



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ  
ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Εφαρμογή Internet of Things:  
Μέτρηση στάθμης υγρού σε δεξαμενή**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

**ΠΑΝΑΓΙΩΤΗ Κ. ΒΛΟΝΤΖΟΥ**

**Επιβλέπων:** Ευστάθιος Συκάς  
Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2017





ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ  
ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Εφαρμογή Internet of Things:  
Μέτρηση στάθμης υγρού σε δεξαμενή**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

**ΠΑΝΑΓΙΩΤΗ Κ. ΒΛΟΝΤΖΟΥ**

**Επιβλέπων:** Ευστάθιος Συκάς  
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την .... Οκτωβρίου 2017.

.....  
Ευστάθιος Συκάς  
Καθηγητής Ε.Μ.Π.

.....  
Γεώργιος Στασινόπουλος  
Καθηγητής Ε.Μ.Π.

.....  
Ιωάννα Ρουσσάκη  
Επικουρη Καθηγήτρια Ε.Μ.Π.

Αθήνα, Οκτώβριος 2017

.....  
Παναγιώτης Κ. Βλόντζος

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © ΒΛΟΝΤΖΟΣ ΠΑΝΑΓΙΩΤΗΣ, 2017

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, 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 αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

## Περίληψη

Η διπλωματική αυτή εργασία έχει ως σκοπό τη κατασκευή ενός πρωτότυπου οργάνου μέτρησης της στάθμης της δεξαμενής, χρησιμοποιώντας επαφές reed. Η κατασκευή παρουσίασε μεγάλη ακρίβεια στη μέτρηση της στάθμης, χωρίς να υπάρχει η πιθανότητα σφαλμάτων στις μετρήσεις, όπως υπάρχει σε άλλες τεχνολογίες που χρησιμοποιούνται για τη μέτρηση της στάθμης. Οι μετρήσεις συλλέγονται από το μικροελεγκτή C8051F340 της Silicon Labs, ο οποίος είναι τύπου 8051, και στέλνονται μέσω της Bluetooth επικοινωνίας σε android συσκευή, χρησιμοποιώντας την αντίστοιχη εφαρμογή για τη παρουσίασή τους στο χρήστη. Έχοντας ως κύριο στόχο τη χρησιμοποίηση της ολοκληρωμένης αυτής λύσης από καταναλωτές για τη παρακολούθηση της κατανάλωσης του πετρελαίου, οι λειτουργίες που προσφέρονται στον χρήστη είναι η παρουσίαση της κατανάλωσης της κάθε μέρας και η ακριβής μέτρηση της στάθμης κατά τη διάρκεια της παραλαβής. Με βάση την έρευνα αγοράς που πραγματοποιήθηκε στα πλαίσια της διπλωματικής εργασίας, εξαιτίας του υψηλού κόστους των υπάρχοντων λύσεων για τη μέτρηση της στάθμης και χάρη στο χαμηλό κόστος της κατασκευής που υλοποιήθηκε, η ολοκληρωμένη λύση που δημιουργήθηκε μπορεί να προσφέρει σημαντικά περιθώρια κέρδους.

**Λέξεις κλειδιά:** Internet of Things, Μέτρηση στάθμης, 8051, Bluetooth, Android

## **Abstract**

The purpose of this diploma thesis is the creation of an original instrument for measuring the level of the tank using reed contacts. The created instrument has shown high precision in the level measurement, without the possibility of measurement errors, as is the case with other technologies used for level measuring. The measurements are collected by the 8051 type Silicon Labs microcontroller C8051F340 and are sent via Bluetooth communication to an android device using the proper application in order to present these measurements to the user. Since the main purpose of this complete solution is to be used by consumers to monitor oil consumption, the features offered to the user are presenting the daily consumption and accurately measuring the tank level during the process of receiving oil to the tank. Based on the conducted market research, due to the high cost of existing solutions for level measuring and thanks to the low cost of the whole project, this complete solution can offer significant profit margins.

**Keywords:** Internet of Things, Level measuring, 8051, Bluetooth, Android

## **Ευχαριστίες**

Θα ήθελα να ευχαριστήσω θερμά τον καθηγητή της διπλωματικής μου εργασίας κ. Ευστάθιο Συκά που μου εμπιστεύτηκε την ανάληψη της εργασίας αυτής και μου έδωσε την ευκαιρία να ασχοληθώ με ένα τόσο ενδιαφέρον θέμα. Ιδιαίτερες ευχαριστίες θα ήθελα να δώσω επίσης στον Υποψήφιο Διδάκτορα κ. Απόστολο Κοτοπούλη για τη συνεχή καθοδήγησή του και την άριστη επικοινωνία που είχαμε καθ' όλη τη διάρκεια της διπλωματικής αυτής εργασίας.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου και τους φίλους μου για την αγάπη και τη συμπαράσταση που μου παρείχαν σε όλη τη διάρκεια των σπουδών μου.





## Περιεχόμενα

Περίληψη.....	5
Abstract .....	6
Ευχαριστίες .....	7
1 Εισαγωγή.....	13
1.1 Πρόλογος.....	13
1.2 Σκοπός.....	13
1.3 Δομή διπλωματικής εργασίας.....	14
2 Internet of Things.....	15
2.1 Ο μελλοντικός αντίκτυπος του Internet of Things .....	16
2.2 Οι προκλήσεις για την ανάπτυξη του Internet of Things .....	17
2.2.1 Πρότυπα και διαλειτουργικότητα .....	17
2.2.2 Ραδιοφάσμα .....	18
2.2.3 Ασφάλεια .....	18
2.2.4 Ιδιωτικότητα και προστασία των δεδομένων .....	19
2.3 Εφαρμογές Internet of Things .....	19
3 Τεχνολογίες για τη μέτρηση της στάθμης υγρού.....	23
3.1 Αισθητήρες επαφής.....	23
3.1.1 Δείκτης παρακολούθησης στάθμης.....	23
3.1.2 Φλοτέρ.....	24
3.1.3 Εκτοπιστής.....	24
3.1.4 Χωρητικοί αισθητήρες.....	24
3.1.5 Αισθητήρες υπερήχων με επαφή .....	25
3.2 Αισθητές χωρίς επαφή.....	25
3.2.1 Ραντάρ .....	25
3.2.2 Αισθητήρες υπερήχων χωρίς επαφή .....	26
3.3 Τι υπάρχει στην αγορά .....	27
4 Η κατασκευή.....	31
4.1 Οι διακόπτες reed .....	31
4.1.1 Βασική λειτουργία.....	31
4.1.2 Βασικού τύποι διακοπών reed.....	33
4.1.3 Πλεονεκτήματα των διακοπών reed.....	33
4.1.4 AT, η μονάδα μαγνητικής ευαισθησίας των διακοπών reed.....	34
4.1.5 Υστέρηση στους διακόπτες reed .....	34

4.1.6	Διάρκεια ζωής.....	34
4.2	Η ιδέα για τη κατασκευή του οργάνου μέτρησης της στάθμης υγρού.....	34
4.3	Η υλοποίηση της κατασκευής.....	36
5	Ο μικροελεγκτής.....	43
5.1	Ο μικροελεγκτής 8051.....	43
5.1.1	Τα timers του 8051.....	45
5.1.1.1	Πως μετράει ένα timer.....	46
5.1.1.2	Πόσο χρειάζεται ένα timer για να μετρήσει.....	46
5.1.1.3	Καταχωρητές ειδικού σκοπού (Special Function Registers, SFR) των timer.....	46
5.1.1.4	Το SFR TMOD.....	47
5.1.1.5	13-bit Time Mode (mode 0).....	48
5.1.1.6	16-bit Time Mode (mode 1).....	49
5.1.1.7	8-bit Time Mode (mode 2).....	49
5.1.1.8	Split Timer Mode (mode 3).....	49
5.1.1.9	Το SFR TCON.....	49
5.1.1.10	Τα Timers στον C8051F340.....	50
5.1.2	Η σειριακή επικοινωνία στον 8051.....	50
5.1.2.1	Ρύθμιση της λειτουργίας της σειριακής θύρας.....	50
5.1.2.2	Ρύθμιση του ρυθμού μετάδοσης των σειριακών θυρών.....	52
5.1.2.3	Γράφοντας στη σειριακή θύρα.....	52
5.1.2.4	Διαβάζοντας από τη σειριακή θύρα.....	52
5.1.3	Διακοπές (Interrupts) στο 8051.....	53
5.1.3.1	Ποια συμβάντα ενεργοποιούν τα interrupts.....	53
5.1.3.2	Ρυθμίζοντας τα interrupts.....	54
5.1.3.3	Σειριακα interrupts.....	54
5.2	Το εργαλείο μVision του Keil.....	55
5.2.1	μVision IDE.....	55
5.2.1.1	μVision Device Database.....	56
5.2.1.2	μVision Debugger.....	56
5.2.2	Δημιουργία ενός ολοκληρωμένου project στο μVision.....	57
5.3	Η εφαρμογή στο μικροελεγκτή.....	60
5.3.1	Analog to Digital Conversion.....	61
5.3.2	Βαθμονόμηση.....	61
5.3.3	Λειτουργία Real Time Clock (RTC).....	70

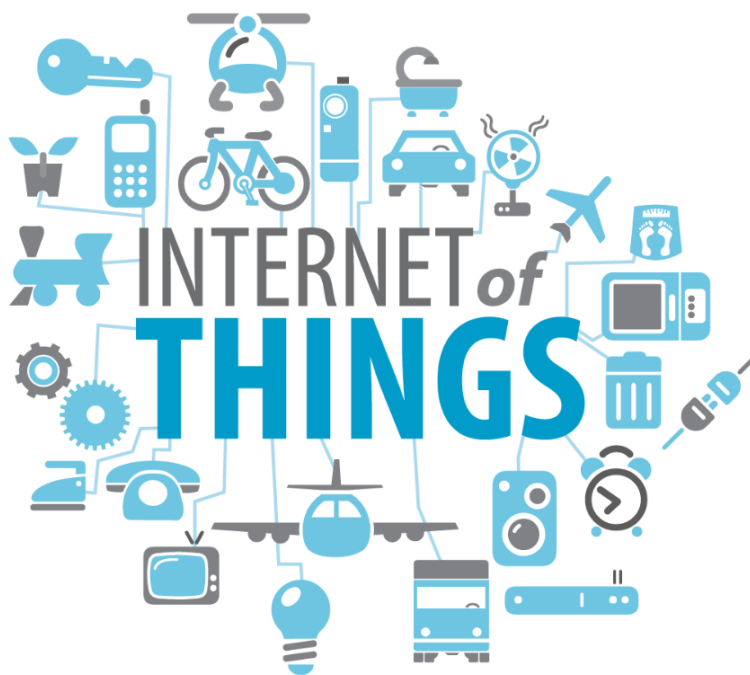
5.3.4	Αποθήκευση μετρήσεων.....	72
5.3.5	Ρύθμιση της σειριακής επικοινωνίας.....	73
5.3.6	Οι επιλογές στην εφαρμογή.....	74
5.3.6.1	Χαρακτήρας «a»: Εισαγωγή τιμών της βαθμονόμησης.....	74
5.3.6.2	Χαρακτήρας «c»: Αρχικοποίηση και ενεργοποίηση του RTC.....	76
5.3.6.3	Χαρακτήρας «m»: Αποστολή των αποθηκευμένων μετρήσεων.....	77
5.3.6.4	Χαρακτήρας «t»: Λειτουργία παραλαβής.....	78
6	Η Bluetooth επικοινωνία.....	81
6.1	Τα τεχνολογικά χαρακτηριστικά του Bluetooth.....	82
6.1.1	Βασικοί και ενισχυμένοι ρυθμοί μετάδοσης.....	83
6.1.2	Υψηλός ρυθμός μετάδοσης με εναλλασσόμενα MAC/PHY.....	84
6.1.3	Χαμηλής κατανάλωσης.....	84
6.2	Η αρχιτεκτονική του Bluetooth.....	85
6.3	Το Bluetooth module HC-05.....	86
6.3.1	Τα χαρακτηριστικά του HC-05.....	87
6.3.2	Περιγραφή των επαφών του HC-05.....	88
6.3.3	Το HC-05 module στη κατασκευή.....	88
7	Η Android εφαρμογή.....	91
7.1	Το λειτουργικό σύστημα Android.....	91
7.1.1	Τα επίπεδα του λογισμικού στο λειτουργικό σύστημα Android.....	92
7.1.1.1	Ο πυρήνας Linux.....	92
7.1.1.2	Βιβλιοθήκες.....	93
7.1.1.3	Περιβάλλον Android Runtime.....	93
7.1.1.4	Πλαίσιο εφαρμογών.....	93
7.1.1.5	Οι Android εφαρμογές.....	94
7.1.2	Οι Android εκδόσεις.....	94
7.2	Το προγραμματιστικό περιβάλλον Android Studio.....	95
7.2.1	Η δομή του project.....	96
7.3	Το σύστημα διαχείρισης βάσεων δεδομένων SQLite.....	97
7.4	Ανάπτυξη της εφαρμογής.....	97
7.4.1	Λήψη των μετρήσεων από το μικροελεγκτή.....	98
7.4.2	Αποθήκευση σε βάση δεδομένων SQLite.....	100
7.4.3	Δημιουργία γραφήματος για τη παρουσίαση των μετρήσεων.....	103
7.5	Παρουσίαση της android εφαρμογής.....	103

7.5.1	Εισαγωγή νέου μετρητή.....	103
7.5.2	Επικοινωνία με το μικροελεγκτή.....	107
7.5.2.1	Λήψη μετρήσεων.....	107
7.5.2.2	Συγχρονισμός του RTC του μικροελεγκτή.....	108
7.5.2.3	Παραλαβή.....	110
7.5.3	Η επικοινωνία με το Bluetooth.....	111
7.5.4	Παρουσίαση των μετρήσεων της κατανάλωσης.....	114
8	Επίλογος.....	119
8.1	Κόστος.....	119
8.2	Παρατηρήσεις.....	119
8.3	Μελλοντικές επεκτάσεις.....	120
	Βιβλιογραφία.....	121

## 1 Εισαγωγή

### 1.1 Πρόλογος

Βρισκόμαστε στην αρχή μιας νέας εποχής, της εποχής του Διαδικτύου των Πραγμάτων ή αλλιώς Internet of Things (IoT, επίσης γνωστό και ως Internet of Objects). Σε γενικές γραμμές, το IoT αναφέρεται στη διασύνδεση καθημερινών αντικειμένων, τα οποία είναι συχνά εξοπλισμένα με ευρέως χρησιμοποιούμενη τεχνολογία και νοημοσύνη. Το IoT θα αυξήσει σημαντικά τον αριθμό των συσκευών στο Διαδίκτυο, ενσωματώνοντας κάθε αντικείμενο για αλληλεπίδραση μέσω ενσωματωμένων συστημάτων, γεγονός που οδηγεί σε ένα εξαιρετικά καταναμημένο δίκτυο συσκευών που επικοινωνούν με ανθρώπους καθώς και με άλλες συσκευές. Χάρη στην ταχεία πρόοδο στις βασικές τεχνολογίες, το IoT δημιουργεί τεράστιες ευκαιρίες για ένα μεγάλο αριθμό καινοτόμων εφαρμογών που υπόσχονται να βελτιώσουν την ποιότητα της ζωής μας. Τα τελευταία χρόνια, το IoT έχει κερδίσει μεγάλη προσοχή από ερευνητές και επαγγελματίες από όλο τον κόσμο.



Εικόνα 1: Internet of Things

### 1.2 Σκοπός

Ο σκοπός της παρούσας διπλωματικής εργασίας είναι η δημιουργία μιας πρωτότυπης και σχετικά χαμηλού κόστους κατασκευής για τη μέτρηση με ακρίβεια της στάθμης του υγρού σε μία δεξαμενή, η οποία θα είναι μία πρακτική εφαρμογή στο τομέα του IoT.

Χρησιμοποιείται ο μικροελεγκτής C8051F340, ο οποίος είναι τύπου 8051, της εταιρίας Silicon Labs για τη λήψη της μέτρησης της στάθμης του υγρού και για τη συλλογή και αποθήκευση των επιθυμητών μετρήσεων. Η επικοινωνία μεταξύ του χρήστη και της

κατασκευής γίνεται μέσω της Bluetooth επικοινωνίας, ενώ η παρουσίαση των μετρήσεων στον χρήστη πραγματοποιείται μέσω μίας εφαρμογής για Android συσκευές.

### 1.3 Δομή διπλωματικής εργασίας

Η διπλωματική αυτή εργασία χωρίζεται σε 8 επιμέρους κεφάλαια:

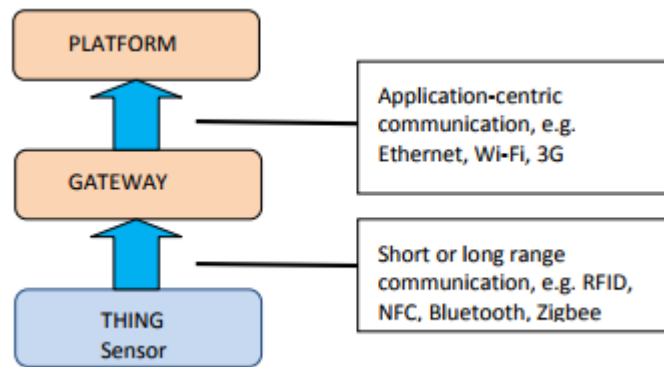
- **Κεφάλαιο 1: Εισαγωγή:** Είναι το παρόν κεφάλαιο όπου γίνεται μία εισαγωγή στο θέμα και το σκοπό της εργασίας, καθώς και μία συνοπτική παρουσίαση των περιεχομένων του κάθε κεφαλαίου.
- **Κεφάλαιο 2: Internet of Things:** Περιγράφεται η σημασία του IoT, καθώς και οι νέες προοπτικές που παρουσιάζονται στο τομέα αυτό.
- **Κεφάλαιο 3: Τεχνολογίες για τη μέτρηση της στάθμης υγρού:** Στο κεφάλαιο αυτό παρουσιάζονται κάποιες υπάρχοντες τεχνολογίες, καθώς και κάποιες έτοιμες λύσεις για τη μέτρηση της στάθμης του υγρού σε δεξαμενές.
- **Κεφάλαιο 4: Η κατασκευή:** Παρουσιάζεται αναλυτικά η ιδέα για τη δημιουργία της πρωτότυπης κατασκευής για τη μέτρηση της στάθμης του υγρού, καθώς και η διαδικασία υλοποίησης της κατασκευής αυτής.
- **Κεφάλαιο 5: Ο μικροελεγκτής:** Εδώ γίνεται εκτενής παρουσίαση της αρχιτεκτονικής των 8051 μικροελεγκτών, περιγράφεται η διαδικασία με την οποία δημιουργήθηκε το project για το μικροελεγκτή που χρησιμοποιήθηκε χάρη στο πρόγραμμα mvision και στη συνέχεια παρουσιάζονται οι λειτουργίες που εκτελεί ο μικροελεγκτής στη συγκεκριμένη κατασκευή.
- **Κεφάλαιο 6: Η Bluetooth επικοινωνία:** Στο κεφάλαιο αυτό γίνεται περιγραφή της Bluetooth επικοινωνίας και παρουσιάζεται το bluetooth module που θα χρησιμοποιηθεί.
- **Κεφάλαιο 7: Η Android εφαρμογή:** Εδώ αρχικά περιγράφονται συνοπτικά το λειτουργικό σύστημα Android και το IDE Android Studio και στη συνέχεια παρουσιάζεται αναλυτικά η εφαρμογή που αναπτύχθηκε για την επικοινωνία με το μικροελεγκτή.
- **Κεφάλαιο 8: Επίλογος:** Αυτό είναι το τελευταίο κεφάλαιο και συζητούνται τα συμπεράσματα και οι παρατηρήσεις για τη διπλωματική αυτή εργασία.

## 2 Internet of Things

Το IoT έχει οριστεί με διάφορους τρόπους. Σε γενικές γραμμές, αναφέρεται σε ένα παγκόσμιο, καταμετρημένο δίκτυο (ή δίκτυα) φυσικών αντικειμένων που είναι ικανά να ανιχνεύουν ή να δρουν στο περιβάλλον τους και να επικοινωνούν μεταξύ τους, με άλλα μηχανήματα ή με υπολογιστές. Αυτά τα «έξυπνα» αντικείμενα έρχονται σε ένα ευρύ φάσμα μεγεθών και δυνατοτήτων, συμπεριλαμβανομένων απλών αντικειμένων με ενσωματωμένους αισθητήρες, οικιακές συσκευές, βιομηχανικά ρομπότ, αυτοκίνητα, τρένα και αντικείμενα που φοριούνται, όπως ρολόγια, βραχιόλια ή πουκάμισα. Η αξία τους έγκειται στις τεράστιες ποσότητες δεδομένων που μπορούν να συλλέξουν και στην ικανότητά τους για επικοινωνία, υποστηρίζοντας έλεγχο σε πραγματικό χρόνο ή ανάλυση δεδομένων που δημιουργεί νέες ιδέες και προτρέπει για νέες ενέργειες.

Η εφαρμογή του IoT σε διάφορους τομείς δημιουργεί επίσης συγκεκριμένους όρους. Τα έξυπνα σπίτια ή τα έξυπνα κτίρια αναφέρονται στις έννοιες του IoT που εφαρμόζονται στη διαχείριση και τον έλεγχο των κτιρίων, συμπεριλαμβανομένων των συστημάτων θέρμανσης, ψύξης, φωτισμού, ψυχαγωγίας, συστημάτων ασφαλείας και οικιακών συσκευών. Οι έξυπνες πόλεις συνήθως χρησιμοποιούν δίκτυα αισθητήρων και υπολογιστών για να μεγιστοποιήσουν την αποτελεσματικότητα της κυκλοφορίας, των δημόσιων συγκοινωνιών, του φωτισμού του δρόμου ή άλλων υποδομών των πόλεων. Το IoT σε βιομηχανικό περιβάλλον (συμπεριλαμβανομένων των βιομηχανιών παροχής υπηρεσιών όπως ο νοσοκομειακός τομέας) μπορεί να αναφέρεται ως Βιομηχανικό IoT (Industrial Internet of Things, IIoT) ή να περιγράφεται ως η αρχιτεκτονική στην οποία βρίσκεται η βιομηχανία 4.0, η επικείμενη (τέταρτη) βιομηχανική επανάσταση.

Η σύνδεση των φυσικών συσκευών με το Διαδίκτυο δεν είναι μια νέα ιδέα. Ωστόσο, η ταχεία μείωση του κόστους των αισθητήρων, το Radio Frequency Identification (RFID) και η μεγαλύτερη κάλυψη και διαθεσιμότητα των ασύρματων και κινητών δικτύων (συμπεριλαμβανομένων των δικτύων Wi-Fi και 2G / 3G / 4G) δημιούργησαν νέες ευκαιρίες. Η έκδοση 6 του πρωτοκόλλου Internet (IPv6) υποστηρίζει επαρκείς διευθύνσεις IP που επιτρέπουν θεωρητικά  $3,4 \times 10^{38}$  συνδεδεμένες στο διαδίκτυο συσκευές. Επιπλέον, το Cloud που είναι προσβάσιμο από το Διαδίκτυο παρέχει πλέον μια εύκολα προσβάσιμη, χαμηλού κόστους πλατφόρμα για την αποθήκευση και επεξεργασία των δεδομένων που συλλέγονται, ενώ τα Big Data εργαλεία υποστηρίζουν την ανάλυση τεράστιων ποσοτήτων δεδομένων. Η υπόσχεση για τα μελλοντικά κοινωνικά και οικονομικά οφέλη που θα μπορούσαν να προσφέρουν αυτές οι εξελίξεις έχει αυξήσει το ενδιαφέρον για το IoT.



Εικόνα 2: Η δομή του IoT

## 2.1 Ο μελλοντικός αντίκτυπος του Internet of Things

Αν και οι εκτιμήσεις ποικίλλουν, οι συμβουλευτικές εταιρείες και άλλοι εμπειρογνώμονες προβλέπουν εκθετική αύξηση του IoT τα επόμενα χρόνια. Σύμφωνα με έρευνες, παγκοσμίως μέχρι το 2020 το IoT θα συνδέσει 26 δισεκατομμύρια συσκευές, οι προμηθευτές προϊόντων και υπηρεσιών IoT θα παράγουν έσοδα άνω των 300 δισεκατομμυρίων δολαρίων και το IoT θα έχει ως αποτέλεσμα προστιθέμενη αξία 1,9 τρισεκατομμύρια δολάρια μέσω πωλήσεων σε διάφορες αγορές.

Αυτή η ταχεία ανάπτυξη βασίζεται στις προσδοκίες ότι το IoT θα προσφέρει σημαντικά οφέλη στις επιχειρήσεις και τους καταναλωτές. Σε μια διαβούλευση που πραγματοποιήθηκε το 2012 από την Ευρωπαϊκή Επιτροπή με πολίτες και οργανισμούς, τρεις στους τέσσερις ερωτηθέντες δήλωσαν ότι αναμένουν οφέλη από τις IoT εφαρμογές, ενώ σε έρευνα που διεξήχθη από το ερευνητικό κέντρο Pew το 2014 περισσότεροι από τέσσερις στους πέντε εμπειρογνώμονες δήλωσαν ότι το IoT θα έχει ευρύτητα, ευεργετικά οφέλη μέχρι το 2025. Αυτά τα οφέλη μπορούν να λάβουν διαφορετικές μορφές για τους πολίτες, τις επιχειρήσεις και τις κυβερνήσεις.

Οι **καταναλωτές** μπορούν να λάβουν περισσότερες προσωπικές προσφορές προϊόντων ή υπηρεσιών, με βάση αυτό που πραγματικά κάνουν ή τη τοποθεσία τους. Μπορούν να ταξιδεύουν πιο αποτελεσματικά, αποφεύγοντας κυκλοφοριακές συμφορήσεις όταν το συνδεδεμένο αυτοκίνητο τους προτείνει μια εναλλακτική διαδρομή με βάση την κυκλοφορία που αναφέρουν άλλα οχήματα. Μπορούν να εξοικονομήσουν χρήματα μειώνοντας τη κατανάλωση ενέργειας ή με την καταβολή χαμηλότερων ασφαλιστρών για το αυτοκίνητο με βάση την επαληθευμένη πραγματοποίηση ασφαλούς οδήγησης. Μπορούν να είναι πιο υγιείς, ασφαλείς και πιο ανεξάρτητοι λόγω φορητών συσκευών που παρέχουν πληροφορίες και προτάσεις για την υγεία ή που παρακολουθούν τους ηλικιωμένους στο σπίτι.

Οι **επιχειρήσεις** μπορούν να παρέχουν καλύτερα προϊόντα και υπηρεσίες, μελετώντας τον τρόπο με τον οποίο συμπεριφέρονται οι πελάτες. Μπορούν επίσης να ανακαλύψουν ανάγκες για νέα προϊόντα ή υπηρεσίες. Μπορούν να προστατεύσουν τα κτίρια μέσω απομακρυσμένης ασφάλειας, να προστατεύσουν περιουσιακά στοιχεία όπως



αυτοκίνητα και μηχανήματα με συσκευές εντοπισμού θέσης και συσκευές απομακρυσμένου κλειδώματος και να διασφαλίσουν ότι ευαίσθητα προϊόντα, όπως π.χ. φαρμακευτικά προϊόντα, αποθηκεύονται με συνέπεια σε σωστές συνθήκες. Μπορούν να γίνουν πιο αποδοτικές, όπως στην περίπτωση των υπηρεσιών κοινής ωφέλειας που χρησιμοποιούν έξυπνες συσκευές μέτρησης για την εξάλειψη των απωλειών ή στην περίπτωση των πωλητών εξοπλισμού που παρέχουν προληπτική συντήρηση. Οι αγρότες μπορούν να είναι πιο παραγωγικοί με έξυπνη άρδευση που παρέχει νερό ακριβώς όπου και όταν χρειάζεται. Η Verizon υπολογίζει ότι μέχρι το 2025 οι οργανώσεις που έχουν αφομοιώσει πλήρως το IoT θα είναι 10% πιο κερδοφόρες από τους ανταγωνιστές που δεν έχουν. Η Εθνική Ακαδημία Επιστημών και Τεχνολογίας της Γερμανίας (acatech) εκτιμά ότι οι επιχειρήσεις θα μπορούσαν να αυξήσουν την παραγωγικότητα κατά 30% εφαρμόζοντας τη βιομηχανία 4.0.

**Οι κυβερνήσεις και οι δημόσιες αρχές** μπορούν επίσης να επωφεληθούν από το IoT. Για παράδειγμα, το κόστος υγείας και μακροχρόνιας περίθαλψης μπορεί να μειωθεί με την καλύτερη απομακρυσμένη υποστήριξη των ηλικιωμένων στα σπίτια τους. Η οδική ασφάλεια μπορεί να βελτιωθεί βάσει δεδομένων που θα παρέχονται από χιλιάδες οδηγούς. Η αποτελεσματικότητα του φωτισμού των δρόμων μπορεί να βελτιωθεί με τη μείωση των φώτων σε άδειους δρόμους.

Φυσικά τα οφέλη για τους καταναλωτές, τις επιχειρήσεις και τις κυβερνήσεις μπορεί να προκύψουν ταυτόχρονα από μια ενιαία εφαρμογή του IoT. Για παράδειγμα, οι έξυπνοι μετρητές που αναφέρουν λεπτομερή δεδομένα σχετικά με τη χρήση ηλεκτρικής ενέργειας μέσω του Διαδικτύου μπορούν να γλιτώσουν τους πελάτες από την αναστάτωση μιας επίσκεψης στο σπίτι από έναν υπάλληλο για τη καταγραφή της κατανάλωσης ηλεκτρικής ενέργειας, να τους δώσουν τα μέσα για να παρακολουθούν (και να μειώσουν) τη κατανάλωση και να τους επιτρέψουν να επωφεληθούν από τα χαμηλότερα τέλη εκτός ωρών αιχμής. Οι κυβερνήσεις και η κοινωνία μπορούν να επωφεληθούν από τη μειωμένη ενεργειακή εξάρτηση και τις λιγότερες αρνητικές περιβαλλοντικές επιπτώσεις.

Βεβαίως, υπάρχουν σημαντικές προκλήσεις που πρέπει να επιλυθούν για την ευρέως εφαρμογή του IoT πριν οι πολίτες, οι επιχειρήσεις και οι κυβερνήσεις μπορούν να επωφεληθούν πλήρως από τα οφέλη που προσφέρει αυτή η νέα τεχνολογία.

## **2.2 Οι προκλήσεις για την ανάπτυξη του Internet of Things**

### **2.2.1 Πρότυπα και διαλειτουργικότητα**

Τα πρότυπα είναι σημαντικά στη δημιουργία αγορών για νέες τεχνολογίες. Εάν οι συσκευές από διαφορετικούς κατασκευαστές δεν χρησιμοποιούν τα ίδια πρότυπα, η διαλειτουργικότητα θα είναι πιο δύσκολη, απαιτώντας επιπλέον πύλες για τη μετάβαση από το ένα πρότυπο στο άλλο. Επιπλέον, μια εταιρεία που ελέγχει διαφορετικά τμήματα μιας αγοράς (π.χ. η απόκτηση δεδομένων, η ενσωμάτωσή τους σε άλλες ροές δεδομένων και η χρήση αυτών των ροών δεδομένων για τη δημιουργία καινοτόμων λύσεων ή για την παροχή υπηρεσιών) μπορεί να κυριαρχήσει σε μια αγορά, περιορίζοντας τον ανταγωνισμό και

δημιουργώντας εμπόδια για τις μικρότερες επιχειρήσεις και οργανισμούς. Τα διαφορετικά πρότυπα δεδομένων μπορεί επίσης να τείνουν τους καταναλωτές στο να χρησιμοποιούν μόνο μία οικογένεια προϊόντων, δημιουργώντας δυσκολίες στη μεταφορά δεδομένων μεταξύ συσκευών διαφορετικών κατασκευαστών στη περίπτωση αλλαγής συσκευής. Η Ευρωπαϊκή Επιτροπή υπογράμμισε την ανάγκη να αναπτυχθούν τεχνολογικά πρότυπα για την υποστήριξη του IoT.

Ωστόσο, μεγάλο μέρος της τρέχουσας ανάπτυξης του IoT πραγματοποιείται σε αγορές όπου συγκεκριμένες τεχνολογικές προσεγγίσεις στη μετάδοση δικτύου και τη τοπολογία μπορούν να υιοθετηθούν (π.χ. μοντέλα όπου κάθε συσκευή συνδέεται απευθείας με το Διαδίκτυο ή αποκεντρωμένα μοντέλα όπου συσκευές που είναι φυσικά κοντά μεταξύ τους μπορούν να ανταλλάξουν πληροφορίες απευθείας). Οι δημόσιες αρχές πρέπει να διαδραματίσουν σημαντικό ρόλο στην προώθηση των προτύπων και της διαλειτουργικότητας όσον αφορά τη μορφή των δεδομένων, τα δίκτυα μεταφοράς και τους μηχανισμούς ασφαλείας, ενθαρρύνοντας τη δημιουργία κοινοπραξιών και ομάδων ανάπτυξης και ενθαρρύνοντας τη συναίνεση σχετικά με τα πρότυπα με άλλες κυβερνήσεις και διεθνείς οργανισμούς τυποποίησης, που απαιτούν ορισμένα πρότυπα.

### **2.2.2 Ραδιοφάσμα**

Η προβλεπόμενη αύξηση του αριθμού των ασύρματων συσκευών θα απαιτήσει μεγαλύτερο φάσμα. Ωστόσο, ο τρόπος με τον οποίο πρέπει να κατανεμηθεί το ραδιοφάσμα εξαρτάται από τον άγνωστο παράγοντα του πως θα χρησιμοποιηθούν οι διάφορες τεχνολογίες όπως η ασύρματη κινητή (χρησιμοποιώντας αδειοδοτημένο φάσμα) ή το Wi-Fi (με χρήση φάσματος χωρίς άδεια) στο IoT. Η κατανομή φάσματος θα πρέπει να εναρμονιστεί σε ολόκληρο το κόσμο. Αυτό που κάνει τις αποφάσεις για το φάσμα ακόμη πιο δύσκολες είναι ότι οι IoT συσκευές τείνουν να έχουν μεγάλη διάρκεια ζωής (περίπου 30 χρόνια) σε αντίθεση με τη πολύ μικρότερη διάρκεια ζωής των κινητών τηλεφώνων (5 έως 7 έτη). Για παράδειγμα, οι φορείς ενδέχεται να αντιμετωπίσουν προβλήματα στο μέλλον όταν τερματίσουν την ασύρματη υποστήριξη 2G, καθώς αυτό θα καθιστούσε παρωχημένους τους έξυπνους μετρητές (smart meters) που χρησιμοποιούν αυτήν την τεχνολογία.

### **2.2.3 Ασφάλεια**

Το IoT παρουσιάζει έναν αριθμό κινδύνων για την ασφάλεια τόσο για τους καταναλωτές όσο και για τις επιχειρήσεις. Οι χάκερς θα μπορούσαν να αποκτήσουν μη εξουσιοδοτημένη πρόσβαση σε συσκευές ή να παρεμποδίσουν τις τοπικές ασύρματες επικοινωνίες προκειμένου να καταγράψουν ευαίσθητα δεδομένα. Θα μπορούσαν επίσης να προσβάλλουν διακομιστές (servers) ή cloud διακομιστές, καθώς οι μεγάλες συγκεντρωμένες ποσότητες δεδομένων τους καθιστούν έναν ελκυστικό στόχο. Οι κακόβουλοι χάκερς θα μπορούσαν να πάρουν τον έλεγχο των τοπικών δικτύων ή συσκευών για να προκαλέσουν βλάβη, όπως για παράδειγμα με την αποστολή ανεπιθύμητων μηνυμάτων, σταματώντας τη λειτουργία ενός εργοστασίου ή του ηλεκτρικού δικτύου, τη διατάραξη της κανονικής

λειτουργίας μιας προσωπικής συσκευής υγείας ή προκαλώντας ένα συνδεδεμένο αυτοκίνητο να αντιδρά με τρόπο που οδηγεί σε σύγκρουση.

Η κρυπτογράφηση των δεδομένων μπορεί να συμβάλει στη μείωση των κινδύνων ασφαλείας, αλλά πολλές από τις συσκευές αισθητήρων που κυκλοφορούν σήμερα στην αγορά δεν διαθέτουν αρκετή χωρητικότητα μπαταρίας ή αρκετή υπολογιστική ικανότητα για την εφαρμογή περίπλοκων τεχνικών κρυπτογράφησης. Παρόλο που δεν υπάρχει σήμερα η τέλεια λύση, οι εταιρείες IoT μπορούν να ελαχιστοποιήσουν τους κινδύνους ασφαλείας λαμβάνοντας υπόψη τα θέματα ασφαλείας κατά τη διάρκεια του σχεδιασμού των προϊόντων τους, όπως συλλέγοντας μόνο βασικά δεδομένα, εφαρμόζοντας πολλαπλά επίπεδα άμυνας κατά των απειλών, δημιουργώντας στοιχεία ελέγχου πρόσβασης για τον περιορισμό της πρόσβασης στη συσκευή τους ή ενημερώνοντας, όπου είναι δυνατόν, τακτικά το λογισμικό για να καλυφθούν τα τρωτά σημεία.

#### **2.2.4 Ιδιωτικότητα και προστασία των δεδομένων**

Οι IoT συσκευές για καταναλωτές συχνά συλλέγουν ιδιωτικά και προσωπικά δεδομένα, όπως πληροφορίες σχετικά με την τοποθεσία των χρηστών, τις καθημερινές τους συνήθειες ή την κατάσταση της υγείας τους. Τα δεδομένα μπορεί να μεταδίδονται σε κεντρικούς διακομιστές ή και να μοιράζονται με άλλες συσκευές ή τρίτους, χωρίς ο χρήστης να έχει την ευκαιρία να ελέγξει τα δεδομένα αυτά. Πολλές συσκευές, όπως φορητά βραχιόλια, δεν διαθέτουν μεγάλες οθόνες ή οθόνες αφής, καθιστώντας δύσκολη την παροχή πληροφοριών στον καταναλωτή σχετικά με πιθανές χρήσεις και τη συγκατάθεση των χρηστών. Η συγκατάθεση δεν απαιτείται μόνο για τη συλλογή δεδομένων, αλλά και για την ανάλυση ή την κοινοποίησή τους σε τρίτους.

Οι επιχειρήσεις μπορούν να ελαχιστοποιήσουν τα προβλήματα συλλέγοντας μόνο τις απαραίτητες πληροφορίες. Ωστόσο, αυτό περιορίζει τη χρησιμότητα των δεδομένων δεδομένου ότι μέρος της χρησιμότητας μπορεί να επιτευχθεί μόνο συνδυάζοντας ένα σύνολο δεδομένων με άλλα ή μέσω της ανάλυσης πολύ μεγάλων ποσοτήτων δεδομένων για τον προσδιορισμό τάσεων. Μία επιλογή των καταναλωτών είναι να μην αλληλεπιδρούν καθόλου με τη συσκευή. Ωστόσο, σε πολλές περιπτώσεις, αυτό σημαίνει ότι ο καταναλωτής δεν μπορεί να επωφεληθεί από μεγάλο μέρος της λειτουργικότητας της συσκευής.

Επιπλέον, η, όσο το δυνατόν νωρίτερα, καθιέρωση των ανώνυμων δεδομένων στην αλυσίδα μεταφοράς και επεξεργασίας δεδομένων μπορεί να συμβάλει στη μείωση των κινδύνων. Ωστόσο, και πάλι, όλες οι συσκευές IoT δεν θα έχουν την απαραίτητη υπολογιστική ισχύ για να το κάνουν και τα δεδομένα ενδέχεται να είναι τόσο εκτεταμένα ώστε η απόλυτη ανωνυμία τους να είναι τεχνικά δύσκολη ή αδύνατη.

### **2.3 Εφαρμογές Internet of Things**

Παρακάτω βλέπουμε κάποιες IoT εφαρμογές σε διάφορους κλάδους:

- **Τεχνολογία για καταναλωτές:** Τα smartphones και τα tablet, τα προσωπικά trackers δραστηριότητας και άλλες φορητές συσκευές, οι έξυπνες οικιακές συσκευές και οι έξυπνοι θερμοστάτες είναι ήδη ευρέως διαθέσιμα και χρησιμοποιούνται. Το Amazon Dash, η συσκευή Wi-Fi που επιτρέπει στους χρήστες να παραγγέλνουν το αγαπημένο τους προϊόν μέσω του Amazon με το πάτημα ενός κουμπιού, όχι μόνο άρχισε κατευθείαν να χρησιμοποιείται ευρέως αλλά σύντομα τροποποιήθηκε από τους χρήστες για να επιτρέπεται η εφαρμογή του και σε άλλα πράγματα, όπως για παραγγελία πίτσας ή για τη κλήση Uber. Μέσα από αυτή την έκθεση σε κίνδυνο, παρουσιάστηκε η ευκαιρία να προσαρμοστεί και να βελτιωθεί. Το Amazon προσφέρει τώρα ένα διαμορφωμένο κουμπί Dash που μπορούν να χρησιμοποιήσουν οι καταναλωτές για να συνδεθούν με μια σειρά από IoT υπηρεσίες. Αυτό είναι μόνο ένα παράδειγμα του πως οι ίδιοι οι καταναλωτές κατευθύνουν την αγορά για την IoT τεχνολογία και τις αναξιολογήτες δυνατότητες εκεί.
- **Ηλεκτρισμός και υπηρεσίες κοινής ωφέλειας:** Η αυτόματη ανίχνευση διακοπών από έξυπνους μετρητές μπορεί να οδηγήσει σε γρηγορότερες επισκευές. Άλλες εξελίξεις στο IoT, όπως η δυνατότητα προγραμματισμού έξυπνων οικιακών συσκευών για τη λειτουργία κατά τη διάρκεια περιόδων χαμηλότερης χρήσης (εκτός δηλαδή ωρών αιχμής), συμβάλλουν στη μείωση της κατανάλωσης ενέργειας των καταναλωτών.
- **Πετρέλαιο και φυσικό αέριο:** Η τεχνολογία IoT βοηθά τις επιχειρήσεις στον τομέα αυτό να αυξήσουν την αποτελεσματικότητά τους μέσω της εξέλιξης της παρακολούθησης της πίεσης, της θερμοκρασίας και του ρυθμού της ροής, καθώς και στη μέτρηση του όγκου και της ακεραιότητας του αγωγού. Οι αισθητήρες στον τομέα αυτό μπορούν να πραγματοποιήσουν έξυπνες προβλέψεις και να βοηθήσουν τις εταιρείες να βελτιστοποιήσουν την παραγωγή τους. Με την καθιέρωση "εταιρειών ψηφιακής τεχνολογίας", οι εταιρείες πετρελαίου και φυσικού αερίου μπορούν να βελτιώσουν περαιτέρω το χρόνο λειτουργίας και τα ποσοστά ανάκτησης του πετρελαίου, να μειώσουν τη διαρροή πετρελαίου, να αυξήσουν την παραγωγικότητα των εργαζομένων, να μειώσουν το κόστος και πολλά άλλα. Για παράδειγμα, μια αμερικανική εταιρεία πετρελαϊκών υπηρεσιών που χρησιμοποιεί προηγμένες τεχνικές γεώτρησης, οι οποίες απαιτούν ιδιαίτερη τεχνογνωσία για τη λειτουργία και τη συντήρησή τους, χρησιμοποιεί συνεργαζόμενες τεχνολογίες, όπως ενοποιημένες επικοινωνίες, για τη παροχή άμεσης καθοδήγησης από εμπειρογνώμονες με σκοπό τη ταχύτερη επίλυση προβλημάτων, οδηγώντας σε μειωμένα κόστη και μειωμένο χρόνο διακοπής για την επιχείρηση.
- **Ασφάλεια:** Οι γεωπεριβαλλοντικές εφαρμογές μπορούν να προειδοποιήσουν τους οδηγούς για ενδεχόμενα σοβαρά καιρικά φαινόμενα (π.χ. χιονοθύελλες), βοηθώντας τους να αποφύγουν τη ζημία των οχημάτων. Οι περιβαλλοντικοί αισθητήρες στους χώρους εργασίας και σε άλλα κτίρια και εγκαταστάσεις χρησιμοποιούνται ήδη για την ανίχνευση της θερμοκρασίας, του καπνού, των τοξικών αναθυμιάσεων, της μούχλας, της σεισμικής κίνησης κ.α.
- **Αυτοκίνητα:** Τα αυτόνομα αυτοκίνητα μπορούν να βοηθήσουν στη μείωση της κυκλοφοριακής συμφόρησης και στην αύξηση της οδικής ασφάλειας. Οι οδικοί

αισθητήρες μπορούν να προειδοποιήσουν τους οδηγούς των εξοπλισμένων με αισθητήρα αυτοκινήτων για βροχή, παγετό και πάγο.

- **Ιατρική:** Η φροντίδα των ασθενών είναι μια προφανής εφαρμογή για τις IoT τεχνολογίες, όπως ο προγραμματισμός ραντεβού ή η παρακολούθηση του διαβήτη για τη διασφάλιση της σωστής δόσης του φαρμάκου. Ο χρόνος διακοπής της ιατρικής συσκευής μπορεί επίσης να μειωθεί μέσω της απομακρυσμένης παρακολούθησης και υποστήριξης. Η τεχνολογία IoT βοηθάει ήδη τα νοσοκομεία να βελτιστοποιήσουν την αλυσίδα εφοδιασμού μειώνοντας παράλληλα τον κίνδυνο: οι πίνακες εφοδιασμού με ενσωματωμένους αναγνώστες RFID με κεραίες μπορούν να καταγράφουν ποια μέλη του προσωπικού έχουν πρόσβαση στο απόθεμα, τι πήραν και πότε.



### 3 Τεχνολογίες για τη μέτρηση της στάθμης υγρού

Όπως αναφέρθηκε, στόχος αυτής της διπλωματικής εργασίας ήταν η δημιουργία μιας πρωτότυπης κατασκευής για τη μέτρηση της στάθμης του υγρού σε μία δεξαμενή. Στο κεφάλαιο αυτό παρουσιάζονται κάποιες υπάρχουσες τεχνολογίες για τη μέτρηση της στάθμης. Οι τεχνολογίες αυτές χωρίζονται σε δύο κύριες κατηγορίες, στις τεχνολογίες που πραγματοποιείται επαφή του αισθητήρα με το υγρό (contact sensors) και στις τεχνολογίες που η μέτρηση πραγματοποιείται χωρίς επαφή του αισθητήρα με το υγρό (non-contact sensors).

#### 3.1 Αισθητήρες επαφής

##### 3.1.1 Δείκτης παρακολούθησης στάθμης

Ο πρώτος τύπος συσκευής μέτρησης στάθμης που πρέπει να εξεταστεί εδώ είναι επίσης και ο πιο απλός. Ένας δείκτης παρακολούθησης στάθμης συνήθως αποτελείται από ένα σωλήνα συνδεδεμένο σε ανοίγματα κοντά στο πάνω και στο κάτω μέρος της δεξαμενής. Ο σωλήνας είναι διάφανος έτσι ώστε ο χειριστής να μπορεί να δει το επίπεδο της στάθμης στο οποίο βρίσκεται το υγρό. Αυτό είναι τεχνικά μέτρηση χωρίς επαφή επειδή δεν υπάρχει επαφή του αισθητήρα με το υγρό. Ωστόσο, πρέπει να ληφθούν υπόψη πολλές από τις ανησυχίες που υπάρχουν και στους αισθητήρες επαφής, όπως η ανάδευση, η διαβρωτικότητα, η κολλητικότητα, η θερμοκρασία, η πίεση κ.τ.λ., που θα μπορούσαν να προκαλέσουν βλάβη ή διαστρέβλωση του δείκτη.



Εικόνα 3: Παράδειγμα δείκτη παρακολούθησης στάθμης

Παρόλο που αυτή η μέθοδος μέτρησης στάθμης μπορεί να θεωρηθεί ως η πιο αξιόπιστη επειδή το επίπεδο του υγρού μπορεί πράγματι να παρατηρηθεί από τον χειριστή, οι διαφορές πίεσης ή θερμοκρασίας μεταξύ της δεξαμενής και της περιοχής στην οποία βρίσκεται ο δείκτης μπορούν να επηρεάσουν την ακρίβεια μέτρησης. Οι δείκτες παρακολούθησης στάθμης μπορούν να χρησιμοποιηθούν μόνο όταν μετρώνται υγρά, καθώς στερεά υλικά δεν μπορούν να ρέουν μέσω του δείκτη. Επίσης, δεν υπάρχει τρόπος να ενσωματωθεί κανένας τύπος αυτοματισμού στη διαδικασία χρησιμοποιώντας αυτόν τον τύπο συσκευής.

### **3.1.2 Φλοτέρ**

Άλλες συσκευές, όπως τα φλοτέρ και οι εκτοπιστές, βασίζονται στη πυκνότητα του υλικού για τη μέτρηση της στάθμης. Συχνά χρησιμοποιούν ένα μετρητή παρόμοιο με το δείκτη που είδαμε προηγουμένως (ένας σωλήνας που συνδέεται στην πλευρά της δεξαμενής στη κορυφή και το κάτω μέρος). Οι τύποι αισθητήρων με φλοτέρ βασίζονται στη μέτρηση της επέκτασης και της συστολής ενός συνδεδεμένου καλωδίου στο δείκτη. Αυτός ο τύπος αισθητήρα δεν απαιτεί ενέργεια για να λειτουργήσει αλλά είναι επιρρεπής σε βλάβη από την εμπλοκή του φλοτέρ ή την προσκόλληση.

### **3.1.3 Εκτοπιστής**

Οι συσκευές εκτόπισης είναι ανόμοιες με τις διατάξεις με φλοτέρ, επειδή είναι πυκνότερες από το υγρό και αναρτώνται από ένα ελατήριο. Βασίζονται στην αρχή του Αρχιμήδη για να ανιχνεύσουν τις μεταβολές της δύναμης άνωσης, η οποία είναι ίση με το βάρος του εκτοπισμένου υγρού, που προκαλείται από την άνοδο ή την πτώση της στάθμης. Ένας αισθητήρας στην κορυφή του μετρητή μετρά την αλλαγή στο φαινομενικό βάρος του εκτοπιστή. Και οι δύο τεχνολογίες, οι εκτοπιστές και τα φλοτέρ, έχουν τους ίδιους περιορισμούς και προβλήματα που συναντιούνται στους δείκτες παρακολούθησης στάθμης που είδαμε προηγουμένως.

### **3.1.4 Χωρητικοί αισθητήρες**

Οι χωρητικοί αισθητήρες χρησιμοποιούν την ίδια αρχή λειτουργίας με έναν ηλεκτρονικό πυκνωτή για να μετρήσουν το επίπεδο της στάθμης. Ένας ηλεκτρονικός πυκνωτής αποτελείται από δύο αγωγικές μεταλλικές πλάκες που χωρίζονται από κάποιο είδος μονωτικού υλικού. Η χωρητικότητα είναι η μέτρηση της ποσότητας της ενέργειας που μπορεί να αποθηκεύσει ένας πυκνωτής. Χρησιμοποιώντας μια μέθοδο που είναι πολύ παρόμοια με αυτή ενός ηλεκτρονικού πυκνωτή, η χωρητικότητα ενός υλικού διεργασίας μπορεί επίσης να μετρηθεί και στη συνέχεια να συσχετιστεί με το επίπεδο της στάθμης.

Εάν μετράται ένα μη αγωγικό υλικό, το υλικό αυτό χρησιμοποιείται ως το μονωτικό τμήμα του πυκνωτή και το τοίχωμα της δεξαμενής ενεργεί ως η δεύτερη αγωγική πλάκα. Εάν μετράται ένα αγωγικό υλικό, ο αισθητήρας είναι ενισχυμένος με έναν μονωτήρα και το υλικό ενεργεί ως η άλλη αγωγική πλάκα. Και στις δύο περιπτώσεις, η άνοδος του επιπέδου της



στάθμης θα συσχετιστεί με την αύξηση της μετρούμενης χωρητικότητας (σε σχέση με εκείνη του αέρα).

### **3.1.5 Αισθητήρες υπερήχων με επαφή**

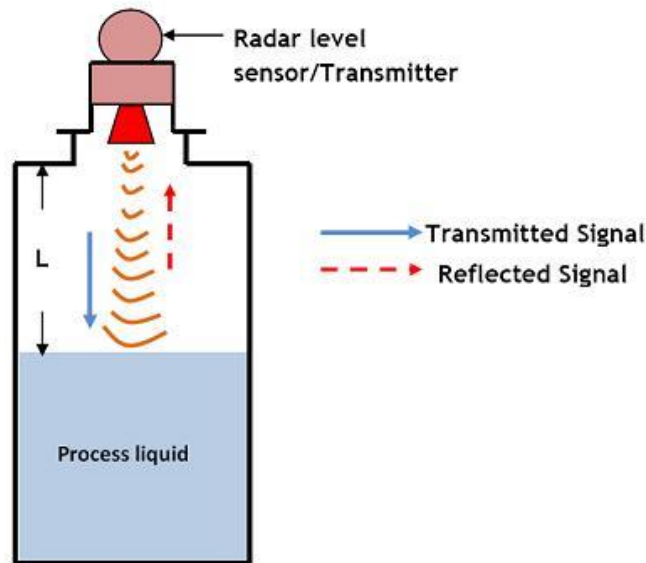
Οι δονούμενοι ή υπερηχητικοί διακόπτες στάθμης λειτουργούν σύμφωνα με την αρχή ότι ο αισθητήρας θα δονείται στη συχνότητα συντονισμού του όταν δεν είναι βυθισμένος στο υγρό. Όταν η συχνότητα αυτή εντοπιστεί, η στάθμη του υγρού θα έχει φτάσει στο επίπεδο του διακόπτη. Αυτή η τεχνική λειτουργεί με υγρά και κόκκους, αν και πρέπει να ληφθεί υπόψη η επικάλυψη ή τα διαβρωτικά υλικά. Οι δονούμενοι διακόπτες στάθμης τυπικά χρησιμοποιούνται μόνο για τη μέτρηση του επιπέδου ενός σημείου, σε αντίθεση με τη συνεχή μέτρηση στάθμης, και για την ενεργοποίηση συναγερμών στα χαμηλά ή υψηλά επίπεδα στάθμης. Οι ίδιοι οι αισθητήρες τείνουν να είναι εύθραυστοι λόγω του αναγκαίου επιπέδου ακρίβειας.

## **3.2 Αισθητές χωρίς επαφή**

### **3.2.1 Ραντάρ**

Το ραντάρ λειτουργεί συνήθως εκπέμποντας ηλεκτρομαγνητικούς παλμούς προς την κατεύθυνση ενός αντικειμένου, περιμένοντας τον παλμό να ανακλαστεί στο αντικείμενο και να επιστρέψει στην πηγή, μετρώντας το χρόνο μέχρι να ολοκληρωθεί η διαδικασία αυτή. Η απόσταση μεταξύ της πηγής του παλμού και του αντικειμένου μπορεί να υπολογιστεί ως το αποτέλεσμα του μισού χρόνου του ταξιδιού και της ταχύτητας του φωτός.

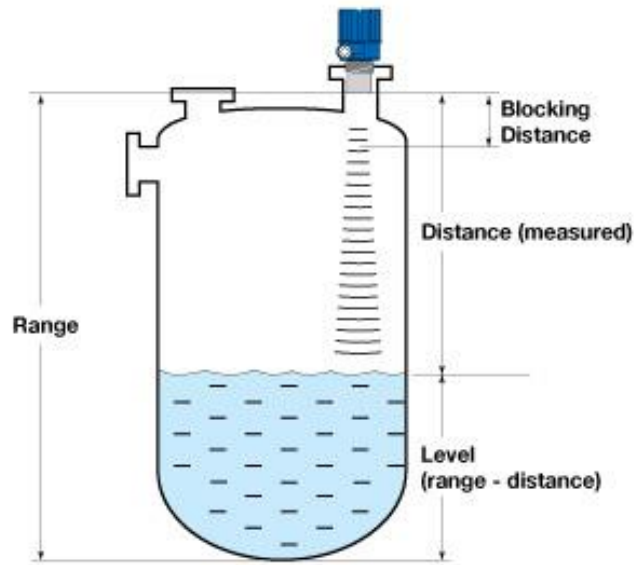
Για τη χρήση σε μέτρηση στάθμης, ένας πομποδέκτης ραντάρ αναρτάται από την κορυφή της δεξαμενής και μετρά την απόσταση από τη κορυφή της στάθμης του υγρού. Το συνολικό μήκος της δεξαμενής μπορεί στη συνέχεια να χρησιμοποιηθεί για τον υπολογισμό της στάθμης. Ως αισθητήρας χωρίς επαφή, το ραντάρ έχει το ξεχωριστό πλεονέκτημα ότι δεν επηρεάζεται από την κατάσταση του υλικού, όπως η ανάδευση, η διαβρωτικότητα, η κολλητικότητα, η θερμοκρασία, η πίεση κλπ. Τα ραντάρ μπορούν να χρησιμοποιηθούν για τη μέτρηση υγρών και κοκκωδών στερεών. Ωστόσο, τα υλικά με υψηλή αγωγιμότητα τείνουν να είναι τα καλύτερα υποψήφια για τη μέτρηση στάθμης με ραντάρ επειδή αντανακλούν περισσότερο το εκπεμπόμενο ραδιοσήμα. Τα αγωγιμα υλικά εκτός του υλικού μέσα στη δεξαμενή, όπως οι αναδευτήρες, μπορούν να προκαλέσουν παρεμβολή στο σήμα ραντάρ.



Εικόνα 4: Απεικόνιση της μέτρησης στάθμης με ραντάρ

### 3.2.2 Αισθητήρες υπερήχων χωρίς επαφή

Οι αισθητήρες στάθμης υπερήχων χωρίς επαφή χρησιμοποιούν μια πολύ παρόμοια μέθοδο με τους αισθητήρες ραντάρ για να μετρήσουν το επίπεδο της στάθμης. Ωστόσο, αντί να χρησιμοποιούν ραδιοκύματα, χρησιμοποιούν ηχητικά κύματα και η απόσταση υπολογίζεται ως το προϊόν του μισού χρόνου του ταξιδιού και της ταχύτητας του ήχου. Σε αντίθεση με την ταχύτητα του φωτός, η ταχύτητα του ήχου εξαρτάται από τη θερμοκρασία και επομένως η θερμοκρασία της δεξαμενής πρέπει να μετράται και να λαμβάνεται υπόψη. Όπως το ραντάρ, οι αισθητήρες στάθμης υπερήχων χωρίς επαφή μπορούν να χρησιμοποιηθούν για τη μέτρηση υγρών και κοκκωδών στερεών. Τα υλικά που παράγουν ισχυρότερη ηχητική αντανάκλαση μπορούν να εφαρμοστούν καλύτερα σε αυτόν τον τύπο μέτρησης. Η συμπύκνωση, η συσσώρευση σκόνης και η παρουσία πρόσθετων αντικειμένων μέσα στη δεξαμενή μπορούν να προκαλέσουν ανακρίβειες μέτρησης.



Εικόνα 5: Απεικόνιση της μέτρησης με αισθητήρες στάθμης υπερήχων χωρίς επαφή

### 3.3 Τι υπάρχει στην αγορά

Κάνοντας μία έρευνα αγοράς για την εύρεση έτοιμων λύσεων για τη μέτρηση της στάθμης σε μία δεξαμενή, είδαμε ότι αν και οι λύσεις που προσφέρονται είναι πολλές, παρουσιάζουν συγκεκριμένα μειονεκτήματα, κάποια από τα οποία είναι η υψηλή τιμή, η χαμηλή ακρίβεια, η μη δυνατότητα αυτοματισμού κ.α.

Ένας αριθμός προϊόντων που είναι σχετικά φθηνά ανήκουν στη 1<sup>η</sup> κατηγορία αισθητήρων που είδαμε. Χρησιμοποιούν δηλαδή τη τεχνολογία με τον απλό δείκτη παρακολούθησης της στάθμης που όμως έχει τα μειονεκτήματα ότι δεν παρουσιάζει μεγάλη ακρίβεια στη μέτρηση και της μη δυνατότητας κάποιου αυτοματισμού. Παρακάτω βλέπουμε κάποια από τα προϊόντα αυτής της κατηγορίας που προσφέρονται:

## ΜΕΤΡΗΤΗΣ ΣΤΑΘΜΗΣ ΠΕΤΡΕΛΑΙΟΥ 2m



Μάρκα: SKAR  
Κωδικός Προϊόντος: ΜΕΤΡΗΤΗΣ ΣΤΑΘΜΗΣ ΠΕΤΡΕΛΑΙΟΥ 2m  
Διαθεσιμότητα: Διαθέσιμο

Τιμή: 102,09€  
Άνευ ΦΠΑ: 83,00€

Ποσότητα:

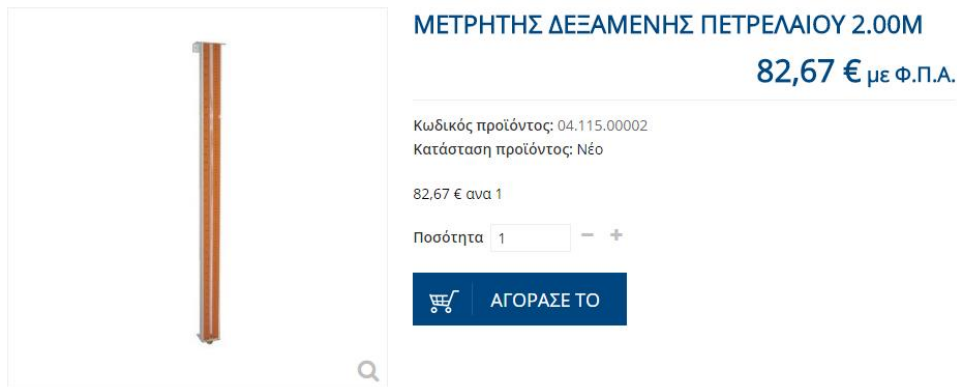
[Αγοράστε το](#)

- 1 - [Επιθυμητό Σύνκριση](#)

★★★★★ [0 κριτικές](#) | [Γράψτε μια κριτική](#)

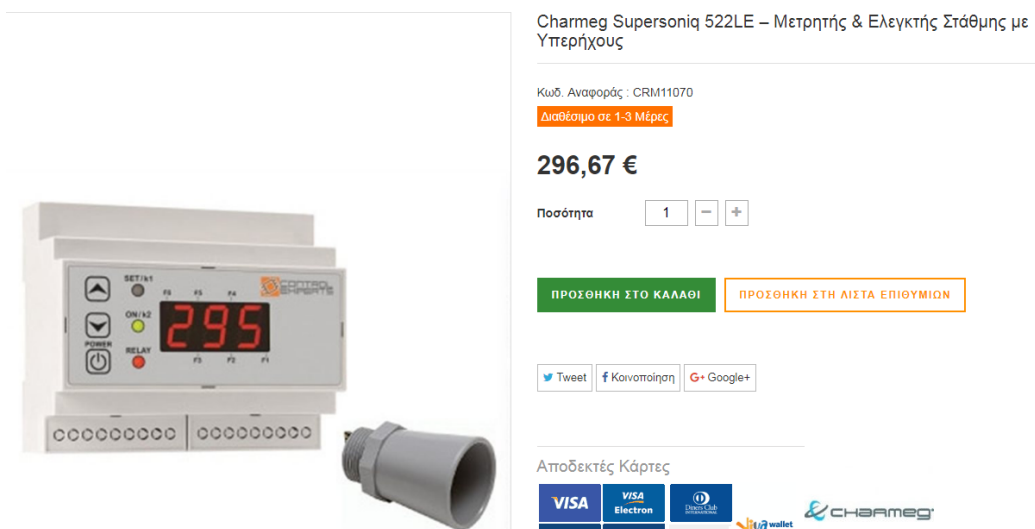
[Μοιραστείτε](#)

Εικόνα 6: Δείκτης στάθμης μου προσφέρεται από ελληνικό κατάστημα

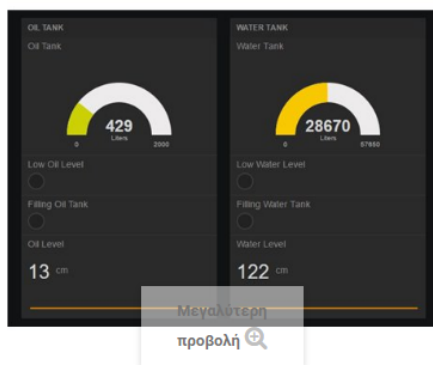


Εικόνα 7: Δείκτης στάθμης μου προσφέρεται από ελληνικό κατάστημα

Τα προϊόντα που συναντιούνται περισσότερο και προσφέρουν τη δυνατότητα συνεχούς παρακολούθησης της στάθμης με τη δυνατότητα κάποιου αυτοματισμού είναι τα προϊόντα που περιέχουν αισθητήρες υπερήχων. Η τιμή αυτών των προϊόντων όμως είναι σχετικά υψηλή, ενώ τα μειονεκτήματα που παρουσιάζει η μέτρηση με υπερήχους, όπως παρουσιάστηκε παραπάνω, παραμένουν. Κάποια από τα προϊόντα της κατηγορίας αυτής τα βλέπουμε παρακάτω:



Εικόνα 8: Μετρητής στάθμης με υπερήχους με τη δυνατότητα παρακολούθησης μέσω ενσωματωμένης οθόνης



Στείλτε σ' ένα φίλο Εκτύπωση

## ΜΕΤΡΗΤΗΣ ΣΤΑΘΜΗΣ ΔΕΞΑΜΕΝΗΣ VIC4\_ip

Κωδικός είδους: 1780

Σύστημα VIC4\_ip παρακολούθησης μέσω του internet, της στάθμης του υγρού σε δεξαμενές νερού, πετρελαίου κλπ. με υπέρηχους. Ελέγξτε τη κατάσταση της δεξαμενής σας από το κινητό σας τηλέφωνο. Απομακρυσμένος έλεγχος δεξαμενών.

4 αντικείμενα **Άμεσα Διαθέσιμο**

Tweet Κοινή χρήση Google+ Pinterest

Γράψτε μια κριτική

Manufacturer  
ASD & Συνεργάτες



Αγοράζοντας αυτό το προϊόν μπορείτε να μαζέψετε μέχρι και **45 πόντοι επιβράβευσης**. Το καλάθι σου θα είναι συνολικά **45 πόντοι** που μπορούν να μετασχηματιστούν σε κουπόνι αξίας 4,50 €.

**453,66 €**

Εικόνα 9: Μετρητής στάθμης με υπέρηχους με τη δυνατότητα online παρακολούθησης της στάθμης

Στο 1<sup>ο</sup> προϊόν που είδαμε, η τιμή δεν ήταν σίγουρα χαμηλή, ενώ η παρακολούθηση της στάθμης γίνονταν τοπικά μέσω οθόνης lcd. Στο 2<sup>ο</sup> προϊόν, προσφέρεται η δυνατότητα online παρακολούθησης της στάθμης, η τιμή αυξάνεται όμως ακόμα παραπάνω.

Από τις διάφορες τεχνολογίες που είδαμε, η πιο αξιόπιστη τεχνολογία για τη μέτρηση της στάθμης ήταν αυτή με αισθητήρα ραντάρ. Οι τιμές όμως των προϊόντων που ανήκουν σε αυτή τη κατηγορία είναι πολύ υψηλές, καθώς τα προϊόντα αυτά προσφέρονται για κυρίως βιομηχανική χρήση. Ένα παράδειγμα το βλέπουμε παρακάτω:

## Flowline, Vertical Mounting Level Sensor, -40°C to +130°C, LR25 Series

Κωδικός Προϊόντος της RS 889-1157  
Εμπορικό σήμα Flowline  
Κωδικός Κατασκευαστή LR25-0310-40

FLOWLINE



Temporarily out of stock

**€ 4.190,53**  
Μονάδας

Ποσότητα	Τιμή Μονάδας
1 +	€ 4.190,53

Ποσότητα  units **Αγορά**  
Προσθήκη σε λίστα

**Τεχνικά Χαρακτηριστικά**

- Πιστοποιητικό Συμμόρφωσης προς την RoHS
- Τεχνικά Χαρακτηριστικά
- Δήλωση συμμόρφωσης

Εικόνα 10: Μετρητής στάθμης με ραντάρ που προσφέρεται από ελληνικό κατάστημα

Κάτι σημαντικό που επίσης παρατηρήθηκε μέσα από την έρευνα αγοράς, ήταν ότι υπάρχει αρκετά μεγάλη ζήτηση για μετρητές που μετρούν τη ροή του πετρελαίου κατά τη

διάρκεια της παραλαβής πετρελαίου στις δεξαμενές των καταναλωτών. Κάποια από αυτά τα προϊόντα που δίνουν λύση στο πρόβλημα αυτό φαίνονται παρακάτω:



Μετρητής Πετρελαίου MPS Plastic

Μοντέλο Μετρητές Πετρελαίου  
Κατάσταση Νέο

Οι μετρητές MPS συμβάλλουν στην προσπάθεια επίλυσης του προβλήματος ελλιπούς παραλαβής καυσίμου από τους τελικούς πελάτες και παρέχουν τη δυνατότητα για ακριβή έλεγχο της ποσότητας και της ποιότητας του καυσίμου

181,46 €

Ποσότητα  
1

Αγορά

VISA MasterCard Eurobank  
Ασφαλείς Πληρωμές On-Line  
Μέσω της τράπεζας EUROBANK

Γράψτε μας εντυπώσεις

Εικόνα 11: Μετρητής ροής πετρελαίου

Ηλεκτρονικός μετρητής πετρελαίου 1" Piusi K24 AdBlue™ Flow Meter made in italy  
» ΜΗΧΑΝΗΜΑΤΑ ΠΡΑΘΗΡΙΩΝ, ΠΛΥΝΤΗΡΙΩΝ » Μετρητές πετρελαίου μηχανικός/ηλεκτρονικός



Κάντε κλικ εδώ για μεγέθυνση

Κωδ.: BR36020-1 | ΤΙΜΗ: 308.94€ 154.47€ | Ποσότητα: 1 | Αγορά

Εικόνα 12: Μετρητής ροής πετρελαίου

Όπως βλέπουμε, οι τιμές των προϊόντων αυτών είναι σχετικά υψηλές, ενώ οι λειτουργίες που προσφέρουν στους καταναλωτές είναι περιορισμένες.

## 4 Η κατασκευή

### 4.1 Οι διακόπτες reed

Ένας διακόπτης reed αποτελείται από δύο ή τρεις μεταλλικές προεξέχουσες επαφές (ή λεπίδες) που είναι ερμητικά σφραγισμένες μέσα σε ένα γυάλινο σωλήνα που περιέχει μια αδρανή ατμόσφαιρα. Οι διακόπτες τύπου reed διατίθενται σε διάφορα μεγέθη, μαγνητικές ευαισθησίες, δυνατότητες υψηλής ισχύος, δυνατότητες υψηλής τάσης, ρυθμίσεις επαφών και διατάξεις μολύβδου.

#### 4.1.1 Βασική λειτουργία

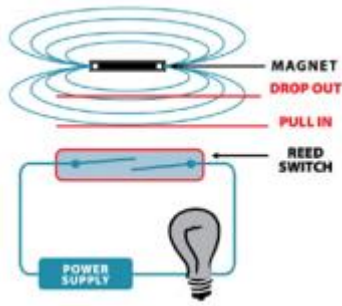
Οι διακόπτες reed έχουν τρεις βασικές ρυθμίσεις: κανονικά ανοιχτές (Normally Open, NO), κανονικά κλειστές (Normally Closed, NC) και μορφή C οι οποίες έχουν τόσο NO όσο και NC. Ανάλογα με αυτή την αρχική διαμόρφωση της επαφής, οι επαφές είτε θα κλείσουν είτε θα ανοίξουν ένα κύκλωμα όταν ενεργοποιηθούν από ένα μαγνητικό πεδίο.



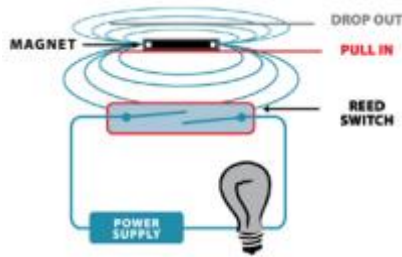
Εικόνα 13: Normally Open διακόπτης reed

Οι επαφές των διακοπών reed λειτουργούν ως μαγνητικοί αγωγοί ή ως μαγνητικές κεραίες. Όταν εκτίθενται σε εξωτερικό μαγνητικό πεδίο από μαγνήτη ή από ηλεκτρομαγνητικό πηνίο, δημιουργούνται πόλοι αντίθετης πολικότητας κατά μήκος της ανοικτής επαφής. Όταν η μαγνητική δύναμη υπερβαίνει τη δύναμη που συγκρατεί τις reed επαφές, οι επαφές θα κλείσουν (Pull-In). Όταν η μαγνητική δύναμη του μαγνητικού πεδίου είναι μικρότερη από την δύναμη που συγκρατεί τις επαφές, οι επαφές θα ανοίξουν (Drop-Out).

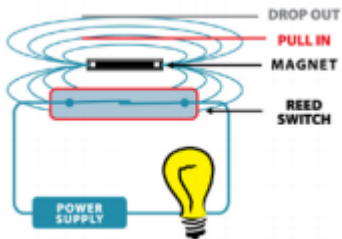
Στις 3 παρακάτω εικόνες φαίνεται η διαδικασία που ακολουθείται όταν ένας μαγνήτης έρχεται κοντά σε ένα κύκλωμα με ένα διακόπτη reed τύπου NO (ανοιχτού).



Εικόνα 14: Ο μαγνήτης πλησιάζει το διακόπτη

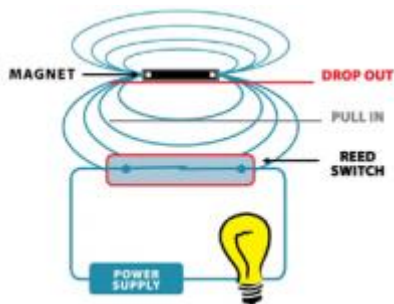


Εικόνα 15: Ο μαγνήτης ενεργοποιεί το διακόπτη



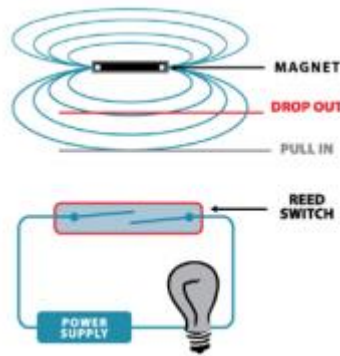
Εικόνα 16: Ο μαγνήτης κλείνει το διακόπτη και το κύκλωμα ολοκληρώνεται (το φως ανάβει)

Στις παρακάτω εικόνες φαίνεται τι συμβαίνει στο παραπάνω κύκλωμα όταν ο μαγνήτης απομακρυνθεί από το διακόπτη.



Εικόνα 17: Ο μαγνήτης απομακρύνεται, ενώ παράλληλα ο διακόπτης παραμένει κλειστός



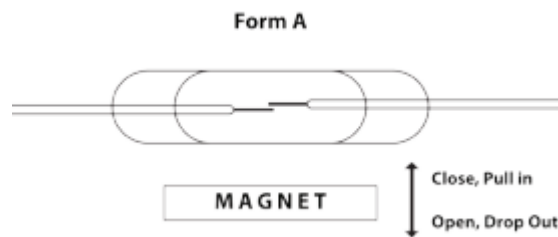


Εικόνα 18: Ο μαγνήτης έχει απομακρυνθεί εντελώς και ο διακόπτης ξαναανοίγει

#### 4.1.2 Βασικοί τύποι διακοπών reed

Υπάρχουν τρεις βασικοί τύποι διακοπών reed.

- Τύπος A, SPST (Single pole, single throw). Οι κανονικά ανοιχτές επαφές του διακόπτη reed κλείνουν με την παρουσία μαγνητικού πεδίου.



Εικόνα 19: Τύπος A

- Τύπος B, SPST (Single pole, single throw). Οι κανονικά κλειστές επαφές διακόπτη reed ανοίγουν στη παρουσία μαγνητικού πεδίου.
- Τύπος C, SPDT (Single pole, double throw). Σε αυτό το διακόπτη reed η κανονικά κλειστή επαφή ανοίγει και στη συνέχεια η κανονικά ανοιχτή επαφή κλείνει με την παρουσία μαγνητικού πεδίου.

#### 4.1.3 Πλεονεκτήματα των διακοπών reed

Οι διακόπτες reed έχουν τα παρακάτω πλεονεκτήματα.

- Εναλλαγή μεταξύ κλειστών και ανοιχτών επαφών χωρίς την ανάγκη απευθείας επαφής
- Αρκετά οικονομικοί
- Δεν υπάρχει η απαίτηση για κάποια ελάχιστη τιμή ρεύματος προκειμένου να επιτευχθεί η εναλλαγή
- Υψηλά αξιόπιστη εναλλαγή
- Μικρή εσωτερική αντίσταση (από 0,07 έως 0,02 Ohm)

- Κατάλληλοι για προστατευμένο και σκληρό περιβάλλον, καθώς οι επαφές είναι ερμητικά σφραγισμένες

#### 4.1.4 AT, η μονάδα μαγνητικής ευαισθησίας των διακοπών reed

Το AT σημαίνει Ampere-Turns και είναι μια τυπική μονάδα που κατηγοριοποιεί τη μαγνητική ευαισθησία των διακοπών reed. Η τιμή AT μετράται με κεντράρισμα του διακόπτη reed σε ένα τυπικό πηνίο, στο οποίο εφαρμόζεται ένα αυξανόμενο ρεύμα. Σε ένα συγκεκριμένο ρεύμα (Amperes) ο διακόπτης reed κλείνει μέσω του μαγνητικού πεδίου που παράγεται από το πηνίο. Αυτή είναι η τιμή λειτουργίας (ή Pull In), υπολογιζόμενη ως:

$$\text{Amperes} \times \text{αριθμός σπειροειδών στροφών} = AT$$

Μειώνοντας το ρεύμα που διέρχεται από το πηνίο μέχρι να ξανανοίξει ο διακόπτης, μπορεί να επιτευχθεί η τιμή απελευθέρωσης (ή Drop Out).

Υψηλή τιμή AT σημαίνει χαμηλή ευαισθησία. Έτσι, είναι απαραίτητο ένα ισχυρό μαγνητικό πεδίο για τη λειτουργία του διακόπτη reed (αντίστοιχα η απόσταση μεταξύ του μαγνήτη και του διακόπτη με την οποία επιτυγχάνεται η εναλλαγή θα είναι μικρότερη).

Χαμηλή τιμή AT σημαίνει υψηλή ευαισθησία, επιτρέποντας την ενεργοποίηση του διακόπτη ακόμα και με ασθενέστερο μαγνητικό πεδίο (αντίστοιχα, η απόσταση μεταξύ του μαγνήτη και του διακόπτη με την οποία επιτυγχάνεται η εναλλαγή μπορεί να αυξηθεί).

#### 4.1.5 Υστέρηση στους διακόπτες reed

Υστέρηση είναι η διαφορά μεταξύ AT λειτουργίας και AT απελευθέρωσης ενός διακόπτη reed. Π.χ. μια επαφή διακόπτη reed κλείνει πλησιάζοντας έναν μαγνήτη σε απόσταση 10 mm. Ωστόσο, ο διακόπτης θα ανοίξει εκ νέου σε απόσταση από τον μαγνήτη 12 mm.

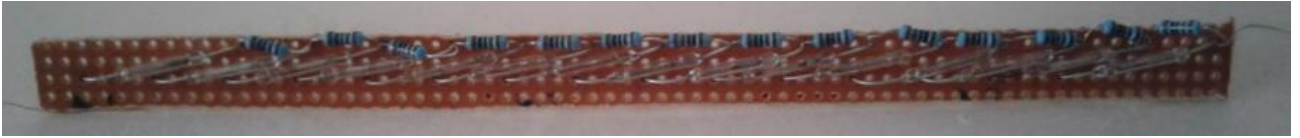
#### 4.1.6 Διάρκεια ζωής

Η διάρκεια ζωής εξαρτάται από τις συνθήκες φορτίου. Με μοναδικά φορτία τα φορτία που συνεισφέρουν στην εναλλαγή, η διάρκεια ζωής είναι εκατοντάδες εκατομμύρια έως μερικά δισεκατομμύρια κύκλοι εναλλαγής. Με υψηλότερα φορτία (π.χ. υψηλά ρεύματα) η διάρκεια ζωής κυμαίνεται από 10 000 έως μερικά εκατομμύρια κύκλους εναλλαγής. Συνήθως η διάρκεια ζωής του διακόπτη reed είναι πολύ μεγαλύτερη από τη διάρκεια ζωής της συσκευής στην οποία είναι ενσωματωμένος ο διακόπτης reed.

## 4.2 Η ιδέα για τη κατασκευή του οργάνου μέτρησης της στάθμης υγρού

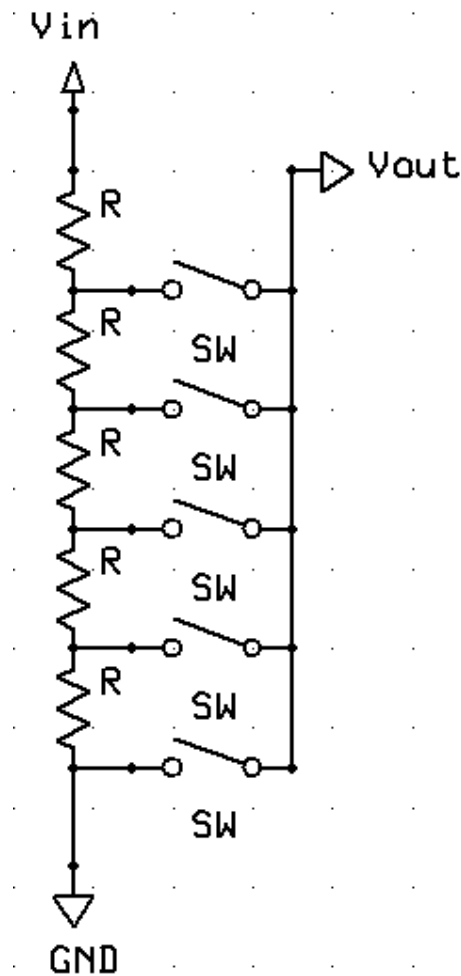
Στόχος ήταν η δημιουργία ενός πρωτότυπου οργάνου μέτρησης της στάθμης υγρού. Πραγματοποιήθηκε έτσι η ιδέα ενός κυκλώματος με ένα διαιρέτη τάσης μέσα σε ένα

ηλεκτρολογικό σωλήνα, όπου είναι τοποθετημένες αντιστάσεις σε σειρά, με κάθε αντίσταση να είναι συνδεδεμένη με ένα μαγνητικό διακόπτη reed τύπου NO (Normally open, κανονικά ανοιχτού).



Εικόνα 20: Ένα μέρος της κατασκευής του διαιρέτη τάσης

Οι διακόπτες reed είναι με τη σειρά τους συνδεδεμένοι με τη τάση εξόδου του διαιρέτη τάσης. Το κύκλωμα αυτό το βλέπουμε παρακάτω (για 5 αντιστάσεις και διακόπτες reed):



Εικόνα 21: Κύκλωμα διαιρέτη τάσης με 5 αντιστάσεις και διακόπτες reed

Όπως παρατηρούμε, χρησιμοποιώντας το τύπο για το διαιρέτη τάσης:

$$V_{out} = \frac{R_i}{R_{ολ}} * V_{in}$$

η τάση εξόδου θα εξαρτάται από το ποιος διακόπτης reed είναι κλειστός. Τοποθετώντας έτσι στη πλακέτα το κάθε διακόπτη reed με μία συγκεκριμένη απόσταση μεταξύ τους, όπως και τη κάθε αντίσταση, και χρησιμοποιώντας ένα μαγνητικό φλοτέρ όπως αυτό που βλέπουμε στη παρακάτω φωτογραφία:



Εικόνα 22: Μαγνητικό φλοτέρ

προκειμένου να εφαρμόζει πάνω στο ηλεκτρολογικό σωλήνα για το κλείσιμο του κάθε διακόπτη reed, μπορούμε να λάβουμε διακριτές τιμές για τη τάση εξόδου του διαιρέτη τάσης, ανάλογα με το ποιος διακόπτης reed έχει κλείσει. Καθώς ο κάθε διακόπτης reed είναι τοποθετημένος σε συγκεκριμένο ύψος πάνω στο διαιρέτη τάσης του κυκλώματος που βρίσκεται μέσα στο σωλήνα, κάθε διακριτή τιμή της τάσης εξόδου του διαιρέτη τάσης θα αντιστοιχεί στο ύψος στο οποίο θα βρίσκεται το φλοτέρ πάνω στο σωλήνα.

#### 4.3 Η υλοποίηση της κατασκευής

Στη κατασκευή που υλοποιήσαμε επιλέξαμε η απόσταση μεταξύ των διακοπών reed να είναι 1 cm. Αυτό έγινε για να επιτύχουμε όσο το δυνατόν μεγαλύτερη ακρίβεια στη μέτρηση του ύψους (ακρίβεια ενός εκατοστού), διατηρώντας ταυτόχρονα ένα λογικό κόστος για τη κατασκευή, υποθέτοντας ότι το φλοτέρ θα κλείνει ένα διακόπτη reed τη φορά. Τις θεωρητικές μετρήσεις της τάσης εξόδου του διαιρέτη τάσης σε αντιστοιχία με το ύψος για 49 cm της κατασκευής τις βλέπουμε στο παρακάτω πίνακα:

Ύψος (cm)	Τάση (V)
1	0,00
2	0,10
3	0,20
4	0,31
5	0,41
6	0,51

7	0,61
8	0,71
9	0,82
10	0,92
11	1,02
12	1,12
13	1,22
14	1,33
15	1,43
16	1,53
17	1,63
18	1,73
19	1,84
20	1,94
21	2,04
22	2,14
23	2,24
24	2,35
25	2,45
26	2,55
27	2,65
28	2,76
29	2,86
30	2,96
31	3,06

32	3,16
33	3,27
34	3,37
35	3,47
36	3,57
37	3,67
38	3,78
39	3,88
40	3,98
41	4,08
42	4,18
43	4,29
44	4,39
45	4,49
46	4,59
47	4,69
48	4,80
49	4,90

**Πίνακας 1: Θεωρητικές μετρήσεις για 49 cm της κατασκευής**

Στη πραγματικότητα όμως, τα αποτελέσματα που λάβαμε από τις μετρήσεις στη κατασκευή ήταν διαφορετικά. Το φλοτέρ είχε διαστάσεις 40 mm x 51mm x 15mm και επίσης κάθε διακόπτης reed έκλεινε σε διαφορετική απόσταση από το φλοτέρ σε σύγκριση με τους άλλους, με την απόσταση αυτή να κυμαίνεται από 1 έως και 4 cm, με την πλειοψηφία όμως να είναι στο εύρος από 2 έως 3 cm. Αυτό είχε ως αποτέλεσμα να κλείνουν πολλοί διακόπτες μαζί, καθιστώντας τα αποτελέσματα που λαμβάναμε από τη μέτρηση της τάσης μη αναμενόμενα σε σύγκριση με τις θεωρητικές μετρήσεις. Τα αποτελέσματα για 49 cm της κατασκευής, χρησιμοποιώντας ως τάση εισόδου τα 5V και ένα πολύμετρο για τη μέτρηση της τάσης, φαίνονται παρακάτω:

Ύψος (cm)	Τάση (V)
1	0,11
3,3	0,12
6,9	0,36
8,4	0,39
8,9	0,62
9,9	0,72
10,4	0,74
10,7	0,76
11,4	0,87
11,8	0,89
12,9	0,94
13,3	1,14
13,6	1,23
14,9	1,27
15,4	1,39
15,8	1,48
16,2	1,52
17,3	1,6
17,7	1,65
19	1,69
20	1,82
20,4	1,85
21	2

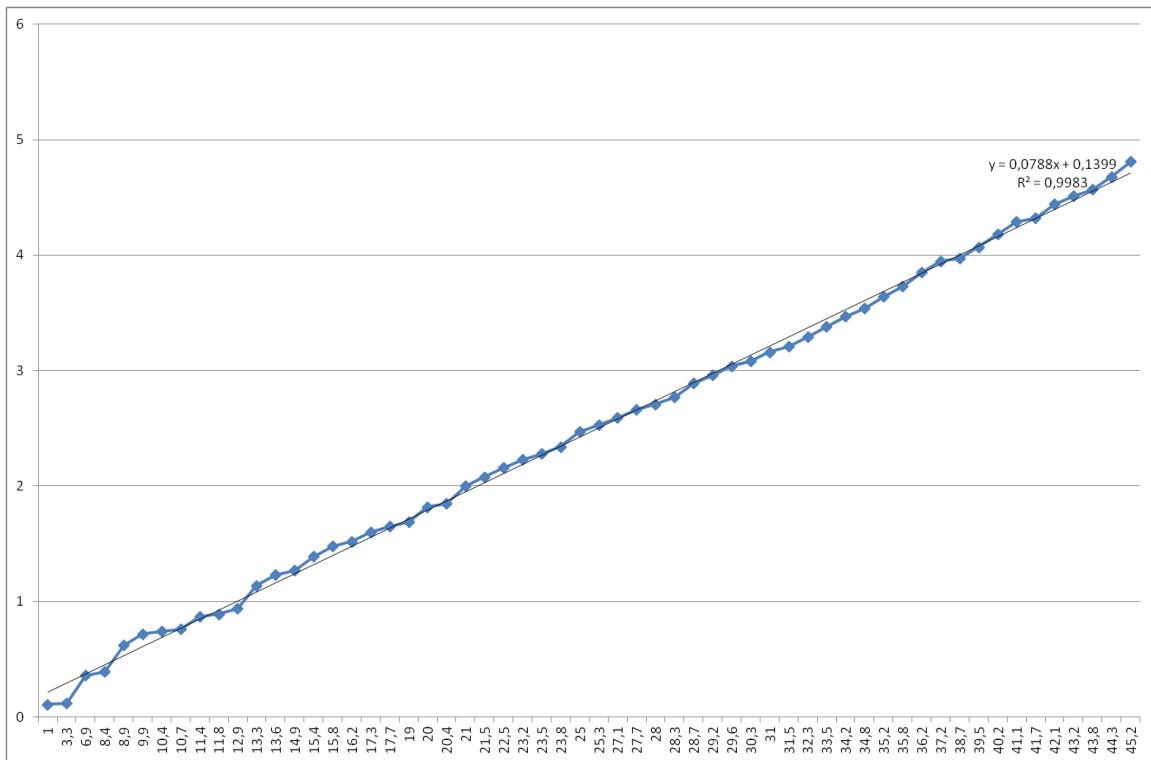
21,5	2,08
22,5	2,16
23,2	2,23
23,5	2,28
23,8	2,34
25	2,47
25,3	2,53
27,1	2,59
27,7	2,66
28	2,71
28,3	2,77
28,7	2,89
29,2	2,96
29,6	3,04
30,3	3,08
31	3,16
31,5	3,21
32,3	3,29
33,5	3,38
34,2	3,47
34,8	3,54
35,2	3,64
35,8	3,73
36,2	3,85
37,2	3,95



38,7	3,97
39,5	4,07
40,2	4,18
41,1	4,29
41,7	4,32
42,1	4,44
43,2	4,51
43,8	4,57
44,3	4,68
45,2	4,81

**Πίνακας 2: Πραγματικές μετρήσεις για 49 cm της κατασκευής**

Παρά τα μη αναμενόμενα αποτελέσματα όμως, βλέπουμε ότι η κατασκευή που υλοποιήσαμε δουλεύει, δηλαδή μπορούμε να πάρουμε τη μέτρηση του ύψους στο οποίο βρίσκεται το φλοτέρ σε αντιστοιχία με τη τιμή της τάσης. Παρατηρούμε επίσης από τις μετρήσεις που λάβαμε ότι η ακρίβεια του 1 cm επιτυγχάνεται όπως αρχικά επιθυμούσαμε, ενώ υπάρχουν και σημεία όπου η ακρίβεια είναι ακόμα μεγαλύτερη (εκτός από τα σημεία για ύψος μικρότερο των 10 cm). Όπως μπορούμε να δούμε στο παρακάτω διάγραμμα, οι μετρήσεις παρουσιάζουν σχεδόν τέλεια θετική γραμμική συσχέτιση:



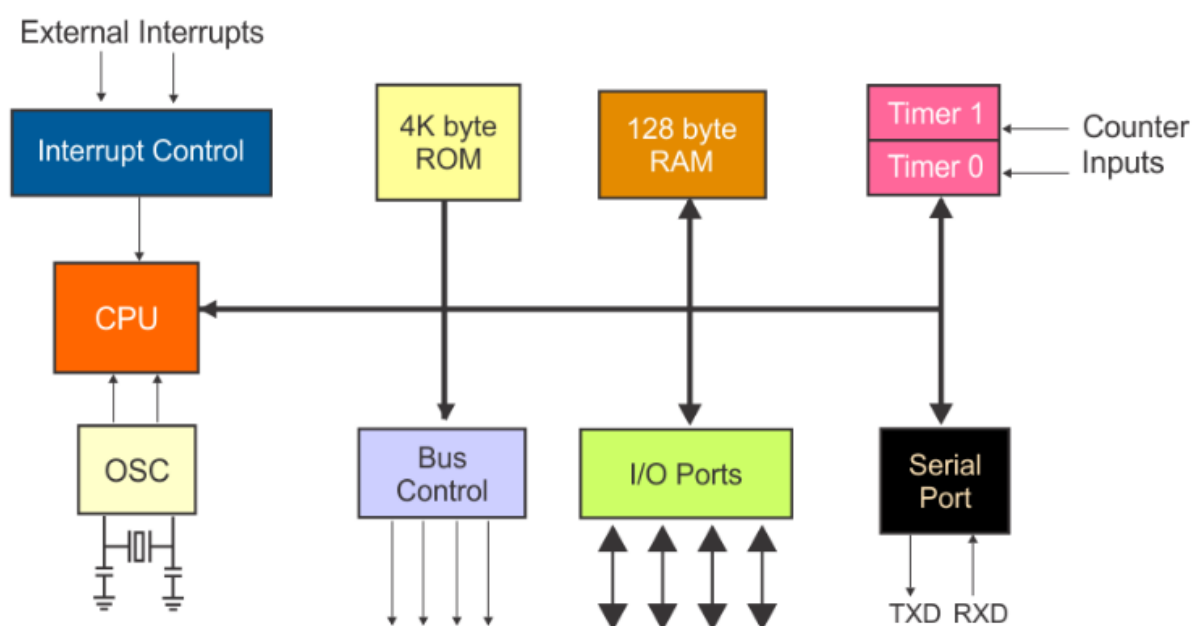
**Εικόνα 23: Διάγραμμα γραμμικής συσχέτισης των μετρήσεων**

Βλέπουμε ότι ο συντελεστής γραμμικής συσχέτισης των παραπάνω μετρήσεων είναι  $R^2 = 0,9983$ . Εξαιτίας της μη τέλει θετικής γραμμικής συσχέτισης, οι τιμές της τάσης που λαμβάνουμε δεν μπορούν να βρεθούν μέσω κάποιας εξίσωσης όπου η τάση είναι συνάρτηση του ύψους, καθώς έτσι κάποιες τιμές δεν θα αντιστοιχούσαν στο πραγματικό ύψος που θα βρίσκονταν το φλοτέρ. Αυτό το πρόβλημα θα λυθεί με τη μέθοδο της βαθμονόμησης, όπως θα δούμε σε επόμενο κεφάλαιο.

## 5 Ο μικροελεγκτής

### 5.1 Ο μικροελεγκτής 8051

Το έτος 1981, ένας μικροελεγκτής 8-bit δημιουργήθηκε από την Intel Corporation. Ήταν ο γνωστός ως 8051. Ο κλασικός 8051 μικροελεγκτής περιλαμβάνει 128 bytes RAM, 4K bytes ROM, 2 timers, 1 σειριακή θύρα και 4 θύρες που βρίσκονται σε ένα ενιαίο τσιπ και αναφέρονται ως SOC (system on chip). Ο μικροελεγκτής 8051 είναι ένας επεξεργαστής 8-bit, που σημαίνει ότι το CPU μπορεί να λειτουργήσει με 8 bits δεδομένα κάθε φορά. Ο μικροελεγκτής 8051 περιλαμβάνει τέσσερις θύρες I / O που έχουν πλάτος 8 bits η κάθε μία. Αν και πολλοί σχεδιαστές θέτουν μόνο 4K bytes ROM στο τσιπ., υπάρχει δυνατότητα έως 64K bytes.



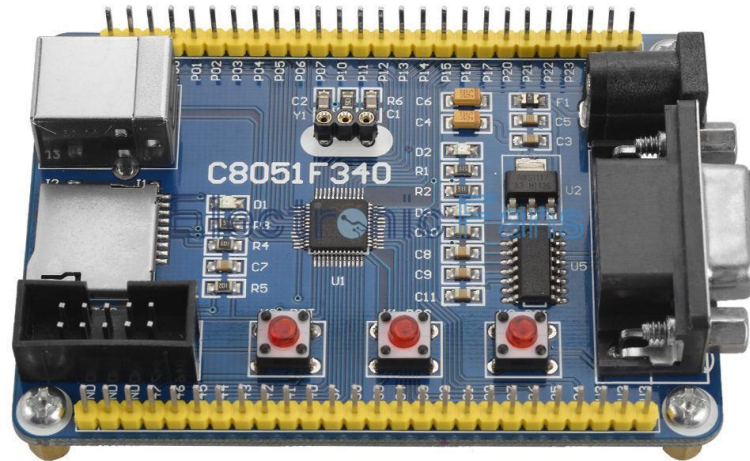
Εικόνα 24: Αρχιτεκτονική κλασικού 8051

Ο κύριος λόγος για τη μεγάλη του επιτυχία και δημοτικότητα είναι η επιδέξια επιλεγμένη διαμόρφωση του που ικανοποιεί τις διαφορετικές ανάγκες ενός μεγάλου αριθμού χρηστών, επιτρέποντας ταυτόχρονα τη δυνατότητα συνεχών επεκτάσεων. Επιπλέον, το λογισμικό έχει αναπτυχθεί σε τέτοια έκταση που απλά δεν ήταν κερδοφόρο να αλλάξει τίποτα στον βασικό πυρήνα του μικροελεγκτή. Αυτός είναι ο λόγος για την ύπαρξη ενός μεγάλου αριθμού διαφόρων μικροελεγκτών από διάφορες εταιρίες, οι οποίοι απλά είναι αναβαθμισμένες εκδόσεις της οικογένειας 8051.

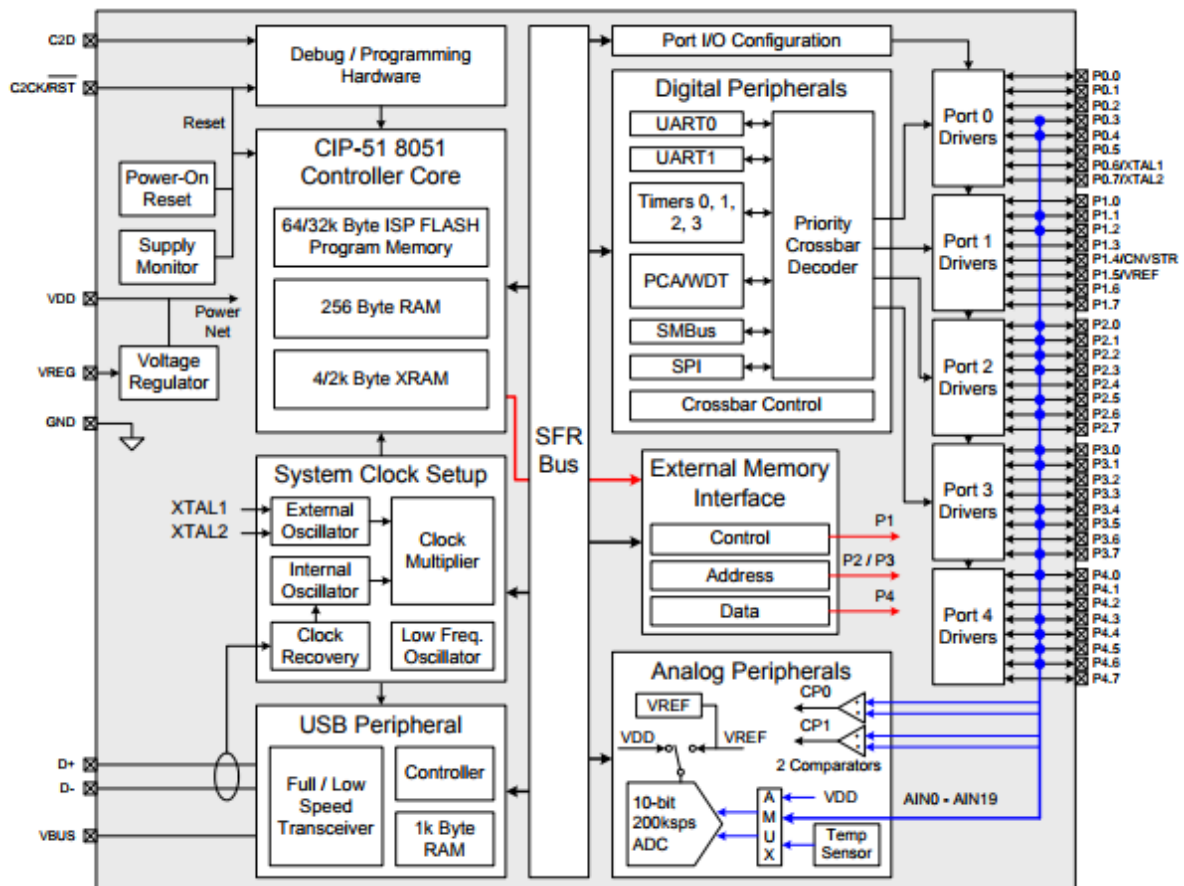
Στη περίπτωση μας επιλέχθηκε ο μικροελεγκτής C8051F340 της εταιρίας Silicon Labs. Είναι ένας μικροελεγκτής χαμηλού κόστους, ο οποίος έχει όλα τα απαραίτητα χαρακτηριστικά για τη χρήση για την οποία προορίζονταν:

- 64K bytes Flash memory
- 4 διαφορετικούς timers

- 2 θύρες UART (UART 0 και UART 1)
- 5 θύρες I/O (από 8 bit η κάθε μια)
- Ενσωματωμένο Analog to Digital Converter, ακρίβειας 10 bit



Εικόνα 25: Ο μικροελεγκτής C8051F340



Εικόνα 26: Το διάγραμμα του μικροελεγκτή C8051F340

### 5.1.1 Τα timers του 8051

Πολλές από τις εφαρμογές μικροελεγκτών απαιτούν μέτρηση εξωτερικών συμβάντων, όπως είναι η συχνότητα των παλμών και η δημιουργία εσωτερικών χρονικών καθυστερήσεων ακριβείας μεταξύ των ενεργειών του υπολογιστή. Και οι δύο αυτές εργασίες μπορούν να υλοποιηθούν χρησιμοποιώντας λύσεις λογισμικού, αλλά οι βρόχοι στο λογισμικό για την μέτρηση και το συγχρονισμό δεν μπορούν να δώσουν ακριβές αποτέλεσμα. Για να αποφευχθούν αυτά τα προβλήματα, τα timers και τα counters στους μικροελεγκτές είναι καλύτερες επιλογές για απλές και χαμηλού κόστους εφαρμογές. Αυτά τα timers και counters χρησιμοποιούνται για τη δημιουργία interrupts (διακοπών) στον μικροελεγκτή 8051.

Οι κλασικοί 8051 είναι εξοπλισμένοι με δύο timers, οι οποίοι μπορούν να ελέγχονται, να ρυθμίζονται, να διαβάζονται και να διαμορφώνονται ξεχωριστά. Τα timers του 8051 έχουν τρεις γενικές λειτουργίες:

- Παρακολούθηση του πραγματικού χρόνου και / ή υπολογισμό του χρονικού διαστήματος μεταξύ δύο συμβάντων
- Μέτρηση των ίδιων των συμβάντων
- Δημιουργία του ρυθμού μετάδοσης για τη σειριακή θύρα.

### 5.1.1.1 Πως μετράει ένα timer

Ένα timer μετράει πάντα προς τα πάνω. Δεν έχει σημασία αν το timer χρησιμοποιείται ως χρονομετρητής, μετρητής ή γεννήτρια ρυθμού μετάδοσης, το timer αυξάνεται πάντα από το μικροελεγκτή.

### 5.1.1.2 Πόσο χρειάζεται ένα timer για να μετρήσει

Πρώτον, αξίζει να σημειωθεί ότι όταν ένα timer βρίσκεται σε λειτουργία μέτρησης χρονικού διαστήματος (σε αντίθεση με τον τρόπο λειτουργίας του μετρητή συμβάντων) και έχει ρυθμιστεί σωστά, θα αυξάνεται κατά 1 σε κάθε κύκλο μηχανής (machine cycle). Εάν ένας κύκλος μηχανής αποτελείται από 12 παλμούς του κρυστάλλου, ένας τρέχον χρονομετρητής θα αυξηθεί:

$$12000000 / 12 = 1000000$$

1000000 φορές ανά δευτερόλεπτο (δηλαδή με 1 MHz συχνότητα), όπου 12000000 είναι η συχνότητα του εσωτερικού ταλαντωτή (12 MHz). Τα timers είναι συνεπείς, δηλαδή θα αυξάνονται πάντοτε μία φορά σε κάθε κύκλο μηχανής.

### 5.1.1.3 Καταχωρητές ειδικού σκοπού (Special Function Registers, SFR) των timer

Όπως αναφέρθηκε προηγουμένως, ο 8051 έχει δύο timers, τα οποία λειτουργούν ουσιαστικά με τον ίδιο τρόπο. Ένα timer είναι το TIMER 0 και το άλλο είναι το TIMER 1. Τα δύο timers μοιράζονται δύο SFR (TMOD και TCON) που ελέγχουν τα timers και κάθε timer διαθέτει επίσης δύο SFRs αποκλειστικά για τον εαυτό του, TH0 / TL0 και TH1 / TL1 αντίστοιχα.

Όνομα SFR	Περιγραφή
TH0	Timer 0 High Byte
TL0	Timer 0 Low Byte
TH1	Timer 1 High Byte
TL1	Timer 1 Low Byte
TCON	Timer Control
TMOD	Timer Mode

Πίνακας 3: Συνοπτικός πίνακας των SFR των timer

Το Timer 0 έχει δύο SFRs αποκλειστικά για τον εαυτό του: TH0 και TL0. Αυτοί οι καταχωρητές είναι ουσιαστικά το μεγάλο (high) και μικρό (low) byte αντίστοιχα του timer. Δηλαδή, όταν το Timer 0 έχει τιμή 0, τα TH0 και TL0 θα περιέχουν τη τιμή 0. Όταν το Timer 0 έχει την τιμή 1000, το TH0 θα κρατήσει το high byte της τιμής (3 σε δεκαδικό) και το TL0 θα περιέχει το low byte της τιμής (232 σε δεκαδικό). Για να υπολογίσουμε τη τελική τιμή, πολλαπλασιάζουμε το high byte κατά 256 και προσθέτουμε το low byte για να υπολογίσουμε τη τελική τιμή. Δηλαδή:

$$TH0 * 256 + TL0 = 1000$$

$$3 * 256 + 232 = 1000$$

Το Timer 1 λειτουργεί με τον ίδιο ακριβώς τρόπο, αλλά τα SFR του είναι τα TH1 και TL1.

Δεδομένου ότι υπάρχουν μόνο δύο byte που συνεισφέρουν στην τιμή του κάθε timer, είναι προφανές ότι η μέγιστη τιμή που μπορεί να έχει ένα timer είναι 65535 (0xFFFF σε HEX μορφή). Εάν ένα timer περιέχει τη τιμή 65535 και στη συνέχεια αυξάνεται, γίνεται επαναφορά, ή υπερχείλιση (overflow), πίσω στη τιμή 0.

#### 5.1.1.4 Το SFR TMOD

Το SFR TMOD χρησιμοποιείται για τον έλεγχο του τρόπου λειτουργίας και των δύο timer. Κάθε bit του SFR δίνει στο μικροελεγκτή συγκεκριμένες πληροφορίες σχετικά με τον τρόπο με τον οποίο θα χρησιμοποιηθεί ένα timer. Τα τέσσερα high bits (bits 4 έως 7) σχετίζονται με το Timer 1 ενώ τα τέσσερα low bits (bits 0 έως 3) εκτελούν τις ίδιες ακριβώς λειτουργίες, αλλά για το Timer 0. Τα μεμονωμένα bits του TMOD έχουν τις ακόλουθες λειτουργίες:

Bit	Όνομα	Εξήγηση της λειτουργίας	Timer
7	GATE1	Όταν το bit αυτό γίνει 1, το timer θα εκτελείται μόνο όταν η INT1 θύρα (external interrupt) είναι 1. Όταν το bit είναι 0, το timer θα τρέξει ανεξάρτητα από την κατάσταση του INT1.	1
6	C/T1	Όταν το bit αυτό γίνει 1, το timer θα μετρήσει συμβάντα στο Timer 1. Όταν το bit είναι 0, το timer αυξάνεται σε κάθε κύκλο μηχανής.	1
5	T1M1	Timer mode bit	1
4	T1M0	Timer mode bit	1

3	GATE0	Όταν το bit αυτό γίνει 1, το timer θα εκτελείται μόνο όταν η INT0 θύρα (external interrupt) είναι 1. Όταν το bit είναι 0, το timer θα τρέξει ανεξάρτητα από τη κατάσταση του INT0.	0
2	C/T0	Όταν το bit αυτό γίνει 1, το timer θα μετρήσει συμβάντα στο Timer 0. Όταν το bit είναι 0, το timer αυξάνεται σε κάθε κύκλο μηχανής.	0
1	T0M1	Timer mode bit	0
0	T0M0	Timer mode bit	0

Πίνακας 4: Συνοπτικός πίνακας των bit του TMOD SFR

Όπως βλέπουμε στο παραπάνω πίνακα, τέσσερα bits (δύο για κάθε timer) χρησιμοποιούνται για τον καθορισμό ενός τρόπου λειτουργίας. Οι τρόποι λειτουργίας είναι:

TxM1	TxM0	Timer Mode	Περιγραφή της λειτουργίας
0	0	0	13-bit Timer.
0	1	1	16-bit Timer
1	0	2	8-bit auto-reload
1	1	3	Split timer mode

Πίνακας 5: Συνδυασμός των bit για το καθορισμό της λειτουργίας των timer

### 5.1.1.5 13-bit Time Mode (mode 0)

Το timer mode "0" είναι ένα timer 13-bit. Είναι κάτι που ενσωματώθηκε στο 8051 για να διατηρήσει τη συμβατότητα με τον προκάτοχό του, το 8048. Γενικά, το 13-bit timer mode δεν χρησιμοποιείται σε νέες εφαρμογές.

Όταν το timer είναι σε λειτουργία 13 bit, το TLx θα μετρήσει από το 0 έως το 31. Όταν το TLx αυξάνεται από το 31, επαναφέρεται στο 0 και αυξάνεται το THx. Έτσι, ουσιαστικά, χρησιμοποιούνται μόνο 13 bits των δύο timer bytes : τα bits 0-4 του TLx και τα bits 0-7 του THx. Αυτό σημαίνει επίσης ότι το timer μπορεί να περιέχει μόνο 8192 τιμές. Εάν θέσουμε ένα timer 13-bit σε 0, θα υπερχειλίσει εκ νέου πίσω στο 0 ύστερα από 8192 κύκλους μηχανής.



### 5.1.1.6 16-bit Time Mode (mode 1)

Το timer mode "1" είναι ένα timer 16-bit. Αυτή είναι μια πολύ συχνά χρησιμοποιούμενη λειτουργία. Λειτουργεί ακριβώς όπως η λειτουργία με 13 bit εκτός από το ότι χρησιμοποιούνται και τα 16 bits.

Το TLx αυξάνεται από 0 σε 255. Όταν το TLx παίρνει τιμή μεγαλύτερη του 255, επαναφέρεται στο 0 και αυξάνεται η τιμή του THx κατά 1. Δεδομένου ότι πρόκειται για ένα πλήρες timer 16 bit, το timer μπορεί να περιέχει έως και 65536 διαφορετικές τιμές. Εάν θέσουμε ένα timer 16-bit σε 0, θα υπερχειλίσει πίσω στο 0 μετά από 65536 κύκλους μηχανής.

### 5.1.1.7 8-bit Time Mode (mode 2)

Το timer mode "2" είναι μια λειτουργία 8-bit αυτόματης επαναφόρτωσης (auto reload). Όταν ένα timer βρίσκεται στη λειτουργία 2, το THx διατηρεί τη reload τιμή και το TLx είναι το ίδιο το timer. Έτσι, το TLx αρχίζει να μετράει. Όταν το TLx φτάσει στο 255 και στη συνέχεια αυξηθεί, αντί για επαναφορά στο 0 (όπως γίνεται στα modes 0 και 1), θα επαναφέρεται στη τιμή που είναι αποθηκευμένη στο THx.

### 5.1.1.8 Split Timer Mode (mode 3)

Το timer mode "3" είναι μια λειτουργία split timer. Όταν το timer 0 τοποθετείται στη λειτουργία 3, ουσιαστικά γίνεται δύο ξεχωριστά timers 8-bit. Δηλαδή, το Timer 0 γίνεται το TL0 και το Timer 1 γίνεται το TH0. Και τα δύο timers υπολογίζονται από 0 έως 255 και η υπερχειλίση επιστρέφει στο 0. Όλα τα bits που σχετίζονται με το Timer 1 θα συνδεθούν τώρα με το TH0.

Ενώ το Timer 0 βρίσκεται σε λειτουργία split, το πραγματικό Timer 1 (δηλ. TH1 και TL1) μπορεί να τεθεί κανονικά στις λειτουργίες 0, 1 ή 2 - ωστόσο, δεν μπορούμε να ξεκινήσουμε ή να σταματήσουμε το πραγματικό Timer 1, που τώρα συνδέεται με το TH0. Το πραγματικό Timer 1, σε αυτή την περίπτωση, θα αυξάνεται πάντα σε κάθε κύκλο μηχανής.

### 5.1.1.9 Το SFR TCON

Τέλος, υπάρχει ακόμα ένα SFR που ελέγχει τα δύο timers και παρέχει πολύτιμες πληροφορίες για αυτά. Το SFR TCON έχει την ακόλουθη δομή:

Bit	Όνομα	Εξήγηση της λειτουργίας	Timer
7	TF1	<b>Timer 1 Overflow.</b> Αυτό το bit γίνεται 1 όταν το Timer 1 υπερχειλίζει.	1

6	TR1	<b>Timer 1 Run.</b> Όταν αυτό το bit γίνεται 1, το Timer 1 ξεκινά. Όταν το bit αυτό είναι 0, το Timer 1 δεν λειτουργεί.	1
5	TF0	<b>Timer 0 Overflow.</b> Αυτό το bit γίνεται 1 όταν το Timer 0 υπερχειλίζει.	0
4	TR0	<b>Timer 0 Run.</b> Όταν αυτό το bit γίνεται 1, ο Timer 0 ξεκινά. Όταν το bit αυτό είναι 0, το Timer 0 δεν λειτουργεί.	0

Πίνακας 6: Συνοπτική παρουσίαση των bit του TCON SFR

### 5.1.1.10 Τα Timers στον C8051F340

Κάθε μικροελεγκτής C8051F340 περιλαμβάνει τέσσερις counter/timers: δύο είναι 16-bit counter/timers που είναι συμβατά με εκείνα που βρίσκονται στο πρότυπο 8051 και δύο είναι 16 bit auto-reload timer για χρήση με συσκευές ADC (Analog to Digital Converter), SMBus (System Management Bus), USB (μετρήσεις πλαισίων), ταλαντωτή χαμηλής συχνότητας (μετρήσεις περιόδου), ή για γενική χρήση. Το Timer 0 και το Timer 1 είναι σχεδόν πανομοιότυπα και έχουν τέσσερις κύριους τρόπους λειτουργίας. Τα Timer 2 και Timer 3 προσφέρουν 16-bit και split 8-bit timer με auto-reload.

### 5.1.2 Η σειριακή επικοινωνία στον 8051

Ένα από τα πολλά σημαντικά χαρακτηριστικά του 8051 είναι το ενσωματωμένο Universal Asynchronous Receiver/Transmitter (UART), γνωστό και ως σειριακή θύρα (serial port). Το γεγονός ότι ο 8051 διαθέτει μια ενσωματωμένη σειριακή θύρα σημαίνει ότι μπορούμε πολύ εύκολα να διαβάζουμε και να γράφουμε τιμές στη σειριακή θύρα. Εμείς απλά πρέπει να διαμορφώσουμε το τρόπο λειτουργίας των σειριακών θυρών και τη ταχύτητα μετάδοσης. Μόλις ρυθμιστούν, το μόνο που πρέπει να κάνουμε είναι να γράψουμε στο κατάλληλο SFR για να στείλουμε μια τιμή στη σειριακή θύρα ή να διαβάσουμε το ίδιο SFR για να διαβάσουμε μια τιμή από τη σειριακή θύρα.

#### 5.1.2.1 Ρύθμιση της λειτουργίας της σειριακής θύρας

Για να μπορούμε να χρησιμοποιούμε τη σειριακή θύρα, πρέπει αρχικά να τη ρυθμίσουμε. Αυτό επιτρέπει στον μικροελεγκτή να ξέρει πόσα bits δεδομένων θέλουμε, το ρυθμό μετάδοσης που θα χρησιμοποιήσουμε και πως θα προσδιοριστεί αυτός ο ρυθμός μετάδοσης. Η ρύθμιση της σειριακής θύρας επιτυγχάνεται με το παρακάτω SFR, το οποίο ονομάζεται Serial Control (SCON):

Bit	Όνομα	Εξήγηση της λειτουργίας
7	SM0	Serial port mode bit 0

6	SM1	Serial port mode bit 1.
5	SM2	Multiprocessor Communications Enable
4	REN	Receiver Enable. Αυτό το bit πρέπει να είναι 1 για να μπορεί η σειριακή θύρα να λάβει χαρακτήρες.
3	TB8	Μετάδοση 8 bit. Είναι το ένατο bit για μετάδοση στις λειτουργίες 2 και 3.
2	RB8	Λήψη 8 bit. Είναι το ένατο bit που λαμβάνεται στις λειτουργίες 2 και 3.
1	TI	Transmit Flag (σημαία μετάδοσης). Το bit αυτό γίνεται 1 όταν ένα byte έχει σταλθεί με επιτυχία.
0	RI	Receive Flag (σημαία λήψης). Το bit αυτό γίνεται 1 όταν ένα byte έχει ληφθεί με επιτυχία.

**Πίνακας 7: Συνοπτικός πίνακας των bit του SCON SFR**

Τα bits SM0 και SM1 επιτρέπουν την επιλογή του serial mode, όπου υπάρχουν επιλογές για διάφορους ρυθμούς μετάδοσης. Στο μικροελεγκτή C8051F340 που χρησιμοποιήθηκε, το bit SM0 χρησιμοποιείται για την επιλογή μεταξύ του mode 0 με 8-bit UART με μεταβλητό ρυθμό μετάδοσης και mode 1 με 9-bit UART, επίσης με μεταβλητό ρυθμό μετάδοσης. Το bit 6 δεν χρησιμοποιείται.

Το bit SM2 (MCEX στον C8051F340) είναι μια σημαία για την επικοινωνία πολλαπλών επεξεργαστών (Multiprocessor communication). Γενικά, κάθε φορά που έχει ληφθεί ένα byte, ο 8051 θέτει τη σημαία RI (Receive Interrupt). Αυτό επιτρέπει στο πρόγραμμα να γνωρίζει ότι έχει ληφθεί ένα byte και ότι πρέπει να υποβληθεί σε επεξεργασία. Ωστόσο, όταν το SM2 έχει οριστεί, η σημαία RI θα ενεργοποιηθεί μόνο αν το 9ο bit που λήφθηκε ήταν 1. Δηλαδή, αν το SM2 έχει οριστεί 1 και έχει ληφθεί ένα byte του οποίου το 9ο bit είναι 0, η σημαία RI δεν θα ενεργοποιηθεί.

Το επόμενο bit, REN, σημαίνει ενεργοποίηση της λήψης (Receiver Enable). Αυτό το bit είναι πολύ απλό. Αν θέλουμε να λάβουμε δεδομένα μέσω της σειριακής θύρας, ορίζουμε αυτό το bit. Αυτό το bit είναι δηλαδή σχεδόν πάντα 1 (έχει οριστεί).

Το bit TB8 χρησιμοποιείται μόνο σε λειτουργία 9-bit UART στο C8051F340. Η τιμή αυτού του bit αναθέτεται στο 9ο bit της μετάδοσης.

Το bit RB8 λειτουργεί όπως ακριβώς το προηγούμενο bit, αλλά αντίστοιχα για την αποστολή.

Το TI σημαίνει διακοπή μετάδοσης (Transmit Interrupt). Όταν ένα πρόγραμμα γράψει μια τιμή στη σειριακή θύρα, θα περάσει ένας ορισμένος χρόνος προτού τα

μεμονωμένα bits του byte σταλούν από τη σειριακή θύρα. Εάν το πρόγραμμα έπρεπε να γράψει ένα άλλο byte στην σειριακή θύρα πριν από την πλήρη αποστολή του πρώτου byte, τα δεδομένα που αποστέλλονταν θα ήταν αλλοιωμένα. Έτσι, ο 8051 επιτρέπει στο πρόγραμμα να γνωρίζει ότι έχει ολοκληρώσει την αποστολή του τελευταίου byte ορίζοντας το bit TI. Όταν το bit TI είναι ενεργοποιημένο, το πρόγραμμα μπορεί να υποθέσει ότι η σειριακή θύρα είναι ελεύθερη και είναι έτοιμη να στείλει το επόμενο byte.

Τέλος, το bit RI σημαίνει διακοπή λήψης (Receive Interrupt). Λειτουργεί παρόμοια με το bit TI, αλλά δείχνει ότι έχει ληφθεί ένα byte. Δηλαδή, κάθε φορά που ο 8051 λαμβάνει ένα πλήρες byte, ενεργοποιεί το bit RI για να ενημερώσει το πρόγραμμα ότι πρέπει να διαβάσει την τιμή γρήγορα, πριν διαβάσει ένα άλλο byte.

### **5.1.2.2 Ρύθμιση του ρυθμού μετάδοσης των σειριακών θυρών**

Αφού έχει αποφασιστεί η λειτουργία της σειριακής θύρας, πρέπει μέσω του προγράμματος να ρυθμιστεί και η ταχύτητα μετάδοσης των σειριακών θυρών. Ο μικροελεγκτής που χρησιμοποιήθηκε προσφέρει μόνο τις λειτουργίες του κλασικού 8051 όπου ο ρυθμός μετάδοσης καθορίζεται από το πόσο συχνά υπερχειλίζει το Timer 1. Όσο πιο συχνά υπερχειλίζει το Timer 1, τόσο υψηλότερη είναι η ταχύτητα μετάδοσης. Υπάρχουν πολλοί τρόποι με τους οποίους μπορεί κανείς να προκαλέσει υπερχειλίση του Timer 1 με τέτοιο ρυθμό που καθορίζει ένα ρυθμό μετάδοσης, αλλά η πιο κοινή μέθοδος είναι να τοποθετηθεί το Timer 1 σε λειτουργία 8-bit auto-reload (timer mode 2) και να ρυθμιστεί μια τιμή reload (TH1) που προκαλεί υπερχειλίση του Timer 1 σε μια συχνότητα κατάλληλη για τη δημιουργία ρυθμού μετάδοσης.

### **5.1.2.3 Γράφοντας στη σειριακή θύρα**

Μόλις η σειριακή θύρα ρυθμιστεί σωστά, είναι έτοιμη να χρησιμοποιηθεί για την αποστολή και τη λήψη δεδομένων. Για να γραφτεί ένα byte στη σειριακή θύρα πρέπει απλά να γραφτεί η τιμή στο SFR SBUF. Προφανώς η μετάδοση δεν είναι στιγμιαία. Χρειάζεται κάποιος χρόνος για την ολοκλήρωση της μετάδοσης. Και δεδομένου ότι ο 8051 δεν έχει αποκλειστικό buffer για τη σειριακή θύρα εξόδου, πρέπει να είμαστε βέβαιοι ότι ένας χαρακτήρας έχει μεταδοθεί πλήρως προτού προσπαθήσουμε να μεταδώσουμε τον επόμενο. Ο 8051 μας επιτρέπει να γνωρίζουμε πότε έχει ολοκληρωθεί η μετάδοση ένας χαρακτήρα, ορίζοντας το bit TI στο SCON. Όταν ενεργοποιηθεί αυτό το bit, γνωρίζουμε ότι έχει μεταδοθεί και ο τελευταίος χαρακτήρας και ότι μπορούμε να στείλουμε τον επόμενο χαρακτήρα, αν υπάρχει.

### **5.1.2.4 Διαβάζοντας από τη σειριακή θύρα**

Η ανάγνωση δεδομένων που λαμβάνονται από τη σειριακή θύρα είναι εξίσου εύκολη. Για να διαβάσουμε ένα byte από τη σειριακή θύρα πρέπει να διαβάσουμε τη τιμή που είναι αποθηκευμένη στο SFR SBUF αφού ο 8051 έχει ενεργοποιήσει τη σημαία RI στο SCON.

### 5.1.3 Διακοπές (Interrupts) στο 8051

Μια διακοπή (interrupt) είναι ένα γεγονός το οποίο διακόπτει την κανονική εκτέλεση του προγράμματος. Τα interrupts μας δίνουν ένα μηχανισμό για να θέσουμε σε αναμονή την κανονική ροή του προγράμματος, να εκτελέσουμε μια υπορουτίνα και στη συνέχεια να ξαναρχίσουμε τη κανονική ροή του προγράμματος σαν να μην τη σταματήσαμε ποτέ. Αυτή η υπορουτίνα, που ονομάζεται interrupt handler, εκτελείται μόνο όταν συμβαίνει κάποιο συμβάν (interrupt). Το συμβάν μπορεί να είναι η υπερχειλίση ενός από τα timers, λήψη ενός χαρακτήρα μέσω της σειριακής θύρας, μετάδοση ενός χαρακτήρα μέσω της σειριακής θύρας κ.α. Ο 8051 μπορεί να ρυθμιστεί έτσι ώστε όταν συμβεί κάποιο από αυτά τα γεγονότα, το κύριο πρόγραμμα να αναστέλλεται προσωρινά και ο έλεγχος να μεταβιβάζεται σε ένα ειδικό τμήμα κώδικα που πιθανώς θα εκτελέσει κάποια λειτουργία που σχετίζεται με το συμβάν που συνέβη. Αφού ολοκληρωθεί, ο έλεγχος θα επιστραφεί στο αρχικό πρόγραμμα. Το κύριο πρόγραμμα δεν γνωρίζει καν ότι διακόπτεται.

#### 5.1.3.1 Ποια συμβάντα ενεργοποιούν τα interrupts

Μπορούμε να ρυθμίσουμε το 8051 έτσι ώστε οποιοδήποτε από τα ακόλουθα συμβάντα να προκαλέσει διακοπή:

- Timer 0 overflow
- Timer 1 overflow
- Λήψη/Μετάδοση σειριακού χαρακτήρα
- Εξωτερικό συμβάν 0 (External Event 0)
- Εξωτερικό συμβάν 1 (External Event 1)

Με άλλα λόγια, μπορούμε να ρυθμίσουμε το 8051, έτσι ώστε όταν το Timer 0 υπερχειλίσει ή όταν ένας χαρακτήρας στέλνεται/λαμβάνεται να καλούνται οι κατάλληλες interrupt handler ρουτίνες. Παρακάτω βλέπουμε τα διάφορα interrupts και τις σημαίες τους:

<b>Interrupt</b>	<b>Flag</b>
External 0	IE0
Timer 0	TF0
External 1	IE1
Timer 1	TF1
Serial	RI/TI

Πίνακας 8: Τα interrupts και οι σημαίες τους

### 5.1.3.2 Ρυθμίζοντας τα interrupts

Από προεπιλογή, κατά την ενεργοποίηση, όλα τα interrupts είναι απενεργοποιημένα. Αυτό σημαίνει ότι ακόμα και αν έχει ενεργοποιηθεί το bit TF0, ο 8051 δεν θα εκτελέσει το interrupt. Το πρόγραμμά πρέπει να πει συγκεκριμένα στο 8051 ότι επιθυμεί να ενεργοποιήσει τα interrupts και τα συγκεκριμένα interrupts που επιθυμεί να ενεργοποιήσει. Το πρόγραμμά μπορεί να ενεργοποιήσει και να απενεργοποιήσει τα interrupts, τροποποιώντας το SFR IE:

Bit	Όνομα	Εξήγηση της λειτουργίας
7	EA	Global Interrupt Enable/Disable
6	-	Undefined
5	-	Undefined
4	ES	Enable Serial Interrupt
3	ET1	Enable Timer 1 Interrupt
2	EX1	Enable External 1 Interrupt
1	ET0	Enable Timer 0 Interrupt
0	EX0	Enable External 0 Interrupt

Πίνακας 9: Συνοπτικός πίνακας των bit του IE SFR

Όπως βλέπουμε, κάθε ένα από τα interrupts του 8051 έχει το δικό του bit στο SFR IE. Ένα συγκεκριμένο interrupt ενεργοποιείται ρυθμίζοντας το αντίστοιχο bit.

Στο C8051F340 το bit 5 ονομάζεται ET2 και χρησιμοποιείται για την ενεργοποίηση του interrupt από το Timer 2 (Enable Timer 2 Interrupt).

### 5.1.3.3 Σειριακά interrupts

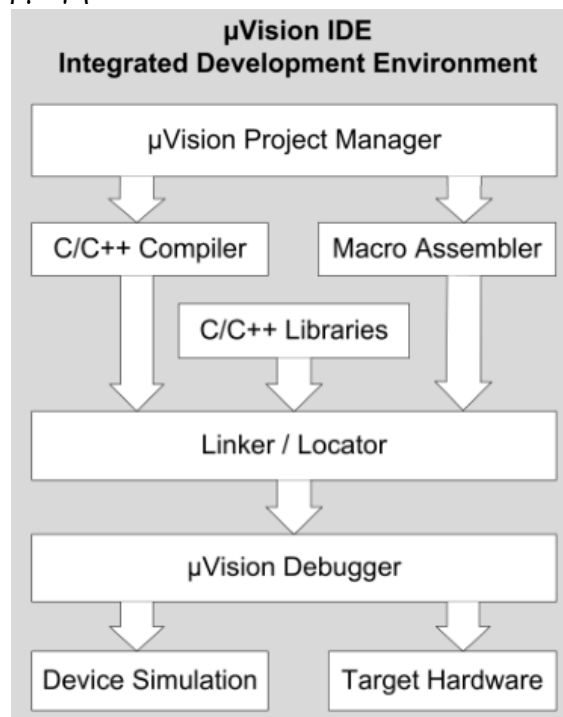
Τα σειριακά interrupts είναι ελαφρώς διαφορετικά από τα υπόλοιπα. Αυτό οφείλεται στο γεγονός ότι υπάρχουν δύο interrupt flags: RI και TI. Εάν έχει οριστεί κάποια σημαία, ενεργοποιείται ένα σειριακό interrupt. Όπως είδαμε προηγουμένως, το bit RI ορίζεται όταν ένα byte λαμβάνεται από τη σειριακή θύρα και το bit TI ορίζεται όταν έχει αποσταλεί ένα byte. Αυτό σημαίνει ότι όταν εκτελείται το σειριακό interrupt, μπορεί να έχει ενεργοποιηθεί επειδή έχει ενεργοποιηθεί η σημαία RI ή επειδή έχει ενεργοποιηθεί η σημαία TI ή επειδή έχουν ενεργοποιηθεί και οι δύο σημαίες. Έτσι, το πρόγραμμα πρέπει να ελέγξει την κατάσταση αυτών των σημαίων για να καθορίσει ποια ενέργεια είναι η κατάλληλη. Επίσης,

δεδομένου ότι ο 8051 δεν καθαρίζει αυτόματα τις σημαίες RI και TI,σε αντίθεση με τις υπόλοιπες σημαίες, τα bits αυτά πρέπει να καθαριστούν από τον χρήστη.

## 5.2 Το εργαλείο **μVision** του Keil

Για την ανάπτυξη της εφαρμογής στο μικροελεγκτή χρησιμοποιήθηκε το εργαλείο **μVision** του Keil. Όταν χρησιμοποιούνται τα εργαλεία του Keil, ο κύκλος ανάπτυξης του project είναι παρόμοιος με οποιοδήποτε άλλο λογισμικό ανάπτυξης project:

1. Δημιουργούμε ένα project, επιλέγουμε τη συσκευή για την οποία προορίζεται από τη βάση δεδομένων συσκευών και διαμορφώνουμε τις ρυθμίσεις του εργαλείου.
2. Δημιουργούμε τα αρχεία προέλευσης σε C / C ++ ή Assembly.
3. Δημιουργούμε την εφαρμογή μας με το Project Manager.
4. Πραγματοποιούμε Debug και διόρθωση των σφαλμάτων στα αρχεία προέλευσης, επιβεβαιώνουμε και βελτιστοποιούμε την εφαρμογή μας.
5. Φορτώνουμε τον κώδικα μας σε Flash ROM ή SRAM και δοκιμάζουμε τη συνδεδεμένη εφαρμογή.



Εικόνα 27: Δημιουργία project στο **μVision IDE**

Κάποια από τα blocks του διαγράμματος περιγράφονται παρακάτω.

### 5.2.1 **μVision IDE**

Το **μVision IDE** είναι μια πλατφόρμα ανάπτυξης λογισμικού που βασίζεται σε παράθυρα και συνδυάζει ισχυρό editor, Project Manager και το εργαλείο Make Utility. Το **μVision** υποστηρίζει όλα τα εργαλεία Keil όπως το C / C ++ Compiler, το Macro Assembler,

το Linker, τη Διαχείριση Βιβλιοθήκης και τον Μετατροπέα Object-HEX. Το μVision συμβάλλει στην επιτάχυνση της διαδικασίας ανάπτυξης παρέχοντας:

- Βάση συσκευών για την επιλογή μιας συσκευής και τη διαμόρφωση των εργαλείων ανάπτυξης για τη συγκεκριμένη συσκευή
- Project Manager για τη δημιουργία και τη διατήρηση των project
- Make Utility για τη συναρμολόγηση, τη σύνταξη και τη σύνδεση των εφαρμογών
- Πλήρης επεξεργαστής πηγαίου κώδικα
- Source Browser για γρήγορη εξερεύνηση αντικειμένων κώδικα, εντοπισμό και ανάλυση δεδομένων στην εφαρμογή
- Λειτουργία Browser για γρήγορη πλοήγηση μεταξύ των λειτουργιών του προγράμματος
- Το προσομοιωτή και το Debugger μVision που είναι πλήρως ενσωματωμένα
- Configuration Wizard που παρέχει γραφική επεξεργασία για τον κωδικό εκκίνησης του μικροελεγκτή
- Βοηθητικά προγράμματα προγραμματισμού Flash, όπως τα Keil ULINK USB-JTAG adapters

#### **5.2.1.1 μVision Device Database**

Η βάση δεδομένων συσκευών μVision προσφέρει έναν βολικό τρόπο επιλογής και διαμόρφωσης των παραμέτρων της συσκευής και του project. Περιλαμβάνει προκαθορισμένες ρυθμίσεις, ώστε να είναι δυνατό να επικεντρωθούμε πλήρως στις απαιτήσεις της εφαρμογής. Τα χαρακτηριστικά της βάσης δεδομένων των συσκευών μπορούν να χρησιμοποιηθούν για να:

- Αρχικοποιήσουμε το κώδικα εκκίνησης και τις ρυθμίσεις της συσκευής
- Φορτώσουμε τις επιλογές διαμόρφωσης για τον συναρμολογητή, τον μεταγλωττιστή και τον σύνδεσμο
- Προσθέτουμε και να αλλάζουμε τις ρυθμίσεις διαμόρφωσης του μικροελεγκτή

#### **5.2.1.2 μVision Debugger**

Ο μVision Debugger είναι πλήρως ενσωματωμένος στο μVision IDE. Παρέχει τα ακόλουθα χαρακτηριστικά:

- Πολλαπλές επιλογές από break points
- Δυνατότητα ελέγχου και τροποποίησης των τιμών της μνήμης, των μεταβλητών και των καταχωρητών
- Έλεγχος της κατάστασης των περιφερειακών του μικροελεγκτών
- Execution Profiling για την καταγραφή και εμφάνιση του χρόνου που καταναλώνεται, καθώς και των κύκλων που απαιτούνται για κάθε εντολή



- Διάφορα εργαλεία ανάλυσης για τη προβολή στατιστικών στοιχείων, τη καταγραφή τιμών των μεταβλητών και των σημάτων εισόδου / εξόδου των περιφερειακών και την εμφάνισή τους σε χρονικό άξονα

Ο μVision Debugger προσφέρει δύο τρόπους λειτουργίας - Simulator Mode και Target Mode.

Το Simulator Mode ρυθμίζει το Debugger ως ένα λογισμικό που προσομοιώνει με ακρίβεια το μικροελεγκτή που έχουμε διαλέξει, συμπεριλαμβανομένων των εντολών και των περισσοτέρων περιφερειακών συσκευών. Σε αυτή τη λειτουργία, μπορούμε να δοκιμάσουμε το κώδικα της εφαρμογής χωρίς να χρειάζεται κάποιο hardware. Μας δίνει σημαντικά οφέλη για την ταχεία ανάπτυξη αξιόπιστου λογισμικού για ενσωματωμένα. Το Simulator Mode προσφέρει:

- Δοκιμές του λογισμικού στην επιφάνεια εργασίας του υπολογιστή χωρίς την ανάγκη για κάποιο hardware
- Breakpoints που είναι αδύνατο με τα debuggers για hardware
- Ανίχνευση σεναρίων αποτυχίας που θα κατάστρεφαν τα πραγματικά περιφερειακά του hardware

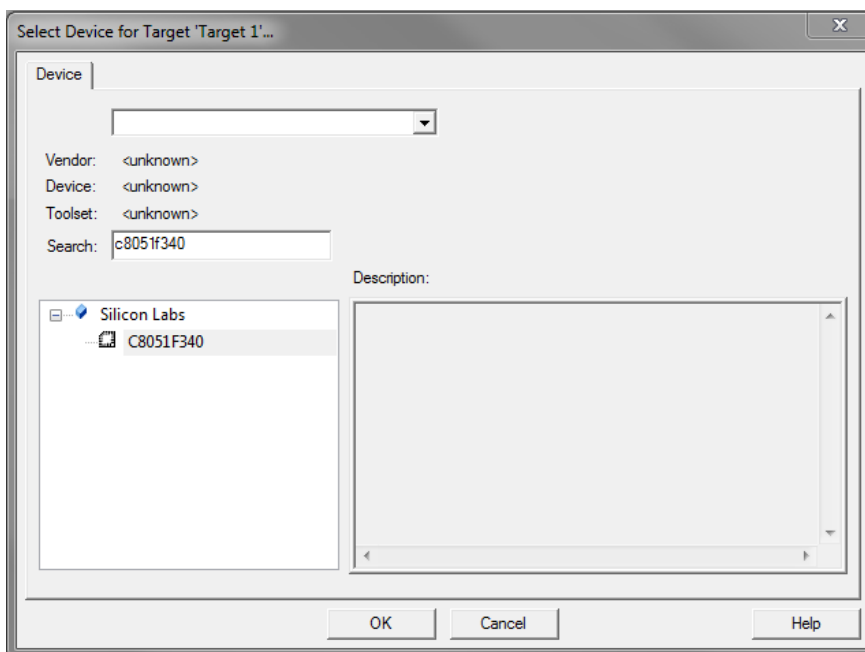
Το Target Mode συνδέει το Debugger με το πραγματικό hardware. Στη περίπτωση μας χρησιμοποιήθηκε ο USB debug adapter για μικροελεγκτές της σειράς C8051F της Silicon Labs που βλέπουμε στη παρακάτω φωτογραφία:



Εικόνα 28: Ο USB debug adapter

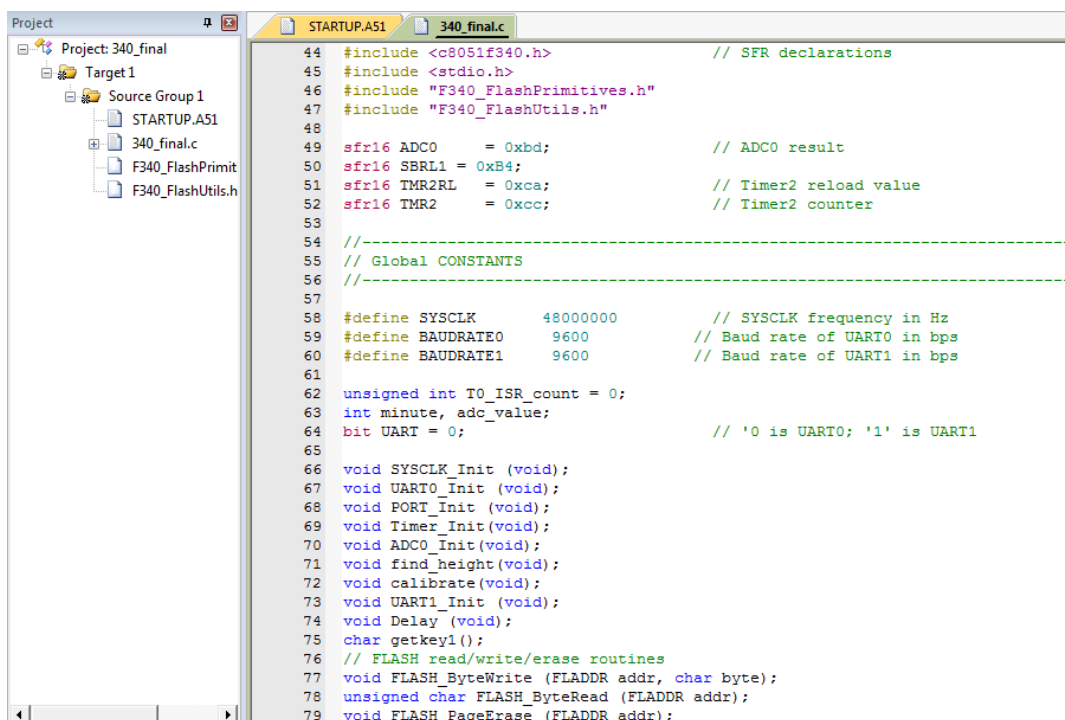
### 5.2.2 Δημιουργία ενός ολοκληρωμένου project στο μVision

Αρχικά, διαλέγοντας την επιλογή New μVision project επιλέγουμε το όνομα του project και το φάκελο στο οποίο θα αποθηκευτεί αυτό το project. Στη συνέχεια επιλέγουμε τη συσκευή η οποία είναι το target hardware στη περίπτωση μας:



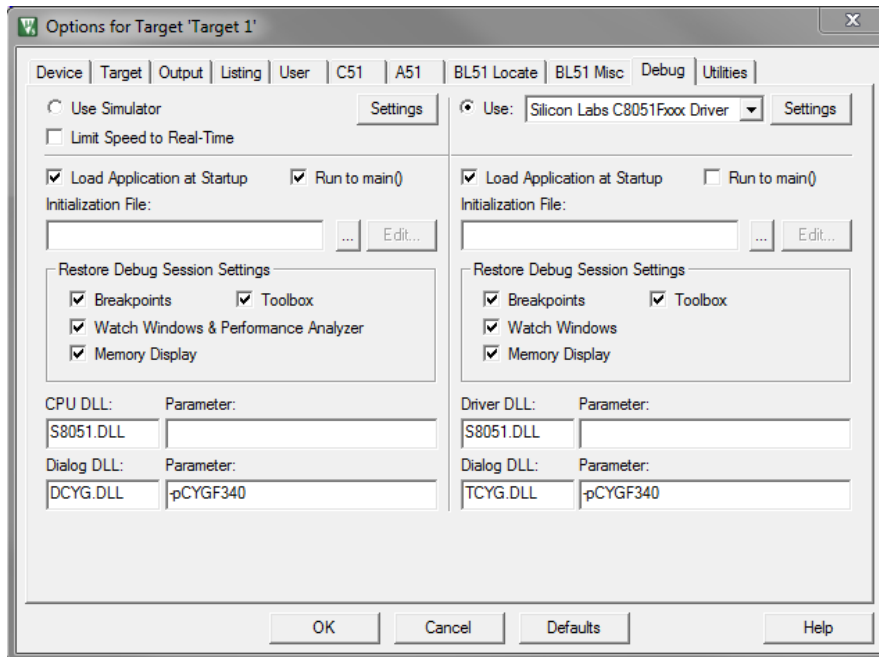
Εικόνα 29: Επιλογή συσκευής

Αφού έχουμε ολοκληρώσει τη συγγραφή του προγράμματός μας, έχοντας συμπεριλάβει και τα διάφορα headers που θα χρειαστούμε, το project έχει πάρει τη παρακάτω μορφή:



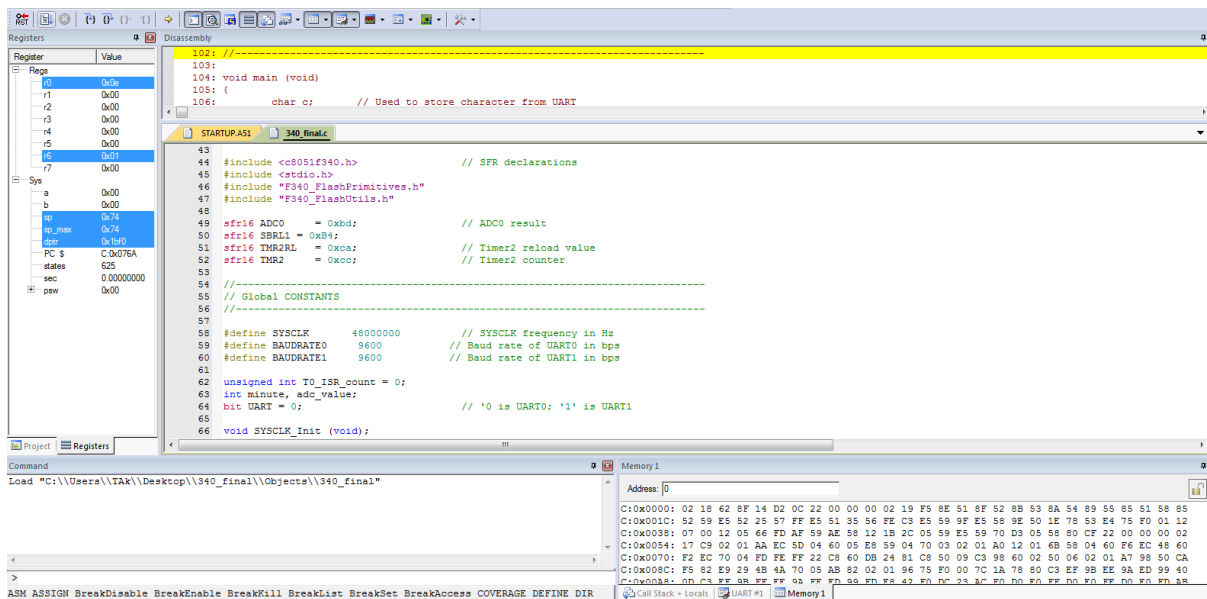
Εικόνα 30: Δομή τελικού του τελικού project

Για να δοκιμάσουμε το πρόγραμμα που δημιουργήσαμε στο πραγματικό hardware, διαλέγουμε την επιλογή Configure Flash Tools και στη καρτέλα debug επιλέγουμε από τη λίστα το driver για τις συσκευές Silicon Labs:



Εικόνα 31: Επιλογή του adapter για το debugging

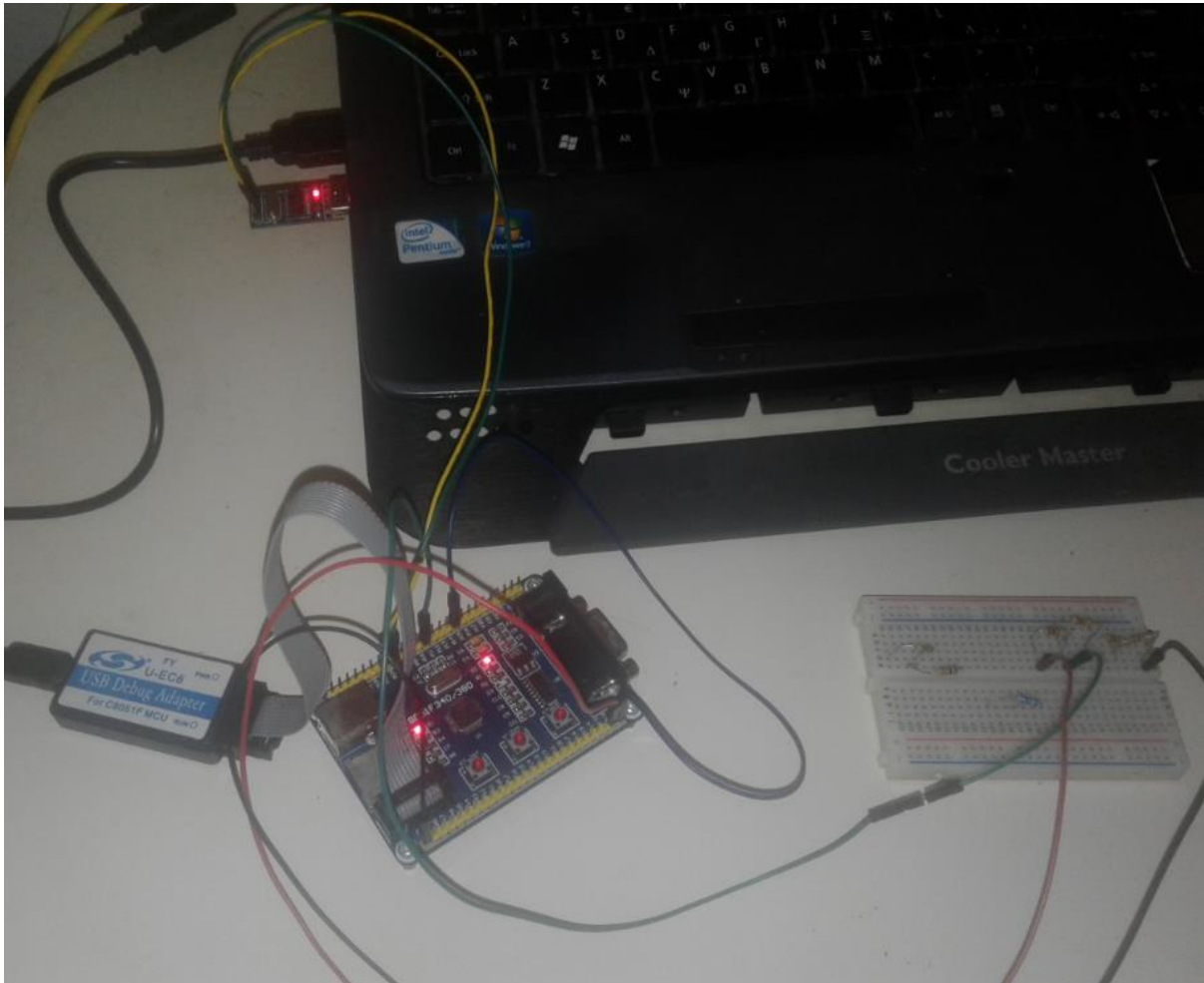
Αφού ολοκληρωθεί η διαδικασία εγγραφής του προγράμματος στο Flash ROM της συσκευής, μπορεί να ξεκινήσει το debugging:



Εικόνα 32: Screenshot κατά την εκκίνηση του debugging

Όπως βλέπουμε, υπάρχει μεγάλη πληθώρα παραθύρων που προσφέρουν πληροφορίες σχετικά με τους καταχωρητές, τις θέσεις μνήμης (Flash ROM) του μικροελεγκτή όπου έχει γίνει εγγραφή κώδικα, παράθυρο για το UART κ.α. Εάν κατά τη διάρκεια του debugging δεν βρεθεί κάτι το μεμπτό, έχει η ολοκληρωθεί η δημιουργία του project με το εργαλείο μVision και ο μικροελεγκτής μας μπορεί να χρησιμοποιηθεί στη κατασκευή που επιθυμούμε. Παρακάτω βλέπουμε τη διάταξη κατά τη διάρκεια του debugging, όπου φαίνεται ο

υπολογιστής, το USB debug adapter, ο μικροελεγκτής C8051F340 και το breadboard όπου έγινε η προσομοίωση του διαιρέτη τάσης της κατασκευής:



Εικόνα 33: Η διάταξη κατά τη διάρκεια του debugging

### 5.3 Η εφαρμογή στο μικροελεγκτή

Ο μικροελεγκτής στη κατασκευή που υλοποιήσαμε εξυπηρετεί συγκεκριμένους σκοπούς:

- Αντιστοίχιση της τιμής της τάσης του αισθητήρα με το ύψος στο οποίο βρίσκεται το φλοτέρ πάνω στη ράβδο, χρησιμοποιώντας το analog to digital converter που είναι ενσωματωμένο στο μικροελεγκτή
- Λειτουργία ως ρολογιού για τη παρακολούθηση του πραγματικού χρόνου
- Αποθήκευση της μέτρησης της κατανάλωσης στις θέσεις μνήμης του μικροελεγκτή όταν εμείς επιθυμούμε
- Επικοινωνία μέσω του bluetooth module με την android εφαρμογή για την αποστολή των αποθηκευμένων μετρήσεων και του τρέχοντος ύψους, ανάλογα με το τι έχει επιλέξει ο χρήστης

### 5.3.1 Analog to Digital Conversion

Για το διάβασμα της τιμής της τάσης του αισθητήρα στην αντίστοιχη θύρα του μικροελεγκτή χρησιμοποιήθηκε ο ενσωματωμένος analog to digital converter του μικροελεγκτή. Ο κώδικας που χρειάστηκε για αυτή τη λειτουργία παίρνει και υπολογίζει κατά μέσο όρο 2048 αναλογικές μετρήσεις από την είσοδο P1.1 χρησιμοποιώντας το ADC0. Για το χρονισμό χρησιμοποιήθηκε ο εσωτερικός ταλαντωτής στη μέγιστη συχνότητα 48 MHz. Ο Timer 2, έχοντας ρυθμιστεί σε 16-bit auto-reload mode, ενεργοποιεί μια μετατροπή στο ADC0 σε κάθε υπερχειλίση (ανά 100μs). Η ολοκλήρωση αυτής της μετατροπής με τη σειρά της ενεργοποιεί μια ρουτίνα υπηρεσίας διακοπής (Interrupt Service Routine, ISR). Το ISR υπολογίζει κατά μέσο όρο 2048 μετρήσεις και έτσι είναι δυνατό το πέρασμα της τιμής στη main συνάρτηση πριν από την έναρξη ενός άλλου μέσου κύκλου. Όταν ολοκληρωθούν οι 2048 μετρήσεις, η τάση υπολογίζεται με το παρακάτω κώδικα:

```
result = accumulator / 2048;  
mV = result * 3300 / 1023;
```

όπου η μεταβλητή accumulator περιέχει το άθροισμα των 2048 μετρήσεων της τιμής του ADC, result είναι ο μέσος όρος των μετρήσεων της τιμής του ADC που λάβαμε (περιλαμβάνοντας τιμές από 0 έως 1023) και mV η μέτρηση της τάσης. Η τάση υπολογίζεται πολλαπλασιάζοντας τη τιμή του ADC με τη τάση αναφοράς  $V_{ref}$  (3,3V στη περίπτωση μας) και διαιρώντας με το 1023 (το ADC του μικροελεγκτή C8051F340 έχει ακρίβεια 10 bit).

Ο αναλογικός πολυπλέκτης επιλέγει το P1.1 ως θετική είσοδο ADC0. Αυτή η θύρα είναι διαμορφωμένη ως αναλογική είσοδος στη συνάρτηση αρχικοποίησης των θυρών. Η αρνητική είσοδος του ADC0 συνδέεται μέσω πολυπλέκτη στη γείωση, παρέχοντας έτσι ADC μονής εισόδου.

### 5.3.2 Βαθμονόμηση

Βαθμονόμηση (calibration) ονομάζεται η διαδικασία κατά την οποία ο κατασκευαστής ενός οργάνου πραγματοποιεί μετρήσεις και ρυθμίσεις προκειμένου να εξασφαλιστεί η ομαλότητα της τιμής του μετρούμενου μεγέθους του οργάνου.

Όπως είδαμε σε προηγούμενο κεφάλαιο, η τιμή του ύψους σε συνάρτηση με τη τιμή της τάσης δεν παρουσιάζουν τέλεια θετική γραμμική συσχέτιση και δεν μπορεί να χρησιμοποιηθεί κάποια εξίσωση για την αντιστοίχισή τους. Το πρόβλημα αυτό λύθηκε μέσω της βαθμονόμησης των μετρήσεων της τάσης και του αντίστοιχου ύψους.

Αρχικά έγινε προσπάθεια να γίνει η βαθμονόμηση πραγματοποιώντας ελεύθερη πτώση του φλοτέρ από το μεγαλύτερο ύψος του σωλήνα μέχρι τη βάση του και με βάση το τύπο της ελεύθερης πτώσης:

$$h = \frac{1}{2}gt^2$$

μετρώντας το χρονικό διάστημα μεταξύ κάθε αλλαγής της τάσης του αισθητήρα να υπολογιστεί του ύψους που διάνυσε το φλοτέρ. Το πρόγραμμα που δημιουργήσαμε συγκρίνει συνεχώς τις τιμές της τάσης του αισθητήρα μέσω του ADC, υπολόγιζε το χρονικό διάστημα στο οποίο άλλαξε η τιμή της τάσης αυτής χρησιμοποιώντας το Timer 0 σε 16-bit counter/timer mode και χάρη στο τύπο της ελεύθερης πτώσης υπολόγιζε το διάστημα που διανύθηκε. Το κυρίως μέρος του κώδικα το βλέπουμε παρακάτω:

```
do {
    if (volt_temp != volts) {
        volt_temp = volts;
        printf("Voltage: %d mV, ", volts);
        x = (unsigned long)((TH0 << 8) | TL0 | ((unsigned long)T0_ISR_count <<
16));
        t= (((float)x/4000000));
        h = 490 * t * t;
        printf ("Time: %.2f s, ", t);
        printf("Height: %.0f cm.\n", h);
    }
}
while (volts >= 10);
printf("Done!\n");
```

όπου η μεταβλητή volts είναι η τιμή της τάσης που λαμβάνουμε από το ADC, t ο χρόνος σε δευτερόλεπτα και h το διάστημα που διανύθηκε σε cm. Το πρόγραμμα σταματάει να εκτελείται όταν η τάση φτάσει σε πολύ μικρή τιμή (μικρότερη των 10mV), δηλαδή όταν φτάσουμε στη βάση του ηλεκτρολογικού σωλήνα. Τα αποτελέσματα που λάβαμε από το πρόγραμμα αυτό και τη πραγματοποίηση της ελεύθερης πτώσης είναι τα παρακάτω (εικόνα 34):

```
cVoltage: 3296 mV, Time: 0.02 s, Height: 0 cm.
Voltage: 2641 mV, Time: 0.46 s, Height: 104 cm.
Voltage: 193 mV, Time: 0.87 s, Height: 371 cm.
Voltage: 3 mV, Time: 1.28 s, Height: 800 cm.
Done!
```

Εικόνα 34: Αποτελέσματα ελεύθερης πτώσης

Όπως βλέπουμε, οι εντολές δεν ήταν δυνατό να πραγματοποιηθούν αρκετά γρήγορα (παρατηρήθηκαν μόνο 4 αλλαγές τάσης στη συγκεκριμένη περίπτωση), ενώ εξαιτίας της μικρής τριβής που υπήρχε μεταξύ του ηλεκτρολογικού σωλήνα και του φλοτέρ επηρεάστηκε ο χρόνος της ελεύθερης πτώσης, με αποτέλεσμα να λάβουμε λανθασμένα αποτελέσματα για το ύψος στο οποίο πραγματοποιούνταν η κάθε αλλαγή τάσης.

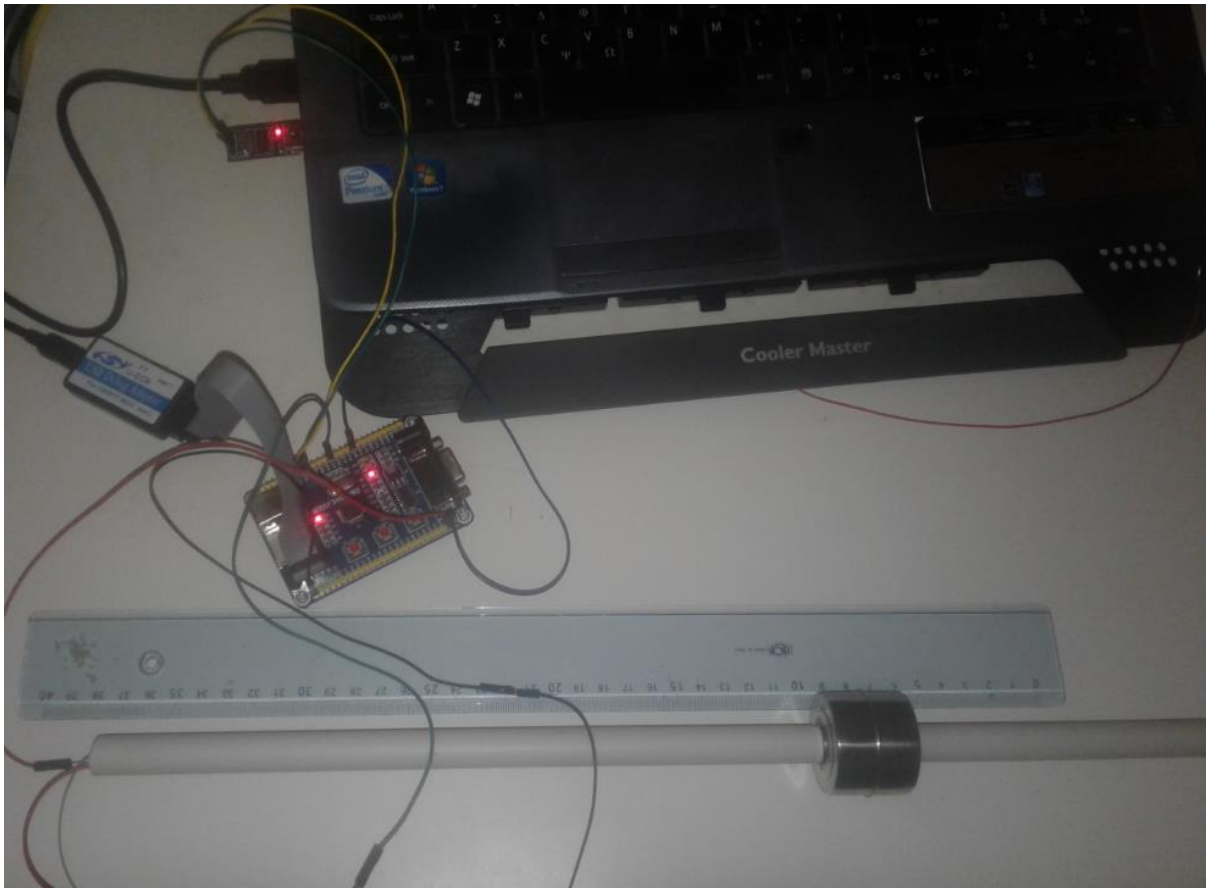
Τελικά, η βαθμονόμηση υλοποιήθηκε δημιουργώντας ένα πρόγραμμα που ελέγχει την αλλαγή της τάσης του αισθητήρα μέσω του ADC και εκτυπώνει τα αποτελέσματα κάθε φορά που συμβαίνει αυτή η αλλαγή:

```

do {
    if (volt_temp != volts) {
        volt_temp = volts;
        printf("Voltage: %d mV, ", volts);
        printf("ADC: %d mV, ", adc);
    }
}
while (volts >= 10);
printf("Done!\n");

```

όπου volts είναι η τάση του αισθητήρα και adc η μέτρηση που λαμβάνουμε από το ADC. Η μετακίνηση του φλοτέρ κατά μήκος της ράβδου έγινε χειροκίνητα, παρακολουθώντας ταυτόχρονα με ένα χάρακα σε ποιο μήκος αντιστοιχεί η κάθε τάση (εικόνα 35):



Εικόνα 35: Φωτογραφία της ράβδου κατά τη διάρκεια της βαθμονόμησης

Όλες οι τιμές της τάσης και το αντίστοιχο ύψος που βρήκαμε για 134 εκατοστά της κατασκευής που υλοποιήσαμε φαίνονται στο παρακάτω πίνακα:

mV	Ύψος
3294	134

3268	133,5
3242	128,7
3216	127,2
3187	126,6
3165	126,1
3161	125,1
3158	124
3135	123,8
3106	122,2
3081	121,8
3055	120,9
3052	119,4
3003	119,2
3000	118,9
2997	118,4
2948	118,1
2945	116,9
2923	116,4
2919	115,6
2916	115,3
2871	115
2868	113,8
2861	113,1
2858	112,3
2790	111,5



2787	110,3
2765	109,8
2755	108,9
2732	108,5
2729	107,8
2706	107,5
2684	107,3
2681	106,4
2674	105,7
2652	105,3
2632	104,9
2626	104,5
2619	103,9
2613	103
2594	102,8
2571	102
2526	101,3
2519	100,5
2490	100
2484	99,7
2477	99,1
2439	98,6
2432	97,8
2410	96,3
2394	95,8

2384	95,6
2368	94,9
2348	94,3
2342	93,2
2323	92,5
2303	92,3
2294	91,3
2268	90,8
2248	90,2
2232	89,3
2203	88,7
2187	87,9
2177	87,3
2142	87,1
2116	86,1
2106	84,8
2097	84,3
2081	83
2045	82,5
2035	81,8
1994	81,5
1971	80,5
1955	80,1
1935	79,7
1919	78

1903	77,3
1865	76,9
1848	76
1835	74,5
1823	73,8
1810	73,4
1797	73,1
1784	72,7
1755	72,4
1742	71,6
1716	70,2
1690	69,4
1677	68,5
1661	67,5
1648	66,8
1623	66
1606	65,7
1571	64,6
1555	64,1
1542	62,7
1513	62,3
1490	61,8
1474	61,3
1458	60,8
1448	60,3

1429	59,2
1394	58,3
1381	56,8
1352	56,4
1326	55,5
1313	54,4
1303	53,7
1261	52,5
1252	51,3
1226	50,1
1181	49,5
1171	48,8
1155	48,3
1145	47,7
1129	47,5
1100	47,2
1081	46,7
1065	46
1029	45,1
1013	44,2
990	42,9
968	41,5
958	40,8
939	40,5
913	39,8

887	39,3
877	38,4
861	37,5
852	36,2
832	35,9
777	35,3
771	34,4
755	33,8
745	33,4
703	32,7
697	31,6
694	30,5
671	29,9
655	28,5
616	28,2
610	28
587	27,4
565	26,8
542	25,6
494	24,5
465	23,3
439	22,1
413	21,3
403	20,5
355	19,8

300	17,3
274	16,7
245	14,8
194	13,3
165	12,7
139	10,8
113	9,8
84	9,2
55	7,8
29	6,5
23	5
3	4

Πίνακας 10: Αντιστοίχιση του ύψους με τη τάση του αισθητήρα

Οι τιμές αυτές που βρέθηκαν θα αποθηκευτούν χειροκίνητα στις θέσεις μνήμης του μικροελεγκτή όπως θα δούμε σε επόμενη παράγραφο.

### 5.3.3 Λειτουργία Real Time Clock (RTC)

Το Real Time Clock (RTC) είναι ένα ρολόι το οποίο παρακολουθεί το πραγματικό χρόνο. Το RTC είναι μία πολύ σημαντική λειτουργία που εκτελεί ο μικροελεγκτής στη συγκεκριμένη κατασκευή. Καθώς η μέτρηση της κατανάλωσης αποφασίστηκε να υπολογίζεται στο τέλος της ημέρας, ενώ ταυτόχρονα με τη μέτρηση αποθηκεύεται και η ημερομηνία της μέτρησης, δημιουργήθηκε η ανάγκη για τη χρήση του RTC στην εφαρμογή.

Στη συγκεκριμένη εφαρμογή, χρησιμοποιήθηκε το Timer 0 του μικροελεγκτή για τη παρακολούθηση του πραγματικού χρόνου, προκειμένου να αποφευχθεί η χρήση επιπλέον hardware. Το Timer 0 επιλέχθηκε σε 16 bit Counter/Timer mode:

`TMOD |= 0x01;`

Για την επιλογή της συχνότητας με την οποία θα αυξάνεται το timer, η συχνότητα του τοπικού ταλαντωτή διαιρέθηκε με το 12 ( $SYSCLK/12$ ). Η συχνότητα του ταλαντωτή επιλέχθηκε στα 48 MHz, επομένως το timer θα αυξάνεται με συχνότητα 4 MHz. Όταν οι καταχωρητές του Timer 0 high και low byte (TH0 και TL0 αντίστοιχα) υπερχειλίσουν,

ξεπεράσουν δηλαδή τη τιμή 65536 (0xFFFF σε HEX μορφή), συμβαίνει το interrupt για αυτό το Timer:

```
void Timer0_ISR () interrupt 1
```

Χρησιμοποιώντας έναν μετρητή που αυξάνεται κάθε φορά που συμβαίνει το interrupt αυτό από το Timer 0:

```
T0_ISR_count++;
```

υπολογίζουμε το πραγματικό χρόνο που έχει περάσει ως εξής:

```
x = (((unsigned long)T0_ISR_count << 16));  
time = (((float)x/(float)4000000));
```

όπου time είναι ο χρόνος που έχει περάσει σε δευτερόλεπτα. Πλέον, γνωρίζοντας με ακρίβεια το χρόνο που έχει περάσει σε δευτερόλεπτα, μπορούμε να ενσωματώσουμε το RTC στην εφαρμογή.

Αρχικά, όταν η μεταβλητή time γίνει μεγαλύτερη του 60 σημαίνει ότι το λεπτό άλλαξε. Τότε μηδενίζουμε τις μεταβλητές x, time και το μετρητή T0\_ISR\_count που αυξάνεται σε κάθε interrupt από το Timer 0, και το λεπτό αυξάνεται κατά 1. Όταν τα λεπτά ξεπεράσουν τα 60, η ώρα αυξάνει κατά 1 και η μεταβλητή που αντιστοιχεί στα λεπτά μηδενίζεται. Αντίστοιχη διαδικασία ακολουθείται για τις ώρες, τις μέρες, τους μήνες και το έτος. Ο ολοκληρωμένος κώδικας για το RTC, που βρίσκεται μέσα στη συνάρτηση η οποία πραγματοποιείται χάρη στο interrupt από το Timer 0, φαίνεται παρακάτω:

```
if (time >= 60) {  
    T0_ISR_count = 0;  
    x = 0;  
    time = 0;  
    minute++;  
    if (minute > 60) {  
        minute = 0;  
        hour++;  
        if (hour > 23) {  
            flag = 1;  
            hour = 0;  
            day++;  
            if (day > 28) {  
                if (month == 2) {  
                    day = 1;  
                    month++;  
                }  
            }  
        }  
        if (day > 30) {
```

```

    if ((month == 4) || (month == 6) || (month == 9) || (month == 11)) {
        day = 1;
        month++;
    }
}
if (day > 31) {
    if ((month == 1) || (month == 3) || (month == 5) || (month == 7) || (month == 8) ||
(month == 10) || (month == 12)) {
        day = 1;
        month++;
        if (month > 12) {
            month = 1;
            year++;
        }
    }
}
}
}
}
}
}

```

Μπορούμε έτσι να γνωρίζουμε με ακρίβεια την πραγματική ώρα και την ημερομηνία. Αυτό καθιστά δυνατό την αποθήκευση των μετρήσεων τη χρονική στιγμή που εμείς επιθυμούμε.

### 5.3.4 Αποθήκευση μετρήσεων

Για την εκκίνηση της διαδικασίας αποθήκευσης των μετρήσεων και της ημερομηνίας χρησιμοποιήθηκε η σημαία `flag`. Όπως βλέπουμε στον παραπάνω κώδικα, με το πέραςμα της κάθε ημέρας, η σημαία αυτή ενεργοποιείται (`flag = 1;`) και εκκινεί στη `main` συνάρτηση η διαδικασία με την οποία εκτελείται ο υπολογισμός της κατανάλωσης και στη συνέχεια η αποθήκευσή αυτής της μέτρησης, όπως και της αντίστοιχης ημερομηνίας, στις κατάλληλες θέσεις μνήμης του μικροελεγκτή. Αρχικά, γίνεται σύγκριση της τρέχουσας τάσης του αισθητήρα με τις `calibrated` τιμές της τάσης που είναι αποθηκευμένες και αντιστοίχιση αυτής της τάσης με το ύψος. Έπειτα, διαβάζεται από τη θέση μνήμης που έχουμε αναθέσει (θέση μνήμης σε HEX μορφή: `0x200C`) η τιμή της μέτρησης του ύψους της προηγούμενης ημέρας (`prevheight`) και υπολογίζεται η διαφορά του ύψους αυτού με το τρέχον ύψος (`height`), προκειμένου να υπολογιστεί η κατανάλωση:

```
FLASH_Read((char*) &prevheight, 0x200C, 2);
difference = prevheight - height;
```

Η διαφορά αυτή (ή κατανάλωση) και η ημερομηνία αποθηκεύονται στις κατάλληλες θέσεις μνήμης:

```
FLASH_Clear(m_day, 2);
```



```
FLASH_Write (m_day,(char*) &difference, 2);
FLASH_Clear (m_day+2, 2);
FLASH_Write (m_day+2,(char*) &day, 2);
FLASH_Clear (m_day+4, 2);
FLASH_Write (m_day+4,(char*) &month, 2);
FLASH_Clear (m_day+6, 2);
FLASH_Write (m_day+6,(char*) &year, 2);
```

όπου η μεταβλητή m\_day δείχνει στη θέση μνήμης όπου ξεκινάει η αποθήκευση των μετρήσεων (αρχικά αυτή η θέση μνήμης σε HEX μορφή είναι η 0x2028). Αποθηκεύονται δηλαδή με τη παρακάτω μορφή στη μνήμη του μικροελεγκτή:

difference	day	month	year
------------	-----	-------	------

Οι μεταβλητές αυτές είναι ακέραιες (integer), επομένως κάθε ένα από αυτά τα blocks καταλαμβάνει 2 bytes στη μνήμη του μικροελεγκτή. Αφού ολοκληρωθεί η διαδικασία της αποθήκευσης της μέτρησης και της ημερομηνίας, αποθηκεύεται στη κατάλληλη θέση μνήμης που χρησιμοποιήθηκε και πριν (θέση μνήμης σε HEX μορφή: 0x200C) η τιμή του τρέχοντος ύψους, ώστε να μπορεί να γίνει η σύγκριση με το ύψος στο τέλος της επόμενης μέρας:

```
FLASH_Clear (0x200C, 2);
FLASH_Write (0x200C,(char*) &height, 2);//Αποθήκευση του τρέχοντος ύψους
```

Η μεταβλητή m\_day αυξάνεται κατά 8 bytes, προκειμένου να είναι δυνατή η αποθήκευση της μέτρησης και της ημερομηνίας της επόμενης μέρας:

```
m_day += 8;
```

Τέλος, η σημαία η οποία σήμανε την εκκίνηση της διαδικασίας της αποθήκευσης των μετρήσεων, απενεργοποιείται (flag = 0;) μέχρι την επόμενη αλλαγή μέρας.

### 5.3.5 Ρύθμιση της σειριακής επικοινωνίας

Για τη σειριακή επικοινωνία του μικροελεγκτή με την android συσκευή μέσω του bluetooth module χρησιμοποιήθηκε το UART1 (θύρα P0.1, RX1) για τη λήψη δεδομένων στο μικροελεγκτή και το UART0 (θύρα P0.4, TX0) για την αποστολή των μετρήσεων από το μικροελεγκτή. Αυτή η επιλογή έγινε γιατί ενώ ο μικροελεγκτής μπορούσε να στείλει δεδομένα με τη σειριακή θύρα του UART0 στην android συσκευή μέσω του bluetooth, δεν μπορούσε να λάβει δεδομένα στην αντίστοιχη θύρα από την android συσκευή μέσω του bluetooth module. Παρόμοιο πρόβλημα όμως δεν υπήρχε με τη σειριακή θύρα του UART1, όπου τα δεδομένα λαμβάνονταν όπως προβλέπονταν.

Το UART0 ρυθμίστηκε σε μεταβλητό ρυθμό μετάδοσης 8-bit, με το REN bit ενεργοποιημένο και τις σημαίες RI και TI καθαρισμένες (cleared):

```
SCON0 = 0x10;
```

ενώ για το ρυθμό μετάδοσης χρησιμοποιήθηκε το Timer 1 σε 8-bit auto-reload mode:

```
TMOD |= 0x20;
```

Το UART1 ρυθμίστηκε αντίστοιχα σε μεταβλητό ρυθμό μετάδοσης 8-bit, με το REN bit ενεργοποιημένο και τις σημαίες RI και TI καθαρισμένες:

```
SCON1 = 0x10;
```

Για την επιλογή μέσα στο κώδικα για το ποιο UART θα λειτουργήσει σε κάθε περίπτωση, χρησιμοποιείται το bit UART, το οποίο στη περίπτωση που θέλουμε να χρησιμοποιήσουμε το UART0 παίρνει τη τιμή 0, ενώ στη περίπτωση που θέλουμε να χρησιμοποιήσουμε το UART1 παίρνει τη τιμή 1.

### 5.3.6 Οι επιλογές στην εφαρμογή

Αρχικά, αφού γίνουν οι απαραίτητες αρχικοποιήσεις των θυρών, των σειριακών θυρών επικοινωνίας UART0 και UART1, του Analog to Digital Converter (ADC0) και του τοπικού ταλαντωτή για την παροχή παλμών χρονισμού, ο μικροελεγκτής περιμένει να λάβει κάποιο χαρακτήρα στη σειριακή του θύρα. Όταν η σημαία RI1 (Receive Interrupt flag) του SCON1 (Serial Control Register 1) ενεργοποιηθεί, δηλαδή:

```
UART = 1;  
if ((SCON1 & 0x01))
```

που σημαίνει ότι η σειριακή θύρα έλαβε νέο χαρακτήρα, εκτελείται η αντίστοιχη λειτουργία, ανάλογα με το χαρακτήρα που έχει ληφθεί.

#### 5.3.6.1 Χαρακτήρας «a»: Εισαγωγή τιμών της βαθμονόμησης

Λαμβάνοντας το χαρακτήρα «a», ο μικροελεγκτής επιτρέπει στον χρήστη την εισαγωγή και αποθήκευση στις θέσεις μνήμης που έχουμε επιλέξει των τιμών που έχουμε υπολογίσει μέσω της βαθμονόμησης. Ξεκινώντας από τη θέση μνήμης σε HEX μορφή 0x2028, εισάγουμε αρχικά την υψηλότερη τιμή τάσης που αντιστοιχεί στο μεγαλύτερο ύψος, και στη συνέχεια, αυξάνοντας το δείκτη της θέσης μνήμης κατά 2 (2 bytes είναι το μέγεθος που καταλαμβάνει ένας ακέραιος στη μνήμη), εισάγουμε το ίδιο το ύψος, μέχρι να φτάσουμε σε μηδενικό ύψος:

```
m_volts = 0x2028;//Η αρχική θέση μνήμης που έχουμε επιλέξει  
do {  
    UART = 0;  
    printf("Enter mV: \n");  
    UART = 1;
```

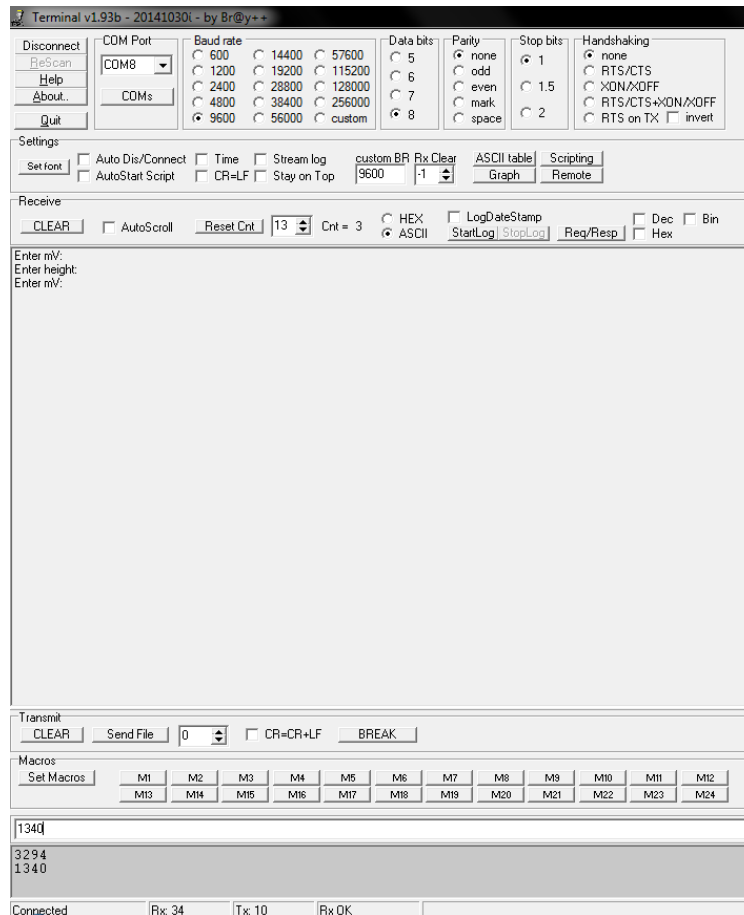
```

scanf("%d", &volts);//Εδώ εισάγουμε τη τιμή της τάσης
UART = 0;
printf("Enter height: \n");
UART = 1;
scanf("%d", &height);//Εδώ εισάγουμε το ύψος
FLASH_Clear (m_volts, 2);
FLASH_Write (m_volts,(char*) &volts, 2);
FLASH_Clear ((m_volts + 2), 2);
FLASH_Write ((m_volts + 2),(char*) &height, 2);
if (height != 0) {
    m_volts += 4;
}
}
}
while (height != 0);

```

Αυτές οι τιμές θα χρησιμοποιηθούν στη συνέχεια για την αντιστοίχιση της τρέχουσας τιμής τάσης του αισθητήρα με το ύψος.

Μέσω της android εφαρμογής όμως δεν υπάρχει επιλογή για να σταλεί ο χαρακτήρας «a» και να γίνει η εισαγωγή των calibrated τιμών. Η διαδικασία αυτή είναι απαραίτητο να πραγματοποιηθεί μόνο μία φορά στην αρχή της λειτουργίας κάθε μικροελεγκτή. Η εισαγωγή των τιμών αυτών μπορεί να γίνει με ένα οποιοδήποτε πρόγραμμα terminal που χρησιμοποιείται για τη παρακολούθηση και έλεγχο των σειριακών θυρών του μικροελεγκτή, όπως βλέπουμε παρακάτω (εικόνα 36):



Εικόνα 36: Εισαγωγή calibrated τιμών

Το ύψος αποθηκεύεται σε χιλιοστά για μεγαλύτερη ακρίβεια.

### 5.3.6.2 Χαρακτήρας «c»: Αρχικοποίηση και ενεργοποίηση του RTC

Εφόσον στη σειριακή θύρα ληφθεί ο χαρακτήρας «c», ο μικροελεγκτής αναμένει να λάβει από τον χρήστη τις τιμές για την αρχικοποίηση του RTC:

```
scanf("%d", &minute);
scanf("%d", &hour);
scanf("%d", &day);
scanf("%d", &month);
scanf("%d", &year);
```

Στη συνέχεια, γίνεται σύγκριση της τρέχουσας τιμής της τάσης του αισθητήρα με τις βαθμονομημένες τιμές των τάσεων και αντιστοίχιση με το βαθμονομημένο ύψος, το οποίο ύψος αποθηκεύεται στη θέση μνήμης που έχουμε αναθέσει ώστε να είναι δυνατή η σύγκριση για τον υπολογισμό της κατανάλωσης με το πέρασμα της ημέρας:

```
FLASH_Clear (0x200C, 2);
FLASH_Write (0x200C,(char*) &height, 2);//Αποθήκευση του τρέχοντος ύψους
```

Έπειτα πραγματοποιείται η αρχικοποίηση του timer, η αρχικοποίηση των high και low byte, όπως και του μετρητή σε περίπτωση υπερχείλισής τους:

```
T0_ISR_count = 0;//Ο μετρητής σε περίπτωση υπερχείλισης  
TH0    = 0x00;  
TL0    = 0x00;
```

Καθώς το RTC είναι μία βασική λειτουργία που εκτελεί ο μικροελεγκτής στη συγκεκριμένη κατασκευή, είναι απαραίτητη η αρχικοποίηση και ενεργοποίηση του τη πρώτη φορά που θα λειτουργήσει ο μικροελεγκτής, όπως και μετά από κάθε επαναφορά σε περίπτωση διακοπής της τροφοδότησης. Όπως θα δούμε σε επόμενο κεφάλαιο, ο συγχρονισμός του RTC του μικροελεγκτή με το πραγματικό χρόνο πραγματοποιείται μέσω της android εφαρμογής, με το πάτημα ενός κουμπιού από τον χρήστη. Με την ολοκλήρωση της αρχικοποίησης του RTC, στέλνεται το κατάλληλο string προκειμένου να προβληθεί στην εφαρμογή ότι ο συγχρονισμός ολοκληρώθηκε με επιτυχία.

### 5.3.6.3 Χαρακτήρας «m»: Αποστολή των αποθηκευμένων μετρήσεων

Εάν στη σειριακή θύρα λάβουμε το χαρακτήρα «m», πραγματοποιείται αποστολή όλων των αποθηκευμένων μετρήσεων της κατανάλωσης για κάθε μέρα που πέρασε. Αρχικά, διαβάζονται οι τιμές της μέτρησης και της ημερομηνίας που είναι αποθηκευμένες στις αντίστοιχες θέσης μνήμης του μικροελεγκτή:

```
FLASH_Read( (char*) &pr_reading, m_send, 2);  
FLASH_Read( (char*) &pr_day, (m_send+2), 2);  
FLASH_Read( (char*) &pr_month, (m_send+4), 2);  
FLASH_Read( (char*) &pr_year, (m_send+6), 2);
```

και η αποστολή γίνεται με τη γραμμή του κώδικα:

```
printf("am%04d%02d%02d%04de\n", pr_reading, pr_day, pr_month, pr_year);
```

όπου m\_send είναι η θέση μνήμης από την οποία ξεκινάει η συγκεκριμένη μέτρηση, pr\_reading είναι η μέτρηση της κατανάλωσης και οι μεταβλητές pr\_day, pr\_month και pr\_year αντιπροσωπεύουν την ημερομηνία της αντίστοιχης μέτρησης. Οι μετρήσεις στέλνονται με τη σειρά, ξεκινώντας από τη θέση μνήμης που έχουμε διαλέξει να αποθηκεύεται η 1<sup>η</sup> μέτρηση με την αντίστοιχη ημερομηνία μέχρι τη θέση μνήμης της τελευταίας μέτρησης και ημερομηνίας που έχει αποθηκευτεί. Χρησιμοποιώντας το terminal πρόγραμμα για τη παρακολούθηση της σειριακής επικοινωνίας με το μικροελεγκτή, βλέπουμε ότι οι μετρήσεις στέλνονται ως εξής (εικόνα 37):

```
am000016122017e  
am000017122017e  
am003718122017e  
am003719122017e  
am004920122017e
```

Εικόνα 37: Σειριακή αποστολή μετρήσεων

Οι μετρήσεις πρέπει να στέλνονται σε μορφή η οποία να είναι αναγνωρίσιμη από την android εφαρμογή που θα χρησιμοποιηθεί. Στη συγκεκριμένη περίπτωση, το string της κάθε μέτρησης που μεταδίδεται πρέπει να ξεκινάει με τους χαρακτήρες «am» και να ολοκληρώνεται με το χαρακτήρα «e», ενώ η μέτρηση, η μέρα, ο μήνας και το έτος της μέτρησης πρέπει να έχουν συγκεκριμένο πλάτος πεδίου (τέσσερις, δύο, δύο και τέσσερις χαρακτήρες αντίστοιχα), με μηδενικά να συμπληρώνονται σε περίπτωση που το πλάτος είναι μικρότερο του απαιτούμενου.

Η αποθήκευση των επόμενων μετρήσεων ξεκινάει από τη θέση μνήμης του μικροελεγκτή που είχαμε θέσει αρχικά (θέση μνήμης σε HEX μορφή: 0x2028).

#### 5.3.6.4 Χαρακτήρας «r»: Λειτουργία παραλαβής

Η λήψη του χαρακτήρα «r» στη σειριακή θύρα σημαίνει ότι ο χρήστης θέλει να πραγματοποιήσει παραλαβή κάποιου υγρού στη δεξαμενή, όπως πχ παραλαβή πετρελαίου. Εδώ ο μικροελεγκτής αρχίζει και στέλνει συνεχώς τη τρέχουσα μέτρηση του ύψους (αφού έχει γίνει η αντιστοίχιση της τρέχουσας τάσης με το calibrated ύψους), μέχρι να ληφθεί στη σειριακή θύρα ο χαρακτήρας «s», ο οποίος δηλώνει την ολοκλήρωση της παραλαβής από τη μεριά του χρήστη (εικόνα 38):

```
ar1098e  
ar1103e  
ar1115e  
ar1131e  
ar1131e  
ar1131e  
an0523e
```

Εικόνα 38: Αποστολή των πληροφοριών στη λειτουργία της παραλαβής

Οι μετρήσεις, όπως και στη προηγούμενη περίπτωση, στέλνονται σε τέτοια μορφή ώστε να είναι δυνατή η παρουσίασή τους μέσω της εφαρμογής. Αρχικά το string που μεταδίδεται ξεκινάει με τους χαρακτήρες «ar» και ολοκληρώνεται με το χαρακτήρα «e», ενώ η μέτρηση του ύψους πρέπει να έχει συγκεκριμένο αριθμό χαρακτήρων, όπου όπως και πριν συμπληρώνονται όσα μηδενικά χρειάζονται στη περίπτωση που δεν φτάνει αυτόν τον επιθυμητό αριθμό χαρακτήρων. Με την ολοκλήρωση της παραλαβής από τη μεριά του

χρήστη, στέλνεται η διαφορά του τελικού ύψους με το αρχικό, προκειμένου να προβληθεί στην οθόνη της συσκευής η ακριβής ποσότητα που παραλήφθηκε. Το string που στέλνεται έχει και στη συγκεκριμένη περίπτωση συγκεκριμένη μορφή, ξεκινώντας με τους χαρακτήρες «an» και καταλήγοντας με το χαρακτήρα «e».

Η τελευταία μέτρηση του ύψους αποθηκεύεται στη θέση μνήμης που χρησιμοποιούμε για τη σύγκριση του ύψους και τον υπολογισμό της κατανάλωσης με το πέρασμα της ημέρας. Αυτό γίνεται για την αποφυγή λάθους κατά τον υπολογισμό της κατανάλωσης, καθώς διαφορετικά με το πέρασμα της ημέρας που έγινε η παραλαβή, η κατανάλωση θα υπολογίζονταν ως αρνητικός αριθμός (το ύψος της προηγούμενης μέρας θα ήταν μικρότερο του ύψους που υπολογίστηκε με το πέρασμα της μέρας της παραλαβής) και δεν θα μπορούσαμε να βγάλουμε τα σωστά συμπεράσματα.





## 6 Η Bluetooth επικοινωνία

Το Bluetooth είναι ένα ανοικτό πρότυπο για την επικοινωνία μέσω ραδιοσυχνοτήτων (Radio Frequency, RF) μικρής εμβέλειας. Η τεχνολογία Bluetooth χρησιμοποιείται κυρίως για την εγκατάσταση ασύρματων προσωπικών δικτύων (wireless personal area networks, WPAN). Η τεχνολογία Bluetooth έχει ενσωματωθεί σε πολλούς τύπους συσκευών για επιχειρήσεις και καταναλωτές, όπως κινητά τηλέφωνα, φορητούς υπολογιστές, αυτοκίνητα, εκτυπωτές, ηλεκτρολόγια, ποντίκια και ακουστικά. Αυτό επιτρέπει στους χρήστες να σχηματίζουν δίκτυα ad hoc (αυτοργανωμένο δίκτυο ή δίκτυο κατ' απαίτηση, δηλαδή δίκτυο που δεν βασίζεται σε κάποια προϋπάρχουσα υποδομή) μεταξύ μιας ευρείας ποικιλίας συσκευών για τη μεταφορά φωνής και δεδομένων. Το Bluetooth είναι μια χαμηλού κόστους, χαμηλής κατανάλωσης τεχνολογία που παρέχει το μηχανισμό για τη δημιουργία μικρών ad hoc ασύρματων δικτύων, γνωστά ως piconets. Ένα piconet αποτελείται από δύο ή περισσότερες συσκευές με Bluetooth σε κοντινή απόσταση μεταξύ τους που λειτουργούν στο ίδιο κανάλι χρησιμοποιώντας την ίδια hopping sequence συχνότητα. Ένα παράδειγμα ενός piconet είναι μια σύνδεση που βασίζεται στο Bluetooth μεταξύ ενός κινητού τηλεφώνου και ενός ακουστικού.



Εικόνα 39: Το λογότυπο του Bluetooth

Τα Bluetooth piconets είναι συχνά εγκατεστημένα σε μια προσωρινή και μεταβαλλόμενη βάση, η οποία προσφέρει ευελιξία στις επικοινωνίες και δυνατότητα επεκτασιμότητας μεταξύ των κινητών συσκευών. Μερικά βασικά πλεονεκτήματα της τεχνολογίας Bluetooth είναι:

- Αντικατάσταση καλωδίων. Η τεχνολογία Bluetooth αντικαθιστά μία ποικιλία καλωδίων, όπως αυτά που παραδοσιακά χρησιμοποιούνται για περιφερειακές συσκευές (π.χ. καλώδια για τη σύνδεση ποντικιού και ηλεκτρολογίου), εκτυπωτές και ακουστικά που συνδέονται με επιτραπέζιους υπολογιστές, φορητούς υπολογιστές, κινητά τηλέφωνα κ.τ.λ.
- Εύκολη κοινή χρήση αρχείων. Μια συσκευή με δυνατότητα Bluetooth μπορεί να διαμορφώσει ένα piconet για τη κοινή χρήση αρχείων με άλλες συσκευές με Bluetooth, όπως φορητούς υπολογιστές.
- Ασύρματος συγχρονισμός. Το Bluetooth μπορεί να παρέχει αυτόματο συγχρονισμό μεταξύ συσκευών με Bluetooth.

- Σύνδεση στο Διαδίκτυο. Μια συσκευή Bluetooth με δυνατότητα σύνδεσης στο διαδίκτυο μπορεί να μοιραστεί αυτή τη σύνδεση με άλλες συσκευές Bluetooth.

## 6.1 Τα τεχνολογικά χαρακτηριστικά του Bluetooth

Το Bluetooth λειτουργεί χωρίς να χρειάζεται άδεια χρήσης στη ζώνη συχνοτήτων ISM (Industrial, Scientific, and Medical) από 2,400 GHz έως 2,4835 GHz. Σε αυτήν τη ζώνη λειτουργούν ποικίλες τεχνολογίες, συμπεριλαμβανομένου του προτύπου ασύρματου τοπικού δικτύου (wireless local area network, WLAN) IEEE 802.11b/g. Το Bluetooth χρησιμοποιεί τη τεχνολογία εξάπλωσης φάσματος (frequency hopping spread spectrum, FHSS) για τις μεταδόσεις. Το FHSS μειώνει τα σφάλματα στις μεταδόσεις και τα σφάλματα εξαιτίας των παρεμβολών αλλά παρέχει ελάχιστη ασφάλεια μετάδοσης. Με την τεχνολογία FHSS, οι επικοινωνίες μεταξύ των συσκευών Bluetooth με βασικό/ενισχυμένο ρυθμό μετάδοσης (BR/EDR) χρησιμοποιούν 79 διαφορετικά 1 megahertz (MHz) κανάλια μεταβάλλοντας τις συχνότητες περίπου 1600 φορές ανά δευτερόλεπτο για συνδέσεις δεδομένων ή φωνής και 3200 φορές ανά δευτερόλεπτο κατά το paging (το paging χρησιμοποιείται για να συνδεθεί μία συσκευή με μία άλλη όταν είναι γνωστή η διεύθυνση και η τιμή του ρολογιού της συσκευής αυτής) και τη σάρωση για inquiries (αιτημάτων). Ένα κανάλι χρησιμοποιείται για πολύ μικρό χρονικό διάστημα (π.χ., 625 ms για συνδέσεις δεδομένων ή φωνής), ακολουθούμενο από αλλαγή σε άλλο κανάλι, χάρη σε μια προκαθορισμένη ψευδοτυχαία ακολουθία. Αυτή η διαδικασία επαναλαμβάνεται συνεχώς με αυτή την ακολουθία αλλαγής συχνότητας.

Το Bluetooth παρέχει επίσης έλεγχο ισχύος ραδιοζεύξεων, το οποίο επιτρέπει στις συσκευές να διαπραγματεύονται και να ρυθμίζουν τη ραδιοισχύ τους σύμφωνα με τις μετρήσεις ισχύος του σήματος. Κάθε συσκευή σε ένα Bluetooth δίκτυο μπορεί να καθορίσει την ένδειξη της έντασης του λαμβανόμενου σήματος (received signal strength indication, RSSI) και να ζητήσει από την άλλη συσκευή στο δίκτυο να ρυθμίσει το σχετικό επίπεδο ραδιοισχύος, δηλαδή να αυξήσει ή να μειώσει προοδευτικά την ισχύ μετάδοσης. Αυτό γίνεται για εξοικονόμηση ενέργειας και / ή για να διατηρούνται τα χαρακτηριστικά του λαμβανόμενου σήματος εντός ενός προτιμώμενου εύρους.

Ο συνδυασμός ενός συστήματος αλλαγής συχνότητας και ελέγχου ισχύος ραδιοζεύξεων παρέχει στο Bluetooth μία επιπλέον, αν και περιορισμένη, προστασία από υποκλοπές και την κακόβουλη πρόσβαση. Το σύστημα αλλαγής συχνοτήτων, το οποίο είναι κυρίως μια τεχνική για την αποφυγή παρεμβολών, καθιστά λίγο πιο δύσκολο για κάποιον να εντοπίσει και να καταγράψει τις Bluetooth μεταδόσεις από το να υποκλέψει μεταδόσεις από τεχνολογίες που χρησιμοποιούν σταθερή συχνότητα, όπως εκείνες που χρησιμοποιούνται στο IEEE 802.11b/g.

Το φάσμα των συσκευών Bluetooth BR/EDR χαρακτηρίζεται από τρεις κατηγορίες που καθορίζουν τη διαχείριση ενέργειας. Ο πίνακας 1 συνοψίζει τις κατηγορίες, συμπεριλαμβανομένων των επιπέδων ισχύος τους σε milliwatts (mW) και ντεσιμπέλ αναφορικά με ένα milliwatt (dBm), καθώς και την εμβέλειά τους σε μέτρα (m). Οι περισσότερες μικρές συσκευές που λειτουργούν με μπαταρία είναι κλάσης 2, ενώ κλάσης 1

είναι συνήθως συσκευές όπως προσαρμογείς USB για επιτραπέζιους και φορητούς υπολογιστές, σημεία πρόσβασης, καθώς και άλλες συσκευές που τροφοδοτούνται από το ηλεκτρολογικό δίκτυο.

Τύπος	Ισχύς	Μέγιστη ισχύς	Εμβέλεια	Παραδείγματα συσκευών
Κλάση 1	Υψηλή	100 mW (20 dBm)	Μέχρι 100 μέτρα	USB προσαρμογείς, σημεία πρόσβασης
Κλάση 2	Μεσαία	2,5 mW (4 dBm)	Μέχρι 10 μέτρα	Φορητές συσκευές, προσαρμογείς Bluetooth
Κλάση 3	Χαμηλή	1 mW (0 dBm)	Μέχρι 1 μέτρο	Προσαρμογείς Bluetooth

**Πίνακας 11: Κλάσεις Bluetooth BR/EDR**

Για να επιτρέπεται στις συσκευές Bluetooth να βρίσκουν άλλες συσκευές και να συνδέονται μεταξύ τους, έχουν καθοριστεί οι λειτουργίες ανιχνευσιμότητας (discoverable) και συνδεσιμότητας (connectable). Μια συσκευή σε discoverable κατάσταση παρακολουθεί περιοδικά ένα φυσικό κανάλι όπου πραγματοποιούνται αιτήματα για συνδέσεις (βασισμένο σε ένα συγκεκριμένο σύνολο συχνοτήτων) και απαντά σε ένα αίτημα στο συγκεκριμένο κανάλι με τη διεύθυνση της συσκευής, την τιμή του τοπικού ρολογιού και άλλα χαρακτηριστικά που απαιτούνται για το paging και στη συνέχεια για τη σύνδεση σε αυτό. Μια συσκευή σε connectable κατάσταση μπορεί να παρακολουθεί περιοδικά το φυσικό κανάλι σάρωσης page της και να ανταποκρίνεται σε ένα paging που προέρχεται από το εν λόγω κανάλι για να ξεκινήσει μια σύνδεση δικτύου. Οι συχνότητες που σχετίζονται με το φυσικό κανάλι σάρωσης page μιας συσκευής βασίζονται στη Bluetooth διεύθυνση της συσκευής αυτής. Επομένως, είναι σημαντικό να είναι γνωστή η διεύθυνση και το τοπικό ρολόι μιας συσκευής για το paging και, στη συνέχεια, για τη σύνδεση με τη συσκευή αυτή.

### 6.1.1 Βασικοί και ενισχυμένοι ρυθμοί μετάδοσης

Οι συσκευές Bluetooth μπορούν να υποστηρίξουν πολλαπλούς ρυθμούς μετάδοσης δεδομένων χρησιμοποιώντας το Bluetooth και εναλλάσσοντας τα επίπεδα ελέγχου μεσαίας πρόσβασης (Medium Access Controls, MAC) και φυσικά επίπεδα (PHY). Επειδή οι προδιαγραφές του Bluetooth σχεδιάζονται ώστε να είναι συμβατές προς τα πίσω, μια συσκευή με νέες προδιαγραφές που υποστηρίζει υψηλότερους ρυθμούς μετάδοσης, μπορεί να υποστηρίξει επίσης χαμηλότερους ρυθμούς μετάδοσης δεδομένων που υποστηρίζονται

από προηγούμενες συσκευές με παλαιότερες προδιαγραφές (π.χ. μια συσκευή EDR υποστηρίζει επίσης ρυθμούς που ορίζονται για συσκευές BR).

Οι εκδόσεις Bluetooth 1.1 και 1.2 υποστηρίζουν μόνο ταχύτητες μετάδοσης έως και 1 megabit ανά δευτερόλεπτο (Mbps), η οποία είναι γνωστή ως βασικός ρυθμός (Basic Rate, BR) και μπορεί να επιτύχει απόδοση περίπου 720 kilobits ανά δευτερόλεπτο (kbps). Ο ενισχυμένος ρυθμός μετάδοσης Bluetooth (Enhanced Data Rate, EDR), που εισήχθη στην έκδοση Bluetooth 2.0, υποστηρίζει ταχύτητες μετάδοσης έως 3 Mbps και απόδοση περίπου 2,1 Mbps. Το BR χρησιμοποιεί Gaussian Shift Keying Shift (GFSK) διαμόρφωση για να επιτευχθεί η ταχύτητα του 1 Mbps. Το EDR χρησιμοποιεί διαμόρφωση Differential Quaternary Phase Shift Keying (DQPSK) περιστροφής  $\pi/4$  για την επίτευξη ταχύτητας 2 Mbps και διαμόρφωση 8 phase Differential Phase Shift Keying (8DPSK) για επίτευξη ταχύτητας 3 Mbps.

### 6.1.2 Υψηλός ρυθμός μετάδοσης με εναλλασσόμενα MAC/PHY

Με την προδιαγραφή υψηλής ταχύτητας Bluetooth 3.0 +, οι συσκευές μπορούν να υποστηρίξουν ταχύτερους ρυθμούς μετάδοσης δεδομένων με τη χρήση εναλλασσόμενου MAC/PHY (Alternate MAC/PHY, AMP). Αυτό είναι γνωστό ως Bluetooth High Speed (HS). Το πρότυπο 802.11 AMP έχει σχεδιαστεί για να παρέχει ταχύτητες μετάδοσης έως 24 Mbps χρησιμοποιώντας διαμόρφωση Orthogonal Frequency-Division Multiplexing (OFDM).

### 6.1.3 Χαμηλής κατανάλωσης

Το Bluetooth χαμηλής κατανάλωσης ή Low Energy (LE) εισήχθη στην έκδοση Bluetooth 4.0. Το Bluetooth LE σχεδιάστηκε κατά κύριο λόγο για να φέρει τη τεχνολογία Bluetooth σε συσκευές που λειτουργούν με μπαταρίες, όπως ιατρικές συσκευές και άλλοι αισθητήρες. Οι βασικοί τεχνολογικοί στόχοι του Bluetooth LE (σε σύγκριση με το Bluetooth BR/EDR) περιλαμβάνουν χαμηλότερη κατανάλωση ενέργειας, μειωμένες απαιτήσεις μνήμης, αποτελεσματικότερες διαδικασίες εντοπισμού και σύνδεσης, μικρά μήκη πακέτων και απλά πρωτόκολλα και υπηρεσίες.

Οι βασικές διαφορές μεταξύ Bluetooth BR/EDR και LE φαίνονται στο παρακάτω πίνακα:

Χαρακτηριστικό	Bluetooth BR/EDR	Bluetooth LE
Φυσικά κανάλια RF	79 κανάλια, 1 MHz το καθένα	40 κανάλια, 2 MHz το καθένα
Εντοπισμός/Σύνδεση	Inquiry/Paging	Advertising
Αριθμός των piconet slaves	7 (ενεργά)/255(σύνολο)	Απεριόριστα

Ιδιωτικότητα διεύθυνσης συσκευής	Καμία	Ιδιωτική διεύθυνση συσκευής είναι διαθέσιμη
Μέγιστη ταχύτητα μετάδοσης	1–3 Mbps	1 Mbps μέσω διαμόρφωσης GFSK
Αλγόριθμος κρυπτογράφησης	E0/SAFER+	AES-CCM
Εμβέλεια	30 μέτρα	50 μέτρα
Μέγιστη ισχύς εξόδου	100 mW (20 dBm)	10 mW (10 dBm)

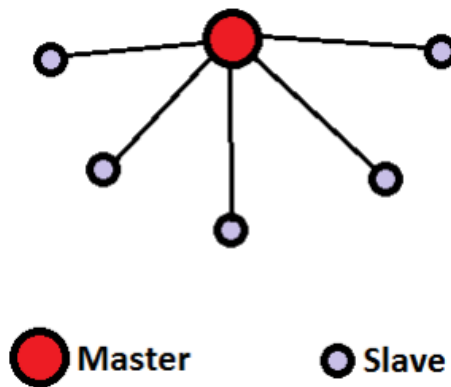
Πίνακας 12: Διαφορές μεταξύ Bluetooth BR/EDR και LE

## 6.2 Η αρχιτεκτονική του Bluetooth

Το Bluetooth επιτρέπει στις συσκευές να δημιουργούν ad hoc δίκτυα. Τα δίκτυα ad hoc επιτρέπουν την εύκολη δημιουργία συνδέσεων μεταξύ συσκευών στην ίδια φυσική περιοχή (π.χ. το ίδιο δωμάτιο) χωρίς τη χρήση επιπλέον συσκευών. Ένας Bluetooth client είναι απλώς μια συσκευή με ασύρματο πομποδέκτη και λογισμικό Bluetooth που ενσωματώνει τη στοίβα πρωτοκόλλου Bluetooth και τις διεπαφές.

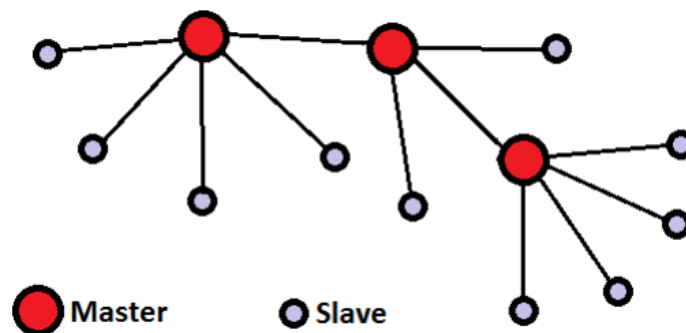
Η προδιαγραφή Bluetooth παρέχει διαχωρισμό των καθηκόντων για την εκτέλεση λειτουργιών της στοίβας μεταξύ ενός host και ενός controller. Ο host είναι υπεύθυνος για τα πρωτόκολλα υψηλότερου επιπέδου, όπως για το Logical Link Control and Adaptation Protocol (L2CAP) και το Protocol Discovery Service (SDP). Οι λειτουργίες του host εκτελούνται από μια συσκευή όπως ένα φορητό υπολογιστή ή ένα smartphone. Ο controller είναι υπεύθυνος για τα χαμηλότερα επίπεδα, συμπεριλαμβανομένου του ασυρμάτου πομποδέκτη, του baseband και του ελέγχου και της διαχείρισης συνδέσεων. Οι λειτουργίες του controller εκτελούνται από ενσωματωμένο ή εξωτερικό (π.χ. USB) προσαρμογέα Bluetooth. Ο host και ο controller ανταλλάσσουν πληροφορίες μεταξύ τους χρησιμοποιώντας τυποποιημένες επικοινωνίες μέσω της διεπαφής Host Controller (HCI). Αυτό το τυποποιημένο HCI επιτρέπει στους host και τους controllers από διαφορετικούς κατασκευαστές να συνεργάζονται. Σε ορισμένες περιπτώσεις, οι λειτουργίες του host και του controller είναι ενσωματωμένες σε μία μόνο συσκευή. Τα Bluetooth ακουστικά είναι ένα τέτοιο παράδειγμα.

Στην εικόνα 40 απεικονίζεται η βασική τοπολογία δικτύου Bluetooth. Σε ένα piconet μια συσκευή χρησιμεύει ως master, με όλες τις άλλες συσκευές στο piconet να δρουν ως slaves. Τα BR/EDR piconets μπορούν να επεκταθούν για να συμπεριλάβουν έως και 7 ενεργές slave συσκευές και μέχρι 255 ανενεργές slave συσκευές. Η νέα τεχνολογία Bluetooth LE επιτρέπει έναν απεριόριστο αριθμό slave συσκευών.



Εικόνα 40: Τοπολογία piconet

Η master συσκευή ελέγχει και εγκαθιστά το δίκτυο, συμπεριλαμβανομένου του καθορισμού του συστήματος αλλαγής της συχνότητας του δικτύου. Αν και μόνο μία συσκευή μπορεί να χρησιμεύσει ως master για κάθε piconet, η πολυπλεξία διαίρεσης χρόνου (time division multiplexing, TDM) επιτρέπει σε ένα slave σε ένα piconet να ενεργεί ταυτόχρονα ως master για ένα άλλο piconet, δημιουργώντας έτσι μια αλυσίδα δικτύων. Αυτή η αλυσίδα ονομάζεται scatternet και επιτρέπει τη δικτύωση διαφόρων συσκευών σε μια εκτεταμένη απόσταση σε μια δυναμική τοπολογία που μπορεί να αλλάξει κατά τη διάρκεια μιας συγκεκριμένης περιόδου. Καθώς μια συσκευή πλησιάζει ή απομακρύνεται από τη master συσκευή, η τοπολογία μπορεί να αλλάξει, όπως και οι συνδέσεις των συσκευών στο άμεσο δίκτυο. Η παρακάτω εικόνα απεικονίζει ένα scatternet που περιλαμβάνει τρία piconets:



Εικόνα 41: Τοπολογία scatternet

Τα βασικά πρωτόκολλα Bluetooth δεν παρέχουν δυνατότητες δρομολόγησης για συσκευές που βρίσκονται σε διαφορετικά scatternets, δεν μπορούν δηλαδή να επικοινωνήσουν μεταξύ τους εάν δεν βρίσκονται στο ίδιο piconet. Τα scatternets διατίθενται μόνο στις συσκευές BR / EDR, επειδή η τεχνολογία Bluetooth LE δεν υποστηρίζει αυτή τη λειτουργία.

### 6.3 Το Bluetooth module HC-05

Στη κατασκευή μας χρειαζόμασταν μία οικονομική, χωρίς την απαίτηση για κάποια ιδιαίτερη ασφάλεια σε περίπτωση υποκλοπής, και εύκολη λύση για την επικοινωνία μεταξύ

του μικροελεγκτή και της android συσκευής. Επιλέχθηκε έτσι η επικοινωνία μέσω της bluetooth τεχνολογίας. Το hardware που διαλέξαμε για να επιτύχουμε την επικοινωνία αυτή ήταν το Bluetooth module HC-05.



Εικόνα 42: Το Bluetooth module HC-05

Το HC-05 module είναι μια εύχρηστη συσκευή Bluetooth SPP (πρωτόκολλο σειριακής θύρας, Serial Port Protocol), σχεδιασμένο για ασύρματες σειριακές συνδέσεις. Αυτό το σειριακό Bluetooth module χρησιμοποιεί τεχνολογία Bluetooth V2.0 + EDR 3Mbps με πλήρη ασύρματο πομποδέκτη 2.4GHz και baseband. Χρησιμοποιεί το ενιαίο κύκλωμα Bluetooth CSR Bluecore 04—External με τεχνολογία CMOS και AFH (Adaptive Frequency Hopping Feature). Οι μικρές διαστάσεις του, 12,7mmx27mm το καθιστούν μία πολύ καλή λύση για εφαρμογές ενσωματωμένων.

Το Bluetooth module HC-05 είναι ένα master/slave module. Χωρίς κάποια επιπλέον ρύθμιση, η προεπιλογή είναι να δουλεύει σαν slave. Ο ρόλος τον οποίο εξυπηρετεί το module αυτό (Master ή Slave) μπορεί να ρυθμιστεί μόνο χρησιμοποιώντας τις εντολές AT (AT COMMANDS). Τα modules σε λειτουργία slave δεν μπορούν να εκκινήσουν μία σύνδεση με μία άλλη συσκευή Bluetooth, μπορούν όμως να δεχτούν αιτήματα για συνδέσεις από άλλες συσκευές. Ένα module σε λειτουργία master μπορεί να εκκινήσει μια σύνδεση με άλλες συσκευές.

### 6.3.1 Τα χαρακτηριστικά του HC-05

Τα τεχνικά χαρακτηριστικά του είναι:

- Τυπική -80dBm ευαισθησία.
- Έως + 4dBm ισχύ εκπομπής RF.
- 3,3V έως 5V I/O.
- Σειριακή διεπαφή (UART) με προγραμματιζόμενη ταχύτητα μετάδοσης.
- Ενσωματωμένη κεραία.

Τα χαρακτηριστικά του λογισμικού του είναι τα παρακάτω:

- Προεπιλεγμένη ταχύτητα μετάδοσης baud σε slave λειτουργία: 9600, bits δεδομένων: 8, bit διακοπής (stop bit): 1.

- Ως προεπιλογή, αυτόματη σύνδεση κατά την εκκίνηση με τη συσκευή που συνδέθηκε τελευταία.
- Κωδικός ασφαλείας «1234» ως προεπιλογή.

### 6.3.2 Περιγραφή των επαφών του HC-05

Το HC-05 module έχει 6 επαφές (εικόνα 43):



Εικόνα 43: Οι επαφές του HC-05

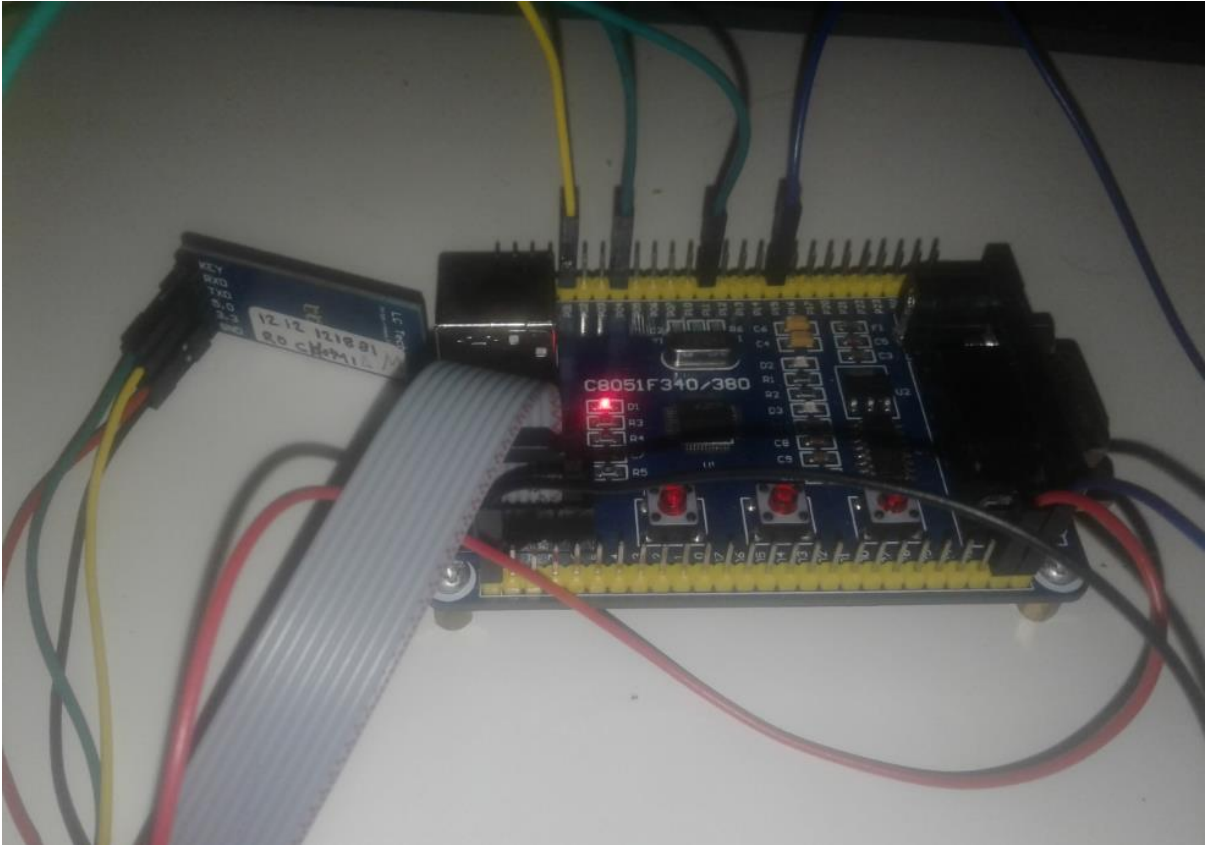
- Key: Χρησιμοποιείται για να εισέλθουμε στη λειτουργία των AT εντολών.
- RXD και TXD: Αυτές οι δύο επαφές χρησιμοποιούνται για τη σειριακή επικοινωνία. Η επαφή RXD χρησιμοποιείται για τη λήψη δεδομένων, ενώ η επαφή TXD για την αποστολή δεδομένων.
- 5.0 και 3.3: Τροφοδοσία με 5 και 3,3V αντίστοιχα.
- GND: Η επαφή η οποία πρέπει να συνδεθεί με τη γείωση

### 6.3.3 Το HC-05 module στη κατασκευή

Για την επικοινωνία του μικροελεγκτή με τη android συσκευή χρησιμοποιήθηκε το HC-05 module με τις εργοστασιακές ρυθμίσεις, δηλαδή χρησιμοποιήθηκε σε λειτουργία slave, με 9600 ρυθμό μετάδοσης baud και με κωδικό ασφαλείας το «1234». Το HC-05 module συνδέθηκε με το μικροελεγκτή C8051F340 ως εξής:

- Η επαφή RXD με την επαφή P0.4 του μικροελεγκτή, η οποία έχει ρυθμιστεί ως η σειριακή θύρα για την αποστολή δεδομένων από το μικροελεγκτή (UART0 TX).
- Η επαφή TXD με την επαφή P0.1 του μικροελεγκτή, η οποία έχει ρυθμιστεί ως η σειριακή θύρα για την λήψη δεδομένων στο μικροελεγκτή (UART1 RX).
- Η επαφή 3.3 με την αντίστοιχη του μικροελεγκτή για τη τροφοδοσία του module.
- Η επαφή GND με την αντίστοιχη του μικροελεγκτή για τη σύνδεση του module με τη γείωση.





Εικόνα 44: Η σύνδεση του HC-05 module με το μικροελεγκτή C8051F340



## 7 Η Android εφαρμογή

### 7.1 Το λειτουργικό σύστημα Android

Το Android είναι η πιο δημοφιλής πλατφόρμα για φορητές συσκευές στο κόσμο. Το Android χρησιμοποιείται από δισεκατομμύρια κινητές συσκευές σε περισσότερες από 190 χώρες σε όλο τον κόσμο. Είναι η περισσότερο εγκατεστημένη βάση οποιασδήποτε πλατφόρμας κινητής τηλεφωνίας και αυξάνεται γρήγορα καθώς εκατομμύρια χρήστες προστίθενται καθημερινά, οι οποίοι ενεργοποιούν τις συσκευές Android για πρώτη φορά και αρχίζουν να αναζητούν εφαρμογές, παιχνίδια και άλλο ψηφιακό περιεχόμενο. Το Android προσφέρει μια πλατφόρμα παγκόσμιας κλάσης για τη δημιουργία εφαρμογών και παιχνιδιών για χρήστες του Android, καθώς και μια ανοιχτή αγορά για την άμεση διανομή τους.



Εικόνα 45: Το εικονίδιο του Android

Υπάρχουν πάνω από 1,3 εκατομμύρια ενεργές εφαρμογές στο Google Play App Store. Το Android είναι η πρώτη πλατφόρμα ανοιχτού κώδικα (open source), βασισμένη σε Linux και στις σύγχρονες φορητές συσκευές. Η Google το έχει αναπτύξει για τους κατασκευαστές φορητών συσκευών όπως η T-Mobile, η Sprint Nextel, η Google, η Intel, η Samsung κ.τ.λ. Προσφέρει στους καταναλωτές μια πλουσιότερη, λιγότερο δαπανηρή και καλύτερη κινητή εμπειρία, καθώς και διάφορες λειτουργίες όπως 3D, SQLite, Connectivity, WebKit, Dalvik και FreeType κ.τ.λ. Δεδομένου ότι το Android παρέχει λειτουργικό σύστημα ανοικτού κώδικα, οι χρήστες και οι προγραμματιστές μπορούν να πάρουν τον πηγαίο κώδικα, αλλά μόνο κάτω από συγκεκριμένους κανόνες και προϋποθέσεις. Κάθε φορά που ο χρήστης θέλει να εγκαταστήσει οποιαδήποτε εφαρμογή, η περιγραφή της, καθώς και μια λίστα αιτημάτων δικαιωμάτων, παρέχονται με δυνατότητα αναθεώρησης πριν από την εγκατάστασή της, αν διαπιστώσει ο χρήστης ότι τα δικαιώματα είναι υπερβολικά πολλά ή απαράδεκτα. Η πλατφόρμα ανοιχτού κώδικα Android απαιτεί ισχυρή και σύνθετη αρχιτεκτονική ασφαλείας για την εξασφάλιση της ασφάλειας των προσωπικών δεδομένων, των προσωπικών πληροφοριών, των εφαρμογών και του δικτύου του χρήστη αλλά έχει και κάποιους περιορισμούς για τους προγραμματιστές, γεγονός που αυξάνει τον κίνδυνο για τους τελικούς χρήστες.

### 7.1.1 Τα επίπεδα του λογισμικού στο λειτουργικό σύστημα Android

Το Android OS έχει σχεδιαστεί με τη μορφή διαφορετικών επιπέδων όπου είναι στοιβαγμένα λογισμικά. Τα επίπεδα αυτά περιλαμβάνουν εφαρμογές Android, λειτουργικό σύστημα, χρόνο που τρέχει το Android, ενδιαμέσο λογισμικό, υπηρεσίες και βιβλιοθήκες. Κάθε επίπεδο της στοίβας και τα αντίστοιχα στοιχεία σε κάθε επίπεδο είναι στενά συνδεδεμένα και παρέχουν διαφορετικό είδος υπηρεσιών στο επίπεδο ακριβώς πάνω από αυτό καθώς και ένα βέλτιστο περιβάλλον ανάπτυξης και εκτέλεσης εφαρμογών για κινητές συσκευές.

Η στοίβα λογισμικού του Android αποτελείται από διαφορετικά επίπεδα που παρέχουν διαφορετικές υπηρεσίες στο επίπεδο ακριβώς πάνω από αυτό, όπως φαίνεται στο παρακάτω σχήμα (εικόνα 46):



Εικόνα 46: Η στοίβα των λογισμικών της αρχιτεκτονικής του Android

#### 7.1.1.1 Ο πυρήνας Linux

Στο κάτω μέρος της στοίβας του λογισμικού Android, υπάρχει ένας πυρήνας Linux. Λειτουργεί ως η καρδιά ολόκληρου του συστήματος. Παρέχει διάφορες λειτουργίες όπως διαχείριση μνήμης, διαχείριση διαδικασιών, διαχείριση συσκευών, ρυθμίσεις ασφαλείας κ.τ.λ. στο σύστημα Android, όπως επίσης και όλα τα βασικά προγράμματα οδήγησης υλικού (device drivers) για το υλικό με το οποίο αλληλεπιδρά η συσκευή.

### 7.1.1.2 Βιβλιοθήκες

Πάνω από το επίπεδο του πυρήνα υπάρχει ένα σύνολο βιβλιοθηκών, συμπεριλαμβανομένου του διαχειριστή επιφάνειας που συνθέτει τα παράθυρα στην οθόνη, όπως Open GL ES για 3D βιβλιοθήκη, SGL για 2D γραφικά, Media Framework για αναπαραγωγή και καταγραφή διαφόρων μορφών ήχου, εικόνας και βίντεο, μηχανή περιήγησης WebKit, βάση δεδομένων SQLite για αποθήκευση, βιβλιοθήκη ασφαλείας για το διαδίκτυο Open SSL κ.τ.λ. Αυτές οι βιβλιοθήκες βασίζονται σε γλώσσα C ή C++.

### 7.1.1.3 Περιβάλλον Android Runtime

Το περιβάλλον Runtime βρίσκεται στο ίδιο επίπεδο με τις βιβλιοθήκες, είναι το τρίτο τμήμα της αρχιτεκτονικής και διατίθεται στο δεύτερο επίπεδο της στοίβας από το κάτω μέρος. Περιλαμβάνει ένα σύνολο βασικών βιβλιοθηκών Java που επιτρέπει στους προγραμματιστές εφαρμογών Android να γράφουν εφαρμογές Android χρησιμοποιώντας την γνωστή γλώσσα προγραμματισμού Java. Περιλαμβάνει επίσης το ART (Android Runtime), το οποίο είναι παρόμοιο με το DVM (Dalvik Virtual Machine) και ειδικά σχεδιασμένο και βελτιστοποιημένο για το Android. Κάθε διαδικασία εκτελείται ξεχωριστά σε μια εικονική μηχανή (virtual machine). Λειτουργεί με αρχεία DEX και κώδικες σε dex byte. Παρέχει Ahead-of-Time (AOT) σύνταξη και βελτιωμένο debugging, βελτιωμένη ασφάλεια, διαχείριση μνήμης, γρήγορη απόδοση και υποστήριξη threading. Βοηθάει επίσης τον χρήστη να εκτελεί ταυτόχρονα πολλαπλές εφαρμογές.

### 7.1.1.4 Πλαίσιο εφαρμογών

Πάνω από το επίπεδο των βιβλιοθηκών και του Android Runtime, υπάρχει το επίπεδο του πλαισίου εφαρμογής. Το επίπεδο αυτό παρέχει πολλά πακέτα υπηρεσιών υψηλότερου επιπέδου σε εφαρμογές. Παρέχει τις λειτουργίες του τηλεφώνου όπως τη κοινή χρήση δεδομένων, τη διαχείριση των πόρων κ.τ.λ. Τα πακέτα αυτά είναι τα ακόλουθα:

- Διαχειριστής δραστηριότητας (Activity Manager): Ελέγχει και διαχειρίζεται τον κύκλο ζωής των εφαρμογών.
- Διαχειριστής πόρων (Resource Manager): Διαχειρίζεται και παρέχει πρόσβαση σε ενσωματωμένους πόρους όπως γραφικά, strings και ρυθμίσεις χρώματος.
- Διαχειριστής ειδοποιήσεων (Notification Manager): Επιτρέπει σε όλες τις εφαρμογές να εμφανίζουν προσαρμοσμένες ειδοποιήσεις στη γραμμή κατάστασης (status bar) και ειδοποιήσεις προς το χρήστη.
- Διαχειριστής τοποθεσίας (Location Manager): Όταν ο χρήστης εισέρχεται ή φεύγει από μια συγκεκριμένη γεωγραφική θέση, το location manager ενεργοποιεί τις ειδοποιήσεις σχετικά με τις αλλαγές τοποθεσίας χρησιμοποιώντας το GPS ή τις κεραίες κινητής.
- Διαχείριση πακέτων (Package Manager): Το σύστημα με το οποίο οι εφαρμογές είναι σε θέση να ανακτήσουν δεδομένα για άλλες εφαρμογές που είναι εγκατεστημένες στη συσκευή.

- Διαχειριστής τηλεφωνίας (Telephony Manager): Διαχειρίζεται και επιτρέπει σε μία εφαρμογή την πρόσβαση στις φωνητικές κλήσεις, τις ρυθμίσεις της σύνδεσης δικτύου και τη κατάσταση και την υπηρεσία πληροφοριών των συνδρομητών.
- Διαχειριστής παραθύρων (Window Manager): Χρησιμοποιείται ένα επεκτάσιμο σύνολο προβολών και διατάξεων (layouts) για τη δημιουργία διεπαφών του χρήστη με τις εφαρμογές.
- Παροχέας περιεχομένου (Content Provider): Είναι το σύστημα το οποίο επιτρέπει και διαχειρίζεται την ανταλλαγή δεδομένων μεταξύ των εφαρμογών.

#### 7.1.1.5 Οι Android εφαρμογές

Οι εφαρμογές Android βρίσκονται στο ανώτατο επίπεδο της στοίβας λογισμικού Android. Περιλαμβάνει τόσο τις εργοστασιακές εφαρμογές όσο και τις εφαρμογές από άλλες πηγές. Οι εργοστασιακές εφαρμογές παρέχουν τις βασικές εφαρμογές Android, όπως την εφαρμογή για SMS, το πρόγραμμα για κλήσεις, το πρόγραμμα περιήγησης στο διαδίκτυο και το διαχειριστή επαφών. Οι εφαρμογές από άλλες πηγές εγκαθίστανται αργότερα από τους προγραμματιστές για τον εντοπισμό σφαλμάτων ή για δοκιμές και από τους χρήστες μετά την αγορά της συσκευής.

#### 7.1.2 Οι Android εκδόσεις

Η πρώτη έκδοση του Android παρουσιάστηκε στις 23 Σεπτεμβρίου 2008. Το Android λαμβάνει πολλές ενημερώσεις που βελτιώνουν διαδοχικά το λειτουργικό σύστημα προσθέτοντας νέα χαρακτηριστικά και διορθώνοντας προηγούμενα σφάλματα. Παρακάτω βλέπουμε τις διαφορετικές εκδόσεις του Android:

Όνομα	Έκδοση	Ημερομηνία κυκλοφορίας
Χωρίς όνομα	1.0	23 Σεπτεμβρίου, 2008
Αρχικά γνωστό σαν «Petit Four»	1.1	9 Φεβρουαρίου, 2009
Cupcake	1.5	27 Απριλίου, 2009
Donut	1.6	15 Σεπτεμβρίου, 2009
Eclair	2.0 - 2.1	26 Οκτωβρίου, 2009

Froyo	2.2 - 2.2.3	20 Μαΐου, 2010
Gingerbread	2.3 - 2.3.7	6 Δεκεμβρίου, 2010
Honeycomb	3.0 - 3.2.6	22 Φεβρουαρίου, 2011
Ice Cream Sandwich	4.0 - 4.0.4	18 Οκτωβρίου, 2011
Jelly Bean	4.1 - 4.3.1	9 Ιουλίου, 2012
KitKat	4.4 - 4.4.4	31 Οκτωβρίου, 2013
Lollipop	5.0 - 5.1.1	12 Νοεμβρίου, 2014
Marshmallow	6.0 - 6.0.1	5 Οκτωβρίου, 2015
Nougat	7.0 - 7.1.2	22 Αυγούστου, 2016

Πίνακας 13: Οι Android εκδόσεις

## 7.2 Το προγραμματιστικό περιβάλλον Android Studio

Το Android Studio είναι το επίσημο ολοκληρωμένο προγραμματιστικό περιβάλλον (Integrated Development Environment, IDE) για ανάπτυξη εφαρμογών Android, βασισμένο στην τεχνολογία IntelliJ IDEA. Εκτός από το ισχυρό εργαλείο επεξεργασίας κώδικα και τα εργαλεία ανάπτυξης IntelliJ, το Android Studio προσφέρει ακόμα περισσότερες λειτουργίες που βελτιώνουν την παραγωγικότητα κατά την ανάπτυξη Android εφαρμογών, όπως:

- Ένα ευέλικτο σύστημα κατασκευής με βάση το Gradle.
- Ένας γρήγορος και πλούσιος σε λειτουργίες εξομοιωτής.
- Ένα ενοποιημένο περιβάλλον όπου δίνεται η δυνατότητα ανάπτυξης εφαρμογών για όλες τις συσκευές Android.
- Instant Run για τη προώθηση αλλαγών στην τρέχουσα εφαρμογή χωρίς την ανάγκη δημιουργίας ενός νέου APK (Android Package Kit).
- Πρότυπα κώδικα και ενσωμάτωση του GitHub για βοήθεια στη δημιουργία κοινών λειτουργιών των εφαρμογών και στην εισαγωγή δείγματος κώδικα.
- Εκτεταμένα εργαλεία και πλαίσια δοκιμών.

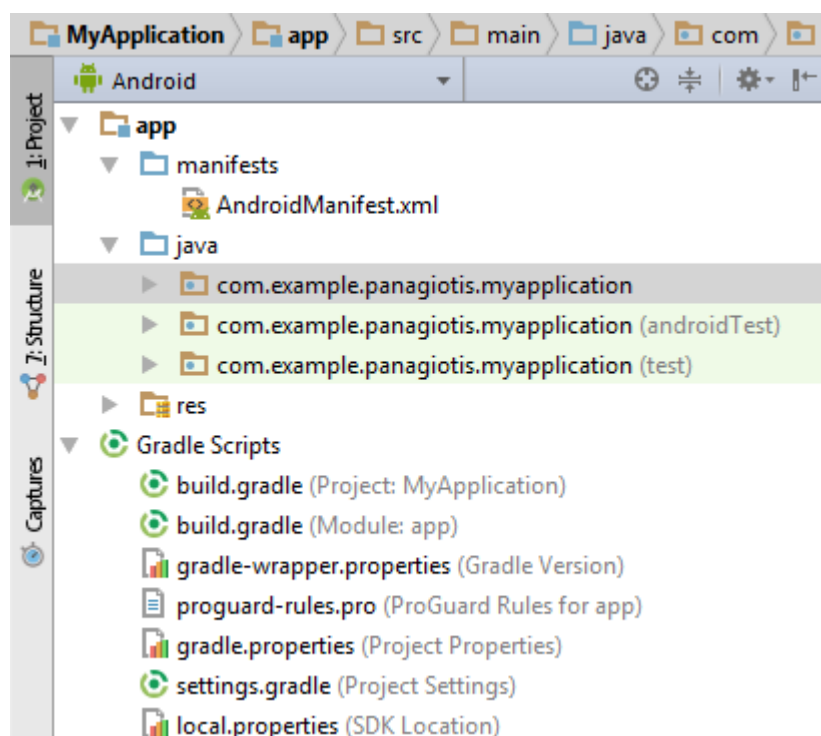
- Εργαλεία για τον εντοπισμό προβλημάτων σχετικά με την απόδοση, τη χρηστικότητα, τη συμβατότητα της έκδοσης και άλλων προβλημάτων.
- Υποστήριξη C ++ και NDK.
- Ενσωματωμένη υποστήριξη για την πλατφόρμα Google Cloud, διευκολύνοντας την ενσωμάτωση του Google Cloud Messaging και του App Engine.

### 7.2.1 Η δομή του project

Κάθε project στο Android Studio περιέχει μία ή περισσότερες ενότητες με αρχεία πηγαίου κώδικα (source code) και αρχεία πόρων (resource files). Οι τύποι ενότητων περιλαμβάνουν:

- Ενότητες (modules) εφαρμογών Android
- Ενότητες βιβλιοθήκης
- Ενότητες Google App Engine

Από προεπιλογή, το Android Studio εμφανίζει τα αρχεία του project στην προβολή Android project, όπως φαίνεται στην εικόνα 47.



Εικόνα 47: Η προβολή του Android project

Αυτή η προβολή οργανώνεται σε ενότητες για γρήγορη πρόσβαση στα βασικά αρχεία προέλευσης του project.

Όλα τα αρχεία build είναι ορατά κάτω από την ενότητα Gradle Scripts και κάθε ενότητα της εφαρμογής περιέχει τους ακόλουθους φακέλους:

- Manifests: Περιέχει το αρχείο AndroidManifest.xml.



- Java: Περιέχει τα αρχεία πηγαίου κώδικα Java, συμπεριλαμβανομένου του κώδικα δοκιμής JUnit.
- Res: Περιέχει resources όπως διατάξεις XML, UI strings και εικόνες bitmap.

### 7.3 Το σύστημα διαχείρισης βάσεων δεδομένων SQLite

Το SQLite είναι μια δυναμική βιβλιοθήκη που υλοποιεί μια αυτοματοποιημένη, χωρίς την ανάγκη κάποιου server και συγκεκριμένων διαμορφώσεων, μηχανή βάσης δεδομένων SQL (Structured Query Language). Είναι μια βάση δεδομένων η οποία δεν χρειάζεται κάποια διαμόρφωση για την ενσωμάτωση στο σύστημά μας. Η SQLite μηχανή δεν είναι μια αυτόνομη διαδικασία όπως άλλες βάσεις δεδομένων, δηλαδή μπορούμε να τη συνδέσουμε είτε στατικά είτε δυναμικά, ανάλογα με τις απαιτήσεις της εφαρμογής. Το SQLite επεξεργάζεται απευθείας τα αρχεία στα οποία αποθηκεύει δεδομένα.

Οι λόγοι για τους οποίους το SQLite είναι μία καλή επιλογή για ένα σύστημα βάσης δεδομένων είναι οι εξής:

- Το SQLite δεν απαιτεί ξεχωριστή διαδικασία σε server ή κάποιο σύστημα για να λειτουργήσει (serverless).
- Το SQLite χρειάζεται μηδενική διαμόρφωση, κάτι που σημαίνει πως δεν χρειάζεται εγκατάσταση ή κάποια διαχείριση από το χρήστη.
- Μια πλήρης βάση δεδομένων SQLite αποθηκεύεται σε ένα ενιαίο αρχείο στο δίσκο.
- Το SQLite είναι πολύ μικρό και ελαφρύ, καθώς καταλαμβάνει λιγότερο από 400KB στη περίπτωση που είναι πλήρως διαμορφωμένο ή λιγότερο από 250KB με κάποιες προαιρετικές λειτουργίες να έχουν παραλειφθεί.
- Το SQLite είναι αυτόνομο, κάτι που σημαίνει πως δεν εξαρτάται από εξωτερικούς παράγοντες.
- Το SQLite είναι πλήρως προσβάσιμο από πολλαπλές διεργασίες ή threads.
- Το SQLite υποστηρίζει τα περισσότερα από τα χαρακτηριστικά της γλώσσας ερωτήσεων που βρίσκονται στο πρότυπο SQL92 (SQL2).
- Το SQLite είναι γραμμένο σε ANSI-C και παρέχει απλή και εύκολη χρήση του API.
- Το SQLite είναι διαθέσιμο σε UNIX (Linux, Mac OS-X, Android, iOS) και Windows (Win32, WinCE, WinRT).

### 7.4 Ανάπτυξη της εφαρμογής

Για την ολοκλήρωση της κατασκευής και την επικοινωνία του μικροελεγκτή με τον χρήστη στόχος ήταν η δημιουργία μιας εφαρμογής για λογισμικό Android η οποία θα είναι απλή, εύχρηστη και θα προσφέρει τις απαραίτητες λειτουργίες που χρειάζεται ο μέσος χρήστης. Τα βασικά χαρακτηριστικά στην Android εφαρμογή που θέλαμε να αναπτύξουμε ήταν η λήψη και αποθήκευση των μετρήσεων της κατανάλωσης από το μικροελεγκτή, η χρησιμοποίηση της λειτουργίας της παραλαβής όταν ο χρήστης θέλει να πραγματοποιήσει παραλαβή στη δεξαμενή κάποιου υγρού, όπως π.χ. πετρελαίου, καθώς και η συνοπτική και ευανάγνωστη παρουσίαση των μετρήσεων αυτών.

#### 7.4.1 Λήψη των μετρήσεων από το μικροελεγκτή

Αρχικά, για την επικοινωνία της Android συσκευής με το Bluetooth module που είναι συνδεδεμένο με το μικροελεγκτή χρησιμοποιήθηκε το παράδειγμα κώδικα «BluetoothChat» που βρίσκεται στην επίσημη σελίδα του Android για προγραμματιστές. Το παράδειγμα αυτό προσαρμόστηκε στις ανάγκες της εφαρμογής μας, κρατώντας όμως τα βασικά χαρακτηριστικά του ίδια.

Η εφαρμογή λαμβάνει δεδομένα από το μικροελεγκτή σε δύο περιπτώσεις: η μία είναι όταν γίνεται λήψη και αποθήκευση των μετρήσεων της κατανάλωσης και η άλλη περίπτωση όταν επιθυμούμε να πραγματοποιήσουμε παραλαβή και πρέπει να γνωρίζουμε με ακρίβεια το τρέχον περιεχόμενο της δεξαμενής, καθώς και την ακριβή ποσότητα που παραλήφθηκε. Για να ξεχωρίζουμε αυτές περιπτώσεις όταν λαμβάνουμε δεδομένα, επιλέξαμε τα δεδομένα που στέλνει ο μικροελεγκτής να τα στέλνει με μία συγκεκριμένη μορφή, προκειμένου, ανάλογα με το πρώτο χαρακτήρα του string των δεδομένων αυτών, να καταλαβαίνει ο μικροελεγκτής αν πρόκειται για μέτρηση της κατανάλωσης ή για το τρέχον περιεχόμενο της δεξαμενής.

Τα bytes των δεδομένων που στέλνει ο μικροελεγκτής τα αποθηκεύουμε σε ένα string, στη μεταβλητή readmessage:

```
readMessage = new String((byte[]) msg.obj, "UTF-8");
```

και στη συνέχεια, με τη μέθοδο append του StringBuilder συμπληρώνεται η μεταβλητή recDataString, μέχρι να ολοκληρωθεί η αποστολή της μέτρησης, είτε αυτή είναι η μέτρηση της κατανάλωσης, είτε είναι το ύψος στο οποίο βρίσκεται τη δεδομένη στιγμή το φλοτέρ. Το string της μέτρησης και στις δύο περιπτώσεις έχει σαν τελευταίο χαρακτήρα το «e»:

```
recDataString.append(readMessage);  
int endOfLineIndex = recDataString.indexOf("e");
```

Έπειτα, ανάλογα με το πρώτο χαρακτήρα της μεταβλητής recDataString, η εφαρμογή καταλαβαίνει εάν η μέτρηση πρόκειται για τη κατανάλωση ή για το τρέχον ύψος στο οποίο βρίσκεται το φλοτέρ.

Εάν αυτός ο χαρακτήρας είναι ο χαρακτήρας «m», έχουμε τη μέτρηση της κατανάλωσης. Το string των δεδομένων έχει τη μορφή που είδαμε σε προηγούμενο κεφάλαιο, ξεκινάει δηλαδή με το χαρακτήρα «m» και ακολουθεί η μέτρηση της κατανάλωσης (δηλαδή η διαφορά του ύψους με το πέρασμα της ημέρας) και η ημερομηνία, με το «e» να είναι ο τελευταίος χαρακτήρας. Το string δηλαδή θα έχει τη μορφή «mxxxxddmmyyyye», όπου xxxx είναι η μέτρηση της διαφοράς του ύψους σε χιλιοστά. Διαχωρίζοντας το string σε κομμάτια ώστε να πάρουμε τις σωστές τιμές για τη κατανάλωση και την ημερομηνία, ο κώδικας πήρε τη παρακάτω μορφή:

```
if (recDataString.charAt(0) == 'm') {
```

```

String dataInPrint = recDataString.substring(1, endOfLineIndex); //extract string
String temp_received = recDataString.substring(1, 5);
String day1_received = recDataString.substring(5, 7);
String month1_received = recDataString.substring(7, 9);
String year1_received = recDataString.substring(9, 13);

int num = new Integer(Integer.parseInt(temp_received));

day_received = new Integer(Integer.parseInt(day1_received));

month_received = new Integer(Integer.parseInt(month1_received));

year_received = new Integer(Integer.parseInt(year1_received));

float surface = new Float(Float.parseFloat(meter.getNumber()));
count = (int) ((float) surface * num);
//εδώ γίνεται ο υπολογισμός της κατανάλωσης σε λίτρα
Calendar newDate = getCalendar(day_received,
                                month_received, year_received);

reading.setDate(newDate);
reading.setCount(count);
meter.add(reading);
setResult(RESULT_OK);
Toast.makeText(getApplicationContext(), "Wait before pressing back!",
Toast.LENGTH_LONG).show();
recDataString.delete(0, recDataString.length());
dataInPrint = " ";
}

```

Ο παραπάνω κώδικας εκτελείται κάθε φορά που λαμβάνουμε δεδομένα που περιλαμβάνουν μέτρηση της κατανάλωσης, μέχρι να ολοκληρωθεί η αποστολή των μετρήσεων από τη μεριά του μικροελεγκτή. Η τιμή της διαφοράς του ύψους αποθηκεύεται στη μεταβλητή τύπου ακέραιου (integer) num, αφού έχει προηγηθεί ο διαχωρισμός του string στην επιθυμητή μορφή. Η μέτρηση αποθηκεύεται στη βάση δεδομένων καλώντας τη συνάρτηση add της κλάσης Meter, έχοντας ως μεταβλητή τη κλάση Reading, η οποία περιέχει τις πληροφορίες για τη τιμή της μέτρησης της κατανάλωσης και την ημερομηνία της μέτρησης. Κατά τη διάρκεια της λήψης των μετρήσεων επιλέχθηκε να προβάλλεται ένα μήνυμα στην οθόνη, το «Wait before pressing back!», προκειμένου να περιμένει ο χρήστης να ολοκληρωθεί η λήψη των μετρήσεων προτού επιστρέψει στην αρχική οθόνη.

Όταν ο χρήστης επιθυμεί να πραγματοποιήσει παραλαβή και έχει στείλει το σχετικό αίτημα στο μικροελεγκτή, ο μικροελεγκτής στέλνει τη μέτρηση του τρέχοντος ύψους στο οποίο βρίσκεται το φλοτέρ. Η μέτρηση αυτή στέλνεται στη μορφή «xxxxxx», όπου xxxx είναι η μέτρηση του ύψους. Σε αυτή τη περίπτωση κρατάμε από τη μεταβλητή recDataString

τη μέτρηση όπως έγινε και πριν και προβάλλεται στο σχετικό πλαίσιο στην οθόνη της συσκευής η μέτρηση του περιεχομένου της δεξαμενή σε λίτρα. Έχουμε έτσι το κώδικα:

```
else if (recDataString.charAt(0) == 'r') {
    String dataInPrint = recDataString.substring(1, endOfLineIndex); // extract string
    int num = new Integer(Integer.parseInt(dataInPrint));
    float surface = new Float(Float.parseFloat(meter.getNumber()));
    count = (int) ((float) surface * num);
    reading.setCount(count);

    if (viewCount != count) {
        updateViews();
    }

    recDataString.delete(0, recDataString.length()); //clear all string data
    dataInPrint = " ";
}
```

Αφού έχει βρεθεί το περιεχόμενο της δεξαμενή σε λίτρα, προβάλλεται αυτή η τιμή (μεταβλητή count) στην οθόνη χάρη στη συνάρτηση updateViews(). Η διαδικασία αυτή πραγματοποιείται συνέχεια, με τη τιμή του περιεχομένου της δεξαμενή να ανανεώνεται στην οθόνη της συσκευής σε περίπτωση που έχει αλλάξει. Όταν η παραλαβή έχει ολοκληρωθεί, ο μικροελεγκτής στέλνει το string που περιέχει τη διαφορά του τελικού ύψους με το αρχικό, προκειμένου ο χρήστης να έχει ακριβής εικόνα της ποσότητας που παρέλαβε. Η διαδικασία που ακολουθείται είναι παρόμοια με πριν, με τη διαφορά ότι το string ξεκινάει με το χαρακτήρα «n», ενώ η τιμή που περιέχει το string χρησιμοποιείται για να υπολογιστεί η ποσότητα που παραλήφθηκε, η οποία προβάλλεται σε ένα πεδίο textview:

```
int litre_received = (int) ((float) surface * num);
mLitReceived.setText(String.valueOf(litre_received));
```

Όπως είδαμε σε προηγούμενο κεφάλαιο, τα strings των δεδομένων που στέλνει ο μικροελεγκτής ξεκινάνε με το χαρακτήρα «a». Στην εφαρμογή όμως υποθέσαμε ότι τα strings αυτά ξεκινάνε με ένα από τους χαρακτήρες «m», «s» και «r». Αυτό γίνεται γιατί όταν χρησιμοποιείται ο κώδικας του παραδείγματος «BluetoothChat», στον οποίο όπως αναφέρθηκε προηγουμένως βασίστηκε η επικοινωνία της εφαρμογής με το Bluetooth module του μικροελεγκτή, ο πρώτος χαρακτήρας που λαμβάνει η συσκευή μέσω της Bluetooth επικοινωνίας χάνεται.

#### 7.4.2 Αποθήκευση σε βάση δεδομένων SQLite

Οι μετρήσεις που λαμβάνονται από το μικροελεγκτή πρέπει να αποθηκεύονται στη μνήμη της Android συσκευής, προκειμένου να είναι δυνατή η παρουσίαση τους είτε σε μορφή γραφήματος, είτε σε μορφή λίστας στο χρήστη. Για αυτό το λόγο υλοποιήθηκε στην εφαρμογή μια βάση δεδομένων τύπου SQLite για την αποθήκευση των απαραίτητων δεδομένων.

Για τις ανάγκες της εφαρμογής δημιουργήθηκαν δύο πίνακες βάσης δεδομένων: ένας πίνακας που περιλαμβάνει πληροφορίες για το μετρητή, δηλαδή όνομα και τιμή της διαμέτρου της δεξαμενής στην οποία βρίσκεται η κατασκευή, και ένας πίνακας όπου αποθηκεύονται όλες οι μετρήσεις που λαμβάνονται από το μικροελεγκτή. Η δημιουργία των δύο πινάκων γίνεται με τη παρακάτω συνάρτηση:

```
private Database(Context context) {  
  
    super(context, DATABASE_NAME, null, DATABASE_VERSION);  
    // Create table column definitions  
    meter = new TableMeter(this);  
    reading = new TableReading(this);  
  
    // Mapping between class and table  
    tables = new HashMap<>();  
    tables.put(Meter.class, meter);  
    tables.put(Reading.class, reading);  
}
```

Δημιουργήθηκαν έτσι οι δύο πίνακες δεδομένων, meter και reading. Ο πίνακας meter περιέχει τις πληροφορίες για το μετρητή. Η δημιουργία των στηλών του πίνακα γίνεται στο παρακάτω κώδικα:

```
public TableMeter(Database database) {  
  
    super(database, METER, true);  
  
    addColumn(new DatabaseColumn(NUMBER, DataType.TEXT));  
    addColumn(new DatabaseColumn(NAME, DataType.TEXT));  
  
    meter.setForeignKeyConstraint(true);  
    meter.setForeignTable(METER);  
    meter.setForeignColumn(_ID);  
}
```

Οι στήλες δημιουργούνται με τη συνάρτηση addcolumn, η οποία έχει ως μεταβλητή τη νέα στήλη δεδομένων που δημιουργείται. Η στήλη NUMBER περιλαμβάνει τη τιμή της διαμέτρου, ενώ η στήλη NAME το όνομα του μετρητή. Οι στήλες λαμβάνουν τιμές με τη παρακάτω συνάρτηση:

```
public ContentValues getValues(Object object, boolean forUpdate) {  
  
    final Meter meter = (Meter) object;  
  
    final ContentValues values = new ContentValues();  
    if (forUpdate)
```

```

values.put(_ID, meter.getId());
values.put(NUMBER, meter.getNumber());
values.put(NAME, meter.getName());

return values;
}

```

Στο πίνακα reading καταγράφονται οι πληροφορίες για τη κάθε μέτρηση, δηλαδή η τιμή της μέτρησης της κατανάλωσης, η ημερομηνία της μέτρησης αυτής, καθώς και το όνομα του μετρητή από τον οποίο λάβαμε τη μέτρηση. Ο τρόπος με τον οποίο γίνεται η προσθήκη των στηλών που περιέχουν αυτές τις πληροφορίες φαίνεται παρακάτω:

```

public TableReading(Database database) {

    super(database, READING, true);

    addColumn(new DatabaseColumn(METER, DataType.INTEGER));
    addColumn(new DatabaseColumn(DATE, DataType.TEXT));
    addColumn(new DatabaseColumn(COUNT, DataType.INTEGER));

    final DatabaseColumn meter = getColumn(METER);
    meter.setForeignKeyConstraint(true);
    meter.setForeignTable(METER);
    meter.setForeignColumn(_ID);
}

```

Η διαδικασία που ακολουθείται είναι η ίδια με πριν, δηλαδή με τη συνάρτηση addColumn προστίθεται η κάθε στήλη, μία για το όνομα του μετρητή (METER), μία για την ημερομηνία στην οποία έγινε η μέτρηση (DATE) και μία στήλη για τη τιμή της μέτρησης της κατανάλωσης (COUNT). Η διαδικασία που ακολουθείται για τη συμπλήρωση των στηλών με τις τιμές είναι επίσης παρόμοια με πριν:

```

public ContentValues getValues(Object object, boolean forUpdate) {

    final Reading reading = (Reading) object;

    final ContentValues values = new ContentValues();
    if (forUpdate)
        values.put(_ID, reading.getId());
        values.put(METER, reading.getMeter().getId());
        values.put(DATE, DateHelper.getDatabaseDate(reading.getDate()));
        values.put(COUNT, reading.getCount());

    return values;
}

```

### 7.4.3 Δημιουργία γραφήματος για τη παρουσίαση των μετρήσεων

Για τη παρουσίαση των μετρήσεων της κατανάλωσης σε μορφή γραφήματος χρησιμοποιήθηκε η βιβλιοθήκη MPAndroidChart, η οποία είναι μία βιβλιοθήκη που υποστηρίζει την ενσωμάτωση γραφημάτων σε διάφορες μορφές στην Android εφαρμογή μας. Αφού εισάγουμε τη βιβλιοθήκη αυτή στις βιβλιοθήκες μας, μπορούμε να χρησιμοποιήσουμε τις λειτουργίες που προσφέρει. Στη περίπτωση μας, ο σκοπός ήταν η δημιουργία ενός γραφήματος για τη παρουσίαση της μηνιαίας κατανάλωσης συναρτήσει των αντίστοιχων μηνών. Υλοποιήσαμε έτσι χάρη στη βιβλιοθήκη MPAndroidChart ένα γράφημα στηλών (bar graph), όπου στον οριζόντιο άξονα x εισάγαμε τους μήνες του έτους και στον κάθετο άξονα y το σύνολο της κατανάλωσης για το κάθε μήνα του έτους. Ένα κομμάτι της συνάρτησης με την οποία εισάγουμε τα δεδομένα στους άξονες του γραφήματος φαίνεται παρακάτω:

```
for (int month = 0; month < 12; month++) {  
    float x = calendar.get(Calendar.MONTH);  
    float y = meter.getConsumptionFor(calendar);  
  
    if (calendar.get(Calendar.YEAR) == year)  
        currentYear.add(new BarEntry(x, y));  
    else  
        priorYear.add(new BarEntry(x, y));  
    calendar.add(Calendar.MONTH, -1);  
}
```

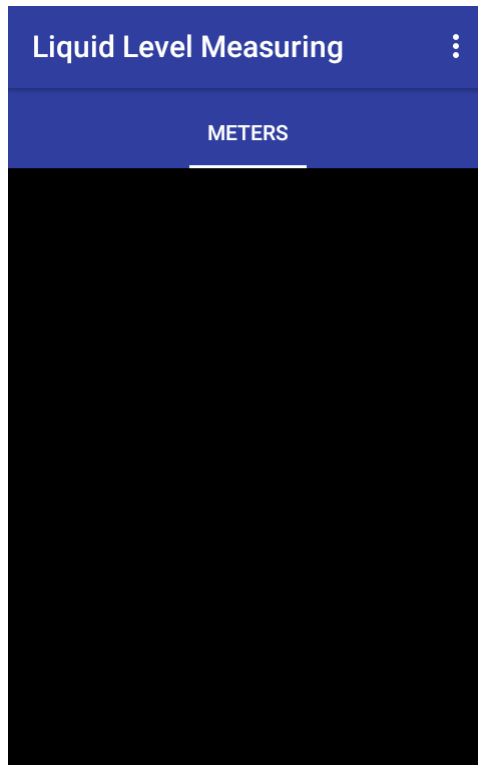
Βλέπουμε ότι στον οριζόντιο άξονα x έχουμε εισάγει τους μήνες και στο κάθετο άξονα y εισάγουμε τη μηνιαία κατανάλωση, χρησιμοποιώντας τη συνάρτηση `getconsumptionFor` της κλάσης `Meter`, έχοντας ως μεταβλητή το μήνα για τον οποίο θέλουμε αυτή τη μηνιαία κατανάλωση.

## 7.5 Παρουσίαση της android εφαρμογής

Τα χαρακτηριστικά και οι λειτουργίες της εφαρμογής που αναπτύχθηκε για τις ανάγκες μας παρουσιάζονται συνοπτικά παρακάτω.

### 7.5.1 Εισαγωγή νέου μετρητή

Αφού εγκαταστήσουμε και τρέξουμε την εφαρμογή, η αρχική σελίδα της εφαρμογής (εικόνα 48) είναι κενή:



**Εικόνα 48: Αρχική σελίδα**

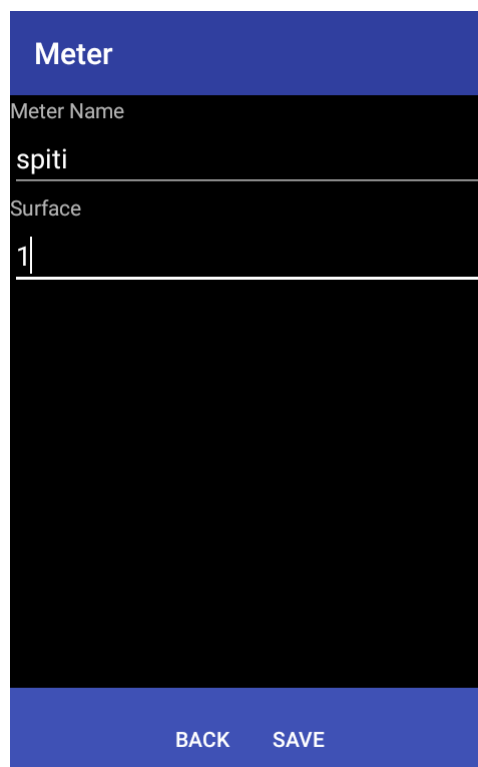
και μπορούμε μέσω του μενού που βρίσκεται πάνω δεξιά της οθόνης να εισάγουμε ένα νέο μετρητή διαλέγοντας την επιλογή «New Meter» (εικόνα 49). Οι επιλογές «Export» και «Import» που βλέπουμε στη παρακάτω εικόνα μας δίνουν την επιλογή είτε να εξάγουμε σε μορφή αρχείου text (.txt) τα δεδομένα προηγούμενων μετρήσεων που είναι αποθηκευμένα στη μνήμη της συσκευής είτε να εισάγουμε δεδομένα μετρήσεων σε μορφή που είναι αναγνωρίσιμη από τη εφαρμογή.





Εικόνα 49: Το μενού για την επιλογή του νέου μετρητή

Στη συνέχεια, έχοντας επιλέξει «New Meter», μεταφερόμαστε στη σελίδα όπου διαλέγουμε το όνομα που επιθυμούμε για το μετρητή και εισάγουμε σε μέτρα την επιφάνεια της δεξαμενής στην οποία βρίσκεται η κατασκευή (εικόνα 50):

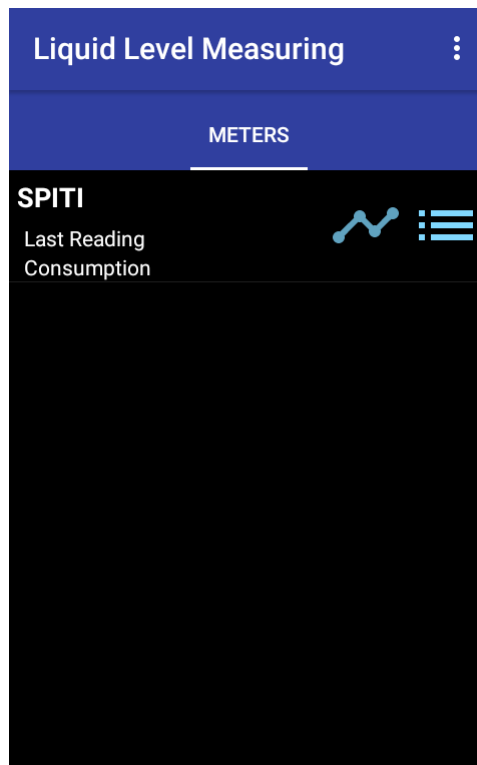


Εικόνα 50: Επιλογή ονόματος μετρητή και διαμέτρου δεξαμενής

Η εισαγωγή της τιμής της επιφάνειας της δεξαμενής στην οποία έχει τοποθετηθεί η κατασκευή είναι απαραίτητη, καθώς ο μικροελεγκτής υπολογίζει μόνο το ύψος στο οποίο βρίσκεται το φλοτέρ και αποθηκεύει τη κατανάλωση σαν διαφορά του ύψους. Επομένως, για τον υπολογισμό της κατανάλωσης και του τρέχοντος περιεχομένου της δεξαμενής σε λίτρα, είναι απαραίτητο να γνωρίζουμε την επιφάνεια προκειμένου να χρησιμοποιήσουμε το τύπο για τον υπολογισμό του όγκου ορθογωνίου:

$$V = \text{μήκος} * \text{πλάτος} * \text{ύψος}$$

Πατώντας την επιλογή «SAVE» ο νέος μετρητής αποθηκεύεται και εμφανίζεται στην αρχική σελίδα της εφαρμογής (εικόνα 51):



Εικόνα 51: Η αρχική σελίδα όπου εμφανίζεται στη λίστα και ο καινούριος μετρητής

Στη καρτέλα που περιέχει το μετρητή βλέπουμε ότι εμφανίζονται πληροφορίες για την ημερομηνία της τελευταίας μέτρησης (Last Reading), τη τελευταία μέτρηση της κατανάλωσης (Consumption), ενώ τα εικονίδια στις εικόνες 52 και 53 παρουσιάζουν σε γράφημα τη μηνιαία κατανάλωση και τη λίστα με όλες τις μετρήσεις που ελήφθησαν αντίστοιχα:



Εικόνα 52: Εικονίδιο γραφήματος της κατανάλωσης συναρτήσει του χρόνου

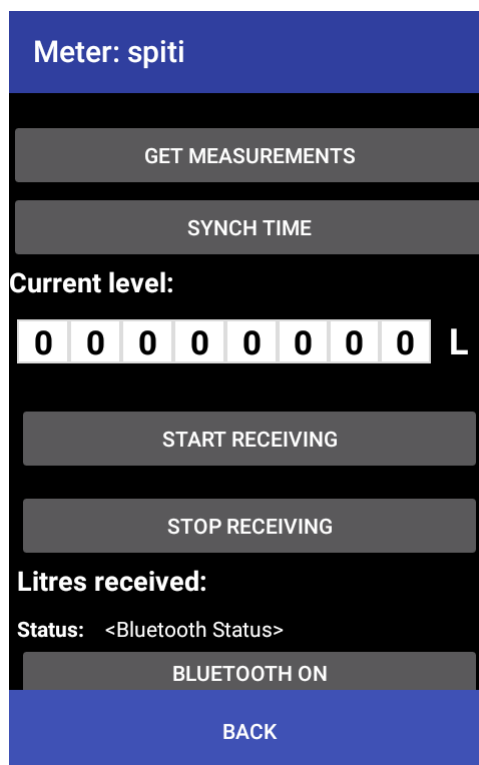


Εικόνα 53: Εικονίδιο που παρουσιάζει σε λίστα το σύνολο των μετρήσεων

Αρχικά δεν έχουμε λάβει κάποια μέτρηση, επομένως δεν υπάρχουν δεδομένα για εμφάνιση στη καρτέλα του μετρητή.

### 7.5.2 Επικοινωνία με το μικροελεγκτή

Για να λάβουμε τις αποθηκευμένες μετρήσεις του μικροελεγκτή, για να συγχρονίσουμε το RTC του μικροελεγκτή με το πραγματικό χρόνο και για να χρησιμοποιήσουμε τη λειτουργία παραλαβής επιλέγουμε απλά τη καρτέλα του μετρητή. Η σελίδα που οδηγούμαστε φαίνεται παρακάτω (εικόνα 54):



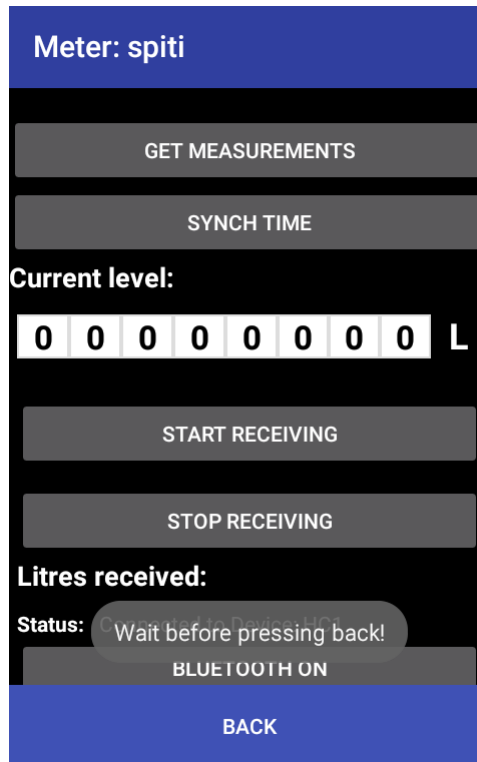
Εικόνα 54: Η σελίδα που οδηγούμαστε από τη καρτέλα του μετρητή

Στο παραπάνω στιγμιότυπο βλέπουμε τις επιλογές που μας δίνονται για την επικοινωνία με τον μικροελεγκτή της κατασκευής.

#### 7.5.2.1 Λήψη μετρήσεων

Με το κουμπί «GET MEASUREMENTS» πραγματοποιείται η λήψη των μετρήσεων της κατανάλωσης που έχει αποθηκεύσει ο μικροελεγκτής. Ο χαρακτήρας «m» στέλνεται μέσω του ενσωματωμένου Bluetooth της συσκευής προκειμένου να ληφθεί από τη σειριακή θύρα του μικροελεγκτή, ο οποίος με τη σειρά του στέλνει τις μετρήσεις στη μορφή που είναι

αναγνωρίσιμη από την εφαρμογή. Κάθε ξεχωριστή μέτρηση της κατανάλωσης λαμβάνεται και αποθηκεύεται απευθείας. Μέχρι να ολοκληρωθεί η αποστολή των μετρήσεων, προβάλλεται στην οθόνη της συσκευής το κατάλληλο μήνυμα για να μην ακυρώσει ο χρήστης την αποστολή επιστρέφοντας στην προηγούμενη οθόνη (εικόνα 55):



Εικόνα 55: Προειδοποίηση κατά τη διάρκεια λήψης των μετρήσεων

### 7.5.2.2 Συγχρονισμός του RTC του μικροελεγκτή

Όπως αναφέραμε σε προηγούμενο κεφάλαιο, για να λειτουργήσει ομαλά το RTC του μικροελεγκτή απαιτείται ο συγχρονισμός του με το πραγματικό χρόνο. Αυτή η διαδικασία πρέπει να γίνει τη πρώτη φορά που θα εκκινήσει ο μικροελεγκτής και πρέπει να γίνεται κάθε φορά που πραγματοποιείται επαναφορά (reset) του μικροελεγκτή σε περίπτωση διακοπής της τροφοδοσίας. Πατώντας το κουμπί «SYNCH TIME» στέλνεται μέσω του Bluetooth ο χαρακτήρας «c», με τον οποίο ο μικροελεγκτής μπαίνει σε αναμονή, περιμένοντας να λάβει τις τρέχον τιμές για τα λεπτά, την ώρα, τη μέρα, το μήνα και το έτος. Έτσι, η Android συσκευή στέλνει με τη σειρά της με μία μικρή καθυστέρηση τις τιμές αυτές, σύμφωνα με το δικό της ρολόι. Μέρος της διαδικασίας που ακολουθεί η συσκευή φαίνεται στο παρακάτω κώδικα:

```
mConnectedThread.write("c");
try
{
    Thread.sleep(500);
}
catch(InterruptedException ex)
```

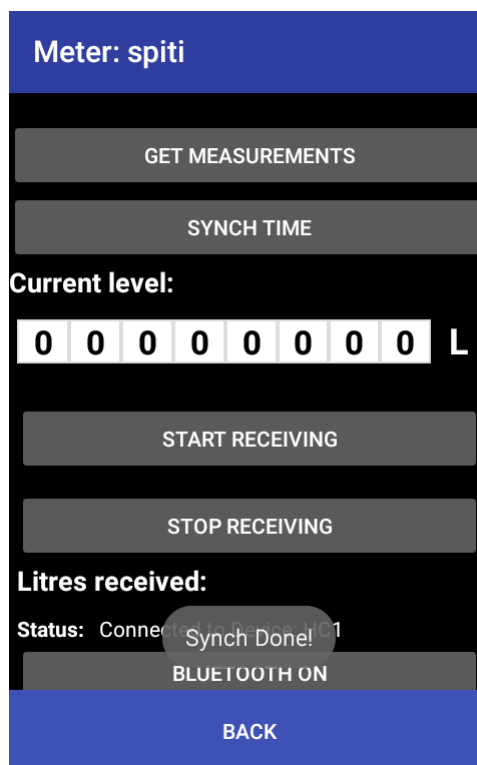
```

{
    Thread.currentThread().interrupt();
}
int minutes = calendar.get(Calendar.MINUTE);
mConnectedThread.write(Integer.toString(minutes));
try
{
    Thread.sleep(500);
}
catch (InterruptedException ex)
{
    Thread.currentThread().interrupt();
}
int hours = calendar.get(Calendar.HOUR_OF_DAY);
mConnectedThread.write(Integer.toString(hours));

```

όπου mConnectedThread είναι το thread που ανοίγεται για την αποστολή και λήψη δεδομένων μέσω του Bluetooth. Η ίδια διαδικασία ακολουθείται για τη μέρα, το μήνα και το έτος.

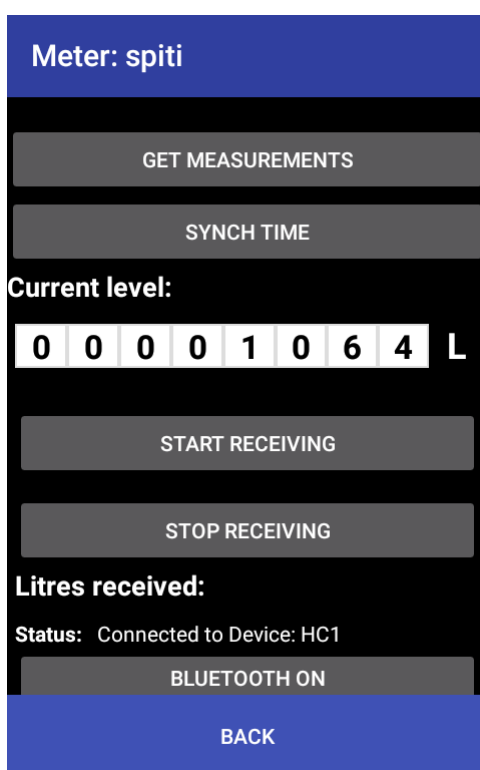
Όταν ο συγχρονισμός έχει ολοκληρωθεί, προβάλλεται το αντίστοιχο μήνυμα στην οθόνη (εικόνα 56):



Εικόνα 56: Επιβεβαίωση του συγχρονισμού

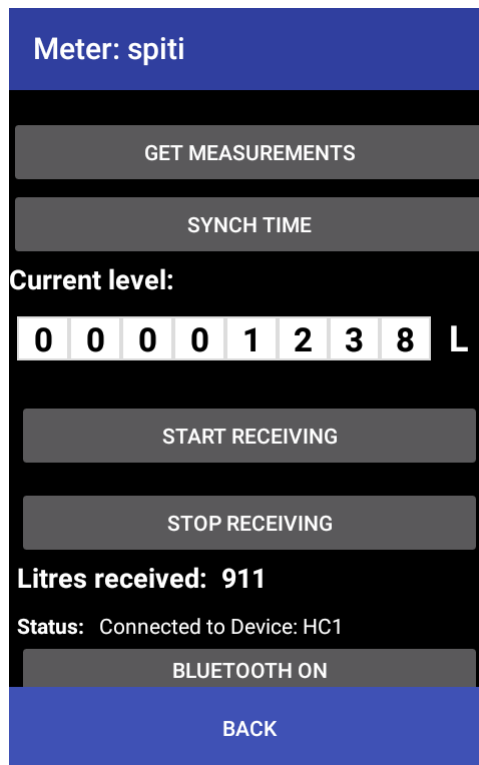
### 7.5.2.3 Παραλαβή

Στη περίπτωση που ο χρήστης επιθυμεί να ξεκινήσει τη διαδικασία της παραλαβής κάποιου υγρού στη δεξαμενή, αυτό που πρέπει να κάνει είναι να πατήσει το κουμπί «START RECEIVING». Η συσκευή τότε στέλνει μέσω Bluetooth στο μικροελεγκτή το χαρακτήρα «r», με τον οποίο ο μικροελεγκτής καταλαβαίνει ότι ο χρήστης επιθυμεί να γνωρίζει συνεχώς τη τιμή του περιεχομένου της δεξαμενής, στέλνοντας με τη σειρά του στη σειριακή θύρα ανά μικρά χρονικά διαστήματα τη τιμή του ύψους στο οποίο βρίσκεται το φλοτέρ. Τα ψηφία τότε στην οθόνη της συσκευής ανανεώνονται και παρουσιάζουν με ακρίβεια τη μέτρηση σε λίτρα του περιεχομένου της δεξαμενής, προκειμένου να γνωρίζει ο χρήστης την ακριβή ποσότητα που βρίσκεται τη δεδομένη στιγμή στη δεξαμενή (εικόνα 57):



Εικόνα 57: Στιγμιότυπο κατά τη διάρκεια της παραλαβής

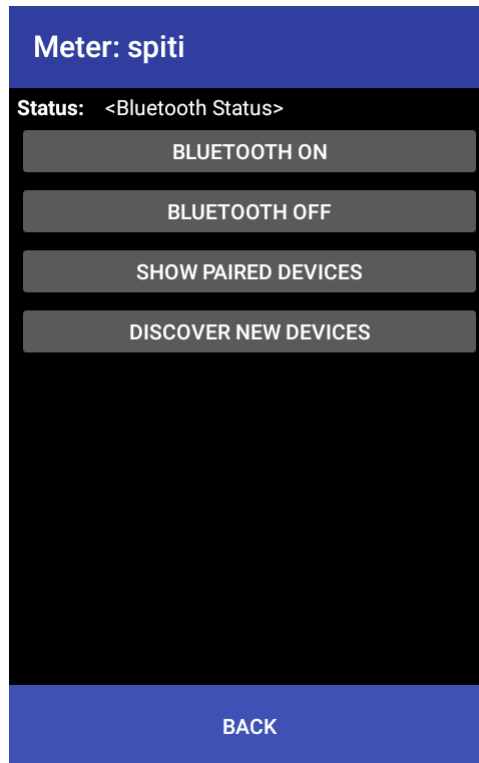
Όταν η παραλαβή έχει ολοκληρωθεί, ο χρήστης πρέπει να πατήσει το κουμπί «STOP RECEIVING», με το οποίο στέλνεται ο χαρακτήρας «s» προκειμένου να ενημερωθεί ο μικροελεγκτής ότι η παραλαβή ολοκληρώθηκε και να σταματήσει τη μετάδοση της μέτρησης. Προβάλλεται τότε στην οθόνη στο πεδίο «Litres received:» η ακριβής ποσότητα σε λίτρα που έλαβε ο χρήστης:



Εικόνα 58: Στιγμιότυπο μετά την ολοκλήρωση της παραλαβής

### 7.5.3 Η επικοινωνία με το Bluetooth

Για να επιτευχθεί η παραπάνω επικοινωνία μεταξύ του μικροελεγκτή και της Android συσκευής, είναι απαραίτητο να έχει αρχικά συνδεθεί η Android συσκευή με το Bluetooth module που είναι συνδεδεμένο με το μικροελεγκτή. Για να συνδεθεί η συσκευή με το Bluetooth module χρησιμοποιούνται τα κουμπιά που βλέπουμε στο παρακάτω στιγμιότυπο:



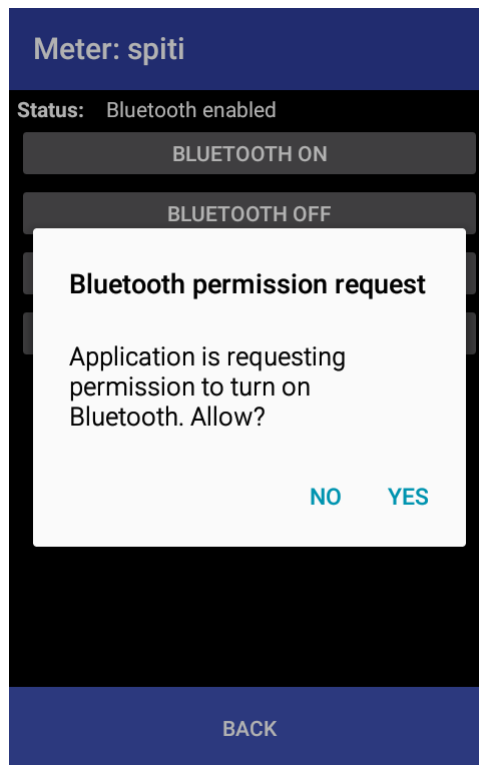
Εικόνα 59: Η συνέχεια της σελίδας

Τα κουμπιά που βλέπουμε παραπάνω παρέχουν τις εξής λειτουργίες:

- Το κουμπί «BLUETOOTH ON» ενεργοποιεί το ενσωματωμένο Bluetooth της Android συσκευής.
- Το κουμπί «BLUETOOTH OFF» απενεργοποιεί το ενσωματωμένο Bluetooth της Android συσκευής.
- Το κουμπί «SHOW PAIRED DEVICES» δείχνει τις συσκευές με τις οποίες έχει συνδεθεί προηγουμένως η συσκευή.
- Το κουμπί «DISCOVER NEW DEVICES» ανακαλύπτει συσκευές με τις οποίες δεν έχει επικοινωνήσει προηγουμένως η συσκευή.

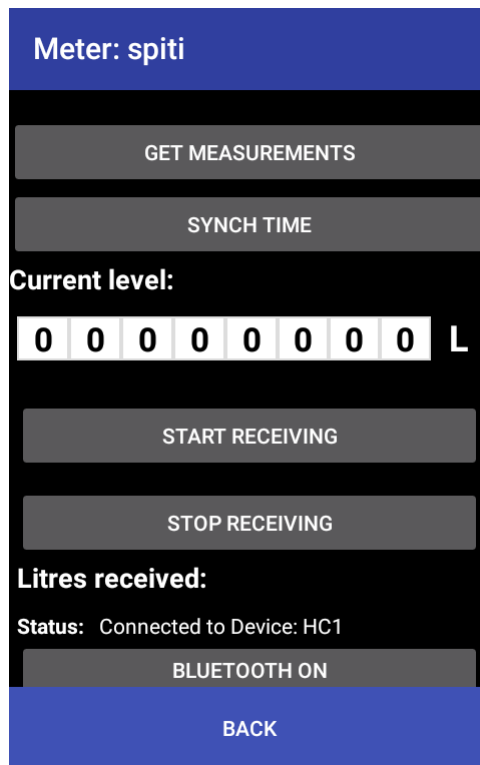
Αρχικά ο χρήστης πρέπει να ενεργοποιήσει το ενσωματωμένο Bluetooth. Αφού συμφωνήσει στην επιλογή για την ενεργοποίηση του Bluetooth, μπορεί να συνδεθεί με τη Bluetooth συσκευή της επιλογής του.





Εικόνα 60: Παράθυρο διαλόγου για την επιβεβαίωση της ενεργοποίησης του ενσωματωμένου Bluetooth

Το Bluetooth module της κατασκευής θα βρίσκεται είτε στη λίστα που εμφανίζεται με το κουμπί «SHOW PAIRED DEVICES», είτε στη λίστα που εμφανίζεται με το κουμπί «DISCOVER NEW DEVICES», ανάλογα με το εάν έχει ξανασυνδεθεί ή όχι προηγουμένως ο χρήστης με το συγκεκριμένο module. Διαλέγοντας από τη λίστα το Bluetooth module, η Android συσκευή συνδέεται με τη κατασκευή και ο χρήστης μπορεί να επικοινωνήσει με το μικροελεγκτή για να χρησιμοποιήσει τις λειτουργίες που αναφέρθηκαν προηγουμένως.



Εικόνα 61: Επιβεβαίωση της σύνδεσης με το Bluetooth module

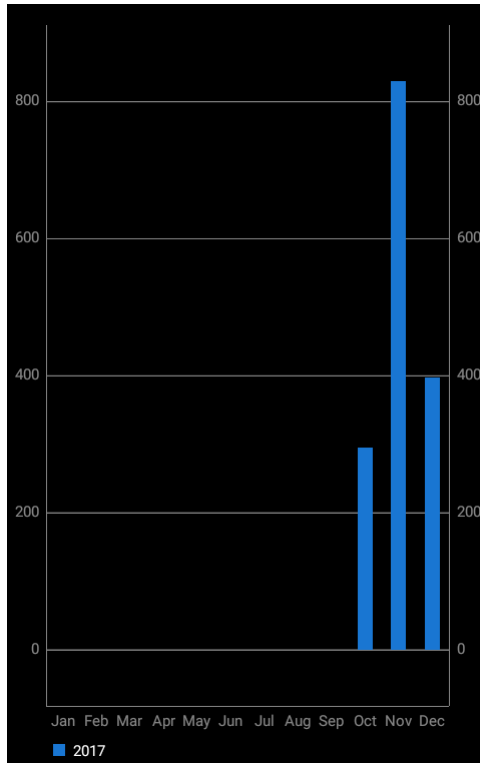
#### 7.5.4 Παρουσίαση των μετρήσεων της κατανάλωσης

Αφού έχουμε ολοκληρώσει τη λήψη των μετρήσεων της κατανάλωσης, η καρτέλα με τον μετρητή παίρνει τη παρακάτω μορφή (εικόνα 62):



Εικόνα 62: Η ανανεωμένη καρτέλα του μετρητή

Όπως βλέπουμε, οι πληροφορίες για την ημερομηνία της τελευταίας μέτρησης (Last Reading) και τη μέτρηση της κατανάλωσης αυτής της μέρας (Consumption) έχουν συμπληρωθεί. Μπορούμε πλέον είτε να εμφανίσουμε όλες τις μετρήσεις σε μορφή λίστας, είτε να δούμε το συνολικό γράφημα της κατανάλωσης σε λίτρα συναρτήσει του χρόνου (οριζόντιος άξονας οι μήνες του έτους, κάθετος άξονας η μηνιαία). Πατώντας το κουμπί του γραφήματος (εικόνα 52) μας εμφανίζεται το παρακάτω γράφημα, με ένα δείγμα μετρήσεων:



Εικόνα 63: Το γράφημα με τις μηνιαίες καταναλώσεις

Μπορούμε έτσι να δούμε μια συνοπτική εικόνα της κατανάλωσης κάθε μήνα του έτους. Για να δούμε το σύνολο των μετρήσεων, μπορούμε να τις εμφανίσουμε σε λίστα πατώντας το αντίστοιχο κουμπί (εικόνα 53). Το σύνολο των μετρήσεων παρουσιάζεται ως εξής:

Meter: spiti	Daily	Monthly
Dec 20, 2017	49 L	397 L
Dec 19, 2017	37 L	348 L
Dec 18, 2017	37 L	311 L
Dec 17, 2017	0 L	274 L
Dec 16, 2017	0 L	274 L
Dec 15, 2017	0 L	274 L
Dec 14, 2017	0 L	274 L
Dec 13, 2017	0 L	274 L
Dec 12, 2017	0 L	274 L
Dec 11, 2017	0 L	274 L
Dec 10, 2017	0 L	274 L
Dec 9, 2017	0 L	274 L
Dec 8, 2017	0 L	274 L
Dec 7, 2017	0 L	274 L
Dec 6, 2017	38 L	274 L
Dec 5, 2017	28 L	236 L
Dec 4, 2017	46 L	208 L
Dec 3, 2017	65 L	162 L
Dec 2, 2017	78 L	97 L
Dec 1, 2017	19 L	19 L
Nov 30, 2017	15 L	829 L
Nov 29, 2017	33 L	814 L
Nov 28, 2017	36 L	781 L

Εικόνα 64: Η λίστα με το σύνολο των μετρήσεων

Εμφανίζονται έτσι όλες οι μετρήσεις της κατανάλωσης που έχουμε λάβει μέχρι στιγμής. Η 1<sup>η</sup> στήλη είναι η ημερομηνία της κάθε μέτρησης, 2<sup>η</sup> στήλη είναι η τιμή της μέτρησης αυτής και στη 3<sup>η</sup> στήλη εμφανίζεται το σύνολο της κατανάλωσης του μήνα, το οποίο μηδενίζεται στις αρχές κάθε μήνα.



## 8 Επίλογος

Στη διπλωματική αυτή εργασία είδαμε πως υλοποιήθηκε το πρωτότυπο όργανο μέτρησης της στάθμης σε μία δεξαμενή, το οποίο σε συνδυασμό με το μικροελεγκτή και την android εφαρμογή συνθέτουν μία ολοκληρωμένη και σύγχρονη λύση για τη παρακολούθηση της στάθμης της δεξαμενής.

Το προϊόν αυτό δημιουργήθηκε με κύριο σκοπό τη χρησιμοποίησή του από τον μέσο καταναλωτή για την καταγραφή της προσωπικής κατανάλωσης πετρελαίου, καθώς και για την μέτρηση με ακρίβεια της ποσότητας του πετρελαίου που προστίθεται στη δεξαμενή κατά τη διάρκεια της παραλαβής. Η android εφαρμογή δημιουργήθηκε με στόχο να είναι εύκολη στη χρήση και να προσφέρει όλες τις απαραίτητες λειτουργίες στο μέσο χρήστη. Με κάποιες όμως τροποποιήσεις δεν αποκλείεται η χρησιμοποίησή του προϊόντος και σε άλλες εφαρμογές, είτε βιομηχανικές, είτε καταναλωτικές.

### 8.1 Κόστος

Το συνολικό κόστος της κατασκευής ανήλθε στα 40€. Πιο συγκεκριμένα:

- Ηλεκτρολογικός σωλήνας: 1€
- Μαγνητικό φλοτέρ: 6€
- Σύνολο επαφών reed: 20€
- Σύνολο αντιστάσεων: 1€
- Μικροελεγκτής C8051F340: 9€
- Bluetooth module HC-05: 3€

Το κόστος θα μπορούσε επιπλέον να μειωθεί αν χρησιμοποιούνταν διαφορετικός, πιο φθηνός μικροελεγκτής από το C8051F340, ο οποίος επιλέχτηκε για λόγους ευκολίας. Συγκρίνοντας το κόστος της κατασκευής με τις έτοιμες λύσεις που είδαμε στο 3<sup>ο</sup> κεφάλαιο, βλέπουμε ότι το ολοκληρωμένο προϊόν μπορεί να γίνει ένα ανταγωνιστικό προϊόν με σημαντικά περιθώρια κέρδους.

### 8.2 Παρατηρήσεις

Εδώ θα πρέπει να επισημανθούν κάποιες δυσκολίες που προέκυψαν κατά τη διάρκεια της κατασκευής του αισθητήρα. Οι επαφές reed είναι αρκετά εύθραυστες, με αποτέλεσμα ένας μικρός αριθμός επαφών να σπάσουν κατά τη τοποθέτηση τους στη πλακέτα. Ένα πρόβλημα που όμως λύνεται δίνοντας απλά παραπάνω προσοχή στη τοποθέτηση αυτών των επαφών. Επίσης, οι πλακέτες πάνω στις οποίες υλοποιήθηκε το κύκλωμα ήταν γενικού σκοπού, με αποτέλεσμα να γίνει χρονοβόρα η διαδικασία της κατασκευής.

Μία επίσης χρονοβόρα διαδικασία που προέκυψε ήταν αυτή της βαθμονόμησης. Είδαμε ότι τελικά η διαδικασία αυτή έγινε χειροκίνητα, κάτι που σημαίνει πως μπορεί να υπήρξαν και σφάλματα στη μέτρηση του ύψους. Στο μέλλον, αυτή η διαδικασία θα

μπορούσε να γίνει αυτόματα για την περαιτέρω απλούστευση της διαδικασίας της παραγωγής.

### 8.3 Μελλοντικές επεκτάσεις

Κάποιες μελλοντικές επεκτάσεις που μπορεί να υλοποιηθούν χάρη στις δυνατότητες αυτοματισμού και διασύνδεσης που προσφέρονται από το όργανο μέτρησης που δημιουργήθηκε, φαίνονται παρακάτω:

- Χρησιμοποιώντας κάποιο wifi module, όπως για παράδειγμα το module ESP8266, τα δεδομένα που συλλέγει ο μικροελεγκτής θα μπορούσαν να προωθούνται στο διαδίκτυο προκειμένου η παρακολούθηση της κατανάλωσης να είναι δυνατή όχι μόνο μέσω της android συσκευής χρησιμοποιώντας τη bluetooth επικοινωνία αλλά και μέσω του διαδικτύου από αντίστοιχη σελίδα.
- Ενσωματώνοντας ένα GSM module στη κατασκευή, θα μπορούσε να προσφέρεται στον χρήστη η δυνατότητα να λαμβάνει άμεσες ειδοποιήσεις μέσω SMS στο κινητό του τηλέφωνο σε περίπτωση χαμηλής στάθμης στη δεξαμενή ή σε περίπτωση που η κατανάλωση ξεπεράσει κάποια συγκεκριμένη ποσότητα.
- Αξιοποιώντας κάποιου είδους ρελέ με δυνατότητες ασύρματης επικοινωνίας, θα μπορούσε να υπάρξει η δυνατότητα να διακόπτεται η κατανάλωση, σε περίπτωση που αυτή ξεπεράσει μία συγκεκριμένη ποσότητα κατά της διάρκεια ενός χρονικού διαστήματος, είτε αυτόματα είτε με επιλογή του χρήστη.



## Βιβλιογραφία

- 1) Feng Xia, Laurence T. Yang, Lizhe Wang and Alexey Vinel. 2012. Internet of Things
- 2) Ron Davies. 2015. The Internet of Things, Opportunities and challenges
- 3) The Internet of Things:What Is It and Why Should Internal Audit Care?:  
[https://www.protiviti.com/sites/default/files/united\\_states/insights/internal-audit-and-the-internet-of-things-whitepaper-protiviti.pdf](https://www.protiviti.com/sites/default/files/united_states/insights/internal-audit-and-the-internet-of-things-whitepaper-protiviti.pdf)
- 4) Simon Paonessa. LEVEL MEASUREMENT TECHNOLOGIES
- 5) Reed Switch Application Notes-HSI Sensing: [http://www.hsisensing.com/wp-content/uploads/2016/03/HSI\\_Reed\\_Switch\\_Application\\_Notes\\_v12\\_2013.pdf](http://www.hsisensing.com/wp-content/uploads/2016/03/HSI_Reed_Switch_Application_Notes_v12_2013.pdf)
- 6) PIC-Reed Switches-FAQ: <http://www.pic-gmbh.com/en/tech-center/reed-switches/faq/>
- 7) Craig Steiner. 8051 Tutorial: <http://www.8052.com/tutintro.phtml>
- 8) Tarun Agarwal. Infographics: A Brief About 8051 Microcontroller:  
<https://www.elprocus.com/brief-about-8051-microcontroller/>
- 9) Keil-Getting Started: <https://www.keil.com/product/brochures/uv4.pdf>
- 10) John Padgette, John Bahr, Mayank Batra, Marcel Holtmann, Rhonda Smithbey, Lily Chen, Karen Scarfone. May 2017. Guide to Bluetooth Security
- 11) Helen Fornazier, Aurélien Martin, Scott Messner. March 2012. Wireless Communication : Wi-Fi, Bluetooth, IEEE 802.15.4, DASH7
- 12) Android, the world's most popular mobile platform:  
<https://developer.android.com/about/android.html>
- 13) Aaditya Jain, Samridha Raj, Dr. Bala Buksh. July 2016. A Comparative Study of Mobile Operating Systems with Special Emphasis on Android OS
- 14) Meet Android Studio: <https://developer.android.com/studio/intro/index.html>
- 15) SQLite Tutorial: <https://www.tutorialspoint.com/sqlite/>