

Τομέας: Ηλεκτρολογίας – Ηλεκτρονικής

Εκπαιδευτικός: Μπουλαδάκης Στέλιος
Κασάμπαλης Στέλιος

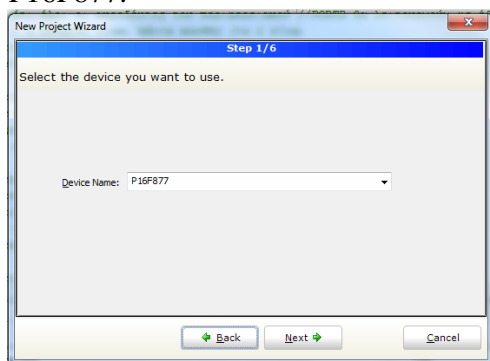
Μάθημα: Μηχατρονική

Αντικείμενο: Οδηγίες Προγραμματισμού μικροελεγκτή PIC με γλώσσα mikroC

Τι θα χρησιμοποιήσουμε:

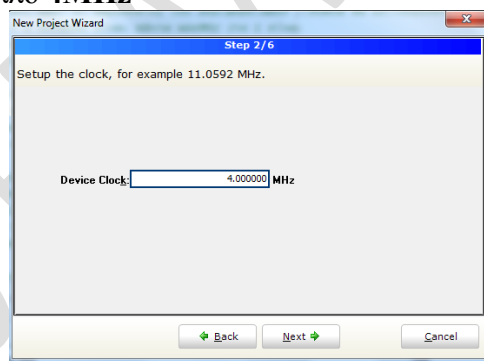
- Γλώσσα προγραμματισμού mikroC Pro

Βήμα 1^ο: Φορτώνουμε το ολοκληρωμένο προγραμματιστικό περιβάλλον mikroC μέσα από τη διαδρομή : **Project / New Project**. Στο πρώτο βήμα-παράθυρο του οδηγού δημιουργίας νέου έργου, επιλέγουμε τον τύπο του μικροελεγκτή, που στο παράδειγμά μας είναι ο P16F877.



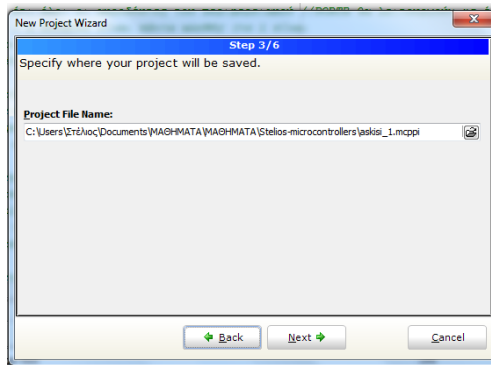
Εικόνα 1: Το παράθυρο επιλογής του μικροελεγκτή

Βήμα 2^ο: Ενημερώνουμε για τη συχνότητα λειτουργίας του κρυστάλλου που βρίσκεται στο κύκλωμα χρονισμού του μικροελεγκτή, όπως φαίνεται στην Εικόνα 2. Στην περίπτωσή μας θα ορίσουμε συχνότητα **4MHz** επειδή η πλακέτα δοκιμών **SE1001** έχει κρύσταλλο **4MHz**



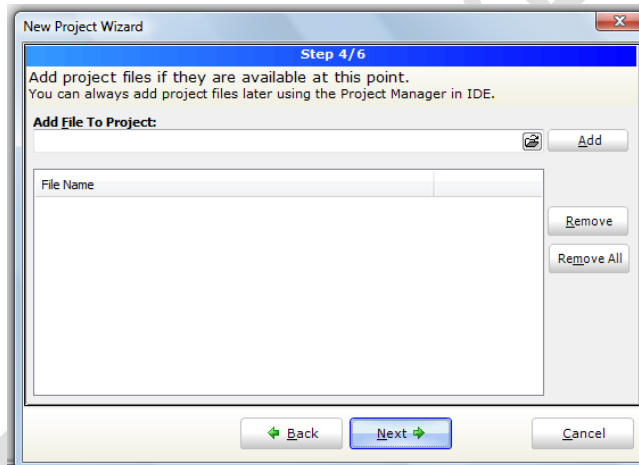
Εικόνα 2: Η επιλογή της συχνότητας χρονισμού του μικροελεγκτή

Βήμα 3^ο: Καθορίζουμε το φάκελο αποθήκευσης και το όνομα του αρχείου έργου-project της εφαρμογής μας. Το IDE αποδίδει την προέκταση **.mcpri** όπως φαίνεται και στην Εικόνα 3.**ΠΡΟΣΟΧΗ:** για κάθε άσκηση (έργο-project) συστήνουμε να δημιουργείτε ένας καινούριος φάκελος και μέσα σε αυτόν να αποθηκεύετε το αρχείο **.mcpri** ώστε να μην μπερδεύονται με αρχεία άλλης άσκησης. Έτσι στον φάκελο αυτό θα περιέχονται όλα τα αρχεία που παράγει το προγραμματιστικό περιβάλλον της mikroC. Ανάμεσα σε αυτά θα είναι το αρχείο με προέκταση **.c** που έχει το *source file* σε κώδικα και το αρχείο **.hex** το οποίο θα μεταφέρουμε στο MPLAB για τον προγραμματισμό του μικροελεγκτή



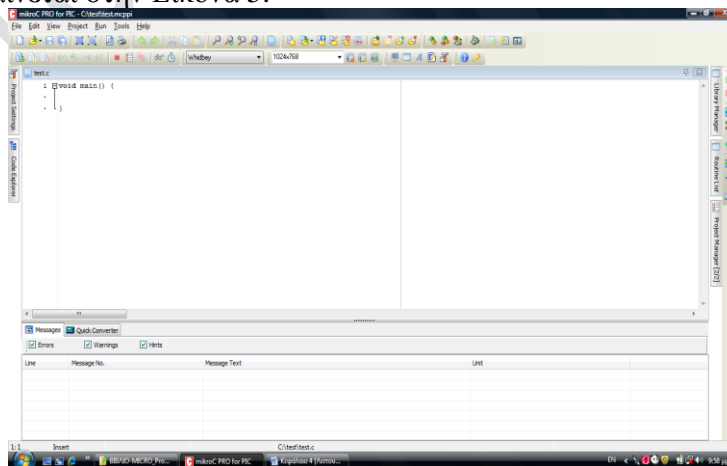
Εικόνα 3: Καθορισμός διαδρομής φακέλου και αρχείου project.

Βήμα 4^ο: Κάθε φορά που δημιουργείται ένα καινούριο project, το IDE αυτόματα δημιουργεί ένα αρχείο κώδικα με όνομα ίδιο με εκείνο του project αλλά με προέκταση **.c**. Στο επόμενο βήμα, μας προτρέπει να δηλώσουμε εάν επιθυμούμε να δηλώσουμε επιπλέον αρχεία κώδικα .c που θέλουμε να ενσωματωθούν στο αρχείο έργου (project). Σε όλες τις εφαρμογές που υπάρχουν σε αυτό το βιβλίο αφήνουμε κενό το αντίστοιχο πεδίο καθώς δεν χρησιμοποιούμε επιπλέον αρχεία και επιλέγουμε **Next**.



Εικόνα 4: Το βήμα εισαγωγής περισσότερων αρχείων κώδικα

Βήμα 5^ο: Στα επόμενα δύο παράθρα πατάμε αντίστοιχα **Next** και **Finish** για να εμφανισθεί μετά από λίγο το περιβάλλον εργασίας της γλώσσας mikroC PRO for PIC που φαίνεται στην Εικόνα 5.




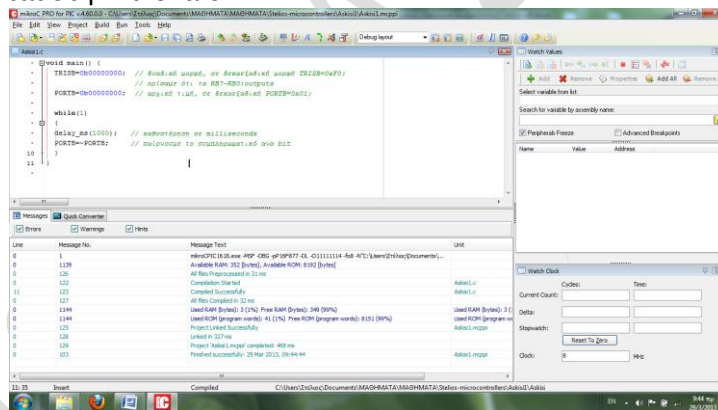
Εικόνα 5: Το περιβάλλον εργασίας για την συγγραφή κώδικα στο IDE mikroC PRO for PIC.

Βήμα 6^ο: Στο παράθυρο συγγραφής κώδικα της Εικόνας 5 γράφουμε τον παρακάτω κώδικα όπου στα σχόλια (//) δίνονται επεξηγήσεις των εντολών (προσοχή κάθε εκτελέσιμη αυτοτελής εντολή κλείνει με το σύμβολο ;)

```
void main()  
{  
    TRISB=0b00000000; // αγκύλη έναρξης εντολών προγράμμ.  
                    // ορίζουμε ότι όλοι οι ακροδέκτες  
                    // RB7-RB0:outpruts  
                    // σε δεκαεξαδική μορφή: TRISB=0xF0 ;  
    PORTB=0b00000000; //αρχική τιμή ακροδεκτών μηδέν ώστε  
                    // να είναι όλοισβηστοί  
    while(1) // δομή επανάληψης όσο η συνθήκη  
            // είναι αληθής (το 1 πάντα 1)  
    {  
        delay_ms(1000); // αγκύλη έναρξης εντολών επανάληψης  
                        // εντολή καθυστέρησης σε ms  
        PORTB=~PORTB; // εναλλαγή των τιμών Bit 0<-->1  
    } // τέλος εντολών επανάληψης  
} // τέλος εντολών προγράμματος
```

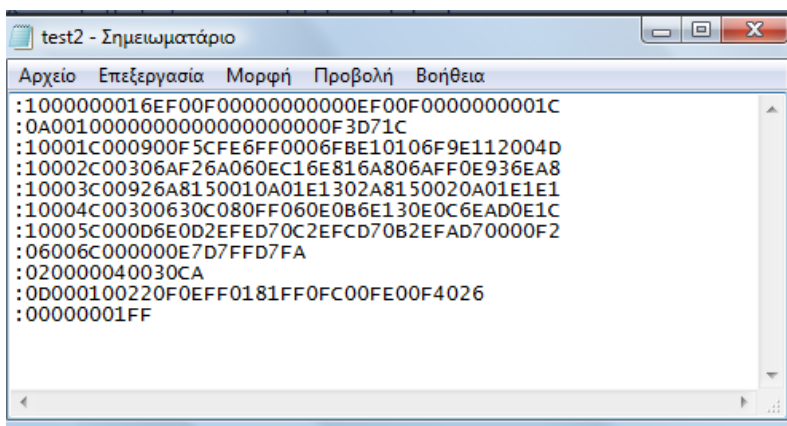
Βήμα 7^ο: Κάνουμε συμβολομετάφραση του κώδικα είτε μέσω της επιλογής

Project/Build είτε πατώντας **Ctrl+F9** είτε κάνοντας κλικ στο εικονίδιο  που βρίσκεται στη γραμμή εργαλείων. Εφόσον δεν υπάρχουν συντακτικά λάθη το περιβάλλον εργασίας και το παράθυρο μηνυμάτων θα είναι περίπου όπως αυτά που φαίνονται στην Εικόνα 6.



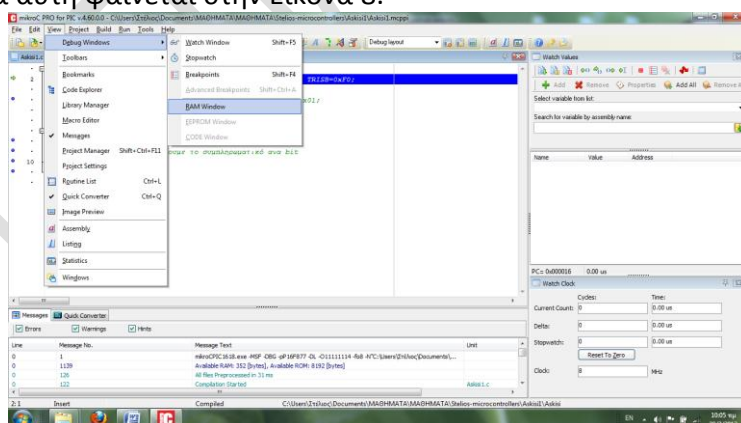
Εικόνα 6: Το περιβάλλον εργασίας μετά τη συμβολομετάφραση.

Βήμα 8^ο: Εκτελούμε τη συμβολομετάφραση του κώδικα ώστε αν χρειαστεί να διορθώσουμε τυχόν λάθη (μέσω της επιλογής **Project/Build**). Η διαδικασία αυτή δημιουργεί ένα αρχείο σε γλώσσα μηχανής στον ίδιο φάκελο με τα υπόλοιπα αρχεία της εργασίας μας. Το αρχείο αυτό έχει προέκταση **.hex** και τα περιεχόμενά του μπορούν να διερευνηθούν με τη βοήθεια του ενσωματωμένου editor των Windows, του **NotePad**, όπως φαίνεται στην Εικόνα 7. Είναι εκείνο το αρχείο που θα κατεβάσουμε στο μικροελεγκτή μέσα από τη διαδικασία προγραμματισμού του επόμενου βήματος.



Εικόνα 7: Το αρχείο .hex που προέκυψε από τη συμβολομετάφραση του κώδικα. Αυτό είναι το αρχείο με το οποίο θα προγραμματιστεί ο μικροελεγκτής.

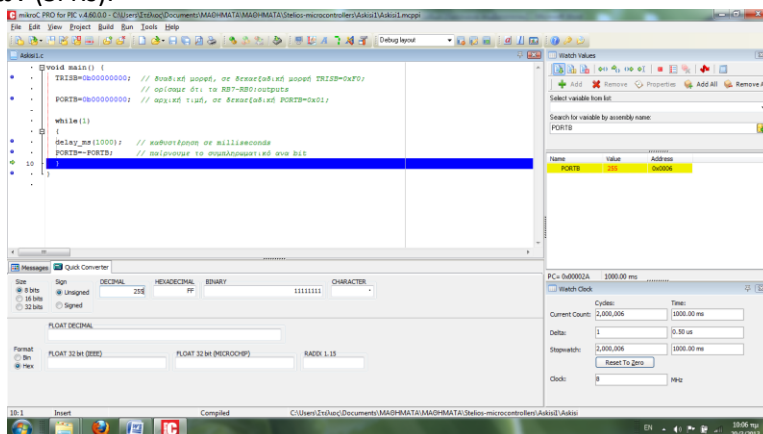
Βήμα 9ο: Ενεργοποιούμε τον εκσφαλματωτή (debugger) για να παρακολουθήσουμε σε ένα στάδιο δοκιμών τις τιμές που παίρνουν οι διάφοροι καταχωρητές κατά την εκτέλεση του κώδικα. Η ενεργοποίησή του γίνεται επιλέγοντας διαδοχικά **Run/Start Debugger** οπότε το παράθυρο της εφαρμογής μας γίνεται όπως αυτό της Εικόνας 7. Στο νέο παράθυρο **Watch Values** από το κυλιόμενο μενού **Select Variable from list** επιλέγουμε τον καταχωρητή **PORTB** και κάνουμε κλικ στο πλήκτρο **Add**. Σε αυτό το παράθυρο βλέπουμε όλους τους καταχωρητές ειδικών λειτουργιών που σχετίζονται με το μικροελεγκτή που έχουμε επιλέξει. Για να μπορέσουμε να δούμε επιπλέον και τις τιμές γενικών μεταβλητών (π.χ μια μεταβλητή *i* που θέλουμε να αποθηκεύεται στον καταχωρητή της δεκαεξαδικής διεύθυνσης 30 και η οποία δηλώνεται με την εντολή `unsigned short i absolute 0x30;`) πρέπει να ενεργοποιήσουμε το παράθυρο μνήμης **RAM**. Στη μνήμη RAM απεικονίζονται όλοι οι καταχωρητές (και οι γενικού σκοπού π.χ. κάποια μεταβλητή όπως η *i* και οι ειδικού σκοπού όπως η PORTB). Η ενεργοποίηση του παραθύρου RAM γίνεται επιλέγοντας διαδοχικά **View/Debug Windows/RAM** και εφόσον έχει προηγηθεί η ενεργοποίηση του debugger. Η διαδικασία αυτή φαίνεται στην Εικόνα 8.



Εικόνα 8: Η διαδικασία ενεργοποίησης του παραθύρου RAM.

Στη συνέχεια πατάμε το πλήκτρο **F7** ή κάνουμε διαδοχικά κλικ στο πλήκτρο **Step Into** όπως φαίνεται στην Εικόνα 9 και παρακολουθούμε την επίδραση στα περιεχόμενα των μεταβλητών *i* που βρίσκεται στη διεύθυνση 0x30 και PORTB που βρίσκεται στη διεύθυνση 0x0F81. Την τελευταία μπορούμε να τη δούμε επίσης στο παράθυρο RAM εάν προχωρήσουμε την μπάρα απεικόνισης προς τα κάτω και βρούμε τη διεύθυνση 0x0F81. Η δομή της μνήμης RAM μας δείχνει την απεικόνιση

και των καταχωρητών γενικού σκοπού (GPRs) και των καταχωρητών ειδικών λειτουργιών (SFRs).



Εικόνα 9: Η παρακολούθηση αλλαγής των τιμών των μεταβλητών κατά την εκσφαλμάτωση του προγράμματος

Η διαδικασία τερματίζεται κάνοντας κλικ στο πλήκτρο **Stop debugger**. Ιδιαίτερα χρήσιμη είναι η καρτέλα **Quick Converter** το κάτω μέρος του προγραμματιστικού περιβάλλοντος όπου οι μαθητές μπορούν να εξοικειωθούν με τα διάφορα αριθμητικά συστήματα και να ελέγχουν τις τιμές των καταχωρητών που φαίνονται στα παράθυρα **Watch Values** του debugger

Προγραμματισμός του Μικροελεγκτή:

Ακολουθούμε τις οδηγίες του φύλλου Εργου :

- ‘Μηχατρ_1_Οδηγίες προγραμματισμού μικροελεγκτών στο σχολείο’

Δοκιμές της εφαρμογής μας : αφαιρούμε το μικροελεγκτή από τη βάση της προγραμματίστριας PIC και τον τοποθετούμε στην πλακέτα SE1001