Σχολή τεχνολογικών εφαρμογών

Τμήμα αυτοματισμού

Πτυχιακή εργασία

Θέμα:

" ΈΛΕΓΧΟΣ ΧΩΡΟΥ ΜΕΣΟ ΔΙΚΤΥΟΥ GSM "

Επιβλέπον καθηγητής: Μιχάλης Παπουτσιδάκης

Φοιτητές :

Νικολαΐδης Μάρκος Α.Μ.: 38691

Παρπας Μάριος Α.Μ.: 38109

Κονδύλης Στέλιος Α.Μ.: 38672

Αιγάλεω, Νοέμβριος 2013

Περιεχόμενα

Κεφάλαιο 1

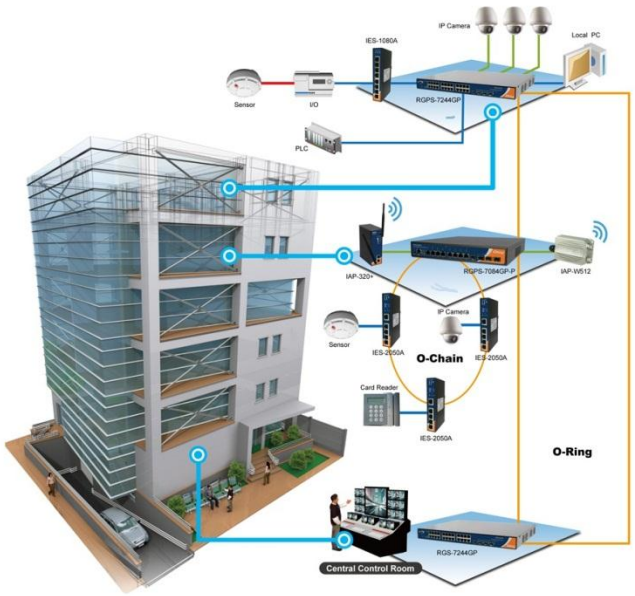
Εισαγωγή

Ο Κτιριακός αυτοματισμός

περιγράφει την προηγμένη λειτουργικότητα που παρέχεται από το σύστημα ελέγχου ενός κτιρίου.

΄Ενα σύστημα κτιριακού αυτοματισμού είναι παράδειγμα ενός κατανεμημένου συστήματος ελέγχου. Αυτό το σύστημα είναι ένα αυτοματοποιημένο έξυπνο δίκτυο που αποτελείται από ηλεκτρονικές, ηλεκτρικές συσκευές και αισθητήρια που έχουν σχεδιαστεί για την παρακολούθηση και τον έλεγχο διαφόρων παραμέτρων σε ένα κτίριο.

Οι βασικές λειτουργίες ενός B.A.S. (Building Automation System) ή B.M.S (Building Management System) στην αγορά είναι να κρατά το κλίμα ενός κτιρίου σε ένα προκαθορισμένο εύρος, να παρέχει φωτισμό βασισμένο σε ένα χρονοδιάγραμμα, να παρακολουθεί τις αποδόσεις και αποτυχίες διαφόρων λειτουργιών και να ενημερώνει για τις λειτουργίες μέσω email και μηνυμάτων στο αρμόδιο προσωπικό.



Σύστημα Διαχείρισης Κτιρίου

Η λειτουργία ενός τέτοιου συστήματος μειώνει το ενεργειακό κόστος και το κόστος συντήρησης σε σύγκριση με ένα κτίριο χωρίς αυτόματο έλεγχο.

Ένα κτίριο που χρησιμοποιεί αυτόματο σύστημα ονομάζεται επίσης έξυπνο κτίριο ή έξυπνο σπίτι.

Ο Οικιακός αυτοματισμός :

είναι η οικιακή επέκταση του κτιριακού αυτοματισμού.

Η διάδοση του οικιακού αυτοματισμού έχει εξαπλωθεί εκθετικά τα τελευταία χρόνια χάρη στη μείωση κόστους διαφόρων συσκευών απαραίτητες για την δημιουργία ενός τέτοιου συστήματος. Οι κύριες λειτουργίες ενός τέτοιου οικιακού συστήματος είναι παρόμοιες με αυτές του κτιριακού αυτοματισμού συν πολλές άλλες εφαρμογές όπως για παράδειγμα το σύστημα διασκέδασης, η συντήρηση κήπου, η συντήρηση κατοικίδιων.

http://www.smarthouse.sg/

Αυτοματοποιημένο Σπίτι μέσω smartphone

Ο στόχος:

Ο στόχος της παρούσας μελέτης/κατασκευής είναι η κατασκευή ενός συστήματος παρόμοιο με αυτού ενός κτιριακού/οικιακού αυτοματισμού, με όσο το δυνατό χαμηλότερο κόστος.

Συγκεκριμένα, ένα έξυπνο σύστημα όπου θα εκτελεί προκαθορισμένες λειτουργίες σε ένα χώρο με ιδιαίτερο χαρακτηριστικό το γεγονός ότι θα έχουμε τη δυνατότητα να αλληλεπιδράμε με το σύστημα αυτό μέσω του δικτύου GSM, δηλαδή με ένα κινητό τηλέφωνο θα έχουμε την δυνατότητα να λαμβάνουμε πληροφορίες, να αλλάζουμε παραμέτρους και να ενημερωνόμαστε για δυσλειτουργίες και προβλήματα.

Οι λειτουργίες ενός τέτοιου συστήματος είναι δύσκολο να οριοθετηθούν αφού οι πρακτικές εφαρμογές περιορίζονται στη φαντασία κάθε χρήστη σύμφωνα με της ανάγκες του, από τον έλεγχο μιας καφετιέρας μέχρι το πότισμα του κήπου βάση ενός χρονοδιαγράμματος.

Για την επίδειξη των δυνατοτήτων αυτού του συστήματος αποφασίζουμε να χρησιμοποιήσουμε ένα αισθητήριο θερμοκρασίας ,έναν ηλεκτρονόμο, έναν κινητήρα συνεχούς τάσης, ένα servomotor και τέλος ένα αισθητήριο κίνησης PIR στο ρόλο της προστασίας.

Δίκτυο *GSM*

GSM (Global System for Mobile Communications)

Το GSM είναι μια τυποποιημένη διάταξη που αναπτύχθηκε από το ινστιτούτο ETSI (*European Telecommunication Standard Institute*) για να ενώσει διάφορα πρωτόκολλα κινητού δικτύου δεύτερης γενιάς (*2G*) το οποίο και χρησιμοποιείται από τα κινητά τηλέφωνα.

Στη πραγματικότητα το δίκτυο *GSM* έγινε το παγκόσμιο πρότυπο της κινητής επικοινωνίας με μερίδιο στην αγορά περισσότερο από 80%.

Το δίκτυο *GSM* αναπτύχθηκε για να αντικαταστήσει το πρώτης γενιάς (1G) αναλογικό δίκτυο και αρχικά περιγράφηκε ως το ψηφιακό κύκλωμα εναλλαγής δικτύων, που βελτιώθηκε για πλήρη αμφίδρομη φωνητική επικοινωνία. Στη συνέχεια, επεκτάθηκε για να συμπεριληφθεί η μετάδοση δεδομένων, πρώτα με το κύκλωμα αμφίδρομης μεταφοράς και μετά με το κύκλωμα μεταφοράς πακέτων δεδομένων *GPRS* (*General Packet Radio Service*).

Περαιτέρω βελτιώσεις έγιναν όταν η *3GPP* ανέπτυξε το δίκτυο τρίτης γενιάς 3G και κατόπιν το δίκτυο 4G.

Κεφάλαιο 2

Η κατασκευή

Η πλακέτα Arduino Mega

Κάθε ηλεκτρονικό αυτόματο σύστημα χρειάζεται μια μονάδα έλεγχου, “εγκέφαλο”, έτσι και η συγκεκριμένη κατασκευή χρησιμοποιεί τον Arduino Mega.

Το Arduino είναι μια υπολογιστική πλατφόρμα βασισμένη σε μια απλή μητρική πλακέτα με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους, η οποία μπορεί να προγραμματιστεί με τη γλώσσα Wiring (ουσιαστικά πρόκειται για τη C++ με κάποιες μετατροπές).

Μία πλακέτα Arduino Mega αποτελείται από ένα μικροελεγκτή Atmel ATmega328 και κάποια συμπληρωματικά εξαρτήματα για τη διευκόλυνση του χρήστη στον προγραμματισμό και την ενσωμάτωση του σε άλλα κυκλώματα. Όλες οι πλακέτες περιλαμβάνουν ένα γραμμικό ρυθμιστή τάσης 5V και έναν κρυσταλλικό ταλαντωτή 16MHz. Ο μικροελεγκτής είναι από κατασκευής προγραμματισμένος με ένα boot loader, έτσι ώστε να μην χρειάζεται εξωτερικός προγραμματιστής

Πρόσοψη Arduino Mega

Το Ολοκληρωμένο Περιβάλλον Ανάπτυξης (IDE) του Arduino είναι μία εφαρμογή γραμμένη σε Java, που λειτουργεί σε πολλές πλατφόρμες, και προέρχεται από το IDE για τη γλώσσα προγραμματισμού Processing και το σχέδιο Wiring.

Περιλαμβάνει ένα πρόγραμμα επεξεργασίας κώδικα με χαρακτηριστικά όπως είναι η επισήμανση σύνταξης και ο συνδυασμός αγκύλων και είναι επίσης σε θέση να μεταγλωττίζει και να φορτώνει προγράμματα στην πλακέτα με ένα μόνο κλικ!

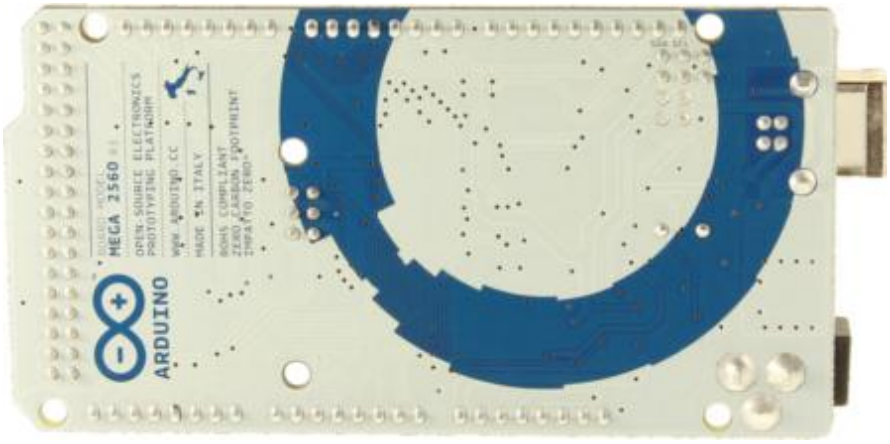
Ένα πρόγραμμα ή κώδικας που γράφτηκε για Arduino ονομάζεται σκίτσο (sketch)

Περιβάλλον Προγραμματισμού Arduino (Wiring)

Την παρούσα στιγμή υπάρχουν πολλά είδη Arduino που εξυπηρετούν διάφορες ανάγκες, ίσως ο πιο διαδεδομένος και εύκολα προσιτός Arduino είναι ο Uno. Υπάρχει επίσης η έκδοση Micro, μια Mini έκδοση για εφαρμογές, όπου το μέγεθος παίζει σημαντικό ρόλο, ο Mega που έχουμε χρησιμοποιήσει, είναι από τους μεγαλύτερους σε υπολογιστική δύναμη με μέγεθος ιδανικό καθώς μπορεί και να εξυπηρετήσει πολλές εξόδους, ομοίως με τον Due.

Η πλακέτα arduino είναι open source που σημαίνει ότι ο καθένας μπορεί να “κατεβάσει” από το διαδίκτυο τα διαγράμματα της πλακέτας και τα κατάλληλα παθητικά στοιχεία έτσι ώστε να κατασκευάσει μόνος του τον arduino.

Το περιβάλλον ανάπτυξης του arduino είναι επίσης δωρεάν και open source.



Πίσω όψη Arduino Mega

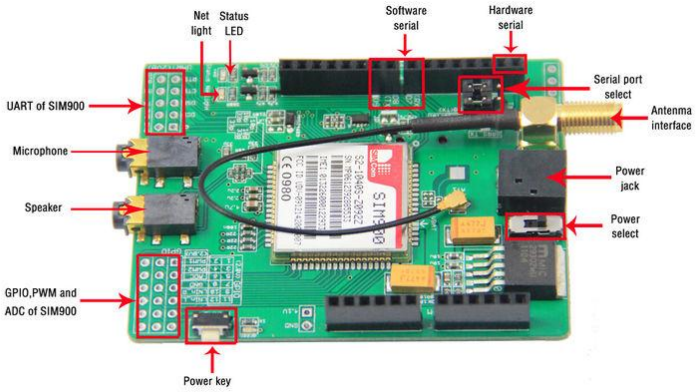
Summary

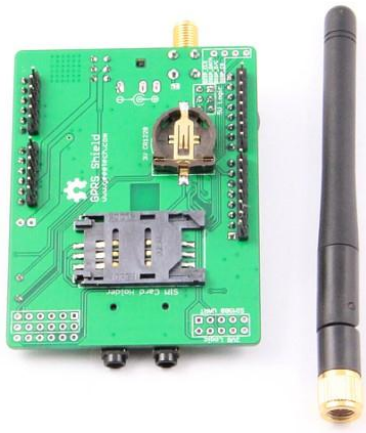
Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Τεχνικά χαρακτηριστικά Arduino Mega από σελίδα http://arduino.cc/en/Main/ArduinoBoardMega2560

Arduino GPRS Shield

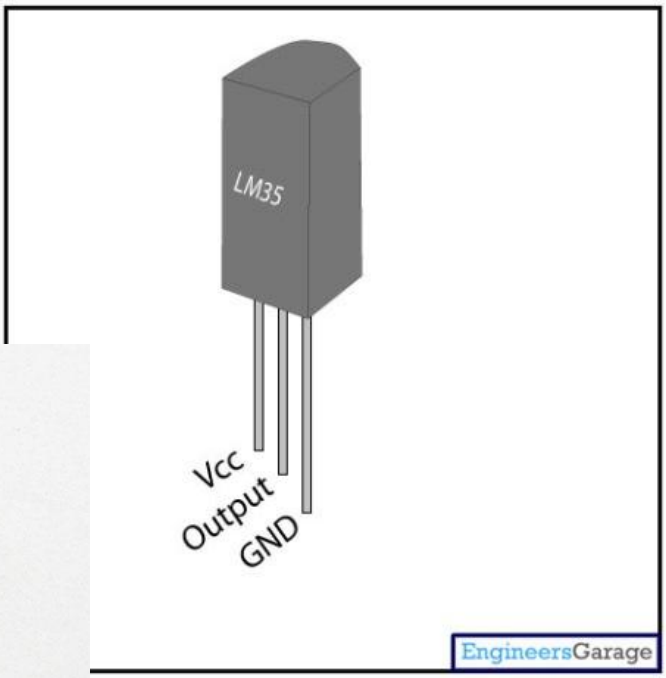
Το GPRS Shield είναι βασισμένο στο module SIM900 της εταιρίας SIMCOM και είναι συμβατό με τις περισσότερες πλακέτες Arduino. Αυτό το Shield μας παρέχει την ικανότητα να επικοινωνούμε με το δίκτυο κινητής τηλεφωνίας GSM. Αυτή η δυνατότητα μας επιτρέπει να επιτύχουμε συνδυασμό SMS (Short Message Service), MMS, GPRS και ήχο μέσω UART στέλνοντας εντολές AT.

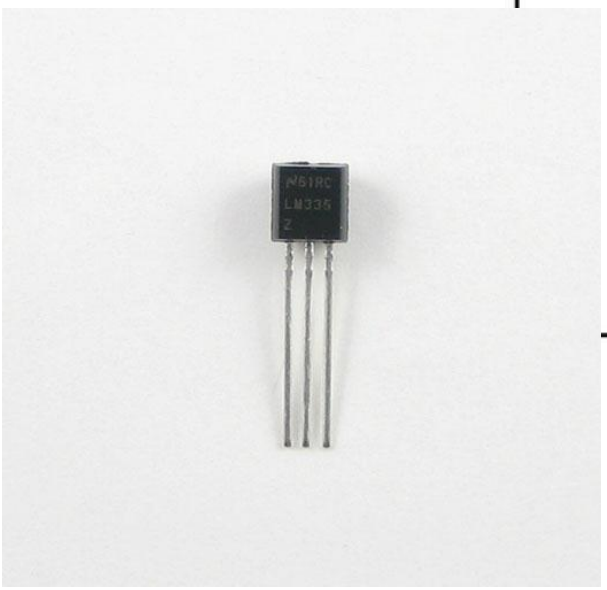




Αισθητήριο θερμοκρασίας

Το αισθητήριο LM35 είναι ένα ακριβείας ολοκληρωμένο κύκλωμα μέτρησης θερμοκρασίας με τάση εξόδου γραμμικώς ανάλογη της κλίμακας κελσίου, καθιστώντας το εύκολο στη ρύθμιση και στην εγκατάστασή του, χωρίς να χρειάζεται επιπλέον εξωτερικά παθητικά στοιχεία. Χρειάζεται μόλις 60 μA για τη λειτουργία του και εκτιμάται ότι μπορεί να λάβει μετρήσεις από -50 ως και 150 βαθμούς κελσίου.

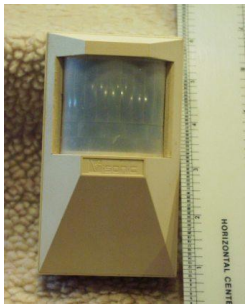




Αισθητήριο κίνησης

Όλα τα αντικείμενα με θερμοκρασία πάνω από 0 βαθμούς κελσίου εκπέμπουν θερμική ενέργεια υπό μορφή ακτινοβολίας. Συνήθως αυτή η ακτινοβολία δεν είναι ορατή στο ανθρώπινο μάτι επειδή εκπέμπεται σε υπέρυθρα μήκη κύματος. Άλλα μπορούν να ανιχνευτούν από ηλεκτρονικές συσκευές ειδικά σχεδιασμένες για αυτόν τον σκοπό.

Μια τέτοια συσκευή είναι και το αισθητήριο κίνησης PIR sensor (Passive infrared Sensor), όπου ανιχνεύει την υπέρυθρη ακτινοβολία των αντικειμένων



Τυπικό Αισθητήριο PIR

Το αισθητήριο που χρησιμοποιείται στην κατασκευή.

Κινητήρας και γέφυρα Η

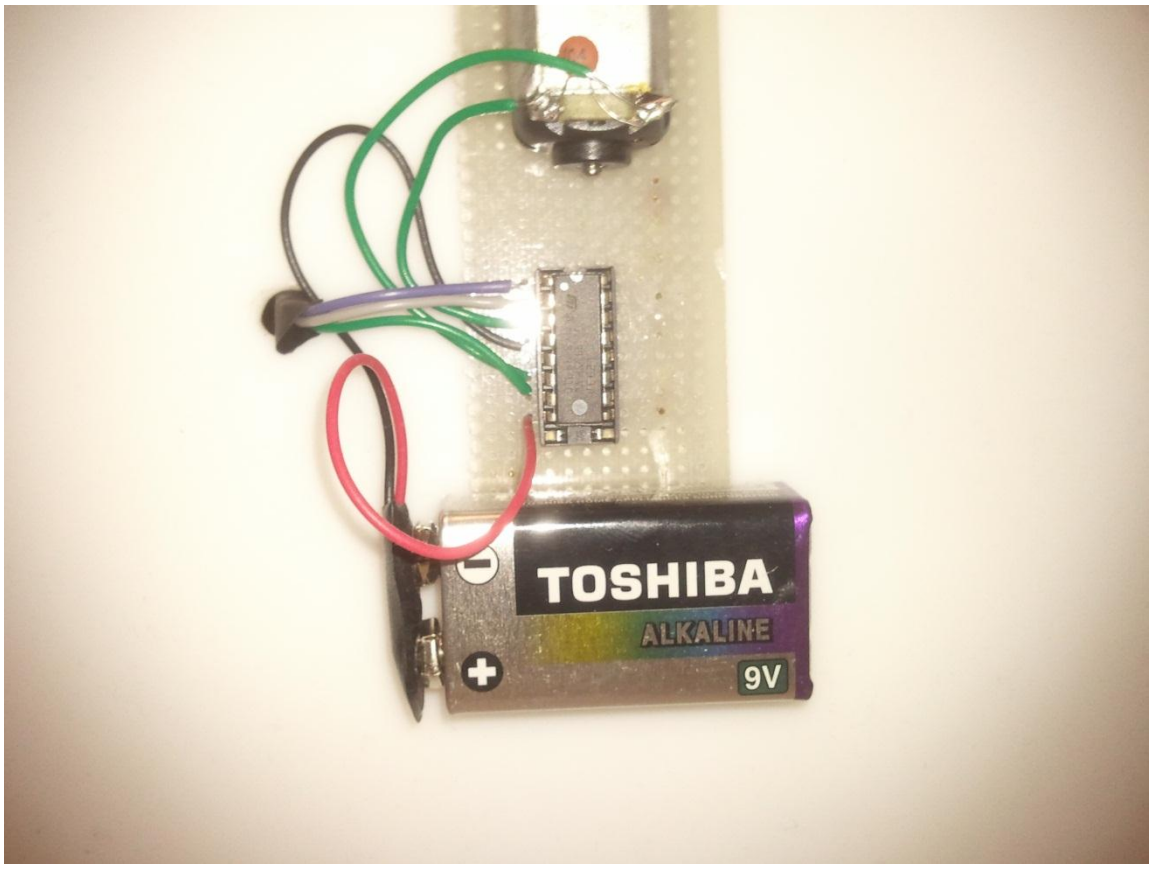
Ηλεκτροκινητήρας

Ο κινητήρας είναι ένας απλός 6V DC κινητήρας όπου προμηθευθήκαμε από ένα τηλεκατευθυνόμενο παιχνίδι.

Γέφυρα Η L293D

Μια γέφυρα Η είναι ένα ηλεκτρονικό κύκλωμα που επιτρέπει σε μια τάση που εφαρμόζεται σε ένα φορτίο να ρέει σε οποιαδήποτε κατεύθυνση επιθυμούμε. Αυτό το κύκλωμα χρησιμοποιείται σε ρομποτικές εφαρμογές και γενικά σε εφαρμογές όπου επιθυμούμε να ελέγχουμε τη φορά στροφής του κινητήρα.

Επίσης με μια γέφυρα Η μπορούμε να ελέγχουμε και την ταχύτητα του κινητήρα άλλα στη συγκεκριμένη εργασία δεν χρησιμοποιούμε αυτήν τη δυνατότητα. Το συγκεκριμένο ολοκληρωμένο αποτελείται από 16 ακροδέκτες και μπορούμε να ενσωματώσουμε πάνω του μέχρι και δυο κινητήρες.



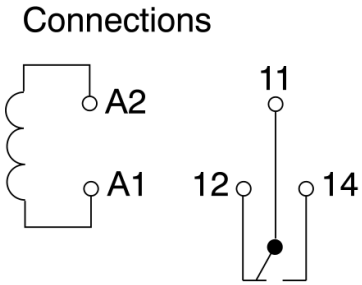
Γέφυρα Η, κινητήρας και μπαταρία

Ηλεκτρονόμος και φωτιστικό

Ένας ηλεκτρονόμος (Relay) είναι στην ουσία ένας ηλεκτρονικός διακόπτης, που χρησιμοποιεί επί το πλείστον ηλεκτρομαγνητισμό για να ενεργοποιήσει έναν μηχανισμό μεταγωγής με μηχανικό τρόπο. Οι ηλεκτρονόμοι χρησιμοποιούνται σε εφαρμογές όπου ένα ηλεκτρονικό κύκλωμα μεγάλης τάσης πρέπει να ελεγχθεί από μια μικρότερη τάση, ή όπως στη συγκεκριμένη περίπτωση, ένα σήμα πρέπει να ενεργοποιήσει ένα κύκλωμα.

Συγκεκριμένα όταν επιθυμήσουμε εμείς, ο Arduino θα στείλει ένα σήμα τάσης 5 Volts στον ηλεκτρονόμο και αυτός θα ενεργοποιηθεί για να «κλείσει» το κύκλωμα ενός φωτιστικού τάσης 240 Volts.

Ηλεκτρονόμος SCHRACK



Συνδεσμολογία Ηλεκτρονόμου

Η συνδεσμολογία του ηλεκτρονόμου επιτυγχάνεται ως εξής: Στην επαφή A1 και A2 έχουμε το σήμα των 5 Volts από τον Arduino, στους ακροδέκτες 11-12 έχουμε τη Normally Open επαφή, δηλαδή όταν δεν υπάρχει σήμα από τον Arduino τότε το κύκλωμα θα είναι «κλειστό», στους ακροδέκτες 11-14 συνδέουμε το καλώδιο του φωτιστικού έτσι ώστε, όταν έρθει σήμα στους ακροδέκτες A1-A2 θα ενεργοποιηθεί ο μηχανισμός και θα «κλείσει» το κύκλωμα.

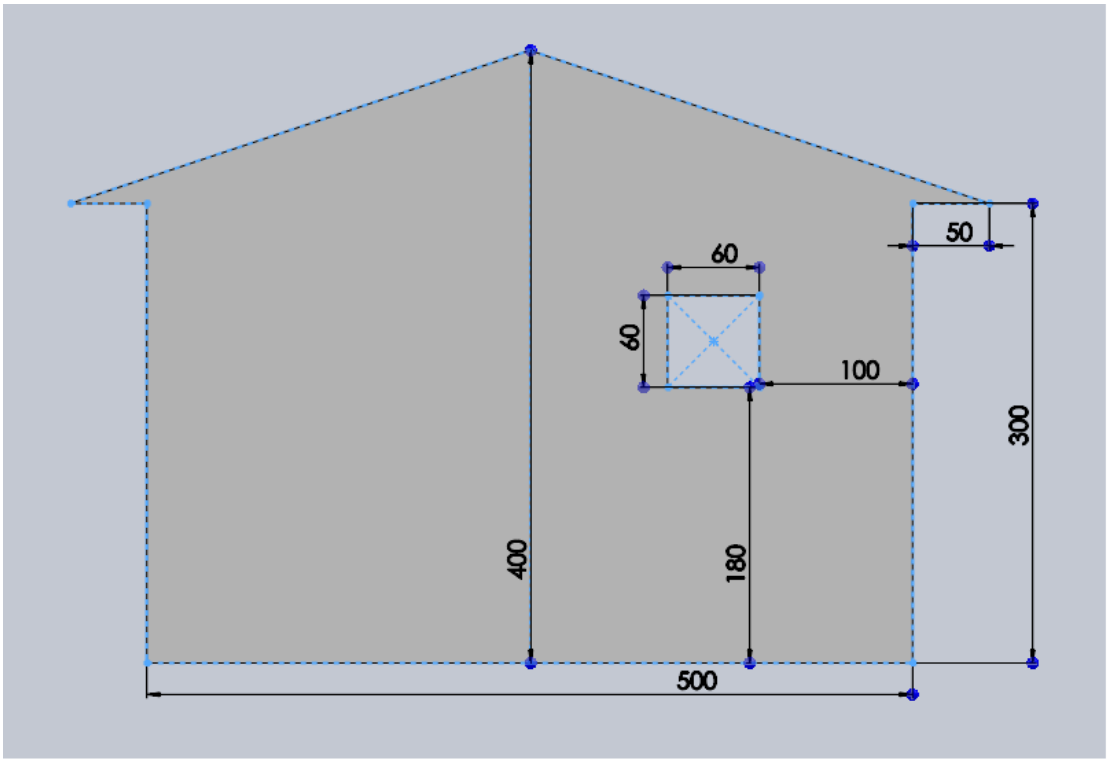
Το φωτιστικό

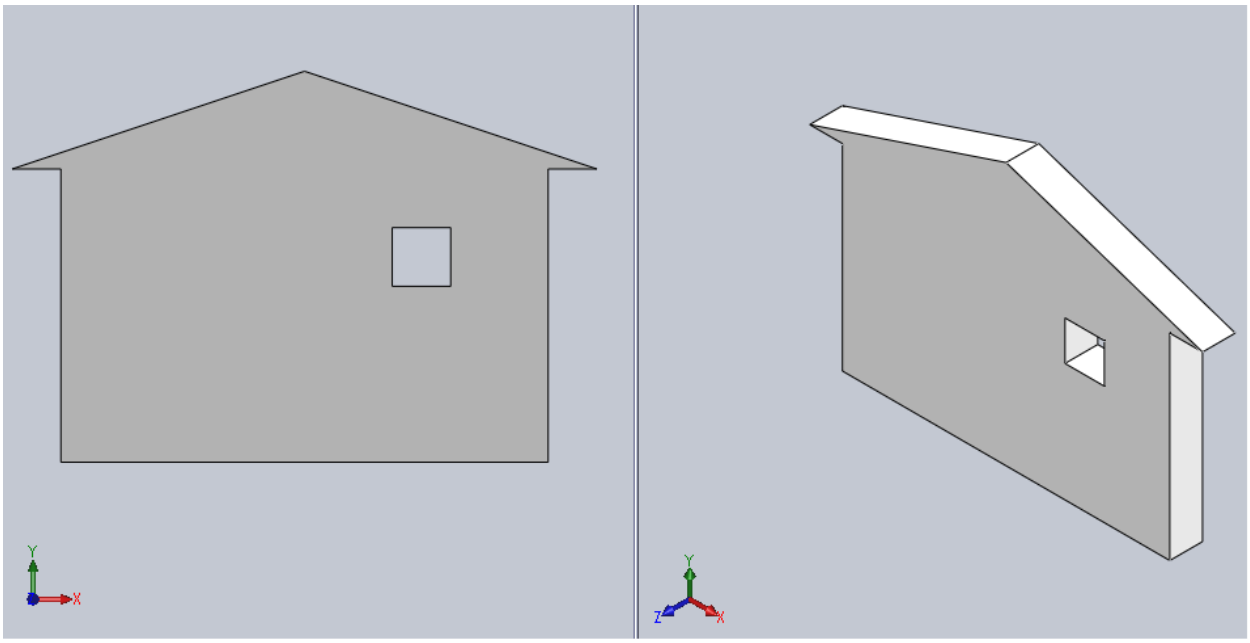
Ένα απλό οικιακό επιτραπέζιο φωτιστικό 220 Volts

Μακέτα επίδειξης

Για σκοπούς παρουσίασης, αποφασίζεται να γίνει ενσωμάτωση όλων των εξαρτημάτων της κατασκευής, πάνω σε ένα κομμάτι πλαστικό (P.V.C.). Συγκεκριμένα, ένα λευκό κομμάτι πλαστικού που σε όρθια θέση, σχηματικά, θα θυμίζει σπίτι.

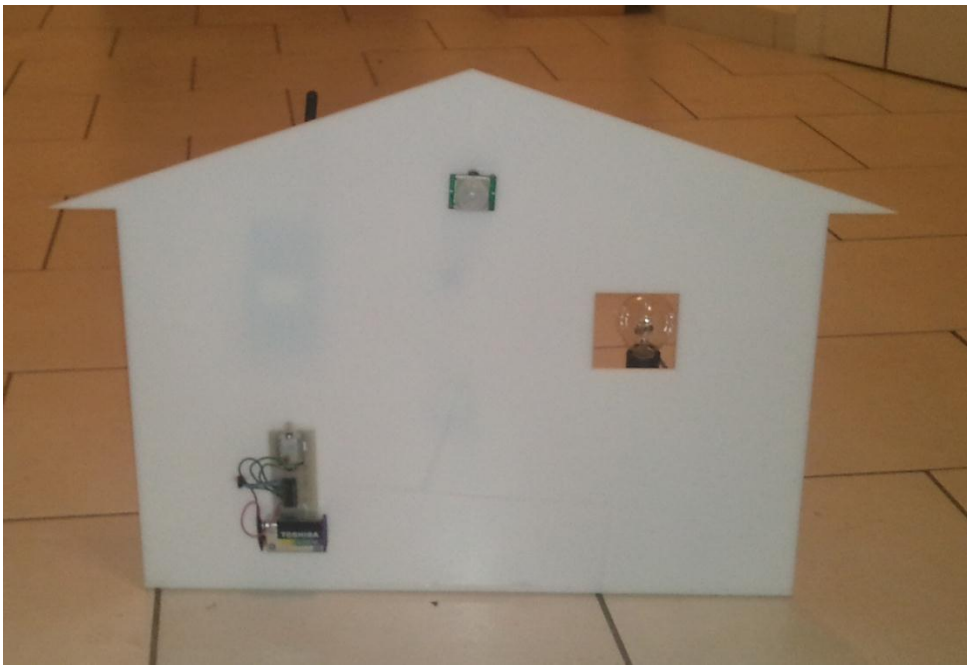
Διαστάσεις μακέτας



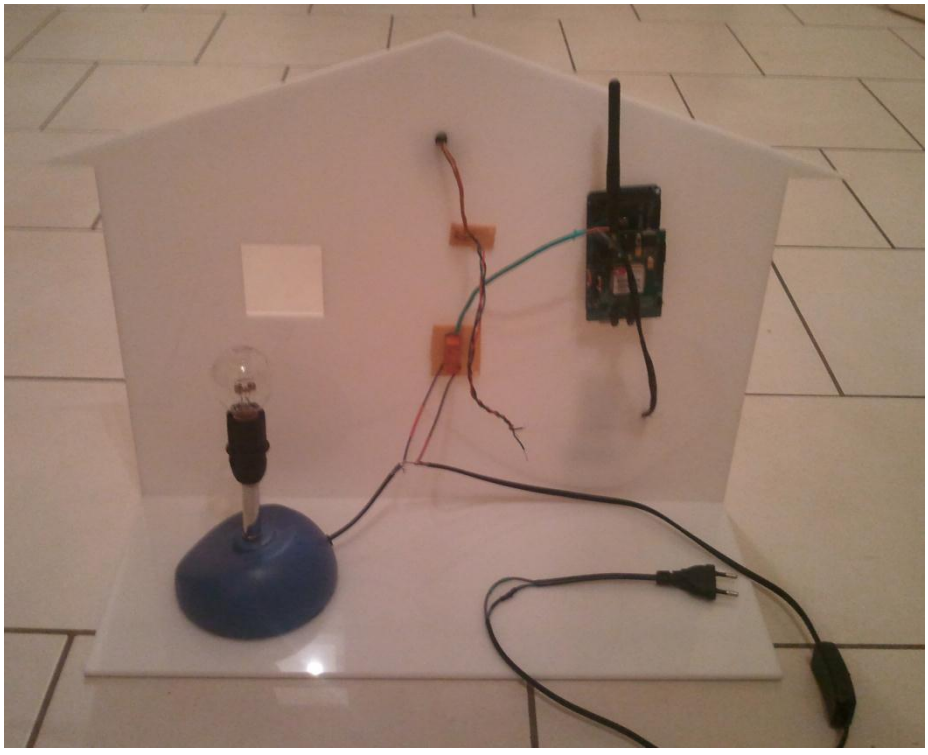


Τρισδιάστατη όψη της μακέτας στο σχεδιαστικό πρόγραμμα Solid Works.

Στην παραπάνω μακέτα τοποθετήσαμε τον Arduino Mega μαζί με το Shield, το φωτιστικό μαζί με το relay, και το θερμόμετρο στο πίσω μέρος. Στο μπροστά μέρος τοποθετήσαμε το αισθητήριο κίνησης και τον κινητήρα μαζί με τη γέφυρα Η, επίσης κόψαμε και ένα παραθυράκι διαστάσεων 6x6.



Πρόσοψη μακέτας

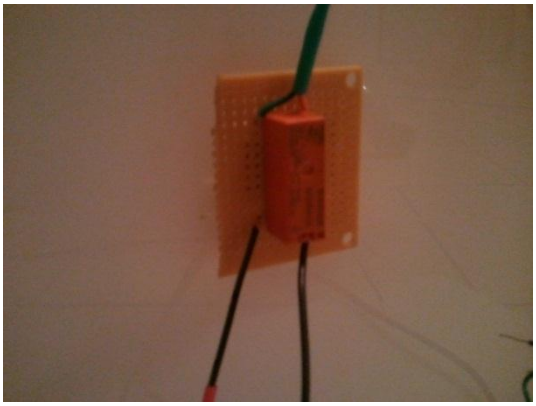


Πίσω όψη μακέτας

Συνδεσμολογία

Για την επικοινωνία του Arduino με τον υπολογιστή χρειαζόμαστε ένα καλώδιο USB τύπου (A TO B).

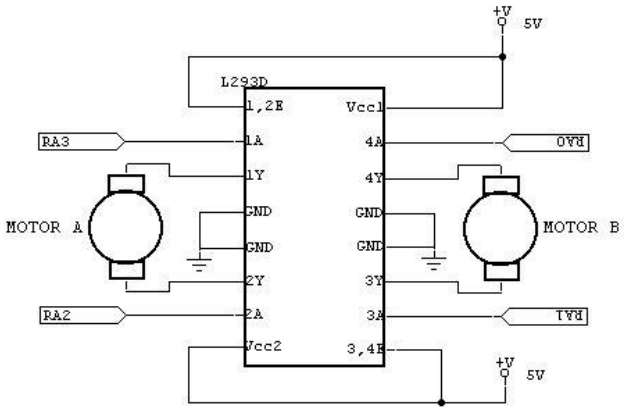
Η συσκευή (Shield) GPRS δεν χρειάζεται κάποιου είδους καλώδιο για να επικοινωνήσει με τον arduino αφού είναι απόλυτα συμβατό, δηλαδή, απλά “κουμπώνει πάνω” στον Arduino.



Συνδεσμολογία ηλεκτρονόμου

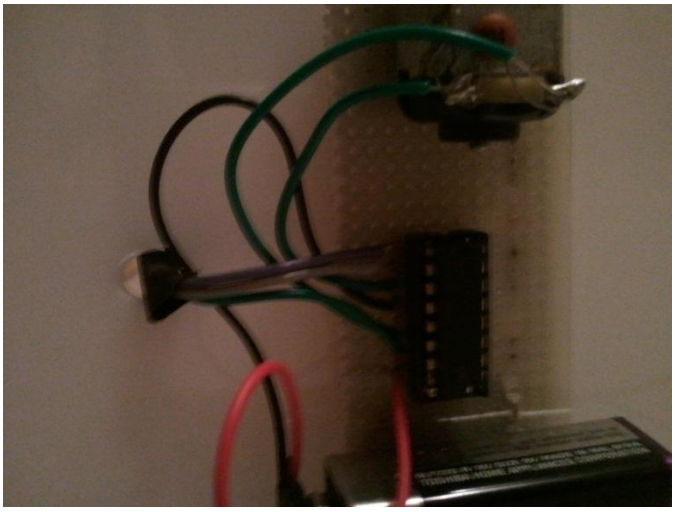
Το φωτιστικό ενώνεται με το Relay όπου και οδηγείται στον Arduino, και το καλώδιο τάσης, σε μια πρίζα. Προσεκτικά κόβουμε ένα από τα δυο καλώδια του φωτιστικού και οδηγούμε τη μια άκρη στην NO επαφή 11 και την άλλη άκρη στην επαφή 14 του ηλεκτρονόμου. Από τον Arduino παίρνουμε την ψηφιακή έξοδο 23 και πηγαίνουμε στην επαφή A1, και γειώνουμε την A2.

Κινητήρας και γέφυρα Η. Η συνδεσμολογία γίνεται σύμφωνα με την δίπλα εικόνα.



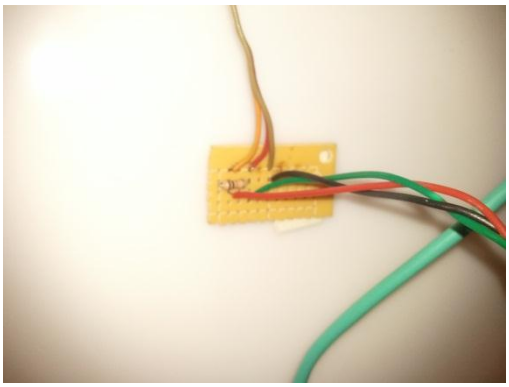
Στο πρώτο «πόδι» οδηγούμε ένα καλώδιο από τη Ψηφιακή PWM έξοδο 16 του arduino. Στα πόδια 2 και 7 οδηγούμε δυο καλώδια από της ψηφιακές εξόδους 15 και 18 του Arduino.

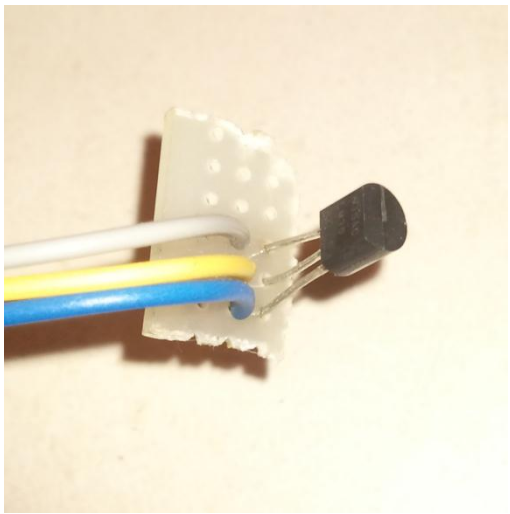
Ο κινητήρας ενώνεται με τα πόδια 3 και 6 της γέφυρας και τέλος τα πόδια 4 και 5 γειώνονται. Σύμφωνα με το παραπάνω σχεδιάγραμμα μπορούμε να τοποθετήσουμε δυο κινητήρες στο ολοκληρωμένο, αλλά όπως προαναφέρθηκε, στην κατασκευή χρησιμοποιούμε μόνο έναν, την αριστερή πλευρά του ολοκληρωμένου.



Συνδεσμολογία γέφυρας και κινητήρα

Το αισθητήριο κίνησης αποτελείται από τρεις ακροδέκτες. Στα αριστερά βρίσκεται ο ακροδέκτης γείωσης, στη μέση η έξοδος (data) και στα δεξιά η τάση τροφοδοσίας. Ενώνουμε παράλληλα το καλώδιο σήματος με το καλώδιο τροφοδοσίας. Παίρνουμε τάση 5 volts από τον Arduino και από τη ψηφιακή έξοδο 21 ενώνουμε το «πόδι» σήματος.

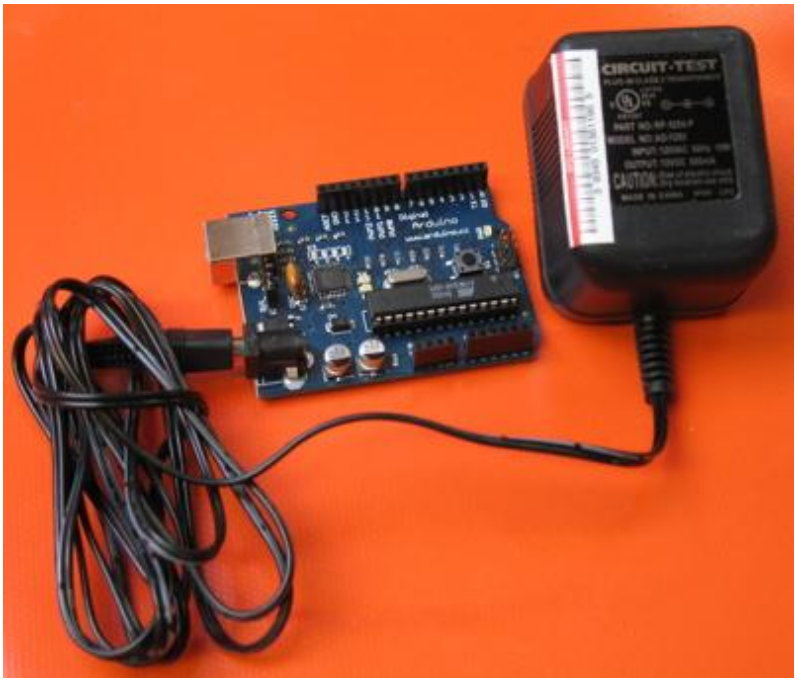




Το αισθητήριο θερμοκρασίας LM35 αποτελείται και αυτό από τρεις ακροδέκτες. Στα αριστερά η τάση τροφοδοσίας στη μέση ο ακροδέκτης εξόδου (data) και στα δεξιά η γείωση, παίρνουμε τάση από τον Arduino για να τροφοδοτήσουμε το αισθητήριο και τον ακροδέκτη εξόδου τον συνδέουμε στην αναλογική έξοδο 16 του Arduino.

Τροφοδοσία συστήματος

Ο Arduino για τη λειτουργία του χρειάζεται τάση 7 ως 12 volts D.C. Αυτήν την τάση μπορούμε να την έχουμε από το καλώδιο USB από τον ηλεκτρονικό υπολογιστή ή από μια 9 Volt μπαταρία εάν και δεν συνίσταται λόγω της περιορισμένης διάρκειας «ζωής» της. Η καταλληλότερη λύση για ένα σύστημα το οποίο θα είναι εξ' ολοκλήρου αυτόνομο χωρίς να εξαρτάται από τον υπολογιστή είναι ένας adaptor.



Τροφοδοτικό 9-12 volts d.c.

Το συγκεκριμένο τροφοδοτικό είναι επαρκές για να τροφοδοτήσει τον Arduino, το GPRS Shield, τα αισθητήρια θερμοκρασίας, τους ανιχνευτές κίνησης και τον ηλεκτρονόμο. Ο κινητήρας δεν αποτελεί επιπλέον βάρος στην τροφοδοσία του συστήματος αφού για την λειτουργία του καταναλώνει τάση από μια 9-volts μπαταρία όπου και έχουμε συνδέσει στη γέφυρα Η.

Σενάριο λειτουργίας

Το σκεπτικό λειτουργίας έχει ως εξής: Η πλατφόρμα Arduino έχει αποθηκευμένη στη μνήμη της ορισμένα τηλέφωνα χρηστών όπου χειροκίνητα προσθέσαμε εμείς. Αυτοί οι χρήστες θα έχουν πρόσβαση στο σύστημα.

Η λειτουργία γίνεται εξ' ολοκλήρου με την υπηρεσία SMS (Short Message Service), δηλαδή ένας χρήστης θα μπορεί να στέλνει ένα μήνυμα στην πλατφόρμα με κάποια αιτήματα και να λαμβάνει πίσω ορισμένες πληροφορίες.

Ο Arduino όταν λάβει ένα εισερχόμενο μήνυμα το πρώτο πράγμα που θα κάνει είναι να ελέγξει εάν ο αριθμός του αποστολέα είναι στη λίστα χρηστών (users), στην περίπτωση που ο αριθμός είναι άγνωστος, το μήνυμα αγνοείται και διαγράφεται αμέσως, εάν ο αριθμός είναι καταχωρημένος τότε το περιεχόμενο του μηνύματος προωθείται στη "μηχανή" αναγνώρισης αιτημάτων, και το μήνυμα πάλι διαγράφεται αμέσως για αποφυγή υπερχείλισης της μνήμης.

Η μηχανή αναγνώρισης αιτήματος λειτουργεί ως εξής: έχουμε προκαθορίσει στη μνήμη του Arduino μια λίστα με λέξεις όπου κάθε λέξη εκτελεί κάποιο συγκεκριμένο κομμάτι του κώδικα. Παίρνουμε το περιεχόμενο του μηνύματος, το βάζουμε σε έναν πίνακα χαρακτήρων, συγκρίνουμε τον πίνακα με κάθε λέξη μέχρι να ταυτιστεί και εφόσον δεν υπάρξει ταύτιση, τότε ο πίνακας μηδενίζεται και δεν γίνεται καμία λειτουργία.

Ο χρήστης μπορεί να στείλει το αίτημα του με κεφαλαία γράμματα ή μικρά, αλλά πρέπει να είναι προσεκτικός με την ορθογραφία διαφορετικά η ταύτιση δε θα γίνει σωστά.

Η λίστα με τα αιτήματα είναι η εξής:

Temp : για να λάβουμε μήνυμα τη θερμοκρασία του χώρου

Motorl : ο κινητήρας να κινηθεί αριστερά

Motorr: ο κινητήρας να κινηθεί δεξιά

Relay0: ο ηλεκτρονόμος να μη λειτουργεί

Relay1: ο ηλεκτρονόμος να λειτουργεί

Security1 : να ενεργοποιηθεί το αισθητήριο κίνησης

Security0 : να απενεργοποιηθεί το αισθητήριο κίνησης

Στην περίπτωση όπου ληφθεί μήνυμα με περιεχόμενο τη λέξη *temp,* ο αλγόριθμος θα ανταποκριθεί στον αποστολέα με ένα μήνυμα όπου θα αναγράφεται η θερμοκρασία του χώρου εκείνη την στιγμή.

Όταν ληφθεί ένα μήνυμα με περιεχόμενο τη λέξη “Motorl” η” Motorr” τότε ο DC κινητήρας θα λειτουργήσει αριστερόστροφα ή δεξιόστροφα αντίστοιχα, για 3 δευτερόλεπτα. Αυτό υποθέτουμε θα ήταν ανάλογο με το άνοιγμα ή το κλείσιμο μιας πόρτας ή κάποιου παραθύρου κλπ.

Για περιεχόμενο “relay1” το φωτιστικό θα ξεκινήσει να λειτουργεί, σε αντίθεση με το περιεχόμενο “relay0” όπου το φωτιστικό θα σταματήσει να λειτουργεί. Επίσης, έχουμε προσθέσει και τη λειτουργία “relay2” όπου το φωτιστικό θα αναβοσβήνει λίγες φορές.

Τέλος για περιεχόμενο “*Secutity1”*, θα ενεργοποιείται το αισθητήριο κίνησης. Όταν ενεργοποιηθεί το αισθητήριο κίνησης ενεργοποιείται ένας αλγόριθμος έλεγχου που όταν ανιχνεύει κίνηση ενεργοποιείται η λειτουργία *relay2* που προαναφέρθηκε, και ενημερώνονται όλοι οι χρήστες με SMS με το περιεχόμενο “*Security alarm*”. Όταν το περιεχόμενο είναι “*Security0*” τότε αυτή η λειτουργία απενεργοποιείται.

Επίσης έχουμε προγραμματίσει των κώδικα έτσι ώστε να κάνει πάντα έλεγχο θερμοκρασίας. Στην περίπτωση όπου η θερμοκρασία έχει υπερβεί τους 80 βαθμούς κελσίου, όλοι οι χρήστες ενημερώνονται με το μήνυμα “fire alarm”.

Κεφάλαιο 3

Εντολές AT, επίσης γνωστές και σαν Hayes AT commands.

Η έννοια των αρχικών AT είναι αμφίλογοι, λέγεται ότι τα αρχικά AT είναι για τις λέξεις "Attention Telephone" αν και μερικοί υποστηρίζουν πως είναι από τις λέξεις "Attention Terminal".

Οι εντολές αυτές μας δίνουν τη δυνατότητα να στέλνουμε οδηγίες σε κινητά τηλέφωνα. Συγκεκριμένα, οι εντολές αυτές στέλνονται στο modem ενός κινητού τηλεφώνου, ή στη δική μας περίπτωση στο chip SIM900 όπου θεωρείται το Modem ή ο επεξεργαστής του *GPRS* Shield που χρησιμοποιήσαμε.

Οι εντολές AT χρησιμοποιούνται για λειτουργίες (συνήθως γίνονται από ένα πληκτρολόγιο), όπως η κλήση ενός τηλεφωνικού αριθμού, η αποστολή, η ανάγνωση ή η διαγραφή του SMS. Ακόμη, χρησιμοποιούνται για την ανάγνωση ή τη διαγραφή μιας επαφής, την ανάγνωση της δύναμης λήψης κλπ.

Όταν θέλουμε να κάνουμε μια εφαρμογή που βασίζεται σε κάποια μορφή ηλεκτρονικού υπολογιστή, όπως για παράδειγμα η παρούσα εργασία, τότε οι εντολές AT είναι απαραίτητες για την επικοινωνία με την “κινητή” συσκευή.

Command	Description
ATA	Answer command
ATD	Dial command
ATH	Hang up call
ATL	Monitor speaker loudness
ATM	Monitor speaker mode
ATO	Go on-line
ATP	Set pulse dial as default
ATT	Set tone dial as default
AT+CSTA	Select type of address
AT+CRC	Cellular result codes

εντολές που αφορούν κλίσης ήχου

Command	Description
AT+CSMS	Select message service
AT+CPMS	Preferred message storage
AT+CMGF	Message format
AT+CSCA	Service centre address
AT+CSMP	Set text mode parameters
AT+CSDH	Show text mode parameters
AT+CSCB	Select cell broadcast message types
AT+CSAS	Save settings
AT+CRES	Restore settings
AT+CNMI	New message indications to TE
AT+CMGL	List messages
AT+CMGR	Read message
AT+CMGS	Send message
AT+CMSS	Send message from storage
AT+CMGW	Write message to memory
AT+CMGD	Delete message

εντολές που αφορούν γραπτά μηνύματα

Οι εντολές που χρησιμοποιήθηκαν από εμάς για αυτή την εργασία είναι οι εξής

AT+CMGF=1 ενεργοποίηση λειτουργίας μηνύματος (SMS)

AT+CMGR=1 διάβασμα του μηνύματος που βρίσκεται στη θέση 1 της μνήμης

AT+CMGD=1,4 διαγραφή όλων των μηνυμάτων από τη μνήμη

AT + CMGS =\"+30XXXXXXXX\" αποστολή γραπτού μηνύματος σε έναν αριθμό, ο κώδικας της χώρας είναι υποχρεωτικό να γραφτεί.

Ο κώδικας στη γλώσσα IDE (Arduino)

Οι δυο πρώτες γραμμές αναφέρονται στις βιβλιοθήκες που ήταν απαραίτητο να συμπεριλάβουμε για να επιτύχουμε το επιθυμητό αποτέλεσμα. Η πλακέτα Arduino έχει ενσωματωμένη υποστήριξη σειριακής επικοινωνίας που επιτρέπει στον μικροελεγκτή ATmega να λαμβάνει σειριακά δεδομένα, ενόσω εκτελεί άλλες λειτουργίες. Αυτή η βιβλιοθήκη χρησιμοποιείται από εμάς για να μπορούμε να επικοινωνούμε με το GPRS Shield μέσω του Arduino. Η συνάρτηση String εξυπηρετεί στη διαχείριση λέξεων, λόγω των γραπτών μηνυμάτων και αυτή η βιβλιοθήκη είναι απαραίτητη.

```
#include <SoftwareSerial.h>

#include <String.h>
```

Παρακάτω είναι όλες οι μεταβλητές που χρησιμοποιήσαμε στον κώδικα. Οι τύποι είναι ακέραιοι (int) χαρακτήρες (char) strings Boolean και softwareserial. Οι μεταβλητές data και data2 είναι πίνακες όπου θα καταχωρούμε μέσα το κείμενο που θα λαμβάνουμε σειριακά από το Shield. Οι πίνακες user1-3 είναι οι τηλεφωνικοί αριθμοί μας. Οι μεταβλητές Boolean είναι όλες “σημαίες” flags όπου σε κάποιο σημείο στον κώδικα θα πρέπει να ενημερώνουν για διάφορες καταστάσεις.

```
int n_sms,x,sms_start,ok,u,h;

char data[256];

char data2[256];

char user1[12]={'3','0','6','9','8','9','7','6','1','5','4','0'};//Markos

char user2[12]={'3','0','6','9','4','6','3','1','2','3','9','0'};//Marios

char user3[12]={'3','0','6','9','7','9','3','6','1','1','8','4'};//Stelios

boolean response=false;

boolean security=false;

boolean worningsend=false;
```

```
char
num1,num2,num3,num4,num5,num6,num7,num8,num9,num10,num11,num12,num13,num14
,num15;

char
arithmos[12]={'num1','num2','num3','num4','num5','num6','num7','num8','num9','num10','nu
m11','num12'};
```

Τη μεταβλητή cell τη δηλώνουμε σαν αντικείμενο software serial και όποτε θέλουμε να λάβουμε ή να στείλουμε σειριακά δεδομένα από και προς το Shield θα το χρησιμοποιούμε με τον εξής τρόπο:
cell.read() για να λάβουμε δεδομένα και cell.print() για να στείλουμε δεδομένα. Οι αριθμοί στην παρένθεση μάς ενημερώνουν για το ποια πόδια (Pins) ορίσαμε σαν TransmitX και RisiveX όπου θα στέλνουν δεδομένα στο Rx και θα λαμβάνουμε δεδομένα από το TX του Shield αντίστοιχα.

```
SoftwareSerial cell(10,11);

int flag=1;

int therm;

char a,b,c,d,e,f,g;

int enablepin=16; // podi energopoieisis kinitira

int motorpin1=17;

int motorpin2=18;

int relay=23; // pin energopoiieisis hlektronomoy

int from=0;  //1 gia marko - 2 gia Mario - 3 gia stelio

int temphigh=51; //pin trofodosias temperature sensor

int tempin=15; //analog pin for data from temp sensor

int pirpin=21 ; // pin data apo motion sensor

int pirpinhigh=47; //pin trofodosias motion sensor
```

Η συνάρτηση setup() καλείται όταν ένα “σκίτσο*” ξεκινάει. Χρησιμοποιείται για να αρχικοποιήσουμε μεταβλητές, λειτουργίες Pins κλπ. Αυτή η συνάρτηση θα λειτουργήσει μόνο μια φορά όταν ξεκινήσει ο κώδικας και γενικά κάθε φορά που κάνει επανεκκίνηση ο Arduino.

```
void setup()
{
```

Ενημερώνουμε τον κώδικα ότι τα ψηφιακά pins 23, 16, 17, 18, 47, και 51 θα είναι έξοδοι, δηλαδή θα δίνουμε εντολή για έξοδο 0 ή 5 volts. Το ψηφιακό pin 21 θα είναι είσοδος, δηλαδή θα δέχεται τάση 0 ή 5 volts, και αυτό θα γίνεται με την εντολή pinMode(x, INPUT ή OUTPUT).

Στη συνέχεια θέτουμε τα ψηφιακά πόδια που θα τροφοδοτούν το αισθητήριο κίνησης, θερμοκρασίας και τον κινητήρα σε υψηλή τάση, δηλαδή HIGH με έξοδο 5 volts, με την εντολή digital Write(x,HIGH)

```
  pinMode(relay, OUTPUT);
  pinMode(pirpinhigh,OUTPUT);
  pinMode(temphigh,OUTPUT);
  pinMode(enablepin, OUTPUT);
  pinMode(motorpin1, OUTPUT);
  pinMode(motorpin2, OUTPUT);
  pinMode(pirpin,INPUT);
  digitalWrite(enablepin, HIGH);
  digitalWrite(pirpinhigh,HIGH);
```

Η εντολή Serial.begin είναι για να ξεκινήσει η σειριακή επικοινωνία του Arduino με τον υπολογιστή και ο αριθμός στην παρένθεση είναι για να θέσουμε τον ρυθμό μετάδοσης των ψηφίων *Baud rate*. Η εντολή cell.begin είναι, επίσης, για να ξεκινήσει η σειριακή επικοινωνία του Arduino, αλλά με το *GPRS* παρελκόμενο με τον επιθυμητό αριθμό μετάδοσης δεδομένων.

Ακολουθεί μια μικρή καθυστέρηση 2 δευτερολέπτων και στέλνουμε σειριακά την εντολή AT+CMGF=1 στο GPRS Shield, η οποία ενημερώνει πώς θέλουμε να λειτουργεί σε κατάσταση γραπτών μηνυμάτων SMS. Η εντολή Serial.println είναι ένα μήνυμα που θα τυπωθεί στην οθόνη του υπολογιστή για να μας ενημερώσει ότι όλες οι παραπάνω εντολές εκτελέστηκαν επιτυχώς.

```
  Serial.begin(9600);

  cell.begin(9600);

  delay(2000);

  cell.println("AT+CMGF=1");        // sets the SMS mode to text

  delay(1000);

  Serial.println("Enarxh kwdika");

}
```

Μετά την συνάρτηση setup, κάθε κώδικας arduino πρέπει να περιλαμβάνει την συνάρτηση Loop, όπου και θεωρείται το κύριο κομμάτι του κώδικα, καθώς θα εκτελείται επ' αόριστον. Δηλαδή όταν φτάσει η τελευταία γραμμή του κώδικα τότε θα εκτελεστεί αμέσως μετά η πρώτη, όπως άλλωστε λέει και το όνομά της "λούπα".

```
void loop()

 {

     digitalWrite(temphigh, HIGH);
```

η μεταβλητή therm είναι μια εξίσωση όπου βρήκαμε στο διαδίκτυο για να υπολογίζει την θερμοκρασία του αισθητηρίου LM35, καθώς αυτή η ένδειξη θα είναι μια τιμή από 0 ως 5 volts.

```
therm=(5*analogRead(tempin)*100)/1023;
```

Η σημαία flag είναι για να ενημερώνει αν δεχτήκαμε νέο μήνυμα. Όταν είναι 1 σημαίνει πώς δεν έχουμε κάποιο μήνυμα και όταν είναι 0, τότε ισχύει το αντίθετο.

```
flag=1;
```

Στέλνουμε την εντολή AT+CMGR=1 στο Shield, που σημαίνει “διάβασε το μήνυμα που βρίσκεται στη θέση μνήμης 1, εάν υπάρχει”.

```
cell.println("AT+CMGR=1");    //Reads the first SMS
```

Η εντολή flush δίνει προτεραιότητα στη μετάδοση των bits έτσι ώστε να μην εκτελεστεί κάποια άλλη εντολή.

```
cell.flush();
```

Ο πίνακας data[] θα κρατήσει το περιεχόμενο των δεδομένων, που το shield θα μας στείλει πίσω, μόλις του ζητήσουμε το μήνυμα. Γι’ αυτό τον λόγο πρέπει να τον καθαρίζουμε πριν από κάθε αίτημα και αυτό γίνεται με τις παρακάτω 4 γραμμές:

```
for (x=0;x < 255;x++)
{
   data[x]='\0';        // μηδενισμός κάθε θέσης πίνακα.
}
x=0;
```

Ενόσω η σειριακή επικοινωνία είναι διαθέσιμη, δηλαδή δεχόμαστε δεδομένα από το shield τότε αποθηκεύουμε τα ληφθέντα στον πίνακα data[]. Τα δεδομένα που θα λαμβάνουμε κάθε φορά θα περιέχουν πληροφορίες, όπως αν το μήνυμα διαβάζεται για πρώτη φορά, την ημερομηνία που στάλθηκε το μήνυμα, και φυσικά τον αριθμό που έστειλε το μήνυμα. Συγκεκριμένα θα είναι της μορφής ***+CMGR: "REC UNREAD","+85291234567",,"07/02/18,00:05:10+32" .*** Εμείς όμως για την συγκεκριμένη εφαρμογή χρειαζόμαστε μόνο τον αριθμό, όπου εμπειρικά βρήκαμε πως θα ξεκινά πάντα στη θέση 24 του πίνακα data[].

```
do{
    while(cell.available()==0);
    data[x]=cell.read();
    x++;
```

Συνθήκη ότι τελείωσε η μετάδοση του μηνύματος και εφόσον είναι αληθής, τότε βάζουμε στον πίνακα arithmos[] τα νούμερα που βρίσκονται στη θέση 24 ως 37.

```
    if(data[x-1]==0x0D&&data[x-2]=='"')
    {

      for (int j=0; j<12; j++)
     {
        arithmos[j]=data[24+j];
      }
```

Μηδενίζουμε ξανά τον πίνακα, ώστε τα αμέσως επόμενα δεδομένα που θα λάβουμε, δηλαδή αυτό που μας έγραψε ο αποστολέας, να μπουν στη θέση data[0]

```
      x=0;
```

Η σημαία flag γίνεται 0 άρα έχουμε νέο μήνυμα.

```
        flag=0;
      }
     }
```

{ Όταν το Shield στείλει τα δεδομένα OK τότε σημαίνει πως η μετάδοση τελείωσε και δεν απομένει να λάβουμε κάτι άλλο. Έτσι ο πίνακας μετά το οκ μηδενίζεται. }

```
while(!(data[x-1]=='K'&&data[x-2]=='O'));
data[x-3]='\0';       //finish the string before the OK
delay(3000);
```

{ Στην οθόνη του υπολογιστή θα τυπωθεί ο αριθμός του αποστολέα και τα δεδομένα που μας έστειλε. }

```
Serial.println("apo arithmo");
delay(500);
Serial.println(arithmos);
Serial.println(data);
```

{ Ok είναι η σημαία που θα μας ενημερώσει στη συνέχεια αν ο αποστολέας είναι έγκυρος }

```
ok=0;
```

{ Παρακάτω γίνεται ο έλεγχος αν ο arithmos[] ταιριάζει σε κάποιον από τους χρήστες. Για κάθε αριθμό θα γίνεται σύγκριση στοιχείο με στοιχείο και κάθε φορά που υπάρχει ταύτιση, το οκ θα αυξάνεται. Όταν γίνει 13 τότε σημαίνει πως ταυτίστηκε πλήρως με κάποιον αριθμό, άρα και ο αποστολέας είναι έγκυρος. }

```
 for (int u=0; u<12; u++)    //elenxos gia proto user
 {
  if(user1[u]==arithmos[u])
```

```
        {
         ok++;
         if (ok==12)
         {
{ Η σημαία response θα γίνει true όταν το οκ είναι 13, που σημαίνει ότι ο αποστολέας είναι έγκυρος και αν είναι false, τότε θα βγαίνει άκυρος. }
         response=true;
{ Για να γνωρίζουμε ποιο πρόσωπο έστειλε το μήνυμα: 1 για Μάρκο 2 για Μάριο 3 για Στέλιο }
         from=1;
         }
        }
       }
       ok=0;
{Ομοίως με παραπάνω για δεύτερο χρήστη }

       for (int p=0; p<12; p++)    //elenxos gia deftero user
       {
        if(user2[p]==arithmos[p])
        {
         ok++;
         if (ok==12)
          {
              response=true;
              from=2;
```

```
    }
   }
  }
  ok=0;
```

{Ομοίως με παραπάνω για τρίτο χρήστη}

```
  for (int l=0; l<12; l++)
  {
   if(user3[l]==arithmos[l])
    {
     ok++;
     if (ok==12)
      {
       response=true;
       from=3;
      }
    }
  }
  ok=0;
```

{Μήνυμα στην οθόνη του υπολογιστή που ενημερώνει ότι αποστολέας είναι έγκυρος.}

```
  If (response==true)
  {
```

```
        Delay (1000);
        Serial.println("number ok");
       }
```

Τα δεδομένα του μηνύματος θα αποθηκευτούν στις παρακάτω 6 μεταβλητές. Στην περίπτωση όπου ο αποστολέας στείλει παραπάνω από 6 χαρακτήρες τότε οι περιττοί θα αγνοηθούν.

```
    a=data[1];
    b=data[2];
    c=data[3];
    d=data[4];
    e=data[5];
    f=data[6];
```

Παρακάτω ξεκινά η διαδικασία αναγνώρισης αιτήματος, η συνθήκη θα ισχύσει όταν όντως έχουμε νέο μήνυμα, flag==0 και αν ο αποστολέας είναι έγκυρος response==true.

```
    If ((flag==0) &&(response==true))
    {
```

Μηδενίζουμε την σημαία response

```
        response=false;
        Serial.println("elenxos minimatos me pinaka");
```

Συγκρίνουμε τις μεταβλητές a,b,c,d,e, και f ξεχωριστά με τα γράμματα Τ ή τ, e,m,p αντίστοιχα, αν υπάρχει ταύτιση, τότε το αίτημα αφορά θερμοκρασία. Τότε θα κληθεί η συνάρτηση θερμοκρασίας όπου αναλύεται παρακάτω.

```
        if (((a=='t')||(a=='T'))&&(b=='e')&&(c=='m')&&(d=='p'))
        {
```

```
        Serial.println("zitises 8ermokrasia");
        temp(); //sinartisi 8ermokrasias
      }
```
{ Σύγκριση με τη λέξη Motorl για κινητήρα αριστερά, να σημειωθεί στο σημείο αυτό ότι το πρώτο γράμμα σε όλες τις συγκρίσεις μπορεί να είναι είτε κεφαλαίο είτε μικρό, για να διευκολυνθεί ο χρήστης. }

```
  if (((a=='M')||(a=='m'))&&(b=='o')&&(c=='t')&&(d=='o')&&(e=='r')&&(f=='l'))
      {
        Serial.println("kinitiras aristera");
        motorl();
```
{ καλείται η συνάρτηση Motorl που αναλύεται παρακάτω. }

```
      }
```
{Σύγκριση με λέξη Motorr }

```
        if (((a=='M')||(a=='m'))&&(b=='o')&&(c=='t')&&(d=='o')&&(e=='r')&&(f=='r'))
      {
        Serial.println("kinitiras deksia");
        motorr();
```
{ καλείται η συνάρτηση Motorr που αναλύεται παρακάτω.}

```
      }
```
{Σύγκριση με λέξη relay1 όπου το φωτιστικό θα μείνει μόνιμα αναμμένο }

```
      if  (((a=='R')||(a=='r'))&&(b=='e')&&(c=='l')&&(d=='a')&&(e=='y')&&(f=='1'))
      {
       digitalWrite(relay,HIGH);
      }
```
{Σύγκριση με λέξη relay0, όπου το φωτιστικό θα σβήσει.}

```
      if  (((a=='R')||(a=='r'))&&(b=='e')&&(c=='l')&&(d=='a')&&(e=='y')&&(f=='0'))
```

```
{
  digitalWrite(relay,LOW);
}
 if  (((a=='R')||(a=='r'))&&(b=='e')&&(c=='l')&&(d=='a')&&(e=='y')&&(f=='2'))
{
  for(int z=0; z<6; z++)
  {
  digitalWrite(relay,HIGH);
  delay(1000);
  digitalWrite(relay,LOW);
  delay(1000);
  }
}
```

Σύγκριση με λέξη secur0 όπου δηλαδή επιθυμούμε να απενεργοποιήσουμε το φωτοκύτταρο, δηλαδή την ασφάλεια και σε αυτή την περίπτωση η σημαία security θα γίνει false.

```
 if (((a=='S')||(a=='s'))&&(b=='e')&&(c=='c')&&(d=='u')&&(e=='r')&&(f=='0'))
 {
   security=false;
 }
```

Σύγκριση με λέξη secur1 όπου δηλαδή επιθυμούμε να ενεργοποιήσουμε το φωτοκύτταρο, δηλαδή την ασφάλεια και σε αυτή την περίπτωση η σημαία security θα γίνει true.

```
 if (((a=='S')||(a=='s'))&&(b=='e')&&(c=='c')&&(d=='u')&&(e=='r')&&(f=='1'))
 {
   security=true;
```

```
        }
    }
```

Το παρακάτω κομμάτι κώδικα αναφέρεται στον καθαρισμό της μνήμης από εισερχόμενα μηνύματα, δηλαδή αν η σημαία flag είναι 0, δηλαδή υπάρχει μήνυμα, τότε να γίνει και η διαγραφή του.

```
        if(flag==0)
        {
        Serial.println("diagrafh mhnymatos");
        cell.println("at");
        delay(500);
        cell.println("ate0");
        delay(500);
        cell.println("AT+CMGD=1,4");
        delay(1000);
```

Επίσης σε αυτό το κομμάτι θα γίνει και ο καθαρισμός του πίνακα arithmos[] όπου περιέχει τον τηλεφωνικό αριθμό του αποστολέα

```
        for (x=0;x < 12;x++)
          {
            arithmos[x]='\0';
          }
        }
```

Συνθήκη πυρασφάλειας, αν η θερμοκρασία υπερβεί τους 80 βαθμούς κελσίου και οι χρήστες δεν έχουν ενημερωθεί, τότε θα κληθεί η συνάρτηση firealarm που αναλύεται παρακάτω.

```
        if ((therm>=80)&&(worningsend==false))
```

```
{
 firealarm();
 Serial.println("fire");
}
```

{ Αν η ασφάλεια έχει προηγουμένως ενεργοποιηθεί μέσω μηνύματος, τότε ενεργοποιείται η ασφάλεια κλοπής, δηλαδή αν το φωτοκύτταρο εντοπίσει κάποια κίνηση, τότε όλοι οι χρήστες θα ενημερωθούν μέσω SMS. }

```
if (security==true)
{
 delay(1000);
 if (digitalRead(pirpin)==HIGH)
 {
  motionalarm();
  Serial.println("motion");
 }
}
```

} { Τέλος συνάρτησης loop }

{ Συνάρτηση Motorl καλείται όταν έρθει ανάλογο εισερχόμενο αίτημα. Το pin 16 θα γίνει high και το pin15 low πετυχαίνοντας αριστερόστροφη φορά για 3 δευτερόλεπτα. }

```
void motorl()
{
   digitalWrite(motorpin1, HIGH);
   digitalWrite(motorpin2, LOW);
   delay(3000);
   digitalWrite(motorpin1, LOW);
   digitalWrite(motorpin2, LOW);
}
```

Συνάρτηση Motorr καλείται όταν έρθει ανάλογο εισερχόμενο αίτημα. Το pin 15 θα γίνει high και το pin16 θα γίνει low, πετυχαίνοντας αριστερόστροφη φορά για 3 δευτερόλεπτα.

```
void motorr()
{
   digitalWrite(motorpin1, LOW);
   digitalWrite(motorpin2, HIGH);
   delay(3000);
   digitalWrite(motorpin1, LOW);
   digitalWrite(motorpin2, LOW);
}
```

Συνάρτηση αποστολή μηνύματος σε αίτημα Temp, η συνάρτηση θα στείλει τη θερμοκρασία χώρου μόνο στον αποστολέα, κοιτάζοντας την σημαία from.

```
void temp()
{
{ Θέτουμε ξανά το Shield σε λειτουργία text }
 cell.println("AT+CMGF=1");    //Because we want to send the SMS in text mode
 delay(1000);

{ Αν ο αποστολέας ήταν ο user1 τότε θα στείλουμε συριακά την εντολή
AT+CMGS=XXXXXXXXXX δηλαδή λειτουργία αποστολής μηνύματος μαζί με τον αριθμό που
επιθυμούμε να στείλουμε, ο έλεγχος γίνεται τρείς φορές. }

 if (from==1)
 {
 cell.println("AT + CMGS =\"+306989761540\"");
 }
 if (from==2)
 {
 cell.println("AT + CMGS =\"+306946312390\"");
 }
 if (from==3)
 {
 cell.println("AT + CMGS =\"+306979361184\"");
 }
```

Αφού βρήκαμε τον κατάλληλο αποστολέα περιμένουμε ένα δευτερόλεπτο και τότε πρέπει να τυπώσουμε το κατάλληλο μήνυμα με την εντολή cell.println και στην παρένθεση το κείμενο που θέλουμε μέσα σε εισαγωγικά.

Αμέσως μετά τυπώνουμε την μεταβλητή temp όπου έχει μέσα αποθηκευμένη την θερμοκρασία. Για να καταλάβει το Shield ότι δεν επιθυμούμε να στείλουμε κάτι άλλο, τότε πρέπει να του στείλουμε την εντολή ctrl+z όπου στον πίνακα ASCII είναι ο χαρακτήρας 26.

```
 delay(1000);
 cell.println("i thermokrasia einai : ");
 delay(100);
 cell.println(therm);
 cell.println((char)26);
 delay(100);
 cell.println();
```

Μηδενίζουμε την σημαία from

```
 from=0;
}
```

Συνάρτηση αποστολής μηνύματος σε όλους τους users μετά από εντοπισμό θερμοκρασίας άνω των 80 βαθμών κελσίου. Συγκεκριμένα, θα επαναληφθεί τρεις φορές η διαδικασία αποστολής μηνύματος που περιγράφεται παραπάνω, μία για κάθε χρήστη.

Η σημαία worningsend θα πρέπει να γίνει true έτσι ώστε, όταν αποσταλούν όλα τα μηνύματα, να μην ξανακληθεί αυτή η συνάρτηση.

```
void firealarm()
{
  worningsend=true;
  cell.println("AT+CMGF=1");
  delay(1000);
  cell.println ("AT + CMGS =\"+306989761540\"");
  delay(1000);
  cell.println("kindinos pirkagias : ");
  delay(100);
  cell.println(therm);
  cell.println((char)26);
  delay(100);
  cell.println();
  delay(1000);
  cell.println ("AT + CMGS =\"+306946312390\"");
  delay(1000);
  cell.println("kindinos pirkagias : ");
  delay(100);
  cell.println(therm);
  cell.println((char)26);
  delay(100);
  cell.println();
  delay(1000);
  cell.println ("AT + CMGS =\"+306979361184\"");
```

```
 delay(1000);
 cell.println("kindinos pirkagias : ");
 delay(100);
 cell.println(therm);
 cell.println((char)26);
 delay(100);
 cell.println();
}
```

Συνάρτηση ενημέρωσης όλων τον Users ότι το αισθητήριο εντόπισε κίνηση στο χώρο αφού ενεργοποιήθηκε η λειτουργία Secur1. Τρία μηνύματα θα σταλθούν και εδώ όπως περιγράφηκε στη συνάρτηση Temp.

```
void motionalarm()
{
 cell.println("AT+CMGF=1");
 delay(1000);
 cell.println ("AT + CMGS =\"+306989761540\"");
 delay(1000);
 cell.println("kindinos klopis ");
 cell.println((char)26);//the ASCII code of the ctrl+z is 26
 delay(100);
 cell.println();
 delay(2000);
 cell.println ("AT + CMGS =\"+306946312390\"");
```

```
 delay(1000);
 cell.println("kindinos klopis ");
 delay(100);
 cell.println((char)26);
 delay(100);
 cell.println();
 delay(2000);
 cell.println ("AT + CMGS =\"+306979361184\"");
 delay(1000);
 cell.println("kindinos klopis ");
 delay(100);
 cell.println(therm);
 cell.println((char)26);//the ASCII code of the ctrl+z is 26
 delay(100);
 cell.println();
}
```

*σκίτσο Sketch ονομάζεται ένα κομμάτι κώδικα γραμμένο στη γλώσσα IDE

*σε περίπτωση που δεν υπάρχει κάρτα SIM στην ειδική υποδοχή του Shield, τότε ο κώδικας δεν θα λειτουργήσει κανονικά !

*για να λειτουργήσει κανονικά ο κώδικας, τότε η κάρτα SIM θα πρέπει να είναι ξεκλείδωτη, να μην υπάρχει κωδικός PIN !

Μελλοντικές Βελτιώσεις

Οι μελλοντικές βελτιώσεις χωρίζονται σε δυο κατηγορίες, στις κατασκευαστικές και στις βελτιώσεις κώδικα.

Κατασκευαστικές βελτιώσεις

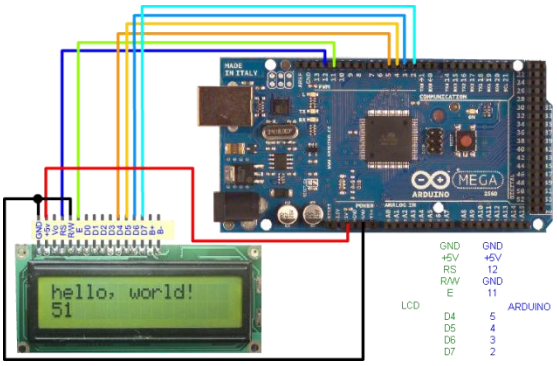
- Αισθητήρια μεγαλύτερης ακρίβειας. Η επιλογή τον αισθητηρίων επιλέγεται φυσικά, όπως προαναφέρθηκε, σύμφωνα με τις ανάγκες του κάθε ανθρώπου.

- Προσθήκη μικροφώνου και μεγαφώνου. Το GPRS Shield έχει υποδοχές για μικρόφωνο και μεγάφωνο και προσθέτοντάς τα μαζί θα μπορούμε να ακούμε όποτε επιθυμούμε τί γίνεται στο χώρο ή ακόμη και να μιλήσουμε.

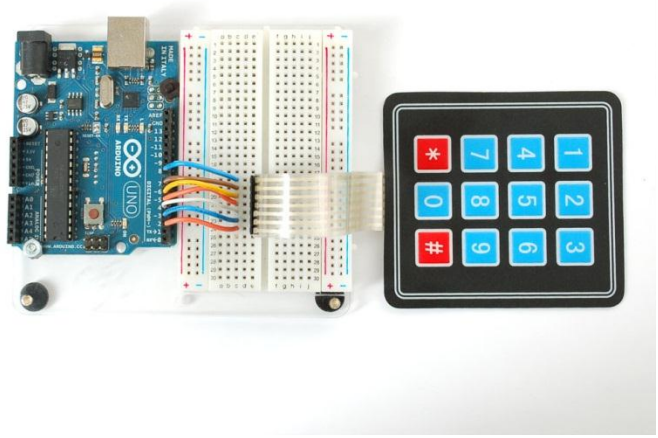
- Προσθήκη κάμερας για σκοπούς παρακολούθησης. Για παράδειγμα, σε περίπτωση προβλήματος ασφάλειας μια κάμερα θα μπορούσε να βιντεοσκοπεί το χώρο ή ακόμα και να γίνεται βίντεο κλήση σε πραγματικό χρόνο με τους χρήστες.

- Παρόλο που ο επεξεργαστής ATmega του Arduino εξυπηρετούσε και με το παραπάνω τις ανάγκες της κατασκευής, θα μπορούσαμε να επιλέξουμε έναν πιο γρήγορο επεξεργαστή με περισσότερες υπολογιστικές ικανότητες, όπως για παράδειγμα ο Arduino due, που θεωρείται η δυνατότερη πλακέτα απ' όλες της πλακέτες Arduino.

- Μια οθόνη ενδείξεων με ένα πληκτρολόγιο θα ήταν πολύ καλή προσθήκη στην κατασκευή. Προς το παρόν η εργασία δεν μπορεί να λειτουργήσει χωρίς την οθόνη του υπολογιστή, γιατί στην πραγματικότητα δεν είναι πολύ πρακτικό. Με μια οθόνη όμως θα μπορούσαν να εξυπηρετηθούν πολλές ανάγκες, όπως αλλαγή παραμέτρων χωρίς να χρειάζεται να αλλάζουμε τον κώδικα κάθε φορά, να απενεργοποιούμε λειτουργίες, όπως η ασφάλεια χωρίς να χρειάζεται να χρεωνόμαστε κάθε φορά το κόστος ενός γραπτού μηνύματος και άλλα.



LCD οθόνη



Βελτιώσεις κώδικα

- Η προσθήκη ή η διαγραφή χρηστών θα μπορούσε να γίνεται μέσω γραπτών μηνυμάτων, για παράδειγμα με το μήνυμα Add user θα μας επιτρεπόταν να προσθέσουμε έναν καινούριο αριθμό.

- Σε περίπτωση που κάποιος μη εξουσιοδοτημένος αριθμός προσπαθούσε να αποκτήσει πρόσβαση στην κατασκευή, εμείς θα ενημερωνόμασταν γι' αυτό το γεγονός αναλυτικά.

- Θα μπορούσαμε να αφιερώσουμε ένα κομμάτι μνήμης για σκοπό δημιουργίας μιας λίστας όπου θα καταχωρούνταν μέσα οι προηγούμενές μας αιτήσεις για δημιουργία αρχείου ιστορικού.

- Η πρόσβαση στην κατασκευή μέσο μίας ηλεκτρονικής ιστοσελίδας θα ήταν πολύ καλή αναβάθμιση εφόσον το GPRS Shield είναι έτοιμο για επικοινωνία με το διαδίκτυο.

Πηγές

Κεφάλαιο 1

Οικιακός αυτοματισμός
http://en.wikipedia.org/wiki/Home_automation

κτιριακός αυτοματισμός
http://en.wikipedia.org/wiki/Building_automation

δίκτυο GSM

http://en.wikipedia.org/wiki/GSM

κεφάλαιο 2

Η πλακέτα Arduino
http://www.arduino.cc/
http://arduino.cc/en/Main/ArduinoBoardMega2560

Το GSM/GPRS Shield
http://www.geeetech.com/wiki/index.php/Arduino_GPRS_Shield

το αισθητήριο θερμοκρασίας lm35
http://www.datasheets360.com/search/results?query=lm35&se=ggka&setag=d360&cid=goog&cid=paidsearch&gclid=CIW-odH9wLoCFbMbtAodiSgA_Q\

αισθητήριο κίνησης.
http://en.wikipedia.org/wiki/Passive_infrared_sensor

Η γέφυρα Η L293d
http://en.wikipedia.org/wiki/H_bridge
http://www.electroons.com/electroons/dc_motor_control.html

κεφάλαιο 3

οι εντολές AT
http://en.wikipedia.org/wiki/Hayes_command_set

http://www.simcom.us/act_admin/supportfile/SIM900_ATC_V1.00.pdf

ο κώδικας

http://www.cooking-hacks.com/documentation/tutorials/arduino-gprs-gsm-quadband-sim900

http://playground.arduino.cc/Main/LM35HigherResolution