EE381: EC LAB, PROJECT REPORT

Inventory Management & Automation System

(Initial Idea - Speed Sensor for Traffic Control)

Submitted by

1.Adrija Bera (210071)
2. Aniket Sen (210134)
3. Pratyush Amrit (210762)

Section – A, Table No. - 14

Bachelor Of Technology in Electrical Engineering

INDIAN INSTITUTE OF TECHNOLOGY,

KANPUR

April 2024

Under the guidance of Prof. Baquer Mazhari

Department of Electrical Engineering, IIT Kanpur

Table of Contents

Why did we change the theme ?

Our initial project was to build a Speed Sensor (with personal identification using RFID). However, as the project advanced, we encountered several issues, few of them were :

1. The **range** of a typical IR sensor available to us in the lab was at max 30 cm which is **too small** to track the speed of incoming traffic on roads, (a safe distance also needs to be maintained).

2. The whole setup was meant to be used **outdoors** which reduced **efficiency** and accuracy of IR sensors during day time.

3. We intended to identify the person violating the speed limit and send his information on our database. For this, we started off with using **RFID** but the RF emitter in the lab was **passive**; meaning it required a contact with RF receiver to detect the unique ID. For our application, this was not possible, although one of the solutions could have been to use an **Active RF emitter**.

After analysing these problems, as advised by the instructor, we tried to devise an application which could possibly use the basic working of IR sensors as well as give more accurate results.

So we thought of developing an "Inventory Management & Automation System" which uses the same key idea as our previous idea.

Inventory Management & Automation System

DESCRIPTION :

What problem are you trying to solve, and why is it important/interesting?

We try to capture the problem of keeping a record of inflow/outflow of a system. As the name suggests, in our project we record entries into laboratories/office space by swiping left or right, and capture images through our integrated camera module for precise timestamping and authentication.
This can be very well implemented in our EC labs where students may take and use things from inventory while the track record is maintained using automated databasing through our device.
We may further extend it to precisely keeping record of material taken in/out of the workspace through identification.

Why it's Important/Interesting:

- **Cost-effective alternative:** Compared to complex software solutions, Arduino offers a budget-friendly approach for basic inventory tracking.
- **Simple setup and use:**setting up and maintaining this system is relatively straightforward.
- **Improved accuracy:** Photos provide visual verification of stock changes, minimising errors compared to manual counting.
- **Data for analysis:** Captured images can be used later to analyze trends, identify popular items, or even implement rudimentary theft detection.
- **Real-time inventory visibility:** You can see exactly how much stock you have of each item at any given time as the data is sent to the smartphone in real time.
- **Automated monitoring :** Cuts off the need for manual monitoring, being fully automated with high accuracy and credibility of results.

What are the existing solutions?

Several inventory management solutions exist, each with pros and cons.

1. **Manual Spreadsheets**: Simple, but error-prone and limited for large inventories.
2. **Inventory Management Software:** Powerful software offers real-time tracking, reporting, and automation, but can be expensive and complex.
3. **Barcode Scanners & Point-of-Sale Systems**: Efficient for check-in/out, but may lack features and require extra hardware.

Shortcomings of Existing Solutions:

- **Cost:** Software solutions can be expensive, while scanners require additional hardware investment.
- **Complexity:** Feature-rich software can have a steep learning curve for non-technical users.
- **Manual** : Might not be possible to automate fully, given results need to be accurate and reliable

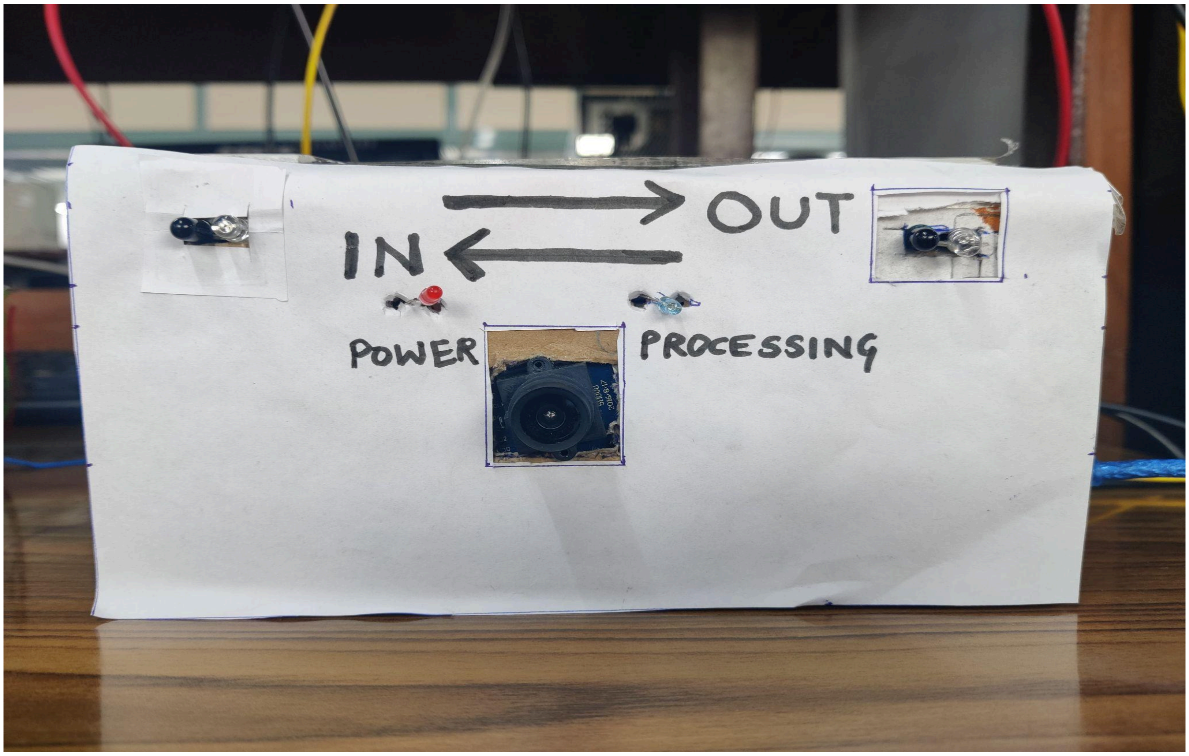
Uniqueness of the Arduino Approach:

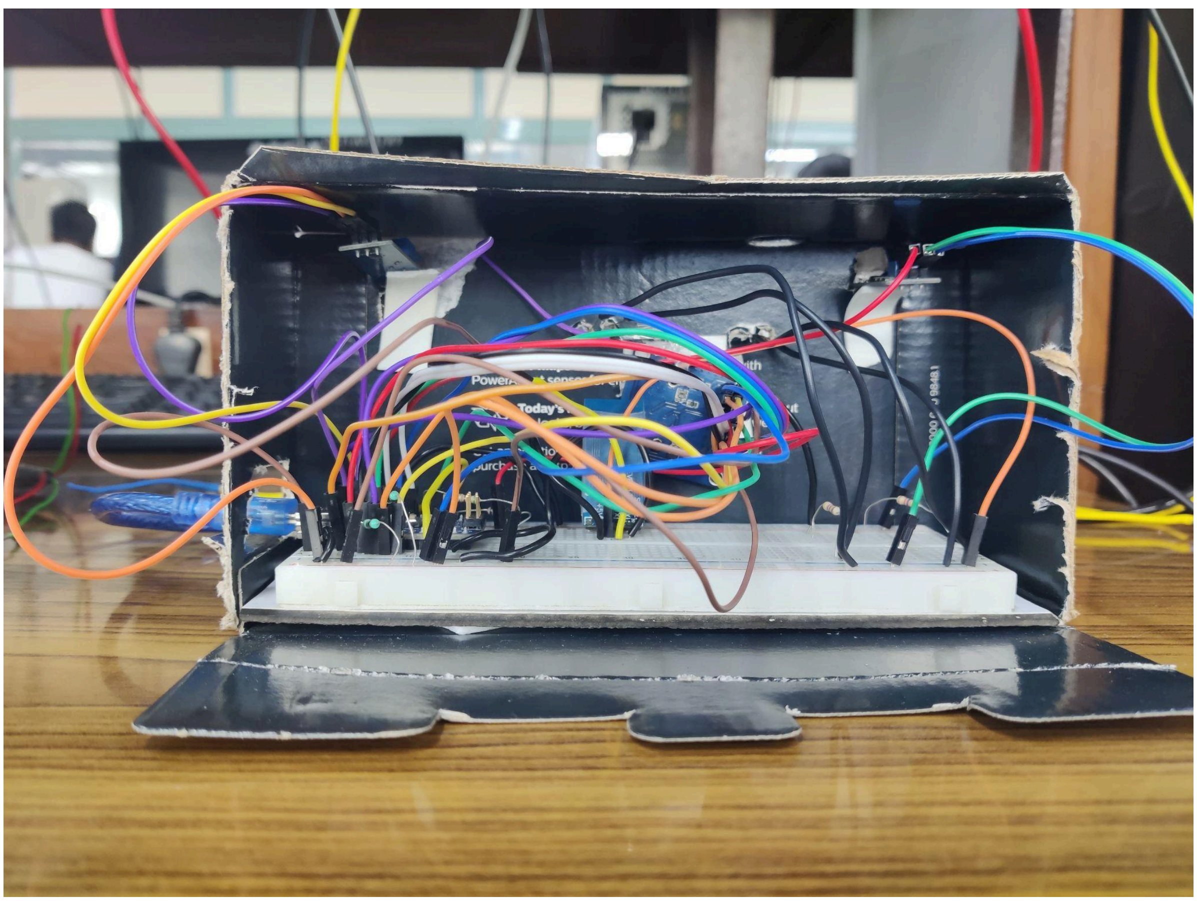
The Arduino-based system offers a unique approach with several advantages:

- **Customizable:** Can be tailored to specific needs with additional coding or hardware modifications.
- **Cost-effective:** Utilises readily available and relatively inexpensive components.
- **Simple Setup:** Easier to set up and maintain compared to complex software.
- **Real-time monitoring:** You can see exactly how much stock you have of each item at any given time.

Overall, the Arduino solution provides a low-cost, user-friendly option for basic inventory tracking, particularly for smaller businesses or personal projects. While it can't compete with comprehensive software in terms of features, it offers a unique and accessible alternative for specific needs.

DESIGN :





Guideline to use the Device :

Follow the guidelines for the Inventory Management if you are going inside the room :

- Swipe across the region from top-right to top-left if you are getting inside the room with some item. Make sure the power LED is ON.
- The processing LED would get powered which would ensure you that the process has been initiated.
- Stand still till the LED gets off by facing the camera to process the image with the specific item.
- Once the LED gets off, this would imply the processing is done and you would get a message in your phone about the updated entry recorded in our database.

Follow the guidelines for the Inventory Management if you are going outside the room :

- Swipe across the region from top-left to top-right if you are getting outside the room with some item.
- The processing LED would get powered which would ensure you that the process has been initiated.
- Stand still till the LED gets off by facing the camera to process the image with the specific item.
- Once the LED gets off, this would imply the processing is done and you would get a message in your phone about the updated exit recorded in our database.

Working:

The hand gesture is detected by the IR sensor as an IN or OUT process using the time-stamps recorded. These are recorded at the moment a disturbance in the line of sight of the IR sensor is detected. So, consider for IN process when we wave our hand from IR_sensor 1 to IR_sensor 2 :

$$t_1 < t_2$$

where : t_1 = time recorded at IR_sensor 1
t_2 = time recorded at IR_sensor 2

Hence the device detects an IN process. Similarly an Out process can be detected. Now, once a process is detected, the processing LED is turned ON to let the user know that they have successfully initiated a check-in/check-out process.

Once the process is initiated, the camera module is activated automatically and the photo of the user along with the item is clicked. Further, it is either stored on the SD card or on the desktop.

All the live updates of the procedure are sent to the owner/authority of the Inventory using the HC-05 bluetooth module through the smartphone. After the photo is clicked and stored, the processing LED turns off indicating that the device is ready for next check-in/check-out.

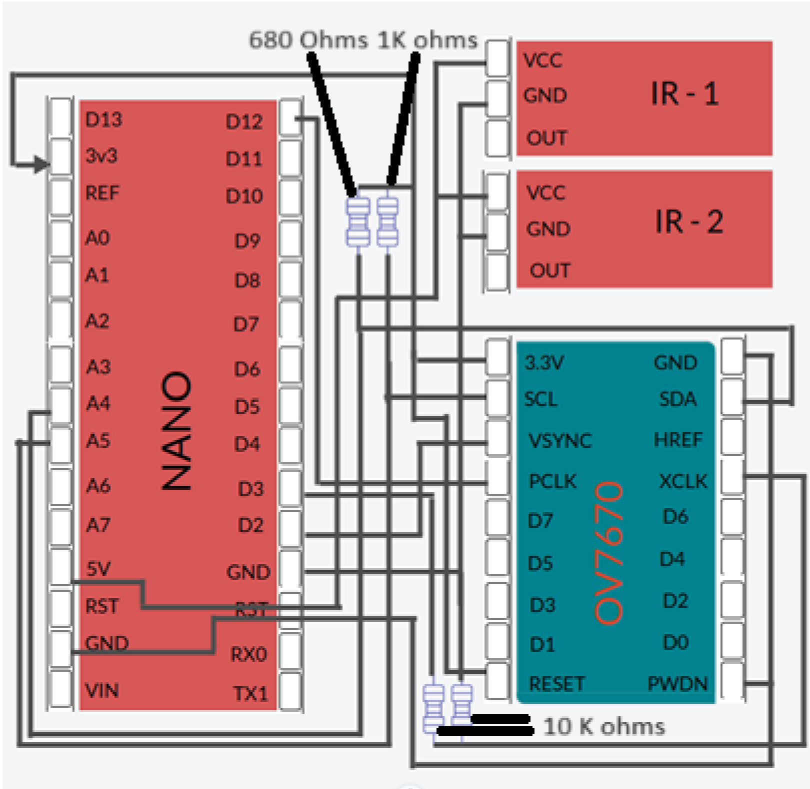
The power LED is switched on throughout the duration when power is supplied to the device. This informs the user that the device is presently switched ON. Hence, it is directly connected between 5V arduino output and ground, through a resistor.

Basically, IR sensors track time stamps of motion. In the case of Speed Sensor, we use this timestamps' data to analyse the speed of moving objects.
Similarly, here in case of Inventory Management, we track timestamps to decide on the process initiated. But, the basic essence of the use of IR sensors remains the same.

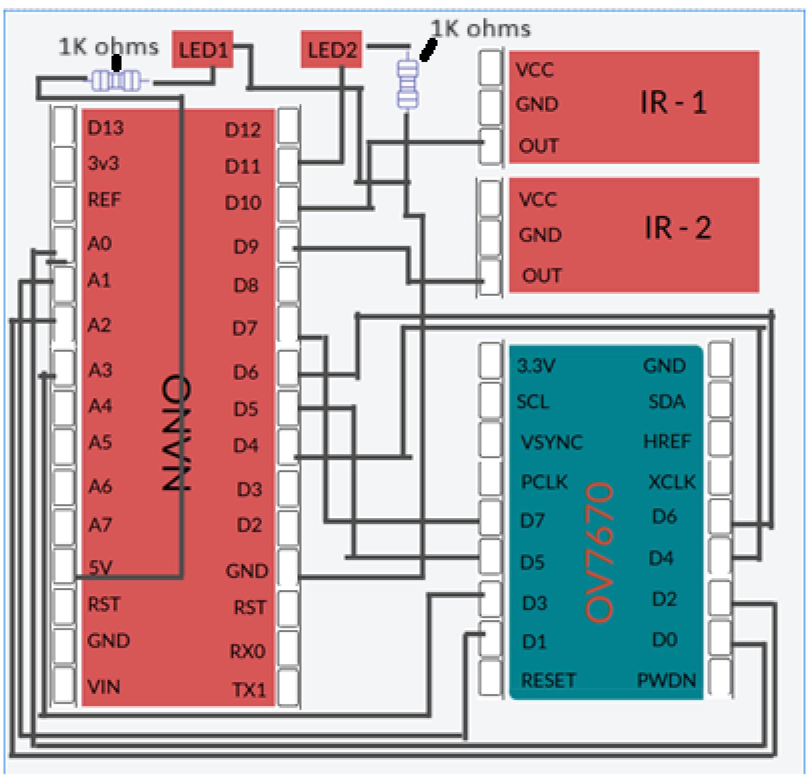
Components Used :

1. Arduino Nano
2. Digital IR Sensors (x2)
3. Camera Module OV7670
4. Breadboard
5. HC-05 Bluetooth Module
6. Blue LED, Red LED
7. Power Supply (preferably through USB cord using laptop)
8. Bluetooth App Terminal
9. Two 10kΩ, three 1kΩ and one 680Ω resistors

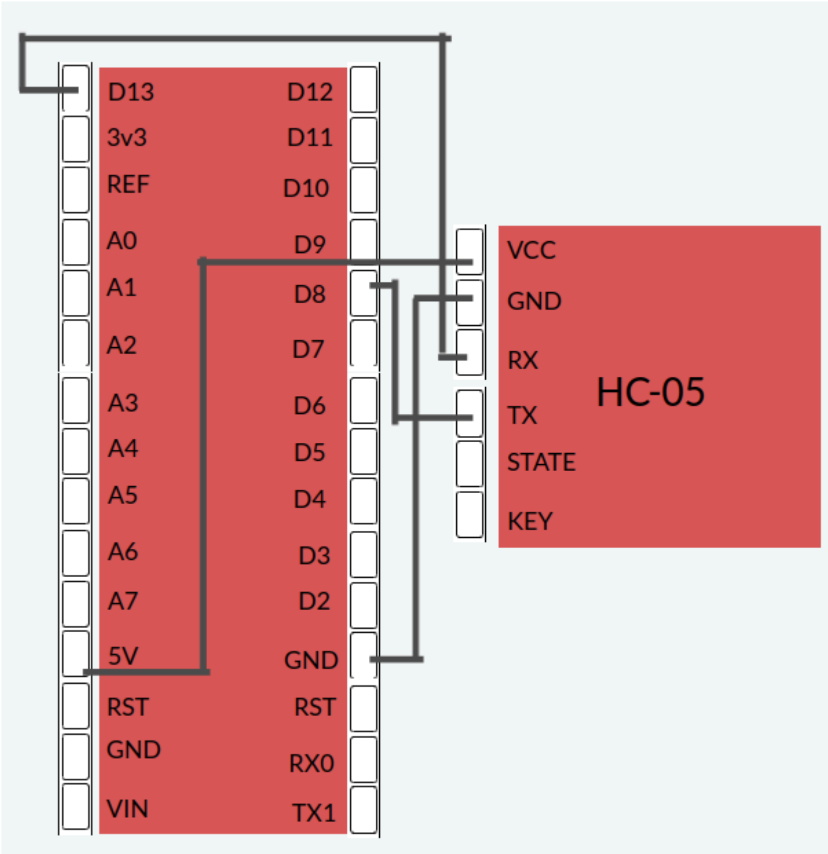
Circuit Diagram :



Connections of camera (OV7670) and 2 IR sensors, Resistors also shown



Power and Processing LEDs are added along with remaining connections of camera module (OV7670)



Connections with HC-05 bluetooth module

Code Used :

```
#include "setup.h"             // library used for camera module
#include <SoftwareSerial.h>    // for bluetooth
int timer1;                    // variable for recording time stamp
int timer2;                    // of each IR sensor

float Time;

int flag1 = 0;                 // flag for a detection in IR_sensor1
int flag2 = 0;                 // flag for a detection in IR_sensor2
int detect=0;                  // flag for a detection
int INFLAG = 0;                // flag for IN operation
int OUTFLAG = 0;               // flag for out operation

int totalin = 0;               // counting total no. of IN process
int totalout = 0;              // counting total no. of IN process

int RX = 13;                   // defining RX pin of bluetooth
int LED_Processing = 11;       // processing led pin
int TX = 8;                    // defining TX pin of bluetooth

int ir_s1 = 9;                 // IR_sensor 1 pin
int ir_s2 = 10;                // IR_sensor 2 pin

SoftwareSerial BT(RX, TX);     // SRX | STX

void setup() {

  BT.begin(9600);
  BT.println("hello!");        // printing setup message on smartphone using tooth
  CLKPR = 0x80;                // enter clock rate change mode
  CLKPR = 0;                   // set prescaler to 0. WAVGAT MCU has it 3 by default.

  initializeScreenAndCamera();
  delay(5000);                 // for setup
  processFrame();
  delay(2000);

  pinMode(ir_s1, INPUT);
  pinMode(ir_s2, INPUT);
}

void loop() {

if(flag1 == 1 && flag2==0 && millis()-timer1> 1000){timer1 = millis(); flag1=0;}
if(flag2 == 1 && flag1==0 && millis()-timer1> 1000){timer2 = millis(); flag1=0;}
if(digitalRead (ir_s1) == LOW && flag1==0){timer1 = millis(); flag1=1;}
if(digitalRead (ir_s2) == LOW && flag2==0){timer2 = millis(); flag2=1;}

if (flag1==1 && flag2==1){
    detect=1;
    //digitalWrite(LED_Idle, LOW);
```

```
    digitalWrite(LED_Processing, HIGH);
    BT.println("PROCESSING");

    if(timer1 > timer2){

    OUTFLAG = 0;
    BT.println("IN");
    totalin = totalin + 1;

    }
else if(timer2 > timer1)
 {
   OUTFLAG = 1;
   BT.println("OUT");
   totalout = totalout + 1;

  }

}
if(detect==0)
 {
  //Serial.println("Idle");
}
else{
    processFrame();
    if(OUTFLAG == 0)
    {
        BT.print("IN, totalin :");
        BT.println(totalin);
    }
    if(OUTFLAG == 1)
    {
        BT.print("OUT, totalout :");
        BT.println(totalout);
    }
    delay(2000);
    BT.println("DONE!!!!!!");

    digitalWrite(LED_Processing, LOW);
    delay(2000);

    detect = 0;
    flag1 = 0;
    flag2 = 0;
 }
}

```

This code was fed to the Arduino Nano which controlled all the sensors present and gave results on the terminal. An additional bluetooth terminal interface was used to get live updates on the process as well.

Workflow

Week 1: Familiarisation with Arduino Nano, HC-05 bluetooth module and IR sensors
We studied all the different sensors required in our project including HC-05 bluetooth module, IR sensors, Arduino Nano and implemented small tasks such as printing resistor values etc.

Week 2: Implemented speed sensor successfully and found flaws while performing identification using RFID.
Successful implementation of the speed sensor was done and results were printed on the terminal on our laptop. However, there was significant inaccuracy in results. Moreover, when efforts were made to implement the process of identification using RFID, we encountered the problems which were listed before.
As advised by the instructor, we brainstormed to either solve this problem using a certain variation of a sensor or try to identify an application which used similar principles as before.

Week 3: Ideation of "Inventory Management" and familiarisation with OV7670 camera module
After we arrived at the new idea, we familiarised ourselves with the OV7670 camera module and implemented the circuit connections accordingly. After the complete circuit was made, we tested our setup using a Bluetooth interface as well.

Final Result :

The device was implemented successfully with very high accuracy of detection of hand gestures and the photos were saved to the local folder on the desktop. The images can now be used for face recognition using machine learning interfaces.

The video of the working demo, the arduino code and libraries required of the project is attached below.

Inventory Manager Working

Future enhancements :

1. We may use the stored images and use machine learning techniques to identify the person.
2. We can extend this idea of identification to items as well. Using QR codes, we can use a QR code scanner to keep a track of the items moved in/out of the inventory.
3. A bluetooth enabled customised application can be created, which stores the database of timestamps, date and type of process - check-in / check-out.
4. A new feature of theft detection can be added. This can be done using a buzzer and 2 additional IR sensors. When a person enters/exits without following procedure, the direction of motion and hence timestamps would show anomaly and hence, the buzzer would beep.

References :

1. EE381 EC modules list
2. Camera Module (OV7670) working