

Ghulam Ishaq Khan Institute of Engineering Sciences and Technology

IF-102 Project Report

AI – Faeda

0-9 Seven segment Display Counter with IR sensor

Mian Abdul Moiez
2023314
FEE - D

Muhammad Rayan Ur Rehman
Khan
2023281
FEE – D

Essa Faisal
2023182
FEE - D

Mohammad Abdullah
2023324
FEE - D

This project was made by Faculty of Electrical Engineering students and is in accordance to the list of Projects given to us. This project counts from 0 to 9 with the help of IR sensor. Which works on the principal of detecting the InfraRed emitted by the object and converting it to electrical signals. The Ir sensor consists of Photodiode or Phototransistor.

I. This project is comprises of following components :

II. Arduino Uno

III. Seven segment Display

IV. Lightening Emitting Diode / LED (Blue color)

V. Push Buttons

VI. Jumper wires

We proposed to add a Led display with some advanced features to use it instead of 7-segment display but we were bounded to make it as the Title suggests. There is a Reset button attached to the circuit which could reset the current counting to zero.

With the help of IR-sensor we are able to count without any push buttons by the utilization of infrared sensor that detect object passing through field and then result in increment of a number on & segment display. There is a Reset button attached to the circuit which could reset the current counting to zero.

Applications:

This counter can be used in many fields in practical life that may include the following.

- 1) Manufacturing and Assembly Lines
- 2) Track production quantity
- 3) Monitoring System in Vehicle flow on Highway or roads

The 0-9 counter can be further improved if we add further Led Display module or more & segment display.

=

• Code Breakdown:

- The functioning of the cop code programmed on Arduino IDE is given below:
- The code sets up several constant variables (a, b, c, etc.) to represent pins on a microcontroller, making it easier to refer to them later in the code. These pins are used to control a display segment and an IR sensor.
- It initializes some integer variables and pins for input and output. For instance, **upping** and **downPin** are set as input pins to detect button presses, while **irSensorPin** is defined for an IR sensor.
- In the **setup()** function, it configures the pins for output and input, and sets the initial state for the display.
- The **loop()** function continuously checks for sensor inputs and button presses. When motion is detected by the IR sensor, it increments the display count and updates the display accordingly.
- Two separate functions, **checkupxButtonPress()** and **checkdownxButtonPress()**, handle the logic for

incrementing the display count when the up or down buttons are pressed, respectively.

- There's also a **segmentdigit()** function that controls which segments of the display are activated to show different digits. It turns on or off specific segments based on the digit passed to it.
- Lastly, the **turnOff()** function is used to turn off all segments of the display when not needed.
- `const int a = 11;`
- `const int b = 10;`
- `const int c = 9;`
- `const int d = 8;`
- `const int e = 7;`
- `const int f = 6;`
- `const int g = 5;`
- `const int l = 4;`
-
- `const int irSensorPin = 12; // IR sensor pin`
-
- `int k = 0;`
- `const int upPin = 3;`
- `const int downPin = 2;`
- `int x = 0;`
- `int upx = 0;`
- `int upy = 0;`
- `int downx = 0;`
- `int downy = 0;`
-
- `void setup() {`
- `pinMode(a, OUTPUT);`
- `pinMode(b, OUTPUT);`
- `pinMode(c, OUTPUT);`
- `pinMode(d, OUTPUT);`
- `pinMode(e, OUTPUT);`
- `pinMode(f, OUTPUT);`
- `pinMode(g, OUTPUT);`
- `pinMode(l, OUTPUT);`
- `pinMode(upPin, INPUT_PULLUP);`
- `pinMode(downPin, INPUT_PULLUP);`
-
- `pinMode(irSensorPin, INPUT);`
- `segmentdigit(x);`

- `}`
-
- `void loop() {`
- `sensor_check();`
- `upx = digitalRead(upPin);`
- `downx = digitalRead(downPin);`
- `checkupxButtonPress();`
- `checkdownxButtonPress();`
- `if (k) {`
- `k = 0;`
- `turnOff();`
- `segmentdigit(x);`
- `}`
- `}`
-
- `void sensor_check() {`
- `int irSensorState = digitalRead(irSensorPin);`
-
- `// If motion is detected, increment count and display it`
- `if (irSensorState == LOW) {`
- `x++;`
- `if(x > 9) {`
- `x = 0;`
- `}`
- `digitalWrite(l, HIGH);`
- `delay(500);`
- `segmentdigit(x);`
- `}`
- `}`
-
- `void checkupxButtonPress() {`
- `if (upx != upy) {`
- `if (upx == LOW) {`
- `k = 1;`
- `x++;`
- `digitalWrite(l, HIGH);`
- `if (x > 9) x = 0;`
- `}`
- `}`
- `delay(100); // Adjust delay for debouncing`
- `}`

```

• upy = upx;
• }
•
• void checkdownxButtonPress() {
•   if (downx != downy) {
•     if (downx == LOW) {
•       k = 1;
•       x = 0; // Reset x when down button is pressed
•       digitalWrite(l, HIGH);
•     }
•     delay(100); // Adjust delay for debouncing
•   }
•   downy = downx;
• }
•
• void segmentdigit(int digit) {
•   turnOff();
•
•   switch (digit) {
•     case 0:
•       digitalWrite(a, HIGH);
•       digitalWrite(b, HIGH);
•       digitalWrite(c, HIGH);
•       digitalWrite(d, HIGH);
•       digitalWrite(e, HIGH);
•       digitalWrite(f, HIGH);
•       break;
•     case 1:
•       digitalWrite(b, HIGH);
•       digitalWrite(c, HIGH);
•       break;
•     case 2:
•       digitalWrite(a, HIGH);
•       digitalWrite(b, HIGH);
•       digitalWrite(d, HIGH);
•       digitalWrite(e, HIGH);
•       digitalWrite(g, HIGH);
•       break;
•     case 3:
•       digitalWrite(a, HIGH);
•       digitalWrite(b, HIGH);
•       digitalWrite(c, HIGH);
•       digitalWrite(d, HIGH);
•       digitalWrite(e, HIGH);
•       digitalWrite(f, HIGH);
•       digitalWrite(g, HIGH);
•       break;
•     case 4:
•       digitalWrite(b, HIGH);
•       digitalWrite(c, HIGH);
•       digitalWrite(f, HIGH);
•       digitalWrite(g, HIGH);
•       break;
•     case 5:
•       digitalWrite(a, HIGH);
•       digitalWrite(c, HIGH);
•       digitalWrite(d, HIGH);
•       digitalWrite(f, HIGH);
•       digitalWrite(g, HIGH);
•       break;
•     case 6:
•       digitalWrite(a, HIGH);
•       digitalWrite(c, HIGH);
•       digitalWrite(d, HIGH);
•       digitalWrite(e, HIGH);
•       digitalWrite(f, HIGH);
•       digitalWrite(g, HIGH);
•       break;
•     case 7:
•       digitalWrite(a, HIGH);
•       digitalWrite(b, HIGH);
•       digitalWrite(c, HIGH);
•       break;
•     case 8:
•       digitalWrite(a, HIGH);
•       digitalWrite(b, HIGH);
•       digitalWrite(c, HIGH);
•       digitalWrite(d, HIGH);
•       digitalWrite(e, HIGH);
•       digitalWrite(f, HIGH);
•       digitalWrite(g, HIGH);
•       break;

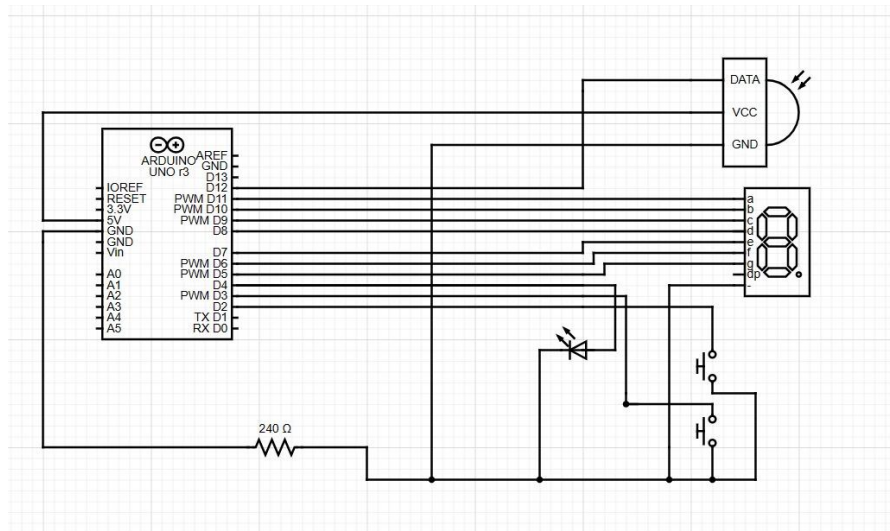
```

- case 9:
- digitalWrite(a, HIGH);
- digitalWrite(b, HIGH);
- digitalWrite(c, HIGH);
- digitalWrite(d, HIGH);
- digitalWrite(f, HIGH);
- digitalWrite(g, HIGH);
- break;
- default:
- break;
- }
- }
- }
- void turnOff()
- {

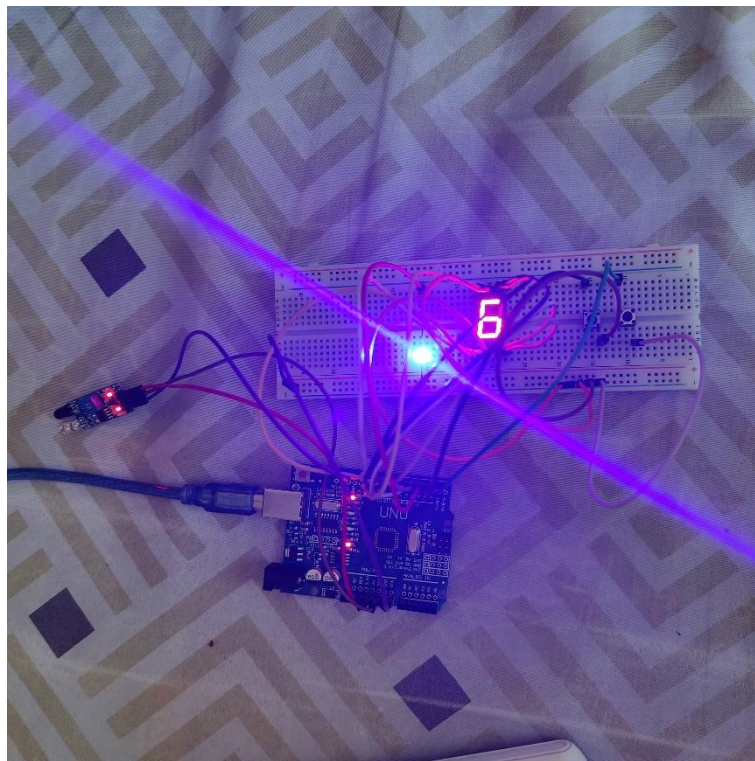
- digitalWrite(a, LOW);
- digitalWrite(b, LOW);
- digitalWrite(c, LOW);
- digitalWrite(d, LOW);
- digitalWrite(e, LOW);
- digitalWrite(f, LOW);
- digitalWrite(g, LOW);
- digitalWrite(l, LOW);
- }

[1] **Power Management :**

The whole system work when the power cable from arduino is connected to the usb connector which could either work from the Power Bank or the Laptop or Pc itself . Furturemore a battery can also be used.



Circuit Diagram



Prototype working on Breadboard which was later transformed into Veroboard