Timer **Course: CSCI 3331**
Lab 5 **Section: 01**

Ahmed Muntasir Hossain

Description of the Lab

The purpose of the lab was to implement three tasks that would be performed autonomously using a timer interrupt. A timer is set for a particular time period and each task executes for that time. After the timer is complete, the interrupt service routine (ISR) is called, and the next task starts executing.

The tasks being executed are the following:

> Task 1: generate random numbers between 0 and 255 inclusive and fill up a character array with those randomly generated numbers.
>
> Task 2: sort the array in ascending order (lowest to highest).
>
> Task 3: print the decimal values of the elements in the array

The timer interrupt is set for 0.1 seconds. Each task is allocated for that time period after which the program switches to the next task by calling the interrupt service routine. If the previous task is incomplete, then the current task would not be able to make any changes and would wait for the timer to complete. After the timer is triggered, the program will switch to the next task and would do the same until it returns to the first incomplete task. It will then resume its execution and try to complete the task before the ISR is called. If the task successfully finishes before the ISR call, then the next task would be able to start executing and would be able to make changes after the ISR call. If the task is unable to finish, then it will continue to loop until it is assigned time again to execute by the timer interrupt.

Additionally, different colors of LED were used to indicate the current task being executed. Red LED indicated that it was Task 1, Blue LED indicated it was Task 2, and Green LED indicated it was Task 3.

Power

The breadboard has two columns marked + (positive) and – (negative). One wire is connected to each column and it can be placed in any hole on that column. This is used to supply power from the Arduino to the breadboard. The positive and negative wires connected to the breadboard are connected to the 5V power pin and GND power pin on the Arduino, respectively.

Output

The output of this device is the RGB LED. It is an electrical component consisting of three LEDs Red, Blue, and Green. The RGB LED has 4 pins “R” (red), “G” (green), “B” (blue), and “–” (negative) which are connected parallelly on the breadboard (different rows). One wire is connected to each pin serially (same row). The R, G, and B wires are connected to any three digital i/o pins (pin number 2 – 13) on the Arduino. The negative wire is connected to any hole (socket) on the negative line on the breadboard. This completes the wiring connection of the RGB LED. The R, G, and B pin markings on the LED may be incorrect and therefore to configure it correctly, the device should be tested. This should be done by setting each individual pin (LED color) to HIGH and observing its color. The color determined from the test can be used to know the correct configuration of the pins.

Ahmed Muntasir Hossain

Another output of this device is the Universal Asynchronous Receiver-Transmitted (UART) port on the Arduino. The port is referred to as COM3 on my device. The name of the port may vary depending on the Arduino. The port is connected to a PC using a USB 2.0 cable (type A/B). Using the Serial Monitor, a message can be sent to the Arduino to be processed. Using the same port, a message can be transmitted to the PC to be displayed on the Serial Monitor.

Input

There are no external inputs for this device. The program runs autonomously and switches between tasks by using the timer interrupt.

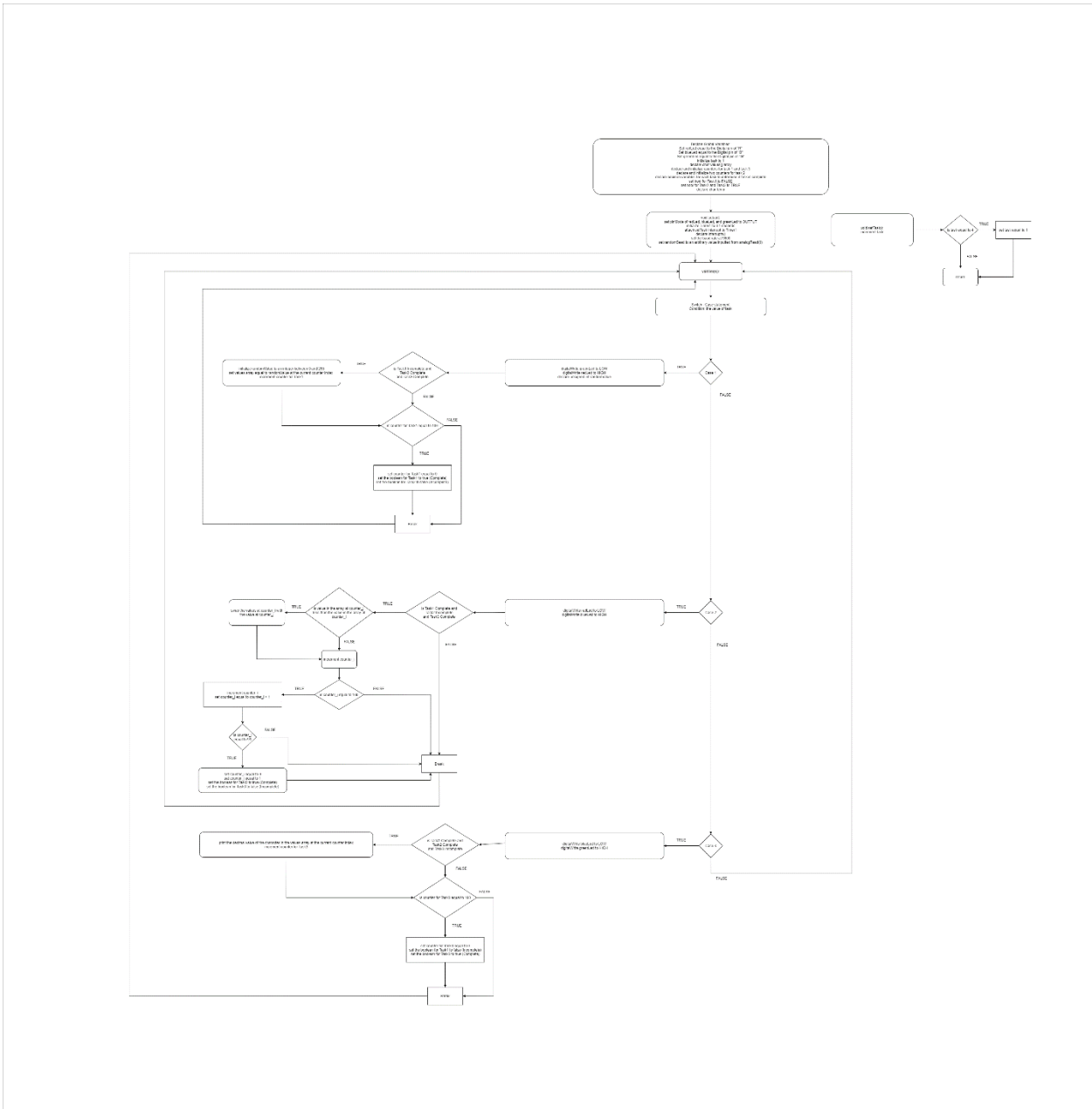
Parts of Arduino

Interrupts are used to share resources and manage multiple applications running at the same time. The timer interrupt on the Arduino is initialized with a particular time value. The timer interrupt calls the interrupt service routine after the assigned time passes. The ISR is responsible for switching the currently executing task to the next task. This allows all the tasks to share resources (CPU time) among themselves and run at the same time.

Timer **Course: CSCI 3331**
Lab 5 **Section: 01**

Ahmed Muntasir Hossain

Flow Chart of the Program Logic



Please note: Quality degraded when inserting image. Flowchart is available as a separate image in the submission folder.

Timer **Course: CSCI 3331**
Lab 5 **Section: 01**

Ahmed Muntasir Hossain

Code

```
#include <TimerOne.h>

const int redLed = 4;
const int blueLed = 2;
const int greenLed = 3;

int task = 1;
char values[100];

int counterTask1 = 0;
int counteriTask2 = 0;
int counterjTask2 = 1;
int counterTask3 = 0;
bool task1Complete = false;
bool task2Complete = true;
bool task3Complete = true;

char temp;

void setup(void)
{
  pinMode(redLed, OUTPUT);
  pinMode(blueLed, OUTPUT);
  pinMode(greenLed, OUTPUT);

  Timer1.initialize(100000);
  Timer1.attachInterrupt(setTask);
  interrupts();

  Serial.begin(9600);
```

Ahmed Muntasir Hossain

```
  randomSeed(analogRead(0));
}

void setTask()
{
  task++;

  if (task == 4)
    task = 1;
}




void loop(void)
{
  switch (task)
  {
    case 1:
      {
        digitalWrite(greenLed, LOW);
        digitalWrite(redLed, HIGH);
        unsigned int randomValue;
        if (!task1Complete && task2Complete && task3Complete)  //initialize values[] with randomly generated
numbers
        {
          randomValue = random(0, 256);
          values[counterTask1] = randomValue;
          counterTask1++;
        }
        if (counterTask1 == 100)
        {
```

Ahmed Muntasir Hossain

```
        counterTask1 = 0;
        task1Complete = true;
        task2Complete = false;
       }

      break;
     }

  case 2:
    {
     digitalWrite(redLed, LOW);
     digitalWrite(blueLed, HIGH);

     if (task1Complete && !task2Complete && task3Complete)
     {
      if (values[counterjTask2] < values[counteriTask2])   //sort array in ascending order
      {
       temp = values[counteriTask2];
       values[counteriTask2] = values[counterjTask2];
       values[counterjTask2] = temp;
      }
      counterjTask2++;

      if (counterjTask2 == 100)
      {
       counterjTask2 = ++counteriTask2 + 1;

       if (counteriTask2 == 99)
       {
        counteriTask2 = 0;
        counterjTask2 = 1;
```

Ahmed Muntasir Hossain

```
            task2Complete = true;
            task3Complete = false;
          }
        }
      }
      break;
    }

  case 3:
    {
      digitalWrite(blueLed, LOW);
      digitalWrite(greenLed, HIGH);
      if (task1Complete && task2Complete && !task3Complete)  //print the values in the array
      {
        Serial.print("Values ");
        Serial.print(counterTask3 + 1);
        Serial.print(": ");
        Serial.println(values[counterTask3], DEC);
        counterTask3++;
      }

      if (counterTask3 == 100)
      {
        counterTask3 = 0;
        task1Complete = false;
        task3Complete = true;
      }
      break;
    }
  }
}
```