



ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

**ΣΧΕΔΙΑΣΗ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΠΗΓΗΣ ΡΕΥΜΑΤΟΣ ΠΟΥ ΕΛΕΓΧΕΤΑΙ
ΑΠΟ ΜΙΚΡΟΕΛΕΓΚΤΗ ΚΑΙ Η/Υ**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΦΟΙΤΗΤΗΣ: ΑΣΗΜΙΔΗΣ ΒΑΣΙΛΕΙΟΣ (ΑΜ 3466)

ΕΠΙΛΒΕΠΩΝ: ΙΩΑΝΝΗΣ ΚΑΛΟΜΟΙΡΟΣ ΑΝΑΠΛ. ΚΑΘΗΓΗΤΗΣ

ΣΕΡΡΕΣ, ΙΑΝΟΥΑΡΙΟΣ 2017

Υπεύθυνη Δήλωση :

Βεβαιώνω ότι είμαι συγγραφέας αυτής της πτυχιακής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην πτυχιακή εργασία. Επίσης έχω αναφέρει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επίσης βεβαιώνω ότι αυτή η πτυχιακή εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για τις απαιτήσεις του προγράμματος σπουδών του Τμήματος Μηχανικών Πληροφορικής του Τ.Ε.Ι. Σερρών

Περίληψη

Σε αυτή την εργασία κατασκευάστηκε μια πηγή ρεύματος η οποία ελέγχεται από ένα Η/Υ με τη βοήθεια ενός μικροελεγκτή. Το κύκλωμα που υλοποιεί την πηγή ρεύματος είναι το κύκλωμα Improved Howland. Το κύκλωμα αυτό δεν χρησιμοποιεί τρανζίστορ, όπως τα συνηθισμένα κυκλώματα πηγών ρεύματος, αλλά εκμεταλλεύεται τις ιδιότητες ενός τελεστικού ενισχυτή. Χρησιμοποιώντας το πρόγραμμα Multisim 13 προσομοιώσαμε το Improved Howland για διάφορους τελεστικούς ενισχυτές και καταλήξαμε στο συμπέρασμα ότι τελεστικοί όπως ο OPA445AP είναι κατάλληλοι για την υλοποίηση της πηγής ρεύματος. Για να τροφοδοτήσουμε τον τελεστικό κατασκευάσαμε ένα κύκλωμα τροφοδοσίας που δίνει +43 και -43Volt στην έξοδό του. Για τον ψηφιακό έλεγχο του κυκλώματος Improved Howland χρησιμοποιήσαμε τον μικροελεγκτή της Microchip, PIC18F4550 που είναι ένας 8 bit 40 PIN μικροελεγκτής. Επίσης, χρησιμοποιήσαμε το DAC0832 που είναι ένα 8 bit DAC και επομένως μπορεί να δώσει 256 βήματα μιας καθορισμένης τάσης αναφοράς, σε ρεύμα στη έξοδό του. Το ρεύμα αυτό μετατρέπεται σε τάση και η τάση εισάγεται στο κύκλωμα Imp Howland και καθορίζει το ρεύμα στην έξοδο της πηγής ρεύματος. Τέλος ένα ολοκληρωμένο κύκλωμα MAX232 διαχειρίζεται την επικοινωνία μέσω της σειριακής θύρας με τον Η/Υ. Από την πλευρά του Η/Υ το πρόγραμμα LabView 2011 είναι υπεύθυνο για τον ψηφιακό έλεγχο της πηγής ρεύματος. Το κύκλωμα ψηφιακού ελέγχου σχεδιάστηκε με το πρόγραμμα Eagle 7.6.0.

Περιεχόμενα

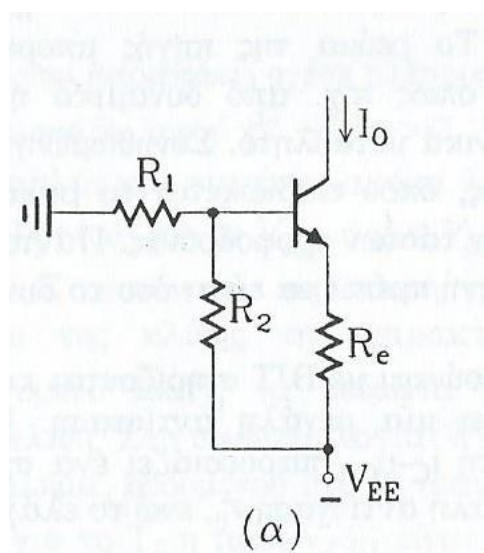
Περίληψη.....	3
1. Εισαγωγή	6
2. Τροφοδοτικό.....	11
2.1. Γέφυρα ανόρθωσης.....	11
2.2. Φίλτρο Πυκνωτή.....	12
2.3. Σταθεροποίηση με zener	13
3. Πηγή ρεύματος Howland.....	15
3.1. Μαθηματική Επίλυση Howland	15
3.2. Η βελτιωμένη πηγή ρεύματος Howland (Improved Howland Current Pump).....	17
3.3. Μια εισαγωγή στο Multisim και σχεδιασμός της πηγής ρεύματος improved Howland current pump	20
3.4. Προσομοίωση στο Multisim 13 της Improved Howland Current Pump.....	24
3.4.1. Προσομοίωση Improved Howland με τον OPA445AP	24
3.4.2. Προσομοίωση Improved Howland με τον LM741.....	31
3.5. Συμπεράσματα και τρόποι βελτιστοποίησης του κυκλώματος Improved Howland 35	
3.5.1. Ασύμμετρη τροφοδοσία του τελεστικού OPA445AP	36
3.5.2. Bootstrapping στο κύκλωμα Improved Howland Current Pump με τον OPA445AP.....	37
4. Hardware ψηφιακού ελέγχου της τάσης εισόδου του κυκλώματος improved Howland. 39	
4.1. Ο μικροελεγκτής PIC18F4550 της Microchip.	41
4.2. Προγραμματισμός του PIC18F4550.....	45
5. Περιγραφή Hardware και Software για επικοινωνία του PIC18F4550 με την LCD.	50
6. Περιγραφή Hardware και Software για την επικοινωνία του PIC με το MAX232 και τον Η/Υ(σειριακή επικοινωνία).....	60
6.1. Περιγραφή επικοινωνίας του PIC με το MAX232 και τη σειριακή θύρα.....	60
6.2. Περιγραφή επικοινωνίας του Η/Υ μέσω του προγράμματος LabView με τη σειριακή θύρα.....	65
7. Περιγραφή Hardware και Software για την επικοινωνία του PIC με το DAC0832 και μετατροπή του ρεύματος του DAC σε τάση.....	73
7.1. Περιγραφή Hardware για την επικοινωνία του PIC18F4550 με το DAC0832.....	73
7.2. Περιγραφή Software για την επικοινωνία του PIC18F4550 με το DAC0832	76

7.3. Μετατροπή του ρεύματος εξόδου του DAC0832 σε τάση	77
8. Παρουσίαση του τελικού κώδικα με τον οποίο προγραμματίσαμε τον PIC18F4550.....	79
9. Χαρακτηρισμός της πηγής ρεύματος improved Howland Current Pump.....	85
Βιβλιογραφία.....	90
Παράρτημα 1:Κατασκευή κυκλώματος ψηφιακού ελέγχου στο Eagle 7.6.0.....	91
Παράρτημα 2 Υπολογισμός κόστους υλικών.	100
Ευρετήριο Σχημάτων και Πινάκων.....	101

1. Εισαγωγή

Οι πηγές ρεύματος είναι διατάξεις που παρέχουν ρεύμα ανεξάρτητο από το μέγεθος του φορτίου. Χρησιμοποιούνται πολύ συχνά σε διακριτά ή ολοκληρωμένα κυκλώματα. Το ρεύμα της πηγής ρεύματος μπορεί να ρυθμίζεται με κάποιο τρόπο εξωτερικά πχ από κάποιο δυναμικό ή από μια αντίσταση και μπορεί ακόμη να είναι χρονικά μεταβλητό. Συνηθισμένη είναι η περίπτωση πηγής σταθερού ρεύματος όπου επιδιώκεται το ρεύμα να είναι σταθερό και ανεξάρτητο της θερμοκρασίας και των τάσεων τροφοδοσίας. Πάντως το ρεύμα που παρέχει η πηγή πρέπει να είναι σε κάθε περίπτωση όσο το δυνατό ανεξάρτητο από το φορτίο που τροφοδοτεί.

Εδώ θα δώσουμε μερικές υλοποιήσεις πηγών ρεύματος. Μια απλή πηγή ρεύματος υλοποιείται με ένα τρανζίστορ και ένα διαιρέτη τάσης.



Σχήμα 1-1 Πηγή ρεύματος με τρανζίστορ και διαιρέτη τάσης

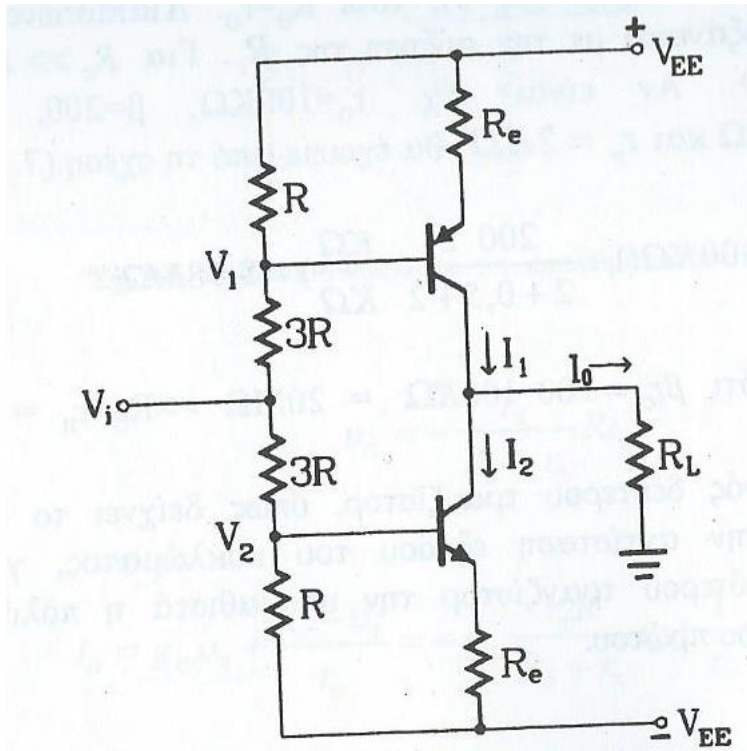
Αν θεωρήσουμε ότι ο διαιρέτης τάσης είναι ισχυρός και αγνοήσουμε το ρεύμα της βάσης του τρανζίστορ τότε το ρεύμα εξόδου I_o είναι

$$I_o = -\frac{V_{BE} + (1 - \lambda)V_{EE}}{R_e}$$

Όπου $\lambda = \frac{R_1}{R_1 + R_2}$

Το ρεύμα που παρέχεται από την πηγή επηρεάζεται από τις μεταβολές της τάσης τροφοδοσίας και από τις θερμικές μεταβολές του V_{BE} .

Το παρακάτω σχήμα δείχνει ένα κύκλωμα που μπορεί να δώσει θετικό ή αρνητικό ρεύμα. Είναι ουσιαστικά το ίδιο κύκλωμα με το παραπάνω σε συμμετρική συνδεσμολογία. Αυτό το κύκλωμα ονομάζεται αμφιπολική πηγή ρεύματος.

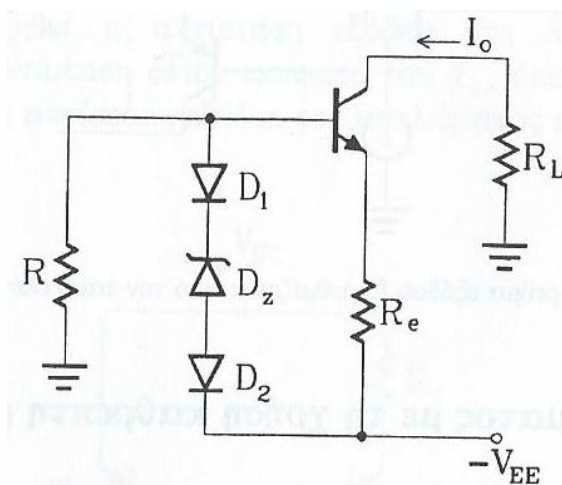


Σχήμα 1-2 Αμφιπολική πηγή ρεύματος

Προκύπτει για το ρεύμα εξόδου η παρακάτω σχέση

$$I_o = -\frac{V_i}{2R_e}$$

Στο παρακάτω κύκλωμα βλέπουμε μια πηγή ρεύματος με αναφορά την τάση zener. Αυτό το κύκλωμα αποτελεί μια παραλλαγή του κυκλώματος του σχήματος 1 όπου έχει τοποθετηθεί μια zener σε σειρά με δύο επιπλέον διόδους.



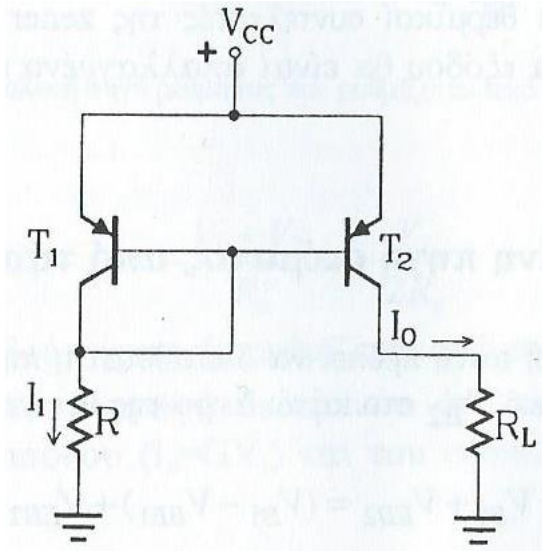
Σχήμα 1-3 Πηγή ρεύματος με αναφορά την τάση zener

Το ρεύμα στην έξοδο είναι τώρα

$$I_0 = \frac{V_z + V_D}{R_e}$$

Εδώ παρατηρούμε ότι το ρεύμα εξόδου δεν εξαρτάται από την τάση τροφοδοσίας.

Στο παρακάτω σχήμα παρουσιάζεται μια πηγή ρεύματος που χρησιμοποιεί καθρέπτη ρεύματος.



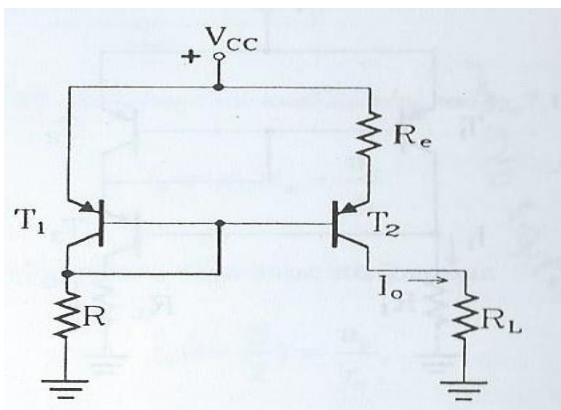
Σχήμα 1-4 Πηγή ρεύματος με χρήση καθρέπτη

Για τελείως όμοια τρανζίστορ το ρεύμα εξόδου θα είναι

$$I_0 = I_1 = \frac{V_{CC} - V_{BE}}{R}$$

Αυτή η πηγή ρεύματος έχει το μειονέκτημα ότι το ρεύμα εξόδου εξαρτάται από το V_{CC} και από τα V_{BE} και R που με τη σειρά τους εξαρτώνται από τη θερμοκρασία, ενώ η αντίσταση εξόδου του δεν έχει πολύ υψηλή τιμή.

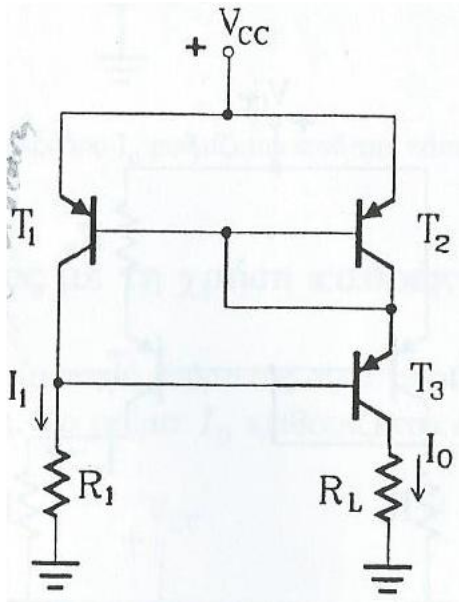
Παρακάτω παρουσιάζεται η πηγή ρεύματος Widlar



Σχήμα 1-5 Πηγή ρεύματος Widlar

Το κύκλωμα αυτό είναι όμοιο με το κύκλωμα του σχήματος 4. Η διαφορά είναι ότι προσθέσαμε την αντίσταση R_e και αυτό έγινε για να αυξήσουμε την αντίσταση εξόδου του προηγούμενου κυκλώματος. Το ρεύμα εξόδου δίνεται από τον ίδιο τύπο με το παραπάνω κύκλωμα.

Στο επόμενο κύκλωμα παρουσιάζουμε την πηγή ρεύματος Wilson.

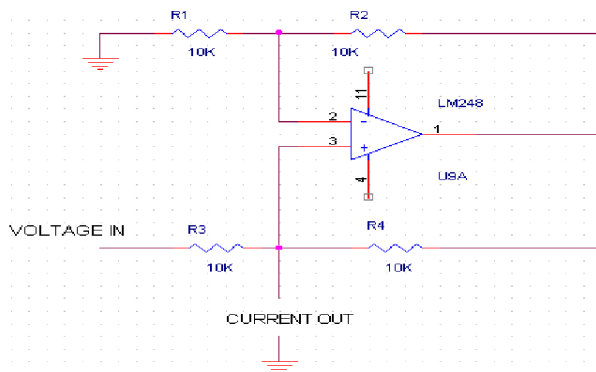


Σχήμα 1-6 Πηγή ρεύματος Wilson

Αν αντί για απλό καθρέπτη ρεύματος χρησιμοποιήσουμε τον καθρέπτη ρεύματος Wilson παίρνουμε το παραπάνω κύκλωμα που λέγεται πηγή ρεύματος Wilson. Το ρεύμα εξόδου εδώ δίνεται από τη σχέση

$$I_o = I_1 = \frac{V_{CC} - 2V_{EB}}{R_1}$$

Τέλος θα αναφέρουμε τη πηγή ρεύματος Howland που είναι και η πηγή που επιλέξαμε να χρησιμοποιήσουμε στο κύκλωμά μας. Αυτή φαίνεται στο παρακάτω σχήμα.



Σχήμα 1-7 Πηγή ρεύματος Howland

Όπως βλέπουμε και στο σχήμα 7 η πηγή ρεύματος Howland δεν υλοποιείται με τρανζίστορ αλλά στην καρδιά του κυκλώματος υπάρχει ένας τελεστικός ενισχυτής. Η πηγή ρεύματος Howland ουσιαστικά εκμεταλλεύεται την ιδιότητα του τελεστικού ενισχυτή να κρατά τις τάσεις στις εισόδους του ίσες μεταξύ τους, όταν χρησιμοποιείται σε διάταξη ίδια με αυτή του σχήματος 7. Όπως θα δούμε αναλυτικά σε αυτή την εργασία το κύκλωμα αυτό όπως επιλέξαμε να το υλοποιήσουμε κρατά το ρεύμα πολύ σταθερό σε μια περιοχή ρευμάτων 0-17mA και για φορτία από 1 Ohm έως και 1500 Ohm.

Το ρεύμα εξόδου σε αυτό το κύκλωμα, όπως θα αποδείξουμε σε παρακάτω κεφάλαιο δύνεται από τη σχέση

$$I_{out} = \frac{V_{in}}{R}$$

Όπου $R=R_3=R_4$.

Το βασικό κύκλωμα Howland Current Pump εφευρέθηκε από τον καθηγητή Bradford Howland του MIT το 1962 και η εφεύρεση αποκαλύφθηκε σε κάποιο συνάδελφό του. Το κύκλωμα δεν συνοδεύονταν από πατέντα. Η Howland Current Pump προδημοσιεύτηκε τον Ιανουάριο του 1964 στο περιοδικό "Lightning Empiricist", Volume 12, Αριθμός 1. Είναι το σχήμα 5A στη σελίδα 7 ενός άρθρου από τον D. H. Sheingold, με τίτλο "Impedance & Admittance Transformations using Operational Amplifiers". Το σχήμα αυτό φαίνεται παρακάτω, χωρίς λεζάντα, για λόγους πληρότητας. Το περιοδικό μπορεί να βρεθεί στη διεύθυνση www.philbrickarchive.org/1964-1_v12_no1_the_lightning_empiricist.htm.

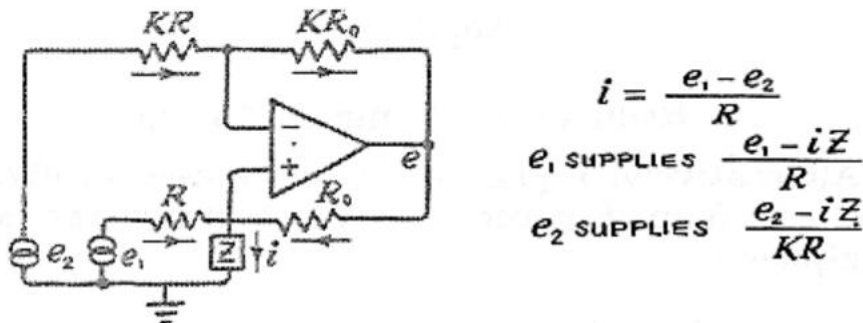
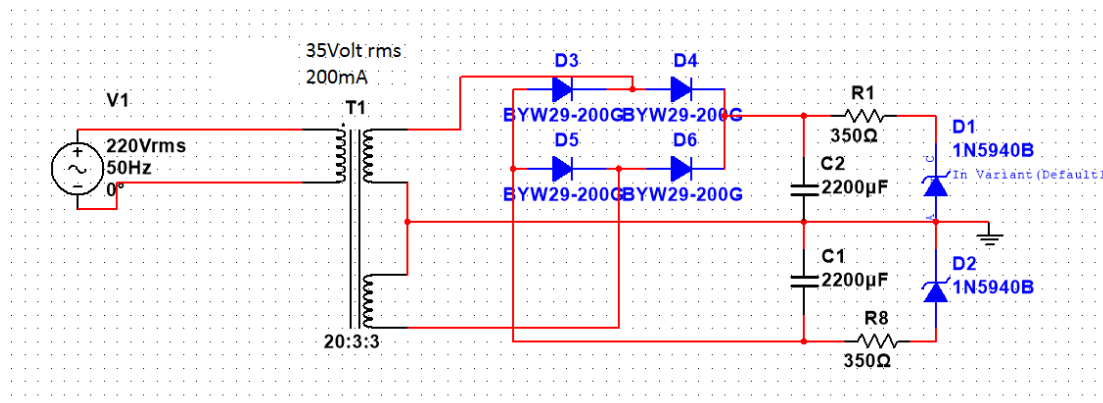


Fig. 5a Differential Voltage to Current Transducer
(Grounded sources, grounded load,
differential amplifier)

2. Τροφοδοτικό

Το τροφοδοτικό που χρησιμοποιούμε για την τροφοδοσία των επιμέρους κυκλωμάτων της εργασίας αυτής φαίνεται στο παρακάτω σχήμα.

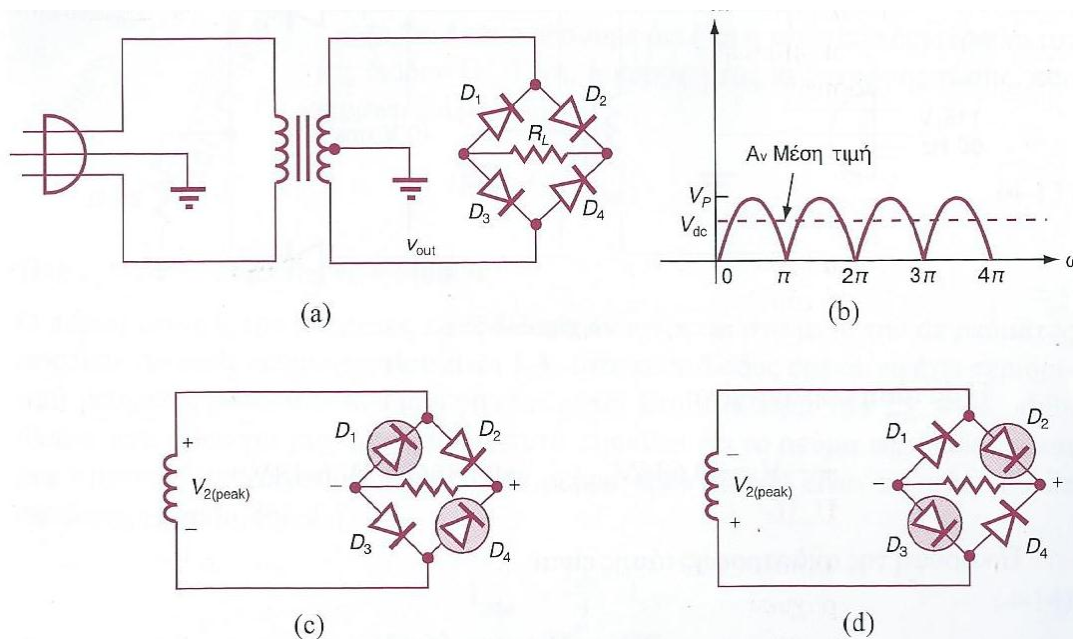


Σχήμα 2-1 Τροφοδοτικό εργασίας

Σε αυτό το κεφάλαιο θα αναλύσουμε τη λειτουργία του τροφοδοτικού. Όπως βλέπουμε και στο σχήμα 1 το τροφοδοτικό αποτελείται από ένα μετασχηματιστή με το πρωτεύον στα 220 volt και το δευτερεύον να μας δίνει 35Volt rms και 200 mA. Έπειτα χρησιμοποιείται μια γέφυρα διόδων, ένα φίλτρο πυκνωτή και μια zener στα 43Volt ώστε στην έξοδο να έχουμε περίπου 43 volt που είναι η επιθυμητή τάση τροφοδοσίας για το κύκλωμά μας.

2.1.Γέφυρα ανόρθωσης

Η Γέφυρα ανόρθωσης φαίνεται στο παρακάτω σχήμα



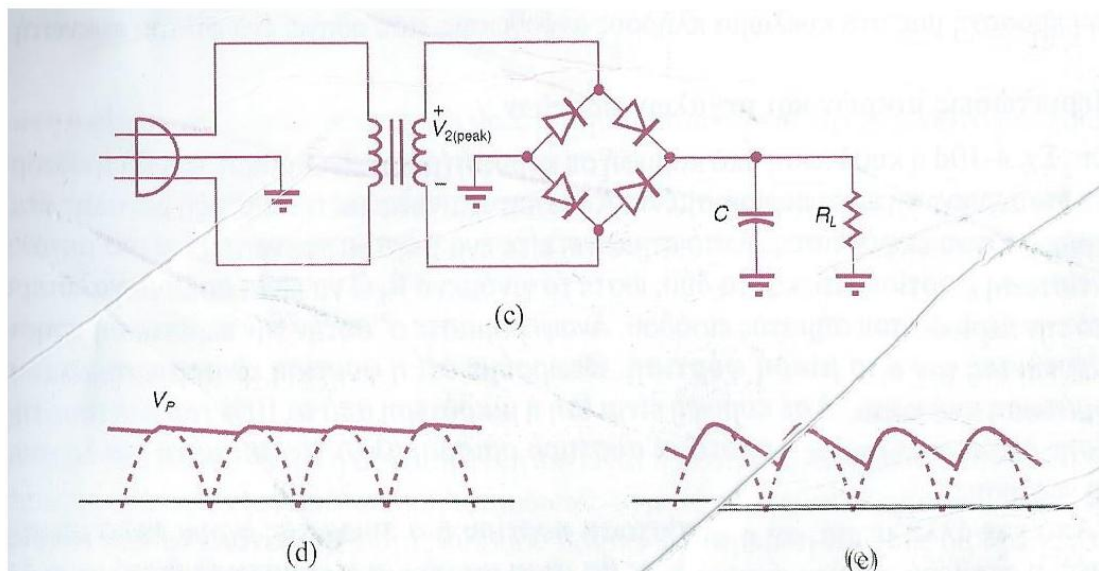
Σχήμα 2-2 Γέφυρα ανόρθωσης

Εδώ θα περιγράψουμε τη λειτουργία της γέφυρας ανόρθωσης. Όπως φαίνεται στη σχήμα 2 κατά τη θετική ημιπερίοδο της τάσης στο δευτερεύον οι δίοδοι D1 και D4 είναι πολωμένες ανάστροφα (σχήμα 2 C)ενώ οι δίοδοι D2 και D3 είναι πολωμένες ορθά επομένως το ρεύμα ακολουθεί τη διαδρομή D2 R_L D3. Κατά τη διάρκεια της αρνητικής ημιπεριόδου οι δίοδοι D2 και D3 είναι πολωμένες ανάστροφα (σχήμα 2 C)ενώ οι δίοδοι D1 και D4 είναι πολωμένες ορθά έτσι τώρα το ρεύμα ακολουθεί τη διαδρομή D4 R_L D1.

Κατά τη διάρκεια κάθε κύκλου το ρεύμα εισέρχεται στο δεξιό άκρο της αντίστασης φορτίου και εξέρχεται από το αριστερό, έτσι η πολικότητα της τάσης εξόδου παραμένει η ίδια. Έτσι η τάση εξόδου είναι ένα πλήρως ανορθωμένο σήμα όπως φαίνεται και στο σχήμα 2 (b).

2.2.Φίλτρο Πυκνωτή

Το κύκλωμα πλήρους ανόρθωσης που μελετήσαμε στην παραπάνω παράγραφο (σχήμα 2) έχει μία κυμάτωση στην τάση πάνω στο φορτίο, όπως φαίνεται στο σχήμα 2 (b). Για να κάνουμε την κυμάτωση αυτή ελάχιστη χρησιμοποιούμε το φίλτρο πυκνωτή. Όπως φαίνεται στο παρακάτω σχήμα.



Σχήμα 2-3 Φίλτρο Πυκνωτή

Όπως είπαμε παραπάνω η τάση στην έξοδο της γέφυρας πλήρους ανόρθωσης είναι πάντα θετική, όμως παρουσιάζει μια κυμάτωση. Μέχρι το μέγιστο της τάσης ο πυκνωτής φορτίζεται, ενώ όταν η τάση πέφτει από το μέγιστο ο πυκνωτής εκφορτίζεται πάνω στο φορτίο. Έτσι η τάση εξόδου έχει πλέον τη μορφή του σχήματος 3 (d). Αν η σταθερά χρόνου εκφόρτισης είναι μικρή τότε η τάση στην έξοδο έχει τη μορφή του σχήματος 3 (c). Αν όμως η σταθερά εκφόρτισης είναι μεγάλη τότε η τάση στην έξοδο είναι σχεδόν ευθεία γραμμή, όπως στο σχήμα 3 (d) που είναι και το επιθυμητό αποτέλεσμα.

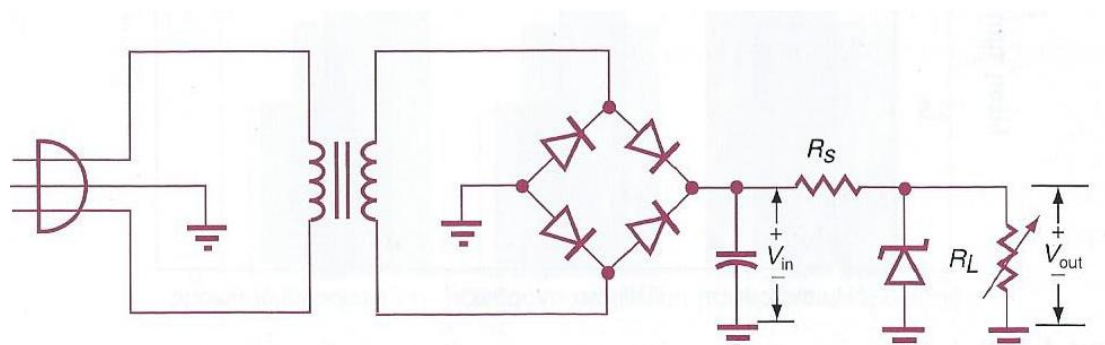
Για να υπολογίσουμε την τάση κυμάτωσης χρησιμοποιούμε τον τύπο

$$V_{rip} = \frac{I_{DC}}{fC} \quad \text{Όπου } V_{rip} = \text{τάση κυμάτωσης, } I_{DC} = \text{ρεύμα φορτίου } f = \text{συχνότητα κυμάτωσης και}$$

C=χωρητικότητα πυκνωτή. Όπως βλέπουμε όσο μεγαλύτερη είναι η χωρητικότητα πυκνωτή η τάση κυμάτωσης μειώνεται. Επομένως στο κύκλωμά μας είναι καλό να διαλέξουμε πυκνωτές με μεγάλη χωρητικότητα για να μειώσουμε την τάση κυμάτωσης.

2.3.Σταθεροποίηση με zener

Το τελικό βήμα για την υλοποίηση του τροφοδοτικού μας είναι να χρησιμοποιήσουμε από μια δίοδο zener στα δύο επιμέρους τμήματα του τροφοδοτικού για να σταθεροποιήσουμε ακόμη περισσότερο την τάση τροφοδοσίας. Εδώ θα δήξουμε τι κάνει η zener σε ένα κύκλωμα σταθεροποίησης όπως αυτό στο παρακάτω σχήμα.



Σχήμα 2-4 Σταθεροποίηση με zener

Στο σχήμα 4 φαίνεται ένα κύκλωμα σταθεροποίησης με φίλτρο πυκνωτή και zener πριν το φορτίο R_L . Η δίοδος zener έχει την ιδιότητα όταν είναι ανάστροφα πολωμένη να κρατά στα άκρα της την τάση σταθερή και ίση με V_{zener} . Επομένως στην έξοδο του κυκλώματος μπορούμε να θεωρήσουμε ότι η τάση είναι εντελώς σταθερή, (δεν έχει κυμάτωση) και ίση περίπου με V_{zener} . Η αντίσταση R_S είναι μια προστατευτική αντίσταση που τη βάζουμε για να περιορίσουμε το ρεύμα που περνά από την zener όταν το φορτίο R_L είναι μηδέν.

Αυτά τα επιμέρους στοιχεία που αναλύσαμε παραπάνω, δηλαδή ο μετασχηματιστής, η γέφυρα διόδων, το φίλτρο πυκνωτή και η δίοδος zener χρησιμοποιούνται για την υλοποίηση του κυκλώματος τροφοδοσίας που χρησιμοποιείται σε αυτή την εργασία. Η zener στο τελικό μας κύκλωμα που φαίνεται στο σχήμα 1 είναι στα 43Volt επομένως η τάση του τροφοδοτικού μας είναι 43Volt. Επίσης χρησιμοποιήσαμε μετασχηματιστή με μεσαία λήψη και έτσι έχουμε ένα τροφοδοτικό που δίνει θετική και αρνητική τάση στην έξοδό του +43Volt και -43Volt. Προσέξαμε επίσης η αντίσταση R_S που περιορίζει το ρεύμα που περνά από τη zener να είναι στο μισό watt γιατί μέσω αυτής περνάνε 100mA (από τον θετικό και άλλα τόσα από τον αρνητικό κλάδο του κυκλώματος). Ένας τρόπος να υπολογίσουμε το ρεύμα που περνά από την προστατευτική αντίσταση R_S είναι ο παρακάτω τύπος

$$I_S = \frac{V_{in} - V_{zener}}{R_S}$$

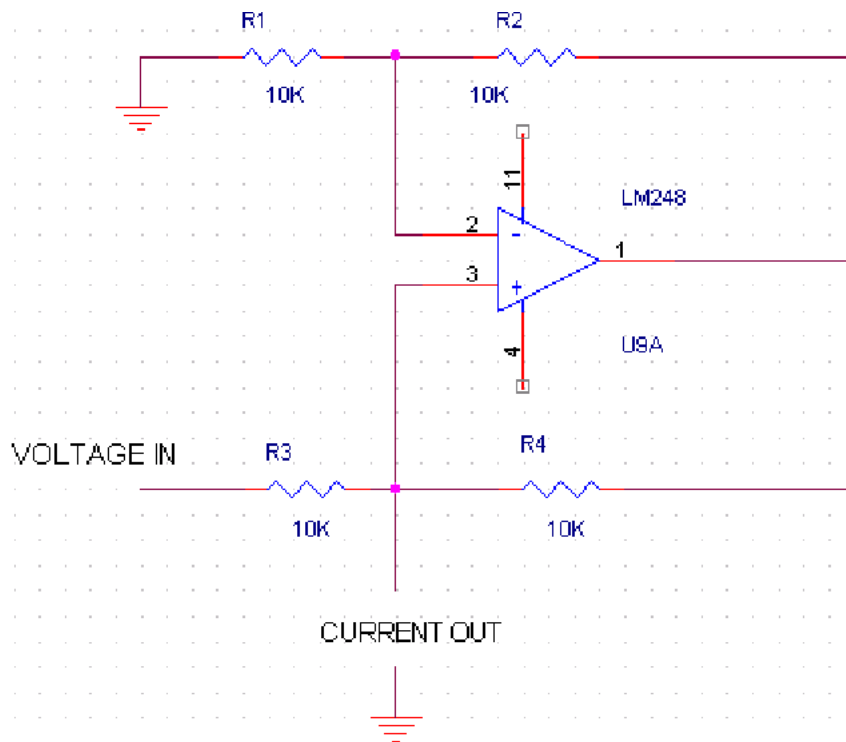
Επομένως σχεδιάζοντας το τροφοδοτικό επιλέγουμε την R_S έτσι ώστε να έχουμε το επιθυμητό ρεύμα, το οποίο θα έχει μέγιστο το ρεύμα που μπορεί να δώσει ο

μετασχηματιστής μας. (Εδώ ο μετασχηματιστής μας από την κατασκευή του, δίνει το πολύ 100mA στο θετικό και 100mA στο αρνητικό μέρος του κυκλώματός μας).

3. Πηγή ρεύματος Howland

3.1. Μαθηματική Επίλυση Howland

Στο παρακάτω σχήμα φαίνεται μια υλοποίηση της πηγής ρεύματος Howland. Οι πηγές ρεύματος συχνά αποκαλούνται στα αγγλικά και αντλίες ρεύματος (current pump). Χρησιμοποιούνται για να μετατρέψουν μια τιμή τάσης σε μια τιμή ρεύματος που θα διαρρέει ένα φορτίο ανεξαρτήτως του φορτίου. Στην καρδιά του κυκλώματός μας υπάρχει ένας τελεστικός ενισχυτής. Ο τρόπος λειτουργίας του κυκλώματος αυτού δεν είναι προφανής. Δεν έχει την κλασική διαμόρφωση ενός κυκλώματος τελεστικού ενισχυτή όπου μια είσοδος του τελεστικού έχει γνωστή σταθερή τιμή. Αυτό γίνεται γιατί έχουμε τόσο θετική όσο και αρνητική ανάδραση. Για να λύσουμε αυτό το κύκλωμα θα εφαρμόσουμε 2 βασικές αρχές, καταρχήν έχουμε αρνητική ανάδραση, επομένως οι εισόδους του τελεστικού θα είναι ίσες μεταξύ τους. Επιπλέον το ρεύμα που περνά στις εισόδους του τελεστικού μπορεί να θεωρηθεί μηδέν. Τώρα μπορούμε να χρησιμοποιήσουμε μαθηματικά για να λύσουμε το κύκλωμα.



Σχήμα 3-1 Αντλία ρεύματος Howland

Αν θεωρήσουμε ότι R1 και R2 έχουν ίδια τιμή, η είσοδος “-” του τελεστικού ενισχυτή θα έχει τάση ίση με το μισό της τάσης εξόδου του τελεστικού. Τώρα επειδή όπως είπαμε παραπάνω οι τάσεις στην είσοδο του τελεστικού είναι ίσες, λόγω αρνητικής ανάδρασης, η τάση στην “+” είσοδο του τελεστικού θα είναι επίσης ίση με το μισό της τάσης εξόδου του τελεστικού. Παρατηρήστε εδώ ότι οι R1 και R2 πρέπει να είναι ίσες αλλά δεν χρειάζεται να είναι 10k, μπορούν να έχουν οποιαδήποτε τιμή (όσο μεγαλύτερη τόσο το καλύτερο) αρκεί να είναι ίσες.

Πώς θα βρούμε λοιπόν την σχέση μεταξύ της τάσης εισόδου και του ρεύματος εξόδου; Άς αρχίσουμε με αυτά που ξέρουμε. Η τάση στην “-” είσοδο του τελεστικού είναι ίση με το μισό της τάσης εξόδου του. Άς ονομάσουμε την τάση “-” V_a και την τάση εξόδου του τελεστικού V_{out} .

$$V_a = \frac{1}{2} V_{out}$$

Άς δούμε τώρα τα ρεύματα: Το ρεύμα που διαρρέει την R3 θα δύνεται από τη σχέση

$$I_1 = \frac{V_{in} - V_a}{R3}$$

Το ρεύμα που περνά από την R4 είναι

$$I_2 = \frac{V_{out} - V_a}{R4}$$

Παρατηρούμε εδώ ότι υποθέτουμε ότι η φορά του ρεύματος I_2 είναι από την έξοδο του τελεστικού προς την R4. Τώρα επειδή από την είσοδο του τελεστικού δεν περνά ρεύμα, το ρεύμα στην έξοδο (Current Out) θα ισούται με το άθροισμα των ρευμάτων $I_1 + I_2$.

$$I_{out} = I_1 + I_2 = \frac{V_{in} - V_a}{R3} + \frac{V_{out} - V_a}{R4}$$

Η V_a όμως όπως δήξαμε παραπάνω είναι ίση με $V_{out}/2$ και η $R3 = R4 = R$ επομένως η παραπάνω εξίσωση γίνεται:

$$I_{out} = I_1 + I_2 = \frac{V_{in} - V_{out}/2}{R} + \frac{V_{out} - V_{out}/2}{R}$$

Λύνοντας την παραπάνω εξίσωση παίρνουμε

$$I_{out} = \frac{V_{in}}{R}$$

Αυτή είναι η τελική εξίσωση της πηγής ρεύματος Howland, όπως φαίνεται υπάρχει μια απλή σχέση που συνδέει την τάση στην είσοδο με το ρεύμα στην έξοδο της πηγής ρεύματός μας.

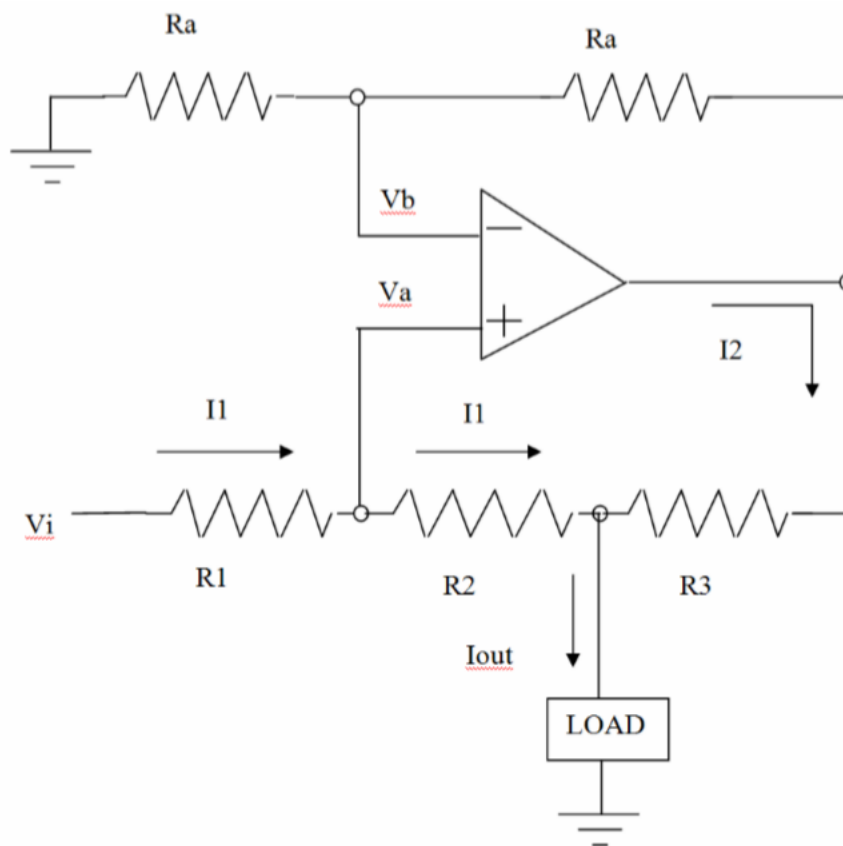
Όταν αλλάζει η αντίσταση φορτίου το ρεύμα μέσω του φορτίου πρέπει να παραμένει σταθερό. Το μόνο πράγμα που μπορεί να μεταβάλλεται ώστε να εξυπηρετεί τις αλλαγές στο φορτίο είναι η τάση εξόδου του τελεστικού ενισχυτή. Έτσι λοιπόν για να δουλεύει σωστά η πηγή ρεύματός μας, δηλαδή να δίνει σταθερό ρεύμα όσο και αν είναι η αντίσταση στο φορτίο, πρέπει η τάση εξόδου του τελεστικού ενισχυτή μας να αυξάνει ή να μειώνεται αντίστοιχα. Σε έναν ιδανικό τελεστικό αυτό συμβαίνει. Όμως σε ένα πραγματικό η τάση εξόδου περιορίζεται καταρχήν από τις τιμές της τάσης τροφοδοσίας του τελεστικού, αλλά και από άλλους παράγοντες. Αν δούμε στο φύλλο δεδομένων (datasheet) του τελεστικού που επιλέξαμε, αυτή η ιδιότητα αναφέρεται ως compliance voltage. Έτσι αν ψάχνουμε ένα τελεστικό κατάλληλο να χρησιμοποιηθεί σε μια πηγή ρεύματος Howland πρέπει αυτός να

έχει όσο γίνεται υψηλές τιμές τροφοδοσίας και υψηλή τιμή compliance voltage (τάσης συμμόρφωσης).

Το μόνο πρόβλημα του παραπάνω κυκλώματος είναι ότι το ρεύμα εξόδου δεν μπορεί να πάρει μεγάλες τιμές. Στο παράδειγμά μας χρησιμοποιήσαμε 10k αντιστάσεις. Εάν χρησιμοποιούσαμε 1k αντιστάσεις το ρεύμα στην έξοδο θα προσπαθούσε να πάρει την τιμή των 10mA με τάση εισόδου 10V. Αν είχαμε φορτίο 1k Ohm η πτώση τάσης πάνω στο φορτίο θα ήταν 10 Volts. Άρα έχουμε στην είσοδο 10V και στο $V_a = 10\text{Volts}$, άρα από την αντίσταση R_3 δεν θα περνά καθόλου ρεύμα και όλο το ρεύμα στο φορτίο θα πρέπει να παράγεται από τον τελεστικό ενισχυτή. Για να γίνει αυτό ο τελεστικός πρέπει να παράξει 20V στην έξοδό του. Αυτό είναι αρκετά μεγάλη τάση για ένα κοινό τελεστικό ενισχυτή. Επίσης πρέπει ο τελεστικός να παράξει όλα τα 10mA που είναι πάλι πολλά.

Επομένως παρατηρούμε ότι αυτό το κύκλωμα απαιτεί από τον τελεστικό να δώσει στην έξοδό του πολύ ρεύμα και πολύ τάση. Γι' αυτό το λόγο χρησιμοποιούμε ένα ελαφρώς τροποποιημένο κύκλωμα που μας λύνει (κατά ένα μεγάλο μέρος) αυτούς τους περιορισμούς. Αυτό το κύκλωμα λέγεται improved Howland current pump και θα το περιγράψουμε παρακάτω.

3.2. Η βελτιωμένη πηγή ρεύματος Howland (Improved Howland Current Pump)



Σχήμα 3-2 Improved Howland

Με την προσθήκη της αντίστασης R_3 (σχήμα 2), η τάση που πρέπει να δώσει ο τελεστικός στην έξοδό του μειώνεται αρκετά σε σχέση με τις απαιτήσεις του κλασικού

κυκλώματος Howland. Ποια είναι όμως τώρα η εξίσωση που συνδέει την τάση εισόδου σε σχέση με το ρεύμα εξόδου;

Για να λύσουμε το κύκλωμα θα χρησιμοποιήσουμε το εξής τέχνασμα. Μιας και η πηγή ρεύματος εξ ορισμού δίνει σταθερό ρεύμα ανεξαρτήτως φορτίου, θα βάλουμε μηδενική αντίσταση στο φορτίο, δηλαδή θα γειώσουμε την έξοδο.

Με την έξοδο γειωμένη, το I_1 στο σχήμα 2 είναι

$$I_1 = \frac{V_{in}}{R_1 + R_2}$$

Η τάση στο V_a είναι

$$V_a = I_1 * R_2$$

Η τάση στο V_b λόγω αρνητικής ανάδρασης είναι ίση με την V_a . Το ρεύμα I_2 που περνά από την R_3 είναι

$$I_2 = \frac{V_{out}}{R_3}$$

Το ρεύμα στην έξοδο του κυκλώματος I_{out} είναι $I_{out} = I_1 + I_2$.

$$I_{out} = \frac{V_{in}}{R_1 + R_2} + \frac{V_{out}}{R_3}$$

Ένας άλλος τρόπος να εκφράσουμε το V_{out} είναι $V_{out} = 2V_a = 2 * I_1 * R_2$. Και αντικαθιστώντας το I_1 στην παραπάνω εξίσωση έχουμε

$$V_{out} = \frac{2 * V_{in} * R_2}{R_1 + R_2}$$

Έτσι στην εξίσωση για το ρεύμα εξόδου I_{out} έχουμε

$$I_{out} = \frac{V_{in}}{R_1 + R_2} + \frac{\frac{2 * V_{in} * R_2}{R_1 + R_2}}{R_3}$$

Βγάζουμε κοινό παράγοντα στην παραπάνω εξίσωση το $V_{in}/(R_1 + R_2)$ και έχουμε

$$I_{out} = \frac{V_{in}}{R_1 + R_2} \left(1 + \frac{2 * R_2}{R_3}\right)$$

Και αυτή είναι η τελική εξίσωση που συνδέει το ρεύμα εξόδου με την τάση στην είσοδο στο κύκλωμα improved Howland υπό την προϋπόθεση ότι οι δύο αντιστάσεις R_a είναι ίσες μεταξύ τους. Αυτή η εξίσωση μπορεί να απλοποιηθεί περισσότερο, όπως δείχνουμε παρακάτω.

Επειδή R_1 περίπου ίσο με R_2 θέτοντας $R_1=R_2$ στην παραπάνω εξίσωση, γίνεται

$$I_{out} = \frac{V_{in}}{2 * R_1} + \frac{V_{in}}{R_3}$$

Και επιπλέον επειδή η R_1 είναι πολύ μεγαλύτερη από την R_3 ο πρώτος όρος της παραπάνω εξίσωσης μπορεί να θεωρηθεί μηδενικός, άρα τελικά έχω

$$I_{out} = \frac{V_{in}}{R_3}$$

Εδώ χρειάζεται προσοχή, διότι αυτή η εξίσωση ισχύει υπό τις προϋποθέσεις που αναπτύξαμε παραπάνω.

Ας βάλουμε μερικές αριθμητικές τιμές στις παραπάνω εξισώσεις για να δούμε τι απαιτήσεις έχουμε τώρα από τον τελεστικό ενισχυτή μας. Πάλι για να λυθεί εύκολα το κύκλωμα πρέπει να γιωθεί η έξοδος.

Στην περίπτωση που εξετάζουμε $R_1 = R_2 + R_3$ και επιλέγουμε τιμές για τις αντιστάσεις μας βασιζόμενοι σε αυτή τη σχέση. Αν $R_1=10k\Omega$ τότε $R_2=9.9k\Omega$ και $R_3=100\Omega$. Για ευκολία θέτουμε $V_{in}=10V$.

Ποιο είναι λοιπόν το ρεύμα που διαρρέει τον κλάδο των R_1 και R_2 ; Θα είναι η τάση 10V διαιρούμενη από το άθροισμα των δύο αντιστάσεων, δηλαδή 50mA. Η πτώση τάσης κατά πάνω στην R_2 θα είναι περίπου 4,975V. Τόση θα είναι και η τάση στην είσοδο του τελεστικού (V_a), επομένως η τάση στην έξοδο του τελεστικού θα είναι $2*V_a=9.95V$. Άρα όντως οι απαιτήσεις σε τάση εξόδου στον τελεστικό ενισχυτή έχουν πλέον μειωθεί πολύ (στο μισό) από το απλό κύκλωμα Howland.

Αυτά τα 9.95V πέφτουν πάνω στην R_3 επομένως το ρεύμα μέσω της R_3 είναι 0,0995A. Αυτό είναι πολύ μεγάλη τιμή για ρεύμα εξόδου τελεστικού ενισχυτή. Το αντίστοιχο χαρακτηριστικό που κοιτάμε στα φύλλα δεδομένων είναι το short circuit current, το οποίο έχει τιμές γύρω στα 25 mA. Επομένως στην πράξη δεν περιμένουμε να πάρουμε ρεύματα της τάξης των 100mA αλλά έχουμε ένα ιδανικό μέγιστο τα 25mA. Το παραπάνω κύκλωμα το υλοποιήσαμε στην πράξη με αυτές τις τιμές αντιστάσεων που προτείναμε παραπάνω με τον τελεστικό ενισχυτή LM741, υπάρχουν όμως κάποια πρακτικά προβλήματα.

Όπως θα αναφέρουμε σε επόμενη παράγραφο, ο LM741 δεν είναι κατάλληλος για την εφαρμογή που θέλουμε, και έτσι προτιμήσαμε τον OPA445AP. Αυτός είναι ένας τελεστικός που έχει μεγάλη τάση τροφοδοσίας, μεγάλη τάση συμμόρφωσης (compliance voltage) και αρκετά μεγάλο ρεύμα εξόδου (short circuit current) σε σχέση με τον LM741 που είναι ένας κοινός τελεστικός ενισχυτής. Αυτός ο τελεστικός όπως θα δούμε παρακάτω μπορεί να μας δώσει περισσότερο ρεύμα εξόδου και να κρατήσει την τάση στο φορτίο αρκετά μεγάλη επομένως είναι προτιμότερος για την πηγή ρεύματός μας. Θα δούμε τα προβλήματα που παρουσιάζονται στην πράξη πιο αναλυτικά σε επόμενες παραγράφους.

Πριν κλείσουμε το κεφάλαιο για το κύκλωμα Improved Howland Current Pump πρέπει να αναφερθεί το εξής. Το κύκλωμα που δείξαμε στο σχήμα 13 είναι θεωρητικό κύκλωμα που μας βοηθά να εξάγουμε εύκολα τη σχέση για το ρεύμα εξόδου της πηγής ρεύματός μας.

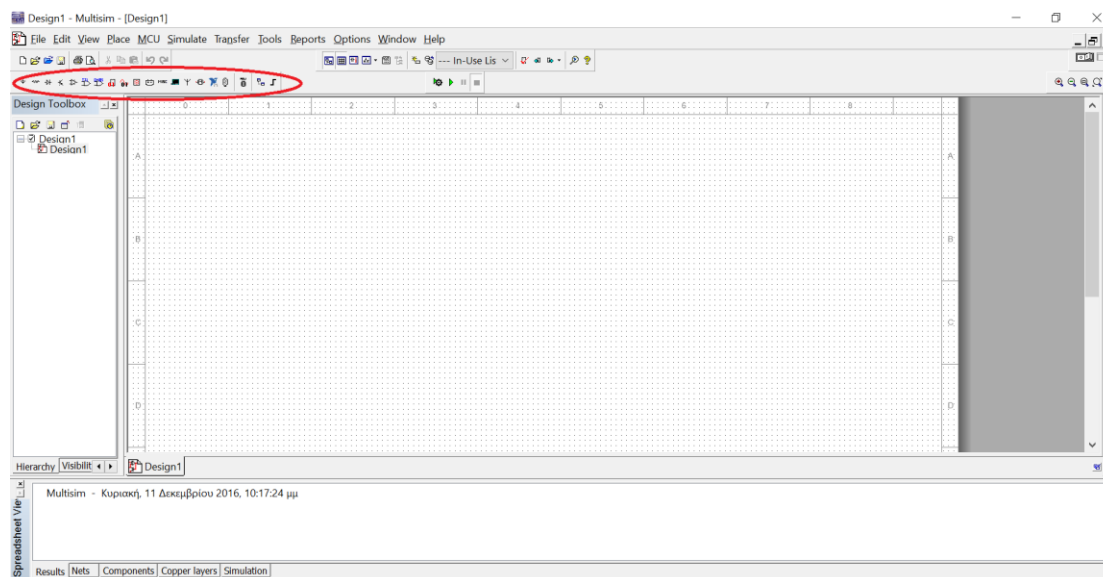
Υλοποιώντας αυτό το κύκλωμα στην πράξη παρατηρήσαμε ότι ενώ δουλεύει για έναν κοινό τελεστικό ενισχυτή όπως ο LM741, το κύκλωμα δεν δουλεύει για τον OPA445AP. Όπως θα δούμε ο LM741 έχει τάση τροφοδοσίας 15Volt (άρα δεν μπορεί να δώσει στην έξοδό του περισσότερη τάση από 15Volt) και έτσι δεν μας βολεύει για την πηγή ρεύματός μας. Προτιμήσαμε λοιπόν τον OPA445AP ο οποίος έχει τάση τροφοδοσίας 45Volt και άρα μπορεί να δώσει αρκετά μεγαλύτερη τάση στην έξοδό του και έτσι μπορούμε να πάρουμε μεγαλύτερες τιμές ρεύματος στην πηγή ρεύματός μας.

Υλοποιώντας όμως το κύκλωμα του σχήματος 13 με τον OPA445AP διαπιστώσαμε ότι δεν λειτουργεί. Έτσι αναφερθήκαμε στα φίλα δεδομένων του τελεστικού και βρήκαμε εκεί μια πηγή ρεύματος improved howland current pump ελαφρώς τροποποιημένη. Αυτή τη σχεδίαση χρησιμοποιήσαμε στο κύκλωμά μας. Θα παρουσιαστεί λεπτομερώς σε επόμενη παράγραφο.

3.3.Μια εισαγωγή στο Multisim και σχεδιασμός της πηγής ρεύματος improved Howland current pump

Πριν περάσουμε στην υλοποίηση του κυκλώματος Improved Howland με πραγματικά στοιχεία (αντιστάσεις και τελεστικούς ενισχυτές) πάνω στο breadboard, θα το προσομοιώσουμε στο Multisim για να μελετήσουμε τη συμπεριφορά του. Επομένως αυτό το τμήμα της εργασίας είναι αφιερωμένο στο Multisim και το σχεδιασμό του κυκλώματος improved Howland current pump. Αντί για μια γενική εισαγωγή στο πρόγραμμα, εδώ θα δήξουμε ότι χρειάζεται για την υλοποίηση του κυκλώματος που μας ενδιαφέρει.

Το Multisim είναι πρόγραμμα σχεδιασμού και προσομοίωσης κυκλωμάτων της National Instruments. Το γραφικό περιβάλλον του προγράμματος φαίνεται στο παρακάτω σχήμα.



Σχήμα 3-3 Το περιβάλλον του Multisim 13.0

Στη κόκκινη έλλειψη είναι τονισμένα τα στοιχεία που μας ενδιαφέρουν και με τα οποία θα δουλέψουμε στο πρόγραμμα. Είναι ουσιαστικά μια παλέτα από ηλεκτρονικά εξαρτήματα που μπορούμε να βάλουμε στο κύκλωμά μας και να προσομοιώσουμε.

Για την υλοποίηση του κυκλώματος στο Multisim 13 από την εργαλειοθήκη αυτή πρέπει να επιλεγούν τα παρακάτω στοιχεία και τα μπουν στο κύκλωμα.

Για το τροφοδοτικό μας:

1 μια πηγή 220Volt 50Hz

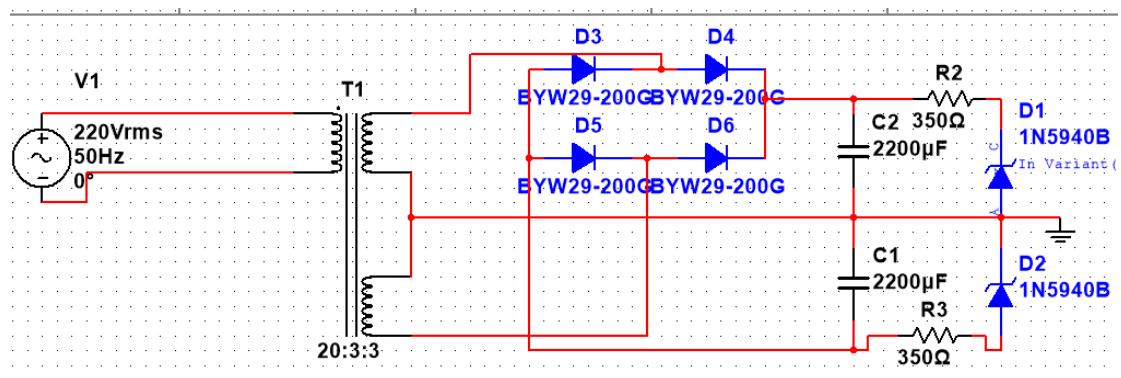
2 ένα μετασχηματιστή με μεσαία λήψη, τον φέρνουμε στο περιβάλλον σχεδίασης και με δεξί κλικ στα properties τον τροποποιούμε ώστε να δίνει 35Volt rms και 200mA ρεύμα.

3 Τέσσερις διόδους οι οποίες θα συνδεθούν μεταξύ τους σε διάταξη γέφυρας διόδων.

4 Δύο πυκνωτές αρκετά μεγάλης χωρητικότητας για την υλοποίηση του φίλτρου πυκνωτή. Εδώ επιλέξαμε χωρητικότητα 2200mF.

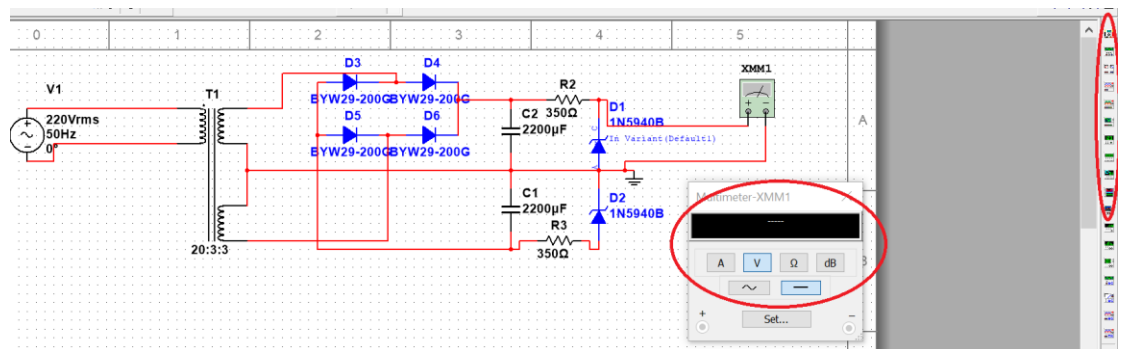
5 Δύο διόδους zener στα 43volt.

Τα παραπάνω στοιχεία τοποθετούνται στη περιοχή σχεδίασης του multisim και συνδέονται ώστε να πάρουμε το παρακάτω ηλεκτρονικό κύκλωμα.



Σχήμα 3-4 Τροφοδοτικό εργασίας σχεδιασμένο στο Multisim13

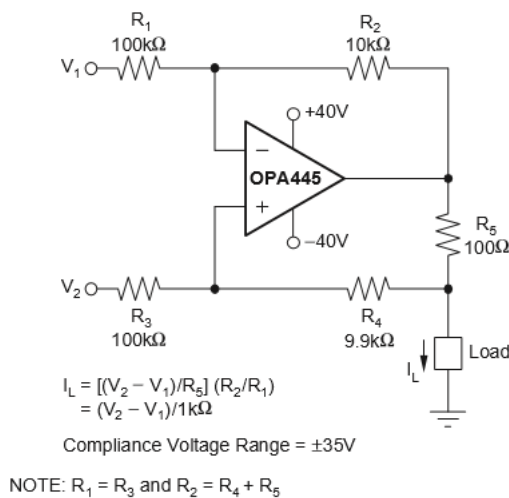
Για την προσομοίωση του κυκλώματος που σχεδιάσαμε παίρνουμε ένα πολύμετρο από την παλέτα οργάνων που προσφέρονται στο Multisim και βρίσκεται στην δεξιά πλευρά του περιβάλλοντος σχεδίασης. Το πολύμετρο είναι το πρώτο στοιχείο της παλέτας οργάνων του προγράμματος. Αφού τοποθετήσουμε το πολύμετρο στη περιοχή σχεδίασης κάνουμε διπλό κλικ πάνω του και επιλέγουμε να μετρήσουμε Volt ή Ampere και για συνεχές ρεύμα. Στο παρακάτω σχήμα φαίνονται οι επιλογές που περιγράψαμε.



Σχήμα 3-5 Σχεδίαση τροφοδοτικού στο Multisim 13

Στο σχήμα 16 με κόκκινο χρώμα φαίνονται οι επιλογές που αναφέρθηκαν παραπάνω. Όπως τοποθετήσαμε το πολύμετρο μπορούμε να μετρήσουμε την τάση πάνω στη δίοδο zener. Κάνουμε κλικ στο κουμπί run (F5) που βρίσκεται πάνω από την περιοχή σχεδίασης και η προσομοίωση ξεκινά, δίνοντάς μας τη τιμή της τάσης που επιλέξαμε να μετρήσουμε με το πολύμετρο.

Έχοντας σχεδιάσει το τροφοδοτικό μας στο Multisim μπορούμε τώρα να σχεδιάσουμε το κύκλωμα της πηγής ρεύματος. Πριν προχωρήσουμε στη σχεδίαση πρέπει να αναφέρουμε ότι το κύκλωμα improved howland που θα σχεδιάσουμε διαφέρει ελάχιστα από αυτό που μελετήσαμε στην ενότητα 3.2. Η διαφορά εντοπίζεται στις αντιστάσεις και τη συμμετρία του κυκλώματος. Το κύκλωμα που θα υλοποιήσουμε φαίνεται στο παρακάτω σχήμα.



Σχήμα 3-6 Το κύκλωμα improved Howland που θα χρησιμοποιήσουμε

Παρατηρώντας τις διαφορές του παραπάνω κυκλώματος με αυτό που μελετήθηκε στην παράγραφο 3.2 πρέπει να αναφερθούν τα εξής. Καταρχήν το κύκλωμα είναι ακριβώς το ίδιο, απλώς αλλάζουν οι τιμές των αντιστάσεων. Επιπλέον οι αντιστάσεις στη δεξιά πλευρά του κυκλώματος είναι ίσες μεταξύ τους, (βλέπε R_1 και R_3). Επίσης στη δεξιά πλευρά του κυκλώματος έχουμε την $R_2 = R_4 + R_5$.

Η εξίσωση που μας δίνει το ρεύμα εξόδου σε αυτό το κύκλωμα δίνεται παρακάτω.

$$I_L = \frac{V_2 - V_1}{R_5} * \frac{R_2}{R_1}$$

Όπως φαίνεται στο σχήμα 3-6 η είσοδος για το κύκλωμα improved Howland είναι οι τάσεις V_1 και V_2 . Εμείς επιλέγουμε να γειώσουμε τη V_1 , επομένως η εξίσωση για το ρεύμα εξόδου γίνεται.

$$I_L = \frac{V_2}{R_5} * \frac{R_2}{R_1}$$

Για την προσομοίωση του κυκλώματος του σχήματος 3-6 θα εργαστούμε στο ίδιο σχέδιο με το κύκλωμα τροφοδοσίας που προαναφέραμε. Τα εξαρτήματα που θα χρειαστούν αναφέρονται παρακάτω.

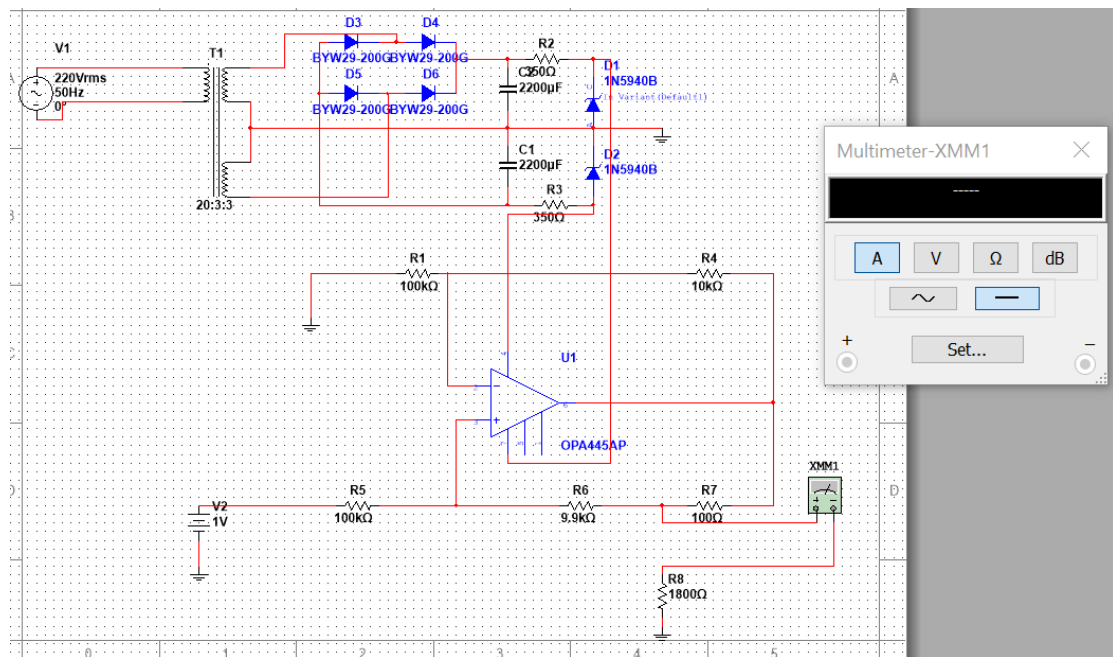
1 Πέντε αντιστάσεις των οποίων τις τιμές τροποποιούμε ώστε να είναι ίδιες με αυτές του σχήματος 3-6.

2 Τον τελεστικό OPA445AP, ο οποίος είναι τελεστικός που υπάρχει στην παλέτα τελεστικών ενισχυτών του Multisim13.

3 Μια πηγή DC τάσης.

4 Μια αντίσταση που θα παίζει το ρόλο του φορτίου στο κύκλωμα improved Howland.

Στο παρακάτω σχήμα φαίνεται το τελικό κύκλωμα, δηλαδή το τροφοδοτικό και η improved Howland Current Pump.



Σχήμα 3-7 Κύκλωμα τροφοδοσίας και improved Howland. Παρατηρήστε το φορτίο 1800Ωm στην έξοδο του Howland και το πολύμετρο που μετρά συνεχές ρεύμα.

Παρατηρώντας το σχήμα 3-7 βλέπουμε ότι έχουμε μειώσει την τάση V_1 του κυκλώματος, ενώ στην θέση της V_2 βάλαμε μια πηγή συνεχούς τάσης. Σε αυτή τη θέση στο τελικό κύκλωμα θα έχουμε μια τάση ψηφιακά ελεγχόμενη από PC μέσω σειριακής θύρας και ενός DAC.

Πριν προσομοιώσουμε το παραπάνω κύκλωμα θα εργαστούμε λίγο με τον τύπο που μας δίνει το ρεύμα εξόδου για να δούμε τι τιμές ρεύματος να περιμένουμε από την προσομοίωση και τι τιμές τάσης πρέπει να δοθούν στην είσοδο του Howland(τάση V_2).

Αντικαθιστώντας τις τιμές στον τύπο για το ρεύμα έχουμε

$$I_L = \frac{V_1}{100} * \frac{10k}{100k}$$

Έτσι για τάση $V_1=10\text{Volt}$ το $I_L=10\text{mAmpere}$. Επομένως αλλάζοντας την τιμή V_1 (V_2 στο σχήμα 3-7) από το μηδέν έως το 20Volt περιμένουμε να πάρουμε ρεύμα στην έξοδο του improved Howland στην περιοχή από 0 έως 20mAmps.

Στην παρακάτω ενότητα θα προσομοιώσουμε το κύκλωμα που σχεδιάσαμε εδώ (σχήμα 3-7) για διάφορες τιμές της V_2 και διάφορες τιμές της αντίστασης φορτίου.

3.4. Προσομοίωση στο Multisim 13 της Improved Howland Current Pump

Σε αυτή την ενότητα θα παρουσιάσουμε τις προσομοιώσεις στο Multisim 13 της πηγής ρεύματος Improved Howland. Εδώ θα παρουσιαστούν γραφικές για ρεύμα στο φορτίο 1mA 10mA 15mA 20mA σε συνάρτηση με την αντίσταση φορτίου. Για τον τελεστικό ενισχυτή OPA445AP και για τον τελεστικό ενισχυτή LM741. Επίσης θα παρουσιαστεί μια γραφική με ρεύμα εξόδου 30mA και μια γραφική με ρεύμα εξόδου 20 mA και τάση τροφοδοσίας του τελεστικού ενισχυτή τα 100Volt (για τον OPA445AP). Και θα βγουν μερικά γενικά συμπεράσματα για την πηγή ρεύματος και την υλοποίησή της στην πράξη.

3.4.1. Προσομοίωση Improved Howland με τον OPA445AP

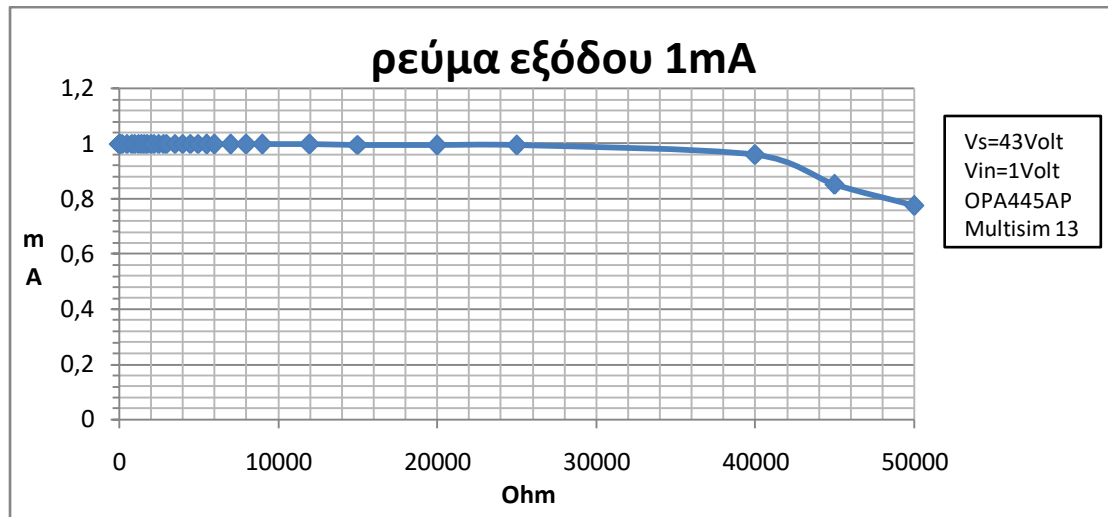
Το κύκλωμα που προσομοιώσαμε στο Multisim για αυτή την ενότητα φαίνεται στο σχήμα 3-7. Για τιμή ρεύματος εξόδου το 1mA ο πίνακας τιμών της γραφικής ρεύματος εξόδου σε συνάρτηση με το φορτίο φαίνεται παρακάτω.

Πίνακας 3-1 Ρεύμα εξόδου σε συνάρτηση με το φορτίο για ρεύμα εξόδου 1mA.

A/A	Έξοδος σε mA	φορτίο σε Ohm	A/A	Έξοδος σε mA	φορτίο σε Ohm
1	1	1	27	0,998	12000
2	1	10	28	0,996	15000
3	1	50	29	0,994	20000
4	1	100	30	0,994	25000
5	1	200	31	0,959	40000
6	1	500	32	0,852	45000
7	1	800	33	0,774	50000
8	1	1000			
9	1	1200			
10	1	1400			
11	1	1600			
12	1	1800			
13	1	2000			
14	1	2200			
15	1	2500			
16	1	2800			
17	1	3000			
18	1	3500			
19	1	4000			
20	1	4500			
21	1	5000			
22	0,998	5500			

23	0,998	6000		
24	0,998	7000		
25	0,998	8000		
26	0,998	9000		

Η γραφική ρεύματος εξόδου σε συνάρτηση με την αντίσταση φορτίου φαίνεται παρακάτω



Σχήμα 3-8 Γραφική ρεύματος εξόδου σε συνάρτηση με την αντίσταση φορτίου για ρεύμα 1mA

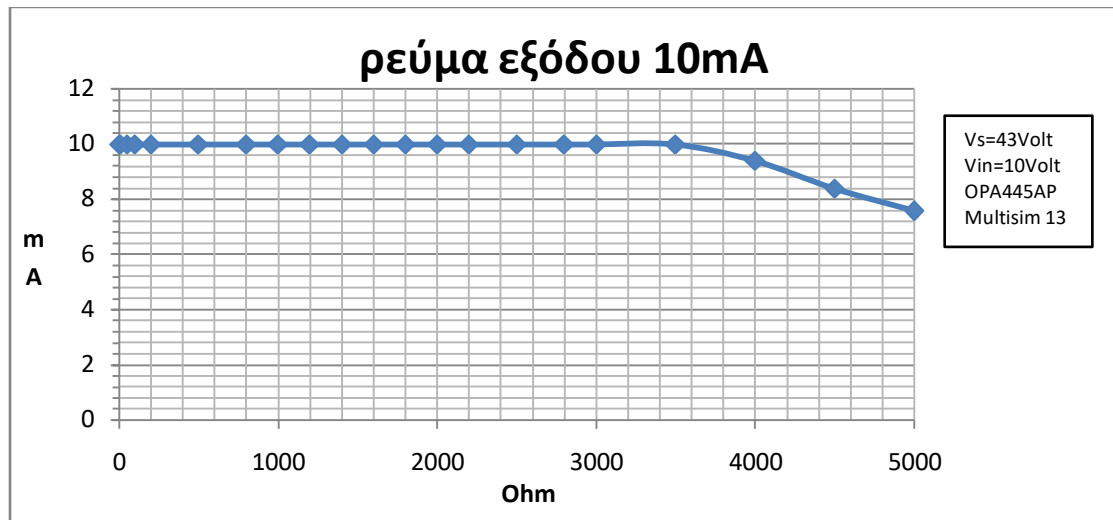
Παρατηρώντας τον πίνακα τιμών διαπιστώνεται ότι μετά τα 40kOhm η τιμή του ρεύματος εξόδου αλλάζει στο δεύτερο δεκαδικό ψηφίο, ενώ στα 45kOhm αλλάζει αρκετά ώστε να μην μπορεί να χρησιμοποιηθεί το κύκλωμα πλέον ως πηγή ρεύματος.

Για τιμή ρεύματος εξόδου το 10mA ο πίνακας τιμών της γραφικής ρεύματος εξόδου σε συνάρτηση με το φορτίο φαίνεται παρακάτω.

Πίνακας 3-2 Ρεύμα εξόδου σε συνάρτηση με το φορτίο για ρεύμα εξόδου 10mA.

A/A	Έξοδος σε mA	φορτίο σε Ohm	A/A	Έξοδος σε mA	φορτίο σε Ohm
1	10	1	13	9,994	2000
2	10	10	14	9,994	2200
3	10	50	15	9,994	2500
4	10	100	16	9,994	2800
5	9,999	200	17	9,994	3000
6	9,998	500	18	9,99	3500
7	9,998	800	19	9,4	4000
8	9,997	1000	20	8,384	4500
9	9,997	1200	21	7,567	5000
10	9,997	1400			
11	9,997	1600			
12	9,997	1800			

Η γραφική ρεύματος εξόδου σε συνάρτηση με την αντίσταση φορτίου φαίνεται παρακάτω.



Σχήμα 3-9 Γραφική ρεύματος εξόδου σε συνάρτηση με την αντίσταση φορτίου για ρεύμα 10mA

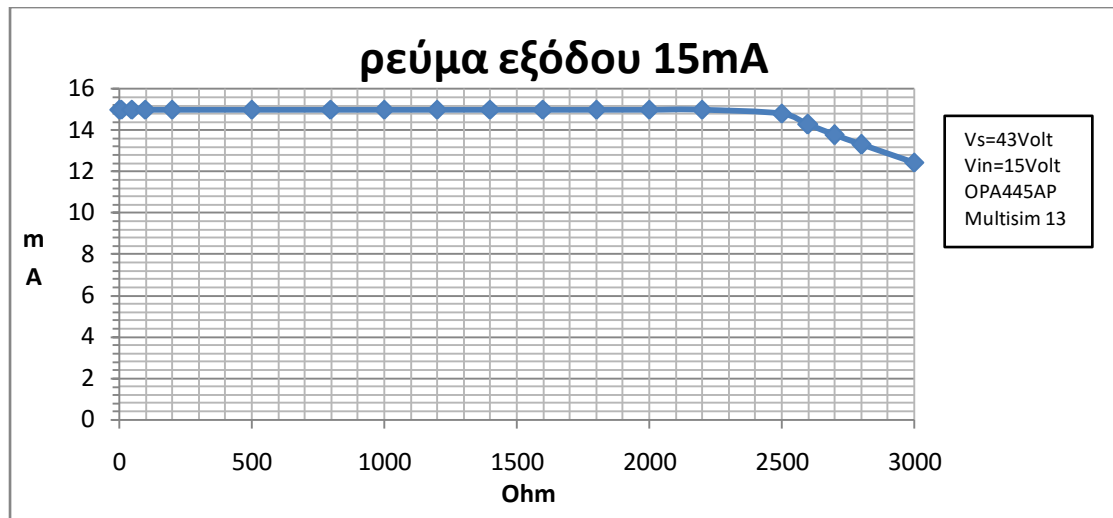
Εδώ παρατηρείται ότι μετά τα 3500 Ohm το κύκλωμα δεν μπορεί να δουλέψει ως πηγή ρεύματος μιας και η τιμή του ρεύματος αλλάζει σημαντικά με την αντίσταση φορτίου.

Για τιμή ρεύματος εξόδου το 15mA ο πίνακας τιμών της γραφικής ρεύματος εξόδου σε συνάρτηση με το φορτίο φαίνεται παρακάτω.

Πίνακας 3-3 Ρεύμα εξόδου σε συνάρτηση με το φορτίο για ρεύμα εξόδου 15mA.

A/A	Έξοδος σε mA	φορτίο σε Ohm	A/A	Έξοδος σε mA	φορτίο σε Ohm
1	15	1	12	14,99	1800
2	15	10	13	14,99	2000
3	15	50	14	14,99	2200
4	15	100	15	14,81	2500
5	14,99	200	16	14,27	2600
6	14,99	500	17	13,77	2700
7	14,99	800	18	13,3	2800
8	14,99	1000	19	12,44	3000
9	14,99	1200			
10	14,99	1400			
11	14,99	1600			

Και η αντίστοιχη γραφική παράσταση φαίνεται στο παρακάτω σχήμα



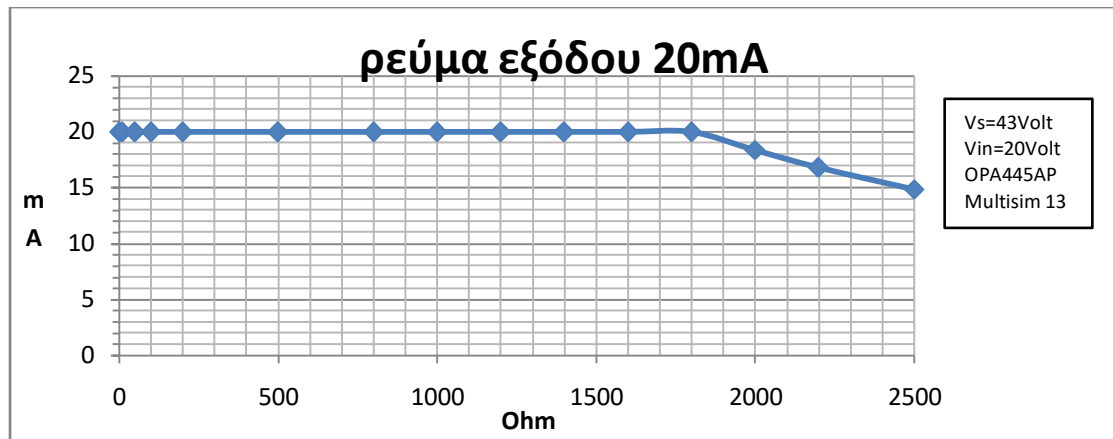
Σχήμα 3-10 Γραφική ρεύματος εξόδου σε συνάρτηση με την αντίσταση φορτίου για ρεύμα 15mA

Το κύκλωμα λειτουργεί ως πηγή ρεύματος μέχρι τα 2500Ohm φορτίου. Για τιμή ρεύματος εξόδου το 20mA ο πίνακας τιμών της γραφικής ρεύματος εξόδου σε συνάρτηση με το φορτίο φαίνεται παρακάτω.

Πίνακας 3-4 Ρεύμα εξόδου σε συνάρτηση με την αντίσταση φορτίου για ρεύμα εξόδου 20mA.

A/A	Έξοδος σε mA	φορτίο σε Ohm	A/A	Έξοδος σε mA	φορτίο σε Ohm
1	19,99	1	9	19,99	1200
2	19,99	10	10	19,99	1400
3	19,99	50	11	19,99	1600
4	19,99	100	12	19,99	1800
5	19,99	200	13	18,37	2000
6	19,99	500	14	16,79	2200
7	19,99	800	15	14,85	2500
8	19,99	1000			

Η αντίστοιχη γραφική παράσταση φαίνεται στο παρακάτω σχήμα.



Σχήμα 3-11 Γραφική ρεύματος εξόδου σε συνάρτηση με την αντίσταση φορτίου για ρεύμα 20mA

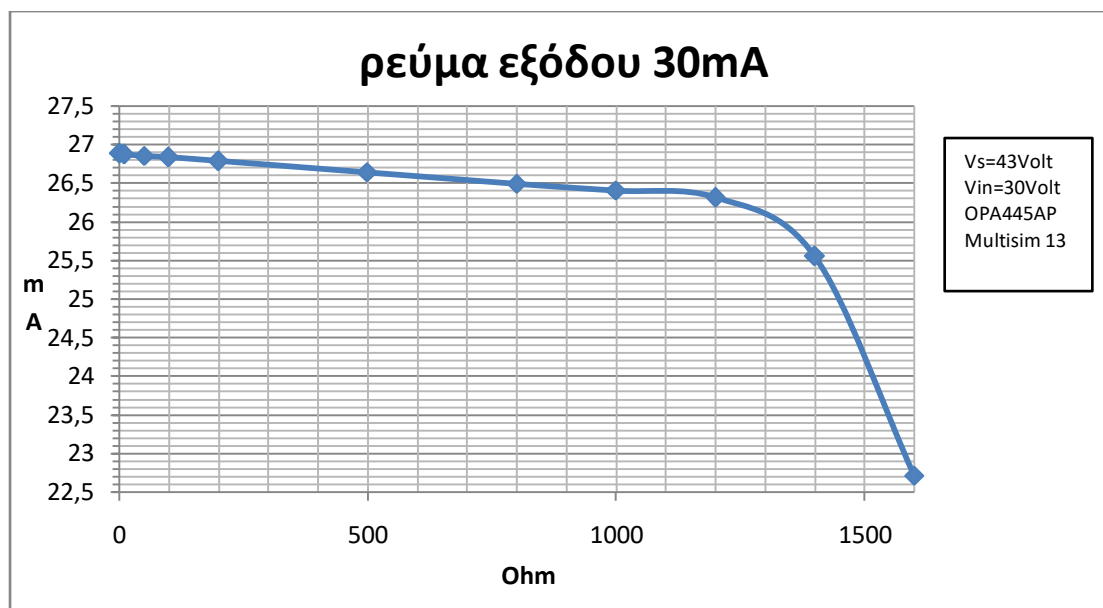
Εδώ παρατηρούμε ότι η πηγή ρεύματος λειτουργεί σωστά μέχρι τα 1800Ohm.

Για ρεύμα στην έξοδο τα 30mA ο πίνακας τιμών της συνάρτησης ρεύματος εξόδου με αντίσταση φορτίου φαίνεται παρακάτω.

Πίνακας 3-5 Ρεύμα εξόδου σε συνάρτηση με την αντίσταση φορτίου για ρεύμα εξόδου 30mA.

A/A	Έξοδος σε mA	φορτίο σε Ohm
1	26,89	1
2	26,88	10
3	26,86	50
4	26,84	100
5	26,79	200
6	26,651	500
7	26,5	800
8	26,41	1000
9	26,32	1200
10	25,56	1400
11	22,72	1600

Η αντίστοιχη γραφική παράσταση φαίνεται στο παρακάτω σχήμα.



Σχήμα 3-12 Γραφική ρεύματος εξόδου σε συνάρτηση με την αντίσταση φορτίου για ρεύμα 30mA

Εδώ παρατηρείται ότι το κύκλωμα δεν μπορεί να εργαστεί ως πηγή ρεύματος. Τα επιθυμητά 30 mA δεν τα παίρνουμε ποτέ από το κύκλωμα, ενώ η χαρακτηριστική δεν είναι καθόλου σταθερή όπως φαίνεται στο σχήμα.

Η τελευταία προσομοίωση που θα παρουσιαστεί εδώ και αξίζει να αναφερθεί είναι η περίπτωση όπου η τάση τροφοδοσίας του τελεστικού OPA445AP γίνεται πολύ μεγάλη, εδώ βάλουμε 100Volt. Αυτή φυσικά είναι μια φανταστική περίπτωση γιατί ο OPA445AP δεν μπορεί να τροφοδοτηθεί με τάση μεγαλύτερη των 45Volt (συμμετρικά) όπως φαίνεται στα φύλλα δεδομένων. Στην προσομοίωση όμως το κύκλωμα δουλεύει κανονικά και εμείς θα παρουσιάσουμε αυτή την περίπτωση για να βγάλουμε ένα ξεκάθαρο συμπέρασμα για το πώς επηρεάζει η τάση τροφοδοσίας του τελεστικού ενισχυτή την συμπεριφορά της πηγής ρεύματος.

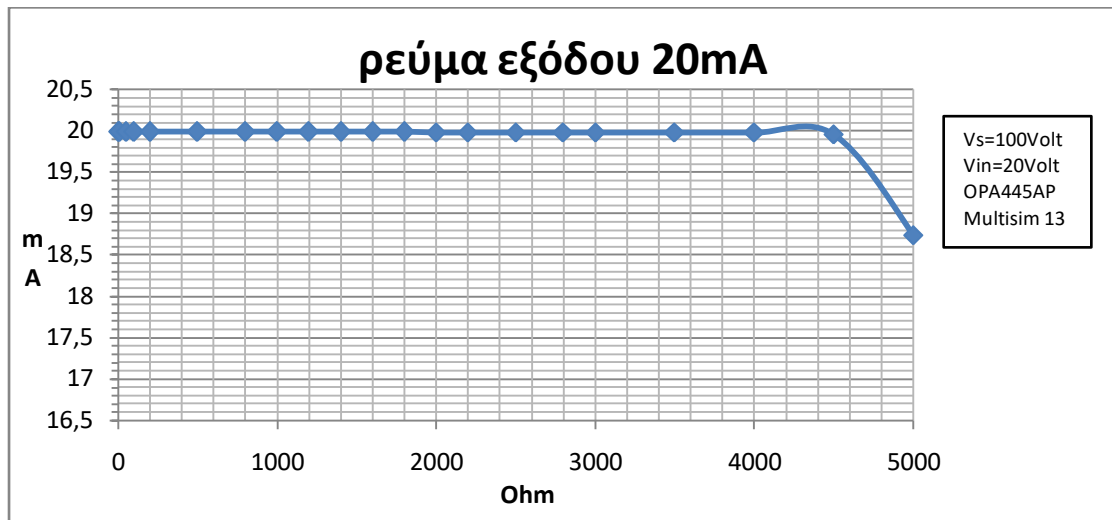
Παρακάτω φαίνεται ο πίνακας τιμών για ρεύμα εξόδου 20 mA και τάση τροφοδοσίας του τελεστικού τα 100Volt. Προσοχή, αυτό δεν μπορεί να εφαρμοστεί στην πράξη για τον OPA445AP. (μέγιστη τάση τροφοδοσίας για τον συγκεκριμένο τελεστικό είναι τα 45 Volt).

Πίνακας 3-6 Ρεύμα εξόδου(20mA) σε συνάρτηση με την αντίσταση φορτίου για τάση τροφοδοσίας του τελεστικού τα 100Volt (δεν γίνεται στην πράξη).

A/A	Έξοδος σε mA	φορτίο σε Ohm	A/A	Έξοδος σε mA	φορτίο σε Ohm
1	19,99	1	14	19,98	2200
2	19,99	10	15	19,98	2500
3	19,99	50	16	19,98	2800
4	19,99	100	17	19,98	3000
5	19,99	200	18	19,98	3500
6	19,99	500	19	19,98	4000
7	19,99	800	20	19,96	4500

8	19,99	1000	21	18,74	5000
9	19,99	1200	22	17,07	5500
10	19,99	1400			
11	19,99	1600			
12	19,99	1800			
13	19,98	2000			

Η αντίστοιχη χαρακτηριστική ρεύματος εξόδου συναρτήσει της αντίστασης φορτίου φαίνεται στο παρακάτω σχήμα.

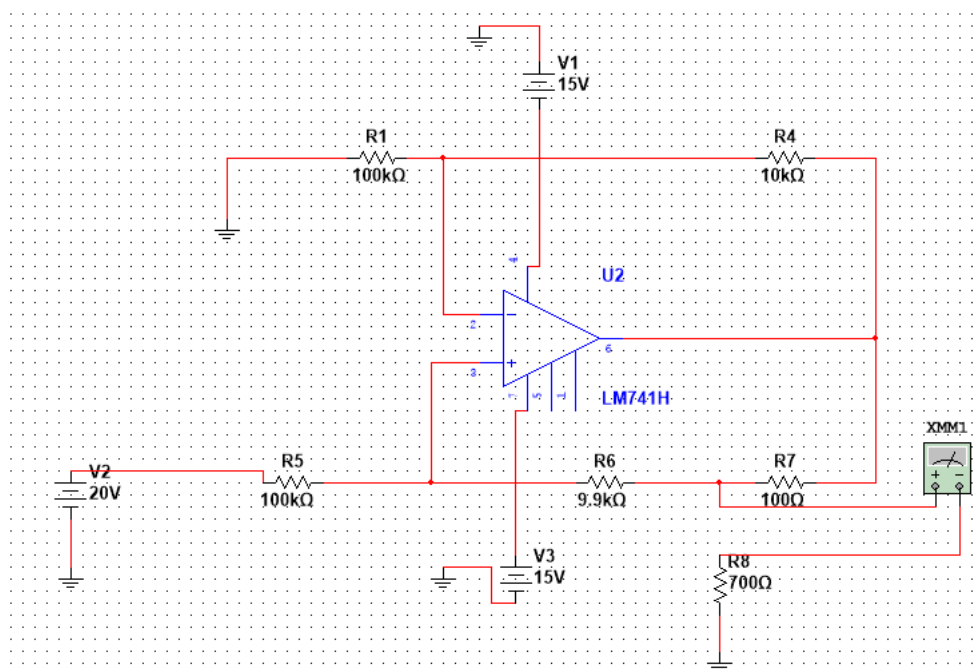


Σχήμα 3-13 Γραφική ρεύματος εξόδου σε συνάρτηση με την αντίσταση φορτίου για ρεύμα 20mA και τάση τροφοδοσίας του τελεστικού τα 100Volt. (Προσοχή! Δεν μπορεί να εφαρμοστεί στην πράξη.)

Παρατηρούμε ότι τώρα η πηγή ρεύματος δουλεύει καλά μέχρι και τα 4000Ohm. Συγκρίνοντας αυτή την γραφική με το σχήμα 3-11 όπου η πηγή ρεύματος δουλεύει σωστά μέχρι τα 1800Ohm βλέπουμε ξεκάθαρα ότι η τάση τροφοδοσίας του τελεστικού καθορίζει τα όρια λειτουργίας της πηγής ρεύματος, κάτι που είναι αναμενόμενο, αφού οποιοσδήποτε τελεστικός ενισχυτής δεν μπορεί να δώσει τάση στην έξοδό του παραπάνω από την τάση με την οποία τροφοδοτείται. Άρα γενικά όσο μεγαλύτερη είναι η τάση τροφοδοσίας του τελεστικού ενισχυτή, τόσο πιο καλή πηγή ρεύματος μπορούμε να κατασκευάσουμε επιλέγοντάς τον. Επίσης πρέπει να αναφερθεί εδώ ότι αν δεν θέλουμε η πηγή ρεύματός να λειτουργεί συμμετρικά, δηλαδή να είναι και πηγή και καταβόθρα ρεύματος μπορούμε να τροφοδοτήσουμε τον τελεστικό ενισχυτή μας μη συμμετρικά. Έτσι αν θέλουμε ο τελεστικός μας να δίνει “θετικό” ρεύμα μόνο, δηλαδή να είναι πηγή ρεύματος, μπορούμε να τον τροφοδοτήσουμε με 0Volt στην αρνητική τροφοδοσία και έως +80 με 90 Volt (αναφερόμαστε στον OPA445AP) στην θετική τροφοδοσία, οπότε το κύκλωμα της πηγής ρεύματος θα έχει καλύτερες επιδόσεις. Στο κύκλωμα που υλοποιήθηκε σε αυτή την εργασία ο τελεστικός ενισχυτής μας τροφοδοτείται συμμετρικά.

3.4.2. Προσομοίωση Improved Howland με τον LM741

Το κύκλωμα που προσομοιώσαμε στο Multisim για αυτή την ενότητα φαίνεται στο παρακάτω σχήμα. Εδώ χρησιμοποιούμε το ίδιο κύκλωμα improved Howland απλώς αντί για τον OPA445AP χρησιμοποιούμε ένα κοινό τελεστικό ενισχυτή τον LM741. Αυτό γίνεται για να καταλάβει ο αναγνώστης ότι το κύκλωμα improved Howland χρειάζεται ιδιαίτερο τελεστικό ενισχυτή με συγκεκριμένες επιθυμητές ιδιότητες και δεν είναι σωστό να το υλοποιήσουμε με οποιοδήποτε τελεστικό. Οι γραφικές που θα παρουσιαστούν παρακάτω είναι αντίστοιχες με αυτές που παρουσιάστηκαν στην ενότητα 3.4.1.



Σχήμα 3-14 Improved Howland με τον τελεστικό LM741

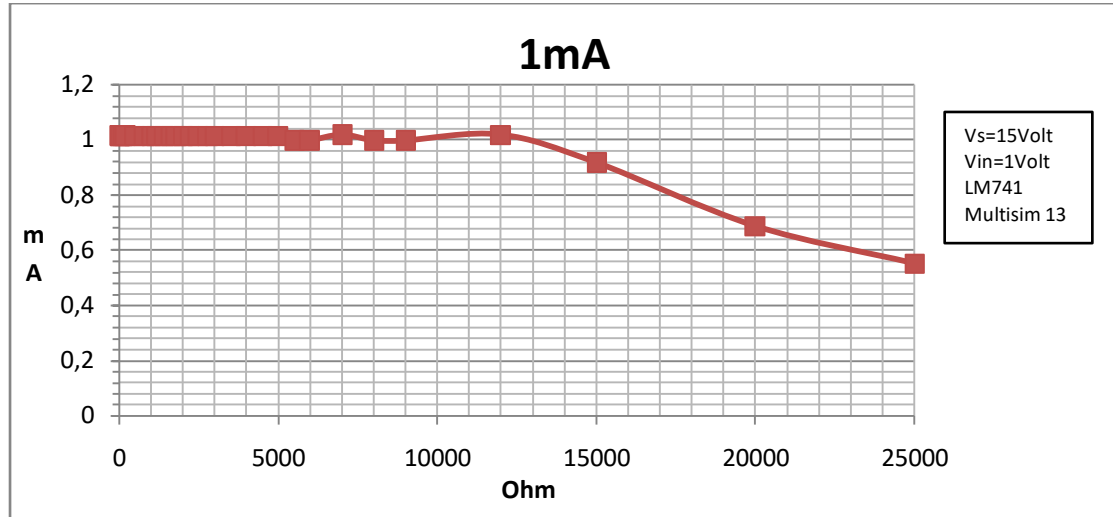
Για τιμή ρεύματος εξόδου το 1mA ο πίνακας τιμών της γραφικής ρεύματος εξόδου σε συνάρτηση με το φορτίο φαίνεται παρακάτω.

Πίνακας 3-7 Ρεύμα εξόδου(1mA) σε συνάρτηση με το φορτίο για τον τελεστικό LM741.

A/A	Έξοδος σε mA	Φορτίο σε Ohm	A/A	Έξοδος σε mA	Φορτίο σε Ohm
1	1,015	1	16	1,017	2800
2	1,015	10	17	1,017	3000
3	1,015	50	18	1,017	3500
4	1,015	100	19	1,017	4000
5	1,015	200	20	1,017	4500
6	1,016	500	21	1,017	5000
7	1,016	800	22	0,998	5500
8	1,016	1000	23	0,998	6000
9	1,016	1200	24	1,018	7000
10	1,016	1400	25	0,998	8000
11	1,016	1600	26	0,998	9000
12	1,016	1800	27	1,02	12000
13	1,016	2000	28	0,918	15000

14	1,016	2200	29	0,689	20000
15	1,016	2500	30	0,554	25000

Η γραφική παράσταση του παραπάνω πίνακα δεδομένων φαίνεται εδώ.



Σχήμα 3-15 Γραφική ρεύματος εξόδου σε συνάρτηση με την αντίσταση φορτίου για ρεύμα 1mA, και τελεστικό τον LM741.

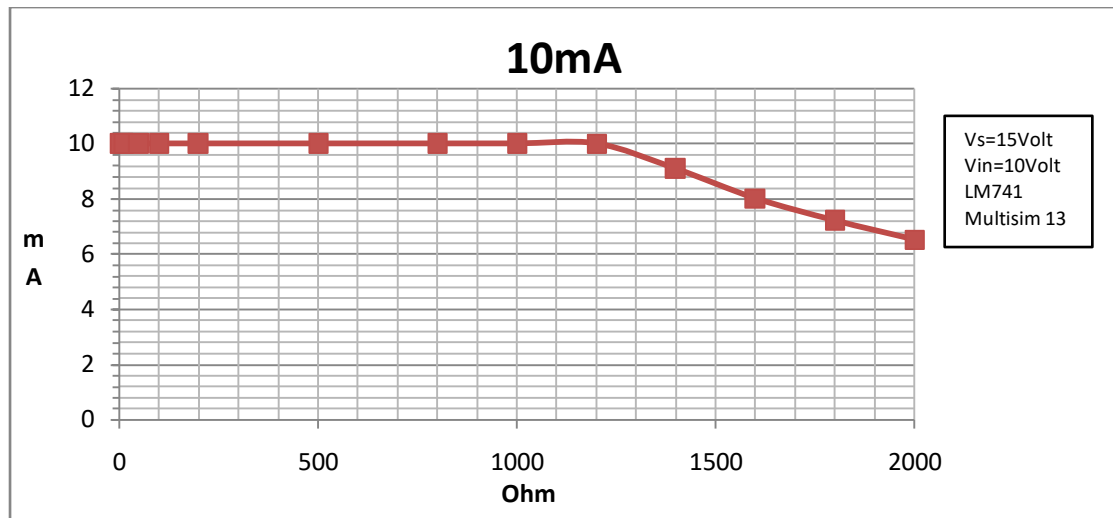
Όπως φαίνεται στο σχήμα 26 ότι η πηγή ρεύματος λειτουργεί σωστά ως τα 12000Ohm.

Για ρεύμα στην έξοδο τα 10mA ο πίνακας τιμών της συνάρτησης ρεύματος εξόδου με αντίσταση φορτίου φαίνεται παρακάτω.

Πίνακας 3-8 Ρεύμα εξόδου(10mA) σε συνάρτηση με το φορτίο για τον τελεστικό LM741.

A/A	Έξοδος σε mA	Φορτίο σε Ohm	A/A	Έξοδος σε mA	Φορτίο σε Ohm
1	10,01	1	8	10,01	1000
2	10,01	10	9	10,02	1200
3	10,01	50	10	9,11	1400
4	10,01	100	11	8,05	1600
5	10,01	200	12	7,22	1800
6	10,01	500	13	6,54	2000
7	10,01	800			

Η γραφική παράσταση του παραπάνω πίνακα φαίνεται στο σχήμα 3-16.



Σχήμα 3-16 Γραφική ρεύματος εξόδου σε συνάρτηση με την αντίσταση φορτίου για ρεύμα 10mA, και τελεστικό τον LM741.

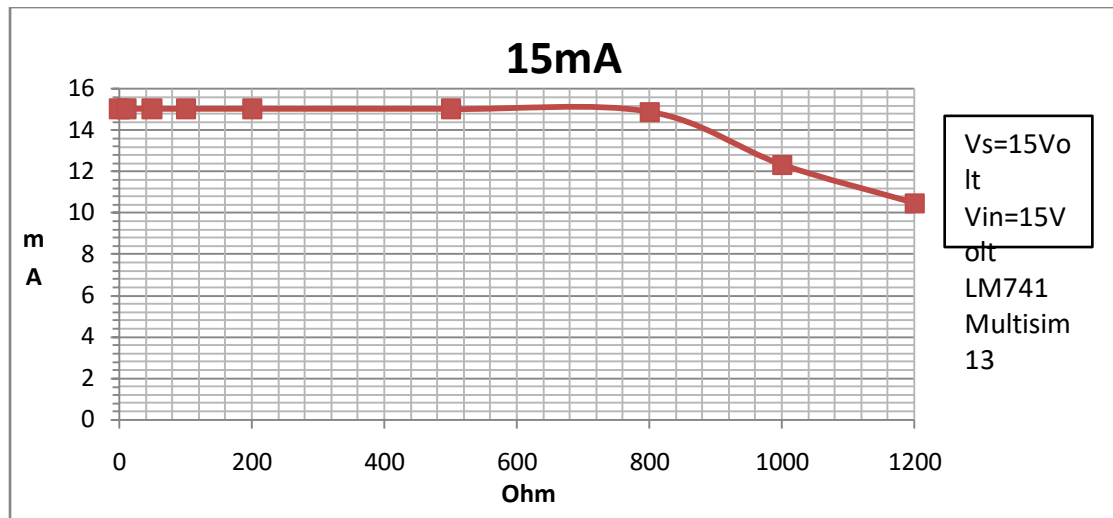
Παρατηρούμε ότι η πηγή ρεύματος λειτουργεί με τον αναμενόμενο τρόπο μέχρι 1200Ohm αντίστασης φορτίου.

Για ρεύμα στην έξοδο τα 15mA ο πίνακας τιμών της συνάρτησης ρεύματος εξόδου με αντίσταση φορτίου φαίνεται παρακάτω.

Πίνακας 3-9 Ρεύμα εξόδου(10mA) σε συνάρτηση με το φορτίο για τον τελεστικό LM741.

A/A	Έξοδος σε mA	Φορτίο σε Ohm
1	15,02	1
2	15,02	10
3	15,02	50
4	15,02	100
5	15,02	200
6	15,02	500
7	14,9	800
8	12,36	1000
9	10,49	1200

Παρακάτω φαίνεται η γραφική παράσταση των τιμών του πίνακα για ρεύμα στην έξοδο τα 15mA.



Σχήμα 3-17 Γραφική ρεύματος εξόδου σε συνάρτηση με την αντίσταση φορτίου για ρεύμα 15mA, και τελεστικό τον LM741.

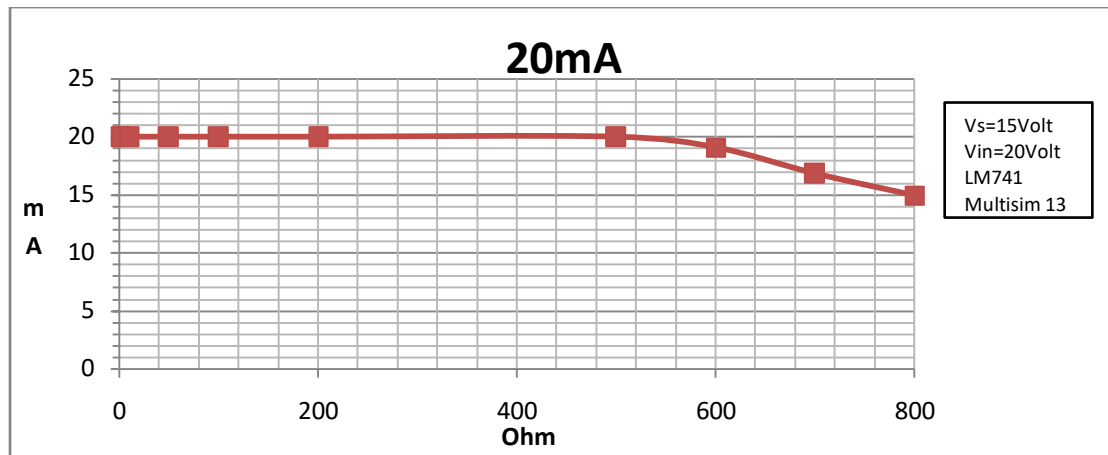
Η πηγή ρεύματος λειτουργεί σωστά ως τα 800 Ohm αντίστασης φορτίου.

Τέλος στον παρακάτω πίνακα φαίνονται οι τιμές ρεύματος εξόδου σε σχέση με την αντίσταση φορτίου για ρεύμα εξόδου τα 20mA.

Πίνακας 3-10 Ρεύμα εξόδου(20mA) σε συνάρτηση με το φορτίο για τον τελεστικό LM741.

A/A	Έξοδος σε mA	Φορτίο σε Ohm
1	20	1
2	20	10
3	20	50
4	20	100
5	20	200
6	20	500
7	19,12	600
8	16,84	700
9	15	800

Η αντίστοιχη γραφική φαίνεται παρακάτω.



Σχήμα 3-18 Γραφική ρεύματος εξόδου σε συνάρτηση με την αντίσταση φορτίου για ρεύμα 20mA, και τελεστικό τον LM741

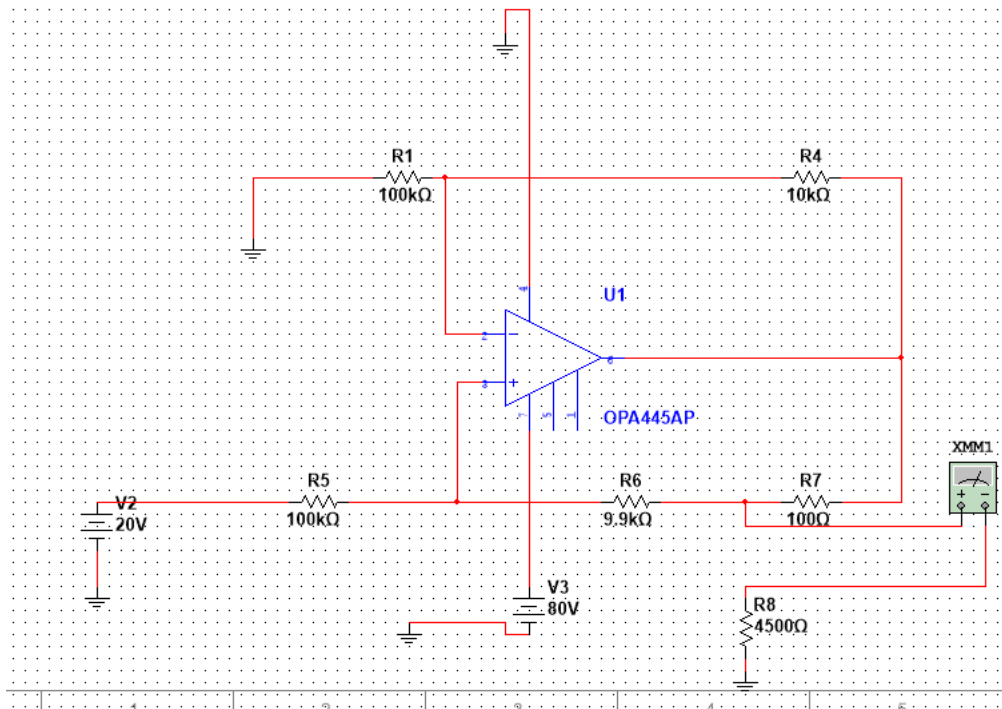
Η πηγή ρεύματος εδώ λειτουργεί μέχρι τα 500Ohm.

3.5.Συμπεράσματα και τρόποι βελτιστοποίησης του κυκλώματος Improved Howland

Το βασικό συμπέρασμα που βγαίνει μελετώντας αυτές τις καμπύλες είναι ότι η τάση τροφοδοσίας του τελεστικού ενισχυτή είναι το βασικό χαρακτηριστικό που κοιτάμε όταν θέλουμε να κατασκευάσουμε μια πηγή ρεύματος Howland. Έτσι ο OPA445AP είναι πολύ καλύτερος για υλοποίηση κυκλώματος πηγής ρεύματος Howland από τον κοινό τελεστικό LM741. Αν δούμε τα φύλλα δεδομένων του τελεστικού LM741 το μέγιστο ρεύμα που μπορεί να δώσει στην έξοδό του είναι 25mA. (short circuit current) η τιμή αυτή είναι ίδια στον τελεστικό OPA445AP. Επομένως περιμένουμε χρησιμοποιώντας αυτούς τους τελεστικούς ένα μέγιστο ρεύμα εξόδου τα 25mA. Όμως για να επιτευχθεί αυτό το ρεύμα στην έξοδο ο τελεστικός πρέπει να δώσει αντίστοιχα μεγάλη τάση ($V=I \cdot R$), άρα πάλι συμπεραίνουμε ότι ο τελεστικός που πρέπει να επιλέξουμε για την κατασκευή ενός κυκλώματος improved Howland πρέπει να έχει όσο μεγαλύτερη γίνεται τάση τροφοδοσίας. Ένας τρόπος βελτιστοποίησης του κυκλώματος είναι αυτός που αναφέραμε στο τέλος της παραγράφου 3.4.1 δηλαδή **να τροφοδοτήσουμε τον τελεστικό ενισχυτή μας μη συμμετρικά**. Αυτό θα βελτιώσει σημαντικά την απόδοση της πηγής ρεύματος όμως το ρεύμα εξόδου τώρα θα έχει μόνο μια φορά (θετική ή αρνητική). Ένας καλύτερος τελεστικός από τον OPA445AP είναι ο **OPA454**. Αυτός έχει τάση τροφοδοσίας (συμμετρική) τα 50Volt. Επομένως θα δώσει καλύτερες χαρακτηριστικές από τον OPA445AP είναι όμως πιο ακριβός. Βλέπουμε λοιπόν ότι υπάρχουν τρόποι περεταίρω βελτίωσης του κυκλώματός μας. Τέλος μια μέθοδος βελτίωσης είναι και το λεγόμενο **Bootstrapping**. Αυτή η διαδικασία δεν αυξάνει το ρεύμα εξόδου της πηγής μας αλλά αναφέρεται στη βιβλιογραφία ότι αυξάνει αρκετά (κατά μια τάξη μεγέθους) το εύρος αντιστάσεων φορτίου στο οποίο μπορεί να δουλέψει η πηγή μας. Παρακάτω θα εξετάσουμε συνοπτικά αυτούς τους τρόπους βελτίωσης του κυκλώματος.

3.5.1. Ασύμμετρη τροφοδοσία του τελεστικού OPA445AP

Σε αυτή την ενότητα θα προσομοιώσουμε το κύκλωμα Improved Howland τροφοδοτώντας ασύμμετρα τον τελεστικό OPA445AP. Η μέγιστη ασύμμετρη τροφοδοσία στον τελεστικό μας είναι τα 90Volt, σύμφωνα με τα φύλλα δεδομένων. Εμείς θα τροφοδοτήσουμε τον τελεστικό με 80Volt για να είμαστε σίγουροι ότι το κύκλωμα που θα προσομοιώσουμε δεν έχει κίνδυνο να κάψει τον τελεστικό μας αν εφαρμοστεί στην πραγματικότητα. Στο παρακάτω σχήμα φαίνεται το κύκλωμα στο multisim 13.



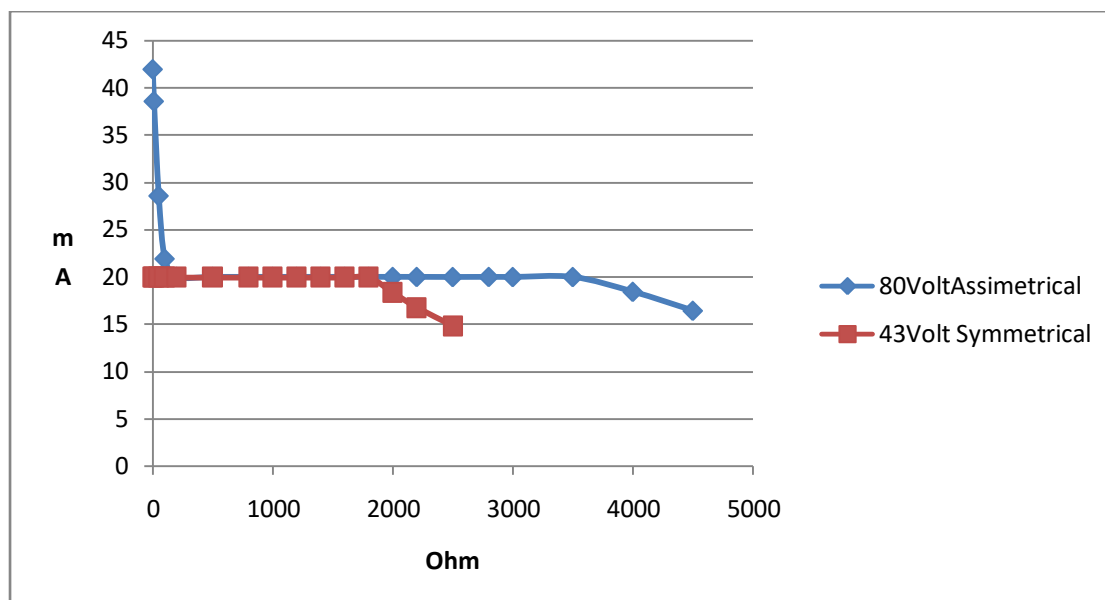
Σχήμα 3-19 Κύκλωμα improved Howland με ασύμμετρη τροφοδοσία +80Volt.

Η προσομοίωση έγινε για τάση εισόδου 20Volt επομένως το ρεύμα εξόδου θα είναι 20 mA. Ο πίνακας τιμών του ρεύματος εξόδου σε συνάρτηση με την αντίσταση φορτίου φαίνεται παρακάτω.

Πίνακας 3-11 Ρεύμα εξόδου(20mA) σε συνάρτηση με το φορτίο για ασύμμετρη τροφοδοσία του OPA445AP.

A/A	Έξοδος σε mA	Φορτίο σε Ohm	A/A	Έξοδος σε mA	Φορτίο σε Ohm
1	41,92	1	11	20	1600
2	38,542	10	12	20	1800
3	28,56	50	13	20	2000
4	21,919	100	14	20	2200
5	20,008	200	15	19,99	2500
6	20	500	16	20	2800
7	20	800	17	19,99	3000
8	20	1000	18	19,99	3500
9	20	1200	19	18,43	4000
10	20	1400	20	16,43	4500

Η γραφική παράσταση των παραπάνω δεδομένων φαίνεται στο παρακάτω σχήμα. Εδώ έχουμε προσθέσει και τη γραφική της προσομοίωσης για 20mA ρεύμα εξόδου και 43Volt τροφοδοσία του τελεστικού OPA445AP για σύγκριση

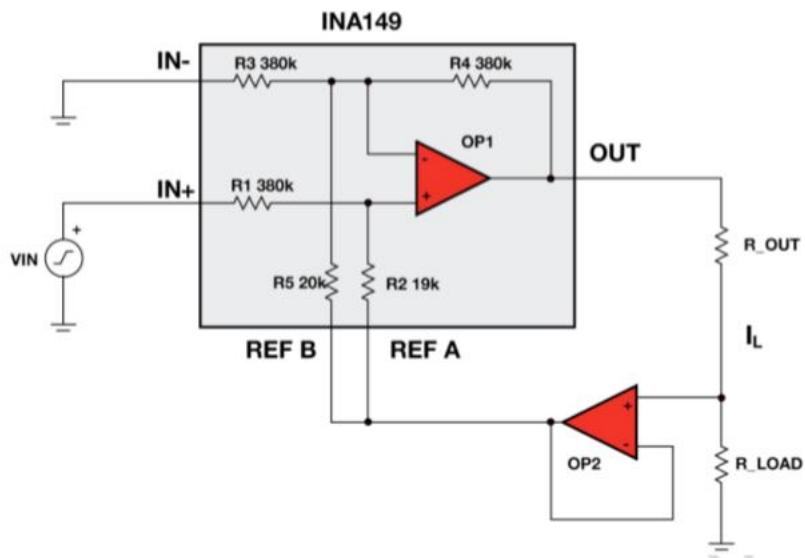


Σχήμα 3-20 Προσομοίωση με 80Volt Ασύμμετρη τροφοδοσία και 43Volt συμμετρική με τον OPA445AP.

Η τάση εισόδου για τα κυκλώματα που αναλύονται στη γραφική του σχήματος 3-20 είναι τα 20Volt. Όπως φαίνεται από τον πίνακα τιμών, το κύκλωμα δουλεύει σαν πηγή ρεύματος από τα 2000Ohm περίπου μέχρι τα 3500 Ohm αντίσταση φορτίου, δεν δουλεύει σωστά για μικρά φορτία της τάξης των 1-100Ohm. Παρατηρείται επίσης στο σχήμα 3-20 ότι το κύκλωμα δουλεύει καλύτερα για μεγαλύτερα φορτία έως 3500 Ohm, κάτι που δεν μπορεί να καταφέρει το κύκλωμα με τα 43Volt συμμετρικής τροφοδοσίας.

3.5.2. Bootstrapping στο κύκλωμα Improved Howland Current Pump με τον OPA445AP.

Στο παρακάτω σχήμα φαίνεται το βελτιωμένο με τη διαδικασία Bootstrapping κύκλωμα improved Howland. Από τις παρακάτω γραφικές παραστάσεις γίνεται αντιληπτό ότι η απόδοση του κυκλώματος βελτιώνεται σημαντικά με αυτή τη διαδικασία. Αν κάποιος θέλει να υλοποιήσει το κύκλωμα του σχήματος 3-21, στη θέση των τελεστικών ενισχυτών OP1 και OP2 προτείνεται να χρησιμοποιήσει τον τελεστικό OPA445AP ή τον OPA454.

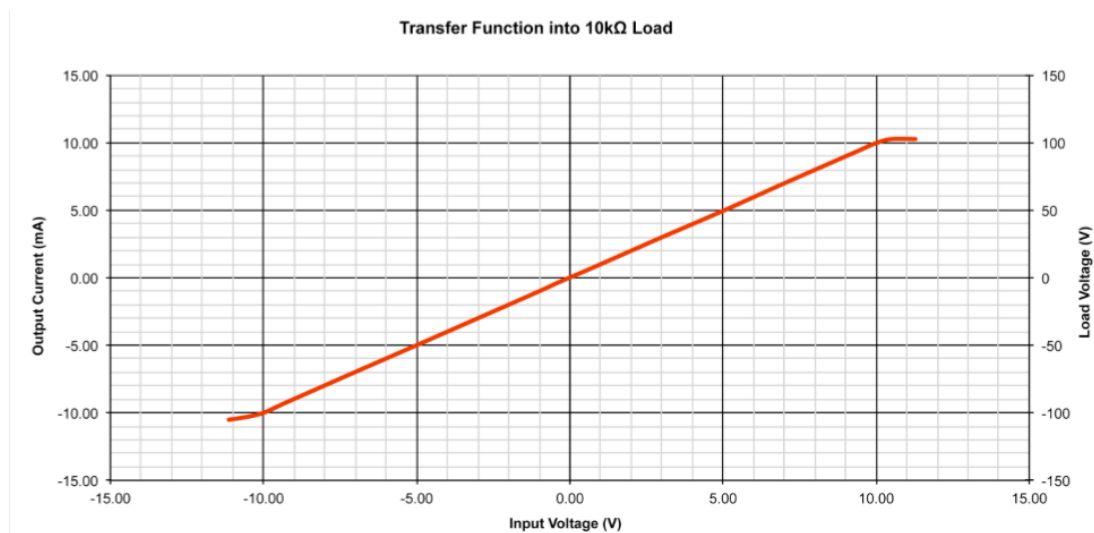


Σχήμα 3-21 Bootstrapping Improved Howland Current Pump.

Σύμφωνα με τη βιβλιογραφία το κύκλωμα αυτό μπορεί να δώσει πολύ περισσότερο ρεύμα στην έξοδο και η τάση συμμόρφωσης του τελεστικού αυξάνεται σημαντικά.

Η μέγιστη τάση στην έξοδο του τελεστικού αναφέρεται να είναι 125Volt και μέγιστο ρεύμα στην έξοδο για φορτίο 10kOhm είναι 11.4mA.

Η συνάρτηση μεταφοράς σε φορτίο 10kOhm (δηλαδή η σχέση μεταξύ της τάσης εισόδου και του ρεύματος εξόδου) φαίνεται στο παρακάτω σχήμα.

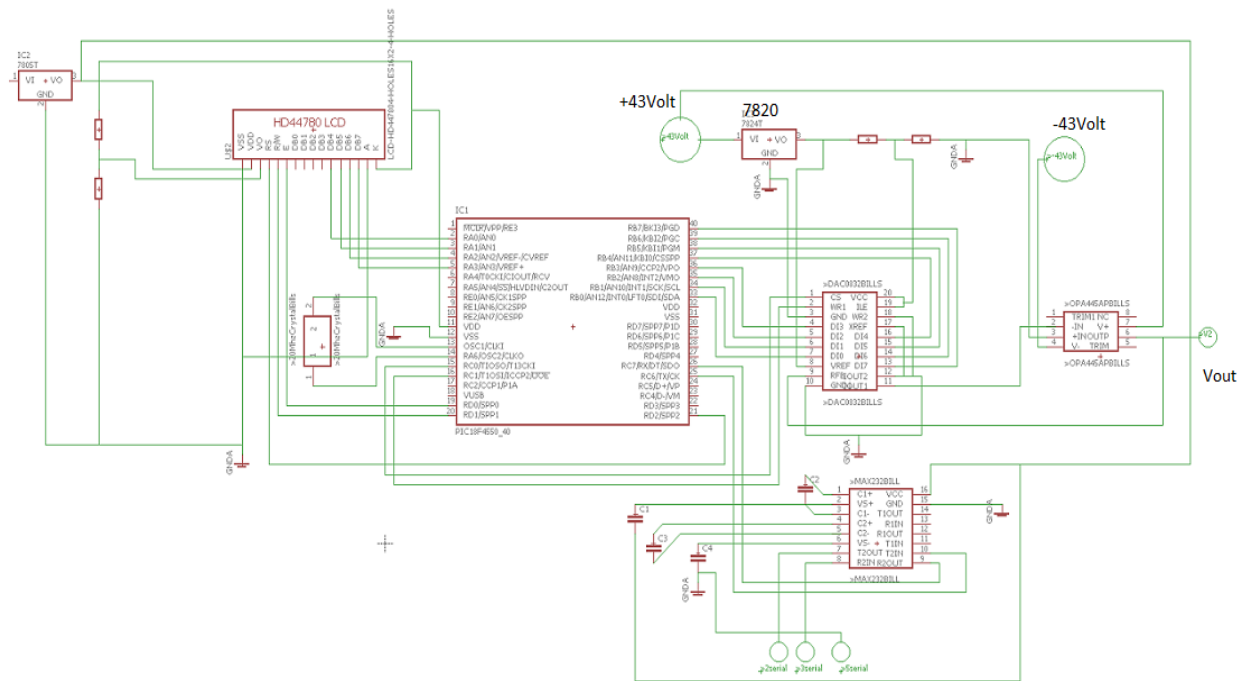


Σχήμα 3-22 Σχέση μεταξύ τάσης εισόδου και ρεύματος εξόδου για φορτίο 10kOhm.

4. Hardware ψηφιακού ελέγχου της τάσης εισόδου του κυκλώματος improved Howland.

Σε αυτό το κεφάλαιο θα ασχοληθούμε με το hardware που αναπτύξαμε για τον ψηφιακό έλεγχο της τάσης εισόδου του κυκλώματος Improved Howland Current Pump που φαίνεται στο σχήμα 3-7. Όταν λέμε τάση εισόδου, αναφερόμαστε στην τάση V2 του σχήματος 3-7. Για να κάνουμε τις προσομοιώσεις στο Multisim στη θέση V2 στο σχήμα 3-7 βάλαμε μια πηγή συνεχούς τάσης. Εδώ στη θέση αυτής της πηγής θα προσαρμοστεί η έξοδος του κυκλώματος ψηφιακού ελέγχου που θα περιγράψουμε παρακάτω.

Το κύκλωμα του ψηφιακού ελέγχου φαίνεται στο παρακάτω σχήμα. Το σχήμα έγινε με την εφαρμογή EAGLE 7.6.0. Για πληροφορίες σχετικά με την κατασκευή του κυκλώματος του σχήματος 4-1 με το EAGLE μπορεί να ανατρέξει κανείς στα παραρτήματα της εργασίας.



Σχήμα 4-1 Κύκλωμα ψηφιακού ελέγχου της τάσης εισόδου στο κύκλωμα improved Howland

Στις παρακάτω ενότητες θα περιγραφεί αναλυτικά κάθε κομμάτι του παραπάνω σχήματος. Θα ασχοληθούμε τόσο με το hardware όσο και με το software που χρειάστηκε για την υλοποίηση και λειτουργία του κυκλώματος.

Ας δούμε όμως εδώ τα βασικά χαρακτηριστικά του κυκλώματος ψηφιακού ελέγχου που φαίνεται στο σχήμα 4-1. Στο κάτω μέρος του κυκλώματος, υπάρχουν τρεις κύκλοι που είναι η είσοδος του κυκλώματος ψηφιακού ελέγχου και γίνεται μέσω της σειριακής θύρας. Έτσι ένας υπολογιστής με κατάλληλο software που θα περιγραφεί σε επόμενο κεφάλαιο επικοινωνεί με το κύκλωμα ψηφιακού ελέγχου και μεταβάλλει την τάση εξόδου Vout η οποία είναι η τάση εισόδου στο κύκλωμα Improved Howland.

Για την επικοινωνία μέσω της σειριακής θύρας αλλά και για τον έλεγχο των επιμέρους κυκλωμάτων του σχήματος 4-1 χρησιμοποιούμε έναν μικροελεγκτή, τον PIC18F4550 της Microchip.

Ο μικροελεγκτής PIC18F4550 της Microchip είναι η καρδιά του κυκλώματος. Αυτός είναι υπεύθυνος για την επικοινωνία με τη σειριακή θύρα, μέσω του MAX323 ολοκληρωμένου κυκλώματος. Δέχεται εντολές από τη σειριακή θύρα για τον έλεγχο της τάσης Vout .

Ο έλεγχος της τάσης Vout γίνεται μέσω του DAC0832. Ο μικροελεγκτής αφού λάβει μέσω σειριακής θύρας την τιμή που πρέπει να πάρει η έξοδος Vout βγάζει τα κατάλληλα σήματα ελέγχου προς το DAC0832 το οποίο στη συνέχεια μετατρέπει το ψηφιακό σήμα που δέχεται από τον PIC18F4550 σε αναλογικό ρεύμα. Το ρεύμα αυτό στέλνεται στον τελεστικό ενισχυτή OPA445AP όπου μετατρέπεται σε τάση και βγαίνει στην έξοδο του κυκλώματος (Vout).

Επιπλέον ο μικροελεγκτής συνδέεται με μια οθόνη LCD 2 σειρών 16 χαρακτήρων, μέσω της οποίας μας πληροφορεί για το τι γίνεται στο κύκλωμα. Έτσι η LCD είναι το μέσο, μέσω του οποίου γίνεται η επικοινωνία του κυκλώματος ελέγχου με το χρήστη. Ο χρήστης μέσω της LCD πληροφορείται για την σωστή λειτουργία του κυκλώματος. Κάθε φορά που μια τιμή τάσης Vout στέλνεται από το ψηφιακό κύκλωμα στο κύκλωμα improved Howland στην LCD οθόνη εμφανίζεται το ρεύμα που πρέπει να πάρουμε στην έξοδο του κυκλώματος improved Howland κατά προσέγγιση.

Στο κύκλωμα του σχήματος 4-1 βλέπουμε ότι υπάρχουν δύο ακόμη ολοκληρωμένα κυκλώματα, το LM7805 και το LM7820.

Το LM7805 χρησιμοποιείται για την τροφοδοσία του PIC18F4550 και της LCD οθόνης. Η λειτουργία του LM7805 είναι να δέχεται μια τάση στην είσοδό του (εμείς το τροφοδοτούμε μέσω μιας 9VotI μπαταρίας) και να βγάζει στην έξοδό του 5Volt συνεχή τάση, η οποία μένει σταθερή. Αυτά τα 5Volt όπως είπαμε τροφοδοτούν τον PIC και την LCD οθόνη.

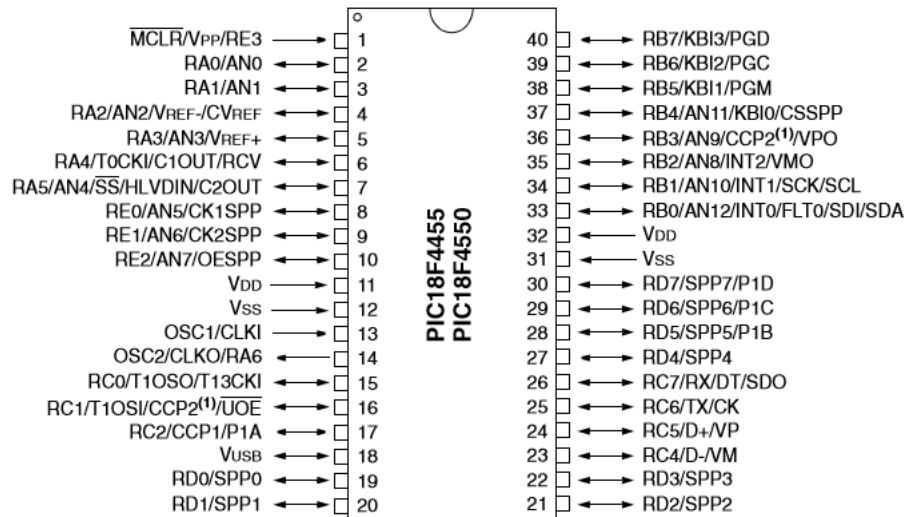
Το LM7820 είναι ένα ολοκληρωμένο που τροφοδοτούμε με μια συνεχή τάση (μεγαλύτερη των 20 Volt) μας δίνει στην έξοδό του μια σταθερή τάση ίση με 20Volt. Για να τροφοδοτήσουμε το LM7820 χρησιμοποιούμε τα +43Volt από το τροφοδοτικό που φτιάξαμε για την πηγή ρεύματος Improved Howland. Βλέπε και σχήμα 2-1. Τα 20Volt που παρέχονται από το LM7820 χρησιμοποιούνται ως τάση αναφοράς (Vref) για το DAC0832. Περισσότερα θα πούμε στη σχετική ενότητα.

Τέλος ο κρύσταλλος 20Mhz χρησιμοποιείται για τον χρονισμό του PIC18F4550. Εδώ πρέπει να αναφέρουμε ότι δεν είναι υποχρεωτικό να χρονίσουμε το κύκλωμά μας στα 20Mhz. Θα μπορούσαμε να το χρονίσουμε με κρύσταλλο στα 4,8 10Mhz η οποιοδήποτε άλλη τιμή μας βολεύει, ή και να χρησιμοποιήσουμε εσωτερικό χρονισμό. Η επιλογή του κρυστάλλου είναι καθαρά δική μας επιλογή και δεν επηρεάζεται από κανένα στοιχείο του κυκλώματος. Έχοντας όμως επιλέξει τα 20Mhz εξωτερικού χρονισμού, πρέπει να προσαρμόσουμε κατάλληλα τις ρουτίνες του που θα γράψουμε στο software ώστε να επιτύχουμε τους σωστούς χρόνους που χρειάζονται για την επικοινωνία του μικροελεγκτή μας με τα περιφερειακά ολοκληρωμένα κυκλώματα.

4.1.0 μικροελεγκτής PIC18F4550 της Microchip.

Στην παρακάτω εικόνα φαίνεται ο μικροελεγκτής PIC4550 της Microchip. Όπως παρατηρείται είναι ένας 40 pin μικροελεγκτής.

40-Pin PDIP



Σχήμα 4-2 Ο μικροελεγκτής PIC18F4550 της Microchip.

Ο μικροελεγκτής PIC18F4550 είναι ένας 8bit 40PIN μικροελεγκτής της εταιρίας MICROCHIP. Εδώ θα περιγράψουμε τα χαρακτηριστικά του μικροελεγκτή που μας ενδιαφέρουν για την εργασία μας. Για περισσότερες πληροφορίες σχετικά με τον 18F4550 ο αναγνώστης μπορεί να αναφερθεί στα φύλλα δεδομένων του.

Ο PIC18F4550 αποτελείται από 5 θύρες A,B,C, D και E. Η θύρα A είναι 7 Bit. Αποτελείται από 7 Pins που μπορούν να χρησιμοποιηθούν τόσο για είσοδο όσο και για έξοδο, ανάλογα με τις προτιμήσεις μας. Από αυτά τα 7 Pins, τα 6 βρίσκονται όπως φαίνεται στο σχήμα 4-2 στα PINS 2 έως και 7 της συσκευασίας PDIP του μικροελεγκτή. Το έβδομο PIN της θύρας A βρίσκεται στο PIN14 του μικροελεγκτή. Το PIN2 είναι το RA0 (least significant bit) και το PIN7 είναι το RA6 (most significant bit).

Αυτή τη θύρα (A) θα χρησιμοποιήσουμε για την επικοινωνία με την LCD οθόνη 2 σειρών 16 χαρακτήρων. Όπως θα δούμε στο εδάφιο για την LCD, για να επικοινωνήσουμε με αυτή χρειαζόμαστε 4 PINS για μεταφορά δεδομένων και 3 PINS για τον έλεγχο της LCD. Τα PINS για την μεταφορά δεδομένων επιλέξαμε να είναι αυτά της θύρας A του μικροελεγκτή. Έτσι το RA0 είναι το lsb και το RA3 είναι το MSB. Επομένως η θύρα A χρησιμοποιείται ως έξοδος στο κύκλωμά μας.

Η θύρα B του μικροελεγκτή μας έχει εύρος 8bit. Αποτελείται από τα RB0 έως και RB7. Το RB0 βρίσκεται στο ποδαράκι (PIN) 33, το RB1 εντοπίζεται στο PIN 34 και ούτο καθ' εξής έως το ποδαράκι RB7 που εντοπίζεται στο PIN40 του μικροελεγκτή (Δείτε σχήμα 4-2).

Αυτή τη θύρα επιλέξαμε να χρησιμοποιήσουμε για την επικοινωνία με το DAC0832. Το DAC0832 χρειάζεται 8 bit για δίαυλο δεδομένων. Αυτά τα 8 bit δημιουργούν μια 8bit λέξη στην είσοδό του, επομένως το DAC0832 δέχεται μια ψηφιακή λέξη από 0 έως 255 στην είσοδό του. Αυτό το ψηφιακό σήμα το μετατρέπει σε αναλογικό ρεύμα, όπως θα εξηγήσουμε αναλυτικότερα στην ενότητα που αναφέρεται στο DAC0832. Για να δώσουμε αυτή την ψηφιακή λέξη επιλέξαμε να χρησιμοποιήσουμε τη θύρα B του μικροελεγκτή μας. Επομένως η θύρα B ορίζεται και αυτή σαν έξοδος του μικροελεγκτή, και χρησιμοποιούνται όλα τα bits της θύρας. Το PIN 33 (RB0) είναι το LSB και συνδέεται στο PIN7 του DAC0832, ενώ το PIN 40 (RB7) είναι το MSB και συνδέεται στο PIN 13 του DAC0832.

Από τη θύρα C, που είναι μια 8 bit θύρα, του PIC18F4550 χρησιμοποιούμε τα ποδαράκια RC0 (PIN15) και RC1 (PIN16) για τα σήματα ελέγχου του DAC0832. Επίσης στη θύρα C στα ποδαράκια RC7 (PIN 26) και RC6 (PIN 25) υλοποιείται με hardware μέσα στον μικροελεγκτή η σειριακή επικοινωνία. Έτσι το ποδαράκι RC7 (PIN 26) είναι το RX (receive) και το ποδαράκι RC6 (PIN 25) είναι το TX (transmit) της σειριακής επικοινωνίας. Αυτά τα ποδαράκια χρησιμοποιούμε για την σειριακή επικοινωνία με το MAX232. Περισσότερα για τη σειριακή επικοινωνία θα αναφέρουμε στην αντίστοιχη ενότητα.

Η θύρα D είναι 8 bit θύρα και βρίσκεται στον μικροελεγκτή μας στα PINS 19 για το RD0 και PIN 20 για το RD1, PIN21 για το RD2, PIN 22 για το RD3 και PIN27 έως PIN30 για τα RD4 έως και RD7.

Εμείς επιλέξαμε να χρησιμοποιήσουμε τα PINS RD0 έως και RD2 ως PIN ελέγχου της LCD οθόνης. Μέσω αυτών των PIN στέλνονται στη LCD οθόνη της Raystar τα σήματα ελέγχου.

Η θύρα E του μικροελεγκτή είναι η τελευταία που παρέχει ο PIC18F4550. (Συνολικά 5 θύρες). Αυτή εντοπίζεται στα PINS 8 έως και 10. Και είναι μια θύρα τριών Bit. Αυτή τη θύρα δεν τη χρησιμοποιούμε στο κύκλωμά μας.

Το PIN12 του μικροελεγκτή συνδέεται με την γή, ενώ το PIN11 συνδέεται στα +5Volt που παίρνουμε από το 7805. Επίσης το PIN 31 συνδέεται στη γή και το PIN 32 στα +5Volt. Αυτά τα PIN είναι υπεύθυνα για την τροφοδοσία του μικροελεγκτή μας.

Στα PIN13 και PIN14 συνδέεται ο εξωτερικός κρύσταλλός μας. Εδώ πρέπει να σημειωθεί ότι ο εξωτερικός κρύσταλλος δεν έχει πολικότητα, επομένως δεν έχει σημασία ποιο ποδαράκι του κρυστάλλου θα αντιστοιχιστεί σε ποιο PIN του μικροελεγκτή.

Για να παρουσιαστούν καλύτερα αυτά που αναφέραμε παραπάνω σχηματίσαμε τον παρακάτω πίνακα που μας δείχνει ποια PINS του μικροελεγκτή χρησιμοποιούνται σε αυτή την εργασία και πως συνδέονται.

Πίνακας 4-1 Τα PIN του PIC18F4550 που χρησιμοποιούμε.

A/A	Θύρα	PIN 18F4550	Χρησιμότητα
1	RA0	PIN2	Δίαυλος Δεδομένων LCD
2	RA1	PIN 3	Δίαυλος Δεδομένων LCD
3	RA2	PIN 4	Δίαυλος Δεδομένων

			LCD
4	RA3	PIN 5	Δίαυλος Δεδομένων LCD
5	RC0	PIN 15	Έλεγχος DAC0832
6	RC1	PIN 16	Έλεγχος DAC0832
7	RD0	PIN 19	Έλεγχος LCD 2 x16
8	RD1	PIN 20	Έλεγχος LCD 2 x16
9	RD2	PIN 21	Έλεγχος LCD 2 x16
10	RC6	PIN 25	Σειριακή επικοινωνία
11	RC7	PIN 26	Σειριακή επικοινωνία
12	RB0	PIN 33	Δίαυλος δεδομένων DAC0832
13	RB1	PIN 34	Δίαυλος δεδομένων DAC0832
14	RB2	PIN 35	Δίαυλος δεδομένων DAC0832
15	RB3	PIN 36	Δίαυλος δεδομένων DAC0832
16	RB4	PIN 37	Δίαυλος δεδομένων DAC0832
17	RB5	PIN 38	Δίαυλος δεδομένων DAC0832
18	RB6	PIN 39	Δίαυλος δεδομένων DAC0832
19	RB7	PIN 40	Δίαυλος δεδομένων DAC0832
20	OSC1	PIN 13	Κρύσταλλος
21	OSC2	PIN 14	Κρύσταλλος
22	VDD	PIN 11	Τροφοδοσία +5Volt
23	Vss	PIN 12	Ground
24	VDD	PIN 32	Τροφοδοσία +5Volt
25	Vss	PIN 31	Ground

Επομένως τελικά χρησιμοποιούμε 25 από τα 40 PINS του μικροελεγκτή PIC18F4550. Ο αριθμός των PIN που χρειαζόμαστε για την υλοποίηση του κυκλώματος ψηφιακού ελέγχου είναι το βασικό κριτήριο για τον μικροελεγκτή που πρέπει να επιλέξουμε. Ο PIC18F4550 έχει 40 PIN και επομένως υπερκαλύπτει τις ανάγκες μας. Οποιοσδήποτε μικροελεγκτής με παραπάνω από 25 PINS θα βόλευε για την υλοποίηση του κυκλώματός μας. Το μόνο που πρέπει να προσέξουμε είναι ότι εκτός από τα 25 PINS χρειαζόμαστε έναν μικροελεγκτή που να προσφέρει σειριακή επικοινωνία μέσω hardware (PIN 25 και 26 RX και TX). Επομένως οποιοσδήποτε μικροελεγκτής που ικανοποιεί τα δύο παραπάνω κριτήρια θα μπορούσε να χρησιμοποιηθεί σε αυτή την εργασία. Ένας ακόμη λόγος που επιλέξαμε τον PIC18F4550 είναι ότι μπορεί να προγραμματιστεί χρησιμοποιώντας τον C18 compiler που είναι συμβατός με την οικογένεια 18F μικροελεγκτών της Microchip. Αυτός ο compiler προσφέρεται με βιβλιοθήκες ελέγχου περιφερειακών όπως τη βιβλιοθήκη επικοινωνίας με την LCD οθόνη (που υλοποιείται με software) και την βιβλιοθήκη επικοινωνίας με το MAX232 (σειριακή επικοινωνία) που υλοποιείται με hardware. Επιπλέον αυτός ο

μικροελεγκτής προϋπήρχε στην εργαλειοθήκη μου, και επομένως επέλεξα να τον χρησιμοποιήσω.

Έχοντας αποφασίσει να χρησιμοποιήσουμε τον PIC18F4550 για την υλοποίηση του ψηφιακού ελέγχου, πρέπει να δούμε ποια εργαλεία θα χρησιμοποιήσουμε για τον προγραμματισμό του.

Καταρχήν για τον προγραμματισμό του PIC18F4550 χρειαζόμαστε το πρόγραμμα MPLABX IDE της Microchip. Αυτό το πρόγραμμα διατίθεται δωρεάν από την εταιρία και μέσω αυτού μπορούμε να προγραμματίσουμε πάρα πολλούς μικροελεγκτές της εταιρίας. Εμείς χρησιμοποιούμε την έκδοση 3.45 που είναι η τελευταία έκδοση του προγράμματος τη στιγμή που γράφεται αυτή η εργασία. Γενικώς έχουμε καταβάλει προσπάθεια όλα τα εργαλεία software που χρησιμοποιούνται να είναι τα πιο καινούρια που προσφέρονται αναφορικά με τον προγραμματισμό του PIC μας.

Εκτός από το MPLABX IDE, για να προγραμματίσουμε τον PIC μας χρειαζόμαστε έναν compiler. Εμείς επιλέξαμε να χρησιμοποιήσουμε compiler που μετατρέπει τον κώδικα C σε κώδικα assembly. Έτσι ο κώδικας που θα γράψουμε θα είναι σε C. Πρέπει να τονιστεί ότι το MPLABX IDE δεν έχει εγκατεστημένους compilers όταν το κατεβάζουμε. Πρέπει επομένως να επιλέξουμε εμείς με ποιόν compiler θα εργαστούμε και να τον κατεβάσουμε και να τον εγκαταστήσουμε στον υπολογιστή μας.

Στη σελίδα του MPLAB X IDE στο internet παρέχονται οι τελευταίοι compilers της εταιρίας. Αυτοί είναι την στιγμή που γράφεται αυτή η εργασία οι XC8 XC16 και XC32 compilers. Αυτούς θα τους βρείτε πηγαίνοντας στο <http://www.microchip.com/mplab/mplab-x-ide>. πλοηγούμαστε στο κάτω μέρος της σελίδας και πάμε στην ενότητα Download Archive. Σύρουμε πάλι προς τα κάτω και στην ενότητα language tool archives περιέχονται οι MPLAB XC8 XC16 και XC32 compilers. Εδώ επίσης θα βρείτε παλαιότερα εργαλεία προγραμματισμού της εταιρίας.

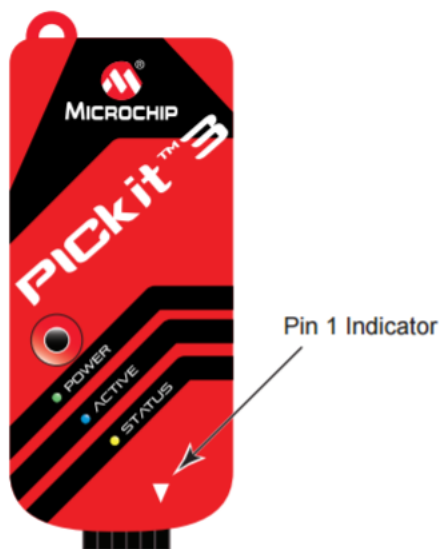
Αν και σε αυτή την εργασία προσπαθήσαμε να χρησιμοποιήσουμε τις τελευταίες εκδόσεις των εργαλείων προγραμματισμού, επομένως ο αναγνώστης θα περίμενε να χρησιμοποιήσουμε τον XC8 compiler για τον προγραμματισμό του PIC μας, αυτό δεν κατέστη δυνατό γιατί η εγκατάσταση του XC8 compiler της Microchip έχει ένα πρόβλημα. Το πρόβλημα είναι ότι μαζί με τον compiler δεν παρέχεται documentation για τις καινούριες βιβλιοθήκες του. Έτσι επιλέξαμε να χρησιμοποιήσουμε τον C18 compiler που είναι συμβατός με τον PIC18F4550 και οι βιβλιοθήκες του συνοδεύονται με λεπτομερές documentation αν και είναι παλαιότερος χρονολογικά από τον XC8. Για πληροφορίες σχετικά με τον C18 compiler μπορείτε να κάνετε google to "microchip C18 compiler". Σε αυτή την εργασία θα επεξηγήσουμε αναλυτικά μόνο τις βιβλιοθήκες του C18 που χρησιμοποιήσαμε. Περισσότερες πληροφορίες μπορούν να βρεθούν στις σχετικές ενότητες.

Τέλος πρέπει να αναφερθεί ότι αφού έχουμε γράψει τον κώδικα μας και τον έχουμε κάνει compile και build πρέπει να γίνει προγραμματισμός της συσκευής του μικροελεγκτή μας. Αυτό γίνεται μέσω κάποιου hardware που προσφέρει η εταιρία και το αναφέρουμε εδώ ως προγραμματίστρια. Εμείς χρησιμοποιούμε για προγραμματίστρια το PIC KIT 3 της microchip. Περισσότερα θα αναφερθούν στην αντίστοιχη ενότητα.

4.2. Προγραμματισμός του PIC18F4550.

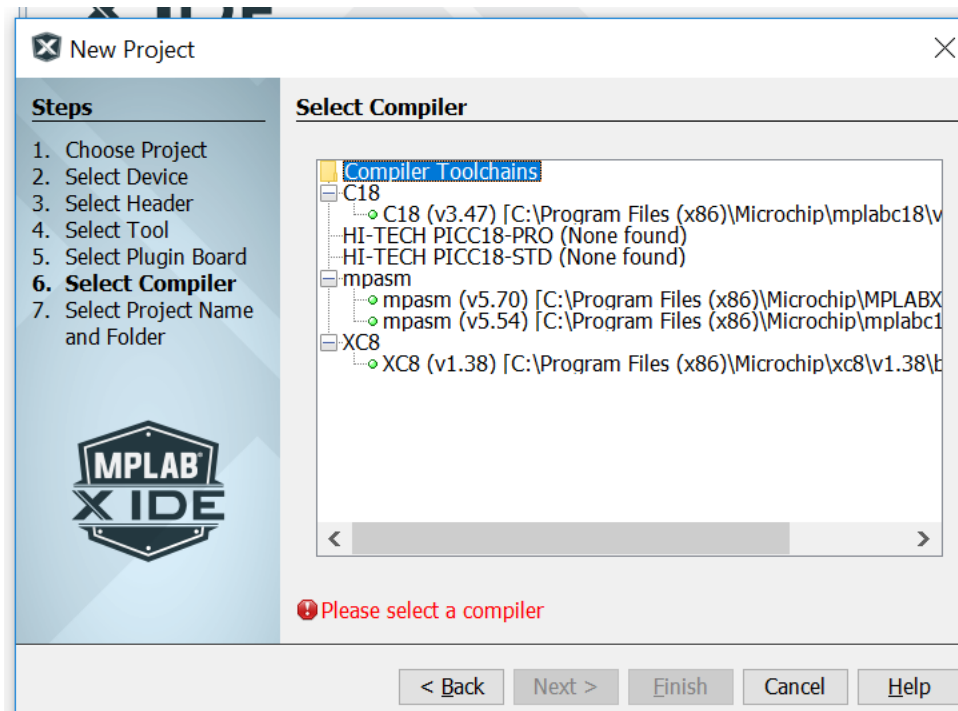
Σε αυτή την ενότητα θα αναφερθούμε στα εργαλεία προγραμματισμού του PIC18F4550. Όπως προείπαμε σε προηγούμενη ενότητα το εργαλείο μέσω του οποίου γράφουμε τον κώδικα για τον μικροελεγκτή μας είναι το MPLABX IDE (Integrated Development Environment). Επίσης χρειαζόμαστε έναν compiler, εδώ επιλέξαμε τον C18 compiler. Τέλος αφού ο compiler μας δώσει τον κώδικα σε assembly πρέπει αυτός ο κώδικας να “κατεβεί” στην συσκευή του μικροελεγκτή μας, ή με άλλα λόγια να προγραμματίσουμε τη συσκευή του μικροελεγκτή. Αυτό γίνεται με κάποιο hardware, που μπορούμε ή να το φτιάξουμε μόνοι μας ή να το αγοράσουμε από την Microchip. Υπάρχουν πολλά Hardware πακέτα που προσφέρονται για τον προγραμματισμό διαφόρων οικογενειών PIC. Εμείς επιλέξαμε να ασχοληθούμε με το PIC KIT 3. Περισσότερα θα πούμε στις παραγράφους που ακολουθούν.

Αφού εγκατασταθεί το MPLABX IDE και ο C18 compiler, συνδέουμε το PIC KIT 3 στον υπολογιστή (συνδέεται μέσω USB θήρας). Δεν χρειάζεται να εγκατασταθούν drivers για το PIC KIT 3. Μια εικόνα του PIC KIT 3 δίνεται στο παρακάτω σχήμα.



Σχήμα 4-3 Το PIC KIT 3

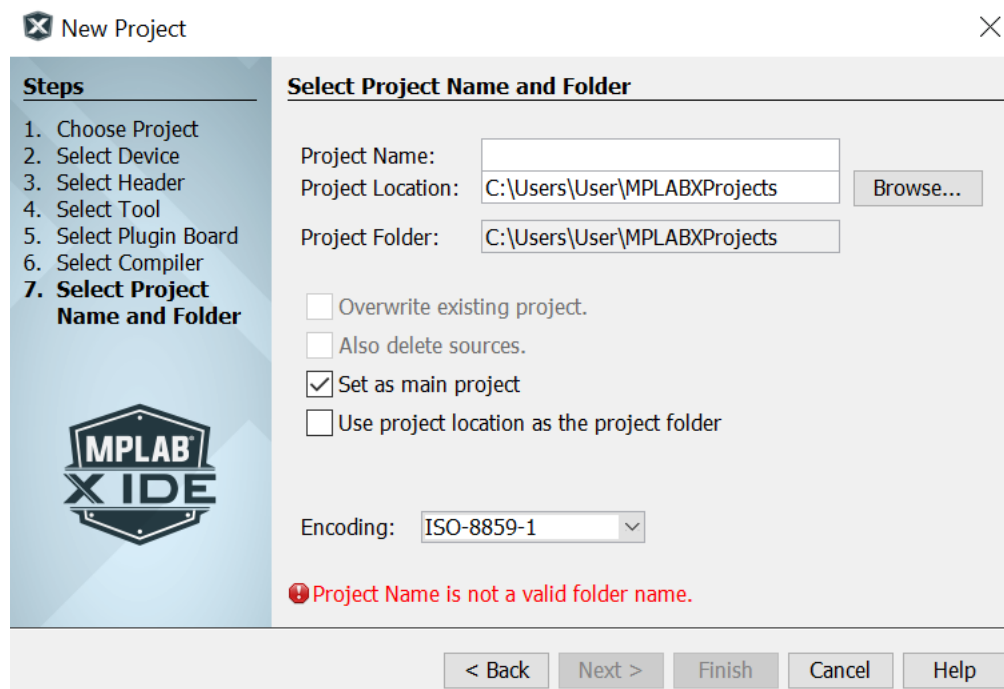
Ανοίγουμε την εφαρμογή MPLABX IDE και επιλέγουμε New Project... επιλέγουμε microchip embedded πατάμε next. Στο παράθυρο που εμφανίζεται πρέπει να επιλέξουμε τον μικροελεγκτή που θα χρησιμοποιήσουμε. Εδώ χρησιμοποιούμε τον PIC18F4550. Οπότε στο πεδίο device γράφουμε PIC18F4550. Πατώντας next το παράθυρο που εμφανίζεται ονομάζεται “select tool”. Εδώ λέμε στο MPLABX ποιο εργαλείο θα χρησιμοποιήσουμε για τον προγραμματισμό του PIC μας. Επιλέγουμε το Pickit 3. Αν το pic kit 3 έχει εγκατασταθεί σωστά και είναι συνδεδεμένο μέσω USB θα δούμε κάτω από την επιλογή Pickit3 το σειριακό αριθμό της συσκευής μας. Αφού έχουμε επιλέξει το Pickit3 πατάμε next. Θα περάσουμε στο παράθυρο που φαίνεται στο παρακάτω σχήμα.



Σχήμα 4-4 Compiler select in MPLABX IDE

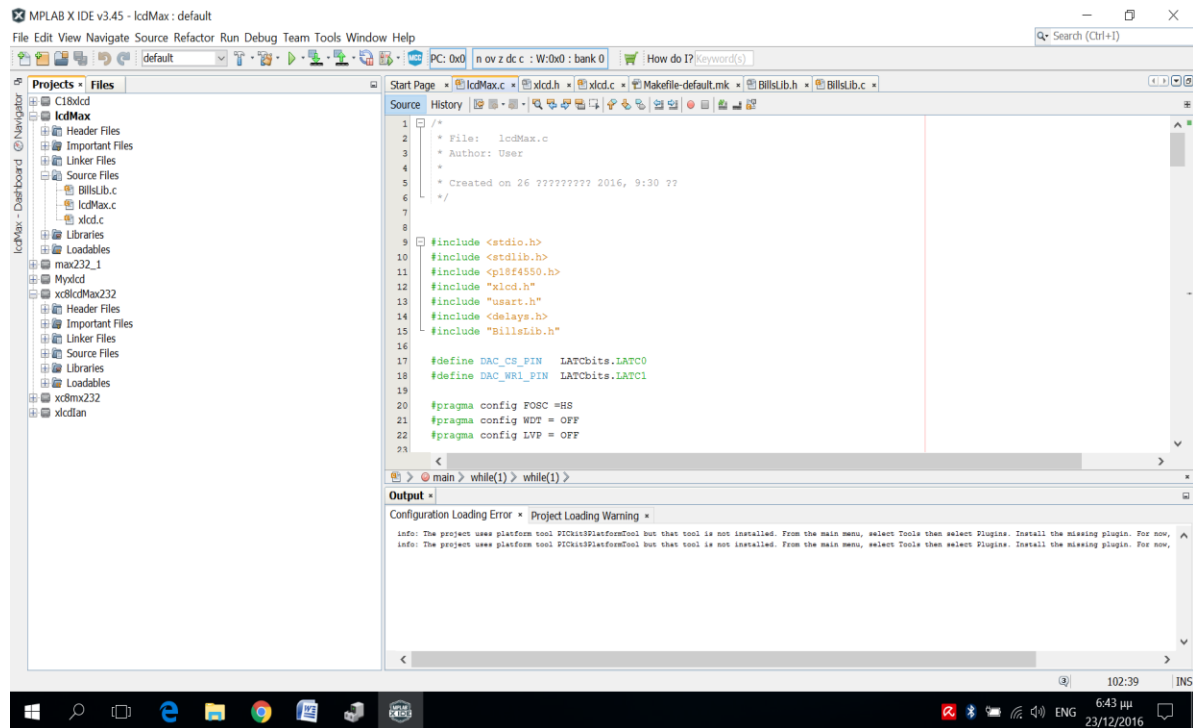
Εδώ φαίνονται οι compilers που έχουν εγκατασταθεί στον υπολογιστή μας. Στον δικό μου υπολογιστή φαίνονται ο C18 compiler και ο XC8 compiler. Επιλέγουμε τον C18 compiler και δίνουμε next.

Θα μεταφερθούμε στο παρακάτω menu.



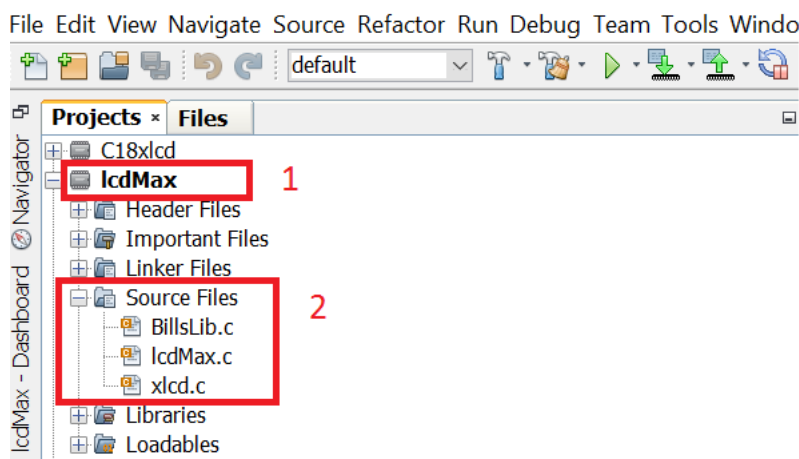
Σχήμα 4-5 Δημιουργία νέου Project στο MPLABX IDE

Εδώ πρέπει να εισαχθεί ένα όνομα για το καινούριο project. Αφού δώσουμε το όνομα πατάμε finish. Το καινούριο project έχει δημιουργηθεί και μεταφέρεστε στο περιβάλλον του MPLABX IDE, όπου έχουμε έναν κειμενογράφο στο βασικό παράθυρο μέσω του οποίου θα γράφουμε τον κώδικά μας. Μια όψη του MPLABX IDE φαίνεται στο παρακάτω σχήμα.



Σχήμα 4-6 Το περιβάλλον του MPLABX IDE

Αριστερά βλέπουμε το μενού Projects. Εδώ υπάρχουν όλα τα project που είναι ανοιχτά στο MPLABX IDE. Δεξιά ο κειμενογράφος μας δείχνει τον κώδικα του αρχείου που είναι επιλεγμένο στο αντίστοιχο tab από πάνω. Εδώ πρέπει να προσέξουμε τα εξής. Δείτε στο παρακάτω σχήμα.



Σχήμα 4-7 Όψη του μενού Projects

Το πρόγραμμα που είναι τονισμένο με Bold γράμματα στο σχήμα 4-7 είναι το Current Project (τονισμένο με κόκκινο και το γράμμα 1 στο σχήμα 4-7). Στην αρχή το Current Project

θα είναι άδειο. Κάνουμε δεξί κλικ στο εικονίδιο Source Files και επιλέξτε new -> c main file. Δίνουμε ένα όνομα σε αυτό το αρχείο, είναι το αρχείο που θα περιέχει την συνάρτηση main του προγράμματός μας. Τώρα κάνουμε διπλό κλικ στο όνομα του αρχείου και στον κειμενογράφο θα έχουμε ανοιχτό το αρχείο που μόλις δημιουργήθηκε. Παρατηρούμε ότι έχει προστεθεί κώδικας για την συνάρτηση main. Αυτό τον κώδικα φυσικά μπορούμε να τον αλλάξουμε και να προσθέσουμε ότι θέλουμε. Αφού ολοκληρωθεί την συγγραφή του κώδικα, από το μενού Run επιλέγουμε Build main project. Αυτή η ενέργεια θα γίνει στο project που είναι τονισμένο με Bold γράμματα στο μενού Projects του σχήματος 4-7 (Main project).

Πριν προχωρήσουμε παρακάτω πρέπει να αναφέρουμε ότι υπάρχουν κάποιες μεταβλητές (κάποιοι καταχωρητές) που πρέπει να προκαθοριστούν στον κώδικά μας. Καθώς θα μαθαίνουμε περισσότερα για τους μικροελεγκτές PIC θα μπορούμε να αποφασίζουμε τι τιμές πρέπει να έχουν αυτοί οι καταχωρητές. Εδώ όμως θα δώσουμε μερικές τιμές που χρησιμοποιούνται συνήθως για να μπορέσουμε να αναπτύξουμε το πρόγραμμά μας χωρίς προβλήματα. Πριν τη συνάρτηση main γράφουμε τον παρακάτω κώδικα:

```
#pragma config FOSC =HS
```

```
#pragma config WDT = OFF
```

```
#pragma config LVP = OFF
```

Αυτές είναι εντολές προς τον compiler και του λένε ότι, 1 χρησιμοποιούμε εξωτερικό κρύσταλλο, 2 ότι ο watchdog timer είναι off και 3 ότι δεν χρησιμοποιούμε Low Voltage Programming. Υπάρχουν και άλλες τέτοιες εντολές προς τον compiler, αλλά για αρχή πρέπει να ξέρουμε να χρησιμοποιούμε αυτές. Φυσικά αργότερα, έχοντας ασχοληθεί με τους μικροελεγκτές αρκετά μπορεί να αποφασίσει κάποιος ότι χρειάζεται wach dog timer ή ότι θα χρησιμοποιήσει τον εσωτερικό χρονισμό του PIC και να αλλάξει αυτές τις εντολές αντίστοιχα.

Αφού έχουμε κάνει Build, αν υπάρχουν λάθη στον κώδικα θα φανούν στο κάτω μέρος του περιβάλλοντος MPLABX IDE. Έχοντας αποσφαλμάτωσει τον κώδικα είμαστε έτοιμοι να προγραμματίσουμε την συσκευή του PIC. Για να γίνει αυτό πρέπει το PIC KIT 3 να συνδεθεί με τον PIC. Ο τρόπος σύνδεσης φαίνεται στο παρακάτω σχήμα.

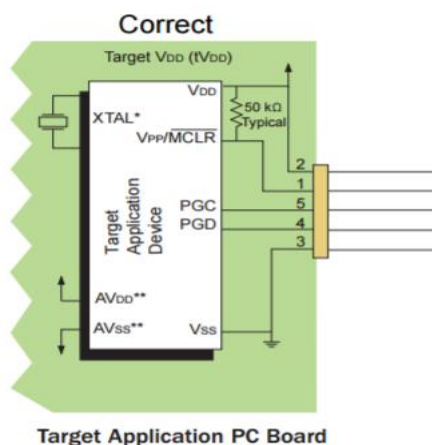
Target Connector Pinout

Pin	Signal
1	MCLR/VPP
2	VDD Target
3	VSS Ground
4	PGD (ICSPDAT)
5	PGC (ICSPCLK)
6	Do not connect*

*Reserved for future use.

Σχήμα 4-8 Αντιστοίχιση των PINS του PIC KIT 3 με αυτά του μικροελεγκτή μας.

Το PIN 1 του PIC KIT 3 (δείτε και σχήμα 4-3) εμφανίζεται με ένα άσπρο βελάκι πάνω στη συσκευή PICKIT3 και πρέπει να συνδεθεί σύμφωνα με τον πίνακα του σχήματος 4-8 στο PIN MCLR του μικροελεγκτή. Το PIN 2 του PICKIT3 πρέπει να συνδεθεί στο PIN VDD του μικροελεγκτή και ούτο καθ'εξής. Προσέχουμε επίσης ότι ο μικροελεγκτής πρέπει να τροφοδοτείται ώστε να μπορέσουμε να τον προγραμματίσουμε. Επιπλέον πρέπει να τοποθετηθεί μια αντίσταση ανάμεσα στα ποδαράκια 1 και 2 του PIC KIT 3 όπως συνδέονται στον μικροελεγκτή. Η αντίσταση μπορεί να έχει μια τυπική τιμή τα 50kOhms. Δείτε το παρακάτω σχήμα.



Σχήμα 4-9 Σωστός τρόπος σύνδεσης του PIC KIT 3.

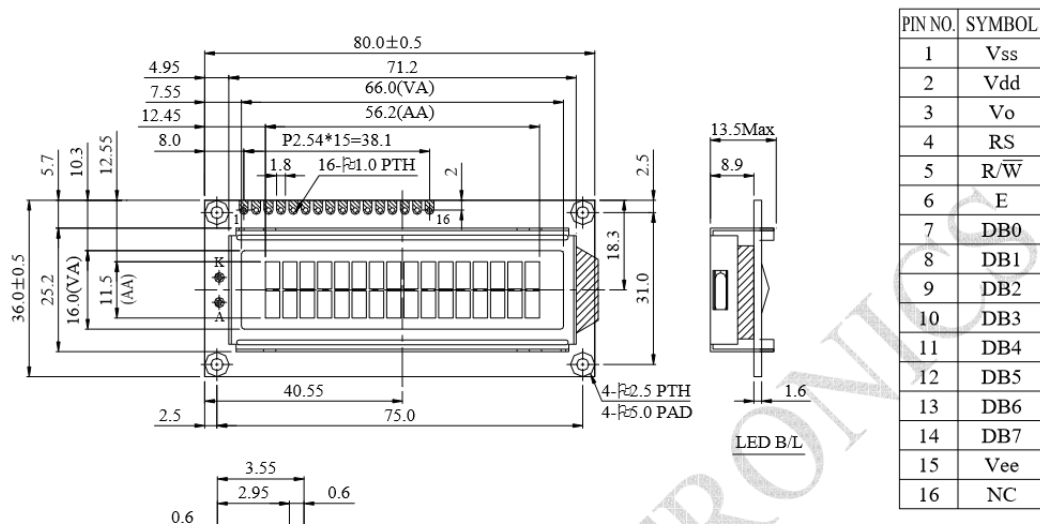
Έχοντας κάνει τη σωστή σύνδεση του μικροελεγκτή μας με το PICKIT3, στο MPLABX πατάμε το Make and Program Device Main Project, που βρίσκεται στην βασική εργαλειοθήκη στο πάνω μέρος της οθόνης του MPLABX IDE. Αν όλα έχουν γίνει σωστά το πρόγραμμα θα κατέβει στον μικροελεγκτή μας (Θα κατέβει το main Project που είναι τονισμένο με Bold στο tab Projects) και το MPLABX θα μας ενημερώσει ότι αυτό έγινε χωρίς προβλήματα. Εδώ πρέπει να σημειώσουμε ότι μπορούμε να προγραμματίσουμε με αυτό τον τρόπο μικροελεγκτές που τροφοδοτούνται με +5Volt. Αν έχουμε +3.5Volt τροφοδοσία, πρέπει να ενημερώσουμε το MPLABX πριν προγραμματίσουμε τον μικροελεγκτή, αλλιώς μπορεί να χαλάσουμε το PIC KIT 3.

Μια καλή τακτική για τους αρχάριους είναι να βάζουμε κάποια LEDs, στο κύκλωμα που κατασκευάζουμε, τα οποία θα μας ειδοποιούν ότι το πρόγραμμα τρέχει σωστά. (πχ ένα πράσινο LED μπορεί να μας ειδοποιεί ότι όλα πάνε καλά ενώ ένα κόκκινο μπορεί να μας πληροφορεί ότι υπάρχει κάποιο σφάλμα). Αφού αποκτήσουμε κάποια εμπειρία με το hardware και το software μπορούμε να βάλουμε μια LCD οθόνη στο κύκλωμά μας όπου θα εμφανίζονται μηνύματα για το αν όλα πάνε καλά ή ότι έχουμε κάποιο σφάλμα και ποιο είναι αυτό.

5. Περιγραφή Hardware και Software για επικοινωνία του PIC18F4550 με την LCD.

Σε αυτή την ενότητα θα παρουσιαστεί και θα επεξηγηθεί το hardware και το software που είναι απαραίτητα για την επικοινωνία του PIC18F4550 με την LCD οθόνη 2 σειρών 16 χαρακτήρων.

Θα ξεκινήσουμε παρουσιάζοντας το hardware, δηλαδή τα χαρακτηριστικά της LCD και τον τρόπο σύνδεσης της LCD με τον PIC. Παρακάτω φαίνονται τα PINS της LCD όπως δίνονται στα φύλλα δεδομένων της.



Σχήμα 5-1 Η LCD οθόνη 2 σειρών 16 χαρακτήρων της Raystar

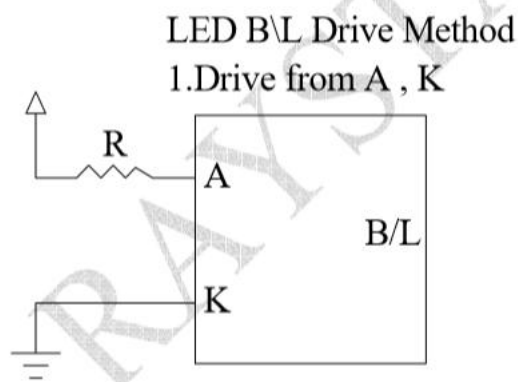
Ας δούμε αναλυτικά ποια είναι η λειτουργία κάθε ενός από τα PINS.

Πίνακας 5-1 Τα PINS της LCD Raystar και η λειτουργία τους.

PIN	Σύμβολο	Επίπεδο	Περιγραφή
1	Vss	0Volt	Ground (Γείωση)
2	V _{DD}	3.3Volt	Τάση τροφοδοσίας (μπορεί να δουλέψει και στα +5Volt)
3	Vo	Μεταβλητό	Η τάση λειτουργίας της LCD στο κύκλωμά μας 0.8Volt
4	RS	Υψηλό/Χαμηλό	Υψηλό Δεδομένα Χαμηλό Οδηγίες
5	RW	Υψηλό / Χαμηλό	Υψηλό Διάβασε Χαμηλό Γράψε
6	E	Υψηλό→Χαμηλό	Σήμα ενεργοποίησης του Chip.
7	DB0	Υψηλό / Χαμηλό	Γραμμή διαύλου δεδομένων

8	DB1	Υψηλό / Χαμηλό	Γραμμή διαύλου δεδομένων
9	DB2	Υψηλό / Χαμηλό	Γραμμή διαύλου δεδομένων
10	DB3	Υψηλό / Χαμηλό	Γραμμή διαύλου δεδομένων
11	DB4	Υψηλό / Χαμηλό	Γραμμή διαύλου δεδομένων
12	DB5	Υψηλό / Χαμηλό	Γραμμή διαύλου δεδομένων
13	DB6	Υψηλό / Χαμηλό	Γραμμή διαύλου δεδομένων
14	DB7	Υψηλό / Χαμηλό	Γραμμή διαύλου δεδομένων
15	Vee	-	Έξοδος αρνητικής τάσης
16	NC	-	Δεν συνδέεται.

Όπως φαίνεται στο σχήμα 5-1 στα αριστερά υπάρχουν δύο υποδοχές οι A και K. Με αυτά τα PINS μπορούμε να τροφοδοτήσουμε το Backlight (φωτισμό) της LCD. Αυτή η επιλογή υπάρχει στην οθόνη της Raystar, αλλά μπορεί να μην υπάρχει σε μοντέλα άλλων κατασκευαστών ή μπορεί τα PINS να είναι διαφορετικά. Δείτε παρακάτω στο σχήμα 5-2 το κύκλωμα για το backlight της LCD.

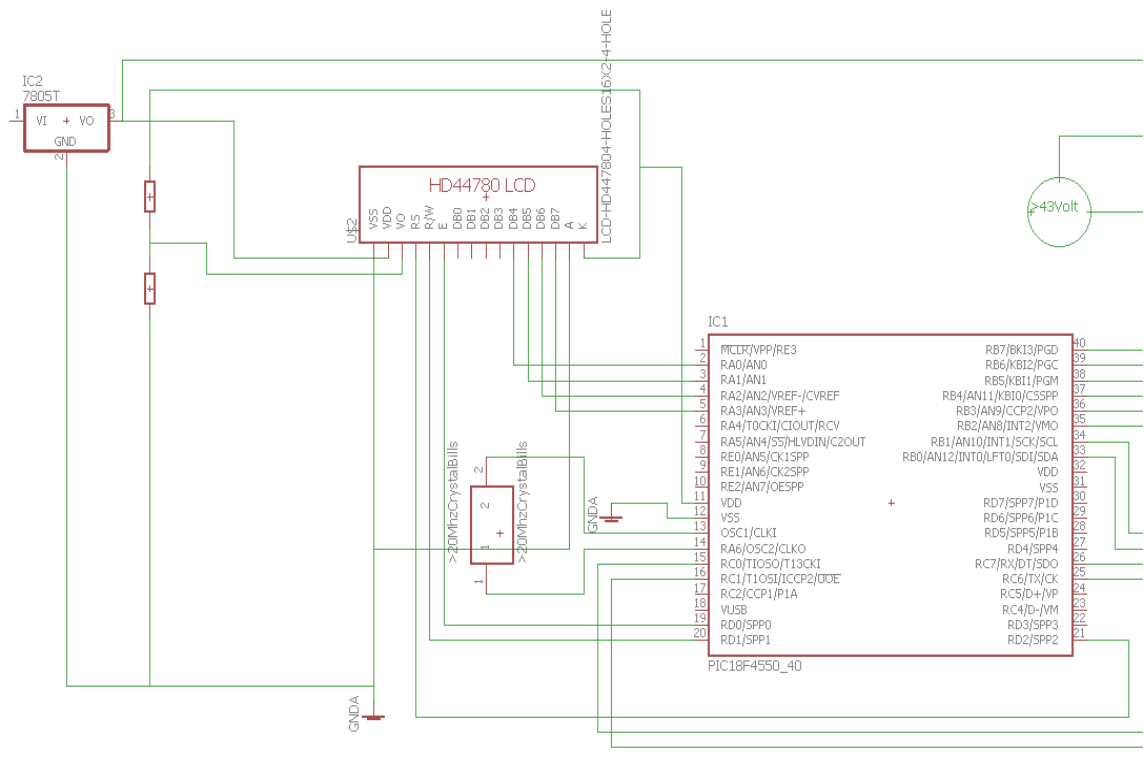


Σχήμα 5-2 Το κύκλωμα για τον φωτισμό της LCD της Raystar.

Τα φύλλα δεδομένων μας πληροφορούν ότι το ρεύμα για τον φωτισμό της LCD πρέπει να είναι 32 ως 40mA και η τάση τροφοδοσίας 3.5 Volt. Οπότε πρέπει να επιλέξουμε κατάλληλη αντίσταση για να πετύχουμε αυτό το ρεύμα.

Τα PINS όμως 1-16, έχουν σε όλους τους κατασκευαστές που είναι συμβατοί με την Hitachi HD44780, την ίδια λειτουργία. Άρα για να λειτουργήσει το κύκλωμα που υλοποιούμε παρακάτω χρειαζόμαστε μια οποιαδήποτε LCD 2 σειρών 16 χαρακτήρων αρκεί να είναι συμβατή με την Hitachi HD44780 (ουσιαστικά να έχει τον ίδιο controller).

Στο παρακάτω σχήμα 5-3 φαίνεται η σύνδεση της LCD οθόνης με τον PIC.



Σχήμα 5-3 Η σύνδεση της LCD 2 σειρών 16 χαρακτήρων με τον PIC48F4550.

Το ολοκληρωμένο LM7805 που φαίνεται πάνω αριστερά στο σχήμα 5-3 συνδέεται με μια μπαταρία 9Volt. Αυτό το ολοκληρωμένο όπως αναφέρθηκε και στην εισαγωγή δίνει στην έξοδό του 5Volt. Τα 5Volt συνδέονται στο PIN 2 (V_{DD}). Το PIN 1 συνδέεται στην γη. Το PIN 3 είναι η τάση λειτουργίας της LCD και σε αυτό το κύκλωμα την έχουμε στα 0.8Volt με ένα διαιρέτη τάσης από τα +5Volt. Τα PINs RS RW και E είναι σήματα ελέγχου της LCD. Αυτά συνδέονται στα PINS 21 20 και 19 του PIC αντίστοιχα. Έχει συνδεθεί επίσης στο κύκλωμά μας και ο κρύσταλλος 20MHz που είναι απαραίτητος για τον χρονισμό του PIC.

Αφού έχουμε ολοκληρώσει το Hardware πρέπει τώρα να γράψουμε το software για την επικοινωνία με την LCD. Αν ψάξουμε στο internet θα βρούμε πολλά tutorials που δίνουν software για τον έλεγχο μιας LCD οθόνης από μικροελεγκτές PIC. Σε αυτά υπάρχουν ρουτίνες φτιαγμένες από τους εκάστοτε χρήστες που χρησιμοποιούν τα σήματα ελέγχου και ρουτίνες που στέλνουν δεδομένα στην LCD. Σε αυτή την εργασία έγινε προσπάθεια να χρησιμοποιηθούν, όπου είναι δυνατό, βιβλιοθήκες της microchip, ώστε ο κώδικάς μας να είναι σωστά δομημένος, εύχρηστος, εύκολος στην κατανόηση και επαναχρησιμοποιήσιμος. Επίσης υπάρχει documentation από την microchip που περιγράφει λεπτομερώς την χρήση των βιβλιοθηκών, έτσι ένας νέος χρήστης θα μπορέσει εύκολα να κατανοήσει τον κώδικά αν χρησιμοποιηθούν αυτές οι βιβλιοθήκες. Αν διαθέτουμε τον XC8 compiler της microchip θα διαπιστώσουμε ότι στην τρέχουσα έκδοση δεν υπάρχει documentation για τις βιβλιοθήκες του. Ίσως αυτό διορθωθεί σε επόμενη έκδοση όμως αυτός είναι ο λόγος που επιλέχθηκε να χρησιμοποιηθεί ο παλαιότερος C18 compiler. Πάντα πρέπει να ελέγχεται αν ο PIC που θα χρησιμοποιηθεί στο project υποστηρίζεται από τον compiler που θέλουμε να χρησιμοποιήσουμε, ή αν θέλουμε να χρησιμοποιήσουμε ένα συγκεκριμένο compiler πρέπει

να ελεγχθεί ποιους μικροελεγκτές μπορεί να υποστηρίξει, αυτό αναφέρεται στην ενότητα *supported devices* (υποστηριζόμενες συσκευές) των εγγράφων του *compiler*. Εδώ βασικό κριτήριο για την επιλογή του PIC ήταν ο αριθμός των PINs που χρειάζονται για την επικοινωνία με όλα τα περιφερικά, έτσι επιλέχθηκε ο PIC18F4550 γιατί έχει 40 PINs και καλύπτει τις ανάγκες μας. Ένα ακόμη κριτήριο ήταν η ύπαρξη σειριακής επικοινωνίας υλοποιημένη σε *hardware* στον PIC. Αυτό σημαίνει ότι δεν θα χρειαστεί να γράψουμε *software* για την υλοποίηση της σειριακής επικοινωνίας, απλώς θα χρησιμοποιήσουμε τις βιβλιοθήκες της *microchip* για σειριακή επικοινωνία.

Όπως προείπαμε έγινε προσπάθεια να χρησιμοποιηθούν βιβλιοθήκες της *microchip* για τον έλεγχο των περιφερειακών. Στο *documentation* για τον C18 *compiler* μπορούν να βρεθούν βιβλιοθήκες για επικοινωνία με οθόνη LCD 2 σειρών 16 χαρακτήρων. Παρακάτω παραθέτουμε τις συναρτήσεις ελέγχου της LCD που προτείνει η *microchip*. Αυτές μπορούν να βρεθούν στο *internet* στο έγγραφο «MPLAB C18 COMPILER LIBRARIES». (Η LCD που θα χρησιμοποιηθεί πρέπει να έχει *controller* συμβατό με αυτόν της Hitachi HD44780).

Πίνακας 5-2 Προτεινόμενες συναρτήσεις ελέγχου LCD της *microchip*

Συνάρτηση	Περιγραφή
BusyXLCD	Επιστρέφει λογικό 1 αν η οθόνη είναι απασχολημένη.
OpenXLCD	Συνάρτηση αρχικοποίησης της LCD.
putcXLCD	Γράφει ένα <i>byte</i> στον <i>controller</i> της LCD.
putsXLCD	Στέλνει ένα <i>string</i> από τη μνήμη δεδομένων στην LCD.
putrsXLCD	Στέλνει ένα <i>string</i> από τη μνήμη προγράμματος στην LCD.
ReadAddrXLCD	Διαβάζει ένα <i>byte</i> διεύθυνσης από τον ελεγκτή της LCD.
ReadDataXLCD	Διαβάζει ένα <i>byte</i> από τον ελεγκτή της LCD.
SetCGRamAddr	Καθορίζει την διεύθυνση του δημιουργού χαρακτήρων.
SetDDRamAddr	Καθορίζει την διεύθυνση προβολής δεδομένων στην LCD.
WriteCmdXLCD	Γράφει μια εντολή στον ελεγκτή της LCD.
WriteDataXLCD	Γράφει ένα <i>byte</i> στην LCD.

Αυτή είναι η παλέτα συναρτήσεων που προσφέρει η *microchip* για τον έλεγχο της LCD. Για να μπορέσουμε να χρησιμοποιήσουμε τις συναρτήσεις αυτές πρέπει να κάνουμε μερικές αρχικές ρυθμίσεις μέσω *software*. Αυτές οι συναρτήσεις ορίζονται στο αρχείο *lcd.h*. Σε αυτό το αρχείο υπάρχουν οι αρχικοποιήσεις των PINs ελέγχου και των PINs δεδομένων, δηλαδή δηλώνεται ποια PINs του μικροελεγκτή αντιστοιχούν σε ποια PINs της LCD. Επίσης ορίζονται κάποιες συναρτήσεις καθυστέρησης που χρειάζονται για την επικοινωνία με την LCD. Ο σχετικός κώδικας από το αρχείο επικεφαλίδας *lcd.h* παρατίθεται ολόκληρος παρακάτω, όπως έχει αλλαχθεί για να ταιριάζει με το δικό μας κύκλωμα, στα σχόλια περιγράφεται η λειτουργία του κώδικα και οι αλλαγές που έχουν γίνει. Είναι σημαντικό να κατανοηθεί και να τροποποιηθεί σωστά ο παρακάτω κώδικας αν θέλουμε να

χρησιμοποιήσουμε τις προτεινόμενες συναρτήσεις της microchip. Εδώ πρέπει να αναφερθεί ότι το αρχείο επικεφαλίδας xlcd.h που παρέχει η microchip και τα αρχεία .c που υλοποιούν τις συναρτήσεις που περιγράφονται από το .h αρχείο **δεν είναι σωστές** !. Υπάρχουν λάθη στα αρχεία αυτά, και δεν γίνονται compile !. Δυστυχώς η microchip έχει λάθη στον κώδικα της. Έγινε μεγάλη προσπάθεια σε αυτή την εργασία να χρησιμοποιηθούν οι προτεινόμενες βιβλιοθήκες, διότι έχουν documentation και ο κώδικάς τους είναι σωστά δομημένος. Έτσι οι συναρτήσεις που υπάρχουν στα αρχεία .h και .c ξαναγράφηκαν. Χρησιμοποιώντας τα αρχεία αυτά μπορούμε να υλοποιήσουμε τις βιβλιοθήκες και να τις χρησιμοποιήσουμε ακριβώς όπως περιγράφεται στο documentation της microchip. Τα αρχεία αυτά , είναι 2, το xlcd.h και το xlcd.c και μπορούν να βρεθούν στα περιεχόμενα αυτής της εργασίας (CD). Παρακάτω φαίνεται το αρχείο xlcd.h όπως το έχει τροποποιηθεί για να λειτουργεί σωστά. Σε αυτό το αρχείο πρέπει να κάνουμε επιπλέον αλλαγές που χρειάζονται για να το κάνουμε συμβατό με το κύκλωμά μας. Η επεξήγηση του κώδικα είναι σε σχόλια για να είναι δυνατό το drag and drop.

```
#ifndef __XLCD_H //Κώδικας για την χρήση της βιβλιοθήκης, αποτρέπει η βιβλιοθήκη
//να φορτωθεί περισσότερες από μία φορές.
```

```
#define __XLCD_H
```

```
/* η μεταβλητή DATA_PORT παρακάτω καθορίζει ποια PORT του PIC θα χρησιμοποιηθεί
σαν δίαυλος δεδομένων, δηλαδή από πού θα στέλνονται τα δεδομένα στα PINs δεδομένων
της LCD. Η μεταβλητή TRIS πρέπει να ορισθεί επίσης. Αυτή καθορίζει αν τα PINs του PIC θα
είναι είσοδοι ή έξοδοι. Στην εφαρμογή μας χρησιμοποιούμε την θύρα A του PIC για σαν
δίαυλο δεδομένων, έτσι τροποποιήσαμε τον παρακάτω κώδικα αντιστοίχως.*/
```

```
#define DATA_PORT PORTA
```

```
#define TRIS_DATA_PORT TRISA
```

```
/* τα παρακάτω #define καθορίζουν ποια θα είναι τα PINs ελέγχου, δηλαδή τα PINs του PIC
που θα στέλνουν τα σήματα ελέγχου στα αντίστοιχα PINs της LCD.*/
```

```
#define RW_PIN LATDbits.LATD1 /* για το σήμα ελέγχου RW χρησιμοποιούμε το PIN RD1
της θύρας D του PIC.*/
```

```
#define TRIS_RW TRISDbits.TRISD1 /* πρέπει επίσης να ορισθεί και το TRISD1 για το σήμα
RW της LCD. (Για είναι διαθέσιμη σε άλλες συναρτήσεις της microchip η διεύθυνση του
καταχωρητή TRIS του αντίστοιχου PIN του PIC μας.) */
```

```
#define RS_PIN LATDbits.LATD2 /* για το σήμα ελέγχου RS χρησιμοποιούμε το PIN RD2
της θύρας D του PIC.*/
```

```
#define TRIS_RS TRISDbits.TRISD2 /* Εδώ ορίζεται και το αντίστοιχο TRIS δείτε και
παραπάνω για επεξήγηση.*/
```

```
#define E_PIN LATDbits.LATD0 /* για το σήμα ελέγχου E χρησιμοποιούμε το PIN RD0 της θύρας D του PIC.*/
```

```
#define TRIS_E TRISDbits.TRISD0 /* Ορισμός του αντίστοιχου καταχωρητή TRIS για το E */
```

Τα παρακάτω #define ορίζουν μεταβλητές για τον έλεγχο λειτουργιών της LCD όπως το αν θα απεικονίζει κάτι ή όχι, αν θα φαίνεται ο κέρσορας, εντολές μετατόπισης του κέρσορα και λοιπά. Αυτές οι εντολές έχουν κάποιες συγκεκριμένες προκαθορισμένες τιμές στον ελεγκτή της LCD που χρησιμοποιούμε. Για αυτές τις τιμές πρέπει να ενημερωθεί το αρχείο επικεφαλίδας της microchip. Για να μπορέσουμε να τις καθορίσουμε πρέπει να αναφερθούμε στο αντίστοιχο εδάφιο στα φύλλα δεδομένων της LCD. Οι εντολές που μας ενδιαφέρουν παρατίθενται παρακάτω για την Raystar 2 σειρών 16 χαρακτήρων.

Πίνακας 5-3 Τιμές του control byte για τον έλεγχο μερικών λειτουργιών απεικόνισης της Raystar.

Εντολή	Κώδικας εντολής στον ελεγκτή της LCD της Raystar										Περιγραφή
	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0	
Καθαρισμός απεικόνισης	0	0	0	0	0	0	0	0	0	1	
Επιστροφή στην αρχή	0	0	0	0	0	0	0	0	1	-	
Display, Cursor, Blink (cursor position) On / Off	0	0	0	0	0	0	1	D	C	B	D: on / off απεικόνιση C: on/off κέρσορας B: on/off ένδειξη θέσης κέρσορα
Μετατόπιση κέρσορα ή απεικόνισης	0	0	0	0	0	1	S/C	R/L	-	-	Ελέγχει τη μετατόπιση του κέρσορα και της απεικόνισης χωρίς να αλλάζει τα δεδομένα της DDRAM.

Αφού συμβουλευτούμε τον παραπάνω πίνακα πρέπει να δώσουμε τις σωστές τιμές στα παρακάτω #define.

```
/*#define για Έλεγχο απεικόνισης, έλεγχο κέρσορα και έλεγχο θέσης κέρσορα. */
```

```
#define DON 0b00001111 /* Display on */
```

```
#define DOFF 0b00001011 /* Display off */
```

```

#define CURSOR_ON 0b00001111 /* Cursor on */
#define CURSOR_OFF 0b00001101 /* Cursor off */
#define BLINK_ON 0b00001111 /* Cursor Blink */
#define BLINK_OFF 0b00001110 /* Cursor No Blink */

/* #define για το έλεγχο μετατόπισης του κέρσορα (Cursor or Display Shift) */
#define SHIFT_CUR_LEFT 0b00010011 /* Ο κέρσορας μετατοπίζεται αριστερά */
#define SHIFT_CUR_RIGHT 0b00010111 /* Ο κέρσορας μετατοπίζεται δεξιά */
#define SHIFT_DISP_LEFT 0b00011011 /* Η απεικόνιση μετατοπίζεται αριστερά (τα περιεχόμενα της απεικόνισης δεν αλλάζουν.) */
#define SHIFT_DISP_RIGHT 0b00011111 /* Η απεικόνιση μετατοπίζεται δεξιά */

```

Έπειτα το αρχείο επικεφαλίδας της microchip, (όπως έχει τροποποιηθεί για να λειτουργεί lcd.h), απαιτεί να ορίσουμε στα παρακάτω #define μεταβλητές που αντιστοιχούν σε αντίστοιχες εντολές 8bit του ελεγκτή της LCD για τον καθορισμό ορισμένων λειτουργιών αρχικοποίησης της οθόνης μας. Ανατρέχοντας στα φύλλα δεδομένων της Raystar βρίσκουμε τον πίνακα με τις παρακάτω εντολές.

Πίνακας 5-4 Τιμές του control byte για την αρχικοποίηση της LCD της Raystar.

Εντολή	Κώδικας εντολής στον ελεγκτή της LCD της Raystar										Περιγραφή
	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0	
Καθορισμός λειτουργίας	0	0	0	0	1	DL	N	F	-	-	DL: διεπαφή δεδομένων 8/4 bit N: αριθμός γραμμών =2/1 F: Μέγεθος γραμματοσειράς 5x11/5x8

```

/* Σύμφωνα με τον παραπάνω πίνακα καθορίζουμε τις παρακάτω μεταβλητές */
#define FOUR_BIT 0b00101111 /* 4-bit Interface */
#define EIGHT_BIT 0b00111111 /* 8-bit Interface */

#define LINE_5X7 0b00110011 /* 5x7 characters, single line (σε εμάς είναι 5x8 και αναφερόμαστε στα dots που διαθέτει η LCD για την απεικόνιση ενός χαρακτήρα). */
#define LINE_5X10 0b00110111 /* 5x10 characters */
#define LINES_5X7 0b00111011 /* 5x7 characters, multiple line */

```



```
#define PARAM_SCLASS auto /*Αυτή την μεταβλητή αφήσετε ως έχει, δεν θα την χρησιμοποιήσουμε.*/
```

```
#define MEM_MODEL far /* Αν αυτή τη μεταβλητή αλλαχτεί σε near τότε αναφερόμαστε σε μικρό μοντέλο μνήμης αν δεν είναι γνωστό το μοντέλο μνήμης που χρησιμοποιείται μπορεί να αγνοηθεί αυτή η σειρά κώδικα.*/
```

Εδώ τελειώνει ο καθορισμός των μεταβλητών #define που απαιτεί το αρχείο επικεφαλίδας xlcd.h. Η δουλειά μας με το αρχείο αυτό δεν έχει τελειώσει ακόμη. Παρακάτω φαίνονται οι συναρτήσεις που ορίζονται στο αρχείο επικεφαλίδας, αυτές είναι οι συναρτήσεις που επεξηγήθηκαν παραπάνω σε αυτή την ενότητα, και περιγράφονται στο documentation του C18 compiler. Παρακάτω φαίνονται τα πρότυπά τους στο αρχείο xlcd.h

```
void OpenXLCD(PARAM_SCLASS unsigned char);
```

```
void SetCGRamAddr(PARAM_SCLASS unsigned char);
```

```
void SetDDRamAddr(PARAM_SCLASS unsigned char);
```

```
unsigned char BusyXLCD(void);
```

```
unsigned char ReadAddrXLCD(void);
```

```
char ReadDataXLCD(void);
```

```
void WriteCmdXLCD(PARAM_SCLASS unsigned char);
```

```
void WriteDataXLCD(PARAM_SCLASS char);
```

```
#define putcXLCD WriteDataXLCD
```

```
void putsXLCD(PARAM_SCLASS char *);
```

```
void putrsXLCD(const rom char *);
```

Μετά τον ορισμό των παραπάνω προτύπων των συναρτήσεων, το αρχείο xlcd.h της microchip ορίζει τρεις συναρτήσεις καθυστέρησης. Αυτές οι καθυστερήσεις είναι απαραίτητες για την επικοινωνία του PIC με την LCD, γιατί κάθε εντολή που στέλνουμε στην οθόνη απαιτεί κάποιο χρονικό διάστημα για να εκτελεστεί από την οθόνη. Ορίζοντας τα πρότυπά τους η microchip μας συμβουλεύει πόση χρονική καθυστέρηση πρέπει να εισάγουμε σε κάθε συνάρτηση καθυστέρησης. Παρακάτω φαίνονται τα πρότυπα των συναρτήσεων αυτών (είναι τρεις συναρτήσεις).

```
extern void DelayFor18TCY(void);
```

```
extern void DelayPORXLCD(void);
```

```
extern void DelayXLCD(void);
```

Μετά τον ορισμό αυτών των συναρτήσεων το αρχείο xlcd.h τελειώνει. Για την συνάρτηση DelayFor18TCY(void) πρέπει ο χρήστης να υλοποιήσει μια καθυστέρηση 18 κύκλων εντολής,

όπως προτρέπει και το όνομά της. Ο κώδικας που γράφτηκε για την υλοποίηση αυτής της συνάρτησης φαίνεται παρακάτω.

```
void DelayFor18TCY(void)
{
    Delay10TCYx(10); //10 κύκλοι εντολής.
    Delay1TCY(); //1 ένας κύκλος εντολής.
    Delay1TCY();
    Delay1TCY();
    Delay1TCY();
    Delay1TCY();
    Delay1TCY();
    Delay1TCY();
    Delay1TCY();
    Delay1TCY(); //18 κύκλοι εντολής συνολικά.
}
```

Εδώ πρέπει να αναφέρουμε ότι για την χρήση της συνάρτησης Delay10TCY() και της Delay1TCY() πρέπει να συμπεριλάβουμε στον κώδικά μας το αρχείο επικεφαλίδας delays.h με την εντολή #include <delays.h>.

Η επόμενη συνάρτηση ορίζεται ως void DelayPORXLCD(void) και η microchip μας λέει ότι πρέπει να υλοποιεί μια καθυστέρηση 15msec. Για να την υλοποιήσουμε πρέπει να κάνουμε κάποιους υπολογισμούς. Ο μικροελεγκτής χρονίζεται με κρύσταλλο συχνότητας 20MHz. Άρα η περίοδος είναι:

$$T = \frac{1}{20} * 10^{-6} sec = 0.05 \mu sec$$

Άρα ο κύκλος ρολογιού είναι 0.05μsec. Ο κύκλος εντολής στον μικροελεγκτή είναι 4*(κύκλος ρολογιού) = 0.2μsec. Η εντολή Delay1kKTCY() δέχεται παραμέτρους από 1 έως 255. Και προκαλεί καθυστέρηση 1000 (εξάγεται από το 1k στο όνομα της εντολής) * x κύκλους εντολής όπου x από 1 ως 255. Έτσι αν δώσουμε x=75 θα έχουμε καθυστέρηση 1000*75*0.2μsec = 15000μsec=15msec. Που είναι όση καθυστέρηση προτείνει η microchip να εισάγουμε στον κώδικά μας. Όμως γενικώς οι καθυστερήσεις αλλάζουν ελαφρώς ανάλογα με την συγκεκριμένη συσκευή που χρησιμοποιούμε, έτσι σε εμάς η καθυστέρηση 15msec δεν είναι αρκετή για την LCD που χρησιμοποιούμε. Αυτό το καταλαβαίνουμε γιατί αφού έχουμε ορίσει τις καθυστερήσεις στις ακριβείς τιμές που προτείνονται, παίρνουμε στην LCD τυχαίους χαρακτήρες. Μεγαλώνοντας ελάχιστα τις καθυστερήσεις από τις προτεινόμενες τιμές παίρνουμε το απαιτούμενο αποτέλεσμα. Οπότε είναι καλή πρακτική, να δίνονται στον κώδικα που γράφουμε μεγαλύτερες τιμές καθυστέρησης από τις

προτεινόμενες και αφού δουλέψει σωστά το κύκλωμα μπορούμε να τις μειώσουμε σταδιακά μέχρι να βρεθούν οι ελάχιστες τιμές που δουλεύουν στο κύκλωμά μας (αν τίθεται θέμα χρόνου).

Η υλοποίηση της συνάρτησης void DelayPORXLCD(void) στον δικό μου κώδικα φαίνεται παρακάτω. Παρατηρήστε ότι δώθηκε παράμετρος 90 στη συνάρτηση Delay1KTCYx() και έτσι η καθυστέρηση που εισάγουμε είναι 18msec. Για πληροφορίες σχετικά με τις συναρτήσεις καθυστέρησης της microchip στον C18 compiler μπορεί να αναφερθεί κανείς στο documentation του compiler. Το TCY στο όνομα της εντολής σημαίνει κύκλος εντολής.

```
void DelayPORXLCD(void)
```

```
{  
  
    Delay1KTCYx(90);  
  
    // χρειάζεται καθυστέρηση 15msec, σε εμάς είναι 18msec.  
  
}
```

Η τελευταία συνάρτηση καθυστέρησης που πρέπει να υλοποιηθεί είναι η DelayXLCD(). Σε αυτή την συνάρτηση πρέπει να φτιάξουμε μια καθυστέρηση 5msec. Πάλι στον κώδικά μας υλοποιούμε ελάχιστα μεγαλύτερη καθυστέρηση. Ο τελικός κώδικας της DelayXLCD() φαίνεται παρακάτω. Η καθυστέρηση που εισάγεται είναι $1000*40*0.2\text{msec} = 8\text{msec}$.

```
void DelayXLCD(void)
```

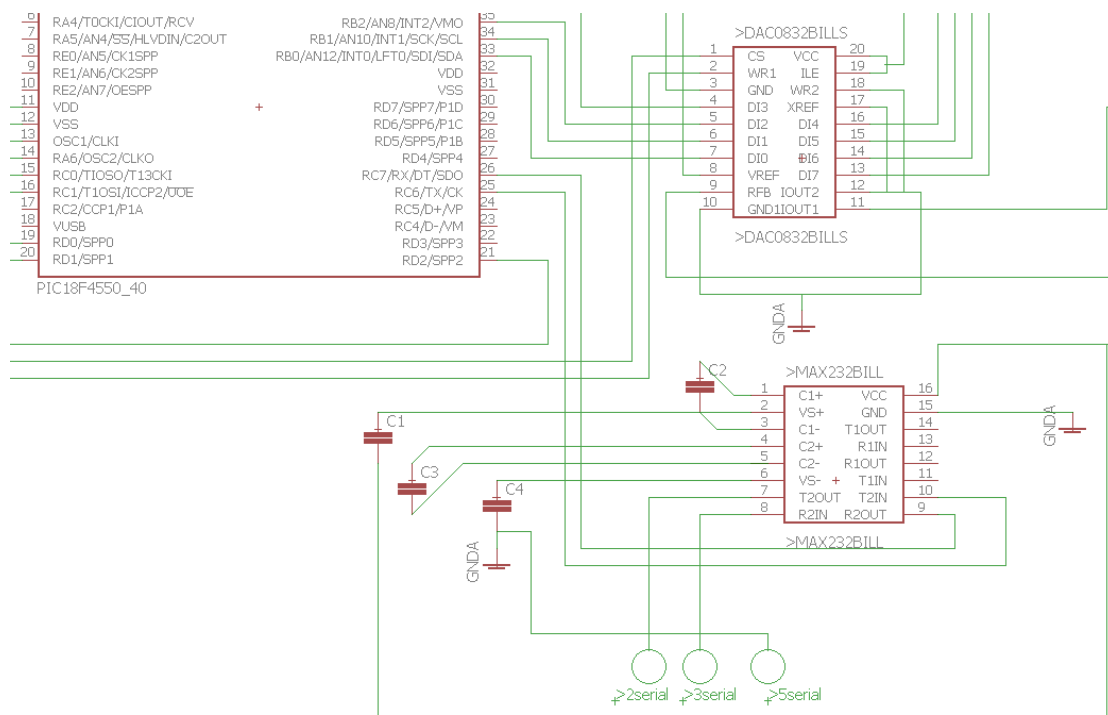
```
{  
  
    Delay1KTCYx(40);  
  
    //χρειάζεται καθυστέρηση 5 msec, σε εμάς είναι 8msec;  
  
}
```

Εδώ τελειώνει η δουλειά που πρέπει να γίνει για να λειτουργούν οι συναρτήσεις της microchip στο κύκλωμά μας. Αν αυτές οι αλλαγές που προτάθηκαν εδώ γίνουν στις αρχικές συναρτήσεις της microchip ο κώδικας δεν θα λειτουργήσει. Οι συναρτήσεις που περιγράφηκαν εδώ είναι φτιαγμένες από τον γράφων, με βοήθεια από το internet, και έχουν διορθωθεί. Οι διορθωμένες συναρτήσεις βρίσκονται στο CD αυτής της εργασίας. Μπορούμε επομένως προσαρμόζοντας όπως περιγράφηκε σε αυτή την ενότητα τις παραπάνω συναρτήσεις, να χρησιμοποιήσουμε πλέον τις «έτοιμες» συναρτήσεις της microchip για τον έλεγχο της LCD της Raystar. Θα φανεί στο αντίστοιχο εδάφιο ότι ο κώδικάς μας έχει απλοποιηθεί πάρα πολύ, σε σχέση με το αν φτιάχναμε δικές μας συναρτήσεις για τον έλεγχο της οθόνης. Επιπλέον τώρα το documentation του compiler περιγράφει αναλυτικά τη λειτουργία κάθε συνάρτησης και ένας νέος χρήστης θα μπορέσει εύκολα να καταλάβει τη χρήση τους. Ο κώδικας επικοινωνίας με την LCD θα παρατεθεί στο εδάφιο που ασχολείται με τον τελικό κώδικα της εργασίας, όπου θα εξηγηθεί η επικοινωνία όλων των περιφερειακών και τη τελική λειτουργία του κυκλώματος.

6. Περιγραφή Hardware και Software για την επικοινωνία του PIC με το MAX232 και τον Η/Υ(σειριακή επικοινωνία).

6.1.Περιγραφή επικοινωνίας του PIC με το MAX232 και τη σειριακή θύρα.

Σε αυτή την ενότητα θα περιγραφεί το hardware και το software που χρειάζεται για την επικοινωνία μέσω σειριακής θύρας με ένα PC. Οι καινούριοι υπολογιστές δεν έχουν σειριακή θύρα, έτσι πρέπει να διαθέτουμε ένα καλώδιο USB to serial για να υλοποιήσουμε την επικοινωνία. Το καλώδιο αυτό συνοδεύεται από CD με drivers και πρέπει να τους εγκαταστήσουμε αν θέλουμε να το χρησιμοποιήσουμε. Από τη πλευρά του μικροελεγκτή για την επικοινωνία με τη σειριακή θύρα χρειαζόμαστε το ολοκληρωμένο MAX232. Ο PIC μας συνδέεται με το MAX232 με τα PINS RX για λήψη (Recive) και TX για αποστολή (Transmit) δεδομένων. Το MAX232 συνδέεται με 3 PINS με ένα βύσμα σειριακής θύρας στο οποίο μετά συνδέουμε στο καλώδιο USB to serial. Παρακάτω φαίνεται το hardware της επικοινωνίας του PIC με το MAX232. Το παρακάτω σχήμα έγινε με το πρόγραμμα Eagle 7.6.0. Το component MAX232 δεν υπάρχει στις έτοιμες βιβλιοθήκες εργαλείων του προγράμματος. Έτσι το component MAX232 που εμφανίζεται στο παρακάτω σχήμα φτιάχτηκε από τον γράφον. Περισσότερα για την δημιουργία νέων components στο πρόγραμμα Eagle θα αναφέρουμε στην αντίστοιχη ενότητα.



Σχήμα 6-1 Hardware για επικοινωνία του PIC18F4550 με το MAX232.

Παρατηρείται στο σχήμα 6-1 ότι ο PIC συνδέεται με τα PINS 25 (TX) και 26(RX) με το MAX232 στα PINS 10 (R2OUT)και 9(T2IN) αντίστοιχα. Από την πλευρά του μικροελεγκτή η επικοινωνία με το MAX232 υλοποιείται με hardware γι' αυτό ο PIC διαθέτει PINS RX και TX. Αν στον μικροελεγκτή που χρησιμοποιείται δεν υπάρχουν PINS RX και TX σημαίνει ότι

πρέπει να υλοποιηθεί η επικοινωνία με το MAX232 με software, και να οριστούν, όποια PINs επιλεγούν από το χρήστη, για είσοδο και έξοδο (RX και TX).

Από την πλευρά του MAX232 χρησιμοποιούμε 3 PINs για σύνδεση με τη σειριακή θύρα. Αυτά είναι τα PIN 7 (T2OUT) και PIN 8 (R2IN), και το PIN Vs- που είναι η γη του κυκλώματος. Το PIN 7 πρέπει να κολληθεί στο PIN 2 του βύσματος της σειριακής θύρας, και το PIN 8 πρέπει να κολληθεί στο PIN 3 της σειριακής. Η γη πρέπει να κολληθεί στο PIN 5 του βύσματος. Το ολοκληρωμένο MAX232 που χρησιμοποιούμε έχει τη δυνατότητα επικοινωνίας με 2 σειριακές θύρες. Έτσι τα PINS του ομαδοποιούνται σε 2 ομάδες, την 1 με τα PINs T1IN, T1OUT, R1IN και R1OUT, και την 2, με τα PINs T2IN, T2OUT, R2IN, R2OUT. Εμείς όπως φαίνεται και στο σχήμα 6-1 επιλέξαμε να χρησιμοποιήσουμε την ομάδα 2.

Τέλος δεν πρέπει να ξεφύζει από την προσοχή μας ότι για την υλοποίηση του κυκλώματος του σχήματος 6-1 χρειαζόμαστε και 4 πυκνωτές. Η χωρητικότητα των πυκνωτών είναι 1μF. Συνδέουμε τους πυκνωτές όπως φαίνεται στο σχήμα 6-1. Αν παραλειφθεί η σύνδεση των πυκνωτών το κύκλωμα δεν θα λειτουργήσει. Το ολοκληρωμένο MAX232 που χρησιμοποιούμε εδώ είναι της εταιρίας MAXIM και ονομάζεται MAX232CPE. Τα φύλλα δεδομένων του υπάρχουν στο internet αλλά περιέχονται και στο CD της παρούσας εργασίας.

Έχοντας ολοκληρώσει την κατασκευή του Hardware μπορούμε να περάσουμε στην συγγραφή του software για τη σειριακή επικοινωνία. Στο documentation του C18 compiler υπάρχει στο κεφάλαιο hardware peripheral functions περιγραφή των συναρτήσεων που προσφέρει η microchip για την σειριακή επικοινωνία (υλοποιημένη με hardware στον PIC18F4550). Μπορούν να βρεθούν στην ενότητα USART functions. Εδώ θα περιγραφεί συνοπτικά η λειτουργία των συναρτήσεων μονής σειριακής θύρας, οι συναρτήσεις πολλαπλών σειριακών θυρών λειτουργούν αντίστοιχα, απλώς αλλάζει λίγο το όνομα των συναρτήσεων, αυτό δεν θα μας απασχολήσει εδώ μιας και δουλεύουμε με μια σειριακή θύρα.

Πίνακας 6-1 Οι προσφερόμενες από την Microchip συναρτήσεις επικοινωνίας με τη σειριακή του C18 compiler.

Συνάρτηση	Περιγραφή
BusyUSART	Επιστρέφει 1 αν η σειριακή θύρα είναι απασχολημένη.
CloseUSART	Τερματισμός σειριακής επικοινωνίας
DataRdyUSART	Υπάρχουν δεδομένα διαθέσιμα στο read buffer της σειριακής?
getcUSART	Διαβάζει 1 byte από την σειριακή
getsUSART	Διαβάζει ένα string από την σειριακή
OpenUSART	Αρχικοποιεί την σειριακή επικοινωνία
putcUSART	Στέλνει ένα byte στην σειριακή
putsUSART	Στέλνει ένα string από τη μνήμη δεδομένων στην σειριακή θύρα
putrsUSART	Στέλνει ένα string από τη μνήμη προγράμματος στην σειριακή
ReadUSART	Διαβάζει ένα byte από την σειριακή
WriteUSART	Στέλνει ένα byte στην σειριακή

baudUSART	Ορίζει τα bits που καθορίζουν το baud rate χρησιμοποιείται όταν έχουμε ενισχυμένη σειριακή
-----------	--

Στο software χρησιμοποιούνται μερικές μόνο από τις παραπάνω συναρτήσεις. Αυτές τις συναρτήσεις θα περιγράψουμε αναλυτικά παρακάτω. Καταρχήν αρχικοποιούμε τις παραμέτρους της σειριακής επικοινωνίας και ξεκινάμε την επικοινωνία με τον παρακάτω κώδικα:

OpenUSART(USART_TX_INT_OFF & USART_RX_INT_OFF & USART_ASYNC_MODE & USART_EIGHT_BIT & USART_CONT_RX & USART_BRGH_HIGH, 129);

Η δήλωση της συνάρτησης OpenUSART() όπως αναφέρεται στο αντίστοιχο header αρχείο είναι η παρακάτω:

```
void OpenUSART( unsigned char config, unsigned int spbrg);
```

Δέχεται δύο παραμέτρους, την config που είναι unsigned char (παίρνει τιμές από 0 έως 255) και την spbrg που είναι ένας θετικός ακέραιος. Οι τιμές που μπορούμε να δώσουμε σε αυτές τις παραμέτρους αναλύονται παρακάτω, όπως περιγράφονται στο documentation του C18 compiler.

Η μεταβλητή config είναι ένα bitmask που δημιουργείται κάνοντας AND (&) τις τιμές από κάθε μια κατηγορία που περιγράφεται παρακάτω. Αυτές οι τιμές έχουν γίνει define στο αρχείο βιβλιοθήκης usart.h του C18 compiler.

Πίνακας 6-2 Οι defined τιμές των bit της μεταβλητής config (πρώτη παράμετρος της OpenUSART()).

Τιμή	Περιγραφή
USART_TX_INT_ON	Ενεργοποιεί το interrupt (διακοπή) εκπομπής
USART_TX_INT_OFF	Απενεργοποιεί τη διακοπή εκπομπής
USART_RX_INT_ON	Ενεργοποιεί τη διακοπή λήψης
USART_RX_INT_OFF	Απενεργοποιεί τη διακοπή λήψης
USART_ASYNC_MODE	Ασύγχρονη επικοινωνία
USART_SYNC_MODE	Σύγχρονη επικοινωνία
USART_EIGHT_BIT	8-bit αποστολή/λήψη
USART_NINE_BIT	9-bit αποστολή/λήψη
USART_SYNC_SLAVE	Slave mode στη σύγχρονη επικοινωνία
USART_SYNC_MASTER	Master mode στη σύγχρονη επικοινωνία
USART_SINGLE_RX	Μονή λήψη
USART_CONT_RX	Συνεχής λήψη
USART_BRGH_HIGH	Υψηλό Baud rate
USART_BRGH_LOW	Χαμηλό Baud rate

Η δεύτερη μεταβλητή που δέχεται η συνάρτηση OpenUSART() είναι η spbrg και καθορίζει το baud rate στο οποίο θα λειτουργεί η σειριακή θύρα. Είναι μια ακέραια μη αρνητική τιμή και

αν έχουμε επικοινωνία υψηλής ταχύτητας (USART_BRGH_HIGH) μπορεί να υπολογιστεί από τον τύπο:

$$\text{Baud Rate} = \frac{F_{osc}}{16 * (spbrg + 1)}$$

Εδώ επικοινωνούμε με Baud Rate = 9600, επομένως η τιμή που προκύπτει για το spbrg είναι 129.

Έχοντας αρχικοποιήσει την επικοινωνία με την σειριακή θύρα ο κώδικας που χρησιμοποιείται για την επικοινωνία είναι πολύ απλός. Καταρχήν ορίζεται μια μεταβλητή, η UsartmA, η οποία είναι **unsigned char, δηλαδή παίρνει τιμές από 0 έως 255**. Αυτό γίνεται γιατί οι συναρτήσεις επικοινωνίας με την σειριακή δέχονται και επιστρέφουν χαρακτήρες και συμβολοσειρές. Έτσι επειδή πρόθεσή μας είναι να πάρουμε μια τιμή από το 0 έως το 255 μέσω της σειριακή θύρας, ορίζουμε μια μεταβλητή unsigned char. **Πρέπει να προσεχθεί εδώ ότι αν ορίσει μια μεταβλητή char, αυτή θα παίρνει τιμές θετικές και αρνητικές έως το 127**. Έτσι όταν θα μετατραπεί το χαρακτήρας που παίρνουμε από τη σειριακή σε ακέραιο, με type casting, θα πάρουμε μια μέγιστη θετική τιμή το 127. Οι τιμές που στέλνονται στη σειριακή ως chars πάνω από το 127 θα μετατρέπονται με το type casting σε αρνητικές τιμές. Επειδή δεν το θέλουμε αυτό στην παρούσα εφαρμογή, δηλαδή θέλουμε να παίρνουμε μια τιμή από 0 έως 255, ορίζουμε τη μεταβλητή που θα δέχεται το χαρακτήρα μέσω σειριακής επικοινωνίας ως unsigned char. Ο κώδικας φαίνεται παρακάτω.

```
unsigned char UsartmA;
```

```
UsartmA =getcUSART();
```

Στη δεύτερη σειρά του κώδικα αποθηκεύουμε την είσοδο της σειριακής επικοινωνίας στη μεταβλητή μας. Και αυτή τη μεταβλητή τη χρησιμοποιείται όπως φαίνεται στον παρακάτω κώδικα.

```
int intmAIn;
```

```
intmAIn =(int)UsartmA;
```

```
BillsWrite(UsartmA); // Μπορεί να χρησιμοποιηθεί και ως BillsWrite(intmAIn);
```

Κάνουμε type casting σε ακέραιο, και αυτό τον ακέραιο τον δίνουμε σαν είσοδο στην συνάρτηση BillsWrite(), η οποία είναι υπεύθυνη να δώσει μια συγκεκριμένη έξοδο στην LCD οθόνη μας, ανάλογα με την είσοδο που δίνεται από την σειριακή θύρα. Βέβαια εδώ δώσαμε στην BillsWrite() κατευθείαν τη μεταβλητή UsartmA, όμως η BillsWrite() δέχεται int, επομένως το type casting γίνεται αυτόματα. Αυτός ο απλός κώδικας που παρατέθηκε παραπάνω ολοκληρώνει την σειριακή επικοινωνία. Όπως φέεται ο κώδικας του software που φτιάξαμε σε αυτή την εργασία είναι πολύ απλός και εύκολος στην κατανόηση γιατί χρησιμοποιούμε βιβλιοθήκες του C18 compiler, οι οποίες συνοδεύονται και από documentation που τις περιγράφει αναλυτικά. Η συνάρτηση BillsWrite() είναι μια δικιά μου συνάρτηση, που αναλαμβάνει να τυπώσει στην LCD ότι δεχόμαστε από τη σειριακή. Είναι επίσης εύκολη στην κατανόηση, περισσότερα για αυτή τη συνάρτηση περιέχονται στο

αρχείο επικεφαλίδας BillsLib.h και η υλοποίησή της εμφανίζεται στο BillsLib.c. Στο αρχείο βιβλιοθήκης BillsLib.h έχουμε την περιγραφή μιας μοναδικής συνάρτησης, ο κώδικας του αρχείου βιβλιοθήκης φαίνεται παρακάτω:

```
#ifndef BILLSLIB_H
#define BILLSLIB_H
#ifdef __cplusplus
extern "C" {
#endif
#ifdef __cplusplus
}
#endif
void BillsWrite( int intmAIn); /*Ορισμός της συνάρτησης που είναι υπεύθυνη να
εμφανίζει στην LCD την τιμή ρεύματος που περιμένουμε στην έξοδο.*/
#endif /* BILLSLIB_H */
```

Η συνάρτηση BillsWrite() υλοποιείται όπως παρακάτω, στο αρχείο BillsLib.c:

```
void BillsWrite( int intmAIn)
{
//Case statement begins
    switch (intmAIn) //Η switch έχει 255 περιπτώσεις, εδώ δείχνουμε μόνο 2
    {
        case 0:
            Delay1KTCYx(20); //Μια καθυστέρηση εισάγεται εδώ για εξασφάλιση ομαλής
//λειτουργίας.
            putsXLCD( "zero " ); //Στην περίπτωση που ο χρήστης έδωσε βήμα 0 στο DAC
//έχουμε στην έξοδο μηδενικό ρεύμα. Η τιμή "zero" θα εμφανιστεί στην LCD.
            break;
        case 1:
            Delay1KTCYx(20);
            putsXLCD( "0.07" ); //Για βήμα 1 στο DAC το ρεύμα μέσω της 4670hm
//αντίστασης είναι 0.07mA.
```


break; // Το break εδώ αν και γενικά τα βιβλία για C λένε ότι είναι

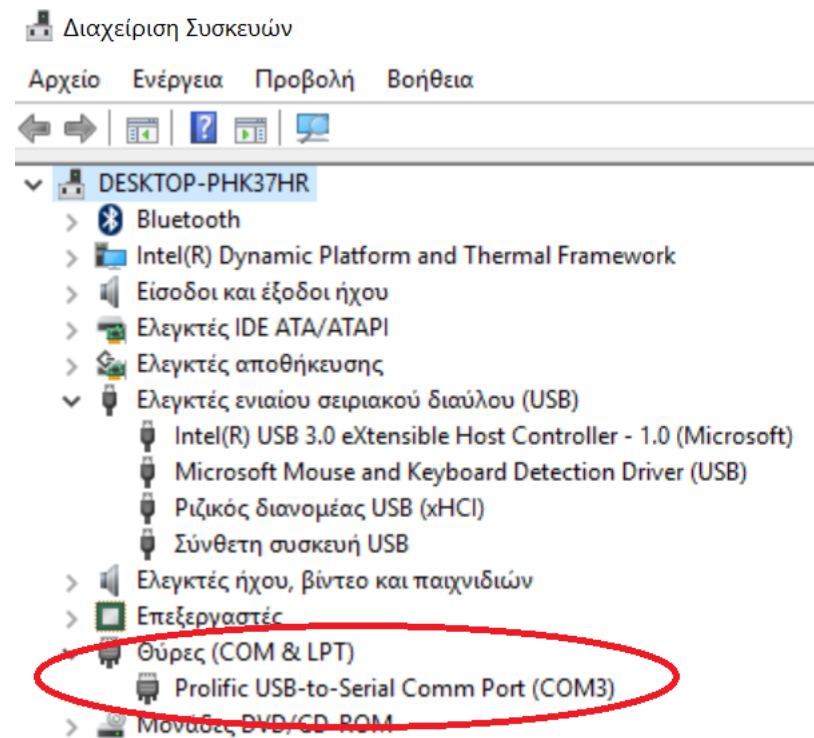
//προαιρετικό, χρειάζεται. Χωρίς το break ο κώδικας δεν θα δουλέψει με τον αναμενόμενο //τρόπο.

Καταλαβαίνουμε λοιπόν ότι η συνάρτηση BillWrite() είναι υπεύθυνη για τη βαθμονόμηση του οργάνου μας. Αν κάποιος θέλει να φτιάξει τη δική του πηγή ρεύματος, ή αν θέλει να βαθμονομήσει την πηγή που φτιάχθηκε εδώ για άλλη τιμή αντίστασης φορτίου, το μόνο που έχετε να κάνετε είναι να αλλάξετε τις τιμές που δέχεται η putrsXLCD().

6.2.Περιγραφή επικοινωνίας του Η/Υ μέσω του προγράμματος LabView με τη σειριακή θύρα.

Έχοντας ολοκληρώσει το hardware και το software για την επικοινωνία μέσω σειριακής θύρας από την πλευρά του PIC, θα περιγράψουμε τώρα πως γίνεται η επικοινωνία από την πλευρά του Η/Υ, μέσω του προγράμματος LabView.

Καταρχήν πρέπει να εγκατασταθεί το καλώδιο USB to serial. Συνδέεται σε μια θύρα USB και εγκαθίστανται οι drivers που το συνοδεύουν. Αν γίνει σωστά αυτή η εγκατάσταση θα εμφανίζεται στη διαχείριση συσκευών του συστήματός κάτι παρόμοιο με το παρακάτω σχήμα(6-2).

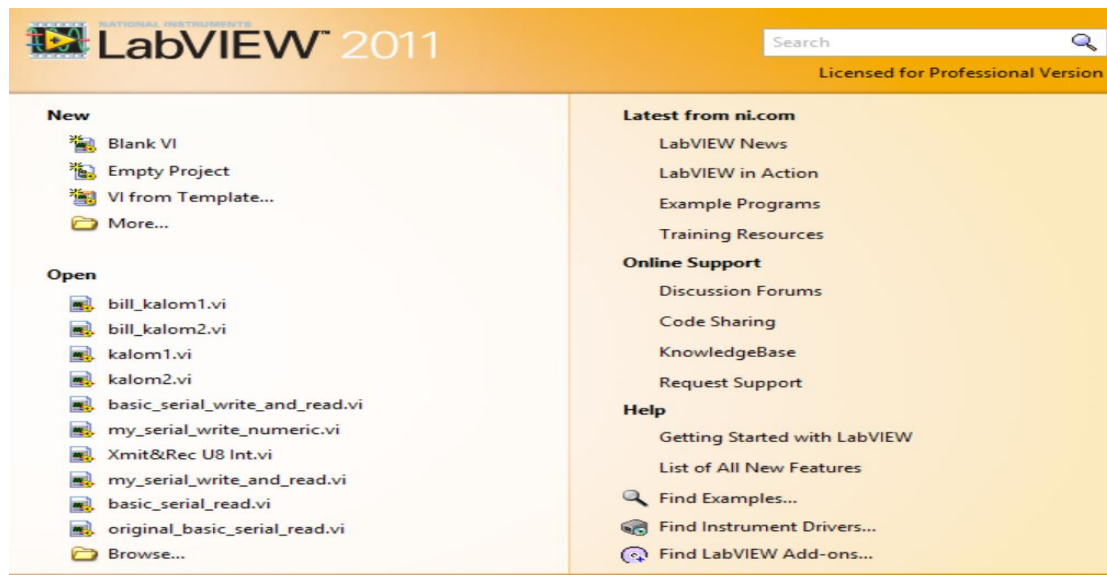


Σχήμα 6-2 Σωστή εγκατάσταση του καλωδίου USB to Serial

Το όνομα της σειριακής θύρας που έχει δημιουργηθεί πρέπει να ληφθεί υπόψη. Για το δικό μου σύστημα είναι η θύρα COM3, όπως φαίνεται στο σχήμα 6-2. Αν αποσυνδεθεί το

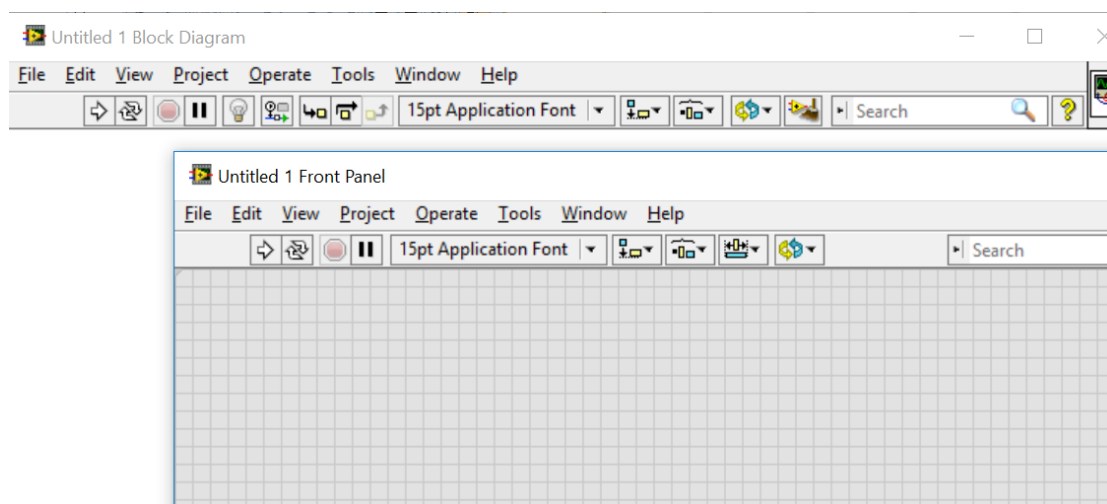
καλώδιο και το συνδεθεί σε άλλη θύρα USB, η θύρα COM θα αλλάξει όνομα (πχ. Από COM3 μπορεί να γίνει COM4), αυτό το όνομα το χρειαζόμαστε για να αρχικοποιήσουμε την επικοινωνία μέσω σειριακής θύρας.

Τη σειριακή επικοινωνία του Η/Υ με τον PIC μας θα την διαχειριστούμε από τη πλευρά του Η/Υ με το πρόγραμμα LABVIEW. Σε αυτή την εργασία χρησιμοποιούμε το LabView 2011, στην 64 bit έκδοση. Έχοντας εγκαταστήσει και τρέξει το πρόγραμμα εμφανίζεται το αρχικό interface όπως φαίνεται στην παρακάτω εικόνα.



Σχήμα 6-3 Το αρχικό menu του LabView 2011

Από το menu που φαίνεται στο σχήμα 6-3 επιλέγουμε στην κατηγορία New το Blank VI. Θα μεταφερθούμε σε ένα περιβάλλον όπως αυτό της παρακάτω εικόνας.

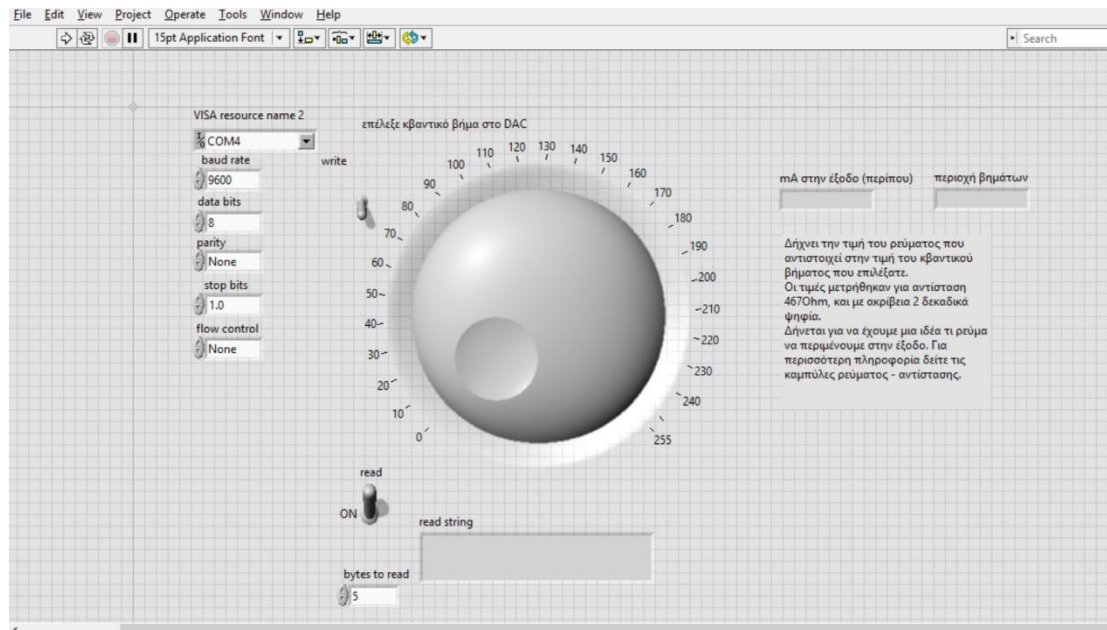


Σχήμα 6-4 Το user interface του LabView 2011

Εδώ παρατηρούμε δύο παράθυρα. Το Front Panel και το Block Diagram. Στο Front Panel ο χρήστης εισάγει αντικείμενα ελέγχου της εφαρμογής του. Σε αυτό το παράθυρο θα

υλοποιήσουμε το περιβάλλον επικοινωνίας του χρήστη με το πρόγραμμα, εν προκειμένω τη σειριακή επικοινωνία. Επομένως από αυτό το παράθυρο ο χρήστης θα δίνει τιμές που θα μεταφέρονται σειριακά στον PIC. Πίσω από το Front Panel στο σχήμα 6-4 φαίνεται το παράθυρο Block Diagram. Σε αυτό το παράθυρο θα γράψουμε τον κώδικα του προγράμματος, που θα είναι υπεύθυνος να παίρνει τις τιμές που δίνει ο χρήστης και να τις μεταδίδει σειριακά στον PIC.

Το περιβάλλον που φτιάξαμε και με το οποίο ο χρήστης θα δίνει τιμές προς τον PIC (το Front Panel) φαίνεται στο παρακάτω σχήμα (6-5).



Σχήμα 6-5 Το τελικό περιβάλλον επικοινωνίας του χρήστη με τον PIC μέσω σειριακής θύρας (Front Panel LabView 2011).

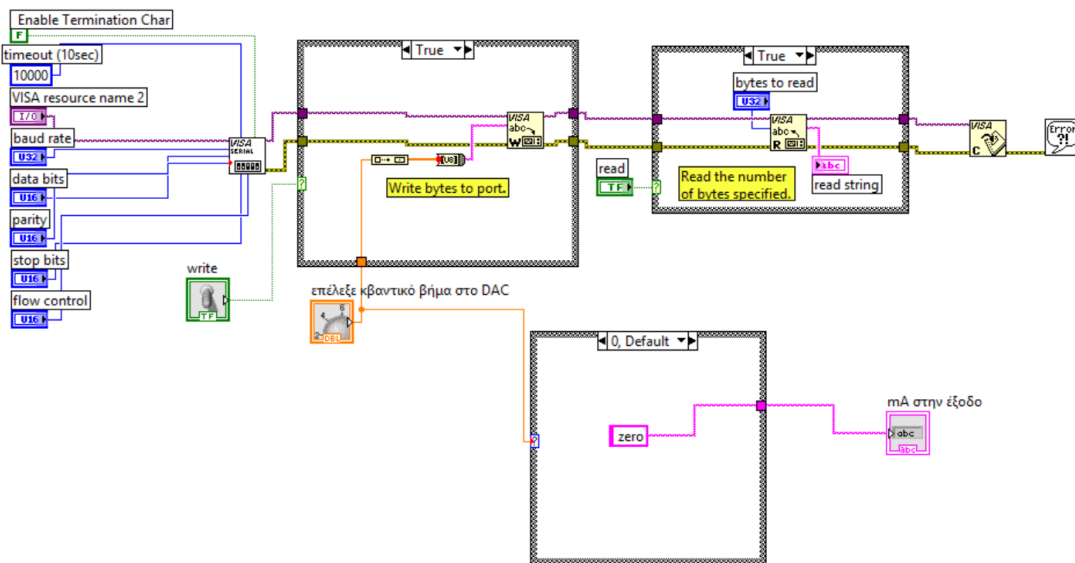
Στο κέντρο του Front Panel υπάρχει ένα knob. Με αυτό ο χρήστης μπορεί να δώσει τιμές από 0 έως 255. Οι τιμές αυτές όπως θα δούμε μετατρέπονται σε ένα χαρακτήρα και αποστέλλονται σαν χαρακτήρας πλέον στην σειριακή θύρα. Για να γίνει η επικοινωνία με τη σειριακή θύρα πρέπει να αρχικοποιηθεί αυτή με κάποιες τιμές. Τις τιμές αυτές μπορεί να τις ορίσει ο χρήστης στα πλαίσια που υπάρχουν αριστερά και πάνω στο σχήμα 6-5, αριστερά από το knob. Εδώ επιλέγουμε την COM Port που θα χρησιμοποιήσουμε για την επικοινωνία, το baud rate, τα bits που θα σταλούν, το αν θα υπάρχει parity bit, το stop bit, και τον έλεγχο ροής. Αυτές οι default τιμές όπως φαίνονται στο σχήμα 6-5 χρειάζονται για την έναρξη της σειριακής επικοινωνίας. Με αντίστοιχο τρόπο έχει αρχικοποιηθεί η επικοινωνία και από την πλευρά του PIC, επομένως αυτές οι τιμές δεν πρέπει να αλλάξουν από τον χρήστη. Το μόνο που πρέπει να αλλαχθεί για την επίτευξη της επικοινωνίας είναι να καθοριστεί σωστά το COM PORT, ανάλογα με σε ποια θύρα USB έχει μπει το καλώδιο USB to Serial.

Επομένως ο χρήστης αφού πατήσει το run (άσπρο βελάκι πάνω αριστερά στο menu του Front Panel) και σηκώσει το διακόπτη write μπορεί να γυρίσει το knob και να δώσει μια τιμή 0 ως 255 στη σειριακή θύρα. Αυτή η τιμή (με τη μορφή χαρακτήρα) θα μεταδοθεί στον PIC18F4550. Ο PIC αφού λάβει την τιμή είναι υπεύθυνος να στείλει αυτή την τιμή στο

DAC0832, το οποίο λαμβάνει τιμές ενός byte στην είσοδο δεδομένων του και τις μετατρέπει σε αναλογικό ρεύμα. Επειδή οι τιμές που μπορεί να λάβει το DAC0832 στην ψηφιακή είσοδό του είναι από 0 ως 255 γι' αυτό και το knob που σχεδιάσαμε στο Front Panel μπορεί να δώσει τιμές 0 ως 255. Οι τιμές αυτές θα μετατραπούν σε αναλογικό ρεύμα, το οποίο αφού μετατραπεί σε τάση με ένα τελεστικό ενισχυτή(OPA445AP), με μια διαδικασία που θα περιγράψουμε αναλυτικά στο αντίστοιχο εδάφιο, η τάση θα αποτελέσει είσοδο στο κύκλωμα improved Howland current rump. Είναι η τάση V2 του σχήματος 3-7.

Στο κάτω μέρος του σχήματος 6-5 έχουμε ένα text component που ονομάζεται read string. Εδώ διαβάζουμε τιμές από τη σειριακή θύρα. Οι τιμές αυτές στέλνονται από τον PIC στη σειριακή, κάθε φορά που αυτός λαμβάνει μια τιμή από το LabView. Αυτό έχει γίνει για να ξέρουμε ότι ο PIC επικοινωνεί σωστά με τον Η/Υ. Έτσι αν η επικοινωνία γίνει σωστά, η τιμή που θα στείλουμε με το knob θα εμφανιστεί στο text component. Το πόσες φορές θα εμφανιστεί καθορίζεται από το bytes to read component που υπάρχει αριστερά και κάτω στο Front Panel.

Ας ασχοληθούμε τώρα με το Block Diagram του προγράμματος που φτιάξαμε με το LABVIEW 2011. Αυτό φαίνεται ολοκληρωμένο στο παρακάτω σχήμα. Για να εμφανιστεί το block diagram επιλέγουμε από το Menu Window-> Show Block Diagram.



Σχήμα 6-6 Το τελικό Block Diagram της εργασίας στο LabView 2011

Το βασικό component που χρησιμοποιούμε είναι το visa serial. Βρίσκεται στην εργαλειοθήκη Instrument IO στο μενού Serial. Από αυτό το μενού θα χρησιμοποιήσουμε τρία components. Το visa configure serial port. Με αυτό αρχικοποιούμε την επικοινωνία με τη σειριακή θύρα. Είναι το άσπρο τετράγωνο αριστερά στο Block Diagram με την επικεφαλίδα VISA serial. Για να το αρχικοποιήσουμε αρχικοποιούμε τις σταθερές που φαίνονται αριστερά από το component. Αφού αρχικοποιήσουμε την επικοινωνία χρησιμοποιούμε το component VISA write για να γράψουμε στην θύρα. Αυτό το component στο Block Diagram είναι μέσα σε μία case structure, δεξιά από το VISA serial στο δικό μας Block Diagram. Ένας διακόπτης (που ονομάζεται write) ελέγχει την case structure, και όταν είναι ON γράφουμε στη σειριακή θύρα. Αυτό που γράφουμε στην σειριακή θύρα είναι η

τιμή που παίρνουμε από το knob, την οποία εισάγει ο χρήστης και είναι από 0 έως 255. Αυτή η τιμή μετατρέπεται σε character και στέλνεται στην σειριακή ως χαρακτήρας. Η μετατροπή γίνεται ως εξής: Η τιμή του knob στέλνεται σε ένα στοιχείο build array, μπορεί να βρεθεί στην παλέτα Programming→Arrays→Build Array. Έπειτα η έξοδος του build array στέλνεται σε ένα στοιχείο byte array to string. Αυτό βρίσκεται στην παλέτα Programming→String→Conversion→Byte Array To String. Την έξοδο από το στοιχείο byte array to string συνδέουμε στην αντίστοιχη είσοδο του Visa write.

Δεξιά από το VISA write υπάρχει το VISA read. Αυτό περιέχεται πάλι σε μια δομή case και ελέγχεται επίσης από ένα διακόπτη, τον read. Όταν ο διακόπτης είναι ON διαβάζουμε από τη σειριακή τόσα bytes όσα καθορίζονται από τη μεταβλητή bytes to read. (Υπάρχει μέσα στην case structure δεξιά στο Block Diagram). Τέλος πρέπει να κλείσει η σειριακή επικοινωνία με ένα στοιχείο Visa close.

Επιπλέον πρέπει να γίνει βαθμονόμηση του οργάνου. Δηλαδή πρέπει να δείχνουμε στον χρήστη πόσο ρεύμα θα δώσει στην έξοδό του το κύκλωμα όταν αυτός στέλνει ένα byte από το 0 ως το 255 στην σειριακή θύρα. Ή αντίστοιχα αναφερόμενοι στο DAC0832 θα λέγαμε ότι πρέπει να ειδοποιούμε το χρήστη πόσο ρεύμα στέλνει στην έξοδό του κύκλωμά μας όταν αυτός επιλέγει μια κβαντική τιμή στο DAC. Για να το κάνουμε αυτό δουλέψαμε με μια αντίσταση 467 Ohm ως το φορτίο του κυκλώματός. Φυσικά το κύκλωμα είναι πηγή ρεύματος, επομένως περιμένουμε να στέλνει σταθερό ρεύμα στην έξοδο ανεξαρτήτως της αντίστασης φορτίου. Επειδή όμως αυτό δεν ισχύει για οποιαδήποτε τιμή αντίστασης φορτίου, και καθώς ανεβαίνουμε σε τιμές πάνω από 1500 Ohm το κύκλωμα αρχίζει να μην συμπεριφέρεται σαν πηγή ρεύματος, πρέπει να αναφέρουμε στον χρήστη πόσο ήταν το φορτίο που χρησιμοποιήσαμε όταν κάναμε τη βαθμονόμηση του οργάνου. Όπως προείπαμε για την βαθμονόμηση χρησιμοποιήθηκε φορτίο 467 Ohm. Παρακάτω φαίνεται ο πίνακας βήματος στο DAC σε σχέση με το ρεύμα στην έξοδο για φορτίο 467Ohm που χρησιμοποιήθηκε στην βαθμονόμηση.

Πίνακας 6-3 Ο πίνακας βαθμονόμησης της πηγής ρεύματός μας. Έγινε με αντίσταση φορτίου 467Ohm.

Βήμα στο DAC	Ρεύμα σε mA	Βήμα στο DAC	Ρεύμα σε mA
0	0,00	128	10,15
1	0,07	129	10,24
2	0,15	130	10,30
3	0,23	131	10,41
4	0,31	132	10,49
5	0,39	133	10,58
6	0,47	134	10,64
7	0,55	135	10,73
8	0,63	136	10,81
9	0,71	137	10,88
10	0,79	138	10,96
11	0,87	139	11,05
12	0,95	140	11,11

13	1,03	141	11,20
14	1,10	142	11,28
15	1,19	143	11,35
16	1,27	144	11,43
17	1,34	145	11,52
18	1,43	146	11,61
19	1,50	147	11,67
20	1,59	148	11,76
21	1,66	149	11,84
22	1,74	150	11,91
23	1,82	151	11,99
24	1,91	152	12,08
25	1,99	153	12,14
26	2,06	154	12,23
27	2,14	155	12,31
28	2,22	156	12,40
29	2,30	157	12,46
30	2,38	158	12,55
31	2,46	159	12,63
32	2,54	160	12,70
33	2,62	161	12,78
34	2,70	162	12,87
35	2,78	163	12,93
36	2,86	164	13,02
37	2,94	165	13,10
38	3,02	166	13,19
39	3,10	167	13,25
40	3,18	168	13,34
41	3,26	169	13,43
42	3,34	170	13,49
43	3,42	171	13,58
44	3,50	172	13,66
45	3,58	173	13,73
46	3,66	174	13,81
47	3,74	175	13,90
48	3,82	176	13,96
49	3,90	177	14,05
50	3,98	178	14,13
51	4,06	179	14,20
52	4,13	180	14,28
53	4,22	181	14,35
54	4,28	182	14,43
55	4,37	183	14,52
56	4,45	184	14,58
57	4,52	185	14,67
58	4,60	186	14,73

59	4,69	187	14,82
60	4,75	188	14,90
61	4,84	189	14,97
62	4,93	190	15,05
63	5,01	191	15,12
64	5,07	192	15,22
65	5,16	193	15,31
66	5,25	194	15,37
67	5,31	195	15,44
68	5,40	196	15,52
69	5,48	197	15,61
70	5,57	198	15,67
71	5,63	199	15,74
72	5,72	200	15,82
73	5,80	201	15,89
74	5,89	202	16,00
75	5,95	203	16,06
76	6,04	204	16,12
77	6,12	205	16,21
78	6,19	206	16,27
79	6,27	207	16,34
80	6,36	208	16,42
81	6,45	209	16,49
82	6,51	210	16,55
83	6,60	211	16,62
84	6,68	212	16,68
85	6,75	213	16,75
86	6,83	214	16,83
87	6,92	215	17,00
88	6,98	216	17,04
89	7,07	217	17,11
90	7,15	218	17,17
91	7,24	219	17,24
92	7,32	220	17,30
93	7,39	221	17,37
94	7,47	222	17,43
95	7,56	223	17,49
96	7,62	224	17,56
97	7,71	225	17,62
98	7,79	226	17,69
99	7,88	227	17,75
100	7,94	228	17,82
101	8,03	229	17,88
102	8,12	230	17,92
103	8,18	231	17,97
104	8,27	232	18,03

105	8,35	233	18,09
106	8,44	234	18,09
107	8,50	235	18,09
108	8,59	236	18,09
109	8,67	237	18,09
110	8,74	238	18,09
111	8,82	239	18,09
112	8,89	240	18,09
113	8,97	241	18,09
114	9,06	242	18,09
115	9,12	243	18,09
116	9,21	244	18,09
117	9,29	245	18,09
118	9,38	246	18,09
119	9,44	247	18,09
120	9,53	248	18,09
121	9,61	249	18,09
122	9,68	250	18,09
123	9,76	251	18,09
124	9,85	252	18,09
125	9,94	253	18,09
126	10,00	254	18,09
127	10,09	255	18,20

Επειδή χρησιμοποιήσαμε αντίσταση 4670hm, ο χρήστης θα διαπιστώσει μικρές διαφορές στο ρεύμα για σχετικά μεγάλες διαφορές στην αντίσταση φορτίου από τα 4670hm. Επιπλέον μετά από κάποια Ohm το κύκλωμα δεν δουλεύει ως πηγή ρεύματος. Τέλος, υπάρχει περιορισμός στο ρεύμα, δηλαδή για ρεύματα μεγαλύτερα από κάποια τιμή (15mA για αντίσταση 1000Ohm) το κύκλωμα πάλι δεν λειτουργεί με τον αναμενόμενο τρόπο. Περισσότερα θα πούμε σε επόμενη ενότητα, όπου θα παρουσιάσουμε αναλυτικά τις χαρακτηριστικές ρεύματος αντίστασης για διάφορες τιμές ρεύματος.

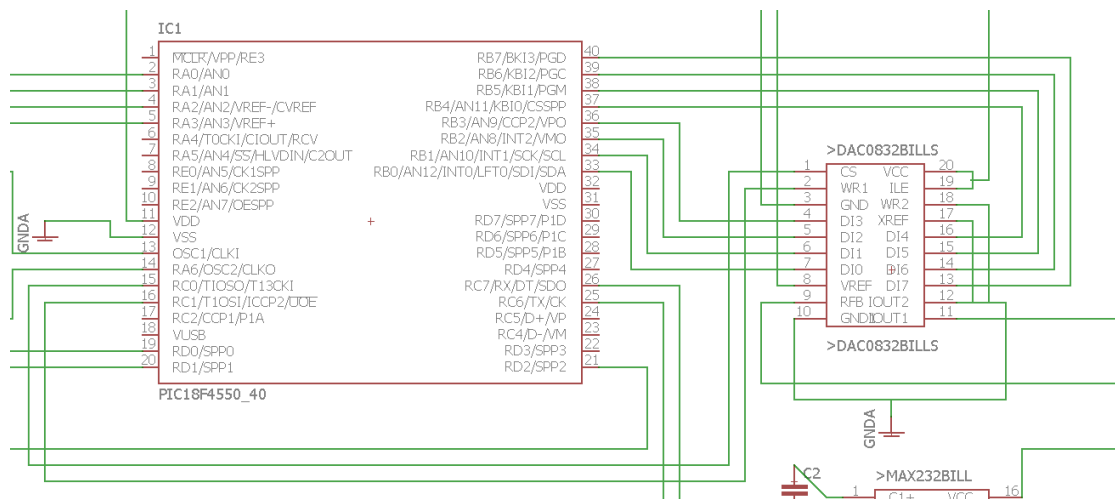
Έχοντας τον παραπάνω πίνακα βαθμονόμησης, κατασκευάστηκε μια case structure, η οποία φαίνεται στο κάτω μέρος του Block Diagram (σχήμα 6-6) στην οποία αντιστοιχίστηκε κάθε τιμή κβαντικού βήματος του DAC σε μια τιμή ρεύματος. Η τιμή ρεύματος εμφανίζεται σε ένα στοιχείο κειμένου στο πρόγραμμά μας. Έτσι όταν ο χρήστης επιλέγει μια τιμή στο knob, η τιμή αυτή μετατρέπεται σε χαρακτήρα και στέλνεται στη σειριακή θύρα. Από εκεί λαμβάνεται από τον PIC μέσω του κυκλώματος του MAX232 και μετατρέπεται από χαρακτήρα σε αριθμός μεταξύ των τιμών 0 και 255. Αφού μετατραπεί σε αριθμό, στέλνεται στο DAC0832, σαν ένα byte και έπειτα το DAC μας δίνει μια αντίστοιχη τιμή ρεύματος, η οποία μετατρέπεται σε τάση και μπαίνει στην είσοδο του κυκλώματος improved Howland. Η αντίστοιχη τιμή ρεύματος που θα περιμέναμε στην έξοδο του κυκλώματος εμφανίζεται σε ένα πλαίσιο κειμένου, που ονομάζεται "mA στην έξοδο" στο Front Panel του LabView 2011.

7. Περιγραφή Hardware και Software για την επικοινωνία του PIC με το DAC0832 και μετατροπή του ρεύματος του DAC σε τάση.

7.1. Περιγραφή Hardware για την επικοινωνία του PIC18F4550 με το DAC0832

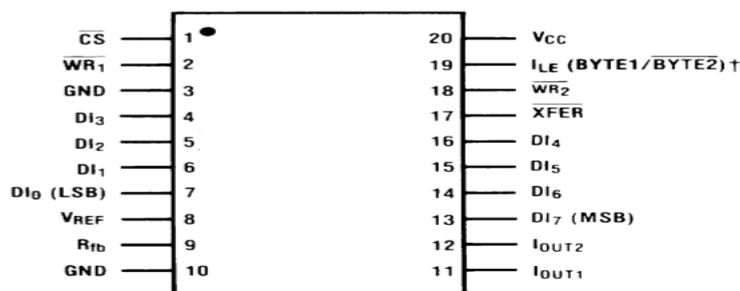
Εδώ θα περιγραφεί το Hardware που κατασκευάσαμε για την επικοινωνία του PIC18F4550 με το DAC0832. Επίσης θα περιγραφεί το software που είναι υπεύθυνο για την επικοινωνία αλλά και το Hardware που μετατρέπει το ρεύμα που βγάζει το DAC0832 στην έξοδό του σε τάση. Η τάση αυτή στέλνεται σαν είσοδο στο κύκλωμα Improved Howland, και είναι η τάση V_2 στο κύκλωμα του σχήματος 3-7. Με την περιγραφή αυτού του hardware ολοκληρώνουμε την περιγραφή όλου του hardware που κατασκευάσαμε σε αυτή την εργασία, που είναι υπεύθυνο για τον ψηφιακό έλεγχο. Ολόκληρο το Hardware, υπεύθυνο για τον ψηφιακό έλεγχο της τάσης V_2 του σχήματος 3-7 φαίνεται στο σχήμα 4-1.

Το hardware για την επικοινωνία του PIC με το DAC0832 φαίνεται στο παρακάτω σχήμα:



Σχήμα 7-1 Hardware υπεύθυνο για την επικοινωνία του PIC18F4550 με το DAC0832.

Όπως βλέπετε στο σχήμα 7-1 το DAC0832 έχει 20 PINs. Η διάταξη των PINs και η σημασία τους όπως δίνονται από τα φύλλα δεδομένων φαίνεται παρακάτω:



Σχήμα 7-2 PINs του DAC0832 στη συσκευασία PDIP.

Εδώ θα περιγραφεί τη λειτουργία του κάθε PIN. Πρώτα θα περιγραφούν τα PINs που σχετίζονται με τα σήματα ελέγχου του DAC0832.

Το PIN 1 είναι το CS (Chip Select), ενεργοποιείται όταν έχει την τιμή μηδέν (active low). Το CS σε συνδυασμό με το ILE ενεργοποιούν το WR_1 .

Το PIN 19 είναι το ILE (Input Latch Enable). Ενεργοποιείται στην τιμή ένα. Το ILE σε συνδυασμό με το CS ενεργοποιούν το WR_1 .

Το PIN 2 είναι το WR_1 (Write 1). Όταν το WR_1 είναι 0 μπορούμε να φορτώσουμε τα ψηφιακά δεδομένα από το δίαυλο δεδομένων στο μάνταλο εισόδου του DAC. Έπειτα πρέπει στείλουμε 1 στο WR_1 για να μανταλωθούν τα δεδομένα στο DAC. Για να ανανεώσουμε το μάνταλο εισόδου πρέπει τα CS και WR_1 να είναι low (0) ενώ το ILE πρέπει να είναι high (1).

Το PIN 18 είναι το WR_2 (Write 2). Ενεργό όταν είναι στο μηδέν (active low). Αυτό το PIN σε συνδυασμό με το XREF προκαλεί την εισαγωγή των 8 bit δεδομένων που είναι διαθέσιμα στο μάνταλο εισόδου στον καταχωρητή του DAC.

Το PIN 17 είναι το XREF (transfer control signal). Είναι ενεργό στο μηδέν. Όταν είναι μηδέν ενεργοποιείται το WR_2 .

Ο δίαυλος δεδομένων στο DAC είναι τα PINs **13(MSB) 14 15 16 4 5 6 7(LSB)**. Όπως γίνεται αντιληπτό ο δίαυλος δεδομένων έχει εύρος 8 bit. Κατά την χρήση του DAC θα μπορούσαμε να χρησιμοποιήσουμε για σήματα δεδομένων 4 από τα 8 bit του διαύλου. Αυτή η τεχνική εξοικονομεί PINs από την πλευρά του PIC, αλλά το βήμα στο ρεύμα εξόδου της πηγής ρεύματος θα είναι μεγαλύτερο (μικρότερη ακρίβεια στο βήμα). Εδώ επειδή έχουμε PIC με πολλά PINs χρησιμοποιούμε όλο το εύρος του διαύλου δεδομένων (και τα 8 bit).

Το PIN 11 είναι το Iout 1. (DAC current output 1). Είναι η αναλογική τιμή ρεύματος που δίνει το DAC στην έξοδό του, στην οποία μετατρέπεται η ψηφιακή είσοδος του DAC. Όταν στην είσοδο έχουμε 0 το ρεύμα στο PIN 11 θα είναι μηδέν, ενώ στην είσοδο όλα τα bit είναι 1, δηλαδή έχουμε τον αριθμό 0b11111111 (255 decimal), θα πάρουμε το μέγιστο ρεύμα που μπορεί να προσφέρει το DAC στην έξοδό του.

Το PIN 12 είναι το Iout 2 (DAC current output 2). Το PIN αυτό είναι μια ακόμη έξοδος ρεύματος του DAC. Η διαφορά με το Iout 1 είναι ότι όταν στην ψηφιακή είσοδο έχουμε μηδέν, το ρεύμα αυτό θα είναι το μέγιστο που μπορεί να δώσει το DAC ενώ όταν η ψηφιακή τιμή στην είσοδο είναι 255 το ρεύμα αυτό θα είναι μηδέν.

Το PIN 9 είναι το R_{FB} (Feedback resistor). Είναι μια αντίσταση που παρέχεται εσωτερικά στη συσκευασία του DAC και την οποία μπορούμε να χρησιμοποιήσουμε για να πάρουμε ανάδραση σε ένα εξωτερικό τελεστικό ενισχυτή, ο οποίος θα χρησιμοποιηθεί για τη μετατροπή του ρεύματος που εξάγει το DAC σε τάση. Το κύκλωμα μετατροπής του ρεύματος του DAC σε τάση θα μελετηθεί ποιο αναλυτικά παρακάτω, απλώς εδώ σημειώνουμε ότι τα φύλλα δεδομένων μας συμβουλεύουν να χρησιμοποιήσουμε τη συγκεκριμένη αντίσταση που παρέχεται στο PIN 9 ως αντίσταση ανάδρασης γιατί αυτή «παρακολουθεί» τις εσωτερικές αντιστάσεις του DAC καθώς αλλάζει η θερμοκρασία.

Το PIN 8 είναι το V_{REF} (Voltage Reference). Είναι η τάση αναφοράς. Αυτή η είσοδος συνδέει μια εξωτερική πηγή τάσης στην εσωτερική R2R σκάλα. Η τάση V_{REF} είναι υπεύθυνη για το ρεύμα που θα πάρουμε στην έξοδο του DAC. Θα δούμε παρακάτω σε αυτή την εργασία την εξίσωση που δίνει το ρεύμα στην έξοδο. Εδώ απλώς θα αναφέρουμε ότι εμείς χρησιμοποιήσαμε για τάση αναφοράς την έξοδο από το 7820 που είναι 20 Volt. Οι τιμές που μπορούμε να δώσουμε στην V_{REF} είναι σύμφωνα με τα φύλλα δεδομένων από -25 ως +25 Volt.

Το PIN 20 είναι το V_{CC} . Είναι η τάση τροφοδοσίας του DAC0832. Η τάση τροφοδοσίας στο DAC μπορεί να κυμαίνεται από 5 ως 15 Volt. Εμείς χρησιμοποιούμε ένα διαιρέτη τάσης από τα 20 Volt που δίνει το 7820 με δύο αντιστάσεις, μια στα 3.38kOhm και μια στα 9.95kOhm. Έτσι έχουμε τάση τροφοδοσίας τα 14.91Volt για το DAC. Πρέπει να αναφερθεί ότι το ρεύμα με το οποίο θα πρέπει να τροφοδοτηθεί το DAC έχει τυπική τιμή 1.2 mA και μέγιστη τιμή τα 3mA. Εδώ με το διαιρέτη τάσης που κατασκευάστηκε τροφοδοτούμε με 1.5mA το DAC, επομένως είμαστε μέσα στα όρια που δίνει ο κατασκευαστής.

Τέλος το PIN 10 είναι το GND (Ground). Αυτό το PIN χρησιμοποιείται για τη γείωση του DAC0832. Όλες οι γειώσεις στο κύκλωμά μας πρέπει να συνδέονται μεταξύ τους.

Ας δούμε τώρα πως συνδέονται τα PINs που αναφέραμε παραπάνω στον PIC18F4550 και γενικότερα το κύκλωμα.

Καταρχήν ο διάυλος δεδομένων του DAC συνδέεται με την θύρα B. Το PIN 0 της θύρας B συνδέεται στο PIN 7 του DAC που είναι και το LSB. Το PIN 7 της θύρας B συνδέεται στο PIN 13 του DAC που είναι και το MSB. Τα ενδιάμεσα PINs συνδέονται με τη αντίστοιχη σειρά. Άρα η επικοινωνία δεδομένων του PIC με το DAC δεν έχει κάποια ιδιαιτερότητα. Ορίζουμε τη θύρα B ως έξοδο στον PIC και στέλνουμε σε αυτή bytes δεδομένων με τον κλασικό τρόπο.

Έπειτα έχουμε τα PINs που είναι υπεύθυνα για τα σήματα ελέγχου στο DAC. Αυτά συνδέονται ως εξής. Το PIN ILE είναι συνδεδεμένο με την τροφοδοσία του DAC, επομένως είναι συνέχεια σε λογικό 1. Το PIN WR_2 είναι συνδεδεμένο με τη γη, επομένως είναι συνέχεια σε λογικό μηδέν. Το ίδιο συμβαίνει και με το PIN XREF. Το PIN Iout 2 είναι επίσης συνδεδεμένο με τη γη, σε αυτό το κύκλωμα χρησιμοποιούμε το PIN Iout 1 για έξοδο επομένως το Iout 2 δεν μας χρειάζεται και πρέπει να γειωθεί. Το PIN Chip Select του DAC συνδέεται στο PIN 15 του PIC που είναι το RC_0 , πρώτο bit της θύρας C. Το PIN WR_1 συνδέεται στο PIN 16 του PIC που είναι το RC_1 . Επομένως για τον έλεγχο του DAC0832 χρησιμοποιούνται τα δύο πρώτα bites της θύρας C του PIC, ενώ για την αποστολή δεδομένων χρησιμοποιείται ολόκληρη την θύρα B.

Μετά την παραπάνω περιγραφή μπορούμε να δούμε τώρα πως στέλνουμε τιμές στο DAC0832 για να μετατραπούν σε ρεύμα στο PIN Iout 1 του DAC. Έχοντας το ILE συνεχώς στην τιμή 1, όταν θέτουμε στο CS την τιμή μηδέν ενεργοποιούμε το WR_1 . Το WR_2 και το XREF τα έχουμε γειώσει, επομένως το WR_2 είναι συνεχώς ενεργοποιημένο. Αυτό σημαίνει ότι τα δεδομένα που έρχονται στο μάνδαλο του DAC μπαίνουν αμέσως στον καταχωρητή του (βγαίνουν δηλαδή πλέον στην έξοδο του DAC ως αναλογικό ρεύμα). Με αυτή τη διάταξη είναι πολύ εύκολο να στείλουμε μια τιμή μέσω του διαύλου δεδομένων στο DAC.

Το μόνο που έχουμε να κάνουμε είναι να θέσουμε το CS και το WR₁ στο μηδέν, να στείλουμε τα δεδομένα στη PORT B και έπειτα να αλλάξουμε το WR₁ σε 1 για να μανταλωθούν τα δεδομένα (που σημαίνει ότι δεν μπορούν να εισαχθούν άλλα δεδομένα μέχρι το WR₁ γίνει μηδέν και πάλι). Λόγω των τιμών που δώσαμε στα WR 2 και XREF η τιμή στο μάνδαλο θα βγει αμέσως στην έξοδο του DAC. Τέλος αλλάζουμε ξανά την τιμή του WR₁ σε 0 για να αρχίσει ο επόμενος κύκλος αποστολής δεδομένων στο DAC. Αυτή η διαδικασία υλοποιείται με software στον PIC το οποίο θα περιγραφεί αναλυτικά παρακάτω.

Πριν περάσουμε στην περιγραφή του software για την επικοινωνία με το DAC θα αναφερθεί πως γίνεται η μετατροπή της ψηφιακής τιμής που λαμβάνει το DAC στον δίαυλο δεδομένων του σε αναλογική τιμή ρεύματος στην έξοδο Iout 1 του DAC. Η εξίσωση που δίνει το ρεύμα Iout 1 δύνεται παρακάτω:

$$I_{out\ 1} = \frac{V_{REF}}{15k\Omega} * \frac{Digital\ Input}{256}$$

Εξηγώντας τις παραμέτρους της παραπάνω εξίσωσης, το V_{REF} όπως προαναφέραμε είναι η τάση αναφοράς που δώσαμε στο DAC (20Volt). 15kΩm είναι η R_{FB}, η εσωτερική αντίσταση ανάδρασης του DAC, Digital Input είναι η ψηφιακή είσοδος του DAC που παίρνει τιμές από 0 ως 255 και διαιρείται με το 256. Έτσι για παράδειγμα αν έχουμε 0 στην ψηφιακή είσοδο το Iout 1 θα είναι μηδέν, ενώ αν έχουμε 255 στο δίαυλο δεδομένων θα έχουμε το μέγιστο ρεύμα Iout 1 περίπου 1.3mA στην έξοδο.

Μια αντίστοιχη εξίσωση ισχύει και για το ρεύμα Iout2:

$$I_{out\ 2} = \frac{V_{REF}}{15k\Omega} * \frac{255 - Digital\ Input}{256}$$

Εδώ δεν χρησιμοποιείται το Iout 2 και επομένως ο παραπάνω τύπος δεν χρησιμοποιείται αναφέρεται απλώς για την πληρότητα του κειμένου.

7.2.Περιγραφή Software για την επικοινωνία του PIC18F4550 με το DAC0832

Ας δούμε τώρα τον κώδικα που υλοποιεί την επικοινωνία του PIC18F4550 με το DAC0832. Ο κώδικας που καταστρώθηκε για την επικοινωνία με το DAC είναι πολύ απλός και αποτελείται από μια μοναδική συνάρτηση. Σε αυτή τη συνάρτηση δίνουμε μια ακέραια τιμή και αναλαμβάνει να τη στείλει στην έξοδο του DAC, τροποποιώντας κατάλληλα τα σήματα ελέγχου. Ο κώδικας φαίνεται παρακάτω:

```
void dac(int quando)
```

```
{
```

```
    Delay10TCYx(20); //Εισάγεται μια καθυστέρηση για να εξασφαλισθεί η σωστή
//επικοινωνία με το DAC (για να προλαβαίνει δηλαδή το DAC να ανταποκρίνεται στις νέες
//τιμές που έρχονται στην είσοδό του).
```

```

DAC_CS_PIN =0; //Το σήμα ελέγχου CS στο DAC

DAC_WR1_PIN=0; //Το σήμα ελέγχου WR1 στο DAC. Παραπάνω περιγράψαμε τι τιμές
//πρέπει να δώσουμε σε αυτά τα σήματα.

PORTB=qvando; //Στέλνουμε στην PORTB τον δεκαδικό που θέλουμε

Delay10TCYx(20); //Πάλι μια καθυστέρηση για την εξασφάλιση σωστής λειτουργίας

DAC_WR1_PIN=1; //Μόλις το WR1 γίνει 1 τα δεδομένα μανταλώνονται στην είσοδο και
//λόγω των τιμών του WR2 και XREF τα δεδομένα βγαίνουν σαν αναλογικό ρεύμα στην
//έξοδο

DAC_CS_PIN =0; //Κρατάμε το CS στο μηδέν

DAC_WR1_PIN=0; //Αλλάζουμε το WR1 για να είναι έτοιμο το DAC να λάβει καινούρια
//δεδομένα

Delay10TCYx(20); //Εισάγουμε μια μικρή καθυστέρηση.

}

```

Για να μπορέσει να χρησιμοποιηθεί η συνάρτηση dac() που περιγράφηκε παραπάνω πρέπει να οριστούν μερικές σταθερές, αυτές είναι:

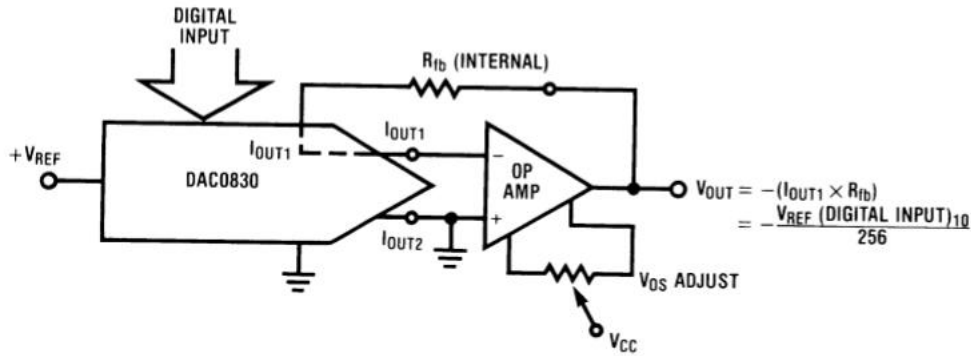
```
#define DAC_CS_PIN LATCbits.LATC0
```

```
#define DAC_WR1_PIN LATCbits.LATC1
```

Τα παραπάνω #define πρέπει να εισαχθούν στην αρχή του κώδικα. Ανάλογα με το ποια PINS του μικροελεγκτή επιλέχθηκαν για τα σήματα ελέγχου CS και WR1 θα πρέπει να οριστούν οι αντίστοιχες θύρες, όπως στήθηκε το κύκλωμα είναι για το CS η θύρα C0 και για το WR₁ η C1. Αξίζει να αναφερθεί ότι χρησιμοποιούμε τις εντολές LAT και όχι PORT. Στην συγκεκριμένη εφαρμογή δεν θα ήχε διαφορά αν επιλέγαμε να ορίσουμε τις θύρες με τις εντολές PORT αλλά γενικά είναι καλύτερη πρακτική να ορίζονται με την εντολή LAT γιατί έτσι είναι εύκολο να ελεγχθεί τι εμφανίζεται στην έξοδο.

7.3.Μετατροπή του ρεύματος εξόδου του DAC0832 σε τάση

Για τη μετατροπή του ρεύματος που δίνει το DAC στην έξοδό του σε τάση χρησιμοποιούμε έναν τελεστικό ενισχυτή σε συνδεσμολογία αναστρέφουσας ανάδρασης τάσης. Η διάταξη φαίνεται παρακάτω:



Σχήμα 7-3 Μετατροπή του ρεύματος εξόδου του DAC0832 σε τάση.

Το ρεύμα εξόδου του DAC εφαρμόζεται στην αναστρέφουσα είσοδο του τελεστικού ενισχυτή. Ο τελεστικός έχει πολύ μεγάλη σύνθετη αντίσταση εισόδου. Το ρεύμα που εισέρχεται στη μη αναστρέφουσα είσοδο είναι μηδέν (είναι γειωμένη). Όπως φαίνεται στο σχήμα στον κλάδο της ανάδρασης χρησιμοποιούμε την αντίσταση R_{FB} που μας παρέχεται από το DAC. Η εξίσωση που δίνει την τάση στην έξοδο του κυκλώματος δίνεται παρακάτω:

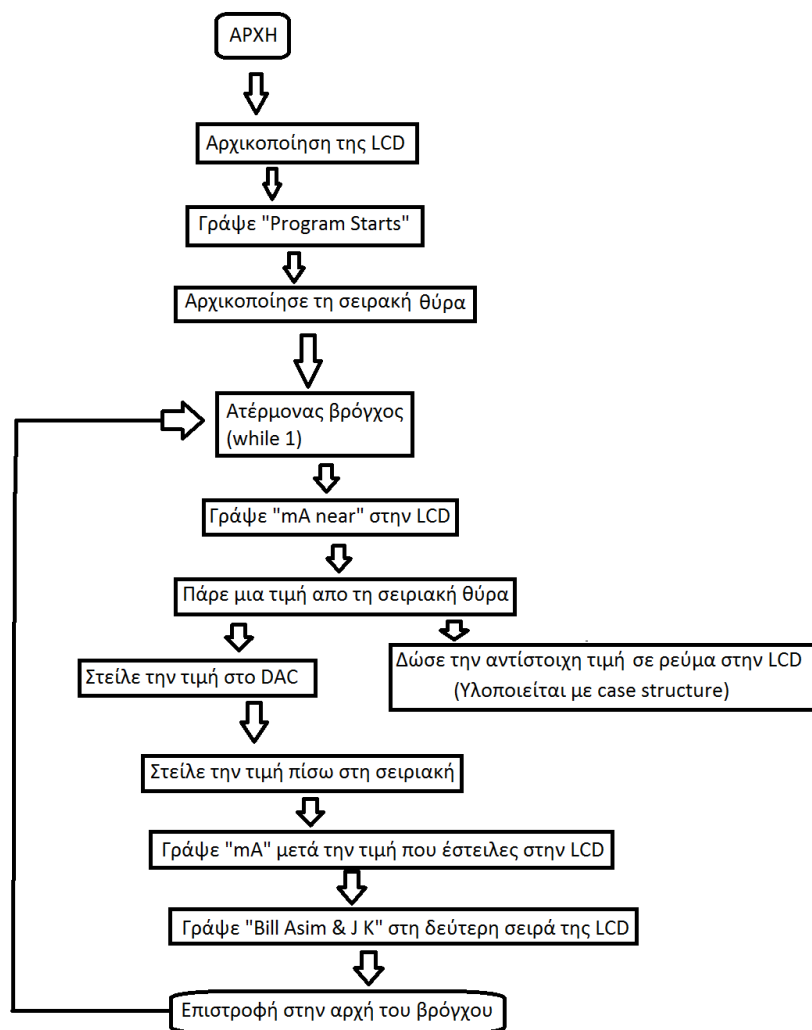
$$V_{out} = -(I_{out} * R_{FB}) = -\frac{V_{REF}(Digital Input)}{256}$$

Έτσι όταν δίνεται στο DAC είσοδος μηδέν, η τάση στην έξοδο του κυκλώματος θα είναι μηδέν, ενώ όταν δίνεται είσοδος το 256 η τάση στην έξοδο θα είναι $-20.(=V_{REF})$.

Επιπλέον, όπως δείχνεται και στο σχήμα μπορούμε να τριμάρουμε την θετική τροφοδοσία του τελεστικού, αυτή η διαδικασία λέγεται zero adjustment. Επειδή το DAC εκτός από το ρεύμα που δίνει στην έξοδό του, εμφανίζει και μια τάση, τριμάροντας την θετική τροφοδοσία του τελεστικού μπορούμε να μηδενίσουμε τη ανεπιθύμητη θετική τάση που εμφανίζεται στην έξοδο του DAC, για να πάρουμε μεγαλύτερη ακρίβεια στο βήμα του DAC. Εμείς δεν χρειάστηκε να τριμάρουμε τη θετική τροφοδοσία σε αυτή την εφαρμογή.

8. Παρουσίαση του τελικού κώδικα με τον οποίο προγραμματίσαμε τον PIC18F4550

Σε αυτή την ενότητα θα παρουσιάσει ο τελικός κώδικας με τον οποίο προγραμματίστηκε ο PIC18F4550. Καταρχήν θα παρουσιαστεί το διάγραμμα ροής. Είναι επίτηδες πολύ απλό. Θα μπορούσαμε να εμπλουτίσουμε τον κώδικα, για παράδειγμα θα μπορούσαμε να προνοήσουμε ώστε αν η αρχικοποίηση της σειριακής ή της επικοινωνίας με την LCD δεν καταστεί δυνατή, ένα μήνυμα να εμφανίζεται στην LCD ή να στέλνεται στον Η/Υ που να πληροφορεί τον χρήστη για το σχετικό σφάλμα που συνέβη, αυτό όμως δεν κρίθηκε αναγκαίο, και γενικώς προσπαθήσαμε να κρατήσουμε τον κώδικα όσο πιο απλό γίνεται.



Σχήμα 8-1 Διάγραμμα ροής του κώδικα με τον οποίο προγραμματίσαμε τον PIC18F4550.

Η υλοποίηση με κώδικα c φαίνεται παρακάτω. Αυτός ο κώδικας περιέχεται στο αρχείο lcdMax.c, που είναι και το αρχείο που περιέχει την συνάρτηση main() του προγράμματός μας.

```

#include <stdio.h>

#include <stdlib.h>

#include <p18f4550.h>

#include "xlcd.h"

#include "usart.h"

#include <delays.h>

#include "BillsLib.h"

#define DAC_CS_PIN LATCbits.LATC0

#define DAC_WR1_PIN LATCbits.LATC1

#pragma config FOSC =HS

#pragma config WDT = OFF

#pragma config LVP = OFF

void dac(int qvando);

void main(void) {

    //char myNum[3];

    //char *p;

    //int myIntOut;

    //char mAChar[3];

    unsigned char UsartmA;

    int intmAIn;

    OpenXLCD(FOUR_BIT & LINES_5X7);

    while(BusyXLCD());

    Delay1KTCYx(140);

    TRISB=0x00; //b port exodoi

    PORTB=0x00;

    TRISC=0x00; //C port exodoi

    putrsXLCD( "Program" );

```



```

while(BusyXLCD());

WriteCmdXLCD(0xC0);

while(BusyXLCD());

putsXLCD( "Starts" );

while(BusyXLCD());

Delay10KTCYx(55);

    OpenUSART(USART_TX_INT_OFF & USART_RX_INT_OFF & USART_ASYNCH_MODE &
USART_EIGHT_BIT & USART_CONT_RX & USART_BRGH_HIGH, 129);

    while(BusyUSART());

    Delay10KTCYx(10);

    while(BusyXLCD());

    while(BusyXLCD());

    Delay10KTCYx(100);

    WriteCmdXLCD(0x01); //clear display

    Delay10KTCYx(10);

    WriteCmdXLCD(0x0C);

    while(1)
    {

        WriteCmdXLCD(0x01);

        Delay1KTCYx(20);

        WriteCmdXLCD(0x0C); //so that same message appears on same place in lcd

        Delay1KTCYx(20);

        Delay1KTCYx(20);

        putsXLCD( "mA near " );

        Delay10KTCYx(20);

        UsartmA =getcUSART();

        intmAIn =(int)UsartmA;

        BillsWrite(UsartmA); //Case statement here

```

```

    dac(intmAIn);

    Delay10KTCYx(20);

    putcUSART((char)intmAIn);

    while(BusyXLCD());

    putrsXLCD("mA");

    while(BusyXLCD());

    Delay10KTCYx(10);

    SetDDRamAddr( 0x28 );

    while(BusyXLCD());

    Delay10KTCYx(10);

    putrsXLCD("Bill Asim & J K");

    while(BusyXLCD());

    Delay10KTCYx(40);
}

void DelayFor18TCY(void)
{
    Delay10TCYx(10); //10 instruction cycle

    Delay1TCY(); //1 instruction cycle

    Delay1TCY();

    Delay1TCY();

    Delay1TCY();

    Delay1TCY();

    Delay1TCY();

    Delay1TCY();

    Delay1TCY();

    Delay1TCY(); //18 cycles total

    // __delay_us(18);
}

```

```

/*-----*/
void DelayPORXLCD(void)
{
    Delay1KTCYx(90);

    //__delay_ms(15);
}

/*-----*/

void DelayXLCD(void)
{
    Delay1KTCYx(40);

    //__delay_ms(5);
}

void dac(int qvando)
{
    Delay10TCYx(20);

    DAC_CS_PIN =0;

    DAC_WR1_PIN=0;

    PORTB=qvando;

    Delay10TCYx(20);

    DAC_WR1_PIN=1;

    DAC_CS_PIN =0;

    DAC_WR1_PIN=0;

    Delay10TCYx(20);
}

```

Η επεξήγηση του παραπάνω κώδικα έχει δοθεί κατά τμήματα στα αντίστοιχα παραπάνω κεφάλαια. Εδώ θα παραθέσουμε τις συναρτήσεις που φτιάξαμε εμείς και θα αναφέρουμε που ορίζονται και που υλοποιούνται.

Η `void dac(int quando)` ορίζεται και υλοποιείται στο αρχείο που περιέχει τον παραπάνω κώδικα (Τοπικά στο `LcdMax.c`). Όπως εξηγήθηκε σε προηγούμενο κεφάλαιο αυτή η συνάρτηση αναλαμβάνει την αποστολή τιμών στο DAC.

Η `BillsWrite(UsartmA)` αναλαμβάνει να στείλει την κατάλληλη τιμή στην LCD, ανάλογα με την τιμή βήματος του DAC που παίρνουμε από τον χρήστη (βαθμονόμηση οργάνου). Ορίζεται στο αρχείο `BillsLib.h` και υλοποιείται με `case structure` στο αρχείο `BillsLib.c`.

Οι συναρτήσεις `void DelayFor18TCY(void)`, `void DelayPORXLCD(void)` και `void DelayXLCD(void)` υλοποιούνται από τον χρήστη της βιβλιοθήκης `xlcd.h`. Ορίζονται στο αρχείο επικεφαλίδας `xlcd.h` και υλοποιούνται στο αρχείο `LcdMax.c`. Απαιτούνται από το αρχείο βιβλιοθήκης `xlcd.h` και χρειάζονται για την σωστή επικοινωνία με την LCD.

Επίσης το αρχείο `xlcd.h` πρέπει να τροποποιηθεί ανάλογα με το τι `hardware` χρησιμοποιούμε και πώς το έχουμε συνδέσει. Αυτή η τροποποίηση περιγράφηκε αναλυτικά στο κεφάλαιο που ασχολείται με την LCD.

9. Χαρακτηρισμός της πηγής ρεύματος improved Howland Current Pump

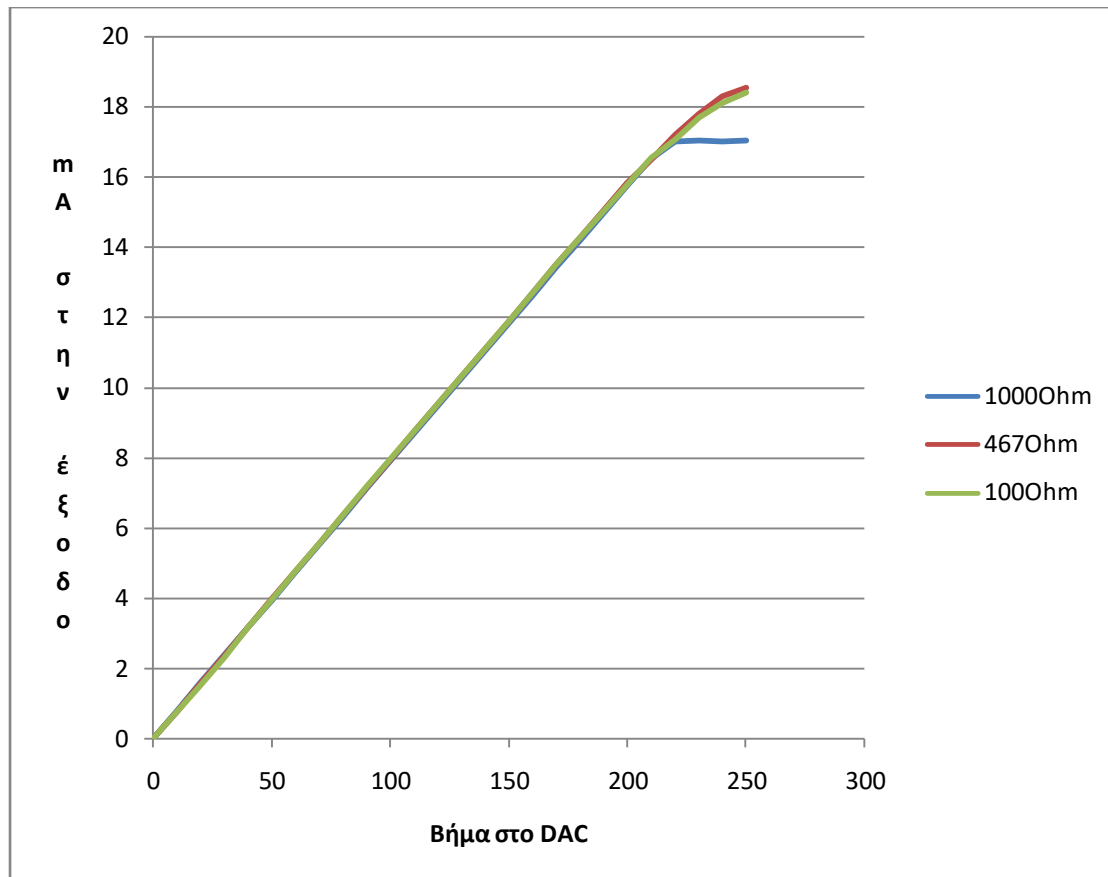
Έχοντας περιγράψει όλο το hardware και το software της ψηφιακά ελεγχόμενης πηγής ρεύματος, θα προχωρήσουμε σε αυτό το κεφάλαιο στον χαρακτηρισμό της. Δηλαδή θα κατασκευαστούν και θα παρουσιαστούν χαρακτηριστικές τάσης εισόδου και ρεύματος εξόδου για διάφορες τιμές αντίστασης φορτίου στην πηγή ρεύματος improved Howland.

Αυξάνοντας το βήμα στο DAC κατά 10, για τις αντιστάσεις 100, 467 και 1000 Ohm, ο πίνακας τιμών των βημάτων στο DAC σε σχέση με το ρεύμα στην έξοδο φαίνεται παρακάτω:

Πίνακας 9-1 Χαρακτηριστικές ρεύματος εξόδου σε σχέση με το βήμα στο DAC, για 100, 467 και 1000 Ohm.

Βήμα στο DAC	mA στα 1000Ohm	mA στα 467Ohm	mA στα 100Ohm
0	0	0	0
10	0,79	0,77	0,76
20	1,58	1,56	1,5
30	2,38	2,35	2,3
40	3,16	3,18	3,17
50	3,95	3,98	3,97
60	4,74	4,75	4,76
70	5,53	5,56	5,56
80	6,32	6,36	6,36
90	7,12	7,15	7,16
100	7,91	7,94	7,96
110	8,7	8,76	8,75
120	9,48	9,55	9,55
130	10,26	10,3	10,3
140	11,05	11,1	11,1
150	11,85	11,9	11,9
160	12,63	12,72	12,71
170	13,42	13,51	13,5
180	14,2	14,28	14,27
190	14,98	15,07	15,05
200	15,75	15,82	15,77
210	16,51	16,5	16,55
220	17	17,19	17,07
230	17,03	17,81	17,7
240	17,02	18,3	18,1
250	17,04	18,54	18,4

Η αντίστοιχη γραφική παράσταση φαίνεται στο παρακάτω σχήμα.



Σχήμα 9-1 Χαρακτηριστικές ρεύματος εξόδου σε σχέση με το βήμα στο DAC για 100, 467 και 1000 Ohm.

Όπως φαίνεται από τη γραφική, η πηγή ρεύματος λειτουργεί σωστά ως τα 18mA περίπου για τιμές αντιστάσεων ως τα 467 Ohm, ενώ για τιμές πάνω από 1000 Ohm λειτουργεί σωστά ως τα 17mA περίπου.

Για τις αντιστάσεις 1563, 2390 και 4740 Ohm, κατασκευάστηκαν οι παρακάτω πίνακες τιμών που δίνουν την έξοδο σε mA και την αντίστοιχη έξοδο σε τάση της πηγής ρεύματος σε σχέση με το βήμα στο DAC. Το βήμα στο DAC αυξάνει πάλι κατά 10. Οι πίνακες φαίνονται παρακάτω:

Πίνακας 9-2 mA στην έξοδο και τάση στην έξοδο σε σχέση με το βήμα στο DAC για αντίσταση 1563 Ohm

Αντίσταση 1563 Ohm		
Βήμα στο DAC	mA στην έξοδο	Τάση στην έξοδο
0	0,01	0,013
10	0,78	1,226
20	1,57	2,45
30	2,36	3,69
40	3,15	4,93
50	3,95	6,17
60	4,74	7,41
70	5,53	8,65
80	6,33	9,89
90	7,11	11,12
100	7,91	12,36

110	8,70	13,6
120	9,49	14,84
130	10,27	16,05
140	11,06	17,28
150	11,85	18,52
160	12,64	19,75
170	13,37	20,9
180	13,44	21
190	13,44	21
200	13,44	21
210	13,44	21
220	13,44	21
230	13,44	21
240	13,44	21
250	13,44	21
255	13,44	21

Ο αντίστοιχος πίνακας για αντίσταση 2390 Ohm παρατίθεται παρακάτω:

Πίνακας 9-3 mA στην έξοδο και τάση στην έξοδο σε σχέση με το βήμα στο DAC για αντίσταση 2390Ohm

Αντίσταση 2390 Ohm		
Βήμα στο DAC	mA στην έξοδο	Τάση στην έξοδο
0	0,01	0,02
10	0,78	1,869
20	1,56	3,74
30	2,36	5,63
40	3,15	7,52
50	3,93	9,4
60	4,72	11,29
70	5,51	13,18
80	6,30	15,06
90	7,09	16,95
100	7,88	18,83
110	8,39	20,06
120	9,41	22,5
130	10,17	24,3
140	10,50	25,1
150	10,50	25,1
160	10,50	25,1
170	10,50	25,1
180	10,50	25,1
190	10,50	25,1
200	10,50	25,1
210	10,50	25,1
220	10,50	25,1
230	10,50	25,1
240	10,50	25,1
250	10,50	25,1

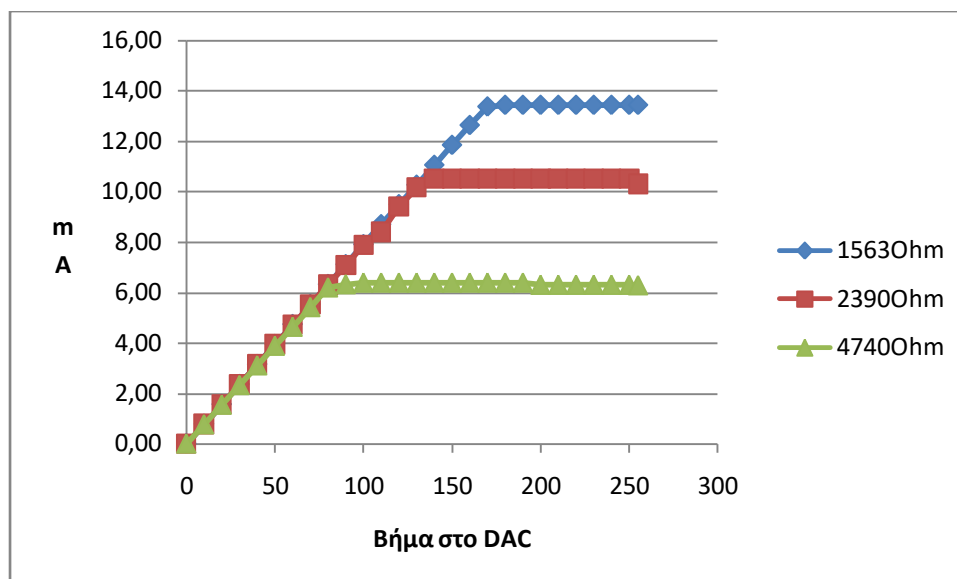
255	10,29	24,6
-----	-------	------

Ο πίνακας για αντίσταση 2390 Ohm παρατίθεται παρακάτω:

Πίνακας 9-4 mA στην έξοδο και τάση στην έξοδο σε σχέση με το βήμα στο DAC για αντίσταση 4740Ohm

Αντίσταση 4740Ohm		
Βήμα στο DAC	mA στην έξοδο	Τάση στην έξοδο
0	0,01	0,038
10	0,77	3,65
20	1,55	7,34
30	2,33	11,03
40	3,11	14,74
50	3,89	18,43
60	4,64	22
70	5,42	25,7
80	6,20	29,4
90	6,33	30
100	6,39	30,3
110	6,39	30,3
120	6,39	30,3
130	6,39	30,3
140	6,39	30,3
150	6,39	30,3
160	6,39	30,3
170	6,39	30,3
180	6,39	30,3
190	6,39	30,3
200	6,31	29,9
210	6,31	29,9
220	6,31	29,9
230	6,31	29,9
240	6,31	29,9
250	6,31	29,9
255	6,29	29,8

Η παρακάτω γραφική παράσταση μας δείχνει τις χαρακτηριστικές βήματος στο DAC σε σχέση με το ρεύμα εξόδου της πηγής ρεύματος για φορτίο 1563, 2390 και 4740 Ohm.



Σχήμα 9-2 Χαρακτηριστική ρεύματος στην έξοδο σε σχέση με το βήμα στο DAC για 1563 2390 και 4740 Ohm φορτίο.

Παρατηρώντας τις παραπάνω χαρακτηριστικές και συγκρίνοντας αυτές με τις χαρακτηριστικές του σχήματος 9-1 διαπιστώθηκαν τα εξής. Καταρχήν βλέπουμε ότι όσο μεγαλώνει το φορτίο η πηγή ρεύματος σταματά να λειτουργεί σωστά σε όλο και μικρότερα ρεύματα. Στα 1563 Ohm η πηγή δουλεύει σωστά ως τα 13mA περίπου ενώ για φορτίο 4740Ohm η πηγή δουλεύει σωστά ως τα 5,5mA περίπου. Στις χαρακτηριστικές του σχήματος 9-1 η πηγή ρεύματος δουλεύει σωστά ως τα 18mA περίπου.

Από τον πίνακα 9-4 βλέπουμε ότι η πηγή μας μπορεί να δώσει τάση ως και 29,8Volt. Επίσης από τον πίνακα 9-1 βλέπουμε ότι η πηγή μας δίνει ρεύμα ως 18.5mA. Παρόλα αυτά βλέπουμε ότι όσο ανεβάζουμε την αντίσταση, πρέπει να περιμένουμε η πηγή μας να δουλεύει σωστά για όλο και μικρότερα ρεύματα. Τα 30Volt μέγιστη τάση που παίρνουμε από τον τελεστικό είναι αρκετά χαμηλότερα από την τάση τροφοδοσίας του τελεστικού μας, που είναι 43Volt, ενώ τα 18,5mA μέγιστο ρεύμα που παίρνουμε είναι επίσης αρκετά χαμηλότερα από τα 25mA ρεύματος βραχυκυκλώματος(short circuit current) που μπορεί να δώσει ο τελεστικός μας. Όμως εδώ δεν έχουμε βραχυκύκλωμα αλλά έχουμε ένα φορτίο που κυμαίνεται από 100 ως 4740 Ohm και προφανώς το ρεύμα που μπορεί να μας δώσει ο τελεστικός για τα αντίστοιχα φορτία μειώνεται αντιστοίχως. Επομένως συμπεραίνουμε ότι τα όρια που βλέπουμε στο κύκλωμα μας οφείλονται στο χαρακτηριστικό short circuit current του τελεστικού ενισχυτή μας. Έτσι αν θέλουμε να κατασκευάσουμε καλύτερη πηγή ρεύματος, ένας τρόπος βελτίωσης θα ήταν να επιλέξουμε τελεστικό ενισχυτή με μεγαλύτερο ρεύμα βραχυκυκλώματος (short circuit current). Υπάρχουν και άλλοι τρόποι βελτίωσης, οι οποίοι αναπτύχθηκαν στο κεφάλαιο 3,5.

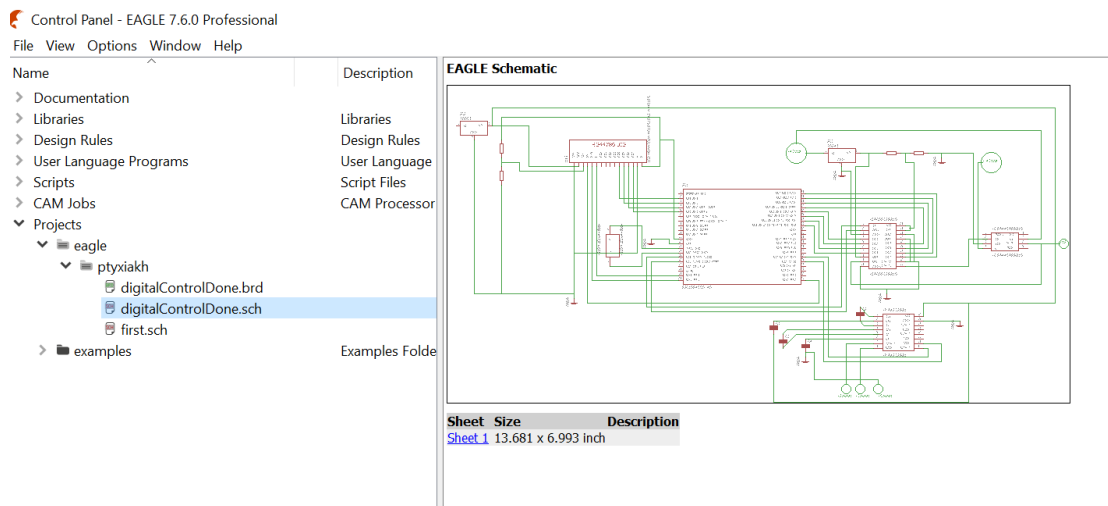
Βιβλιογραφία

ALBERT PAUL MALVINO «Βασική Ηλεκτρονική»	1
Κ.Α. ΚΑΡΥΜΠΑΚΑΣ «Ηλεκτρονική»	2
John Caldwell «A High-Voltage Bidirectional Current Source» Analog Applications Engineer, Texas Instruments	3
Texas Instruments «Comprehensive Study of the Howland Current Pump» AN-1515 A	4
JERRY STEELE and TIM GREEN «Proper Compensation Techniques Will Let The Improved-Howland And Floating-Load Current Sources Perform Up to Par» Apex Microtechnology Corp	5
Microchip C18 Documentation	6
LabView 2011 Help https://zone.ni.com/reference/en-XX/help/371361H-01/	7
Learn LabView http://www.ni.com/academic/students/learn-labview/	8
Interfacing a PIC to LCD: Introduction http://www.pyroelectro.com/tutorials/pic_lcd/	9
Using UART of PIC Microcontroler https://electrosome.com/uart-pic-microcontroller-mplab-xc8/	10
Create Custom Parts in Eagle https://www.youtube.com/watch?v=U36_et5Unxl	11
D. H. Sheingold «Impedance & Admittance Transformations using Operational Amplifiers» περιοδικό "Lightning Empiricist", Τεύχος 12, Αριθμός 1 www.philbrickarchive.org/1964-1_v12_no1_the_lightning_empiricist.htm	12

Παράρτημα 1:Κατασκευή κυκλώματος ψηφιακού ελέγχου στο Eagle 7.6.0

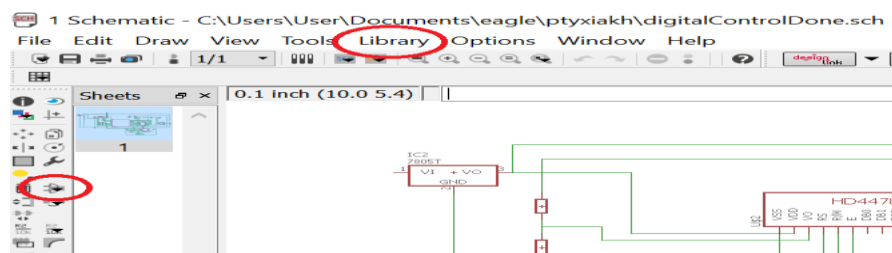
Το κύκλωμα που φαίνεται στο σχήμα 4-1 της εργασίας έχει κατασκευαστεί με το πρόγραμμα Eagle 7.6.0. Κάποια από τα στοιχεία του κυκλώματος αυτού είναι διαθέσιμα στις βιβλιοθήκες του προγράμματος, κάποια άλλα όμως δεν είναι διαθέσιμα και έτσι χρειάστηκε να σχεδιαστούν από την αρχή. Στο παράρτημα αυτό θα παρουσιαστεί ο σχεδιασμός του κυκλώματος ψηφιακού ελέγχου της πηγής ρεύματος όπως έγινε με το πρόγραμμα Eagle. Το παράρτημα 1 μπορεί να θεωρηθεί μια εισαγωγή στο πρόγραμμα Eagle, αν και θα παρουσιαστούν μόνο οι δυνατότητες του προγράμματος που μας ενδιαφέρουν για το σχεδιασμό του κυκλώματός αυτής της εργασίας.

Έχοντας εγκαταστήσει το πρόγραμμα μπαίνουμε στο αρχικό interface (διεπαφή) του προγράμματος. Όπως φαίνεται παρακάτω:



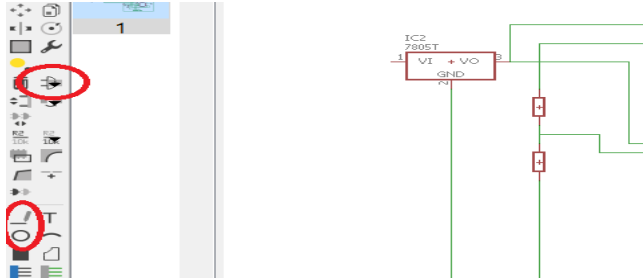
Σχήμα 0-1 Το αρχικό interface του Eagle 7.6.0.

Όπως βλέπουμε στο σχήμα 10-1 έχουμε ήδη φτιάξει το κύκλωμα και μια προεπισκόπηση δίνεται στα δεξιά. Αν κάποιος ξεκινά ένα νέο σχέδιο αρχικά δεν θα υπάρχει κάποιο κύκλωμα στα δεξιά. Δίνουμε File ->New->Schematic.Θα ζητηθεί ένα όνομα για το νέο σχέδιο υπό κατασκευή. Αφού ονομάσουμε το αρχείο, μεταφερόμαστε σε ένα νέο παράθυρο όπου μπορεί να αρχίσει η σχεδίαση.



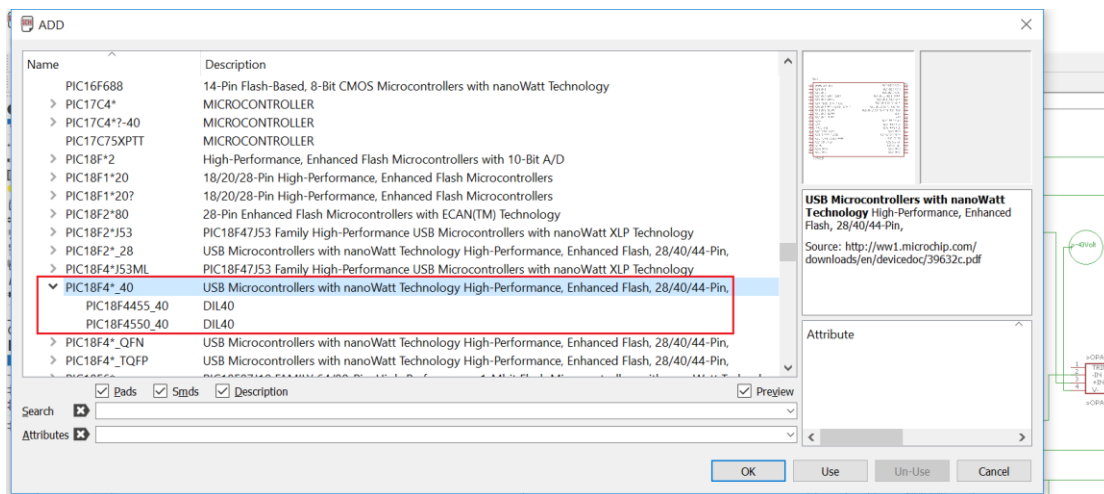
Σχήμα 0-2 Το Menu του schematic editor.

Ανοίγουμε το menu Library->Use... Εδώ θα ανοίξει ένα παράθυρο στο φάκελο με τις βιβλιοθήκες που είναι διαθέσιμες στο πρόγραμμα. Δήνουμε Cntrl+A για να επιλεγούν όλα τα αρχεία και επιλέγουμε άνοιγμα. Τώρα έχουν φορτωθεί όλες τις διαθέσιμες βιβλιοθήκες. Τα εργαλεία που θα μας χρειαστούν για τον σχεδιασμό φαίνονται με κόκκινο στο παρακάτω σχήμα.



Σχήμα 0-3 Τα εργαλεία Add, Wire, και Circle, που θα μας χρειαστούν στο σχεδιασμό.

Στο σχήμα 10-2 φαίνονται με κόκκινους κύκλους τα εργαλεία Add, Wire και Zoom που είναι τα εργαλεία που θα χρησιμοποιήσουμε για το σχεδιασμό. Για να βάλουμε ένα ηλεκτρονικό εξάρτημα μέσα στο σχέδιό μας, πρέπει να το βρούμε σε κάποια βιβλιοθήκη του προγράμματος (αν δεν βρεθεί θα πρέπει να το σχεδιάσουμε μόνοι μας όπως θα παρουσιαστεί παρακάτω). Πρώτα θα τοποθετηθεί ο μικροελεγκτής. Πατώντας Add από την εργαλειοθήκη αριστερά. Πλοηγούμαστε προς τα κάτω στο menu που εμφανίζεται μέχρι να βρεθεί η εταιρία Microchip (είναι η εταιρία που κατασκευάζει τους PIC) Χρησιμοποιώντας το βελάκι για την επέκταση του menu, εντοπίζεται η οικογένεια μικροελεγκτών PIC18F4*. Το menu που μόλις περιγράφηκε φαίνεται στο παρακάτω σχήμα.



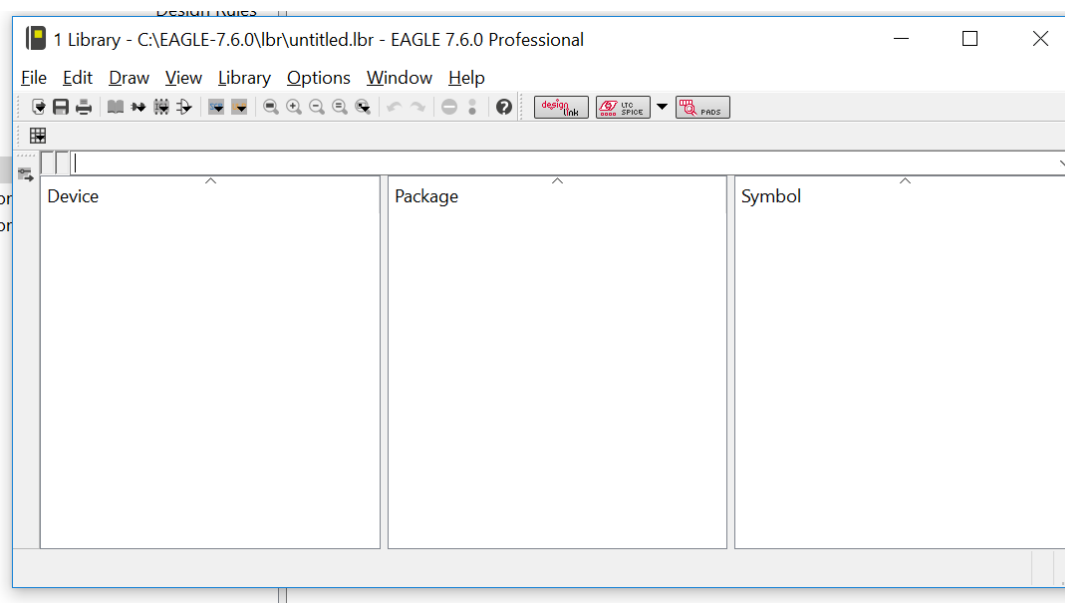
Σχήμα 0-4 Εισαγωγή του component PIC18F4550 στο Eagle 7.6.0

Γενικώς τα component που χρειάζονται μπορούν να βρεθούν και από την επιλογή Search, όμως πρέπει να είμαστε προσεκτικοί ώστε να δίνονται γενικές πληροφορίες στο search. Για παράδειγμα αν δοθεί PIC18F4550 στο Search δεν το πρόγραμμα δεν θα επιστρέψει κανένα αποτέλεσμα, παρόλο που ο μικροελεγκτής υπάρχει στις βιβλιοθήκες. Απαιτείται λίγος χρόνος για την εξοικείωση με την εύρεση εξαρτημάτων στις βιβλιοθήκες του προγράμματος. Για παράδειγμα τα στοιχεία HD44780LCD (είναι της Hitachi) και 7805T.

υπάρχουν στις προσφερόμενες βιβλιοθήκες. Τα υπόλοιπα στοιχεία του κυκλώματος που σχεδιάσαμε σε αυτή την εργασία δεν υπάρχουν στις βιβλιοθήκες του Eagle.

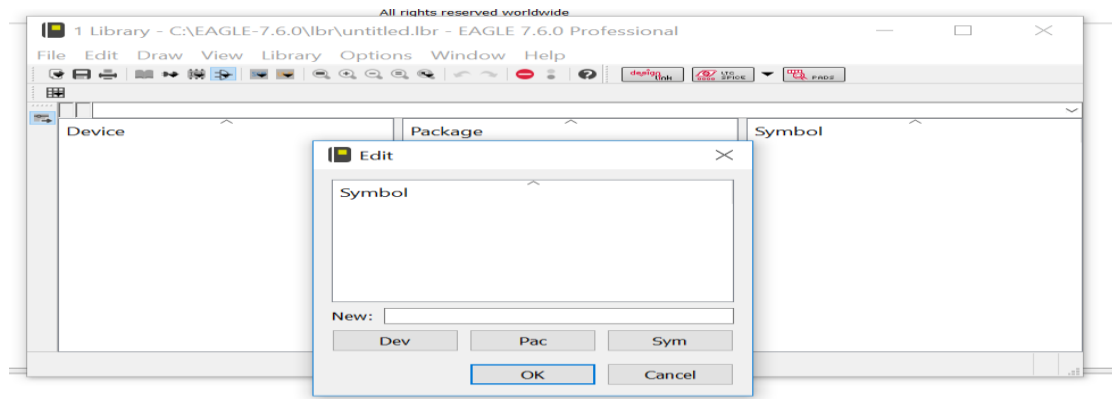
Γενικώς μπορεί κανείς να βρει custom βιβλιοθήκες στο internet, που είναι φτιαγμένες από άλλους χρήστες, για διάφορα ηλεκτρονικά εξαρτήματα. Επειδή όμως τα στοιχεία που βάζουμε στο κύκλωμά μας πρέπει να έχουν συγκεκριμένες διαστάσεις, όπως αυτές αναφέρονται στα φύλλα δεδομένων τους, είναι καλύτερα να μάθει κανείς να φτιάχνει τις βιβλιοθήκες μόνος του γιατί έτσι μπορεί να είναι σίγουρος ότι οι διαστάσεις που έχουν τα εξαρτήματα που εισάγονται στο σχέδιό είναι οι σωστές. Αυτό είναι ιδιαίτερα χρήσιμο αν θέλουμε το Eagle να μας σχεδιάσει το PCB του κυκλώματός μας. Αν στο PCB οι διαστάσεις δεν είναι σωστές, τα στοιχεία δεν θα μπαίνουν στην τελική πλακέτα που θα σχεδιαστεί. Επιπλέον πολλά ηλεκτρονικά εξαρτήματα δεν περιγράφονται σε βιβλιοθήκες και έτσι θα πρέπει αναγκαστικά να φτιάξουμε τις δικές μας.

Μια βιβλιοθήκη στο Eagle μπορεί να περιέχει πολλά εξαρτήματα. Εδώ θα ξεκινήσουμε φτιάχνοντας μια καινούρια βιβλιοθήκη, και μέσα σε αυτή θα φτιάξουμε τα εξαρτήματα που μας χρειάζονται. Από το αρχικό menu του Eagle 7.6.0 μεταβαίνουμε στο New-> Library. Θα εμφανιστεί το παρακάτω παράθυρο.



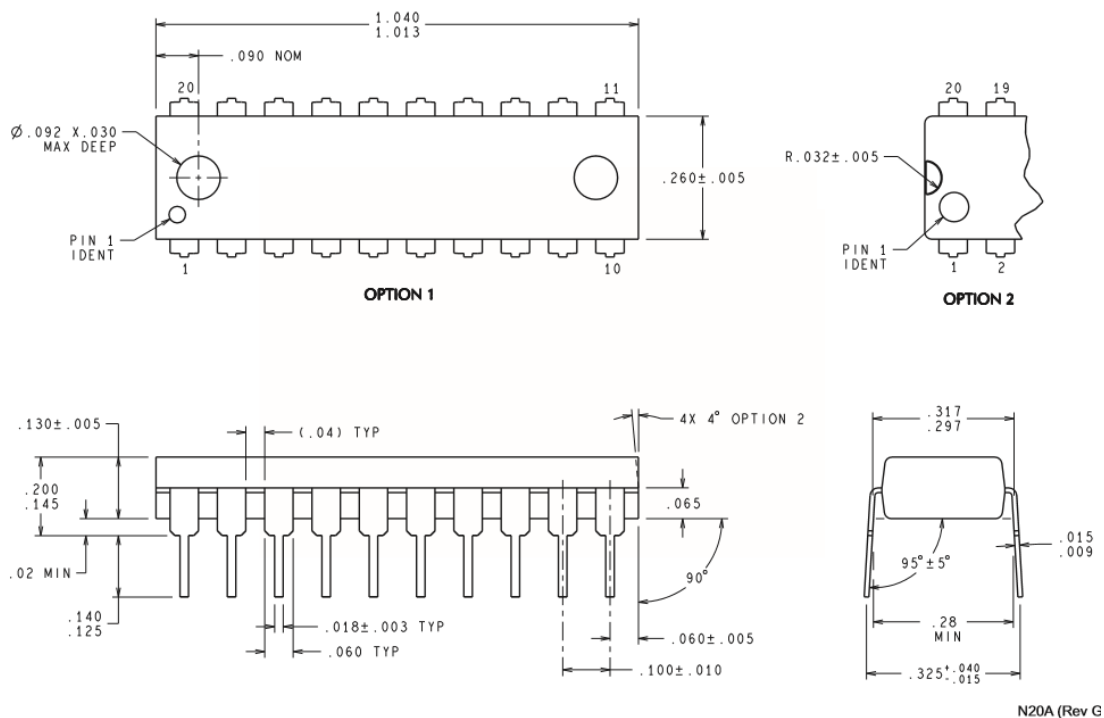
Σχήμα 0-5 Δημιουργία νέα βιβλιοθήκης στο Eagle 7.6.0.

Εμφανίζονται τώρα τα τρία παράθυρα Device Package και Symbol τα οποία αρχικά θα πρέπει να είναι άδεια. Στην εργαλειοθήκη που υπάρχει στο πάνω μέρος του παραθύρου πατάμε το κουμπί Symbol. Θα εμφανιστεί το menu που φαίνεται στο παρακάτω σχήμα. Δίνουμε ένα όνομα στο νέο σύμβολο που θα δημιουργηθεί, εδώ το ονομάζουμε DAC0832BILLS. Χρειάζεται ένα περιγραφικό όνομα για το σύμβολό ώστε να είναι ξεκάθαρο σε τι αντιστοιχεί όταν θα εμφανίζεται στις βιβλιοθήκες του. Συνήθης τακτική είναι να χρησιμοποιείται το όνομα του ηλεκτρονικού εξαρτήματος όπως **ακριβώς** αναφέρεται στα φύλλα δεδομένων του. Το Eagle μας αναγκάζει να γράφουμε με κεφαλαία γράμματα τα ονόματα που δίνουμε. Αν οι παρακάτω οδηγίες δεν ακολουθηθούν ακριβώς με τη σειρά που αναφέρονται, θα αποτύχει η δημιουργία ενός σωστού εξαρτήματος.



Σχήμα 0-6 Δημιουργία Νέου Συμβόλου

Σε αυτό το παράδειγμα θα κατασκευαστεί το DAC0832. Οι πληροφορίες για τις διαστάσεις του παρέχονται στα φύλλα δεδομένων και φαίνονται παρακάτω στο σχήμα.

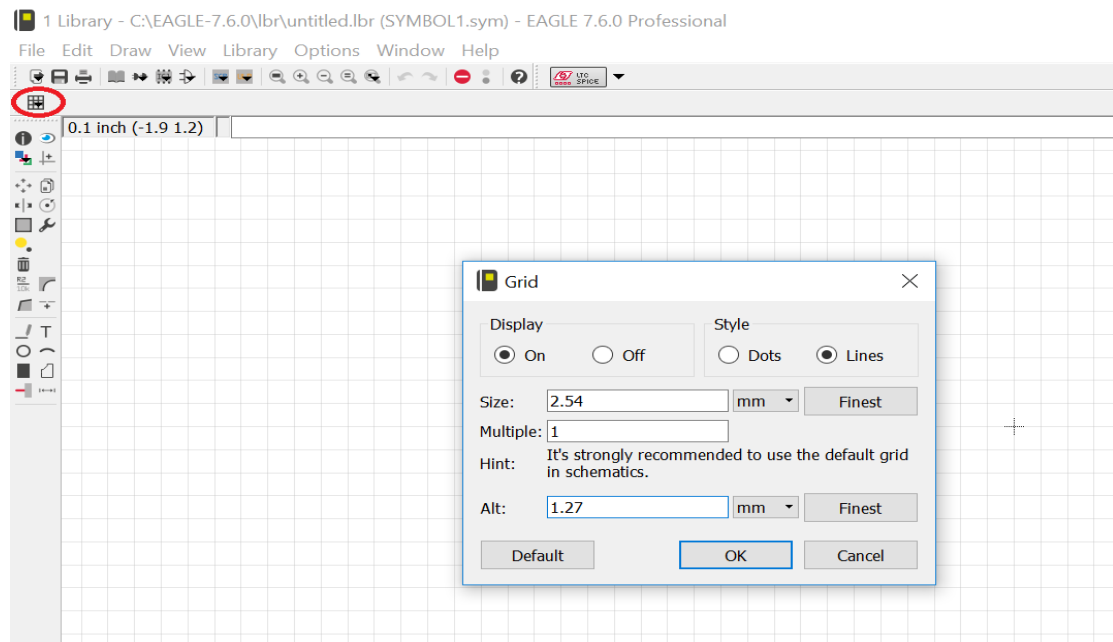


N20A (Rev G)

Σχήμα 0-7 Οι διαστάσεις του DAC0832 όπως παρέχονται στα φύλλα δεδομένων του.

Εδώ πρέπει να αναφερθεί ότι οι διαστάσεις δίνονται σε ίντσες και όχι σε χιλιοστά όπως ισχυρίζονται τα φύλλα δεδομένων του. Οι μονάδες στις οποίες δίνονται οι διαστάσεις διαφέρουν στα διάφορα εξαρτήματα, μπορεί κανείς να συμβουλευτεί τα φύλλα δεδομένων για πληροφορίες σχετικά με τις διαστάσεις, οι πληροφορίες αναφορικά με τις μονάδες πρέπει να διασταυρωθούν, με ένα χάρακα πάνω στο εξάρτημα, ώστε να είναι κανείς σίγουρος ότι έχει να κάνει με χιλιοστά ή με ίντσες. Το πιο σημαντικό στοιχείο εδώ είναι η απόσταση ανάμεσα σε δύο ποδαράκια στο εξάρτημα. Εδώ η απόσταση αναφέρεται ως 0.1 ίντσες, δηλαδή 0.254cm. Αυτή η απόσταση θα είναι ίδια σε όλα τα εξαρτήματα που θα χρησιμοποιηθούν τα οποία δίνονται σε συσκευασία DIP. Επίσης μας ενδιαφέρει το πάχος ενός PIN, για να είμαστε σίγουροι ότι θα μπαίνει στις τρύπες που θα γίνουν στην PCB μας.

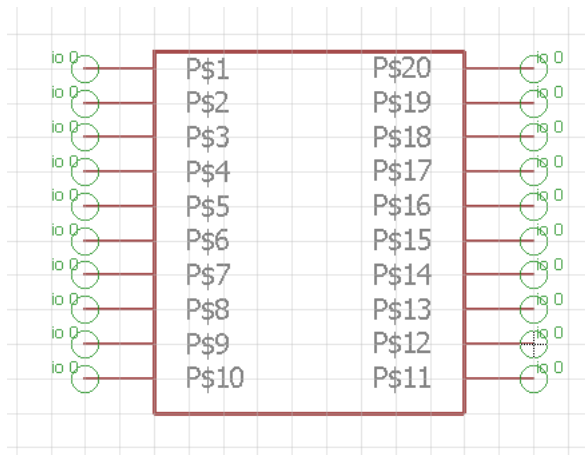
Έχοντας δώσει όνομα στο Symbol μας στο σχήμα 10-6 μεταφερόμαστε στο παρακάτω παράθυρο σχεδιασμού.



Σχήμα 0-8 Το παράθυρο σχεδιασμού νέου συμβόλου και το menu Grid.

Εδώ έχουμε πατήσει στο εικονίδιο πάνω αριστερά που λέγεται Grid (είναι τονισμένο με κόκκινο). Δίνουμε διαστάσεις στο Grid όπως φαίνονται στο menu. Έτσι έχουμε 2.54 mm για το μέγεθος του τετραγώνου στο Grid. Όταν πάμε να σχεδιάσουμε μια γραμμή στο Grid το πρόγραμμα μας επιτρέπει να ξεκινάμε και να τελειώνουμε τη γραμμή μόνο πάνω στους κόμβους του Grid. Στο πεδίο Alt πρέπει να δοθεί 1.27 mm (το μισό) για την εναλλακτική απόσταση (Alt), σε αυτή την απόσταση έχουμε πρόσβαση όταν κρατάμε πατημένο το Alt στο πληκτρολόγιο, οπότε μπορούμε με το Alt πατημένο να ξεκινήσουμε ή να τελειώσουμε μια γραμμή ανάμεσα στους κόμβους του πλέγματος.

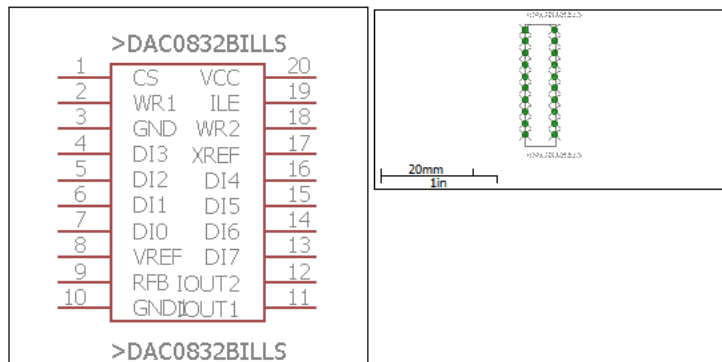
Έχοντας φτιάξει σωστά το Grid μπορούμε να αρχίσουμε το σχεδιασμό. Πρώτα τοποθετούνται τα PINs του εξαρτήματός μας, με τον κύκλο προς τα έξω. Φροντίζουμε κάθε PIN να έχει απόσταση από το άλλο 2,54mm χρησιμοποιώντας το Grid. Η απόσταση ανάμεσα σε αντιδιαμετρικά PINs δεν μας απασχολεί (μπορεί να είναι όση θέλουμε ώστε να έχουμε σωστό οπτικό αποτέλεσμα). Τέλος με το εργαλείο Wire αριστερά στην παλέτα δημιουργείται το περίγραμμα του εξαρτήματος. Στο περίγραμμα δεν χρειάζεται να είμαστε ακριβείς. Παρακάτω φαίνεται το σύμβολο που φτιάχτηκε. Αφήσαμε αρκετή απόσταση ανάμεσα στα αντιδιαμετρικά PINs, επίσης τα ονόματα των PINs είναι σημαντικά. Το DAC0832 έχει 20 PINs, έχοντας κατασκευάσει το παρακάτω σχήμα είναι ώρα να δώσουμε ονόματα στα PINs.



Σχήμα 0-9 Κατασκευή του συμβόλου για το DAC0832.

Για να ονομάσουμε τα PINS χρησιμοποιούμε το εργαλείο Name κλικάρωντας στο πράσινο κύκλο στο πρώτο PIN, θα εμφανιστεί ένα παράθυρο όπου εισάγεται το όνομα. Συμβουλευόμενοι τα ονόματα από τα φύλλα δεδομένων, προσέχουμε το όνομα να αντιστοιχεί στον αντίστοιχο αριθμό του PIN. Το ίδιο γίνεται για όλα τα PINS του εξαρτήματος, το τελειωμένο Symbol φαίνεται παρακάτω.

DAC0832BILLS



Σχήμα 0-10 Το DAC0832 όπως φαίνεται στη βιβλιοθήκη μου στο πρόγραμμα Eagle 7.6.0.

Στο πάνω σχήμα φαίνεται το τελικό Symbol, Η διαδικασία δεν έχει τελειώσει ακόμη. Πατώντας το button Text και γράφουμε ">DAC0832BILLS" δίνουμε OK και επιλέγουμε αριστερά πάνω όπου φαίνονται τα Layers το Layer Names, έπειτα κάνουμε κλικ κάπου πάνω από το στοιχείο που σχεδιάστηκε όπου θέλουμε να φαίνεται το όνομά του. Έπειτα κάνουμε κλικ πάλι στο button Text και γράφουμε πάλι το ">DAC0832BILLS" δίνουμε OK και επιλέγουμε αριστερά πάνω στα Layers το Layer Values. Με το Layer Values επιλεγμένο κάνουμε κλικ κάπου κάτω από το αντικείμενο που σχεδιάστηκε όπου θέλουμε να φαίνεται το Value του αντικειμένου. Δείτε και το σχήμα 10-10 για καθοδήγηση. Φροντίζουμε το όνομα που δώσαμε στο Layer Names και Values να είναι το ίδιο και ίδιο με το όνομα που δώσαμε για το Symbol. Το σύμβολό μας είναι έτοιμο. Τώρα πρέπει να κατασκευάσουμε το PACKAGE

Για την κατασκευή του PACKAGE πατάμε πάνω αριστερά το Button Package. Δήνομε στο Package το ίδιο όνομα που δώσαμε και στο Symbol. Τροποποιούμε το Grid όπως και παραπάνω για το Symbol. Χρησιμοποιούμε το εργαλείο της γραμμής πρώτα για να κατασκευάσουμε το περίβλημα του στοιχείου φροντίζοντας να έχει τις ακριβείς διαστάσεις όπως αναφέρονται στα φύλλα δεδομένων του. Τώρα στις θέσεις των PINs που βάλαμε στο Symbol θα βάλουμε τις τρύπες που θα γίνουν στο PCB. Αριστερά στην παλέτα κλικάρουμε στο Button Draw Pet (πράσινο εικονίδιο). Στο menu πάνω ανοίγουν τα Options για τις διαστάσεις της τρύπας (Drill). Ορίζουμε τις διαστάσεις σύμφωνα με τα φύλλα δεδομένων του εξαρτήματος. Το Diameter είναι η διάμετρος του κύκλου που θα σχεδιαστεί, αυτό επηρεάζει μόνο το οπτικό αποτέλεσμα, αφήνουμε εδώ τη προεπιλεγμένη διάσταση. Ξεκινώντας από το σημείο που βάλαμε το πρώτο PIN (PIN 1) στο Symbol, επιλέγουμε ένα τετράγωνο Pet και το βάζουμε στο σχήμα. Συνεχίζουμε βάζοντας τις υπόλοιπες τρύπες με στρόγγυλο Pet. Μπορεί κανείς να συμβουλευτεί και το τελικό σχέδιο στο παρακάτω σχήμα.

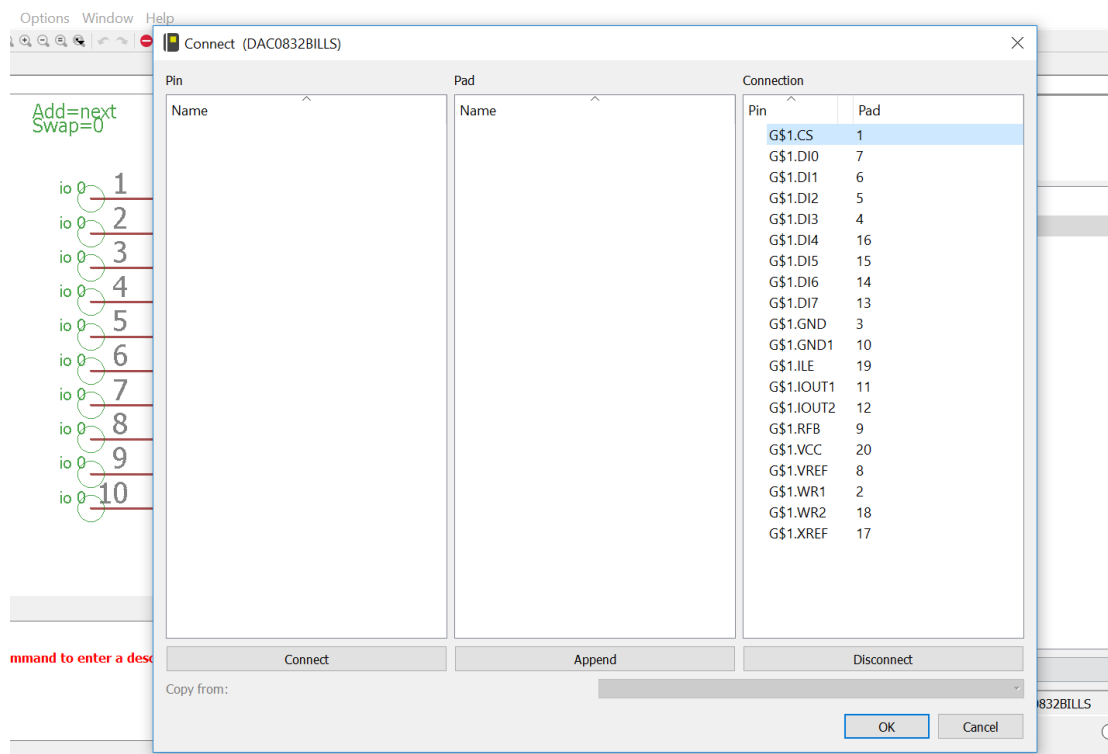


Σχήμα 0-11 Το Package για το DAC0832 στο Eagle 7.6.0.

Έχοντας σχεδιάσει το εξάρτημα χρησιμοποιούμε το εργαλείο Name (βρίσκετε στην αριστερή εργαλειοθήκη) για να αλλάξουμε τα ονόματα στα Pet. Εργαζόμαστε με τον ίδιο τρόπο όπως και για το Symbol. Τέλος κάνουμε κλικ στο button Text και δίνουμε ">DAC0832BILLS" και αφού επιλέγουμε πάνω αριστερά το Layer tNames τοποθετούμε το κείμενο πάνω από το εξάρτημα. Κάνουμε πάλι κλικ στο Button Text και δίνουμε "DAC0832BILLS", επιλέγουμε το layer tValues και τοποθετούμε το κείμενο κάτω από το εξάρτημα. Μπορεί κανείς να συμβουλευτεί και το σχήμα 10-11. Έτσι ολοκληρώνεται η κατασκευή του PACKAGE.

Έχοντας ολοκληρώσει τα Symbol και Package απομένει ένα τελευταίο βήμα για την ολοκλήρωση της βιβλιοθήκης. Κάνουμε κλικ στο button Device στην εργαλειοθήκη κάτω από το βασικό μενού (File). Δίνουμε το ίδιο όνομα με παραπάνω "DAC0832BILLS". Στο παράθυρο που εμφανίζεται επιλέγουμε αριστερά το button Add. Από το παράθυρο που εμφανίζεται επιλέγουμε το Symbol που έχουμε φτιάξει. Πρέπει να είναι και το μοναδικό που εμφανίζεται. Τοποθετούμε το Symbol κάπου στη μέση του χώρου σχεδίασης. Έπειτα πατώντας New δεξιά κάτω επιλέγουμε το Package που μόλις φτιάξαμε. Αυτό θα εμφανιστεί

σε ένα παράθυρο δεξιά πάνω. Τώρα πατάμε Connect, που βρίσκεται δεξιά κάτω. Θα εμφανιστεί το παρακάτω menu.



Σχήμα 0-12 Αντιστοίχιση των PINs στο Device που κατασκευάσαμε.

Εδώ πρέπει να αντιστοιχηθούν τα PINs του Symbol με το Package. Επιλέγουμε τα αντίστοιχα PINs στα παράθυρα Pin και Pad και πατάμε το Connect για κάθε PIN οπότε θα δούμε την αντιστοίχιση στο δεξιό παράθυρο "Connection". Προσέχουμε να αντιστοιχηθούν σωστά τα PINs. Πατάμε OK και θα δούμε ένα πράσινο tic στο παράθυρο δεξιά. Αυτό σημαίνει ότι η αντιστοίχιση όλων των PINs έχει γίνει. Τέλος πατάμε το button Prefix (δεξιά κάτω) και δίνουμε πάλι το όνομα "DAC0832BILLS". Πατάμε Save από το βασικό menu (πάνω) δίνουμε πάλι το ίδιο όνομα. Τώρα η βιβλιοθήκη είναι έτοιμη για χρήση. Μπορούμε να προσθέσουμε και άλλα εξαρτήματα στην ίδια βιβλιοθήκη. Προσέχουμε τα ονόματα που δίνονται στις βιβλιοθήκες και τα εξαρτήματά που φτιάχνουμε να είναι περιγραφικά ώστε να μπορούμε να τα βρούμε αργότερα ανάμεσα στις πολλές βιβλιοθήκες που παρέχει το Eagle 7.6.0.

Έχοντας φτιάξει τις βιβλιοθήκες για τα Component DAC0832 MAX232 και OPA445AP τα προσθέσαμε στο κύκλωμά μας και φτιάξαμε το τελικό σχέδιο του σχήματος 4-1. Αν θέλουμε τώρα να δούμε το PCB απλώς πατάμε File->Switch to Board το Eagle αναλαμβάνει να μας φτιάξει το PCB το οποίο μπορούμε να σώσουμε σαν αρχείο με κατάληξη .bdr. Με το PCB αυτής της εργασίας δεν θα ασχοληθούμε διότι είναι έξω από τα πλαίσια του θέματος.

Σχήμα 0-13 Το αρχικό PCB όπως το σχεδίασε το Eagle 7.6.0.

Η διαδικασία που περιγράφηκε παραπάνω πρέπει να ακολουθηθεί κατά γράμμα για να έχουμε το σωστό τελικό αποτέλεσμα τις βιβλιοθήκες μας. Δυστυχώς δεν υπάρχει ποιο εύκολος τρόπος για να κατασκευάσουμε βιβλιοθήκες στο συγκεκριμένο πρόγραμμα και πολλές από τις έτοιμες βιβλιοθήκες φτιαγμένες από χρήστες που κυκλοφορούν στο internet δεν είναι αξιόπιστες κυρίως γιατί οι διαστάσεις των εξαρτημάτων δεν αποδίδονται σωστά. Την πρώτη φορά θα πρέπει να αφιερώσει κανείς αρκετό χρόνο για την κατασκευή μιας βιβλιοθήκης όμως αποκτώντας εμπειρία, ο χρήστης θα προτίμα να κατασκευάζει μόνος τις βιβλιοθήκες που είναι απαραίτητες.

Παράρτημα 2 Υπολογισμός κόστους υλικών.

Εδώ θα παρουσιαστούν τα υλικά που χρησιμοποιήθηκαν για την υλοποίηση όλου του κυκλώματος της εργασίας και θα συμπεριληφθούν οι τιμές τους. Προφανώς οι τιμές που θα αναφερθούν εδώ δεν είναι οπωσδήποτε ίδιες σε όλους τους προμηθευτές. Εδώ χρησιμοποιήθηκε για όλα τα εξαρτήματα ένας συγκεκριμένος προμηθευτής. Γενικώς όμως οι τιμές στις οποίες μπορούν να βρεθούν τα εξαρτήματα που αναφέρονται παρακάτω θα είναι κοντά στις δικές μας τιμές.

Στο παρακάτω πίνακα φαίνονται οι τιμές των εξαρτημάτων και τα ονόματά τους. Δεν περιλαμβάνονται οι τιμές για τις αντιστάσεις και τους μικρούς πυκνωτές (1mF) γιατί θεωρήθηκαν αμελητέες. Οι παρακάτω τιμές περιλαμβάνουν ΦΠΑ.

Εξάρτημα	Ποσότητα	Τιμή Euros
Τροφοδοτικό		
2x35V/200mA Μετασχηματιστής	x1	7
2200μF//63V	x2	1.7
Γέφυρα B250-C1500GD	x1	0.15
Πλακέτα διάτριτη RE-318HP	x1	5.1
Zener 43Volt 5Watt	x2	0.9
Ψηφιακός Έλεγχος		
Πλακέτα δοκιμών (breadboard)	x1	27
PIC18F4550 i/p	x1	9.3
100Ohm trimmer	x1	1.1
7805 Voltage Stabilizer	x1	0.25
7820 Voltage Stabilizer		0.5
20MHz Crystal	x1	0.25
RC-1602B-YHW-ESV Display LCD 2x16	x1	8.5
OPA445AP Op Amp DIP 8	x2	34
MAX232CPE	x1	2.8
DAC0832	x1	5
Καλώδιο USB to Serial A plug D Sub 9pin male	x1	12
DS-9S Θηλυκός D Sub Connector	x1	0.15
9Volt Μπαταρία	x1	2.2
Σύνολο		117,9
Εργαλεία Προγραμματισμού Microchip		
PIC KIT 3	x1	50

Ευρετήριο Σχημάτων και Πινάκων

Σχήμα 1-1 Πηγή ρεύματος με τρανζίστορ και διαιρέτη τάσης.....	6
Σχήμα 1-2 Αμφιπολική πηγή ρεύματος.....	7
Σχήμα 1-3 Πηγή ρεύματος με αναφορά την τάση zener	7
Σχήμα 1-4 Πηγή ρεύματος με χρήση καθρέπτη	8
Σχήμα 1-5 Πηγή ρεύματος Widlar.....	8
Σχήμα 1-6 Πηγή ρεύματος Wilson	9
Σχήμα 1-7 Πηγή ρεύματος Howland.....	9
Σχήμα 2-1 Τροφοδοτικό εργασίας	11
Σχήμα 2-2 Γέφυρα ανόρθωσης.....	11
Σχήμα 2-3 Φίλτρο Πυκνωτή.....	12
Σχήμα 2-4 Σταθεροποίηση με zener	13
Σχήμα 3-1 Αντλία ρεύματος Howland.....	15
Σχήμα 3-2 Improved Howland	17
Σχήμα 3-3 Το περιβάλλον του Multisim 13.0	20
Σχήμα 3-4 Τροφοδοτικό εργασίας σχεδιασμένο στο Multisim13	21
Σχήμα 3-5 Σχεδίαση τροφοδοτικού στο Multisim 13.....	21
Σχήμα 3-6 Το κύκλωμα improved Howland που θα χρησιμοποιήσουμε	22
Σχήμα 3-7 Κύκλωμα τροφοδοσίας και improved Howland. Παρατηρήστε το φορτίο 1800Ohm στην έξοδο του Howland και το πολύμετρο που μετρά συνεχές ρεύμα.....	23
Σχήμα 3-8 Γραφική ρεύματος εξόδου σε συνάρτηση με την αντίσταση φορτίου για ρεύμα 1mA	25
Σχήμα 3-9 Γραφική ρεύματος εξόδου σε συνάρτηση με την αντίσταση φορτίου για ρεύμα 10mA	26
Σχήμα 3-10 Γραφική ρεύματος εξόδου σε συνάρτηση με την αντίσταση φορτίου για ρεύμα 15mA	27
Σχήμα 3-11 Γραφική ρεύματος εξόδου σε συνάρτηση με την αντίσταση φορτίου για ρεύμα 20mA	28
Σχήμα 3-12 Γραφική ρεύματος εξόδου σε συνάρτηση με την αντίσταση φορτίου για ρεύμα 30mA	29
Σχήμα 3-13 Γραφική ρεύματος εξόδου σε συνάρτηση με την αντίσταση φορτίου για ρεύμα 20mA και τάση τροφοδοσίας του τελεστικού τα 100Volt. (Προσοχή! Δεν μπορεί να εφαρμοστεί στην πράξη.).....	30
Σχήμα 3-14 Improved Howland με τον τελεστικό LM741	31
Σχήμα 3-15 Γραφική ρεύματος εξόδου σε συνάρτηση με την αντίσταση φορτίου για ρεύμα 1mA, και τελεστικό τον LM741.....	32
Σχήμα 3-16 Γραφική ρεύματος εξόδου σε συνάρτηση με την αντίσταση φορτίου για ρεύμα 10mA, και τελεστικό τον LM741.....	33
Σχήμα 3-17 Γραφική ρεύματος εξόδου σε συνάρτηση με την αντίσταση φορτίου για ρεύμα 15mA, και τελεστικό τον LM741.....	34
Σχήμα 3-18 Γραφική ρεύματος εξόδου σε συνάρτηση με την αντίσταση φορτίου για ρεύμα 20mA, και τελεστικό τον LM741.....	35
Σχήμα 3-19 Κύκλωμα improved Howland με ασύμμετρη τροφοδοσία +80Volt.....	36

Σχήμα 3-20 Προσομοίωση με 80Volt Ασύμμετρη τροφοδοσία και 43Volt συμμετρική με τον OPA445AP.....	37
Σχήμα 3-21 Bootstrapping Improved Howland Current Pump.....	38
Σχήμα 3-22 Σχέση μεταξύ τάσης εισόδου και ρεύματος εξόδου για φορτίο 10kOhm.	38
Σχήμα 4-1 Κύκλωμα ψηφιακού ελέγχου της τάσης εισόδου στο κύκλωμα improved Howland	39
Σχήμα 4-2 Ο μικροελεγκτής PIC18F4550 της Microchip.	41
Σχήμα 4-3 Το PIC KIT 3.....	45
Σχήμα 4-4 Compiler select in MPLABX IDE	46
Σχήμα 4-5 Δημιουργία νέου Project στο MPLABX IDE.....	46
Σχήμα 4-6 Το περιβάλλον του MPLABX IDE	47
Σχήμα 4-7 Όψη του μενού Projects	47
Σχήμα 4-8 Αντιστοίχιση των PINS του PIC KIT 3 με αυτά του μικροελεγκτή μας.	48
Σχήμα 4-9 Σωστός τρόπος σύνδεσης του PIC KIT 3.....	49
Σχήμα 5-1 Η LCD οθόνη 2 σειρών 16 χαρακτήρων της Raystar.....	50
Σχήμα 5-2 Το κύκλωμα για τον φωτισμό της LCD της Raystar.	51
Σχήμα 5-3 Η σύνδεση της LCD 2 σειρών 16 χαρακτήρων με τον PIC48F4550.	52
Σχήμα 6-1 Hardware για επικοινωνία του PIC18F4550 με το MAX232.	60
Σχήμα 6-2 Σωστή εγκατάσταση του καλωδίου USB to Serial.....	65
Σχήμα 6-3 Το αρχικό menu του LabView 2011.....	66
Σχήμα 6-4 Το user interface του LabView 2011.....	66
Σχήμα 6-5 Το τελικό περιβάλλον επικοινωνίας του χρήστη με τον PIC μέσω σειριακής θύρας (Front Panel LabView 2011).....	67
Σχήμα 6-6 Το τελικό Block Diagram της εργασίας στο LabView 2011	68
Σχήμα 7-1 Hardware υπεύθυνο για την επικοινωνία του PIC18F4550 με το DAC0832.	73
Σχήμα 7-2 PINS του DAC0832 στη συσκευασία PDIP.....	73
Σχήμα 7-3 Μετατροπή του ρεύματος εξόδου του DAC0832 σε τάση.	78
Σχήμα 8-1 Διάγραμμα ροής του κώδικα με τον οποίο προγραμματίσαμε τον PIC18F4550. .	79
Σχήμα 9-1 Χαρακτηριστικές ρεύματος εξόδου σε σχέση με το βήμα στο DAC για 100, 467 και 1000 Ohm.	86
Σχήμα 9-2 Χαρακτηριστική ρεύματος στην έξοδο σε σχέση με το βήμα στο DAC για 1563 2390 και 4740 Ohm φορτίο.....	89
Σχήμα 10-1 Το αρχικό interface του Eagle 7.6.0.....	91
Σχήμα 10-2 Το Menu του schematic editor.....	91
Σχήμα 10-3 Τα εργαλεία Add, Wire, και Circle, που θα μας χρειαστούν στο σχεδιασμό.	92
Σχήμα 10-4 Εισαγωγή του component PIC18F4550 στο Eagle 7.6.0	92
Σχήμα 10-5 Δημιουργία νέα βιβλιοθήκης στο Eagle 7.6.0.....	93
Σχήμα 10-6 Δημιουργία Νέου Συμβόλου	94
Σχήμα 10-7 Οι διαστάσεις του DAC0832 όπως παρέχονται στα φύλλα δεδομένων του.....	94
Σχήμα 10-8 Το παράθυρο σχεδιασμού νέου συμβόλου και το menu Grid.	95
Σχήμα 10-9 Κατασκευή του συμβόλου για το DAC0832.....	96
Σχήμα 10-10 Το DAC0832 όπως φαίνεται στη βιβλιοθήκη μου στο πρόγραμμα Eagle 7.6.0.96	
Σχήμα 10-11 Το Package για το DAC0832 στο Eagle 7.6.0.	97
Σχήμα 10-12 Αντιστοίχιση των PINS στο Device που κατασκευάσαμε.....	98
Σχήμα 10-13 Το αρχικό PCB όπως το σχεδίασε το Eagle 7.6.0.....	98

Πίνακας 3-1 Ρεύμα εξόδου σε συνάρτηση με το φορτίο για ρεύμα εξόδου 1mA.....	24
Πίνακας 3-2 Ρεύμα εξόδου σε συνάρτηση με το φορτίο για ρεύμα εξόδου 10mA.....	25
Πίνακας 3-3 Ρεύμα εξόδου σε συνάρτηση με το φορτίο για ρεύμα εξόδου 15mA.....	26
Πίνακας 3-4 Ρεύμα εξόδου σε συνάρτηση με την αντίσταση φορτίου για ρεύμα εξόδου 20mA.	27
Πίνακας 3-5 Ρεύμα εξόδου σε συνάρτηση με την αντίσταση φορτίου για ρεύμα εξόδου 30mA.	28
Πίνακας 3-6 Ρεύμα εξόδου(20mA) σε συνάρτηση με την αντίσταση φορτίου για τάση τροφοδοσίας του τελεστικού τα 100Volt (δεν γίνεται στην πράξη).....	29
Πίνακας 3-7 Ρεύμα εξόδου(1mA) σε συνάρτηση με το φορτίο για τον τελεστικό LM741....	31
Πίνακας 3-8 Ρεύμα εξόδου(10mA) σε συνάρτηση με το φορτίο για τον τελεστικό LM741...	32
Πίνακας 3-9 Ρεύμα εξόδου(10mA) σε συνάρτηση με το φορτίο για τον τελεστικό LM741...	33
Πίνακας 3-10 Ρεύμα εξόδου(20mA) σε συνάρτηση με το φορτίο για τον τελεστικό LM741.	34
Πίνακας 3-11 Ρεύμα εξόδου(20mA) σε συνάρτηση με το φορτίο για ασύμμετρη τροφοδοσία του OPA445AP.	36
Πίνακας 4-1 Τα PIN του PIC18F4550 που χρησιμοποιούμε.	42
Πίνακας 5-1 Τα PINS της LCD Raystar και η λειτουργία τους.....	50
Πίνακας 5-2 Προτεινόμενες συναρτήσεις ελέγχου LCD της microchip	53
Πίνακας 5-3 Τιμές του control byte για τον έλεγχο μερικών λειτουργιών απεικόνισης της Raystar.....	55
Πίνακας 5-4 Τιμές του control byte για την αρχικοποίηση της LCD της Raystar.	56
Πίνακας 6-1 Οι προσφερόμενες από την Microchip συναρτήσεις επικοινωνίας με τη σειριακή του C18 compiler.....	61
Πίνακας 6-2 Οι defined τιμές των bit της μεταβλητής config (πρώτη παράμετρος της OpenUSART()).	62
Πίνακας 6-3 Ο πίνακας βαθμονόμησης της πηγής ρεύματός μας. Έγινε με αντίσταση φορτίου 4670hm.....	69
Πίνακας 9-1 Χαρακτηριστικές ρεύματος εξόδου σε σχέση με το βήμα στο DAC, για 100, 467 και 1000 Ohm.	85
Πίνακας 9-2 mA στην έξοδο και τάση στην έξοδο σε σχέση με το βήμα στο DAC για αντίσταση 15630hm.....	86
Πίνακας 9-3 mA στην έξοδο και τάση στην έξοδο σε σχέση με το βήμα στο DAC για αντίσταση 23900hm.....	87
Πίνακας 9-4 mA στην έξοδο και τάση στην έξοδο σε σχέση με το βήμα στο DAC για αντίσταση 47400hm.....	88

Αριθμός λέξεων κειμένου 24.266