AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH (AIUB)
Faculty of Engineering
Department of EEE and CoE
Undergraduate Program

Course: MICROPROCESSOR AND EMBEDDED SYSTEMS **Summer 2021-22, MID**

Experiment 4: Taking external inputs in Arduino: Implementation of runway approach lights

Submitted by:

Name	ID
MD SHAMIM SIDDIKY	20-42649-1
WASIUDDIN	15-30843-3
TANJIM HOSSAIN	18-36964-1
MD. SAKIB HOSSAIN RIJON	19-39460-1
SADIA AFRIN ETY	19-39659-1
FERDOUS SUNY	19-40485-1
DIPONKAR SUTRA DHAR	19-41004-2

Section: O

Group: 6

Submitted To

Md Ali Noor
Lecturer
Faculty of Computer Engineering
American International University-Bangladesh

Objective:

To learn how to take external inputs in Arduino. Here the external input will be given by a push switch. And to get familiarized with Debouncing: Implementation and effects in Microcontroller.

Theory and methodology:

Runway approach lights are a set of lights that flash rapidly one after the other to direct an aircraft to the runway. These lights are particularly useful when there is low visibility due to inclement weather, such as persistent rain, fog, etc. This experiment aims to replicate this mechanism.

When the switch is pressed, a set of 6 LEDs will flash in a particular order, and the order will then reverse. An I/O pin will be used to connect the switch, and it will be configured as an input. The microcontroller will read the change in state of the I/O pin (here set as an input) upon depressing the switch and carry out instructions to reverse the flash sequence.

Debouncing: Bouncing is the tendency of any two metal contacts in an electronic device to generate multiple signals as the contacts close or open; debouncing is any kind of hardware device or software that ensures that only a single signal will be acted upon for a single opening or closing of a contact.

Circuit Diagram:

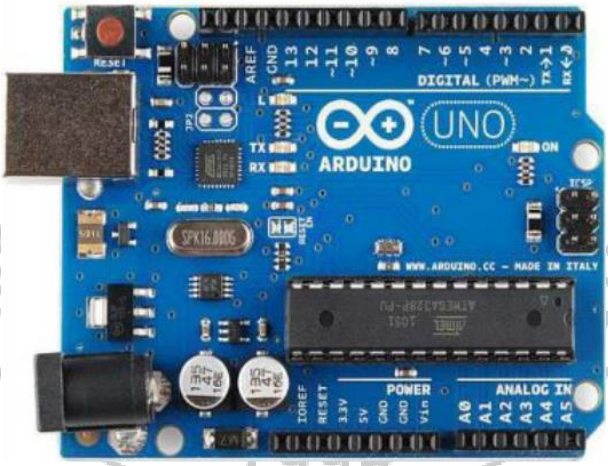

Figure 1: Arduino Uno (R3)

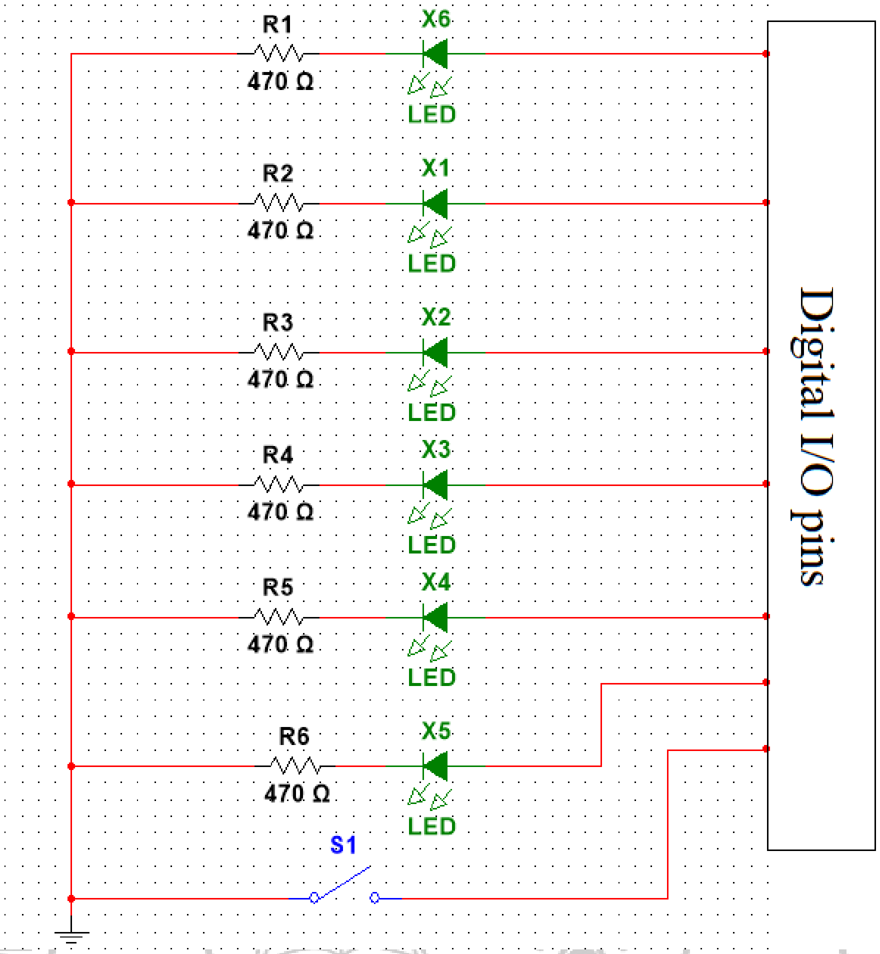

Fig 2: Setup configuration

Hardware Setup:

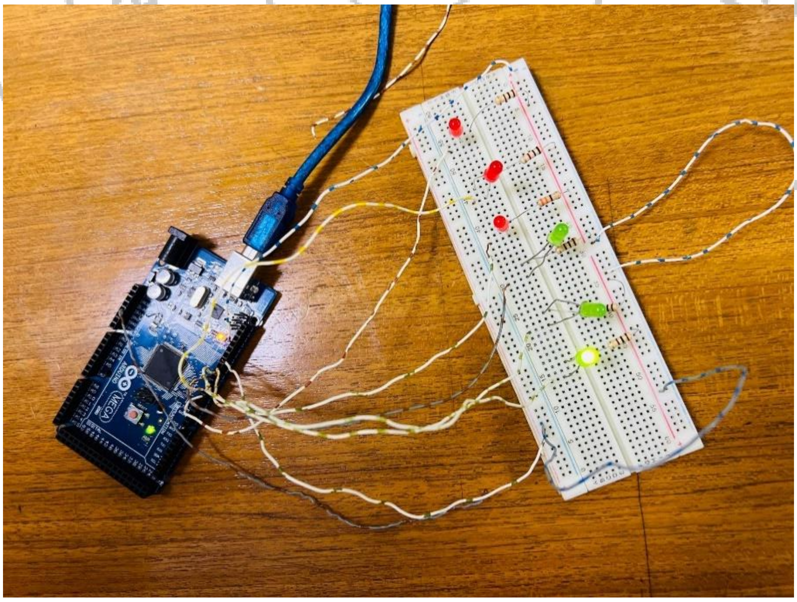

Figure 3: Hardware Set-up for runway approach lights

Runway approach lights:

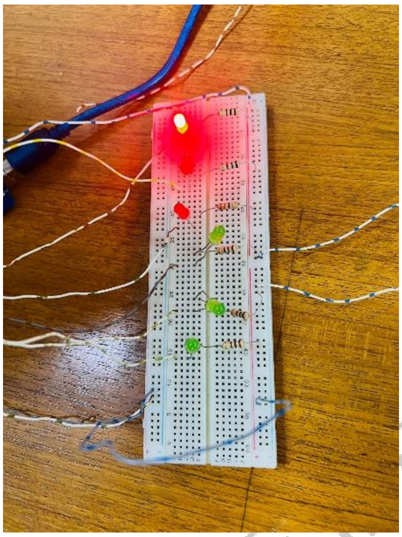

Figure 4.1: Red LED on

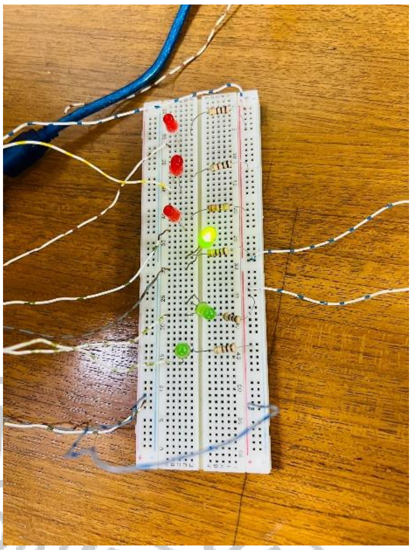

Figure 4.2:Green LED on

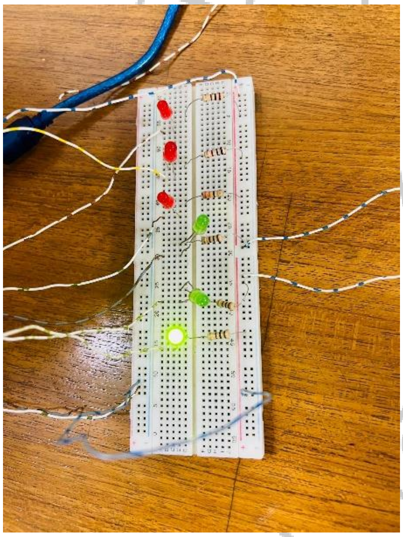

Figure 4.3: Green LED on

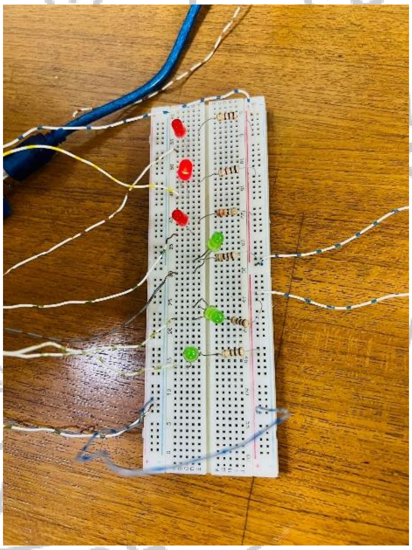

Figure 4.4: Red LED on

Code Analysis:

```
#define PIN_1 5 //define name of pins used
#define PIN_2 6
#define PIN_3 7
#define PIN_4 8
#define PIN_5 9
#define PIN_6 10
#define PIN_7 11

//define the delays for each LED
int LED_blink = 700;
//define variable for switch press
int switch_read; //defining a variable which will read the state of the switch
int LED_sequence=1; //defining which way the LEDs will light up (left to right or right to

int delay_timer (int miliseconds)
{
 int count = 0;
 while(1)
 {
 if(TCNT0 >= 16) // Checking if 1 milisecond has passed
 {
 TCNT0=0;
 count++;
 if (count == miliseconds) //checking if required miliseconds delay has passed
 {
 count=0;
 break; // exits the loop
 }
 }
 }
 return 0;
}
void setup() {
 //define pins connected to LEDs as outputs and the switch as input
 pinMode(PIN_1, OUTPUT);
 pinMode(PIN_2, OUTPUT);
 pinMode(PIN_3, OUTPUT);

 pinMode(PIN_4, OUTPUT);
 pinMode(PIN_5, OUTPUT);
 pinMode(PIN_6, OUTPUT);
 pinMode(PIN_7, INPUT);

 //set up timer
 TCCR0A = 0b00000000;
 TCCR0B = 0b00000101; //setting prescaler for timer clock
 TCNT0=0;
}
void loop() {
 switch_read=digitalRead(PIN_7);
 if (switch_read==LOW){
 LED_sequence=!LED_sequence;
 }
 if (LED_sequence==1){
 //to make green1 LED blink
 digitalWrite(PIN_1, HIGH);
 delay_timer(LED_blink);
 digitalWrite(PIN_1, LOW);

 //to turn red1 LED blink
 digitalWrite(PIN_2, HIGH);
 delay_timer(LED_blink);
 digitalWrite(PIN_2, LOW);

 //green2 blink and so on
 digitalWrite(PIN_3, HIGH);
 delay_timer(LED_blink);
 digitalWrite(PIN_3, LOW);

 digitalWrite(PIN_4,HIGH);
 delay_timer(LED_blink);
 digitalWrite(PIN_4, LOW);

 digitalWrite(PIN_5, HIGH);
 delay_timer(LED_blink);
 digitalWrite(PIN_5, LOW);
 //green2 blink and so on
 digitalWrite(PIN_6, HIGH);
 delay_timer(LED_blink);
 digitalWrite(PIN_6, LOW);

 }
 else {

 digitalWrite(PIN_6, HIGH);
 delay_timer(LED_blink);
 digitalWrite(PIN_6, LOW);


 digitalWrite(PIN_5, HIGH);
 delay_timer(LED_blink);
 digitalWrite(PIN_5, LOW);

 digitalWrite(PIN_4, HIGH);
 delay_timer(LED_blink);
  digitalWrite(PIN_4, LOW);


 digitalWrite(PIN_3, HIGH);
 delay_timer(LED_blink);
  digitalWrite(PIN_3, LOW);

 digitalWrite(PIN_2, HIGH);
 delay_timer(LED_blink);
 digitalWrite(PIN_2, LOW);
 digitalWrite(PIN_1, HIGH);
 delay_timer(LED_blink);
 digitalWrite(PIN_1, LOW);

  }
}
```

Discussion:

In this exercise, we learned how to input externally in Arduino to control LED blink. Initially, the system code implemented by Arduino IDE. Then arranged all the components in the circuit board and then connect the circuit with Arduino Uno. Instead of mechanical switch we use manually created switch using resistor. We use Hardware debouncing to avoid several interrupts from pressing the button just once. In this experiment there may be an only scope of error is typing mistake while typing the codes in the Arduino IDE. Also, we face some error, when we try to implement manual switch instead of using mechanical switch.