Department of Computer Science and Engineering

CSE360: Computer Interfacing

Project Report

On

Compact Office Security System

Submitted by:

1. MD Arman Islam — ID: 19101639 — Section: 02
2. Nafis Bin Reza — ID: 17101411 — Section: 02
3. Alim Aldin Rohan — ID: 17101202 — Section: 02
4. Chayan Saha — ID: 17101415 — Section: 02

Submission date: 20th February, 2020

Submitted to: Nazmus Sakeef, Lecturer, BRAC University

Abstract— The objective of the project is to build a Compact Office Security System. The system is designed and implemented using Arduino, Sensors and GPIO devices and I/O devices. With the help of this system, the security of an office or a firm can be ensured under a surveillance monitored area.

Keywords— Arduino, Sensors, GPIO device, I/O devices, Security.

I. Introduction

As newer offices and firms are created, it has become one of our prime concerns to provide strict security to these places when we are not physically present in that area. For which a compact office security system can be introduced which will provide top notch security in different areas such as proper authentication of employers, detection of unwanted presence and alarming everyone in case of danger etc. Arduino; functions almost as a small computer; will be connected with different sensors and GPIO devices in order to achieve our objectives. After building the compact security system, a user can keep track of what is happening in that particular area even when he is not close to that place. Though there is a fingerprint sensor which will verify the employee's identification, if somehow someone breaches the entrance, it will be in our sight as we watch through our phone. That is, when a PIR motion sensor detects an unusual movement inside the office, it will provide essential information and transmit it to a smartphone using a web application. Despite of security breaches, if suddenly some unwanted disaster occurs such as fire, the smoke sensor will instantly detect it and will notify us via Bluetooth. Moreover, the smoke sensor will use an LCD to show a message whether everything is fine or not. The advantage of this system is that it covers different areas when it comes to security and it can be easily made using Arduino, sensors, GPIO devices and I/O devices.

II. Application Area

The Compact Office Security System provides security mainly in 4 different areas.

- Authentication of employers
- Detection of unwanted presence
- Alarming in case of danger
- Surveillance on a particular area

Authentication of employers: Fingerprint sensor/Touch sensor along with USB to serial converter, push buttons and LCD. Everyone who wants to enter the office must have to place his/her finger on the sensor. And the sensor will let the system know if he is permitted to enter or not. Once his/her authentication is verified he will be able to enter inside the office.

Detection of unwanted presence: In order to detect unwanted presence, PIR (passive infrared) sensor is used with a Bluetooth module. This is actually a burglar alarm system. That is, when someone who is not permitted enters the room, the PIR motion sensor will automatically detect it and will notify us.

Alarming in case of danger: Apart from burglary and security breach, there are other unwanted accidents that might occur like fire. In that case, a smoke sensor will detect the smoke and notify us by alarming us via Bluetooth.

Surveillance on a particular area: As a camera is placed inside the office, a user from a different location can access the camera which has the ability to capture image and record videos. So, better security can be provided and necessary initiatives can be taken in case of emergency.

III. Technology and Tools

The following components are used to build the system**:**

- Arduino Uno
- Breadboard
- Fingerprint sensor
- PIR Motion sensor
- Smoke sensor
- Jump wires
- USB cable A to B
- Bluetooth Serial Module
- 470 Ohm Resister
- 1K Ohm Register

- 16x2 LCD (GPIO device)
- I/R Positioning Camera (I/O device)
- Mobile Phone (I/O device)

IV. Language

For this specific task, we have to execute a certain program inside the Arduino. We know, Arduino. It does not run either C or C++. It basically runs machine code compiled from either C, C++ or any other language that has a compiler for the Arduino instruction set using special rules of code structuring.

Though Arduino can be used with python or any other high-level language; for this project, we are using C, C++. The Arduino Integrated Development Environment (IDE) is the main text editing program used for Arduino programming. We will type our code in Arduino IDE before uploading it on our board.

V. Working Mechanisms of Sensors

Finger Print Scanner Sensor: There are two main ways of scanning fingers. An optical scanner works by shining a bright light over your fingerprint and taking what is effectively a digital photograph. If you've ever photocopied your hand, you'll know exactly how this works. Instead of producing a dirty black photocopy, the image feeds into a computer scanner. The scanner uses a light-sensitive microchip (either a CCD, charge-coupled device, or a CMOS image sensor) to produce a digital image. The computer analyzes the image automatically, selecting just the fingerprint, and then uses sophisticated pattern-matching software to turn it into a code. Another type of scanner, known as a capacitive scanner, measures your finger electrically. When your finger rests on a surface, the ridges in your fingerprints touch the surface while the hollows between the ridges stand slightly clear of it. In other words, there are varying distances between each part of your finger and the surface below. A capacitive scanner builds up a picture of your fingerprint by measuring these distances. Scanners like this are a bit like the touchscreens on things like iPhones and iPads.

Smoke Sensor: Ionization smoke alarms are generally more responsive to flaming fires. Ionization-type smoke alarms have a small amount of radioactive material between two electrically charged plates, which ionizes the air and causes current to flow between the plates. When smoke enters the chamber, it disrupts the flow of ions, thus reducing the flow of current and activating the alarm. Download this chart on ionization smoke alarms. Photoelectric smoke alarms are generally more responsive to fires that begin with a long period of smoldering (called “smoldering fires”). Photoelectric-type alarms aim a light source into a sensing chamber at an angle away from the sensor. Smoke enters the chamber, reflecting light onto the light sensor; triggering the alarm. For each type of smoke alarm, the advantage it provides may be critical to life safety in some fire situations. Home fatal fires, day or night, include a large number of smoldering fires and a large number of flaming fires. You cannot predict the type of fire you may have in your home or when it will occur. Any smoke alarm technology, to be acceptable, must perform acceptably for both types of fires in order to provide early warning of fire at all times of the day or night and whether you are asleep or awake.

PIR Motion Sensor: PIRs are basically made of a pyroelectric sensor (which you can see below as the round metal can with a rectangular crystal in the center), which can detect levels of infrared radiation. Everything emits some low-level radiation, and the hotter something is, the more radiation is emitted. The sensor in a motion detector is actually split in two halves. The reason for that is that we are looking to detect motion (change) not average IR levels. The two halves are wired up so that they cancel each other out. If one half sees more or less IR radiation than the other, the output will swing high or low. Along with the pyroelectric sensor is a bunch of supporting circuitry, resistors and capacitors. It seems that most small hobbyist sensors use the BISS0001 ("Micro Power PIR Motion Detector IC"), undoubtedly a very inexpensive chip. This chip takes the output of the sensor and does some minor processing on it to emit a digital output pulse from the analog sensor. For many basic projects or products that need to detect when a person has left or entered the area, or has approached, PIR sensors are great. They are low power and low cost, pretty rugged, have a wide lens range, and are easy to interface with. Note that PIRs won't tell you how many people are around or how close they are to the sensor, the lens is often fixed to a certain sweep and distance (although it can be hacked somewhere) and they are also sometimes set off by house pets.

VI. Connection with ICs

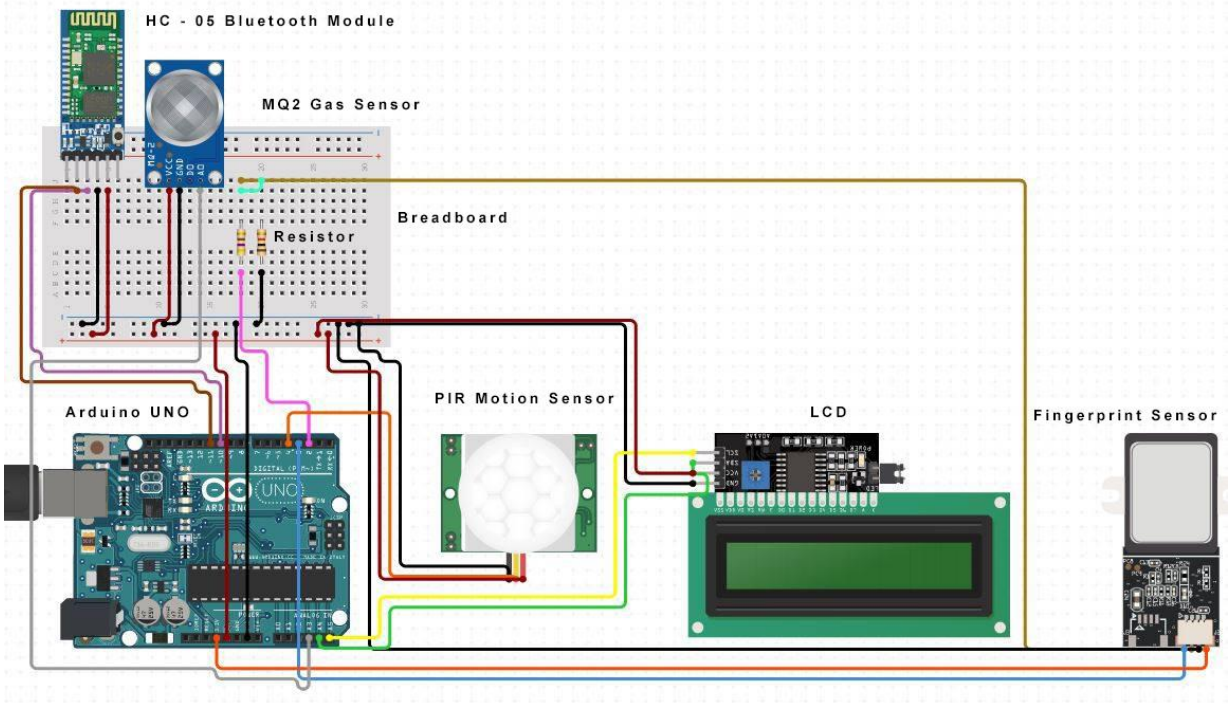

Fig1: Connection of Sensors, GPIO device, I/O device with Arduino

Compact Office Security system is based on computer vision which also give security. The concept of behavioral gestures enables the system to be used for various setups of the environment, making the solution easily extensible with heterogeneous devices and sensors. The platform's processing layer easily triggers the system's awareness, by providing the desired user experience according to the selected gestures. We tried to make the system platform-independent so that it can be used by all classes of people. When the sensor notices any human structure's presence in the room, the motion detection system generates a parameter which then goes to the Arduino uno. This signal is then processed by the Arduino uno using a software which helps to send alert massage to our mobile/ computer.

Fingerprint Sensor:

In a Fingerprint Sensor which has 4 pin VCC, TX, RX and GND are connected with following Arduino pins.

Fingerprint Sensor	**Arduino**
VCC	5v(it also works with3.3v)
TX	RX(digital pin 2, software serial)

RX	TX (digital pin 3, software serial)
GND	GND

Display:

we used a display which uses the SPI protocol in order to communicate with the Arduino board. We only need to connect 8 wires in order to make it work.

The Display is connected with Arduino in following way:

Vcc ▶ 5V pin of the Arduino

GND ▶ Arduino GND pin

CS ▶ Digital Pin 10

RST ▶ Digital Pin 9

A0 ▶ Digital Pin 8

SDA ▶ Digital Pin 11

SCK ▶ Digital Pin 13

LED ▶ 3.3V pin of the Arduino

PIR sensor:

PIR sensor module has only 3 pins – one is Vcc which is a +5 volts input, a ground pin and finally the digital output pin. Connect +5V from Arduino to Vcc of PIR
sensor module, connect a GND from Arduino to ground of PIR sensor and finally connect the output pin (marked as 'out') to any digital pin of arduino.

Gas sensor:

In a Gas sensor we have 4 pins Vcc, GND, D0 and A0.

VCC supplies power for the module. we connect it to 5V output from our Arduino.

GND is the Ground Pin and needs to be connected to GND pin on the Arduino.

D0 provides a digital representation of the presence of combustible gases.

A0 provides analog output voltage in proportional to the concentration of smoke/gas.

Bluetooth module:

The Arduino is connected with Bluetooth module in following way:

RX (Pin 0) ▶ TX
TX (Pin 1) ▶ RX
5V ▶ VCC
GND ▶ GND

To combined all this connection with The Arduino we used a breadboard so that we can use same pin for multiple components.

VII. Data flow from sensors through ICs to I/O devices

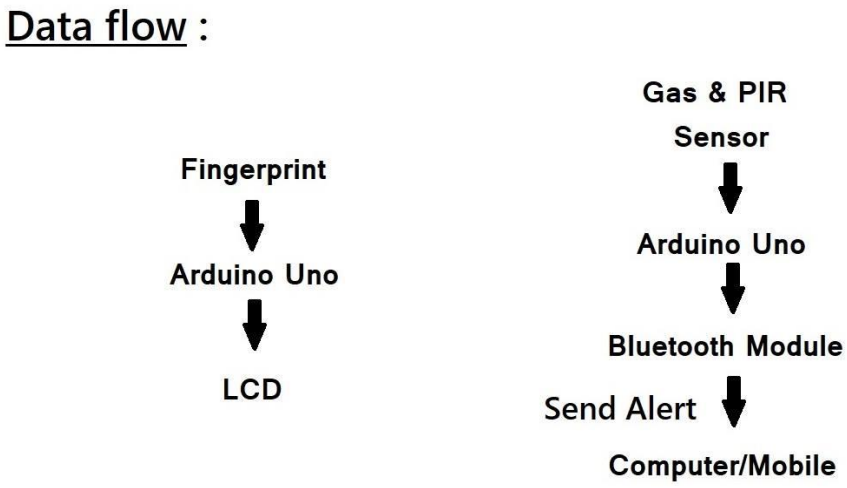

Fig2: Data flow diagram of the system

In Compact Office Security System, we are using Arduino as IC, LCD as GIPO device, camera and phone as I/O devices and finally PIR motion sensor, fingerprint scanner, smoke sensor as the required sensors. Initially, we will use Arduino to interface with all the other devices. For that a breadboard is used to connect all the devices using jump wires. In the Arduino Integrated Development Environment (IDE), we write the codes to command Arduino so that he can direct all the devices to perform specific tasks. For example, in fingerprint sensor, after pressing our finger in the scanner, the data is passed from finger print scanner to Arduino first then the result is shown in LCD. Here, LCD is connected with Arduino. In case of smoke/gas sensor and PIR motion sensor the data is first sent to the Arduino from which the Bluetooth module will take the information and will send to our mobile phone via Bluetooth. The surveillance camera is connected with Arduino which monitors the area 24/7 that we can view through our phone.

VIII. Code

In this part, we have mentioned the programming part for our project. The entire code will be sketched in Arduino.

//Gas Sensor Detection

```
#include <LiquidCrystal.h>
int sensorValue;
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
void setup(){  lcd.begin(16, 2);
Serial.begin(9600);                    // sets the serial port to 9600
 }
void loop(){sensorValue = analogRead(0);       // read analog input pin 0
Serial.print("AirQua=");
Serial.print(sensorValue, DEC);              // prints the value read
Serial.println(" PPM");
lcd.setCursor(0,0);
lcd.print("ArQ=");
lcd.print(sensorValue,DEC);
```

```
lcd.print(" PPM");
lcd.println("       ");
lcd.print("  ");
delay(100);                                // wait 100ms for next reading
}
```

//Fingerprint Sensor Detection

```
#include
#include
#include
#include
int getFingerprintIDez();
// pin #2 is IN from sensor (GREEN wire)
// pin #3 is OUT from arduino (YELOW wire)
SoftwareSerial mySerial(2, 3);
LiquidCrystal lcd(9, 8, 7, 6, 5, 4); // initialize the library with the numbers of the interface pins
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);
void setup()
{ Serial.begin(9600); // initialize the serial communications:
lcd.begin(16,2); lcd.setCursor(0,0); lcd.print("Scan your finger");
pinMode(13,OUTPUT);
pinMode(12,OUTPUT);
pinMode(11, OUTPUT);
pinMode(A0, INPUT);
finger.begin(57600); // set the data rate for the sensor serial port }
void loop() // run over and over again
{
getFingerprintID();
```

```
delay(100);
digitalWrite (13,HIGH);
}
uint8_t getFingerprintID()
{ uint8_t p = finger.getImage();
switch (p){
case FINGERPRINT_OK:
lcd.clear();
lcd.print(" Image taken... ");
delay(1000);
break;
case FINGERPRINT_NOFINGER:
return p;
case FINGERPRINT_PACKETRECIEVEERR:
return p;
case FINGERPRINT_IMAGEFAIL:
return p;
default:
return p; }
// OK success!
p = finger.image2Tz();
switch (p) {
case FINGERPRINT_OK:
break;
case FINGERPRINT_IMAGEMESS:
return p;
case FINGERPRINT_PACKETRECIEVEERR:
return p;
```

```
case FINGERPRINT_FEATUREFAIL:
return p;
case FINGERPRINT_INVALIDIMAGE:
return p;
default:
return p; }
// converted
p = finger.fingerFastSearch();
if (p == FINGERPRINT_OK){
lcd.clear();
lcd.print(" Found match! ");
digitalWrite(11, HIGH);
delay(1000);
digitalWrite(11,LOW); // turn on green LED to indicate match
}
else if(p == FINGERPRINT_NOTFOUND){
lcd.clear();
lcd.setCursor(0,0);
lcd.print(" Did not match! ");
delay(1000);
lcd.clear();
lcd.setCursor(0,0);
lcd.print(" scan finger! ");
return p;
}
else
{ return p; }
// IF FOUND A MATCH.....
```

```
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Found ID #");
lcd.print(finger.fingerID);
lcd.setCursor(0,1);
lcd.print("confidence ");
lcd.print(finger.confidence); }
// returns -1 if failed, otherwise returns ID #
int getFingerprintIDez() {
uint8_t p = finger.getImage();
if (p != FINGERPRINT_OK) return -1;
p = finger.image2Tz();
if (p != FINGERPRINT_OK) return -1;
p = finger.fingerFastSearch();
if (p != FINGERPRINT_OK) return -1;
// found a match!
digitalWrite(13, LOW);
delay(10);
digitalWrite(13, HIGH);
delay(10);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Found ID # ");
lcd.print(finger.fingerID);
lcd.setCursor(0, 1);
lcd.print("confidence ");
lcd.print(finger.confidence);
return finger.fingerID;
```

```
}
```

//PIR Sensor Detection

```
void setup() {
  pinMode(2, INPUT); //Pin 2 as INPUT
  pinMode(3, OUTPUT); //PIN 3 as OUTPUT
}
void loop() {
  if (digitalRead(2) == HIGH) // check if PIR is triggered.
  {
  digitalWrite(3, HIGH);   // turn the LED/Buzz ON
  delay(100);                  // wait for 100 msecond
  digitalWrite(3, LOW);   // turn the LED/Buzz OFF
  delay(100);                  // wait for 100 msecond
  }
}
```

//Bluetooth Sensor Detection

```
#include <SoftwareSerial.h>
SoftwareSerial EEBlue(10, 11); // RX | TX
void setup()
{
  Serial.begin(9600);
  EEBlue.begin(9600);  //Default Baud for comm, it may be different for your Module.
  Serial.println("The bluetooth gates are open.\n Connect to HC-05 from any other bluetooth device with 1234 as pairing key!.");
}
void loop()
{
```

```
  // Feed any data from bluetooth to Terminal.
  if (EEBlue.available())
    Serial.write(EEBlue.read());

  // Feed all data from termial to bluetooth
  if (Serial.available())
    EEBlue.write(Serial.read());
}
//Transferring data through Bluetooth in device
#include <SoftwareSerial.h>
SoftwareSerial EEBlue(10, 11); // RX | TX
void setup()
{
  Serial.begin(9600);
  EEBlue.begin(38400);  //Baud Rate for command Mode.
  Serial.println("Enter AT commands!");
}
void loop()
{
  // Feed any data from bluetooth to Terminal.
  if (EEBlue.available())
    Serial.write(EEBlue.read());
  // Feed all data from termial to bluetooth
  if (Serial.available())
    EEBlue.write(Serial.read());
}
```

IX. Estimated Cost Analysis

Here is the estimated costing for our projects.

Components	Quantity	Estimated Price (BDT)
Breadboard	1pc	65
Arduino UNO	1pc	399
PIR Motion Sensor	1pc	88
HC – 05 Bluetooth Serial Module	1pc	299
MQ-2 Flammable Gas & Smoke Sensor	1pc	120
Fingerprint Scanner - TTL (GT-511C3)	1pc	4800
470 Ohm Resistor	1pc	5
1K Ohm Resistor	1pc	5
Jumper Wire	1 set	40
Other Accessories	---	500
Total Cost		6255/-

(Source: https://store.roboticsbd.com)

X. Conclusion

Nowadays in the era of technology, we cannot hide anything from the entire world. There is always a presence of security risk around us. For every denial of attack, we need always to be conscious one step ahead. Ensuring security is not optional anymore, it is necessity. Intruder in never welcome in a place like office where security should be strictly maintained. Now we all want such security packages where it includes almost everything to protect your office from some uninvited person. In our project, we also tried to come up with such security package solution at a convenient and cheaper price where it will ensure security as well cost effective. We want to believe that, our project will do its assigned tasks successfully and bring an advancement towards a secured world.

XI. Reference

- https://en.wikipedia.org/wiki/Arduino
- https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/how-pirs-work
- https://www.ifsecglobal.com/smoke-detectors/
- https://www.arrow.com/en/research-and-events/articles/how-fingerprint-sensors-work
- https://create.arduino.cc/projecthub/Guitarman1/displaying-sensor-values-on-lcd-c0c44f
- https://en.wikipedia.org/wiki/Motion_detector