ENGINE

Teaching online electronics, microcontrollers and programming in Higher Education

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

Practice leaflet: Module_2-1 pins as outputs

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 Where other published and unpublished source materials have been used, these have been acknowledged.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 outputs.

Chapter 1: **Overview**

Table 1. Overview

Title / short summary	**1. Use of microcontroller parallel ports as outputs**
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 send a binary word to a parallel port that has been defined as output • The student will be able to connect LEDs to the pins of a microcontroller • The student will be able to write a program in C Language to turn ON/OFF LEDs that are connected to the pins of a microcontroller • The student will be able to load and animate a microcontroller program in the Proteus Design Suite
Keywords	Direction Register, Data Register, Parallel Port, LED
Duration	The duration of the module_2-1 is 3 hours • Presentation of the module_2-1 by the teacher, 30 minutes • 1st activity, turn ON 8 LEDs connected to a parallel port, 1h • 2nd activity, turn ON/OFF 8 LEDs connected to a parallel port, 20 minutes • 3rd activity, rotate the value or PORTB, 20 minutes • 4th activity, animate a moving dot (LED on) from left to right, 20 minutes

Involved	**The teacher:** Presents the slides associated with the module_2-1 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_2-1 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_2-1 slides • Module_2-1 Evaluation leaflet • Module_2-1 Open project leaflet • Module_2-1 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.*

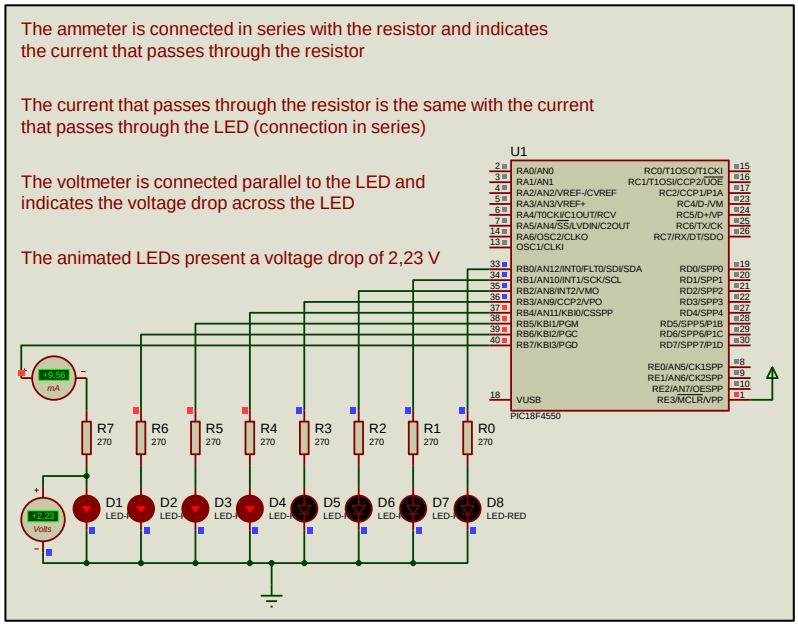
Chapter 2: **Activities**

2.1 Activity 1. Turn ON/OFF four LEDs

The purpose of this activity is to turn ON 4 LEDs that are connected to the PORTB of the microcontroller.

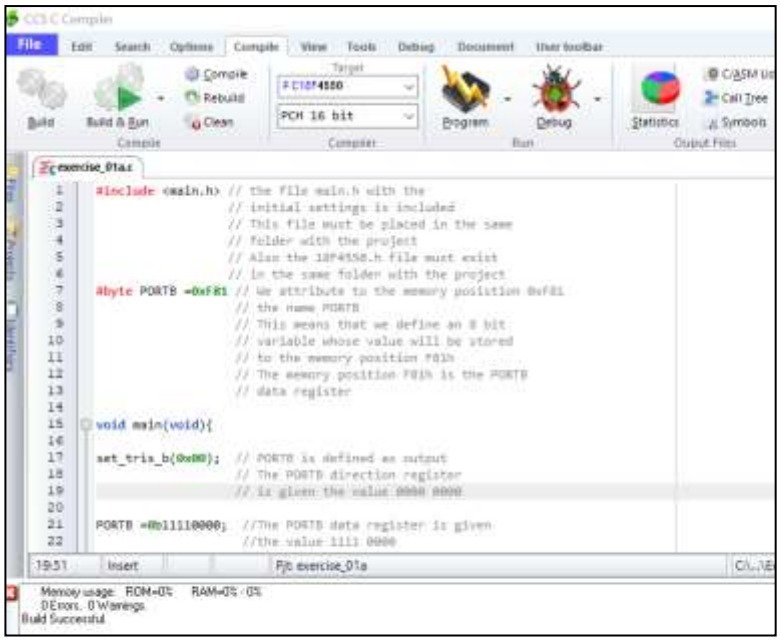
Table 2. Activity 1

Activity 1st (1 hour)	**Step 1.** The circuit is drawn in the Proteus Design Suite. In this step 8 LEDs are connected to the PORTD parallel port. A voltmeter and an ammeter are also connected to check the voltage drop across one LED and the current through this LED. **Step 2.** The values of the resistors in series with the LEDs are calculated so that the current through the LEDs is in the range from 10 to 15 mA. The voltage drop across the animated red LEDs at the Proteus Design Suite 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 and we check that the 8 LEDs turn ON. We check that the voltage drop across the LEDs and the current through them are as predicted. **Step 7.** Modifications and discussion.

Step 1 (25 minutes)	Draw the circuit of the picture in the Proteus Design Suite.  *Figure 1. LEDs connection to the PORTB parallel port*

Step 2
(10 minutes)
	Write in CCS Compiler the program in C language
	
	`#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`
	`#byte PORTB =0xF81`
	`// We attribute to the memory posistion 0xF81`
	`// the name PORTB`
	`// This means that we define an 8 bit`
	`// variable whose value will be stored`
	`// to the memory position F81h`
	`// The memory position F81h is the PORTB`
	`// data register`
	
	`void main(void){`
	
	`set_tris_b(0x00);`
	`// PORTB is defined as output`
	`// The PORTB direction register`
	`// is given the value 0000 0000`
	
	`PORTB =0b11110000;`
	`//The PORTB data register is given`
	`//the value 1111 0000`
	
	`while(TRUE) { }`
	`// eternal loop`
	`}`
	`// closes the bracket of main()`

Step 3 (5 minutes)	Calculate the value of the resistors so that the current through the LEDs is in the range from 10 to 15 mA. ***Tip1***. *We accept that the voltage drop across the red LED is 2,2 V and we calculate the value of the resistor so that the current through the resistor (and the LED) is 10 mA.* ***Tip2***. *The voltage drop across a LED depends on the color of the LED and we should consult the manufacturers datasheet*

Step 4 (10 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)	Run the simulation and check that the 4 LEDs to the left are turned ON. Check that the voltage drop across the LED and the current through it is as predicted. $V_{LED} =$ _____ $I_{LED} =$ _____

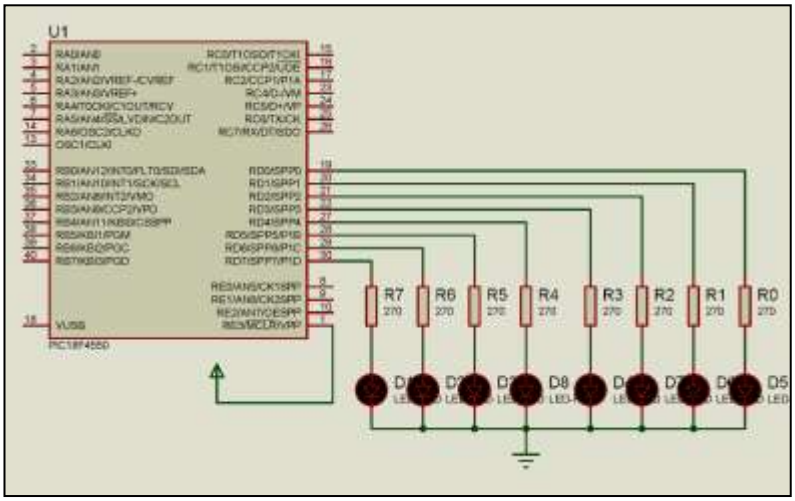
Step 7 (5 minutes)	Suggested modifications and discussion: • Change the value of a resistor in series to a LED to 1 KΩ and check if the LED turns ON. Give an explanation to what you observe • Change the value of a resistor in series to a LED to 330 Ω and check if the luminance of the LED changes. Give an explanation to what you observe

2.2 Activity 2. Turn ON/OFF LEDs every 500ms

The purpose of this activity is to turn ON and OFF every 500 ms eight LEDs that are connected to the PORTB of the microcontroller.

Table 3. Activity 2

Activity 2nd (20 minutes)	**Step 1.** The circuit is drawn in the Proteus Design Suite. In this step 8 LEDs are connected to the PORTD parallel port. A voltmeter and an ammeter are also connected to check the voltage drop across one LED and the current through this LED. **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.

Step 1 (5 minutes)	Draw the circuit of the picture at the Proteus Design Suite.  *Figure 3. LEDs connection to the PORTD parallel port*

Step 2
(10 minutes)
	Write in CCS Compiler the program in C language
	
	`#include <main.h>`
	`// The file of the initial settings`
	`// main.h is included`
	`#byte PORTD =0xF83`
	`// We attribute to the memory position`
	`// 0xF83 the name PORRD`
	`// This means that we create an 8 bit variable`
	`// whose value will be stored`
	`// at the memory position 0xF83`
	`// The memory position F81h is`
	`// the PORTD data register`
	
	`void main(void){`
	
	`set_tris_d(0x00);`
	`// PORTD is made output`
	`// This is done by giving to the`
	`// PORTD direction register the value 0000 0000`
	`PORTD =0b11111111;`
	`//We give to the PORTD data register`
	`//the value 1111 1111`
	`while(TRUE) {`
	`// eternal loop`
	`delay_ms(250);`
	`//delay 250 ms`
	`PORTD=0b00000000;`
	`// turn off the 8 LEDs`
	`delay_ms(250);`
	`//delay 250 ms`
	`PORTD=0b11111111;`
	`//turn on the 8 LEDs`
	`}`
	`// closes the bracket or while`
	
	`}`
	`// closes the 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 that the 8 LEDs turn on and off.

2.3 Activity 3. Rotate the value of PORTB

The purpose of this activity is to set an initial value to the PORTB of the microcontroller and to rotate it continuously from right to left.

Table 4. Activity 3

<table>
<tr>
<td>Activity 3rd
(20 minutes)</td>
<td>The initial value of PORTB will be set to 11110000 and this value will be rotated to the left in time steps of 100 ms.

Step 1. The circuit is drawn at the Proteus Design Suite. In this step 8 LEDs are connected to the PORTB parallel port.

Step 2. The program in C language is written.

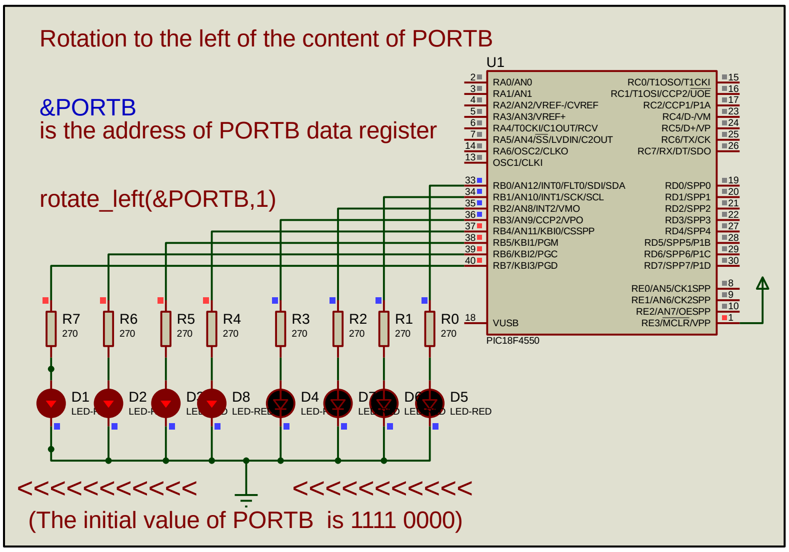
Step 3. The program is compiled with the use of CCS C compiler to the microcontroller machine code (the hex.file is created). The program in machine code is loaded to the microcontroller. The animation is activated, and we check that the 8 LEDs turn ON and OFF.</td>
</tr>
<tr>
<td>Step 1
(10 minutes)</td>
<td>Draw the circuit of the picture in the Proteus Design Suite


Figure 4. LEDs connection to the PORTB parallel port</td>
</tr>
</table>

Step 2
(5 minutes)
	The student must complete the following program in C language.
	
	`#include <main.h>`
	`// We include the file of`
	`// the initial settings  main.h`
	`#byte PORTB =0xF81`
	`// we assign to the memory position 0xF81`
	`// the name PORTB`
	`// This means that we create an 8 bit`
	`// variable whose value will be stores`
	`// at the memory position F81h`
	`// The memory position F81h`
	`// is the PORTB data register`
	`void main(void) {`
	`set_tris_b(0x00);`
	`// We define all the pins of PORTB as outputs`
	`// This is done by giving to the`
	`//PORTB direction register the value 00000000`
	`PORTB =0b11110000;`
	`//We give the PORTB data register`
	
	`//the initial value 111100. This is the value`
	`// that will be rotated`
	`while(TRUE) {`
	`// eternal loop`
	
	……
	`Complete the program with the necessary commands`
	……
	`}`
	`//closes the bracket of the while`
	
	`}`
	`// closes the bracket of main()`
	
	Tip1. *The command rotate_left(&PORTB,1) will be used to rotate the value of PORTB.*
	
	Tip2. *The command delay_ms(100) causes delay of 100 ms*

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 program runs properly.

2.4 Activity 4. Create a moving dot with LEDs

The purpose of this activity is to create a moving dot with the 8 LEDs that are connected to PORTB of the microcontroller. The dot will move from left to right and from right to left.

Table 5. Activity 4

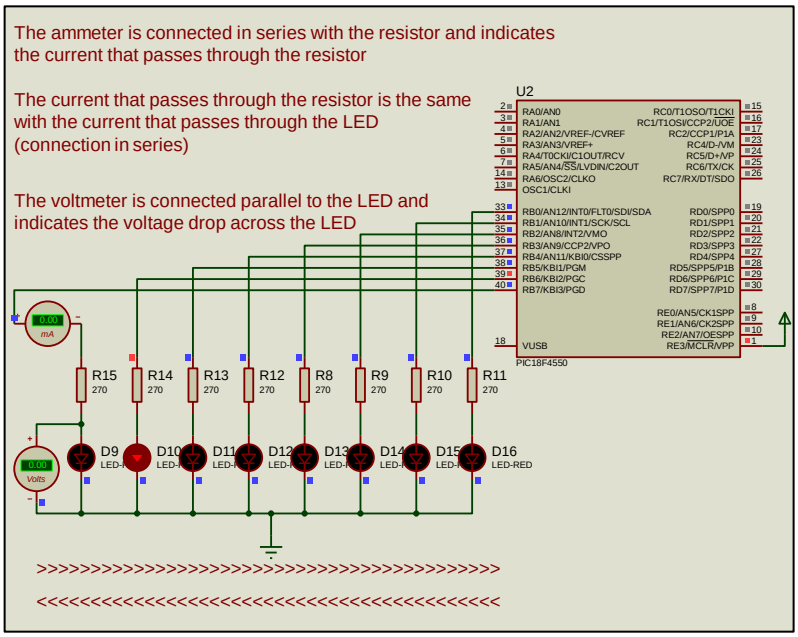
Activity 4rd (20 minutes)	**Step 1.** The circuit is drawn in the Proteus Design Suite. In this step 8 LEDs are connected to the PORTB parallel port. **Step 2.** The program in C language is written. **Step 3.** The program is compiled with the use of CCS C compiler to the microcontroller machine code (the hex.file is created). The machine code is loaded to the microcontroller. The animation is activated and we check that the ON LED moves from left to right and from right to left.
Step 1 (5 minutes)	Draw the circuit of the picture in the Proteus Design Suite. 

Figure 5. LEDs connection to the PORTB parallel port

Step 2
(10 minutes)
	The student must complete the following program in C language.
	
	`#include <main.h>`
	`// We include the file`
	
	`// of the initial settings main.h`
	`#byte PORTB =0xF81`
	`// we assign to the memory position 0xF81`
	
	`// the name PORTB`
	
	`// This means that we create an 8 bit`
	
	`// variable whose value will`
	
	`// be stored at the memory address F81h`
	
	`// At the memory address F81h is the PORTB`
	
	`// data register`
	`int8 i;`
	`void main(void){`
	
	`set_tris_b(0x00);`
	`// We define PORTB as output`
	
	`// This is done by giving to the PORTB`
	
	`// direction register the value 0000 0000`
	
	`PORTB =0b10000000;`
	`// We give to the PORTB data register`
	
	`//the initial value 1000 0000`
	
	`while(TRUE) {`
	`// eternal loop`
	`for(i=1;i<=7;i++){`
	……
	Complete the program with the necessary commands
	……
	`}`
	`// closes the bracket of for`
	
	`for(i=7;i>=1;i--){`
	……
	Complete the program with the necessary commands
	……
	`}`
	`//closes the bracket of for`
	`}`
	`// Closes the bracket of while(TRUE)`
	`}`
	`// Closes the bracket of main()`
	`//The initial value of PORTB is 10000000.`

	Tip1. *Division of the value of PORTB by 2 moves the 1 to the right.* ***Tip2***. *Multiplication of the value of PORTB with 2 moves the 1 to the left.*
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 runs properly.

Chapter 3: **Recapitulation**

- ☞ The schematic of the circuit was drawn with Proteus Design Suite.
- ☞ LEDs were connected to the pins of the parallel PORTB.
- ☞ The program in C was written in CCS C compiler.
- ☞ The parallel PORTB of the microcontroller was defined as output.
- ☞ An 8 bit binary word was sent to PORTB and the LEDs were turned ON/OFF.
- ☞ The delay_ms() was used to cause delays in the program.
- ☞ The program in C was compiled to the microcontroller machine code (hex file).
- ☞ The machine code was “loaded” to the microcontroller and the animation was activated.

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

The ammeter is connected in series with the resistor and indicates the current that passes through the resistor

The current that passes through the resistor is the same with the current that passes through the LED (connection in series)

The voltmeter is connected parallel to the LED and indicates the voltage drop across the LED

The animated LEDs present a voltage drop of 2,23 V

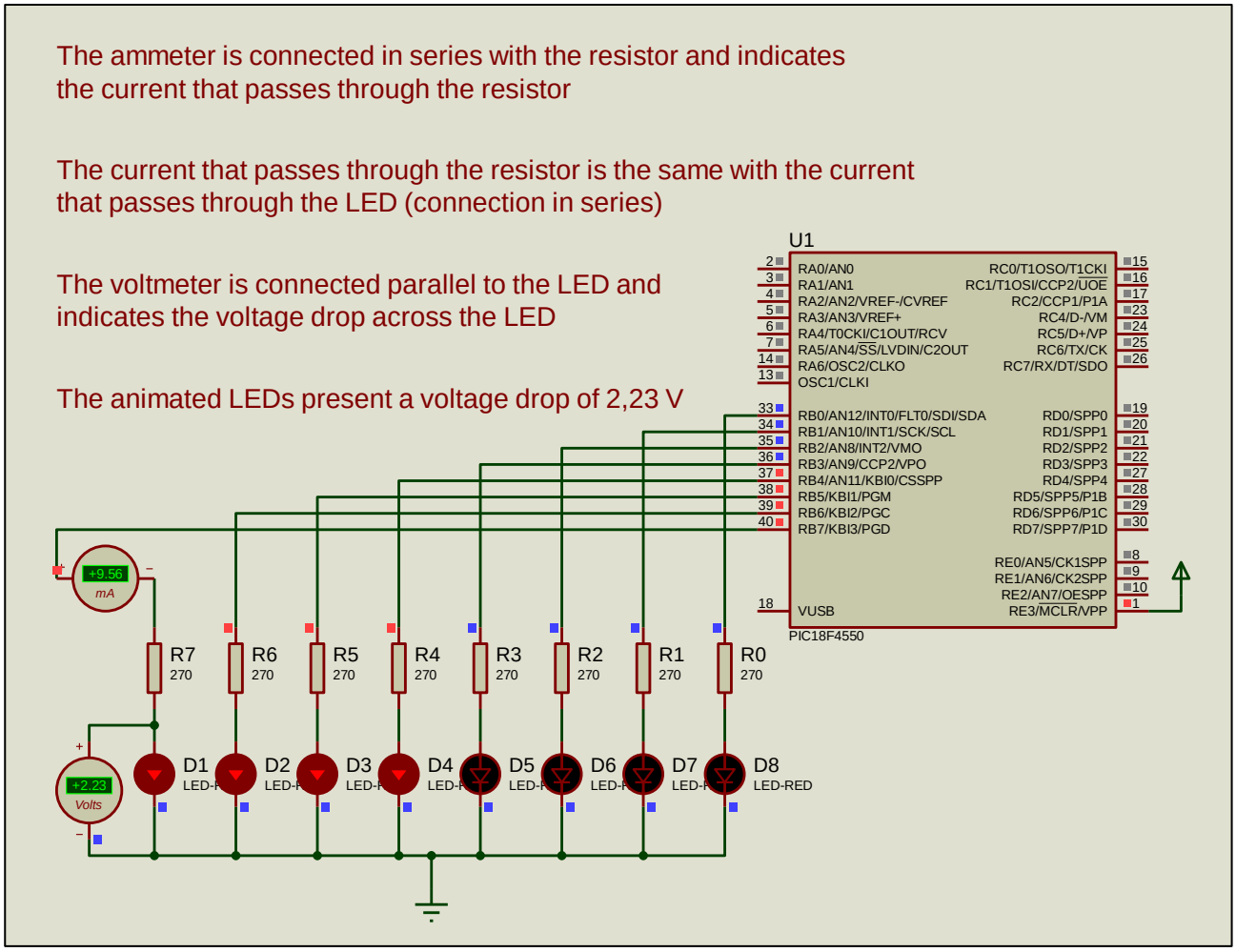


Figure 1. LEDs connection to the PORTB parallel port

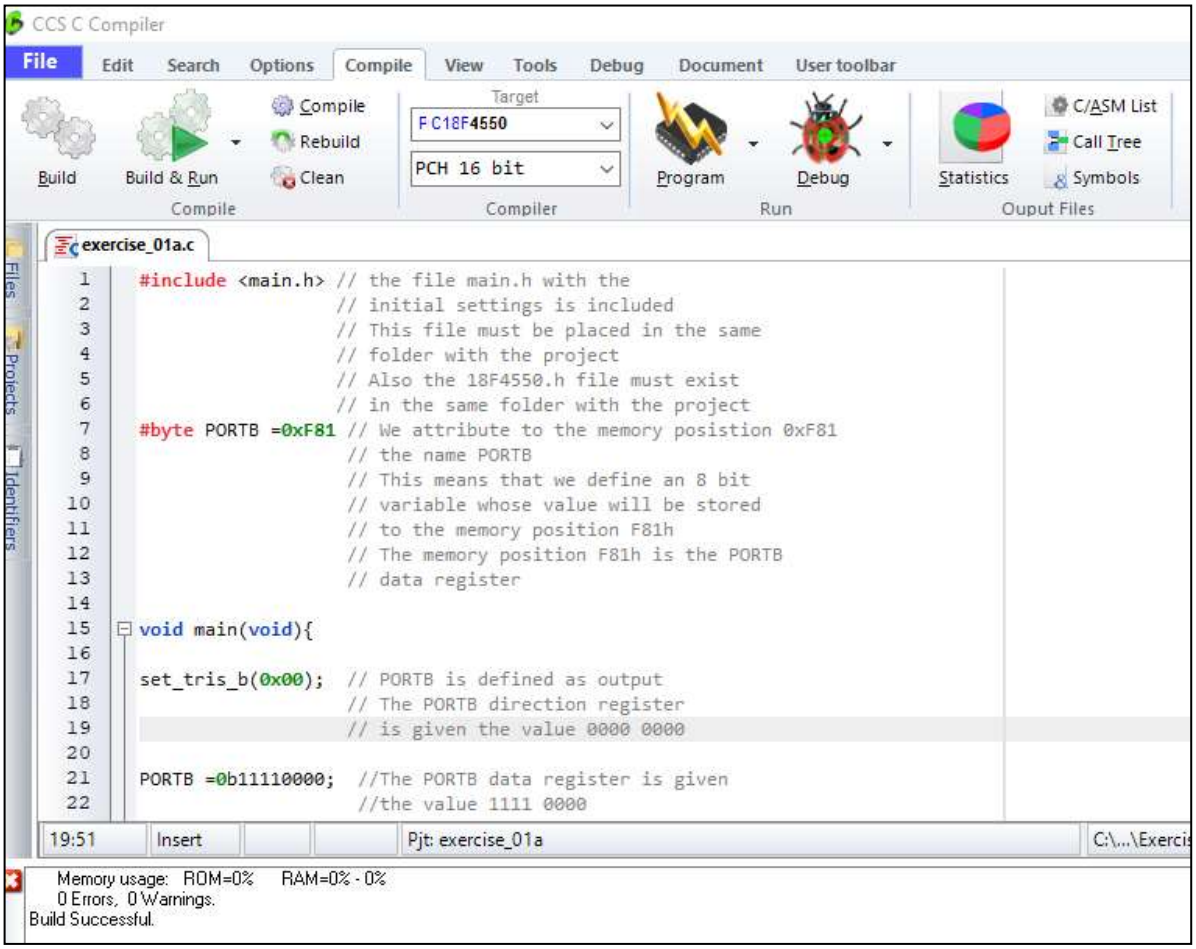


Figure 2. CCS C Compiler, translation to machine code (hex file)

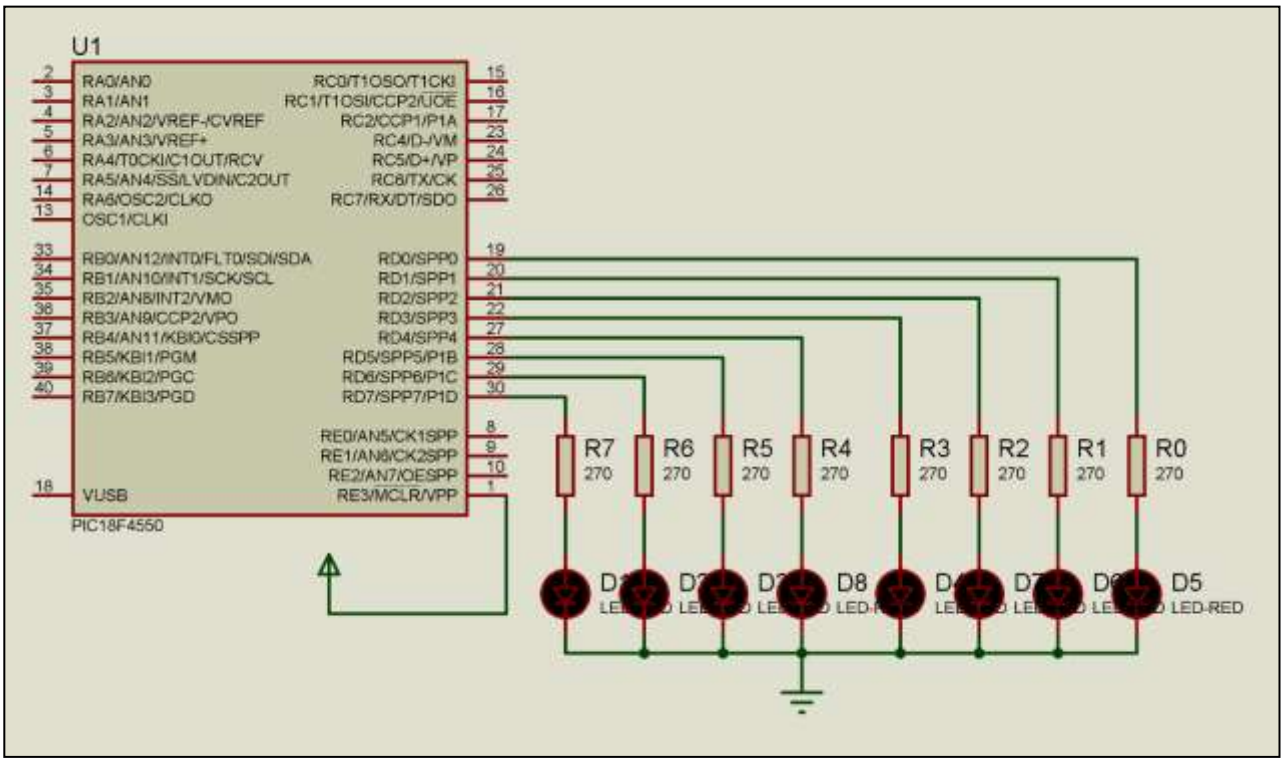


Figure 3. LEDs connection to the PORTD parallel port

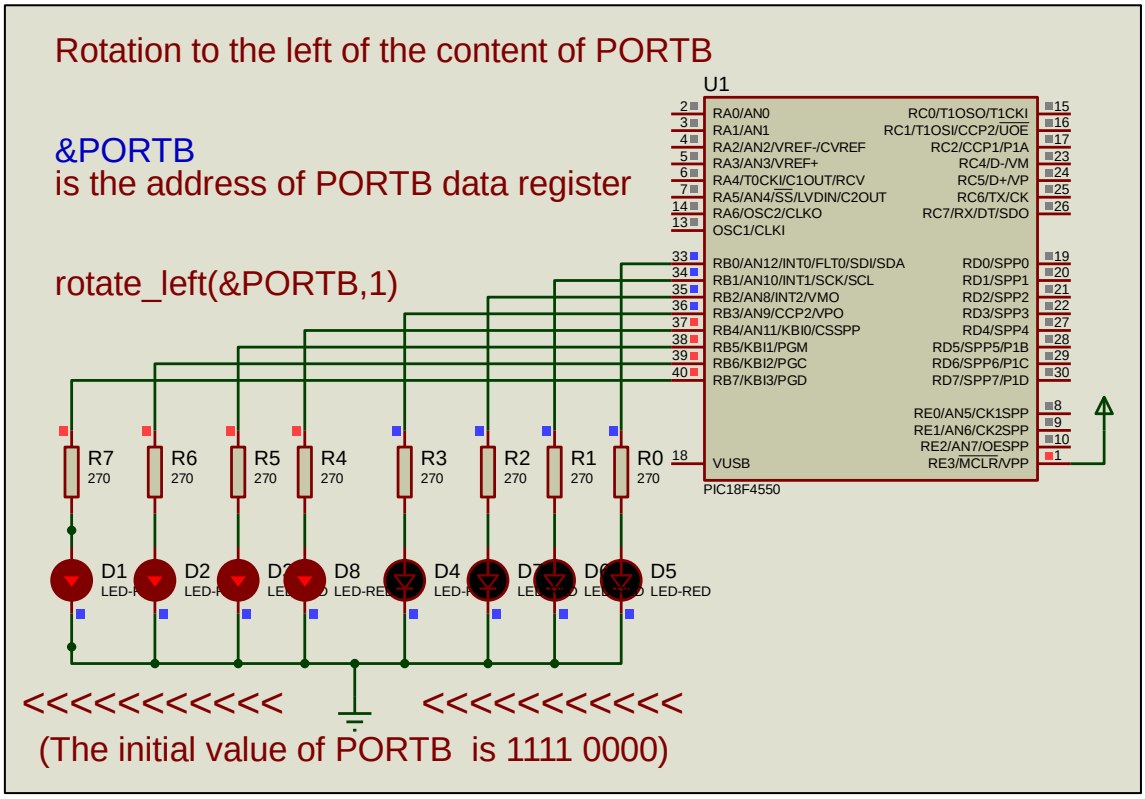


Figure 4. LEDs connection to the PORTB parallel port

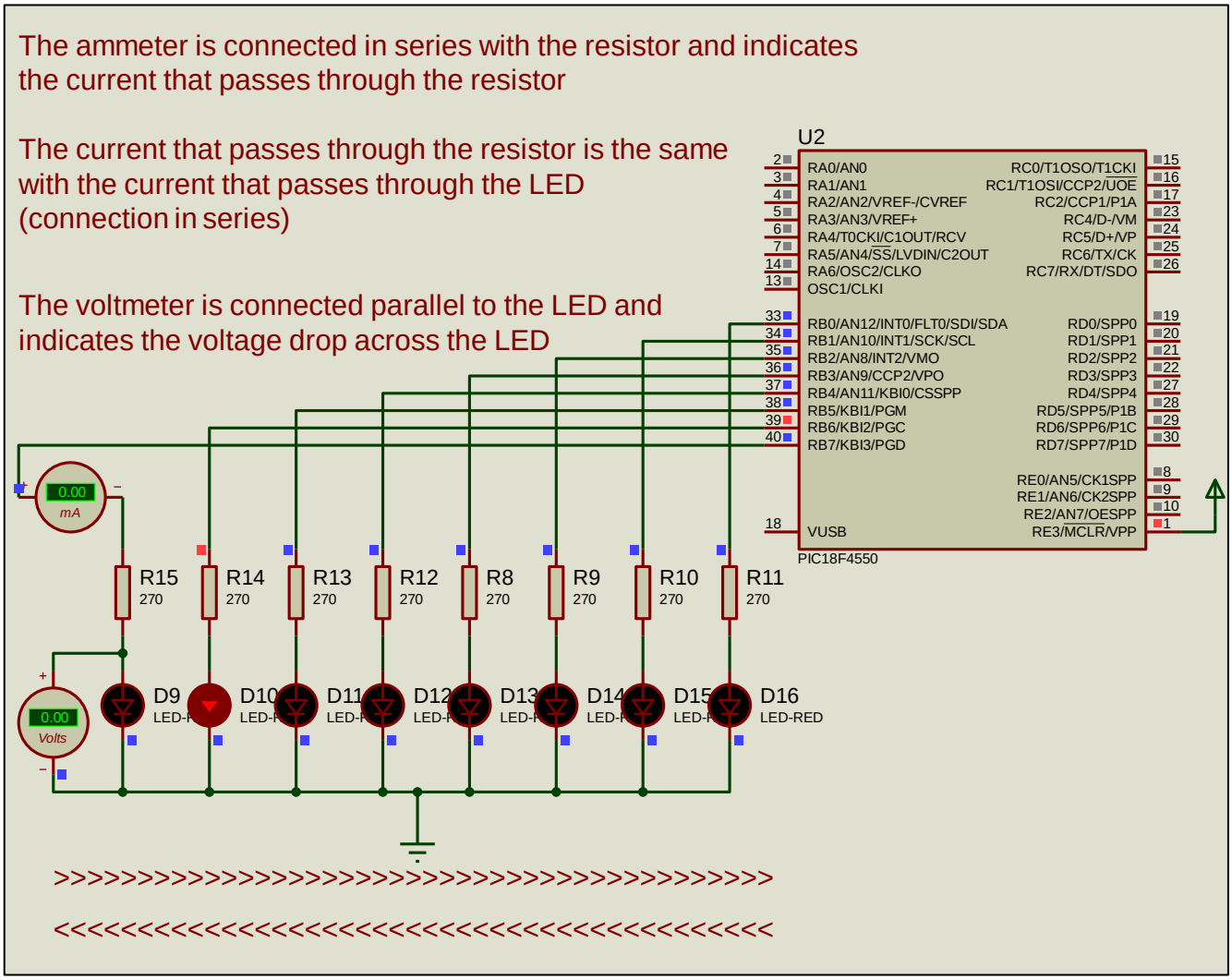


Figure 6. LEDs connection to the PORTB parallel port (moving dot)