

FORMAN CHRISTIAN COLLEGE (A CHARTERED UNIVERSITY)



CSCS 306 – Embedded Systems

Fall 22

Lab - 02

Adeel Akmal – 231465420

Muhammad Sameed Gilani - 231488347

You should attach the hard copy of signed and graded lab / assignment handout as second page of this report.

From third page onwards following headings should be included:

- **Introduction**
 - **Should carry information of all major library /user defined functions used in this lab/assignment.**
- **Your logic / algorithm in simple English. Bullet points are appreciated.**
- **Your code**
- **Circuit diagram (drawn using some online tool. Hand drawn diagrams are not accepted)**
- **Picture of the running hardware.**
- **Screen shots of at least three outputs of your code with appropriate inputs.**
- **References if any**

INTRODUCTION:

TASK1:

Simply had to turn 4 LEDs from right to left. Then blink all of them thrice, after which the 4 LEDs turned on from left to right. This is repeated in an infinite loop until the power is disconnected.

TASK2:

The user enters a string. The program calculates its length, not including new line and empty characters. Then the length is displayed on the serial monitor.

LOGIC:

Task1:

- The necessary pins are set to output
- A loop runs, that one by one turns each LED on from right to left, sequentially with a 200ms delay.
- Now a loop runs, which first simultaneously turns each LED on and after a 200ms delay, it turns each LED simultaneously off.
- Now a loop runs, that one by one turns each LED off from left to right, sequentially with a 200ms delay.
- A loop runs at the end to turn all the LEDs off and reset the LEDs to repeat the pattern

Task2:

- We use the Serial method, `readStringUntil()`, to read the users input until the \$ character is used.
- A while loop runs, which counts the length of the character excluding spaces.
- At the end a loop runs equal to the length calculated previously, and blinks an LED that amount of times. After which the program enters an infinite loop, using `while(1)` to essentially stop the programs from rerunning.
-

CODE:

Task1:

```
int ports[4] = {2,3,4,5};

void setup() {
    // put your setup code here, to run once:
    Serial.begin(9600);
    for(int i=0; i<5; i++){
        pinMode(ports[i], OUTPUT);
    }
}

void loop() {
    // put your main code here, to run repeatedly:
    for(int j=4; j>=0; j--){
        delay(200);
        digitalWrite(ports[j], HIGH);
        delay(200);
    }

    for(int i=0; i<4; i++){
        delay(200);
        for(int j=0; j<5; j++){
```

```

        digitalWrite(ports[j], HIGH);
    }
    delay(200);
    for(int k=0; k<5;k++){
        digitalWrite(ports[k], LOW);
    }
    delay(200);
}

for(int j=0; j<5; j++){
    delay(200);
    digitalWrite(ports[j], HIGH);
    delay(200);
}

for(int k=0; k<5;k++){
    digitalWrite(ports[k], LOW);
}

}

```

Task2:

```

String inp = "";
int j=0;
int count;
bool check = 0;
void setup() {
    // put your setup code here, to run once:
    Serial.begin(9600);
}

```

```

    pinMode(2, OUTPUT);

}

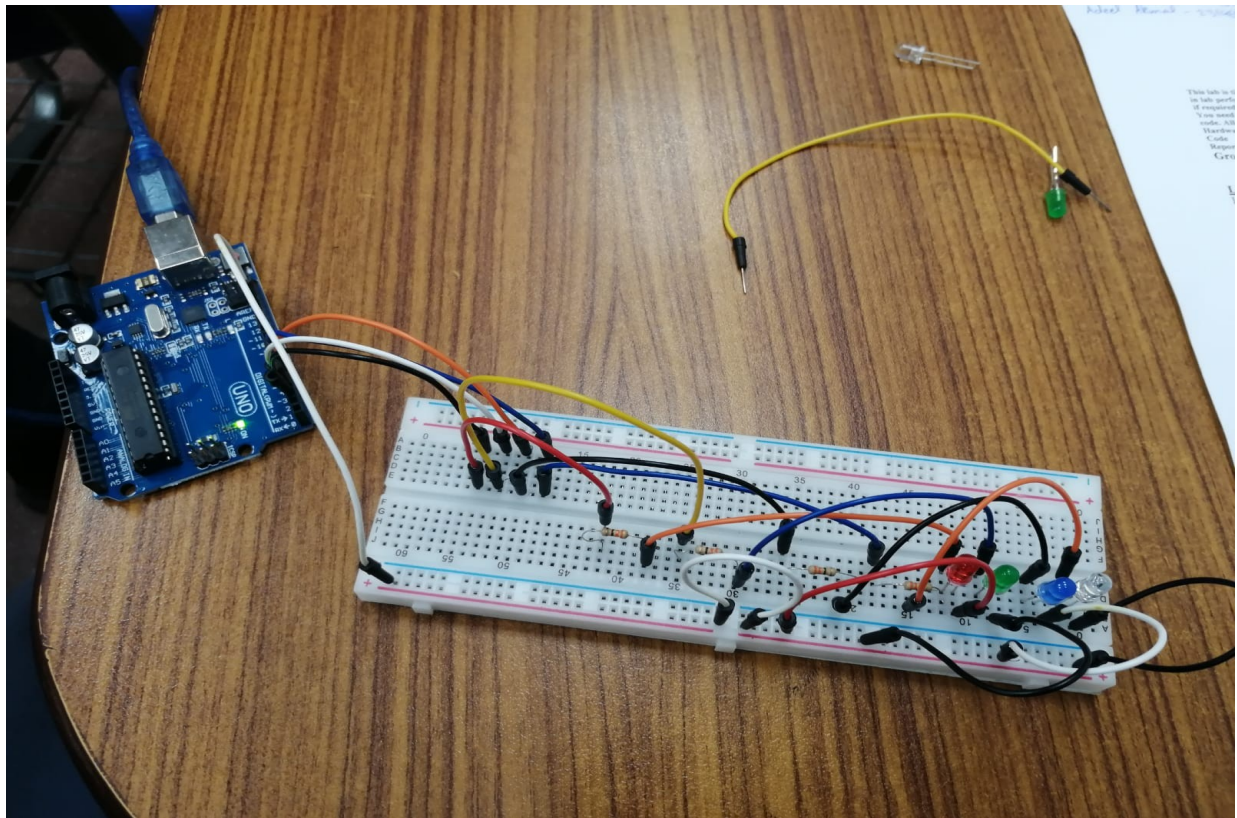
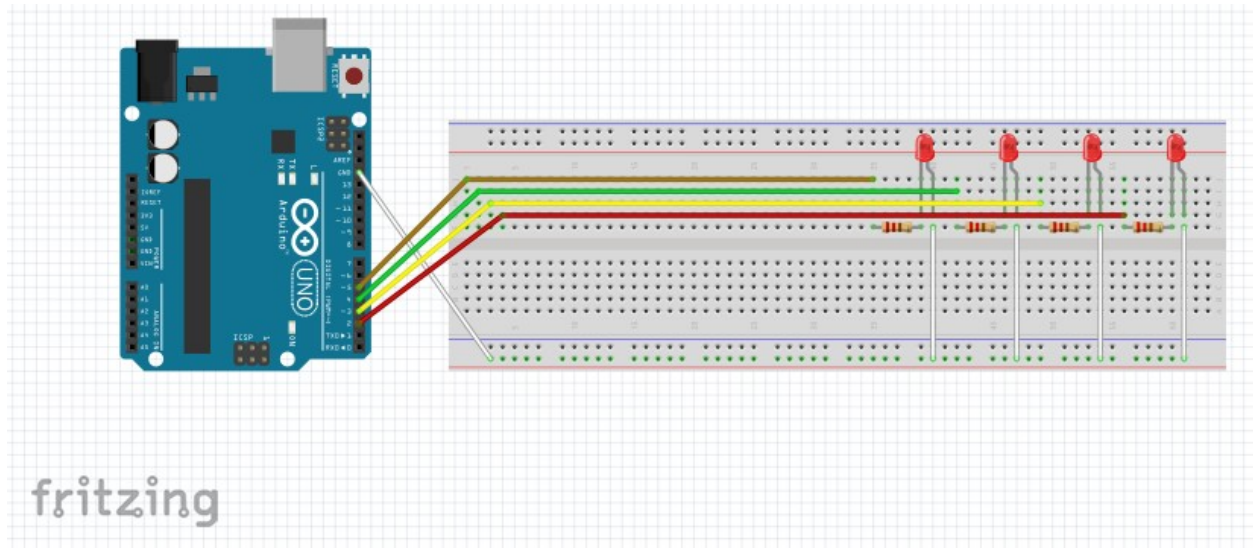
void loop() {
    // put your main code here, to run repeatedly:

    while(Serial.available() == 0){}

    inp = Serial.readStringUntil('$');
    while(j != inp.length()){
        if(inp[j] != ' '){
            count++;
        }
        j++;
    }
    if(inp.length() > 0){
        check = 1;
    }
    Serial.println(count);
    for(int i=0; i<count; i++){
        delay(500);
        digitalWrite(2, HIGH);
        delay(500);
        digitalWrite(2, LOW);
    }
    digitalWrite(2, LOW);
    while(1);}

```

CIRCUIT DIAGRAM:



OUTPUT:

