



# ***Ενσωματωμένα Συστήματα***

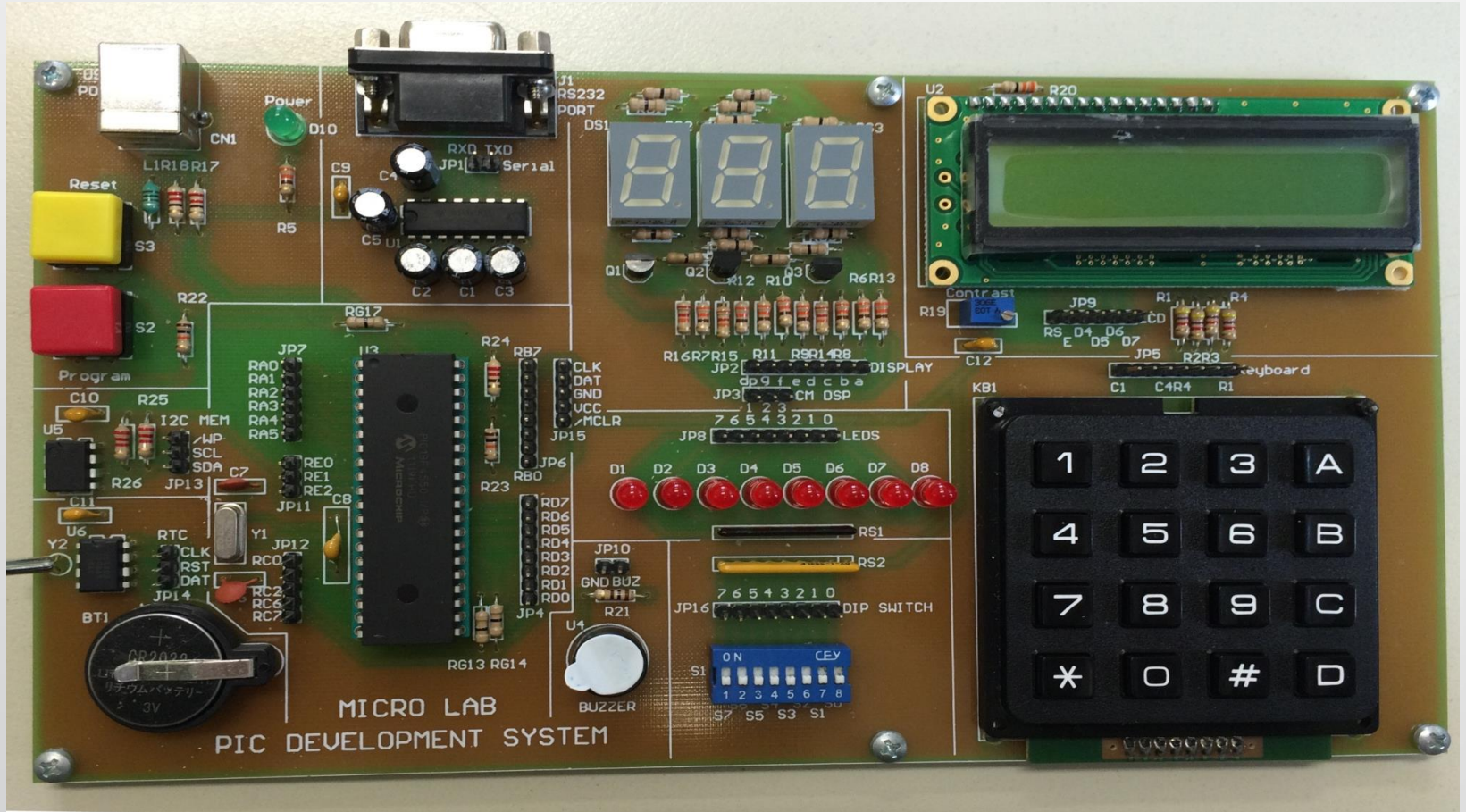
(6<sup>ο</sup> εξάμηνο)

**03-Digital-In-Out Ports**

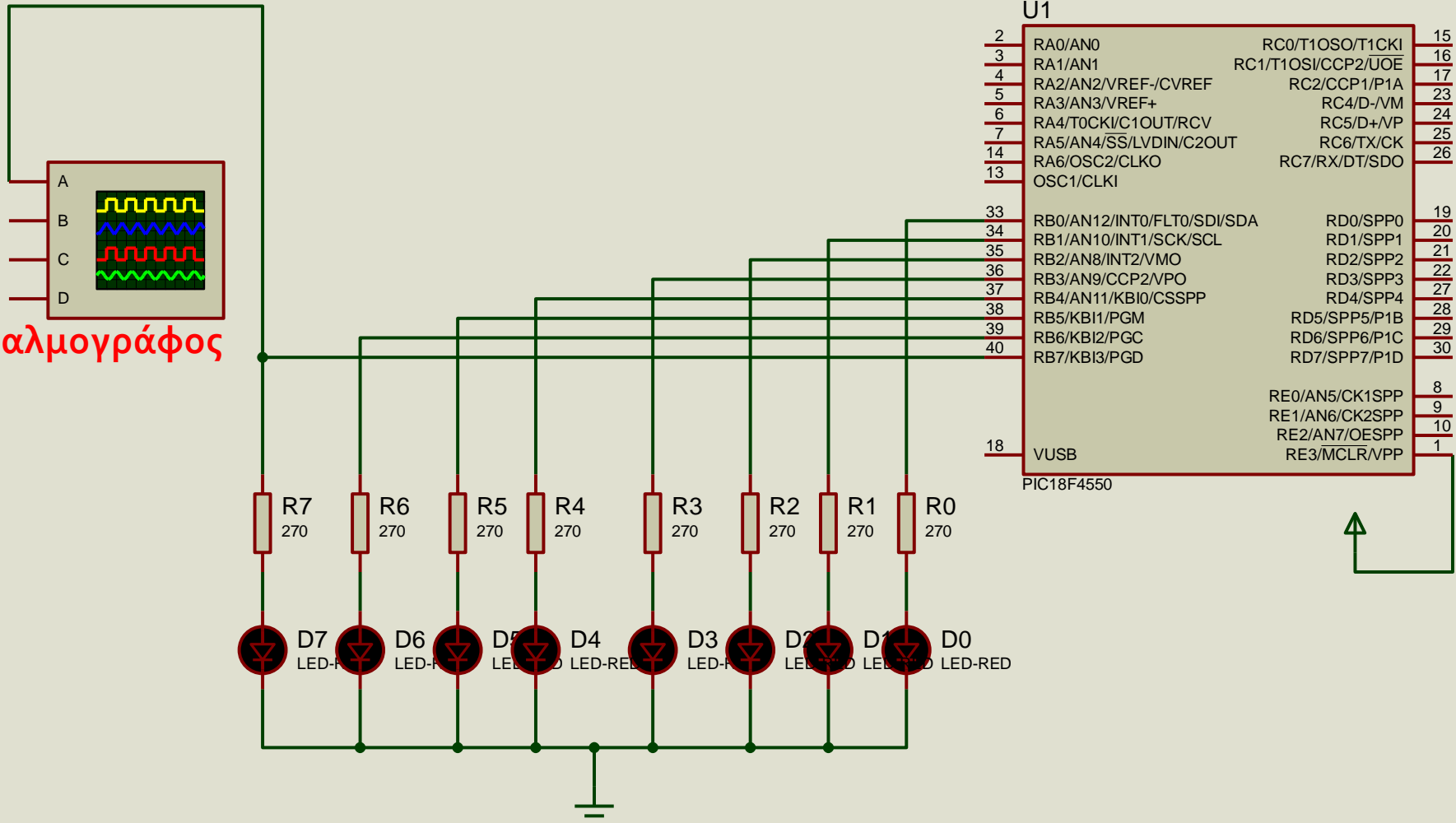
**Διδάσκοντες:** Παπαδοπούλου Μαρία  
Επίκουρη Καθηγήτρια

Θεσσαλονίκη 2025

# Πλακέτα ανάπτυξης εφαρμογών του εργαστηρίου «Ενσωματωμένα Συστήματα»



# Άσκηση 01a. Αναβοσβήσιμο των 8 LED Σύνδεση παλμογράφου στο Proteus



# Άσκηση 01a. Αναβοσβήσιμο των 8 LED

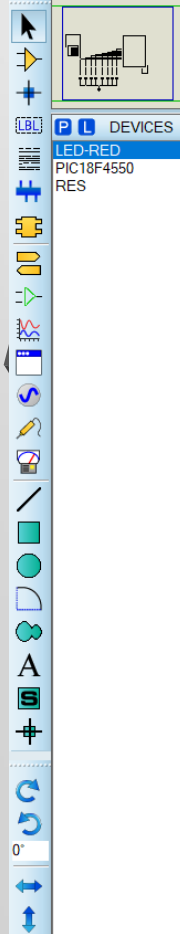
## Σύνδεση παλμογράφου στο Proteus

exercise\_01a - Proteus 8 Professional - Schematic Capture

File Edit View Tool Design Graph Debug Library Template System Help

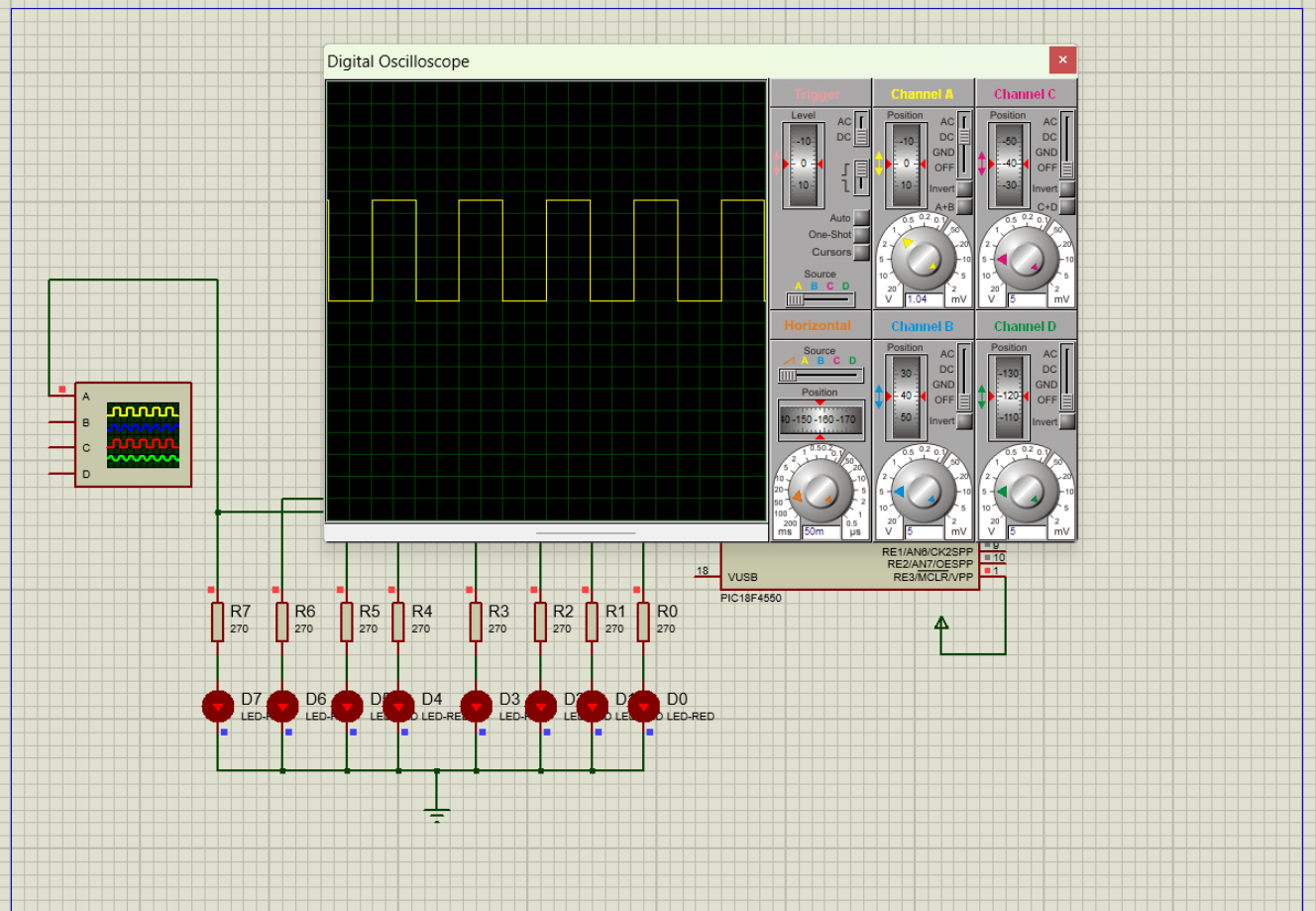
Base Design

Schematic Capture



DEVICES

LED-RED  
PIC18F4550  
RES



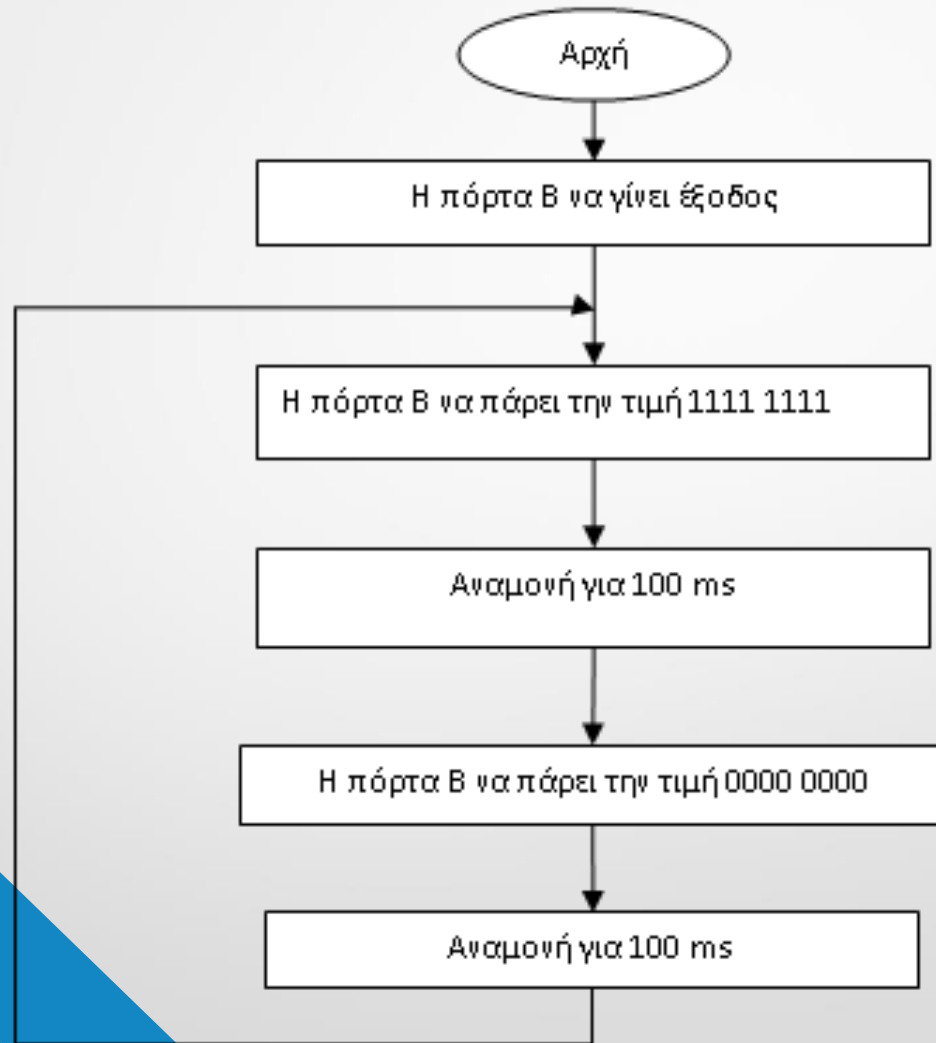
7 Message(s)

ANIMATING: 00:02:46.250000 (CPU load 37%)

+5900.0

## Άσκηση 01a. Αναβοσβήσιμο των 8 LED

### Διάγραμμα Ροής



## Άσκηση 01a. Αναβοσβήσιμο των 8 LED

**Φάκελος στον υπολογιστή όπου δημιουργούμε το Project της άσκησης**

### **Exer01a.c**

Το αρχείο σε γλώσσα C  
(Το αρχείο του προγράμματος που  
δημιουργήσαμε)

### **Exer01a.hex**

Είναι το αποτέλεσμα της  
μετάφρασης του προγράμματος  
από γλώσσα c σε γλώσσα μηχανής.  
Αυτό είναι το αρχείο που θα  
φορτωθεί στον μικροελεγκτή  
Ανοίγει με το notepad

### **18F4550.h**

Αρχείο με πληροφορίες για τον  
Μικροελεγκτή που χρησιμοποιούμε.  
Ανοίγει με notepad

### **main.h**

Αρχείο με αρχικές ρυθμίσεις.  
Ανοίγει με notepad

### **add.txt**

Αυτό το αρχείο περιέχει μια γραμμή που θα πρέπει  
να προσθέτουμε κάθε φορά που δημιουργείται  
νέο αρχείο askisi-1a.hex στην πρώτη γραμμή (είναι  
πρόβλημα του προγράμματος).  
Ανοίγει με notepad

## Άσκηση 01a. Αναβοσβήσιμο των 8 LED

### Κώδικας σε γλώσσα C

```
#include <main.h>    // Συμπεριλαμβάνουμε το αρχείο
                     // αρχικών ρυθμίσεων main.h
#byte PORTB =0xF81   // Αποδίδουμε στη θέση μνήμης 0xF81
                     // το όνομα PORTB, δηλαδή δημιουργούμε μια μεταβλητή
                     // των 8-bit, της οποίας η τιμή θα
                     // αποθηκεύεται στη θέση μνήμης F81h.
                     // Η θέση μνήμης F81h είναι ο καταχωρητής
                     // δεδομένων της πόρτας B

void main(void){      // Ανοίγει η αγκύλη της main()
    set_tris_b(0x00); // Κάνουμε την πόρτα B έξοδο,
                     // δηλαδή δίνουμε στον καταχωρητή κατεύθυνσης
                     // της πόρτας B την τιμή 0000 0000.

    PORTB =0b11110000; // Αποδίδουμε στον καταχωρητή δεδομένων
                     // της πόρτας B την τιμή 1111 0000.

    while(TRUE){ // Ατέρμων βρόχος
        PORTB=0b11111111; // Όλοι οι ακροδέκτες της πόρτας B λαμβάνουν τιμή 1
        delay_ms(100);    // Καθυστέρηση 100 ms
        PORTB=0b00000000; // Όλοι οι ακροδέκτες της πόρτας B λαμβάνουν τιμή 0
        delay_ms(100);    // Καθυστέρηση 100ms
    }
}                      // Κλείνει η αγκύλη της main()
```



## Άσκηση 01a. Αναβοσβήσιμο των 8 LED

### Συμβουλές

1. Τοποθετείτε σχόλια στα προγράμματα σας. Με αυτόν τον τρόπο, όταν θα τα δείτε μετά από καιρό, θα μπορείτε να θυμηθείτε πώς λειτουργούν. Επίσης, θα διευκολύνετε και αυτόν που θα διαβάσει το πρόγραμμά σας να καταλάβει τι έχετε κάνει.
2. Τοποθετείτε το άγκιστρο που κλείνει στην ίδια θέση με το αντίστοιχο άγκιστρο που ανοίγει ή τη δομή την οποία κλείνει. Με αυτόν τον τρόπο θα μπορείτε εύκολα να ελέγξετε αν έχουν τοποθετηθεί σωστά τα αντίστοιχα ζευγάρια.



## Άσκηση 01a. Αναβοσβήσιμο των 8 LED

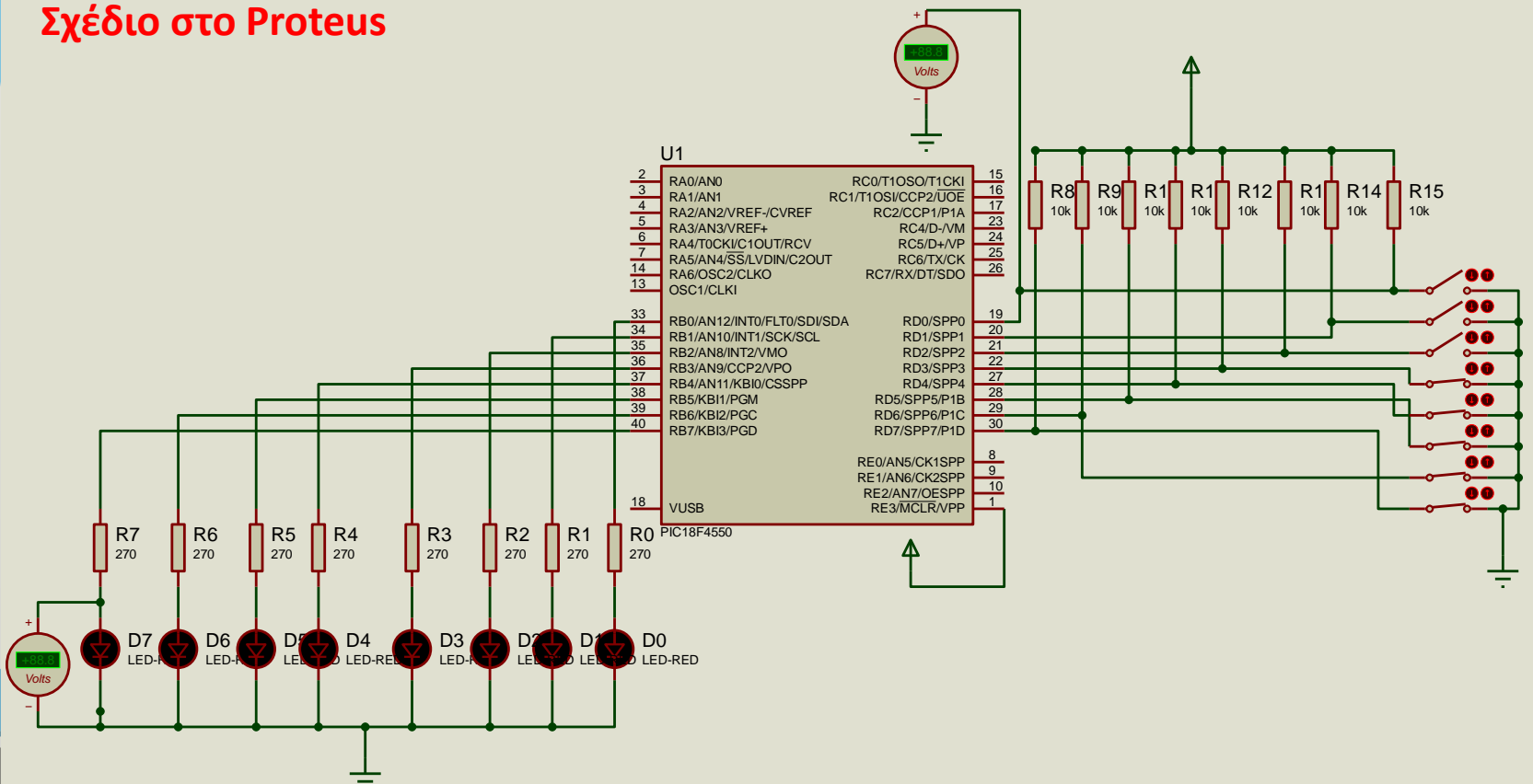
### Ερωτήσεις

1. Γράψτε μια εντολή με την οποία ο ακροδέκτης RB0, γίνεται είσοδος και οι ακροδέκτες RB1, RB2, RB3, RB4, RB5, RB6 και RB7 γίνονται έξοδοι.
2. Γράψτε μια εντολή με την οποία οι ακροδέκτες RD0, RD1, RD2 γίνονται είσοδοι και οι ακροδέκτες RD3, RD4, RD5, RD6, RD7 γίνονται έξοδοι.
3. Από το εγχειρίδιο (manual) του CCS Compiler βρείτε και γράψτε τη συνάρτηση που προκαλεί καθυστέρηση 200  $\mu$ s. Υπόδειξη: Manual  $\rightarrow$  Bulit-in Functions  $\rightarrow$  delay
4. Ποια εντολή θα κάνει όλη την πόρτα D του μικροελεγκτή είσοδο;
5. Πώς λέγεται ο καταχωρητής της πόρτας D (στα Ελληνικά και στα Αγγλικά) που καθορίζει ποιοι ακροδέκτες της πόρτας D θα είναι είσοδοι και ποιοι έξοδοι;
6. Σε ποια διεύθυνση είναι ο καταχωρητής κατεύθυνσης της πόρτας B του μικροελεγκτή PIC18F4550; Υπόδειξη: Θα το βρείτε στο εγχειρίδιο (manual) του PIC18F4550. Memory Organization  $\rightarrow$  Data Memory Organization  $\rightarrow$  Special Function Registers.
7. Σε ποια διεύθυνση είναι ο καταχωρητής δεδομένων της πόρτας B του μικροελεγκτή PIC18F4550; Υπόδειξη: Θα το βρείτε στο εγχειρίδιο (manual) του PIC18F4550. Memory Organization  $\rightarrow$  Data Memory Organization  $\rightarrow$  Special Function Registers.
8. Αν αντί για while(TRUE) γράψουμε while(3>2), τι θα αλλάξει στην εκτέλεση του προγράμματος;
9. Αναζητήστε στο εγχειρίδιο (manual) του CCS Compiler την εντολή set\_tris\_x() και διαβάστε τι κάνει. Υπόδειξη: Manual  $\rightarrow$  Bulit-in Functions  $\rightarrow$  set\_tris\_x()
10. Σε τι διαφέρει η εντολή set\_tris\_a(0x0F) από την εντολή set\_tris\_a(0b00001111);

## Άσκηση 01b. Χρήση των παράλληλων θυρών του μικροελεγκτή για ανάγνωση και εξαγωγή δεδομένων

Να γραφεί πρόγραμμα σε γλώσσα προγραμματισμού C για τον μικροελεγκτή PIC18F4550 που να διαβάζει έναν αριθμό από την πόρτα D, να προσθέτει σε αυτόν τον αριθμό των γραμμάτων του ονόματος σας και να αφαιρεί τον αριθμό των γραμμάτων του επιθέτου σας. Στη συνέχεια να εμφανίζει το αποτέλεσμα στην πόρτα B.

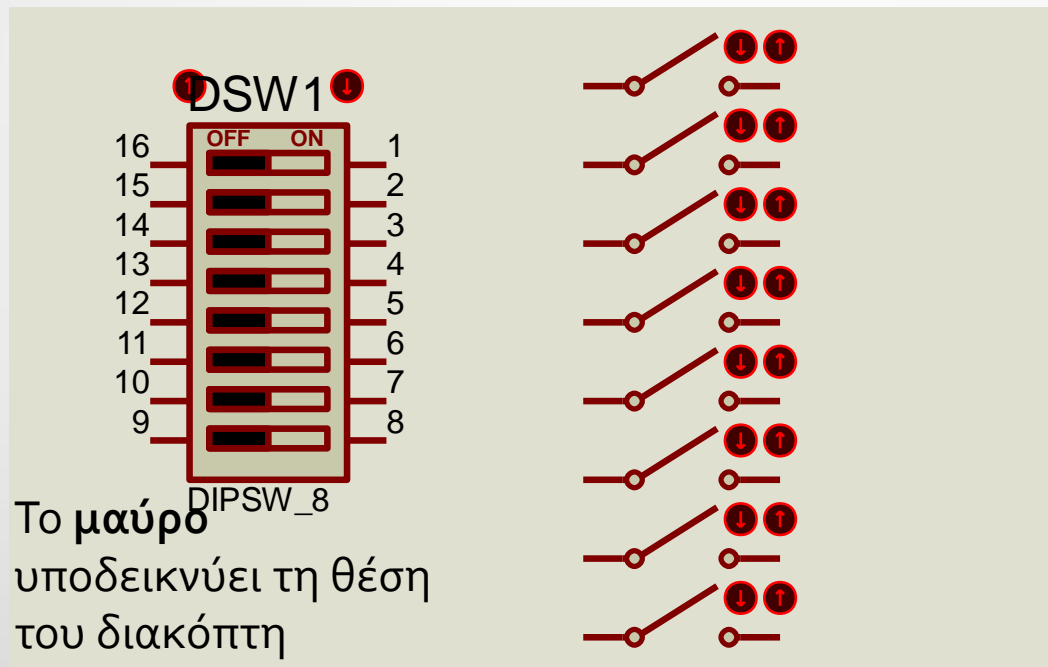
### Σχέδιο στο Proteus



# Σχέδια διακοπών στο Proteus (1/2)

Ισοδύναμα ηλεκτρονικά εξαρτήματα:

- Dipswitches
- Διακόπτες κατάστασης

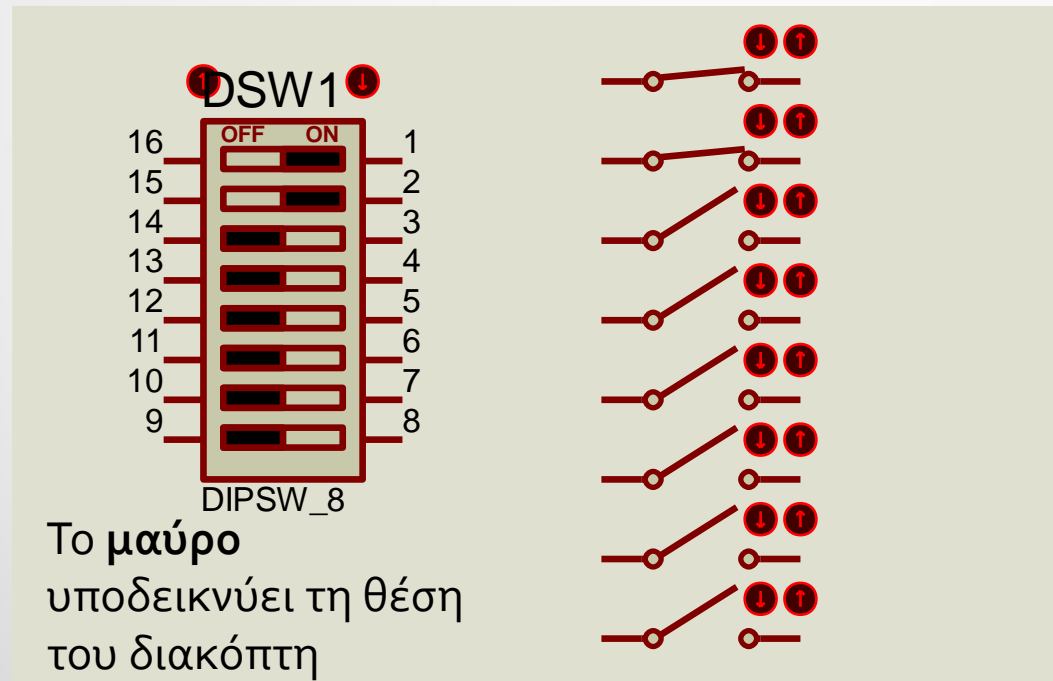


- Dipswitches: Όλα στο off
- Διακόπτες κατάστασης: Όλοι ανοιχτοί

# Σχέδια διακοπών στο Proteus (2/2)

Ισοδύναμα ηλεκτρονικά εξαρτήματα:

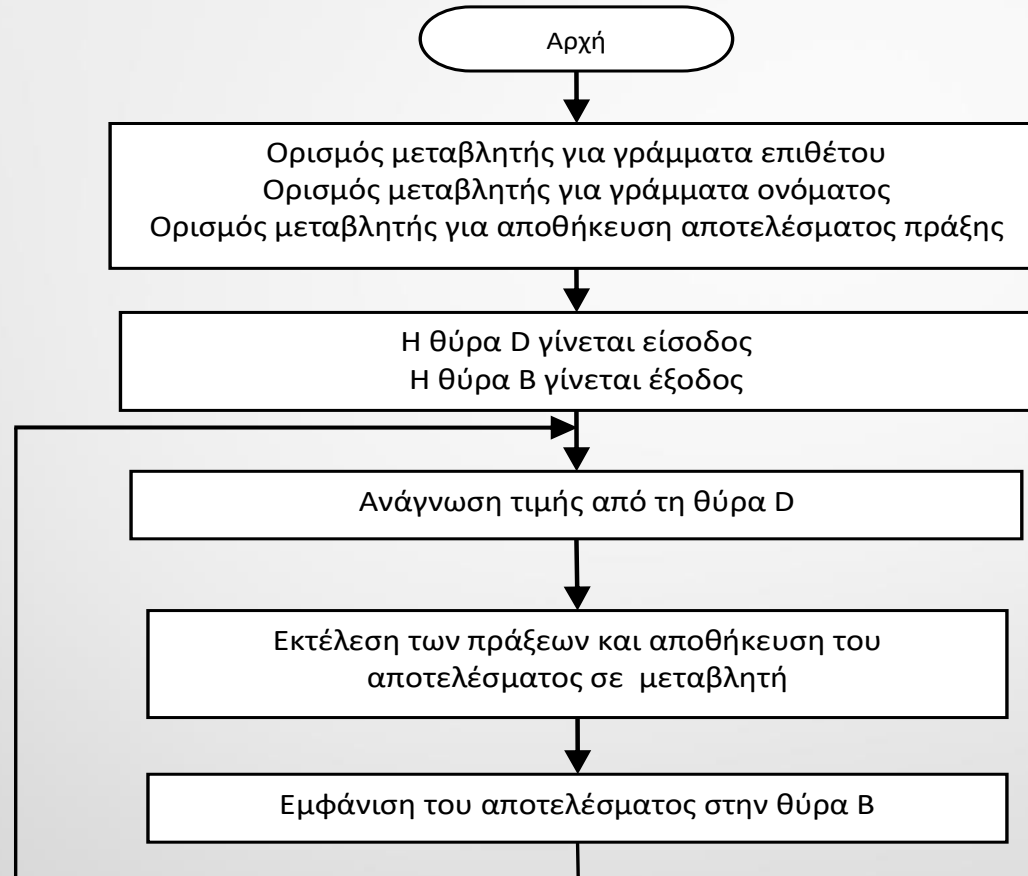
- Dipswitches
- Διακόπτες κατάστασης



- Dipswitches: Μόνο τα δύο πρώτα στο on
- Διακόπτες κατάστασης: Μόνο οι δύο πρώτοι κλειστοί

## Άσκηση 1b. Χρήση των παράλληλων θυρών του μικροελεγκτή για ανάγνωση και εξαγωγή δεδομένων

### Διάγραμμα ροής



**Άσκηση 01b.** Διαβάζεται μια τιμή από την πόρτα D, προστίθεται ο αριθμός των γραμμάτων του ονόματος, αφαιρείται ο αριθμός των γραμμάτων του επωνύμου. Το αποτέλεσμα αποστέλλεται στην πόρτα B.

## Πρόγραμμα σε C

```
#include<main.h>           // Το αρχείο <main.h> περιέχει αρχικές ρυθμίσεις
                             // Πρέπει να τοποθετηθεί οπωσδήποτε στον ίδιο φάκελο στον οποίο θα
                             // αναπτύξετε το project σας.

#byte PORTB=0xF81          // F81 είναι η θέση του καταχωρητή δεδομένων της πόρτας B
                             // στη μνήμη του μικροελεγκτή

#byte PORTD=0xF83          // F83 είναι η θέση του καταχωρητή δεδομένων της πόρτας D
                             // στη μνήμη του μικροελεγκτή

// *****Από εδώ αρχίζει το κύριο πρόγραμμα*****

void main()
{
    // Άνοιγμα αγκύλης της συνάρτησης main
    set_tris_b(0x00);        // Η θύρα B γίνεται έξοδος (καταχωρητής κατεύθυνσης=0000 0000)
    set_tris_d(0xff);        // Η θύρα D γίνεται είσοδος (καταχωρητής κατεύθυνσης=1111 1111)
    int a;                   // Ορισμός αέρας μεταβλητής a
    int onoma=5;              // Ορισμός αέρας μεταβλητής onoma. Απόδοση τιμής 5
    int eponymo=7;            // Ορισμός αέρας μεταβλητής eponymo. Απόδοση τιμής 7
    // Με την παρακάτω δομή while(TRUE){ } εκτελείται αενάως (για πάντα) το σύνολο των
    // εντολών που είναι μέσα στις αγκύλες. Η δεσμευμένη λέξη TRUE στη γλώσσα C αντιστοιχεί
    // στην αληθή συνθήκη. Αντί για TRUE θα μπορούσαμε για παράδειγμα να βάλουμε 5>1,
    // δηλαδή μια συνθήκη που ισχύει πάντα.
    while(TRUE) { // Βρόχος που δεν τελειώνει ποτέ(συνθήκη πάντα αληθής)
        a=PORTD+onoma-eponymo; // Υπολογισμός της ζητούμενης τιμής
        PORTB=a; // Μεταφορά της ζητούμενης τιμής στον καταχωρητή δεδομένων
                  // της πόρτας B
    } // κλείσιμο της αγκύλης while
    // κλείσιμο της αγκύλης main
}
```

# Μεταβλητές στον CCS C Compiler

## DATA DEFINITIONS



C Compiler

### Basic Types

Type-Specifier	Size	Range		
		Unsigned	Signed	Digits
int1	1 bit number	0 to 1	N/A	1/2
int8	8 bit number	0 to 255	-128 to 127	2-3
int16	16 bit number	0 to 65535	-32768 to 32767	4-5
int32	32 bit number	0 to 4294967295	-2147483648 to 2147483647	9-10
float32	32 bit float	$-1.5 \times 10^{45}$ to $3.4 \times 10^{38}$		7-8

C Standard Type	Default Type
short	int1
char	unsigned int8
int	int8
long	int16
long long	int32
float	float32

### Παραδείγματα

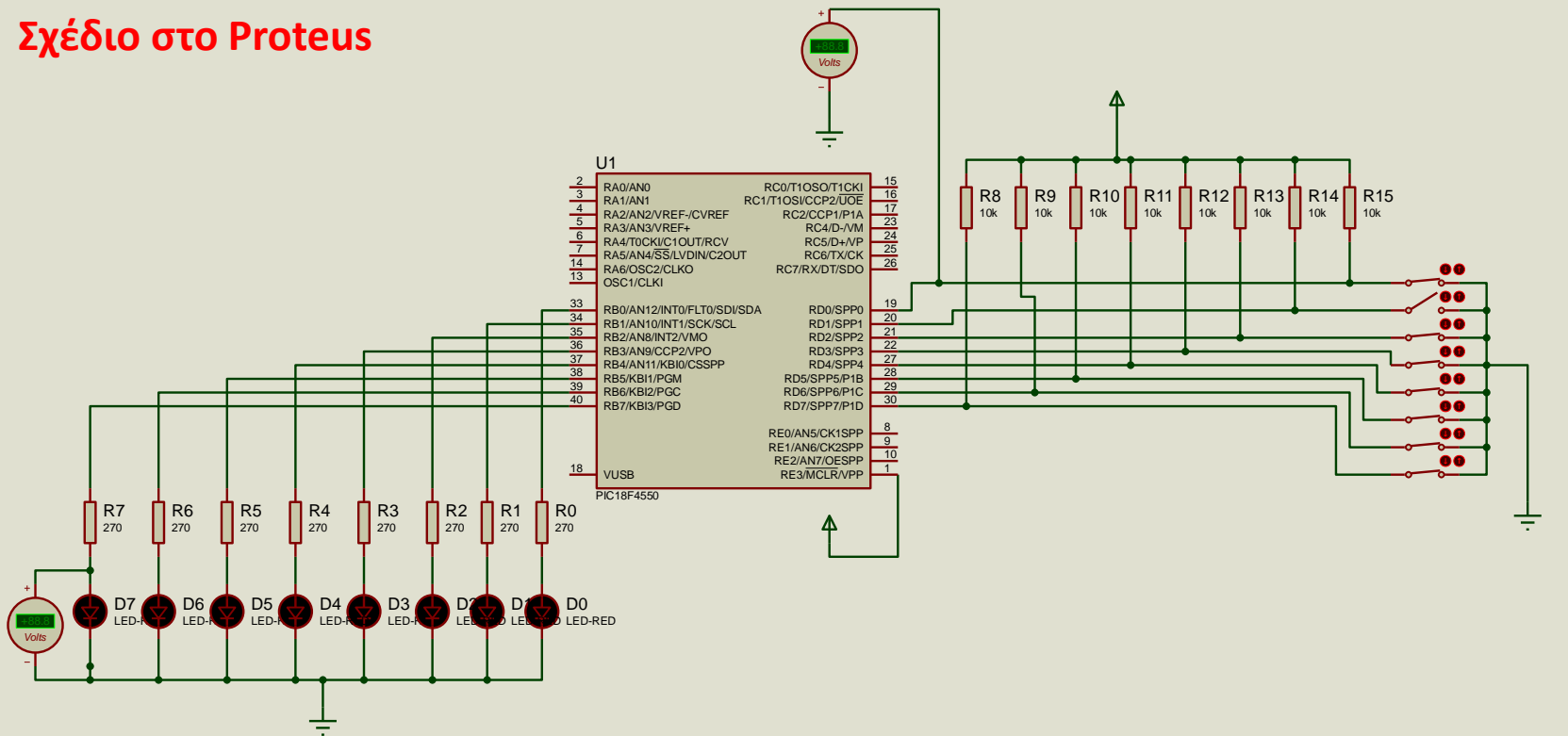
```
int1 a           // Μεταβλητή a του 1 bit
int8 kostas      // Μεταβλητή με όνομα kostas
                 // των 8 bit
int16 nik        // Μεταβλητή με όνομα nik
                 // των 16 bit
```



## Άσκηση 01c. Χρήση των παράλληλων θυρών του μικροελεγκτή για ανάγνωση και εξαγωγή δεδομένων.

Να γραφεί πρόγραμμα σε γλώσσα προγραμματισμού C για τον μικροελεγκτή PIC18F4550 που διαβάζει έναν αριθμό από την πόρτα D, τον διαιρεί με το 2 και το αποτέλεσμα εμφανίζεται στην πόρτα B.

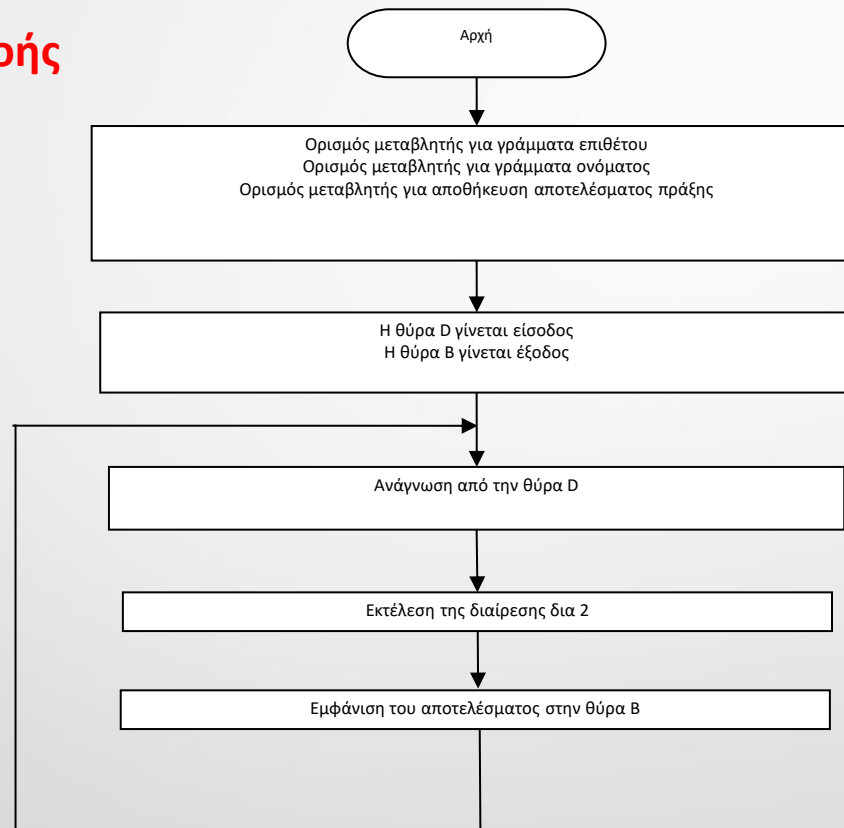
### Σχέδιο στο Proteus



## Άσκηση 01c. Χρήση των παράλληλων θυρών του μικροελεγκτή για ανάγνωση και εξαγωγή δεδομένων.

Να γραφεί πρόγραμμα σε γλώσσα προγραμματισμού C για τον μικροελεγκτή PIC18F4550 που διαβάζει έναν αριθμό από την πόρτα D, τον διαιρεί με το 2 και το αποτέλεσμα εμφανίζεται στην πόρτα B.

### Διάγραμμα Ροής



## Άσκηση 01c. Χρήση των παράλληλων θυρών του μικροελεγκτή για ανάγνωση και εξαγωγή δεδομένων.

Να γραφεί πρόγραμμα σε γλώσσα προγραμματισμού C για τον μικροελεγκτή PIC18F4550 που διαβάζει από την πόρτα D έναν αριθμό, τον διαιρεί με το 2 και το αποτέλεσμα εμφανίζεται στην πόρτα B.

### Πρόγραμμα σε C

```
#include<main.h> // Το αρχείο <main.h> περιέχει αρχικές ρυθμίσεις
                  // Πρέπει να τοποθετηθεί οπωσδήποτε στον ίδιο φάκελο στον οποίο θα
                  // αναπτύξετε το project σας.

#byte PORTB=0xF81 // F81 είναι η θέση τη καταχωρητή δεδομένων της πόρτας B
                  // στη μνήμη του μικροελεγκτή
#byte PORTD=0xF83 // F83 είναι η θέση του καταχωρητή δεδομένων της πόρτας D
                  // στη μνήμη του μικροελεγκτή
// *****Από εδώ αρχίζει το κύριο πρόγραμμα*****
void main()
{
    // Άνοιγμα αγκύλης της συνάρτησης main
    set_tris_b(0x00); // Η θύρα B γίνεται έξοδος(καταχωρητής κατεύθυνσης=0000 0000)
    set_tris_d(0xff); // Η θύρα D γίνεται είσοδος(καταχωρητής κατεύθυνσης=1111 1111)

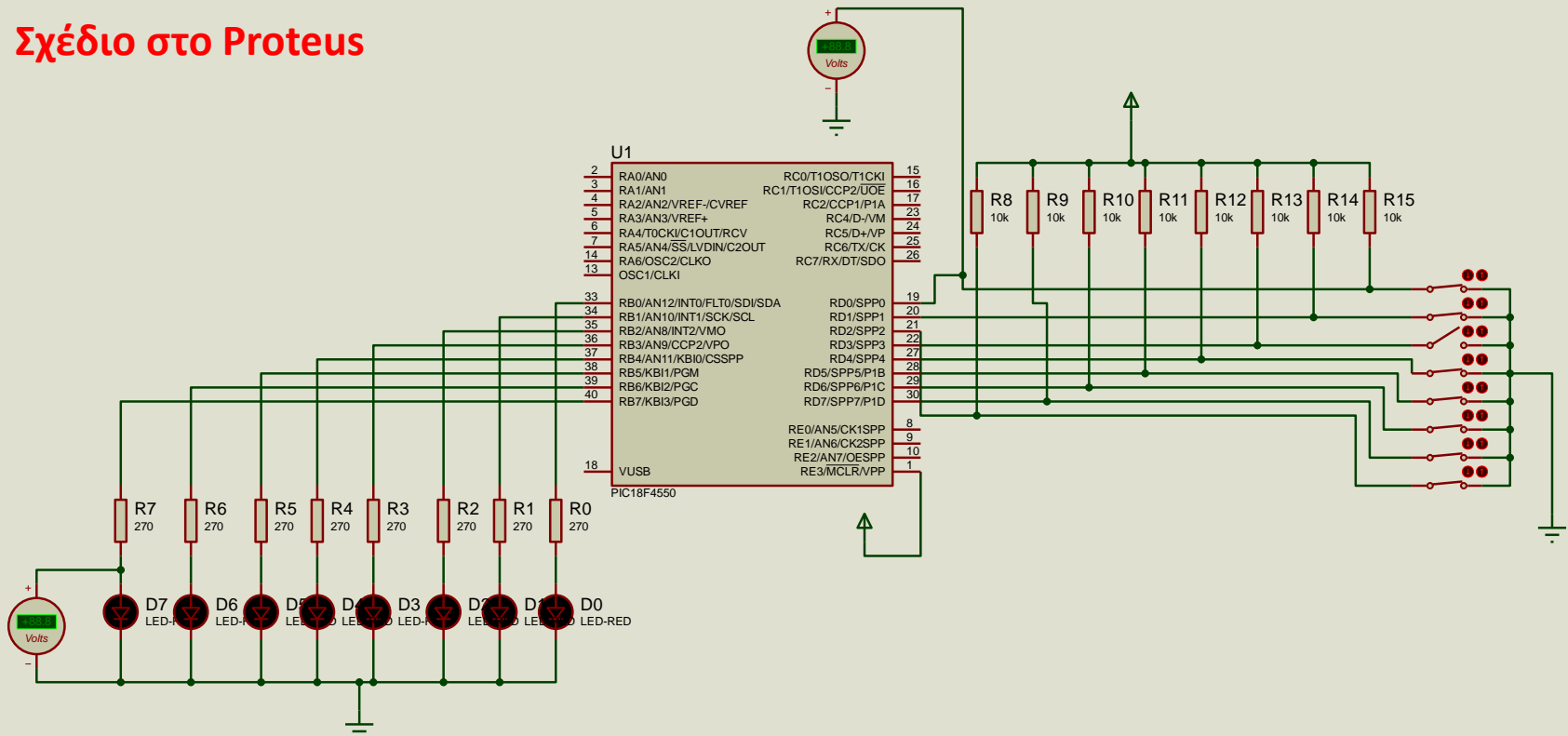
    int8 a; // Ορισμός ακέραιας μεταβλητής a
            // Με την παρακάτω δομή while(TRUE){} εκτελείται αενάως (για πάντα) το σύνολο των
            // εντολών που είναι μέσα στις αγκύλες.
            // Η δεσμευμένη λέξη TRUE στη γλώσσα C αντιστοιχεί στην αληθή συνθήκη.
            // Αντί για TRUE θα μπορούσαμε για παράδειγμα να βάλουμε 5>1,
            // δηλαδή μια συνθήκη που ισχύει πάντα.

    while(TRUE) { // Βρόχος που δεν τελειώνει ποτέ(συνθήκη πάντα αληθής)
        a=PORTD; // Μεταφορά του περιεχομένου της πόρτας D στην μεταβλητή a
        PORTB=a/2; // Διαίρεση δια 2 και εμφάνιση του αποτελέσματος στην πόρτα B
    } // Κλείσιμο της αγκύλης while
} // Κλείσιμο της αγκύλης main
```

## Άσκηση 01d. Επιλογή προγράμματος αναβοσβήσιματος

Διαφορετικό πρόγραμμα “άναμμα και σβήσιμο” των 8 led που είναι συνδεδεμένα στην πόρτα B (έξοδος), ανάλογα με την τιμή της πόρτας D (είσοδος).

### Σχέδιο στο Proteus



## Άσκηση 01d. Επιλογή προγράμματος αναβοσβήσιματος

### Πρόγραμμα σε γλώσσα C (έχουν παραληφθεί οι αρχικές ρυθμίσεις)

```
while(TRUE) { // Βρόχος που δεν τελειώνει ποτέ(συνθήκη πάντα αληθής)
    a=PORTD;    // Ανάγνωση του περιεχομένου της πόρτας D και μεταφορά του στη μεταβλητή a
    switch (a){
        case 0: PORTB=0xFF; // Πρόγραμμα αναβοσβήσιματος όταν a=0 (PORTD = 0000 0000)
            delay_ms(100);
            PORTB=0x00;
            delay_ms(100);
            break;
        case 1: PORTB=0b11000011; // Πρόγραμμα αναβοσβήσιματος όταν a=1 (PORTD = 0000 0001)
            delay_ms(100);
            PORTB=0b00111100;
            delay_ms(100);
            break;
        case 2: PORTB=0b10101010; // Πρόγραμμα αναβοσβήσιματος όταν a=2 (PORTD = 0000 0010)
            delay_ms(100);
            PORTB=0b01010101;
            delay_ms(100);
            break;
        case 3: PORTB=0b10000000; // Πρόγραμμα αναβοσβήσιματος όταν a=3 (PORTD = 0000 0011)
            for(i=1;i<=7;i++){
                delay_ms(50);
                PORTB=PORTB/2;
            }
            delay_ms(50);
            break;
        case 4: PORTB=0xF0; // Πρόγραμμα αναβοσβήσιματος όταν a=4 (PORTD = 0000 0100)
            delay_ms(100);
            PORTB=0x0F;
            delay_ms(100);
            break;
    } // Κλείσιμο switch
} // Κλείσιμο while
} // Κλείσιμο main
```