

**International Center for Free and Open Source  
Software**



**SIMULATING OR GATE**

**Azwa Harshad**

**Internship**

**Open IoT**

**FEB 2023**

# Contents

<b>List of Figures</b>	<b>ii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 General Background . . . . .	1
1.2 Working Principle . . . . .	1
1.3 Components Needed . . . . .	2
<b>2 Circuit Diagram</b>	<b>3</b>
2.1 Circuit Connections . . . . .	3
2.2 Block Diagram . . . . .	4
<b>3 Implementation</b>	<b>5</b>
3.1 Procedure . . . . .	5
<b>4 Code</b>	<b>6</b>
4.1 Code for implementing OR gate . . . . .	6
4.2 Serial Monitor Output . . . . .	7
<b>5 Result</b>	<b>8</b>
5.1 Result . . . . .	8

# List of Figures

1.1	Truth Table . . . . .	2
2.1	Circuit Diagram . . . . .	3
2.2	Block Diagram . . . . .	4
4.1	Code . . . . .	6
4.2	Serial Monitor . . . . .	7
5.1	Input1=HIGH, Input2=HIGH . . . . .	8
5.2	Input1=HIGH, Input2=Low . . . . .	9
5.3	Input1=LOW, Input2=HIGH . . . . .	9
5.4	Input1=LOW, Input2=LOW . . . . .	9

# **Chapter 1**

## **Introduction**

### **1.1 General Background**

This project is a simple way of using the Arduino to simulate the behaviour of OR gates. The project does not actually carry out the function of the OR gate, just turns a light on or off based on one or two inputs. Effectively showing the truth table for a given OR gate.

### **1.2 Working Principle**

A logical OR operation has a high output (1) if one or both the inputs to the gate are high (1). If neither input is high, a low output (0) results. Just like an AND gate, an OR gate may have any number of input probes but only one output probe.

The function of a logical OR gate effectively finds the maximum between two binary digits, just as the complementary AND function finds the minimum. It's a fundamental building block in digital logic circuits and is used in various applications such as arithmetic operations, control circuits, and decision-making circuits. The OR gate has a propagation delay, which is the time taken for the output to respond to a change in the input. This delay is usually very small in modern electronic circuits.

OR



Inputs		Output
A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

Figure 1.1: Truth Table

### 1.3 Components Needed

1. 2 x LEDs Green
2. 1 x LED Red
3. 3 x 330 Ohm Resistors
4. 2 x 10 KOhm Resistors
5. 2 x Pushbuttons
6. Jumper Wires

The yellow LEDs will be used to represent the input values, the red LED the output.

# Chapter 2

## Circuit Diagram

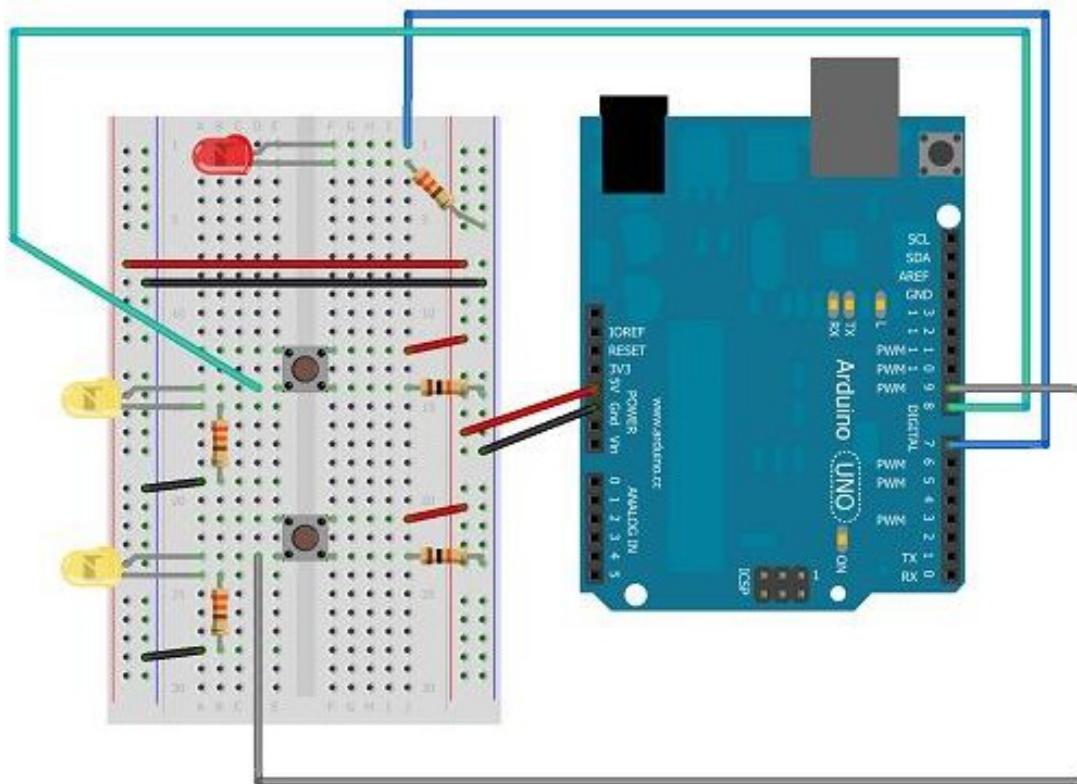


Figure 2.1: Circuit Diagram

### 2.1 Circuit Connections

The pushbuttons are connected with 10K resistors to GND. An LED is placed on the same row as the connections to the input pins (8 and 9). This gives us a push-to-make configuration for the pushbuttons. It also means that we don't have to write any code

to make the two input indicator LEDs light up. You can test the button connections early on. The Yellow LEDs in this diagram should light when the buttons are pressed and be off when not.

## 2.2 Block Diagram

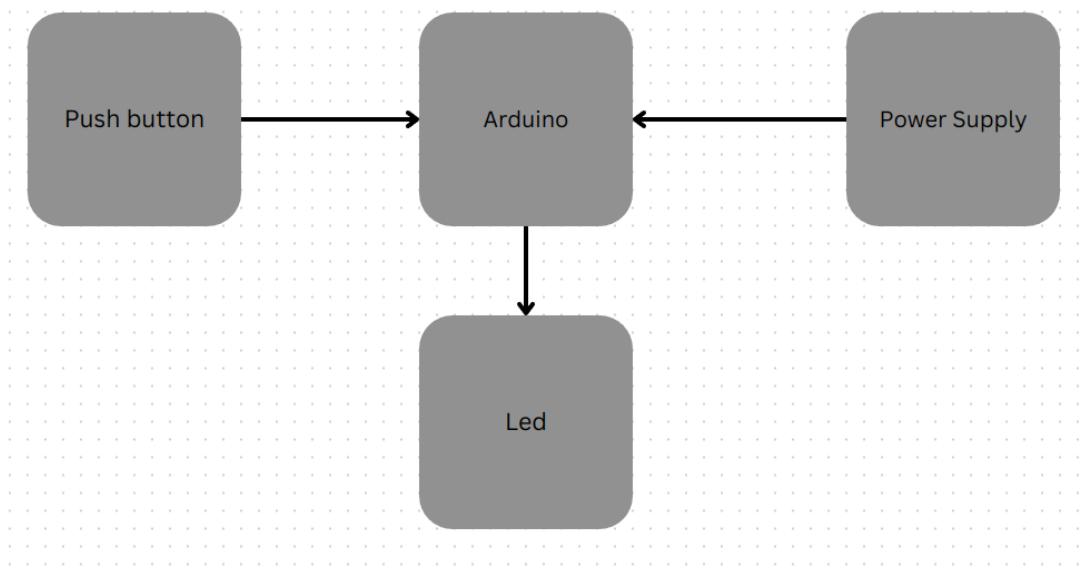


Figure 2.2: Block Diagram

# **Chapter 3**

## **Implementation**

### **3.1 Procedure**

1. Setup Hardware: Gather all necessary components. Connect them according to the circuit diagram.
2. Install Arduino IDE: Download and install Arduino IDE from the official website.
3. Open Arduino IDE: Launch the Arduino IDE software.
4. Write Code: Compose your program using Arduino programming language (based on C/C++).  
Write setup and loop functions.
5. Verify Code: Click on the Verify button (checkmark icon) to check for any errors in the code.
6. Upload Code: Connect Arduino board to the computer via USB. Select the correct board and port from Tools menu.  
Click on the Upload button (right arrow icon) to upload the code to the Arduino board.
7. Test: Make sure the hardware is powered on.

# Chapter 4

## Code

### 4.1 Code for implementing OR gate

It is done through ArduinoIDE

```
1 const int pushbutton1 = 8;
2 const int pushbutton2 = 9;
3 int led = 7;
4 void setup() {
5   pinMode(pushbutton1,INPUT);
6   pinMode(pushbutton2,INPUT);
7   pinMode(led,OUTPUT);
8   Serial.begin(9600);
9 }
10 void loop() {
11   int value1 = digitalRead(pushbutton1);
12   int value2 = digitalRead(pushbutton2);
13   int value = value1 || value2;
14   digitalWrite(led,value);
15   if(value == HIGH) {
16     digitalWrite(led,value);
17   }
18   else {
19     digitalWrite(led,LOW);
20   }
21   Serial.print("Button 1 state:");
22   Serial.print(",");
23   Serial.print("value1");
24   Serial.print("Button 2 state:");
25   Serial.print(",");
26   Serial.print("value2");
27   Serial.println("Output state:");
28   Serial.println(value);
29   delay(2000);
30 }
31 }
```

Figure 4.1: Code

## 4.2 Serial Monitor Output

/dev/ttyUSB0

Figure 4.2: Serial Monitor

# Chapter 5

## Result

### 5.1 Result

The output is obtained and ORgate integrated with arduino.

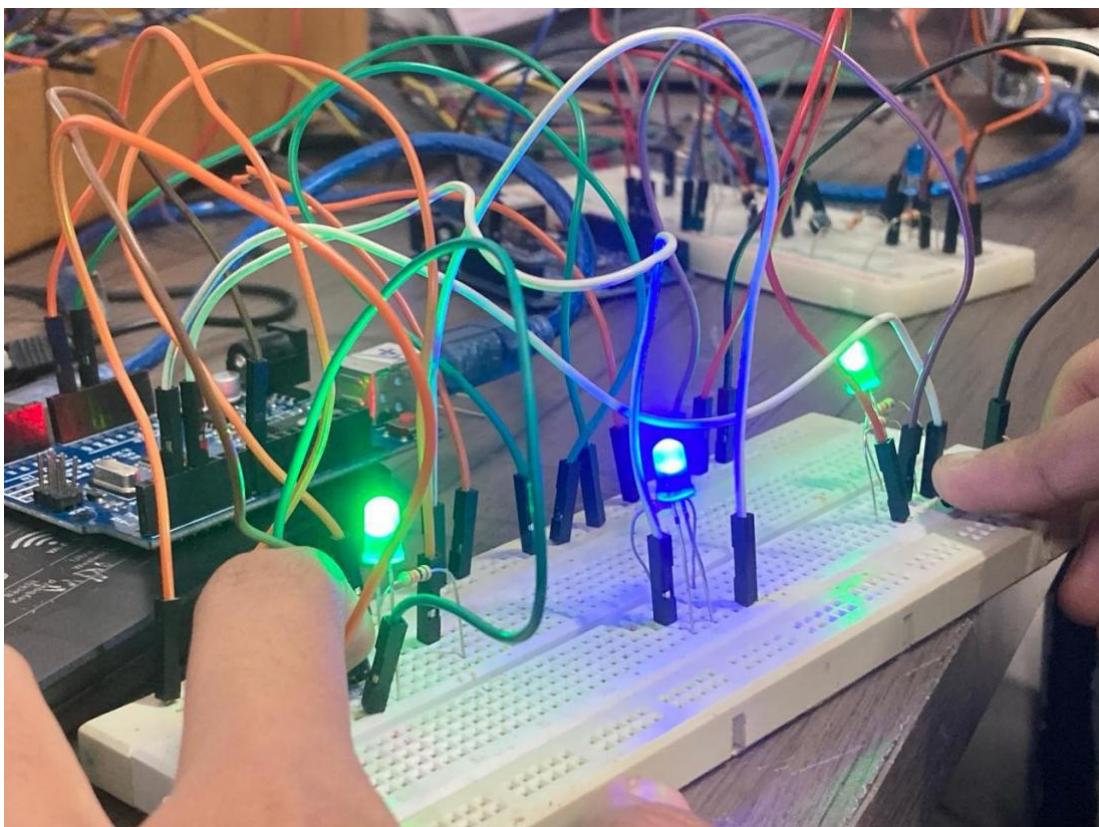


Figure 5.1: Input1=HIGH, Input2=HIGH

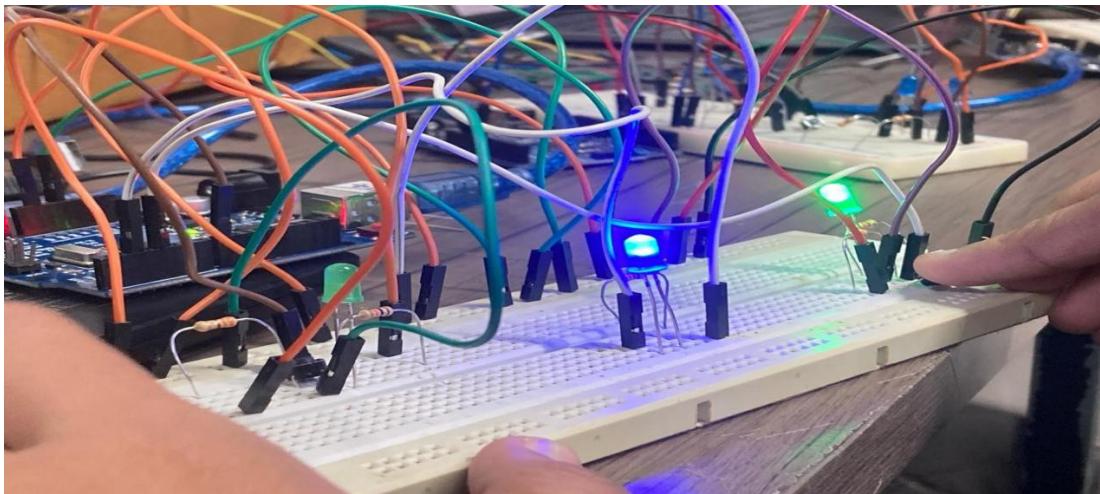


Figure 5.2: Input1=HIGH, Input2=Low

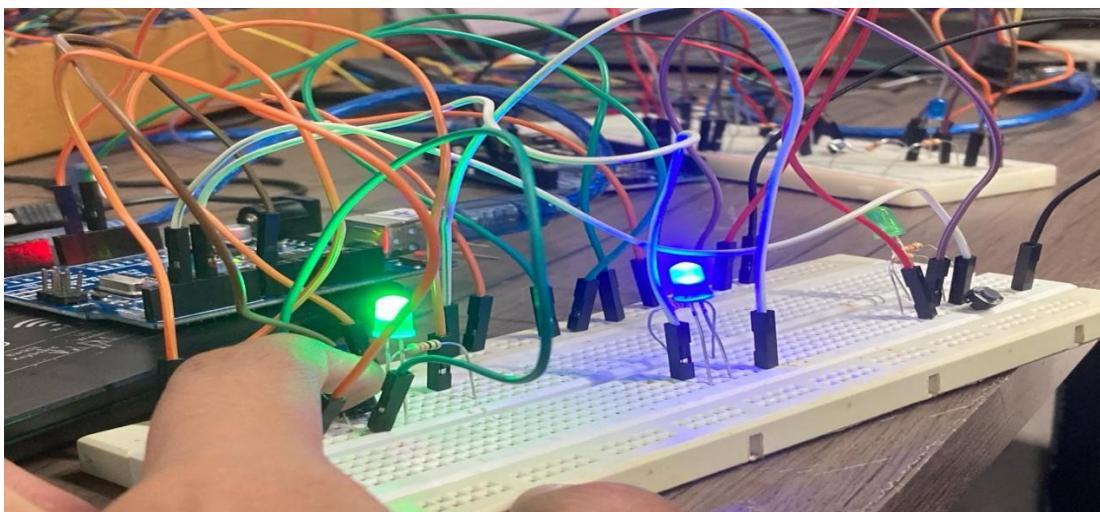


Figure 5.3: Input1=Low, Input2=High

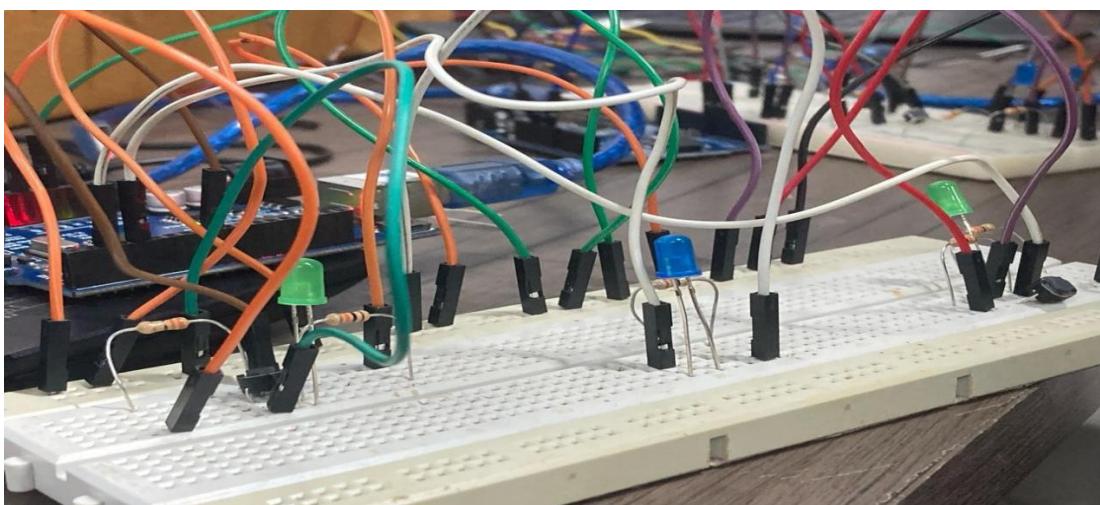


Figure 5.4: Input1=Low, Input2=Low