

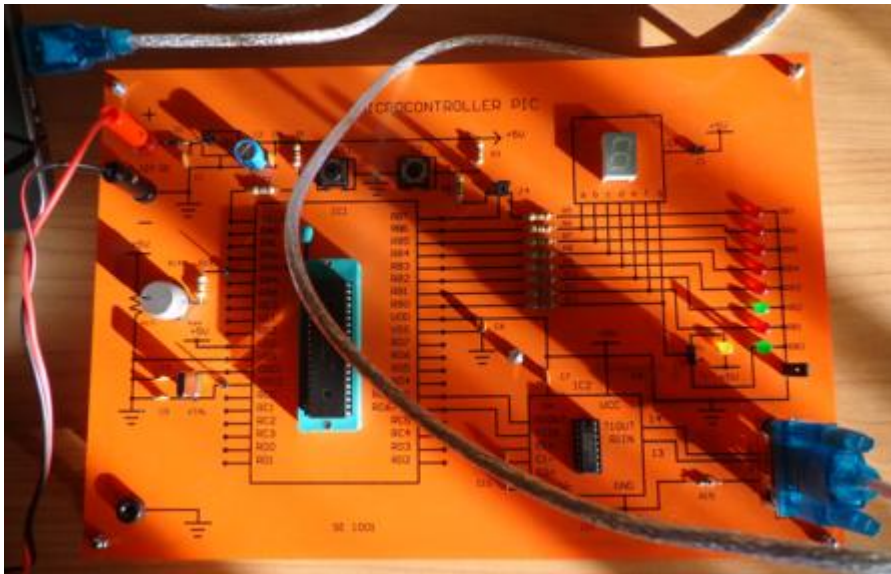
**Μάθημα:** Εργαστήριο Ψηφιακών Ηλεκτρονικών Συστημάτων

**Αντικείμενο:** Ασύγχρονη Σειριακή Επικοινωνία του PIC16F877 (PIC18F452) με προσωπικό Η/Υ  
*Αντιστοιχεί στην ΑΣΚΗΣΗ 5η: Ασύγχρονη Σειριακή Επικοινωνία με έναν προσωπικό Η/Υ*

Υλικά που απαιτούνται:

- Η εκπαιδευτική πινακίδα SE1001
- Ένας μ/κ PIC16F877 ή PIC18F452
- Τροφοδοτικό 9-12 Volts
- Εγκατεστημένη γλώσσα προγραμματισμού mikroC για να γράψουμε τον κώδικα
- Προγραμματίστρια MICROCHIP PicStart Plus με εγκατεστημένο το περιβάλλον MPLAB (συνήθως με σειριακό καλώδιο με ακροδέκτη DB9) **ή** οποιαδήποτε άλλη USB προγραμματίστρια για μ/κ PIC με εγκατεστημένο το αντίστοιχο περιβάλλον προγραμματισμού που να είναι συμβατό με το λειτουργικό σύστημα του Η/Υ που χρησιμοποιούμε. Στην άσκηση αυτή θα χρησιμοποιήσουμε την **USB PIC Programmer** με στοιχεία **K149-BC** της **DIY Electronics** που φαίνεται στην φωτογραφία και εγκαθιστούμε τον αντίστοιχο driver για το λειτουργικό σύστημα. Για τη μεταφορά του κώδικα hex εγκαθιστούμε το λογισμικό **MicroPro** που τρέχει σε λειτουργικό σύστημα έως την έκδοση Windows 7 των 32 bits.
- Στην εκπαιδευτική πινακίδα SE1001 ο μικροελεγκτής PIC17F877 (PIC18F452) συνδέεται σε συνδετήρα DB9 ώστε να είναι δυνατή η ασύγχρονη σειριακή επικοινωνία του (USART) με προσωπικό Η/Υ. Αυτό γίνεται μέσω των ακροδεκτών:
  - ο **RC7/RX** : με τον οποίο λαμβάνει ο PIC ASCII χαρακτήρες από τον προσωπικό Η/Υ
  - ο **RC6/TX** : με τον οποίο στέλνει ο PIC ASCII χαρακτήρες στον προσωπικό Η/Υ.
- Στην άσκηση αυτή θα χρησιμοποιήσουμε το παράθυρο USART που είναι ενσωματωμένο στο περιβάλλον της γλώσσας προγραμματισμού mikroC ( ή εναλλακτικά την εφαρμογή HyperTerminal των Windows XP ή μπορούμε να κατεβάσουμε αντίστοιχη εφαρμογή HyperTerminal για Windows 7) και το αντικείμενο της άσκησης θα είναι το παρακάτω:
- Όπως φαίνεται στην παρακάτω φωτογραφία, στο σχήμα που υπάρχει στο εργαστηριακό βιβλίο αλλά και στο συνοδευτικό εγχειρίδιο της πινακίδας SE1001 ανάμεσα στους ακροδέκτες του PIC και στο συνδετήρα DB9 παρεμβάλλεται το IC MAX232 το οποίο μετατρέπει τις τάσεις του '0' (0 Volts) και '1' (+5 Volts) με τις οποίες λειτουργεί ο PIC σε αντίστοιχες τάσεις της σειριακής επικοινωνίας '0' (+10 Volts) και '1' (-12 Volts). Στην ίδια φωτογραφία φαίνεται ότι ο συνδετήρας DB9 της πινακίδας SE1001 συνδέεται στις θύρες USB ενός σύγχρονου προσωπικό Η/Υ (που δεν διαθέτει συνδετήρα σειριακής επικοινωνίας DB9) με μετατροπέα USB to serial για τον οποίο φροντίσαμε κατά την αγορά του από τα καταστήματα Η/Υ και περιφερειακών να συνοδεύεται από CD οδηγών που να είναι συμβατοί με λειτουργικό σύστημα WINDOWS 7 (32 και 64 bits). Κατά την πρώτη σύνδεση του μετατροπέα USB to serial επιλέξαμε εγκατάσταση οδηγών από τον Η/Υ μου (CD) και όχι ενημέρωση μέσω διαδικτύου. Όπως έχει περιγραφεί και σε προηγούμενα φύλλα έργου θα πρέπει μέσω της Διαχείρισης Συσκευών/Θύρες Com & LPT να

βρούμε τον αριθμό της ομάδας καταχωρητών COMn στους οποίους τα Windows έχουν αντιστοιχίσει το μετατροπέα USB to Serial



Αντικείμενο της άσκησης:

Όταν ο προσωπικός Η/Υ στέλνει στον PIC το χαρακτήρα 'm' ο PIC θα απαντά με τη γραμματοσειρά : 'HELLO'

- Όταν ο προσωπικός Η/Υ στέλνει στον PIC το χαρακτήρα 'a' ο PIC θα απαντά με τη γραμματοσειρά : 'WORLD'
- Όταν ο προσωπικός Η/Υ στέλνει στον PIC το χαρακτήρα 'c' τότε ο PIC θα ενεργοποιεί την μετατροπή αναλογικού σε ψηφιακό και θα αποστέλλει την τιμή των ψηφιακών επιπέδων (0-255) στον Η/Υ.

Απαιτούμενες γνώσεις για την άσκηση:

Ο PIC16F877(PIC18F452) διαθέτει το περιφερειακό USART για να στέλνει και να λαμβάνει δεδομένα με συγκεκριμένο ρυθμό (baud). Στη συγκεκριμένη εφαρμογή δεν θα χρησιμοποιήσουμε τη μέθοδο διακοπών και επομένως οι καταχωρητές που σχετίζονται με τη λειτουργία του περιφερειακού USART είναι οι παρακάτω:

**TRISC :** Επειδή ο ακροδέκτης 26 θα χρησιμοποιηθεί ως είσοδος για την σειριακή επικοινωνία θα πρέπει το αντίστοιχο bit του καταχωρητή διεύθυνσης της θύρας **TRISC** να τεθεί σε κατάσταση 1. Έτσι για την περίπτωση των ακροδεκτών RX και TX, ο καταχωρητής **TRISC** θα πρέπει να προγραμματιστεί με τα περιεχόμενα του παρακάτω πίνακα.

**TRISC : PortC Data Direction Register**

b7-RX IN	b6-TX OUT	b5	b4	b3	b2	b1	b0	τιμή
1	0	0	0	0	0	0	0	0x80

**TXSTA :** Οι ρυθμίσεις για τη διαδικασία της σειριακής εκπομπής ορίζονται μέσω του ειδικού καταχωρητή ελέγχου **TXSTA**. Τα περιεχόμενα του καταχωρητή αυτού φαίνονται στον παρακάτω πίνακα (για ασύγχρονη επικοινωνία: CSRC=0 και SYNC=0, κωδικοποίηση με 8 δυαδικά ψηφία (TX9=0), ενεργοποίηση της εκπομπής (TXEN=1), επιλογή υψηλής ταχύτητας 9600 bps (BRGH=1), χωρίς bit ισοτιμίας (TX9D=0) .

**TXSTA : Transmit Status and Control Register**

b7	b6	b5	b4	b3	b2	b1	B0	τιμή
<b>CSRC</b>	<b>TX9</b>	<b>TXEN</b>	<b>SYNC</b>	-	<b>BRGH</b>	<b>TRMT</b>	<b>TX9D</b>	
<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>0x24</b>

**RCSTA** : Οι ρυθμίσεις της διαδικασίας της σειριακής λήψης γίνονται μέσω του ειδικού καταχωρητή ελέγχου **RCSTA**. Τα περιεχόμενα του, φαίνονται στον παρακάτω πίνακα για : ενεργοποίηση σειριακής θύρας (SPEN=1), ασύγχρονη επικοινωνία (SREN=0), ενεργοποίηση διαδικασίας συνεχούς λήψης (CREN=1), χωρίς bit ισοτιμίας (ADDEN=0).

**RCSTA : Receive Status and Control Register**

b7	b6	b5	b4	b3	b2	b1	b0	τιμή
<b>SPEN</b>	<b>RX9</b>	<b>SREN</b>	<b>CREN</b>	<b>ADDEN</b>	<b>FERR</b>	<b>OERR</b>	<b>RX9D</b>	
<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0x90</b>

**SPBRG**: Η ταχύτητα της σειριακής επικοινωνίας καθορίζεται με τη βοήθεια του καταχωρητή SPBRG (Baud Rate Generator). Όπως είδαμε παραπάνω, το bit **BRGH** του καταχωρητή **TXSTA<2>** καθορίζει δύο περιοχές ταχυτήτων επικοινωνίας σύμφωνα με τον πίνακα:

<b>SYNC</b>	<b>BRGH=0 (Low Speed)</b>	<b>BRGH=1 (High Speed)</b>
0	(Asynchronous) Baud Rate = $F_{osc}/(64(X+1))$	<b>Baud Rate = <math>F_{osc}/(16(X+1))</math></b>
1	(Synchronous) Baud Rate = $F_{osc}/(4(X+1))$	N/A

με **F<sub>osc</sub>** συμβολίζεται η συχνότητα του εξωτερικού κρυστάλλου στο κύκλωμα χρονισμού του μικροελεγκτή και

με **X** συμβολίζεται η δεκαδική τιμή του καταχωρητή SPBRG για την ταχύτητα σειριακής επικοινωνίας που επιθυμούμε.

Επειδή έχουμε επιλέξει ασύγχρονη σειριακή επικοινωνία με υψηλές ταχύτητες και οι δοκιμές γίνονται στην εκπαιδευτική πλακέτα SE1001 με κρύσταλλο **4 MHz** και :

**BRGH = 1 (TXSTA<2>)**

**F<sub>osc</sub> = 4 MHz**

**Baud Rate = 9600 bps**

αντικαθιστώντας στον τύπο  $9600=4MHz/[16*(X+1)]$ , τις παραπάνω τιμές, υπολογίζουμε ότι η ποσότητα **X=25** (δεκαδική τιμή) or **X=0x19** (δεκαεξαδική τιμή). Άρα τα περιεχόμενα του καταχωρητή **SPBRG** όπως στον παρακάτω πίνακα:

**SPBRG : Baud Rate Generator**

b7	b6	b5	b4	b3	b2	B1	b0	τιμή
<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0x19</b> ή <b>25<sub>10</sub></b>

**RCREG** και **PIR1** : Η λήψη του χαρακτήρα στο πρόγραμμα γίνεται με την ανάγνωση του καταχωρητή **RCREG**. Δηλαδή διαβάζοντας τον καταχωρητή **RCREG** βλέπουμε ποιος χαρακτήρας ήρθε στον PIC από τον προσωπικό Η/Υ. Η ενημέρωση ότι ολοκληρώθηκε η διαδικασία της φόρτωσης του καταχωρητή RCREG υλοποιείται μέσω της σημαίας-bit **RCIF** του καταχωρητή **PIR1** (PIR1<bit5>).

**TXREG:** στον καταχωρητή αυτό φορτώνουμε το χαρακτήρα που θέλουμε να στείλουμε από τον PIC στον προσωπικό Η/Υ. Όταν ολοκληρωθεί η μεταφορά του χαρακτήρα από τον TXREG στον καταχωρητή TSR, εγείρεται (τίθεται σε 1) η σημαία **TXIF** του καταχωρητή PIR1 (PIR1<bit4>).

**PIR1 : Peripheral Interrupts Flag Register ( 0x0C, BANK 0 )**

b7	b6	b5	b4	b3	b2	b1	B0
<b>PSPI F</b>	<b>ADIF</b>	<b>RCIF</b>	<b>TXIF</b>	<b>SSPI F</b>	<b>CCP1IF</b>	<b>TMR2IF</b>	<b>TMR1IF</b>
	✓	✓	✓				

Για την περίπτωση της εφαρμογής μας, τα bits που μας ενδιαφέρουν είναι τα:

**ADIF:** σημαία ένδειξης κατάστασης του αναλογικο-ψηφιακού μετατροπέα

1 = ολοκλήρωση της A/D μετατροπής

0 = μη-ολοκλήρωση της A/D μετατροπής

**RCIF:** σημαία ένδειξης ολοκλήρωσης διαδικασίας λήψης. Τα δεδομένα έχουν ολισθήσει μέσω του RSR και έχουν φορτωθεί παράλληλα στον RCREG και επομένως είναι έτοιμα να διαβαστούν από το πρόγραμμα.

1 = ολοκλήρωση λήψης χαρακτήρα μέσω USART στον RCREG

0 = δεν έχει έρθει ή δεν ολοκληρώθηκε ακόμα η λήψη χαρακτήρα και επομένως δεν υπάρχουν δεδομένα στο RCREG

**TXIF:** σημαία ένδειξης κατάστασης του καταχωρητή TXREG. Μόνο όταν είναι άδειος ο καταχωρητής TXREG μπορούμε να 'του φορτώσουμε' τα δεδομένα που θέλουμε να αποστείλουμε μέσω του USART. Στην συνέχεια τα δεδομένα αυτά φορτώνονται παράλληλα στον καταχωρητή TSR και από εκεί ολισθαίνουν σειριακά μέσω του ακροδέκτη RC6/TX (PORTC<6>) κατά τη διαδικασία της σειριακής εκπομπής.

1 = ο TXREG είναι άδειος

0 = ο TXREG είναι ακόμα γεμάτος με προηγούμενη τιμή

Τα βήματα της άσκησης είναι τα παρακάτω:

- 1) Ακολουθούμε τις οδηγίες Άσκησης 1 και δημιουργούμε φάκελο με όνομα **Askisi5** μέσα στον οποίο θα αποθηκευτούν όλα τα αρχεία που δημιουργούνται από το περιβάλλον της MikroC μεταξύ των οποίων το αρχείο project (.mcpri), αρχείο κώδικα C (.c) και το αρχείο hex. Το όνομα των αρχείων μπορεί να είναι το **Askisi4**.
- 2) Στο παράθυρο κώδικα γράφουμε τις παρακάτω εντολές. Στον κώδικα αυτό συμπεριλαμβάνεται και ο προγραμματισμός του περιφερειακού ADC σύμφωνα με τις οδηγίες της Άσκησης 4. Συμπληρώνουμε σύντομα σχόλια για τη λειτουργία των εντολών όπου υπάρχουν κενές τελείες.

// δηλώσεις μεταβλητών

**unsigned short record;**

**unsigned int k;**

**void main() {**

//αρχικοποίηση του περιφερειακού USART

```

TRISC=0x80; //.....
TXSTA=0x24; //.....
RCSTA=0x90; //.....
SPBRG=0x19; // baud rate=9600 4MHz/[16x(X+1)] x=15 or 0x19
// αρχικοποίηση του περιφερειακού ADC
// configure pins
    TRISA= 0xFF; // PORTA ως input
    TRISB=0x00; // PORTB ως output
// STEP 1
    ADCON1=0x00; // Configure analog inputs and Vref+=Vdd Vref-=Vss
                // left justified
    ADCON0=0x99; // επιλογή συχνότητας δειγματοληψίας 32*Tosc,
                // από το ακροδέκτη AN3, και ενεργοποίηση A/D
// STEP 2
    PIR1=0x00; // clear ADIF flag
    PIE1=0x40; // set ADIE , ενεργοποίηση A/D interrupt
//=====
while(1){
    if(PIR1.F5=1)
    {
        record=RCREG;
        delay_ms(500);
        if(record=='m') // 'HELLO' .....
        {
            TXREG=0x48; // 'H' .....
            delay_ms(500);
            TXREG=0x45; // 'E' .....
            delay_ms(500);
            TXREG=0x4C; // 'L' .....
            delay_ms(500);
            TXREG=0x4C; // 'L' .....
            delay_ms(500);
            TXREG=0x4F; // 'O' .....

```

```

}          //.....
if(record=='a') // 'WORLD' .....
{
    delay_ms(500);
    TXREG='W'; //.....
    delay_ms(500);
    TXREG='O'; //.....
    delay_ms(500);
    TXREG='R'; //.....
    delay_ms(500);
    TXREG='L'; //.....
    delay_ms(500);
    TXREG='D'; //.....
    delay_ms(500);
}          // τέλος δεύτερου εσωτερικού if
if(record=='c') // ADC conversion .....
{
// STEP 3
    delay_ms(100);
// STEP 4
    ADCON0=0b10011101; // set GO/DONE για να αρχίσει η A/D μετατροπή
// STEP 5
// wait for the GO/DONE bit to be cleared
    do {
    } while (ADCON0.F2=1);
// STEP 6_a : read the A/D result from ADRESH and ADRESL and sent to PORTS

    PORTB=ADRESH; // only the 8 most significant bits
    TXREG=ADRESH; // στείλε στον Η/Υ μόνο το High Byte(most significant part)
    delay_ms(500);
// STEP6_b: clear ADIF bit
    PIR1.ADIF=0;
// STEP7 :delay

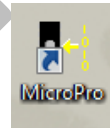
```

```

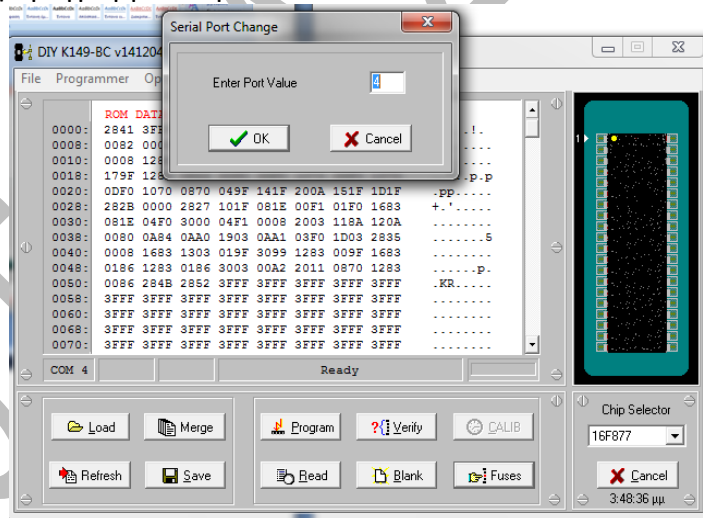
delay_ms(100);
} // τέλος δεύτερου εσωτερικού if
} // end of initial if
} // end of loop
} // end of main

```

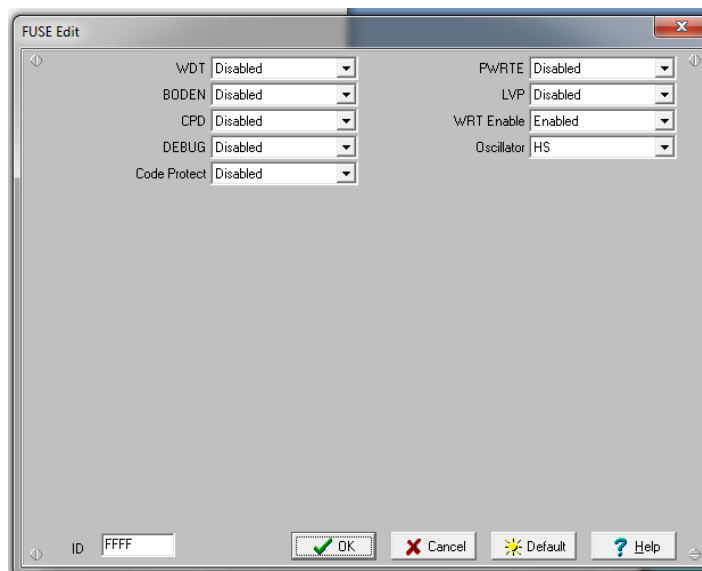
- 3) Κάνουμε συμβολομετάφραση (compilation) ώστε να διορθωθούν τα συντακτικά λάθη και να παραχθεί το αρχείο hex (οδηγίες στην Άσκηση 1)
- 4) Στη συνέχεια συνδέουμε σε μία θύρα USB την **USB PIC Programmer** με στοιχεία **K149-BC** της **DIY Electronics**. Περιμένουμε να ανιχνευθεί και να εγκατασταθεί ο driver και εντοπίζουμε σε ποια θύρα COM φαίνεται από το λειτουργικό σύστημα του Η/Υ μας (δεξί κλικ **Η/Υ μου / Διαχείριση Συσκευών / Θύρες COM** ).
- 5) Τρέχουμε την εφαρμογή **MicroPro** που συνοδεύει την προγραμματίστρια φροντίζοντας η έκδοσή της να είναι συμβατή με το λειτουργικό σύστημα. Εμείς βρήκαμε έκδοση συμβατή έως τα WINDOWS7 32 bits.



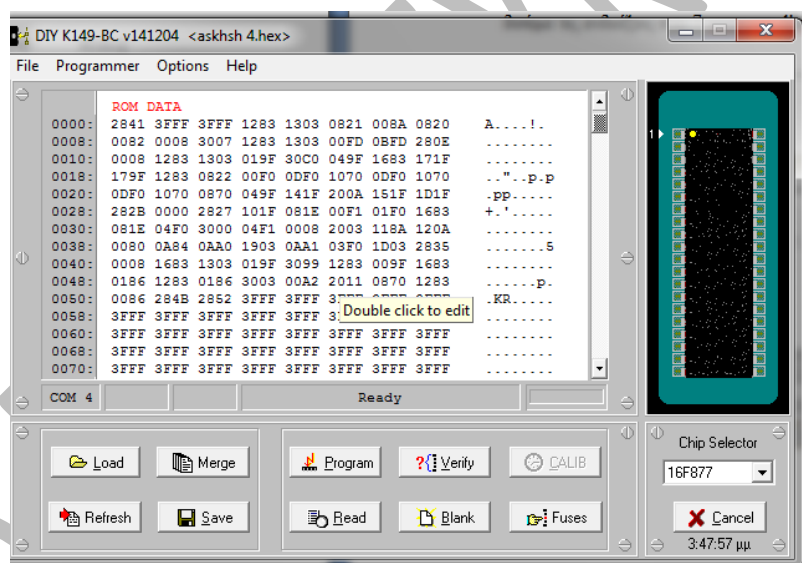
- 6) Από τη διαδρομή **File/Serial Port** δηλώνουμε τον αριθμό της **Com Port** που ανιχνεύτηκε η προγραμματίστρια.



- 7) Τοποθετούμε ένα ολοκληρωμένο PIC16F877 στην προγραμματίστρια και επιλέγουμε τον 16F877 στο πεδίο **Chip Selector** ενώ με το κουμπί **Load** φορτώνουμε το αρχείο **Askisi4.hex**. Με το κουμπί **Fuses** ρυθμίζουμε:



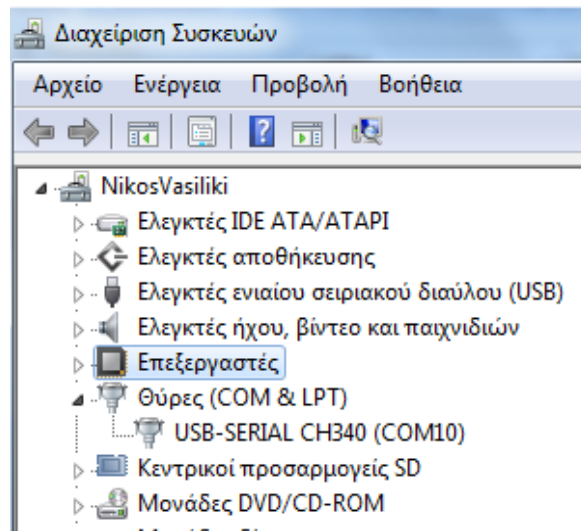
8) Στο παράθυρο ROM DATA βλέπουμε το αρχείο askisi5.hex . Με το κουμπί **Blank** επιλέγουμε **Erase Chip** για να διαγράψουμε τα προηγούμενα περιεχόμενα του chip και στη συνέχεια πατάμε το κουμπί **Program** για να μεταφερθεί ο κώδικας hex από τον Η/Υ στο chip.



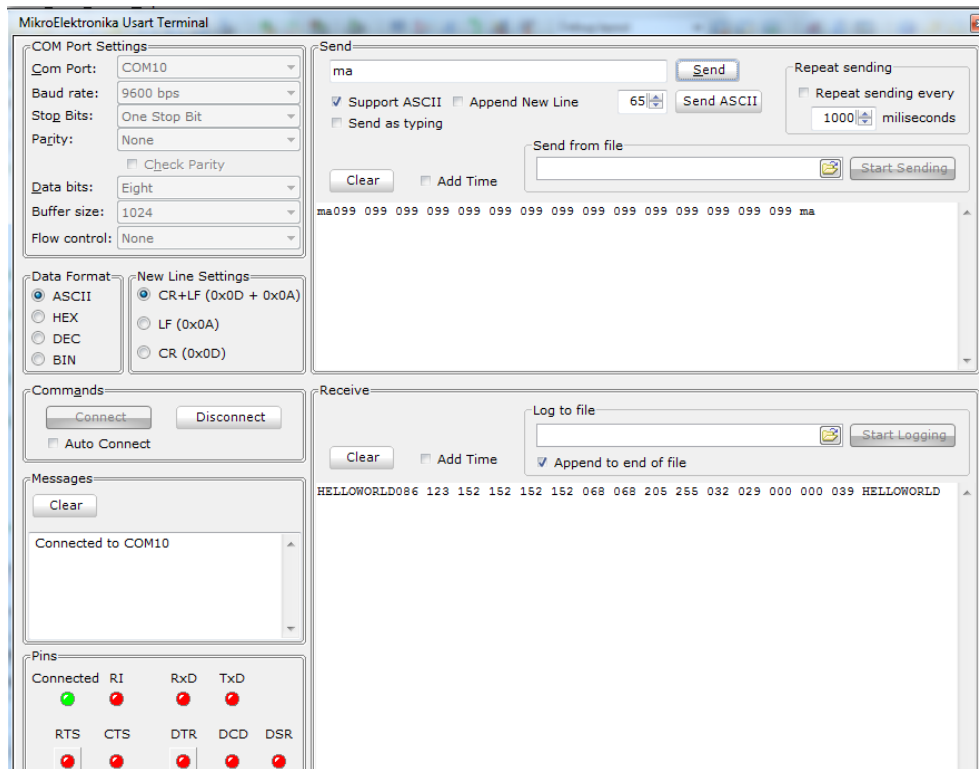
9) Μεταφέρουμε το chip στην εκπαιδευτική πλακέτα SE1001 και την τροφοδοτούμε με τάση 9-12 Volts.

10) Συνδέουμε το σύνδεσμο DB9 της εκπαιδευτικής πλακέτας SE1001 σε μία θύρα USB του υπολογιστή με το καλώδιο μετατροπέα USB to Serial. Αν χρειαστεί περιμένουμε λίγα λεπτά ώστε να ανιχνευθεί από το λειτουργικό σύστημα και από τη διαδρομή Η/Υ μου/Ιδιότητες Συστήματος/Διαχείριση Συσκευών/Θύρες Com & LPT εντοπίζουμε σε ποια θύρα COM φαίνεται πλέον ο μετατροπέας USB to Serial και επομένως και η εκπαιδευτική πλακέτα SE1001.





- 11) Φορτώνουμε το πρόγραμμα HyperTerminal Windows ή κάποιο αντίστοιχο. Εμείς φορτώνουμε την εφαρμογή USART Terminal στο περιβάλλον της mikroc ( στο προγραμματιστικό περιβάλλον της mikroC (μενού Tools / USART Terminal) και ορίζουμε τις ρυθμίσεις της σειριακής επικοινωνίας (Com10, 9600 bps, 8 bit, no parity) και στη συνέχεια πατάμε το πλήκτρο **Connect**.
- 12) Επιλέγουμε DATA FORMAT σε κώδικα ASCII και από στο πεδίο SEND γράφουμε το χαρακτήρα 'm' και 'a' είτε μεμονωμένα είτε μαζί και πατάμε το πλήκτρο SEND. Στο παράθυρο RECEIVE βλέπουμε τους ASCII χαρακτήρες με τους οποίους ο PIC απαντάει με το μήνυμα **'HELLO WORLD'** .
- 13) Επιλέγουμε DATA FORMAT σε κώδικα DECIMAL και από στο πεδίο SEND γράφουμε το χαρακτήρα 'c'. Τότε βλέπουμε τον αριθμό των ψηφιακών επιπέδων(0-255) στα οποία αντιστοιχεί η αναλογική τάση που συνδέεται μέσω του ποτενσιόμετρου στον ακροδέκτη AN3 του PIC. Μεταβάλλουμε ποτενσιόμετρου σε διάφορες θέσεις και πατώντας διαδοχικά το πλήκτρο SEND βλέπουμε τις διάφορες ψηφιακές ενδείξεις που στέλνει ο PIC σε δεκαδική τιμή.



14) Τέλος πατάμε **Disconnect**. Αφαιρούμε πρώτα την τροφοδοσία της εκπαιδευτικής πλακέτας SE1001 και μετά αποσυνδέουμε το μετατροπέα USB to Serial για να αποφύγουμε τυχόν βραχυκυκλώματα.

Όνοματεπώνυμο : ..... Ημερομηνία: .....

### **Βιβλιογραφία :**

- 1) « **Εισαγωγή στον προγραμματισμό μικροελεγκτών, FPGA και CPLD: Επιλεγμένες Εφαρμογές**»  
Σ. Μπουλταδάκης, Γ. Πατουλίδης και Ν. Ασημόπουλος, Εκδόσεις ΤΖΙΟΛΑ, Θεσσαλονίκη 2011, ISBN: 978-960-418-291-6
- 2) « **Υλικό και Λογισμικό Μετρήσεων: Παραδείγματα και Εφαρμογές**»  
Σ. Μπουλταδάκης και Ι. Καλόμοιρος, Εκδόσεις ΤΖΙΟΛΑ, Θεσσαλονίκη 2009, ISBN: 978-960-418-161-2

