TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES

ELECTRONICS ENGINEERING DEPARTMENT

ASSIGNMENT NO. 4

CPET11 – BET-CPET-3A / 7:00 AM – 10:00 AM W

Submitted By:

Arenas, Joseph

Estrada, Cyruz Adrienne

Gutierrez, Geo Kentzer

Pacis, Lian Gil

Recaña, Jordan

Submitted To:

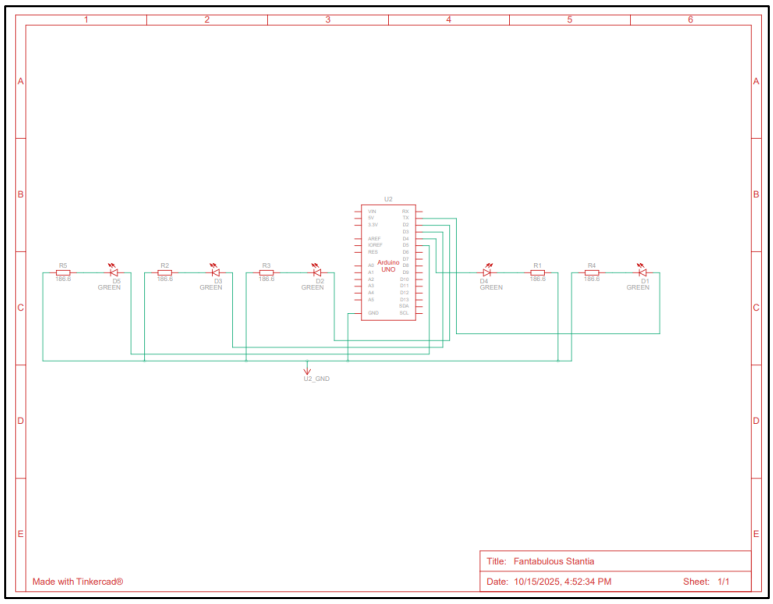
Prof. Ralph Sherwin Corpuz

Faculty

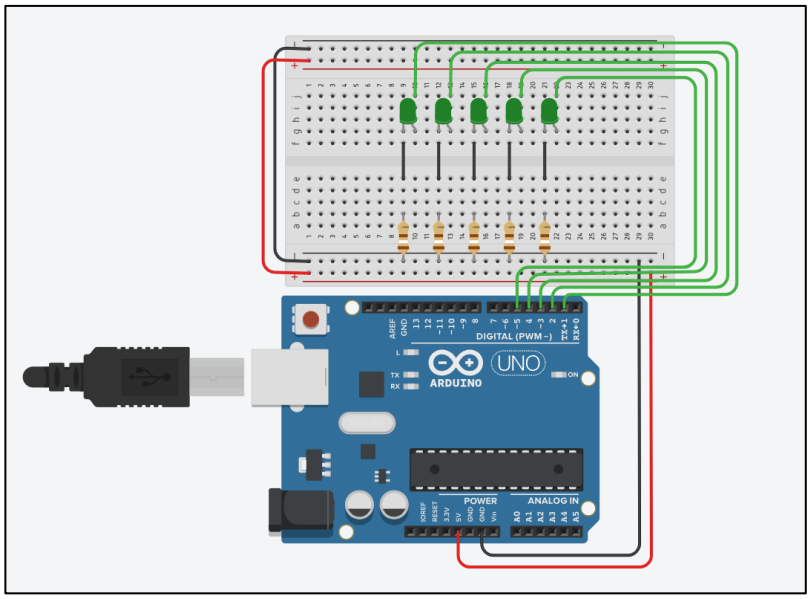
1. Design an Arduino UNO interface using TinkerCAD. For each condition, show the schematic diagram, pictorial diagram, computations, and C++ program. The following are the requirements (100 points)

- Turn on 5 green LEDs for 5 seconds then turn off afterwards.

Schematic Diagram:



Pictorial Diagram:



Computation:

R = V/I

R= Vcc – Vled / 15mA

R = 5 -2.2 / 15 mA

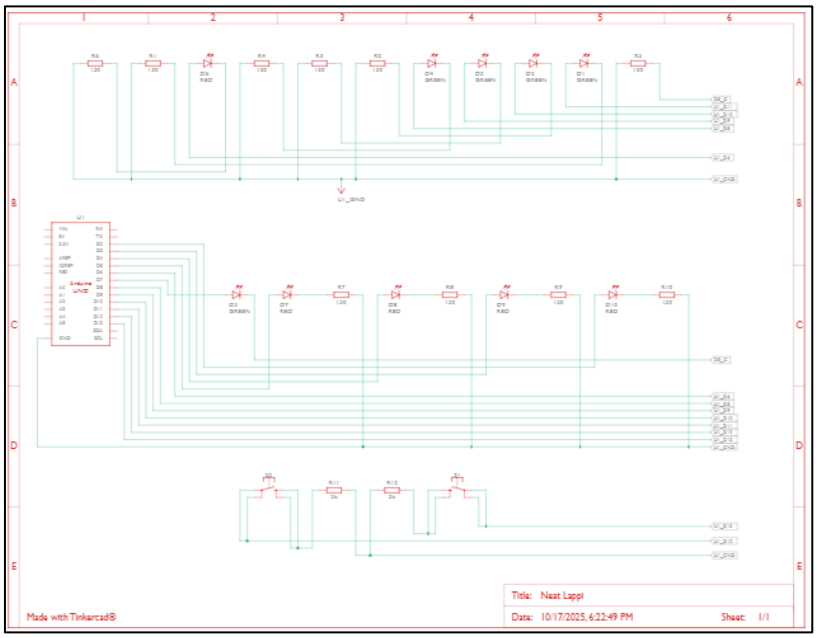
R = 186.6Ω

C++ Code:

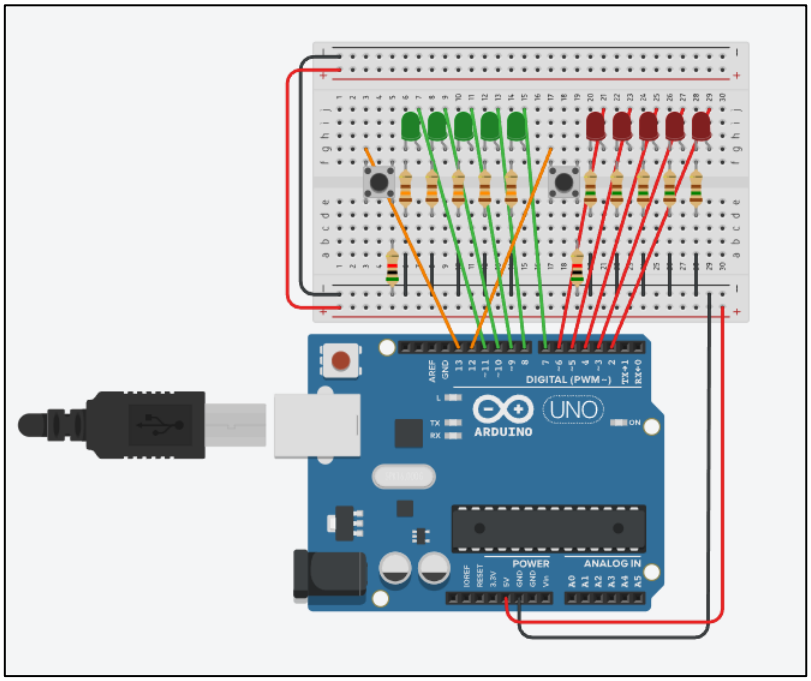
```
const int LED1 = 1;
const int LED2 = 2;
const int LED3 = 3;
const int LED4 = 4;
const int LED5 = 5;

void setup() {
pinMode(LED1, OUTPUT);
pinMode(LED2, OUTPUT);
pinMode(LED3, OUTPUT);
pinMode(LED4, OUTPUT);
pinMode(LED5, OUTPUT);
}
void loop()
{
digitalWrite(LED1, HIGH);
digitalWrite(LED2, HIGH);
digitalWrite(LED3, HIGH);
digitalWrite(LED4, HIGH);
digitalWrite(LED5, HIGH);
delay(5000);

digitalWrite(LED1, LOW);
digitalWrite(LED2, LOW);
digitalWrite(LED3, LOW);
digitalWrite(LED4, LOW);
digitalWrite(LED5, LOW);
delay(5000);
while (true);
}
```

- If pushbutton 1 is activated, 5 green LEDs will turn on for 2 seconds. Meanwhile, if pushbutton 2 is activated, 5 red LEDs will turn on for 5 seconds.

Schematic Diagram:



Pictorial Diagram:



Computations:

- **Limiting Resistor for each Red LED**
  R = V/I
  R = Vcc – Vled / 20mA
  R = 5V – 2V / 20 mA = 3V / 20mA
  R = 150 Ω
- **Limiting Resistor for each Green LED**
  R = V/I
  R = Vcc – Vled / 20mA
  R = 5V – 2.41V / 20 mA = 2.59V / 20mA
  R = 129.5 or 130Ω

- **Pull down Resistor for Push Button**
  (1 mA is chosen as the desired current because it provides a strong enough pull to keep the input pin stable at LOW, while still being small enough to minimize power consumption and prevent unnecessary current flow when the button is pressed.)

  R = V/I
  R = Vcc / 1 mA
  R = 5V / 1 mA
  R = 5k Ω

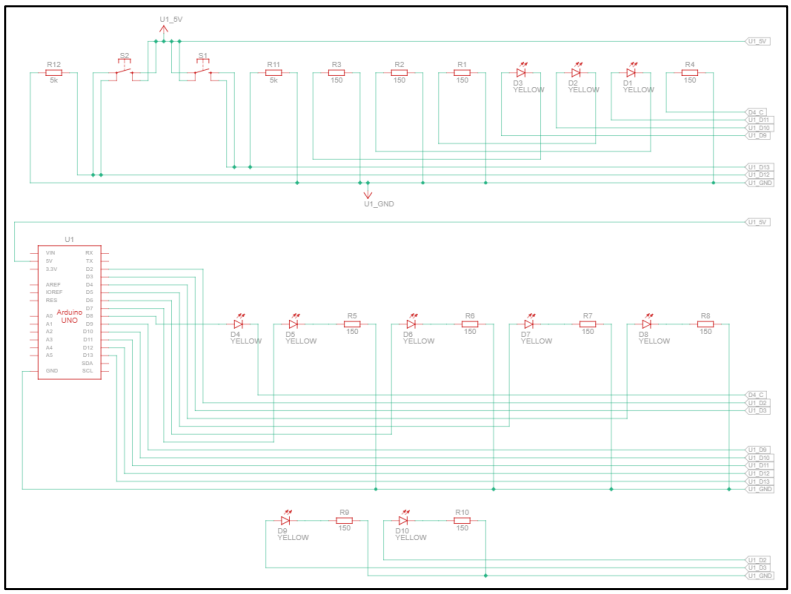
C++ Program:

```
int ledNum = 10;
int ledPin[] = {2, 3, 4, 5, 6, 7, 8, 9, 10, 11};
int button1 = 13;
int button2 = 12;

void setup() {
 for (int x = 0; x < ledNum; x++) {
  pinMode(ledPin[x], OUTPUT);
 }
 pinMode(button1, INPUT_PULLUP);
 pinMode(button2, INPUT_PULLUP);
}

void loop() {
 int buttonState1 = digitalRead(button1);
 int buttonState2 = digitalRead(button2);
```

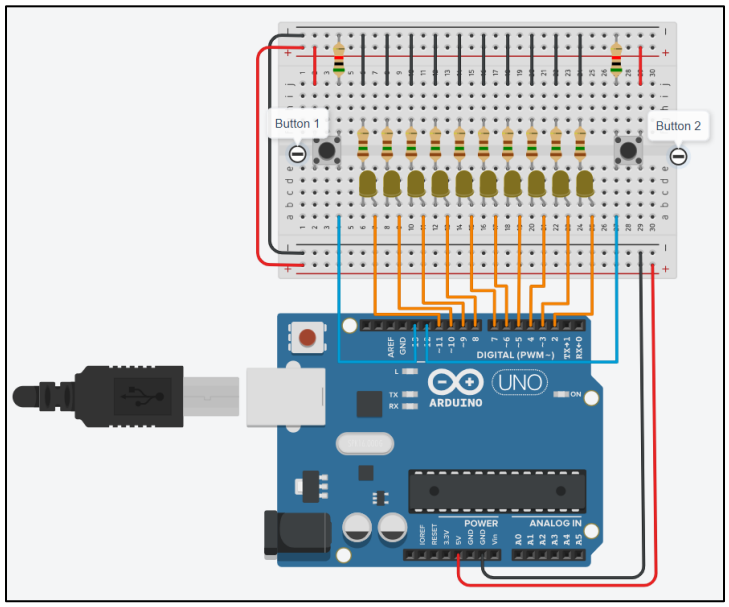
```
  if (buttonState2 == LOW) {
    digitalWrite(ledPin[0], HIGH);
    digitalWrite(ledPin[1], HIGH);
    digitalWrite(ledPin[2], HIGH);
    digitalWrite(ledPin[3], HIGH);
    digitalWrite(ledPin[4], HIGH);
    delay(5000);
    digitalWrite(ledPin[0], LOW);
    digitalWrite(ledPin[1], LOW);
    digitalWrite(ledPin[2], LOW);
    digitalWrite(ledPin[3], LOW);
    digitalWrite(ledPin[4], LOW);
  }

    if (buttonState1 == LOW) {
    digitalWrite(ledPin[5], HIGH);
    digitalWrite(ledPin[6], HIGH);
    digitalWrite(ledPin[7], HIGH);
    digitalWrite(ledPin[8], HIGH);
    digitalWrite(ledPin[9], HIGH);
    delay(2000);
    digitalWrite(ledPin[5], LOW);
    digitalWrite(ledPin[6], LOW);
    digitalWrite(ledPin[7], LOW);
    digitalWrite(ledPin[8], LOW);
    digitalWrite(ledPin[9], LOW);
  }
}
```

- If pushbutton 1 is activated, 10 yellow LEDs will light in sequence (i.e., running lights from left to right). Meanwhile, if pushbutton 2 is activated these LEDs light up in reverse mode. There is a pull-down resistor used to minimize voltage fluctuation.

Schematic Diagram:



Pictorial Diagram:



Computations:

- **Limiting Resistor for each LED**

  R = V/I
  R = Vcc – Vled / 20mA
  R = 5V – 2V / 20 mA = 3V / 20mA
  R = 150 Ω

- **Pull down Resistor for Push Button**

  (1 mA is chosen as the desired current because it provides a strong enough pull to keep the input pin stable at LOW, while still being small enough to minimize power consumption and prevent unnecessary current flow when the button is pressed.)

  R = V/I
  R = Vcc / 1 mA
  R = 5V / 1 mA
  R = 5k Ω

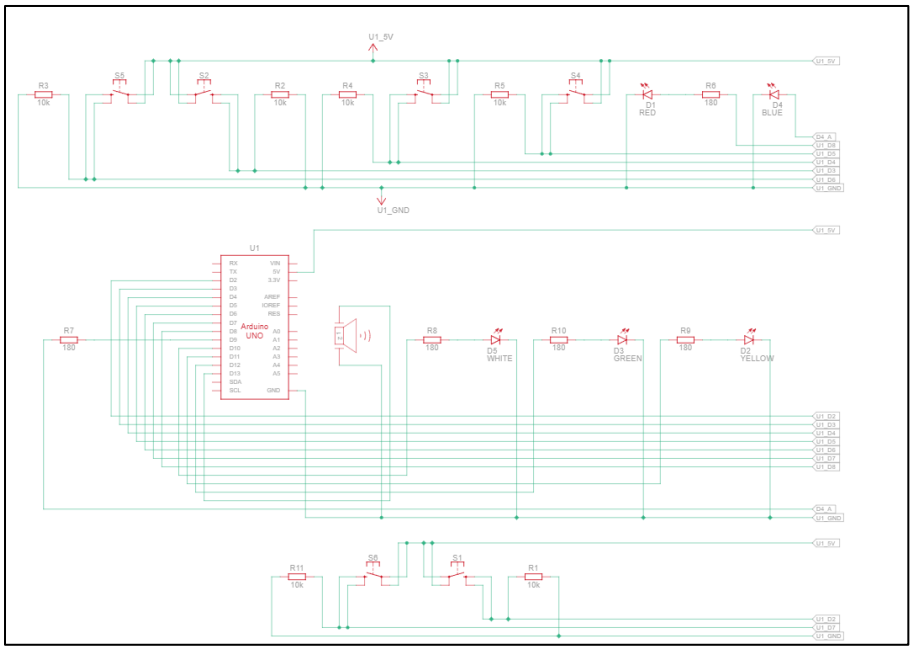
C++ Program:

```
const int numLEDs = 10;
int ledPins[numLEDs] = {11, 10, 9, 8, 7, 6, 5, 4, 3, 2};
int button1 = 13;
int button2 = 12;

void setup() {
 for (int i = 0; i < numLEDs; i++) {
  pinMode(ledPins[i], OUTPUT);
 }

 pinMode(button1, INPUT);
 pinMode(button2, INPUT);
}

void loop() {
// Left-to-right sequence
 if (digitalRead(button1) == HIGH) {
  for (int i = 0; i < numLEDs; i++) {
   digitalWrite(ledPins[i], HIGH);
   delay(200);
   digitalWrite(ledPins[i], LOW);
  }
 }
```

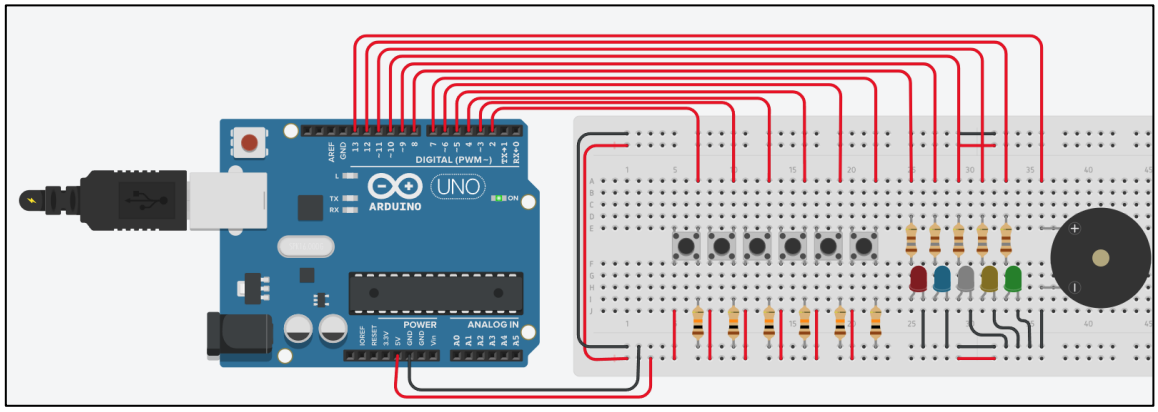
```
  // Right-to-left sequence
 if (digitalRead(button2) == HIGH) {
  for (int i = numLEDs - 1; i >= 0; i--) {
   digitalWrite(ledPins[i], HIGH);
   delay(200);
   digitalWrite(ledPins[i], LOW);
  }
 }
}
```

2. In reference to your proposed microcontroller application study, design the schematic diagram, pictorial diagram, computations, and C++ program. Simulate and take a video recording on the functional operation of the system (100 points)

Schematic Diagram:



Pictorial Diagram:



Computations:

This simple simulation system features a set of buttons, different color LEDs, resistors, 1 buzzer and an Arduino Uno Board. The 5 buttons represent the 5 sensors that will be used in the system, namely, Temperature (DS18B20), Humidity (DHT22), Smoke (MQ-2), Gas (MQ-9), Oxygen (ZE03-O2). These buttons will be represented by a set of LEDs and 1 buzzer. If the temperature button is pressed, a red colored LED will light, and the buzzer will go high (there will be different frequency per button (sensors) just for categorization). Lastly, there will be a main button wherein if pressed, will light up all the LEDs and turn on the buzzer (which signifies that all sensors sensed a change in the environment). Below is the representation of the simulation logic:

Buttons	Simulated Sensor	LED Color	Buzzer Tone
Button 1	Temperature (DS18B20)	Red	1000hz
Button 2	Humidity (DHT22)	Blue	800hz
Button 3	Smoke (MQ-2)	White	1200hz
Button 4	Gas (MQ-9)	Yellow	900hz
Button 5	Oxygen (ZE03-O2)	Green	2000hz
Button 6	All sensor	All LEDs	Default

Below is the pin connection of the various components to the Arduino Uno R3:

Component	Arduino Pin
Temperature Button	2
Humidity Button	3
Smoke Button	4
Gas Button	5
Oxygen Button	6
Main Button	7
Temperature LED (Red)	8
Humidity LED (Blue)	9
Smoke LED (White)	10
Gas LED (Yellow)	11
Oxygen LED (Green)	12

The following information is about the computations for the necessary components (resistor). Using the formula R=Vsupply-Vled/I (required current of LED). We have calculated the necessary resistor value per led:

LED Color	V_{LED}	I	Computed R	Nearest Value
Red	2.0V	0.02A	(5 V – 2 V)/0.02 A = 150 Ω	180 Ω or 220 Ω
Yellow	2.1V	0.02A	(5 – 2.1)/0.02 = 145 Ω	180 Ω or 220 Ω
Green	2.2V	0.02A	(5 – 2.2)/0.02 = 140 Ω	180 Ω or 220 Ω
Blue	3.2V	0.02A	(5 – 3.2)/0.02 = 90 Ω	180 Ω or 220 Ω

White	3.0V	0.02A	(5 – 3.0)/0.02 = 100 Ω	180 Ω or 220 Ω

C++ Code:

```
// Define pins
const int buttonPins[5] = {2, 3, 4, 5, 6};   // Sensor buttons
const int ledPins[5]    = {8, 9, 10, 11, 12}; // Sensor LEDs
const int mainButton = 7;                 // Main alarm button
const int buzzer = 13;                    // Single buzzer

// Frequencies for each sensor
const int frequencies[5] = {1000, 800, 1200, 900, 700};

void setup() {
  for (int i = 0; i < 5; i++) {
    pinMode(buttonPins[i], INPUT);
    pinMode(ledPins[i], OUTPUT);
  }
  pinMode(mainButton, INPUT);
  pinMode(buzzer, OUTPUT);
}

void loop() {
  // Main alarm button pressed?
  if (digitalRead(mainButton) == HIGH) {
    activateAllAlarms();
  } else {
    bool anyActive = false;
```

```
  // Check each "sensor"
  for (int i = 0; i < 5; i++) {
    if (digitalRead(buttonPins[i]) == HIGH) {
      digitalWrite(ledPins[i], HIGH);
      tone(buzzer, frequencies[i]); // Each has unique tone
      anyActive = true;
      delay(100); // Small delay for clarity
    } else {
      digitalWrite(ledPins[i], LOW);
    }
  }

  // Turn off buzzer if no buttons are pressed
  if (!anyActive) {
    noTone(buzzer);
  }
 }
}

// Main alarm function
void activateAllAlarms() {
 // Turn all LEDs ON
 for (int i = 0; i < 5; i++) {
  digitalWrite(ledPins[i], HIGH);
 }

 // Strong general alarm sound
 tone(buzzer, 2000);
 delay(1000);
```

```
 // Turn everything OFF
 for (int i = 0; i < 5; i++) {
   digitalWrite(ledPins[i], LOW);
 }
 noTone(buzzer);
}
```

3. Specify the percentage contribution of each group member (100% total). Each member should sign to confirm the breakdown of contribution.

Contributions of each group member:

ARENAS, JOSEPH 20%

ESTRADA, CYRUS 20%

GUTIERREZ, GEO KENTZER 20%

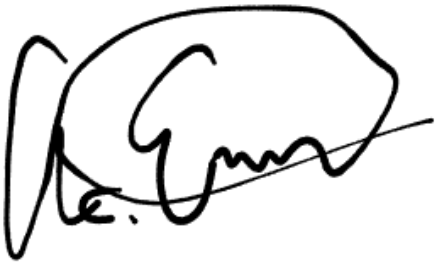
PACIS, LIAN GIL 20%

RECAÑA, JORDAN 20%

TOTAL 100%

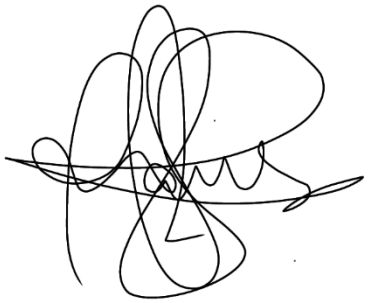
Signature of each group member:

ARENAS, JOSEPH



ESTRADA, CYRUZ

GUTIERREZ, GEO KENTZER



PACIS, LIAN GIL

RECAÑA, JORDAN