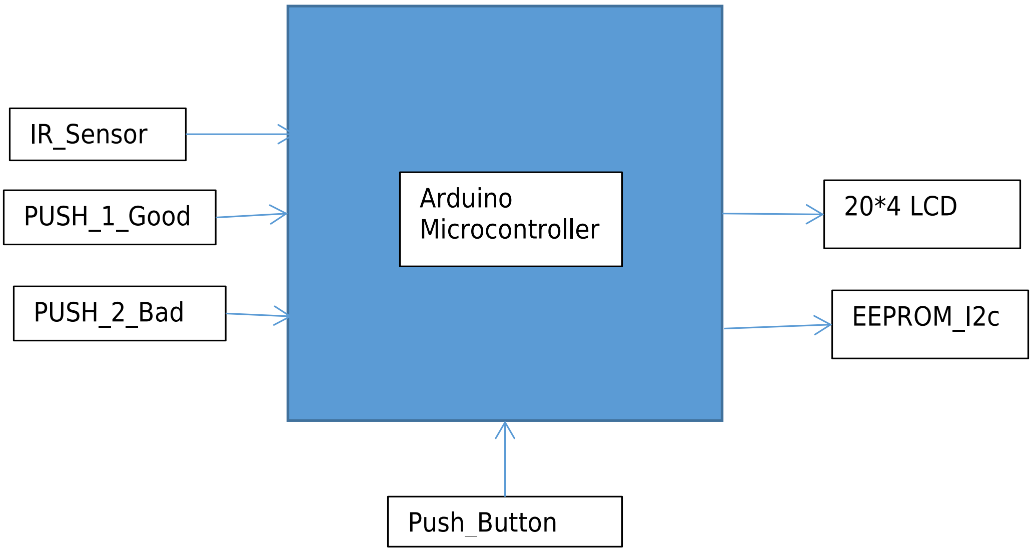
4. Automated Product/Object Counter using Arduino Microcontroller

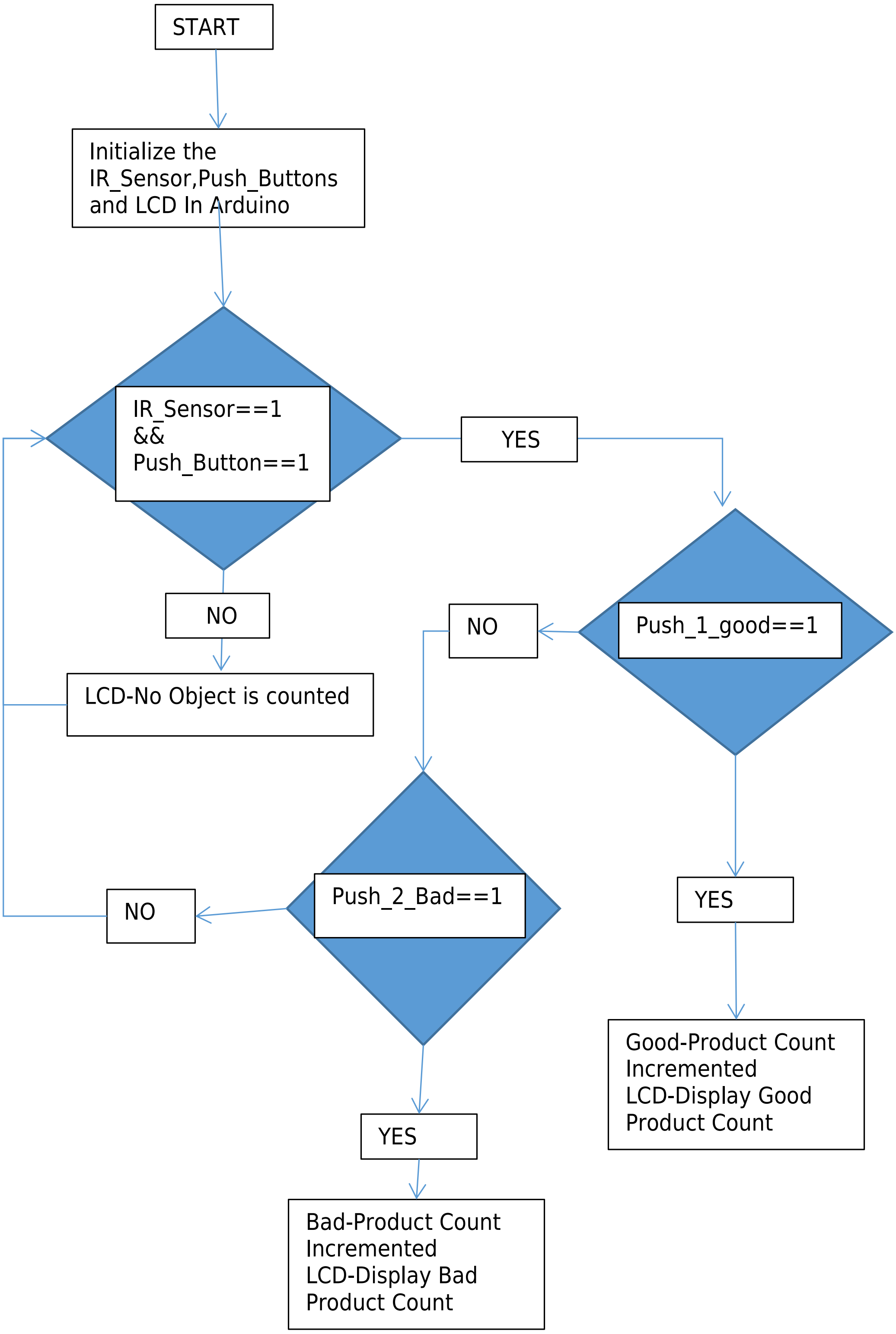
1. Block Diagram



2. Table

s.no .	Discription	Name	Type	Data Direction	Specification	Remarks
1.	***20*4 LCD Display***	20*4 LCD	*OUTPUT*	DO	*5VDC*	
2.	***Pushbutton***	PUSH_1_Good	*INPUT*	DI	*5VDV*	
3.	***Pushbutton***	*Push_2_Bad*	*INPUT*	DI	*5VDC*	
4.	***IR_Sensor***	IR_ Sensor	*INPUT*	DI	*N/A*	
5.	***Pushbutton***	Push_Button	*INPUT*	DI	*5VDC*	
6.	***EEPROM***	EEPROM_I2C	*OUTPUT*	DO	*5VDC*	

3. *Flowchart*


START
Initialize the IR_Sensor,Push_Buttons and LCD In Arduino
IR_Sensor==1 && Push_Button==1
YES
NO
LCD-No Object is counted
Push_1_good==1
NO
YES
Good-Product Count Incremented LCD-Display Good Product Count
Push_2_Bad==1
NO
YES
Bad-Product Count Incremented LCD-Display Bad Product Count

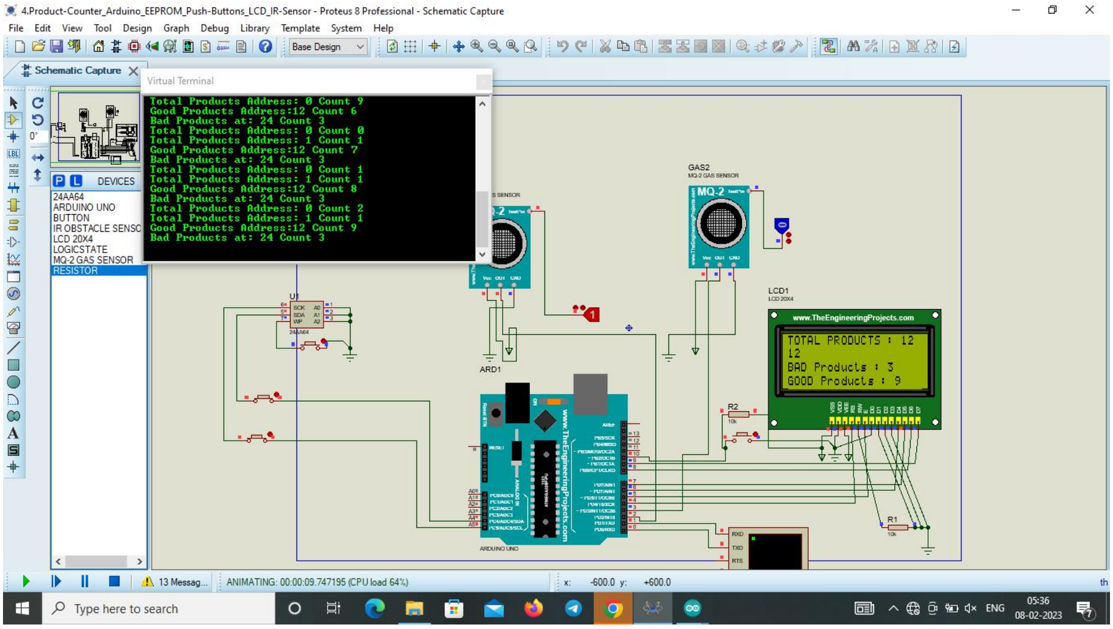
4. Code

```
#include <Wire.h>
#include <LiquidCrystal.h>

#define TotalAddress 0x00 //Intail Address to Store Total Products Count
#define GoodAddress 0x0C //Intail Address to Store Bad Product Count
#define BadAddress 0x18 //Intail Address to Store Good Product Count

int bad = 0;
int good = 0;
int total = 0;
const int eeprom_address = 0x50;   // I2C address of the 24LC256 EEPROM

void LCD_print();
void Clear_EEPROM();

LiquidCrystal lcd(4, 5, 6, 7, 8, 9);

void setup()
{
  Wire.begin();
  Serial.begin(9600);
  for(int i=2 ; i<4 ; i++)
  {
    pinMode(i , INPUT);
  }

  pinMode(10 , INPUT);
  Serial.print("Automated Product Counter System ");
  lcd.begin(20, 4);
  lcd.print("--Product Counter--");
  lcd.setCursor(0, 1);
  lcd.print("        By         ");
  lcd.setCursor(0, 2);
  lcd.print(" Venkata Krishnaiah ");
  delay(1000);
  lcd.clear();


  for (int address = 0; address < 36; address++) //Reset all the Data of
Total, Bad and Good Product Count that Program is loaded before Starting
  {
    writeEEPROM(TotalAddress+address, 0xFF);
  }

}
```

```
void loop()
{

 int clearROM = digitalRead(10); //If clear button is pressed it gives 0 else it
returns 1

 if(!clearROM) //clears EEPROM when it returns 0
 {
  Clear_EEPROM(); //clear the EEPROM
 }

 else
 {
  if(digitalRead(2))
  {
   good++;
  }
  if(digitalRead(3))
  {
   bad++;
  }
  total = good + bad ;
  LCD_print();

  byte i = 0x00 ;
  for(int num = total ; num > 0 ; num = num/10)
  {
   int rem = num % 10 ;
   writeEEPROM(TotalAddress+i, rem); // Writing Total Products Count Byte
by Byte

   Serial.print(" Total Products Address: ");
   Serial.print(TotalAddress+i);
   Serial.print(" Count ");
   Serial.println(rem);


   i++;
  }

  byte j = 0x00 ;
  for(int num = good ; num > 0 ; num = num/10)
  {
   int rem = num % 10 ;
   writeEEPROM(GoodAddress+j, rem); // Writing Good Products Count
Byte by Byte

   Serial.print(" Good Products Address:");
   Serial.print(GoodAddress+j);
```

```
      Serial.print(" Count ");
      Serial.println(rem);

      j++;
    }

    byte k = 0x00 ;
    for(int num = bad ; num > 0 ; num = num/10)
    {
      int rem = num % 10 ;
      writeEEPROM(BadAddress+k, rem); // Writing Bad Products Count Byte
by Byte

      Serial.print(" Bad Products at: ");
      Serial.print(BadAddress+k);
      Serial.print(" Count ");
      Serial.println(rem);

      k++;
    }
  /*
    // Read data from the EEPROM
    byte data = readEEPROM(2);
    Serial.print("Data: ");
    Serial.println(data);
    data = readEEPROM(3);
    Serial.print("Data: ");
    Serial.println(data); */


  }

  delay(500); // Wait for 0.5 second before writing and reading the data
again

}

byte readEEPROM(int address)
{
  byte data;
  Wire.beginTransmission(eeprom_address);
  Wire.write((int)(address >> 8));   // Send the high byte of the address
  Wire.write((int)(address & 0xFF)); // Send the low byte of the address
  Wire.endTransmission();
  Wire.requestFrom(eeprom_address, 1);
  if (Wire.available())
  {
    data = Wire.read();
  }
  return data;
}

void writeEEPROM(int address, byte data) {
  Wire.beginTransmission(eeprom_address);
  Wire.write((int)(address >> 8));   // Send the high byte of the address
  Wire.write((int)(address & 0xFF)); // Send the low byte of the address
  Wire.write(data);
  Wire.endTransmission();
  delay(5);  // wait for the EEPROM to complete the write
}

void LCD_print()
{
  lcd.clear();

  lcd.setCursor(0, 2);
  lcd.print("BAD Products : ");
  lcd.print(bad);

  lcd.setCursor(0, 3);
  lcd.print("GOOD Products : ");
  lcd.print(good);

  lcd.setCursor(0, 0);
  lcd.print("TOTAL PRODUCTS : ");
  lcd.print(total);

}
void Clear_EEPROM()
{
   for (int address = 0; address < 32768; address++)
   {
    Wire.beginTransmission(eeprom_address);
    Wire.write((int)(address >> 8));
    Wire.write((int)(address & 0xFF));
    Wire.write(0xFF);
    Wire.endTransmission();
  }

    Serial.println("EEPROM data cleared!");
}
```

5. Simulation



4.Product-Counter_Arduino_EEPROM_Push-Buttons_LCD_IR-Sensor - Proteus 8 Professional - Schematic Capture
File Edit View Tool Design Graph Debug Library Template System Help
Base Design
Schematic Capture
Virtual Terminal
DEVICES
24AA64
ARDUINO UNO
BUTTON
IR OBSTACLE SENSC
LCD 20X4
LOGICSTATE
MQ-2 GAS SENSOR
RESISTOR
U1
GAS2
MQ-2 GAS SENSOR
ARD1
LCD1
LCD 20X4
www.TheEngineeringProjects.com
R2
R1
13 Messag...
ANIMATING: 00:00:09.747195 (CPU load 64%)
x: -600.0 y: +600.0
Type here to search
ENG
05:36
08-02-2023

```
Total Products Address: 0 Count 9
Good Products Address:12 Count 6
Bad Products at: 24 Count 3
Total Products Address: 0 Count 0
Total Products Address: 1 Count 1
Good Products Address:12 Count 7
Bad Products at: 24 Count 3
Total Products Address: 0 Count 1
Total Products Address: 1 Count 1
Good Products Address:12 Count 8
Bad Products at: 24 Count 3
Total Products Address: 0 Count 2
Total Products Address: 1 Count 1
Good Products Address:12 Count 9
Bad Products at: 24 Count 3
```

```
TOTAL PRODUCTS : 12
12
BAD Products : 3
GOOD Products : 9
```