

EXPERIMENT NO. 6 (Group B)

- **Aim:** Write a program that asks the user for a number and outputs the number squared that is entered
- **Outcome:** Connectivity, configuration and serial communication with Arduino.
- **Hardware Requirement:** Arduino, USB Cable etc.
- **Software Requirement:** Arduino IDE

Theory:

● Arduino Serial Monitor for Beginners

Arduino serial monitor for beginners in electronics. Send and receive data between the serial monitor window on a computer and an Arduino. The serial monitor is a utility that is part of the Arduino IDE. Send text from an Arduino board to the serial monitor window on a computer. In addition, send text from the serial monitor window to an Arduino board. Communications between the serial monitor and Arduino board takes place over the USB connection between the computer and Arduino.

● Demonstration of the Arduino Serial Monitor for Beginners

Part 2 of this Arduino tutorial for beginners shows how to install the Arduino IDE. In addition, it shows how to load an example sketch to an Arduino. It is necessary to know how to load a sketch to an Arduino board in this part of the tutorial. Therefore, first finish the previous parts of this tutorial before continuing with this part. A sketch loaded to an Arduino board demonstrates how the serial monitor works in the sub-sections that follow.

● Load an Example Sketch that uses the Serial Monitor to an Arduino Board

Start the Arduino IDE application. Select File → Examples → 04.Communication → ASCIITable from the top Arduino IDE menu bar. As a result, the ASCIITable example sketch opens in a new Arduino IDE window. Upload the ASCIITable example sketch to the Arduino Uno or MEGA 2560 board.

After the ASCIITable sketch is uploaded, nothing is seen to happen. This is because this example sketch sends text out of the USB port of the Arduino board. Because there is nothing running on the computer to receive this text, nothing is seen.

● How to Open the Arduino Serial Monitor Window for Beginners

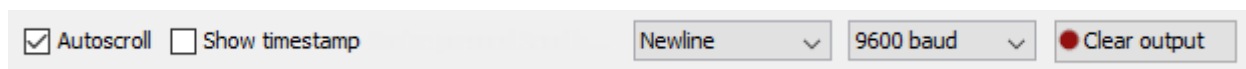
The following image shows the location of the serial monitor window icon on the Arduino IDE toolbar. A red dot near the top right of the image shows the serial monitor toolbar icon location.

MEGA 2560.

Pushing the RESET button in holds the board in reset. This means that the sketch currently loaded on the board stops running. Releasing the RESET button takes the board out of reset. As a result, the sketch currently loaded on the Arduino starts running from the beginning again.

- **Clear the Serial Monitor Window Receive Area**

The red dot in the image below shows the location of the Clear output button at the bottom of the serial monitor window. Click the Clear output button and text is cleared from the receive area of the serial monitor window. Reset the Arduino, and the receive area fills with text from the ASCII Table sketch again.



Serial Monitor Window Clear Output Button

- **What the ASCII Table Sketch Does**

ASCII stands for American Standard Code for Information Interchange. ASCII is a standard way that uses numbers to represent various characters. For example, the decimal number 65 represents the letter A. Another example is the decimal number 125 represents a closing brace: }. This allows computers to send and receive text by sending and receiving numbers. For example when a computer receives the number 65, it knows to display the letter A.

The ASCII Table sketch sends the numbers 33 through to 126 out of the USB port. This results in the printable text characters from the ASCII table displayed in the serial monitor window. In addition to the ASCII characters, the number that represents each character is displayed. Each number is shown in four different numbering systems. These are the decimal, hexadecimal, octal and binary number systems. In the serial monitor window, these number systems are abbreviated to dec, hex, oct and bin.

Conclusion: -

Experiment no 6

```
int number;
int squared;

void setup()
{

Serial.begin(9600); // Initialize serial communication

}

void loop()
{

//Serial.println("Enter a number: ");
while (Serial.available() == 0) {

// Wait for user input

}

number = Serial.parseInt(); // Read the entered number
if (number!=0) {
squared = number *number; //Calculate the square of the number

Serial.print("The square of ");

Serial.print(number);

Serial.print(" is ");

Serial.println(squared);
}
delay(2000); // Delay for 2 second
}
```

Output:

The square of 5 is 25

The square of 9 is 81

The square of 6 is 36

