

# ENGINE



Erasmus+

TEACHING ONLINE ELECTRONICS, MICROCONTROLLERS AND PROGRAMMING  
IN HIGHER EDUCATION

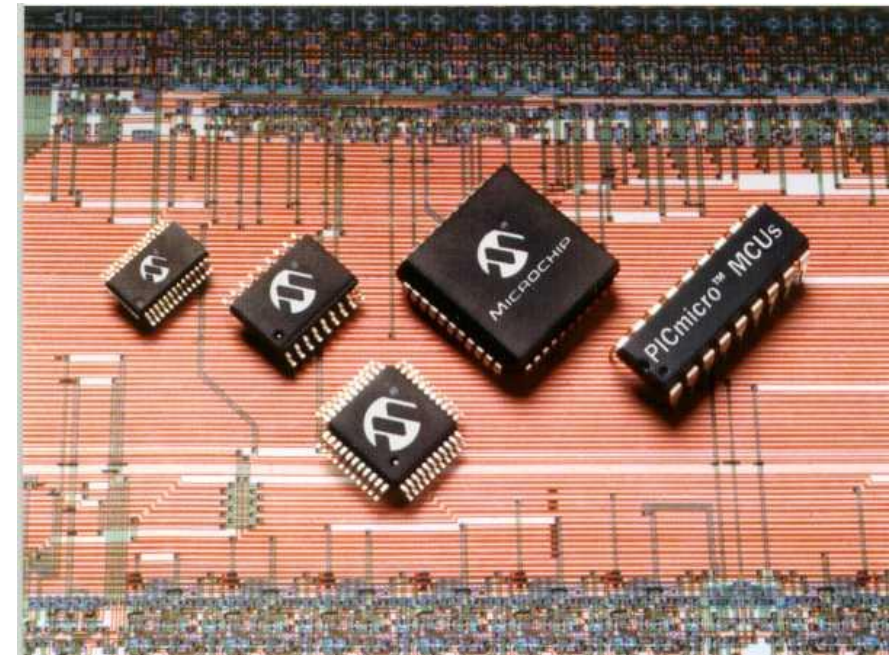
---

## Module\_2-5. Keypad 4x4

*PIC18F4550 with Proteus Simulation*

# Contents

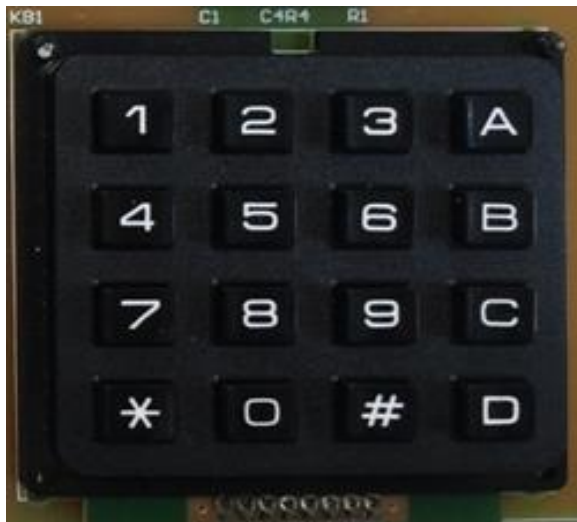
- Keypad layout
- Keypad connections
- CCS C Compiler: Keypad driver
- Example



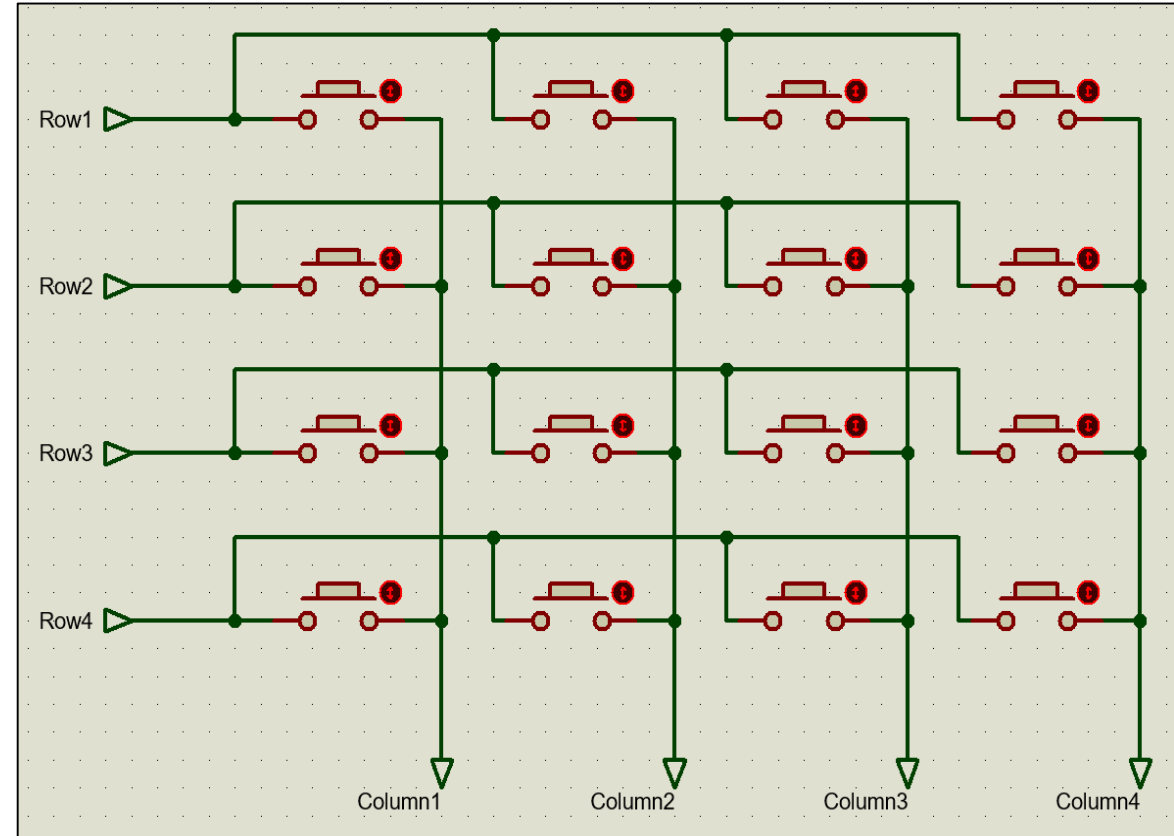
## Module\_2-5. Keypad 4x4 Layout

A keypad 4x4, consists of:

- 4 columns
- 4 rows
- 16 keys



Keypad 4x4



Pinout of Keypad 4x4

<https://www.electronicwings.com/sensors-modules/4x4-keypad-module/>

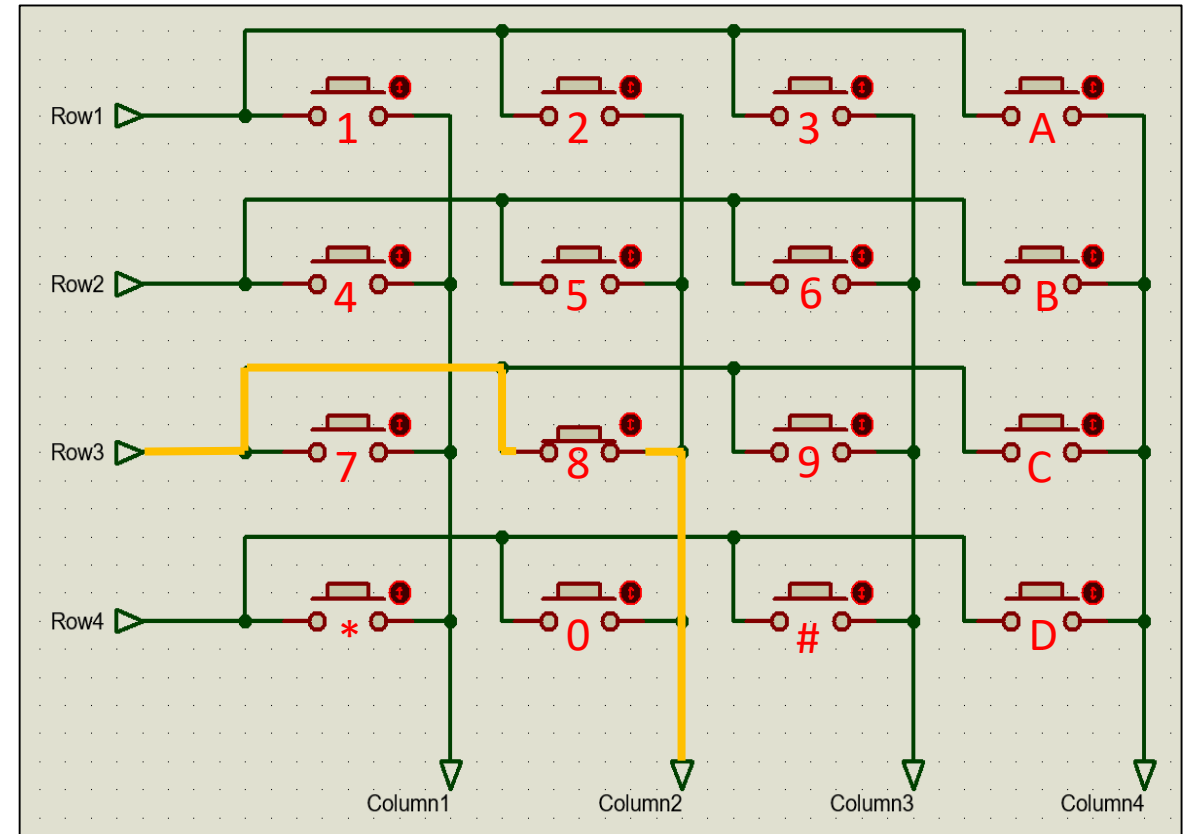
## Module\_2-5. Keypad 4x4 Connections

The keypad rows go to the input pins of the PIC18F4550 with pull-up resistors.

Keypad columns go to PIC18F4550 output pins.

**The microcontroller sequential signals to the columns, and reads the rows.**

For example: if 8 is pressed, row 3 will be connected with column 2.



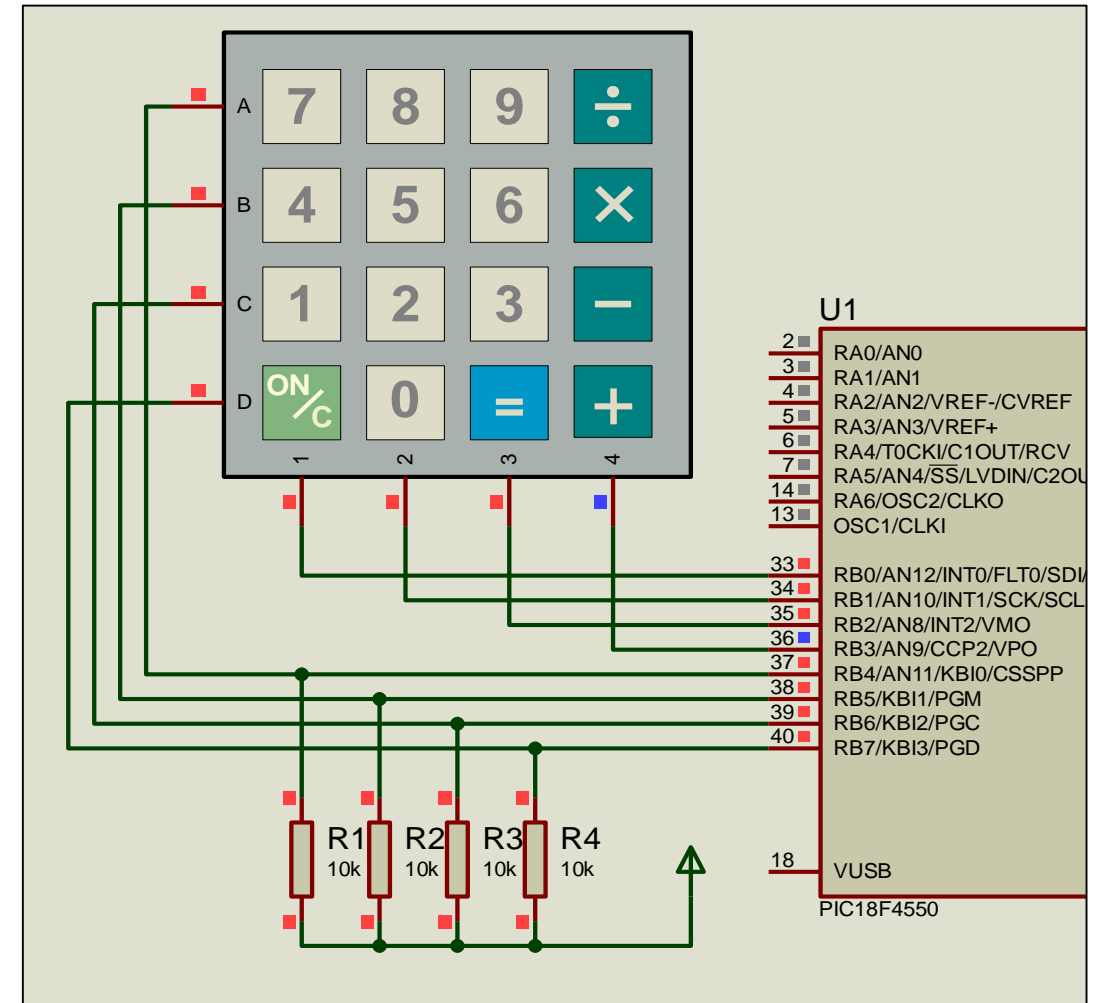
Pinout of Keypad 4x4

# Module\_2-5. Keypad 4x4 Connections

The keypad rows go to the input pins (RB4~RB7) of the PIC18F4550 with pull-up resistors.

Keypad columns go to PIC18F4550 output pins (RB0~RB3).

The microcontroller sequential signals to the columns, and reads the rows.



Pinout of Keypad 4x4

# Module\_2-5. Keypad 4x4 CCS C Compiler: driver

Columns: RB0~RB3  
Rows: RB4~RB7

Keys/characters definition

**kbd\_init();**

Pins are defined as inputs/outputs

```
keypad.h
1 //Keypad connection:
2 #define row0 PIN_b4
3 #define row1 PIN_b5
4 #define row2 PIN_b6
5 #define row3 PIN_b7
6 #define col0 PIN_b0
7 #define col1 PIN_b1
8 #define col2 PIN_b2
9 #define col3 PIN_b3
10
11 //----- Variable definition-----
12 char const KEYS[4][4] =
13 { {'7','8','9','/'},
14   {'4','5','6','X'},
15   {'1','2','3','-'},
16   {'C','0','=','+'}};
17
18
19 #define KBD_DEBOUNCE_FACTOR 200 // Set this number to apx n/333 where
20 // n is the number of times you expect
21 // to call kbd_getc each second
22
23 //-----Function Definition-----
24 short int ALL_ROWS (void);
25 void kbd_init(void);
26 char kbd_getc(void);
27 //-----
28 void kbd_init(void)
29 {
30   set_tris_b(0xF0);
31 }
32 }
```

# Module\_2-5. Keypad 4x4 CCS C Compiler: driver

**k = kbd\_getc ();**

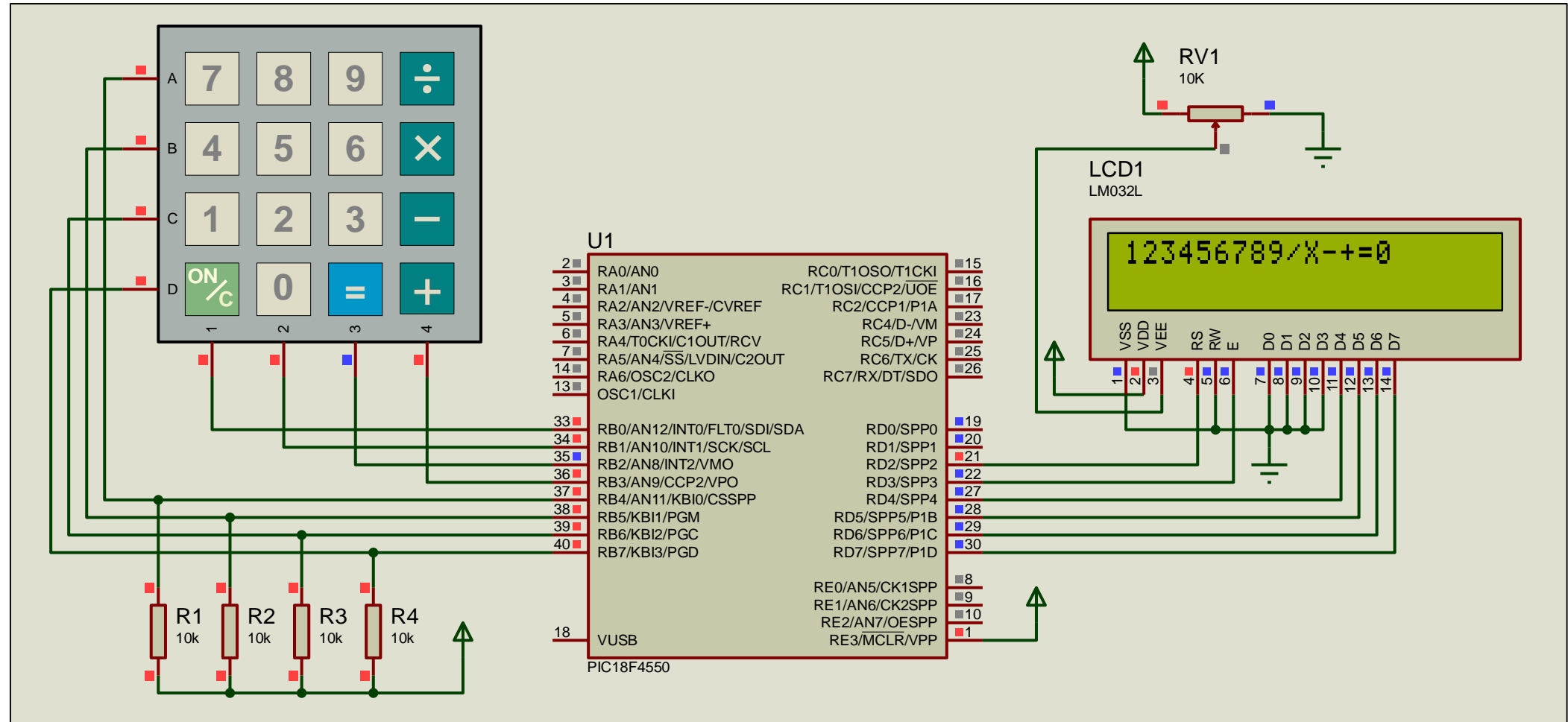
1. When a key is pressed it returns to the variable k the ASCII code of the key pressed.
2. When no key is pressed it returns the value 0 to the variable k.

```
keypad.h
1 //Keypad connection:
2 #define row0 PIN_b4
3 #define row1 PIN_b5
4 #define row2 PIN_b6
5 #define row3 PIN_b7
6 #define col0 PIN_b0
7 #define col1 PIN_b1
8 #define col2 PIN_b2
9 #define col3 PIN_b3
10
11 //----- Variable definition-----
12 char const KEYS[4][4] =
13 {
14     {'7', '8', '9', '/'},
15     {'4', '5', '6', 'X'},
16     {'1', '2', '3', '-'},
17     {'C', '0', '=', '+'}};
18
19 #define KBD_DEBOUNCE_FACTOR 200 // Set this number to apx n/333 where
20 // n is the number of times you expect
21 // to call kbd_getc each second
22
23 //-----Function Definition-----
24 short int ALL_ROWS (void);
25 void kbd_init(void);
26 char kbd_getc(void);
27 //-----
28 void kbd_init(void)
29 {
30     set_tris_b(0xF0);
31 }
32 }
```

# Module\_2-5. Keypad 4x4 Example

The purpose of this activity is for microcontroller to read an ASCII character from a keypad 4x4, and display it in a LCD 16x2.

- If 'C' key is pressed, then the screen clears.





# Module\_2-5. Keypad 4x4 Example

```
#include <main.h> // the file main.h with the
// initial settings is included.
// This file must be placed in the same
// folder with the project.
// Also the 18F4550.h file must exist
// in the same folder with the project

#include <flex_lcd.h> // The h file of the lcd driver
// should be in the same folder where we will save our program.
// The #define LCD_DB4 PIN_B4 etc statements in flex_lcd.c
// should be checked and possibly modified.
// These statements determine the pins of the microcontroller
// that are connected to LCD 16x2.

#include <keypad.h> // The h file of the keypad driver
// should be in the same folder where we will save our program.
// The #define row0 PIN_B4 etc statements in keypad.h
// should be checked and possibly modified.
// These statements determine the pins of the microcontroller
// that are connected to the keypad 4x4.

#byte PORTB =0xF81
// We attribute to the memory position 0xF81 the name PORTB.
// This means that we define a 8 bit variable whose value
// will be stored to the memory position F81h.
// The memory position F81h is the PORTD data register.

#byte PORTD=0xF83 // F83h is the position or PORTD data register
// at the data memory of the microcontroller
// SFR Special Function Register

//initialization routine
void init(void);
```

```
// ***** main program *****
void main(){
    char k; //variable for storing the ASCII code of the key pressed
    init(); //initialization routine
    kbd_init(); //initialization routine for the keypad 4x4
    lcd_init(); //initialization routine for the LCD 16x2
    while(TRUE){
        k=kbd_getc(); //keypad reading

        //If a key is pressed (k!= 0) and if the key pressed is not 'C',
        //the character will appear on the LCD
        if (k!=0 && k!='C'){
            printf(lcd_putc,"%c",k);
        }

        //If 'C' key is pressed then the screen clears
        else if(k=='C'){
            printf(lcd_putc,"\f");
        }
    }
}

// initialization routine
void init(void){
    set_tris_d(0x00); //PORTD is defined as output
}
```

# Module\_2-5. Keypad 4x4 Example

```

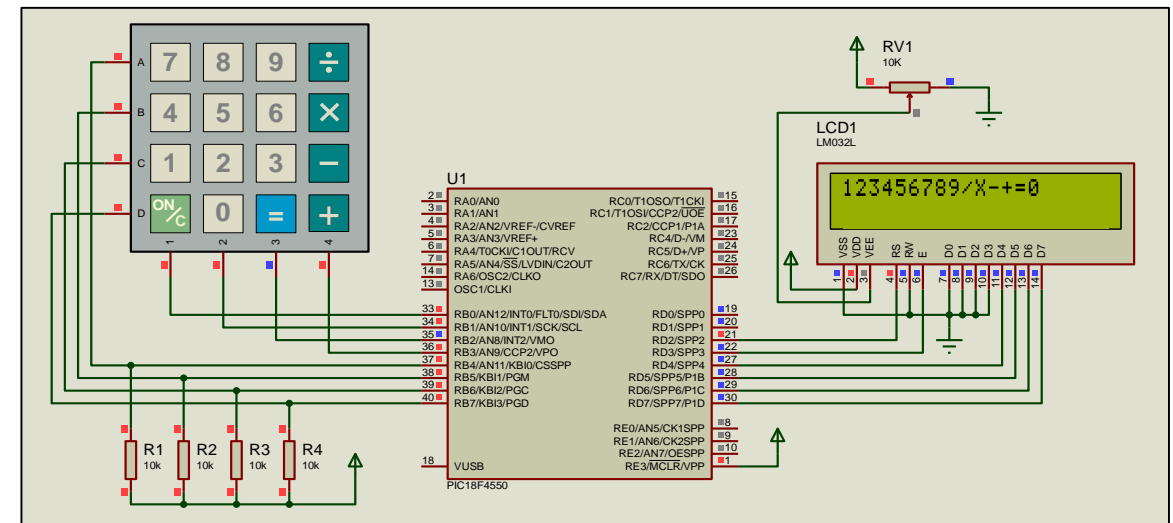
keypad.h flex_lcd.h
1 //Keypad connection:
2 #define row0 PIN_B4
3 #define row1 PIN_B5
4 #define row2 PIN_B6
5 #define row3 PIN_B7
6 #define col0 PIN_B0
7 #define col1 PIN_B1
8 #define col2 PIN_B2
9 #define col3 PIN_B3
10
11 //----- Variable definition-----
12 char const KEYS[4][4] =
13 {{ '7', '8', '9', '/' },
14  { '4', '5', '6', 'X' },
15  { '1', '2', '3', '-' },
16  { 'C', '0', '=', '+' } };
17

```

```

keypad.h flex_lcd.h
1 //-----
2 #define LCD_DB4 PIN_D4
3 #define LCD_DB5 PIN_D5
4 #define LCD_DB6 PIN_D6
5 #define LCD_DB7 PIN_D7
6
7 #define LCD_E PIN_D3
8 #define LCD_RS PIN_D2
9 // #define LCD_RW PIN_D1
10 #define lcd_type 2 // 0=5x7, 1=5x10, 2=2 lines
11 #define lcd_line_two 0x40 // LCD RAM address for the 2nd line

```

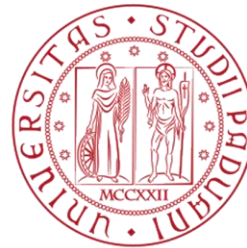


## ENGINE Partnership

- Warsaw University of Technology (PL) - *coordinator*
- IHU - International Hellenic University (GR)
- EDUMOTIVA - European Lab for Educational Technology (GR)
- University of Padova (IT)
- University of Applied Sciences in Tarnow (PL)



INTERNATIONAL  
HELLENIC  
UNIVERSITY



UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA



## CONTACT:



[www.engined.eu](http://www.engined.eu)



[angelika.tefelska@pw.edu.pl](mailto:angelika.tefelska@pw.edu.pl)



[@projectENGINE1](https://twitter.com/projectENGINE1)



[@EUprojectEngine](https://www.facebook.com/EUprojectEngine)



# Erasmus+

This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.