

# Περιεχόμενα

Περίληψη	i
Abstract	iii
Εισαγωγή	v
Ευχαριστίες	ix
Περιεχόμενα	1
1. Παροχή νερού ενός ποταμού – Σύστημα υπολογισμού της στιγμιαίας παροχής	3
1.1 Τεχνικές υπολογισμού παροχής νερού ενός ποταμού	3
1.2 Συστήματα λήψης και επεξεργασίας δεδομένων (DAQ)	11
1.3 Αισθητήρες	11
1.4 Ενσωματωμένα συστήματα και το σύστημα του Arduino	12
2. Ανάπτυξη του συστήματος υπολογισμού της στιγμιαίας παροχής	17
2.1 Σύστημα μέτρησης της τρέχουσας ταχύτητας του νερού	18
2.2 Σύστημα υπολογισμού της στιγμιαίας παροχής	21
2.3 Αισθητήρες του συστήματος	24
2.3.1 Αισθητήρες μέτρησης της απόστασης	24
2.3.2 Αισθητήρας θερμοκρασίας – υγρασίας	27
2.3.3 Συστήματα μέτρησης της ταχύτητας του νερού	28
2.4 Σχεδιασμός του συστήματος	32
2.4.1 Σχεδιαστικό πακέτο OrCAD	33
2.4.2 Κάρτα προσαρμογής βύσματος τηλεπικοινωνιών βρετανικού τύπου σε Arduino	34
2.4.3 Κάρτα μετατροπής ψηφιακού σήματος σε πρότυπο 4-20mA	34
2.4.4 Κάρτα προσαρμογής του φωτοβολταϊκού πάνελ στη μπαταρία	37
2.4.5 Κάρτα μετατροπής 4-20mA σε 1-5V	38
2.4.6 Κάρτα μέτρησης θερμοκρασίας – υγρασίας	39
2.5 Υλοποίηση του συστήματος	39
2.5.1 Οι κάρτες του συστήματος	40
2.5.2 Σύστημα μέτρησης της τρέχουσας ταχύτητας του νερού	43
2.5.3 Υποσύστημα προσδιορισμού της στάθμης του νερού του ποταμού	43
2.5.4 Υποσύστημα υπολογισμού της στιγμιαίας παροχής νερού του ποταμού	44
3. Διαδικασία μέτρησης της παροχής νερού του ποταμού-Λογισμικό	47
3.1 Διαδικασία μέτρησης ταχύτητας του νερού σε κατακόρυφη τομή του ποταμού σε περιοχή όπου η κοίτη είναι σταθερή και η ταχύτητα εξαρτάται από τη θέση της επιφάνειας του νερού	47
3.2 Περιγραφή της διαδικασίας προσδιορισμού της στάθμης του νερού του ποταμού	50
3.3 Περιγραφή της διαδικασίας υπολογισμού της στιγμιαίας παροχής του νερού του ποταμού	51
4. Αποτίμηση του συστήματος	55
4.1 Έλεγχος κάρτας μετατροπής ψηφιακού σήματος σε πρότυπο 4-20mA	55
4.2 Έλεγχος κάρτας μετατροπής 4-20mA σε 1-5V	56
4.3 Διεπαφή του χρήστη	60
4.4 Αποτίμηση της μεθόδου υπολογισμού της παροχής	61
5. Συμπεράσματα	65
Αναφορές	67

Παράρτημα 1. Εναλλακτική πρόταση συστήματος τροφοδοσίας βασισμένο σε αξιοποίηση των διάχυτων ηλεκτρομαγνητικών κυμάτων του περιβάλλοντος	71
Αναφορές Παραρτήματος 1	76
Παράρτημα 2. Λογισμικό συστημάτων	77
Π2.1 Κώδικας υποσυστήματος υπολογισμού της στιγμιαίας παροχής νερού του ποταμού	77
Π2.2 Κώδικας υποσυστήματος προσδιορισμού της στάθμης του νερού του ποταμού	115
Π2.3 Κώδικας συστήματος μέτρησης της τρέχουσας ταχύτητας του νερού	119
Π2.4 Κώδικας για τον έλεγχο της κάρτας 4-20mA σε 1-5V	120
Π2.5 Κώδικας για τον έλεγχο της κάρτας μετατροπής ψηφιακού σήματος σε πρότυπο 4-20mA	121
Π2.6 Διαδικασία φόρτωσης του Bootloader στον μικροελεγκτή ATmega2560 του Arduino Mega 2560	123
Παράρτημα 3. Δημιουργία και τύπωση των πλακετών (PCB)	125
Π3.1 Τυπωμένα κυκλώματα πλακετών	125
Π3.1.1 Κάρτα προσαρμογής βύσματος τηλεπικοινωνιών βρετανικού τύπου σε Arduino	125
Π3.1.2 Κάρτα μετατροπής ψηφιακού σήματος σε πρότυπο 4-20mA	126
Π3.1.3 Κάρτα προσαρμογής του φωτοβολταϊκού πάνελ στη μπαταρία	127
Π3.1.4 Κάρτα μετατροπής 4-20mA σε 1-5V	128
Π3.1.5 Κάρτα μέτρησης θερμοκρασίας – υγρασίας	129
Π3.2 Διαδικασία δημιουργίας αρχείου Gerber	129
Παράρτημα 4. Απαραίτητες συνδέσεις – Καταναλώσεις	131
Π4.1 Ακροδέκτες συστήματος μέτρησης της τρέχουσας ταχύτητας του νερού	131
Π4.2 Συνδέσεις υποσυστήματος προσδιορισμού της στάθμης του νερού του ποταμού	132
Π4.3 Συνδέσεις υποσυστήματος υπολογισμού της στιγμιαίας παροχής νερού του ποταμού	133
Π4.4 Καταναλώσεις Συστημάτων – Αισθητήρων	133
Παράρτημα 5. Ανάλυση της ταχύτητας του νερού σε κατακόρυφη τομή του ποταμού σε περιοχή όπου η κοίτη είναι σταθερή και η ταχύτητα εξαρτάται από τη θέση της επιφάνειας του νερού	135
Π5.1 Γραφικές αναπαραστάσεις $\log z=f(u)$	135
Π5.2 Συναρτήσεις $\log z=f(u)$	140

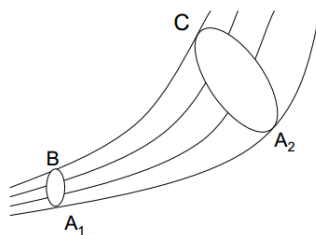
# 1. Παροχή νερού ενός ποταμού – Σύστημα υπολογισμού της στιγμιαίας παροχής

## 1.1 Τεχνικές υπολογισμού παροχής νερού ενός ποταμού

Κάθε ρευστό χαρακτηρίζεται από ιδιότητες όπως η πίεση, η πυκνότητα, η ταχύτητα ροής, η εσωτερική ενέργεια - θερμοκρασία, η δυναμική, κινητική και μηχανική ενέργεια και η δυναμική και κινηματική συνεκτικότητα.

Δύο είναι τα βασικά είδη της ροής, η στρωτή και η τυρβώδης. Ο χαρακτηρισμός αυτός της ροής γίνεται από δυναμική άποψη. Στρωτή είναι η ροή που μπορεί να περιγραφεί με την παραδοχή ότι το ρευστό αποτελείται από πολλά λεπτά στρώματα που ολισθαίνουν το ένα πάνω στο άλλο. Οι δυνάμεις συνεκτικότητας (δηλ. οι δυνάμεις τριβών που οφείλονται στην παρουσία στερεών τοιχωμάτων) είναι μεγαλύτερες από τις δυνάμεις αδράνειας. Οι στρωτές ροές αποτελούν θεωρητική περίπτωση και πολύ σπάνια συμβαίνουν στη φύση. Κατά την τυρβώδη ή στροβυλώδη ροή, οι γραμμές ροής του ρευστού λαμβάνουν τη μορφή ακανόνιστων καμπυλών οι οποίες τέμνουν συνεχώς αλλήλους, δίνοντας έτσι την εικόνα ροής με στροβιλισμούς. Εικόνες τυρβώδους ροής μας παρέχουν οι ποταμοί όταν παρουσιάζουν στροβίλους που μπορεί να οφείλονται σε υποκείμενα ρεύματα, ή σε τριβές από βραχώδεις όχθες. Η ροή γίνεται τυρβώδης όταν οι δυνάμεις αδράνειας είναι μεγαλύτερες από τις δυνάμεις συνεκτικότητας. Η κίνηση των πραγματικών ρευστών είναι πολύπλοκη και χαοτική αλλά κατά τμήματα μπορεί να θεωρηθεί ως στρωτή.

Τα ιδανικά ρευστά θεωρείται ότι έχουν στρωτή ροή η οποία δέν συναντά αντίσταση, είναι ασυμπίεστα και δεν εκτελούν περιστροφική κίνηση. Έστω ότι ένα κινούμενο ρευστό ακολουθεί τις ρευματικές γραμμές όπως φαίνεται στο Σχήμα 2.



Σχήμα 2. Οι ρευματικές γραμμές ενός κινούμενου ρευστού

Στο σημείο B το ρευστό κινείται με ταχύτητα  $u_1$ . Στο χρονικό διάστημα  $dt$ , ένα «σωματίδιο» του ρευστού θα διανύσει απόσταση  $u_1 \cdot dt$  και ο όγκος  $V = A_1 \cdot u_1 \cdot dt$  θα διέλθει από τη διατομή  $A_1$ . Εφόσον το ρευστό είναι ασυμπίεστο ο ίδιος όγκος θα περάσει και από το C. Εάν η ταχύτητα στο σημείο C είναι  $u_2$  τότε ισχύει ότι:

$$dV = A_1 \cdot u_1 \cdot dt = A_2 \cdot u_2 \cdot dt$$

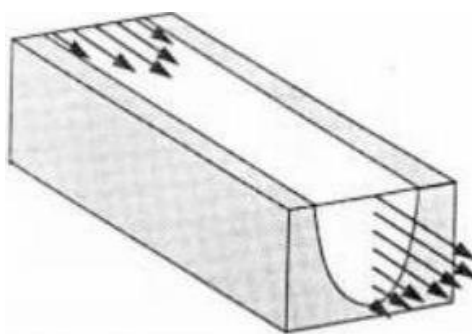
$$\text{ή } Q = \frac{dV}{dt} = A \cdot u = \text{σταθ} \quad (\text{εξίσωση συνέχειας})$$

Παροχή  $Q$  ονομάζεται ο όγκος  $dV$  ρευστού που διέρχεται από μία διατομή σε χρόνο  $dt$ , διά του χρόνου αυτού:  $Q = dV/dt$ . Μονάδα μέτρησης της παροχής είναι το  $\frac{1m^3}{sec}$ .

Η παροχή είναι σταθερή κατά μήκος οποιασδήποτε ροής. Η ροή είναι ταχύτερη στα στενότερα τμήματα ενός σωλήνα όπου οι ρευματικές γραμμές είναι πυκνότερες. Έτσι σε έναν ποταμό σταθερού πλάτους το ρηχό τμήμα του ποταμού έχει μικρότερη εγκάρσια διατομή και γρηγορότερο ρεύμα από το βαθύ τμήμα, αφού η παροχή είναι ίδια και στα δύο. Επομένως, το νερό θα έχει μεγαλύτερη ταχύτητα εκεί που το ποτάμι είναι ρηχό και μικρότερη εκεί όπου είναι βαθύ.

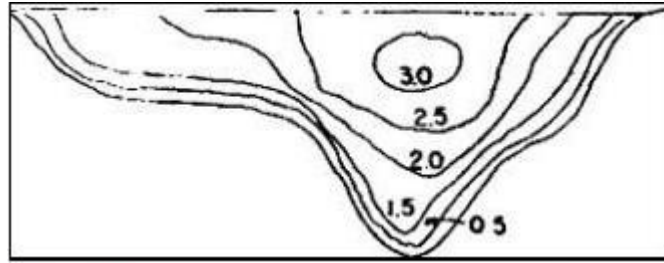
Βασική προϋπόθεση για την ύπαρξη ροής σε έναν ποταμό είναι τουλάχιστον αρχικά, μια μικρή κλίση στο ανάγλυφο ώστε να υπάρξει κίνηση του νερού λόγω βαρύτητας. Εκτός από τη βαρύτητα, σημαντικό ρόλο στη ροή ενός ποταμού παίζει η τριβή στα τοιχώματα της κοίτης του [3]. Η ταχύτητα ενός ποταμού μειώνεται σημαντικά στις ακτές και στο βυθό του λόγω της τριβής (Σχήμα 4). Η ταχύτητα του νερού ενός ποταμού εξαρτάται από τις δυνάμεις τριβής που αναπτύσσονται και αυτές προσδιορίζονται από το σχήμα και το μέγεθός του. Η μέγιστη ταχύτητα του νερού παρατηρείται στο κέντρο του ρεύματος και σε βάθος κάτω από την επιφάνεια, περίπου στο 1/3 της απόστασης επιφάνειας-βυθού. Για τη μελέτη της ροής ενός ποταμού, εκτός από την ταχύτητα, λαμβάνεται υπόψη η παροχή, που εκφράζει τον όγκο του νερού που διέρχεται από μία διατομή του ποταμού στη μονάδα του χρόνου.

Το νερό λοιπόν δεν ρέει με την ίδια ταχύτητα σε όλη την έκταση ενός ποταμού. Το μήκος των βελών στο Σχήμα 3 είναι ανάλογο της ταχύτητας του ρεύματος στο αντίστοιχο σημείο.



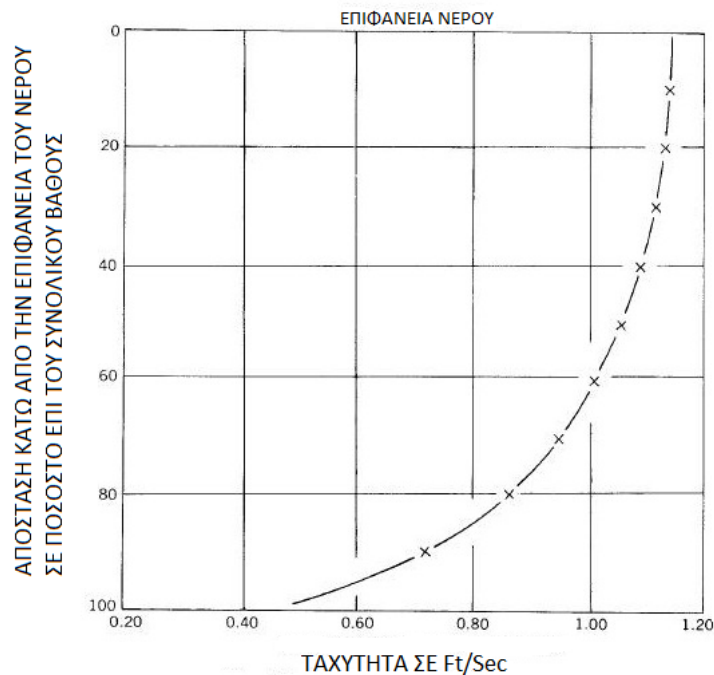
Σχήμα 3. Η ταχύτητα του νερού σε διάφορα σημεία ενός ποταμού

Στο Σχήμα 4 φαίνεται η κατανομή ταχύτητας σε μια διατομή ενός ποταμού.



Σχήμα 4. Η κατανομή της ταχύτητας στη διατομή ενός ποταμού σε ft/sec

Η μεγαλύτερη ταχύτητα σε m/sec, σημειώνεται λίγο κάτω από την επιφάνεια του νερού. Το προφίλ της ταχύτητας ενός ποταμού φαίνεται στο Σχήμα 5.



Σχήμα 5. Προφίλ ταχύτητας κάτω από την επιφάνεια ενός ποταμού

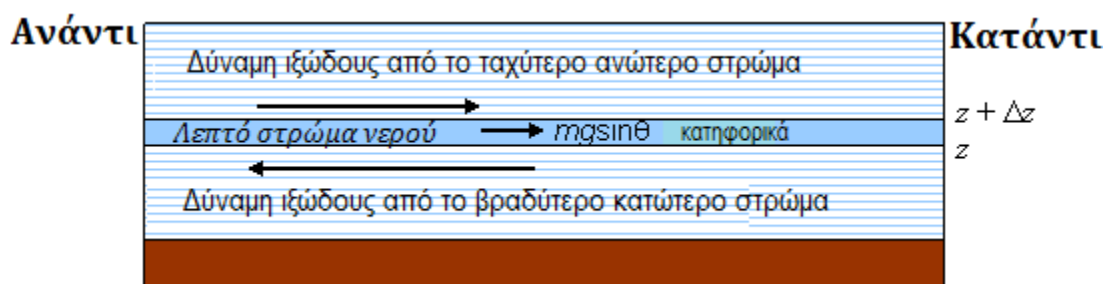
Το προφίλ της ταχύτητας ενός ποταμού έχει τη μορφή του Σχήματος 5, πράγμα το οποίο αποδεικνύεται από τη θεωρία [4], της οποίας η ανάλυση ακολουθεί.

Η ροή σε ένα ποτάμι, μεγάλο σε πλάτος σε σχέση με το βάθος του, το οποίο θεωρείται ότι είναι ομοιόμορφο και αγνοείται η τυρβώδης ροή κοντά στις όχθες μοιάζει με ένα πακέτο χαρτιού A4 σε μια επικλινή επιφάνεια. Το νερό στο ίδιο βάθος κινείται με σταθερή ταχύτητα. Το κάτω φύλλο παραμένει κολλημένο στο γραφείο, κάθε άλλο φύλλο κινείται προς τα κάτω στο επίπεδο, λίγο πιο γρήγορα από ότι το φύλλο που βρίσκεται αμέσως κάτω από αυτό. Η δύναμη ανα μονάδα επιφάνειας δίνεται από τη σχέση:

$$\frac{F}{A} = \eta \frac{dv(z)}{dz}$$

Όπου "η" το ιξώδες του νερού, "v" η ταχύτητα του νερού και "z" το ύψος μέσα στο ποτάμι.

Για ένα ποτάμι που ρέει σταθερά προς τα κάτω, σε μια ήπια κλίση λόγω της βαρύτητας, οι γραμμές ροής θεωρούνται ότι δείχνουν προς την ίδια κατεύθυνση. Εφόσον ο ποταμός είναι μεγάλος και το βάθος είναι μικρότερο από το πλάτος, μπορεί να γίνει η υπόθεση ότι σχεδόν όλη η ροή είναι μακριά από τα άκρα (όχθες του ποταμού). Έτσι η επιβράδυνση σε πρώτη προσέγγιση αγνοείται. Η συνάρτηση  $v(z)$  ονομάζεται "προφίλ ταχύτητας" και το μέγεθός της μικραίνει με το βάθος  $z$ , όπως φαίνεται στο Σχήμα 5. Αν απομονωθεί ένα λεπτό στρώμα του ποταμού, όπως ένα από τα φύλλα χαρτιού, που βρίσκεται σε ύψος μεταξύ  $z$  και  $z+\Delta z$  πάνω από τον πυθμένα, αυτό κινείται με σταθερή ταχύτητα, δηλαδή η συνισταμένη των δυνάμεων που ασκούνται στο λεπτό στρώμα του ποταμού θα είναι μηδενική. Όπως και το σύνολο του ποταμού, αυτό το στρώμα δεν είναι οριζόντιο, και το βάρος του έχει μία μικρή αλλά μη μηδενική συνιστώσα η οποία το εξαναγκάζει σε κίνηση κατάντι. Η συνιστώσα του βάρους εξισορροπείται από τη διαφορά μεταξύ της δύναμης ιξώδους από το ταχύτερο ανώτερο στρώμα νερού και από το βραδύτερο κατώτερο στρώμα. Οι αναφερόμενες δυνάμεις που ασκούνται στο λεπτό στρώμα νερού αναπαριστώνται στο Σχήμα 6.



Σχήμα 6. Διανυσματική ανάλυση ασκούμενων δυνάμεων σε ένα λεπτό στρώμα νερού του ποταμού

Αν το ποτάμι ρέει κατηφορικά με κάποια μικρή κλίση  $\theta$ , και θεωρηθεί ένα τετραγωνικό μέτρο στρώματος (όγκος  $\Delta z \text{ m} \times 1 \text{ m}^2 = \Delta z \text{ m}^3$ , πυκνότητα  $\rho$ ) το στρώμα δέχεται βαρυτική δύναμη  $mg \sin(\theta) \approx \rho g \Delta z \cdot \theta$  που το τραβάει κατάντι (θεωρώντας ότι η γωνία είναι μικρή,  $\sin \theta = \theta$ )

Το τετραγωνικό μέτρο του στρώματος δέχεται δύο δυνάμεις ιξώδους, μία από τα αργό νερό από κάτω, ίση με  $\eta \frac{dv(z)}{dz}$ , η οποία το επιβραδύνει, και μία από το πιο γρήγορο νερό από πάνω του  $\eta \frac{dv(z + \Delta z)}{dz}$ , η οποία το επιταχύνει.

Η βαρύτητα εξισορροπεί τη διαφορά μεταξύ των δύο δυνάμεων ιξώδους:

$$\rho g \theta \Delta z + \eta \frac{d}{dz} v(z + \Delta z) - \eta \frac{d}{dz} v(z) = 0$$

Διαιρώντας με  $\eta$  και  $\Delta z$  προκύπτει :

$$\frac{\frac{d}{dz} v(z + \Delta z) - \frac{d}{dz} v(z)}{\Delta z} = - \frac{\rho g \theta}{\eta}$$

Θεωρώντας ότι  $\Delta z \rightarrow 0$  και  $\frac{df(x)}{dx} = \lim_{\Delta x \rightarrow 0} \frac{f(x+\Delta x) - f(x)}{\Delta x}$

Προκύπτει η διαφορική εξίσωση :

$$\frac{d^2v(z)}{dz^2} = -\frac{\rho g \theta}{\eta}$$

η οποία έχει την λύση :

$$v(z) = -\frac{\rho g \theta}{2\eta} z^2 + Cz + D$$

Όπου C,D σταθερές που προσδιορίζονται από τις συνοριακές συνθήκες:

Στον πυθμένα η ταχύτητα είναι μηδενική δηλαδή για  $z=0$ ,  $v(z)=0$ . Και επομένως  $D=0$ .

Στο στρώμα της επιφάνειας του ποταμού που βρίσκεται σε επαφή με τον αέρα (άνεμος αμελητέος), η δύναμη ιξώδους από το ανώτερο στρώμα δεν υπάρχει και η εξίσωση ισορροπίας των δυνάμεων στο ανώτερο στρώμα είναι :

$$\rho g \theta \Delta z - \eta \frac{d}{dz} v(z) = 0$$

Εάν θεωρηθεί ότι το ανώτερο στρώμα είναι παρα πολύ λεπτό, δηλαδή  $\Delta z \rightarrow 0$ , ο όρος  $\rho g \theta \Delta z$  τείνει στο μηδέν, και

$$\frac{d}{dz} v(z) = 0$$

Αφού το ιξώδες "η" είναι σταθερά διάφορη του μηδενός.

Το προφίλ της ταχύτητας παίρνει την εξής μορφή:

$$v(z) = \frac{-\rho g \theta}{2\eta} z^2 + Cz$$

Παραγωγίζοντας την εξίσωση προκύπτει ότι  $\frac{dv(z)}{dz} = -\frac{\rho g \theta}{2\eta} z + C$

Λαμβάνοντας υπόψη πως  $\frac{dv(z=h)}{dz} = 0$ ,

η σταθερά C προσδιορίζεται:  $C = \frac{\rho g \theta h}{\eta}$

Συνεπώς η ταχύτητα του νερού σε μια κατακόρυφη τομή, κάθετα στη διεύθυνση κίνησής του νερού του ποταμού, σε συνάρτηση με το ύψος z του νερού του παίρνει τη μορφή:

$$v(z) = \frac{\rho g \theta}{2\eta} z(2h - z)$$

Υπάρχουν διάφοροι μέθοδοι για τη μέτρηση της ταχύτητας του νερού ενός ποταμού. Οι πιο διαδεδομένοι τρόποι μέτρησης ταχύτητας [5] είναι:

(α) Με πλωτήρες. Αυτή η μέθοδος είναι απλή. Μετρώντας το χρόνο κίνησης του πλωτήρα σε μια γνωστή απόσταση μπορεί να υπολογιστεί η επιφανειακή ταχύτητα.

(β) Με μυλίσκους. Μετράται η ταχύτητα σε διάφορα σημεία της επιφάνειας τα οποία είναι συστηματικά κατανεμημένα κατα πλάτος μίας διατομής.

(γ) Ηλεκτρομαγνητικός μετρητής ροής. Οι ηλεκτρομαγνητικοί μετρητές ροής αξιοποιούν το νόμο επαγωγής του Faraday, σύμφωνα με τον οποίο όταν ένας αγωγός κινείται κάθετα μέσα σε ένα μαγνητικό πεδίο, προκαλείται διαφορά δυναμικού από επαγωγή η οποία είναι ανάλογη της ταχύτητάς του. Στα ηλεκτρομαγνητικά ροόμετρα, το ρόλο του αγωγού έχει το νερό και η τάση από επαγωγή μετράται από ηλεκτρόδια που έχουν τοποθετηθεί σε ένα πηνίο το οποίο είναι τοποθετημένο κάτω από τον πυθμένα εγκάρσια του ποταμού. Αυτή η μέθοδος είναι αρκετά δαπανηρή αλλά και περίπλοκη.

(δ) Υπερηχητικός μετρητής ροής. Τα υπερηχητικά ροόμετρα αξιοποιούν το φαινόμενο Doppler. Εκπέμπουν κύματα υπερήχων, υπό γωνία  $\theta$  πάνω στο νερό που κινείται και λαμβάνουν την ανάκλαση των κυμάτων αυτών. Ένα μέρος της εκπεμπόμενης ακτινοβολίας ανακλάται πάνω σε στερεά σωματίδια που υπάρχουν μέσα στο νερό ή στις διαχωριστικές επιφάνειες που δημιουργούνται από τις φυσαλίδες αέρα που υπάρχουν μέσα σε αυτό. Η διαφορά στη συχνότητα μεταξύ του εκπεμπόμενου και του λαμβανόμενου σήματος (συχνότητα Doppler), είναι ανάλογη της ταχύτητας του νερού. Έτσι υπολογίζεται η ταχύτητα του υδατορεύματος. Αυτή η μέθοδος απαιτεί πολύπλοκο ηλεκτρονικό εξοπλισμό και μπορεί να χρησιμοποιηθεί μόνο σε καθαρό νερό και ποταμούς σταθερού πυθμένα.

(ε) Ενσωματωμένοι πλωτήρες. Εδώ οι πλωτήρες είναι κινούμενες φυσαλίδες αέρα που ελευθερώνονται από ακροφύσια τα οποία είναι τοποθετημένα σε κανονικά διαστήματα από έναν σωλήνα με συμπιεσμένο αέρα που εκτείνεται εγκάρσια του πυθμένα του ρεύματος. Η έκταση της διασποράς των ανερχόμενων φυσαλίδων στην επιφάνεια του νερού δίνει το μέτρο της μέσης ταχύτητας του ρεύματος.

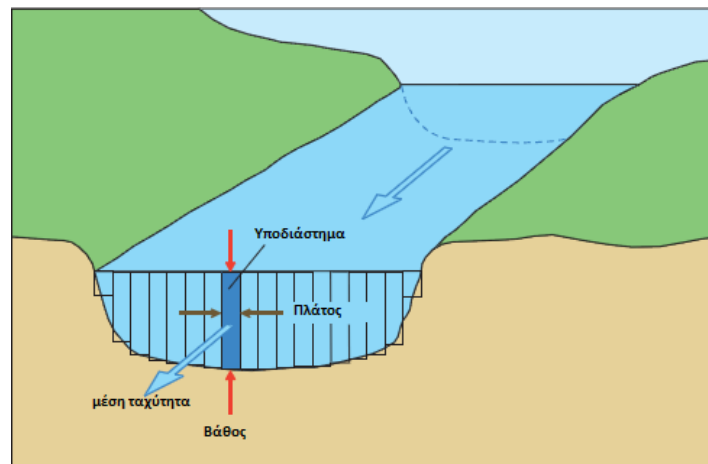
(στ) Με χρήση ροομέτρων. Ο πιο οικονομικός αλλά και εύκολος τρόπος να μετρηθεί η ταχύτητα σε διάφορα σημεία κατά μήκος, πλάτος και βάθος ενός ποταμού, είναι με τη χρήση ροομέτρων.

Η διαδικασία της μέτρησης της ταχύτητας του νερού ενός ποταμού, μπορεί να απλουστευθεί εάν η κοίτη του ποταμού είναι σταθερή, χωρίς μεγάλες πέτρες που προεξέχουν και εμπόδια όπως προβλήτες που θα μπορούσαν να προκαλέσουν αναταράξεις και τυρβώδη ροή και εάν είναι επίπεδη, για την εξάλειψη κάθετων συνιστωσών της ταχύτητας.

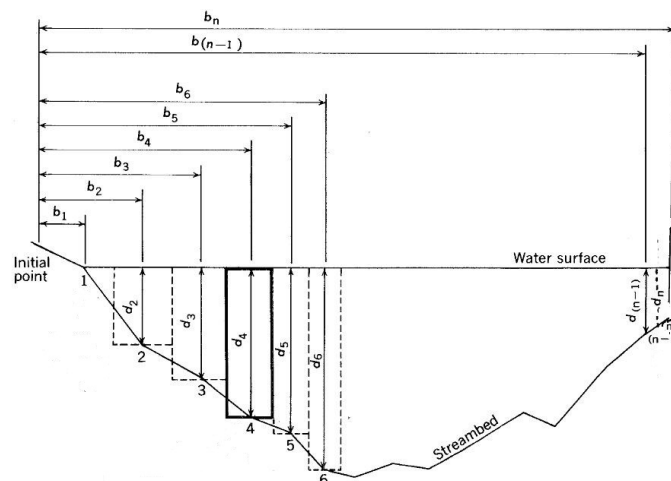
Για τον υπολογισμό της παροχής σε ποταμούς με μεγάλο βάθος και πλάτος, όπου υπάρχει μεγάλη ποσότητα νερού, χρησιμοποιούνται μέθοδοι ολοκλήρωσης στοιχειωδών παροχών, που προκύπτουν από τη μέτρηση των σημειακών ταχυτήτων σε διάφορα σημεία μιας διατομής. Για τη μέτρηση της παροχής, εξετάζεται σε ένα σημείο του ποταμού, μια διατομή κάθετη ως προς τη διεύθυνση του ρεύματος, της



οποίας ο πυθμένας είναι κατά το δυνατόν σταθερός. Η διατομή διαιρείται σε κατακόρυφες νοητές λωρίδες όπως φαίνεται στα Σχήματα 7 και 8.



Σχήμα 7. Διατομή της κοίτης ενός ποταμού διαιρεμένη σε κατακόρυφες νοητές λωρίδες



Σχήμα 8. Διατομή της κοίτης ενός ποταμού χωρισμένη σε στοιχειώδη υπομήματα

Το εμβαδόν του κάθε υπομήματος υπολογίζεται είναι :

$$q_x = u_x \left[ \frac{(b_x - b_{(x-1)})}{2} + \frac{(b_{(x+1)} - b_x)}{2} \right] d_x$$

$$= u_x \left[ \frac{b_{(x+1)} - b_{(x-1)}}{2} \right] d_x,$$

όπου  $u_x$  η μέση ταχύτητα του υπομήματος.

Απαιτούνται τουλάχιστον 25-30 τμήματα για τους περισσότερους ποταμούς και ακόμη περισσότερα όταν έχουμε πιο πολύπλοκες διατομές (ύπαρξη βράχων, μεγάλη αυξομείωση του βάθους κατά μήκος της κοίτης). Η συνολική παροχή προκύπτει από τη σχέση:

$$Q = \sum A_i u_i$$

Όπου  $A_i$  το εμβαδόν κάθε λωρίδας και  $u_i$  η μέση ταχύτητά της.

Ο υπολογισμός της μέσης ταχύτητας σε κάθε υποτήμα προϋποθέτει μετρήσεις σε αρκετά σημεία βάθου. Έξι μέθοδοι [6] χρησιμοποιούνται για τον προσδιορισμό της μέσης ταχύτητας στα υποτήματα της διατομής της κοίτης του ποταμού.

#### (1) Μέθοδος καμπύλης κάθετης ταχύτητας

Οι μετρήσεις λαμβάνονται με διάστημα 0.1 βαθμών του βάθους, μεταξύ του 0.1 και 0.9 του βάθους. Είναι καλή μέθοδος αλλά απαιτεί χρόνο για τη συλλογή των δεδομένων

#### (2) Μέθοδος δύο σημείων

Οι μετρήσεις γίνονται στα 0.2 και 0.8 του βάθους. Ο μέσος όρος των δύο αυτών σημειακών ταχυτήτων λαμβάνεται ως ο μέσος όρος του κάθε υποτήματος.

$$u_i = \frac{u_{0.2} + u_{0.8}}{2}$$

Παρόλο που είναι προσεγγιστική, αυτή η μέθοδος χρησιμοποιείται από το Γεωλογικό Ινστιτούτο των ΗΠΑ [7].

#### (3) Έξι δέκατα του βάθους

Σε αυτή τη μέθοδο πραγματοποιείται μια μέτρηση της ταχύτητας στο 0.6 του βάθους κάτω από την επιφάνεια του νερού και χρησιμοποιείται ως η μέση ταχύτητα στην κατακόρυφο. Δίνει αξιόπιστα αποτελέσματα όταν το βάθος κυμαίνεται από 0.1 m έως 0.8 m, όταν υπάρχουν μεγάλες ποσότητες λάσπης ή πάγου και όταν η στάθμη δεν είναι σταθερή και η μέτρηση πρέπει να γίνει γρήγορα.

#### (4) Δύο δέκατα του βάθους

Όταν η στάθμη δεν είναι σταθερή και πρέπει να γίνει γρήγορα η μέτρηση, ή όταν οι ταχύτητες του νερού είναι μεγάλες, δεν μπορεί να τοποθετηθεί ο μετρητής στο 0.6 και 0.8 του βάθους. Τότε γίνεται μια μέτρηση στο 0.2 του βάθους.

#### (5) Μέθοδος τριών σημείων

Σε αυτή τη μέθοδο λαμβάνονται μετρήσεις στα 0, 0.2, 0.6 και 0.8 του βάθους, έτσι συνδυάζει τις μεθόδους (2) και (3). Η μέση τιμή υπολογίζεται από την σχέση:

$$u_i = \frac{u_{0.6}}{2} + \frac{u_{0.2} + u_{0.8}}{4}$$

Η μέθοδος αυτή χρησιμοποιείται για βάθος μεγαλύτερο των 0.8 m.

#### (6) Υπόγεια μέτρηση

Αυτή η μέθοδος χρησιμοποιείται για μέτρηση ταχύτητας σε κάποια απόσταση κάτω από το νερό (τουλάχιστον 0.6 m) όταν είναι αδύνατο να εκτιμηθεί το βάθος και έχει μικρή αξιόπιστία.

Στην παρούσα διπλωματική εργασία χρησιμοποιήθηκε μία διαφορετική μέθοδος με μέτρηση της ταχύτητας του νερού σε σημεία κατανεμημένα κάτω από την επιφάνεια τα οποία είναι σε απόσταση 0.5m μεταξύ τους. Στη συνέχεια έγινε προσδιορισμός κάποιων παραγόντων ώστε να μπορεί να υπολογίζεται η ταχύτητα σε κάθε σημείο κάτω από την επιφάνεια για διαφορετικές τιμές της στάθμης του νερού (Κεφάλαιο 3).

## 1.2 Συστήματα λήψης και επεξεργασίας δεδομένων (DAQ)

Η Λήψη Δεδομένων (Data Acquisition – DAQ) [8] περιλαμβάνει τη συλλογή των σημάτων από τις πηγές μέτρησης καθώς και την ψηφιοποίηση, αποθήκευση και ανάλυσή τους. Τα συστήματα συλλογής δεδομένων περιλαμβάνουν :

- Αισθητήρες που μετατρέπουν φυσικές παραμέτρους σε ηλεκτρικά σήματα.
- Κύκλωμα ρύθμισης σήματος για τη μετατροπή των σημάτων από τους αισθητήρες σε μια μορφή που να μπορούν να μετατραπούν σε ψηφιακές τιμές.
- Μετατροπείς αναλογικού σήματος σε ψηφιακό, οι οποίοι μετατρέπουν τα ρυθμισμένα σήματα του αισθητήρα σε ψηφιακές τιμές.
- Λογισμικό Εφαρμογών.

## 1.3 Αισθητήρες

Οι αισθητήρες είναι όργανα που μετατρέπουν φυσικές παραμέτρους σε ηλεκτρικά σήματα και χωρίζονται σε δύο μεγάλες κατηγορίες. Τους αναλογικούς και τους ψηφιακούς. Οι αναλογικοί αισθητήρες παράγουν σήμα - τάση (της τάξης mV ή και V) το οποίο μέσω κάποιων μαθηματικών εξισώσεων αντιστοιχίζεται σε κάποια τιμή του μετρούμενου φυσικού μεγέθους. Σε αντίθεση με τους αναλογικούς, οι ψηφιακοί επιστρέφουν στην έξοδό τους τη μέτρηση του φυσικού μεγέθους υπο μορφή ψηφιακής πληροφορίας.

Οι αισθητήρες ανάλογα με τη λειτουργία τους ή τη φυσική αρχή στην οποία στηρίζεται η λειτουργία τους [9], ταξινομούνται σε :

- *Θερμικοί* - θερμοκρασία, ροή θερμότητας, εντροπία (π.χ. τα θερμόμετρα)
- *Ακτινοβολίας* - ακτίνες γ, ορατή, υπέρυθρη ακτινοβολία, μικροκύματα, (π.χ. οι ανιχνευτές αερίου, οι υγροί σπινθηριστές, ο σωλήνας Geiger–Müller)
- *Μηχανικοί* - μετατόπιση, ταχύτητα, δύναμη, πίεση (π.χ. τα radar)
- *Μαγνητικοί* - μαγνητικό πεδίο, μαγνητική ροπή (π.χ. οι ανιχνευτές μετάλλων)
- *Χημικοί* - PH, συγκέντρωση αερίων (π.χ. οι αισθητήρες pH)
- *Βιολογικοί* - πρωτεΐνες, ορμόνες
- *Ηλεκτρικοί* - τάση, αγωγιμότητα, επαγωγή, πόλωση (π.χ. τα αμπερόμετρα)
- *Οπτικοί* – μήκος κύματος, φάσης, απορρόφησης (π.χ. οι φωτοδιόδοι)

Τα κύρια χαρακτηριστικά γνωρίσματα των αισθητήρων [10] είναι το εύρος, η ακρίβεια και το σφάλμα.

## 1.4 Ενσωματωμένα συστήματα και το σύστημα του Arduino

Ενσωματωμένο σύστημα θεωρείται οποιαδήποτε συσκευή η οποία περιλαμβάνει κάποιον επεξεργαστή ή μικροελεγκτή, στην οποία λειτουργεί συγκεκριμένη εφαρμογή. Σε ένα ενσωματωμένο σύστημα εν γένει ο χρήστης δεν έχει τη δυνατότητα αλλαγής της λειτουργικότητας της εφαρμογής. Ένα χαρακτηριστικό παράδειγμα ενσωματωμένου συστήματος είναι το κινητό τηλέφωνο.

Το κυριότερο στοιχείο ενός ενσωματωμένου συστήματος είναι ο μικροελεγκτής (*microcontroller*) [11]. Ένας μικροελεγκτής είναι ένα ολοκληρωμένο κύκλωμα το οποίο περιέχει ένα κεντρικό στοιχείο για εκτέλεση αριθμητικών ή λογικών πράξεων. Επιπλέον έχει ενσωματωμένα πολλά περιφερειακά κυκλώματα καθιστώντας τον ικανό να λειτουργεί με ελάχιστα εξωτερικά εξαρτήματα. Ο μικροελεγκτής δηλαδή είναι μία επέκταση του μικροεπεξεργαστή [12] προορισμένος όμως για συγκριμένες εργασίες οι οποίες καθορίζονται από την εφαρμογή που λειτουργεί εντός του. Οι μικροελεγκτές χρησιμοποιούνται ευρύτατα σε όλα τα ενσωματωμένα συστήματα (*embedded systems*) ελέγχου χαμηλού και μεσαίου κόστους, όπως αυτά που χρησιμοποιούνται σε αυτοματισμούς, ηλεκτρονικά καταναλωτικά προϊόντα (από ψηφιακές φωτογραφικές μηχανές έως παιχνίδια), ηλεκτρικές συσκευές και κάθε είδους αυτοκινούμενα τροχοφόρα οχήματα. Για τη λειτουργία ενός πλήρους ενσωματωμένου υπολογιστικού συστήματος, απαιτούνται πολλά υποσυστήματα ή περιφερειακά τα οποία είναι τοποθετημένα μέσα στο ίδιο ολοκληρωμένο (IC). Στους σύγχρονους μικροεπεξεργαστές για μη ενσωματωμένα συστήματα (π.χ. τους μικροεπεξεργαστές των προσωπικών υπολογιστών), δίνεται έμφαση στην υπολογιστική ισχύ. Η ευελιξία ανάπτυξης διαφορετικών εφαρμογών είναι μεγάλη και η λειτουργικότητα του τελικού συστήματος καθορίζεται από τα εξωτερικά περιφερειακά τα οποία διασυνδέονται με την κεντρική μονάδα (μικροεπεξεργαστή), η οποία δεν είναι εξειδικευμένη. Αντίθετα, στους μικροελεγκτές οι οποίοι έχουν μικρότερες δυνατότητες συνεργασίας με εξωτερικά περιφερειακά, η ευελιξία είναι περιορισμένη, καθώς και η υπολογιστική ισχύς.

Οι θέσεις όπου εγκαθίστανται οι μικροελεγκτές μαζί με τα αισθητήρια και τα υπόλοιπα στοιχεία (π.χ. κεραίες) τα οποία είναι απαραίτητα για τη λειτουργία του συστήματος ονομάζονται σταθμοί ή κόμβοι. Οι σταθμοί μπορούν επίσης να επικοινωνούν με άλλους ή με κάποιον κεντρικό σταθμό.

Το σύστημα που χρησιμοποιήθηκε για τον τοπικό σταθμό υπολογισμού της παροχής αποτελείται από δύο διασυνδεδεμένα υποσυστήματα. Το ένα είναι βασισμένο στον Arduino Mega 1280 και το δεύτερο στον Arduino Mega 2560 [13]. Για το κάθε υποσύστημα αναπτύχθηκε διαφορετική εφαρμογή. Ο Arduino είναι μια απλή μητρική πλακέτα με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους [14]. Είναι ανοικτού κώδικα και μπορεί να προγραμματιστεί με τη γλώσσα Wiring [15]. Ο Arduino είναι σαν ένας μικροσκοπικός υπολογιστής που μπορεί να προγραμματιστεί ώστε να διαχειρίζεται εισόδους και εξόδους μεταξύ αυτού και άλλων εξωτερικών στοιχείων που συνδέονται σε αυτόν. Μπορεί να αλληλεπιδρά με το περιβάλλον κάνοντας λήψη και αποστολή σημάτων μέσα από μια ποικιλία αισθητήρων. Επίσης παρέχεται η δυνατότητα σύνδεσης σε έναν υπολογιστή, μέσω μιας θύρας USB που διαθέτει, ένα δίκτυο ή ακόμα και το διαδίκτυο για την ανάκτηση και αποστολή δεδομένων, δια μέσω μιας θύρας Ethernet. Το σχηματικό διάγραμμα και οι πληροφορίες για το υλικό είναι

ελεύθερα και διαθέσιμα για αυτούς που θέλουν να συναρμολογήσουν τον Arduino μόνοι τους.

Κάθε πλακέτα Arduino Mega αποτελείται από ένα μικροελεγκτή ATmega της Atmel AVR και κάποια συμπληρωματικά εξαρτήματα τα οποία χρησιμεύουν στον προγραμματισμό και επιτρέπουν την ενσωμάτωση του Arduino σε άλλα κυκλώματα. Ο μικροελεγκτής από κατασκευής εμπεριέχει στη μνήμη του ένα πρόγραμμα φόρτωσης λογισμικού – εκκίνησης λειτουργικού (bootloader) [16], έτσι ώστε να μην χρειάζεται εξωτερικός προγραμματιστής. Το πιο σημαντικό κομμάτι που αφορά τον Arduino είναι ο τυποποιημένος τρόπος που είναι εκτεθειμένες οι υποδοχές, επιτρέποντας στην πλακέτα να συνδεθεί με μια ποικιλία από εναλλάξιμες πρόσθετες μονάδες γνωστές ως shields [17]. Shields είναι τα εξαρτήματα που συνδέονται απευθείας με όλους τους απαραίτητους ακροδέκτες του Arduino. Μερικά shields επικοινωνούν με την πλακέτα του Arduino άμεσα μέσω ακροδεκτών, αλλά πολλά είναι μεμονωμένα προσπελάσιμα μέσω ενός σειριακού διαύλου I<sup>2</sup>C [18].







Το λογισμικό αποτελείται από έναν τυπικό μεταγλωττιστή γλώσσας προγραμματισμού ο οποίος λειτουργεί σε Η/Υ και το σύστημα φόρτωσης λογισμικού – εκκίνησης λειτουργικού (bootloader) που λειτουργεί στην πλακέτα. Το υλικό του Arduino μπορεί να προγραμματιστεί χρησιμοποιώντας ένα αναπτυξιακό περιβάλλον (Integrated Development Environment – IDE), το οποίο είναι μια εφαρμογή η οποία λειτουργεί σε Η/Υ και είναι γραμμένη σε γλώσσα Java [19]. Η γλώσσα του Arduino βασίζεται στη γλώσσα Wiring [13] μια παραλλαγή C/C++ [20] για μικροελεγκτές αρχιτεκτονικής AVR όπως ο ATmega, και υποστηρίζει όλες τις βασικές δομές της C καθώς και μερικά χαρακτηριστικά της C++. Για μεταγλωττιστής (compiler) χρησιμοποιείται ο AVR gcc και ως βασική βιβλιοθήκη C χρησιμοποιείται η AVR libc. Το λογισμικό του αναπτυξιακού περιβάλλοντος περιλαμβάνει επίσης μια σειριακή οθόνη που επιτρέπει απλά κειμενικά δεδομένα να σταλούν από την πλακέτα του Arduino. Τα RX και TX LEDs της πλακέτας ενεργοποιούνται όταν μεταδίδονται τα δεδομένα μέσω του ολοκληρωμένου κυκλώματος FTDI [21] που είναι ενσωματωμένο στη πλακέτα του Arduino. Μια βιβλιοθήκη λογισμικού περιέχει συναρτήσεις για τη σειριακή επικοινωνία με οποιοδήποτε από τους ψηφιακούς ακροδέκτες του Arduino Mega. Ο ATmega επιπλέον, υποστηρίζει επικοινωνία I<sup>2</sup>C [18] και SPI [22]. Το λογισμικό του Arduino περιλαμβάνει μια βιβλιοθήκη Wire για απλοποίηση της χρήσης του διαύλου I<sup>2</sup>C.

Το περιβάλλον ανάπτυξης Arduino [23] περιέχει ένα πρόγραμμα επεξεργασίας κειμένου, για τη σύνταξη του κώδικα της εφαρμογής, μια περιοχή στην οποία εμφανίζονται μηνύματα κατά τη μετάφραση της εφαρμογής, μία κονσόλα κειμένου όπου εμφανίζεται ο κώδικας που συντάσσεται (sketch) και μια γραμμή εργαλείων υπό μορφή ενεργών επιλογών. Το αναπτυξιακό αυτό περιβάλλον συνδέεται δια μέσω των θυρών USB του Η/Υ στον οποίο και λειτουργεί, με την πλακέτα του Arduino, για να φορτώσει προγράμματα και να επικοινωνεί μαζί της. Το Arduino IDE είναι βασισμένο σε Java και συγκεκριμένα παρέχει:

- Ένα πρακτικό περιβάλλον για τη συγγραφή των προγραμμάτων, με συντακτική χρωματική σήμανση.
- Μερικές έτοιμες βιβλιοθήκες.
- Τον μεταγλωττιστή (compiler) για τη μεταγλώττιση των προγραμμάτων.

- Μία σειριακή οθόνη (serial monitor), η οποία εμφανίζει τα σειριακά δεδομένα που αποστέλλονται από την πλακέτα Arduino.
- Την επιλογή για φόρτωση των μεταγλωττισμένων προγραμμάτων στον Arduino.

Στον Πίνακα 1 παρουσιάζονται τα εργαλεία του περιβάλλοντος ανάπτυξης, υπό μορφή επιλογών.

	Verify	Ελέγχει για συντακτικά λάθη στον κώδικα.
	Upload	Μεταγλωττίζει τον κώδικα και τον φορτώνει στο Arduino.
	New	Δημιουργεί ένα νέο sketch.
	Open	Παραθέτει ένα μενού με όλα τα sketch. Κάνοντας κλικ σε ένα από αυτά, θα ανοίξει αυτόματα στο τρέχον παράθυρο.
	Save	Αποθηκεύει ένα sketch.
	Serial Monitor	Ανοίγει την σειριακή οθόνη.

Πίνακας 1. Εργαλεία του περιβάλλοντος ανάπτυξης Arduino

Για τη μεταφορά ενός προγράμματος στη μνήμη του Arduino (uploading), απαιτείται η επιλογή του συγκεκριμένου τύπου της πλακέτας Arduino και της σειριακής θύρας που χρησιμοποιείται, στις επιλογές του περιβάλλοντος IDE.

Μία από τις εκδόσεις του Arduino είναι ο Arduino Mega που φαίνεται στο Σχήμα 9 και πάνω στον οποίο στηρίχθηκε η κατασκευή του τοπικού συστήματος της παρούσας Διπλωματικής Εργασίας. Ο Arduino Mega έχει δύο παραλλαγές, τον 1280 και τον 2560. Ο Arduino Mega 1280 είναι μια πλακέτα που βασίζεται στον μικροελεγκτή ATmega1280 της εταιρείας Atmel [24] ενώ ο Arduino Mega 2560 βασίζεται στον μικροελεγκτή ATmega2560 της ίδιας εταιρείας.



Σχήμα 9. Η πλακέτα του Arduino Mega 2560

Και οι δύο πλακέτες διαθέτουν οτιδήποτε είναι απαραίτητο για την ανάπτυξη μίας εφαρμογής. Τα βασικά χαρακτηριστικά των Arduino Mega 1280 και Arduino Mega 2560 παρουσιάζονται στον Πίνακα 2.

Μικροελεγκτής	ATmega1280	ATmega2560
Τάση λειτουργίας	5V	5V
Τάση εισόδου (ενδεδειγμένη)	7-12V	7-12V
Τάση εισόδου (όρια)	6-20V	6-20V
Ψηφιακοί ακροδέκτες	54 (15 παρέχουν PWM έξοδο)	54 (15 παρέχουν PWM έξοδο)
Αναλογικοί ακροδέκτες εισόδου	16	16
Ρεύμα DC ανά ακροδέκτη	40 mA	40 mA
Ρεύμα DC για ακροδέκτες 3.3V	50 mA	50 mA
Μνήμη Flash	128 Kb (4Kb για τον bootloader)	256 Kb (8Kb για τον bootloader)
SRAM	8 Kb	8 Kb
EEPROM	4Kb	4Kb
Ταχύτητα ρολογιού	16 MHz	16 MHz

Πίνακας 2. Χαρακτηριστικά των Arduino Mega 1280 και Arduino Mega 2560

Η μόνη διαφορά μεταξύ των δύο παρατηρείται στη μνήμη flash. Ο Arduino Mega 1280 έχει 128KB ενώ ο Arduino Mega 2560 έχει 256KB.

Ο Arduino μπορεί να τροφοδοτηθεί με ρεύμα είτε από τον υπολογιστή μέσω της σύνδεσης στη θύρα USB, είτε από εξωτερική τροφοδοσία που παρέχεται μέσω μιας διεπαφής (βύσμα των 2.1mm, με θετικό πόλο στο κέντρο) και βρίσκεται στην κάτω αριστερή γωνία του Arduino όπως φαίνεται στο Σχήμα 9. Για την αποφυγή προβλημάτων, η εξωτερική τροφοδοσία θα πρέπει να κυμαίνεται από 7 ως 12V.

Ο ακροδέκτης με την ένδειξη RESET, όταν γειωθεί (σε οποιοδήποτε από τα 3 pin με την ένδειξη GND που υπάρχουν στον Arduino) έχει ως αποτέλεσμα την επανεκκίνηση του Arduino. Ο ακροδέκτης με την ένδειξη 3.3V, μπορεί να τροφοδοτήσει τα διάφορα εξαρτήματά με τάση 3.3V. Η τάση αυτή παράγεται από το ολοκληρωμένο FTDI[21]. Ο τρίτος ακροδέκτης με την ένδειξη 5V, μπορεί να τροφοδοτήσει εξωτερικά εξαρτήματα με τάση 5V. Ο τέταρτος και ο πέμπτος ακροδέκτης με την ένδειξη GND, είναι γειώσεις. Ο έκτος ακροδέκτης, με την ένδειξη Vin έχει διπλό ρόλο. Σε συνδυασμό με τον ακροδέκτη γείωσης δίπλα του, μπορεί να λειτουργήσει ως μέθοδος εξωτερικής τροφοδοσίας του Arduino, ή μπορεί να χρησιμοποιηθεί για να τροφοδοτηθούν εξαρτήματα με την πλήρη τάση της εξωτερικής τροφοδοσίας (7~12V), πριν αυτή περάσει από τον ρυθμιστή τάσης όπως γίνεται με το pin των 5V.

Οι μικροεπεξεργαστές ATmega1280 και ATmega2560 έχουν τρεις ομάδες μνήμης. Διαθέτουν *Flash memory*, στην οποία αποθηκεύονται οι εφαρμογές, *SRAM* (static random access memory), στην οποία δημιουργείται το sketch και χρησιμοποιεί τις μεταβλητές όταν τρέχει, και *EEPROM*, η οποία χρησιμοποιείται από τους προγραμματιστές για την αποθήκευση μακροχρόνιων πληροφοριών.

Η πλακέτα του Arduino Mega περιέχει 54 ψηφιακούς ακροδέκτες που φαίνονται στο Σχήμα 9, οι οποίοι είναι αριθμημένοι και μπορούν να χρησιμοποιηθούν σαν είσοδοι ή έξοδοι. Η λειτουργία για κάθε έναν από τους ακροδέκτες, παρουσιάζεται αναλυτικά παρακάτω.

**Ακροδέκτες 0, 1 και 14 έως 19:** Λειτουργούν ως RX και TX της σειριακής θύρας όταν το πρόγραμμα ενεργοποιεί τη σειριακή θύρα.

**Ακροδέκτες 2 , 3, 18, 19, 20 και 21:** Ρυθμίζονται μέσα από το πρόγραμμα ώστε να λειτουργούν αποκλειστικά ως ψηφιακές είσοδοι στις οποίες όταν συμβαίνουν συγκεκριμένες αλλαγές (interrupts), η κανονική ροή του προγράμματος σταματάει άμεσα και εκτελείται μια συγκεκριμένη συνάρτηση.

**Ακροδέκτες 2 έως 13 και 44 έως 46:** Μπορούν να λειτουργήσουν ως ψευδο-αναλογικές έξοδοι με το σύστημα PWM (Pulse Width Modulation) [25].

**Ακροδέκτες 50 έως 53:** Χρησιμοποιούνται στην επικοινωνία μέσω του πρωτοκόλλου SPI [22].

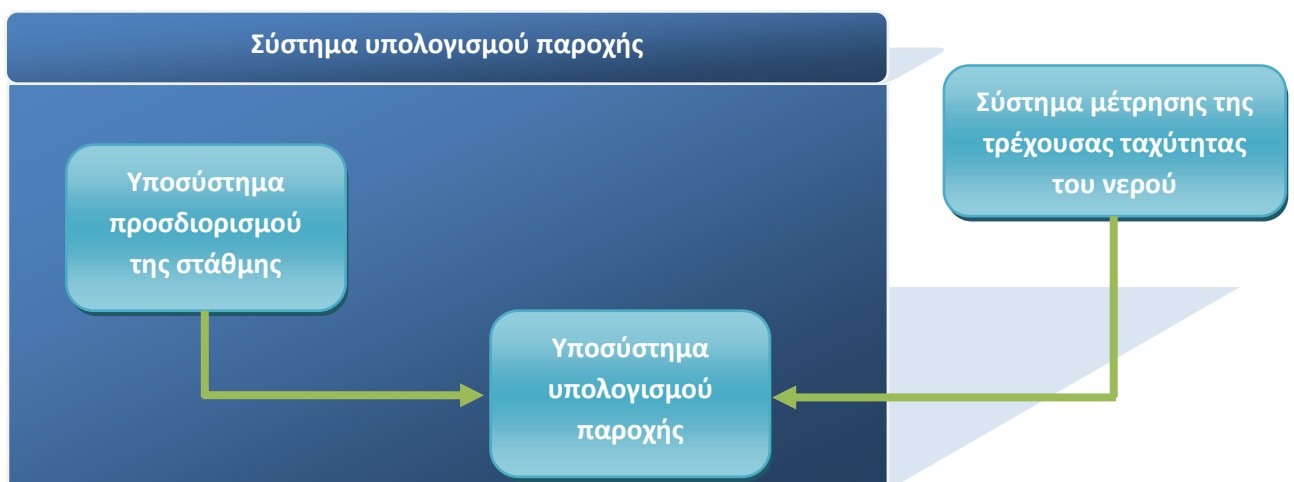


## 2. Ανάπτυξη του συστήματος υπολογισμού της στιγμιαίας παροχής

Ο σταθμός υπολογισμού της παροχής που είναι τοποθετημένος στο γεφύρι της Πολιτσάς (Σχήμα 10α) έχει ως στόχο τον υπολογισμό της στιγμιαίας παροχής του νερού που διέρχεται από μια διατομή της κοίτης του ποταμού. Στη συνέχεια τα αποτελέσματα των υπολογισμών ψηφιοποιούνται, αποθηκεύονται και περιοδικά μεταφέρονται στον τερματικό κόμβο διαδικτύου [26]. Στην παρούσα διπλωματική εργασία περιγράφεται η ανάπτυξη του συστήματος υπολογισμού – αποθήκευσης της παροχής (Q), που βρίσκεται στον σταθμό της γέφυρας της Πολιτσάς. Το γενικό διάγραμμα του συστήματος φαίνεται στο Σχήμα 10β. Η επιλογή του συγκεκριμένου σημείου έγινε διότι εκεί υπάρχει σταθερή διατομή της κοίτης του ποταμού. Ο σταθμός υπολογισμού της στιγμιαίας παροχής του νερού του ποταμού Άραχθου, έγινε στα πλαίσια του έργου WWQM\_AVR [27].



Σχήμα 10α. Φωτογραφία από το γεφύρι της Πολιτσάς

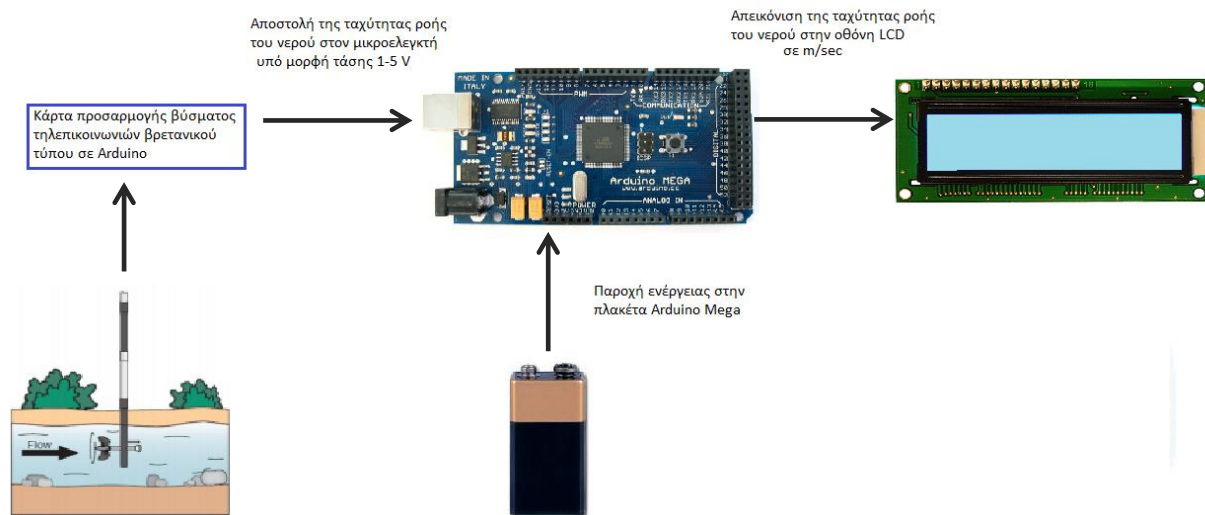


Σχήμα 10β. Γενικό διάγραμμα του συστήματος υπολογισμού της παροχής

## 2.1 Σύστημα μέτρησης της τρέχουσας ταχύτητας του νερού

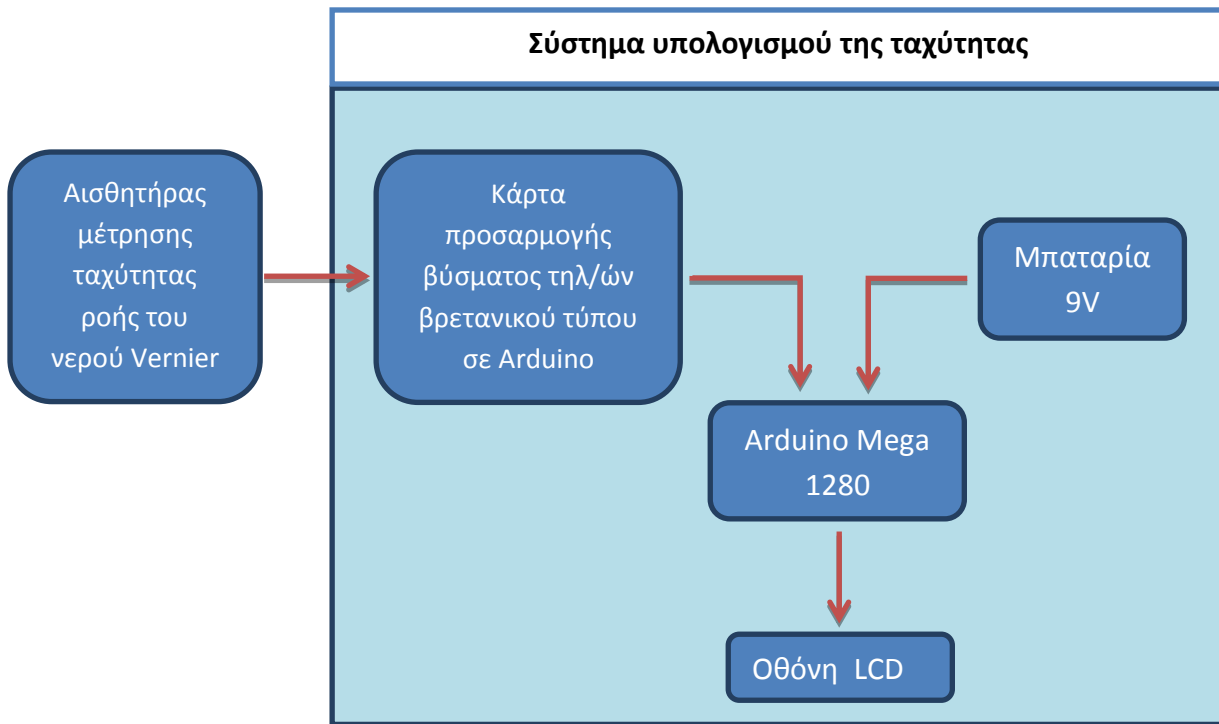
Αρχικά μετρήθηκε η ταχύτητα του νερού σε διάφορα υποτμήματα της διατομής της κοίτης του ποταμού, για συγκεκριμένο ύψος της στάθμης του νερού. Η συγκεκριμένη διαδικασία έγινε μία μόνο φορά ώστε να ελεγχθεί η ισχύς της σχέσης  $u=f(z)$  της παραγράφου 1.1 (σελ. 7) και ο προσδιορισμός των παραγόντων της σχέσης αυτής για κάθε μία από τις 30 στήλες νερού που χωρίστηκε η διατομή της κοίτης. Για την μέτρηση της τρέχουσας ταχύτητας του νερού αναπτύχθηκε ένα ηλεκτρονικό και ένα μηχανολογικό σύστημα.

Στο ηλεκτρονικό κομμάτι του συστήματος μέτρησης της τρέχουσας ταχύτητας του νερού, βασικό στοιχείο είναι το σύστημα μέτρησης της ταχύτητας του νερού της εταιρίας Vernier [28] (Σχήμα 11). Στο σύστημα μέτρησης της ταχύτητας του νερού χρησιμοποιήθηκε η πλατφόρμα Arduino Mega 1280, μια οθόνη LCD (στην οποία εμφανίζεται η μετρούμενη ταχύτητα), μία κάρτα προσαρμογής βύσματος τηλεπικοινωνιών βρετανικού τύπου σε Arduino και μια μπαταρία των 9V για την τροφοδοσία του συστήματος.



Σχήμα 11. Τα τμήματα του συστήματος μέτρησης της τρέχουσας ταχύτητας του νερού

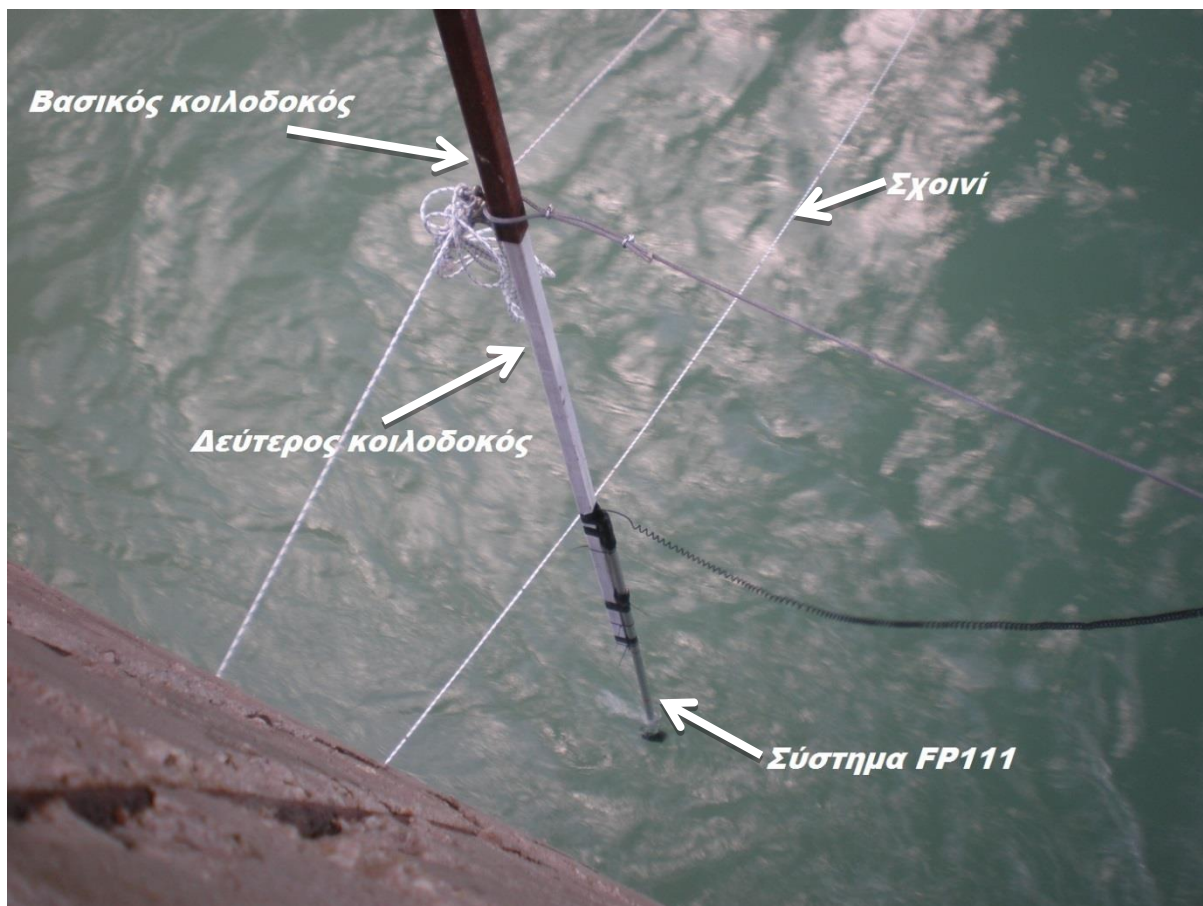
Το γενικό διάγραμμα του συστήματος μέτρησης της ταχύτητας του νερού παρουσιάζεται στο Σχήμα 12.



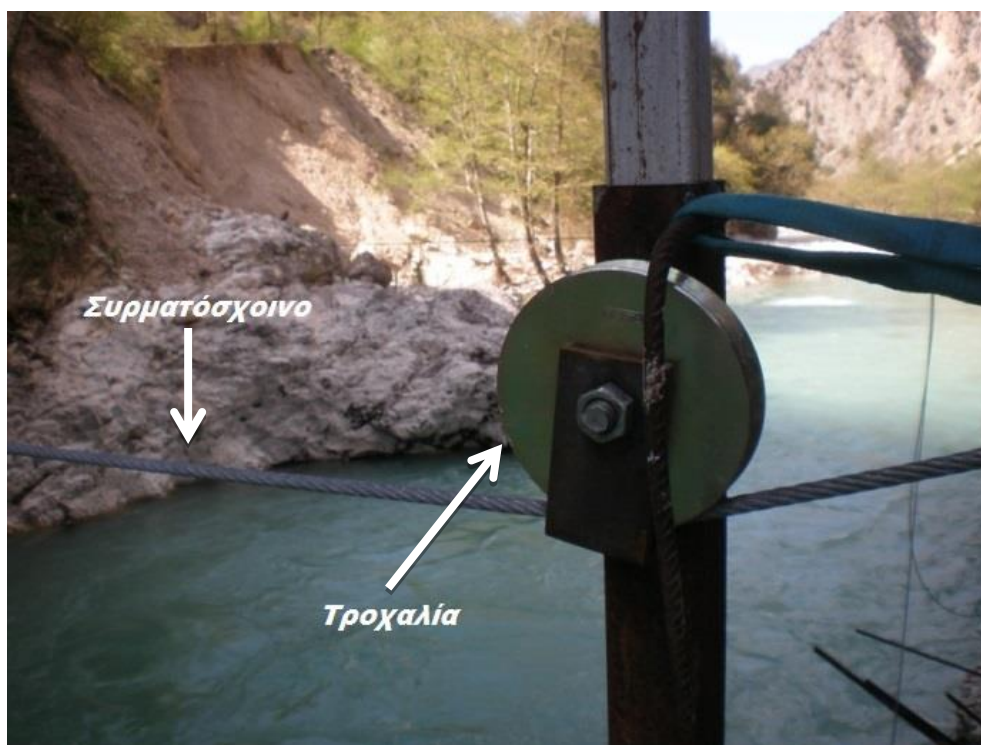
Σχήμα 12. Γενικό διάγραμμα του συστήματος μέτρησης της ταχύτητας του νερού

Αντί του συστήματος Vernier, μπορεί να χρησιμοποιηθεί το σύστημα FP111 της εταιρίας Global Water [30], το οποίο περιέχει ενσωματωμένο μικρολεγκτή, οθόνη και μπαταρία και προτιμήθηκε λόγω της ανθεκτικότητας της αλουμινένιας επεκτάσιμης λαβής του. Αναλυτικότερη περιγραφή των δύο παραπάνω συστημάτων ακολουθεί στην παράγραφο 2.3.3.

Όσον αφορά το μηχανολογικό μέρος, έγινε μια ειδική κατασκευή ανάρτησης του συστήματος FP111, η οποία ελέγχεται χειροκίνητα και μπορεί να κινείται κατά μήκος και βάθος της διατομής έτσι ώστε να μετράται η ταχύτητα του νερού στα διάφορα σημεία. Ειδικότερα τοποθετήθηκαν δύο παράλληλα συρματόσχοινα. Στο ένα εξ' αυτών τοποθετήθηκε κατακόρυφα ένας κοιλοδοκός (βασικός), ο οποίος με μία τροχαλία και τη βοήθεια ενός σχοινιού, μπορεί να κινείται κατα μήκος του συρματόσχοινου. Το άλλο συρματόσχοινο συνδέεται με τον κοιλοδοκό έτσι ώστε να μην επιτρέπει την εκτροπή του από το ρεύμα. Μέσα στον βασικό κοιλοδοκό τοποθετήθηκε ένας δεύτερος κοιλοδοκός ο οποίος έχει τη δυνατότητα να κινείται κατακόρυφα, έτσι ώστε να μπορεί να εισέρχεται σε διάφορα βάθη μέσα στον ποταμό. Το σύστημα FP111 αναρτήθηκε πάνω σε αυτόν τον κοιλοδοκό και έτσι δόθηκε η δυνατότητα να μετρηθεί η ταχύτητα του νερού στα διάφορα σημεία μέσα στον ποταμό. Η κατασκευή ανάρτησης του συστήματος FP111 παρουσιάζεται στα Σχήματα 13, 14 και 15.



Σχήμα 13. Κατασκευή ανάρτησης του συστήματος FP111



Σχήμα 14. Ο βασικός κοιλοδοκός με την ενσωματωμένη τροχαλία που κινείται κατα μήκος του συρματόσχοινου

Στην σύλληψη ιδέας για την διαδικασία της μέτρησης της ταχύτητας του νερού αλλά και στην υλοποίηση του μηχανολογικού μέρους του συστήματος, πρωταγωνιστικό ρόλο διαδραμάτισε ο Νικόλαος Μάνθος της Via Natura [31].



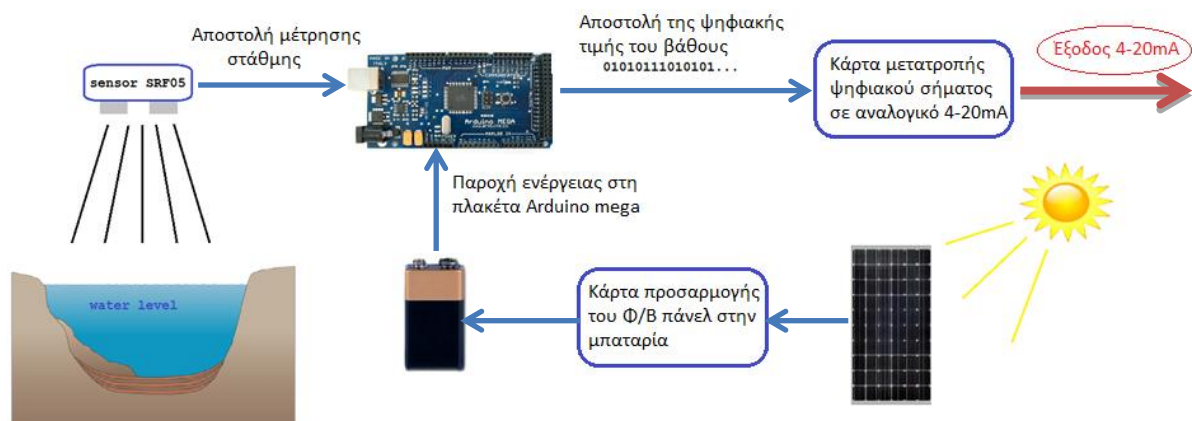
Σχήμα 15. Συνολική κατασκευή του συστήματος μέτρησης της ταχύτητας του νερού

## 2.2 Σύστημα υπολογισμού της στιγμιαίας παροχής

Το σύστημα που χρησιμοποιήθηκε για τον υπολογισμό της στιγμιαίας παροχής του νερού στον τοπικό σταθμό του γεφυριού της Πολιτσάς αποτελείται από δύο υποσυστήματα.

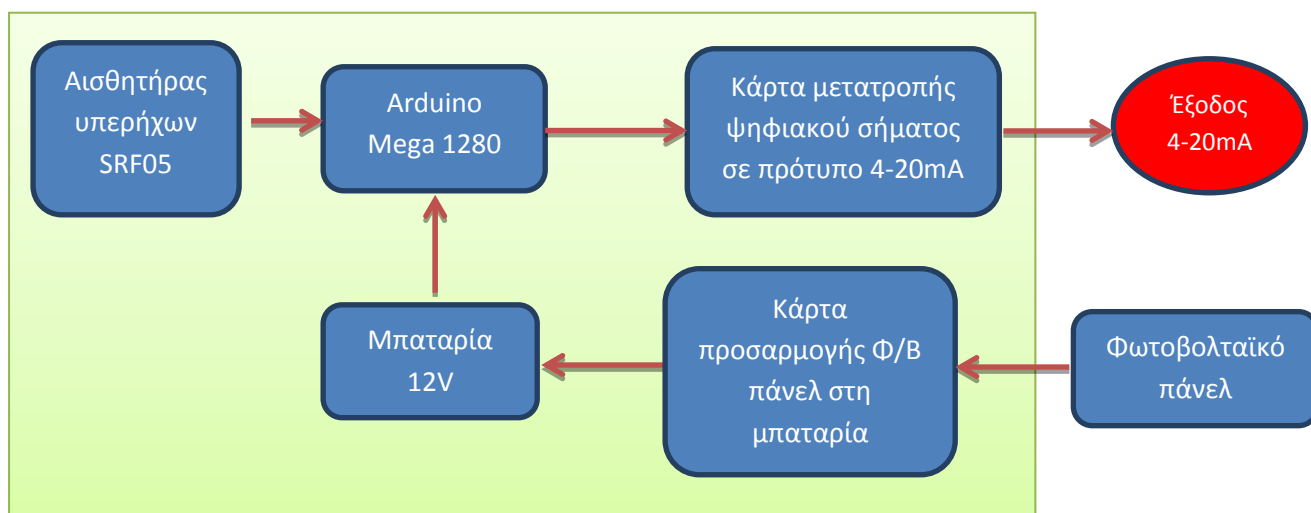
Το πρώτο υποσύστημα είναι εκείνο που προσδιορίζει τη θέση της στάθμης του νερού του ποταμού. Πιο συγκεκριμένα ο αισθητήρας υπερήχων προσδιορίζει την απόστασή του από την επιφάνεια του νερού. Κατόπιν ο μικροελεγκτής με την μέτρηση του αισθητήρα υπολογίζει τη θέση της στάθμης του νερού του ποταμού με μια αφαίρεση της απόστασης αισθητήρα-στάθμης, από την γνωστή απόσταση αισθητήρα-πυθμένα του ποταμού. Επειδή το υποσύστημα προσδιορισμού της στάθμης του νερού βρίσκεται κάτω από τη γέφυρα και δίπλα στο ποτάμι, και το υποσύστημα υπολογισμού της παροχής του νερού είναι εγκατεστημένο πάνω από τη γέφυρα στον κεντρικό σταθμό, σε απόσταση περίπου 10m, το ψηφιακό σήμα TTL [31] είναι αδύνατον να φτάσει ως εκεί αμετάβλητο. Έτσι το σήμα μετατρέπεται στο πρότυπο 4-20mA μέσω μιας κάρτας μετατροπής με βασικό στοιχείο έναν μετατροπέα ψηφιακού σήματος σε αναλογικό (DAC), η οποία παρουσιάζεται στην παράγραφο 2.4.3. Για την ενεργειακή αυτονομία του υποσυστήματος χρησιμοποιήθηκε μπαταρία των 12V η οποία φορτίζεται από ένα φωτοβολταϊκό πάνελ. Ανάμεσα στο πάνελ και τη μπαταρία, κατασκευάστηκε και

τοποθετήθηκε μια κάρτα για τη ρύθμιση της τάσης και την ομαλή φόρτιση της μπαταρίας, που περιγράφεται στην παράγραφο 2.4.4.



Σχήμα 16. Τα κύρια μέρη του υποσυστήματος προσδιορισμού της στάθμης του ποταμού

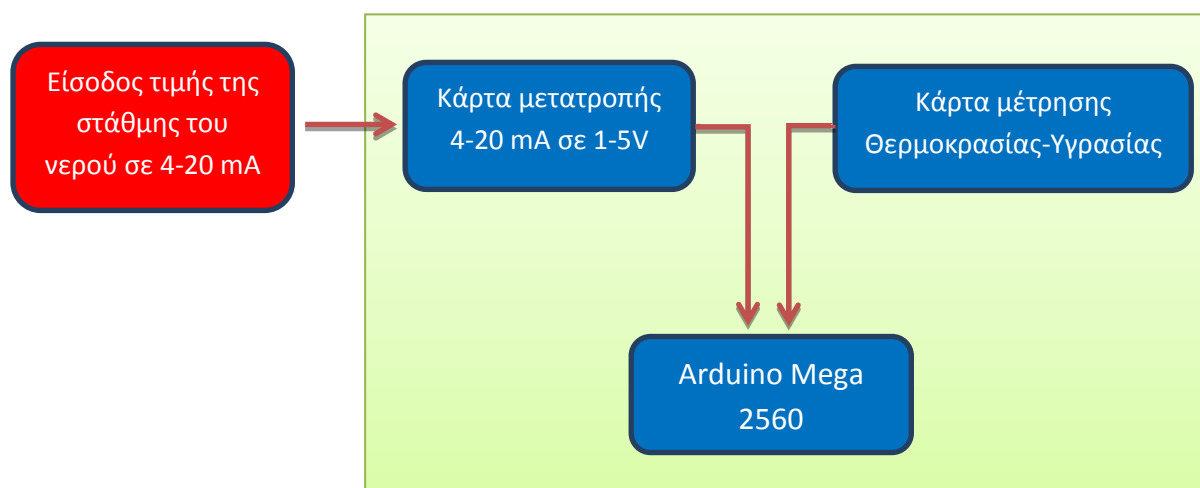
Το βασικό στοιχείο του ηλεκτρονικού υποσυστήματος είναι η πλατφόρμα Arduino Mega 1280 (Σχήμα 16). Το υποσύστημα αποτελείται επίσης από έναν αισθητήρα υπερήχων τον SRF05, ο οποίος μετρά την απόσταση ως την επιφάνεια του νερού. Μία κάρτα μετατροπής του ψηφιακού σήματος σε αναλογικό, μετατρέπει την τιμή της στάθμης σε αναλογικό σήμα στο πρότυπο 4-20mA. Το όλο σύστημα τροφοδοτείται από μία μπαταρία των 12V, η οποία μπορεί να φορτίζεται από ένα φωτοβολταϊκό πάνελ, μέσω μιας κάρτας προσαρμογής. Το σχηματικό διάγραμμα του υποσυστήματος παρουσιάζεται στο Σχήμα 17.



Σχήμα 17. Σχηματικό διάγραμμα του υποσυστήματος προσδιορισμού της στάθμης του ποταμού

Το δεύτερο υποσύστημα, είναι αυτό του υπολογισμού της στιγμιαίας παροχής του νερού του ποταμού και είναι εγκατεστημένο στο σταθμό που βρίσκεται πάνω στη γέφυρα. Όταν το ρολόι πραγματικού χρόνου (RTC) που συνδέεται στον μικροελεγκτή

δείχνει ώρα, 00:00, 12:00, 18:00 ή 06:00, τότε ο μικροελεγκτής λαμβάνει τη μέτρηση της στάθμης του ποταμού από το υποσύστημα προσδιορισμού της στάθμης, το οποίο όπως προαναφέρθηκε έχει αναλογική έξοδο 4-20mA. Η αναλογική αυτή τιμή μετατρέπεται σε αναλογική τιμή 1-5V, μέσω μίας κάρτας μετατροπής 4-20mA σε 1-5V που κατασκευάστηκε και ψηφιοποιείται από τον μετατροπέα αναλογικού σήματος σε ψηφιακό (analog to digital converter - ADC) του μικροελεγκτή. Αφού ψηφιοποιηθεί η μέτρηση, ο μικροελεγκτής κάνει τον υπολογισμό της στιγμιαίας παροχής και αποθηκεύει το αποτέλεσμα σε μια κάρτα μνήμης microSD. Μαζί με την παροχή και την στάθμη του ποταμού, στην κάρτα μνήμης αποθηκεύονται και η θερμοκρασία και η υγρασία του περιβάλλοντος, οι οποίες μετρώνται από έναν αισθητήρα θερμοκρασίας – υγρασίας που έχει προσαρμοστεί στο σύστημα μέσω μιας κάρτας. Οι τιμές των εν λόγω παραμέτρων αποστέλλονται, όταν και εάν ζητηθεί, στο ασύρματο σύστημα μετάδοσης. Το βασικό στοιχείο του ηλεκτρικού-ηλεκτρονικού υποσυστήματος υπολογισμού της στιγμιαίας παροχής του νερού είναι η πλατφόρμα Arduino 2560 [32]. Το σχηματικό διάγραμμα του υποσυστήματος παρουσιάζεται στο Σχήμα 18.



Σχήμα 18. Σχηματικό διάγραμμα του υποσυστήματος υπολογισμού της στιγμιαίας παροχής του νερού του ποταμού

Οι περιφερειακές κάρτες του υποσυστήματος υπολογισμού της στιγμιαίας παροχής του νερού, διασυνδέονται με τους ακροδέκτες του μικροελεγκτή μέσω βυσμάτων (headers) στην κεντρική κάρτα. Οι περιφερειακές ηλεκτρονικές κάρτες είναι:

- Κάρτα μετατροπής 4-20mA σε 1-5V
- Κάρτα μέτρησης θερμοκρασίας – υγρασίας.

Αρχικά, και για μία μόνο φορά, φορτώθηκε στον μικροελεγκτή του υποσυστήματος, ο bootloader με τη βοήθεια του προγράμματος Atmel studio 6.2 [33]. Οι απαραίτητες ρυθμίσεις του προγράμματος Atmel studio 6.2 για την φόρτωση του bootloader, παρουσιάζονται αναλυτικά στο Παράρτημα 2. Στη συνέχεια φορτώθηκε ο κώδικας του υποσυστήματος, με τον οποίο υπολογίζεται η στιγμιαία παροχή. Επίσης με τη βοήθεια της μονάδας δικτύου WIZ812MJ που χρησιμοποιήθηκε στην κάρτα μετατροπής Ethernet σε σειριακή επικοινωνία [32], αναπτύχθηκε ένας διακομιστής διαδικτύου (web server).

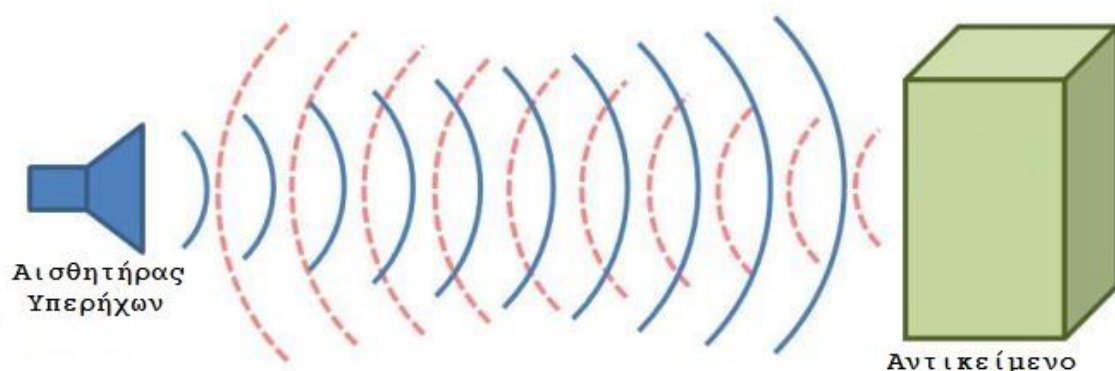
Όλα τα ηλεκτρονικά στοιχεία του συστήματος υπολογισμού της στιγμιαίας παροχής του νερού, τροφοδοτούνται από ένα σύστημα τροφοδοσίας που βασίζεται σε ηλιακές κυψέλες και ένα σύστημα αποθήκευσης ενέργειας [26]. Τέλος, το σύστημα μέτρησης συνδέεται σε ένα σύστημα πομποδέκτη με κατευθυντική κεραία Wi-Fi, για την ασύρματη επικοινωνία του σταθμού με τους αναμεταδότες.

## 2.3 Αισθητήρες του συστήματος

Στην παρούσα εργασία χρησιμοποιήθηκαν αισθητήρες για τη μέτρηση της θέσης της στάθμης του νερού, την μέτρηση της θερμοκρασίας και της υγρασίας του περιβάλλοντος, καθώς και συστήματα που περιέχουν αισθητήρες για τη μέτρηση της ταχύτητας του νερού. Σε αυτή τη παράγραφο γίνεται αναφορά στους αισθητήρες και τα συστήματα που χρησιμοποιήθηκαν, παρουσιάζονται τα χαρακτηριστικά τους και γίνεται ανάλυση του τρόπου λειτουργίας για το κάθε ένα από αυτά.

### 2.3.1 Αισθητήρες μέτρησης της απόστασης

Όλοι οι αισθητήρες μέτρησης απόστασης λειτουργούν κατά τον ίδιο τρόπο. Ο ανιχνευτής απόστασης στέλνει ένα σήμα υπερήχων και μετρά το χρόνο που παίρνει ώστε ο παλμός να ανακλασθεί από το αντικείμενο και να επιστρέψει πίσω (Σχήμα 19). Όσο μεγαλύτερος είναι αυτός ο χρόνος, τόσο μεγαλύτερη είναι και η απόσταση από το αντικείμενο - στόχο.



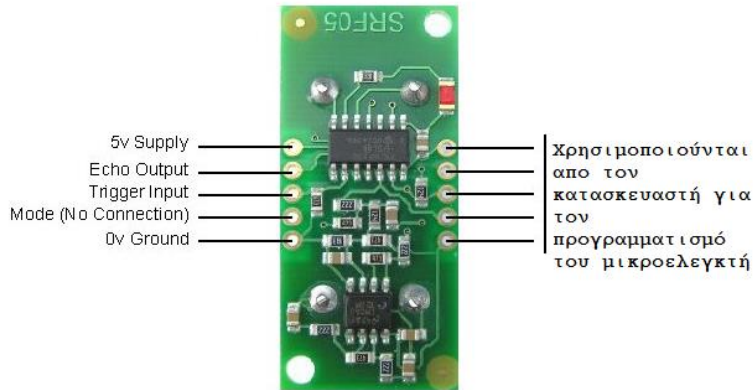
Σχήμα 19. Βασική λειτουργία αισθητήρων υπερήχων

Στην παρούσα εργασία χρησιμοποιήθηκε ο αισθητήρας υπερήχων SRF05 [34] για την μέτρηση της απόστασης του υποσυστήματος προσδιορισμού της στάθμης από την επιφάνεια του νερού. Η εμπρόσθια όψη του αισθητήρα, με τον πομπό υπερήχων στα αριστερά και τον δέκτη υπερήχων στα δεξιά, φαίνεται στο Σχήμα 20. Ο αισθητήρας SRF05 έχει μέγιστη απόσταση ανίχνευσης 4.5m και λειτουργεί με δύο τρόπους. Όταν ο ακροδέκτης Mode, που φαίνεται στο Σχήμα 21 παραμένει ασύνδετος, ο αισθητήρας SRF05 λειτουργεί με ξεχωριστούς ακροδέκτες trigger και echo (λειτουργία 1). Αντίθετα όταν ο ακροδέκτης Mode, είναι συνδεδεμένος στη γείωση, τότε επιτρέπει στον αισθητήρα να χρησιμοποιήσει έναν ενιαίο ακροδέκτη για trigger και echo (λειτουργία 2). Ο πρώτος τρόπος λειτουργίας είναι ο απλούστερος.





Σχήμα 20. Εμπρόσθια όψη του αισθητήρα υπερήχων SRF05 με τον πομπό και τον δέκτη



Σχήμα 21. Οι ακροδέκτες της κάρτας του αισθητήρα υπερήχων SRF05

Για τη λειτουργία του αισθητήρα χρειάζεται να σταλεί ένας σύντομος παλμός, ελάχιστης χρονικής διάρκειας 10μS, στην είσοδο του ακροδέκτη Trigger. Στη συνέχεια ο πομπός του αισθητήρα SRF05 θα στείλει μια ομάδα 8 παλμών συχνότητας 40kHz ή  $2.5 \cdot 10^{-5}$  sec, και θέτει την κατάσταση του ακροδέκτη Echo από low σε high. Μόλις η ομάδα των 8 παλμών ανακλαστεί από το αντικείμενο και επιστρέψει στον αισθητήρα, ανιχνεύεται από τον δέκτη του αισθητήρα και ο ακροδέκτης Echo τίθεται πάλι σε κατάσταση low. Στον ακροδέκτη Echo λοιπόν δημιουργείται ένας παλμός του οποίου το πλάτος είναι ανάλογο με την απόσταση του αισθητήρα από το αντικείμενο. Μετρώντας το χρόνο που ο παλμός είναι high μπορούμε να υπολογίσουμε την απόσταση αισθητήρα – αντικειμένου από την εξής σχέση:

$$D = \frac{T \cdot 340 \frac{m}{sec}}{2}$$

Όπου,

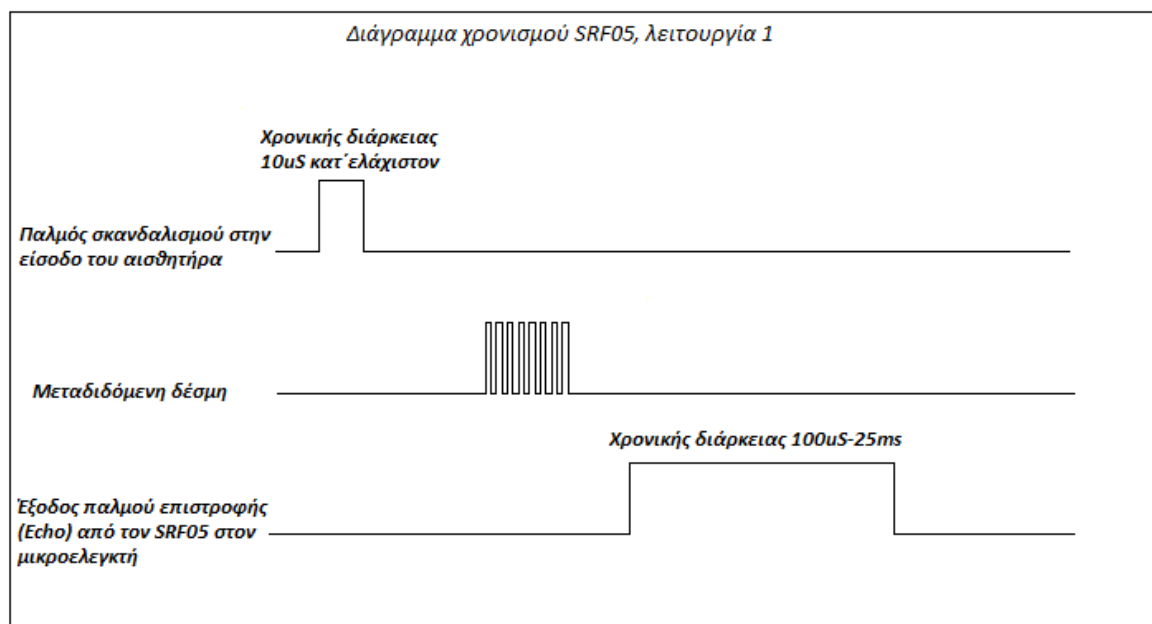
D: Απόσταση αισθητήρα – αντικειμένου σε m

T: Χρόνος κατά τον οποίο ο ακροδέκτης Echo βρίσκεται σε κατάσταση high σε sec

Η ταχύτητα των υπερήχων στον αέρα σε κανονικές συνθήκες είναι 340m/sec και η διαίρεση με 2 γίνεται διότι ο παλμός διανύει δύο φορές την απόσταση. Πρέπει να

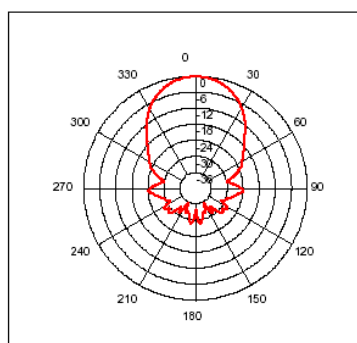
σημειωθεί πως εάν ο ακροδέκτης Echo δεχθεί έναν παλμό, χρονικής διάρκειας μεγαλύτερης των 25ms τότε τίθεται αυτομάτως σε κατάσταση low.

Το διάγραμμα χρονισμού της λειτουργίας του αισθητήρα στη λειτουργία 1 φαίνεται στο Σχήμα 22.



Σχήμα 22. Διάγραμμα χρονισμού του αισθητήρα SRF05 στη λειτουργία 1

Εάν ο ακροδέκτης mode της κάρτας του αισθητήρα παραμείνει ασύνδετος, η κάρτα λειτουργεί στη λειτουργία 1 και έχει διασυνδεδεμένη εκ κατασκευής εσωτερική αντίσταση pull up. Επίσης ο SRF05 μπορεί να ενεργοποιηθεί για μέτρηση απόστασης το πολύ 20 φορές κάθε δευτερόλεπτο. Αυτό γίνεται για να εξασφαλιστεί ότι το προηγούμενο υπερηχητικό σήμα που έχει αποσταλεί, έχει εξασθενήσει και δεν θα προκαλέσει παρεμβολές στην επόμενη μέτρηση. Το διάγραμμα της έντασης της εξερχόμενης δέσμης υπερήχων του αισθητήρα SRF05 στον χώρο, είναι κωνικό και φαίνεται στο Σχήμα 23. Το πλάτος της δέσμης είναι συνάρτηση του εμβαδού επιφανείας του αισθητήρα και είναι σταθερό.



Σχήμα 23. Διάγραμμα έντασης της εξερχόμενης δέσμης του αισθητήρα SRF05 στο χώρο



Σχήμα 24. Αισθητήρας υπερήχων WL705-012 της Global Water

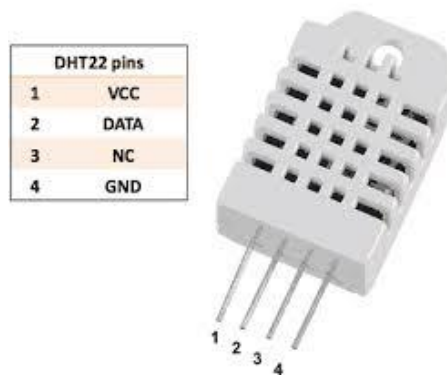
Ένας άλλος αισθητήρας υπερήχων ο οποίος μπορεί να χρησιμοποιηθεί για τον προσδιορισμό της θέσης της στάθμης του νερού είναι ο WL705-012 [35], που φαίνεται στο Σχήμα 24. Ο αισθητήρας αυτός είναι σφραγισμένος μέσα σε ανοξείδωτο περίβλημα από χάλυβα και παρέχει μία έξοδο 4-20 mA. Η έξοδος αποτελείται από τρία καλώδια μήκους 1.8m. Τα δύο είναι η τροφοδοσία του αισθητήρα (καφέ: τάση 10V-30V) και γείωση (μπλέ), και το τρίτο (μαύρο) για τη μεταφορά του σήματος 4-20mA. Τα χαρακτηριστικά του αισθητήρα παρουσιάζονται στον Πίνακα 3.

Έυρος	0.10 έως 3.6 m
Ανάλυση	0.001 m
Ακρίβεια	0.5% του εύρους
Τροφοδοσία	10 έως 30 VDC
Τύπος αισθητηρίου	Ενισχυμένος πιεζοηλεκτρικός κρύσταλλος
Γωνία ακτίνων	12 ° κωνικό σχήμα
Ενημέρωση Τιμής	Ανά 50 ms
Εύρος θερμοκρασίας λειτουργίας	-40 ° C έως 70 ° C

Πίνακας 3. Χαρακτηριστικά του αισθητήρα υπερήχων WL705-012 της Global Water

### 2.3.2 Αισθητήρας θερμοκρασίας - υγρασίας

Ο αισθητήρας θερμοκρασίας - υγρασίας DHT22 [36], διαθέτει τεχνολογία ανίχνευσης της θερμοκρασίας και της υγρασίας η οποία βασίζεται σε έναν πολυμερή πυκνωτή. Χρησιμοποιεί ψηφιακό σήμα για την αποστολή των δεδομένων. Έχει μικρό μέγεθος, χαμηλή κατανάλωση και μεγάλη απόσταση μετάδοσης των δεδομένων (άνω των 20m). Στο Σχήμα 25 φαίνονται οι ακροδέκτες του αισθητήρα καθώς και η χρήση τους.



Σχήμα 25. Οι ακροδέκτες του αισθητήρα DHT22 και η χρήση τους

Τα τεχνικά χαρακτηριστικά του αισθητήρα παρουσιάζονται στον Πίνακα 4.

Αντικείμενο	Κλίμακα μέτρησης	Ακρίβεια μέτρησης υγρασίας	Ακρίβεια μέτρησης θερμοκρασίας	Τροφοδοσία	Αισθητήριο στοιχείο
Αισθητήρας θερμοκρασίας-υγρασίας DHT22	0-100% RH -40 ~ 80 °C	± 2% RH	± 0.5 °C	3.3V έως 6V DC	Πολυμερής πυκνωτής

Πίνακας 4. Τεχνικά χαρακτηριστικά του αισθητήρα θερμοκρασίας - υγρασίας DHT22

Η επικοινωνία μεταξύ υπολογιστή και αισθητήρα γίνεται μέσω διαύλου μιας γραμμής (one - wire bus). Όταν ο μικροελεγκτής στείλει σήμα εκκίνησης, αλλάζει την κατάσταση του αισθητήρα από κατάσταση αναμονής, σε κατάσταση λειτουργίας. Μόλις ο μικροελεγκτής τερματίσει την αποστολή του σήματος εκκίνησης, τότε ο αισθητήρας στέλνει σήμα απόκρισης που περιέχει τα δεδομένα. Τα δεδομένα που λαμβάνονται από τον αισθητήρα έχουν μέγεθος 40 bits. Τα πρώτα 16bits είναι για την υγρασία, τα επόμενα 16 για τη θερμοκρασία και τα 8 τελευταία για τον υπολογισμό του αθροίσματος για έλεγχο.

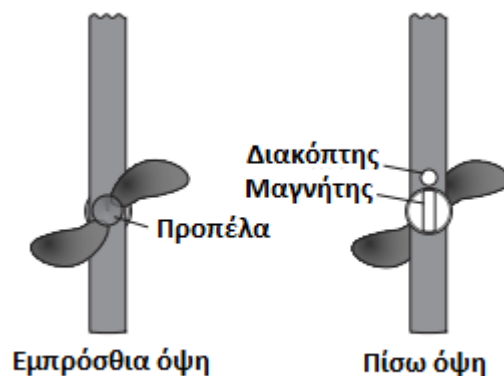
### 2.3.3 Συστήματα μέτρησης της ταχύτητας του νερού

Για τη μέτρηση της ταχύτητας του νερού του ποταμού χρησιμοποιήθηκαν δύο συστήματα. Το σύστημα της εταιρίας Vernier [28], και το σύστημα FP111 [37] της εταιρίας Global Water. Όταν το σύστημα μέτρησης ταχύτητας της εταιρίας Vernier τοποθετηθεί σε ένα ρεύμα, το νερό που ρέει εξαναγκάζει την προπέλα του να περιστραφεί. Όσο πιο γρήγορα κινείται το νερό, τόσο πιο γρήγορα γυρίζει και η προπέλα. Ένας μαγνήτης ο οποίος βρίσκεται τοποθετημένος στο πίσω μέρος της



Σχήμα 26. Αισθητήρας μέτρησης ταχύτητας ροής Vernier

προπέλας περιστρέφεται με αυτήν. Με κάθε μισό περιστροφής ο μαγνήτης ενεργοποιεί έναν διακόπτη μέσα στο καλάμι (Σχήμα 27). Ο διακόπτης στέλνει παλμούς στο κουτί επεξεργασίας σήματος (Σχήμα 26) όπου οι παλμοί μετατρέπονται σε τάση ανάλογη με το ρυθμό ροής. Στη συνέχεια αυτή η αναλογική τάση ψηφιοποιείται από τον μετατροπέα αναλογικού σήματος σε ψηφιακό (analog to digital converter - ADC) του μικροελεγκτή ATmega128 που υπολογίζει την ταχύτητα, η οποία και αποστέλεται στην οθόνη LCD για ανάγνωση. Ο ρυθμός της ταχύτητας ροής μετριέται σε m/sec.



Σχήμα 27. Αισθητήρας μέτρησης ταχύτητας Vernier

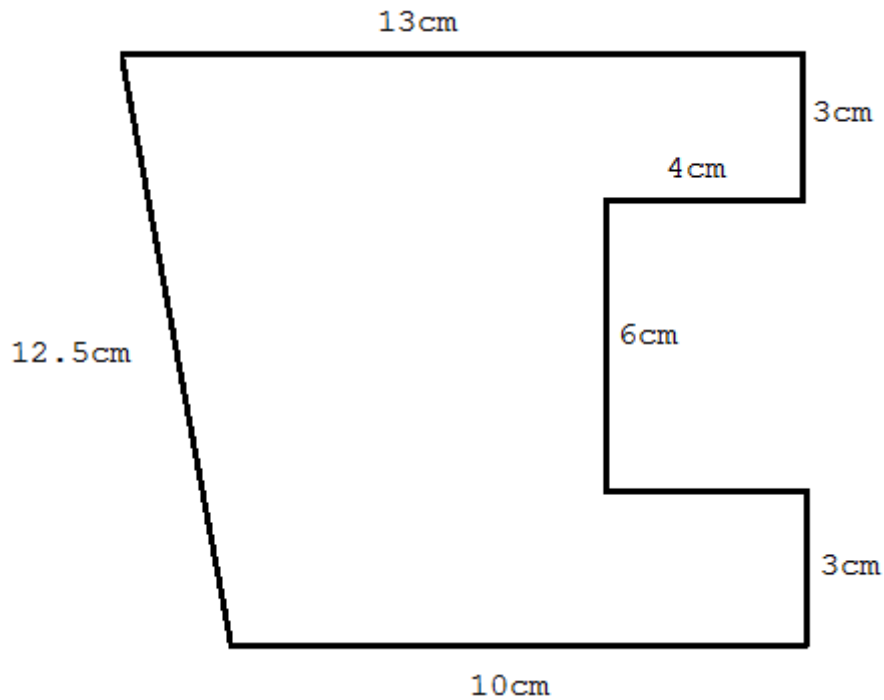
Στον Πίνακα 5 παρουσιάζονται τα χαρακτηριστικά του αισθητήρα.

Ανάλυση	0.005 m/sec
Χρόνος απόκρισης	98% ανάγνωση της πλήρους κλίμακας σε 5sec, 100% ανάγνωση της πλήρους κλίμακας σε 15sec
Εύρος θερμοκρασίας που μπορεί να τοποθετηθεί	0 έως 70 C
Ακρίβεια	1% ανάγνωση της πλήρους κλίμακας
Βαθμονόμηση	Κλίση 1[m/s]/V

Πίνακας 5. Χαρακτηριστικά αισθητήρα μέτρησης ταχύτητας νερού της Vernier

Λόγω της ορμητικότητας των νερών του ποταμού, προκαλείται μια εκτροπή του συστήματος μέτρησης της ταχύτητας του νερού από την αρχική διεύθυνση η οποία είναι παράλληλη με τη διεύθυνση ροής του νερού. Αυτό έχει σαν αποτέλεσμα η προπέλα του συστήματος να βρίσκεται σε κάποια δεδομένη στιγμή υπό γωνία με τη ροή του νερού (γωνία εκτροπής) και η μέτρηση της ταχύτητας να είναι ανακριβής. Για το λόγο αυτό κατασκευάστηκε το πηδάλιο ευθυγράμμισης [38]. Σκοπός της κατασκευής του πηδαλίου είναι να αποτελέσει το διορθωτικό μέσον, ώστε να διατηρηθεί ο αισθητήρας

σε θέση παράλληλη με τη στιγμιαία ροή του νερού. Το σχήμα του πηδαλίου πρέπει να μοιάζει με αυτό μιας αεροτομής, ώστε να μειώνεται στο ελάχιστο η απώλεια λόγω τριβών από το νερό. Έχοντας υδροδυναμικό σχήμα, η ροή «χαϊδεύει» ομαλά την επιφάνειά του και δεν δημιουργούνται πολλές δίνες, οι οποίες αυξάνουν την αντίσταση. Το πηδάλιο που θα χρησιμοποιηθεί πρέπει να μπορεί να λειτουργήσει αποτελεσματικά σε μεγάλες γωνίες, έτσι έγινε η επιλογή του σχεδίου που φαίνεται στο Σχήμα 28 ενώ στο Σχήμα 29 παρουσιάζεται το πηδάλιο.



Σχήμα 28. Σχέδιο πηδαλίου ευθυγράμμισης για τον αισθητήρα μέτρησης ταχύτητας νερού Vernier



Σχήμα 29. Κατασκευή του πηδαλίου ευθυγράμμισης του αισθητήρα μέτρησης ταχύτητας νερού Vernier

Η κατασκευή προσαρτήθηκε στο ρόμετρο Vernier με δύο μεταλλικούς σφιγκτήρες όπως φαίνεται στο Σχήμα 30.



Σχήμα 30. Το ρόμετρο Vernier με προσαρμοσμένο το πηδάλιο ευθυγράμμισης

Το σύστημα μέτρησης της ταχύτητας του νερού FP111 της εταιρίας Global Water [30], που φαίνεται στο Σχήμα 31, αποτελείται από μία προπέλα πάνω στην οποία υπάρχει ένα μαγνητικό υλικό το οποίο δημιουργεί ηλεκτρικό σήμα κάθε φορά που περνά από ένα σταθερό σημείο της λαβής. Η προπέλα μπορεί και περιστρέφεται ελεύθερα στον άξονα ενός ρουλεμάν χωρίς μηχανικές διασυνδέσεις για την ελάχιστη τριβή και προστατεύεται μέσα σε μία πλαστική κυλινδρική θήκη. Ο υπολογιστής της συσκευής (Σχήμα 32) λαμβάνει το ηλεκτρικό σήμα που δημιουργείται από την προπέλα, το οποίο είναι ανάλογο της ταχύτητας του νερού, και αφού το ενισχύσει υπολογίζει την ταχύτητα του νερού σε πόδια ή μέτρα ανά δευτερόλεπτο. Η οθόνη LCD εμφανίζει την υπολογισθείσα ταχύτητα. Ο υπολογιστής ταχύτητας του νερού καλύπτεται από ένα αδιάβροχο περίβλημα, έχει ενσωματωμένα τέσσερα κουμπιά για την αλλαγή των λειτουργιών και την επαναφορά της οθόνης και τροφοδοτείται από μια μη αντικαταστάσιμη μπαταρία που διαρκεί περίπου πέντε χρόνια με κανονική χρήση. Ο υπολογιστής ταχύτητας και η προπέλα διασυνδέονται με μια επεκτάσιμη λαβή αλουμινίου, μήκους 1.7m έως 4.6m.



Σχήμα 31. Αισθητήρας μέτρησης ταχύτητας ροής του νερού FP111 της εταιρίας Global Water



Σχήμα 32. Ο υπολογιστής του Ροομέτρου FP111

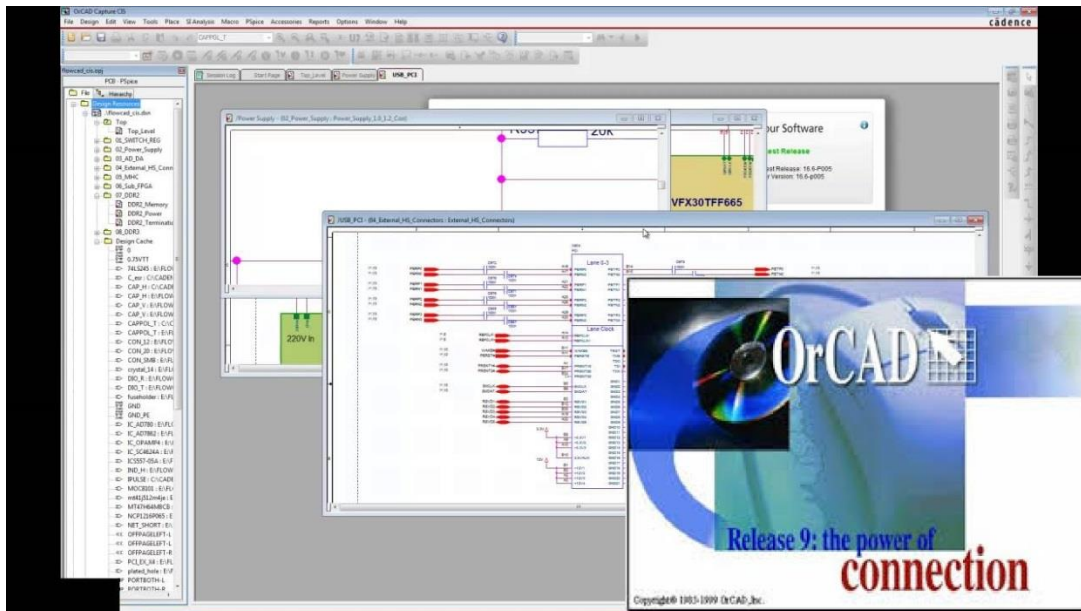
## 2.4 Σχεδιασμός του συστήματος

Σε αυτή τη παράγραφο γίνεται η παρουσίαση όλων των καρτών που σχεδιάστηκαν για κάθε υποσύστημα. Οι κάρτες του συστήματος σχεδιάστηκαν με το σχεδιαστικό πακέτο OrCAD [39].

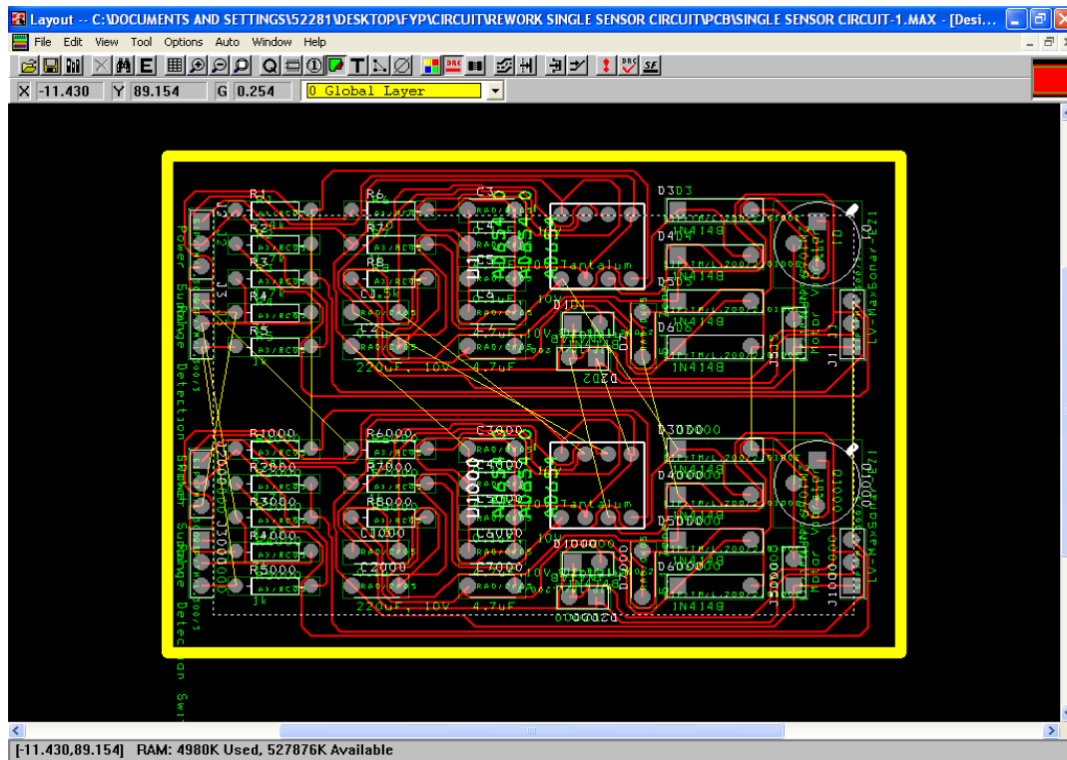


## 2.4.1 Σχεδιαστικό πακέτο OrCAD

Η έκδοση του OrCAD που χρησιμοποιήθηκε είναι η OrCAD 9.1. Για τον σχεδιασμό των σχηματικών κυκλωμάτων χρησιμοποιήθηκε το OrCAD Capture CIS (Σχήμα 33) και για τον σχεδιασμό των τυπωμένων κυκλωμάτων το OrCAD Layout (Σχήμα 34).



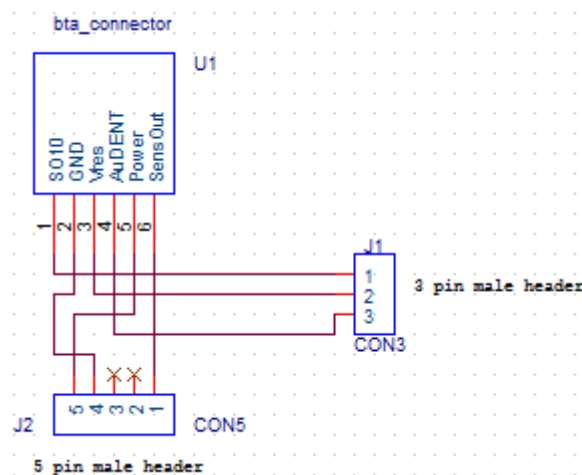
Σχήμα 33. Το περιβάλλον εργασίας του Orcad, CAPTURE CIS



Σχήμα 34. Το περιβάλλον εργασίας του Orcad LAYOUT

## 2.4.2 Κάρτα προσαρμογής βύσματος τηλεπικοινωνιών βρετανικού τύπου σε Arduino

Ο αισθητήρας μέτρησης της ταχύτητας νερού Vernier, διαθέτει για έξοδο ένα καλώδιο με απόληξη έναν αρσενικό ακροδέκτη τηλεπικοινωνιών βρετανικού τύπου. Για να προσαρμοστεί ο αισθητήρας στην πλατφόρμα Arduino, σχεδιάστηκε η συγκεκριμένη κάρτα η οποία διαθέτει έναν θηλυκό ακροδέκτη τηλεπικοινωνιών βρετανικού τύπου, ο οποίος λαμβάνει τα σήματα από τον αισθητήρα και στη συνέχεια μέσω των ακροδεκτών της κάρτας οδηγούνται στις ανάλογες εισόδους του Arduino. Ο τρόπος διασύνδεσης της κάρτας στην πλακέτα Arduino Mega 1280 καθώς και τα απαραίτητα υλικά για την κατασκευή της κάρτας, παρουσιάζονται αναλυτικά στο Παράρτημα 4. Το σχηματικό διάγραμμα του κυκλώματος της κάρτας φαίνεται στο Σχήμα 35.



Σχήμα 35. Σχηματικό διάγραμμα κυκλώματος κάρτας προσαρμογής βύσματος τηλεπικοινωνιών βρετανικού τύπου σε Arduino

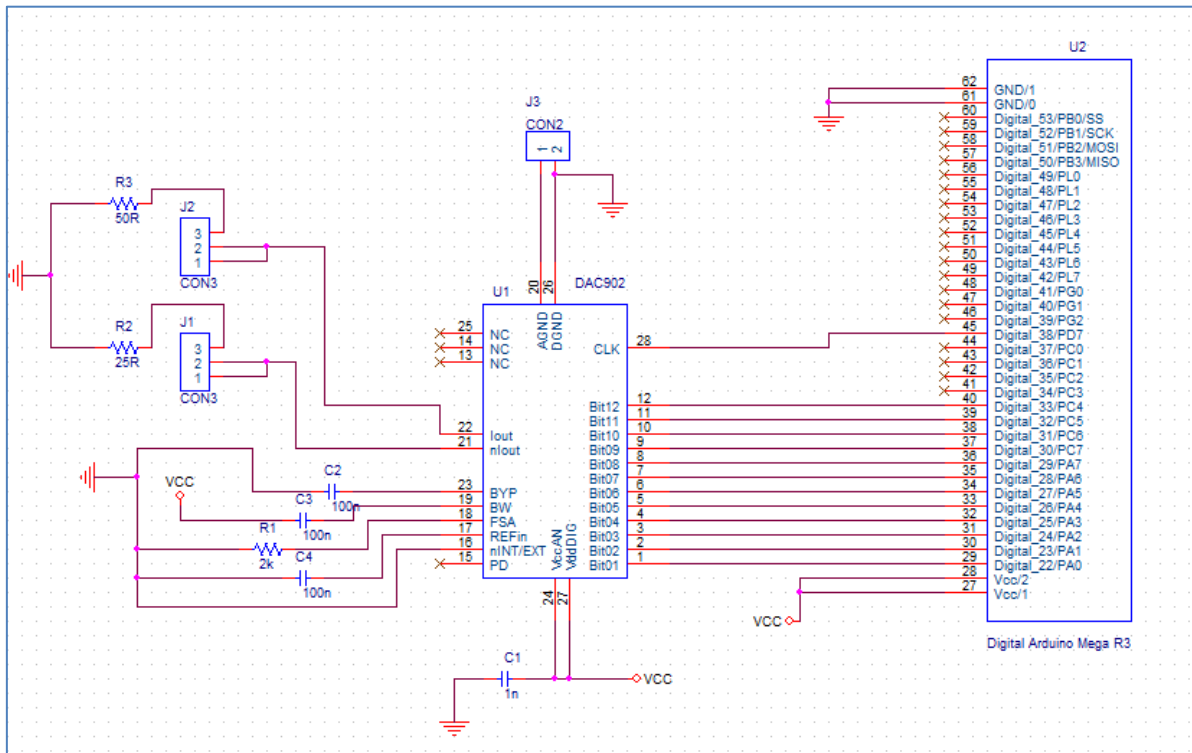
Χρησιμοποιήθηκε ένα θηλυκό βύσμα τηλεπικοινωνιών βρετανικού τύπου U1 [29] και δύο αρσενικά βύσματα, J1 τριών ακροδεκτών και J2 πέντε ακροδεκτών. Τα δεδομένα που λαμβάνονται από τον αισθητήρα μέσω του θηλυκού βύσματος, περνάνε στους αρσενικούς ακροδέκτες και από εκεί στον μικροελεγκτή.

## 2.4.3 Κάρτα μετατροπής ψηφιακού σήματος σε πρότυπο 4-20mA

Η συγκεκριμένη κάρτα αναπτύχθηκε για να μετατραπεί η ψηφιακή τιμή της στάθμης του νερού του ποταμού, η οποία μετράται από το πρώτο υποσύστημα, σε πρότυπο 4-20mA. Έτσι, υπό αυτή τη μορφή, η τιμή της στάθμης θα μπορεί να μεταφερθεί χωρίς απώλειες στο δεύτερο υποσύστημα. Το βασικό στοιχείο αυτής της κάρτας είναι ένας μετατροπέας ψηφιακού σήματος σε αναλογικό (DAC) [40].

Ειδικότερα στην κάρτα που αναπτύχθηκε, χρησιμοποιήθηκε το ολοκληρωμένο κύκλωμα DAC902 [41], της εταιρίας Texas Instruments. Συνδέοντας στην έξοδο  $i_{out}$  του

μετατροπέα ψηφιακού σήματος σε αναλογικό, μια αντίσταση φορτίου 50Ω R3, στην έξοδο αναπτύσσεται τάση 0-1V. Στη διαφορική έξοδο συνδέθηκε μία αντίσταση 25Ω R2. Στους ακροδέκτες τροφοδοσίας του DAC συνδέθηκε ένας πυκνωτής 1nF C1 για την εξάλειψη του θορύβου. Οι ακροδέκτες εισόδου δεδομένων Bit01 έως Bit12 του DAC, συνδέονται μέσω ενός ακροδέκτη U2 στις ψηφιακές εξόδους 29 έως 40 του Arduino Mega 1280. Το σχηματικό διάγραμμα του κυκλώματος της ηλεκτρονικής κάρτας παρουσιάζεται στο Σχήμα 36.



Σχήμα 36. Σχηματικό διάγραμμα κυκλώματος κάρτας μετατροπής ψηφιακού σήματος σε πρότυπο 4-20mA

Στον Πίνακα 6 παρουσιάζονται αναλυτικά οι λειτουργίες για κάθε ακροδέκτη του DAC 902 καθώς και οι συνδέσεις τους.

Ακροδέκτης DAC902	Ένδειξη	Περιγραφή
1 έως 12	Bit 1 έως Bit 12	Δεδομένο Bit 1 (D11),MSB Εώς Δεδομένο Bit 12 (D0), LSB Συνδέονται στους ψηφιακούς ακροδέκτης 29 έως 40 του Arduino Mega
13	NC	Χωρίς σύνδεση
14	NC	Χωρίς σύνδεση
15	PD	Είσοδος ελέγχου ισχύος Χωρίς σύνδεση
16	INT/EXT	Ακροδέκτης επολογής αναφοράς Συνδέεται στην αναλογική γείωση
17	REFin	Αναφορά εισόδου/εξόδου Συνδέεται στην αναλογική γείωση
18	FSA	Προσαρμογή εξόδου πλήρους κλίμακας Συνδέεται στην αναλογική γείωση
19	BW	Μείωση εύρους ζώνης θορύβου Συνδέεται στην τροφοδοσία
20	AGND	Αναλογική γείωση
21	-Iout	Συμπληρωματικό ρεύμα εξόδου του DAC
22	Iout	Ρεύμα εξόδου του DAC
23	BYP	Παράκαμψη κόμβου Συνδέεται στην αναλογική γείωση
24	+Va	Αναλογική τάση τροφοδοσίας, 2.7 V-5.5 V
25	NC	Χωρίς σύνδεση
26	DGND	Ψηφιακή γείωση
27	+Vd	Ψηφιακή τάση τροφοδοσίας, 2.7 V-5.5 V
28	CLK	Είσοδος ρολογιού Συνδέεται στον ψηφιακό ακροδέκτη 45 του Arduino Mega

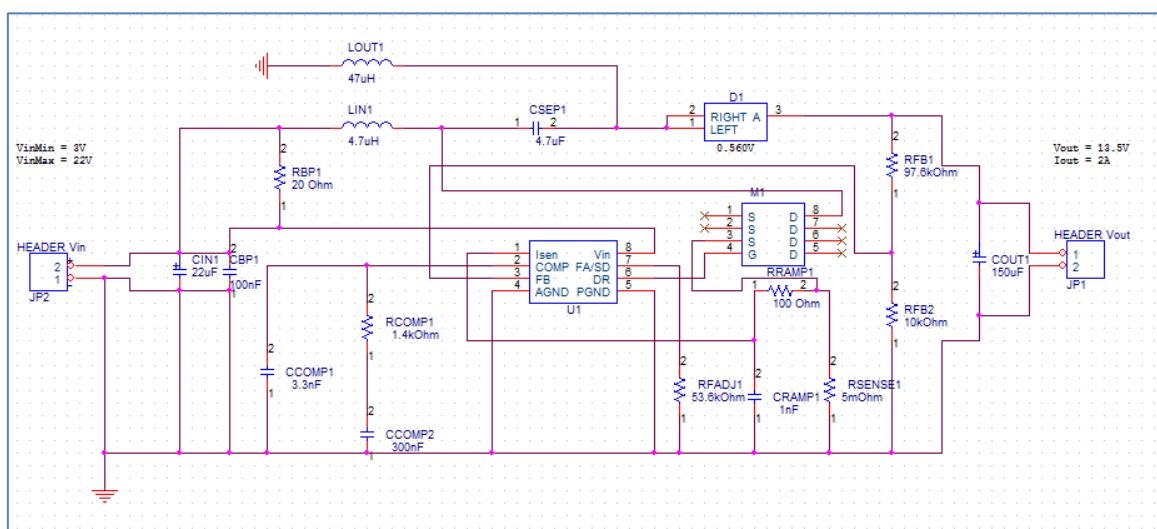
Πίνακας 6. Λειτουργίες και συνδέσεις των ακροδεκτών του DAC 902

## 2.4.4 Κάρτα προσαρμογής του φωτοβολταϊκού πάνελ στη μπαταρία

Η συγκεκριμένη κάρτα έχει ως σκοπό να προσαρμόζει την έξοδο του φωτοβολταϊκού πάνελ στη μπαταρία 12V, η οποία τροφοδοτεί το σύστημα προσδιορισμού της στάθμης του ποταμού. Λαμβάνει την εξερχόμενη τάση από το πάνελ, η οποία κυμαίνεται από 3V έως 22V και βγάζει τάση 13.5V και ρεύμα 2A. Το κύριο στοιχείο της κάρτας είναι ένας ρυθμιστής τάσης (regulator).

Στην κάρτα που αναπτύχθηκε χρησιμοποιήθηκε το ολοκληρωμένο LM3478MM1 [42] (switching regulator), της εταιρίας Texas Instruments, έτσι ώστε να διατηρείται σταθερή τάση εξόδου στα 13.5V η οποία είναι και η επιθυμητή τάση για να φορτιστεί η μπαταρία. Όταν η τάση είναι μικρότερη από 13.5V, στην περιοχή 1-13V, τότε μπορεί να φορτιστεί η μπαταρία με την υπάρχουσα τάση και ρεύμα, έτσι ώστε να μην υπάρχει απώλεια ενέργειας.

Η τάση που δημιουργείται από την εξωτερική αντίσταση 100Ω RRAMP1 τροφοδοτεί τον ακροδέκτη I<sub>sen</sub>. Η αντίσταση 1.4kΩ RCOMP1 σε σειρά με τον πυκνωτή 300nF CCOMP2 και όλο αυτό παράλληλα με τον πυκνωτή 3.3nF CCOMP1, συνδέονται με τον ακροδέκτη COMP του ολοκληρωμένου LM3478MM1, ο οποίος παρέχει αντιστάθμιση για τον βρόχο ελέγχου. Ο πυκνωτής 4.7μF CSEP1 απομονώνει την είσοδο από την έξοδο για την αποφυγή βραχυκυκλώματος. Οι αντιστάσεις 97.6kΩ RFB1 και 10kΩ RFB2 δημιουργούν έναν διαιρέτη τάσης και παρέχουν 1.26V στον ακροδέκτη FB του ολοκληρωμένου LM3478MM. Η πύλη του MOSFET M1 συνδέεται στον ακροδέκτη DR. Η αντίσταση 53.6 kΩ RFADJ1 που συνδέεται στον ακροδέκτη FA/SD καθορίζει την συχνότητα του ταλαντωτή. Ένα υψηλό επίπεδο τάσης σε αυτό το pin για περισσότερο από 30 μsec, απενεργοποιεί τη συσκευή. Οι πυκνωτές 22μF CIN και 100μF CBP1, που είναι τοποθετημένοι παράλληλα, συνδέονται στον ακροδέκτη τροφοδοσίας του ολοκληρωμένου έτσι ώστε να έχουμε μείωση του θορύβου στην είσοδο. Επίσης χρησιμοποιήθηκαν δυο πηνία, 4.7μH LIN1 και 47μH LOUT1 για την υποβάθμιση της τάσης εισόδου στα 13.5V. Τέλος η διάοδος D1 σε συνδυασμό με τον πυκνωτή 150μF COUT1 δημιουργούν έναν ανορθωτή τάσης στην έξοδο του κυκλώματος. Στο Σχήμα 37 παρουσιάζεται το σχηματικό διάγραμμα του κυκλώματος της κάρτας.



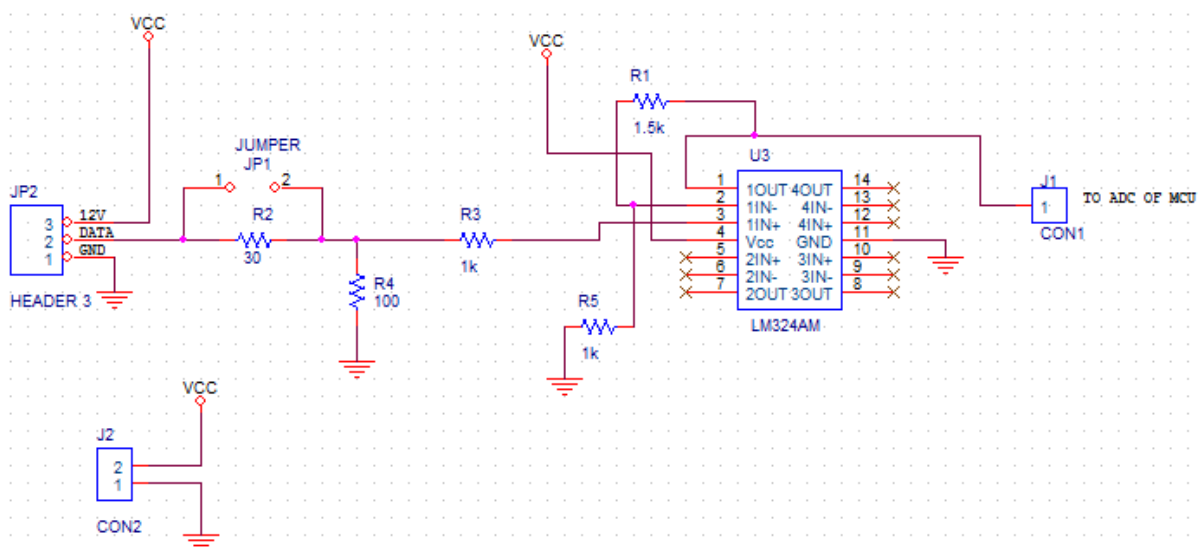
Σχήμα 37. Σχηματικό διάγραμμα της κάρτας προσαρμογής του φωτοβολταϊκού πάνελ στη μπαταρία

## 2.4.5 Κάρτα μετατροπής 4-20mA σε 1-5V

Η κάρτα μετατροπής 4-20mA σε 1-5V [43] προσαρμόζει το εξερχόμενο σήμα από το υποσύστημα προσδιορισμού της στάθμης του ποταμού, το οποίο εισέρχεται στο υποσύστημα υπολογισμού της στιγμιαίας παροχής. Το σήμα είναι ρεύμα στη περιοχή των 4-20mA και μέσω της κάρτας μετατρέπεται σε τάση στην περιοχή των 1-5V ώστε να ψηφιοποιηθεί στον ADC του μικροελεγκτή.

Το κύριο στοιχείο της κάρτας είναι ένας τελεστικός ενισχυτής. Ο τελεστικός ενισχυτής έχει υψηλή απολαβή, μεγάλη αντίσταση εισόδου και μικρή αντίσταση εξόδου. Στην κάρτα που κατασκευάστηκε λειτουργεί ως διαφορικός ενισχυτής, και ενισχύει τη διαφορά δύο σημάτων. Είναι επιθυμητό να έχουμε υψηλή τάση εξόδου όταν εφαρμόζεται διαφορικό σήμα και μηδενική τάση εξόδου στην εφαρμογή κοινού σήματος.

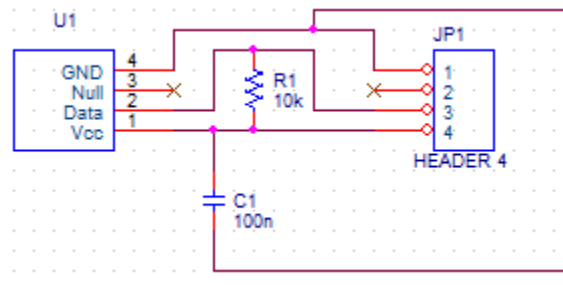
Στήν κάρτα που αναπτύχθηκε χρησιμοποιήθηκε ως διαφορικός ενισχυτής που ενισχύει τη διαφορά μεταξύ δύο σημάτων, το ολοκληρωμένο LM324AM [44] στο Σχήμα 38 το U3. Το σήμα εισόδου της κάρτας, είναι μια πηγή ρεύματος 4-20mA, η οποία συνδέεται με τη μη ανάστροφη είσοδο του τελεστικού ενισχυτή μέσω μιας αντίστασης 1kΩ R3, μιας αντίστασης 100Ω R4 ως προς τη γη και μιας επιλογής που μπορεί μέσω ενός διακόπτη να εισάγει μια αντίσταση 30kΩ R2 σε σειρά με το σήμα εισόδου. Η επιλογή αυτή εφαρμόζεται για λόγους προστασίας όταν οι αισθητήρες 4-20mA λειτουργούν με τάση 24V. Η αντίσταση των 100Ω R4 παίζει τον ρόλο του φόρτου για το σήμα εισόδου και στην αντίσταση εμφανίζεται τάση εισόδου 0.4V για ρεύμα εισόδου 4mA. Η αντίσταση ανάδρασης 1.5kΩ R1 μαζί με την αντίσταση 1kΩ R5 που ενώνεται με την αναστρέφουσα είσοδο του τελεστικού, ρυθμίζει την απολαβή του τελεστικού ώστε για ρεύμα εισόδου 20mA η τάση εξόδου να είναι 5V. Η ακρίβεια και η γραμμικότητα του κυκλώματος σχολιάζεται στο Κεφάλαιο 4. Η κάρτα αυτή χρησιμοποιείται για τη διασύνδεση του υποσυστήματος προσδιορισμού της στάθμης του νερού του ποταμού, με τους ADCs του μικροελεγκτή του υποσυστήματος υπολογισμού της στιγμιαίας παροχής. Στο Σχήμα 38 φαίνεται το σχηματικό διάγραμμα του κυκλώματος της κάρτας μετατροπής 4-20mA σε 1-5V.



Σχήμα 38. Σχηματικό διάγραμμα κάρτας μετατροπής 4-20mA σε 1-5V

## 2.4.6 Κάρτα μέτρησης θερμοκρασίας – υγρασίας

Το σχηματικό διάγραμμα του κυκλώματος της κάρτας μέτρησης θερμοκρασίας – υγρασίας η οποία κατασκευάστηκε, φαίνεται στο Σχήμα 39.



Σχήμα 39. Σχηματικό διάγραμμα κάρτας μέτρησης θερμοκρασίας – υγρασίας

Οι ακροδέκτες του αισθητήρα θερμοκρασίας – υγρασίας DHT22 καταλήγουν σε έναν αρσενικό ακροδέκτη JP1 για την σύνδεση με τον μικροελεγκτή. Μια pull-up αντίσταση των 10kΩ R1 ανάμεσα στα data του αισθητήρα και την τροφοδοσία Vcc εξασφαλίζει ότι το σήμα διατηρείται σε καλά καθορισμένο λογικό επίπεδο (αποφυγή floating). Ο πυκνωτής 100nF C1, ανάμεσα στην τροφοδοσία και στη γείωση του αισθητήρα, λειτουργεί σαν φίλτρο. Ο φορτισμένος πυκνωτής απελευθερώνει φορτίο όταν υπάρχει πτώση τάσης στην τροφοδοσία, με αποτέλεσμα να παρέχεται στον αισθητήρα σταθερή τροφοδοσία 5V.

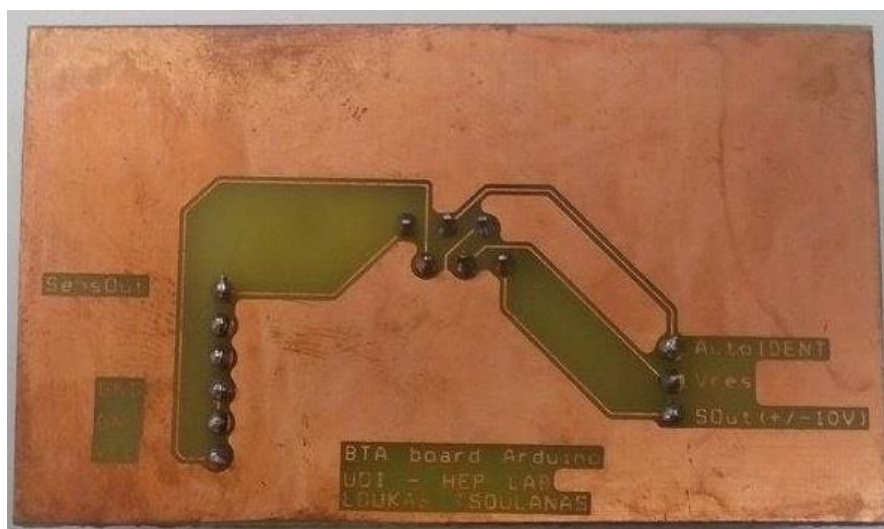
## 2.5 Υλοποίηση του συστήματος

Η σχεδίαση των τυπωμένων κυκλωμάτων των ηλεκτρονικών καρτών, έγινε στο υποπρόγραμμα LAYOYT του OrCAD [39]. Τα τυπωμένα κυκλώματα εκτυπώθηκαν σε ριζόχαρτα από εκτυπωτή με υψηλή ακρίβεια, αφού ακολουθήθηκε η απαραίτητη διαδικασία στο πρόγραμμα Layout για τη δημιουργία των αρχείων Gerber [45] η οποία αναλύεται στο Παράρτημα 3. Τα εκτυπωμένα ριζόχαρτα τοποθετήθηκαν πάνω σε φωτοευαίσθητες πλακέτες και τα σχέδια μεταφέρθηκαν σε αυτές με τη χρήση σαρωτή με λάμπα ακτινοβολίας UV. Ο χρόνος έκθεσης στην ακτινοβολία UV ήταν περίπου 5 λεπτά. Ακολούθησε η διαδικασία εμφάνισης των κυκλωμάτων στις πλακέτες με εμφότιση σε διάλυμα K<sub>2</sub>O<sub>2</sub> (υπεροξειδίου του καλίου) και νερού σε αναλογία 10ml K<sub>2</sub>O<sub>2</sub> σε 700ml H<sub>2</sub>O, για 3 έως 6 λεπτά. Τέλος οι πλακέτες τοποθετήθηκαν μέσα σε ένα χημικό διάλυμα που περιείχε 350ml H<sub>2</sub>O, 120ml HCl (υδροχλωρικό οξύ) και 80ml H<sub>2</sub>O<sub>2</sub> (υπεροξειδίου του υδρογόνου-Peridrol), έτσι ώστε να γίνει η αποχάλκωση. Ο χρόνος που εκτέθηκαν οι πλακέτες μέσα στο διάλυμα ήταν περίπου 2 λεπτά, ικανός για σωστή αποχάλκωση στα σημεία που αυτή απαιτείται και να μην καταστραφούν οι λεπτοί διάδρομοι. Οι ηλεκτρονικές κάρτες κατασκευάστηκαν στο Εργαστήριο Φυσικής Υψηλών Ενεργειών του Τμήματος Φυσικής και παρουσιάζονται στις επόμενες

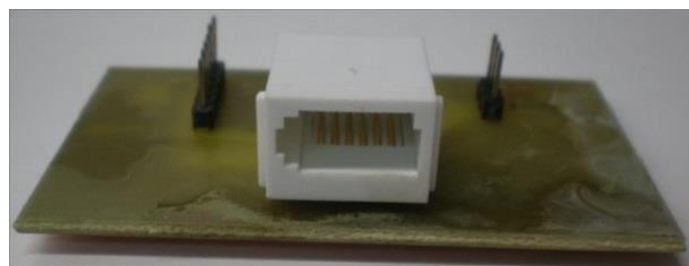
παραγράφους, ενώ τα σχέδια των τυπωμένων κυκλωμάτων παρατίθενται στο Παράτημα 3.

### 2.5.1 Οι κάρτες του συστήματος

Η κάρτα προσαρμογής βύσματος τηλεπικοινωνιών βρετανικού τύπου σε Arduino, είναι μονής όψης και παρουσιάζεται στα Σχήματα 40 και 41. Στο πρώτο σχήμα φαίνεται η επάνω όψη (top layer) της κάρτας και στο άλλο η κάτω όψη της (bottom layer) με το θηλυκό βύσμα τηλεπικοινωνιών βρετανικού τύπου.



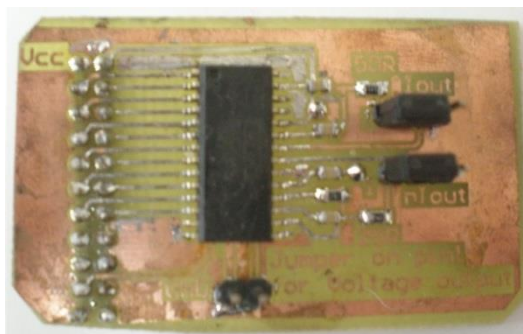
Σχήμα 40. Επάνω όψη της κάρτας προσαρμογής βύσματος τηλεπικοινωνιών βρετανικού τύπου σε Arduino



Σχήμα 41. Κάτω όψη της κάρτας με το θηλυκό βύσμα τηλεπικοινωνιών βρετανικού τύπου

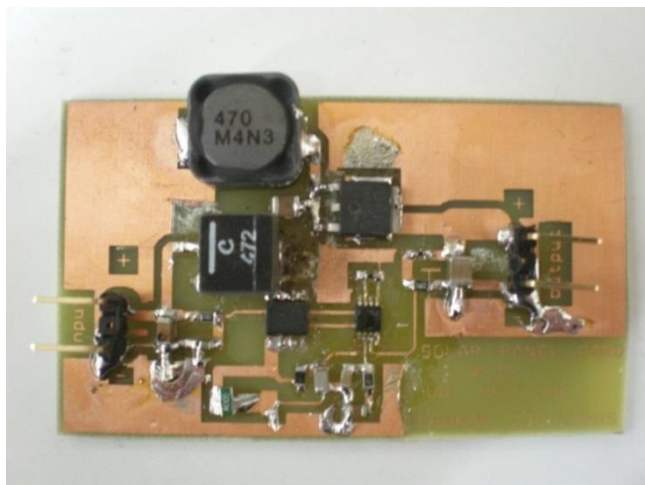
Η κάρτα μετατροπής ψηφιακού σήματος σε πρότυπο 4-20mA είναι ενός επιπέδου. Η επάνω όψη (top layer) της ηλεκτρονικής κάρτας με τον μετροπέα ψηφιακού σήματος σε αναλογικό DAC902, παρουσιάζεται στο Σχήμα 42.





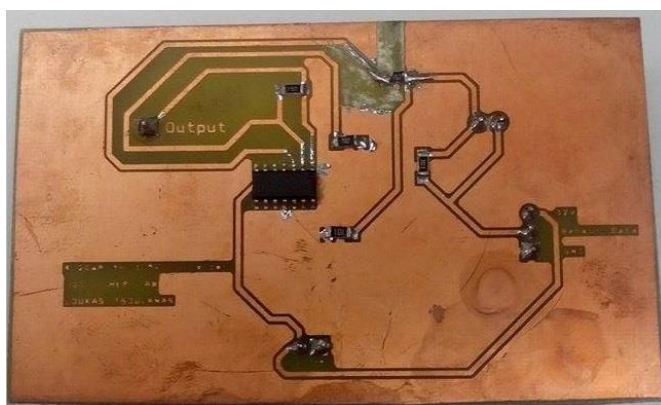
Σχήμα 42. Επάνω όψη της κάρτας μετατροπής ψηφιακού σήματος σε πρότυπο 4-20mA

Η κάρτα προσαρμογής του φωτοβολταϊκού πάνελ στη μπαταρία η οποία κατασκευάστηκε για την ομαλή φόρτισή της, είναι ενός επιπέδου και φαίνεται στο Σχήμα 43. Η κάρτα διαθέτει ακροδέκτες εισόδου για τη σύνδεσή της με το φωτοβολταϊκό πάνελ και εξόδου για την τάση 13.5V που παράγεται.

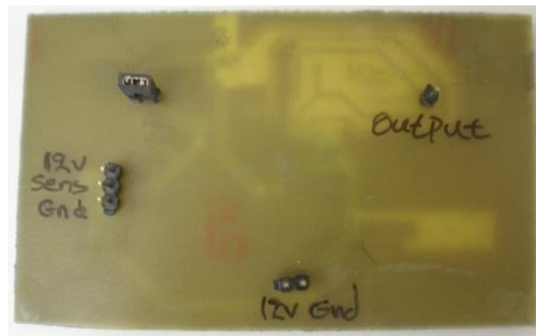


Σχήμα 43. Επάνω όψη της κάρτας προσαρμογής του φωτοβολταϊκού πάνελ στη μπαταρία

Η κάρτα μετατροπής 4-20mA σε 1-5V είναι ενός επιπέδου και φαίνεται στα Σχήματα 44 και 45. Η κάρτα διαθέτει ακροδέκτες για τη τροφοδοσία της, για την σύνδεση με αισθητήρα αλλά και για την τάση εξόδου που παράγεται.



Σχήμα 44. Επάνω όψη κάρτας μετατροπής 4-20mA σε 1-5V



Σχήμα 45. Κάτω όψη κάρτας μετατροπής 4-20mA σε 1-5V

Η κάρτα μέτρησης θερμοκρασίας – υγρασίας η οποία παρουσιάζεται στα Σχήματα 46 και 47, είναι ενός επιπέδου. Διαθέτει μία υποδοχή για τον αισθητήρα θερμοκρασίας-υγρασίας DHT22 και ακροδέκτες για την σύνδεση της κάρτας με τον μικροελεγκτή ATmega2560 της πλακέτας Arduino mega 2560.



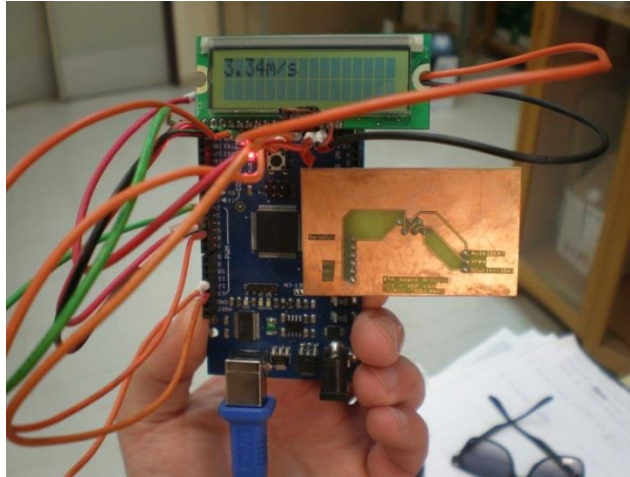
Σχήμα 46. Κάτω όψη κάρτας μέτρησης θερμοκρασίας-υγρασίας



Σχήμα 47. Επάνω όψη κάρτας μέτρησης θερμοκρασίας-υγρασίας

## 2.5.2 Σύστημα μέτρησης της τρέχουσας ταχύτητας του νερού

Στα Σχήματα 48 και 49 φαίνονται ο υπολογιστής ταχύτητας και το ροόμετρο Vernier αντίστοιχα, τα οποία είναι τα μέρη του συστήματος μέτρησης της τρέχουσας ταχύτητας του νερού.



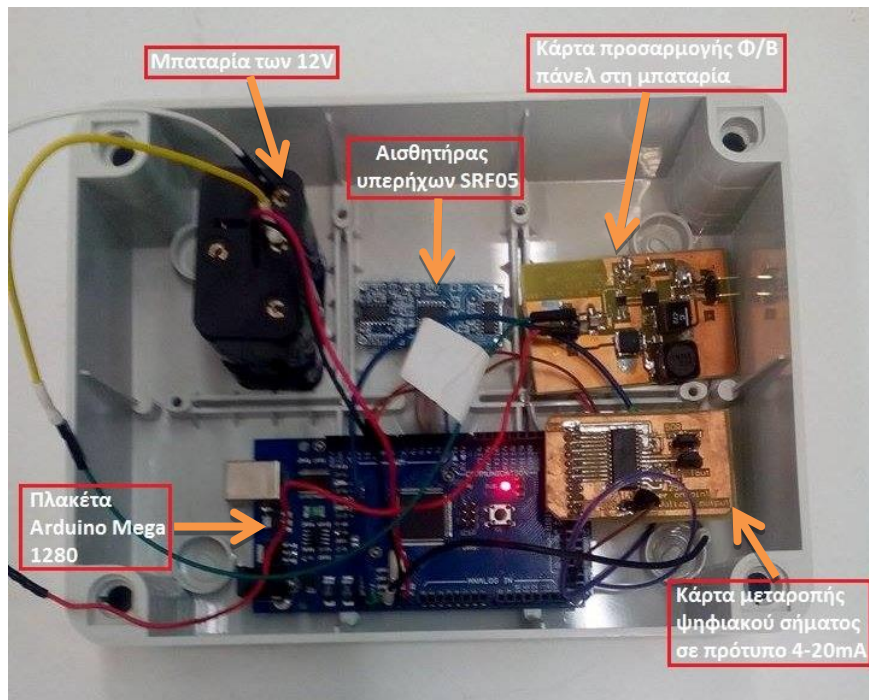
Σχήμα 48. Υπολογιστής ταχύτητας συστήματος μέτρησης της τρέχουσας ταχύτητας νερού



Σχήμα 49. Ροόμετρο Vernier συστήματος μέτρησης της τρέχουσας ταχύτητας νερού

## 2.5.3 Υποσύστημα προσδιορισμού της στάθμης του νερού του ποταμού

Το υποσύστημα προσδιορισμού της στάθμης του νερού του ποταμού τοποθετημένο σε κουτί με προστασία IP56 [46] (προστασία έναντι σκόνης και από νερό που εκτοξεύεται με δύναμη) παρουσιάζεται στο Σχήμα 50 και η κάτω όψη του κουτιού με τον πομπό και τον δέκτη του αισθητήρα υπερήχων SRF05 στο Σχήμα 51.



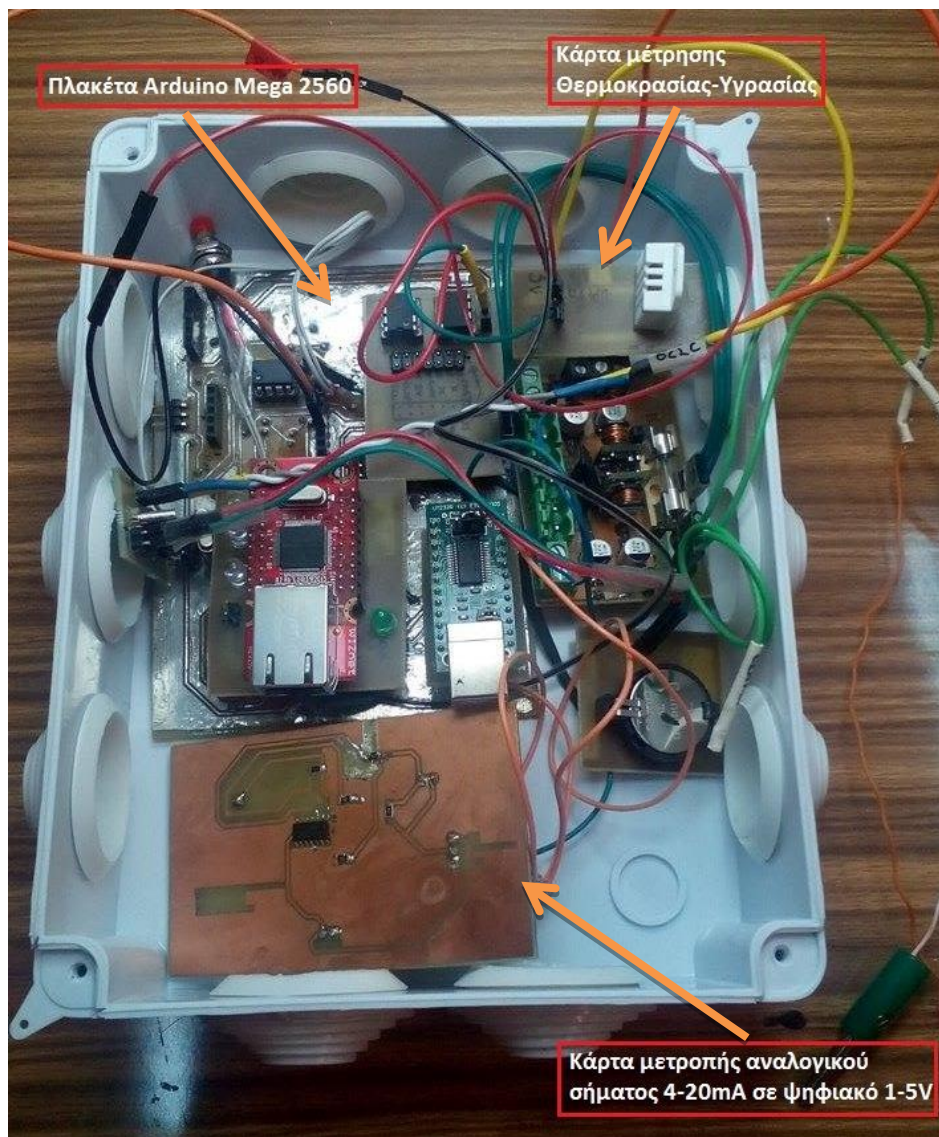
Σχίμα 50. Το υποσύστημα προσδιορισμού της στάθμης του ποταμού τοποθετημένο σε στεγανό κουτί



Σχίμα 51. Κάτω όψη του κουτιού του υποσυστήματος προσδιορισμού της στάθμης του ποταμού

#### 2.5.4 Υποσύστημα υπολογισμού της στιγμιαίας παροχής νερού του ποταμού

Το υποσύστημα που χρησιμοποιήθηκε για τον υπολογισμό της στιγμιαίας παροχής του νερού, με όλα τα μέρη του διασυνδεδεμένα μεταξύ τους και τοποθετημένο σε κουτί με προστασία IP65 [46] (αδιαπέραστο από σκόνη και νερό που εκτοξεύεται από οποιαδήποτε κατεύθυνση) παρουσιάζεται στο Σχίμα 52.



Σχημα 52. Το υποσύστημα υπολογισμού της στιγμιαίας παροχής νερού με όλα τα μέρη του διασυνδεδεμένα τοποθετημένο σε κουτί



### **3. Διαδικασία μέτρησης της παροχής νερού του ποταμού - Λογισμικό**

Στο κεφάλαιο αυτό περιγράφεται η διαδικασία μέτρησης της ταχύτητας του νερού του ποταμού και η διαδικασία υπολογισμού της στιγμιαίας παροχής για τον σταθμό μέτρησης στον ποταμό Άραχθο που κατασκευάστηκε στη γέφυρα της Πολιτσάς στα πλαίσια του έργου *wwqm\_avr* [27]. Επίσης περιγράφεται το λογισμικό που χρησιμοποιήθηκε στα επιμέρους υποσυστήματα. Ο κώδικας για το σύστημα μέτρησης της τρέχουσας ταχύτητας του νερού, ο κώδικας του υποσυστήματος προσδιορισμού της στάθμης και ο κώδικας του υποσυστήματος υπολογισμού της στιγμιαίας παροχής παρατίθενται στο συνημμένο Παράρτημα 2. Στο Παράρτημα 2, παρατίθενται επίσης ο κώδικας που χρησιμοποιείται για τον έλεγχο της ορθής λειτουργίας της κάρτας μετατροπής 4-20mA σε 1-5V, καθώς και ο κώδικας που χρησιμοποιείται για τον έλεγχο της ορθής λειτουργίας της κάρτας μετατροπής ψηφιακού σήματος σε πρότυπο 4-20mA.

#### **3.1 Διαδικασία μέτρησης ταχύτητας του νερού σε κατακόρυφη τομή του ποταμού σε περιοχή όπου η κοίτη είναι σταθερή και η ταχύτητα εξαρτάται από τη θέση της επιφάνειας του νερού**

Για τον υπολογισμό της παροχής απαιτείται να μετρηθεί η ταχύτητα του νερού του ποταμού σε διάφορα υποτμήματα μιας τομής της κοίτης του. Για το λόγο αυτό επιλέχθηκε μια διατομή της κοίτης του ποταμού και χωρίστηκε σε υποτμήματα κατά μήκος και βάθος. Έτσι έγινε μια χαρτογράφηση της ταχύτητας του νερού στην αναφερόμενη τομή (Σχήμα 53).

Σύμφωνα με μετρήσεις που έγιναν τα μέσα Απριλίου βρέθηκε ότι το βάθος του ποταμού κυμνείται από 1.35m το ελάχιστο έως 1.9m το μέγιστο. Η διατομή της κοίτης χωρίστηκε συγκεκριμένα σε 118 υποτμήματα, 30 τμήματα κατά μήκος και 3 έως 5 τμήματα κατά βάθος (ανάλογα με το βάθος στο συγκεκριμένο σημείο). Επομένως υπάρχουν 118 υποτμήματα με εμβαδόν  $0.5m * 0.5m = 0.25m^2$ . Λόγω όμως της μορφολογίας της κοίτης κάποια από τα πλαϊνά υποτμήματα επικαλύπτονται από βράχους οπότε είτε αφαιρούνται τελείως είτε συνυπολογίζονται με κάποιο ποσοστό του εμβαδού των  $0.25m^2$ . Το σύστημα μέτρησης της τρέχουσας ταχύτητας του νερού τοποθετήθηκε σε κάθε ένα από αυτά τα υποτμήματα ξεχωριστά και μετρήθηκε η ταχύτητα του νερού. Τα αποτελέσματα της χαρτογράφησης φαίνονται αναλυτικά στο Σχήμα 53.

DEPTH																																
10cm	1.04	1.58	2.07	2.38	2.13	2.13	2.04	1.98	2.29	2.19	2.19	1.98	2.29	2.29	2.13	2.07	1.98	2.13	1.95	1.92	1.95	1.77	1.68	1.74	1.74	1.77	1.68	1.16	0.76	0.46		
50cm	0.76	0.88	1.22	1.28	1.52	1.58	1.89	1.98	1.92	1.83	1.89	1.74	1.83	1.68	1.68	1.98	1.89	1.92	1.8	1.8	1.83	1.58	1.52	1.58	1.46	1.46	1.52	1.22	0.82	0.67		
1m	0.61	1.52	1.62	1.46	1.74	1.74	1.89	1.83	2.04	1.92	1.8	1.74	1.68	1.46	1.46	1.62	1.58	1.52	1.98	1.52	1.74	1.52	1.52	1.68	1.4	1.52	1.58	1.1	0.46	0.4		
1.5m	rock	rock	rock	rock	rock	rock	rock	rock	rock	0.06	0.06	0.12	0.24	0.98	0.85	0.49	0.61	0.98	1.22	1.16	0.76	0.61	1.04	0.91	0.98	0.91	0.85	0.94	0.52	rock		
1.8m	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	0.03	0.09	0.4	0.46	0.76	0.61	0.82	0.46	rock	

Σχήμα 53. Χάρτης ταχύτητας νερού στην κάθετη διατομή της κοίτης του ποταμού

Η μονάδα της ταχύτητας είναι m/sec.

Στη συνέχεια δημιουργήθηκε για κάθε κάθετη στήλη νερού η γραφική παράσταση ύψους (z) – ταχύτητας (u). Παρατηρήθηκε ότι η σχέση μεταξύ αυτών των δύο μεγεθών είναι σχέση παραβολική. Αυτό είναι αναμενόμενο καθώς προκύπτει και από τη θεωρία που αναφέρθηκε στην παράγραφο 1.1 (σελ. 7).

Ακολούθως έγινε έλεγχος συμβατότητας των μετρήσεων της ταχύτητας του νερού του ποταμού, σε αυτά τα σημεία που καλύπτουν την τομή του και βρίσκονται σε περιοχή όπου η κοίτη είναι σταθερή και αλλάζει μόνο το ύψος της επιφάνειας του νερού, με την προσεγγιστική σχέση  $u = \frac{\rho g \theta}{2\eta} z(2h - z)$ .

Γνωρίζοντας την ταχύτητα σε κάθε ένα από τα τμήματα της τομής, καθώς και το ύψος τους μέσα στον ποταμό, προσδιορίστηκε από την  $u=f(z)$  ο παράγοντας  $\frac{\rho g \theta}{2\eta}$  που φαίνεται στο Σχήμα 54 και αναμένεται να είναι σταθερός σε όλα τα τμήματα.

0.29	0.44	0.58	0.66	0.6	0.59	0.64	0.64	0.64	0.61	0.61	0.55	0.64	0.64	0.59	0.58	0.64	0.59	0.54	0.53	1.11	0.49	0.47	0.48	0.48	0.49	0.47	0.32	0.21	0.13		
0.22	0.26	0.35	0.37	0.5	0.46	0.57	0.57	0.56	0.54	0.55	0.5	0.53	0.49	0.49	0.57	0.55	0.56	0.55	0.52	0.55	0.46	0.44	0.46	0.42	0.42	0.44	0.35	0.24	0.19		
0.22	0.54	0.58	0.52	0.6	0.62	0.67	0.65	0.73	0.69	0.64	0.62	0.6	0.53	0.57	0.64	0.6	0.54	0.71	0.54	0.62	0.54	0.54	0.6	0.5	0.54	0.57	0.39	0.16	0.14		
rock	rock	rock	rock	rock	rock	rock	rock	rock	0.04	0.04	0.07	0.15	0.59	0.52	0.3	0.37	0.59	0.74	0.7	0.46	0.37	0.63	0.55	0.59	0.55	0.52	0.57	0.31	rock		
rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	0.04	0.13	0.55	0.64	1.06	0.85	1.14	0.64	rock	
rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock	rock

Σχήμα 54. Παράγοντας  $\frac{\rho g \theta}{2\eta}$  κατά μήκος και βάθος της διατομής της κοίτης του ποταμού Αράχθου

Στις μεσαίες γραμμές ο παράγοντας  $\rho g \theta / 2\eta$  είναι σταθερός όπως αναμενόταν ενώ στην επιφάνεια και τον πυθμένα παρουσιάζει απόκλιση η οποία οφείλεται στο γεγονός ότι η ροή δέν είναι στρωτή.

Ο μέσος όρος των τιμών του Σχήματος 54, είναι  $\bar{A} = 0.49$ . Επίσης με τη βοήθεια αλφαδιού, μετρήθηκε η κλίση του ποταμού σε σχέση με τον ορίζοντα, κάτω από τη γέφυρα. Παρατηρήθηκε ότι σε μήκος 100 μέτρων, λόγω της γεωμετρίας της κοίτης, η κλίση του ποταμού ήταν σχεδόν μηδενική στο σημείο αυτό. Έτσι για τους υπολογισμούς θεωρήθηκε πως το ποτάμι σε αυτό το σημείο έχει κλίση 1mm ανά 100m. Η γωνία κλίσης του ποταμού ως προς τον ορίζοντα είναι  $\theta = 0.00001^\circ$

και  $\sin\theta = 0.00000017$  ή  $1.7 \cdot 10^{-7}$

Θεωρώντας τη σχέση του παράγοντα  $A = \frac{\rho g \theta}{2\eta}$  και λύνοντας ως προς το ιξώδες ( $\eta$ ), προσδιορίζεται η τιμή του.

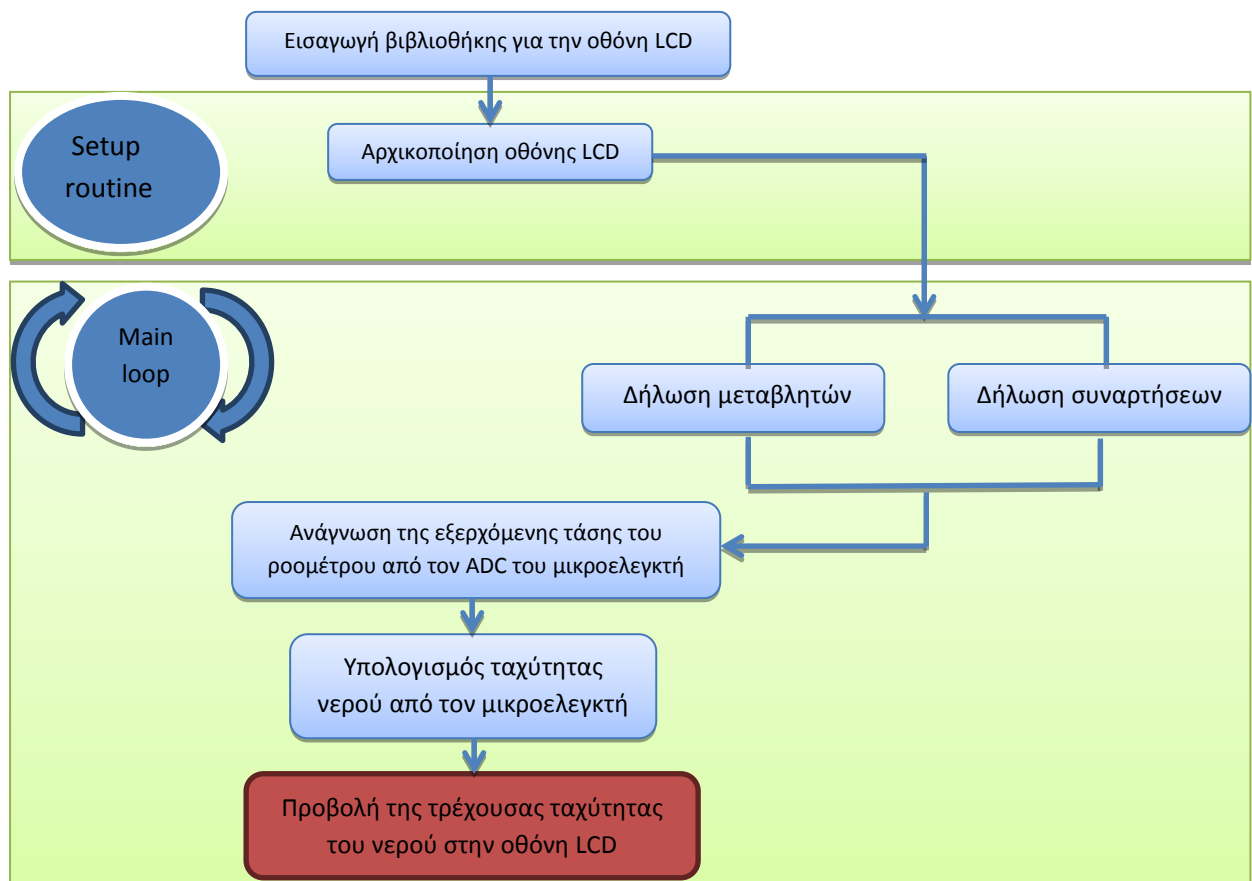


Προκύπτει ότι 
$$\eta = \frac{\rho g \theta}{2A} = \frac{999 \frac{kg}{m^3} \cdot 9.8 \frac{m}{s^2} \cdot 1.7 \cdot 10^{-7}}{2 \cdot 0.49 \frac{1}{m \cdot s}} = 1.6 \cdot 10^{-3} \frac{kg}{m \cdot s} \text{ ή } 1.6 mPa \cdot S$$

Από πίνακες ιδώδους του νερού σε συνάρτηση με τη θερμοκρασία του [47], παρατηρείται πως η υπολογισθείσα αυτή τιμή αντιστοιχεί σε νερό θερμοκρασίας περίπου  $0^\circ - 5^\circ C$ , που είναι συμβατό με την πραγματικότητα.

Η σχέση της ταχύτητας σε συνάρτηση με το βάθος είναι όπως προαναφέρθηκε παραβολική. Ωστόσο λογαριθμίζοντας το βάθος η σχέση γίνεται γραμμική. Εν κατακλείδι, με μια απλή μέτρηση της στάθμης (h), είναι γνωστή η ταχύτητα του νερού σε κάθε ύψος (z) στην τομή, από τη σχέση  $\log z = f(u)$  κάθε μιας από τις 30 στήλες νερού. Οι συναρτησιακές σχέσεις  $\log z = f(u)$  για κάθε στήλη παρουσιάζονται στο Παράρτημα 5.

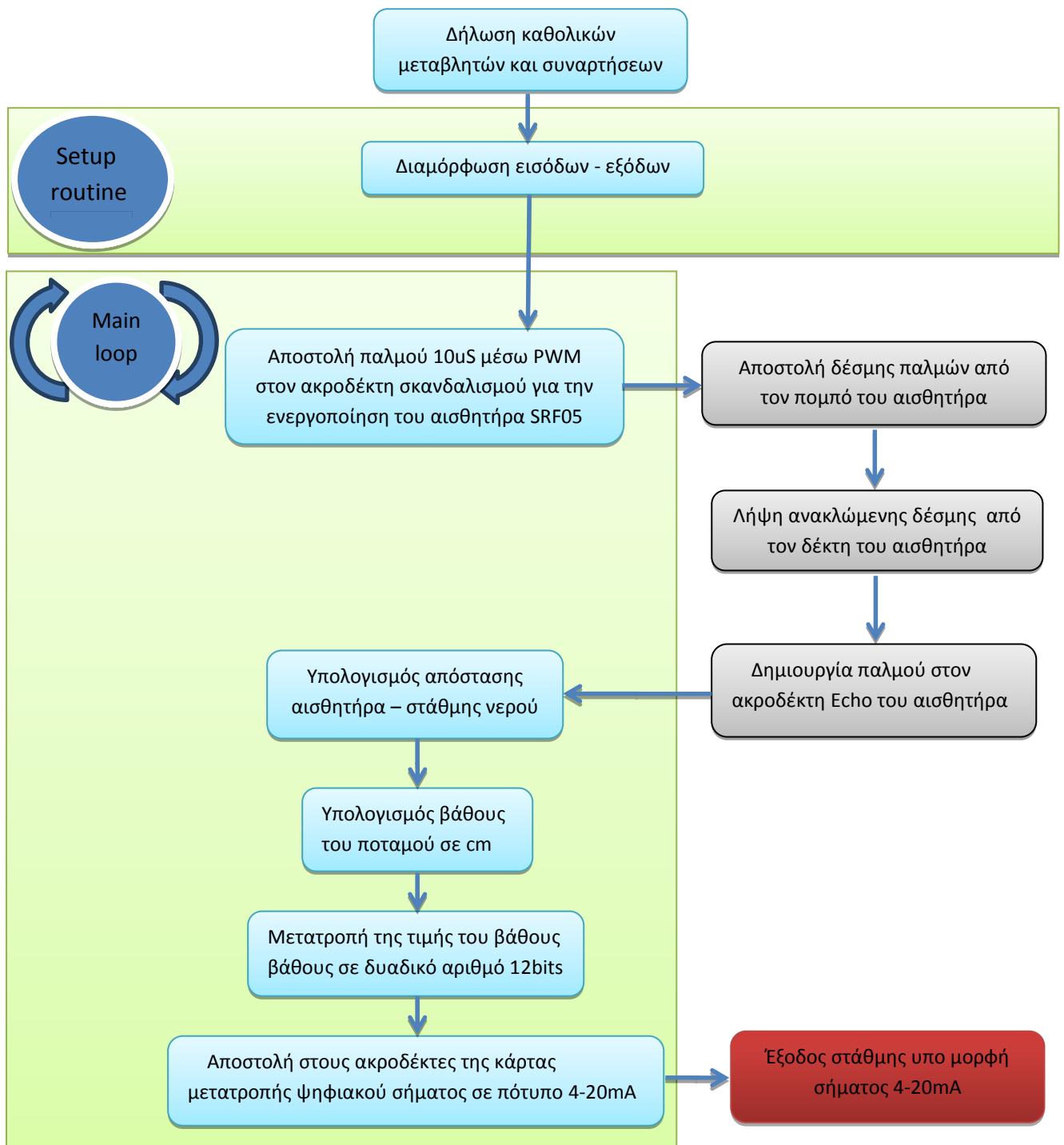
Για την περίπτωση χρήσης του συστήματος μέτρησης της τρέχουσας ταχύτητας του νερού Vernier αναπτύχθηκε μια εφαρμογή στον μικροελεγκτή του συστήματος Arduino Mega 1280. Ο ADC του μικροελεγκτή της πλακέτας Arduino Mega 1280 λαμβάνει την τιμή από τον αισθητήρα μέτρησης ταχύτητας του νερού Vernier και την ψηφιοποιεί (8bit). Στη συνέχεια αυτή η τιμή πολλαπλασιάζεται με τη σταθερά που ισούται με  $1m/sec/V = 1m/sec/1024bits/5V$  και έτσι προκύπτει η ταχύτητα σε m/sec η οποία αποστέλλεται στην οθόνη LCD για ανάγνωση. Το πρόγραμμα καταλαμβάνει περίπου το 6% της συνολικής μνήμης flash. Στο Σχήμα 55 φαίνεται το διάγραμμα ροής του λογισμικού του συστήματος μέτρησης της στιγμιαίας ταχύτητας του νερού.



Σχήμα 55. Διάγραμμα ροής λογισμικού συστήματος μέτρησης της τρέχουσας ταχύτητας νερού

## 3.2 Περιγραφή της διαδικασίας προσδιορισμού της στάθμης του νερού του ποταμού

Το λογισμικό που εγκαταστάθηκε και λειτουργεί στο υποσύστημα προσδιορισμού της στάθμης του ποταμού καταλαμβάνει περίπου το 5% της μνήμης flash του μικροελεγκτή. Στο Σχήμα 55 φαίνεται το διάγραμμα ροής του λογισμικού του υποσυστήματος προσδιορισμού της στάθμης του νερού του ποταμού.



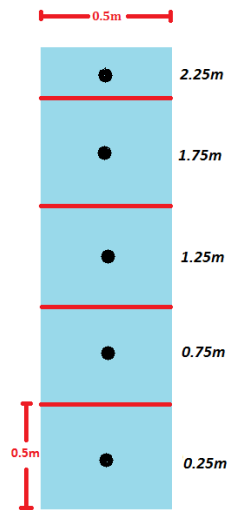
Σχήμα 55. Διάγραμμα ροής λογισμικού του υποσυστήματος προσδιορισμού της στάθμης του νερού του ποταμού

Όπως ήδη αναφέρθηκε στο Κεφάλαιο 2, το υποσύστημα προσδιορισμού της στάθμης του νερού του ποταμού αποτελείται από την πλατφόρμα Arduino Mega 1280, τον αισθητήρα μέτρησης απόστασης με υπερήχους SRF05 και την κάρτα μετατροπής ψηφιακού σήματος σε πρότυπο 4-20mA. Το υποσύστημα τροφοδοτείται από μία μπαταρία των 12V, η οποία φορτίζεται από φωτοβολταϊκό πάνελ μέσω μιας κάρτας προσαρμογής. Όταν μετρήθηκε η ταχύτητα του νερού (παράγραφος 3.1), η απόσταση του αισθητήρα από τη στάθμη του νερού του ποταμού ήταν 4.97m, και το βάθος του ποταμού κατακόρυφα στην άκρη της όχθης ήταν 1.35m. Άρα η απόσταση αισθητήρα – πυθμένα στην άκρη του ποταμού ήταν 6.32m. Αφαιρώντας από τα 6.32m την απόσταση αισθητήρα – στάθμης που υπολογίζεται από το υποσύστημα προσδιορισμού της στάθμης, προκύπτει το ύψος της στάθμης του ποταμού τη δεδομένη στιγμή.

### 3.3 Περιγραφή της διαδικασίας υπολογισμού της στιγμιαίας παροχής του νερού του ποταμού

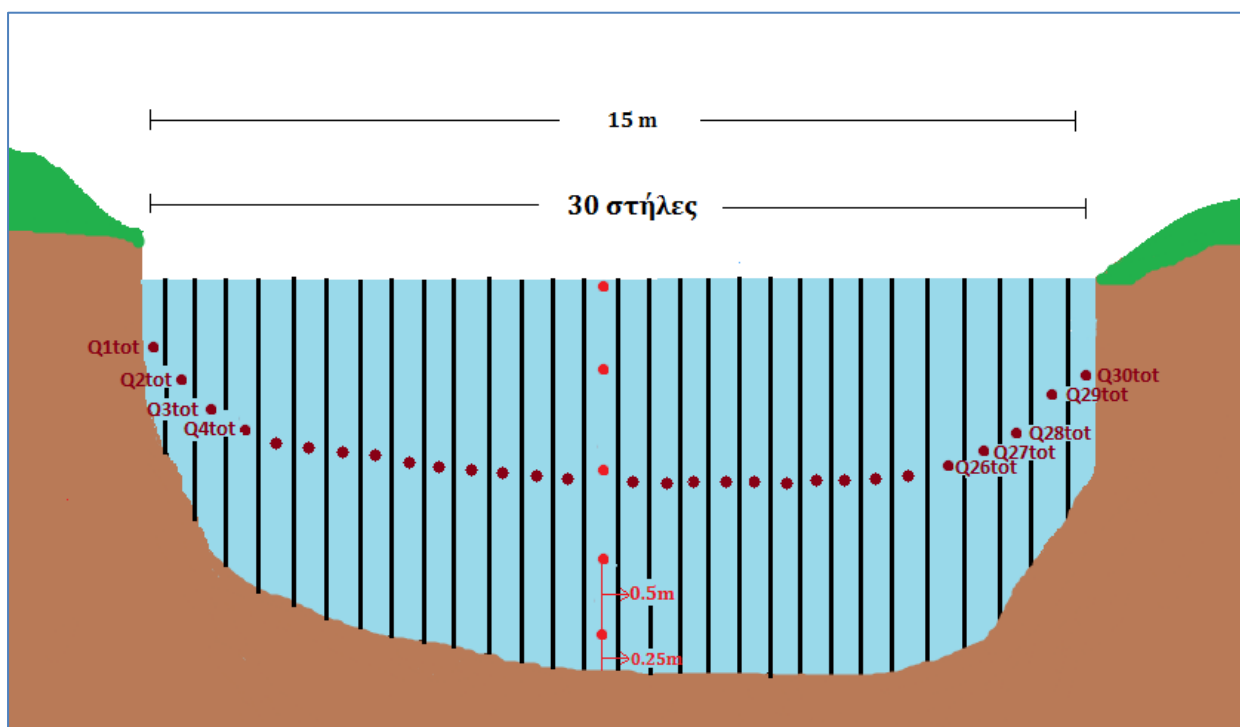
Στο έργο `wwqm_avg`, το υποσύστημα υπολογισμού της στιγμιαίας παροχής του νερού του ποταμού, λειτουργεί συγκεκριμένες ώρες την ημέρα. Ειδικότερα τροφοδοτείται τις ώρες 00:00, 06:00, 12:00 και 18:00. Το σύστημα τροφοδοσίας [26] βασίζεται σε φωτοβολταϊκά με συσκευές αποθήκευσης ενέργειας (υπερπυκνωτές). Η κεντρική κάρτα του υποσυστήματος [32] περιέχει έναν μικροελεγκτή και ένα ρολόι πραγματικού χρόνου. Στις παραπάνω αναφερόμενες ώρες το σύστημα τροφοδοσίας εκτός των άλλων, τροφοδοτεί με 12V το υποσύστημα υπολογισμού της στιγμιαίας παροχής. Τότε αρχίζει να εκτελείται ο κώδικας του μικροελεγκτή.

Την πρώτη φορά της ημέρας που το σύστημα μπαίνει σε λειτουργία, δηλαδή στις 00:00, δημιουργείται ένα αρχείο με την τρέχουσα ημερομηνία στην κάρτα μνήμης, η οποία είναι ενσωματωμένη στη πλακέτα του Arduino Mega 2560, και αποθηκεύονται οι πρώτες μετρήσεις. Συγκεκριμένα η αναλογική τιμή 4-20mA της στάθμης, που προέρχεται από το υποσύστημα προσδιορισμού της στάθμης του νερού, μετατρέπεται σε αναλογική τιμή 1-5V, μέσω της κάρτας μετατροπής 4-20mA σε 1-5V που αναπτύχθηκε στα πλαίσια της παρούσας μεταπτυχιακής διπλωματικής εργασίας, και ψηφιοποιείται από τον ενσωματωμένο στον μικροελεγκτή ADC. Στη συνέχεια στον μικροελεγκτή εκτελείται η διαδικασία υπολογισμού της στιγμιαίας παροχής. Υπολογίζεται το ύψος για κάθε μία από τις 30 στήλες και έπειτα η ταχύτητα του νερού στο κέντρο κάθε τμήματος (50x50cm), ξεκινώντας από το σημείο που βρίσκεται 0.25m ψηλότερα από τον πυθμένα και συνεχίζοντας ανά 0.5m προς τα πάνω. Η ταχύτητα για κάθε τμήμα υπολογίζεται από τη σχέση  $u = \frac{\rho g \theta}{2\eta} z(2h - z)$ , χρησιμοποιώντας ως  $\rho g \theta / 2\eta$  την μέση τιμή κάθε στήλης νερού, εκτός από τα τμήτα που βρίσκονται στην επιφάνεια και τον βυθό και στα οποία ο παράγοντας αυτός διατηρείται ίδιος. Στο Σχήμα 56 φαίνεται μια μεμονωμένη στήλη της κατακόρυφης τομής χωρισμένη σε υποτμήματα (50x50cm) και τα σημεία στα οποία υπολογίζεται η ταχύτητα του νερού.



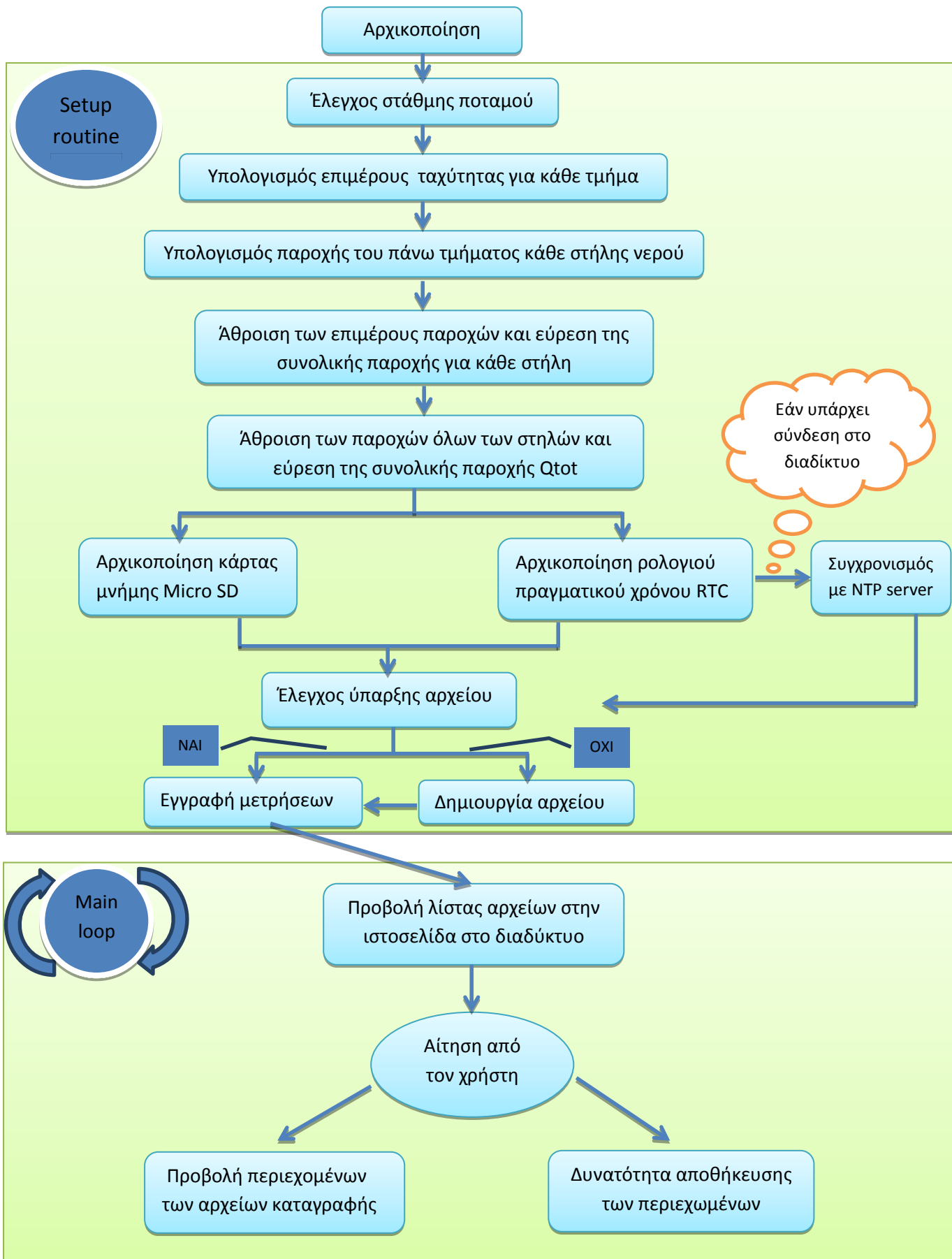
Εικόνα 56. Σημεία υπολογισμού ταχύτητας σε μία στήλη νερού

Γνωρίζοντας την ταχύτητα του νερού και το εμβαδόν κάθε τμήματος υπολογίζεται η παροχή για κάθε τμήμα της στήλης, από τη σχέση  $Q_i = U_i \cdot A_i$ . Οι τιμές των παροχών αποθηκεύονται και στη συνέχεια στο πρόγραμμα αθροίζονται όλες οι κάθετες επιμέρους παροχές των τμημάτων της στήλης, όπως φαίνεται στο Σχήμα 57, και υπολογίζεται η συνολική στιγμιαία παροχή για κάθε μία στήλη νερού  $Q_{i,tot}$ , σε m/sec. Πρέπει να σημειωθεί πως στο πάνω μέρος κάθε στήλης περισεύει κάθε φορά ένα κομμάτι του οποίου η παροχή συνυπολογίζεται και αυτή από τη σχέση  $u=f(z)$ , αλλά με το 0.95 της ταχύτητας λόγω της τριβής του αέρα στην επιφάνεια του νερού και με το εκάστοτε εμβαδόν που είναι μικρότερο η ίσο με  $0.25m^2$ .



Εικόνα 57. Κάθετη τομή της κοίτης του ποταμού χωρισμένη σε στήλες νερού

Με την άθροιση όλων των παροχών των 30 στηλών προκύπτει η συνολική στιγμιαία παροχή. Η τιμή αυτή μαζί με την τιμή της στάθμης, αλλά και τις τιμές της θερμοκρασίας και της σχετικής υγρασίας, που μετρώνται απο την κάρτα μέτρησης θερμοκρασίας - υγρασίας, αποθηκεύονται στο αρχείο της ημέρας που δημιουργήθηκε. Ενώ στις 00:00, 06:00 και 12:00, το υποσύστημα μέτρησης τροφοδοτείται για 10 λεπτά, το ασύρματο σύστημα μετάδοσης πληροφορίας παραμένει εκτός λειτουργίας. Το υποσύστημα τροφοδοτείται στις 18:00 για 15 λεπτά και στις 18:10 τροφοδοτείται και το ασύρματο σύστημα μετάδοσης, για να αποστείλει το αρχείο της ημέρας που δημιουργήθηκε στον τερματικό σταθμό, που βρίσκεται στο Τμήμα Φυσικής του Πανεπιστημίου Ιωαννίνων. Πρέπει να σημειωθεί πως λαμβάνεται μέτρηση μόνο όταν επαρκεί η ισχύς για να τροφοδοτηθεί το σύστημα. Το πρόγραμμα καταλαμβάνει περίπου το 24% της συνολικής μνήμης flash του μικροελεγκτή. Το λογισμικό παρατίθεται στο Παράρτημα 2 ενώ το διάγραμμα ροής του φαίνεται στο Σχήμα 58.



Σχήμα 58. Διάγραμμα ροής λογισμικού του υποσυστήματος υπολογισμού της στιγμιαίας παροχής του νερού

## 4. Αποτίμηση του συστήματος

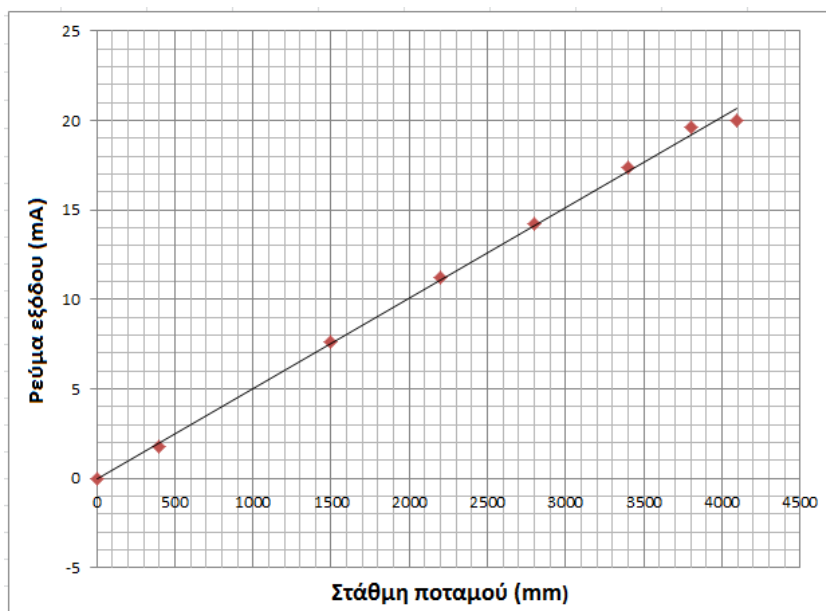
Στο κεφάλαιο αυτό παρατίθενται η διαδικασία και τα αποτελέσματα του ελέγχου της ορθής λειτουργίας των ηλεκτρονικών καρτών που κατασκευάστηκαν, καθώς και η βαθμονόμηση των αισθητήρων οι οποίοι χρησιμοποιήθηκαν στο σύστημα. Τέλος παρουσιάζεται και αναλύεται το περιβάλλον του λογισμικού χρήστη – υπολογιστή.

### 4.1 Έλεγχος κάρτας μετατροπής ψηφιακού σήματος σε πρότυπο 4-20mA

Η συγκεκριμένη κάρτα που έχει ως βασικό στοιχείο τον 12μπιτο μετατροπέα ψηφιακού σήματος σε αναλογικό, DAC902 όπως έχει αναφερθεί χρησιμεύει στην μετατροπή της μετρούμενης τιμής της στάθμης του νερού του ποταμού από τον αιθητήρα υπερήχων, σε αναλογική τιμή ρεύματος από 4mA έως 20mA. Με την βοήθεια ενός προγράμματος που αναπτύχθηκε στον Arduino Mega 1280, ο δεκαδικός αριθμός της τιμής της στάθμης του νερού μετατρέπεται σε δυαδικό αριθμό των 12-bits και αποστέλλεται στον αντίστοιχο ακροδέκτη του DAC902. Για τον έλεγχο λειτουργίας της κάρτας καταγράφηκε το ρεύμα εξόδου από τον DAC902 σε mA για διάφορες τιμές εισόδου της στάθμης σε mm. Τα αποτελέσματα του ελέγχου καταγράφηκαν Πίνακα 7. Στο Σχήμα 59 φαίνεται η γραφική παράσταση μεταξύ του ρεύματος στην έξοδο του DAC και του δεκαδικού αριθμού της στάθμης του ποταμού στην είσοδό του.

Στάθμη ποταμού σε mm	Στάθμη ποταμού σε bits	Ρεύμα εξόδου (mA)
0	000000000000	0
400	000110010000	1,8
1500	010111011100	7,6
2200	100101100000	11,2
2800	101011110000	14,2
3400	110101001000	17,4
3800	111011011000	19,6
4095	111111111111	20

Πίνακας 7. Μετρήσεις για τον έλεγχο της κάρτας μετατροπής ψηφιακού σήματος σε αναλογικό



**Σχήμα 59.** Ρεύμα εξόδου σε mA του μετατροπέα ψηφιακού σήματος σε αναλογικό συναρτήσει της τιμής της στάθμης του ποταμού σε mm

Πρέπει επίσης να σημειωθεί πως ο μετατροπέας ψηφιακού σήματος σε αναλογικό DAC902 έχει ρεύμα εξόδου στην περιοχή 0-20mA. Για να παραχθεί η επιθυμητή έξοδος στην περιοχή 4-20mA έγινε βαθμονόμηση ώστε το εύρος των αριθμών που αποστέλλονται στον DAC902 και αντιπροσωπεύουν την υπολογισθείσα τιμή της στάθμης (0m-6m), να αντιστοιχίζεται εντός του διαστήματος 819 και 4095 bits.

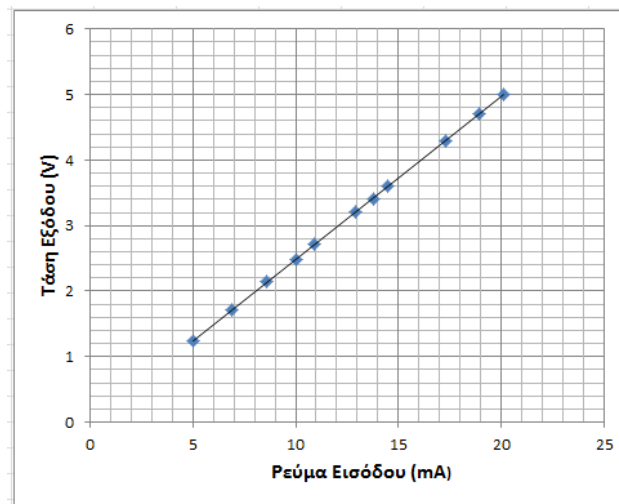
## 4.2 Έλεγχος κάρτας μετατροπής 4-20mA σε 1-5V

Η κάρτα αυτή χρησιμοποιήθηκε με σκοπό να λαμβάνει το αναλογικό σήμα εξόδου 4-20mA της μέτρησης της στάθμης του ποταμού, του υποσυστήματος προσδιορισμού της στάθμης και να το μετατρέπει σε αναλογικό σήμα τάσης 1-5V. Έτσι στη συνέχεια αυτό το σήμα μπορεί να ψηφιοποιείται απο τον ADC του μικροελεγκτή του υποσυστήματος υπολογισμού της στιγμιαίας παροχής. Ο ADC του Arduino mega 2560 έχει ανάλυση 10-bits. Λόγω όμως της ύπαρξης των τελεστικών ενισχυτών στο κύκλωμα της πλακέτας, η κάρτα μετατροπής έχει έξοδο τάσης 1V έως 5V. Αυτό συνεπάγεται μείωση του εύρους κατά  $1024/(5-0) \approx 205$  δηλαδή το εύρος των υποδιαίρέσεων (σταθμών) του ADC θα είναι  $1024-205 \approx 820$  τιμές. Για τον έλεγχο της κάρτας δημιουργήθηκε σε breadboard μια πηγή ρεύματος, και για διάφορες τιμές ρεύματος στην περιοχή 4-20mA, μετρήθηκαν οι αντίστοιχοι αριθμοί καταμέτρησης του ADC, καθώς και η αντίστοιχη τάση εξόδου που φαίνονται στον Πίνακα 8. Στο Σχήμα 60 φαίνεται η γραφική παράσταση της τάσης εξόδου σε συνάρτηση με το ρεύμα εισόδου ενώ στο Σχήμα 61 η γραφική παράσταση αριθμών καταμέτρησης του ADC σε συνάρτηση με το ρεύμα εισόδου.

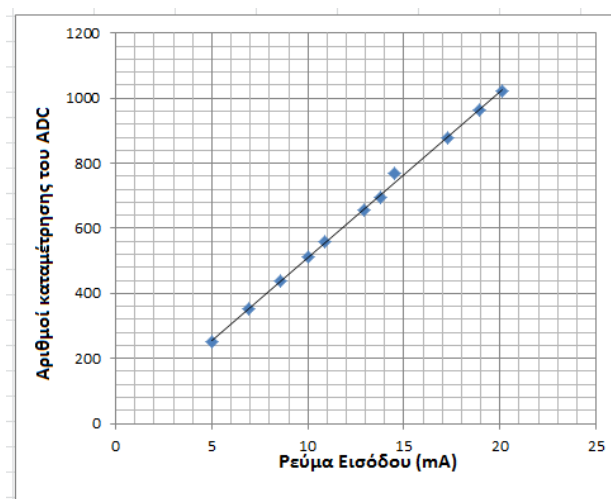


Ρεύμα εισόδου (mA)	Τάση εισόδου (V)	Αριθμοί καταμέτρησης του ADC
5	1,24	253
6,9	1,72	352
8,6	2,15	439
10	2,48	511
10,9	2,72	557
12,9	3,21	656
13,8	3,41	697
14,5	3,61	769
17,3	4,29	877
18,9	4,71	963
20,1	5	1023

Πίνακας 8. Μετρήσεις για τον έλεγχο της κάρτας μετατροπής αναλογικού σήματος 4-20mA σε ψηφιακό στην περιοχή 1-5V



Εικόνα 60. Τάση εξόδου της κάρτας σε συνάρτηση με το ρεύμα εισόδου



Εικόνα 61. Αριθμοί καταμέτρησης του ADC του μικροελεγκτή συναρτήσει του ρεύματος εισόδου στην κάρτα

Οι γραφικές αναπαραστάσεις επιβεβαιώνουν την ορθή λειτουργία της ηλεκτρονικής κάρτας. Για την χάραξη των ευθειών  $y=ax+\beta$  χρησιμοποιήθηκε η μέθοδος των ελαχίστων τετραγώνων [48].

Στην περίπτωση της ευθείας του Σχήματος 75 οι σταθερές της  $y=ax+\beta$  είναι  $\alpha=51.04\pm 0.09$  και  $\beta=0.7\pm 1=0$ .

Έτσι η ευθεία θα είναι της μορφής  $y = 51.04x$ . Άρα η σχέση μεταξύ ρεύματος και αριθμών καταμέτρησης του ADC είναι:

$$C=51.04I \Leftrightarrow I=\frac{C}{51.04}, \text{ σε mA}$$

Όπου :

I : Ρεύμα εισόδου στην κάρτα

C : Αριθμοί καταμέτρησης του ADC

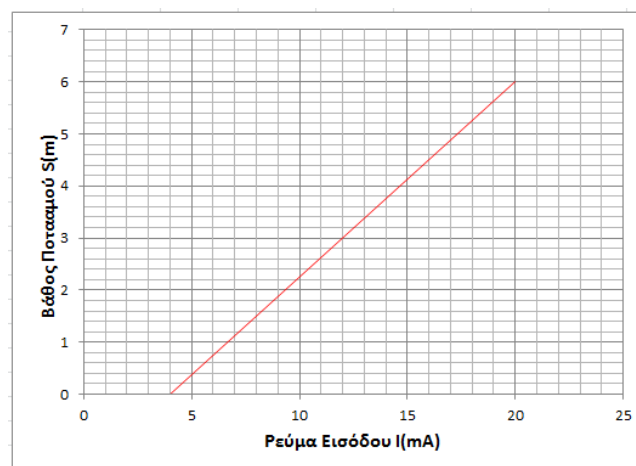
Για το υποσύστημα προσδιορισμού της στάθμης του νερού, για μηδενική στάθμη (0m) το ρεύμα εξόδου από την κάρτα μετατροπής ψηφιακού σήματος σε αναλογικό, είναι 4mA και για τη μέγιστη στάθμη (6m) το ρεύμα εξόδου είναι 20mA. Η σχέση του ρεύματος και της αντίστοιχης φυσικής παραμέτρου είναι γραμμική (Σχήμα 62), έτσι ισχύει η σχέση:

$$S=\alpha'I+\beta'$$

Οπότε απο τις δύο τελευταίες σχέσεις προκύπτει η σχέση ανάμεσα στο βάθος του ποταμού (S) και τις υποδιαίρεσεις C του ADC:

$$S=\alpha\left[\frac{C}{\alpha}\right] + \beta' \Leftrightarrow$$

$$S=\frac{\alpha'}{\alpha}C + \beta'$$



Σχήμα 62. Βάθος του ποταμού συναρτήσει του ρεύματος εξόδου της κάρτας

Για την χάραξη της ευθείας βαθμονόμησης του Σχήματος 76, χρησιμοποιήθηκε η μέθοδος των ελαχίστων τετραγώνων, βάση της οποίας προσδιορίστηκαν οι σταθερές της ευθείας, οι οποίες είναι  $\alpha' = 0.346$  και  $\beta' = 0.692$ .

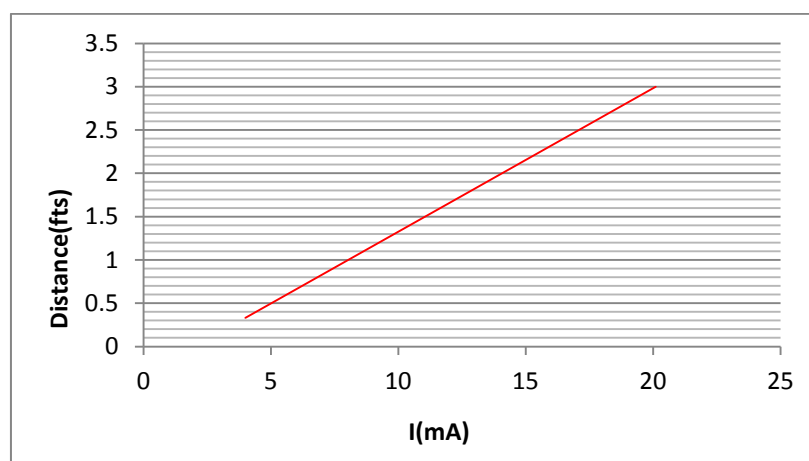
Το σφάλμα στη μέτρηση του βάθους με βάση τη θεωρία διάδοσης σφαιμάτων [49] είναι  $\sigma_s = 0.0075\text{m}$ .

Έπομένως η τελική σχέση του βάθους συναρτήσει των υποδιαίρεσεων του ADC ( $S(C)$ ) είναι:  $S = 0.007 + 0.692 \pm 0.0075 \text{ m}$

Για τον αισθητήρα υπερήχων WL705-012 η βαθμονόμηση από τον κατασκευαστή είναι:

0,33 feet για 4mA  
3 feet για 20,1mA

Βάσει αυτών προκύπτει η ευθεία βαθμονόμησης του Σχήματος 63 που δίνει την απόσταση του αισθητήρα WL705-012 από την στάθμη του ποταμού, συναρτήσει του ρεύματος εξόδου από τον αισθητήρα.



Σχήμα 63. Απόσταση αισθητήρα WL705-012 - στάθμης ποταμού συναρτήσει του ρεύματος εξόδου

Από τη μέθοδο ελαχίστων τετραγώνων οι παράγοντες της σχέσης  $S=f(C)$  για τον αισθητήρα υπερήχων WL705-012 είναι,  $\alpha' = 2.7$  και  $\beta' = -2.4$ .

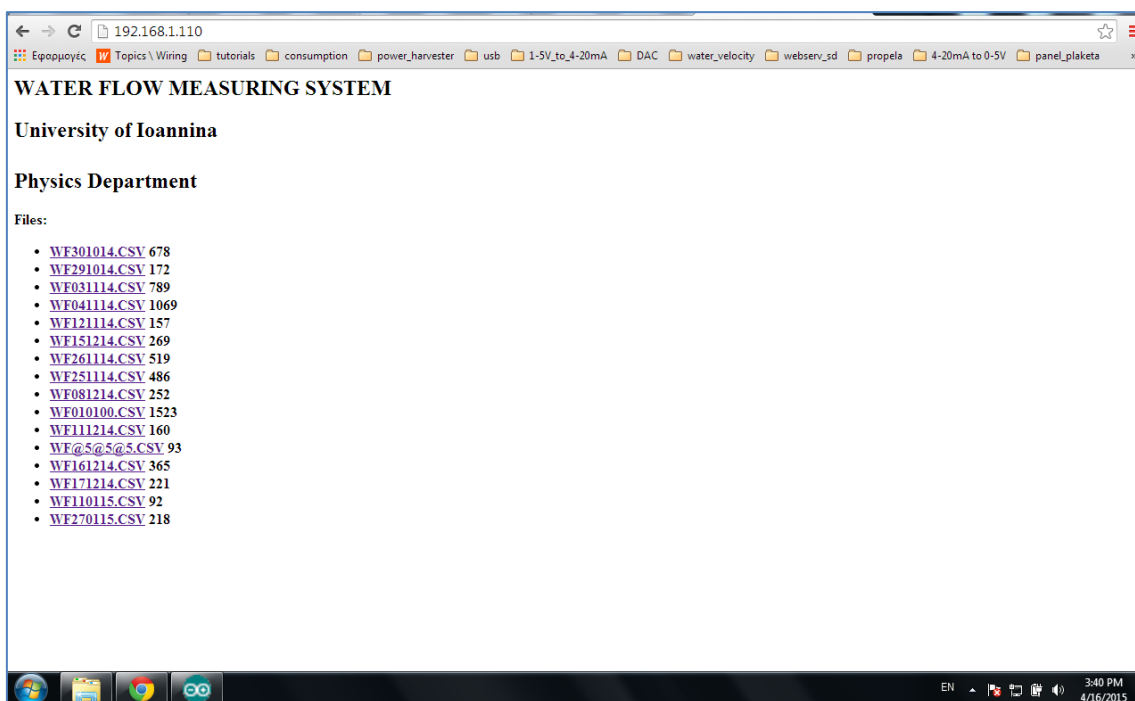
Η  $S=f(C)$  παίρνει τη μορφή:

$$S_{WL705-12} = 0.05C - 2.44$$

Όπου  $S_{WL705-12}$  η απόσταση του αισθητήρα από τον πυθμένα και  $C$  οι αριθμοί καταμέτρησης του ADC.

## 4.3 Διεπαφή του χρήστη

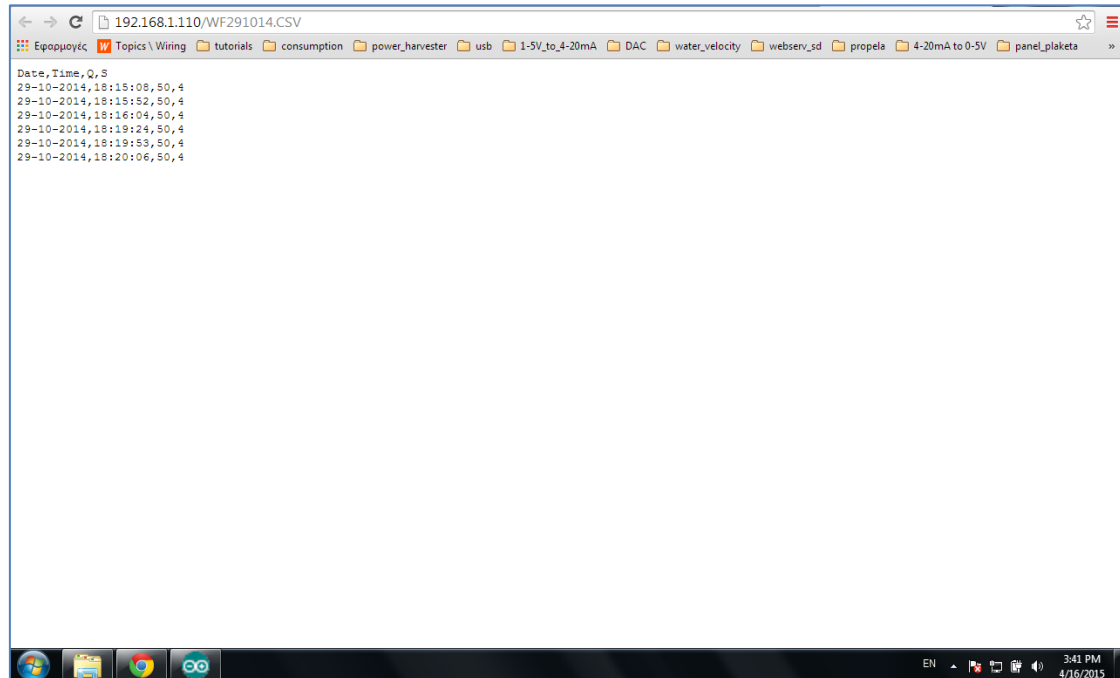
Στην παρούσα διπλωματική εργασία δημιουργήθηκε μια γραφική διεπαφή χρήστη βασισμένη σε έναν διακομιστή παγκοσμίου ιστού ο οποίος χρησιμοποιείται και για διαχείριση των δεδομένων και των αρχείων. Για την επικοινωνία της εφαρμογής που αναπτύχθηκε ένας Web Server που φιλοξενεί μία ιστοσελίδα στην οποία απεικονίζεται η λίστα με τα αρχεία των εγγραφών του υποσυστήματος υπολογισμού της στιγμιαίας παροχής. Διαμέσου της κάρτας Ethernet σε σειριακή επικοινωνία [32] συνδέεται ο ελεγκτής του προτύπου Ethernet, Wiznet WIZ812MJ μέσω SPI με την πλακέτα Arduino Mega 2560 και παρέχει πρόσβαση στο διαδίκτυο. Στο λογισμικό του προγράμματος του υποσυστήματος υπολογισμού της στιγμιαίας παροχής νερού του ποταμού, έχει γραφεί ένα τμήμα σε γλώσσα HTML [50] για την δημιουργία της ιστοσελίδας. Επίσης έχει δηλωθεί η διεύθυνση MAC για τη μονάδα δικτύου WIZ812MJ, η τοπική θύρα για τα εισερχόμενα πακέτα Udp, η διεύθυνση IP του διακομιστή ώρας και η τοπική θύρα για το διακομιστή που αναπτύσσεται στο σύστημα. Έτσι ο Web Server είναι έτοιμος και πληκτρολογώντας σε ένα διακομιστή διαδικτύου, όπως για παράδειγμα το Mozilla Firefox ή το Google Chrome, τη διεύθυνση 192.168.1.110:80, εμφανίζεται η αρχική σελίδα του περιβάλλοντος όπως φαίνεται στο Σχήμα 64.



Σχήμα 64. Η αρχική σελίδα του περιβάλλοντος διεπαφής του χρήστη

Στο παράθυρο που εμφανίζεται με την επιλογή "Files" εμφανίζεται η λίστα με τα αρχεία τύπου .csv, το κάθε ένα ως υπερσύνδεσμος, τα οποία είναι αποθηκευμένα στην κάρτα μνήμης microSD. Στα αρχεία είναι καταγεγραμμένες οι τιμές της παροχής, της στάθμης του νερού, της σχετικής υγρασίας και της θερμοκρασίας του περιβάλλοντος που μετρώνται από το σύστημα. Οι τιμές τους διαχωρίζονται με το ";" και μπορούν να προσπελαστούν από το λογισμικό της Microsoft Office Excel [51] ή από άλλα παρόμοια

προγράμματα. Τα τμήματα του ονόματος κάθε αρχείου αποτελούνται από την ημερομηνία δημιουργίας του και τον τύπο του. Το όνομα είναι της μορφής “WFxxyyzz.csv”, όπου το WF αναφέρεται στο όνομα του συστήματος Water Flow, xx είναι η ημέρα, yy ο μήνας και zz τα δύο τελευταία ψηφία του έτους. Στο Σχήμα 65 προβάλλεται το περιεχόμενο ενός από αυτά τα αρχεία.



Σχήμα 65. Προβολή περιεχομένων ενός αρχείου καταγραφής

## 4.4 Αποτίμηση της μεθόδου υπολογισμού της παροχής

Στη συγκεκριμένη μεταπτυχιακή διπλωματική εργασία χρησιμοποιήθηκε ένας ιδιαίτερος τρόπος υπολογισμού της παροχής του νερού ενός ποταμού που έχει αναφερθεί αναλυτικά στο Κεφάλαιο 3. Προσπαθώντας να αποτιμήσουμε τη διαδικασία υπολογισμού της παροχής κάναμε μια σύγκριση με την μέθοδο 2 σημείων που βασίζεται στη μέτρηση της ταχύτητας σε δύο σημεία, στο 0.2 και στο 0.8 του βάρους και αναφέρεται στην παράγραφο 1.1 σελ. 10. Με βάση τις μετρήσεις ταχύτητας που έγιναν στα τμήματα του ποταμού, εάν χρησιμοποιηθεί η παρούσα μέθοδος η παροχή υπολογίζεται σε 20.8 m<sup>3</sup>/sec ενώ με την μέθοδο των δύο σημείων η παροχή υπολογίζεται σε 20.0 m<sup>3</sup>/sec. Συμπερασματικά μπορούμε να πούμε πως η παρούσα μέθοδος είναι αρκετά ακριβής επειδή χρησιμοποιούνται μετρήσεις της ταχύτητας του νερού από περισσότερα σημεία της τομής του ποταμού, στον υπολογισμό της παροχής του νερού.

Κάθε μία από τις δύο μεθόδους παρουσιάζει ένα στατιστικό σφάλμα λόγω των παραμέτρων που λαμβάνονται υπόψη στον υπολογισμό της παροχής των τμημάτων του ποταμού. Επίσης παρουσιάζει και ένα συστηματικό σφάλμα στον υπολογισμό της παροχής το οποίο οφείλεται στο γεγονός ότι χρησιμοποιείται μια μέση ταχύτητα στα τμήματα του ποταμού η οποία είναι προσεγγιστική.

Οι σχέσεις που χρησιμοποιήθηκαν για τον υπολογισμό των σφαλμάτων σε κάθε μέθοδο παρουσιάζονται παρακάτω:

## Μέθοδος 2 σημείων

*Στατιστικό σφάλμα:*

Αρχικά υπολογίζεται το στατιστικό σφάλμα της ταχύτητας για κάθε τμήμα:

$$\sigma_u = \sqrt{\left(\frac{\partial u}{\partial h}\right)^2 \cdot \sigma_z^2 + \left(\frac{\partial u}{\partial z}\right)^2 \cdot \sigma_h^2}$$

Με αντικατάσταση στην παρακάτω σχέση προκύπτει το στατιστικό σφάλμα στον υπολογισμό της παροχής:  $\Sigma \sigma_p = \sqrt{(s^2 \cdot \sigma_u^2 + u^2 \cdot \sigma_s^2)}$

*Συστηματικό σφάλμα:*

$$\sigma_p = \sqrt{(P_{1\_μεθοδος\_ανα\_0.5m} - P'_{1\_μεθοδος\_δύο\_σημειων}) + \dots + (P_{30\_μεθοδος\_ανα\_0.5m} - P'_{30\_μεθοδος\_δύο\_σημειων})}$$

Όπου P η παροχή της στήλης με χρήση ταχυτήτων ανα 0.5m και P' η παροχή της στήλης με χρήση ταχυτήτων από τη μέθοδο 2 σημείων.

## Σύστημα υπολογισμού παροχής

*Στατιστικό σφάλμα:*

Το στατιστικό σφάλμα προκύπτει από την ίδια σχέση αλλά για περισσότερα σημεία (ανά

$$0.5m): \sigma_u = \sqrt{\left(\frac{\partial u}{\partial h}\right)^2 \cdot \sigma_z^2 + \left(\frac{\partial u}{\partial z}\right)^2 \cdot \sigma_h^2}$$

Με αντικατάσταση στην παρακάτω σχέση προκύπτει το στατιστικό σφάλμα στον υπολογισμό της παροχής:  $\Sigma \sigma_p = \sqrt{(s^2 \cdot \sigma_u^2 + u^2 \cdot \sigma_s^2)}$

*Συστηματικό σφάλμα:*

Το συστηματικό σφάλμα στον υπολογισμό της ταχύτητας προκύπτει από την παραδοχή ότι αυτή είναι ίδια σε όλο το τμήμα 50x50cm. Ωστόσο αυτή διαφέρει και έτσι μπορεί να υπολογιστεί το συστηματικό σφάλμα από τις παρακάτω σχέσεις:

$$\sigma_{u_{\sigma\sigma}} = |u_o' - u_o|, \text{ όπου } u_o' = \frac{4u_0 + u'_1 + u'_2 + u'_3 + u'_4}{8} \text{ με } u_0 \text{ την ταχύτητα στο κέντρο}$$

του τμήματος και  $u'_1, u'_2, u'_3, u'_4$  οι γειτονικές ταχύτητες.

Με αντικατάσταση στην παρακάτω σχέση προκύπτει το συστηματικό σφάλμα στον υπολογισμό της παροχής:  $\Sigma \sigma_p = \sqrt{(s^2 \cdot \sigma_u^2 + u^2 \cdot \sigma_s^2)}$

Επομένως η τιμή της παροχής για κάθε μέθοδο παίρνει την εξής τελική μορφή:

$$\text{Μέθοδος 2 σημείων: } 20.0 \pm 1.4 \pm 2.8 \frac{\text{m}^3}{\text{sec}}$$

$$\text{Σύστημα υπολογισμού παροχής: } 20.8 \pm 1.2 \pm 2.5 \frac{\text{m}^3}{\text{sec}}$$





## 5. Συμπεράσματα

Στην παρούσα μεταπτυχιακή διπλωματική εργασία αναπτύχθηκε και παρουσιάζεται ένα σύστημα μέτρησης της στιγμιαίας παροχής του νερού ενός ποταμού. Το σύστημα που αναπτύχθηκε στο εργαστήριο Φυσικής Υψηλών Ενεργειών είναι χωρισμένο σε δύο επιμέρους υποσυστήματα. Αρχικά με ένα σύστημα μέτρησης της ταχύτητας του νερού μετράται η τρέχουσα ταχύτητα του νερού σε διάφορα σημεία μίας κάθετης τομής της κοίτης του ποταμού, έτσι ώστε να προσδιοριστούν οι σταθερές της σχέσης  $u=f(z)$  που χρησιμοποιούνται στον υπολογισμό της παροχής. Με το πρώτο υποσύστημα μπορεί να μετράται περιοδικά η στάθμη του νερού του ποταμού και με το άλλο υποσύστημα να υπολογίζεται η στιγμιαία παροχή του νερού. Επιπλέον οι τελευταίες τιμές αποθηκεύονται σε μια κάρτα μνήμης και εμφανίζονται σε έναν διακομιστή διαδικτύου. Το όλο σύστημα τροφοδοτείται με τάση 12V, από ένα σύστημα που περιλαμβάνει ηλιακά πάνελς και υπερπυκνωτές. Ο υπολογισμός της κατανάλωσης ισχύος του κάθε υποσυστήματος και των αισθητήρων που χρησιμοποιήθηκαν παρατίθεται στο Παράρτημα 4.

Σκοπός της εργασίας ήταν ο υπολογισμός της στιγμιαίας παροχής με μεγάλη ακρίβεια. Ο τρόπος υπολογισμού της παροχής του νερού από το παρόν σύστημα, είναι ακριβέστερος από τον υπολογισμό με άλλες μεθόδους που χρησιμοποιούνται.

Στο σύστημα αυτό μπορούν να γίνουν διάφορες επεκτάσεις σε επίπεδο υλικού αλλά και λογισμικού. Όσον αφορά το κομμάτι του λογισμικού, θα μπορούσε η διαδικασία μέτρησης της ταχύτητας του νερού των τμημάτων του ποταμού, που εξηγήθηκε αναλυτικά στο Κεφάλαιο 2, να επαναληφθεί για διαφορετικές τιμές της στάθμης του νερού. Αυτό θα είχε ως αποτέλεσμα τον πιο ακριβή προσδιορισμό του παράγοντα  $\rho g \theta / 2 \eta$  για κάθε στήλη νερού και κατά συνέπεια ακριβέστερο υπολογισμό της παροχής. Στο υλικό κομμάτι θα μπορούσε να χρησιμοποιηθεί κάποιος αισθητήρας υπερήχων (όπως αναφέρθηκε ο WL705-12), ο οποίος θα έχει έξοδο 4-20mA και μεγαλύτερο εύρος απόστασης εντοπισμού, έτσι δέν θα ήταν απαραίτητη η κατασκευή του πρώτου υποσυστήματος. Μία άλλη αναβάθμιση η οποία θα μπορούσε να γίνει στο σύστημα, είναι στο κομμάτι της μεταφοράς του σήματος της στάθμης από το ένα υποσύστημα στο άλλο. Η αποστολή της τιμής της στάθμης του ποταμού από το υποσύστημα προσδιορισμού της στάθμης του νερού στο υποσύστημα υπολογισμού της στιγμιαίας παροχής θα μπορούσε να γίνει ασύρματα. Ωστόσο αυτές οι λύσεις ανεβάζουν το κόστος και την πολυπλοκότητα της κατασκευής.



# Αναφορές

[1] Βικιπαιδεία, Νερό.

<https://el.wikipedia.org/wiki/%CE%9D%CE%B5%CF%81%CF%8C>

[2] Χάρτες google.

<https://www.google.gr/maps>

[3] Βαρβάρα Μοντεσάντου, Σημειώσεις Λιμνολογίας, Πανεπιστήμιο Αθηνών τμήμα Βιολογίας, Τομέας οικολογίας και ταξινομικής, Αθήνα 1999.

[4] Michael Fowler, Calculating Viscous Flow: Velocity Profiles in Rivers and Pipes, 2012

[5] Δ. Παναγούλια, Γ Δήμου, Σημειώσεις μηχανικής φερτών υλικών 1, Εθνικό Μετσόβιο Πολυτεχνείο, Διατμηματικό, Διεπιστημονικό πρόγραμμα μεταπτυχιακών σπουδών, Επιστήμη και τεχνολογία υδατικών πόρων, Αθήνα, 29/10/2010, σελ. 3.

[6] Thomas J. Buchanan and William P. Somers, Techniques of Water-Resources Investigations of the United States Geological Survey, chapter A8 DISCHARGE MEASUREMENTS AT GAGING STATIONS, WASHINGTON, 1969.

[7] U.S. Geological Survey.

<https://www.usgs.gov/>

[8] Wikipedia, Data acquisition.

[https://en.wikipedia.org/wiki/Data\\_acquisition](https://en.wikipedia.org/wiki/Data_acquisition)

[9] Αντώνιος Γαστεράτος, Μουρούτσος Σπυρίδων, Ιωάννης Ανδρεάδης, Τεχνική των μετρήσεων, Ξάνθη, 2006.

[10] Βικιπαιδεία, Αισθητήρας.

<https://el.wikipedia.org/wiki/%CE%91%CE%B9%CF%83%CE%B8%CE%B7%CF%84%CE%AE%CF%81%CE%B1%CF%82>

[11] Μικροεπεξεργαστές ,θεωρία και εφαρμογές Gilmore , Εκδόσεις Τζιόλα.

[12] Wikipedia, Μικροεπεξεργαστής.

<http://el.wikipedia.org/wiki/Μικροεπεξεργαστής>

[13] Arduino, Arduino Mega.

<http://www.arduino.cc/>

[14] Wikipedia, Arduino.

<https://en.wikipedia.org/wiki/Arduino>

[15] Wiring.

<http://wiring.org.co/>

[16] Roger L. Traylor, Bootloaders, Oregon State University, Αμερική, Oregon.

[17] Arduino, Arduino shields.

<http://www.arduino.cc/en/Main/ArduinoShields>

[18] Μούζης Νικόλαος, Μετάδοση Δεδομένων μέσω του Πρωτοκόλλου I2C, Αθήνα, Εθνικό Μετσόβιο Πολυτεχνείο, Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Τομέας Επικοινωνιών Ηλεκτρονικής και Συστημάτων Πληροφορικής, 2005.

[19] Oak language specification, Java programming language, First Person Inc. (sun microsystems) 1994

[20] Σταμάτης Σταματιάδης, Εισαγωγή στη γλώσσα προγραμματισμού C++ [σημειώσεις διαλέξεων], Κρήτη, Πανεπιστήμιο Κρήτης, Τμήμα Επιστήμης και Τεχνολογίας Υλικών.

[21] FTDI, UM232R USB - Development module, Future Technology Devices International.  
[http://www.ftdichip.com/Support/Documents/DataSheets/Modules/DS\\_UM232R.pdf](http://www.ftdichip.com/Support/Documents/DataSheets/Modules/DS_UM232R.pdf)

[22] Motorola Inc., SPI Block Guide V03.06, 2000, rev 2003

[23] Arduino, Arduino software.

<http://www.arduino.cc/en/main/Software>

[24] Atmel, Atmega1280/2560.

<http://www.atmel.com/Images/doc2549.pdf>

[25] Hirak Patangia, Sri Nikhil Gupta Gourisetti, A Novel Strategy for Selective Harmonic Elimination Based on a Sine-Sine PWM Model, MWSCAS, U.S.A, 2012

[26] Μπόχλος Αθανάσιος, Ανάπτυξη ενεργειακά αυτόνομου ασύρματου συστήματος αναμετάδοσης ψηφιακής πληροφορίας, Ιωάννινα, Πανεπιστήμιο Ιωαννίνων, Τμήμα Φυσικής, Εργαστήριο Φυσικής Υψηλών Ενεργειών, 2013.

[27] Μάνθος Νικόλαος, Wireless Water Quality Monitoring of Arachthos and Vjosë Rivers (WWQM\_AVR) [presentation notes], Greece - Albania Cross Border Programme 2007-2013, WWQM\_AVR kick-off meeting No1, 2012.

<http://www.greece-albania.eu/projects.html?view=item&id=14>

<http://www.interreg.gr/en/managing-authority/2007-2013.html>

[28] Vernier, Flow rate sensor.

<http://www.vernier.com/products/sensors/flo-bta/>

[29] Wikipedia, British telecom sockets.

[http://en.wikipedia.org/wiki/British\\_telephone\\_sockets](http://en.wikipedia.org/wiki/British_telephone_sockets)

[30] Global water.

<http://www.globalw.com/>

[31] Via Natura

<http://www.vianatura.gr/>

[31] Wikipedia, TTL.

[https://en.wikipedia.org/wiki/Transistor%E2%80%93transistor\\_logic](https://en.wikipedia.org/wiki/Transistor%E2%80%93transistor_logic)

[32] Αυδίκου Χριστιάνα, Ανάπτυξη συστήματος μέτρησης της ποιότητας νερού με απομακρυσμένη διαχείριση μέσω παγκοσμίου ιστού, Τμήμα Φυσικής, Πανεπιστήμιο Ιωαννίνων, Μάρτιος 2013.

[33] Atmel, Atmel Studio.

<http://www.atmel.com/tools/atmelstudio.aspx>

[34] Robot Electronics, SRF05 - Ultra-Sonic Ranger, Technical Specification.

<https://www.robot-electronics.co.uk/htm/srf05tech.htm>

[35] Global water, WL705 Ultrasonic Water Level Sensor.

<http://www.globalw.com/downloads/WL700/WL705.pdf>

[36] Spark Fun Electronics, Aosong Electronics Co.,Ltd, Digital-output relative humidity & temperature sensor/module DHT22 (DHT22 also named as AM2302)

<https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>

[37] Global water, FP111 flow probe.

<http://www.globalw.com/products/flowprobe.html>

[38] Μάκης Ματιάτος, Το πηδάλιο του σκάφους και πως λειτουργεί.

[39] Kalen Brunham, Introduction to OrCAD, Hands on workshop, 2002.

[40] Kester Walt, The Data Conversion Handbook, 2005.

[41] Texas Instruments, DAC902.

<http://www.ti.com/product/dac902>

[42] Texas Instruments, LM3478.

<http://www.ti.com/product/lm3478>

[43] Παγώνης Αθανάσιος, Σχεδιασμός και ανάπτυξη διεπαφών ATM/PDH και CPU/αισθητήρα, Ιωάννινα, Πανεπιστήμιο Ιωαννίνων, Τμήμα Φυσικής, Εργαστήριο Φυσικής Υψηλών Ενεργειών, Οκτώβριος 1999.

[44] alldatasheet.com, LM324AM.

<http://www.alldatasheet.com/datasheet-pdf/pdf/53576/FAIRCHILD/LM324AM.html>

[45] Ucamco, The Gerber Format Specification.

[http://www.ucamco.com/files/downloads/file/81/the\\_gerber\\_file\\_format\\_specification.pdf](http://www.ucamco.com/files/downloads/file/81/the_gerber_file_format_specification.pdf)

[46] Wikipedia, IP Code.

[https://en.wikipedia.org/wiki/IP\\_Code](https://en.wikipedia.org/wiki/IP_Code)

[47] Viscopedia, A free encyclopedia for viscosity, Water.

<http://www.viscopedia.com/viscosity-tables/substances/water/>

[48] Βικιπαίδεια, Απλή γραμμική παλινδρόμηση, Μέθοδος ελαχίστων τετραγώνων.

[https://el.wikipedia.org/wiki/%CE%91%CF%80%CE%BB%CE%AE\\_%CE%B3%CF%81%CE%B1%CE%BC%CE%BC%CE%B9%CE%BA%CE%AE\\_%CF%80%CE%B1%CE%BB%CE%B9%CE%BD%CE%B4%CF%81%CF%8C%CE%BC%CE%B7%CF%83%CE%B7](https://el.wikipedia.org/wiki/%CE%91%CF%80%CE%BB%CE%AE_%CE%B3%CF%81%CE%B1%CE%BC%CE%BC%CE%B9%CE%BA%CE%AE_%CF%80%CE%B1%CE%BB%CE%B9%CE%BD%CE%B4%CF%81%CF%8C%CE%BC%CE%B7%CF%83%CE%B7)

[49] Πανεπιστήμιο Αθηνών, Τμήμα Φυσικής, Εργαστήριο Φυσικής, Εισαγωγή στη θεωρία σφαλμάτων, Αθήνα 2014.

[50] Elizabeth Castro, HTML for the World Wide Web, Fifth Edition with XHTML and CSS, 2003.

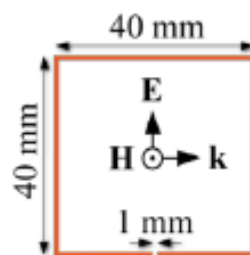
[51] Saylor Academy, How to Use Microsoft Excel.

<http://www.saylor.org/site/textbooks/How%20to%20Use%20Microsoft%20Excel.pdf>

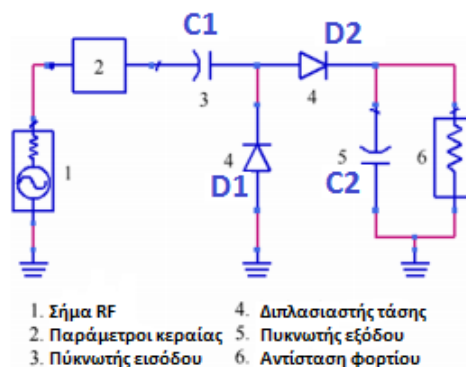
## Εναλλακτική πρόταση συστήματος τροφοδοσίας βασισμένο σε αξιοποίηση των διάχυτων ηλεκτρομαγνητικών κυμάτων του περιβάλλοντος

Στα πλαίσια της παρούσας εργασίας έγινε προσπάθεια χρήσης διάχυτης ηλεκτρομαγνητικής ακτινοβολίας για παραγωγή ενέργειας. Αναπτύχθηκε και δοκιμάστηκε ένα σύστημα φόρτισης, αποτελούμενο από μια κεραία (microwave power harvester) που περιγράφεται στην αναφορά [1], η οποία λαμβάνει την ενέργεια από τα μικροκύματα που χρησιμοποιούνται στις τηλεπικοινωνίες.

Η κεραία που αναπτύχθηκε μετατρέπει την διάχυτη ηλεκτρομαγνητική ενέργεια σε ηλεκτρικό σήμα. Το σχέδιο και το ηλεκτρονικό κύκλωμα της κεραίας που κατασκευάστηκε στο εργαστήριο φαίνονται αντιστοίχα στα Σχήματα 66 και 67.



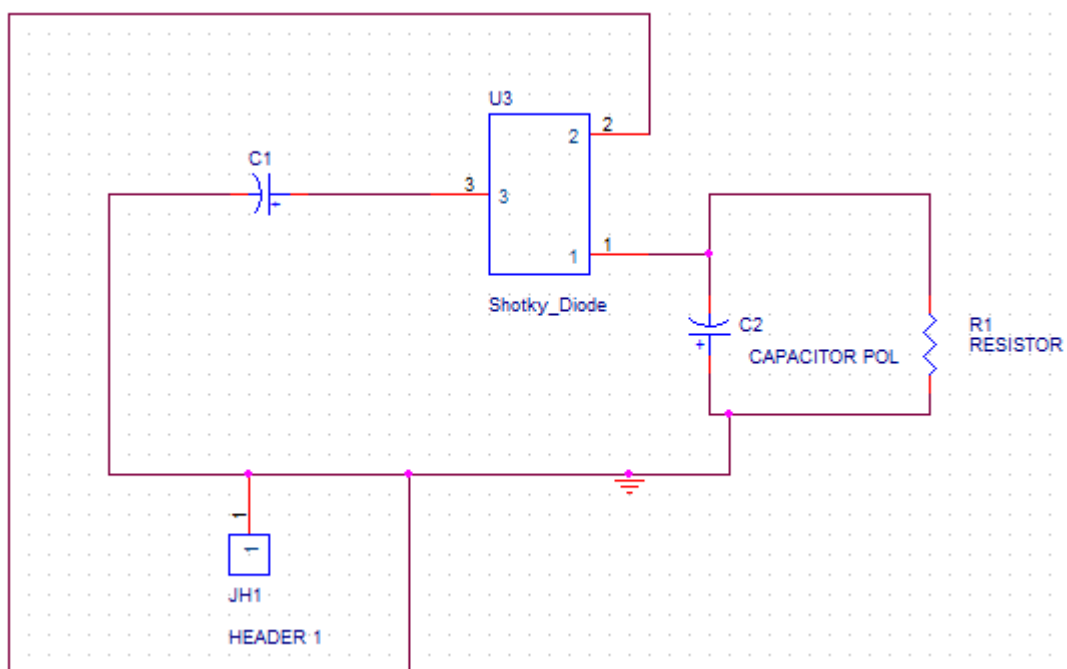
Σχήμα 66. Σχέδιο κεραίας συλλογής ενέργειας μικροκυμάτων [1]



Σχήμα 67. Ηλεκτρονικό κύκλωμα μετατροπής της ηλεκτρομαγνητικής ενέργειας σε ηλεκτρικό σήμα [1]

Στο Σχήμα 66 απεικονίζεται ο χάλκινος διάδρομος πάχους 2mm, και με τα σύμβολα E, H και k, το ηλεκτρικό πεδίο, το μαγνητικό πεδίο και το κυματοδιάνυσμα (έχει τη διεύθυνση διάδοσης του ηλεκτρομαγνητικού κύματος) αντίστοιχα. Οι διαστάσεις αυτές κάνουν τη συσκευή να λειτουργεί σαν κεραία χρησιμοποιώντας ένα αντηχείο διαχωρισμένου δακτυλίου (split ring resonator) [2], στο οποίο έχουμε συντονισμό στα

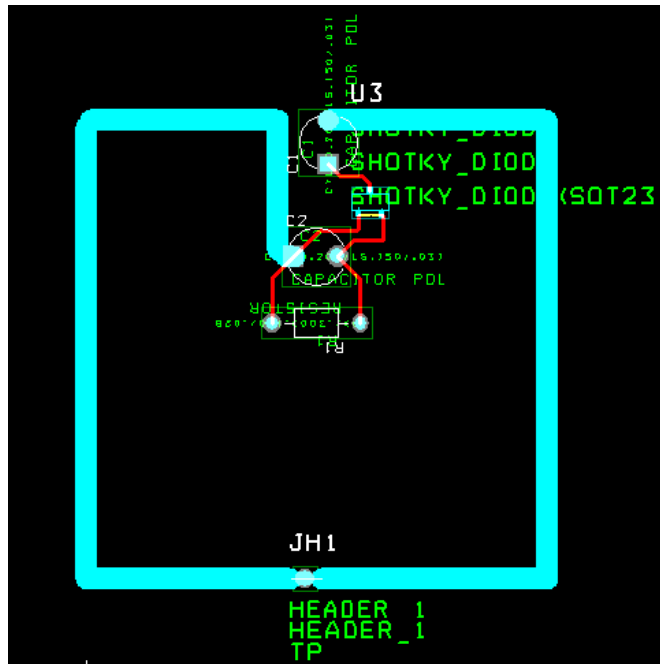
900MHz (0.33m μήκος κύματος). Με βάση τα παραπάνω σχεδιάστηκε το κύκλωμα το οποίο περιλαμβάνει την κεραία με τις κατάλληλες παραμέτρους και το κύκλωμα μετατροπής της ηλεκτρομαγνητικής ενέργειας σε ηλεκτρικό σήμα (Σχήμα 67). Το ηλεκτρονικό κύκλωμα λειτουργεί ως εξής. Αρχικά ο πυκνωτής C1 φορτίζεται από την εναλλασσόμενη τάση εισόδου και η διόδος D1 πολώνεται ανάστροφα με αποτέλεσμα να μην μπορεί ο πυκνωτής C1 να εκφορτισθεί. Έτσι ο πυκνωτής C1 αποκτά σταθερή τάση στα άκρα του ίση με την κορυφή της εναλλασσόμενης τάσης εισόδου και αυτή μεταφέρεται στην αντίσταση φορτίου. Όταν η εναλλασσόμενη τάση εισόδου αλλάξει πολικότητα, ενεργοποιείται ο πυκνωτής C2 που βλέπει δύο πηγές τάσης. Η μία είναι η εναλλασσόμενη τάση εισόδου και η άλλη η τάση του πυκνωτή C1. Αυτές οι δύο τάσεις είναι σε σειρά μεταξύ τους και έτσι ο πυκνωτής C2 φορτίζεται μέσω της διόδου D2 από τη συνολική τάση. Επομένως ο C2 φορτίζεται με το διπλάσιο της τιμής της τάσης εισόδου, την οποία και μεταφέρει στην αντίσταση φορτίου R1. Στο Σχήμα 68 παρουσιάζεται το σχηματικό διάγραμμα του κυκλώματος της κεραίας συλλογής ενέργειας μικροκυμάτων που σχεδιάστηκε με βάση την αναφορά [1]



Σχήμα 68. Σχηματικό διάγραμμα κεραίας συλλογής ενέργειας μικροκυμάτων

Στην κατασκευή του κυκλώματος χρησιμοποιήθηκε ένας πυκνωτής 100pF C1 στην είσοδο του. Επίσης χρησιμοποιήθηκαν δύο διόδους Schottky οι οποίες εμπεριέχονται στο ολοκληρωμένο κύκλωμα HSMS-2852 U3 [3]. Η επιλογή διόδων είναι πολύ σημαντική για την επίδοση της μονάδας. Η διόδος Schottky, προσφέρει χαμηλή τάση αγωγής (0.15-0.45V αντί μιας κανονικής διόδου που προσφέρει 0.6-1.7V) και πολύ μικρό χρόνο επανόρθωσης, της τάξης των nanosecond. Τέλος χρησιμοποιήθηκε ένας πυκνωτής 100uF C2 στην έξοδο, παράλληλα με την αντίσταση φορτίου R1. Η βέλτιστη τιμή της αντίστασης φορτίου σύμφωνα με τις προσομοιώσεις είναι 70Ω. Το τυπωμένο σχέδιο της κεραίας, η οποία είναι δύο επιπέδων, φαίνεται στο Σχήμα 69.



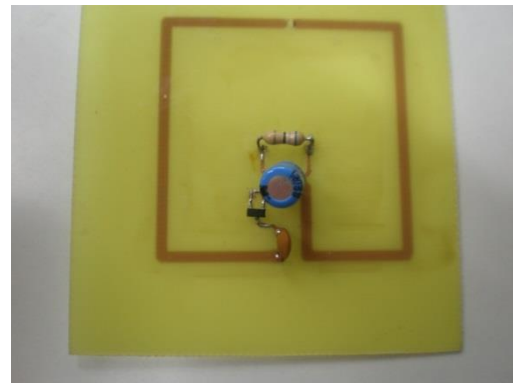


Σχήμα 69. Σχέδιο κεραίας συλλογής ενέργειας μικροκυμάτων (microwave power harvester) στο Orcad Layout

Οι δύο όψεις της κεραίας με το κύκλωμα που κατασκευάστηκε φαίνονται στα Σχήματα 70 και 71.



Σχήμα 70. Επάνω όψη κάρτας microwave power harvester

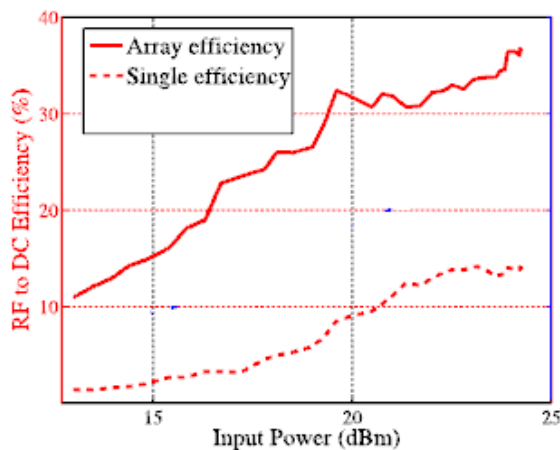


Σχήμα 71. Κάτω όψη κάρτας microwave power harvester

Ένας δείκτης της αποδοτικότητας της κεραίας είναι η απόδοση μετατροπής RF σε DC.

$$n = \frac{P_{DC}}{P_{RF}}$$

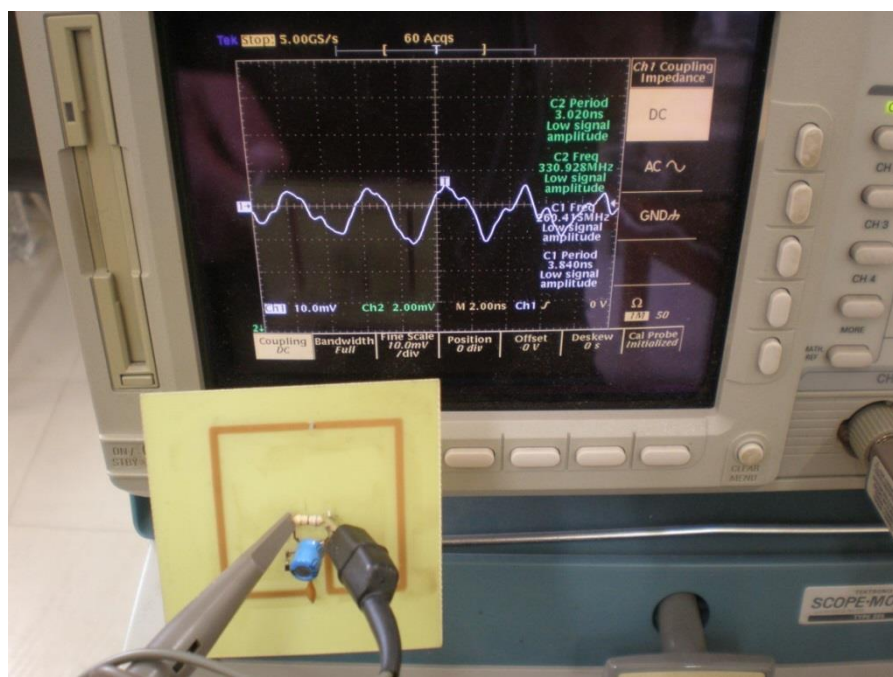
Όπου  $P_{RF}$  η ισχύς του σήματος που προσπίπτει στη κεραία και  $P_{DC}$  η εξερχόμενη ηλεκτρική ισχύς. Η μέγιστη τυπική απόδοση για μία κεραία είναι 14.2%, ενώ για μια συστοιχία από πέντε κεραίες 36.8% όπως φαίνεται στο γράφημα του Σχήματος 72.



Σχήμα 72. Γράφημα απόδοσης RF σε DC (%) για μια κεραία και για συστοιχία πέντε κεραίων [1]

Στο εργαστήριο Φυσικής Υψηλών Ενεργειών μετρήθηκε η εξερχόμενη τάση της κεραίας με παλμογράφο (Σχήμα 73) και βρέθηκε ότι είναι ίση με 25mV (peak to peak). Η ενεργός τιμή αυτής της εναλλασσόμενης τάσης θα είναι:

$$V_{rms} = \frac{V_p}{\sqrt{2}} = \frac{12.5}{\sqrt{2}} = 8.83mV$$



Σχήμα 73. Μετρήσεις απόδοσης κεραίας συλλογής ενέργειας μικροκυμάτων με παλμογράφο

Στη συνέχεια για τον έλεγχο λειτουργίας της κεραίας, πραγματοποιήθηκε μία τηλεφωνική κλήση απο κινητό τηλέφωνο, έτσι ώστε να αυξηθεί η ισχύς της ηλεκτρομαγνητικής ακτινοβολίας. Τα κινητά τηλέφωνα εκπέμπουν ηλεκτρομαγνητική ακτινοβολία μόνο κατά την διάρκεια της τηλεφωνικής επικοινωνίας. Συνήθως η τυπική ισχύς εκπομπής κινητών τηλεφώνων κυμαίνεται απο 15 έως 500mW. Η νέα εξερχόμενη

τάση της κεραίας με το κινητό σε λειτουργία είναι 85mV (peak to peak) όπως φαίνεται στο Σχήμα 74.



Σχήμα 74. Μετρήσεις απόδοσης κεραίας συλλογής ενέργειας μικροκυμάτων με τηλεφωνική κλήση

Η νέα ενεργός τιμή της εναλλασσόμενης τάσης είναι  $V_{rms} = \frac{V_p}{\sqrt{2}} = \frac{42.5}{\sqrt{2}} = 30.01mV$ .

Όπως αναμενόταν παρατηρείται σημαντική αύξηση καθώς η νέα ενεργός τιμή της εναλλασσόμενης τάσης είναι κατά 70% μεγαλύτερη.

Συμπέρασμα: Παρατηρείται πως η προσπάθεια που έγινε για τη συλλογή ενέργειας από τα ηλεκτρομαγνητικά κύματα που υπάρχουν στο περιβάλλον μέσω της κεραίας που κατασκευάστηκε, μπορεί να παρέχει μία τάση εξόδου από 8 έως 30mV. Ωστόσο με μία συστοιχία από περισσότερες κεραίες η τιμή αυτή μπορεί να αυξηθεί σημαντικά.

# Αναφορές Παραρτήματος 1

[1] Allen M. Hawkes, Alexander R. Katko, and Steven A. Cummer, A microwave metamaterial with integrated power harvesting functionality.

[2] Wikipedia, Split-ring resonator.  
[https://en.wikipedia.org/wiki/Split-ring\\_resonator](https://en.wikipedia.org/wiki/Split-ring_resonator)

[3] alldatasheet.com , HSMS-2852.  
<http://pdf1.alldatasheet.com/datasheet-pdf/view/103040/HP/HSMS-2852.html>

### Λογισμικό συστημάτων

#### Π2.1 Κώδικας υποσυστήματος υπολογισμού της στιγμιαίας παροχής νερού του ποταμού

Το συγκεκριμένο πρόγραμμα αναπτύχθηκε και φορτώθηκε στο υποσύστημα υπολογισμού της στιγμιαίας παροχής. Σκοπός του είναι να λαμβάνει την τιμή της στάθμης από το υποσύστημα προσδιορισμού της στάθμης και να υπολογίζει τη στιγμιαία παροχή του νερού του ποταμού. Αρχικά στο πρόγραμμα γίνεται η δήλωση των καθολικών (global) μεταβλητών καθώς επίσης και η διεύθυνση MAC για τη μονάδα δικτύου WIZ812MJ, η τοπική θύρα για τα εισερχόμενα πακέτα Udp, η διεύθυνση IP του διακομιστή ώρας και η τοπική θύρα για το διακομιστή που αναπτύσσεται στο σύστημα. Έπειτα δηλώνονται και αναπτύσσονται κάποιες συναρτήσεις οι οποίες καλούνται στο κυριώς πρόγραμμα. Αυτές είναι οι εξής:

- void error P(const char\* str)

Είναι συνάρτηση σφάλματος. Τυπώνει σφάλματα και σε περίπτωση που παρουσιαστούν προβλήματα διακόπτει το πρόγραμμα. Επίσης αποθηκεύει συμβολοσειρές (strings) στη μνήμη flash για εξοικονόμηση μνήμης RAM.

- ostream& operator

Μορφοποιεί την ώρα και την ημερομηνία του ρολογιού.

- void PrintDateTime(DateTime t)

Χρησιμοποιείται για να τυπωθεί η ώρα του ρολογιού.

- unsigned long sendNTPpacket(IPAddress& address)

Επειδή το ρολόι πραγματικού χρόνου (RTC) δεν είναι ακριβές, χάνει δηλαδή κάποια δευτερόλεπτα την ημέρα, για τη ρύθμισή του χρησιμοποιείται το πρωτόκολλο Udp για το συγχρονισμό του με έναν διακομιστή ώρας (NTP server - Network Time Protocol). Η

συνάρτηση `sendNTPpacket()` χρησιμοποιείται για την αποστολή αίτησης στο διακομιστή NTP.

- `void ListFiles(EthernetClient client, uint8 t flags)`

Η συνάρτηση αυτή τυπώνει τη λίστα με τα αρχεία της κάρτας microSD στον πελάτη (Client) του διακομιστή σε μορφή html.

Στη συνέχεια του προγράμματος ακολουθεί η ρουτίνα `setup()` όπου δηλώνονται οι λειτουργίες που εκτελούνται στην αρχή εκτέλεσης του κώδικα. Αρχικά ορίζεται ο ρυθμός μετάδοσης των δεδομένων (`Serial.begin(9600)`) για την προβολή των εξόδων του προγράμματος στη σειριακή θύρα του περιβάλλοντος ανάπτυξης IDE του Arduino. Σε αυτό το σημείο γίνεται ο ορισμός των τιμών των παραγόντων  $a, b$  των ευθειών  $\log z = f(u)$  για κάθε στήλη νερού που παρουσιάζονται στο Παράρτημα 5. Επίσης ορίζεται το εμβαδόν  $A_i$  του κάθε υποτιμήματος. Έπειτα γίνεται η ανάγνωση της στάθμης  $S$  απο το πρώτο υποσύστημα και υπολογίζεται η στάθμη κάθε στήλης αλλά και η ταχύτητα  $U_i$  του νερού για κάθε ένα από τα τμήματα των στηλών. Ακολουθεί ο υπολογισμός των επιμέρους παροχών των τμημάτων και η εύρεση της συνολικής παροχής κάθε στήλης  $Q_{itot}$ . Στη συνέχεια αθροίζονται οι παροχές των στηλών και υπολογίζεται η συνολική παροχή  $Q_{tot}$ . Αφού ολοκληρωθεί η διαδικασία υπολογισμού, γίνεται η αρχικοποίηση της κάρτας microSD και προβάλλεται η λίστα με τα αρχεία που υπάρχουν σε αυτή. Στη συνέχεια αρχικοποιείται το ρολόι πραγματικού χρόνου (RTC). Εφόσον υπάρχει σύνδεση στο δίκτυο, το ρολόι λαμβάνει την ώρα από το διακομιστή ώρας (NTP), ειδάλλως συνεχίζεται η εκτέλεση του προγράμματος. Ύστερα ελέγχεται εαν υπάρχει το αρχείο της ημέρας στην κάρτα μνήμης. Εάν όχι, τότε αυτό δημιουργείται και εγγράφεται σε αυτό η παροχή (Q), η στάθμη (S), η σχετική υγρασία (H) και η θερμοκρασία (T). Στη συνέχεια, ακολουθεί η ρουτίνα `loop()`, μέσω της οποίας εμφανίζεται η ιστοσελίδα (σε html) που εμπεριέχει τη λίστα με τα αρχεία της κάρτας μνήμης microSD και άλλες πληροφορίες του συστήματος. Έτσι εφόσον σταλεί μια αίτηση στο διακομιστή από κάποιο πρόγραμμα περιήγησης, τότε εμφανίζονται τα αρχεία της κάρτας μνήμης και εφόσον επιλεγεί κάποιο απο αυτά να εμφανίζονται τα περιεχομενά του.

```
#include <SdFat.h>
```

```
#include <SdFatUtil.h>
```

```
#include <Ethernet.h>
```

```
#include <SPI.h>
```

```
#include <EthernetUdp.h>
```

```
// libraries for realtime clock
```

```
#include <Wire.h>    /// I2C library
```

```
#include <RTCLib.h>
```

```

#include <dht.h>

#define dht_dpın A0 //no ; here. Set equal to channel sensor is on

dht DHT;

/* On the Ethernet Shield, CS is pin 4. Note that even if it's not used as
the CS pin, the hardware CS pin (10 on most Arduino boards, 53 on the Mega)
must be left as an output or the SD library functions will not work. */

const int chipSelect = 4;

/***** ETHERNET STUFF *****/

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };

byte ip[] = { 192, 168, 1, 110 };

EthernetServer server(80);

unsigned int localPort = 8888; // local port to listen for UDP packets

//IPAddress timeServer(192, 43, 244, 18); // time.nist.gov NTP server

IPAddress timeServer(192, 168, 1, 110); // ntp2c.mcc.ac.uk NTP server, University of Manchester,
England

const int NTP_PACKET_SIZE= 48; // NTP time stamp is in the first 48 bytes of the message

byte pb[NTP_PACKET_SIZE]; // buffer to hold incoming and outgoing packets

EthernetUDP Udp; // An EthernetUDP instance to let us send and receive packets over UDP

// Initialize the Ethernet client library with the IP address and port of the server
// that you want to connect to (port 80 is default for HTTP):

EthernetClient client;

/***** SDCARD STUFF *****/

Sd2Card card;

SdVolume volume;

SdFile root;

SdFile file;

SdFat sd; // file system object

SdFile myFile;

RTC_DS1307 RTC; // define the Real Time Clock object

// store error strings in flash to save RAM

#define error(s) error_P(PSTR(s))

```

```

// How big our line buffer should be. 100 is plenty!

#define BUFSIZ 100

// buffer to format data - makes it easier to echo to Serial

char buf[80];

// Serial print stream

ArduinoOutputStream cout(Serial);

char name[] = "WATMES00.CSV"; //WATER MEASUREMENT FILE

int flag=0;

// store error strings in flash to save RAM

#define error(s) error_P(PSTR(s))

void error_P(const char* str) {

    PgmPrint("error: ");

    SerialPrintln_P(str);

    if (card.errorCode()) {

        PgmPrint("SD error: ");

        Serial.print(card.errorCode(), HEX);

        Serial.print(',');

        Serial.println(card.errorData(), HEX);

    }

    while(1);

}

////////////////////////////////////

//-----

// format date/time

ostream& operator << (ostream& os, DateTime& dt) {

os << dt.year() << '/' << int(dt.month()) << '/' << int(dt.day()) << ',';

os << int(dt.hour()) << ':' << setfill('0') << setw(2) << int(dt.minute());

os << ':' << setw(2) << int(dt.second()) << setfill(' ');

```



```

return os;

}//-----

void PrintDateTime(DateTime t)

{

char datestr[24];

sprintf(datestr, "%04d-%02d-%02d %02d:%02d:%02d ", t.year(), t.month(), t.day(), t.hour(),
t.minute(), t.second());

Serial.print(datestr);

}

////////////////////////////////////

//-----

// send an NTP request to the time server at the given address

unsigned long sendNTPpacket(IPAddress& address)

{

// set all bytes in the buffer to 0

memset(pb, 0, NTP_PACKET_SIZE);

// Initialize values needed to form NTP request

// (see URL above for details on the packets)

pb[0] = 0b11100011; // LI, Version, Mode

pb[1] = 0; // Stratum, or type of clock

pb[2] = 6; // Polling Interval

pb[3] = 0xEC; // Peer Clock Precision

// 8 bytes of zero for Root Delay & Root Dispersion

pb[12] = 49;

pb[13] = 0x4E;

pb[14] = 49;

pb[15] = 52;

// all NTP fields have been given values, now

// you can send a packet requesting a timestamp:

Udp.beginPacket(address, 123); //NTP requests are to port 123

Udp.write(pb,NTP_PACKET_SIZE);

```

```

Udp.endPacket();

}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//-----

//*****

void setup() {

  Serial.begin(9600);

  delay(300);//Let system settle

 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
  //-----

  //*****

  //*****

  float h[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27,
  28, 29, 30};  // height of 30 water columns

  float zcom[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13};          //first columns height (used to
  find the water level)

 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
  //-----

  //-----

  float z1[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};          // height inside the water
  column //-----

  float z2[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};
  //-----

  float z3[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};

  float z4[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};

  float z5[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};

```

```
float z6[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float z7[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float z8[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float z9[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float z10[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float z11[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float z12[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float z13[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float z14[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float z15[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float z16[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float z17[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float z18[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float z19[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float z20[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float z21[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float z22[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float z23[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float z24[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float z25[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float z26[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float z27[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float z28[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float z29[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float z30[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
```

```
////////////////////////////////////
////////////////////////////////////
```

```
float A1[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};      // A is the factor  $pg\theta/2\eta$  for each column
float A2[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float A3[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float A4[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
```

```
float A5[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float A6[] = { 1, 2, 3, 4, 5, 6, 7, 8 ,9 , 10, 11, 12};
float A7[] = { 1, 2, 3, 4, 5, 6, 7, 8 ,9 , 10, 11, 12};
float A8[] = { 1, 2, 3, 4, 5, 6, 7, 8 ,9 , 10, 11, 12};
float A9[] = { 1, 2, 3, 4, 5, 6, 7, 8 ,9 , 10, 11, 12};
float A10[] = { 1, 2, 3, 4, 5, 6, 7, 8 ,9 , 10, 11, 12};
float A11[] = { 1, 2, 3, 4, 5, 6, 7, 8 ,9 , 10, 11, 12};
float A12[] = { 1, 2, 3, 4, 5, 6, 7, 8 ,9 , 10, 11, 12};
float A13[] = { 1, 2, 3, 4, 5, 6, 7, 8 ,9 , 10, 11, 12};
float A14[] = { 1, 2, 3, 4, 5, 6, 7, 8 ,9 , 10, 11, 12};
float A15[] = { 1, 2, 3, 4, 5, 6, 7, 8 ,9 , 10, 11, 12};
float A16[] = { 1, 2, 3, 4, 5, 6, 7, 8 ,9 , 10, 11, 12};
float A17[] = { 1, 2, 3, 4, 5, 6, 7, 8 ,9 , 10, 11, 12};
float A18[] = { 1, 2, 3, 4, 5, 6, 7, 8 ,9 , 10, 11, 12};
float A19[] = { 1, 2, 3, 4, 5, 6, 7, 8 ,9 , 10, 11, 12};
float A20[] = { 1, 2, 3, 4, 5, 6, 7, 8 ,9 , 10, 11, 12};
float A21[] = { 1, 2, 3, 4, 5, 6, 7, 8 ,9 , 10, 11, 12};
float A22[] = { 1, 2, 3, 4, 5, 6, 7, 8 ,9 , 10, 11, 12};
float A23[] = { 1, 2, 3, 4, 5, 6, 7, 8 ,9 , 10, 11, 12};
float A24[] = { 1, 2, 3, 4, 5, 6, 7, 8 ,9 , 10, 11, 12};
float A25[] = { 1, 2, 3, 4, 5, 6, 7, 8 ,9 , 10, 11, 12};
float A26[] = { 1, 2, 3, 4, 5, 6, 7, 8 ,9 , 10, 11, 12};
float A27[] = { 1, 2, 3, 4, 5, 6, 7, 8 ,9 , 10, 11, 12};
float A28[] = { 1, 2, 3, 4, 5, 6, 7, 8 ,9 , 10, 11, 12};
float A29[] = { 1, 2, 3, 4, 5, 6, 7, 8 ,9 , 10, 11, 12};
float A30[] = { 1, 2, 3, 4, 5, 6, 7, 8 ,9 , 10, 11, 12};
```

```
////////////////////////////////////
////////////////////////////////////
```

```
float U1[] = { 1, 2, 3, 4, 5, 6, 7, 8 ,9 , 10, 11, 12};
(in m/sec^2)
```

////// U is the water velocity in the column

```
float U2[] = { 1, 2, 3, 4, 5, 6, 7, 8 ,9 , 10, 11, 12};
```

```
float U3[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float U4[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float U5[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float U6[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float U7[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float U8[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float U9[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float U10[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float U11[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float U12[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float U13[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float U14[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float U15[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float U16[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float U17[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float U18[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float U19[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float U20[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float U21[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float U22[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float U23[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float U24[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float U25[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float U26[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float U27[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float U28[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float U29[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
float U30[] = { 1, 2, 3, 4, 5, 6, 7, 8,9 , 10, 11, 12};
```

```
////////////////////////////////////
////////////////////////////////////
```

float Q1[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};  
m<sup>3</sup>/sec)

////// Q is water discharge in the column (in

float Q2[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};

float Q3[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};

float Q4[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};

float Q5[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};

float Q6[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};

float Q7[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};

float Q8[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};

float Q9[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};

float Q10[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};

float Q11[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};

float Q12[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};

float Q13[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};

float Q14[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};

float Q15[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};

float Q16[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};

float Q17[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};

float Q18[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};

float Q19[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};

float Q20[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};

float Q21[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};

float Q22[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};

float Q23[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};

float Q24[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};

float Q25[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};

float Q26[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};

float Q27[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};

float Q28[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};

float Q29[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};

float Q30[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};

////////////////////////////////////  
////////////////////////////////////

float Qtot; // total water discharge of each column //

float Q1tot;

float Q2tot;

float Q3tot;

float Q4tot;

float Q5tot;

float Q6tot;

float Q7tot;

float Q8tot;

float Q9tot;

float Q10tot;

float Q11tot;

float Q12tot;

float Q13tot;

float Q14tot;

float Q15tot;

float Q16tot;

float Q17tot;

float Q18tot;

float Q19tot;

float Q20tot;

float Q21tot;

float Q22tot;

float Q23tot;

float Q24tot;

float Q25tot;

float Q26tot;

float Q27tot;

float Q28tot;

float Q29tot;

float Q30tot;

//  
//

zcom[1]=0 ; // height in first column above the sensor  
////////////////////////////////////

zcom[2]=0.5 ; // used for water level finding  
////////////////////////////////////

zcom[3]=1 ;

zcom[4]=1.5 ;

zcom[5]=2 ;

zcom[6]=2.5 ;

zcom[7]=3 ;

zcom[8]=3.5 ;

zcom[9]=4 ;

zcom[10]=4.5 ;

zcom[11]=5 ;

zcom[12]=5.5 ;

zcom[13]=6 ;

//  
//

//////////////////////////////////// factor  $\rho g \theta / 2 \eta$   
////////////////////////////////////

A1[1]=0.29;

A1[2]=A1[3]=A1[4]=A1[5]=A1[6]=A1[7]=A1[8]=A1[9]=A1[10]=A1[11]=0.22;

A1[12]=0.22;

A2[1]=0.44;

A2[2]=A2[3]=A2[4]=A2[5]=A2[6]=A2[7]=A2[8]=A2[9]=A2[10]=A2[11]=0.26;

A2[12]=0.54;

A3[1]=0.58;

A3[2]=A3[3]=A3[4]=A3[5]=A3[6]=A3[7]=A3[8]=A3[9]=A3[10]=A3[11]=0.35;

A3[12]=0.58;



A4[1]=0.66;  
A4[2]=A4[3]=A4[4]=A4[5]=A4[6]=A4[7]=A4[8]=A4[9]=A4[10]=A4[11]=0.37;  
A4[12]=0.52;  
A5[1]=0.6;  
A5[2]=A5[3]=A5[4]=A5[5]=A5[6]=A5[7]=A5[8]=A5[9]=A5[10]=A5[11]=0.4;  
A5[12]=0.6;  
A6[1]=0.59;  
A6[2]=A6[3]=A6[4]=A6[5]=A6[6]=A6[7]=A6[8]=A6[9]=A6[10]=A6[11]=0.46;  
A6[12]=0.62;  
A7[1]=0.57;  
A7[2]=A7[3]=A7[4]=A7[5]=A7[6]=A7[7]=A7[8]=A7[9]=A7[10]=A7[11]=0.55;  
A7[12]=0.67;  
A8[1]=0.55;  
A8[2]=A8[3]=A8[4]=A8[5]=A8[6]=A8[7]=A8[8]=A8[9]=A8[10]=A8[11]=0.57;  
A8[12]=0.65;  
A9[1]=0.64;  
A9[2]=A9[3]=A9[4]=A9[5]=A9[6]=A9[7]=A9[8]=A9[9]=A9[10]=A9[11]=0.56;  
A9[12]=0.73;  
A10[1]=0.61;  
A10[2]=A10[3]=A10[4]=A10[5]=A10[6]=A10[7]=A10[8]=A10[9]=A10[10]=A10[11]=0.61;  
A10[12]=0.04;  
A11[1]=0.61;  
A11[2]=A11[3]=A11[4]=A11[5]=A11[6]=A11[7]=A11[8]=A11[9]=A11[10]=A11[11]=0.59;  
A11[12]=0.04;  
A12[1]=0.55;  
A12[2]=A12[3]=A12[4]=A12[5]=A12[6]=A12[7]=A12[8]=A12[9]=A12[10]=A12[11]=0.56;  
A12[12]=0.07;  
A13[1]=0.64;  
A13[2]=A13[3]=A13[4]=A13[5]=A13[6]=A13[7]=A13[8]=A13[9]=A13[10]=A13[11]=0.53;  
A13[12]=0.15;

A14[1]=0.61;  
A14[2]=A14[3]=A14[4]=A14[5]=A14[6]=A14[7]=A14[8]=A14[9]=A14[10]=A14[11]=0.51;  
A14[12]=0.59;  
A15[1]=0.59;  
A15[2]=A15[3]=A15[4]=A15[5]=A15[6]=A15[7]=A15[8]=A15[9]=A15[10]=A15[11]=0.53;  
A15[12]=0.52;  
A16[1]=0.58;  
A16[2]=A16[3]=A16[4]=A16[5]=A16[6]=A16[7]=A16[8]=A16[9]=A16[10]=A16[11]=0.61;  
A16[12]=0.03;  
A17[1]=0.55;  
A17[2]=A17[3]=A17[4]=A17[5]=A17[6]=A17[7]=A17[8]=A17[9]=A17[10]=A17[11]=0.57;  
A17[12]=0.37;  
A18[1]=0.59;  
A18[2]=A18[3]=A18[4]=A18[5]=A18[6]=A18[7]=A18[8]=A18[9]=A18[10]=A18[11]=0.55;  
A18[12]=0.59;  
A19[1]=0.54;  
A19[2]=A19[3]=A19[4]=A19[5]=A19[6]=A19[7]=A19[8]=A19[9]=A19[10]=A19[11]=0.63;  
A19[12]=0.74;  
A20[1]=0.53;  
A20[2]=A20[3]=A20[4]=A20[5]=A20[6]=A20[7]=A20[8]=A20[9]=A20[10]=A20[11]=0.53;  
A20[12]=0.07;  
A21[1]=0.54;  
A21[2]=A21[3]=A21[4]=A21[5]=A21[6]=A21[7]=A21[8]=A21[9]=A21[10]=A21[11]=0.58;  
A21[12]=0.46;  
A22[1]=0.49;  
A22[2]=A22[3]=A22[4]=A22[5]=A22[6]=A22[7]=A22[8]=A22[9]=A22[10]=A22[11]=0.45;  
A22[12]=0.04;  
A23[1]=0.47;  
A23[2]=A23[3]=A23[4]=A23[5]=A23[6]=A23[7]=A23[8]=A23[9]=A23[10]=A23[11]=0.53;  
A23[12]=0.13;

A24[1]=0.48;  
A24[2]=A24[3]=A24[4]=A24[5]=A24[6]=A24[7]=A24[8]=A24[9]=A24[10]=A24[11]=0.53;  
A24[12]=0.55;  
A25[1]=0.48;  
A25[2]=A25[3]=A25[4]=A25[5]=A25[6]=A25[7]=A25[8]=A25[9]=A25[10]=A25[11]=0.5;  
A25[12]=0.64;  
A26[1]=0.49;  
A26[2]=A26[3]=A26[4]=A26[5]=A26[6]=A26[7]=A26[8]=A26[9]=A26[10]=A26[11]=0.5;  
A26[12]=1.06;  
A27[1]=0.47;  
A27[2]=A27[3]=A27[4]=A27[5]=A27[6]=A27[7]=A27[8]=A27[9]=A27[10]=A27[11]=0.51;  
A27[12]=0.85;  
A28[1]=0.32;  
A28[2]=A28[3]=A28[4]=A28[5]=A28[6]=A28[7]=A28[8]=A28[9]=A28[10]=A28[11]=0.43;  
A28[12]=0.14;  
A29[1]=0.21;  
A29[2]=A29[3]=A29[4]=A29[5]=A29[6]=A29[7]=A29[8]=A29[9]=A29[10]=A29[11]=0.23;  
A29[12]=0.64;  
A30[1]=0.13;  
A30[2]=A30[3]=A30[4]=A30[5]=A30[6]=A30[7]=A30[8]=A30[9]=A30[10]=A30[11]=0.19;  
A30[12]=0.14;

//

// depth calculation//

//

int analogPin = 1; // 4-20mA input connected to analog pin 1

float H = 0;

int input = 0;

float L = 0.78; // line slope from calibration

```

input = analogRead(analogPin); // read the input pin

H = (input * L)/100;

////////////////////////////////////
////////////////////////////////////

h[1]=h[2]=h[3]=h[4]=h[5]=h[6]=h[7]=h[8]=h[9]=H-0.5 ; //////////////////////////////////////////////////depth of
each column////////////////////////////////////

h[10]=h[11]=h[12]=h[13]=h[14]=h[15]=h[16]=h[17]=h[18]=h[19]=h[20]=h[21]=H-0.2 ;

h[22]=h[23]=h[24]=h[25]=h[26]=h[27]=h[28]=h[29]=H ;

h[30]=H-0.5 ;

//////////////////////////////////// height inside each column
////////////////////////////////////

z1[1]=z2[1]=z3[1]=z4[1]=z5[1]=z6[1]=z7[1]=z8[1]=z9[1]=z10[1]=z11[1]=z12[1]=z13[1]=z14[1]=z15[1]=
z16[1]=z17[1]=z18[1]=z19[1]=z20[1]=z21[1]=z22[1]=z23[1]=z24[1]=z25[1]=z26[1]=z27[1]=z28[1]=z29
[1]=z30[1]=0.25;

z1[2]=z2[2]=z3[2]=z4[2]=z5[2]=z6[2]=z7[2]=z8[2]=z9[2]=z10[2]=z11[2]=z12[2]=z13[2]=z14[2]=z15[2]=
z16[2]=z17[2]=z18[2]=z19[2]=z20[2]=z21[2]=z22[2]=z23[2]=z24[2]=z25[2]=z26[2]=z27[2]=z28[2]=z29
[2]=z30[2]=0.75;

z1[3]=z2[3]=z3[3]=z4[3]=z5[3]=z6[3]=z7[3]=z8[3]=z9[3]=z10[3]=z11[3]=z12[3]=z13[1]=z14[3]=z15[3]=
z16[3]=z17[3]=z18[3]=z19[3]=z20[3]=z21[3]=z22[3]=z23[3]=z24[3]=z25[3]=z26[3]=z27[3]=z28[3]=z29
[3]=z30[3]=1.25;

z1[4]=z2[4]=z3[4]=z4[4]=z5[4]=z6[4]=z7[4]=z8[4]=z9[4]=z10[4]=z11[4]=z12[4]=z13[1]=z14[4]=z15[4]=
z16[4]=z17[4]=z18[4]=z19[4]=z20[4]=z21[4]=z22[4]=z23[4]=z24[4]=z25[4]=z26[4]=z27[4]=z28[4]=z29
[4]=z30[4]=1.75;

z1[5]=z2[5]=z3[5]=z4[5]=z5[5]=z6[5]=z7[5]=z8[5]=z9[5]=z10[5]=z11[5]=z12[5]=z13[1]=z14[5]=z15[5]=
z16[5]=z17[5]=z18[5]=z19[5]=z20[5]=z21[5]=z22[5]=z23[5]=z24[5]=z25[5]=z26[5]=z27[5]=z28[5]=z29
[5]=z30[5]=2.25;

z1[6]=z2[6]=z3[6]=z4[6]=z5[6]=z6[6]=z7[6]=z8[6]=z9[6]=z10[6]=z11[6]=z12[6]=z13[1]=z14[6]=z15[6]=
z16[6]=z17[6]=z18[6]=z19[6]=z20[6]=z21[6]=z22[6]=z23[6]=z24[6]=z25[6]=z26[6]=z27[6]=z28[6]=z29
[6]=z30[6]=2.75;

z1[7]=z2[7]=z3[7]=z4[7]=z5[7]=z6[7]=z7[7]=z8[7]=z9[7]=z10[7]=z11[7]=z12[7]=z13[1]=z14[7]=z15[7]=
z16[7]=z17[7]=z18[7]=z19[7]=z20[7]=z21[7]=z22[7]=z23[7]=z24[7]=z25[7]=z26[7]=z27[7]=z28[7]=z29
[7]=z30[7]=3.25;

```

z1[8]=z2[8]=z3[8]=z4[8]=z5[8]=z6[8]=z7[8]=z8[8]=z9[8]=z10[8]=z11[8]=z12[8]=z13[1]=z14[8]=z15[8]=z16[8]=z17[8]=z18[8]=z19[8]=z20[8]=z21[8]=z22[8]=z23[8]=z24[8]=z25[8]=z26[8]=z27[8]=z28[8]=z29[8]=z30[8]=3.75;

z1[9]=z2[9]=z3[9]=z4[9]=z5[9]=z6[9]=z7[9]=z8[9]=z9[9]=z10[9]=z11[9]=z12[9]=z13[1]=z14[9]=z15[9]=z16[9]=z17[9]=z18[9]=z19[9]=z20[9]=z21[9]=z22[9]=z23[9]=z24[9]=z25[9]=z26[9]=z27[9]=z28[9]=z29[9]=z30[9]=4.25;

z1[10]=z2[10]=z3[10]=z4[10]=z5[10]=z6[10]=z7[10]=z8[10]=z9[10]=z10[10]=z11[10]=z12[10]=z13[10]=z14[10]=z15[10]=z16[10]=z17[10]=z18[10]=z19[10]=z20[10]=z21[10]=z22[10]=z23[10]=z24[10]=z25[10]=z26[10]=z27[10]=z28[10]=z29[10]=z30[10]=4.75;

z1[11]=z2[11]=z3[11]=z4[11]=z5[11]=z6[11]=z7[11]=z8[11]=z9[11]=z10[11]=z11[11]=z12[11]=z13[11]=z14[11]=z15[11]=z16[11]=z17[11]=z18[11]=z19[11]=z20[11]=z21[11]=z22[11]=z23[11]=z24[11]=z25[11]=z26[11]=z27[11]=z28[11]=z29[11]=z30[11]=5.25;

z1[12]=z2[12]=z3[12]=z4[12]=z5[12]=z6[12]=z7[12]=z8[12]=z9[12]=z10[12]=z11[12]=z12[12]=z13[12]=z14[12]=z15[12]=z16[12]=z17[12]=z18[12]=z19[12]=z20[12]=z21[12]=z22[12]=z23[12]=z24[12]=z25[12]=z26[12]=z27[12]=z28[12]=z29[12]=z30[12]=5.75;

////////////////////////////////////  
////////////////////////////////////

float a1[] = { 1, 2, 3, 4, 5, 6, 7,8 ,9 , 10, 11, 12}; // area of column parts  
////////////////////////////////////

float a2[] = { 1, 2, 3, 4, 5, 6, 7,8 ,9 , 10, 11, 12};

float a3[] = { 1, 2, 3, 4, 5, 6, 7,8 ,9 , 10, 11, 12};

float a4[] = { 1, 2, 3, 4, 5, 6, 7,8 ,9 , 10, 11, 12};

float a5[] = { 1, 2, 3, 4, 5, 6, 7,8 ,9 , 10, 11, 12};

float a6[] = { 1, 2, 3, 4, 5, 6, 7,8 ,9 , 10, 11, 12};

float a7[] = { 1, 2, 3, 4, 5, 6, 7,8 ,9 , 10, 11, 12};

float a8[] = { 1, 2, 3, 4, 5, 6, 7,8 ,9 , 10, 11, 12};

float a9[] = { 1, 2, 3, 4, 5, 6, 7,8 ,9 , 10, 11, 12};

float a10[] = { 1, 2, 3, 4, 5, 6, 7,8 ,9 , 10, 11, 12};

float a11[] = { 1, 2, 3, 4, 5, 6, 7,8 ,9 , 10, 11, 12};

float a12[] = { 1, 2, 3, 4, 5, 6, 7,8 ,9 , 10, 11, 12};

float a13[] = { 1, 2, 3, 4, 5, 6, 7,8 ,9 , 10, 11, 12};

float a14[] = { 1, 2, 3, 4, 5, 6, 7,8 ,9 , 10, 11, 12};

float a15[] = { 1, 2, 3, 4, 5, 6, 7,8 ,9 , 10, 11, 12};

float a16[] = { 1, 2, 3, 4, 5, 6, 7,8 ,9 , 10, 11, 12};

float a17[] = { 1, 2, 3, 4, 5, 6, 7,8 ,9 , 10, 11, 12};

```
float a18[] = { 1, 2, 3, 4, 5, 6, 7,8 ,9 , 10, 11, 12};
float a19[] = { 1, 2, 3, 4, 5, 6, 7,8 ,9 , 10, 11, 12};
float a20[] = { 1, 2, 3, 4, 5, 6, 7,8 ,9 , 10, 11, 12};
float a21[] = { 1, 2, 3, 4, 5, 6, 7,8 ,9 , 10, 11, 12};
float a22[] = { 1, 2, 3, 4, 5, 6, 7,8 ,9 , 10, 11, 12};
float a23[] = { 1, 2, 3, 4, 5, 6, 7,8 ,9 , 10, 11, 12};
float a24[] = { 1, 2, 3, 4, 5, 6, 7,8 ,9 , 10, 11, 12};
float a25[] = { 1, 2, 3, 4, 5, 6, 7,8 ,9 , 10, 11, 12};
float a26[] = { 1, 2, 3, 4, 5, 6, 7,8 ,9 , 10, 11, 12};
float a27[] = { 1, 2, 3, 4, 5, 6, 7,8 ,9 , 10, 11, 12};
float a28[] = { 1, 2, 3, 4, 5, 6, 7,8 ,9 , 10, 11, 12};
float a29[] = { 1, 2, 3, 4, 5, 6, 7,8 ,9 , 10, 11, 12};
float a30[] = { 1, 2, 3, 4, 5, 6, 7,8 ,9 , 10, 11, 12};
```

```
////////////////////////////////////
////////////////////////////////////
```

```
//////////////////////////////////// area value
////////////////////////////////////
```

```
a1[1]=a2[1]=a3[1]=a4[1]=a5[1]=a6[1]=a7[1]=a8[1]=a9[1]=a10[1]=a11[1]=a12[1]=a13[1]=a14[1]=a15[
1]=a16[1]=a17[1]=a18[1]=a19[1]=a20[1]=a21[1]=a22[1]=a23[1]=a24[1]=a25[1]=a26[1]=a27[1]=a28[1
]=a29[1]=a30[1]=0.19;
```

```
a1[2]=a2[2]=a3[2]=a4[2]=a5[2]=a6[2]=a7[2]=a8[2]=a9[2]=a10[2]=a11[2]=a12[2]=a13[2]=a14[2]=a15[
2]=a16[2]=a17[2]=a18[2]=a19[2]=a20[2]=a21[2]=a22[2]=a23[2]=a24[2]=a25[2]=a26[2]=a27[2]=a28[2
]=a29[2]=a30[2]=0.25;
```

```
a1[3]=a2[3]=a3[3]=a4[3]=a5[3]=a6[3]=a7[3]=a8[3]=a9[3]=a10[3]=a11[3]=a12[3]=a13[1]=a14[3]=a15[
3]=a16[3]=a17[3]=a18[3]=a19[3]=a20[3]=a21[3]=a22[3]=a23[3]=a24[3]=a25[3]=a26[3]=a27[3]=a28[3
]=a29[3]=a30[3]=0.25;
```

```
a1[4]=a2[4]=a3[4]=a4[4]=a5[4]=a6[4]=a7[4]=a8[4]=a9[4]=a10[4]=a11[4]=a12[4]=a13[1]=a14[4]=a15[
4]=a16[4]=a17[4]=a18[4]=a19[4]=a20[4]=a21[4]=a22[4]=a23[4]=a24[4]=a25[4]=a26[4]=a27[4]=a28[4
]=a29[4]=a30[4]=0.25;
```

```
a1[5]=a2[5]=a3[5]=a4[5]=a5[5]=a6[5]=a7[5]=a8[5]=a9[5]=a10[5]=a11[5]=a12[5]=a13[1]=a14[5]=a15[
5]=a16[5]=a17[5]=a18[5]=a19[5]=a20[5]=a21[5]=a22[5]=a23[5]=a24[5]=a25[5]=a26[5]=a27[5]=a28[5
]=a29[5]=a30[5]=0.25;
```

```
a1[6]=a2[6]=a3[6]=a4[6]=a5[6]=a6[6]=a7[6]=a8[6]=a9[6]=a10[6]=a11[6]=a12[6]=a13[1]=a14[6]=a15[
6]=a16[6]=a17[6]=a18[6]=a19[6]=a20[6]=a21[6]=a22[6]=a23[6]=a24[6]=a25[6]=a26[6]=a27[6]=a28[6
]=a29[6]=a30[6]=0.25;
```

a1[7]=a2[7]=a3[7]=a4[7]=a5[7]=a6[7]=a7[7]=a8[7]=a9[7]=a10[7]=a11[7]=a12[7]=a13[1]=a14[7]=a15[7]=a16[7]=a17[7]=a18[7]=a19[7]=a20[7]=a21[7]=a22[7]=a23[7]=a24[7]=a25[7]=a26[7]=a27[7]=a28[7]=a29[7]=a30[7]=0.25;

a1[8]=a2[8]=a3[8]=a4[8]=a5[8]=a6[8]=a7[8]=a8[8]=a9[8]=a10[8]=a11[8]=a12[8]=a13[1]=a14[8]=a15[8]=a16[8]=a17[8]=a18[8]=a19[8]=a20[8]=a21[8]=a22[8]=a23[8]=a24[8]=a25[8]=a26[8]=a27[8]=a28[8]=a29[8]=a30[8]=0.25;

a1[9]=a2[9]=a3[9]=a4[9]=a5[9]=a6[9]=a7[9]=a8[9]=a9[9]=a10[9]=a11[9]=a12[9]=a13[1]=a14[9]=a15[9]=a16[9]=a17[9]=a18[9]=a19[9]=a20[9]=a21[9]=a22[9]=a23[9]=a24[9]=a25[9]=a26[9]=a27[9]=a28[9]=a29[9]=a30[9]=0.25;

a1[10]=a2[10]=a3[10]=a4[10]=a5[10]=a6[10]=a7[10]=a8[10]=a9[10]=a10[10]=a11[10]=a12[10]=a13[10]=a14[10]=a15[10]=a16[10]=a17[10]=a18[10]=a19[10]=a20[10]=a21[10]=a22[10]=a23[10]=a24[10]=a25[10]=a26[10]=a27[10]=a28[10]=a29[10]=a30[10]=0.25;

a1[11]=a2[11]=a3[11]=a4[11]=a5[11]=a6[11]=a7[11]=a8[11]=a9[11]=a10[11]=a11[11]=a12[11]=a13[11]=a14[11]=a15[11]=a16[11]=a17[11]=a18[11]=a19[11]=a20[11]=a21[11]=a22[11]=a23[11]=a24[11]=a25[11]=a26[11]=a27[11]=a28[11]=a29[11]=a30[11]=0.25;

a1[12]=a2[12]=a3[12]=a4[12]=a5[12]=a6[12]=a7[12]=a8[12]=a9[12]=a10[12]=a11[12]=a12[12]=a13[12]=a14[12]=a15[12]=a16[12]=a17[12]=a18[12]=a19[12]=a20[12]=a21[12]=a22[12]=a23[12]=a24[12]=a25[12]=a26[12]=a27[12]=a28[12]=a29[12]=a30[12]=0.20;

////////////////////////////////////  
////////

```
for (int i=0; i <= 12; i++){
```

```
    for (int k=0; k <= 13; k++){
```

```
        while (zcom[k]<h[1]){
```

```
////////////////////////////////////
```

```
////////////////////////////////////Calculate U and Q for each area////////////////////////////////////
```

```
////////////////////////////////////
```

```
U1[i]=A1[i]*z1[i]*(2*h[i]-z1[i]) ;
```

```
Q1[i]=U1[i]*a1[i];
```

```
////////////////////////////////////
```

```
U2[i]=A2[i]*z2[i]*(2*h[i]-z2[i]) ;
```

```
Q2[i]=U2[i]*a2[i];
```

```
////////////////////////////////////
```

$$U3[i]=A3[i]*z3[i]*(2*h[i]-z3[i]) ;$$

$$Q3[i]=U3[i]*a3[i];$$

////////////////////////////////////

$$U4[i]=A4[i]*z4[i]*(2*h[i]-z4[i]) ;$$

$$Q4[i]=U4[i]*a4[i];$$

////////////////////////////////////

$$U5[i]=A5[i]*z5[i]*(2*h[i]-z5[i]) ;$$

$$Q5[i]=U5[i]*a5[i];$$

////////////////////////////////////

$$U6[i]=A6[i]*z6[i]*(2*h[i]-z6[i]) ;$$

$$Q6[i]=U6[i]*a6[i];$$

////////////////////////////////////

$$U7[i]=A7[i]*z7[i]*(2*h[i]-z7[i]) ;$$

$$Q7[i]=U7[i]*a7[i];$$

////////////////////////////////////

$$U8[i]=A8[i]*z8[i]*(2*h[i]-z8[i]) ;$$

$$Q8[i]=U8[i]*a8[i];$$

////////////////////////////////////

$$U9[i]=A9[i]*z9[i]*(2*h[i]-z9[i]) ;$$

$$Q9[i]=U9[i]*a9[i];$$

////////////////////////////////////

$$U10[i]=A10[i]*z10[i]*(2*h[i]-z10[i]) ;$$

$$Q10[i]=U10[i]*a10[i];$$

////////////////////////////////////

$$U11[i]=A11[i]*z11[i]*(2*h[i]-z11[i]) ;$$

$$Q11[i]=U11[i]*a11[i];$$

////////////////////////////////////

$$U12[i]=A12[i]*z12[i]*(2*h[i]-z12[i]) ;$$

$$Q12[i]=U12[i]*a12[i];$$

////////////////////////////////////



U13[i]=A13[i]\*z13[i]\*(2\*h[i]-z13[i]) ;

Q13[i]=U13[i]\*a13[i];

////////////////////////////////////

U14[i]=A14[i]\*z14[i]\*(2\*h[i]-z14[i]) ;

Q14[i]=U14[i]\*a14[i];

////////////////////////////////////

U15[i]=A15[i]\*z15[i]\*(2\*h[i]-z15[i]) ;

Q15[i]=U15[i]\*a15[i];

////////////////////////////////////

U16[i]=A16[i]\*z16[i]\*(2\*h[i]-z16[i]) ;

Q16[i]=U16[i]\*a16[i];

////////////////////////////////////

U17[i]=A17[i]\*z17[i]\*(2\*h[i]-z17[i]) ;

Q17[i]=U17[i]\*a17[i];

////////////////////////////////////

U18[i]=A18[i]\*z18[i]\*(2\*h[i]-z18[i]) ;

Q18[i]=U18[i]\*a18[i];

////////////////////////////////////

U19[i]=A19[i]\*z19[i]\*(2\*h[i]-z19[i]) ;

Q19[i]=U19[i]\*a19[i];

////////////////////////////////////

U20[i]=A20[i]\*z20[i]\*(2\*h[i]-z20[i]) ;

Q20[i]=U20[i]\*a20[i];

////////////////////////////////////

U21[i]=A21[i]\*z21[i]\*(2\*h[i]-z21[i]) ;

Q21[i]=U21[i]\*a21[i];

////////////////////////////////////

U22[i]=A22[i]\*z22[i]\*(2\*h[i]-z22[i]) ;

Q22[i]=U22[i]\*a22[i];

////////////////////////////////////

$$U23[i]=A23[i]*z23[i]*(2*h[i]-z23[i]) ;$$

$$Q23[i]=U23[i]*a23[i];$$

////////////////////////////////////

$$U24[i]=A24[i]*z24[i]*(2*h[i]-z24[i]) ;$$

$$Q24[i]=U24[i]*a24[i];$$

////////////////////////////////////

$$U25[i]=A25[i]*z25[i]*(2*h[i]-z25[i]) ;$$

$$Q25[i]=U25[i]*a25[i];$$

////////////////////////////////////

$$U26[i]=A26[i]*z26[i]*(2*h[i]-z26[i]) ;$$

$$Q26[i]=U26[i]*a26[i];$$

////////////////////////////////////

$$U27[i]=A27[i]*z27[i]*(2*h[i]-z27[i]) ;$$

$$Q27[i]=U27[i]*a27[i];$$

////////////////////////////////////

$$U28[i]=A28[i]*z28[i]*(2*h[i]-z28[i]) ;$$

$$Q28[i]=U28[i]*a28[i];$$

////////////////////////////////////

$$U29[i]=A29[i]*z29[i]*(2*h[i]-z29[i]) ;$$

$$Q29[i]=U29[i]*a29[i];$$

////////////////////////////////////

$$U30[i]=A30[i]*z30[i]*(2*h[i]-z30[i]) ;$$

$$Q30[i]=U30[i]*a30[i];$$

////////////////////////////////////

////////////////////////////////////add column discharge////////////////////////////////////

////////////////////////////////////

$$Q1tot=Q1tot + Q1[i] ;$$

$$Q2tot=Q2tot + Q2[i] ;$$

$$Q3tot=Q3tot + Q3[i] ;$$

```
Q4tot=Q4tot + Q4[i] ;
Q5tot=Q5tot + Q5[i] ;
Q6tot=Q6tot + Q6[i] ;
Q7tot=Q7tot + Q7[i] ;
Q8tot=Q8tot + Q8[i] ;
Q9tot=Q9tot + Q9[i] ;
Q10tot=Q10tot + Q10[i] ;
Q11tot=Q11tot + Q11[i] ;
Q12tot=Q12tot + Q12[i] ;
Q13tot=Q13tot + Q13[i] ;
Q14tot=Q14tot + Q14[i] ;
Q15tot=Q15tot + Q15[i] ;
Q16tot=Q16tot + Q16[i] ;
Q17tot=Q17tot + Q17[i] ;
Q18tot=Q18tot + Q18[i] ;
Q19tot=Q19tot + Q19[i] ;
Q20tot=Q20tot + Q20[i] ;
Q21tot=Q21tot + Q21[i] ;
Q22tot=Q22tot + Q22[i] ;
Q23tot=Q23tot + Q23[i] ;
Q24tot=Q24tot + Q24[i] ;
Q25tot=Q25tot + Q25[i] ;
Q26tot=Q26tot + Q26[i] ;
Q27tot=Q27tot + Q27[i] ;
Q28tot=Q28tot + Q28[i] ;
Q29tot=Q29tot + Q29[i] ;
Q30tot=Q30tot + Q30[i] ;
}
}
}
```

```
Qtot = Q1tot + Q2tot + Q3tot + Q4tot + Q5tot + Q6tot + Q7tot + Q8tot + Q9tot + Q10tot + Q11tot +
Q12tot + Q13tot + Q14tot + Q15tot + Q16tot + Q17tot + Q18tot + Q19tot + Q20tot + Q21tot +
Q22tot + Q23tot + Q24tot + Q25tot + Q26tot + Q27tot + Q28tot + Q29tot + Q30tot;
```

```
Serial.print("PAROXH(m^3/sec): ");
```

```
Serial.println(Qtot);
```

```
delay(1000);
```

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
//-----
```

```
PgmPrint("Free RAM: ");
```

```
Serial.println(FreeRam());
```

```
// initialize the SD card at SPI_HALF_SPEED to avoid bus errors with
```

```
// breadboards. use SPI_FULL_SPEED for better performance.
```

```
pinMode(10, OUTPUT);          // set the SS pin as an output (necessary!)
```

```
digitalWrite(10, HIGH);      // but turn off the W5100 chip!
```

```
if (!card.init(SPI_HALF_SPEED, 4)) error("card.init failed!");
```

```
// initialize a FAT volume
```

```
if (!volume.init(&card)) error("vol.init failed!");
```

```
PgmPrint("Volume is FAT");
```

```
Serial.println(volume.fatType(),DEC);
```

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

```
if (!root.openRoot(&volume)) error("openRoot failed!");
```

```

// list file in root with date and size
PgmPrintln("Files found in root:");
root.ls(LS_DATE | LS_SIZE);
Serial.println();

// Recursive list of all directories
PgmPrintln("Files found in all dirs:");
root.ls(LS_R);

Serial.println();
PgmPrintln("Done");
// Debugging complete, we start the server!
Ethernet.begin(mac, ip);
server.begin();

// init RTC
Wire.begin();
RTC.begin();
//RTC.adjust(DateTime(__DATE__, __TIME__));
Serial.println("Initializing RTC ...");
Serial.print("RTC without syncing with NTP server: ");
Serial.print("----> ");
PrintDateTime(RTC.now());
Serial.println();
Serial.println();

// Debugging complete, we start the server! start the Ethernet connection:
/*if (Ethernet.begin(mac) == 0) {
Serial.println("Failed to configure Ethernet using DHCP");
// no point in carrying on, so do nothing forevermore:

```

```

for(;;)

;

}*/

//if (Ethernet.begin(mac) != 0) { //if network exists_____

Serial.println("Initializing Network ...");

// print your local IP address:

Serial.print("My IP address: ");

for (byte thisByte = 0; thisByte < 4; thisByte++) {

// print the value of each byte of the IP address:

Serial.print(Ethernet.localIP()[thisByte], DEC);

Serial.print(".");

}

Serial.println();

Serial.println();

Serial.println("NTP-to-RTC");

Udp.begin(localPort);

// send an NTP packet to a time server

sendNTPpacket(timeServer);

// wait to see if a reply is available

delay(1000);

if ( Udp.parsePacket() ) {

// We've received a packet, read the data from it

Udp.read(pb, NTP_PACKET_SIZE);

// NTP contains four timestamps with an integer part and a fraction part

// we only use the integer part here

unsigned long t1, t2, t3, t4;

t1 = t2 = t3 = t4 = 0;

for (int i=0; i< 4; i++){

t1 = t1 << 8 | pb[16+i];

t2 = t2 << 8 | pb[24+i];

```

```

t3 = t3 << 8 | pb[32+i];
t4 = t4 << 8 | pb[40+i];
}

// part of the fractional part could be 4 bytes but this is more precise than the 1307 RTC
// which has a precision of ONE second. In fact one byte is sufficient for 1307
float f1,f2,f3,f4;
f1 = ((long)pb[20] * 256 + pb[21]) / 65536.0;
f2 = ((long)pb[28] * 256 + pb[29]) / 65536.0;
f3 = ((long)pb[36] * 256 + pb[37]) / 65536.0;
f4 = ((long)pb[44] * 256 + pb[45]) / 65536.0;

// NOTE:
// one could use the fractional part to set the RTC more precise
// 1) at the right (calculated) moment to the NEXT second!
// t4++;
// delay(1000 - f4*1000);
RTC.adjust(DateTime(t4));
// keep in mind that the time in the packet was the time at
// the NTP server at sending time so one should take into account
// the network latency (try ping!) and the processing of the data
// ==> delay (850 - f4*1000);
// 2) simply use it to round up the second
// f > 0.5 => add 1 to the second before adjusting the RTC
// (or lower threshold eg 0.4 if one keeps network latency etc in mind)
// 3) a SW RTC might be more precise, => ardomic clock :)
// convert NTP to UNIX time, differs seventy years = 2208988800 seconds
// NTP starts Jan 1, 1900
// Unix time starts on Jan 1 1970.
const unsigned long seventyYears = 2208988800UL;
t1 -= seventyYears;
t2 -= seventyYears;

```

```

t3 -= seventyYears;

t4 -= seventyYears;

Serial.print("UTC Time is : ");

PrintDateTime(DateTime(t4)); Serial.print(f4,4);

Serial.println();

RTC.adjust(DateTime(t4));

Serial.print("RTC with UTC Time : ");

PrintDateTime(RTC.now());

Serial.println();

DateTime now = RTC.now(); // get current time

DateTime PlusTwo (now. get() + 7200);

// add two hours

RTC.adjust(PlusTwo);

// set to new time

Serial.print("In Greece time is: ");

PrintDateTime(RTC.now());

Serial.println();

Serial.println("Done ...");

Serial.println("-----");

Serial.println();

}

else{

Serial.println("No UDP available ...");

}

//endif network exists_____

//else{ //else network does not exist

//Serial.println("Failed to configure Ethernet using DHCP");

// Initialize SdFat or print a detailed error message and halt

// Use half speed like the native library.

```



```

// change to SPI_FULL_SPEED for more performance.
if (!sd.begin(chipSelect, SPI_HALF_SPEED)) sd.initErrorHalt();

int day;

int month;

int year;

DateTime now = RTC.now();

name[0] = 'W';

name[1] = 'F';

day = now.day();

/*Serial.print("Day: ");

Serial.print(day);

Serial.println(); */

if(day<10){

name[2]='0';

name[3]= day % 10 + '0';}

else{

name[2]= (day - (day % 10))/10 + '0';

name[3]= day % 10 + '0';

}

month = now.month();

if(month<10){

name[4]='0';

name[5]= month % 10 + '0';}

else{

name[4]= (month - (month % 10))/10 + '0';

name[5]= month % 10 + '0';

}

year = now.year();

year = year - 2000;

if(year<10){

```

```

name[6]='0';
name[7]= year % 10 + '0';}
else{
name[6]= (year - (year % 10))/10 + '0';
name[7]= year % 10 + '0';
}

// open the file for write at end like the Native SD library
if (myFile.open(name, O_RDWR | O_CREAT | O_EXCL | O_AT_END )) {

//O_RDWR - Open for reading and writing.
//O_AT_END - Set the initial position at the end of the file.
//O_CREAT - If the file exists, this flag has no effect except as noted under O_EXCL below.
// Otherwise, the file shall be created.
//O_EXCL - If O_CREAT and O_EXCL are set, open() shall fail if the file exists.

Serial.print("Creating ");
Serial.println(name);

// if the file opened okay, write to it:
myFile.print("Date,Time,Q(m^3/sec),S(m),H(%),T(C)");
Serial.print("Date,Time,Q(m^3/sec),S(m),H(%),T(C)");

myFile.println(" ");
Serial.println();

// close the file:
myFile.close();}

else{

Serial.println("Opening file .csv for writing the header failed!!!");
Serial.println();

//sd.errorHalt("opening file .csv for writing the header failed");
}

// open the file for writing, at end, the values from sensors

```

```
if (!myFile.open(name, O_RDWR | O_AT_END)) {  
sd.errorHalt("opening file .csv for writing from sensor failed");  
}
```

```
////////////////////////////////////// WRITE TO THE FILE  
//////////////////////////////////////
```

```
Serial.println("Writing to the file...");
```

```
DateTime now1 = RTC.now();
```

```
int dayf,monthf,yearf;
```

```
dayf = now1.day();
```

```
monthf = now1.month();
```

```
yearf = now1.year();
```

```
myFile.print(dayf);
```

```
myFile.print("-");
```

```
myFile.print(monthf);
```

```
myFile.print("-");
```

```
myFile.print(yearf);
```

```
myFile.print(",");
```

```
Serial.print(dayf);
```

```
Serial.print("-");
```

```
Serial.print(monthf);
```

```
Serial.print("-");
```

```
Serial.print(yearf);
```

```
Serial.print(",");
```

```
int hourf,minf,secf;
```

```
hourf = now1.hour();
```

```
minf = now1.minute();
```

```
secf = now1.second();
```

```
myFile.print(hourf);
```

```
myFile.print(":");
```

```
myFile.print(minf);
```

```

myFile.print(":");
myFile.print(secf);
Serial.print(hourf);
Serial.print(":");
Serial.print(minf);
Serial.print(":");
Serial.print(secf);

//***** write to file *****//
myFile.print(",");
//Serial.print(",");
myFile.print(Qtot); // write discharge value
//Serial.print(Q);
////////////////////////////////////
myFile.print(",");
//Serial.print(",");
myFile.print(H); // write water height value
//Serial.print(S);
myFile.print(",");
////////////////////////////////////
DHT.read11(dht_dp); // read DHT22 sensor
delay(500);
myFile.print(DHT.humidity); // write humidity
myFile.print(",");
////////////////////////////////////
myFile.print(DHT.temperature); // write temperature
//*****//
delay(500);

myFile.println(" "); // change line

```

```

Serial.println();

// close the file:
myFile.close();
Serial.println("File closed..");
Serial.println();
// re-open the file for reading:
if (!myFile.open(name, O_READ)) {
sd.errorHalt("opening file .csv for read failed");
}
Serial.print("Opening ");
Serial.print(name);
Serial.print(" for reading: ");
Serial.println();
// read from the file until there's nothing else in it:
int data;
while ((data = myFile.read()) >= 0) Serial.write(data);
// close the file:
myFile.close();
Serial.println("Finished reading!!!");
Serial.println();
delay(700); //Wait rest of 1000ms recommended delay before
//accessing sensor
}

void ListFiles(EthernetClient client, uint8_t flags) {
// This code is just copied from SdFile.cpp in the SDFat library
// and tweaked to print to the client output in html!
dir_t p;

```

```

root.rewind();

client.println("<ul>");

while (root.readDir(&p) > 0) {

    // done if past last used entry

    if (p.name[0] == DIR_NAME_FREE) break;

    // skip deleted entry and entries for . and ..

    if (p.name[0] == DIR_NAME_DELETED || p.name[0] == '.') continue;

    // only list subdirectories and files

    if (!DIR_IS_FILE_OR_SUBDIR(&p)) continue;

    // print any indent spaces

    client.print("<li><a href=\"");

    for (uint8_t i = 0; i < 11; i++) {

        if (p.name[i] == ' ') continue;

        if (i == 8) {

            client.print('.');

        }

        client.print((char)p.name[i]);

    }

    client.print(">");

    // print file name with possible blank fill

    for (uint8_t i = 0; i < 11; i++) {

        if (p.name[i] == ' ') continue;

        if (i == 8) {

            client.print('.');

        }

        client.print((char)p.name[i]);

    }

```

```

}

client.print("</a>");

if (DIR_IS_SUBDIR(&p)) {
    client.print('/');
}

// print modify date/time if requested
if (flags & LS_DATE) {
    root.printFatDate(p.lastWriteDate);
    client.print(' ');
    root.printFatTime(p.lastWriteTime);
}

// print size if requested
if (!DIR_IS_SUBDIR(&p) && (flags & LS_SIZE)) {
    client.print(' ');
    client.print(p.fileSize);
}

client.println("</li>");
}

client.println("</ul>");
}

// How big our line buffer should be. 100 is plenty!
#define BUFSIZ 100

//*****
*****

// the loop routine runs over and over again forever:
void loop()
{
    char clientline[BUFSIZ];

```

```

int index = 0;

EthernetClient client = server.available();
if (client) {
  // an http request ends with a blank line
  boolean current_line_is_blank = true;

  // reset the input buffer
  index = 0;

  while (client.connected()) {
    if (client.available()) {
      char c = client.read();

      // If it isn't a new line, add the character to the buffer
      if (c != '\n' && c != '\r') {
        clientline[index] = c;
        index++;

        // are we too big for the buffer? start tossing out data
        if (index >= BUFSIZ)
          index = BUFSIZ - 1;

        // continue to read more data!
        continue;
      }

      // got a \n or \r new line, which means the string is done
      clientline[index] = 0;

      // Print it out for debugging
      Serial.println(clientline);
    }
  }
}

```



```

// Look for substring such as a request to get the root file
if (strstr(clientline, "GET / ") != 0) {
    // send a standard http response header
    client.println("HTTP/1.1 200 OK");
    client.println("Content-Type: text/html");
    client.println();
    client.println("<h2><b>WATER FLOW MEASURING SYSTEM</b></h2>");
    client.println();
    client.println("<h2><b>University of Ioannina</b></h2>");
    client.println();
    client.println("</h2>Physics Department</h2>");

    // print all the files, use a helper to keep it clean
    //client.println("<body style=background-color:green>");
    client.println("<body>");

    // print all the files, use a helper to keep it clean
    client.println("<h2>Files:</h2>");
    // client.println("<form action=\"\"><fieldset><legend><b>Files:</b></form>");
    ListFiles(client, LS_SIZE);
    client.println("</legend></fieldset>");

} else if (strstr(clientline, "GET /") != 0) {
    // this time no space after the /, so a sub-file!
    char *filename;

    filename = clientline + 5; // look after the "GET /" (5 chars)
    // a little trick, look for the " HTTP/1.1" string and
    // turn the first character of the substring into a 0 to clear it out.
    (strstr(clientline, " HTTP"))[0] = 0;

```

```

// print the file we want
Serial.println(filename);
if (! file.open(&root, filename, O_READ)) {
    client.println("HTTP/1.1 404 Not Found");
    client.println("Content-Type: text/html");
    client.println();
    client.println("<h2>File Not Found!</h2>");
    break;
}

Serial.println("Opened!");
client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/plain");
client.println();

int16_t c;
while ((c = file.read()) > 0) {
    // uncomment the serial to debug (slow!)
    //Serial.print((char)c);
    client.print((char)c);
}
file.close();
} else {
    // everything else is a 404
    client.println("HTTP/1.1 404 Not Found");
    client.println("Content-Type: text/html");
    client.println();
    client.println("<h2>File Not Found!</h2>");
}
break;

```

```

    }
}
// give the web browser time to receive the data
delay(1);
client.stop();
}
}

```

## Π2.2 Κώδικας υποσυστήματος προσδιορισμού της στάθμης του νερού του ποταμού

Το πρόγραμμα αυτό αναπτύχθηκε και φορτώθηκε στο υποσύστημα προσδιορισμού της στάθμης του ποταμού. Σκοπός του είναι να λαμβάνει την τιμή της απόστασης του αισθητήρα υπερήχων από την επιφάνεια του νερού και αφαιρώντας από την απόσταση αισθητήρα – πυθμένα να υπολογίζει το ύψος της στάθμης του νερού. Στη συνέχεια αποστέλλει την τιμή της στάθμης υπο μορφή σήματος 4-20mA στο υποσύστημα προσδιορισμού της παροχής. Στην αρχή του προγράμματος ορίζονται οι καθολικές (global) μεταβλητές οι οποίες χρησιμοποιούνται στο πρόγραμμα. Στην ρουτίνα setup() που ακολουθεί ορίζεται αρχικά ο ρυθμός μετάδοσης των δεδομένων (Serial.begin(9600)) για την προβολή των εξόδων του προγράμματος και στη συνέχεια γίνεται ορισμός των ακροδεκτών που χρησιμοποιούνται, ως εισόδου η εξόδου. Έπειτα εκτελείται η ρουτίνα loop() όπου δηλώνονται οι λειτουργίες που επαναλαμβάνονται κατά την εκτέλεση του κώδικα σε έναν ατέρμονα βρόχο. Ειδικότερα ενεργοποιείται ο ακροδέκτης σκανδαλισμού (trigger) του αισθητήρα υπερήχων έτσι ώστε να σταλεί μια δέσμη παλμών υπερήχων κάθετα στην επιφάνεια του νερού του ποταμού. Η ανακλώμενη από τη στάθμη του νερού δέσμη δημιουργεί ένα παλμό στον ακροδέκτη echo του αισθητήρα. Ο παλμός είναι ανάλογος της απόστασης απόσταση αισθητήρα – στάθμης. Η τιμή της απόστασης, αφαιρείται μέσω του μικροελεγκτή του συστήματος, από την απόσταση αισθητήρα – πυθμένα η οποία είναι γνωστή και ίση με 6.32m. Έτσι υπολογίζεται το βάθος του ποταμού σε cm και μετατρέπεται από το πρόγραμμα σε αριθμό των 12bits. Αφού αποθηκευτεί σε πίνακα, αποστέλλεται κάθε bit στον αντίστοιχο ακροδέκτη του μετατροπέα ψηφιακού σήματος σε αναλογικό, από τον οποίο έχουμε έξοδο της στάθμης του νερού σε τιμή 4 έως 20mA. Κάποιες από τις συναρτήσεις που καλούνται στο κυρίως πρόγραμμα του κώδικα είναι:

- digitalWrite(pin,pinstatus)

Με την εν λόγω συνάρτηση τίθεται η κατάσταση pinstatus (High ή Low) στο συγκεκριμένο ψηφιακό pin

- bitRead(num, i)

Διαβάζει το bit i από τον αριθμό num

- pulseIn(Pin, HIGH)

με την συνάρτηση ελέγχεται η κατάσταση ενός παλμού και μετράται ο χρόνος για τον οποίο είναι high. Επιστρέφεται η διάρκεια του παλμού σε μικροδευτερόλεπτα.

```
int numpins = 12;
int pins[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 };
int Pin[] = { 33, 32, 31, 30, 29, 28, 27, 26, 25, 24, 23, 22 };

const int Pin0 = 22;    // pin connected to digital pin 22
const int Pin1 = 23;
const int Pin2 = 24;
const int Pin3 = 25;
const int Pin4 = 26;
const int Pin5 = 27;
const int Pin6 = 28;
const int Pin7 = 29;
const int Pin8 = 30;
const int Pin9 = 31;
const int Pin10 = 32;
const int Pin11 = 33;
const int Pin12 = 34;
const int clk = 38 ;

////////////////////////////////////
```

```

#define initPin 8      // the SRF05's init pin
#define echoPin 9     // the SRF05's echo pin

unsigned long pulseTime = 0; // variable for reading the pulse

int old_distance, distance;

// initialize the library with the numbers of the interface pins
/////////////////////////////////////////////////////////////////

void setup() {
  Serial.begin(9600);

  // make the init pin an output:
  pinMode(initPin, OUTPUT);
  // make the echo pin an input:
  pinMode(echoPin, INPUT);
  pinMode(Pin0, OUTPUT);
  pinMode(Pin1, OUTPUT);
  pinMode(Pin2, OUTPUT);
  pinMode(Pin3, OUTPUT);
  pinMode(Pin4, OUTPUT);
  pinMode(Pin5, OUTPUT);
  pinMode(Pin6, OUTPUT);
  pinMode(Pin7, OUTPUT);
  pinMode(Pin8, OUTPUT);
  pinMode(Pin9, OUTPUT);
  pinMode(Pin10, OUTPUT);
  pinMode(Pin11, OUTPUT);
  pinMode(Pin12, OUTPUT);
  pinMode(clk, OUTPUT);
}

/////////////////////////////////////////////////////////////////

void loop() {

  digitalWrite(initPin, HIGH);

```

```

delayMicroseconds(10);
digitalWrite(initPin, LOW);

pulseTime = pulseIn(echoPin, HIGH);
distance = int(pulseTime/58);
// if (distance != old_distance){
// Serial.print("DISTANCE: ");
// Serial.print(distance, DEC);
// Serial.println(" cm");
// }
// old_distance = distance;
delay(1000);
int y;    // vathos potamou
int z = 600; //apostasi aisthitira-patou
y = z - distance;
Serial.println(y);

////////////////////////////////////
float L = 6.8 ;    // klish ths eytheias// gia metaropi tou vathous "y" se 0-4095 (12bits)
int x = y * L;
int num = x;      // Get num from DISTANCE SENSOR
digitalWrite(clk, LOW);
delay(10);
for (int i=0; i<numpins; i++) { // convert decimal to bits
  byte state = bitRead(num, i);
digitalWrite(pins[i], state);
digitalWrite(Pin[i] , state) ; // Write to digital pins
  delay(1000);
}

digitalWrite(clk, HIGH);
delay(100);
}

```

## Π2.3 Κώδικας συστήματος μέτρησης της τρέχουσας ταχύτητας νερού

Το πρόγραμμα αυτό φορτώθηκε στον μικροελεγκτή του συστήματος μέτρησης της τρέχουσας ταχύτητας του νερού Vernier με σκοπό να λαμβάνει την εξερχόμενη τάση από την προπέλα και να υπολογίζει την ταχύτητα του νερού σε m/sec. Κατόπιν αυτή η τιμή αποστέλλεται στην οθόνη LCD για ανάγνωση. Αρχικά στο πρόγραμμα δηλώνεται η βιβλιοθήκη για την οθόνη υγρών κρυστάλλων. Ακολουθεί η ρουτίνα `setup()` όπου αρχικά ορίζεται ο ρυθμός μετάδοσης των δεδομένων (`Serial.begin(9600)`) για την προβολή των εξόδων του προγράμματος, για λόγους αποσφαλμάτωσης, στη σειριακή θύρα του περιβάλλοντος ανάπτυξης IDE του Arduino και στη συνέχεια δηλώνονται οι στήλες και οι γραμμές της οθόνης LCD. Στη συνέχεια, εκτελείται η ρουτίνα `loop()` όπου δηλώνονται οι λειτουργίες που επαναλαμβάνονται κατά την εκτέλεση του κώδικα σε έναν ατέρμονα βρόχο (`loop`). Στο τμήμα `loop()` του κώδικα, δηλώνεται μια μεταβλητή η οποία είναι αποτέλεσμα της βαθμονόμησης του αισθητήρα. Στη συνέχεια γίνεται ανάγνωση της τιμής που λαμβάνει ο αισθητήρας (1-5V) η οποία διαβάζεται από τον ADC του μικροελεγκτή και μετατρέπεται σε τιμή 0-1023 bits. Μετά αυτή η τιμή πολλαπλασιάζεται με την μεταβλητή βαθμονόμησης και προκύπτει η τιμή της ταχύτητας σε m/sec. Τέλος αυτή η τιμή αποστέλλεται στην οθόνη για ανάγνωση. Οι κύριες συναρτήσεις που χρησιμοποιούνται στο λογισμικό είναι:

- [analogRead](#)

Επιστρέφει έναν ακέραιο από 0 έως 1023, ανάλογα με την τάση της αναλογικής εισόδου στην κλίμακα 0 ως  $V_{ref}$ .

- [lcd.print](#)

Τυπώνει το κείμενο στην οθόνη LCD που είναι διασυνδεδεμένη στην πλακέτα Arduino Mega 1280 του συστήματος μέτρησης της ταχύτητας του νερού.

```
// include the library code:
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  Serial.begin(9600);

  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
```

```

}

void loop() {

float L = 0.0049; // L=y/x //0.00371; // flow meter calibration

float x;

x = analogRead(0); // read pin A0

float y;

y = x * L;

//Serial.println(y);

// Turn on the display:

lcd.display();

lcd.print(y);

lcd.print("m/s");

delay(500); // wait to see results

lcd.clear(); // clear LCD

delay(500); }

```

## Π2.4 Κώδικας για τον έλεγχο της κάρτας 4-20mA σε 1-5V

```

void setup() {

analogReference(DEFAULT);

// initialize serial communication at 9600 bits per second:

Serial.begin(9600);

}

// the loop routine runs over and over again forever:

void loop() {

// sensor

int sensorValue = analogRead(A0);

Serial.println("---vathos---");

Serial.print("Counts: ");

Serial.println(sensorValue);

// Convert the analog reading (which goes from 0 - 1023) to a voltage (0 - 5V):

```



```

float voltage = sensorValue * (5.0 / 1023.0);

// print out the value you read:
Serial.print("Voltage: ");
Serial.println(voltage);
Serial.println("-----");
delay(1000);
}

```

## Π2.5 Κώδικας για τον έλεγχο της κάρτας μετατροπής ψηφιακού σήματος σε πρότυπο 4-20mA

```

int numpins = 12;

int pins[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 };

int Pin[] = { 33, 32, 31, 30, 29, 28, 27, 26, 25, 24, 23, 22 };

const int Pin0 = 22;    // pin connected to digital pin 22
const int Pin1 = 23;
const int Pin2 = 24;
const int Pin3 = 25;
const int Pin4 = 26;
const int Pin5 = 27;
const int Pin6 = 28;
const int Pin7 = 29;
const int Pin8 = 30;
const int Pin9 = 31;
const int Pin10 = 32;
const int Pin11 = 33;
const int Pin12 = 34;
const int clk = 38 ;

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// initialize the library with the numbers of the interface pins
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

```

```

void setup() {
Serial.begin(9600);

pinMode(Pin0, OUTPUT);
pinMode(Pin1, OUTPUT);
pinMode(Pin2, OUTPUT);
pinMode(Pin3, OUTPUT);
pinMode(Pin4, OUTPUT);
pinMode(Pin5, OUTPUT);
pinMode(Pin6, OUTPUT);
pinMode(Pin7, OUTPUT);
pinMode(Pin8, OUTPUT);
pinMode(Pin9, OUTPUT);
pinMode(Pin10, OUTPUT);
pinMode(Pin11, OUTPUT);
pinMode(Pin12, OUTPUT);
pinMode(clk, OUTPUT);
}

void loop() {
int x = 4095;      //edw vazoume thn timh pou elegxoume 0 - 4095
int num = x;      // Get num from DISTANCE SENSOR
digitalWrite(clk, LOW);
delay(10);

for (int i=0; i<numpins; i++) { // convert decimal to bits
byte state = bitRead(num, i);
digitalWrite(pins[i], state);
// Serial.print(state);
digitalWrite(Pin[i] , state); // Write to digital pins
delay(1000);

int outputValue = analogRead(A1);
// Serial.println(outputValue);

float voltage = outputValue * (5.0 / 1023.0);

Serial.println(x);

Serial.print("Voltage(V): ");

```

```

Serial.println(voltage);

Serial.print("current(A): ");

float i = voltage / 50;

Serial.println(i);

Serial.println("-----");

}

digitalWrite(clk, HIGH);

delay(700);

Serial.print("/");

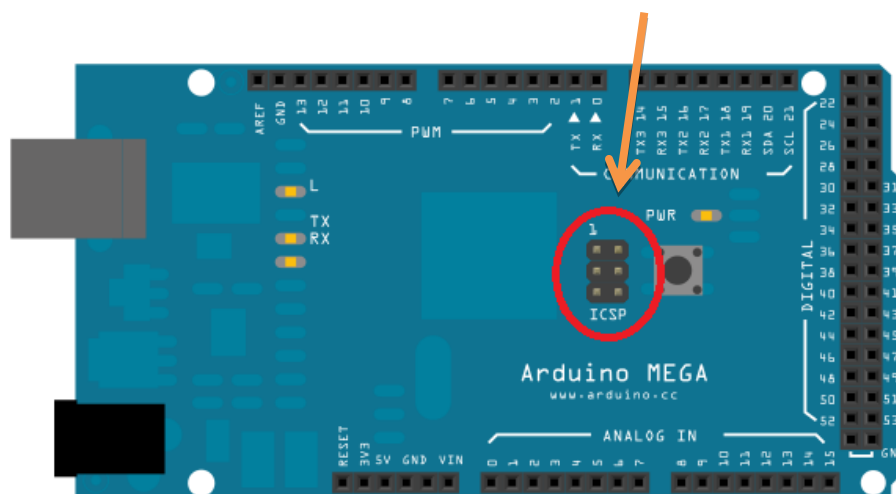
}

```

## Π2.6 Διαδικασία φόρτωσης του Bootloader στον μικροελεγκτή ATmega 2560 του Arduino Mega 2560

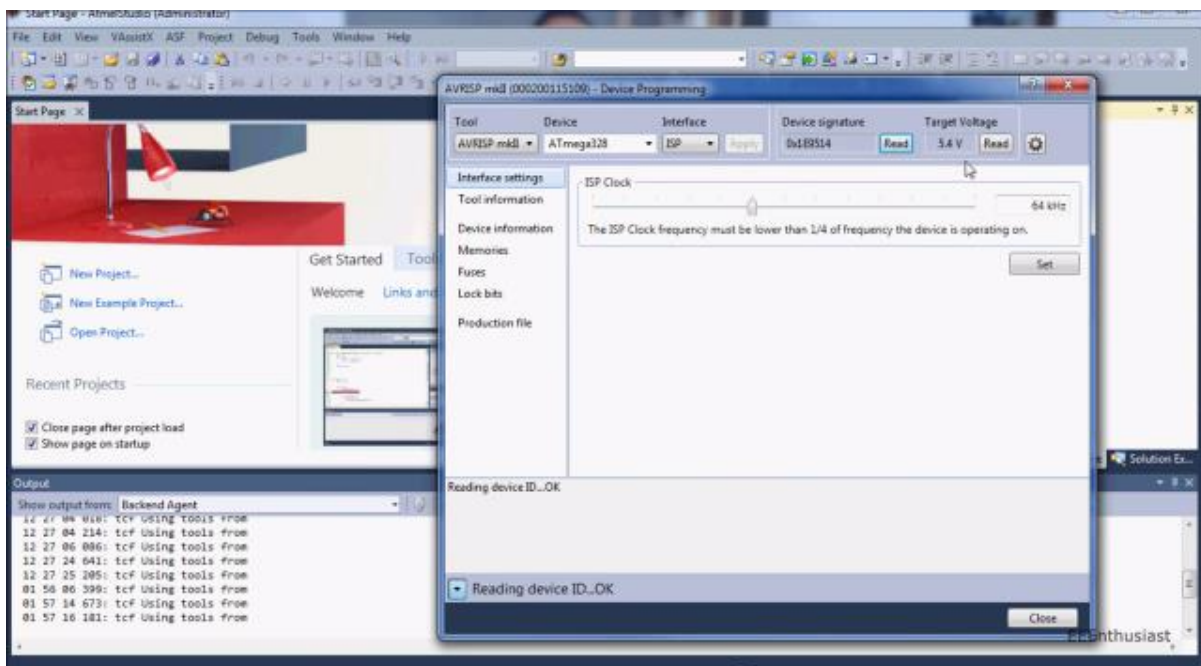
Για την φόρτωση του Bootloader στον μικροελεγκτή χρησιμοποιείται ο προγραμματιστής AVR JTAGICE MKII και το πρόγραμμα Atmel studio 6 με την εξής διαδικασία:

1. Συνδέεται στον ηλεκτρονικό υπολογιστή το σύστημα AVR JTAGICE MKII με καλώδιο USB. Τα 6 καλώδια του JTAGICE MKII με τους αντίστοιχους ακροδέκτες της θύρας ISP (Σχήμα 75) του μικροελεγκτή. Ενεργοποιείται το JTAGICE MKII, και ο μικροελεγκτής τροφοδοτείται (τάση τροφοδοσίας 5V).



Σχήμα 75. ISP pins του μικροελεγκτή της πλακέτας Arduino Mega

2. Στο λογισμικό Atmel Studio 6 επιλέγεται στην παλέτα εργαλείων, η επιλογή Tools και στη συνέχεια η επιλογή Device programming όπως φαίνεται στο Σχήμα 76.



Σχήμα 76. Το περιβάλλον του προγράμματος Atmel Studio 6

3. Επιλέγονται :	Tools	AVRISP mkII
	Device	ATmega256(στην δική μας περίπτωση)
	Interface	ISP

Κλίκ στο Apply και read στο Device signature.

4. Επιλέγεται Memories και άνοιγμα του αρχείου stk500boot\_v2\_mega2560.hex που βρίσκεται στο φάκελο: My computer > Arduino > hardware.

5. Στην επιλογή Fuses επιλέγεται :	EXTENDED	0xFD
	HIGH	0xD8
	LOW	0xC6

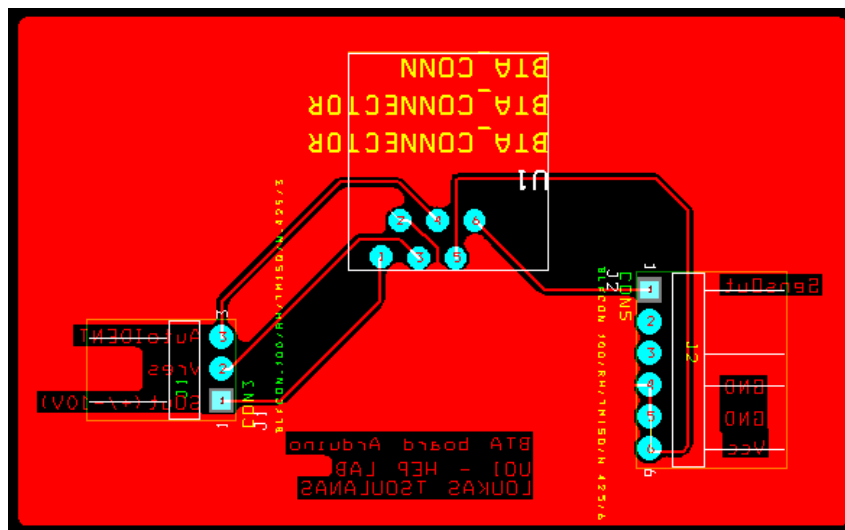
Κλίκ στο program και ο bootloader φορτώνεται στη μνήμη του μικροελεγκτή.

## Δημιουργία και τύπωση των πλακετών (PCB)

### Π3.1 Τυπωμένα κυκλώματα πλακετών

Στο συγκεκριμένο Παράρτημα παρατίθενται όλα τα σχέδια των τυπωμένων κυκλωμάτων των ηλεκτρονικών καρτών που κατασκευάστηκαν καθώς και τα υλικά τους.

#### Π3.1.1 Κάρτα προσαρμογής βύσματος τηλεπικοινωνιών βρετανικού τύπου σε Arduino



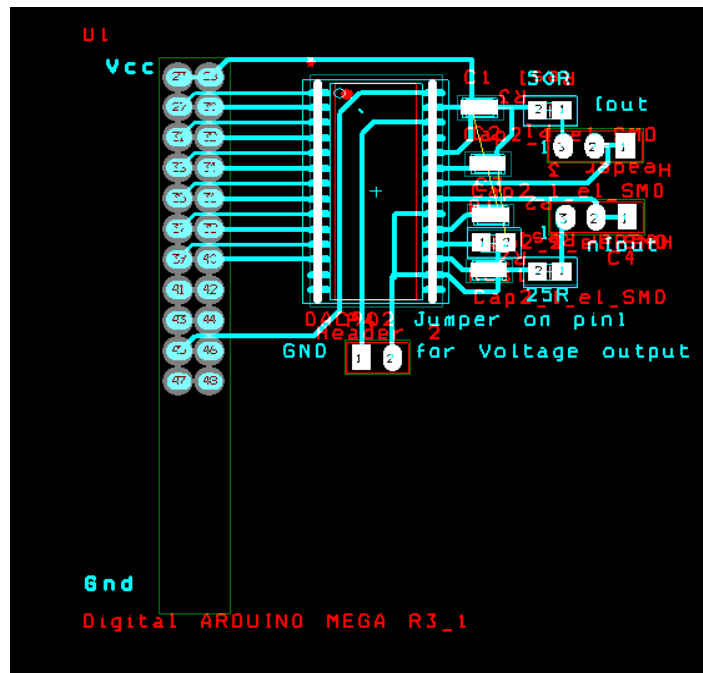
Σχήμα 77. Σχέδιο κάρτας προσαρμογής βύσματος τηλεπικοινωνιών βρετανικού τύπου σε Arduino

Ο Πίνακας 10 παρουσιάζει τα στοιχεία της κάρτας προσαρμογής βύσματος τηλεπικοινωνιών βρετανικού τύπου σε Arduino.

Όνομα στοιχείου	Τιμή - Τύπος	Περιγραφή
J1	Header 3	3xΑρσενικός ακροδέκτης
J2	Header 6	6xΑρσενικός ακροδέκτης
U1	British telecom female adapter	Θυληκός ακροδέκτης τηλεπικοινωνιών βρετανικού τύπου

Πίνακας 10. Στοιχεία κάρτας προσαρμογής βύσματος τηλεπικοινωνιών βρετανικού τύπου σε Arduino

### Π3.1.2 Κάρτα μετατροπής ψηφιακού σήματος σε πρότυπο 4-20mA



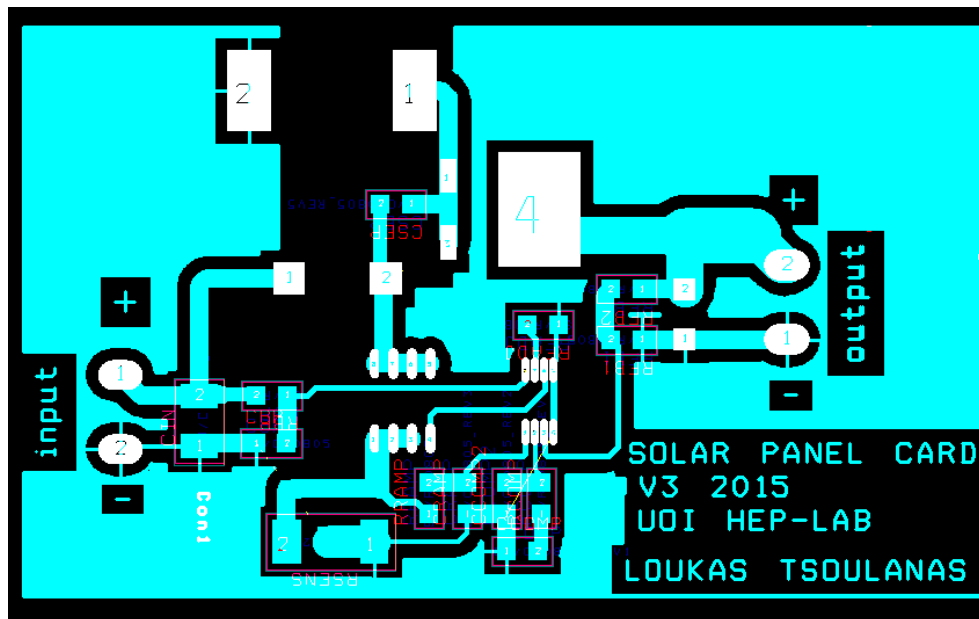
Σχήμα 78. Σχέδιο κάρτας μετατροπής ψηφιακού σήματος σε πρότυπο 4-20mA

Ο Πίνακας 11 παρουσιάζει τα στοιχεία της κάρτας μετατροπής ψηφιακού σήματος σε πρότυπο 4-20mA.

Όνομα στοιχείου	Τιμή - Τύπος	Περιγραφή
R1	2K	Αντιστάτης
R2	25R	Αντιστάτης
R3	50R	Αντιστάτης
C1	1N	Πυκνωτής
C2	100n	Πυκνωτής
C3	100n	Πυκνωτής
C4	100n	Πυκνωτής
J1	Header 3	3xΑρσενικός ακροδέκτης
J2	Header 3	3xΑρσενικός ακροδέκτης
J3	Header 2	2xΑρσενικός ακροδέκτης
U1	DAC 902	Μετατροπέας αναλογικού σήματος σε ψηφιακό
U2	Header 2x11	2x11 Αρσενικός ακροδέκτης

Πίνακας 11. Στοιχεία κάρτας μετατροπής ψηφιακού σήματος σε αναλογικό (4-20mA)

### Π3.1.3 Κάρτα προσαρμογής του φωτοβολταϊκού πάνελ στη μπαταρία



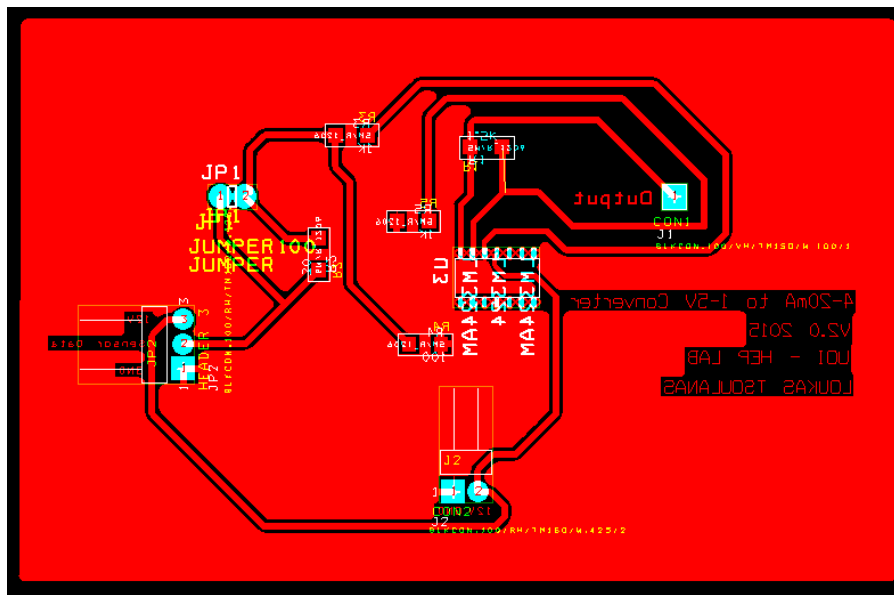
Σχήμα 79. Σχέδιο κάρτας προσαρμογής του φωτοβολταϊκού πάνελ στη μπαταρία

Στον Πίνακα 12 φαίνονται τα στοιχεία της κάρτας προσαρμογής του φωτοβολταϊκού πάνελ στη μπαταρία.

Όνομα στοιχείου	Τιμή - Τύπος	Περιγραφή
JP1	Header 2	2xΑρσενικός ακροδέκτης
JP2	Header 2	2xΑρσενικός ακροδέκτης
CIN	22uF	Πυκνωτής
CBP1	100nF	Πυκνωτής
CCOMP1	3.3nf	Πυκνωτής
CCOMP2	330nF	Πυκνωτής
CSEP1	4.7uF	Πυκνωτής
CRAMP1	1nF	Πυκνωτής
COUT1	150uF	Πυκνωτής
RBP1	20R	Αντιστάτης
RCOMP1	1.4k	Αντιστάτης
RRAMP1	100R	Αντιστάτης
RFADJ1	53.6R	Αντιστάτης
RSENSE1	5mR	Αντιστάτης
RFB1	97.6K	Αντιστάτης
RFB2	10K	Αντιστάτης
LIN1	4.7uH	Πηνίο
LOUT1	47uH	Πηνίο
U1	LM3478MM	Switching regulator
M1	CSD18563Q5A	Mosfet
D1	PDS760-13	Δίοδος shottky

Πίνακας 12. Στοιχεία κάρτας προσαρμογής του φωτοβολταϊκού πάνελ στη μπαταρία

### Π3.1.4 Κάρτα μετατροπής 4-20mA σε 1-5V



Σχήμα 80. Σχέδιο κάρτας μετατροπής 4-20mA σε 1-5V

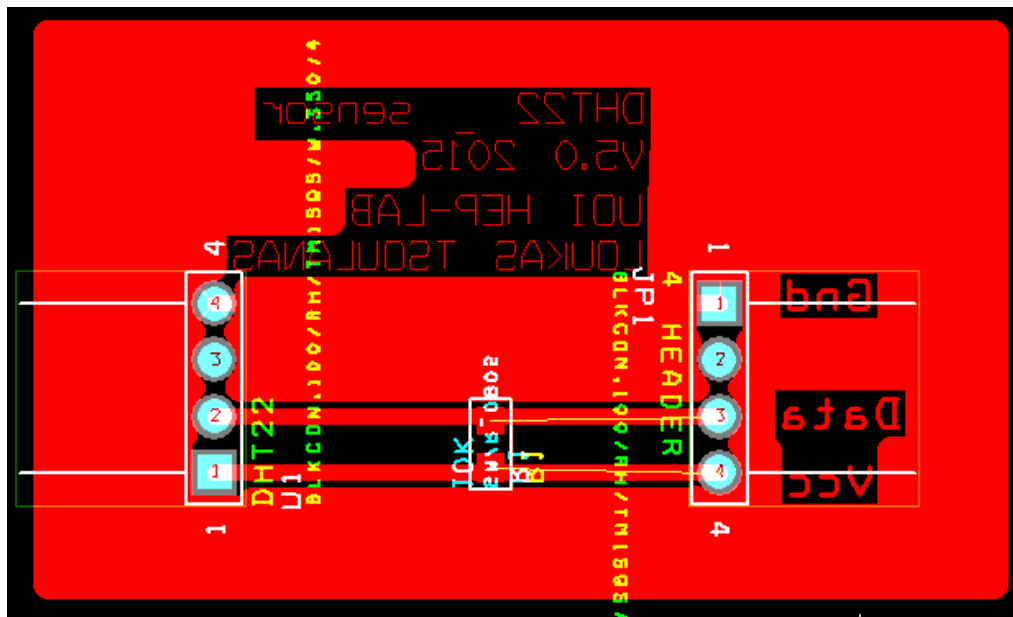
Στον Πίνακα 12 φαίνονται τα στοιχεία της κάρτας μετατροπής 4-20mA σε 1-5V.

Όνομα στοιχείου	Τιμή - Τύπος	Περιγραφή
JP1	Jumper	1xΑρσενικός ακροδέκτης
JP2	Header 3	3xΑρσενικός ακροδέκτης
CON1	Connector 1	2xΑρσενικός ακροδέκτης
CON2	Connector 2	2xΑρσενικός ακροδέκτης
R1	1.5K	Αντιστάτης
R2	30R	Αντιστάτης
R3	1K	Αντιστάτης
R4	100R	Αντιστάτης
R5	1K	Αντιστάτης
U3	LM324AM	Τελεστικός Ενισχυτής

Πίνακας 13. Στοιχεία κάρτας μετατροπής 4-20mA σε 1-5V



### Π3.1.5 Κάρτα μέτρησης θερμοκρασίας - υγρασίας



Σχήμα 81. Σχέδιο κάρτας μέτρησης θερμοκρασίας - υγρασίας

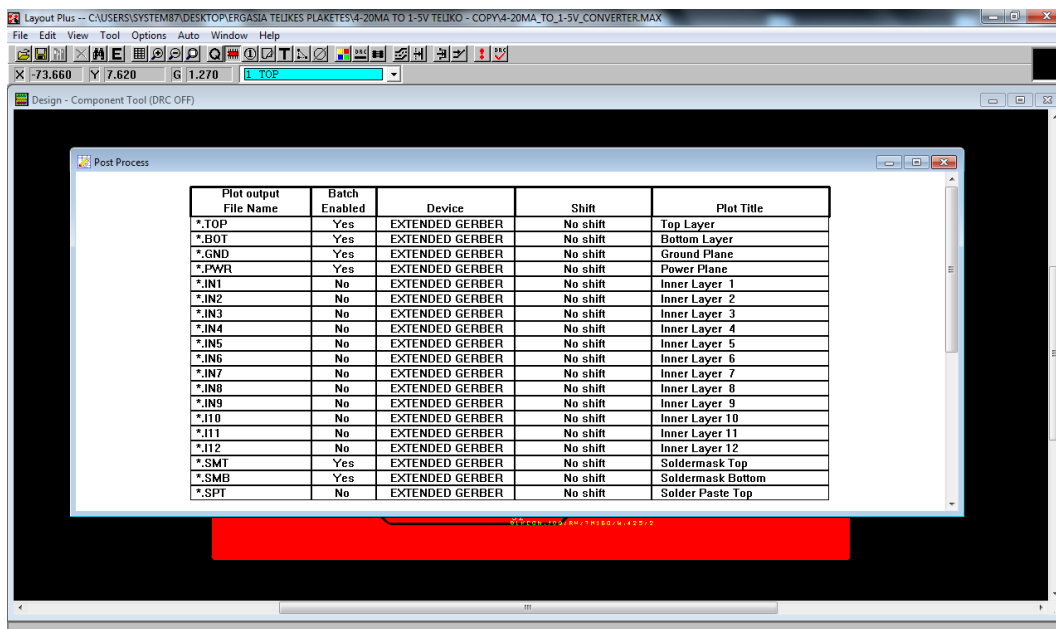
Στον Πίνακα 14 παρουσιάζονται τα στοιχεία της κάρτας μέτρησης θερμοκρασίας - υγρασίας

Όνομα στοιχείου	Τιμή - Τύπος	Περιγραφή
U1	DHT22	Αισθητήρας θερμοκρασίας-υγρασίας
R1	10K	Αντιστάτης
C1	100n	Πυκνωτής
JP1	Header 4	4xΑρσενικός ακροδέκτης

Πίνακας 14. Στοιχεία κάρτας μέτρησης θερμοκρασίας - υγρασίας

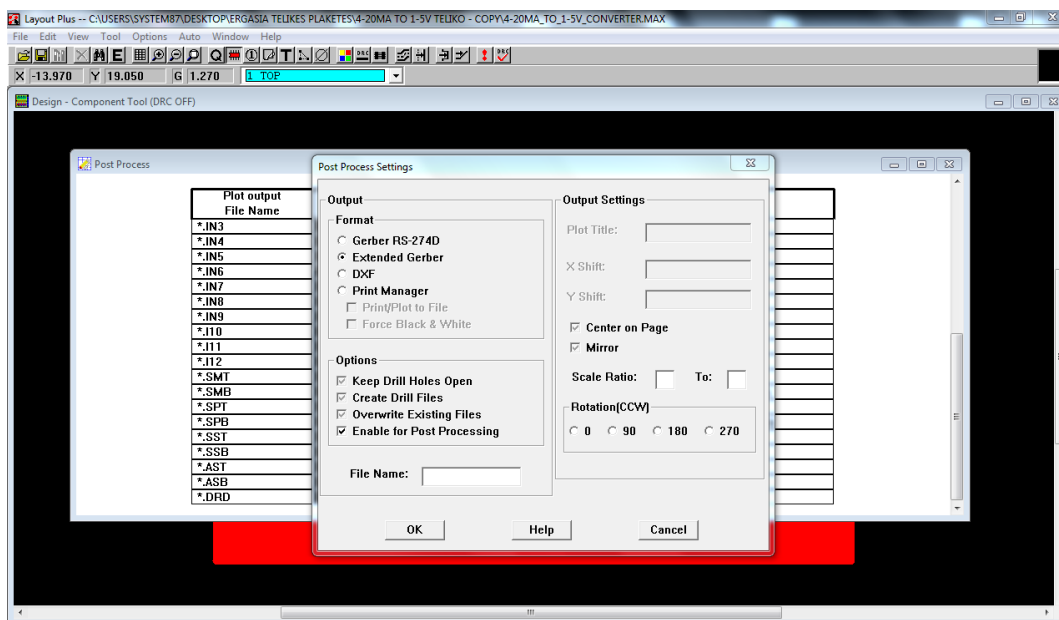
### Π3.2 Διαδικασία δημιουργίας αρχείου Gerber

Αφού ολοκληρώνεται το σχέδιο μιας πλακέτας στο πρόγραμμα Orcad Layout στη συνέχεια γίνεται εξαγωγή του αρχείου Gerber που είναι αρχείο που μπορεί να διαχειρισθεί από διάφορα λογισμικά. Έχει κατάληξη .GTD και είναι αρχείο εικόνας δύο διαστάσεων. Για τη δημιουργία του αρχείου επιλέγεται η επιλογή Gerber από την παλέτα του υποπρογράμματος Layout η επιλογή "Options" και "Post process settings". Τότε εμφανίζεται το παράθυρο με όλα τα επίπεδα σχεδίασης όπως φαίνεται στο Σχήμα 82.



Σχήμα 82. Παράθυρο με τα επίπεδα σχεδίασης που εμφανίζεται στο πρόγραμμα Orcad Layout

Στη συνέχεια τα επιλέγονται όλα τα αρχεία στην οθόνη και με δεξί click ανοίγει το παράθυρο post process settings στο οποίο επιλέγονται οι επιλογές “Extended Gerber”, “Enable for post processing” καθώς και “Keep Drill Holes Open”, “Create Drill Files” για τη δημιουργία των σπών στο αρχείο Gerber, όπως φαίνεται στο Σχήμα 83.



Σχήμα 83. Το παράθυρο post process settings στο πρόγραμμα Orcad Layout

Με δεξί click στο παράθυρο post process και επιλογή στο Run Batch, το αρχείο Gerber δημιουργείται και αποθηκεύεται στο φάκελο με το αρχείο .max και τα υπόλοιπα αρχεία του σχεδίου. Με το υποπρόγραμμα Layout μπορεί να ανοιχτεί το αρχείο Gerber, επιλέγοντας διαδοχικά Tools >> Gerber Tool>> Open και να εκτυπωθεί σε ριζόχαρτο ώστε να κατασκευαστεί η αντίστοιχη ηλεκτρονική πλακέτα.

### Απαραίτητες συνδέσεις - Καταναλώσεις

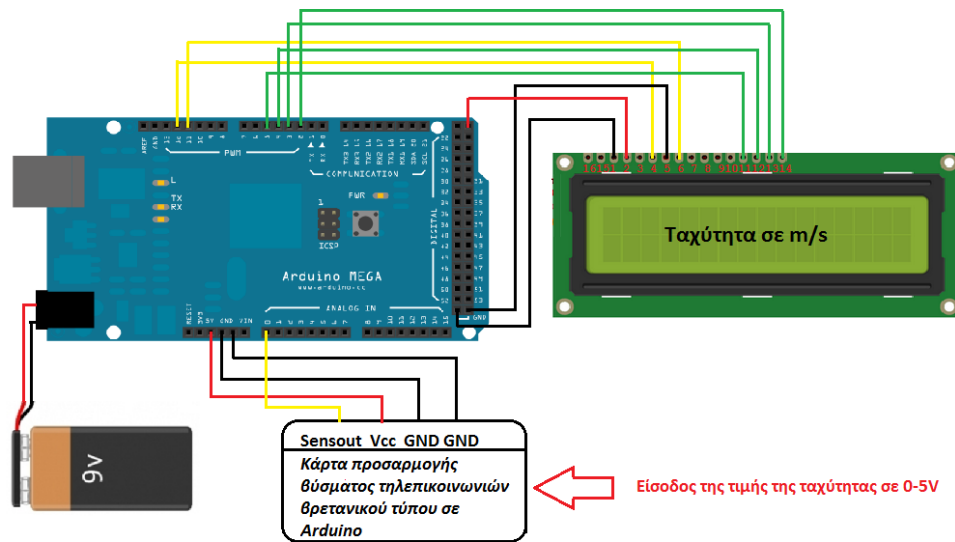
Σε αυτό το Παράρτημα παρατίθενται πίνακες με τις απαραίτητες συνδέσεις μεταξύ των ηλεκτρονικών πλακετών του συστήματος καθώς και τα σχηματικά διάγραμμα αυτών.

#### Π4.1 Ακροδέκτες συστήματος μέτρησης της τρέχουσας ταχύτητας του νερού

Πίνακας διασυνδέσεων :

Ακροδέκτης Arduino Mega 1280	Ακροδέκτης πλακέτας
5V	Vcc της κάρτας προσαρμογής βύσματος τηλ/ων βρετανικού τύπου σε Arduino
A0	SensOut της κάρτας προσαρμογής βύσματος τηλ/ων βρετανικού τύπου σε Arduino
0V	GND της κάρτας προσαρμογής βύσματος τηλ/ων βρετανικού τύπου σε Arduino
0V	GND της κάρτας προσαρμογής βύσματος τηλ/ων βρετανικού τύπου σε Arduino
0V	Pin 1 της οθόνης LCD
5V	Pin 2 της οθόνης LCD
PWM12	Pin 4 της οθόνης LCD
0V	Pin 5 της οθόνης LCD
PWM 11	Pin 6 της οθόνης LCD
PWM 5	Pin 11 της οθόνης LCD
PWM 4	Pin 12 της οθόνης LCD
PWM 3	Pin 13 της οθόνης LCD
PWM 2	Pin 14 της οθόνης LCD

Πίνακας 15. Πίνακας διασυνδέσεων συστήματος μέτρησης της τρέχουσας ταχύτητας του νερού



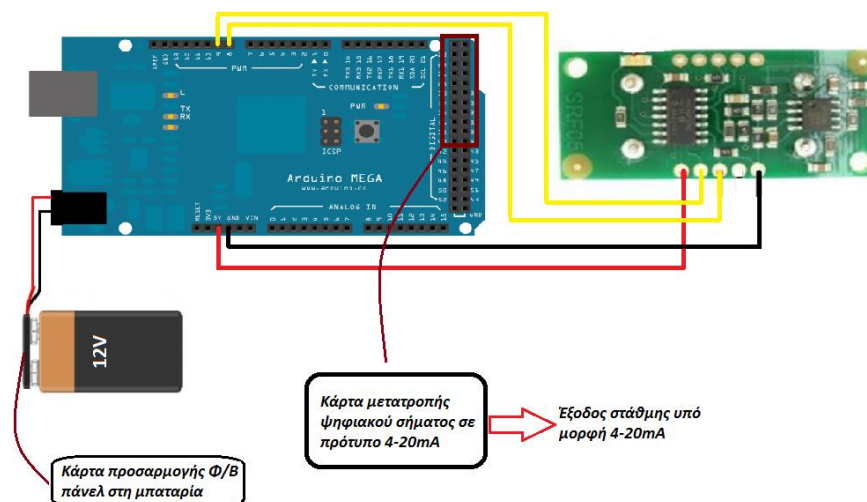
Σχήμα 84. Διάγραμμα διασυνδέσεων συστήματος μέτρησης της τρέχουσας ταχύτητας του νερού

## Π4.2 Ακροδέκτες υποσυστήματος προσδιορισμού της στάθμης του νερού του ποταμού

Πίνακας διασυνδέσεων :

Ακροδέκτης Arduino Mega 1280	Ακροδέκτης πλακέτας
5V	Vcc
0V	Gnd
PWM 8	Trigger
PWM 9	Echo

Πίνακας 16. Πίνακας διασυνδέσεων υποσυστήματος προσδιορισμού της στάθμης του ποταμού



Σχήμα 85. Διάγραμμα συνδέσεων υποσυστήματος προσδιορισμού της στάθμης του ποταμού

### Π4.3 Συνδέσεις υποσυστήματος υπολογισμού της στιγμιαίας παροχής νερού του ποταμού

Πίνακας συνδέσεων :

Ακροδέκτης Arduino Mega 2560	Ακροδέκτης πλακέτας
5V	Vcc της κάρτας μέτρησης θερμοκρασίας - υγρασίας
0V	Gnd της κάρτας μέτρησης θερμοκρασίας - υγρασίας
A0	Data της κάρτας μέτρησης θερμοκρασίας - υγρασίας
A1	Έξοδος της κάρτας μετατροπής 4-20mA σε 1-5V
SDA	SDA της κάρτας με το ρολόι πραγματικού χρόνου
SCL	SCL της κάρτας με το ρολόι πραγματικού χρόνου
5V	Vcc της κάρτας με το ρολόι πραγματικού χρόνου
0V	Gnd της κάρτας με το ρολόι πραγματικού χρόνου

Πίνακας 17. Πίνακας συνδέσεων υποσυστήματος υπολογισμού της στιγμιαίας παροχής νερού του ποταμού

### Π4.4 Καταναλώσεις Συστημάτων - Αισθητήρων

Οι καταναλώσεις των επιμέρους υποσυστημάτων που απαρτίζουν το σύστημα φαίνονται στον πίνακα 18.

Σύστημα	Κατανάλωση (mA)
Μέτρησης ταχύτητας ροής	14
Προσδιορισμού της στάθμης	12
Υπολογισμού στιγμιαίας παροχής	21
Αισθητήρας WL705	20

Πίνακας 18. Πίνακας καταναλώσεων Συστημάτων - Αισθητήρων



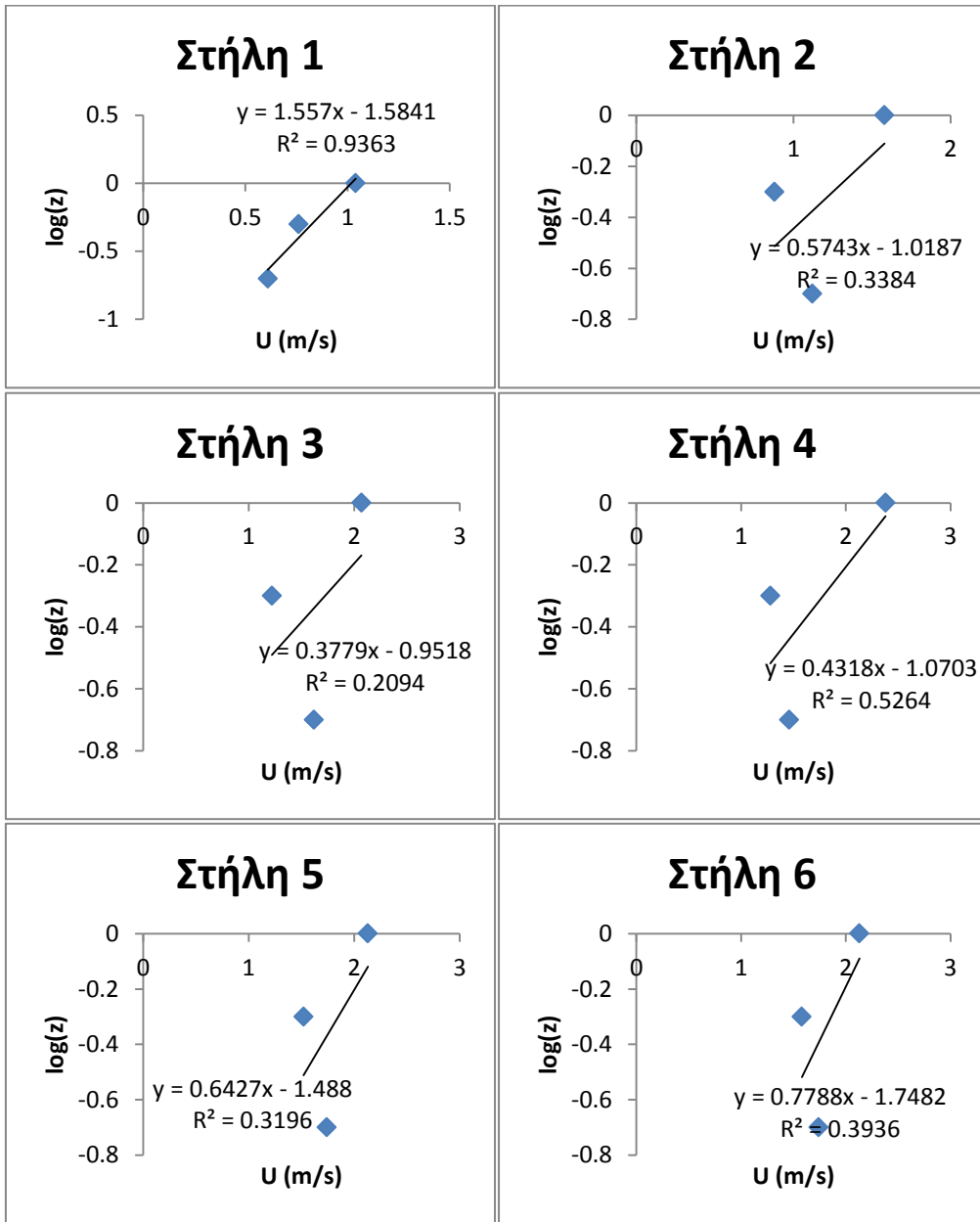
### **Ανάλυση της ταχύτητας του νερού σε κατακόρυφη τομή του ποταμού σε περιοχή όπου η κοίτη είναι σταθερή και η ταχύτητα εξαρτάται από τη θέση της επιφάνειας του νερού**

#### **Π5.1 Γραφικές αναπαραστάσεις $\log z=f(u)$**

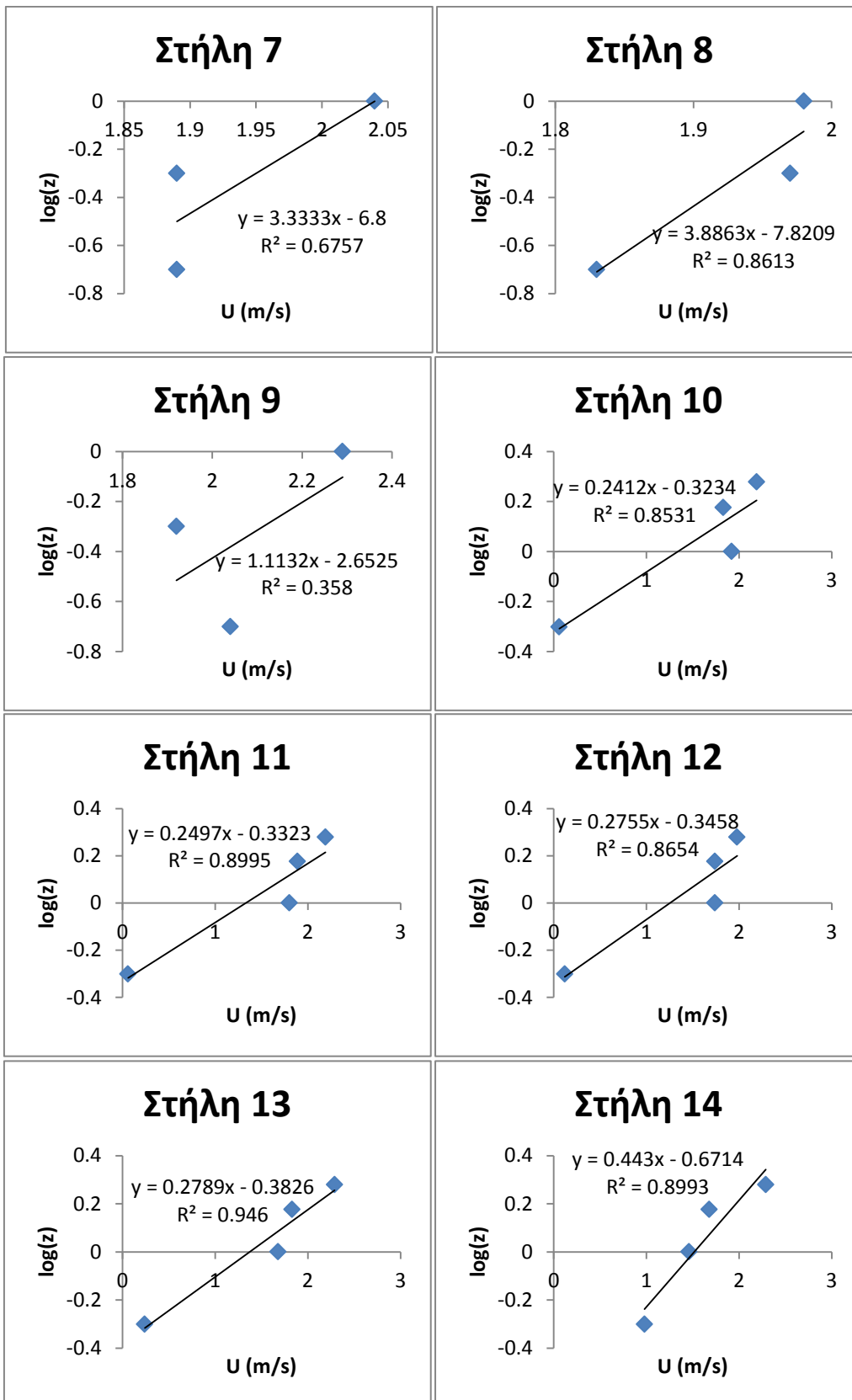
Στο παράρτημα αυτό παρατίθενται οι γραφικές παραστάσεις του λογαρίθμου του ύψους μέσα στο ποτάμι ( $\log z$ ), σε συνάρτηση με τη ταχύτητα του ποταμού ( $u$ ) στο ύψος αυτό, οι οποίες καθορίζονται από τη σχέση  $\log z_i = \alpha \cdot u_i(z) + b$ . Η διατομή της κοίτης έχει χωριστεί σε 30 κάθετες στήλες, άρα υπάρχουν 30 γραφικές παραστάσεις.

Παρατηρείται πως στις άκρες του ποταμού λόγω της τυρβώδους ροής από την ύπαρξη της όχθης, οι μετρήσεις της ταχύτητας του νερού παρουσιάζουν μεγαλύτερη διασπορά σε σχέση με αυτές που βρίσκονται στο κέντρο της διατομής της κοίτης. Ειδικότερα οι μετρήσεις της ταχύτητας στις γραφικές παραστάσεις 1 έως 9 και 29, 30 παρουσιάζουν την μεγαλύτερη διασπορά. Ένας ακόμα λόγος της αυξημένης διασποράς των μετρήσεων της ταχύτητας, είναι πως στα σημεία εκείνα (Σχήμα 53) το βάθος του ποταμού είναι μικρότερο, επομένως υπάρχει έντονη τριβή ανάμεσα στον πυθμένα και το νερό με αποτέλεσμα να δημιουργούνται δίνες, οι οποίες επηρεάζουν τις μετρήσεις της ταχύτητας.

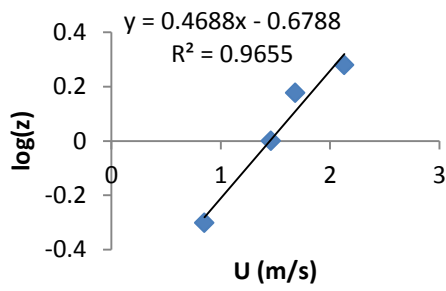
Παρακάτω παρουσιάζονται οι γραφικές παραστάσεις του λογαρίθμου του ύψους μέσα στο ποτάμι ( $\log z$ ), σε συνάρτηση με τη ταχύτητα του ποταμού ( $u$ ) στο ύψος αυτό για κάθε μία από τις 30 στήλες νερού (με την αντίστοιχη αρίθμηση) που έχει χωριστεί η διατομή της κοίτης.



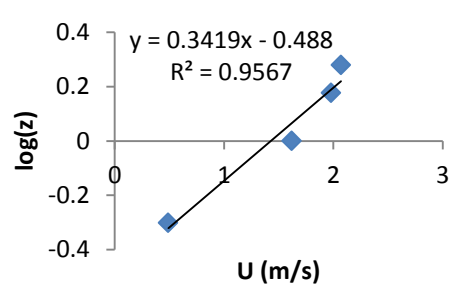




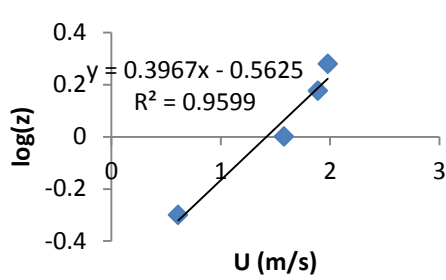
**Στήλη 15**



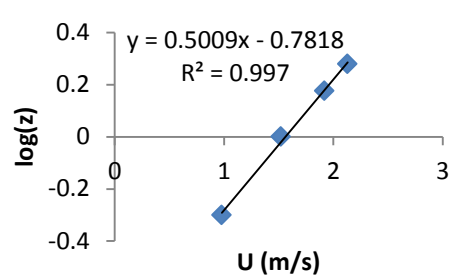
**Στήλη 16**



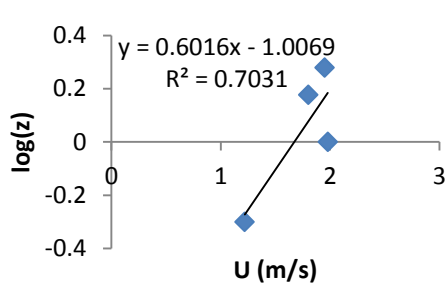
**Στήλη 17**



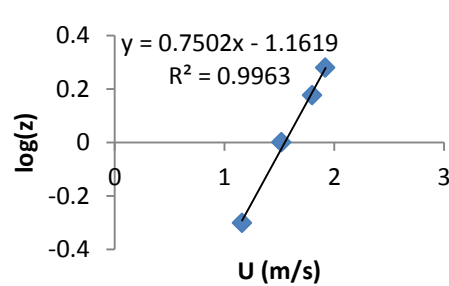
**Στήλη 18**



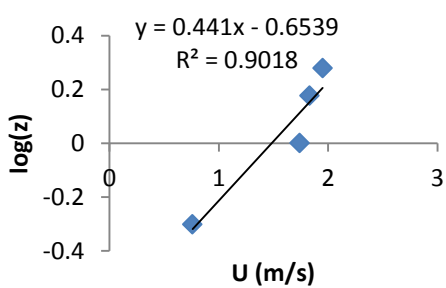
**Στήλη 19**



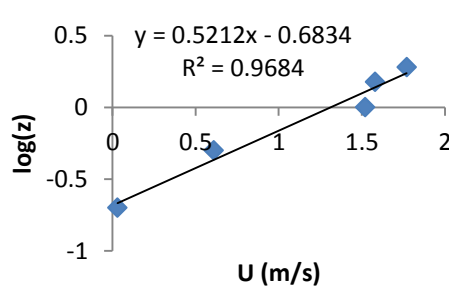
**Στήλη 20**



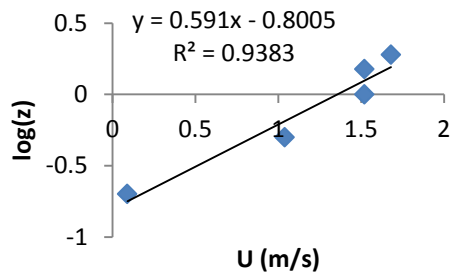
**Στήλη 21**



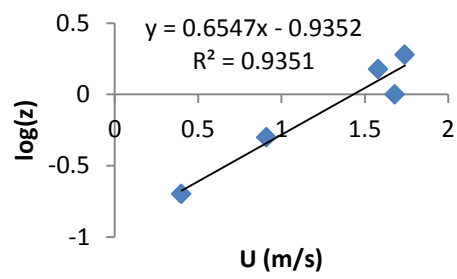
**Στήλη 22**



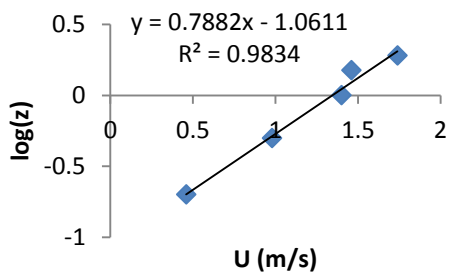
### Στήλη 23



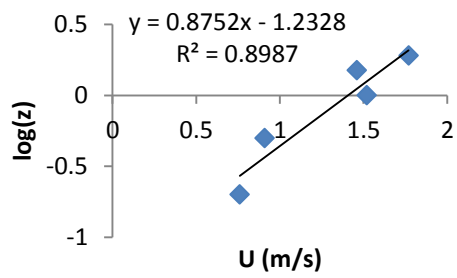
### Στήλη 24



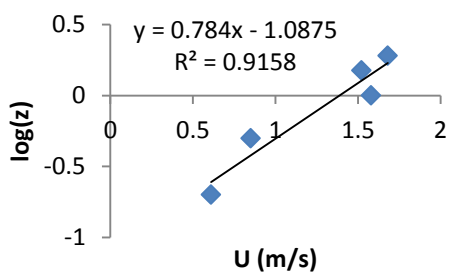
### Στήλη 25



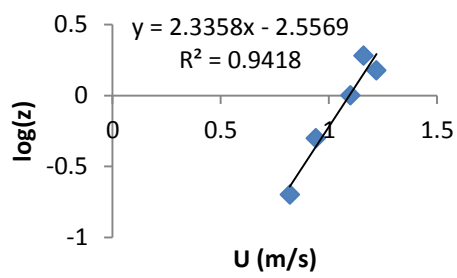
### Στήλη 26



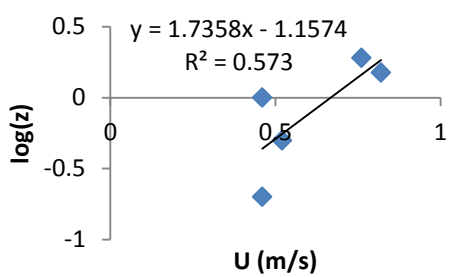
### Στήλη 27



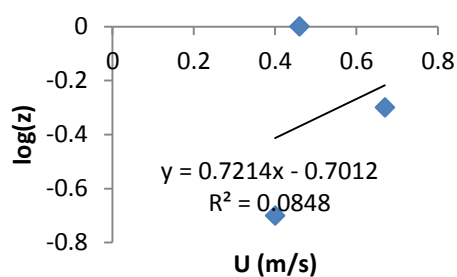
### Στήλη 28



### Στήλη 29



### Στήλη 30



## Π5.2 Συναρτήσεις $\log z=f(u)$

Στον Πίνακα 19 φαίνονται οι παράγοντες  $\alpha$  και  $b$  των εξισώσεων  $\log z_i = \alpha \cdot u_i(z) + b$ , για κάθε μια εκ των 30 στηλών νερού.

Στήλη	$\alpha$	$b$
1	1.56	-1.58
2	0.57	-1.02
3	0.38	-0.95
4	0.43	-1.07
5	0.64	-1.49
6	0.78	-1.75
7	3.33	-6.8
8	3.88	-7.82
9	1.11	-2.65
10	0.24	-0.32
11	0.25	-0.33
12	0.28	-0.35
13	0.28	-0.38
14	0.44	-0.67
15	0.47	-0.68
16	0.34	-0.49
17	0.39	-0.56
18	0.50	-0.78
19	0.60	-1
20	0.75	-1.16
21	0.44	-0.65
22	0.52	-0.68
23	0.59	-0.80
24	0.66	-0.94
25	0.79	-1.06
26	0.88	-1.23
27	0.78	-1.09
28	2.34	-2.55
29	1.74	-1.16
30	0.72	-0.70

Πίνακας 19. Παράγοντες των εξισώσεων  $\log z=f(u)$  των 30 στηλών νερού