

# ΜΑΘΗΜΑ : ΕΡΓΑΣΤΗΡΙΟ ΨΗΦΙΑΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ

## ΚΑΘΗΓΗΤΗΣ : ΜΠΟΥΛΤΑΔΑΚΗΣ ΣΤΕΛΙΟΣ

### 1. Η γλώσσα προγραμματισμού mikroC

Η εταιρεία Mikroelektronika ([www.mikroe.com](http://www.mikroe.com)) παρέχει την δυνατότητα ανάπτυξης κώδικα με διάφορες γλώσσες ανώτερου επιπέδου όπως η C, Pascal και Basic. Έτσι, προσφέρει το προγραμματιστικό περιβάλλον και το μεταγλωττιστή **mikroC PRO for PIC** που συνδυάζει τις δυνατότητες δομημένου προγραμματισμού της γλώσσας C με τις αναφορές απευθείας στα ονόματα των καταχωρητών και των περιφερειακών πολλών μικροελεγκτών PIC.

Η εγκατάσταση του ολοκληρωμένου περιβάλλοντος προγραμματισμού (IDE) περιλαμβάνει το συντάκτη, το μεταγλωττιστή, τις εσωτερικές βιβλιοθήκες της γλώσσας προγραμματισμού, τον προσομοιωτή ενώ ενσωματώνει και την πλακέτα προγραμματισμού. Την πλακέτα προγραμματισμού μπορούμε να μη τη χρησιμοποιούμε εάν διαθέτουμε εξοπλισμό προγραμματισμού του μικροελεγκτή άλλης εταιρείας όπως για παράδειγμα το εξοπλισμό PICStart Plus που υπάρχει στα σχολεία.

Η διαδικασία προγραμματισμού του μικροελεγκτή περιλαμβάνει τα παρακάτω τρία βήματα:

- Βήμα 1<sup>ο</sup>: στο περιβάλλον της MikroC PRO μπορούμε να αναπτύξουμε τον κώδικα σε γλώσσα mikroC,
- Βήμα 2<sup>ο</sup>: στη συνέχεια με τη συμβολομετάφραση του κώδικα παίρνουμε το αρχείο .hex και
- Βήμα 3<sup>ο</sup>: προγραμματίζουμε τον μικροελεγκτή με την πλακέτα και το εργαλείο προγραμματισμού της ίδιας ή μιας άλλης εταιρείας.

Στην περίπτωση που χρησιμοποιούμε τον προγραμματιστή PICStart Plus της MICROCHIP και δεν επιθυμούμε την ανάπτυξη κώδικα με γλώσσα assembly, τότε μπορούμε να εκτελέσουμε το Βήμα 1ο και το Βήμα 2ο της παραπάνω διαδικασίας και στη συνέχεια:

- Βήμα 3<sup>ο</sup>: μέσω του IDE **MPLAB** της MICROCHIP από τη διαδρομή **File/Import / Import to Memory** να φορτώσουμε το αρχείο .hex στο host H/Y.
- Βήμα 4<sup>ο</sup>: Στη συνέχεια καθορίζουμε τα **Configuration bits** μέσα από το μενού **Configure** του MPLAB, σύμφωνα με ότι έχει επιλεγθεί στο project της mikroC και τέλος στο
- Βήμα 5<sup>ο</sup>: η μεταφορά του αρχείου .hex γίνεται με την επιλογή του προγραμματιστή μέσα από τη διαδρομή **Programmer/Select Programmer/ PicStart Plus** και τις επιλογές **Enable Programmer** και **Program**.

Στην περίπτωση του λειτουργικού συστήματος Windows Vista, τόσο το λογισμικό προγραμματισμού όσο και το λογισμικό του IDE απαιτούν την εκτέλεσή τους ως Διαχειριστής ώστε να μην εμφανίζονται προβλήματα στο στάδιο προγραμματισμού, στην αποθήκευση και στη συμβολομετάφραση των αρχείων.

Για τις ασκήσεις που απαιτούνται μετατροπείς USB σε Serial απαιτείται:

- πρώτα η εγκατάσταση του οδηγού που υπάρχει στο συνοδευτικό CD και ο οποίος πρέπει να είναι συμβατός με το λειτουργικό σύστημα που έχει ο Η/Υ (συνήθως με την εκτέλεση ενός αρχείου setup.exe που να είναι συμβατό με το λειτουργικό σύστημα που χρησιμοποιούμε)
- στη συνέχεια επανεκκίνηση του Η/Υ και
- τέλος, σύνδεση του μετατροπέα στον Η/Υ (συνήθως μετά την επανεκκίνησή του) ώστε να είναι δυνατή η σωστή ανίχνευσή του από τον Η/Υ, η εγκατάσταση και λειτουργία του.

Η σωστή εγκατάσταση όλων των οδηγών που προαναφέραμε μπορεί να ελεγχθεί στη διαδρομή **H/Yμου/δεξί κλικ Ιδιότητες/ Διαχείριση συσκευών**. Με την ίδια επιλογή μπορούμε να απεγκαταστήσουμε ή να ενημερώσουμε τα προγράμματα οδήγησης υλικού του Η/Υ μας. Εδώ βλέπουμε και τους μετατροπέες **USB to Serial** που έχουν εγκατασταθεί ως επιπλέον θύρες **Com** για σειριακή επικοινωνία.

Ο κώδικας σε γλώσσα μικροC περιέχει μεταβλητές, σταθερές, δομές δεδομένων ενώ η ανάπτυξή του γίνεται σύμφωνα με τις αρχές του δομημένου προγραμματισμού. Ακολουθούμε δηλαδή τις γνωστές δομές ακολουθίας, επιλογής και επανάληψης. Το πλεονέκτημα από τη χρήση του compiler της μικροC είναι ότι εκτός από τις μεταβλητές μπορούμε να αποδίδουμε τιμές απευθείας στους καταχωρητές των μικροελεγκτών χρησιμοποιώντας απλά τα ονόματά τους.

Μια τυπική δομή κώδικα είναι η παρακάτω:

```
// ΤΜΗΜΑ ΔΗΛΩΣΗΣ (declaration) ΜΕΤΑΒΛΗΤΩΝ και ΣΥΝΑΡΤΗΣΕΩΝ

unsigned1 short1 mask(unsigned2 short2 num);
unsigned short i;
unsigned short i absolute 0x30;

// ΚΥΡΙΩΣ ΠΡΟΓΡΑΜΜΑ

void main() { // αρχή της συνάρτησης main
.....
.....
.....
.....
} // τέλος συνάρτησης main
```

η έναρξη του τμήματος με το κύριο πρόγραμμα γίνεται με τη δήλωση της συνάρτησης **main**, η οποία δεν έχει ορίσματα εισόδου ανάμεσα στις παρενθέσεις και ταυτόχρονα δεν επιστρέφει κάποια παράμετρο. Το τελευταίο ορίζεται από τη δήλωση **void** μπροστά από το όνομα της συνάρτησης.

```
// ΤΜΗΜΑ ΟΡΙΣΜΟΥ (definition) ΣΥΝΑΡΤΗΣΕΩΝ

unsigned short mask(unsigned short num) {
.....
.....
.....
return μεταβλητή ή τιμή ;
.....
.....
return μεταβλητή ή τιμή ;

} // τέλος συνάρτησης mask
```

ΠΑΡΑΔΕΙΓΜΑΤΑ ΔΟΜΩΝ ΕΠΙΛΟΓΗΣ

<pre> <b>if</b> (x==100)     PORTB=0xFE;         </pre>	<pre> <b>if</b> (x==100)     PORTB=0xFE; <b>else</b>     PORTB=0x42;         </pre>
<pre> <b>if</b> (x==100) {     PORTB=0xFE; //ομάδα εντολών     PORTA=0xF2; }         </pre>	<pre> <b>if</b> (x==100) {     PORTB=0xFE; //ομάδα εντολών     PORTA=0xF2; } <b>else</b> {     PORTB=0xEE; //ομάδα εντολών     PORTA=0x22; }         </pre>
<pre> <b>if</b> (x&gt;0)     PORTB=0x48; <b>else if</b> (x&lt;0)     PORTB=0x38; <b>else</b>     PORTD=0x45;         </pre>	<pre> <b>if</b> (x&gt;0) {     PORTB=0x48;     PORTD=0x33; } <b>else if</b> (x&lt;0) {     PORTB=0x38;     PORTD=0x45; } <b>else</b>     PORTD=0x45;         </pre>
<pre> <b>switch</b> (n) { <b>Case 1:</b> // n=1     PORTD=0x38; //ομάδα εντολών <b>break;</b> <b>Case 2:</b> // n=2     PORTD=0x48; //ομάδα εντολών <b>break;</b> <b>default :</b>     PORTD=0x2A; //ομάδα εντολών }         </pre>	<pre> <b>Switch</b> (num) { <b>Case 0:</b> return 0x3F; <b>Case 1:</b> return 0xFF; }         </pre>

ΠΑΡΑΔΕΙΓΜΑΤΑ ΔΟΜΩΝ ΕΠΑΝΑΛΗΨΗΣ

<pre>for (i=0; i&lt;10; i++) PORTB=0x35;</pre>	<p>Μία εντολή και γνωστό πλήθος επαναλήψεων</p>
<pre>for (i=0;, i&lt;10;, i++) {   PORTB=0x35;   PORTD=0x42; }</pre>	<p>Ομάδα εντολών και γνωστό πλήθος επαναλήψεων</p>
<pre>i=0; while (i&lt;10)   i=i+1;</pre>	<p>Έλεγχος συνθήκης στην αρχή και άγνωστο πλήθος επαναλήψεων.</p>
<pre>i=0; while (i&lt;10) {   PORTB=0x35;   PORTD=0x42;   i=i+1; }</pre>	<p>Η επανάληψη γίνεται όσο η συνθήκη είναι αληθής.          Η επανάληψη τερματίζεται όταν η συνθήκη γίνει ψευδής.          Στη δομή της While δεν μπαίνει ερωτηματικό</p>
<pre>while (5) {   ..... }</pre>	<p>Ατέρμων βρόχος επειδή το 5 είναι πάντα 5</p>
<pre>i=0; do {   PORTB=0x00;   delay_ms(1000);   PORTB=0xFF;   delay_ms(1000);   i=i+1; } while (i&lt;=4);</pre>	<p>Η επανάληψη τερματίζεται όταν η συνθήκη είναι αληθής.          Η επανάληψη γίνεται όσο η συνθήκη είναι ψευδής.  <b>ΠΡΟΣΟΧΗ:</b> στη δομή της do....while μπαίνει ερωτηματικό στο τέλος της συνθήκης</p>