

Άσκηση έκτη του Εργαστηρίου

Στην έκτη άσκηση εξετάζεται η οδήγηση ενδείκτων επτά τομέων (seven segment displays) με τη χρήση πινάκων μετατροπής (Look-up tables) σε συνδυασμό με δεδομένα που εισάγονται από dip-switches .

Για την εισαγωγή δεδομένων από τα dip-switches της αναπτυξιακής πλακέτας μπορεί να χρησιμοποιηθεί η πόρτα D και για την οδήγηση του ενδείκτη η πόρτα B και η πόρτα C. Σε ότι αφορά τη λειτουργία του προγράμματος χρησιμοποιούνται οι έτοιμες συναρτήσεις που δίδονται από το CCS C Compiler όπως έχουν ήδη αναφερθεί στην άσκηση 2.

Παράδειγμα δίδεται παρακάτω.

Να γραφεί πρόγραμμα που να διαβάζει δύο αριθμούς των 4 bits από τη πόρτα D.

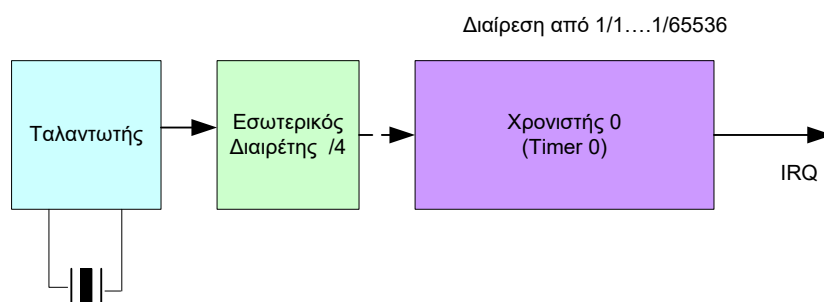
Ο πρώτος όταν το RD7 είναι σε λογικό 0 και ο δεύτερος όταν το bit RD7 είναι σε λογικό 1.

Το άθροισμα των δύο αριθμών θα πρέπει να εμφανίζεται στον ενδείκτη. Η λειτουργία των ενδείκτων γίνεται με τη μέθοδο της διακοπής.

Για την ανάγνωση των διακοπών να γίνει ξεχωριστή συνάρτηση που να ελέγχει τα διακοπτάκια και να επιστρέφει τη τιμή που τίθεται από τους διακόπτες.

Λύση

Για τον έλεγχο των ενδείκτων θα χρησιμοποιηθεί η μέθοδος της διακοπής. Και επειδή κάθε χρονική στιγμή ένας μόνο ενδείκτης είναι ενεργός για να μπορεί το μάτι του παρατηρητή να βλέπει όλους τους ενδείκτες αναμμένους θα πρέπει να ληφθεί χρόνος για το interrupt τέτοιος ώστε, καθώς κάθε φορά θα οδηγείται ένα display για τον χρόνο αυτό και να ξεγελά το μάτι του παρατηρητή. Ο χρόνος αυτός θα πρέπει να είναι τουλάχιστον $T=1/(3*66,66\text{Hz})=5\text{ msec}$. Δηλαδή συχνότητα εναλλαγής 66,66 Hz. Το μάτι με τη συχνότητα αυτή θα τα βλέπει όλα αναμμένα. Καλό είναι στις περιπτώσεις υπολογισμού του χρόνου με τη μέθοδο διακοπής να σχεδιάζεται το παρακάτω σχέδιο, που βοηθά στον υπολογισμό της αρχική τιμής του Hardware μετρητή (Timer0).



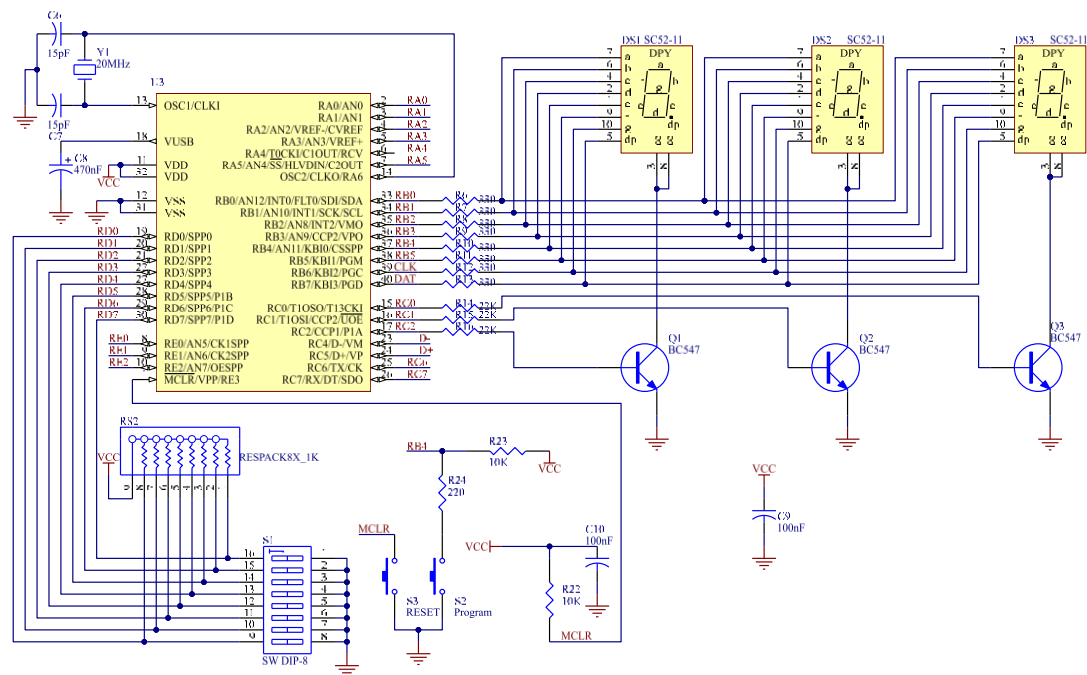
Για τον υπολογισμό του χρόνου των 5msec χρησιμοποιώντας το παραπάνω σχήμα έχουμε:

Εργαστήριο στα Ενσωματωμένα Συστήματα – Άσκηση 6^η

20,833333 nsec * 4 * 1 * (65536 – αρχική τιμή) = 5000000 nsec => αρχική τιμή = 5536.

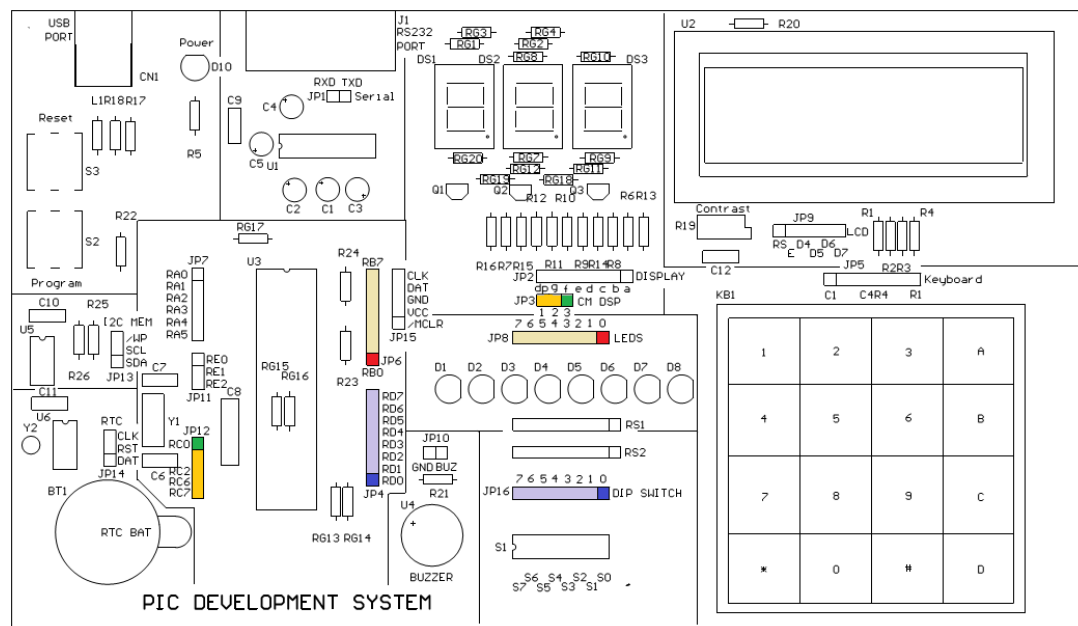
Η τιμή της διαίρεσης του Prescaler λαμβάνεται 1 (χωρίς τη χρήση του με τον timer0) διότι ο επιθυμητός χρόνος είναι πολύ μικρός (5 msec) Με όλες τις παραπάνω πληροφορίες μπορεί να γραφεί το πρόγραμμα χρησιμοποιώντας τις έτοιμες συναρτήσεις του C Compiler.

Το σχηματικό του κυκλώματος



ΑΣΚΗΣΗ 6_1

Οι συνδέσεις στην πλακέτα του εργαστηρίου



Το πρόγραμμα του παραδείγματος

```
#include <main.h>
```

```
#use standard_io ( A ) // Standard είσοδοι και έξοδοι
```

```
#use standard_io ( B ) // Standard είσοδοι και έξοδοι
```

```
#use standard_io ( C ) // Standard είσοδοι και έξοδοι
```

```
#byte PORTA      =0xF80 //ορισμός των θυρών με τη θέση τους στη
```

```
#byte PORTB      =0xF81 // μνήμη
```

```
#byte PORTC      =0xF82
```

```
#byte PORTD      =0xF83
```

```
#byte PORTE      =0xF84
```

```
// Ορισμός των μεταβλητών (Variable definition )
```

```
int8 number1=0;
```

```
int8 number2=0;
```

```
int16 sum=0;
```

```
int8 des=0;
```

```
int8 table[16] = { 0x3f, //Πίνακας μετατροπής για ενδείκτη 7 τομέων
                  0x06, //Οι 10 τιμές του πίνακα
                  0x5b, //
                  0x4f,
                  0x66,
                  0x6d,
                  0x7d,
                  0x07,
                  0x7f,
```

```
        0x6f,  
        0x77,  
        0x7c,  
        0x39,  
        0x5e,  
        0x79,  
        0x71};  
  
int8 dig[3] = {1,2,4}; // Πίνακας μετατροπής για την επιλογή των ψηφίων  
  
// Δήλωση συναρτήσεων  
  
void timer0_int(void);  
void init (void);  
int8 number(void);  
  
// Κύριο πρόγραμμα (Main program)  
  
void main()  
{  
    init();  
  
    while (1){  
        //  
        while (input(PIN_D7) ==1){ // Έλεγχος του RD1 για λογικό 1  
            }  
            number1 = int8 number(); // Κλήση της υπορουτίνας number  
            με //επιστροφή int8  
            sum = number1;  
            while (input(PIN_D7) ==0){  
            }  
            number2 = int8 number();  
            sum = number2;  
            delay_ms(500);  
            sum = number1 + number2 ;  
            delay_ms(2000);  
            sum = 0;  
        }  
    }  
  
// Ορισμός συναρτήσεων (Function definition)  
  
// Ρουτίνα εξυπηρέτησης της διακοπής (Interrupt service routine)  
  
#INT_TIMER0 HIGH // Διακοπή με μεγάλη προτεραιότητα  
void timer0_int(void){  
    int16 mon,dec,eka;
```

```
    eka = 0;
    dec = (int8) (sum/10); // των δεκάδων
    mon = (int8) (sum%10); // των μονάδων
        set_timer0(5536); // Αρχική τιμή του Hardware μετρητή
        des = ++des%3; // Modulo 3 μετρητής που λειτουργεί ως
                        // δείκτης για το πίνακα επιλογής
                        // του ψηφίου που δείχνει κάθε στιγμή

    PORTC = dig[des]; // Επιλογή ενός από τα 3 ψηφία
    if (des==0){
        PORTB = table[mon]; // Δείξε τις μονάδες
    }

    if (des==1){
        PORTB = table[dec]; // Δείξε τις δεκάδες
    }

    if (des==2){
        PORTB = table[eka]; // Δείξε τις εκατοντάδες
    }
}

// Ρουτίνα αρχικοποίησης (Initialization routine)
void init (void){
    set_tris_b(0x00); // Καθορισμός της πόρτας B ως έξοδος
    set_tris_c(0x00); // Καθορισμός της πόρτας C ως έξοδος
    set_tris_d(0xff); // Καθορισμός της πόρτας D ως είσοδος
    PORTB = 0;
    PORTC = 0;

    des = 0; // Αρχική τιμή επιλογής ψηφίου
    SETUP_TIMER_0(T0_INTERNAL | T0_DIV_1 ); // Prescaler
                                           // /1
    set_timer0(5536); // Αρχική τιμή του hardware μετρητή
    enable_interrupts(INT_TIMER0); // Ενεργοποίηση της
                                   // διακοπής του timer0
    enable_interrupts(GLOBAL); // Ενεργοποίηση του γενικού
                                // διακόπτη των διακοπών
}

int8 number(void){
    int8 temp;
    delay_ms(30);
    temp = PORTD & 0x0F;
    return(temp);
}
```

}

Ασκήσεις που θα πρέπει να ετοιμαστούν για το έκτο εργαστήριο

1. Να γραφεί πρόγραμμα σε γλώσσα προγραμματισμού C για τον μικροελεγκτή PIC 18F4550 με το οποίο ο μικροελεγκτής θα διαβάζει από τα dip-switches όπως και στην παραπάνω άσκηση τρεις αριθμούς δηλαδή (Ο πρώτος όταν το RD7,RD6 είναι σε λογικό 0 ο δεύτερος όταν το bit RD7,RD6 είναι σε λογικό 1 και ο τρίτος όταν το RD7 σε λογικό 0 και RD6 σε λογικό 1. Οι αριθμοί είναι των 4 bits και εισάγονται από τα τέσσερα λιγότερο σημαντικά bits της πόρτας D RD0,RD1,RD2,RD3). Ο πρώτος αριθμός ορίζει την αριθμητή πράξη που θα εκτελεστεί μεταξύ των δύο επόμενων. Αν ο πρώτος αριθμός είναι 1 τότε η πράξη μεταξύ του δεύτερου και τρίτου θα είναι πρόσθεση αν ο πρώτος αριθμός είναι 2 τότε αφαιρεί από το δεύτερο το τρίτο, αν ο πρώτος αριθμός είναι 3 τότε να εκτελείται πολλαπλασιασμός μεταξύ του δεύτερου και τρίτου. Το αποτέλεσμα των πράξεων θα εμφανίζεται στους ενδείκτες 7 τομέων.

2. Να γραφεί πρόγραμμα σε γλώσσα προγραμματισμού C για το μικροελεγκτή PIC18F4550 που να λειτουργεί το σύστημα ως μετρητή συμβάντων:

Το σύστημα θα λειτουργεί ως εξής:

- Η είσοδος των παλμών (συμβάντων) θα γίνεται από το bit RB1. Η μέτρηση των παλμών θα γίνεται στη κάθοδό τους.
- Η απεικόνιση του αριθμού των εισερχομένων παλμών θα γίνεται στους ενδείκτες 7 τομέων από τη πόρτα D για τα 7 τμήματα και από τη πόρτα C για την επιλογή του ενεργού ενδείκτη.
- Να χρησιμοποιηθεί η μέθοδος της διακοπής από την υπερχείλιση του timer0 για τη μέτρηση του χρόνου και οι πίνακες μετατροπής για την οδήγηση των ενδείκτων.
- Να χρησιμοποιηθεί η μέθοδος της διακοπής από την αλλαγή στο μέτωπο του RB1.

Ερωτήσεις που πρέπει να απαντηθούν.

1. Στον ακόλουθο κώδικα ποια είναι η τιμή του x μετά την εκτέλεση του προγράμματος

```
char s1[20] = "Hello";  
char s2[15];  
int8 x;  
strcpy(s2, s1);  
x = strlen(s2);
```

- a. 20
- b. 15
- c. 5
- d. 6
- e. 0

2. Στο παρακάτω κώδικα ποιο string είναι αυτό που βγαίνει στην έξοδο

```
char s1[20] = "This is a test";  
char s2[] = "at";
```

```
strcpy(strstr(s1,"is"),s2);  
cout << s1;
```

- a. This is a test
- b. That is a test
- c. This at a test
- d. That
- e. Δεν είναι σωστή η έκφραση στη C

3. Ποια είναι η έξοδος από το παρακάτω κώδικα

```
int8 xx=1;  
#define xx 2  
#if xx==1  
    cout << "one";  
#elif xx == 2  
    cout << "two";  
#else  
    cout << "three";  
#endif
```

- a. one
- b. two
- c. three
- d. τίποτε
- e. Υπάρχει συντακτικό λάθος

4. Υποθέστε ότι ο PIC λειτουργεί στη συχνότητα των 20MHz και ο timer 0 λειτουργεί σε 8 bit λειτουργία και ο prescaler διά 4. Πόσες διακοπές θα παράγονται σε κάθε δευτερόλεπτο

- a. 1221
- b. 1953
- c. 4883
- d. 78125
- e. 625000