



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΙΩΑΝΝΙΝΩΝ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.
ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**Δημιουργία Διαδραστικού Προγράμματος Smart LED με Χρήση
Arduino και Αλληλεπίδραση με Εφαρμογή**

Πάυλος Πουσκούλογλου

Επιβλέποντες:

Γεώργιος Τσουμάνης, ΥΑΔΕ

Χρυσόστομος Στύλιος, ΔΕΠ Καθηγητής

Άρτα, Σεπτέμβριος, 2021

**Implementation of an Interactive “Smart LED Lighting” Program
Using Arduino and a Smartphone App**

Εγκρίθηκε από διμελή εξεταστική επιτροπή

Άρτα, 21 Σεπτεμβρίου 2021

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. Επιβλέπων καθηγητής

Γεώργιος Τσουμάνης,

2. Μέλος επιτροπής

Χρυσόστομος Στύλιος

© Πουσκούλογλου, Πάυλος, 2021.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Δήλωση μη λογοκλοπής

Δηλώνω υπεύθυνα και γνωρίζοντας τις κυρώσεις του Ν. 2121/1993 περί Πνευματικής Ιδιοκτησίας, ότι η παρούσα μεταπτυχιακή εργασία είναι εξ ολοκλήρου αποτέλεσμα δικής μου ερευνητικής εργασίας, δεν αποτελεί προϊόν αντιγραφής ούτε προέρχεται από ανάθεση σε τρίτους. Όλες οι πηγές που χρησιμοποιήθηκαν (κάθε είδους, μορφής και προέλευσης) για τη συγγραφή της περιλαμβάνονται στη βιβλιογραφία.

Πουσκούλογλου Πάυλος

Υπογραφή:

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου, κύριο Γεώργιο Τσουμάνη αλλά και τον κύριο Χρυσόστομο Στύλιο, για την κατανόηση και τη στήριξη που μου έδειξαν όλο αυτό το καιρό.

ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή εργασία αφορά την υλοποίηση ενός project στο οποίο ο χρήστης θα είναι σε θέση να ελέγχει απομακρυσμένα μία λάμπα μέσω εφαρμογής στο κινητό του τηλέφωνο. Για την επίτευξη αυτού του σχεδίου, χρησιμοποιήθηκε ο μικροελεγκτής Arduino Uno R2, καθώς και πλήθος άλλων εργαλείων, όπως το περιβάλλον Arduino IDE αλλά και Shields όπως ο αισθητήρας κίνησης.

Όμως για να κατορθώσουμε να φτάσουμε στην υλοποίηση του project, έπρεπε να κατανοήσουμε τη λειτουργία των μικροελεγκτών (*Κεφάλαιο 1*), καθώς και να περιγράψουμε πως λειτουργεί ένα έξυπνο σπίτι, τα πλεονεκτήματα και μειονεκτήματα που έχει (*Κεφάλαιο 2*).

Στη συνέχεια (*Κεφάλαιο 3*) αναλύθηκε εκτενώς ο εξοπλισμός που χρησιμοποιήθηκε, όπως ο μικροελεγκτής Arduino, οι αισθητήρες που χρησιμοποιήθηκαν στην υλοποίηση του project και διάφορα εξαρτήματα όπως ο ρελές και οι φωτοδίοδοι.

Στο επόμενο κεφάλαιο (*Κεφάλαιο 4*) παρουσιάστηκε βήμα-βήμα όλη η υλοποίηση του project, ενώ αναλύθηκε εκτενώς με σχέδια και φωτογραφίες.

Στο τελευταίο κεφάλαιο τοποθετήθηκαν τα συμπεράσματα και οι προτάσεις για περεταίρω αξιοποίηση της εφαρμογής.

Λέξεις-κλειδιά: Αισθητήρες. Μικροελεγκτής. Διαδραστική εφαρμογή. Απομακρυσμένος χειρισμός.

ABSTRACT

In this project the user will be able to remote control a lighting bulb using an app on his mobile phone. To achieve this, the Arduino microcontroller was used, as well as the Arduino IDE and shields, such as the motion sensor.

For completing the project, we must firstly understand the principles of the microcontrollers (*Chapter 1*) and explain the advantages and disadvantages of smart homes (*Chapter 2*).

In Chapter 3, we analyzed the basics components of the project, such as the Arduino board, the sensors, the relay and the leds.

In Chapter 4, the process followed for the completion of the project is described, along with a lot of photographic material regarding it.

In the final Chapter, some ideas are proposed for further expansion of the project and conclusions are drawn.

Keywords: Programming Arduino board. Sensors and shields. Application.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΕΥΧΑΡΙΣΤΙΕΣ.....	iv
ΠΕΡΙΛΗΨΗ.....	v
ABSTRACT	vi
ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ	x
ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ	xi
ΑΠΟΔΟΣΗ ΑΡΚΤΙΚΟΛΕΞΩΝ	xii
1 ΜΙΚΡΟΕΛΕΓΚΤΕΣ.....	1
1.1 Εισαγωγή	1
1.2 Μικροελεγκτές και μικροεπεξεργαστές.....	1
1.3 Ιστορικά στοιχεία.....	3
1.4 Ανάπτυξη	3
1.5 Όγκος και κόστος.....	4
1.6 Ο μικρότερος υπολογιστής	5
1.7 Δομή.....	5
1.8 Τύποι μικροελεγκτών.....	6
1.9 Γλώσσες προγραμματισμού.....	7
2 ΕΞΥΠΝΟ ΣΠΙΤΙ.....	8
2.1 Έξυπνο σπίτι (Smart home)	8
2.1.1 Ιστορικά στοιχεία	8
2.1.2 Εφαρμογές	9
2.1.3 Προβληματισμοί.....	10
2.2 Το διαδίκτυο των πραγμάτων (IoT).....	11
2.2.1 Πλεονεκτήματα IoT.....	11
2.2.2 Μειονεκτήματα IoT	12
2.2.3 Παραδείγματα IoT.....	13
3 ΜΙΚΡΟΕΛΕΓΚΤΗΣ ARDUINO.....	15

3.1	Πλακέτα Arduino	15
3.1.1	Ιστορική αναδρομή.....	15
3.1.2	Δομικά στοιχεία μιας πλακέτας Arduino.....	15
3.1.3	Οικογένειες πλακετών Arduino και χαρακτηριστικά τους.....	18
3.2	Χαρακτηριστικά της πλακέτας Arduino UNO R2.....	20
3.3	Αναφορά στους μικροελεγκτές.....	22
3.4	Αισθητήρες και Shields	23
3.4.1	Φωτοδίοδος (Led).....	23
3.4.2	Αισθητήρας κίνησης.....	24
3.4.3	Αισθητήρας φωτός.....	26
3.4.4	Relay	27
4	ΥΛΟΠΟΙΗΣΗ PROJECT	30
4.1	Συνοπτική περιγραφή Project	30
4.2	Προγραμματιστικό περιβάλλον MIT APP INVENTOR	30
4.2.1	Οπτική υλοποίηση εφαρμογής	31
4.2.2	Προγραμματισμός της εφαρμογής	32
4.2.3	Συμπεράσματα από την εφαρμογή	36
4.3	Προγραμματιστικό περιβάλλον ArduinoIDE	38
4.4	Κατασκευαστικό τμήμα	42
4.4.1	Κύκλωμα ελέγχου.....	42
4.4.2	Κύκλωμα λειτουργίας.....	47
4.4.3	Φωτογραφικό υλικό κατά τη φάση υλοποίησης της εργασίας.....	48
5	ΣΥΜΠΕΡΑΣΜΑΤΑ.....	52
5.1	Συμπεράσματα	52
5.2	Ιδέες για μελλοντική επέκταση της εφαρμογής.....	53
5.3	Επίλογος.....	53
	ΠΑΡΑΡΤΗΜΑ Α.	54

ΒΙΒΛΙΟΓΡΑΦΙΑ.....	59
ΠΗΓΕΣ ΕΙΚΟΝΩΝ.....	62

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 1: Διαφορές μεταξύ μικροελεγκτών και μικροεπεξεργαστών	3
Πίνακας 2: Κόστος κατασκευής μικροελεγκτών.....	5
Πίνακας 3: Τεχνικά χαρακτηριστικά Arduino UNO R2 [18].....	22
Πίνακας 4: Χρωματικός κώδικας Button εφαρμογής.....	36

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1: Δομικά στοιχεία μικροελεγκτή	1
Εικόνα 2: Εσωτερικό διάγραμμα μικροελεγκτή (u)	2
Εικόνα 3: Δομικά χαρακτηριστικά των πλακετών Arduino [vii]	17
Εικόνα 4: Χαρακτηριστικές πλακέτες οικογενειών Arduino [ix]	20
Εικόνα 5: Διάγραμμα θυρών (Pins) του “Arduino UNO R2”. [x]	21
Εικόνα 6: Δομικά στοιχεία φωτοδιόδου. [iii]	24
Εικόνα 7: Αισθητήρας κίνησης τύπου PIR [iv]	25
Εικόνα 8: Φωτοαντίσταση. [v]	26
Εικόνα 9: Δομικά στοιχεία ενός Ρελέ. [vi]	27
Εικόνα 10: Relay 5 V. [xi]	28
Εικόνα 11: Relay, επαφή NO. [xi]	28
Εικόνα 12: Relay, επαφή NC. [xi]	29
Εικόνα 13: Οπτική σχεδίαση της εφαρμογής για smartphone	31
Εικόνα 14: Δήλωση μεταβλητών	32
Εικόνα 15: Εισαγωγή κώδικα (General Button)	33
Εικόνα 16: Φωτογραφίες Λάμπας. (a) B1=1, (b) B1=2, (c) B1=3	33
Εικόνα 17: Εισαγωγή κώδικα (MoveButton, LightButton)	34
Εικόνα 18: Εισαγωγή κώδικα συνάρτησης LambdaCommunication	35
Εικόνα 19: Παράδειγμα από App	37
Εικόνα 20: Παράδειγμα από App	37
Εικόνα 21: Παραδείγματα από App	38
Εικόνα 22: Εμφάνιση χαρακτηριστικών τοπικού δικτύου	40
Εικόνα 23: Κώδικας για τη λήψη της πληροφορίας από το App	41
Εικόνα 24: Κώδικας συνάρτησης LightOperation	42
Εικόνα 25: Αισθητήρας κίνησης – Shield [xiii]	44
Εικόνα 26: Αισθητήρα φωτός – Φωτοαντίσταση και συνδεσμολογία. [xii]	45
Εικόνα 27: Κουμπί ελέγχου τροφοδοσίας και led ελέγχου τροφοδοσίας	46
Εικόνα 28: Κυκλωματικό διάγραμμα ισχυρών ρευμάτων	47
Εικόνα 29: Μετάβαση από Buttons σε αισθητήρες	48
Εικόνα 30: Συνδεσμολογία Αισθητήρων	49
Εικόνα 31: Ρύθμιση αισθητήρα κίνησης	49
Εικόνα 32: Τελική συνδεσμολογία του σχεδίου	50
Εικόνα 33: Τελική παρουσίαση σχεδίου	50
Εικόνα 34: Τελική εμφάνιση σχεδίου	51
Εικόνα 35: Λειτουργία του project	51

ΑΠΟΔΟΣΗ ΑΡΚΤΙΚΟΛΕΞΩΝ

AMEA:	Άτομα Με Ειδικές Ανάγκες
KME:	Κεντρική Μονάδα Επεξεργασίας
ADC:	Analog to Digital Converter
CPU:	Central processing unit
DAC:	Digital to Analog Converter
EEPROM:	Electronically Erasable Programmable Read Only Memory
EPROM:	Erasable Programmable Read Only Memory
GSM:	Global System for Mobile communication
IC:	Integrated Circuit
ICE:	In-Circuit Emulator
ICT:	Information and Communication Technologies
IoT:	Internet of Things
IP:	Internet Protocol
LCD:	Liquid Crystal Display
LED:	Light Emitting Diode
MCU:	Microcontroller unit.
NC:	Normally Closed (contact)
NO:	Normally Open (contact)
OTP:	One Time Programmable
PIR:	Passive InfraRed (Sensor)
PROM:	Programmable Read Only Memory
PWM:	Pulse Width Modulation
RAM:	Random Access Memory.
RF:	Radio Frequency
ROM:	Read Only Memory.
RX:	Receive
SIM:	Subscriber Identification Module
TTL:	Transistor-Transistor Logic
TX:	Transmit
US:	UltraSonic (Sensor)

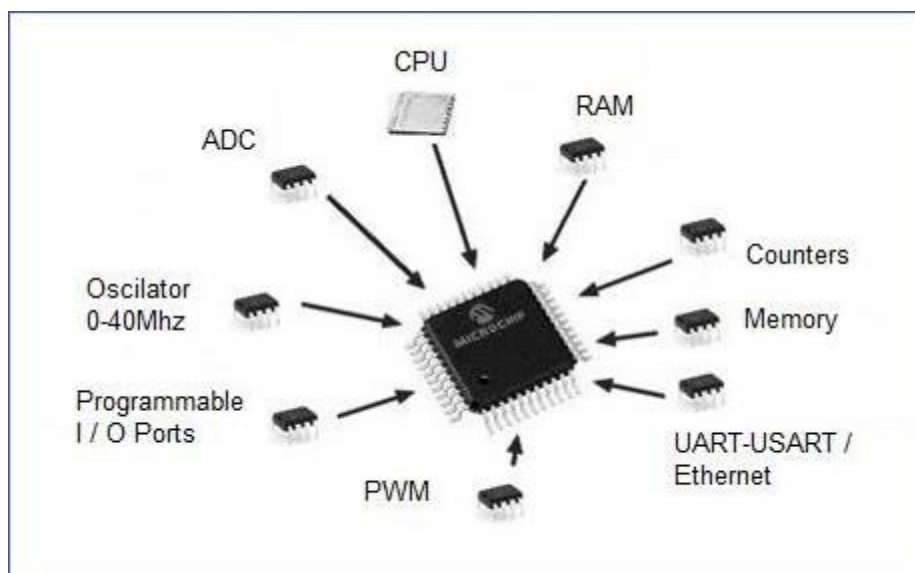
1 ΜΙΚΡΟΕΛΕΓΚΤΕΣ

1.1 Εισαγωγή

Ένας μικροελεγκτής (γνωστός και με τα αρκτικόλεξα MCU (microcontroller unit), μC ή μC) είναι ουσιαστικά ένας μικροϋπολογιστής που αποτελείται από ένα μόνο μικροτσίπ. Λέγεται και ενσωματωμένος (*embedded*) ελεγκτής. Χρησιμοποιείται για τον έλεγχο λειτουργιών σε ενσωματωμένα συστήματα σε διάφορες εφαρμογές όπως μηχανήματα, ρομπότ, οικιακές συσκευές, οχήματα και προσωπικά αξεσουάρ (gadgets).

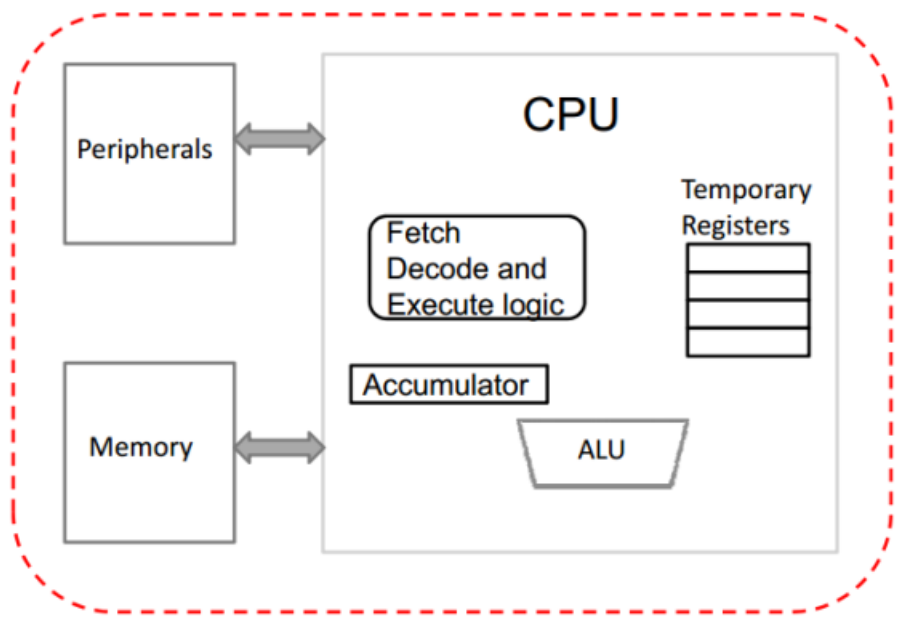
1.2 Μικροελεγκτές και μικροεπεξεργαστές

Οι μικροελεγκτές περιέχουν μικροεπεξεργαστή ώστε να μπορέσουν να λειτουργήσουν και να εκτελέσουν κομμάτια κώδικα και λογικές πράξεις, όμως στην ουσία εμπεριέχουν κι άλλα εξαρτήματα και μπορούν να θεωρηθούν ως μικροεπεξεργαστές με ενσωματωμένα περιφεριακά. Στο κέντρο της Εικόνα 1 φαίνεται ένας μικροελεγκτής, ενώ γύρω από αυτόν φαίνονται τα επιμέρους κομμάτια που εμπεριέχονται σε αυτόν, όπως χιονιστές, θύρες εισόδου/εξόδου, καταχωρητές. Ακόμα στην Εικόνα 2 φαίνεται το διάγραμμα της εσωτερικής του δομής.



Εικόνα 1: Δομικά στοιχεία μικροελεγκτή (ι)

Microcontroller(uC)



Εικόνα 2: Εσωτερικό διάγραμμα μικροελεγκτή (ii)

Οι κυριότερες διαφορές μεταξύ μικροελεγκτών και μικροεπεξεργαστών συνοψίζονται στον παρακάτω Πίνακας 1: [1] [2]

Μικροεπεξεργαστής	Μικροελεγκτής
Το κέντρο ενός υπολογιστή	Το κέντρο ενός ενσωματωμένου συστήματος
Η μνήμη και οι είσοδοι / έξοδοι πρέπει να συνδεθούν ξεχωριστά	Διαθέτει μικροεπεξεργαστή, μνήμη καθώς και εισόδους / εξόδους
Υψηλή κατανάλωση ενέργειας.	Χαμηλή κατανάλωση ενέργειας
Υψηλό κόστος	Χαμηλό κόστος
Χρειάζεται περιφερειακές συσκευές	Διαθέτει ενσωματωμένες περιφερειακές
Δεν διαθέτει διατάξεις εξοικονόμησης ενέργειας.	Διαθέτει διατάξεις εξοικονόμησης ενέργειας.
Χρήση κυρίως σε υπολογιστές	Χρήση σε πολλές διαφορετικές συσκευές και σε ενσωματωμένα συστήματα
Βασίζονται στο μοντέλο του Von Neuman	Βασίζονται στην αρχιτεκτονική Harvard
Η βασική μονάδα επεξεργασίας του είναι ένα μικροτσίπ από πυρίτιο	Διαθέτει κεντρική μονάδα επεξεργασίας, μνήμες RAM και ROM σε ένα μόνο μικροτσίπ
Επικοινωνεί με τις μνήμες RAM και ROM με εξωτερικούς διαύλους (external buses)	Επικοινωνεί με τις μνήμες RAM και ROM με εσωτερικούς διαύλους (internal buses)
Πιο αργή η επικοινωνία με τις μνήμες	Γρηγορότερη η επικοινωνία με τις μνήμες

Τα συστήματα μικροεπεξεργαστών μπορούν να λειτουργήσουν σε πολύ υψηλές συχνότητες	Οι μικροελεγκτές μπορούν να φτάσουν μέχρι τα 200 MHz ή περισσότερο, ανάλογα με την αρχιτεκτονική τους
Χρησιμοποιούνται σε εφαρμογές γενικού σκοπού που διαχειρίζονται μεγάλο όγκο δεδομένων	Χρησιμοποιούνται σε επίπεδο εφαρμογής (application – specific systems)
Χρειάζονται μεγάλο αριθμό εντολών	Χρειάζονται μικρό αριθμό εντολών

Πίνακας 1: Διαφορές μεταξύ μικροελεγκτών και μικροεπεξεργαστών

1.3 Ιστορικά στοιχεία

Οι απαρχές των μικροελεγκτών ξεκινούν από την τεχνολογία των τρανζίστορ τύπου MOSFET. Το transistor MOS ανακαλύφθηκε από τους Mohamed M. Atalla και Dawon Kahng στα εργαστήρια Bell το 1959 και κυκλοφόρησε το 1960. Ο Atalla πρότεινε την κατασκευή ενός ολοκληρωμένου (integrated) κυκλώματος φτιαγμένου εξ' ολοκλήρου από τρανζίστορ MOSFET. Το 1964 τα τσιπ MOS είχαν μεγαλύτερη πυκνότητα τρανζίστορ από τα διπολικά (bipolar) τσιπ. Η πυκνότητα των τσιπ αυξήθηκε ακολουθώντας τον νόμο του Moore, διπλασιάζοντας κάθε περίπου 2 χρόνια. Αυτή η εξέλιξη οδήγησε στην ανάπτυξη της μεγάλης κλίμακας ολοκλήρωσης (Large Scale Integration – LSI), με εκατοντάδες τρανζίστορ σε ένα μικροτσίπ στα τέλη της δεκαετίας του 1960. Η ανάπτυξη των LSI οδήγησε στην κατασκευή των πρώτων μικροεπεξεργαστών (microprocessors). [3] [1]

1.4 Ανάπτυξη

Ο πρώτος μικροελεγκτής κατασκευάστηκε από τους μηχανικούς της Texas Instruments Gary Boone και Michael Cochran το 1971. Ονομάστηκε TMS 1000 και κυκλοφόρησε στην αγορά το 1974. Διέθετε μνήμη μόνο ανάγνωσης (ROM – Read Only Memory), μνήμη εγγραφής / ανάγνωσης (read/write), μικροεπεξεργαστή και είχε ρολόι στο ένα μικροτσίπ.

Τη δεκαετία του 1970 η Ιαπωνία ξεκίνησε την κατασκευή μικροελεγκτών για αυτοκίνητα. Οι μικροελεγκτές που χρησιμοποιήθηκαν σε υαλοκαθαρηστήρες, ραδιόφωνα αυτοκινήτων και ηλεκτρονικές κλειδαριές είχαν 4 bits. Αυτοί που έλεγχαν τον κινητήρα είχαν 8 bits.

Το 1977 η Intel κυκλοφόρησε τον 8048. Συνδύαζε μνήμες τυχαίας προσπέλασης (RAM) και μόνο ανάγνωσης (ROM) στο ίδιο μικροτσίπ. Χρησιμοποιήθηκε σε πάνω από 1 δισεκατομμύριο ηλεκτρονικά υπολογιστών.

Ταυτόχρονα κυκλοφορούσαν και άλλοι παρόμοιοι μικροελεγκτές. Μερικοί διέθεταν μνήμη EPROM. Το πλεονέκτημα μιας μνήμης EPROM είναι ότι διατηρεί τα δεδομένα της ακόμη και όταν χάσει την τροφοδοσία της. Το μειονέκτημα είναι ότι για να αναπρογραμματιστεί θα πρέπει να εκτεθεί σε υπεριώδη ακτινοβολία ώστε να διαγραφούν τα προϋπάρχοντα δεδομένα. Για να επιτευχθεί αυτό, στο καπάκι υπήρχε ένα παραθυράκι από χαλαζία για να μπορεί να σβηστεί η μνήμη από υπεριώδη ακτινοβολία. Ένα άλλο είδος μικροελεγκτών είχαν μνήμη ROM ή PROM, μόνο ανάγνωσης, που προγραμματίζονταν μια φορά μόνο. Οι τελευταίοι είχαν την ένδειξη “OTP” (One Time Programmable). Δεν διέθεταν παραθυράκι χαλαζία, επειδή η μνήμη αυτή δεν μπορούσε να σβηστεί. Οι επαναπρογραμματίσιμες μνήμες με χαλαζία είχαν κεραμικό περίβλημα και ήταν αρκετά ακριβότερες.

Το 1993 η μνήμη EEPROM έκανε εφικτό το σβήσιμο των μικροελεγκτών με ηλεκτρισμό. Ο πρώτος μικροελεγκτής με μνήμη EEPROM ήταν το μικροτσίπ PIC16C84. Επιπλέον δεν χρειαζόταν το ακριβό κεραμικό περίβλημα. Την ίδια χρονιά η εταιρεία Atmel κυκλοφόρησε τον πρώτο μικροελεγκτή με μνήμη flash, που είναι ένα είδος μνήμης EEPROM. Η μνήμη EEPROM προϋπήρχε, όμως δεν χρησιμοποιήθηκε νωρίτερα λόγω υψηλού κόστους. Σήμερα οι μικροελεγκτές είναι τόσο φθινοί και εύκολα / άμεσα διαθέσιμοι, που χρησιμοποιούνται και από ερασιτέχνες. [3] [1]

1.5 Όγκος και κόστος

Το 2002, το 55% των πωλήσεων παγκοσμίως ήταν μικροελεγκτές των 8 bits. Ένα μέσο νοικοκυριό σε μια ανεπτυγμένη χώρα έχει 3 μικροελεγκτές γενικού σκοπού και πάνω από 30 συνολικά. Παραδείγματα συσκευών με μικροελεγκτές είναι πλυντήρια, φούρνοι μικροκυμάτων, σταθερά τηλέφωνα. Το κόστος κατασκευής για κάθε κομμάτι είναι συνήθως κάτω από 0,1 \$. Το κόστος μειώνεται με την πάροδο του χρόνου όπως φαίνεται και από τον Πίνακα 2, εκτός από το 2012, όπου αυξήθηκε απότομα λόγω της οικονομικής κρίσης. [1]

Κόστος (σε \$) για 1000 μικροελεγκτές			
Έτος	8 bits	16 bits	32 bits
2012	0,59	0,69	1,76
2015	0,311	0,385	0,503
2018	0,03	0,378	0,393

Πίνακας 2: Κόστος κατασκευής μικροελεγκτών

1.6 Ο μικρότερος υπολογιστής

Το πανεπιστήμιο του Michigan ανακοίνωσε στις 21 Ιουνίου 2018 την κατασκευή του μικρότερου υπολογιστή στην ιστορία. Το μικροτσίπ είχε διαστάσεις $0,04 \text{ mm}^3$ (μικρότερος από έναν κόκκο ρυζιού) και τροφοδοσία 16 nW. Η συσκευή είχε ασύρματους αισθητήρες και δεν διέθετε μπαταρία. Είχε ενσωματωμένο έναν μικροεπεξεργαστή Cortex-M0+. Διέθετε επίσης μνήμη RAM, φωτοβολταϊκά στοιχεία για την τροφοδοσία του και ασύρματο πομποδέκτη. Λόγω του μικρού του μεγέθους δε μπορεί να φέρει τυπικές κεραίες ραδιοσυχνότητας. Επομένως χρησιμοποιεί το ορατό φως για αποστολή και λήψη δεδομένων. Ένας σταθμός βάσης (Base station) παρέχει το φως για τροφοδοσία και προγραμματισμό εντολών και λαμβάνει τα δεδομένα. Η συσκευή έχει το 1/10 του μεγέθους του προηγούμενου μικρότερου υπολογιστή που κατασκεύασε η IBM το 2018. Ο τελευταίος περιείχε 1 εκατομμύριο τρανζίστορ, κόστιζε λιγότερο από 0,10 \$ και διέθετε τεχνολογία blockchain και προοριζόταν για εφαρμογές κρυπτογράφησης και ασφάλειας των επικοινωνιών. [1]

1.7 Δομή

Κάθε ηλεκτρική συσκευή που αποθηκεύει, μετράει και τυπώνει ή προβάλλει πληροφορίες σε μια οθόνη ή με κάποιον άλλο τρόπο. Η βασική δομή ενός μικροελεγκτή περιλαμβάνει τα παρακάτω στοιχεία:

- **CPU:** Ο μικροελεγκτής με κεντρική μονάδα επεξεργασίας (CPU). Η CPU λαμβάνει τα δεδομένα από την προγραμματισμένη μνήμη. Μετά τα αποκωδικοποιεί και τα μεταφράζει σε εντολές. Στο τέλος εκτελεί τις εντολές.
- **Μνήμη:** Η μνήμη λειτουργεί με παρόμοιο τρόπο με την CPU. Τα μικροτσίπ της μνήμης αποθηκεύουν όλες τις εντολές και τα δεδομένα που λαμβάνει η συσκευή που είναι εγκατεστημένος ο μικροελεγκτής από το περιβάλλον. Διαθέτει μια ορισμένη ποσότητα RAM ή ROM μνήμης (EPROM, EEPROM, μνήμη flash).

- Είσοδοι – έξοδοι (I/O): Μεταφέρουν δεδομένα από το περιβάλλον ή το χρήστη στη μνήμη. Οδηγούν διάφορες συσκευές, όπως εκτυπωτές, οθόνες LCD, LED.
- Σειριακές θύρες (Serial Ports) : Μερικοί μικροελεγκτές διαθέτουν μια διεπαφή χρήστη (interface) που μεσολαβεί ανάμεσα στο χρήστη και στη μνήμη. Δίνουν τη δυνατότητα χρήσης περιφερειακών συσκευών.
- Χρονιστές / Μετρητές (Timers / Counters) : Ένας μικροελεγκτής μπορεί να διαθέτει έναν ή περισσότερους χιονιστές. Εκτελεί όλες τις λειτουργίες μέτρησης και χρονισμού. Οι χρονιστές χρησιμοποιούν ένα εσωτερικό ρολόι (ταλαντωτής) για να κρατούν το χρόνο (π.χ. χρονόμετρο σε ένα φανάρι δρόμου). Οι μετρητές κρατούν πόσες φορές έχει εκτελεστεί μια εντολή ή πόσες φορές έχει εμφανιστεί μια τιμή στη μνήμη. Επίσης μπορούν να μετράνε συχνότητα και να κάνουν διαμορφώσεις (modulations).
- ADC – DAC converter (Μετατροπέας αναλογικού σε ψηφιακό ή αντίστροφα): Οι μετατροπείς ADC χρησιμοποιούνται σε συσκευές μέτρησης και σε gadgets. Οι μετατροπείς DAC ελέγχουν την ορθή λειτουργία συσκευών όπως κινητήρες.
- Έλεγχος διερμηνευτή (Interpret control) : Χρησιμοποιείται για να δώσει εντολές ελέγχου με χρονοκαθυστέρηση (delayed control). Το πρόγραμμα που μεταφράζει τις εντολές σε γλώσσα μηχανής λέγεται διερμηνευτής (interpreter). [2] [4]

1.8 Τύποι μικροελεγκτών

Οι μικροελεγκτές χωρίζονται σε κατηγορίες ανάλογα με την αρχιτεκτονική σχεδίασης, το μέγεθος της μνήμης, το σύνολο εντολών (instruction set) και τον αριθμό των bits που χρησιμοποιούν. Ακολουθούν οι 3 πιο χαρακτηριστικές κατηγορίες:

- **8 bits** : Εκτελεί μόνο αριθμητικές και λογικές πράξεις λόγω του μικρού αριθμού bits. Παράδειγμα μικροελεγκτή είναι ο Intel 8031 / 8051.
- **16 bits** : Ίδιες λειτουργίες με τον μικροελεγκτή 8 των bits. Χρησιμοποιείται σε εφαρμογές που απαιτούν μεγαλύτερη ακρίβεια. Παράδειγμα μικροελεγκτή είναι ο Intel 8096.
- **32 bits** : Χρησιμοποιείται σε ευρύτητα και είναι ο πιο διαδεδομένος. Εφαρμογές του περιλαμβάνουν τη βιομηχανία, ιατρικά μηχανήματα. Οι εντολές του έχουν μέγεθος 32 bits. [4] [5]

1.9 Γλώσσες προγραμματισμού

Οι πρώτοι μικροελεγκτές αρχικά προγραμματίζονταν σε assembly. Ο μικροελεγκτής 8052 της Intel έτρεχε σε Basic, ενώ ο Zilog Z8 χρησιμοποιούσε τη γλώσσα Forth. Η Basic χρησιμοποιείται ακόμη σε ορισμένους σύγχρονους μικροελεγκτές. Αργότερα ξεκίνησαν να χρησιμοποιούνται και άλλες γλώσσες, όπως οι C, Python και Java script, ιδιαίτερα στα ενσωματωμένα συστήματα. Οι μεταγλωττιστές (compilers) γενικού σκοπού έχουν κάποιους περιορισμούς και βελτιώσεις για καλύτερη απόδοση ορισμένων χαρακτηριστικών τους. Οι κατασκευαστές συχνά διαθέτουν δωρεάν ορισμένα εργαλεία για να κάνουν ευκολότερη την υιοθέτησή τους από την αγορά.

Ορισμένοι μικροελεγκτές τρέχουν κάποιο ειδικό λογισμικό. Παράδειγμα είναι ο 8051 που τρέχει σε SDCC που είναι μια εκδοχή της γλώσσας C. Το μειονέκτημά τους είναι η μη χρήση των βιβλιοθηκών των περισσότερων γλωσσών, λόγω ασυμβατότητας. Οι διερμηνευτές (interpreters), μερικές φορές χρησιμοποιούν γλώσσες όπως οι Micro Python και η Circuit Python. Αυτοί οι διερμηνευτές υποστηρίζουν τον διαδραστικό προγραμματισμό (interactive programming).

Οι προσομοιωτές (simulators) είναι συμβατοί με ορισμένους μικροελεγκτές. Επιτρέπουν στο χρήστη να κάνει μια δοκιμή του εξοπλισμού σαν να λειτουργούσε σε σύνδεση με πραγματικές συσκευές. Ο προσομοιωτής δείχνει την κατάσταση του εσωτερικού επεξεργαστή και των εξόδων και δημιουργεί σήματα στις εισόδους. Παρ' όλο που δεν είναι δυνατή η προσομοίωση όλων των συσκευών και η εξέταση όλων των πιθανών σεναρίων, οι προσομοιωτές είναι ο πιο εύκολος, γρήγορος και φθηνός τρόπος για ανάλυση και αποσφαλμάτωση (debugging) του κώδικα.

Οι μικροελεγκτές τελευταίας τεχνολογίας διαθέτουν εξομοιωτές ενσωματωμένους στο μικροτσίπ. Αυτοί οι εξομοιωτές λέγονται In – Chip Emulators (ICE). Ο εξομοιωτής ICE πραγματικού χρόνου επιτρέπει την διαχείριση / αλλαγή των εσωτερικών καταστάσεων του μικροελεγκτή ενώ λειτουργεί. Για την καταγραφή των εσωτερικών καταστάσεων χρησιμοποιεί σημεία εκκίνησης (trigger points). [1]

2 ΕΞΥΠΝΟ ΣΠΙΤΙ

2.1 Έξυπνο σπίτι (Smart home)

Η λέξη “domotics” (και η λέξη “domotica” όταν χρησιμοποιείται ως ρήμα) είναι σύντμηση της λατινικής λέξης “domus” (σπίτι) και της λέξης “robotics”. Η φράση “έξυπνο σπίτι” (smart home) αναφέρεται σε ένα οικιακό σύστημα που γνωρίζει την κατάσταση των συσκευών του κάθε στιγμή. Αυτή η επιτήρηση γίνεται μέσω συστημάτων επιτήρησης που βασίζονται σε τεχνολογίες πληροφορικής και τηλεπικοινωνιών (Information and Communication Technologies – ICT). Τα τελευταία χρόνια μεγάλη επίδραση στα έξυπνα σπίτια έχει το διαδίκτυο των πραγμάτων (IoT). [6]

2.1.1 Ιστορικά στοιχεία

Η αυτοματοποίηση των σπιτιών ξεκίνησε με την εισαγωγή των ηλεκτρικών συσκευών για τις οικιακές εργασίες. Η ανάπτυξη του δικτύου διανομής ηλεκτρικής ενέργειας στις αρχές του προηγούμενου αιώνα έφερε στα σπίτια αυτές τις συσκευές. Παραδείγματα είναι ο βραστήρας νερού (1889), το πλυντήριο ρούχων (1904), το ψυγείο (1913) και άλλες όπως η ραπτομηχανή, το πλυντήριο πιάτων και το στεγνωτήριο ρούχων.

Το πρώτο δίκτυο επικοινωνίας των ηλεκτρικών συσκευών αναπτύχθηκε το 1975 και λεγόταν “X10”. Χρησιμοποιεί το δίκτυο διανομής ηλεκτρικής ενέργειας για μετάδοση σημάτων ελέγχου των συσκευών. Τα σήματα ελέγχου είναι σύντομες ριπές (bursts) δεδομένων σε ψηφιακή μορφή. Το μήκος κύματος ήταν στο φάσμα των ραδιοσυχνοτήτων (radio frequency – RF). Το 1978 το σύστημα X10 διέθετε έναν πίνακα ελέγχου εντολών 16 καναλιών, ένα στοιχείο ελέγχου των λαμπτήρων και ένα για τις υπόλοιπες οικιακές συσκευές. Σύντομα προστέθηκε μια μονάδα ελέγχου των διακοπών τοίχου και ο πρώτος χρονιστής (timer).

Σύμφωνα με την εταιρεία ABI, το 2012 υπήρχαν στις ΗΠΑ 1,5 εκατομμύριο συστήματα οικιακού αυτοματισμού. Η εταιρεία Statista υπολόγισε τον αριθμό των έξυπνων οικιακών συσκευών στα 45 εκατομμύρια για το έτος 2018 στις ΗΠΑ. [6]

2.1.2 Εφαρμογές

Κάθε χρήστης έχει διαφορετικές απαιτήσεις από ένα έξυπνο σπίτι. Όμως όλες έχουν ως κύριο γνώμονα την αύξηση της άνεσης που παρέχει το σπίτι προς τους ενοίκους καθώς και την εξοικονόμηση ενέργειας, είτε πρόκειται για ηλεκτρική ενέργεια είτε για άλλη μορφή όπως το πετρέλαιο του λέβητα για τη θέρμανση του σπιτιού. Παράλληλα ο χρήστης επιτυγχάνει και την αύξηση της ασφάλειας του σπιτιού του, καθώς πέραν του εγκατεστημένου συστήματος συναγερμού του δίνεται η δυνατότητα να χειρίζεται απομακρυσμένα τα φώτα του σπιτιού του ή την τηλεόραση, με αποτέλεσμα να δημιουργείται η ψευδαίσθηση της ύπαρξης ενοίκου εντός οικίας και να αποθαρρύνονται τυχόν διαρρήκτες.

Συνοψίζοντας, οι κυριότερες εφαρμογές της τεχνολογίας των έξυπνων σπιτιών είναι οι εξής:

- Θέρμανση, εξαερισμός και κλιματισμός. Αυτοί οι παράγοντες είναι πολύ σημαντικοί για τους ενοίκους ενός σπιτιού. Η θερμική άνεση καθώς και το σωστό μικρόκλιμα που επιτυγχάνεται όπως καθαρός αέρας, έλεγχος υγρασίας, κ.α. συντελούν στην ψυχική ευεξία των ενοίκων. Ο έλεγχος όλων αυτών των παραγόντων, μπορεί να γίνει και απομακρυσμένα μέσω συσκευών που συνδέονται στο διαδίκτυο ή ακόμη και προγραμματισμένα με βάση τις προτιμήσεις και τις ανάγκες του χρήστη. Τέλος μπορεί να χρειαστεί να θέλουμε να πετύχουμε διαφορετική θερμοκρασία σε διαφορετικά δωμάτια, για παράδειγμα παιδικό δωμάτιο και τουαλέτα, ή για λόγους εξοικονόμησης ενέργειας να μη ρυθμίζουμε καθόλου τη θερμοκρασία σε κλειστά δωμάτια του σπιτιού.
- Φωτισμός. Ένα "έξυπνο" δίκτυο περιλαμβάνει τη διασύνδεση των εισόδων – εξόδων του συστήματος. Με τη χρήση αυτού του συστήματος ο ένοικος μπορεί να σβήνει τα «ξεχασμένα» φώτα στο σπίτι απομακρυσμένα, να τα θέτει να ανάβουν μέσω αισθητήρων για εξοικονόμηση ενέργειας και γενικά να προσαρμόσει τον τρόπο λειτουργίας τους στις ανάγκες του.
- Έλεγχος φυσικής παρουσίας. Είναι ένα σύστημα, συνήθως από αισθητήρες κίνησης και θερμικές κάμερες, το οποίο απενεργοποιεί τις συσκευές που πρέπει να είναι ανενεργές όταν λείπουν οι ιδιοκτήτες, όπως θερμοσίφωνα, θέρμανση,

τηλεόραση, για εξοικονόμηση ενέργειας, ενώ ενεργοποιεί το συναγερμό του σπιτιού και κλειδώνει όλες τις ελεγχόμενες εισόδους του.

- Συναγερμός. Χρήσιμο για την ασφάλεια σε περίπτωση διάρρηξης / παραβίασης. Χρησιμοποιούνται κάμερες, αισθητήρες κίνησης, μαγνητικές επαφές και άλλα αισθητήρια ώστε να αποτρέψουν την πιθανή διάρρηξη της οικίας ή να ειδοποιηθεί έγκαιρα ο ένοικος και οι αρμόδιες αρχές όταν αυτό συμβεί.
- Πυρασφάλεια. Με χρήση αισθητήρων καπνού και εκπομπής διοξειδίου του άνθρακα σε περίπτωση πυρκαγιάς, το έξυπνο σπίτι θα ειδοποιήσει άμεσα τις αρμόδιες αρχές και τους ενοίκους ενώ θα εκτελέσει όλες τις προγραμματισμένες ενέργειες για περιορισμό ή ακόμη και σβήσιμο της πυρκαγιάς.
- Αισθητήρες διαρροής νερού, αερίου. Αν υπάρξει διαρροή νερού ή γκαζιού το έξυπνο σπίτι μπορεί να προβεί σε ενέργειες για να περιορίσει τη ζημιά, όπως διακοπή της τροφοδοσίας νερού ή φυσικού αερίου του σπιτιού, και να ειδοποιήσει τους ενοίκους για την ύπαρξη του προβλήματος.
- Έξυπνες συσκευές ειδικά για ηλικιωμένους και άτομα με ειδικές ανάγκες (AMEA).
- Σύστημα παρακολούθησης των κινήσεων και περιορισμού της πρόσβασης για κατοικίδια ζώα και μωρά / νήπια.
- Έλεγχος ποιότητας του αέρα. Χρησιμοποιείται κυρίως σε κατοικίες που βρίσκονται κοντά σε βιομηχανικές ζώνες. Το σύστημα μπορεί να χρησιμοποιεί φίλτρα αέρα ή / και να κλείνει τα παράθυρα αν εντοπίσει αυξημένη μόλυνση.
- Έξυπνη κουζίνα. Η εισαγωγή ρομπότ δίνει τη δυνατότητα μαγειρέματος με βάση μια έτοιμη συνταγή.
- Έλεγχος μέσω φωνής. Προγράμματα τεχνητής νοημοσύνης όπως το “Alexa” και το “Google Home” λειτουργούν σαν οικιακοί σύμβουλοι και βοηθούν στον καθημερινό προγραμματισμό των εργασιών. [6]

2.1.3 Προβληματισμοί

Παρ’ όλα τα αδιαμφισβήτητα πλεονεκτήματα που προσφέρουν, οι σκεπτικιστές παραθέτουν μια σειρά από αδυναμίες που έχουν τέτοιου είδους συστήματα τα κυριότερα από τα οποία είναι:

- Πρόσβαση και επεξεργασία προσωπικών δεδομένων από τρίτους.
- Ζητήματα ασφαλείας, όπως για παράδειγμα αν εισβολείς καταφέρουν να «ξεκλειδώσουν» έναν κωδικό πρόσβασης, θα είναι σε θέση να απενεργοποιήσουν όλες τις κάμερες, και τα συστήματα ασφαλείας του σπιτιού.
- Βλάβη στο σύστημα. Αν και σπάνιο, μία βλάβη σε ένα πλήρως αυτοματοποιημένο έξυπνο σπίτι μπορεί να προκαλέσει μεγάλη αναστάτωση στον ιδιοκτήτη, όπως για παράδειγμα αν δεν τον αναγνωρίζει και δεν τον αφήνει να απενεργοποιήσει τα συστήματα ασφαλείας του σπιτιού για να μπει στο σπίτι του. [7]

2.2 Το διαδίκτυο των πραγμάτων (IoT)

Το διαδίκτυο των πραγμάτων ή IoT (Internet of Things), είναι ο συνδυασμός «Πραγμάτων», δηλαδή αισθητήρια, προγράμματα, συσκευές και άλλα, που διασυνδέονται μεταξύ τους μέσω του διαδικτύου με σκοπό τη συλλογή και ανταλλαγή πληροφοριών, δεδομένων και υπηρεσιών, για την διευκόλυνση του χρήστη. Για παράδειγμα το έξυπνο σπίτι που αναπτύχθηκε στην προηγούμενη παράγραφο αποτελεί ένα τέτοιο παράδειγμα IoT. [8]

2.2.1 Πλεονεκτήματα IoT

Με τη συλλογή και ανταλλαγή δεδομένων και υπηρεσιών με ολοένα και ταχύτερους ρυθμούς, είναι πλέον φανερό ότι τα IoT κατέχουν πλέον ένα αναπόσπαστο κομμάτι της καθημερινής μας ζωής. Τα πλεονεκτήματά τους μπορούν να ομαδοποιηθούν σε κατηγορίες, τα κυριότερα όμως μπορούν να συνοψιστούν στις εξής:

- Παρακολούθηση δεδομένων. Αποτελεί ίσως το σημαντικότερο πλεονέκτημα των εφαρμογών IoT, με πληθώρα παραδειγμάτων στην καθημερινή μας ζωή. Με βάση τη συλλεγόμενη πληροφορία, που υπό άλλες συνθήκες θα ήταν αρκετά δύσκολο να την αποκτήσουμε, μπορούμε να αποφύγουμε αρκετά χρονοβόρες καταστάσεις, όπως για παράδειγμα την κίνηση στους δρόμους σε ώρα αιχμής. Ακόμη μπορούμε να γνωρίζουμε δεδομένα με άμεσο αντίκτυπο στην υγεία και την ευεξία μας, όπως την ποιότητα του αέρα του σπιτιού μας, τους καρδιακούς μας παλμούς σε γυμναστική ή ακόμη και αν κάποιο προϊόν στο ράφι του σούπερ μάρκετ έχει λήξει.

- Προσβασιμότητα. Με την ανάπτυξη των IoT, η πρόσβαση σε δεδομένα και υπηρεσίες έχει πάρει ραγδαία ανάπτυξη. Καθημερινά μπορούμε να έχουμε πρόσβαση ή να διεκπεραιώσουμε τις διάφορες υποχρεώσεις μας μέσω του διαδικτύου. Πλέον μπορούμε να μεταφέρουμε χρήματα μέσω του smart κινητού μας, να περιηγηθούμε σε δεκάδες καταλύματα πριν κάνουμε κράτηση για διακοπές ή ακόμη και να διαβάσουμε χιλιάδες επιστημονικά άρθρα.
- Εξοικονόμηση χρόνου. Χρησιμοποιώντας τις εφαρμογές IoT με όλες τις πληροφορίες που μας παρέχονται, μπορούμε να αυτοματοποιήσουμε κάποιες διαδικασίες οι οποίες επαναλαμβάνονται. Συνεπώς μπορούμε να γλυτώσουμε πολύτιμο χρόνο, τον οποίο μπορούμε να τον διαθέσουμε αλλού. Ακόμη μπορούμε να έχουμε αποδοτικότερο προγραμματισμό των εργασιών μας.
- Έλεγχος αντικειμένων. Εφόσον οι αισθητήρες και πολλά εξαρτήματα συνδέονται μεταξύ τους, μπορούμε πλέον να χειριζόμαστε πάρα πολλά πράγματα από απόσταση διευκολύνοντας με αυτό τον τρόπο τη ζωή μας και γλυτώνοντας χρόνο.
- Εξοικονόμηση χρημάτων. Τέλος δεν μπορεί να μη γίνει αναφορά στο οικονομικό όφελος που προκύπτει από την χρήση των IoT. Αρχικά ας αρχίσουμε ότι δεν τίθεται ζήτημα αν η αξίας της πληροφορίας που λαμβάνουμε είναι μεγαλύτερη από τα χρήματα που δαπανάμε για να την αποκτήσουμε, καθώς πλέον είναι ευρέως αποδεκτό ότι χρησιμοποιώντας τα IoT εξοικονομούμε χρήματα τα οποία θα ξοδεύαμε για την απόκτηση πληροφοριών με άλλα μέσα. [9] [8] [10]

2.2.2 Μειονεκτήματα IoT

Τα μειονεκτήματα των εφαρμογών IoT είναι παρόμοια με τα μειονεκτήματα που αναφέραμε για τα έξυπνα σπίτια. Τα πιο χαρακτηριστικά από αυτά είναι:

- Ζητήματα ασφαλείας και παραβίαση προσωπικών δεδομένων. Πιστεύω ότι δεν χρειάζεται καν να αναφερθεί κανείς στο δυσάρεστο ξάφνιασμά όταν ανακαλύπτει ότι οι ηλεκτρονικοί κωδικοί της τράπεζάς του έχουν υποκλαπεί. Υπάρχουν πολλά είδη κλοπής και παραβίασης προσωπικών δεδομένων, τα οποία κάποιοι μπορούν να τα εκμεταλλευτούν προς όφελός τους.
- Βλάβη στον προγραμματισμό. Δυστυχώς όσο και απλές να φαίνονται κάποιες διεργασίες, στην πραγματικότητα δεν είναι. Πολλές φορές χρειάζονται πολλές

γραμμές κώδικα για να εκτελεστεί μία λειτουργία, συνεπώς αυξάνονται και οι πιθανότητες να υπάρξει κάποιο λάθος στον κώδικα, με ποικίλα αποτελέσματα.

- Εξάρτηση από την τεχνολογία. Με την πρόοδο των εφαρμογών IoT βασιζόμαστε όλο και περισσότερο σε αυτές τις εφαρμογές για τη ρύθμιση και επίλυση των καθημερινών ζητημάτων μας, από απλά μέχρι και σύνθετα, με συνέπεια σιγά σιγά να μην λαμβάνουμε πρωτοβουλίες για την επίλυση των ζητημάτων αυτών. Φυσικά αν αναλογιστεί κανείς πόσο συνδεδεμένη είναι η λειτουργία της σημερινής κοινωνίας με το διαδίκτυο σε σχέση με τα προηγούμενα χρόνια, μπορεί να καταλάβει και το μέγεθος της εξάρτησης που έχουμε με αυτού του είδους τις εφαρμογές.
- Συμβατότητα. Αν και πλέον έχει εξαλειφθεί σε μεγάλο βαθμό, υπάρχει ακόμη πρόβλημα συμβατότητας στο συνδυασμό κάποιων υλικών. Όπως είπαμε τα IoT είναι ένας συνδυασμός πραγμάτων, οπότε είναι πολύ πιθανό αυτά τα πράγματα να μην έχουν μεταξύ τους συμβατότητα ή να μη μπορούν να δουλέψουν αρμονικά μεταξύ τους. [9] [8] [10]

2.2.3 Παραδείγματα IoT

Τα IoT έχουν τόσο ευρεία χρήση που είναι δύσκολο να περιγράψει κανείς και να οριοθετήσει πλέον τις εφαρμογές τους. Την κατάσταση επιδεινώνει το γεγονός ότι η ενεργειακή κατανάλωση των αισθητηρίων συνεχώς μειώνεται, όπως και το κόστος αγοράς τους, με αποτέλεσμα να έχουν υλοποιηθεί αρκετά, σχεδόν για κάθε πτυχή στη ζωή μας. Από τις χρήσεις και πιθανές εφαρμογές των IoT ξεχωρίζουν οι:

- Ιατρική. Ήδη εφαρμόζεται αλλά είναι από τους τομείς που μπορούν να γίνουν μεγάλες αλλαγές κυρίως για την παρακολούθηση των ασθενών στα νοσοκομεία της χώρας. Χρησιμοποιώντας μια εφαρμογή IoT, οι γιατροί και οι νοσηλευτές θα μπορούν να λαμβάνουν απευθείας δεδομένα για την κατάσταση του ασθενή. Τέτοια δεδομένα θα μπορούσε να είναι η πίεση, οι καρδιακοί παλμοί ή θερμοκρασία και άλλα. Συνεπώς θα μπορούσαν να έχουν μια πιο ολοκληρωμένη εικόνα για την εξέλιξη της ανάρρωσής του ασθενή, και να επεμβαίνουν άμεσα όταν η κατάσταση του ασθενή υποτροπιάσει.
- Τηλεκπαίδευση και Τηλεργασία. Αν και το σενάριο της τηλεκπαίδευσης και της τηλεργασίας δεν είναι κάποιο καινούριο εγχείρημα, αναπτύχθηκε όμως πολύ απότομα μετά την αρχή της υγειονομικής κρίσης του Covid-19. Συνεπώς τα IoT

πέρα ότι είναι μια εναλλακτική ασφαλής μέθοδος για εκπαίδευση, έχει ταυτόχρονα πάρα πολλές προοπτικές εξέλιξης στον τομέα αυτόν. Μέσω των IoT το μάθημα θα μπορούσε να γίνει πιο ενδιαφέρον, ο υπεύθυνος δάσκαλος να εντόπιζε άμεσα τις αδυναμίες των μαθητών του για να μπορέσει να τις αντιμετωπίσει, ενώ ο γονιός να έχει άμεση ενημέρωση για την πρόοδο του παιδιού του. Επίσης μια άλλη εφαρμογή των IoT στην εκπαίδευση είναι ότι λόγω των πληροφοριών που συλλέγουν, μπορούν να εντοπίζουν πιο γρήγορα τις αδυναμίες των μαθητών, και σε συνδυασμό με διάφορα εκπαιδευτικά παιχνίδια και δραστηριότητες θα μπορούν να τις αντιμετωπίσουν και να τις λύσουν αυτόνομα, και κυρίως χωρίς ο μαθητής να νοιώσει πίεση εξετάσεων ή άγχους όπως με τις παραδοσιακές μεθόδους.

- Κρατικές υπηρεσίες – Διακυβέρνηση. Έχουν πραγματοποιηθεί άλματα για τη μείωση της γραφειοκρατίας και τη διεκπεραίωση υποθέσεων μέσω του διαδικτύου. Ίσως τα πιο σημαντικά από αυτά στην Ελλάδα να είναι οι υπηρεσίες του taxisnet και του GOV.gr. Χρησιμοποιώντας αυτές τις πλατφόρμες, ο κάθε φορολογούμενος ή ευρύτερα ο κάθε Έλληνας πολίτης, μπορεί να δει αναλυτικά τα στοιχεία του ή να κάνει αιτήσεις για έκδοση ηλεκτρονικών πιστοποιητικών. Τέτοια συστήματα εκτός ότι διευκολύνουν τις συναλλαγές του πολίτη με το κράτος, όταν υπάρξει διασύνδεση πολλών τέτοιων συστημάτων θα αυξηθεί και ο βαθμός ασφαλείας των δεδομένων των πολιτών.
- Βιομηχανία. Είναι πραγματικά αμέτρητοι οι τρόποι που τα IoT μπορούν να υποστηρίξουν την εύρυθμη λειτουργία μιας εταιρείας και να αυξήσουν την κερδοφορία της. Ανάλογα με τον τομέα ειδίκευσης της επιχείρησης, μπορούν να συλλεχθούν δεδομένα όπως παρακολούθηση τάσεων του αγοραστικού κοινού, να αυτοματοποιηθούν κομμάτια της παραγωγικής της διαδικασίας, να λαμβάνονται δεδομένα από τα μηχανήματα για πρόληψη πιθανών βλαβών και προγραμματισμού της συντήρησης των μηχανημάτων, να αυξηθεί η ποιότητα του εξαγόμενου προϊόντος, να μειωθεί η κατανάλωση ενέργειας με ορθολογικότερη κατανομή αυτής και άλλα. Τέλος μπορεί να συμβάλουν και στη μείωση των εργατικών ατυχημάτων, καθώς θα δίνεται η δυνατότητα του απομακρυσμένου χειρισμού επικίνδυνων μηχανημάτων καθώς και η πρόβλεψη και αποφυγή εργατικών ατυχημάτων. [7]

3 ΜΙΚΡΟΕΛΕΓΚΤΗΣ ARDUINO

3.1 Πλακέτα Arduino

Σε μία πλακέτα Arduino είναι ενσωματωμένα ένας μικροεπεξεργαστής, μονάδες εισόδου/εξόδου καθώς και όλα τα απαραίτητα δομικά στοιχεία (όπως μονάδες αποθήκευσης, ρυθμιστές τάσης, ταλαντωτές) που την καθιστούν αυτόνομη και ευπροσάρμοστη σε ένα μεγάλο εύρος εφαρμογών. Ο προγραμματισμός του μικροελεγκτή πραγματοποιείται σε γλώσσα προγραμματισμού βασισμένη στο “wiring”, έχει πολλά κοινά σημεία με την C++, ενώ υπάρχουν αρκετές διαθέσιμες βιβλιοθήκες που ο χρήστης μπορεί να εισάγει διευκολύνοντας το έργο του. [11] [12]

Η πλακέτα μπορεί να χρησιμοποιηθεί αυτόνομα, είτε να αποτελέσει μέρος ενός πιο σύνθετου συστήματος, ενώ υπάρχει δυνατότητα επεκτάσεων των δυνατοτήτων της μέσω shields τα οποία προσαρμόζονται στην αρχική πλακέτα. [13]

3.1.1 Ιστορική αναδρομή

Οι μικροελεγκτές Arduino έκαναν την εμφάνισή τους στην αρχή του 21^{ου} αιώνα (2006) κυρίως με εκπαιδευτικό προσανατολισμό, καθώς στόχος ήταν η κατασκευή και παρασκευή σε ευρεία κλίμακα οικονομικών πλακετών για την υλοποίηση διαδραστικών σχεδίων από μαθητές, μη μηχανικούς και ερασιτέχνες. [11] [12]

Ιδρυτές της εταιρίας ήταν ο Μάσιμο Μπάντζι (Massimo Banzi) και ο Δαβίδ Κουαρτιέλ (David Cueartielles), όπου το 2005 ξεκίνησαν την παραγωγή πλακετών στην Κωμόπολη Ιβρέα της Ιταλίας. [11] [12]

3.1.2 Δομικά στοιχεία μιας πλακέτας Arduino

Υπάρχουν αρκετές εκδόσεις και παραλλαγές των πλακετών Arduino, προσαρμοσμένες η κάθε μία για διαφορετικούς σκοπούς. Όμως τα δομικά χαρακτηριστικά των περισσότερων εκδόσεων είναι κοινά και συνοπτικά είναι:

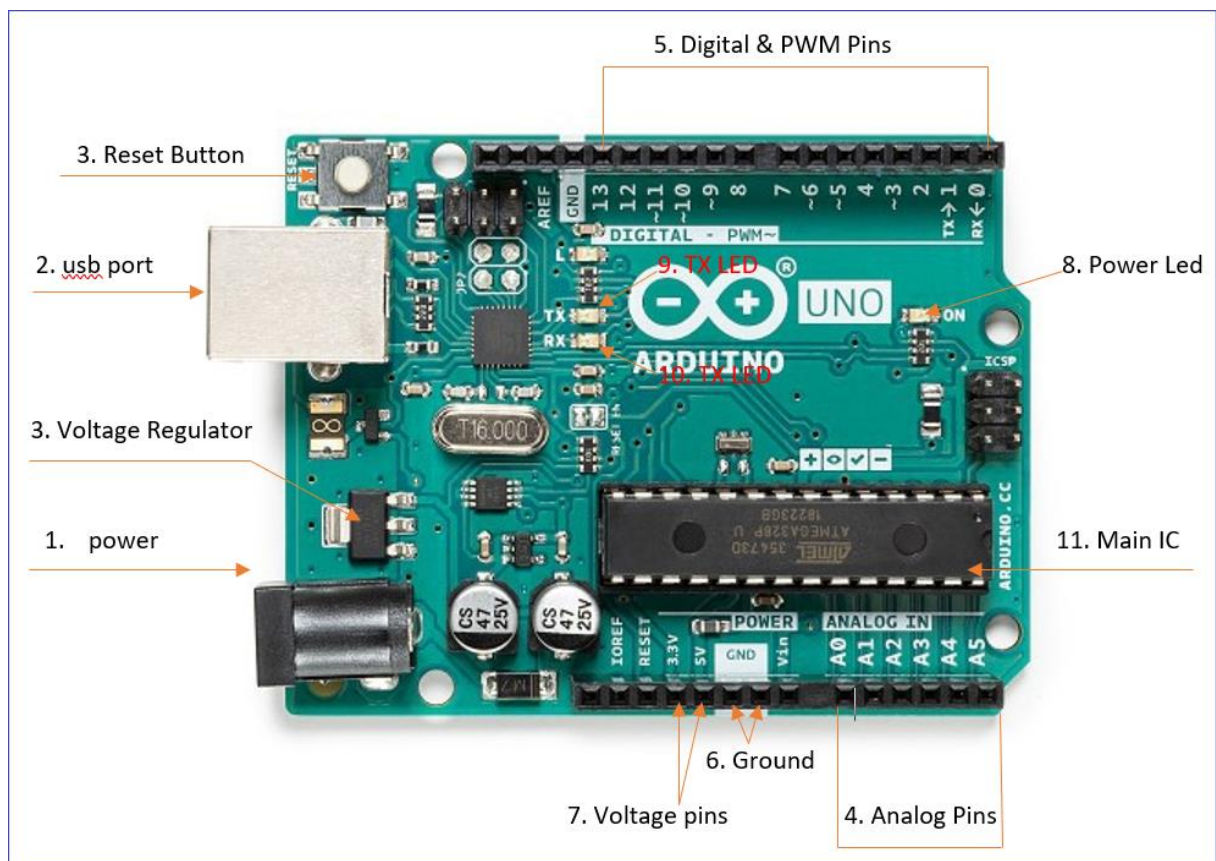
- Θύρα τροφοδοσίας και θύρα μεταφοράς δεδομένων: Όλες οι εκδόσεις πρέπει να έχουν μία θύρα τροφοδοσίας, έτσι ώστε η πλακέτα να μπορεί να τροφοδοτηθεί με ενέργεια για να λειτουργήσει. Αντίστοιχα πρέπει να έχουν και μια θύρα επικοινωνίας για να μπορέσει ο προγραμματιστής να της εισάγει τον

απαιτούμενο κώδικα. Η θύρα επικοινωνίας είναι συνήθως μια θύρα usb και μπορεί να χρησιμεύσει παράλληλα και σαν θύρα τροφοδοσίας. Συνεπώς η θύρα usb μπορεί να χρησιμεύσει και σαν θύρα μεταφοράς δεδομένων καθώς και σαν θύρα τροφοδοσίας της πλακέτας. Στην Εικόνα 3, στην οποία εικονίζεται η πλακέτα Arduino UNO, φαίνονται η θύρα τροφοδοσίας και η θύρα usb με τα νούμερα 1 και 2 αντίστοιχα.

- Κουμπί επανεκκίνησης ή reset button: Πιέζοντάς το αναγκάζει την πλακέτα να εκτελέσει από την αρχή τον τελευταίο εγκατεστημένο κώδικα – Νούμερο 3, Εικόνα 3.
- Υποδοχές, Θύρες ή Pins: Με τη σωστή συνδεσμολογία των θυρών επιτυγχάνεται η εύρυθμη λειτουργία του σχεδίου, όσο αφορά το κατασκευαστικό τμήμα του. Υπάρχουν διάφορες τύπου θύρες, οι οποίες φαίνονται στην Εικόνα 3 με αρίθμηση από το 4 έως και το 7, όπως θύρες εισόδου/εξόδου, ψηφιακές, αναλογικές, σταθερής τάσης και γείωσης. Συνοπτικά οι κυριότερες θύρες είναι:
 - ❖ Αναλογικές θύρες εισόδου (Analog pins): Οι θύρες αυτές χρησιμεύουν για να εισάγουμε αναλογικά σήματα, όπως η έξοδος από έναν αισθητήρα φωτός ή θερμοκρασίας.
 - ❖ Ψηφιακές θύρες εισόδου/εξόδου (Digital I/O pins): Αυτές οι θύρες μπορούν να χρησιμοποιηθούν είτε για την είσοδο ψηφιακών σημάτων (όπως το πάτημα ενός κουμπιού) είτε για την έξοδο ψηφιακών εντολών (όπως η τροφοδοσία ενός λαμπτήρα led).
Τέλος κάποιες ψηφιακές θύρες, στην Εικόνα 3 φαίνονται με το σήμα ~ πριν από τον αριθμό της θύρα, μπορούν να παράγουν ένα σήμα που να προσομοιάζει μία αναλογική έξοδο. Αυτό επιτυγχάνεται με τη χρήση μίας τεχνικής που ονομάζεται PWM (Pulse Width Modulation).
 - ❖ Υποδοχές γείωσης και τάσης (Ground pins, 5V pins, 3,3V Pins): Αυτές οι υποδοχές είναι απαραίτητες για την αδιάλειπτη τροφοδοσία που απαιτείται από τους αισθητήρες και τα άλλα εξαρτήματα για να λειτουργήσουν.
- Ενδεικτική λυχνία τάσης (Power led): Αυτή η λυχνία (8 Εικόνα 3) φωτοβολεί όταν η πλακέτα είναι συνδεδεμένη σε μία πηγή τάσης.
- Ενδεικτικές λυχνίες αποστολής και λήψης δεδομένων (TX, RX Leds): Αυτές οι λυχνίες (9, 10 Εικόνα 3) φωτοβολούν όταν ο Arduino «στέλνει» ή «λαμβάνει»

δεδομένα αντίστοιχα. Η ονομασία τους προέρχεται από τις αγγλικές λέξεις Transmit (TX) και Receive (RX).

- Μικροεπεξεργαστής (Main IC): Είναι η Κεντρική Μονάδα Επεξεργασίας (ΚΜΕ) μιας πλακέτας Arduino. Είναι ένα ολοκληρωμένο κύκλωμα (11 Εικόνα 3) το οποίο επεξεργάζεται όλα τα δεδομένα, εκτελεί τις λογικές και αριθμητικές πράξεις και ελέγχει τη ροή δεδομένων προς τις θύρες εισόδου/εξόδου.
- Ρυθμιστής τάσης (Voltage Regulator): Πρόκειται για το εξάρτημα το οποίο μετατρέπει την τάση τροφοδοσίας στην τάση που απαιτείται για τη σωστή λειτουργία της πλακέτας. Το πλεονέκτημα της χρήσης του ρυθμιστή τάσης είναι ότι η πλακέτα μπορεί να λειτουργήσει σε ένα εύρος τάσεων εισόδου, ενώ δεν είναι ευαίσθητη σε τυχόν μικροδιακυμάνσεις της τάσεως τροφοδοσίας. Επιτυγχάνεται δηλαδή αδιάλειπτη λειτουργία του μικροελεγκτή σε ένα εύρος τάσεων εισόδου. [11]



Εικόνα 3: Δομικά χαρακτηριστικά των πλακετών Arduino [vii]

3.1.3 Οικογένειες πλακετών Arduino και χαρακτηριστικά τους

Υπάρχουν πολλοί τύποι διαθέσιμων πλακετών Arduino, η κάθε μία με ιδιαίτερα χαρακτηριστικά που την καθιστούν περισσότερο ή λιγότερο κατάλληλη για τη χρήση της σε ένα σχέδιο με βάση τον μικροελεγκτή Arduino. Επιπλέον με την ευχέρεια που διαθέτει ένα προϊόν ανοικτού κώδικα, πολλοί κατασκευαστές έχουν «αντιγράψει» την πλακέτα προσδίδοντάς της διαφορετικά χαρακτηριστικά και δυνατότητες.

Όταν γίνεται λόγος για μια οικογένεια πλακετών Arduino, πρόκειται ουσιαστικά για πλακέτες της κατασκευάστριας εταιρίας Arduino οι οποίες στηρίζονται στην ίδια λογική κατασκευής, έχουν παρόμοια χαρακτηριστικά και εμφάνιση. Οι κυριότερες οικογένειες πλακετών, οι οποίες φαίνονται και στην Εικόνα 4 είναι:

- Arduino UNO: Αποτελεί την πιο εμπορικά επιτυχημένη οικογένεια πλακετών Arduino. Απευθύνεται κυρίως σε αρχάριους καθώς είναι κατασκευασμένη για να είναι φιλική προς το χρήστη, χωρίς όμως να σημαίνει ότι υστερεί σε δυνατότητες, καθώς σχεδόν όλα τα διαδραστικά σχέδια μπορούν να υλοποιηθούν χρησιμοποιώντας τη συγκεκριμένη πλακέτα.

Το Arduino UNO βασίζεται στον μικροεπεξεργαστή «ATMEGA328», ενώ διαθέτει μία θύρα USB για τη σύνδεση με άλλη συσκευή, δεκατέσσερις (14) ψηφιακές θύρες εισόδου/εξόδου και έξι (6) αναλογικές θύρες εισόδου. Η τάση τροφοδοσίας του είναι από 6-20 Volts, ενώ η τάση λειτουργίας του είναι τα 5 volt.

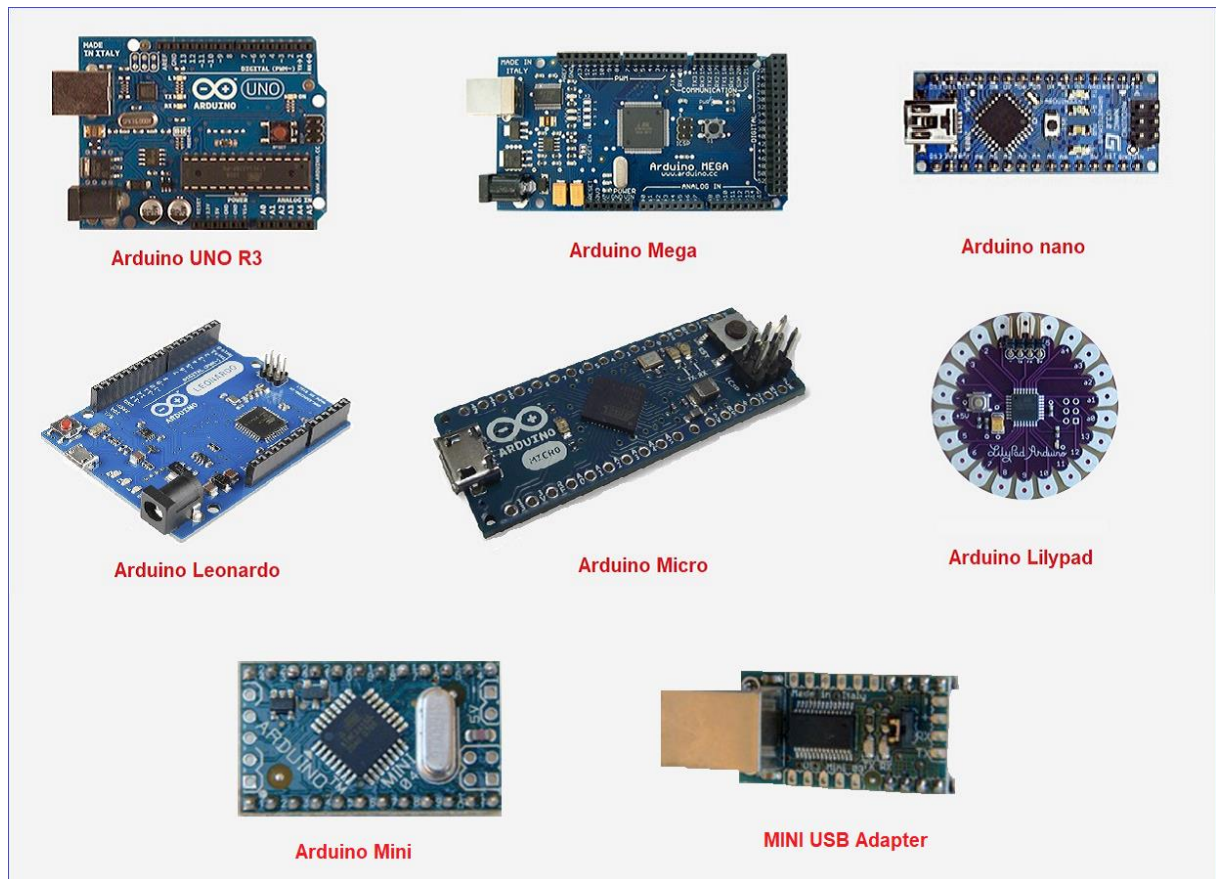
- Arduino LilyPad: Από σχεδιαστικής άποψης αποτελεί την πιο ευφάνταστη πλακέτα. Σχεδιάστηκε με γνώμονα τις απαιτήσεις που υπάρχουν στην υλοποίηση σχεδίων που αφορούν ρουχισμό. Συνεπώς η οικογένεια αυτή των πλακετών μπορεί να ραφτεί πάνω σε ρούχα, να χρησιμοποιεί ως διαύλους επικοινωνίας αγωγίμες κλωστές, να μπορεί να «πλυθεί» και ως τροφοδοσία να χρησιμοποιεί επαναφορτιζόμενες μπαταρίες.

Με την εξέλιξη της οικογένειας αυτής προστέθηκαν ολοένα και περισσότερα χαρακτηριστικά, όπως περισσότερες θύρες εισόδου/εξόδου, όμως σε όλες τις εκδόσεις χρησιμοποιήθηκαν μικροεπεξεργαστές της σειράς ATMEGA.

- Arduino Mega: Έχει παρόμοια δομικά χαρακτηριστικά με τον Arduino uno, αλλά, όπως δηλώνει και το όνομά του, έχει μεγαλύτερη υπολογιστική ικανότητα ενώ διαθέτει και αρκετές περισσότερες θύρες εισόδου/εξόδου

δίνοντας στον κατασκευαστή τη δυνατότητα να υλοποιήσει πιο εκτεταμένα διαδραστικά σχέδια. Ενδεικτικά αναφέρεται πως ο Arduino Mega 2560 διαθέτει πενήντα τέσσερις (54) ψηφιακές και δεκαέξι (16) αναλογικές θύρες. Αυτή η σειρά χρησιμοποιεί μικροεπεξεργαστές ATMEGA1280 για τα παλαιότερά της μοντέλα, ενώ για τα πιο σύγχρονα ATMEGA2560.

- Arduino Leonardo: Πρόκειται για την πρώτη πλακέτα που ο μικροεπεξεργαστής της μπορεί να διαχειριστεί απευθείας μια σύνδεση usb, δίνοντάς του το πλεονέκτημα να είναι κατασκευαστικά πιο οικονομικός και απλός. Παράλληλα αυτή η δυνατότητα της άμεσης διαχείρισης μιας σύνδεσης usb, του δίνει το πλεονέκτημα να μπορεί να διαχειριστεί συσκευές όπως το πληκτρολόγιο και το ποντίκι ενός Ηλεκτρονικού Υπολογιστή.
- Arduino mini: Πρόκειται για την πιο μικρή σε μέγεθος εμπορικά διαθέσιμη πλακέτα που έχει κυκλοφορήσει. Για να επιτευχθεί όσο το δυνατόν μεγαλύτερη εξοικονόμηση χώρου, αυτές οι εκδόσεις διατίθενται με μόνο έναν ρυθμιστή τάσης, (στα 5 ή στα 3,3 V), ενώ δεν διαθέτουν ενσωματωμένη θύρα usb. Συνεπώς ο προγραμματιστής για να εισάγει τον απαιτούμενο κώδικα στην πλακέτα θα πρέπει να διαθέτει έναν επιπλέον μετατροπέα από usb σε σειριακό τύπου TTL (Transistor-Transistor Logic) (Usb to TTL Serial converter).
- Arduino nano, Arduino micro: Στην ουσία αποτελούν σμικρύνσεις με μικρές παραλλαγές των πλακετών Arduino Uno και Arduino Leonardo αντίστοιχα. Αυτές οι οικογένειες πλακετών συνδυάζουν το πλεονέκτημα του συγκριτικά μικρού τους όγκου, με τις ανέσεις που παρέχουν οι μεγαλύτερες πλακέτες Arduino όπως η ενσωματωμένη θύρα usb για μεταφορά δεδομένων. [11] [14]



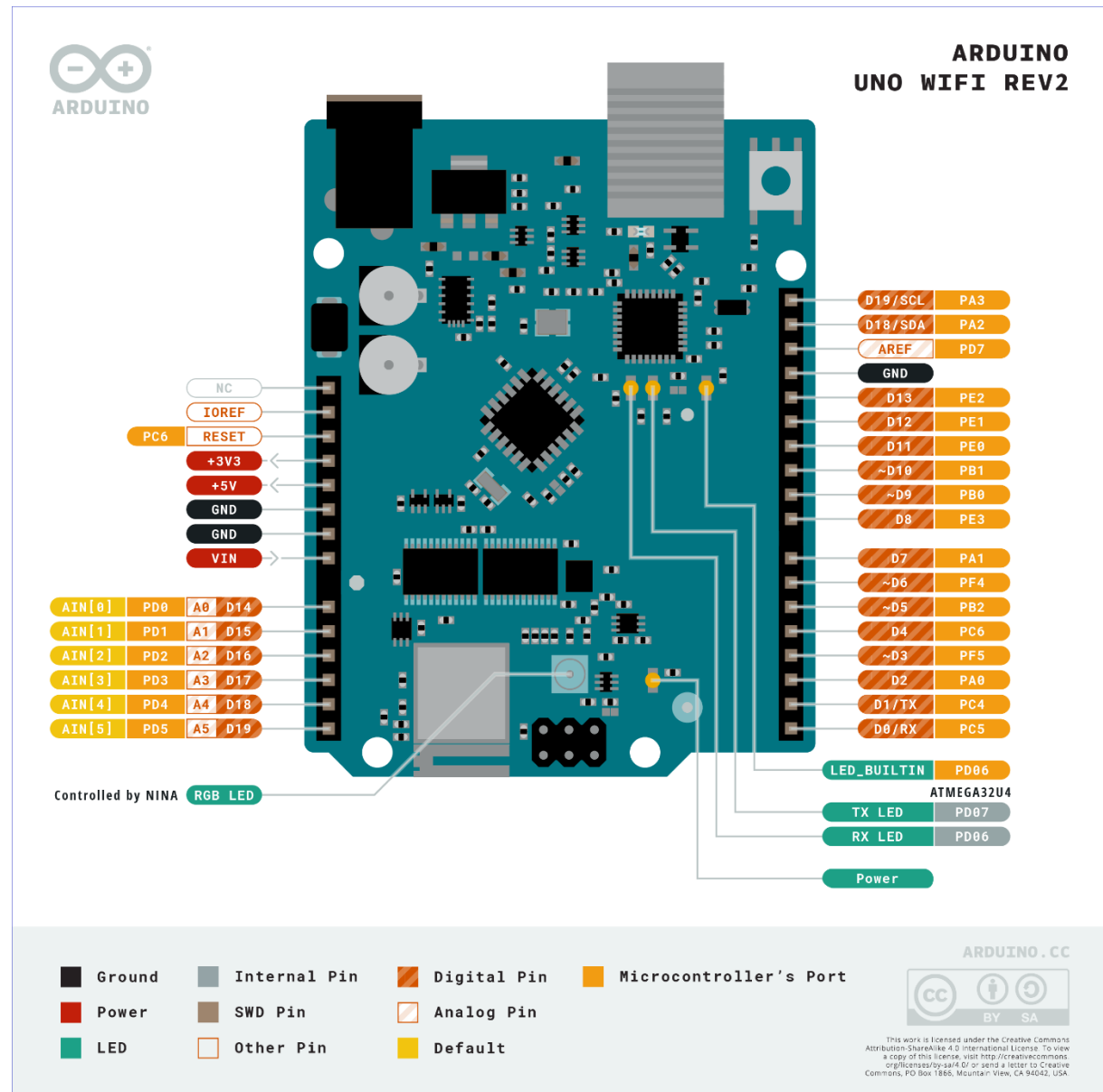
Εικόνα 4: Χαρακτηριστικές πλακέτες οικογενειών Arduino [ix]

3.2 Χαρακτηριστικά της πλακέτας Arduino UNO R2

Η πλακέτα Arduino UNO Wifi Rev.2 ανήκει στην οικογένεια πλακετών Arduino UNO. Το ιδιαίτερο χαρακτηριστικό της συγκεκριμένης πλακέτας είναι ότι διαθέτει μία ενσωματωμένη λειτουργική μονάδα WIFI. Συνοπτικά η πλακέτα είναι βασισμένη στον μικροεπεξεργαστή MEGA 4809, ενώ το ενσωματωμένο κύκλωμα WIFI είναι το ESP32, NINA-W13.

Χρησιμοποιώντας τη συγκεκριμένη πλακέτα, ο χρήστης είναι σε θέση να επικοινωνήσει με άλλες συσκευές χρησιμοποιώντας το δίκτυο WIFI, να συνδεθεί σε ένα δίκτυο WIFI, ή ακόμη και να δημιουργήσει το δικό του τοπικό ασύρματο δίκτυο όπου άλλες συσκευές θα συνδεθούν μεταξύ τους. Διαθέτει δεκατέσσερις (14) ψηφιακές θύρες, εκ' των οποίων οι πέντε (5) αποτελούν θύρες PWM, καθώς και έξι (6) αναλογικές θύρες. Οι αναλογικές θύρες είναι της τάξεως των δέκα Bytes η κάθε μία, και έχουν διακριτική ικανότητα χιλίων είκοσι τεσσάρων σταθμών (0-1023).

Το πλήρες λειτουργικό διάγραμμα της πλακέτας φαίνεται στην Εικόνα 5, ενώ ένας συνοπτικός πίνακας των χαρακτηριστικών του παρουσιάζεται στον Πίνακα 3. [5] [9] [15] [16] [17]



Εικόνα 5: Διάγραμμα θυρών (Pins) του “Arduino UNO R2”. [x]

Μικροεπεξεργαστής	ATmega4809
Τάση λειτουργίας	5 V
Τάση εισόδου (προτεινόμενη)	7-12 V

Ψηφιακές θύρες εισόδου/εξόδου	14 – εκ των οποίων 5 PWM
Αναλογικές θύρες εισόδου	6
Ρεύμα D/C ανά θύρα εισόδου/εξόδου	20 mA
Ρεύμα D/C για την θύρα 3.3 V	50 mA
Μέγιστο ολικό ρεύμα D/C	200 mA
Flash memory	48 KB (ATMega4809)
SRAM	6.144 Bytes (ATMega4809)
EEPROM	256 Bytes (ATMega4809)
Clock speed	16 MHz
Radio Module	u-blox NINA-W102
Secure Element	ATECC608A
Inertial Measurement Unit	LSM6DS3TR
LED_BUILTIN	25

Πίνακας 3: Τεχνικά χαρακτηριστικά Arduino UNO R2 [18]

3.3 Αναφορά στους μικροελεγκτές

Ο μικροελεγκτής είναι ένας επεξεργαστής, που διαθέτει πολλά ενσωματωμένα υποσυστήματα και αυτό μας βοηθά στο να μη χρειάζεται να προσθέσουμε περαιτέρω εξαρτήματα. Θα το συναντήσουμε συχνά σε όλα τα ενσωματωμένα συστήματα, είτε σε αυτοματισμούς, είτε σε ηλεκτρικές συσκευές, είτε σε αυτόνομα αυτοκίνητα κ.α. Ενδεικτικά μερικά πλεονεκτήματα είναι:

- Ενσωματωμένη μνήμη.
- Χαμηλό κόστος.
- Μικρό μέγεθος.
- Πολλές θύρες και διαθέσιμοι ακροδέκτες.

- Χαμηλή κατανάλωση ισχύος.

Στο εμπόριο θα συναντήσουμε αρκετών ειδών μικροελεγκτές, δηλαδή 8,16 και 32 bit με χαμηλό κόστος, είναι γενικής χρήσης και όσο ανεβαίνουν τα bit ανεβαίνουν και οι δυνατότητες. Όλοι οι μικροελεγκτές διαθέτουν δικά τους εργαλεία ανάπτυξης, όπως μεταγλωττιστές οι οποίοι μετατρέπουν μια γλώσσα υψηλού επιπέδου σε γλώσσα μηχανής, αλλά και εργαλεία εκσφαλμάτωσης ή αλλιώς debuggers. Η C, C++ και κάποιες παραλλαγές τους αποτελούν τις βασικές γλώσσες προγραμματισμού ενός μικροελεγκτή. [2]

3.4 Αισθητήρες και Shields

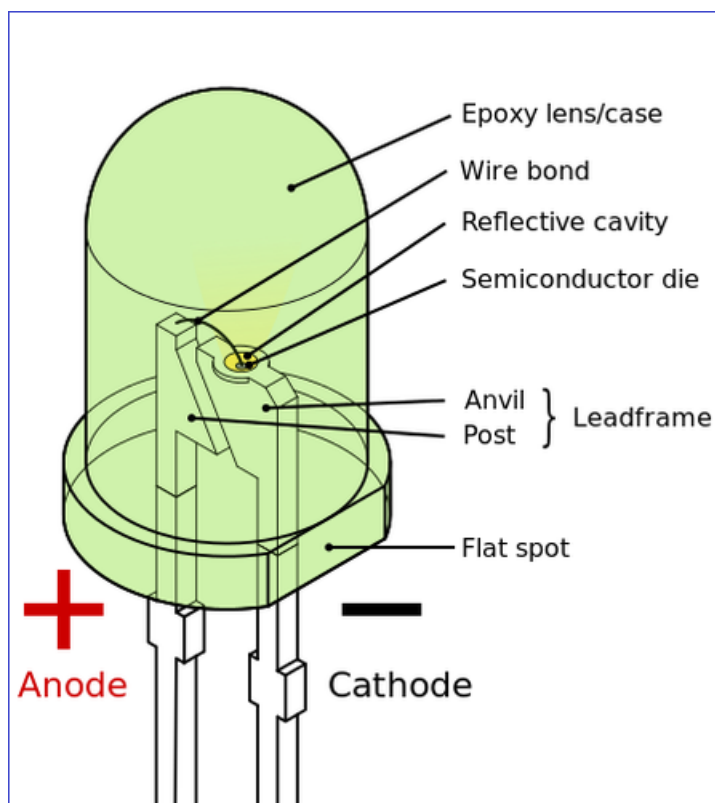
Για να υλοποιηθεί μια συγκεκριμένη εργασία, εκτός από τον Arduino, είναι απαραίτητο να χρησιμοποιηθούν και αισθητήρες για να μπορεί ο μικροελεγκτής να λαμβάνει δεδομένα. Ένας αισθητήρας μπορεί να είναι από μια απλή επαφή με δύο καταστάσεις (όπως για παράδειγμα ένας ελεγκτής στάθμης ρευστού) ή ένα πιο περίπλοκο σύστημα, το οποίο συχνά έχει αναλογική έξοδο.

Για την αποφυγή των δυσκολιών που παρουσιάζονται με την απευθείας χρήση των αισθητήρων, έχουν αναπτυχθεί πακέτα ή “Shields” τα οποία ουσιαστικά λειτουργούν ως μετατροπείς και προσαρμόζουν την έξοδο του αισθητήρα στις απαιτήσεις του Arduino, διευκολύνοντας και απλοποιώντας με αυτό τον τρόπο την κατασκευή διάφορων projects. Φυσικά η τεχνολογία των Shields δεν περιορίζεται μόνο σε προσαρμογή των δεδομένων των αισθητήρων, αλλά έχει αναπτυχθεί για να καλύπτουν τις περισσότερες απαιτήσεις που υπάρχουν, όπως για παράδειγμα ένας gsm shield μπορεί να προσαρμόσει μία κάρτα sim σε μια πλακέτα Arduino. [13] [19]

3.4.1 Φωτοδίοδος (Led)

Μία φωτοδίοδος led διαθέτει μία απλή επαφή θετικών και αρνητικών φορτίων. Με βάση τον κανόνα των ομώνυμων και ετερόνυμων, αν εφαρμόσουμε σε μία θετική άκρη με πολλά αρνητικά φορτία μία θετική τάση, τότε το ρεύμα θα περάσει. Στην αντίθετη περίπτωση όμως, αν εφαρμόσουμε μια αρνητική τάση, το ρεύμα δε θα περάσει και αυτό έχει ως αποτέλεσμα να μη λειτουργήσει. Σε περίπτωση που εφαρμόσουμε μεγάλη ανάστροφη τάση στα άκρα της υπάρχει κίνδυνος να καταστραφεί. Δηλαδή μία φωτοδίοδος είναι αυτό ακριβώς που δηλώνει το όνομά της.

Το ρεύμα μπορεί να διέρχεται μόνο όταν έχει τη σωστή πολικότητα, προκαλώντας τη να φωτοβολεί, ενώ δεν μπορεί να διέλθει ανάποδα. Στην Εικόνα 6 φαίνονται σχηματικά τα κατασκευαστικά στοιχεία μιας φωτοδιόδου, όπως το προστατευτικό της κάλυμμα, η επιφάνεια ανάκλασης, η κάθοδος (συνδέεται στο πλιν της τάσης) και η άνοδος (συνδέεται στο συν της τάσης τροφοδοσίας).



Εικόνα 6: Δομικά στοιχεία φωτοδιόδου. [iii]

Ανάλογα με την κατασκευή της και το μήκος κύματος που πρόκειται να εκπέμψει, έχουμε και διάφορα χρώματα (λευκό, κόκκινο, πράσινο, μπλε κ.α.). [20] [21] [22] [23]

3.4.2 Αισθητήρας κίνησης

Η κύρια λειτουργία του είναι, αφού ανιχνεύσει τη κίνηση σε κάποιο χώρο που έχει τοποθετηθεί, να κάνει την αντίστοιχη λειτουργία που του έχει ζητηθεί. Για παράδειγμα θα μπορούσε να ανάψει τα φώτα σε κάποιο δωμάτιο και να τα διατηρήσει ανάμενα για κάποιο χρονικό διάστημα, όσο δηλαδή ανιχνεύει τη κίνηση. Έπειτα θα τα κλείνει, αφού δε θα υπάρχει κίνηση στο χώρο. Τέτοιους ανιχνευτές θα βρούμε πολλούς στο εμπόριο, είτε για εξωτερικούς, είτε για εσωτερικούς χώρους. Οι αισθητήρες αυτοί, διακρίνονται σε κάποιες τεχνολογίες όπως PIR (passive infrared) και US (ultrasonic).

- Τεχνολογία PIR (passive infrared): Η τεχνολογία αυτή χρησιμοποιεί αισθητήρες που αντιδρούν σε υπέρυθρες πηγές θερμοκρασίας, όπως αυτές που παράγει το σώμα του ανθρώπου. Έτσι, στη πράξη όταν λάβει μία τέτοια ακτινοβολία λειτουργεί ως διακόπτης και ενεργοποιεί το κύκλωμα μίας συσκευής. Στην Εικόνα 7 φαίνεται αυτού του τύπου ο αισθητήρας, όπου λόγου κόστους είναι και αυτός που χρησιμοποιείται πιο συχνά σε project χαμηλού κόστους.



Εικόνα 7: Αισθητήρας κίνησης τύπου PIR [iv]

- Έχει εμβέλεια από 8m έως 45m, με συνήθη τιμή τα 15m.
- Έχει γωνία κάλυψης μεταξύ 900 και 1450.
- Τα PIR τοποθετούνται πάνω σε τοίχους ή άλλες σταθερές επιφάνειες που είναι κάθετες στο έδαφος και σε απόσταση περίπου 2,20 – 2,50 m από αυτό.

Τα πλεονεκτήματά του είναι:

- Χαμηλό κόστος και εύκολη εγκατάσταση.
- Μπορούν να συνυπάρξουν στον ίδιο χώρο πάνω από 2 αισθητήρες, χωρίς κανένα πρόβλημα.
- Είναι αρκετά αξιόπιστοι και λειτουργούν αποτελεσματικά, δηλαδή δε θα ανιχνεύσουν μια κίνηση πίσω από ένα τοίχο ή κάτι άλλο πέραν του χώρου που είναι τοποθετημένοι.

Το μειονέκτημά τους είναι ότι δε μπορούν να ανταπεξέλθουν το ίδιο καλά σε υψηλές θερμοκρασίες.

- Τεχνολογία US (ultrasonic): Ο ανιχνευτής εκπέμπει υπέρηχους, τους οποίους λαμβάνει πίσω μετά από ανάκλασή τους στα αντικείμενα του χώρου. Αν αλλάξει κάτι στο χώρο θα το καταλάβει αφού και η συχνότητα υπερήχων θα αλλάξει εκ νέου. Η τεχνολογία αυτή είναι αρκετά ευαίσθητη αφού μπορεί να

λειτουργήσει αρκετά αποτελεσματικά σε χώρους με εμπόδια και σε χώρους με χαμηλή κινητικότητα. Προσοχή, οι US αισθητήρες δε πρέπει να τοποθετούνται σε επιφάνειες που δονούνται, σε εξωτερικούς χώρους, σε σημεία που έχει αέρα, όπως για παράδειγμα στον κήπο ή στη πυλωτή μιας πολυκατοικίας, γιατί με την παραμικρή κίνηση θα ενεργοποιείται.

- Διπλή τεχνολογία (PIR/US): Οι ανιχνευτές αυτοί διαθέτουν και τις δύο τεχνολογίες που προαναφέραμε, κάτι το οποίο κάνει τη συσκευή πιο αποτελεσματική σε όλες τις συνθήκες. [24] [25] [26]

3.4.3 Αισθητήρας φωτός

Οι αισθητήρες φωτός αναφέρονται ως τύποι φωτοανιχνευτών ή φωτοαισθητήρων. Οι αισθητήρες φωτός ανιχνεύουν την πυκνότητα του φωτός αλλά δεν καταγράφουν εικόνες Στην Εικόνα 8 φαίνεται παράδειγμα ενός τέτοιου αισθητήρα.



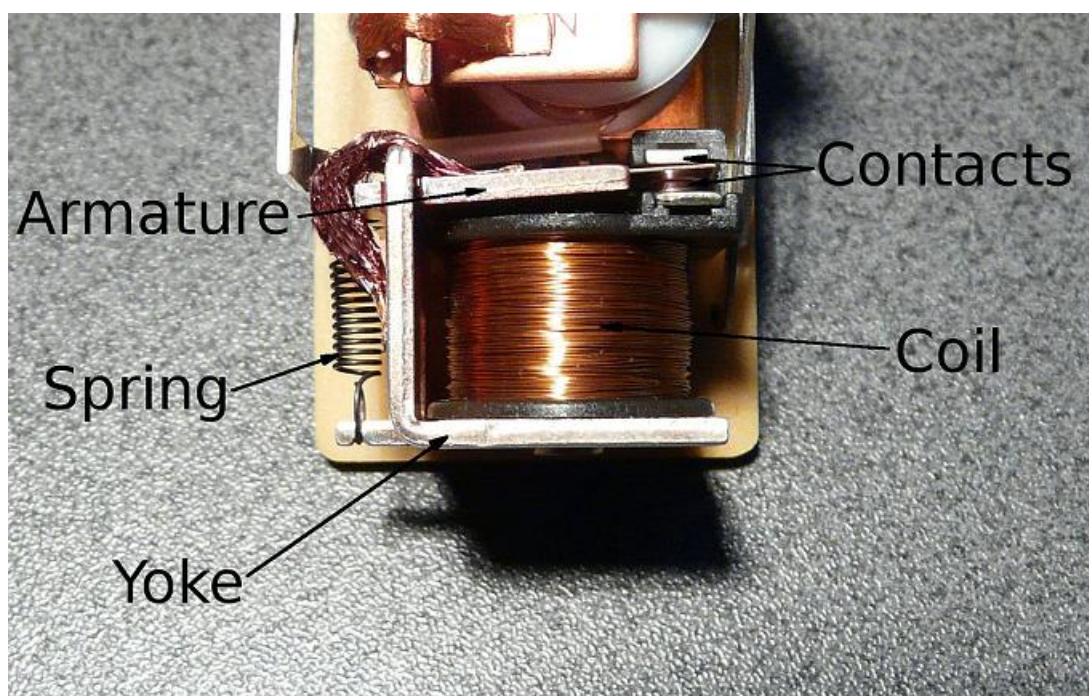
Εικόνα 8: Φωτοαντίσταση. [v]

Οι πιο συνηθισμένοι αισθητήρες φωτεινής πυκνότητας είναι:

- LDR Light Dependent Resistors: Είναι ικανοί να μετατρέπουν το φως είτε σε ρεύμα είτε σε τάση.
- Φωτοαντίσταση: Είναι στην πραγματικότητα μία μεταβαλλόμενη αντίσταση η οποία μειώνεται όσο αυξάνεται η ένταση του φωτός που πέφτει πάνω της. [27]

3.4.4 Relay

Ο ηλεκτρονόμος ή αλλιώς ρελές είναι ένας ηλεκτρικός διακόπτης που ανοίγει και κλείνει ένα ηλεκτρικό κύκλωμα. Η λειτουργία του ρελέ εξαρτάται από τις επαφές του, δηλαδή με το άνοιγμα ή κλείσιμο μιας ή περισσότερων επαφών από έναν ηλεκτρομαγνήτη, λειτουργεί σα διακόπτης. Κάθε επαφή ενός ηλεκτρονόμου μπορεί να είναι Κανονικά-Ανοικτή (Normally Open, NO), Κανονικά-Κλειστή (Normally Closed, NC) ή μεταγωγικός (change-over), ανάλογα με τον τύπο της. Στην παρακάτω Εικόνα 9 φαίνονται τα δομικά στοιχεία ενός Ρελέ, όπως οι επαφές του (NC, NO), το πηνίο του (ηλεκτρομαγνήτης) ο οποίος «τραβάει» την επαφή σε θέση NO, όταν διέρχεται από ρεύμα, και το ελατήριο επαναφοράς το οποίο «σπρώχνει» την επαφή ξανά στη θέση NC όταν σταματήσει να διέρχεται ρεύμα από το πηνίο.



Εικόνα 9: Δομικά στοιχεία ενός Ρελέ. [vi]

Στην εργασία χρησιμοποιήθηκε ένας ρελές 5 volt, παρόμοιος με αυτόν της Εικόνα 10, ο οποίος ελέγχει συσκευές με τάση έως 250V. Έχει τρία τερματικά (NC, C, and NO), στα οποία συνδέεις πάνω τη συσκευή που θες για να συνδεθεί με το Arduino.

5V Relay Terminals and Pins



Εικόνα 10: Relay 5 V. [xi]

- NC: Normally closed 120-240V terminal
- NO: Normally open 120-240V terminal
- C: Common terminal
- Ground: Connects to the ground pin on the Arduino
- 5V Vcc: Connects the Arduino's 5V pin
- Signal: Carries the trigger signal from the Arduino that activates the relay

Κανονικά-Ανοικτή επαφή (NO), φαίνεται στην Εικόνα 11, συνδέει το κύκλωμα όταν ο ηλεκτρονόμος ενεργοποιείται ενώ το κύκλωμα αποσυνδέεται όταν ο ηλεκτρονόμος είναι ανενεργός.

Κανονικά-Κλειστή επαφή (NC), φαίνεται στην Εικόνα 12, αποσυνδέει το κύκλωμα όταν ο ηλεκτρονόμος ενεργοποιείται, ενώ το κύκλωμα συνδέεται όταν ο ηλεκτρονόμος είναι ανενεργός. [28] [29] [30]



Εικόνα 11: Relay, επαφή NO. [xi]



Εικόνα 12: Relay, επαφή NC. [xi]

4 ΥΛΟΠΟΙΗΣΗ PROJECT

4.1 Συνοπτική περιγραφή Project

Σκοπός της παρούσας εργασίας είναι η υλοποίηση ενός διαδραστικού σχεδίου όπου ο χρήστης θα είναι σε θέση να χειρίζεται μία φωτεινή λυχνία από απόσταση με το προσωπικό του smartphone. Ο χειρισμός της φωτεινής λυχνίας πραγματοποιείται μέσω του μικροελεγκτή Arduino Uno R2, ο οποίος περιγραφικέ αναλυτικά στο προηγούμενο κεφάλαιο, ενώ η επικοινωνία του smartphone και του μικροελεγκτή θα πραγματοποιείται μέσω ενός τοπικού δικτύου.

Αναλυτικότερα, ο χρήστης θα είναι σε θέση να χειρίζεται τη λάμπα απομακρυσμένα όπως ένας διακόπτης, δηλαδή να την ανάβει και να τη σβήνει. Επιπλέον ο μικροελεγκτής θα λαμβάνει και σήματα κατάστασης από δύο αισθητήρια, ένα αισθητήριο κίνησης και ένα αισθητήριο φωτεινότητας, επομένως ο χρήστης θα έχει την επιπλέον δυνατότητα να θέσει τη λειτουργία της λάμπας στηριζόμενος στους αισθητήρες αυτούς. Για παράδειγμα να θέσει τη λειτουργία της λάμπας από τον αισθητήρα φωτός, οπότε η λάμπα να φωτοβολεί τις βραδινές ώρες.

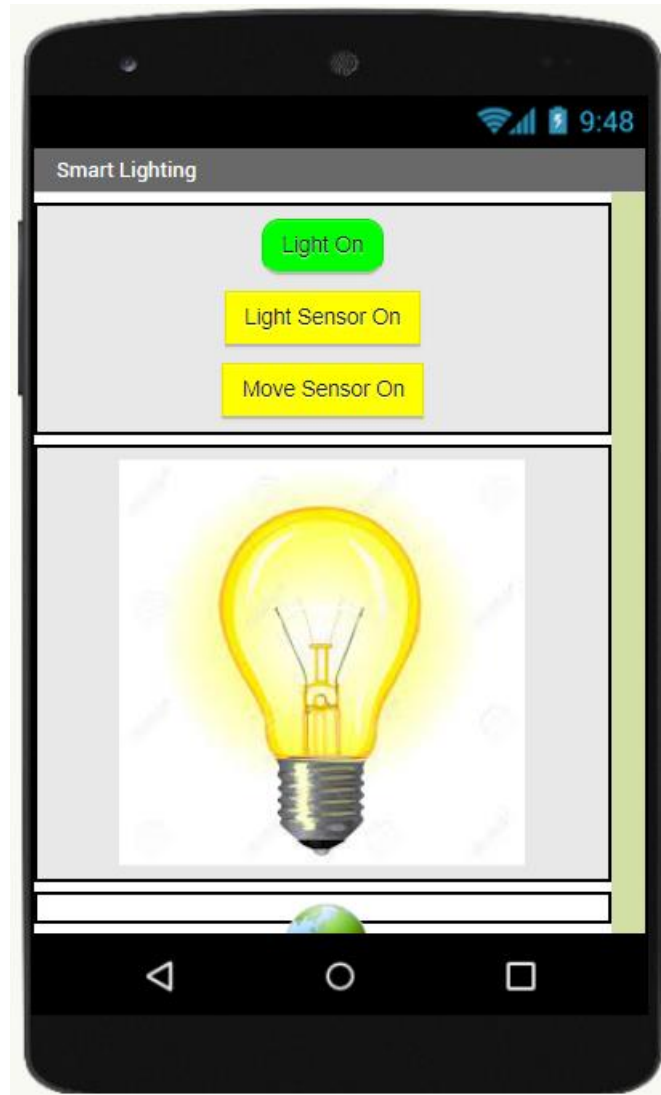
Προγραμματιστικά χρησιμοποιήθηκαν δύο (2) προγραμματιστικά περιβάλλοντα για την επίτευξη του συγκεκριμένου σχεδίου. Το πρώτο προγραμματιστικό περιβάλλον που χρησιμοποιήθηκε είναι το Arduino IDE και χρησιμοποιήθηκε στην υλοποίηση του κώδικα για τον Arduino Uno R2, ενώ το δεύτερο περιβάλλον είναι το MIT App Inventor και χρησιμοποιήθηκε για την υλοποίηση της εφαρμογής (app) για το smartphone.

4.2 Προγραμματιστικό περιβάλλον MIT APP INVENTOR

Η επιλογή του συγκεκριμένου προγραμματιστικού περιβάλλοντος έγινε διότι είναι αρκετά εύχρηστο, ευπροσάρμοστο, φιλικό προς το χρήστη, διαχωρίζονται η οπτική υλοποίηση από το προγραμματιστικό τμήμα, ενώ έχει διευρυμένες δυνατότητες κυρίως ως προς την οπτική υλοποίηση του σχεδίου. Δομικά πρόκειται για ένα προγραμματιστικό περιβάλλον, όπου κάθε στοιχείο που εισάγεται συμπεριφέρεται σαν αντικείμενο.

4.2.1 Οπτική υλοποίηση εφαρμογής

Στην Εικόνα 13 φαίνεται ο οπτικός σχεδιασμός, δηλαδή πως θα φαίνεται στο κινητό του χρήστη.



Εικόνα 13: Οπτική σχεδίαση της εφαρμογής για smartphone

Τα κύρια στοιχεία της εφαρμογής είναι:

- Button εντολών. Υπάρχουν τρία Button.

Το πρώτο- “GenButton” - ελέγχει απευθείας τη λάμπα, δηλαδή τη θέτει σε θέσεις “on”, “off”, και έλεγχος από αισθητήρες. Η εναλλαγή μεταξύ των καταστάσεων γίνεται διαδοχικά, ενώ αλλάζει ταυτόχρονα και το κείμενο μέσα στο πλαίσιο του button καθώς και το χρώμα του.

Τα άλλα δύο Button- “MoveButton”, “LightButton” – ελέγχουν τους αισθητήρες κίνησης και φωτός αντίστοιχα, δηλαδή αν η λάμπα θα πρέπει να

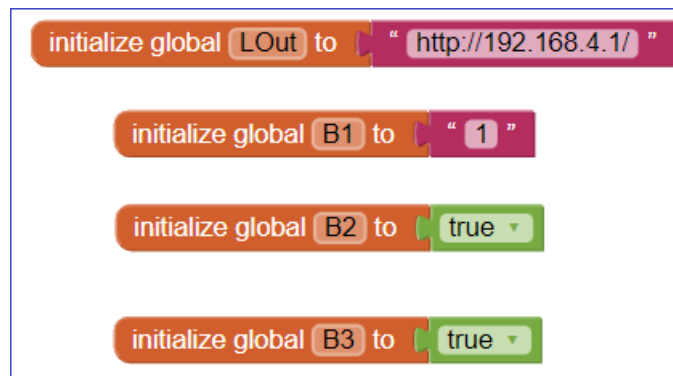
ανάψει ή όχι με βάση τους αισθητήρες. Για παράδειγμα αν ο αισθητήρας φωτός έχει τεθεί σε θέση “off”, τότε η λάμπα θα πρέπει να παραμείνει σβηστεί ακόμη και στο απόλυτο σκοτάδι.

- Η εικόνα: Το επόμενο στοιχείο της εφαρμογής είναι η εικόνα της λάμπας. Η εικόνα εναλλάσσεται ανάλογα με την κατάσταση του πρώτου “button”.
- Στοιχείο επικοινωνίας: Το τρίτο στοιχείο είναι το στοιχείο επικοινωνίας ή ένας “webViewer”. Αυτό το στοιχείο είναι ο διαμεσολαβητής που μεταδίδει και λαμβάνει πληροφορίες με τον μικροελεγκτή.

4.2.2 Προγραμματισμός της εφαρμογής

Στη συνέχεια, αφού ολοκληρώθηκε η οπτική απεικόνιση της εφαρμογής, θα αναλυθούν όλα τα στοιχεία και οι επιμέρους κώδικες που τη συνθέτουν ώστε να λειτουργήσει και να επικοινωνεί σωστά με τον μικροελεγκτή.

Δήλωση μεταβλητών:



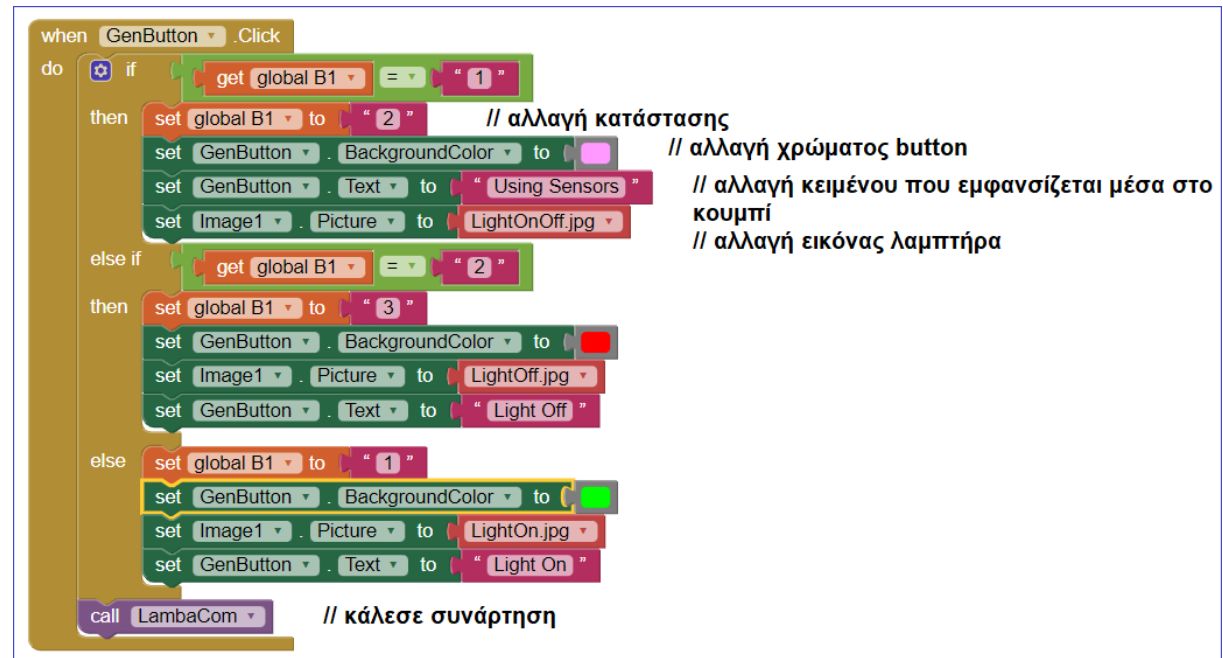
Εικόνα 14: Δήλωση μεταβλητών

Στην παραπάνω εικόνα (Εικόνα 14) φαίνεται η δήλωση των μεταβλητών που θα χρησιμοποιηθούν για την υλοποίηση της εφαρμογής. Ουσιαστικά πρόκειται για τη διεύθυνση (IP) του μικροελεγκτή, στην οποία θα αποστέλλονται τα δεδομένα της εφαρμογής, και η κατάσταση των Button. Η μεταβλητή “B1” αντιστοιχεί στην κατάσταση του πρώτου Button, (GenButton) ενώ οι μεταβλητές “B2” και “B3” στα button που ελέγχουν τη λειτουργία των αισθητήρων.

Όπως είναι αναμενόμενο, οι πιθανές καταστάσεις της μεταβλητής “B1” είναι τρεις. Η κατάσταση «1» αναφέρεται στην περίπτωση που ο χρήστης θέλει η λάμπα να είναι μόνιμα αναμμένη. Η κατάσταση «3», υποδηλώνει ότι ο χρήστης έχει επιλέξει η λάμπα

να παραμείνει μόνιμα σβηστή, ενώ στην κατάσταση «2» ότι επιθυμεί να ελέγχεται από τους αισθητήρες.

Button – GenButton:



Εικόνα 15: Εισαγωγή κώδικα (General Button)

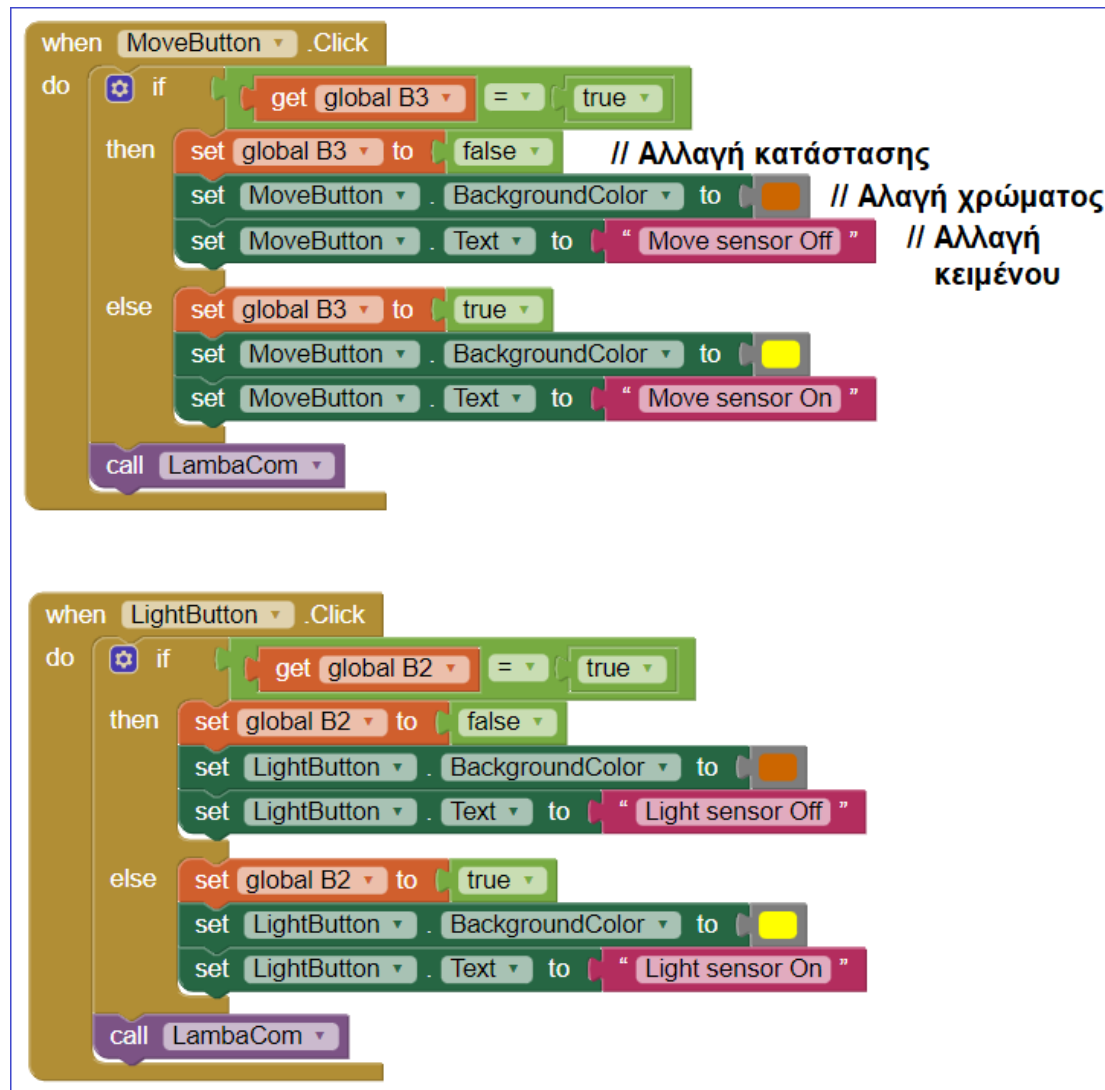


Εικόνα 16: Φωτογραφίες Λάμπας. (a) $B1=1$, (b) $B1=2$, (c) $B1=3$

Κάθε φορά που πιέζεται το αντίστοιχο κουμπί, αλλάζει η κατάστασή του, δηλαδή από την κατάσταση «1» μεταβαίνει στην κατάσταση «2» από την «2» στην «3» και από την «3» πίσω στην «1». Με κάθε εναλλαγή πρέπει να αλλάζουν το χρώμα του, το κείμενό του και η εικόνα που εμφανίζεται στο κάτω μέρος της οθόνης, ώστε να συμφωνούν με την παρούσα κατάστασή του. Οι εντολές που εκτελούνται κάθε φορά που πιέζεται το συγκεκριμένο button φαίνονται στην Εικόνα 15, ενώ τα αντίστοιχα εικονίδια μεταξύ των οποίων γίνεται η εναλλαγή αναλόγως της κατάστασης της μεταβλητής “B1” φαίνεται στην Εικόνα 16.

Τέλος, αφού γίνουν οι απαραίτητες οπτικές αλλαγές, καλείται η συνάρτηση “LambaCom”, η οποία θα αναλυθεί παρακάτω και η οποία είναι υπεύθυνη για να αποστείλει τα δεδομένα και τις αλλαγές στον μικροελεκτή.

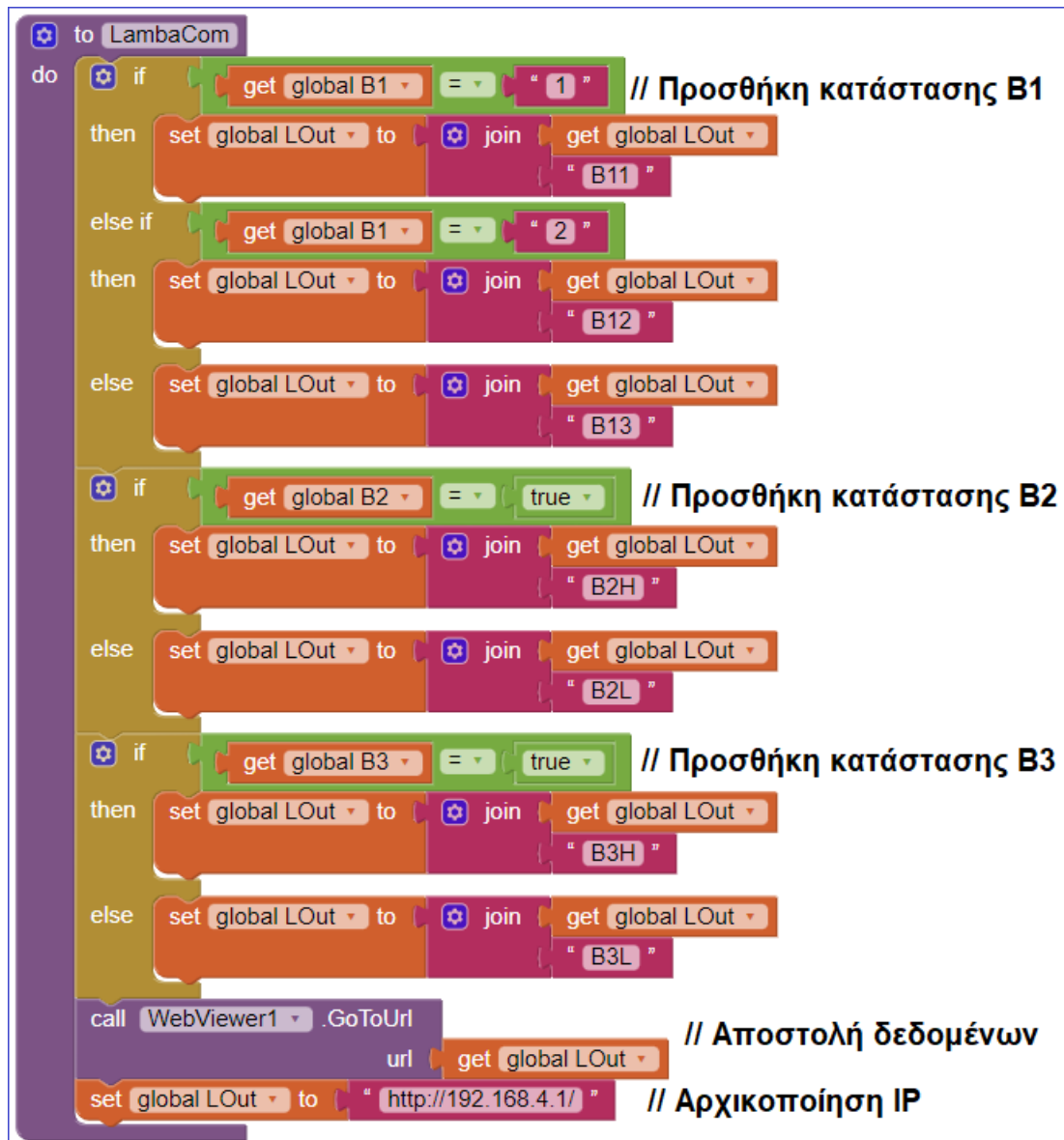
Buttons – LightButtonκαι MoveButton:



Εικόνα 17: Εισαγωγή κώδικα (MoveButton, LightButton)

Αντίστοιχη διαδικασία, όπως φαίνεται και στην Εικόνα 17, ακολουθείται για το πάτημα των άλλων δύο button, με τη μόνη διαφορά ότι αυτά εναλλάσσονται μεταξύ δύο καταστάσεων, επομένως παίρνουν τις λογικές τιμές “True” ή “False”.

Συνάρτηση LambaCom:



Εικόνα 18: Εισαγωγή κώδικα συνάρτησης LambdaCommunication

Ουσιαστικά η συνάρτηση αυτή είναι υπεύθυνη να συγκεντρώνει όλες τις αλλαγές που πραγματοποιήθηκαν, να τις συνθέτει μεταξύ τους, και να τις αποστέλλει στην διεύθυνση του μικροελεκτή. Για παράδειγμα αν “B1”=“2”, “B2”=“False”, “B3”=“True”, τότε στον μικροελεκτή θα σταλεί το σήμα “B12B2LB3H”.

Επιπλέον με τη χρήση αυτής της συνάρτησης μειώνεται η πιθανότητα μετάδοσης εσφαλμένης πληροφορίας, καθώς ακόμη και σε περίπτωση μη μετάδοσης δεδομένων, όπως για παράδειγμα κακής ποιότητας σύνδεση, ο συντονισμός του προγράμματος θα επιτευχθεί στην αμέσως επόμενη επικοινωνία με το μικροελεκτή.

Η σύνθεση του τελικού πακέτου προς αποστολή φαίνεται βήμα-βήμα στην Εικόνα 18, ενώ όπως παρατηρείται, κάθε φορά που συνθέτετε ένα πακέτο και αποστέλλεται με

επιτυχία, η μεταβλητή “LOut” θα πρέπει να αρχικοποιηθεί στην IP διεύθυνση του μικροελεγκτή για να είναι εφικτή η σύνθεση νέου πακέτου.

4.2.3 Συμπεράσματα από την εφαρμογή

Όπως έχει προαναφερθεί σκοπός της εφαρμογής είναι πέρα από την αποστολή εντολών προς το μικροελεγκτή για τον ορθό χειρισμό της λειτουργίας της λάμπας, να είναι εύχρηστη και πλήρως κατανοητή από το χρήστη, ανεξαρτήτως από το μορφωτικό του επίπεδο ή την ηλικία του. Κατά τη χρήση της εφαρμογής, ο χρήστης είναι σε θέση να κατανοήσει πλήρως την κατάσταση λειτουργίας της λάμπας, καθώς έχουν χρησιμοποιηθεί διαφορετικά χρώματα, για διαφορετικές καταστάσεις των διάφορων Button. Τα χρώματα που χρησιμοποιήθηκαν φαίνονται στον Πίνακας 4

	“On”	“Off”	“Sensors”
GenButton	Πράσινο	Κόκκινο	Μωβ
MoveButton	Κίτρινο	Καφέ	-
LightButton	Κίτρινο	Καφέ	-

Πίνακας 4: Χρωματικός κώδικας Button εφαρμογής

Συνοπτικά ο χρήστης έχει τη δυνατότητα τριών κυρίως επιλογών, οι οποίες είναι:

1. Η λάμπα να φωτοβολεί μόνιμα. Σε αυτή την κατάσταση το κουμπί είναι πράσινο, ενώ η κατάσταση των MoveButton και LightButton δεν επηρεάζει τη λειτουργία της λάμπας. Αντίστοιχα, οπτικά η φωτογραφία της λάμπας από την Εικόνα 16 είναι η LightOn. Η κατάσταση αυτή φαίνεται στην Εικόνα 19.
2. Η λάμπα να είναι μη φωτοβολεί. Σε αυτή την κατάσταση το κουμπί λειτουργίας (GenButton) είναι κόκκινο, η κατάσταση των αισθητήρων δεν επηρεάζουν τη λειτουργία της λάμπας, ενώ η φωτογραφία της λάμπας (από Εικόνα 16) είναι η LightOff. Η κατάσταση αυτή φαίνεται στην Εικόνα 20.
3. Η λάμπα να λειτουργεί με βάση τα σήματα από τους αισθητήρες. Σε αυτή την περίπτωση, το κουμπί λειτουργίας είναι χρώματος μωβ, ενώ η φωτογραφία της λάμπας είναι η “LightOnOff” (από Εικόνα 16) για να υποδηλώσει ότι η λάμπα αναβοσβήνει ανάλογα με την κατάσταση των αισθητήρων. Αντίστοιχα, στην κατάσταση αυτή, ο χρήστης μπορεί να επιλέξει να λειτουργεί μόνο ο αισθητήρας κίνησης (να ανάβει όταν

ο αισθητήρας αντιλαμβάνεται κάποια κίνηση), μόνο ο αισθητήρας φωτός (να φωτοβολεί όταν υπάρχει σκοτάδι) ή και οι δύο.

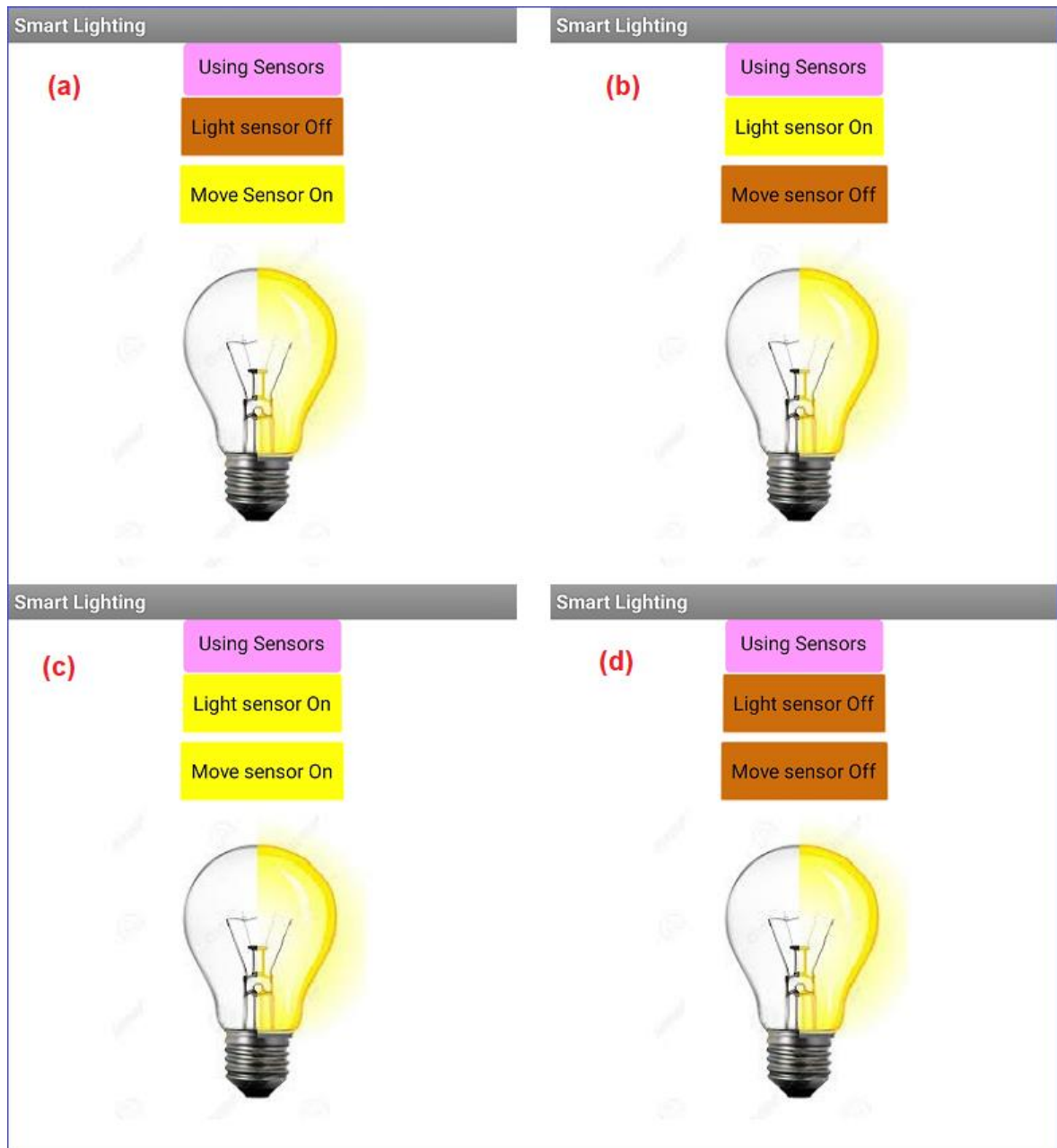
Διάφορα παραδείγματα λειτουργίας της εφαρμογής φαίνονται στις παρακάτω εικόνες. Στην Εικόνα 19 η λάμπα είναι μόνιμα αναμμένη. Αντίθετα στην Εικόνα 20, ο χρήστης έχει επιλέξει η λάμπα να παραμείνει μόνιμα σβηστή. Στην Εικόνα 21 (a), η λάμπα θα λειτουργεί μόνο όταν λαμβάνει εντολή από τον αισθητήρα κίνησης, ενώ στην Εικόνα 21 (b), μόνο όταν λαμβάνει εντολή από τον αισθητήρα φωτός, αγνοώντας τον αισθητήρα κίνησης. Στην Εικόνα 21 (c) η λάμπα θα φωτοβολεί όταν έχει χαμηλό φωτισμό ή όταν ανιληφθεί κίνηση ο αισθητήρας κίνησης, ενώ στο Εικόνα 21 (d), θα αγνοηθούν και οι δύο αισθητήρες, οπότε η λάμπα θα παραμείνει σβηστή.



Εικόνα 19: Παράδειγμα από App



Εικόνα 20: Παράδειγμα από App



Εικόνα 21: Παραδείγματα από App

4.3 Προγραμματιστικό περιβάλλον ArduinoIDE

Το προγραμματιστικό περιβάλλον Arduino IDE, είναι το επίσημο προγραμματιστικό περιβάλλον για την εγγραφή κώδικα και μεταγλώττισής του και για τον προγραμματισμό όλων των πλακετών Arduino. Έχει πολλές ομοιότητες με την γλώσσα προγραμματισμού C, ενώ υπάρχουν πολλές διαθέσιμες βιβλιοθήκες, είτε από την ίδια την εταιρεία, είτε από άλλους χρήστες καθώς είναι ανοικτού κώδικα.

Ο κώδικας που γράφεται στο περιβάλλον αυτό, η εταιρία Arduino τον αποκαλεί sketch, αποτελείται από δύο κυρίως συναρτήσεις. Τη συνάρτηση `setup()`, η οποία εκτελείται μόνο μία φορά στην εκκίνηση εκτέλεσης του προγράμματος και στην οποία ο χρήστης δηλώνει τις θύρες εισόδου/εξόδου της πλακέτας που θα χρησιμοποιήσει καθώς και τμήματα κώδικα που θέλει να εκτελεστούν μόνο μία φορά. Η δεύτερη συνάρτηση είναι η συνάρτηση `loop()`. Η συνάρτηση αυτή είναι επαναλαμβανόμενη και εκτελείται έως τη βίαιη διακοπή της από το χρήστη, όπως αποσύνδεση της τροφοδοσίας της πλακέτας. Πέρα από αυτά τα δύο κύρια μέρη, ο προγραμματιστής μπορεί να προσθέσει πλήθος συναρτήσεων, ενώ υπάρχει δυνατότητα αλληλεπίδρασης με το εκτελέσιμο πρόγραμμα μέσω του serial editor, όπως για να μάθει την τιμή μιας αναλογικής εισόδου από έναν αισθητήρα.

Ο κώδικας που γράφτηκε για τον έλεγχο του “Arduino Uno R2” παρατίθεται στην Παραπομπή, ενώ τα κυριότερα σημεία του αναλύονται παρακάτω: [31]

1. Δήλωση γενικών μεταβλητών (global variables):

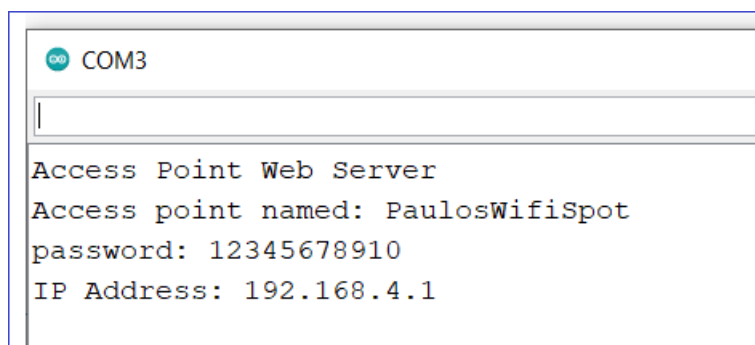
Οι γενικές ή Global μεταβλητές είναι οι μεταβλητές που θέλουμε να διατηρούν την τιμή τους ή να τις επικαλεστούμε, ανεξάρτητα από το αν το πρόγραμμα εκτελεί μια συνάρτηση ή βρίσκεται στο επαναλαμβανόμενο τμήμα του περιβάλλοντος IDE. Οι κυριότερες γενικές μεταβλητές που θα χρησιμοποιηθούν είναι:

`ApGenButton`, `ApLightButton` και `ApMoveButton`, οι οποίες αποθηκεύουν τις τιμές των αντίστοιχων μεταβλητών της εφαρμογής στο smartphone, και η κατάστασή τους αλλάζει μόνο κατά την αλληλεπίδραση του Arduino με την εφαρμογή.

`MoveSensorStatus`, `LightSensorStatus`, οι οποίες ουσιαστικά δέχονται και αποθηκεύουν την κατάσταση των αισθητήρων. Επιπρόσθετα η μεταβλητή `LightSensorSens` εισάγεται από το χρήστη, και ρυθμίζει τη στάθμη της φωτεινότητας, δηλαδή την ευαισθησία, στην οποία η εφαρμογή θα θεωρεί ότι υπάρχει φως ή σκοτάδι.

`MoveSensorPin`, `MoveLightPin`, `OutCommandPin`, οι οποίες καταχωρούν τις θύρες του Arduino στις οποίες θα συνδεθούν οι αισθητήρες ή οι εντολές προς τον ρελέ που ελέγχει τη λάμπα.

ssid, Pass, Sport, στις οποίες ο χρήστης καταχωρεί τα χαρακτηριστικά όπως το όνομα δικτύου, τον κωδικό πρόσβασης και τη θύρα. Στην Εικόνα 22 φαίνονται τα χαρακτηριστικά του τοπικού δικτύου και της πλακέτας, όπως έχουν αποτυπωθεί στο Serial Monitor.



Εικόνα 22: Εμφάνιση χαρακτηριστικών τοπικού δικτύου

2. Δήλωση τοπικών μεταβλητών.

Οι μεταβλητές αυτές δηλώνονται και έχουν λόγο ύπαρξης μόνο κατά την εκτέλεση μιας συγκεκριμένης συνάρτησης. Οι κυριότερες από τις οποίες είναι:

CharReceive, η οποία υποδηλώνει το πλήθος των χαρακτήρων του πακέτου που αναμένεται από την εφαρμογή. Στη συγκεκριμένη περίπτωση ισούται με 9, πχ. "B12B2LB3H".

ComReceived, currentLine, είναι μεταβλητές που αποθηκεύουν προσωρινά τμήματα από την εισερχόμενη επικοινωνία μεταξύ της εφαρμογής και του Arduino. Αυτές χρησιμοποιούνται για να ανιχνεύσουν την ύπαρξη του επιθυμητού πακέτου πληροφορίας από την εφαρμογή.

3. Διαχωρισμός πακέτου πληροφορίας από το App.

Κατά τη διάρκεια της επικοινωνίας της εφαρμογής με τον Arduino, ο Arduino δέχεται πακέτα πληροφοριών από την εφαρμογή και στη συνέχεια πρέπει να διαχωρίσει την εντολή του χρήστη μέσω της εφαρμογής. Αυτό επιτυγχάνεται με τα εξής βήματα:

- λήψη των εισερχόμενων πακέτων χαρακτήρα-χαρακτήρα έως τη συμπλήρωση γραμμής.
- Έλεγχος αν η συμπληρωμένη γραμμή έχει περισσότερους χαρακτήρες από τους αναμενόμενους, στην περίπτωσή μας εννέα.

- Εάν έχει περισσότερους από εννέα χαρακτήρες, αποθηκεύουμε και κρατάμε τους τελευταίους εννέα, δηλαδή το υποψήφιο τμήμα του πακέτου που περιμένουμε.
- Στη συνέχεια συγκρίνουμε τους χαρακτήρες αυτούς με την αναμενόμενη μορφή η οποία είναι η “B1_B2_B3_”
- Αν οι ληφθέντες χαρακτήρες συμφωνούν με την παραπάνω σειρά, τότε αποθηκεύονται οι αντίστοιχες τιμές κατάστασης των B1, B2 και B3 μεταβλητών του App.

Στην Εικόνα 23 παρατίθεται το τμήμα του κώδικα όπου φαίνεται η διαδικασία αυτή.

```
// Αποκωδικοποίηση του κειμένου που λήφθηκε:
if (currentLine.length() > 9) { // αν είναι περισσότεροι από 9 χαρακτήρες
    ComReceived = "";
    ComLenght = currentLine.length(); // πάρε τους τελευταίους 9
    for (i = 0; i < 9; i++) {
        ComReceived += currentLine[ComLenght - 9 + i];
    }

    // και σύγκρινέ τους με την αναμενόμενη ακολουθία
    // Αν η ακολουθία είναι σωστή, τότε ενημέρωσε τις αλλαγές στις αντίστοιχες μεταβλητές.
    if (ComReceived[0] == 'B' && ComReceived[1] == '1' && ComReceived[3] == 'B' &&
        ComReceived[4] == '2' && ComReceived[6] == 'B' && ComReceived[7] == '3') { //B1XB2XB3X
        if (ComReceived[2] == '1') {
            APGenButton = 1;
        } else if (ComReceived[2] == '2') {
            APGenButton = 2;
        } else {
            APGenButton = 3;
        }
        if (ComReceived[5] == 'H') {
            APLightSensor = HIGH;
        } else {
            APLightSensor = LOW;
        }
        if (ComReceived[8] == 'H') {
            APMoveSensor = HIGH;
        } else {
            APMoveSensor = LOW;
        }
    }
}
```

Εικόνα 23: Κώδικας για τη λήψη της πληροφορίας από το App

4. Έλεγχος λάμπας. Ο έλεγχος της λάμπας πραγματοποιείται από μια συνάρτηση (LightOperation) η οποία ελέγχει τον τρόπο λειτουργίας που θέλει ο χρήστης (μέσω της μεταβλητής ApGenButton). Στη συνέχεια αν ο χρήστης έχει επιλέξει τη λειτουργία μέσω αισθητήρων, η συνάρτηση συγκρίνει τα δεδομένα των αισθητήρων με τις επιλογές του χρήστη για τη χρήση των αισθητήρων και θέτει αναλόγως τη μεταβλητή LedStatus στις καταστάσεις “0” ή “1” ανάλογα αν η λάμπα πρέπει να φωτοβολεί ή όχι. Ο κώδικας της συγκεκριμένης συνάρτησης φαίνεται στην Εικόνα 24.

```

void LightOperation() {

  if (APGenButton == 1) { //anamenō apo app
    digitalWrite(led, LOW); // anapoda. Ενδικοτικό led στην πλακέτα
    LedStatus = HIGH;
  }
  else if (APGenButton == 3) { //svismenō apo app
    digitalWrite(led, HIGH);
    LedStatus = LOW;
  }
  else {
    if (((MoveSensorStatus == HIGH) && (APMoveSensor == HIGH)) ||
        ((LightSensorStatus < LightSensorSens) && (APLightSensor == HIGH))) { //Έλεγχος με αισθητήρες.
      digitalWrite(led, LOW);
      LedStatus = HIGH;
    }
    else {
      digitalWrite(led, HIGH);
      LedStatus = LOW;
    }
  }
}
}

```

Εικόνα 24: Κώδικας συνάρτησης LightOperation

4.4 Κατασκευαστικό τμήμα

Για την ολοκλήρωση του project, είναι απαραίτητο πέρα από το προγραμματιστικό κομμάτι, να υλοποιηθεί και το κατασκευαστικό κομμάτι, δηλαδή η κατάλληλη επιλογή υλικών, η παραμετροποίησή τους και τέλος η συνδεσμολογία τους σε ένα ενιαίο κύκλωμα.

Το κατασκευαστικό κομμάτι μπορεί να χωριστεί σε δύο διακριτά τμήματα:

1. Τμήμα ελέγχου (αναφέρεται και ως ασθενή ρεύματα). Αυτό το τμήμα περιλαμβάνει όλους τους αισθητήρες, την καλωδίωσή τους, καθώς και όλα τα στοιχεία που είναι υποβοηθητικά στο κύκλωμα όπως για παράδειγμα οι ενδεικτικές λυχνίες led. Συνοπτικά είναι το τμήμα της κατασκευής που λειτουργεί με τάση 5V και σκοπός του είναι να ελέγχει το κύκλωμα των 220 V.

2. Τμήμα λειτουργίας (ή ισχυρά ρεύματα). Πρόκειται για το λειτουργικό τμήμα της κατασκευής που λειτουργεί στα 220V. Λόγω της υψηλής τάσης και της επικινδυνότητάς της πρέπει οι εργασίες να πραγματοποιούνται με πολύ μεγάλη προσοχή.

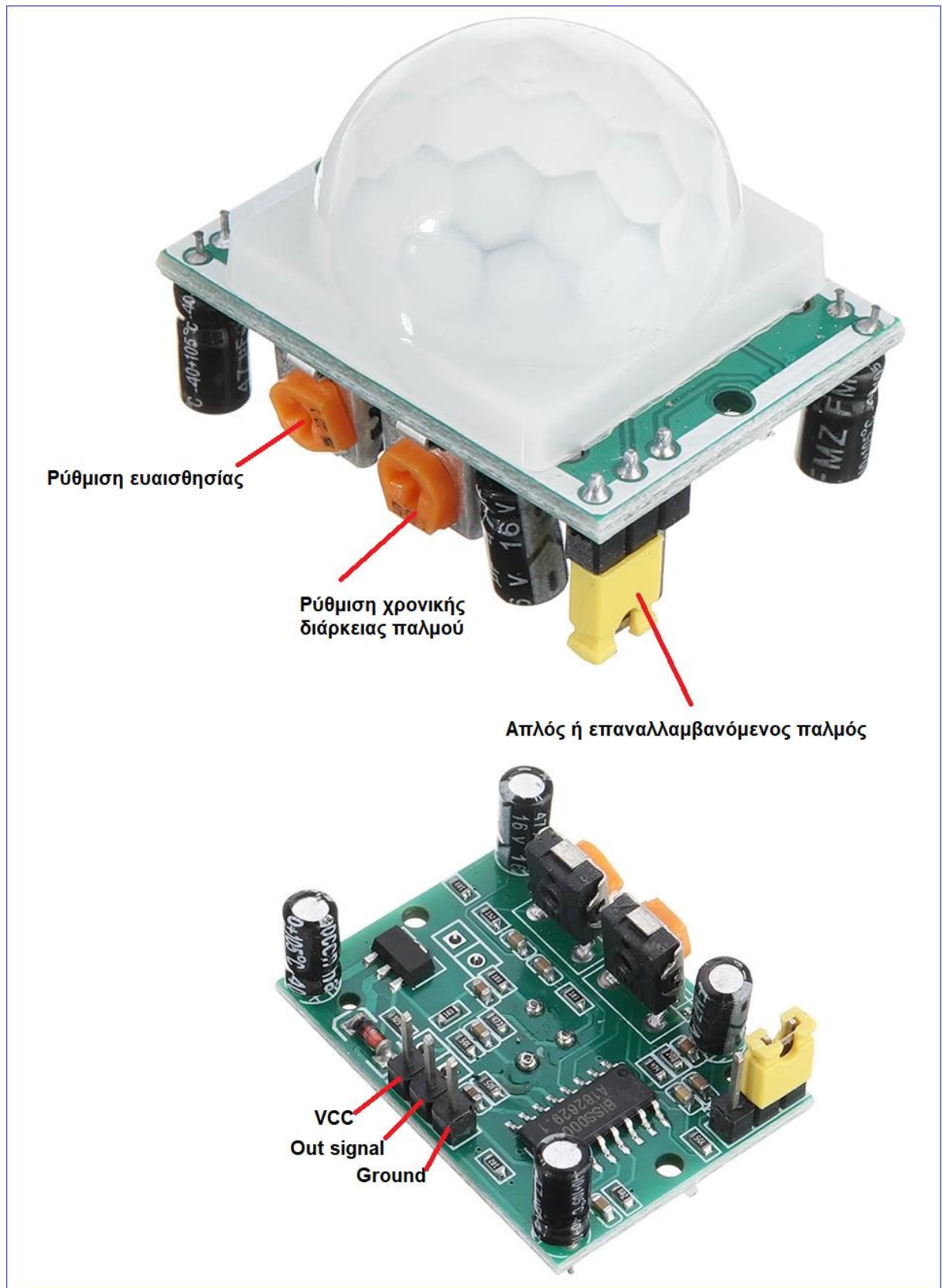
4.4.1 Κύκλωμα ελέγχου

Το κύκλωμα ελέγχου περιλαμβάνει όλους τους αισθητήρες και τα εξαρτήματα με τάση λειτουργίας τα 5V. Πιο συγκεκριμένα χρησιμοποιήθηκαν:

- 1 πλακέτα breadboard. Σε αυτή συνδέονται όλα τα εξαρτήματα, με ασφάλεια, χωρίς να χρειάζονται κολλήσεις.
- 1 αισθητήρας κίνησης.
- 1 αισθητήρας φωτός (φωτοαντίσταση).
- 1 Relay.
- Καλώδια
- Αντιστάσεις διαφόρων τιμών.
- Ενδεικτικές λυχνίες Led

Αισθητήρας κίνησης: Στην παρακάτω Εικόνα 25 παρουσιάζεται ο αισθητήρας που χρησιμοποιήθηκε. Έχει διευρυμένες δυνατότητες και ο χρήστης μπορεί να ρυθμίσει την ευαισθησία του χρησιμοποιώντας το δεξί ροοστάτη (μεταβλητή αντίσταση) ενώ με τον αριστερό μπορεί να προσαρμόσει την χρονική διάρκεια για την οποία η έξοδος του θα παραμείνει υψηλή μετά από αντίκρουση κάποιας κίνησης. Τέλος παρέχει τη δυνατότητα για πολλαπλή λειτουργία ανά περιόδους (ο παλμός εξόδου να επαναλαμβάνεται μετά από εντοπισμό κίνησης).

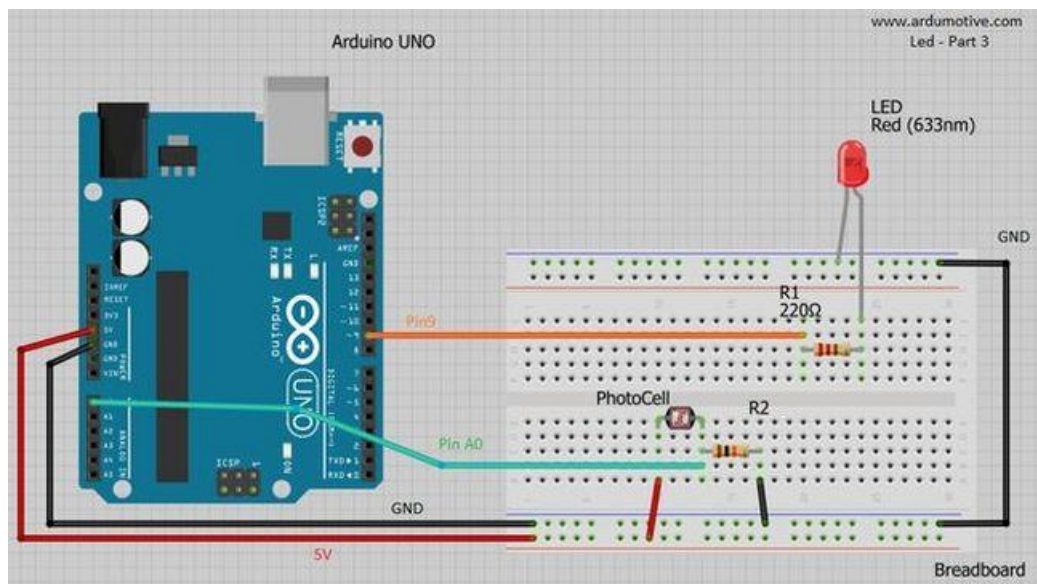
Στα πλαίσια της συγκεκριμένης εργασίας η ευαισθησία του ορίστηκε στο ελάχιστο, ενώ ο χρόνος λειτουργίας στα ~12 sec. Το σήμα εξόδου του αισθητήρα είναι ψηφιακό και συνδέθηκε στο Pin No. 2 του Arduino.



Εικόνα 25: Αισθητήρας κίνησης – Shield [xiii]

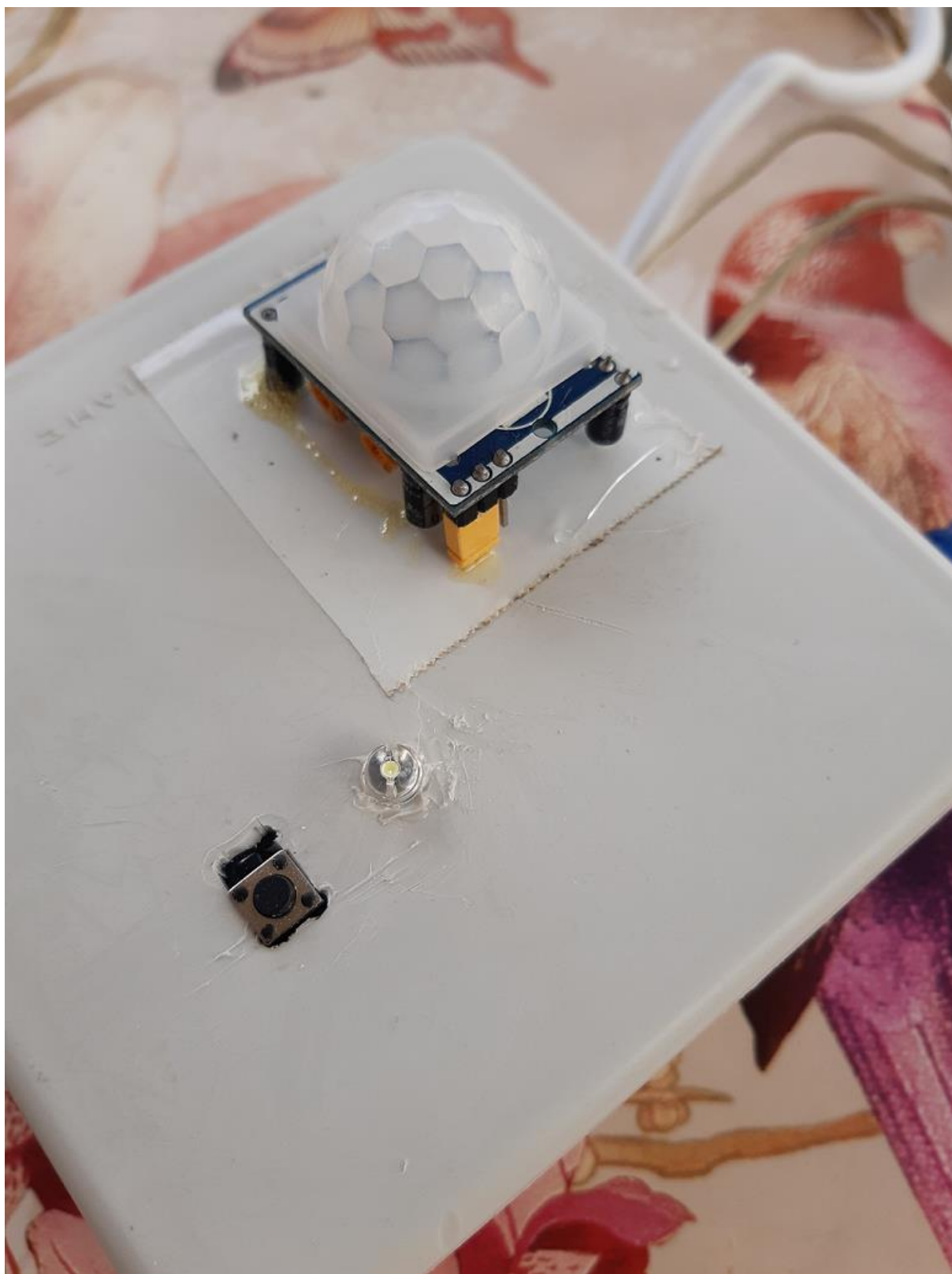
Αισθητήρας Φωτός: Στην παρακάτω Εικόνα 26 παρουσιάζεται ο αισθητήρας που χρησιμοποιήθηκε. Πρόκειται στην ουσία για μία φωτοαντίσταση, δηλαδή ένα παθητικό στοιχείο το οποίο έχει την ικανότητα να αλλάζει την τιμή της αντίστασής του ανάλογα με την ένταση της ακτινοβολίας που δέχεται. Δυστυχώς η τιμή της φωτοαντίστασης δεν μεταβάλλεται γραμμικά σύμφωνα με τη μεταβολή της φωτεινότητας ενώ το εύρος τιμών της είναι από 1 KΩ έως 1GΩ. Για τη χρήση της στο project χρησιμοποιήθηκε ως ένας διαιρέτης τάσης με μία αντίσταση τιμής 2,2 KΩ.

Μια τυπική συνδεσμολογία φαίνεται στο ακόλουθο σχήμα της Εικόνα 26, ενώ στην παρούσα εργασία συνδέθηκε με τον μικροελεγκτή στο Pin No. A1 (Αναλογική τιμή εισόδου).



Εικόνα 26: Αισθητήρα φωτός – Φωτοαντίσταση και συνδεσμολογία. [xii]

Ενδεικτική λυχνία Led λειτουργίας: Είναι ένα Led το οποίο δέχεται απευθείας τροφοδοσία (μέσω μιας αντίστασης για να μην καεί) και ελέγχεται από ένα κουμπί. Προστέθηκε για να μπορεί ο χρήστης να ελέγχει αν ο μικροελεγκτής έχει παροχή τάσης, καθώς τα ενσωματωμένα ενδεικτικά led της πλακέτας δεν είναι πλέον εμφανή. Η ενδεικτική λυχνία και το κουμπί φαίνεται στην παρακάτω Εικόνα 27.



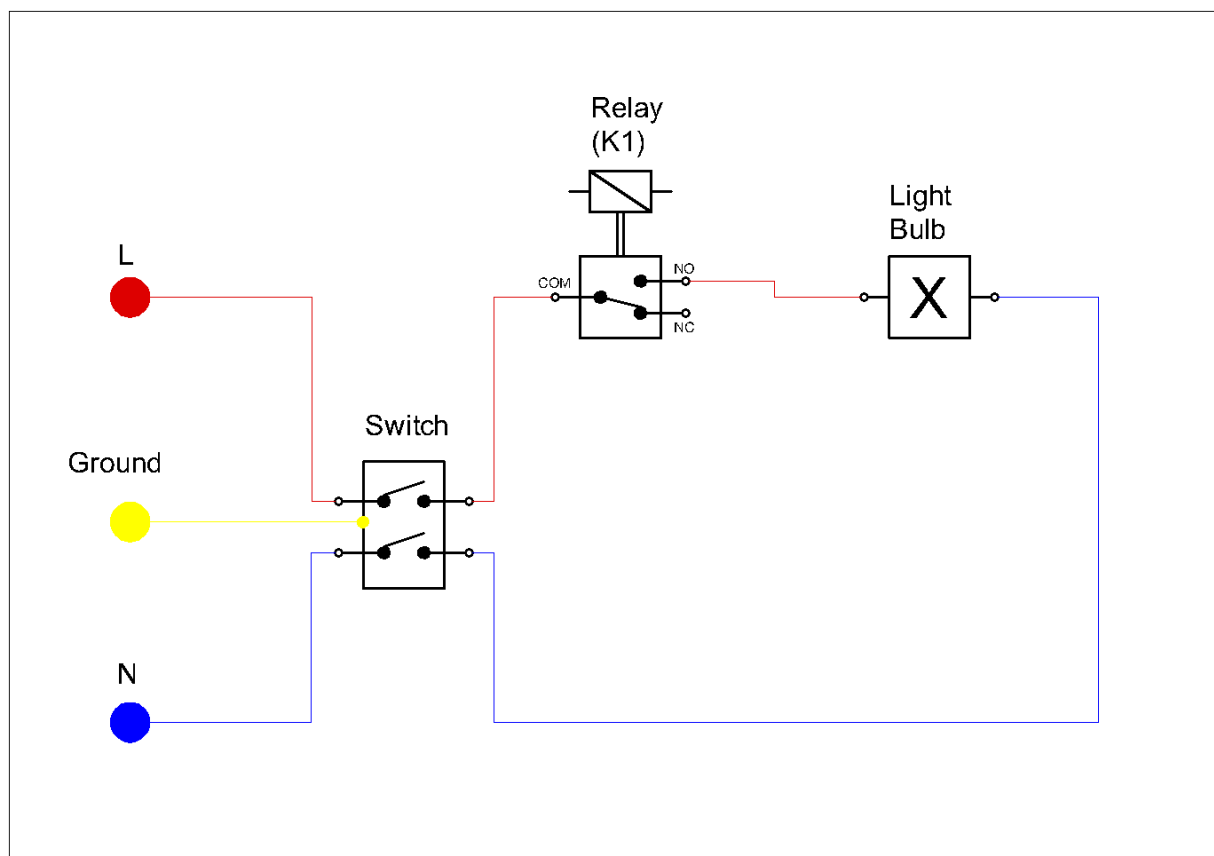
Εικόνα 27: Κουμπί ελέγχου τροφοδοσίας και led ελέγχου τροφοδοσίας

4.4.2 Κύκλωμα λειτουργίας

Η κυκλωματική κατασκευή (διάγραμμα λειτουργίας) που υλοποιήθηκε φαίνεται στο σχήμα της Εικόνα 28. Όπως μπορούμε να διακρίνουμε το κύκλωμα φωτισμού μπορεί να τροφοδοτηθεί είτε από κάποια σταθερή παροχή – απευθείας από γραμμή πίνακα – είτε, όπως και στην περίπτωση μας, από έναν ρευματοδότη (πολύμπριζο).

Το κύκλωμα φωτισμού ελέγχεται απομακρυσμένα μέσω του Η/Ν (relay) K1. Στην περίπτωση που μελετάμε, ο Η/Ν ελέγχεται μέσω εντολής από τον Arduino uno. Στην κατάσταση Normal Close, ο ΗΝ δεν διεγείρεται (έξοδος από “Arduino” = “LOW” – 0 V) με συνέπεια να μη διέρχεται ρεύμα από το κύκλωμα και η λάμπα να μη φωτοβολεί. Αντίθετα, αν υπάρχει διέγερση του Η/Ν (έξοδος/εντολή από arduino = “HIGH” – 5 V) αλλάζει θέση η επαφή του ΗΝ με αποτέλεσμα η λάμπα να φωτοβολεί.

Τέλος ο διακόπτης μπορεί να διακόψει τοπικά την τροφοδοσία στο κύκλωμα όταν το επιθυμεί ο ένοικος (όπως για αλλαγή λάμπας στο φωτιστικό σημείο ή απομόνωση του φωτιστικού σημείου σε περίπτωση βραχυκυκλώματος)

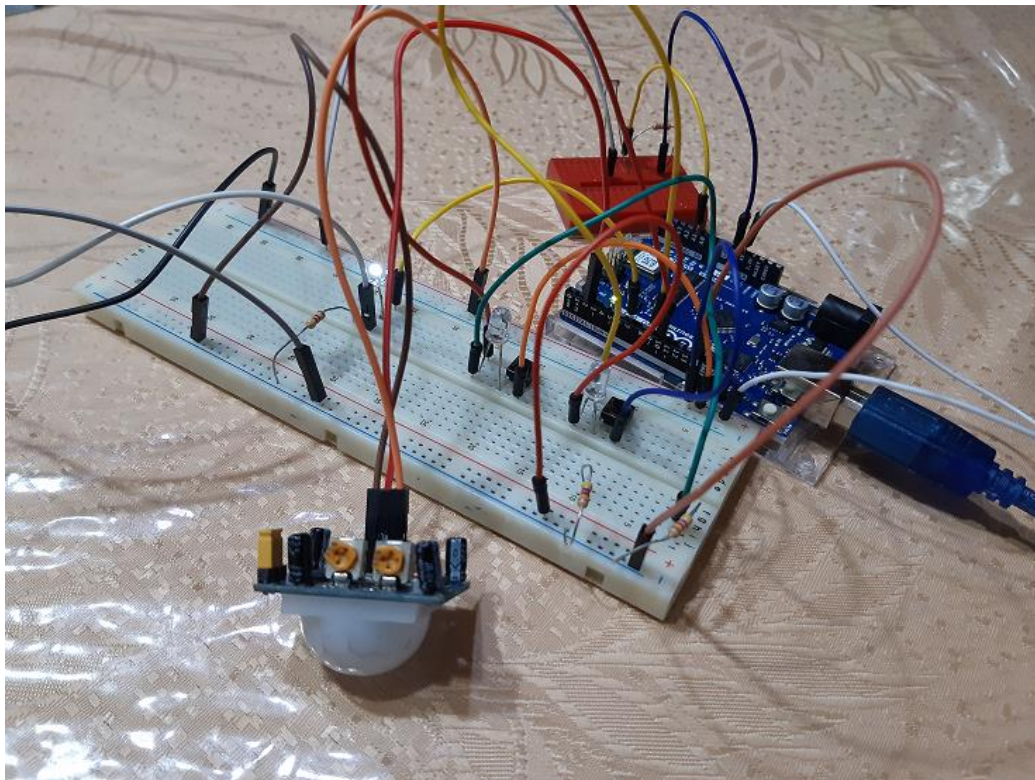


Εικόνα 28: Κυκλωματικό διάγραμμα ισχυρών ρευμάτων

4.4.3 Φωτογραφικό υλικό κατά τη φάση υλοποίησης της εργασίας

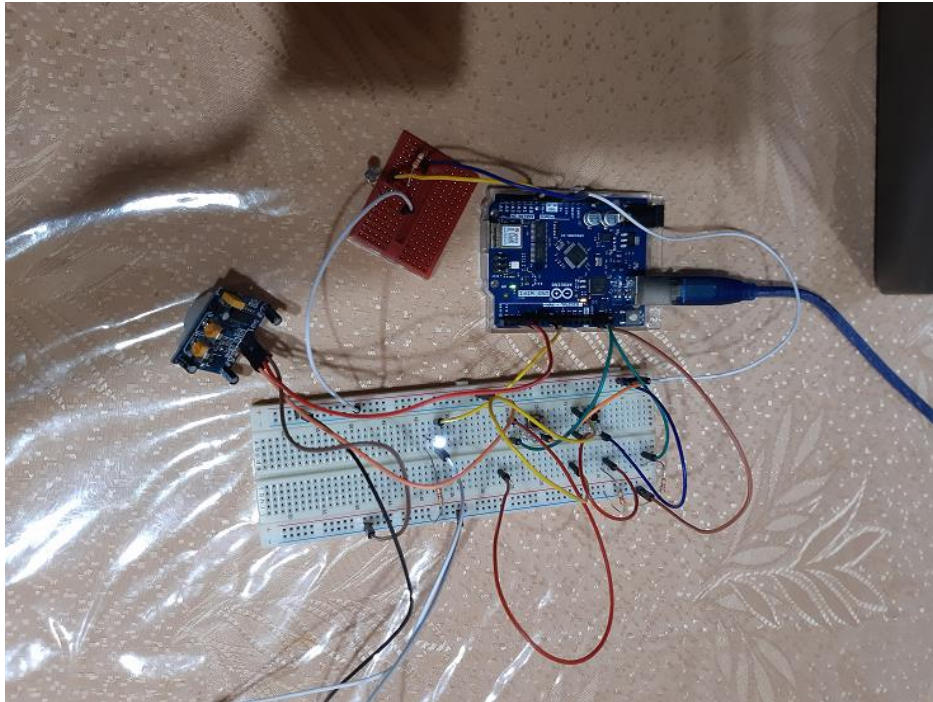
Ακολουθούν εικόνες σε διάφορα στάδια κατασκευής, μέχρι την τελική υλοποίηση του Project.

Εικόνα 29: Στα αρχικά στάδια κατασκευής του project, λόγω δυσκολιών στην εύρεση των κατάλληλων αισθητήρων, οι αισθητήρες είχαν προσομοιωθεί με τα εικονιζόμενα δύο button. Σε σειρά με τους «αισθητήρες» είχαν τοποθετηθεί ενδεικτικές λυχνίες led για να υπάρχει και οπτική επαφή λειτουργίας του αισθητήρα ή μη. Τέλος η εντολή προς τον Ρελέ είχε αντικατασταθεί από το λαμπάκι που φαίνεται στην εικόνα να φωτοβολεί. Η εικόνα είναι τραβηγμένη κατά την προσπάθεια μετάβασης και της αντικατάστασης των Buttons με τους πραγματικούς αισθητήρες, και έλεγχος καλής λειτουργίας τους.



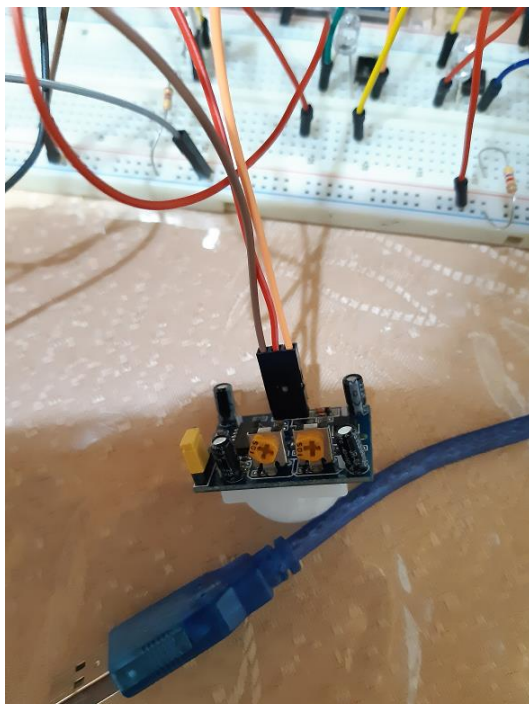
Εικόνα 29: Μετάβαση από Buttons σε αισθητήρες

Εικόνα 30: Η φωτογραφία τραβήχτηκε στην ίδια χρονική περίοδο με αυτήν της Εικόνας 29, όμως επικεντρώνεται στην υλοποίηση και συνδεσμολογία του αισθητήρα φωτός ο οποίος βρίσκεται στο κόκκινο breadboard.



Εικόνα 30: Συνδεσμολογία Αισθητήρων

Εικόνα 31: Ρύθμιση της ευαισθησίας και του χρόνου λειτουργίας του αισθητήρα κίνησης, μέσω των δύο ροοστατών που φαίνονται. Η διαδικασία ρύθμισης έχει αναλυθεί προηγουμένως στην Παράγραφο 4.4.1.



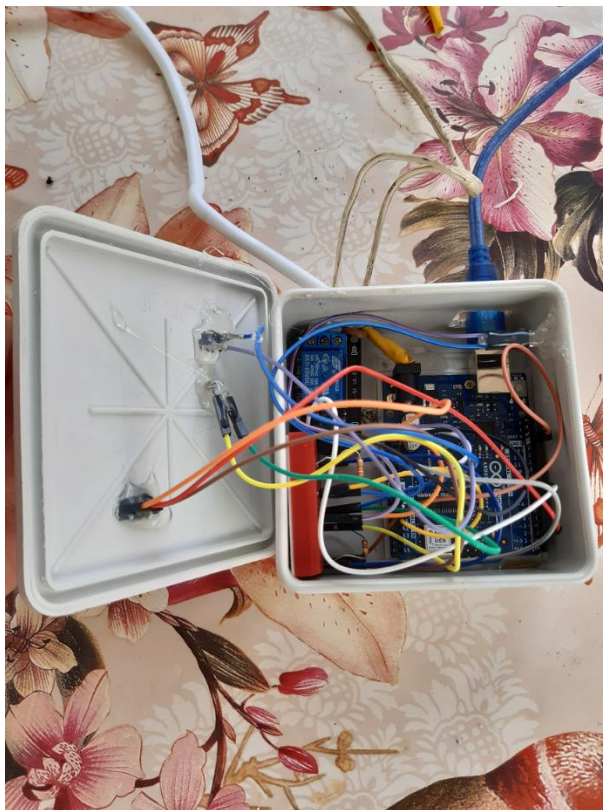
Εικόνα 31: Ρύθμιση αισθητήρα κίνησης

Εικόνα 32: Φαίνεται η τελική καλωδίωσή του project πριν την τοποθέτησή του σε κατάλληλο κουτί.



Εικόνα 32: Τελική συνδεσμολογία του σχεδίου

Εικόνα 33, Εικόνα 34: Εικονίζεται η τελική εμφάνιση του σχεδίου, όπου όλα τα υλικά έχουν τοποθετηθεί σε ένα προστατευτικό πλαστικό κουτί.



Εικόνα 33: Τελική παρουσίαση σχεδίου



Εικόνα 34: Τελική εμφάνιση σχεδίου

Εικόνα 35: Στην εικόνα αυτή φαίνεται το ολοκληρωμένο project σε λειτουργία, όπου η λάμπα φωτοβολεί σύμφωνα με τις εντολές που δέχεται από το App και τους αισθητήρες.



Εικόνα 35: Λειτουργία του project

5 ΣΥΜΠΕΡΑΣΜΑΤΑ

5.1 Συμπεράσματα

Φτάνοντας στα συμπεράσματα, θα ήθελα να πω ότι μέσα από αυτή την εργασία έμαθα πολλά πράγματα, που μόνο καλό έχουν να μου προσφέρουν. Έμαθα να χειρίζομαι το περιβάλλον του Arduino και να τον προγραμματίζω προσθέτοντας επιπλέον εξαρτήματα, με σκοπό να πραγματοποιήσω την αρχική μου σκέψη. Έπειτα για να καταφέρω να δημιουργήσω την εφαρμογή που θα αλληλοεπιδρά με τον Arduino, χρησιμοποίησα το περιβάλλον ανάπτυξης εφαρμογών MIT App Inventor. Εκεί έχτισα την εφαρμογή που είναι συμβατή με Android συσκευές, εύκολα και απλά.

Αρχικά, η ιδέα μου ήρθε διαβάζοντας για την παραγωγή αυγών μίας κότας. Όπως ξέρουμε μια κότα γεννά περίπου 250 αυγά το χρόνο. Στις περιόδους αυγοπαραγωγής μια κότα συνήθως γεννάει κάθε 24 ώρες, και μάλιστα για 1-2 μήνες συνεχόμενα. Αν μια ράτσα γεννάει, γεννάει αυγά κάθε 2 η και 3 μέρες, και η διάρκεια δεν φτάνει τον ένα μήνα. Όσο πιο γερασμένη είναι τόσο μικρότερη η παραγωγή. Οπότε σκέφτηκα ότι με κάποιο τρόπο θα μπορούσε να αυξηθεί η παραγωγή αυγών, χωρίς όμως να χρειαστεί να αγοράσει κάποιος περισσότερες κότες. Αν τις έβαζε σε ένα μέρος και δημιουργούσε με τεχνητό φως τη ψευδαίσθηση ότι είχε ξημερώσει, κάνοντας αυτή τη διαδικασία τουλάχιστον δύο φορές τη μέρα, η παραγωγή θα αυξανόταν.

Εδώ έρχεται η δική μου ιδέα, φτιάχνοντας ένα σύστημα, με χαμηλό κόστος και μπορεί να χειριστεί το φως, θα έκανε τη δουλειά πιο εύκολη. Έτσι, λοιπόν, ο κάθε παραγωγός θα μπορούσε να συνδέσει αυτό που δημιούργησα και μέσα από την εφαρμογή θα μπορεί να χειρίζεται το φως, χωρίς να χρειάζεται να βάλει διακόπτες και άλλα πράγματα που θα έπρεπε να τα χρησιμοποιεί δια χειρός. Αυτό δε σημαίνει όμως ότι θα μπορούσαν μόνο επαγγελματίες να μπορούν να χρησιμοποιούν τη κατασκευή αυτή. Ο κάθε άνθρωπος θα μπορούσε να τη χρησιμοποιήσει ακόμα και στο σπίτι του, για παράδειγμα σα λάμπα το δωμάτιο του και όταν ξαπλώνει με ένα πάτημα από το κινητό του να κλείνει το φως, χωρίς να χρειαστεί να ξανασηκωθεί.

5.2 Ιδέες για μελλοντική επέκταση της εφαρμογής

Σίγουρα, όμως, δε σταματάει εδώ η ιδέα. Θα μπορούσε να ενισχυθεί και να εξελιχθεί με πολλά πράγματα. Αυτή τη στιγμή η εφαρμογή λειτουργεί μόνο σε τοπικό δίκτυο, επομένως μελλοντικά θα μπορούσε να φτιαχτεί έτσι ώστε να μπορείς να ελέγξεις τη λάμπα από οποιοδήποτε δίκτυο είσαι συνδεδεμένος. Ένα ακόμα είναι να μπορεί να ελέγχει πάνω από μία λάμπες ταυτόχρονα. Όπως ανέφερα πριν, για κάποιον επαγγελματία (και μη) θα ήταν χρησιμότερο να μπορεί να συνδέει όσες λάμπες εκείνος επιθυμεί, προκειμένου να κάνει τη δουλειά του.

Επιπλέον θα μπορούσε να προστεθεί κάτι σαν αφύπνιση, δηλαδή μέσω της εφαρμογής να δίνεται η δυνατότητα στο χρήστη να επιλέγει ένα εύρος ωρών που θα ήθελε να παραμένουν ανοιχτά τα φώτα και το αντίθετο. Για παράδειγμα σε μία εταιρία, με γραφεία και πολλούς ορόφους, όταν κλείνει μετά από κάποιο χρονικό διάστημα να κλείνουν τα φώτα μέσω της αφύπνισης, που προανέφερα. Αυτό θα βοηθάει και τον επιστάτη ή το σεκιούριτι χωρίς κόπο να ελέγχει τα φώτα.

Τέλος θα μπορούσε να υπάρχει μεγαλύτερη ευχέρεια στο χρήστη για τη ρύθμιση των αισθητήρων, όπως την ευαισθησία του αισθητήρα φωτός, όπως και την πρόσθεση άλλων δυνατοτήτων όπως για παράδειγμα ένα “reset button”, με το οποίο ο χρήστης να μπορεί να κάνει επαναφορά του μικροελεγκτή ή να λαμβάνει πίσω μία ανάδραση για την κατάσταση της λάμπας.

5.3 Επίλογος

Τέλος, όλα αυτά, μου πήραν αρκετές ώρες μελέτης και προσπάθειας για μπορέσουν να έρθουν στη μορφή που βρίσκονται τώρα. Σίγουρα θα τα έχει σκεφτεί και κάποιος άλλος πριν από μένα, αλλά ο καθένας στη πράξη τα δείχνει αλλιώς. Η ιδέα είναι σίγουρα εξελίξιμη, δίνοντας μου κίνητρο στο μέλλον αν μου παρουσιαστεί η ευκαιρία να αναπτύξω κάτι καλύτερο και σίγουρα με πολλές και πρωτότυπες λειτουργίες.

ΠΑΡΑΡΤΗΜΑ Α.

Προγραμματισμός της πλακέτας Arduino (μέσω της πλατφόρμας Arduino IDE):

```
#include<SPI.h> // εισαγωγήβιβλιοθηκών
#include <WiFiNINA.h>

char ssid[] = "PaulosWifiSpot"; // Όνομαδικτύου
char pass[] = "12345678910"; // Password

int led = LED_BUILTIN;
int status1 = WL_IDLE_STATUS;

int APGenButton = 1; //Ληφθήσατιμήγια GenButton
bool APMoveSensor = LOW; //Ληφθήσατιμήγια MoveButton
bool APLightSensor = LOW; //Ληφθήσατιμήγια LightButton
int LedStatus = LOW; //Κατάστασηλάμπας (on/off)
int MoveSensorStatus = HIGH; //τιμή από το φυσικό Αισθητήρας κίνησης
int LightSensorStatus = HIGH; //τιμή από το φυσικό Αισθητήρας Φωτός

int MoveSensorPin = 2; // θέση pin για τον αισθητήρα κίνησης
int LightSensorPin = A1;
int OutCommandPin = 7; // Pin εντολήςγιατονέλεγχοτηςλάμπας

int LightSensorSens = 500; //the sensitivity of the light sensor

int Sport = 80; //θύρα

WiFiServer server(Sport);

void setup() {

    pinMode(led, OUTPUT); // set the pin modes
    pinMode(OutCommandPin = 7, OUTPUT);
    pinMode(MoveSensorPin, INPUT);
    pinMode(LightSensorPin, INPUT);

    //Initialize serial port:
    Serial.begin(9600);

    while (!Serial) {

        ; // αναμονή να ανοίξει η σειριακή θύρα. Μόνο για H/Y και ανοικτή εφαρμογή.
```

```

}

Serial.println("Access Point Web Server");

// Create open network.
status1 = WiFi.beginAP(ssid, pass);

if (status1 != WL_AP_LISTENING) {
  Serial.println("Creating access point failed");

  // don't continue
  while (true);
}

// wait 10 seconds for connection:
delay(1000);

// start the web server on port 80
server.begin();

// Καταγραφή και εμφάνιση δεδομένων σύνδεσης στη σειριακή πόρτα.
Serial.print("Access point named: ");
Serial.println(ssid);
Serial.print("password: ");
Serial.println(pass);
IPAddress ip = WiFi.localIP();
Serial.print("IP Address: ");
Serial.println(ip);
}

void loop() {

  String ComReceived = ""; // The received string command from the app
  int ComLenght = 0;
  int i = 0;
  int CharReceive = 9; //The length of the received string from the app.

  // Ελεγχος για αλλαγή και ενημέρωση status
  if (status1 != WiFi.status()) {
    // it has changed update the variable
    status1 = WiFi.status();
  }
}

```

```

WiFiClient client = server.available(); // listen for incoming clients
if (client) {                          // if you get a client,
    String currentLine = "";           // make a String to hold incoming data from the
client
    while (client.connected()) {       // loop while the client's connected
        if (client.available()) {      // if there's bytes to read from the client,
            char c = client.read();     // read a byte, then
            if (c == '\n') {           // if the byte is a newline character
                // if the current line is blank, you got two newline characters in a row.
                // that's the end of the client HTTP request, so send a response:
                if (currentLine.length() == 0) {
                    // HTTP headers always start with a response code (e.g. HTTP/1.1 200 OK)
                    // and a content-type so the client knows what's coming, then a blank line:
                    client.println("HTTP/1.1 200 OK");
                    client.println("Content-type:text/html");
client.println();

                    break; // Αφού τελείωσε το request του πελάτη, φύγε από την επανάληψη.
                }
            }
            else { // if you got a newline, then clear currentLine:
                currentLine = "";
            }
        }
        else if (c != '\r') { // if you got anything else but a carriage return character,
            currentLine += c; // add it to the end of the currentLine
        }
        else {

            // Αποκωδικοποίηση του κειμένου που λήφθηκε:
            if (currentLine.length() > 9) { // αν είναι περισσότεροι από 9 χαρακτήρες
ComReceived = "";
                ComLenght = currentLine.length(); // πάρετουςτελευταίους 9
                for (i = 0; i < 9; i++) {
                    ComReceived += currentLine[ComLenght - 9 + i];
                }

                // και σύγκρινέ τους με την αναμενόμενη ακολουθία
                // Αν η ακολουθία είναι σωστή, τότε ενημέρωσε τις αλλαγές στις αντίστοιχες
//μεταβλητές.
                if (ComReceived[0] == 'B' && ComReceived[1] == '1' && ComReceived[3] == 'B'
&& ComReceived[4] == '2' && ComReceived[6] == 'B' && ComReceived[7] == '3')
                { //B1XB2XB3X
                    if (ComReceived[2] == '1') {
                        APGenButton = 1;

```

```

    } else if (ComReceived[2] == '2') {
        APGenButton = 2;
    } else {
        APGenButton = 3;
    }
    if (ComReceived[5] == 'H') {
        APLightSensor = HIGH;
    } else {
        APLightSensor = LOW;
    }
    if (ComReceived[8] == 'H') {
        APMoveSensor = HIGH;
    } else {
        APMoveSensor = LOW;
    }
}
}
}
}
}
client.stop();
}

//Read the sensors
MoveSensorStatus = digitalRead(MoveSensorPin);
LightSensorStatus = analogRead(LightSensorPin);
//Serial.println(LightSensorStatus);

LightOperation(); // κάλεσε τη συνάρτηση για υπολογισμό της κατάστασης της
λάμπας

if (LedStatus == HIGH) { // εναλλαγή κατάστασης (LOW-->αναμένο)
digitalWrite(OutCommandPin, LOW);
} else {
    digitalWrite(OutCommandPin, HIGH);
}
delay(100);
}

void LightOperation() {
if (APGenButton == 1) { //anamenos apo app
    digitalWrite(led, LOW); // anapoda. Ενδεικτικό led στην πλακέτα
    LedStatus = HIGH;
}
}

```

```

else if (APGenButton == 3) { //svismeno apo app
    digitalWrite(led, HIGH);
    LedStatus = LOW;
}
else {
    if (((MoveSensorStatus == HIGH) && (APMoveSensor == HIGH)) ||
        ((LightSensorStatus > LightSensorSens) && (APLightSensor == HIGH))) {
        //Ελεγχος με αισθητήρες.
        digitalWrite(led, LOW);
        LedStatus = HIGH;
    }
    else {
        digitalWrite(led, HIGH);
        LedStatus = LOW;
    }
}
}
}

```


ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] «wikipedia,» [Ηλεκτρονικό]. Available: <https://en.wikipedia.org/wiki/Microcontroller>. [Πρόσβαση 20 09 2021].
- [2] «microcontroller-versus-microprocessors,» [Ηλεκτρονικό]. Available: <https://akhiljain07.wordpress.com/microcontroller-versus-microprocessors/>. [Πρόσβαση 20 09 2021].
- [3] «microchip.com,» [Ηλεκτρονικό]. Available: <https://www.microchip.com/en-us/products/microcontrollers-and-microprocessors/8-bit-mcus/pic-mcus>. [Πρόσβαση 20 09 2021].
- [4] «electronicshub,» [Ηλεκτρονικό]. Available: <https://www.electronicshub.org/microcontrollers-basics-structure-applications/>. [Πρόσβαση 20 09 2021].
- [5] «break-ic,» [Ηλεκτρονικό]. Available: <https://www.break-ic.com/topics/break-microcontroller.asp>. [Πρόσβαση 20 09 2021].
- [6] «uaeintegrationsolutions,» [Ηλεκτρονικό]. Available: <https://uaeintegrationsolutions.com/home-automation/>. [Πρόσβαση 20 09 2021].
- [7] Ν. Τσουρέλης, Μελέτη και Σχεδιασμός δικτύου Internet of Things για εφαρμογές Smart Home, Σάμος: Πανεπιστήμιο Αιγαίου, 2018.
- [8] «oracle,» [Ηλεκτρονικό]. Available: <https://www.oracle.com/internet-of-things/what-is-iot/>. [Πρόσβαση 20 09 2021].
- [9] «geeksforgeeks,» [Ηλεκτρονικό]. Available: <https://www.geeksforgeeks.org/characteristics-of-internet-of-things/>. [Πρόσβαση 20 09 2021].
- [10] «robu,» [Ηλεκτρονικό]. Available: <https://robu.in/internet-of-things-iot-advantages-and-disadvantages-2021/>. [Πρόσβαση 20 09 2021].
- [11] «learn.sparkfun,» [Ηλεκτρονικό]. Available: <https://learn.sparkfun.com/tutorials/what-is-an-arduino/all>. [Πρόσβαση 20 09 2021].
- [12] «icsdweb.aegean,» [Ηλεκτρονικό]. Available: https://icsdweb.aegean.gr/aegeanrbtcs/?page_id=742. [Πρόσβαση 09 20 2021].
- [13] «arduino.cc,» [Ηλεκτρονικό]. Available: <https://www.arduino.cc/en/Main/arduinoShields>. [Πρόσβαση 20 09 2021].

- [14] «microcontrollerslab,» [Ηλεκτρονικό]. Available: <https://microcontrollerslab.com/introduction-arduino-lilypad/>. [Πρόσβαση 20 09 2021].
- [15] «circuitio,» [Ηλεκτρονικό]. Available: <https://www.circuitio.io/blog/arduino-uno-pinout/>. [Πρόσβαση 20 09 2021].
- [16] «arduino.cc,» [Ηλεκτρονικό]. Available: <https://docs.arduino.cc/hardware/uno-wifi-rev2>. [Πρόσβαση 20 09 2021].
- [17] «arduino.cc,» [Ηλεκτρονικό]. Available: <https://www.arduino.cc/en/Guide/ArduinoUnoWiFiRev2/>. [Πρόσβαση 20 09 2021].
- [18] «store.arduino.cc,» [Ηλεκτρονικό]. Available: <https://store.arduino.cc/products/arduino-uno-wifi-rev2>. [Πρόσβαση 20 09 2021].
- [19] «learn.sparkfun,» [Ηλεκτρονικό]. Available: <https://learn.sparkfun.com/tutorials/arduino-shields>. [Πρόσβαση 20 09 2021].
- [20] «elprocus,» [Ηλεκτρονικό]. Available: <https://www.elprocus.com/different-types-of-arduino-boards/>. [Πρόσβαση 20 09 2021].
- [21] «illustrationprize.com,» [Ηλεκτρονικό]. Available: <https://illustrationprize.com/el/298-photodiode.html>. [Πρόσβαση 20 09 2021].
- [22] «elprocus,» [Ηλεκτρονικό]. Available: <https://www.elprocus.com/photodiode-working-principle-applications/>. [Πρόσβαση 20 09 2021].
- [23] «ardumotive,» [Ηλεκτρονικό]. Available: <https://www.ardumotive.com/how-to-use-a-photoresistor-en.html>. [Πρόσβαση 20 09 2021].
- [24] «safewise,» [Ηλεκτρονικό]. Available: <https://www.safewise.com/resources/motion-sensor-guide/>. [Πρόσβαση 20 09 2021].
- [25] «programming-electronics,» [Ηλεκτρονικό]. Available: <https://www.programming-electronics-diy.xyz/2020/10/repair-of-outdoor-pir-d203s-thermal.html>. [Πρόσβαση 20 09 2021].
- [26] «fierceelectronics,» [Ηλεκτρονικό]. Available: <https://www.fierceelectronics.com/sensors/what-a-motion-sensor>. [Πρόσβαση 20 09 2021].
- [27] «seeedstudio,» [Ηλεκτρονικό]. Available: <https://www.seeedstudio.com/blog/2020/01/08/what-is-a-light-sensor-types-uses-arduino-guide/>. [Πρόσβαση 20 09 2021].

- [28] «amperite,» [Ηλεκτρονικό]. Available: <https://amperite.com/blog/relays/>. [Πρόσβαση 20 09 2021].
- [29] «thesciencetech,» [Ηλεκτρονικό]. Available: <https://thesciencetech.com/electronics/relay/>. [Πρόσβαση 20 09 2021].
- [30] «circuitbasics,» [Ηλεκτρονικό]. Available: <https://www.circuitbasics.com/setting-up-a-5v-relay-on-the-arduino/>. [Πρόσβαση 20 09 2021].
- [31] «arduino.cc,» [Ηλεκτρονικό]. Available: <https://www.arduino.cc/reference/en/>. [Πρόσβαση 20 09 2021].
- [32] «learn.sparkfun,» [Ηλεκτρονικό]. Available: <https://learn.sparkfun.com/tutorials/what-is-an-arduino/all>. [Πρόσβαση 20 09 2021].

ΠΗΓΕΣ ΕΙΚΟΝΩΝ.

- [i] «researchgate,» [Ηλεκτρονικό]. Available: https://www.researchgate.net/figure/Internal-components-of-the-microcontrollers_fig3_322436662. [Πρόσβαση 20 09 2021].
- [ii] «microcontroller-versus-microprocessors,» [Ηλεκτρονικό]. Available: <https://akhiljain07.wordpress.com/microcontroller-versus-microprocessors/>. [Πρόσβαση 20 09 2021].
- [iii] «https://commons.wikimedia,» [Ηλεκτρονικό]. Available: [https://commons.wikimedia.org/wiki/File:LED,_5mm,_green_\(labelled,_full\).svg](https://commons.wikimedia.org/wiki/File:LED,_5mm,_green_(labelled,_full).svg). [Πρόσβαση 20 09 2021].
- [iv] «programming-electronics» [Ηλεκτρονικό]. Available: <https://www.programming-electronics-diy.xyz/2020/10/repair-of-outdoor-pir-d203s-thermal.html>. [Πρόσβαση 20 09 2021].
- [v] «grobotronics,» [Ηλεκτρονικό]. Available: <https://grobotronics.com/photo-resistor-ldr-5mm.html?sl=en>. [Πρόσβαση 20 09 2021].
- [vi] «wikipedia» [Ηλεκτρονικό]. Available: https://en.wikipedia.org/wiki/File:Relay_Parts.jpg. [Πρόσβαση 20 09 2021].
- [vii] «arduino,» [Ηλεκτρονικό]. Available: <https://www.arduino.cc/>. [Πρόσβαση 20 09 2021].
- [viii] «courses.ntua,» [Ηλεκτρονικό]. Available: <https://www.google.com/imgres?imgurl=x-raw-image%3A%2F%2F%2Ffad4005cbe59adab4a2b1646225fbe25b624207418567309d80256f6f2220286&imgrefurl=https%3A%2F%2Fcourses.chemeng.ntua.gr%2Fdownload%2F1742&tbnid=q61ZvlwA1Pp76M&vet=12ahUKEwirgNbqvqnzAhX9JMUkHc62CG4QMyhPegUIARDfAQ..i&docid=Pq1kY5VP-2417M&w=926&h=696&q=Arduino%20Uno&hl=el&ved=2ahUKEwirgNbqvqnzAhX9JMUkHc62CG4QMyhPegUIARDfAQ>. [Πρόσβαση 20 09 2021].
- [ix] «arduino,» [Ηλεκτρονικό]. Available: <https://www.arduino.cc/en/main/boards>. [Πρόσβαση 20 09 2021].
- [x] «store-usa.arduino,» [Ηλεκτρονικό]. Available: <https://store-usa.arduino.cc/products/arduino-uno-wifi-rev2>. [Πρόσβαση 20 09 2021].
- [xi] «circuitbasics,» [Ηλεκτρονικό]. Available: <https://www.circuitbasics.com/setting-up-a-5v-relay-on-the-arduino/>. [Πρόσβαση 09 20 2021].
- [xii] «ardumotive,» [Ηλεκτρονικό]. Available: <https://www.ardumotive.com/how-to-use-a-photoresistor-en.html>. [Πρόσβαση 20 09 2021].

[xiii] «banggood,» [Ηλεκτρονικό]. Available: https://www.banggood.com/HC-SR501-Adjustable-Infrared-IR-Pyroelectric-PIR-Module-Motion-Sensor-Human-Body-Induction-Detector-p-1545488.html?utm_source=googleshopping&utm_medium=cpc_organic&gmcCountry=GR&utm_content=minha&utm_campaign=minha-gr-en-pc¤cy=EUR&warehouse=CN&createTmp=1&utm_source=googleshopping&utm_medium=cpc_bgs&utm_content=sxxx&utm_campaign=sxxx-ssc-gr-en-all-newcustom-ncv65-0409&ad_id=430724963733&gclid=CjwKCAjw49qKBhAoEiwAHQVTo-e0Byzk2ig8vDOUfM3B0X1SysKf6qoqPD3TrSM6cJEBEOWqBQXbZR0Cy5cQAvD_BwE. [Πρόσβαση 20 09 2021].