



Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

Πολυτεχνική Σχολή

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

**Υλοποίηση συστήματος μετρήσεων πεδίου σε  
ηλεκτρομαγνήτη, μεταφορά και ενσωμάτωση των  
αποτελεσμάτων σε αρχείο για τη ανάγνωσή τους από το  
λογισμικό MultiVu**

Διπλωματική Εργασία

**Γεώργιος Βουδούρης**

**Κωνσταντίνος Παρασκευαΐδης**

Επιβλέπων καθηγητής:

**Γεώργιος Λιτσαρδάκης**

**Θεσσαλονίκη, Νοέμβριος 2020**



## Ευχαριστίες

Με την ολοκλήρωση της διπλωματικής εργασίας και γενικότερα των σπουδών μου θα ήθελα να ευχαριστήσω ορισμένους ανθρώπους οι οποίοι συνέβαλλαν ο καθένας με τον τρόπο του στη μέχρι τώρα πορεία μου.

Αρχικά εκφράζω τις θερμές μου ευχαριστίες στον επιβλέποντα καθηγητή κ. Γεώργιο Λιτσαρδάκη για την άψογη συνεργασία που είχαμε στα πλαίσια της παρούσας μελέτης αλλά και για τις γνώσεις που αποκόμισα από αυτόν τα περασμένα χρόνια.

Τέλος θα ήθελα να ευχαριστήσω την οικογένεια μου και τους φίλους μου για τη στήριξη που μου παρείχαν κατά τη διάρκεια αυτού του ταξιδιού.

-Γεώργιος Βουδούρης

Αρχικά θα ήθελα να εκφράσω την ευγνωμοσύνη και τις ευχαριστίες μου στον καθηγητή κ. Γεώργιο Λιτσαρδάκη για τη συνεργασία μας καθώς και για την εμπιστοσύνη που μου έδειξε για την περάτωση της παρούσας εργασίας.

Θα ήθελα να ευχαριστήσω του γονείς μου, Φώτιο και Μαριάννα και την αδερφή μου Πολυξένη για την υποστήριξη και τη βοήθεια που μου παρείχαν όλα αυτά τα χρόνια των σπουδών μου και ήταν πάντα εκεί για να μου δώσουν δύναμη και κουράγιο να ξεπεράσω όλες τις δύσκολες στιγμές. Χωρίς την βοήθειά τους η ολοκλήρωση των σπουδών μου θα ήταν αδύνατη.

Τέλος, θα ήθελα να ευχαριστήσω όλους μου τους φίλους μου για όλες τις όμορφες στιγμές που περάσαμε και την βοήθεια που μου προσέφεραν.

-Κωνσταντίνος Παρασκευαΐδης



## Πρόλογος

Τα μαγνητικά υλικά έχουν ένα πολύ μεγάλο φάσμα εφαρμογών στην ιατρική, στη βιομηχανία στην έρευνα αλλά και στην καθημερινή ζωή. Συνεπώς η κατανόησή τους και ο χαρακτηρισμός των ιδιοτήτων τους αποτελεί ένα σημαντικό κομμάτι των εργαστηριακών μελετών.

Για την ανίχνευση της συμπεριφοράς των μαγνητικών υλικών διεξάγονται μετρήσεις με την είσοδο ομογενούς ηλεκτρομαγνητικού πεδίου. Το Versalab της Quantum Design είναι ένα τέτοιο σύστημα μετρήσεων το οποίο χρησιμοποιείται στο εργαστήριο Ηλεκτροτεχνικών Υλικών του ΑΠΘ. Τα μειονεκτήματα που έχει το συγκεκριμένο σύστημα ως προς το κόστος λειτουργίας του και τον απαιτούμενο χρόνο πραγματοποίησης των μετρήσεων ευνοεί την αναζήτηση εναλλακτικών μέσων τα οποία αποτελούν μια πιο οικονομική και πρακτική λύση.

Σκοπός αυτής της διπλωματικής είναι η δημιουργία μιας διάταξης μετρητή με τον οποίο θα γίνει δυνατή η μέτρηση του πεδίου που προκύπτει από ηλεκτρομαγνήτη ο οποίος δε διαθέτει αισθητήρες που πραγματοποιούν αυτή τη διαδικασία. Έτσι θα μπορέσει να υπάρξει μια εικόνα για το πεδίο καθώς επίσης θα καταστεί δυνατή η παραγωγή του επιθυμητού βρόχου υστέρησης. Για την υλοποίηση της παρούσας μελέτης θα χρησιμοποιηθεί ένα μαγνητόμετρο δονούμενου δείγματος, ένας μετρητής πυκνότητας μαγνητικής ροής και ένα Arduino για την μέτρηση της αναλογικής τάσης που παράγεται στην έξοδο του οργάνου μέτρησης.



## Abstract

Magnetic materials have a wide variety of applications in medicine, research, industries as well as in everyday life. Therefore, characterizing their properties is an important part of laboratory studies.

In order to detect behavior of magnetic materials when homogenous electromagnetic field is assigned proper measurements must take place. Quantum Design's Versalab is such a measurement system which is widely used in the laboratory of Materials for Electrotechnics at Aristotle University of Thessaloniki. The disadvantages of this system in terms of operating costs and the time required to carry out the measurements favor the search for alternative means which constitute an economical and practical solution.

The aim of this diploma thesis is to create a measurement system which will be used in order to achieve measurement of electromagnetic field on electromagnet that does not include sensors for this purpose. In this way it is possible for a measurement of electromagnetic field to be taken and for the desired hysteresis loop to be produced. For the implementation of this study, a variety of equipment will be used, such as a vibrating sample magnetometer, a gaussmeter and an Arduino which will measure analog voltage produced at gaussmeter's output.





## Περιεχόμενα

|  |    |
|--|----|
| Ευχαριστίες .....  | 2  |
| Πρόλογος.....  | 4  |
| Abstract .....   | 6  |
| Περιεχόμενα.....   | 8  |
| 1. Εισαγωγή.....   | 10 |
| 1.1. Σκοπός διπλωματικής .....   | 10 |
| 1.2. Διάρθρωση εγγράφου .....  | 11 |
| 2. Vsm – MultiVu - Ηλεκτρομαγνήτης.....                                | 13 |
| 2.1. Vibrating Sample Magnetometer (VSM).....                          | 13 |
| 2.2. MultiVu.....  | 16 |
| 2.3. Ηλεκτρομαγνήτης.....  | 17 |
| 3. Θεωρία .....  | 19 |
| 3.1. Βασικές Παράμετροι Μετρήσεων.....                                 | 19 |
| 3.2. Κατηγορίες Υλικών .....   | 21 |
| 3.3. Μετατροπή Αναλογικού Σήματος Σε Ψηφιακό .....                     | 23 |
| 3.3.1. Ορισμοί.....  | 23 |
| 3.3.2. Analog to Digital Converter .....                               | 25 |
| 3.3.3. Μετατροπέας A/D διαδοχικών προσεγγίσεων.....                    | 26 |
| 4. Συστήματα Μετρητικής Διάταξης.....                                  | 28 |
| 4.1. Arduino.....  | 28 |
| 4.1.1. Τεχνικά Χαρακτηριστικά .....                                    | 28 |
| 4.1.2. Λογισμικό .....   | 28 |
| 4.2. ATmega328p .....  | 29 |
| 4.2.1. Τεχνικά χαρακτηριστικά.....                                     | 29 |
| 4.2.2. Analog to Digital Converter (ADC).....                          | 31 |
| 4.2.3. Σφάλματα Μετατροπών.....  | 33 |
| 4.3. FW Bell 5180 Gaussmeter .....                                     | 36 |
| 4.3.1. Μέτρηση Πυκνότητας Μαγνητικής Ροής (Magnetic Flux Density)..... | 36 |
| 4.3.2. Σύνομη Περιγραφή.....   | 38 |
| 4.3.3. Επεξήγηση Κώδικα C++ .....                                      | 39 |
| 4.4. Προγραμματισμός Arduino – Προετοιμασία Μέτρησης .....             | 42 |

|        |   |    |
|--------|---|----|
| 4.4.1. | Διαιρέτης τάσης .....                         | 42 |
| 4.4.2. | Oversampling and Decimation .....             | 43 |
| 4.4.3. | Βαθμονόμηση του Arduino .....                 | 47 |
| 4.4.4. | Επεξήγηση Κώδικα Arduino .....                | 48 |
| 5.     | Προγραμματισμός και Διεξαγωγή Μετρήσεων ..... | 51 |
| 5.1.   | Δημιουργία μέτρησης στο MultiVu .....         | 51 |
| 5.2.   | Εκτελέσιμο αρχείο in_vsm.exe .....            | 55 |
| 5.2.1. | Περιγραφή.....                                | 55 |
| 5.2.2. | Δημιουργία εκτελέσιμου αρχείου .....          | 55 |
| 5.3.   | Κώδικας .....                                 | 56 |
| 5.4.   | Μέτρηση.....                                  | 61 |
| 6.     | Συμπεράσματα – Επεκτάσεις.....                | 65 |
| 6.1.   | Σύνοψη – Συμπεράσματα .....                   | 65 |
| 6.2.   | Μελλοντικές επεκτάσεις .....                  | 66 |
|        | Αναφορές .....                                | 67 |
|        | Ευρετήριο εικόνων.....                        | 69 |
|        | Παράρτημα.....                                | 71 |

## 1. Εισαγωγή

Βασικό αντικείμενο του εργαστηρίου Ηλεκτροτεχνικών Υλικών της Πολυτεχνικής Σχολής του ΑΠΘ είναι η διεξαγωγή μετρήσεων προκειμένου να καθοριστούν οι ιδιότητες κάποιων μαγνητικών υλικών. Αυτή η διαδικασία πραγματοποιείται κατά βάση με τη χρήση ειδικού εξοπλισμού και συγκεκριμένα της συσκευής PPMS Versalab (Physical Property Measurement System) και μιας κεφαλής που περιλαμβάνει ένα VSM (Vibrating Sample Magnetometer) ή Μαγνητόμετρο Δονούμενου δείγματος. Το Versalab είναι μία φορητή πλατφόρμα χαρακτηρισμού υλικών, η οποία μπορεί να λειτουργήσει σε ένα εύρος θερμοκρασίας 50 – 400 Kelvin. Ένα VSM περιλαμβάνει ένα μαγνητικό δείγμα το οποίο δονείται κάθετα σε ένα ομογενές μαγνητικό πεδίο σε μικρό πλάτος και δύο πηνία λαμβάνουν την επαγόμενη ηλεκτρομαγνητική δύναμη που οφείλεται στην μαγνητική ροπή που δημιουργείται από αυτή τη δόνηση του δείγματος [1].

Για την καταγραφή των μετρήσεων στον υπολογιστή καθώς και για τη γραφική αναπαράσταση των αποτελεσμάτων χρησιμοποιείται η εφαρμογή της κατασκευάστριας εταιρείας του VSM και του Versalab. Αυτή ονομάζεται MultiVu και αποτελεί ουσιαστικά το κέντρο ελέγχου όλων των μηχανημάτων της εταιρείας.

Ενώ αυτή η διάταξη του Versalab με το VSM αποτελεί μία πολύ αξιόπιστη λύση για την ανίχνευση των ιδιοτήτων των μαγνητικών υλικών, υπάρχουν και ορισμένα προβλήματα που δυσχεραίνουν την διεξαγωγή μετρήσεων. Αρχικά όλες οι μετρήσεις πραγματοποιούνται κάτω από ένα επιθυμητό εύρος θερμοκρασίας. Στο Versalab έχει υπεραγωγίμο μαγνήτη που πρέπει να ψυχθεί σε θερμοκρασία 4 Kelvin προκειμένου να μπορέσει να πραγματοποιηθεί μια μέτρηση. Αυτό έχει ως αποτέλεσμα την αύξηση του χρόνου που απαιτείται από τη στιγμή που θα εκκινήσει το μηχάνημα μέχρι τη στιγμή που θα είναι έτοιμο για τη διεξαγωγή μετρήσεων που σε ορισμένες περιπτώσεις μπορεί να φτάσει και τις 24 ώρες. Αυτό συνεπάγεται με αυξημένες απαιτήσεις σε χρήματα για τη συντήρηση και την προμήθεια ψυκτικού μέσου άρα και αυξημένο συνολικό κόστος λειτουργίας.

Έτσι προκειμένου να εξαλειφθούν αυτά τα προβλήματα παρουσιάστηκε η ευκαιρία να χρησιμοποιηθεί ένας εξωτερικός ηλεκτρομαγνήτης προκειμένου να λάβουμε το ομογενές ηλεκτρομαγνητικό πεδίο που απαιτείται για την πραγματοποίηση των μετρήσεων σε συνδυασμό με τη χρήση του VSM.

### 1.1. Σκοπός διπλωματικής

Σε προηγούμενα χρόνια έγιναν οι κατάλληλες μετατροπές και ήταν δυνατή η διεξαγωγή των μετρήσεων με τον εξωτερικό ηλεκτρομαγνήτη του εργαστηρίου σε συνδυασμό με το VSM. Το ζήτημα που προέκυψε ήταν ότι ενώ στο MultiVu απεικονιζόταν η μέτρηση της μαγνητικής ροπής από τους αισθητήρες του VSM

δεν υπήρχε η εικόνα για το ηλεκτρομαγνητικό πεδίο λόγω του ότι οι αισθητήρες για τη μέτρηση του τελευταίου βρίσκονται εσωτερικά στο Versalab.

Σκοπός αυτής της διπλωματικής είναι η υλοποίηση μιας διάταξης μετρητή με τον οποίο θα καταστεί δυνατή η λήψη της τιμής του παραγόμενου ηλεκτρομαγνητικού πεδίου και η ενσωμάτωσή της στο αρχείο καταγραφής που εξάγεται από το πρόγραμμα. Με αυτόν τον τρόπο θα μπορεί ο χρήστης του προγράμματος να έχει μια γραφική απεικόνιση αυτών των μετρήσεων και να αξιολογήσει ευκολότερα τα παραγόμενα αποτελέσματα.

Για τη μέτρηση του ηλεκτρομαγνητικού πεδίου θα χρησιμοποιηθεί το γκαουσόμετρο 5180 της FW Bell. Στα πλαίσια της παρούσας διπλωματικής θα αναπτυχθούν δύο υλοποιήσεις για τη μέτρηση και καταγραφή των αποτελεσμάτων του ηλεκτρομαγνητικού πεδίου. Στην πρώτη θα επιχειρήσουμε απευθείας σειριακή επικοινωνία του οργάνου μέτρησης με τον υπολογιστή. Αυτό θα επιτευχθεί με χρήση καλωδίου usb και ανάπτυξη προγράμματος σε C++ χρησιμοποιώντας τις έτοιμες βιβλιοθήκες της εταιρείας για την επικοινωνία με το γκαουσόμετρο οι οποίες παρέχονται με το cd εγκατάστασης. Για την καταγραφή των τιμών που λαμβάνει το όργανο στο αρχείο επίσης θα αναπτυχθεί κώδικας σε C++ ο οποίος θα ενσωματωθεί στο κυρίως πρόγραμμα.

Για τη δεύτερη υλοποίηση θα χρησιμοποιηθεί η αναλογική έξοδος του 5180 η οποία έχει εύρος από -5 έως 5 Volt. Για την ανάγνωσή της θα ενταχθεί στη διάταξη ένα Arduino το οποίο με τη χρήση του adc (Analog to Digital Converter) που διαθέτει θα μετατρέψει αυτή την αναλογική είσοδο σε ψηφιακή προκειμένου να διαβαστεί από τον υπολογιστή επίσης μέσω σειριακής επικοινωνίας. Η συγγραφή του κώδικα για τη ρύθμιση του Arduino προκειμένου να διαβάσει αυτή την τάση θα γίνει στη γλώσσα του Arduino IDE, ενώ για τη σειριακή επικοινωνία με αυτό και την αντιγραφή των αποτελεσμάτων στο αρχείο καταγραφής του MultiVu θα χρησιμοποιηθεί η Python.

## 1.2. Διάρθρωση εγγράφου

Στο **2<sup>ο</sup> κεφάλαιο** γίνεται περιγραφή του VSM για να γίνει κατανοητός ο τρόπος λειτουργίας του καθώς και του ηλεκτρομαγνήτη που θα χρησιμοποιηθεί για την παραγωγή του ηλεκτρομαγνητικού πεδίου. Επίσης γίνεται μια σύντομη αναφορά στην εφαρμογή διαχείρισης των μετρήσεων MultiVu.

Στο **3<sup>ο</sup> κεφάλαιο** επεξηγούνται κάποιες θεωρητικές έννοιες οι οποίες είναι απαραίτητες για την κατανόηση της διπλωματικής.

Στο **4<sup>ο</sup> κεφάλαιο** περιγράφεται το Arduino και το περιβάλλον στο οποίο συντάσσεται ο κώδικας του προγράμματος, αναλύεται το κύκλωμα του adc που διαθέτει, γίνεται μία περιγραφή του 5180 και επεξηγείται η διαδικασία της

σειριακής επικοινωνίας με αυτό. Τέλος γίνεται αναφορά στη μέθοδο oversampling που χρησιμοποιείται για να διαβάσουμε την τάση στην είσοδο αυτού και εξηγείται η χρήση διαιρέτη τάσης για το διάβασμα αυτής, καθώς επίσης εξηγείται ο κώδικας που αναπτύχθηκε για τον προγραμματισμό του Arduino.

Στο **5<sup>ο</sup> κεφάλαιο** γίνεται περιγραφή του προγραμματισμού των μετρήσεων με τη χρήση του λογισμικού MultiVu, εξηγείται η λειτουργία του εκτελέσιμου αρχείου που είναι υπεύθυνο για τη λήψη της μέτρησης του πεδίου και παρουσιάζονται μερικές μετρήσεις που διεξήχθησαν με τη νέα διάταξη.

Στο **6<sup>ο</sup> κεφάλαιο** καταγράφονται τα συμπεράσματα που προέκυψαν κατά τη διάρκεια της μελέτης και περιγράφονται ορισμένες μελλοντικές επεκτάσεις.

## 2. Vsm – MultiVu - Ηλεκτρομαγνήτης

### 2.1. Vibrating Sample Magnetometer (VSM)

Το VSM είναι ουσιαστικά ένα γρήγορο και ευαίσθητο μαγνητόμετρο δονούμενου δείγματος. Στο πλαίσιο της διπλωματικής εργασίας εργασίας χρησιμοποιήθηκε το VSM από την Quantum Design [2] .

Η βασική μέτρηση επιτυγχάνεται ταλαντώνοντας ένα δείγμα κοντά σε ένα πηνίο ανίχνευσης (pickup coil) ενώ ταυτόχρονα διαβάζεται η τάση επαγωγής. Χρησιμοποιώντας μία διάταξη κλισιμέτρου με πηνίο ανίχνευσης, ένα σχετικά μεγάλο πλάτος ταλάντωσης (1-3 mm peak) και μια συχνότητα 40 Hz, το σύστημα μπορεί να ανιχνεύσει αλλαγές στη μαγνήτιση μικρότερες από  $10^{-6}$  emu σε συχνότητα δεδομένων 1 Hz.

Η βασική αρχή λειτουργίας για ένα VSM είναι ότι η μεταβολή της μαγνητικής ροής επάγει τάση σε ένα πηνίο ανίχνευσης. Η χρονικά εξαρτημένη επαγωγική τάση δίνεται από την εξίσωση (2.1).

$$V_{coil} = \frac{d\Phi}{dt} = \left(\frac{d\Phi}{dz}\right)\left(\frac{dz}{dt}\right) \quad (2.1)$$

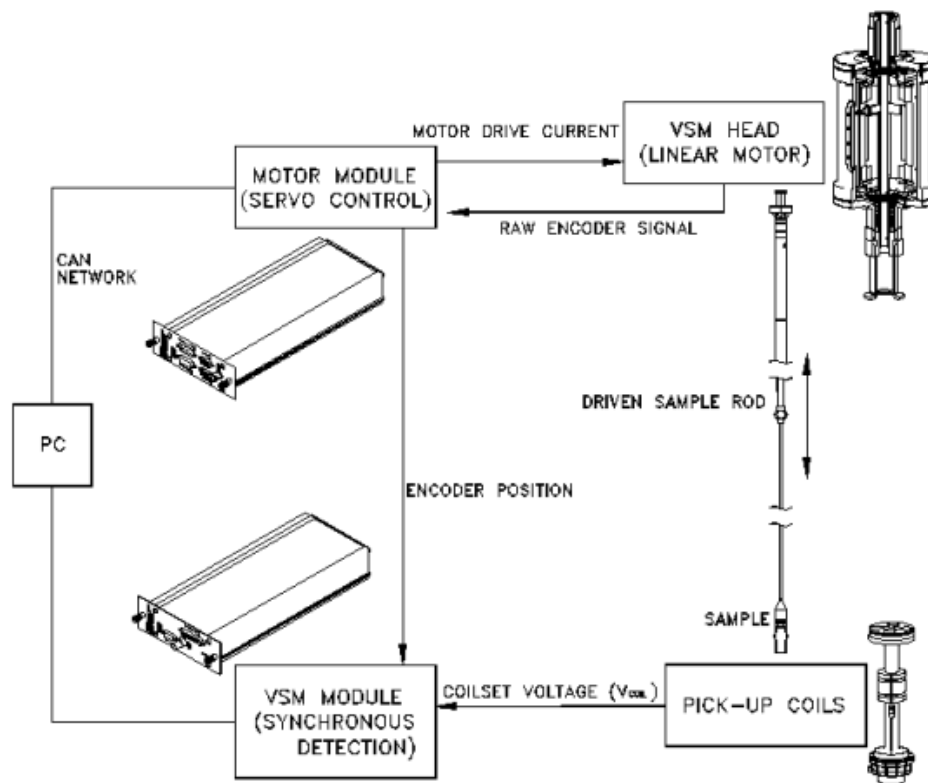
Στην εξίσωση (2.1),  $\Phi$  είναι η μαγνητική ροή που λαμβάνεται από το πηνίο ανίχνευσης,  $z$  είναι η κάθετη θέση του δείγματος σε σχέση με το πηνίο και  $t$  ο χρόνος.

Για ημιτονοειδή ταλάντωση του δείγματος, η τάση στο πηνίο βασίζεται στην εξίσωση (2.2).

$$V_{coil} = 2\pi f C m A \sin(2\pi f t) \quad (2.2)$$

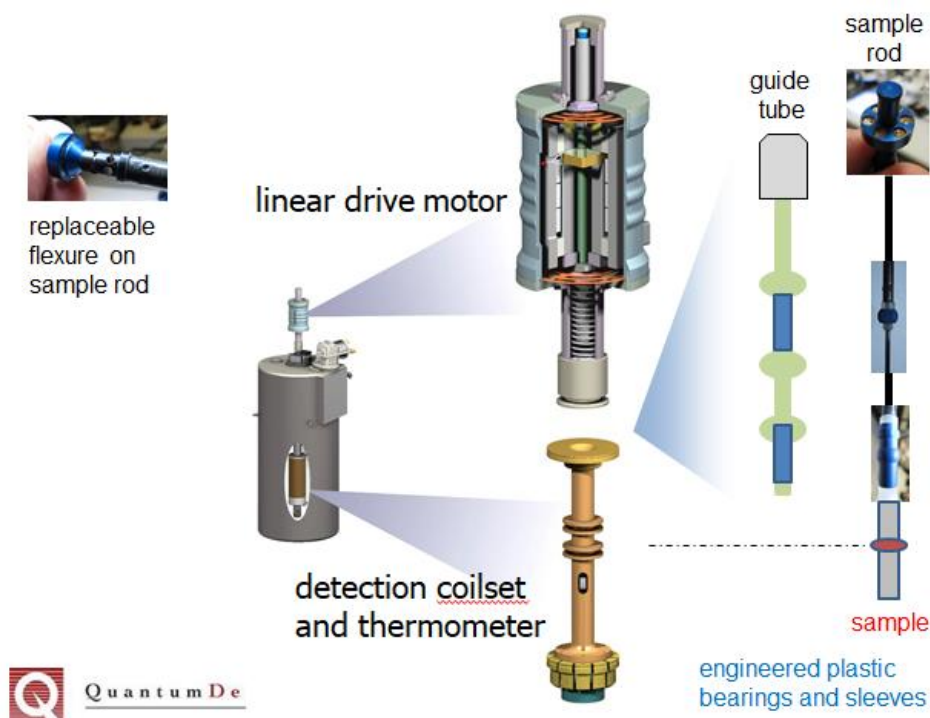
Στην εξίσωση (2.2),  $C$  είναι μία σταθερά σύζευξης,  $m$  είναι η μαγνήτιση του δείγματος,  $A$  είναι το πλάτος της ταλάντωσης και  $f$  είναι η συχνότητα της ταλάντωσης.

Η διεξαγωγή μετρήσεων μαγνήτισης προϋποθέτει τη μέτρηση του συντελεστή απόκρισης της ημιτονοειδούς τάσης των πηνίων ανίχνευσης.



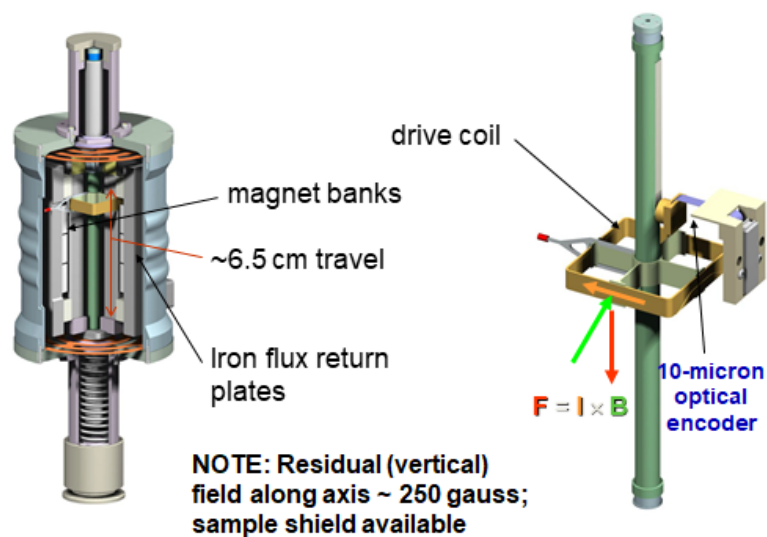
Εικόνα 1 - Αρχή λειτουργίας VSM

Το δείγμα τοποθετείται στο τέλος μίας ράβδου η οποία οδηγείται ημιτονοειδώς. Το κέντρο της ταλάντωσης τοποθετείται στο κατακόρυφο κέντρο του κλισιμέτρου. Η ακριβής θέση καθώς και το πλάτος της ταλάντωσης ελέγχεται από τον ελεγκτή του κινητήρα του VSM (motor module) με τη χρήση ενός οπτικού γραμμικού κωδικοποιητή. Η επαγωγική τάση του πηνίου ανίχνευσης ενισχύεται και λαμβάνεται από τον ελεγκτή ανίχνευσης του VSM (detection module). Ο ελεγκτής ανίχνευσης χρησιμοποιεί αυτό το σήμα θέσης του κωδικοποιητή ως σημείο αναφοράς για τη σύγχρονη ανίχνευση της τάσης. Τελικά από τα σήματα που λαμβάνει ο ελεγκτής ανίχνευσης προκύπτουν οι μέσοι όροι αυτών και στη συνέχεια μεταφέρονται στον υπολογιστή και στην εφαρμογή που τρέχει τη μέτρηση.



Εικόνα 2 - Αποτελούμενα μέρη ενός VSM

## VSM hardware: linear motor

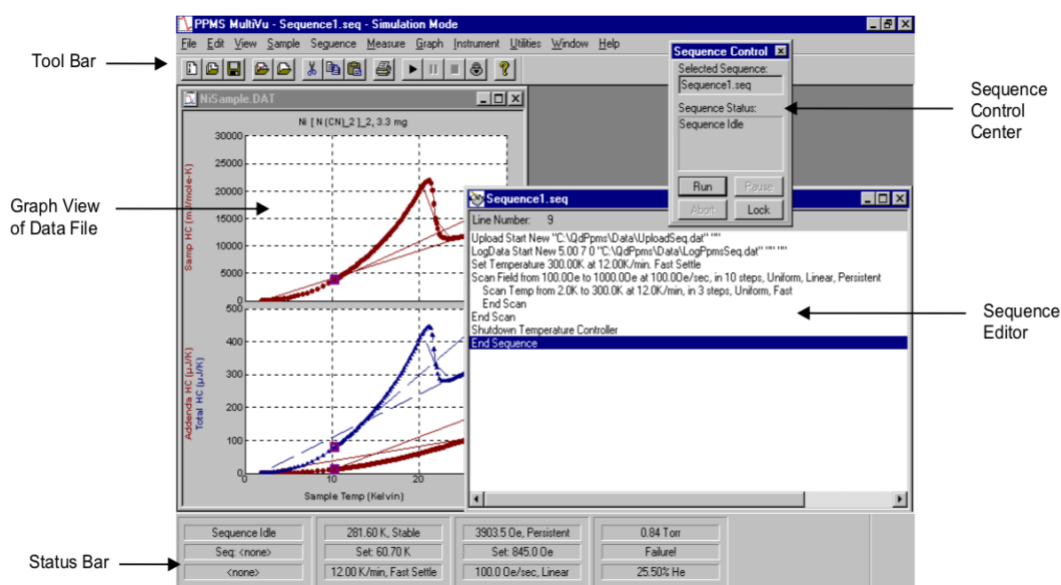


Εικόνα 3 - Γραμμικός κινητήρας VSM



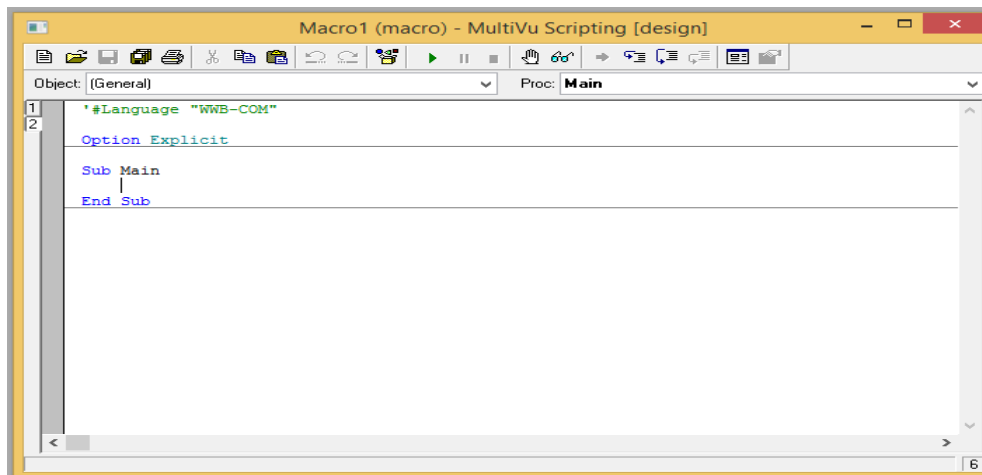
## 2.2. MultiVu

Η εταιρία Quantum Design προσφέρει το λογισμικό MultiVu [3] ως ενιαία διεπαφή για τη λειτουργία των πλατφορμών μετρήσεων της όπως το Versalab. Το λογισμικό δίνει τη δυνατότητα καθορισμού των μετρήσεων όπως επίσης και του ελέγχου και παρακολούθησης αυτών όπως επίσης προσφέρει τη δυνατότητα ανάλυσης των δεδομένων και αναφορά αποτελεσμάτων. Επιπροσθέτως διαθέτει έτοιμες ακολουθίες για αυτοματοποιημένες μετρήσεις καθώς και τη δυνατότητα δημιουργίας προσαρμοσμένων ακολουθιών. Πιο αναλυτικά ενσωματώνει οθόνη κατάστασης συστήματος, άμεσο έλεγχο συστήματος, επεξεργασία ακολουθίας, εκτέλεση ακολουθίας, εγκατάσταση επιλογών μετρήσεων, εκτέλεση μέτρησης, συλλογή δεδομένων και προβολή δεδομένων.



Εικόνα 4 - Το λογισμικό MultiVu

Επιπλέον η Quantum Design έχει ενσωματώσει έναν διερμηνέα της Visual Basic στο MultiVu την WinWrap Basic [4] και έναν editor για τη δημιουργία scripts. Η προσθήκη αυτή παρέχει πολύ πιο ισχυρές επιλογές προγραμματισμού στον προχωρημένο χρήστη σε σύγκριση με την αποκλειστική χρήση ακολουθιών. Μερικά παραδείγματα των πλεονεκτημάτων που είναι διαθέσιμα μέσω της χρήσης scripting της Visual Basic είναι η χρήση δηλώσεων υπό όρους για τη βελτιστοποίηση των μετρήσεων, η μετά-επεξεργασία ή η ανάλυση δεδομένων εν κινήσει και η ενσωμάτωση οργάνων τρίτων εταιρειών σε μετρήσεις.



Εικόνα 5 - Το περιβάλλον σύνταξης μακροεντολών του MultiVu

### 2.3. Ηλεκτρομαγνήτης

Το σύστημα του ηλεκτρομαγνήτη αποτελείται από το κύριο μέρος του μαγνήτη, το τροφοδοτικό και το σύστημα ψύξης. Τόσο το κύριο μέρος όσο και το τροφοδοτικό του ηλεκτρομαγνήτη αποτελούν ξεχωριστά μέρη του Υστερησιογράφου MH/40-20 της Walker Magnemetrics [5]. Πέραν αυτών η διάταξη του υστερησιογράφου περιλάμβανε τους ροομετρητές και το καταγραφικό μέρη τα οποία πλέον δε χρησιμοποιούνται.

Ο ηλεκτρομαγνήτης είναι τοποθετημένος σε βάση μάζας 226.7 κιλών με διαστάσεις μήκος x πλάτος x ύψος 73.7 cm x 78.7 cm x 83.8 cm. Η ψύξη του πραγματοποιείται με χρήση ψυκτικού νερού ροής 0.6814 m<sup>3</sup>/h σε μέγιστη ισχύ. Αποτελείται από δύο πλέγματα και μπορεί να παράξει σταθερό μαγνητικό πεδίο μέχρι 38 kOe (Oersteds) όταν οι πόλοι έχουν διάκενο 0.3175 cm [6].

Για την τροφοδοσία του μαγνήτη χρησιμοποιείται τροφοδοτικό των 5 kW με 100 A ρεύμα σε τάση 50 V. Η ψύξη του τροφοδοτικού επιτυγχάνεται επίσης μέσω υδρόψυξης με ψυκτικό μέσο το νερό με ροή στα 3 gpm (gallons per minute ή 0.6814 m<sup>3</sup>/h). Περιλαμβάνει τρεις διακόπτες για τη λειτουργία του (AC ON, DC OFF, DC ON) έναν ροοστάτη για τον έλεγχο του ρεύματος που πηγαίνει στον ηλεκτρομαγνήτη καθώς και έναν αναλογικό μετρητή που μας δίνει το ποσοστό του μέγιστου ρεύματος, που μπορεί να δώσει το τροφοδοτικό, που στέλνει στον ηλεκτρομαγνήτη. Η τροφοδοσία του γίνεται με τη σύνδεσή του σε τριφασικό ρεύμα συχνότητας 50 με 60 Hz και τάση από 208 – 240 Volts με χρήση τετραπολικού καλωδίου. Ο διακόπτης AC ON μας δείχνει όταν είναι αναμμένος ότι καλύπτονται οι ανάγκες τροφοδοσίας με μονοφασική ισχύ του low level τροφοδοτικού του ηλεκτρομαγνήτη καθώς και διάφορων οργάνων. Ο διακόπτης DC OFF όταν πατηθεί διακόπτει την τροφοδοσία του ηλεκτρομαγνήτη. Επίσης όταν η φωτεινή του ένδειξη είναι αναμμένη υποδηλώνει ότι η παροχή ψυκτικού

μέσου είναι επαρκής. Με τον διακόπτη DC ON εισάγεται DC ρεύμα στον ηλεκτρομαγνήτη για την παραγωγή μαγνητικού πεδίου.

Ο έλεγχος της έντασης του ρεύματος που θα φτάσει στον ηλεκτρομαγνήτη γίνεται μέσω δύο ποτενσιόμετρων. Το ένα ποτενσιόμετρο είναι συνδεδεμένο με το instrument assembly και ρυθμίζει το ποσοστό του μέγιστου ρεύματος που θα έχουμε διαθέσιμο για τον ηλεκτρομαγνήτη. Το δεύτερο ποτενσιόμετρο είναι στην ουσία ο ροοστάτης Coarse Current Control πάνω στον πίνακα ελέγχου του τροφοδοτικού και ρυθμίζει το ποσοστό της τιμής του ρεύματος του παραπάνω ποτενσιόμετρου που θα φτάσει τελικά στον ηλεκτρομαγνήτη.

Τα κυκλώματα ψύξης του ηλεκτρομαγνήτη και του τροφοδοτικού είναι συνδεδεμένα σε σειρά και απαιτείται ροή ψυκτικού νερού για την λειτουργία του συστήματος σε πλήρη ισχύ στα 3 grm. Εφόσον η ροή ξεπεράσει το κατώφλι των 1.9 grm διακόπτεται η λειτουργία του συστήματος για ασφάλεια. Το νερό που ρέει μέσα στα κυκλώματα ψύξης δε θα πρέπει να ξεπερνάει τη θερμοκρασία των 27 °C και η πίεσή του θα πρέπει να βρίσκεται μεταξύ των 3.1 και 7.9 bar. Αν κριθεί απαραίτητη η χρήση του συστήματος σε χαμηλή ισχύ λειτουργίας τότε θα πρέπει να αναμειχθεί το ψυκτικό νερό με ζεστό προκειμένου να εξαλείψουμε την πιθανότητα να αναπτυχθούν συμπυκνώματα στα πηνία του ηλεκτρομαγνήτη. Τα πηνία ψύξης του μαγνήτη είναι κατασκευασμένα από χαλκό, ενώ θωρακίζονται και απομονώνονται από το κύκλωμα τροφοδοσίας. Αντίθετα τα πηνία ψύξης του τροφοδοτικού είναι από αλουμίνιο και συνδέονται με την είσοδο common αυτού.



Εικόνα 6 - Η διάταξη του ηλεκτρομαγνήτη με το VSM

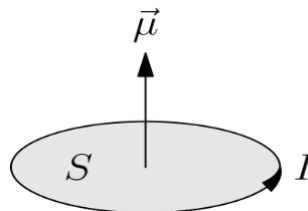
### 3. Θεωρία

#### 3.1. Βασικές Παράμετροι Μετρήσεων

Οι μαγνητικές ιδιότητες των υλικών είναι συνέπεια των μαγνητικών ροπών οι οποίες οφείλονται στα ηλεκτρόνια των ατόμων τους. Η μαγνητική διπολική ροπή είναι μια βασική ποσότητα για τον μαγνητισμό γιατί καθορίζει: (1) το μαγνητοστατικό πεδίο που παράγει το μαγνητικό υλικό που περιέχει μαγνητικές ροπές στον περιβάλλοντα χώρο και (2) την επίδραση που θα έχει ένα μαγνητικό πεδίο στο μαγνητικό υλικό. **Μαγνητική ροπή**  $\mu$  ορίζεται ως το ρεύμα  $I$  που διαρρέει ένα βρόγχο επί την επιφάνεια  $S$  του βρόγχου [7].

$$M = I * S \quad (3.1)$$

Κάθε ηλεκτρόνιο ενός ατόμου έχει μαγνητικές ροπές που οφείλονται σε δύο πηγές. Η μια σχετίζεται με την τροχιακή κίνηση γύρω από τον πυρήνα. Κάθε ηλεκτρόνιο κινούμενο γύρω από τον πυρήνα μπορεί να θεωρηθεί σαν ένας μικρός βρόγχος που δημιουργεί ένα πολύ μικρό μαγνητικό πεδίο με μαγνητική ροπή κατά μήκος του άξονα στροφής. Η δεύτερη πηγή μαγνητικής ροπής οφείλεται στο spin των ηλεκτρονίων δηλαδή την περιστροφή τους γύρω από τον άξονά του. Η συνολική μαγνητική ροπή ενός ατόμου είναι η συνισταμένη των δυο αυτών μαγνητικών ροπών.



Εικόνα 7 - Μαγνητική Ροπή

Στη συνέχεια ορίζονται κάποια βασικά μαγνητικά μεγέθη που σχετίζονται με τις μαγνητικές μετρήσεις υλικών.

**Ένταση μαγνητικού πεδίου  $H$ :** Αν ένα σωληνοειδές μήκους  $L$  διαρρέεται από ρεύμα έντασης  $I$  τότε δημιουργείται στο εσωτερικό του μαγνητικό πεδίο έντασης  $H$ . Αν το σωληνοειδές αποτελείται από  $N$  σπείρες τότε η ένταση του μαγνητικού πεδίου δίνεται από την εξίσωση (3.2):

$$H = \left(\frac{N * I}{L}\right) \quad (3.2)$$

**Μαγνητική επαγωγή η πυκνότητα μαγνητικής ροής  $B$ :** αναπαριστά το μέγεθος της έντασης του εσωτερικού πεδίου μέσα σε ένα τμήμα που εκτίθεται σε ένα πεδίο  $H$ .

Η μαγνητική επαγωγή συνδέεται με την ένταση του μαγνητικού πεδίου από την σχέση της εξίσωσης (3.3)

$$B = \mu * H \quad (3.3)$$

Όπου η παράμετρος  $\mu$  ορίζεται ως η μαγνητική διαπερατότητα του μέσου μέσω του οποίου διαδίδεται το πεδίο  $H$  και υπολογίζεται το  $B$ . Η μαγνητική διαπερατότητα στο κενό συμβολίζεται με  $\mu_0$  και έχει τιμή  $\mu_0 = 12,57 * 10^{-7} \text{ Vsec/Am}$ . Η Σχετική μαγνητική διαπερατότητα  $\mu_r$  ενός υλικού που δίνεται από τη σχέση στην εξίσωση (3.4) είναι ένα αδιάστατο μέγεθος που μπορεί να περιγράψει τις μαγνητικές ιδιότητες στερεών. Το  $\mu_r$  ενός υλικού δείχνει τον βαθμό στο οποίο το υλικό αυτό μπορεί να μαγνητιστεί.

$$\mu_r = \frac{\mu}{\mu_0} \quad (3.4)$$

**Μαγνήτιση  $M$ :** εκφράζει την δράση των μαγνητικών διπολικών ροπών των υλικών ως πηγών του πεδίου. Συνδέεται με το  $B$  και  $H$  με τη σχέση (3.5) και (3.6),

$$B = \mu_0 * (H + M) \quad (3.5)$$

$$M = \chi_m * H \quad (3.6)$$

όπου η ποσότητα  $\chi_m$  ονομάζεται μαγνητική επιδεκτικότητα και είναι μια αδιάστατη σταθερά. Η  $\chi_m$  συνδέεται με την μαγνητική διαπερατότητα με την σχέση (3.7).

$$\chi_m = \mu_r - 1 \quad (3.7)$$

Στον παρακάτω πίνακα φαίνονται οι μονάδες των μεγεθών που περιεγράφηκαν στα συστήματα μονάδων SI και CGS.

| Φυσικό μέγεθος  | SI                  | CGS                                  |
|---|---------------------|--------------------------------------|
| $B$ (μαγνητική επαγωγή)                               | Tesla=T             | Gauss=G                              |
| $H$ (μαγνητικό πεδίο)                                 | A/m                 | Oe                                   |
| $M$ (μαγνήτιση)                                       | A/m                 | emu/cm <sup>3</sup>                  |
| $m$ (μαγνητική ροπή)                                  | Am <sup>2</sup>     | emu                                  |
| $m_o=m/\text{mass}$ (μαγνητική ροπή ανά μονάδα μάζας) | Am <sup>2</sup> /Kg | emu/g                                |
|   |                     |                                      |
|   | $B=\mu_o(M+H)$      | $B=H+4\pi M$                         |
| $\chi=dM/dH$ (επιδεκτικότητα)                         | χωρίς διαστάσεις(1) | emu cm <sup>3</sup> Oe <sup>-1</sup> |

Πίνακας 1 - Μονάδες μεγεθών

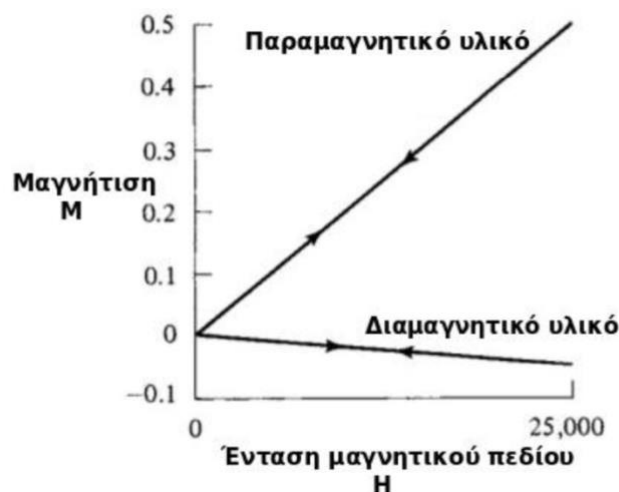
### 3.2. Κατηγορίες Υλικών

Τα διάφορα υλικά μπορούν να χωριστούν σε πέντε κατηγορίες ανάλογα με τις μαγνητικές τους ιδιότητες, τα παραμαγνητικά, τα σιδηρομαγνητικά, τα διαμαγνητικά, τα αντισιδηρομαγνητικά και τα σιδηριμαγνητικά [8].

- 1) **Παραμαγνητικά (paramagnetic)** είναι τα υλικά που προκαλούν ενίσχυση του επιβαλλόμενου εξωτερικού μαγνητικού πεδίου και έχουν σχετική μαγνητική διαπερατότητα λίγο μεγαλύτερη της μονάδας  $\mu_r > 1$ . Τα υλικά αυτά θα απομαγνητιστούν πλήρως αν αφαιρεθεί το εξωτερικό μαγνητικό πεδίο.
- 2) **Διαμαγνητικά (diamagnetic)** είναι τα υλικά που προκαλούν εξασθένηση του επιβαλλόμενου εξωτερικού μαγνητικού πεδίου και έχουν σχετική μαγνητική διαπερατότητα λίγο μικρότερη της μονάδας  $\mu_r < 1$ . Τα διαμαγνητικά υλικά επίσης θα απομαγνητιστούν πλήρως αν αφαιρεθεί το εξωτερικό μαγνητικό πεδίο.
- 3) Τα **σιδηρομαγνητικά (ferromagnetic)** υλικά που εμφανίζουν διαφορετικές ιδιότητες από τις προηγούμενες κατηγορίες. Η παρουσία τους σε εξωτερικό μαγνητικό πεδίο προκαλεί την κατά πολλές φορές ενίσχυση του πεδίου και έχουν μαγνητική διαπερατότητα πολύ μεγαλύτερη της μονάδας  $\mu_r \gg 1$ . Τα υλικά αυτά θα παραμένουν μαγνητισμένα ακόμα και αν αφαιρεθεί το εξωτερικό πεδίο.

- 4) **Αντισιδηρομαγνητικά (antiferromagnetic)**, είναι τα υλικά στα οποία οι διαδοχικές διπολικές μαγνητικές ροπές είναι αντιπαράλληλες και ίσες κατά μέτρο και έχουν συνισταμένη ροπή ίση με μηδέν. Έτσι τα υλικά αυτά εμφανίζονται ως αμαγνήτιστα ακόμα και κατά την επιβολή μαγνητικού πεδίου.
- 5) Τα **σιδηριμαγνητικά (ferrimagnetic)** υλικά εμφανίζουν και αυτά αντιπαράλληλες διπολικές μαγνητικές ροπές όμως τα μέτρα των ροπών δεν είναι ίσα. Η συμπεριφορά των υλικών αυτών είναι παρόμοια με τα σιδηρομαγνητικά αλλά με εξασθενημένα χαρακτηριστικά.

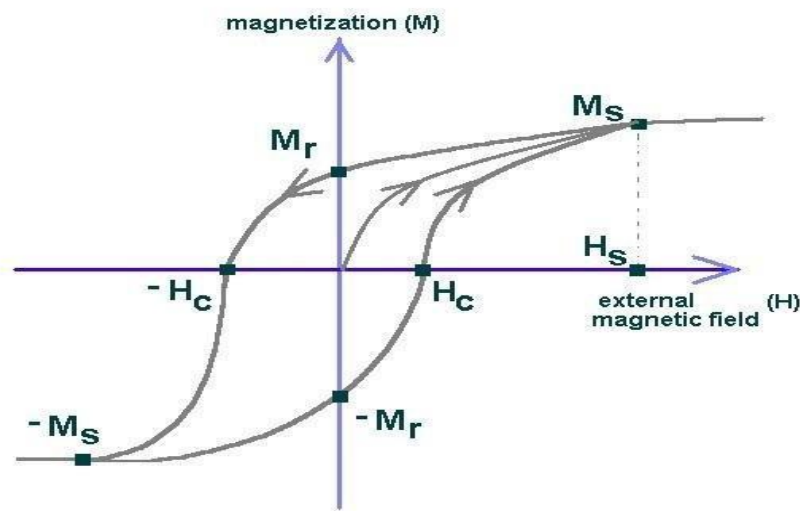
Για την περιγραφή των μαγνητικών ιδιοτήτων συχνά χρησιμοποιείται η καμπύλη μαγνήτισης. Στα επόμενα σχήματα υπάρχει παράδειγμα τέτοιων καμπυλών. Στην εικόνα 8 βλέπουμε την καμπύλη μαγνήτισης για παραμαγνητικά και σιδηρομαγνητικά υλικά, όπου γίνεται σαφές ότι δεν διατηρούν την μαγνήτισή τους μετά την απομάκρυνση του πεδίου.



Εικόνα 8 - Καμπύλη μαγνήτισης

Στην εικόνα 9 παρουσιάζεται η καμπύλη μαγνήτισης σε ένα σιδηρομαγνητικό υλικό αρχικά αμαγνήτιστο. Αρχικά με την αύξηση του εξωτερικού πεδίου αυξάνεται και η μαγνήτιση του υλικού μέχρι το σημείο  $M_s$ . Όταν η μαγνήτιση του υλικού φτάσει αυτό το σημείο τότε περαιτέρω αύξηση του εξωτερικού πεδίου έχει ως αποτέλεσμα πολλή μικρή μεταβολή στη μαγνήτιση του υλικού. Το φαινόμενο αυτό ονομάζεται μαγνητικός κορεσμός. Στη συνέχεια μειώνεται το εξωτερικό πεδίο μέχρι να μηδενιστεί και παρατηρούμε ότι το υλικό παραμένει μαγνητισμένο. Η τιμή αυτή της μαγνήτισης του υλικού ονομάζεται παραμένουσα μαγνήτιση. Το φαινόμενο αυτό ονομάζεται υστέρηση. Για τον μηδενισμό της μαγνήτισης του υλικού πρέπει να εφαρμοστεί εξωτερικό πεδίο αντίθετης φοράς.

Αυξάνοντας το πεδίο κατά την ίδια φορά η μαγνήτιση του υλικού φτάνει στο σημείο κορεσμού με αντίθετη φορά από ότι προηγουμένως αφού οι μαγνητικές ροπές θα έχουν ίδια φορά με το πεδίο. Στη συνέχεια μειώνοντας το πεδίο μέχρι να μηδενιστεί, το υλικό θα μείνει και πάλι μαγνητισμένο. Αλλάζοντας και πάλι τη φορά του εξωτερικού πεδίου και αυξάνοντας το, η μαγνήτιση του υλικού θα φτάσει και πάλι το σημείο κορεσμού. Ο βρόγχος αυτός ονομάζεται βρόγχος υστέρησης [8].



Εικόνα 9 - Βρόχος υστέρησης

### 3.3. Μετατροπή Αναλογικού Σήματος Σε Ψηφιακό

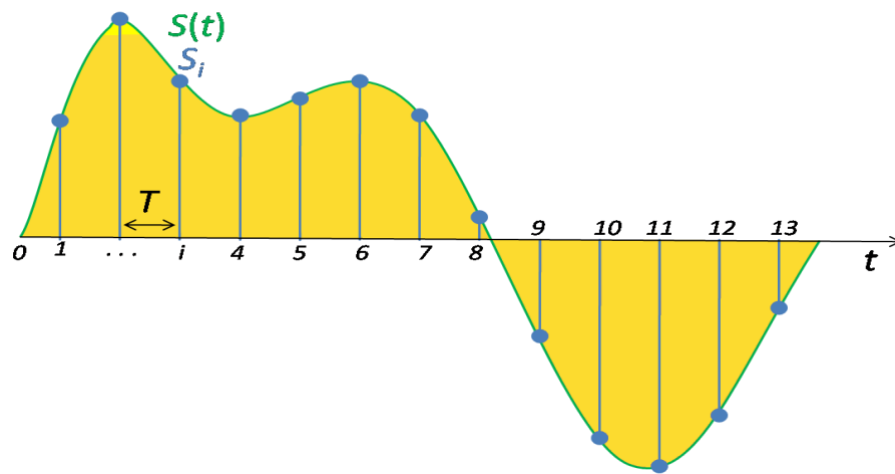
#### 3.3.1. Ορισμοί

Στα πλαίσια της διπλωματικής είναι απαραίτητη η μέτρηση της τάσης του ηλεκτρομαγνήτη και η εισαγωγή των τιμών αυτών σε υπολογιστή. Για το σκοπό αυτό τοποθετήθηκε μια διάταξη στον ηλεκτρομαγνήτη που πραγματοποιεί τη μέτρηση και την ψηφιοποίηση των τιμών. Η μετατροπή αναλογικού σε ψηφιακό σήμα περιλαμβάνει τα στάδια της δειγματοληψίας του κβαντισμού και της κωδικοποίησης και πραγματοποιούνται μέσω ενός μετατροπέα αναλογικού σε ψηφιακό σήματος ή αλλιώς ADC (Analog to Digital Converter).

Δειγματοληψία (sampling) ονομάζεται η διαδικασία με την οποία παράγουμε ένα διακριτό από ένα συνεχές σήμα το οποίο προκύπτει από την καταγραφή των τιμών του συνεχούς σήματος στιγμιαία και με ομοιόμορφο τρόπο. Τα δείγματα λαμβάνονται σε σειρά ως προς τον χρόνο, τον χώρο ή τον χωροχρόνο. Ο ρυθμός με τον οποίο καταγράφονται οι τιμές ονομάζεται συχνότητα δειγματοληψίας ή

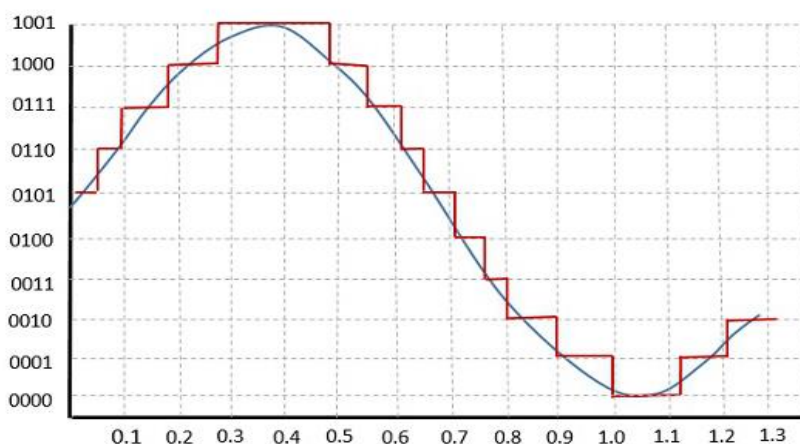


ρυθμός δειγματοληψίας και μας δείχνει πόσα δείγματα έχουν ληφθεί από τον δειγματολήπτη σε διάρκεια ενός δευτερολέπτου.



Εικόνα 10 - Δειγματοληψία σήματος

Η δειγματοληψία όπως αναφέρθηκε διακρίνει ένα συνεχές σήμα ως προς το χρόνο, το χώρο ή χωροχρόνο. Το πλάτος όμως του σήματος παίρνει συνεχείς τιμές. Για την αναπαράσταση του πλάτους του σήματος σε ψηφιακή μορφή είναι απαραίτητη και η διάκριση του πλάτους του σήματος. Η διαδικασία αυτή ονομάζεται κβαντισμός. Κατά τη διαδικασία του κβαντισμού το εύρος του πλάτους του σήματος χωρίζεται σε η μικρότερα εύρη τιμών, τις κβαντικές στάθμες, και σε κάθε ένα από αυτά αντιστοιχίζεται ένας ψηφιακός κωδικός. Κάθε δείγμα παίρνει την κοντινότερη ψηφιακή τιμή σε σχέση με την πραγματική τιμή του.



Εικόνα 11 - Κβαντισμός σήματος

Κωδικοποίηση ονομάζουμε την αναπαράσταση των κβαντισμένων τιμών πλάτους των δειγμάτων του σήματος σε δυαδική ακολουθία (ακολουθία δυαδικών ψηφίων). Δημιουργείται έτσι το τελικό ψηφιακό σήμα σαν μια σειρά bits.

Σηματοθορυβικό λόγο η αλλιώς SNR (Signal to Noise Ratio) ονομάζουμε το λόγο της τιμής του επιθυμητού σήματος προς την τιμή του θορύβου που εμπεριέχεται σε ένα σήμα δηλαδή των ανεπιθύμητων παρεμβολών που επενεργούν στο σήμα. Παραδείγματος χάρη το σφάλμα κβαντισμού μπορεί να θεωρηθεί ως θόρυβος. Ο σηματοθορυβικός λόγος εκφράζεται συνήθως σε db (decibel) [9].

$$SNR = 20 * (\text{Πλάτος Σήματος} / \text{Πλάτος Θορύβου}) \quad (3.8)$$

### 3.3.2. Analog to Digital Converter

[10 ][11]

Μετατροπέας αναλογικού σήματος σε ψηφιακό (ADC) ονομάζεται το κύκλωμα που μετατρέπει ένα αναλογικό σήμα σε ψηφιακό. Για παράδειγμα αν ο μετατροπέας ως έχει ως είσοδο μια τάση, η έξοδός του θα είναι μια ψηφιακή λέξη που θα αντιστοιχεί στην τάση εισόδου. Το ADC λειτουργεί ως εξής, συγκρίνει την τάση εισόδου με μια άλλη τάση η οποία ονομάζεται τάση αναφοράς και συμβολίζεται με το  $V_{Ref}$ . Η  $V_{Ref}$  παρέχεται από μια πηγή αναφοράς. Το αποτέλεσμα της σύγκρισης αυτής είναι η ψηφιακή λέξη που εκφράζει τον λόγο της τάσης εισόδου προς την τάση αναφοράς και αντιστοιχεί στη κβαντική στάθμη στην οποία αντιστοιχίζεται η τιμή.

Βασικά χαρακτηριστικά ενός ADC είναι:

- I. Το εύρος μετατροπής. Αυτό εκφράζει το μέγιστο σήμα εισόδου που μπορεί να έχει ο μετατροπέας ως είσοδο. Ένας μετατροπέας που έχει την ικανότητα για μετατροπή σημάτων μόνο μιας πολικότητας δηλαδή είτε θετικού είτε αρνητικού σήματος ονομάζεται μονοπολικός. Όταν όμως έχει την δυνατότητα για μετατροπή και θετικών και αρνητικών σημάτων ονομάζεται διπολικός. Το εύρος εισόδου είναι από 0 έως την τάση αναφοράς για μονοπολικούς μετατροπείς και  $-V_{Ref}$  έως  $V_{Ref}$  για διπολικούς.
- II. Διακριτική ικανότητα ADC. Η έξοδος ενός μετατροπέα αποτελείται από έναν αριθμό bit. Ο αριθμός αυτός υποδηλώνει τον αριθμό των διαφορετικών διακριτών τιμών που μπορεί να έχει ο μετατροπέας. Ένας μετατροπέας με n bit εξόδου μπορεί να παράγει  $2^n$  διαφορετικές τιμές δηλαδή έχει  $2^n$  κβαντικές στάθμες. Διακριτική ικανότητα του μετατροπέα είναι ο λόγος του εύρους εισόδου ως προς τον αριθμό των τιμών εξόδου, δηλαδή είναι το εύρος κάθε κβαντικής στάθμης του μετατροπέα.

Για μονοπολικούς ADC ισχύει:

$$resolution = \frac{V_{Ref}}{2^n} \quad (3.9)$$

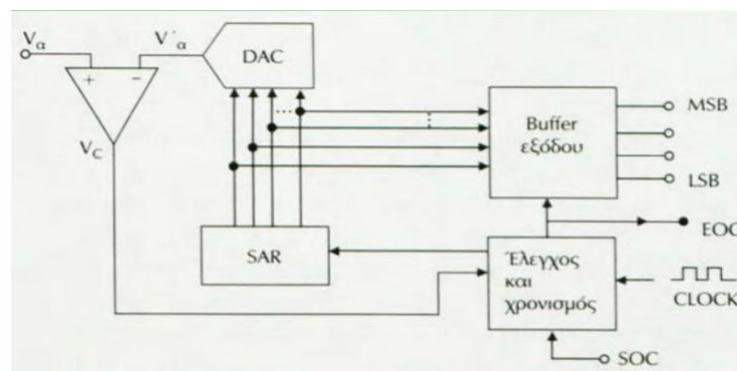
Για διπολικούς ADC ισχύει:

$$resolution = \frac{2*V_{Ref}}{2^n} \quad (3.10)$$

- III. Χρόνος μετατροπής είναι ο χρόνος που χρειάζεται ένας ADC ώστε να μετατρέψει την αναλογική είσοδο σε ψηφιακή έξοδο.

### 3.3.3. Μετατροπέας A/D διαδοχικών προσεγγίσεων [11]

Υπάρχουν διάφορες τεχνικές που χρησιμοποιούνται ώστε να γίνει μια μετατροπή αναλογικού σε ψηφιακό σήμα. Το ADC που διαθέτει το Arduino που χρησιμοποιείται για αυτή την εργασία χρησιμοποιεί την τεχνική των διαδοχικών προσεγγίσεων. Το κύκλωμα για να εφαρμοστεί η τεχνική αυτή φαίνεται στην εικόνα 12.



Εικόνα 12 - Μετατροπέας A/D διαδοχικών προσεγγίσεων

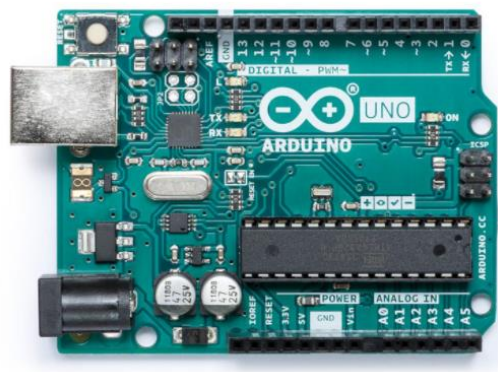
Η λειτουργία του κυκλώματος είναι η εξής. Έστω ότι ο μετατροπέας ADC έχει n-bit εξόδου. Ως MSB (Most Significant Bit) ορίζεται το αριστερό bit ενός δυαδικού καταχωρητή, δηλαδή το n-1 bit. Όταν δοθεί το σήμα εκκίνησης ο καταχωρητής

εξόδου παίρνει τιμή ίση με το μισό του εύρους μετατροπής δηλαδή το MSB γίνεται 1. Η τιμή του καταχωρητή συγκρίνεται με την αναλογική είσοδο. Αν η αναλογική είσοδος είναι μεγαλύτερη μένει το 1 στο MSB αλλιώς μηδενίζεται. Στη συνέχεια το bit n-2 του καταχωρητή εξόδου γίνεται 1 ενώ το προηγούμενο bit διατηρεί την τιμή που ορίστηκε από την σύγκριση. Έτσι τώρα ο καταχωρητής εξόδου έχει τιμή ίση είτε με  $\frac{3}{4}$  είτε με το  $\frac{1}{4}$  της τιμής του εύρους του μετατροπέα. Αν η τιμή που έχει τώρα ο καταχωρητής είναι μεγαλύτερη της αναλογικής εξόδου τότε το n-2 bit του καταχωρητή παραμένει 1 αλλιώς μηδενίζεται. Η διαδικασία αυτή πραγματοποιείται n φορές όσες και τα bit εξόδου του μετατροπέα.

## 4. Συστήματα Μετρητικής Διάταξης

### 4.1. Arduino

Ο Arduino είναι μια πλατφόρμα υπολογισμών ανοιχτού κώδικα η οποία αποτελείται από την πλακέτα που βασίζεται σε έναν απλό μικροεπεξεργαστή και το περιβάλλον προγραμματισμού στο οποίο συντάσσεται ο κώδικας που ρυθμίζει τις διάφορες λειτουργίες του. Μέσα από αυτό το περιβάλλον καθίσταται πολλή εύκολη η μεταγλώττιση και η αποστολή του κώδικα στο μικροελεγκτή. Στα πλαίσια της διπλωματικής επιλέχθηκε ο Arduino Uno ως το μέσο που θα διαβάζει την αναλογική έξοδο του οργάνου μέτρησης του πεδίου.



Εικόνα 13 - Arduino uno

#### 4.1.1. Τεχνικά Χαρακτηριστικά

Ο Arduino Uno [12] είναι ένας μικροελεγκτής βασισμένος στον ATmega328 ο οποίος επίσης είναι ένας μικροελεγκτής υψηλών επιδόσεων των 8 bit με εύρος λειτουργίας από 1.8 – 5.5 volt. Ο Arduino αποτελείται από 14 ψηφιακές εισόδους/εξόδους, 6 αναλογικές εισόδους, έναν κεραμικό συντονιστή 16 MB, παροχή δυνατότητας σύνδεσης USB , υποδοχή εξωτερικής τροφοδοσίας και κουμπί reset [12].

Η τροφοδοσία του μπορεί να γίνει είτε μέσω της θύρας USB (5V,1A) είτε μέσω εξωτερικής τροφοδοσίας DC (μετασχηματιστής, μπαταρίες). Η εξωτερική τροφοδοσία μπορεί να πραγματοποιηθεί και από το pin VIN του Arduino. Το εύρος λειτουργίας του είναι τα 6-20 Volt αλλά η ιδανική τάση θεωρείται από 7 – 12 Volt [12].

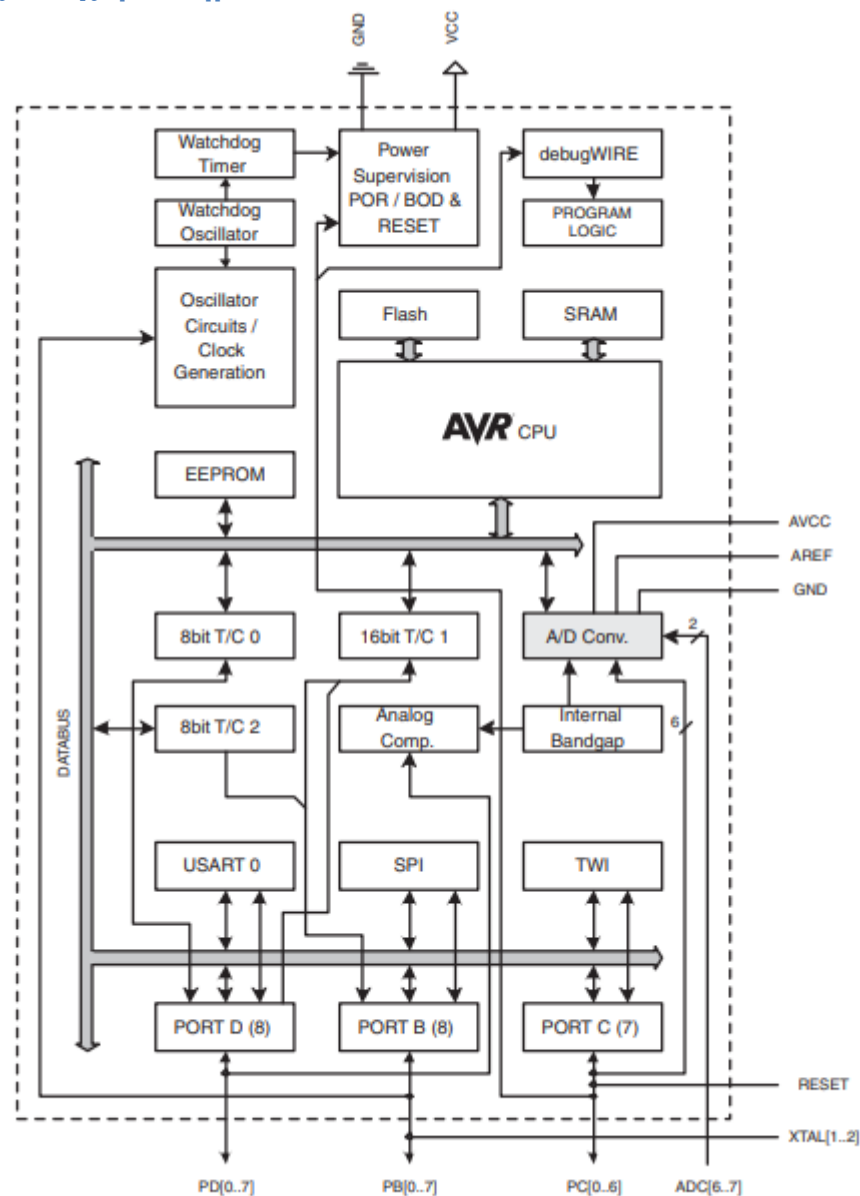
#### 4.1.2. Λογισμικό

Το περιβάλλον του Arduino είναι ανοιχτού κώδικα και πολύ εύκολο στη χρήση. Είναι γραμμένο σε Java και βασίζεται στο Processing και σε άλλα προγράμματα

ανοιχτού κώδικα. Μπορεί να τρέξει σε πολλαπλά λειτουργικά συστήματα όπως Windows, Mac και Linux. Είναι σχεδιασμένο ώστε να εισαγάγει τον προγραμματισμό σε ανθρώπους που δεν είναι εξοικειωμένοι με την ανάπτυξη εφαρμογών. Περιλαμβάνει έναν editor μέσω του οποίου είναι δυνατή η σύνταξη και η επεξεργασία του κώδικα. Μέσω αυτού του IDE είναι δυνατή η μεταγλώττιση του κώδικά μας και το «ανέβασμά» του στον μικροελεγκτή. Τα προγράμματα του Arduino συντάσσονται σε C ή C++ [12].

## 4.2. ATmega328p

### 4.2.1. Τεχνικά χαρακτηριστικά



Εικόνα 14 - Block διάγραμμα

Ο ATmega328p [13] είναι ένας μικροελεγκτής που μπορεί να δεχτεί δεδομένα των 8 bit, κατασκευασμένος από τη Microchip, διαθέτει 28 pins και ακολουθεί την

αρχιτεκτονική RISC έχοντας ενσωματωμένη μνήμη προγραμματισμού στα 32KB. Περιλαμβάνει μία ηλεκτρικά απαλείψιμη προγραμματίσιμη μνήμη μόνο για ανάγνωση (Electrical Erasable Programmable Read-Only Memory ή EEPROM) του 1KB. Αυτή έχει τη δυνατότητα να διατηρεί τα δεδομένα ακόμα και αν σταματήσει η τροφοδοσία προς το μικροελεγκτή και μπορεί να παρέχει τα αποτελέσματα αφού συνδεθεί ξανά στο ρεύμα. Επιπρόσθετα ο ATmega328 περιέχει μία στατική μνήμη τυχαίας προσπέλασης (Static Random Access Memory ή SRAM) στα 2KB. Για τη λειτουργία του ADC (Analog to Digital Converter) διαθέτει 8 pins τα οποία ανήκουν στην PortA (PA0 – PA7). Επιπλέον ενσωματώνει 3 χρονόμετρα, δύο εκ των οποίων είναι των 8 bit ενώ το 3<sup>ο</sup> είναι στα 16 bit. Ο μικροελεγκτής μπορεί να λειτουργήσει σε ένα εύρος τάσης 3.3V – 5.5 V αλλά συνήθως επιλέγεται μια τάση στα 5V. Κάποια επιπρόσθετα χαρακτηριστικά του είναι το χαμηλό κόστος κατασκευής, οι χαμηλές απαιτήσεις σε ενέργεια και η λειτουργία κλειδώματος προγραμματισμού για λόγους ασφαλείας [13].

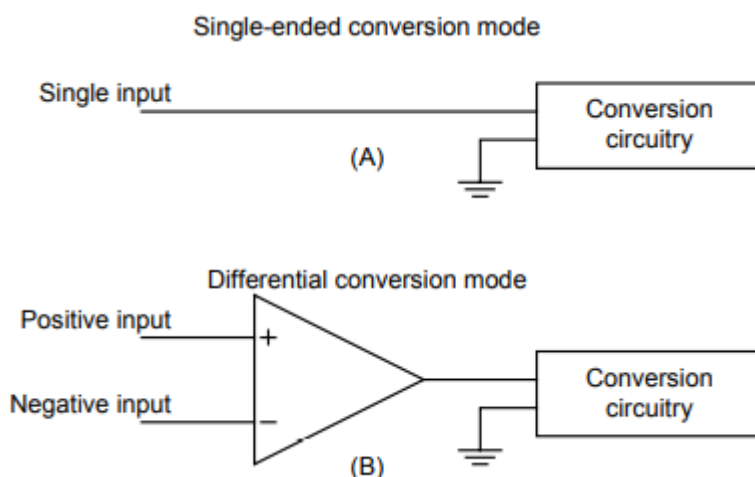
| ATmega328 Χαρακτηριστικά                  |                       |
|---|-----------------------|
| Αριθμός Pin                               | 28                    |
| CPU                                       | RISC 8-Bit AVR        |
| Τάση λειτουργίας                          | 3.3V με 5.5V          |
| Μνήμη Προγραμματισμού                     | 32KB                  |
| Τύπος Μνήμης Προγραμματισμού              | Flash                 |
| SRAM                                      | 2048 Bytes            |
| EEPROM                                    | 1024 Bytes            |
| ADC                                       | 10-Bit                |
| Αριθμός καναλιών ADC                      | 8                     |
| Κανάλια Διαμόρφωσης Πλάτους Παλμού (PWM)  | 6                     |
| Συγκριτές                                 | 1                     |
| Ταλαντωτής                                | Έως 20 MHz            |
| Χρονόμετρα                                | 8-Bit x 2, 16-Bit x 1 |
| Βελτιωμένο Power on Reset                 | Ναι                   |
| Χρονόμετρο Εκκίνησης                      | Ναι                   |
| Κανάλια Εισόδου/Εξόδου                    | 23                    |
| Κατασκευαστής                             | Microchip             |
| Σειριακή Περιφερειακή Διεπαφή (SPI)       | Ναι                   |
| I2C                                       | Ναι                   |
| Ανίχνευση πτώσης τάσης (Brown-out Detect) | Ναι                   |
| Δυνατότητα Επανεκκίνησης                  | Ναι                   |
| USI (Universal Serial Interface)          | Ναι                   |
| Εύρος Θερμοκρασίας Λειτουργίας            | -40 °C - +85 °C       |

Πίνακας 2 - Τεχνικά Χαρακτηριστικά ATmega328

#### 4.2.2. Analog to Digital Converter (ADC)

Το σημαντικότερο κομμάτι στον μικροελεγκτή ATmega328p [13] είναι ο μετατροπέας από Αναλογικό σε Ψηφιακό. Μέσω αυτού η είσοδος της αναλογικής τάσης που θα ληφθεί από το όργανο μέτρησης θα μετατραπεί σε ψηφιακή έξοδο ώστε να είναι δυνατή η ανάγνωσή της από τον υπολογιστή.

Ο ADC του μικροελεγκτή μπορεί να ρυθμιστεί τόσο για απλές μετατροπές όσο και για διαφορικές. Η απλή μετατροπή χρησιμοποιείται για τη μέτρηση τάσης εισόδου από ένα κανάλι ενώ η διαφορική μετράει τη διαφορά μεταξύ δύο καναλιών. Ανεξαρτήτως της επιλογής οι τάσεις εισόδου οποιουδήποτε καναλιού θα πρέπει να βρίσκονται μέσα στα όρια GND και  $AV_{CC}$ . Όταν χρησιμοποιείται η απλή μετατροπή η τάση που μετριέται μεταξύ της εισόδου και της γείωσης μετατρέπεται σε ψηφιακή τιμή. Με τη χρήση διαφορικών καναλιών η έξοδος από έναν διαφορικό ενισχυτή μετατρέπεται σε μία ψηφιακή τιμή.



Εικόνα 15 - Α) Λειτουργία απλής μετατροπής Β) Λειτουργία διαφορικής μετατροπής

Ο ATmega328 [13] διαθέτει μετατροπέα ADC διαδοχικών προσεγγίσεων 10-bit. Το ADC είναι συνδεδεμένο με έναν αναλογικό πολυπλέκτη 8 καναλιών που επιτρέπει οκτώ εισόδους τάσης. Το ADC περιέχει ένα κύκλωμα δείγματος και συγκράτησης που εξασφαλίζει ότι η τάση εισόδου στο ADC διατηρείται σε σταθερό επίπεδο κατά τη διάρκεια της μετατροπής. Το διάγραμμα του κυκλώματος του ADC φαίνεται στην εικόνα 16.

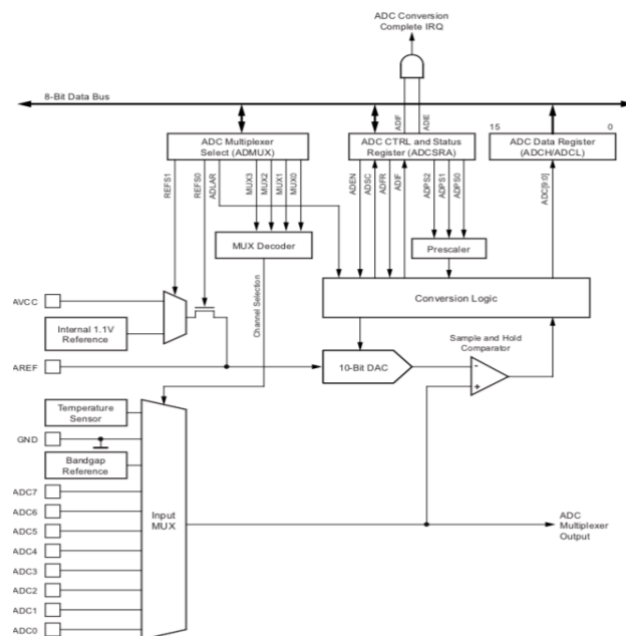


Το ADC μετατρέπει μια αναλογική τάση εισόδου σε ψηφιακή τιμή 10-bit. Η ελάχιστη τιμή αντιπροσωπεύει το GND και η μέγιστη τιμή αντιπροσωπεύει την τάση αναφοράς.

Η τάση αναφοράς  $V_{ref}$  για το ADC υποδεικνύει το εύρος μετατροπής του. Αν εφαρμοστεί τάση εισόδου στον ADC που υπερβαίνει το  $V_{ref}$  θα έχει ως αποτέλεσμα η ψηφιακή τιμή να παίρνει την τιμή της  $V_{ref}$ . Το  $V_{ref}$  μπορεί να επιλεγεί είτε ως η  $AV_{CC}$  που στο Arduino Uno ισούται με περίπου 5V, είτε ως η εσωτερική τάση αναφοράς 1.1V, είτε να συνδεθεί μια εξωτερική τάση στο pin  $A_{ref}$ .

Το  $AV_{CC}$  συνδέεται στο ADC μέσω ενός διακόπτη. Η εσωτερική τάση αναφοράς 1.1V δημιουργείται από την εσωτερική τάση αναφοράς εύρους ζώνης  $V_{BG}$  (bandgap reference) μέσω ενός εσωτερικού ενισχυτή. Το  $V_{ref}$  μπορεί επίσης να μετρηθεί στο pin  $A_{ref}$  με βολτόμετρο υψηλής αντίστασης.

Εάν χρησιμοποιηθεί εξωτερική πηγή σταθερής τάσης ως τάση αναφοράς τότε δεν μπορούν να χρησιμοποιηθούν οι άλλες επιλογές τάσης αναφοράς. Εάν δεν εφαρμόζεται εξωτερική τάση στο pin  $A_{ref}$ , ο χρήστης μπορεί να αλλάξει μεταξύ  $AV_{CC}$  και 1.1V ως επιλογή αναφοράς. Το πρώτο αποτέλεσμα μετατροπής ADC μετά την αλλαγή της πηγής τάσης αναφοράς ενδέχεται να είναι ανακριβές και συνιστάται να μη χρησιμοποιηθεί.

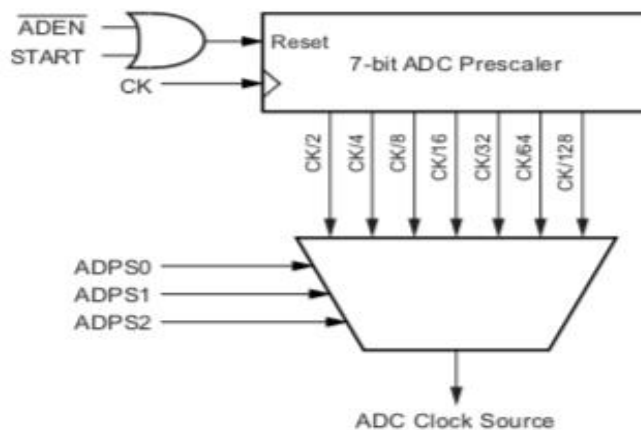


Εικόνα 16 - Κύκλωμα ADC του AtMega328p

Το κύκλωμα διαδοχικών προσεγγίσεων απαιτεί συχνότητα ρολογιού εισόδου μεταξύ 50kHz και 200kHz για τη μέγιστη ανάλυση. Εάν απαιτείται χαμηλότερη ανάλυση από 10 bits, η συχνότητα ρολογιού εισόδου στο ADC μπορεί να είναι υψηλότερη από 200kHz για να λάβει υψηλότερο ρυθμό δείγματος.

Ο μετατροπέας ADC περιέχει ένα prescaler (εικόνα 17), το οποίο δημιουργεί μια αποδεκτή συχνότητα ρολογιού για το ADC από οποιαδήποτε συχνότητα ρολογιού της CPU άνω των 100kHz.

Ο χρόνος μετατροπής για μια κανονική μετατροπή είναι 13 κύκλοι ρολογιού ADC. Για την πρώτη όμως μετατροπή μετά την ενεργοποίηση του ADC απαιτούνται 25 κύκλοι.



Εικόνα 17 - Κύκλωμα prescale

#### 4.2.3. Σφάλματα Μετατροπών

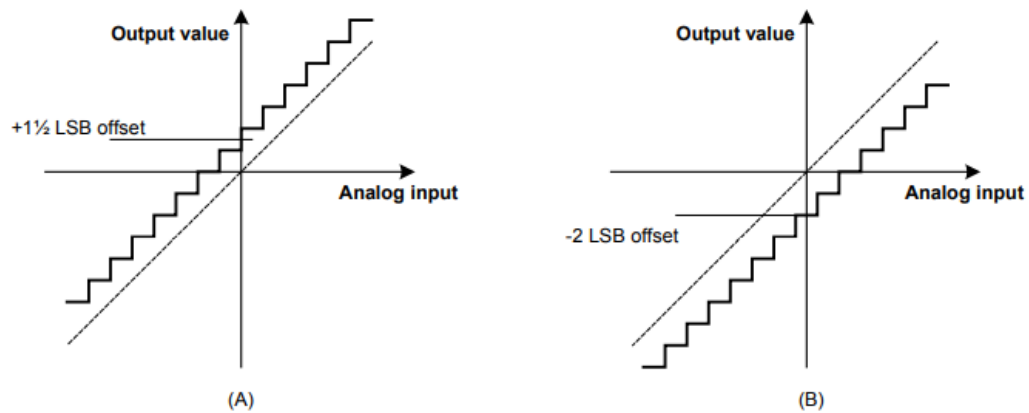
Το συνολικό σφάλμα ενός πραγματικού ADC προκύπτει όχι μόνο από το σφάλμα κβαντισμού. Μεγάλο ρόλο παίζουν και τα σφάλματα αντιστάθμισης (offset) και κέρδους (gain). Το bit ελάχιστης αξίας LSB (Least Significant Bit) που αναφέρεται στη συνέχεια εκφράζει τη μικρότερη δυνατή τιμή τάσης που μπορεί να μετατρέψει σε ψηφιακή τιμή το ADC. [14].

##### 4.2.3.1. Απόλυτο Σφάλμα (Absolute Error)

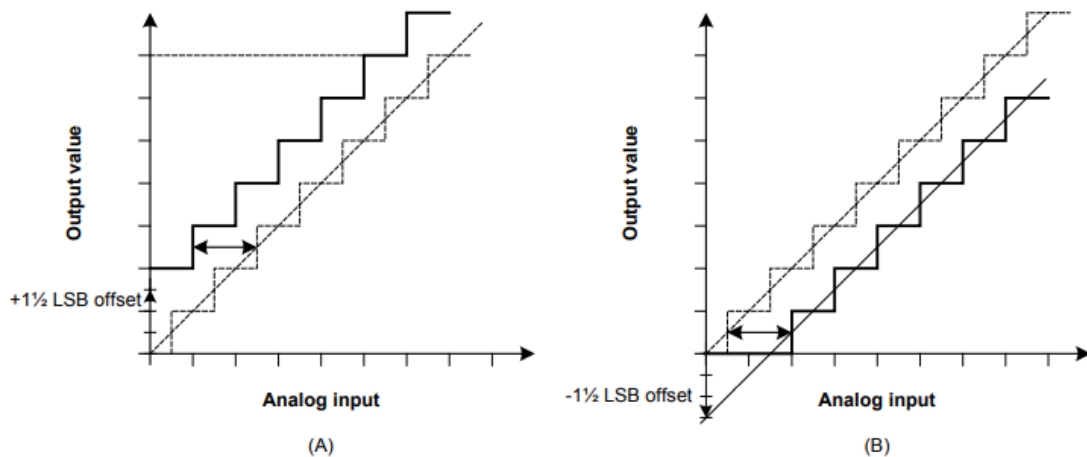
Το απόλυτο σφάλμα είναι η μέγιστη απόκλιση μεταξύ της ευθείας γραμμής ενός ιδανικού ADC και της συνάρτησης μεταφοράς ενός πραγματικού, συμπεριλαμβανομένου των βημάτων κβαντισμού. Έτσι το ελάχιστο απόλυτο σφάλμα είναι  $\frac{1}{2}$  LSB (Least Significant Bit) λόγω του κβαντισμού. Απόλυτο σφάλμα ή απόλυτη ακρίβεια είναι το συνολικό μη σταθμισμένο σφάλμα και περιλαμβάνει το σφάλμα κβαντισμού, αντιστάθμισης, κέρδους και μη γραμμικότητας [14].

#### 4.2.3.2. Σφάλμα Αντιστάθμισης (Offset Error)

Το σφάλμα αντιστάθμισης ορίζεται ως η απόκλιση της συνάρτησης μεταφοράς ενός πραγματικού ADC από την ευθεία γραμμή του ιδανικού στα 0V[14].



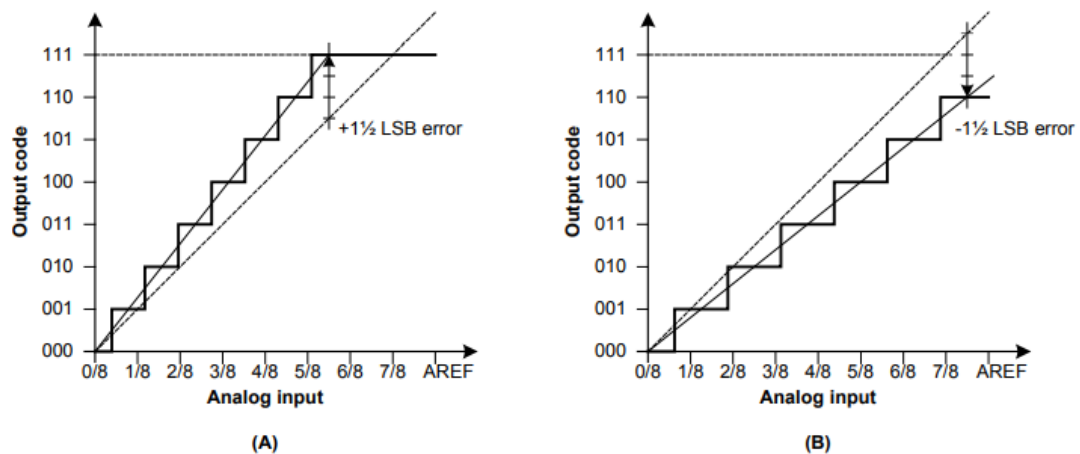
Εικόνα 18 - Παράδειγμα Α) Θετικού Σφάλματος Αντιστάθμισης, Β) Αρνητικού Σφάλματος Αντιστάθμισης



Εικόνα 19 - Α) Θετικό και Β) Αρνητικό Σφάλμα Αντιστάθμισης στη Λειτουργία Απλής Μετατροπής

#### 4.2.3.3. Σφάλμα Κέρδους (Gain Error)

Το σφάλμα κέρδους ορίζεται ως η απόκλιση της εξόδου του μέσου του τελευταίου βήματος από την ευθεία γραμμή ενός ιδανικού ADC, αφού έχει σταθμιστεί το σφάλμα αντιστάθμισης [14].



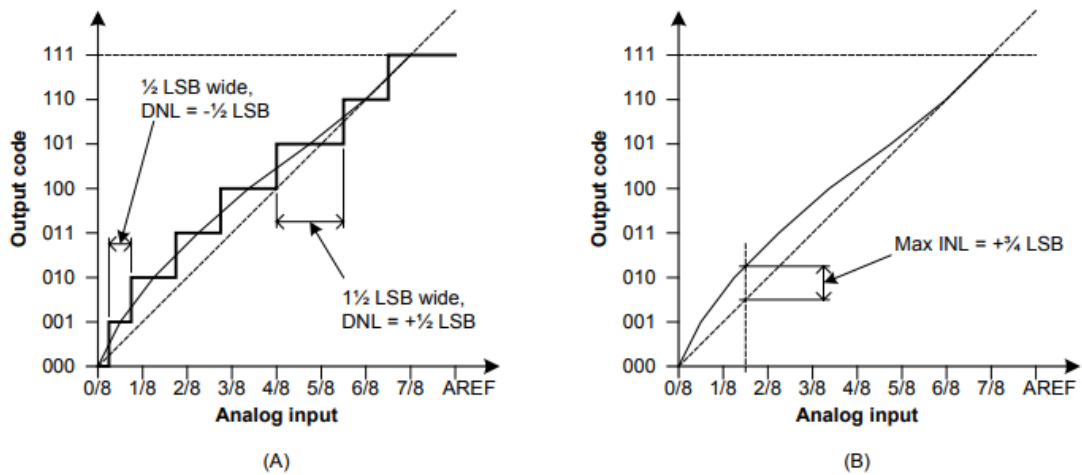
Εικόνα 20 - Παράδειγμα Α) Θετικού και Β) Αρνητικού Σφάλματος Κέρδους

#### 4.2.3.4. Μη – Γραμμικότητα (Non – linearity)

Όταν τα σφάλματα αντιστάθμισης και κέρδους έχουν σταθμιστεί, η πραγματική συνάρτηση μεταφοράς θα πρέπει να ισούται με αυτή ενός τέλει ADC. Ωστόσο, η μη-γραμμικότητα σε έναν ADC μπορεί να προκαλέσει την πραγματική καμπύλη να αποκλίνει από την τέλεια καμπύλη, ακόμα και αν αυτές οι δύο είναι ίσες κοντά στο 0 και στο σημείο που μετρήθηκε το σφάλμα κέρδους. Υπάρχουν δύο μέθοδοι για τη μέτρηση της μη-γραμμικότητας [14].

Η Διαφορική Μη-Γραμμικότητα (Differential Non-Linearity ή DNL) ορίζεται ως η μέγιστη και η ελάχιστη διαφορά μεταξύ του πλάτους βήματος και του τέλει πλάτους (1 LSB) σε οποιοδήποτε βήμα εξόδου. Η μη-γραμμικότητα παράγει βήματα κβαντισμού με ποικίλα πλάτη. Όλα τα βήματα θα έπρεπε να έχουν πλάτος 1 LSB αλλά μερικά είναι στενότερα ή πλατύτερα [14].

Η Ολοκληρωτική μη Γραμμικότητα (Integral Non-Linearity ή INL) ορίζεται ως η μέγιστη κάθετη διαφορά μεταξύ της πραγματικής από την τέλεια καμπύλη. Η INL μπορεί να ερμηνευτεί ως ένα άθροισμα DNL. Αρνητικά INL υποδηλώνουν ότι η πραγματική καμπύλη είναι κάτω από την τέλεια [14].

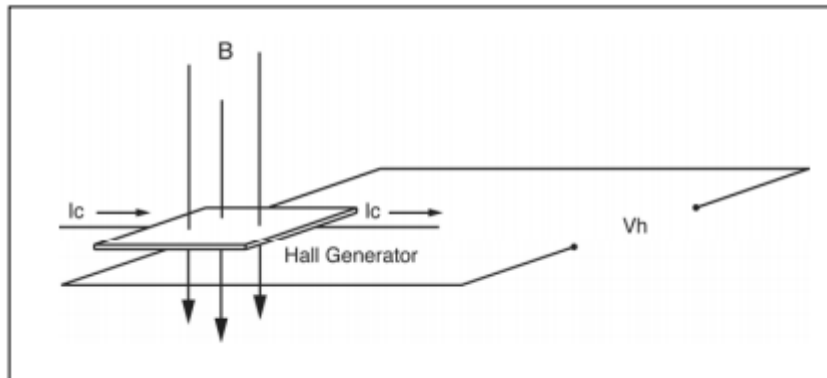


Εικόνα 21 - Παραδείγματα Α) Διαφορικής και Β) Ολοκληρωτικής Μη-Γραμμικότητας

### 4.3. FW Bell 5180 Gaussmeter

#### 4.3.1. Μέτρηση Πυκνότητας Μαγνητικής Ροής (Magnetic Flux Density)

Μία συσκευή που χρησιμοποιείται συχνά για τη μέτρηση της πυκνότητας της μαγνητικής ροής είναι ο αισθητήρας Hall. Ο αισθητήρας Hall είναι μία λεπτή φέτα από ένα ημιαγωγικό υλικό στο οποίο ενσωματώνονται τέσσερις ακροδέκτες, ένας στο μέσο της κάθε πλευράς, όπως φαίνεται και στην εικόνα 22 [15].



Εικόνα 22 - Αισθητήρας Hall

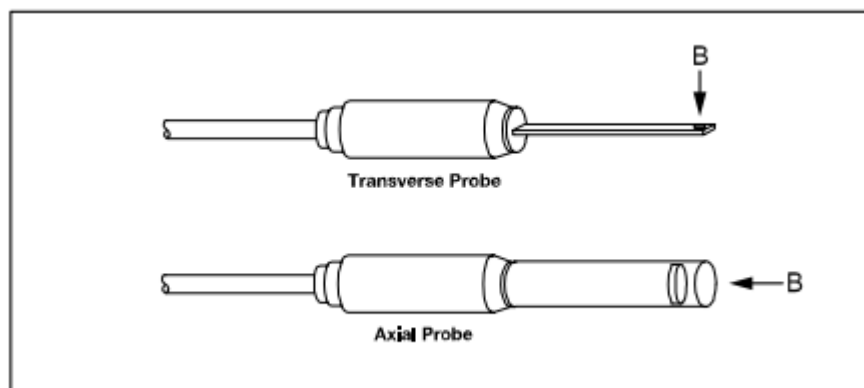
Ένα σταθερό ρεύμα ( $I_c$ ) περνάει μέσα από το υλικό. Σε μηδενικό μαγνητικό πεδίο δεν υπάρχει διαφορά δυναμικού μεταξύ των άλλων δύο ακροδεκτών. Όταν οι μαγνητικές γραμμές περνάν μέσα από το υλικό τότε το ρεύμα έχει την τάση να «γέρνει» προς τη μία πλευρά, δημιουργώντας μία διαφορά δυναμικού γνωστή και ως τάση Hall ( $V_h$ ). Σε ένα ιδανικό αισθητήρα Hall υπάρχει γραμμική συσχέτιση

μεταξύ του αριθμού των μαγνητικών γραμμών που διαπερνούν το υλικό (πυκνότητα ροής) και της τάσης Hall.

Η τάση Hall είναι επίσης συνάρτηση της κατεύθυνσης με την οποία οι μαγνητικές γραμμές διαπερνούν το υλικό, παράγοντας έτσι θετική τάση από τη μία κατεύθυνση και αρνητική από την άλλη. Αν ο ίδιος αριθμός μαγνητικών γραμμών που περνούν μέσα από το υλικό είναι ίσος από τη μία και από την άλλη κατεύθυνση τότε θα έχουμε μηδενική τάση. Αυτή η ευαισθησία στην κατεύθυνση της ροής επιτρέπει τη μέτρηση τόσο συνεχούς όσο και εναλλασσόμενου μαγνητικού πεδίου [15].

Αυτή η τάση είναι επιπρόσθετα συνάρτηση της γωνίας με την οποία οι γραμμές της μαγνητικής ροής διαπερνούν το υλικό. Η μεγαλύτερη τάση προκύπτει όταν οι γραμμές διαπερνούν κάθετα το υλικό. Σε αντίθετη περίπτωση η έξοδος σχετίζεται με το συνημίτονο της διαφοράς των 90 μοιρών με τη γωνία των γραμμών.

Η περιοχή ευαισθησίας ενός αισθητήρα Hall ορίζεται ως η μεγαλύτερη κυκλική περιοχή εντός της φέτας του υλικού. Αυτή η ενεργή περιοχή μπορεί να έχει ένα εύρος από 0.2 mm μέχρι 19 mm σε διάμετρο. Συχνά ο αισθητήρας Hall είναι πολύ ευαίσθητος για να χρησιμοποιηθεί από μόνος του και γι' αυτό ενσωματώνεται σε ένα προστατευτικό πλαίσιο και τερματίζει σε ένα ευλύγιστο καλώδιο και στον αντάπτορα για τη σύνδεση. Αυτή η διάταξη είναι γνωστή και ως Hall probe και παρέχεται σε δύο μορφές [15].



Εικόνα 23 - Hall probes

Στην «εγκάρσια» (transverse) διάταξη ο αισθητήρας Hall τοποθετείται σε ένα λεπτό επίπεδο στέλεχος ενώ στην «αξονική» (axial) διάταξη ο αισθητήρας τοποθετείται πάνω σε ένα κυλινδρικό. Ο άξονας ευαισθησίας είναι η κύρια διαφορά μεταξύ τους. Γενικά η εγκάρσια διάταξη χρησιμοποιείται για τη μέτρηση

### 4.3.2. Σύνοψη Περιγραφή

Τέλος υπάρχει η δυνατότητα το όργανο να ρυθμιστεί πλήρως καθώς επίσης να ληφθούν οι μετρήσεις μέσω ενός ηλεκτρονικού υπολογιστή κάνοντας χρήση της σειριακής επικοινωνίας.



38

### 4.3.3. Επεξήγηση Κώδικα C++

Για την παρούσα μελέτη αρχικά υλοποιήθηκε ένα πρόγραμμα στη γλώσσα C++ προκειμένου να επιτευχθεί η επικοινωνία του υπολογιστή με το 5180. Με αυτόν τον τρόπο είναι δυνατή η καταγραφή της μέτρησης του οργάνου απευθείας χωρίς να πρέπει να ληφθεί μέσω της αναλογικής εξόδου του.

Ο λόγος για τον οποίο τελικά δεν επιλέχθηκε η συγκεκριμένη υλοποίηση έγκειται στο γεγονός ότι δεν υπήρχε συμβατότητα με τον υπολογιστή όπου λαμβάνουν χώρα οι μετρήσεις. Παρ' όλα αυτά σε κάποια μελλοντική αναβάθμιση του υπολογιστή είναι πιθανό το παρακάτω πρόγραμμα να φανεί βοηθητικό για τις μετρήσεις.

Αρχικά για την σειριακή επικοινωνία του υπολογιστή με το γκαουσόμετρο θα πρέπει να γίνει εγκατάσταση του λογισμικού με τα προγράμματα οδήγησης που προσφέρει η εταιρεία FW Bell, τα οποία είναι διαθέσιμα στο διαδίκτυο. Στη συνέχεια θα πρέπει να ληφθούν και οι δυναμικές βιβλιοθήκες της εταιρείας με τη χρήση των οποίων γίνεται δυνατή η επικοινωνία του οργάνου με τον υπολογιστή. Πιο συγκεκριμένα αυτές οι βιβλιοθήκες παρέχονται στο αρχείο usb5100.dll και libusb0.dll τα οποία εγκαθίστανται μαζί με το λογισμικό και βρίσκονται στο φάκελο εγκατάστασής του.

Η σύνταξη του κώδικα έγινε σε περιβάλλον Windows 10 Home 64 bit με χρήση του editor CodeBlocks για την επεξεργασία του και η μεταγλώττιση του προγράμματος σε εκτελέσιμο αρχείο πραγματοποιήθηκε με τον compiler MinGw g++ version 4.9.2 των 32-bit ο οποίος εγκαταστάθηκε αυτόματα με τη λήψη του CodeBlocks.

```
#include <windows.h>
#include <stdio.h>
#include <iostream>
#include <fstream>
#include <ctype.h>
#include <algorithm>
using namespace std;

typedef int (__stdcall *f_func)();
typedef int (__stdcall *func_ptr)(unsigned int, char*, char*, int);
typedef void (__stdcall *func_close)(unsigned int);
unsigned int id;
HINSTANCE hGetProcIDDL = LoadLibrary("F:\\Program Files (x86)\\FW Bell\\PC5180\\usb5100.dll");
f_func open = (f_func)GetProcAddress(hGetProcIDDL, "openUSB5100");
func_ptr data = (func_ptr)GetProcAddress(hGetProcIDDL, "scpiCommand");
func_close close = (func_close)GetProcAddress(hGetProcIDDL, "closeUSB5100");
```

Εικόνα 25 - Κομμάτι κώδικα C++ (α)

Αρχικά ορίζονται οι βιβλιοθήκες που είναι απαραίτητες για την σύνταξη του κώδικα. Στη συνέχεια στα τρία typedef κατασκευάζονται οι παράμετροι για την κλήση των συναρτήσεων που βρίσκονται στη δυναμική βιβλιοθήκη usb5100.dll.



Ακριβώς από κάτω η μεταβλητή `id` θα χρησιμοποιηθεί ως μεταβλητή αναφοράς για την επικοινωνία με το όργανο. Ακριβώς από κάτω καλείται η δυναμική βιβλιοθήκη `usb5100.dll`. Εδώ αξίζει να σημειωθεί ότι η διεύθυνση του αρχείου είναι ενδεικτική και θα πρέπει να αντικαθίσταται με τη διεύθυνση που βρίσκεται το συγκεκριμένο αρχείο. Εν συνεχεία ορίζονται οι συναρτήσεις `open`, η οποία ξεκινάει την επικοινωνία με το όργανο, `data` με την οποία δίνουμε την εντολή με την κατάλληλη μορφή προκειμένου να επιτύχουμε μια συγκεκριμένη λειτουργία και η `close` όπου στο τέλος του προγράμματος καλείται με σκοπό να τερματιστεί η σειριακή επικοινωνία.

```
void openusb() {  
  
void closeusb() {  
  
char* readdata() {  
    char* ac;  
    data(id, (char*) ":MEASURE:FLUX?", ac, 80);  
    return ac;  
}  
  
string returnNumber(char* a) {  
    string b(a);  
    b.erase(remove(b.begin(), b.end(), 'G'), b.end());  
    b.erase(remove(b.begin(), b.end(), ' '), b.end());  
    return b;  
}
```

Εικόνα 26 - Κομμάτι κώδικα C++ (β)

Οι συναρτήσεις `openusb` και `closeusb` ουσιαστικά καλούν τις συναρτήσεις `open` και `close` που ορίστηκαν πιο πάνω. Η `openusb` μετά την κλήση της `open` επιστρέφει τη μεταβλητή `id`. Με τη `readdata` δίνεται εντολή στο όργανο να μετρήσει την πυκνότητα της μαγνητικής ροής και στη συνέχεια να την επιστρέψει στο πρόγραμμα, ενώ η `returnNumber` καθαρίζει το αποτέλεσμα της `readdata` από κενά και τον χαρακτήρα `G` προκειμένου να μείνει μόνο ο αριθμός για να μπορεί να χρησιμοποιηθεί από το πρόγραμμα των μετρήσεων `MultiVu`.

```

string lastf()
{
    WIN32_FIND_DATAW ffd;
    wchar_t const* directory = L"C:\\QdVersaLab\\Data\\";
    wchar_t currentFile[MAX_PATH], lastModifiedFilename[MAX_PATH];
    FILETIME currentModifiedTime, lastModified;
    HANDLE hFile;
    bool first_file = true;

    HANDLE hFind = FindFirstFileW( L"C:\\QdVersaLab\\Data\\*", &ffd );

    if ( INVALID_HANDLE_VALUE == hFind )
    {
        cout<<"cant find path";
    }

    do
    {
        while ( FindNextFileW( hFind, &ffd ) != 0 );

        FindClose( hFind );
        wstring ws(lastModifiedFilename);
        wstring s(directory);
        // your new String
        string str1(s.begin(), s.end());
        string str2(ws.begin(), ws.end());
        string filename=str1+str2;
        return filename;
    }
}

```

Εικόνα 27 - Κομμάτι κώδικα C++ (γ)

Τελευταία συνάρτηση που θα χρησιμοποιηθεί είναι η lastf. Ο ρόλος που διαδραματίζει στο πρόγραμμα είναι αρχικά να επιλεγεί η τοποθεσία που βρίσκονται τα αρχεία των μετρήσεων που εξάγονται από την εφαρμογή MultiVu. Στη συνέχεια πραγματοποιεί σειριακή αναζήτηση όλων των αρχείων που βρίσκονται στη συγκεκριμένη διεύθυνση και επιστρέφει το όνομα του αρχείου το οποίο δημιουργήθηκε τελευταίο. Και εδώ η διεύθυνση που χρησιμοποιήθηκε είναι ενδεικτική και θα πρέπει να αλλαχτεί με αυτή που επιλέγει ο χρήστης για να εξάγονται τα αρχεία των μετρήσεων.

Προχωρώντας στο κυρίως πρόγραμμα ξεκινάει στην αρχή η σειριακή επικοινωνία με το 5180. Εφόσον αυτό καταστεί δυνατό τότε συνεχίζει η ροή του προγράμματος ενώ σε αντίθετη περίπτωση εμφανίζεται μήνυμα λάθους και τερματίζεται η λειτουργία του. Αμέσως μετά καλείται η συνάρτηση lastf για την εύρεση του αρχείου που θα λάβει χώρο η μέτρηση και ανοίγει το αρχείο μαζί με ένα logFile το οποίο θα χρησιμοποιηθεί ως μέσο για τον έλεγχο της ροής της μέτρησης. Στη συνέχεια εισάγεται ένας ατέρμον βρόγχος μέσα στον οποίο το πρόγραμμα διαβάζει το αρχείο των μετρήσεων μέχρι να προστεθεί από την εφαρμογή MultiVu μία νέα γραμμή που θα δηλώνει και την ύπαρξη μέτρησης. Με το που ανιχνευθεί η προσθήκη καλείται η συνάρτηση readdata και η returnNumber προκειμένου το γκαουσόμετρο να λάβει την τιμή του πεδίου και αυτή να επιστραφεί ως καθαρός αριθμός στο κυρίως πρόγραμμα. Τέλος

αντιγράφεται η γραμμή που προστέθηκε στο αρχείο των μετρήσεων, προστίθεται στην πρώτη κενή θέση αυτής η τιμή που λήφθηκε από το 5180 και ξαναστέλνεται πίσω στο αρχείο στη θέση της παλιάς. Όλη αυτή η διαδικασία συνεχίζει μέχρις ότου στο log αρχείο να δοθεί μια ένδειξη.

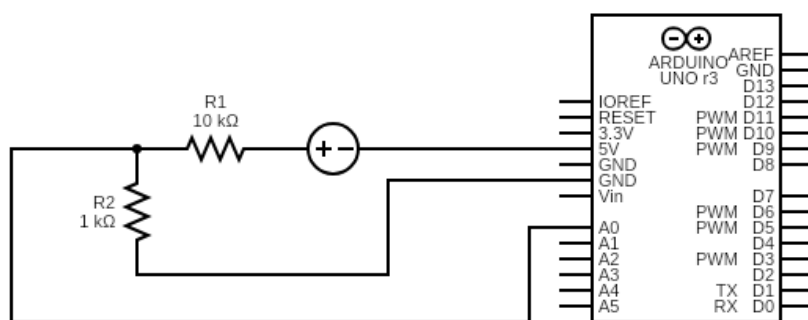
Με την ολοκλήρωση της μέτρησης και την έξοδο του προγράμματος από το βρόγχο τερματίζει η επικοινωνία με την κλίση της συνάρτησης closeusb και τερματίζεται το πρόγραμμα.

#### 4.4. Προγραμματισμός Arduino – Προετοιμασία Μέτρησης

##### 4.4.1. Διαιρέτης τάσης

Η έξοδος του οργάνου 5180 παράγει τάση εύρους -5 έως 5 V. Συνεπώς δεν είναι δυνατή η ανάγνωσή της από την αναλογική είσοδο του Arduino χωρίς να υποστεί κάποια επεξεργασία. Για το λόγο αυτό ακολουθούν κάποιες μετατροπές.

Αρχικά για την εξάλειψη των αρνητικών τάσεων εισάγεται στο κύκλωμα επιπλέον τάση της τάξης των 5V. Η συγκεκριμένη τάση εισάγεται από την έξοδο 5V του Arduino. Αυτό οδηγεί στη μετάβαση από -5 έως 5 V σε ένα εύρος 0 – 10 V. Με αυτή την αναγωγή προκύπτει ένα άλλο ζήτημα στη μέτρηση της τάσης από την αναλογική είσοδο του Arduino. Επειδή ο ADC έχει ως μέγιστη τάση αναφοράς τα 5V οποιαδήποτε τιμή πάνω από αυτή δεν είναι δυνατό να μετρηθεί. Έτσι θα πρέπει να αλλάξει η κλίμακα εισόδου προκειμένου να είναι δυνατή η μέτρηση αυτής της τάσης. Συνεπώς προκύπτει η ανάγκη εισαγωγής ενός κυκλώματος διαιρέτη τάσης με σκοπό τη μείωση της μέγιστης τάσης εισόδου.



Εικόνα 28 - Κύκλωμα διαιρέτη τάσης

Για την υλοποίηση του διαιρέτη τάσης επιλέγονται δύο αντιστάσεις, η μία στα 1kΩ και η δεύτερη στα 10kΩ συνδεδεμένες σε σειρά και οι δύο αντοχής 1%. Η μέτρηση της τάσης θα πραγματοποιηθεί στην αντίσταση των 10kΩ. Αυτό μας δίνει ένα

εύρος τάσης εισόδου ( $10V * \frac{1k\Omega}{1k\Omega+10k\Omega} = 0.909 V$ )  $0 - 0.909 V$ . Η αναγωγή σε κλίμακα  $0 - 0.9 V$  είναι προτιμότερη σε σχέση με την κλίμακα  $0 - 5V$  διότι με αυτόν τον τρόπο η τάση αναφοράς του ADC επιλέγεται στα  $1 V$  με αποτέλεσμα τη μείωση της ακρίβειας ή αλλιώς του σφάλματος κατά τη μετατροπή. Στην εικόνα 28 η πηγή τάσης αντιπροσωπεύει την έξοδο του FW Bell 5180.

#### 4.4.2. Oversampling and Decimation

Το ADC του Arduino Uno όπως αναφέρθηκε και προηγουμένως προσφέρει ανάλυση 10-bit. Η Atmel προτείνει τη μέθοδο oversampling and decimation για τη βελτίωση της ανάλυσης του ADC μέσω software χωρίς τη χρήση κάποιου εξωτερικού ADC. Η μέθοδος αυτή όμως περιορίζεται από τα χαρακτηριστικά του ADC και μπορεί να μειώσει μόνο το σφάλμα κβαντισμού, ενώ δεν μπορεί να σταθμίσει τις μη γραμμικότητες αυτού [16].

Σύμφωνα με το θεώρημα Nyquist για να ανακατασκευάσουμε πλήρως ένα σήμα από την ελλιπόμενη μορφή του πρέπει η συχνότητα δειγματοληψίας να είναι μεγαλύτερη ή ίση από το διπλάσιο της μέγιστης συχνότητάς τους ( $f_{signal}$ ). Σε διαφορετική περίπτωση, δηλαδή για συχνότητες δειγματοληψίας μικρότερες αυτής του θεωρήματος στο ανακατασκευασμένο σήμα, μπορεί να εμφανίζεται το φαινόμενο της αναδίπλωσης συχνότητας [16].

$$f_{nyquist} > 2 * f_{signal} \quad (4.1)$$

Όταν χρησιμοποιούνται συχνότητες δειγματοληψίας σημαντικά μεγαλύτερης της  $f_{nyquist}$  τότε έχουμε υπερδειγματοληψία του σήματος (oversampling). Με τη χρήση της υπερδειγματοληψίας έχουμε καλύτερη αναπαράσταση του σήματος στο πεδίο του χρόνου.

Στο Arduino όπως αναφέρθηκε και προηγουμένως η συχνότητα δειγματοληψίας είναι αποτέλεσμα του prescaling της συχνότητας του ρολογιού της CPU. Με μικρότερο συντελεστή prescaling πετυχαίνουμε μεγαλύτερη συχνότητα στο ρολόι του ADC. Όμως η αύξηση της συχνότητας του ρολογιού του ADC έχει ως συνέπεια τη μείωση της ακρίβειας του ADC διότι υπάρχει μείωση του αριθμού των αποτελεσματικών bit (Effective Number Of Bits ή ENOB) του ADC. Όλοι οι ADC έχουν περιορισμένο εύρος συχνοτήτων. Σύμφωνα με το datasheet για τον ADC της Avr με τα 10 αποτελεσματικά bit η συχνότητα του ρολογιού του ADC πρέπει να είναι στο εύρος 50kHz-200kHz. Η συχνότητα του ρολογιού μπορεί να φτάσει μέχρι και το 1MHz με αποτέλεσμα ωστόσο την μείωση των αποτελεσματικών bit ADC.

Με την μέθοδο oversampling and decimation μπορεί να επιτευχθεί η αύξηση των bit του ADC και της ανάλυσής του με τη χρησιμοποίηση μεγαλύτερου αριθμού δειγμάτων μέσω της υπερδειγματοληψίας του σήματος. Για κάθε επιπλέον bit ανάλυσης που προστίθεται στον ADC η συχνότητα δειγματοληψίας του σήματος πρέπει να είναι 4 φορές μεγαλύτερη της  $f_{nyquist}$ .

Εξίσωση συχνότητας υπερδειγματοληψίας [16]:

$$f_{oversampling} = 4^n * f_{nyquist} \quad (4.2)$$

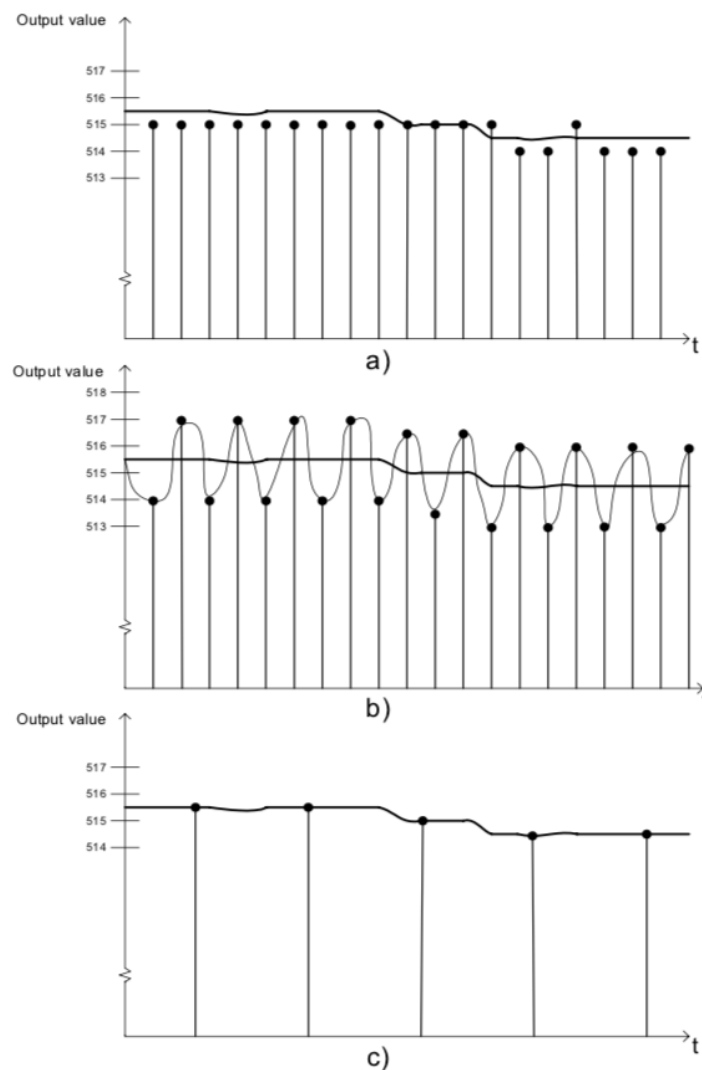
Ο αυξημένος ρυθμός δειγματοληψίας δεν βελτιώνει άμεσα την ανάλυση ADC, αλλά παρέχοντας περισσότερα δείγματα αυτή η τεχνική παρακολουθεί με μεγαλύτερη ακρίβεια το σήμα εισόδου χρησιμοποιώντας καλύτερα το υπάρχον δυναμικό εύρος ADC. Η υπερδειγματοληψία από μόνη της βελτιώνει την ψηφιακή αναπαράσταση του σήματος μόνο μέχρι το όριο φυσικής δυναμικής περιοχής (ελάχιστο μέγεθος βήματος) του ADC.

Για τη σωστή λειτουργία της μεθόδου oversampling and decimation το σήμα δεν πρέπει να μεταβάλλεται σημαντικά κατά τη διάρκεια της μετατροπής από αναλογικό σε ψηφιακό σήμα. Παρ' όλα αυτά μια μικρή μεταβολή μερικών LSB είναι απαραίτητη. Η διακύμανση αυτή πρέπει να θεωρηθεί ως το στοιχείο θορύβου του σήματος. Κατά την υπερδειγματοληψία ενός σήματος, θα πρέπει να υπάρχει θόρυβος για την ικανοποίηση αυτής της ζήτησης μικρών παραλλαγών στο σήμα. Το σφάλμα κβαντισμού του ADC είναι τουλάχιστον 0,5 LSB. Επομένως, το πλάτος θορύβου πρέπει να υπερβαίνει τα 0,5 LSB για εναλλαγή του LSB. Το πλάτος θορύβου 1-2 LSB είναι ακόμη καλύτερο, διότι αυτό θα διασφαλίσει ότι πολλά δείγματα δεν θα καταλήξουν να έχουν την ίδια τιμή [16].

Συνήθως κατά τη διάρκεια μιας μετατροπής υπάρχει κάποιος θόρυβος. Ο θόρυβος μπορεί να είναι θερμικός θόρυβος, θόρυβος από τον πυρήνα της CPU, εναλλαγή θυρών I / O, διακυμάνσεις στην τροφοδοσία και άλλα. Για την σωστή λειτουργία της μεθόδου, αυτός ο θόρυβος στις περισσότερες περιπτώσεις θα είναι αρκετός. Ο θόρυβος αυτός έχει χαρακτηριστικά του λευκού θορύβου, δηλαδή χαρακτηρίζεται από μεγάλο φάσμα συχνοτήτων και σταθερή πυκνότητα φασματικής ισχύος. Σε αυτήν την περίπτωση η μέθοδος λειτουργεί σωστά εάν το πλάτος του θορύβου είναι αρκετό για εναλλαγή του LSB της μετατροπής στον ADC. Ωστόσο, σε συγκεκριμένες περιπτώσεις ενδέχεται να είναι απαραίτητη η προσθήκη τεχνητού θορύβου στο σήμα εισόδου. Η μέθοδος με την προσθήκη τεχνητού θορύβου αναφέρεται ως Dithering. Η εικόνα 29 (α) δείχνει το πρόβλημα της μέτρησης ενός σήματος με τιμή τάσης που είναι μεταξύ δύο σταδίων

κβαντισμού. Ο μέσος όρος τεσσάρων δειγμάτων δεν θα βοηθούσε, καθώς η ίδια χαμηλή τιμή θα ήταν το αποτέλεσμα. Μπορεί να βοηθήσει μόνο στη μείωση της διακύμανσης του σήματος. Η εικόνα 29 (b) δείχνει ότι με την προσθήκη κάποιου τεχνητού θορύβου στο σήμα εισόδου το LSB του αποτελέσματος μετατροπής θα εναλλάσσεται. Η προσθήκη τεσσάρων από αυτά τα δείγματα στα μισά στάδια κβαντισμού, παράγουν αποτελέσματα που δίνουν καλύτερες αναπαραστάσεις της τιμής εισόδου, όπως φαίνεται στην εικόνα 29 (c). Η «εικονική ανάλυση» των ADC αυξήθηκε από 10 σε 11 bits. Αυτή η μέθοδος αναφέρεται ως Decimation.

Ένας άλλος λόγος για τη χρήση αυτής της μεθόδου είναι η αύξηση του λόγου Signal to Noise(SNR). Η ENOB θα εξαπλώσει τον θόρυβο σε μεγαλύτερο δυαδικό αριθμό και η επίδραση των θορύβων σε κάθε δυαδικό ψηφίο θα μειωθεί. Ο διπλασιασμός της συχνότητας δειγματοληψίας θα μειώσει τον θόρυβο της ζώνης κατά 3dB και θα αυξήσει την ανάλυση της μέτρησης κατά 0,5 bit [16].



Εικόνα 29 - Η διαδικασία του Decimation

Ένας απλός και αποτελεσματικός τρόπος εξομάλυνσης της κυματομορφής εισόδου και μείωσης του θορύβου είναι ο απλός μέσος όρος των δειγμάτων. Όμως με τον απλό μέσο ορό δεν αυξάνεται ανάλυση της μετατροπής. Για την βελτίωση της ανάλυσης πρέπει να χρησιμοποιηθεί άλλη μέθοδος όπως η Decimation, όπου σε συνδυασμό με την υπερδειγματοληψία προσφέρει το επιθυμητό αποτέλεσμα [16]. Τα επιπλέον δείγματα,  $m$ , που επιτυγχάνονται με υπερδειγματοληψία του σήματος προστίθενται, όπως και στην κανονική μέση τιμή, αλλά το αποτέλεσμα δεν διαιρείται με  $m$  όπως στην κανονική μέση τιμή. Αντίθετα, στο αποτέλεσμα εφαρμόζονται  $n$  δεξιές ολισθήσεις, όπου  $n$  είναι το επιθυμητό επιπλέον bit ανάλυσης, για να κλιμακωθεί σωστά η απάντηση. Η δεξιά ολίσθηση ενός δυαδικού αριθμού μία φορά ισούται με τη διαίρεση του δυαδικού αριθμού με έναν συντελεστή 2. Όπως φαίνεται από την εξίσωση (4.2), η αύξηση της ανάλυσης από 10-bit σε 12-bit απαιτεί το άθροισμα των 16 τιμών 10-bit. Ένα άθροισμα 16 τιμών 10-bit δημιουργεί ένα αποτέλεσμα 14-bit όπου τα δύο τελευταία bit δεν αναμένεται να περιέχουν πολύτιμες πληροφορίες. Για να επιστρέψουμε στα 12-bit, είναι απαραίτητο να κλιμακωθεί το αποτέλεσμα. Ο συντελεστής κλίμακας,  $sf$ , που δίνεται από την εξίσωση (4.3), είναι ο παράγοντας με τον οποίο πρέπει να διαιρεθεί το άθροισμα των  $4^n$  δειγμάτων για να γίνει σωστή κλιμάκωση του αποτελέσματος, όπου  $n$  είναι ο επιθυμητός αριθμός επιπλέον bit .

$$sf = 2^n \quad (4.3)$$

Στην εικόνα 30 παρουσιάζονται οι διάφορες επιλογές ανάλυσης που μετρούν την τιμή  $V_{in} = 2.4729V$  όταν  $V_{REF} = 5V$  και για συχνότητα ρολογιού του ADC στα 200kHz.

| Resolution | Voltage resolution | Oversampled | Right shifted | Ideal decimated result | Ideal voltage representation | Maximum Bandwidth      |
|------------|--------------------|-------------|---------------|------------------------|------------------------------|------------------------|
| 10 bit     | ~5 mV              | NA          | NA            | NA                     | 2.4658V                      | ~7600Hz <sup>(1)</sup> |
| 11 bit     | ~2.5 mV            | 4X          | 1X            | 1012                   | 2.4707V                      | ~1900Hz <sup>(1)</sup> |
| 12 bit     | ~1.22mV            | 16X         | 2X            | 2025                   | 2.4719V                      | ~475Hz <sup>(1)</sup>  |
| 13 bit     | ~610 uV            | 64X         | 3X            | 4051                   | 2.4725V                      | ~118Hz <sup>(1)</sup>  |
| 14 bit     | ~300 uV            | 256X        | 4X            | 8103                   | 2.4728V                      | ~29Hz <sup>(1)</sup>   |
| 15 bit     | ~150 uV            | 1024X       | 5X            | 16206                  | 2.4728V                      | ~7Hz <sup>(1)</sup>    |
| 16 bit     | ~75 uV             | 4096X       | 6X            | 32413                  | 2.4729V                      | ~3Hz <sup>(1)</sup>    |

Notes: 1. ADC Clock = 200kHz

Εικόνα 30 - Οι αναλύσεις του Adc με τη μέθοδο του Oversampling

Συνοψίζοντας, όταν το ADC λαμβάνει ένα σήμα, ποσοτικοποιεί αυτό το σήμα σε διακριτά βήματα. Αυτό εισάγει κάποιο σφάλμα, που συχνά αναφέρεται ως σφάλμα κβαντισμού. Η κανονική μέση τιμή θα εξισορροπήσει μόνο τις διακυμάνσεις του σήματος, ενώ το Decimation θα αυξήσει την ανάλυση.

#### 4.4.3. Βαθμονόμηση του Arduino

Όπως αναφέρθηκε προηγουμένως μια μετατροπή ADC εισάγει διαφορά σφάλματα στη μέτρηση. Τα σφάλματα αυτά μπορούν να μειωθούν με την κατάλληλη βαθμονόμηση του ADC μέσω software. Για την βαθμονόμηση του ADC πραγματοποιήθηκε υπολογισμός των διάφορων σφαλμάτων [14] και αφαίρεσή τους από τη μέτρηση. Για τον υπολογισμό των σφαλμάτων χρησιμοποιήθηκε πολύμετρο ακριβείας με γνωστό σφάλμα.

Για  $V_{ref}$  των μετατροπών του ADC επιλέχθηκε η εσωτερική τάση η οποία δίνεται περίπου 1.1V από το datasheet. Για μεγαλύτερη ακρίβεια πρέπει να μετρηθεί η πραγματική της τιμή με χρήση πολυμέτρου. Η πραγματική τιμή μετριέται ανάμεσα στα pin του  $A_{ref}$  και GND του Arduino. Η τιμή αυτή χρησιμοποιείται στη συνέχεια για μετατροπή της κβαντικής τιμής του ψηφιακού σήματος σε Volt. Για τη μείωση του σφάλματος κβαντισμού χρησιμοποιήθηκε η μέθοδος oversampling and decimation.

Για τον υπολογισμό του offset error εφαρμόζουμε στην είσοδο του ADC τάση ξεκινώντας από 0V και αυξάνουμε μέχρι να πάρουμε την πρώτη τιμή διάφορη του μηδενός στην έξοδο. Η διαφορά πραγματικής τιμής εισόδου και της εξόδου



του ADC είναι το offset error. Στην περίπτωση που για μηδενική είσοδο έχουμε ένδειξη διάφορη του μηδενός θέτουμε αυτή την τιμή σαν offset error. Στη συνέχεια υπολογίζουμε σε ποσά LSB αντιστοιχεί η τιμή του σφάλματος αυτού και το αφαιρούμε από τον κβαντική στάθμη της μετρούμενης τιμής.

Το gain error υπολογίζεται δίνοντας σαν είσοδο του ADC το  $V_{ref}$ , δηλαδή πρέπει να συνδεθεί το pin του  $A_{ref}$  με ένα από τα pin εισόδου του ADC του Arduino. Από την ένδειξη του ADC και την πραγματική τιμή του  $V_{ref}$  υπολογίζουμε τη διόρθωση του σφάλματος αυτού η οποία ορίζεται ως τον λόγο της μετρούμενης τιμής του  $V_{ref}$  προς την πραγματική τιμή του  $V_{ref}$ . Η διόρθωση αυτή πολλαπλασιάζεται με όλες τις μετρούμενες τιμές.

$$Gain\ error\ factor = \frac{measured\ V_{ref}\ value}{Actual\ V_{ref}\ value} \quad (4.4)$$

Τέλος για τον υπολογισμό των σφαλμάτων μη γραμμικότητας εφαρμόζουμε τάση από 0V και αυξάνουμε μέχρι  $V_{ref}$  στην είσοδο του ADC. Από τις ενδείξεις του ADC βλέπουμε αν υπάρχει κάποιο σταθερό σφάλμα σε κάποιο εύρος τιμών της εισόδου ώστε να εφαρμοστεί κάποια αντιστάθμιση για το εύρος αυτό.

Οι τιμές των σφαλμάτων και του  $V_{ref}$  είναι διαφορετικές για κάθε Arduino. Τα σφάλματα που υπολογίστηκαν με τον παραπάνω τρόπο και οι διορθώσεις που εφαρμόστηκαν στον Arduino που χρησιμοποιήθηκε στα πλαίσια αυτής της διπλωματικής είναι τα εξής :

|                 |                 |
|-----------------|-----------------|
| $V_{REF}$       | 1055 mV         |
| Offset Error    | -6              |
| Gain Error      | 1055 mV/1050 mV |
| Πηγή 5V Arduino | 4990 mV         |

Πίνακας 3 - Τιμές διόρθωσης

#### 4.4.4. Επεξήγηση Κώδικα Arduino

Ο προγραμματισμός του Arduino έγινε μέσω του ολοκληρωμένου περιβάλλοντος ανάπτυξης Arduino IDE. Λειτουργεί ως εξής: κάθε φορά που στέλνεται σήμα στη σειριακή θύρα για μέτρηση από τον υπολογιστή ο Arduino πραγματοποιεί μια μέτρηση με τη μέθοδο oversampling and decimation και γράφει την τιμή στη σειριακή θύρα.

Αρχικά ορίζονται οι μεταβλητές n, η οποία ορίζει τα επιπλέον bit ανάλυσης της μεθόδου oversampling and decimation, οι R1 και R2 που είναι οι τιμές των

αντιστάσεων του διαιρέτη τάσης όπως μετρήθηκαν με πολύμετρο και τέλος οι μεταβλητές corr και offset οι οποίες είναι οι διορθώσεις για το gain error και offset error αντίστοιχα.

```
void setup() {  
  // initialize serial comm  
  Serial.begin(9600);  
  analogReference(INTERNAL);  
  for (int k=0; k<6;k++){  
    analogRead(A0);  
  }  
}
```

Εικόνα 31 - Η συνάρτηση void setup() του Arduino

Στη συνάρτηση void setup() γίνεται αρχικοποίηση της σειριακής επικοινωνίας και επιλογή bandwidth. Επίσης επιλέγεται ως Vref για τις μετατροπές του ADC η internal του Arduino. Μετά την αλλαγή της τάσης αναφοράς του ADC προτείνεται στο datasheet να μη χρησιμοποιούνται οι πρώτες τιμές ως ανακριβείς. Για το λόγο αυτό γίνονται έξι μετατροπές οι οποίες δε χρησιμοποιούνται.

```
void loop() {  
  float divfactor=R2/(R1+R2);  
  int n_samples=1<<(2*n);  
  if (Serial.available()>0){  
    char a=Serial.read();  
    Serial.flush();  
    switch(a){  
      case '1':  
        uint32_t oversampled=0;  
        for (int i=1; i<n_samples; i++){  
          oversampled=oversampled+(offseterror+analogRead(A0));  
        }  
        double scaled = (oversampled>>n);  
        int sf=1<<n;  
        double result=corr*scaled*vref/sf;  
        result=(result/divfactor)-varduino;  
        Serial.print(result);  
        break;  
      default:  
        break;  
    }  
  }  
}
```

Εικόνα 32 - Η συνάρτηση void loop() του Arduino

Στη συνάρτηση void loop() εκτελείται η κύρια λειτουργία του προγράμματος. Όταν σταλθεί σήμα για μέτρηση στη σειριακή θύρα ο Arduino το διαβάζει και ξεκινάει η διαδικασία της μέτρησης μιας τιμής. Τα βήματα που ακολουθούνται είναι τα εξής:

- Αρχικά πραγματοποιείται υπερδειγματοληψία στο σήμα. Για  $n = 4$  επιπλέον bit που έχουν επιλεχθεί θα έχουμε 256 δείγματα στα οποία εφαρμόζεται και η διόρθωση για το offset error. Οι κβαντικές τιμές των δειγμάτων αυτών προστίθενται σε μια μεταβλητή oversampled.
- Στη μεταβλητή oversampled εφαρμόζονται  $n$  δεξιές ολισθήσεις.
- Υπολογίζεται η τάση εξόδου του διαιρέτη τάσης πολλαπλασιάζοντας την μεταβλητή με την  $V_{ref}$  και τη διόρθωση για το gain error και διαιρείται με τον συντελεστή κλιμάκωσης  $sf$ .
- Τέλος υπολογίζεται η επιθυμητή τιμή υπολογίζοντας αρχικά την τάση εισόδου του διαιρέτη τάσης και στη συνέχεια από την τάση αυτή αφαιρούνται τα 5V που προστέθηκαν ώστε να έχουμε το επιθυμητό εύρος εισόδου στον ADC.

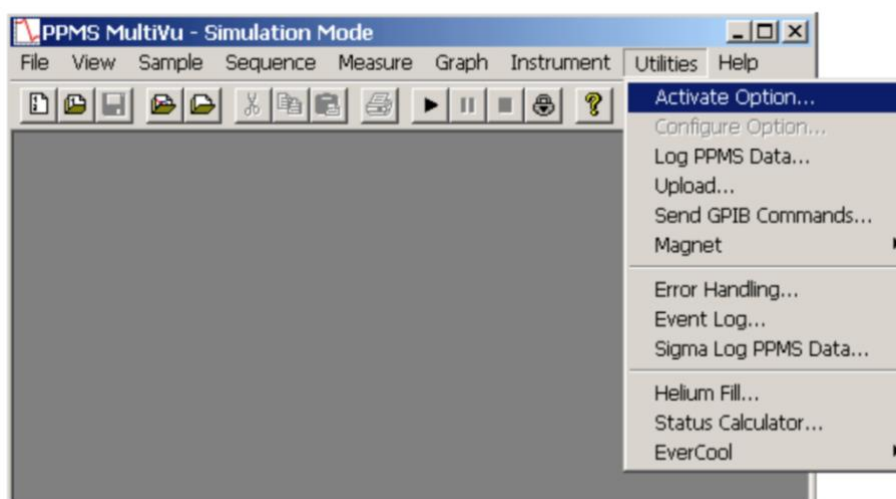
## 5. Προγραμματισμός και Διεξαγωγή Μετρήσεων

Στη διαδικασία της μέτρησης συμμετέχουν η διάταξη του ηλεκτρομαγνήτη με το VSM όπως περιγράφηκαν προηγουμένως σε συνδυασμό με το Gaussmeter και το Arduino. Το VSM το οποίο πραγματοποιεί τη μέτρηση της μαγνητικής ροπής ελέγχεται από το λογισμικό MultiVu της Qd μέσω του οποίου καθορίζεται η μέτρηση που θα γίνει. Για την εισαγωγή των μετρήσεων του πεδίου που γίνονται μέσω του Arduino και του 5180 δημιουργήθηκε το πρόγραμμα `in_vsm` σε γλώσσα προγραμματισμού Python το οποίο επικοινωνεί με το Arduino ώστε να καταγράψει τις τιμές του πεδίου όταν γίνεται μέτρηση με το VSM και στο τέλος να τις προσθέτει στο αρχείο που δημιουργείται από το MultiVu ώστε να ο χρήστης να έχει τη δυνατότητα της ανάλυσης όλων των δεδομένων της μέτρησης.

### 5.1. Δημιουργία μέτρησης στο MultiVu

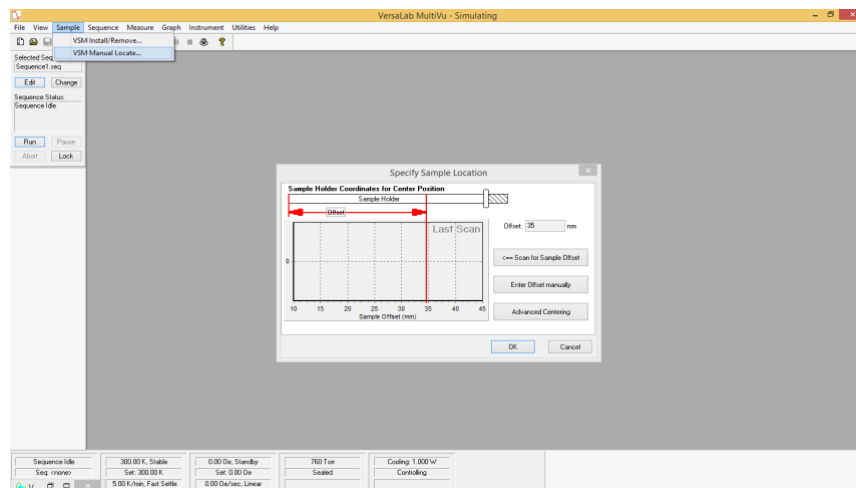
Το λογισμικό MultiVu χρησιμοποιείται όπως έχει ήδη αναφερθεί για την επιλογή της μέτρησης και την προβολή των αποτελεσμάτων. Η επιλογή της μέτρησης μπορεί να γίνει είτε μέσω δημιουργίας ακολουθίας (sequence) είτε μέσω scripting με WinWrap Basic. Το MultiVu προσφέρει την επιλογή της δημιουργίας sequence και μετατροπής σε WinWrap Basic για μεγαλύτερη ευκολία και ευελιξία στο χρήστη. Η τελευταία επιλογή είναι και αυτή που χρησιμοποιήθηκε για τις μετρήσεις στο πλαίσιο αυτής της μελέτης. Τα βήματα για μέτρηση χρησιμοποιώντας το VSM είναι τα εξής :

- 1) Αρχικά πρέπει να ενεργοποιηθεί η επιλογή VSM. Για την ενεργοποίηση πρέπει να επιλεγεί από το menu bar η επιλογή Utilities και στη συνέχεια η Activate Option. Στο αναδυόμενο παράθυρο που θα ανοίξει ο χρήστης πρέπει να επιλέξει αρχικά το VSM και στη συνέχεια το κουμπί Activate.



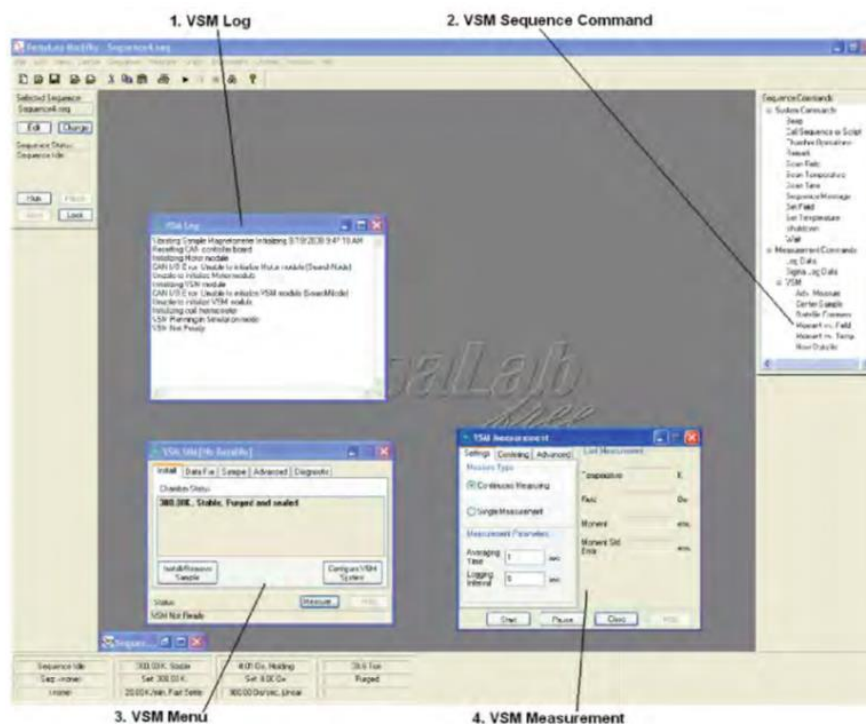
Εικόνα 33 - Επιλογή Activate Option στο MultiVu

- 2) Για να έχουμε μεγάλη ακρίβεια μετρήσεων με το VSM απαιτείται από το σύστημα να ταλαντώσει το δείγμα στο κέντρο του ζεύγους πηνίου και να διατηρηθεί η θέση κεντραρίσματος σταθερή σε απόσταση περίπου 0,1 mm στο κέντρο των πηνίων συλλογής. Έτσι λοιπόν ο χρήστης μέσω του MultiVu πρέπει να κάνει το κεντράρισμα αυτό του δείγματος επιλέγοντας Sample> VSM Manual Locate όπου θα ανοίξει το αντίστοιχο παράθυρο για το κεντράρισμα του δείγματος.



Εικόνα 34 - Η διαδικασία του sample centering στο MultiVu

- 3) Στη συνέχεια δημιουργείται ένα sequence για μέτρηση. Για την δημιουργία νέου sequence ο χρήστης πρέπει να επιλέξει από το menu bar File>New sequence ή να πατήσει το αντίστοιχο εικονίδιο από menu bar. Στη συνέχεια από την πλαϊνή στήλη στα δεξιά του παραθύρου ο χρήστης δημιουργεί το sequence που επιθυμεί. Να σημειωθεί πως για να εμφανιστούν οι επιλογές για μέτρηση με το VSM πρέπει πρώτα να έχει ενεργοποιηθεί η επιλογή του VSM. Επιπλέον αν ο χρήστης επιλέξει να κάνει συνεχόμενες μετρήσεις πρέπει να επιλέξει συχνότητά μετρήσεων τουλάχιστον 0.5s ώστε να προλαβαίνει το Arduino να κάνει μετρήσεις και να στέλνει τις τιμές στη σειριακή θύρα.

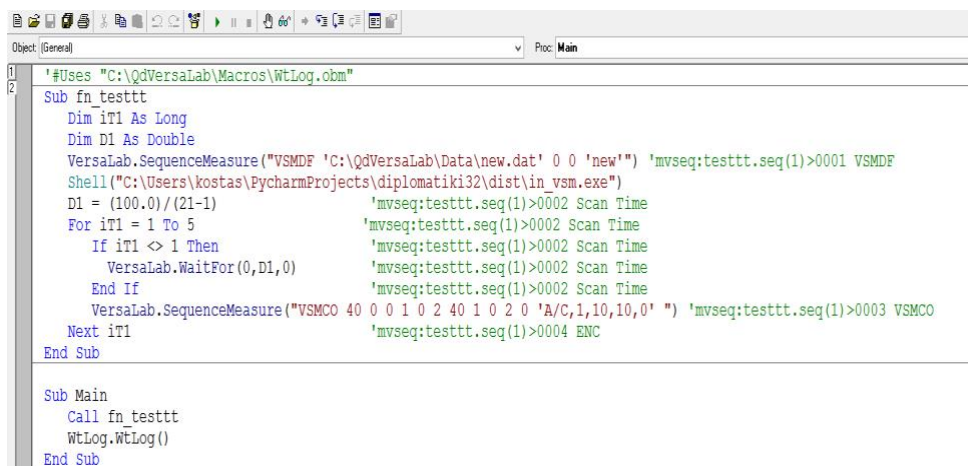


Εικόνα 35 - Εργαλεία διαχείρισης του VSM στο MultiVu

- 4) Μετατροπή της ακολουθίας σε Macro. Αφού ο χρήστης έχει δημιουργήσει το επιθυμητό sequence, για να το μετατρέψει σε Visual Basic script επιλέγει από το menu bar Sequence > Advanced > Compile Macro. Θα εμφανιστεί ένα παράθυρο επεξεργασίας του script.

Για τη σωστή λειτουργία της μετρητικής διάταξης που δημιουργήθηκε ώστε να προστεθεί και το πεδίο του ηλεκτρομαγνητική στα δεδομένα του MultiVu είναι απαραίτητο η προσθήκη ορισμένων εντολών στο script. Πιο συγκεκριμένα προστίθεται η εντολή shell ("path\to\exe") η οποία εκτελεί το πρόγραμμα το όνομα και το path του οποίου πρέπει να δοθεί σαν όρισμα. Η προσθήκη αυτής της εντολής σκοπεύει στην εκκίνηση του προγράμματος in\_vsm.exe. Σε περίπτωση που χρησιμοποιηθεί διαφορετικός μικροελεγκτής από τον προκαθορισμένο ο χρήστης πρέπει να προσθέσει το όνομα ή τη θύρα στην οποία έχει συνδεθεί ο μικροελεγκτής. Δηλαδή μέσα στην εντολή shell προστίθεται και εκτός και από το path και το όνομα ή το COM ή και τα δύο. Για παράδειγμα πρέπει να προστεθεί shell("path/to/in\_vsm.exe Name "). Τέλος προστίθεται η κλήση της συνάρτησης WtLog() που δημιουργήθηκε έτσι ώστε το πρόγραμμα να ενημερώνεται για το πότε τελειώνει μια μέτρηση in\_vsm.exe. Η απλή αυτή συνάρτηση γράφει στο αρχείο log με όνομα eomlog που δημιουργείται στο path την χρονική στιγμή που τελειώνει η μέτρηση το μήνυμα «end of measurement». Για την κλήση της συνάρτησης αυτής πρέπει να προστεθεί στην αρχή του script η

γραμμή «'#Uses "C:\QdVersaLab\Macros\WtLog.obm"» όπου "C:\QdVersaLab\Macros\WtLog.obm" είναι το path του αρχείου WtLog.obm στο οποίο βρίσκεται η συνάρτηση. Πριν το τέλος του script προστίθεται η εντολή WtLog.WtLog() που καλεί την συνάρτηση WtLog(). Παράδειγμα φαίνεται στην εικόνα 36 . Να σημειωθεί πως αν στο script υπάρχει εντολή για δημιουργία νέου αρχείου η εντολή shell() θα πρέπει να προστεθεί μετά την εντολή αυτή. Επειδή το πρόγραμμα in\_vsm ανοίγει το τελευταίο αρχείο data που δημιουργείται από το MultiVu, αν εκκινήσει πριν την δημιουργία του αρχείου της νέας μέτρησης τότε η διαδικασία δεν θα λειτουργήσει σωστά. Ο κώδικας της συνάρτησης WtLog() φαίνεται στην εικόνα 37.

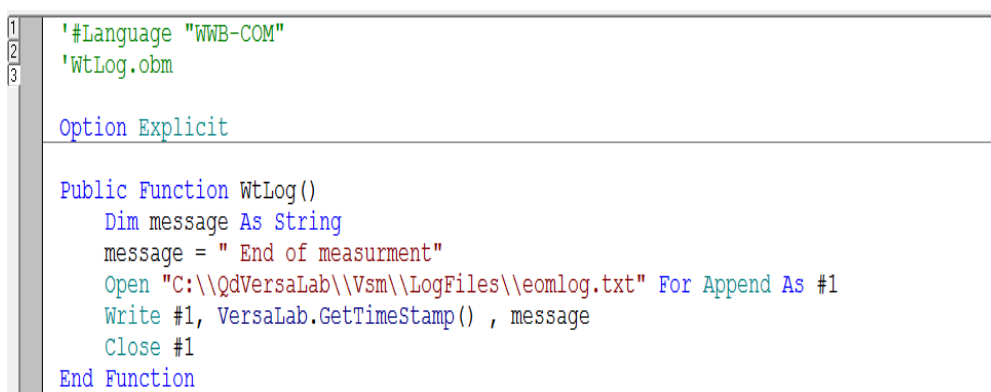


```

1 '#Uses "C:\QdVersaLab\Macros\WtLog.obm"
2
3 Sub fn_testtt
4     Dim iT1 As Long
5     Dim D1 As Double
6     VersaLab.SequenceMeasure("VSMDF 'C:\QdVersaLab\Data\new.dat' 0 0 'new'" 'mvseq:testtt.seq(1)>0001 VSMDF
7     Shell("C:\Users\kostas\PycharmProjects\diomatiki32\dist\in_vsm.exe")
8     D1 = (100.0)/(21-1) 'mvseq:testtt.seq(1)>0002 Scan Time
9     For iT1 = 1 To 5 'mvseq:testtt.seq(1)>0002 Scan Time
10         If iT1 <= 1 Then 'mvseq:testtt.seq(1)>0002 Scan Time
11             VersaLab.WaitFor(0,D1,0) 'mvseq:testtt.seq(1)>0002 Scan Time
12         End If 'mvseq:testtt.seq(1)>0002 Scan Time
13         VersaLab.SequenceMeasure("VSMCO 40 0 0 1 0 2 40 1 0 2 0 'A/C,1,10,10,0' ") 'mvseq:testtt.seq(1)>0003 VSMCO
14     Next iT1 'mvseq:testtt.seq(1)>0004 ENC
15 End Sub
16
17 Sub Main
18     Call fn_testtt
19     WtLog.WtLog()
20 End Sub

```

Εικόνα 36 - Παράδειγμα ενός αρχείου μακροεντολών στο MultiVu



```

1 '#Language "WWB-COM"
2
3 '#WtLog.obm
4
5 Option Explicit
6
7 Public Function WtLog()
8     Dim message As String
9     message = " End of measurment"
10    Open "C:\\QdVersaLab\\Vsm\\LogFiles\\eomlog.txt" For Append As #1
11    Write #1, VersaLab.GetTimeStamp() , message
12    Close #1
13 End Function

```

Εικόνα 37 - Η συνάρτηση WtLog()

## 5.2. Εκτελέσιμο αρχείο in\_vsm.exe

### 5.2.1. Περιγραφή

Το πρόγραμμα in\_vsm αναπτύχθηκε με σκοπό την επικοινωνία με το Arduino και την πραγματοποίηση συγχρονισμένων μετρήσεων μεταξύ VSM και γκαουσόμετρου και την προσθήκη των μετρήσεων πεδίου στο αρχείο που δημιουργείται από το MultiVu για την ανάλυση των δεδομένων. Δημιουργήθηκε με την γλώσσα προγραμματισμού Python και πιο συγκεκριμένα την έκδοση 3.4.3 έτσι ώστε να είναι συμβατό με λειτουργικά Windows XP και μεταγενέστερα. Το πρόγραμμα λειτουργεί ως εξής:

Αρχικά βρίσκει τη θύρα που είναι συνδεδεμένο το Arduino και ανοίγει την σειριακή επικοινωνία. Αν η διαδικασία αποτύχει εμφανίζει σχετικό σφάλμα και κλείνει το πρόγραμμα. Στη συνέχεια βρίσκει και ανοίγει το τελευταίο αρχείο data που δημιουργήθηκε από το MultiVu μέσα στο φάκελο Data(C:\QdVersaLab\Data) η σε κάποιον από τους υποφακέλους του και το ανοίγει. Κάθε φορά που γίνεται μια μέτρηση με VSM στο αρχείο αυτό γράφονται οι τιμές των μετρήσεων που κάνει το VSM. Το πρόγραμμα in\_vsm ελέγχει συνεχώς το αρχείο της μέτρησης έτσι ώστε όταν γίνει καινούρια εγγραφή να την διαβάσει και να δώσει εντολή στο Arduino να κάνει μια μέτρηση. Με τον τρόπο αυτό επιτυγχάνεται και ο συγχρονισμός των μετρήσεων. Το πρόγραμμα αφού διαβάσει την τιμή του πεδίου που στέλνει το Arduino την προσθέτει στην τελευταία εγγραφή που έγινε στο αρχείο και τα γράφει σε ένα προσωρινό αρχείο, το temp.data. Όταν τελειώσει η μέτρηση το πρόγραμμα αντιγράφει τα δεδομένα από το προσωρινό αρχείο στο αρχείο της μέτρησης έτσι ώστε να γίνει ανάλυση της μέτρησης στο MultiVu και το αρχείο temp.data διαγράφεται. Το πρόγραμμα για να καταλάβει πότε τελείωσε η μέτρηση ελέγχει πότε θα γίνει εγγραφή στο αρχείο log eomlog.txt.

Να σημειωθεί ότι το πρόγραμμα κάθε φορά που στέλνει σήμα στο Arduino για να κάνει μέτρηση περιμένει 0.3 δευτερόλεπτα ώστε το Arduino να προλάβει να κάνει τη μέτρηση και να στείλει την τιμή στη σειριακή θύρα.

### 5.2.2. Δημιουργία εκτελέσιμου αρχείου

Το πρόγραμμα MultiVu είναι συμβατό με το λειτουργικό Windows από την έκδοση των Windows 95 μέχρι τα Windows 10. Επιπλέον ο υπολογιστής που χρησιμοποιήθηκε στα πλαίσια της διπλωματικής για τις μετρήσεις έχει εγκατεστημένα Windows XP. Για αυτούς τους λόγους το εκτελέσιμο αρχείο που δημιουργήθηκε είναι συμβατό για εκδόσεις του λειτουργικού συστήματος των Windows XP καθώς και για μεταγενέστερες εκδόσεις λειτουργικών Windows. Για να επιτευχθεί η συμβατότητα αυτή χρησιμοποιήθηκαν οι τελευταίες εκδόσεις της Python και της βιβλιοθήκης PySerial, που δεν περιέχεται στην εγκατάσταση της Python, όπως επίσης και της βιβλιοθήκης Pyinstaller η οποία χρησιμοποιήθηκε



για τη μετατροπή του πηγαίου κώδικα σε εκτελέσιμο αρχείο. Οι εκδόσεις αυτές είναι οι εξής:

- Python 3.4.3 έκδοση 32bit
- PySerial 3.0.1
- Pyinstaller 3.4

Η συγγραφή του κώδικα έγινε στο ολοκληρωμένο περιβάλλον ανάπτυξης Pycharm στην έκδοση Community Edition 2018.3.7 x64. Η εγκατάσταση των βιβλιοθηκών Serial Pyinstaller μέσω του terminal του Pycharm με χρήση των εντολών:

```
Pip install pyserial==3.0.1
```

```
Pip install pyinstaller==3.4
```

Η δημιουργία του εκτελέσιμου αρχείου έγινε επίσης με χρήση terminal του Pycharm με την εντολή:

```
pyinstaller in_vsm.py -F
```

Να σημειωθεί ότι η λειτουργία του προγράμματος δοκιμάστηκε σε λειτουργικά Windows XP, Windows 8.1 64-bit και Windows 10 64-bit.

### 5.3. Κώδικας

Στο σημείο αυτό πραγματοποιείται ανάλυση του κώδικα του προγράμματος in\_vsm. Όπως έχει ήδη αναφερθεί η ανάπτυξη του προγράμματος έγινε με χρήση του ολοκληρωμένου περιβάλλοντος Pycharm.

Αρχικά εισάγονται οι βιβλιοθήκες που χρησιμοποιούνται οι οποίες είναι οι os, time, serial, winsound, ctypes και sys, ενώ στη συνέχεια δημιουργούνται οι συναρτήσεις του προγράμματος.

Η πρώτη συνάρτηση είναι η find\_port(). Κάθε φορά που συνδέεται ένας Arduino με τον υπολογιστή το λειτουργικό σύστημα ορίζει έναν αριθμό εικονικής θύρας COM ο οποίος είναι απαραίτητος για την επικοινωνία με το Arduino. Ο αριθμός αυτός μπορεί να είναι διαφορετικός κάθε φορά που συνδέεται το Arduino. Για τον λόγο αυτό η συνάρτηση find\_port() βρίσκει και επιστρέφει το COM στο οποίο έχει συνδεθεί το Arduino. Η συνάρτηση δέχεται σαν όρισμα το όνομα του μικροεπεξεργαστή το οποίο μπορεί να δώσει σαν όρισμα ο χρήστης σε περίπτωση που χρειαστεί να κάνει χρήση άλλου μικροεπεξεργαστή. Αν ο χρήστης δεν δώσει κάποιο όνομα τότε η συνάρτηση ψάχνει για το Arduino. Σε περίπτωση που δεν βρεθεί η θύρα καλείται η συνάρτηση error\_message.

```

13
14 def find_port(name="Arduino"):
15     """a function that finds and returns
16     in wich serial port Arduino is connected """
17     ports = list(serial.tools.list_ports.comports())
18     for p in ports:
19         if name in p.description:
20             print("Arduino's port:", p[0])
21             return p[0]
22         else:
23             continue
24     error_message("ERROR: can't find Arduino")
25

```

Εικόνα 38 - Η συνάρτηση find\_port

Στη συνάρτηση connect\_with\_arduino(port) αρχικοποιείται και ξεκινάει η σειριακή επικοινωνία με το Arduino. Η συνάρτηση αυτή δέχεται ως όρισμα τον αριθμό COM που είναι συνδεδεμένο το Arduino και δεν επιστρέφει κάποια τιμή. Αν η σύνδεση αποτύχει και εδώ καλείται η συνάρτηση error\_message.

```

27 def connect_with_arduino(port):
28     """this function intialise and stars
29     serial communication with arduino """
30     ser.baudrate = 9600
31     ser.BYTESIZES = 8
32     ser.port = port
33     try:
34         ser.open()
35     except:
36         error_message("ERROR: connection with Arduino failed")
37     else:
38         print("connect with Arduino:", True)
39

```

Εικόνα 39 - Η συνάρτηση connect\_with\_arduino

Η last\_mod\_file() βρίσκει και επιστρέφει το τελευταίο data αρχείο που δημιουργήθηκε η επεξεργάστηκε από το φάκελο και τους υποφακέλους στους οποίους δημιουργεί και αποθηκεύει αρχεία το MultiVu. Σε περίπτωση που δεν βρεθεί κάποιο αρχείο και εδώ καλείται η συνάρτηση error\_message.

```

40
41 def last_mod_file():
42     """this func finds the las modified file in a folder and sub-folders"""
43     path = 'C:\\QdVersaLab\\Data'
44     last_file = max((os.path.join(root, f) for root, _, the_files in os.walk(path)
45                     for f in the_files if f.lower().endswith(".dat")), key=os.path.getctime)
46     if last_file is None:
47         error_message("ERROR: cant find .dat file ")
48     else:
49         print("\nopen file:", last_file)
50     return last_file

```

Εικόνα 40 - Η συνάρτηση last\_mod\_file

Η συνάρτηση log\_file() βρίσκει το αρχείο log eomlog στο οποίο γράφεται από τη συνάρτηση που δημιουργήθηκε για το MultiVu ότι μια μέτρηση έχει τελειώσει. Η

συνάρτηση επιστρέφει το μέγεθος του αρχείου αυτού. Αν το αρχείο δεν υπάρχει τότε η συνάρτηση log\_file το δημιουργεί.

```
53 def log_file():
54     """this func finds the eomlog file which written by multivu when a measurment is over.
55     if log file deleted or doest exit it is created here"""
56     try:
57         log_file_size = os.stat(logfile).st_size
58     except:
59         print("file doesn't exist")
60         log = open(logfile, "w+")
61         log.close()
62         log_file_size = os.stat(logfile).st_size
63     return log_file_size
64
```

Εικόνα 41 - Η συνάρτηση log\_file

Η συνάρτηση get\_field() όταν κληθεί στέλνει σήμα στο Arduino να κάνει μια μέτρηση και διαβάζει την τιμή της μέτρησης και την επιστρέφει. Στη συνάρτηση αυτή έχει οριστεί ένα χρόνος 0.3 δευτερολέπτων όπου το πρόγραμμα μπαίνει σε κατάσταση ύπνου ώστε να προλάβει το Arduino να κάνει τη μέτρηση και να στείλει τη τιμή.

```
66 def get_field():
67     """function that read the magnetic field from arduino"""
68     send = "1"
69     ser.flush()
70     ser.write(send.encode())
71     time.sleep(0.3)
72     x = ser.read(ser.inWaiting())
73     x = x.decode('utf-8')
74     return x
```

Εικόνα 42 - Η συνάρτηση get\_field

Η συνάρτηση write\_to\_tempfile (tempf, l, mf) αντιγράφει κάθε μέτρηση που πραγματοποιείται από το VSM και το Arduino σε ένα προσωρινό αρχείο. Δέχεται τρία ορίσματα και δεν επιστρέφει κάποια τιμή. Τα ορίσματα αυτά είναι το προσωρινό αυτό αρχείο (tempf), η τελευταία γραμμή μετρήσεων (μεταβλητή l) που γράφτηκε από το MultiVu σε ένα data αρχείο και η τιμή του πεδίου που μέτρησε το Arduino (μεταβλητή mf). Η συνάρτηση τοποθετεί στη γραμμή των μετρήσεων του VSM την τιμή του πεδίου και στη συνέχεια αντιγράφει την τιμή αυτή στο προσωρινό αρχείο.

```

77 def write_to_tempfile(tempf, l, mf):
78     """function tha puts the magnetic field in the last line
79     of VSM meassurments and writev the line to a tempfile"""
80     t = l.split(",")
81     t[44] = mf
82     an = ",".join(t)
83     tempf.write(an)

```

Εικόνα 43 - Η συνάρτηση write\_to\_tempfile

Η copy\_data(dat\_file, start) αντιγράφει τα δεδομένα των μετρήσεων από το προσωρινό αρχείο στο αρχικό αρχείο που δημιουργήθηκε από το MultiVu αφού τελειώσουν όλες οι μετρήσεις. Δέχεται ως όρισμα το αρχείο data και το σημείο του αρχείο στο οποίο θα γίνει η αντιγραφή.

```

86 def copy_data(dat_file, start):
87     """function tha copies all meassurments from temp file to the data section of data file"""
88     with open(dat_file, "r+") as file, open(tempfile, "r") as temp:
89         file.seek(start)
90         while True:
91             line = temp.readline()
92             if not line:
93                 break
94             file.write(line)

```

Εικόνα 44 - Η συνάρτηση copy\_data

Η συνάρτηση meassurment() είναι αυτή που συγχρονίζει τη διαδικασία. Η συνάρτηση αυτή δεν δέχεται κάποιο όρισμα ούτε επιστρέφει κάποια τιμή. Αρχικά καλεί την last\_mod\_file() ώστε να βρεθεί το αρχείο στο οποίο θα γράψει τις τιμές των μετρήσεων το MultiVu. Στη συνέχεια αφού βρει το σημείο του αρχείου όπου θα ξεκινήσουν να γίνονται οι εγγραφές των μετρήσεων ξεκινάει ένα βρόχος. Στον βρόχο ελέγχεται συνέχεια το data αρχείο για νέες εγγραφές. Κάθε φορά που γίνεται μια εγγραφή από το MultiVu η συνάρτηση την διαβάζει και καλεί τις συναρτήσεις get\_field() και write\_to\_tempfile() ώστε να πάρει και την τιμή του πεδίου και να γράψει όλες τις τιμές σε ένα προσωρινό αρχείο. Με τον τρόπο αυτό επιτυγχάνεται ο συγχρονισμός των μετρήσεων μεταξύ VSM και Arduino. Για να βγει το πρόγραμμα από το βρόχο χρησιμοποιείται το log αρχείο eomlog. Πριν τον βρόχο καλείται η συνάρτηση log\_file() για να διαβαστεί το μέγεθος του αρχείου. Μέσα στο βρόχο ελέγχεται συνέχεια πότε αλλάζει το μέγεθος του log δηλαδή πότε γράφτηκε από MultiVu ότι η μέτρηση τέλειωσε. Τότε τελειώνει ο βρόχος και καλείται η copy\_data().

```

97 def measurement():
98     """function that synchronizes VSM with Arduino measurements
99     by polling to data file written by VSM when it takes a measure"""
100     data_file = last_mod_file()
101     log_size = log_file()
102     with open(data_file, "r") as f, open(tempfile, "w") as new:
103         f.seek(0, 2)
104         size = f.tell()
105         offset = f.tell()
106         print("\n", "-" * 20, "start", "-" * 20)
107         while True:
108             f.seek(0, 2)
109             if size != f.tell():
110                 field = get_field()
111                 f.seek(size)
112                 line = f.readline()
113                 write_to_tempfile(new, line, field) #####
114                 f.seek(0, 2)
115                 size = f.tell()
116             elif log_size < os.stat(logfile).st_size:
117                 print("end of measurements\n")
118                 print("copy data to file")
119                 break
120             else:
121                 time.sleep(0.001)
122                 continue
123     copy_data(data_file, offset)
124     os.remove(tempfile)

```

Εικόνα 45 - Η συνάρτηση measurement

Η συνάρτηση error\_message (message) καλείται όταν συμβεί κάποιο σφάλμα και εμφανίζει αναδυόμενο παράθυρο σφάλματος όπου εμφανίζει το μήνυμα. Επίσης αναπαράγεται και ήχος των Windows. Η συνάρτηση δέχεται ως όρισμα το μήνυμα του σφάλματος.

```

142 def error_message(message):
143     """function th prints error masseges """
144     winsound.PlaySound('SystemExit', winsound.SND_ALIAS)
145     print("#" * 15, message, "#" * 15)
146     ctypes.windll.user32.MessageBoxW(0, message, u"Error in vsm.exe", 0)
147     exit(1)

```

Εικόνα 46 - Η συνάρτηση error\_message

Η συνάρτηση find\_arduino(arg) επιλεγεί και στέλνει το κατάλληλο όρισμα στην συνάρτηση find\_port() για να βρεθεί το COM του μικροεπεξεργαστή το οποίο και επιστρέφει. Η συνάρτηση διαβάζει τα ορίσματα κατά την εκκίνηση του προγράμματος. Έτσι δίνει τη δυνατότητα στο χρήστη να εισάγει το όνομα ή ακόμα και απευθείας το COM του συνδεδεμένου μικροεπεξεργαστή που χρησιμοποιεί αν δεν είναι ο προκαθορισμένος Arduino. Με αυτόν τρόπο μπορεί να χρησιμοποιηθεί άλλος μικροεπεξεργαστής χωρίς να χρειαστεί ο χρήστης να αλλάξει τον κώδικα και να δημιουργήσει και πάλι το εκτελέσιμο αρχείο.

```

127 def find_arduino(arg):
128     if len(arg) > 1:
129         flag = True
130         for aa in arg:
131             if "COM" in aa:
132                 return aa
133         if flag:
134             name = " ".join(arg[1:])
135             arduinos_port = find_port(name)
136             return arduinos_port
137     else:
138         arduinos_port = find_port()
139         return arduinos_port

```

Εικόνα 47 - Η συνάρτηση find\_arduino

Και τέλος η main(arguments) καλεί τις συναρτήσεις find\_arduino , connect\_with\_arduino() και meassurment() και αφού ολοκληρωθούν οι μετρήσεις κλείνει την σειριακή επικοινωνία.

```

149
150 def main(arguments):
151     arduinos_port = find_arduino(arguments)
152     connect_with_arduino(arduinoo(arduinoo_port))
153     meassurment()
154     ser.close()
155     exit(0)

```

Εικόνα 48 - Η συνάρτηση main

## 5.4. Μέτρηση

Για τον έλεγχο της ορθής λειτουργίας της μετρητικής διάταξης και του προγράμματος που αναπτύχθηκε πραγματοποιήθηκαν 2 μετρήσεις. Τα δείγματα τα οποία χρησιμοποιήθηκαν για αυτές είναι ένα δείγμα οξειδίου του σιδήρου (ferric oxide  $\text{Fe}_3\text{O}_4$ ) και ένα εξαφερίτη του βαρίου (barium hexaferrite  $\text{BaFe}_{12}\text{O}_{19}$ ). Για τη μέτρηση ακολουθήθηκαν τα βήματα που περιγράφηκαν σε προηγούμενες ενότητες.

Το sequence που επιλέχθηκε φαίνεται στην εικόνα 49. Σύμφωνα με αυτό το VSM πραγματοποιεί συνεχείς μετρήσεις κάθε 0.5 δευτερόλεπτα για διάρκεια 100 δευτερολέπτων.

```

Selected Line: 3
VSM Measure for 0.5 sec at 2 mm 40 Hz every 0 sec AutoCenter 10 10 0
Wait For Delay 100 secs (1.7 mins). No Action
End Sequence

```

Εικόνα 49 - Το sequence των μετρήσεων

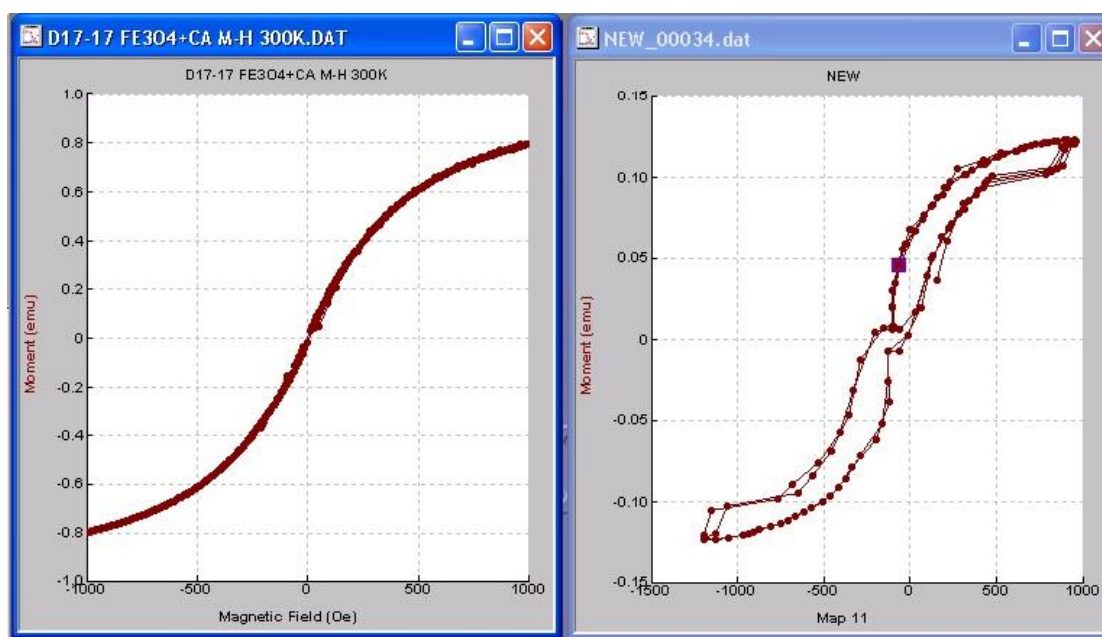
Στη συνέχεια πατήθηκε η επιλογή compile macro προκειμένου να δημιουργηθεί το αρχείο bas και προστέθηκαν οι κατάλληλες εντολές.

```
1 '#Uses "C:\QdVersaLab\Macros\WtLog.obm"
2 Sub fn_Sequence1
VersaLab.SequenceMeasure("VSMDF 'C:\QdVersaLab\Data\new.dat' 0 0 'new'" 'mvseq:testtt.seq(1)>0001 VSMDF
Shell("C:\Documents and settings\VL\Desktop\in_vsm.exe")
VersaLab.SequenceMeasure("VSMCO 20 0 0 0 0 2 40 0.5 0 2 0 'A/C,1,10,10,0' ") 'mvseq:Sequence1.seq(1)>0001 VSMCO
VersaLab.WaitFor(0+2*0+4*0+8*0,100,0) 'mvseq:Sequence1.seq(1)>0002 Wait For 4t
End Sub

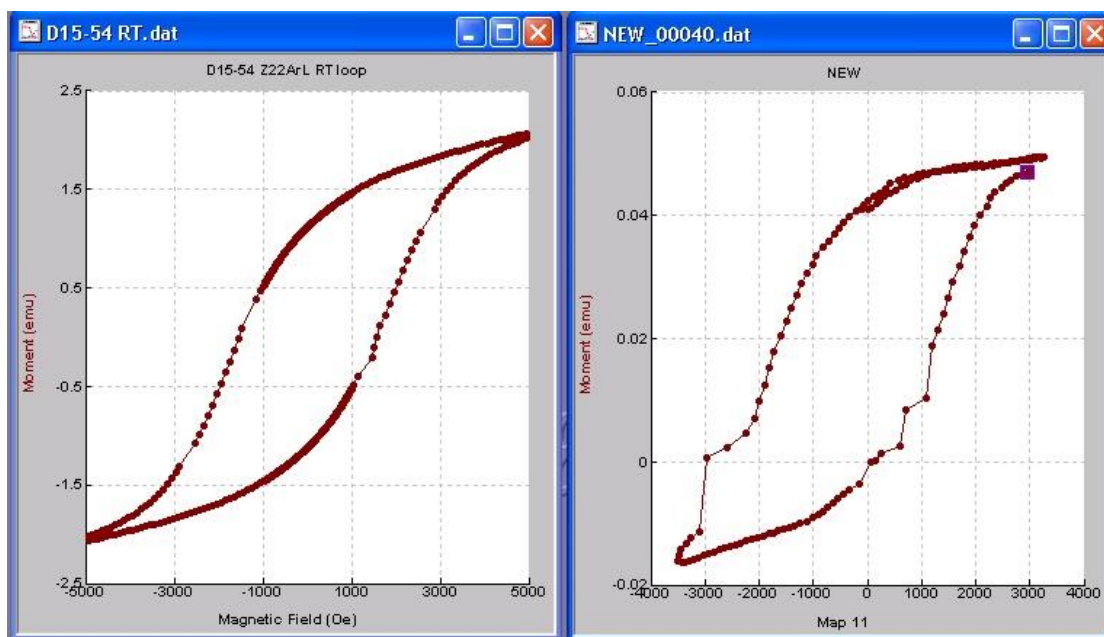
Sub Main
Call fn_Sequence1
WtLog.WtLog()
End Sub
```

Εικόνα 50 - Το αρχείο μακροεντολών των μετρήσεων

Τα αποτελέσματα των μετρήσεων παρουσιάζονται στις εικόνες 51 και 52.



Εικόνα 51 – Μετρήσεις σε δείγμα οξειδίου του σιδήρου



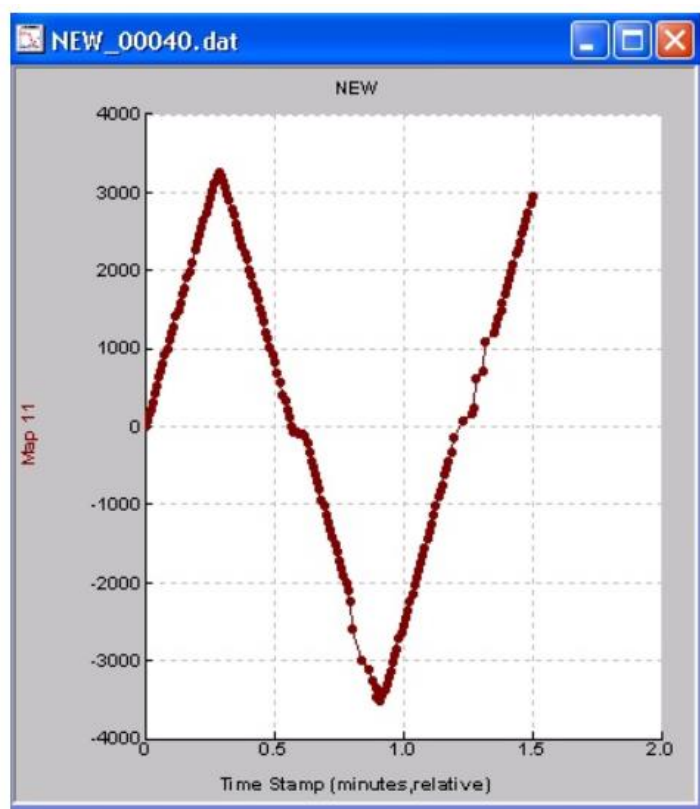
Εικόνα 52 - Μετρήσεις σε δείγμα εξαφερίτη βαρίου

Στην εικόνα 51 φαίνεται το δείγμα του οξειδίου του σιδήρου. Αριστερά είναι μία μέτρηση που είχε πραγματοποιηθεί στο σύστημα Versalab και δεξιά φαίνεται η μέτρηση που διεξήχθη στα πλαίσια της διπλωματικής εργασίας. Το σύστημα φαίνεται να λειτουργεί καθώς προκύπτει ο αναμενόμενος βρόχος. Φυσικά χρήζει κάποιος βελτίωσης διότι σε μερικά σημεία ο βρόχος φαίνεται να αποκλίνει. Εδώ αξίζει να σημειωθεί πως οι τιμές του πεδίου για τη δεξιά εικόνα δεν έχουν υποστεί επεξεργασία και είναι ουσιαστικά η τάση που λήφθηκε από το όργανο σε mV. Η διαφορά με την εικόνα στα αριστερά στις τιμές της μαγνητικής ροπής έγκειται στο γεγονός ότι οι τιμές πεδίου που απαιτούνταν ήταν οι διπλάσιες.

Στην εικόνα 52 φαίνεται το δείγμα του εξαφερίτη του βαρίου. Εδώ διακρίνεται η ικανότητα του υλικού να διατηρεί τη μαγνήτισή του σε συνθήκες μηδενικού πεδίου σε αντίθεση με το παραπάνω δείγμα. Και εδώ υπάρχουν ορισμένα σημεία στα οποία ο βρόχος αποκλίνει.

Τέλος στην εικόνα 53 φαίνεται η απόκριση του πεδίου στην πάροδο του χρόνου. Η συγκεκριμένη εικόνα προέκυψε από τη μέτρηση με το δείγμα του εξαφερίτη του βαρίου.





Εικόνα 53 - Η απόκριση του πεδίου στο χρόνο στη μέτρηση του δείγματος εξαφερίτη βαρίου

## 6. Συμπεράσματα – Επεκτάσεις

Σε αυτό το σημείο είναι σκόπιμο να παρουσιαστούν τα συμπεράσματα τα οποία προέκυψαν από την παρούσα μελέτη καθώς και να αναφερθούν κάποιες μελλοντικές επεκτάσεις.

### 6.1. Σύνοψη – Συμπεράσματα

Σκοπός αυτής της διπλωματικής ήταν η υλοποίηση μιας διάταξης μετρήσεων προκειμένου να καταστεί δυνατή η μέτρηση του ηλεκτρομαγνητικού πεδίου το οποίο παράγεται από τον ηλεκτρομαγνήτη του υστερησιογράφου MH/40-20 της Walker Magnemetrics. Με αυτόν τον τρόπο και σε συνδυασμό με τη χρήση του VSM της Quantum Design για τη μέτρηση της μαγνητικής ροπής ενός δείγματος ήταν δυνατή η παραγωγή βρόχου υστέρησης, ο οποίος διαδραματίζει σημαντικό ρόλο στον καθορισμό των ιδιοτήτων διαφόρων μαγνητικών υλικών.

Έναυσμα για τη συγκεκριμένη μελέτη αποτέλεσε το γεγονός ότι τον προηγούμενο καιρό στο εργαστήριο Ηλεκτροτεχνικών Υλικών του τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών πραγματοποιούνταν μετρήσεις με τη χρήση του VSM και του συστήματος PPMS. Με αυτόν τον τρόπο όμως δημιουργούνται κάποια προβλήματα τα οποία δυσχεραίνουν τη διεξαγωγή μετρήσεων. Ένα από αυτά είναι το γεγονός ότι προκειμένου να μπορέσει γίνει κάποια μέτρηση στο σύστημα PPMS Versalab πρέπει ο υπεραγώγιμος ηλεκτρομαγνήτης που περιέχει να ψυχθεί στους 4 βαθμούς Kelvin και συνεπώς είναι απαραίτητη η παρέλευση μεγάλου χρονικού διαστήματος. Ένα άλλο μειονέκτημα είναι οι αυξημένες ανάγκες συντήρησης και ψυκτικού υγρού που μεγαλώνουν το κόστος λειτουργίας.

Η υλοποίηση της διάταξης έγινε με τη χρήση ενός μετρητή πυκνότητας μαγνητικής ροής, του FW Bell 5180 Gauss/Tesla Meter, όπου από την αναλογική έξοδό του λαμβάνεται μία εξωτερική τάση και ενός Arduino με το οποίο πραγματοποιήθηκε η ανάγνωση αυτής της τάσης. Επίσης συντάχθηκε κώδικας προκειμένου να μεταφέρει την τάση που λάμβανε ο Arduino στο αρχείο που εξήγαγε η εφαρμογή MultiVu κατά τη διάρκεια των μετρήσεων.

Για τη δοκιμή του συστήματος μετρήσεων που σχεδιάστηκε εκτελέστηκαν στο εργαστήριο διάφορες μετρήσεις και συγκρίθηκαν με αντίστοιχες που έγιναν στην ολοκληρωμένη πλατφόρμα PPMS. Τα αποτελέσματα που εξήχθησαν χαρακτηρίστηκαν από τον επιβλέποντα καθηγητή ικανοποιητικά, τα οποία με λίγη διόρθωση θα μπορούσαν να προσφέρουν ακόμα μεγαλύτερη αξιοπιστία. Το γεγονός ότι ο συγκεκριμένος ηλεκτρομαγνήτης χρησιμοποιεί το κοινό νερό ως ψυκτικό μέσο και το ότι απαιτείται ελάχιστος χρόνος για τη διεξαγωγή μετρήσεων αποτελούν τα πλεονεκτήματα της διάταξης του VSM με τον ηλεκτρομαγνήτη, ενώ κάποια μειονεκτήματα είναι ότι ο ηλεκτρομαγνήτης μπορεί να παράξει μικρότερες τιμές πεδίου σε σχέση με αυτόν του Versalab, καθώς και ότι οι

μετρήσεις δύνανται να διεξαχθούν μόνο σε συνθήκες περιβάλλοντος σε αντίθεση με το Versalab που μπορεί να λειτουργήσει σε ένα εύρος θερμοκρασίας από 50 έως 400 K.

## 6.2. Μελλοντικές επεκτάσεις

Με την ολοκλήρωση της παρούσας εργασίας προέκυψαν διάφορες επιλογές που μπορούν να χρησιμοποιηθούν ως μελλοντικές επεκτάσεις.

Μία από αυτές είναι η δυνατότητα ενοποίησης της δουλειάς που είχε γίνει σε προηγούμενη μελέτη με στόχο της χρήσης ενός Arduino για τον έλεγχο ενός κινητήρα με σκοπό την περιστροφή ενός ποτενσιόμετρου για τον έλεγχο του ρεύματος που φτάνει στον ηλεκτρομαγνήτη. Αυτή η χρήση του Arduino επιτρέπει τη μεταφορά του ενός συστήματος στο άλλο και τον έλεγχο και των δύο από τον κεντρικό υπολογιστή που διεξάγονται οι μετρήσεις. Μία άλλη πιθανή υλοποίηση που θα αυτοματοποιήσει ακόμα περισσότερο τη διαδικασία των μετρήσεων είναι η δημιουργία μιας μεθόδου στη γλώσσα της WinWrap Basic που χρησιμοποιεί το MultiVu η οποία θα λαμβάνει μία τιμή πεδίου από το χρήστη και θα ελέγχει το ποτενσιόμετρο προκειμένου να επιτευχθεί η επιθυμητή τιμή. Μία ακόμα επέκταση είναι η χρήση της δεύτερης υλοποίησης που έγινε στη συγκεκριμένη μελέτη, όπου όπως περιγράφηκε και στο κεφάλαιο 4.3 είναι δυνατή η σειριακή επικοινωνία με το όργανο μέτρησης πεδίου. Σε πιθανή μελλοντική αναβάθμιση του υπολογιστή με τον οποίο πραγματοποιούνται οι μετρήσεις μπορεί πολύ εύκολα να εξαλειφθεί το Arduino από τη διάταξη και οι τιμές του πεδίου να λαμβάνονται απευθείας από το γκαουσόμετρο είτε με τη χρήση του κώδικα της C++ που συντάχθηκε είτε με άλλον παρεμφερή τρόπο. Τέλος είναι επιθυμητή η προμήθεια και χρήση ψυκτικού μέσου διαφορετικού του νερού προκειμένου να επιτευχθεί μεγαλύτερο εύρος θερμοκρασίας.

## Αναφορές

- [1] A. Lopes Ribeiro, "Vibrating sample magnetometer for large permanent magnets," AFRICON 2007, Windhoek, 2007, [Ηλεκτρονικό]  
Available:  
<https://ieeexplore.ieee.org/document/4401540>
- [2] PPMS Vibrating Sample Magnetometer(VSM) Option User's Manual, [Ηλεκτρονικό]  
Available:  
[https://web.njit.edu/~tyson/PPMS\\_Documents/PPMS\\_Manual/1096-100%20Rev.%20A3%20VSM.pdf](https://web.njit.edu/~tyson/PPMS_Documents/PPMS_Manual/1096-100%20Rev.%20A3%20VSM.pdf)
- [3] PPMS MultiVu Application User's Manual [Ηλεκτρονικό]  
Available:  
[https://www.mrl.ucsb.edu/sites/default/files/mrl\\_docs/instruments/multiPPMS.pdf](https://www.mrl.ucsb.edu/sites/default/files/mrl_docs/instruments/multiPPMS.pdf)
- [4] WinWrap® Basic, [Ηλεκτρονικό]  
Available:  
<http://www.winwrap.com/web2/home/>
- [5] WALKER/MAGNOMETRICS, INSTRUCTION MANUAL MH-40/20 HYSTERESISGRAPH, Worcester, Mass.: Walker Scientific Inc.
- [6] Συντάκας Σπυρίδων, Μελέτη και υλοποίηση μηχαντρονικής διάταξης για έλεγχο πεδίου ηλεκτρομαγνήτη  
Διπλωματική εργασία, Τμήμα Ηλεκτρολόγων Μηχ. Και Μηχανικών Η/Υ, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, 2017
- [7] W. D. Callister, ΕΠΙΣΤΗΜΗ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑ ΤΩΝ ΥΛΙΚΩΝ, 5<sup>η</sup> Έκδοση, ΕΚΔΟΣΕΙΣ ΤΖΙΟΛΑ, 2008
- [8] Θ. Δ. Τσιμπούκης, ΗΛΕΚΤΡΟΜΑΓΝΗΤΙΚΟ ΠΕΔΙΟ Βασική θεωρία και εφαρμογές ΤΟΜΟΣ 1, Έκδοση 2011, ΠΑΝΕΠΙΣΤΗΜΙΑΚΕΣ ΕΚΔΟΣΕΙΣ ΚΡΗΤΗ.
- [9] Wikipedia, Signal-to-noise ratio [Ηλεκτρονικό]  
Available:  
[https://en.wikipedia.org/wiki/Signal-to-noise\\_ratio](https://en.wikipedia.org/wiki/Signal-to-noise_ratio)
- [10] Wikipedia, Analog-to-digital converter [Ηλεκτρονικό]  
Available:  
[https://en.wikipedia.org/wiki/Analog-to-digital\\_converter#Direct-conversion](https://en.wikipedia.org/wiki/Analog-to-digital_converter#Direct-conversion)
- [11] Β. Πετρίδη, ΜΕΤΡΗΣΕΙΣ, ΗΛΕΚΤΡΙΚΕΣ ΜΕΤΡΗΣΕΙΣ και ΣΥΣΤΗΜΑΤΑ ΜΕΤΡΗΣΕΩΝ, έκδοση 2013

- [12] Arduino, [Ηλεκτρονικό]  
Available:  
<https://www.arduino.cc>
- [13] ATmega328P DATASHEET [Ηλεκτρονικό]  
Available:  
[https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P\\_Datasheet.pdf](https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf)
- [14] AVR120: Characterization and Calibration of the ADC on an AVR, [Ηλεκτρονικό]  
Available:  
[http://ww1.microchip.com/downloads/en/appnotes/atmel-2559-characterization-and-calibration-of-the-adc-on-an-avr\\_applicationnote\\_avr120.pdf](http://ww1.microchip.com/downloads/en/appnotes/atmel-2559-characterization-and-calibration-of-the-adc-on-an-avr_applicationnote_avr120.pdf)
- [15] 5100 Series Gaussmeters User Manual, [Ηλεκτρονικό]  
Available:  
<https://fwbell.com/wp-content/uploads/2017/12/UN1260.pdf>
- [16] AVR121: Enhancing ADC resolution by oversampling [Ηλεκτρονικό]  
Available:  
<http://ww1.microchip.com/downloads/en/appnotes/doc8003.pdf>
- [17] Μ. Η. HAYES, ΨΗΦΙΑΚΗ ΕΠΕΞΕΡΓΑΣΙΑ ΣΗΜΑΤΟΣ, ΕΚΔΟΣΕΙΣ ΤΖΙΟΛΑ
- [18] ΙΝΣΤΙΤΟΥΤΟ ΝΑΝΟΕΠΙΣΤΗΜΗΣ ΚΑΙ ΝΑΝΟΤΕΧΝΟΛΟΓΙΑΣ ΕΚΕΦΕ  
«Δημόκριτος», Μαγνητικές μετρήσεις υλικών [Ηλεκτρονικό]  
Available:  
[http://www.physics.ntua.gr/ergasthria/askhseis\\_ergasthron/magnhtikes\\_metrhseis.pdf](http://www.physics.ntua.gr/ergasthria/askhseis_ergasthron/magnhtikes_metrhseis.pdf)
- [19] Wikipedia, Arduino Uno [Ηλεκτρονικό]  
Available:  
[https://en.wikipedia.org/wiki/Arduino\\_Uno](https://en.wikipedia.org/wiki/Arduino_Uno)
- [20] A. Nayyar and V. Puri, "A review of Arduino board's, Lilypad's & Arduino shields," 2016 3rd International Conference on Computing for Sustainable Global Development ,[Ηλεκτρονικό]  
Available:  
<https://ieeexplore.ieee.org/document/7724514>
- [21] Βουλγαρόγλου Μιχάλης, Προσαρμογή συστήματος μαγνητικών μετρήσεων σε ηλεκτρομαγνήτη  
Διπλωματική εργασία, Τμήμα Ηλεκτρολόγων Μηχ. Και Μηχανικών Η/Υ, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης , 2018

## Ευρετήριο εικόνων

|  |    |
|--|----|
| Εικόνα 1 - Αρχή λειτουργίας VSM.....   | 14 |
| Εικόνα 2 - Αποτελούμενα μέρη ενός VSM.....   | 15 |
| Εικόνα 3 - Γραμμικός κινητήρας VSM.....  | 15 |
| Εικόνα 4 - Το λογισμικό MultiVu .....  | 16 |
| Εικόνα 5 - Το περιβάλλον σύνταξης μακροεντολών του MultiVu.....  | 17 |
| Εικόνα 6 - Η διάταξη του ηλεκτρομαγνήτη με το VSM .....  | 18 |
| Εικόνα 7 - Μαγνητική Ροπή .....  | 19 |
| Εικόνα 8 - Καμπύλη μαγνήτισης.....   | 22 |
| Εικόνα 9 - Βρόχος υστέρησης .....  | 23 |
| Εικόνα 10 - Δειγματοληψία σήματος .....  | 24 |
| Εικόνα 11 - Κβαντισμός σήματος.....  | 24 |
| Εικόνα 12 - Μετατροπέας A/D διαδοχικών προσεγγίσεων .....  | 26 |
| Εικόνα 13 - Arduino uno .....  | 28 |
| Εικόνα 14 - Block διάγραμμα.....   | 29 |
| Εικόνα 15 - Α) Λειτουργία απλής μετατροπής Β) Λειτουργία διαφορικής μετατροπής<br>.....                  | 31 |
| Εικόνα 16 - Κύκλωμα ADC του AtMega328p.....  | 32 |
| Εικόνα 17 - Κύκλωμα prescale .....   | 33 |
| Εικόνα 18 - Παράδειγμα Α) Θετικού Σφάλματος Αντιστάθμισης, Β) Αρνητικού<br>Σφάλματος Αντιστάθμισης ..... | 34 |
| Εικόνα 19 - Α) Θετικό και Β) Αρνητικό Σφάλμα Αντιστάθμισης στη Λειτουργία Απλής<br>Μετατροπής.....       | 34 |
| Εικόνα 20 - Παράδειγμα Α) Θετικού και Β) Αρνητικού Σφάλματος Κέρδους .....                               | 35 |
| Εικόνα 21 - Παραδείγματα Α) Διαφορικής και Β) Ολοκληρωτικής Μη-Γραμμικότητας<br>.....                    | 36 |
| Εικόνα 22 - Αισθητήρας Hall.....   | 36 |
| Εικόνα 23 - Hall probes .....  | 37 |
| Εικόνα 24 - FW Bell 5180 .....   | 38 |
| Εικόνα 25 - Κομμάτι κώδικα C++ (α).....  | 39 |
| Εικόνα 26 - Κομμάτι κώδικα C++ (β).....  | 40 |
| Εικόνα 27 - Κομμάτι κώδικα C++ (γ) .....   | 41 |
| Εικόνα 28 - Κύκλωμα διαιρέτη τάσης.....  | 42 |
| Εικόνα 29 - Η διαδικασία του Decimation.....   | 45 |
| Εικόνα 30 - Οι αναλύσεις του Adc με τη μέθοδο του Oversampling .....                                     | 47 |
| Εικόνα 31 - Η συνάρτηση void setup() του Arduino .....   | 49 |
| Εικόνα 32 - Η συνάρτηση void loop() του Arduino.....   | 49 |
| Εικόνα 33 - Επιλογή Activate Option στο MultiVu .....  | 51 |
| Εικόνα 34 - Η διαδικασία του sample centering στο MultiVu.....   | 52 |
| Εικόνα 35 - Εργαλεία διαχείρισης του VSM στο MultiVu.....  | 53 |
| Εικόνα 36 - Παράδειγμα ενός αρχείου μακροεντολών στο MultiVu .....                                       | 54 |

|   |    |
|---|----|
| Εικόνα 37 - Η συνάρτηση WtLog() .....   | 54 |
| Εικόνα 38 - Η συνάρτηση find_port.....  | 57 |
| Εικόνα 39 - Η συνάρτηση connect_with_arduino .....  | 57 |
| Εικόνα 40 - Η συνάρτηση last_mod_file.....  | 57 |
| Εικόνα 41 - Η συνάρτηση log_file.....   | 58 |
| Εικόνα 42 - Η συνάρτηση get_field .....   | 58 |
| Εικόνα 43 - Η συνάρτηση write_to_tempfile .....   | 59 |
| Εικόνα 44 - Η συνάρτηση copy_data .....   | 59 |
| Εικόνα 45 - Η συνάρτηση measurment .....  | 60 |
| Εικόνα 46 - Η συνάρτηση error_message .....   | 60 |
| Εικόνα 47 - Η συνάρτηση find_arduino .....  | 61 |
| Εικόνα 48 - Η συνάρτηση main.....   | 61 |
| Εικόνα 49 - Το sequence των μετρήσεων .....   | 61 |
| Εικόνα 50 - Το αρχείο μακροεντολών των μετρήσεων .....  | 62 |
| Εικόνα 51 – Μετρήσεις σε δείγμα οξειδίου του σιδήρου .....                                      | 62 |
| Εικόνα 52 - Μετρήσεις σε δείγμα εξαφερίτη βαρίου .....  | 63 |
| Εικόνα 53 - Η απόκριση του πεδίου στο χρόνο στη μέτρηση του δείγματος<br>εξαφερίτη βαρίου ..... | 64 |

## Παράρτημα

Κώδικας C++ για την επικοινωνία με το 5180:

```
#include <windows.h>
#include <stdio.h>
#include <iostream>
#include <fstream>
#include <ctype.h>
#include <algorithm>
using namespace std;

typedef int (__stdcall *f_func1)();
typedef int (__stdcall *func_ptr)(unsigned int,char*,char*,int);
typedef void(__stdcall *func_close)(unsigned int);
unsigned int id;
HINSTANCE hGetProcIDDLL = LoadLibrary("F:\\\\Program Files (x86)\\\\FW
Bell\\\\PC5180\\\\usb5100.dll");
f_func1 open = (f_func1)GetProcAddress(hGetProcIDDLL, "openUSB5100");
func_ptr data = (func_ptr)GetProcAddress(hGetProcIDDLL,"scpiCommand");
func_close close =
(func_close)GetProcAddress(hGetProcIDDLL,"closeUSB5100");

void openusb(){
    id = open();
    if (!id) {
        cout << "could not locate the function" << endl;
    }
}

void closeusb(){
    close(id);
}

char* readdata(){
    char* ac;
    data(id,(char*)" :MEASURE:FLUX?",ac,80);
    return ac;
}

string returnNumber(char* a){
    string b(a);

    int counta=0;
    int countb=0;
    b.erase(remove(b.begin(),b.end(),'G'),b.end());
    b.erase(remove(b.begin(),b.end(),' '),b.end());
    return b;
}

string lastf()
{
    WIN32_FIND_DATAW ffd;
    wchar_t const* directory = L"C:\\\\QdVersaLab\\\\Data\\";
    wchar_t currentFile[MAX_PATH], lastModifiedFilename[MAX_PATH];
    FILETIME currentModifiedTime, lastModified;
    HANDLE hFile;
```



```

    bool first_file = true;

    HANDLE hFind = FindFirstFileW( L"C:\\QdVersaLab\\Data\\*", &ffd
);

    if ( INVALID_HANDLE_VALUE == hFind )
    {
        cout<<"cant find path";
    }

    do
    {
        if ( !( ffd.dwFileAttributes & FILE_ATTRIBUTE_DIRECTORY ) )
        {
            wcsncpy( currentFile, directory );
            wscat( currentFile, ffd.cFileName );
            // open file to read it's last modified time
            hFile = CreateFileW( currentFile, GENERIC_READ,
FILE_SHARE_READ, NULL,
            OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, NULL );
            if( INVALID_HANDLE_VALUE != hFile )
            {
                // get it's last write time
                if( GetFileTime( hFile, NULL, NULL,
&currentModifiedTime ) != 0 )
                {
                    if( first_file )
                    {
                        lastModified = currentModifiedTime;
                        wcsncpy( lastModifiedFilename, ffd.cFileName
);

                        first_file = false;
                    }
                    else
                    {
                        // First file time is earlier than second
file time.

                        if( CompareFileTime( &lastModified,
&currentModifiedTime ) == -1 )
                        {
                            lastModified = currentModifiedTime;
                            wcsncpy( lastModifiedFilename,
ffd.cFileName );
                        }
                    }
                }
                CloseHandle( hFile );
            }
        }
    }
    while ( FindNextFileW( hFind, &ffd ) != 0 );

    FindClose( hFind );
    wstring ws(lastModifiedFilename);
    wstring s(directory);
    // your new String
    string str1(s.begin(), s.end());
    string str2(ws.begin(), ws.end());
    string filename=str1+str2;
    return filename;
}

```

```

void copydata(const char* dest ,const char* source){
    std::ifstream  src(source );
    std::ofstream  dst(dest);
    dst << src.rdbuf();
    dst.close();
    src.close();
}
int main(){

    //  openusb();
    char *aaa;
    string bbb;
    streampos size,begin,end,logsize,prev,endlog;
    string str= lastf();

    //string line;
    const char * c = str.c_str();
    const char* tempfile="C:\\Documents and
Settings\\Administrator\\My Documents\\temp.dat";
    copydata(tempfile,c);
    ifstream file(c);
    ifstream log("C:\\Documents and Settings\\Administrator\\My
Documents\\eomlog.txt");
    ofstream temp(tempfile,ios::app);
    file.seekg( 0, ios::end );
    size=file.tellg();
    prev=size;
    log.seekg( 0, ios::end );
    logsize=log.tellg();
    begin=logsize;
    file.close();
    log.close();
    int pos;
    while(TRUE){
        Sleep(1);
        fstream file(c);
        file.seekg( 0, ios::end );
        end=file.tellg();
        ifstream log("C:\\Documents and Settings\\Administrator\\My
Documents\\eomlog.txt");
        log.seekg( 0, ios::end );
        endlog=log.tellg();
        if (prev<end){
            string line;
            aaa=readdata();
            bbb=returnNumber(aaa);
            file.seekg(prev );
            getline (file,line);
            pos=line.find(",");
            line.insert(pos+1,bbb);
            file.seekg(prev );
            file<<line<<endl;
            temp<<line<<endl;
            prev=file.tellg();
            line.clear();

        }else if (endlog>begin){
            file.close();
            log.close();
            temp.close();
            break;
        }
    }
}

```

```

    }
    file.close();
    log.close();

}
closeusb();
//copydata(c,tempfile);
}

```

Κώδικας Python για το αρχείο in\_vsm.exe:

```

import os
import time
import serial.tools.list_ports
import serial
import winsound
import ctypes
import sys

ser = serial.Serial()
tempfile = "C:\\QdVersaLab\\Data\\ltemp.dat"
logfile = "C:\\QdVersaLab\\Vsm\\LogFiles\\eomlog.txt"

def find_port(name="Arduino"):
    """a function that finds and returns
    in wich serial port Arduino is connected """
    ports = list(serial.tools.list_ports.comports())
    for p in ports:
        if name in p.description:
            print("Arduino's port:", p[0])
            return p[0]
        else:
            continue
    error_message("ERROR: can't find Arduino")

def connect_with_arduino(port):
    """this function intialise and stars
    serial communication with arduino """
    ser.baudrate = 9600
    ser.BYTESIZES = 8
    ser.port = port
    try:
        ser.open()
    except:
        error_message("ERROR: connection with Arduino failed")
    else:
        print("connect with Arduino:", True)

def last_mod_file():
    """this func finds the las modiefied file in a folder and sub-
    folders"""
    path = 'C:\\QdVersaLab\\Data'
    last_file = max((os.path.join(root, f) for root, _, the_files in
    os.walk(path)
                    for f in the_files if
    f.lower().endswith(".dat")), key=os.path.getctime)

```

```

if last_file is None:
    error_message("ERROR: cant find .dat file ")
else:
    print("\nopen file:", last_file)
return last_file

def log_file():
    """this func finds the eomlog file which written by multivu when
    a measurment is over.
    if log file deleted or doest exit it is created here"""
    try:
        log_file_size = os.stat(logfile).st_size
    except:
        print("file doesn't exist")
        log = open(logfile, "w+")
        log.close()
        log_file_size = os.stat(logfile).st_size
    return log_file_size

def get_field():
    """function that read the magnetic field from arduino"""
    send = "1"
    ser.flush()
    ser.write(send.encode())
    time.sleep(0.3)
    x = ser.read(ser.inWaiting())
    x = x.decode('utf-8')
    return x

def write_to_tempfile(tempf, l, mf):
    """function tha puts the magnetic field in the last line
    of VSM measurments and writew the line to a tempfile"""
    t = l.split(",")
    t[44] = mf
    an = ",".join(t)
    tempf.write(an)

def copy_data(dat_file, start):
    """function tha copies all measurments from temp file to the
    data section of data file"""
    with open(dat_file, "r+") as file, open(tempfile, "r") as temp:
        file.seek(start)
        while True:
            line = temp.readline()
            if not line:
                break
            file.write(line)

def measurment():
    """function that sinhronizes VSM with Arduino measurments
    by polling to data file written by VSM when it takes a
    measure"""
    data_file = last_mod_file()
    log_size = log_file()
    with open(data_file, "r") as f, open(tempfile, "w") as new:
        f.seek(0, 2)

```

```

size = f.tell()
offset = f.tell()
print("\n", "-" * 20, "start", "-" * 20)
while True:
    f.seek(0, 2)
    if size != f.tell():
        field = get_field()
        f.seek(size)
        line = f.readline()
        write_to_tempfile(new, line, field) #####
        f.seek(0, 2)
        size = f.tell()
    elif log_size < os.stat(logfile).st_size:
        print("end of measurments\n")
        print("copy data to file")
        break
    else:
        time.sleep(0.001)
        continue
copy_data(data_file, offset)
os.remove(tempfile)

def find_arduino(arg):
    if len(arg) > 1:
        flag = True
        for aa in arg:
            if "COM" in aa:
                return aa
        if flag:
            name = " ".join(arg[1:])
            arduinos_port = find_port(name)
            return arduinos_port
    else:
        arduinos_port = find_port()
        return arduinos_port

def error_message(message):
    """function th prints error massegas """
    winsound.PlaySound('SystemExit', winsound.SND_ALIAS)
    print("#" * 15, message, "#" * 15)
    ctypes.windll.user32.MessageBoxW(0, message, u"Error in_vsm.exe",
0)
    exit(1)

def main(arguments):
    arduinos_port = find_arduino(arguments)
    connect_with_arduino(arduinoo(arduinoo_port))
    meassurment()
    ser.close()
    exit(0)

if __name__ == '__main__':
    main(sys.argv)

```

### Κώδικας Arduino:

```
float vref=1.088;
float corr=vref/1.11187;
float R1=10.11;
float R2=0.986;
int offseterror=3;
float varduino=5030.0;
int n = 4;
void setup() {

    Serial.begin(9600);
    analogReference(INTERNAL);
    for (int k=0; k<6;k++){
        analogRead(A0);
    }
}

void loop() {
    float divfactor=R2/(R1+R2);
    int n_samples=1<<(2*n);
    if (Serial.available()>0){
        char a=Serial.read();
        Serial.flush();
        switch(a){
            case '1':
                uint32_t oversampled=0;
                for (int i=1; i<n_samples; i++){
                    oversampled=oversampled+(offseterror+analogRead(A0));
                }
                double scaled = (oversampled>>n);
                int sf=1<<n;
                double result=corr*scaled*vref/sf;
                result=(result/divfactor)-varduino;
                Serial.print(result);
                break;
            default:
                break;
        }
    }
}
```