American International University- Bangladesh
Department of Electrical and Electronic Engineering
EEE 4103: Microprocessor and Embedded Systems Laboratory

Title: Building a decade counter using Arduino Uno and a 7 segment display.

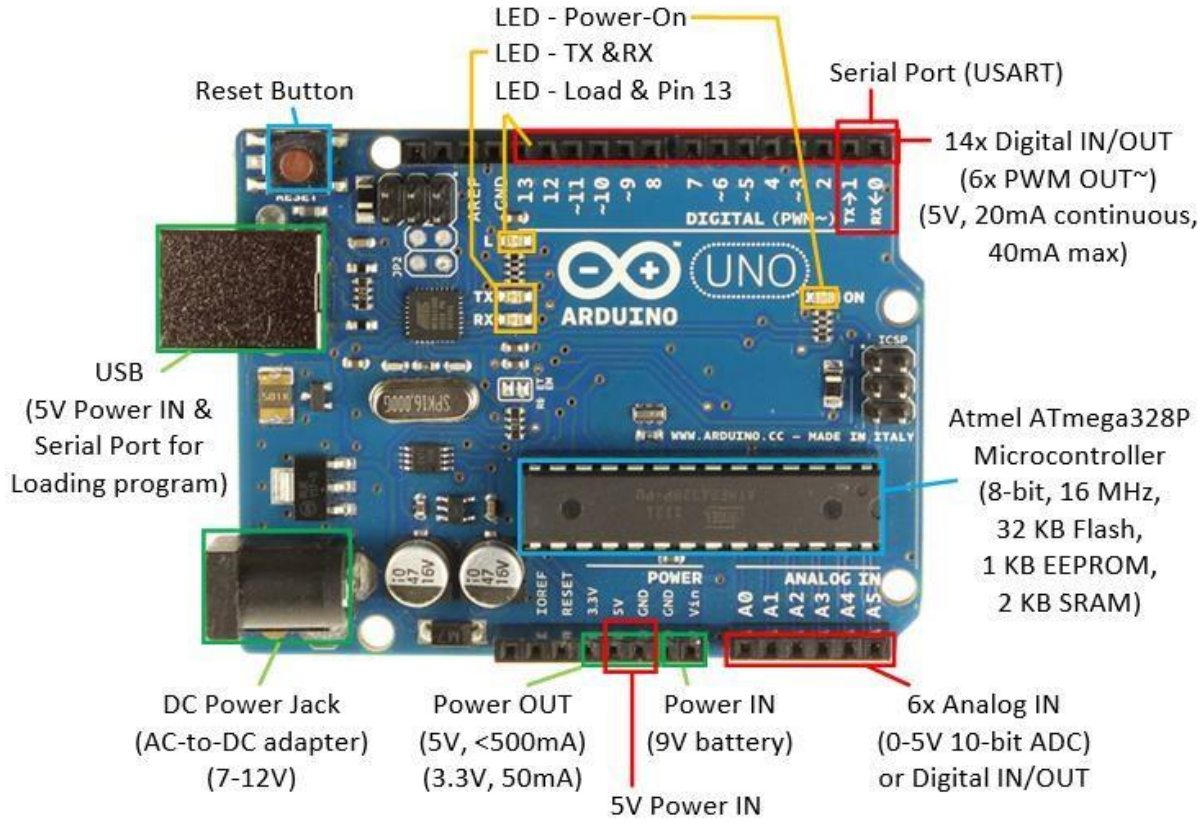
Introduction:

The objective of this experiment is to interface a 7 segment display with Arduino Uno.

Theory and Methodology:

Arduino is an open-source platform used for creating interactive electronics projects. Arduino consists of both a programmable microcontroller and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the microcontroller board. Arduino Uno also doesn't need a hardware circuit (programmer/ burner) to load a new code into the board. We can easily load a code into the board just using a USB cable and the Arduino IDE (that uses an easier version of C++ to write a code).

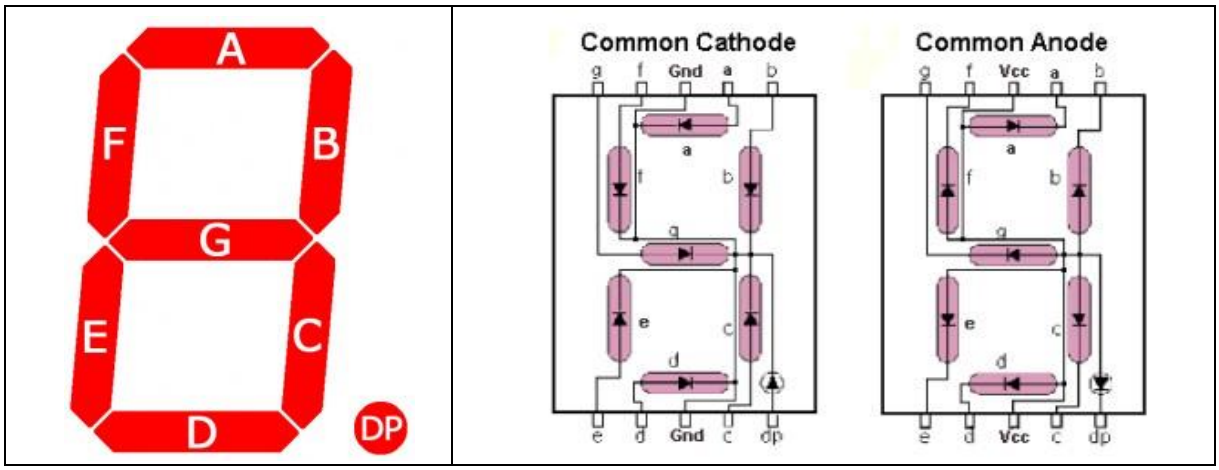
Pin configuration of the board (Arduino Uno R3) and Arduino Mega 2560:



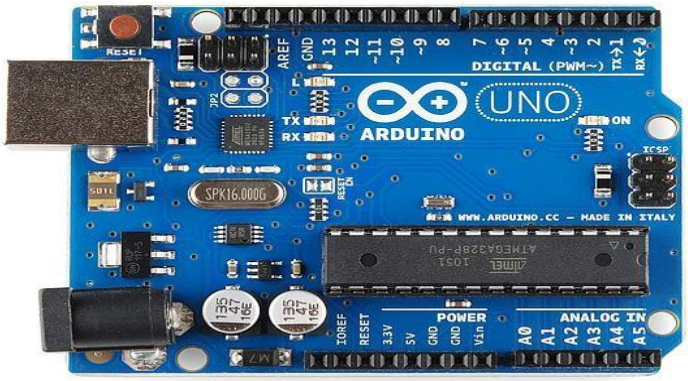
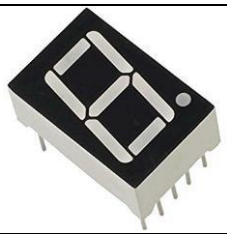
Basis of a 7 segment display:

A seven segment display consists of 7 LEDs/ segments all arranged in the shape of the digit "8". Most of the segment display has 8 segments (with a dot at the right side of the digit

representing the decimal point. All of the seven segments are named by sequence from "A" letter to "G" and "DP" for the decimal point. And each of the segments can be controlled as individually just like a regular LED.



7 Segment Display Apparatus:

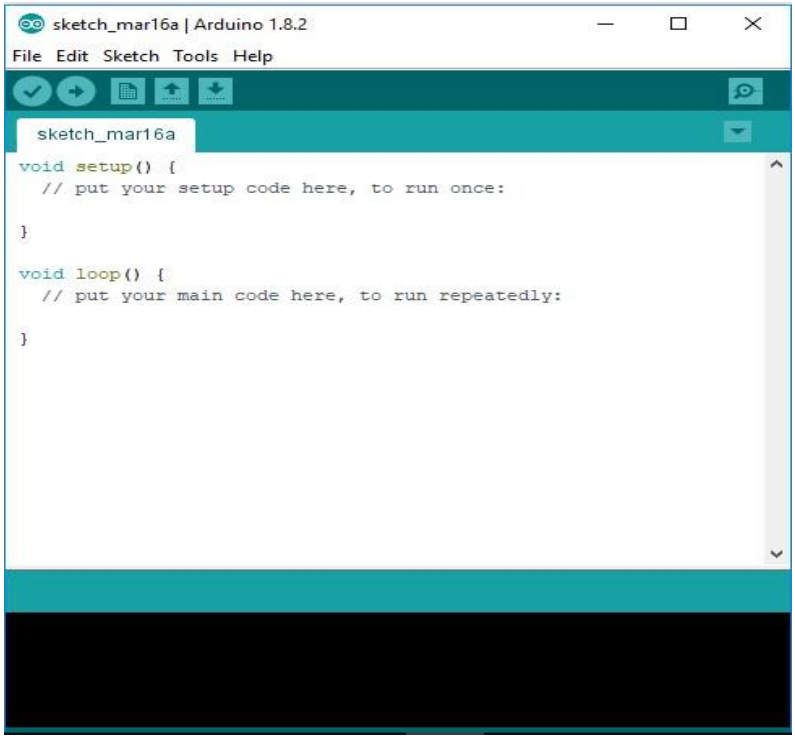
1) Arduino IDE (any version)	Software
2) Arduino Uno (R3) board or Arduino mega 2560.	
3) 7 segment display (HDSP5503).	

Experimental Procedure:

You have all you need to start your Lab work.

Using Arduino IDE to write code

1. Open the Arduino Uno IDE 1.8.2 and a blank sketch will open. The following window will come up on your PC: -



2. Now write following code to a blank sketch.

```
#define segA 2
#define segB 3
#define segC 4
#define segD 5
#define segE 6
#define segF 7
#define segG 8
int COUNT=0; void
setup()
{
      for (int i=2;i<9;i++)
           {
             pinMode(i, OUTPUT);
           }
}

void loop()
{
switch (COUNT)
              {
case 0:
              digitalWrite(segA, HIGH);
digitalWrite(segB, HIGH);
digitalWrite(segC, HIGH);
digitalWrite(segD, HIGH);
digitalWrite(segE, HIGH);
digitalWrite(segF, HIGH);
```

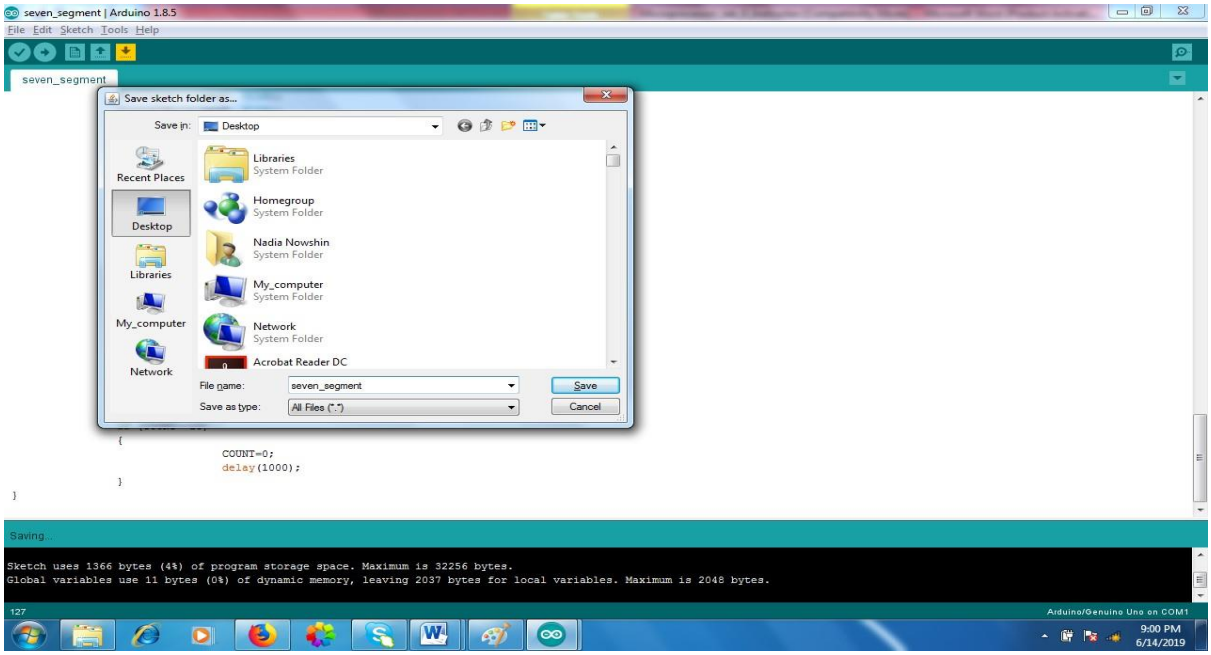
```
            digitalWrite(segG, LOW);
break;
```

```
            case
1:
            digitalWrite(segA, LOW);
digitalWrite(segB, HIGH);
digitalWrite(segC, HIGH);
digitalWrite(segD, LOW);
digitalWrite(segE, LOW);            digitalWrite(segF,
LOW);
            digitalWrite(segG, LOW);
break;
            case
2:
            digitalWrite(segA, HIGH);
digitalWrite(segB, HIGH);
digitalWrite(segC, LOW);
digitalWrite(segD, HIGH);
digitalWrite(segE, HIGH);
digitalWrite(segF, LOW);
            digitalWrite(segG, HIGH);
break;
            case
3:
            digitalWrite(segA, HIGH);
digitalWrite(segB, HIGH);
digitalWrite(segC, HIGH);
digitalWrite(segD, HIGH);
digitalWrite(segE, LOW);            digitalWrite(segF,
LOW);
            digitalWrite(segG, HIGH);
break;
            case
4:
            digitalWrite(segA, LOW);
digitalWrite(segB, HIGH);
digitalWrite(segC, HIGH);
digitalWrite(segD, LOW);
digitalWrite(segE, LOW);            digitalWrite(segF,
HIGH);
            digitalWrite(segG, HIGH);
break;
            case
5:
            digitalWrite(segA, HIGH);
digitalWrite(segB, LOW);
digitalWrite(segC, HIGH);
digitalWrite(segD, HIGH);
digitalWrite(segE, LOW);            digitalWrite(segF,
HIGH);
            digitalWrite(segG, HIGH);
break;
            case
6:
            digitalWrite(segA, HIGH);
```

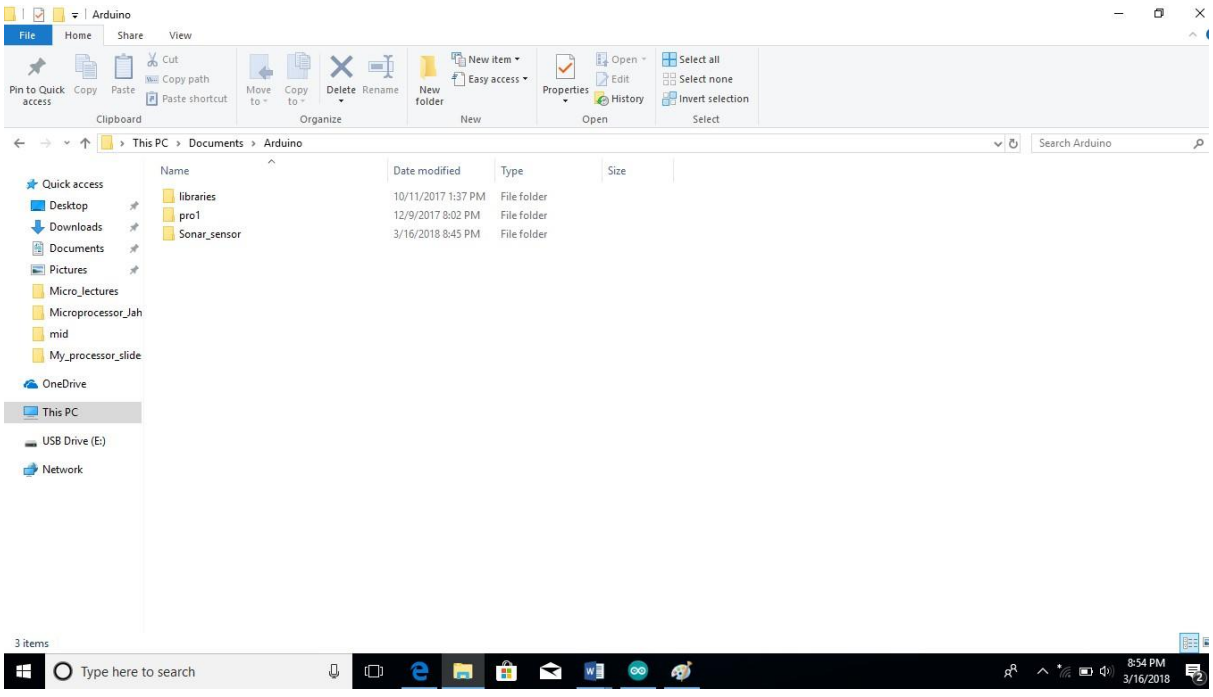
```
            digitalWrite(segB, LOW);
digitalWrite(segC, HIGH);
digitalWrite(segD, HIGH);
digitalWrite(segE, HIGH);
digitalWrite(segF, HIGH);
            digitalWrite(segG, HIGH);
break;
              case
7:
            digitalWrite(segA, HIGH);
digitalWrite(segB, HIGH);
digitalWrite(segC, HIGH);
digitalWrite(segD, LOW);
digitalWrite(segE, LOW);              digitalWrite(segF,
LOW);
            digitalWrite(segG, LOW);
break;
              case
8:
            digitalWrite(segA, HIGH);
digitalWrite(segB, HIGH);
digitalWrite(segC, HIGH);
digitalWrite(segD, HIGH);
digitalWrite(segE, HIGH);
digitalWrite(segF, HIGH);
            digitalWrite(segG, HIGH);
break;
              case
9:
            digitalWrite(segA, HIGH);
digitalWrite(segB, HIGH);
digitalWrite(segC, HIGH);
digitalWrite(segD, HIGH);
digitalWrite(segE, LOW);              digitalWrite(segF,
HIGH);
            digitalWrite(segG, HIGH);
break;

            break;
            }
            if (COUNT<10)
            {
                        COUNT++;
delay(1000);
            }
            if (COUNT==10)
            {
                        COUNT=0;
delay(1000);
            }
}
```

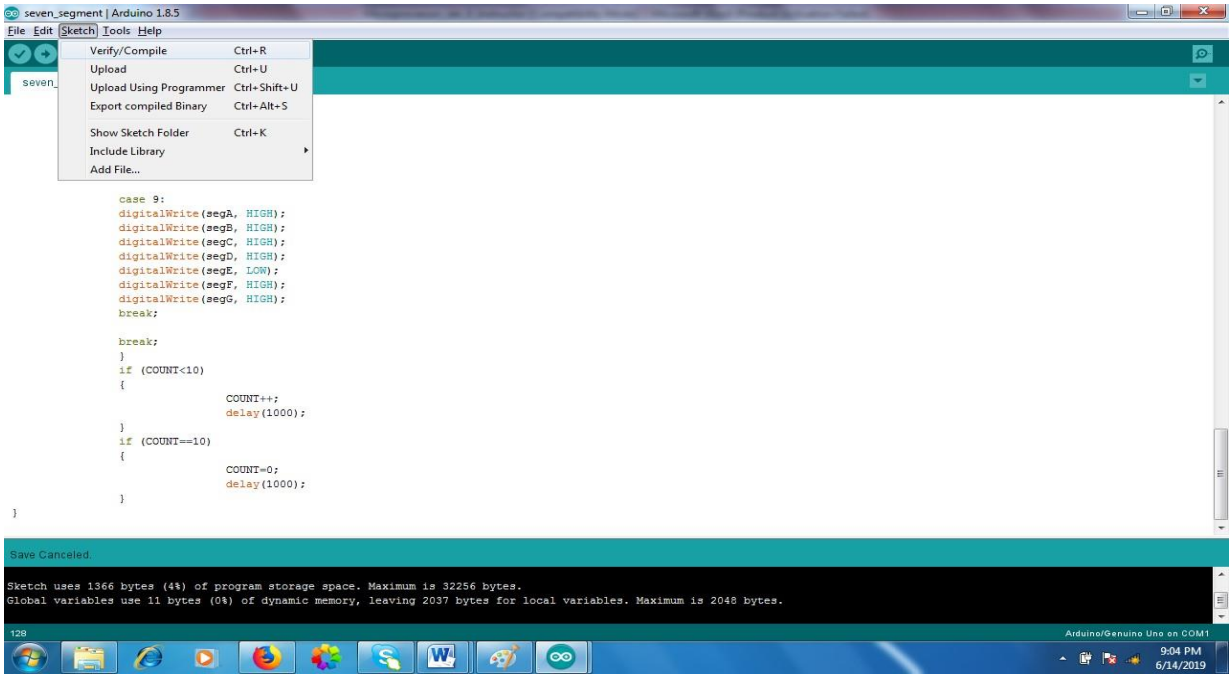
3. After the writing the code you have to save the sketch, go to File->Save As->give a File name (Seven_segment) -> Select Save.



N.B. After saving your code a sketch file with the desired file name will be stored in a folder with same file name.



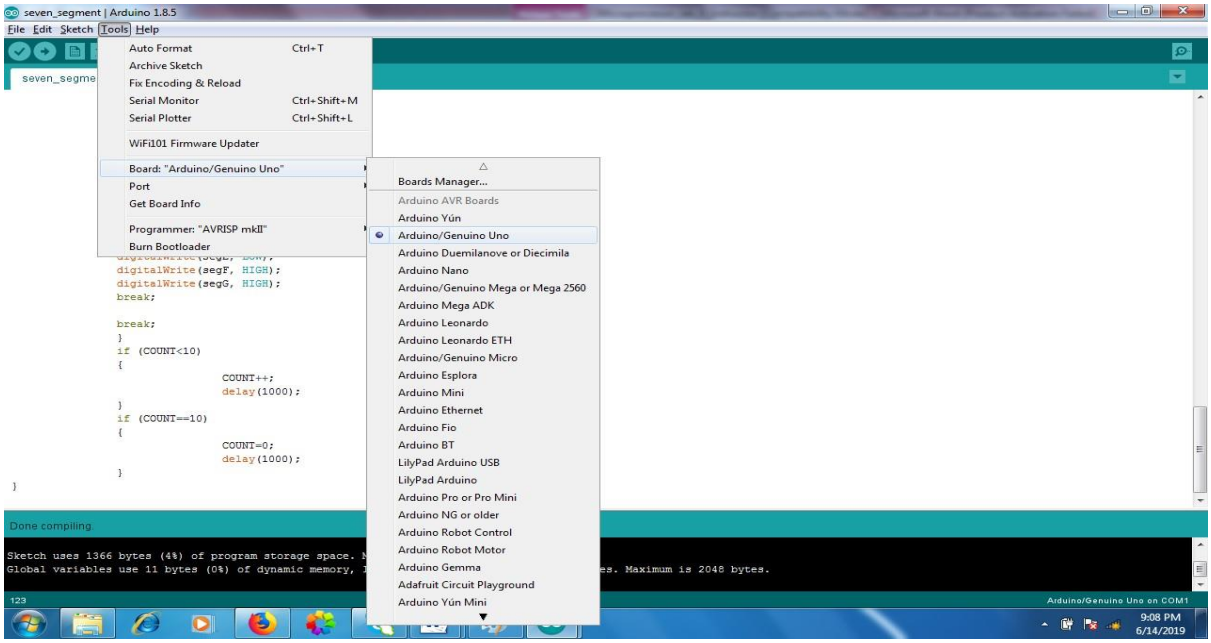
4. Now you need to verify/compile your code to find out and correct the errors, go to Sketch>Verify/Compile.

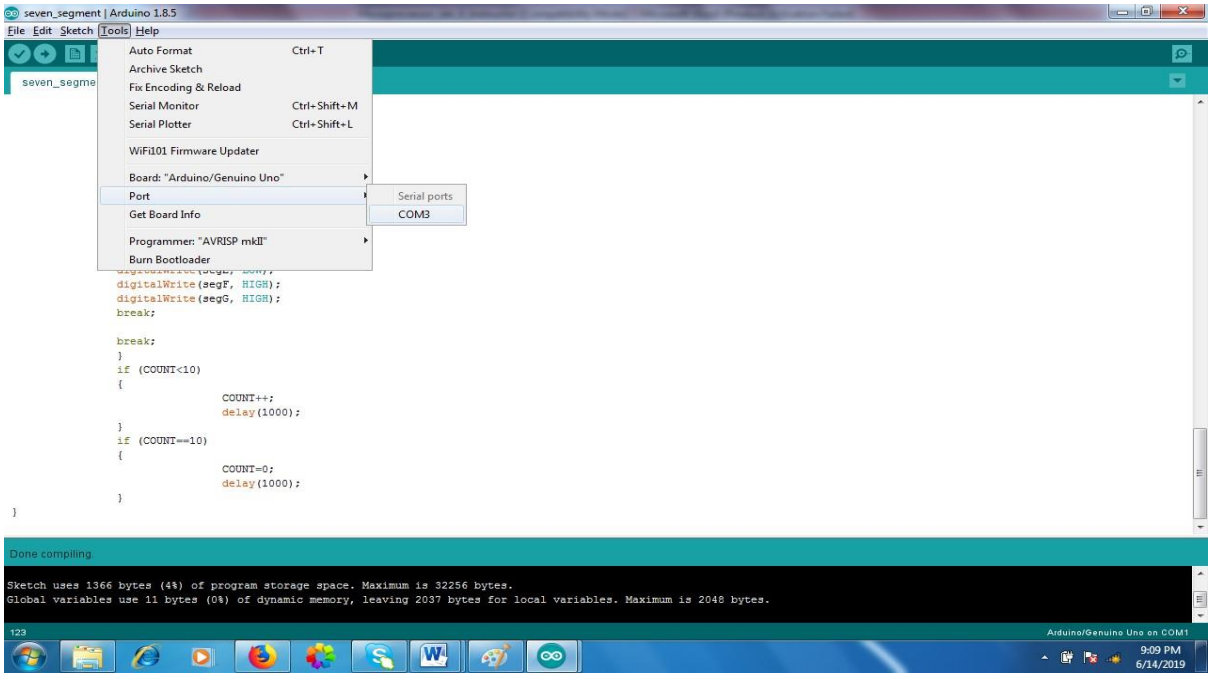


5. After compiling is done you need to upload your code into the Arduino Uno board. To upload the code connect you Arduino Uno R3 board to your PC with USB cable. Before uploading the code select the board type and port at your Arduino IDE, go to

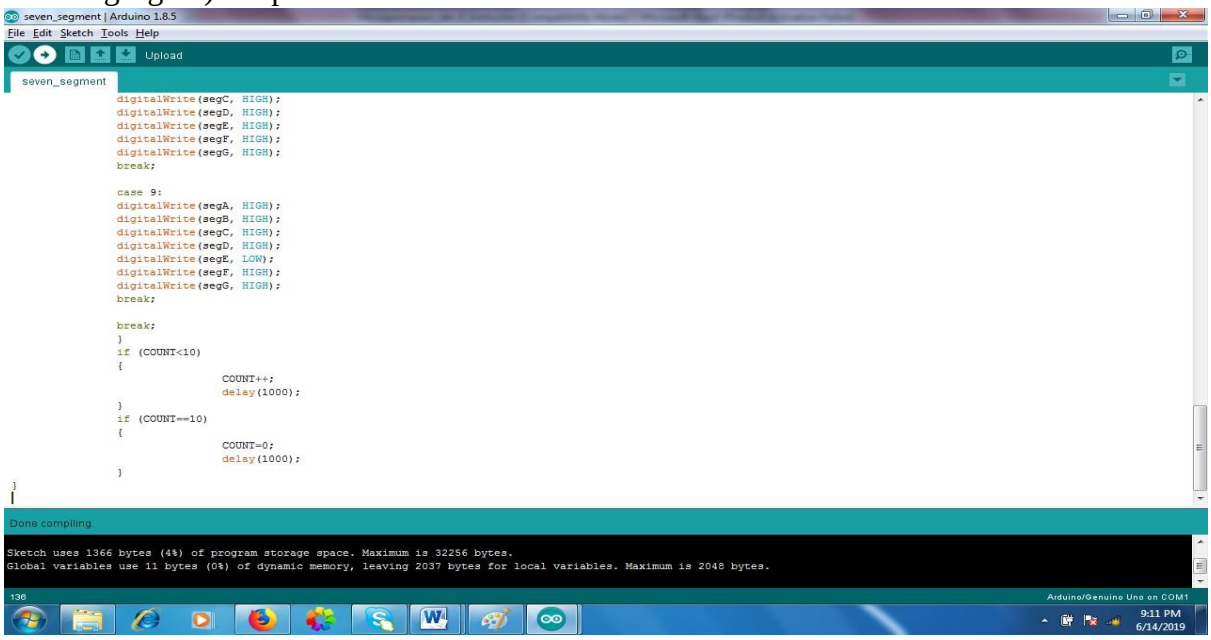
   - Tools-> Board:"Arduino/Genuino Uno" -> Arduino/Genuino Uno.

   - Tools->Ports-> COMx





After you have selected the board and port select the upload option at the Arduino IDE (see the following figure) to upload the code.



Familiarization with the Arduino Commands

In this section,

1. We will learn about some common Arduino commands that will help write code.
2. This section also focuses on the standard Library functions associated with the IDE.

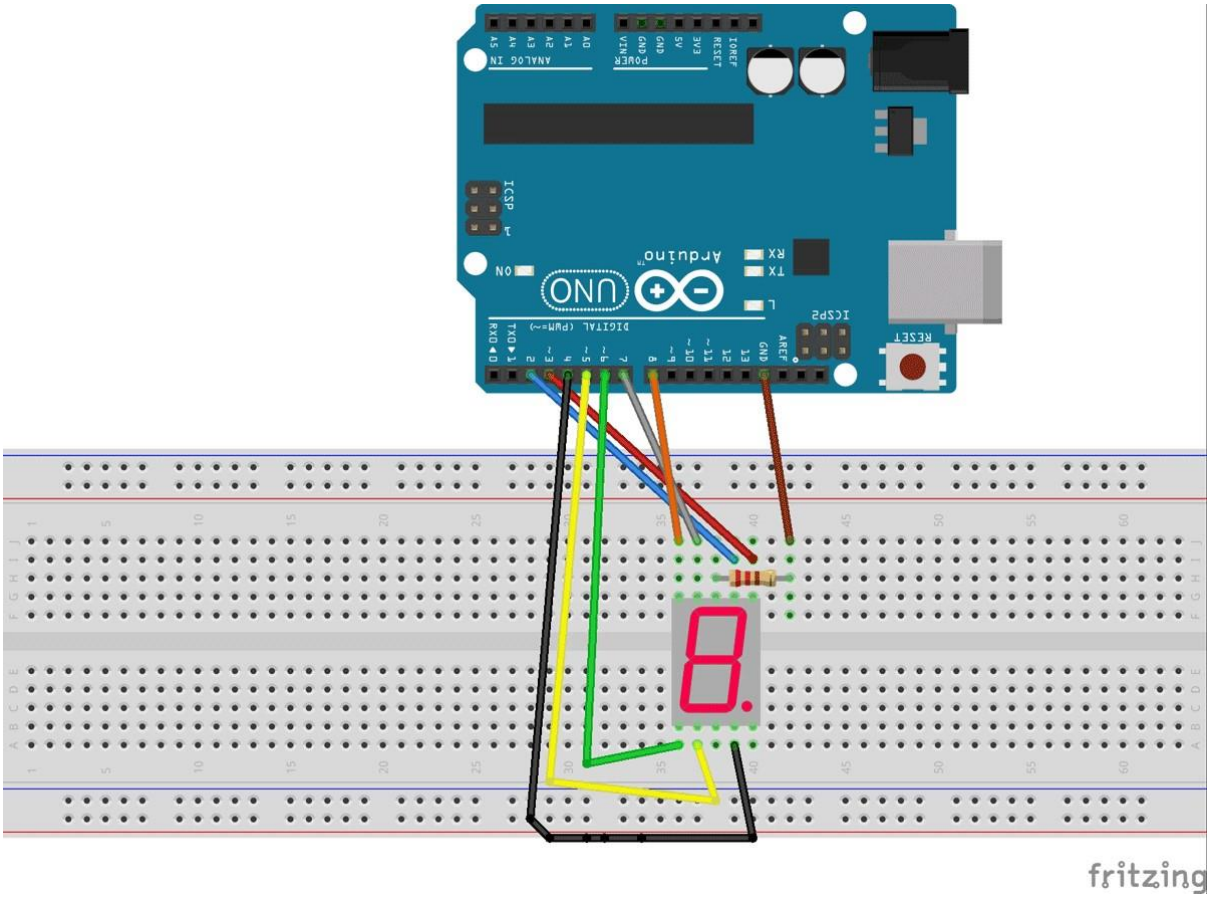
a) ****pinMode(X, INPUT) or pinMode(X,OUTPUT) **** this command will configure any pin at the Arduino board as either input/output.

b) ****digitalWrite(X, LOW) or digitalWrite(X, HIGH) **** this command will provide a HIGH/LOW value to any digital output pin at the Arduino board .

c) **** delayMicroseconds(X) **** this command will provide a delay of "X" microseconds after executing a line of command.

d) ******** Serial.print("Text") ******** this command will print a text at the serial monitor of the Arduino IDE

Setting up the 7 segment display with the Arduino Uno board: (for common cathode)



For common anode just the difference will be: the resistor will be connected to the VCC from common points instead of GND.

The connections which are done for 7 segment display are given below:

E to PIN 6 of ARDUINO UNO

D to PIN 5 of ARDUINO UNO

C to PIN 4 of ARDUINO UNO

H or DP to PIN 9 ///not needed as we are not using decimal point

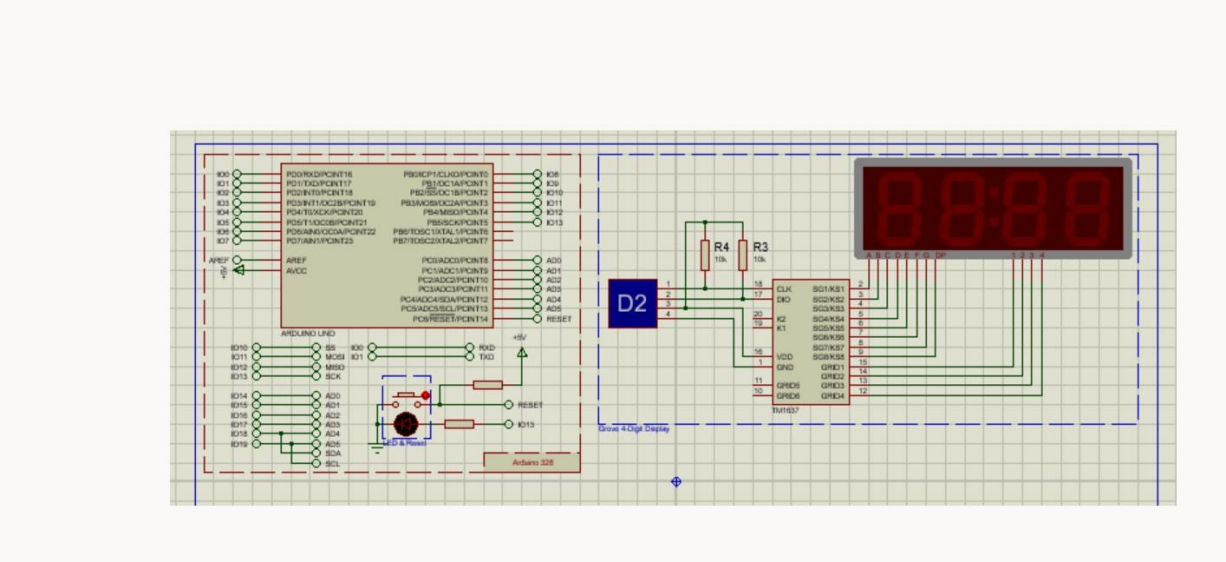
B to PIN 3 of ARDUINO UNO

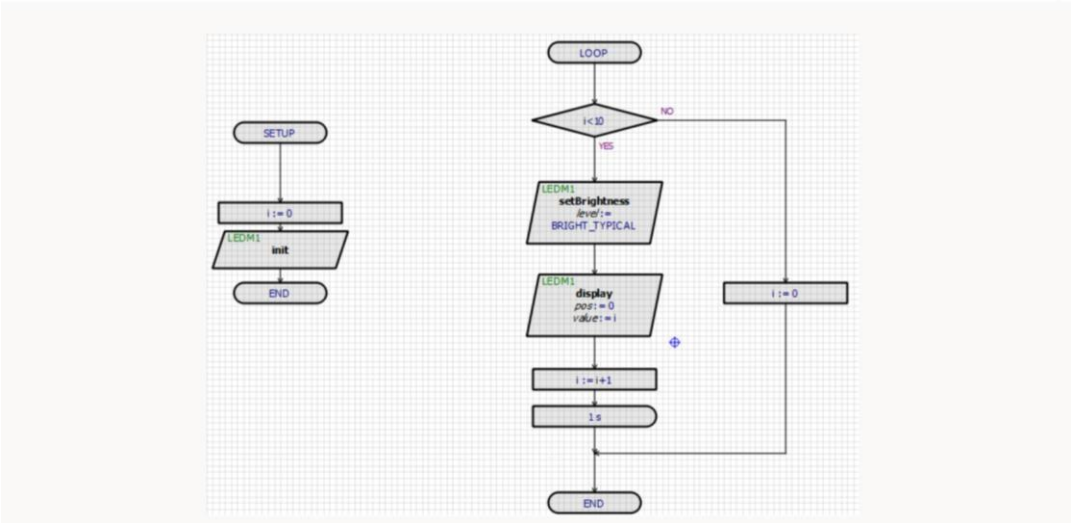
A to PIN 2 of ARDUINO UNO

F to PIN 7 of ARDUINO UNO G to PIN 8 of ARDUINO UNO

ground through a100Ω resistor to the GND of ARDUINO UNO.

Simulation using proteus:





Questions for report writing:

1) Include all codes and scripts into lab report following the writing template mentioned in appendix A of Laboratory Sheet Experiment 8.

Assignments:

Modify your experiment to show only the even/odd digits within the range (0-9) with a delay of two seconds.

Precautions:

Here before using the seven segment display you can individually check all the seven LEDs in the segments by using the multimeter in the diode mode. But one issue must be taken under consideration before checking that is each segment should not be power with a voltage greater than 4 volt otherwise the display will damaged permanently.

Reference(s):

1) https://www.arduino.cc/.
2) HDSP5503 Datasheet.