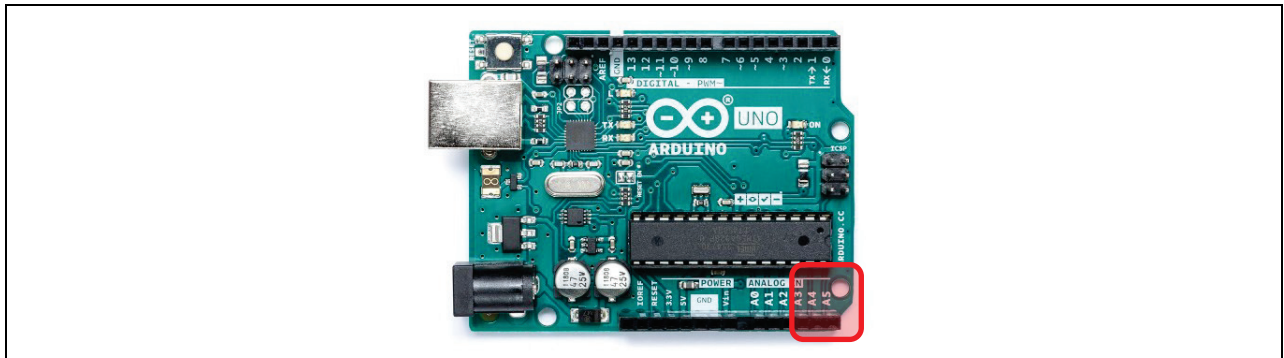
- Ένα μεταξύ του RX του εξαρτήματος και του TX του Arduino Uno.
- Ένα μεταξύ του TX του εξαρτήματος και του RX του Arduino Uno.
- Εάν το εξάρτημα δεν τροφοδοτείται εξωτερικά, ένα καλώδιο για να το τροφοδοτήσει από τους ακροδέκτες τροφοδοσίας του Arduino Uno.
- Ένα για τη σύνδεση της γείωσης.

(πηγή: <https://el.wikipedia.org/wiki/Arduino> και <https://www.arduino.cc>)

2.2.3.3 Ακροδέκτες για I2C (I2C pins)

Το πρωτόκολλο I2C είναι ένας διαύλος επικοινωνίας με πολλαπλούς κύριους και πολλαπλούς σκλάβους (multi-master/multi-slave). Ωστόσο, οι περισσότερες εφαρμογές απαιτούν μία αρχιτεκτονική με έναν κύριο και πολλαπλούς σκλάβους (one-master/multi-slave), και θα επικεντρωθούμε κυρίως σε αυτήν την αρχιτεκτονική.

Μπορείτε να φανταστείτε τον δίαυλο δεδομένων όπου όλα τα δεδομένα κυκλοφορούν. Ο κύριος (master) είναι ο επικεφαλής του διαύλου. Τώρα, μπορείτε να προσθέσετε οποιοδήποτε νέο στοιχείο στο δίαυλο, το οποίο θα είναι συνδεδεμένο ως σκλάβο (slave). Κάθε στοιχείο έχει επίσης ένα μοναδικό αναγνωριστικό (ID). Ο master θα στέλνει δεδομένα και αιτήματα σε έναν slave, επικοινωνώντας μέσω του διαύλου και παρέχοντας το αναγνωριστικό του slave. Εάν ο master χρειάζεται μία απάντηση, ο slave θα την στείλει πίσω στο δίαυλο. Αφού ο master λάβει μία απάντηση, μπορεί να στείλει την επόμενη εντολή ή αίτημα. Συνήθως, η πλακέτα Arduino Uno χρησιμοποιείται ως κύριος και συνδέετε ένα ή περισσότερα εξάρτηματα (συνήθως αισθητήρες) στο δίαυλο I2C, με κάθε ένα να έχει μοναδικό αναγνωριστικό. Στο λογισμικό, μπορείτε να χρησιμοποιήσετε τη βιβλιοθήκη Arduino Wire, η οποία είναι ανοιχτού κώδικα.



Εικόνα 2-17 Ακροδέκτες για I2C



Εικόνα 2-18 Ακροδέκτες για I2C

(Arduino Docs | Arduino Documentation, n.d.)

Απαιτούνται 4 καλώδια για να χρησιμοποιήσετε ένα δίαυλο I2C:

- Ένα για να συνδέσετε τον ακροδέκτη SCL (clock).
- Ένα άλλο για τον ακροδέκτη SDA (data).
- Ένα για να ενεργοποιήσετε τα εξαρτήματα του διαύλου.
- Και ένα για τη σύνδεση της γείωσης.

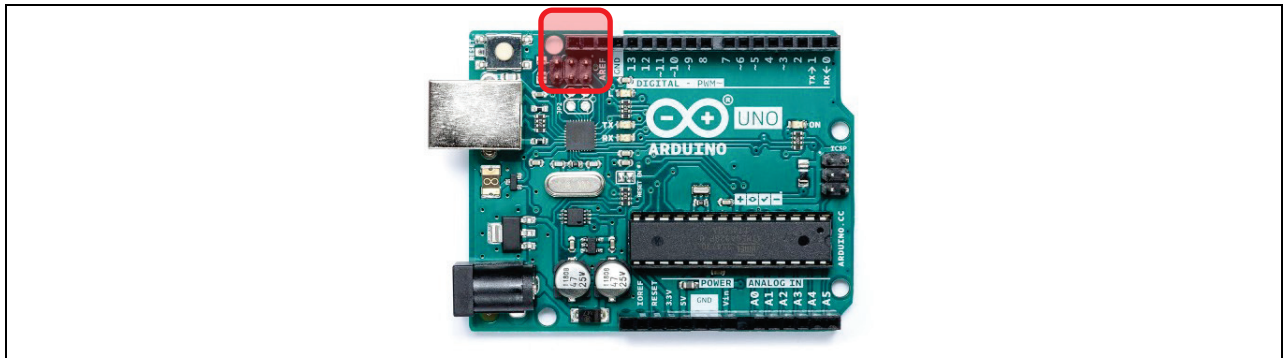
(πηγή: <https://el.wikipedia.org/wiki/Arduino> και <https://www.arduino.cc>)

2.2.3.4 Ακροδέκτες για SPI (SPI pins)

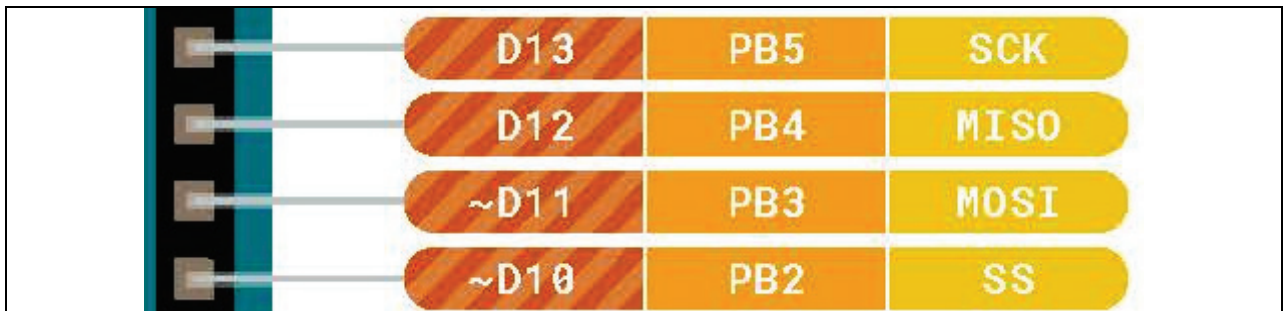
Το πρωτόκολλο SPI (Serial Peripheral Interface) είναι ένα άλλο πρωτόκολλο που βασίζεται στην αρχιτεκτονική master-slave. Μπορεί να χρησιμοποιηθεί για να συνδέσετε την πλακέτα Arduino με πολλές συσκευές. Η ταχύτητα επικοινωνίας του SPI είναι γενικά πιο υψηλή από αυτήν των πρωτοκόλλων I2C και UART, αλλά δεν είναι κατάλληλη για επικοινωνία σε μεγάλες αποστάσεις (περισσότερο από μερικά εκατοστά για το καλώδιο).

Όσον αφορά την υλοποίηση του SPI στο Arduino, μπορείτε να χρησιμοποιήσετε απευθείας την ανοικτού κώδικα βιβλιοθήκη SPI στα προγράμματα σας. Αυτή η βιβλιοθήκη παρέχει συναρτήσεις και μεθόδους που επιτρέπουν την επικοινωνία με συσκευές που χρησιμοποιούν το πρωτόκολλο SPI. Μπορείτε να τη χρησιμοποιήσετε για να αναπτύξετε προγράμματα που επικοινωνούν με αισθητήρες, ενσωματωμένα συστήματα και άλλες συσκευές που υποστηρίζουν το πρωτόκολλο SPI.

Σημειώστε ότι η ταχύτητα επικοινωνίας είναι ταχύτερη από ό,τι για το I2C και το UART, αλλά δεν είναι κατάλληλη για επικοινωνία σε μεσαίες και μεγάλες αποστάσεις (περισσότερο από μερικά εκατοστά για το καλώδιο).



Εικόνα 2-19 Ακροδέκτες για SPI



Εικόνα 2-20 Ακροδέκτες για SPI

(Arduino Docs | Arduino Documentation, n.d.)

Απαιτούνται το λιγότερο 6 καλώδια για να συνδεθεί ένα SPI εξάρτημα:

- Ακροδέκτης SCK (clock).
- Ακροδέκτης MISO (Master In, Slave Out).
- Ακροδέκτης MOSI (Master Out, Slave In).
- Ένα για κάθε CS/SS (Chip Select/Slave Select). Για κάθε επιπλέον slave, θα πρέπει να προσθέσετε μια ακόμη σύνδεση σε ένα ακροδέκτη Arduino.
- Ένα για την τροφοδοσία των εξαρτημάτων.
- Και ένα για την κοινή σύνδεση της γείωσης.

(πηγή: <https://el.wikipedia.org/wiki/Arduino> και <https://www.arduino.cc>)

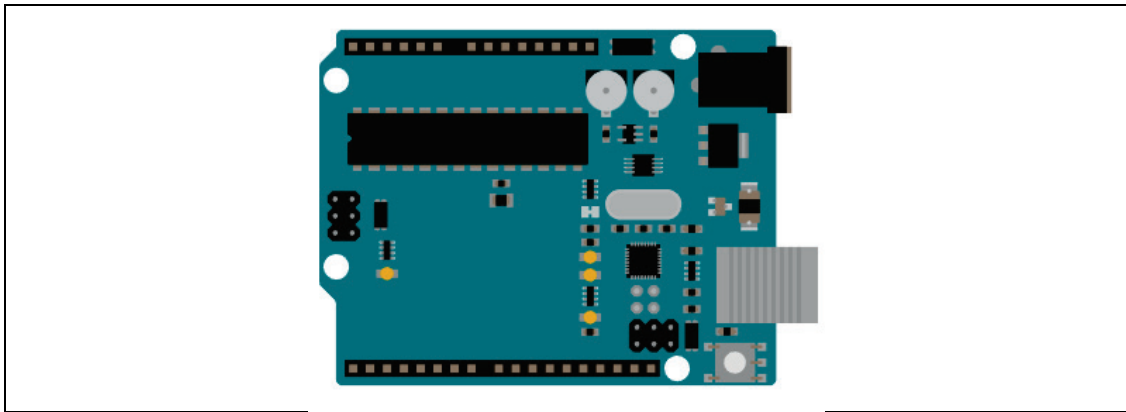
2.2.4 Εκδόσεις Arduino

Η πλατφόρμα Arduino χωρίζεται σε διάφορες οικογένειες, όπου η κάθε μία περιέχει πολλές εκδόσεις.

Οικογένεια Classic

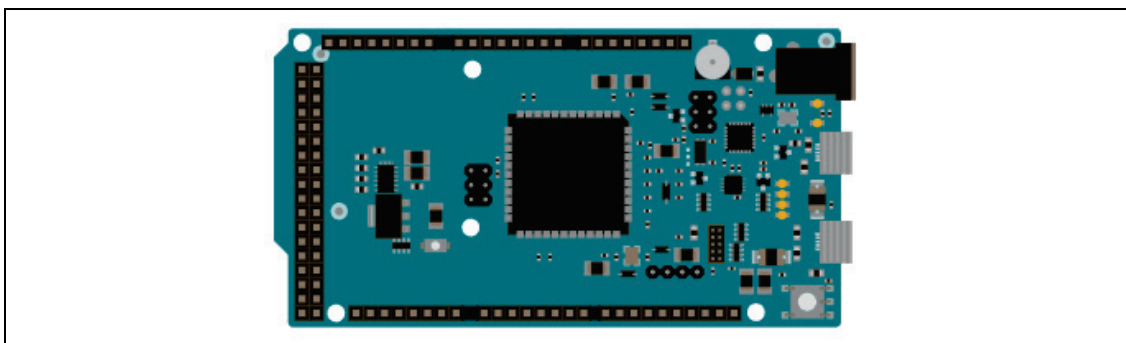
Η οικογένεια Classic (Εικόνες 2-21 έως 2-24) περιλαμβάνει τις αρχικές και πιο χρησιμοποιούμενες υλοποιήσεις.

Uno R3: Το Arduino Uno θεωρείται μία από τις καλύτερες πλακέτες για την ένταξη στον χώρο των ηλεκτρονικών και του προγραμματισμού. Η πλακέτα βασίζεται στον ATmega328P και παρέχει 14 ψηφιακές εισόδους εξόδους, από τις οποίες οι 6 μπορούν να χρησιμοποιηθούν και ως παλμικοί έξοδοι. Επίσης παρέχει 6 αναλογικές εισόδους, USB σύνδεση, βύσμα τροφοδοσίας, ICSP header και κουμπί reset.



Εικόνα 2-21 Arduino Uno R3
(Arduino Docs | Arduino Documentation, n.d.)

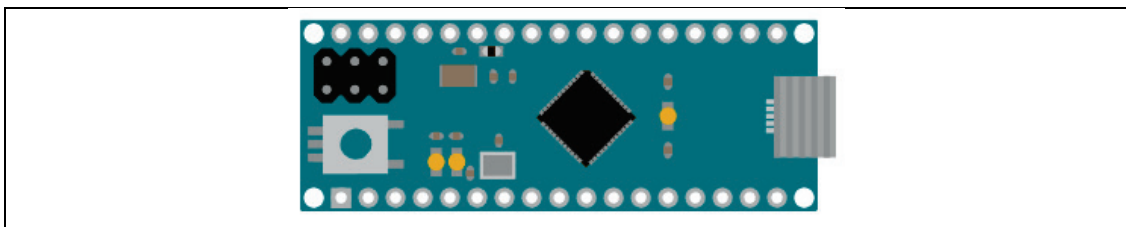
Due: Η πλακέτα βασίζεται στο μικροελεκτή 32-bit Arm. Πιο συγκεκριμένα, χρησιμοποιεί τον Atmel SAM3X8E ARM Cortex-M3, 54 ψηφιακές εισόδους/εξόδους, από τις οποίες οι 12 χρησιμοποιούνται ως παλμικοί (PWM) έξοδοι. Παρέχει 12 αναλογικές εισόδους, 4 UART (σειριακές θύρες), 84 MHz ρολόι (clock), μία USB OTG σύνδεση, 2 DAC (digital to analog), 2 TWI (Two Wire Interface), βύσμα τροφοδοσίας, SPI header, JTAG header και κουμπί reset.



Εικόνα 2-22 Arduino Due

(Arduino Docs | Arduino Documentation, n.d.)

Micro: Η Πλακέτα Micro έχει ως βάση τον ATmega32U4 και αναπτύχθηκε από κοινού με την Adafruit. Παρέχει 20 ψηφιακές εισόδους/εξόδους από τα οποίες 7 χρησιμοποιούνται ως παλμικές έξοδοι και 12 ως αναλογικοί είσοδοι. Περιέχει κρυσταλλικό ταλαντωτή 16MHz, micro USB, ICSP header και κουμπί reset.

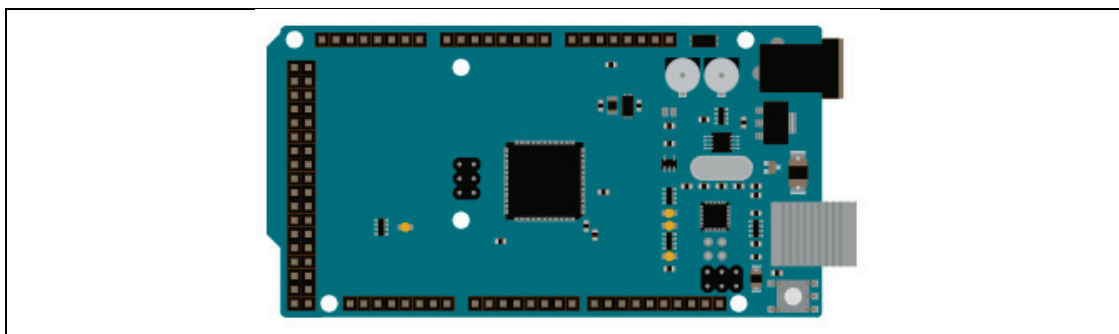


Εικόνα 2-23 Arduino Micro

(Arduino Docs | Arduino Documentation, n.d.)

Mega 2560 Rev3: Η πλακέτα Mega 2560 Rev3 έχει ως βάση τον ATmega2560. Παρέχει 54 ψηφιακές εισόδους/εξόδους από τις οποίες οι 15 μπορούν να χρησιμοποιηθούν και ως παλμικές

έξοδοι. Ακόμα παρέχει 16 αναλογικές εισόδους, 4 UARTs, 16MHz κρυσταλλικό ταλαντωτή, σύνδεση USB, βύσμα τροφοδοσίας, ICSP header και κουμπί reset.

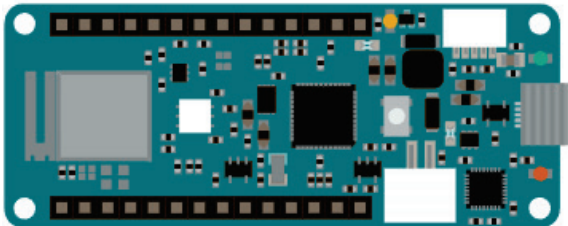
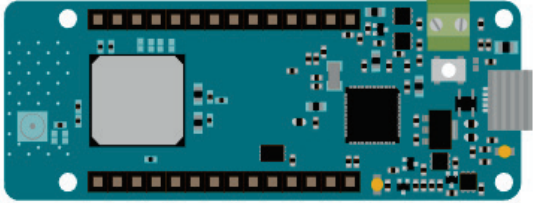
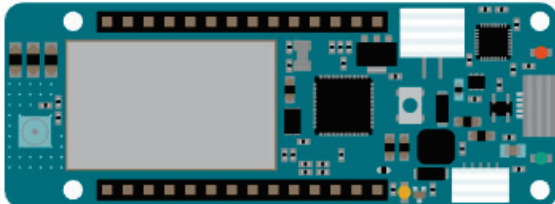
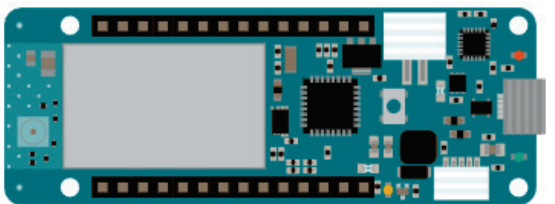


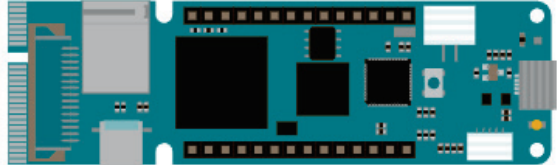
Εικόνα 2-24 Arduino Mega 2560 Rev3

(Arduino Docs | Arduino Documentation, n.d.)

Οικογένεια MKR

Η οικογένεια MKR περιλαμβάνει χαμηλού κόστους υλοποιήσεις που παρέχουν ενσωματωμένες ασύρματες επικοινωνίες, όπως π.χ. το Wi-Fi – GSM, ενώ χρησιμοποιεί τον ARM Cortex M0 32bit SAMD21 και το ολοκληρωμένο ECC508 crypto-για ασφάλεια (πίνακας 2-1).

Τύπος	Χαρακτηριστικά	Πλακέτα
MKR Wi-Fi 1010	Παρέχει Wi-Fi και Bluetooth συνδεσιμότητα χρησιμοποιώντας το Nina W102 uBlox module.	
MKR WAN 1310	Παρέχει LoRa συνδεσιμότητα χρησιμοποιώντας το Murata CMWX1ZZABZ module.	
MKR GSM 1400	Υποστηρίζει GSM/3G δίκτυο χρησιμοποιώντας u-blox SARA-U201.	
MKR NB 1500	Παρέχει συνδεσιμότητα τύπου Narrowband χρησιμοποιώντας το uBlox SARA-R410M-02B module.	

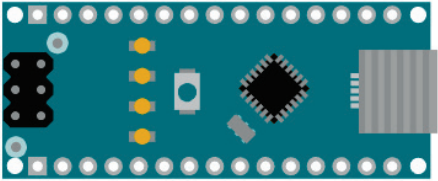
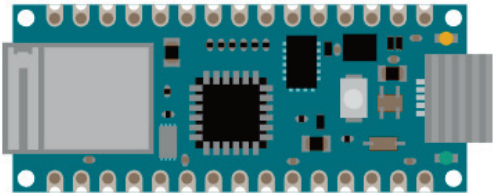
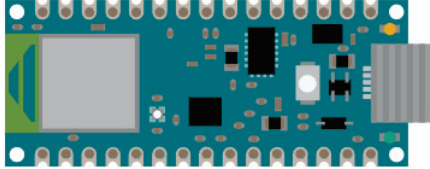
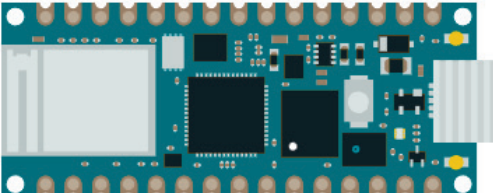
MKR Vidor 4000	Έχει ενσωματωμένο FPGA chip (Intel Cyclone 10CL016), κάμερα, HDMI σύνδεση, Wi-Fi και Bluetooth (NINA-W102).	
----------------	---	--

Πίνακας 2-1 Πλακέτες Arduino της οικογένειας MKR

(Arduino Docs | Arduino Documentation, n.d.)

Οικογένεια Nano

Η οικογένεια Nano προσφέρει υλοποιήσεις μικρών διαστάσεων και χαμηλού κόστους χωρίς να υστερούν σε απόδοση. Παρέχουν σύνδεση Mini-B USB (πίνακας 2-2).

Τύπος	Χαρακτηριστικά	Πλακέτα
Nano	Η πρώτη πλακέτα της σειράς. Έχει το μικρότερο μέγεθος και βάρος.	
Nano 33 IoT	Χρησιμοποιεί τον M0 32-bit SAMD21 επεξεργαστή. Υποστηρίζει σύνδεση Wi-Fi, Bluetooth μέσω του u-blox NINA-W102 Wi-Fi module. Περιέχει γυροσκόπιο και επιταχυνσιόμετρο μέσω του LSM6DS3.	
Nano 33 BLE	Ίδιο pinout με το Nano, χρησιμοποιεί τον μικροελεκτή nRF52840. Υποστηρίζει σύνδεση Bluetooth. Είναι συμβατή με MicroPython και ArmMbed OS.	
Nano RP2040 Connect	Χρησιμοποιεί τον μικροελεκτή Raspberry Pi RP2040. Υποστηρίζει Bluetooth, Wi-Fi, μικρόφωνο και αισθητήρα θερμοκρασίας.	

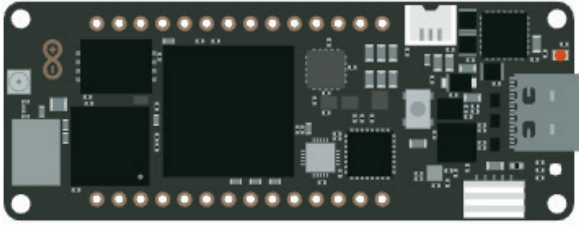
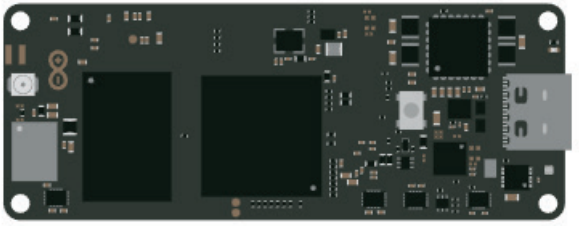
Πίνακας 2-2 Πλακέτες Arduino της οικογένειας Nano

(Arduino Docs | Arduino Documentation, n.d.)

Οικογένεια Portenta

Η οικογένεια Portenta περιλαμβάνει υλοποιήσεις προορισμένες για βιομηχανική χρήση. Έχει βασιστεί στην οικογένεια MKR, έχει όμως περισσότερες δυνατότητες τόσο σε υλικό όσο και σε λογισμικό. Η οικογένεια χωρίζεται κυρίως σε 2 κατηγορίες, τις πλακέτες που βασίζονται στην Portenta H7 και στην Portenta X8. Οι H7 χρησιμοποιούν τον επεξεργαστή STM32H747XI dual Cortex-M7+M4 32bit, τον μικροελεκτή Portenta H7 και για την ασύρματη σύνδεση Murata 1DX

dual Wi-Fi and Bluetooth 5.1 (πίνακας 2-3). (πηγή: <https://el.wikipedia.org/wiki/Arduino> και <https://www.arduino.cc>)

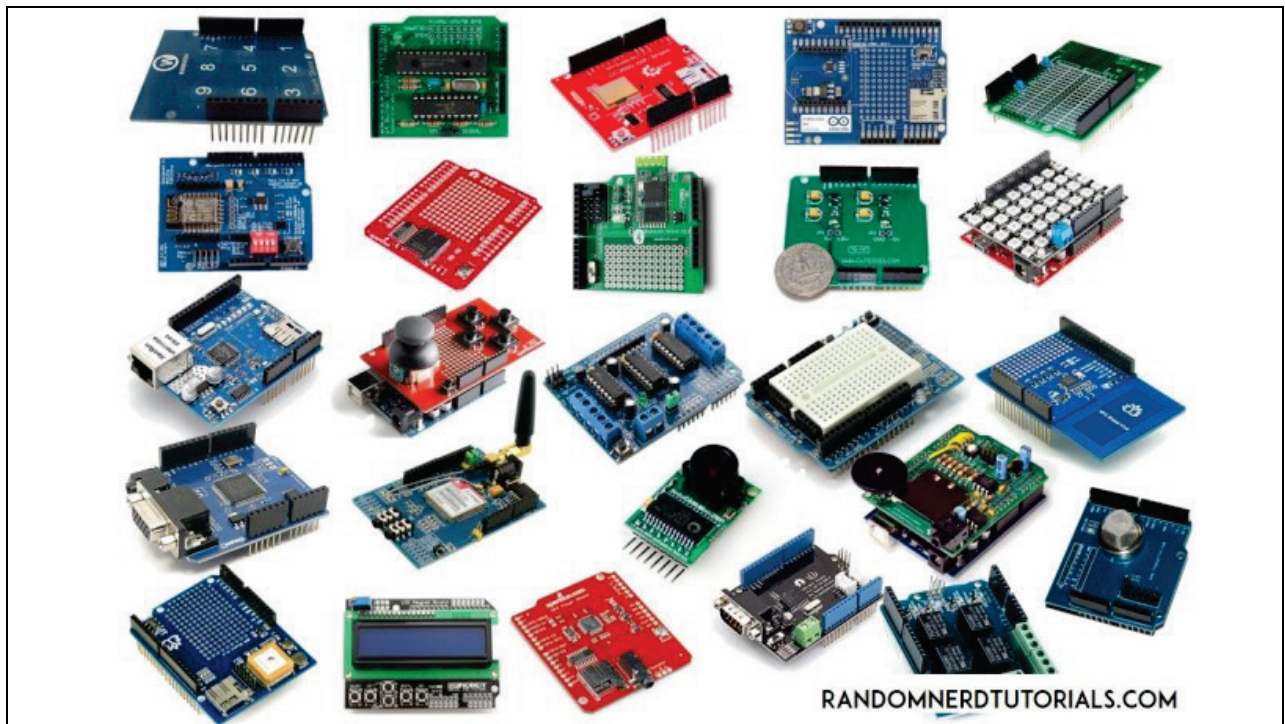
Τύπος	Χαρακτηριστικά	Πλακέτα
Portenta H7	Περιέχει το σύνολο της κατηγορίας H7 και επιπλέον τον επεξεργαστή γραφικών Chrom-ART.	
Portenta X8	Περιέχει το σύνολο της κατηγορίας H7 και επιπλέον τον επεξεργαστή πολυμέσων NXP i.MX 8M Mini.	

Πίνακας 2-3 Πλακέτες Arduino της οικογένειας Portenta

(Arduino Docs | Arduino Documentation, n.d.)

2.3 ARDUINO SHIELDS

Τα Arduino shields είναι προ-κατασκευασμένες πλακέτες κυκλωμάτων, τα οποία συνδέονται πάνω σε συμβατές πλακέτες arduino. Παρέχουν πρόσθετες δυνατότητες όπως σύνδεση στο διαδίκτυο, έλεγχος κινητήρα, παροχή ασύρματης Wifi / ZigBee επικοινωνίας, GPS, Ethernet, έλεγχος οθονών και άλλα.



Εικόνα 2-25 Arduino Shields

(25 Useful Arduino Shields That You Might Need to Get | Randomnerdtutorials, n.d.)

Ακολουθούν μερικά ενδεικτικά Arduino Shields.

Ethernet Shield

Το Ethernet Shield επιτρέπει να συνδεθεί το Arduino στο διαδίκτυο. Απλά πρέπει να συνδέσετε το Shield στην πλακέτα του Arduino και στη συνέχεια να συνδεθεί στο δίκτυο. (πηγή:

<https://randomnerdtutorials.com/25-arduino-shields/>)

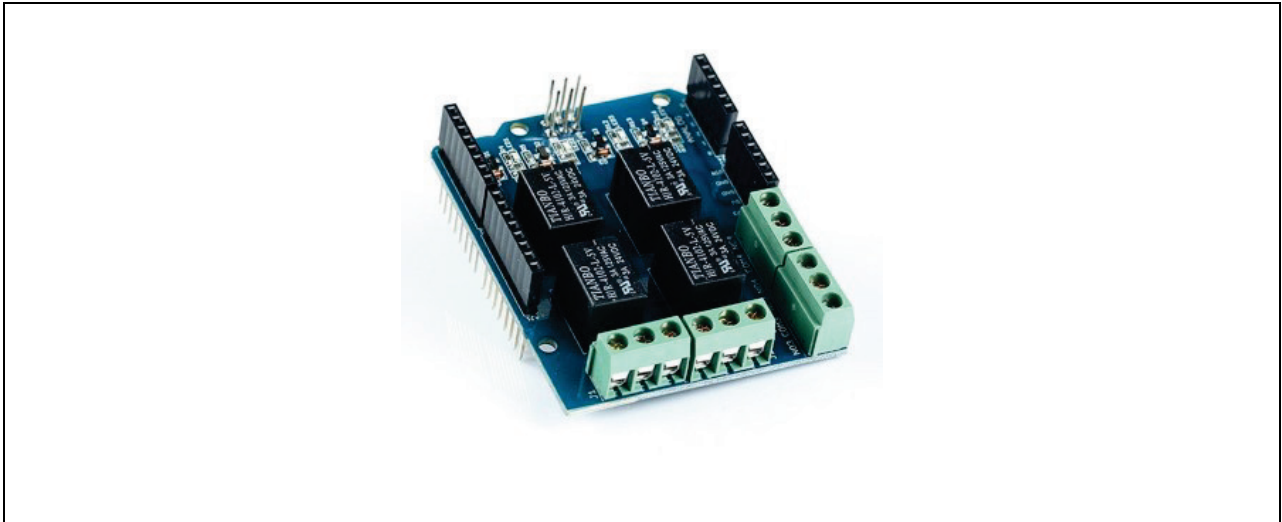


Εικόνα 2-26 Ethernet Shield

(25 Useful Arduino Shields That You Might Need to Get | Randomnerdtutorials, n.d.)

Relay Shield

Το Relay Shield είναι μια μονάδα με 4 μηχανικά ρελέ που παρέχει έναν εύκολο τρόπο ελέγχου υψηλής τάσης. (πηγή: <https://randomnerdtutorials.com/25-arduino-shields/>)



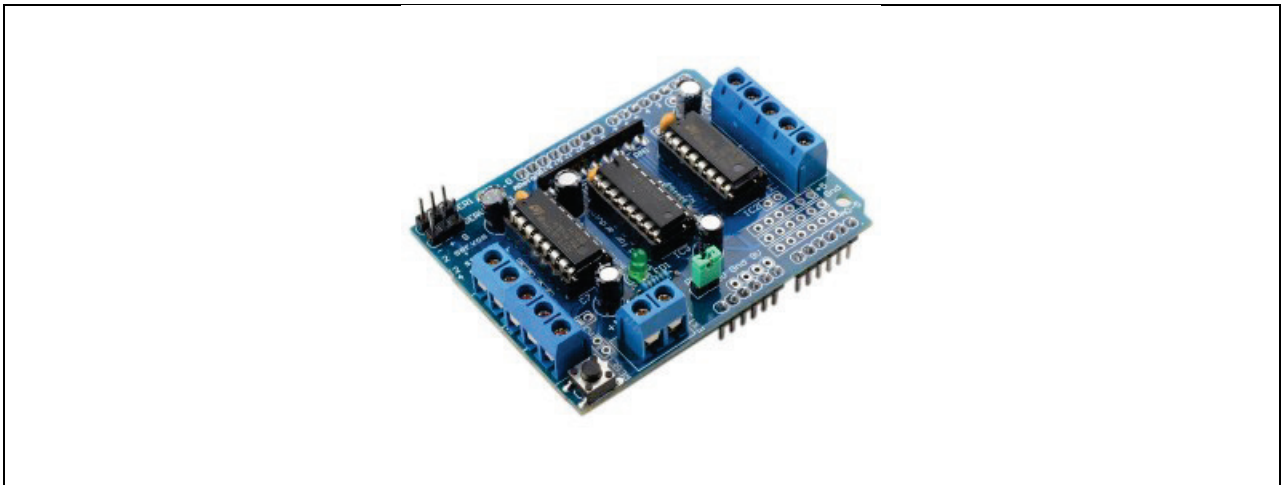
Εικόνα 2-27 Relay Shield

(25 Useful Arduino Shields That You Might Need to Get | Randomnerdtutorials, n.d.)

Motor Shield

Το Motor Shield επιτρέπει τον εύκολο έλεγχο της κατεύθυνσης και της ταχύτητας του κινητήρα. Διευκολύνει την ενσωμάτωση ενός κινητήρα σε οποιοδήποτε έργο. (πηγή:

<https://randomnerdtutorials.com/25-arduino-shields/>)



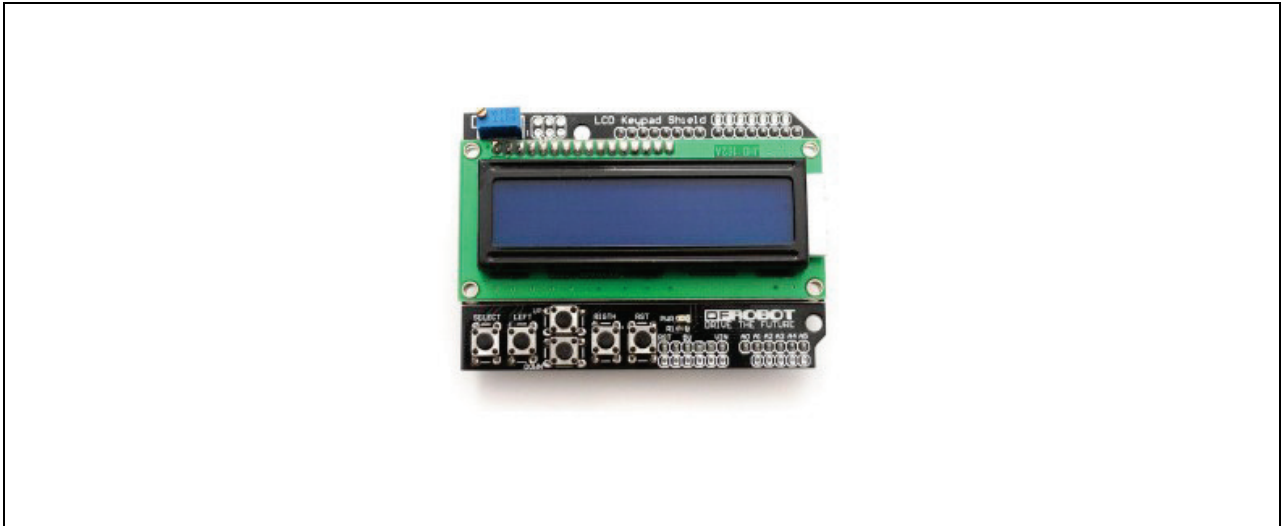
Εικόνα 2-28 Motor Shield

(25 Useful Arduino Shields That You Might Need to Get | Randomnerdtutorials, n.d.)

LCD Shield

Το LCD Shield διευκολύνει τη χρήση μιας οθόνης LCD 16×2 χαρακτήρων. Με αυτό, είναι δυνατό να ελεγχθεί μια LCD 16×2 χαρακτήρων, έως και 3 ακίδες οπίσθιου φωτισμού και 5 ακίδες πληκτρολογίου χρησιμοποιώντας μόνο τις δύο ακίδες I2C του Arduino. (πηγή:

<https://randomnerdtutorials.com/25-arduino-shields/>)



Εικόνα 2-29 LCD Shield

(25 Useful Arduino Shields That You Might Need to Get | Randomnerdtutorials, n.d.)

GSM/GPRS Shield

Το GSM/GPRS Shield επιτρέπει την σύνδεση του Arduino στο δίκτυο κινητής τηλεφωνίας GSM/GPRS. Επιτρέπει την κλήση ενός αριθμού τηλεφώνου ή την αποστολή μηνύματος κειμένου μέσω εύκολων στη χρήση εντολών AT. (πηγή: <https://randomnerdtutorials.com/25-arduino-shields/>)



Εικόνα 2-30 GSM/GPRS Shield

(25 Useful Arduino Shields That You Might Need to Get | Randomnerdtutorials, n.d.)

2.4 ΛΟΓΙΣΜΙΚΟ (SOFTWARE) – ARDUINO IDE

2.4.1 Γενικά

Για να προγραμματιστεί το Arduino χρειάζεται το περιβάλλον προγραμματισμού Arduino IDE. Στο περιβάλλον αυτό γράφεται ο κώδικας (ο οποίος βασίζεται στη γλώσσα C/C++), ο οποίος μετά μεταγλωττίζεται και μεταφορτώνεται στη πλακέτα.

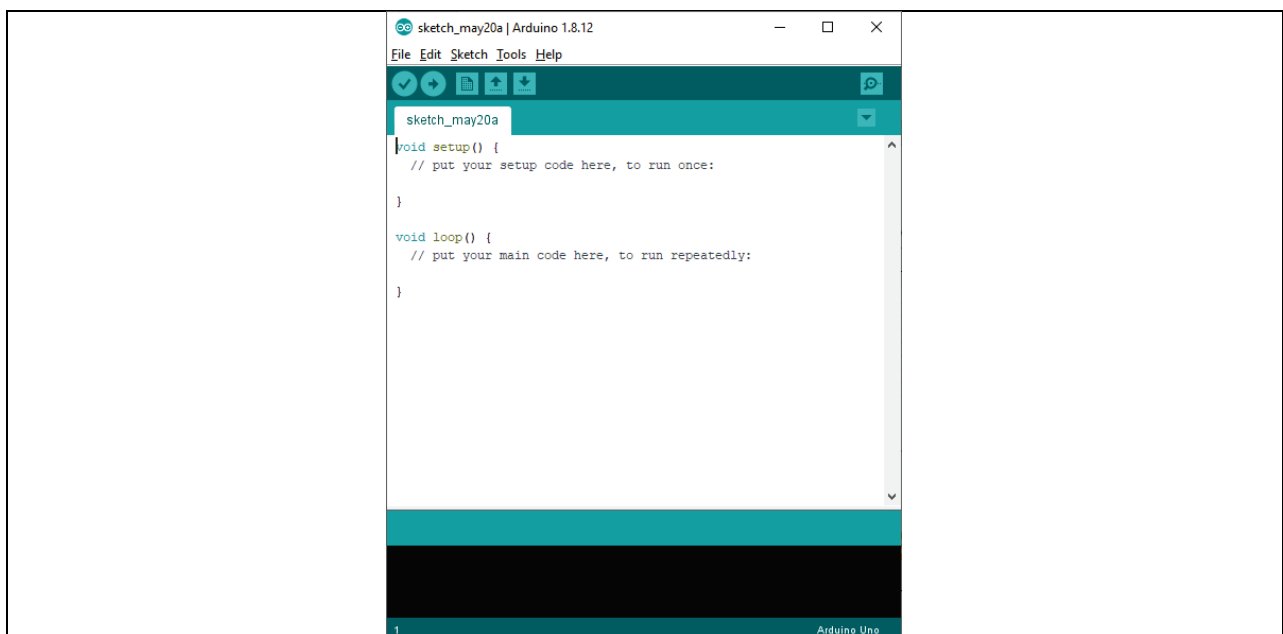
2.4.2 Λογισμικό ARDUINO IDE

Το περιβάλλον ανάπτυξης του Arduino (IDE) είναι μία εφαρμογή διαθέσιμη σε διάφορες πλατφόρμες, όπως Windows, MacOS, Linux, που είναι γραμμένη στη γλώσσα προγραμματισμού Java. Χρησιμοποιείται για την εγγραφή και τη μεταφόρτωση προγραμμάτων συμβατών με το Arduino.

Ένα πρόγραμμα ή κώδικας που γράφεται για το Arduino ονομάζεται "sketch". Ο πηγαίος κώδικας για το Arduino IDE προέρχεται από τη GNU General Public License v.2. Το Arduino IDE υποστηρίζει τις γλώσσες C και C++, και ακολουθεί συγκεκριμένους κανόνες για τη δομή του κώδικα. Το IDE παρέχει μια σειρά από βιβλιοθήκες λογισμικού, οι οποίες προσφέρουν πολλές κοινές διαδικασίες εισόδου και εξόδου.

Μαζί με το Arduino IDE, διατίθεται και η εργαλειοθήκη GNU. Το IDE χρησιμοποιεί το πρόγραμμα avrdude για να μετατρέψει τον εκτελέσιμο κώδικα σε δεκαεξαδικό αρχείο κειμένου, το οποίο στη συνέχεια φορτώνεται στον πίνακα Arduino μέσω του προγράμματος φορτωτή στο firmware του πίνακα.

Επιπλέον, δεδομένου ότι η πλατφόρμα Arduino βασίζεται σε Atmel μικροελεγκτές, το ολοκληρωμένο περιβάλλον ανάπτυξης της Atmel, το AVR Studio ή το νεότερο Atmel Studio, μπορούν επίσης να χρησιμοποιηθούν για την ανάπτυξη λογισμικού για το Arduino. Περιλαμβάνει ένα πρόγραμμα επεξεργασίας κώδικα με χαρακτηριστικά όπως είναι η επισήμανση σύνταξης και ο συνδυασμός αγκύλων και είναι επίσης σε θέση να μεταγλωττίζει και να φορτώνει προγράμματα στην πλακέτα με ένα μόνο κλικ. (πηγή: <https://el.wikipedia.org/wiki/Arduino> και <https://www.arduino.cc>)

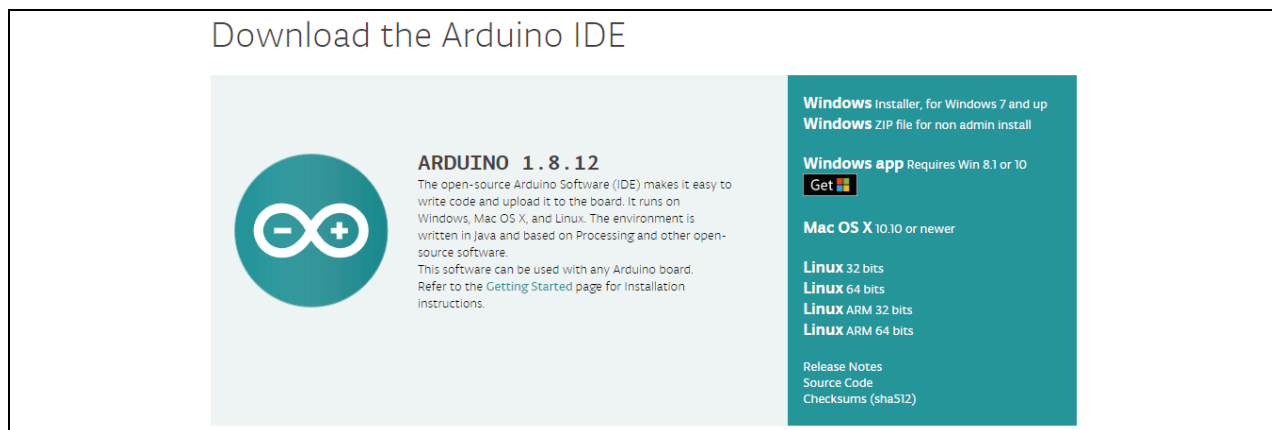


Εικόνα 2-31 Περιβάλλον Arduino IDE

2.4.3 Εγκατάσταση Λογισμικού ARDUINO IDE

Μπορείτε να κατεβάσετε το λογισμικό Arduino IDE από την παρακάτω διεύθυνση:

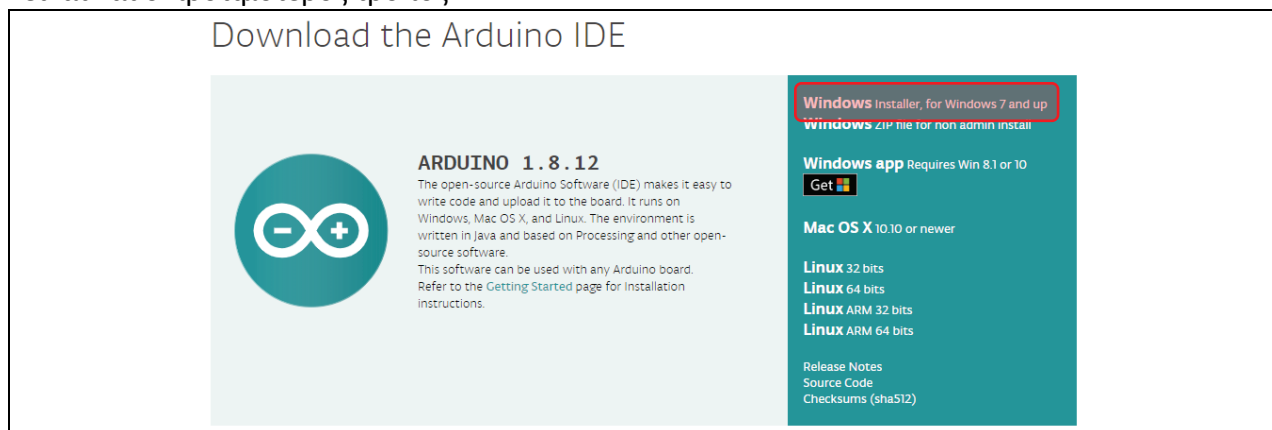
<https://www.arduino.cc/en/Main/Software>.



Εικόνα 2-32 Διαθέσιμες εκδόσεις Arduino IDE

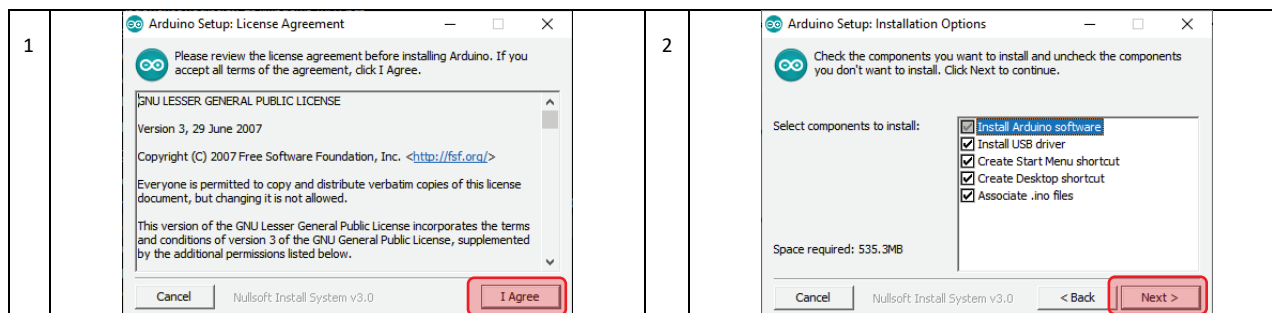
Διαδικασία εγκατάστασης σε περιβάλλον windows.

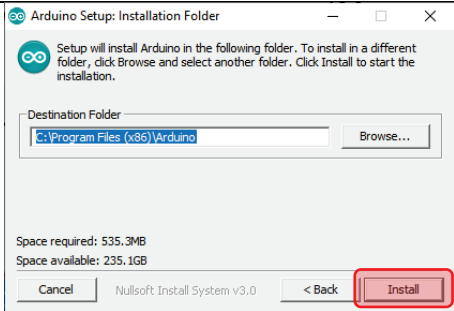
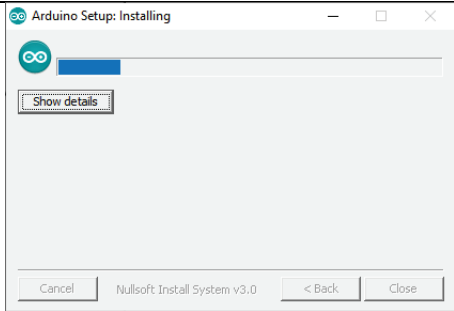
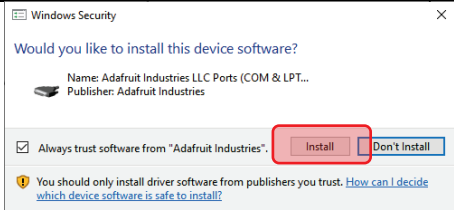
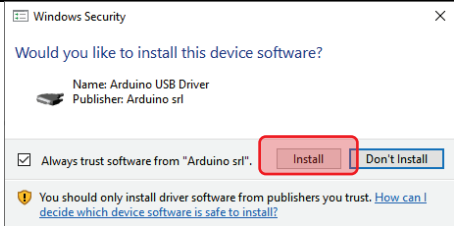
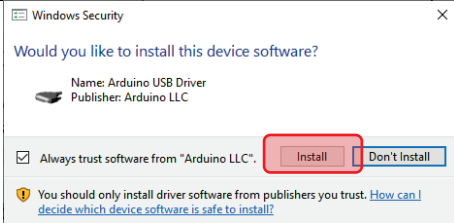
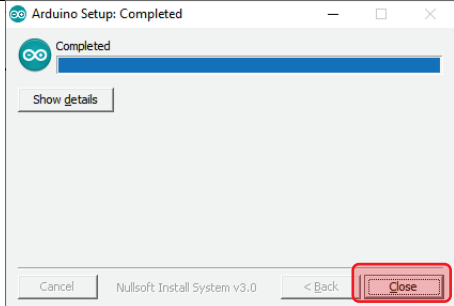
Επιλέξτε την πρώτη έκδοση για windows, που αναφέρει windows installer, ο οποίος κατεβάζει στον υπολογιστή μας ένα εκτελέσιμο αρχείο για αυτοματοποιημένη εγκατάστασή, ο οποίος είναι και ο προτιμότερος τρόπος.



Εικόνα 2-33 Επιλογή έκδοσης Arduino IDE για windows

Τρέχοντας το εκτελέσιμο (.exe) αρχείο σε περιβάλλον Windows, που κατεβάσατε από την παραπάνω διεύθυνση, θα δείτε τα παρακάτω στάδια εγκατάστασης.

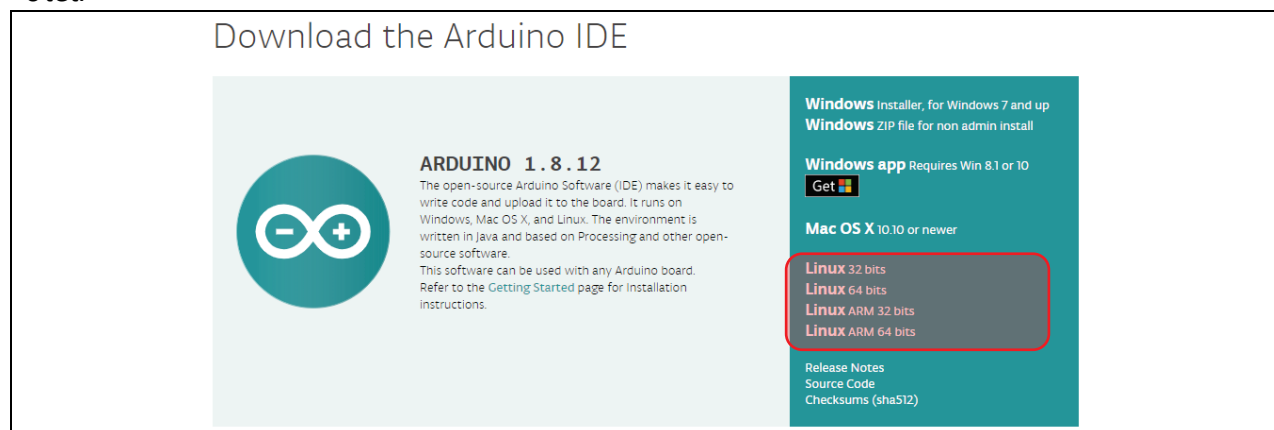


	Άδεια Χρήσης επιλογή I Agree		Λίστα επιλογής των στοιχείων που θα εγκατασταθούν επιλογή Next
3	 Φάκελος εγκατάστασης του προγράμματος επιλογή Install	4	 Γίνεται η εγκατάσταση
5	 Εγκατάσταση οδηγών (drivers) επιλογή Install	6	 Εγκατάσταση οδηγών (drivers) επιλογή Install
7	 Εγκατάσταση οδηγών (drivers) επιλογή Install	8	

Εικόνα 2-34 Στάδια εγκατάστασης Arduino IDE σε περιβάλλον Windows

Διαδικασία εγκατάστασης σε περιβάλλον Linux.

Κατεβάζουμε το συμπιεσμένο αρχείο εγκατάστασης και το αποσυμπιέζουμε σε έναν φάκελο της επιλογής μας. Επιλέγοντας το αρχείο με όνομα arduino, το πρόγραμμα μπορεί να εκτελεστεί.



Εικόνα 2-35 Επιλογή έκδοσης Arduino IDE για Linux

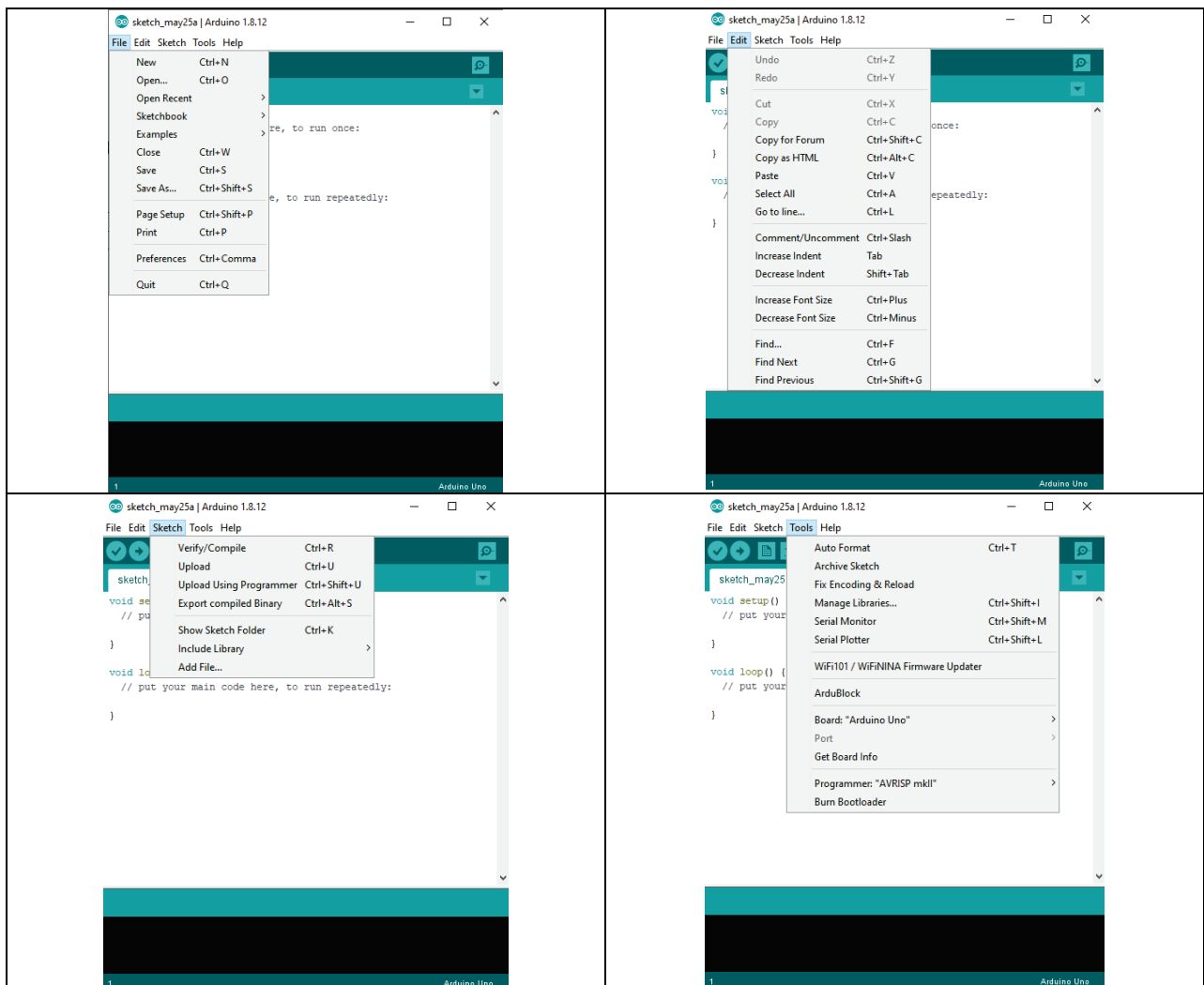
Μπορούμε να ανοίξουμε ένα τερματικό, και πηγαίνοντας στο φάκελο του προγράμματος, δίνουμε την εντολή `./install.sh`. Έτσι δημιουργείται μία συντόμευση στο μενού των εφαρμογών, μία συντόμευση στην επιφάνεια εργασίας και συσχετίζει τα αρχεία τύπου `.ino` με την εφαρμογή.

2.4.4 Περιβάλλον ARDUINO IDE

Στις παρακάτω εικόνες μπορούμε να δούμε τις επιλογές που έχει ο χρήστης, πατώντας στο μενού του περιβάλλοντος του Arduino IDE.

Στην κατηγορία File, οι βασικές επιλογές είναι New (νέο αρχείο), Open (άνοιγμα αρχείου), Save (αποθήκευση αρχείου), Print (εκτύπωση), Preferences (ιδιότητες).

Στην κατηγορία Edit, οι βασικές επιλογές είναι New (νέο αρχείο), Open (άνοιγμα αρχείου), Save (αποθήκευση αρχείου), Print (εκτύπωση), Preferences (ιδιότητες).



Εικόνα 2-36 Επιλογή έκδοσης Arduino IDE για Linux

2.4.5 Λογισμικό ARDUINO IDE – Πρώτος Κώδικας

Ο προγραμματισμός του του Arduino είναι πολύ απλός . Υπάρχουν δύο βασικές συναρτήσεις, η `setup ()` και η `loop ()`, οι οποίες δουλεύουν ως εξής:

- `setup ()` Αρχικοποίηση ρυθμίσεων. Εδώ μπαίνουν οι εντολές που πρέπει να τρέξουν μία φορά όταν ενεργοποιείται το Arduino. Συνήθως μπαίνουν αρχικοποιήσεις τιμών μεταβλητών και σίγουρα ο χαρακτηρισμός των εισόδων/εξόδων που θα χρησιμοποιήσουμε (αν δηλαδή ένας συγκεκριμένος ακροδέκτης (pin) θα είναι είσοδος ή έξοδος).
- `loop ()` Κυρίως πρόγραμμα. Εδώ μπαίνουν οι εντολές, οι οποίες ξανατρέχουν σε ένα ατέρμονο loop. Αυτό θα συμβαίνει συνεχώς, όσο το Arduino τροφοδοτείται από ρεύμα.

Η βασική λειτουργία του Arduino είναι ότι τρέχει η συνάρτηση `setup ()` μία φορά στην αρχή και ακολούθως η συνάρτηση `loop ()` ξανά και ξανά μέχρι να κλείσουμε την τροφοδοσία. Στην περίπτωση που πατηθεί το κουμπί Reset ξανατρέχει η συνάρτηση `setup ()` μία φορά και ακολούθως η `loop ()` ξανά και ξανά. Όπως δηλαδή ακριβώς και όταν αρχικά ενεργοποιείται με ρεύμα ο μικροελεγκτής.

Σύμφωνα με τα παραπάνω, ένα τυπικό πρόγραμμα θα έχει την παρακάτω δομή:

```
void setup()
{
  //οι εντολές εδώ θα τρέξουν μόνο μία φορά στην ενεργοποίηση του Arduino ή μετά από ένα
  Reset
}

void loop()
{
  //οι εντολές εδώ θα τρέχουν ξανά και ξανά, μέχρι να απενεργοποιηθεί του Arduino ή να πατη-
  θεί το Reset
}
```

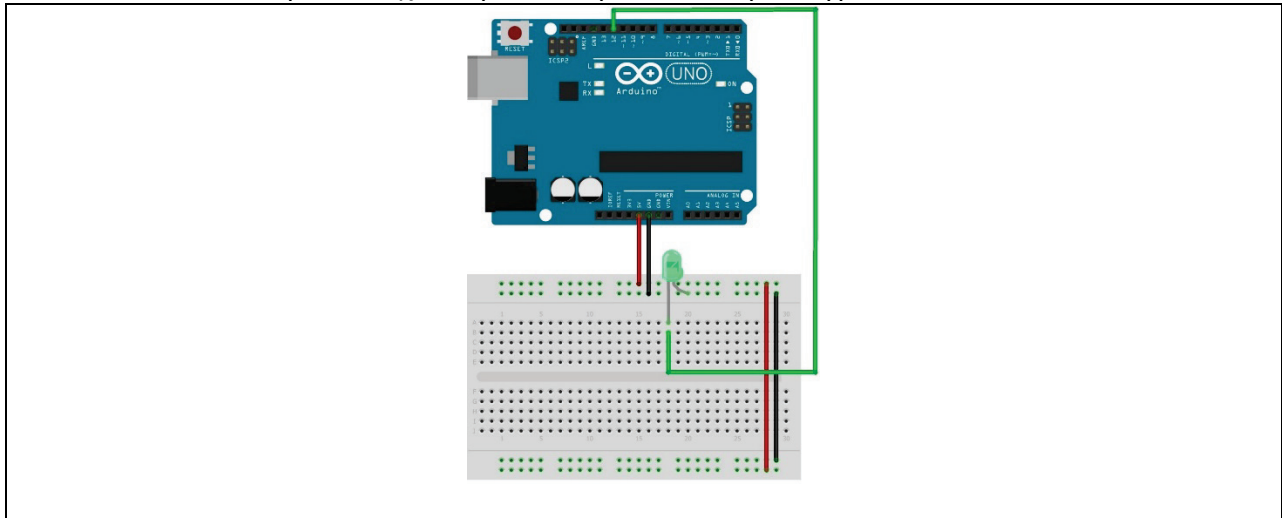
Κώδικας 2-2 Παράδειγμα Κώδικα , Δομή τυπικού κώδικα

Παρακάτω ακολουθεί ένα τυπικό πρόγραμμα το οποίο αναβοσβήνει απλά ένα LED.

```
#define ledpin 12
void setup () {
  pinMode (ledpin, OUTPUT); //ορισμός του pin 12 ως ψηφιακή έξοδο
}
void loop () {
  digitalWrite (ledpin, HIGH); //άναψε το LED
  delay (1000); //περίμενε 1 second (1000 milliseconds)
  digitalWrite (ledpin, LOW); //σβήσε το LED
  delay (1000); //περίμενε 1 second (1000 milliseconds)
}
```

Κώδικας 2-3 Παράδειγμα Κώδικα , Τυπικό πρόγραμμα το οποίο αναβοσβήνει απλά ένα LED

Ακολουθεί το κυκλωματικό σχέδιο για το παραπάνω παράδειγμα κώδικα.



Εικόνα 2-37 Κυκλωματικό σχέδιο για τον έλεγχο του led

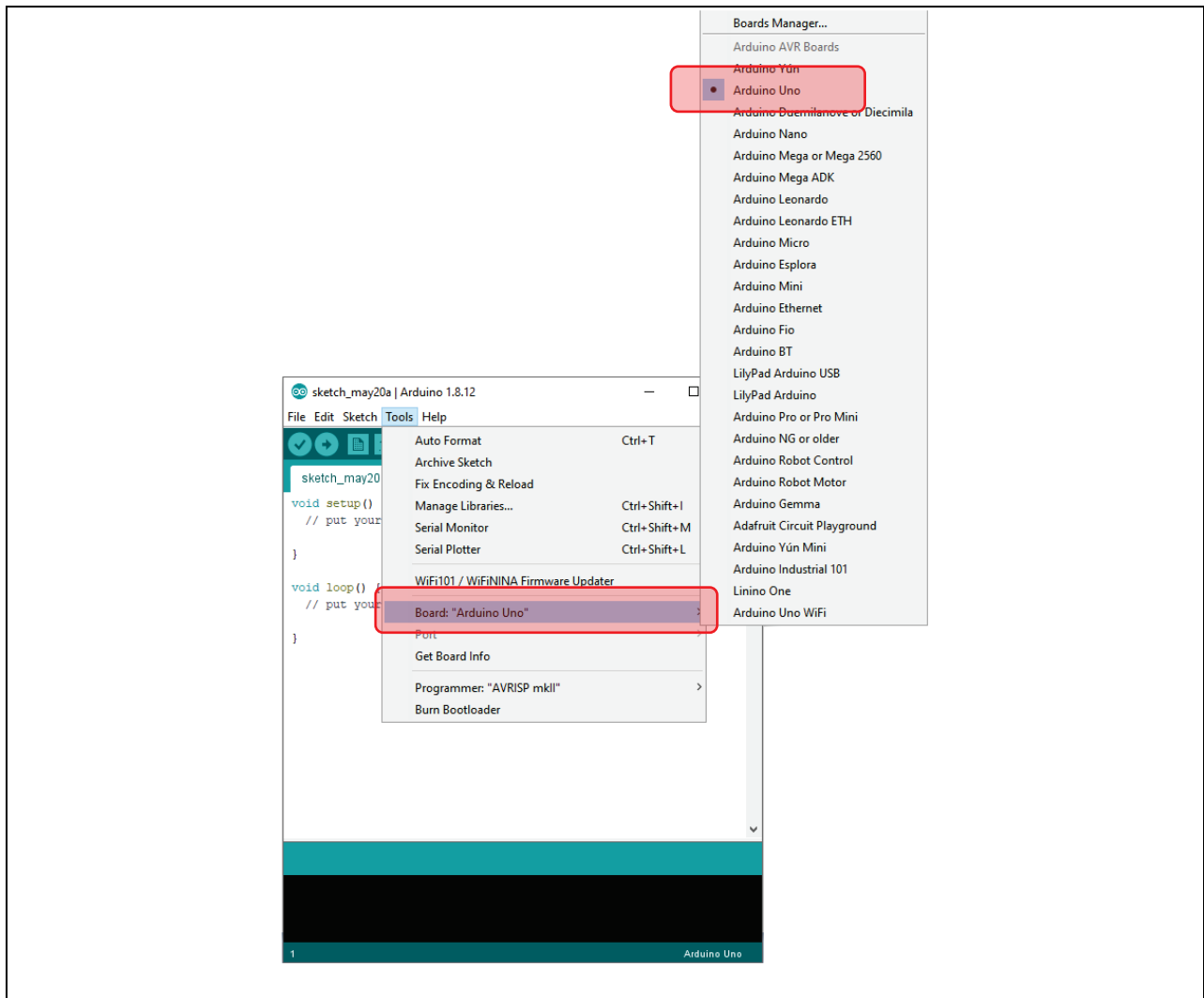
2.4.6 Περιβάλλον ARDUINO IDE – Φόρτωμα κώδικα στην πλακέτα

Για να φορτωθεί ένα πρόγραμμα (κώδικας) από το Arduino IDE σε μία πλακέτα, πρέπει πρώτα να συνδεθεί η πλακέτα με τον υπολογιστή μέσω ενός κατάλληλου καλωδίου USB. Αν έχουν φορτωθεί σωστά οι drivers ο υπολογιστής βλέπει την πλακέτα Arduino ως μία εικονική σειριακή θύρα (π.χ. στα Windows COMx).

Σύντομα βήματα για να ρυθμίσουμε τον τύπο πλακέτας στο Arduino IDE:

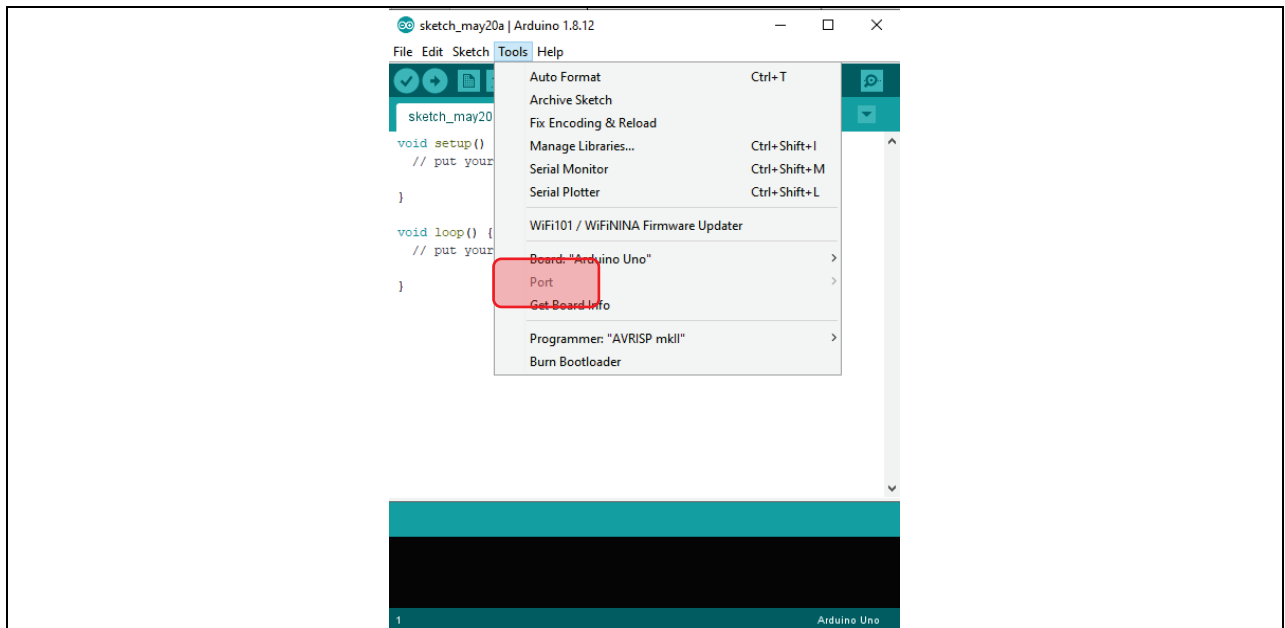
- Στο Arduino IDE, πηγαίνουμε στο μενού "Εργαλεία" (Tools).
- Από το υπομενού "Πλακέτα" (Board), επιλέγουμε τον τύπο της πλακέτας που έχουμε συνδέσει, όπως "Arduino Uno" ή οποιαδήποτε άλλη έκδοση που διαθέτουμε.
- Αφού επιλέξουμε τον κατάλληλο τύπο πλακέτας, το Arduino IDE είναι πλέον έτοιμο να επικοινωνήσει και να προγραμματίσει την πλακέτα μας.

Τα βήματα είναι απλά και επιτρέπουν στο Arduino IDE να αναγνωρίσει και να επικοινωνήσει με την πλακέτα που έχουμε συνδέσει.



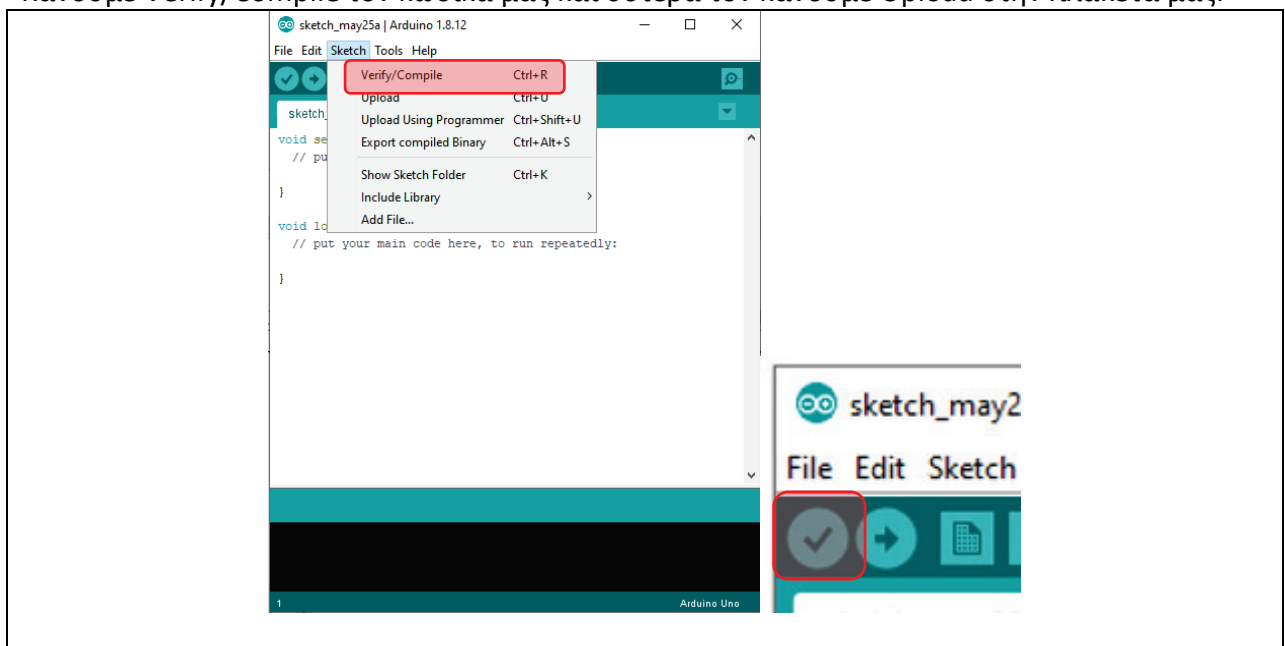
Εικόνα 2-38 Επιλογή πλακέτας Arduino

Στη συνέχεια ορίζουμε τη σειριακή θύρα στην οποία βλέπει ο υπολογιστής μας την πλακέτα. Από το μενού πατάμε Εργαλεία και στη συνέχεια βάζουμε το ποντίκι πάνω στην επιλογή Θύρα. Εμφανίζεται μία λίστα με τις διαθέσιμες σειριακές θύρες του υπολογιστή και επιλέγουμε την κατάλληλη.

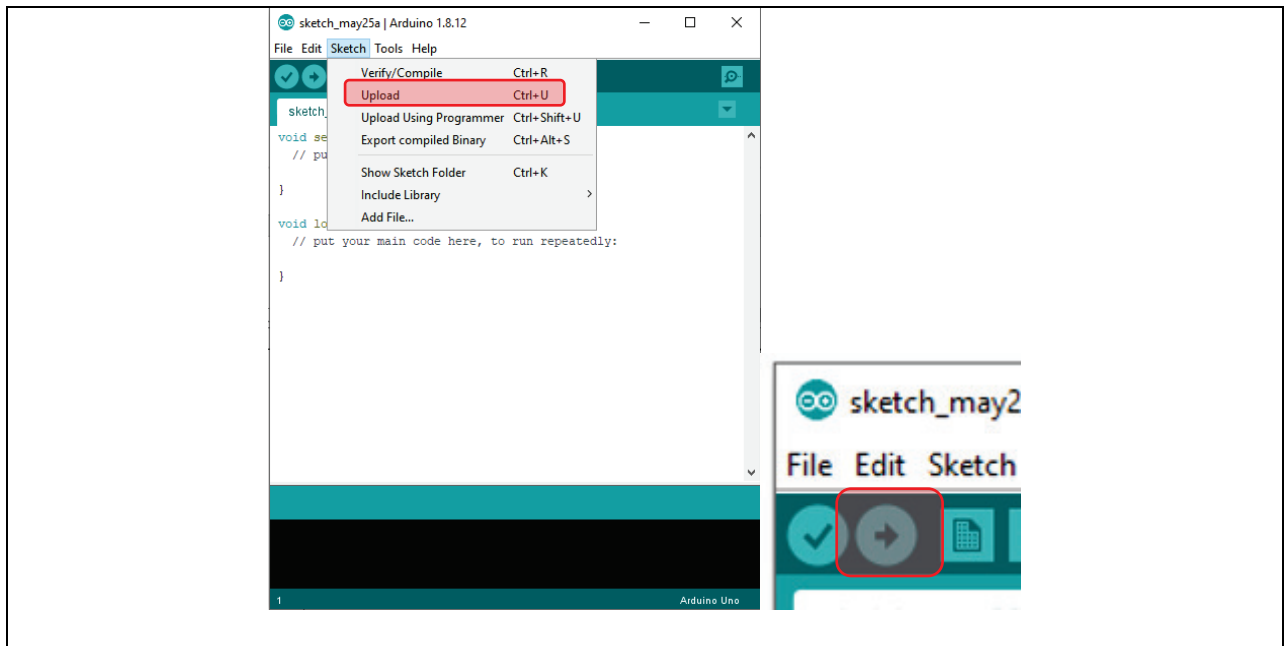


Εικόνα 2-39 Επιλογή θύρας usb την οποία είναι συνδεδεμένη η πλακέτα Arduino

Κάνουμε Verify/Compile τον κώδικά μας και ύστερα τον κάνουμε Upload στην πλακέτα μας.

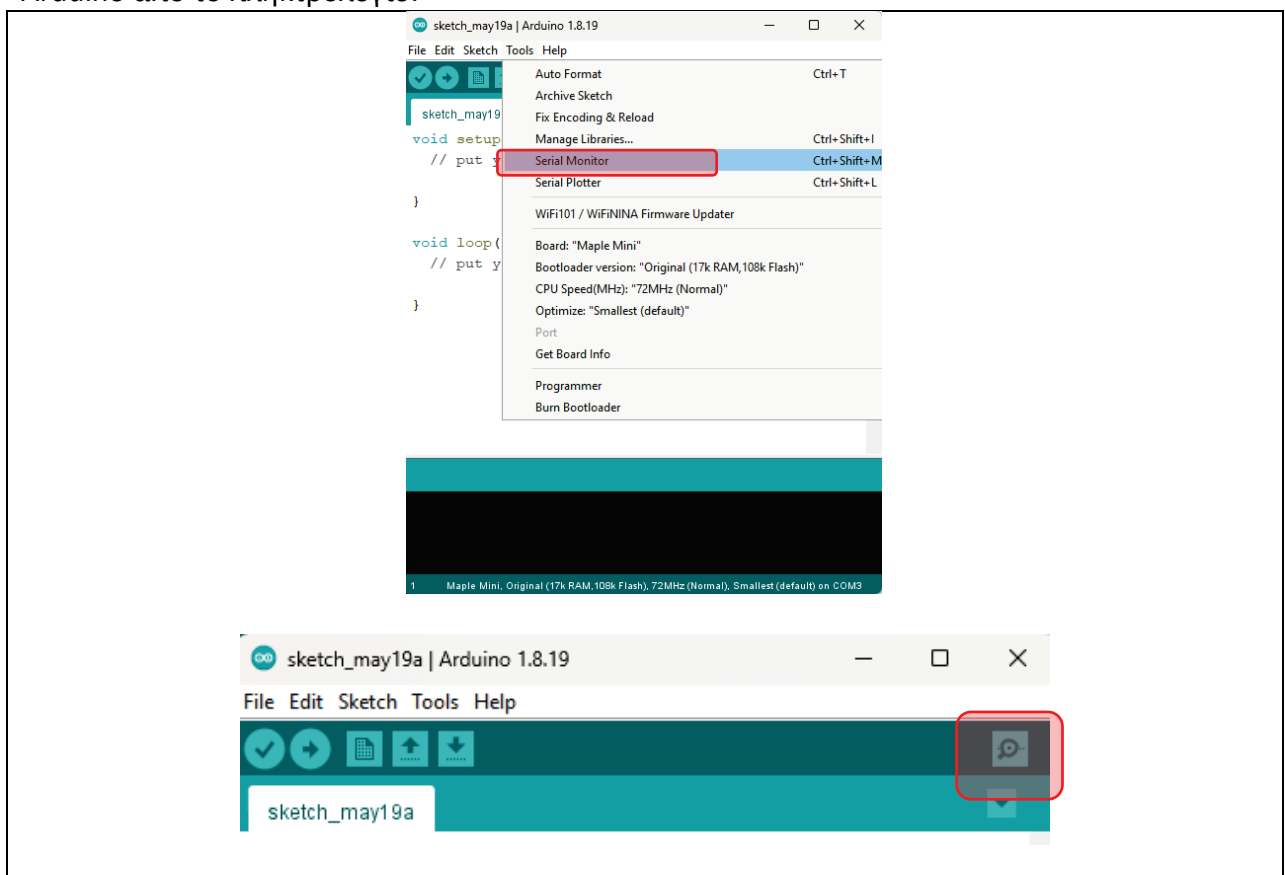


Εικόνα 2-40 Έλεγχος του προγράμματος για συντακτικά λάθη



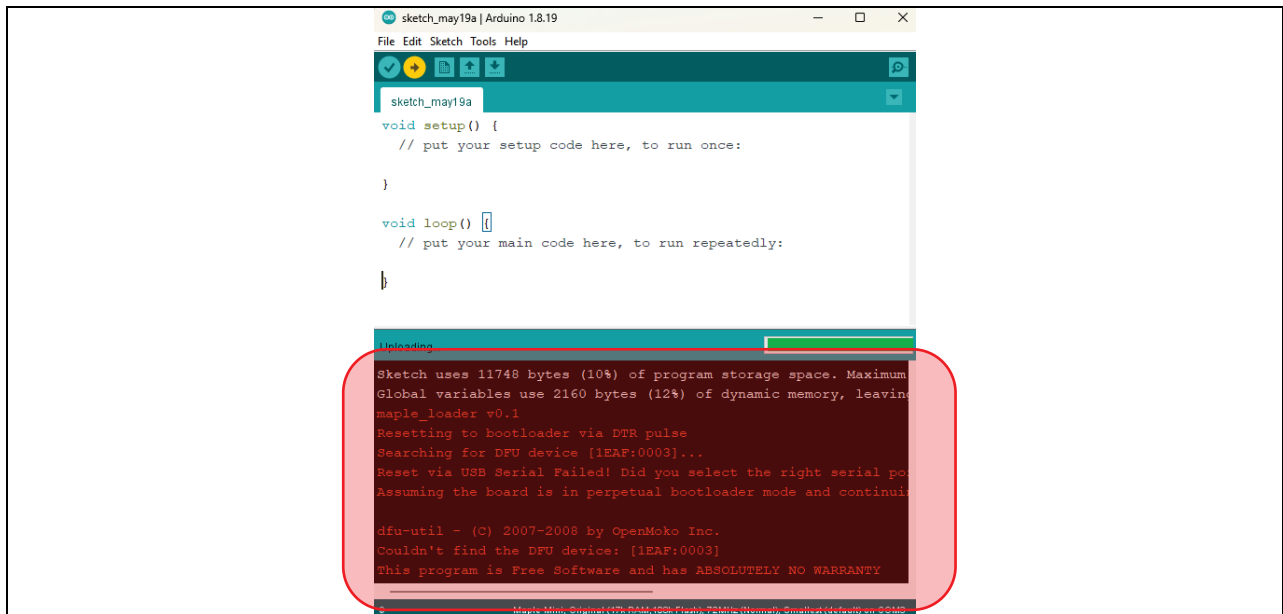
Εικόνα 2-41 Ανέβασμα του κώδικα στην πλακέτα Arduino

Η σειριακή οθόνη είναι ο «σύνδεσμος» μεταξύ του υπολογιστή και του Arduino. Επιτρέπει την αποστολή και την λήψη μηνυμάτων κειμένου, εύχρηστα για αποσφαλμάτωση ή για έλεγχο του Arduino από το πληκτρολόγιο.



Εικόνα 2-42 Σειριακή οθόνη

Ένδειξη σφαλμάτων που μπορεί να προκύψουν είτε κατά την μεταμόρφωση είτε κατά το ανέβασμα



Εικόνα 2-43 Ένδειξη σφαλμάτων

2.3.2 Βασικές εντολές

Η γλώσσα προγραμματισμού Arduino μπορεί να χωριστεί σε τρία κύρια μέρη:

1. Συναρτήσεις
2. Τιμές (μεταβλητές, σταθερές)
3. Δομή

Συναρτήσεις	
Digital I/O	
digitalRead()	Διαβάζει την τιμή από έναν ψηφιακό ακροδέκτη, είτε HIGH είτε LOW.
digitalWrite()	Γράφει μια τιμή HIGH ή LOW σε έναν ψηφιακό ακροδέκτη.
pinMode()	Ρυθμίζει ένα pin ώστε να συμπεριφέρεται ως είσοδος ή έξοδος
Analog I/O	
analogRead()	Διαβάζει την τιμή από έναν αναλογικό ακροδέκτη, είτε HIGH είτε LOW.
analogWrite()	Γράφει μια τιμή HIGH ή LOW σε έναν αναλογικό ακροδέκτη.
Time	
delay()	Παύει το πρόγραμμα για ένα χρονικό διάστημα (σε χιλιοστά του δευτερολέπτου)
millis()	Επιστρέφει τον αριθμό των χιλιοστών του δευτερολέπτου που πέρασαν από την έναρξη λειτουργίας του τρέχοντος προγράμματος Arduino.

Πίνακας 2-4 Συναρτήσεις

(Arduino Docs | Arduino Documentation, n.d.)

Τιμές (μεταβλητές, σταθερές)	
Constants	
Integer Constants	Οι ακέραιες σταθερές είναι αριθμοί που χρησιμοποιούνται απευθείας στον κώδικα, όπως 123.
HIGH LOW	Κατά την εγγραφή ή ανάγνωση σε ένα ψηφιακό pin υπάρχουν δύο τιμές που μπορεί να πάρει: HIGH και LOW
INPUT OUTPUT	Δήλωση ενός ψηφιακού pin ως είσοδος η έξοδος
true false	Σταθερές που δηλώνουν αλήθεια ή ψέμα
Data Types	

byte	Ένα byte αποθηκεύει έναν 8-bit μη υπογεγραμμένο αριθμό, από 0 έως 255.
char	Ένας τύπος δεδομένων που χρησιμοποιείται για την αποθήκευση μιας τιμής χαρακτήρα.
array	Ένας πίνακας είναι μια συλλογή μεταβλητών στον οποίο γίνεται πρόσβαση με έναν αριθμό ευρετηρίου.
bool	Ένα bool έχει δύο περιπτώσεις τιμών αξίες, αληθής ή ψευδής. (Κάθε μεταβλητή bool καταλαμβάνει ένα byte μνήμης.)
float	Αριθμός που περιέχει δεκαδικό ψηφίο
int	Ακέραιος αριθμός
long	Μεταβλητές μεγάλου μεγέθους για την αποθήκευση αριθμών Καταλαμβάνουν 32 bits (4 bytes), από -2,147,483,648 έως 2,147,483,647.
unsigned long	Σε αντίθεση με τις standard longs, οι unsigned longs δεν αποθηκεύουν αρνητικούς αριθμούς, κάνοντας το φάσμα από 0 έως 4.294.967.295 .
void	Χρησιμοποιείται στις δηλώσεις συναρτήσεων.
double	Αριθμός κινητής υποδιαστολής δύο δεκαδικών. Στο Uno και σε άλλες πλακέτες που βασίζονται στο ATMEGA, καταλαμβάνει 4 byte.

Πίνακας 2-5 Τιμές (μεταβλητές, σταθερές)

(Arduino Docs | Arduino Documentation, n.d.)

Αριθμητικοί τελεστές	
+	(πρόσθεση)
-	(αφαίρεση)
*	(πολλαπλασιασμός)
/	(διαίρεση)
%	(υπόλοιπο ακέραιας διαίρεσης)
=	(ισότητα)
Comparison Operators	
==	(ίσο με)
!=	(όχι ίσο)
<	(μικρότερο)
<=	(μικρότερο ή ίσο)
>	(μεγαλύτερο)
>=	(μεγαλύτερο ή ίσο)
Boolean Operators	
!	(λογικό όχι)
&&	(λογικό και)
	(λογικό ή)

Πίνακας 2-6 Αριθμητικοί τελεστές

(Arduino Docs | Arduino Documentation, n.d.)

Δομή Ελέγχου	
break	Έξοδος από for, while, do...while loop και switch case
if (else)	έλεγχος μίας συνθήκης (πολλαπλών συνθηκών)
continue	Η εντολή continue παραλείπει την υπόλοιπη τρέχουσα επανάληψη ενός βρόχου

do...while	Η do ... while loop λειτουργεί με τον ίδιο τρόπο η while, με την εξαίρεση ότι ο έλεγχος γίνεται στο τέλος του βρόχου, οπότε ο βρόχος do θα τρέχει τουλάχιστον μία φορά.
for	Δομή επανάληψης όπου χρησιμοποιείται μετρητής αύξησης για την αύξηση και τον τερματισμό του βρόχου.
while	Ένας βρόχος while θα εκτελείται συνεχώς, και απεριόριστα, μέχρι η έκφραση μέσα στην παρένθεση () να γίνει ψευδής.
switch...case	Μία switch statement συγκρίνει την τιμή μιας μεταβλητής με τις τιμές που καθορίζονται στις case statements. Όταν εντοπιστεί μια case statement της οποίας η τιμή ταιριάζει με εκείνη της μεταβλητής, εκτελείται ο κώδικας.

Πίνακας 2-7 Δομή Ελέγχου

(Arduino Docs | Arduino Documentation, n.d.)

Περαιτέρω σύνταξη	
#define (define)	Απόδοση ονόματος σε μία σταθερά
#include (include)	Χρησιμοποιείται για να συμπεριλάβει εξωτερικές βιβλιοθήκες στον κώδικα.
#define (define)	Απόδοση ονόματος σε μία σταθερά
#include (include)	Χρησιμοποιείται για να συμπεριλάβει εξωτερικές βιβλιοθήκες στον κώδικα.
/* */ (block comment)	Σχόλια πολλών γραμμών. Τα σχόλια είναι γραμμές στο πρόγραμμα που χρησιμοποιούνται για την διευκρίνιση του τρόπου λειτουργίας του προγράμματος.
// (single line comment)	Σχόλιο μίας γραμμής
; (semicolon)	Μπαίνει στο τέλος μίας δήλωσης ή εντολής
{ } (curly braces)	Άνοιγμα/ κλείσιμο δομών επανάληψης, συναρτήσεων και δομών επιλογής

Πίνακας 2-8 Further Syntax

(Arduino Docs | Arduino Documentation, n.d.)

2.3.3 Καταχώρηση βιβλιοθηκών

Στην αρχή κάθε προγράμματος γίνεται εισαγωγή των βιβλιοθηκών για την εύρυθμη λειτουργία του κώδικα και την αναγνώριση των αισθητήρων. Μερικές ενδεικτικές βιβλιοθήκες ακολουθούν παρακάτω:

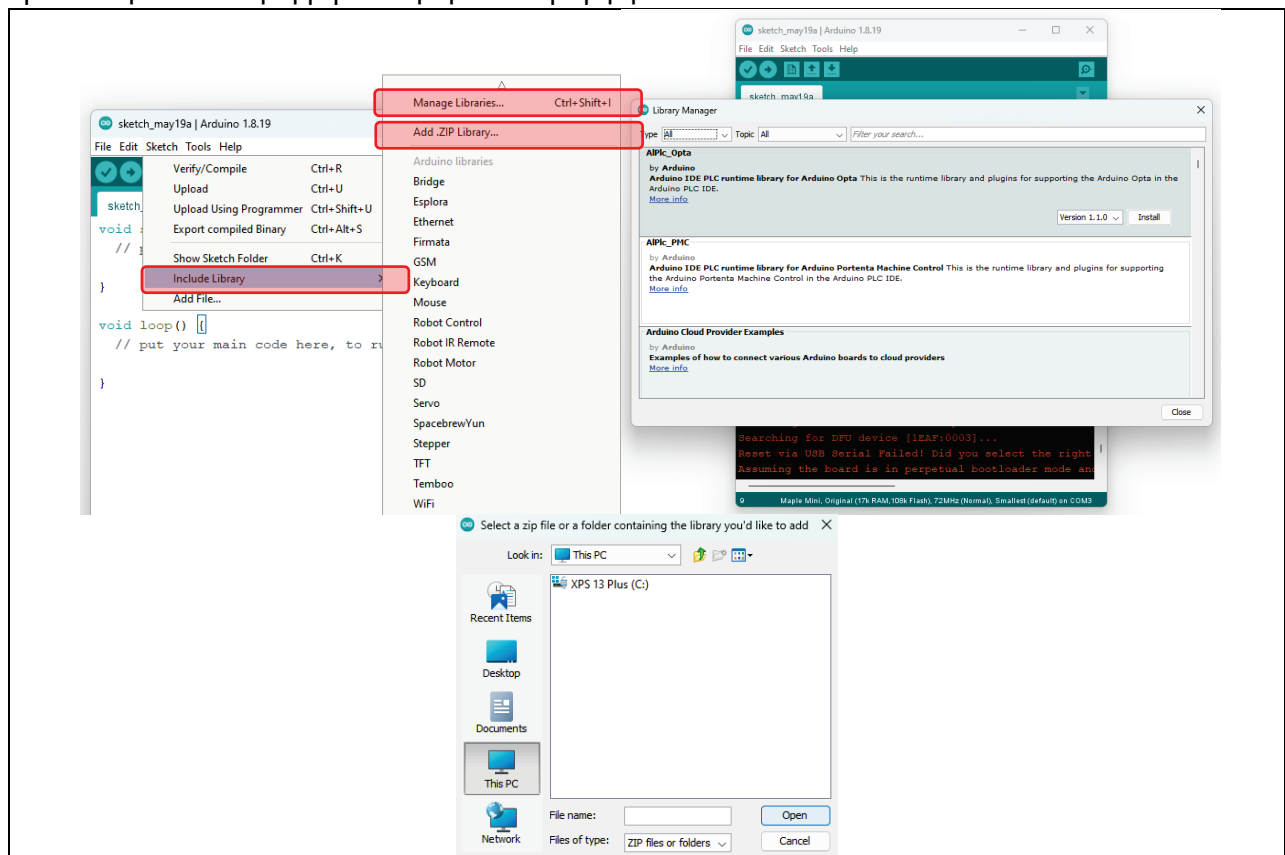
ΒΙΒΛΙΟΘΗΚΗ	ΕΠΕΞΗΓΗΣΗ
ETHERNET	Επιτρέπει τη σύνδεση του Arduino με το Internet. Είτε αποδέχοντας συνδέσεις ως server είτε αποστέλοντας ως client.
WIRE	Επιτρέπει την επικοινωνία με συσκευές που υποστηρίζουν το πρωτόκολλο I2C.
SPI	Χρησιμοποιείται για την σειριακή επικοινωνία μεταξύ περιφερειακών συσκευών. Στην περίπτωση μας του Ethernet Shield με το Arduino ως Master.
SD	Επιτρέπει την εγγραφή και την ανάγνωση από την κάρτα μνήμης SD.

THINGSPEAK	Είναι η βιβλιοθήκη επικοινωνίας της υπηρεσίας Thingspeak με τον κώδικα.
DHT	Είναι η βιβλιοθήκη για την αναγνώριση του αισθητήρα θερμοκρασίας και υγρασίας.
SOFTWARE SERIAL	Επιτρέπει την σειριακή επικοινωνία με άλλες ψηφιακές ακίδες του Arduino, χρησιμοποιώντας λογισμικό για την αναπαραγωγή της λειτουργικότητας (εξ ου και το όνομα "SoftwareSerial").

Πίνακας 2-9 Ενδεικτικές Βιβλιοθήκες

(Arduino Docs | Arduino Documentation, n.d.)

Η εγκατάσταση των βιβλιοθηκών γίνεται μέσω της επιλογής Tools / Include Library. Είτε μέσω της επιλογής Manage Libraries, είτε μέσω της επιλογής Add ZIP Library, εάν έχουμε ήδη κατεβάσει την κατάλληλη βιβλιοθήκη σε ZIP μορφή.



Εικόνα 2-44 Εγκατάσταση των βιβλιοθηκών

2.5 ΛΟΓΙΣΜΙΚΟ (SOFTWARE) - ARDUBLOCK

2.5.1 Γενικά

Όπως αναφέρθηκε προηγουμένως, το Arduino IDE υποστηρίζει τις γλώσσες C και C++ χρησιμοποιώντας ειδικούς κανόνες για τη δομή του κώδικα. Ωστόσο, υπάρχουν και άλλα λογισμικά, όπως το ArduBlock, που προσφέρουν μια εναλλακτική προσέγγιση στον προγραμματισμό του Arduino.

Το ArduBlock είναι ένα περιβάλλον οπτικού προγραμματισμού που επιτρέπει στους χρήστες να δημιουργούν κώδικα για το Arduino χωρίς να γράφουν απευθείας κώδικα C/C++. Αντ' αυτού, οι χρήστες μπορούν να συνδυάζουν έτοιμα "πλακίδια" κώδικα μέσω της τεχνικής drag-and-drop. Το ArduBlock εκτελείται μέσα από το ίδιο το Arduino IDE, παρέχοντας μια φιλική και οπτική εναλλακτική για τον προγραμματισμό του Arduino, ιδιαίτερα για αρχάριους χρήστες.

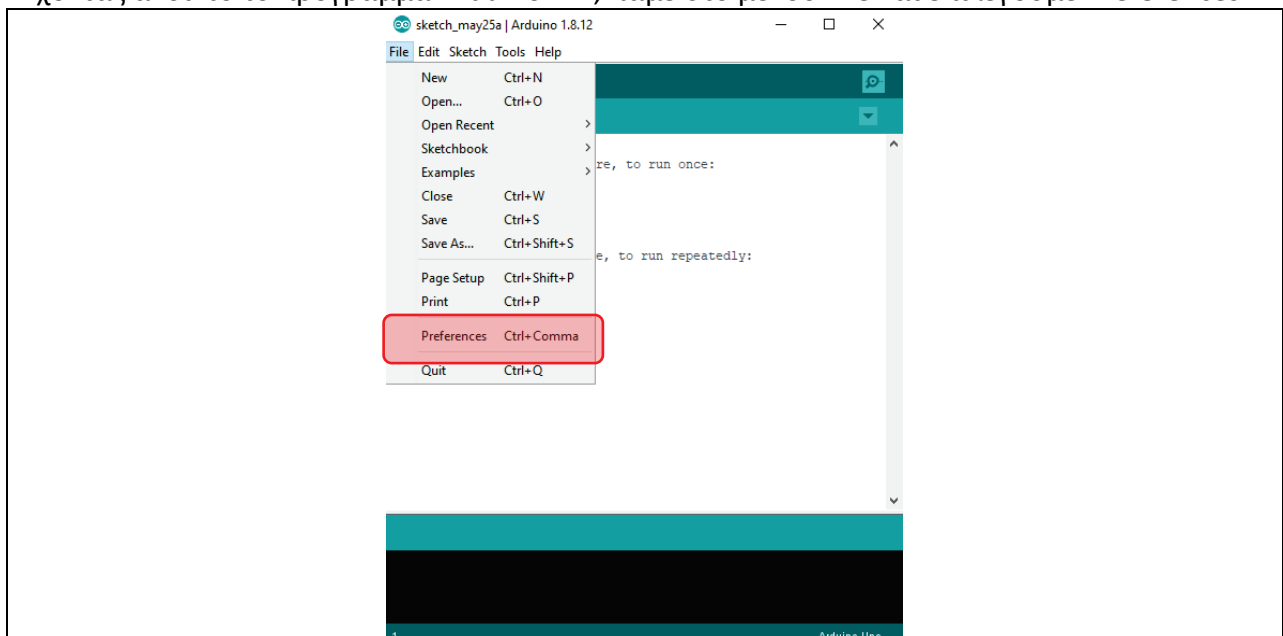
2.5.2 Εγκατάσταση

Μπορείτε να κατεβάσετε το πρόγραμμα μέσω της διεύθυνσης

<https://github.com/taweili/ardublock/releases>. Επιλέξτε το αρχείο ardublock-all-master-beta-20171028.jar ή πιο καινούργια έκδοση.

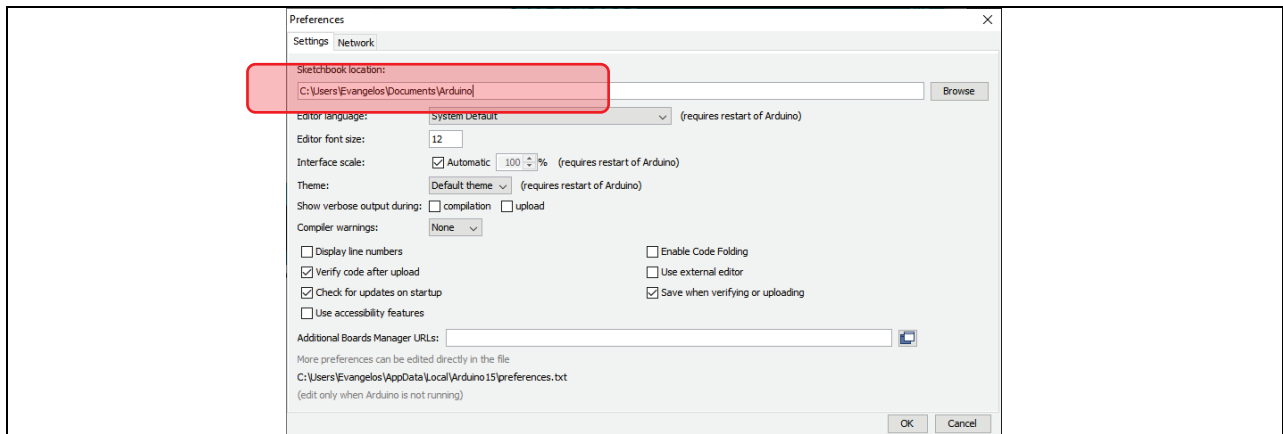
Για να ενσωματωθεί το ArduBlock στο Arduino IDE πρέπει να γίνουν τα παρακάτω βήματα:

Έχοντας ανοικτό το πρόγραμμα Arduino IDE, πάμε στο μενού File και επιλέγουμε Preferences.



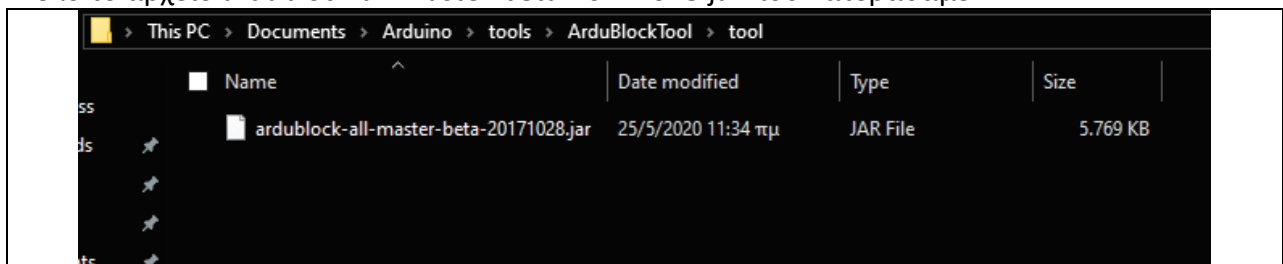
Εικόνα 2-45 Εγκατάσταση Ardublock

Στο παράθυρο που ανοίγει βλέπουμε τη θέση του Sketchbook (Sketchbook Location), δηλαδή του φακέλου όπου αποθηκεύονται τα προγράμμάτα που φτιάχνουμε με το πρόγραμμα Arduino IDE.



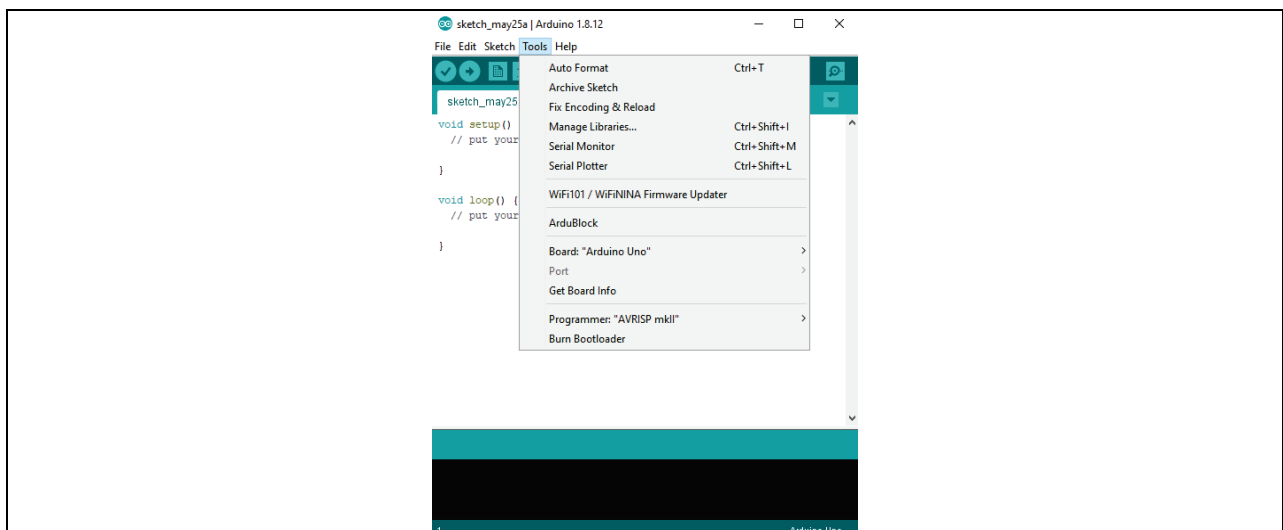
Εικόνα 2-46 Θέση του Sketchbook (Sketchbook Location)

Για την ενσωμάτωση του ArduBlock μέσα στο IDE, πάμε στον παραπάνω φάκελο και δημιουργηθεί την δομή φακέλων tools\ArduBlockTool\tool. Έπειτα αντιγράφουμε μέσα σε αυτόν φάκελο το αρχείο ardublock-all-master-beta-20171028.jar που κατεβάσαμε.



Εικόνα 2-47 Δομή φακέλων

Στην επόμενη επανεκκίνηση του IDE, το ArduBlock θα εμφανιστεί ως επιλογή μέσα στο μενού Εργαλεία.



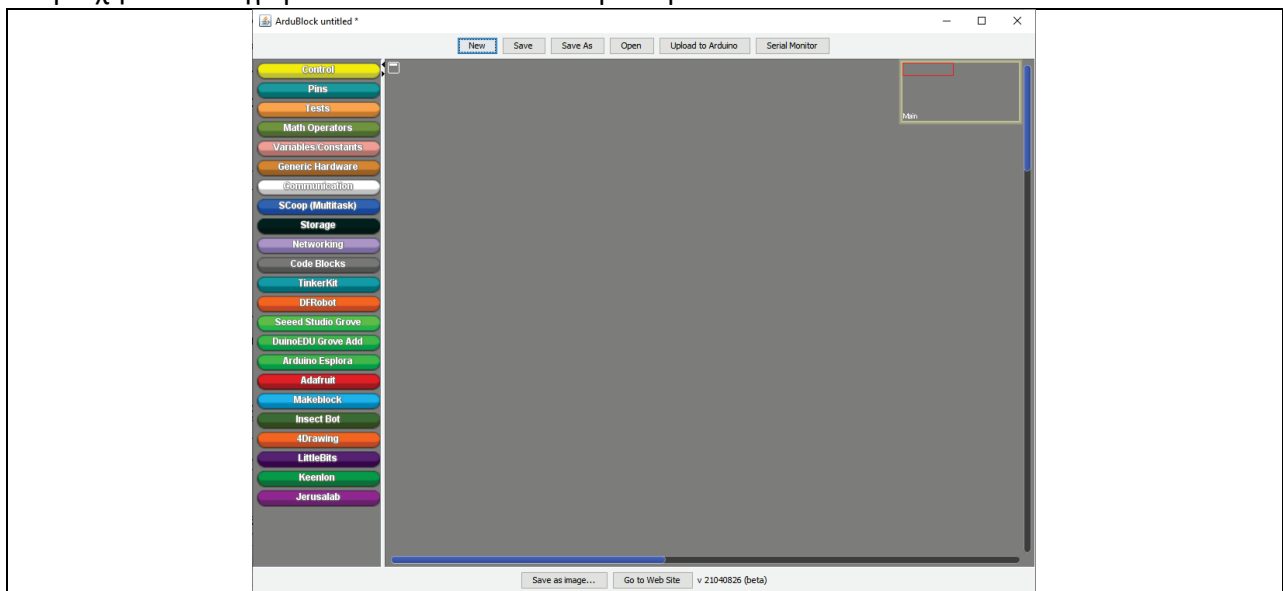
Εικόνα 2-48 Επιλογή ArduBlock από το μενού του Arduino IDE

2.5.3 Περιβάλλον και Χρήση του ArduBlock

Το περιβάλλον του ArduBlock διαχωρίζεται σε δύο βασικά μέρη:

- Στο αριστερό μέρος του παραθύρου υπάρχουν κουμπιά που αντιπροσωπεύουν τις διαφορετικές κατηγορίες των εντολών. Όταν επιλέγετε μία από αυτές τις κατηγορίες, εμφανίζεται μία λίστα με τα αντίστοιχα πλακίδια των εντολών που μπορείτε να προσθέσετε στο πρόγραμμά σας.
- Στο δεξιό μέρος του περιβάλλοντος υπάρχει μία κενή, γκρι περιοχή. Σε αυτήν την περιοχή "σχεδιάζετε" το πρόγραμμά σας, προσθέτοντας εντολές από τη λίστα στα αριστερά. Μπορείτε να συνδέσετε τις εντολές μεταξύ τους για να δημιουργήσετε τη ροή του προγράμματος.

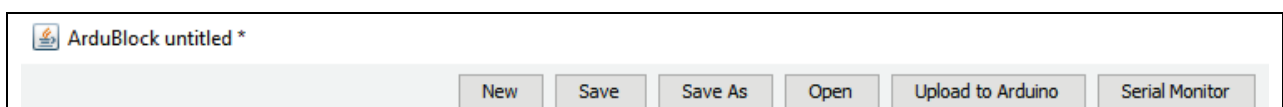
Για να αφαιρέσετε μία ή περισσότερες εντολές από το πρόγραμμα, απλά σύρετέ τις πίσω στην περιοχή των κατηγοριών των εντολών στα αριστερά.



Εικόνα 2-49 Περιβάλλον ArduBlock

Με τα κουμπιά στο επάνω μέρος του παραθύρου μπορούμε να δημιουργήσουμε ένα νέο πρόγραμμα (New), να αποθηκεύσουμε το πρόγραμμα (Save, Save as) και να ανοίξουμε ένα αρχείο (Open).

Όταν πατηθεί το κουμπί Upload to Arduino, το πρόγραμμα που έχουμε φτιάξει στο ArduBlock, μεταφράζεται σε Wiring C μέσα στο ανοικτό παράθυρο του Arduino IDE και στη συνέχεια φορτώνεται από το IDE στην πλακέτα.

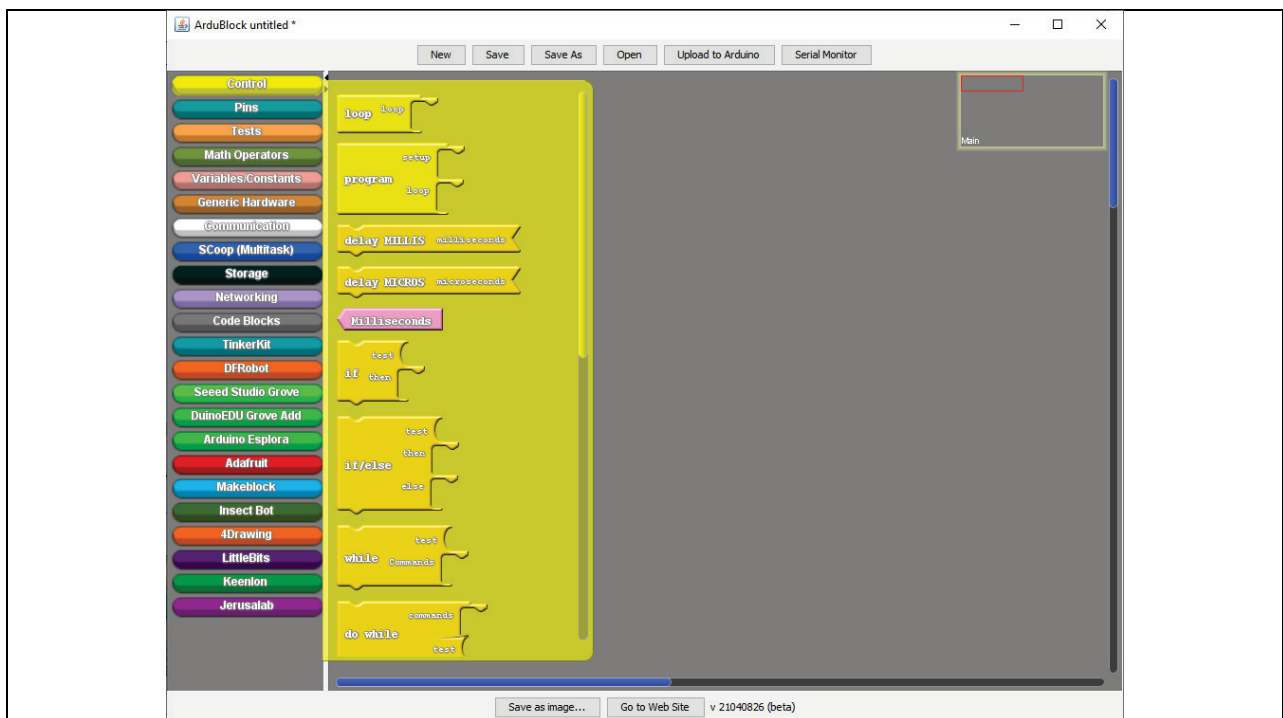


Εικόνα 2-50 Περιβάλλον ArduBlock

Για να ξεκινήσετε ένα νέο σκίτσο ArduBlock, το πρώτο βήμα είναι να προσθέσετε το βασικό πλακίδιο προγράμματος (Program block). Αυτό το πλακίδιο μπορείτε να το βρείτε στην κατηγορία "Control" στο αριστερό μέρος του περιβάλλοντος.

Το πλακίδιο προγράμματος ορίζει τις βασικές λειτουργίες `setup()` και `loop()` που απαιτούνται σε κάθε πρόγραμμα Arduino. Η `setup()` εκτελείται μία φορά κατά την εκκίνηση του προγράμματος, ενώ η `loop()` εκτελείται επαναληπτικά μέχρι να σταματήσει το πρόγραμμα.

Αφού προσθέσετε το πλακίδιο προγράμματος, μπορείτε να σύρετε και να τοποθετήσετε επιπλέον πλακίδια εντολών μέσα στις λειτουργίες `setup()` ή `loop()`, ανάλογα με τις ανάγκες του προγράμματός σας. Έτσι θα δημιουργείτε σταδιακά το τελικό σκίτσο.



Εικόνα 2-51 Περιβάλλον ArduBlock

2.5.4 Λογισμικό Ardublock – Πρώτος Κώδικας

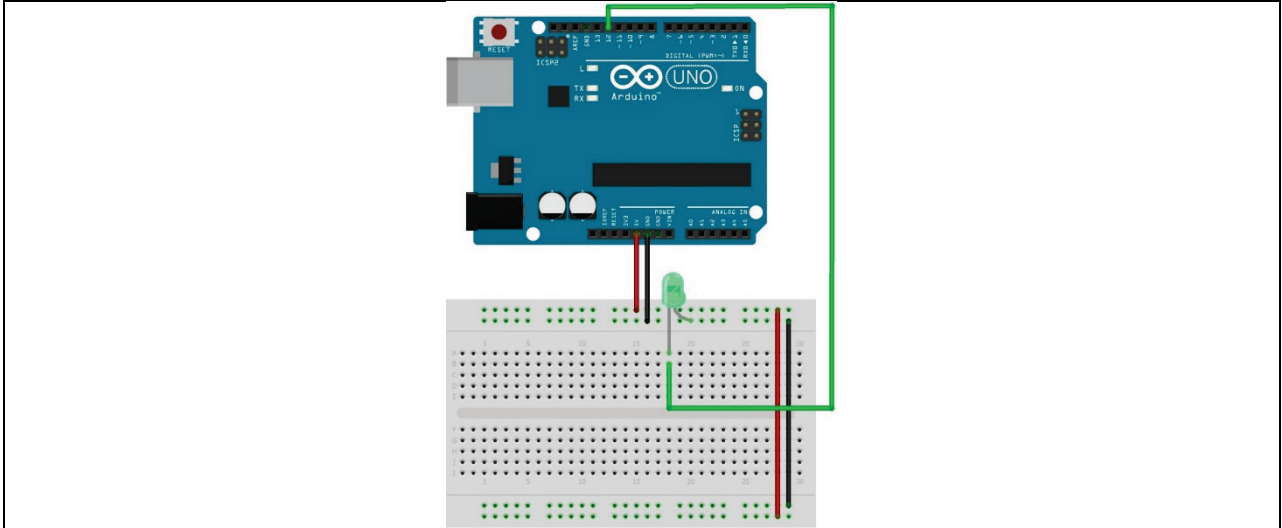
Ας δοκιμάσουμε να υλοποιήσουμε το παρακάτω παράδειγμα (το οποίο αναβοσβήνει απλά ένα LED), που παρουσιάστηκε και στο Arduino IDE.

```
#define ledpin 12
void setup () {
  pinMode (ledpin, OUTPUT); //ορισμός του pin 12 ως ψηφιακή έξοδο
}
void loop () {
  digitalWrite (ledpin, HIGH); //άναψε το LED
  delay (1000); //περίμενε 1 second (1000 milliseconds)
  digitalWrite (ledpin, LOW); //σβήσε το LED
```

```
delay (1000); //περίμενε 1 second (1000 milliseconds)
}
```

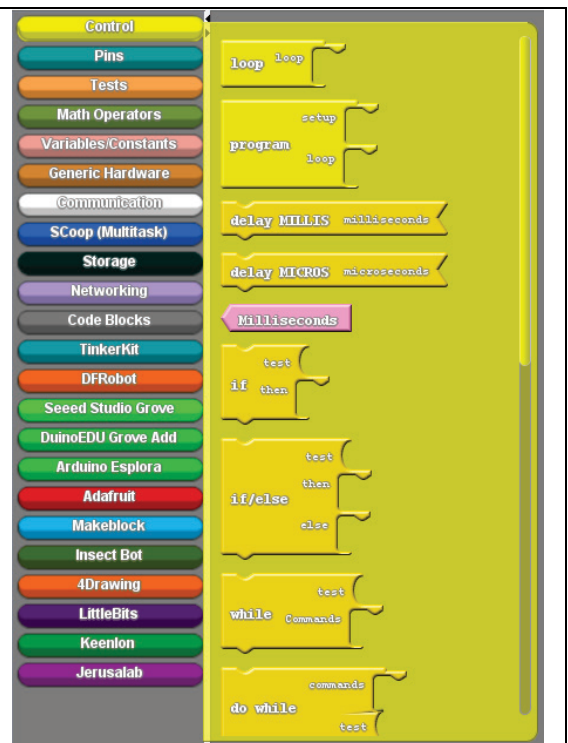
Κώδικας 2-4 Παράδειγμα Κώδικα, Ardublock – Πρώτος Κώδικας

Ακολουθεί το κυκλωματικό σχέδιο για το παραπάνω παράδειγμα κώδικα.

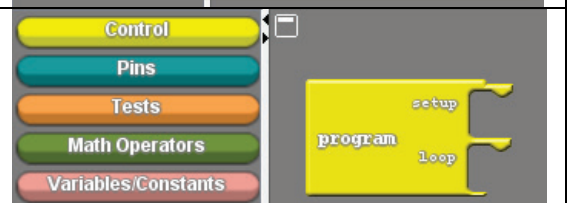


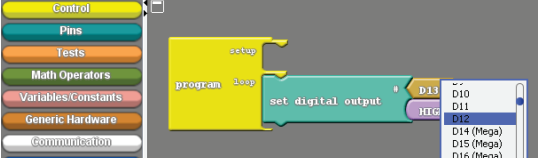
Εικόνα 2-52 Κυκλωματικό σχέδιο για τον έλεγχο του led

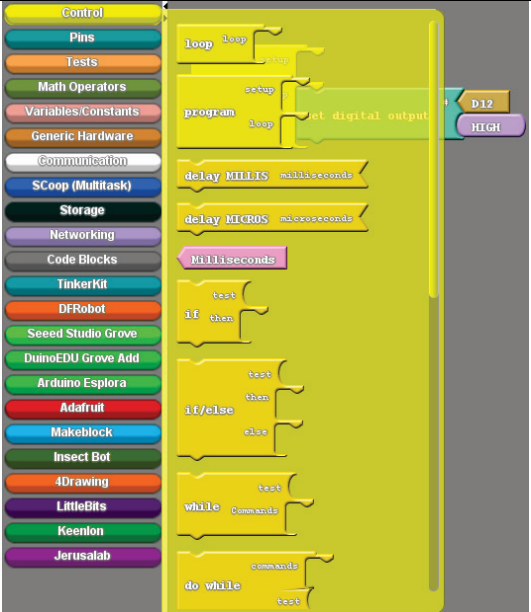
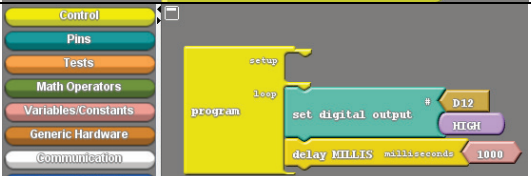
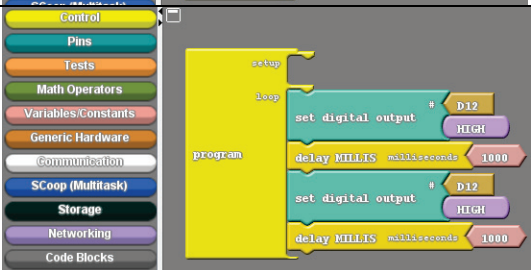
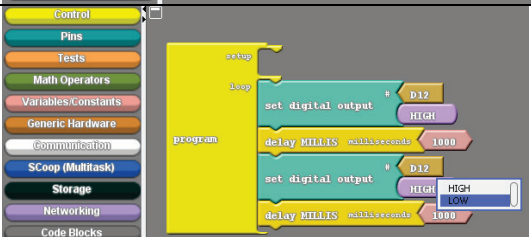
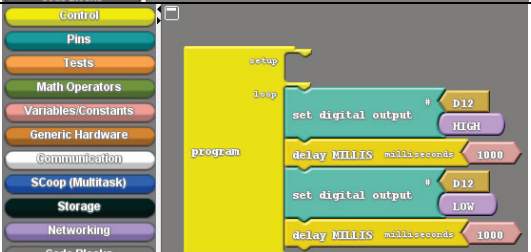
Εμφανίζουμε τα πλακίδια της κατηγορίας Control.



Σύρουμε ένα πλακίδιο Program.

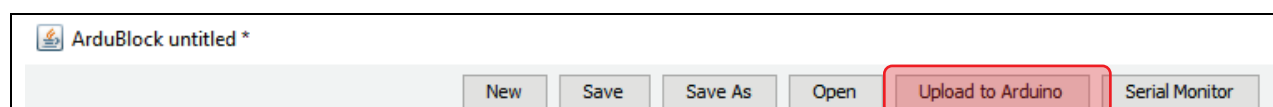


Εμφανίζουμε τα πλακίδια της κατηγορίας Pins.	
Σύρουμε ένα πλακίδιο Set Digital Output, μέσα στο πλακίδιο Programm/Loop.	
Αλλάζουμε στο Set Digital Output, το Pin από D13 σε D12.	
Μετά την αλλαγή του pin.	

Εμφανίζουμε τα πλακίδια της κατηγορίας Control.	
Σύρουμε ένα πλακίδιο Delay millis, κάτω από το πλακίδιο set digital out.	
Επαναλαμβάνουμε τα 3 τελευταία βήματα.	
Αλλάζουμε την κατάσταση του Set Digital Output από High σε Low	
Τελικό Πρόγραμμα	

Πίνακας 2-10 Βήματα δημιουργίας του πρώτου προγράμματος

Πατώντας την επιλογή Upload to Arduino, μας δείχνει στο περιβάλλον του ARDUINO IDE, τον παρακάτω κώδικα που δημιουργήθηκε και τον ανεβάζει και στο Arduino.

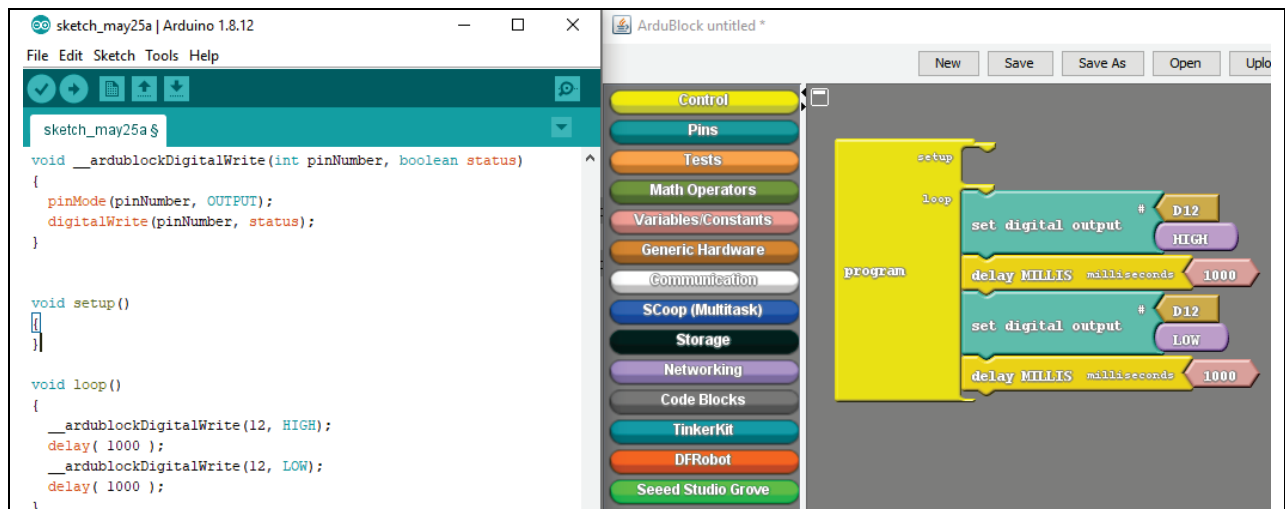


Εικόνα 2-53 Επιλογή Upload to Arduino


```
void __ardublockDigitalWrite(int pinNumber, boolean status)
{
  pinMode(pinNumber, OUTPUT);
  digitalWrite(pinNumber, status);
}
void setup()
{
}
void loop()
{
  __ardublockDigitalWrite(12, HIGH);
  delay( 1000 );
  __ardublockDigitalWrite(12, LOW);
  delay( 1000 );
}
```

Κώδικας 2-5 Παράδειγμα Κώδικα, Ardublock Κώδικας

Ενώ μπορούμε ανά πάσα στιγμή να συγκρίνουμε το γραφικό περιβάλλον με τα πλακίδια του προγράμματος μας με το αντίστοιχο περιβάλλον κώδικα C.



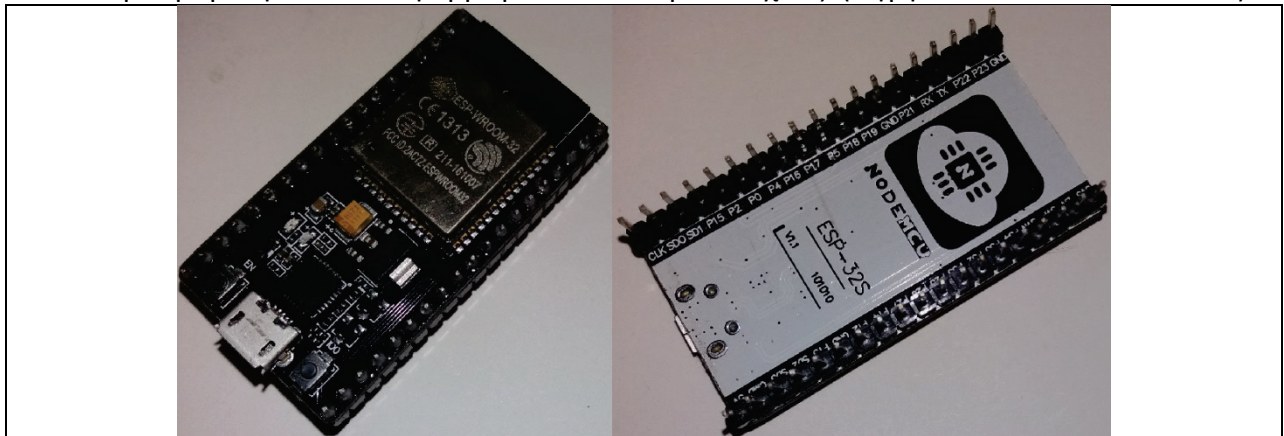
Εικόνα 2-54 Γραφικό περιβάλλον και κώδικας

ΚΕΦΑΛΑΙΟ 3

ΠΛΑΤΦΟΡΜΑ ESP32 – ΥΛΙΚΟ & ΛΟΓΙΣΜΙΚΟ

3.1 ΕΙΣΑΓΩΓΗ

Οι ESP32 πλατφόρμες είναι μια σειρά πλατφορμών-μικροελεγκτών χαμηλού κόστους και χαμηλής κατανάλωσης System on a Chip (SoC) που αναπτύχθηκε από την Espressif, σχεδιασμένο με το χαμηλής ισχύος τσιπ 40 nm της TSMC τεχνολογία. Έχει σχεδιαστεί για να επιτυγχάνει την καλύτερη δυνατή απόδοση ισχύος και RF, επιδεικνύοντας στιβαρότητα, ευελιξία και αξιοπιστία σε μια μεγάλη ποικιλία εφαρμογών και σεναρίων ισχύος. (πηγή: ESP32 Series Datasheet)



Εικόνα 3-1 Πλακέτα ESP32 (2 πλευρές)

(The Internet of Things With ESP32, n.d.)

Η σειρά τσιπ ESP32 περιλαμβάνει τα

- ESP32-D0WD-V3, ESP32-D0WDR2-V3, ESP32-U4WDH, ESP32-S0WD (NRND), ESP32-D0WDQ6-V3 (NRND), ESP32-D0WD (NRND) και ESP32-D0WDQ6 (NRND).
- ESP32-S0WD (NRND), ESP32-D0WD (NRND) και ESP32-D0WDQ6 (NRND) (chip v1 ή chip v1.1).

- Τα ESP32-D0WD-V3, ESP32-D0WDR2-V3, ESP32-U4WDH και ESP32-D0WDQ6-V3 (NRND) (chip v3.0 ή chip v3.1).

3.2 ΒΑΣΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ESP32

Ακολουθεί μία συνοπτική περιγραφή των βασικών χαρακτηριστικών του ESP32.

Επεξεργαστές:

- Ο κύριος επεξεργαστής είναι ο Tensilica Xtensa 32-bit LX6 microprocessor, ο οποίος έχει 1-2 Πυρήνες (ανάλογα από την υλοποίηση) και ο χρονισμός του στα 240MHz με απόδοση έως και 600 DMIPS (Dhrystone MIPS).

Ασύρματη επικοινωνία:

- Διαθέτει Wi-Fi: 802.11 b/g/n/e/i (802.11n 2.4 GHz έως 150 Mbit/s) και Bluetooth: v4.2 BR/EDR and Bluetooth Low Energy (BLE)

Εσωτερική μνήμη:

- ROM: 448 KiB, SRAM: 520 KiB, flash 0 MiB (ESP32-D0WDQ6, ESP32-D0WD, and ESP32-S0WD chips), 2 MiB (ESP32-D2WD chip), 4 MiB (ESP32-PICO-D4 SiP module)

Εξωτερική μνήμη:

- Εξωτερική flash & SRAM: Το ESP32 υποστηρίζει έως και τέσσερις εξωτερικές QSPI flash μνήμες των 16 MiB και SRAM με κρυπτογράφηση βασισμένη στο AES (Advanced Encryption Standard) για την προστασία των προγραμμάτων και των δεδομένων. Επίσης μπορεί να έχει πρόσβαση στις παραπάνω μνήμες μέσω υψηλών ταχυτήτων caches (κρυφές μνήμες).

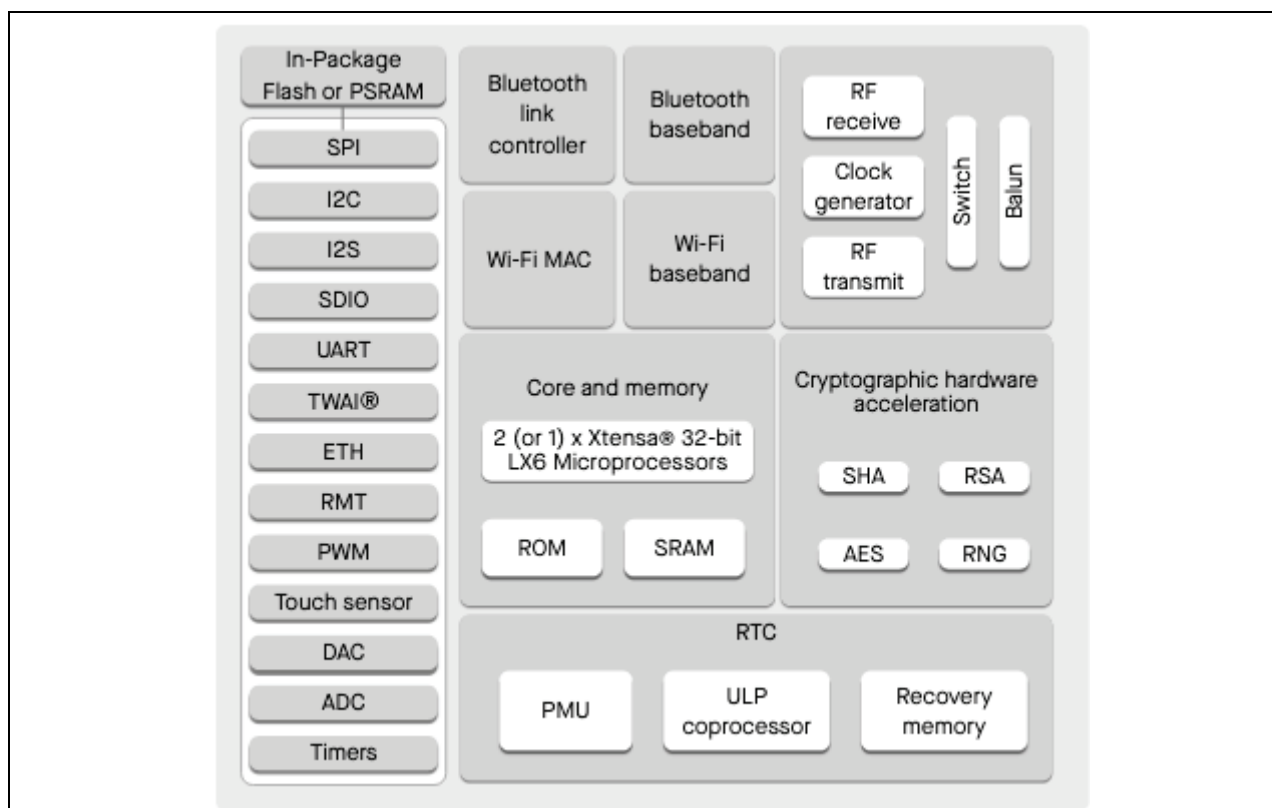
Περιφερειακά είσοδοι/έξοδοι:

- Πλούσια διεπαφή περιφερειακών με DMA (Direct Memory Access) που περιέχει capacitive touch, ADCs, DACs, I2C, UART, CAN 2.0, SPI, I2S, RMII και PWM.

Ασφάλεια :

- Υποστηρίζει όλα τα χαρακτηριστικά IEEE 802.11 συμπεριλαμβανομένου WPA, WPA/WPA2, WAPI, Secure boot, Κρυπτογράφηση της μνήμης flash, 1024bit- OTP, για τους πελάτες έως και 768-bit, Επιταχυντή υλικού κρυπτογράφησης: AES, SHA-2, RSA, ECC και RNG (The Internet of Things With ESP32, n.d.).

Το λειτουργικό μπλοκ διάγραμμα του SoC παρουσιάζεται παρακάτω.



Εικόνα 3-2 Λειτουργικό μπλοκ διάγραμμα ESP32

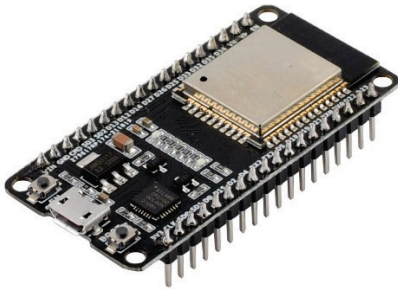
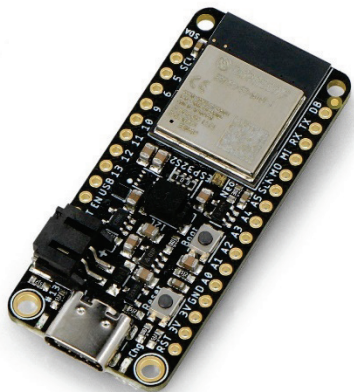
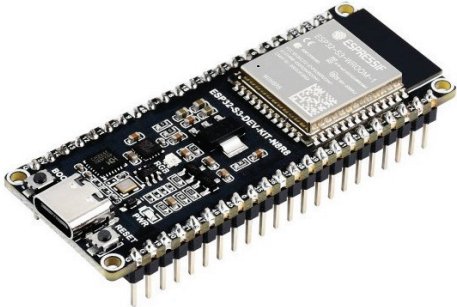
(The Internet of Things with ESP32, n.d.)

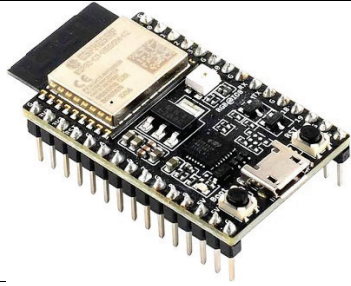
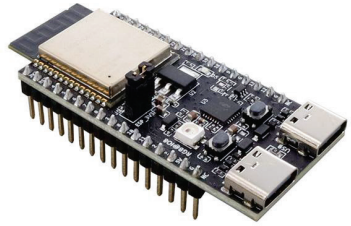
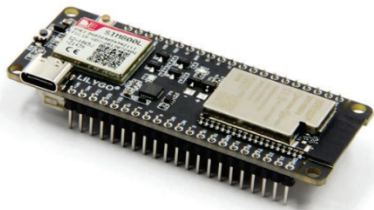
3.3 ΒΑΣΙΚΕΣ ΕΚΔΟΣΕΙΣ ESP32

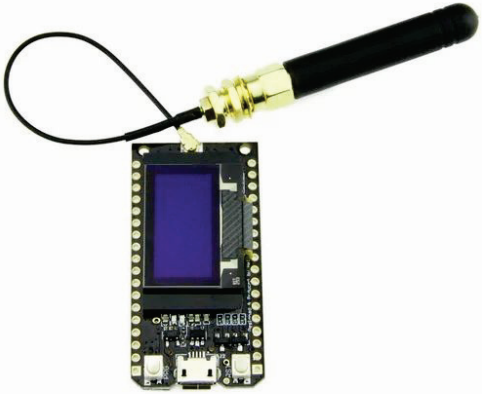
Υπάρχουν διάφορες εκδόσεις ESP32, οι πιο συνηθισμένες σειρές είναι οι ESP32-S, ESP32-C, ESP32-H, ESP32, ESP8266 και διάφορα IoT Devkits. Τα τελευταία χρόνια έχουν βγει πάρα πολλές παραλλαγές, από διάφορους κατασκευαστές.

Παρακάτω παρουσιάζονται συνοπτικά τα βασικά χαρακτηριστικά της κάθε σειράς.

Τύπος	Χαρακτηριστικά	Πλακέτα
-------	----------------	---------

ESP8266	Μονάδα μικροελεγκτή: Tensilica L106 32-bit (έως 160MHz), SRAM: 160KB, Cache: 32KB, Wi-Fi: έως 72.2 Mbps.	
ESP32	Μονάδα μικροελεγκτή: Tensilica Xtensa 32-bit LX6 (έως 240MHz) (προαιρετικά διπύρηνος), SRAM: 520KB, ROM: 448KB, Cache: 64KB, Wi-Fi 4.0, Bluetooth.	
ESP32-S2	Μονάδα μικροελεγκτή: Tensilica Xtensa 32-bit LX7 (έως 240MHz), SRAM: 320KB, ROM: 128KB, Cache: 8/16KB, Wi-Fi 4.0.	
ESP32-S3	Μονάδα μικροελεγκτή: Tensilica Xtensa 32-bit LX7 (έως 240MHz), SRAM: 512KB, ROM: 384KB, Cache: 16KB, Wi-Fi 4.0, Bluetooth 5.0.	

ESP32-C3	Μονάδα μικροελεγκτή: RISC-V 32-bit (έως 160MHz), SRAM: 400KB, ROM: 384KB, Cache: 16KB, Wi-Fi 4.0, Bluetooth 5.0.	
ESP32-C6	Μονάδα μικροελεγκτή: RISC-V 32-bit (έως 160MHz), SRAM: 400KB, ROM: 384KB, Cache: 16KB, Wi-Fi 6.0, Bluetooth 5.0.	
ESP32 T-CALL	Μονάδα μικροελεγκτή (MCU): ESP32, ένα σύστημα χαμηλού κόστους και χαμηλής κατανάλωσης σε τσιπ (SoC) με ενσωματωμένο Wi-Fi και Bluetooth 2 διπλής λειτουργίας, GSM/GPRS: Χρησιμοποιεί μια μονάδα SIM800H, Ταχύτητα ρολογιού: 240 MHz, Μνήμη: Flash 4MB/8MB και PSRAM 8MB, Ασύρματη συνδεσιμότητα: Wi-Fi 802.11 b/g/n και Bluetooth v4.2 BR/EDR και BLE, Περιφερειακές διεπαφές: I2C, SPI, UART, I2S και CAN, Τροφοδοτικό: USB Type-C και Li-Po μπαταρία διπλής τροφοδοσίας.	
ESP32 CAM	Επεξεργαστής: Μικροεπεξεργαστής Xtensa διπλού πυρήνα 32-bit LX6, 240 MHz, 600 DMIPS, Μνήμη: 520 KB SRAM, 4MB εξωτερική PSRAM και 4MB εσωτερική μνήμη flash, Ασύρματη συνδεσιμότητα: Wi-Fi 802.11 b/g/n και Bluetooth v4.2 BR/EDR και BLE, Κάμερα: Αισθητήρας 2 Megapixel OV2640 με μέγεθος πίνακα UXGA 1622x1200 και μορφές εξόδου συμπεριλαμβανομένων των YUV422, YUV420, RGB565, RGB555 και συμπιεσμένων δεδομένων 8 bit. Ο ρυθμός μεταφοράς εικόνας είναι 15 έως 60 fps 1, Ασφάλεια: Υποστηρίζει τυπικές λειτουργίες ασφαλείας IEEE 802.11, όπως WFA, WPA/WPA2, WAPI, Διαχείριση ενέργειας:	

ESP32 LoRa32 SX1276 OLED	Επεξεργαστής: Είναι υλοποιημένο γύρω από την πλατφόρμα ESP32 που παρέχει συνδεσιμότητα Wi-Fi και Bluetooth μαζί με υπολογιστικές δυνατότητες γενικής χρήσης, LoRa Chip: Η μονάδα περιλαμβάνει ένα τσιπ πομποδέκτη SX1276 LoRa, που επιτρέπει ασύρματη επικοινωνία μεγάλης εμβέλειας και χαμηλής κατανάλωσης στις ζώνες συχνοτήτων 868/915 MHz, Οθόνη: Οθόνη OLED 0,96 ιντσών για οπτική ανατροφοδότηση, Κεραίες: Η πλακέτα διαθέτει κεραίες Wi-Fi 3D και βάση IPEX για κεραία LoRa.	
-----------------------------------	---	--

Πίνακας 3-1 Εκδόσεις ESP32

(Chip Series Comparison - ESP32 - ESP-IDF Programming Guide v4.4 Documentation, n.d., TTGO T-Call v1.3, n.d., Camera Module Based on ESP32 With ESP32-CAM-MB Adapter, n.d., TTGO LORA32 868/915Mhz SX1276 ESP32 OLED-Display Bluetooth WIFI LORA, n.d.)

3.4 ΛΟΓΙΣΜΙΚΟ (SOFTWARE)

3.4.1 Γενικά

Για να προγραμματιστεί το ESP32 χρειάζεται το περιβάλλον προγραμματισμού Arduino IDE. Στο περιβάλλον αυτό γράφεται ο κώδικας (ο οποίος βασίζεται στη γλώσσα C/C++), ο οποίος μετά μεταγλωττίζεται και μεταφορτώνεται στη πλακέτα. Δείτε για περισσότερες λεπτομέρειες την ενότητα 2.4.

ΚΕΦΑΛΑΙΟ 4

ΥΛΙΚΟ ΜΕΡΟΣ ΣΥΣΤΗΜΑΤΟΣ

4.1 ΑΙΣΘΗΤΗΡΕΣ, ΕΝΕΡΓΟΠΟΙΗΤΕΣ

4.1.1 Γενικά

Σε αυτό το κεφάλαιο θα παρουσιαστούν οι βασικοί αισθητήρες, ενεργοποιητές που μπορούν να χρησιμοποιηθούν σε βασικά σενάρια αυτοματισμού και ελέγχου σε κτιριακό περιβάλλον βασισμένα στις πλατφόρμες Arduino και ESP32. Παράλληλα θα παρουσιαστούν και μερικά Arduino Shields.

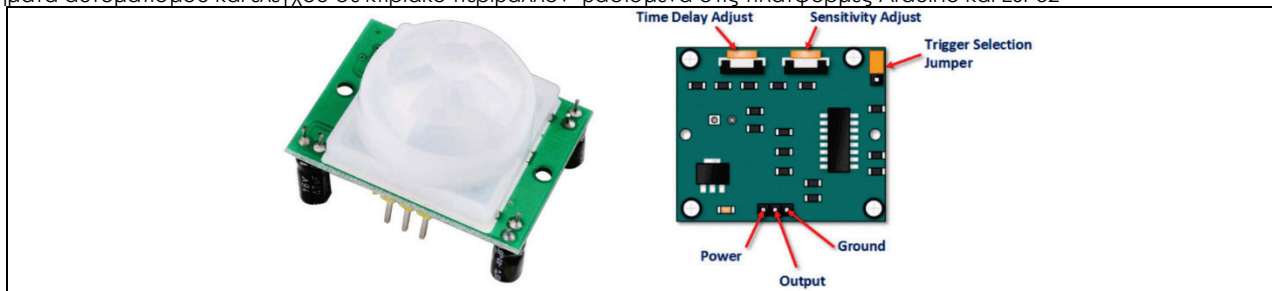
Για όλα θα παρουσιαστούν οι τεχνικές διασύνδεσης και προγραμματισμού τους.

4.1.2 Αισθητήρας κίνησης (HC-SR501 PIR)

Ο αισθητήρας κίνησης HC-SR501 PIR αποτελείται από δύο ξεχωριστά μέρη που είναι κατασκευασμένα από υλικό ευαίσθητο στην υπέρυθρη ακτινοβολία (IR). Όταν και τα δύο αυτά μέρη ανιχνεύουν την ίδια ποσότητα υπέρυθρης ακτινοβολίας στο περιβάλλον, ο αισθητήρας παραμένει αδρανής.

Όμως, όταν ένα θερμό αντικείμενο, όπως ένας άνθρωπος, περάσει μέσα στο πεδίο ανίχνευσης, η υπέρυθρη ακτινοβολία που προσπίπτει στα δύο μέρη του αισθητήρα μεταβάλλεται διαφορετικά. Αυτές οι διαφορικές μεταβολές παράγουν παλμούς που ανιχνεύονται από τον αισθητήρα και ενεργοποιούν την έξοδό του.

Στον αισθητήρα υπάρχει ένα κύκλωμα ρύθμισης της ευαισθησίας, το οποίο μπορεί να ρυθμιστεί από 3 έως 7 μέτρα. Επίσης, υπάρχει ρύθμιση της χρονικής διάρκειας παραμονής της εξόδου ενεργής μετά την ανίχνευση, από 3 δευτερόλεπτα έως 5 λεπτά. Ο αισθητήρας έχει γωνία παρακολούθησης έως 110 μοίρες και διαθέτει επιλογέα (jumpers) για τον τρόπο ενεργοποίησής του.



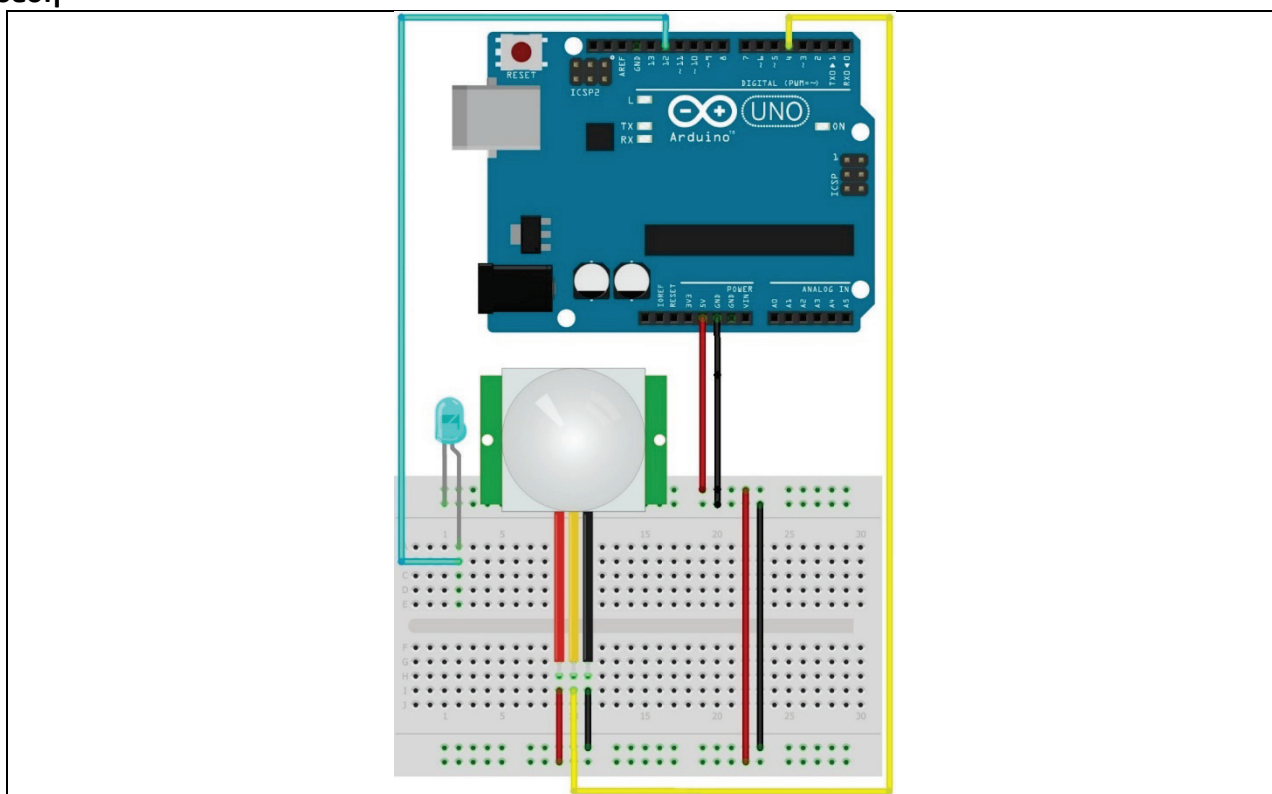
Εικόνα 4-1 Αισθητήρας κίνησης (HC-SR501 PIR)

(The most complete starterkit, n.d.)

Τεχνικά χαρακτηριστικά:

- Τάση λειτουργίας: DC 2.7-12V.
- Στατική κατανάλωση ενέργειας: <0,1mA.
- Χρόνος καθυστέρησης: 2 δευτερόλεπτα.
- Ο χρόνος αποκλεισμού: 2 δευτερόλεπτα.
- Trigger: μπορεί να επαναληφθεί.
- Εύρος ανίχνευσης: $\leq 100^\circ$ γωνία κώνου, 3-5 μέτρα. (απαιτείται ανάλογα με τον φακό).
- Θερμοκρασία λειτουργίας: $-20 - +60^\circ\text{C}$.
- Διαστάσεις PCB: 10mm x 8mm.
- Συνολικό μέγεθος: Περίπου. 12 mm χ 25 mm.

Σύνδεση



Εικόνα 4-2 Αισθητήρας κίνησης (HC-SR501 PIR)

Πρόγραμμα κώδικα

Αυτός ο κώδικας παρακολουθεί μόνο εάν η είσοδος στον ακροδέκτη 2 είναι υψηλή ή χαμηλή. Επίσης παρακολουθεί την κατάσταση του pin, έτσι ώστε να εκτυπώνει ένα μήνυμα όταν η κίνηση ξεκινά και σταματά.

```
int ledPin = 12; //LED pin
int inputPin = 4; //PIR sensor pin
int val = 0;
void setup() { pinMode(ledPin, OUTPUT); //ορισμός του LED ως έξοδο
pinMode(inputPin, INPUT); //ορισμός του PIR sensor ως είσοδο
Serial.begin(9600); }
void loop(){ val = digitalRead(inputPin); //διάβασμα εισόδου
if (val == HIGH) { //είναι high?
digitalWrite(ledPin, HIGH); //άναψε το LED
Serial.println("Motion");
}
else { digitalWrite(ledPin, LOW); // σβήσε το LED
Serial.println("No Motion");
}
```

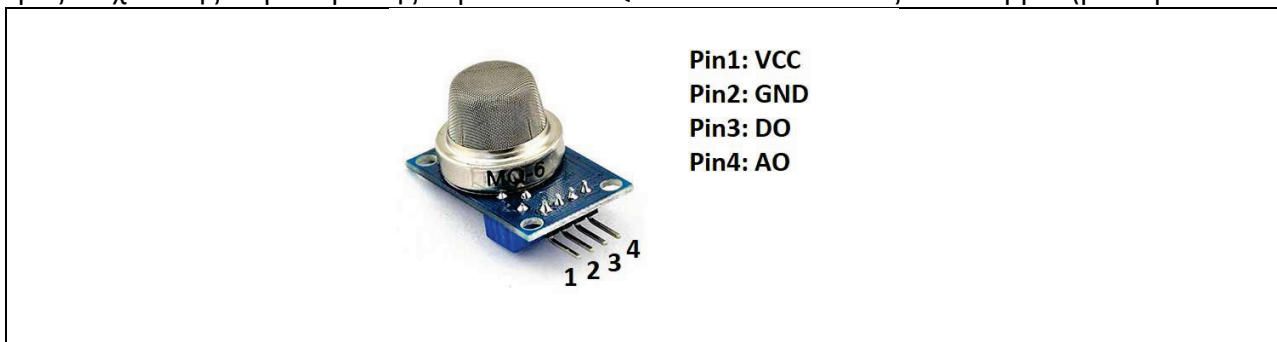
Κώδικας 4-1 Κώδικας αισθητήρα ανίχνευσης κίνησης

4.1.3 Αισθητήρας μέτρησης αερίων (MQ-6 Gas Sensor Module)

Ο αισθητήρας αερίου MQ-6 χρησιμοποιείται για τη μέτρηση της περιεκτικότητας αερίων στο περιβάλλον. Παρέχει υψηλή ευαισθησία σε υγραέριο, βουτάνιο, προπάνιο και φυσικό αέριο, ενώ μπορεί να ανιχνεύσει με μικρότερη αποτελεσματικότητα αλκοόλ και καπνό.

Η λειτουργία του βασίζεται σε έναν αισθητήρα που περιέχει οξείδιο του κασσίτερου (SnO_2). Το SnO_2 έχει χαμηλή αγωγιμότητα στον καθαρό αέρα. Όταν η συγκέντρωση των αερίων μεταβάλλεται, τότε μεταβάλλεται και η ηλεκτρική αγωγιμότητα του αισθητήρα. Αυτή η μεταβολή μετατρέπεται σε αντίστοιχο σήμα εξόδου.

Το σήμα αυτό τροφοδοτείται σε έναν LM393, ο οποίος το ψηφιοποιεί και το καθιστά διαθέσιμο σε μια ακίδα ψηφιακής εξόδου. Μέσω ενός ποτενσιόμετρου ρυθμίζεται το όριο της επιθυμητής συγκέντρωσης. Το εύρος ανίχνευσης συγκέντρωσης αερίων του MQ-6 είναι από 200 έως 10.000 ppm (parts per million).



Εικόνα 4-3 Αισθητήρας αερίων (MQ-6 Gas Sensor Module)

(The most complete starterkit, n.d.)

Υπάρχουν πολλές διαφορετικές εκδόσεις αισθητήρων μέτρησης αερίων, οι κυριότεροι παρουσιάζονται στην παρακάτω εικόνα.

 MQ-2	 MQ-3	 MQ-4	MQ-2: Αισθητήρας καπνού MQ-3: Αισθητήρας αλκοόλ MQ-4: Αισθητήρας μεθανίου MQ-5: Αισθητήρας φυσικού αερίου LPG Αισθητήρας αερίου πόλης MQ-6: Αισθητήρας προπανίου ισοβουτανίου MQ-7: Μονάδα αισθητήρα μονοξειδίου του άνθρακα MQ-8: αισθητήρας υδρογόνου MQ-9: Αισθητήρας καύσιμου αερίου μονοξειδίου του άνθρακα MQ-135: αισθητήρας ανίχνευσης ποιότητας αέρα
 MQ-5	 MQ-6	 MQ-7	
 MQ-8	 MQ-9	 MQ-135	

Εικόνα 4-4 Αισθητήρες αερίων

(The most complete Gas Kit, n.d.)

Τεχνικά χαρακτηριστικά:

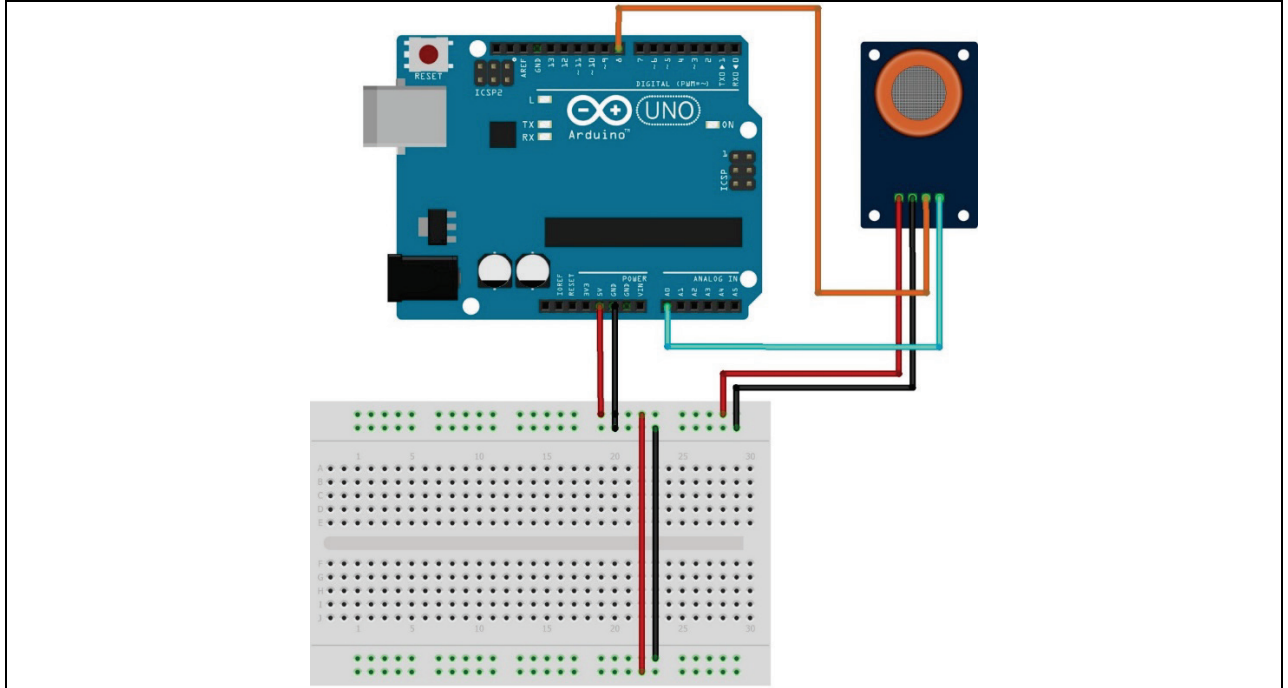
- Ένδειξη ισχύος και ένδειξη σήματος εξόδου TTL.
- Έχει έξοδο σήματος διακόπτη DO (TTL) και έξοδο αναλογικού σήματος AO.
- Όσο μεγαλύτερη είναι η συγκέντρωση αερίου, τόσο μεγαλύτερη είναι η τάση της αναλογικής εξόδου.
- Έχει καλή ευαισθησία στην ανίχνευση αερίων και φυσικού αερίου.
- Διαθέτει τέσσερις σπές βιδών για εύκολη τοποθέτηση.
- Διαστάσεις: 32 (L) * 20 (W) * 22 (H)

Ηλεκτρική απόδοση:

- Ανίχνευση αερίου: υγραέριο, βουτάνιο, προπάνιο και φυσικό αέριο, ενώ μπορεί να ανιχνεύσει με μικρότερη αποτελεσματικότητα αλκοόλ και καπνό.
- Ανίχνευση συγκέντρωσης: 10-1000ppm.
- Τάση εισόδου: DC 5V.
- Κατανάλωση ρεύματος: 150mA.
- DO έξοδος: TTL ψηφιακές τιμές 0 και 1 (0,1 και 5V).
- AO έξοδος: 0,1-0,3V (σχετικά μη ρυπογόνο), η υψηλότερη τάση συγκέντρωσης είναι περίπου 4V.
- Τσιπ: AT89S52.
- Κρύσταλος: 11.0592MHZ.
- Ρυθμός μετάδοσης(Baud) : 9600.

Σύνδεση

Το MQ έχει 4 ακίδες, δηλαδή VCC, Ground, Digital Out και Analog Out. VCC και GND χρησιμοποιούνται για την ενεργοποίηση της μονάδας αισθητήρα. Ψηφιακή έξοδος που χρησιμοποιείται για την απόδοση της έντασης Υψηλή / Χαμηλή όταν η συγκέντρωση του αερίου φτάσει στο κατώφλι. Έτσι θα δώσει μόνο ψηφιακό σήμα. Το Analog Out παρέχει αναλογική τάση ανάλογα με τη συγκέντρωση αερίου. Μεγαλύτερη συγκέντρωση θα δώσει περισσότερη έξοδο τάσης. Δεδομένου ότι θα διαβάσουμε τη συγκέντρωση αερίου, χρησιμοποιούμε την αναλογική έξοδο.



Εικόνα 4-5 Σύνδεση αισθητήρα αερίων (MQ-6 Gas Sensor Module)

```
void setup() {  
  Serial.begin(9600); //αρχικοποίηση σειριακής επικοινωνίας στα 9600 bits ανά δευτερόλε-  
πτο  
}  
void loop() {  
  int sensorValue = analogRead(A0); //διαβάζει την είσοδο analog pin 0  
  Serial.println(sensorValue); //εκτυπώνει την τιμή που διάβασε  
  delay(1000);  
}
```

Κώδικας 4-2 Κώδικας αισθητήρα μέτρησης αερίων (MQ-6 Gas Sensor Module)

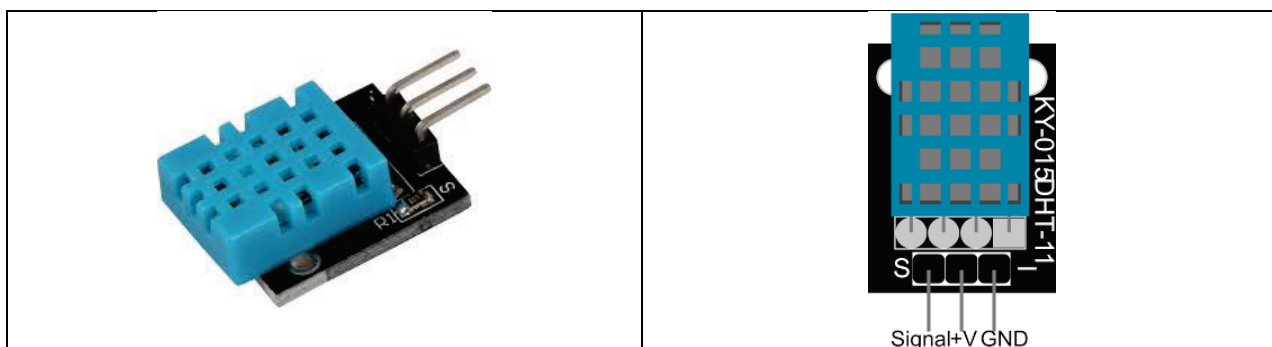
4.1.4 Αισθητήρας Θερμοκρασίας-Υγρασίας DHT-11 (KY-015 Temperature and Humidity Sensor Module)

Ο αισθητήρας DHT11 είναι ένας ολοκληρωμένος αισθητήρας που μπορεί να μετρήσει τόσο τη θερμοκρασία όσο και την υγρασία. Αυτό το επιτυγχάνει παράγοντας ένα βαθμονομημένο ψηφιακό σήμα στην έξοδο του. Χρησιμοποιώντας μια αποκλειστική ψηφιακή τεχνική και τεχνολογία ανίχνευσης θερμοκρασίας

και υγρασίας, ο αισθητήρας διασφαλίζει υψηλή αξιοπιστία και εξαιρετικά σταθερή απόδοση μακροπρόθεσμα.

Ο αισθητήρας περιλαμβάνει ένα ωμικού τύπου στοιχείο για την μέτρηση της υγρασίας και ένα NTC στοιχείο για τη μέτρηση της θερμοκρασίας. Αυτά συνδέονται με έναν υψηλής απόδοσης 8-bit μικροελεγκτή, προσφέροντας άριστη ποιότητα, γρήγορη απόκριση, ανθεκτικότητα σε παρεμβολές και καλή σχέση κόστους-αποτελεσματικότητας.

Αυτός ο αισθητήρας παρέχει ψηφιακή έξοδο ανάλογη με τη θερμοκρασία και την υγρασία που μετρά. Η τεχνολογία που χρησιμοποιείται διασφαλίζει την αξιοπιστία, τη σταθερότητα μακροπρόθεσμα και τον γρήγορο χρόνο απόκρισης. Κάθε στοιχείο του αισθητήρα έχει βαθμονομηθεί με ακρίβεια στο εργαστήριο και ο συντελεστής βαθμονόμησης αποθηκεύεται σε εσωτερική μνήμη, χρησιμοποιούμενος από την ενσωματωμένη διαδικασία ανίχνευσης σήματος. Η σειριακή διασύνδεση καθιστά την ενσωμάτωση του αισθητήρα σε ψηφιακά συστήματα εύκολη και γρήγορη. Το μικρό μέγεθος και το ελαφρύ βάρος του τον καθιστούν ιδανικό για ρομποτικές και περιβαλλοντικές εφαρμογές παρακολούθησης.



Εικόνα 4-6 Αισθητήρας θερμοκρασίας-υγρασίας DHT-11

(The most complete starterkit, n.d.)

Τεχνικά χαρακτηριστικά:

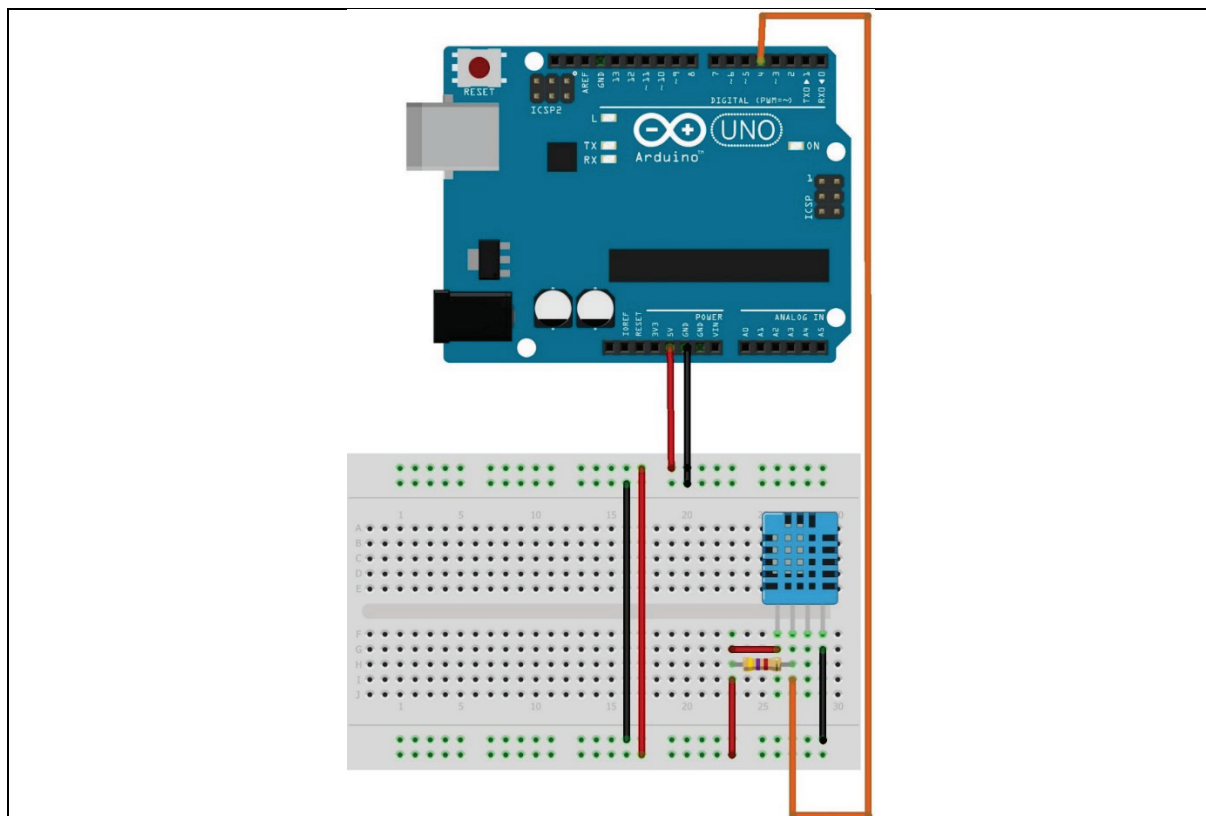
- Μοντέλο αισθητήρα: DHT11.
- Εύρος μετάδοσης σήματος: 20 μέτρα.
- Εύρος μέτρησης υγρασίας: 20 - 95% RH.
- Σφάλμα μέτρησης υγρασίας: +/- 5%.
- Εύρος μέτρησης θερμοκρασίας: 0 - 50 ° C.
- Σφάλμα μέτρησης θερμοκρασίας: + / -2 ° C.
- Τάση λειτουργίας: 3.3 - 5V.
- Τύπος εξόδου: ψηφιακή έξοδος.
- Μικρές πλάκες PCB μέγεθος: 1.05" x 0.7" (connectors excluded).
- Καθαρό βάρος: 2.7g.

Σύνδεση με το Arduino

Η φυσική διασύνδεση του αισθητήρα γίνεται μέσω ενός βύσματος τριών pin τα οποία είναι +5Volt, GND και DATA(δεδομένα). Τα πρώτα δύο pin αποτελούν την παροχή τάσης τροφοδοσίας και γείωσης του αισθητήρα και το τρίτο pin είναι η ψηφιακή έξοδος του αισθητήρα.

DATA	Serial data output
GRD	Ground
+5V	Power supply(+5Volt)

Πίνακας 4-1 PINS OUT



Εικόνα 4-7 Σύνδεση αισθητήρα θερμοκρασίας-υγρασίας DHT-11

Πρόγραμμα κώδικα

Ο παρακάτω κώδικας χρησιμοποιεί τον ακροδέκτη (pin) 4 του Arduino για τη σειριακή αποστολή και λήψη δεδομένων από τον αισθητήρα KY-015. Η σειριακή επικοινωνία αρχιεοθετείται αποστέλλοντας συγκεκριμένα σήματα υψηλής / χαμηλής στάθμης στον αισθητήρα και περιμένοντας την απάντηση. Τα δεδομένα θερμοκρασίας και υγρασίας διαβάζονται ανά δυαδικά ψηφία και επιστρέφονται ως μια σειρά από byte.

```
#include <dht11.h>
#define DHT11PIN 4
dht11 DHT11;
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  Serial.println();
  int chk = DHT11.read(DHT11PIN);
  Serial.print("Humidity (%): ");
  Serial.println((float)DHT11.humidity, 2);
  Serial.print("Temperature (C): ");
  Serial.println((float)DHT11.temperature, 2);
}
```

```
delay(2000);  
}
```

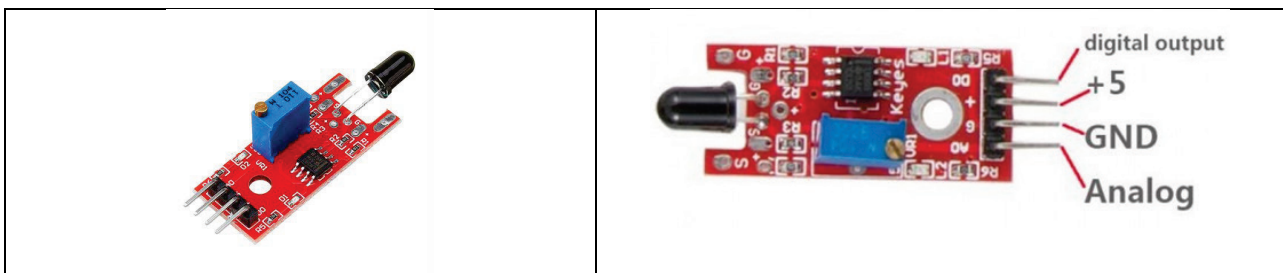
Κώδικας 4-3 Κώδικας αισθητήρα Θερμοκρασίας-Υγρασίας DHT-11 (KY-015 Temperature and Humidity Sensor Module)

4.1.5 Αισθητήρας φωτιάς, KY-026 Flame Sensor Module IR Sensor Detector

Ο αισθητήρας ανίχνευσης φωτιάς KY-026 βασίζεται σε έναν υπέρυθρο δέκτη LED 5 mm, ο οποίος είναι ευαίσθητος στο φασματικό εύρος φωτός που δημιουργείται από μια φλόγα. Επίσης περιλαμβάνει ένα διπλό διαφορικό LM393 συγκριτή, ένα ποτενσιόμετρο για τον ορισμό του σημείου ενεργοποίησης του αισθητήρα και δύο ενδεικτικές λυχνίες LED.

Ο αισθητήρας φωτιάς μπορεί να ανιχνεύσει μήκη κύματος που κυμαίνονται από 760 nm έως 1100 nm στο υπέρυθρο φάσμα. Η μονάδα παρέχει δύο εξόδους:

- Η έξοδος D0 ενεργοποιείται σε High ή Low ανάλογα με το αν η στάθμη της φλόγας υπερβαίνει ή όχι το σημείο ρύθμισης.
- Η έξοδος A0 μεταφράζει την ένταση της φλόγας που ανιχνεύεται, σε σήμα τάσης εξόδου πραγματικού χρόνου.



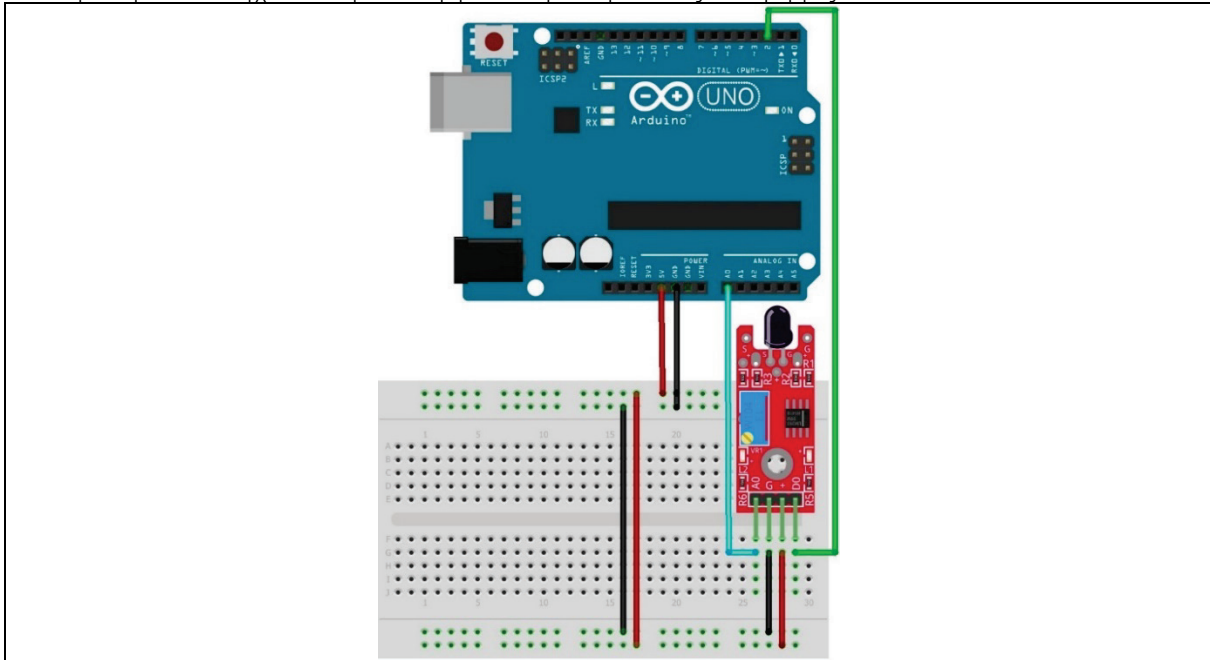
Εικόνα 4-8 Αισθητήρας φωτιάς, KY-026 Flame Sensor Module IR Sensor Detector

(The most complete starterkit, n.d.)

Τεχνικά χαρακτηριστικά:

- Δέκτης IR υψηλής ευαισθησίας.
- Εξαιρετικά ευαίσθητο σε κύματα μεταξύ 760-1100nm.
- Με λυχνία ένδειξης τροφοδοσίας ρεύματος και λυχνία ένδειξης εξόδου συγκριτικού.
- A0, σήμα εξόδου τάσης θερμίστορ σε πραγματικό χρόνο και έξοδος σήματος υψηλής / χαμηλής ηλεκτρικής στάθμης. Παραγωγή αναλογικής ποσότητας.
- Έξοδος ηλεκτρικής στάθμης κατωφλίου (ρυθμίζεται από το ποτενσιόμετρο).
- Εύρος ανίχνευσης: περίπου 60 μοίρες.
- Μέγεθος (L x W): 36 x 16 χιλιοστά.
- Τροφοδοσία: 0-15 V DC.

Σύνδεση με το Arduino



Εικόνα 4-9 Σύνδεση αισθητήρα φωτιάς, KY-026 Flame Sensor Module IR Sensor Detector

Πρόγραμμα κώδικα

Σε αυτό τον κώδικα Arduino θα διαβάσουμε τις τιμές τόσο από ψηφιακές, όσο και από αναλογικές διεπαφές στο KY-026. Μπορείτε να χρησιμοποιήσετε έναν αναπτήρα για να αλληλοεπιδράσετε με τη μονάδα ανιχνευτή φλόγας. Η ψηφιακή διεπαφή θα στείλει ένα σήμα HIGH όταν ανιχνεύεται φλόγα από τον αισθητήρα, ενεργοποιώντας τη λυχνία LED του Arduino (pin 13). Μπορείτε να περιστρέψτε το ποτενσιόμετρο δεξιόστροφα για να αυξήσετε το κατώφλι ανίχνευσης και αριστερόστροφα για να το μειώσετε. Η αναλογική διεπαφή επιστρέφει υψηλή αριθμητική τιμή όταν δεν υπάρχει φλόγα κοντά και θα πέσει σχεδόν στο μηδέν σε περίπτωση πυρκαγιάς.

```
int led = 13; //ορισμός LED pin
int digitalPin = 2; //ορισμός KY-026 digital
int analogPin = A0; //ορισμός KY-026 analog
int digitalValue;
int analogValue;
void setup()
{
  pinMode(led, OUTPUT);
  pinMode(digitalPin, INPUT);

  Serial.begin(9600); Εκκίνηση Serial Port
}
void loop()
{
  digitalValue = digitalRead(digitalPin);
  if(digitalValue == HIGH) //Εάν έχει ανιχνευτεί φωτιά/φλόγα
  {
    digitalWrite(led, HIGH); //άνοιγμα του LED
  }
  else
  {
    digitalWrite(led, LOW); //σβήσιμο του LED
  }
}
```

```

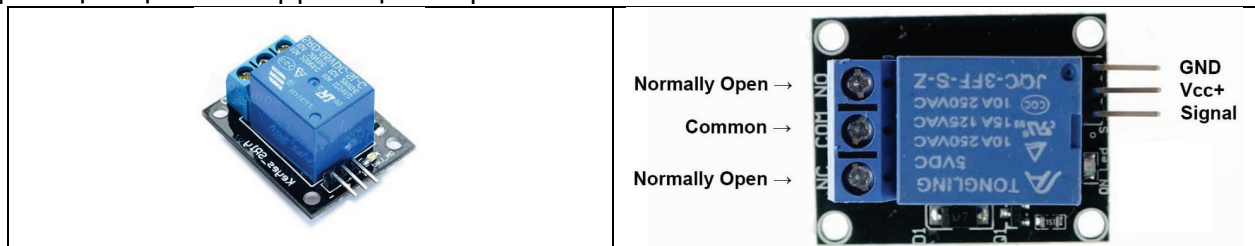
}
analogValue = analogRead(analogPin);
Serial.println(analogValue); //Εκτύπωσε την αναλογική τιμή
delay(1000);
}

```

Κώδικας 4-4 Κώδικας αισθητήρα φωτιάς, KY-026 Flame Sensor Module IR Sensor Detector

4.1.6 Ρελέ, 5V Relay TTL Signal 1 Channel Module High Level Expansion Board For Arduino (KY-019)

Η μονάδα ρελέ 5V KY-019 είναι ένας ηλεκτρομαγνητικός διακόπτης που μπορεί να ελεγχθεί από έναν μικροελεγκτή χαμηλής τάσης, όπως το Arduino. Συνήθως έχει τρεις ακίδες: VCC (παροχή ρεύματος), GND (γείωση) και SIG (σήμα). Ο ακροδέκτης VCC χρησιμοποιείται για την παροχή ρεύματος στη μονάδα (συνήθως 5V). Ο ακροδέκτης GND συνδέεται με τη γείωση του Arduino. Ο ακροδέκτης SIG χρησιμοποιείται για τον έλεγχο του διακόπτη του ρελέ και απαιτεί ένα ψηφιακό σήμα (HIGH ή LOW) από το Arduino για να ενεργοποιήσει ή να απενεργοποιήσει το ρελέ.



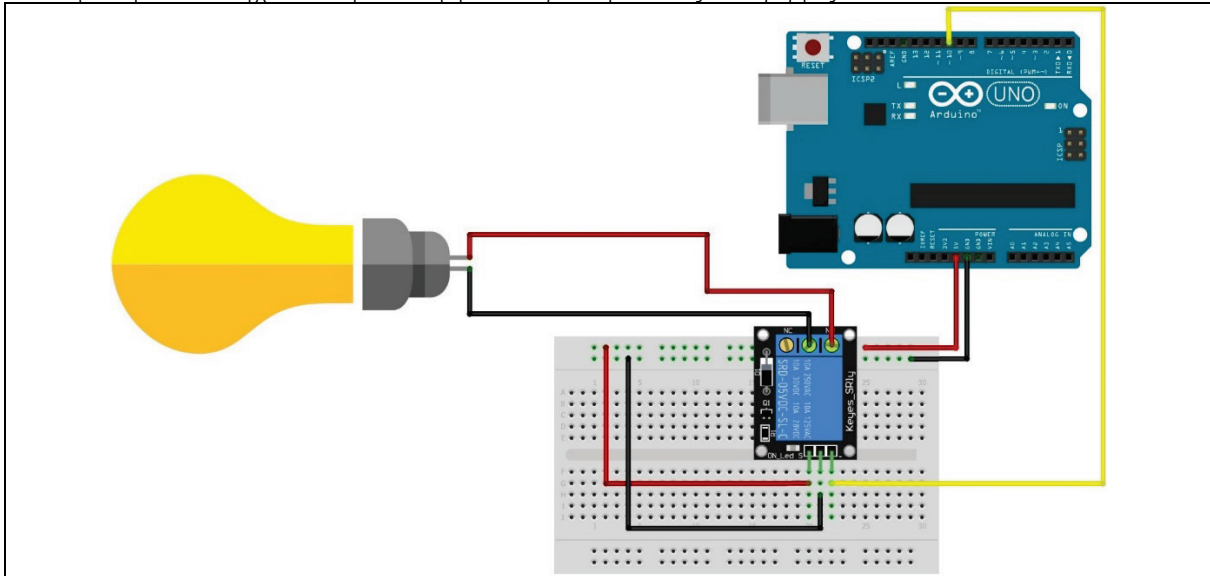
Εικόνα 4-10 Πλακέτα Ρελέ, 5V Relay TTL Signal 1 Channel Module High Level Expansion Board For Arduino (KY-019)

(The most complete starterkit, n.d.)

Τεχνικά χαρακτηριστικά:

- 5V μονάδα ρελέ.
- Μπορεί να χρησιμοποιηθεί για την ανάπτυξη μικροελεγκτών.
- μονάδα ελέγχου ή έλεγχο οικιακών συσκευών.
- 5V - 12V σήμα ελέγχου TTL.
- Μπορεί να ελέγξει σήμα DC ή AC, επίσης φορτίο 220V AC.
- Κανονικά ανοικτή / κλειστή επαφή.
- Ένδειξη λειτουργίας.
- Οι ακροδέκτες ελέγχου έχουν pull down κύκλωμα για την αποφυγή δυσλειτουργίας του ρελέ.

Σύνδεση με το Arduino



Εικόνα 4-11 Σύνδεση πλακέτας Ρελέ, 5V Relay TTL Signal 1 Channel Module High Level Expansion Board For Arduino (KY-019)

Πρόγραμμα κώδικα

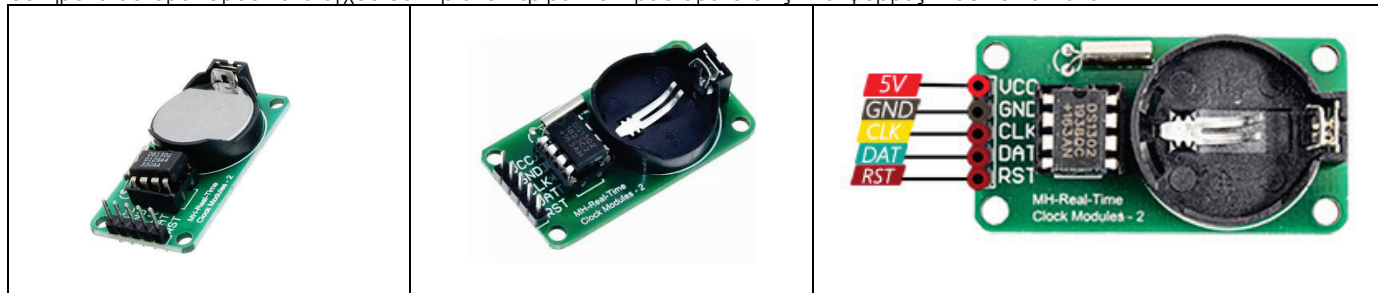
Ο παρακάτω κώδικας Arduino θα ενεργοποιήσει / απενεργοποιήσει το ρελέ κάθε δευτερόλεπτο. Στο ρελέ συνδέεται μία λάμπα χρησιμοποιώντας τη σύνδεση NO (κανονικά ανοιχτή), έτσι ώστε η λάμπα να είναι σβηστή όταν το ρελέ είναι απενεργοποιημένο.

```
int relay = 10; //relay Pin 10
void setup()
{
  pinMode(relay,OUTPUT);
}
void loop()
{
  digitalWrite(relay,HIGH); //άνοιξε το relay (ON)
  delay(1000);
  digitalWrite(relay,LOW); //κλείσε το relay (OFF)
  delay(1000);
}
```

Κώδικας 4-5 Κώδικας Ρελέ, 5V Relay TTL Signal 1 Channel Module High Level Expansion Board For Arduino (KY-019)

4.1.7 Ρολόι πραγματικού χρόνου RTC (Real Time Clock) DS1302

Τα ρολόγια πραγματικού χρόνου έχουν τη δυνατότητα να υπολογίζουν δευτερόλεπτα, λεπτά, ώρες, ημέρες, εβδομάδες, μήνες και χρόνια πριν από το έτος 2100, καθώς και τη δυνατότητα προσαρμογής των δίσεκτων χρόνων. Είναι μία ηλεκτρονική μονάδα ρολογιού πραγματικού χρόνου με εφεδρική μπαταρία χρησιμοποιώντας το εύκολο στη χρήση τσιπ DS1302. Το τσιπ DS1302 χρησιμοποιεί απλή σειριακή διασύνδεση.



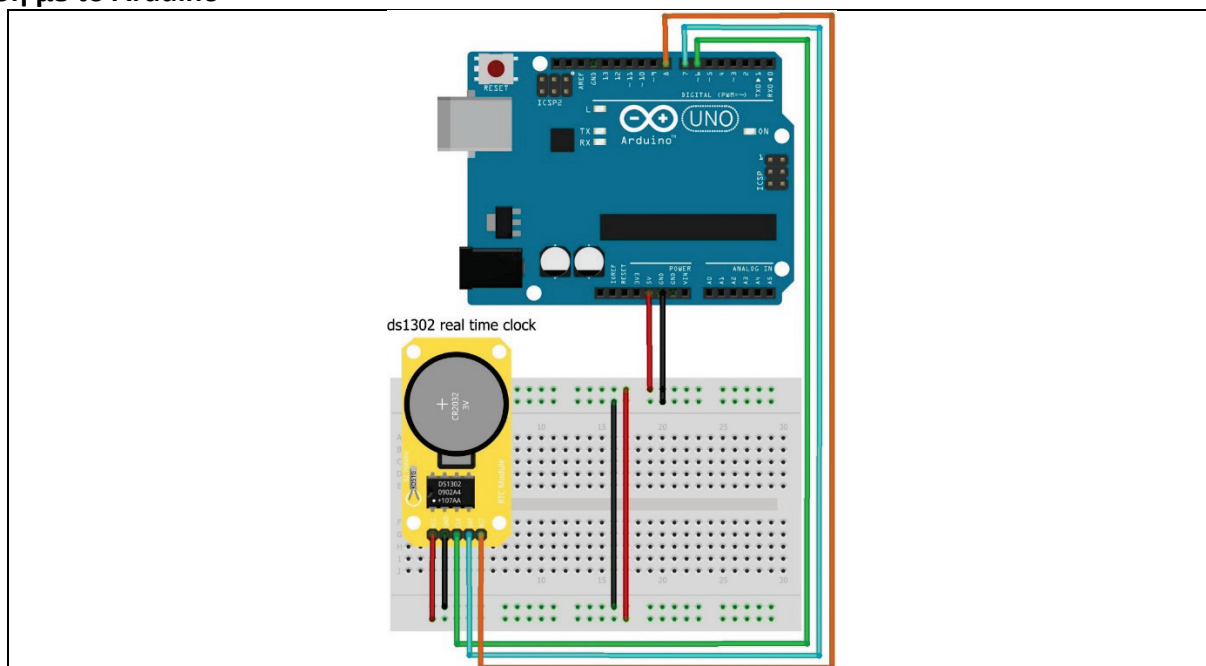
Εικόνα 4-12 Πλακέτα ρολογιού πραγματικού χρόνου RTC (Real Time Clock) DS1302

(The most complete starterkit, n.d.)

Τεχνικά χαρακτηριστικά:

- Μνήμη RAM αποθήκευσης δεδομένων 318 bit.
- Τάση λειτουργίας ευρείας κλίμακας 2- 5.5V.
- Όταν η τάση λειτουργίας είναι 2.0V, το ρεύμα είναι λιγότερο από 300nA.
- Όταν διαβάζονται / γράφονται δεδομένα ρολογιού ή RAM, υπάρχουν δύο τρόποι μετάδοσης: μετάδοση ενός byte και μεταφορά συνόλου χαρακτήρων πολλών byte.
- Η μπαταρία αναμονής είναι η CR2032, τάσης 3V, ρεύματος 260mAh, μη επαναφορτιζόμενη. Θεωρητικά τα δεδομένα διατηρούνται για περισσότερα από 10 χρόνια.
- Απλό interface 3 γραμμών.
- TTL συμβατότητα με Vcc = 5V.
- Μέγεθος PCB πάνελ: 44mm * 23mm * 1.6mm
- Ταλαντωτής κρυστάλλου 32.768KHz, 6pF, μέγεθος 2 * 6mm
- Η τάση λειτουργίας: 3.3V / 5V
- Θερμοκρασία λειτουργίας: 0 - 70 ° C

Σύνδεση με το Arduino



Εικόνα 4-13 Σύνδεση πλακέτας ρολογιού πραγματικού χρόνου RTC (Real Time Clock) DS1302

Πρόγραμμα κώδικα

Ο παρακάτω κώδικας Arduino χρησιμοποιεί το . DS1302 τυπώνοντας τα τρέχοντα seconds, minutes, hours, day of the week, day of the month, month, year.

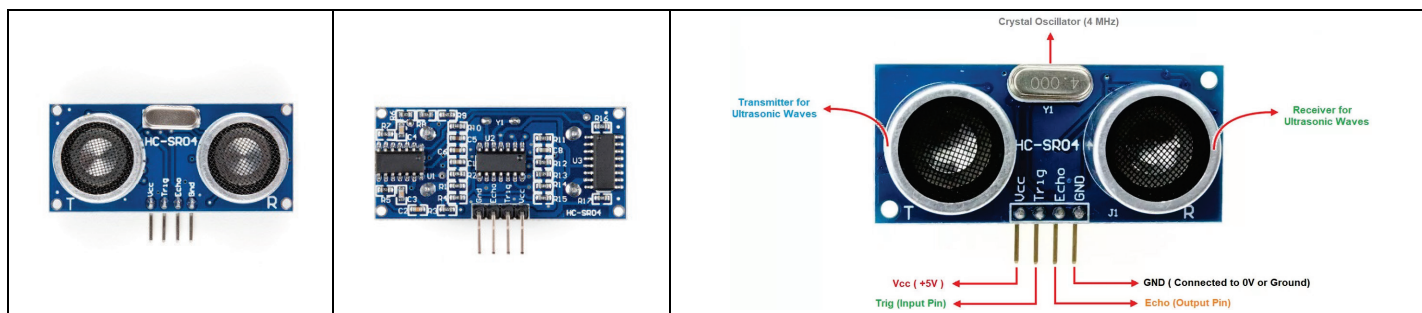
```
#include <virtuabotixRTC.h> //η βιβλιοθήκη που χρειάζεται
virtuabotixRTC myRTC(6, 7, 8); //SCLK -> 6, I/O -> 7, CE -> 8
void setup() {
  Serial.begin(9600); //Εκκίνηση Serial Port
  //Ορίζουμε την τρέχουσα ημερομηνία και ώρα στην ακόλουθη μορφή: seconds, minutes,
  hours, day of the week, day of the month, month, year
  myRTC.setDS1302Time(00, 00, 12, 1, 1, 1, 2020); //Πρέπει να μπει ως σχόλιο ή να την αφαι-
  ρέσουμε αυτήν την εντολή μόλις τελειώσουμε, η αρχικοποίηση γίνεται μόνο μία φορά και η
  μονάδα θα συνεχίσει να την μετρά αυτόματα
}
void loop() {
  myRTC.updateTime();
  Serial.print("Current Date / Time: ");
  Serial.print(myRTC.dayofmonth);
  Serial.print("/");
  Serial.print(myRTC.month);
  Serial.print("/");
  Serial.print(myRTC.year);
  Serial.print(" ");
  Serial.print(myRTC.hours);
  Serial.print(":");
  Serial.print(myRTC.minutes);
  Serial.print(":");
  Serial.println(myRTC.seconds);
  delay(1000); //Καθυστέρηση 1 sec
}
```

Κώδικας 4-6 Κώδικας ρολογιού πραγματικού χρόνου RTC (Real Time Clock) DS1302

4.1.8 Ψηφιακός αισθητήρας υπερήχων, HC-SR04 Ultrasonic Module Distance Measuring Transducer

Ο HC-SR04 είναι ένας ψηφιακός αισθητήρας υπερήχων που χρησιμοποιεί την τεχνολογία σόναρ (ηχητικά κύματα) για να ανιχνεύσει την απόσταση ενός αντικειμένου, παρόμοια με τον τρόπο που χρησιμοποιούν οι νυχτερίδες και τα δελφίνια. Προσφέρει εξαιρετική ανίχνευση με υψηλή ακρίβεια και σταθερές μετρήσεις, σε ένα εύκολο προς χρήση πακέτο. Η λειτουργία του δεν επηρεάζεται από την ηλιοφάνεια ή από την ανίχνευση μαύρων αντικειμένων. Ωστόσο, μπορεί να είναι δύσκολο να ανιχνεύσει μαλακά υλικά όπως ρούχα. Για την υλοποίηση κώδικα χρησιμοποιείται η συνάρτηση pulseIn() του Arduino, όπου υπολογίζει

Συστήματα αυτοματισμού και ελέγχου σε κτιριακό περιβάλλον βασισμένα στις πλατφόρμες Arduino και ESP32 τον χρόνο από την μία κατάσταση pin στην άλλη πχ. High -> Low. Για τον υπολογισμό της απόστασης θα χρησιμοποιηθεί η εξίσωση $distance(cm) = \frac{time(ms)}{58}$.



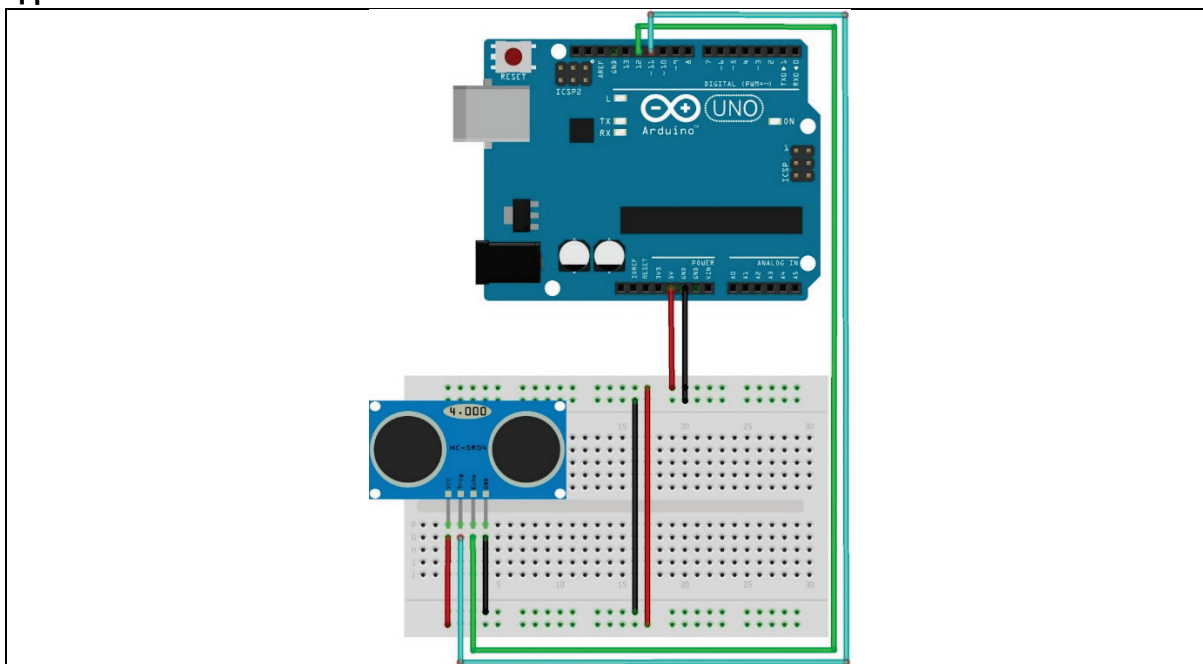
Εικόνα 4-14 Πλακέτα ψηφιακού αισθητήρα υπερήχων, HC-SR04 Ultrasonic Module Distance Measuring Transducer

(The most complete starterkit, n.d.)

Τεχνικά χαρακτηριστικά:

- Τάση λειτουργίας: 5V (DC).
- Στατικό ρεύμα: Λιγότερο από 2mA.
- Σήμα εξόδου: Ηλεκτρικό σήμα συχνότητας, υψηλό 5V, χαμηλό 0V.
- Γωνία αισθητήρα: Το πολύ 15 μοίρες.
- Απόσταση ανίχνευσης: 2cm-450cm.
- Ακρίβεια: Μέχρι 0,3 εκατοστά.
- Σήμα ενεργοποίησης εισόδου: 10us TTL impulse.
- Echo Signal: έξοδος σήματος TTL PWL.

Σύνδεση με το Arduino



Εικόνα 4-15 Σύνδεση πλακέτας ψηφιακού αισθητήρα υπερήχων, HC-SR04 Ultrasonic Module Distance Measuring Transducer

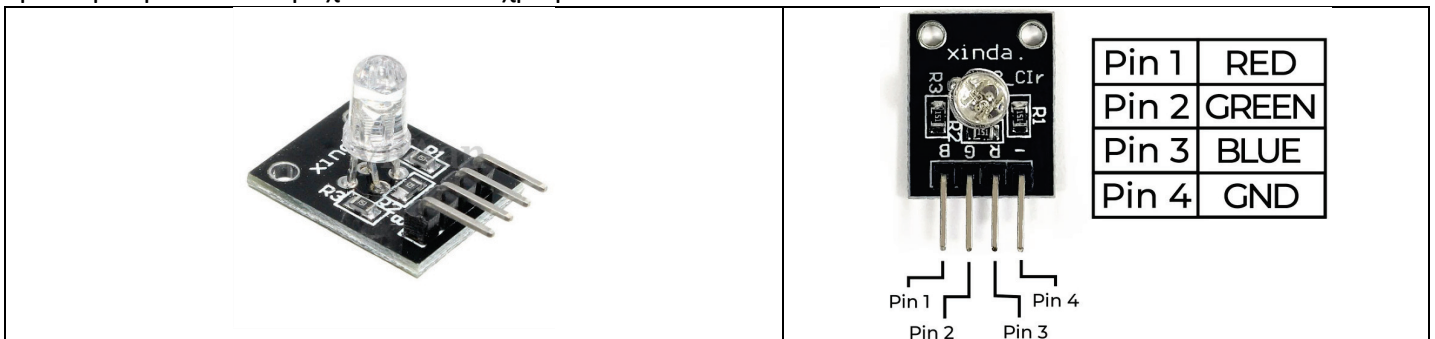
Πρόγραμμα κώδικα


```
int trigPin = 11; //Εναυσμα (Trigger)
int echoPin = 12; //Ηχώ (Echo)
long duration, cm, inches;
void setup() {
  Serial.begin(9600); //Εκκίνηση Serial Port
  pinMode(trigPin, OUTPUT); //Ορισμός εξόδου
  pinMode(echoPin, INPUT); //Ορισμός εισόδου
}
void loop() {
  digitalWrite(trigPin, LOW); //Δίνουμε έναν σύντομο LOW παλμό εκ των προτέρων για να εξα-
  σφαλίσουμε καθαρό HIGH παλμό
  delayMicroseconds(5);
  digitalWrite(trigPin, HIGH); //Ο αισθητήρας ενεργοποιείται από έναν υψηλό παλμό 10 ή πε-
  ρισσότερων microseconds.
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  pinMode(echoPin, INPUT);
  duration = pulseIn(echoPin, HIGH); // Διάβασμα σήματος αισθητήρα: έναν υψηλός παλμός
  του οποίου η διάρκεια είναι ο χρόνος, σε microseconds, από την αποστολή του ping στη λήψη
  της ηχώ από ένα αντικείμενο.
  cm = (duration/2) / 29.1; //Μετατροπή του χρόνου σε απόσταση, διαιρέστε με 29.1
  Serial.print(cm);
  Serial.print("cm");
  Serial.println();
  delay(1000); //Καθυστέρηση 1 sec
}
```

Κώδικας 4-7 Κώδικας ψηφιακού αισθητήρα υπερήχων, HC-SR04 Ultrasonic Module Distance Measuring Transducer

4.1.9 RGB LED, KY-016 Color LED Module

Το συγκεκριμένο module διαθέτει τρία βασικά χρώματα (πράσινο, κόκκινο, μπλε) και μέσω διαμόρφωσης pwm μπορούν να παραχθούν πολλά χρώματα.



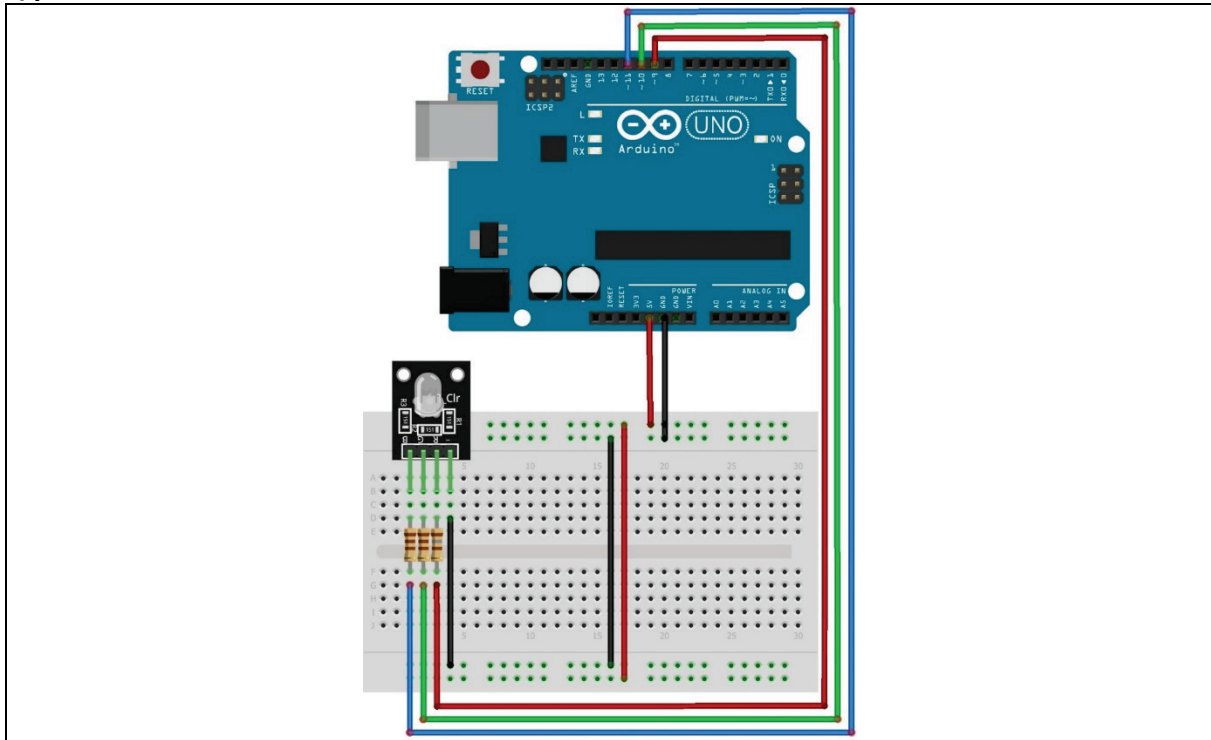
Εικόνα 4-16 Πλακέτα RGB LED, KY-016 Color LED Module

(The most complete starterkit, n.d.)

Τεχνικά χαρακτηριστικά:

- Χρήση πλήρους έγχρωμης λυχνίας LED
- RGB τριχρωματική αντίσταση περιορισμού για την πρόληψη της εξουθένωσης(burnout).
- Μέσω της ρύθμισης PWM τρία βασικά χρώματα μπορούν να αναμιχθούν σε διαφορετικά χρώματα.
- Με μια ποικιλία διεπαφών ενός τσιπ.
- Τάση λειτουργίας: 5V
- LED drive mode: common cathode driver

Σύνδεση με το Arduino



Εικόνα 4-17 Σύνδεση πλακέτας RGB LED, KY-016 Color LED Module

Πρόγραμμα κώδικα

Ο παρακάτω κώδικας Arduino θα αλλάζει διάφορα χρώματα σε ένα loop αλλάζοντας την τιμή PWM σε καθένα από τα τρία βασικά χρώματα.

```
int redledpin = 9; //pin για red LED
int greenledpin = 10; //pin για green LED
int blueledpin = 11; //pin για blue LED
int val; //ορισμός PWM μεταβλητής
void setup() {
  pinMode(redledpin, OUTPUT);
  pinMode(blueledpin, OUTPUT);
  pinMode(greenledpin, OUTPUT);
  Serial.begin(9600); //Εκκίνηση Serial Port
}
void loop()
{
  for(val = 255; val > 0; val--)
  {
```



```

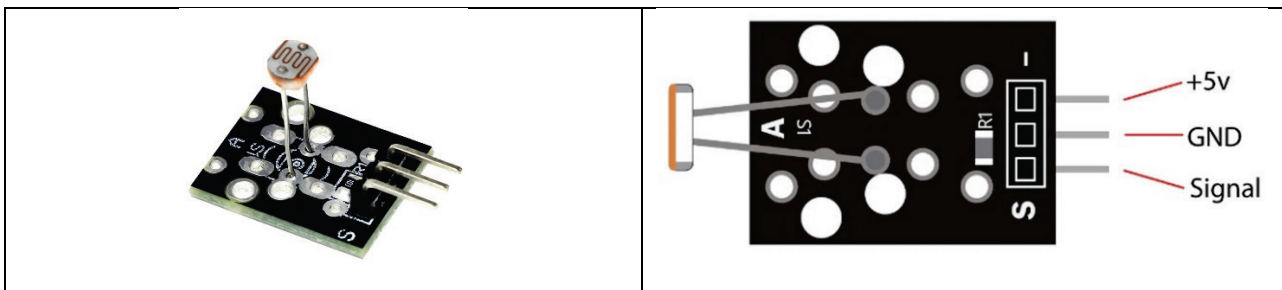
analogWrite(redledpin, val); //ορισμός PWM τιμής για red LED
analogWrite(greenledpin, 128 - val); //ορισμός PWM τιμής για green LED
analogWrite(blueledpin, 255 - val); //ορισμός PWM τιμής για blue LED
Serial.println(val); //εκτύπωση τιμής
delay(100); //Καθυστέρηση
}
for(val = 0; val < 255; val++)
{
analogWrite(redledpin, val);
analogWrite(greenledpin, 128 - val);
analogWrite(blueledpin, 255 - val);
Serial.println(val);
delay(100); //Καθυστέρηση
}
}

```

Κώδικας 4-8 Κώδικας ψηφιακού αισθητήρα υπερήχων, HC-SR04 Ultrasonic Module Distance Measuring Transducer

4.1.10 Φωτοαντίσταση, LDR KY-018 Photoresistor module

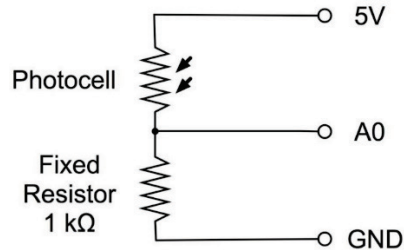
Αυτή η μονάδα αποτελείται από μία φωτοαντίσταση και μία αντίσταση 10 kΩ συνδεδεμένα σε σειρά. Η αντίσταση της φωτοαντίστασης μειώνεται όσο αυξάνεται το φως, ενώ μειώνεται όταν το φως μειώνεται. Η αναλογική έξοδος καθορίζει την ένταση του φωτός.



Εικόνα 4-18 Πλακέτα φωτοαντίσταση, LDR KY-018 Photoresistor module

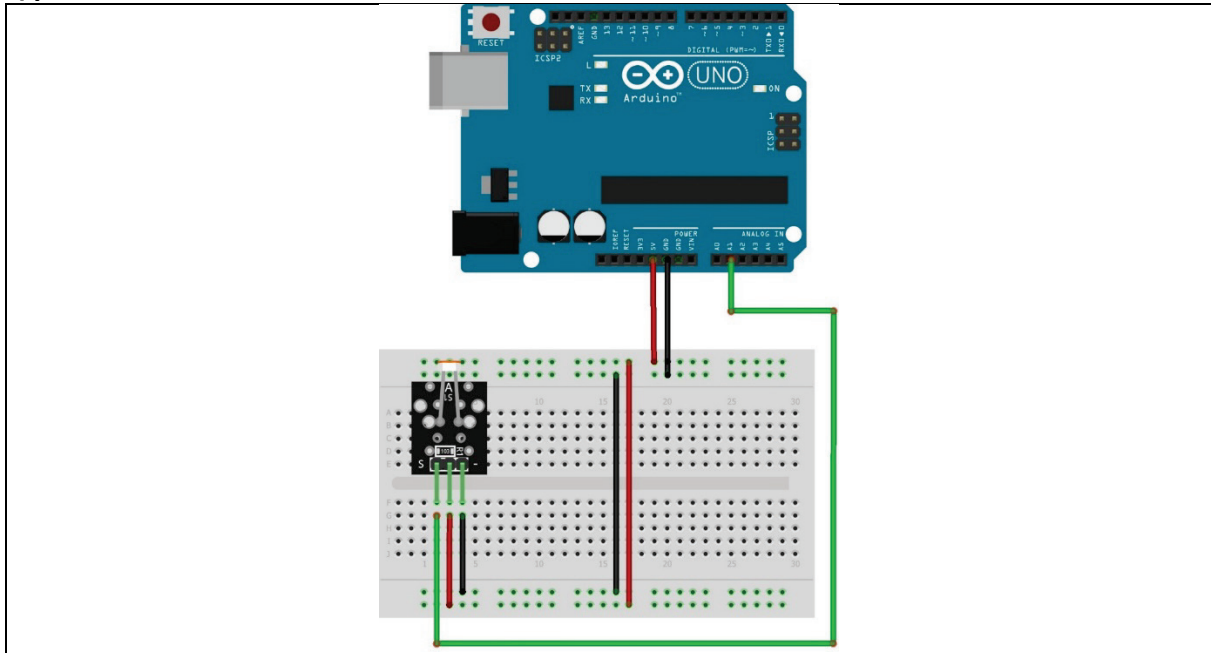
(The most complete starterkit, n.d.)

Πιο συγκεκριμένα, έχει μία αντίσταση περίπου 50 kΩ στο απόλυτο σκοτάδι και 500 Ω σε έντονο φως. Για να μετατραπεί αυτή η μεταβαλλόμενη αντίσταση σε ένα αισθητήρα φωτός, θα πρέπει η προσπίπτουσα ποσότητα φωτός να μετατραπεί σε τάση. Για να γίνει αυτό αρκεί να συνδεθεί με μια σταθερή αντίσταση 1 kΩ δημιουργώντας έτσι ένα διαιρέτη τάσης. Η αντίσταση και το LDR μαζί συμπεριφέρονται σαν ένα ποτενσιόμετρο, όταν το φως είναι πολύ δυνατό, τότε η τιμή του LDR είναι πολύ χαμηλή σε σύγκριση με την σταθερή τιμή της αντίστασης. Όταν το LDR βρίσκεται στο σκοτάδι, η αντίσταση γίνεται μεγαλύτερη από την σταθερή αντίσταση.



Εικόνα 4-19 Κύκλωμα φωτοαντίστασης, LDR KY-018 Photoresistor module
(The most complete starterkit, n.d.)

Σύνδεση με το Arduino



Εικόνα 4-20 Σύνδεση πλακέτας φωτοαντίστασης, LDR KY-018 Photoresistor module

Πρόγραμμα κώδικα

Ο παρακάτω κώδικας Arduino θα εκπέμπει μετρήσεις από τη φωτοαντίσταση. Καλύψτε τη μονάδα με το χέρι σας για να αποφευχθεί ο φωτισμός και οι τιμές εξόδου θα είναι χαμηλές, δείξτε σε φως τον αισθητήρα και οι τιμές θα είναι υψηλές.

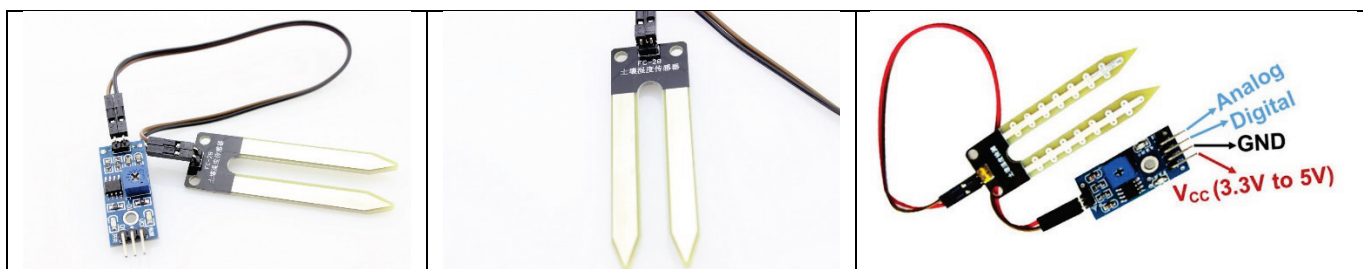
```
int sensorPin = 1; //ορισμός αναλογικού pin 1
int value = 0; //αρχικοποίηση της value
void setup() {
  Serial.begin(9600); //Εκκίνηση Serial Port
}
void loop() {
  value = analogRead(sensorPin);
  Serial.println(value, DEC); //εκτύπωση τιμής, μεγάλη τιμή – φωτεινότητα, μικρή τιμή - σκοτάδι
  delay(1000); //Καθυστέρηση 1 sec
}
```

Κώδικας 4-9 Κώδικας φωτοαντίστασης, LDR KY-018 Photoresistor module

4.1.11 Αισθητήρας υγρασίας εδάφους, Soil moisture detector module

Αυτός ο αισθητήρας υγρασίας εδάφους μπορεί να χρησιμοποιηθεί για να ανιχνεύσει την υγρασία του εδάφους ή να κρίνει εάν υπάρχει νερό γύρω από τον αισθητήρα. Τοποθετήστε τον στο έδαφος και στη συνέχεια ρυθμίστε με το ποτενσιόμετρο την ευαισθησία. Ο αισθητήρας θα εξάγει τη λογική HIGH / LOW όταν η υγρασία είναι υψηλότερη / χαμηλότερη από το όριο που έχει οριστεί από το ποτενσιόμετρο.

Με τη χρήση δύο ανιχνευτών (probes), υπολογίζεται η ογκομετρική περιεκτικότητα του νερού στο έδαφος. Το ρεύμα περνά μέσα από τους ανιχνευτές και το έδαφος μεταξύ τους, και στη συνέχεια, βάσει της αντίστασης που μετράται ανάμεσα στους ανιχνευτές, παράγεται το σήμα του αισθητήρα (τάση εξόδου). Η αγωγιμότητα του εδάφους αυξάνεται με την αύξηση της περιεκτικότητας σε νερό, με αποτέλεσμα να μειώνεται η αντίσταση. Συνεπώς, η τιμή της υγρασίας που εμφανίζεται είναι υψηλότερη. Το ξηρό έδαφος είναι κακός αγωγός του ηλεκτρισμού, οδηγώντας σε μεγαλύτερη αντίσταση. Το σήμα του αισθητήρα τροφοδοτείται σε ένα LM393, το οποίο το ψηφιοποιεί και το καθιστά διαθέσιμο σε μια ακίδα ψηφιακής εξόδου. Η επιθυμητή τιμή υγρασίας ρυθμίζεται μέσω ενός μπλε ποτενσιόμετρου.



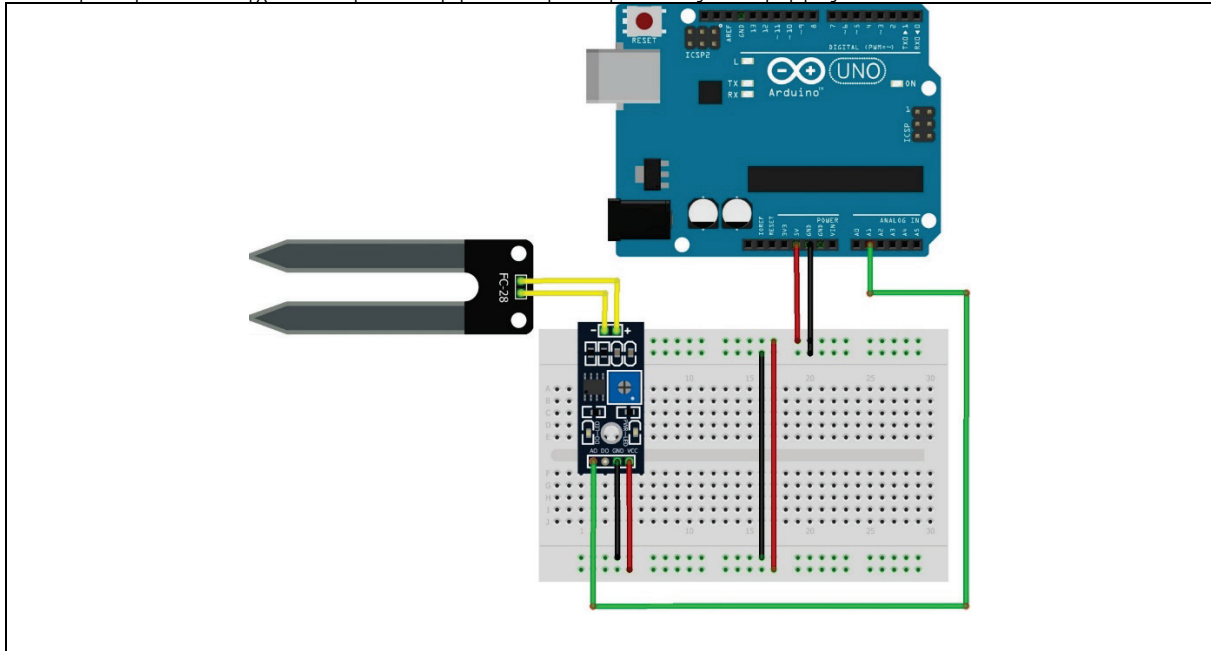
Εικόνα 4-21 Πλακέτα αισθητήρα υγρασίας εδάφους, Soil moisture detector module

(The most complete starterkit, n.d.)

Τεχνικά χαρακτηριστικά:

- Τάση λειτουργίας: 3.3V με 5V.
- Ρυθμιζόμενη ευαισθησία (μπλε ποτενσιόμετρο)
- Λειτουργία διπλής εξόδου, αναλογική έξοδος ακριβέστερη.
- Με μία οπή για εύκολη εγκατάσταση.
- Δείκτης λειτουργίας (κόκκινο) και την ένδειξη ψηφιακής εξόδου (πράσινο).
- Τσιπ σύγκρισης: LM393
- Διάσταση PCB: 3cm x 1.5cm.
- Διάσταση probe εδάφους: 6cm x 2cm.
- Μήκος καλωδίου: 21εκ.

Σύνδεση με το Arduino



Εικόνα 4-22 Σύνδεση αισθητήρα υγρασίας εδάφους, Soil moisture detector module

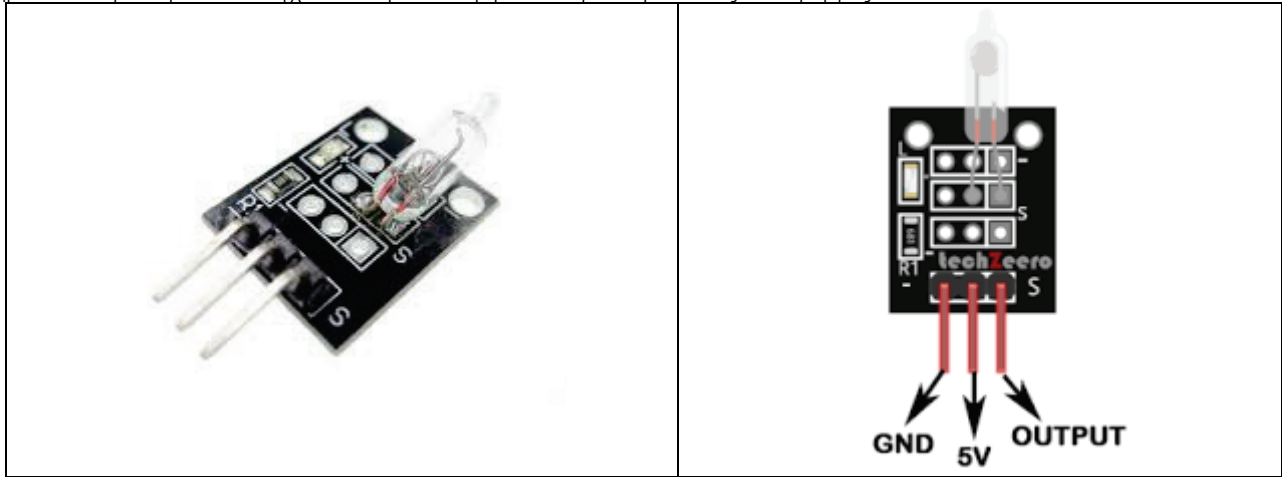
Πρόγραμμα κώδικα

```
const int moisture = A1; //ορισμός αναλογικού pin A1
int value; //αρχικοποίηση της value
void setup()
{
  Serial.begin(9600); //Εκκίνηση Serial Port
}
void loop()
{
  //Με τιμή γύρω στο 400, θεωρούμε ότι το χώμα έχει αρκετή υγρασία (καλά ποτισμένο)
  value = analogRead(moisture); //Διάβασμα αναλογικής τιμής
  value = constrain(value,400,1023); //Ορισμός ορίων
  value = map(value,400,1023,100,0); //Αντιστοίχιση τιμής 400 στο 100και 1023 στο 0
  Serial.print("Soil moisture: ");
  Serial.print(value);
  Serial.println("%");
  delay(1000); //καθυστέρηση 1 sec
}
```

Κώδικας 4-10 Κώδικας αισθητήρα υγρασίας εδάφους, Soil moisture detector module

4.1.12 Διακόπτης κλίσης υδραργύρου, KY-017 Mercury Tilt Switch Module

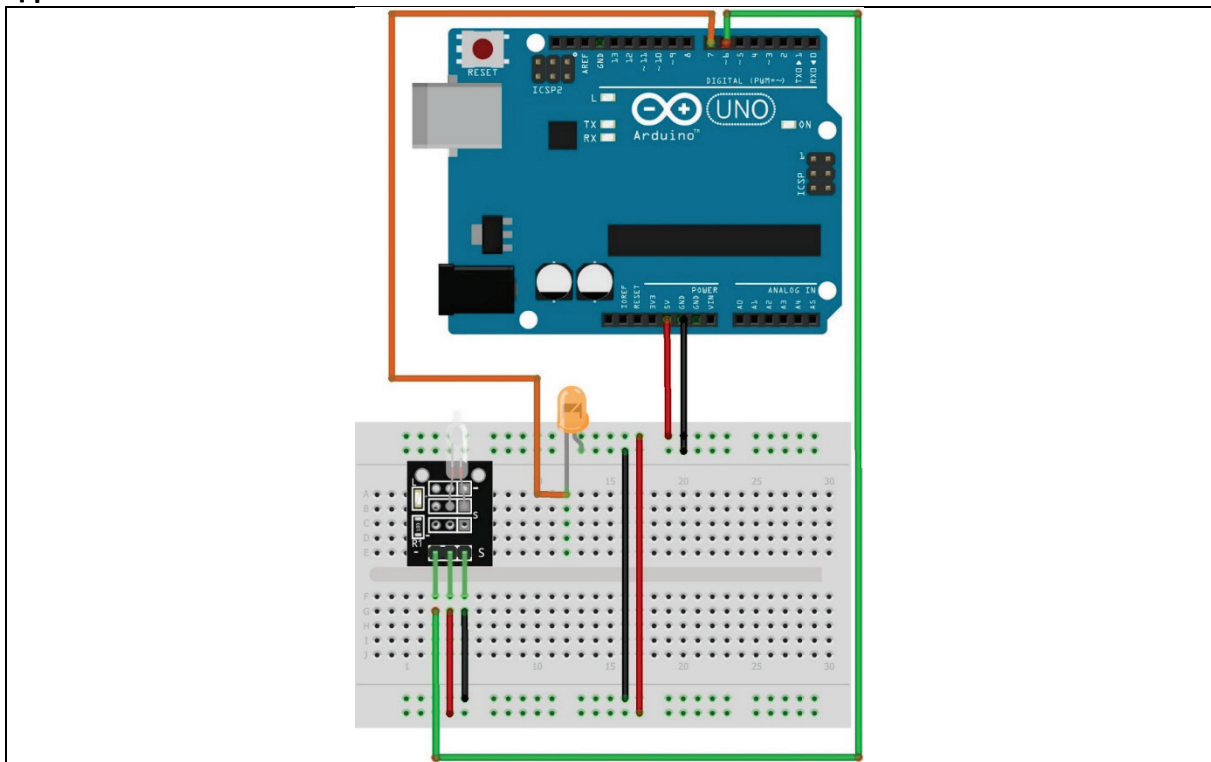
Ο KY-017 διακόπτης κλίσης υδραργύρου χρησιμοποιεί μία μπάλα υδραργύρου για να εντοπίσει αλλαγή στην κλίση. Όταν ο υδράργυρος αλλάξει θέση τότε ο αισθητήρας στέλνει λογικό 1 στην έξοδό του και επίσης ανάβει η ενσωματωμένη προειδοποιητική λυχνία led.



Εικόνα 4-23 Πλακέτα διακόπτη κλίσης υδραργύρου, KY-017 Mercury Tilt Switch Module

(The most complete starterkit, n.d.)

Σύνδεση με το Arduino



Εικόνα 4-24 Σύνδεση πλακέτας διακόπτη κλίσης υδραργύρου, KY-017 Mercury Tilt Switch Module

Πρόγραμμα κώδικα

Ο παρακάτω κώδικας θα ανάψει το led στο pin 7 του Arduino όταν η μονάδα είναι κεκλιμένη.

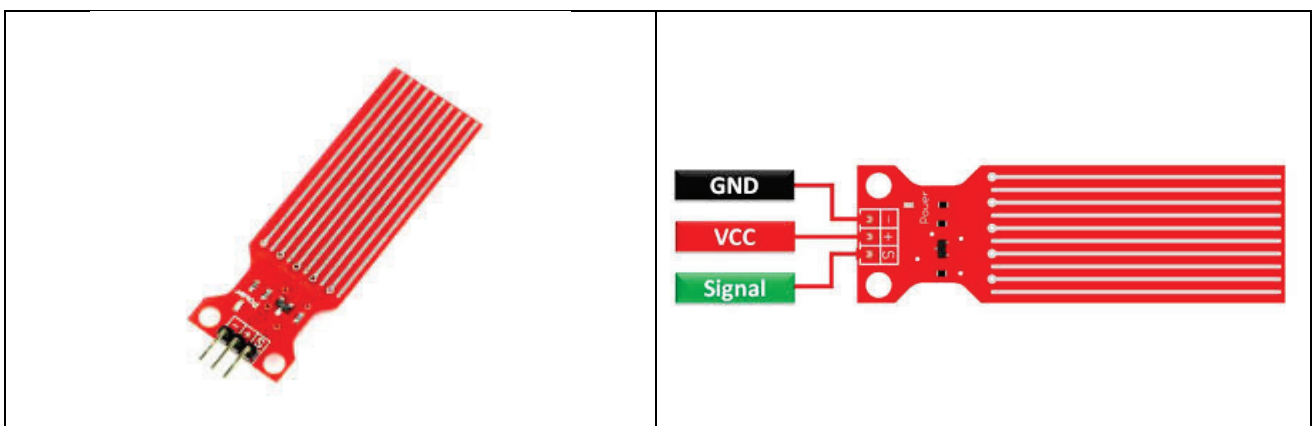
```
int mercurypin = 6; //Ορισμός pin αισθητήρα κλίσης(υδραργύρου) 6
int orangeledpin = 7; //Ορισμός Led Pin 7
int value; //αρχικοποίηση της value
void setup()
{
  pinMode(orangeledpin, OUTPUT);
  pinMode(mercurypin, INPUT);
}
```

```
void loop()
{
  value = digitalRead(mercurypin); //κατάσταση αισθητήρα κλίσης(υδραργύρου)
  if(value == HIGH)
  {
    digitalWrite(orangeledpin, HIGH);
  }
  else
  {
    digitalWrite(orangeledpin, LOW);
  }
  delay(1000); //καθυστέρηση 1 sec
}
```

Κώδικας 4-11 Κώδικας διακόπτη κλίσης υδραργύρου, KY-017 Mercury Tilt Switch Module

4.1.13 Αισθητήρας νερού βροχής, Water Level Sensor

Ο αισθητήρας βροχόπτωσης είναι ένα εύκολο εργαλείο για την καταγραφή της βροχής. Μπορεί να χρησιμοποιηθεί σαν διακόπτης όταν η βροχή πέφτει πάνω στον αισθητήρα, αλλά και σαν μετρητής της έντασης των βροχοπτώσεων. Ο αισθητήρας βροχής ανιχνεύει το νερό το οποίο συμπληρώνει το κύκλωμα των κελιών που τυπώνετε στην πλακέτα του αισθητήρα. Όταν ο αισθητήρας βυθίζεται, οι αγωγοί έρχονται σε επαφή με το υγρό στο επίπεδο της στάθμης του, προκαλώντας μεταβολή στη συνολική αντίσταση. Αυτή η μεταβολή στην αντίσταση αντιστοιχεί στην απόσταση από την κορυφή του αισθητήρα έως την επιφάνεια του νερού. Ο αισθητήρας δέχεται ως είσοδο 5 Volt και παράγει μια μεταβαλλόμενη τάση εξόδου από 0 έως 4,2 Volt, ανάλογα με τη στάθμη του νερού.



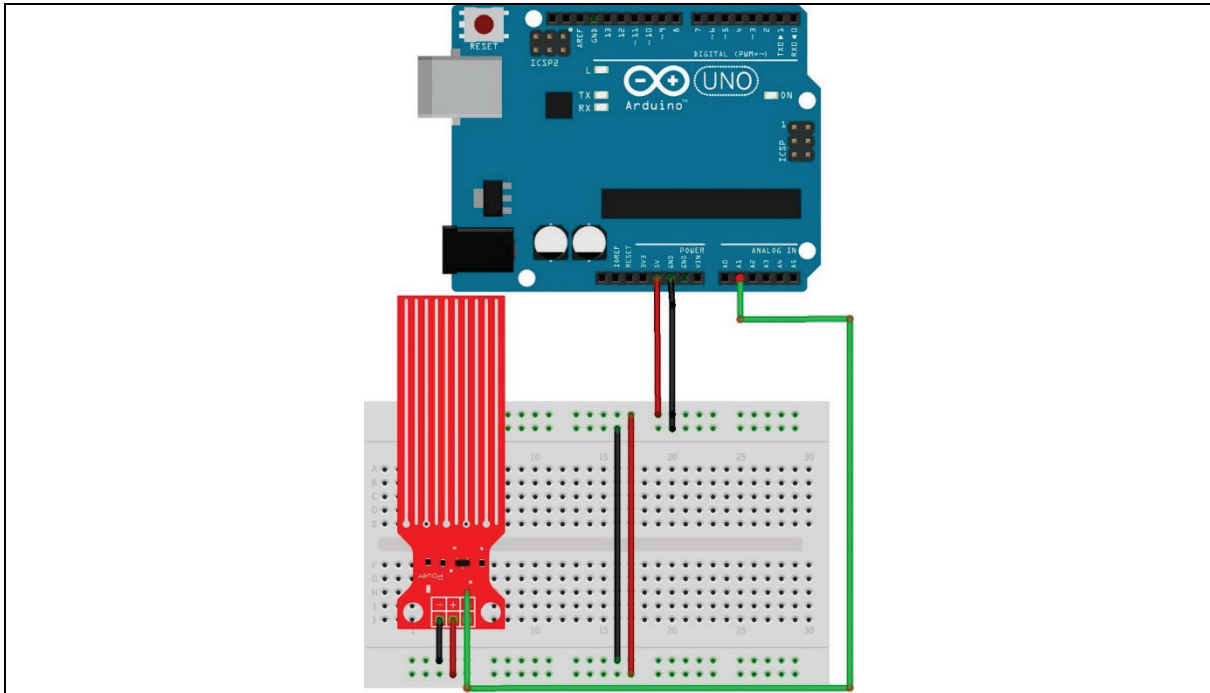
Εικόνα 4-25 Πλακέτα αισθητήρα νερού βροχής, Water Level Sensor

(The most complete starterkit, n.d.)

Τεχνικά χαρακτηριστικά:

- Τάση λειτουργίας: 5V DC.
- Ρεύμα λειτουργίας: λιγότερο από 35mA.

- Τύπος: Αναλογικός αισθητήρας.
- Περιοχή δοκιμής: 40 mm x 18 mm.
- Θερμοκρασία λειτουργίας: 10 °C έως 30 °C.
- Υγρασία λειτουργίας: 10% - 90%.
- Μέγεθος οπών στερέωσης: Φ3mm.
- Μέγεθος προϊόντος: 60mm x 22mm.



Εικόνα 4-26 Σύνδεση αισθητήρα νερού βροχής, Water Level Sensor

Πρόγραμμα κώδικα

Πάρτε ένα ποτήρι νερό και βάλτε αργά στον αισθητήρα. Οι αναλογικές τιμές από αυτή τη διαδικασία είναι:

- 0mm - 480
- 5mm - 530
- 10mm - 615
- 15mm - 660
- 20mm - 680
- 25mm - 690
- 30mm - 700
- 35mm - 705
- 40mm - 710

Η ανάλυση του αισθητήρα μειώνεται καθώς το νερό φτάνει στο μέγιστο επίπεδο.

Στο επόμενο κώδικα χαρτογραφούνται αυτές τις τιμές για να εκτυπωθούν στην σειριακή οθόνη οι στάθμες του νερού σε mm.

```
const int read = A1; //ορισμός αναλογικού pin A1
int value; //αρχικοποίηση της value
void setup()
```



```

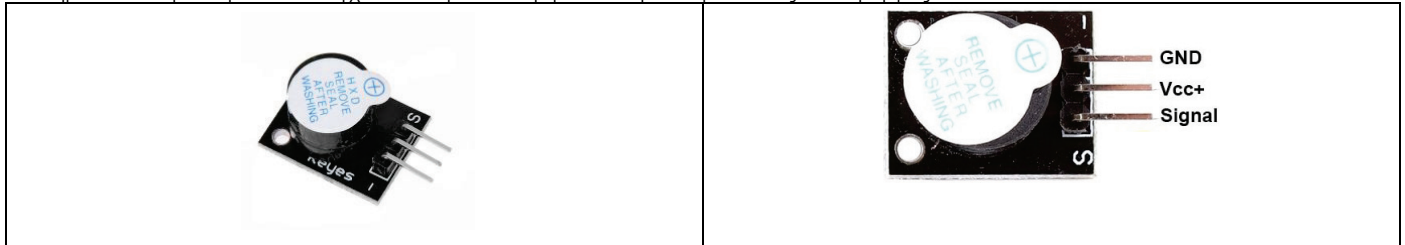
{
  Serial.begin(9600); //Εκκίνηση Serial Port
}
void loop()
{
  value = analogRead(read); //Αποθήκευση της αναλογικής τιμης που διαβάζει στην value
  if (value<=480){
    Serial.println("Water level: 0mm - Empty!");
  }
  else if (value>480 && value<=530){
    Serial.println("Water-level: 0mm to 5mm");
  }
  else if (value>530 && value<=615){
    Serial.println("Water-level: 5mm to 10mm");
  }
  else if (value>615 && value<=660){
    Serial.println("Water level: 10mm to 15mm");
  }
  else if (value>660 && value<=680){
    Serial.println("Water-level: 15mm to 20mm");
  }
  else if (value>680 && value<=690){
    Serial.println("Water-level: 20mm to 25mm");
  }
  else if (value>690 && value<=700){
    Serial.println("Water-level: 25mm to 30mm");
  }
  else if (value>700 && value<=705){
    Serial.println("Water-level: 30mm to 35mm");
  }
  else if (value>705){
    Serial.println("Water-level: 35mm to 40mm");
  }
  delay(1000); //καθυστέρηση 1 sec
}

```

Κώδικας 4-12 Κώδικας αισθητήρα νερού βροχής, Water Level Sensor

4.1.14 Ενεργός πιεζοηλεκτρικός βομβητής, KY-012 Active Buzzer Module

Ο ενεργός πιεζοηλεκτρικός βομβητής KY-012 παράγει έναν μονοτονικό ήχο όταν δέχεται ψηφιακό σήμα 1. Επίσης ο αντίστοιχος παθητικός βομβητής μπορεί να παράγει πολυτονικούς ήχους. Μπορεί να χρησιμοποιηθεί ως συσκευή συναγερμού, για να ειδοποιεί όταν προκύπτει μια κατάσταση alarm.



Εικόνα 4-27 Πλακέτα ενεργού πιεζοηλεκτρικού βομβητή, KY-012 Active Buzzer Module

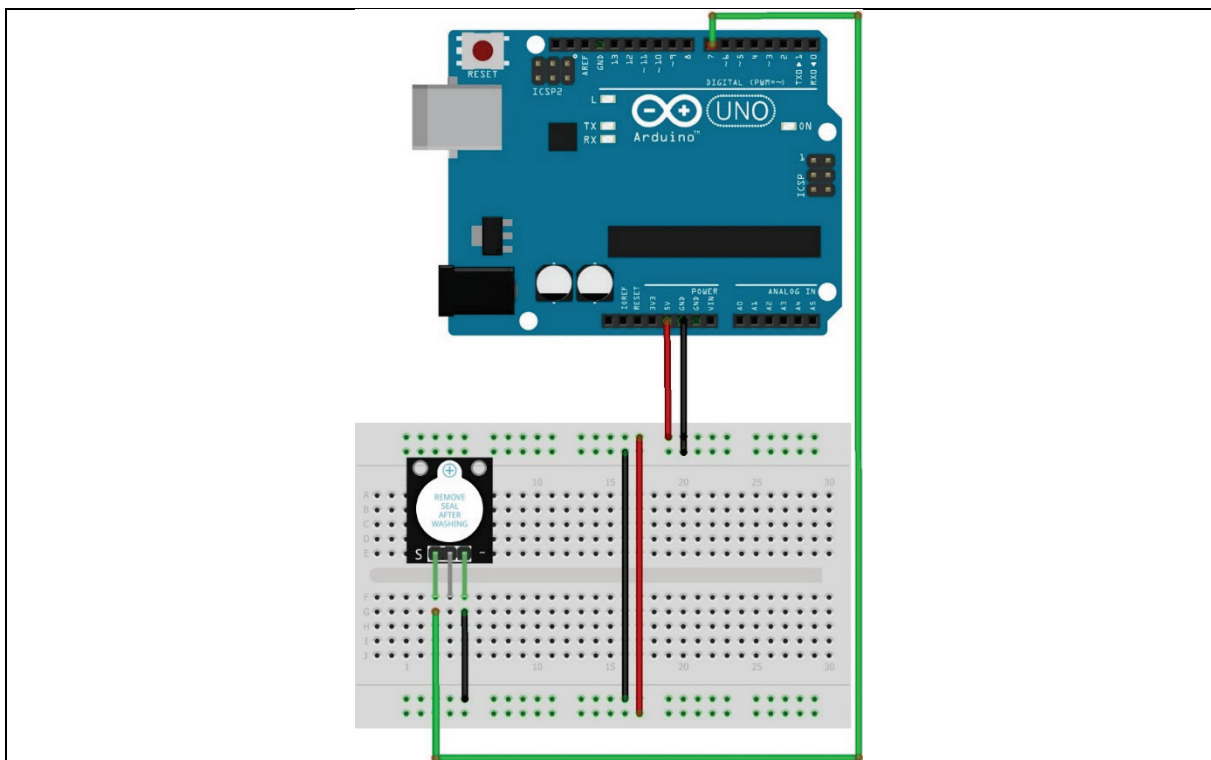
(The most complete starterkit, n.d.)

Τεχνικά χαρακτηριστικά:

- Τάση λειτουργίας: 3.5V ~ 5.5V.
- Μέγιστο ρεύμα: 30mA / 5VDC.
- Συχνότητα συντονισμού: 2500Hz $\pm$ 300Hz.
- Θερμοκρασία λειτουργίας: -20 ° C ~ 70 ° C.
- Θερμοκρασία αποθήκευσης: -30 ° C ~ 105 ° C.
- Διαστάσεις: 18.5mm x 15mm.

Σύνδεση με το Arduino

Συνδέστε το σήμα (S) στον ακροδέκτη 8 στο Arduino και το Ground (-) στο GND. Λάβετε υπόψη σας ότι μερικοί πίνακες έχουν λανθασμένη επισήμανση, δοκιμάστε να αναστρέψετε τα καλώδια εάν δεν μπορείτε να ακούσετε κανέναν ήχο κατά την εκτέλεση του κώδικα.



Εικόνα 4-28 Σύνδεση ενεργού πιεζοηλεκτρικού βομβητή, KY-012 Active Buzzer Module

Πρόγραμμα κώδικα

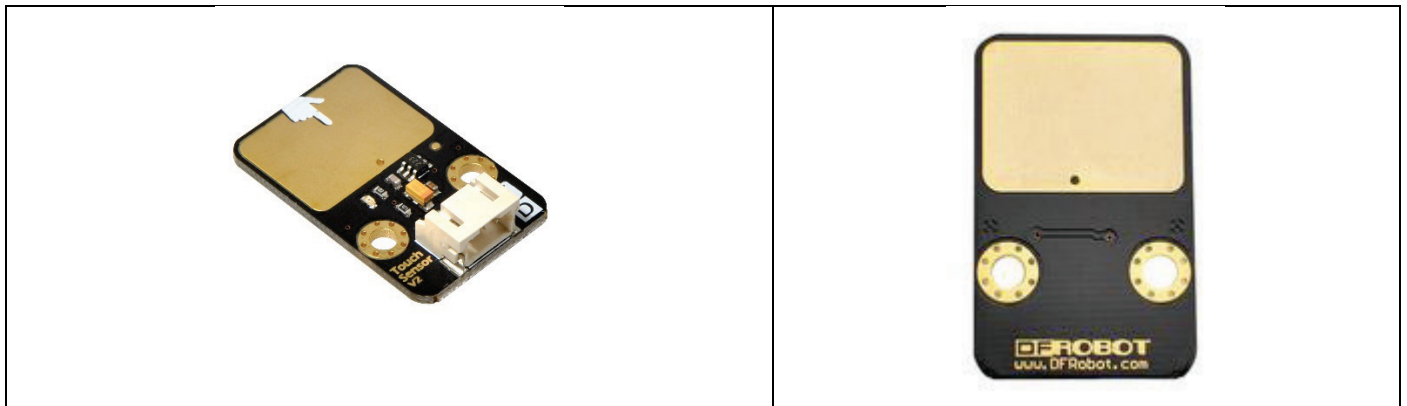
Ο επόμενος κώδικας του Arduino ενεργοποιεί και απενεργοποιεί το βομβητή σε μία συνεχόμενη επανάληψη, δημιουργώντας μια σειρά από σύντομα μεγάλα μπιτ.

```
int buzzerpin = 7; //Ορισμός pin βομβητή 7
void setup ()
{
  pinMode (buzzerpin, OUTPUT);
}
void loop ()
{
  digitalWrite (buzzerpin, HIGH);
  delay (1000); //καθυστέρηση 1 sec
  digitalWrite (buzzerpin, LOW);
  delay (1000); //καθυστέρηση 1 sec
}
```

Κώδικας 4-13 Κώδικας ενεργού πιεζοηλεκτρικού βομβητή, KY-012 Active Buzzer Module

4.1.15 Χωρητικός αισθητήρας αφής, DFRobot Capacitive Touch Sensor

Ο χωρητικός αισθητήρας αφής μπορεί να αισθάνεται την ανθρώπινη αφή ή τα μέταλλα και να στείλει στην έξοδο του σήμα 1 ή 0 αντίστοιχα. έχει πολύ μεγάλη ευαισθησία καθώς μπορεί να ανιχνεύσει αφή και μέσα από κάποιο ύφασμα.



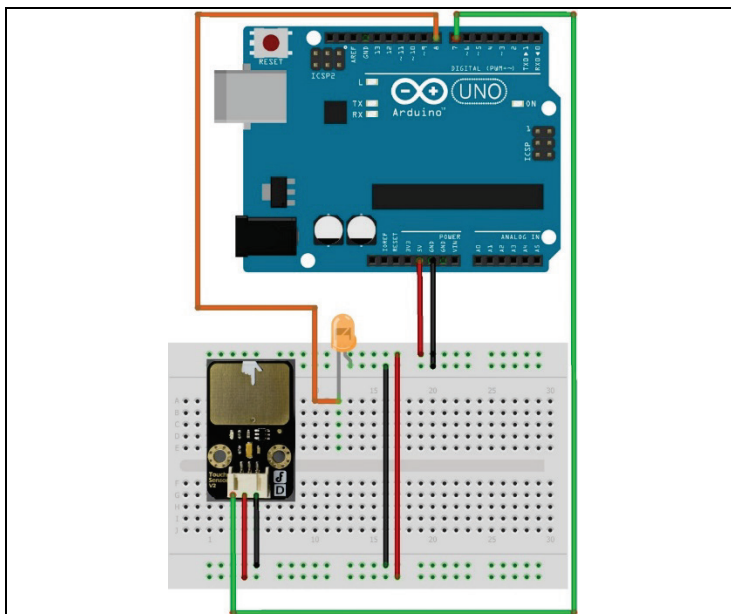
Εικόνα 4-29 Πλακέτα χωρητικού αισθητήρα αφής, DFRobot Capacitive Touch Sensor

(DFRobot, n.d.)

Τεχνικά χαρακτηριστικά:

- Τάση τροφοδοσίας: 3.3 V- 5V
- Ψηφιακή έξοδος
- Μέγεθος: 22x 30 mm

Σύνδεση με το Arduino



Εικόνα 4-30 Σύνδεση χωρητικού αισθητήρα αφής, DFRobot Capacitive Touch Sensor

Πρόγραμμα κώδικα

Μετά την εκκίνηση, αν αγγίξετε την μεταλλική επιφάνεια του αισθητήρα αφής με το δάχτυλο, ανάβει η λυχνία LED.

```
int ledpin = 8; // Ορισμός Led Pin 8
int touch = 7; //Ορισμός pin αισθητήρα αφής 7
void setup(){
  pinMode(ledpin, OUTPUT); //Ορισμός του ledpin σε έξοδο
  pinMode(touch, INPUT); //Ορισμός του touch σε είσοδο
}
void loop(){
  if(digitalRead(touch)==HIGH) { //διάβασμα του αισθητήρα αφής
    digitalWrite(ledPin, HIGH); //άνοιγμα του led εάν ο αισθητήρας αφής είναι high
  }
  else{
    digitalWrite(ledPin, LOW); //σβήσιμο του led εάν ο αισθητήρας αφής είναι low
  }
}
```

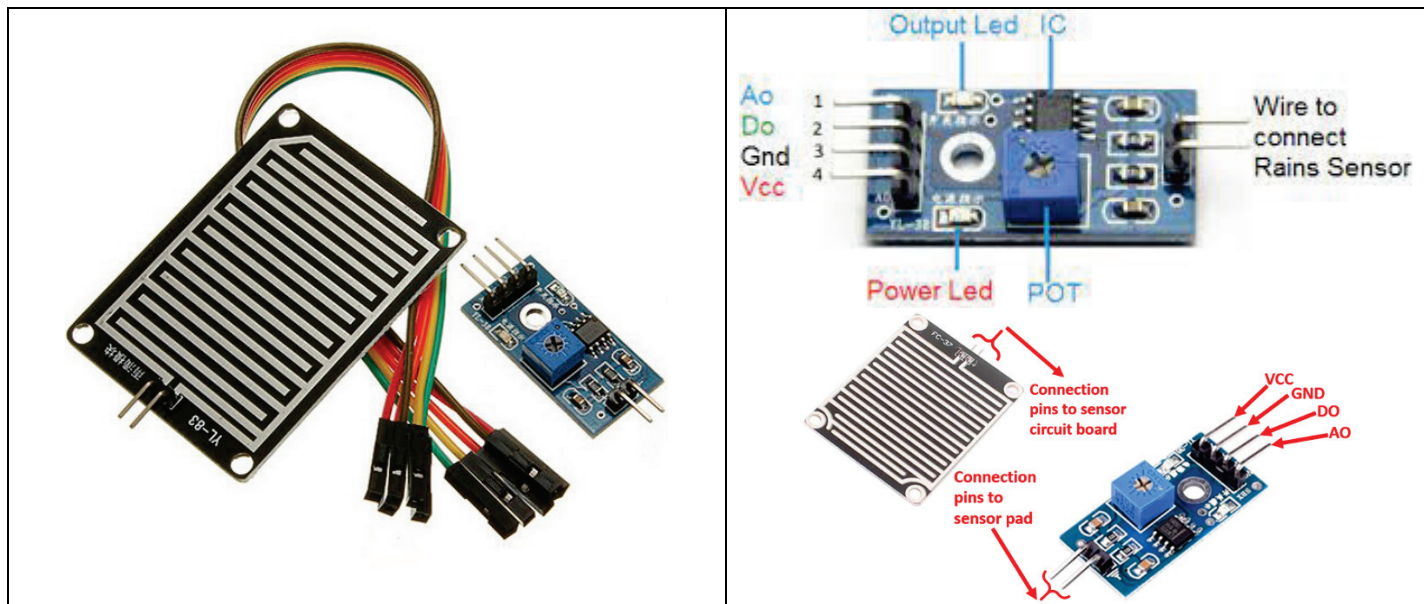
Κώδικας 4-14 Κώδικας χωρητικού αισθητήρα αφής, DFRobot Capacitive Touch Sensor

4.1.16 Αισθητήρας ανίχνευσης βροχής FC-37

Η μονάδα αισθητήρα βροχής προσφέρει έναν οικονομικό και απλό τρόπο ανίχνευσης βροχής. Η λειτουργία της βασίζεται σε μια σειρά από παράλληλους, εκτεθειμένους αγωγούς με συγκεκριμένη αντίσταση. Όταν σταγόνες βροχής πέφτουν πάνω στην επιφάνεια του αισθητήρα, οι αγωγοί γεφυρώνονται, με απο-

τέλεσμα την αλλαγή στη συνολική αντίσταση. Η μεταβολή αυτής της αντίστασης αντιστοιχεί στην ποσότητα του νερού πάνω στην επιφάνεια του αισθητήρα. Η μονάδα δέχεται ως είσοδο 5 Volt και παράγει μια μεταβαλλόμενη τάση εξόδου, ανάλογη με την ποσότητα του νερού.

Η αναλογική έξοδος χρησιμεύει για την ανίχνευση της έντασης της βροχής, ενώ η ψηφιακή έξοδος υποδεικνύει εάν βρέχει ή όχι. Όταν συνδεθεί στα 5 Volt, το LED θα ανάψει, και αν δεν υπάρχει βροχή, η ψηφιακή έξοδος θα είναι HIGH. Αν πέσουν λίγες σταγόνες νερού, η ψηφιακή έξοδος θα γίνει LOW.



Εικόνα 4-31 Πλακέτα αισθητήρα ανίχνευσης βροχής FC-37

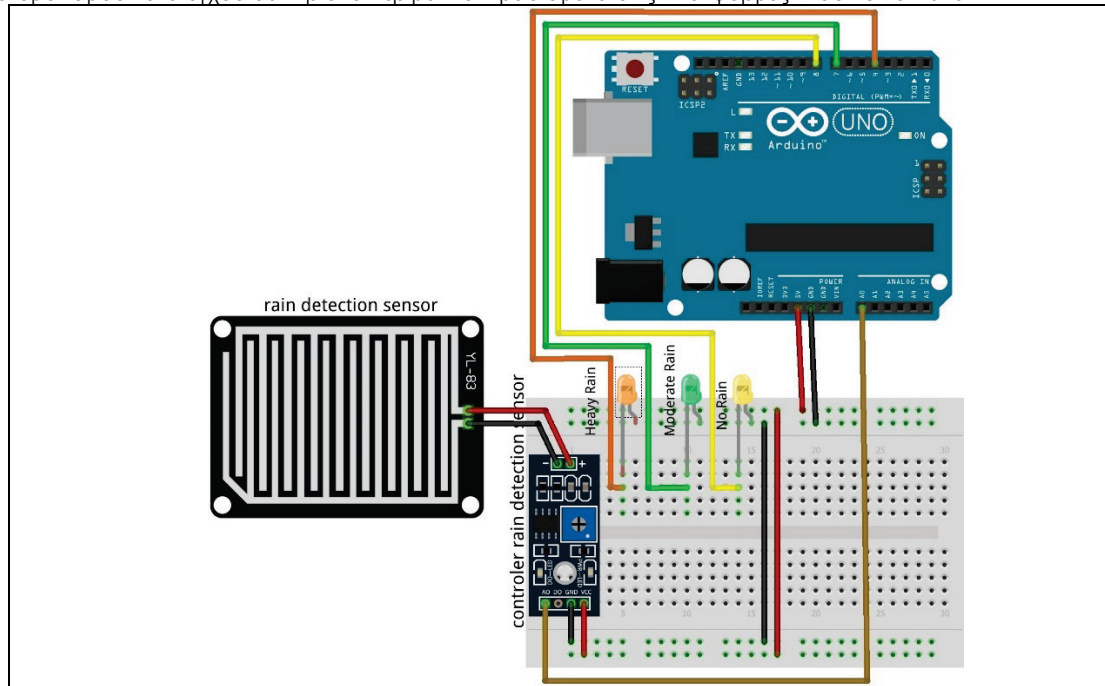
(The most complete starterkit, n.d., selliliar, n.d., androiderode, n.d., kuongshun, n.d.)

Τεχνικά χαρακτηριστικά:

- Υιοθετεί υψηλής ποιότητας υλικό διπλής όψης RF-04.
- Περιοχή: 5cm x 4cm πλάκα νικελίου στην πλευρά.
- Αντιοξειδωτική, αντι-ηλεκτρική, με μεγάλο χρόνο χρήσης.
- Η καθαρή κυματομορφή του σήματος εξόδου του συγκριτή είναι καλή, ικανότητα οδήγησης, πάνω από 15mA.
- Το ποτενσιόμετρο ρυθμίζει την ευαισθησία.
- Τάση εργασίας 5V.
- Μορφή εξόδου: (0 και 1) και αναλογική τάση έξοδος AO.
- Με σπές βιδών για εύκολη εγκατάσταση.
- Μικρό μέγεθος πλακέτας PCB: 3.2cm x 1.4cm.
- Χρησιμοποιεί συγκριτή LM393 ευρείας τάσης

Translated with DeepL.com (free version)

Το κύκλωμα σύνδεσης του αισθητήρα ανίχνευσης βροχής είναι:



Εικόνα 4-32 Σύνδεση αισθητήρα ανίχνευσης βροχής FC-37

Ο κώδικας για τον αισθητήρα ανίχνευσης βροχής:

```
int led=4; // pin led Heavy Rain
int led1=7; // pin led Moderate Rain
int led2=8; // pin led No Rain
void setup(){
  Serial.begin(9600);
  pinMode(2, OUTPUT);
}
void loop(){
  //analog output
  if(analogRead(0)<300) {
    Serial.println("Heavy Rain");
    digitalWrite(led,HIGH);
  }
  else if(analogRead(0)<500) {
    Serial.println("Moderate Rain");
    digitalWrite(led1,HIGH);}
  else {
    Serial.println("No Rain");
    digitalWrite(led2,HIGH);}
  delay(250);
}
```

Κώδικας 4-15 Κώδικας αισθητήρα ανίχνευσης βροχής FC-37

4.1.17 KIT επικοινωνίας υπέρυθρων, Infrared IR receiver module wireless remote control kit

Το kit επικοινωνίας υπέρυθρων αποτελείται από ένα εξαιρετικά λεπτό τηλεχειριστήριο υπέρυθρων και ένα δέκτη υπέρυθρων 38KHz. Το τηλεχειριστήριο υπέρυθρων διαθέτει διάφορα πλήκτρα λειτουργιών. Οι απόσταση μετάδοσης είναι έως και 8 μέτρα, ενώ είναι ιδανικό για το χειρισμό διάφορων συσκευών σε εσωτερικούς χώρους ή εξωτερικούς χώρους. Οι υπέρυθροι ανιχνευτές είναι μικροσκοπικά μικροσίπ με φωτοκύτταρο που έχουν σχεδιαστεί για να αλληλεπιδρούν με τις υπέρυθρες ακτίνες. Αποτελούνται από έναν πομπό, όπως το τηλεχειριστήριο, που εκπέμπει κωδικοποιημένα σήματα μέσω μιας υπέρυθρης (IR) LED. Το εκπεμπόμενο σήμα είναι μορφής υπέρυθρου φωτός (σήμα IR 38KHz), που δεν είναι ορατό στο ανθρώπινο μάτι. Τα σήματα αυτά ανιχνεύονται από το IR LED του δέκτη και αποκωδικοποιούνται. Ουσιαστικά, ο πομπός μετατρέπει ένα ηλεκτρικό σήμα σε υπέρυθρο, ενώ ο δέκτης μετατρέπει το υπέρυθρο σήμα πίσω σε ηλεκτρικό.



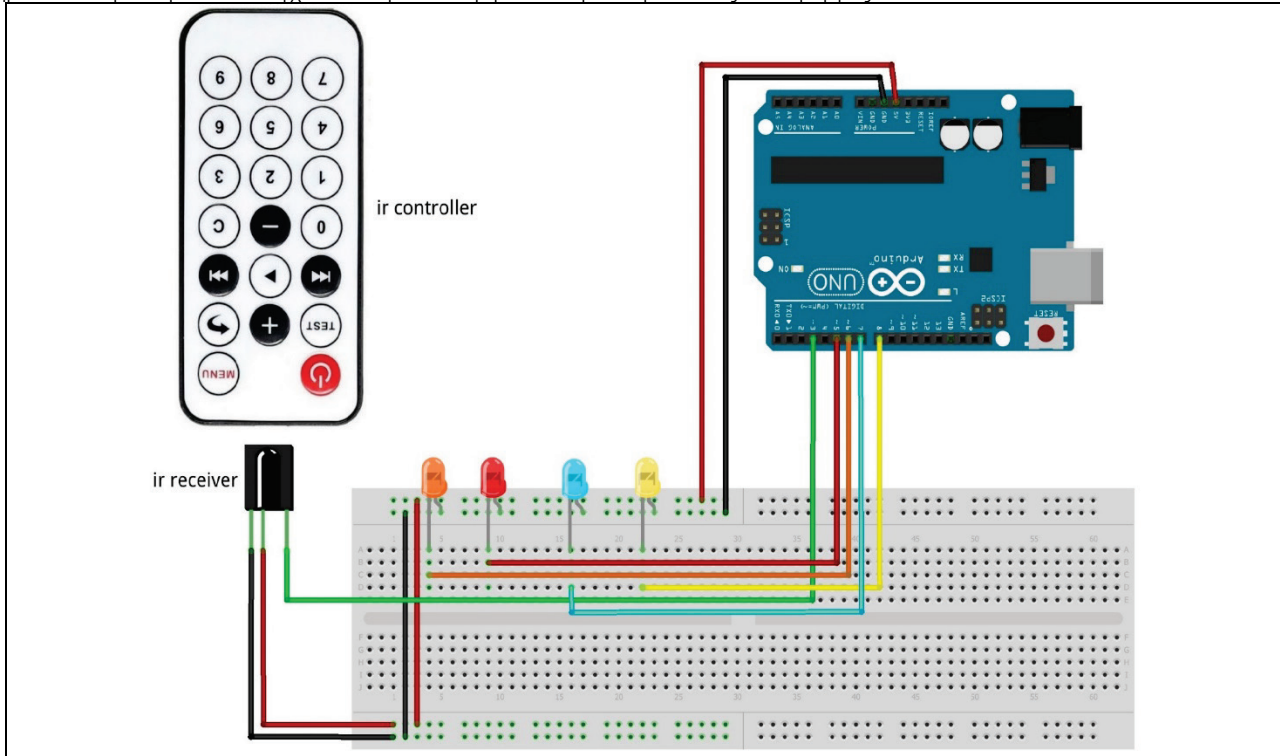
Εικόνα 4-33 KIT επικοινωνίας υπέρυθρων, Infrared IR receiver module wireless remote control kit

(The most complete starterkit, n.d.)

Τεχνικά χαρακτηριστικά:

- Απόσταση μετάδοσης: έως 8m (ανάλογα με το περιβάλλον, την ευαισθησία του δέκτη κλπ).
- Μπαταρία: κουμπί CR2025.
- Χωρητικότητα μπαταρίας: 160mAh.
- Αποτελεσματική γωνία: 60 °.
- Υλικό συγκόλλησης: 0,125mmPET.
- Αποτελεσματική χρήση: 20.000 φορές.
- Στατικό ρεύμα: 3uA - 5uA.
- Δυναμικό ρεύμα: 3mA - 5mA.

Σύνδεση



Εικόνα 4-34 Σύνδεση KIT επικοινωνίας υπέρυθρων, Infrared IR receiver module wireless remote control kit

Πρόγραμμα κώδικα

```
#include <IRremote.h> //Συμπερίληψη βιβλιοθήκης
int RECEIVER-PIN = 3; // Ορισμός ακίδας δέκτη υπέρυθρων
int REDLED = 5; // Ορισμός ακίδας ενδεικτικού led
int ORANGELED = 6; // Ορισμός ακίδας ενδεικτικού led
int CYANLED = 7; // Ορισμός ακίδας ενδεικτικού led
int YELLOWLED = 8; // Ορισμός ακίδας ενδεικτικού led
IRrecv irrecv(RECEIVERPIN); // Δημιουργία αντικειμένου κλάσης IRrecv
decode_results results; // Δήλωση μεταβλητής τύπου decode_results
void setup() {
  //αρχικοποιήσεις
  pinMode(RECEIVERPIN, INPUT);
  pinMode(CYANLED, OUTPUT);
  pinMode(REDLED, OUTPUT);
  pinMode(YELLOWLED, OUTPUT);
  pinMode(ORANGELED, OUTPUT);
  irrecv.enableIRIn(); //Εκκίνηση δέκτη
  Serial.begin(9600);
}
void loop() {
  int i=0;
  if (irrecv.decode(&results)) {
    translateIR();
    unknownRemoter();
    irrecv.resume(); // Reset τον δέκτη ώστε να λάβει τον επόμενο κωδικό
  }
}
void translateIR() //ανάλογα με το τι έχει πατηθεί (IR code που έλαβε) κάνει συγκεκριμένες
```

```

λειτουργίες
{
  switch(results.value){
    case 0x.....:
      Serial.println("LED off_");
      digitalWrite(5,LOW);
      digitalWrite(6,LOW);
      digitalWrite(7,LOW);
      digitalWrite(8,LOW);
      break;
    case 0x.....:
      Serial.println(" RED LED on_");
      digitalWrite(5, HIGH);
      break;
    case 0x.....:
      Serial.println(" ORANGE LED on_");
      digitalWrite(6, HIGH);
      break;
    case 0x.....:
      Serial.println(" CYAN LED on_");
      digitalWrite(7, HIGH);
      break;
    case 0x.....:
      Serial.println(" YELLOW LED on_");
      digitalWrite(8, HIGH);
      break;
    default:
      Serial.print("unknown button_");
      Serial.println(results.value, HEX);
  }
  delay(500);
}

```

Κώδικας 4-16 Κώδικας KIT επικοινωνίας υπέρυθρων, Infrared IR receiver module wireless remote control kit

4.1.18 Ηλεκτροβάννα

Πρόκειται για μία μαγνητική ελεγχόμενη βαλβίδα υγρού, η οποία όταν τεθεί υπό τάση 12V ανοίγει και επιτρέπει στο υγρό να περάσει. Για τη διασύνδεση χρησιμοποιείται ένα τρανζίστορ BJT τύπου npn. Συγκεκριμένα, χρησιμοποιείται το μοντέλο TIP 120. Με αυτόν τον τρόπο, επιτυγχάνεται η σύνδεση συσκευών με υψηλότερη τάση φορτίου (συγκεκριμένα 12V) με τον μικροελεγκτή που λειτουργεί στα 5V. Ουσιαστικά, ο μικροελεγκτής ενεργοποιεί ή απενεργοποιεί τις συσκευές, οι οποίες τροφοδοτούνται μέσω εξωτερικής πηγής, ανεξάρτητα από την αντιστοίχιση του Arduino. Για τη σύνδεση της ηλεκτροβάννας, εκτός του τρανζίστορ BJT χρησιμοποιήθηκαν μία δίοδος και μία αντιστάση των 2.2KΩ.



Εικόνα 4-35 Ηλεκτροβάνα

(GRobotronics, n.d.)

Τεχνικά χαρακτηριστικά:

Θερμοκρασία λειτουργίας: 0 - 100 °C.

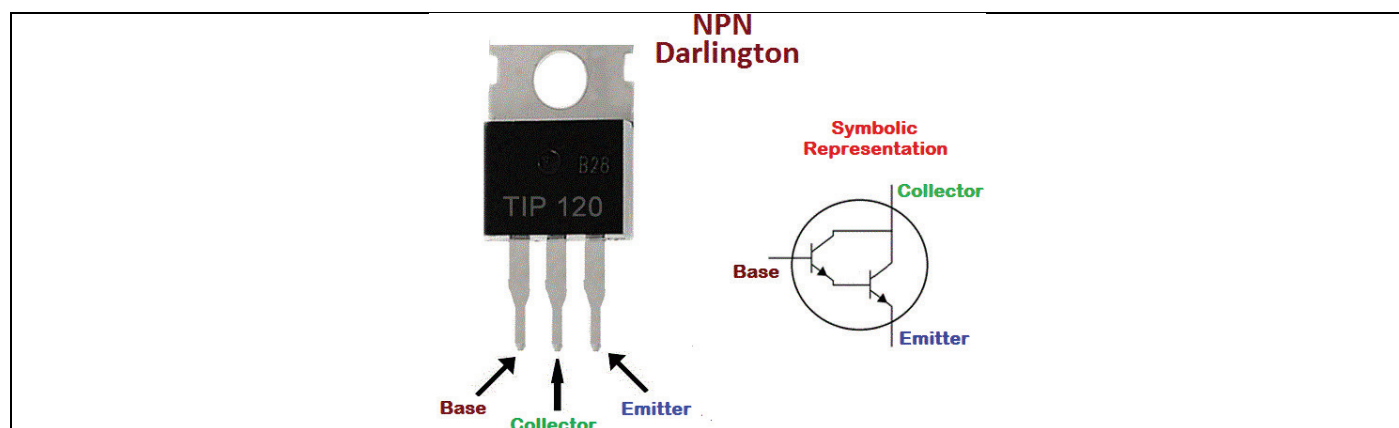
Πίεση λειτουργίας: 400 millimeters water column.

Τάση λειτουργίας: 12V.

Διακύμανση τάσης: 15%.

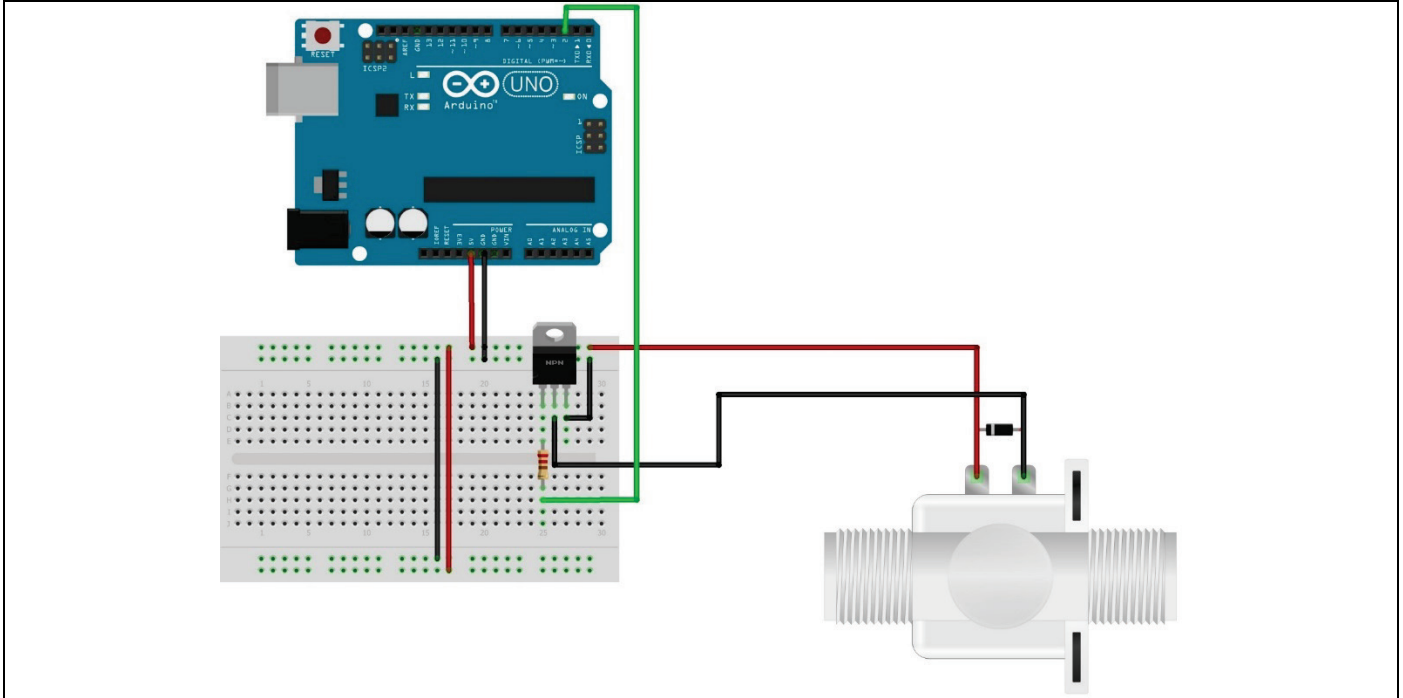
Μέγεθος σπειρώματος: 3/4".

Περιβάλλον εργασίας: Νερό, αέριο, λάδι.



Εικόνα 4-36 Τρανζίστορ TIP120

(netsonic, n.d.)



Εικόνα 4-37 Σύνδεση Ηλεκτροβάνας

Στην παραπάνω διάταξη χρησιμοποιείται μια συσκευή που μπορεί να λειτουργεί με τάση από 5 έως 60 Volt. Ο ακροδέκτης της βάσης του τρανζίστορ συνδέεται σε μια ψηφιακή έξοδο του μικροελεγκτή και ελέγχει τη ροή του ρεύματος μεταξύ του συλλέκτη και του εκπομπού. Όταν ο μικροελεγκτής στέλνει λογικό 1 στην ψηφιακή έξοδο, η βάση του τρανζίστορ επιτρέπει τη ροή ρεύματος από τον συλλέκτη προς τον εκπομπό. Αντίθετα, όταν στέλνει λογικό 0, η ροή ρεύματος από τον συλλέκτη προς τον εκπομπό δεν επιτρέπεται. Έτσι, ο μικροελεγκτής μπορεί να ελέγχει τη συσκευή. Μεταξύ του συλλέκτη και της ψηφιακής εξόδου του μικροελεγκτή παρεμβάλλεται και μια αντίσταση 2200Ω. Η αντίσταση αυτή παίζει πολύ σημαντικό ρόλο, καθώς όταν ο μικροελεγκτής δεν στέλνει σήμα στην έξοδό του, διατηρεί τη τάση στον συλλέκτη σταθερή και χαμηλή. Έτσι, αποτρέπεται κάθε πιθανή επαφή του μικροελεγκτή με τάσεις και ρεύματα που θα μπορούσαν να επηρεάσουν αρνητικά τη λειτουργία του. Είναι ακόμα απαραίτητη και η χρήση μιας διόδου (1A) ανάμεσα στον θετικό και τον αρνητικό πόλο της συσκευής. Αυτό συμβαίνει γιατί, αν η συσκευή που τροφοδοτείται περιλαμβάνει ηλεκτρικό κινητήρα, όταν ο κινητήρας σταματήσει να λειτουργεί, υπάρχει πιθανότητα να επιστρέψει μια τάση με ανάστροφη πολικότητα για τα επόμενα 5-15 ms. Η διάδος είναι τοποθετημένη έτσι ώστε αυτή η αντίστροφη τάση να μην φτάσει στον συλλέκτη του τρανζίστορ, αποτρέποντας την καταστροφή τόσο του τρανζίστορ όσο και του μικροελεγκτή.

Πρόγραμμα κώδικα

Στο παρακάτω παράδειγμα η ηλεκτροβάνα ανοίγει και ρέει νερό ανάλογα με την τιμή της υγρασίας στο χώμα.

```
int sensorPin2 = A4; // soil moisture sensor pin
int sensorPin1 = A3; // photoresistor module pin
int watering=2;      // pin ηλεκτροβάνας
int water_value ;
int sun_Value1 = 0; // variable to store the value coming from the sensor
void setup()
```

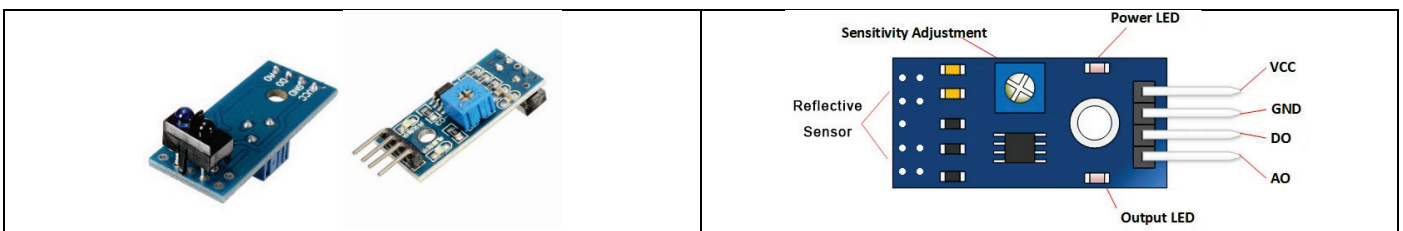
```
{
pinMode(watering,OUTPUT);
}
void loop()
{
water_value= analogRead(sensorPin1);           // watering
sun_Value1 = analogRead(sensorPin2);
water_value = map(water_value,350,1023,120,0);
sun_Value1 = map(sun_Value1,15,800,100,0);
if(sun_Value1<60 && (water_value<30 || water_value<70 ))
{digitalWrite(watering,HIGH); }
else
{digitalWrite(watering,LOW); }
Serial.println(water_value);
Serial.println(sun_Value1);
}
```

Κώδικας 4-17 Κώδικας Ηλεκτροβάνας

4.1.19 Αισθητήρας ανάκλασης TCRT5000

Ο TCRT5000 είναι ένας αισθητήρας ανάκλασης ο οποίος αποτελείται από έναν υπέρυθρο πομπό και ένα φωτοτρανζίστορ (δέκτης) σε μία κοινή συσκευασία, έτσι ώστε να μπλοκάρει το ορατό φως (δηλαδή το φως της ημέρας δεν τον επηρεάζει). Είναι μικρός σε μέγεθος (10.2 mm μήκος, 5.8mm πλάτος και 8mm ύψος) και το μήκος κύματος του εκπομπού είναι 950nm.

Αυτός ο ανακλαστικός αισθητήρας IR χρησιμοποιεί ένα TCRT5000 για την ανίχνευση του χρώματος και της απόστασης. Εκπέμπει IR και στη συνέχεια ανιχνεύει αν λαμβάνει την ηχώ. Αυτός ο αισθητήρας χρησιμοποιείται συχνά σε ρομπότ που ακολουθούν γραμμή, αυτόματη καταγραφή δεδομένων σε μετρητές κοινής ωφέλειας, επειδή αυτή η μονάδα μπορεί να αντιληφθεί αν μια επιφάνεια είναι λευκή ή μαύρη. Η απόσταση μέτρησης κυμαίνεται από 1 mm έως 8 mm και το κεντρικό σημείο είναι περίπου 2,5 mm. Η υπέρυθρη δίοδος θα εκπέμπει τις υπέρυθρες συνεχώς όταν η μονάδα συνδεθεί με την ισχύ, όταν το εκπεμπόμενο υπέρυθρο φως δεν έχει αντανακλάται ή η ισχύς δεν είναι αρκετά μεγάλη, η τρίοδος θα είναι σε κατάσταση απενεργοποίησης, αυτή τη στιγμή, η λογική έξοδος DO LOW και το σήμα υποδεικνύει LED off.

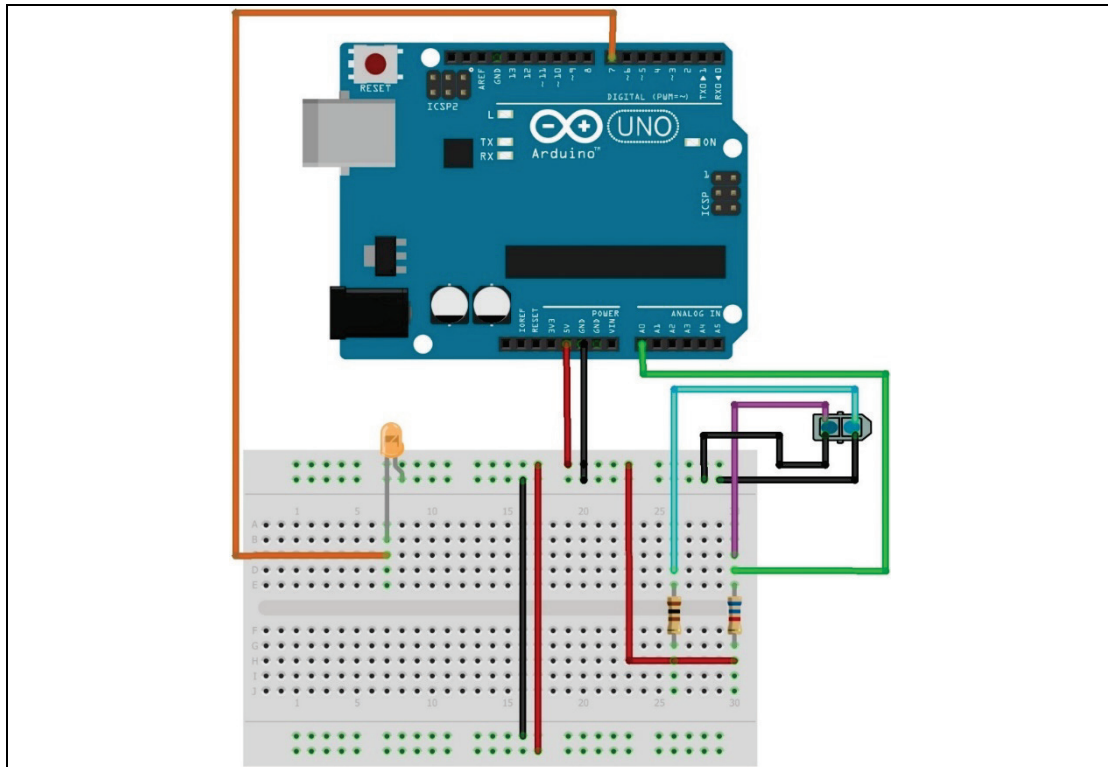


Εικόνα 4-38 Πλακέτα αισθητήρα ανάκλασης TCRT5000

(The most complete starterkit, n.d., instructables, n.d.)

)

Το κύκλωμα σύνδεσης του αισθητήρα ανάκλασης TCRT5000 είναι:



Εικόνα 4-39 Σύνδεση αισθητήρα ανάκλασης TCRT5000

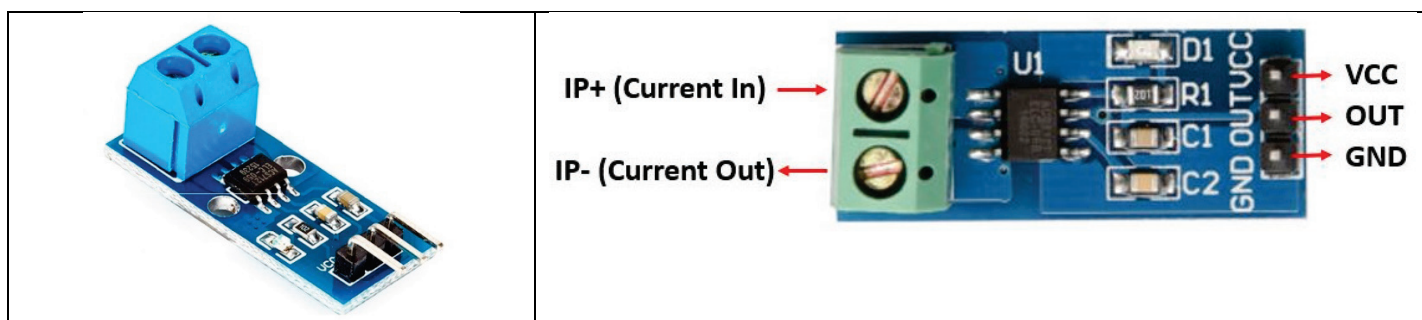
Πρόγραμμα κώδικα

```
int Led = 7; //Led στο pin 7
int input = A0; //Αισθητήρας TCRT5000 στο Pin A0
int value; //Ορισμός value
void setup ()
{
  pinMode (Led, OUTPUT);
  pinMode (input_pin, INPUT);
}
void loop ()
{
  value = analogRead (input);
  if (value > 410) //αν value>410 άναψε led
  {
    digitalWrite (Led, HIGH);
  }
}
```

Κώδικας 4-18 Κώδικας αισθητήρα ανάκλασης TCRT5000

4.1.20 Αισθητήρας ρεύματος, ACS712 Current sensor

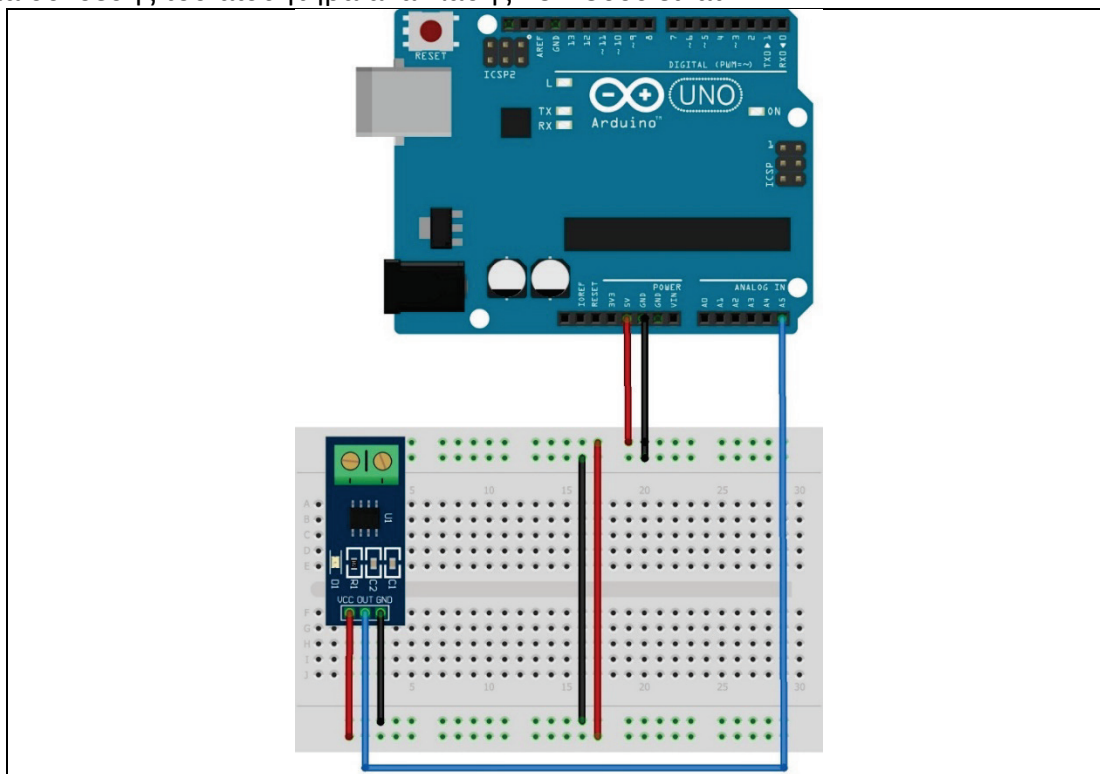
Ο αισθητήρας ρεύματος ACS712 5A είναι ένας πλήρως ενσωματωμένος γραμμικός αισθητήρας ρεύματος που βασίζεται στο φαινόμενο Hall. Παρέχει απομόνωση τάσης 2,1 kVRMS και κατασκευάζεται από την Allegro Microsystems. Το φαινόμενο Hall αναφέρεται στην εμφάνιση τάσης στα άκρα ενός λεπτού αγώγιμου φύλλου που διαρρέεται από ρεύμα και βρίσκεται μέσα σε μαγνητικό πεδίο. Στον αισθητήρα αυτό, η έξοδος εξαρτάται από το μαγνητικό πεδίο ή την πυκνότητα μαγνητικής ροής γύρω από τον μαγνητικό αισθητήρα. Μπορεί να ανιχνεύσει τόσο εναλλασσόμενο όσο και συνεχές ρεύμα, από -5A έως +5A (ACS712 Datasheet, 2022). Έχει τρεις ακίδες: μία για την τάση 5 Vdc, μία για τη γείωση και μία για το αναλογικό σήμα εξόδου. Η τάση εξόδου του αισθητήρα είναι 2,5 Vdc όταν δεν διαρρέεται από ρεύμα, και αυξάνεται κατά 185 mVdc για κάθε Ampere.



Εικόνα 4-40 Πλακέτα αισθητήρα ρεύματος, ACS712 Current sensor

(circuits-diy, n.d.)

Το κύκλωμα σύνδεσης του αισθητήρα ανάκλασης TCRT5000 είναι:



Εικόνα 4-41 Σύνδεση αισθητήρα ρεύματος, ACS712 Current sensor

Πρόγραμμα κώδικα

Μετά τη συλλογή ενός ικανοποιητικού αριθμού μετρήσεων από τον αισθητήρα, υπολογίζουμε τη μέση τιμή. Δεδομένου ότι τα 2,5 Vdc αντιστοιχούν σε μηδενικό ρεύμα και κάθε μεταβολή τάσης 185 mVdc αντιστοιχεί σε 1 A ρεύματος, μπορούμε να υπολογίσουμε την τιμή του ρεύματος και κατά συνέπεια την ισχύ.

```
void setup() {  
  Serial.begin(9600); //Εναρξη της σειριακής οθόνης για την εμφάνιση της  
                      //τρέχουσας τιμής ανάγνωσης στην οθόνη σειριακής οθόνης  
}  
  
void loop() {  
  unsigned int x=0;  
  float AcsValue=0.0;  
  float Samples=0.0;  
  float AvgAcs=0.0;  
  float AcsValueF=0.0;  
  double power=0;  
  
  for (int x = 0; x < 150; x++){ //Παίρνουμε 150 δείγματα  
    AcsValue = analogRead(A0); //Διαβάστε τις τρέχουσες τιμές του αισθητήρα  
    Samples = Samples + AcsValue; //Προσθέστε τα δείγματα μαζί  
    delay (3);  
  }  
  AvgAcs=Samples/150.0; //Μέση τιμή των δειγμάτων  
  
  //((AvgAcs * (5.0 / 1024.0)) μετατρέπει την τάση ανάγνωσης σε 0-5 βολτ.  
  //2.5 είναι το offset(υπέθεσα ότι το arduino λειτουργεί στα 5v οπότε το νιουτ σε μηδενικό ρεύ-  
  μα έρχεται  
  //out να είναι 2,5 που είναι το offset. Εάν το arduino σας λειτουργεί σε διαφορετική τάση από  
  την  
  //θα πρέπει να αλλάξετε το offset ανάλογα με την τάση εισόδου)  
  AcsValueF = (2.5 - (AvgAcs * (5.0 / 1024.0)) )/0.185;  
  
  Serial.print(AcsValueF);  
  delay(50);  
  power= AcsValueF * 220; // Υπολογισμός ισχύος  
  Serial.print("\total power= ");  
  Serial.println(power,2);  
  delay(2500);  
}
```

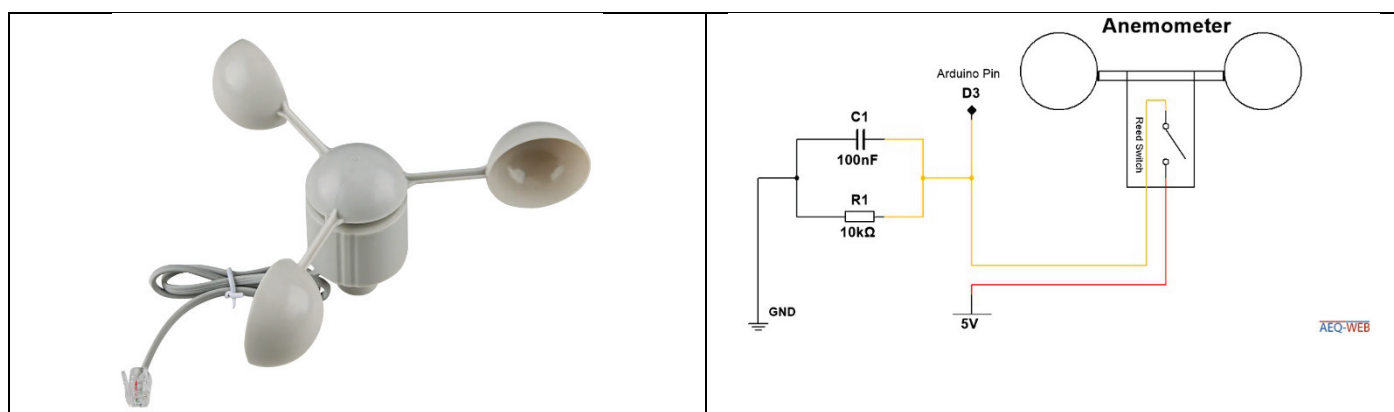
Κώδικας 4-19 Κώδικας αισθητήρα ρεύματος, ACS712 Current sensor

(engineersgarage, n.d.)

4.1.21 Ανεμόμετρο

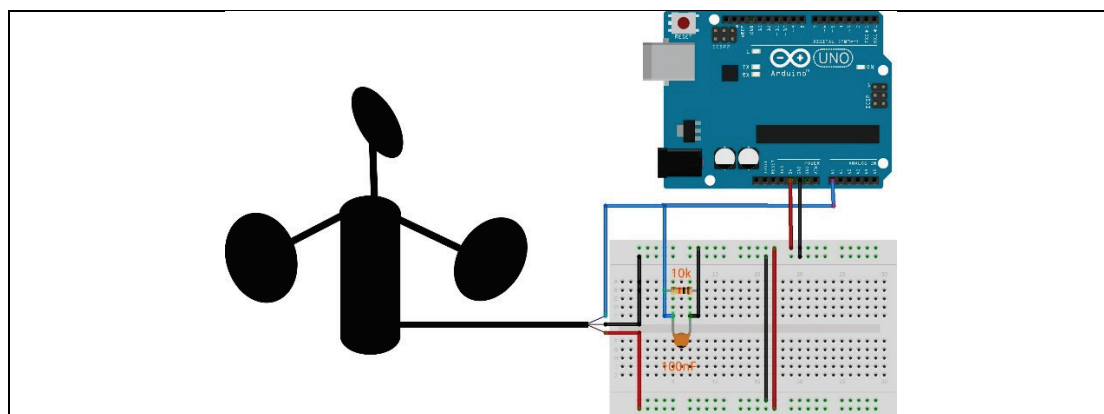
Το ανεμόμετρο είναι συσκευή μέτρησης της ταχύτητας του ανέμου. Ο τρόπος λειτουργίας του είναι πολύ απλός, αφού αποτελείται από έναν περιστρεφόμενο τροχό πάνω στον οποίο είναι στερεωμένες 3 "κού- πες" που κινούνται από τον άνεμο. Υπάρχει ένας μικρός μαγνήτης στον άξονα, ο οποίος περιστρέφεται πάντα μαζί του. Ως αισθητήρας χρησιμοποιείται μια μαγνητική επαφή, η οποία κλείνει το ηλεκτρικό κύ- κλωμα κάθε φορά που ο μαγνήτης περνάει από τον αισθητήρα.

Επειδή λειτουργεί ουσιαστικά ως ένας διακόπτης, το ανεμόμετρο συνδέεται στην είσοδο του μικροελε- γκτή με τον ίδιο τρόπο όπως ένα κουμπί, με μια αντίσταση των 10 KΩ και έναν παράλληλο πυκνωτή για να διασφαλιστεί ότι η είσοδος θα καταγράψει την υψηλή στάθμη του σήματος.



Εικόνα 4-42 Ανεμόμετρο
(GRobotronics, n.d., aeq-web, n.d.)

Το κύκλωμα σύνδεσης του ανεμόμετρου είναι:



Εικόνα 4-43 Σύνδεση ανεμόμετρου

Πρόγραμμα κώδικα

Μια στροφή ανά δευτερόλεπτο αντιστοιχεί σε ταχύτητα ανέμου 2,4 km/h (σύμφωνα με τον κατασκευα- στή). Ο κώδικας στοχεύει στη μέτρηση των μεμονωμένων παλμών στην ψηφιακή είσοδο μέσα σε ένα συ- γκεκριμένο χρονικό διάστημα και στη μετατροπή τους σε ταχύτητα ανέμου. Για την αποφυγή της συνε- χούς ανάγνωσης της ψηφιακής εισόδου, χρησιμοποιούνται διακοπές. Ο μετρητής παλμών μηδενίζεται πριν από κάθε μέτρηση. Έπειτα η διακοπή ενεργοποιείται στη συνδεδεμένη ακίδα σε λειτουργία RISING. Για να ενεργοποιηθεί ο ελεγκτής μία διακοπή θα πρέπει να συμβεί μια αλλαγή από LOW σε HIGH στην

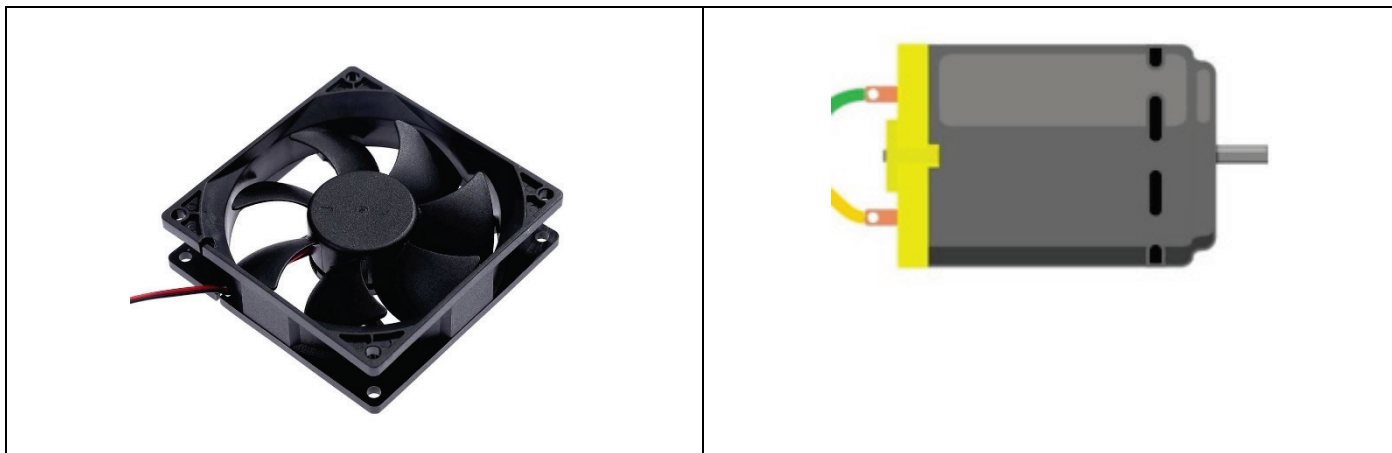
ψηφιακή είσοδο. Κάθε φορά που συμβαίνει μια διακοπή, ο μετρητής παλμών αυξάνεται κατά ένα. Μετά το πέρας του χρόνου μέτρησης, οι παλμοί μετατρέπονται σε ταχύτητα ανέμου και εμφανίζονται στην οθόνη.

```
const int timeWindow = 3; //Χρονικό παράθυρο
const int sensor = A0; //Ακίδα αισθητήρα ανέμου
int InterruptCounter; //Μετρητής διακοπών
float WindSpeed; // Μεταβλητή ταχύτητας
void setup(){
  Serial.begin(9600); // Εκκίνηση σειριακής οθόνης
}
void loop(){
  measure(); //Συνάρτηση μέτρησης ταχύτητας ανέμου
  Serial.print("Wind Speed= "); // Εκτύπωση μέτρησης
  Serial.print(speed); //Σε χιλιόμετρα ανά ώρα
  Serial.print(" km/h - ");
  Serial.print(WindSpeed / 3.6); //Σε μέτρα ανά δευτερόλεπτο
  Serial.println(" m/s");
}
void measure(){
  InterruptCounter = 0; // Αρχικοποίηση της ακίδας 2 σαν διακοπή που θα ενεργοποιείται
  attachInterrupt(digitalPinToInterrupt(sensor), counturpulse, RISING); // Μέτρηση παλμών
  delay(1000 * timeWindow); // Αναμονή για το τέλος του χρονικού παραθύρου
  detachInterrupt(digitalPinToInterrupt(sensor)); // Απενεργοποίηση διακοπών κατά τον υπολο-
  γισμό
  WindSpeed = (float)counter / (float)timeWindow * 2.4; // Υπολογισμός της ταχύτητας
}
void counturpulse() {
  InterruptCounter++; // Μετρητής παλμών
}
```

Κώδικας 4-20 Κώδικας ανεμόμετρου

(aeq-web, n.d.)

4.1.22 Ανεμιστήρας DC Fan



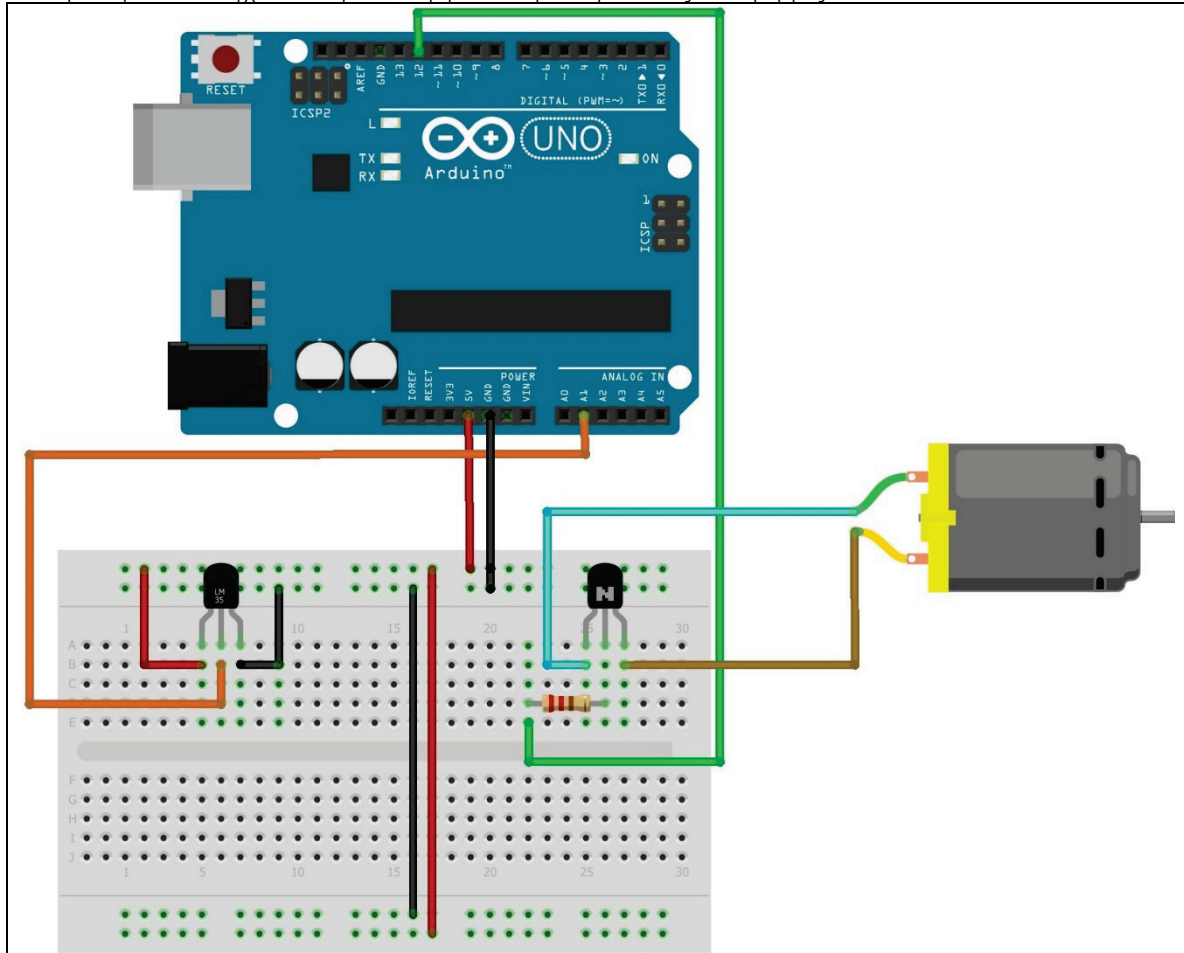
Εικόνα 4-44 Ανεμιστήρας DC

Τεχνικά χαρακτηριστικά DC μοτέρ :

- Μέγεθος κινητήρα: 38x20x15mm.
- Διάμετρος άξονα: 2mm.
- Αξονικό μήκος: 9mm.
- Τάση: 3-6V.
- Ρεύμα (στάση): 600mA.
- Ταχύτητα: 18000RPM.
- Ροπή: 0,55kg.cm.

Σύνδεση με το Arduino

Έλεγχος ανεμιστήρων PWM-DAC: συνδέστε έναν αισθητήρα LM35 ως αναλογική είσοδο στον ακροδέκτη A0 και συνδέστε έναν ανεμιστήρα ως έξοδο στον ακροδέκτη 13 και κατάλληλη παροχή ρεύματος 5v. όταν η θερμοκρασία αυξάνεται, ο ανεμιστήρας ανάβει και ο ανεμιστήρας πέφτει.



Εικόνα 4-45 Σύνδεση ανεμιστήρα DC Fan

Πρόγραμμα κώδικα

```
int fan = 12;
int tempPin = A1;
int temp;
int tempmin = 40;
int tempmax = 70;
int fanspeed;
void setup() {
  pinMode(fan, OUTPUT);
  pinMode(tempPin, INPUT);
  Serial.begin(9600);
}
void loop() {
  temp = readTemp();

  if(temp < tempmin)
  {
    fanspeed = 0;
    digitalWrite(fan, LOW);
  }
  if((temp >= tempmin) && (temp <= tempmax))
  {
    fanSpeed = map(temp,tempmin,tempmax,195,255);
```

```

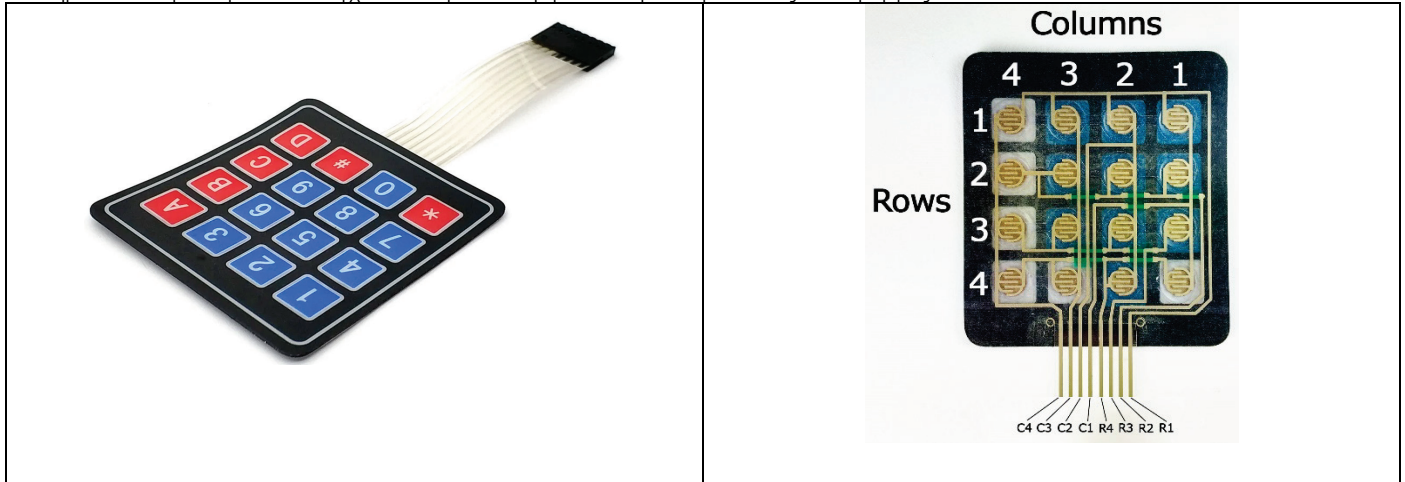
    analogWrite(fan, fanspeed);
  }
  Serial.print("Temp: ");
  Serial.print(temp);
  Serial.print("C\n");
  Serial.println();
  Serial.print("Fan: ");
  Serial.print(fanSpeed);
  Serial.print("");
  delay(200);
}
int readTemp()
{
  temp = analogRead(tempPin);
  return temp * 0.48828125;
}

```

Κώδικας 4-21 Κώδικας ανεμιστήρα DC Fan

4.1.23 Πληκτρολόγιο μεμβράνης, 4x4 Membrane Keypad

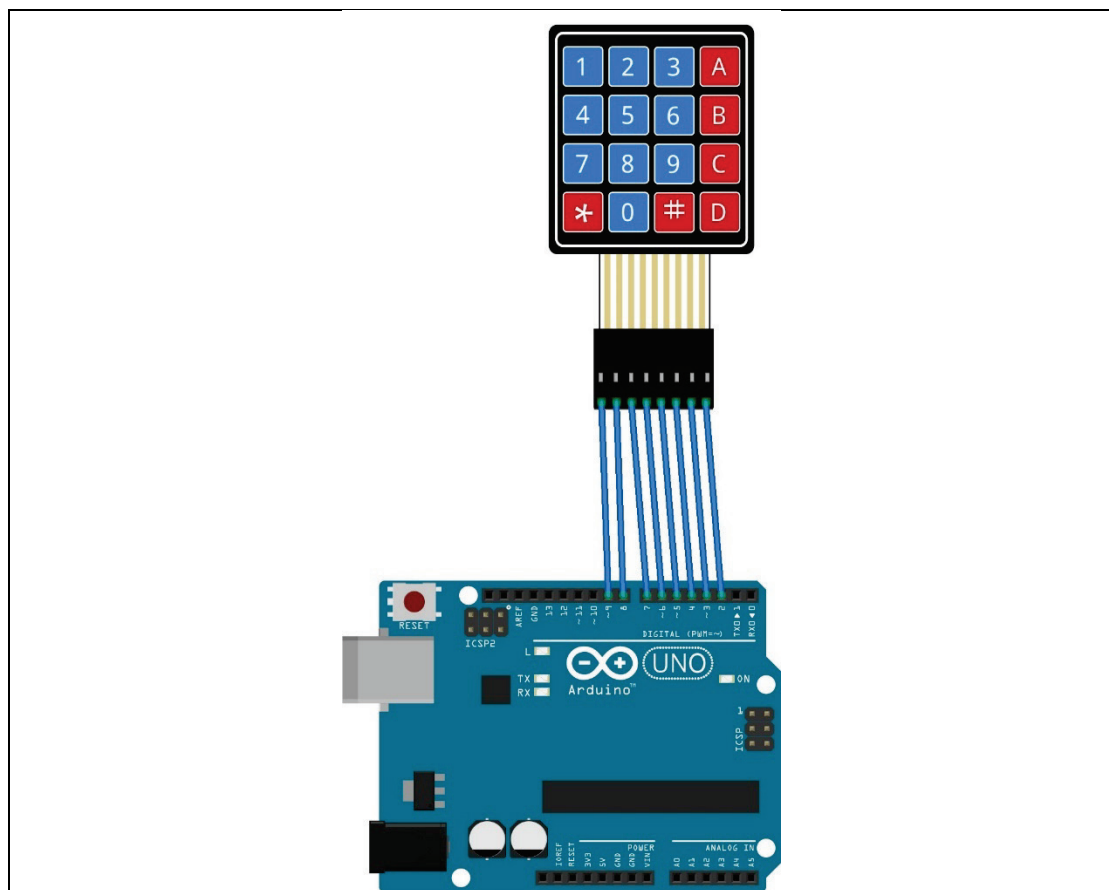
Τα πληκτρολόγια είναι ένας πολύ καλός τρόπος για να αφήσετε τους χρήστες να αλληλεπιδράσουν με το έργο σας. Μπορείτε να τα χρησιμοποιήσετε για την πλοήγηση σε μενού, την εισαγωγή κωδικών πρόσβασης και τον έλεγχο παιχνιδιών και ρομπότ. Τα πληκτρολόγια τύπου μεμβράνης είναι λεπτά και έχουν επίσης αυτοκόλλητη υποστήριξη, ώστε να μπορείτε να τα κολλήσετε στις περισσότερες επίπεδες επιφάνειες. Χρησιμοποιούνται σε διάφορους τύπους συσκευών, συμπεριλαμβανομένων και κλειδαριών θυρών. Τα κουμπιά σε ένα πληκτρολόγιο είναι τοποθετημένα σε σειρές και στήλες. Διατίθενται σε διάφορα μεγέθη, όπως το πληκτρολόγιο 4 γραμμών, 4 στηλών και 16 πλήκτρων (0-9, A-D, *, #), αλλά μόνο 8 ακίδων εξόδου. Έχουν παρόμοια διάταξη με αυτή ενός τυπικού πληκτρολογίου τηλεφώνου, καθιστώντας τα πολύ εύχρηστα. Κάτω από κάθε πλήκτρο υπάρχει ένας διακόπτης μεμβράνης. Κάθε διακόπτης σε μια σειρά συνδέεται με τους άλλους διακόπτες στη σειρά με ένα αγώγιμο ίχνος κάτω από το μαξιλάρι. Κάθε διακόπτης σε μια στήλη συνδέεται με τον ίδιο τρόπο - η μία πλευρά του διακόπτη συνδέεται με όλους τους άλλους διακόπτες της στήλης με ένα αγώγιμο ίχνος. Κάθε γραμμή και στήλη οδηγείται σε έναν μόνο ακροδέκτη, για συνολικά 8 ακροδέκτες σε ένα πληκτρολόγιο 4x4.



Εικόνα 4-46 Πληκτρολόγιο μεμβράνης, 4x4 Membrane Keypad

(The most complete starterkit, n.d., circuitbasics, n.d.)

Το κύκλωμα σύνδεσης του πληκτρολογίου μεμβράνης είναι:



Εικόνα 4-47 Σύνδεση πληκτρολογίου μεμβράνης, 4x4 Membrane Keypad

Πρόγραμμα κώδικα

```
#include <Keypad.h>
const byte Rows = 4; //Ορισμός γραμμών
const byte Columns = 4; //Ορισμός στηλών
//Ορισμός συμβόλων του πληκτρολογίου
char keys[Rows][Columns] = {
  {'1','2','3','A'},
```



```

{'4','5','6','B'},
{'7','8','9','C'},
{'*','0','#','D'}
};
// Ορισμός ακίδων πλήκτρων γραμμής
byte rowPins[Rows] = {9, 8, 7, 6};
// Ορισμός ακίδων πλήκτρων στήλης
byte columnsPins[Columns] = {5, 4, 3, 2};
// Δημιουργία αντικειμένου τύπου Keypad
Keypad mykeypad = Keypad( makeKeymap(keys), rowPins, columnsPins, Rows, Columns);
void setup()
{
  Serial.begin(9600); // Εκκίνηση της σειριακής θύρας
}
void loop()
{ // Μεταβλητή ελέγχου και αποθήκευσης χαρακτήρων
  char customKey = mykeypad.getKey();

  if (customKey){ // Αν πατήθηκε πλήκτρο
    Serial.println(customKey); // Εκτύπωσε τον χαρακτήρα
  }
}

```

Κώδικας 4-22 Κώδικας ηλεκτρολογίου μεμβράνης, 4x4 Membrane Keypad

4.1.24 Μαγνητικός διακόπτης πόρτας, Magnetic sensor switch

Ένα σετ μαγνητικού διακόπτη πόρτας είναι ένα σύστημα ειδικά σχεδιασμένο για να σας ειδοποιεί όταν ανοίγουν πόρτες, συρτάρια ή οποιοδήποτε άλλο άνοιγμα. Αυτοί οι τύποι διακοπών χρησιμοποιούνται κυρίως σε οικιακά συστήματα ασφαλείας. Το ένα μισό του συγκροτήματος τοποθετείται σε ένα πλαίσιο παραθύρου ή πόρτας και το άλλο συνδέεται με το ίδιο το παράθυρο ή την πόρτα. Όταν ένα σετ διακοπών διαχωρίζεται το ένα από το άλλο, η επαφή διακόπτεται και ενεργοποιεί έναν συναγερμό. Όταν κάθε κομμάτι μαγνήτη πλησιάσει το ένα το άλλο σε απόσταση 20 χιλιοστών, ολοκληρώνουν το κύκλωμα με τους εσωτερικούς διακόπτες reed. Κάθε σετ μαγνητικών διακοπών πόρτας μπορεί να δεχτεί μέγιστη τάση έως 100V στα 500mA. Επιπλέον, κάθε σετ στεγάζεται σε ανθεκτικό πλαστικό ABS και, με τις βίδες που περιλαμβάνονται, μπορεί να τοποθετηθεί εύκολα στις περισσότερες επιφάνειες.

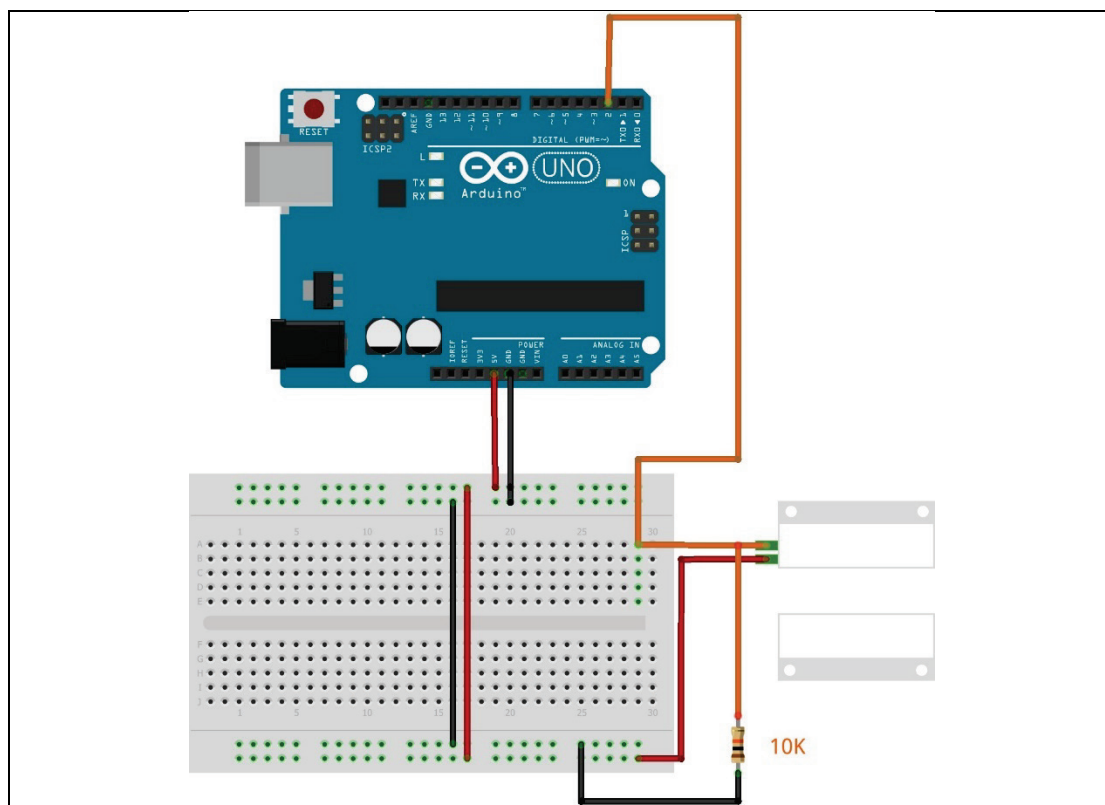


(The most complete starterkit, n.d., circuitbasics, n.d.)

Τεχνικά χαρακτηριστικά:

- Μέγιστη τάση: 100VDC
- Μέγιστο ρεύμα: 500mA
- Μέγιστη βαθμολογία: 10W
- Διάκενο: 20 5mm
- Λειτουργία επαφής: Λειτουργία επαφής: ΟΧΙ
- Διαστάσεις: 34x41x6.5 mm

Το κύκλωμα σύνδεσης του μαγνητικού διακόπτη είναι:



Εικόνα 4-49 Σύνδεση μαγνητικού διακόπτη πόρτας, Magnetic sensor switch

Πρόγραμμα κώδικα

```
const int switchPin = 2; // Ακίδα μαγνητικού διακόπτη
int switchVal = 0; // Μεταβλητή τιμής διακόπτη
const int led= 13; // Ακίδα ενδεικτικού led

void setup()
{
  pinMode(switchPin, INPUT); // Ορισμός ακίδας 2 ως είσοδο
  pinMode(led, OUTPUT); // Ορισμός ακίδας 13 ως έξοδο
}

void loop()
```

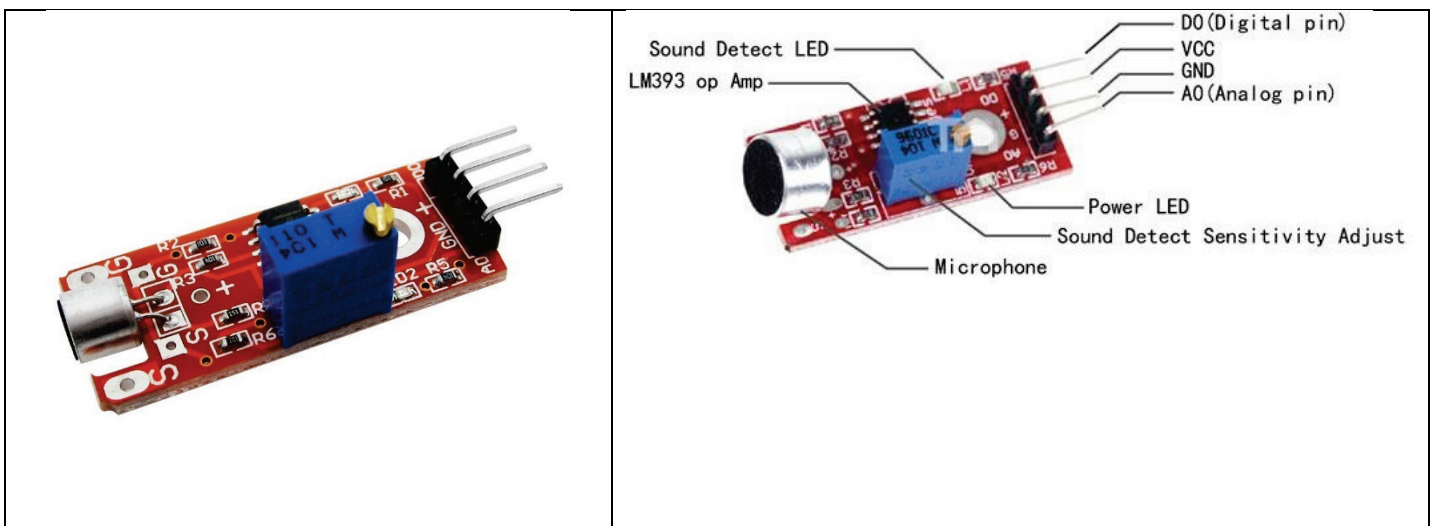
```
{
  // Διάβασμα εισόδων
  switchVal = digitalRead(switchPin);
  // Αν έχει πατηθεί
  if (switchVal == HIGH)
  {
    // Άναψε το ενδεικτικό led
    digitalWrite(led, HIGH);
  } else if (switchVal == LOW)
  {
    // Σβήσε το ενδεικτικό led
    digitalWrite(led, LOW);
  }
}
```

Κώδικας 4-23 Κώδικας μαγνητικού διακόπτη πόρτας, Magnetic sensor switch

4.1.25 Αισθητήρας ανίχνευσης ήχου, KY-038 Sound Detection Sensor

Ο αισθητήρας ανίχνευσης ήχου, αποτελεί έναν εύκολο τρόπο ανίχνευσης της έντασης του ήχου. Ο ήχος ανιχνεύεται μέσω μικροφώνου και τροφοδοτείται στην είσοδο ενός ενισχυτή, ενός ανιχνευτή παλμών αλλαγής και ενός καταχωρητή. Μέσω ποτενσιόμετρου ρυθμίζεται το σημείο της επιθυμητής στάθμης ήχου. Η μονάδα παρέχει δύο εξόδους:

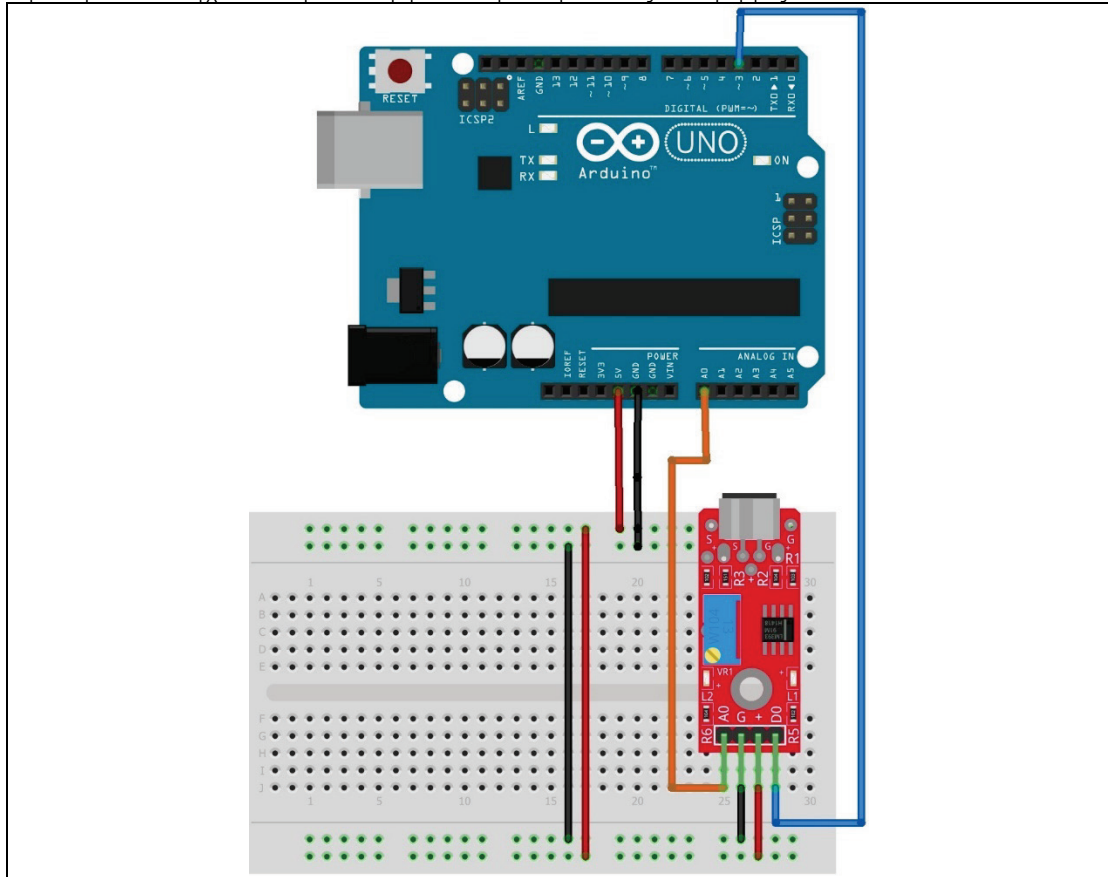
- D0: Όταν η στάθμη ήχου υπερβαίνει ή όχι το σημείο ρύθμισης, ενεργοποιείται η έξοδος σε High ή Low.
- A0: Μεταφράζει τον ήχο που ανιχνεύεται, σε σήμα τάσης εξόδου πραγματικού χρόνου



Εικόνα 4-50 Πλακέτα αισθητήρα ανίχνευσης ήχου, KY-038 Sound Detection Sensor

(The most complete starterkit, n.d., GRobotronics, n.d.)

Το κύκλωμα σύνδεσης του αισθητήρα ανίχνευσης ήχου είναι:



Εικόνα 4-51 Σύνδεση αισθητήρα ανίχνευσης ήχου, KY-038 Sound Detection Sensor

Πρόγραμμα κώδικα

```
int sensorAnalog = A0; //Δήλωση της ακίδας για την αναλογική έξοδο του αισθητήρα ήχου
int sensorDigital = 3; //Δήλωση της ακίδας για την ψηφιακή έξοδο του αισθητήρα ήχου
int analogVal = 0; //Δήλωση της μεταβλητής για την αποθήκευση της τιμής
int digitalVal; //Δήλωση της μεταβλητής για την αποθήκευση της τιμής
int led = 13; //Δήλωση του LED στην ακίδα 13 του Arduino
void setup()
{
  Serial.begin(9600);
  pinMode(sensorDigital,INPUT); // Ορισμός του pin 3 σε είσοδο
  pinMode(led,OUTPUT); // Ορισμός του pin A0 σε έξοδο
}
void loop()
{
  analogVal = analogRead(sensorAnalog); //Διάβασμα της τιμής της αναλογικής εισόδου
  digitalVal=digitalRead(sensorDigital); // Διάβασμα της τιμής της ψηφιακής εισόδου
  Serial.println(analogVal); //Εκτύπωση της αναλογικής τιμής του αισθητήρα
  if(digitalVal==HIGH) //Αν ενεργοποιηθεί ο αισθητήρας
  {
    digitalWrite(led,HIGH); //Οδήγησε την έξοδο 13 σε επίπεδο High
  }
  else
  {
    digitalWrite(led,LOW); //Αλλιώς οδήγησε την έξοδο 13 σε επίπεδο Low
  }
}
```

```
delay(50);  
}
```

Κώδικας 4-24 Κώδικας αισθητήρα ανίχνευσης ήχου, KY-038 Sound Detection Sensor

4.2 SHIELDS & ΑΛΛΕΣ ΠΛΑΚΕΤΕΣ ΕΠΙΚΟΙΝΩΝΙΑΣ

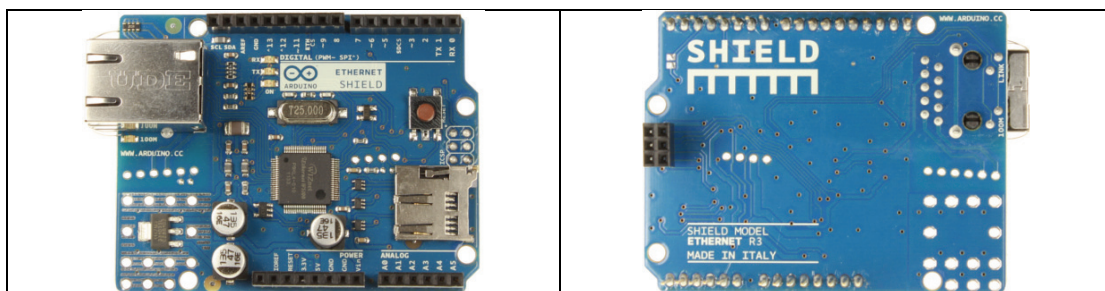
4.2.1 Γενικά

Τα Arduino shields είναι προ-κατασκευασμένες πλακέτες κυκλωμάτων, τα οποία συνδέονται πάνω σε συμβατές πλακέτες arduino. Παρέχουν πρόσθετες δυνατότητες όπως σύνδεση στο διαδίκτυο, έλεγχος κινητήρα, παροχή ασύρματης Wifi / ZigBee επικοινωνίας, GPS, Ethernet, έλεγχος οθονών και άλλα.

+++++πλακέτες επικοινωνίας

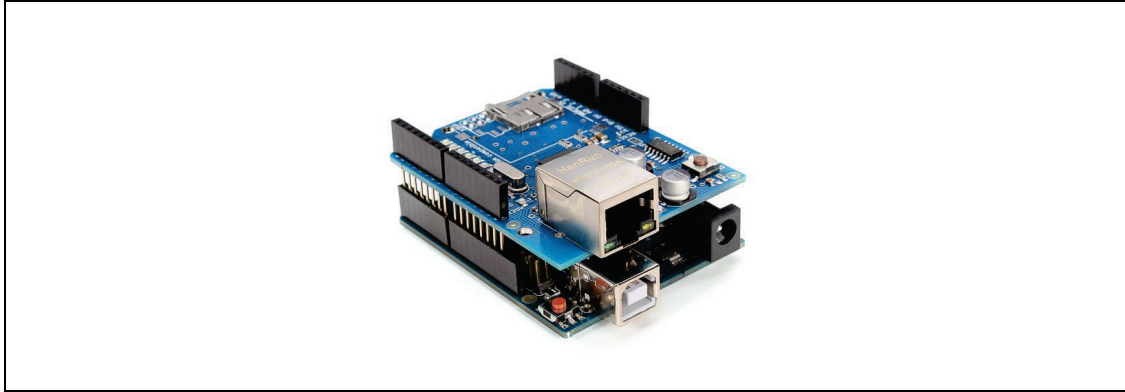
4.2.2 Ethernet shield

Το Ethernet shield δίνει την δυνατότητα ενσύρματης σύνδεσης του Arduino με το διαδίκτυο. Βασίζεται στο τσιπ W5100, το οποίο διαθέτει εσωτερική μνήμη 16K. Η ταχύτητα σύνδεσης μπορεί να φτάσει τα 10/100Mbps. Το shield χρησιμοποιεί την Arduino Ethernet βιβλιοθήκη, η οποία παρέχεται μαζί με το περιβάλλον ανάπτυξης. Επιπλέον, το shield διαθέτει μια ενσωματωμένη υποδοχή micro SD, επιτρέποντας την αποθήκευση μεγάλου όγκου δεδομένων ακόμα και για ολόκληρο δικτυακό τόπο. Ωστόσο, για τη χρήση της μνήμης SD απαιτείται η χρήση εξωτερικής βιβλιοθήκης, η οποία δεν περιλαμβάνεται στο λογισμικό. Το W5100 chip υποστηρίζει δικτύωση IP με TCP και UDP πρωτόκολλα, καθώς και μέχρι και τέσσερις ταυτόχρονες συνδέσεις. Τέλος, η κάρτα SD και το W5100 μοιράζονται το SPI bus του Arduino, οπότε δεν μπορούν να χρησιμοποιηθούν ταυτόχρονα, εκτός αν ελεγχθούν οι σχετικές βιβλιοθήκες. (πηγή: <https://el.wikipedia.org/wiki/Arduino> και <https://www.arduino.cc>)



Εικόνα 4-52 Πλακέτα Ethernet shield (2 πλευρές)

(GRobotronics, n.d.)



Εικόνα 4-53 Σύνδεση shield με arduino

(GRobotronics, n.d.)

Το Arduino επικοινωνεί τόσο με την κάρτα W5100 όσο και με την κάρτα SD μέσω του διαύλου SPI. Πιο, συγκεκριμένα:

- Στο Arduino Uno, η επικοινωνία SPI γίνεται μέσω των ψηφιακών ακίδων 11, 12 και 13.
- Στο Arduino Mega, η επικοινωνία SPI γίνεται μέσω των ακίδων 50, 51 και 52.

Και στις δύο πλακέτες:

- Ο ακροδέκτης 10 χρησιμοποιείται για την επιλογή της κάρτας W5100.
- Ο ακροδέκτης 4 χρησιμοποιείται για την επιλογή της κάρτας SD.

Αυτές οι ακίδες δεν μπορούν να χρησιμοποιηθούν ως γενικές ψηφιακές είσοδοι/έξοδοι.

Στο Arduino Mega, ο ακροδέκτης υλικού SS 53 δεν χρησιμοποιείται για την επιλογή είτε της W5100 είτε της κάρτας SD, αλλά πρέπει να διατηρείται ως έξοδος για να λειτουργεί η διεπαφή SPI σωστά. Διαθέτει πολλές ενδεικτικές λυχνίες LED. Οι λειτουργίες αυτών παρατίθενται παρακάτω:

- PWR: Ένδειξη τροφοδοσίας της πλακέτας.
- LINK: Ένδειξη της παρουσίας σύνδεσης στο δίκτυο. Αναβοσβήνει όταν λαμβάνονται ή διαβιβάζονται δεδομένα.
- FULLD: Ένδειξη της σύνδεσης αμφίδρομης σύνδεσης δικτύου
- 100M: Σύνδεση δικτύου 100 MB / s
- RX: Όταν η shield λαμβάνει δεδομένα αρχίζει να αναβοσβήνει
- TX: Όταν η shield στέλνει δεδομένα αρχίζει να αναβοσβήνει
- COLL: Αναβοσβήνει στην περίπτωση ανίχνευσης συγκρούσεων δικτύου

(πηγή: <https://el.wikipedia.org/wiki/Arduino> και <https://www.arduino.cc>)

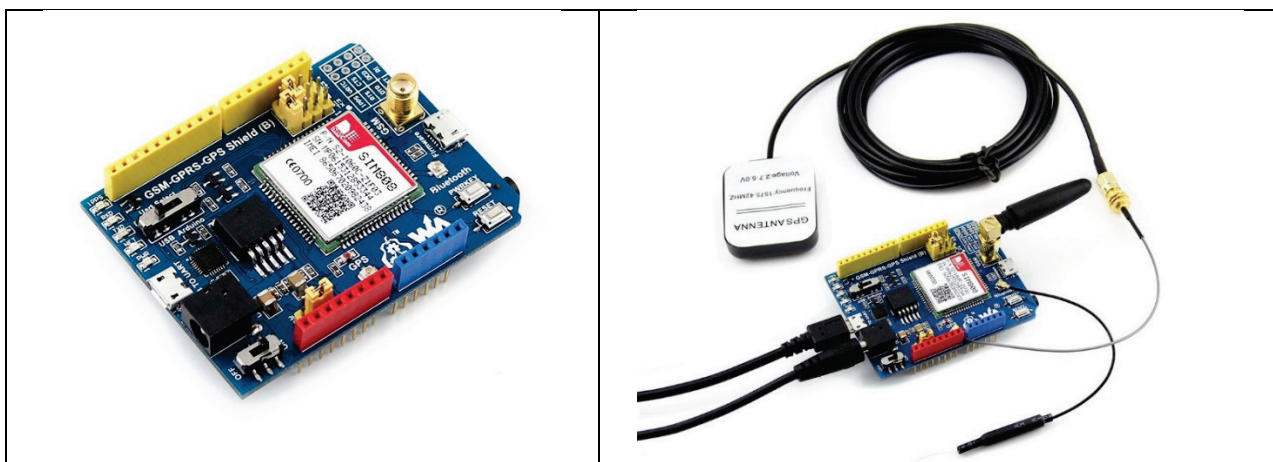
4.2.3 GSM/GPRS/GPS (SIM808) shield

4.2.3.1 Γενικά

Η πλακέτα GSM/GPRS/GPS είναι ένα shield Arduino που βασίζεται στη μονάδα GSM/GPRS/GPS τετραπλής ζώνης SIM808, παρέχοντας κυψελοειδή και εντοπισμό GPS, όλα σε μια μονάδα. Σας επιτρέπει να προσθέσετε εντοπισμό θέσης, φωνή, κείμενο, SMS και δεδομένα στο έργο σας. Αυτή η ασπίδα ταιριάζει ακριβώς

Συστήματα αυτοματισμού και ελέγχου σε κτιριακό περιβάλλον βασισμένα στις πλατφόρμες Arduino και ESP32

πάνω από το Arduino ή το συμβατό σας. Στην καρδιά του βρίσκεται μια ισχυρή μονάδα κινητής τηλεφωνίας GSM (χρησιμοποιούμε την πιο πρόσφατη SIM808) με ενσωματωμένο GPS. Αυτή η μονάδα μπορεί να κάνει σχεδόν τα πάντα. Η μονάδα SIM808 είναι μια μονάδα λειτουργίας GSM και GPS δύο σε ένα. Βασίζεται στη νεότερη μονάδα GSM/GPS SIM808 της SIMCOM και υποστηρίζει δίκτυο GSM/GPRS Quad-Band και συνδυάζει τεχνολογία GPS για δορυφορική πλοήγηση. Διαθέτει εξαιρετικά χαμηλή κατανάλωση ενέργειας σε κατάσταση αναστολής λειτουργίας και είναι ενσωματωμένη με κύκλωμα φόρτισης για μπαταρίες Li-Ion, που την κάνουν να έχει μεγάλο χρόνο αναμονής και είναι χρήσιμη για έργα που χρησιμοποιούν επαναφορτιζόμενη μπαταρία Li-Ion. Διαθέτει υψηλή ευαισθησία λήψης GPS με 22 κανάλια παρακολούθησης και 66 κανάλια δέκτη ac-quisition. Εκτός αυτού, υποστηρίζει επίσης το A-GPS που είναι διαθέσιμο για εντοπισμό σε εσωτερικούς χώρους. Η μονάδα ελέγχεται με εντολή AT μέσω UART και υποστηρίζει λογικό επίπεδο 3,3 VDC και 5 VDC. (πηγή: <https://copperhilltech.com/gsm-gprs-gps-sim808-shield-for-arduino/>)



Εικόνα 4-54 Πλακέτα με τον Sim808

(GRobotronics, n.d., copperhilltech, n.d.)

Τεχνικά Χαρακτηριστικά: (πηγή: <https://copperhilltech.com/gsm-gprs-gps-sim808-shield-for-arduino/>)

Γενικά

- Συνδεσιμότητα Arduino, συμβατή με UNO, Leonardo, NUCLEO, XNUCLEO
- Ενσωματωμένος μετατροπέας USB TO UART CP2102 για αποσφαλμάτωση UART
- 5 x LED για την ένδειξη της κατάστασης λειτουργίας της μονάδας
- Ενσωματωμένος μετατροπέας επιπέδου τάσης, υποστηρίζει συστήματα 3,3V και 5V
- Υποδοχή κάρτας SIM για κάρτα SIM 1,8V/3V
- Αυτόματη ανίχνευση ρυθμού Baudrate (1200bps ~115200bps)
- Bluetooth 3.0, υποστηρίζει τη μεταφορά δεδομένων μέσω Bluetooth
- RTC με διασύνδεση τροφοδοσίας
- Αναβαθμίσимо υλικολογισμικό μέσω USB
- Έλεγχος μέσω εντολών AT (3GPP TS 27.007, 27.005, και SIMCOM enhanced AT Commands)
- Υποστηρίζει εργαλειοθήκη εφαρμογών SIM: GSM 11.14 Release 99
- Τάση λειτουργίας: 6 ~ 12V
- Θερμοκρασία λειτουργίας: -40 °C ~ +85 °C
- Θερμοκρασία αποθήκευσης: -45 °C ~ +90 °C

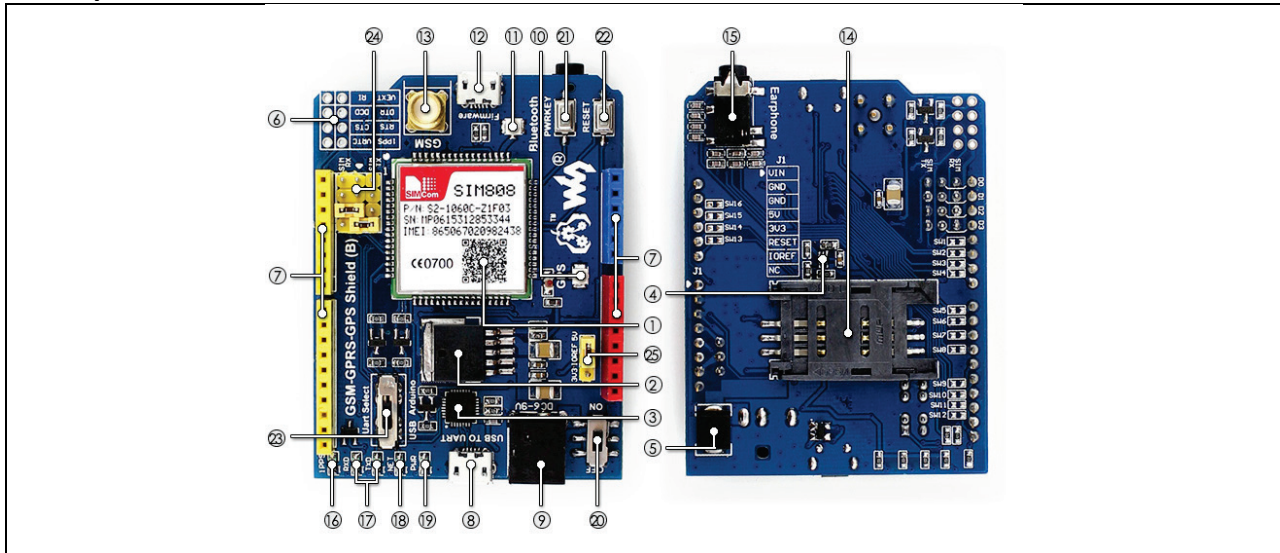
GSM/GPRS

- Μπάντα
 - GSM 850/EGSM 900/DCS 1800/PCS 1900 MHz
 - Αυτόματη αναζήτηση τετραπλής ζώνης
 - Συμβατό με τη φάση GSM 2/2+
- Ισχύς εκπομπής
 - Κατηγορία 4 (2W @ GSM 850/EGSM 900 MHz)
 - Κατηγορία 1 (1W @ DCS 1800/PCS 1900 MHz)
- Συνδεσιμότητα GPRS
 - Κλάση 12 πολλαπλών υποδοχών GPRS (προεπιλογή)
 - Κλάση πολλαπλών υποδοχών GPRS 1~12 (διαμορφώσιμη)
- Χαρακτηριστικό γνώρισμα δεδομένων GPRS
 - Ταχύτητα καθοδικής σύνδεσης: μέγιστο 85,6kbps
 - Ταχύτητα ανόδου: μέγιστο 85,6kbps
 - Σχήματα κωδικοποίησης: CS-1\CS-2\CS-3\CS-4
 - Υποστηρίζει PAP (πρωτόκολλο πιστοποίησης ταυτότητας κωδικού πρόσβασης) για σύνδεση PPP
 - Ενσωματωμένο πρωτόκολλο TCP/IP
 - Υποστηρίζει PBCCH
 - Υποστηρίζει ταχύτητα μετάδοσης CSD: 2,4/4,8/9,6/14,4 kbps
 - Υποστηρίζει USSD
- SMS
 - Υποστηρίζει: MT/MO/CB/Text/PDU mode
 - Αποθήκευση SMS: κάρτα SIM
- Ήχος
 - Λειτουργία κωδικοποίησης/αποκωδικοποίησης φωνής: Ενισχυμένος πλήρης ρυθμός \ Προσαρμοστικός πολλαπλός ρυθμός
 - Υποστηρίζει ακύρωση ηχούς
 - Υποστηρίζει μείωση θορύβου

GPS

- Τύπος δέκτη
 - 22 κανάλια παρακολούθησης
 - 66 κανάλια απόκτησης
 - Κώδικας C/A GPS L1
- Ευαισθησία
 - Παρακολούθηση: -165 dBm
 - Κρύες εκκινήσεις: -148 dBm
- Χρόνος έως την πρώτη διόρθωση
 - Ψυχρές εκκινήσεις: 30 δευτερόλεπτα (τυπ.)
 - Θερμές εκκινήσεις : < 1s
 - Θερμές εκκινήσεις: 28 δευτερόλεπτα
- Ακρίβεια
 - Οριζόντια θέση : <2.5m CEP

Board Components



Εικόνα 4-55 Πλακέτα με τον Sim808

(GRobotronics, n.d., copperhilltech, n.d.)

Επεξήγηση

1. SIM808 module
2. MIC29302 power chip
3. CP2102: USB TO UART converter
4. SMF05C: TVS diode
5. 1N5408: onboard rectifier
6. SIM808 functional pins
7. Arduino expansion connector
8. USB TO UART interface
9. DC power jack
10. GPS antenna connector
11. Bluetooth antenna connector
12. Firmware upgrade interface
13. GSM antenna connector
14. SIM card slot
15. 3.5mm earphone/mic jack
16. GPS status indicator
17. CP2102 UART Tx/Rx indicator
18. NET indicator:
 - flashes fast when the module starts up
 - flashes slowly after GSM register succeed
19. Power indicator
20. Power switch
21. SIM808 control button: press the button and hold for 1s, to startup/shutdown the SIM808
22. Reset button
23. UART selection switch, select controlling the SIM808 via:
 - CP2102
 - UART pins of Arduino interface
24. SIM808 UART configuration:
 - SIM_TX: SIM808 UART TX
 - SIM_RX: SIM808 UART RX
25. IOREF power selection: configure the UART voltage level

AT-Commands

Το SIM808 είναι ένα μικροελεγκτικό σύστημα που ελέγχεται μέσω των λεγόμενων AT-Εντολών. Αυτές οι εντολές αποτελούνται από μικρές σειρές κειμένου που μπορούν να συνδυαστούν για την εκτέλεση διαφόρων λειτουργιών. Η πλειοψηφία των πλακετών SIMCom υποστηρίζει αυτού του είδους τις εντολές. Για να ξεκινήσει η χρήση των AT-Εντολών, απαιτείται τροφοδοσία της μονάδας SIM808 με τάση 9-12V και ρεύμα τουλάχιστον 2A. Στη συνέχεια, μπορούν να χρησιμοποιηθούν οι πιο χρήσιμες εντολές για την εκάστοτε εργασία. (πηγή: [https://www.waveshare.com/wiki/GSM/GPRS/GPS_Shield_\(B\)](https://www.waveshare.com/wiki/GSM/GPRS/GPS_Shield_(B)))

AT-Commands GSM Επικοινωνίας

Ο ακόλουθος πίνακας περιέχει τις AT-Εντολές που σχετίζονται με τις λειτουργίες κινητής τηλεφωνίας, όπως πραγματοποίηση κλήσεων και αποστολή μηνυμάτων. Για την ορθή εκτέλεση ορισμένων εντολών, απαιτείται η προηγούμενη εκτέλεση άλλων εντολών, επομένως η σωστή ακολουθία είναι καθοριστική. Επιπλέον, σχεδόν όλες οι AT-Εντολές χρησιμοποιούν το «=1» στο τέλος τους για την εκτέλεση μιας λειτουργίας και το «=0» για την απενεργοποίησή της.

(πηγή: [https://www.waveshare.com/wiki/GSM/GPRS/GPS_Shield_\(B\)](https://www.waveshare.com/wiki/GSM/GPRS/GPS_Shield_(B)))

Γενικές εντολές κάρτας sim

Εντολή	Περιγραφή	Απάντηση	Επεξήγηση
AT+CPIN?	Ελέγχει την κατάσταση της κάρτας sim.	+CPIN:READY	Βρέθηκε η κάρτα sim
AT+CSQ	Ελέγχει την ποιότητα του σήματος	+CSQ:30,0	Μέγιστο επίπεδο είναι το 31 ικανό για τις βασικές λειτουργίες είναι το 10.
AT+COPS?	Ελέγχει τον πάροχο κινητής τηλεφωνίας	+COPS:0,0,"VODAFONE" or empty	Όνομα παρόχου
AT+CGMI	Ελέγχει τον δημιουργό της μονάδας.	SIMCOM_Ltd	Όνομα Κατασκευαστή
AT+CGMM	Ελέγχει το είδος της μονάδας	SIMCOM_SIM808	Στην περίπτωσή μας SIM808
AT+CGSN	Ελέγχει τον μοναδικό αριθμό imei	858830200504041	Αριθμός κάρτας sim.
AT+CNUM	Ελέγχει το νούμερο της κάρτας sim.	+CNUM:,"6949465421"	Αριθμός κάρτας
AT+ATE1	Ελέγχει αν η μονάδα λειτουργεί ή όχι	ATE0 ή ATE1	ON/OFF
ATD+number;	Καλεί το νούμερο που έχει δηλωθεί	OK	Κλήση
ATA	Απαντά σε κλήση	OK	-
ATH	Τερματίζει την τρέχουσα κλήση	OK	-
AT+COLP	Εμφανίζει τον αριθμό που καλεί η μονάδα	+COLP: "-----"	-
AT+CLIP(=1)	Εμφανίζει τον αριθμό που καλεί τη μονάδα	+CLIP: "-----"	-
AT+VTS=*	Θέτει ένα νούμερο για μία συγκεκριμένη απάντηση όπως γίνεται για παράδειγμα στους αυτόματους τηλεφωνητές	-	-

Πίνακας 4-3 AT-Commands κάρτας sim

Εντολές μηνύματος

Εντολή	Περιγραφή	Απάντηση	Επεξήγηση
AT+CNMI(2,1)	Καθορισμός θέσης αποθήκευσης μηνύματος	+CMTI:"SM",2	Όταν δεν είναι γεμάτη η θέση 2 αποθηκεύεται εκεί το μήνυμα
AT+CMGF(=1)	Ενεργοποιεί τη λειτουργία του μηνύματος	OK	-
AT+CSCS	Θέτει τον χαρακτήρα των μηνυμάτων ώστε να είναι στην αγγλική μορφή	OK	-

Συστήματα αυτοματισμού και ελέγχου σε κτιριακό περιβάλλον βασισμένα στις πλατφόρμες Arduino και ESP32

AT+CMGR(=1)	Διαβάζει το μήνυμα της πρώτης θέσης	-	-
AT+CMGS	Στέλνει το μήνυμα. » Πρέπει στο τέλος να προστεθεί το νούμερο 26 ώστε να καταλάβει η μονάδα ότι τερματίζεται το μήνυμα	">" «μήνυμα	-
AT+CMGD	Διαγράφει το μήνυμα όποιος θέσης υποδειχτεί	OK	-
AT+CPMS?	Ελέγχει τον χώρο αποθήκευσης	+CPMS:"SM",1,50	Σημαίνει ότι υπάρχει χώρος για 50 μηνύματα και αυτή τη στιγμή είναι ένα αποθηκευμένο

Πίνακας 4-4 AT-Commands SMS

AT-Commands GPRS

Ακολουθούν οι εντολές που είναι υπεύθυνες για την σύνδεση της μονάδας με το διαδίκτυο.

Εντολή	Απάντηση	Επεξήγηση
AT+CPIN?	+CPIN:READY,OK	Ελέγχει αν ο αριθμός pin που εισήχθη δίνει πρόσβαση στην κάρτα sim
AT+CREG?	+CREG:0,1.OK	Ελέγχει αν η κάρτα είναι καταχωρημένη στο δίκτυο
AT+CGATT=1	+CGATT:1,OK	Συνδέεται με το δίκτυο κινητής τηλεφωνίας
AT+CSST='APN'	OK	Δηλώνεται το αρη(ξεχωριστό για κάθε εταιρία κινητής) αν υπάρχει κωδικός προστίθεται και αυτός.
AT+CIICR	OK	Ξεκινά τη σύνδεση
AT+CIFSR	-	IP που έχει δεσμευθεί από τη πλακέτα
AT_CIPSTART="TCP","ιστότοπος","80"	OK,CONNECT,OK	Δηλώνεται ποιος ιστότοπος είναι ο στόχος αλλά και η πόρτα σύνδεσης.
AT+CIPSEND > ---	SEND OK	Στέλνει δεδομένα
AT+CIPCLOSE=1	OK	Κλείνει την TCP σύνδεση
AT+CIPSTATUS	-	Ελέγχει την κατάσταση σύνδεσης της μονάδας
AT+CIPSHUT	-	Κλείνει την επικοινωνία GPRS

Πίνακας 4-5 AT-Commands GPRS

AT-Commands GPS

Ακολουθεί ο πίνακας με τις εντολές που χρειάζεται να εκτελεστούν ώστε η πλακέτα να δεχθεί τα δεδομένα της κεραίας του GPS.

Εντολή	Απάντηση	Επεξήγηση
AT+CGPSPWR=1	-	Ενεργοποιεί την κεραία του GPS
AT+CGPSRST=1	-	Επανεκκινεί την κεραία
AT+CGPSIPR=9600	-	Θέτει την ταχύτητα(baudrate) σε 9600bps
AT+CGPSINF=32	Nmea string ,OK	Διαβάζει τα δεδομένα του GPS σε μορφή nmea
AT+CGPSSTATUS?	+CGPSSTATUS:LOCATION:""	Ελέγχει την κατάσταση του GPS

Πίνακας 4-6 AT-Commands GPS

4.2.4 Bluetooth HC-06 Module

Οι πλακέτες Bluetooth HC χρησιμοποιούνται με σκοπό να μετατρέψουν την σειριακή θύρα σε Bluetooth. Μπορούν να λειτουργήσουν και σαν Master και σαν Slave συσκευές.

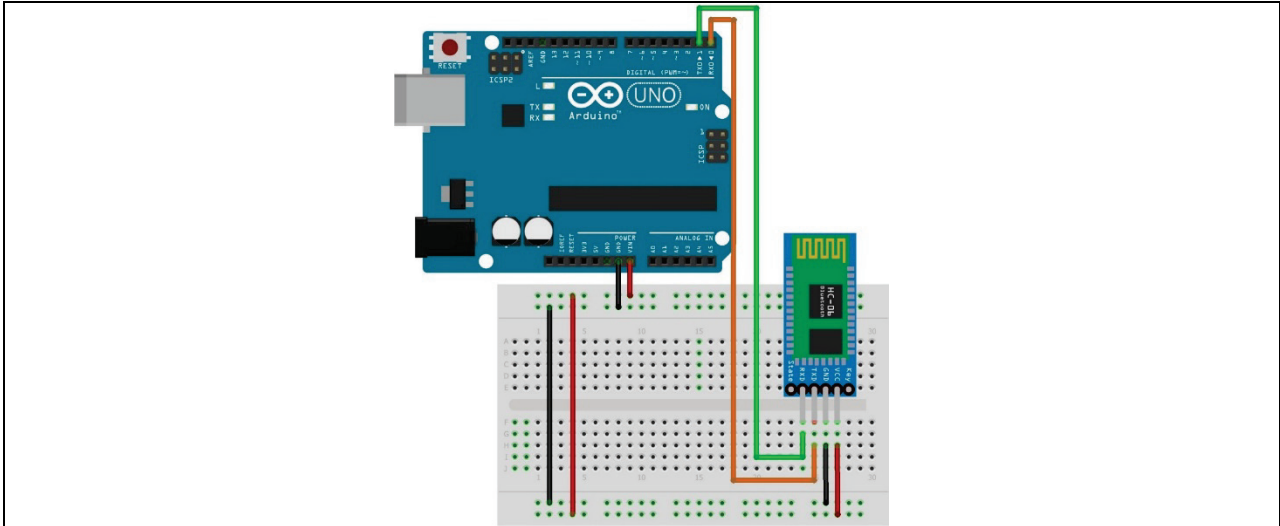
Χαρακτηριστικά:

- Διαφανής μεταφορά δεδομένων TTL μεταξύ μιας κεντρικής συσκευής Bluetooth.
- Λειτουργεί με οποιονδήποτε προσαρμογέα USB Bluetooth.
- Προεπιλεγμένος ρυθμός Baud: 9600 (8 N 1)
- Εμβέλεια έως 30ft.

Συστήματα αυτοματισμού και ελέγχου σε κτιριακό περιβάλλον βασισμένα στις πλατφόρμες Arduino και ESP32

- Ενσωματωμένη κεραία.
- Είσοδος ισχύος: 3.6-5.5V
- Μίνι μέγεθος.
- Διαστάσεις: (4,4 cm x 1,6 cm x 0,7 cm).
- Βάρος: 0,25 oz (7 g)

Σύνδεση με το Arduino:



Εικόνα 4-56 Σύνδεση Bluetooth HC-06 Module

Πρόγραμμα κώδικα

```
boolean LED_state[4] = {0};
int light_sensor = A0;
int ledstate = 0;
int boilerstate = 0;
int boiler = 34;
String voice;
void setup()
{
  Serial1.begin(9600);
}
void loop()
{
  ledstate = digitalRead(led);
  boilerstate = digitalRead(boiler);
  while (Serial1.available()) {
    delay(10);
    char k = Serial1.read();
    if (k == '#') {
      break;
    }
    voice += k;
  }
  if (voice.length() > 0) {
    if (voice == "*lights")
    {
```

```

ledstate = digitalRead(led);
ledstate = change_status(ledstate);
digitalWrite(led, ledstate);
voice = "";
}
if (voice == "*boiler")
{
boilerstate = digitalRead(boiler);
boilerstate = change_status(boilerstate);
digitalWrite(boiler, boilerstate);
voice = "";
}
voice = "";
}
}

```

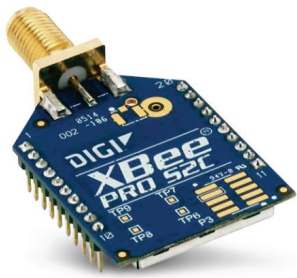
Κώδικας 4-25 Κώδικας Bluetooth HC-06 Module

4.3 XBEE S2

4.3.1 4.4.1 Εκδόσεις

Στην οικογένεια S2 ανήκουν 3 υλοποιήσεις XBee. Η πρώτη υλοποίηση είναι η S2 η οποία είναι η πιο οικονομική και παρέχει τις λιγότερες δυνατότητες. Η δεύτερη υλοποίηση είναι η S2B η οποία είναι πιο αναβαθμισμένη από την S2 και παρέχει κάποιες επιπλέον δυνατότητες. Τέλος, η S2C είναι η πιο ακριβή έκδοση και παρέχει τις περισσότερες δυνατότητες. Παρακάτω στο πίνακα 4-6 ακολουθεί συνοπτική περιγραφή των χαρακτηριστικών τους (*Knowledge Base | Digi International, n.d.*).

Έκδοση	Εικόνα	Χαρακτηριστικά
S2		Ορατότητα RF: 120 μ. Ισχύς Μετάδοσης: +3dBm Ευαισθησία λήψης: -96dBm Programmable Variant :N/A RF Data Rate: 250 Kbps

S2B		Ορατότητα RF: 1.5 χλμ. Ισχύς Μετάδοσης: +10dBm Ευαισθησία λήψης: -102dBm Programmable Variant : 32 KB Flash / 2 KB RAM RF Data Rate: 250 Kbps
S2C		Ορατότητα RF: 1.2 χλμ. Ισχύς Μετάδοσης: +8dBm Ευαισθησία λήψης: -102dBm Programmable Variant : 32 KB Flash / 2 KB RAM RF Data Rate: 250 Kbps

Πίνακας 4-7. Υλοποιήσεις XBee S2

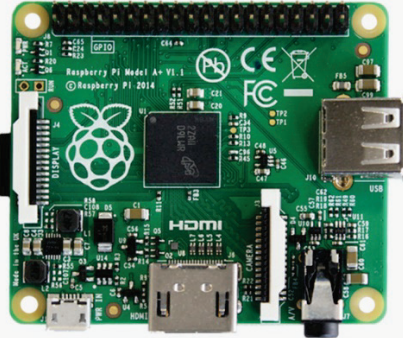
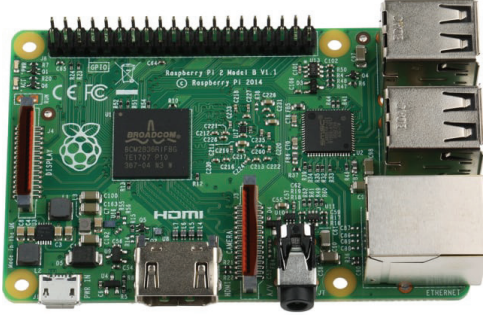
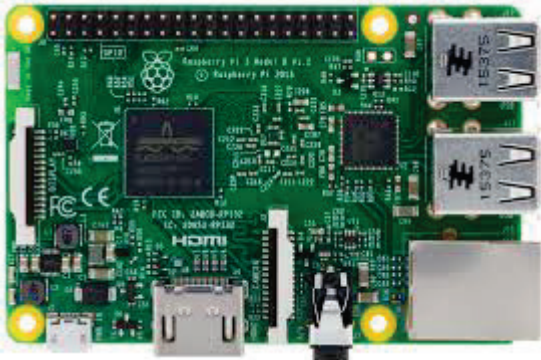
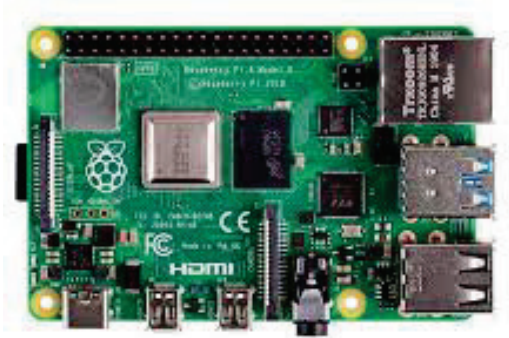
(The Major Differences in the XBee 802.15.4 Vs. The XBee ZB (S2) and (S2C) | Digi International, n.d.)

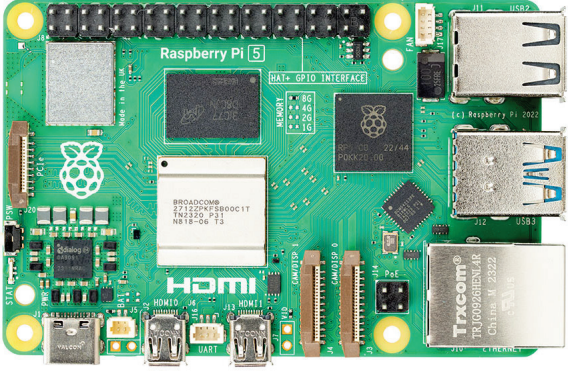
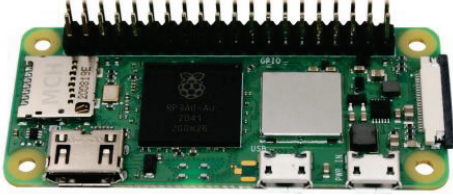
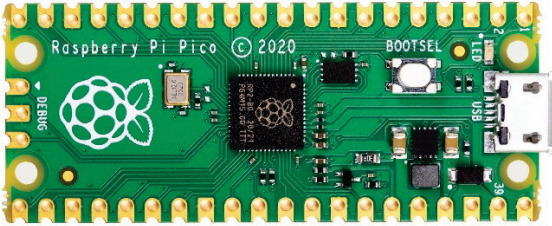
4.4 ΠΛΑΤΦΟΡΜΑ RASPBERRY-PI

4.4.1 Βασικές Εκδόσεις

Το Raspberry Pi είναι ένας υπολογιστής μεγέθους πιστωτικής κάρτας και είναι εξίσου ικανός όπως ένας οποιοσδήποτε σταθερός υπολογιστής. Υπάρχουν αρκετές εκδόσεις Raspberry Pi, στις οποίες ο αριθμός καθορίζει την παλαιότητα και το Model καθορίζει την οικονομική-πλούσια έκδοση. Συνεπώς το Raspberry Pi 1 ήταν το πρώτο που κατασκευάστηκε ενώ το Raspberry Pi 4 είναι το τελευταίο. Επίσης το Model A καθορίζει την ελαφριά έκδοση ενώ το Model B την πλούσια. Παρακάτω στο πίνακα 4-4 θα παρουσιαστούν μερικές από τις εκδόσεις.

Ονομασία	Εικόνα	Βασικά χαρακτηριστικά
----------	--------	-----------------------

1 Model A+		Επεξεργαστής: 700 MHz ARM11 RAM: 512 MB Θύρες USB: 1 Έξοδοι: HDMI, Audio jack Είσοδοι: 40 GPIO pins Διεπαφές: Camera, Display Micro SD
2 Model B		Επεξεργαστής: 900 MHz ARM Cortex-A7 RAM: 1GB Θύρες USB: 4 Έξοδοι: HDMI, Audio jack Είσοδοι: 40 GPIO pins Διεπαφές: Camera, Display Micro SD
3 Model B+		Επεξεργαστής: 1.4GHz Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit RAM: 1GB Θύρες USB: 4 Έξοδοι: HDMI, Audio jack Είσοδοι: 40 GPIO pins Διεπαφές: Camera, Display Micro SD Wi-Fi 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac Bluetooth: Bluetooth 4.2, BLE
4 Model B		Επεξεργαστής: 1.5GHz Broadcom BCM2711, Cortex-A72 (ARM v8) 64-bit Μνήμη: 1GB – 8GB Θύρες USB: 2 Θύρες USB 3.0: 2 Έξοδοι: 2x micro HDMI, Audio jack Είσοδοι: 40 GPIO pins Διεπαφές: Camera, Display Micro SD Wi-Fi 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac Bluetooth: Bluetooth 4.2, BLE

5 Model B		Επεξεργαστής : 2.4GHz quad-core 64-bit Arm Cortex-A76 CPU, with cryptography extensions, 512KB per-core L2 caches, and a 2MB shared L3 cache Μνήμη : 4GB – 8GB Θύρες USB: 2 Θύρες USB 3.0: 2 Έξοδοι: 2x micro HDMI, Audio jack Είσοδοι: 40 GPIO pins Διεπαφές: Camera, Display Micro SD Dual-band 802.11ac Wi-Fi®, Bluetooth 5.0/Bluetooth Low Energy (BLE)
Raspberry Pi Zero W		CPU: 1GHz ARM 11 RAM: 512 MB Θύρες USB: 1 Έξοδοι: mini HDMI, Audio jack Είσοδοι: 40 GPIO pins Διεπαφές: Camera, Micro SD Wi-Fi: 2.4GHz 802.11n Bluetooth: Classic 4.1 και Bluetooth LE
Raspberry Pi Pico		Chip μικροελεγκτή RP2040 Διπύρηνος επεξεργαστής Arm Cortex-M0+, με ευέλικτο ρολόι έως 133 MHz 264kB on-chip SRAM 2MB ενσωματωμένο φλας QSPI Ασύρματο τοπικό δίκτυο 2,4 GHz 802.11n Bluetooth 5.2 26 ακροδέκτες GPIO πολλαπλών λειτουργιών, συμπεριλαμβανομένων 3 αναλογικών εισόδων 2 × UART, 2 × ελεγκτές SPI, 2 × ελεγκτές I2C, 16 × κανάλια PWM 1 × USB 1.1 8 × προγραμματιζόμενες μηχανές κατάστασης I/O (PIO)

Πίνακας 4-8. Εκδόσεις Raspberry Pi

(Raspberry Pi Documentation, n.d.)

4.4.2 Λειτουργικό σύστημα Raspbian

Οι επεξεργαστές ARM υποστηρίζονται από το GNU Linux, από λειτουργικά συστήματα συμβατά με Unix όπως το FreeBSD και το NetBSD, καθώς και από ορισμένες εκδόσεις του Windows for IoT. Έτσι, το Raspberry μπορεί να λειτουργήσει με αυτά τα λειτουργικά συστήματα. Το Raspbian περιλαμβάνει όλες τις απαραίτητες βιβλιοθήκες και τους drivers για το υλικό του Raspberry, χωρίς να απαιτείται επιπλέον εγκατάσταση λογισμικού. Περιλαμβάνει IDEs, εφαρμογές γραφείου, δυνατότητες απομακρυσμένου ελέγχου και πίνακα ελέγχου για το υλικό του Raspberry. Επίσης, προσφέρει διαχείριση πακέτων Debian που περιλαμβάνει όλες τις εφαρμογές που μπορούν να τρέξουν σε αρχιτεκτονική ARM, διευκολύνοντας έτσι την αναζήτηση λογισμικού.

ΚΕΦΑΛΑΙΟ 5

ΕΝΔΕΙΚΤΙΚΑ ΣΕΝΑΡΙΑ ΑΥΤΟΜΑΤΙΣΜΟΥ & ΕΛΕΓΧΟΥ ΣΕ ΚΤΙΡΙΑΚΟ ΠΕΡΙΒΑΛΛΟΝ

5.1 ΓΕΝΙΚΑ

Το κεφάλαιο αυτό αποτελείται από ενδεικτικά σενάρια αυτοματισμού και ελέγχου σε κτιριακό περιβάλλον. Θα αναλυθεί η χρησιμότητα του κάθε σεναρίου στην καθημερινότητα και ο κώδικας που χρησιμοποιήθηκε για κάθε σενάριο ξεχωριστά. Σε όλα τα σενάρια προσθέτοντας ένα ethernet shield ή wifi shield και τον κατάλληλο κώδικα, μπορεί να έχουμε έλεγχο μέσω του διαδικτύου, αποστολή δεδομένων αισθητήρων ή ενεργοποιητών σε βάση δεδομένων και δημιουργία ιστοσελίδων με γραφήματα (πραγματικού χρόνου ή ιστορικά δεδομένα) και ιστορικά στατιστικά.

Επιλέχθηκαν μερικά ενδεικτικά σενάρια, που μπορεί να συναντήσουμε σε ένα οικιακό περιβάλλον. Πολύ εύκολα μπορεί να συνδυαστούν μεταξύ τους ή να αλληλοεπιδράσουν μεταξύ τους.

Ενδεικτικά σενάρια

- Σενάριο ελέγχου φωτισμού
- Σενάριο ελέγχου οικιακών συσκευών
- Σενάριο αυτόματης άντλησης νερών
- Σενάριο ανίχνευσης φωτιάς ή πλημμύρας
- Σενάριο ελέγχου πόρτας γκαράζ (ή κάποιας άλλης πόρτας) – ηχητική ή οπτική ένδειξη κατάστασης της πόρτας
- Σενάριο ανίχνευσης ποσοστού βροχής
- Σενάριο ελέγχου θερμοκρασίας μέσω κλιματισμού και ελέγχου ποιότητας αέρα

- Σενάριο ανοίγματος εξωτερικής πόρτας με δύο βαθμίδες ασφάλειας (αισθ. αφής και κωδικός)
- Σενάριο αυτόματου ποτίσματος γλαστρών σε μπαλκόνι
- Σενάριο αυτοματισμού κήπου
- Σενάριο ελέγχου εξωτερικών ρολών και ελέγχου φωτισμού εξωτερικού χώρου
- Σενάριο ελέγχου με φωνητικές εντολές (μέσω Bluetooth)
- Σενάριο μέτρησης κατανάλωσης (με μετρητή τάσης και ρεύματος)
- Σενάριο ανίχνευσης τοποθεσίας κατοικίδιου ή ατόμου με ειδικές ανάγκες, κουμπιού έκτακτου ανάγκης και συνθηκών περιβάλλοντος χώρου
- Σενάριο κατασκευής σταθμού καιρικών συνθηκών (weather station)

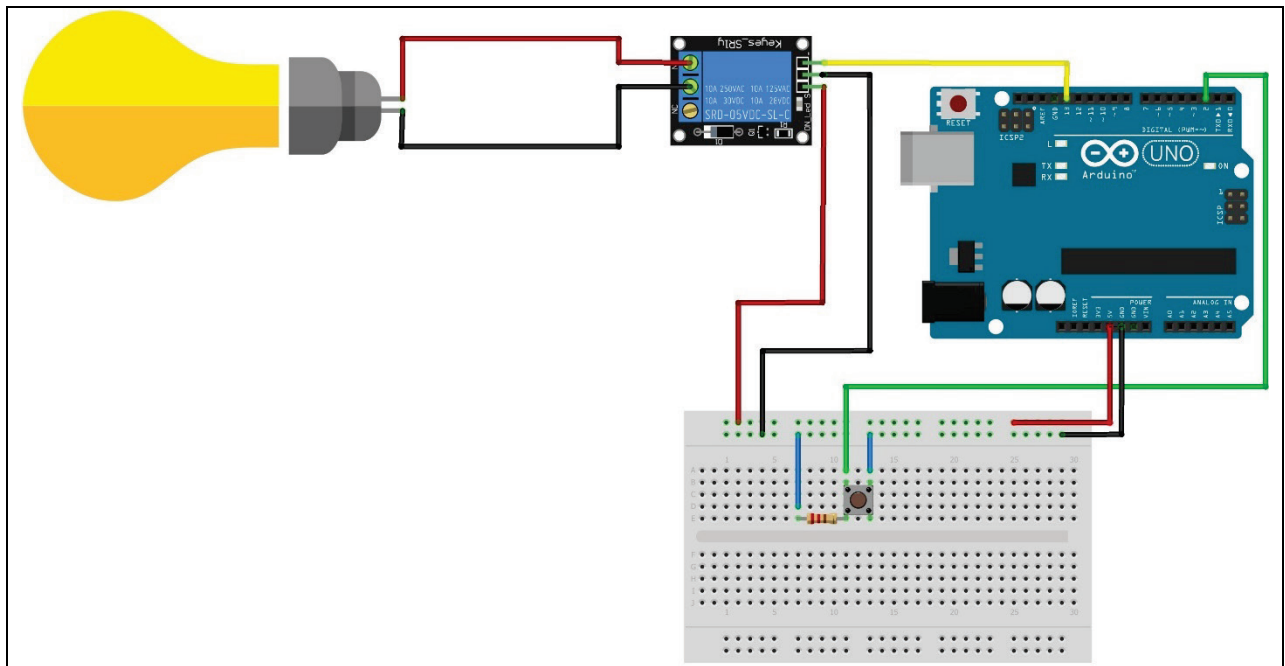
5.2 ΣΕΝΑΡΙΟ ΕΛΕΓΧΟΥ ΦΩΤΙΣΜΟΥ

Σε αυτό το σενάριο θα ελεγχθεί ένα φωτιστικό σώμα χρησιμοποιώντας έναν διακόπτη.

Η λάμπα ενεργοποιείται και απενεργοποιείται όταν πιέζεται ένα μπουτόν. Η λάμπα είναι συνδεδεμένη μέσω ενός ρελέ στον ψηφιακό ακροδέκτη 13, και το μπουτόν συνδέεται στον ακροδέκτη 2.

Τα υλικά που χρησιμοποιήθηκαν είναι:

- Arduino uno
- Λάμπα 240V light bulb with lamp holder 240V/AC
- 5V Relay
- AC power supply
- Button



Εικόνα 5-1 Σενάριο ελέγχου φωτισμού

Ο κώδικας για το σενάριο ελέγχου φωτισμού (τρέχει σε Arduino του έξυπνου σπιτιού):

```
const int PushbuttonPin = 2; //Ορισμός διακόπτη pin 2
const int LampPin = 13; //Ορισμός φωτιστικού pin 13
int PushbuttonState = 0; //Κατάσταση διακόπτη (pushbutton)
void setup() {
  pinMode(LampPin, OUTPUT);
  pinMode(PushbuttonPin, INPUT);
  digitalWrite(LampPin, LOW);
}
void loop(){
  PushbuttonState = digitalRead(PushbuttonPin); //Διάβασε την κατάσταση του διακόπτη (pushbutton)
  if (PushbuttonState == HIGH) { //Εάν η κατάσταση του διακόπτη (pushbutton) είναι HIGH
    ανάψε το φωτιστικό
    digitalWrite(LampPin, HIGH);
  }
  if (PushbuttonState == LOW) { // Εάν η κατάσταση του διακόπτη (pushbutton) είναι LOW
    ανάψε το φωτιστικό
    digitalWrite(LampPin, LOW);
  }
}
```

Κώδικας 5-1 Κώδικας σεναρίου ελέγχου φωτισμού

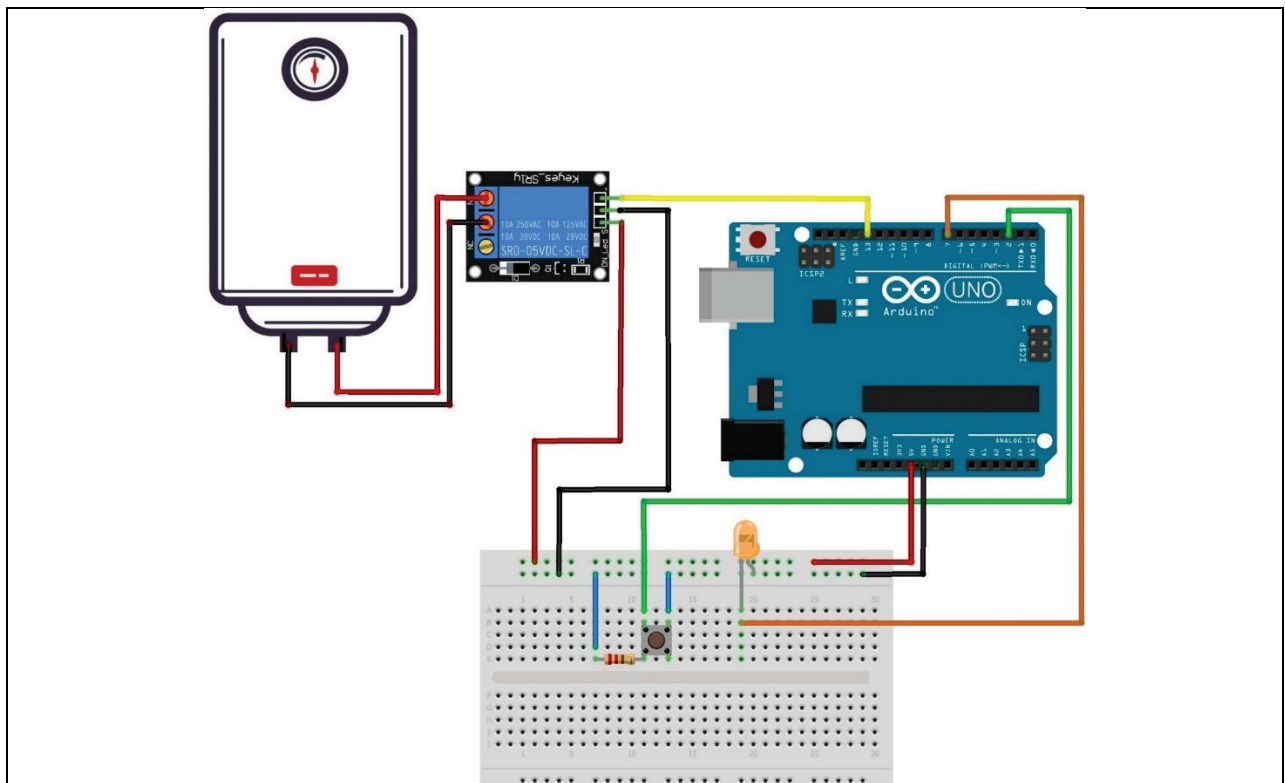
5.3 ΣΕΝΑΡΙΟ ΕΛΕΓΧΟΥ ΟΙΚΙΑΚΩΝ ΣΥΣΚΕΥΩΝ

Σε αυτό το σενάριο θα ελεγχθεί μία οικιακή συσκευή (π.χ. έναν θερμοσίφωνα) χρησιμοποιώντας έναν διακόπτη. Το κύκλωμα το οποίο κατασκευάστηκε για τη δημιουργία του συστήματος ελέγχου μίας οικιακής συσκευής αποτελείται από τα εξής:

- Arduino uno
- Οικιακή συσκευή (π.χ. Θερμοσίφωνα) 240V/AC
- 5V Relay
- AC power supply
- Διακόπτης (Button)
- Led

Το συγκεκριμένο σενάριο δεν χρειάζεται κάποια βιβλιοθήκη για την λειτουργία του.

Η οικιακή συσκευή (θερμοσίφωνα) ενεργοποιείται και απενεργοποιείται όταν πιέζεται ένα μπουτόν. Η οικιακή συσκευή (θερμοσίφωνα) είναι συνδεδεμένη μέσω ενός ρελέ στον ψηφιακό ακροδέκτη 13, και το μπουτόν συνδέεται στον ακροδέκτη 2. Όταν η οικιακή συσκευή (θερμοσίφωνα) είναι ενεργοποιημένη ένα led ανάβει. Το led συνδέεται στον ακροδέκτη 7.



Εικόνα 5-2 Σενάριο ελέγχου οικιακής συσκευής

Ο κώδικας για το σενάριο ελέγχου οικιακής συσκευής (τρέχει σε Arduino του έξυπνου σπιτιού):

```
const int PushbuttonPin = 2; //Ορισμός διακόπτη pin 2
const int HeaterPin = 13; //Ορισμός Θερμοσίφωνα Boiler pin 13
const int LedPin = 7; //Ορισμός LED pin 7
int PushbuttonState = 0; //Κατάσταση διακόπτη pushbutton
```

```
void setup() {
  pinMode(HeaterPin, OUTPUT);
  pinMode(LedPin, OUTPUT);
  pinMode(PushbuttonPin, INPUT);
  digitalWrite(HeaterPin, LOW);
}
void loop(){
  PushbuttonState = digitalRead(PushbuttonPin); //Διάβασε την κατάσταση του διακόπτη
  pushbutton
  if (PushbuttonState == HIGH) { //Εάν η κατάσταση του διακόπτη pushbutton είναι HIGH
    άναψε τον Θερμοσίφωνα Boiler και το led
    digitalWrite(HeaterPin, HIGH);
    digitalWrite(LedPin, HIGH);
  }
  if (PushbuttonState == LOW) { //Εάν η κατάσταση του διακόπτη pushbutton η είναι LOW
    σβήσε τον Θερμοσίφωνα Boiler και το led
    digitalWrite(HeaterPin, LOW);
    digitalWrite(LedPin, LOW);
  }
}
```

Κώδικας 5-2 Κώδικας σεναρίου οικιακής συσκευής

5.4 ΣΕΝΑΡΙΟ ΑΥΤΟΜΑΤΗΣ ΑΝΤΛΗΣΗΣ ΝΕΡΩΝ

Η αυτόματη αντλία νερού είναι πολύ χρήσιμη για την απομάκρυνση των υδάτων που συσσωρεύονται, βοηθώντας στην πρόληψη πλημμυρών σε διάφορους χώρους του σπιτιού, όπως τα φρεάτια, οι χώροι των ανελκυστήρων, τα υπόγεια, κ.ά. Το πρόβλημα αυτό μπορεί να επιδεινωθεί σημαντικά κατά τη διάρκεια των χειμερινών μηνών λόγω των έντονων βροχοπτώσεων.

Η λειτουργία του σεναρίου είναι η εξής:

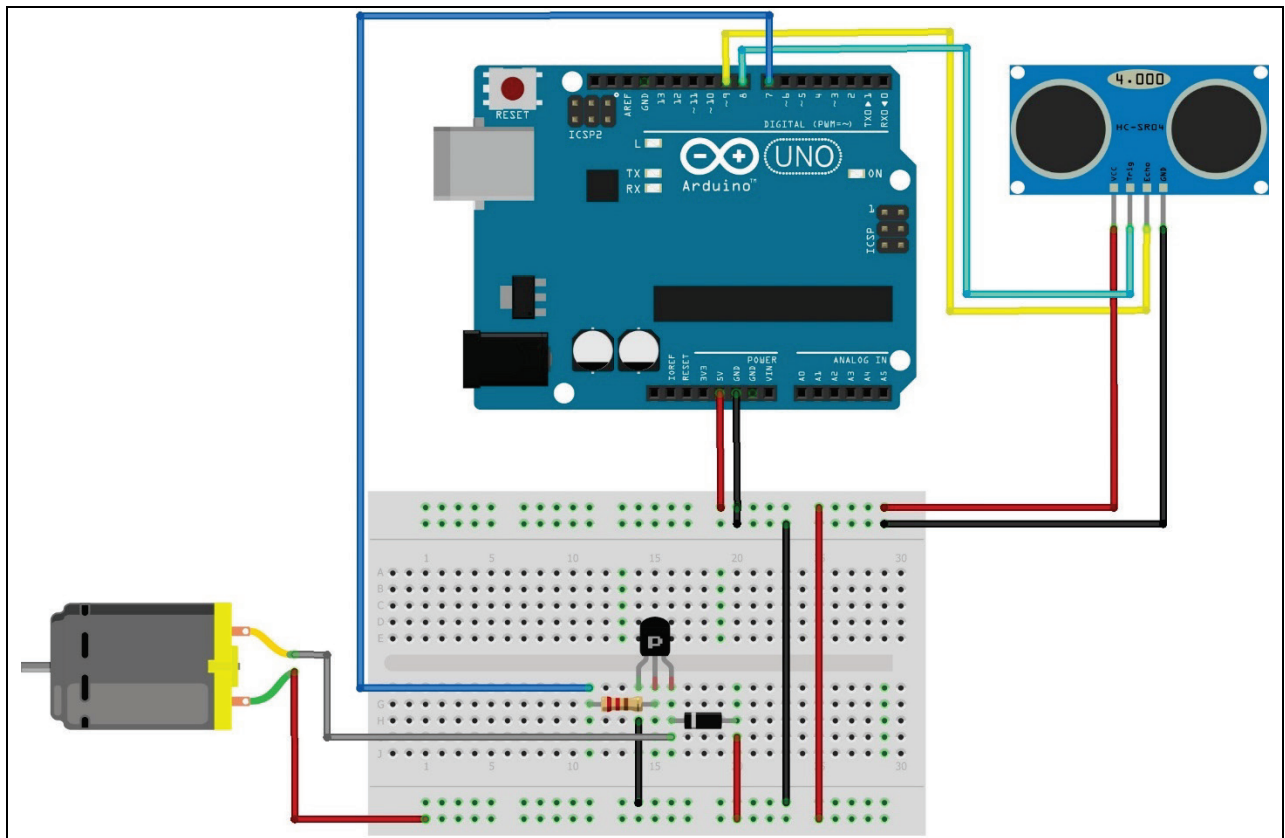
Ο αισθητήρας υπερήχων μετρά την απόσταση από το σημείο εκπομπής του μέχρι το σημείο που βρίσκεται το αντικείμενο (π.χ. η στάθμη του νερού στο φρεάτιο). Όταν ο αισθητήρας υπερήχων εντοπίσει ότι η στάθμη του νερού είναι μικρότερη ή ίση με 8 cm (πάνω από το μέγιστο όριο), τότε θα ενεργοποιηθεί το μοτέρ DC (αντλία) και θα συνεχίσει να λειτουργεί μέχρι η στάθμη του νερού να γίνει μεγαλύτερη ή ίση με 15 cm (κάτω από το ελάχιστο όριο). Το συγκεκριμένο σενάριο δεν χρειάζεται κάποια βιβλιοθήκη για την λειτουργία του.

Τα υλικά που χρησιμοποιήθηκαν είναι:

- Arduino uno
- HC-SR04 Ultrasonic Module Distance Measuring Transducer
- DC Mini Motor
- 270 Ω Resistor

- N4001 diode
- PN2222 Transistor

Το κύκλωμα για το σενάριο αυτόματης άντλησης νερών είναι το εξής:



Εικόνα 5-3 Σενάριο αυτόματης άντλησης νερών

Ο κώδικας για το σενάριο αυτόματης άντλησης νερών (τρέχει σε Arduino του έξυπνου σπιτιού):

```
int pumperPin=7; // Ορισμός αντλίας pin 7
#define ultrasonictriggerPin 8 //Ορισμός αισθητήρα υπερήχων ultrasonic trigger pin 8
#define ultrasonicechoPin 9 //Ορισμός αισθητήρα υπερήχων ultrasonic echo pin 9
int flag=0;
void setup()
{
  pinMode(pumperPin,OUTPUT); //ορισμός της αντλίας ως έξοδο
  pinMode(ultrasonictriggerPin, OUTPUT); //ορισμός του αισθητήρα υπερήχων ultrasonic trigger ως έξοδο
  pinMode(ultrasonicechoPin, INPUT); //ορισμός του αισθητήρα υπερήχων ultrasonic echo ως είσοδο
}
void loop()
{
  int pumperPin =digitalRead(pumperPin);
  long duration, distance;
  digitalWrite(ultrasonictriggerPin, LOW);
  delayMicroseconds(2);
```

```
digitalWrite(ultrasonictriggerPin, HIGH);
delayMicroseconds(10);
digitalWrite(ultrasonictriggerPin, LOW);
duration = pulseIn(ultrasonicechoPin, HIGH);
distance = (duration/2) / 29.1;
if (distance <=8) flag=1;
if (distance >=15) flag=0;
if (flag==1)
{
  do{
    digitalWrite(pumperPin,HIGH);
    Serial.println("Pump-Αντλία is ON"); }
  while(distance>=15);
}
else
{
  Serial.println("Pump-Αντλία is OFF");
  digitalWrite(pumperPin,LOW);
}
}
```

Κώδικας 5-3 Κώδικας σεναρίου αυτόματης άντλησης νερών

5.5 ΣΕΝΑΡΙΟ ΑΝΙΧΝΕΥΣΗΣ ΦΩΤΙΑΣ Ή ΠΛΗΜΜΥΡΑΣ

Σε αυτό το σενάριο, ο χρήστης έχει τη δυνατότητα να ενημερώνεται άμεσα για την ανίχνευση φωτιάς και πλημμύρας στο σπίτι του, εξασφαλίζοντας έτσι την έγκαιρη επέμβαση όταν αυτή είναι απαραίτητη.

Τα υλικά που χρησιμοποιήθηκαν είναι:

- Arduino uno
- Flame Sensor Module IR Sensor Detector
- Water Level Sensor
- Buzzer
- 2x Leds (fire, flood)

Η λειτουργία του σεναρίου ανίχνευσης της φωτιάς και της πλημμύρας είναι η εξής:

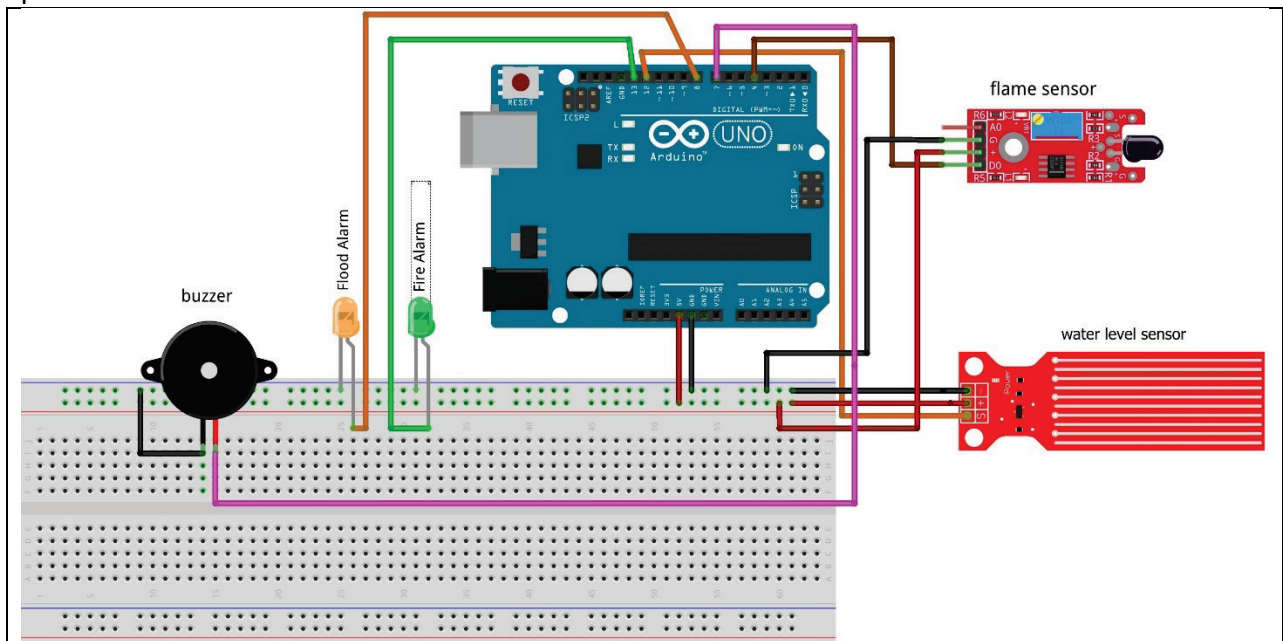
Όταν ο αισθητήρας νερού ανιχνεύσει πλημμύρα σε κάποιο χώρο του σπιτιού, θα καταγράψει την τιμή λογικό 1, θα ανάψει το πορτοκαλί LED (Flood Alarm) και θα ενεργοποιηθεί η ηχητική ένδειξη (Buzzer) με προκαθορισμένο ηχητικό σήμα.

Όταν ο αισθητήρας φωτιάς ανιχνεύσει ακτινοβολία σε κάποιο χώρο του σπιτιού, θα καταγράψει την τιμή λογικό 1, θα ανάψει το πράσινο LED (Fire Alarm) και θα ενεργοποιηθεί η ηχητική ένδειξη (Buzzer) με διαφορετικό προκαθορισμένο ηχητικό σήμα. Ο αισθητήρας φωτιάς μπορεί

να ρυθμιστεί με τη βοήθεια του ποτενσιόμετρου που διαθέτει, ανάλογα με τον χώρο στον οποίο λειτουργεί.

Σε περίπτωση που και οι δύο αισθητήρες ανιχνεύσουν ταυτόχρονα πλημμύρα και ακτινοβολία σε κάποιο χώρο του σπιτιού, θα καταγράψουν την τιμή λογικό 1, θα ανάψουν τα LEDs Flood & Fire Alarm (πορτοκαλί και πράσινο) ταυτόχρονα και θα ενεργοποιηθεί η ηχητική ένδειξη (Buzzer) με προκαθορισμένο ηχητικό σήμα.

Το κύκλωμα που υλοποιήθηκε για το σενάριο ανίχνευσης φωτιάς και πλημμύρας είναι το παρακάτω:



Εικόνα 5-4 Σενάριο ανίχνευσης φωτιάς και πλημμύρας

Ο κώδικας του σεναρίου για την ανίχνευση φωτιάς και πλημμύρας(τρέχει στο Arduino του έξυπνου σπιτιού):

```
int Flood=13; // pin led Flood Alarm
int Flame=4; // pin Fire Sensor
int Alarm=7; // pin led buzzer
int Fire=8; // pin led Fire Alarm
int Grove_Water_Sensor=12; //pin Flood Sensor
void setup()
{
  Serial.println("FIRE & FLOOD TEST");
  pinMode(Flame, INPUT); //define FLAME input pin
  pinMode(Alarm, OUTPUT); //define ALARM output pin
  pinMode (Flood,OUTPUT);
  pinMode(Grove_Water_Sensor, INPUT); // Water Sensor Input
  pinMode(FIRE,OUTPUT ); //LED Output
  void loop()
  {
    int flood=digitalRead(Grove_Water_Sensor);
    int fire = digitalRead(Flame); //read FLAME sensor
    if (flood==HIGH && fire == LOW ) {
```

```

digitalWrite(Alarm,HIGH);
digitalWrite(Fire,HIGH);
digitalWrite(Flood,HIGH);
Serial.println("Flood & Fire alert"); }
else if (digitalRead(Grove_Water_Sensor)==HIGH){
digitalWrite(Alarm,HIGH);
delay(200);
digitalWrite(Alarm,LOW);
delay(200);
Serial.println("Flood Alert");
digitalWrite(Flood,HIGH);
}
else if( fire == LOW)
{
Serial.println("Fire alert");
digitalWrite(Alarm,HIGH); //buzzer ON
delay(50);
digitalWrite(Alarm,LOW);
delay(50);
digitalWrite(Alarm,HIGH);
}
else {
digitalWrite(Alarm,LOW); //buzzer OFF
digitalWrite(Flood,LOW);
Serial.println("NO Flood");
digitalWrite(Fire,LOW);
Serial.println("NO Fire");}
}

```

Κώδικας 5-4 Κώδικας σεναρίου ανίχνευσης φωτιάς και πλημμύρας

5.6 ΣΕΝΑΡΙΟ ΕΛΕΓΧΟΥ ΠΟΡΤΑΣ ΓΚΑΡΑΖ (Η ΑΛΛΗΣ ΠΟΡΤΑΣ) – ΗΧΗΤΙΚΗ Ή ΟΠΤΙΚΗ ΕΝΔΕΙΞΗ ΚΑΤΑΣΤΑΣΗΣ ΠΟΡΤΑΣ

Η κατάσταση της πόρτας του γκαράζ ελέγχεται με τη χρήση ενός αισθητήρα κλίσης υδραργύρου. Αυτό το σύστημα έχει σχεδιαστεί για να προειδοποιεί έγκαιρα τον χρήστη σχετικά με το αν η πόρτα του γκαράζ είναι ανοιχτή ή κλειστή. Η προειδοποίηση προς τον χρήστη γίνεται είτε με ηχητική ένδειξη μέσω ενός Buzzer είτε με οπτική ένδειξη κατάστασης μέσω LED.

Το ίδιο σενάριο μπορεί να λειτουργήσει και με μαγνητικό αισθητήρα (on/off) για έλεγχο οποιαδήποτε πόρτας/κεντρικής εισόδου.

Τα υλικά που χρησιμοποιήθηκαν είναι:

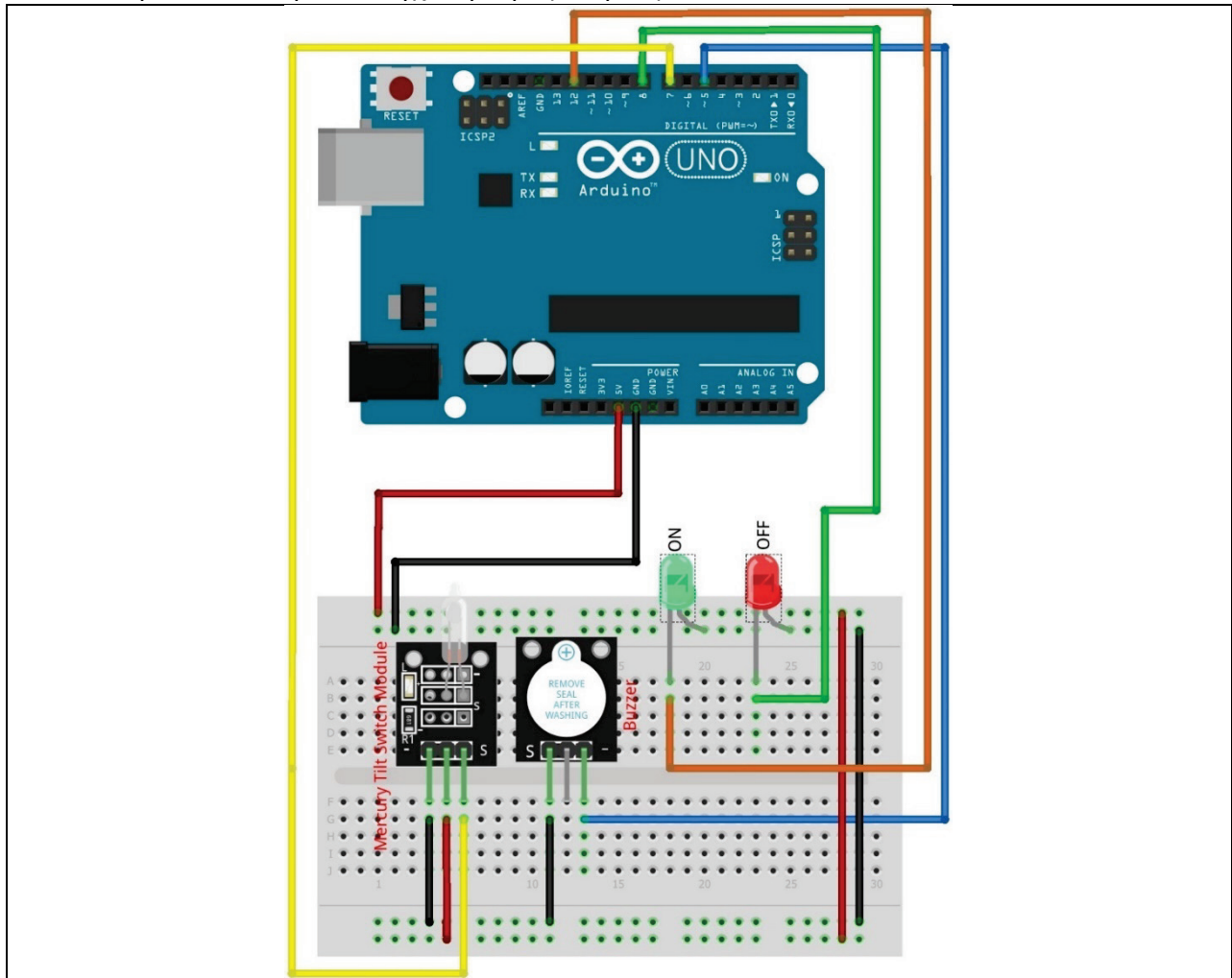
- Arduino uno
- KY-017 Mercury Tilt Switch Module

- 1x Buzzer
- 2x Leds (garage port on, garage port off)

Το συγκεκριμένο σενάριο δεν χρειάζεται κάποια βιβλιοθήκη για την λειτουργία του.

Όταν η πόρτα του γκαράζ είναι κλειστή, ο αισθητήρας θα καταγράψει την τιμή λογικό 0 και θα ανάψει η κόκκινη λυχνία LED. Εάν η πόρτα του γκαράζ μείνει ανοιχτή λόγω λάθους του χρήστη ή αν κάποιος προσπαθήσει να την παραβιάσει, ο αισθητήρας θα καταγράψει την τιμή λογικό 1, θα ανάψει η μπλε λυχνία LED και θα ενεργοποιηθεί η ηχητική ειδοποίηση (Buzzer).

Το κύκλωμα του σεναρίου ελέγχου γκαραζόπορτας:



Εικόνα 5-5 Σχεδίαση ελέγχου γκαραζόπορτας

Ο κώδικας του σεναρίου για τον έλεγχο της γκαραζόπορτας (τρέχει στο Arduino του έξυπνου σπιτιού):

```
int mercurytiltSwitch = 7; //Ορισμός pin mercury tilt Switch 7
int garageport; //Ορισμός variable για το tilt input
int buzzer=5; //Ορισμός pin buzzer 5
int greenled=12; //Ορισμός pin green led garage open 12
int redled=8; //Ορισμός pin red led garage closed 8
void setup ()
{
```



```
pinMode(buzzer,OUTPUT); //ορισμός του Buzzer ως έξοδο
pinMode(greenled,OUTPUT); //ορισμός του green led ως έξοδο
pinMode(redled,OUTPUT); //ορισμός του red led ως έξοδο
Serial.begin(9600); //εκκίνηση σειριακής
pinMode (mercurytiltSwitch, INPUT) ; //ορισμός του mercury tilt Switch ως είσοδο
}
void loop ()
{
garageport = digitalRead (mercurytiltSwitch) ; //διάβασμα της τιμής mercury tilt Switch
if (garageport == LOW) //tilted
{
digitalWrite(buzzer,HIGH);
delay(100);
digitalWrite(buzzer,LOW);
delay(100);
digitalWrite(redled,HIGH);
digitalWrite(greenled,LOW);
Serial.println("GARAZE PORT IS OPENED");
}
else
{
digitalWrite(greenled,HIGH);
digitalWrite(redled,LOW);
digitalWrite(buzzer,LOW);
Serial.println("GARAZE PORT IS CLOSED");
}
delay(1000); //καθυστέρηση 1 sec
}
```

Κώδικας 5-5 Κώδικας σεναρίου ελέγχου γκαραζόπορτας

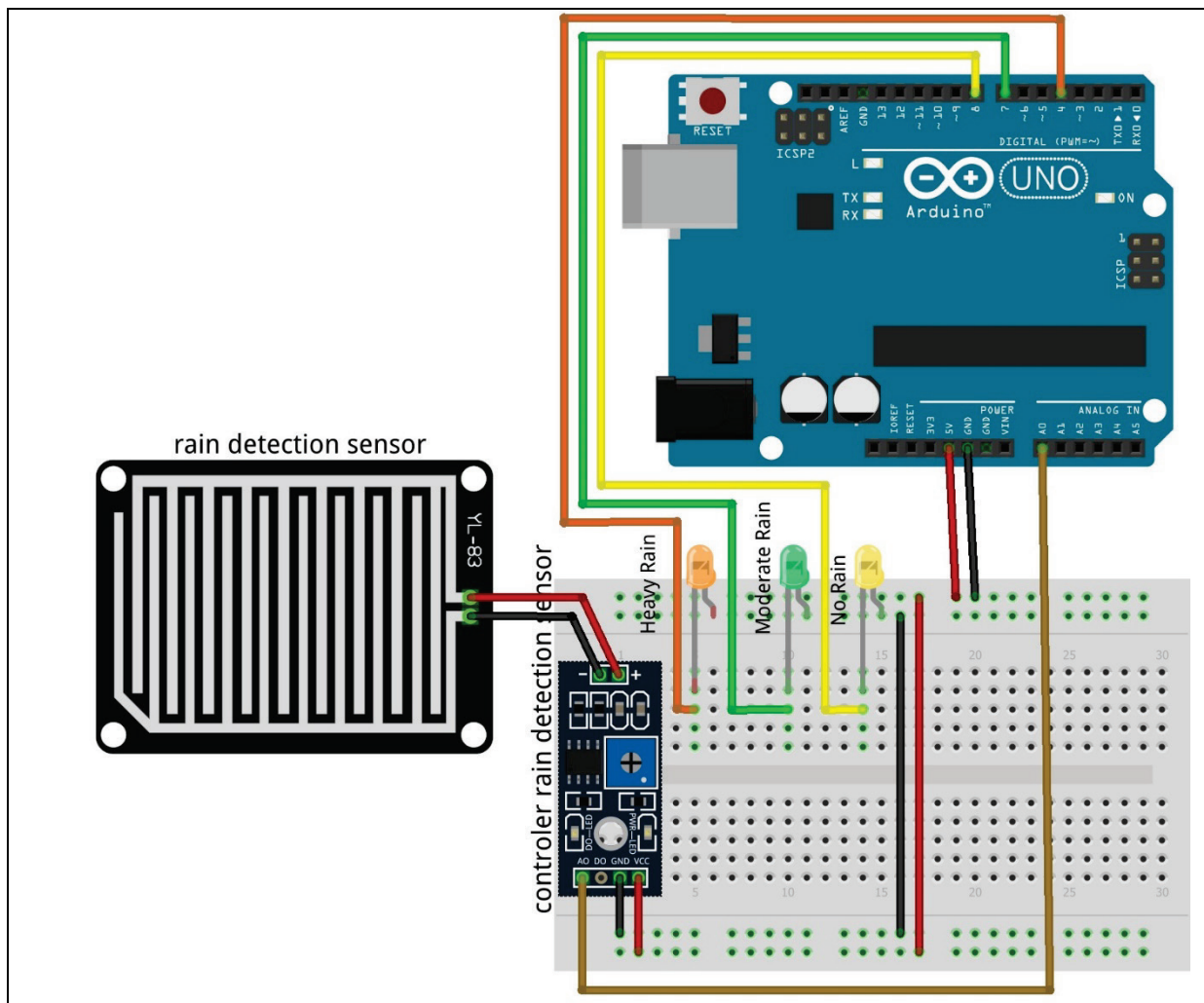
5.7 ΣΕΝΑΡΙΟ ΑΝΙΧΝΕΥΣΗΣ ΠΟΣΟΣΤΟΥ ΒΡΟΧΗΣ

Ο αισθητήρας ανίχνευσης βροχής είναι ένα εύχρηστο εργαλείο για την ανίχνευση βροχής.

Μπορεί να λειτουργεί ως διακόπτης όταν πέφτουν σταγόνες βροχής πάνω του ή να μετρά την ένταση της βροχής. Αποτελείται από μια πλακέτα βροχής και μια πλακέτα ελέγχου που είναι ξεχωριστές για ευκολία, με ενδεικτικό LED λειτουργίας και ποτενσιόμετρο για τη ρύθμιση της ευαισθησίας.

Η αναλογική έξοδος χρησιμοποιείται για την ανίχνευση της έντασης της βροχής, ενώ η ψηφιακή έξοδος δείχνει αν βρέχει ή όχι. Όταν συνδεθεί στα 5V, το LED ανάβει και αν δεν υπάρχει βροχή, η ψηφιακή έξοδος είναι HIGH. Αν πέσουν λίγες σταγόνες νερού, η ψηφιακή έξοδος γίνεται LOW.

Το κύκλωμα σύνδεσης του σεναρίου ανίχνευσης ποσοστού βροχής είναι:



Εικόνα 5-6 Κύκλωμα σύνδεσης του σεναρίου ανίχνευσης βροχής

Ο κώδικας για το σενάριο ανίχνευσης βροχής:

```
int led=4; // pin led Heavy Rain
int led1=7; // pin led Moderate Rain
int led2=8; // pin led No Rain

void setup(){
  Serial.begin(9600);
  pinMode(2, OUTPUT);
}

void loop(){
  //analog output
  if(analogRead(0)<300) {
    Serial.println("Heavy Rain");
    digitalWrite(led,HIGH);
  }
  else if(analogRead(0)<500) {
    Serial.println("Moderate Rain");
    digitalWrite(led1,HIGH);}
  else {
    Serial.println("No Rain");
    digitalWrite(led2,HIGH);}
```

```
delay(250);  
}
```

Κώδικας 5-6 Κώδικας σεναρίου ανίχνευσης βροχής

5.8 ΣΕΝΑΡΙΟ ΕΛΕΓΧΟΥ ΘΕΡΜΟΚΡΑΣΙΑΣ ΜΕΣΩ ΚΛΙΜΑΤΙΣΜΟΥ ΚΑΙ ΕΛΕΓΧΟΥ ΠΟΙΟΤΗΤΑΣ ΑΕΡΑ

Το συγκεκριμένο σενάριο υλοποιήθηκε για τον έλεγχο της θερμοκρασίας μέσα στο σπίτι και την παρακολούθηση της ποιότητας του αέρα. Πιο συγκεκριμένα:

- Έλεγχος της θερμοκρασίας του χώρου μέσω συστήματος κλιματισμού και αισθητήρα μέτρησης θερμοκρασίας.
- Βελτίωση της ποιότητας του αέρα μέσω συστήματος εξαερισμού, με τη βοήθεια αισθητήρα που ανιχνεύει επιβλαβή αέρια και επικίνδυνες ουσίες.

Λειτουργία αυτόματου σεναρίου: Για το σύστημα ποιότητας αέρα χρησιμοποιήθηκε ένας αισθητήρας ανίχνευσης επιβλαβών αερίων και ένας ανεμιστήρας που εξυπηρετεί στον εξαερισμό του σπιτιού. Όταν ο αισθητήρας ανιχνεύσει συγκέντρωση αερίων μεγαλύτερη από την καθορισμένη τιμή (αναλογική τιμή 300), θα ενεργοποιηθεί ο ανεμιστήρας για τον εξαερισμό του χώρου. Ο ανεμιστήρας θα συνεχίσει να λειτουργεί μέχρι η ατμόσφαιρα να καθαρίσει και η τιμή να πέσει κάτω από 300.

Για το σύστημα κλιματισμού υπάρχουν οι εξής επιλογές λειτουργίας:

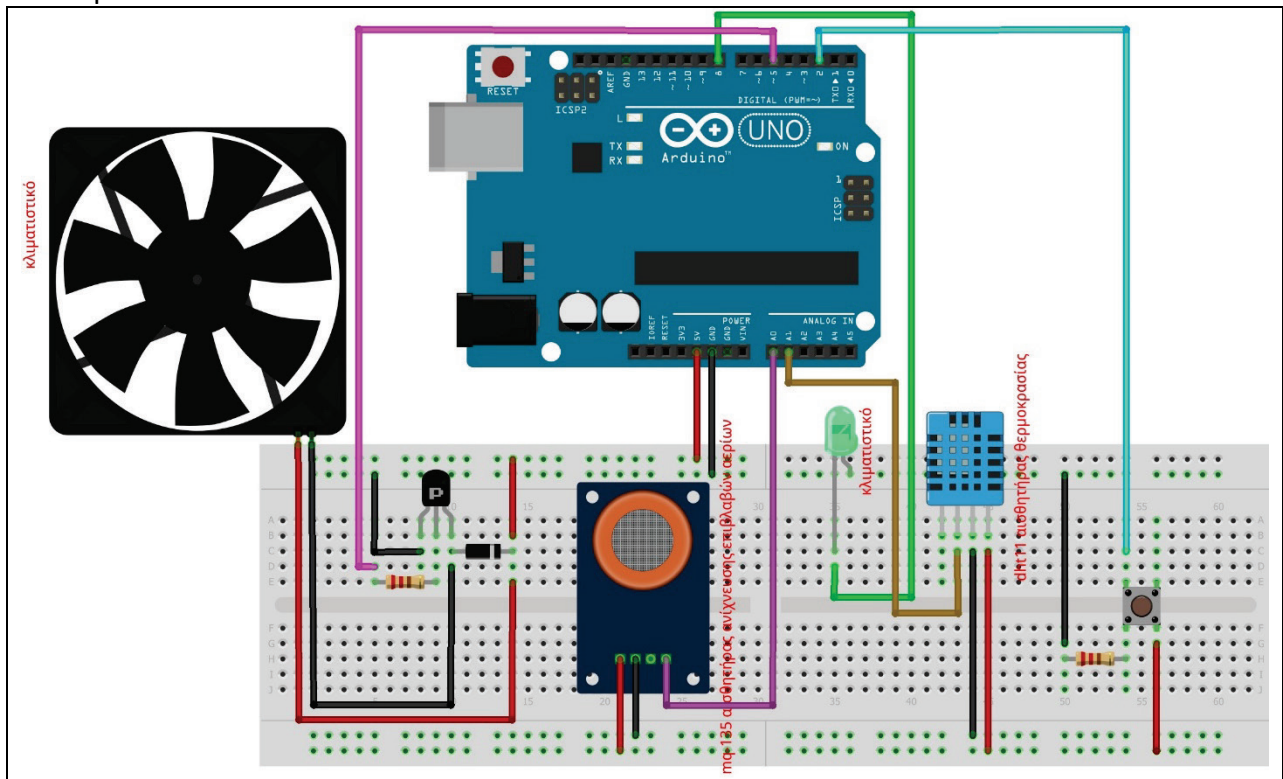
- Αν η θερμοκρασία κυμαίνεται μεταξύ 20 και 24 βαθμών Κελσίου, θεωρείται φυσιολογική και ο χρήστης μπορεί να ενεργοποιήσει το κλιματιστικό χειροκίνητα μέσω ενός κουμπιού, εάν το επιθυμεί.
- Αν η θερμοκρασία υπερβαίνει τους 24 βαθμούς Κελσίου (κατά τη διάρκεια του καλοκαιριού), το κλιματιστικό θα ρυθμιστεί στη λειτουργία ψύξης και θα ενεργοποιηθεί αυτόματα μέχρι η θερμοκρασία να πέσει κάτω από τους 24 βαθμούς.
- Αν η θερμοκρασία πέσει κάτω από τους 20 βαθμούς Κελσίου (κατά τη διάρκεια του χειμώνα), το κλιματιστικό θα ρυθμιστεί στη λειτουργία θέρμανσης και θα ενεργοποιηθεί αυτόματα μέχρι η θερμοκρασία να ανέβει πάνω από τους 20 βαθμούς.

Τα υλικά που χρησιμοποιήθηκαν είναι:

- Arduino uno
- Temperature and humidity sensor module
- MQ-135 gas sensor module
- Fan (ανεμιστήρας)
- 1 x Button
- 1 x Green Led (εξομοίωση λειτουργίας κλιματιστικού)
- 1x PN2222 Transistor

- 1x N4001 diode
- 2x 270 Ω resistor

Το κύκλωμα που υλοποιήθηκε για το σενάριο ελέγχου κλιματισμού και ποιότητας αέρα είναι το παρακάτω:



Εικόνα 5-7 Συνδεσμολογία σεναρίου ελέγχου θερμοκρασίας μέσω κλιματισμού και ελέγχου ποιότητας αέρα με Arduino

Ο κώδικας του σεναρίου ελέγχου κλιματισμού και ποιότητας αέρα (τρέχει στο Arduino του έ-
ξυπνου σπιτιού):

```
#include <dht.h> // βιβλιοθήκη αισθητήρα θερμοκρασίας
#define dht_dpin A1 // pin αισθητήρα θερμοκρασίας
int sensorValue; //αισθητήρας ανίχνευσης επιβλαβών αερίων //Air
int pinButton = 2; // pin button
int stateLED = HIGH;
int stateButton;
int previous = LOW;
long time = 0;
long debounce = 200;
int FAN=5; // pin ανεμιστήρα
int AC=8; // pin led κλιματισμού
dht dht;
void setup()
{
  pinMode(pinButton, INPUT);
  pinMode(FAN, OUTPUT);
  Serial.println("Air Pollution Level – Επίπεδο μόλυνσης αέρα \n\n");
  pinMode(AC, OUTPUT);
}
```

```

Serial.println("Humidity & Temperature – Υγρασία & Θερμοκρασία\n\n");
void loop()
{
stateButton = digitalRead(pinButton);
sensorValue = analogRead(0);
int fan =digitalRead(FAN);
if(sensorValue>300)
{
digitalWrite(FAN,HIGH );
}
else
digitalWrite(FAN,LOW );
Serial.print("Air Pollution Level – Επίπεδο μόλυνσης αέρα= ");
Serial.println(sensorValue, DEC); // prints the value read
if ((dht.temperature>28 || dht.temperature>24) || (dht.temperature<15 ||
dht.temperature<20))
{digitalWrite(AC,HIGH );
}
else digitalWrite(AC,LOW );
if(dht.temperature>=20 && dht.temperature<=24){
if(stateButton == HIGH && previous == LOW && millis() - time > debounce) {
if(stateLED == HIGH){
stateLED = LOW;
} else {
stateLED = HIGH;
}
time = millis();
}
digitalWrite(AC, stateLED);
previous = stateButton;}
dht.read11(dht_dp);
Serial.print("Current Humidity – Τρέχουσα Υγρασία = ");
Serial.print(dht.humidity);
Serial.print("% ");
Serial.print("Current Temperature – Τρέχουσα Θερμοκρασία = ");
Serial.print(dht.temperature);
Serial.println("C ");
}

```

Κώδικας 5-7 Κώδικας σεναρίου ελέγχου θερμοκρασίας μέσω κλιματισμού και ελέγχου ποιότητας αέρα με Arduino

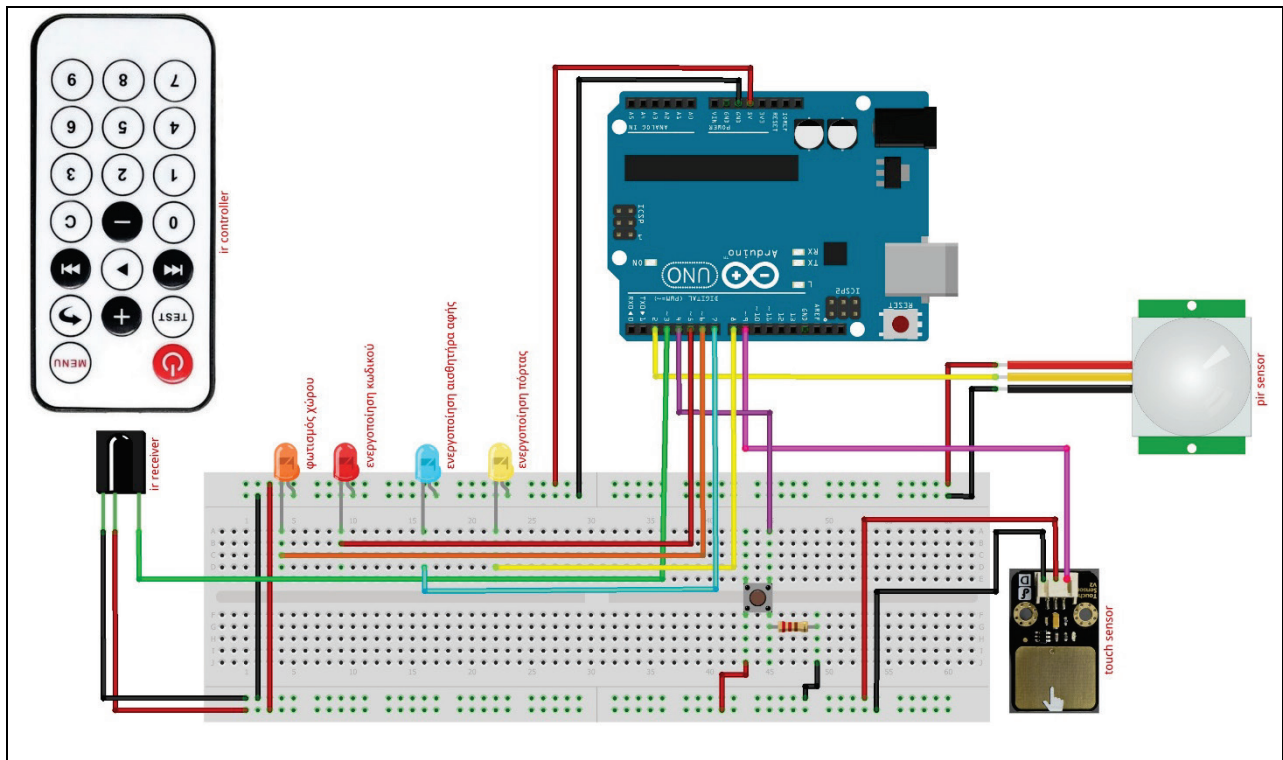
5.9 ΣΕΝΑΡΙΟ ΑΝΟΙΓΜΑΤΟΣ ΕΞΩΤΕΡΙΚΗΣ ΠΟΡΤΑΣ ΜΕ ΔΥΟ ΒΑΘΜΙΔΕΣ ΑΣΦΑΛΕΙΑΣ (ΑΙΣΘ. ΑΦΗΣ ΚΑΙ ΚΩΔΙΚΟΣ)

Για το σενάριο ανοίγματος εξωτερικής πόρτας, χρησιμοποιήθηκαν δύο επίπεδα ασφάλειας. Το πρώτο επίπεδο περιλαμβάνει το πάτημα ενός προκαθορισμένου χρονικού διαστήματος στον αισθητήρα αφής, ενώ το δεύτερο επίπεδο ασφάλειας απαιτεί τον συνδυασμό ενός κωδικού. Χρησιμοποιήθηκε ο αισθητήρας PIR (Passive Infrared) για να διευκολύνει το χρήστη σε περιπτώσεις έλλειψης φωτισμού στον χώρο της εξωτερικής πόρτας. Όταν ο αισθητήρας εντοπίζει κίνηση, ενεργοποιείται ένας λαμπτήρας φωτισμού, ο οποίος αντιστοιχεί στο πορτοκαλί LED (φωτισμός χώρου) και παραμένει αναμμένος για τη διάρκεια της ανίχνευσης κίνησης. Αυτός ο φωτισμός θα σβήσει δύο δευτερόλεπτα μετά τη διακοπή της κίνησης στον χώρο. Στη συνέχεια, τοποθετήθηκε ένα κουμπί (Button) για να υποδεικνύει εάν η πόρτα είναι ανοικτή ή κλειστή. Όταν το κουμπί είναι πατημένο, υποδηλώνει ότι η πόρτα είναι κλειστή.

Αυτόματο σενάριο: Για να λειτουργήσει το αυτόματο σενάριο, η εξωτερική πόρτα πρέπει να είναι κλειστή. Για να ξεκλειδώσει η πόρτα, πρέπει να ισχύουν ταυτόχρονα δύο συνθήκες. Αρχικά, ο χρήστης πρέπει να πατήσει τον αισθητήρα αφής για ένα συγκεκριμένο χρονικό διάστημα, έτσι ώστε να ενεργοποιηθεί η πρώτη συνθήκη. Εάν ο αισθητήρας πατηθεί για λιγότερο ή περισσότερο χρόνο, η συνθήκη απενεργοποιείται. Για την επιτυχή ολοκλήρωση της διαδικασίας, ο χρήστης πρέπει να εισάγει έναν τριψήφιο κωδικό μέσω του τηλεχειριστηρίου, ο οποίος ενεργοποιεί τη δεύτερη συνθήκη. Για λόγους επεξήγησης, η επιτυχής ενεργοποίηση του κωδικού αντιστοιχεί στην ενεργοποίηση του κόκκινου LED (ενεργοποίηση κωδικού). Υπάρχει ένα χρονικό όριο είκοσι δευτερολέπτων, μέσα στο οποίο ο χρήστης πρέπει να ενεργοποιήσει και τις δύο συνθήκες για να ανοίξει η πόρτα. Εάν μία από τις δύο συνθήκες δεν είναι ορθή, η διαδικασία επαναλαμβάνεται από την αρχή. Αν και οι δύο συνθήκες είναι ορθές, τότε η πόρτα ξεκλειδώνει, και για λόγους επεξήγησης, ενεργοποιείται το κίτρινο LED (ενεργοποίηση πόρτας). Τα υλικά που χρησιμοποιήθηκαν είναι:

- Arduino uno
- Mini IR Infrared Pyroelectric PIR Body Motion Human Sensor Detector Module
- Infrared IR Receiver Module Wireless Remote Control Kit
- DFRobot Capacitive Touch Sensor
- Button
- 4 x Leds (φωτισμού, αισθητήρα αφής, κωδικού, πόρτας)

Το κύκλωμα για το σενάριο ασφάλειας εξωτερικής πόρτας είναι το παρακάτω:



Εικόνα 5-8 Σενάριο ανοίγματος εξωτερικής πόρτας με δύο βαθμίδες ασφάλειας (αισθ. αφής και κωδικός)

Ο κώδικας για το σενάριο ανοίγματος εξωτερικής πόρτας με δύο βαθμίδες ασφάλειας (αισθ. αφής και κωδικός) (τρέχει στο Arduino Uno του έξυπνου σπιτιού):

```
int led_fotistikou = 6; // pin led φωτιστικού σώματος
int pir_sensor = 2; //pir sensor pin
int katastasi = LOW; //by default, no motion detected
int val = 0; // sensor status (value)
#define ctsPin 9 //touch sensor pin
int simaia01=0;
int simaia02=0;
int metritis=0;
int led_touch_sensor=7; //pin led touch sensor
int led_portas=8; //pin led πόρτας
int period = 10000;
unsigned long time_now = 0;
int led=5; //pin led κωδικού
int a=0;
int b=0;
int c=0;
const int buttonPin = 4; // pin button
int buttonkatastasi = 0;
#include <IRremote.h> // library IR
int RECV_PIN = 3; // pin ir receiver
IRrecv irrecv(RECV_PIN);
decode_results results;
```



```

void setup()
{
  pinMode(led_fotistikou, OUTPUT); // initialize LED as an output //asfaleia
  pinMode(pir_sensor, INPUT); // initialize sensor as an input
  pinMode(led_touch_sensor, OUTPUT);
  pinMode(led_portas, OUTPUT);
  pinMode(ctsPin, INPUT);
  pinMode(led, OUTPUT);
  Serial.begin(9600);
  irrecv.enableIRIn(); // Start the receiver
}
void loop()
{
  if (irrecv.decode(&results))
  {
    Serial.println(results.value, HEX);
    translateIR();
    for (int z=0; z<2; z++)
    {
      irrecv.resume();
    }
  }
  buttonkatastasi = digitalRead(buttonPin);
  if (buttonkatastasi == HIGH) {
    if(millis() > (time_now + period)){
      if (simaia02+simaia01==2) digitalWrite(led_portas,HIGH);
      else digitalWrite(led_portas,LOW);
      metritis=0;
      Serial.println("-----");
      time_now = millis();
      a=0; b=0; c=0;
      simaia01=0;
      digitalWrite(led,LOW);
    }
    else{
      if (digitalRead(ctsPin)==HIGH) metritis=metritis+1;
      if (metritis<5 &&metritis>1){
        digitalWrite(led_touch_sensor,HIGH);
        simaia02=1;
      }
      else {
        simaia02=0;
        digitalWrite(led_touch_sensor,LOW);}
      if (a+b+c==3){
        digitalWrite(led,HIGH);
        simaia01=1;
      }
    }
  }
}

```

```
Serial.println(metritis);
val = digitalRead(pir_sensor);
if (val == HIGH) {
  digitalWrite(led_fotistikou, HIGH);
  delay(100);
  if (katastasi == LOW) {
    Serial.println("Motion detected!");
    katastasi = HIGH;
  }
}
else {
  digitalWrite(led_fotistikou, LOW);
  delay(100);
  if (katastasi == HIGH){
    Serial.println("Motion stopped!");
    katastasi = LOW;    // update variable katastasi to LOW
  }
}
void translateIR() // takes action based on IR code received
// describing Sony IR codes on LCD module
{
  switch(results.value)
  {
    case 0xFFA25D:
    a=1;
    break;
    case 0xFF629D:
    b=1;
    break;
    case 0xFFE21D:
    c=1;
    break;
    default:
    Serial.println("OTHER");
  }
  delay(200);
}
```

Κώδικας 5-8 Κώδικας σεναρίου ανοίγματος εξωτερικής πόρτας με δύο βαθμίδες ασφάλειας (αισθ. αφής και κωδικός)

5.10 ΣΕΝΑΡΙΟ ΑΥΤΟΜΑΤΟΥ ΠΟΤΙΣΜΑΤΟΣ ΓΛΑΣΤΡΩΝ ΣΕ ΜΠΑΛΚΟΝΙ

Σε αυτό το σενάριο θα ενεργοποιείται το πότισμα σε μία γλάστρα στο μπαλκόνι ανάλογα με την υγρασία του χώματος.

Λειτουργία αυτόματου σεναρίου:

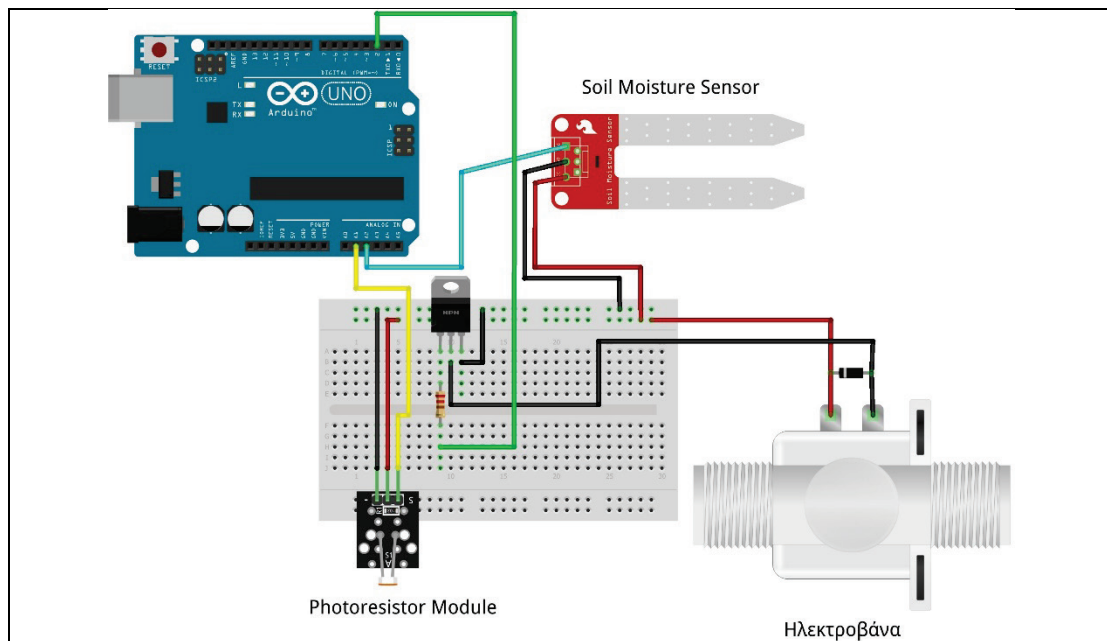
Για την ενεργοποίηση του αυτόματου ποτίσματος, υπάρχουν δύο συνθήκες που πρέπει να ισχύουν. Αρχικά, διαβάζεται η τιμή της υγρασίας του χώματος από τον αισθητήρα soil moisture sensor. Αν το ποσοστό υγρασίας είναι μικρότερο του 30% ή μικρότερο του 70%, τότε ενεργοποιείται η πρώτη συνθήκη.

Συνεπώς, το αυτόματο πότισμα θα ενεργοποιηθεί μόλις ανιχνευθεί υγρασία λιγότερη από 30% και θα προχωρήσει στο πότισμα μέχρι να φτάσει στο 70%. Με αυτόν τον τρόπο, διατηρείται η υγρασία σε επιθυμητά επίπεδα. Έπειτα, διαβάζεται η τιμή της ηλιοφάνειας με τον αισθητήρα photoresistor. Για να ισχύει η δεύτερη συνθήκη, η τιμή πρέπει να είναι μικρότερη από 60%. Αυτό γίνεται για να αποφευχθεί το πότισμα κατά τη διάρκεια έντονης ηλιοφάνειας.

Τα υλικά που χρησιμοποιήθηκαν είναι:

- Arduino uno
- Soil moisture detector module
- KY-018 PHOTORESISTOR MODULE
- Ηλεκτροβάννα

Το κύκλωμα που υλοποιήθηκε για το σενάριο αυτόματου ποτίσματος γλαστρών σε μπαλκόνι είναι:



Εικόνα 5-9 Σενάριο αυτόματου ποτίσματος γλαστρών σε μπαλκόνι

Ο κώδικας για το σενάριο αυτόματου ποτίσματος γλαστρών σε μπαλκόνι:

```
int sensorPin02 = A4; // soil moisture sensor pin
int sensorPin01 = A3; // photoresistor module pin
int watering=2;      // pin ηλεκτροβάννας
int water_value ;
int sun_value = 0; // variable to store the value coming from the sensor
void setup()
{
  pinMode(watering,OUTPUT);
```

```

}
void loop()
{
  water_value= analogRead(sensorPin01);           // watering
  sun_Value1 = analogRead(sensorPin02);
  water_value = map(water_value,350,1023,120,0);
  sun_Value1 = map(sun_value,15,800,100,0);
  if(sun_Value1<60 && (water_value<30 || water_value<70 ))
    {digitalWrite(watering,HIGH); }
    else
    {digitalWrite(watering,LOW); }
    Serial.println(water_value);
    Serial.println(sun_value);
}

```

Κώδικας 5-9 Κώδικας σεναρίου αυτόματου ποτίσματος γλαστρών σε μπαλκόνι

5.11 ΣΕΝΑΡΙΟ ΑΥΤΟΜΑΤΙΣΜΟΥ ΚΗΠΟΥ

Στο σενάριο αυτό υλοποιήθηκε ένα σύστημα εποπτείας και ελέγχου του κήπου, το οποίο έχει τη δυνατότητα να επιβλέπει περιβαλλοντικές συνθήκες και να ελέγχει συσκευές μέσω του διαδικτύου.

Μέσω της δυνατότητας επίβλεψης ο χρήστης έχει τη δυνατότητα να σχηματίσει μία εικόνα για την κατάσταση του κήπου σε πραγματικό χρόνο, ενώ η καταγραφή των περιβαλλοντικών συνθηκών βοηθάει στην εξαγωγή χρήσιμων συμπερασμάτων ώστε να επιτευχθεί το βέλτιστο αποτέλεσμα.

Δημιουργήθηκε ένα σενάριο που αντικαθιστά πλήρως την ανθρώπινη παρέμβαση μέσω της χρήσης αισθητήρων και ενεργοποιητών. Με αυτόν τον τρόπο, ο χρήστης μπορεί να προσδιορίζει τη συμπεριφορά των διάφορων συσκευών βάσει των περιβαλλοντικών συνθηκών του κήπου, λαμβάνοντας υπόψη τις πληροφορίες που παρέχουν οι αισθητήρες.

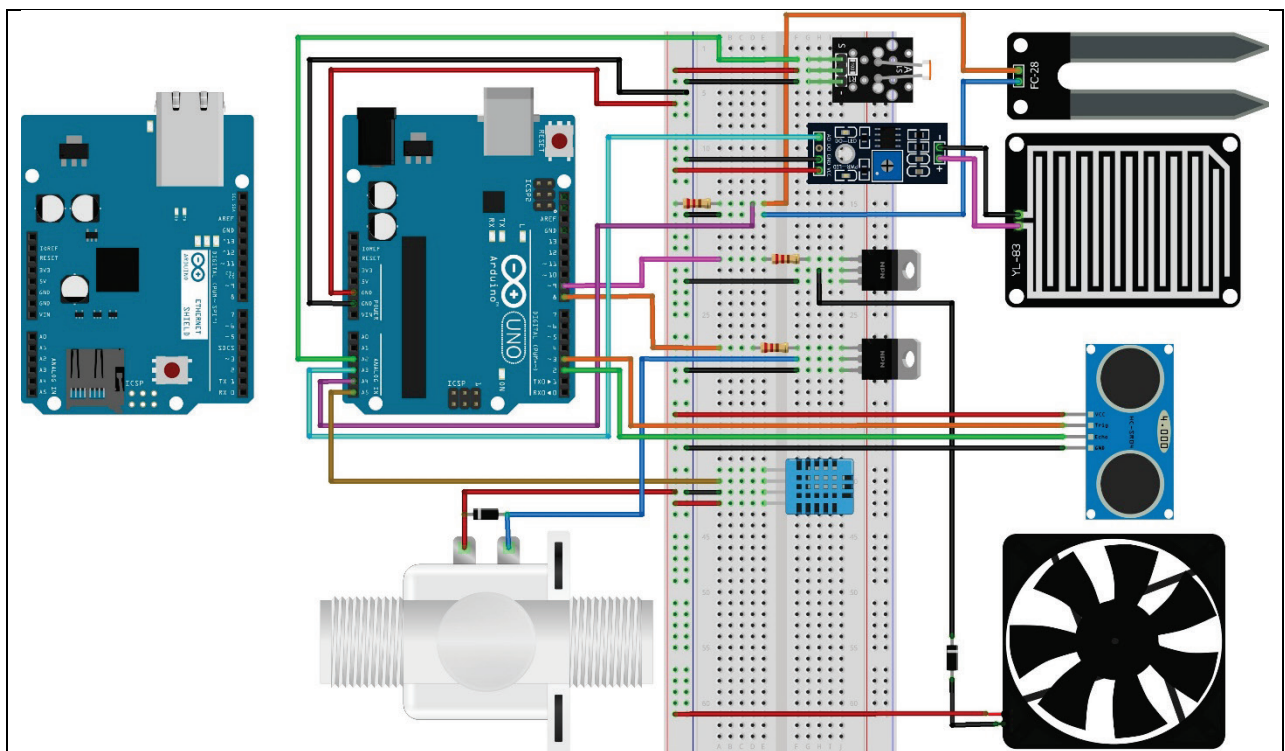
Πιο συγκεκριμένα, το πρόγραμμα αφορά την αυτόματη ποτίστρα του κήπου ή του θερμοκηπίου. Αυτό σημαίνει ότι όταν ο αισθητήρας υγρασίας του εδάφους διαπιστώσει ότι το έδαφος είναι αρκετά ξηρό, ο ελεγκτής θα ενεργοποιήσει τη βαλβίδα πότισης μέχρι το έδαφος να αποκτήσει την απαιτούμενη υγρασία.

Παράλληλα, υλοποιήθηκε και πρόγραμμα αφορά τον αυτόματο αερισμό του κήπου όταν αυτός βρίσκεται σε κλειστό χώρο, όπως ένα θερμοκήπιο. Αναλυτικότερα, όταν ενεργοποιείται αυτό το σενάριο, κάθε φορά που η θερμοκρασία υπερβαίνει ένα συγκεκριμένο όριο, ο ανεμιστήρας ενεργοποιείται μέχρι η θερμοκρασία να επιστρέψει σε φυσιολογικά επίπεδα.

Στο συγκεκριμένο σενάριο μπήκε και η αποστολή των δεδομένων σε βάση μέσω ενός ethernet shield.

Το κύκλωμα το οποίο κατασκευάστηκε για τη δημιουργία του συστήματος εποπτείας και ελέγχου του κήπου αποτελείται από τα εξής:

- Arduino uno
- Ethernet shield
- Ένταση φωτός Photoresistor module
- Αισθητήρας βροχής FC-37
- Αισθητήρας υγρασίας χώματος FC-28
- Αισθητήρας θερμοκρασίας -υγρασίας DHT11
- Αισθητήρας υπέρηχων για το ύψος της δεξαμενής με HC-SR04
- Ηλεκτροβάννα
- Ανεμιστήρας



Εικόνα 5-10 Σενάριο αυτοματισμού κήπου

Ο κώδικας για το σενάριο αυτοματισμού κήπου/θερμοκηπίου (τρέχει σε Arduino του έξυπνου σπιτιού):

```
#include <SPI.h>
#include <Ethernet.h>
#include <SD.h>
#define REQ_BUF_SZ 60
#include <dht.h>
#define dht_dpin A5
dht DHT;
const int trig_pin = 2;
const int echo_pin = 3;
long duration, cm;
```

```

int metritis_db = 0;
byte mac[] = { 0x11, 0x22, 0x33, 0x44, 0x55, 0x66 };
EthernetClient client;
char server[] = "192.168.1.100"; //IP Address (or name) of server to dump data to
IPAddress ip(192,168,1,101);
EthernetServer server1(80); //cserver at port 80
File webFile; //web page file on the SD card
char HTTP_req[REQ_BUF_SZ] = {0}; //buffered HTTP request stored as null terminated string
char req_index = 0; //index into HTTP_req buffer
boolean state[4] = {0}; //stores the states of the LEDs
int interval = 500; //Wait between dumps
void setup() {
    pinMode(10, OUTPUT);
    digitalWrite(10, HIGH);
    SD.begin(4);
    pinMode(8, OUTPUT);
    pinMode(9, OUTPUT);
    Ethernet.begin(mac);
    Ethernet.begin(mac, ip); //initialize Ethernet device
    server1.begin(); //start to listen for clients
}
void loop() {
    //if you get a connection, report back via serial:
    db();
    if (state[0]) {
        if(analogRead(2)>800){
            digitalWrite(8, HIGH);}
        else{digitalWrite(8,LOW);}
    }
    if (state[1]) {
        if(analogRead(4)>650){
            digitalWrite(9, HIGH);}
        else{digitalWrite(9,LOW);}
    }
    pinMode(trig_pin, OUTPUT);
    digitalWrite(trig_pin, LOW);
    delayMicroseconds(2);
    digitalWrite(trig_pin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trig_pin, LOW);
    pinMode(echo_pin, INPUT);
    duration = pulseIn(echo_pin, HIGH);
    cm = microsecondsToCentimeters(duration);
    DHT.read11(dht_dpin);
    EthernetClient client = server1.available(); //try to get client
    if (client) { //got client?
        boolean currentLineIsBlank = true;
        while (client.connected()) {
            if (client.available()) { //client data available to read

```

```

char c = client.read(); //read 1 byte (character) from client
//limit the size of the stored received HTTP request
//buffer first part of HTTP request in HTTP_req array (string)
//leave last element in array as 0 to null terminate string (REQ_BUF_SZ - 1)
if (req_index < (REQ_BUF_SZ - 1)) {
    HTTP_req[req_index] = c; //save HTTP request character
    req_index++;
}
//last line of client request is blank and ends with \n
//respond to client only after last line received
if (c == '\n' && currentLineIsBlank) {
    //send a standard http response header
    client.println("HTTP/1.1 200 OK");
    //remainder of header follows below, depending on if
    //web page or XML page is requested
    //ajax request - send XML file
    if (StrContains(HTTP_req, "GET /logo.jpg")) {
        webFile = SD.open("logo.jpg");
        if (webFile) {
            client.println("HTTP/1.1 200 OK");
            client.println();
        }
        if (webFile) {
            while(webFile.available()) {
                client.write(webFile.read()); //send web page to client
            }
            webFile.close();
        }
    }
    if (StrContains(HTTP_req, "ajax_inputs")) {
        //send rest of HTTP header
        client.println("Content-Type: text/xml");
        client.println("Connection: keep-alive");
        client.println();
        Set();
        //send XML file containing input states
        XML_response(client);
    }
    else{
        //web page request
        //send rest of HTTP header
        client.println("Content-Type: text/html");
        client.println("Connection: keep-alive");
        client.println();
        //send web page
        webFile = SD.open("index.htm"); //open web page file
        if (webFile) {
            while(webFile.available()) {
                client.write(webFile.read()); //send web page to client
            }
        }
    }
}

```



```

        }
        webFile.close();
    }
}
//display received HTTP request on serial port
Serial.print(HTTP_req);
//reset buffer index and all buffer elements to 0
req_index = 0;

break;
}
//every line of text received from the client ends with \r\n
if (c == '\n') {
    //last character on line of received text
    //starting new line with next character read
    currentLineIsBlank = true;
}
else if (c != '\r') {
    //a text character was received from client
    currentLineIsBlank = false;
}
} //end if (client.available())
} //end while (client.connected())
delay(1); //give the web browser time to receive the data
client.stop(); //close the connection
} //end if (client)
}

void db () {
    metritis_db = (metritis_db + 1);
    if (metritis_db >= 1000){
    if (client.connect(server, 80)) {
        Serial.println("-> Connected");
        //make a HTTP request:
        client.print( "GET /arduino/add_data.php?");
        client.print("&");
        client.print("light=");
        client.print( (analogRead(2))/10.23 );
        client.print("&");
        client.print("raindrop=");
        client.print( (analogRead(3))/10.23 );
        client.print("&");
        client.print("moisture=");
        client.print( (analogRead(4))/10.23 );
        client.print("&");
        client.print("temperature=");
        client.print( DHT.temperature );
        client.print("&");
        client.print("humidity=");
        client.print( DHT.humidity );
    }
    }
}

```

```

client.print("&&");
client.print("tank=");
client.print( cm );
metritis_db = 0 ;
client.println( " HTTP/1.1");
client.print( "Host: " );
client.println(server);
client.println( "Connection: close" );
client.println();
client.println();
client.stop();

}
else {
    //you didn't get a connection to the server:
    Serial.println("--> connection failed/n");
}
}
}
void Set(void)
{
    //auto Fan
    if (StrContains(HTTP_req, "LED1=1")) {
        state[0] = 1; //save LED state
    }
    else if (StrContains(HTTP_req, "LED1=0")) {
        state[0] = 0; // save LED state
        digitalWrite(8, LOW);
    }
    //auto Solenoid
    if (StrContains(HTTP_req, "LED2=1")) {
        state[1] = 1; //save LED state
    }
    else if (StrContains(HTTP_req, "LED2=0")) {
        state[1] = 0; //save LED state
    }
    //fan
    if (StrContains(HTTP_req, "LED3=1")) {
        state[2] = 1; //save LED state
        digitalWrite(8, HIGH);
    }
    else if (StrContains(HTTP_req, "LED3=0")) {
        state[2] = 0; //save LED state
        digitalWrite(8, LOW);
    }
    //solenoid
    if (StrContains(HTTP_req, "LED4=1")) {
        state[3] = 1; //save LED state
        digitalWrite(9, HIGH);
    }
}

```

```

}
else if (StrContains(HTTP_req, "LED4=0")) {
    state[3] = 0; //save LED state
    digitalWrite(9, LOW);
}
}
}
void XML_response(EthernetClient cl)
{
    int analog_val; //stores value read from analog inputs
    int count; //used by 'for' loops
    cl.print("<?xml version = \"1.0\" ?>");
    cl.print("<inputs>");
    //read analog inputs
    for (count = 2; count <= 4; count++) { //A2 to A5
        analog_val = (analogRead(count));
        cl.print("<analog>");
        cl.print((analog_val)/10.23);
        cl.println("</analog>");
    }
    cl.print("<analog>");
    cl.print(DHT.temperature);
    cl.print("</analog>");
    cl.print("<analog>");
    cl.print(DHT.humidity);
    cl.println("</analog>");
    cl.print("<analog>");
    cl.print(cm);
    cl.println("</analog>");
    //checkbox auto states
    //LED1
    cl.print("<DEVICE>");
    if (state[0]) {
        cl.print("checked");
    }
    else {
        cl.print("unchecked");
    }
    cl.println("</DEVICE>");
    //LED2
    cl.print("<DEVICE>");
    if (state[1]) {
        cl.print("checked");
    }
    else {
        cl.print("unchecked");
    }
    cl.println("</DEVICE>");
    //button LED states
    //LED3

```

```

cl.print("<DEVICE>");
if (state[2]) {
    cl.print("on");
}
else {
    cl.print("off");
}
cl.println("</DEVICE>");
//LED4
cl.print("<DEVICE>");
if (state[3]) {
    cl.print("on");
}
else {
    cl.print("off");
}
cl.println("</DEVICE>");

cl.print("</inputs>");
}
char StrContains(char *str, char *sfind)
{
    char found = 0;
    char index = 0;
    char len;
    len = strlen(str);

    if (strlen(sfind) > len) {
        return 0;
    }
    while (index < len) {
        if (str[index] == sfind[found]) {
            found++;
            if (strlen(sfind) == found) {
                return 1;
            }
        }
        else {
            found = 0;
        }
        index++;
    }
    return 0;
}
long microsecondsToCentimeters(long microseconds)
{
    return microseconds / 29 / 2;
}

```

5.12 ΣΕΝΑΡΙΟ ΕΛΕΓΧΟΥ ΕΞΩΤΕΡΙΚΩΝ ΡΟΛΩΝ ΚΑΙ ΕΛΕΓΧΟΥ ΦΩΤΙΣΜΟΥ ΕΞΩΤΕΡΙΚΟΥ ΧΩΡΟΥ

Στο σενάριο αυτό, η λειτουργία του φωτισμού στον εξωτερικό χώρο και ο έλεγχος των εξωτερικών ρολών έχουν αυτοματοποιηθεί με βάση τη φωτεινότητα και την ώρα της ημέρας. Αυτό έχει σαν αποτέλεσμα την εξοικονόμηση ενέργειας καθώς ο φωτισμός ενεργοποιείται μόνο όταν απαιτείται, ενώ ο έλεγχος των ρολών προσφέρει βοήθεια στον χρήστη και εξοικονομεί θερμική ενέργεια, ειδικά σε κατοικίες με πολλαπλά ανοίγματα.

Η λειτουργία του σεναρίου είναι η εξής:

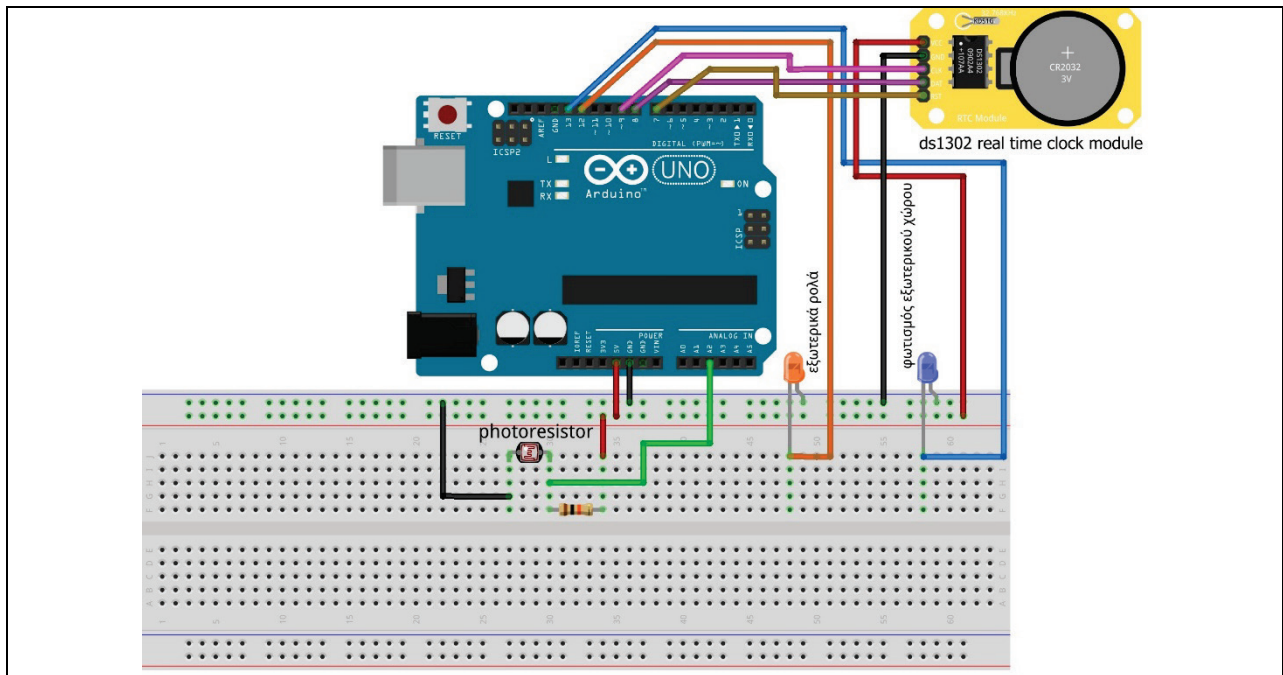
Φωτισμός εξωτερικού χώρου (μπλέ led): Στο σενάριο αυτό, το μπλε LED (φωτισμός εξωτερικού χώρου) υποδηλώνει εάν ο εξωτερικός φωτισμός είναι ενεργός ή ανενεργός. Όταν ο αισθητήρας φωτοαντίστασης ανιχνεύσει ότι η φωτεινότητα είναι κάτω από το 50% (όταν ο ήλιος δύει), τότε ενεργοποιείται ο φωτισμός στον εξωτερικό χώρο. Αντίστοιχα, ο φωτισμός θα απενεργοποιηθεί όταν η φωτεινότητα ξεπεράσει το 40% κατά τη διάρκεια του ξημερώματος.

Εξωτερικά ρολά: Στο σενάριο αυτό, το πορτοκαλί LED (εξωτερικά ρολά) υποδεικνύει εάν τα εξωτερικά ρολά είναι ανοικτά ή κλειστά. Εκτός από την παράμετρο του φωτισμού του εξωτερικού χώρου, έχει καθοριστεί και η παράμετρος της ώρας της ημέρας. Αυτό έχει γίνει ώστε να είναι δυνατό να οριστεί η ώρα κλεισίματος των εξωτερικών ρολών κατά τους μήνες που η μέρα είναι μικρότερη, ανάλογα με τις προτιμήσεις του χρήστη. Η ώρα διαβάζεται με τη βοήθεια του αισθητήρα DS1302 Real Time Clock Module. Έτσι, εάν η φωτεινότητα πέσει κάτω από το 50% και η ώρα είναι αυτή που έχει οριστεί από τον χρήστη, τότε τα εξωτερικά ρολά κλείνουν. Αντίστοιχα, εάν η φωτεινότητα ξεπεράσει το 40% και η ώρα είναι η καθορισμένη από τον χρήστη, τότε τα εξωτερικά ρολά ανοίγουν.

Τα υλικά που χρησιμοποιήθηκαν είναι:

- Arduino uno
- DS1302 Real Time Clock Module
- Photoresistor
- 2x Leds (παντζούρια, εξωτερικός φωτισμός)
- Resistor 10000Ω

Το κύκλωμα του σεναρίου ελέγχου εξωτερικών ρολών και φωτισμού εξωτερικού χώρου είναι:



Εικόνα 5-11 Σενάριο ελέγχου εξωτερικών ρολών και φωτισμού εξωτερικού χώρου

Ο κώδικας για το σενάριο ελέγχου εξωτερικών ρολών και φωτισμού εξωτερικού χώρου:

```
#include <DS1302.h> //βιβλιοθήκη ds1302 clock module
int led_fotismos=13; //pin led εξωτερικού φωτισμού
int photoresisto_pin = A2; //pin photoresistor
int sun_value = 0;
int led_rola=12; // pin led εξωτερικών ρολών
DS1302 rtc(7,8,9); // pin clock module
Time t;
const int on_hour = 00; //ώρα ανοίγματος εξωτερικών ρολών
const int On_min = 04;
const int off_Hour = 00; //ώρα κλεισίματος εξωτερικών ρολών
const int off_min = 06;
void setup()
{
  rtc.halt(false); //εξωτερικός φωτισμός - εξωτερικά ρολά
  rtc.writeProtect(false);
  pinMode (led_fotismos,OUTPUT);
  pinMode (led_rola,OUTPUT);
}
void loop()
{
  t = rtc.getTime(); //εξωτερικός φωτισμός - εξωτερικά ρολά
  Serial.print(t.hour);
  Serial.print(" hour(s), ");
  Serial.print(t.min);
  Serial.print(" minute(s)");
  Serial.println(" ");

  sun_value = analogRead(photoresisto_pin);
```

```

sun_value = map(sun_value,100,800,100,0);
if(sun_value<50){
  digitalWrite(led_fotismos,HIGH);
  Serial.println("e3wterikos fwtismos ON");
  if (t.hour == on_hour && t.min == On_min)
  {digitalWrite(led_rolla,HIGH);
  Serial.println("3ewterika rola OFF");}
}

else if(sun_value>40){
  digitalWrite(led_fotismos,LOW);
  Serial.println("e3wterikos fwtismos OFF");
  if (t.hour == off_Hour && t.min == off_min)
  {digitalWrite(led_rolla,LOW);
  Serial.println("3ewterika rola OPEN");}
}
}

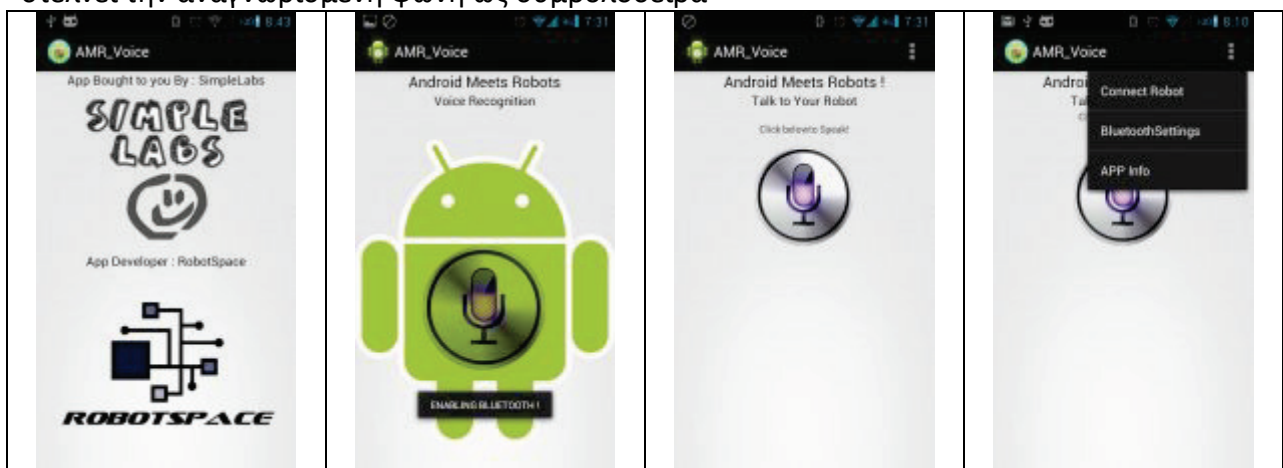
```

Κώδικας 5-11 Κώδικας σεναρίου ελέγχου εξωτερικών ρολών και φωτισμού εξωτερικού χώρου

5.13 ΣΕΝΑΡΙΟ ΕΛΕΓΧΟΥ ΜΕ ΦΩΝΗΤΙΚΕΣ ΕΝΤΟΛΕΣ(ΜΕΣΩ BLUETOOTH)

Σε αυτό το σενάριο, χρησιμοποιούμε τη φωνητική εντολή στο κινητό μας τηλέφωνο για να επικοινωνήσουμε με το Arduino. Αυτό συμβαίνει μέσω Bluetooth, επιτρέποντάς μας να δώσουμε εντολές για τον έλεγχο συσκευών στο σπίτι. Ουσιαστικά, εκμεταλλευόμαστε την τεχνολογία αναγνώρισης φωνής που διαθέτει το κινητό μας για να επικοινωνήσουμε με το Arduino και να εκτελέσουμε διάφορες λειτουργίες στο σπίτι μας από απόσταση.

Η εφαρμογή AMR Voice που χρησιμοποιούμε είναι αρκετά απλή και τη βρίσκουμε στο google play store. Η συγκεκριμένη εφαρμογή χρησιμοποιεί την εσωτερική αναγνώριση φωνής του Android για να μεταβιβάσει φωνητικές εντολές. Συνδέεται με σειριακές μονάδες Bluetooth και στέλνει την αναγνωρισμένη φωνή ως συμβολοσειρά



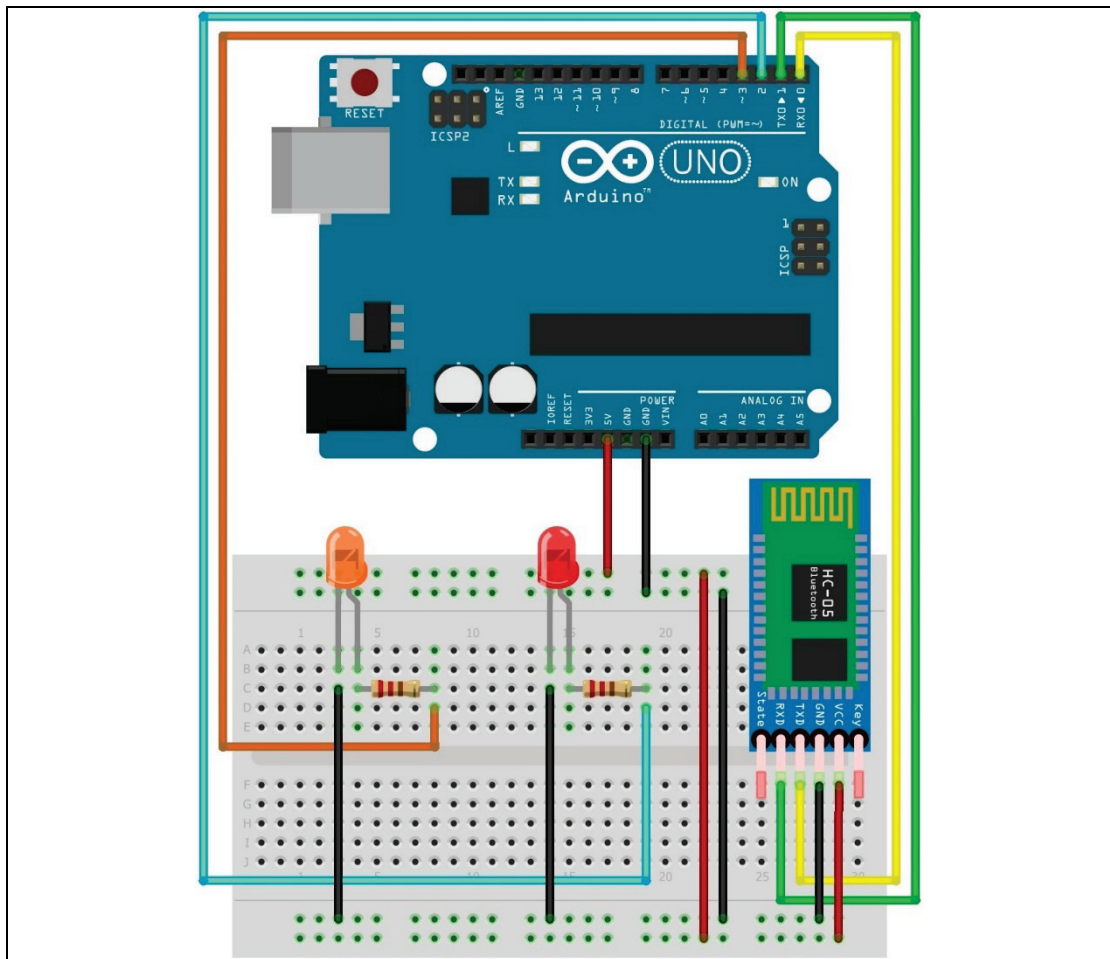
Εικόνα 5-12 Εφαρμογή AMR Voice
(amr-voice, n.d.)

Στην εφαρμογή όταν πατάμε το κουμπί του μικροφώνου, αναμένει μέχρι να του δώσουμε την εντολή μας. Έπειτα, η εφαρμογή μας καταγράφει τις λέξεις που εκφωνήσαμε και τις μετατρέπει σε ένα string, ώστε το Arduino να μπορεί να τις επεξεργαστεί. Στη συνέχεια, ελέγχουμε τη σειριακή θύρα του Arduino και περιμένουμε ένα string που να προέρχεται από τη φωνητική εντολή. Οι φωνητικές εντολές, όπως είδαμε, στέλνονται μέσω Bluetooth από το κινητό μας τηλέφωνο με τη χρήση μιας εφαρμογής που ορίσαμε. Στον παρακάτω κώδικα, ορίζουμε τις εντολές που μπορούμε να χρησιμοποιήσουμε για να ενεργοποιήσουμε ή να απενεργοποιήσουμε ορισμένα LED. Στη συνέχεια, δίνουμε μια εντολή από το κινητό μας τηλέφωνο, το οποίο μετατρέπει τη φωνή μας σε χαρακτήρες string και τους στέλνει μέσω Bluetooth στο Arduino. Το Arduino, όπως φαίνεται στον παρακάτω κώδικα, περιμένει να λάβει κάποιες εντολές που έχουμε ορίσει. Αν αναγνωρίσει τις εντολές που του έχουμε στείλει, τότε θα εκτελέσει τον κώδικα που αντιστοιχεί σε κάθε μία από αυτές.

Τα υλικά που χρησιμοποιήθηκαν είναι:

- Arduino uno
- Bluetooth Module
- Arduino UNO
- Mobile Phone
- LED with resistor
- Εφαρμογή Android AMR Voice

Σύνδεση με το Arduino



Εικόνα 5-13 Συνδεσμολογία Bluetooth HC 06 με Arduino

Στη συνέχεια ακολουθεί και ο κώδικας ελέγχου των εντολών από το Arduino.

```
String voice;
int
led1 = 2, //Connect LED 1 To Pin #2
led2 = 3, //Connect LED 2 To Pin #3
void allon(){
    digitalWrite(led1, HIGH);
    digitalWrite(led2, HIGH);
}
void alloff(){
    digitalWrite(led1, LOW);
    digitalWrite(led2, LOW);
}
void setup() {
    Serial.begin(9600);
    pinMode(led1, OUTPUT);
    pinMode(led2, OUTPUT);
}
void loop() {
    while (Serial.available()){ //Check if there is an available byte to read
        delay(10); //Delay added to make thing stable
        char c = Serial.read(); //Conduct a serial read
```

```

if (c == '#') {break;} //Exit the loop when the # is detected after the word
voice += c; //Shorthand for voice = voice + c
}
if (voice.length() > 0) {
  Serial.println(voice);
  if(voice == "*all on") {allon();} //Turn Off All Pins (Call Function)
  else if(voice == "*all off"){alloff();} //Turn On All Pins (Call Function)

  //-----Turn On One-By-One-----//
  else if(voice == "*TV on") {digitalWrite(led1, HIGH);}
  else if(voice == "*fan on") {digitalWrite(led2, HIGH);}
  //-----Turn Off One-By-One-----//
  else if(voice == "*TV off") {digitalWrite(led1, LOW);}
  else if(voice == "*fan off") {digitalWrite(led2, LOW);}
  //-----//
  voice="";
}

```

Κώδικας 5-12 Κώδικας σεναρίου Bluetooth HC 06 με Arduino

5.14 ΣΕΝΑΡΙΟ ΜΕΤΡΗΣΗΣ ΚΑΤΑΝΑΛΩΣΗΣ (ΜΕ ΜΕΤΡΗΤΗ ΤΑΣΗΣ ΚΑΙ ΡΕΥΜΑΤΟΣ)

Ο μη-επεμβατικός αισθητήρας ρεύματος, επίσης γνωστός ως "split core current transformer", τοποθετείται γύρω από τη γραμμή παροχής ενός ηλεκτρικού φορτίου και μετράει το ρεύμα που διέρχεται μέσω αυτού. Αυτό επιτυγχάνεται ενεργώντας ως επαγωγέας και αντιδρώντας στο μαγνητικό πεδίο γύρω από τον αγωγό ρεύματος. Διαβάζοντας το ρεύμα που παράγεται από το πηνίο, μπορούμε να υπολογίσουμε το ρεύμα που διέρχεται από τον αγωγό. Αυτός ο αισθητήρας μπορεί να μετρήσει φορτίο έως 30 Amps, καθιστώντας τον ιδανικό για μετρητές κατανάλωσης ή για συσκευές προστασίας από υπερφόρτιση. Ωστόσο, δεν περιλαμβάνει ενσωματωμένη αντίσταση φορτίου, επομένως συνήθως απαιτείται η προσθήκη αντιστάτη σε ολόκληρη την έξοδο για τη μετατροπή του επαγόμενου ρεύματος σε μια μικρή μετρήσιμη τάση για το Arduino.

Πώς να κατασκευάσουμε έναν μετρητή τάσης και ρεύματος (πηγή: <https://learn.openenergymonitor.org>)

Με αυτόν τον οδηγό, μπορούμε να κατασκευάσουμε έναν απλό μετρητή που μετρά την τάση σε έναν μετασχηματιστή και το ρεύμα χρησιμοποιώντας έναν CT αισθητήρα. Έπειτα, με τη χρήση μιας κατάλληλης βιβλιοθήκης στο Arduino, μπορούμε να υπολογίσουμε την πραγματική ισχύ, τη φαινόμενη ισχύ, το συντελεστή ισχύος, την ενεργό τάση και το ρεύμα RMS.

Τι θα χρειαστούμε;

- Arduino

Ηλεκτρονικά για τη μέτρηση τάσης:

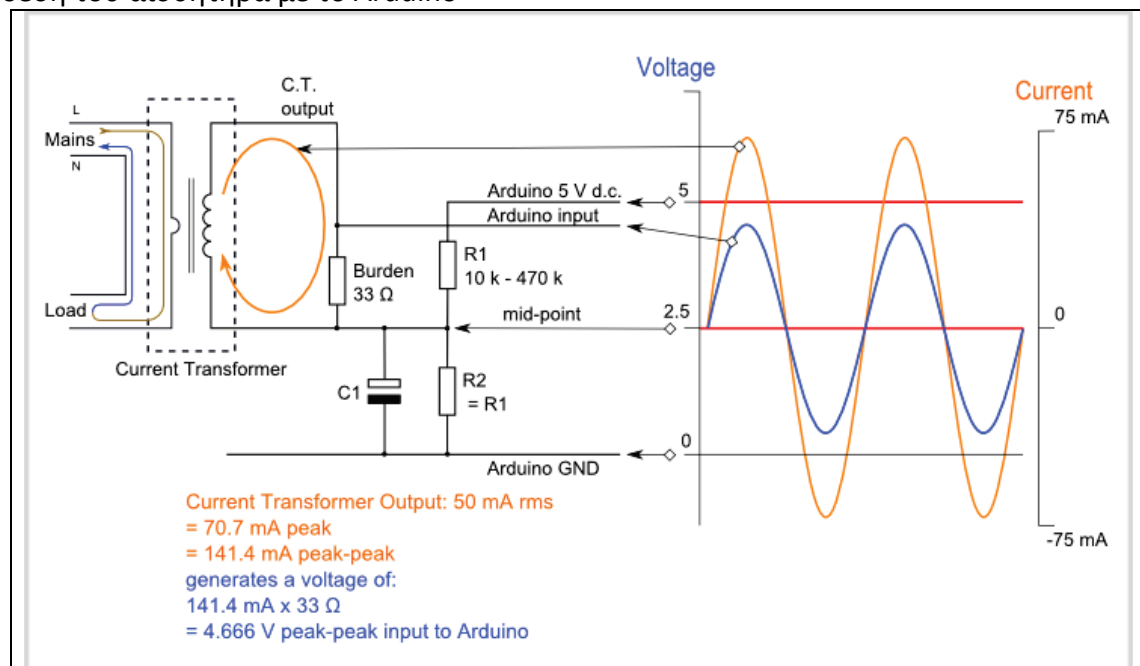
- 9V AC-AC Power Adapter.
- 100kOhm αντίσταση για υποβιβαστικό διαιρέτη τάσης.
- 10kOhm αντίσταση για υποβιβαστικό διαιρέτη τάσης.
- 2x 470kOhm (για διαιρέτη τάσης, οποιοδήποτε ζεύγος αντιστατών αντίστοιχης τιμής έως και 10K).
- 10uF capacitor.

Ηλεκτρονικά για μέτρηση ρεύματος:

- CT sensor SCT-013-000.
- Αντίσταση επιβάρυνσης 18 Ω εάν η τάση τροφοδοσίας είναι 3,3V ή 33 Ω εάν η τάση τροφοδοσίας είναι 5V.
- 470kOhm για διαιρέτη τάσης, οποιοδήποτε ζεύγος αντιστατών αντίστοιχης τιμής έως και 10K).
- 10uF capacitor.

Οι αισθητήρες ρεύματος CTS χρησιμοποιούνται για την ανίχνευση του εναλλασσόμενου ρεύματος και αποτελούν πολύτιμο εργαλείο στην παρακολούθηση της ηλεκτρικής κατανάλωσης (ή παραγωγής) σε ένα κτίριο. Όμοια με έναν μετασχηματιστή, αποτελούνται από ένα πρωτεύον τύλιγμα, ένα μαγνητικό πυρήνα και ένα δευτερεύον τύλιγμα. Στην παρακολούθηση κτιρίων, μετράμε το εναλλασσόμενο ρεύμα που διέρχεται από το πρωτεύον και παράγει ένα μαγνητικό πεδίο στον πυρήνα. Αυτό, στη συνέχεια, προκαλεί ένα ρεύμα στο δευτερεύον τύλιγμα, το οποίο είναι ανάλογο με το ρεύμα στο πρωτεύον. Για να μετρήσουμε το ρεύμα στο δευτερεύον, χρησιμοποιούμε μια αντίσταση φορτίου που κλείνει το κύκλωμα. Η τιμή της αντίστασης επιλέγεται ανάλογα με την τάση του δευτερεύοντος, με σκοπό να μην προκληθεί κορεσμός του πυρήνα. (πηγή: <https://learn.openenergymonitor.org>)

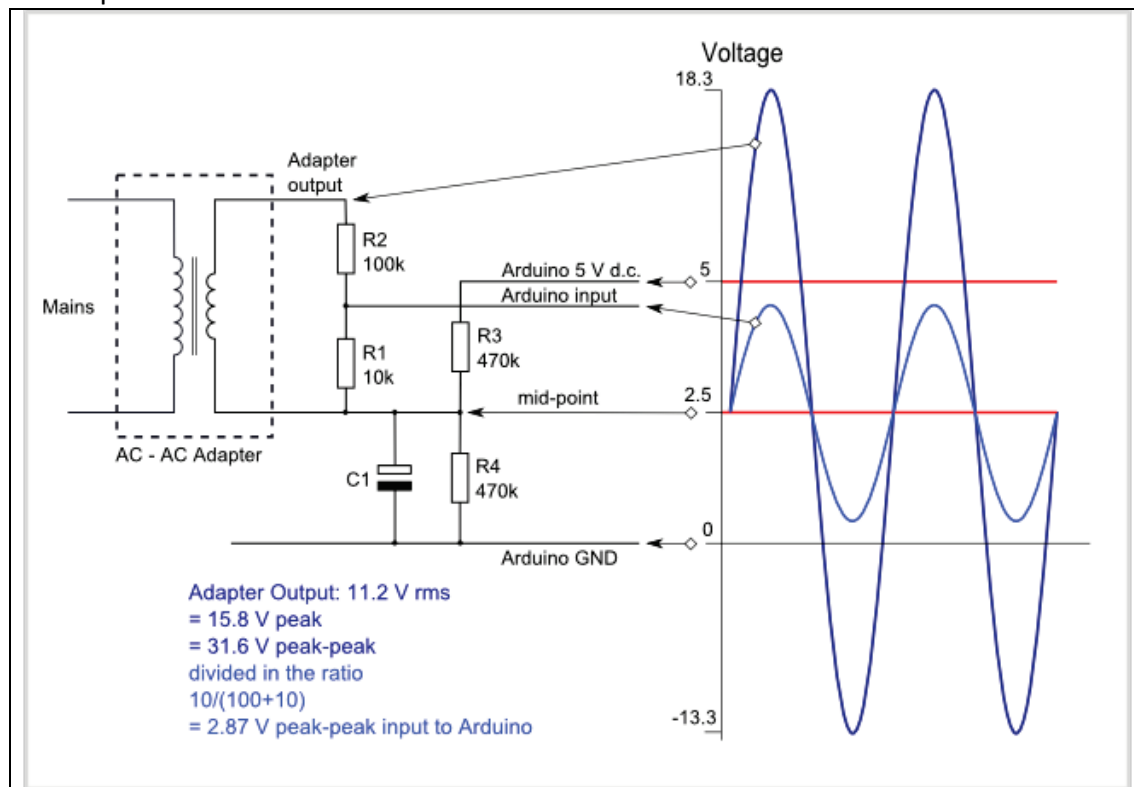
Σύνδεση του αισθητήρα με το Arduino



Εικόνα 5-14 Αναλυτικό κύκλωμα για τη μέτρηση ρεύματος από CT αισθητήρα (openenergymonitor, n.d.)

Αφού φτιάξουμε το παραπάνω κύκλωμα που στην ουσία πρόκειται για έναν διαιρέτη τάσης, με λίγες γραμμές κώδικα και με τη βοήθεια της βιβλιοθήκης EmonLib από της <http://openenergymonitor.org/> (<https://github.com/openenergymonitor/EmonLib>) μπορούμε να μετρήσουμε το ρεύμα που διαρρέεται από έναν αγωγό.

Στη συνέχεια μπορούμε να προχωρήσουμε το κύκλωμά μας με σκοπό να μπορούμε να μετράμε και τάση.

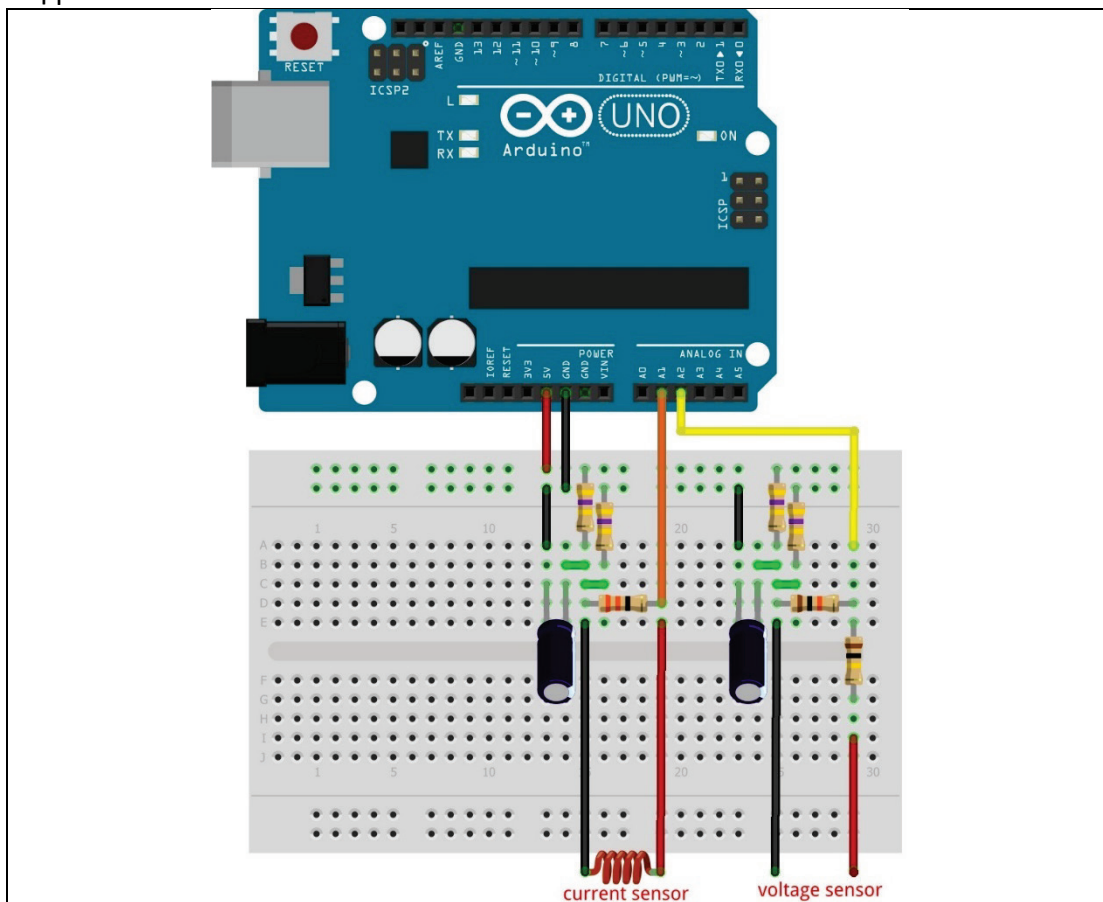


Εικόνα 5-15 Αναλυτικό κύκλωμα για τη μέτρηση ρεύματος από το Arduino (openenergymonitor, n.d.)

Για τον υπολογισμό της πραγματικής ισχύος, της φαινόμενης ισχύος και του συντελεστή ισχύος, είναι απαραίτητο να πραγματοποιηθεί μέτρηση της τάσης εναλλασσόμενου ρεύματος. Μια ασφαλής μέθοδος για την πραγματοποίηση αυτής της μέτρησης είναι με τη χρήση ενός AC-AC μετασχηματιστή, ο οποίος παρέχει απομόνωση μεταξύ των δύο επιπέδων τάσης εναλλασσόμενου ρεύματος. (πηγή: <https://learn.openenergymonitor.org/electricity-monitoring/ctac/how-to-build-an-arduino-energy-monitor>) Για να προσαρμόσουμε το σήμα εξόδου του προσαρμογέα σε μια μορφή κύματος που να πληροί τις απαιτήσεις, πρέπει να επιδιώξουμε μια κυματομορφή με θετική αιχμή χαμηλότερη των 5V (ή 3.3V στην περίπτωση του emonTx), με μια αρνητική κορυφή υψηλότερη του 0V. Για να επιτευχθεί αυτό, μπορούμε να ρυθμίσουμε μια κλίμακα κάτω από το σήμα και να προσθέσουμε μια μετατόπιση, εξασφαλίζοντας ότι δεν υπάρχει αρνητική συνιστώσα. Για τη μείωση της κυματομορφής προς τα κάτω, χρησιμοποιούμε έναν διαιρέτη τάσης που συνδέεται στους ακροδέκτες του προσαρμογέα, ενώ για την προσθήκη της μετατόπισης χρησιμοποιούμε μια πηγή τάσης που παράγεται από έναν άλλο διαιρέτη τάσεως, ο οποίος συνδέεται στην τροφοδοσία του Arduino. Οι αντιστάσεις R1 και R2 λειτουργούν ως διαιρέτες τάσης για τη μείωση της

ισχύος του μετασχηματιστή, ενώ οι R3 και R4 παρέχουν την απαραίτητη τάση πόλωσης. Ο πυκνωτής C1 διευκολύνει το σήμα AC να φτάσει στο γείωση. Η τάση που παρέχεται από τις R3 και R4 πρέπει να είναι το ήμισυ της τάσης τροφοδοσίας του Arduino, εξασφαλίζοντας ότι είναι ίσες. Αν το Arduino λειτουργεί στα 5V, η κυματομορφή του κυκλώματος θα έχει μια θετική κορυφή 3.65V και μια αρνητική κορυφή 1.35V, πληροί δηλαδή τις απαιτήσεις των αναλογικών εισόδων του Arduino και αφήνει αρκετό περιθώριο για ασφάλεια. Το συνδυασμένο ζευγάρι αντιστάσεων R1 και R2, με τις τιμές 10k και 100k αντίστοιχα, λειτουργεί ικανοποιητικά στα 3.3V, όπως απαιτείται από το emonTx, με μια θετική κορυφή των 2.8V και αρνητική κορυφή των 0.5V. (πηγή: <https://learn.openenergymonitor.org/electricity-monitoring/ctac/how-to-build-an-arduino-energy-monitor>)

Σύνδεση με το Arduino



Εικόνα 5-16 Συνδεσμολογία σεναρίου μέτρησης κατανάλωσης (με μετρητή τάσης και ρεύματος) με Arduino

Ένας πολύ απλός κώδικας που μετρά τάση και ρεύμα και μας τυπώνει την πραγματική ισχύ, την φαινόμενη ισχύ, τον συντελεστή ισχύος, την ενεργό τάση και το ενεργό ρεύμα είναι ο εξής. Όλοι οι υπολογισμοί πραγματοποιούνται στην βιβλιοθήκη EmonLib.

```
#include "EmonLib.h"           // Include Emon Library
EnergyMonitor emon1;           // Create an instance

void setup()
{
  Serial.begin(9600);

  emon1.voltage(2, 234.26, 1.7); // Voltage: input pin, calibration, phase_shift
```

```

    emon1.current(1, 111.1); // Current: input pin, calibration.
}

void loop()
{
    emon1.calcVI(20,2000); // Calculate all. No.of crossings, time-out
    emon1.serialprint(); // Print out all variables
}

```

Κώδικας 5-13 Κώδικας σεναρίου

Υπολογισμός Ισχύος και Ενέργειας

Φαινόμενη ισχύς S

Η ισχύς που καταναλώνεται από μια συσκευή ανά μονάδα χρόνου (δευτερόλεπτο) ονομάζεται φαινόμενη ισχύς (S) και μετριέται σε Volt-Ampere (VA) και kiloVolt-Ampere (kVA). Η φαινόμενη ισχύς ($S = V * I$) είναι το γινόμενο της τάσης (V) και της έντασης (I). Το σύνολο της φαινόμενης ισχύος μιας συσκευής χωρίζεται σε δύο υποσύνολα:

- την ενεργό ισχύ (P)
- την άεργο ισχύ (Q)

Ενεργός ισχύς P

Η ενεργός ισχύς (P) μετριέται σε kiloWatt (kW) και αναπαριστά την πραγματική ηλεκτρική ενέργεια που παράγεται ανά δευτερόλεπτο και αντιστοιχεί στο σύνολο της ενέργειας που πραγματικά μετατρέπεται σε χρήσιμο έργο.

Άεργος ισχύς Q

Το δεύτερο υποσύνολο της φαινόμενης ισχύος ονομάζεται άεργος ισχύς (Q) και μετριέται σε Volt-Ampere-reactive (VAr) και kiloVAr. Ονομάζεται άεργος ισχύς επειδή, αντίθετα με την ενεργό ισχύ, δεν μετατρέπεται σε χρήσιμο έργο αλλά αντανακλάται από τη συσκευή και επιστρέφεται στο ηλεκτρικό δίκτυο με διαφορετική μορφή.

Η άεργος ισχύς παρουσιάζεται όταν υπάρχουν επαγωγικά ή χωρητικά στοιχεία (πηνία ή πυκνωτές) ως φορτία στο κύκλωμα που διακινείται το εναλλασσόμενο ρεύμα. Αποτελεί μη επιθυμητή κατάσταση, καθώς δεν παράγει κάποιο έργο και επιβαρύνει τα δίκτυα μεταφοράς ηλεκτρικής ενέργειας, προκαλώντας επιπλέον θερμικές απώλειες λόγω της πρόσθετης ποσότητας ρεύματος. Η ύπαρξή του μπορεί να μειώσει την ενεργό ισχύ (P) και, συνεπώς, την εκμεταλλεύσιμη ισχύ. Η χρέωση της άεργου ισχύος προς τον καταναλωτή προβλέπεται από τον πάροχο, εξαιρουμένων των οικιακών καταναλωτών. (πηγή: <https://learn.openenergymonitor.org/electricity-monitoring/ctac/how-to-build-an-arduino-energy-monitor>)

Ηλεκτρική Ενέργεια

Η ηλεκτρική ενέργεια που καταναλώνεται από μία ηλεκτρική συσκευή ή παρέχεται από μία ηλεκτρική πηγή υπολογίζεται από τον τύπο της Ισχύος $W = P * t$

Όπου P η ισχύς σε W , t ο χρόνος σε S και W η ενέργεια σε J

Η ηλεκτρική ενέργεια υπολογίζεται από το γινόμενο της ηλεκτρικής ισχύς επί το χρόνο. Αν στον προηγούμενο τύπο αντικαταστήσουμε την ηλεκτρική ισχύ P με το γινόμενο $P=U \cdot I$ τότε έχουμε:
 $W=U \cdot I \cdot t$

Όπου W η ηλεκτρική ενέργεια σε J (Joule) , U η τάση σε V (Volt), I η ένταση του ρεύματος σε A (Ampere), t ο χρόνος σε s (δευτερόλεπτα)

Επειδή η μονάδα J είναι πολύ μικρή, στην Ηλεκτροτεχνία χρησιμοποιείται για τη μέτρηση της ηλεκτρικής ενέργειας η μονάδα KWh (κιλοβατώρα). Η κιλοβατώρα ορίζεται ως η ενέργεια που καταναλώνεται από μία συσκευή ισχύος 1KW όταν λειτουργεί επί μία ώρα. Προκύπτει από τον τύπο $W = P \cdot t$ αν θέσουμε $P=1KW$ και $t=1 h$.

W (Ενέργεια) = $1KW \times 1h = 1 KWh$

(πηγή: <https://learn.openenergymonitor.org/electricity-monitoring/ctac/how-to-build-an-arduino-energy-monitor>)

Υπολογισμός Ισχύος στο Arduino

Η πραγματική ισχύς αντιπροσωπεύει το μέσο όρο της στιγμιαίας ισχύος. Για να υπολογιστεί η στιγμιαία ισχύς, πολλαπλασιάζουμε τη στιγμιαία μέτρηση της τάσης με τη μέτρηση του στιγμιαίου ρεύματος. Συγκεντρώνουμε τις στιγμιαίες μετρήσεις ισχύος για έναν καθορισμένο αριθμό δειγμάτων και τις διαιρούμε με αυτόν τον αριθμό δειγμάτων: (πηγή:

<https://learn.openenergymonitor.org/electricity-monitoring/ctac/how-to-build-an-arduino-energy-monitor>)

```
for (n=0; n // inst_voltage and inst_current calculation from raw ADC)
  inst_power = inst_voltage * inst_current;
  sum_inst_power += inst_power;
}
real_power = sum_inst_power / number_of_samples;
```

Κώδικας 5-14 Κώδικας υπολογισμού ισχύος

(RMS)Voltage

Πρώτα υψώνουμε εις το τετράγωνο, στη συνέχεια υπολογίζουμε τον μέσο όρο και, τέλος, την τετραγωνική ρίζα του μέσου όρου των τετραγώνων, έτσι γίνεται:

```
for (n=0; n //inst_voltage calculation from raw ADC)
  squared_voltage = inst_voltage * inst_voltage;
  sum_squared_voltage += squared_voltage;
}
mean_square_voltage = sum_squared_voltage / number_of_samples;
root_mean_square_voltage = sqrt(mean_square_voltage);
```

Κώδικας 5-15 Κώδικας υπολογισμού (RMS)Voltage

(RMS) Current

```
for (n=0; n //inst_current calculation from raw ADC)
  squared_current = inst_current * inst_current;
  sum_squared_current += squared_current;
```

```
}  
mean_square_current = sum_squared_current / number_of_samples;  
root_mean_square_current = sqrt(mean_square_current);
```

Κώδικας 5-16 Κώδικας υπολογισμού (RMS) Current

Φαινόμενη Ισχύς

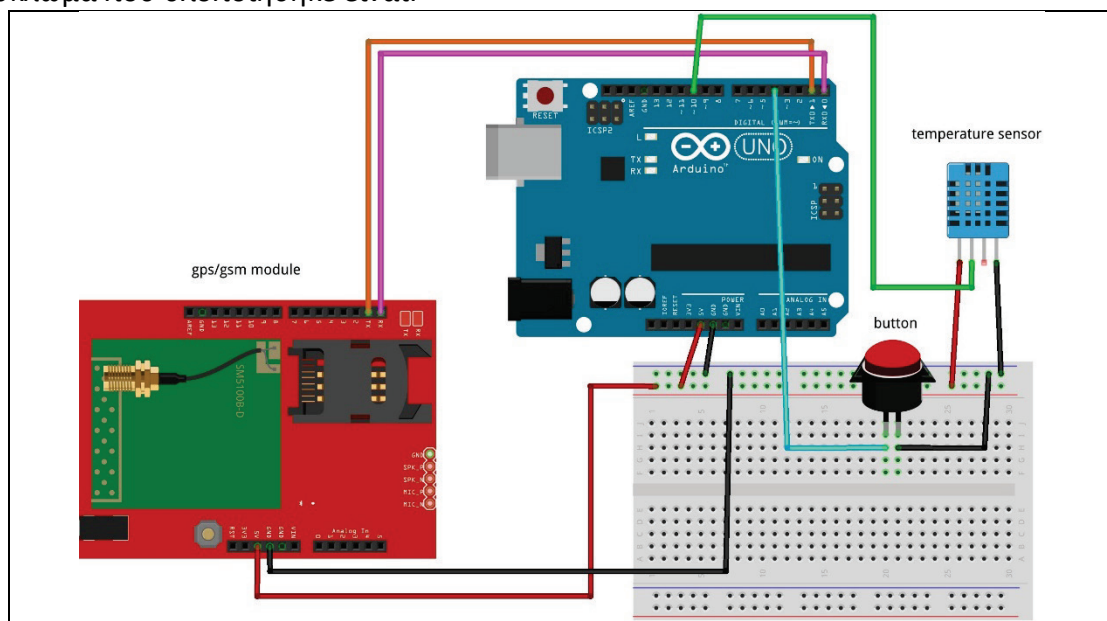
```
apparent_power = root_mean_square_voltage * root_mean_square_current;  
void loop()  
{  
  delay (1000);  
  float supplyVoltage = emon1.Vrms; //extract Vrms into Variable  
  float Irms = emon1.Irms; //extract Irms into Variable  
  float KW=(Irms* supplyVoltage)/1000;  
  Energy=Energy+(KW)*(millis()-last_measurement_timestamp)/(1000.0*3600.0);  
  //Calculate kilowatt hours used  
  float Cost=(Energy*0.0946);
```

Κώδικας 5-17 Κώδικας υπολογισμού φαινόμενης ισχύς

5.15 ΑΝΙΧΝΕΥΣΗ ΤΟΠΟΘΕΣΙΑΣ ΚΑΤΟΙΚΙΔΙΟΥ Ή ΑΤΟΜΟΥ ΜΕ ΕΙΔΙΚΕΣ ΑΝΑΓΚΕΣ, ΚΟΥΜΠΙΟΥ ΕΚΤΑΚΤΟΥ ΑΝΑΓΚΗΣ ΚΑΙ ΣΥΝΘΗΚΩΝ ΠΕΡΙΒΑΛΛΟΝΤΟΣ ΧΩΡΟΥ

Το παρόν σενάριο αφορά την ανίχνευση τοποθεσίας κατοικιδίου ή ατόμου με ειδικές ανάγκες, συνθηκών περιβάλλοντος χώρου και την υλοποίηση κουμπιού εκτάκτου ανάγκης.

Το κύκλωμα που υλοποιήθηκε είναι:



Εικόνα 5-17 Σενάριο ανίχνευσης τοποθεσίας κατοικίδιου ή ατόμου με ειδικές ανάγκες, κουμπιού έκτακτου ανάγκης και συνθηκών περιβάλλοντος χώρου

Βιβλιοθήκη TinyGps

Η βιβλιοθήκη είναι σχεδιασμένη για να δέχεται δεδομένα της μορφής NMEA. Πρακτικά μετατρέπει τα δεδομένα που εισέρχονται από το Gps σε στοιχεία που μπορούμε να καταλάβουμε όπως: γεωγραφικό μήκος, ταχύτητα κ.α. . Ακολουθεί μία λεπτομερειακή αναφορά στις εντολές και τις δομές, των οποίων έγινε χρήση:

- TinyGPS gps: Δίνουμε ένα όνομα στην υπηρεσία, εδώ «gps».
- Gps.encode(): Με αυτή την εντολή η βιβλιοθήκη καταλαβαίνει αν έχουν έρθει σωστού τύπου δεδομένα, δηλαδή αν έχει δεχτεί σωστό σήμα η κεραία του gps. Η συγκεκριμένη συνάρτηση θα δώσει αληθή απάντηση όταν έχει έγκυρα δεδομένα τα οποία στην συνέχεια θα επεξεργαστούν.
- gps.f_get_position(&lat, &lon, &age): Υπολογίζει την γεωγραφική θέση δηλαδή, γεωγραφικό πλάτος, μήκος και τον χρόνο τον οποίο χρειάστηκαν τα δεδομένα για να επεξεργαστούν(τάξης ms).
- gps.f_speed_kmph(): Υπολογίζει την ταχύτητα .
- gps.crack_datetime(&year, &month, &day, &hour, &minute, &second, &hundredths, &age): Εξάγει την ημερομηνία και την ώρα που πήρε την τελευταία μέτρηση.
- gps.satellites(): Υπολογίζει τους δορυφόρους με τους οποίους έχει επικοινωνία η κεραία.
- gps.f_altitude(): Υπολογίζει το υψόμετρο όπου βρίσκεται η κεραία.

Ο κώδικας για το σενάριο ανίχνευσης τοποθεσίας κατοικίδιου ή ατόμου με ειδικές ανάγκες, κουμπιού έκτακτου ανάγκης και συνθηκών περιβάλλοντος χώρου:

```
#include <SoftwareSerial.h>
#include <TinyGPS.h>
#include "DHT.h"
#define button 4 //ακίδα που θα συνδεθεί το κουμπί
//αισθητήρας θερμοκρασίας-υγρασίας
#define sensor_pin 10 //ακίδα που θα συνδεθεί ο αισθητήρας
#define DHTTYPE DHT22 //μοντέλο αισθητήρα χρησιμοποιείται
DHT dht(sensor_pin,DHTTYPE);
//gps
TinyGPS gps;
//SoftwareSerial
SoftwareSerial ss(2, 3); //ακίδες στις οποίες θα στέλνει και θα δέχεται δεδομένα η SoftwareSerial

unsigned long fix_age, time, date, speed, course; //δήλωση των μεταβλητών που είναι απαραίτητες για τον υπολογισμό των δεδομένων χρόνου απόκρισης, χρόνου, ημερομηνίας, ταχύτητας και πορείας
void printFloat(double f, int digits = 2);
```

```

void setup()
{
  pinMode(button, INPUT_PULLUP); //ορίζεται το κουμπί σαν τύπος input_pullup
  Serial.begin(115200);
  ss.begin(9600);
}
void loop()
{
  //επικοινωνία με την πλακέτα sim808 με τις AT-Commands
  bool newData = false;
  ss.println("AT");
  ss.println("AT+CGNSPWR=1");
  ss.println("AT+CGNSTST=1");
  //συνθήκη που θα ελέγχει αν η πλακέτα δέχεται σωστού τύπου δεδομένα
  for (unsigned long start = millis(); millis() - start < 1000;)
  {
    while (ss.available())
    {
      char c = ss.read();
      Serial.write(c); //gps nmea string
      //όταν μεταδοθούν σωστά δεδομένα η μεταβλητή newData θα γίνει αληθής και θα αρχίσουν
      //οι υπολογισμοί
      if (gps.encode(c))
        newData = true;
    }
  }
  //gps
  if (newData)
  {
    float latitude, longitude;
    unsigned long age;
    float fkmph;
    int year;
    byte month, day, hour, minute, second, hundredths;
    //υπολογίζονται οι τιμές που εισέρχονται από την κεραία του gps
    //το γεωγραφικό πλάτος και μήκος, η ταχύτητα, οι δορυφόροι, το υψόμετρο, η ώρα και η ημε-
    //ρομηνία
    //τυπώνονται στην σειριακή του ide για ευκολότερη απεικόνιση
    gps.f_speed_kmph();
    gps.f_get_position(&latitude, &longitude, &age);
    gps.crack_datetime(&year, &month, &day, &hour, &minute, &second, &hundredths, &age);
    Serial.print("LATITUDE=");
    Serial.print(latitude == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : latitude, 6);
    Serial.print("LONTIDUTE=");
    Serial.print(longitude == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : longitude, 6);
    Serial.print("SATELLITES=");
    Serial.print(gps.satellites() == TinyGPS::GPS_INVALID_SATELLITES ? 0 : gps.satellites());
  }
}

```

```

Serial.print("DATE:"); Serial.print(static_cast<int>(month)); Serial.print("/");
Serial.print(static_cast<int>(day)); Serial.print("/"); Serial.print(year);
Serial.print("TIME:"); Serial.print(static_cast<int>(hour+3)); Serial.print(":");
Serial.print(static_cast<int>(minute)); Serial.print(":"); Serial.print(static_cast<int>(second));
Serial.print("."); Serial.print(static_cast<int>(hundredths));
Serial.print("ALTITUDE:"); Serial.print(gps.f_altitude());
Serial.print("KMPH:"); printFloat(gps.f_speed_kmph());
Serial.println();
//ημερομηνία και ώρα σε String
//θα μετατραπούν οι μεταβλητές του χρόνου και της ημερομηνίας σε τύπου string ώστε να
σταλθούν έπειτα με σωστό τρόπο στη βάση δεδομένων
String string_a=String(year);
String string_b=String(month);
String string_c=String(day);
String string_d=String(hour+3);
String string_e=String(minute);
String string_f=String(second);
//όλες οι μεταβλητές συγχωνεύονται σε ένα string
String datetime= string_a + "-" + string_b + "-" + string_c + "/" + string_d + ":" + string_e + ":" +
string_f;
//τυπώνεται σαν μία μεταβλητή πλέον στην σειριακή
Serial.println(datetime);
//απεικόνιση στους χάρτες
//ορίζονται τρεις σταθερές σε μορφή string για την χρησιμοποίηση τους στην υπηρεσία των
χαρτών μέσω της βάσης δεδομένων
String info_a = "LAT=";
String info_b = "LON=";
String info_c = "-";
//μήνυμα με τις συντεταγμένες εάν πατηθεί το μπουτόν έκτακτης ανάγκης
static uint8_t lastBtnState = HIGH;
uint8_t state = digitalRead(button);
if (state != lastBtnState) {
    lastBtnState = state;
    if (state == LOW) {
//μεταβλητές σε string
//εάν πατηθεί το κουμπί το γεωγραφικό πλάτος και μήκος μετατρέπονται σε string για να μπο-
ρούν να σταλούν μέσω μηνύματος
String lat=String(latitude == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : latitude, 6);
String lon=String(longitude == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : longitude, 6);
delay(100);
//στέλνονται οι AT-Commands για να εκτελεστεί η λειτουργία του μηνύματος και δηλώνεται το
κινητό τηλέφωνο του χρήστη
ss.println("AT+CMGF=1");
delay(500);
ss.println("AT+CMGS=\"+3069.....\"");
delay(500);
ss.println("SOS");
ss.println("LATITUDE= ");
ss.print(lat);

```

```

ss.println(" ");
ss.println("LONGTITUDE= ");
ss.print(lon);
ss.println("http://maps.google.com/maps?q=loc:");
ss.print(lat);
ss.print(",");
ss.print(lon);
delay(200);
//αποστολή τερματικού κωδικού ώστε να καταλάβει η πλακέτα ότι τερματίστηκε το μήνυμα
ss.write(26);
Serial.println("SMS SENT");
}
}
//μέτρηση απόστασης και μήνυμα
//υπολογίζει την απόσταση του κατοικίδιου ή του ατόμου με ειδικές ανάγκες από μία τοποθε-
σία που έχει ορίσει ο χρήστης
//αν αυτή η απόσταση είναι μεγαλύτερη από αυτή που θέτουμε, ενεργοποιείται το συμβάν
//δηλώνουμε ξανά το γεωγραφικό πλάτος και μήκος
latitude == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : latitude, 6;
longitude == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : longitude, 6;
//δηλώνουμε τις απαραίτητες μεταβλητές και μέσω μαθηματικών τύπων υπολογίζεται η από-
σταση από το σημείο που έχουμε θέσει με latitude=21.728059 και longitude=38.238707
float X = ((21.728059 - longitude) * 24301 * cos(latitude)) / 360.0;
float Y = ((38.238707 - latitude) * 24280) / 360.0;
float X2 = X * X;
float Y2 = Y * Y;
float distance = sqrt(X2 + Y2);
//Μετατρέπει την απόσταση από μίλια σε χιλιόμετρα
float km = distance*1.609344;
//Μετατρέπει τα χιλιόμετρα σε μέτρα και τα τυπώνει στη σειριακή.
float met = km*1000;
Serial.print(met);
//Η συνθήκη παρακολουθεί αν η απόσταση του κατοικίδιου γίνει μεγαλύτερη από τριάντα μέ-
τρα και ενημερώνει το χρήστη με το αντί-στοιχο μήνυμα. Θα μπορούσε πολύ εύκολα να αλλα-
χθεί το μήνυμα ώστε να ενημερώνει για την απόσταση ενός ανθρώπου, όπως και η τιμή των
μέτρων που ελέγχονται.
if (met>30){
String meters = String(met);
delay(100);
ss.println("AT+CMGF=1");
delay(500);
ss.println("AT+CMGS=\"+306949465421\");
delay(500);
ss.println("Dog is away from home ");
ss.print(meters);
ss.println("meters");
delay(200);
ss.write(26);
}

```

Κώδικας 5-18 Κώδικας ανίχνευσης τοποθεσίας κατοικίδιου ή ατόμου με ειδικές ανάγκες, κουμπιού έκτακτου ανάγκης και συνθηκών περιβάλλοντος χώρου
(Συμμεωνίδης, Υλοποίηση υπηρεσιών απομακρυσμένης εποπτείας/ελέγχου βασισμένων στις πλατφόρμες Arduino/Google Maps- Charts και στις τεχνολογίες 3G/GPS/GPRS 2018)

5.16 ΣΕΝΑΡΙΟ ΚΑΤΑΣΚΕΥΗΣ ΣΤΑΘΜΟΥ ΚΑΙΡΙΚΩΝ ΣΥΝΘΗΚΩΝ (WEATHER STATION)

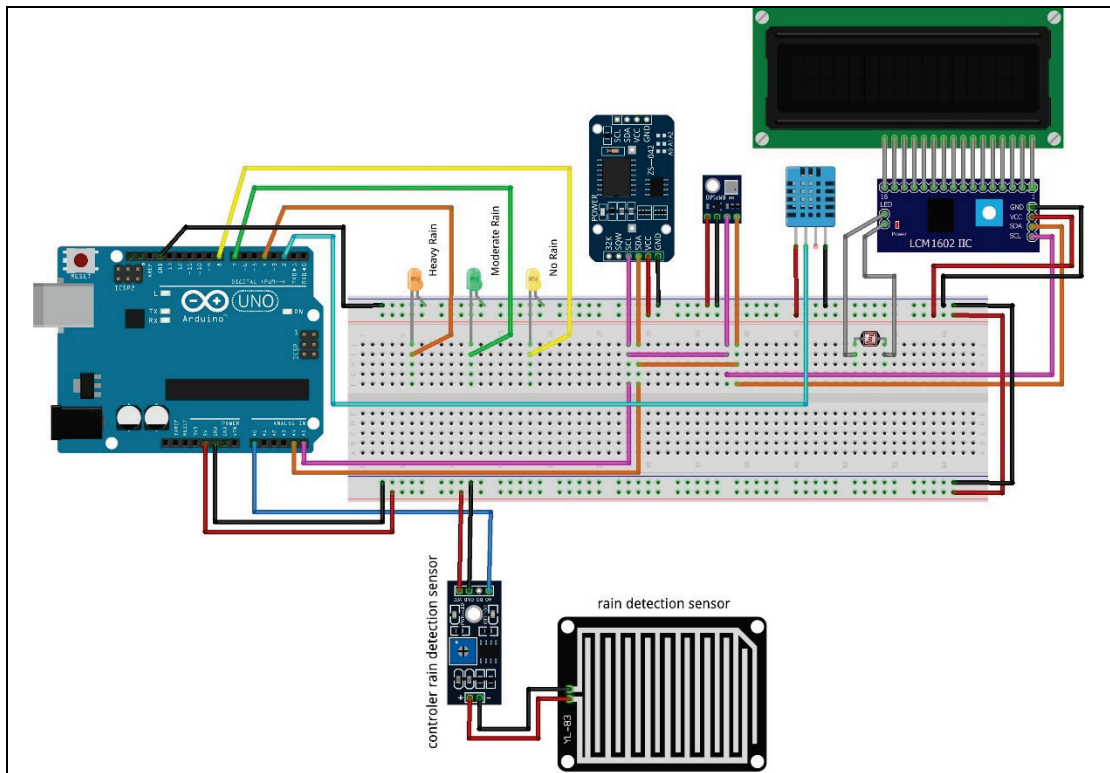
Τα κυριότερα χαρακτηριστικά του σταθμού καιρικών συνθηκών είναι:

- Ανίχνευση βροχής
- Εσωτερική εξωτερική θερμοκρασία
- Εσωτερική εξωτερική υγρασία
- Βαρομετρική πίεση
- Εκτύπωση ημερομηνίας και ώρας
- LCD οθόνη

Οι αισθητήρες που χρησιμοποιήθηκαν για την ανίχνευση της βροχής είναι οι παρακάτω:

- YL-83 – Rain Detection Sensor
- YL-83 - Controler Rain Detection Sensor
- 3 leds
- BMP180 - Barometric Pressure/Temperature/Altitude Sensor
- DHT22 - temperature-humidity sensor
- RTC DS1307

Το κύκλωμα του κατασκευής σταθμού καιρικών συνθηκών (weather station) είναι:



Εικόνα 5-18 Σενάριο κατασκευής σταθμού καιρικών συνθηκών (weather station)

Ο κώδικας για το σενάριο κατασκευής σταθμού καιρικών συνθηκών (weather station):

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <DS1307RTC.h>
#include "DHT.h"

LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE); // Set the LCD I2C address

int led_heavy_rain=4; // pin led Heavy Rain
int led_moderate_rain=7; // pin led Moderate Rain
int led_no_rain=8; // pin led No Rain

void setup()
{
  Serial.begin(9600);
  pinMode(2, OUTPUT);
  Wire.begin();
  RTC.begin();
  if (pressure.begin())
    Serial.println("BMP180 init success");
  else
  {
    Serial.println("BMP180 init fail\n\n");
    while(1);
  }
}
```

```

    lcd.begin(20,4);
    lcd.backlight();
    lcd.clear();
}
void loop()
{
    lcd.clear();
    //DATE-TIME
    tmElements_t tm;
    if (RTC.read(tm)) {
        printDate(5,0,tm);
        printTime(6,1,tm);
    } else {
        if (RTC.chipPresent()) {

        } else {
        }
        delay(5000);
    }
    delay(1000);
    int8_t humidity = dht.readHumidity(); // read humidity
    int16_t temp = dht.readTemperature(TEMPTYPE); // read temperature
    lcd.setCursor(0,2);
    lcd.print("HUMIDITY: ");
    lcd.setCursor(11,2);
    lcd.print(humidity); // Print humidity
    lcd.setCursor(12, 2);
    lcd.print(" % ");
    lcd.setCursor(0, 3);
    lcd.print("TEMP:");
    lcd.setCursor(10, 3);
    lcd.print(temp); // Print temperature
    lcd.setCursor(13, 3);
    delay(1000);
    lcd.clear();
    //rain
    if(analogRead(0)<300) {
        Serial.println("Heavy Rain");
        digitalWrite(led_heavy_rain,HIGH);
        lcd.setCursor(0,0);
        lcd.print("Heavy Rain");
    }
    else if(analogRead(0)<500) {
        Serial.println("Moderate Rain");
        digitalWrite(led_moderate_rain,HIGH);}
    lcd.setCursor(0,0);
    lcd.print("Moderate Rain");
    else {
        Serial.println("No Rain");
    }
}

```

```
digitalWrite(led_no_rain,HIGH);}
lcd.setCursor(0,0);
lcd.print("No Rain");
delay(1000);
char status;
double T,P,p0,a;
Serial.println();
Serial.print("Provided Altitude: ");
Serial.print(ALTITUDE,0); // Print altitude
Serial.print(" meters, ");
Serial.print(ALTITUDE*3.28084,0);
Serial.println(" feet");
lcd.setCursor(0,1);
lcd.print("Provided Altitude:");
lcd.setCursor(0,2);
lcd.print(ALTITUDE,0); // Print altitude
lcd.setCursor(9,2);
lcd.print(" meters, ");
lcd.setCursor(0,3);
lcd.print(ALTITUDE*3.28084,0);
lcd.setCursor(9,3);
lcd.print(" feet");

delay(2000);
lcd.clear();
status = pressure.startTemperature();
if (status != 0)
{
    delay(status);
    status = pressure.getTemperature(T);
    if (status != 0)
    {
        Serial.print("temperature: ");
        Serial.print(T,2); // Print temp
        Serial.print(" deg C, ");
        Serial.print((9.0/5.0)*T+32.0,2);
        Serial.println(" deg F");
        lcd.setCursor(0,0);
        lcd.print("temperature: ");
        lcd.setCursor(0,1);
        lcd.print(T,2);
        lcd.setCursor(9,1);
        lcd.print(" deg C, ");
        lcd.setCursor(0,2);
        lcd.print((9.0/5.0)*T+32.0,2);
        lcd.setCursor(9,2);
        lcd.print(" deg F");
    }
    delay(2000);
    lcd.clear();
}
```

```

status = pressure.startPressure(3);
if (status != 0)
{
    delay(status);
    status = pressure.getPressure(P,T);
    if (status != 0)
    {
        Serial.print("absolute pressure: ");
        Serial.print(P,2); // Print pressure
        Serial.print(" mb, ");
        Serial.print(P*0.0295333727,2);
        Serial.println(" inHg");
        lcd.setCursor(0,0);
        lcd.print("absolute pressure: ");
        lcd.setCursor(0,1);
        lcd.print(P,2);
        lcd.setCursor(9,1);
        lcd.print(" mb, ");
        lcd.setCursor(0,2);
        lcd.print(P*0.0295333727,2);
        lcd.setCursor(9,2);
        lcd.print(" inHg");
    }
    delay(2000);
    lcd.clear();
    p0 = pressure.sealevel(P,ALTITUDE);
    Serial.print("relative (sea-level) pressure: ");
    Serial.print(p0,2); // Print pressure
    Serial.print(" mb, ");
    Serial.print(p0*0.0295333727,2);
    Serial.println(" inHg");
    lcd.setCursor(0,0);
    lcd.print("relative (sea-level) pressure: ");
    lcd.setCursor(0,1);
    lcd.print(p0,2);
    lcd.setCursor(9,1);
    lcd.print(" mb, ");
    lcd.setCursor(0,2);
    lcd.print(p0*0.0295333727,2);
    lcd.setCursor(9,2);
    lcd.print(" inHg");
}
delay(2000);
lcd.clear();
a = pressure.altitude(P,p0);
Serial.print("computed altitude: ");
Serial.print(a,0); // Print altirude
Serial.print(" meters, ");
Serial.print(a*3.28084,0);
Serial.println(" feet");
lcd.setCursor(0,0);

```

```

        lcd.print("computed altitude: ");
        lcd.setCursor(0,1);
        lcd.print(a,0);
        lcd.setCursor(9,1);
        lcd.print(" meters, ");
        lcd.setCursor(0,2);
        lcd.print(a*3.28084,0);
        lcd.setCursor(9,2);
        lcd.print(" feet");
    delay(2000);
    lcd.clear();
    }
    else Serial.println("error retrieving pressure measurement\n");
    }
    else Serial.println("error starting pressure measurement\n");
    }
    else Serial.println("error retrieving temp measurement\n");
    }
    else Serial.println("error starting temp measurement\n");
    delay(2000);
}

void printTime(int character,int line, tmElements_t tm)
{
    String seconds,minutes;
    lcd.setCursor(character,line);
    lcd.print(tm.Hour);
    lcd.print(":");
    if(tm.Minute<10)
    {
        minutes = "0"+String(tm.Minute);
        lcd.print(minutes);
    }else
    {
        lcd.print(tm.Minute);
    }
    lcd.print(":");
    if(tm.Second<10)
    {
        seconds = "0"+String(tm.Second);
        lcd.print(seconds);
    }else
    {
        lcd.print(tm.Second);
    }
}

void printDate(int character,int line, tmElements_t tm)
{
    lcd.setCursor(character,line);
    lcd.print(tm.Month);

```

```
lcd.print("/");  
lcd.print(tm.Day);  
lcd.print("/");  
lcd.print(tmYearToCalendar(tm.Year));  
}
```

Κώδικας 5-19 Κώδικας ανίχνευσης τοποθεσίας κατοικίδιου ή ατόμου με ειδικές ανάγκες, κουμπιού έκτακτου ανάγκης και συνθηκών περιβάλλοντος χώρου

ΚΕΦΑΛΑΙΟ 6

ΔΙΑΣΥΝΔΕΣΗ ΜΕ ΤΟ ΔΙΑΔΙΚΤΥΟ ΚΑΙ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

6.1 ΓΕΝΙΚΑ

Σε αυτό το κεφάλαιο θα δούμε την διασύνδεση με το Διαδίκτυο και δημιουργία της κατάλληλης βάσης δεδομένων. Θα γίνει αναφορά στις δηλώσεις του Arduino ethernet shield που χρειάζονται για την σύνδεση του με το διαδίκτυο και στην δημιουργία της Βάσης Δεδομένων για την αποστολή των μετρήσεων από τους αισθητήρες και καταστάσεων των ενεργοποιητών.

6.2 ΔΙΑΣΥΝΔΕΣΗ ΜΕ ΤΟ ΔΙΑΔΙΚΤΥΟ

Για την διασύνδεση του Arduino ethernet shield με το Διαδίκτυο δηλώνεται η διεύθυνση IP του server από τον υπολογιστή μας και την IP Address του server που δίνει το Arduino. Επίσης θα χρειαστεί και το byte mac Address του Arduino.

Ο Κώδικας σύνδεσης του Ethernet, ο οποίος τρέχει και σε όλα τα Arduino (με αλλαγές στις μεταβλητές των αισθητήρων/ενεργοποιητών) του έξυπνου σπιτιού είναι:

```
#include <SPI.h>           //βιβλιοθήκη για την σειριακή επικοινωνία του Arduino με το Ethernet
#include <Ethernet.h>      //βιβλιοθήκη Ethernet
int Wait_between_dumps = 0;
byte mac[] = { 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF };
EthernetClient client;
IPAddress ip(192, 168, 0, 2); // διεύθυνση ip του Arduino
IPAddress serv(192, 168, 0, 3); // διεύθυνση ip του υπολογιστή
void setup()
```

```
{
Ethernet.begin(mac, ip); // initialize Ethernet device
void loop()
{
  db();
}
void db() { // διάβασμα ενδεικτικών τιμών για αποστολή στην βάση δεδομένων
float temp = dht.temperature;
float hum = dht.humidity;
int fan =digitalRead(FAN);
Wait_between_dumps++;
if (Wait_between_dumps >=10) //αποστολή τιμών ανά 2 sec
{
  if (client.connect(serv, 80)) { //Connecting at the IP address and port we saved before
Serial.println("connected");
client.print("GET /ethernet/data.php?"); //Connecting and Sending values to database
client.print("temperature=");
client.print(temp);
client.print("&&humidity=");
client.print(hum);
client.print("&&FAN=");
client.print(fan );
Wait_between_dumps = 0;
client.println( " HTTP/1.1");
client.print( "Host: " );
client.println(serv);
client.println( "Connection: close" );
client.println();
client.println();
client.stop(); //Close the connection
}
else {
// if you didn't get a connection to the server:
Serial.println("connection failed");
}
}
}
```

Κώδικας 6-1 Κώδικας αρχικοποίησης του ethernet shield και σύνδεσης με το διαδίκτυο

6.3 ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

Η βάση δεδομένων δημιουργήθηκε για την καταγραφή των μετρήσεων από τους αισθητήρες και καταστάσεων των ενεργοποιητών και την αποθήκευση τους για περεταίρω και αξιοποίηση τους από τον χρήστη.

Υλοποιήθηκε στην MySQL, η οποία είναι μία ελεύθερη βάση δεδομένων. Για να επιτευχθεί η καταγραφή των τιμών θα χρειαστεί να συνδεθεί το Arduino με το Ethernet shield. Επίσης θα χρειαστούν οι βιβλιοθήκες Ethernet.h και SPI.h για την λειτουργία του Ethernet και την σύνδεση του στη βάση δεδομένων.

Αφού έχει στηθεί στον υπολογιστή μας ο MySQL server και ο Apache Server, μπορούμε να υποποιήσουμε μία βάση δεδομένων με το όνομα “myhome”, η οποία αποτελείται από τους πίνακες π.χ. “home”, “greenhouse”, “pet”. Για την δημιουργία της βάσης ανοίγουμε την σελίδα phpMyAdmin στον τοπικό Web Server.

myhome.home

Ο παρακάτω πίνακας χρησιμοποιείται για την καταχώριση των μετρήσεων των αισθητήρων και την καταγραφή της ώρας κατά την οποία έγινε εγγραφή για το σπίτι. Ακολουθεί αναλυτική περιγραφή των πεδίων του:

- **id**: Αποτελεί το αναγνωριστικό κάθε εγγραφής. Σε κάθε νέα εγγραφή που πραγματοποιείται έχει προγραμματιστεί να αυξάνεται κατά 1 (Πρόσθετη επιλογή AUTO_INCREMENT). Είναι ακέραιος έως 11 ψηφίων.
- **event**: Γίνεται εγγραφή της ακριβής ημερομηνίας και ώρας (π.χ. 18/07/2019 20:38:12) ώστε ο χρήστης να γνωρίζει ακριβώς πότε έγινε η κάθε καταγραφή δεδομένων.
- **temperature**: Αποθηκεύει την τιμή της θερμοκρασίας περιβάλλοντος.
- **humidity**: Αποθηκεύει την τιμή της υγρασίας περιβάλλοντος.
- **FAN**: Αποθηκεύει την τιμή του εξαερισμού.
- **FIRE**: Αποθηκεύει την τιμή ανίχνευσης της φωτιάς.
- **FLOOD**: Αποθηκεύει την τιμή ανίχνευσης της πλημμύρας.
- **GARAGE**: Αποθηκεύει την τιμή της γκαραζόπορτας όταν αυτή είναι ανοιχτή ή κλειστή.
- **ANTLIA**: Αποθηκεύει την τιμή της αντλίας ανελκυστήρα με ή χωρίς νερό.
- **porta**: Αποθηκεύει την τιμή της πόρτας.

Ο κώδικας SQL του πίνακα “home”:

```
CREATE DATABASE IF NOT EXISTS `myhome` DEFAULT CHARACTER SET latin1 COLLATE latin1_swedish_ci;
USE `myhome`;
CREATE TABLE `home` (
  `id` int(11) NOT NULL COMMENT 'unique ID',
  `event` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT 'Event Date and Time',
  `temperature` varchar(10) NOT NULL COMMENT 'humidity',
  `humidity` varchar(10) NOT NULL COMMENT 'raindrop',
```

```
`FAN` int(10) NOT NULL COMMENT 'FAN',
`FIRE` int(10) NOT NULL COMMENT 'FIRE',
`FLOOD` int(10) NOT NULL COMMENT 'FLOOD',
`GARAGE` int(10) NOT NULL COMMENT 'GARAGE',
`ANTLIA` int(10) NOT NULL COMMENT 'ANTLIA',
`porta` int(10) NOT NULL COMMENT 'porta'
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
ENGINE = InnoDB;
```

Κώδικας 6-2 Κώδικας SQL του πίνακα “home”

Πίνακας myhome.greenhouse

Ο παρακάτω πίνακας χρησιμοποιείται για την καταχώριση των μετρήσεων των αισθητήρων και την καταγραφή της ώρας κατά την οποία έγινε εγγραφή για τον κήπο/θερμοκήπιο. Ακολουθεί αναλυτική περιγραφή των πεδίων του:

- id: Αποτελεί το αναγνωριστικό κάθε εγγραφής. Σε κάθε νέα εγγραφή που πραγματοποιείται έχει προγραμματιστεί να αυξάνεται κατά 1 (Πρόσθετη επιλογή AUTO_INCREMENT). Είναι ακέραιος έως 11 ψηφίων.
- event: Γίνεται εγγραφή της ακριβής ημερομηνίας και ώρας (π.χ. 2015-05-19 19:03:12) ώστε ο χρήστης να γνωρίζει ακριβώς πότε έγινε η κάθε καταγραφή δεδομένων.
- light: Αποθηκεύει την τιμή της φωτεινότητας του περιβάλλοντος.
- raindrop: Αποθηκεύει την τιμή του αισθητήρα βροχής.
- moisture: Αποθηκεύει την τιμή της υγρασίας χώματος.
- temperature: Αποθηκεύει την τιμή της θερμοκρασίας του αέρα.
- humidity: Αποθηκεύει την τιμή της υγρασίας του αέρα.
- tank: Αποθηκεύει την τιμή του ύψους της δεξαμενής νερού.
- fan: Αποθηκεύει την κατάσταση του fan.
- van: Αποθηκεύει την κατάσταση του valve.

Ο κώδικας SQL του πίνακα “ greenhouse”:

```
CREATE DATABASE IF NOT EXISTS `myhome` DEFAULT CHARACTER SET latin1 COLLATE latin1_swedish_ci;
USE `myhome`;
CREATE TABLE `greenhouse` (
`id` int(11) NOT NULL COMMENT 'unique ID',
`event` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT 'Event Date and Time',
`light` varchar(10) NOT NULL COMMENT 'light',
`raindrop` varchar(10) NOT NULL COMMENT 'raindrop',
`moisture` varchar(10) NOT NULL COMMENT 'moisture',
`temperature` varchar(10) NOT NULL COMMENT 'temperature',
`humidity` varchar(10) NOT NULL COMMENT 'humidity',
`tank` varchar(10) NOT NULL COMMENT 'tank',
`fan` varchar(10) NOT NULL COMMENT 'fan',
`van` varchar(10) NOT NULL COMMENT 'van'
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Κώδικας 6-3 Κώδικας SQL του πίνακα “greenhouse”

Πίνακας myhome.pet

Ο παρακάτω πίνακας χρησιμοποιείται για την καταχώριση των μετρήσεων των αισθητήρων και την καταγραφή της ώρας κατά την οποία έγινε εγγραφή για την παρακολούθηση του κατοικιδίου. Ακολουθεί αναλυτική περιγραφή των πεδίων του:

- **id**: Αποτελεί μία στήλη του πίνακα και η τιμή της μεταβλητής παράγεται αυτόματα από τη βάση κάθε φορά που εισέρχεται μία γραμμή με καινούργια δεδομένα. Σε αυτή τη περίπτωση έχουμε επιλέξει να αυξάνεται κατά 1 με κάθε νέα εγγραφή.
- **Datetime**: Καταγράφει τον ακριβή χρόνο που εισέρχεται η νέα γραμμή δεδομένων. Στην περίπτωση μας τροφοδοτείται από τον μικροελεγκτή. Συνήθως επιλέγεται η αυτόματη εγγραφή τιμών από τη βάση.
- **Lat**: Αποθηκεύει την τιμή του γεωγραφικού πλάτους.
- **Lon**: Αποθηκεύει την τιμή του γεωγραφικού μήκους.
- **Speed**: Αποθηκεύει την τιμή της ταχύτητας.
- **Satellites**: Αποθηκεύει την τιμή των δορυφόρων.
- **Altitude**: Αποθηκεύει την τιμή του υψόμετρου.
- **Text1**: Αποθηκεύει τους χαρακτήρες «LAT=» που χρειάζονται για την απεικόνιση στους χάρτες
- **Text2**: Αποθηκεύει τους χαρακτήρες «LON=» που χρειάζονται για την απεικόνιση στους χάρτες
- **Text3**: Αποθηκεύει τον χαρακτήρα «-» που χρειάζεται για την απεικόνιση στους χάρτες
- **Temperature**: Αποθηκεύει την τιμή της θερμοκρασίας.
- **Humidity**: Αποθηκεύει την τιμή της υγρασίας.

Ο κώδικας SQL του πίνακα “pet”:

```
CREATE DATABASE IF NOT EXISTS `myhome` DEFAULT CHARACTER SET latin1 COLLATE latin1_swedish_ci;
USE `myhome`;
CREATE TABLE `pet` (
  `id` int(11) NOT NULL,
  `Datetime` datetime NOT NULL,
  `Lat` decimal(10,6) NOT NULL,
  `Lon` decimal(10,6) NOT NULL,
  `Speed` decimal(8,2) NOT NULL,
  `Satellites` int(10) NOT NULL,
  `Altitude` decimal(5,2) NOT NULL,
  `Text1` char(10) NOT NULL,
  `Text2` char(10) NOT NULL,
  `Text3` char(10) NOT NULL,
  `Temperature` decimal(8,2) NOT NULL,
  `Humidity` decimal(8,2) NOT NULL
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Κώδικας 6-4 Κώδικας SQL του πίνακα "pet"

6.4 ΣΥΝΔΕΣΗ MYSQL ME PHP

Το πρώτο αρχείο θα ονομαστεί **connection.php** το οποίο θα χρησιμοποιηθεί για την σύνδεση με την MySQL.

Ο κώδικας του **connection.php**:

```
<?php
$username = "root";// enter your username for mysql
$password = "password";//enter your password for mysql
$host = "localhost"; // this is usually "localhost" unless your database resides on a different
server
$db_name = "myhome"; // enter your database name
$con = mysqli_connect ($host, $username, $pass);
$db = mysqli_select_db ( $con, $db_name );
?>
```

Κώδικας 6-5 Κώδικας αρχείου connection.php

Φόρτωση δεδομένων

myhome.home

Το δεύτερο αρχείο θα ονομαστεί **data1.php** το οποίο θα χρησιμοποιηθεί για την αποστολή των τιμών των αισθητήρων από το Arduino προς την βάση δεδομένων και το table myhome.home.

Ο κώδικας του **data1.php**:

```
<?php
include ('connection.php');
$sql_insert = "INSERT INTO home (tempera-
ture,humidity,FAN,FIRE,FLOOD,GARAGE,ANTLIA,porta) VALUES ('".$_GET["temperature"]."',
".$_GET["humidity"]."', '".$_GET["FAN"]."', '".$_GET["FIRE"]."', '".$_GET["FLOOD"]."',
".$_GET["GARAGE"]."', '".$_GET["ANTLIA"]."', '".$_GET["porta"]."')";
if(mysqli_query($con,$sql_insert))
{
echo "Done";
mysqli_close($con);
}
else
{
echo "error is ".mysqli_error($con );
}
?>
```

myhome.greenhouse

Το τρίτο αρχείο θα ονομαστεί **data2.php** το οποίο θα χρησιμοποιηθεί για την αποστολή των τιμών των αισθητήρων από το Arduino προς την βάση δεδομένων και το table myhome. greenhouse.

Ο κώδικας του **data2.php**:

```
<?php
include ('connection.php');
$sql_insert = "INSERT INTO home (light,raindrop,moisture,temperature,humidity,tank,fan,van)
VALUES ('".$_GET["light"]."', '".$_GET["raindrop"]."', '".$_GET["moisture"]."',
".$_GET["temperature"]."', '".$_GET["humidity"]."', '".$_GET["tank"]."', '".$_GET["fan"]."',
".$_GET["van"]."')";
if(mysqli_query($con,$sql_insert))
{
echo "Done";
mysqli_close($con);
}
else
{
echo "error is ".mysqli_error($con );
}
?>
```

myhome.pet

Το τέταρτο αρχείο θα ονομαστεί **data3.php** το οποίο θα χρησιμοποιηθεί για την αποστολή των τιμών των αισθητήρων από το Arduino προς την βάση δεδομένων και το table myhome. pet.

Ο κώδικας του **data3.php**:

```
<?php
include ('connection.php');
$sql_insert = "INSERT INTO pet (Lat,Lon,Speed,Satellites,Altitude,Text1,Text2,Text3, Temperature, Humidity) VALUES ('".$_GET["Lat"]."', '".$_GET["Lon"]."', '".$_GET["Speed"]."',
".$_GET["Satellites"]."', '".$_GET["Altitude"]."', '".$_GET["Text1"]."', '".$_GET["Text2"]."',
".$_GET["Text3"]."', '".$_GET["Temperature"]."', '".$_GET["Humidity"]."')";
if(mysqli_query($con,$sql_insert))
{
echo "Done";
mysqli_close($con);
}
else
{
echo "error is ".mysqli_error($con );
}
}
```


?>

Κώδικας 6-8 Κώδικας αρχείου data3.php

ΚΕΦΑΛΑΙΟ 7

ΑΠΟΜΑΚΡΥΣΜΕΝΟΣ ΕΛΕΓΧΟΣ, ΙΣΤΟΣΕΛΙΔΕΣ, ΓΡΑΦΗΜΑΤΑ

7.1 ΓΕΝΙΚΑ

Για την άμεση και πιο ευανάγνωστη μορφή των μεταβλητών θα δημιουργηθούν γραφήματα και γραφικά στοιχεία με την βοήθεια των Google Charts, που έχουν αναλυθεί σε προηγούμενη ενότητα. Η google παρέχει δωρεάν κώδικα για μια μεγάλη ποικιλία διαφορετικών τύπων διαγραμμάτων. Θα παρουσιαστεί ο κώδικας με τον οποίο δημιουργήθηκαν και θα δοθούν απλές οδηγίες για την τροποποίηση τους με βάση τις ανάγκες του χρήστη.

7.2 SCRIPTS

7.2.1 GOOGLE GAUGE

Αρχικά θα σχεδιαστεί το γραφικό μορφής μετρητή(gauge). Ο κώδικας που θα παρουσιαστεί αφορά ένα μετρητή.

«gauge.php»

Δηλώνονται τα στοιχεία της βάσης

```
<?php
// Prepare variables for database connection

$dbusername = "root";
$dbpassword = "password";
$server = "localhost";
```

Κώδικας 7-1 Κώδικας google gauge, στοιχεία βάσης

Γίνεται σύνδεση στη βάση.

```
// Connect to your database
$mysqli = new mysqli($server, $dbusername, $dbpassword, "myhome");
if($mysqli->connect_error)
die('Connect Error (' . mysqli_connect_errno() . ') ' . mysqli_connect_error());
```

Κώδικας 7-2 Κώδικας google gauge, σύνδεση με τη βάση

Δημιουργούμε το query στη βάση όπου ζητείται η τιμή της θερμοκρασίας.

```
// Execute SQL statement

$result = $mysqli->query("SELECT temperature FROM myhome.greenenergy ORDER BY id
DESC LIMIT 1");

if ($result->num_rows > 0) {

// output data of each row
while($row = $result->fetch_assoc()) {
    $LValue = $row["temperature"];

}
}
else {
    $LValue = 0;

}
?>
```

Κώδικας 7-3 Κώδικας google gauge, δημιουργία query στη βάση όπου ζητείται η τιμή της θερμοκρασίας

Ακολουθεί ο html κώδικας που είναι απαραίτητος για την παρουσίαση του γραφικού στην ιστοσελίδα που έχει δημιουργηθεί.

```
<!doctype html>
<html>
<head><meta http-equiv="Content-Type" content="text/html; charset=iso8859-7">

<title>Gauge </title>
```

```
<link rel="stylesheet" href="../fonts/fonts.css">
<link rel="stylesheet" href="buttons.css">
<script src="gauge.js"></script>
```

Κώδικας 7-4 Κώδικας google gauge, κώδικας html

Κύριο κομμάτι ιστοσελίδας.

```
</head>
<body style="background: # 001433">
<div style="width:70%;font-size:28pt;text-align:center;color:#ffffff;text-align:left;">Temperature</div>


<div style="width:100%;font-size:19pt;color:#cca300;text-align:left;"><strong> Temperature
</div>
</div>
</div>
```

Κώδικας 7-5 Κώδικας google gauge, Κύριο κομμάτι κώδικα html

Παρακάτω παρουσιάζεται η σχεδίαση του μετρητή όπου δηλώνεται το πλάτος το ύψος του περιγράμματος, ο τίτλος, το χρώμα αλλά και τις κλίμακες της βελόνας στο εσωτερικό του.

```
<canvas data-type="radial-gauge"
  data-width="250"
  data-height="250"
  data-units="&deg;C"
  data-title="Temperature"
  data-value="<?php echo "\". $TValue. "\"\n"?>
  data-animate-on-init="false"
  data-animated-value="false"
  data-min-value="-50"
  data-max-value="50"
  data-major-ticks="[-50,-40,-30,-20,-10,0,10,20,30,40,50]"
  data-minor-ticks="2"
  data-stroke-ticks="true"
  data-highlights=[
    {"from": -50, "to": 0, "color": "rgba(0,0, 255, .3)"},
    {"from": 0, "to": 20, "color": "rgba(0, 153, 51,1)"},
    {"from": 35, "to": 50, "color": "rgba(77, 0, 0,1)"},
    {"from": 20, "to": 35, "color": "rgba(255, 51, 51,1)"}
  ]
  data-color-plate="#001433"
  data-color-major-ticks="#f5f5f5"
  data-color-minor-ticks="#ddd"
  data-color-title="#fff"
  data-color-units="#ccc"
```

```
data-color-numbers="#eee"
data-color-needle-start="rgba(240, 128, 128, 1)"
data-color-needle-end="rgba(255, 160, 122, .9)"
data-value-box="true"
data-animation-rule="bounce"
data-animation-duration="500"
data-border-outer-width="3"
data-border-middle-width="3"
data-border-inner-width="3"
</canvas>
```

Κώδικας 7-6 Κώδικας google gauge, σχεδίαση μετρητή

Το απαραίτητο script για την δημιουργία του γραφικού είναι το εξής:

```
<script>
if (!Array.prototype.forEach) {
  Array.prototype.forEach = function(cb) {
    var i = 0, s = this.length;
    for (; i < s; i++) {
      cb && cb(this[i], i, this);
    }
  }
}

document.fonts && document.fonts.forEach(function(font) {
  font.loaded.then(function() {
    if (font.family.match(/Led/)) {
      document.gauges.forEach(function(gauge) {
        gauge.update();
        gauge.options.renderTo.style.visibility = 'visible';
      });
    }
  });
});

var timers = [];

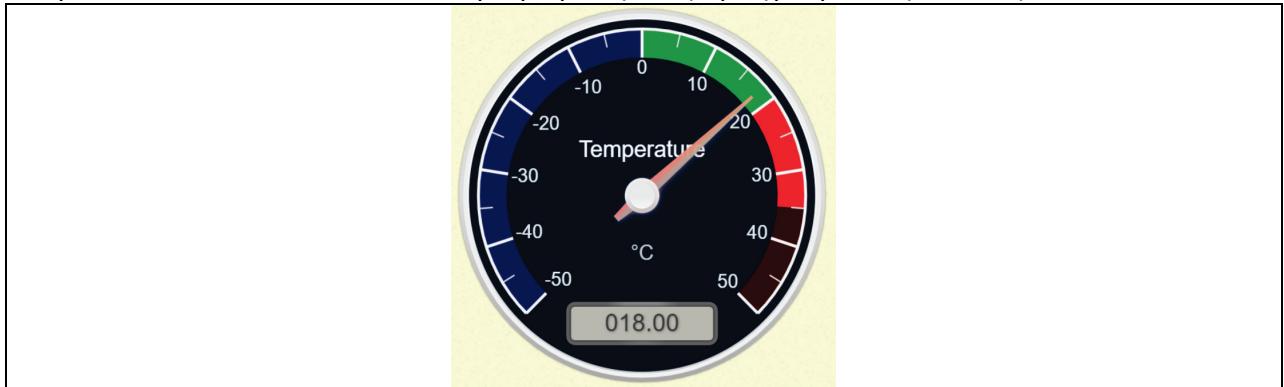
function animateGauges() {
  document.gauges.forEach(function(gauge) {
    timers.push(setInterval(function() {
      gauge.value = Math.random() *
        (gauge.options.maxValue - gauge.options.minValue) +
        gauge.options.minValue;
    }, gauge.animation.duration + 50));
  });
}

function stopGaugesAnimation() {
```

```
timers.forEach(function(timer) {  
    clearInterval(timer);  
});  
}  
  
function resize() {  
    var size = parseFloat(document.getElementById('gauge-size').value) || 400;  
  
    document.gauges.forEach(function (gauge) {  
        gauge.update({ width: size, height: size });  
    });  
}  
  
function setText() {  
    var text = document.getElementById('gauge-text').value;  
  
    document.gauges.forEach(function (gauge) {  
        gauge.update({ valueText: text });  
    });  
}  
</script>
```

Κώδικας 7-7 Κώδικας google gauge, απαραίτητο script για τη σχεδίαση μετρητή

Παρακάτω ακολουθεί το αποτέλεσμα με βάση τους προηγούμενους κώδικες.



Εικόνα 7-1 Ο μετρητής με την τελευταία μέτρηση της θερμοκρασίας

7.2.2 STATISTICS

«statistics.php»

Στο επόμενο γράφημα που θα δημιουργηθεί παρουσιάζονται οι τελευταίες πέντε τιμές της μεταβλητής που θα επιλέξει ο χρήστης, συναρτήσει του χρόνου. Θα παρουσιαστούν ενδεικτικά δύο γραφήματα αυτό της θερμοκρασίας και του υγρασίας .

Το κομμάτι του html κώδικα που υποδεικνύει τα τριάντα δευτερόλεπτα που ανανεώνεται το γράφημα αλλά και το απαραίτητο script που καλείται από το server της Google.

```
<html>
<head>
  <meta http-equiv="refresh" content="30" /> <!--refresh selidas-->
  <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
  <script type="text/javascript">
    google.charts.load('current', {'packages':['line']});
    google.charts.setOnLoadCallback(drawChart);
```

Κώδικας 7-8 Κώδικας statistics

Η συνάρτηση `drawChart()` δημιουργεί το γράφημα και δηλώνει ποιες στήλες καλεί από τη βάση.

```
function drawChart() {
  var data = new google.visualization.DataTable();
  data.addColumn('datetime', 'Datetime');
  data.addColumn('number', 'Temperature');
  data.addColumn('number', 'Humidity');
```

Κώδικας 7-9 Κώδικας statistics, συνάρτηση drawChart()

Το κομμάτι του php κώδικα ορίζει τα στοιχεία της βάσης και πραγματοποιεί τη σύνδεση.

```
<?php
// Prepare variables for database connection

$dbusername = "root";
$dbpassword = "password"; $server = "localhost";

// Connect to your database
$mysqli = new mysqli($server, $dbusername, $dbpassword, "myhome");
if($mysqli->connect_error)
  die('Connect Error (' . mysqli_connect_errno() . ') ' . mysqli_connect_error());
```

Κώδικας 7-10 Κώδικας statistics, php κώδικα ορίζει τα στοιχεία της βάσης και πραγματοποιεί τη σύνδεση

Αφού γίνει το query στη βάση «σπάει» την μεταβλητή `datetime` σε έτος, μήνα, μέρα, ώρα και δευτερόλεπτα για να δηλωθεί κατάλληλα στο script παρακάτω. Με την εντολή `DESC LIMIT 5` ορίζεται πόσες τιμές θα υπάρχουν στο διάγραμμα .

```
$result = $mysqli->query("SELECT YEAR(Datetime) AS year, MONTH(Datetime) AS month,
DAY(Datetime) AS day, HOUR(Datetime) AS hour, MINUTE(Datetime) AS minute, SEC-
OND(Datetime) AS sec, temperature, humidity FROM myhome.greenenergy ORDER BY
Datetime DESC LIMIT 5");

if ($result->num_rows > 0) {

  // output data of each row
```



```

while($row = $result->fetch_assoc()) {
    echo "data.addRow([new Date(".$row["year"].",".$row["month"]."-
1,$row["day"].",".$row["hour"].",".$row["minute"].",".$row["sec"]."),
".$row["temperature"].",".$row["humidity"]);\\n";
}
}
else {
}
?>

```

Κώδικας 7-11 Κώδικας statistics, query στη βάση

Η κάθε μεταβλητή περιέχει τις επιλογές απεικόνισής της. Όπως ο τίτλος, η μονάδα μέτρησης και ο τρόπος που θα παρουσιάζεται.

```

var temperature_options = {
    chart: {
        title: 'Temperature vs Time',
        subtitle: 'measured in C'
    },
    hAxis: { format: 'HH:mm:ss a' },
    width: 600,
    height: 300
};
var humidity_options = {
    chart: {
        title: 'Humidity vs Time',
        subtitle: 'measured in %'
    },
    hAxis: { format: 'HH:mm:ss a' },
    width: 600,
    height: 300
};

```

Κώδικας 7-12 Κώδικας statistics, μεταβλητές και επιλογές απεικόνισής τους

Δηλώνεται σε ποιες στήλες θα αντιστοιχεί η κάθε μεταβλητή.

```

var temperature_chart = new
google.charts.Line(document.getElementById('temperature_div'));

var temperature_view = new google.visualization.DataView(data);
temperature_view.setColumns([0,4]);
temperature_chart.draw(temperature_view,
google.charts.Line.convertOptions(temperature_options));
//-----
var humidity_chart = new google.charts.Line(document.getElementById('humidity_div'));

var humidity_view = new google.visualization.DataView(data);
humidity_view.setColumns([0,5]);

```

```

    humidity_chart.draw(humidity_view,
google.charts.Line.convertOptions(humidity_options));

//-----
}
</script>

```

Κώδικας 7-13 Κώδικας statistics, αντιστοίχιση στηλών και μεταβλητής

Μετά το τέλος του javascript υπάρχει το υπόλοιπο κομμάτι κώδικα html που εκτός από το να δίνει ένα τίτλο στη σελίδα παρουσιάζει και τα γραφικά μέσα σε έναν πίνακα για καλύτερη στοίχιση.

```

<title>Statistics </title>
<body style="background: # 001433">
    <div style="width:100%">
        <div style="width:90%;font-size:30pt;text-align:center;color:#ffffff;text-align:left;">Real-
Time Διαγράμματα </div>
    </div>
    
    
    <div style="width:100%;font-size:19pt;color:#cca300;text-align:left;"><strong>Real-Time
Διαγράμματα
    </div>
    <br>
    <div align="right">

    </div>

    <table style="border-spacing: 15px">
        <tr>
            <td style="border: 1px solid white">
                <div id="temperature_div" style="width: 600px; height: 300px"></div>
            </td>

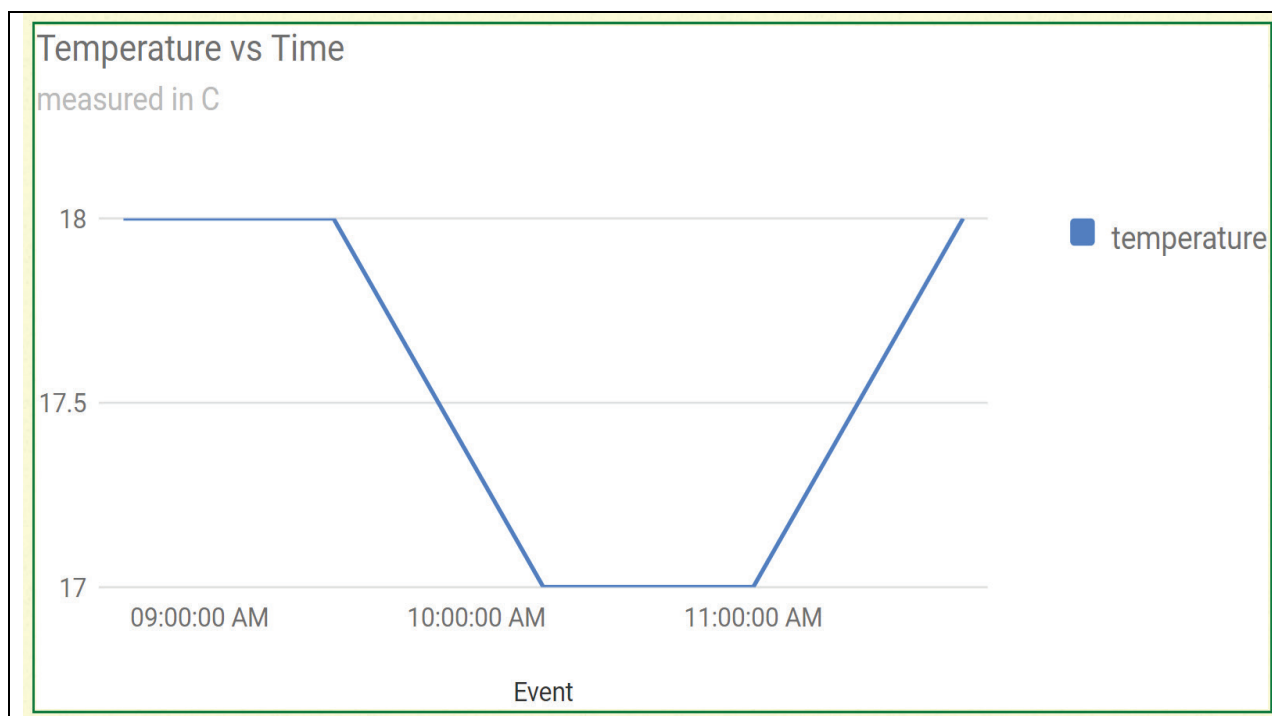
            <td style="border: 1px solid white">
                <div id="humidity_div" style="width: 600px; height: 300px"></div>
            </td>

        </tr>
    </table>
</body>
</html>

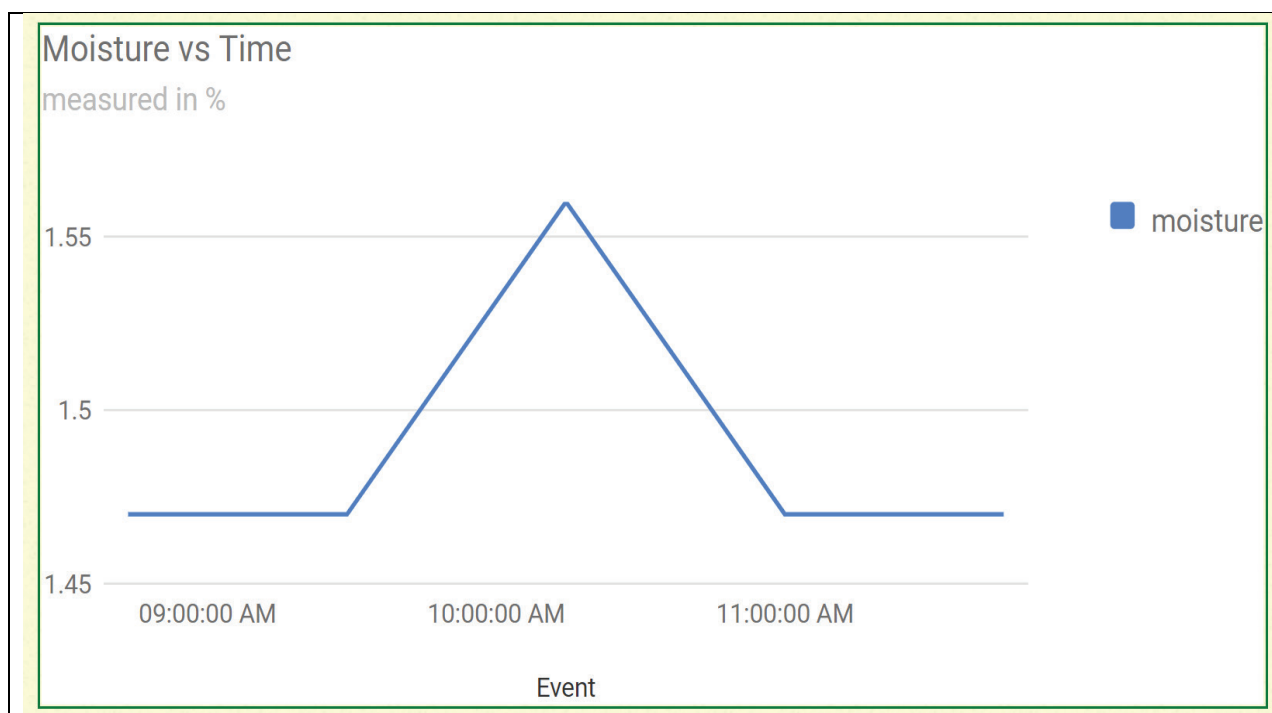
```

Κώδικας 7-14 Κώδικας statistics, υπόλοιπος html κώδικας

Παρακάτω ακολουθεί το αποτέλεσμα με βάση τους προηγούμενους κώδικες.



Εικόνα 7-2 Η γραφική της θερμοκρασίας συναρτήσει του χρόνου



Εικόνα 7-3 Η γραφική της υγρασίας συναρτήσει του χρόνου

7.2.3 WEEKLY STATISTICS

«weekly_statistics.php»

Στα επόμενα γραφήματα που θα δημιουργηθούν παρουσιάζονται η μέση, ελάχιστη και μέγιστη τιμή της μεταβλητής για κάθε ημέρα της τελευταίας βδομάδας. Θα παρουσιαστούν ενδεικτικά δύο γραφήματα αυτό της θερμοκρασίας και του υγρασίας .

Αυτές οι γραφικές αναπαραστάσεις μας δίνουν ένα άμεσο τρόπο παρακολούθησης των μεταβλητών που περιέχει η βάση δεδομένων. Η διαδικασία που θα ακολουθηθεί θα είναι πανομοιότυπη με την προηγούμενη.

Το κομμάτι του html κώδικα που υποδεικνύει τα τριάντα δευτερόλεπτα που ανανεώνεται το γράφημα αλλά και το απαραίτητο script που καλείται από το server της Google.

```
<html>
<head>
  <meta http-equiv="refresh" content="30" />
  <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
  <script type="text/javascript">
    google.charts.load('current', {'packages':['corechart']}));
    google.charts.setOnLoadCallback(drawChart);
```

Κώδικας 7-15 Κώδικας weekly statistics

Η συνάρτηση drawChart() δημιουργεί το γράφημα και δηλώνει ποιες στήλες καλεί από τη βάση. Σε αυτή τη περίπτωση δηλώνονται η ελάχιστη, μέγιστη και μέση τιμή της μεταβλητής ξεχωριστά ώστε να υπολογιστεί με sql κώδικα .

```
function drawChart() {

  var data = new google.visualization.DataTable();
  data.addColumn('datetime', 'event');
  data.addColumn('number', 'Average raindrop');
  data.addColumn('number', 'Max raindrop');
  data.addColumn('number', 'Min raindrop');
  data.addColumn('number', 'Average moisture');
  data.addColumn('number', 'Max moisture');
  data.addColumn('number', 'Min moisture');
  data.addColumn('number', 'Average temperature');
  data.addColumn('number', 'Max temperature');
  data.addColumn('number', 'Min temperature');
  data.addColumn('number', 'Average humidity');
  data.addColumn('number', 'Max humidity');
  data.addColumn('number', 'Min humidity');
  data.addColumn('number', 'Average tank');
  data.addColumn('number', 'Max tank');
  data.addColumn('number', 'Min tank');
```

Κώδικας 7-16 Κώδικας weekly statistics, drawChart()

Έπεται το κομμάτι του κώδικα στο οποίο ορίζονται τα στοιχεία της βάσης και γίνεται η σύνδεση.

```
<?php
```

```
// Prepare variables for database connection
```

```
$dbusername = "root"; // enter database username, I used "arduino" in step 2.2
$dbpassword = "password"; // enter database password, I used "arduinotest" in step 2.2
$server = "localhost";
// Connect to your database
$mysqli = new mysqli($server, $dbusername, $dbpassword, "myhome");
if($mysqli->connect_error)
    die('Connect Error (' . mysqli_connect_errno() . ') ' . mysqli_connect_error());
```

Κώδικας 7-17 Κώδικας weekly statistics, ορισμός στοιχείων βάσης και σύνδεση

Αφού γίνει το query στη βάση «σπάει» την μεταβλητή datetime σε έτος, μήνα, μέρα, ώρα και δευτερόλεπτα για να δηλωθεί κατάλληλα στο script παρακάτω. Με την εντολή DESC LIMIT 7 ορίζεται πόσες τιμές θα υπάρχουν στο διάγραμμα, στην συγκεκριμένη περίπτωση 7.

```
// Execute SQL statement ta grouparw ana imera kai gia kathe mera to min kai to max kai to average kai krataw tis teleytaies 7 meres
```

```
$result = $mysqli->query("SELECT YEAR(event) AS year, MONTH(event) AS month,
DAY(event) AS day, AVG(raindrop),MAX(raindrop),MIN(raindrop),
AVG(moisture),MAX(moisture),MIN(moisture),
AVG(temperature),MAX(temperature),MIN(temperature),
AVG(humidity),MAX(humidity),MIN(humidity), AVG(tank),MAX(tank),MIN(tank) from
myhome.greenenergy group by DAY(event) ORDER BY event DESC limit 7");
```

```
if ($result->num_rows > 0) {
```

```
// output data of each row
```

```
while($row = $result->fetch_assoc()) {
    echo "data.addRow([new Date(".$row["year"].",".$row["month"]."-
1,".$row["day"]."),".$row["AVG(raindrop)"].",".$row["MAX(raindrop)"].",".$row["MIN(raindrop)"]
).",".$row["AVG(moisture)"].",".$row["MAX(moisture)"].",".$row["MIN(moisture)"].",".$row["A
VG(temperature)"].",".$row["MAX(temperature)"].",".$row["MIN(temperature)"].",".$row["A
VG(humidity)"].",".$row["MAX(humidity)"].",".$row["MIN(humidity)"].",".$row["AVG(tank)"].",
".$row["MAX(tank)"].",".$row["MIN(tank)"]."]);\\n";
}
}
else {
}
?>
```

Κώδικας 7-18 Κώδικας weekly statistics, query στη βάση

Η κάθε μεταβλητή περιέχει τις επιλογές απεικόνισής της. Όπως ο τίτλος, η μονάδα μέτρησης και ο τρόπος που θα παρουσιάζεται.

```
var temperature_options = {
    title: 'Temperature (C) Last Week',
    subtitle: 'measured in C',
    hAxis: { format: 'MMM dd, yyyy' },
    seriesType: 'bars',
```

```
width: 600,
height: 300
};

var humidity_options = {
  title: 'Humidity (%) Last Week',
  subtitle: 'measured in %',
  hAxis: { format: 'MMM dd, yyyy' },
  seriesType: 'bars',
  width: 600,
  height: 300
};
```

Κώδικας 7-19 Κώδικας weekly statistics, μεταβλητές και επιλογές απεικόνισής τους

Δηλώνεται σε ποιες στήλες θα αντιστοιχεί η κάθε μεταβλητή. Σε αυτή την περίπτωση δηλώνονται σε κάθε μεταβλητή τέσσερις στήλες που περιέχουν την ημερομηνία, την ελάχιστη, μέγιστη και μέση τιμή.

```
//-----
var temperature_chart = new
google.visualization.ComboChart(document.getElementById('temperature_div'));

var temperature_view = new google.visualization.DataView(data);
temperature_view.setColumns([0,7,8,9]);
temperature_chart.draw(temperature_view, temperature_options);
//-----
var humidity_chart = new
google.visualization.ComboChart(document.getElementById('humidity_div'));

var humidity_view = new google.visualization.DataView(data);
humidity_view.setColumns([0,10,11,12]);
humidity_chart.draw(humidity_view, humidity_options);
//-----
```

Κώδικας 7-20 Κώδικας weekly statistics, αντιστοίχιση στηλών και μεταβλητής

Μετά το τέλος του javascript υπάρχει το υπόλοιπο κομμάτι κώδικα html που εκτός από το να δίνει ένα τίτλο στη σελίδα παρουσιάζει και τα γραφικά μέσα σε έναν πίνακα για καλύτερη στοίχιση.

```
</script>
</head>
<title>Weekly </title>
<body style="background: # 001433">
  <div style="width:100%">
    <div style="width:90%;font-size:30pt;text-align:center;color:# cca300;text-align:left;"><strong> Weekly Statistics</div>
  </div>
```

```

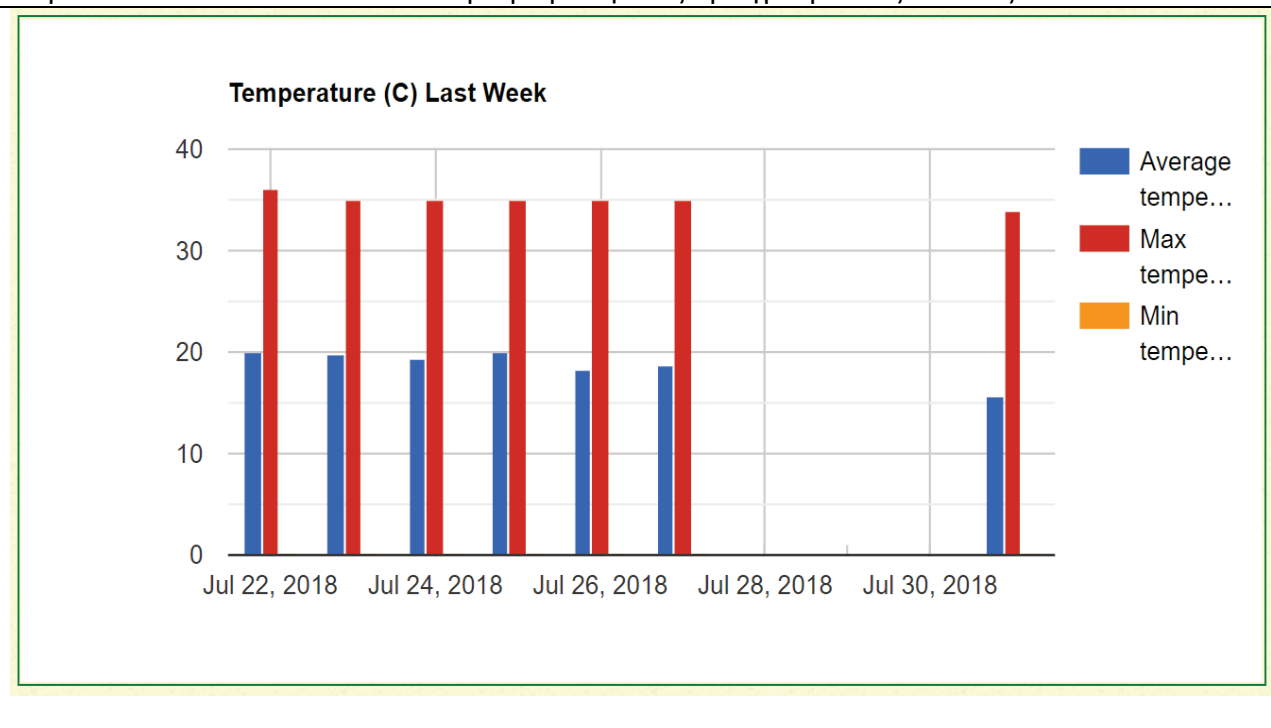


<div style="width:100%;font-size:19pt;color:#ffffff;text-align:left;">Εβδομαδιαίες
Μετρήσεις
</div>
<br>
<div align="right">
</div>
<table style="border-spacing: 15px">
<tr>
<td style="border: 1px solid white">
<div id="temperature_div" style="width: 600px; height: 300px"></div>
</td>
<td style="border: 1px solid white">
<div id="humidity_div" style="width: 600px; height: 300px"></div>
</td>
</tr>
</table>
</body>

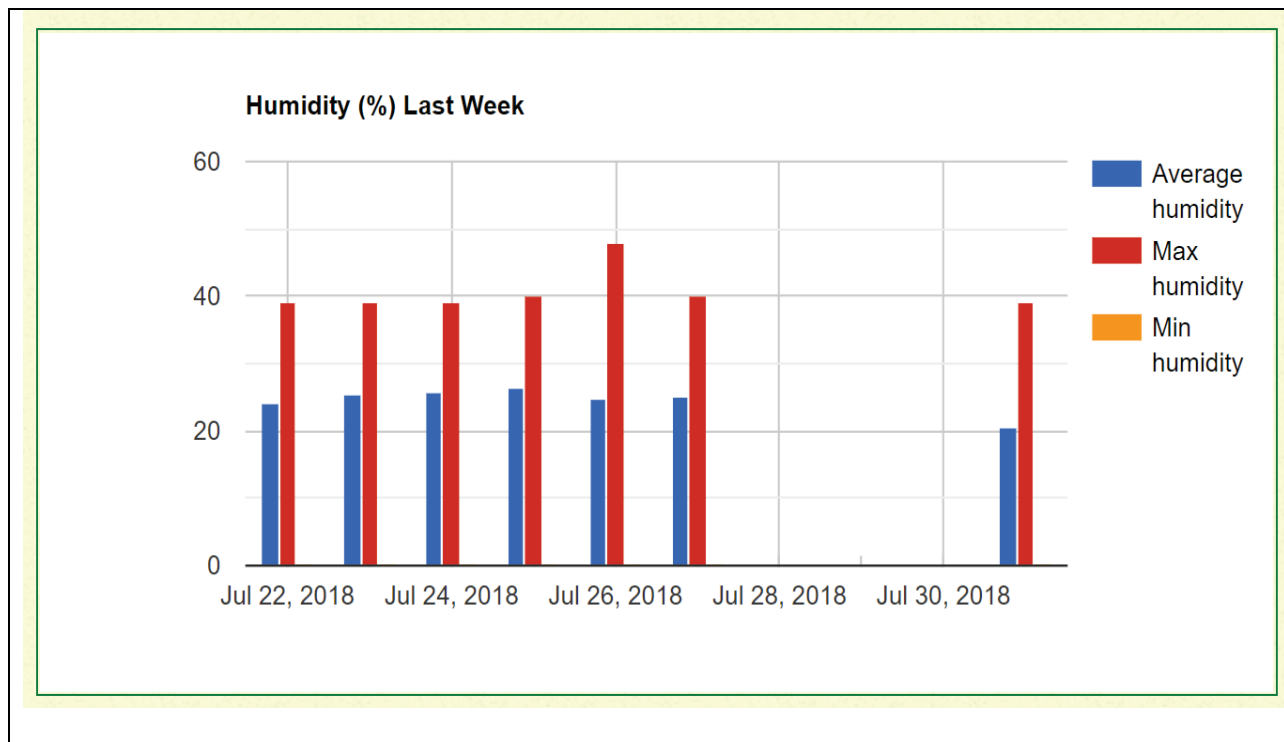
```

Κώδικας 7-21 Κώδικας weekly statistics, υπόλοιπος html κώδικα

Παρακάτω ακολουθεί το αποτέλεσμα με βάση τους προηγούμενους κώδικες.



Εικόνα 7-4 Απεικόνιση της ελάχιστης, μέγιστης και μέσης τιμής της θερμοκρασίας



Εικόνα 7-5 Απεικόνιση της ελάχιστης, μέγιστης και μέσης τιμής της υγρασίας

7.2.4 MONTHLY STATISTICS

«monthly_statistics.php»

Στα επόμενα γραφήματα που θα δημιουργηθούν παρουσιάζονται η μέση, ελάχιστη και μέγιστη τιμή της μεταβλητής για κάθε μήνα του τελευταίου χρόνου. Θα παρουσιαστούν ενδεικτικά δύο γραφήματα αυτό της θερμοκρασίας και της υγρασίας. Αυτές οι γραφικές αναπαραστάσεις μας δίνουν ένα άμεσο τρόπο παρακολούθησης των μεταβλητών που περιέχει η βάση δεδομένων. Η διαδικασία που θα ακολουθηθεί θα είναι πανομοιότυπη με την προηγούμενη.

Το κομμάτι του html κώδικα που υποδεικνύει τα τριάντα δευτερόλεπτα που ανανεώνεται το γράφημα αλλά και το απαραίτητο script που καλείται από το server της Google.

```
<html>
<head>
  <meta http-equiv="refresh" content="30" />
  <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
  <script type="text/javascript">
    google.charts.load('current', {'packages':['corechart']});
    google.charts.setOnLoadCallback(drawChart);
```

Κώδικας 7-22 Κώδικας monthly statistics

Η συνάρτηση drawChart() δημιουργεί το γράφημα και δηλώνει ποιες στήλες καλούνται από τη βάση. Σε αυτή τη περίπτωση δηλώνονται η ελάχιστη, μέγιστη και μέση τιμή της μεταβλητής ξεχωριστά ώστε να υπολογιστεί με sql κώδικα .

```
function drawChart() {  
  
    var data = new google.visualization.DataTable();  
    data.addColumn('datetime', 'event');  
    data.addColumn('number', 'Max temperature');  
    data.addColumn('number', 'Min temperature');  
    data.addColumn('number', 'Average humidity');  
    data.addColumn('number', 'Max humidity');  
    data.addColumn('number', 'Min humidity');
```

Κώδικας 7-23 Κώδικας monthly statistics, drawChart()

Έπεται το κομμάτι του κώδικα στο οποίο ορίζονται τα στοιχεία της βάσης και γίνεται η σύνδεση.

```
<?php  
    // Prepare variables for database connection  
  
    $dbusername = "root"; // enter database username, I used "arduino" in step 2.2  
    $dbpassword = "password"; // enter database password, I used "arduinotest" in step 2.2  
    $server = "localhost";  
  
    // Connect to your database  
    $mysqli = new mysqli($server, $dbusername, $dbpassword, "myhome");  
    if($mysqli->connect_error)  
        die('Connect Error (' . mysqli_connect_errno() . ') ' . mysqli_connect_error());
```

Κώδικας 7-24 Κώδικας monthly statistics, ορισμός στοιχείων βάσης και σύνδεση

Αφού γίνει το query στη βάση «σπάει» την μεταβλητή datetime σε έτος, μήνα, μέρα, ώρα και δευτερόλεπτα για να δηλωθεί κατάλληλα στο script παρακάτω. Με την εντολή DESC LIMIT 12 ορίζεται πόσες τιμές θα υπάρχουν στο διάγραμμα ,στην συγκεκριμένη περίπτωση 12 δηλαδή των 12 τελευταίων μηνών.

```
    // Execute SQL statement ta grouparw ana mina kai krataw teleytaia 12 +min max +average  
    $result = $mysqli->query("SELECT YEAR(event) AS year, MONTH(event) AS month,  
    DAY(event) AS day, AVG(temperature),MAX(temperature),MIN(temperature),  
    AVG(humidity),MAX(humidity),MIN(humidity), from myhome.greenhouse group by  
    MONTH(event) ORDER BY event DESC limit 12");  
  
    if ($result->num_rows > 0) {  
  
        // output data of each row  
        while($row = $result->fetch_assoc()) {  
            echo "data.addRow([new Date('.$row[\"year\"].\", \".$row[\"month\"].\"-1,\".$row[\"day\"].\"),
```

```

".$row["AVG(temperature)"].",".$row["MAX(temperature)"].",".$row["MIN(temperature)"].",".
$row["AVG(humidity)"].",".$row["MAX(humidity)"].",".$row["MIN(humidity)"].");\n";
    }
}
else {
}
?>

```

Κώδικας 7-25 Κώδικας monthly statistics, query στη βάση

Η κάθε μεταβλητή περιέχει τις επιλογές απεικόνισής της. Όπως ο τίτλος, η μονάδα μέτρησης και ο τρόπος που θα παρουσιάζεται.

```

var temperature_options = {
    title: 'Temperature (C) Last Month ',
    subtitle: 'measured in C',
    hAxis: { format: ' ' },
    seriesType: 'bars',
    width: 600,
    height: 300
};

var humidity_options = {
    title: 'Humidity (%) Last Month ',
    subtitle: 'measured in %',
    hAxis: { format: ' ' },
    seriesType: 'bars',
    width: 600,
    height: 300
};

```

Κώδικας 7-26 Κώδικας monthly statistics, μεταβλητές και επιλογές απεικόνισής τους

Δηλώνεται σε ποιες στήλες θα αντιστοιχεί η κάθε μεταβλητή. Σε αυτή την περίπτωση δηλώνονται σε κάθε μεταβλητή τέσσερις στήλες που περιέχουν την ημερομηνία, την ελάχιστη, μέγιστη και μέση τιμή.

```

//-----
var temperature_chart = new
google.visualization.ComboChart(document.getElementById('temperature_div'));

var temperature_view = new google.visualization.DataView(data);
temperature_view.setColumns([0,10,11,12]);
temperature_chart.draw(temperature_view, temperature_options);
//-----
var humidity_chart = new
google.visualization.ComboChart(document.getElementById('humidity_div'));

var humidity_view = new google.visualization.DataView(data);

```

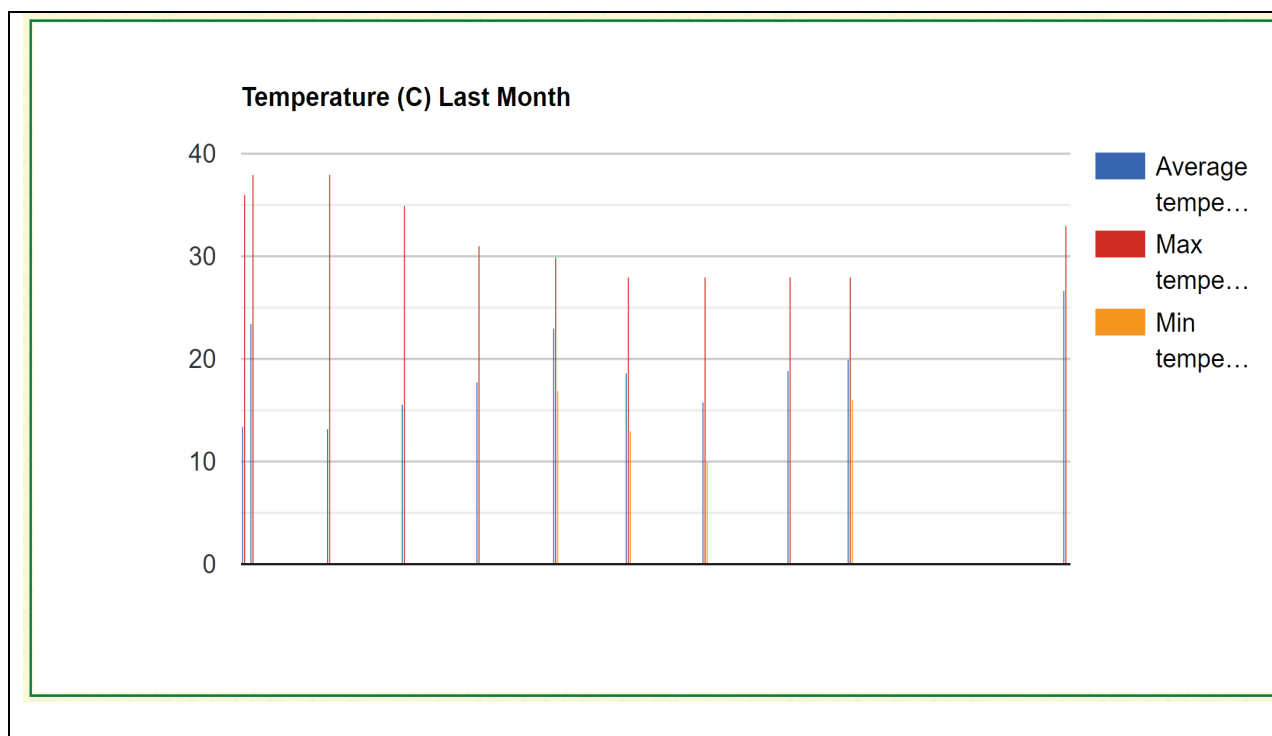
```
humidity_view.setColumns([0,13,14,15]);
humidity_chart.draw(humidity_view, humidity_options);
//-----
```

Κώδικας 7-27 Κώδικας monthly statistics, αντιστοίχιση στηλών και μεταβλητής

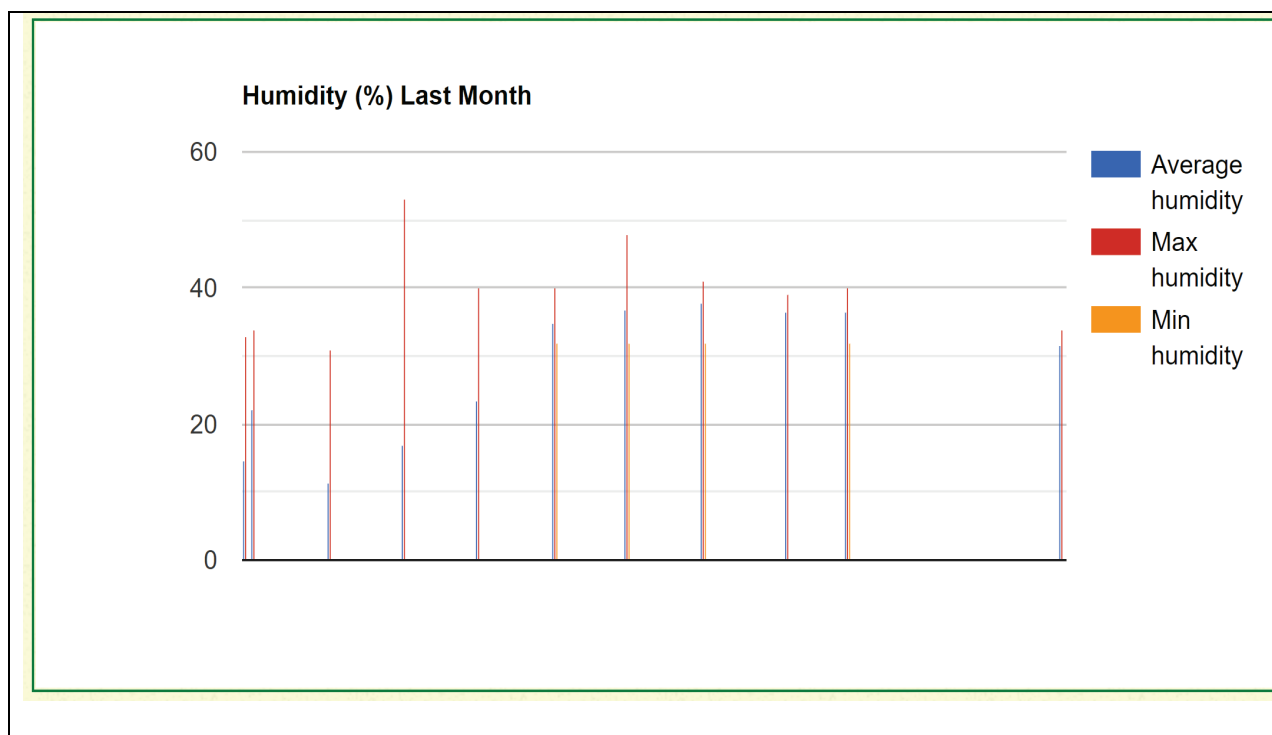
Μετά το τέλος του javascript υπάρχει το υπόλοιπο κομμάτι κώδικα html που εκτός από το να δίνει ένα τίτλο στη σελίδα παρουσιάζει και τα γραφικά μέσα σε έναν πίνακα για καλύτερη στοίχιση.

```
</script>
</head>
<title>Monthly Statistics</title>
<body style="background: # 001433">
  <div style="width:100%">
    <div style="width:90%;font-size:30pt;text-align:center;color:# cca300;text-align:left;"><strong> Monthly Statistics </div>
    </div>
    
    
    <div style="width:100%;font-size:19pt;color:#ffffff;text-align:left;">Μηνιαίες Μετρήσεις
    </div>
    <div align="right">
      <br>
    </div>
    <table style="border-spacing: 15px">
      <tr>
        <td style="border: 1px solid white">
          <div id="Speed_div" style="width: 600px; height: 300px"></div>
        </td>
        <td style="border: 1px solid white">
          <div id="Altitude_div" style="width: 600px; height: 300px"></div>
        </td>
      </tr>
    </table>
  </body>
</html>
```

Κώδικας 7-28 Κώδικας weekly statistics, υπόλοιπος html κώδικας



Εικόνα 7-6 Απεικόνιση της ελάχιστης, μέγιστης και μέσης τιμής της θερμοκρασίας

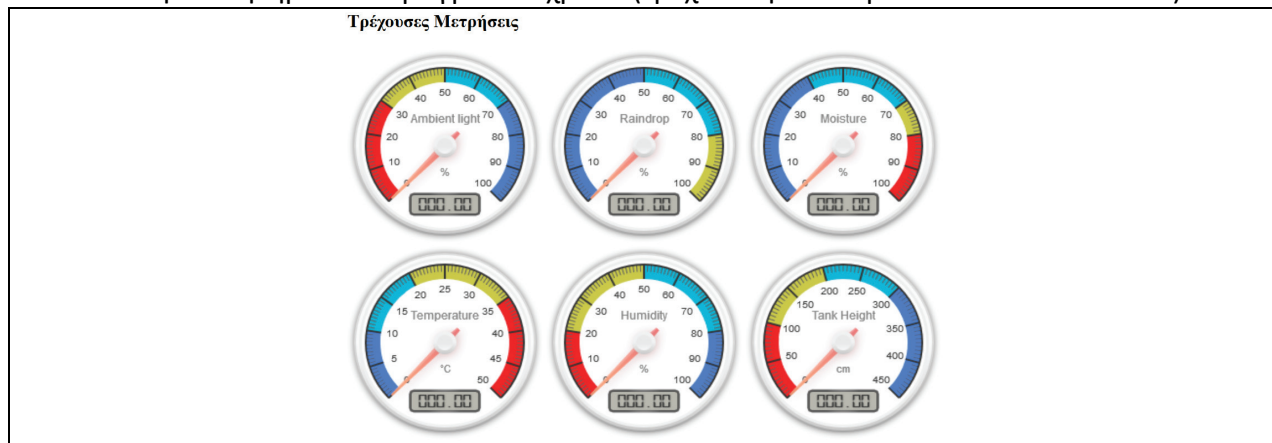


Εικόνα 7-7 Απεικόνιση της ελάχιστης, μέγιστης και μέσης τιμής της υγρασίας

7.3 ΕΝΔΕΙΚΤΙΚΕΣ ΙΣΤΟΣΕΛΙΔΕΣ - ΓΡΑΦΗΜΑΤΑ

7.3.1 ΚΑΤΑΣΤΑΣΗ, ΑΥΤΟΜΑΤΑ ΣΕΝΑΡΙΑ (SD ΚΑΡΤΑ)

Κατάσταση αισθητήρων σε πραγματικό χρόνο (τρέχει στην sd κάρτα του ethernet shield)



Εικόνα 7-8 Τρέχουσες τιμές

Αυτόματα σενάρια

- αυτόματος αερισμός

όταν ενεργοποιηθεί αυτό το σενάριο, κάθε φορά που η θερμοκρασία ξεπεράσει μία συγκεκριμένη τιμή, τότε θα ενεργοποιηθεί ο ανεμιστήρας έως ότου η θερμοκρασία επιστρέψει ξανά σε φυσιολογικά επίπεδα.

- αυτόματο πότισμα

όταν ο αισθητήρας υγρασίας εδάφους αντιληφθεί ότι το χώμα είναι αρκετά ξερό, ο μικροελεγκτής θα θέσει σε λειτουργία την ηλεκτροβάνα, μέχρι το χώμα να πάρει την απαιτούμενη υγρασία.

Αυτοματοποιημένα Σενάρια

☐ αυτόματος αερισμός

☐ αυτόματο πότισμα

Εικόνα 7-9 Αυτόματα σενάρια

Πίνακας ελέγχου

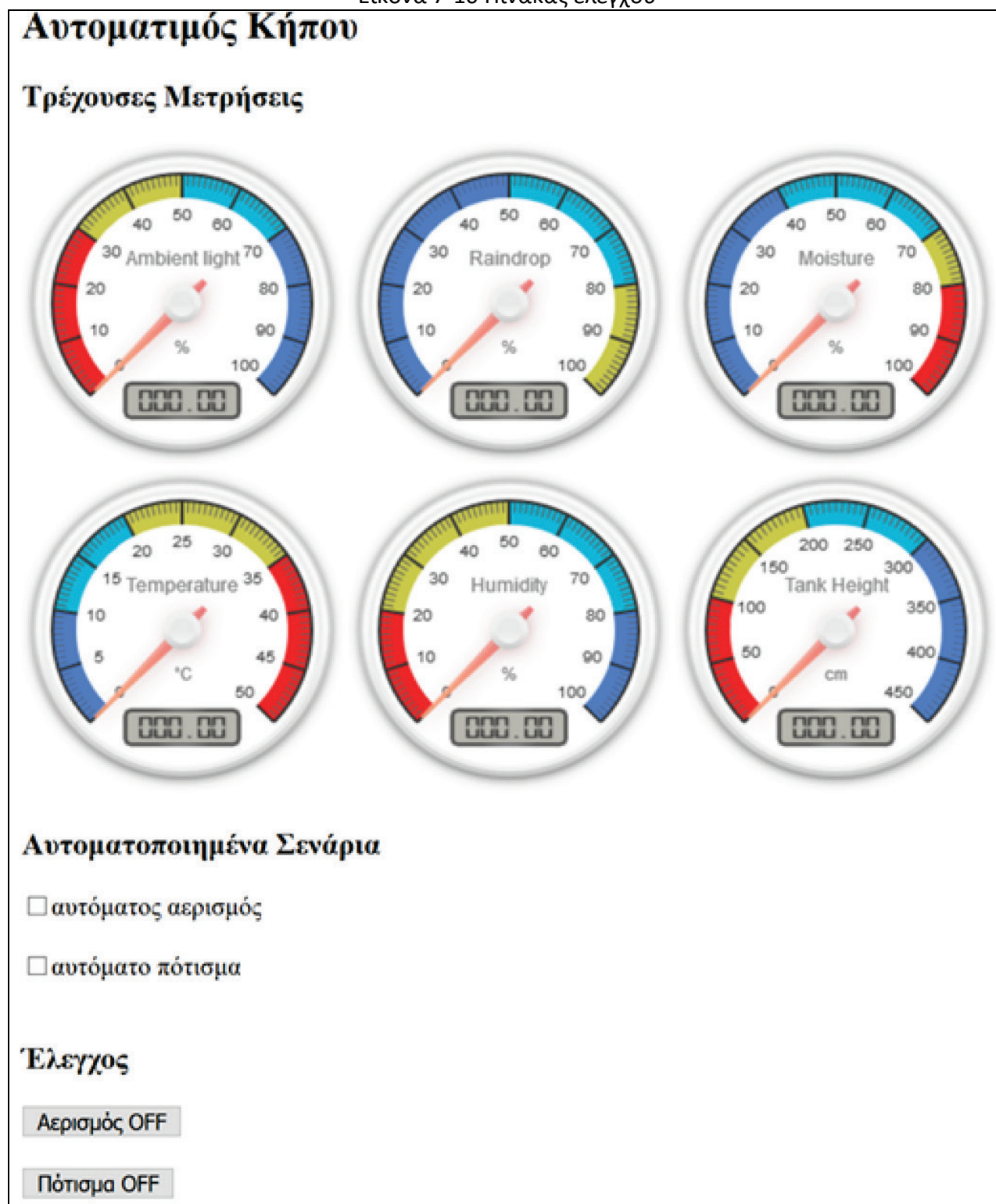
Το τελευταίο μέρος της σελίδας εποπτείας και ελέγχου αφορά τον πίνακα ελέγχου. Ο πίνακας ελέγχου αποτελείται από δύο κουμπιά τα οποία ενεργοποιούν και απενεργοποιούν τις συσκευές δράσης, δηλαδή τον ανεμιστήρα και την ηλεκτροβάνα. Πατώντας το κάθε κουμπί αλλάζει η κατάσταση της συσκευής. Εφ' όσον γίνει αυτό, αλλάζει και η ένδειξη του κουμπιού και από ON γίνεται OFF και αντίστροφα.

Έλεγχος

Αερισμός OFF

Πότισμα OFF

Εικόνα 7-10 Πίνακας ελέγχου



Εικόνα 7-11 Ολοκληρωμένη Ιστοσελίδα

Κώδικας Ιστοσελίδας

```
<!DOCTYPE html>
<html>
  <head>
    <title>Αυτοματισμός </title>
    <script>
```



```

autoFAN = "";
autoSOLENOID = "";
FAN = "";
SOLENOID = "";
var FAN_state = 0;
var SOLENOID_state = 0;
var light_ambient = 0;
var raindrop = 0;
var moisture = 0;
var temperature = 0;
var humidity = 0;
var tank_height = 0;

```

```

eval(function(p,a,c,k,e,r){e=function(c){return(c<a?'':e(parseInt(c/a)))+(c=c%a)>35?String.fromCharCode(c+29):c.toString(36)};if(!''.replace(/^/,String)){while(c-->0){r[e(c)]=k[c]||e(c);k=[function(e){return r[e]}];e=function(){return '\\w+'};c=1};while(c-->0){if(k[c])p=p.replace(new RegExp('\\b'+e(c)+'\\b','g'),k[c]);return p}}t=W=v(f){W:2t.3T(A);A.B={Z:2u,19:1q,1h:1q,1J:U,1b:D,K:0,V:['\\0\\','\\20\\','\\40\\','\\2A\\','\\2B\\','\\D\\'],2r:10,2C:M,1E:U,2q:{2D:3,2E:2},2H:M,1c:{2j:10,2m:3y,2O:'\\3v\\'},J:{2P:'\\#4h\\',V:'\\#31\\',2r:'\\#3k\\',1J:'\\#37\\',1E:'\\#37\\',3e:'\\#31\\',1t:{2s:'\\1e(3Y, 3d, 3d, 1)\\',3c:'\\1e(1Y, 5l, 5E, .9)\\'}},1o:[{1n:20,1D:2A,1F:'\\#3j\\'},{1n:2A,1D:2B,1F:'\\#36\\'},{1n:2B,1D:D,1F:'\\#5n\\'}]};t.g=0,1p=A,N=0,1S=0,1G=U;A.5d=v(a){N=f.1c?g:a;t b=(f.1b-f.K)/D;1S=a>f.1b?1S=f.1b+b:a<f.K?f.K-b:a;g=a;f.1c?3l():A.1g();C A};A.3m=v(a){N=g;a;A.1g();C A};A.4T=v(){g=N=1S=A.B.K;A.1g();C A};A.4R=v(){C g};A.13=v(){v 2k(a,b){Q(t i 4P b){z(1H b[i]==="1W"&&!(4O.4y.2V.4p(b[i])==\\"[1W 4n]\\")&&i!=""Z\\")}{z(1H a[i]!="1W"){a[i]=}}2k(a[i],b[i]))O{a[i]=b[i]}};2k(A.B,f);A.B.K=1R(A.B.K);A.B.1b=1R(A.B.1b);f=A.B;N=g=f.K;z(!f.Z){4m 4j("4g 4d 4b 46 44 41 3Z 3W W 1W!");};t j=f.Z.5K?f.Z:2R.5v(f.Z),q=j.3u('\\2d\\'),1i,1y,1A,14,17,u,1d;v 2M(){j.19=f.19;j.1h=f.1h;1i=j.4s(M);1d=1i.3u('\\2d\\');1y=j.19;1A=j.1h;14=1y/2;17=1A/2;u=14<17?14:17;1i.2J=U;1d.3P(14,17);1d.G();q.3P(14,17);q.G();2M();A.4Z=v(a){2k(A.B,a);2M();A.1g();C A};t k={4q:v(p){C p},4e:v(p){C E.1L(p,2)},4c:v(p){C E.1L(p,5)},3v:v(p){C 1-E.1O(E.5C(p))},5k:v(p){C 1-(v(p){Q(t a=0,b=1;1;a+=b,b/=2){z(p>=(7-4*a)/11){C-E.1L((11-6*a-11*p)/4,2)+E.1L(b,2)}}}(1-p)},4S:v(p){C 1-(v(p){t x=1.5;C E.1L(2,10*(p-1))*E.1T(20*E.1a*x/3*p)})(1-p)};t l=2u;v 3S(d){t e=2v 3R;l=2x(v){t a=2v 3R-e,1M=a/d.2m;z(1M>1){1M=1;t b=1H d.2g=="v"?d.2g:k[d.2g];t c=b(1M);d.3Q(c);z(1M==1){2b(l)},d.2j||10});v 3l(){l&&2b(l);t b=(1S-N),1n=N,29=f.1c;3S({2j:29.2j,2m:29.2m,2g:29.2O,3Q:v(a){N=1R(1n)+b*a;1p.1g()}});q.5l="3O";A.1g=v(){z(!1i.2J){1d.3M(-14,-17,1y,1A);1d.G();t a=q;q=1d;3L();3K();3l();3H();3F();3D();3z();1i.2J=M;q=a;5G a}q.3M(-14,-17,1y,1A);q.G();q.4a(1i,-14,-17,1y,1A);z(!W.28){t b=2x(v){z(!W.28){C 2b(b);2K();2L();z(!1G){1p.13&&1p.13();1G=M}},10)}O{2K();2L();z(!1G){1p.13&&1p.13();1G=M}}C A};v S(a){C a*E.1a/4J};v l(a,b,c){t d=q.4Y(0,0,0,c);d.1V(0,a);d.1V(1,b);C d};v 3L(){t a=u/D*5g,3x=u-a,2a=u/D*5q,5u=u-2a,1f=u/D*5z,5A=u-1f;3t=u/D*5F;q.G();z(f.2H){q.2o=3x;q.2n='\\1e(0, 0, 0, 0.5)\\'}q.P();q.16(0,0,a,0,E.1a*2,M);q.L=1l('\\#42\\','\\#43\\',a);q.T();q.R();q.P();q.16(0,0,2a,0,E.1a*2,M);q.L=1l('\\#49\\','\\#36\\',2a);q.T();q.P();q.16(0,0,1f,0,E.1a*2,M);q.L=1l('\\#3j\\','\\#3s\\',1f);q.T();q.P();q.16(0,0,3t,0,E.1a*2,M);q.L=f.J.2P;q.T();q.G();v 3H(){t r=u/D*2T;q.2e=2;q.2U=f.J.V;q.G();Q(t i=0;i<f.V.H;++i){t a=45+i*(1U/(f.V.H-1));q.1z(S(a));q.P();q.1K(0,r);q.F(0,r-u/D*15);q.1X();q.R();q.G()}{z(f.2C){q.1z(S(2X));q.P();q.16(0,0,r,S(45),S(4N),U);q.1X();q.R();q.G()}};

```



```

v 3l(){t r=u/D*2T;q.2e=1;q.2U=f.J.2r;q.G();t b=f.2r*(f.V.H-1);Q(t i=0;i<b;++i){t
a=45+i*(1U/b);q.1z(S(a));q.P();q.1K(0,r);q.F(0,r-u/D*7.5);q.1X();q.R();q.G());v 3F(){t
r=u/D*55;Q(t i=0;i<f.V.H;++i){t a=45+i*(1U/(f.V.H-1)),p=1w(r,S(a));q.1x=20*(u/1q)+"2i
2Y";q.L=f.J.3e;q.2e=0;q.2h="2f";q.27(f.V[i],p.x,p.y+3)}};v
3D(){z(!f.1J){C}q.G();q.1x=24*(u/1q)+"2i 2Y";q.L=f.J.1J;q.2h="2f";q.27(f.1J,0,-u/4.25);q.R();v
3z(){z(!f.1E){C}q.G();q.1x=22*(u/1q)+"2i 2Y";q.L=f.J.1E;q.2h="2f";q.27(f.1E,0,u/3.25);q.R();v
32(a){t b=f.2q.2E,34=f.2q.2D;a=1R(a);t n=(a<0);a=E.35(a);z(b>0){a=a.5t(b).2V().1j('\');Q(t
i=0,s=34-a[0].H;i<s;++i){a[0]='\0'+a[0]}a=(n?'-\':'\')+a[0]+\'.\'+a[1]}O{a=E.3O(a).2V();Q(t
i=0,s=34-a.H;i<s;++i){a='\0'+a}a=(n?'-\':'\')+a}C a};v 1w(r,a){t
x=0,y=r,1O=E.1O(a),1T=E.1T(a),X=x*1T-y*1O,Y=x*1O+y*1T;C{x:X,y:Y}};v 3K(){q.G();t a=u/D*2T;t
b=a-u/D*15;Q(t i=0,s=f.1o.H;i<s;i++){t c=f.1o[i],39=(f.1b-f.K)/1U,1P=S(45+(c.1n-
f.K)/39),1N=S(45+(c.1D-f.K)/39);q.P();q.1z(S(2X));q.16(0,0,a,1P,1N,U);q.R();q.G());t
d=1w(b,1P),3a=1w(a,1P);q.1K(d.x,d.y);q.F(3a.x,3a.y);t
e=1w(a,1N),3b=1w(b,1N);q.F(e.x,e.y);q.F(3b.x,3b.y);q.F(d.x,d.y);q.1C();q.L=c.1F;q.T();q.P());q.1z(S
(2X));q.16(0,0,b,1P-0.2,1N+0.2,U);q.R();q.1C();q.L=f.J.2P;q.T();q.G());v 2L(){t
a=u/D*12,1f=u/D*8,1u=u/D*3X,1r=u/D*20,2l=u/D*4,1B=u/D*2,38=v(){q.3f=2;q.3g=2;q.2o=10;
q.2n='\1e(5L, 3h, 3h, 0.45)\';38();q.G();z(N<0){N=E.35(f.K-N)}O z(f.K>0){N-
=f.K}O{N=E.35(f.K)+N}q.1z(S(45+N/((f.1b-f.K)/1U)));q.P();q.1K(-1B,-1r);q.F(-2l,0);q.F(-
1,1u);q.F(1,1u);q.F(2l,0);q.F(1B,-1r);q.1C();q.L=1l(f.J.1t.2s,f.J.1t.3c,1u-1r);q.T());q.P();q.F(-
0.5,1u);q.F(-1,1u);q.F(-2l,0);q.F(-1B,-1r);q.F(1B/2-2,-1r);q.1C();q.L='\1e(1Y, 1Y, 1Y,
0.2)\';q.T());q.R();38();q.P();q.16(0,0,a,0,E.1a*2,M);q.L=1l('\#3s\','\#36\';a);q.T());q.R());q.16(
0,0,1f,0,E.1a*2,M);q.L=1l("\#47", "\#48",1f);q.T());v 3i(x,y,w,h,r){q.P();q.1K(x+r,y);q.F(x+w-
r,y);q.23(x+w,y,x+w,y+r);q.F(x+w,y+h-r);q.23(x+w,y+h,x+w-r,y+h);q.F(x+r,y+h);q.23(x,y+h,x,y+h-
r);q.F(x,y+r);q.23(x,y,x+r,y);q.1C());v 2K(){q.G();q.1x=40*(u/1q)+"2i 30";t a=32(g),2Z=q.4f('\-
\'+32(0)).19,y=u-u/D*33,x=0,2W=0.12*u;q.G();3i(-2Z/2-0.21*u,y-2W-
0.4i*u,2Z+0.3n*u,2W+0.4k*u,0.21*u);t b=q.4l(x,y-0.12*u-0.21*u+(0.12*u+0.3o*u)/2,u/10,x,y-
0.12*u-
0.21*u+(0.12*u+0.3o*u)/2,u/5);b.1V(0,"#37");b.1V(1,"#3k");q.2U=b;q.2e=0.3n*u;q.1X();q.2o=0
.3p*u;q.2n='\1e(0, 0, 0,
1)\';q.L="#40";q.T());q.R());q.3f=0.3q*u;q.3g=0.3q*u;q.2o=0.3p*u;q.2n='\1e(0, 0, 0,
0.3)\';q.L="#31";q.2h="2f";q.27(a,-x,y);q.R());W.28=U;(v){t
d=2R,h=d.3r('\4r')[0],2S=4t.4u.4v().4w('\4x')!=-1,2Q='\4z://4A-4B.4C/4D/4E/4F-7-
4G.\'+(2S?'\4H':'\4l'),1l="@1x-4K {"+"1x-4L: '\30\';"+"4M:
2Q('\'+2Q+"'\');"+"})",1s,r=d.3w('\1v\');r.2N='\1l/4Q\';z(2S){h.2p(r);1s=r.2l;1s.3A=1l}O{4U{r.2p(d.
4V(1l))}4W(e){r.3A=1l}h.2p(r);1s=r.2l?r.2l:(r.4X| |d.3B[d.3B.H-1])}t b=2x(v){z(!d.3C){C}2b(b);t
a=d.3w('\50\');a.1v.51='\30\';a.1v.52='\53\';a.1v.1h=a.1v.19=0;a.1v.54='\56\';a.57='\.';d.3C.2p
(a);58(v){W.28=M;a.59.5a(a),3y},1)}();W.2t=[];W.2t.5b=v(a){z(1H(a)=='5c\'){Q(t
i=0,s=A.H;i<s;i++){z(A[i].B.Z.18('\5e\')==a){C A[i]}}O z(1H(a)=='5f\'){C A[a]}O{C 2u}};v
3E(a){z(2G.3G){2G.3G('\5h',a,U)}O{2G.5i('\5j',a)}}3E(v){v 2F(a){t b=a[0];Q(t
i=1,s=a.H;i<s;i++){b+=a[i].1Z(0,1).5m()+a[i].1Z(1,a[i].H-1)}C b};v 3J(a){C
a.5o(/^\s+|\\s+$/,\');t c=2R.3r('\5p\');Q(t i=0,s=c.H;i<s;i++){z(c[i].18('\1k-2N\')=='\5r-5s\'){t
d=c[i],B={},1m,w=2c(d.18('\19\')),h=2c(d.18('\1h\'));B.Z=d;z(w){B.19=w;z(h){B.1h=h}Q(t
e=0,1s=d.3N.H;e<1s;e++){1m=d.3N.5w(e).5x;z(1m!='1k-2N'\&&1m.1Z(0,5)=='1k-\'){t
f=1m.1Z(5,1m.H-5).5y().1j('\-
\'),l=d.18(1m);z(!){2z}5B(f[0]){2y\J\':{z(f[1]){z(!B.J){B.J={}}z(f[1]=='1t\'){t
k=1.j(/^\s+/);z(k[0]&&k[1]){B.J.1t={2s:k[0],3c:k[1]}}O{B.J.1t=l}}O{f.5D();B.J[2F(f)]=l}}26}2y\1o\':{
z(!B.1o){B.1o=[]}2w=l.1j('\');Q(t j=0,l=2w.H;j<l;j++){t
m=3J(2w[j]).1j(/^\s+/),1Q={};z(m[0]&&m[0]!='\'){1Q.1n=m[0]}z(m[1]&&m[1]!='\'){1Q.1D=m[1]
}z(m[2]&&m[2]!='\'){1Q.1F=m[2]}B.1o.3T(1Q)}26}2y\1c\':{z(f[1]){z(!B.1c){B.1c={}}z(f[1]=='2O\

```

[illegible]

```

{
    if (this.readyState == 4) {
        if (this.status == 200) {
            if (this.responseXML != null) {
                light_ambient =
this.responseXML.getElementsByTagName('analog')[0].childNodes[0].nodeValue;
                raindrop =
this.responseXML.getElementsByTagName('analog')[1].childNodes[0].nodeValue;
                moisture =
this.responseXML.getElementsByTagName('analog')[2].childNodes[0].nodeValue;
                temperature =
this.responseXML.getElementsByTagName('analog')[3].childNodes[0].nodeValue;
                humidity =
this.responseXML.getElementsByTagName('analog')[4].childNodes[0].nodeValue;
                tank_height =
this.responseXML.getElementsByTagName('analog')[5].childNodes[0].nodeValue;
                // LED 1
                if
(this.responseXML.getElementsByTagName('LED')[0].childNodes[0].nodeValue === "checked")
{
                    document.LED_form.LED1.checked = true;
                }
                else {
                    document.LED_form.LED1.checked = false;
                }
                // LED 2
                if
(this.responseXML.getElementsByTagName('LED')[1].childNodes[0].nodeValue === "checked")
{
                    document.LED_form.LED2.checked = true;
                }
                else {
                    document.LED_form.LED2.checked = false;
                }
                // LED 3
                if
(this.responseXML.getElementsByTagName('LED')[2].childNodes[0].nodeValue === "on") {
                    document.getElementById("LED3").innerHTML = "Airing is ON";
                    FAN_state = 1;
                }
                else {
                    document.getElementById("LED3").innerHTML = "Airing is OFF";
                    FAN_state = 0;
                }
                // LED 4
                if
(this.responseXML.getElementsByTagName('LED')[3].childNodes[0].nodeValue === "on") {
                    document.getElementById("LED4").innerHTML = "Watering is ON";
                    SOLENOID_state = 1;
                }
            }
        }
    }
}

```

```

    }
    else {
        document.getElementById("LED4").innerHTML = " Watering is OFF";
        SOLENOID_state = 0;
    }
}
}
}
}
}
}
}
// send HTTP GET request with LEDs to switch on/off if any
request.open("GET", "ajax_inputs" + autoFAN + autoSOLENOID + FAN + SOLENOID +
nocache, true);
request.send(null);
setTimeout('GetArduinoIO()', 1000);
autoFAN = "";
autoSOLENOID = "";
FAN = "";
SOLENOID = "";
}
// service LEDs when checkbox checked/unchecked
function GetCheck()
{
    if (LED_form.LED1.checked) {
        autoFAN = "&LED1=1";
    }
    else {
        autoFAN = "&LED1=0";
    }
    if (LED_form.LED2.checked) {
        autoSOLENOID = "&LED2=1";
    }
    else {
        autoSOLENOID = "&LED2=0";
    }
}
function GetButton1()
{
    if (FAN_state === 1) {
        FAN_state = 0;
        FAN = "&LED3=0";
    }
    else {
        FAN_state = 1;
        FAN = "&LED3=1";
    }
}
function GetButton2()
{
    if (SOLENOID_state === 1) {

```

```

        SOLENOID_state = 0;
        SOLENOID = "&LED4=0";
    }
    else {
        SOLENOID_state = 1;
        SOLENOID = "&LED4=1";
    }
}
</script>

</head>
<body onload="GetArduinoIO()">
<div1a>
<h2>Αυτοματιμός Κήπου/Θερμοκηπίου</h2>
</div1a>
<div1b></div1b>
<div2a><h3>Τρέχουσες Μετρήσεις</h3></div2a>
<div2b>
<canvas id="an_gauge_1" width="200" height="200" data-title="Ambient light"
data-units="%" data-highlights="0 30 #f33, 30 50 #cc5, 50 70 #25B8D9, 70 100 #4D89F2"
data-major-ticks="0 10 20 30 40 50 60 70 80 90 100" data-type="canv-gauge"
data-min-value="0" data-max-value="100"
data-onready="setInterval( function() {
Gauge.Collection.get('an_gauge_1').setValue(light_ambient);}, 200);"></canvas>
<canvas id="an_gauge_2" width="200" height="200" data-title="Raindrop" data-
units="%" data-highlights="0 50 #4D89F2, 50 80 #25B8D9, 80 100 #cc5" data-major-ticks="0
10 20 30 40 50 60 70 80 90 100" data-type="canv-gauge" data-min-value="0" data-max-
value="100" data-onready="setInterval( function() {
Gauge.Collection.get('an_gauge_2').setValue(raindrop);}, 200);"></canvas>
<canvas id="an_gauge_3" width="200" height="200" data-title="Moisture" data-
units="%" data-highlights="0 40 #4D89F2, 40 70 #25B8D9, 70 80 #cc5, 80 100 #f33" data-
major-ticks="0 10 20 30 40 50 60 70 80 90 100" data-type="canv-gauge" data-min-value="0"
data-max-value="100" data-onready="setInterval( function() {
Gauge.Collection.get('an_gauge_3').setValue(moisture);}, 200);"></canvas>
<canvas id="an_gauge_4" width="200" height="200" data-title="Temperature" data-
units="°C" data-highlights="0 10 #4D89F2, 10 20 #25B8D9, 20 35 #cc5, 35 50 #f33" data-
major-ticks="0 5 10 15 20 25 30 35 40 45 50" data-type="canv-gauge" data-min-value="0" da-
ta-max-value="50" data-onready="setInterval( function() {
Gauge.Collection.get('an_gauge_4').setValue(temperature);}, 200);"></canvas>
<canvas id="an_gauge_5" width="200" height="200" data-title="Humidity" data-
units="%" data-highlights="0 20 #f33, 20 50 #cc5, 50 80 #25B8D9, 80 100 #4D89F2" data-
major-ticks="0 10 20 30 40 50 60 70 80 90 100" data-type="canv-gauge" data-min-value="0"
data-max-value="100" data-onready="setInterval( function() {
Gauge.Collection.get('an_gauge_5').setValue(humidity);}, 200);"></canvas>
<canvas id="an_gauge_6" width="200" height="200" data-title="Tank Height" data-
units="cm" data-highlights="0 100 #f33, 100 200 #cc5, 200 300 #25B8D9, 300 450 #4D89F2"
data-major-ticks="0 50 100 150 200 250 300 350 400 450" data-type="canv-gauge" data-min-
value="0" data-max-value="450" data-onready="setInterval( function() {

```

```
Gauge.Collection.get('an_gauge_6').setValue(tank_height);}, 200);"></canvas>
</div2b>
<div3a>
<h3>Αυτοματοποιημένα Σενάρια</h3></div3a>
<div3b><form id="check_LEDs" name="LED_form">
  <input type="checkbox" name="LED1" value="0" onclick="GetCheck()" />αυτόματος
αερισμός θερμοκηπίου<br /><br />
  <input type="checkbox" name="LED2" value="0" onclick="GetCheck()" />αυτόματο
πότισμα κήπου/θερμοκηπίου<br /><br />
</form>
</div3b>
<div4a><h3>Έλεγχος</h3></div4a>
<div4b>
  <button type="button" id="LED3" onclick="GetButton1()">Αερισμός OFF</button><br
/><br />
  <button type="button" id="LED4" onclick="GetButton2()">Πότισμα OFF</button><br /><br
/>
</div4b>
</body>
</html>
```

Κώδικας 7-29 Κώδικας Ιστοσελίδας

7.3.2 ΚΑΤΑΣΤΑΣΗ (WEB SERVER ΣΕ RASPBERRY PI)

Κατάσταση αισθητήρων σε πραγματικό χρόνο (τρέχει σε web server)



Εικόνα 7-12 Τρέχουσες τιμές (web server)

Κώδικας Ιστοσελίδας

```
<!DOCTYPE HTML>
<html>
<head>
  <meta charset="UTF-8">
  <title>Gauge</title>
  <link rel="stylesheet" href="../css/style.css" type="text/css">
  <?php
    // Prepare variables for database connection
    $dbusername = "root"; // enter database username, I used "arduino" in step 2.2
    $dbpassword = "password"; // enter database password, I used "arduinotest" in step 2.2
    $server = "localhost";

    // Connect to your database
    $mysqli = new mysqli($server, $dbusername, $dbpassword, "myhome");
    if($mysqli->connect_error)
      die('Connect Error (' . mysqli_connect_errno() . ') ' . mysqli_connect_error());

    // Execute SQL statement

    $result = $mysqli->query("SELECT light,raindrop,moisture,temperature,humidity,tank,fan,van
FROM myhome.greenhouse ORDER BY id DESC LIMIT 1"); // tin teleytaia metrisi pernw apo vasi
me to limit 1

    if ($result->num_rows > 0) {

      // output data of each row
```



```

while($row = $result->fetch_assoc()) {
    $LValue = $row["light"];
    $RValue = $row["raindrop"];
    $MValue = $row["moisture"];
    $TValue = $row["temperature"];
    $HValue = $row["humidity"];
    $TAValue = $row["tank"];
    $FValue = $row["fan"];
    $VValue = $row["van"];
}
}
else {
    $LValue = 0;
    $RValue = 0;
    $MValue = 0;
    $TValue = 0;
    $HValue = 0;
    $TAValue = 0;
    $FValue = 0;
    $VValue = 0;
}
?>

<script src="gauge.js"></script>
<meta http-equiv="refresh" content="180" />

<script>
if (!Array.prototype.forEach) {
    Array.prototype.forEach = function(cb) {
        var i = 0, s = this.length;
        for (; i < s; i++) {
            cb && cb(this[i], i, this);
        }
    }
}

document.fonts && document.fonts.forEach(function(font) {
    font.loaded.then(function() {
        if (font.family.match(/Led/)) {
            document.gauges.forEach(function(gauge) {
                gauge.update();
                gauge.options.renderTo.style.visibility = 'visible';
            });
        }
    });
});
});

```

```
var timers = [];
```

```
function animateGauges() {
  document.gauges.forEach(function(gauge) {
    timers.push(setInterval(function() {
      gauge.value = Math.random() *
        (gauge.options.maxValue - gauge.options.minValue) +
        gauge.options.minValue;
    }, gauge.animation.duration + 50));
  });
}
```

```
function stopGaugesAnimation() {
  timers.forEach(function(timer) {
    clearInterval(timer);
  });
}
```

```
function resize() {
  var size = parseFloat(document.getElementById('gauge-size').value) || 400;

  document.gauges.forEach(function (gauge) {
    gauge.update({ width: size, height: size });
  });
}
```

```
function setText() {
  var text = document.getElementById('gauge-text').value;

  document.gauges.forEach(function (gauge) {
    gauge.update({ valueText: text });
  });
}
```

```
</script>
```

```
</head>
```

```
<body>
```

```

  <div class="content">
    
    <br>
    <strong>Γραφική Αποικόνιση με Google Gauge </strong><br>
  <p>
```

```

  <br><br>Στα παρακάτω γραφήματα απεικονίζονται οι τελευταίες τιμές των αισθητήρων με
  μορφή Gauge.
```

Στα δύο τελευταία γραφήματα απεικονίζονται οι τιμές του ανεμιστήρα και της ηλεκτροβάνας που λειτουργούν ως αυτοματισμοί. Την στιγμή που θα ενεργοποιηθεί η συνθήκη του ανεμιστήρα ή της ηλεκτροβάνας η τιμή θα δείξει '1' και θα λειτουργήσουν τα αντίστοιχα στοιχεία. </p>

```
<canvas data-type="radial-gauge"
data-width="250"
data-height="250"
data-units="%"
data-title="Light"
data-value=<?php echo "\"".$LValue."\""\n"?>
data-animate-on-init="false"
data-animated-value="false"
data-min-value="0"
data-max-value="100"
data-major-ticks="0,10,20,30,40,50,60,70,80,90,100"
data-minor-ticks="0.1"
data-stroke-ticks="false"
data-highlights=[
  { "from": 0, "to": 40, "color": "rgba(0,0,0)" },
  { "from": 40, "to": 70, "color": "rgba(204, 153, 0,1)" },
  { "from": 70, "to": 100, "color": "rgba(0, 153, 51,1)" }
]
data-color-plate="#001433"
data-color-major-ticks="#f5f5f5"
data-color-minor-ticks="#ddd"
data-color-title="fff"
data-color-units="ccc"
data-color-numbers="eee"
data-color-needle-start="rgba(240, 128, 128, 1)"
data-color-needle-end="rgba(255, 160, 122, .9)"
data-value-box="true"
data-animation-rule="bounce"
data-animation-duration="500"
data-border-outer-width="3"
data-border-middle-width="3"
data-border-inner-width="3"
></canvas>

<canvas data-type="radial-gauge"
data-width="250"
data-height="250"
data-units="%"
data-title="Raindrop"
data-value=<?php echo "\"".$RValue."\""\n"?>
data-animate-on-init="false"
data-animated-value="false"
data-min-value="0"
```

```

data-max-value="100"
data-major-ticks="0,10,20,30,40,50,60,70,80,90,100"
data-minor-ticks="0.1"
data-stroke-ticks="false"
data-highlights=[
  { "from": 0, "to": 40, "color": "rgba(65, 65, 65)" },
  { "from": 40, "to": 100, "color": "rgba(0, 0, 183)" }
]
data-color-plate="#001433"
data-color-major-ticks="#f5f5f5"
data-color-minor-ticks="#ddd"
data-color-title="#fff"
data-color-units="#ccc"
data-color-numbers="#eee"
data-color-needle-start="rgba(240, 128, 128, 1)"
data-color-needle-end="rgba(255, 160, 122, .9)"
data-value-box="true"
data-animation-rule="bounce"
data-animation-duration="500"
data-border-outer-width="3"
data-border-middle-width="3"
data-border-inner-width="3"
</canvas>

```

```

<canvas data-type="radial-gauge"
  data-width="250"
  data-height="250"
  data-units="%"
  data-title="Moisture"
  data-value=<?php echo "\"".$MValue."\""\n"?>
  data-animate-on-init="false"
  data-animated-value="false"
  data-min-value="0"
  data-max-value="100"
  data-major-ticks="0,10,20,30,40,50,60,70,80,90,100"
  data-minor-ticks="0.1"
  data-stroke-ticks="false"
  data-highlights=[
    { "from": 0, "to": 40, "color": "rgba(65, 65, 65)" },
    { "from": 40, "to": 100, "color": "rgba(0, 0, 183)" }

  ]
  data-color-plate="#001433"
  data-color-major-ticks="#f5f5f5"
  data-color-minor-ticks="#ddd"
  data-color-title="#fff"
  data-color-units="#ccc"
  data-color-numbers="#eee"
  data-color-needle-start="rgba(240, 128, 128, 1)"

```

```

data-color-needle-end="rgba(255, 160, 122, .9)"
data-value-box="true"
data-animation-rule="bounce"
data-animation-duration="500"
data-border-outer-width="3"
data-border-middle-width="3"
data-border-inner-width="3"
<</canvas>
<canvas data-type="radial-gauge"
  data-width="250"
  data-height="250"
  data-units="&deg;C"
  data-title="Temperature"
  data-value=<?php echo "\"".$TValue."\""\n"?>
  data-animate-on-init="false"
  data-animated-value="false"
  data-min-value="-50"
  data-max-value="50"
  data-major-ticks="[-50,-40,-30,-20,-10,0,10,20,30,40,50]"
  data-minor-ticks="2"
  data-stroke-ticks="true"
  data-highlights=[
    {"from": -50, "to": 0, "color": "rgba(0,0, 255, .3)"},
    {"from": 0, "to": 20, "color": "rgba(0, 153, 51,1)"},
    {"from": 35, "to": 50, "color": "rgba(77, 0, 0,1)"},
    {"from": 20, "to": 35, "color": "rgba(255, 51, 51,1)"}
  ]
  data-color-plate="#001433"
  data-color-major-ticks="#f5f5f5"
  data-color-minor-ticks="#ddd"
  data-color-title="#fff"
  data-color-units="#ccc"
  data-color-numbers="#eee"
  data-color-needle-start="rgba(240, 128, 128, 1)"
  data-color-needle-end="rgba(255, 160, 122, .9)"
  data-value-box="true"
  data-animation-rule="bounce"
  data-animation-duration="500"
  data-border-outer-width="3"
  data-border-middle-width="3"
  data-border-inner-width="3"
<</canvas>
<canvas data-type="radial-gauge"
  data-width="250"
  data-height="250"
  data-units="%"
  data-title="Humidity"
  data-value=<?php echo "\"".$HValue."\""\n"?>

```

```

data-animate-on-init="false"
data-animated-value="false"
data-min-value="0"
data-max-value="100"
data-major-ticks="0,10,20,30,40,50,60,70,80,90,100"
data-minor-ticks="0.1"
data-stroke-ticks="false"
data-highlights=[
  { "from": 0, "to": 40, "color": "rgba(204, 153, 0,1)" },
  { "from": 40, "to": 70, "color": "rgba(0, 153, 51,1)" },
  { "from": 70, "to": 100, "color": "rgba(255, 51, 51,1)" }
]
data-color-plate="#001433"
data-color-major-ticks="#f5f5f5"
data-color-minor-ticks="#ddd"
data-color-title="#fff"
data-color-units="#ccc"
data-color-numbers="#eee"
data-color-needle-start="rgba(240, 128, 128, 1)"
data-color-needle-end="rgba(255, 160, 122, .9)"
data-value-box="true"
data-animation-rule="bounce"
data-animation-duration="500"
data-border-outer-width="3"
data-border-middle-width="3"
data-border-inner-width="3"
</canvas>

<canvas data-type="radial-gauge"
  data-width="250"
  data-height="250"
  data-units="CM"
  data-title="Tank"
  data-value="<?php echo "\".$TAValue. "\n"?>"
  data-animate-on-init="false"
  data-animated-value="false"
  data-min-value="0"
  data-max-value="100"
  data-major-ticks="0,10,20,30,40,50,60,70,80,90,100"
  data-minor-ticks="0.1"
  data-stroke-ticks="false"
  data-highlights=[
    { "from": 0, "to": 50, "color": "rgba(0, 153, 51,1)" },
    { "from": 50, "to": 70, "color": "rgba(60, 62, 10)" },
    { "from": 70, "to": 100, "color": "rgba(204, 0, 0,1)" }
  ]
  data-color-plate="#001433"
  data-color-major-ticks="#f5f5f5"
  data-color-minor-ticks="#ddd"

```

```

data-color-title="#fff"
data-color-units="#ccc"
data-color-numbers="#eee"
data-color-needle-start="rgba(240, 128, 128, 1)"
data-color-needle-end="rgba(255, 160, 122, .9)"
data-value-box="true"
data-animation-rule="bounce"
data-animation-duration="500"
data-border-outer-width="3"
data-border-middle-width="3"
data-border-inner-width="3"
<</canvas>
  <canvas data-type="radial-gauge"
data-width="250"
data-height="250"
data-units=""
data-title="Fan"
data-value=<?php echo "\"".$FValue.\""\n"?>
data-animate-on-init="false"
data-animated-value="false"
data-min-value="0"
data-max-value="1"
data-major-ticks="0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1"
data-minor-ticks="0.1"
data-stroke-ticks="false"
data-highlights=[
  { "from": 0, "to": 0.5, "color": "rgba(0,0,0)" },
  { "from": 0.5, "to": 1, "color": "rgba(255,0,0.8)" }

]'
data-color-plate="#001433"
data-color-major-ticks="#f5f5f5"
data-color-minor-ticks="#ddd"
data-color-title="#fff"
data-color-units="#ccc"
data-color-numbers="#eee"
data-color-needle-start="rgba(240, 128, 128, 1)"
data-color-needle-end="rgba(255, 160, 122, .9)"
data-value-box="true"
data-animation-rule="bounce"
data-animation-duration="500"
data-border-outer-width="3"
data-border-middle-width="3"
data-border-inner-width="3"
<</canvas>
  <canvas data-type="radial-gauge"
data-width="250"
data-height="250"
data-units=""

```



```

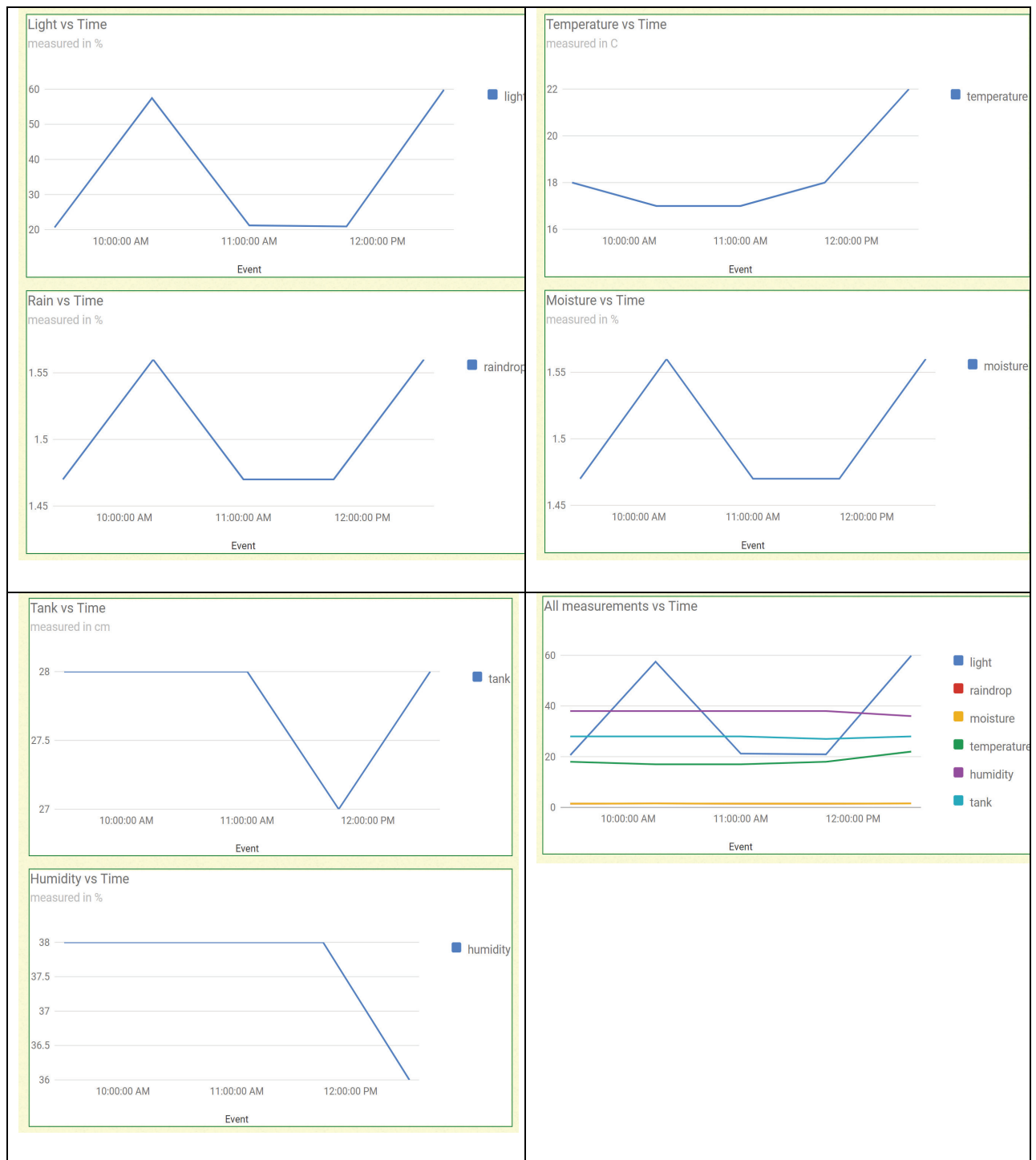
data-title="Van"
data-value=<?php echo "\"".$VValue."\"\\n"?>
data-animate-on-init="false"
data-animated-value="false"
data-min-value="0"
data-max-value="1"
data-major-ticks="0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1"
data-minor-ticks="0.1"
data-stroke-ticks="false"
data-highlights=[
    { "from": 0, "to": 0.5, "color": "rgba(0,0,0)" },
    { "from": 0.5, "to": 1, "color": "rgba(255,0,0.8)" }
]
data-color-plate="#001433"
data-color-major-ticks="#f5f5f5"
data-color-minor-ticks="#ddd"
data-color-title="#fff"
data-color-units="#ccc"
data-color-numbers="#eee"
data-color-needle-start="rgba(240, 128, 128, 1)"
data-color-needle-end="rgba(255, 160, 122, .9)"
data-value-box="true"
data-animation-rule="bounce"
data-animation-duration="500"
data-border-outer-width="3"
data-border-middle-width="3"
data-border-inner-width="3"
></canvas>
<p></p>
<p></p>
<p></p>
</div>
</div>
</body>
</html>

```

Κώδικας 7-30 Κώδικας Ιστοσελίδας

7.3.3 ΕΝΔΕΙΚΤΙΚΑ ΣΤΑΤΙΣΤΙΚΑ (WEB SERVER ΣΕ RASPBERRY PI)

Ενδεικτικά στατιστικά δεδομένων αισθητήρων (τρέχει σε web server)



Εικόνα 7-13 Τρέχουσες τιμές (web server)

Κώδικας Ιστοσελίδας

```
<html>
<head>
  <meta charset="UTF-8">
  <title>Statistics</title>
  <link rel="stylesheet" href="../css/style.css" type="text/css">
  <meta http-equiv="refresh" content="60" /> <!--refresh selidas-->
  <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
  <table align="center">
```

```

<script type="text/javascript">
  google.charts.load('current', {'packages':['line']});
  google.charts.setOnLoadCallback(drawChart);

  function drawChart() {

    var data = new google.visualization.DataTable();// ftiaxnw pinaka gia na valw mesa ta data
    moy ta opoia ta pernw apo ti vasi
    data.addColumn('datetime', 'Event');
    data.addColumn('number', 'light');
    data.addColumn('number', 'raindrop');
    data.addColumn('number', 'moisture');
    data.addColumn('number', 'temperature');
    data.addColumn('number', 'humidity');
    data.addColumn('number', 'tank');

    <?php
    // Prepare variables for database connection
    $dbusername = "root"; // enter database username, I used "arduino" in step 2.2
    $dbpassword = "password"; // enter database password, I used "arduinotest" in step 2.2
    $server = "localhost";

    // Connect to your database
    $mysqli = new mysqli($server, $dbusername, $dbpassword, "myhome");
    if($mysqli->connect_error)
    die('Connect Error (' . mysqli_connect_errno() . ') ' . mysqli_connect_error());

    // Execute SQL statement travaw apo ti vasi to time stamp kai to spaw se year month day
    hour minute kai second gia na valw sto pinaka tin timestamp opws to thelei to javascript+ oles
    tis metriseis (teleytaies 40)
    $result = $mysqli->query("SELECT YEAR(event) AS year, MONTH(event) AS month,
    DAY(event) AS day, HOUR(event) AS hour, MINUTE(event) AS minute, SECOND(event) AS sec,
    light,raindrop,moisture,temperature,humidity,tank FROM myhome.greenhouse ORDER BY
    event DESC LIMIT 5");

    if ($result->num_rows > 0) {

      // output data of each row
      while($row = $result->fetch_assoc()) {
        echo "data.addRow([new Date(".$row["year"].",".$row["month"]."-
        1,".$row["day"].",".$row["hour"].",".$row["minute"].",".$row["sec"]."),
        ".$row["light"].",".$row["raindrop"].",".$row["moisture"].",".$row["temperature"].",".$row["hu
        midity"].",".$row["tank"]."]);\\n";
      }
    }
    else {
    }

    ?>

    var light_options = {

```

```

chart: {
    title: 'Light vs Time',
    subtitle: 'measured in %'
},
hAxis: { format: 'HH:mm:ss a' },
width: 600,
height: 300
};

var raindrop_options = {
chart: {
    title: 'Rain vs Time',
    subtitle: 'measured in %'
},
hAxis: { format: 'HH:mm:ss a' },
width: 600,
height: 300
};

var moisture_options = {
chart: {
    title: 'Moisture vs Time',
    subtitle: 'measured in %'
},
hAxis: { format: 'HH:mm:ss a' },
width: 600,
height: 300
};

var temperature_options = {
chart: {
    title: 'Temperature vs Time',
    subtitle: 'measured in C'
},
hAxis: { format: 'HH:mm:ss a' },
width: 600,
height: 300
};

var humidity_options = {
chart: {
    title: 'Humidity vs Time',
    subtitle: 'measured in %'
},
hAxis: { format: 'HH:mm:ss a' },
width: 600,
height: 300
};

```

```

var tank_options = {
  chart: {
    title: 'Tank vs Time',
    subtitle: 'measured in cm'
  },
  hAxis: { format: 'HH:mm:ss a' },
  width: 600,
  height: 300
};
var all_options = {
  chart: {
    title: 'All measurements vs Time',
    subtitle: ''
  },
  hAxis: { format: 'HH:mm:ss a' },
  width: 600,
  height: 300
};
var light_chart = new google.charts.Line(document.getElementById('light_div'));

var light_view = new google.visualization.DataView(data);
light_view.setColumns([0,1]);
light_chart.draw(light_view, google.charts.Line.convertOptions(light_options));
//-----
var raindrop_chart = new google.charts.Line(document.getElementById('raindrop_div'));

var raindrop_view = new google.visualization.DataView(data);
raindrop_view.setColumns([0,2]);
raindrop_chart.draw(raindrop_view, google.charts.Line.convertOptions(raindrop_options));
//-----
var moisture_chart = new google.charts.Line(document.getElementById('moisture_div'));

var moisture_view = new google.visualization.DataView(data);
moisture_view.setColumns([0,3]);
moisture_chart.draw(moisture_view,
google.charts.Line.convertOptions(moisture_options));
//-----
var temperature_chart = new
google.charts.Line(document.getElementById('temperature_div'));

var temperature_view = new google.visualization.DataView(data);
temperature_view.setColumns([0,4]);
temperature_chart.draw(temperature_view,
google.charts.Line.convertOptions(temperature_options));
//-----
var humidity_chart = new google.charts.Line(document.getElementById('humidity_div'));

var humidity_view = new google.visualization.DataView(data);

```

```

    humidity_view.setColumns([0,5]);
    humidity_chart.draw(humidity_view,
google.charts.Line.convertOptions(humidity_options));

//-----
    var tank_chart = new google.charts.Line(document.getElementById('tank_div'));

    var tank_view = new google.visualization.DataView(data);
    tank_view.setColumns([0,6]);
    tank_chart.draw(tank_view, google.charts.Line.convertOptions(tank_options));
//-----
    var all_chart = new google.charts.Line(document.getElementById('all_div'));
    all_chart.draw(data, google.charts.Line.convertOptions(all_options));
}
</script>
</table>
</head>
<body>
    <div class="content"><br>
        <strong>Διαγράμματα συναρτήσει χρόνου δυναμικής ανανέωσης </strong><br>
        <p>Στα παρακάτω διαγράμματα απεικονίζονται οι τελευταίες τιμές των αισθητήρων του
        κήπου/θερμοκηπίου σε συνάρτηση με το χρόνο. Το τελευταίο διάγραμμα περιέχει όλες τις
        τιμές των αισθητήρων συναρτήσει του χρόνου.</p>
        <div style="width:100%">
        </div>

        <table style="border-spacing: 15px">
            <tr>
                <td style="border: 1px solid green">
                    <div id="light_div" style="width: 600px; height: 300px"></div>
                </td>
            </tr>
            <tr>
                <td style="border: 1px solid green">
                    <div id="raindrop_div" style="width: 600px; height: 300px"></div>
                </td>
            </tr>
            <tr><td style="border: 1px solid green">
                <div id="temperature_div" style="width: 600px; height: 300px"></div>
            </td>
            </tr>
            <tr>
                <td style="border: 1px solid green">
                    <div id="moisture_div" style="width: 600px; height: 300px"></div>
                </td>
            </tr>
            <tr>
                <td style="border: 1px solid green">
                    <div id="tank_div" style="width: 600px; height: 300px"></div>

```

```

        </td>
      </tr>
    <tr>
      <td style="border: 1px solid green">
        <div id="humidity_div" style="width: 600px; height: 300px"></div>
      </td>
    </tr>
    <tr>
      <td style="border: 1px solid green">
        <div id="all_div" style="width: 600px; height: 300px"></div>
      </td>
    </tr>
  </table>

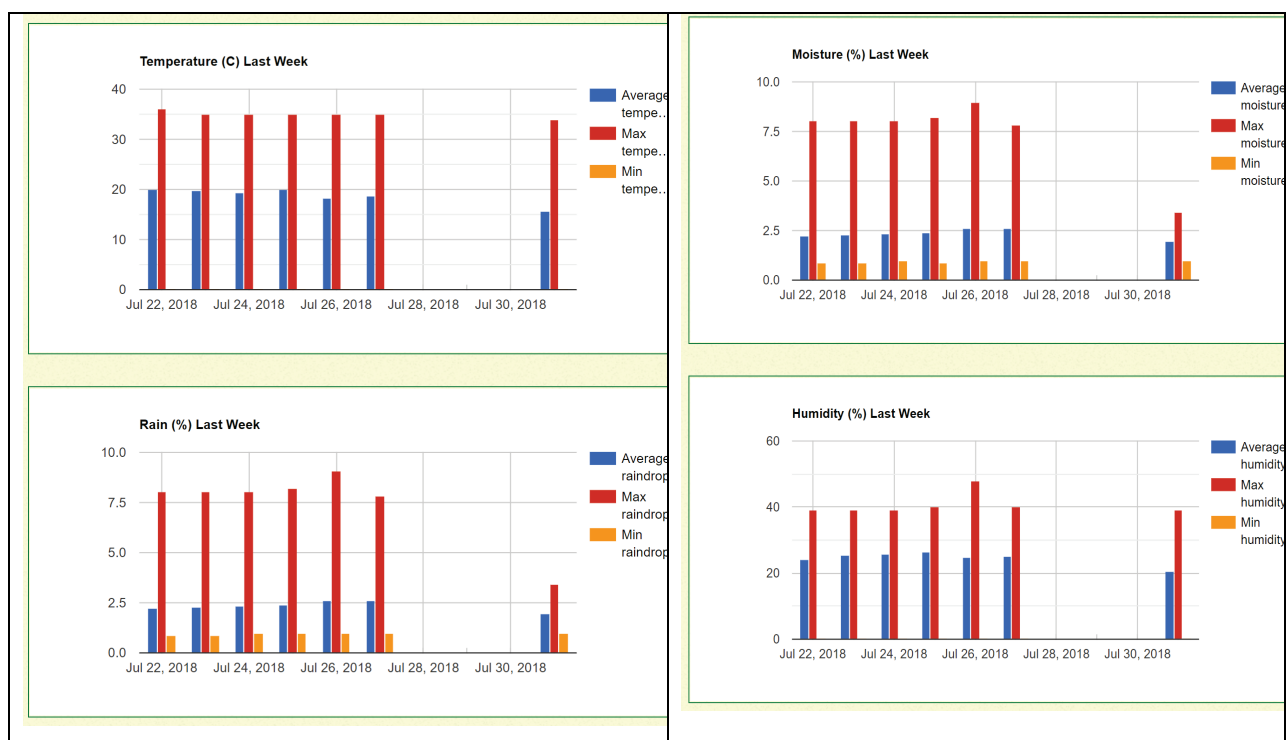
  <p></p>
</div>
</div>
</body>
</html>

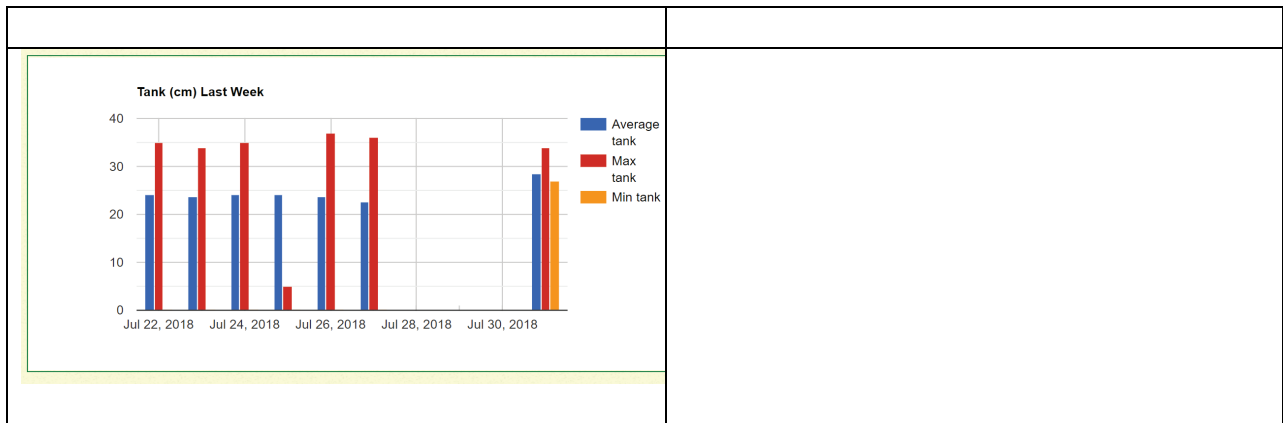
```

Κώδικας 7-31 Κώδικας Ιστοσελίδας

7.3.4 WEEKLY STATISTICS (WEB SERVER ΣΕ RASPBERRY PI)

Εβδομαδιαία ενδεικτικά στατιστικά δεδομένων αισθητήρων (τρέχει σε web server)





Εικόνα 7-14 Weekly Statistics (web server)

Κώδικας Ιστοσελίδας

```
<html>
<head>
<meta charset="UTF-8">
<title>Weekly Statistics</title>
<link rel="stylesheet" href="../../css/style.css" type="text/css">
<meta http-equiv="refresh" content="86400" />
<script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
<script type="text/javascript">
    google.charts.load('current', {'packages':['corechart']});
    google.charts.setOnLoadCallback(drawChart);

    function drawChart() {

        var data = new google.visualization.DataTable();
        data.addColumn('datetime', 'event');
        data.addColumn('number', 'Average light');
        data.addColumn('number', 'Max light');
        data.addColumn('number', 'Min light');
        data.addColumn('number', 'Average raindrop');
        data.addColumn('number', 'Max raindrop');
        data.addColumn('number', 'Min raindrop');
        data.addColumn('number', 'Average moisture');
        data.addColumn('number', 'Max moisture');
        data.addColumn('number', 'Min moisture');
        data.addColumn('number', 'Average temperature');
        data.addColumn('number', 'Max temperature');
        data.addColumn('number', 'Min temperature');
        data.addColumn('number', 'Average humidity');
        data.addColumn('number', 'Max humidity');
        data.addColumn('number', 'Min humidity');
        data.addColumn('number', 'Average tank');
        data.addColumn('number', 'Max tank');
        data.addColumn('number', 'Min tank');

    }
</script>
<?php
    // Prepare variables for database connection

    $dbusername = "root"; // enter database username, I used "arduino" in step 2.2
    $dbpassword = "password"; // enter database password, I used "arduinotest" in step 2.2
    $server = "localhost";
```



```
// Connect to your database
$mysqli = new mysqli($server, $dbusername, $dbpassword, "myhome");
if($mysqli->connect_error)
    die('Connect Error (' . mysqli_connect_errno() . ') ' . mysqli_connect_error());

// Execute SQL statement ta grouparw ana imera kai gia kathe mera to min kai to max kai to average kai krataw
tis teleytaies 7 meres
$result = $mysqli->query("SELECT YEAR(event) AS year, MONTH(event) AS month, DAY(event) AS day,
AVG(light),MAX(light),MIN(light), AVG(raindrop),MAX(raindrop),MIN(raindrop),
AVG(moisture),MAX(moisture),MIN(moisture), AVG(temperature),MAX(temperature),MIN(temperature),
AVG(humidity),MAX(humidity),MIN(humidity), AVG(tank),MAX(tank),MIN(tank) from sensors group by DAY(event)
ORDER BY event DESC limit 7");

if ($result->num_rows > 0) {

    // output data of each row
    while($row = $result->fetch_assoc()) {
        echo "data.addRow([new Date(".$row["year"].",".$row["month"]."-
1,".$row["day"]."),".$row["AVG(light)"].",".$row["MAX(light)"].",".$row["MIN(light)"].",".$row["AVG(raindrop)"].",
".$row["MAX(raindrop)"].",".$row["MIN(raindrop)"].",".$row["AVG(moisture)"].",".$row["MAX(moisture)"].",".$ro
w["MIN(moisture)"].",".$row["AVG(temperature)"].",".$row["MAX(temperature)"].",".$row["MIN(temperature)"].
",".$row["AVG(humidity)"].",".$row["MAX(humidity)"].",".$row["MIN(humidity)"].",".$row["AVG(tank)"].",".$row["
MAX(tank)"].",".$row["MIN(tank)"]."]);\\n";
    }
}
else {
}

?>
var light_options = {
    title: 'Light (%) Last Week',
    subtitle: 'measured in %',
    hAxis: { format: 'MMM dd, yyyy' },
    seriesType: 'bars',
    width: 600,
    height: 300
};

var raindrop_options = {
    title: 'Rain (%) Last Week',
    subtitle: 'measured in %',
    hAxis: { format: 'MMM dd, yyyy' },
    seriesType: 'bars',
    width: 600,
    height: 300
};

var moisture_options = {
    title: 'Moisture (%) Last Week',
    subtitle: 'measured in %',
    hAxis: { format: 'MMM dd, yyyy' },
    seriesType: 'bars',
    width: 600,
    height: 300
};

var temperature_options = {
    title: 'Temperature (C) Last Week',
    subtitle: 'measured in C',
```

```

    hAxis: { format: 'MMM dd, yyyy' },
    seriesType: 'bars',
    width: 600,
    height: 300
  };

  var humidity_options = {
    title: 'Humidity (%) Last Week',
    subtitle: 'measured in %',
    hAxis: { format: 'MMM dd, yyyy' },
    seriesType: 'bars',
    width: 600,
    height: 300
  };

  var tank_options = {
    title: 'Tank (cm) Last Week',
    subtitle: 'measured in cm',
    hAxis: { format: 'MMM dd, yyyy' },
    seriesType: 'bars',
    width: 600,
    height: 300
  };

  var light_chart = new google.visualization.ComboChart(document.getElementById('light_div'));

  var light_view = new google.visualization.DataView(data);
  light_view.setColumns([0,1,2,3]);
  light_chart.draw(light_view, light_options);

  //-----
  var raindrop_chart = new google.visualization.ComboChart(document.getElementById('raindrop_div'));

  var raindrop_view = new google.visualization.DataView(data);
  raindrop_view.setColumns([0,4,5,6]);
  raindrop_chart.draw(raindrop_view, raindrop_options);

  //-----
  var moisture_chart = new google.visualization.ComboChart(document.getElementById('moisture_div'));

  var moisture_view = new google.visualization.DataView(data);
  moisture_view.setColumns([0,7,8,9]);
  moisture_chart.draw(moisture_view, moisture_options);

  //-----
  var temperature_chart = new
  google.visualization.ComboChart(document.getElementById('temperature_div'));

  var temperature_view = new google.visualization.DataView(data);
  temperature_view.setColumns([0,10,11,12]);
  temperature_chart.draw(temperature_view, temperature_options);

  //-----
  var humidity_chart = new google.visualization.ComboChart(document.getElementById('humidity_div'));

  var humidity_view = new google.visualization.DataView(data);
  humidity_view.setColumns([0,13,14,15]);
  humidity_chart.draw(humidity_view, humidity_options);

```

```

var tank_chart = new google.visualization.ComboChart(document.getElementById('tank_div'));

var tank_view = new google.visualization.DataView(data);
tank_view.setColumns([0,16,17,18]);
tank_chart.draw(tank_view, tank_options);
}
</script>

</head>
<body>

  <div class="content">
    
    <br>
    <strong>Εβδομαδιαία Στατιστικά</strong><br>
    <p>Στα παρακάτω διαγράμματα απεικονίζονται η ελάχιστη, μέγιστη και μέση τιμή των αισθητή-
    ρων (φωτός, βροχής, υγρασίας εδάφους, Θερμοκρασίας, υγρασίας, επιπέδου νερού) για την κάθε μέρα της τελευ-
    ταιάς εβδομάδας.</p>

    <body>
    <div style="width:100%">

    </div>

    <table style="border-spacing: 15px">
      <tr>
        <td style="border: 1px solid green">
          <div id="temperature_div" style="width: 600px; height: 300px"></div>
        </td>
        <td style="border: 1px solid green">
          <div id="raindrop_div" style="width: 600px; height: 300px"></div>
        </td>
      </tr>
      <tr>
        <td style="border: 1px solid green">
          <div id="moisture_div" style="width: 600px; height: 300px"></div>
        </td>
        <td style="border: 1px solid green">
          <div id="light_div" style="width: 600px; height: 300px"></div>
        </td>
      </tr>
      <tr>
        <td style="border: 1px solid green">
          <div id="humidity_div" style="width: 600px; height: 300px"></div>
        </td>
        <td style="border: 1px solid green">
          <div id="tank_div" style="width: 600px; height: 300px"></div>
        </td>
      </tr>
    </table>
  </body>

```

```

<tr>
  <td style="border: 0px solid green">
    <div id="all_div" style="width: 600px; height: 300px"></div>
  </td>
</tr>
</table>

</div>
</div>
</body>
</html>

```

Κώδικας 7-32 Κώδικας Ιστοσελίδας

7.4 ΔΙΑΔΙΚΑΣΙΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΣΕΝΑΡΙΟΥ ΚΗΠΟΥ/ΘΕΡΜΟΚΗΠΙΟΥ ΜΕ THINKSPEAK ΚΑΙ IFTTT

7.4.1 THINKSPEAK

Η υπηρεσία Thingspeak είναι μια εφαρμογή του Internet of Things (IOT) που επιτρέπει την αποθήκευση και επεξεργασία δεδομένων χρησιμοποιώντας το πρωτόκολλο HTTP, είτε μέσω ασύρματης είτε ενσύρματης σύνδεσης στο διαδίκτυο. Η πλατφόρμα υποστηρίζει κώδικα Matlab, κάνοντάς την ιδιαίτερα εύχρηστη για νέους χρήστες που θέλουν να απεικονίζουν τα δεδομένα τους γρήγορα και εύκολα. Η υπηρεσία προσφέρεται δωρεάν, αλλά υπάρχουν περιορισμοί στον όγκο των δεδομένων που μπορούν να ανέβουν καθώς και στο χρονικό διάστημα ανανέωσής τους. Συγκεκριμένα, προσφέρονται 3.000.000 (τρία εκατομμύρια) δωρεάν μετρήσεις με ελάχιστο χρόνο ανανέωσης τα 15 δευτερόλεπτα. Επιπλέον, ανά κανάλι επιτρέπεται η εισαγωγή μόνο 8 πεδίων, δηλαδή 8 τιμών.

Δημιουργία λογαριασμού

Αφού μεταβούμε στην διεύθυνση <https://thingspeak.com>, πατάμε sign up και συμπληρώνουμε τα στοιχεία μας στη φόρμα που εμφανίστηκε.

Εικόνα 7-15 Δημιουργία λογαριασμού

Κανάλια

Πατάμε την μπάρα “my channels” και επιλέγουμε “new channel”.

Εικόνα 7-16 Δημιουργία καναλιών

Εκεί μπορούμε να:

- Προσθέτουμε πεδία για να ανεβάζουμε τις τιμές μας.
- Ονομάζουμε τα πεδία κατάλληλα.
- Γράφουμε συμπληρωματικά δεδομένα για το κανάλι μας.
- Προσθέτουμε ετικέτες ώστε άλλοι χρήστες να μπορούν να βρουν το κανάλι μας.
- Δείχνουμε την πραγματική τοποθεσία του καναλιού στο Google Maps.
- Προσθέτουμε εξωτερικό σύνδεσμο της προσωπικής μας σελίδας.
- Επισυνάπτουμε κάποιο δικό μας βίντεο.
- Τέλος πατάμε save ώστε να αποθηκεύσουμε τυχόν αλλαγές.

Για να ανεβάσουμε δεδομένα μέσω του Arduino, ακολουθούμε τα εξής βήματα:

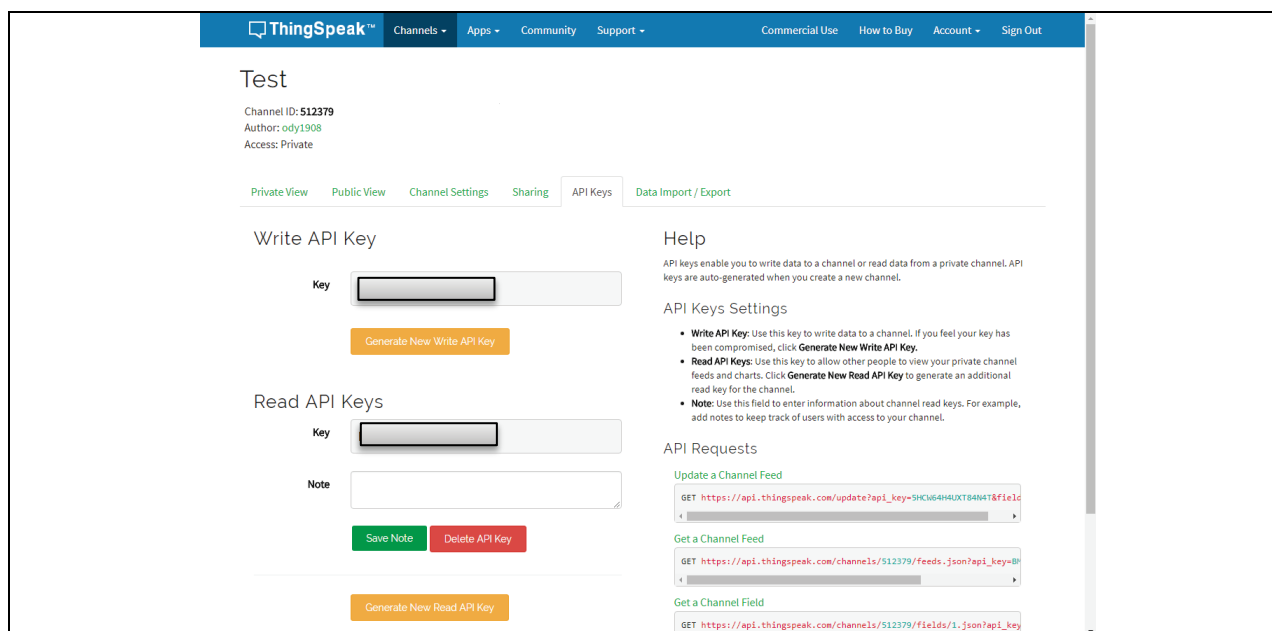
- Κατεβάζουμε και εγκαθιστούμε την επίσημη βιβλιοθήκη του ThingSpeak, "thing-speak.h", από αυτόν τον σύνδεσμο: ThingSpeak Arduino Library.
- Ανοίγουμε το Arduino IDE και πηγαίνουμε στο File → Examples → ThingSpeak → WriteMultipleVoltages.
- Αυτό το αρχείο περιέχει τον κώδικα που χρειαζόμαστε για να ανεβάσουμε τα δεδομένα μας στην υπηρεσία.
- Επιλέγουμε το κατάλληλο τμήμα κώδικα ανάλογα με το module που χρησιμοποιούμε για να συνδέσουμε το Arduino στο διαδίκτυο (Ethernet shield, WiFi101 shield, MKR1000).
- Στο παράδειγμά μας, θα χρησιμοποιήσουμε το Ethernet shield και ο προσαρμοσμένος κώδικας είναι:

"thingspeakexample.ino"

```
#include <SPI.h>
#include <Ethernet.h>
#include <ThingSpeak.h>
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED};
EthernetClient client;
// On Arduino: 0 - 1023 maps to 0 - 5 volts
#define VOLTAGE_MAX 5.0
#define VOLTAGE_MAXCOUNTS 1023.0
unsigned long myChannelNumber = .....;
const char * myWriteAPIKey = ".....";
void setup() {
  Ethernet.begin(mac);
  ThingSpeak.begin(client);
}
void loop() {
  float pinVoltage = analogRead(A0) * (VOLTAGE_MAX / VOLTAGE_MAXCOUNTS);
  ThingSpeak.setField(1,pinVoltage);
  ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
  delay(20000);
}
```

Κώδικας 7-33 Κώδικας Ino

Στο προηγούμενο παράδειγμα, μετράμε την τάση στο αναλογικό ακροδέκτη του Arduino, μετατρέπουμε το αναλογικό σήμα σε τιμή τάσης και στη συνέχεια αποστέλλουμε αυτήν την τιμή στο πρώτο πεδίο του αντίστοιχου καναλιού. Οι αλλαγές που χρειάζονται είναι στις τιμές των μεταβλητών "myChannelNumber" και "myWriteAPIKey". Αυτές οι δύο παράμετροι είναι μοναδικές για κάθε κανάλι και μπορούν να βρεθούν επισκεπτόμενοι το κανάλι μας και πατώντας στην επιλογή "API Keys". Μια νέα μέτρηση λαμβάνει χώρα κάθε 20 δευτερόλεπτα, καθορισμένη από την εντολή "delay(20000);".



Εικόνα 7-17 WriteAPIKey

Η μεταβλητή "myChannelNumber" περιέχει το Channel ID και η μεταβλητή "myWriteAPIKey" το Write API Key.

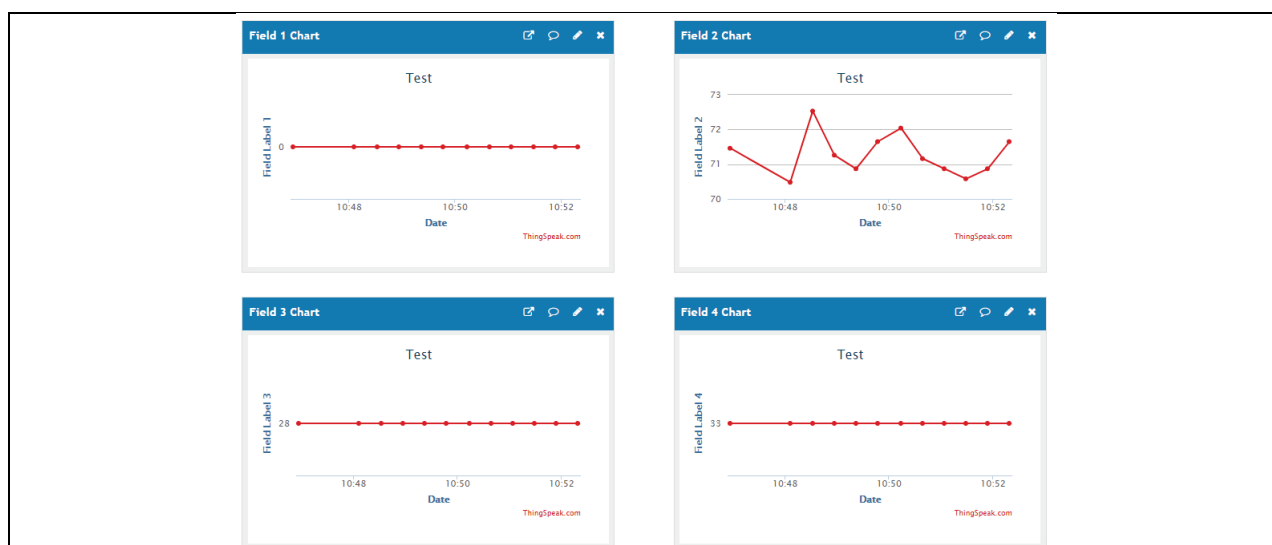
Με το "myChannelNumber" και το "myWriteAPIKey". δηλώνουμε σε ποιο field θέλουμε να ανεβάσουμε την τιμή μας ,την μεταβλητή που θα περιέχει.

```
ThingSpeak.setField(1,pinVoltage);
```

```
ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
```

Επεξεργασία Fields

Έχουμε ανεβάσει τυπικά κάποιες τιμές για να επεξηγήσουμε τις ιδιότητες των fields.



Εικόνα 7-18 Επεξεργασία Fields

Οι ιδιότητες της απεικόνισης των fields είναι οι εξής:

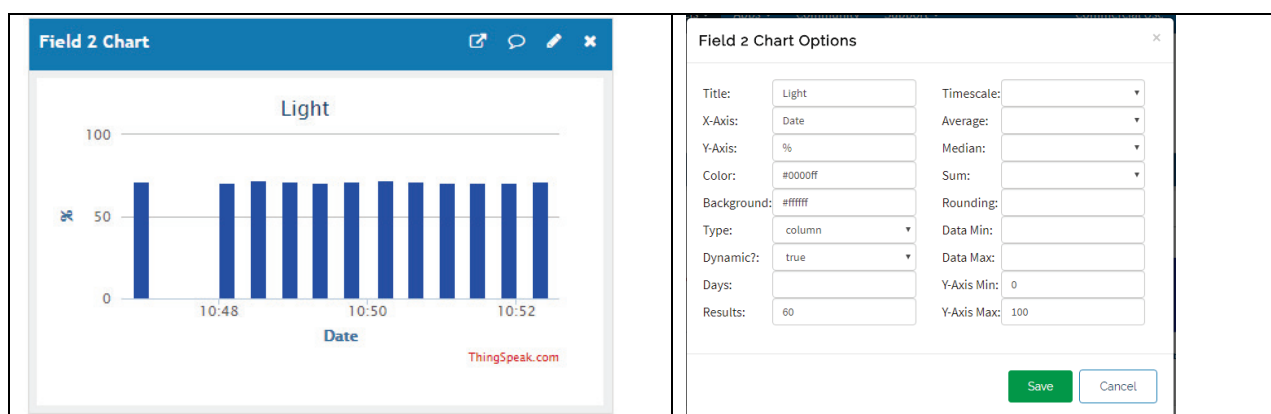
Field 1 Chart Options	
Title:	
X-Axis:	
Y-Axis:	
Color:	#d62020
Background:	#ffffff
Type:	line
Dynamic?:	true
Days:	
Results:	60
Timescale:	
Average:	
Median:	
Sum:	
Rounding:	
Data Min:	
Data Max:	
Y-Axis Min:	
Y-Axis Max:	

Save Cancel

Εικόνα 7-19 Ιδιότητες της απεικόνισης των fields

- Τίτλος.
- Όνομα αξόνων.
- Χρώμα της απεικόνισης των τιμών.
- Χρώμα του παρασκηνίου.
- Επιλογή του στυλ της γραφικής.
- Επιλογή για δυναμική ανανέωση των δεδομένων.
- Τον περιορισμό ως προς τις μέρες που θα ανεβαίνουν δεδομένα.
- Πόσες μετρήσεις θα παρουσιάζονται στο γράφημα.
- Κάθε πότε θα ανανεώνεται το κανάλι.
- Υπολογισμός της διαμέσου τιμής.
- Υπολογισμός της μέσης τιμής.
- Άθροισμα των τιμών .
- Δυνατότητα στρογγυλοποίησης των τιμών.
- Μικρότερη και μεγαλύτερη τιμή της γραφικής.
- Τέλος μεγαλύτερη και μικρότερη τιμή απεικόνισης στους άξονες του γραφήματος.

Παράδειγμα παραμετροποιημένου field.



Εικόνα 7-20 Παράδειγμα παραμετροποιημένου field

Επεξεργασία τιμών

Η υπηρεσία επιτρέπει την επεξεργασία των δεδομένων που αποστέλλετε χρησιμοποιώντας κώδικα MATLAB. Θα παρέχετε μερικά βασικά παραδείγματα, όπως τον υπολογισμό της μέγιστης, ελάχιστης και μέσης τιμής, αν και οι χρήστες μπορούν να προσθέσουν τον δικό τους MATLAB κώδικα. Αυτές οι λειτουργίες μπορούν να εκτελεστούν στην ενότητα "MATLAB Analysis" της υπηρεσίας σας.

Το κείμενο δίνει μια επισκόπηση των δυνατοτήτων επεξεργασίας δεδομένων που προσφέρει η υπηρεσία σας μέσω της χρήσης MATLAB. Επισημαίνει ότι παρέχετε κάποια βασικά παραδείγματα ανάλυσης, αλλά οι χρήστες μπορούν να προσθέσουν τον δικό τους MATLAB κώδικα για περαιτέρω επεξεργασία στην ενότητα "MATLAB Analysis".

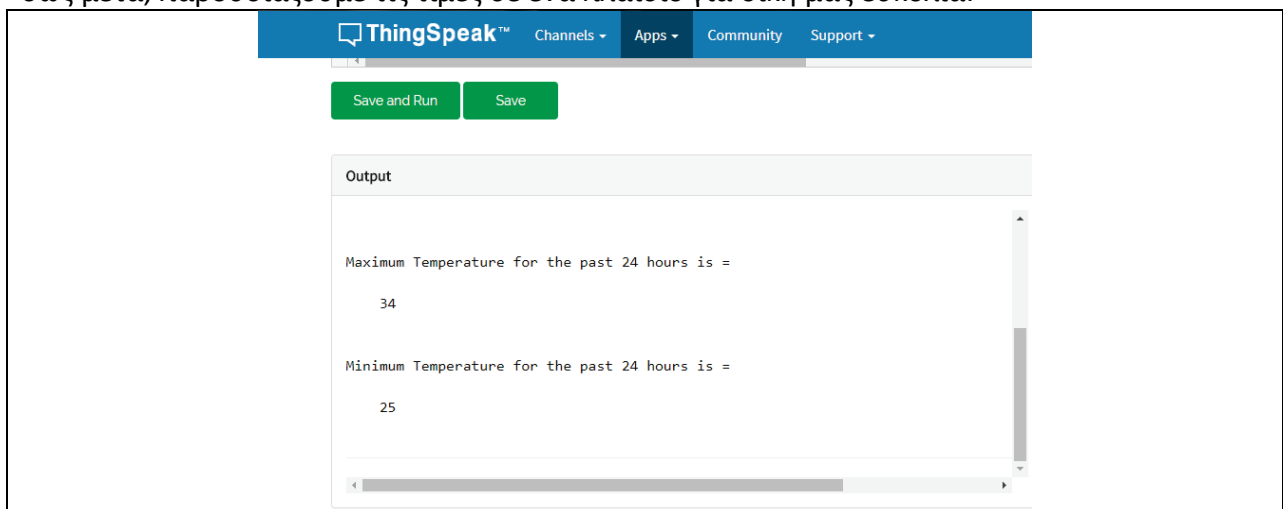
Εικόνα 7-21 Επεξεργασία τιμών

Για τον υπολογισμό της μέγιστης και ελάχιστης θερμοκρασίας μιας ημέρας επιλέγουμε το παράδειγμα "Calculate high and low temperatures", το τροποποιούμε ταυτόχρονα ώστε να γράφει τους υπολογισμούς σε ένα δεύτερο κανάλι.

```
readChannelID = .....;
TemperatureFieldID = 2;
readAPIKey = '.....';
[tempF,timeStamp] = thing-
SpeakRead(readChannelID,'Fields',TemperatureFieldID,'numDays',1,'ReadKey',readAPIKey);
[maxTempF,maxTempIndex] = max(tempF);
[minTempF,minTempIndex] = min(tempF);
timeMaxTemp = timeStamp(maxTempIndex);
timeMinTemp = timeStamp(minTempIndex);
display(maxTempF,'Maximum Temperature for the past 24 hours is');
display(minTempF,'Minimum Temperature for the past 24 hours is');
writeChannelID = [];
writeAPIKey = '';
thingSpeakWrite(writeChannelID,{maxTempF,minTempF},'Writekey',writeAPIKey);
```

Κώδικας 7-34 Κώδικας

Στις πρώτες τρεις γραμμές του κώδικα, εισάγουμε τα στοιχεία του καναλιού από το οποίο παίρνουμε τα δεδομένα μας, το οποίο θα ονομάσουμε "Βασικό". Στην 4η γραμμή παρακολουθείται η θερμοκρασία για το χρονικό διάστημα που έχουμε ορίσει (σε αυτή την περίπτωση είναι 1 μέρα). Αυτό καθορίζεται ως εξής: για μία ώρα χρησιμοποιούμε 'numminutes', 60, για μία μέρα 'numDays', 1, και για επτά μέρες 'numDays', 7. Οι επόμενες τέσσερις γραμμές (6-9) υπολογίζουν την ελάχιστη και μέγιστη θερμοκρασία καθώς και την ώρα που καταγράφηκαν. Αμέσως μετά, παρουσιάζουμε τις τιμές σε ένα πλαίσιο για δική μας ευκολία.



Εικόνα 7-22 Παρουσίαση τιμών σε ένα πλαίσιο

Αν θέλουμε να εμφανίσουμε τα αποτελέσματα σε ένα άλλο κανάλι συμπληρώνουμε τα στοιχεία του στις γραμμές 12,13 προσέχοντας στην γραμμή 12 το νούμερο του καναλιού να γρα-

φτεί επάνω στις αγκύλες και όχι ενδιάμεσα τους. Τέλος στην γραμμή 14 δηλώνουμε ποιες μεταβλητές θέλουμε να γραφτούν στο κανάλι που έχουμε επιλέξει. Εάν δεν θέλουμε να απεικονιστεί σε άλλο κανάλι το αποτέλεσμα απλά κάνουμε σχόλιο την γραμμή 14 προσθέτοντας το σύμβολο του “%” στην αρχή της.

Για τον υπολογισμό της μέσης υγρασίας για μία ώρα επιλέγουμε το παράδειγμα “Calculate and display average humidity”, το τροποποιούμε ταυτόχρονα ώστε να γράφει τον υπολογισμό σε ένα δεύτερο κανάλι.

```
readChannelID = .....;
humidityFieldID = 3;
readAPIKey = '.....';
humidity = thing-
SpeakRead(readChannelID,'Fields',humidityFieldID,'NumMinutes',60,'ReadKey',readAPIKey);
avgHumidity = mean(humidity);
display(avgHumidity,'Average Humidity');
writeChannelID = [];
writeAPIKey = "";
thingSpeakWrite(writeChannelID,avgHumidity,'writekey',writeAPIKey);
```

Κώδικας 7-35 Κώδικας

Ακριβώς τα ίδια ισχύουν και για αυτό τον υπολογισμό. Εάν δεν θέλουμε να απεικονιστεί σε άλλο κανάλι το αποτέλεσμα απλά κάνουμε σχόλιο την γραμμή 14 προσθέτοντας το σύμβολο του % στην αρχή της.

Time control

Το time control είναι μία εξαιρετικά χρήσιμη εφαρμογή που μας επιτρέπει να εκτελέσουμε αυτόματα τα Matlab analysis που έχουμε δημιουργήσει. Πατώντας στην ενότητα time control και έπειτα new μπορούμε να αρχίσουμε να φτιάχνουμε το δικό μας με βάση τις ανάγκες μας.

The screenshot shows the ThingSpeak web interface for configuring a new TimeControl app. The main form includes fields for Name, Time Zone, Frequency, Date, Time, Fuzzy Time, and Action. The right sidebar provides detailed help and settings for the TimeControl app, including instructions on how to use it and a list of available actions like MATLAB Analysis, ThingHTTP, ThingTweet, and TalkBack.

Εικόνα 7-23 Time Control

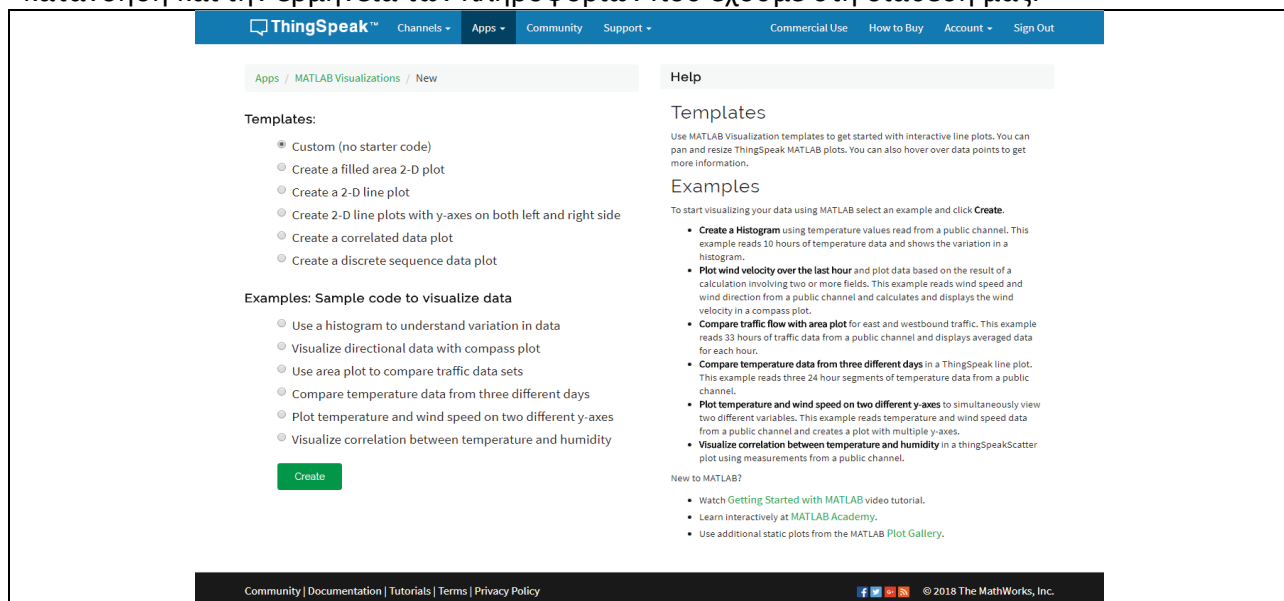
Αρχικά του δίνουμε ένα όνομα και ρυθμίζουμε τη ζώνη ώρας που βρισκόμαστε, δηλώνουμε αν θέλουμε να τρέχουμε τον κώδικά μας μία φορά ή επαναλαμβανόμενα μέσω της επιλογής frequency. Οι επιλογές που έχουμε είναι:

- Να επιλέξουμε συγκεκριμένη μέρα και ώρα (στην επιλογή one time).
- Να επιλέξουμε recurring.
- Να επιλέξουμε να τρέχει ανά 1,2,3 και ούτω καθεξής ώρες.
- Να επιλέξουμε να τρέχει μία φορά ανά μέρα μία συγκεκριμένη ώρα.
- Να επιλέξουμε να τρέχει μία φορά την εβδομάδα.
- Να επιλέξουμε να τρέχει συγκεκριμένες μέρες μέσα στη βδομάδα .
- Τέλος επιλέγουμε ποιο Matlab analysis να ενεργοποιηθεί και πατάμε save time control.

Matlab Visualizations

Αυτή η δυνατότητα μας επιτρέπει να δημιουργήσουμε διάφορα είδη γραφικών παραστάσεων, προσφέροντας μια πιο άμεση και οπτική απεικόνιση των δεδομένων μας. Επιλέγοντας την ενότητα "Matlab visualization", θα δούμε μια εικόνα που παρουσιάζει τις διαθέσιμες λειτουργίες για την οπτικοποίηση των δεδομένων.

Με αυτό το εργαλείο, μπορούμε να δημιουργήσουμε γραφικές αναπαραστάσεις, όπως γραφήματα, διαγράμματα και άλλες μορφές οπτικοποίησης, οι οποίες μας επιτρέπουν να αντιληφθούμε και να αναλύσουμε καλύτερα τα δεδομένα μας. Αυτή η δυνατότητα ενισχύει την κατανόηση και την ερμηνεία των πληροφοριών που έχουμε στη διάθεσή μας.

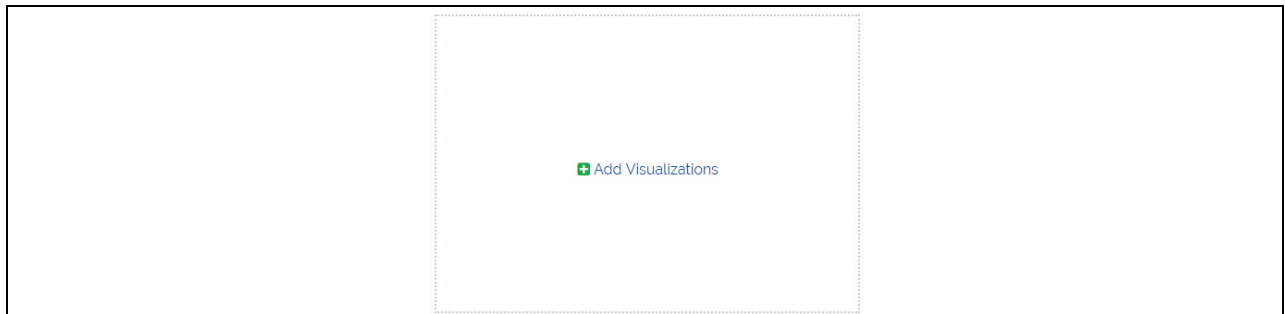


Εικόνα 7-24 Matlab Visualizations 1

Το στοιχείο που παραμένει ίδιο σε σχέση με τα Matlab analysis είναι ότι και εδώ θα πρέπει κάθε φορά να δηλώνουμε τα στοιχεία του καναλιού που έχουμε τα δεδομένα μας . Τα Matlab visualizations μπορούν να εμφανιστούν στην δημόσια μορφή του καναλιού μας.

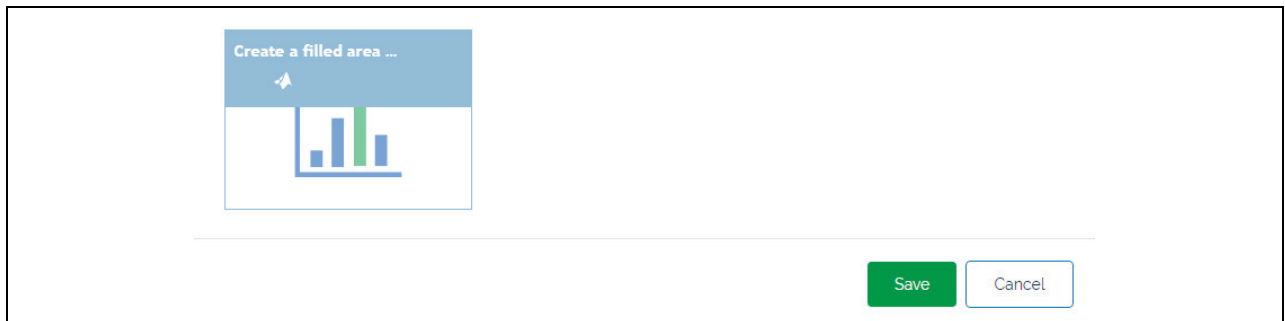
Παρουσίαση Matlab visualizations

Όταν έχουμε φτιάξει το γραφικό στοιχείο που θέλουμε(εδώ:Create a filled area 2-D plot) κατευθυνόμαστε στο κανάλι που θέλουμε να το παρουσιάσουμε, πατάμε:



Εικόνα 7-25 Matlab Visualizations 2

Θα δούμε την εξής εικόνα:



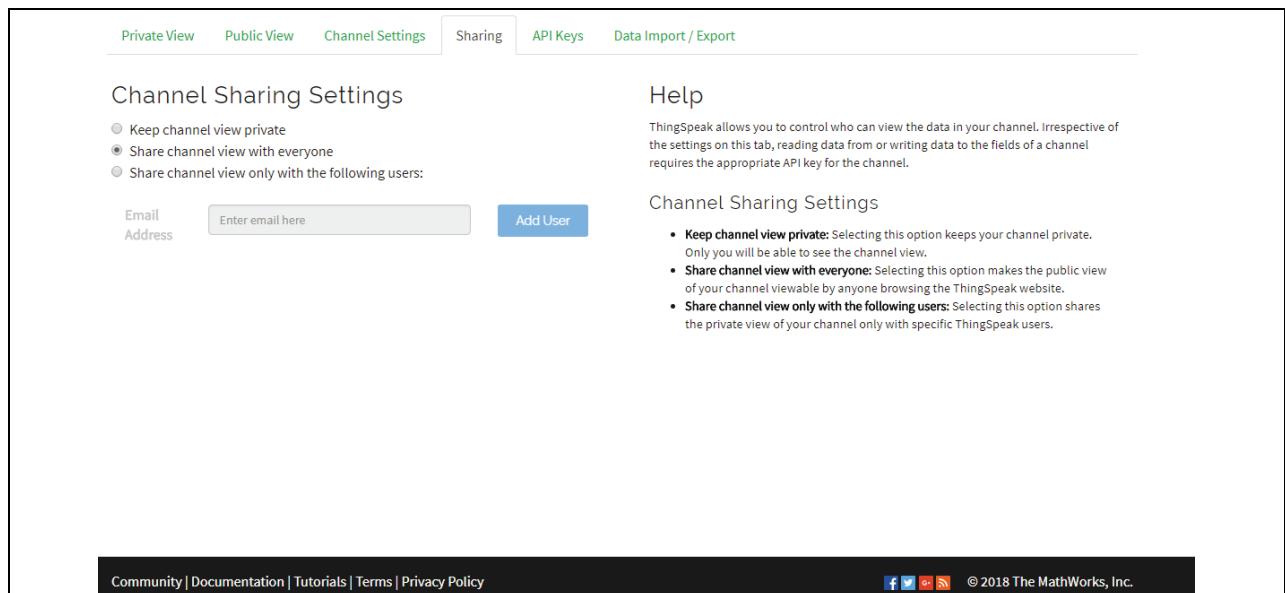
Εικόνα 7-26 Matlab Visualizations 3

Όταν επιλέξουμε το γραφικό και το αποθηκεύσουμε θα έχει εμφανιστεί κανονικά στο κανάλι μας.

Public View

Ένα κανάλι μπορεί να γίνει δημόσιο αν θέλουμε να μοιραστούμε τα δεδομένα και τις γραφικές μας με άλλους χρήστες.

Πατώντας στην ενότητα sharing έχουμε τις παρακάτω επιλογές.



Εικόνα 7-27 Public View/Sharing

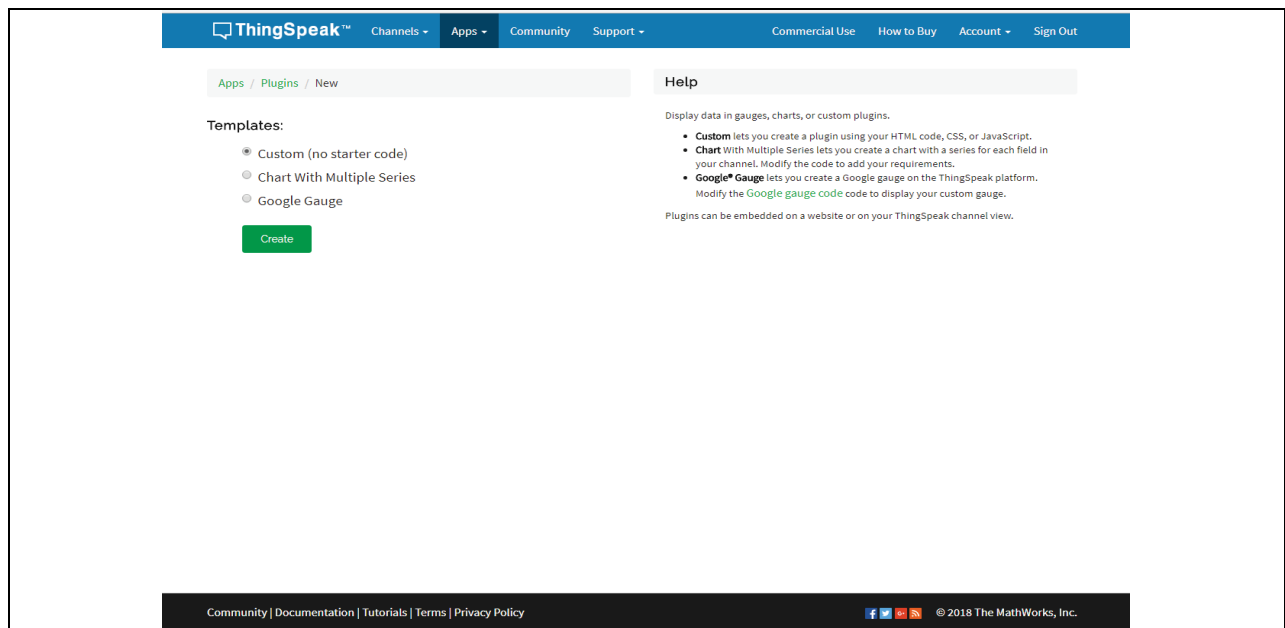
- Να κρατήσουμε το κανάλι ιδιωτικό .
- Να το μοιραστούμε με όλους τους χρήστες της πλατφόρμας.
- Να το μοιραστούμε με συγκεκριμένους χρήστες στους οποίους θα σταλθεί μέσω email.
 - Ο περιορισμός που υπάρχει στον δωρεάν λογαριασμό μας είναι ότι μπορούμε να το στείλουμε το πολύ σε τρία διαφορετικά emails.

Εξαγωγή/Εισαγωγή στοιχείων

Υπάρχει η δυνατότητα να ανεβάσουμε κατάλληλα δομημένα αρχεία στην πλατφόρμα ώστε να έχουμε έτοιμα δεδομένα αλλά και να τα αποθηκεύσουμε στον υπολογιστή μας σε μορφή CSV αφού πρώτα συμπληρώσουμε την ζώνη ώρας που βρισκόμαστε.

Plugins

Πατώντας στην ενότητα plugins→ εμφανίζονται οι εξής επιλογές :



Εικόνα 7-28 Plugins

Δηλαδή να τρέξουμε τον δικό μας κώδικα ,να φτιάξουμε μια γραφική που θα περιέχει πολλές μεταβλητές και κυκλικούς μετρητές.

Θα δούμε ένα παράδειγμα μετρητή(gauge).

Επιλέγουμε Google Gauge και θα μεταφερθούμε στην εξής σελίδα όπου θα έχουμε τη δυνατότητα να γράψουμε κώδικα html, css ,javascript.

HTML

```
<html>
<head>
<!--
NOTE: This plugin will not be visible on public views of a channel.
If you intend to make your channel public, consider using the
MATLAB Visualization App to create your visualizations.
-->
<title>Google Gauge - ThingSpeak</title>
%%PLUGIN_CSS%%
%%PLUGIN_JAVASCRIPT%%
</head>
<body>
<div id="container">
<div id="inner">
<div id="gauge_div"></div>
</div>
</div>
</body>
</html>
```

Κώδικας 7-36 Κώδικας HTML

CSS

```
<style type="text/css">
body { background-color: #ddd; }
```

```
#container { height: 100%; width: 100%; display: table; }  
#inner { vertical-align: middle; display: table-cell; }  
#gauge_div { width: 120px; margin: 0 auto; }  
</style>
```

Κώδικας 7-37 Κώδικας CSS

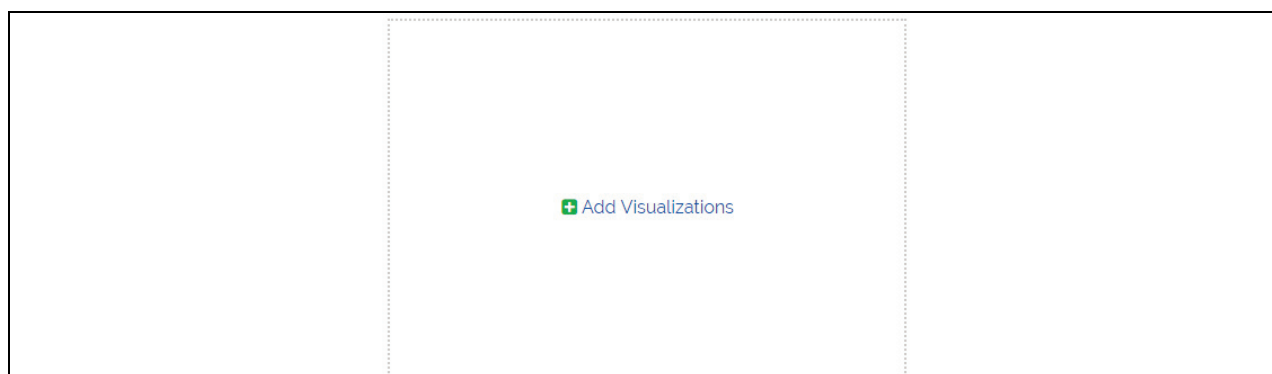
Στο παράδειγμά μας θα επεξεργαστούμε μόνο το javascript κώδικα.

Javascript

```
<script type='text/javascript' src='https://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js'></script>  
<script type='text/javascript' src='https://www.google.com/jsapi'></script>  
<script type='text/javascript'>  
var channel_id = 9;  
var api_key = '';  
var max_gauge_value = 1023;  
var gauge_name = 'Light Level';  
var chart, charts, data;  
google.load('visualization', '1', {packages:['gauge']});  
google.setOnLoadCallback(initChart);  
function displayData(point) {  
data.setValue(0, 0, gauge_name);  
data.setValue(0, 1, point);  
chart.draw(data, options);  
}  
function loadData() {  
var p;  
$.getJSON('https://api.thingspeak.com/channels/' + channel_id + '/feed/last.json?api_key=' + api_key, func-  
tion(data) {  
p = data.field1;  
if (p) {  
p = Math.round((p / max_gauge_value) * 100);  
displayData(p);  
}  
});  
}  
function initChart() {  
data = new google.visualization.DataTable();  
data.addColumn('string', 'Label');  
data.addColumn('number', 'Value');  
data.addRows(1);  
chart = new google.visualization.Gauge(document.getElementById('gauge_div'));  
options = {width: 120, height: 120, redFrom: 90, redTo: 100, yellowFrom: 75, yellowTo: 90, minorTicks: 5};  
loadData();  
setInterval('loadData()', 15000);  
}  
</script>
```

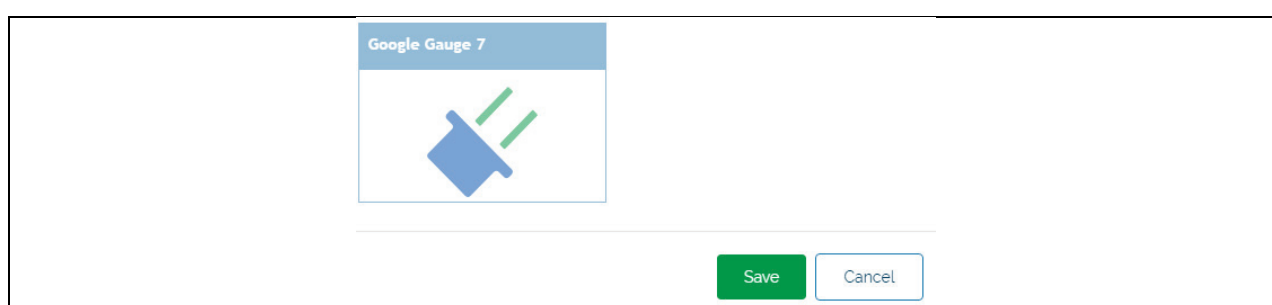
Κώδικας 7-38 Κώδικας Javascript

Τέλος πατάμε αποθήκευση και μεταφερόμαστε στο κανάλι που θέλουμε να το εμφανίσουμε.



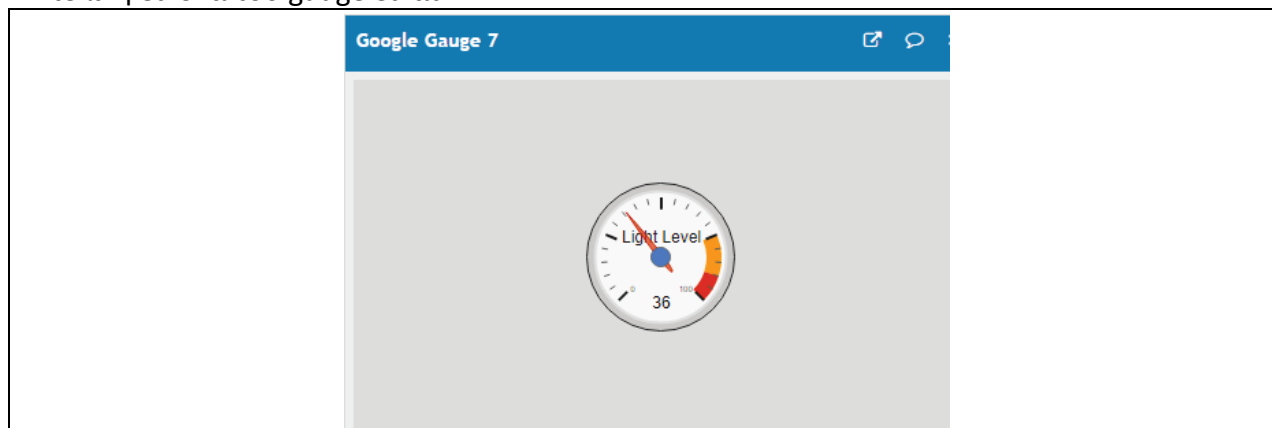
Εικόνα 7-29 Matlab Visualizations 4

Έπειτα και στη συνέχεια αποθήκευση.



Εικόνα 7-30 Matlab Visualizations 5

Η τελική εικόνα του gauge είναι:



Εικόνα 7-31 Gauge

Πατώντας

+ Add Widgets

Στην κύρια σελίδα του καναλιού επιλέγουμε το gauge→next και εμφανίζεται η παρακάτω εικόνα:

Configure widget parameters

Title

Enter Title for the widget

Field

Field 1

Min

0

Max

100

Display Value

☒

Units

Enter Measurement Units

Tick Interval

10

Update Interval

15

second(s)

Range

90

100

+

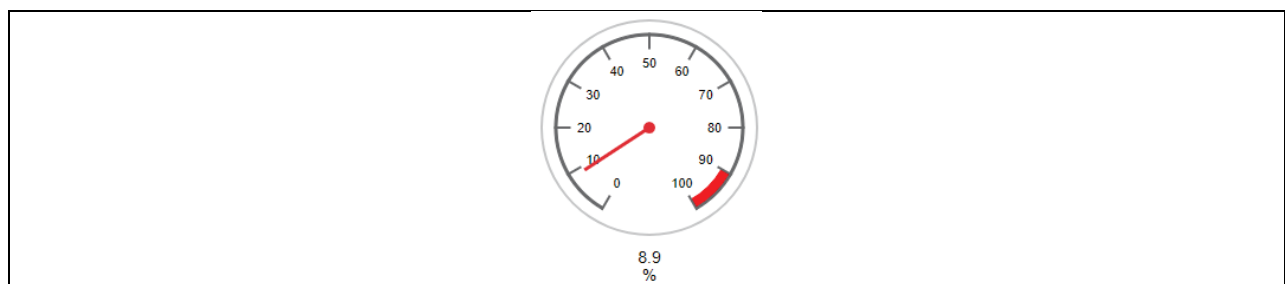
×

Create

Cancel

Εικόνα 7-32 Ιδιότητες του γραφήματος (Configure Gauge)

Στην οποία συμπληρώνουμε τις ιδιότητες του γραφήματος.



Εικόνα 7-33 Gauge

Chart With Multiple Series

Το άλλο είδος plugin που μπορούμε να χρησιμοποιήσουμε είναι το Chart With Multiple Series

Με την ίδια διαδικασία με το προηγούμενο επεξεργαζόμαστε τον κώδικα του javascript.

HTML

```

<!DOCTYPE html>
<html style="height: 100%;">
<head>
<!--
NOTE: This plugin will not be visible on public views of a channel.
If you intend to make your channel public, consider using the
MATLAB Visualization App to create your visualizations.
-->
<script type="text/javascript" src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
<script type="text/javascript" src="//thingspeak.com/highcharts-3.0.8.js"></script>
<script type="text/javascript" src="//thingspeak.com/exporting.js"></script>
%%PLUGIN_CSS%%
%%PLUGIN_JAVASCRIPT%%
</head>
    
```

```
<body>
<div id="chart-container">

</div>
</body>
</html>
```

Κώδικας 7-39 Κώδικας HTML

CSS

```
<style type="text/css">
body { background-color: white; height: 100%; margin: 0; padding: 0; }
#chart-container { width: 425px; height: 235px; display: block; position: absolute; bottom: 0; top: 0; left: 0; right: 0;
margin: 5px 15px 15px 0; overflow: hidden; }
</style>
```

Κώδικας 7-40 Κώδικας CSS

JAVASCRIPT

```
<script type="text/javascript">
var series_1_channel_id = 496902;
var series_1_field_number = 1;
var series_1_read_api_key = '74MLQ3N31PDFTE15';
var series_1_results = 10;
var series_1_color = '#d62020';
var series_2_channel_id = 496902;
var series_2_field_number = 2;
var series_2_read_api_key = '74MLQ3N31PDFTE15';
var series_2_results = 10;
var series_2_color = '#00aaff';
var series_3_channel_id = 496902;
var series_3_field_number = 3;
var series_3_read_api_key = '74MLQ3N31PDFTE15';
var series_3_results = 10;
var series_3_color = '#33cc33';
var chart_title = 'Temperature';
var y_axis_title = 'Celcius';
var my_offset = new Date().getTimezoneOffset();
var my_chart;
$(document).on('ready', function() {
addChart();
addSeries(series_1_channel_id, series_1_field_number, series_1_read_api_key, series_1_results, series_1_color);
addSeries(series_2_channel_id, series_2_field_number, series_2_read_api_key, series_2_results, series_2_color);
addSeries(series_3_channel_id, series_3_field_number, series_3_read_api_key, series_3_results, series_3_color);
});
function addChart() {
var localDate;
var chartOptions = {
chart: {
renderTo: 'chart-container',
defaultSeriesType: 'column',
backgroundColor: 'ffffff',
events: { }
},
title: { text: chart_title },
```

```

plotOptions: {
  series: {
    marker: { radius: 3 },
    animation: true,
    step: false,
    borderWidth: 0,
    turboThreshold: 0
  }
},
tooltip: {
  formatter: function() {
    var d = new Date(this.x + (my_offset*60000));
    var n = (this.point.name === undefined) ? '' : '<br>' + this.point.name;
    return this.series.name + ':<b>' + this.y + '</b>' + n + '<br>' + d.toDateString() + '<br>' +
    d.toTimeString().replace(/\(.*\)/, '');
  }
},
xAxis: {
  type: 'datetime',
  title: { text: 'Date' }
},
yAxis: { title: { text: y_axis_title } },
exporting: { enabled: false },
legend: { enabled: false },
credits: {
  text: 'ThingSpeak.com',
  href: 'https://thingspeak.com/',
  style: { color: '#D62020' }
}
};
my_chart = new Highcharts.Chart(chartOptions);
}

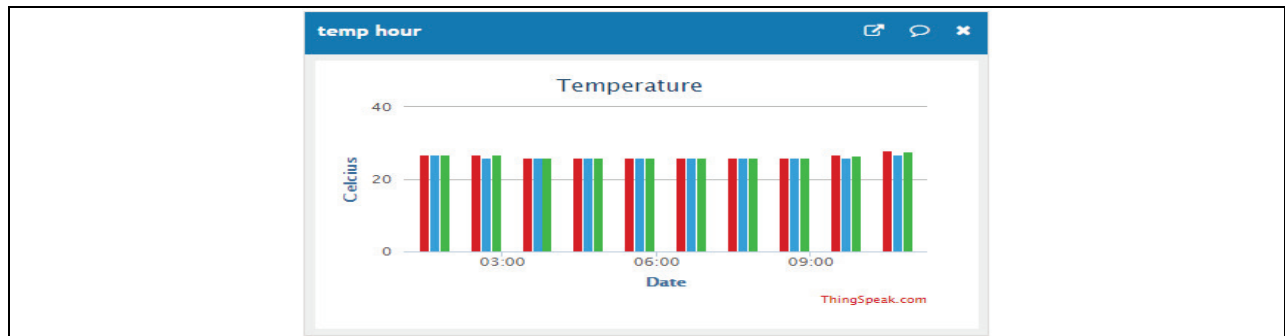
function addSeries(channel_id, field_number, api_key, results, color) {
  var field_name = 'field' + field_number;
  $.getJSON('https://api.thingspeak.com/channels/' + channel_id + '/fields/' + field_number +
  '.json?offset=0&round=2&results=' + results + '&api_key=' + api_key, function(data) {
    var chart_data = [];
    $.each(data.feeds, function() {
      var point = new Highcharts.Point();
      var value = this[field_name];
      point.x = getChartDate(this.created_at);
      point.y = parseFloat(value);
      if (this.location) { point.name = this.location; }
      if (!isNaN(parseInt(value))) { chart_data.push(point); }
    });
    my_chart.addSeries({ data: chart_data, name: data.channel[field_name], color: color });
  });
}

function getChartDate(d) {
  return Date.parse(d) - (my_offset * 60000);
}
</script>

```

Κώδικας 7-41 Κώδικας Javascript

Η παρακάτω εικόνα παρουσιάζει σε ένα γράφημα την ελάχιστη, μέγιστη και μέση τιμή της θερμοκρασίας για την προηγούμενη ώρα.



Εικόνα 7-34 Γράφημα την ελάχιστη, μέγιστη και μέση τιμή της θερμοκρασίας για την προηγούμενη ώρα

7.4.2 THINKSPEAK &-IFTTT

Η πλατφόρμα IFTTT της οποίας το πλήρες όνομα είναι "if this then that", είναι μια δικτυακή υπηρεσία που προκαλεί αλυσιδωτές αντιδράσεις τα λεγόμενα "applets". Η υπηρεσία μπορεί να εκτελεστεί από λογισμικό windows, android αλλά και ios. Το applet που έχουμε δημιουργήσει με τη βοήθεια της υπηρεσίας "webhooks" ελέγχει την θερμοκρασία του θερμοκηπίου μέσω της υπηρεσίας Thingspeak. Με βάση αυτή την πληροφορία μας στέλνει ένα email στην διεύθυνση που θα του ορίσουμε. Το περιεχόμενό του αποτελείται από την τιμή της θερμοκρασίας εκείνη τη στιγμή, ένα μήνυμα (παραμετροποιήσιμο) το οποίο μας γράφει "Comfortable" αν η θερμοκρασία είναι κάτω των 30 βαθμών κελσίου και "Uncomfortable" αν η θερμοκρασία είναι άνω των 30 βαθμών κελσίου.

Αυτό υλοποιείται φτιάχνοντας ένα καινούργιο Matlab analysis, με τον παρακάτω κώδικα.

```
channelID=493691;
iftttURL='https://maker.ifttt.com/trigger/trigger/with/key/yKZwJFa2iFcLAGtXXIIMd';
tempData=thingSpeakRead(channelID,'NumDays',30,'Fields',1);
maxValue= 30 ;
url=strcat('https://api.thingspeak.com/channels/',string(493691),'fields/3/last.txt');
lastValue=str2num(webread(url));
if (lastValue<maxValue)
    Message='Comfortable';
    webwrite(iftttURL,'value1',lastValue,'value2',Message);
end
if (lastValue>=maxValue)
    Message='Uncomfortable';
    webwrite(iftttURL,'value1',lastValue,'value2',Message);
end
```

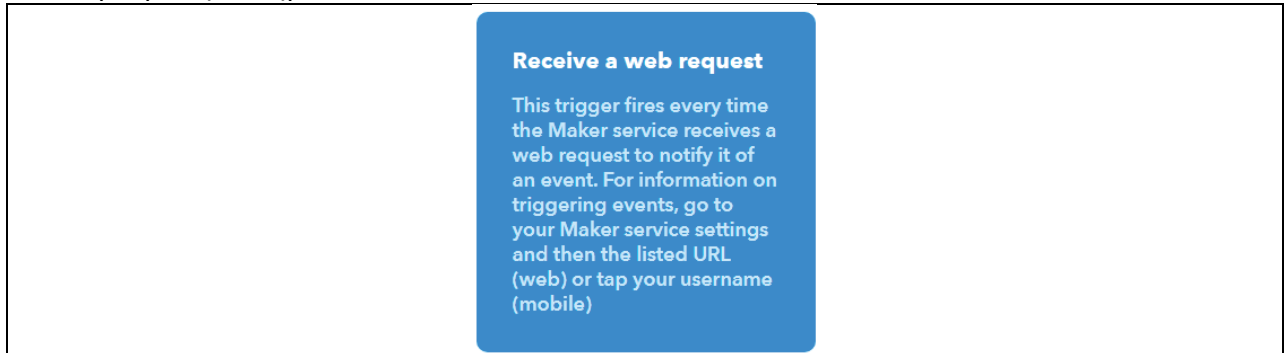
Κώδικας 7-42 Κώδικας ThinkSpeak & IFTTT

Ρυθμίζουμε τα στοιχεία του καναλιού (channel id ,field) που θέλουμε να ελέγχουμε. Με δύο απλά if ελέγχουμε αν η θερμοκρασία είναι κάτω ή πάνω από 30 βαθμούς κελσίου. Στις γραμμές 9,14 ρυθμίζουμε το μήνυμα που θέλουμε να φαίνεται στο email μας. Για το κλειδί που έχουμε προσθέσει στην δεύτερη γραμμή θα μιλήσουμε παρακάτω.

Στην υπηρεσία IFTTT αφού φτιάξουμε λογαριασμό διαλέγουμε New Applet

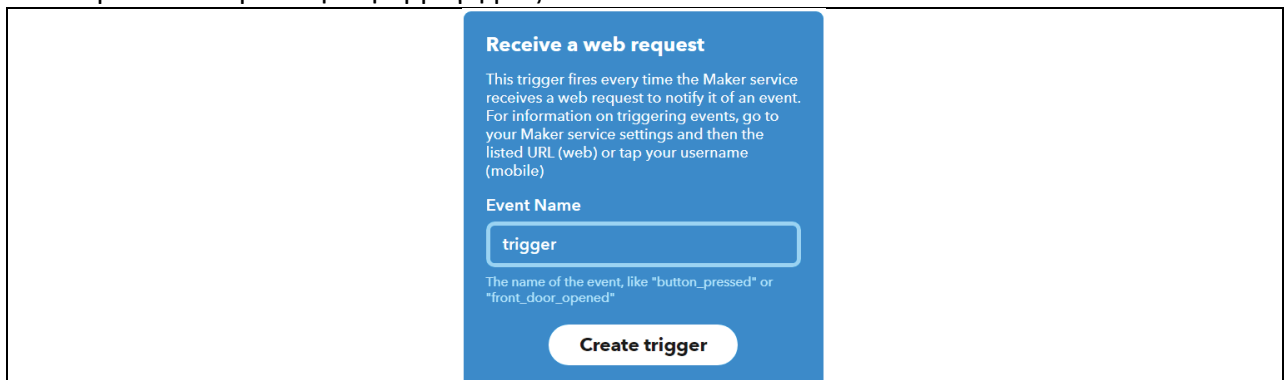
Έπειτα πατάμε **if + this**

Διαλέγουμε την υπηρεσία webhooks



Εικόνα 7-35 IFTTT 1

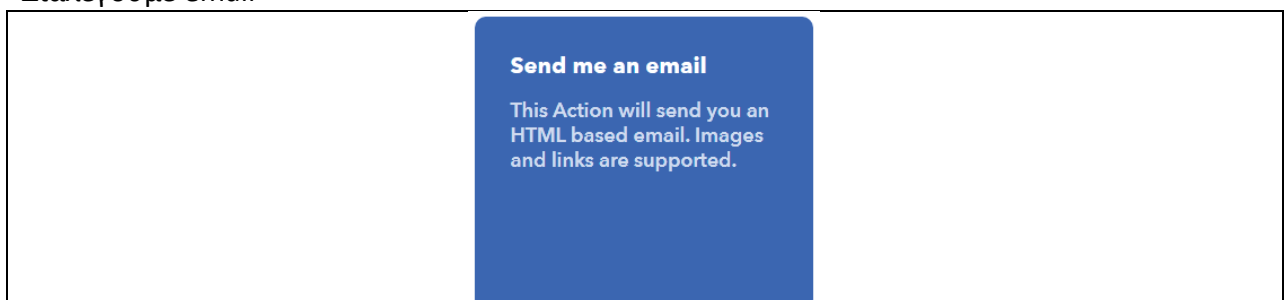
Δίνουμε ένα όνομα στην εφαρμογή μας



Εικόνα 7-36 IFTTT 2

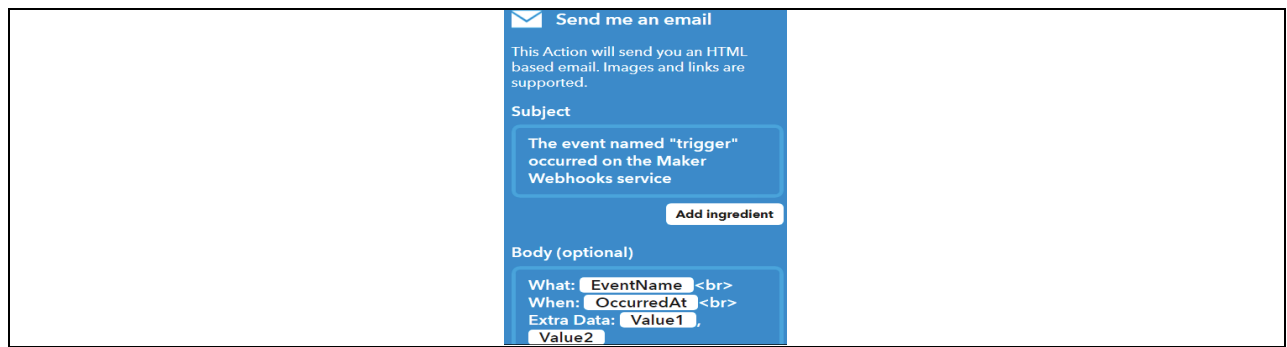
Πατάμε **+ that**

Διαλέγουμε email



Εικόνα 7-37 IFTTT 3

Και ρυθμίζουμε τις πληροφορίες του μηνύματος.



Send me an email

This Action will send you an HTML based email. Images and links are supported.

Subject

The event named "trigger" occurred on the Maker Webhooks service

Body (optional)

What: EventName
 When: OccurredAt
 Extra Data: Value1, Value2

Add ingredient

Εικόνα 7-38 IFTTT 4

Τέλος Create action και ο αυτοματισμός μας είναι έτοιμος.

Για να βρούμε το key της εφαρμογής μας πρέπει να πατήσουμε στην υπηρεσία webhooks→documentation και ανακατευθυνόμαστε στην εξής σελίδα .



Your key is:

[Back to service](#)

To trigger an Event

Make a POST or GET web request to:

`https://maker.ifttt.com/trigger/{event}/with/key/yKZwJFa2iFcLAGtXXIIMd`

With an optional JSON body of:

`{ "value1" : "", "value2" : "", "value3" : "" }`

The data is completely optional, and you can also pass value1, value2, and value3 as query parameters or form variables. This content will be passed on to the Action in your Recipe.

You can also try it with `curl` from a command line.

`curl -X POST https://maker.ifttt.com/trigger/{event}/with/key/yKZwJFa2iFcLAGtXXIIMd`

Test It

Εικόνα 7-39 IFTTT Key

Όπου βρίσκουμε και το key που μας έχει παρέχει η υπηρεσία.

Τέλος για να έχουμε συνεχή ενημέρωση για την τιμή του αισθητήρα μας δημιουργήσαμε μια εφαρμογή time control στο thingspeak με τον τρόπο που εξηγήσαμε παραπάνω, με σκοπό να εκτελείτε ο αυτοματισμός ανά κάποιο χρονικό διάστημα που επιθυμούμε.

7.5 ΟΛΟΚΛΗΡΩΜΕΝΟ ΣΕΝΑΡΙΟ – ΚΑΤΑΣΤΑΣΗ/ΑΥΤΟΜΑΤΑ ΣΕΝΑΡΙΑ ΜΕ THINKSPEAK/IFTTT

7.5.1 THINKSPEAK

Οπότε εφαρμόζοντας όλα τα παραπάνω για το Σενάριο αυτοματισμού κήπου/θερμοκηπίου έχουμε τα παρακάτω γραφήματα και κώδικες.

ThinkSpeak.php

Υπηρεσία Thingspeak

Η επεξεργασία των μετρήσεων, η μετάδοσή τους στο cloud ή στη τοπική βάση, η σχεδίαση των διαγραμμάτων και η παρουσίαση τους έχει γίνει από τον Δρ. Τοπάλη Ευάγγελο σε συνεργασία με τον φοιτητή Συμεωνίδη Οδυσσέα.

Η υπηρεσία Thingspeak είναι μια IOT(internet of things) εφαρμογή που αποθηκεύει και επεξεργάζεται δεδομένα χρησιμοποιώντας το πρωτόκολλο HTTP μέσω ασύρματης ή ενσύρματης σύνδεσης στο διαδίκτυο. Η πλατφόρμα υποστηρίζει κώδικα Matlab γεγονός που την καθιστά πολύ εύχρηστη ειδικά στους νέους χρήστες οι οποίοι θέλουν να απεικονίσουν εύκολα και γρήγορα τα δεδομένα που ανεβάζουν. Με μία απλή διαδικασία εγγραφής(δωρεάν για συγκεκριμένες δυνατότητες) και την δημιουργία καναλιών όπως τα αποκαλεί η υπηρεσία, μπορεί να πραγματοποιηθεί ανέβασμα τιμών από αισθητήρες που επιθυμούμε. Στην περίπτωση μας για να συνδέσουμε το arduino με την πλατφόρμα χρησιμοποιήθηκε η επίσημη βιβλιοθήκη που παρέχει η εταιρία "ThingSpeak.h". Η βιβλιοθήκη αυτή σε συνδυασμό με ένα απλό ethernet shield για την σύνδεση του arduino με το διαδίκτυο κατέστησαν δυνατή την μεταφορά των τιμών των αισθητήρων στο cloud της υπηρεσίας αλλά και την γραφική τους απεικόνιση. Παρακάτω παρουσιάζονται τα γραφήματα των έξι αισθητήρων που υπάρχουν στην μακέτα του θερμοκηπίου και ανανεώνονται δυναμικά ανά σαράνταπέντε λεπτά.

[Στάθμη Νερού](#) |
 [Φως](#) |
 [Θερμοκρασία](#) |
 [Υγρασία](#) |
 [Υγρασία Εδάφους](#) |
 [Ποσοστό Βροχής](#)

Εικόνα 7-40 ThinkSpeak Ιστοσελίδα

```

<html>
<head>
  <meta charset="UTF-8">
  <title>ThingSpeak</title>
  <link rel="stylesheet" href="../css/style.css" type="text/css">
</head>
<body>
  <div id="header"> </div>
  <div id="body">
    <div class="content">
<p></p>

```

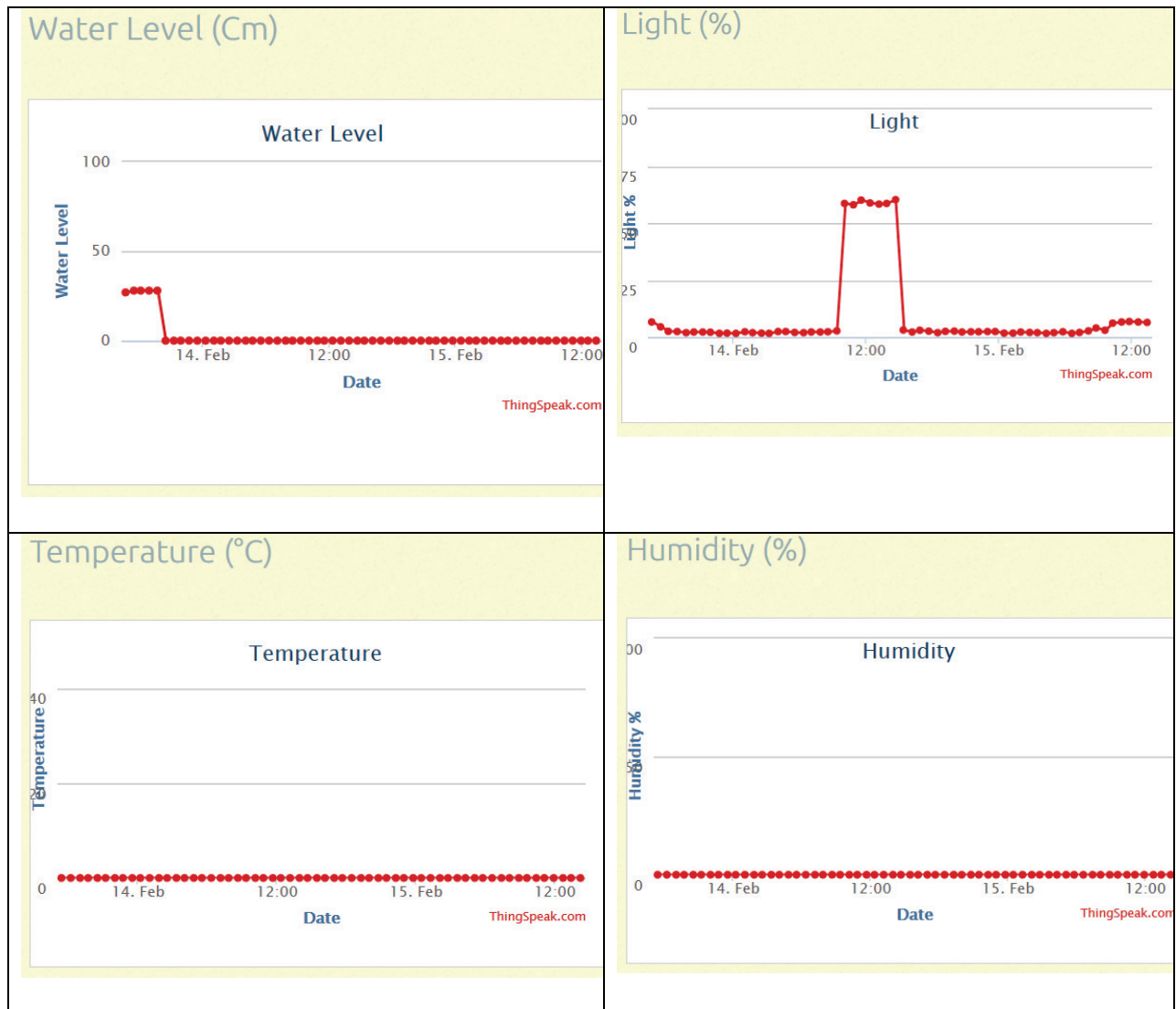
Υπηρεσία Thingspeak

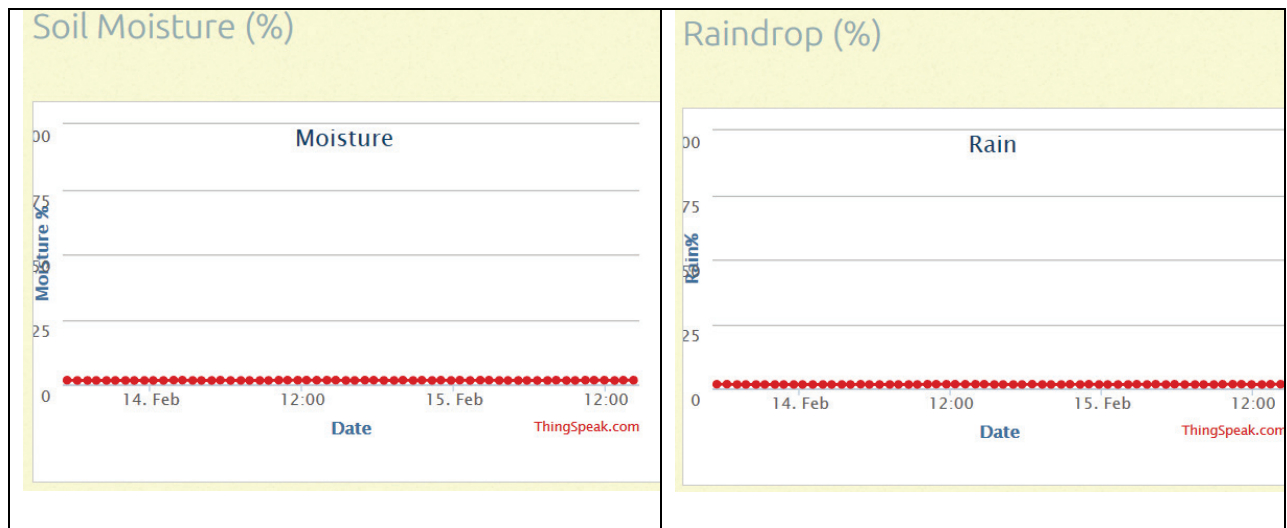
<p>Η υπηρεσία Thingspeak είναι μια IOT(internet of things) εφαρμογή που αποθηκεύει και επεξεργάζεται δεδομένα χρησιμοποιώντας το πρωτόκολλο HTTP μέσω ασύρματης ή ενσύρματης σύνδεσης στο διαδίκτυο. Η πλατφόρμα υποστηρίζει κώδικα Matlab γεγονός που την καθιστά πολύ εύχρηστη ειδικά στους νέους χρήστες οι οποίοι θέλουν να απεικονίσουν εύκολα και γρήγορα τα δεδομένα που ανεβάζουν. Με μία απλή διαδικασία εγγραφής(δωρεάν για συγκεκριμένες δυνατότητες) και την δημιουργία καναλιών όπως τα αποκαλεί η υπηρεσία, μπορεί να πραγματοποιηθεί ανέβασμα τιμών από αισθητήρες που επιθυμούμε. Στην περίπτωση μας για να συνδέσουμε το arduino με την πλατφόρμα χρησιμοποιήθηκε η επίσημη βιβλιοθήκη που παρέχει η εταιρία "ThingSpeak.h". Η βιβλιοθήκη αυτή σε συνδυασμό με ένα απλό

ethernet shield για την σύνδεση του arduino με το διαδίκτυο κατέστησαν δυνατή την μεταφορά των τιμών των αισθητήρων στο cloud της υπηρεσίας αλλά και την γραφική τους απεικόνιση. Παρακάτω παρουσιάζονται τα γραφήματα των έξι αισθητήρων και ανανεώνονται δυναμικά ανά σαρανταπέντε λεπτά.</p>
<div style="width:100%">
</div>
<script type="text/javascript">
// <![CDATA[
function change(id) {
var i;
for (i=1; i<=6; i++){
if (i != id) {
document.getElementById('showhide'+i).style.display = "none";
}
}
if (document.getElementById('showhide'+id).style.display == "none") {
document.getElementById('showhide'+id).style.display = "inline";
//document.getElementById("thetext"+id).innerHTML = "Απόκρυψη απάντησης";
}
else {
document.getElementById('showhide'+id).style.display = "none";
//document.getElementById("thetext"+id).innerHTML = "Εμφάνιση απάντησης";
}
}
//]]>
</script>
Στάθμη Νερού | Φως | Θερμοκρασία | Υγρασία | Υγρασία Εδάφους | Ποσοστό Βροχής
</p>
<div id="showhide1" style="display:none">
<h2>Water Level (cm)</h2>
<p align:center style="text-align:justify"><?php require_once('water.php'); ?>
</p>
</div>
<div id="showhide2" style="display:none">
<h2>Light (%)</h2>
<p style="text-align:justify"><?php require_once('light.php'); ?>
</p>
</div>
<div id="showhide3" style="display:none">
<h2>Temperature (°C)</h2>
<p style="text-align:justify"><?php require_once('temp.php'); ?>
</p>
</div>
<div id="showhide4" style="display:none">
<h2>Humidity (%)</h2>
<p style="text-align:justify"><?php require_once('humidity.php'); ?>
</p>
</div>
<div id="showhide5" style="display:none">
<h2>Soil Moisture (%)</h2>
<p style="text-align:justify"><?php require_once('moisture.php'); ?>
</p>
</div>

```
</div>
<div id="showhide6" style="display:none">
<h2>Raindrop (%)</h2>
<p style="text-align:justify"><?php require_once('rain.php'); ?>
</p>
</div>
</div>
</div>
</body>
</html>
```

Κώδικας 7-43 Κώδικας ThinkSpeak Ιστοσελίδα





Εικόνα 7-41 ThinkSpeak Ιστοσελίδες

water.php

```
<iframe width="450" height="280" style="border: 1px solid #cccccc;"  
src="https://thingspeak.com/channels/493691/charts/1?bgcolor=%23ffffff&color=%23d62020&dynamic=true&results=60&title=Water Level &type=line&+%35&yaxismax=100&yaxismin=0"></iframe>
```

light.php

```
<iframe width="450" height="260" style="border: 1px solid #cccccc;" src="  
https://thingspeak.com/channels/493691/charts/2?bgcolor=%23ffffff&color=%23d62020&dynamic=true&results=60&title=Light&type=line&yaxis=Light+%25&yaxismax=100&yaxismin=0"></iframe>
```

temp.php

```
<iframe width="450" height="260" style="border: 1px solid #cccccc;"  
src="https://thingspeak.com/channels/493691/charts/3?bgcolor=%23ffffff&color=%23d62020&dynamic=true&results=60&title=Temperature&type=line&yaxis=Temperature&yaxismax=50&yaxismin=0"></iframe>
```

Humidity.php

```
<iframe width="450" height="260" style="border: 1px solid #cccccc;"  
src="https://thingspeak.com/channels/493691/charts/4?bgcolor=%23ffffff&color=%23d62020&dynamic=true&results=60&title=Humidity&type=line&yaxis=Humidity+%25&yaxismax=100&yaxismin=0"></iframe>
```

Soil Moisture.php

```
<iframe width="450" height="260" style="border: 1px solid #cccccc;"  
src="https://thingspeak.com/channels/493691/charts/5?bgcolor=%23ffffff&color=%23d62020&dynamic=true&results=60&title=Moisture&type=line&yaxis=Moisture+%25&yaxismax=100&yaxismin=0"></iframe>
```

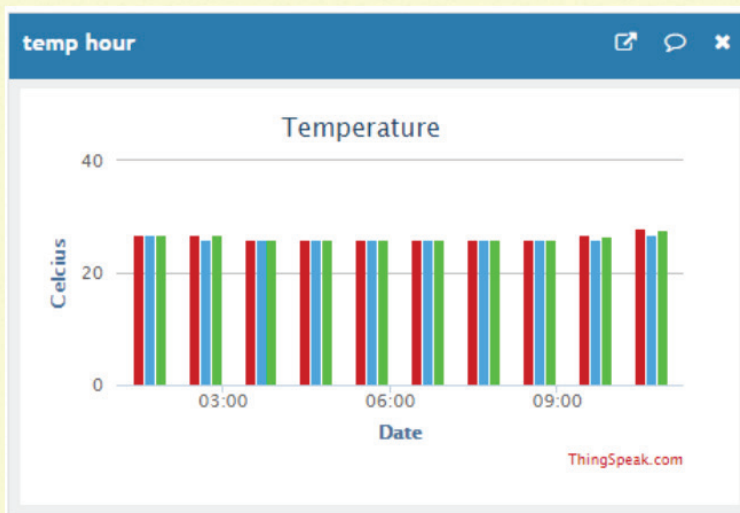
rain.php

```
<iframe width="450" height="260" style="border: 1px solid #cccccc;"  
src="https://thingspeak.com/channels/493691/charts/6?bgcolor=%23ffffff&color=%23d62020&dynamic=true&results=60&title=Rain&type=line&yaxis=Rain+%25&yaxismax=100&yaxismin=0"></iframe>
```

Κώδικας 7-44 Κώδικας ThinkSpeak Ιστοσελίδες

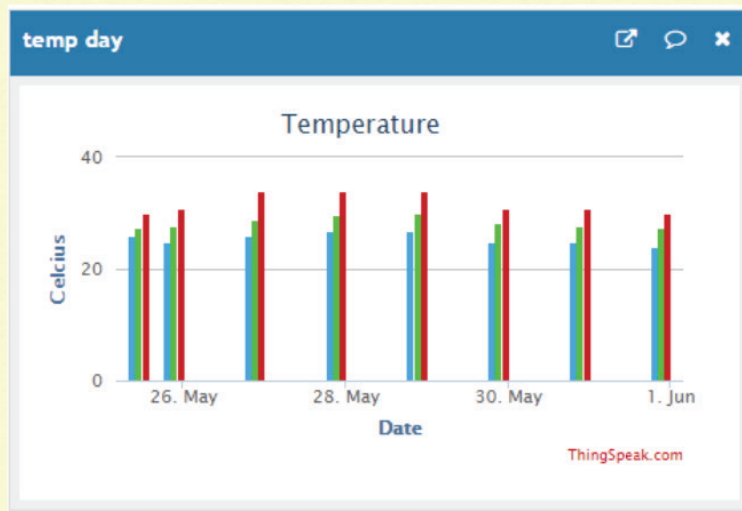
Παρακάτω παρουσιάζονται τα διαγράμματα για την θερμοκρασία και την υγρασία από την υπηρεσία Computing 'Thingspeak'. Τα διαγράμματα είναι μόνο για προσωπική χρήση από την υπηρεσία Thingspeak για αυτό και έχουν συμπεριληφθεί κάποια ενδεικτικά Screenshots. Για την δημιουργία των διαγραμμάτων χρησιμοποιήθηκε κώδικας Matlab αλλά και μία εφαρμογή της υπηρεσίας (time control) η οποία μας δίνει την δυνατότητα να εκτελούμε αυτόματα των κώδικά μας ανά οποιοδήποτε χρονικό διάστημα επιθυμούμε. Το κόκκινο χρώμα αντιστοιχεί στην υψηλότερη τιμή, το μπλε στην χαμηλότερη και το πράσινο στην μέση τιμή.

Στο παρακάτω διάγραμμα παρατηρούμε την ωριαία μεταβολή της θερμοκρασίας και τη μέγιστη, ελάχιστη και μέση τιμή της.

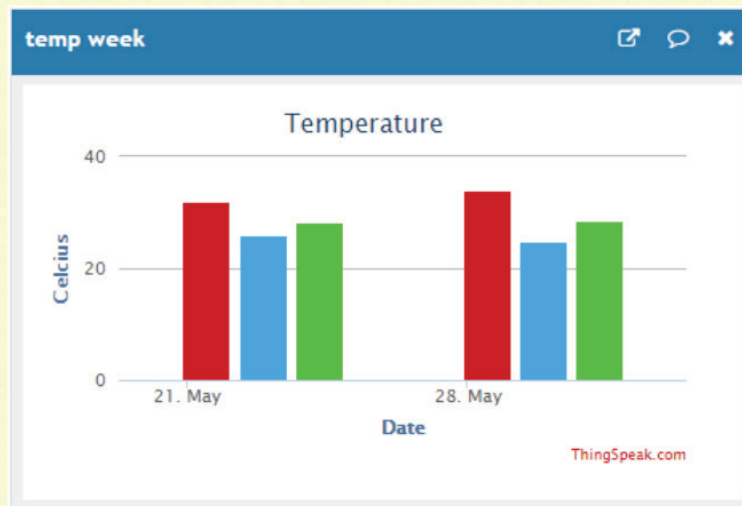


Εικόνα 7-42 ThinkSpeak Ιστοσελίδες

Στο παρακάτω διάγραμμα παρατηρούμε την ημερήσια μεταβολή της θερμοκρασίας και τη μέγιστη,ελάχιστη και μέση τιμή της.

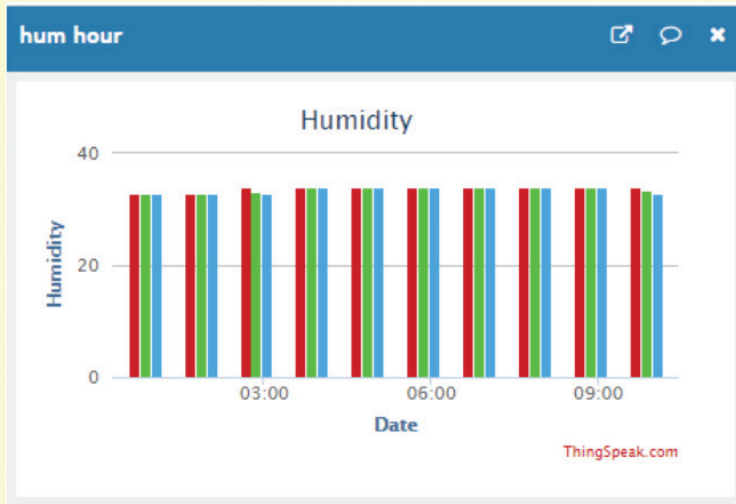


Στο παρακάτω διάγραμμα παρατηρούμε την εβδομαδιαία μεταβολή της θερμοκρασίας και τη μέγιστη,ελάχιστη και μέση τιμή της.

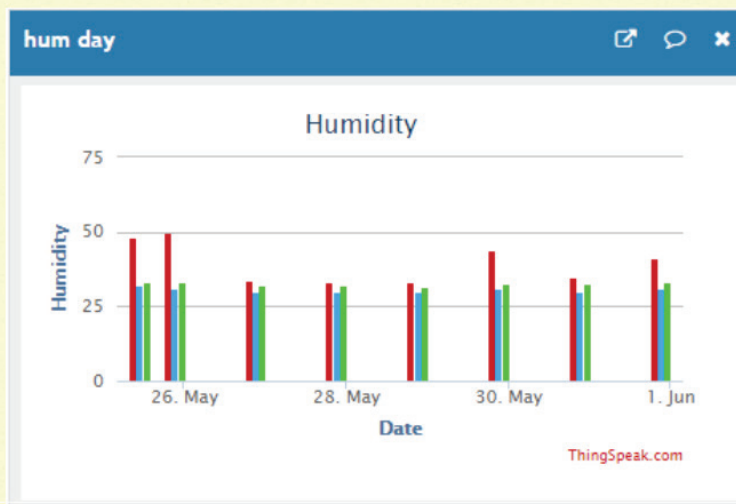


Εικόνα 7-43 ThinkSpeak Ιστοσελίδες

Στο παρακάτω διάγραμμα παρατηρούμε την ωριαία μεταβολή της υγρασίας και τη μέγιστη,ελάχιστη και μέση τιμή της.

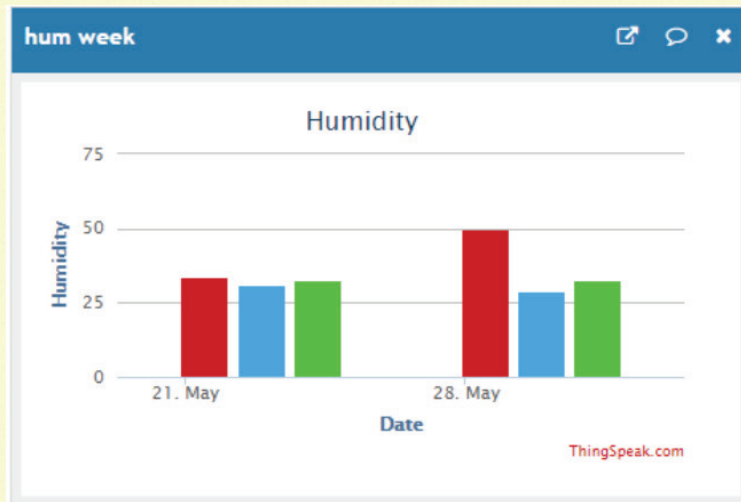


Στο παρακάτω διάγραμμα παρατηρούμε την ημερήσια μεταβολή της υγρασίας και τη μέγιστη,ελάχιστη και μέση τιμή της.



Εικόνα 7-44 ThinkSpeak Ιστοσελίδες

Στο παρακάτω διάγραμμα παρατηρούμε την εβδομαδιαία μεταβολή της υγρασίας και τη μέγιστη,ελάχιστη και μέση τιμή της.



Εικόνα 7-45 ThinkSpeak Ιστοσελίδες

```
<html>
<head>
  <meta charset="UTF-8">
  <title>Multiple Series Charts</title>
  <link rel="stylesheet" href="../../css/style.css" type="text/css">

  <meta http-equiv="refresh" content="86400" />
  <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
  <script type="text/javascript">
    google.charts.load('current', {'packages':['corechart']});
    google.charts.setOnLoadCallback(drawChart);

    function drawChart() {

      var data = new google.visualization.DataTable();
      data.addColumn('datetime', 'event');
      data.addColumn('number', 'Average light');
      data.addColumn('number', 'Max light');
      data.addColumn('number', 'Min light');
      data.addColumn('number', 'Average raindrop');
      data.addColumn('number', 'Max raindrop');
      data.addColumn('number', 'Min raindrop');
      data.addColumn('number', 'Average moisture');
      data.addColumn('number', 'Max moisture');
      data.addColumn('number', 'Min moisture');
      data.addColumn('number', 'Average temperature');
      data.addColumn('number', 'Max temperature');
      data.addColumn('number', 'Min temperature');
      data.addColumn('number', 'Average humidity');
      data.addColumn('number', 'Max humidity');
      data.addColumn('number', 'Min humidity');
      data.addColumn('number', 'Average tank');
      data.addColumn('number', 'Max tank');
```

```

data.addColumn('number', 'Min tank');

<?php
// Prepare variables for database connection

$dbusername = "root"; // enter database username, I used "arduino" in step 2.2
$dbpassword = "....."; // enter database password, I used "arduinotest" in step 2.2
$server = "localhost";

// Connect to your database
$mysqli = new mysqli($server, $dbusername, $dbpassword, myhome");
if($mysqli->connect_error)
    die('Connect Error (' . mysqli_connect_errno() . ') ' . mysqli_connect_error());

// Execute SQL statement ta grouparw ana imera kai gia kathe mera to min kai to max kai to average kai krataw
tis teleytaies 7 meres
$result = $mysqli->query("SELECT YEAR(event) AS year, MONTH(event) AS month, DAY(event) AS day,
AVG(light),MAX(light),MIN(light), AVG(raindrop),MAX(raindrop),MIN(raindrop),
AVG(moisture),MAX(moisture),MIN(moisture), AVG(temperature),MAX(temperature),MIN(temperature),
AVG(humidity),MAX(humidity),MIN(humidity), AVG(tank),MAX(tank),MIN(tank) from sensors group by DAY(event)
ORDER BY event DESC limit 7");

if ($result->num_rows > 0) {

// output data of each row
while($row = $result->fetch_assoc()) {
    echo "data.addRow([new Date(".$row["year"].",".$row["month"]."-1,".$row["day"]."),
".$row["AVG(light)"].",".$row["MAX(light)"].",".$row["MIN(light)"].",".$row["AVG(raindrop)"].",".$row["MAX(raindrop)"].",".$row["MIN(raindrop)"].",".$row["AVG(moisture)"].",".$row["MAX(moisture)"].",".$row["MIN(moisture)"]
].",".$row["AVG(temperature)"].",".$row["MAX(temperature)"].",".$row["MIN(temperature)"].",".$row["AVG(humidi-
ty)"].",".$row["MAX(humidity)"].",".$row["MIN(humidity)"].",".$row["AVG(tank)"].",".$row["MAX(tank)"].",".$row["
MIN(tank)"]."]);");
}
}
else {
}

?>

var light_options = {
    title: 'Darkness (%) Last Week',
    subtitle: 'measured in %',
    hAxis: { format: 'MMM dd, yyyy' },
    seriesType: 'bars',
    width: 600,
    height: 300
};

var raindrop_options = {
    title: 'Dryness-Raindrop (%) Last Week',
    subtitle: 'measured in %',
    hAxis: { format: 'MMM dd, yyyy' },
    seriesType: 'bars',
    width: 600,
    height: 300
};

var moisture_options = {

```



```

    title: 'Dryness-Moisture (%) Last Week',
    subtitle: 'measured in %',
    hAxis: { format: 'MMM dd, yyyy' },
    seriesType: 'bars',
    width: 600,
    height: 300
};

var temperature_options = {
    title: 'Temperature (C) Last Week',
    subtitle: 'measured in C',
    hAxis: { format: 'MMM dd, yyyy' },
    seriesType: 'bars',
    width: 600,
    height: 300
};

var humidity_options = {
    title: 'Humidity (%) Last Week',
    subtitle: 'measured in %',
    hAxis: { format: 'MMM dd, yyyy' },
    seriesType: 'bars',
    width: 600,
    height: 300
};

var tank_options = {
    title: 'Tank (cm) Last Week',
    subtitle: 'measured in cm',
    hAxis: { format: 'MMM dd, yyyy' },
    seriesType: 'bars',
    width: 600,
    height: 300
};

var light_chart = new google.visualization.ComboChart(document.getElementById('light_div'));

var light_view = new google.visualization.DataView(data);
light_view.setColumns([0,1,2,3]);
light_chart.draw(light_view, light_options);

//-----
var raindrop_chart = new google.visualization.ComboChart(document.getElementById('raindrop_div'));

var raindrop_view = new google.visualization.DataView(data);
raindrop_view.setColumns([0,4,5,6]);
raindrop_chart.draw(raindrop_view, raindrop_options);

//-----
var moisture_chart = new google.visualization.ComboChart(document.getElementById('moisture_div'));

var moisture_view = new google.visualization.DataView(data);
moisture_view.setColumns([0,7,8,9]);
moisture_chart.draw(moisture_view, moisture_options);

//-----
var temperature_chart = new google.visualization.ComboChart(document.getElementById('temperature_div'));

var temperature_view = new google.visualization.DataView(data);
temperature_view.setColumns([0,10,11,12]);
temperature_chart.draw(temperature_view, temperature_options);

```

```
//-----
var humidity_chart = new google.visualization.ComboChart(document.getElementById('humidity_div'));

var humidity_view = new google.visualization.DataView(data);
humidity_view.setColumns([0,13,14,15]);
humidity_chart.draw(humidity_view, humidity_options);
var tank_chart = new google.visualization.ComboChart(document.getElementById('tank_div'));

var tank_view = new google.visualization.DataView(data);
tank_view.setColumns([0,16,17,18]);
tank_chart.draw(tank_view, tank_options);
}
</script>
</head>
<body>
  <div id="header">
    <p>Παρακάτω παρουσιάζονται τα διαγράμματα για την θερμοκρασία και την υγρασία από την υπηρεσία
    Computing 'Thingspeak'. Τα διαγράμματα είναι μόνο για προσωπική χρήση από την υπηρεσία Thingspeak για αυ-
    τό και έχουν συμπεριληφθεί κάποια ενδεικτικά Screenshots. Για την δημιουργία των διαγραμμάτων χρησιμοποιή-
    θηκε κώδικας Matlab αλλά και μία εφαρμογή της υπηρεσίας (time control) η οποία μας δίνει την δυνατότητα να
    εκτελούμε αυτόματα των κώδικά μας ανά οποιοδήποτε χρονικό διάστημα επιθυμούμε. Το κόκκινο χρώμα αντι-
    στοιχεί στην υψηλότερη τιμή ,το μπλε στην χαμηλότερη και το πράσινο στην μέση τιμή .</p>

    <body>
      <div style="width:100%"> </div>
      <p>Στο παρακάτω διάγραμμα παρατηρούμε την ωριαία μεταβολή της θερμοκρασίας και τη μέγι-
      στη,ελάχιστη και μέση τιμή της. </p>
      
      <p>Στο παρακάτω διάγραμμα παρατηρούμε την ημερήσια μεταβολή της θερμοκρασίας και τη μέγι-
      στη,ελάχιστη και μέση τιμή της. </p>
      
      <p>Στο παρακάτω διάγραμμα παρατηρούμε την εβδομαδιαία μεταβολή της θερμοκρασίας και τη μέγι-
      στη,ελάχιστη και μέση τιμή της. </p>
      
      <p>Στο παρακάτω διάγραμμα παρατηρούμε την ωριαία μεταβολή της υγρασίας και τη μέγιστη,ελάχιστη
      και μέση τιμή της. </p>
      
      <p>Στο παρακάτω διάγραμμα παρατηρούμε την ημερήσια μεταβολή της υγρασίας και τη μέγι-
      στη,ελάχιστη και μέση τιμή της. </p>
      
      <p>Στο παρακάτω διάγραμμα παρατηρούμε την εβδομαδιαία μεταβολή της υγρασίας και τη μέγι-
      στη,ελάχιστη και μέση τιμή της. </p>
      </div>
    </body>
  </html>

</body>
</html>
```

7.5.2 THINKSPEAK - IFTTT

Οπότε εφαρμόζοντας όλα τα παραπάνω για το Σενάριο αυτοματισμού κήπου/θερμοκηπίου έχουμε τα παρακάτω γραφήματα και κώδικες.

Ifft.php

Πλατφόρμα IFTTT

Η πλατφόρμα IFTTT της οποίας το πλήρες όνομα είναι "if this then that", είναι μια δικτυακή υπηρεσία που προκαλεί αλυσιδωτές αντιδράσεις τα λεγόμενα "applets". Η υπηρεσία μπορεί να εκτελεστεί από λογισμικό windows, android αλλά και ios. Το applet που έχουμε δημιουργήσει με τη βοήθεια της υπηρεσίας "webhooks" ελέγχει την θερμοκρασία του θερμοκηπίου μέσω της υπηρεσίας Thingspeak. Με βάση αυτή την πληροφορία μας στέλνει ένα email στην διεύθυνση που θα του ορίσουμε. Το περιεχόμενό του αποτελείται από την τιμή της θερμοκρασίας εκείνη τη στιγμή, ένα μήνυμα (παραμετροποιήσιμο) το οποίο μας γράφει "Comfortable" αν η θερμοκρασία είναι κάτω των 30 βαθμών κελσίου και "Uncomfortable" αν η θερμοκρασία είναι άνω των 30 βαθμών κελσίου.

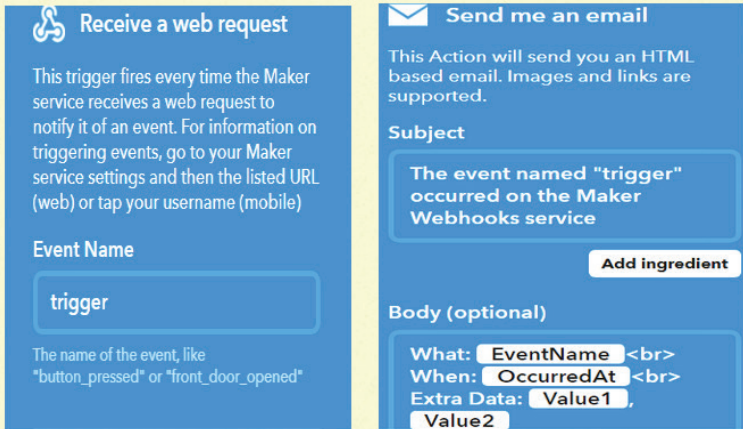
Κώδικας matlab στην υπηρεσία Thingspeak

MATLAB Code

```
1 channelID=493691;
2 iftttURL='https://maker.ifttt.com/trigger/trigger/with/key/yKZwJFa2iFcLAGtXXlMd';
3 tempData=thingSpeakRead(channelID,'NumDays',30,'Fields',1);
4 maxValue= 30 ;
5 url=strcat('https://api.thingspeak.com/channels/',string(493691),'fields/3/last.txt');
6 lastValue=str2num(webread(url));
7
8 if (lastValue<maxValue)
9     Message='Comfortable';
10    webwrite(iftttURL,'value1',lastValue,'value2',Message);
11 end
12
13 if (lastValue>=maxValue)
14     Message='Uncomfortable';
15    webwrite(iftttURL,'value1',lastValue,'value2',Message);
16 end
```

Εικόνα 7-46 ThinkSpeak-IFTTT Ιστοσελίδες

Στο λογαριασμό μας στο Iftt θα έχουμε την παρακάτω εικόνα



Η εικόνα που θα έρθει στο email αν η θερμοκρασία είναι κάτω από 30 βαθμούς κελσίου θα είναι η εξής:

```
What: trigger
When: June 4, 2018 at 03:31PM
Extra Data: 28, Comfortable
```

Στην άλλη περίπτωση η εικόνα που θα έρθει στο email αν η θερμοκρασία είναι ίση ή μεγαλύτερη από 30 βαθμούς κελσίου θα είναι η εξής:

```
What: trigger
When: June 4, 2018 at 10:36PM
Extra Data: 31, Uncomfortable
```

Εικόνα 7-47 ThinkSpeak-IFTTT Ιστοσελίδες

```
<html>
<head>
  <meta charset="UTF-8">
  <title>IFTTT</title>
  <link rel="stylesheet" href="../css/style.css" type="text/css">
</head>
<body>
  <div id="header"><strong>Πλατφόρμα IFTTT</strong><br>
</div>
  <div class="content">
    <p> Η πλατφόρμα IFTTT της οποίας το πλήρες όνομα είναι "if this then that",είναι μια δικτυακή υπηρεσία που προκαλεί αλυσιδωτές αντιδράσεις τα λεγόμενα "applets". Η υπηρεσία μπορεί να εκτελεστεί από λογισμικό windows, android αλλά και ios .Το applet που έχουμε δημιουργήσει με τη βοήθεια της υπηρεσίας "webhooks" ελέγχει την θερμοκρασία του θερμοκηπίου μέσω της υπηρεσίας Thingspeak. Με βάση αυτή την πληροφορία μας στέλνει ένα email στην διεύθυνση που θα του ορίσουμε. Το περιεχόμενό του αποτελείται από την τιμή της θερμοκρασίας εκείνη τη στιγμή, ένα μήνυμα(παραμετροποίηση) το οποίο μας γράφει "Comfortable" αν η θερμοκρασία είναι κάτω των 30 βαθμών κελσίου και "Uncomfortable" αν η θερμοκρασία είναι άνω των 30 βαθμών κελσίου.
    <p><strong>Κώδικας matlab στην υπηρεσία Thingspeak</strong></p>
  </div>
</body>
</html>
```

```


<p>Στο λογαριασμό μας στο Ifft θα έχουμε την παρακάτω εικόνα </p>
<table width="465" height="287" style="border-spacing: 15px">
  <tr>
    <td width="209" style="border: 1px transparent">
      
    </td>

    <td style="border: 1px solid transparent">
      
    </td>
  </tr>
</table>
<p>Η εικόνα που θα έρθει στο email αν η θερμοκρασία είναι κάτω από 30 βαθμούς κελσίου θα είναι η ε-
ξής:</p>

<p>Στην άλλη περίπτωση η εικόνα που θα έρθει στο email αν η θερμοκρασία είναι ίση ή μεγαλύτερη από
30 βαθμούς κελσίου θα είναι η εξής:</p>


<body>
<div style="width:100%">

</div>

</div>
<div id="footer">
<div> </div>
</div>
</body>
</html>

```

Κώδικας 7-46 Κώδικας ThinkSpeak Ιστοσελίδες

ΚΕΦΑΛΑΙΟ 8 ΒΙΒΛΙΟΓΡΑΦΙΑ

8.1 ΕΛΛΗΝΟΓΛΩΣΣΗ

- Αδαμόπουλος, Δ. (2008). *Τηλεματική, Τηλεματική Διαδίκτυα και Κοινωνία* (Β' εκδ., Τόμ. Α'). Πάτρα: ΕΑΠ.
- Αλεξίου, Γ. (2001). *Μικροεπεξεργαστές, Ψηφιακά Συστήματα* (Τόμ. Γ'). Πάτρα: ΕΑΠ.
- Εμβάλωτης, Α. (2002). *Υπολογιστές & Κοινωνία, Information Society, Τηλεματική Διαδίκτυα και Κοινωνία* (Τόμ. Γ'). Πάτρα: ΕΑΠ.
- Ζορκάδης, Β. (2002). *Θεωρία Πληροφορίας και Κωδικοποίησης, Βασικά Ζητήματα Δικτύων Η/Υ* (Τόμ. Α'). Πάτρα: ΕΑΠ.
- Θραμπουλίδης, Κ. (2000). *Γλώσσες Προγραμματισμού, Εισαγωγή στην Πληροφορική* (Τόμ. Δ'). Πάτρα: ΕΑΠ.
- Θραμπουλίδης, Κ. (2001). *Γλώσσες Προγραμματισμού II, Αντικειμενοστρεφής Προγραμματισμός, Σχεδιασμός Λογισμικού* (Τόμ. Γ'). Πάτρα: ΕΑΠ.
- Καμέας, Α. (2000). *Τεχνικές Προγραμματισμού, Εισαγωγή στην Πληροφορική* (Τόμ. Β'). Πάτρα: ΕΑΠ.
- Νικολός, Δ. (2001). *Αρχιτεκτονική Υπολογιστών I, Ψηφιακά Συστήματα* (Τόμ. Β'). Πάτρα: ΕΑΠ.
- Ξένος, Μ., & Χριστοδουλάκης, Δ. (2000). *Βάσεις Δεδομένων, Αρχές Τεχνολογίας Λογισμικού* (Τόμ. Γ'). Πάτρα: ΕΑΠ.
- Τοπάλης Ε. (2002), "Σχεδίαση και Ανάπτυξη μιας Ολοκληρωμένης Πλατφόρμας Διαχείρισης Δικτύων Οικιακής/Κτιριακής Αυτοματοποίησης σε Τοπικό ή Απομακρυσμένο Περιβάλλον", Πανεπιστήμιο Πατρών, Phd Dissertation, Τμήμα Ηλεκτρολόγων Μηχανικών & Τεχνολογίας Υπολογιστών, Πανεπιστήμιο Πατρών, Πάτρα.

8.2 ΞΕΝΟΓΛΩΣΣΗ

A.Sofwan, S. Sumardi, A. I. Ahmada, I. Ibrahim, K. Budiraharjo and K. Karno, "Smart Greetthings: Smart Greenhouse Based on Internet of Things for Environmental Engineering," 2020 International Conference on Smart Technology and Applications (ICoSTA), 2020, pp. 1-5, doi: 10.1109/ICoSTA48221.2020.1570614124.

ACS712 Datasheet (2022). Allegro Microsystems.
<https://www.allegromicro.com/en/search?q=ACS712%20Datasheet>

Adair, M., Allen, S., Kuramochi, Y., Parry, C., Richards, J., & Weller, J. (2005). *Easy X10 Projects for Creating a Smart Home: Easy smart home projects foe 100\$ or less. Covers wireless, Pc, voice and internet controls. Automate lights, security systems, HVAC, and much more!*. Indy-Tech Publishing.

Adame, T., Bel, A., Bellalta, B., Barcelo, J., & Oliver, M, (2014). *IEEE 802.11AH: the WiFi approach for M2M communications*. In IEEE Wireless Communications, vol. 21, no. 6, pp. 144-152. doi: 10.1109/MWC.2014.7000982.

Akilan, T., Srivastava, R., Chandraprabha., Ahmad, B., Islam H., & Saif, M. (2021). *Arduino Based Smart Security For Home Automation*. 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), pp. 804-808. doi: 10.1109/ICAC3N53548.2021.9725436.

Aldrich, F.K. (2003). *Smart Homes: Past, Present and Future*. In: Harper, R. (eds) Inside the Smart Home (pp. 17-39). Springer. London. doi: 10.1007/1-85233-854-7_2.

Allen, S. (2012). *HTTP Succinctly*. Morrisville: Syncfusion Inc.

Arduino Communication Protocols. (2021, October 14). DevicePlus Editorial Team.
<https://www.deviceplus.com/arduino/arduino-communication-protocols-tutorial/#UART>

Arduino Docs (χ.χ.). Arduino Documentation.
https://docs.arduino.cc/?_gl=1*1f8v8y*_ga*OTA2MTE5MTY2LjE2MzEzNjUzODc.*_ga_NEXN8H46L5*MTY0NjU1OTg4Mi42Mi4xLjE2NDY1NTk5MzAuMA

Arduino Integrated Development Environment (IDE) v1. (2022, December 15). Arduino Documentation. <https://docs.arduino.cc/software/ide-v1/tutorials/arduino-ide-v1-basics>

Arduino® MEGA 2560 Rev3. (2023, March 10). Arduino Documentation.
<https://docs.arduino.cc/static/c515f4ec61ce3138338c6df7b9db5a21/A000067-datasheet.pdf>

Arduino Sketches (2023, February 20). Arduino Documentation.
<https://docs.arduino.cc/learn/programming/sketches>

Arduino, <https://www.cooking-hacks.com/documentation/tutorials/geolocation-tracker-gprs-gps-geoposition-sim908-arduino-raspberry-pi/>

Arduino, <http://arduinoiana.org/libraries/tinygps/>

Arduino, <https://forum.arduino.cc/index.php?topic=116867.0>

Arduino, <http://www.circuitstoday.com/pulse-sensor-arduino>

Arduino IDE, <https://www.arduino.cc/en/Main/Software?>

Arduino,sim808, www.stackoverflow.com.

Ashok Baste, P., Jadkar, S. R., Pathak, A. M. (2021). *Weather Station for Solar PV Power Plant Using Arduino Mega*, International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, pp. 1-6, doi: 10.1109/ICCCI50826.2021.9402478.

Baraka, K., Ghobril, M., Malek, S., Kanj, R., & Kayssi, A. (2013). *Low Cost Arduino/Android-Based Energy-Efficient Home Automation System with Smart Task Scheduling*. Fifth International Conference on Computational Intelligence, Communication Systems and Networks, pp. 296-301, doi: 10.1109/CICSYN.2013.47.

Compatiable Digi XBee products (2019, April 19). Digi Knowledge Base. Ανακτήθηκε Φεβρουάριο 21, 2023, από <https://www.digi.com/support/knowledge-base/compatible-digi-xbee-products>

CSS, <https://www.w3schools.com/CSS/>

Damirchi, M. (2021, June 19). Interfacing Passive Buzzer Module with Arduino. Electropeak. <https://electropeak.com/learn/interfacing-passive-buzzer-module-with-arduino/>

Dawoud, S., & Peplow, R. (2010). *Digital system design-use of microcontroller* (p. 570). Taylor & Francis. doi:10.1201/9781003337942

DeKoenigsberg, G. (2008). *How Successful Open Source Projects Work, and How and Why to Introduce Students to the Open Source World*. 21st Conference on Software Engineering Education and Training, pp. 274-276, doi: 10.1109/CSEET.2008.42.

Digital Input Pull-Up Resistor (2023, February 7). Arduino Documentation. Ανακτήθηκε Μάρτιο 5, 2023, Από <https://docs.arduino.cc/tutorials/generic/digital-input-pullup>

Digi XBee (χ, χ). Digi Support Resources. Ανακτήθηκε Φεβρουάριο 16, 2023, από <https://hub.digi.com/support/products/digi-xbee/>

Digi XBee 3 (2021, June 19). Digi Documentation. Ανακτήθηκε Φεβρουάριο 20, 2023, από <https://hub.digi.com/support/products/digi-xbee/digi-xbee3/>

Dyer, R. J. (2015). *Learning MySQL and MariaDB: Heading in the right direction with MySQL and MariaDB*. " O'Reilly Media, Inc."

Documentation Group. (n.d.). About the Apache HTTP Server Project - The Apache HTTP Server Project. https://httpd.apache.org/ABOUT_APACHE.html

Duckett, J. (2011). *HTML & CSS: design and build websites* (Vol. 15). Indianapolis, IN: Wiley.

EBC Exercise 38 Logging to ThingSpeak - eLinux.org. (n.d.).
https://elinux.org/EBC_Exercise_38_Logging_to_ThingSpeak

ESP32 Series (2023). Espressif Documentation. Ανακτήθηκε Μάρτιο 05, 2023, Από
https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32e_esp32-wroom-32ue_datasheet_en.pdf

Ethernet Shield Rev2 (2023, February 20). Arduino Documentation. Ανακτήθηκε Μάρτιο 03, 2023, Από <https://docs.arduino.cc/hardware/ethernet-shield-rev2>

Faludi, R. (2011). *Building Wireless Sensor Networks*. 1st Edition. O'Reilly.

Fan & Blower (2023, January 12). Sunon Products. Ανακτήθηκε Φεβρουάριο 28, 2023, από
[https://www.sunon.com/eu/MANAGE/Docs/PRODUCT/299/480/SUNON%20DC%20Brushless%20Fan%20&%20Blower\(255-E\).pdf](https://www.sunon.com/eu/MANAGE/Docs/PRODUCT/299/480/SUNON%20DC%20Brushless%20Fan%20&%20Blower(255-E).pdf)

FC-28 Soil Moisture Sensor Module (2022). Datasheethub. Ανακτήθηκε Φεβρουάριο 26, 2023, από <https://datasheethub.com/fc-28-soil-moisture-sensor-module/>

Google maps, <https://developers.google.com/maps/documentation/javascript/mysql-to-maps>

Google charts, <https://developers.google.com/chart/>

Goswami, S. A., Padhya, B. P., & Patel, K. D. (2019). *Internet of Things: Applications, Challenges and Research Issues, Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)* (I-SMAC), pp. 47-50, doi: 10.1109/I-SMAC47947.2019.9032474.

GX270 Service Manual (2004, June). Dell manuals. https://dl.dell.com/manuals/all-products/esuprt_desktop/esuprt_optiplex_desktop/optiplex-gx270_service-manual_en-us.pdf

Harold, E. R. (2004). *Java network programming*. " O'Reilly Media, Inc."

H. Andrianto, Suhardi and A. Faizal, "Development of Smart Greenhouse System for Hydroponic Agriculture," 2020 International Conference on Information Technology Systems and Innovation (ICITSI), 2020, pp. 335-340, doi: 10.1109/ICITSI50517.2020.9264917.

HTTP request methods - HTTP | MDN. (2022, September 9). <https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods>

How to Wire and Program a Button Arduino Documentation.
<https://docs.arduino.cc/learn/programming/sketches>

How Does a Rain Sensor Work and how to Interface it with Arduino? .
<https://circuitdigest.com/microcontroller-projects/interfacing-rain-sensor-with-arduino>

IoT Analytics - ThingSpeak Internet of Things. <https://thingspeak.com/>

Java Platform, Enterprise Edition (Java EE) | Oracle Technology Network | Oracle. (2019). Oracle.com. <https://www.oracle.com/java/technologies/java-ee-glance.html>

I2C Serial Interface 20x4 LCD Module (2008, June 6). Handson Technology User Guide.

Ανακτήθηκε Φεβρουάριο 25, 2023, από

https://www.handsontec.com/dataspecs/I2C_2004_LCD.pdf

Kailasam, S., Iswarya, M., Ramyasri, S., & Anu Keerthika, M. (2021). *Arduino based Smart Irrigation System for Home Gardening*. 6th International Conference on Inventive Computation Technologies (ICICT), pp. 1284-1288, doi: 10.1109/ICICT50816.2021.9358498.

Keyanfar, A., Meh, L., & Rabbani, R. (2022). *Application of intelligence in solving architectural problems in the field of housing*. IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), pp. 1-6, doi: 10.1109/CONECCT55679.2022.9865713.

Kilari, G., Mohammed, R., Jayaraman, R. (2020). *Automatic Light Intensity Control using Arduino UNO and LDR*. International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 2020, pp. 0862-0866, doi: 10.1109/ICCSP48568.2020.9182238.

Labiod, H., Afifi, H., De Santis, C.(2007). *WI-FI™, BLUETOOTH™, ZIGBEE™ and WiMAX™*. 1st Edition. Springer.

Learn More About ThingSpeak (χ.χ.). ThingSpeak Documentation.
https://thingspeak.com/pages/learn_more

Lindley, C. (2013). *Exploring JavaScript and the modern DOM: DOM Enlightenment*, First Edition, O'Reilly.

LPG Gas Sensor - MQ-6 (2014, May 1). SparkFun Documents.
<https://www.sparkfun.com/datasheets/Sensors/Biometric/MQ-6.pdf>

M. Aghaseyedabdollah, Y. Alaviyan and A. Yazdizadeh, "IoT Based Smart Greenhouse Design with an Intelligent Supervisory Fuzzy Optimized Controller," 2021 7th International Conference on Web Research (ICWR), 2021, pp. 311-317, doi: 10.1109/ICWR51868.2021.9443022.

Moisture Sensor. (n.d.). Wwww.waveshare.com. <https://www.waveshare.com/moisture-sensor.htm>

MQ-2 Gas Sensor. (n.d.). <https://www.waveshare.com/mq-2-gas-sensor.htm>

Mysql , <https://www.mysql.com/>

Mysql, <https://en.wikipedia.org/wiki/MySQL>

Naramore, E., Gerner, J., Le Scouarnec, Y., Stolz, J., Glass, M. K. (2005). *Beginning PHP5, Apache, and MySQL® Web Development*. 2nd edition. Wrox.

Nixon, R. (2014). *Learning PHP, MySQL, JavaScript, CSS & HTML5: A Step-by-Step Guide to Creating Dynamic Websites*, 3rd Edition, O'Reilly.

JavaScript, <https://en.wikipedia.org/wiki/JavaScript>

Pang, D-F., Lu, S-L., & Zhu, Q-Y. (2014). *Design of Intelligent Home Control System Based on KNX/EIB Bus Network*. International Conference on Wireless Communication and Sensor Network, pp. 330-333, doi: 10.1109/WCSN.2014.74.

Papazoglou, M. P. (2008). *Web Services: Principles and Technology*. Harlow:Pearson Education Ltd.

Php, <https://el.wikipedia.org/wiki/PHP>.

PHP, <https://www.w3schools.com/pHP/default.asp>

Ransing, R., & Rajput, M. (2015). *Smart home for elderly care based on wireless sensor network*. International Conference on Nascent Technologies in the Engineering Field (ICNTE), pp. 1-5, doi: 10.1109/ICNTE.2015.7029932.

Raspberry Pi Documentation. (n.d.). <https://www.raspberrypi.com/documentation/>

Ray, E. (2001). *Creating Self-Describing Data: Learning XML*, 1rd Edition, O'Reilly.

Sim808, [https://www.waveshare.com/wiki/GSM/GPRS/GPS_Shield_\(B\)](https://www.waveshare.com/wiki/GSM/GPRS/GPS_Shield_(B))

Sim808, <http://www.raviyp.com/embedded/194-sim900-gprs-http-at-commands3>

SparkFun: SparkFun XBee Explorer Regulated. <https://www.sparkfun.com/products/11373>

SparkFun: SparkFun XBee Explorer USB . <https://www.sparkfun.com/products/11812>

Sutapa, R., Pritom, A., Subrata, I., Saikat, M., Trideep, H., Mayurakshi, P., ... Debojit, S. (2018). *A Survey Paper on Architecture of Internet of Things*. 9th Annual Information Technology, Electronics and Mobile Communication Conference, pp. 908-913, doi: 10.1109/IEMCON.2018.8614931.

The most complete starterkit (2022, March 24). Elegoo Inc.
<https://www.elegoo.com/pages/arduino-kits-support-files>

Toutsop, O., Kornegay, K., & Smith, E. (2021). *A Comparative Analyses of Current IoT Middle-ware Platforms*. 8th International Conference on Future Internet of Things and Cloud (FiCloud), pp. 413-420, doi: 10.1109/FiCloud49777.2021.00067.

Using Google Charts (2019, September 23). Google Charts Documentation.
<https://developers.google.com/chart/interactive/docs>

Using the Arduino Software (IDE) (2023, February 7). Arduino Documentation.

<https://docs.arduino.cc/built-in-examples/digital/Button?queryID=a5fb25a3d1be51d96dbfdfbbedd0bc10>

Vaidya, V., & Vishwakarma P. (2018). *A Comparative Analysis on Smart Home System to Control, Monitor and Secure Home, based on technologies like GSM, IOT, Bluetooth and PIC Micro-controller with ZigBee Modulation*. International Conference on Smart City and Emerging Technology (ICSCET), pp. 1-4, doi:10.1109/ICSCET.2018.8537381.

What is Arduino? (2023, February 7). Arduino Documentation.

<https://docs.arduino.cc/learn/starting-guide/whats-arduino/>

What is a web server? (2022). MDN Web Docs. https://developer.mozilla.org/en-US/docs/Learn/Common_questions/What_is_a_web_server

What is DBMS? (2022). AppDynamics (Cisco). <https://www.appdynamics.com/topics/database-management-systems>

What is Fritzing? (χ.χ.). FritzingInfoBrochure-highRes.

<https://fritzing.org/media/uploads/publications/FritzingInfoBrochure-highRes.pdf>

What is MySQL? (χ.χ.). MySQL Documentation.

<https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html>

What is the Apache HTTP Server ? (2022). The Apache Software Foundation.

https://httpd.apache.org/ABOUT_APACHE.html

What is XAMPP? (2022). Apache Friends.: <https://www.apachefriends.org/about.html>

Xampp, <https://www.apachefriends.org/index.html>

XCTU User Guide. (2018, June 11). Digi Documentation.

https://www.digi.com/resources/documentation/digidocs/90001458-13/default.htm#concept/c_90001458-13_start.htm?TocPath=____1

Zigbee RF Modules (2022). Digi Documentation.

<https://www.digi.com/resources/documentation/digidocs/pdfs/90000976.pdf>

