**ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 1**

**Τίτλος:**

ΑΝΑΠΤΥΞΗ ΑΠΛΗΣ ΕΦΑΡΜΟΓΗΣ ΟΔΗΓΗΣΗΣ ΕΝΔΕΙΚΤΗ 7 ΤΜΗΜΑΤΩΝ ΣΤΗΝ ΤΗΝ ΕΙΚΟΝΙΚΗ ΠΛΑΚΕΤΑ PICSIMLAB

**Περιγραφή:**

Χρησιμοποιώντας το παράδειγμα που σας δίνεται  δημιουργήστε μια εφαρμογή η οποία θα τρέχει στην πλακετα 4 (PICGenios) του PICSIMLAB και σε  μικροελεγκτή PIC18F4520 και θα εμφανίζει στο δεκαεξαδικό display DISP1 διαδοχικά (κάθε 0,5 sec) τους αριθμούς 0...9.

Για την δημιουργία της εφαρμογής χρειάζεται να γνωρίζουμε την λειτουργία του δεκαεξαδικού display κοινής καθόδου και την σύνδεσή του με τον μικροελεγκτή στην πλακέτα.

Όπως φαίνεται από το παρακάτω σχέδιο το display DISP1 είναι συνδεδεμένο ως εξής. Οι άνοδοι των LED είναι συνδεδεμένες στις ακίδες RD7-RD0 της θύρας D και η κοινή κάθοδος μπορεί να γειωθεί μέσω ενός τρανζίστορ του οποίου η βάση οδηγείται από την ακίδα 2 της θύρας Α.



Για να εμφανίσουμε ένα σχήμα στο DISP1, πρέπει να δώσουμε 5V ( λογικό 1) στις ακίδες που είναι συνδεδεμένες με τα LED που θέλουμε αναμμένα και 0V ( Λογικό 0) σε όσα θέλουμε να είναι σβηστά. Για να ενεργοποιήσουμε το DISP1 πρέπει να δώσουμε 5V (Λογικό 1) στην ακίδα που οδηγεί τη βάση του τρανζίστορ.

Ακολουθώντας το σχήμα και το παρακάτω πίνακα που δείχνει ποιο ποδαράκι αντίστοιχεί σε κάθε LED μπορούμε να καθορίσουμε τα εξής βήματα:



1. Στα ποδαράκια της θύρας D δίνουμε τις τιμές 0 ή 1 ανάλογα με το σχήμα που θέλουμε να εμφανιστεί.. Για παράδειγμα για να εμφανιστεί το 3 πρέπει να δώσουμε τις τιμές 01001111. Αυτό επιτυγχάνεται με την εντολή PORTD = 0x01001111.
2. Για να ανάψει το DISP1 πρέπει να γειώσουμε την κάθοδο και αυτό γίνεται γράφοντας 0 στο ποδαράκι 2 της θύρας Α. Αν θέλουμε να αλλάξουμε την τιμή μόνο ενός ψηφίου μιας θύρας μπορούμε να χρησιμοποιήσουμε την εντολή PORTA.F2=1;
3. Για να απενεργοποιήσουμε το display γράφουμε 0 στο ίδιο ποδαράκι και απενεργοποιούμε το τρανζίστορ (PORTA.F2 =0 );
4. Για να δημιουργήσουμε μια καθυστέρηση στις αλλαγές ώστε να είναι ορατές μπορούμε να χρησιμοποιήσουμε μια ενσωματωμένη ρουτίνα όπως delay\_ms(nnn) η οποία δέχεται ως παράμετρο nnn τον αριθμό των millisecond της καθυστέρησης.
5. Επειδή όλες οι θύρες του συγκεκριμένου μικροελεγτή μπορούν να λειτουργήσουν και ως είσοδοι και ως έξοδοι, πριν τη χρήση τους πρέπει να προσδιορίσουμε αν θέλουμε να είναι είσοδοι ή έξοδοι. Αυτό επιτυγχάνεται μέσω καταχωρητών διαμόρφωσης TRIS?. Για παράδειγμα, για την λειτουργία όλων των ακίδων της θύρας D ως εξόδους δίνουμε στον καταχωρητή TRISD τις τιμές 0 σε όλα τα ψηφία (TRISD = 0b00000000) ενώ αν θέλουμε στην θύρα Α μόνο το ποδαράκι 2 να είναι έξοδος και τα υπόλοιπα είσοδοι δίνουμε στον καταχωρητή TRISA τις τιμές 0b11111011.
6. Για την συνεχή επανάληψη της εκτέλεσης των εντολών μας, χρησιμοποιούμε τις τυποποιημένες δομές επανάληψης της γλώσσας προγραμματισμού C.

Μια δυνατή υλοποίηση της ζητούμενης εφαρμογής παρατίθεται στη συνέχεια.

main ()

{

int i;

//define a char array to hold the sixteen combinations for 0-F

char hex\_digit[16] = {0x3F, 0x06, 0x5B, 0x4F,

0x66, 0x6D, 0x7D, 0x07,

0x7F, 0x6F, 0x77, 0x7C, 0x39, 0x5E, 0x79, 0x71} ;

TRISD = 0x00; /SET register TRISD to make PORTD to operate as output

TRISA = 0xFB; // SET bit 2 of register TRISA to 0 to make PORTA pin 2 operate as output.

do {

for (i=0;i<10;i++) {

PORTD = hex\_digit[i]; //Write the appropriate combination of bits to PORTD

PORTA.F2 =1; //Enable DISP1 by turning on Transistor connected to pin RA2

delay\_ms(500);

PORTA.F2 =0; // Turn DISP1 off by writing 0 to RA2 which drives the transistor base;

delay\_ms(500);

}

} while(1); //loop continuously

}