

Title: Smart Billing System using object counting for departmental store.

Introduction:

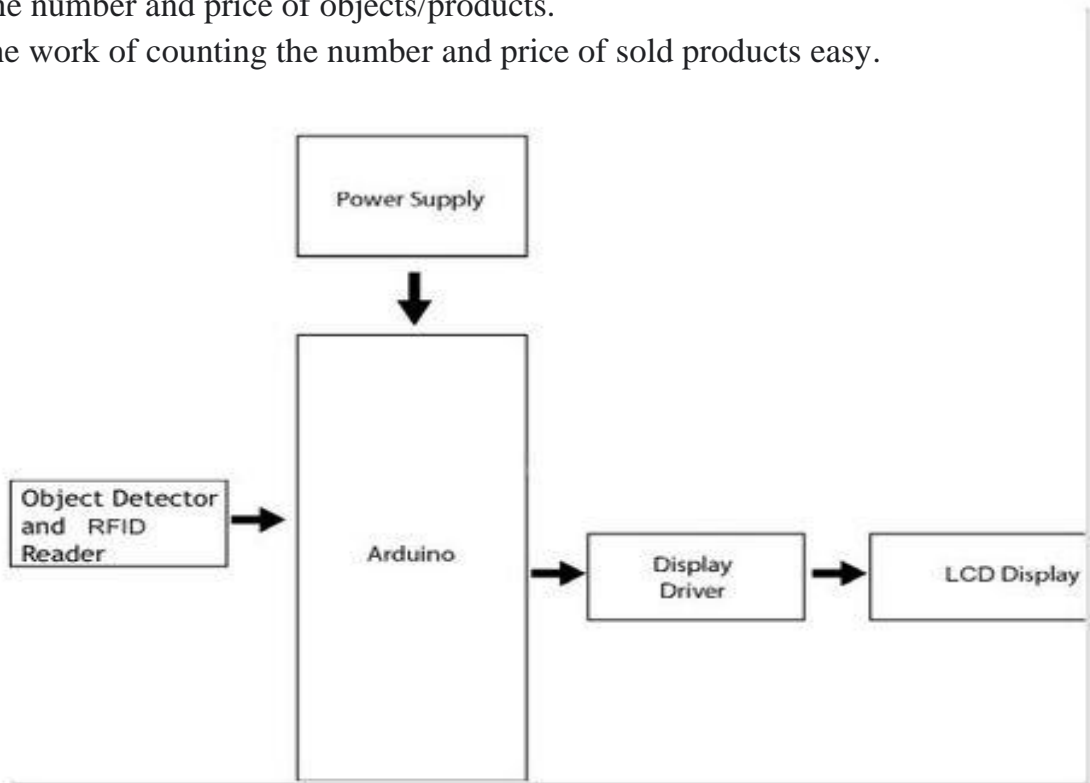
Counting the number of sold products and their price is a part and parcel of a departmental store. The process is quite complex while selling a huge number of products. Smart Billing System using object counting is a simple solution to this complex problem. This project scans the projects one by one, counts their number and calculates their price. When one is done with all the sold products, it shows the total number of products and their price using an LCD screen. Thus, counting the total number of sold products and calculating their price becomes very east.

The process of counting the objects and their price can be can be done manually, but that produces the risk of miscalculating the number of products and their price since a departmental store may be very much busy at times. Using this system, the process becomes risk free of miscalculation, saves the time of selling a product and also reduces the work pressure of a salesman. However, the project is all about making the selling of a product of a departmental store as easy and risk free as possible.

Objectives:

- To understand how RFID (Radio Frequency Identification) sensor and Transmitter work.
- To understand how Arduino works and how to interact with it.
- To count the number and price of objects/products.
- To make the work of counting the number and price of sold products easy.

Model Design:



Major Components/Materials:

Hardware:

1. Arduino UNO
2. LCD (Liquid Crystal Display)
3. 16 × 2 LCD Module
4. Power Supply Module
5. RFID Sensor

Software/Language:

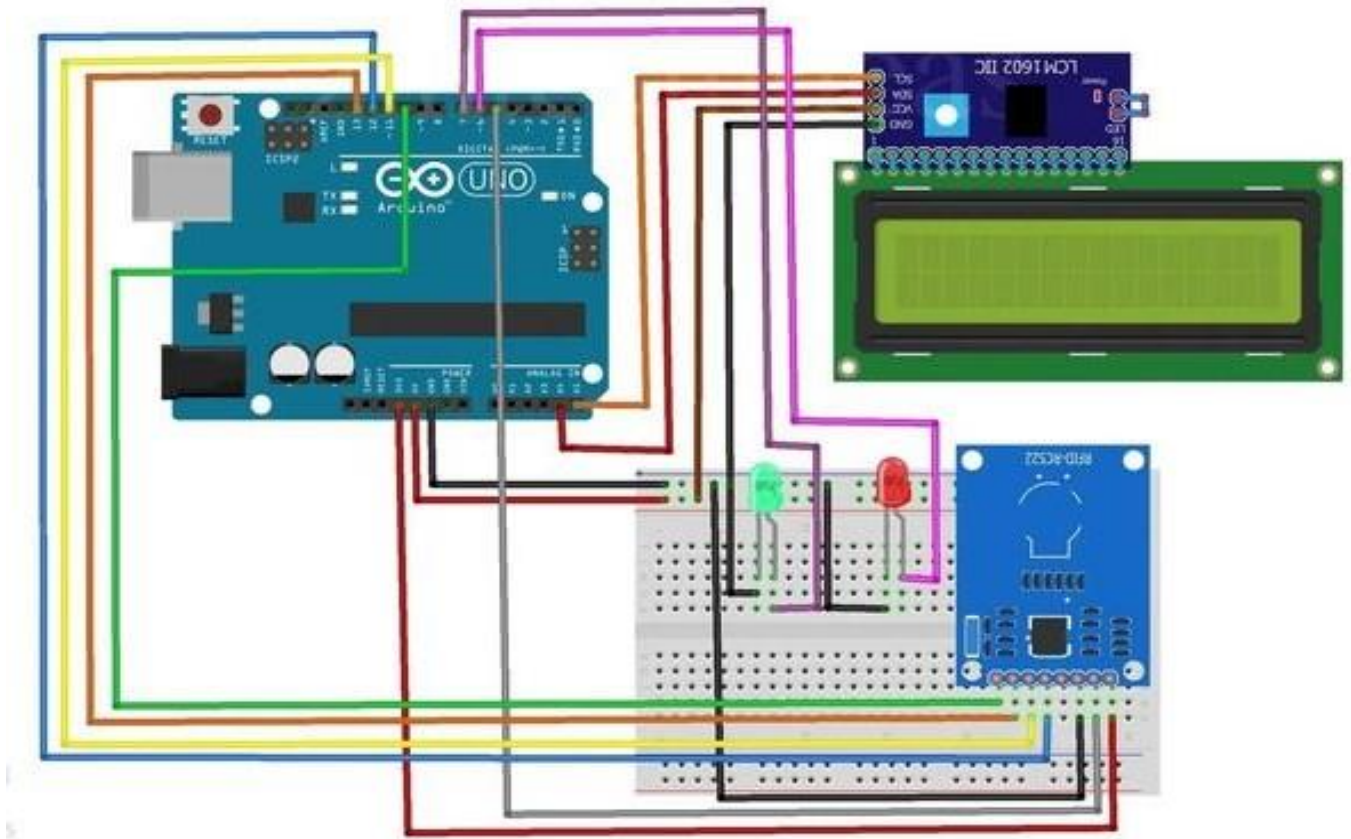
1. Arduino IDE
2. Programming in C/C++

Budget Proposal:

1. Arduino UNO = 1100 tk.
2. LCD = 200 tk.
3. 16 × 2 LCD Module = 150 tk.
4. Power Supply Module = 100 tk.
5. RFID Sensor = 200 tk.
6. Jumper Wire 40 Pcs Set = 100 tk.
7. Others = 200 tk.

Total = 2050 tk (approx.)

Circuit Diagram:



Implementation Details:

1. At first, the hardware required to accomplish the project were placed properly.
2. Then the connections were formed between the components as shown in the figure.
3. The Arduino UNO was connected with the object sensor to sense the products, power supply to supply necessary power to the circuit and LCD to show the count and price of the products.
4. The next process was coding. We have code using the programming language C on Arduino IDE and made the Arduino UNO work as our requirement.
5. There was a control set in the breadboard to control the power ON/OFF state of the system.
6. The RFID reader was connected to the suitable pin of the Arduino UNO and the work after sensing a product was defined through the code.
7. Finally, the project was accomplished by completing all the required processes.

Conclusion:

The main goal of the project Smart Billing System using object counting for departmental store is to make the work of counting the number of sold products and their price in a departmental store as easy as possible. Using this project, the time required to do such works will be lessen and the calculation of the price of total sold products will be without mistakes. There is a demerit of this prototype besides these merits. The RFID sensor can sense one object at a time and only one altogether in this prototype. So, more amount of tags needs to be added to the project and of course to the code to make the project more usable and portable.

However, there were some difficulties faced while completing the project such as problems while connecting the components, code was not being compiled, sensor was not working and so on. The project was successfully completed at last.

Code:

```
// Including Libraries
#include <SPI.h>
#include <MFRC522.h>
#include <LCD_I2C.h>
// Defining Pins
LCD_I2C lcd(0x27, 16, 2);
#define RST_PIN 9
#define SS_PIN 10
// Defining Tags
byte readCard[4];
```

```
String MasterTag = "13EB26";
String tagID = "";
MFRC522 mfrc522(SS_PIN,
RST_PIN);
int totalPrice = 0, item = 0; //
initial price and item is 0
const unsigned long
MESSAGE_INTERVAL = 3000; //
message switch interval in
milliseconds
```

```

unsigned long lastMessageTime =
0; // time of last message
switch
bool showMessageScan = true; //
flag indicating which message to
display
void setup() {
    lcd.begin();
    lcd.backlight();
    SPI.begin(); // SPI bus
    mfrc522.PCD_Init(); // MFRC522
    lcd.clear();
    lcd.setCursor(2, 0);
    lcd.print("Scan Product"); //
initial message on display
}

void loop() {
    Serial.begin(9600);
    // switch messages every
MESSAGE_INTERVAL milliseconds
    unsigned long currentTime =
millis();
    if (currentTime -
lastMessageTime >=
MESSAGE_INTERVAL) {
        lastMessageTime =
currentTime;
        showMessageScan =
!showMessageScan;
        // update display with new
message
        lcd.clear();
        if (showMessageScan) { //
showing the message
            lcd.setCursor(2, 0);
            lcd.print("Scan Product");
        } else {
            lcd.print("Total Item: ");
            lcd.print(item);
            lcd.setCursor(0, 1);
            lcd.print("Total Bill: ");
            lcd.print(totalPrice);
        }
    }
    // checking for new RFID tag
while (getID()) {
    if (tagID == MasterTag) {
        totalPrice += 50;
        ++item;
        // updating display with new
item and total price

```

```

        lcd.clear();
        lcd.print("Total Item: ");
        lcd.print(item);
        lcd.setCursor(0, 1);
        lcd.print("Total Bill: ");
        lcd.print(totalPrice);
    }
    else {
        // display final totals and
reset counters
        lcd.clear();
        lcd.setCursor(2, 0);
        lcd.print(item);
        if (item <= 1) lcd.print("
Item Sold");
        else lcd.print(" Items
Sold");
        lcd.setCursor(3, 1);
        lcd.print("at TK. ");
        lcd.print(totalPrice);
        delay(5000);
        totalPrice = item = 0;
    }
}

boolean getID() {
    if
(!mfrc522.PICC_IsNewCardPresent(
)) { // If a new PICC placed
to RFID reader continue
        return false;
    }
    if
(!mfrc522.PICC_ReadCardSerial())
{ // Since a PICC placed get
Serial and continue
        return false;
    }
    tagID = "";
    for (uint8_t i = 0; i < 4; ++i)
    { // The MIFARE PICCs that we
use have 4 byte UID
        tagID.concat(String(mfrc522.uid.
uidByte[i], HEX)); // Adds the
4 bytes in a single String
variable
    }
    tagID.toUpperCase();
    mfrc522.PICC_HaltA(); // Stop
reading
    return true;
}

```

Output:

