ENGINE

Teaching online electronics, microcontrollers and programming in Higher Education

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

Quick start Guide: CCS C Compiler / Proteus Design Suite

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 Where other published and unpublished source materials have been used, these have been acknowledged.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

This document serves as a Quick Start Guide for CCS C Compiler (5.008) and Proteus Design Suite (8.9).

Chapter 1: **Get start with CCS C Compiler**

1.1 Create project

Figure 1 shows the CCS C Compiler environment. To create a new project we must go to the menu File => New => Project Wizard, as shown in Figure 2.

Figure 1. CCS C Compiler

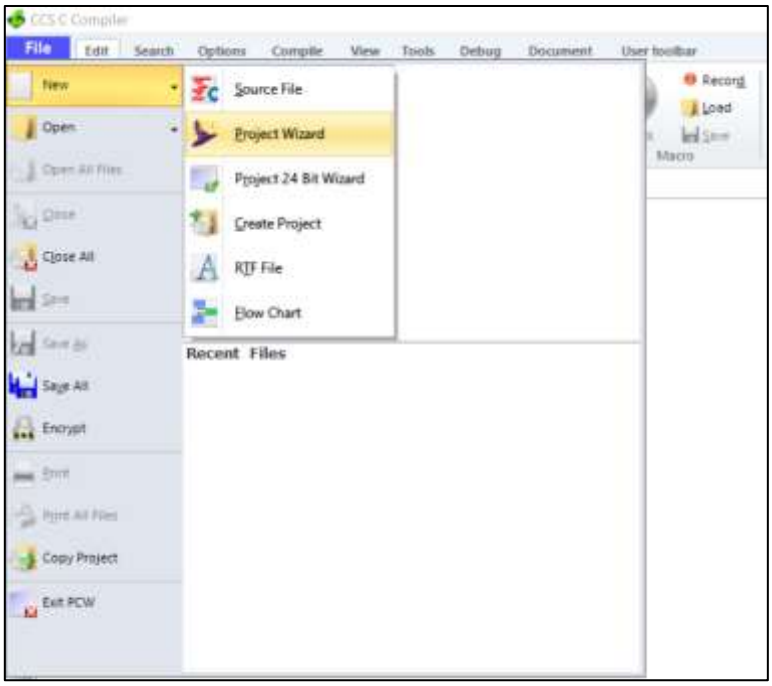

Figure 2. New project

After naming the project and selecting the path in which it will be saved, a window appears as shown in Figure 3. There, we need to define the device we will use. That is, the PIC18F4550. Finally, we click “Create Project”.

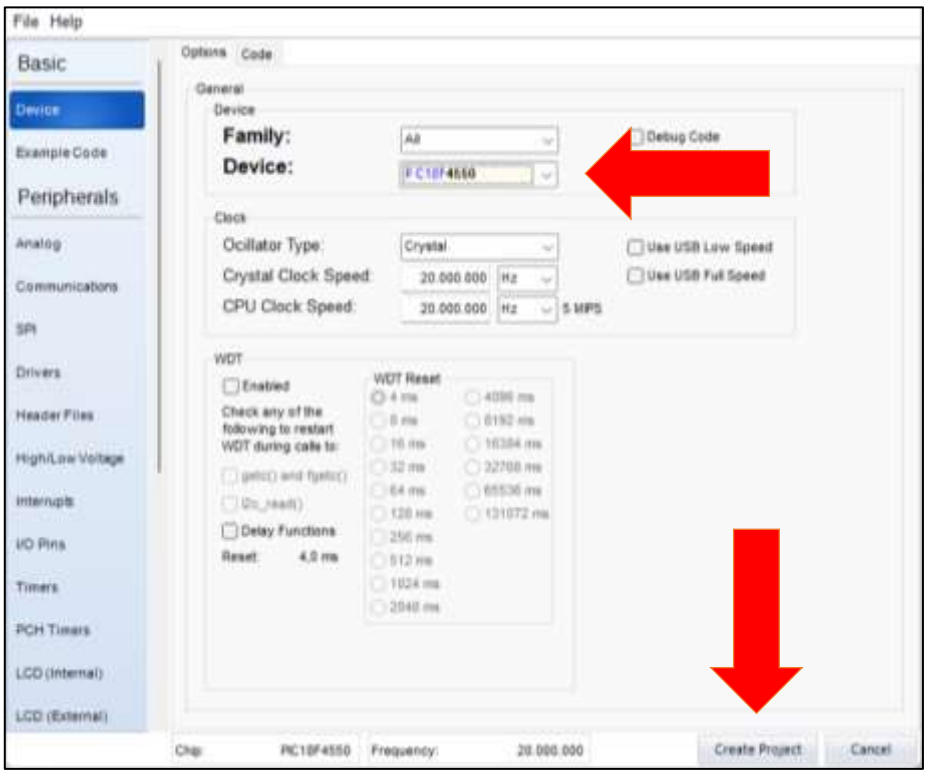

Figure 3. Define device

In the folder we chose to create the project, various files have appeared, as shown in Figure 4.

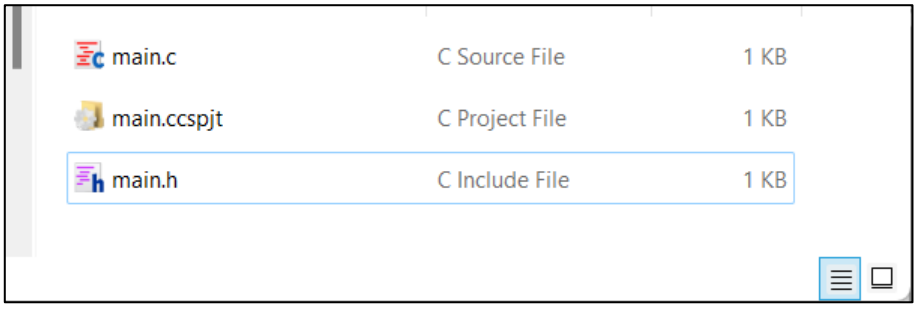

Figure 4. Project files

1.2 Code and .hex file

In the .c file we have to write our code, as shown in Figure 5.

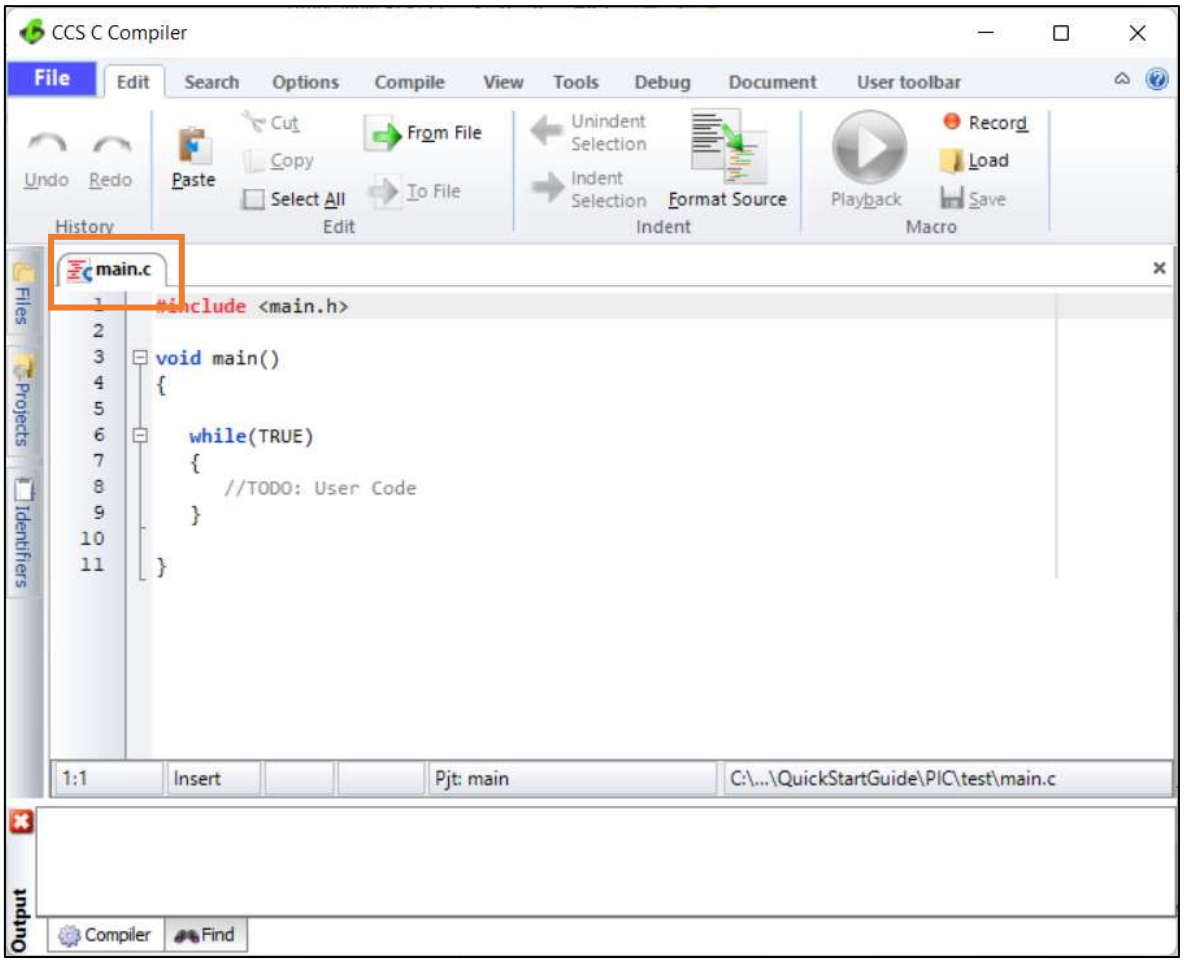

Figure 5. Code file

When we have completed the code, we need to convert it to a hexadecimal (.hex) file and load it into the microcontroller. For this, we go to the Compile tab, and click “Compile”, as shown in Figure 6.

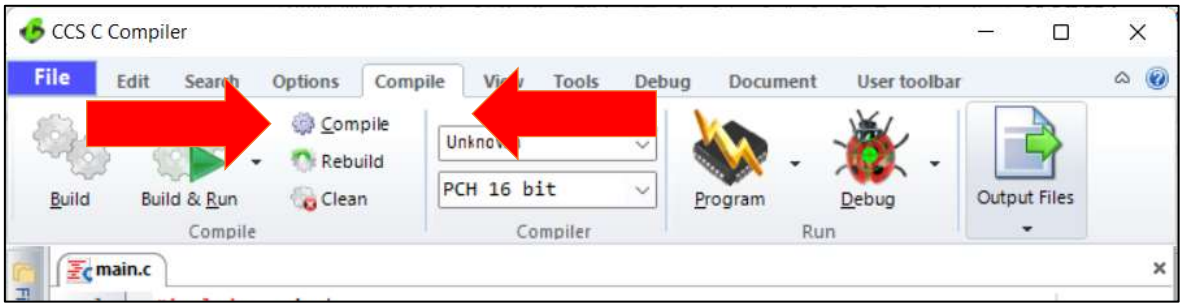

Figure 6. Compile

If the program is syntactically correct, then the message "Build Successful" appears in the Output, as shown in Figure 7.

After a successful compile, many files have appeared in the project folder, including .hex, as shown in Figure 8.

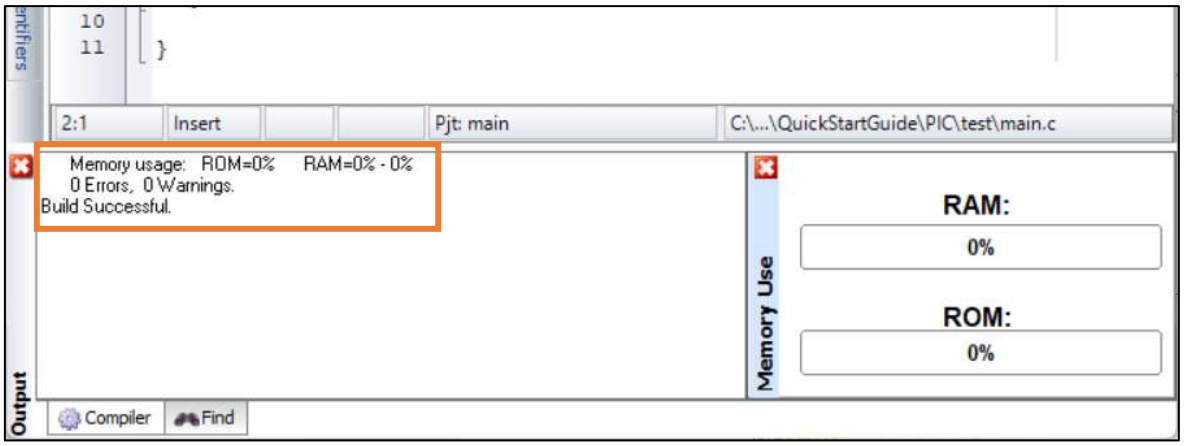

Figure 7.

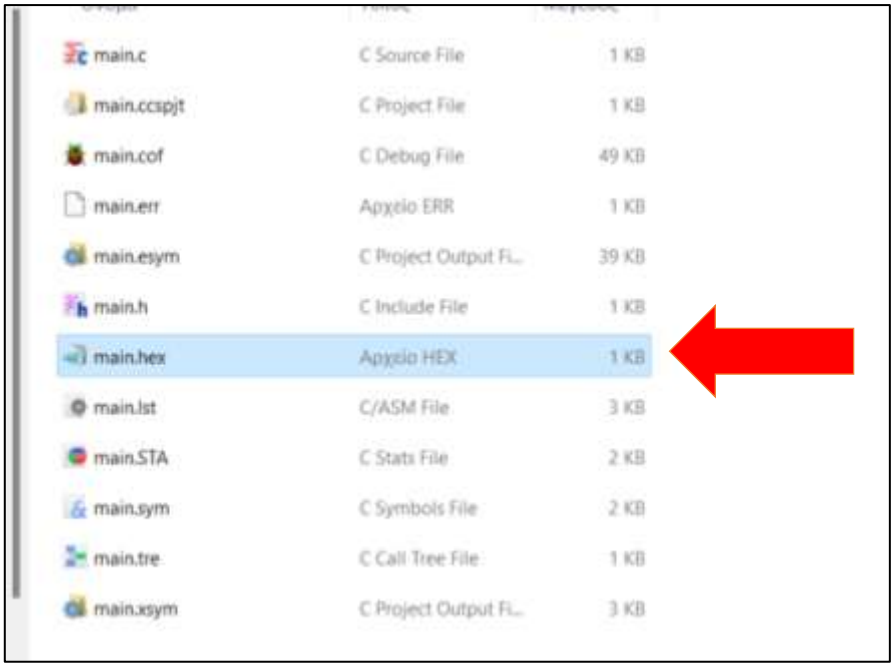

Figure 8. .hex file

Chapter 2: **Get start with Proteus Design Suite**

2.1 Create project

Figure 9 shows the environment of the Proteus Design Suite. To create a new project, we must go to the menu File => New Project. Then a window appears and we choose a name and path for the project, as shown in Figure 10.

Figure 9. Start page

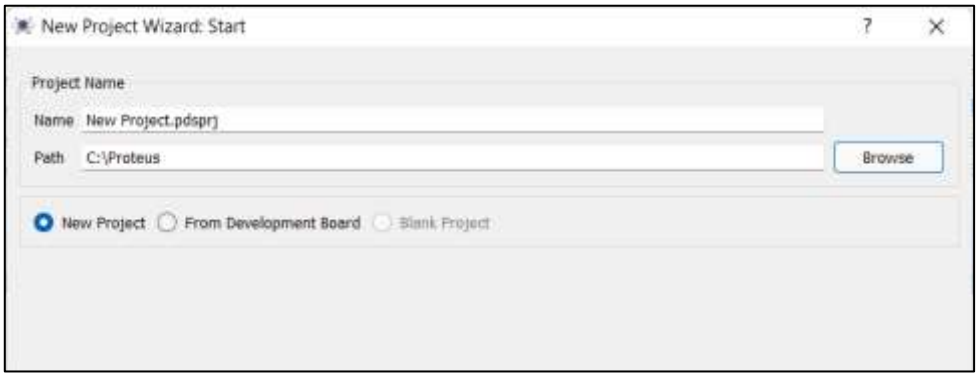

Figure 10. New project

Next, select the “default schematic”, as shown in Figure 11.

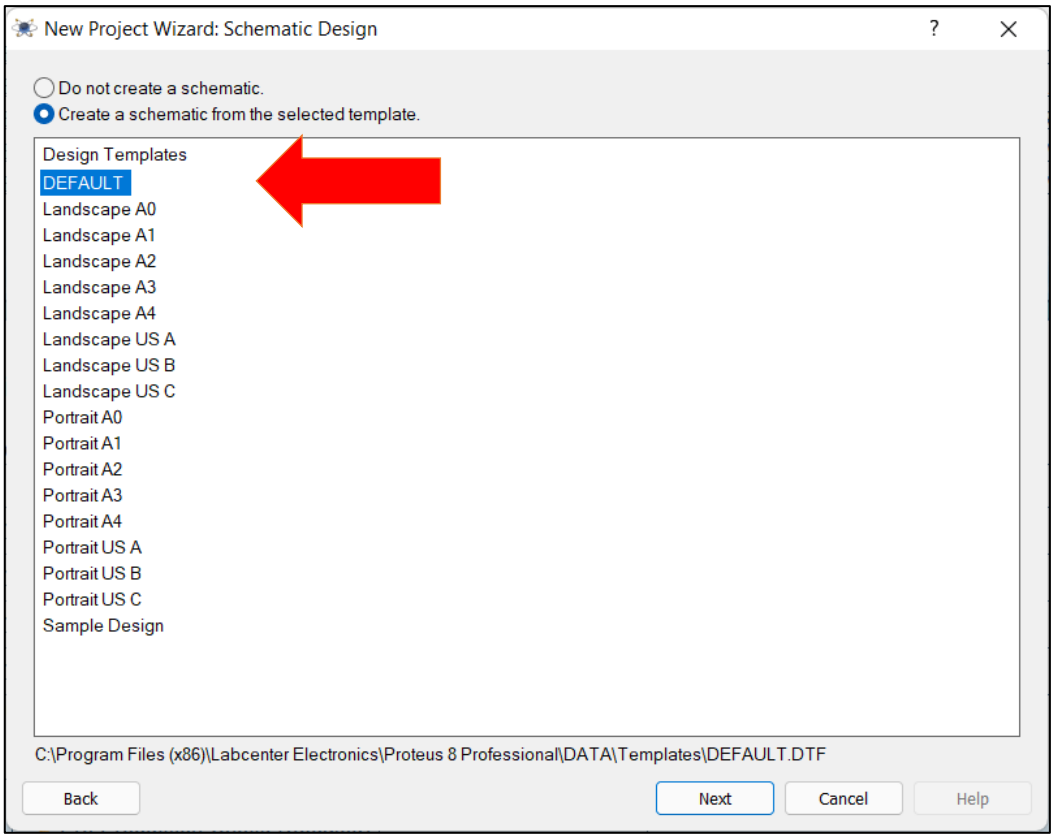

Figure 11. Default schematic

Next, we choose not to create a PCB layout and not include firmware, as shown in Figure 12 and Figure 13.

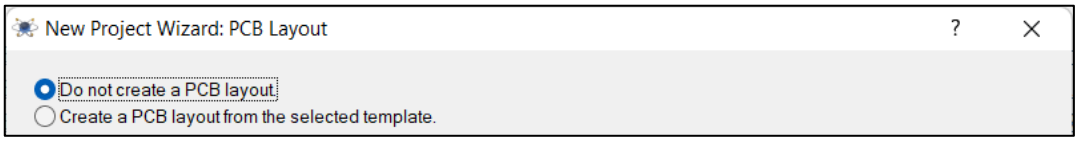

Figure 12. PCB layout

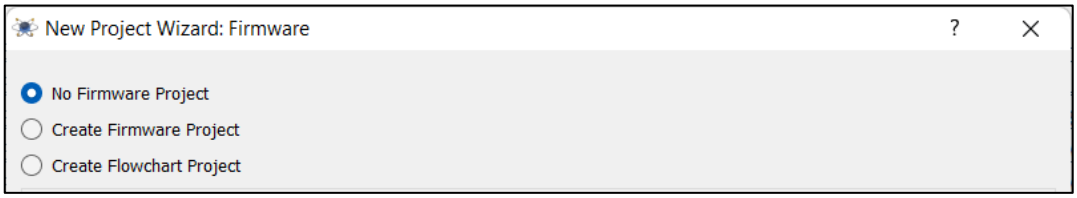

Figure 13. Project firmware

Finally, we complete the project creation by pressing the "Finish" button, as shown in Figure 14.

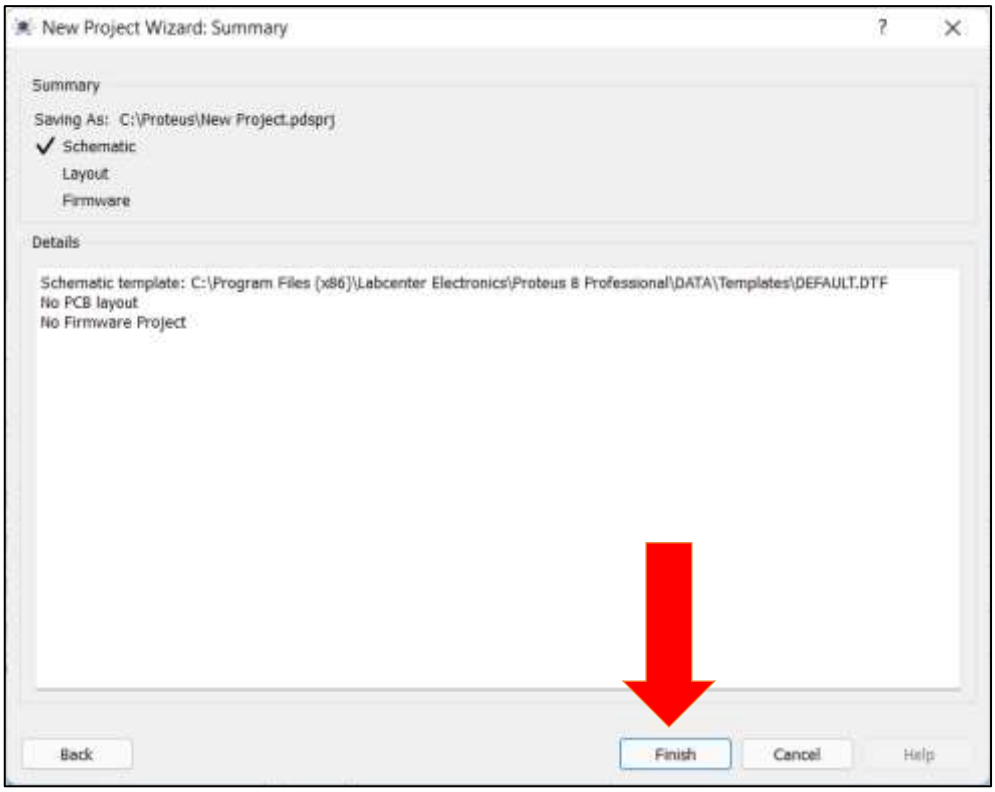

Figure 14. Project summary and "Finish"

2.2 Schematic

Figure 15 shows the Schematic capture, in which we can create circuits and simulate them.

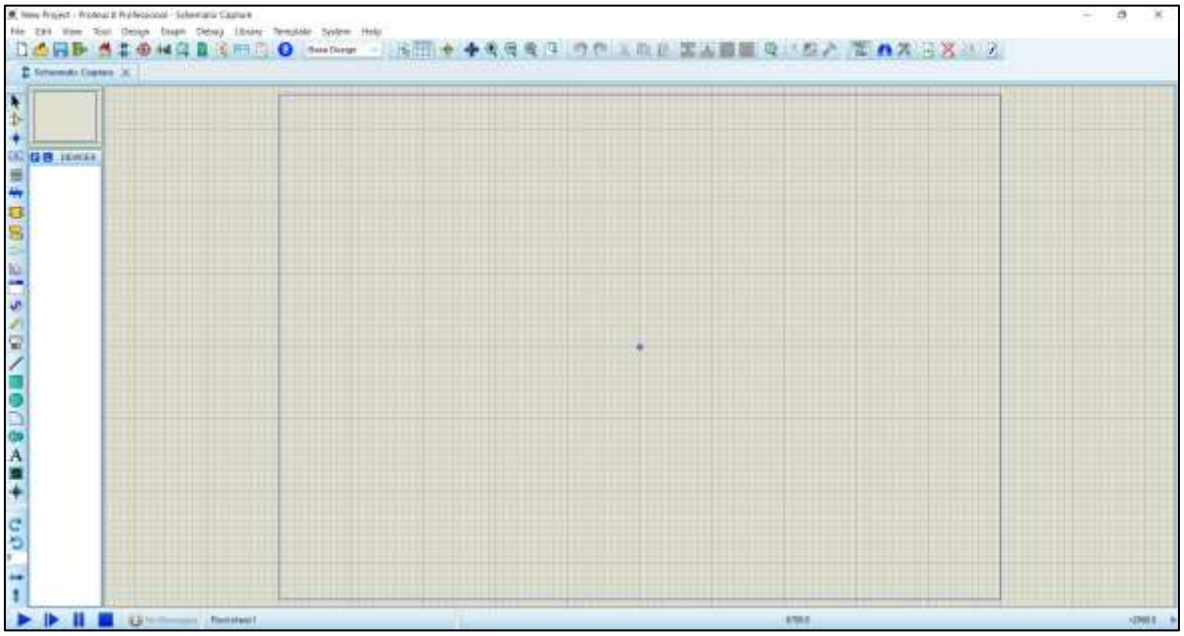

Figure 15. Schematic capture

To the left on the toolbar we find useful operations, as described in Figure 16.

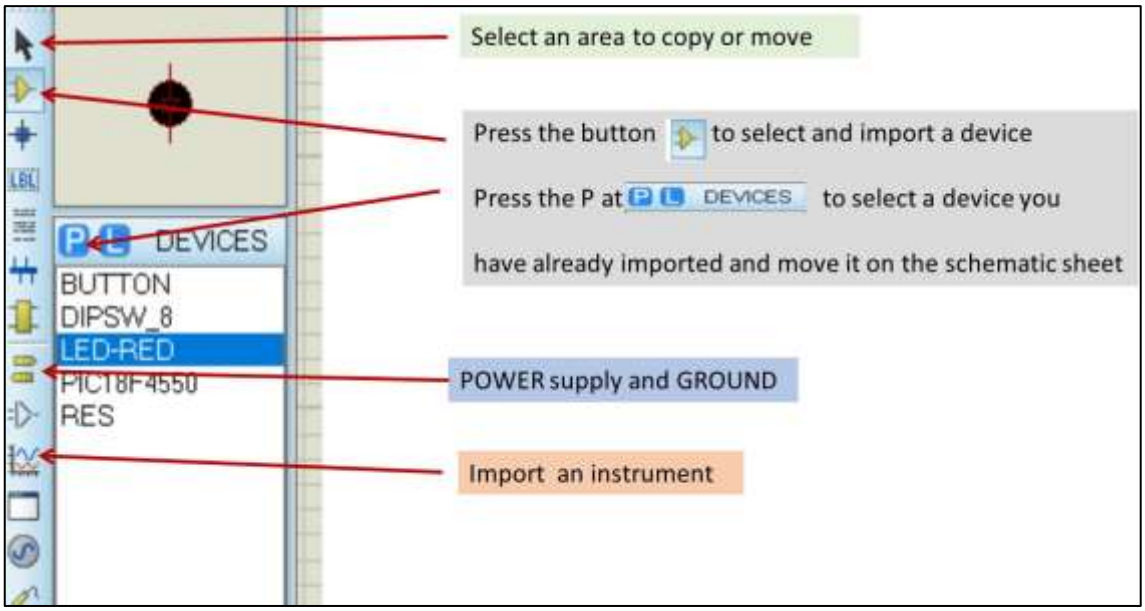

Figure 16. Toolbar

2.3 PIC and .hex file

To insert a microcontroller must press the button and then "P". In the window that appears, search for the device we want, as shown in Figure 17.

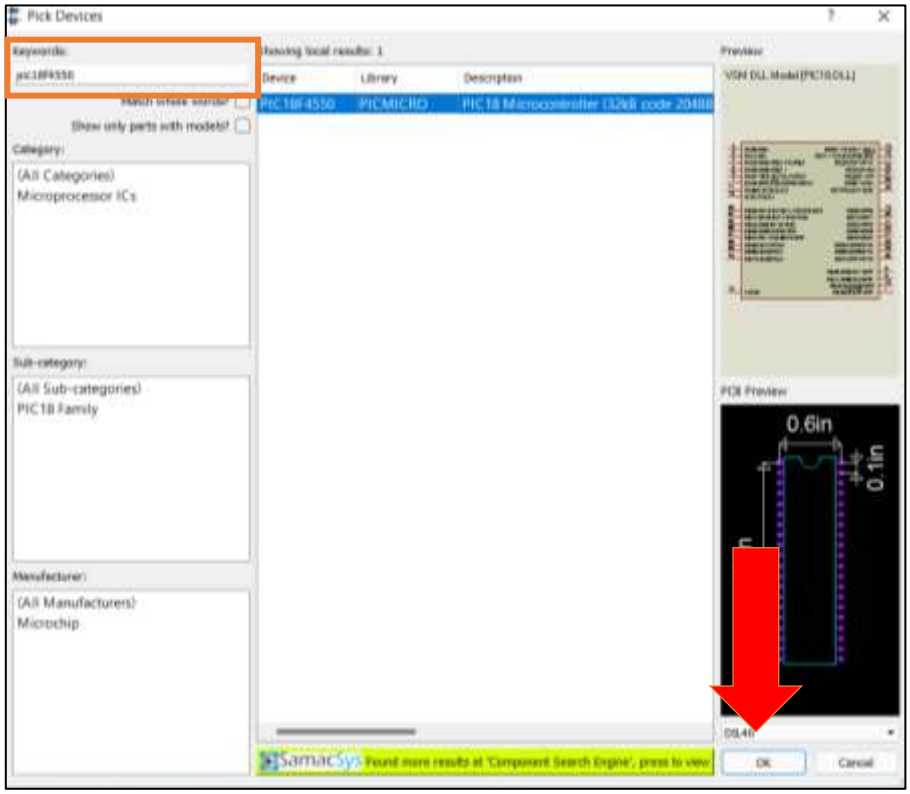

Figure 17. Pick device

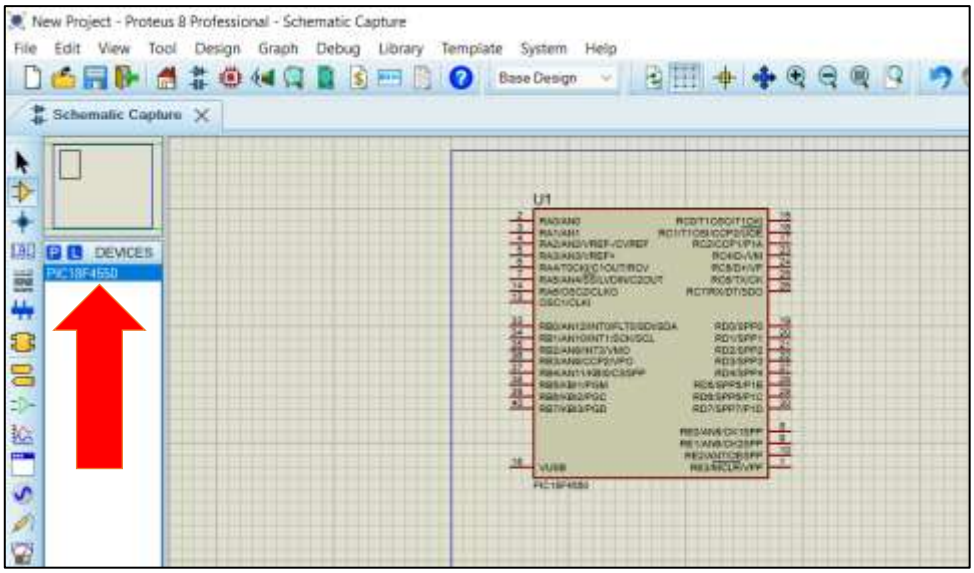

Figure 18. PIC18F4550

To load the .hex file into the microcontroller, double-click it to display the window shown in Figure 19.

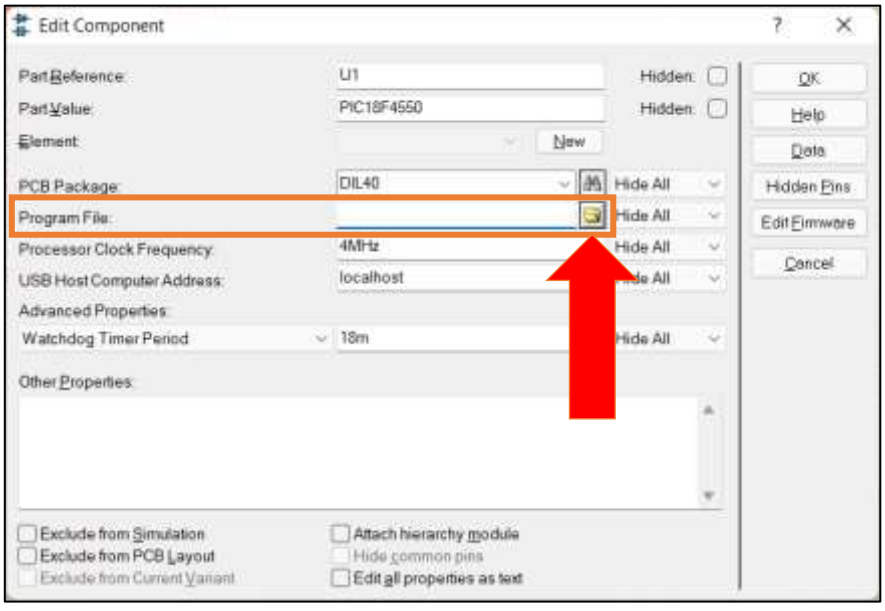

Figure 19. Edit component

In the "Program File" field, specify the path that have the .hex file.

Chapter 3: **Example**

3.1 Code

The example will use the PIC18F4550 to flash an LED every second. The .hex file is created, as shown in Figure 20.

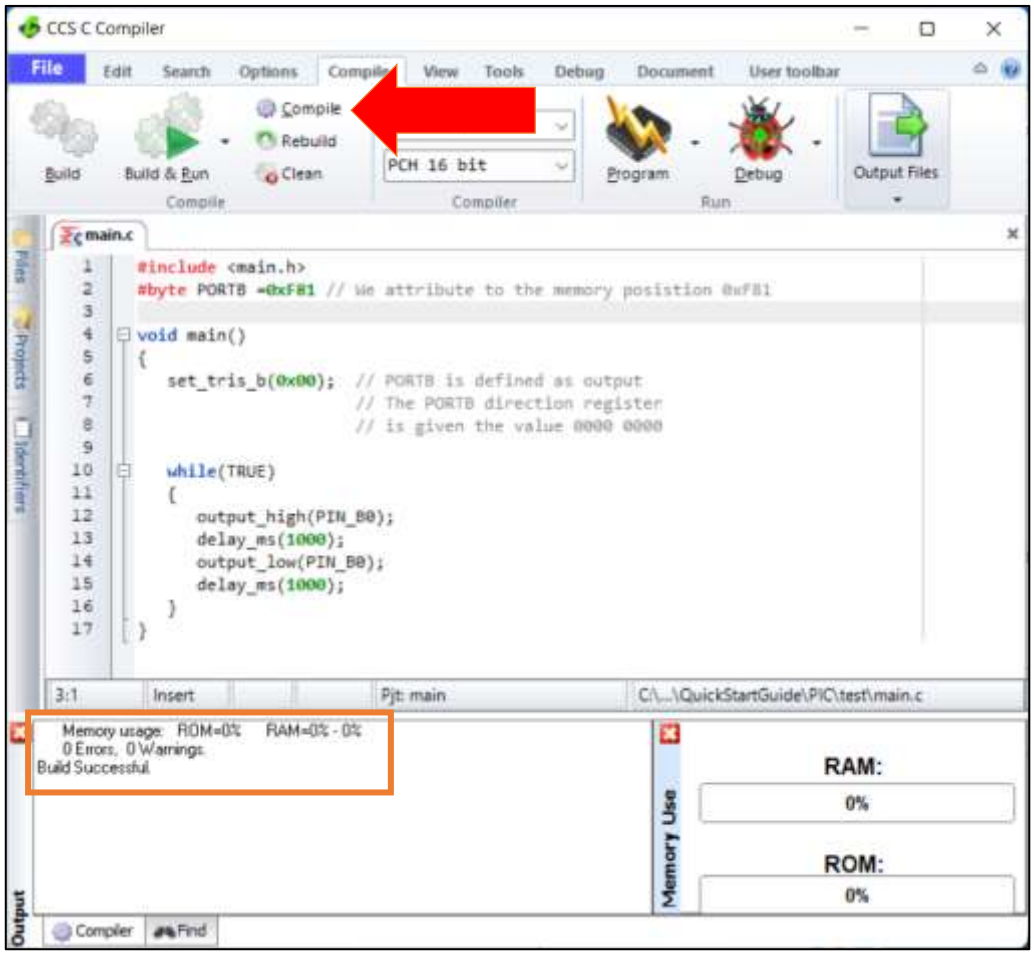

Figure 20. Compile the code

The code is:

```
#include <main.h>
#byte PORTB =0xF81 // We attribute to the memory posistion 0xF81

void main()
{
   set_tris_b(0x00);  // PORTB is defined as output
                      // The PORTB direction register
                      // is given the value 0000 0000

   while(TRUE)
   {
      output_high(PIN_B0);
      delay_ms(1000);
      output_low(PIN_B0);
```

```
        delay_ms(1000);
    }
}
```

3.2 Components and circuit connections

The components are placed as shown in Figure 21.

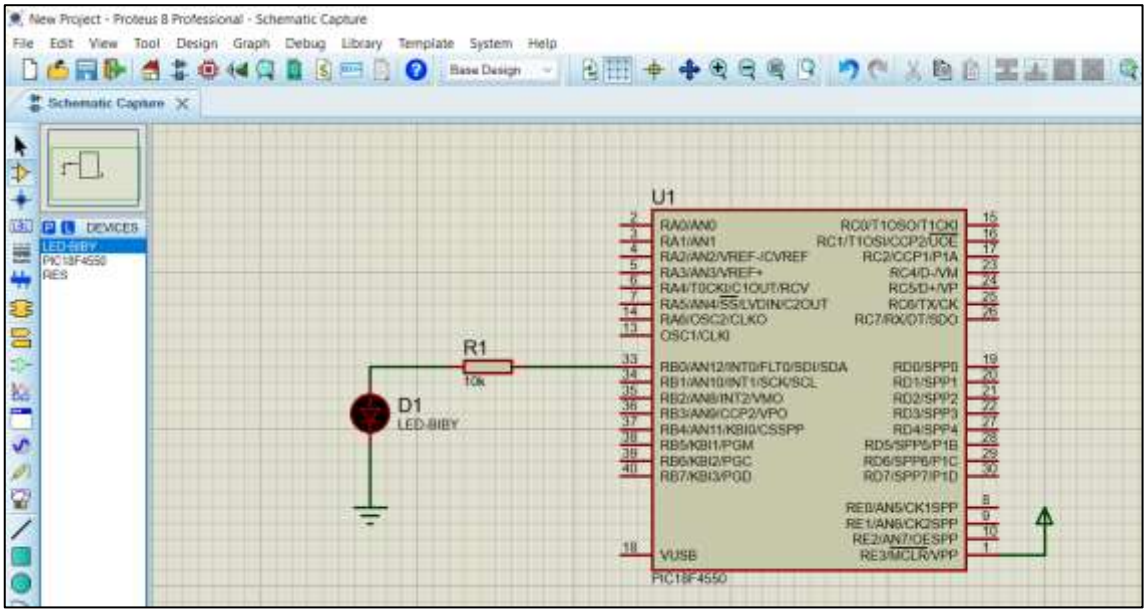

Figure 21. Schematic

By double-clicking on the resistor, we can adjust its value, as shown in Figure 22.

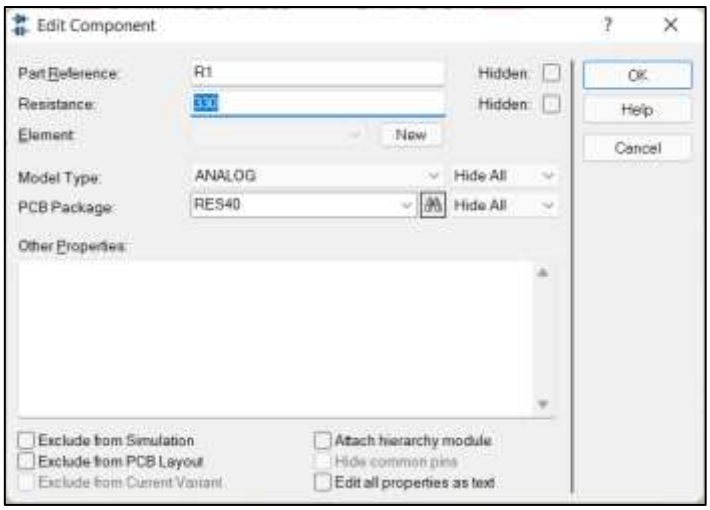

Figure 22. Adjusting the resistor value

Finally, we load the hexadecimal file into the microcontroller, as shown in Figure 23.

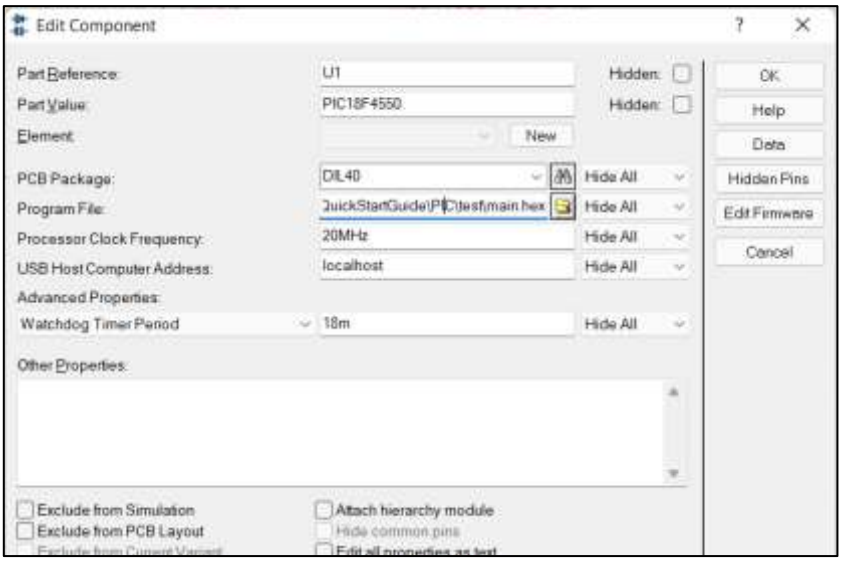

Figure 23. Load the .hex file

3.3 Simulation

In Proteus Design Suite we can run a simulation to check if the circuit is working properly. To perform the simulation, we must click on "Run the simulation", as shown in Figure 24.

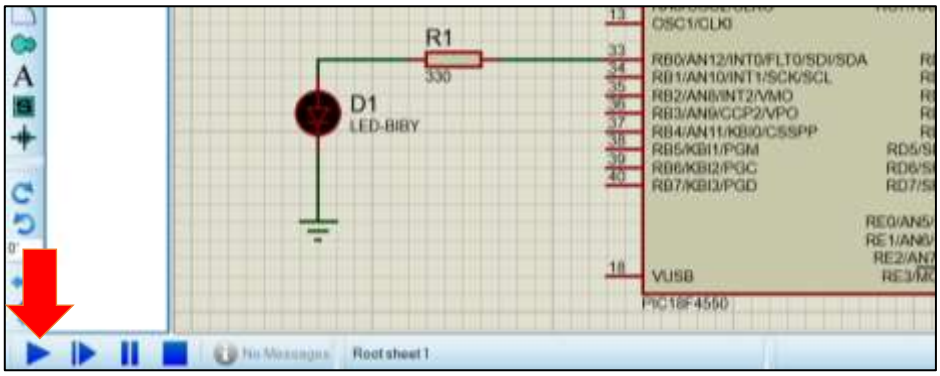

Figure 24. Start simulation

In Figure 25 we can see that the LED is ON.

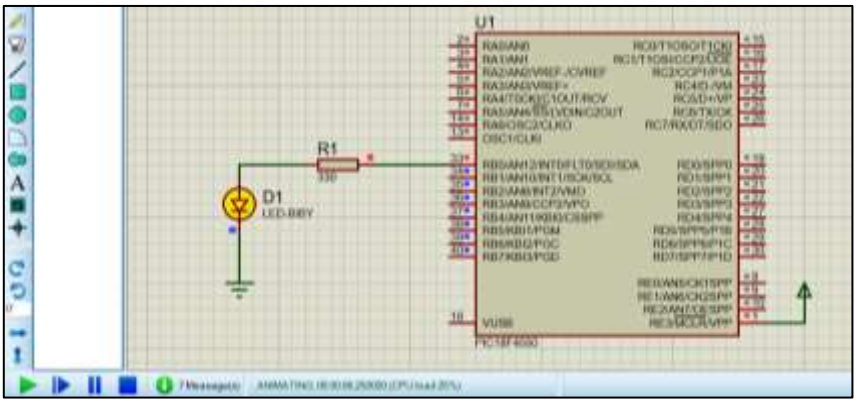

Figure 25. Perform simulation