

MICROPROCESSORS AND PROGRAMMING

LABORATORY WORK TASK NR. 1

MCU PORTS

GOAL: Find out how the ports of the PIC16F1518 microcontroller are controlled.

TASK: According to your selected variant number, create and connect the circuit and program the microcontroller so that the LED connected to it lights up. When the button is pressed, the LED should go out and light up again only when the button is released. Use the microcontroller's internal pull-up resistors for the button circuit.

EXTRA TASK: Understand the operation of the special register STATUS.

REPORT CONTENTS:

- ☐ Goal and task;
- ☐ Task variant data;
- ☐ Circuit diagram corresponding to task variant;
- ☐ Code with comments;
- ☐ Program algorithm;
- ☐ Conclusions.

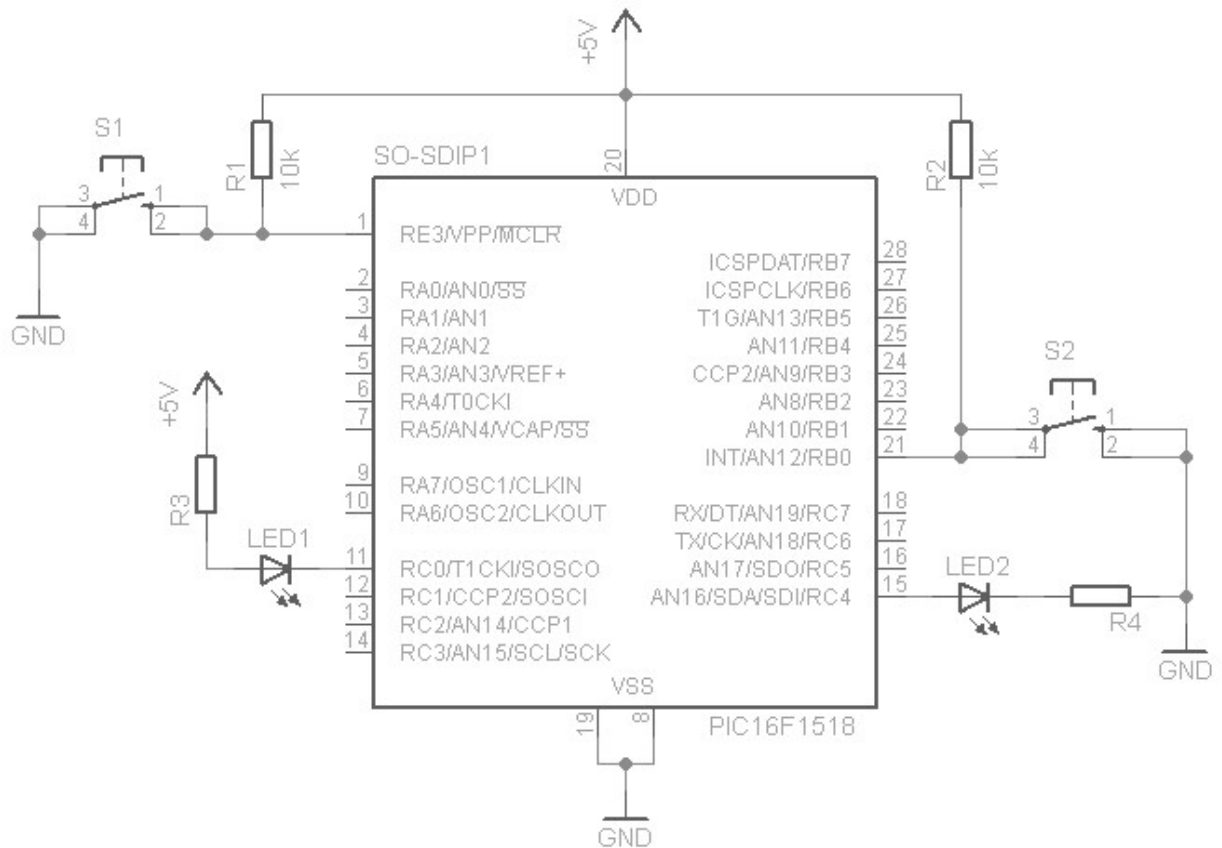
VARIANT LIST (select one):

Nr.	LED anode connected to	LED cathode connected to	Pushbutton connected to
1.	PORTB.4	VSS (GND)	PORTB.0
2.	PORTB.5	VSS (GND)	PORTB.1
3.	PORTB.6	VSS (GND)	PORTB.2
4.	PORTB.7	VSS (GND)	PORTB.3
5.	PORTB.1	VSS (GND)	PORTB.4
6.	PORTB.2	VSS (GND)	PORTB.5
7.	PORTB.3	VSS (GND)	PORTB.6
8.	PORTB.4	VSS (GND)	PORTB.7
9.	PORTC.0	VSS (GND)	PORTB.0
10.	PORTC.1	VSS (GND)	PORTB.1
11.	PORTC.2	VSS (GND)	PORTB.2
12.	PORTC.3	VSS (GND)	PORTB.3
13.	PORTC.4	VSS (GND)	PORTB.4
14.	PORTC.5	VSS (GND)	PORTB.5
15.	PORTC.6	VSS (GND)	PORTB.6
16.	PORTC.7	VSS (GND)	PORTB.7
17.	VDD (+5V)	PORTB.2	PORTB.0
18.	VDD (+5V)	PORTB.3	PORTB.1
19.	VDD (+5V)	PORTB.0	PORTB.2
20.	VDD (+5V)	PORTB.1	PORTB.3
21.	VDD (+5V)	PORTB.6	PORTB.4
22.	VDD (+5V)	PORTB.7	PORTB.5
23.	VDD (+5V)	PORTB.4	PORTB.6
24.	VDD (+5V)	PORTB.5	PORTB.7
25.	VDD (+5V)	PORTC.0	PORTB.0
26.	VDD (+5V)	PORTC.1	PORTB.1
27.	VDD (+5V)	PORTC.2	PORTB.2
28.	VDD (+5V)	PORTC.3	PORTB.3
29.	VDD (+5V)	PORTC.4	PORTB.4
30.	VDD (+5V)	PORTC.5	PORTB.5

ŠIAULIŲ VALSTYBINĖ KOLEGIJA
VERSLO IR TECHNOLOGIJŲ FAKULTETAS
TRANSPORTO INŽINERIJOS KATEDRA

31.	VDD (+5V)	PORTC.6	PORTB.6
32.	VDD (+5V)	PORTC.7	PORTB.7

EXAMPLES OF CIRCUIT CONNECTION SOLUTIONS (YOU NEED TO ALTER ACCORDING TO YOUR TASK VARIANT)



EXAMPLES OF CODE:

```
#include <stdio.h>
#include <stdlib.h>
#include <xc.h>

// CONFIG1
#pragma config FOSC = INTOSC
#pragma config WDTE = OFF
#pragma config PWRTE = ON
#pragma config MCLRE = ON
#pragma config CP = OFF
#pragma config BOREN = OFF
#pragma config CLKOUTEN = OFF
#pragma config IESO = OFF
#pragma config FCMEN = OFF
// CONFIG2
#pragma config WRT = OFF
#pragma config VCAPEN = OFF
#pragma config STVREN = ON
#pragma config BORV = LO
#pragma config LPBOR = OFF
#pragma config LVP = OFF
```

ŠIAULIŲ VALSTYBINĖ KOLEGIJA
VERSLO IR TECHNOLOGIJŲ FAKULTETAS
TRANSPORTO INŽINERIJOS KATEDRA

```
int main() {  
    ANSELB=0; //all pins of PORTB are digital  
    TRISB=0b00001100; //PORTB.2 and RB3 – inputs. All other pins of PORTB - outputs  
    TRISA=0; //PORTA – all pins of PORTA are outputs  
  
    nWPUEN=0; //pull-up resistors are enabled  
    WPUB2=1; // pull-up resistor of PORTB.2 is activated  
  
    PORTA=0xFF; //all PORTA pins output high logic level (+5V)  
  
    if(RB2==1){ //if PORTB.2 input logic level is high, then,...  
        PORTA=0; //all PORTA output pins are set to low logic level (GND)  
    }else { //otherwise  
        RA0=1; // PORTA.0 pin level is set to high (+5V)  
    }  
  
    while(1){ //while 1==true (1 is always true)...  
        //do nothing.... endless loop  
    }  
}
```