

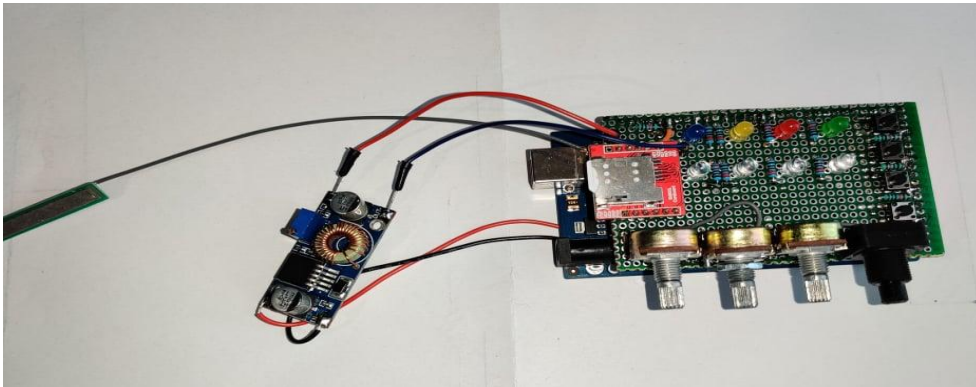


ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«ΕΛΕΓΧΟΣ ΣΥΣΚΕΥΗΣ ΜΕΣΩ SMS ΑΠΟ ΚΙΤΗΤΟ
ΤΗΛΕΦΩΝΟ ΜΕ ΤΗ ΧΡΗΣΗ GSM MODULE»



Των φοιτητών:
Παπαζαχαρίου Φώτιος
Τσινής Γεώργιος
Αρ. Μητρώου: 513140 / 513175

Επιβλέπων
Ονοματεπώνυμο:
Ιωάννης Κιοσκερίδης
Βαθμίδα Καθηγητής

Ημερομηνία 21/09/2021

Τίτλος Δ.Ε. :
ΕΛΕΓΧΟΣ ΣΥΣΚΕΥΗΣ ΜΕΣΩ SMS ΑΠΟ ΚΙΤΗΤΟ ΤΗΛΕΦΩΝΟ ΜΕ ΤΗ ΧΡΗΣΗ GSM
MODULE

Κωδικός Δ.Ε. 18143

Όνοματεπώνυμο φοιτητή/των: Παπαζαχαρίου Φώτιος, Τσινής Γεώργιος

Όνοματεπώνυμο εισηγητή: Ιωάννης Κιοσκερίδης

Ημερομηνία ανάληψης Δ.Ε. 15/03/2018

Ημερομηνία περάτωσης Δ.Ε. 21/09/2021

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία των φοιτητών Παπαζαχαρίου Φώτιο και Τσινή Γεώργιο που την εκπόνησαν. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

«Αφιέρωση»

Πρόλογος

Η επιλογή του συγκεκριμένου θέματος έγινε με βάση τις γνώσεις που είχαμε προσκομίσει τα προηγούμενα εξάμηνα από τη σχολή, το ενδιαφέρον του αντικειμένου όσον αφορά τον έλεγχο μιας συσκευής ή ακόμα και πολλών συσκευών από απόσταση καθώς και την εξέλιξη που μπορεί να έχει μια τέτοια κατασκευή στην καθημερινότητά μας τα επόμενα χρόνια. Κατά την συγγραφή της εργασίας αλλά και την υλοποίηση της κατασκευής οι γνώσεις που αποκτήσαμε είναι πολλές και σημαντικές, για τη συνέχεια της ενασχόλησης μας επαγγελματικά και μη, με το αντικείμενο της πληροφορικής αλλά και της ηλεκτρονικής ταυτόχρονα.

Περίληψη

Η παρούσα διπλωματική εργασία πραγματεύεται στη μελέτη, σχεδίαση, και υλοποίηση ενός συστήματος ελέγχου διαφόρων εισόδων-εξόδων και πραγματοποιεί την ενημέρωση-ειδοποίηση του χρήστη μέσω ενός σύντομου γραπτού μηνύματος (sms) κατά την αλλαγή κατάστασης κάθε εξόδου. Ταυτόχρονα ο χρήστης μπορεί να ενεργοποιήσει, να απενεργοποιήσει αλλά και να μάθει την κατάσταση όλων των εξόδων του συστήματος στέλνοντας sms. Η υλοποίηση γίνεται μέσω της πλακέτας ανάπτυξης Arduino και συγκεκριμένα του MEGA2560. Επίσης τοποθετούνται τέσσερις αναλογικές και τέσσερις ψηφιακές εισοδοί (ποτενσιόμετρα – button) αλλά και οχτώ ψηφιακές έξοδοι τύπου LED. Η επικοινωνία του χρήστη με το σύστημα πραγματοποιείται με ένα GSM module όπου συνδέεται στο κυψελοειδές ψηφιακό σύστημα κινητής τηλεφωνίας προκειμένου να επιτευχθεί η επικοινωνία συστήματος – χρήστη.

BACHELOR'S THESIS

«DEVICE CONTROL VIA SMS FROM MOBILE PHONE USING GSM MODULE»

«Fotis Papazachariou, George Tsinis»

Abstract

This thesis deals with the study, design, and implementation of a control system of various inputs-outputs and carries out the information-notification of the user through a short text message (sms) when changing the status of each output. At the same time the user can activate, deactivate and also find out the status of all system outputs by sending sms. The implementation takes place through the Arduino development board and specifically MEFA2560. They are also placed four (4) analog and four (4) digital inputs (potentiometers-button) and eight (8) digital LED outputs. The user communicates with the system by a GSM module where it connects to the cellular digital mobile telephony system in order to achieve system-user communication.

Ευχαριστίες

Θα θέλαμε να ευχαριστήσουμε ιδιαιτέρως τον καθηγητή με τον οποίο αναλάβαμε το συγκεκριμένο θέμα Διπλωματικής Εργασίας, τον Κ. Κιοσκερίδη Ιωάννη. Παράλληλα θα θέλαμε να ευχαριστήσουμε και το τμήμα μηχανικών πληροφορικής και ηλεκτρονικών συστημάτων του ΔΙ.ΠΑ.Ε. για τις παροχές που μας προσέφερε.

Περιεχόμενα

Πρόλογος.....	ii
Περίληψη.....	iii
Abstract	iv
Ευχαριστίες	v
Περιεχόμενα	vi
Κατάλογος Σχημάτων	x
Κατάλογος Πινάκων.....	xi
Συνομογραφίες.....	xii
Κεφάλαιο 1ο: Απομακρυσμένος έλεγχος.....	1
1.1 Εισαγωγή.....	1
1.2 Ασύρματη επικοινωνία.....	1
1.3 Ασύρματος έλεγχος συσκευών.....	3
1.4 Το Ευρωπαϊκό πρότυπο GSM.....	3
1.5 Αρχιτεκτονική GSM.....	4
1.5.1 Υποσύστημα BSS.....	5
1.5.2 Υποσύστημα NSS.....	5
1.5.3 Υποσύστημα OSS.....	6
1.6 Επίλογος.....	6
Κεφάλαιο 2ο: Συστήματα Μικροελεγκτών	6
2.1 Εισαγωγή.....	6
2.2 Arduino.....	7
2.2.1 Επίσημες πλακέτες Arduino	7
2.3 Εφαρμογές του Arduino	8
2.4 Λειτουργία του Arduino	9
2.5 Επίλογος.....	9
Κεφάλαιο 3ο: Arduino MEGA2560.....	10
3.1 Εισαγωγή.....	10
3.2 Ανάλυση ακροδεκτών του Arduino Mega2560	11
3.2.1 Ακροδέκτες Ρεύματος	11
3.2.2 Αναλογικοί Ακροδέκτες.....	11
3.2.3 Ψηφιακοί Ακροδέκτες.....	11

3.3	Περιγραφή Arduino.....	12
3.4	Επίλογος.....	13
Κεφάλαιο 4ο: Σχεδίαση του Συστήματος.....		13
4.1	Εισαγωγή.....	13
4.2	Ποτενσιόμετρο	14
4.3	Φωτοδίοο LED	14
4.4	Διακόπτης Μπουτόν.....	15
4.5	Αντιστάτης	15
4.6	GSM/GPRS module	15
4.6.1	SIM800L	16
4.7	Step-down μονάδα μετατροπέα τάσης DC-DC.....	17
4.8	Breadboard	19
4.9	Διατηρητή πλακέτα (prototype board)	19
4.10	Τροφοδοτικό.....	20
4.11	Επίλογος.....	20
Κεφάλαιο 5ο: Κατασκευή του Συστήματος		20
5.1	Εισαγωγή.....	20
5.2	Σύνδεση Εξαρτημάτων.....	21
5.3	Τροφοδοσία.....	22
5.4	GSM Module.....	22
5.5	Ψηφιακές Έξοδοι.....	22
5.6	Αναλογικές Είσοδοι	22
5.7	Ψηφιακές Είσοδοι.....	23
5.8	Ράστερ και Διατηρητή πλακέτα.....	23
5.9	Κόστος υλοποίησης κατασκευής.....	25
5.10	Επίλογος.....	26
Κεφάλαιο 6ο: Ανάλυση και τρόπος λειτουργίας του Συστήματος.....		26
6.1	Εισαγωγή.....	26
6.2	Προεργασία	26
6.3	Αποστολή SMS	27
6.3.1	Επικοινωνία με Αναλογικές εισόδους.....	28
6.3.2	Επικοινωνία με Ψηφιακές εισόδους.....	28
6.4	Επίλογος.....	29
Κεφάλαιο 7ο: SMS (Short Message Service).....		30

7.1	Εισαγωγή.....	30
7.2	Το SMS και η διαδικασία της ασύρματης επικοινωνίας.....	30
7.3	Αποστολή sms μέσω Ηλεκτρονικού Υπολογιστή με GSM μόντεμ.....	31
7.4	Λήψη sms μέσω Ηλεκτρονικού Υπολογιστή με GSM μόντεμ.....	32
7.5	Εντολές AT.....	32
7.6	Τρόποι λειτουργίας των GSM μόντεμ/ κινητών συσκευών.....	32
7.7	Επίλογος.....	34
Κεφάλαιο 8ο: Προγραμματιστικός Κώδικας Arduino.....		34
8.1	Εισαγωγή.....	34
8.2	Προεργασία προγραμματισμού.....	34
8.3	Διαγράμματα ροής προγράμματος.....	35
8.3.1	Η βασική δομή.....	35
8.3.2	Η συνάρτηση Setup().....	36
8.3.3	Η συνάρτηση loop().....	37
8.3.4	Η συνάρτηση readbutton().....	39
8.3.5	Η συνάρτηση analogin().....	41
8.3.6	Η συνάρτηση res().....	42
8.4	Ανάλυση sketch της εφαρμογής.....	42
8.4.1	Εισαγωγή βιβλιοθηκών.....	42
8.4.2	Αρχικοποίηση – Ορισμός μεταβλητών.....	43
8.4.3	Η συνάρτηση setup().....	44
8.4.4	Η συνάρτηση res().....	44
8.4.5	Η συνάρτηση analogin().....	45
8.4.6	Η Συνάρτηση readbutton().....	46
8.4.7	Η συνάρτηση loop().....	47
8.4.8	Έλεγχος σφαλμάτων κατά την αποστολή sms.....	48
8.5	Επίλογος.....	49
Κεφάλαιο 9ο: Συμπεράσματα και προτάσεις βελτίωσης.....		49
9.1	Εισαγωγή.....	49
9.2	Συμπεράσματα.....	50
9.3	Πιθανά προβλήματα.....	50
9.4	Προτάσεις βελτίωσης.....	51
9.5	Τομείς εφαρμογής της κατασκευής.....	52
9.6	Προβλήματα κατασκευής του συστήματος.....	52

9.7	Επίλογος.....	52
-----	---------------	----

Κατάλογος Εικόνων

Εικόνα 1.1: Αρχιτεκτονική GSM	Σφάλμα! Δεν έχει οριστεί σελιδοδείκτης.
Εικόνα 2.1 Διασύνδεση Αισθητήρων με το Arduino.....	9
Εικόνα 3.1 Ανάλυση των ακροδεκτών Arduino Mega2560.....	12
Εικόνα 4.1 Ποτενσιόμετρο.....	14
Εικόνα 4.2 Σύμβολο ποτενσιόμετρου.....	14
Εικόνα 4.3 Δίοδος LED.....	14
Εικόνα 4.4 Διακόπτης μπουτόν.....	15
Εικόνα 4.5 Χρωματικός Κώδικας Αντιστάσεων.....	15
Εικόνα 4.6 GSM Module SIM800L.....	17
Εικόνα 4.7 Step-down Μονάδα DC-DC.....	18
Εικόνα 4.8 Breadboard - Ράστερ.....	19
Εικόνα 4.9 Διατηρητή Πλακέτα.....	19
Εικόνα 4.10 Τροφοδοτικό.....	20
Εικόνα 5.1 Σχηματική Απεικόνιση Συνδεσμολογίας του Συστήματος.....	21
Εικόνα 5.2 Τελική Μορφή του Συστήματος.....	24
Εικόνα 5.3 Σύνδεση Step-down Μονάδας DC-DC.....	25
Εικόνα 7.1 Διαδικασία Ασύρματης Επικοινωνίας.....	30
Εικόνα 8.1 Διάγραμμα Ροής Βασικής Δομής του Προγράμματος.....	36
Εικόνα 8.2 Διάγραμμα Ροής της Συνάρτησης Setup().....	37
Εικόνα 8.3 Διάγραμμα Ροής της Συνάρτησης loop().....	38
Εικόνα 8.4 Διάγραμμα Ροής της Συνάρτησης readbutton().....	40
Εικόνα 8.5 Διάγραμμα της Συνάρτησης analogin().....	41
Εικόνα 8.6 Διάγραμμα Ροής Βασικής της Συνάρτησης res().....	42

Κατάλογος Πινάκων

Πίνακας 2.1 Τεχνικά Χαρακτηριστικά Arduino Mega2560.....	10
Πίνακας 5.1 Κόστος Εργασίας.....	26
Πίνακας 7.1 Εντολές AT κατά την Αποστολή.....	31
Πίνακας 7.2 Εντολές AT κατά τη Λήψη.....	32
Πίνακας 7.3 Αντιστοιχία Text και PDU mode.....	33

Συντομογραφίες

Δ.Ε. - Διπλωματική Εργασία
ΔΙΠΑΕ - Διεθνές Πανεπιστήμιο Ελλάδος
Η/Υ – Ηλεκτρονικός Υπολογιστής
AC - Alternating current
AUC - Authentication Center
BCS - Base Control Station
BSS - Base Station Subsystem
BTS - Base Transceiver Station
DC - Direct current
EIR - Equipment Identity Register
GND - Ground
GSM - Global System for Mobile
HLR - Home Location Register
ICSP – In Circuit Serial Programming
IDE - Integrated Development Environment
IMSI - International Mobile Subscriber Identity
IoT - Internet of Things
LED - Light Emitting Diode
MS - Mobile Station
MSC - Mobile Services Switching Center
NSS - Network and Switching Subsystem
OMC - Operation Maintenance Centres
OSS - Operation Support Subsystem
PDU - Protocol Data Unit
PWM – Pulse Width Modulation
SCCP - Signalling Correction Control Part
SIM - *Subscriber Identity Module*
SMS - Short Message Service
SS7 - Signalling System No. 7
UART - Universal Asynchronous Receiver and Transmitter
USB - Universal Serial Bus
VLR - Visitor Location Register

Κεφάλαιο 1ο: Απομακρυσμένος έλεγχος

1.1 Εισαγωγή

Στο πρώτο κεφάλαιο της εργασίας θα αναπτύξουμε την διαδικασία του ελέγχου αλλά και της επικοινωνίας με ασύρματα μέσα και τα πλεονεκτήματα της σε σχέση με την ενσύρματη. Θα αναφερθούν οι τρόποι με τους οποίους μπορεί ο κάθε χρήστης να επικοινωνεί και να ελέγχει απομακρυσμένα μια συσκευή. Στο τέλος του κεφαλαίου θα γίνει μια εισαγωγική αναφορά στα GSM, τι είναι και σε ποια η χρησιμότητα τους.

1.2 Ασύρματη επικοινωνία

Ο όρος επικοινωνία αναφέρεται στη διαδικασία μεταφοράς πληροφοριών και ιδεών μεταξύ ενός ή πολυπληθών αποστολέων και ενός ή περισσότερων παραληπτών. Πολλές φορές χρειαζόμαστε να επικοινωνήσουμε απομακρυσμένα με κάποιο δέκτη. Οι μέθοδοι που χρησιμοποιούνται για να το επιτύχουμε αυτό εξαρτώνται από την εποχή και από τα μέσα που διαθέτουμε. Από τα τέλη του 19^{ου} αιώνα με την ανακάλυψη του ασυρμάτου, υπήρξε ραγδαία εξέλιξη στο τομέα των ασύρματων επικοινωνιών, χάρη στην οποία μπορούμε να μεταφέρουμε πληροφορίες απομακρυσμένα μέσω ραδιοκυμάτων. Ως αποτέλεσμα στις ασύρματες επικοινωνίες δεν χρειαζόμαστε ηλεκτρικούς αγωγούς ή καλώδια για να μεταφέρουμε την πληροφορία. Στην παρούσα περίοδο, το πιο διαδεδομένο μέσο για ασύρματη επικοινωνία είναι το κινητό τηλέφωνο – smartphone, με εκατομμύρια χρήστες παγκοσμίως. Χάρη στη κινητή τεχνολογία ένα άτομο μπορεί να μεταδώσει ή να παραλάβει πληροφορίες έχοντας την ελευθερία των κινήσεων του. Επίσης αποφεύγουμε τα αρνητικά στοιχεία που θα έχουμε με τη χρήση καλωδίου, όπως για παράδειγμα η διαδικασία της εγκατάστασης της συσκευής, που με τα καλώδια θα ήταν πιο δύσκολη και χρονοβόρα ή τις πιθανές βλάβες που μπορούν να προκύψουν στο καλώδιο από φυσικά αίτια. Τα κινητά επικοινωνούν με σταθερούς σταθμούς βάσης, οι οποίοι είναι συνδεδεμένοι σε ένα σταθερό δίκτυο κορμού. Η ευκολία λοιπόν που προκύπτει στον έλεγχο μιας συσκευής από ένα κινητό τηλέφωνο αλλά και η αμεσότητα που προσδίδει ή φορητή συσκευή μεταξύ χρήστη και κατασκευής ήταν βασικός παράγοντας που οδήγησε στην ανάπτυξη του συγκεκριμένου θέματος αλλά και στη δημιουργία της συγκεκριμένης διπλωματικής εργασίας. Παρακάτω καταγράφεται ένας πίνακας ορισμών όπου περιέχει τις βασικές έννοιες που χρησιμοποιούνται για να περιγράψουμε ένα σύστημα κινητής τηλεφωνίας.

Όρος	Περιγραφή
Σταθμός Βάσης Base Station (BS)	Ένας σταθερός σταθμός σε ένα δίκτυο κινητής τηλεφωνίας που χρησιμοποιείται για επικοινωνία με τους κινητούς σταθμούς. Οι σταθμοί βάσης τοποθετούνται στο κέντρο ή την άκρη της περιοχής κάλυψης και αποτελούνται από κανάλια ραδιοεπικοινωνίας και κεραίες εκπομπής και λήψης τοποθετημένες σε κατάλληλα διαμορφωμένο ιστό.
Κινητός Σταθμός Mobile Stations (MS)	Σταθμός εντασσόμενος στα πλαίσια δικτύου κυψελωτής τηλεφωνίας που μπορεί να χρησιμοποιηθεί εν κινήσει και σε μη προκαθορισμένες τοποθεσίες.
Δίαυλος Ελέγχου Control Channel	Δίαυλοι ραδιοεπικοινωνίας που χρησιμοποιούνται για την μετάδοση πληροφορίας που αφορά στην προετοιμασία, αίτηση και έναρξη της κλήσης και άλλων παραμέτρων ελέγχου και σηματοδοσίας μεταξύ σταθμού βάσης και κινητού.
Καθοδικός Δίαυλος Forward Channel	Δίαυλος ραδιοεπικοινωνίας που χρησιμοποιείται για την μετάδοση πληροφορίας από το σταθμό βάσης προς το κινητό.
Ανοδικός Δίαυλος Reverse Channel	Δίαυλος ραδιοεπικοινωνίας που χρησιμοποιείται για την μετάδοση πληροφορίας από το κινητό προς το σταθμό βάσης.
Κέντρο Μεταγωγής Mobile Switching Centre (MSC)	Κέντρο μεταγωγής που συντονίζει την δρομολόγηση κλήσεων στα όρια μιας περιοχής εξυπηρέτησης ενός δικτύου κινητής τηλεφωνίας. Το κέντρο μεταγωγής συνδέει το συγκεκριμένο δίκτυο κινητής τηλεφωνίας με άλλα σταθερά ή κινητά δίκτυα.
Μεταπομπή Handoff	Η διαδικασία της μεταφοράς ενός κινητού σταθμού από ένα κανάλι ή σταθμό βάσης σε κάποιο άλλο ή άλλον, αντίστοιχα.
Συνδρομητής Subscriber	Ένας χρήστης ο οποίος καταβάλλει συνδρομητικά τέλη σε ένα δίκτυο κινητής τηλεφωνίας για να το χρησιμοποιήσει.

Πίνακας 1.1 Ορισμοί Συστήματος Κινητής Τηλεφωνίας

Μπορούμε να ταξινομήσουμε τα συστήματα κινητής επικοινωνίας σε τρεις μεγάλες κατηγορίες. Η πρώτη κατηγορία είναι τα Συστήματα μονοδρομικής κατεύθυνσης. Σε αυτά τα συστήματα η επικοινωνία είναι εφικτή μόνο προς μία κατεύθυνση. Η δεύτερη κατηγορία είναι τα Συστήματα Ημι-αμφίδρομης κατεύθυνσης. Τα ημι-αμφίδρομα συστήματα επιτρέπουν την αμφίδρομη επικοινωνία, ωστόσο γίνεται χρήση κοινού ραδιοδιαύλου τόσο για την εκπομπή, όσο και για την λήψη. Αυτό έχει ως αποτέλεσμα ο χρήστης να μπορεί είτε να λαμβάνει είτε να αποστέλλει ένα μήνυμα ή μία εντολή σε μια δεδομένη χρονική στιγμή. Η τρίτη κατηγορία είναι τα Συστήματα Αμφίδρομης κατεύθυνσης. Στη κατηγορία αυτή, τα συστήματα αμφίδρομης κατεύθυνσης, η διαδικασία λήψης και αποστολής πληροφορίας μπορεί να γίνεται ταυτόχρονα και από τα δύο άκρα του συστήματος, χάρη στη χρήση δυο ξεχωριστών ραδιοδιαύλων.

1.3 Ασύρματος έλεγχος συσκευών

Οι ανταλλαγές πληροφοριών δεν είναι απαραίτητο να εφαρμόζεται μονάχα μεταξύ ανθρώπων. Με τα κατάλληλα συστήματα μπορούμε να προγραμματίσουμε μια συσκευή να επικοινωνεί μαζί μας, δίνοντας μας πληροφορίες που είναι χρήσιμες στο χρήστη ή ακόμη και να εκτελεί μια ενέργεια, λαμβάνοντας απομακρυσμένα την σωστή εντολή. Το κατάλληλο μέσο για να λαμβάνουμε και να στέλνουμε εντολές σε μια συσκευή απομακρυσμένα, είναι το κινητό τηλέφωνο. Υπάρχουν πολλοί τρόποι με τους οποίους ένα κινητό τηλέφωνο μπορεί να επικοινωνήσει με μία άλλη συσκευή. Η τεχνολογία επικοινωνίας είναι το GSM, το GPRS, το WiFi, το 3G, το 4G και στις μέρες μας πλέον το 5G. Η κάθε τεχνολογία διαφέρει από την άλλη κυρίως στο λογισμικό επικοινωνιών, δηλαδή, στις εφαρμογές ή στα προγράμματα που είναι σχεδιασμένα για να μεταφέρουν τις πληροφορίες μεταξύ των συσκευών. Στη συγκεκριμένη Διπλωματική Εργασία η τεχνολογία που επιλέχθηκε είναι η επικοινωνία των συσκευών μέσω GSM και συγκεκριμένα ενός GSM module. Αυτό διότι στην καθημερινότητά του ανθρώπου πλέον το κινητό τηλέφωνο βρίσκεται παντού και πάντα. Ωστόσο η τεχνολογία WiFi ή το 4G ακόμα και το 5G για παράδειγμα, δεν είναι διαθέσιμο για όλους ανά πάσα στιγμή. Με τη τεχνολογία GSM όμως η αποστολή ενός μηνύματος ή η πραγματοποίηση μιας κλήσης μέσω του κινητού τηλεφώνου μπορεί να είναι εφικτή από τον καθένα, από οπουδήποτε και οποτεδήποτε. Επομένως θεωρήθηκε η ιδανική τεχνολογία για την ανάπτυξη της κατασκευής.

1.4 Το Ευρωπαϊκό πρότυπο GSM

Με την άνοδο των ασύρματων επικοινωνιών και συστημάτων, δημιουργήθηκε η ανάγκη να εφαρμοστεί ένα γενικό πρότυπο που θα καθορίζει τον σχεδιασμό του συστήματος και τις λειτουργίες του. Για αυτό το σκοπό αναπτύχθηκε το πρότυπο GSM που σημαίνει Global Standard for Mobile, και αποτελεί ένα πρότυπο 2^{ης} γενιάς ψηφιακών κινητών επικοινωνιών που δημιούργησε η εταιρεία ETSI, με σκοπό να εφαρμοστεί από όλα τα ευρωπαϊκά κράτη. Οι υπηρεσίες GSM ακολουθούν τις κατευθυντήριες γραμμές του ολοκληρωμένου δικτύου ψηφιακών υπηρεσιών (Integrated Services Digital Network – ISDN) και κατηγοριοποιούνται σε τηλεπηρεσίες και υπηρεσίες δεδομένων. Τηλεπηρεσίες χαρακτηρίζονται όλες οι τηλεπικοινωνιακές κινήσεις που προέρχονται είτε από το σταθμό βάσης, είτε από το κινητό. Οι υπηρεσίες δεδομένων είναι αυτές που κάνουν επικοινωνίες με

υπολογιστές και κάνουν χρήση του διαδικτύου. Ο χρήστης έχει επίσης στη διάθεση του τις συμπληρωματικές υπηρεσίες ISDN, όπως εκτροπή κλήσης, αναγνώριση κλήσης, την υπηρεσία αποστολής γραπτών μηνυμάτων που έχει και άμεση χρήση για εφαρμογές ασύρματου ελέγχου, και άλλες. Ένα ιδιαίτερα σπουδαίο χαρακτηριστικό του προτύπου GSM είναι η κάρτα ταυτότητας συνδρομητή (Subscriber Identity Module – SIM), που είναι ουσιαστικά ένα στοιχείο μνήμης που αποθηκεύει τον αριθμό ταυτότητας του συνδρομητή, τα δίκτυα και τις χώρες όπου ο συνδρομητής έχει την δυνατότητα να εξυπηρετηθεί, κωδικούς ασφαλείας και γενικά προσωπικού χαρακτήρα πληροφορία για τον χρήστη. Η κάρτα SIM είναι φορητή και μπορεί να τοποθετηθεί σε οποιαδήποτε GSM συμβατή συσκευή, η οποία ενεργοποιείται με την πληκτρολόγηση ενός τετραψήφιου κωδικού. Ένα άλλο αξιοσημείωτο χαρακτηριστικό του GSM είναι η ασφάλεια που παρέχει το σύστημα λόγω της χρήσης εξελιγμένης ψηφιακής κωδικοποίησης στη ραδιοδιεπαφή (στην επικοινωνία δηλαδή μεταξύ σταθμού βάσης και κινητού τηλεφώνου). Το μυστικό κλειδί κρυπτογράφησης είναι γνωστό μόνο στον πάροχο και αλλάζει με τον χρόνο για κάθε χρήστη. Κάθε πάροχος υπηρεσιών κινητής τηλεφωνίας και κατασκευαστής εξοπλισμού GSM πρέπει να υπογράψει το Πρωτόκολλο Συνεργασίας (Memorandum of Understanding – MoU) πριν αναπτύξει δίκτυο ή κατασκευάσει εξοπλισμό, αντίστοιχα. Το Πρωτόκολλο Συνεργασίας αποτελεί διεθνή συμφωνία που επιτρέπει την κοινή γνώση των αλγορίθμων κρυπτογράφησης και άλλης εμπιστευτικής πληροφορίας μεταξύ χωρών και παρόχων.

1.5 Αρχιτεκτονική GSM

Τα τρία βασικά υποσυστήματα που αποτελούν την αρχιτεκτονική του πρότυπου GSM, είναι το υποσύστημα σταθμού βάσης (Base Station Subsystem – BSS), το υποσύστημα δικτύου και μεταγωγής (Network and Switching Subsystem – NSS), και το υποσύστημα υποστήριξης λειτουργιών (Operation Support Subsystem – OSS). Επίσης υπάρχει και ο κινητός σταθμός (Mobile Station – MS), ο οποίος αποτελεί ένα ξεχωριστό υποσύστημα αλλά κατά σύμβαση το θεωρούμε τμήμα του υποσυστήματος BSS. Τα υποσυστήματα αυτά αλληλεπιδρούν μεταξύ τους με διασυνδέσεις, καθώς επίσης και με τους χρήστες δια μέσου ειδικών δικτύων διεπαφών. Στην παρακάτω εικόνα απεικονίζεται η συγκεκριμένη αρχιτεκτονική σε διάγραμμα.

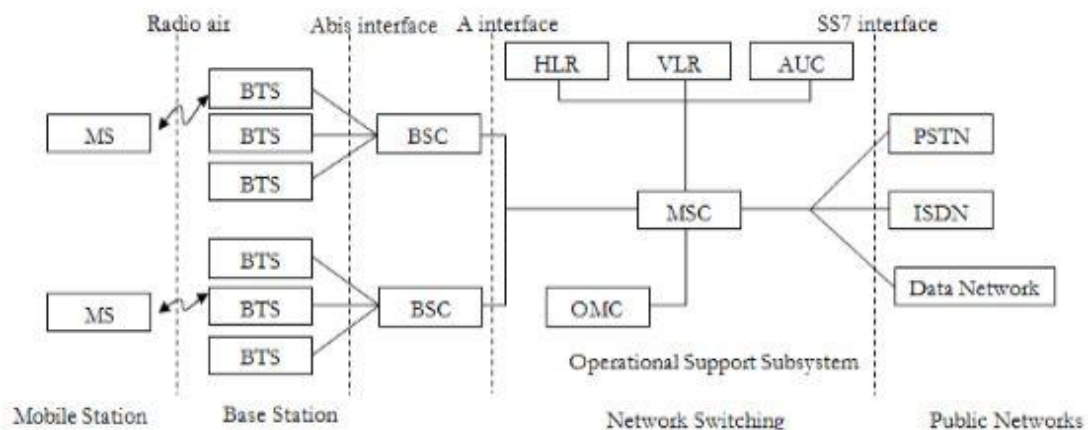


Fig: GSM Architecture

Εικόνα 1.1 Αρχιτεκτονική GSM

1.5.1 Υποσύστημα BSS

Το υποσύστημα BSS διαχειρίζεται τις οδούς επικοινωνίας μεταξύ των κινητών σταθμών και του κέντρου μεταγωγής και επίσης την ραδιοεπαφή μεταξύ των κινητών σταθμών και των υπόλοιπων υποσυστημάτων του GSM. Χωρίζεται σε δύο μέρη, τον σταθμό πομποδέκτη βάσης (Base Transceiver Station – BTS) και το σταθμό ελέγχου βάσης (Base Control Station – BCS), οι οποίοι συνδέονται μεταξύ τους χάρη στη διεπαφή Abis. Η Abis-διεπαφή μεταφέρει τηλεπικοινωνιακή κίνηση και δεδομένα ελέγχου και είναι προτυποποιημένη από το GSM. Έτσι BTS και BSS εξοπλισμός που προέρχεται από διαφορετικούς κατασκευαστές μπορεί να διασυνδεθεί λόγω της προτυποποίησης της διεπαφής. Το BTS φιλοξενεί τους ραδιοφωνικούς πομποδέκτες που ορίζουν ένα κελί και χειρίζονται τα πρωτόκολλα ραδιοσύνδεσης με το MS. Σε μια μεγάλη αστική περιοχή μπορεί να αναπτυχθεί μεγάλος αριθμός BTS. Οι λειτουργίες του BTS εντοπίζονται κυρίως στο φυσικό στρώμα και από αυτή την άποψη δεν διαθέτει την “νοημοσύνη” να διαχειριστεί κάποιες από τις υψηλότερου επιπέδου λειτουργίες του δικτύου. Αυτός είναι ο λόγος που πολλά BTS ελέγχονται από ένα τηλεπικοινωνιακό κόμβο που ονομάζεται BSC. Οι σταθμοί ελέγχου βάσης BCS ελέγχουν τους BTS σταθμούς και συνδέουν το MS με το NSS δια μέσου του Mobile Services Switching Center – MSC, η σύνδεση τους γίνεται είτε με μικροκυματικές ζεύξεις είτε με κυκλώματα μισθωμένων γραμμών. Η διεπαφή μεταξύ ενός BSC και ενός MSC ονομάζεται A-διεπαφή και είναι επίσης προτυποποιημένη στο πρότυπο GSM. Η A-διεπαφή χρησιμοποιεί ένα πρωτόκολλο SS7 (Signalling System No. 7) που ονομάζεται Τμήμα Ελέγχου Διόρθωσης Σηματοδοσίας (Signalling Correction Control Part – SCCP), το οποίο υποστηρίζει την επικοινωνία μεταξύ του MSC και του BSS, όπως επίσης και την αποστολή μηνυμάτων δικτύου μεταξύ μεμονωμένων χρηστών και του MSC. Η A-διεπαφή επιτρέπει στους παρόχους να χρησιμοποιούν σταθμούς βάσης και εξοπλισμό μεταγωγής κατασκευασμένο από διαφορετικούς κατασκευαστές.

1.5.2 Υποσύστημα NSS

Το υποσύστημα NSS διαχειρίζεται τις διαδικασίες μεταπομπής του συστήματος και επιτρέπει στο MSC να επικοινωνεί με άλλα δίκτυα. Το MSC είναι η κεντρική μονάδα του NSS και ελέγχει την τηλεφωνική κίνηση όλων των BSCs. Οι τρεις βάσεις που υπάρχουν στο NSS ονομάζονται Home Location Register – HLR, Visitor Location Register – VLR και Authentication Center – AUC. Η βάση δεδομένων HLR περιέχει πληροφορίες σχετικά με τον συνδρομητή και τη θέση κάθε χρήστη που ευρίσκεται στην περιοχή ελέγχου του οικείου MSC. Στο πρότυπο GSM αποδίδεται σε κάθε συνδρομητή ένας μοναδικός αριθμός που ονομάζεται International Mobile Subscriber Identity (IMSI) και χρησιμοποιείται για την ταυτοποίησή του. Η βάση δεδομένων VLR αποθηκεύει προσωρινά πληροφορίες και της ταυτότητα IMSI κάθε περιαγόμενου χρήστη που δραστηριοποιείται σε περιοχή που ελέγχεται από το μη οικείο του MSC ως επισκέπτης. Από τη στιγμή που κάποιο κινητό εγγραφεί στο VLR, το αρμόδιο MSC ενημερώνει σχετικά την HLR του συνδρομητή-επισκέπτη, έτσι ώστε οι κλήσεις που αυτός πραγματοποιεί να δρομολογηθούν κατάλληλα εντός ή εκτός του δικτύου. Η βάση AUC είναι αυστηρά προστατευμένη και χειρίζεται τα κλειδιά αναγνώρισης και κρυπτογράφησης για κάθε συνδρομητή εγγεγραμμένο στις βάσεις HLR και VLR. Τέλος, η βάση AUC περιέχει ένα καταχωρητή που ονομάζεται Equipment Identity Register (EIR) ο οποίος καταγράφει κλεμμένες ή παρανόμως τροποποιημένες κινητές συσκευές που εκπέμπουν στοιχεία ταυτότητας που δεν ταιριάζουν με τα στοιχεία που περιέχονται στις βάσεις HLR και VLR.

1.5.3 Υποσύστημα OSS

Το υποσύστημα OSS υποστηρίζει ένα ή περισσότερα Operation Maintenance Centres (OMC), τα οποία εποπτεύουν και διατηρούν την ποιοτική απόδοση κάθε MS, BTS, BSC και MSC εντός του δικτύου GSM. Επίσης το OSS διάγει τρεις βασικές λειτουργίες:

- συντήρηση όλου του τηλεπικοινωνιακού εξοπλισμού και όλων των λειτουργιών δικτύου.
- διαχείριση όλων των διαδικασιών χρέωσης και τιμολόγησης.
- διαχειρίζεται όλο τον κινητό εξοπλισμό στο δίκτυο.

1.6 Επίλογος

Στο κεφάλαιο που ακολούθησε έγινε ανάλυση και η αναφορά των πλεονεκτημάτων στη διαδικασία του ελέγχου αλλά και της επικοινωνίας με ασύρματα μέσα, αναλύθηκε ο ασύρματος έλεγχος συσκευών αλλά και οι τεχνολογίες που διαθέτει και πιο συγκεκριμένα επισημάνθηκε η τεχνολογία GSM αφού είναι και η τεχνολογία που επιλέχτηκε για την κατασκευή της συσκευής της εργασίας.

Κεφάλαιο 2ο: Συστήματα Μικροελεγκτών

2.1 Εισαγωγή

Ο μικροελεγκτής (microcontroller) είναι ένας τύπος επεξεργαστή ο οποίος μπορεί να παράγει έργο με ελάχιστα εξωτερικά εξαρτήματα λόγω των πολλών ενσωματωμένων συστημάτων που διαθέτει. Χρησιμοποιείται ευρύτατα σε όλα τα ενσωματωμένα συστήματα ελέγχου χαμηλού και μεσαίου κόστους, όπως αυτά που χρησιμοποιούνται σε αυτοματισμούς, ηλεκτρονικά καταναλωτικά προϊόντα (από ψηφιακές φωτογραφικές μηχανές έως παιχνίδια), ηλεκτρικές συσκευές και κάθε είδους αυτοκινούμενα τροχοφόρα οχήματα. Η ευελιξία ανάπτυξης διαφορετικών εφαρμογών είναι μεγάλη, καθώς η λειτουργικότητα του τελικού συστήματος καθορίζεται από τα εξωτερικά περιφερειακά τα οποία διασυνδέονται με την κεντρική μονάδα. Οι μικροελεγκτές δίνουν έμφαση στο μικρό αριθμό ολοκληρωμένων κυκλωμάτων που απαιτείται για τη λειτουργία μιας συσκευής, το χαμηλό κόστος και την εξειδίκευση. Πιο συγκεκριμένα, το Arduino είναι ένας μικροελεγκτής μονής πλακέτας, δηλαδή μια απλή μητρική πλακέτα ανοικτού κώδικα με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους, η οποία μπορεί να προγραμματιστεί με τη γλώσσα Wiring (ουσιαστικά πρόκειται για τη γλώσσα προγραμματισμού C++ και ένα σύνολο από βιβλιοθήκες, υλοποιημένες επίσης στην C++). Το Arduino μπορεί να χρησιμοποιηθεί για την ανάπτυξη ανεξάρτητων διαδραστικών αντικειμένων αλλά και να συνδεθεί με υπολογιστή μέσω προγραμμάτων. Στην αγορά υπάρχουν πολλές εκδόσεις του Arduino που μπορούν να αγοραστούν προ-συναρμολογημένες. Τα διαγράμματα αλλά και πληροφορίες για το υλικό είναι ελεύθερα διαθέσιμα για αυτούς που θέλουν να συναρμολογήσουν το Arduino μόνοι τους. Φυσικά υπάρχουν και άλλες πλατφόρμες και υλοποιήσεις που μπορούν να κάνουν τα ίδια πράγματα. Ωστόσο υπάρχει μια σημαντική διαφορά. Το Arduino βασίζεται σε τεχνολογίες ανοιχτού κώδικα. Μπορεί δηλαδή να κατασκευαστεί εύκολα από τον καθένα, μπορεί να ενσωματωθεί σε συσκευές, ακόμα και για εμπορικούς σκοπούς, και το σημαντικότερο είναι ότι υπάρχει μια ολόκληρη κοινότητα που χρησιμοποιεί το Arduino σε κατασκευές, άρα υπάρχει μεγάλος όγκος ελεύθερης πληροφορίας για ευκολότερη υλοποίηση εργασιών. Γενικά, τα Projects στον εν λόγω Μικροελεγκτή μπορούν να είναι αυτόνομα (σε επίπεδο hardware) ή να επικοινωνούν με κάποιο software στον Η/Υ του προγραμματιστή.

2.2 Arduino

Ο μικροεπεξεργαστής ενός Arduino συνήθως προγραμματίζεται εκ των προτέρων ώστε να παρέχει κάποιο φορτωτή εκκίνησης (BootLoader). Ο φορτωτής εκκίνησης υπάρχει ώστε να απλοποιεί την διαδικασία της αποθήκευσης των προγραμμάτων στην Flash Memory του Arduino μέσω σειριακής USB θύρας. Επιπλέον, η γλώσσα προγραμματισμού, οι διάφορες βιβλιοθήκες και το ολοκληρωμένο περιβάλλον ανάπτυξης που υπάρχουν για τον προγραμματισμό της πλατφόρμας Arduino αποτελούν ανοιχτό λογισμικό προσφέροντας έτσι ανεκτίμητη γνώση σε όλους. Κάποια από τα βασικά προτερήματα που οδηγεί τον χρήστη στην αγορά και υλοποίηση εργασιών με Arduino είναι ότι η συγκεκριμένη πλατφόρμα είναι μια αρκετά οικονομική λύση και παράλληλα αρχιτεκτονικά ανοικτή, ώστε να μπορεί ο οποιοσδήποτε να την προγραμματίσει – αναπτύξει μόνος του. Είναι μια πλατφόρμα που παρέχει πλήρη μεταφερισιμότητα με αποτέλεσμα να μπορεί να προγραμματιστεί στα περισσότερα λειτουργικά συστήματα. Τέλος λόγω του λογισμικού της, που είναι ανοικτό και ελεύθερο για όλους, αναπτύσσονται καθημερινά διάφορες βιβλιοθήκες για την υποστήριξη της πλατφόρμας, με αποτέλεσμα το υλικό αλλά και η αρχιτεκτονική του Arduino να εξελίσσονται συνεχώς. Κάθε πλακέτα Arduino αποτελείται από δύο τμήματα. Το πρώτο τμήμα είναι το hardware και το δεύτερο το software. Πρέπει δηλαδή να διαθέτουμε το απαραίτητο hardware, την κατάλληλη πλακέτα Arduino και τους κατάλληλους αισθητήρες προκειμένου να τα συνδυάσουμε και να συνδέσουμε μεταξύ τους με το σωστό διάγραμμα σύνδεσης. Έπειτα, χρησιμοποιώντας το Arduino IDE προγραμματίζουμε την πλακέτα και δημιουργούμε αυτό που αποκαλούμε κώδικα. Ο προγραμματισμός του Arduino γίνεται με την γλώσσα Wiring η οποία έχει αρκετές ομοιότητες με την C++. Στη συνέχεια ανεβάζουμε τον κώδικα στην πλακέτα και το Project είναι έτοιμο να αρχίσει. Μερικές από τις πλατφόρμες Arduino που έχουν αναπτυχθεί αυτόνομα ή ως εξέλιξη κάποιας άλλης είναι οι: Arduino Uno, Arduino Diecimila, Arduino Duemilanove, Arduino Mega1280, Arduino Nano και τέλος το Arduino Mega2560 που χρησιμοποιήθηκε για την υλοποίηση της συγκεκριμένης Διπλωματικής Εργασίας. Το κόστος της αγοράς μιας πλακέτας Arduino είναι σχετικά μικρό. Σε σχέση πάντα με έναν υπολογιστή ή έτοιμα προϊόντα στην αγορά που καλύπτουν τις ίδιες ανάγκες. Αυτό ίσως είναι και ένα από τα βασικά στοιχεία αλλά και το κύριο ανταγωνιστικό του πλεονέκτημα που διαθέτει το Arduino και έχει γίνει τόσο δημοφιλής.

2.2.1 Επίσημες πλακέτες Arduino

Όπως προαναφέρθηκε το Arduino είναι ένας μικροελεγκτής ικανός να διαβάσει καταστάσεις από το περιβάλλον χρησιμοποιώντας διάφορους αισθητήρες, αλλά και να δράσει σε αυτό με τα κατάλληλα εξαρτήματα. Υπάρχουν διάφορα είδη από Arduino, τα οποία τα αποκαλούμε πλακέτες. Σε αυτά τα είδη δεν υπάρχει καλή και κακή πλακέτα. Η επιλογή για την κάθε πλακέτα είναι ανάλογα την φύση του Project για το οποίο θα χρησιμοποιηθεί. Για παράδειγμα άλλη πλακέτα θα επιλέξουμε για ένα project που θα αναπτύσσεται στην υγεία, άλλο για ρομποτική, άλλο για διασκέδαση κ.τ.λ. Οι πιο διαδεδομένες πλακέτες οι οποίες έχουν χρησιμοποιηθεί μέχρι τώρα είναι οι παρακάτω:

- Το Serial Arduino, προγραμματισμένο με μία σειριακή DE-9 σύνδεση χρησιμοποιώντας τον μικροελεγκτή ATmega8.
- Το Arduino Extreme, με ένα USB interface για προγραμματισμό χρησιμοποιώντας τον μικροελεγκτή ATmega8.
- Το Arduino Mini, μία έκδοση μινιατούρα του Arduino χρησιμοποιώντας τον μικροελεγκτή ATmega168.

- Το Arduino Nano, ένα ακόμα πιο μικρό, τροφοδοτούμενο με Usb εκδοχή του Arduino χρησιμοποιώντας τον μικροελεγκτή ATmega168.
- Το LilyPad Arduino, ένα μινιμαλιστικό σχέδιο χρησιμοποιώντας τον μικροελεγκτή ATmega328.
- Το Arduino NG, με ένα USB interface για προγραμματισμό χρησιμοποιώντας τον μικροελεγκτή ATmega8.
- Το Arduino NG plus, με ένα USB interface για προγραμματισμό χρησιμοποιώντας τον μικροελεγκτή ATmega168.
- Το Arduino Bluetooth, με Bluetooth interface για προγραμματισμό χρησιμοποιώντας τον μικροελεγκτή ATmega168.
- Το Arduino Diecimila, με ένα USB interface, χρησιμοποιεί τον μικροελεγκτή ATmega168.
- Το Arduino Duemilanove (“2009”), χρησιμοποιεί τον μικροελεγκτή ATmega168 (ATmega328 για την καινούργια έκδοση) και τροφοδοτείται μέσω USB/DC.
- Το Arduino Mega, χρησιμοποιώντας τον μικροελεγκτή ATmega1280 για περισσότερες εισόδους/εξόδους και μνήμη.
- Το Arduino Uno, χρησιμοποιώντας την ίδιο μικροελεγκτή ATmega328 όπως το τελευταίο μοντέλο Duemilanove, αλλά ενώ το Duemilanove χρησιμοποιεί ένα FTDI chipset για το USB, το Uno χρησιμοποιεί τεχνολογία ATmega8U2 προγραμματισμένο ως σειριακός μετατροπέας.
- Το Arduino Mega2560, χρησιμοποιεί τον μικροελεγκτή ATmega2560 με μνήμη στα 256KB.
- Το Arduino Leonardo, χρησιμοποιεί τον μικροελεγκτή ATmega32U4, καταργεί την ανάγκη για συνδεσιμότητα μέσω USB και μπορεί να χρησιμοποιηθεί ως ψηφιακό πληκτρολόγιο ή ποντίκι.
- Το Arduino Esplora, με εμφάνιση που παραπέμπει σε χειριστήριο κονσόλας βιντεοπαιχνιδιών με joystick και ενσωματωμένους αισθητήρες για ήχο, φώς, θερμοκρασία και επιτάχυνση.
- Το Arduino Due είναι ένα μικροχειριστήριο board βασισμένο στον μικροελεγκτή Atmel SAM3X8E ARM Cortex-M3 CPU. Είναι η πρώτη πλακέτα του Arduino βασισμένη σε επεξεργαστή 32-bit ARM microcontroller.
-

2.3 Εφαρμογές του Arduino

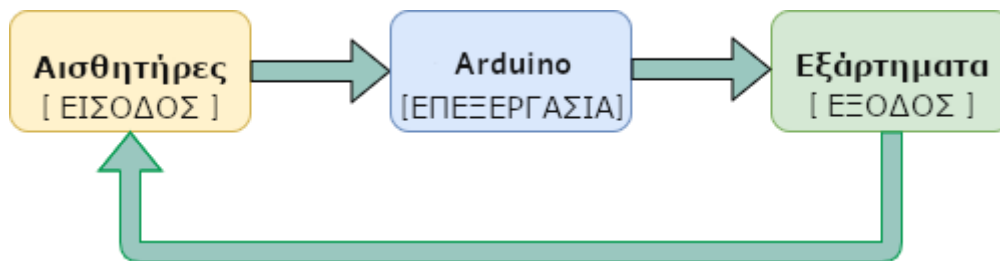
Η πρώτη Arduino πλακέτα ήταν το Uno. Ωστόσο οι ανάγκες ήταν πολλές με αποτέλεσμα σήμερα να έχουν δημιουργηθεί αρκετές παραλλαγές του Arduino Uno όπως αναφέρθηκαν και προηγουμένως όπως το Mega, Nano κτλ. Κάθε μια έχει ιδιαίτερα χαρακτηριστικά και προορίζεται για Project σε συγκεκριμένο τομέα. Επιπλέον το κάθε Project δεν αποτελείται μόνο από ένα Arduino αλλά και από αισθητήρες. Οι αισθητήρες χρησιμοποιούνται για να καταλαβαίνει το Arduino το περιβάλλον αλλά και να δρα σε αυτό. Παραδείγματα αισθητήρων είναι οι αισθητήρες θερμοκρασίας, υγρασίας, φωτεινότητας, ήχου και πολλοί άλλοι. Παραδείγματα εξόδων είναι το μοτέρ, το LED, οι οθόνες κ.τ.λ. Κάποιοι από τους τομείς στους οποίους δημιουργούνται τα διάφορα Project είναι:

- Ρομποτικές Εφαρμογές
- Internet of Things (IoT) στα πλαίσια έξυπνου σπιτιού
- Arcade παιχνίδια
- Συστήματα ασφάλειας και παρακολούθησης
- Αυτοματοποίηση και συγχρονισμός μουσικής
- Έξυπνα ρούχα με αισθητήρες
-

Στις μέρες μας πλέον γίνεται εύκολα αντιληπτό ότι όλα λειτουργούν με μια ηλεκτρονική πλακέτα και ρεύμα Θεωρητικά, σχεδόν τα πάντα μπορούν να υλοποιηθούν με ένα Arduino και τους κατάλληλους αισθητήρες. Αν όχι τα πάντα με μια αναζήτηση στο google σίγουρα μπορεί να βρει ο καθένας πολλά Project.

2.4 Λειτουργία του Arduino

Σχεδόν όλες οι εφαρμογές του Arduino ακολουθούν μια βασική λογική. Όπως έχει αναφερθεί προηγουμένως είναι γνωστό πως στο Arduino μπορούν να τοποθετηθούν διάφοροι αισθητήρες εισόδου, αισθητήρες που καταλαβαίνουν την θερμοκρασία, υγρασία, φωτεινότητα κ.τ.λ. από το περιβάλλον και επίσης μπορούν να τοποθετηθούν διάφορα άλλα εξαρτήματα όπως μοτέρ, οθόνες, LED κ.τ.λ. τα οποία αποκαλούνται και ως εξαρτήματα εξόδου καθώς έχουν ένα ενεργό ρόλο στο περιβάλλον. Με αυτό τον τρόπο, οι αισθητήρες συνδέονται με τα εξαρτήματα μέσω του Arduino, επομένως μπορούμε να θεωρήσουμε το arduino ως μια γέφυρα που παίζει το ρόλο της επεξεργασίας. Η παρακάτω εικόνα δείχνει τι είναι το Arduino σε ένα πιο λογικό επίπεδο. Θα μπορούσαμε να πούμε πως είναι ένα επιπλέον επίπεδο ανάμεσα στην λογική του “κατανοώ το περιβάλλον και πράττω”.



Εικόνα 2.1 Διασύνδεση Αισθητήρων με το Arduino

Arduino μπορεί να διαβάσει τις τάσεις και τις τιμές που λαμβάνει από τα καλώδια. Συνήθως ακολουθεί μια φάση επεξεργασίας όπου για παράδειγμα μπορεί να μετατρέπονται οι βαθμοί κελσίου σε Φαρενάιτ. Αφού γίνει και αυτό, ακολουθεί η τελική φάση της εξόδου με κάποιο εξάρτημα. αυτό μπορεί να είναι μια οθόνη που δείχνει τις τιμές που διαβάσαμε, μετά την επεξεργασία, ή μπορεί να είναι ακόμη και ένα SMS στο κινητό μας τηλέφωνο όπως στη συγκεκριμένη κατασκευή που θα αναπτυχθεί σ’ αυτή τη Διπλωματική Εργασία. Υπάρχουν και περιπτώσεις βέβαια όπου η έξοδος είναι η νέα είσοδος. Συμπερασματικά λοιπόν το Arduino είναι η γέφυρα μετατροπής των ανθρώπινων αισθήσεων σε πληροφορία κατανοητή από τον υπολογιστή αλλά και από τον ίδιο τον άνθρωπο.

2.5 Επίλογος

Στο παραπάνω κεφάλαιο έγινε αναφορά στους μικροελεγκτές και τη χρησιμότητα τους. Στη συνέχεια αναλύθηκαν οι μικροελεγκτές Arduino ως προς τον προγραμματισμό τους αλλά και την πρακτικότητα που προσφέρουν σε διαφορές κατασκευές. Επίσης αναφέρθηκαν οι επίσημες πλακέτες Arduino καθώς και ποια πλακέτα επιλέχτηκε για την κατασκευή στη συγκεκριμένη εργασία. Τέλος και με τη βοήθεια εικόνας έγινε πιο κατανοητή η λειτουργία μιας πλακέτας Arduino.

Κεφάλαιο 3ο: Arduino MEGA2560

3.1 Εισαγωγή

Το Arduino MEGA2560 είναι μια πλακέτα μικροελεγκτή που βασίζεται στο ATmega2560. Διαθέτει 54 ψηφιακούς ακροδέκτες εισόδου και εξόδου εκ των οποίων οι 15 μπορούν να χρησιμοποιηθούν και ως έξοδοι PWM. Επιπλέον διαθέτει 16 αναλογικές εισόδους, 4 σειριακές θύρες υλικού (UARTs), έναν ταλαντωτή κρυστάλλου στα 16 MHz, μια υποδοχή για σύνδεση με USB, μια υποδοχή τροφοδοσίας, μια κεφαλίδα ICSP και τέλος ένα κουμπί επαναφοράς. Περιέχει δηλαδή όλα όσα χρειάζονται για την υποστήριξη του μικροελεγκτή. Το μόνο που χρειάζεται να κάνει ο χρήστης είναι, να συνδέσει την πλακέτα του Arduino με έναν υπολογιστή μέσω καλωδίου USB προκειμένου να το προγραμματίσει και στη συνέχεια με έναν προσαρμογέα AC to DC ή με μια μπαταρία, να δώσει την απαραίτητη τροφοδοσία στην πλακέτα για να λειτουργήσει. Παρακάτω βλέπουμε έναν πίνακα με κάποια τεχνικά χαρακτηριστικά που προαναφέρθηκαν.

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
LED_BUILTIN	13
Length	101.52 mm
Width	53.3 mm
Weight	37 g

Πίνακας 2.1 Τεχνικά Χαρακτηριστικά Arduino Mega2560

Ο προγραμματισμός του μικροελεγκτή πραγματοποιείται στη γλώσσα Wiring όπως αναφέρθηκε και σε προηγούμενο κεφάλαιο, που στην ουσία πρόκειται για τη γλώσσα C++ και ένα σύνολο από βιβλιοθήκες υλοποιημένες επίσης στη C++. Το Software που χρησιμοποιήθηκε για τον προγραμματισμό του Arduino είναι το Arduino IDE. Το ATmega2560 στο MEGA2560 έρχεται προ-προγραμματισμένο με ένα bootloader που του επιτρέπει να δέχεται και να διαβάσει κάθε νέο κώδικα που δημιουργεί ο χρήστης χωρίς να χρειάζεται η χρήση κάποιου εξωτερικού υλικού προγραμματισμού. Ωστόσο δίνεται η δυνατότητα στο χρήστη να παρακάμψει το bootloader και να προγραμματίσει το μικροελεγκτή μέσω της κεφαλίδας ICPS(In- Circuit Serial Programming), χρησιμοποιώντας το Arduino ISP ή παρόμοιο. Μια σημαντική λεπτομέρεια για το Mega2560 είναι ότι διαθέτει μια επαναρυθμιζόμενη πολυσφάλεια που προστατεύει τις θύρες του υπολογιστή κατά τη διάρκεια του προγραμματισμού από υπερβολικά ρεύματα. Φυσικά οι περισσότεροι υπολογιστές παρέχουν τη δική τους εσωτερική προστασία, ωστόσο η ασφάλεια αυτή παρέχει ένα επιπλέον επίπεδο προστασίας.

3.2 Ανάλυση ακροδεκτών του Arduino Mega2560

3.2.1 Ακροδέκτες Ρεύματος

Οι ακροδέκτες ρεύματος του Arduino Mega2560 αποτελούνται από έναν ακροδέκτη τάσης εξόδου 3,3Volt, τρεις ακροδέκτες τάσης εξόδου 5Volt, πέντε γειώσεις (GND) και την τάση εισόδου (Vin) στην πλακέτα Arduino όταν είναι με μια εξωτερική πηγή ενέργειας (σε αντιδιαστολή με 5 volt από τη σύνδεση USB ή άλλη ρυθμιζόμενη πηγή ενέργειας).

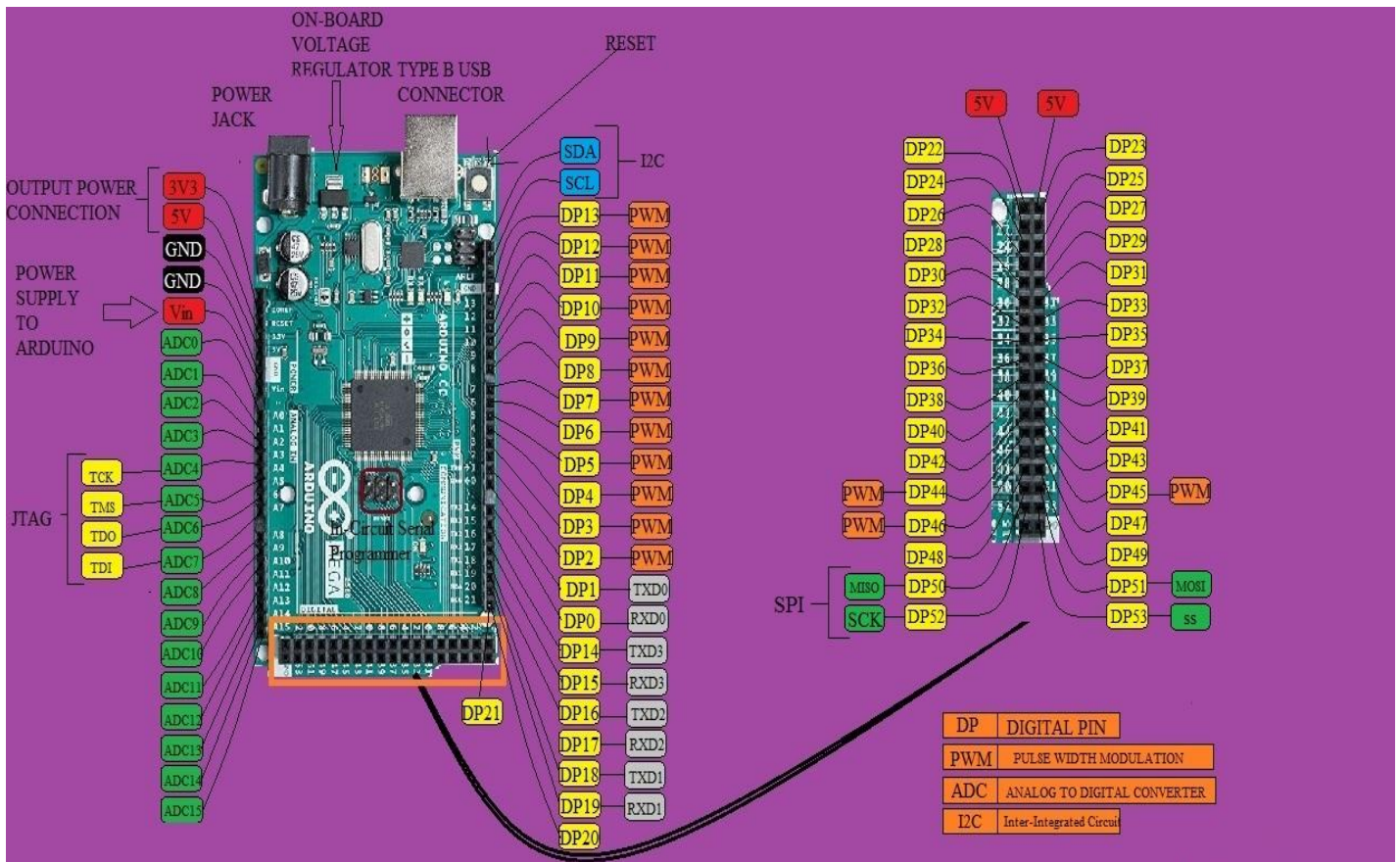
3.2.2 Αναλογικοί Ακροδέκτες

Αναλογικοί ακροδέκτες είναι οι συντελεστές όπου θα συνδεθούν το αναλογικά συστατικά, όπως τα ποτενσιόμετρα και άλλοι αισθητήρες. Αν και οι ψηφιακές εισοδοί / έξοδοι λειτουργούν μόνο με 0 και 1 τιμές, οι αναλογικές inputs λειτουργούν με τιμές 0 έως 1023. Στο Mega2560 οι αναλογικοί ακροδέκτες είναι 16 όπως αναφέρθηκε και παραπάνω.

3.2.3 Ψηφιακοί Ακροδέκτες

Ψηφιακοί ακροδέκτες μπορεί να λειτουργήσουν ως εισροές ή εκροές και να ορίσει ο προγραμματιστής πώς θα λειτουργήσουν με την pinMode λειτουργία (). Οι πείροι που έχουν το "~" μπροστά από τα νούμερα είναι PWM (Pulse Width Modulation) έξοδοι, και μπορούν να μιμηθούν αναλογικής εξόδου με λειτουργία analogWrite (). Όταν ο χρήστης χρησιμοποιεί αυτή τη λειτουργία, μπορεί να διαμορφώνει το πλάτος του παλμού, μεταβάλλοντας τη συχνότητα max και χαμηλών παλμών (0 ή 1), παριστάνοντας ένα αναλογικό παλμό. Όπως αναφέρθηκε και παραπάνω οι ψηφιακοί ακροδέκτες είναι 54 και 15 από αυτούς μπορούν να αξιοποιηθούν ως PWM έξοδοι.

3.3 Περιγραφή Arduino



Εικόνα 3.1 Ανάλυση των ακροδεκτών Arduino Mega2560

Όπως επισημάνθηκε παραπάνω το Arduino MEGA2560 έχει συνολικά 54 ακίδες input/output οι οποίες μπορούν να λάβουν ή να μεταφέρουν ένα ψηφιακό σήμα.

- Ο αριθμός PIN0 που ονομάζεται RX και ο αριθμός PIN1 που ονομάζεται TX είναι οι ακίδες λήψης και μετάδοσης, δηλαδή ο πομπός και ο δέκτης. (UART). Τα PINS 14,16,18 αλλά και τα PINS 15,17,19 είναι επίσης TX-RX αντίστοιχα. Επομένως στο Arduino Mega2560 έχουμε συνολικά 4 ζευγάρια UART (Universal Asynchronous Receiver and Transmitter)
- Υπάρχουν συνολικά 15 pins PWM στην πλακέτα του Arduino τα οποία είναι από το PIN2 έως το PIN13 και χρωματίζονται με κίτρινο χρώμα στην εικόνα συμβολίζοντας παράλληλα τους ψηφιακούς ακροδέκτες
- Αντίστοιχα με πράσινο χρώμα συμβολίζονται οι 16 αναλογικοί ακροδέκτες και είναι τα PINS από 0 έως 15. Σε αντίθεση λοιπόν με τους ψηφιακούς ακροδέκτες, οι αναλογικοί μπορούν μόνο να λαμβάνουν σήμα και όχι να παρέχουν.
- Τα PINS 22 έως 53 είναι ψηφιακοί ακροδέκτες με κάποιες ιδιαιτερότητες σε συγκεκριμένα PINS. Πιο συγκεκριμένα τα PINS 50,51,52,53 ανάλογα την κωδικοποίηση χρησιμοποιούνται είτε ως ψηφιακές εισοδοι έξοδοι είτε ως σειριακή περιφερειακή διεπαφή (SPI).

- Το PIN Vin είναι ο ακροδέκτης για τη σύνδεση τροφοδοσίας του Arduino (π.χ. μια μπαταρία τάσης 5v). Στην εικόνα φαίνεται ότι το Arduino έχει ρυθμιστή τάσης που μετατρέπει τα 5 volt σε 3v3.
- Οι άλλοι ακροδέκτες τάσης που διαθέτει η πλακέτα είναι για τάση εξόδου και ονομάζονται 5v και 3v3. Με την ύπαρξη αυτών των ακροδεκτών η ανάγκη πρόσθετων τροφοδοτικών εξαλείφεται.
- Ωστόσο η πλακέτα παρέχει τη δυνατότητα εξωτερικής τροφοδοσίας με μια ξεχωριστή υποδοχή (Power Jack)
- Στην πλακέτα υπάρχει και η USB υποδοχή (Type B USB connector)
- Το Arduino δίνει την επιλογή για reset της πλακέτας πατώντας το αντίστοιχο κουμπί πάνω στην πλακέτα.

-

3.4 Επίλογος

Το 2^ο κεφάλαιο είχε ως πρωταρχικό στόχο αρχικά την γνωριμία και στη συνέχεια την εξοικείωση με το Arduino Mega 2560. Συμπερασματικά, έγινε μια γρήγορη ανάλυση της πλακέτας όσον αφορά τις εισόδους αλλά και τις εξόδους που διαθέτει, την γλώσσα προγραμματισμού της αλλά και τον τρόπο που τροφοδοτείται. Στη συνέχεια αναλύθηκαν οι ακροδέκτες (PINS) που διαθέτει το Arduino MEGA2560 ανά κατηγορία και πιο συγκεκριμένα ως προς ακροδέκτες ρεύματος, αναλογικούς και ψηφιακούς ακροδέκτες και τέλος η ανάλυση της πλακέτας Arduino ήταν για κάθε ακροδέκτη (PIN) που διαθέτει η πλακέτα.

Κεφάλαιο 4ο: Σχεδίαση του Συστήματος

4.1 Εισαγωγή

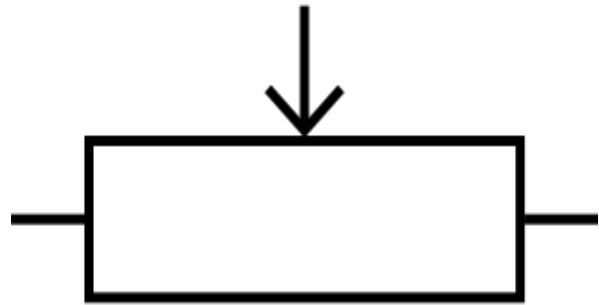
Για την υλοποίηση του συστήματος χρειάστηκαν τα εξής εξαρτήματα: 4 ποτενσιόμετρα (2x50KΩ, 1x10KΩ, 1x1KΩ), 8 φωτοдиодοι LED διαφορετικού φωτισμού μεταξύ τους, 4 διακόπτες μπουτόν (button switch), 14 αντιστάτες διαφορετικής χωρητικότητας, ένα GSM/GPRS module, μια step-down μονάδα μετατροπέα τάσης DC-DC, ένα breadboard, μια διατηρητή πλακέτα (prototype board) καθώς και ένα τροφοδοτικό. Παρακάτω αναλύονται αυτά τα εξαρτήματα.

4.2 Ποτενσιόμετρο

Το ποτενσιόμετρο είναι ένα αναλογικό ηλεκτρονικό εξάρτημα, με τρεις ακροδέκτες. Είναι ένας μεταβλητός αντιστάτης που χρησιμοποιείται συνήθως ως διαιρέτης τάσης, για τη μεταβολή δηλαδή του ηλεκτρικού δυναμικού. Ωστόσο μπορεί να λειτουργήσει και ως ροοστάτης αν χρησιμοποιηθούν μόνο οι δύο ακροδέκτες, ο μεσαίος και ένας από τους δύο άλλους ακροδέκτες του. Στο σύστημα που κατασκευάστηκε χρησιμοποιήθηκαν 4 περιστροφικά ποτενσιόμετρα.



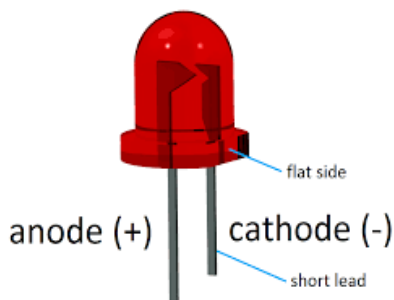
Εικόνα 4.1 Ποτενσιόμετρο



Εικόνα 4.2 Σύμβολο ποτενσιόμετρου

4.3 Φωτοδίοδος LED

Μια δίοδος εκπομπής φωτός (LED) είναι μια πηγή φωτός ημιαγωγού που εκπέμπει φως, όταν ρέει ρεύμα διαμέσου αυτής. Στο σύστημα της κατασκευής τοποθετήθηκαν 8 LED.



Εικόνα 4.3 Δίοδος LED

4.4 Διακόπτης Μπουτόν

Ένας διακόπτης μπουτόν ελέγχει μια ενέργεια σε ένα μηχάνημα ή σε κάποια άλλου είδους διαδικασία. Τα κουμπιά είναι συνήθως κατασκευασμένα από πλαστικό ή μέταλλο. Στη κατασκευή χρησιμοποιήθηκαν 4 διακοπτες.



Εικόνα 4.4 Διακόπτης μπουτόν

4.5 Αντιστάτης

Ο αντιστάτης είναι ένα ηλεκτρολογικό/ηλεκτρονικό εξάρτημα το οποίο χρησιμοποιείται σε διάφορα κυκλώματα για τον έλεγχο της ροής του ρεύματος. Στην κατασκευή χρησιμοποιήθηκαν 14 ωμικές ηλεκτρικές αντιστάσεις. Παρακάτω φαίνεται ο χρωματικός κώδικας αντιστάσεων.

Χρώμα	Σημαντικά ψηφία			Πολλαπλασιαστής	Ανοχή (%)	Θερμοκρασία	Βαθμός αποτυχίας
Μαύρο	0	0	0	X1		250 (U)	
Καφέ	1	1	1	X10	1 (F)	100 (S)	1
Κόκκινο	2	2	2	X100	2 (G)	50 (R)	0.1
Πορτοκαλί	3	3	3	X1K		15 (P)	0.01
Κίτρινο	4	4	4	X10K		25 (Q)	0.001
Πράσινο	5	5	5	X100K	0.5 (D)	20 (Z)	
Μπλε	6	6	6	X1M	0.25 (D)	10 (Z)	
Βιολετί	7	7	7	X10M	0.1 (B)	5 (M)	
Γκρι	8	8	8	X100M	0.05 (A)	1 (K)	
Λευκό	9	9	9	X1G			
Χρυσό				X0.1	5 (J)		
Ασημί			**	X0.01	10 (K)		
Κενό					20 (M)		

Εικόνα 4.5 Χρωματικός Κώδικας Αντιστάσεων

4.6 GSM/GPRS module

Η μονάδα GSM/GPRS διαθέτει υποδοχή κάρτας SIM, δίνοντας έτσι στη μονάδα δικό της τηλεφωνικό αριθμό. Ο χρήστης μπορεί να στείλει ή να λάβει ένα SMS ή λήψη φωνητικών κλήσεων μέσω της μονάδας, και μπορεί να συνδεθεί σε έναν μικροελεγκτή χρησιμοποιώντας τους κατάλληλους ακροδέκτες. Έτσι, η χρήση της μονάδας μπορεί να υπάρξει για την ανάπτυξη ενσωματωμένων εφαρμογών. Αυτό επιτυγχάνεται με ένα σύνολο εντολών AT που χρησιμοποιούνται για την επικοινωνία μεταξύ του μικροελεγκτή και της μονάδας GSM/GPRS. Στην αγορά υπάρχει μεγάλη ποικιλία από τέτοιες μονάδες. Κάθε μοντέλο όμως χρησιμοποιεί διαφορετική αρχιτεκτονική (shield). Παρακάτω θα δούμε κάποιες από αυτές.

- SIM800L: Συχνότητες λειτουργίας 850/900/1800/1900MHz
- SIM900: Συχνότητες λειτουργίας 850 / 900 / 1800 / 1900MHz
- SIM900A: Συχνότητες λειτουργίας 850 / 900 / 1800 / 1900 MHz
- SIM800: Συχνότητες λειτουργίας 850/ 900/ 1800/ 1900 MHz

Το GSM module SIM800L είναι αυτό που επιλέχτηκε για την υλοποίηση της κατασκευής.

4.6.1 SIM800L

Το GSM module SIM800L είναι ένα μικροσκοπικό μόντεμ GSM το οποίο μπορεί να ενσωματωθεί σε έναν μεγάλο αριθμό εργασιών. Μέσω αυτού του μόντεμ μπορεί κανείς να πραγματοποιήσει όσες ενέργειες θα μπορούσε να πράξει μέσω ενός τηλεφώνου. Πιο συγκεκριμένα, η αποστολή SMS, η πραγματοποίηση αλλά και η λήψη τηλεφωνικών κλήσεων, η σύνδεση στο διαδίκτυο αλλά και πολλά άλλα είναι μερικά από τις επιλογές που έχει ο χρήστης και μπορεί να κάνει μέσω αυτού του module. Για να ολοκληρωθεί η λειτουργία του, το μόντεμ υποστηρίζει δίκτυο GSM/GPRS τετραπλής ζώνης, που σημαίνει ότι λειτουργεί σχεδόν οπουδήποτε στον κόσμο. Αυτός είναι και ο λόγος της επιλογής του συγκεκριμένου μόντεμ. Στο κεντρικό σημείο της μονάδας βρίσκεται ένα κυψελοειδές τσιπ SIM800L GSM από τη SimCom. Η τάση λειτουργίας του τσιπ είναι από 3,4V έως 4,4V γεγονός που το καθιστά ιδανικό για άμεση τροφοδοσία μπαταρίας LiPo. Στην κατασκευή μας ωστόσο προτιμήθηκε η τροφοδοσία του μόντεμ να γίνεται μέσω ενός μετατροπέα τάσης ο οποίος συνδέεται απευθείας στην θύρα τροφοδοσίας του Arduino. Περισσότερη ανάλυση για τη συνδεσμολογία του συστήματος θα γίνει στο επόμενο κεφάλαιο. Το μόντεμ χρειάζεται μια εξωτερική κεραία για να συνδεθεί σε ένα δίκτυο και συνήθως συνοδεύεται από μια ελικοειδή κεραία. Στο module υπάρχει μια υποδοχή SIM στο πίσω μέρος και οποιαδήποτε ενεργοποιημένη κάρτα micro SIM 2G λειτουργεί ιδανικά. Προκειμένου να καταλάβει ο χρήστης την κατάσταση του μόντεμ, υπάρχει ένα LED στην πάνω δεξιά πλευρά το οποίο δείχνει την κατάσταση του δικτύου της κινητής τηλεφωνίας. Ανάλογα με τον ρυθμό που αναβοσβήνει το LED καταλαβαίνουμε και την κατάσταση του μόντεμ.

- Κάθε δευτερόλεπτο: Το module είναι ενεργοποιημένο αλλά δεν έχει πραγματοποιηθεί σύνδεση.
- Κάθε 2 δευτερόλεπτα: Η σύνδεση δεδομένων GPRS είναι ενεργοποιημένη
- Κάθε 3 δευτερόλεπτα: Το μόντεμ έχει κάνει τη σύνδεση με το κυψελοειδές δίκτυο και μπορεί να στείλει και να λάβει τηλεφωνικές κλήσεις και SMS

Η σωστή επιλογή της κεραίας είναι σημαντική για την ομαλή λειτουργία του μόντεμ. Όπως αναφέρθηκε προηγουμένως το μόντεμ συνοδεύεται συνήθως από μια ελικοειδή κεραία η οποία ωστόσο δυσκολεύεται να αποκτήσει σύνδεση στο δίκτυο σε εσωτερικούς χώρους. Γι' αυτό προτιμήθηκε μια κεραία τύπου 3dBi με προσαρμογέα U.FL σε SMA για καλύτερη απόδοση. Η σύνδεση της στο module γίνεται μέσω της u.fl υποδοχής που βρίσκεται στο πάνω αριστερή γωνία της συσκευής. Δεδομένου ότι το module δεν συνοδεύεται από ενσωματωμένο ρυθμιστή τάσης, απαιτείται εξωτερική τροφοδοσία προσαρμοσμένη όπως προαναφέρθηκε στα 3,4v και 4,4V (ιδανικά 4,1V). Σημαντικό είναι το τροφοδοτικό να μπορεί να τροφοδοτεί 2A ρεύματος, διαφορετικά το module θα κάνει συνεχώς reset. Τέλος, το SIM800L έχει συνολικά 12 pins. Οι συνδέσεις είναι οι εξής:

- NET: είναι το pin στο οποίο μπορεί να συγκολληθεί η ελικοειδής κεραία που συνοδεύει το μόντεμ.

- VCC: είναι το pin που παρέχει την ισχύ στο μόντεμ με τις τιμές που αναφέρθηκαν πιο πάνω για την ορθή λειτουργία του.
- RST: είναι το pin για την επαναφορά. Εάν για παράδειγμα υπάρχει πρόβλημα με τη σύνδεση για λόγους όπως ο χώρος στον οποίο βρίσκεται η συσκευή, μέσω αυτού του pin μπορούμε να κάνουμε reset του module.
- RxD: είναι ο δέκτης και χρησιμοποιείται για σειριακή επικοινωνία.
- TxD: είναι ο πομπός και χρησιμοποιείται για σειριακή επικοινωνία.
- GND: είναι η γείωση και συνδέεται με τη γείωση του Arduino.
- RING: είναι το pin διακοπής. Είναι από προεπιλογή σε κατάσταση high και μεταβάλλεται σε low για 120ms όταν λαμβάνεται μια κλήση. Μπορεί επίσης να ρυθμιστεί να μεταβάλλεται και κατά τη λήψη SMS.
- DTR: είναι το pin που ενεργοποιεί/απενεργοποιεί τη λειτουργία αναστολής λειτουργίας. Σε κατάσταση high το μόντεμ θα τεθεί σε κατάσταση αναστολής λειτουργίας, απενεργοποιώντας τη σειριακή επικοινωνία. Σε κατάσταση low το μόντεμ βγαίνει από αναστολή λειτουργίας και επικοινωνεί κανονικά.
- MIC+: είναι μια διαφορική είσοδος μικροφώνου. Οι δύο άκρες του μικροφώνου μπορούν να συνδεθούν απευθείας σε αυτά τα pin.
- SPK: είναι μια διεπαφή διαφορικών ηχείων. Και σε αυτή την περίπτωση οι 2 άκρες του ηχείου μπορούν να συνδεθούν απευθείας στα αντίστοιχα pin.



Εικόνα 4.6 GSM Module SIM800L

4.7 Step-down μονάδα μετατροπέα τάσης DC-DC

Μια μονάδα μετατροπέα τάσης DC-DC μπορεί και μειώνει την τάση από την είσοδο (παροχή) στην έξοδο (φορτίο). Είναι μια τροφοδοσία εναλλαγής της λειτουργίας του ρεύματος (switched-mode power supply) που συνήθως εμπεριέχει τουλάχιστον δύο ημιαγωγούς (μια δίοδο και ένα τρανζίστορ, να και οι σύγχρονοι μετατροπείς συχνά αντικαθιστούν τη δίοδο με ένα δεύτερο τρανζίστορ) και τουλάχιστον ένα στοιχείο για αποθήκευση ενέργειας που συνήθως είναι ένας πυκνωτής ή ένας επαγωγέας ή ο συνδυασμός των δύο. Για να μειωθεί επομένως η τάση, χρησιμοποιούνται φίλτρα πυκνωτών ή και επαγωγέων τόσο στην είσοδο όσο και στην έξοδο του μετατροπέα. Οι μετατροπείς τάσης αυτοί ονομάζονται buck. Ένας buck μετατροπέας έχει πολύ μεγαλύτερη απόδοση ισχύος ως

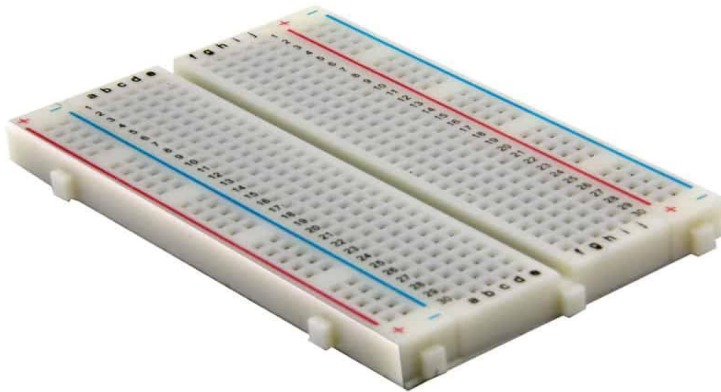
έναν μετατροπέα τάσης DC-DC σε αντίθεση με τους γραμμικούς ρυθμιστές (linear regulators), μιας και οι γραμμικοί ρυθμιστές είναι απλούστερα κυκλώματα οι οποίοι προκειμένου να μειώσουν την τάση, διασκορπίζουν την ισχύ ως θερμότητα. Οι μετατροπείς buck μπορούν να είναι εξαιρετικά αποδοτικοί καθιστώντας τους έτσι χρήσιμους για εργασίες όπως για παράδειγμα η μετατροπή της κύριας τάσης τροφοδοσίας ενός υπολογιστή σε χαμηλότερες τάσεις που απαιτούνται από USB, CPU κ.α.



Εικόνα 4.7 Step-down Μονάδα DC-DC

4.8 Breadboard

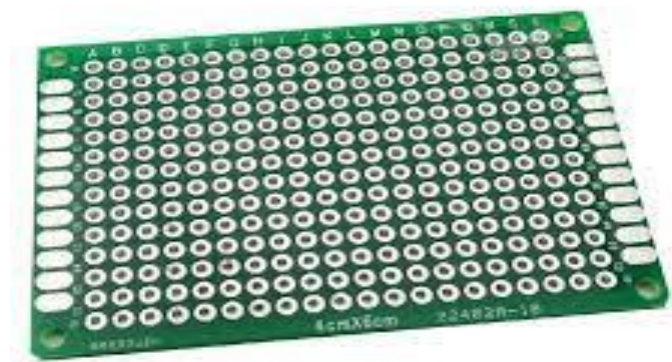
Το Breadboard επιτρέπει την διασύνδεση κυκλωμάτων, καλωδίων και άλλων εξαρτημάτων χωρίς συγκόλληση. Χρησιμοποιείται συχνά, ως εκπαιδευτικό εργαλείο και ως μέθοδος για την προτυποποίηση μιας ιδέας πριν από την συγκόλληση οποιαδήποτε εξαρτήματος. Σχεδιάστηκε από τον Ronald J. Portugal το 1971.



Εικόνα 4.8 Breadboard - Ράστερ

4.9 Διατηρητή πλακέτα (prototype board)

Η διατηρητή πλακέτα μας βοηθάει να συγκολλήσουμε οποιοδήποτε ηλεκτρονικό εξάρτημα και μηχανισμό έχουμε χρησιμοποιήσει για την υλοποίηση του κυκλώματος μας και κατ' επέκταση της κατασκευής μας.



Εικόνα 4.9 Διατηρητή Πλακέτα

4.10 Τροφοδοτικό

Το τροφοδοτικό είναι το εξάρτημα που τροφοδοτεί την πλακέτα Arduino Mega2560. Επιλέχτηκε τροφοδοτικό με είσοδο 220V AC και μετατροπή σε 12V DC σύμφωνα με τις προδιαγραφές του Arduino και συνδέεται απευθείας με την θύρα (jack) της πλακέτας.



Εικόνα 4.10 Τροφοδοτικό

4.11 Επίλογος

Στο παραπάνω κεφάλαιο αφού αναφέρθηκαν τα υλικά που χρησιμοποιήθηκαν για την δημιουργία της κατασκευής του συστήματος ελέγχου της Διπλωματικής Εργασίας, στη συνέχεια αναλύθηκαν το κάθε ένα ξεχωριστά ως προς τον τρόπο λειτουργίας τους, το λόγο χρήση τους στη συγκεκριμένη εργασία αλλά και στα χαρακτηριστικά τους. Πιο συγκεκριμένα τα υλικά ήταν τέσσερα ποτενσιόμετρα, τέσσερα μπουτόν, 8 φωτοдиодοι LED, δεκατέσσερις ωμικές αντιστάσεις, το GSM module και πιο συγκεκριμένα το SIM800I, μια step-down μονάδα μετατροπέα τάσης DC-DC, ένα breadboard (ράστερ), μια διατηρητή πλακέτα (prototype board) και ένα τροφοδοτικό.

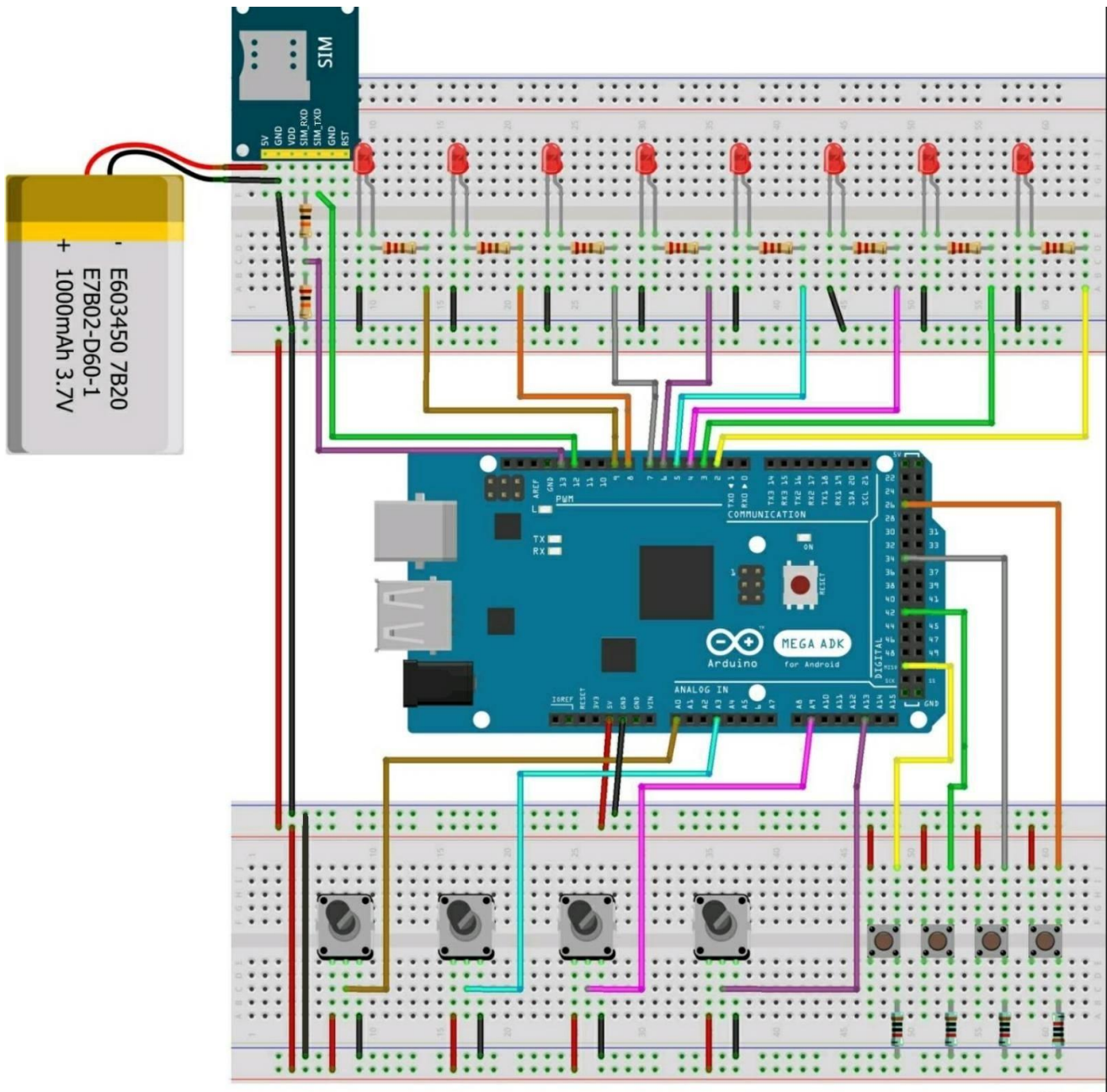
Κεφάλαιο 5ο: Κατασκευή του Συστήματος

5.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα γίνει αναφορά αλλά και ανάλυση της συνδεσμολογίας των εξαρτημάτων του συστήματος έλεγχου που κατασκευάστηκε, τον τρόπο υλοποίησης του αλλά και το συνολικό κόστος της κατασκευής.

5.2 Σύνδεση Εξαρτημάτων

Στην παρακάτω εικόνα απεικονίζεται η συνδεσμολογία που πραγματοποιήθηκε μεταξύ των εξαρτημάτων του συστήματος πάνω σε 2 ράστερ, προκειμένου να γίνουν οι απαραίτητες μελέτες και δοκιμές πριν την τελική συγκόλληση των υλικών στην πλακέτα. Η τελική συνδεσμολογία του συστήματος είναι αυτή που φαίνεται παρακάτω.



Εικόνα 5.1 Σχηματική Απεικόνιση Συνδεσμολογίας του Συστήματος

5.3 Τροφοδοσία

Πάνω αριστερά βλέπουμε την πηγή τροφοδοσίας που συστήματος ελέγχου. Στο σχηματικό διάγραμμα έχει τοποθετηθεί μια μπαταρία 3.7V ωστόσο στην πράξη υπάρχει και η επιλογή για πηγή εισόδου ενός τροφοδοτικού σε συνδυασμό έναν DC-DC converter όπως αναφέρθηκε σε προηγούμενο κεφάλαιο. Ο θετικός πόλος της τροφοδοσίας συνδέεται με τα 5V του GSM και της πλακέτας του Arduino, και ο αρνητικός πόλος με τη γείωση του GSM και τη γείωση του Arduino αντίστοιχα.

5.4 GSM Module

Αναλύοντας πρώτα το κύκλωμα που έχει υλοποιηθεί στο πάνω ράστερ όπως βλέπουμε την εικόνα, η σύνδεση του GSM Module και της πλακέτας του Arduino γίνεται μέσω των TX και RX ακροδεκτών. Αρχικά συνδέουμε τον RX ακροδέκτη του SIM800L με το pin 13 της πλακέτας Arduino και τον TX ακροδέκτη του module με το pin 12 του Arduino. Παρατηρείται ωστόσο η τοποθέτηση δύο αντιστάσεων κατά τη σύνδεση του ακροδέκτη RX του SIM800L με το PIN 13 της πλακέτας του Arduino. Μια αντίσταση 10K μεταξύ RX του SIM800L και Arduino και μια αντίσταση 20K μεταξύ RX και γείωσης. Δημιουργούμε δηλαδή έναν διαιρέτη τάσης πριν την σύνδεση του RX με το Arduino.

5.5 Ψηφιακές Έξοδοι

Στη συνέχεια συνδέουμε τις 8 ψηφιακές εξόδους του συστήματος ελέγχου όπου τις αναπαριστούν 8 LED, τα οποία είναι συνδεδεμένα το καθένα ξεχωριστά με ορθή πόλωση, που σημαίνει ότι η κάθοδος του κάθε LED είναι συνδεδεμένη στη γείωση του κυκλώματος και οι άνοδος του κάθε LED είναι συνδεδεμένη και στα PIN του Arduino. Βλέποντας στην εικόνα από τα αριστερά προς τα δεξιά τα LED είναι συνδεδεμένα με το Arduino στα PINS 9,8,7,6,5,4,3,2 αντίστοιχα. Η επιλογή των PINS είναι αυτή, διότι όπως αναφέρθηκε σε προηγούμενο κεφάλαιο, τα PINS 2-13 στην πλακέτα του Arduino αποτελούν τους ψηφιακούς ακροδέκτες. Επίσης τοποθετούνται οχτώ αντιστάσεις 220Ω σε σειρά με την άνοδο της κάθε διόδου LED, ξεχωριστά η κάθε αντίσταση για κάθε ψηφιακή έξοδο. Η ύπαρξη αυτών των αντιστάσεων είναι αναγκαία για την ομαλή και σωστή λειτουργία των LED. Χωρίς την τοποθέτηση των αντιστάσεων η πιθανότητα να καούν τα LED είναι παραπάνω από πιθανές.

5.6 Αναλογικές Είσοδοι

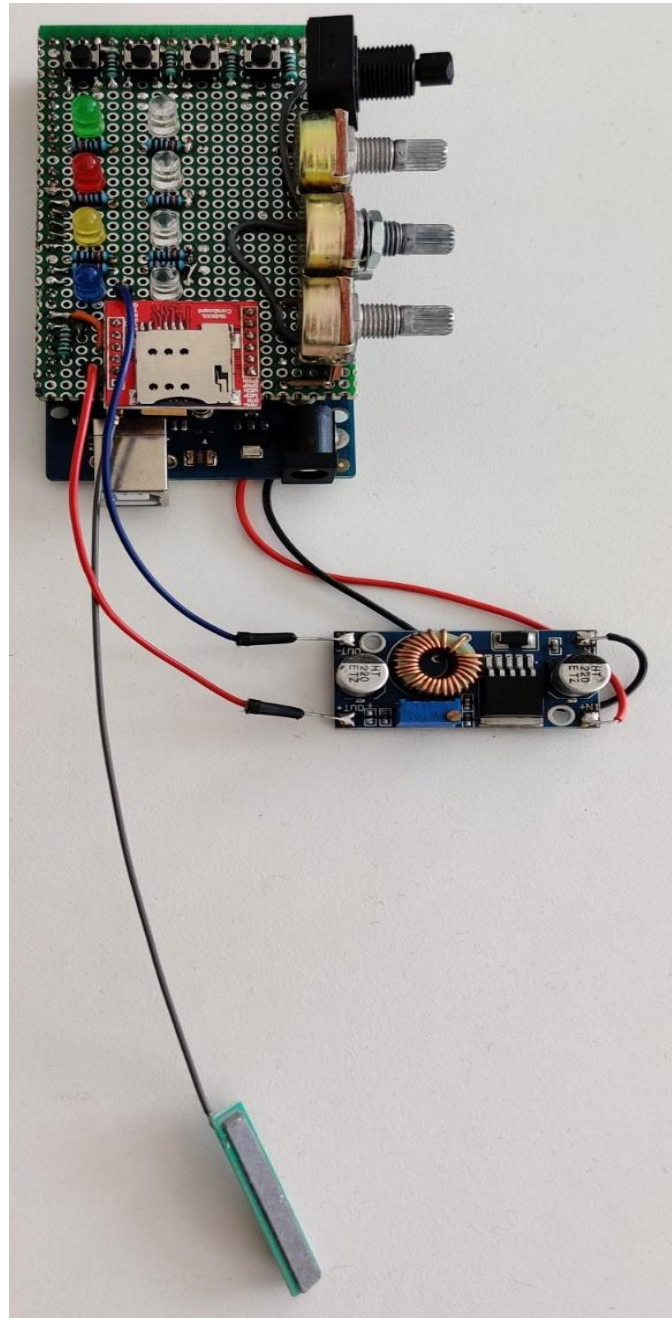
Συνεχίζοντας την ανάλυση του κυκλώματος παρατηρούμε ότι στο κάτω ράστερ της εικόνας έχουμε τις οχτώ εισόδους του κυκλώματος. Οι τέσσερις είναι αναλογικές και οι τέσσερις ψηφιακές. Από αριστερά προς τα δεξιά έχουν τοποθετηθεί οι τέσσερις αναλογικές εισοδοι (ποτενσιόμετρα) και στη συνέχεια οι τέσσερις ψηφιακοί εισοδοι (button). Όσον αφορά τα ποτενσιόμετρα, οι δύο ακριανές ακίδες έχουν συνδεθεί έτσι, ώστε να πραγματοποιείται η σωστή τροφοδοσία αλλά η και γείωση τους. Η μεσαία ακίδα συνδέει το κάθε ποτενσιόμετρο με το αντίστοιχο PIN στην πλακέτα του Arduino. Επομένως ξεκινώντας την ανάλυση από αριστερά προς τα δεξιά για τις αναλογικές εισόδους του συστήματος, η πρώτη είσοδος συνδέεται στο PIN 0 του Arduino, η δεύτερη είσοδος στο PIN 3 του Arduino, η Τρίτη είσοδος στο PIN 9 του Arduino και τέλος η τέταρτη είσοδος στο PIN 13 της πλακέτας του Arduino. Εφόσον πρόκειται για αναλογικές εισόδους, επιλεγούμε τα αναλογικά PINS του Arduino όπως έχουν αναλυθεί σε προηγούμενο κεφάλαιο.

5.7 Ψηφιακές Είσοδοι

Συνεχίζουμε με τη σύνδεση των τεσσάρων ψηφιακών εισόδων του συστήματος ελέγχου. Στη σύνδεση των button χρησιμοποιήθηκαν και πάλι 3 ακροδέκτες. Όπως αναπαρίσταται το κύκλωμα στην εικόνα, ο πάνω αριστερά ακροδέκτης δίνει την τροφοδοσία στην ψηφιακή είσοδο, ο κάτω δεξιά γειώνει και ο πάνω δεξιά συνδέει την ψηφιακή είσοδο με το Arduino. Η γείωση του κάθε button γίνεται με μια αντίσταση 120Ω προκειμένου να βεβαιωθούμε ότι η λειτουργία της κάθε εισόδου θα είναι σωστή και δεν θα υπάρχει ο κίνδυνος να καεί. Η σύνδεση των ψηφιακών εισόδων με την πλακέτα του Arduino πραγματοποιείται με τον πάνω δεξιά ακροδέκτη και βλέποντας την εικόνα από αριστερά προς τα δεξιά το πρώτο button συνδέεται με το PIN 50 του Arduino, η δεύτερη είσοδος με το PIN 42 του arduino, η τρίτη είσοδος με το PIN 34 του Arduino και τέλος η τέταρτη ψηφιακή είσοδος συνδέεται με τη πλακέτα του Arduino με το PIN 26. Η επιλογή των PINS έγινε έτσι ώστε οι ψηφιακές εισοδοι του κυκλώματος να συνδεθούν στα PIN που υποστηρίζουν αυτή την ιδιότητα.

5.8 Ράστερ και Διατρητή πλακέτα

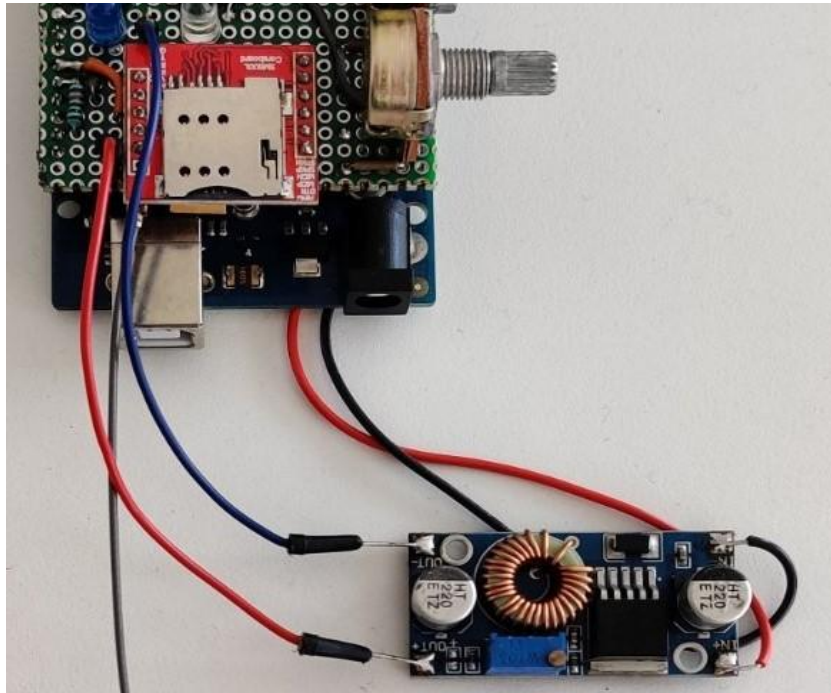
Προκειμένου το σύστημα έλεγχου που έχει αναλυθεί στα παραπάνω κεφάλαια και έχει κατασκευαστεί σε ράστερ, να γίνει πιο επαγγελματικό, πιο εύκολο στη μεταφορά και πιο πρακτικό προτιμήθηκε η ανακατασκευή του με μία διατρητή πλακέτα. Κατά τη διαδικασία αυτή το κύκλωμα του συστήματος προσαρμόστηκε έτσι ώστε το μέγεθος του αλλά και το μέγεθος της πλακέτας να είναι όσο το δυνατόν μικρότερα σε όγκο, ώστε να πληρούν τους λόγους που αναφέρθηκαν προηγουμένως για τη μεταφορά του συστήματος από το ράστερ στην πλακέτα. Η τελική μορφή του συστήματος απεικονίζεται στην παρακάτω φωτογραφία.



Εικόνα 5.2 Τελική Μορφή του Συστήματος

Όπως γίνεται εύκολα αντιληπτό στο κάτω μέρος της κατασκευής έχει τοποθετηθεί η πλακέτα του Arduino Mega και από πάνω η διατηρητή πλακέτα με όλα τα υλικά που έχουν αναφερθεί παραπάνω, συγκολλημένα και αποτεθειμένα με τον τρόπο που αναλύθηκε σε προηγούμενά κεφάλαια. Υπάρχει ωστόσο μια διάφορα στην κατασκευή σε σχέση με την εικόνα στο κεφάλαιο 3.2. Όπως αναφέρθηκε στο κεφάλαιο 3.3, η τροφοδοσία του κυκλώματος κατά την μεταφορά αυτού από το ράστερ στην πλακέτα, προτιμήθηκε να γίνεται μέσω ενός μετατροπέα DC-DC και ενός τροφοδοτικού, παρά από μία μπαταρία που ήταν η αρχική πηγή τροφοδοσίας του κυκλώματος. Ο λόγος είναι πρακτικός και αφορά την εγγύηση στη λειτουργία του συστήματος, διότι η μπαταρία μπορεί να είναι

επαναφορτιζόμενη αλλά μπορεί ανά πάσα στιγμή να τελειώσει και το σύστημα να πάψει να λειτουργεί. Επομένως η σύνδεση ενός τροφοδοτικού είναι ο τελικός τρόπος κατασκευής. Αυτό που άλλαξε επομένως στο σύστημα είναι ότι με την προσθήκη του μετατροπέα υπάρχει νέα συνδεσμολογία στην τροφοδοσία. Στην είσοδο του μετατροπέα έχει συνδεθεί η παροχή εξωτερικής τροφοδοσίας της πλακέτας του Arduino και στην έξοδο του μετατροπέα έχει συνδεθεί το GSM module όπως ακριβώς ήταν συνδεδεμένη και η μπαταρία. Το out + του μετατροπέα συνδέεται στο pin που παρέχει την ισχύ (Vcc) του GSM και το out – του μετατροπέα στο pin της γείωσης (GND) του GSM. Η σύνδεση αυτή φαίνεται και στην παρακάτω εικόνα.



Εικόνα 5.3 Σύνδεση Step-down Μονάδας DC-DC

5.9 Κόστος υλοποίησης κατασκευής

Κατά την επιλογή των υλικών για την κατασκευή προτιμήθηκε περισσότερο μια ποιοτική συνολικά κατασκευή από μια οικονομική και με χαμηλότερη ποιότητα υλικών. Στον παρακάτω πίνακα γίνεται αναλυτική κοστολόγηση των υλικών.

ΕΞΑΡΤΗΜΑΤΑ	ΚΟΣΤΟΣ
ARDUINO MEGA 2560	50 €
SIM800L	13€
DC-DC ΜΕΤΑΤΡΟΠΕΑΣ	6€
ΠΟΤΕΝΣΙΟΜΕΤΡΟ X4	2€
ΔΙΑΤΡΗΤΗ ΠΛΑΚΕΤΑ	1€

ΤΡΟΔΟΦΟΤΙΚΟ	8€
ΔΙΟΔΟΙ LED	1,50 €
ΑΝΤΙΣΤΑΣΕΙΣ	2€
ΡΑΣΤΕΡ Χ2	15€
BUTTON Χ4	1€
ΚΑΡΤΑ SIM + ΜΟΝΑΔΕΣ	15€
ΣΥΝΟΛΟ	114,50€

Πίνακας 5.1 Κόστος Εργασίας

Από τα παραπάνω γίνεται αντιληπτό ότι το συνολικό κόστος της κατασκευής είναι ένα σεβαστό πόσο. Σίγουρα μπορεί να κατασκευαστεί έτσι ώστε το συνολικό ποσό να είναι μικρότερο αλλά θα μπορούσε να είναι και μεγαλύτερο. Στα έξοδα της κατασκευής δεν υπολογίζονται τα έξοδα για πιθανή αγορά πολυμέρου, κολλητηριού, σύρμα συγκόλλησης αλλά και τα έξοδα μεταφοράς των υλικών. Στο επόμενο κεφάλαιο θα γίνει αναφορά στον τρόπο λειτουργίας του κυκλώματος.

5.10 Επίλογος

Το 4^ο κεφάλαιο αναφέρεται στην κατασκευή του συστήματος. Αρχικά γίνεται μία σχεδιαστική απεικόνιση της κατασκευής. Στη συνέχεια αναλύονται οι συνδεσμολογίες όλως τον εξαρτημάτων που χρησιμοποιήθηκαν και ποιο συγκεκριμένα γίνεται αναφορά στην συνδεσμολογία της τροφοδοσίας του συστήματος, του GSM module, των ψηφιακών εξόδων, των αναλογικών και ψηφιακών εισόδων αλλά και του ράστερ και της διατηρητής πλακέτας. Στο τέλος του κεφαλαίου αναφέρεται και το συνολικό κόστος της κατασκευής.

Κεφάλαιο 6ο: Ανάλυση και τρόπος λειτουργίας του Συστήματος

6.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα ακολουθήσει η ανάλυση της λειτουργίας του συστήματος ελέγχου, ο τρόπος με τον οποίο μπορούμε και επικοινωνούμε με το σύστημα, η διαδικασία που απαιτείται για να επιτευχτεί ορθά ο έλεγχος της επικοινωνίας του χρήστη με τη κατασκευή και τέλος η προεργασία που χρειάζεται για την ολοκλήρωση μιας επιτυχημένης χρήσης του συστήματος ελέγχου.

6.2 Προεργασία

Ξεκινώντας τη διαδικασία προετοιμασίας του συστήματος, βεβαιωνόμαστε ότι έχουμε προγραμματίσει την πλακέτα του Arduino με τον κώδικα τον οποίο θέλουμε να εκτελέσει το σύστημα μας. Επομένως συνδέουμε την πλακέτα Mega2560 με τον υπολογιστή μέσω της USB υποδοχής και με το πρόγραμμα Arduino IDE γράφουμε τον κώδικα και προγραμματίζουμε την πλακέτα. Περισσότερα

για τη διαδικασία του προγραμματισμού της πλακέτας αλλά και τη δημιουργία του κώδικα, θα αναφερθούν σε επόμενο κεφάλαιο. Στη συνέχεια αποσυνδέω την πλακέτα από τον υπολογιστή, αφαιρούμε το καλώδιο USB με το οποίο πραγματοποιήθηκε η σύνδεση της πλακέτας με τον ηλεκτρονικό υπολογιστή, τοποθετούμε την κάρτα SIM (micro) στο GSM module, αφού πρώτα βεβαιωθούμε ότι έχουμε καταργήσει τον κωδικό PIN της τηλεφωνικής κάρτας, ώστε να μπορέσει να συνδεθεί με το GSM χωρίς πρόβλημα. Έπειτα συνδέουμε την πλακέτα του Arduino με την υποδοχή που έχει αναφερθεί σε προηγούμενο κεφάλαιο για εξωτερική τροφοδοσία και τέλος συνδέουμε το τροφοδοτικό στον τοίχο. Μόλις η πλακέτα τεθεί σε λειτουργία, η φωτεινή ένδειξη LED που διαθέτει το GSM, αρχίζει να αναβοσβήνει. Με το ρυθμό που αναβοσβήνει το LED, καταλαβαίνουμε και την κατάσταση του modem, δηλαδή την κατάσταση στην οποία βρίσκεται το δίκτυο κινητής τηλεφωνίας της κάρτας SIM που τοποθετήθηκε εισάγει στο GSM. Αρχικά ο ρυθμός με τον οποίο αναβοσβήνει στο LED είναι μια φορά το δευτερόλεπτο. Αυτό μας δείχνει ότι το GSM module έχει ενεργοποιηθεί αλλά δεν έχει συνδεθεί ακόμα στο δίκτυο τηλεφωνικής επικοινωνίας. Στη συνέχεια, έπειτα από μερικά δευτερόλεπτα ο ρυθμός της φωτεινής ένδειξης LED αλλάζει και αρχίζει να αναβοσβήνει κάθε τρία δευτερόλεπτα. Αυτό δηλώνει ότι το μόντεμ έχει κάνει τη σύνδεση με το κυψελοειδές δίκτυο της κάρτας SIM που έχουμε τοποθετήσει και μπορεί να στείλει και να λάβει τηλεφωνικές κλήσεις και SMS. Σε αυτό το σημείο αξίζει να αναφερθεί το γεγονός ότι η τάση εισόδου που δέχεται το GSM είναι εξαιρετικά σημαντική για την σωστή λειτουργία του συστήματος, διότι με χαμηλότερη τάση από αυτή που έχει αναφερθεί σε προηγούμενα κεφάλαια το GSM module παρόλο που θα ενεργοποιείται δεν θα μπορεί να κάνει την σύνδεση με το δίκτυο κινητής τηλεφωνίας της κάρτας SIM και θα αναβοσβήνει η λυχνία LED κάθε δευτερόλεπτο. Αντίστοιχα μια τάση μεγαλύτερη από την προτεινόμενη πάλι δεν επιτρέπει το GSM να πραγματοποιήσει τη σύνδεση με το δίκτυο και ταυτόχρονα θέτει σε κίνδυνο το GSM για πιθανή καταστροφή και κάψιμο της πλακέτας του. Επομένως εφόσον έχει προγραμματιστεί η πλακέτα Arduino, έχει τοποθετηθεί η κάρτα SIM στο GSM, έχει συνδεθεί και τροφοδοτηθεί σωστά με την εξωτερική τροφοδοσία και από τη στιγμή που το LED του GSM αναβοσβήνει κάθε τρία δευτερόλεπτα, το σύστημα έλεγχου είναι έτοιμο για λειτουργία και επικοινωνία με το χρήστη, δηλαδή για την αποστολή αλλά και την λήψη SMS από τον χρήστη προς το GSM module αλλά και το αντίστροφο.

6.3 Αποστολή SMS

Ξεκινώντας ο χρήστης την επικοινωνία με τη συσκευή έλεγχου, βασική προϋπόθεση είναι η αποστολή SMS . Αναλόγως τη λειτουργία που θέλουμε να εκτελέσει η συσκευή μας πρέπει να στείλουμε και την αντίστοιχη εντολή – SMS σ' αυτή. Οι εντολές που λαμβάνει και ανταποκρίνεται το σύστημα μας είναι συγκεκριμένες. Αυτό πρακτικά σημαίνει ότι στον κώδικα τον οποίο μεταφορτώνουμε στη συσκευή, έχουν δοθεί συγκεκριμένες μεταβλητές και ονομασίες για την κάθε εντολή που θέλουμε να εκτελέσει το σύστημα έλεγχου. Επομένως αφού γνωρίζουμε την ονομασία της κάθε εντολής, αποστέλλουμε το αντίστοιχο SMS στη συσκευή. Εξίσου σημαντικό για μια επιτυχημένη λειτουργία είναι η σωστή δήλωση του τηλεφώνου επικοινωνίας. Στη συγκεκριμένη εργασία η συσκευή δέχεται και στέλνει SMS με ένα νούμερο τηλεφώνου. Το τηλέφωνο δηλώνεται και αυτό στον κώδικα. Περισσότερα όσον αφορά την ανάλυση του κώδικα θα αναφερθούν σε επόμενα κεφάλαια. Από την στιγμή που έχουν δηλωθεί σωστά τα στοιχεία του χρήστη, δηλαδή το νούμερο του τηλεφώνου του, και ο χρήστης γνωρίζει τις κωδικές λέξεις των εντολών που θέλει να δώσει, τότε είναι όλα έτοιμα για χρήση.

6.3.1 Επικοινωνία με Αναλογικές εισόδους

Όπως έχει αναφερθεί σε προηγούμενο κεφάλαιο, το σύστημα ελέγχου που έχει κατασκευαστεί έχει τέσσερις αναλογικές και τέσσερις ψηφιακές εισόδους. Σε αυτό το κεφάλαιο θα αναλυθεί η διαδικασία επικοινωνίας του χρήστη με το σύστημα όσον αφορά τις αναλογικές εισόδους. Λέγοντας αναλογικές εννοούμε τα τέσσερα ποτενσιόμετρα τα οποία έχουν τοποθετηθεί στην πλακέτα. Η κάθε αναλογική είσοδος ή το κάθε ποτενσιόμετρο αντιστοιχίζεται με μία ψηφιακή έξοδο που στην συγκεκριμένη εργασία είναι φωτοδιόδοι LED. Κατά την περιστροφή του ποτενσιόμετρου αλλάζει η τιμή τάσης που δέχεται το LED με αποτέλεσμα να αρχίζει να φωτίζει. Όσο μειώνεται η τιμή του ποτενσιόμετρου τόσο περισσότερο τροφοδοτείται το LED άρα τόσο περισσότερο φωτίζει. Όπως είναι φυσικό ο χρήστης δεν μπορεί να επηρεάσει την τιμή του ποτενσιόμετρου στέλνοντας κάποιο SMS στη συσκευή, αλλά μπορεί ανά πάσα στιγμή να μάθει την τιμή που έχει εκείνη την ώρα η αναλογική είσοδος στέλνοντας το SMS για την αντίστοιχη είσοδο. Η κάθε αναλογική είσοδος έχει την δική της ονομασία και για κάθε αναλογική είσοδο ο χρήστης στέλνει διαφορετικό SMS με διαφορετικό όνομα εντολής. Επομένως αν θεωρήσουμε ότι μια αναλογική είσοδος αντιστοιχεί σε ένα θερμοπομπό τότε ο χρήστης μπορεί όποτε επιθυμεί να μάθει την τιμή που έχει ο θερμοπομπός. Επομένως η επικοινωνία του χρήστη με τη συσκευή στο κομμάτι των αναλογικών εισόδων αφορά την αποστολή στοιχείων της αναλογικής εισόδου που είναι σε λειτουργία την κάθε στιγμή και όχι την ενεργοποίηση ή απενεργοποίηση της. Η διαδικασία που ακολουθούμε αλλά και οι εντολές τις οποίες στέλνουμε για να πραγματοποιήσουμε την επικοινωνία αυτή είναι εξαιρετικά απλή για τον χρήστη. Το μόνο που χρειάζεται να κάνει είναι να στείλει την σωστή εντολή. Η κάθε εντολή αποτελείται από έναν χαρακτήρα και έναν αριθμό. Ο χρήστης πρώτα πληκτρολογεί τον χαρακτήρα # όπου συμβολίζει τις αναλογικές εισόδους και στη συνέχεια τοποθετεί τον αριθμό της αναλογικής εισόδου που θέλει να επικοινωνήσει. Όπως έχει ειπωθεί, η κάθε αναλογική είσοδος οδηγείται σε μια ψηφιακή έξοδο. Επομένως, εάν ο χρήστης θέλει να μάθει πληροφορία π.χ. για την δεύτερη αναλογική είσοδο, θα πρέπει να στείλει μήνυμα με την εντολή #2. Ως απάντηση θα του έρθει η πληροφορία για την αντίστοιχη έξοδο. Σε περίπτωση που ο χρήστης στείλει λανθασμένο μήνυμα, άρα και λάθος εντολή στο σύστημα, θα δεχτεί ενημερωτικό μήνυμα για το σφάλμα στο οποίο έχει υποπέσει.

6.3.2 Επικοινωνία με Ψηφιακές εισόδους

Στο συγκεκριμένο κεφάλαιο θα γίνει ανάλυση στην διαδικασία επικοινωνίας του χρήστη με το σύστημα όσον αφορά τις τέσσερις ψηφιακές εισόδους του συστήματος ελέγχου. Λέγοντας ψηφιακές εννοούμε τα τέσσερα button που έχουμε τοποθετήσει στο σύστημα και επομένως στην πλακέτα της κατασκευής. Τα τέσσερα button έχουν ως έξοδο τις ψηφιακές διόδους LED. Για το κάθε button αντιστοιχεί και μια δίοδος LED όπως ακριβώς και για τις αναλογικές εισόδους που αναφέρθηκαν σε προηγούμενο κεφάλαιο. Η αρχική κατάσταση όλων των φωτοδίοδων LED είναι LOW δηλαδή να είναι σβηστά. Αυτό πρακτικά σημαίνει ότι με την ενεργοποίηση της συσκευής η φωτοδίοδοι LED είναι σβηστοί. Προκειμένου να τεθεί η φωτοδίοδος σε κατάσταση HIGH και επομένως να φωτίσει το LED πρέπει να πιεστεί το αντίστοιχο button. Όπως αναφέρθηκε προηγουμένως το κάθε button αντιστοιχεί και σε μία έξοδο LED. Σε αυτό το σημείο αξίζει να αναφερθεί ότι κατά τον προγραμματισμό της συσκευής τα button έχουν προγραμματιστεί με τέτοιο τρόπο ώστε η κατάσταση τους να αλλάζει από LOW σε HIGH κατά το πάτημα. Δηλαδή με ένα πάτημα η κατάσταση γίνεται HIGH και ενεργοποιείται η έξοδος και με δεύτερο πάτημα η κατάσταση γίνεται LOW και το LED σβήνει. Αυτό αναφέρεται διότι η κατάσταση των button θα μπορούσε να μεταβάλλεται με τρόπο τέτοιο κατά τον οποίο με την πίεση του button αυτό να τίθεται σε κατάσταση HIGH και να ενεργοποιείται το LED και

με την απελευθέρωση να τίθεται αυτόματα σε κατάσταση LOW και το LED να σβήνει. Επομένως ο χρήστης ελέγχει την κατάσταση της ψηφιακής εισόδου με το κάθε πάτημα που πραγματοποιεί σε κάθε button. Με το που πατηθεί μια φορά το button, τότε το σύστημα στέλνει αυτόματα μήνυμα στο χρήστη και τον ενημερώνει για την κατάσταση της ψηφιακής εισόδου που έχει ενεργοποιήσει ή απενεργοποιήσει αντίστοιχα. Για παράδειγμα έστω ότι ο χρήστης πιέζει το button 2 και ενεργοποιείται το LED. Τότε το σύστημα ελέγχου θα στείλει μήνυμα στο κινητό του χρήστη το οποίο έχουμε δηλώσει στον κώδικα, ότι έχει πραγματοποιηθεί μια αλλαγή στην κατάσταση της κατασκευής και έχει ενεργοποιηθεί η έξοδος (LED) που του αντιστοιχεί button. Το μήνυμα που θα σταλεί είναι και αυτό μέρος του κώδικα και επομένως ορίζουμε εμείς τι ακριβώς θέλουμε να λέει. Αντίστοιχα η ίδια διαδικασία θα πραγματοποιηθεί και αν η κατάσταση του LED ήταν HIGH και με το πάτημα του button από τον χρήστη το θέταμε σε κατάσταση LOW. Τότε πάλι το σύστημα θα έστειλε το μήνυμα που θα είχαμε ορίσει προκειμένου να ενημερώσει τον χρήστη για την κατάσταση στην οποία βρίσκεται η αντίστοιχη έξοδος. Εκτός από την ενεργοποίηση ή την απενεργοποίηση των LED με το πάτημα των button, η διαδικασία αυτή μπορεί να πραγματοποιηθεί και μέσω SMS. Αναλυτικότερα, ο χρήστης μπορεί επηρεάσει την κατάσταση του button και να θέσει σε κατάσταση HIGH το LED στέλνοντας με SMS, στη συσκευή ελέγχου, την εντολή που έχουμε ορίσει κατά τον προγραμματισμό του Arduino. Αντίστοιχα, ακολουθώντας την ίδια διαδικασία και στέλνοντας την προγραμματισμένη εντολή στο σύστημα, ο χρήστης μπορεί να επηρεάσει την κατάσταση του button και αν υποθέσουμε ότι το LED ήταν ανοιχτό, δηλαδή σε κατάσταση HIGH, να το θέσει σε κατάσταση LOW και να το σβήσει. Σε αυτή την περίπτωση το σύστημα ελέγχου, στέλνει και πάλι μήνυμα στον χρήστη και τον ενημερώνει για την κατάσταση που βρίσκεται η έκαστοτε έξοδος (LED). Όσον αφορά τις εντολές που πρέπει να στείλει ο χρήστης στο GSM module για να ελέγξει τις ψηφιακές εισόδους στην συσκευή ελέγχου, η ανάλυση τους θα ακολουθήσει παρακάτω. Έστω λοιπόν ότι ο χρήστης θέλει να ενεργοποιήσει την δεύτερη έξοδο LED του συστήματος. Τότε όπως και στις αναλογικές εισόδους, θα πρέπει να στείλει μήνυμα στο σύστημα με την εντολή που έχουμε ορίσει. Όπως και στις αναλογικές εισόδους, η εντολή αποτελείται από έναν χαρακτήρα και έναν αριθμό. Ο χρήστης πρώτα θα πρέπει να πληκτρολογήσει τον χαρακτήρα που έχει προγραμματιστεί. Στην περίπτωση των ψηφιακών το σύμβολο που πληκτρολογεί είναι το \$ όπου συμβολίζει τις ψηφιακές εισόδους. Στη συνέχεια τοποθετεί τον αριθμό της ψηφιακής εισόδου που θέλει να επικοινωνήσει. Επομένως για το παράδειγμα μας, εφόσον ο χρήστης θέλει να ενεργοποιήσει την δεύτερη έξοδο θα πρέπει να πληκτρολογήσει και να στείλει στο GSM module την εντολή \$2. Με τον ίδιο τρόπο μπορεί να απενεργοποιήσει την έξοδο. Στέλνοντας δηλαδή και πάλι \$2.

6.4 Επίλογος

Στο κεφάλαιο που ακολούθησε έγινε αναφορά στην διαδικασία επικοινωνίας του χρήστη με το σύστημα ελέγχου. Πιο συγκεκριμένα αναλύθηκε ο τρόπος με τον οποίο ο χρήστης πρέπει να έχει προετοιμάσει την συσκευή ελέγχου, καθώς και ποιές ενέργειες απαιτούνται ώστε το GSM Module αλλά και η πλακέτα Arduino να είναι έτοιμα για χρήση. Στη συνέχεια έγινε επεξήγηση στην διαδικασία επικοινωνίας του χρήστη με τη συσκευή ελέγχου και πως αυτή πραγματοποιείται με επιτυχία. Τέλος έγινε ανάλυση στον έλεγχο που μπορεί να έχει ο χρήστης στη συσκευή και πως μπορεί να επικοινωνεί με τις αναλογικές αλλά και τις ψηφιακές εισόδους που έχουμε τοποθετήσει στην κατασκευή.

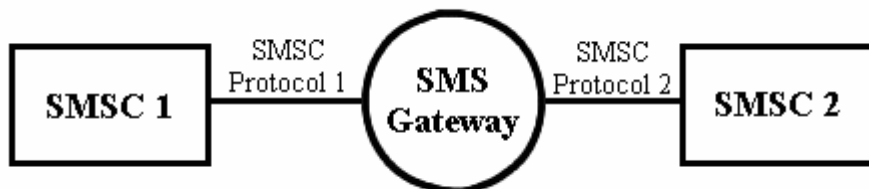
Κεφάλαιο 7ο: SMS (Short Message Service)

7.1 Εισαγωγή

Στο παρόν κεφάλαιο θα αναλύσουμε πώς λειτουργεί το sms για ασύρματη επικοινωνία και πώς μπορούμε να χρησιμοποιήσουμε τον υπολογιστή για να στείλουμε και να λάβουμε μηνύματα. Θα δούμε επίσης με πιο τρόπο μπορούμε να δώσουμε εντολές στο GSM για να εκτελέσουμε κάποιες λειτουργίες που θα μας είναι χρήσιμες για την εφαρμογή μας.

7.2 Το SMS και η διαδικασία της ασύρματης επικοινωνίας

Το sms τα τελευταία χρόνια έχει χρησιμοποιηθεί σε πολλές εφαρμογές ασύρματων επικοινωνιών. Ιστορικά, δημιουργήθηκε στην Ευρώπη το 1992 και η κύρια χρήση του ήταν για την επικοινωνία μεταξύ κινητών τηλεφώνων. Όπως εκδηλώνει και η ονομασία του, ένα μήνυμα sms μπορεί να έχει μέγιστη χωρητικότητα μήνυμς 1120 bits. Ενώ υποστηρίζει όλες τις γλώσσες σε παγκόσμια κλίμακα. Εκτός από την χρησιμότητα του sms για την επικοινωνία ατόμων, μπορούμε να μεταχειριστούμε αυτή τη τεχνολογία ώστε να λαμβάνουμε ειδοποιήσεις στην κινητή μας συσκευή ή ακόμη και να την ελέγχουμε. Όταν γίνεται αποστολή ενός μηνύματος sms, το περιεχόμενο του αρχικά θα αποθηκευτεί προσωρινά σε ένα κέντρο, το οποίο συνήθως καθορίζεται από το δίκτυο τηλεπικοινωνίας που χρησιμοποιούμε, ανάλογα δηλαδή με την κάρτα SIM που έχουμε. Στην συνέχεια και εφόσον δεν υπάρχει κάποιο σφάλμα στο μήνυμα, αυτό θα σταλεί στον προορισμό του από τον οποίο θα λάβει στην συνέχεια ο αποστολέας μια ένδειξη ότι το μήνυμα έχει φτάσει, ειδικά θως θα του εμφανίσει ειδοποίηση ότι η αποστολή απέτυχε. Εάν ο αποστολέας και ο δέκτης δεν ανήκουν στο ίδιο δίκτυο τότε θα χρειαστεί να μεσολαβήσουν περισσότερα από ένα κέντρα ώστε να σταλθεί το μήνυμα. Σε πολλές περιπτώσεις τα διαφορετικά δίκτυα δεν χρησιμοποιούν τα ίδια πρωτόκολλα επικοινωνίας, για να αντιμετωπίσουμε αυτό το πρόβλημα τοποθετούμε ανάμεσα στα κέντρα sms μια πύλη δικτύου (sms gateway) η οποία θα κάνει την μετάφραση από το ένα πρωτόκολλο στο άλλο για να μην υπάρξουν σφάλματα στη μετάδοση μεταξύ των κέντρων.



Εικόνα 7.1 Διαδικασία Ασύρματης Επικοινωνίας

7.3 Αποστολή sms μέσω Ηλεκτρονικού Υπολογιστή με GSM μόντεμ

Στη περίπτωση που θέλουμε να στείλουμε ένα sms από έναν υπολογιστή, θα χρειαστούμε ένα GSM μόντεμ. Ο διαμορφωτής GSM είναι ένα ασύρματο μόντεμ ο οποίος είναι συμβατός με τα ασύρματα GSM δίκτυα. Για να γίνει η αποστολή του μηνύματος θα πρέπει αρχικά να συνδέσουμε στο μόντεμ μια κάρτα SIM και να το συνδέσουμε με τον υπολογιστή με ένα USB καλώδιο. Στην συνέχεια για να ελέγξουμε το μόντεμ ως προς την ορθή λειτουργία του θα πρέπει να του δώσουμε κάποιες εντολές. Οι εντολές αυτές ονομάζονται εντολές AT. Ένα παράδειγμα εντολών AT βλέπουμε στον πίνακα παρακάτω.

AT Εντολή	Περιγραφή
"+CMGS"	αποστολή μηνύματος.
"+CMSS"	αποστολή μηνύματος από το χώρο αποθήκευσης.
"+CMGW"	αποθήκευση μηνύματος στη μνήμη.
"+CMGD"	διαγραφή μηνύματος.
"+CMGC"	αποστολή εντολής.
"+CMMS"	περισσότερα μηνύματα προς αποστολή.

Πίνακας 7.1 Εντολές AT κατά την Αποστολή

Για να στείλουμε τις εντολές AT στο μόντεμ θα κάνουμε χρήση του command window του υπολογιστή, ειδικά όμως μπορούμε να το γράψουμε τον πηγαίο κώδικα της AT στην εφαρμογής μας, σε οποιαδήποτε γλώσσα προγραμματισμού.

7.4 Λήψη sms μέσω Ηλεκτρονικού Υπολογιστή με GSM μόντεμ

Η λήψη sms με χρήση GSM διαμορφωτή, είναι παρόμοια με την αποστολή. Αυτό που χρειαζόμαστε είναι να στείλουμε τις οδηγίες για τη λήψη και την ανάγνωση, δίνοντας κάποιες εντολές AT. Οι βασικές εντολές φαίνονται στον πίνακα παρακάτω.

Εντολή AT	Περιγραφή
"+CNMI"	νέα ένδειξη μηνύματος.
"+CMGL"	τοποθέτηση μηνυμάτων σε λίστα.
"+CMGR"	ανάγνωση μηνυμάτων.
"+CNMA"	αναγνώριση νέων μηνυμάτων.

Πίνακας 7.2 Εντολές AT κατά τη Λήψη

7.5 Εντολές AT

Όπως αναφέρθηκε και στην προηγούμενη ενότητα, οι εντολές AT μας δίνουν την δυνατότητα ελέγχου των μόντεμ. Η λέξη AT προέρχεται από την συντομογραφία της λέξης Attention και χρησιμοποιείται στην αρχή κάθε γραμμής του κώδικα για να ενημερώσει το μόντεμ ότι πρόκειται να ακολουθήσει μια εντολή AT, πχ. AT+CMGS. Υπάρχουν δυο είδη εντολών AT, οι βασικές εντολές και οι εκτενής εντολές. Μπορούμε να ξεχωρίσουμε τις εκτενής εντολές από τις βασικές από το σύμβολο "+" που παρέρχεται της εντολής. Οι κανόνες σύνταξης μια εκτενούς εντολής AT είναι αρχικά ότι πρέπει κάθε γραμμή εντολής να ξεκινά με το "AT" και να τελειώνει με έναν χαρακτήρα επιστροφής, όπως πχ. το enter στο command window. Επίσης μπορούμε να έχουμε παραπάνω από μια εντολή σε μία γραμμή εντολών, γράφοντας "AT" μόνο στην αρχή της γραμμής και διαχωρίζοντας τις εντολές με το σύμβολο ";" πχ. AT+CMGL;+CGMI. Τα string πρέπει να περικλείονται από αποσιωπητικά πχ. AT+CMGL="ALL". Τέλος οι εντολές θα πρέπει να γράφονται με κεφαλαία γράμματα, ωστόσο σε κάποια GSM δεν επηρεάζει τον κώδικα. Αφού εκτελεστεί μια εντολή AT ο χρήστης θα πρέπει στο τέλος μια ειδοποίηση για το αν η εντολή εκτελέστηκε επιτυχώς ή αν υπήρξε κάποιο σφάλμα. Τα μηνύματα αυτά ονομάζονται κωδικοί τελικού αποτελέσματος και οι πιο συνηθισμένοι είναι το "OK" και το "ERROR". Μπορούμε να κατανοήσουμε ότι μετά την εκτέλεση ο κωδικός "OK" μας δείχνει ότι η εντολή εκτελέστηκε επιτυχώς από το μόντεμ ή την κινητή συσκευή. Ενώ ο κωδικός "ERROR" θα εμφανιστεί εάν κατά την διάρκεια της εκτέλεσης βρεθεί κάποιο σφάλμα στον κώδικα μας, τότε οι υπόλοιπες εντολές που ακολουθούν δε θα εκτελεστούν και θα μας εμφανιστεί ο κωδικός "ERROR". Τα λάθη μπορεί να είναι συντακτικά ή να έχουμε δώσει άκυρη τιμή σε μια μεταβλητή ή ακόμη η εντολή να είναι σωστή αλλά να μην υποστηρίζεται από το GSM μόντεμ που χρησιμοποιούμε.

7.6 Τρόποι λειτουργίας των GSM μόντεμ/ κινητών συσκευών

Υπάρχουν δυο τρόποι λειτουργίας που καθορίζουν τα sms. Η ονομασίες του είναι SMS text mode και SMS PDU mode (Protocol Data Unit – Μονάδα Δεδομένων Πρωτοκόλλου). Ο τρόπος με

τον οποίο λειτουργεί το μόντεμ ή η κινητή συσκευή καθορίζει τη σύνταξη κάποιων εντολών AT και τη μορφή των ειδοποιήσεων μετά την εκτέλεση των εντολών αυτών. Οι εντολές που επηρεάζονται από τον τρόπο λειτουργίας είναι οι εξής:

- +CMGS (Αποστολή μηνύματος)
- +CMSS (Αποστολή μηνύματος από το χώρο αποθήκευσης)
- +CMGR (Ανάγνωση μηνύματος)
- +CMGL (Τοποθέτηση μηνυμάτων σε λίστα)
- +CMGW (Αποθήκευση μηνύματος στη μνήμη)
- +CNMA (Αναγνώριση νέου μηνύματος)
- +CMGC (Αποστολή εντολής)

Επίσης οι παρακάτω δυο εντολές είναι χρήσιμες μόνο όταν κάνουμε χρήση του text mode :

- +CSMP (Καθορισμός παραμέτρων του Text Mode)
- +CSDH (Εμφάνιση παραμέτρων του Text Mode)

Η κύρια διαφορά των δυο μεθόδων λειτουργίας είναι ότι στο text mode τα μηνύματα εισέρχονται και εξέρχονται σε μορφή αναγνώσιμου κειμένου, ενώ με την PDU μορφή γίνεται κωδικοποίηση σε δεκαεξαδικό σύστημα. Επίσης η ορίζουσα τιμή κάποιων παραμέτρων του συστήματος είναι ορισμένοι διαφορετικοί για το κάθε σύστημα. Στο text mode οι τιμές αναγράφονται σαν strings, ενώ σε PDU mode χρησιμοποιούνται αριθμοί. Η αντιστοιχία απεικονίζεται στον πίνακα που ακολουθεί.

Κατάσταση μηνύματος	Ορίζουσα τιμή σε text mode	Ορίζουσα τιμή PDU mode
Σταλμένο και μη αναγνωσμένο	"REC UNREAD"	0
Σταλμένο και αναγνωσμένο	"REC READ"	1
Καταχωρημένο και μη σταλμένο	"STO UNSEND"	2
Καταχωρημένο και σταλμένο	"STO SEND"	3
Όλα τα μηνύματα	"ALL"	4

Πίνακας 7.3 Αντιστοιχία Text και PDU mode

7.7 Επίλογος

Κλείνοντας το κεφάλαιο αυτό, είδαμε μια εισαγωγή για το τι είναι το sms, πώς ξεκίνησε και πώς λειτουργεί στην πράξη. Στην συνέχεια καταδείξαμε τον τρόπο με τον οποίο μπορούμε να αποστείλουμε και να λάβουμε μηνύματα μεταξύ ενός υπολογιστή και ένα GSM, χρησιμοποιώντας τις εντολές AT, οι οποίες μας βοηθούν στο να ελέγχουμε και να χειριζόμαστε τα μόντεμ. Στο τέλος του κεφαλαίου αναπτύξαμε τους δυο διαφορετικούς τρόπους λειτουργίας των GSM μόντεμ, το text mode και το PDU mode, και πώς διαφέρουν οι AT εντολές ανάλογα με την μέθοδο που χρησιμοποιούμε.

Κεφάλαιο 8ο: Προγραμματιστικός Κώδικας Arduino

8.1 Εισαγωγή

Το παρόν κεφάλαιο περιέχει τον προγραμματιστικό κώδικα της εφαρμογής μας, καθώς και την ανάλυση του. Θα μελετήσουμε πώς μπορούμε να προγραμματίσουμε το Arduino ώστε να πραγματοποιείται η ορθή επικοινωνία με το GSM και μαζί σαν σύστημα να αποστέλλουν και να λαμβάνουν μηνύματα από το χρήστη. Τέλος θα περιγράψουμε την γενική δομή του προγραμματιστικού κώδικα της εφαρμογής, με τη βοήθεια διαγραμμάτων ροής.

8.2 Προεργασία προγραμματισμού

Η εφαρμογές με τον μικροελεγκτή Arduino απαιτούν, εκτός από το κύκλωμα και τα εξαρτήματα, και υλοποίηση προγραμματιστικού κώδικα. Τα προγράμματα για Arduino γράφονται σε περιβάλλον IDE (Integrated Development Environment), το οποίο είναι ένα πρόγραμμα που μας επιτρέπει να γράφουμε κώδικα για τα προγράμματα σε διάφορες πλακέτες Arduino. Τα προγράμματα στη γλώσσα του Arduino ονομάζονται sketches, τα οποία ελέγχονται και μεταφράζονται από το IDE σε κώδικα που καταλαβαίνει η πλακέτα.

Το πρώτο μας βήμα είναι να συνδέσουμε τον υπολογιστή με τη πλακέτα με ένα USB καλώδιο, πετυχαίνουμε έτσι την σειριακή επικοινωνία μεταξύ τους. Το επόμενο βήμα είναι να κατεβάσουμε το Arduino IDE στον υπολογιστή για να μπορούμε να γράψουμε εκεί τον κώδικα και να το μεταφέρουμε στη πλακέτα για εκτέλεση. Στη συνέχεια αφού έχουμε το γραφικό μας περιβάλλον έτοιμο ξεκινάμε με την γραφεί του sketch. Τα sketch του Arduino γράφονται σε γλώσσα προγραμματισμού C++, μαζί με τη προσθήκη κάποιων ειδικών μεθόδων και συναρτήσεων.

Κατά τη διάρκεια του προγραμματισμού, είναι θεμιτό να εκτελούμε ελέγχους για να είμαστε βέβαιοι ότι δεν υπάρχουν σφάλματα. Μπορούμε εύκολα να κάνουμε τον έλεγχο πατώντας το κουμπί compile στο περιβάλλον Arduino IDE, με αυτό το τρόπο θα γίνει έλεγχος από το πρόγραμμα για συντακτικά λάθη στο κώδικα και θα δούμε ειδοποιηθούμε με μία ένδειξη γραμμένη με κόκκινα γράμματα στην οποία θα αναγράφεται το λάθος μαζί με την γραμμή στην οποία βρίσκεται. Εφόσον δεν υπάρχουν συντακτικά λάθη μένει να ελέγξουμε αν το πρόγραμμα μας λειτουργεί σωστά, κάνοντας δοκιμές στη πλακέτα.

Η δομή ενός sketch ξεκινάει με την εισαγωγή των βιβλιοθηκών που θα μας δώσουν πρόσβαση στις μεθόδους και τις συναρτήσεις για την επικοινωνία με την πλακέτα και τον χειρισμό της. Μετά ορίζουμε τις global μεταβλητές που θα είναι ορατές σε όλο το πρόγραμμα. Ακολουθεί η συνάρτηση setup() η οποία εκτελείται μία φορά και σκοπός της είναι η προετοιμασία του προγράμματος για

εκτέλεση. Τέλος, γράφουμε την συνάρτηση loop() που αντιστοιχεί στο κύριο πρόγραμμα και εκτελείται συνεχώς μέχρι ο χρήστης να ορίσει τον τρόπο τερματισμού της.

Για την δική μας εφαρμογή χρειαζόμαστε ένα πρόγραμμα στο οποίο αρχικά θα ορίζουμε τις αναλογικές και ψηφιακές εισόδους και εξόδους σε συγκεκριμένα pin του Arduino. Θα πρέπει να ξεχωρίζουμε τις θέσεις στις οποίες θα έχουμε τα κουμπιά από αυτές που θα έχουμε τα ποτενσιόμετρα και τις θέσεις εξόδου που θα είναι τα led. Επίσης πρέπει να δώσουμε τις κατάλληλες εντολές για την σειριακή επικοινωνία του Arduino με το GSM, ώστε να μπορούμε να χειριζόμαστε το GSM για την αποστολή και λήψη μηνυμάτων. Θέλουμε ο GSM να λειτουργεί σαν αποστολέας όταν αλλάζει κατάσταση κάποιο led ή όταν του στείλουμε μήνυμα για τον έλεγχο μιας αναλογικής εξόδου, ενώ τις υπόλοιπες στιγμές θα πρέπει να λειτουργεί σαν δέκτης και να αναμένει μήνυμα από το χρήστη για να εκτελέσει μια ενέργεια.

8.3 Διαγράμματα ροής προγράμματος

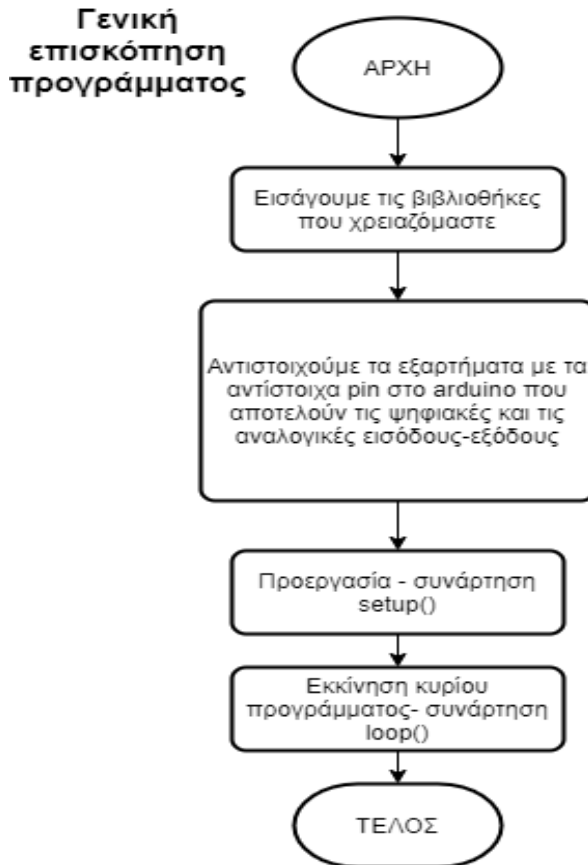
Ένας απλός και εύκολος τρόπος να περιγράψουμε ένα πρόγραμμα είναι να το χωρίσουμε με βάση τις διαφορετικές συναρτήσεις που το αποτελούν και να δημιουργήσουμε ένα διάγραμμα ροής για το κάθε κομμάτι. Τα διαγράμματα ροής είναι σχηματικές απεικονίσεις του κώδικα, με κάποια βασικά στοιχεία. Κάθε διάγραμμα γράφεται από πάνω προς τα κάτω ξεκινώντας από την αρχή και καταλήγει πάντα στο τέλος του προγράμματος, με τα ενδιάμεσα σημεία να απεικονίζουν τις ενέργειες που εκτελούνται σε κάθε σημείο του κώδικα συνοπτικά.

Σημείωση : Για να μην υπάρχει σύγχυση στο διαχωρισμό των εξόδων των αναλογικών εισόδων με τις εξόδους των ψηφιακών, θα αναγράφουμε στην περιγραφή του κώδικα τις εξόδους των αναλογικών εισόδων ως αναλογικές εξόδους, παρότι στη πράξη μετατρέπουμε το αναλογικό σήμα εισόδου σε ψηφιακό στην έξοδο του.

Σημείωση - 2: Σε πολλά μέρη του προγράμματος γίνεται έλεγχος για την αντιστοιχία των εξόδων ανάλογα με το μήνυμα που έχει σταλθεί από το χρήστη ή στη περίπτωση των ψηφιακών εξόδων εκτός από το μήνυμα γίνεται έλεγχος και για το αντίστοιχο κουμπί που μπορεί να πατήθηκε. Για να διευκολύνουμε τη σχεδίαση και την ανάγνωση των διαγραμμάτων θα ονομάζουμε τις μεταβλητές αυτές x, όπου το x θα αντιστοιχεί σε κάθε διαφορετική έξοδο με όρια από 1 μέχρι 4 (όσες και οι ψηφιακές και οι αναλογικές έξοδοι) και θα παρουσιάζεται σαν ένας μόνο έλεγχος στο διάγραμμα ροής.

8.3.1 Η βασική δομή

Ξεκινώντας το πρόγραμμα για το Arduino θα πρέπει να εισάγουμε τις κατάλληλες βιβλιοθήκες για να έχουμε πρόσβαση στις συναρτήσεις της σειριακής επικοινωνίας με το GSM και το Arduino. Στη συνέχεια για να μπορέσουμε να κάνουμε την αντιστοιχία των ψηφιακών και των αναλογικών εισόδων και εξόδων με τα εξαρτήματα, και να τα αναπαραστήσουμε στο κώδικα, θα δημιουργήσουμε μεταβλητές και θα τις θέσουμε με τα αντίστοιχα pin του Arduino. Με αυτό τον τρόπο θα μπορέσουμε να ελέγχουμε προγραμματιστικά την εφαρμογή μας στα συγκεκριμένα σημεία του συστήματος. Μετά τις προαναφερθέντες αρχικοποιήσεις, σε κάθε πρόγραμμα Arduino υπάρχει η συνάρτηση setup() στην οποία γίνεται η προεργασία για να ακολουθήσει η συνάρτηση loop() που αποτελεί το κύριο πρόγραμμα. Η βασική ροή του προγράμματος φαίνεται και στο διάγραμμα ροής που ακολουθεί.



Εικόνα 8.1 Διάγραμμα Ροής Βασικής Δομής του Προγράμματος

8.3.2 Η συνάρτηση Setup()

Η συνάρτηση `setup()` είναι υπεύθυνη για την προεργασία που απαιτείται πριν το κύριο πρόγραμμα. Στην εφαρμογή μας, χρησιμοποιούμε την `setup()` για να αρχικοποιήσουμε τα pin εισόδου και εξόδου και να θέσουμε τα led εξόδου στη τιμή LOW, που σημαίνει ότι θα ξεκινάμε με τα led σβηστά στην αρχή της εφαρμογής. Σε αυτό το σημείο επίσης απαιτείται να ξεκινήσουμε την σειριακή επικοινωνία μεταξύ του GSM και του Arduino, ώστε να μπορούμε να στέλνουμε και να λαμβάνουμε τα SMS. Σβήνοντας όσα SMS μπορεί να υπάρχουν στη μνήμη από προηγούμενες χρήσεις και αφού πετύχει η σειριακή επικοινωνία, θέτουμε το GSM σε λειτουργία δέκτη για να αναμένει κάποιο μήνυμα από το χρήστη.

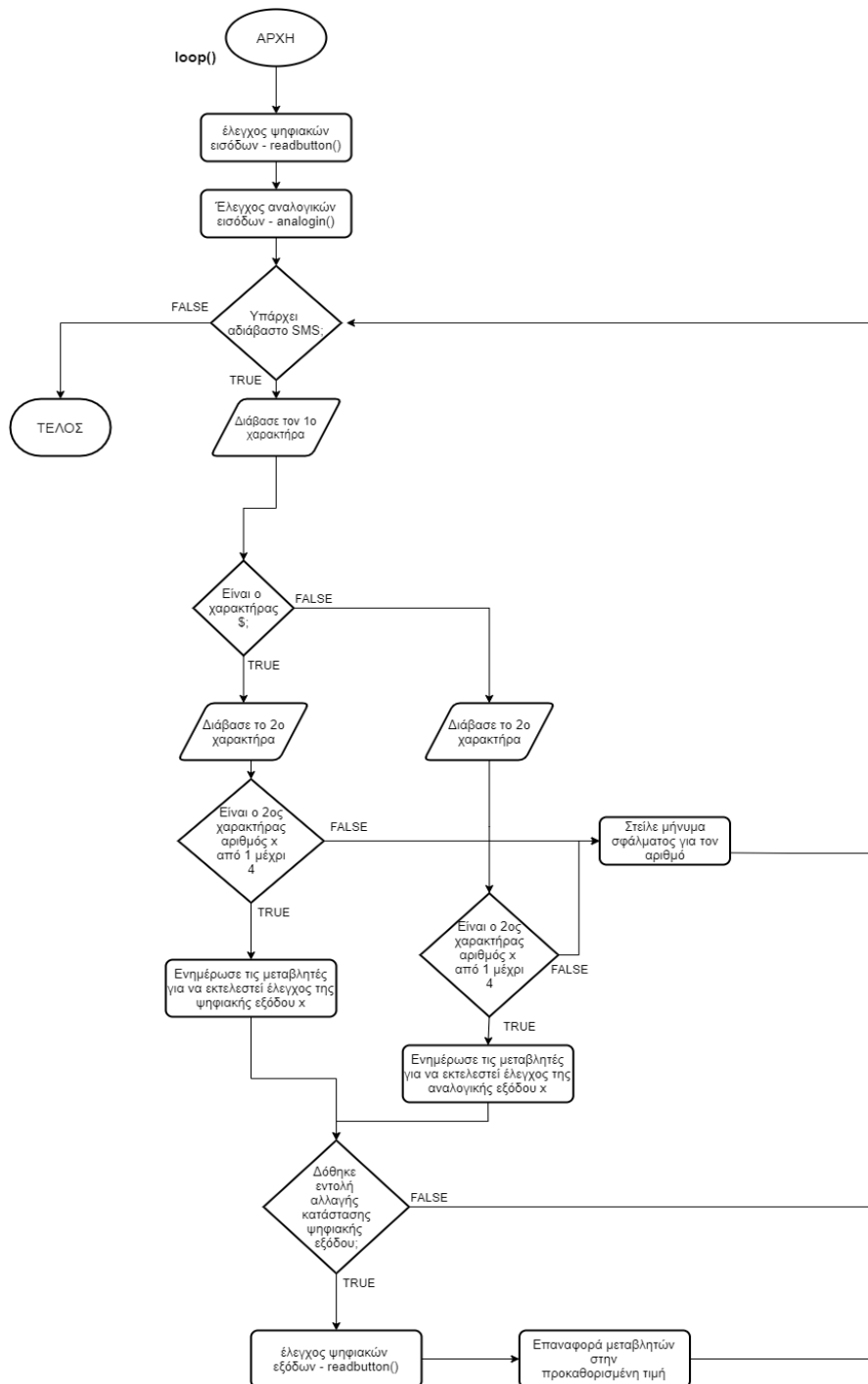


Εικόνα 8.2 Διάγραμμα Ροής της Συνάρτησης Setup()

8.3.3 Η συνάρτηση loop()

Η συνάρτηση loop() είναι επί της ουσίας η κύρια μέθοδος του προγράμματος και έχει την εξής ιδιαιτερότητα. Σε αντίθεση με τις υπόλοιπες συναρτήσεις, η συνάρτηση loop() επαναλαμβάνεται συνεχώς κατά την διάρκεια εκτέλεσης της εφαρμογής, επομένως στο σημείο του διαγράμματος ροής που φτάνουμε στο τέλος η ροή θα συνεχιστεί και πάλι από την αρχή. Στο κύριο πρόγραμμα θέλουμε η εφαρμογή να είναι σε θέση να ελέγχει συνεχώς εάν πατήθηκε κάποιο κουμπί στις ψηφιακές εισόδους και να ενημερώνεται η τιμή των αναλογικών εξόδων. Επίσης χρειάζεται να ελέγχει εάν έχει στείλει ο χρήστης κάποιο νέο μήνυμα. Με την παραλαβή του νέου μηνύματος η εφαρμογή θα πρέπει να είναι σε θέση να ελέγξει εάν το μήνυμα αντιπροσωπεύει μια εντολή, στην δική μας περίπτωση μια εντολή θα πρέπει να ξεκινά με τον χαρακτήρα “\$” για τις ψηφιακές εξόδους ή με τον χαρακτήρα “#” για τις αναλογικές, ακολουθούμενο από έναν αριθμό από το 1 μέχρι το 4. Στην περίπτωση σφάλματος θα ενημερώνεται ο χρήστης με SMS και το πρόγραμμα θα συνεχίζει. Εφόσον δοθεί η εντολή, το πρόγραμμα εκτελεί τις κατάλληλες ενέργειες ξεχωρίζοντας αρχικά εάν πρόκειται για εντολή που αφορά τις ψηφιακές εξόδους ή τις αναλογικές και στην συνέχεια να καθορίζει πια, ανάμεσα από τις 4

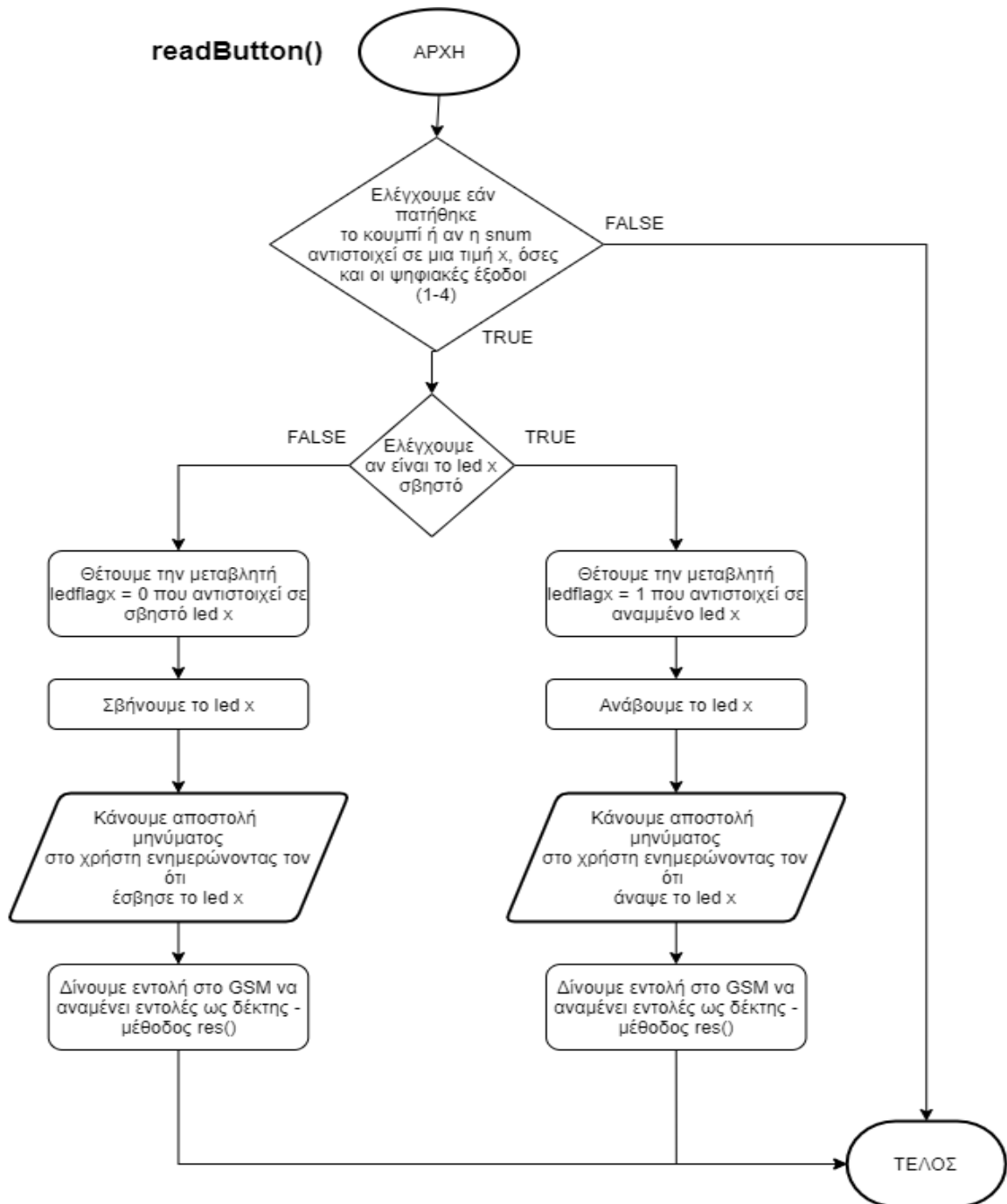
εξόδους, είναι αυτή που θα επεξεργαστεί. Για να επεξεργαστούμε τις εξόδους κατασκευάσαμε δύο συναρτήσεις ή οποίες εκτελούνται στην αρχή του προγράμματος και επηρεάζονται από τις εντολές που στέλνει ο χρήστης. Η πρώτη συνάρτηση είναι η `readbutton()` η οποία αφορά τις ψηφιακές εισόδους και εξόδους και η δεύτερη είναι η `analogin()` που επεξεργάζεται τις αναλογικές. Το διάγραμμα ροής και η μεθοδολογία των συναρτήσεων αυτών θα μελετηθεί στη συνέχεια του κεφαλαίου. Ακολουθεί το διάγραμμα ροής της συνάρτησης `loop()` στο οποίο δίνεται και η γενική περιγραφή της λειτουργικότητας της.



Εικόνα 8.3 Διάγραμμα Ροής της Συνάρτησης `loop()`

8.3.4 Η συνάρτηση readbutton()

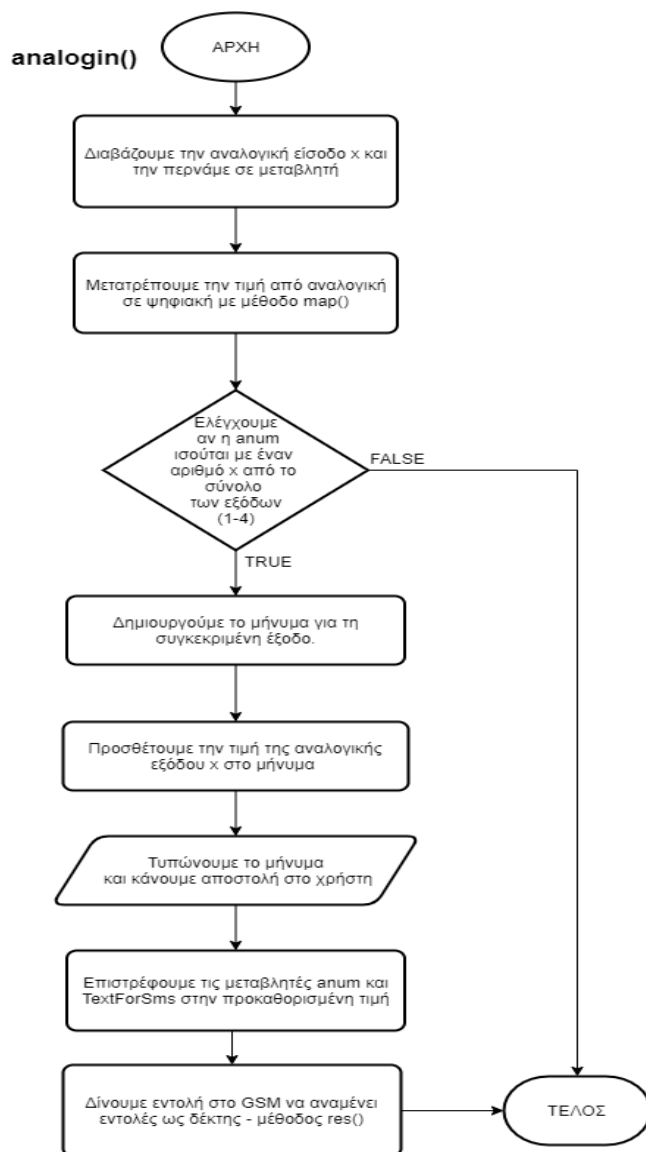
Σκοπός της συνάρτησης readbutton() είναι να ανάβει και να σβήνει τα led των ψηφιακών εξόδων τις εφαρμογής, όταν ο χρήστης είτε πατήσει το αντίστοιχο κουμπί χειροκίνητα, είτε εάν στείλει με SMS την κατάλληλη εντολή για το led σε μία από τις 4 θέσεις ψηφιακών εξόδων. Όπως αναφέραμε προηγουμένως, υπάρχουν δύο συνθήκες για να τεθεί σε λειτουργία η readbutton() και αυτές όπως βλέπουμε στο διάγραμμα ροής παρακάτω, ελέγχονται στην αρχή της συνάρτησης και εάν δεν ισχύουν η συνάρτηση θα τερματίσει χωρίς να εκτελέσει κάποια ενέργεια, η μία συνθήκη είναι να πατηθεί το κουμπί και η άλλη να σταλθεί μήνυμα για μια από τις θέσεις. Ο έλεγχος γίνεται και για τις δύο συνθήκες και η ροή συνεχίζει εάν ισχύει ένα από τα δύο. Εφόσον ισχύει έστω μία από τις δυο συνθήκες, ακολουθεί μια δεύτερη συνθήκη που έχει να κάνει με την κατάσταση του led στη θέση x για την οποία αναφερόμαστε. Το led θα είναι είτε σβηστό, είτε αναμμένο, στην εφαρμογή μας θα ελέγξουμε εάν είναι σβηστό. Στη περίπτωση που είναι σβηστό θέλουμε να γίνει αναμμένο και να ενημερωθεί ο χρήστης ότι το led στη θέση x έχει ανάψει και στην περίπτωση που ήταν είδη αναμμένο θέλουμε να γίνει το αντίστροφο και να ενημερωθεί ο χρήστης ότι το led στη θέση x έχει σβήσει. Σε κάθε μία από τις περιπτώσεις πρέπει να βεβαιωθούμε ότι το GSM θα επανέλθει σε λειτουργία δέκτη μετά την αποστολή του μηνύματος, προτού τερματίσει η συνάρτηση μας για αυτό θα χρησιμοποιήσουμε την συνάρτηση res() και στις δύο περιπτώσεις αμέσως μετά την αποστολή του SMS.



Εικόνα 8.4 Διάγραμμα Ροής της Συνάρτησης readbutton()

8.3.5 Η συνάρτηση analogin()

Η χρήση της analogin() είναι να διαβάζει την τιμή μιας αναλογικής εξόδου που ζητάει ο χρήστης, να μετατρέπει την τιμή αυτή από αναλογική σε ψηφιακή και να ενημερώνει τον χρήστη με SMS για το μέγεθος της. Επομένως, αφού διαβάσουμε και μετατρέψουμε τις τιμές των αναλογικών εξόδων σε ψηφιακή μορφή, ελέγχουμε εάν ο χρήστης έστειλε μήνυμα για μια από τις 4 αναλογικές εξόδους για να μάθει την τιμή. Στη συνέχεια εφόσον πληρείται η συνθήκη, η συνάρτηση δημιουργεί ένα μήνυμα που ενημερώνει το χρήστη για τη τιμή στο συγκεκριμένο ελεγκτή και κάνει αποστολή μέσω SMS. Πριν τερματίσει η συνάρτηση, πρέπει να γίνει ορισμός των μεταβλητών στην προκαθορισμένη τιμή τους και επίσης να τεθεί ο GSM σε λειτουργία δέκτη, ώστε να αναμένει νέο μήνυμα από το χρήστη.



Εικόνα 8.5 Διάγραμμα της Συνάρτησης analogin()

8.3.6 Η συνάρτηση res()

Σε πολλά σημεία του προγράμματος, καλούμαστε να επαναφέρουμε το GSM από κατάσταση αποστολέα SMS σε κατάσταση δέκτη, ώστε να είναι σε θέση ο χρήστης να στείλει εκ νέου κάποιο μήνυμα. Για να μη χρειάζεται να γράφουμε συνέχεια τις συγκεκριμένες γραμμές κώδικα, δημιουργήσαμε την συνάρτηση res(), η οποία έχει αυτή την απλή λειτουργία ωστόσο μας διευκολύνει αρκετά ώστε να έχουμε ένα πιο “καθαρό” κώδικα.



Εικόνα 8.6 Διάγραμμα Ροής Βασικής της Συνάρτησης res()

8.4 Ανάλυση sketch της εφαρμογής

8.4.1 Εισαγωγή βιβλιοθηκών

- Εντολή `#include <stdio.h>` : Εισάγει την βιβλιοθήκη της C `stdio.h` , η οποία μας δίνει πρόσβαση σε μεθόδους επεξεργασίας εισόδων και εξόδων. Θα μας χρησιμεύσει στο να προσθέσουμε τα κουμπιά και τα led στην εφαρμογή.
- Εντολή `#include <stdlib.h>` : Εισάγει την βιβλιοθήκη της C `stdlib.h`, η οποία μας δίνει πρόσβαση σε μεθόδους γενικών καθηκόντων που μας χρησιμεύσουν στην λειτουργία του προγράμματος.
- Εντολή `#include <SoftwareSerial.h>` : Εισαγωγή της βιβλιοθήκης C `SoftwareSerial.h` , ώστε να έχουμε πρόσβαση στις μεθόδους και τα αρχεία της. Η βιβλιοθήκη `SoftwareSerial.h` δημιουργήθηκε με σκοπό να επιτρέπει στον χρήστη να ενεργοποιεί την σειριακή επικοινωνία, αντιγράφοντας την ενσωματωμένη μέθοδο που χρησιμοποιούν τα pin 0 και 1 του Arduino με την λειτουργικότητα του hardware UART, χρησιμοποιώντας software ώστε να μπορεί να

εφαρμόζεται και στα υπόλοιπα pin. Σημειώνεται ότι για το Arduino Mega του πειράματος, ισχύουν οι περιορισμοί που μας εμποδίζουν να αλλάξουμε τη λειτουργικότητα σε κάποια pins, έτσι τα pin που μπορούμε να χρησιμοποιήσουμε είναι τα ακόλουθα: RX: 10, 11, 12, 13, 14, 15, 50, 51, 52, 53, A8 (62), A9 (63), A10 (64), A11 (65), A12 (66), A13 (67), A14 (68), A15 (69).

- `SoftwareSerial smsSerial(12, 13);` : Καλεί τον constructor της software serial. `SoftwareSerial(rxPin, txPin, inverse_logic)` Όπου η μεταβλητή `rxPin` ορίζει το pin που θα λειτουργεί ως δέκτης σειριακών δεδομένων στο pin 12 και η μεταβλητή `txPin` ορίζει το pin που θα λειτουργεί ως δέκτης στο pin 13. Η μεταβλητή `inverse_logic` είναι προαιρετική με προκαθορισμένη τιμή `false` όταν δεν ορίζεται και χρησιμεύει στο να αντιστρέφει της λογικές τιμές των bit της εισόδου.

-

8.4.2 Αρχικοποίηση – Ορισμός μεταβλητών

- `const int buttonx = y;` : Δημιουργούμε μια σταθερά για το κουμπί x στην οποία περνάμε μια τιμή y τύπου `integer` η οποία ορίζει το pin στο οποίο θα γίνει η σύνδεση π.χ `const int button1 = 42;` Ορίζεται η σταθερά τύπου `integer` για το κουμπί 1 με τιμή τον αριθμό 42 που αντιστοιχεί στο pin 42 του arduino.
- `const int ledx = y;` : Δημιουργούμε μια σταθερά για το led x στην οποία περνάμε μια τιμή y τύπου `integer` η οποία ορίζει το pin στο οποίο θα γίνει η σύνδεση π.χ `const int led1 = 3;` Ορίζεται η σταθερά τύπου `integer` για το led 1 με τιμή τον αριθμό 3 που αντιστοιχεί στο pin 3 του arduino.
- `int ledflagx = 0;` : Η μεταβλητή `ledflagx`, όπου x είναι ο αριθμός του led που χρησιμοποιούμε για έξοδο των ψηφιακών εισόδων των κουμπιών, μας βοηθάει στο να εντοπίζουμε την κατάσταση του συγκεκριμένου led. Δίνοντας ως αρχική τιμή το 0, καθορίζουμε ότι η αρχική κατάσταση του led είναι κλειστό. Αν κατά την λειτουργία της συσκευής το led ανάψει θα αλλάξουμε την τιμή από 0 σε 1. `const`
- `int analog1 = A9;` : Δημιουργούμε μια σταθερά τύπου `integer` με όνομα `analog1` και της δίνουμε την τιμή A9 που αντιστοιχεί στο pin A9 του arduino το οποίο θα χρησιμοποιηθεί ως η πρώτη αναλογική είσοδος του κυκλώματος. Στην συνέχεια, ακολουθώντας την ίδια λογική ορίζουμε και τις υπόλοιπες τρεις αναλογικές εισόδους στα αντίστοιχα pin A3, A13 και A0.
- `const int analogOut1 = 4;` : Ορίζουμε την σταθερά τύπου `integer` με όνομα `analogOut1`, η οποία αντιπροσωπεύει την πρώτη αναλογική έξοδο του κυκλώματος και της δίνουμε την τιμή 4 η οποία αντιστοιχεί στο pin 4 του arduino. Στην συνέχεια με τον ίδιο τρόπο ορίζουμε τα pin των υπόλοιπων τριών αναλογικών εξόδων στα αντίστοιχα pin 5, 6 και 9.
- `int sensorValuex = 0;` : Δημιουργούμε την μεταβλητή `sensorValuex` που αντιπροσωπεύει την τάση που μετράμε στο ποτενσιόμετρο και την αρχικοποιούμε με την τιμή 0.
- `int outputValuex = 0;` : Δημιουργούμε την μεταβλητή `outputValuex` που αντιπροσωπεύει την τάση στην αναλογική έξοδο x και την αρχικοποιούμε με την τιμή 0.
- `String TextForSms ;` : Δημιουργούμε την μεταβλητή `TextForSms` τύπου `String`, την οποία θα χρησιμοποιούμε κατά την διάρκεια εκτέλεσης του προγράμματος, ώστε να αποθηκεύσουμε το μήνυμα που θα στέλνουμε με sms στον χρήστη.
- `char input_string[15];` : Δημιουργούμε έναν πίνακα τύπου `char` για να αποθηκεύουμε και να επεξεργαζόμαστε τα μηνύματα που θα λαμβάνουμε στο σύστημα μας.

- `int var_1 = 0;` : Η `var_1` είναι μια βοηθητική μεταβλητή που θα παίρνει τις τιμές 0 και 1. Όποτε θα παίρνει την τιμή 1 θα καλούμε την μέθοδο `readButton()` που θα αναλύσουμε στην συνέχεια.
- `int var_2 = 0;` : Η `var_2` χρησιμοποιείται για να ελέγχουμε με τη σειρά τους χαρακτήρες στα απεσταλμένα μηνύματα που θα αποθηκεύονται στον πίνακα `input_string`.
- `int snum = 0;` : Η `snum` αντιπροσωπεύει μεταβολή στην ψηφιακή έξοδο `x`. Το εύρος τιμών της στο σύστημα μας είναι 0-4. Από 1-4 για κάθε μεταβολή στα 4 led της ψηφιακής εξόδου της εφαρμογής και 0 για όλες τις υπόλοιπες στιγμές που δεν έχουμε εντολή για αλλαγή τιμής.
- `int anum = 0;` : Η `anum` αντιπροσωπεύει την μεταβολή στη τιμή στον αναλογικό αισθητήρα `x`. Το εύρος τιμών της στο σύστημα μας είναι 0-4. Από 1-4 για κάθε μεταβολή στους 4 αισθητήρες της εφαρμογής και 0 για όλες τις υπόλοιπες στιγμές που δεν έχουμε κάποια αλλαγή τιμής.

8.4.3 Η συνάρτηση `setup()`

Η συνάρτηση `setup` , εκτελείται πριν από την έναρξη του κυρίου προγράμματος μία φορά και κάνει τις απαραίτητες προεργασίες ώστε να αρχικοποιηθούν τα pin του arduino στις κατάλληλες καταστάσεις, καθώς και η προετοιμασία για την ανταλλαγή των sms.

- `pinMode(X,mode);` : Η `pinMode` είναι μια μέθοδος τύπου `void` του arduino, η οποία χρησιμεύει στο να ορίζει ένα pin `X` είτε ως είσοδο (`input`) , είτε ως έξοδο (`output`).
- `digitalWrite(pin, mode);` : Η `digitalWrite` είναι μια μέθοδος τύπου `void` του arduino, η οποία δέχεται σαν παραμέτρους το pin στο οποίο έχουμε συνδέσει ένα led και ένα mode που πρέπει να είναι HIGH ή LOW. Ανάλογα με την ρύθμιση του pin, δηλαδή αν λειτουργεί σαν είσοδος ή έξοδος, έχουμε και διαφορετική χρησιμότητα στην μέθοδο. Εάν το pin είναι ρυθμισμένο σαν output και το mode είναι HIGH τότε θέτουμε το pin στα 5V, με αποτέλεσμα το led να ανάψει. Ενώ όταν βάζουμε mode LOW η τιμή θα οριστεί στα 0V της γείωσης και το led θα σβήνει.
- `Serial.begin(9600);` : Με την μέθοδο `begin` ξεκινάμε την σειριακή επικοινωνία μεταξύ του arduino και του συστήματος. Ο αριθμός 9600 ονομάζεται baud rate και είναι ο ρυθμός με τον οποίο θα στέλνονται τα δεδομένα, ο αριθμός αυτός θα πρέπει να είναι ίδιος και για τον αποστολέα και για τον δέκτη ώστε να πετύχει η επικοινωνία των συσκευών.
- `smsSerial.begin(9600);` : Με τον ίδιο τρόπο ξεκινάμε την επικοινωνία για το σύστημα sms της εφαρμογής.
- `smsSerial.println("AT+CMGD=1,4");` : Με αυτή την AT εντολή πραγματοποιούμε διαγραφή όλων των μηνυμάτων.
- `smsSerial.println("AT+CMGF=1");` : Θέτουμε το GSM σε λειτουργία text mode.
- `smsSerial.println("AT+CNMI=1,2,0,0,0");` : Ενημερώνουμε το GSM να αποστείλει αυτόματα ειδοποίηση στη σειριακή του θύρα σε περίπτωση που λάβει κάποιο μήνυμα.

8.4.4 Η συνάρτηση `res()`

Η συνάρτηση `res()` βάζει το GSM σε λειτουργία text mode με την εντολή “AT+CMGF=1” και στην συνέχεια δίνει την εντολή “AT+CNMI=1,2,0,0,0” στο σύστημα μας. Με αυτό τον τρόπο ενημερώνουμε το GSM να αποστείλει αυτόματα μια ειδοποίηση στη σειριακή του θύρα, σε περίπτωση που λάβει κάποιο μήνυμα και να λειτουργήσει σαν δέκτης. Μας είναι χρήσιμη για να

συνεχίσει η εφαρμογή μας να λειτουργεί και να είναι έτοιμη να δεχθεί νέα μηνύματα – εντολές, αφότου έχει εκτελέσει κάποια άλλη ενέργεια όπως θα δούμε παρακάτω.

8.4.5 Η συνάρτηση analogIn()

Ο σκοπός της συνάρτησης analogIn() είναι να διαβάζει τις τιμές στις αναλογικές εξόδους του κυκλώματος και να στέλνει μέσω sms την τιμή της αναλογικής εξόδου που ζητά ο χρήστης. Αρχίζουμε την συνάρτηση διαβάζοντας τις εξόδους στα αναλογικά pin του συστήματος:

- `sensorValueX = analogRead(analogX);` : Περνάμε σε μια μεταβλητή `sensorValueX` την τιμή που θα διαβάσει η `analogRead` από την αναλογική είσοδο `analogX`.
- `outputValueX = map(sensorValueX, 0, 1023, 0, 255);` : χρησιμοποιούμε την μέθοδο `map` για να πάρουμε την τιμή εξόδου του `x` σε μορφή 8bit, αλλάζοντας τα όρια των τιμών από `min=0` και `max=1023` (αναλογικές τιμές) σε `min=0` και `max=255`. Χρειαζόμαστε αυτή τη μετατροπή για να καλέσουμε στην συνέχεια την `analogWrite`, η οποία προϋποθέτει τα συγκεκριμένα όρια τιμών.
- `analogWrite(analogOutX, outputValueX);` : Αφού αλλάξαμε τα όρια της `sensorValueX`, έχουμε την δυνατότητα να χρησιμοποιήσουμε την `analogWrite` για να δώσουμε την μετρήσιμη τιμή που πήραμε για την αναλογική έξοδο `x`.

Στην συνέχεια ελέγχουμε αν η μεταβλητή `anum` έχει πάρει κάποια από τις τιμές 1-4, που σηματοδοτεί αλλαγή της προηγούμενης τιμής της `analogOut` για την οποία ο χρήστης θα ειδοποιηθεί με sms. Ο έλεγχος γίνεται με την εντολή `if (anum == x)`. Αρχικά γράφουμε το μήνυμα που θέλουμε να στείλουμε στην μεταβλητή `TextForSms`, στο οποίο θα αναγράφεται ο αισθητήρας που υπέστη την μεταβολή και η νέα του τιμή.

- `TextForSms = TextForSms + "Senx:";`
- `TextForSms.concat(sensorValueX);`

Μετά την δημιουργία του μηνύματος χρησιμοποιούμε την `smsSerial` για να στείλουμε τις κατάλληλες εντολές στο GSM ώστε να γίνει η αποστολή του.

- `smsSerial.println("AT+CMGF=1");` : Αλλάζουμε τον τρόπο λειτουργίας του μόντεμ σε Text mode.

`delay(500);`

- `smsSerial.println("AT+CMGS=\"" + 30 + number + "\"");` : Η AT εντολή `AT+CMGS` δέχεται σαν παράμετρο αρχικά τον αριθμό στον οποίο θέλουμε να γίνει η αποστολή και στην συνέχεια το μήνυμα.
- `delay(500);`
- `smsSerial.print(TextForSms);` : Γράφουμε το περιεχόμενο της `TextForSms` στο μήνυμα.
- `delay(500);`
- `smsSerial.write(26);` : Ο αριθμός 26 είναι η κωδικοποίηση του CTRL + Z στην ASCII και σηματοδοτεί το τέλος του μηνύματος για να αρχίσει η αποστολή του στον αριθμό που δώσαμε.
- `delay(100);`

Αρχικοποιούμε ξανά τις μεταβλητές `TextForSms` και `anum` στις προεπιλεγμένες τιμές τους.

- `TextForSms = "";`

- `anum = 0;`

Τέλος καλούμε τη συνάρτηση `res()` για να βάλουμε το GSM σε κατάσταση αναμονής για νέες ενέργειες και να συνεχίσει η λειτουργία της εφαρμογής.

8.4.6 Η Συνάρτηση `readbutton()`

Ο σκοπός της συνάρτησης `readbutton` είναι να κάνει έλεγχο της κατάστασης των ψηφιακών εξόδων του κυκλώματος, να ενημερώνει το χρήστη και εάν τυχόν πάρει την εντολή μέσω sms από αυτόν να αλλάξει και την κατάσταση ενός συγκεκριμένου led. Ξεκινάμε την συνάρτηση κάνοντας έλεγχο εάν το κουμπί έχει πατηθεί ή αν έχει δοθεί εντολή μέσω μηνύματος για την ενεργοποίηση της ψηφιακής εξόδου ώστε να φωτοβολήσει το led x. Ο έλεγχος γίνεται με την `if (digitalRead(button) == HIGH || snum == x)`. Στην συνέχεια ελέγχουμε εάν το led ήταν αναμμένο ή κλειστό. Κάνουμε τον έλεγχο με μια εμφωλευμένη `if (ledflagx == 0)` που μεταφράζεται σε εάν το led του pin x είναι σβηστό. Τότε ανάβουμε το led δίνοντας τις εντολές:

- `ledflagx = 1;` : Ενημερώνουμε την εφαρμογή για την αλλαγή της κατάστασης του led x από σβηστό σε αναμμένο.
- `digitalWrite(ledx,HIGH);` : Αλλάζουμε την κατάσταση στην ψηφιακή είσοδο του led x από LOW σε HIGH ώστε να ανάψει το led.

Ακολούθως ενημερώνουμε τον χρήστη με sms για την μεταβολή της κατάστασης του led x.

- `smsSerial.println("AT+CMGF=1");` : Δίνουμε εντολή στο GSM να μπει σε λειτουργία Text mode.
- `delay(500);`
- `smsSerial.println("AT+CMGS=\"" + 30 + number + "\");` : Δίνουμε την εντολή για αποστολή μηνύματος στον αριθμό τηλεφώνου που έχουμε αποθηκεύσει στην μεταβλητή `number`.
- `delay(500);`
- `smsSerial.print("LEDx ON");` : Δημιουργούμε το μήνυμα που θα αποσταλεί στον αριθμό, στο οποίο θα ενημερώνουμε τον χρήστη ότι το led x είναι σε κατάσταση ON.
- `delay(500);`
- `smsSerial.wirte(26);` : Ο αριθμός 26 είναι η κωδικοποίηση του CTRL + Z στην ASCII και σηματοδοτεί το τέλος του μηνύματος για να αρχίσει η αποστολή του στον αριθμό που δώσαμε.
- `delay(100);`
- `res();` : Αφού τελειώσαμε με την αποστολή του μηνύματος καλούμε την `res()` για να θέσουμε το GSM σε κατάσταση αναμονής για νέες ενέργειες.

Στην περίπτωση `else` που αντιπροσωπεύει την κατάσταση `ledflagx == 1`, δηλαδή το led x να είναι είδη αναμμένο, τότε ακολουθούμε την αντίστροφη λογική ώστε να αλλάξει κατάσταση και να σβήσει. Αρχικά σβήνουμε το led x δίνοντας τις εντολές :

- `ledflagx =0;` : Ενημερώνουμε την εφαρμογή για την αλλαγή της κατάστασης του led x από αναμμένο σε σβηστό.
- `digitalWrite(ledx,LOW);` : Αλλάζουμε την κατάσταση στην ψηφιακή είσοδο του led x από HIGH σε LOW ώστε να σβήσει το led.

Ακολούθως ενημερώνουμε τον χρήστη με sms για την μεταβολή της κατάστασης του led x.

- `smsSerial.println("AT+CMGF=1");` : Δίνουμε εντολή στο GSM να μπει σε λειτουργία Text mode.
- `delay(500);`
- `smsSerial.println("AT+CMGS=\"+30\"+number+\" \");` : Δίνουμε την εντολή για αποστολή μηνύματος στον αριθμό τηλεφώνου που έχουμε αποθηκεύσει στην μεταβλητή `number`.
- `delay(500);`
- `smsSerial.print("LEDx OFF");` : Δημιουργούμε το μήνυμα που θα αποσταλεί στον αριθμό, στο οποίο θα ενημερώνουμε τον χρήστη ότι το led x είναι σε κατάσταση OFF.
- `delay(500);`
- `smsSerial.wirte(26);` : Ο αριθμός 26 είναι η κωδικοποίηση του CTRL + Z στην ASCII και σηματοδοτεί το τέλος του μηνύματος για να αρχίσει η αποστολή του στον αριθμό που δώσαμε.
- `delay(100);`
- `res();` : Αφού τελειώσαμε με την αποστολή του μηνύματος καλούμε την `res()` για να θέσουμε το GSM σε κατάσταση αναμονής για νέες ενέργειες.
-

8.4.7 Η συνάρτηση `loop()`

Η συνάρτηση `loop()` είναι η κύρια μέθοδος του προγράμματος η οποία θα τρέχει σε όλη τη διάρκεια εκτέλεσης του και σκοπός της είναι να ενημερώνει το σύστημα όποτε δέχεται ένα μήνυμα από το χρήστη για να πραγματοποιήσει τις καθορισμένες ενέργειες, καθώς επίσης να ενημερώνει το χρήστη για τυχόν μεταβολές στη κατάσταση των αναλογικών και ψηφιακών εισόδων και εξόδων. Ξεκινώντας καλούμε τις συναρτήσεις `readbutton()` και `analogin()` για να ελέγξουμε όλες τις αναλογικές και ψηφιακές θύρες του κυκλώματος και να ενημερώνουμε τον χρήστη για τυχόν αλλαγές με γραπτό μήνυμα `sms`. Εκτός από το να ενημερώνουμε τον χρήστη θέλουμε να έχουμε και την δυνατότητα ελέγχου του συστήματος μέσω `sms`.

- `while (smsSerial.available())` : Κάνουμε χρήση της εντολής επανάληψης `while` για να ελέγξουμε εάν υπάρχουν ληφθέντα μηνύματα διαθέσιμα για ανάγνωση από την σειριακή πύλη.
- `char input_char = smsSerial.read();` : Διαβάζουμε το πρώτο διαθέσιμο byte από τον σειριακό δέκτη του GSM και τον αποθηκεύουμε στην μεταβλητή `input_char`.
- `input_char[var_2++] = input_char;` : Στην συνέχεια βάζουμε το περιεχόμενο της `input_char` στην πρώτη θέση του πίνακα `input_char` και αλλάζουμε την θέση του pointer στη θέση `var_2 + 1`.
- `Serial.println(input_char);`

Ο πρώτος χαρακτήρας καθορίζει εάν θέλουμε να πάρουμε πληροφορίες για τις ψηφιακές εξόδους ή για τις αναλογικές. Για τις ψηφιακές θα πρέπει ο χρήστης να ξεκινά το μήνυμα του με το χαρακτήρα "\$", ενώ για τις αναλογικές τον χαρακτήρα "#".

- `if (input_char == "$")` : Εάν ο πρώτος χαρακτήρας είναι "\$" προχωράμε στην ανάγνωση του δεύτερου χαρακτήρα που θα αφορά τη ψηφιακή έξοδο.
- `char input_char = smsSerial.read();` : Κάνουμε ανάγνωση του επόμενου byte για να πάρουμε το δεύτερο χαρακτήρα.
- `input_char[var_2++] = input_char;` : Αποθηκεύουμε το περιεχόμενο στην επόμενη θέση του πίνακα `input_char`.

- `if (input_char == "x")` : Ο χαρακτήρας που διαβάστηκε θα πρέπει να είναι ένας αριθμός x με εύρος τιμών 1 έως 4, όσες και οι ψηφιακές μας έξοδοι. Ο αριθμός θα καθορίζει πιο led θα αλλάξει την κατάσταση του.
- `snum = x;` : Δίνουμε στην μεταβλητή `snum` την τιμή x για να αλλάξουμε την κατάσταση του led x όταν αργότερα θα καλεστεί η `readbutton()`.
- `var_1 = 1;` : Χρησιμοποιούμε την βοηθητική μεταβλητή `var_1` και ορίζουμε την τιμή της στο 1, ώστε να καλέσουμε στη συνέχεια την `readbutton()`.
- `if (input_char == "#")` : Εάν ο πρώτος χαρακτήρας είναι "#" προχωράμε στην ανάγνωση του δεύτερου χαρακτήρα που θα αφορά την αναλογική έξοδο.
- `char input_char = smsSerial.read();` : Κάνουμε ανάγνωση του επόμενου byte για να πάρουμε το δεύτερο χαρακτήρα.
- `input_char[var_2++] = input_char;` : Αποθηκεύουμε το περιεχόμενο στην επόμενη θέση του πίνακα `input_char`.
- `if (input_char == "x")` : Ο χαρακτήρας που διαβάστηκε θα πρέπει να είναι ένας αριθμός x με εύρος τιμών 1 έως 4, όσες και οι αναλογικές μας έξοδοι. Ο αριθμός καθορίζει για πια από τις 4 αναλογικές εξόδους θα ενημερωθεί για την τιμή της ο χρήστης.
- `anum = x;` : Δίνουμε στη μεταβλητή `anum` την τιμή x για να ελεγχθεί η τιμή της αναλογικής εξόδου x από την μέθοδο `analogin()` και να ενημερωθεί για αυτήν ο χρήστης.
- `analogin();` : Καλούμε την μέθοδο `analogin()`.

Τέλος ελέγχουμε εάν η μεταβλητή `var_1` είναι ίση με 1 για να καλέσουμε την `readbutton()` που θα μας ενημερώσει για την κατάσταση των ψηφιακών εξόδων και θα αλλάξει την κατάσταση στο led που έχει υποδείξει ο χρήστης στο sms.

- `if (var_1 == 1)`
- `readbutton();`
- Μετά την εκτέλεση της `readbutton()` θέτουμε τις μεταβλητές `var_1`, `var_2` και `snum` στην αρχική, προκαθορισμένη τους τιμή.
- `var_1 = 0;`
- `var_2 = 0;`
- `snum = 0;`
- `delay(5);`
-

8.4.8 Έλεγχος σφαλμάτων κατά την αποστολή sms

Όπως έχει αναφερθεί και σε προηγούμενο κεφάλαιο, βασική προϋπόθεση για να στείλει το σύστημα μας ενημερωτικό μήνυμα στο χρήστη για την κατάσταση την οποία βρίσκεται η κάθε έξοδος ή για να επικοινωνήσει ο χρήστης μια από τις ψηφιακές ή αναλογικές εισόδους, είναι το μήνυμα που θα αποστείλει ο χρήστης να έχει την μορφή "\$ {και ένας αριθμός από 1 μέχρι 4 που αντιπροσωπεύει τις εξόδους }" για τις ψηφιακές εισόδους ή "# {και ένας αριθμός από 1 μέχρι 4 που αντιπροσωπεύει τις εξόδους } για τις αναλογικές. Χρειαζόμαστε λοιπόν έναν τρόπο με τον οποίο θα γίνεται έλεγχος του μηνύματος ώστε ο χρήστης να ενημερωθεί με sms από την εφαρμογή, όταν το μήνυμα του δεν έχει την κατάλληλη μορφή. Είναι αναγκαίο ο έγκυρος χαρακτήρας στην εντολή που δίνει ο χρήστης να ακολουθείται και από έναν έγκυρο αριθμό. Έγκυρος αριθμός θεωρείται κάθε αριθμός από το 1 μέχρι και το 4. Όπως εξηγήθηκε προηγουμένως, κάθε αριθμός ανάλογα το σύμβολο που προηγείται στην εντολή αντιστοιχεί σε μια είσοδο. Η προϋπόθεση αυτή επομένως πρέπει να ισχύει και για τις ψηφιακές εξόδους αλλά και για τις αναλογικές. Επομένως και για τις δύο περιπτώσεις όπου ο χρήστης

έβαλε το σωστό χαρακτήρα προσθέτουμε την εντολή ελέγχου if (input_char != '1' && input_char != '2' && input_char != '3' && input_char != '4'). Εάν λοιπόν ο επόμενος χαρακτήρας δεν είναι κάποιο από τα έγκυρα νούμερα, συνεχίζουμε στην αποστολή του ενημερωτικού μηνύματος σφάλματος, αυτή τη φορά για το δεύτερο χαρακτήρα του μηνύματος, δηλαδή τον αριθμό, με τον ίδιο τρόπο όπως και στην πρώτη περίπτωση όπως φαίνεται παρακάτω:

```
    smsSerial.println("AT+CMGF=1");  
  
    delay(500);  
  
    smsSerial.println("AT+CMGS=\"+30"+number+"\"");  
  
    delay(500);  
  
    smsSerial.print("Wrong Message! The only valid output ports are between 1 and 4!");  
  
    delay(500);  
  
    smsSerial.write(26);  
  
    delay(100);  
  
    res();
```

Είναι σημαντικό για την εφαρμογή μας να υπάρχουν αυτές οι εντολές ελέγχου, ώστε να υπάρχει αλληλεπίδραση με το χρήστη και σε περίπτωση που γίνει κάποιο λάθος να μπορεί να ενημερώνεται από το σύστημα, ειδάλλως το σύστημα θα έμενε απλά αδρανές χωρίς να γνωρίζουμε σε τι μπορεί να οφείλεται αλλά και χωρίς να γνωρίζουμε την κατάσταση στην οποία θα βρισκόταν.

8.5 Επίλογος

Στο κεφάλαιο που ακολούθησε έγινε μια αναφορά στην προεργασία του προγραμματισμού της πλακέτας του Arduino και στη συνέχεια καταγράφηκε το διαγραμμάτων ροής της εφαρμογής. Ακολούθησε η ανάλυση sketch της εφαρμογής καθώς και η ανάλυση του κώδικα σε κομμάτια ανάλογα το περιεχόμενό τους. Τέλος έγινε και μια αναφορά όσον αφορά κάποιο σφάλμα που πιθανόν να προκύψει ως προς την επικοινωνία του χρήστη με το σύστημα ελέγχου.

Κεφάλαιο 9ο: Συμπεράσματα και προτάσεις βελτίωσης

9.1 Εισαγωγή

Στο κεφάλαιο που θα ακολουθήσει θα αναφερθούν κάποια συμπεράσματα που προέκυψαν έπειτα από την ενασχόλησή μας με τη συγκεκριμένη εφαρμογή και κατασκευή. Επιπλέον θα καταγραφούν και κάποιοι τρόποι όπου η συγκεκριμένη εργασία μπορεί να αναπτυχθεί και να εξελιχθεί στο μέλλον βελτιώνοντας ακόμα περισσότερο τη λειτουργία και τις δυνατότητες του Arduino και κατ'επέκταση και της συσκευής, αλλά και την εμπειρία του χρήστη κατά τη διάρκεια επικοινωνίας του με το σύστημα ελέγχου. Επίσης θα καταγραφούν κάποια πιθανά προβλήματα που μπορεί να προκύψουν κατά τη χρήση της κατασκευής, οι τομείς που μπορεί να φανεί χρήσιμη η κατασκευή αλλά και κάποια προσωπικά προβλήματα που αντιμετωπίσαμε κατά τη διάρκεια της κατασκευής της συσκευής.

9.2 Συμπεράσματα

Μέσω της συγκεκριμένης Διπλωματικής εργασίας, μπορέσαμε και ανακαλύψαμε τον κόσμο του Arduino με μεγαλύτερη λεπτομέρεια. Καταλάβαμε ότι οι δυνατότητες που έχει μια πλακέτα Arduino έπειτα από σωστό προγραμματισμό, είναι υπέρογκες, πρακτικές και άκρως ενδιαφέρουσες. Αυτός ο μικροελεγκτής λοιπόν είναι ικανός να διαβάζει και να παρουσιάζει καταστάσεις από το περιβάλλον χρησιμοποιώντας διάφορους αισθητήρες αλλά και να δράσει σε αυτό με τα κατάλληλα εξαρτήματα. Αντιλαμβάνεται λοιπόν κανείς ότι για την ορθή χρήση του Arduino αλλά και για τη σωστή λειτουργία μιας κατασκευής βασισμένη σε Arduino χρειάζονται δύο τμήματα. Το πρώτο τμήμα είναι το Software, δηλαδή ο προγραμματισμός του Arduino και το δεύτερο τμήμα είναι το Hardware, δηλαδή οι αισθητήρες τα button τα led και οτιδήποτε χρησιμοποιηθεί για κάθε κατασκευή. Συμπερασματικά, καταλήγουμε ότι μια κατασκευή Arduino μπορεί να είναι από πολύ απλή και οικονομική έως πολύ σύνθετη και με περισσότερα έξοδα.

9.3 Πιθανά προβλήματα

Τα προβλήματα που μπορεί να προκύψουν σε μια τέτοια κατασκευή είναι αρκετά γι' αυτό και σε επόμενο κεφάλαιο θα υπάρξουν προτάσεις βελτίωσης της κατασκευής. Ένα από τα βασικότερα θέματα που πιθανόν να αντιμετωπίσει ο χρήστης κατά τη διάρκεια της εμπειρίας του με το σύστημα είναι μια πιθανή διακοπή ρεύματος. Με την κατασκευή να έχει αποκλειστική τροφοδοσία μέσω της εξωτερικής τροφοδοσίας, μια διακοπή ρεύματος θα μπορούσε να έχει ακόμα και καταστροφικές συνέπειες για τον χρήστη, διότι η επικοινωνία του με το συγκεκριμένο σύστημα ελέγχου θα ήταν αδύνατη. Επομένως ο χρήστης δεν θα μπορούσε να δώσει κάποια εντολή στο σύστημα αλλά ούτε και να μάθει την οποιαδήποτε κατάσταση του συστήματος.

Ένα επίσης αρκετά μεγάλο πρόβλημα που είναι πιθανό να αντιμετωπίσει ο χρήστης στην επικοινωνία του με το σύστημα είναι λόγω σήματος. Πολλές φορές αντιμετωπίζουμε προβλήματα σύνδεσης και σήματος με τον πάροχο της κινητής μας τηλεφωνίας. Σε περίπτωση δηλαδή που ο χρήστης δεν έχει σήμα στη κινητή του συσκευή η επικοινωνία με το σύστημα καθίσταται αδύνατη ή μπορεί να υπάρξει καθυστέρηση στην επικοινωνία. Εκτός από το σήμα ένας ακόμη περιορισμός που έχουμε κάνοντας χρήση ενός δικτύου που μας προσφέρει κάποιος πάροχος κινητής τηλεπικοινωνίας είναι ότι μας υποχρεώνει να χρησιμοποιούμε τα πακέτα τηλεφωνίας που διαθέτει με τις ανάλογες χρεώσεις, στην εφαρμογή μας δεν είναι και τόσο σημαντικό αυτό το κόστος ωστόσο σε μεγαλύτερες εφαρμογές είναι κάτι που θα πρέπει να ληφθεί υπόψη. Στο κεφάλαιο που θα ακολουθήσει θα γίνει αναφορά στα συγκεκριμένα προβλήματα επικοινωνίας αλλά θα αναφερθούν και προτάσεις βελτίωσης και εξάλειψης αυτών των προβλημάτων.

Τέλος ένα πρόβλημα που ίσως είναι αναπόφευκτο κυρίως σε κατασκευές που είναι εκτεθειμένες στις καιρικές συνθήκες είναι η φθορά των υλικών. Στη συγκεκριμένη κατασκευή, εάν η στεγανοποίηση και ασφάλεια των υλικών δεν είναι αυτή που πρέπει τότε η πιθανότητα για δυσλειτουργία του συστήματος είτε λόγω υγρασίας, είτε λόγω υπερθέρμανσης ή ακόμα και σκόνης είναι μεγάλη. Τα περισσότερα από τα πιθανά προβλήματα λύνονται και οι πιθανές βελτιώσεις που μπορεί να γίνουν στη συσκευή θα αναφερθούν στο επόμενο κεφάλαιο.

Ένα άλλο πιθανό πρόβλημα που μπορεί να αντιμετωπίσει ο χρήστης κατά την επικοινωνία του με το σύστημα ελέγχου είναι η λανθασμένη σύνταξη της εντολής. Αναλυτικότερα, ο χρήστης, στέλλοντας λάθος εντολή θα πρέπει να ενημερώνεται από το σύστημα για το λάθος που έχει κάνει. Στην συγκεκριμένη εργασία, όταν ο χρήστης στέλνει λάθος αριθμό, δέχεται μήνυμα από το σύστημα

για το λάθος του. Σε περίπτωση που ο χρήστης στείλει σκέτο αριθμό ως εντολή χωρίς να προηγείται # ή \$ τότε το σύστημα δεν αλλάζει την κατάσταση του και δεν στέλνει κάποιο ενημερωτικό μήνυμα στον χρήστη. Σε αυτή την περίπτωση θα μπορούσε το σύστημα να στείλει ένα επιπλέον μήνυμα ελέγχου στο χρήστη για την ορθή σύνταξη της εντολής.

9.4 Προτάσεις βελτίωσης

Στο συγκεκριμένο project το οποίο πραγματεύεται η Διπλωματική εργασία και έπειτα από την εμπειρία που αποκτήσαμε κατά τη διάρκεια της συγγραφής της, οι πιθανές βελτιώσεις που μπορούν να γίνουν στην κατασκευή αλλά και στο προγραμματιστικό τμήμα είναι αρκετές. Μια βασική αναβάθμιση του συστήματος είναι η επικοινωνία του χρήστη με τη κατασκευή μέσω WiFi ή 5G. Η επικοινωνία αποκλειστικά μέσω SMS βάζει περιορισμούς στο χρήστη. Η επιλογή επικοινωνίας μέσω SMS έγινε διότι πολλές φορές βρισκόμαστε σε κατάσταση που η εύρεση σήματος WiFi δεν είναι εύκολη, ωστόσο η επιλογή να μπορούμε να επικοινωνήσουμε με πολλούς τρόπους με τη συσκευή είναι μια σημαντική αναβάθμιση.

Ακόμα μεγαλύτερη βελτίωση θα ήταν και η δημιουργία μιας εφαρμογής μέσω της οποίας θα μπορούσαμε να επικοινωνούμε και να ελέγχουμε το σύστημα. Στον κατασκευαστικό τομέα μπορούμε να αναβαθμίσουμε την εργασία μας προσθέτοντας μια επικάλυψη πλαστικού για την προστασία της πλακέτας. Με αυτό τον τρόπο επίσης, θα είναι ευκολότερο να την τοποθετήσουμε στο χώρο εργασίας. Η συντήρηση των υλικών αποτελεί ένα από τα πιθανά προβλήματα που μπορούν να προκύψουν και σίγουρα με αυτή τη προσθήκη θα αυξάναμε την διάρκεια ζωής της συσκευής μας.

Επιπλέον αναβάθμιση της κατασκευής είναι και η δημιουργία-προσθήκη πολλαπλών χρηστών. Στην εφαρμογή μας μέσω του κώδικα προσθέτουμε έναν αριθμό τηλεφώνου στον οποίο γίνεται η αποστολή των μηνυμάτων, αυτός ο αριθμός παραμένει σταθερός, κάτι που αντιλαμβανόμαστε ότι μπορεί να δημιουργήσει προβλήματα. Για να μην χρειάζεται να αλλάζουμε συνεχώς την μεταβλητή στον κώδικα θα ήταν θετικό να υπάρχει δυνατότητα να αποθηκεύει έναν αριθμό που θα καταγράφει στο σύστημα ο χρήστης και το πλήθος των αριθμών να είναι περισσότερα από ένα. Να μπορεί δηλαδή να επικοινωνεί και να ελέγχει την κατασκευή παραπάνω από ένας χρήστης. Είτε μέσω SMS είτε μέσω δημιουργίας πολλαπλών χρηστών σε κάποια εφαρμογή.

Στην εφαρμογή μας κάναμε χρήση μίας και μόνο συσκευής, η οποία κάνει τους ελέγχους και επικοινωνεί με τον χρήστη. Για να αναπτύξουμε ακόμη περισσότερο την εφαρμογή μας θα μπορούσαμε να δημιουργήσουμε ένα δίκτυο το οποίο θα κάνει διαφορετικούς ελέγχους ξεχωριστά και να στέλνει τις πληροφορίες στο χρήστη, αλλά να είναι επίσης και δυνατή η επικοινωνία μεταξύ των συσκευών. Έτσι μπορεί να δημιουργηθεί ένα πλήρες σύστημα ελέγχου στα πρότυπα Internet Of Things (IOT) εφαρμογών.

Ακόμα μια σημαντική αναβάθμιση που θα βοηθήσει στη βελτίωση της λειτουργίας της κατασκευής είναι η προσθήκη μπαταρίας στο σύστημα. Η επιλογή που έγινε για τροφοδοσία της πλακέτας μέσω εξωτερικής τροφοδοσίας έγινε βασισμένη στην οικονομία όσον αφορά την αλλαγή της μπαταρίας αλλά και την πιθανότητα πλήρης αποφόρτισης αυτής που θα οδηγούσε στην δυσκολία επικοινωνίας με το χρήστη. Μια αναβάθμιση στον τομέα αυτό θα ήταν ο συνδυασμός τροφοδοσίας της κατασκευής με την εξωτερική τροφοδοσία μέσω του ρεύματος αλλά και μέσω μπαταρίας. Πρακτικά αυτό θα οδηγούσε σε εξάλειψη προβλημάτων επικοινωνίας που πιθανό να προέκυπταν από την έλλειψη τροφοδοσίας. Αυτό διότι η συσκευή θα ήταν μόνιμα συνδεδεμένη με το ρεύμα μέσω της εξωτερικής τροφοδοσίας και η χρήση της μπαταρίας θα γινόταν αποκλειστικά και μόνο σε περίπτωση

διακοπής του ρεύματος. Η μπαταρία θα ήταν επαναφορτιζόμενη και με την επιστροφή του ρεύματος θα φόρτιζε συνεχώς. Επομένως το πρόβλημα της επικοινωνίας δεν θα απασχολούσε τον χρήστη.

9.5 Τομείς εφαρμογής της κατασκευής

Υπάρχουν πολλοί τομείς στους οποίους μπορεί κανείς να κάνει χρήση της συγκεκριμένης συσκευής ελέγχου. Μία πιθανή χρήση θα μπορούσε να είναι ως κάποιο σύστημα ασφαλείας. Με τις απαραίτητες προσθήκες θα μπορούσε ο χρήστης να ενημερώνεται για την κατάσταση στην οποία βρίσκεται για παράδειγμα μια πόρτα ή ακόμα και να μεταβάλλει την κατάσταση αυτής. Στέλνοντας επομένως το αντίστοιχο μήνυμα θα μπορούσε να μάθει αν μια πόρτα είναι ξεκλειδωτή, κλειδωμένη, αν ξεκλειδώσε ή κλειδώσε αντίστοιχα αλλά και να ξεκλειδώσει η να κλειδώσει την πόρτα ο ίδιος.

Άλλη χρήση θα μπορούσε να είναι ο έλεγχος μιας συσκευής όπως για παράδειγμα το καλοριφέρ ενός σπιτιού. Με τις αντίστοιχες προσθήκες και την ίδια διαδικασία που προαναφέρθηκε ο χρήστης μπορεί να ενεργοποιήσει η να απενεργοποιήσει τη θέρμανση, να μάθει τη θερμοκρασία του χώρου ή να μάθει την θερμοκρασία που έχει οριστεί για να ενεργοποιηθεί ή να απενεργοποιηθεί η θέρμανση. Χρησιμότητα θα μπορούσε να έχει η συγκεκριμένη κατασκευή ακόμα και στη γεωργία. Ο χρήστης θα μπορούσε να ελέγχει την κατάσταση που βρίσκεται ένα ποτιστήρι για παράδειγμα. Να ενεργοποιήσει η να απενεργοποιήσει τη λειτουργία του με την αποστολή του αντιστοίχου SMS.

Τέλος η χρησιμότητα του σε ένα έξυπνο σπίτι είναι τεράστια ειδικά αν αναβαθμιστεί η επικοινωνία με τη χρήση WiFi ή 5G. Το Internet of things είναι πλέον στη ζωή μας και ο απομακρυσμένος έλεγχος αρχίζει να γίνεται καθημερινή ανάγκη του καθενός. Ο έλεγχος της κάθε συσκευής μέσα σε ένα σπίτι απομακρυσμένα θα μπορούσε να εξυπηρετηθεί από την συγκεκριμένη συσκευή. Από το άναμμα και σβήσιμο μιας λάμπας έως τον έλεγχο της θερμοκρασίας του σπιτιού. Πάντα με τις σωστές προσθήκες και αναβαθμίσεις της κατασκευής.

9.6 Προβλήματα κατασκευής του συστήματος.

Κατά την κατασκευή του συστήματος ελέγχου αντιμετωπίσαμε μερικά προβλήματα. Αρχικά παρουσιάστηκε θέμα με την εύρεση σήματος του GSM module. Απαραίτητη ήταν η προσθήκη του DC-DC converter προκειμένου το GSM να τροφοδοτείται σωστά ώστε να μπορεί και να επικοινωνήσει το κυψελοειδές σύστημα. Επίσης, η παρουσία ράστερ στη κατασκευή όσον αφορά την τελική της μορφή θα ήταν άσχημη και καθόλου πρακτική, επομένως η παρουσία και συγκόλληση όλων των εξαρτημάτων σε μια διατηρητή πλακέτα ήταν αναγκαία. Ωστόσο βασικό εργαλείο για να γίνει συγκόλληση είναι το κολλητήρι. Δεν είχαμε διαθέσιμο κάποιο κολλητήρι, από τα εργαστήρια της σχολής δεν μπορούσαμε να εξυπηρετηθούμε επομένως αναγκαστήκαμε και αγοράσαμε ότι χρειαζόταν. Γενικότερα τα προβλήματα που αντιμετωπίσαμε δεν είναι άξια αναφοράς και σε γενικές γραμμές η κατασκευή ολοκληρώθηκε επιτυχώς.

9.7 Επίλογος

Στο κεφάλαιο που ακολούθησε έγινε αναφορά σε κάποια γενικά συμπεράσματα που προέκυψαν κατά τη διάρκεια της συγγραφής και κατασκευής της εργασίας, τα πιθανά προβλήματα που μπορεί να κληθεί να αντιμετωπίσει ο χρήστης κατά τη χρήση της κατασκευής και παρουσιάστηκαν οι προτάσεις βελτίωσης και αναβάθμισης του συγκεκριμένου συστήματος ελέγχου. Επίσης έγινε αναφορά στους τομείς όπου μπορεί να εφαρμοστεί η συσκευή με την προσθήκη ή και όχι μερικών ακόμα

απαραίτητων εξαρτημάτων και τέλος παρουσιάστηκαν κάποια προβλήματα που αντιμετωπίσαμε κατά τη δημιουργία και κατασκευή του συστήματος ελέγχου.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- 1) https://lastminuteengineers.com/sim800l-gsm-module-arduino-tutorial/?fbclid=IwAR1Ffh1XH1H0CLR4X88qrAxkNmUWo0W19jLrF89bAh_cDMnntLEa_hbBWfuU
- 2) <http://apothesis.teicm.gr/xmlui/bitstream/handle/123456789/2426/Georgiadis.pdf?sequence=1&isAllowed=y&fbclid=IwAR23gTj5eb3VEjjkGqGF9nqQPvluIWnN25JVFz8RrU918CAdX6TXCAg1PNM>
- 3) <https://edurobotics.gr/what-is-arduino/?fbclid=IwAR0X4jPBE6EURPg9vfAafN5wfDyafV-IF1hEwZduyd2KTGfRxSpdo14hYYM>
- 4) https://www.arduino.cc/en/Main/arduinoBoardMega2560/?fbclid=IwAR1k6gbgYU7gW9Uyv9VGDh5PngdJQB9cXr_IBe1NxA4cypuFwMyL7rpQTI
- 5) Κινητές Επικοινωνίες - <http://teachers.cm.ihu.gr/papatsoris/Mobile%20Communications.pdf>
- 6) Κινητά Δίκτυα, 2008 – <https://eclass.upatras.gr/modules/document/file.php/CEID1024/KINHTA%20DIKTYA%20%282008%29.pdf>
- 7) <http://www.uky.edu/~jclark/mas355/GSM.PDF>
- 8) https://nettigo.eu/products/sim800l-gsm-grps-module?fbclid=IwAR26JjAqTTiPi2GMYeOiOlnsBdmVy49WMCx2J9zPCercFnhlZO_9urNVBmo
- 9) http://www.cs.nccu.edu.tw/~jang/teaching/MobileSystem_files/Global%20System%20for%20Mobile%20Communications.pdf
- 10) <http://index-of.es/Varios-2/Programming%20Arduino.pdf>
- 11) Ανάπτυξη Εφαρμογών με το Arduino – Ένας πλήρης οδηγός για το Arduino & ένα χρήσιμο εργαλείο για την ανάπτυξη εφαρμογών STEM, 2^η έκδοση – Παναγιώτης Παπάζογλου, Σπύρος – Πολυχρόνης Λιωνής ΕΚΔΟΣΕΙΣ ΤΖΙΟΛΑ 2018, ISBN 9789604185504

ΠΑΡΑΡΤΗΜΑ Α : ΚΩΔΙΚΑΣ ΛΕΙΤΟΥΡΓΙΑΣ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

```
#include <stdio.h>
#include <stdlib.h>
#include <SoftwareSerial.h>
SoftwareSerial smsSerial(12,13);
String number ="6980959782";
```

```
const int button = 50;
const int led =2;
int ledflag=0;
const int button1 = 42;
const int led1 =3;
int ledflag1=0;
const int button2 = 34;
const int led2 =7;
int ledflag2=0;
const int button3 = 26;
const int led3 =8;
int ledflag3=0;
const int analog1 = A9;
const int analogOut1 = 4;
int sensorValue1 = 0;
int outputValue1 = 0;
```

```
const int analog2 = A3;
const int analogOut2 = 5;
int sensorValue2 = 0;
int outputValue2 = 0;
const int analog3 = A13;
const int analogOut3 = 6 ;
int sensorValue3 = 0;
```

```

int outputValue3 = 0;
const int analog4 = A0;
const int analogOut4 = 9;
int sensorValue4 = 0;
int outputValue4 = 0;
String TextForSms ;
char input_string[15];
int var_1 = 0;
int var_2 = 0;
int snum = 0;
int anum = 0;

void setup() {
  pinMode(button,INPUT);
  pinMode(led,OUTPUT);
  digitalWrite(led,LOW);
  pinMode(button1,INPUT);
  pinMode(led1,OUTPUT);
  digitalWrite(led1,LOW);
  pinMode(button2,INPUT);
  pinMode(led2,OUTPUT);
  digitalWrite(led2,LOW);
  pinMode(button3,INPUT);
  pinMode(led3,OUTPUT);
  digitalWrite(led3,LOW);
  Serial.begin(9600);
  smsSerial.begin(9600);
  smsSerial.println("AT+CMGD=1,4"); // delete all SMS
  delay(5000);
  smsSerial.println("AT+CMGF=1");
  delay(500);
  smsSerial.println("AT+CNMI=1,2,0,0,0");
  delay(500);

```

```
}
```

```
void loop(){
```

```
    readbutton();
```

```
    analogin();
```

```
    while (smsSerial.available()){
```

```
        char input_char = smsSerial.read();
```

```
        input_string[var_2++] = input_char;
```

```
        Serial.println(input_char);
```

```
        if (input_char == '$')
```

```
        {
```

```
            char input_char = smsSerial.read();
```

```
            input_string[var_2++] = input_char;
```

```
            if (input_char == '1')
```

```
            {
```

```
                snum = 1;
```

```
                var_1 = 1;
```

```
            }
```

```
            if (input_char == '2')
```

```
            {
```

```
                snum = 2;
```

```
                var_1 = 1;
```

```
            }
```

```
            if (input_char == '3')
```

```
            {
```

```
                snum = 3;
```

```
                var_1 = 1;
```

```

    }
    if (input_char == '4')
    {
        snum = 4;
        var_1 = 1;
    }
    if (input_char != '1' && input_char != '2' && input_char != '3' && input_char != '4')
{
    smsSerial.println("AT+CMGF=1");
    delay(500);
    smsSerial.println("AT+CMGS=\"+30"+number+"\"");
    delay(500);
    smsSerial.print("Wrong Message! ");
    delay(500);
    smsSerial.write(26);
    delay(100);
    res();
}
}

if (input_char == '#')
{
    char input_char = smsSerial.read();
    input_string[var_2++] = input_char;
    if (input_char == '1')
    {
        anum = 1;
    }
    if (input_char == '2')
    {
        anum = 2;
    }
}

```

```

    if (input_char == '3')
    {
        anum = 3;
    }
    if (input_char == '4')
    {
        anum = 4;
    }
    if (input_char != '1' && input_char != '2' && input_char != '3' && input_char != '4')
{
    smsSerial.println("AT+CMGF=1");
    delay(500);
    smsSerial.println("AT+CMGS=\"+30"+number+"\"");
    delay(500);
    smsSerial.print("Wrong Message! ");
    delay(500);
    smsSerial.write(26);
    delay(100);
    res();

}
    analogin();
}

if (var_1 == 1)
{
    readbutton();
    var_1 = 0;
    var_2 = 0;
    snum = 0;
    delay(5);
}

```

```

    delay(5);
}

}

void readbutton() {

if (digitalRead(button)==HIGH || snum == 1){
    if (ledflag==0) {
        ledflag=1;
        digitalWrite(led,HIGH);
        smsSerial.println("AT+CMGF=1"); // Configuring TEXT mode
        delay(500);
        smsSerial.println("AT+CMGS=\"+30"+number+"");
        delay(500);
        smsSerial.print("LED1 ON"); //text content
        delay(500);
        smsSerial.write(26);
        delay(100);
        res();
    }
else {
    ledflag=0;
    digitalWrite(led,LOW);
    smsSerial.println("AT+CMGF=1"); // Configuring TEXT mode
    delay(500);
    smsSerial.println("AT+CMGS=\"+30"+number+"");
    delay(500);
    smsSerial.print("LED1 OFF"); //text content
    delay(500);
    smsSerial.write(26);
    delay(100);
    res();
}
}

```

```

}
}
if (digitalRead(button1)==HIGH || snum == 2){
  if (ledflag1==0) {
    ledflag1=1;
    digitalWrite(led1,HIGH);
    smsSerial.println("AT+CMGF=1"); // Configuring TEXT mode
    delay(500);
    smsSerial.println("AT+CMGS=\"+30"+number+"\"");
    delay(500);
    smsSerial.print("LED2 ON"); //text content
    delay(500);
    smsSerial.write(26);
    delay(100);
    res();
  }
else {
  ledflag1=0;
  digitalWrite(led1,LOW);
  smsSerial.println("AT+CMGF=1"); // Configuring TEXT mode
  delay(500);
  smsSerial.println("AT+CMGS=\"+30"+number+"\"");
  delay(500);
  smsSerial.print("LED2 OFF"); //text content
  delay(500);
  smsSerial.write(26);
  delay(100);
  res();
}
}
if (digitalRead(button2)==HIGH || snum == 3){
  if (ledflag2==0) {
    ledflag2=1;

```

```

digitalWrite(led2,HIGH);
  smsSerial.println("AT+CMGF=1"); // Configuring TEXT mode
  delay(500);
  smsSerial.println("AT+CMGS=\"+30"+number+"\"");
  delay(500);
  smsSerial.print("LED3 ON"); //text content
  delay(500);
  smsSerial.write(26);
  delay(100);
  res();
}
else {
  ledflag2=0;
  digitalWrite(led2,LOW);
  smsSerial.println("AT+CMGF=1"); // Configuring TEXT mode
  delay(500);
  smsSerial.println("AT+CMGS=\"+30"+number+"\"");
  delay(500);
  smsSerial.print("LED3 OFF"); //text content
  delay(500);
  smsSerial.write(26);
  delay(100);
  res();
}
}
if (digitalRead(button3)==HIGH || snum == 4){
if (ledflag3==0) {
  ledflag3=1;
  digitalWrite(led3,HIGH);
  smsSerial.println("AT+CMGF=1"); // Configuring TEXT mode
  delay(500);
  smsSerial.println("AT+CMGS=\"+30"+number+"\"");
  delay(500);

```

```

    smsSerial.print("LED4 ON"); //text content
    delay(500);
    smsSerial.write(26);
    delay(100);
    res();
}
else {
    ledflag3=0;
    digitalWrite(led3,LOW);
    smsSerial.println("AT+CMGF=1"); // Configuring TEXT mode
    delay(500);
    smsSerial.println("AT+CMGS=\"" +30"+number+"\"");
    delay(500);
    smsSerial.print("LED4 OFF"); //text content
    delay(500);
    smsSerial.write(26);
    delay(100);
    res();
}
}
}

```

```

void analogin()
{
    sensorValue1 = analogRead(analog1);
    outputValue1 = map(sensorValue1, 0, 1023, 0, 255);
    analogWrite(analogOut1, outputValue1);

    sensorValue2 = analogRead(analog2);
    outputValue2 = map(sensorValue2, 0, 1023, 0, 255);
    analogWrite(analogOut2, outputValue2);

    sensorValue3 = analogRead(analog3);

```

```
outputValue3 = map(sensorValue3, 0, 1023, 0, 255);
analogWrite(analogOut3, outputValue3);
```

```
sensorValue4 = analogRead(analog4);
outputValue4 = map(sensorValue4, 0, 1023, 0, 255);
analogWrite(analogOut4, outputValue4);
```

```
if (anum == 1){
  TextForSms = TextForSms + "Sen1: ";
  TextForSms.concat(sensorValue1);
  smsSerial.println("AT+CMGF=1");
  delay(500);
  smsSerial.println("AT+CMGS=\"+30"+number+"\"");
  delay(500);
  smsSerial.print(TextForSms);
  delay(500);
  smsSerial.write(26);
  delay(100);
  TextForSms = "";
  anum = 0;
  res();
}
if (anum == 2){
  TextForSms = TextForSms + "Sen2: ";
  TextForSms.concat(sensorValue2);
  smsSerial.println("AT+CMGF=1");
  delay(500);
  smsSerial.println("AT+CMGS=\"+30"+number+"\"");
  delay(500);
  smsSerial.print(TextForSms);
  delay(500);
  smsSerial.write(26);
  delay(100);
```

```

TextForSms = "";
anum = 0;
res();
}
if (anum == 3){
TextForSms = TextForSms + "Sen3: ";
TextForSms.concat(sensorValue3);
smsSerial.println("AT+CMGF=1");
delay(500);
smsSerial.println("AT+CMGS=\"+30"+number+"\"");
delay(500);
smsSerial.print(TextForSms);
delay(500);
smsSerial.write(26);
delay(100);
TextForSms = "";
anum = 0;
res();
}
if (anum == 4){
TextForSms = TextForSms + "Sen4: ";
TextForSms.concat(sensorValue4);
smsSerial.println("AT+CMGF=1");
delay(500);
smsSerial.println("AT+CMGS=\"+30"+number+"\"");
delay(500);
smsSerial.print(TextForSms);
delay(500);
smsSerial.write(26);
delay(100);
TextForSms = "";
anum = 0;
res();
}

```

```

    }
}

void res()
{
  smsSerial.println("AT+CMGF=1");
  delay(500);
  smsSerial.println("AT+CNMI=1,2,0,0,0");
  delay(500);
}

```

ΠΑΡΑΡΤΗΜΑ Β : DATASHEETS SIM800L

https://img.filipeflop.com/files/download/Datasheet_SIM800L.pdf

- Supply voltage: 3.8V - 4.2V
- Recommended supply voltage: 4V
- Power consumption:
 - sleep mode < 2.0mA
 - idle mode < 7.0mA
 - GSM transmission (avg): 350 mA
 - GSM transmission (peek): 2000mA
- Module size: 25 x 23 mm
- Interface: UART (max. 2.8V) and AT commands
- SIM card socket: microSIM (bottom side)
- Supported frequencies: Quad Band (850 / 950 / 1800 /1900 MHz)
- Antenna connector: IPX
- Status signaling: LED
- Working temperature range: -40 do + 85 ° C

ΠΑΡΑΡΤΗΜΑ Γ : DATASHEETS ARDUINO MEGA2560

<https://www.mantech.co.za/datasheets/products/A000047.pdf>

- Microcontroller ATmega2560
- Operating Voltage 5V
- Input Voltage (recommended) 7-12V
- Input Voltage (limits) 6-20V
- Digital I/O Pins 54 (of which 14 provide PWM output)
- Analog Input Pins 16
- DC Current per I/O Pin 40 mA
- DC Current for 3.3V
- Pin 50 mA

- Flash Memory 256 KB of which 8 KB used by bootloader
- SRAM 8 KB
- EEPROM 4 KB
- Clock Speed 16 MHz

ΠΑΡΑΡΤΗΜΑ Δ : ΜΕΤΑΤΡΟΠΕΑΣ DC-DC

<https://www.alldatasheet.com/datasheet-pdf/pdf/763181/ETC2/XL4005.html>