|  |  |
| --- | --- |
| Experiment No : 5 | Seven segment |

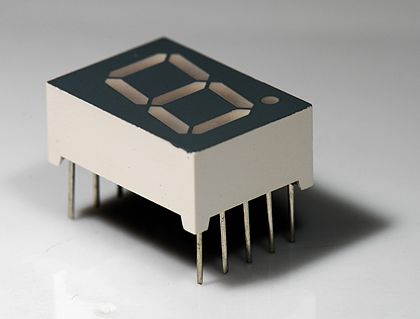
**Objectives:**

To learn about seven segment and how to connect it to PIC 16f877a and do some application.

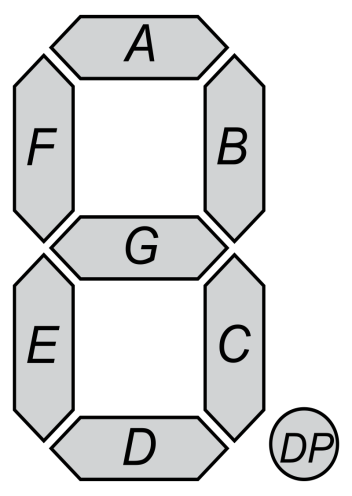
**Seven segment:**

A seven-segment display is a form of electronic [display device](https://en.wikipedia.org/wiki/Display_device) for displaying [decimal](https://en.wikipedia.org/wiki/Decimal) [numerals](https://en.wikipedia.org/wiki/Numeral_system) that is an alternative to the more complex [dot matrix displays](https://en.wikipedia.org/wiki/Dot_matrix_display).

Seven-segment displays are widely used in [digital clocks](https://en.wikipedia.org/wiki/Digital_clock), electronic meters, basic calculators, and other electronic devices that display numerical information.









**Functions**

In c, we can divide a large program into the basic building blocks known as function. The function contains the set of programming statements enclosed by {}. A function can be called multiple times to provide reusability and modularity to the C program. In other words, we can say that the collection of functions creates a program. The function is also known as procedure or subroutine in other programming languages.

**Advantage of functions in C:**

There are the following advantages of C functions.

* By using functions, we can avoid rewriting same logic/code again and again in a program.
* We can call C functions any number of times in a program and from any place in a program.
* We can track a large C program easily when it is divided into multiple functions.
* Reusability is the main achievement of C functions.
* However, Function calling is always a overhead in a C program.

The arranging of the function in the program is given below:

Deceleration about function

Void main () {

**Recall the function**

}

Define of function

**The general from for function is written as below:**

(The type of return variable)(Name of the function)(Type of sending variables)

**Note**: Each time the function is called, only one value is retrieved, but an unlimited number of variables can be sent to be processed within the function.

**There are four main shapes for function:**

1. Int micro (int)

This function send variable with type integer to function to process it and return value with type integer.

1. int micro ()

This function does not send variable to function to process it and the function return value with type integer

1. void micro (int)

This function send variable with type integer to function to process it and the function does not return value

1. void micro (void)

This function does not send any variable to function to process it and does not return value

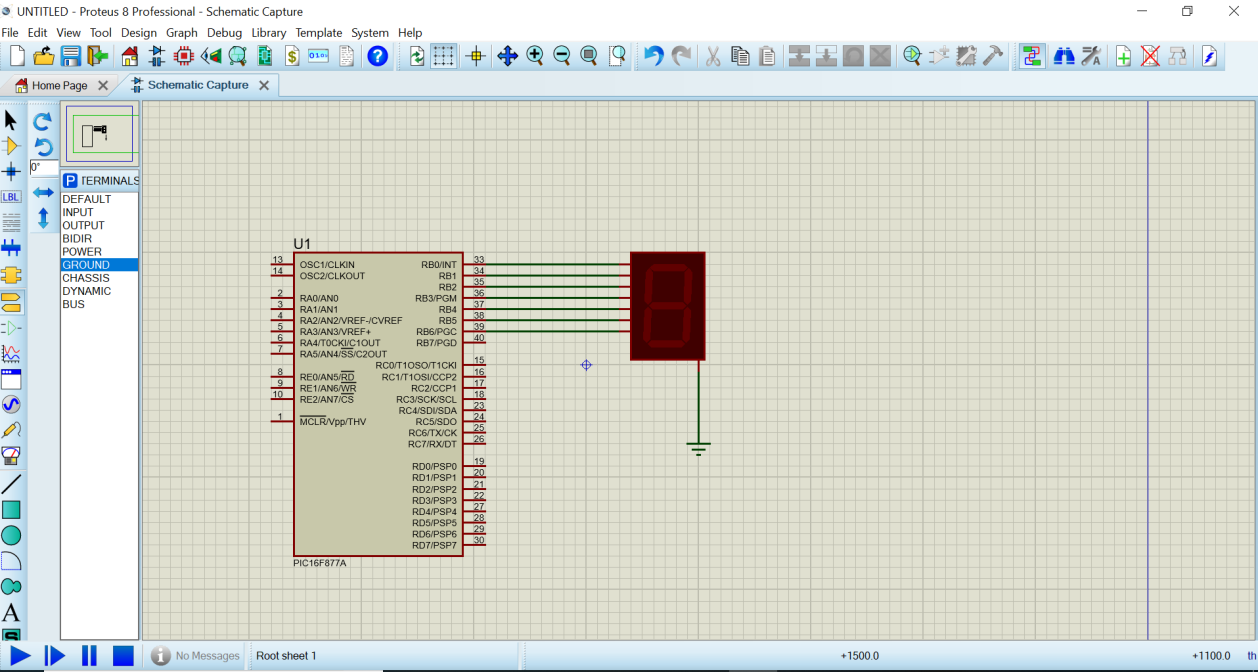
**Example 1:** counter using seven segment

The goal from this example to is to make an counter form 0 to 9 by using seven segment.

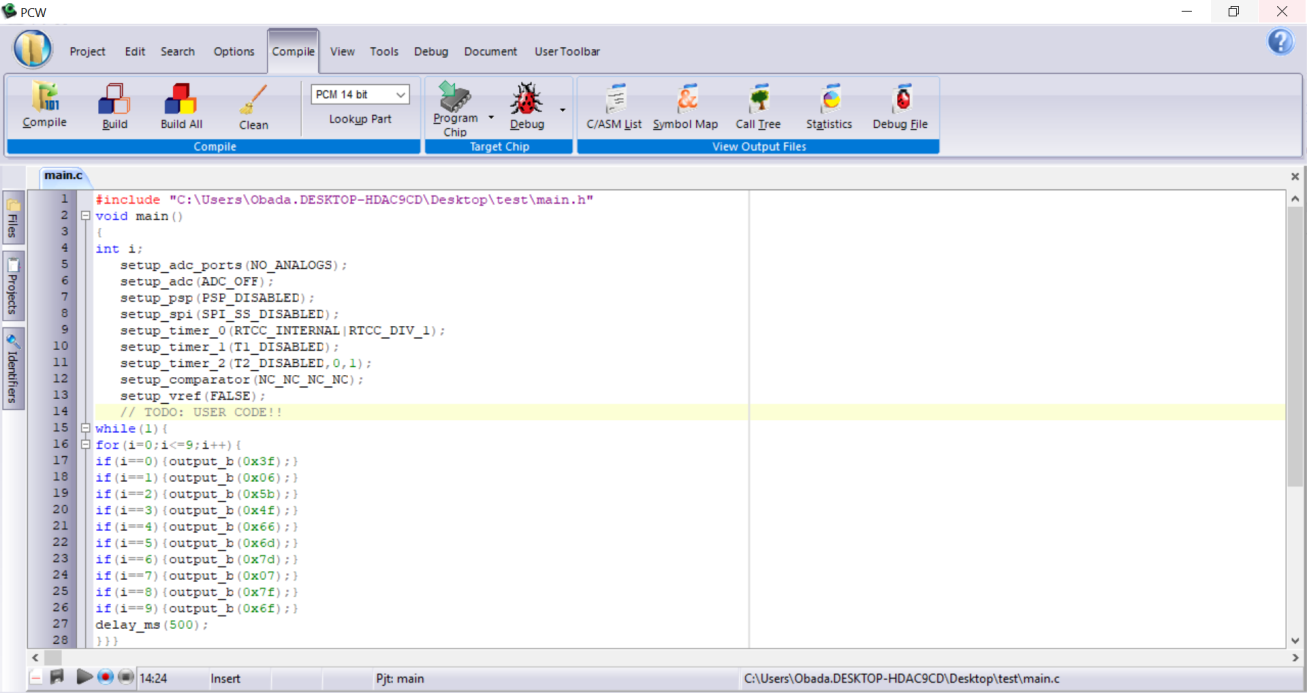
**Components:**

1. PIC16f877a form (Component mode)
2. 7seg-com-cathode(Component mode)
3. Ground (terminal mode )

**Proteus:**



**PIC C:**



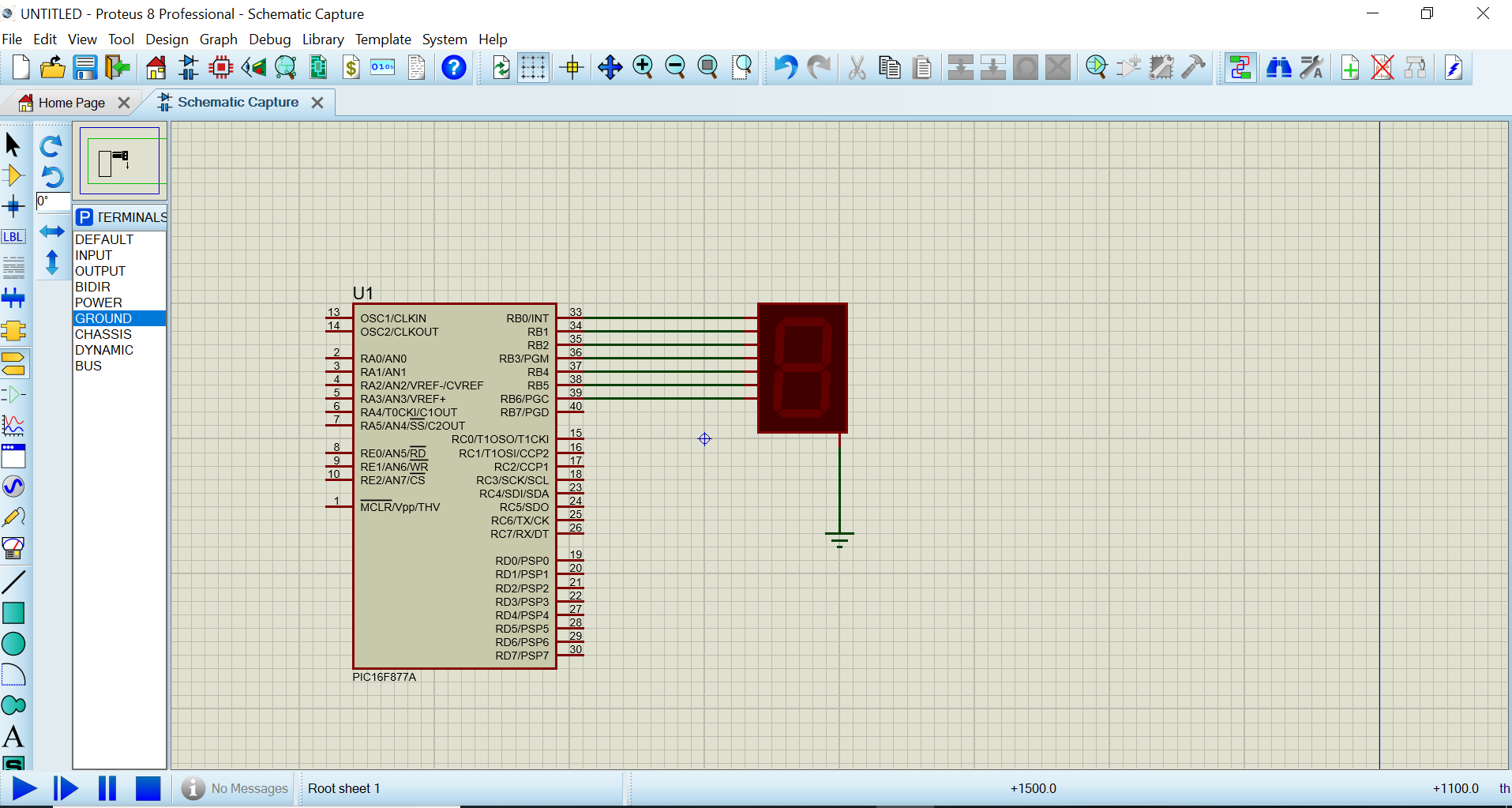
**Example 2:** counter using seven segment

The goal from this example to is to make an counter form 0 to 9 by using seven segment and using function type 1.

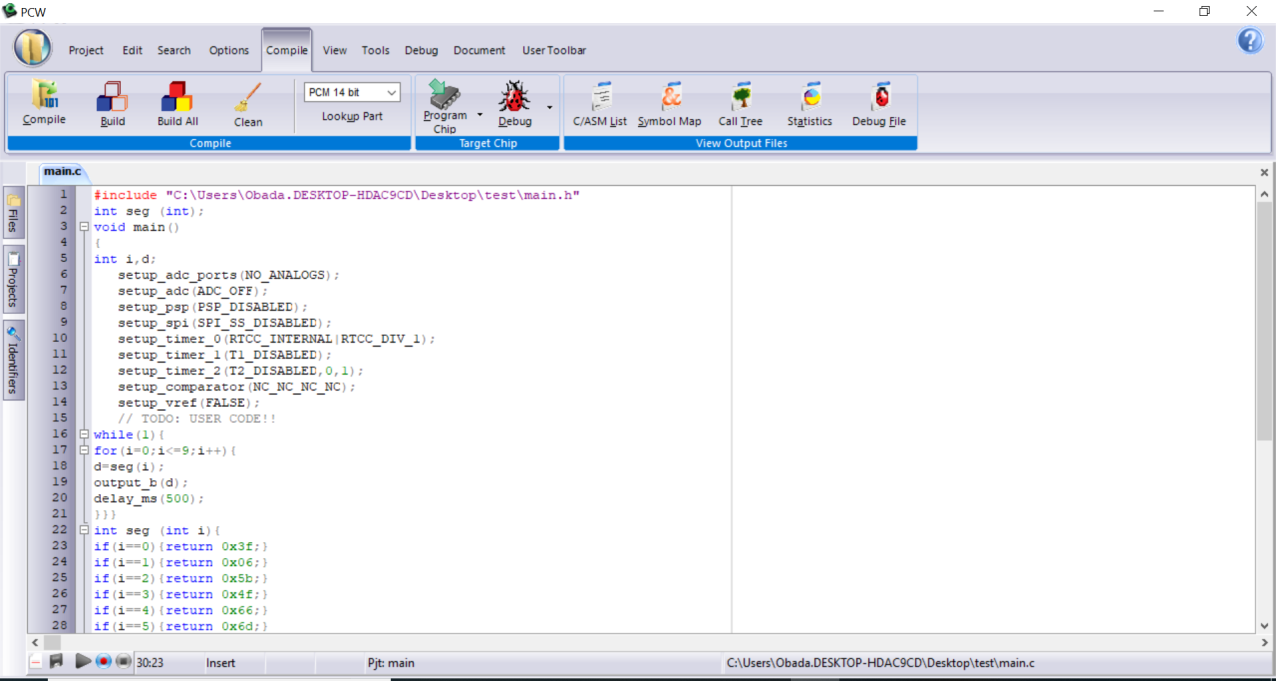
**Components:**

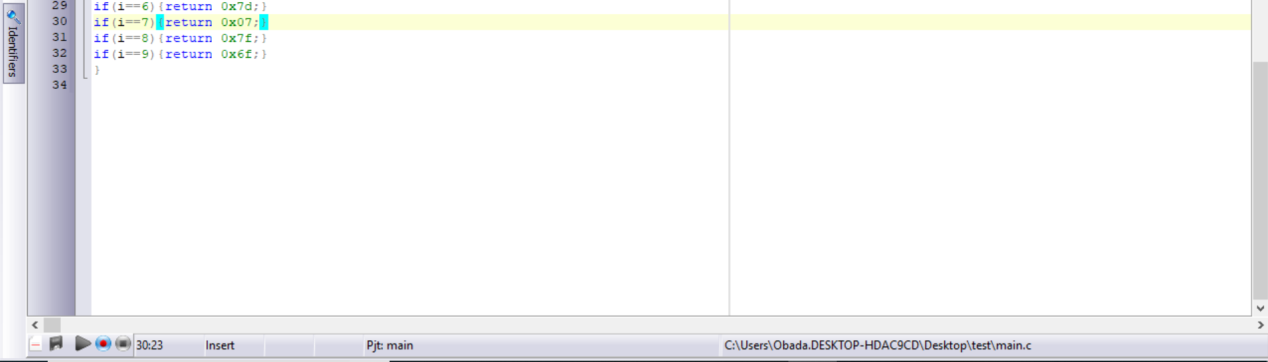
1. PIC16f877a form (Component mode)
2. 7seg-com-cathode(Component mode)
3. Ground (terminal mode )

**Proteus:**



**PIC C:**





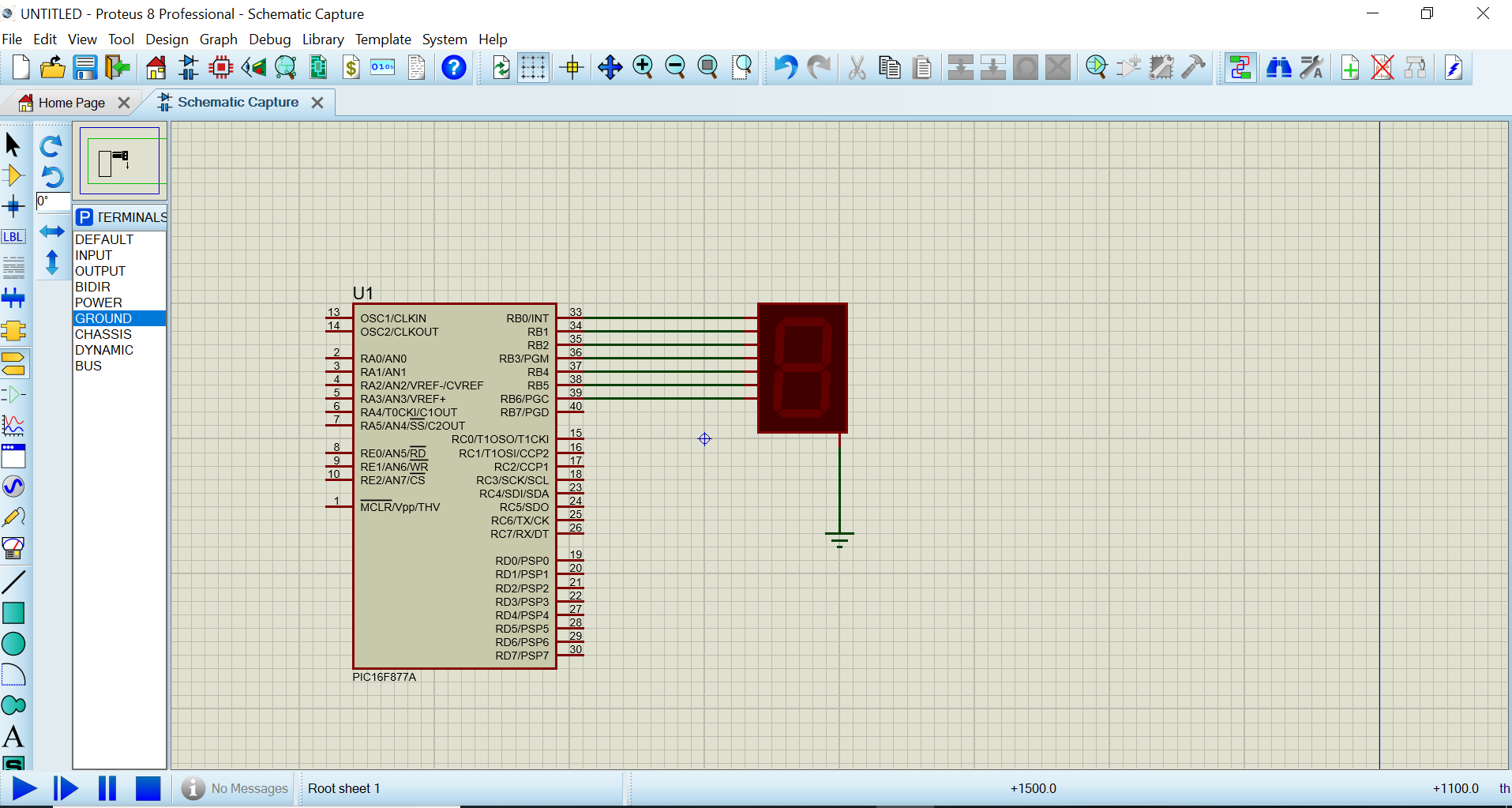
**Example 3:** counter using seven segment

The goal from this example to is to make an counter form 0 to 9 by using seven segment and using function type 3 .

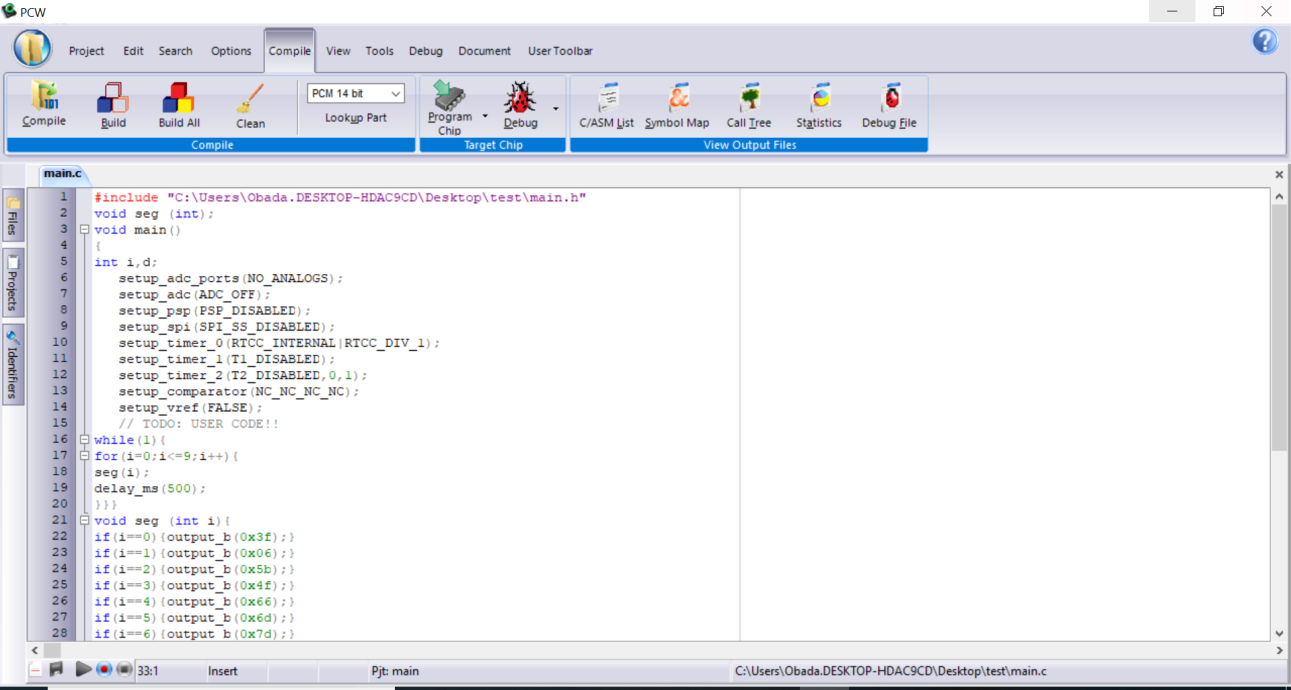
**Components:**

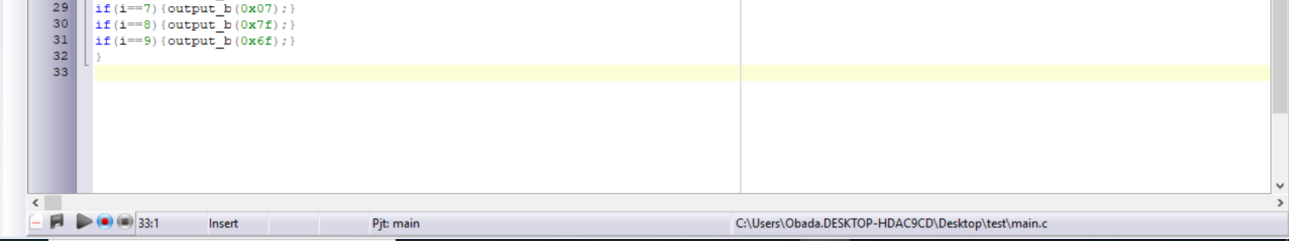
1. PIC16f877a form (Component mode)
2. 7seg-com-cathode(Component mode)
3. Ground (terminal mode )

**Proteus:**



**PIC C:**





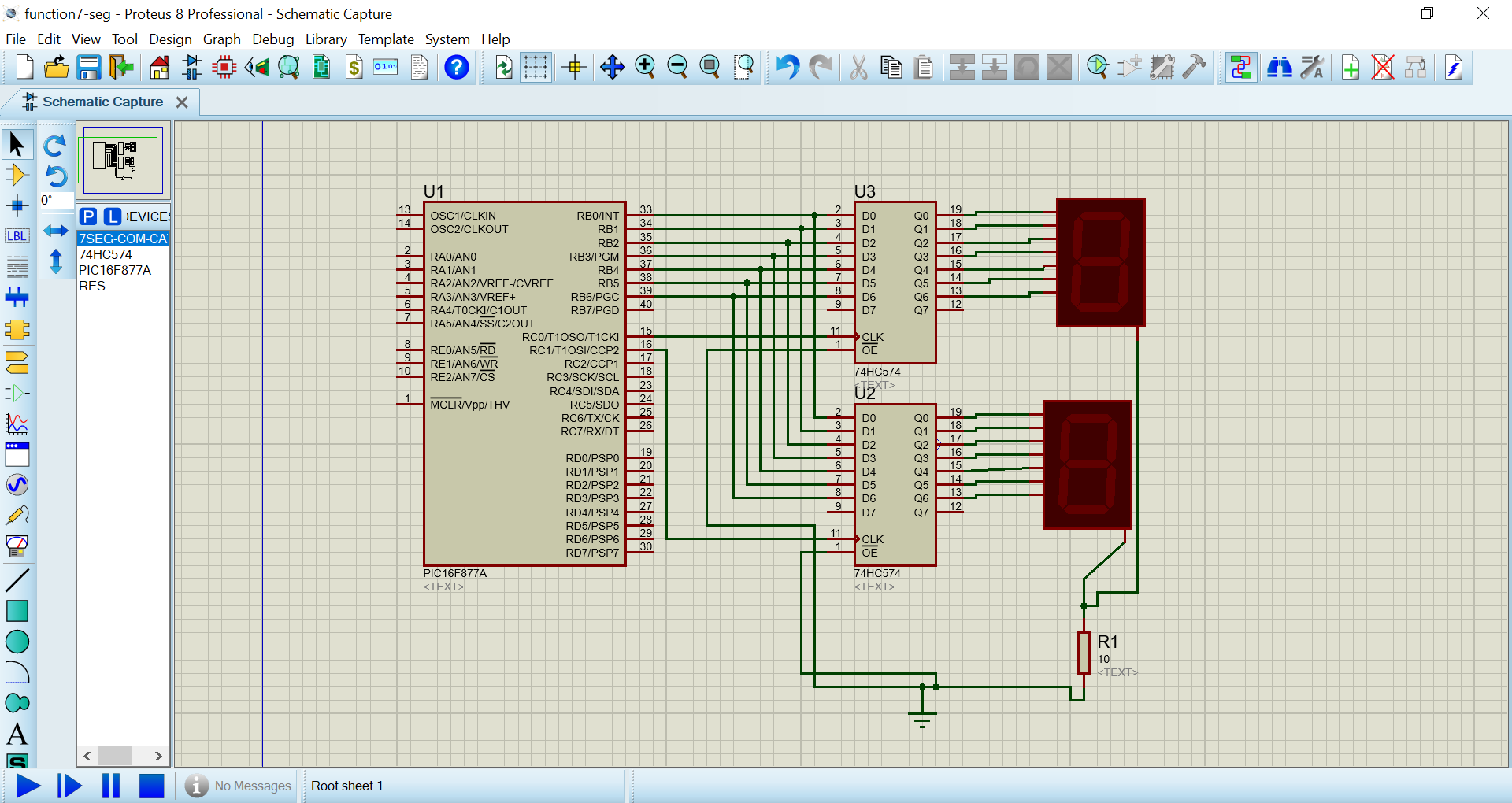
**Example 4:** counter using two seven segment

The goal from this example to is to make a counter form 0 to 99 by using seven segments

**Components:**

1. PIC16f877a form (Component mode)
2. 7seg-com-cathode(Component mode)
3. Ground (terminal mode )
4. 74hc574 (Component mode)
5. Res (Component mode)

**Proteus:**

****

**PIC C:**

