ENGINE

Teaching online electronics, microcontrollers and programming in Higher Education

Output 2: Online Course for Microcontrollers: syllabus, open educational resources

Practice leaflet: Module_2-2 pins as inputs

Lead Partner: International Hellenic University (IHU)

Authors: Theodosios Sapounidis [IHU], Aristotelis Kazakopoulos [IHU], Aggelos Giakoumis [IHU], Sokratis Tselegkaridis [IHU]

Erasmus+

GRANT NUMBER: 2020-1-PL01-KA226-HE-095653

Declaration

This report has been prepared in the context of the ENGINE project. Where other published and unpublished source materials have been used, these have been acknowledged.

Copyright

Funding Disclaimer

This project has been funded with support from the European Commission. This report 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.

Table of Contents

Executive summary

In this Module we will use PIC18F4550 parallel ports as inputs.

Chapter 1: **Overview**

Table 1. Overview

Title / short summary	**2. Use of microcontroller parallel ports as inputs**
Expected learning outcomes	• The student will be able to define the pins of a microcontroller parallel port as inputs or outputs by using commands in C language • The student will be able to connect switches and pull up resistors to parallel port and use it as an input • The student will learn how to read a whole 8-bit word from a parallel input port • The student will learn how to read the value of a pin of a parallel port that has been defined as an input. • The student will learn how to check the value of a specific bit of a parallel input port • The student will learn to select a different program based on the value given to an input parallel port
Keywords	Direction Register, Data Register, Parallel Port, LED, switch (a){}, input(PIN_A0), bit_test(PORTD,i)

Duration	The duration of the module_2-2 is 3 hours • Presentation of the module_2-2 by the teacher, 30 minutes • 1st activity, read a binary value from PORTD and send to the PORTB the result of the read value divided by 2, 1 hour • 2nd activity, program the microcontroller so that the LEDs connected to PORTB turn ON and OFF in accordance with a program selected with the switches connected on PORTD, 30 minutes • 3rd activity, read the value of pin A_0 and the value of pin A_1 and turn ON an LED only when A0=1 and A1=1 (Logical function AND between A0 and A1), 30 minutes • 4th activity, count the 1s of PORTD and show the result at output PORTB, 30 minutes
Involved	**The teacher:** Presents the slides associated with the module_2-2 and answers question **The students:** Draw circuits in Proteus Schematic, write programs in C language, load programs to a microcontroller and run the simulation using the Proteus Design Suite
Assignment	At the end of the Module_2-2 will be given: • Open Project
Educational tools and equipment	• **Material:** PC • **Software:** CCS C compiler, Proteus Design Suite

Prerequisites / pre-existing knowledge	• The student must know the characteristic of a LED and how to connect it to a DC source (link1) • The student must know the use of the Direction Register and the Data Register of a microcontroller parallel port *PIC18F4550 datasheet (I/O ports, p. 113)* • The student must know the commands of CCS C Compiler associated with the use of parallel ports *CCS C Compiler Manual (Built In Functions, DISCRETE I/O p. 159, set_tris_x() command p.244)* • The student must be familiarized with the Proteus Design Suite (link2)
Educational content	• CCS C Compiler manual (C Compiler Reference Manual) • MICROCHIP, PIC18F2455/2550/4455/4550 Data Sheet • Module_2-2 slides • Module_2-2 Evaluation leaflet • Module_2-2 Open project leaflet • Module_2-2 Programs, Schematic Proteus (Compressed folder)
Tips	***Tip1.*** *Some devices have polarity, for instance the LED, they must be connected in the right way.* ***Tip2.*** *The program must include the main.h and the 18F4550.h files. These files must be in the same folder with your project.*

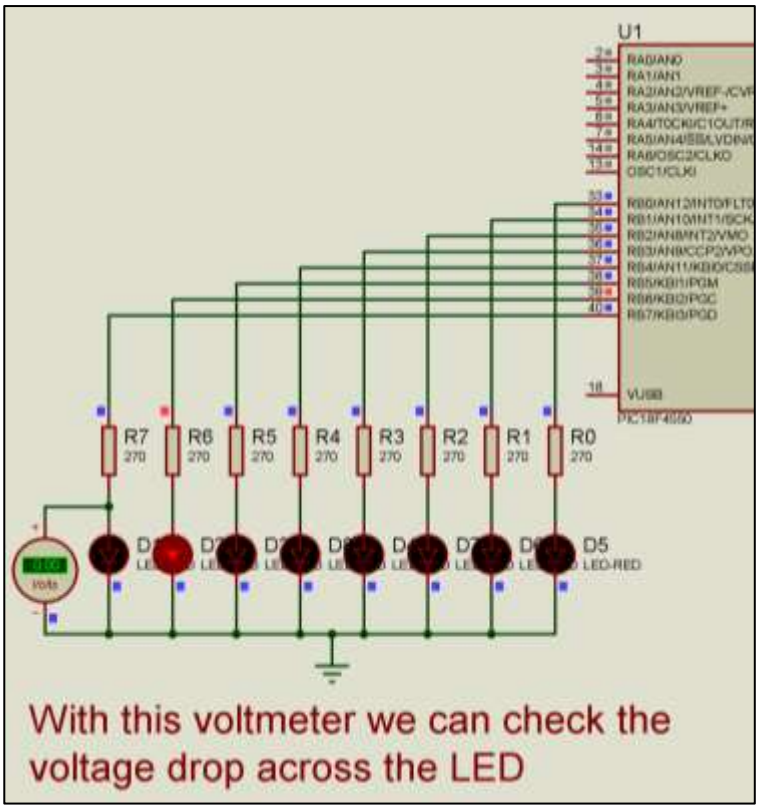
Chapter 2: **Activities**

2.1 Activity 1. Read a binary word from PORTD

The purpose of this activity is to read a binary value from PORTD and send to the PORTB the result of the read value divided by 2.

Table 2. Activity 1

Activity 1st (1 hour)	**Step 1.** The circuit is drawn in the Proteus Design Suite. In this step 8 LEDs are connected to the PORTB parallel port. 8 switches are connected to PORTD and 8 pull up resistors. **Step 2.** The values of the resistors are calculated so that the current through the LEDs is 10mA. It is accepted that the voltage drop across the animated red LEDs is 2.2 V. **Step 3.** The program in C language is written. **Step 4.** The program is compiled with the use of CCS C compiler to the microcontroller machine code. **Step 5.** The machine code is loaded to the microcontroller. **Step 6.** The animation is activated. **Step 7.** We form a binary value by closing and opening the switches connected to PORTD that has been defined as input port. **Step 8.** We check that the value of PORTB, is equal to the half to the value of PORTD **Step 7.** Modifications and discussion.

Step 1 (20 minutes)	Draw the circuit of the picture in the Proteus Design Suite.  *Figure 1(a). PORTD connected as input and PORTB as output*

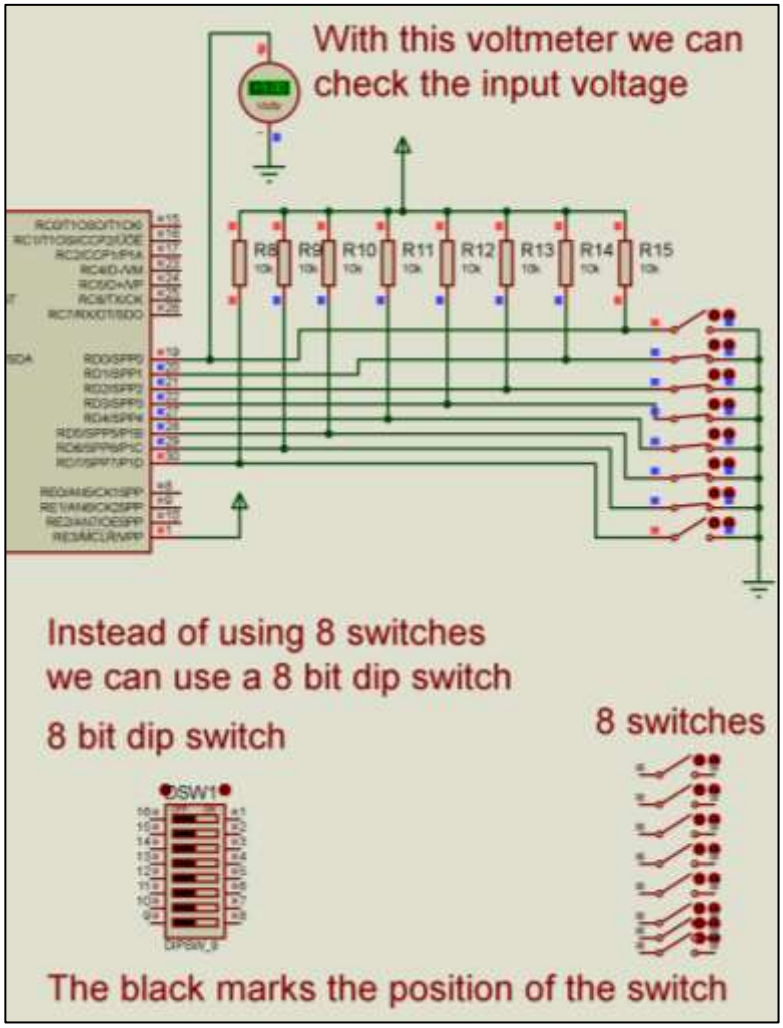
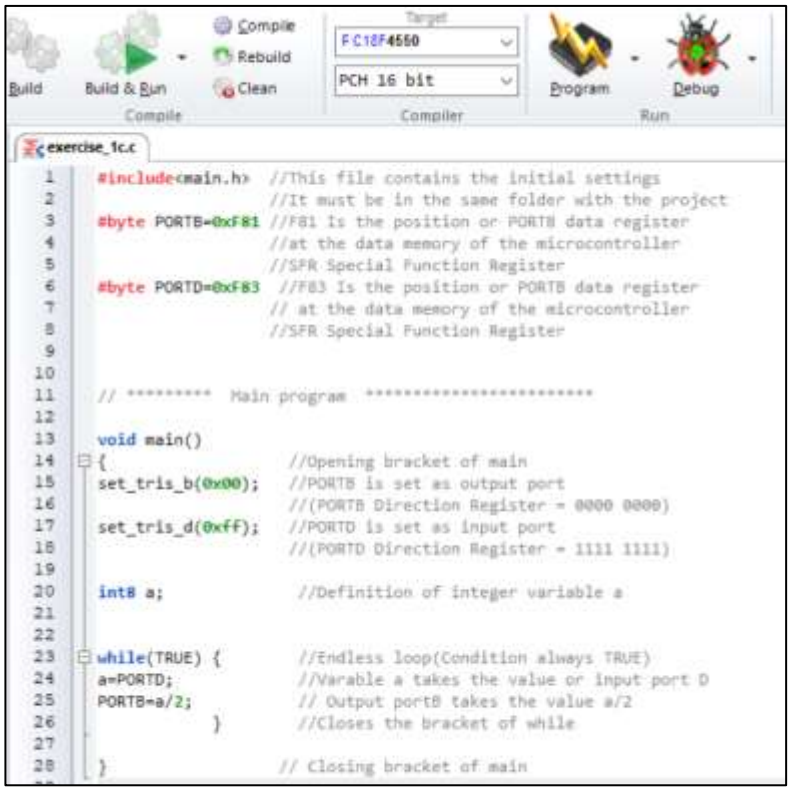


Figure 1(b). PORTD connected as input and PORTB as output

Step 2
(10 minutes)

Write in CCS Compiler the program in C language

```
#include<main.h>
// This file contains the initial settings

// It must be in the same folder with the project
#byte PORTB=0xF81
// F81 Is the position or PORTB data register

// at the data memory of the microcontroller

// SFR Special Function Register
#byte PORTD=0xF83
// F83 Is the position or PORTB data register

// at the data memory of the microcontroller

//SFR Special Function Register


//*********Main program***********

void main()
{
// Opening bracket of main
set_tris_b(0x00);
// PORTB is set as output port

// (PORTB Direction Register = 0000 0000)
set_tris_d(0xff);
// PORTD is set as input port

// (PORTD Direction Register = 1111 1111)

int8 a;
//Definition of integer variable a


while(TRUE) {
//Endless loop(Condition always TRUE)
a=PORTD;
//Varable a takes the value or input port D
PORTB=a/2;
// Output portB takes the value a/2
            }
//Closes the bracket of while

}
// Closing bracket of main
```

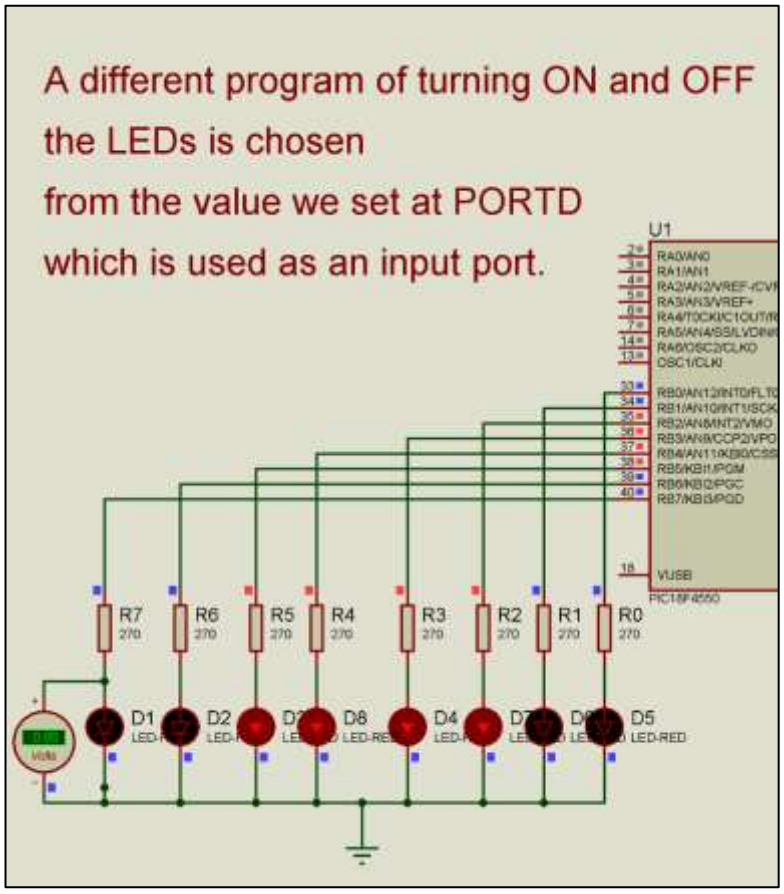
Step 3 (5 minutes)	• Calculate in binary form the expected value of PORTB if we form at PORTD the value 1111 1111. • Calculate in binary form the expected value of PORTB if we form at PORTD the value 0000 0001. • Calculate in binary form the expected value of PORTB if we form at PORTD the value 0000 0001. ***Tip***. *Division by 2 is equivalent of shifting all the digits to the right with 0 input to the MSB (Most Significant Bit).*
Step 4 (3 minutes)	Compile the program in C in order to create the program in the microcontroller machine code (hex file).  *Figure 2. CCS C Compiler, translation to machine code (hex file)*
Step 5 (1 minute)	Load to the microcontroller the hex file (program in machine code) that was created from the CCS Compiler.

Step 6 (4 minutes)	• By turning on and off the switches form at PORTD the number 6. Check that the value that appears at PORTB is 3. • By turning on and off the switches form at PORTD the value 255. Check that the value of PORTB is the expected one. • By turning on and off the switches form at PORTD the value 1. Check that PORTB takes the expected value.
Step 7 (17 minutes)	Suggested modifications and discussion: • Change the value of the pull up resistors to 1k. Does this create a problem? Would you prefer 10k or 1k as the value for the pull up resistors? Why? • Change the program so that PORTB equals PORTD+2. If PORTD=254 what will be the value of PORTB? Form at PORTD the value 254 and check if the value at PORTB is in accordance with the value expected. • Change the program so that PORTB equals 2 x PORTD. If PORTD=129 what is the expected value of PORTB. Form at PORTD the value 129 and check if the value of PORTB is the expected one. • Change the 8 switches with an 8 bit dipswitch (DIPSW_8). Check that the circuit works properly. • If in the program we define the variable **a** as **char a** instead of **int8 a**, would this create a problem? • Could we use the PORTC in the same way we used PORTD? Explain why not.

2.2 Activity 2. LEDs on PORTB turn ON and OFF in accordance with a program selected with the switches connected on PORTD

The purpose of this activity is to program the microcontroller so that the LEDs connected to PORTB turn ON and OFF in accordance with a program selected with the switches connected on PORTD.

Table 3. Activity 2

Activity 2nd (30 minutes)	**Step 1.** The circuit is drawn in the Proteus Design Suite. In this step 8 LEDs are connected to the PORTB parallel output port and 8 switches to the PORTD input port. **Step 2.** The program in C language is written. **Step 3.** The program is compiled with the use of CCS C compiler to the microcontroller machine code. The machine code is loaded to the flash memory of the microcontroller. **Step 4.** The animation is activated, and we check that the 8 LEDs turn ON and OFF in accordance with the program selected with the help of the switches.
Step 1 (10 minutes)	Draw the circuit of the picture at the Proteus Design Suite.  *Figure 3 (a). The schematic circuit of the activity*

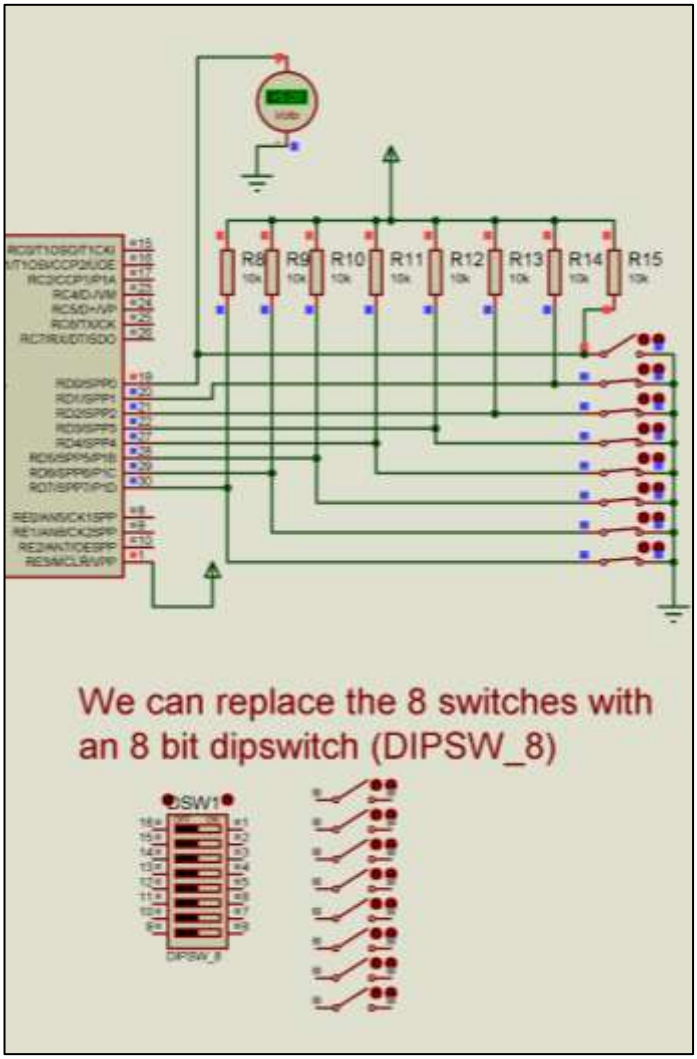


Figure 3 (b). The schematic circuit of the activity

Step 2 (15 minutes)	Write in CCS Compiler the program in C language

```
#include<main.h>
//The file <main.h> contains the initial settings

//This file must be in the same folder

//with your project
#byte PORTB=0xF81
//F81 is the address of the PORTB data register

//  in  the  data  memory  of  the  PIC18F4550
microcontroller.

// It's a Special Function Register (SFR)
#byte PORTD=0xF83
//F83 is the address of the PORTB data register

//  in  the  data  memory  of  the  PIC18F4550
microcontroller.

// It's a Special Function Register (SFR)

// ********* Main program *********************

void main()
{
// Opening bracket of main()

set_tris_b(0x00);
//  PORTB  becomes  output(Direction  Register=0000
0000)

set_tris_d(0xff);
//  PORTD  becomes  input(Direction  Register=1111
1111)

PORTB=0b00000000;
// PORTB takes the initial value 0000 0000

int a;
// definition of integer variable a for storing
the value of PORTD

int i;
// Integer variable we use inside the for

// With the while (TRUE){    } the content inside
the brackets

// is executed endless

// The word TRUE corespondes to a true condition.

// Instead of TRUE we could use the condition 5>1

// or any othere condition that is always true.
while(TRUE) {
//Eternal loop (condition always true)
```

```
            a=PORTD;
//The content of PORTD data register is transfered

// to the variable a
            switch (a){
                case     0:           PORTB=0xFF;
delay_ms(100);
PORTB=0x00;
delay_ms(100);

// Progran turning ON and OFF when a=0 (PORTD =
0000 0000)

break;


               case 1:  PORTB=0b11000011;

delay_ms(100);

PORTB=0b00111100;

delay_ms(100);


// Progran turning ON and OFF when  a=1 (PORTD =
0000 0001)


                                  break;


                case 2:  PORTB=0b10101010;

delay_ms(100)

;PORTB=0b01010101;delay_ms(100);


// Progran turning ON and OFF when a=2 (PORTD =
0000 0010)


                                 break;


                   case  3:  PORTB=0b10000000;  for
(i=1; i<=7; i++)

{delay_ms(50);

PORTB=PORTB/2;}
delay_ms(50);
                                    //         Progran
turning ON and OFF when a=3 (PORTD = 0000 0011)


                                   break;


                       case      4:      PORTB=0xF0;
delay_ms(100);

PORTB=0x0F;
```

	`delay_ms(100);`
	
	`// Progran turning ON and OFF when a=4 (PORTD = 0000 0100)`
	
	`break;`
	
	`}`
	`// Closing bracket of switch ()`
	
	`} //Closing bracket of while ()`
	
	`} // Closing bracket of main ()`

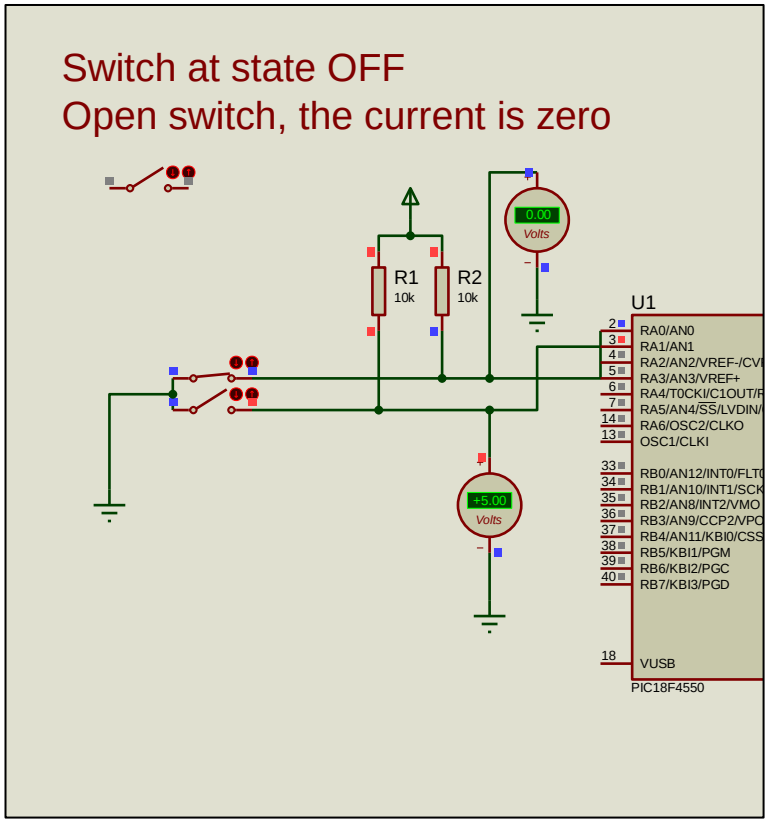
Step 3 (3 minutes)	Use the CCS C Compiler to translate the programm from C language to the microcontroller machine code. Load to the microcontroller the hex file (machine code) that was created from the CCS Compiler.
Step 4 (2 minutes)	Run the simulation and check by setting to PORTD the values 0, 1, 2, 3, 4 the correspondent program of turning ON and OFF the LEDs is executed.

2.3 Activity 3. Read the value of a specific pin of an input parallel port

The purpose of this activity is to read the value of pin A0 and the value of pin A1 and turn ON an LED only when A0=1 and A1=1 (Logical function AND between A0 and A1).

Table 4. Activity 3

Activity 3rd (30 minutes)	**Step 1.** The circuit is drawn at the Proteus Design Suite. In this step 2 switches are connected to the pins A0 and A1 and 1 LED to the pin D0. **Step 2.** The program in C language is written. The LED must be turned ON only when A0=1 and A1=1. **Step 3.** The program is compiled with the use of CCS C compiler to the microcontroller machine code (the hex.file is created). The

	program in machine code is loaded to the microcontroller. The animation is activated, and we check that the LED turns ON only when A0=1 and A1=1.
Step 1 (10 minutes)	Draw the circuit of the picture in the Proteus Design Suite  *Figure 4(a). The LED is turned ON only when A0=1 and A1=1*

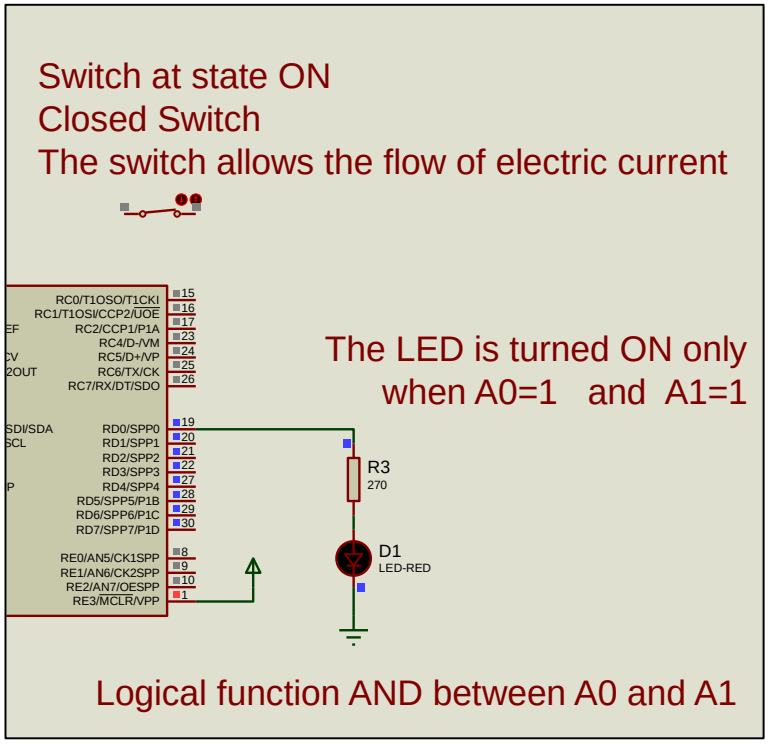


Figure 4(b). The LED is turned ON only when A0=1 and A1=1

Step 2
(15 minutes)

The student must complete the following program in C language.

```
#include<main.h>
//The <main.h> containd the initial settings

//It must be placed in the same folder with your
project.
#byte PORTA=0xF80
//F80 is the PORTA data register in the data memory
of the
//  microcontroller.  It's  a  Special  Function
Register
#byte PORTD=0xF83
//F83 is the PORTD data register in the data memory
of the

//  microcontroller.  It's  a  Special  Function
Register

// ********* main program *********************

void main()
{
//Opening of the main() bracket

set_tris_a(0xff);
//PORTA is set as input (Direction Register=1111
1111)
set_tris_d(0x00);
```

	`//PORTD is set as output (Direction Register=0000 0000)`
	
	`int1 a;`
	`// Definition of 1 bit integer for storing the value of A0`
	`int1 b;`
	`// Definition of 1 bit integer for storing the value of A1`
	
	`while(TRUE) {`
	`//Eternal loop (condition always TRUE)`
	`a=input(PIN_A0);`
	`b=input(PIN_A1);`
	`……`
	Complete the program with the necessary commands `……..…….`
	`} //Closing bracket of while()`
	`} // Closing bracket of main()`

Step 3 (5 minutes)	Compile the program in order to create the hex.file (program in machine code). Load the program (hex.file) to the microcontroller. Check that the LED is turned ON only when A0=1 and A1=1.

2.4 Activity 4. Counter of the 1s of input PORTD

The purpose of this activity is to count the 1s of PORTD and show the result at output PORTB.

Table 5. Activity 4

Activity 4rd (30 minutes)	**Step 1.** The circuit is drawn in the Proteus Design Suite. In this step 8 LEDs are connected to the PORTB parallel output port and 8 switches to the PORTD input port. **Step 2.** The program in C language is written. **Step 3.** The program is compiled with the use of CCS C compiler to the microcontroller machine code. The machine code is loaded to the flash memory of the microcontroller. **Step 4.** By turning ON and OFF the switches connected to PORTD we form an 8 bit number. We check that the multitude of the 1s at PORTD is shown on PORTB

Step 1
(10 minutes)

Draw the circuit of the picture in the Proteus Design Suite.

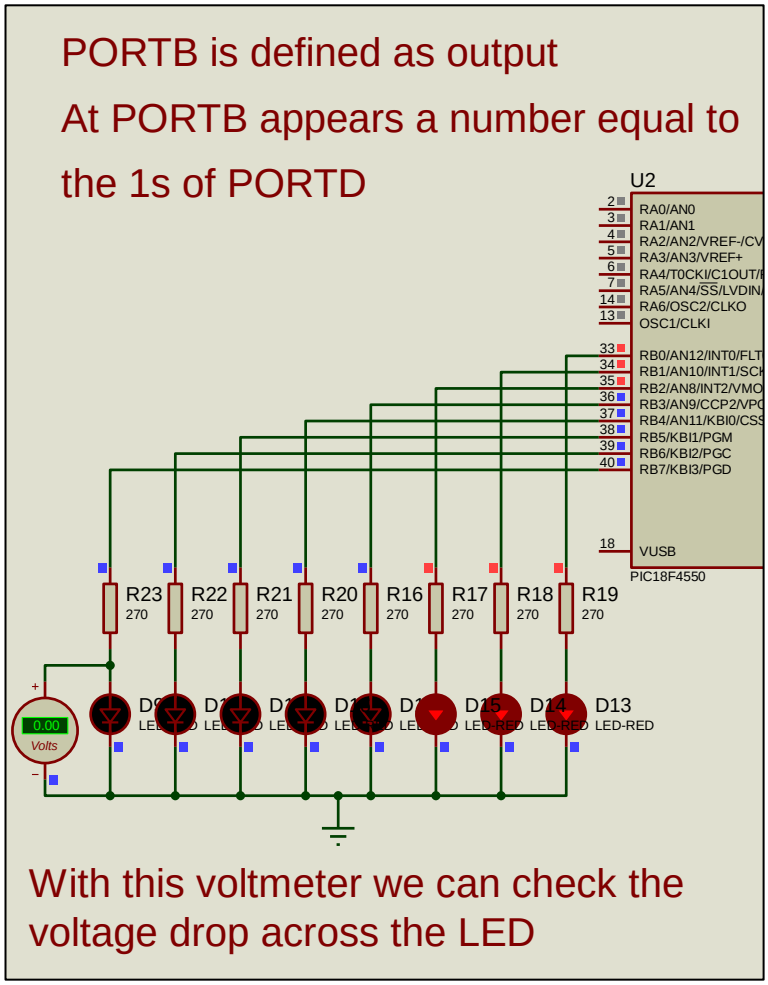


Figure 5(a). On PORTB shows the multitude of the 1s of PORTD

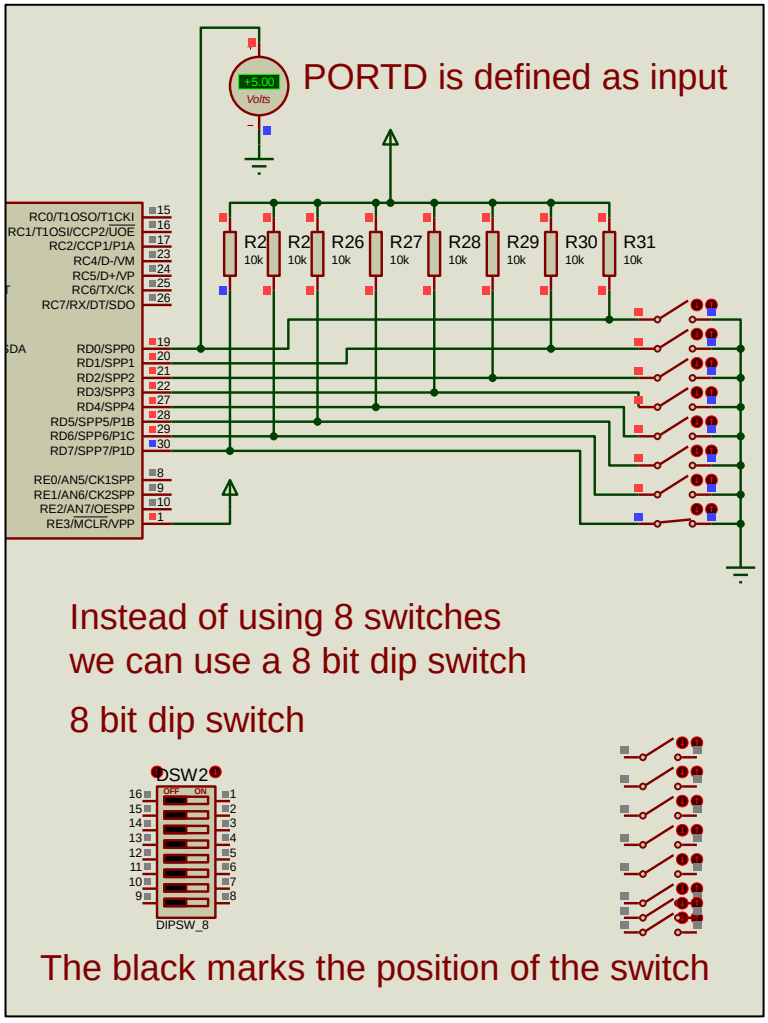


Figure 5(b). On PORTB shows the multitude of the 1s of PORTD

Step 2
(15 minutes)
	The student must complete the following program in C language.
	
	`#include<main.h>`
	`// This file contains the initial settings`
	
	`// It must be in the same folder with the project`
	
	`#byte PORTB=0xF81`
	`// F81 Is the position or PORTB data register`
	
	`// at the data memory of the microcontroller`
	
	`// SFR Special Function Register`
	
	`#byte PORTD=0xF83`
	`// F83 Is the position or PORTB data register`
	
	`// at the data memory of the microcontroller`
	
	`// SFR Special Function Register`
	
	`// ********* main program *********************`
	
	`void main()`
	`{`
	`// Opening bracket of main`
	
	`set_tris_b(0x00);`
	`// PORTB is set as output port`
	
	`// (PORTB Direction Register = 0000 0000)`
	
	`set_tris_d(0xff);`
	`// PORTD is set as input port`
	
	`// (PORTD Direction Register = 1111 1111)`
	
	`PORTB=0b00000000;`
	`// PORTB takes the initial value of 0000 0000`
	
	`int i=0;`
	`// Integer varable used in the  for() { }`
	
	`int a;`
	`// Integer varable a`
	`while(TRUE) {`
	`//Endless loop(condition always true)`
	`a=0;`
	`for (i=0; i<=7; i++){`
	…….
	Complete the program with the necessary commands ……
	`}`
	`// Closing bracket of for() { }`
	`PORTB=a;`
	`// PORTB takes the value of the variable a`

	`// Variable a equals the multitude of 1s of PORTD`
	`}`
	`// Closing bracket of while`
	
	`}`
	`// Closing bracket of main`
	
	Tip1. *The function bit_test(PORTD,i) checks the bit i of PORTD data register. If the bit equals to 1 then the function returns the value 1. If the bit is 0, then the function returns the value 0.*
	
	Tip2. *We can use the counter a=a+ bit_test(PORTD,i) to count the 1s of PORTD*

Step 3 (5 minutes)	Compile the program in order to create the hex.file (machine code). Load the program (hex.file) to the microcontroller. Check that the program shows on PORTB the multitude of the 1s of PORTD.

Chapter 3: **Recapitulation**

☞ The schematic of the circuits was drawn with Proteus Design Suite

☞ Animated LEDs were connected to the pins of the parallel output PORTB

☞ Switches with pull up resistors were connected to the pins of input PORTD

☞ The parallel PORTB of the microcontroller was defined as output and the PORTD was defined as input

☞ Programs in C were written in CCS C compiler

☞ In the programs the value of PORTD was read as an 8-bit word or the value of a specific input pin was read

☞ The programs in C were compiled to the microcontroller machine code (hex file)

☞ The machine codes were “loaded” to the microcontroller and the animation was activated

☞ It was checked that the programs run properly

References

CCS C Compiler Manual. Ccsinfo.com. (2021). Retrieved from https://www.ccsinfo.com/downloads/ccs_c_manual.pdf.

PIC18F2455/2550/4455/4550 Data Sheet. Ww1.microchip.com. (2006). Retrieved from https://ww1.microchip.com/downloads/en/devicedoc/39632c.pdf.

Proteus Tutorial : Getting Started with Proteus PCB Design (Version 8.6). Youtube.com. (2017). Retrieved from https://www.youtube.com/watch?v=GYAHwYUUs34.

Simple LED Circuits. Electronics Hub. (2017). Retrieved from https://www.electronicshub.org/simple-led-circuits/.

Appendix. Figures with high resolution

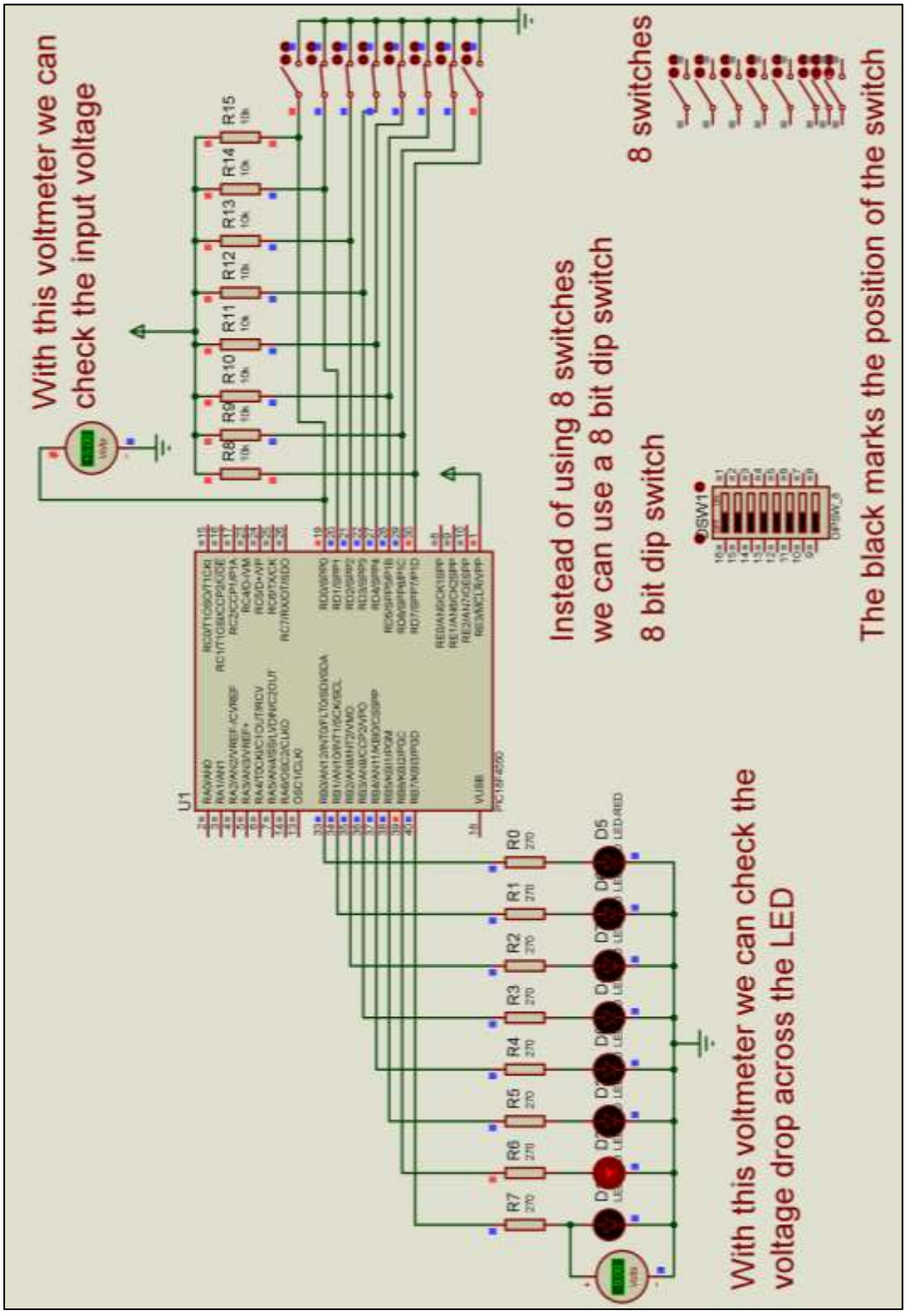


Figure 1. PORTD connected as input and PORTB as output. PORTB=PORTD/2

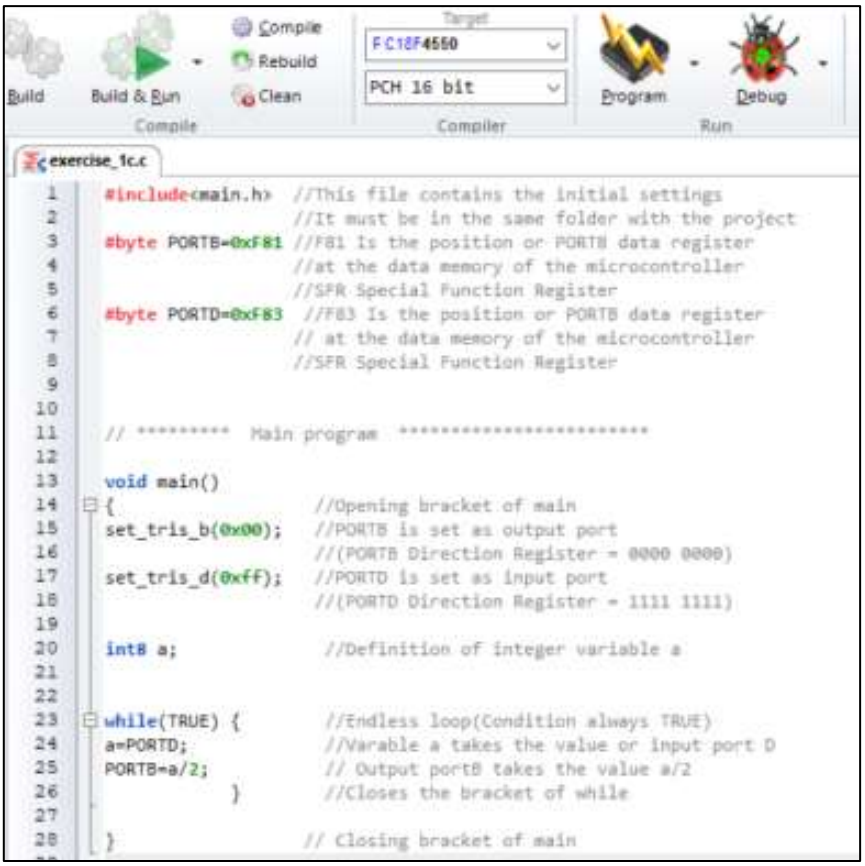


Figure 2. CCS C Compiler, translation to machine code (hex file)

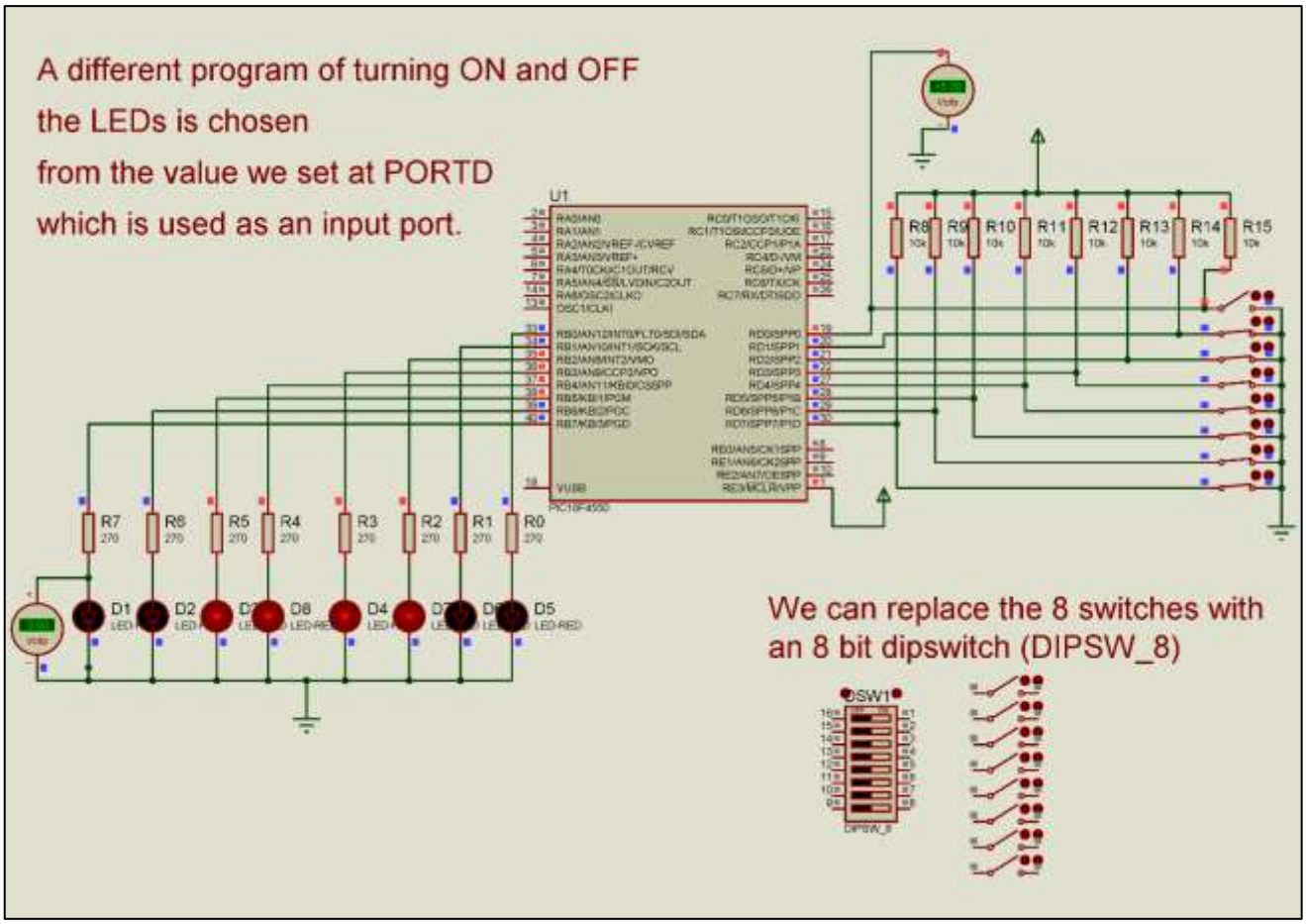


Figure 3. The schematic circuit of activity 2. Selection of program with the switches

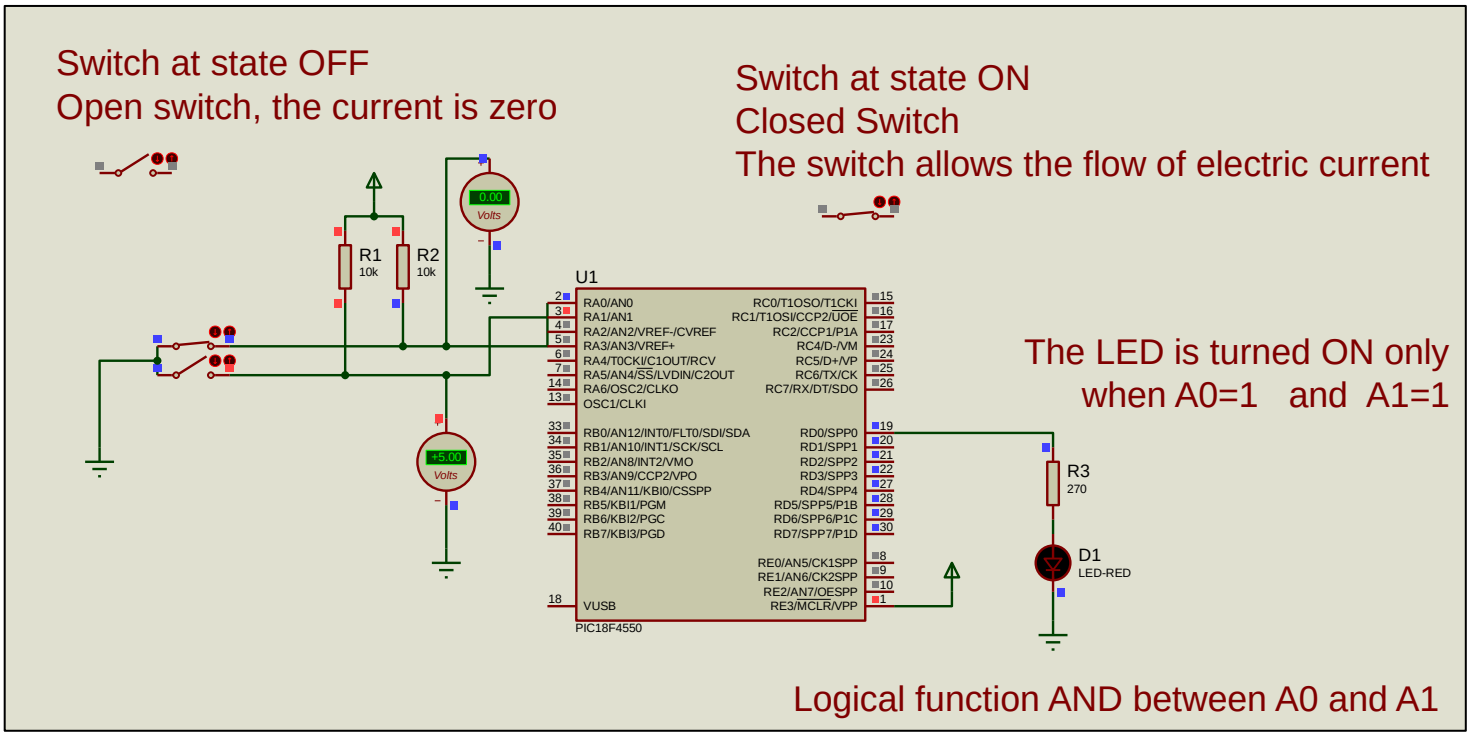


Figure 4. The LED is turned ON only when A0=1 and A1=1

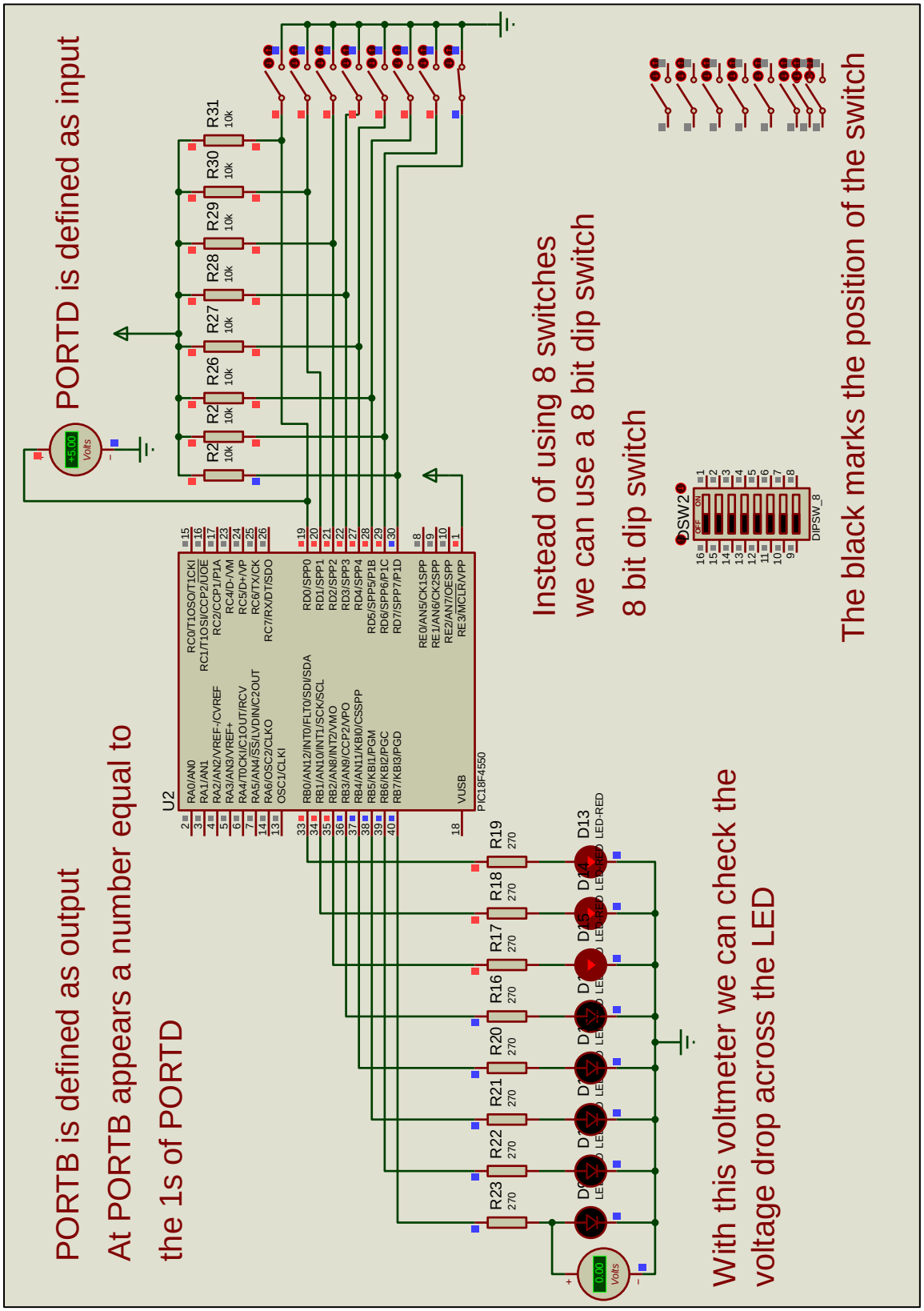


Figure 5. PORTB shows the multitude of the 1s of PORTD