1. ***Automated Product/Object Counter using Arduino Microcontroller***
2. ***Block Diagram***

IR\_Sensor

Arduino

Microcontroller

PUSH\_1\_Good

20\*4 LCD

EEPROM\_I2c

PUSH\_2\_Bad

Push\_Button

1. ***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* |  |

1. ***Flowchart***

START

Initialize the IR\_Sensor,Push\_Buttons and LCD In Arduino

IR\_Sensor==1 && Push\_Button==1

YES

NO

Push\_1\_good==1

NO

LCD-No Object is counted

Push\_2\_Bad==1

YES

NO

Good-Product Count Incremented

LCD-Display Good Product Count

YES

Bad-Product Count Incremented

LCD-Display Bad Product Count

1. ***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);

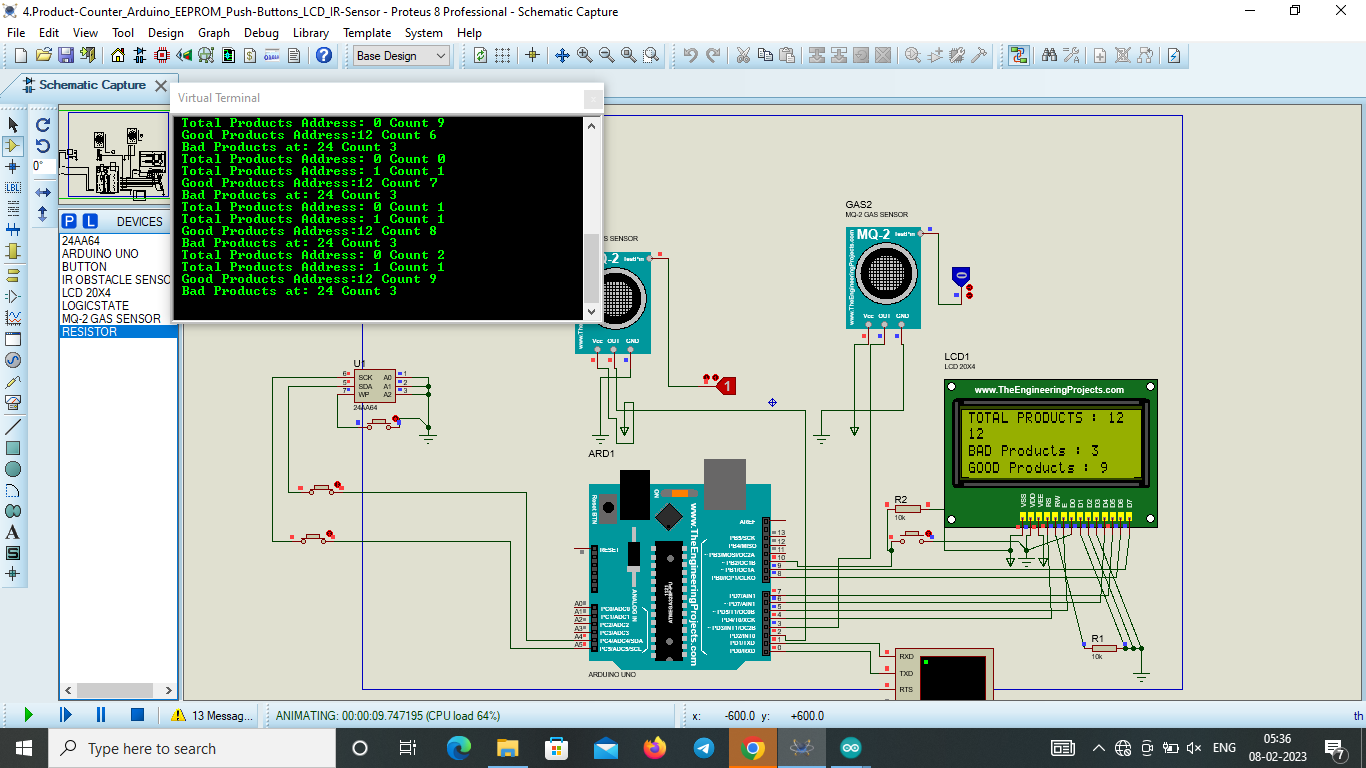
Wire.endTransmission();

}

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

}

1. ***Simulation***

******