



ΑΝΩΤΑΤΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΠΕΙΡΑΙΑ
ΤΜΗΜΑ ΑΥΤΟΜΑΤΙΣΜΟΥ



«Σύστημα ασφάλειας οχήματος με χρήση τεχνολογιών RFID, GPS, 3G/GSM»

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΨΑΡΡΑΣ ΓΕΩΡΓΙΟΣ 36825

Επιβλέπων Καθηγητής:

Δρ. Παπουτσιδάκης Μιχαήλ

Πειραιάς, Ιούνιος 2013

Ευχαριστίες

Η ολοκλήρωση της παρούσας εργασίας, πήρε αρκετό χρόνο. Ωστόσο, η ηθική ικανοποίηση από το αποτέλεσμα καθώς και η γνώση που αποκόμισα από την μελέτη των απαραίτητων γνωστικών πεδίων για την εκπόνησή της, ήταν το μέγιστο κέρδος. Σε αυτό το σημείο οφείλω να ευχαριστήσω όσους στάθηκαν αρωγοί σε αυτή την προσπάθεια.

Θα ήθελα να ευχαριστήσω θερμά τον κ. Παπουτσιδάκη Μιχαήλ, Καθηγητή του τμ. Αυτοματισμού του Ανώτατου Τεχνολογικού Εκπαιδευτικού Ιδρύματος Πειραιά για την δυνατότητα που μου έδωσε να μελετήσω έναν τομέα που με ενδιέφερε πάρα πολύ, την επίβλεψη της εργασίας καθώς και για την εμπιστοσύνη που έδειξε στο πρόσωπό μου.

Τους καθηγητές μου, κ. Δημογιαννόπουλο Δημήτριο και κ. Πολίτη Γεώργιο για την ηθική συμπαράσταση και την παραχώρηση του εργαστηρίου των Συστημάτων Αυτομάτου Ελέγχου II ως χώρο μελέτης και δοκιμών των εφαρμογών της εργασίας μου.

Τέλος θα ήθελα να την αφιερώσω στην Άννα, που με στήριξε με την υπομονή και την αγάπη της, στην Κρίστη για τις καλλιτεχνικές πινελιές και στον Γιάννη για τις πολύτιμες ιδέες του.

*Όλα τα πράγματα είναι δύσκολα,
προτού γίνουν εύκολα. ...*

*(Thomas Fuller, 1608-1661,
Άγγλος στοχαστής)*

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

Εισαγωγή	5
ΚΕΦΑΛΑΙΟ 1	6
1.1 Arduino	6
1.2 RFID	13
1.3 GPS	17
1.4 3G/GPRS	23
ΚΕΦΑΛΑΙΟ 2	27
2.1 Η κατασκευή	27
2.2 Περιγραφή σεναρίου	37
2.3 Επεξήγηση Κώδικα	41
ΚΕΦΑΛΑΙΟ 3	64
3.1 Μελλοντική εξέλιξη	64
Βιβλιογραφία	66

Εισαγωγή

Η ιστορία των μεταφορών είναι συνυφασμένη με την ύπαρξη της ανθρώπινης ζωής. Ο πρωτόγονος άνθρωπος μετακινούνταν βαδίζοντας σε αναζήτηση τροφής ή από περιέργεια να γνωρίσει το περιβάλλον του, ή ακόμα και για την προστασία του από τους διάφορους φυσικούς κινδύνους (όπως σε αναζήτηση κάποιου καταφυγίου - σπηλιάς). Γρήγορα όμως κατάλαβε ότι οι φυσικές του αντοχές για να διανύει μεγάλες αποστάσεις ήταν περιορισμένες και πολύ περισσότερο περιορισμένη η ικανότητά του να μεταφέρει βάρη σε σημαντικές αποστάσεις.

Οι αδυναμίες αυτές οδήγησαν τον άνθρωπο σε αναζήτηση διαφόρων μέσων μεταφοράς τόσο για τον ίδιο όσο και για τα αγαθά του, ξεκινώντας αρχικά τη χρησιμοποίηση ζώων στη ξηρά και από το πρωτόγονο μονόξυλο στις λίμνες και τους ποταμούς βγήκε στη θάλασσα. Έτσι παράλληλα με τις καταπληκτικές του εφευρέσεις έφθασε από τον τροχό, το κουπί, το πανί και τον ατμό στους σύγχρονους αεροστρόβιλους των εξελιγμένων σύγχρονων μέσων μεταφορών.

Στις μέρες μας, τα περισσότερο ευρέως χρησιμοποιούμενα μέσα μεταφοράς είναι το αυτοκίνητο και η μοτοσυκλέτα. Ο μεγάλος αριθμός των οχημάτων αυτών των τύπων, τα έχει κάνει βασικό υποψήφιο στόχο για κλοπή. Οι εμπορικές λύσεις που προτείνονται αυτή τη στιγμή για την προστασία του ιδιοκτήτη του οχήματος από κλοπή χαρακτηρίζονται από μειωμένη αποτελεσματικότητα. Ταυτόχρονα παρατηρείται μειωμένη διείσδυση νέων τεχνολογιών στον τομέα της ασφάλειας των οχημάτων. Αυτός ο προβληματισμός αποτέλεσε το έναυσμα για την εκπόνηση της παρούσας πτυχιακής εργασίας.

Θεματικά, η παρούσα πτυχιακή εργασία έχει δομηθεί από τρία κεφάλαια. Στο πρώτο κεφάλαιο γίνεται αναφορά στις τεχνολογίες που έχουν χρησιμοποιηθεί για την κατασκευή, παρατίθενται ιστορικά στοιχεία για τη χρήση τους και αναλύεται ο τρόπος λειτουργίας τους.

Στο δεύτερο κεφάλαιο γίνεται εκτεταμένη μελέτη της κατασκευής, παρουσιάζονται αναλυτικά τα σενάρια λειτουργίας της κατασκευής και τέλος παρουσιάζεται και αναλύεται ο κώδικας του προγράμματος που χρησιμοποιήθηκε.

Τέλος, η εργασία κλείνει με το τρίτο κεφάλαιο στο οποίο παρατίθενται ιδέες για μελλοντικές επεκτάσεις, βελτιστοποιήσεις και περεταίρω εξέλιξη της κατασκευής.

Κεφάλαιο 1

1.1 Arduino

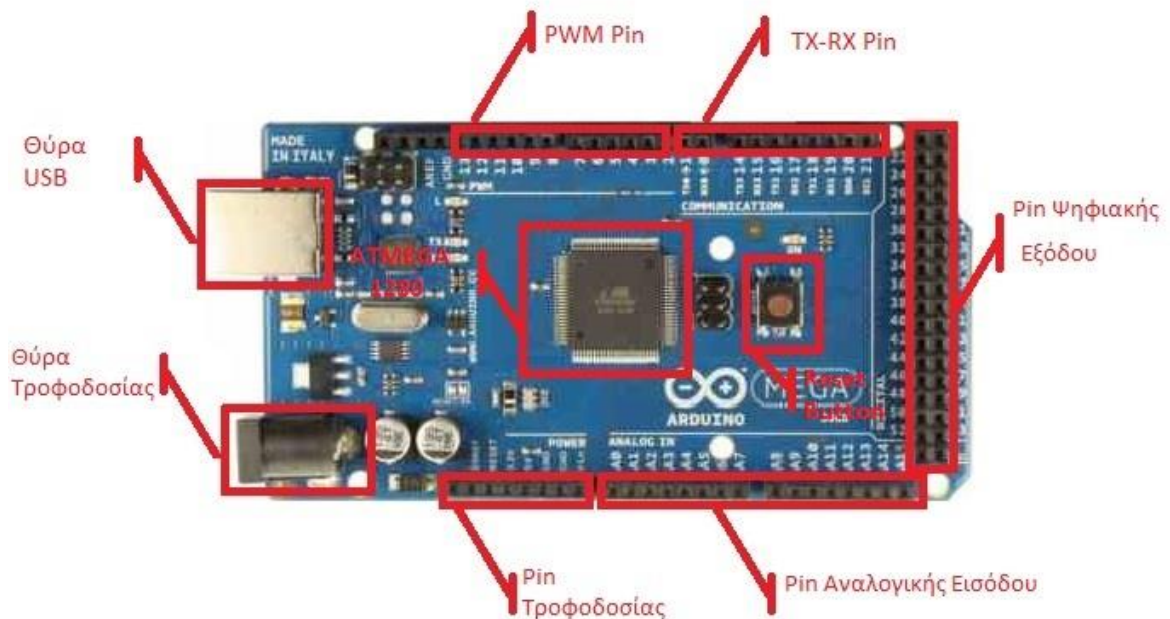
Το Arduino είναι μια «ανοικτού κώδικα» πλατφόρμα «πρωτοτυποποίησης» ηλεκτρονικών βασισμένη σε ευέλικτο και εύκολο στη χρήση υλικό (hardware) και λογισμικό (software) που προορίζεται για οποιονδήποτε έχει λίγη προγραμματιστική εμπειρία, στοιχειώδεις γνώσεις ηλεκτρονικών και ενδιαφέρεται να δημιουργήσει διαδραστικά αντικείμενα ή περιβάλλοντα.

Στην ουσία, πρόκειται για ένα ηλεκτρονικό κύκλωμα που βασίζεται στον μικροελεγκτή ATmega της Atmel και του οποίου όλα τα σχέδια, καθώς και το software που χρειάζεται για την λειτουργία του, διανέμονται ελεύθερα και δωρεάν ώστε να μπορεί να κατασκευαστεί από τον καθένα (απ' όπου και ο περίεργος -για hardware- χαρακτηρισμός «ανοικτού κώδικα»). Αφού κατασκευαστεί, μπορεί να συμπεριφερθεί σαν ένας μικροσκοπικός υπολογιστής, αφού ο χρήστης μπορεί να συνδέσει επάνω του πολλαπλές μονάδες εισόδου/εξόδου και να προγραμματίσει τον μικροελεγκτή να δέχεται δεδομένα από τις μονάδες εισόδου, να τα επεξεργάζεται και να στέλνει κατάλληλες εντολές στις μονάδες εξόδου.

Το Arduino βέβαια, δεν είναι ούτε ο μοναδικός, ούτε και ο καλύτερος δυνατός τρόπος για την δημιουργία μιας οποιασδήποτε διαδραστικής ηλεκτρονικής συσκευής. Όμως το κύριο πλεονέκτημά του είναι η τεράστια κοινότητα που το υποστηρίζει και η οποία έχει δημιουργήσει, συντηρεί και επεκτείνει μια ανάλογοι μεγέθους online γνωσιακή βάση. Έτσι, παρότι ένας έμπειρος ηλεκτρονικός μπορεί να προτιμήσει διαφορετική πλατφόρμα ή εξαρτήματα ανάλογα με την εφαρμογή που έχει στον νου του, το Arduino, με το εκτενές documentation, καταφέρνει να κερδίσει, εκτός από τους προχωρημένους, και όλους εκείνους τους χρήστες των οποίων οι γνώσεις στα ηλεκτρονικά περιορίζονται στα βασικά. (βλέπε[1]).

Το Arduino κυκλοφορεί σε αρκετές διαφορετικές εκδόσεις, επίσημες και ανεπίσημες. Οι κύριες διαφορές των εκδόσεων εντοπίζονται κυρίως η ποσότητα της διαθέσιμης μνήμης, ο αριθμός των διαθέσιμων pin εισόδου και εξόδου, ο αριθμός των σειριακών interface καθώς και η ύπαρξη κάποιων έξτρα χαρακτηριστικών (usb θύρα, αποσπώμενο chip μικροελεγκτή κ.α.). Για τις ανάγκες τις παρούσας κατασκευής επελέγη η επίσημη έκδοση «Arduino Mega 2560 Rev.3» (Εικόνα 1). Αυτό συνέβη λόγω των αυξημένων αναγκών σε θύρες εισόδου, εξόδου και σειριακών interface.

Στην προγραμματιστική αυτή πλατφόρμα, ιδιαίτερη αναφορά αξίζει να γίνει σε πέντε θεμελιώδη στοιχεία: τον μικροελεγκτή, τις εισόδους/εξόδους, την τροφοδοσία, τα ενσωματωμένα κουμπιά και LED και τα shield.



Εικόνα 1. Arduino Mega 2560 Rev.3

Μικροελεγκτής

Το Arduino Mega βασίζεται στον ATmega1280, έναν 8-bit RISC μικροελεγκτή, τον οποίο χρονίζει στα 16MHz. Ο ATmega1280 διαθέτει ενσωματωμένη μνήμη τριών τύπων:

- 8Kb μνήμης SRAM που είναι η ωφέλιμη μνήμη που μπορούν να χρησιμοποιήσουν τα προγράμματα του χρήστη για να αποθηκεύουν μεταβλητές, πίνακες κ.λπ. κατά το runtime. Όπως και σε έναν υπολογιστή, αυτή η μνήμη χάνει τα δεδομένα της όταν η παροχή ρεύματος στο Arduino σταματήσει ή αν γίνει reset.

- 4Kb μνήμης EEPROM η οποία μπορεί να χρησιμοποιηθεί για «ωμή» εγγραφή/ανάγνωση δεδομένων (χωρίς datatype) ανά byte από τα προγράμματα κατά το runtime. Σε αντίθεση με την SRAM, η EEPROM δεν χάνει τα περιεχόμενά της με απώλεια τροφοδοσίας ή reset οπότε είναι το ανάλογο του σκληρού δίσκου.

- 128Kb μνήμης Flash, από τα οποία τα 2Kb χρησιμοποιούνται από το firmware του Arduino που έχει εγκαταστήσει ήδη ο κατασκευαστής του. Το firmware αυτό που στην ορολογία του Arduino ονομάζεται bootloader είναι αναγκαίο για την εγκατάσταση των προγραμμάτων του χρήστη στον μικροελεγκτή μέσω της θύρας USB, χωρίς δηλαδή να χρειάζεται εξωτερικός hardware programmer. Τα υπόλοιπα 126Kb της μνήμης Flash χρησιμοποιούνται για την αποθήκευση αυτών ακριβώς των προγραμμάτων, αφού πρώτα μεταγλωττιστούν στον υπολογιστή του χρήστη. Η μνήμη Flash, όπως και η EEPROM δεν χάνει τα περιεχόμενά της με απώλεια τροφοδοσίας ή reset. Επίσης, ενώ η μνήμη Flash υπό κανονικές συνθήκες δεν προορίζεται για

χρήση runtime μέσα από τα προγράμματα του χρήστη, λόγω της μικρής συνολικής μνήμης που είναι διαθέσιμη σε αυτά (8Kb SRAM + 4Kb EEPROM), έχει σχεδιαστεί μια βιβλιοθήκη που επιτρέπει την χρήση όσου χώρου περισσεύει.

Είσοδοι / έξοδοι

Καταρχήν το Arduino διαθέτει σειριακό interface. Ο μικροελεγκτής ATmega1280 υποστηρίζει σειριακή επικοινωνία, την οποία το Arduino προωθεί μέσα από έναν ελεγκτή Serial-over-USB ώστε να συνδέεται με τον υπολογιστή μέσω USB. Η σύνδεση αυτή χρησιμοποιείται για την μεταφορά των προγραμμάτων που σχεδιάζονται από τον υπολογιστή στο Arduino αλλά και για αμφίδρομη επικοινωνία του Arduino με τον υπολογιστή μέσα από το πρόγραμμα την ώρα που εκτελείται. Επιπλέον, στην πάνω πλευρά του Arduino Mega 2560 βρίσκονται 54 θηλυκά pin, αριθμημένα από 0 ως 53, που μπορούν να λειτουργήσουν ως ψηφιακές εισοδοι και έξοδοι. Λειτουργούν στα 5V και καθένα μπορεί να παρέχει ή να δεχτεί το πολύ 40mA.

Ως ψηφιακή έξοδος, ένα από αυτά τα pin μπορεί να τεθεί από το πρόγραμμα του χρήστη σε κατάσταση HIGH ή LOW, οπότε το Arduino θα ξέρει αν πρέπει να διοχετεύσει ή όχι ρεύμα στο συγκεκριμένο pin. Με αυτόν τον τρόπο μπορεί λόγω χάρη να ανάψει και να σβήσει ένα LED που έχει συνδεθεί στο συγκεκριμένο pin. Αν πάλι ρυθμιστεί ένα από αυτά τα pin ως ψηφιακή είσοδος μέσα από το πρόγραμμα, μπορεί με την κατάλληλη εντολή να διαβαστεί η κατάστασή του (HIGH ή LOW) ανάλογα με το αν η εξωτερική συσκευή που έχει συνδεθεί σε αυτό το pin διοχετεύει ή όχι ρεύμα στο pin.

Μερικά από αυτά τα 54 pin, εκτός από ψηφιακές εισοδοι/έξοδοι έχουν και δεύτερη λειτουργία. Συγκεκριμένα:

- Τα pin 0,1 και 14-21 λειτουργούν ως RX και TX της σειριακής όταν το πρόγραμμα ενεργοποιεί την σειριακή θύρα. Το Arduino Mega 2560 έχει 5 ζευγάρια TX-RX pin δηλαδή συνολικά 5 σειριακά interface. Έτσι, όταν λόγω χάρη το πρόγραμμα στέλνει δεδομένα στην σειριακή, αυτά προωθούνται και στην θύρα USB μέσω του ελεγκτή Serial-Over-USB αλλά και στο pin 0 για να τα διαβάσει ενδεχομένως μια άλλη συσκευή (π.χ. ένα δεύτερο Arduino στο δικό του pin 1). Αυτό φυσικά σημαίνει ότι για κάθε διαφορετικό σειριακό interface που ενεργοποιείται στο πρόγραμμα, χάνονται 2 ψηφιακές εισοδοι/έξοδοι (ένα ζευγάρι TX-RX pin).
- Τα pin 2,3,18-21 λειτουργούν και ως εξωτερικά interrupt (interrupt 0,1,5,4,3,2 αντίστοιχα). Με άλλα λόγια, μπορούν να ρυθμιστούν μέσα από το πρόγραμμα ώστε να λειτουργούν αποκλειστικά ως ψηφιακές εισοδοι στις οποίες όταν συμβαίνουν συγκεκριμένες αλλαγές, η κανονική ροή του προγράμματος σταματάει *άμεσα* και εκτελείται μια συγκεκριμένη συνάρτηση. Τα εξωτερικά interrupts είναι ιδιαίτερα χρήσιμα σε εφαρμογές που απαιτούν συγχρονισμό μεγάλης ακρίβειας.

- Τα pin 2-13 και 44-46 μπορούν να λειτουργήσουν και ως ψευδοαναλογικές εξόδους με το σύστημα PWM (Pulse Width Modulation), δηλαδή το ίδιο σύστημα που διαθέτουν οι μητρικές των υπολογιστών για να ελέγχουν τις ταχύτητες των ανεμιστήρων. Έτσι, μπορεί να συνδεθεί λόγω χάρη ένα LED σε κάποιο από αυτά τα pin και να ελεγχθεί πλήρως η φωτεινότητά του με ανάλυση 8bit (256 καταστάσεις από 0-σβηστό ως 255-πλήρως αναμμένο) αντί να υπάρχει απλά η δυνατότητα να είναι αναμμένο-σβηστό που παρέχεται από τις υπόλοιπες ψηφιακές εξόδους. Είναι σημαντικό να τονιστεί ότι το PWM δεν είναι πραγματικά αναλογικό σύστημα και ότι θέτοντας στην έξοδο την τιμή 127, δεν σημαίνει ότι η έξοδος θα δίνει 2.5V αντί της κανονικής τιμής των 5V, αλλά ότι θα δίνει ένα παλμό που θα εναλλάσσεται με μεγάλη συχνότητα και για ίσους χρόνους μεταξύ των τιμών 0 και 5V.

Στην κάτω πλευρά του Arduino, με τη σήμανση ANALOG IN, βρίσκεται μια ακόμη σειρά από 16 pin, αριθμημένα από το 0 ως το 15. Το καθένα από αυτά λειτουργεί ως αναλογική είσοδος κάνοντας χρήση του ADC (Analog to Digital Converter) που είναι ενσωματωμένο στον μικροελεγκτή. Για παράδειγμα, μπορεί να τροφοδοτηθεί ένα από αυτά με μια τάση η οποία μεταβάλλεται με ποτενσιόμετρο από 0V ως μια τάση αναφοράς V_{ref} η οποία, αν δεν γίνει κάποια αλλαγή είναι προ ρυθμισμένη στα 5V. Τότε, μέσα από το πρόγραμμα επιτυγχάνεται η «ανάγνωση» της τιμής του pin ως ένας ακέραιος αριθμός ανάλυσης 10-bit, από 0 (όταν η τάση στο pin είναι 0V) μέχρι 1023 (όταν η τάση στο pin είναι 5V). Η τάση αναφοράς μπορεί να ρυθμιστεί με μια εντολή στο 1.1V, ή σε όποια τάση επιθυμεί ο χρήστης (μεταξύ 2 και 5V) τροφοδοτώντας εξωτερικά με αυτή την τάση το pin με την σήμανση AREF που βρίσκεται στην απέναντι πλευρά της πλακέτας. Έτσι, αν τροφοδοτηθεί το pin AREF με 3.3V και στην συνέχεια γίνει ανάγνωση κάποιου pin αναλογικής εισόδου στο οποίο εφαρμόζεται τάση 1.65V, το Arduino θα επιστρέψει την τιμή 512.

Τέλος, καθένα από τα 16 αυτά pin, με κατάλληλη εντολή μέσα από το πρόγραμμα μπορεί να μετατραπεί σε ψηφιακό pin εισόδου/εξόδου όπως τα 54 που βρίσκονται στην απέναντι πλευρά και τα οποία περιεγράφηκαν πριν. Σε αυτή την περίπτωση τα pin μετονομάζονται από 0~15 σε 54~70 αντίστοιχα.

Τροφοδοσία

Το Arduino μπορεί να τροφοδοτηθεί με ρεύμα είτε από τον υπολογιστή μέσω της σύνδεσης USB, είτε από εξωτερική τροφοδοσία που παρέχεται μέσω μιας υποδοχής φισ των 2.1mm (θετικός πόλος στο κέντρο) και βρίσκεται στην κάτω-αριστερή γωνία του Arduino. Για να μην υπάρχουν προβλήματα, η εξωτερική τροφοδοσία πρέπει να είναι από 7 ως 12V και μπορεί να προέρχεται από ένα κοινό μετασχηματιστή του εμπορίου, από μπαταρίες ή οποιαδήποτε άλλη πηγή DC.

Δίπλα από τα pin αναλογικής εισόδου, υπάρχει μια ακόμα συστοιχία από 6 pin με την σήμανση POWER. Η λειτουργία του καθενός έχει ως εξής:

•Το πρώτο, με την ένδειξη RESET, όταν γειωθεί (σε οποιοδήποτε από τα 3 pin με την ένδειξη GND που υπάρχουν στο Arduino) έχει ως αποτέλεσμα την επανεκκίνηση του Arduino.

•Το δεύτερο, με την ένδειξη 3.3V, μπορεί να τροφοδοτήσει τα εξαρτήματα που συνδέει ο χρήστης με τάση 3.3V. Η τάση αυτή δεν προέρχεται από την εξωτερική τροφοδοσία αλλά παράγεται από τον ελεγκτή Serial-Over-USB και έτσι η μέγιστη ένταση που μπορεί να παρέχει είναι μόλις 50mA.

•Το τρίτο, με την ένδειξη 5V, μπορεί να τροφοδοτήσει τα εξαρτήματα με τάση 5V. Ανάλογα με τον τρόπο τροφοδοσίας του ίδιου του Arduino, η τάση αυτή προέρχεται είτε άμεσα από την θύρα USB (που ούτως ή άλλως λειτουργεί στα 5V), είτε από την εξωτερική τροφοδοσία αφού αυτή περάσει από ένα ρυθμιστή τάσης για να την «φέρει» στα 5V.

•Το τέταρτο και το πέμπτο pin, με την ένδειξη GND, είναι γειώσεις.

•Το έκτο και τελευταίο pin, με την ένδειξη *Vin* έχει διπλό ρόλο. Σε συνδυασμό με το pin γείωσης δίπλα του, μπορεί να λειτουργήσει ως μέθοδος εξωτερικής τροφοδοσίας του Arduino, στην περίπτωση που ο χρήστης δεν επιθυμεί να χρησιμοποιήσει την υποδοχή του φιν των 2.1mm. Αν όμως υπάρχει ήδη συνδεδεμένη εξωτερική τροφοδοσία μέσω του φιν, μπορεί να χρησιμοποιηθεί αυτό το pin για να τροφοδοτηθούν εξαρτήματα με την πλήρη τάση της εξωτερικής τροφοδοσίας (7~12V), πριν αυτή περάσει από τον ρυθμιστή τάσης όπως γίνεται με το pin των 5V.

Ενσωματωμένα κουμπιά και LED

Πάνω στην πλακέτα του Arduino υπάρχει ένας διακόπτης micro-switch και 4 μικροσκοπικά LED επιφανειακής στήριξης. Η λειτουργία του διακόπτη (που έχει την σήμανση RESET) και του ενός LED με την σήμανση POWER είναι προφανής.

Τα δύο LED με τις σημάνσεις TX και RX, χρησιμοποιούνται ως ένδειξη λειτουργίας του σειριακού interface, καθώς ανάβουν όταν το Arduino στέλνει ή λαμβάνει (αντίστοιχα) δεδομένα μέσω USB. Σημειώστε ότι τα LED αυτά ελέγχονται από τον ελεγκτή Serial-over-USB και συνεπώς δεν λειτουργούν όταν η σειριακή επικοινωνία γίνεται αποκλειστικά μέσω των ψηφιακών pin 0 και 1.

Τέλος, υπάρχει το LED με την σήμανση L. Το πρώτο πρόγραμμα που καλείται να γράψει ο χρήστης για να δοκιμάσει τη λειτουργία του Arduino μετά την αγορά του (κάτι αντίστοιχο του κλασικού «Hello world» των γλωσσών προγραμματισμού) είναι το «Blink», ένα απλό πρόγραμμα το οποίο κάνει ένα LED να αναβοσβήνει ρυθμικά. Για να μπορεί ο χρήστης να το πραγματοποιήσει αυτό από την πρώτη στιγμή, χωρίς να συνδεθεί τίποτα πάνω στο Arduino, οι κατασκευαστές του σκέφτηκαν να ενσωματώσουν ένα LED στην πλακέτα, το οποίο σύνδεσαν στο ψηφιακό pin 13. Έτσι, ακόμα και αν δεν έχει συνδεθεί τίποτα πάνω στο φυσικό pin 13, αναθέτοντάς του την τιμή HIGH μέσα από το πρόγραμμα, θα ανάψει αυτό το ενσωματωμένο LED. (βλέπε [2])

Shields

Τα shield είναι ολοκληρωμένες πλακέτες που είναι σχεδιασμένες ώστε να κουμπώνουν πάνω στο Arduino προεκτείνοντας την λειτουργικότητά του. Είναι η hardware αντίστοιχη έννοια των plugin, add-on και extension που υπάρχουν στο software.

Μερικά από τα πιο δημοφιλή shield που κυκλοφορούν στο εμπόριο για το Arduino είναι:

- Ethernet shield: Δίνει στο Arduino την δυνατότητα να δικτυωθεί σε ένα LAN ή στο internet μέσω ενός τυπικού καλωδίου Ethernet.
- Wi-Fi shield: Όμοιο με το Ethernet shield, χωρίς φυσικά το καλώδιο.
- Διάφορα shield οθόνης: Προσθέτουν οθόνη στο Arduino. Κυκλοφορούν από απλές οθόνες τύπου calculator μέχρι OLED touchscreen υψηλής ανάλυσης τύπου iPhone.
- Wave shield: Δίνει στο Arduino την δυνατότητα να παίζει ήχους/μουσική από κάρτες SD.
- GPS shield: Προσθέτει GPS δυνατότητες στο Arduino (εντοπισμό στίγματος).
- Διάφορα Motor Shields: Σας επιτρέπουν να οδηγήσετε εύκολα μοτέρ διάφορων τύπων (απλά DC, servo, stepper κ.λπ.) από το Arduino.
- Proto-Shield: Μια προσχεδιασμένη πλακέτα πρωτοτυποποίησης, συμβατή στις διαστάσεις του Arduino και χωρίς εξαρτήματα για να φτιάξετε το δικό σας shield.

Τα shield είναι σχεδιασμένα ώστε αφού κουμπωθούν πάνω στο Arduino να προωθούν τις υποδοχές του, ώστε να μπορείτε να συνδέσετε επιπλέον τα δικά σας εξαρτήματα ή να κουμπώσετε και επόμενο shield. Φυσικά, το κάθε shield χρησιμοποιεί ορισμένους από τους πόρους συνδεσιμότητας του Arduino και έτσι δεν μπορείτε να συνδέσετε απεριόριστα shield. Μάλιστα κάποια shield μπορεί να μην είναι συμβατά μεταξύ τους γιατί χρησιμοποιούν τα ίδια pin του Arduino για επικοινωνία με αυτό. Επίσης, επειδή κάποια shield δεν προωθούν τις συνδέσεις του Arduino (όπως π.χ. οι οθόνες οι οποίες δεν έχουν νόημα αν τις καλύψετε από πάνω με ένα επόμενο shield), υπάρχουν ειδικά shields που κουμπώνουν στο Arduino και δίνουν την δυνατότητα σε δύο άλλα shield να κουμπώσουν πάνω τους, λειτουργώντας σαν πολύπριζα.

Όπως και για το ίδιο το Arduino, το βασικό πλεονέκτημα των shield δεν είναι τόσο το προφανές πλεονέκτημα του έτοιμου hardware όσο ότι συνοδεύονται από έτοιμες βιβλιοθήκες που επιτρέπουν στον χρήστη να γράφει τον κώδικά του σε υψηλό επίπεδο. Έτσι, αφ' ενός δεν χρειάζεται ο χρήστης να καταναλώνει τον χρόνο του στην ανάγνωση πολυσέλιδων datasheet και αφ' εταίρου εκμηδενίζεται η πιθανότητα να γίνει σφάλμα κατά τη σύνδεση του περιφερειακού στο Arduino. Πρώτα συνδέεται το shield, έπειτα εγκαθίσταται η βιβλιοθήκη που το συνοδεύει και

τέλος χρησιμοποιούνται έτοιμες συναρτήσεις τις οποίες παρέχει ο κατασκευαστής μαζί με το shield.

Τα shield είναι το άλφα και το ωμέγα στην δημιουργία ενός πραγματικά πρακτικού project. Αυτός είναι και ο λόγος που δεν συνιστάται η αγορά κάποιας έκδοσης του Arduino που δεν είναι 100% συμβατή με τα shield.

1.2 RFID

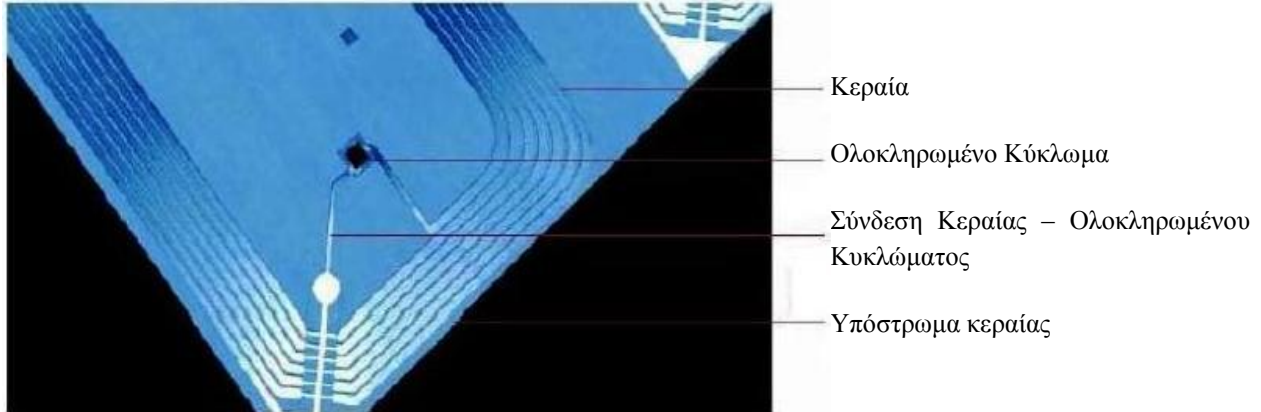
RFID είναι τα αρχικά του όρου Radio Frequency Identification, η απόδοση του οποίου στα ελληνικά ορίζεται ως «ταυτοποίηση μέσω ραδιοσυχνότητων». Τα συστήματα RFID αποτελούν ένα υποσύνολο των Συστημάτων Αυτόματου Προσδιορισμού (Automatic Identification Systems). Ειδικότερα, το RFID λειτουργεί ως γενικός όρος των τεχνολογιών που χρησιμοποιούν ραδιοκύματα για να προσδιορίσουν αυτόματα ανθρώπους ή αντικείμενα και αποτελεί την τεχνολογική εξέλιξη των ραβδωτών κωδίκων (barcode). (βλέπε [3])

Το RFID χαρακτηρίζεται σαν η παλαιότερη νέα τεχνολογία, γιατί οι ρίζες της ξεκινούν από τον Β' Παγκόσμιο πόλεμο. Εκείνη την εποχή, Γερμανοί, Ιάπωνες, Αμερικανοί και Σύμμαχοι χρησιμοποιούσαν το ραντάρ, το οποίο είχε ανακαλυφθεί το 1935 από τον Σκωτσέζο φυσικό Sir Robert Alexander Watson-Watt, για να ενημερώνονται για την προσέγγιση των αεροπλάνων. Δεν υπήρχε όμως τρόπος να προσδιορίσουν ποια αεροπλάνα ήταν εχθρικά και ποια όχι. Οι Γερμανοί πρώτοι παρατήρησαν ότι το ραδιοσήμα άλλαζε όταν οι πιλότοι τους έκαναν μανούβρες καθώς προσεγγίζαν στη βάση. Αυτή την παρατήρηση χρησιμοποίησαν για να αναγνωρίζουν τα δικά τους αεροπλάνα. Αυτό ήταν ουσιαστικά το πρώτο παθητικό σύστημα RFID. Αργότερα, μια ομάδα Βρετανών με επικεφαλής τον Watson-Watt ανέπτυξε το πρώτο ενεργό σύστημα RFID για την συστηματική αναγνώριση φιλικών και εχθρικών αεροπλάνων.

Κατά τις δεκαετίες του '50 και '60, οι επιχειρήσεις άρχισαν να χρησιμοποιούν αντικλεπτικά συστήματα (EAS – Electronic Article Surveillance), τα γνωστά «κόκαλα» που βλέπουμε σήμερα στα μαγαζιά ένδυσης και υπόδησης για να αναγνωρίζουν αν κάτι έχει αγοραστεί νόμιμα ή όχι. Σταθμός στην τεχνολογία RFID ήταν το 1999, όταν οι Uniform Code Council, EAN International, Procter & Gamble και Gillette χρηματοδότησαν το Κέντρο Αυτόματης Αναγνώρισης (Auto ID Center) στο MIT με σκοπό την δραματική μείωση του κόστους εφαρμογής της τεχνολογίας, ώστε να μπορεί να χρησιμοποιηθεί από την Βιομηχανία και το Εμπόριο. Ο επόμενος σταθμός είναι το 2003 όταν η WalMart, η μεγαλύτερη Εταιρία στον κόσμο, ανακοίνωσε ότι από 1/1/2005 οι 100 μεγαλύτεροι προμηθευτές της οφείλουν να την εφοδιάζουν με κιβώτια και παλέτες μαρκαρισμένες με RFID. (βλέπε [4]).

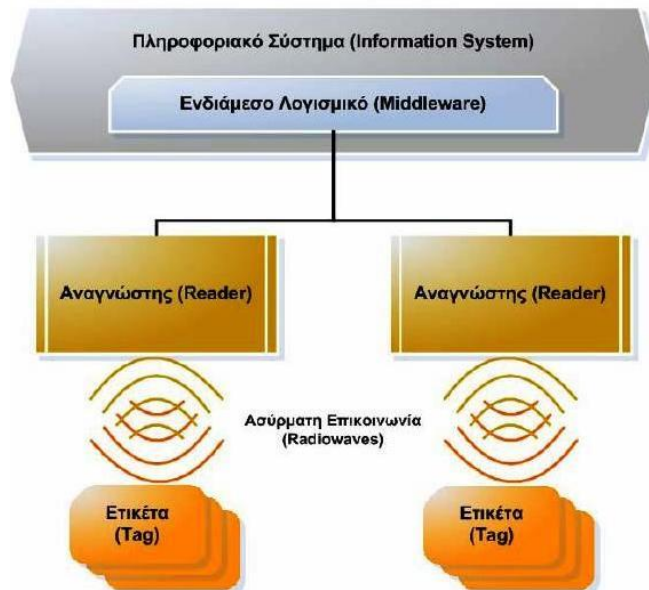
Τα συστήματα RFID απαρτίζονται από δύο κύρια μέρη. Το πρώτο είναι οι πομποδέκτες (transponders) που συχνά αναφέρονται και ως ετικέτες RFID (RFID tags). Οι ετικέτες RFID είναι μικρά chips που αποτελούνται από ένα ολοκληρωμένο κύκλωμα, το οποίο περιλαμβάνει μνήμη ώστε να αποθηκεύει δεδομένα- πληροφορίες, και μία κεραία. Το μέγεθός τους μπορεί να είναι τόσο μικρό όσο το μισό ενός κόκκου άμμου (1/3 του χιλιοστού), ανάλογα με το τύπο τις ετικέτας. (Εικόνα 2)

Το δεύτερο μέρος είναι οι αναγνώστες ή αισθητήρες (readers), οι οποίοι ανακτούν τα δεδομένα από τις ετικέτες RFID. Οι αναγνώστες RFID έχουν ενσωματωμένα μια κεραία και μια μονάδα ελέγχου.



Εικόνα 2. Ετικέτα RFID

Η λειτουργία των συστημάτων RFID είναι απλή και βασίζεται στη δυναμική και αμφίδρομη επικοινωνία των ετικετών και των αναγνώστην. Όταν οι ετικέτες RFID βρεθούν στην εμβέλεια της κεραίας του αναγνώστη, η μονάδα ελέγχου επικοινωνεί με ραδιοκύματα με την κεραία των ετικετών RFID. Οι ετικέτες RFID ενεργοποιούνται με τη σειρά τους και επιστρέφουν τα αναζητούμενα δεδομένα στους αναγνώστες. Στη συνέχεια παρεμβαίνει ένα ενδιάμεσο λογισμικό, το οποίο κατανοεί τις πληροφορίες, οι οποίες αποστέλλονται από τη μονάδα ελέγχου του αναγνώστη. Ο αναγνώστης τις μεταφέρει στο εκάστοτε πληροφοριακό σύστημα. (Εικόνα 3)



Εικόνα 3. Αρχιτεκτονική συστήματος RFID

Οι ετικέτες RFID κατηγοριοποιούνται σε τρεις τύπους ανάλογα με τον τρόπο επικοινωνίας μεταξύ των ετικετών και των αναγνωστών, στις ενεργές ετικέτες, στις παθητικές ετικέτες και στις ημι-παθητικές ετικέτες. Ένα ολοκληρωμένο κύκλωμα στις ετικέτες RFID μπορεί να περιέχει μνήμη μόνο για ανάγνωση (read only memory - ROM), επανεγγράψιμη μνήμη (Read – Write), μνήμη μιας εγγραφής και πολλών αναγνώσεων (Write Once and Read Many memory - WORM). Στο ολοκληρωμένο κύκλωμα με μνήμη ROM, η αναγνώριση της ταυτότητας κωδικοποιείται κατά τη διάρκεια της παραγωγής της και δεν επανεγγράφεται. Συμβάλει στην αποθήκευση των δεδομένων ασφαλείας, με ένα μοναδικό σειριακό αριθμό. Αντίθετα, τα ολοκληρωμένα κύκλωμα με επανεγγράψιμη μνήμη χρησιμοποιούνται για να αποθηκεύουν δεδομένα – πληροφορίες, όταν η ετικέτα βρίσκεται στην ακτίνα του αναγνώστη και παρουσιάζουν μεγαλύτερη ευελιξία, καθώς έχουν τη δυνατότητα τροποποίησης και προσθήκης πληροφοριών. Τέλος, τα ολοκληρωμένα κυκλώματα με μνήμη “WORM” προγραμματίζονται από τον οργανισμό που τα χρησιμοποιεί, χωρίς όμως να έχουν τη δυνατότητα της επανεγγραφής.

Τα δεδομένα που αποθηκεύονται στις ετικέτες αποτελούνται από ένα μοναδικό αναγνωριστικό και μπορούν, επίσης, να περιλαμβάνουν ένα λειτουργικό σύστημα, μία αποθήκη δεδομένων (πητική ή όχι) και έναν ηλεκτρονικό κώδικα προϊόντων (Electronic Product Code - EPC) Το μέγεθος των δεδομένων, που μια ετικέτα RFID έχει την δυνατότητα να υποθηκεύσει, καθορίζεται από τον εκάστοτε προμηθευτή αλλά και την ίδια την εφαρμογή, με ανώτερο όριο αποθήκευσης τα 2KB, χωρητικότητα αρκετή για να αποθηκευτούν τα απαραίτητα δεδομένα του κάθε αντικειμένου. Μια άλλη σημαντική κατηγοριοποίηση που μπορούμε να διακρίνουμε στις ετικέτες RFID σχετίζεται με την κατασκευή και την εφαρμογή τους. Δεδομένου ότι τα συστήματα RFID έχουν εφαρμογή σε διάφορους τομείς στην καθημερινή ζωή του σύγχρονου ανθρώπου, η κατασκευή των ετικετών RFID αλλάζει ανάλογα με τις εφαρμογές και τις ανάγκες που χρειάζεται κάθε φορά, να καλύψει.

Οι αναγνώστες RFID αποτελούνται από μία κεραία, η οποία αναλαμβάνει την επικοινωνία, μέσω ραδιοσυχνοτήτων, με τις ετικέτες καθώς και μία μονάδα ελέγχου, που εκτελεί δύο συγκεκριμένα έργα: πρωτίστως τον καθορισμό των διάφορων ενεργειών (αποστολή/ λήψη σημάτων, ανάγνωση/ εγγραφή ετικετών κ.ά.), ενέργεια που πραγματοποιείται μέσω του ενδιάμεσου λογισμικού, και δευτερευόντως την επικοινωνία με το πληροφοριακό σύστημα. Οι αναγνώστες RFID μπορούν να κατηγοριοποιηθούν σε σχέση με τις φυσικές τους διαστάσεις, την εφαρμογή τους και τις τεχνικές ιδιότητες σε "σταθερούς αναγνώστες", "ολοκληρωμένους αναγνώστες", "αναγνώστες χειρός" και σε "ενσωματωμένους αναγνώστες".

Οι εφαρμογές του RFID είναι τεράστιες, με κλασικό παράδειγμα τα προϊόντα που έχουν συρμάτινες ταινίες στις αλυσίδες καταστημάτων. Τα σημαντικά πλεονεκτήματα που προσφέρει το RFID είναι:

- Η αναγνώριση μπορεί να γίνει από απόσταση μιας και υπάρχουν RFID tags που είναι σε θέση παίρνοντας ενέργεια από κάποια πηγή που συνήθως είναι μπαταρία να στείλουν τις πληροφορίες στον δέκτη.
- Δυνατότητα αποθήκευσης περισσότερων δεδομένων σε σχέση με τα Bar Code
- Μπορούν να μην είναι ορατά στο ανθρώπινο μάτι τα RFID tags μιας και για την αναγνώριση τους δεν χρειάζεται οπτικό μέσο.
- Δυνατότητα προγραμματισμού εξ' αποστάσεως
- Επιπρόσθετες λειτουργίες. Π.χ. Παρακολούθηση και καταγραφή της θερμοκρασίας. (βλέπε [3])

1.3 GPS

Το GPS (Global Positioning System), είναι ένα παγκόσμιο σύστημα εντοπισμού θέσης, το οποίο βασίζεται σε ένα "πλέγμα" εικοσιτεσσάρων τεχνητών δορυφόρων της Γης, στους οποίους υπάρχουν ειδικές συσκευές, οι οποίες ονομάζονται "δέκτες GPS". Οι δέκτες αυτοί παρέχουν ακριβείς πληροφορίες για τη θέση ενός σημείου, το υψόμετρό του, την ταχύτητα και την κατεύθυνση της κίνησης του. Επίσης, σε συνδυασμό με ειδικό λογισμικό χαρτογράφησης μπορούν να απεικονίσουν γραφικά τις πληροφορίες αυτές.

Το σύστημα ξεκίνησε από το Υπουργείο Άμυνας των ΗΠΑ και ονομάστηκε "NAVSTAR GPS" (Navigation Signal Timing and Ranging Global Positioning System). Το δορυφορικό αυτό σύστημα ρυθμίζεται καθημερινά από τη Βάση Πολεμικής Αεροπορίας Σρίβερ (Shriver).

Για την καλύτερη κατανόηση της λειτουργίας του GPS, καθώς και για την ανάγκη ύπαρξης μιας συντονισμένης προσπάθειας τέτοιου βεληνεκούς, αξίζει να γίνει μια αναφορά στην ιστορική εξέλιξη αυτής της τεχνολογίας. Τα σημεία του ορίζοντα, ή ακόμη και τα αστέρια, χρησιμοποιούνταν από την αρχαιότητα για τον προσανατολισμό των ανθρώπων. Ένα σταθερό άστρο στον ουρανό, με γνωστή γεωγραφική θέση ως προς το σημείο παρατήρησης, αποτελούσε σημείο αναφοράς και βοηθούσε τους ανθρώπους στο να βρουν τη σωστή πορεία τους. Στον προσανατολισμό συνέβαλαν αργότερα και άλλα μέσα, όπως η πυξίδα και ο εξάντας. Ωστόσο ο εξάντας είναι εύχρηστος μόνο για τον προσδιορισμό του γεωγραφικού πλάτους, ενώ η χρήση του για τον προσδιορισμό του γεωγραφικού μήκους είναι δύσκολη και εξαιρετικά σύνθετη, πράγμα που αποτελεί ένα σημαντικό μειονέκτημα για προσδιορισμό του στίγματος στην θάλασσα. Ως αποτέλεσμα, τον 17ο αιώνα, το Ηνωμένο Βασίλειο συνέστησε ένα συμβούλιο επιστημόνων, το οποίο θα επιβράβευε χρηματικά όποιον θα μπορούσε να εφεύρει ένα όργανο, το οποίο θα επέτρεπε τον ακριβή υπολογισμό και των δύο γεωγραφικών συντεταγμένων, δηλαδή μήκους και πλάτους.

Το 1761, ο Άγγλος ωρολογοποιός Τζον Χάρισσον (John Harrison), ύστερα από προσπάθειες δώδεκα ετών, κατασκεύασε ένα όργανο, το οποίο δεν ήταν άλλο από το γνωστό σημερινό χρονόμετρο. Σε συνδυασμό με τον εξάντα, το χρονόμετρο επέτρεπε τον υπολογισμό του στίγματος των πλοίων με εξαιρετική ακρίβεια (για τα δεδομένα της εποχής). Πέρασαν αρκετά χρόνια μέχρι να δημιουργηθούν τα πρώτα συστήματα εντοπισμού θέσης που βασίζονταν σε ηλεκτρομαγνητικά κύματα (ραντάρ), στα μέσα του 20ού αιώνα. Τα συστήματα αυτά χρησιμοποιήθηκαν ευρύτατα κατά τη διάρκεια του Δευτέρου Παγκοσμίου Πολέμου (και χρησιμοποιούνται ακόμη). Τα συστήματα εντοπισμού θέσης της εποχής αποτελούνταν από ένα δίκτυο σταθμών βάσης και κατάλληλους δέκτες.

Ανάλογα με την ισχύ του σήματος που λάμβανε κάθε δέκτης από σταθμούς γνωστής γεωγραφικής θέσης, σχηματιζόνταν δύο ή περισσότερες συντεταγμένες, μέσω των οποίων προσδιοριζόταν η θέση των σημείων ενδιαφέροντος επάνω σε ένα χάρτη. Στην περίπτωση αυτή, όμως, υπήρχαν δύο διαφορετικά προβλήματα: Στην πρώτη περίπτωση η χρήση σταθμών βάσης,

που θα εξέπεμπαν σήμα σε υψηλή συχνότητα, διέθεταν μεν υψηλή ακρίβεια εντοπισμού, αλλά είχαν μικρή εμβέλεια. Στη δεύτερη περίπτωση συνέβαινε το ακριβώς αντίθετο, δηλαδή ο σταθμός βάσης χρησιμοποιούσε μεν χαμηλή συχνότητα εκπομπής σήματος, προσφέροντας έτσι υψηλότερη εμβέλεια, αλλά και η ακρίβεια που παρείχε ήταν χαμηλή.

Έστω και με αυτά τα προβλήματα, η αρχή της χρήσης ραδιοκυμάτων για τον εντοπισμό της θέσης ενός σημείου είχε ήδη γίνει. Το Global Positioning System στη σημερινή του μορφή βασίζεται σε παρεμφερή τεχνολογία. Συνδυάζει όλες τις μεθόδους που είχαν χρησιμοποιηθεί στον ουρανό, δηλαδή την τεχνολογία των ηλεκτρομαγνητικών κυμάτων καθώς και την παρατήρηση ενός –τεχνητού αυτή τη φορά– ουράνιου σώματος. Οι σταθμοί βάσης που λαμβάνουν και δέχονται τα απαραίτητα ηλεκτρομαγνητικά κύματα δεν είναι πλέον επίγειοι, αλλά εδρεύουν σε δορυφόρους.

Ένα δίκτυο πολυάριθμων (24 - 32) δορυφόρων που βρίσκεται σε σταθερή θέση γύρω από τον πλανήτη μας (Εικόνα 4), βοηθά τους δέκτες GPS να παράξουν το ακριβές στίγμα ενός σημείου οπουδήποτε στον κόσμο. Όταν, το 1957, πραγματοποιήθηκε η εκτόξευση του δορυφόρου Σπούτνικ, οι άνθρωποι είχαν ήδη αντιληφθεί ότι ένα τεχνητό ουράνιο σώμα κοντά στη Γη είναι



Εικόνα 4. Πλέγμα δορυφόρων GPS

δυνατό να χρησιμοποιηθεί για να εντοπιστεί η θέση ενός σημείου πάνω στον πλανήτη. Αμέσως μετά την εκτόξευσή του, οι ερευνητές του Ινστιτούτου Τεχνολογίας της Μασαχουσέτης (MIT) διαπίστωσαν ότι το σήμα που λαμβανόταν από τον δορυφόρο αυξανόταν καθώς αυτός πλησίαζε προς το επίγειο σημείο παρατήρησης και μειωνόταν όταν ο δορυφόρος απομακρυνόταν από αυτό. Με τον ίδιο τρόπο που η θέση ενός δορυφόρου μπορούσε να εντοπιστεί ανάλογα με την ισχύ του σήματος που λαμβάνεται από αυτόν, υπήρχε και η δυνατότητα να συμβεί το ακριβώς αντίθετο: Ο δορυφόρος να εντοπίσει την ενός σημείου θέση με ιδιαίτερη ακρίβεια. Στην πραγματικότητα ένας δορυφόρος δεν είναι αρκετός για να υπάρξουν ακριβή αποτελέσματα, αλλά απαιτούνται τουλάχιστον τρεις. Κάθε δορυφόρος μεταδίδει ένα ηλεκτρομαγνητικό σήμα - μία δέσμη μικροκυμάτων - που αναγγέλλει την παρουσία του σε οποιοδήποτε άτομο στη Γη που διαθέτει ένα δέκτη έτοιμο να λάβει το σήμα. Συνεπώς, ένας λήπτης GPS λαμβάνει ανά πάσα στιγμή σήματα από τέσσερις δορυφόρους. Ο ενσωματωμένος ηλεκτρονικός υπολογιστής χρησιμοποιεί αυτά τα σήματα για να υπολογίσει την ακριβή σας απόσταση από καθένα από τους τέσσερις δορυφόρους και στη συνέχεια να υπολογίσει την ακριβή σας θέση επί του πλανήτη με απόκλιση λίγων μέτρων βάσει αυτών των αποστάσεων. (βλέπε [5])

Στην πραγματικότητα απαιτούνται σήματα από τρεις μόνο δορυφόρους για τη διεξαγωγή αυτής της διαδικασίας τριπλευρισμού. Ο υπολογισμός της θέσης σας στη Γη βασίζεται στην απόστασή σας από τρεις δορυφόρους. Το σήμα του τέταρτου δορυφόρου είναι πλεονάζον και

χρησιμοποιείται για την επιβεβαίωση των αποτελεσμάτων του αρχικού υπολογισμού. Εάν η θέση που υπολογίζεται βάσει των αποστάσεων από τους δορυφόρους A-B-Γ δεν ταυτίζεται με τον υπολογισμό βάσει των στοιχείων των δορυφόρων A-B-Δ, τότε ελέγχονται άλλοι συνδυασμοί μέχρι να προκύψει ένα συνεκτικό αποτέλεσμα.

Η διαδικασία της μέτρησης της απόστασης μεταξύ δορυφόρου και δέκτη GPS βασίζεται σε χρονισμένα σήματα. Για παράδειγμα, ακριβώς στις 16:45, οι δορυφόροι μπορεί να αρχίσουν να μεταδίδουν το σήμα τους. Ο δέκτη GPS θα αρχίσει επίσης να επεξεργάζεται την ίδια ακολουθία στις 16:45 τοπική ώρα, αλλά δεν τη μεταδίδει. Όταν ο δέκτη λάβει το σήμα από τους διάφορους δορυφόρους, θα προκύψει μία χρονική υστέρηση, επειδή τα μικροκύματα χρειάζονται ένα κλάσμα του δευτερολέπτου για να διανύσουν με την ταχύτητα του φωτός την απόσταση μεταξύ δορυφόρου και δέκτη. Η χρονική υστέρηση μετατρέπεται εύκολα στην απόσταση προς κάθε δορυφόρο. Οι μικρές διαφορές μεταξύ των σημάτων κάθε δορυφόρου χρησιμοποιούνται στη συνέχεια για τον υπολογισμό της θέσης του δέκτη. (βλέπε [6])

Το σύστημα εντοπισμού θέσης GPS σχηματίζει ένα παγκόσμιο δίκτυο, με εμβέλεια που καλύπτει ξηρά, θάλασσα και αέρα. Εξαιτίας αυτής της έκτασής του, είναι απαραίτητος ο διαχωρισμός του σε επιμέρους τμήματα όπου πραγματοποιούνται όλες οι λειτουργίες του αλλά και ο συντονισμός του. Αναλυτικά, τα τμήματα αυτά είναι:

- Διαστημικό τμήμα: Αποτελείται από το δίκτυο των 24 - 32 δορυφόρων που ήδη αναφέραμε. Οι δορυφόροι αυτοί «σκεπάζουν» ομοίμορφα με το σήμα τους ολόκληρο τον πλανήτη, γεγονός που αποδεικνύει τη φιλοσοφία που κρύβεται πίσω από τη λειτουργία του συστήματος GPS, δηλαδή τη διαθεσιμότητά του σε κάθε σημείο της Γης, ώστε να μην υπάρχει κίνδυνος να αποπροσανατολιστεί κανείς ποτέ και πουθενά. Όλοι οι δορυφόροι βρίσκονται σε ύψος 12.552 μιλίων (20.200 χιλιομέτρων) πάνω από την επιφάνεια της θάλασσας και εκτελούν δύο περιστροφές γύρω από τη Γη κάθε 24ωρο. Η κατασκευάστρια εταιρεία είναι η Rockwell International, η εκτόξευσή τους πραγματοποιήθηκε από το ακρωτήριο Canaveral, ενώ η τροφοδοσία τους με ηλεκτρική ενέργεια πραγματοποιείται μέσω των ηλιακών στοιχείων που διαθέτουν (Εικόνα 5).



Εικόνα 5. Δορυφόρος GPS

- Επίγειο τμήμα ελέγχου: Οι δορυφόροι, όπως είναι αναμενόμενο, είναι πολύ πιθανό να αντιμετωπίσουν ανά πάσα στιγμή προβλήματα στη σωστή λειτουργία τους. Οι έλεγχοι που πραγματοποιούνται σε αυτούς αφορούν στη σωστή τους ταχύτητα και υψόμετρο και στην κατάσταση της επάρκειάς τους σε ηλεκτρική ενέργεια. Παράλληλα, εφαρμόζονται όλες οι διορθωτικές ενέργειες που αφορούν στο σύστημα χρονομέτρησης των δορυφόρων, ώστε να αποτρέπεται η παροχή λανθασμένων πληροφοριών στους χρήστες του συστήματος. Το τμήμα επίγειου ελέγχου αποτελείται από ένα επανδρωμένο και τέσσερα μη επανδρωμένα κέντρα, εγκατεστημένα σε ισάριθμες περιοχές του πλανήτη. Οι περιοχές αυτές είναι οι εξής: α) Κολοράντο (ΗΠΑ) β) Χαβάη (Ανατολικός Ειρηνικός Ωκεανός) γ) Ascension Island (Ατλαντικός Ωκεανός) δ) Diego Garcia (Ινδικός Ωκεανός) ε) Kwajalein (Δυτικός Ειρηνικός Ωκεανός). Ο κυριότερος σταθμός βάσης είναι αυτός του Κολοράντο, ο οποίος είναι μάλιστα και ο μοναδικός που βρίσκεται στην ξηρά. Αναλαμβάνει τον έλεγχο της σωστής λειτουργίας των εναπομεινάντων τεσσάρων σταθμών, καθώς και τον συντονισμό τους. Σημειώνοντας τη θέση των σταθμών αυτών πάνω σε έναν παγκόσμιο χάρτη, παρατηρεί κανείς ότι η διάταξή τους δεν είναι τυχαία, αλλά ακολουθούν μια γραμμή παράλληλη με τα γεωγραφικά μήκη της Γης. (Εικόνα 6)



Εικόνα 6. Επίγεια κέντρα ελέγχου GPS

- Το τμήμα τελικού χρήστη: Απαρτίζεται από τους χιλιάδες δέκτες χρηστών GPS ανά την υφήλιο. Οι δέκτες αυτοί μπορούν να χρησιμοποιηθούν τόσο κατά τη διάρκεια μιας απλής πεζοπορίας, όσο και σε οχήματα ή θαλάσσια σκάφη και κατά κανόνα διαθέτουν αρκετά μικρές διαστάσεις. Για να προσφέρουν όσο το δυνατόν περισσότερες πληροφορίες, οι δέκτες συνδυάζονται με ειδικό λογισμικό, που προβάλλει ένα χάρτη στην οθόνη της συσκευής GPS. Πρόκειται, δηλαδή, για λογισμικό που λαμβάνει από τους δορυφόρους τις πληροφορίες για το στίγμα του σημείου στο οποίο βρίσκεται ο δέκτης και τις μετατρέπει σε κατανοητή «ανθρώπινη» μορφή, πληροφορώντας το χρήστη για την ακριβή γεωγραφική του θέση.

Λόγω της φύσης της παρούσας εργασίας, κρίνεται σημαντική μια περεταίρω εμβάθυνση στα δομικά στοιχεία καθώς και τη λειτουργία του φορητού δέκτη GPS.

Φορητός δέκτης GPS

Ένας φορητός δέκτης αποτελείται από:

- Την εσωτερική δορυφορική κεραία, η οποία λαμβάνει το σήμα GPS από τους δορυφόρους με τους οποίους έχει οπτική επαφή. Επίσης, λαμβάνει σήμα και από ανακλάσεις, π.χ. σε τοίχους, κάνοντας δυνατή τη λήψη σε δρόμους που περιβάλλονται από πολύ ψηλά κτήρια (στην καθιερωμένη αγγλόφωνη σχετική ορολογία, οι συνθήκες αυτές αποκαλούνται "urban canyon") ή ακόμη και σε κάποιους εσωτερικούς χώρους. Πάντως, αρκετοί δέκτες διαθέτουν υποδοχή για εξωτερική κεραία. Οι εξωτερικές δορυφορικές κεραίες διαθέτουν πάντα προενισχυτή και δίνουν καλύτερη λήψη, λόγω της δυνατότητας τοποθέτησης πάνω από πιθανά εμπόδια (π.χ. στην οροφή του αυτοκινήτου) και της ενίσχυσης που διαθέτουν (στις εσωτερικές κεραίες δεν μπορεί να χρησιμοποιηθεί επιπλέον στάδιο προενίσχυσης, καθώς αυτό θα οδηγούσε σε ανεπιθύμητη ανάδραση, λόγω της γειννίας με το αναλογικό τμήμα του δέκτη).
- Τον κυρίως δέκτη GPS ο οποίος χρησιμοποιεί κυκλώματα εξαιρετικά χαμηλού θορύβου και ειδικές τεχνικές επεξεργασίας σήματος ώστε να ξεχωρίζει τα εξαιρετικά ασθενή σήματα από τους δορυφόρους, από τον ισχυρό τηλεπικοινωνιακό θόρυβο ο οποίος έχει τη μορφή τυχαίου σήματος. Ο κυρίως δέκτης αποτελείται από το αναλογικό τμήμα εισόδου και το ψηφιακό, το οποίο περιέχει σύνθετο ψηφιακό υλικό (hardware), συνήθως κάποιο εξειδικευμένο ολοκληρωμένο κύκλωμα τύπου ASIC και μικροελεγκτή (microcontroller) χαμηλής κατανάλωσης ισχύος. Αυτό το hardware χρησιμοποιεί λογισμικό με πολύ εξελιγμένους αλγόριθμους επεξεργασίας, για να μπορέσει να εξάγει χρήσιμο στίγμα σε συνθήκες urban canyon ή δύσκολης λήψης εν γένει. Σε τέτοιες περιπτώσεις, η ακρίβεια λήψης, λόγω των πολλαπλών σημάτων, τα οποία λαμβάνει η κεραία από τον ίδιο δορυφόρο, με χρονική καθυστέρηση μεταξύ τους (φαινόμενο ηχούς), μπορεί να υποβαθμίσει σημαντικά την ακρίβεια θέσης. Το αποτέλεσμα εξαρτάται έντονα από την ποιότητα των αλγορίθμων και βελτιώνεται σημαντικά από την μία γενιά δεκτών στην επόμενη. Η τελική έξοδος του δέκτη είναι το στίγμα (θέση) του και η ακριβής παγκόσμια ώρα UMT. Αυτά τα δύο δεδομένα, μαζί με άλλες χρήσιμες πληροφορίες όπως ο αριθμός των λαμβανόμενων δορυφορικών σημάτων και η στάθμη τους, αποστέλλονται σε μια θύρα επικοινωνίας του δέκτη, συνήθως σειριακής μορφής, δηλαδή ασύγχρονη (UART) ή σύγχρονη (π.χ. SPI). Ο ρυθμός με τον οποίο βγαίνει νέο στίγμα στην έξοδο του δέκτη είναι συνήθως 1 φορά το δευτερόλεπτο (δηλαδή 1 Hz), αν και υπάρχουν δέκτες που μπορούν να δίνουν στίγμα με ταχύτερους ρυθμούς (π.χ. 10 Hz). Σε συσκευές που λειτουργούν με μπαταρία, ο κυρίως δέκτης GPS διαθέτει και καταστάσεις λειτουργίας όπου ο ρυθμός αποστολής στίγματος μειώνεται σημαντικά, με αποτέλεσμα την εξοικονόμηση ισχύος.
- Τον κυρίως μικροελεγκτή, την οθόνη απεικόνισης (συνήθως υγρών κρυστάλλων) και το υπόλοιπο hardware επικοινωνία με το χρήστη της συσκευής. Ο μικροελεγκτής αυτός, μέσω του ενσωματωμένου λογισμικού του, επεξεργάζεται το στίγμα που λαμβάνει από τον

κυρίως δέκτη GPS, μέσω της αντίστοιχης σειριακής του θύρας. Το αποτέλεσμα της επεξεργασίας είναι μια πιο κατανοητή για τον άνθρωπο μορφή του στίγματος, και συνήθως εμφανίζεται σε οθόνη με δυνατότητες γραφικών, πάνω σε ψηφιακό χάρτη, μαζί με άλλες πληροφορίες όπως ώρα, υψόμετρο και ταχύτητα κίνησης. Η ακρίβεια του ενσωματωμένου χάρτη μπορεί να είναι αρκετά μεγάλη, στα ακριβότερα μοντέλα, ενώ συχνά υπάρχει η δυνατότητα αναβάθμισης ή επαύξησής του μέσω σύνδεσης με προσωπικό υπολογιστή (PC). (βλέπε [5])

1.4 3G/GPRS

Το 3G είναι τα αρχικά των λέξεων 3rd Generation και αποτελεί ένα γενικό όρο ο οποίος αναφέρεται στην τρίτη γενιά τεχνολογίας κινητής τηλεφωνίας. Ως γενιά χαρακτηρίζεται το σύνολο των ασύρματων τεχνολογιών που επιτρέπουν τη μετάδοση φωνής ή και δεδομένων στα δίκτυα κινητής τηλεφωνίας. Μεταξύ των τεχνολογιών αυτών είναι οι WCDMA, CDMA2000, UMTS και EDGE. Το WCDMA ή Wideband Code Multiple Division Access, έχει ήδη επιλεγεί ως το "σύστημα" τρίτης γενιάς, που θα χρησιμοποιηθεί στην Ευρώπη, Ιαπωνία και στις ΗΠΑ.

Ως πρώτη γενιά (1G) χαρακτηρίστηκε η αναλογική τεχνολογία κινητής τηλεφωνίας, ενώ ως δεύτερη γενιά (2G) η ψηφιακή τεχνολογία που ακολούθησε. Τεχνολογίες όπως το GSM, TDMA και CDMA αποτελούν τρέχουσες τεχνολογίες μετάδοσης δεδομένων στα δίκτυα δεύτερης γενιάς. Υπάρχουν βέβαια και τεχνολογίες που προσφέρουν υψηλότερες ταχύτητες στα δίκτυα δεύτερης γενιάς όπως αυτή του GPRS που αποκαλείται 2.5G αλλά και η πολλά υποσχόμενη τεχνολογία EDGE που έχει ήδη ονομαστεί χαρακτηριστικά 2.75G, υποδηλώνοντας μια ενδιάμεση μεταβατική γενιά πριν το 3G. (βλέπε [7])

Η λειτουργία της κινητής τηλεφωνίας στην Ελλάδα, όπως άλλωστε και στις περισσότερες χώρες του κόσμου, στηρίζεται στο σύστημα GSM (Global System for Mobile Communications), το οποίο δημιουργήθηκε το 1987 στο πλαίσιο της Ενιαίας Ευρωπαϊκής Πολιτικής για τις Τηλεπικοινωνίες. Το Ευρωπαϊκό σύστημα GSM καθορίζει τις ζώνες συχνοτήτων, που είναι τα 900 και 1800MHz, καθώς και τα πρωτόκολλα για τη μετάδοση του τηλεπικοινωνιακού σήματος κινητής τηλεφωνίας.

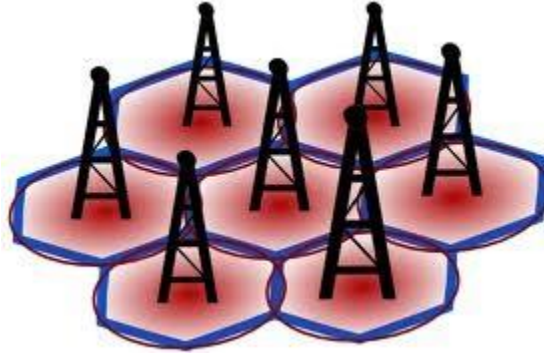
Με την εξέλιξη της τεχνολογίας και την αύξηση των τηλεπικοινωνιακών αναγκών, οι εταιρείες κινητής τηλεφωνίας ξεκίνησαν να παρέχουν και υπηρεσίες κινητής τηλεφωνίας τρίτης γενιάς, οι οποίες συνδυάζουν τις τηλεπικοινωνίες με την πληροφορική και λειτουργούν στο φάσμα συχνοτήτων 2000MHz. Οι υπηρεσίες κινητής τηλεφωνίας 3ης γενιάς στηρίζονται στο τηλεπικοινωνιακό πρωτόκολλο UMTS (Universal Mobile Telecommunications System). Η δομή και η λειτουργία των δικτύων 3ης γενιάς είναι όμοια με αυτή των δικτύων 2ης γενιάς που λειτουργούν στην Ελλάδα από το 1993. (βλέπε [8])

Σε αντίθεση με προηγούμενες τεχνολογίες που βασίζονται σε μετάδοση δεδομένων πάνω από απευθείας σύνδεση των δύο μερών (circuit-switched), οι τεχνολογίες που ανήκουν στην ομάδα τρίτης γενιάς βασίζονται σε υψηλής ταχύτητας μετάδοση δεδομένων μοιρασμένων σε πακέτα (packet-switched). Η τεχνολογία στηρίζεται στα γνωστά δίκτυα GSM με μια παραλλαγή του CDMA με το όνομα WCDMA (Wideband-CDMA) η οποία είναι ικανή να επιτύχει ταχύτητες μετάδοσης έως και 2Mbps! Στα υπάρχοντα δίκτυα τρίτης γενιάς όμως, ο συνδυασμός του WCDMA με τις υπό διάθεση συσκευές είναι ικανός να προσφέρει στον τελικό χρήστη ταχύτητες έως και 384Kbps, οι οποίες όμως είναι αρκετές για να μετατρέψουν το κινητό σε μια ασύρματη συσκευή πολυμέσων.

Τα σημαντικότερα πλεονεκτήματα της τεχνολογίας τρίτης γενιάς παρατίθενται παρακάτω:

- Οι βίντεο-κλήσεις είναι χωρίς αμφιβολία μια από τις πιο πολυσυζητημένες υπηρεσίες των δικτύων 3G. Πλέον, εκτός από το να ακούει ο χρήστης τον συνομιλητή του, μπορεί και να τον βλέπει ζωντανά στην οθόνη του κινητού του. Φυσικά, θα πρέπει να έχουν και οι δυο κάποια συμβατή συσκευή. Αξίζει πάντως να σημειωθεί ότι από το 2004 που τα δίκτυα τρίτης γενιάς μέχρι και τις μέρες μας, που ξεκινά η εμπορική διάθεση υπηρεσιών τέταρτης γενιάς (LTE-4G), οι βίντεο-κλήσεις δεν έγιναν αποδεκτές από την πλειοψηφία των χρηστών, παρόλο που πλέον οι συσκευές τηλεπικοινωνίας τις υποστηρίζουν σχεδόν στο σύνολό τους.
- Οι υψηλές ταχύτητες ασύρματης μεταφοράς δεδομένων είναι ένα ακόμη από τα πλεονεκτήματα των δικτύων 3G. Η σύνδεση στο Internet εκτός από άμεση και απρόσκοπτη, δίνει στον χρήστη πλέον και ταχύτητες που φθάνουν τα 384kbps - ανάλογες δηλαδή με αυτές της σταθερής τεχνολογίας xDSL.
- Οι υψηλές ταχύτητες μεταφοράς δεδομένων βοηθούν αρκετά στην πιο γρήγορη και άμεση χρήση διαφόρων multimedia εφαρμογών. Έτσι, ένα MMS σε κάποιον άλλο συνδρομητή αποστέλλεται σε λιγότερο από 10 δευτερόλεπτα - όταν στα δίκτυα 2G ο χρόνος αυτός μπορούσε να ξεπεράσει και το 1 λεπτό.
- Το video-streaming είναι μια ακόμη από τις υπηρεσίες που παρέχουν τα δίκτυα 3G. Το αυξημένο bandwidth επιτρέπει τη μετάδοση σε πραγματικό χρόνο, κινούμενης εικόνας και ήχου υψηλής ανάλυσης. Έτσι, ο χρήστης μπορεί να παρακολουθήσει τηλεοπτικά προγράμματα, ζωντανά ή μαγνητοσκοπημένα, ανεξαρτήτως τόπου και χρόνου.
- Υψηλής ποιότητας παιχνίδια, τα οποία μπορούν να παίζονται online σε πραγματικό χρόνο και ταυτόχρονα με άλλους παίκτες.
- Υπηρεσίες εύρεσης θέσεως, σε συνδυασμό με την τεχνολογία GPS, οι οποίες παρέχουν χάρτες τη περιοχής που βρίσκεται ο χρήστης, εύρεση βέλτιστης διαδρομής προς τον προορισμό του, γειτονικά σημεία ενδιαφέροντος και άλλα. (βλέπε [7])

Το GSM/UMTS είναι ένα σύστημα μετάδοσης τηλεπικοινωνιακών σημάτων με την χρήση ηλεκτρομαγνητικών κυμάτων. Η λειτουργία του βασίζεται σε μία αλληλουχία κεραιών, οι οποίες ονομάζονται Σταθμοί Βάσης. Η διάταξη των Σταθμών Βάσης δημιουργεί μία σειρά από κυψέλες, δηλαδή περιοχές που η κάθε μία καλύπτεται από την εμβέλεια ενός συγκεκριμένου σταθμού βάσης. Έτσι, οι εφαπτόμενες κυψέλες έχουν διαφορετική συχνότητα εκπομπής, ώστε να προλαμβάνουν τυχόν παρεμβολές και τα επίπεδα εκπομπής ισχύος διατηρούνται στο ελάχιστο δυνατό επίσης για να μη δημιουργούν παρεμβολές με μη εφαπτόμενες κυψέλες που χρησιμοποιούν την ίδια συχνότητα. (Εικόνα 7)



Εικόνα 7. Σταθμοί βάσης - κυψέλες

Ένας τυπικός σταθμός βάσης λειτουργεί με ισχύ περίπου 60 Watt. Για το λόγο αυτό, το GSM ονομάζεται και κυψελοειδές ή κυψελωτό σύστημα. Το μέγεθος μιας κυψέλης σχετίζεται και με τον αναμενόμενο αριθμό των χρηστών κινητών τηλεφώνων κάθε περιοχής. Έτσι, σε αραιοκατοικημένες περιοχές, όπως οι αγροτικές, οι κυψέλες είναι μεγάλες με διάμετρο έως και 35χλμ. Αντίθετα, σε πυκνοκατοικημένες περιοχές, όπως οι μεγαλουπόλεις, οι κυψέλες είναι μικρές με διάμετρο που μπορεί να φτάνει μόνο τα 300 m.

Όταν αυξηθεί πολύ ο αριθμός των κινητών τηλεφώνων που χρησιμοποιούνται μέσα σε μία κυψέλη ο Σταθμός Βάσης δεν επαρκεί για την εξυπηρέτηση όλων των χρηστών και παρουσιάζονται φαινόμενα υπερφόρτωσης. Τότε, γίνεται η λεγόμενη υποδιαίρεση των κυψελών, χωρίζονται δηλαδή αυτές σε μικρότερες και καλύπτονται από Σταθμούς Βάσης με κατά κανόνα μικρότερη ισχύ. Κατά συνέπεια, όσο περισσότεροι Σταθμοί Βάσης είναι τοποθετημένοι σε μία δεδομένη περιοχή, τόσο μικρότερη είναι η ισχύς λειτουργίας του κάθε Σταθμού, ενώ ανάλογα μικρότερη είναι και η ισχύς εκπομπής των κινητών τηλεφώνων που επικοινωνούν με τους σταθμούς αυτούς που βρίσκονται πιο κοντά στο χρήστη.

Κάθε φορά που ο χρήστης καλεί ή δέχεται κλήση μέσω κινητού τηλεφώνου, η μετάδοση των τηλεπικοινωνιακών σημάτων γίνεται με τη χρήση ηλεκτρομαγνητικών κυμάτων που εκπέμπονται από το Σταθμό Βάσης και από το κινητό του τηλέφωνο. Και τα δύο εκπέμπουν και λαμβάνουν ηλεκτρομαγνητικά κύματα. Για να λειτουργήσει λοιπόν η κινητή τηλεφωνία χρειάζεται σταθερούς Σταθμούς Βάσης και κινητά τηλέφωνα. Αυτές οι πληροφορίες, χρησιμοποιούν ως «μέσο μεταφοράς» ηλεκτρομαγνητικά κύματα που εκπέμπονται και λαμβάνονται από τα κινητά τηλέφωνα και τους σταθερούς Σταθμούς Βάσης.

Ας υποθέσουμε ότι ο χρήστης θέλει να επικοινωνήσει μέσω του κινητού του τηλεφώνου με κάποιον που βρίσκεται στο δρόμο και έχει και αυτός κινητό τηλέφωνο. Μόλις πληκτρολογήσει τον αριθμό κλήσης, αυτόματα το κινητό τηλέφωνό του αποκαθιστά επαφή με τον πλησιέστερο Σταθμό Βάσης. Χρησιμοποιεί δηλαδή, τις διαθέσιμες συχνότητες, οι οποίες αποτελούν το μέσο μεταφοράς του σήματος με το αίτημά του, να βρεθεί η επαφή με την οποία θέλει να επικοινωνήσει.

Η ηλεκτρομαγνητική ενέργεια που εκπέμπει κάθε κινητό τηλέφωνο κατά τη διάρκεια μιας κλήσης μειώνεται ή αυξάνεται ανάλογα με το πόσο δυνατό είναι το σήμα, δηλαδή ανάλογα με την απόσταση του κινητού τηλεφώνου από τον πιο κοντινό σταθμό βάσης.

Όταν το κινητό τηλέφωνο βρίσκεται σε μεγάλη απόσταση από τον σταθμό βάσης, δεν έχει δηλαδή καλό σήμα, τότε εκπέμπει με τη μέγιστη ισχύ του, που είναι 250mW. Αν όμως υπήρχε σταθμός βάσης κοντά στο κινητό τηλέφωνο, τότε η συσκευή θα λειτουργούσε με μέση ισχύ 62,5mW, τιμή πολύ μικρότερη της μέγιστης. Άρα, ένα πυκνό δίκτυο σταθμών κινητής τηλεφωνίας εξασφαλίζει ότι τα κινητά τηλέφωνα λειτουργούν με την ελάχιστη δυνατή ισχύ και οι χρήστες δέχονται το μικρότερο ποσοστό ηλεκτρομαγνητικής ενέργειας από τη συσκευή τους.

Ο Σταθμός Βάσης κινητής τηλεφωνίας λαμβάνει και μεταβιβάζει το σήμα με το αίτημα του χρήστη για αναζήτηση και εντοπισμό της επαφής του στο τηλεπικοινωνιακό κέντρο της εταιρίας, με το δίκτυο της οποίας είμαστε συνδεδεμένοι. Το κέντρο της εταιρείας εντοπίζει την κυψέλη στην οποία βρίσκεται η επαφή και στέλνει το σήμα στον πλησιέστερο σε αυτόν Σταθμό Βάσης. Από εκεί, πάλι με τη χρήση των διαθέσιμων συχνοτήτων, στέλνεται το σήμα στο κινητό του τηλεφώνου και έτσι μπορεί να επικοινωνήσει μαζί του. (βλέπε [8])

Κεφάλαιο 2

2.1 Η κατασκευή

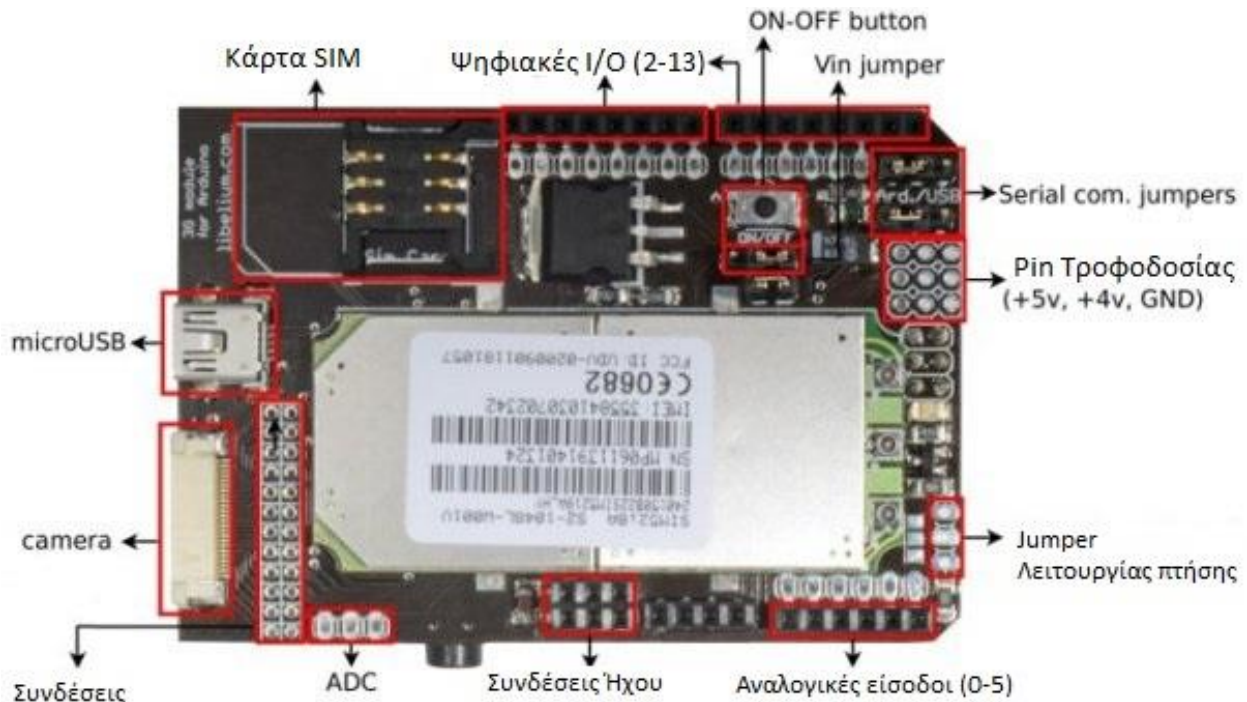
Για την κατασκευή της παρούσας πτυχιακής εργασίας χρησιμοποιήθηκαν τα παρακάτω υλικά:

1) Arduino Mega 2560

Είναι το θεμελιώδες στοιχείο της κατασκευής. Για εκτενή παρουσίασή του, καθώς και φωτογραφία της συσκευής, μπορείτε να ανατρέξετε στο προηγούμενο κεφάλαιο (Κεφάλαιο 1.1, σελ. 6)

2) 3G/GPS Shield

Το shield αυτό είναι κατασκευασμένο από την Cooking Hacks. Δίνει τη δυνατότητα στο Arduino να λειτουργήσει ταυτόχρονα σαν ένας δέκτης GPS και σαν κινητό τηλέφωνο χρησιμοποιώντας δίκτυο κινητής τηλεφωνίας τρίτης γενιάς. (Εικόνα 8)



Εικόνα 8. Cooking Hacks 3G/GPS Shield για Arduino

Το shield αυτό, σαν πλήρης εξομοιωτής κινητού τηλεφώνου, διαθέτει pin για σύνδεση μικροφώνου, μεγαφώνου και ηχείου (Συνδέσεις ήχου). Επίσης, παρέχει την δυνατότητα σύνδεσης κάμερας για καταγραφή εικόνων, βίντεο καθώς και για πραγματοποίηση βιντεοκλήσεων. Φυσικά, για την λειτουργία του απαιτείται κάρτα SIM από οποιαδήποτε εταιρεία κινητής τηλεφωνίας, όπως επίσης και χρήση εξωτερικών κεραιών. Τέλος παρέχεται στον χρήστη η δυνατότητα να χρησιμοποιήσει και κάρτα SD, μεγέθους ως 2 GB, για την αποθήκευση των δεδομένων του.

Η σύνδεση με το arduino είναι η πλέον εύκολη καθώς το shield «κουμπώνει» επάνω στο Arduino. Το shield δεσμεύει τα δυο pin σειριακής επικοινωνίας του Arduino (TX-RX), και μεταφέρει όλα τα υπόλοιπα στην επιφάνειά του καθιστώντας τα διαθέσιμα για τον χρήστη. (Εικόνα 9)



Εικόνα 9. 3G/GPS Shield “Κουμπωμένο” πάνω σε Arduino

3) RFID Module

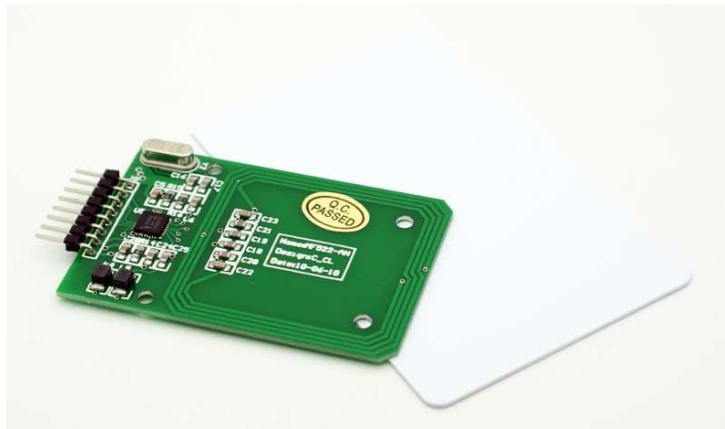
Το RFID module που επελέγη για την εφαρμογή είναι της B2CQSHOP και λειτουργεί στα 13.56 MHz. (Εικόνα 10). Για την σύνδεση με το Arduino χρησιμοποιεί τον διάλογο SPI (Serial Peripheral Interface). Το SPI είναι ένα σύγχρονο σειριακό πρωτόκολλο δεδομένων το οποίο χρησιμοποιείται από μικροελεγκτές για να επικοινωνούν με μια ή περισσότερες συσκευές, με μεγάλη ταχύτητα και σε μικρές αποστάσεις. Σε μια σύνδεση SPI υπάρχει πάντα μια Master συσκευή (συνήθως ο μικροελεγκτής) η οποία ελέγχει τις περιφερειακές συσκευές. Όλες οι συσκευές έχουν τρεις κοινές γραμμές συνδεδεμένες:

- **MISO** (Master In, Slave Out) – Η γραμμή με την οποία η Slave συσκευή στέλνει δεδομένα στην Master συσκευή
- **MOSI** (Master Out, Slave In) – Η γραμμή με την οποία η Master συσκευή στέλνει δεδομένα στην Slave συσκευή
- **SCK** (Serial Clock) – Οι παλμοί ρολογιού που συγχρονίζουν την μεταφορά δεδομένων

Μια γραμμή είναι συγκεκριμένη για κάθε συσκευή:

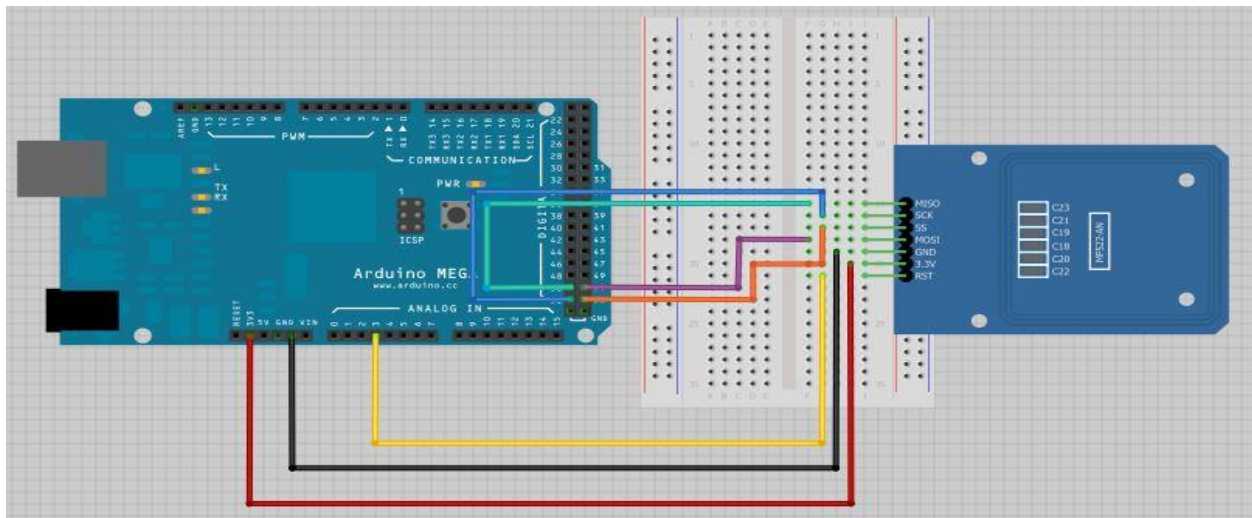
- **SS** (Serial Clock) – Το pin σε κάθε Slave συσκευή το οποίο η Master συσκευή χρησιμοποιεί για να την ενεργοποιήσει / απενεργοποιήσει. (βλέπε [9])

Κάθε πλακέτα Arduino έχει συγκεκριμένα pin τα οποία μπορούν να χρησιμοποιηθούν σαν MISO, MOSI SCK και SS. Στην πλακέτα Arduino Mega 2560 που χρησιμοποιήθηκε στην κατασκευή, τα pin αυτά είναι: MOSI → Pin 51, MISO → Pin 50, SCK → Pin 52 και SS → Pin 53



Εικόνα 10. RFID Module και κάρτα ανάγνωσης δεδομένων

Το RFID Module που χρησιμοποιήθηκε στην κατασκευή έχει 7 pin εισόδου/εξόδου. Η σύνδεση με τον Arduino φαίνεται στο παρακάτω διάγραμμα:



Εκτός από τα 4 Pin αναφέρθηκαν προηγουμένως, το RFID Module διαθέτει ακόμα 3 pin: τα pin GND και 3.3V συνδέονται στα αντίστοιχα pin τροφοδοσίας του Arduino ενώ το pin RST συνδέεται με το pin 3 του Arduino.

4) Οθόνη LCD

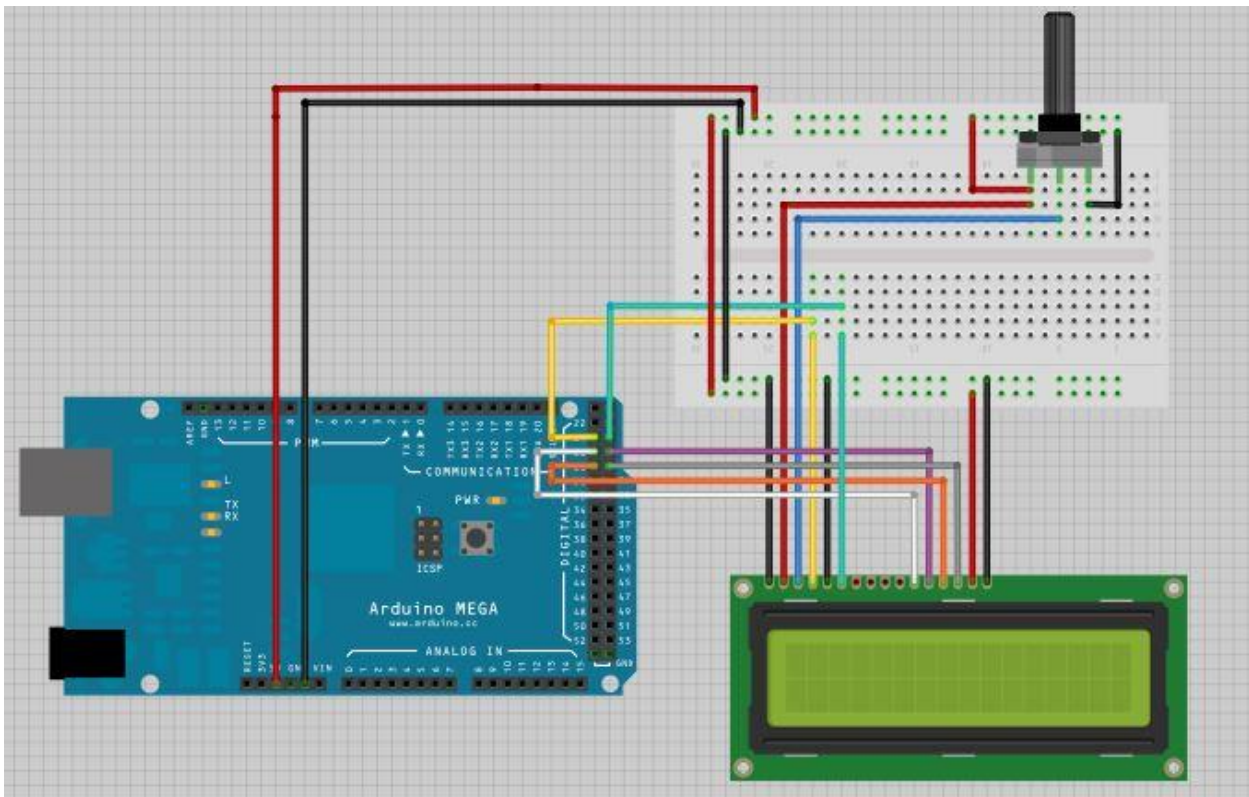
Στην κατασκευή χρησιμοποιήθηκε μια οθόνη LCD 2 γραμμών με 16 στοιχεία σε κάθε γραμμή. (Εικόνα 11).



Εικόνα 11. Οθόνη LCD 2x16

Η οθόνη έχει δεκαέξι pin εισόδου/εξόδου από τα οποία, λόγω των απαιτήσεων της εργασίας συνδέθηκαν μόνο τα δώδεκα. Τα pin GND και RW συνδέονται στην γείωση, τα Pin RS και E συνδέονται αντίστοιχα στα pin 24 και 25 του Arduino. Το Pin V_0 συνδέεται στη μεσαία λήψη ενός ποτενσιόμετρου του οποίου οι δυο ακριανές λήψεις συνδέονται στα pin τροφοδοσίας του Arduino (5V και GND). Με αυτόν τον τρόπο επιτυγχάνεται η ρύθμιση της φωτεινότητας της LCD οθόνης με απλή περιστροφή του άξονα του ποτενσιόμετρου.

Η σύνδεση της LCD οθόνης με το Arduino φαίνεται στο παρακάτω διάγραμμα:



5) Αριθμητικό πληκτρολόγιο

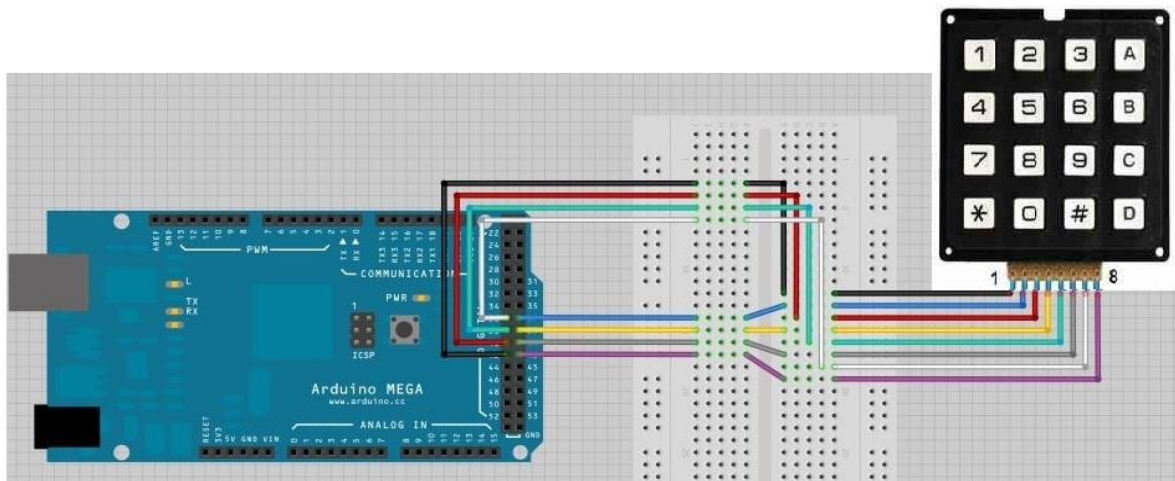
Για τις ανάγκες της κατασκευής χρησιμοποιήθηκε ένα αριθμητικό πληκτρολόγιο 16 στοιχείων (Εικόνα 12). Κάθε στοιχείο λειτουργεί σαν ανοιχτός διακόπτης. Η συσκευή δεν



Εικόνα 12. Αριθμητικό Πληκτρολόγιο

χρειάζεται κάποια τροφοδοσία και συνδέεται με το Arduino μέσω 8 Pin. Κάθε pin αντιστοιχεί σε μια γραμμή ή στήλη. Η λειτουργία του είναι αρκετά απλή. Μόλις ο χρήστης πιέσει ένα πλήκτρο τότε ο ανοικτός διακόπτης κλείνει, επιτρέποντας να περάσει ρεύμα στα δυο pin που αντιστοιχούν στην γραμμή και τη στήλη του πλήκτρου που πατήθηκε. Με αυτόν τον τρόπο αντιλαμβάνεται ο μικροελεγκτής ποιο πλήκτρο πίεσε ο χρήστης.

Η σύνδεση του αριθμητικού πληκτρολογίου με το Arduino φαίνεται στο παρακάτω διάγραμμα:



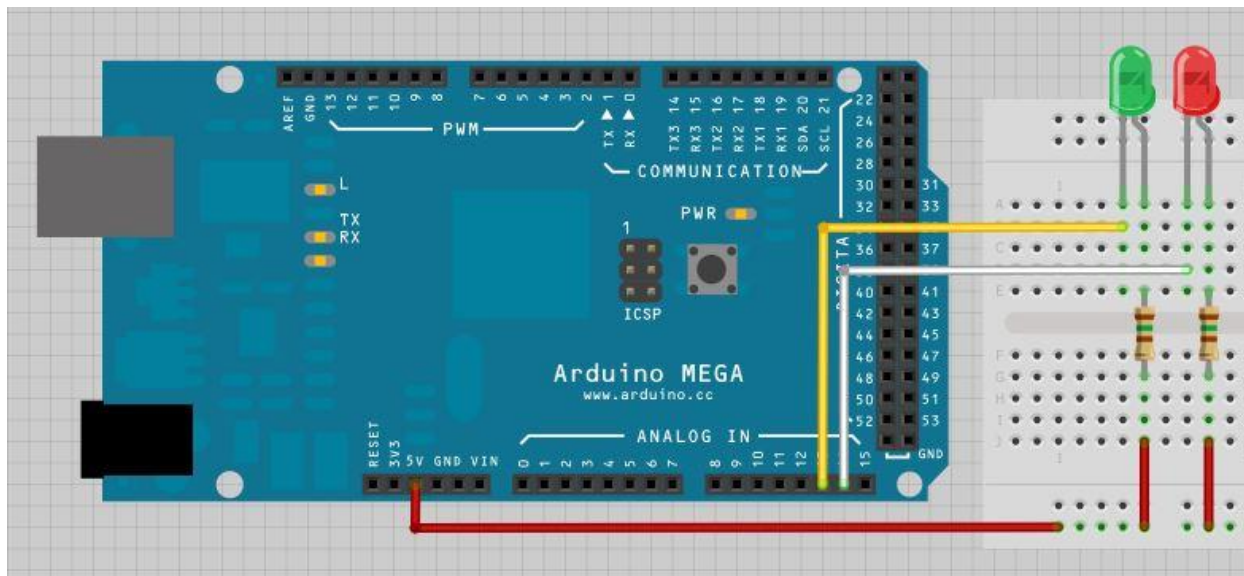
6) Διακόπτης 3^{ων} θέσεων με κλειδί

Στην κατασκευή, ο διακόπτης 3^{ων} θέσεων με κλειδί που χρησιμοποιήθηκε (Εικόνα 13), παίζει το ρόλο του μηχανισμού εκκίνησης του οχήματος (Εκκινητής / μίζα). Η μίζα είναι ένας μικρός ηλεκτροκινητήρας και τροφοδοτείται από το συσσωρευτή (Μπαταρία). Τα τυλίγματα του δρομέα και των ηλεκτρομαγνητών του εκκινητή είναι από χοντρό αγωγό ώστε να δέχονται τη μεγάλη ένταση, που είναι απαραίτητη για να δώσει την ισχύ στην εκκίνηση του κινητήρα. Η ισχύς εξαρτάται από το μέγεθος του κινητήρα, το είδος του, τον αριθμό των κυλίνδρων και τις στροφές

7) LED κατάστασης GPS

Μέσω των LED κατάστασης του GPS, ο χρήστης ενημερώνεται για το αν το 3G/GPS shield βρίσκεται σε οπτική επαφή με τουλάχιστον τρεις δορυφόρους και κατά συνέπεια μπορεί να λάβει και να στείλει στον χρήστη την τοποθεσία του. Τα LED κατάστασης είναι δυο: Ένα κόκκινο, το οποίο ενημερώνει τον χρήστη ότι δεν έχει επιτευχθεί σύνδεση, και ένα πράσινο, το οποίο ενεργοποιείται όταν η σύνδεση είναι επιτυχής.

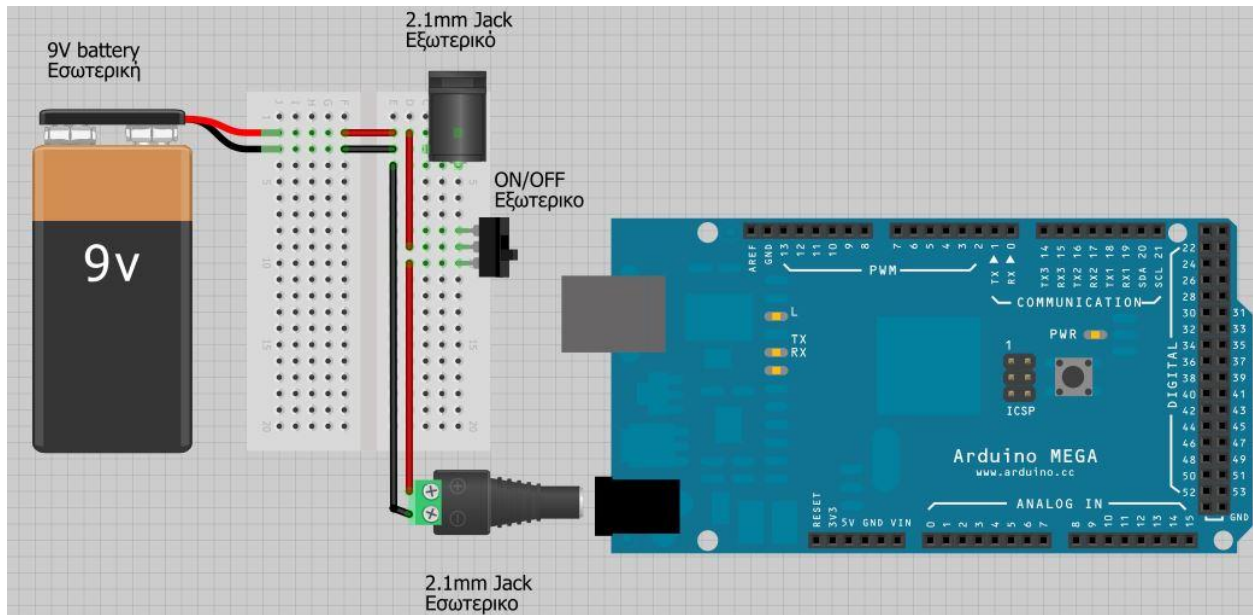
Η σύνδεση των LED κατάστασης του GPS φαίνεται στο παρακάτω διάγραμμα:



8) Τροφοδοσία – Μεταφορά δεδομένων

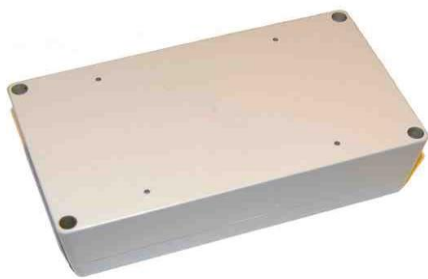
Το Arduino μπορεί να τροφοδοτηθεί με ρεύμα είτε από τον υπολογιστή μέσω της σύνδεσης USB, είτε από εξωτερική τροφοδοσία που παρέχεται μέσω μιας υποδοχής φισ των 2.1mm (θετικός πόλος στο κέντρο) και βρίσκεται στην κάτω-αριστερή γωνία του Arduino. Για να μην υπάρχουν προβλήματα, η εξωτερική τροφοδοσία πρέπει να είναι από 7 ως 12V και μπορεί να προέρχεται από ένα κοινό μετασχηματιστή του εμπορίου, από μπαταρίες ή οποιαδήποτε άλλη πηγή DC.

Λόγω της ανάγκης της κατασκευής για φορητότητα, έπρεπε να εξασφαλιστεί ότι πέρα από τον μετασχηματιστή, η συσκευή θα μπορούσε να λειτουργήσει και με τροφοδοσία από μπαταρία. Επίσης, λόγω της μεγάλης κατανάλωσης σε ενέργεια του 3G/GPS shield, προστέθηκε και ένας γενικός διακόπτης τροφοδοσίας δυο θέσεων (ON/OFF). Ο συνδυασμός αυτός οδήγησε στην παρακάτω διάταξη:



Η 9V μπαταρία τοποθετείται σε σταθερό σημείο στο εσωτερικό της κατασκευής. Επίσης, το εσωτερικό 2.1mm Jack είναι μόνιμα συνδεδεμένο πάνω στο Arduino. Για να τροφοδοτηθεί το Arduino με ρεύμα πρέπει να τεθεί ο Εξωτερικός διακόπτης σε θέση ON. Σε περίπτωση που συνδεθεί εξωτερική τροφοδοσία στο εξωτερικό 2.1mm jack τότε αποσυνδέεται η εσωτερική μπαταρία, με αποτέλεσμα της αύξηση αυτονομίας της κατασκευής.

9) Κουτί κατασκευής



Εικόνα 14. Κουτί κατασκευής

Για την εγκατάσταση της κατασκευής χρησιμοποιήθηκε ένα πλαστικό (PVC) κουτί 250x190x90mm. (Εικόνα 14). Το μέγεθος του κουτιού είναι αρκετά μεγαλύτερο από το ελάχιστο αναγκαίο για να στεγάσει την κατασκευή. Αυτό συνέβει αφ' ενός επειδή στα αρχικά στάδια της κατασκευής δεν ήταν δεδομένες οι απαιτήσεις σε χώρο και αφ' εταίρου επειδή, σαν πρωτότυπη, η κατασκευή δεν θα έχει εμπορική χρήση σε αυτή τη μορφή με αποτέλεσμα να προτιμηθεί η ευκολία στις συνδέσεις από την εξοικονόμηση χώρου. Όσον αφορά το υλικό από το οποίο είναι κατασκευασμένο το κουτί, επελέγη το πλαστικό αντί για μέταλλο λόγω της ευκολίας που παρουσιάζει το συγκεκριμένο υλικό στην παραμετροποίηση.

10) Φωτογραφίες Κατασκευής



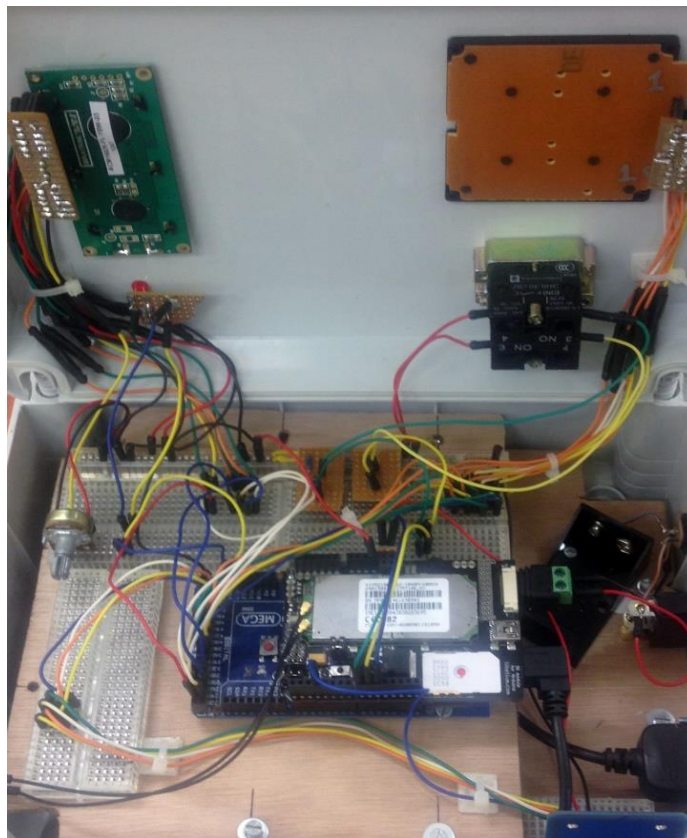
Εικόνα 15. Μπροστινή όψη κατασκευής



Εικόνα 16. Πάνω όψη κατασκευής



Εικόνα 17. Κάτω όψη κατασκευής



Εικόνα 18 Εσωτερικό κατασκευής

2.2 Περιγραφή Σεναρίου

Η παρούσα κατασκευή είναι ένα σύστημα ασφάλειας οχήματος. Το σενάριο λειτουργίας της κατασκευής ξεκινά μόλις ο χρήστης θέσει το κλειδί λειτουργίας του διακόπτη εκκίνησης στη θέση 0, δηλαδή μόλις τερματίσει τη λειτουργία του κινητήρα του οχήματός του. Σε αυτό το χρονικό σημείο ενεργοποιείται το σύστημα ασφάλειας μέσω GPS και η κατασκευή μπαίνει σε κατάσταση «Αναμονής».

Σε κατάσταση «Αναμονής», η κατασκευή ελέγχει κάθε 5 δευτερόλεπτα τα δεδομένα που λαμβάνει από το GPS και αναζητά συγκεκριμένα την ταχύτητα (U_{GPS}). Σε περίπτωση που αντιληφθεί ότι το όχημα κινείται ($U_{GPS} \neq 0$), τότε ενημερώνει τον χρήστη μέσω γραπτού μηνύματος ότι το όχημά του βρίσκεται σε κίνηση και του αποστέλλει τις συντεταγμένες του σημείου στο οποίο βρίσκεται το όχημα εκείνη τη στιγμή. Επίσης, σε κατάσταση «αναμονής», ο χρήστης έχει την δυνατότητα, μέσω γραπτών μηνυμάτων, να ζητήσει και να λάβει τις συντεταγμένες του σημείου στο οποίο βρίσκεται το όχημα.

Για να βγει το όχημα από την κατάσταση «αναμονής» πρέπει ο χρήστης να τοποθετήσει το κλειδί στον διακόπτη εκκίνησης και να τον γυρίσει στη θέση 1. Τότε η κατασκευή μπαίνει σε κατάσταση «Αναγνώρισης». Στην κατάσταση «Αναγνώρισης» η κατασκευή ζητάει από τον χρήστη να τοποθετήσει την κάρτα RFID στον αναγνώστη καρτών για να αναγνωρίσει την ταυτότητά του. Ο χρήστης έχει 5 ευκαιρίες να τοποθετήσει την σωστή κάρτα στον αναγνώστη καρτών. Μόλις εισαχθεί η σωστή κάρτα τότε η κατασκευή περνάει στο δεύτερο επίπεδο αναγνώρισης, την εισαγωγή κωδικού PIN. Και σε αυτή την περίπτωση, ο χρήστης έχει 5 ευκαιρίες να εισάγει τον σωστό κωδικό. Μόλις εισαχθεί ο σωστός κωδικός PIN, το σύστημα ξεκλειδώνει και επιτρέπει στον χρήστη να ξεκινήσει το όχημα, γυρίζοντας το διακόπτη εκκίνησης στη θέση 2 περνώντας σε κατάσταση «Πορείας». Σε διαφορετική περίπτωση, εάν δηλαδή ο χρήστης είτε δεν εισάγει την σωστή κάρτα είτε δεν εισάγει τον σωστό κωδικό, τότε το σύστημα τερματίζεται και μπαίνει σε κατάσταση «Κλειδώματος»

Μόλις μπει σε κατάσταση «Κλειδώματος», η κατασκευή ενημερώνει τον χρήστη μέσω γραπτού μηνύματος. Δεν επιτρέπει την εκκίνηση του κινητήρα μέχρι να ξεκλειδώσει. Για να συμβεί αυτό, ο χρήστης πρέπει να στείλει ένα γραπτό μήνυμα με την αντίστοιχη εντολή. Τότε επιτρέπει στον χρήστη να γυρίσει τον διακόπτη εκκίνησης στη θέση 2 και να περάσει σε κατάσταση «πορείας».

Σε κατάσταση «Πορείας», η κατασκευή σταματάει να ελέγχει τις συντεταγμένες του GPS, αφού η ταυτότητα του χρήστη έχει επιβεβαιωθεί. Για να βγει η κατασκευή από την κατάσταση «πορείας» πρέπει ο χρήστης θέσει το κλειδί λειτουργίας του διακόπτη εκκίνησης στη θέση 0, δηλαδή να τερματίσει τη λειτουργία του κινητήρα του οχήματος. Με αυτή την κίνηση ο κύκλος ενεργειών τελειώνει και το σενάριο επανέρχεται στην εκκίνηση.

Εντολές μηνυμάτων (Αποστολή από χρήστη)

Για να αναγνωρίσει η κατασκευή ένα γραπτό μήνυμα πρέπει το μήνυμα να έχει την ακόλουθη μορφή:

**** "ΕΝΤΟΛΗ"

Το **** είναι ένας μοναδικός κωδικός ο οποίος θα δίνεται στον χρήστη, θα είναι μοναδικός για κάθε κατασκευή και θα εξασφαλίζει ότι κανένας δεν θα μπορεί να απενεργοποιήσει τον έλεγχο ταυτότητας πέρα από τον ιδιοκτήτη του οχήματος. Στην παρούσα κατασκευή έχει επιλεγεί ο κωδικός 5204. Σημειώνεται εδώ ότι ανάμεσα στον κωδικό και την εντολή υπάρχει κενό και ότι οι χαρακτήρες " παραλείπονται.

Οι εντολές που αναγνωρίζει η κατασκευή είναι οι ακόλουθες:

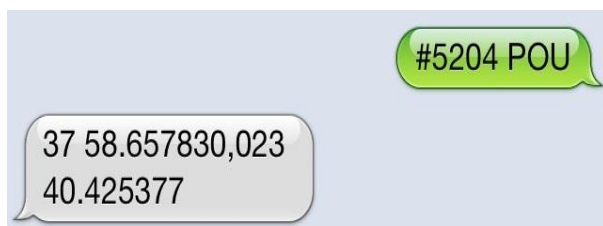
POU: Μπορεί να ενεργοποιηθεί σε όλες τις καταστάσεις λειτουργίας της κατασκευής. Μόλις αναγνωριστεί, το σύστημα στέλνει ένα γραπτό μήνυμα στο χρήστη με τις συντεταγμένες του σημείου στο οποίο βρίσκεται το όχημα. Μπορεί να φανεί χρήσιμη σε περίπτωση που ο ιδιοκτήτης έχει ξεχάσει σε ποιο σημείο έχει παρκάρει το όχημά του καθώς και σε περίπτωση που ο ιδιοκτήτης έχει δανείσει το όχημα σε κάποιο τρίτο πρόσωπο και θέλει να δει που βρίσκεται.

RFID: Ενεργοποιείται στις καταστάσεις «Αναγνώρισης» και «Κλειδώματος» σε περίπτωση που ο ιδιοκτήτης του οχήματος επιθυμεί την απενεργοποίηση του ελέγχου κάρτας RFID. Αυτό μπορεί να χρειαστεί σε περίπτωση που χαθεί η κάρτα RFID ή σε περίπτωση βλάβης του αναγνώστη RFID.

PW: Ενεργοποιείται στις καταστάσεις «Αναγνώρισης» και «Κλειδώματος» σε περίπτωση που ο ιδιοκτήτης του οχήματος επιθυμεί την απενεργοποίηση του ελέγχου κωδικού πρόσβασης. Αυτό μπορεί να χρειαστεί σε περίπτωση που ο ιδιοκτήτης ξεχάσει τον κωδικό, σε περίπτωση βλάβης του πληκτρολογίου ή σε περίπτωση που ο ιδιοκτήτης θέλει να δανείσει το όχημα σε κάποιον τρίτο κα δεν θέλει να του αποκαλύψει τον κωδικό.

Εντολές μηνυμάτων (Λήψη από χρήστη)

Η κατασκευή, ακολουθώντας το σενάριο λειτουργίας, μπορεί να στείλει στον χρήστη τα παρακάτω μηνύματα:

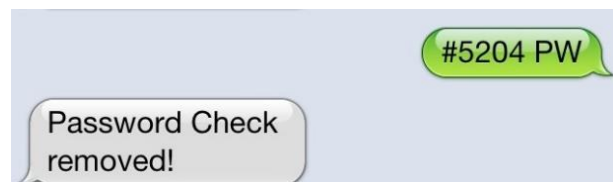


Το μήνυμα έχει τη μορφή **xx xx.xxxxxx,yy yy.yyyyyy** Οι χαρακτήρες x αντιπροσωπεύουν το γεωγραφικό μήκος του σημείου στο οποίο βρίσκεται το όχημα ενώ οι χαρακτήρες y αντιπροσωπεύουν το γεωγραφικό πλάτος του σημείου. Η κατασκευή στέλνει ένα τέτοιο

μήνυμα σαν απάντηση στο χρήστη σε περίπτωση που λάβει την εντολή **POU**. Για να μετατρέψει ο χρήστης τις συντεταγμένες αυτές σε φυσικό σημείο, αρκεί να συνδεθεί στο internet από οποιονδήποτε υπολογιστή ή smartphone, να μεταβεί στην ιστοσελίδα maps.Google.gr και να εισάγει στην μπάρα αναζήτησης τις συντεταγμένες που έλαβε από την κατασκευή. Ο χάρτης της υπηρεσίας θα μεταβεί αυτόματα στο φυσικό σημείο που αντιστοιχεί σε αυτές τις συντεταγμένες (Εικόνα 19).

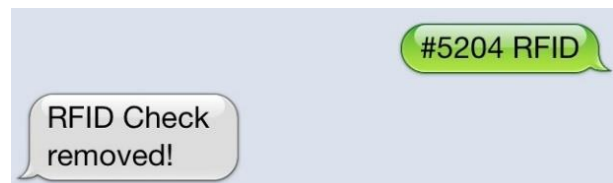


Το μήνυμα αποστέλλεται όταν η κατασκευή βρίσκεται σε κατάσταση «Αναμονής». Ενημερώνει τον χρήστη ότι το όχημα κινείται χωρίς να έχει γίνει επιβεβαίωση ταυτότητας του οδηγού και αποστέλλει, με μορφή συντεταγμένων, την θέση του οχήματος. Οι χαρακτήρες των συντεταγμένων λειτουργούν όπως στο παραπάνω μήνυμα.



Το μήνυμα αυτό αποστέλλεται όταν η κατασκευή βρίσκεται σε κατάσταση «Κλειδώματος» και έχει αποτύχει η αναγνώριση μέσω προσωπικού κωδικού. Η κατασκευή στέλνει ένα τέτοιο μήνυμα σαν απάντηση στο

χρήστη σε περίπτωση που λάβει την εντολή **PW** ώστε να ενημερώσει τον χρήστη ότι έχει αρθεί ο Έλεγχος ταυτότητας μέσω προσωπικού κωδικού και μπορεί να εκκινήσει το όχημα.



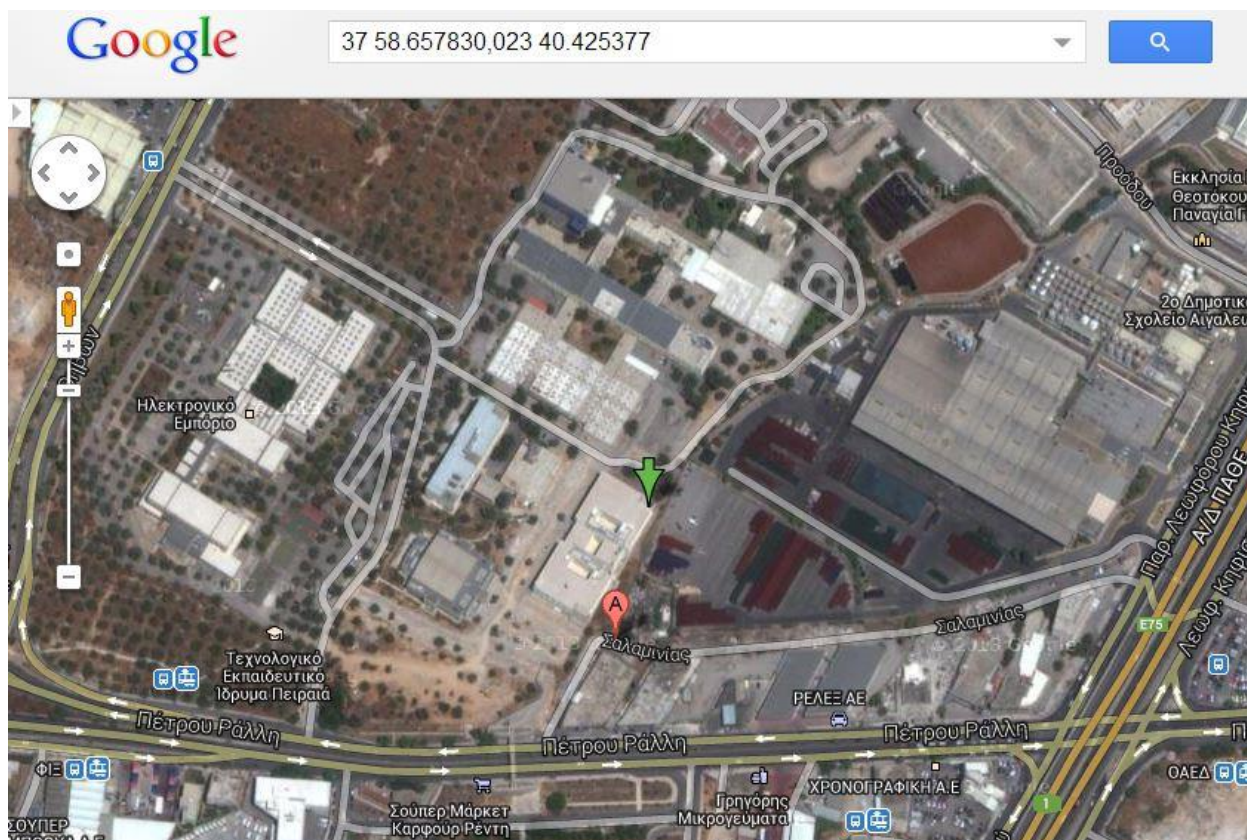
Το μήνυμα αυτό αποστέλλεται όταν η κατασκευή βρίσκεται σε κατάσταση «Κλειδώματος» και έχει αποτύχει η αναγνώριση μέσω RFID. Η κατασκευή στέλνει ένα τέτοιο μήνυμα σαν απάντηση στο χρήστη σε περίπτωση

που λάβει την εντολή **RFID** ώστε να ενημερώσει τον χρήστη ότι έχει αρθεί ο Έλεγχος ταυτότητας μέσω RFID και μπορεί να εκκινήσει το όχημα.



Το μήνυμα αυτό αποστέλλεται τη στιγμή που η κατασκευή μπαίνει από κατάσταση «Αναγνώρισης» σε κατάσταση «Κλειδώματος» λόγω αποτυχίας αναγνώρισης της ταυτότητας

του χρήστη, είτε μέσω προσωπικού κωδικού είτε μέσω RFID.



Εικόνα 19. Λειτουργικό περιβάλλον ιστοσελίδας maps.google.gr

2.3 Επεξήγηση Κώδικα

Για τον προγραμματισμό του Arduino, χρησιμοποιείται η γλώσσα προγραμματισμού processing. Η processing είναι μια ανοιχτού κώδικα γλώσσα προγραμματισμού. Αναπτύχθηκε αρχικά ως ένα σχεδιαστικό πρόγραμμα, αλλά στη συνέχεια εξελίχθηκε σε εργαλείο δημιουργίας ολοκληρωμένων έργων. Η processing βασίζεται στη JAVA και έτσι τρέχει σε όλες τις πλατφόρμες. Ο κώδικας που γράφεται με αυτήν, μπορεί να γίνει export σε αρχείο jar (applet) και να τρέχει μέσα από τον φυλλομετρητή (π.χ. Firefox, Internet Explorer, Chrome κτλ. η processing παράγει και τον κώδικα σε html) ή μπορεί να γίνει εκτελέσιμο το οποίο θα τρέχει χωρίς να είναι απαραίτητη η processing. Το μόνο που απαιτείται για αυτό είναι ο JRE (Java Runtime Environment). (βλέπε [11]) Τα κύρια μέρη ενός προγράμματος γραμμένου σε γλώσσα processing είναι τέσσερα:

- **Δήλωση μεταβλητών:** Είναι το πρώτο που γράφεται σε ένα πρόγραμμα. Σε αυτό το μέρος δηλώνονται οι βιβλιοθήκες και οι μεταβλητές που θα χρησιμοποιηθούν και δεσμεύεται το κομμάτι της μνήμης στο οποίο θα αποθηκευτούν.
- **Ρουτίνα setup:** Η ρουτίνα setup τρέχει μόνο μια φορά στην αρχή του προγράμματος. Σε αυτό το μέρος καθορίζεται ο τύπος των pin που δηλώθηκαν πριν (Input/Output) καθώς επίσης και όλα τα αντικείμενα που θα χρησιμοποιηθούν στο πρόγραμμα.
- **Ρουτίνα loop:** Η ρουτίνα loop τρέχει συνέχεια. Είναι το κύριο μέρος ενός προγράμματος και αποτελείται από εντολές και κλήσεις ρουτίνων του χρήστη.
- **Ρουτίνες χρήστη:** Οι ρουτίνες που έχει δημιουργήσει ο χρήστης.

Η συγγραφή του κώδικα της παρούσας κατασκευής έγινε διατηρώντας την παραπάνω δομή. Στη συνέχεια παρατίθεται ο κώδικας που χρησιμοποιήθηκε, με μια σύντομη επεξήγηση.

Δήλωση μεταβλητών

```
#include <SPI.h> // Φόρτωση της βιβλιοθήκης του RFID
#include <LiquidCrystal.h> // Φόρτωση της βιβλιοθήκης της LCD
#include "Keypad.h" // Φόρτωση της βιβλιοθήκης του NUMPAD
#define MAX_LEN 16 // Μέγιστο μέγεθος του πίνακα δεδομένων του RFID
#define uchar unsigned char
#define uint unsigned int
LiquidCrystal lcd (22, 23, 24, 25, 26, 27); // Δημιουργία του αντικειμένου LCD

// Αρχικοποίηση του RFID
const int chipSelectPin = 53; // Screen Shield (Select Pin)
const int NRSTPD = 3; // Screen Shield (Reset Pin)
uchar serNum[5]; // Δήλωση 5byte κωδικού κάρτας RFID
int CardNo1[5]={26,37,79,33,81}; // Κωδικός κάρτας Admin
int CardNo2[5]={102,105,24,224,247}; // Κωδικός κάρτας User
```

```
boolean unlock1 = false;           // Μεταβλητή ελέγχου RFID
boolean unlock2 = false;           // Μεταβλητή ελέγχου RFID
boolean correctPW=false;           // Μεταβλητή ελέγχου Password
boolean correctRFID=false;         // Μεταβλητή ελέγχου RFID
int unauthPW=0;                    // Μετρητής λανθασμένων Password
int unauthRFID=0;                  // Μετρητής λανθασμένων RFID

// Αρχικοποίηση του NUPAD
const byte ROWS = 4;               // 4 Σειρές
const byte COLS = 4;               // 4 Στήλες
char keys[ROWS][COLS] = {
    {'1','2','3','A'},
    {'4','5','6','B'},
    {'7','8','9','C'},
    {'*','0','#','D'};
byte rowPins[ROWS] = {36, 38, 40, 42} // Row1-->Pin36,Row2-->Pin38,Row3-->Pin40,Row4-->Pin42
byte colPins[COLS] = {37, 39, 41, 43}; // Col1-->Pin37,Col2-->Pin39,Col3-->Pin41,Col4-->Pin43

//Δημιουργία του αντικειμένου KEYPAD
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );

//Δήλωση του PASSWORD-->5204BC
char PIN[6]={'5','2','0','4','B','C'};
char attempt[6]={0,0,0,0,0,0};
int z=0;
boolean passCheck=false;           // Μετρητής στοιχείων του password

// Αρχικοποίηση κλειδιού
int keyLeft=8;
int keyRight=9;
int leftkey=0;
int rightkey=0;

//Flags
int Start;                          // Flag Start
int PasswordsOK=0;                  // Flag Σωστών Κωδικών και RFID
int HaltRFID=0;                     // Flag Λάθος Κωδικού/RFID
int HaltPW=0;                       // Flag Λάθος PW
int EngineOK=0;                     // Flag Μηχανής σε λειτουργία

//Σεταρισμα 3G/GPS shield
int onModulePin = 2;                // Το pin που ανοίγει το shield
int a,b,c,d;                         // Μεταβλητές για τον υπολογισμό των μηνυμάτων
int n_sms;                            // Αριθμός νέων μηνυμάτων
int x;                                 // Μεταβλητή ελέγχου μηνυμάτων
int q,y;                               // Μεταβλητές ελέγχου GPS
boolean GPSready=false;              // Μεταβλητή ελέγχου κατάστασης GPS
char data[256];                       // Πίνακας αποθήκευσης μηνύματος
char data2[256];                     // Πίνακας επεξεργασίας GPS
```

```
char phone_number[]="+306945964069"; // Αριθμός τηλεφώνου επικοινωνίας ιδιοκτήτη

//Πίνακες δεδομένων GPS
char latitude[12]={0,0,0,0,0,0,0,0,0,0,0,0};
char longitude[13]={0,0,0,0,0,0,0,0,0,0,0,0,0};
char date[6]={0,0,0,0,0,0};
char UTC_time[8]={0,0,0,0,0,0,0,0};
char speed_OG[6]={0,0,0,0,0,0};
char altitude[6]={0,0,0,0,0,0};

//Setarisma LED κατάστασης GPS
int GPSON = A13;
int GPSOFF = A14;
```

Το παραπάνω τμήμα του κώδικα, αποτελεί το τμήμα δήλωσης των μεταβλητών. Σε αυτό το κόμματι φορτώνονται οι βιβλιοθήκες και δηλώνονται όλες οι μεταβλητές και τα αντικείμενα που θα χρησιμοποιηθούν. Τέλος, αρχικοποιούνται όλες οι μεταβλητές ελέγχου και οι πίνακες δεδομένων που θα χρησιμοποιηθούν στο πρόγραμμα.

Ρουτίνα Setup

```
void setup()
{
  digitalWrite(10,HIGH); // Ρύθμιση του Software Reset PIN
  delay(200);
  pinMode(10,OUTPUT);
  pinMode(GPSON, OUTPUT); // Setarisma LED κατάστασης GPS
  pinMode(GPSOFF, OUTPUT); // Setarisma LED κατάστασης GPS
  digitalWrite(GPSOFF, HIGH); // LED OFF
  digitalWrite(GPSON, LOW); // LED OFF
  lcd.begin(16, 2); // Δήλωση αριθμού στηλών και σειρών της οθόνης LCD
  lcd.clear(); // Καθαρισμός της οθόνης
  lcd.print("Security System"); // Εκτύπωση κειμένου στην 1η γραμμή
  lcd.setCursor(0, 1);
  lcd.print("initialising..."); // Εκτύπωση κειμένου στη 2η γραμμή
  delay(3000);
  pinMode(keyLeft,INPUT); // Δήλωση του pin 8 ως input για την αριστερή θέση του κλειδιού
  pinMode(keyRight,INPUT); // Δήλωση του pin 9 ως input για τη δεξιά θέση του κλειδιού
  Serial.begin(115200); // Εκκίνηση της σειριακής θύρας
  SPI.begin(); // Εκκίνηση της SPI βιβλιοθήκης
  pinMode(chipSelectPin,OUTPUT);
  digitalWrite(chipSelectPin, LOW); // Ενεργοποίηση του RFID
  pinMode(NRSTPD,OUTPUT);
  digitalWrite(NRSTPD, HIGH);
  MFRC522_Init(); // Εκκίνηση του MFRC522
  delay(2000);
```

```
pinMode(onModulePin, OUTPUT);
switchModule(); // Εκκίνηση του 3G Shield
for (int i=0;i< 5;i++){ // Καθυστέρηση μέχρι να ενεργοποιηθεί το shield
delay(5000);}
Serial.println("AT+CMGF=1"); // SMS mode
delay(100);
Serial.println("AT+CGPS=1,1"); // GPS σε stand-alone mode
delay(2000);
Serial.println("AT+CGPSINFO"); // Ζητάει την τοποθεσία από το GPS
delay(2000);
Serial.flush();}
```

Το παραπάνω τμήμα του κώδικα αποτελεί τη ρουτίνα setup(). Εδώ καθορίζονται οι καταστάσεις των pin που θα χρησιμοποιηθούν (Είσοδοι / Έξοδοι), ενεργοποιείται η LCD οθόνη, η σειριακή θύρα, το RFID και το 3G/GPS shield. Αξίζει να σημειωθεί ότι, όπως και στα κινητά τηλέφωνα, για να πραγματοποιηθεί επιτυχής σύνδεση απαιτείται ελάχιστος χρόνος 25 δευτερολέπτων, κάτι που επιβάλει την ύπαρξη αρκετά μεγάλης καθυστέρησης στον κώδικα.

Ρουτίνα Loop

Η ρουτίνα loop() έχει δομηθεί με βάση την θέση του κλειδιού. Σε κάθε επανάληψή της, ελέγχεται η κατάσταση του κλειδιού και ανάλογα με το αποτέλεσμα και την κατάσταση λειτουργίας της κατασκευής, καλούνται οι ανάλογες ρουτίνες. Για την καλύτερη κατανόηση του κώδικα η ρουτίνα loop() θα παρουσιαστεί τμηματικά.

```
void loop()
{leftkey = digitalRead(keyLeft);
rightkey = digitalRead(keyRight);

if (leftkey == LOW && rightkey == LOW && EngineOK == 0 && PasswordsOK == 0)
{ lcd.clear();
  lcd.print(" Turn Key ");
  lcd.setCursor(0, 1);
  lcd.print(" to begin");
  delay(1000);
  receiveSMS();
  if (GPSready == false)
  {GetGPS();}
  delay (1000);
  SpeedCheck();}
```

Το παραπάνω κομμάτι κώδικα αντιστοιχεί στην κατάσταση «Αναμονής». Η κατασκευή εξετάζει την ύπαρξη μηνύματος από τον χρήστη (ρουτίνα *receiveSMS()*) και έπειτα ελέγχει εάν το όχημα κινείται χωρίς προηγούμενη επιβεβαίωση ταυτότητας (ρουτίνα *SpeedCheck()*).

```
if (leftkey == HIGH && PasswordsOK == 0 && EngineOK == 0 && HaltRFID == 1 && HaltPW == 0)
{receiveSMS(); }
```

```
if (leftkey == HIGH && PasswordsOK == 0 && EngineOK == 0 && HaltRFID == 0 && HaltPW == 1)
{receiveSMS(); }
```

Το παραπάνω κομμάτι κώδικα αντιστοιχεί στις καταστάσεις «Κλειδώματος», στην πρώτη περίπτωση λόγω αποτυχημένης επιβεβαίωσης ταυτότητας μέσω RFID ενώ στη δεύτερη μέσω προσωπικού κωδικού. Η κατασκευή εξετάζει την ύπαρξη μηνύματος από τον χρήστη (ρουτίνα *receiveSMS()*) για να του επιτραπεί η παράκαμψη του κλειδώματος μέσω γραπτού μηνύματος.

```
if (leftkey == HIGH && PasswordsOK == 0 && EngineOK == 0 && HaltRFID == 0 && HaltPW == 0)
{ delay(200);
  lcd.clear();
  lcd.print(" Waiting for");
  lcd.setCursor(0, 1);
  lcd.print(" Keycard");
  delay(200);
  Start=1
  receiveSMS();
  if (GPSready == false)
  { GetGPS();}
if (Start == 1)
{while(correctRFID==false && unauthRFID<5)
{if (GPSready == false)
  { GetGPS();}
  uchar i,tmp;
  uchar status;
  uchar str[MAX_LEN];
  status = MFRC522_Request(PICC_REQIDL, str); // Επιστρέφει τον τύπο της κάρτας
  status = MFRC522_Anticoll(str); // Επιστρέφει τον αριθμό της κάρτας
  memcpy(serNum, str, 5);
  if (status == MI_OK)
  { //Ταυτοποίηση κάρτας RFID
    for (int q=0; q<5; q++)
    { if (serNum[q] == CardNo1[q])
      { unlock1 = true } // Αναγνώριση Admin
      if (serNum[q] == CardNo2[q])
      { unlock2 = true; } // Αναγνώριση User
    }
    if(unlock1)
    {correctRFID=true;
      lcd.clear();
      lcd.print("Find out a card");
      delay(500);
      lcd.setCursor(0, 1);
      lcd.print("Card Identified!");
      delay(2000);
      lcd.clear();
```

```
lcd.print(" Hello Admin");
delay(3000);
lcd.clear();
Start=0;}
else if(unlock2)
{correctRFID=true;
lcd.clear();
lcd.print("Find out a card");
delay(500);
lcd.setCursor(0, 1);
lcd.print("Card Identified!");
delay(2000);
lcd.clear();
lcd.print(" Hello User");
delay(3000);
lcd.clear();
Start=0;}
else
{correctRFID=false;
lcd.clear();
lcd.print("Find out a card");
delay(500);
lcd.setCursor(0, 1);
lcd.print(" Unknown Card");
delay(1500);
lcd.clear();
lcd.print(" Please Retry");
lcd.setCursor(0, 1);
lcd.print("Attempts left ");
lcd.print((4-unauthRFID));
unauthRFID++;
receiveSMS();}
if (unauthRFID >= 5)
{ correctRFID=false;
lcd.clear();
lcd.print(" Access Denied!");
SendSMS5();
delay(5000);
Serial.println("AT+CMGD=1,4");
Serial.flush();
delay(100);
Start=0;
HaltRFID=1;}}

//Ταυτοποίηση μέσω Password
if (unlock1||unlock2)
{lcd.clear();
if (GPSready == false) //Έλεγχος κατάστασης GPS
{GetGPS();}
```

```
lcd.print(" Enter Password");
lcd.setCursor(0, 1);
lcd.print("  and press #");
readKeypad();
while (correctPW==false && unauthPW<5)
{receiveSMS();
 lcd.clear();
 lcd.print(" P/W incorrect");
 delay(1500);
 lcd.clear();
 lcd.print(" Please Retry");
 lcd.setCursor(0, 1);
 lcd.print("Attempts left ");
 lcd.print((5-unauthPW));
 readKeypad(); }
if (unauthPW >= 5)
{ lcd.clear();
 lcd.print(" Access Denied!");
 SendSMS5();
 delay(5000);
 Serial.println("AT+CMGD=1,4");
 Serial.flush();
 delay(100);
 Start=0;
 HaltPW=1;}
else
{if (GPSready == false)
 { GetGPS();}
 lcd.clear();
 lcd.print(" Start your ");
 lcd.setCursor(0, 1);
 lcd.print(" Engine!");
 PasswordsOK=1;
 MFRC522_Halt();}}
```

Το παραπάνω κομμάτι κώδικα, αντιστοιχεί στην κατάσταση «Αναγνώρισης». Πρώτα γίνεται η επιβεβαίωση ταυτότητας μέσω κάρτας RFID (ρουτίνες *MFRC522_Request()* και *MFRC522_Anticoll()*) και έπειτα μέσω προσωπικού κωδικού (ρουτίνα *readKeypad()*). Σε περίπτωση επιτυχούς εισαγωγής στοιχείων απενεργοποιείται το RFID (ρουτίνα *MFRC522_Halt()*) και η κατασκευή περνάει σε κατάσταση «Πορείας» ενώ σε διαφορετική περίπτωση περνάει σε κατάσταση «Κλειδώματος».

```
if (leftkey == HIGH && PasswordsOK ==1 && EngineOK == 0 && HaltPW == 1)
{ lcd.clear();
 lcd.print(" Start your ");
 lcd.setCursor(0, 1);
 lcd.print(" Engine!");}
```

```
if (leftkey == HIGH && PasswordsOK == 1 && EngineOK == 0 && HaltRFID == 1)
{ lcd.clear();
  lcd.print(" Start your ");
  lcd.setCursor(0, 1);
  lcd.print(" Engine!");
  delay(1000);}
```

Το παραπάνω κομμάτι κώδικα χρησιμοποιείται για την μετάβαση της κατασκευής από κατάσταση «Κλειδώματος» σε κατάσταση «Πορείας» μετά από αποστολή μηνύματος του χρήστη.

```
if (rightkey == HIGH && PasswordsOK == 1 && EngineOK == 1)
{ lcd.clear();
  lcd.print(" Engine is ");
  lcd.setCursor(0, 1);
  lcd.print(" running!");
  delay(1000);
  receiveSMS();}
```

Σε αυτό το κομμάτι κώδικα παρουσιάζεται η κατάσταση «Πορείας» της κατασκευής. Η κατασκευή μπορεί να δεχτεί μήνυμα από τον χρήστη (ρουτίνα *receiveSMS()*) ώστε να του στείλει τις συντεταγμένες του οχήματος.

```
if (rightkey == LOW && leftkey == LOW && PasswordsOK == 1 && EngineOK == 1)
{ lcd.clear();
  lcd.print(" Reseting... ");
  delay(1000);
  unlock1=false;
  unlock2=false;
  correctPW=false;
  correctRFID=false;
  unauthPW=0;
  unauthRFID=0;
  Start=0;
  PasswordsOK=0;
  HaltRFID=0;
  HaltPW=0;
  EngineOK=0;}}
```

Στο τελευταίο κομμάτι της ρουτίνας *loop()* παρουσιάζεται η μετάβαση της κατασκευής από την κατάσταση «Πορείας» σε κατάσταση «Αναμονής». Αυτό πραγματοποιείται με την επαναφορά των μεταβλητών ελέγχου στις αρχικές τους τιμές.

Ρουτίνες χρήστη

Οι ρουτίνες που χρησιμοποιήθηκαν στην κατασκευή χωρίζονται σε δυο κατηγορίες: Σε αυτές που συνόδευαν το 3G/GPS shield και τα εξαρτήματα (Keypad και RFID) και σε αυτές που

δημιουργήθηκαν για την ανάγκη της κατασκευής. Οι ρουτίνες τα πρώτης κατηγορίας χρησιμοποιήθηκαν όπως τις έδωσε ο κατασκευαστής και εντάχθηκαν στον κώδικα σύμφωνα με τις οδηγίες που συνόδευαν το κάθε εξάρτημα αλλά πάντα σύμφωνα με τις ανάγκες της κατασκευής. Όσες συνόδευαν το RFID παρατίθενται παρακάτω:

```
/*
 * Function: MFRC522_Init
 * Description: Εκκίνηση RC522
 * Input parameters: None
 * Return value: None
 */
void MFRC522_Init(void)
{digitalWrite(NRSTPD,HIGH);
 MFRC522_Reset();
 Write_MFRC522(TModeReg, 0x8D);
 Write_MFRC522(TPrescalerReg, 0x3E);
 Write_MFRC522(TReloadRegL, 30);
 Write_MFRC522(TReloadRegH, 0);
 Write_MFRC522(TxAutoReg, 0x40);
 Write_MFRC522(ModeReg, 0x3D);
 AntennaOn();}

/*
 * Function: MFRC522_Reset
 * Description: Reset RC522
 * Input parameters: None
 * Return value:None
 */
void MFRC522_Reset(void)
{ Write_MFRC522(CommandReg, PCD_RESETPHASE);}

/*
 * Function: Write_MFRC522
 * Description: Γράφει στο MFRC522
 * Input parameters: addr - register address; val - value to be written
 * Return value:
 */
void Write_MFRC522(uchar addr, uchar val)
{digitalWrite(chipSelectPin, LOW);
 SPI.transfer((addr<<1)&0x7E);
 SPI.transfer(val);
 digitalWrite(chipSelectPin, HIGH);}
```

```
/*
* Function: AntennaOn
* Description: Ανοίγει την κεραία λήψης του RFID
* Input parameters: None
* Return value:
*/
void AntennaOn(void)
{uchar temp;
  temp = Read_MFRC522(TxControlReg);
  if (!(temp & 0x03))
  {SetBitMask(TxControlReg, 0x03);}}
```

```
/*
* Function: Read_MFRC522
* Description: Διαβάζει από το MFRC522
* Input Parameters: addr - register address
* Return value: Returns a byte of data read
*/
uchar Read_MFRC522(uchar addr)
{uchar val;
  digitalWrite(chipSelectPin, LOW);
  SPI.transfer(((addr<<1)&0x7E) | 0x80);
  val =SPI.transfer(0x00);
  digitalWrite(chipSelectPin, HIGH);
  return val; }
```

```
/*
* Letter of several: SetBitMask
* Description: set RC522 register bit
* Input parameters: reg - register address; mask - set value
* Return value:
*/
void SetBitMask(uchar reg, uchar mask)
{uchar tmp;
  tmp = Read_MFRC522(reg);
  Write_MFRC522(reg, tmp | mask);}
```

```
/*
* Function: MFRC522_Request
* Description: look for the card, read the card type
* Enter parameters: reqMode, - to find the card the way,
* -----TagType - return type of card
* Return values: successful return MI_OK
*/
uchar MFRC522_Request(uchar reqMode, uchar *TagType)
{uchar status;
  uint backBits;
  Write_MFRC522(BitFramingReg, 0x07);}
```

```
TagType[0] = reqMode;
status = MFRC522_ToCard(PCD_TRANSCEIVE, TagType, 1, TagType, &backBits);
if ((status != MI_OK) || (backBits != 0x10))
{ status = MI_ERR; }
return status; }

/*
* Function: MFRC522_ToCard
* Description: RC522 and ISO14443 card communication
* -----Input Parameters: command - MF522 command word,
* SendData - RC522 sent to the card data
* sendLen - Send data length
* BackData card - received data is returned,
* BackLen - Returns the length of the data bits
* Return values: successful return MI_OK
*/
uchar MFRC522_ToCard(uchar command, uchar *sendData, uchar sendLen, uchar *backData, uint *backLen)
{ uchar status = MI_ERR;
  uchar irqEn = 0x00;
  uchar waitIRq = 0x00;
  uchar lastBits;
  uchar n;
  uint i;
  switch (command)
  { case PCD_AUTHENT:
    { irqEn = 0x12;
      waitIRq = 0x10;
      break; }
    case PCD_TRANSCEIVE:
    { irqEn = 0x77;
      waitIRq = 0x30;
      break; }
    default:
    { break; }
  }
  Write_MFRC522(CommIEnReg, irqEn|0x80);
  ClearBitMask(CommIrqReg, 0x80);
  SetBitMask(FIFOLevelReg, 0x80);
  Write_MFRC522(CommandReg, PCD_IDLE);
  //Write data to the FIFO
  for (i=0; i<sendLen; i++)
  { Write_MFRC522(FIFODataReg, sendData[i]); }
  //Execute the command
  Write_MFRC522(CommandReg, command);
  if (command == PCD_TRANSCEIVE)
  { SetBitMask(BitFramingReg, 0x80); }
  //Wait for data is received
  i = 2000;
```

```
do
{n = Read_MFRC522(CommIrqReg);
 i--;}
while ((i!=0) && !(n&0x01) && !(n&waitIRq));
ClearBitMask(BitFramingReg, 0x80);
if (i != 0)
{if(!(Read_MFRC522(ErrorReg) & 0x1B))
 {status = MI_OK;
  if (n & irqEn & 0x01)
  {status = MI_NOTAGERR;}
  if (command == PCD_TRANSCEIVE)
  {n = Read_MFRC522(FIFOLevelReg);
   lastBits = Read_MFRC522(ControlReg) & 0x07;
   if (lastBits)
   {*backLen = (n-1)*8 + lastBits;}
   else
   {*backLen = n*8;}
   if (n == 0)
   {n = 1;}
   if (n > MAX_LEN)
   {n = MAX_LEN;}
   //Read the received data in the FIFO
   for (i=0; i<n; i++)
   {backData[i] = Read_MFRC522(FIFODataReg); }}
  else
  {status = MI_ERR;}}
return status;}

/*
* Letter of several: ClearBitMask
* Description: clear RC522 register bit
* Input parameters: the address of the reg - register; mask - clear bit value
* Return value:
*/
void ClearBitMask(uchar reg, uchar mask)
{uchar tmp;
 tmp = Read_MFRC522(reg);
 Write_MFRC522(reg, tmp & (~mask)); }

/*
* Function: MFRC522_Anticoll
* Description: anti-collision detection, and read the card serial number of the selected card
* Input parameters: serNum - return to the 4-byte card serial number, the first 5 bytes of the checksum byte
* Return values: successful return MI_OK
*/
uchar MFRC522_Anticoll(uchar *serNum)
{uchar status;
 uchar i;
 uchar serNumCheck=0;
```

```
uint unLen;
Write_MFRC522(BitFramingReg, 0x00);
serNum[0] = PICC_ANTICOLL;
serNum[1] = 0x20;
status = MFRC522_ToCard(PCD_TRANSCEIVE, serNum, 2, serNum, &unLen);
if (status == MI_OK)
{ //Check card serial number
  for (i=0; i<4; i++)
  { serNumCheck ^= serNum[i]; }
  if (serNumCheck != serNum[i])
  { status = MI_ERR; } }
return status; }

/*
* Function: MFRC522_Halt
* Description: Αδρανοποιεί το MFRC522
* Input parameters: None
* Return value: None
*/
void MFRC522_Halt(void)
{ uchar status;
  uint unLen;
  uchar buff[4];
  buff[0] = PICC_HALT;
  buff[1] = 0;
  CalculateCRC(buff, 2, &buff[2]);
  status = MFRC522_ToCard(PCD_TRANSCEIVE, buff, 4, buff,&unLen); }

/*
* Function: CalculateCRC
* Function Description: MF522 calculate the CRC
* Input parameters: pIndata - to be reading a CRC data, len - the length of the data, pOutData - calculated
* CRC results
* Return value:
*/
void CalculateCRC(uchar *pIndata, uchar len, uchar *pOutData)
{ uchar i, n;
  ClearBitMask(DivIrqReg, 0x04);
  SetBitMask(FIFOLevelReg, 0x80);
  //Write data to the FIFO
  for (i=0; i<len; i++)
  { Write_MFRC522(FIFODataReg, *(pIndata+i)); }
  Write_MFRC522(CommandReg, PCD_CALCCRC);
  //Wait for the CRC calculation is done
  i = 0xFF;
  do
  { n = Read_MFRC522(DivIrqReg);
    i--;}
  ;}
```

```
while ((i!=0) && !(n&0x04));  
pOutData[0] = Read_MFRC522(CRCResultRegL);  
pOutData[1] = Read_MFRC522(CRCResultRegM);}
```

Επίσης χρησιμοποιήθηκε μία ρουτίνα που παρείχε ο κατασκευαστής του Numpad, η οποία προσαρμόστηκε στις ανάγκες της κατασκευής και η τελική της μορφή παρατίθεται παρακάτω:

```
/*  
* Function:readKeypad()  
* Description: Διαβάζει τα δεδομένα απο το NUMPAD  
* Input parameters:  
* Return value:  
*/  
void readKeypad()  
{passCheck=true;  
while(passCheck){  
char key = keypad.getKey();  
if (key != NO_KEY)  
{switch(key)  
{case '#':  
delay(100);  
passCheck=false;  
z=0;  
lcd.clear();  
checkPIN();  
break;  
default:  
lcd.print("*");  
attempt[z]=key;  
z++;  
if (z==1){  
lcd.clear();  
lcd.print("P/W:*"); } } } } }
```

Τέλος, παρατίθεται η ρουτίνα που παρείχε ο κατασκευαστής του 3G/GPS shield, με την οποία γίνεται η εκκίνησή του.

```
/*  
* Function: switchModule  
* Description: Ανοίγει το Shield  
* Input parameters:-  
* Return value:-  
*/  
void switchModule()  
{ digitalWrite(onModulePin,HIGH);  
delay(2000);  
digitalWrite(onModulePin,LOW);}
```

Οι παρακάτω ρουτίνες δημιουργήθηκαν για τις ανάγκες της κατασκευής. Παρατίθενται μαζί με μια σύντομη περιγραφή της λειτουργίας της κάθε μιας:

```
/*  
* Function: correctPIN()  
* Description: Διαδικασία για σωστο Pin  
* Input parameters:  
* Return value:  
*/  
void correctPIN()  
{correctPW=true;  
  lcd.clear();  
  lcd.print(" Password");  
  lcd.setCursor(0, 1);  
  lcd.print(" Verified");  
  delay(1000);  
  lcd.clear();}
```

Η ρουτίνα *correctPIN()* καλείται από την ρουτίνα *checkPIN()* σε περίπτωση εισαγωγής από τον χρήστη σωστού προσωπικού κωδικού στην κατάσταση «Αναγνώρισης». Αφού τρέξει, επιτρέπει στον χρήστη να εκκινήσει τον κινητήρα.

```
/*  
* Function:incorrectPIN()  
* Description: Διαδικασία για Λαθος Pin  
* Input parameters:  
* Return value:  
*/  
void incorrectPIN()  
{unauthPW++;}
```

Η ρουτίνα *incorrectPIN()* καλείται από την ρουτίνα *checkPIN()* σε περίπτωση εισαγωγής από τον χρήστη λανθασμένου προσωπικού κωδικού στην κατάσταση «Αναγνώρισης». Αφού τρέξει, ανεβάζει τον μετρητή λανθασμένων κωδικών κατά 1.

```
/*  
* Function:checkPIN()  
* Description: Checkαρει τον PIN  
* Input parameters:  
* Return value:  
*/  
void checkPIN()  
{int correct=0;  
  for (int q=0; q<6; q++)  
  { if (attempt[q]==PIN[q])
```

```
{correct++;}  
if (correct==6)  
{correctPIN();}  
else  
{incorrectPIN();}
```

Η ρουτίνα *check PIN()* καλείται από τη ρουτίνα *readKeypad()* και ελέγχει την ορθότητα του εισαγόμενου από τον χρήστη κωδικού. Σε περίπτωση που διαπιστώσει ορθό κωδικό, καλεί την ρουτίνα *correctPIN()*, ενώ σε διαφορετική περίπτωση καλεί την ρουτίνα *incorrectPIN()*.

```
/*  
* Function: GetGPS  
* Description: Ζητάει τις συντεταγμένες από το GPS  
* Input parameters:  
* Return value:  
*/  
  
void GetGPS()  
{ while (Serial.available() !=0)  
  {Serial.read();}  
  Serial.println("AT+CGPSINFO");  
  Serial.flush();  
  for (q=0;q<255;q++)  
  {data2[q]='\0';}  
  q=0;  
  do{ while(Serial.available()==0);  
    data2[q]=Serial.read();  
    q++;}  
  while(!((data2[q-1] == 'K') && (data2[q-2] == 'O')));  
  q=24;  
  y=0;  
  if(data2[q]!=';')  
  {GPSready = true;  
   digitalWrite(GPSoff, LOW);  
   digitalWrite(GPSon, HIGH);  
   //Latitude  
   do{ latitude[y]=data2[q];  
     y++;  
     q++;}  
   while(data2[q]!=';');  
   q++;  
   latitude[11]=data2[q];           //north or south  
   q+=2;  
   y=0;  
   //Longitude  
   do{ longitude[y]=data2[q];  
     y++;  
     q++;}
```



```
while(data2[q]!=',');
q++;
longitude[12]=data2[q];           //west or east
q+=2;
y=0;
//Date
do{ date[y]=data2[q];
  y++;
  q++;}
while(data2[q]!=',');
q++;
y=0;
//Time
do{UTC_time[y]=data2[q];
  y++;
  q++;}
while(data2[q]!=',');
q++;
y=0;
//Altitude σε μετρα
do{UTC_time[y]=data2[q];
  y++;
  q++;}
while(data2[q]!=',');
q++;
y=0;
//Speed σε κομβους
do{speed_OG[y]=data2[q];
  y++;
  q++; }
while(data2[q]!=','); }
else
{Serial.println("GPS information not available, please wait...");
GPSready = false;
digitalWrite(GPSoff, HIGH);
digitalWrite(GPSon, LOW);}
```

Η ρουτίνα *GetGPS()* καλείται σε όλες τις καταστάσεις λειτουργίας της κατασκευής. Με την κλήση της ενεργοποιείται η σύνδεση της κατασκευής με τους δορυφόρους GPS, ανακτώνται οι πληροφορίες τοποθεσίας και αποθηκεύονται σε πίνακες για χρήση από άλλες ρουτίνες της κατασκευής. Επίσης, ανάλογα με το αν έχει επιτευχθεί σύνδεση με τους δορυφόρους ή όχι, ενεργοποιεί το αντίστοιχο LED κατάστασης GPS.

```
/*
* Function: receiveSMS
* Description: Διαβάζει το τελευταίο μήνυμα και μετά το διαγράφει
* Input parameters:
* Return value:
*/
void receiveSMS()
{delay(1500);
  Serial.println("AT+CPMS=\"SM\", \"SM\", \"SM\"); //selects SIM memory
  Serial.flush();
  for (x=0;x< 255;x++)
  {data[x]='\0';}
  x=0;
  do
  { while(Serial.available()==0);
    data[x]=Serial.read();
    x++;}
  while(!(data[x-1]=='K'&&data[x-2]=='O'));
  delay(1000);
  d = data[x-11];
  if(d == 44) // we have less than 10 SMS
  {b = 0;
   c = (data[x-10]) - 48;}
  else // we have 10 or more SMS
  { a = (data[x-11])-48;
   b = a * 10;
   c = (data[x-10]) - 48;}
  delay(1000);
  n_sms = (b + c);
  Serial.print(n_sms,DEC);
  Serial.println(" SMS stored in SIM memory");
  if (n_sms != 0 && n_sms <10)
  { Serial.println("Showing last SMS:");
    Serial.print("AT+CMGR=");
    Serial.println(n_sms-1,DEC);
    Serial.flush();
    for (x=0;x< 255;x++)
    {data[x]='\0';}
    x=0;
    do
    { while(Serial.available()==0);
      data[x]=Serial.read();
      x++;
      if(data[x-1]==0x0D&&data[x-2]==""){
        x=0;}}
    while(!(data[x-1]=='K'&&data[x-2]=='O'));
    data[x-3]='\0'; //finish the string before the OK
    Serial.print("Message:");
    Serial.println(data);
```

```
Serial.print(" Arithmos stoixeiwn: ");
Serial.println(x,DEC);
delay(5000);
if(data[1]!='#' &&
  data[2]=='5' &&
  data[3]=='2' &&
  data[4]=='0' &&
  data[5]=='4' &&
  data[7]=='P' &&
  data[8]=='O' &&
  data[9]=='U')
{ Serial.println("Password OK");
  SendSMS();
  delay(5000);
  Serial.println("AT+CMGD=1,4");}
if(data[1]!='#' &&
  data[2]=='5' &&
  data[3]=='2' &&
  data[4]=='0' &&
  data[5]=='4' &&
  data[7]=='P' &&
  data[8]=='W')
{ Serial.println("Password OK");
  unauthPW=6;
  PasswordsOK=1;
  lcd.clear();
  SendSMS3();
  delay(5000);
  Serial.println("AT+CMGD=1,4");}
if(data[1] == '#' &&
  data[2] == '5' &&
  data[3] == '2' &&
  data[4] == '0' &&
  data[5] == '4' &&
  data[7] == 'R' &&
  data[8] == 'F' &&
  data[9] == 'T' &&
  data[10] == 'D')
{ Serial.println("Password OK");
  PasswordsOK=1;
  unauthRFID=6;
  lcd.clear();
  SendSMS4();
  delay(5000);
  Serial.println("AT+CMGD=1,4");}
  Serial.println("-----");}
Serial.flush();
delay(100);}
```

Η ρουτίνα *receiveSMS()* καλείται και αυτή σε όλες τις καταστάσεις λειτουργίας της κατασκευής. Με την κλήση της, ανατρέχει στην μνήμη μηνυμάτων του 3G/GPS shield και ελέγχει αρχικά εάν υπάρχει νέο μήνυμα. Σε τέτοια περίπτωση, ελέγχει το περιεχόμενό του και εάν συμφωνεί με τους κανόνες που έχουν τεθεί, καλεί τις ανάλογες ρουτίνες, διαφορετικά το αγνοεί. Σε κάθε περίπτωση πάντως, διαγράφει το μήνυμα.

```
/*
 * Function: SpeedCheck
 * Description: Έλεγχος κίνησης
 * Input parameters:
 * Return value:
 */
void SpeedCheck()
{if (GPSready == true)
 {GetGPS();
  delay(1500);
  if(speed_OG[0] != '0')
  {SendSMS2();
   delay(5000);
   Serial.println("AT+CMGD=1,4");
   Serial.println("-----");
   Serial.flush();
   delay(100);}}}
```

Η ρουτίνα *SpeedCheck()* καλείται σε κατάσταση «Αναμονής». Καλεί με τη σειρά της τη ρουτίνα *GetGPS()*, ελέγχει εάν το όχημα κινείται χωρίς την προηγούμενη επιβεβαίωση ταυτότητας χρήστη και, εάν αυτό συμβαίνει, στέλνει στον χρήστη μέσω της ρουτίνας *SendSMS2()* γραπτό μήνυμα και τις συντεταγμένες του οχήματος.

```
/*
 * Function: SendSMS
 * Description: Στέλνει SMS
 * Input parameters:
 * Return value:
 */
void SendSMS()
{GetGPS();
 delay(1500);
 Serial.print("AT+CMGS=\"");
 Serial.print(phone_number);
 Serial.println("\"");
 delay(1500);
 //Δημιουργία του μηνυματος
 for(int l=0; l<2; l++)
 {Serial.print(latitude[l]);}
 Serial.print(" ");}
```

```
for(int l=2; l<11; l++)
{Serial.print(latitude[l]);}
Serial.print(",");
for(int l=0; l<3; l++)
{Serial.print(longitude[l]);}
Serial.print(" ");
for(int l=3; l<12; l++)
{Serial.print(longitude[l]);}
delay(500);
Serial.write(0x1A); //sends ++
Serial.write(0x0D);
Serial.write(0x0A);
delay(5000);
Serial.flush();
delay(100);
a,b,c,d=0; }
```

/*

* **Function:** SendSMS2

* **Description:** Στελνει SMS σε περιπτωση κλοπής

* **Input parameters:**

* **Return value:**

*/

```
void SendSMS2()
{delay(1500);
Serial.print("AT+CMGS=\"");
Serial.print(phone_number);
Serial.println("\");
delay(1500);
Serial.println("UNAUTHORISED VEHICLE MOVEMENT!");
//Δημιουργια του μηνυματος
for(int l=0; l<2; l++)
{Serial.print(latitude[l]);}
Serial.print(" ");
for(int l=2; l<11; l++)
{Serial.print(latitude[l]);}
Serial.print(",");
for(int l=0; l<3; l++)
{Serial.print(longitude[l]);}
Serial.print(" ");
for(int l=3; l<12; l++)
{Serial.print(longitude[l]);}
delay(500);
Serial.write(0x1A); //sends ++
Serial.write(0x0D);
Serial.write(0x0A);
delay(5000);
Serial.flush();
```

```
delay(100);
a,b,c,d=0;
}

/*
* Function: SendSMS3
* Description: Στέλνει SMS σε περίπτωση αφαίρεσης PW Check
* Input parameters:
* Return value:
*/
void SendSMS3()
{delay(1500);
Serial.print("AT+CMGS=\"");
Serial.print(phone_number);
Serial.println("\");
delay(1500);
Serial.print("Password Check removed!");
//Δημιουργια του μηνυματος
delay(500);
Serial.write(0x1A);
Serial.write(0x0D);
Serial.write(0x0A);
delay(5000);
Serial.flush();
delay(100);
a,b,c,d=0;}

/*
* Function: SendSMS4
* Description: Στέλνει SMS σε περίπτωση αφαίρεσης RFID Check
* Input parameters:
* Return value:
*/
void SendSMS4()
{ delay(1500);
Serial.print("AT+CMGS=\""); // send the SMS number
Serial.print(phone_number);
Serial.println("\");
delay(1500);
Serial.print("RFID Check removed!");
//Δημιουργια του μηνυματος
delay(500);
Serial.write(0x1A); //sends ++
Serial.write(0x0D);
Serial.write(0x0A);
delay(5000);
Serial.flush();
delay(100);
a,b,c,d=0;}
```

```
/*  
* Function: SendSMS5  
* Description: Στέλνει SMS σε περίπτωση κλειδώματος  
* Input parameters:  
* Return value:  
*/  
void SendSMS5()  
{delay(1500);  
  Serial.print("AT+CMGS=\""); // send the SMS number  
  Serial.print(phone_number);  
  Serial.println("\"");  
  delay(1500);  
  Serial.print("SYSTEM LOCKED");  
  //Δημιουργία του μηνυματος  
  delay(500);  
  Serial.write(0x1A); //sends ++  
  Serial.write(0x0D);  
  Serial.write(0x0A);  
  delay(5000);  
  Serial.flush();  
  delay(100);  
  a,b,c,d=0;}
```

Οι ρουτίνες *SendSMS()*-*SendSMS5* καλούνται σε όλες τις καταστάσεις της κατασκευής. Με την κλήση της, κάθε μια στέλνει στον κινητό τηλέφωνο του χρήστη ένα διαφορετικό γραπτό μήνυμα. Η *SendSMS()* αποστέλλει τις συντεταγμένες του οχήματος μετά από αίτηση του χρήστη. Η *SendSMS2()* αποστέλλει τις συντεταγμένες του οχήματος μετά από κίνηση του οχήματος χωρίς προηγούμενη επιβεβαίωση ταυτότητας χρήστη. Η *SendSMS3()* ενημερώνει τον χρήστη ότι η αίτησή του για αφαίρεση του ελέγχου μέσω κωδικού πρόσβασης έγινε δεκτή από την κατασκευή. Η *SendSMS4()* ενημερώνει τον χρήστη ότι η αίτησή του για αφαίρεση του ελέγχου μέσω RFID έγινε δεκτή από την κατασκευή. Τέλος, η *SendSMS5()* ενημερώνει τον χρήστη ότι η κατασκευή πέρασε σε κατάσταση κλειδώματος μετά από αποτυχημένο έλεγχο ταυτότητας, είτε προσωπικού κωδικού είτε RFID.

Κεφάλαιο 3

3.1 Μελλοντική Εξέλιξη

Για την πραγματοποίηση της παρούσας πτυχιακής εργασίας προηγήθηκε έρευνα σε τομείς οι οποίοι παρόλο που εντάσσονται στον Αυτοματισμό ή σε συγγενείς κλάδους δεν έχουν διδαχθεί διεξοδικά στο πρόγραμμα σπουδών του Τμήματος. Η περεταίρω εμβάθυνση σε αυτούς τους τομείς καθώς εξελισσόταν η εργασία, οδήγησε αναπόφευκτα σε επαναξιολόγηση κάποιων αποφάσεων σχετικά με την εργασία οι οποίες δεν μπορούσαν να αλλάξουν και γέννησε νέες ιδέες οι οποίες δεν μπορούσαν, λόγω κόστους σε χρόνο, χρήμα ή πολυπλοκότητα, να υλοποιηθούν. Αυτοί οι προβληματισμοί και οι προτάσεις θα συζητηθούν σε αυτό το κεφάλαιο.

Το Arduino είναι μια προγραμματιστική πλατφόρμα η επιλογή της οποίας για την κατασκευή αναλύθηκε διεξοδικά στο πρώτο κεφάλαιο (χαμηλό κόστος, χαμηλής πολυπλοκότητας γλώσσα προγραμματισμού, μεγάλη υποστήριξη από την κοινότητα κ.α.). Η εμπορική εκμετάλλευση όμως μιας κατασκευής τέτοιου τύπου απαιτεί αυξημένη αξιοπιστία, κάτι που δεν εξασφαλίζει σε αυτό το επίπεδο το Arduino. Απαιτείται αξιοπιστία που μπορούν να προσφέρουν μικροεπεξεργαστές επιπέδου AMR (32 bit), οι οποίοι έχουν μεγάλο αριθμό ακροδεκτών και χαρακτηρίζονται από έμφαση στην ταχύτητα εκτέλεσης εντολών, υψηλή αυτάρκεια περιφερειακών και μεγάλες δυνατότητες εσωτερικής ή εξωτερικής μνήμης προγράμματος (FLASH) και RAM.

Για την επικοινωνία με τον χρήστη επελέγη μια LCD οθόνη χαμηλού κόστους. Σε μια περισσότερο εξελιγμένη μορφή της κατασκευής θα μπορούσε να χρησιμοποιηθεί μια οθόνη αφής η οποία εκτός από την ενημέρωση του χρήστη, θα μπορούσε να παρέχει αμφίδρομη επικοινωνία αντικαθιστώντας το αριθμητικό πληκτρολόγιο. Έτσι, ο χρήστης θα μπορούσε αλλάξει τον προσωπικό κωδικό του, να εισάγει στο σύστημα παραπάνω από ένα τηλέφωνο επικοινωνίας ή ακόμα και να χρησιμοποιήσει τον ενσωματωμένο δέκτη 3G/GPS σαν συσκευή πλοήγησης και κινητό τηλέφωνο.

Το 3G/GPS shield παρέχει τη δυνατότητα για λήψη φωτογραφιών και βίντεο. Αυτό θα μπορούσε να χρησιμοποιηθεί σαν ένα ακόμα μέσο ασφάλειας είτε με χρήση τεχνολογίας αναγνώρισης προσώπου είτε με αποστολή φωτογραφίας στο κινητό τηλέφωνο του ιδιοκτήτη του ατόμου που οδηγεί το όχημα.

Ακόμα το 3G/GPS shield διαθέτει ενσωματωμένη κάρτα μνήμης SD χωρητικότητας έως 8GB. Θα μπορούσε να χρησιμοποιηθεί μελλοντικά για την αποθήκευση των διαδρομών του οχήματος ώστε ο ιδιοκτήτης του οχήματος να έχει στατιστικά δεδομένα για τις διαδρομές που πραγματοποιεί (Μέση ταχύτητα, κατανάλωση καυσίμου, διάρκεια διαδρομής). Ακόμα, σε περίπτωση που παραχωρεί το όχημά του για χρήση από κάποιον τρίτο, θα μπορεί να παρακολουθήσει τα δεδομένα αυτά ώστε να αξιολογήσει τον τρόπο οδήγησής του.

Τέλος, η κατασκευή με τις κατάλληλες μετατροπές, θα μπορούσε να συνδεθεί απ' ευθείας με τον εγκέφαλο του οχήματος και να επιτρέπει πλήρη έλεγχο του μέσω δικτύου 3G. Έτσι, θα μπορούσε ο χρήστης να παρακολουθήσει την πορεία του οχήματος σε πραγματικό χρόνο από την κάμερα του 3G/GPS shield, να απενεργοποιήσει τον κινητήρα σε περίπτωση κλοπής και να επέμβει ακόμα και στο σύστημα διεύθυνσης του σε περίπτωση ανάγκης.

Βιβλιογραφία

- [1] <http://deltahacker.gr/2009/08/01/arduino-intro/>
- [2] <http://arduino.cc/en/Main/arduinoBoardMega>
- [3] <http://el.wikipedia.org/wiki/RFID>
- [4] <http://rfid.gr/%ce%b3%ce%b9%ce%b1-%cf%84%ce%bf-rfid/>
- [5] http://en.wikipedia.org/wiki/Global_Positioning_System
- [6] http://eu.mio.com/el_gr/global-positioning-system_how-does-gps-work.htm
- [7] <http://www.myphone.gr/library/article-37.html#faq-107>
- [8] <http://www.wind.com.gr/gr/wind/upostirixi/i-kiniti-tilefonia-sti-zoi-mas/ola-osa-prepei-na-xeroume-gia-tin-kiniti-tilefonia/>
- [9] <http://arduino.cc/en/Reference/SPI>
- [10] <http://iceal.wikidot.com/ilektriko-systima-aytokiniton>
- [11] <http://www.grobot.gr/phpBB3/viewtopic.php?f=5&t=957>

Τμήμα Αυτοματισμού
Σχολή Τεχνολογικών Εφαρμογών
Α.Τ.Ε.Ι Πειραιά

Copyright© Ψαρράς Ι. Γεώργιος

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας πτυχιακής εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για μη κερδοσκοπικό σκοπό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. *Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.*

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτή την πτυχιακή εργασία εκφράζουν τον συγγραφέα και δε πρέπει να ερμηνευτεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Τμήματος Αυτοματισμού της Σχολής Τεχνολογικών Εφαρμογών του Α.Τ.Ε.Ι Πειραιά