Embedded System Design

Unit code: ENEX20001

Term 2, 2019

Portfolio Group Report

Group Name: ROK_1

Group Members: 1. Anand Sebastian 12124306

2. Sandesh Chhetri 12122731

Unit Coordinator: Preethi Preethichandra

Week of Submission: Week 12

Date of Submission: 11/10/2019

School of Engineering and Technology

Central Queensland University

Contents

1.0 Executive Summary

The main aim of the project is to build a prototype house that has home automation facilities. The automation facilities include temperature measure and control, pin protected door, automatic lighting and humidity measure, and control. The house we are trying to build is aimed to sustain the weather in Rockhampton city and provide facilities to the user accordingly. By taking account of the problem of intense heat and humidity in Rockhampton during the summer season, we came with the idea to build a prototype house that has temperature control and passive humidity control facilities. For temperature measurement we are using LM35 sensor and a DC fan to control the temperature, and for humidity measurement we are using DHT22 sensor and for humidity control we are using passive humidity control measurements, because using air conditioner or air dehumidifier in our prototype model was not feasible, so we came up with the solution to use passive ventilation to remove internal moisture by simply designing windows in specific places in the prototype house so that there is a good cross airflow. Similarly, for automatic lighting, we are using PIR (Passive Infrared Sensor) to detect movements and switch light accordingly. Our project requires us to use PIC18F4321 microcontroller interfaced with the sensors that include LM35, DHT22, PIR along with actuators like servo motor, DC fan, relay module and other peripherals like keypad. In addition, the project was done to showcase our skill of C programming and project management skills.

2.0 Introduction

Home automation refers to the automatic and electronic control of various features, activities and electronic appliances in the house. In simple words, it is the network of different sensors, actuators, and appliances inside the home so you can easily control the utilities and features of your home via use of remote control, smartphone, or computer or through internet or some communication protocols. So that means you don't have to physically be present to do activities like turning on the lights, locking the door or turn down the temperature, etc. There are various features that can be included in home automation systems like temperature control, lighting control, automatic door lock and keyless entry, alarm systems, etc. (Xfinity, 2019)

The main purpose of home automation is to provide facilities to the people through remote access to different appliances and services to provide comfort and convenience.

There are three main elements in achieving home automation and they are sensors, controllers and actuators. The sensors collect the environmental conditions and the actuators are used to change those environmental conditions with the help of controllers. For example, a temperature sensor collects temperature and with the help of microcontroller we can process those values and programs accordingly to run fan to change the temperature conditions. In this case the microcontroller is the controller and fan are the actuator.

3.0 Project Scope

3.1 Aim

Taking account of the problem of intense heat and humidity in Rockhampton during the summer season, the project aim is to study and implement home automation facilities in the city of Rockhampton that includes temperature control and passive humidity control facilities along with automatic lighting and pin protected security systems. The benefits of this project, home automation, can be described in a few categories including savings, safety, convenience, and control.

- Savings: Automatic temperature control and lighting control ensure the fans and lights are on when they are necessary, hence the usage in electricity can be controlled.

- Safety: Pin protected door ensures intruders from using fake keys, and hence maximum security is ensured.

- Convenience and control: The most important aspect of home automation is convenience and control, it provides users with a convenient method to control the facilities available inside their home.

3.2 Requirements

The project requirements can be divided into two parts hardware requirements and software requirements.

3.2.1 Hardware Requirements

The list of hardware requirements for our project are as follows:

- CQU PIC development board (PIC 18F4321 microcontroller)
- CQU sensor/actuator board
- Connecting wires
- Micro Servo SG90
- PIR Sensor
- 5V Relay Module

- LM35 Temperature Sensor
- DHT22 Digital Humidity Sensor
- LEDs
- DC Fan
- LCD Module
- 3D Printer
- 4x3 Matrix-Membrane Type Keypad

3.2.2 Software Requirements

The list of software requirements for our project are as follows:

- MPLAB X IDE
- Autodesk Tinker CAD
- C programming
- Draw.io

3.3 Design Objectives

Considering the project aim the present design objective is to create a small prototype house model that has some components of home automation facilities using the PIC microcontroller 18F4321 interfaced with different sensors and actuators to provide temperature measurement and control, automatic lighting, pin protected door and humidity measurement and control.

3.4 Constraints and Limitations

Even though there were some project constraints and limitations, we were able to achieve our project design objectives within the project's proposed lifetime. One of the major project constraints was in using DHT22 to measure humidity. As most of the sensors and actuators were controlled by one of the PIC development board with internal oscillator running on 1MHz, so essential timing and delays for most of the sensors were provided accordingly taking account of 1Mhz internal oscillator, but for DHT22 humidity sensor we had to use internal oscillator at 4MHz, thus we had to use another PIC board for smooth operation of other sensors and actuators and DHT22 separately.

3.5 Inclusions

The feasibility of including new sensors which was not included in the initial project plan was discussed during the project meetings. We have decided to use Digital Humidity sensor, DHT22 as part of our project. This decision was made due to the sole reason of the precision of DHT22 in measuring the Relative Humidity. In order to use DHT22, we used the 4MHz oscillator on the MCU, which helps to send adequate signal to the DHT22 from MCU and read the data from the sensor.

3.6 Exclusions

In the project, we have excluded three components. The mini humidifier, heating pad and analogue Humidity sensor.

3.6.1 Mini Humidifier

The scope of humidifier is to provide the adequate humidity, whenever there is any shortage of humidity. We have decided to avoid the usage of humidifier in our project. The reason is that, the addition of moisture inside the prototype will risk the working of all the electronic components. Due to this reason, we have decided to add humidity manually without the use of humidifier

3.6.2 Analogue Humidity Sensor

The inclusion of digital Humidity sensor, DHT22 in our project leads to the exclusion of analogue Humidity sensor. The higher accuracy of digital sensor compared to the analogue sensor favours the exclusion of analogue humidity sensor.

3.6.3 Heating Pad

The scope of heating pad in the project was to increase the temperature whenever the temperature falls below a pre-set value. The heating pad was excluded from the project, as it draws a current of 0.6A to 1A. This will overload the power supply. So, we have decided to exclude the heating pad.

3.7 Deliverables

The Home Automation Project which we have designed is expected to deliver the following outcome.

- The password entry through keypad which facilitate the secure entry through front door.
- The closure of door after the visitor passes through the alleyway.
- Display of the Relative Humidity on LCD.
- Display of Temperature on 7 segment display.
- The fan turns ON when the temperature exceeds the pre-set value.

4.0 Project Management

Project management is the practice of initiating, planning, executing, controlling and closing the work of a team to achieve required goals at a specific time period. It is the application of processes, skills, methods and knowledge and experience to achieve project objectives (Apm.org.uk, 2019). The project management in our project includes planning, executing, documentation of hardware and software.

4.1 Methodology

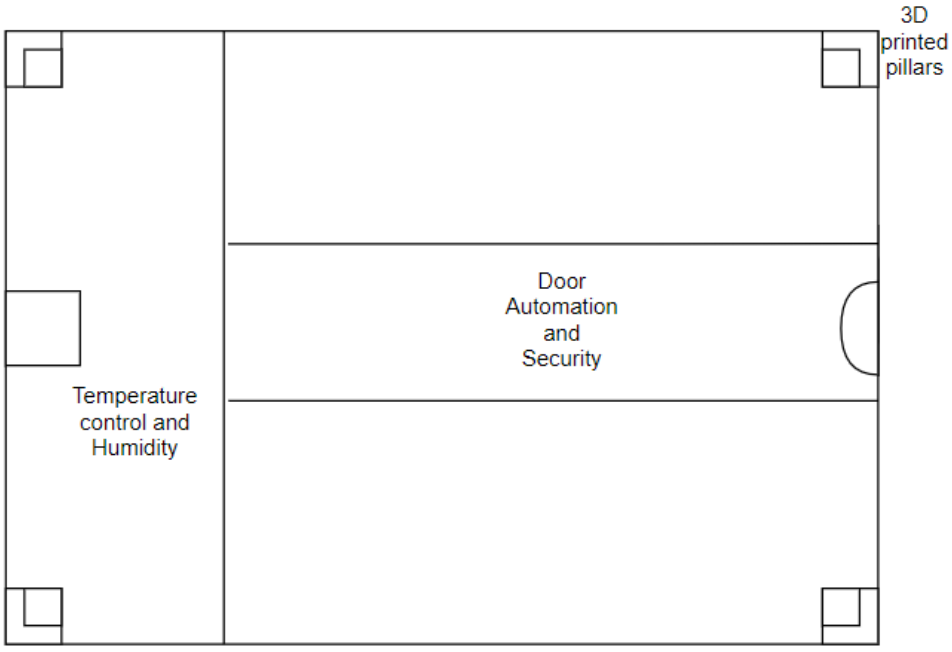


Figure 1 Project Prototype

The project is divided into two parts Door Automation and Security and Temperature Control and Humidity display. The Door Automation and Security includes a servo-controlled door that has security and the door is automated in a sense that it closes automatically as a PIR sensor detects a movement. The Temperature control and Humidity display portion are controlled by another microcontroller where the LM35 temperature sensor is used to measure temperature and a DC motor fan is used to control temperature accordingly. Similarly, a DHT22 humidity/temperature sensor is used to measure humidity.

4.1.1 Door Automation and Security

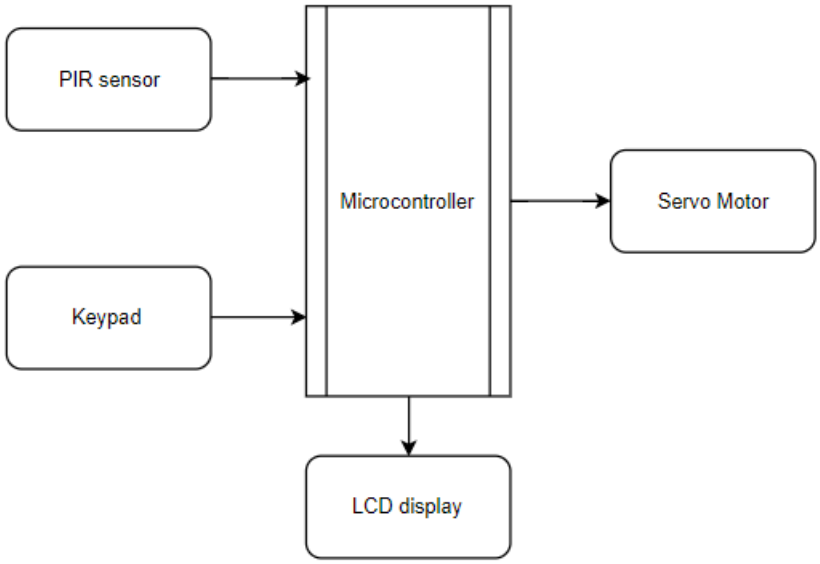


Figure 2 Door Automation and Security

As the user enters the security key through the keypad, if the entered password is correct the servo motor actuates and opens the main door. And, as the person enters through the main door to the alleyway the PIR sensor detects the movement and actuates the servo motor again to close the door.

Keypad (4x3 matrix membrane-type keypad)

4x3 matrix type keypad is used to enter 6 digit pin code, the user is prompted with the information in the LCD display to enter the 6 digit code and as the user enters pin code, if the code is correct the LCD displays with the "Welcome " message and the servo actuates otherwise the LCD displays with "Incorrect PWD" message.

Servo Motor (SG90)

The servo motor is used for its precision and high efficiency. Since we only require certain rotation of door i.e. clockwise and anti-clockwise 90-degree rotation to open and close it, we decided to use servo motor (SG90). As the user enters with the correct pin code the servo motor actuates and opens the door.

PIR Sensor (HC-SR501/ XC4444)

The PIR sensor (Passive Infra Rodration) is used to detect movements as a person enters through the main door, as it detects some movement servo motor is actuated and the main door is closed automatically.

4.1.2 Temperature Control and Humidity

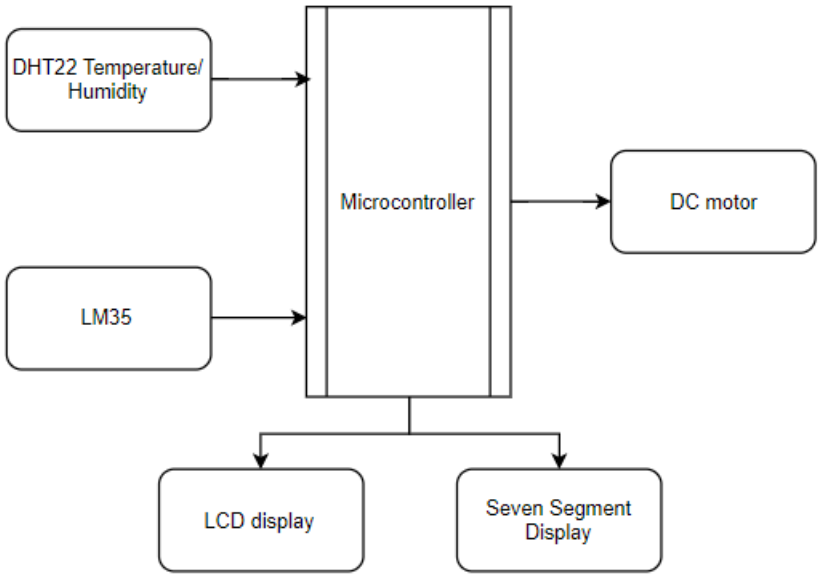


Figure 3 Temperature Control and Humidity

The DHT22 and LM35 are used to measure the humidity and temperature respectively, and LCD display and Seven segment displays are used to display those values. A DC motor fan is used for temperature control.

DHT22 Temperature/Humidity

DHT22 Temperature/Humidity sensor is used to measure humidity and temperature. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air and spits out a digital signal on the data pin. The DHT22 was chosen because it was already available in our CQU sensor board when the switch 2 in the CQU microcontroller board is pressed the relative humidity value is displayed in the LCD display.

LM35

When Switch 3 in the CQU microcontroller board is pressed the LM35 analogue temperature sensor displays with the temperature value in the seven-segment display of the CQU microcontroller board. In the microcontroller, there is a pre-set value of temperature, and if the temperature from LM35 exceeds the pre-set temperature value the DC motor fan actuates. to control the temperature.

DC Motor Fan

The DC motor fan only runs when the house temperature goes beyond the pre-set values in the microcontroller program.

4.2 Project Flow Diagram

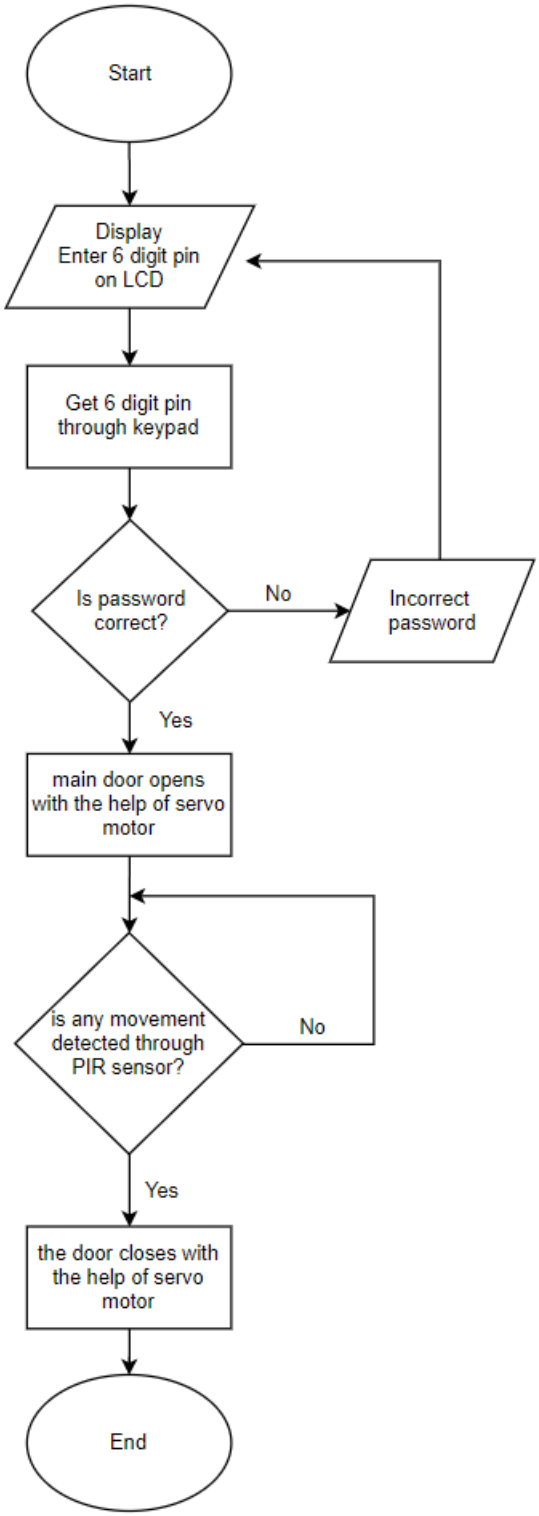


Figure 4 Flow Diagram Door Automation and Security

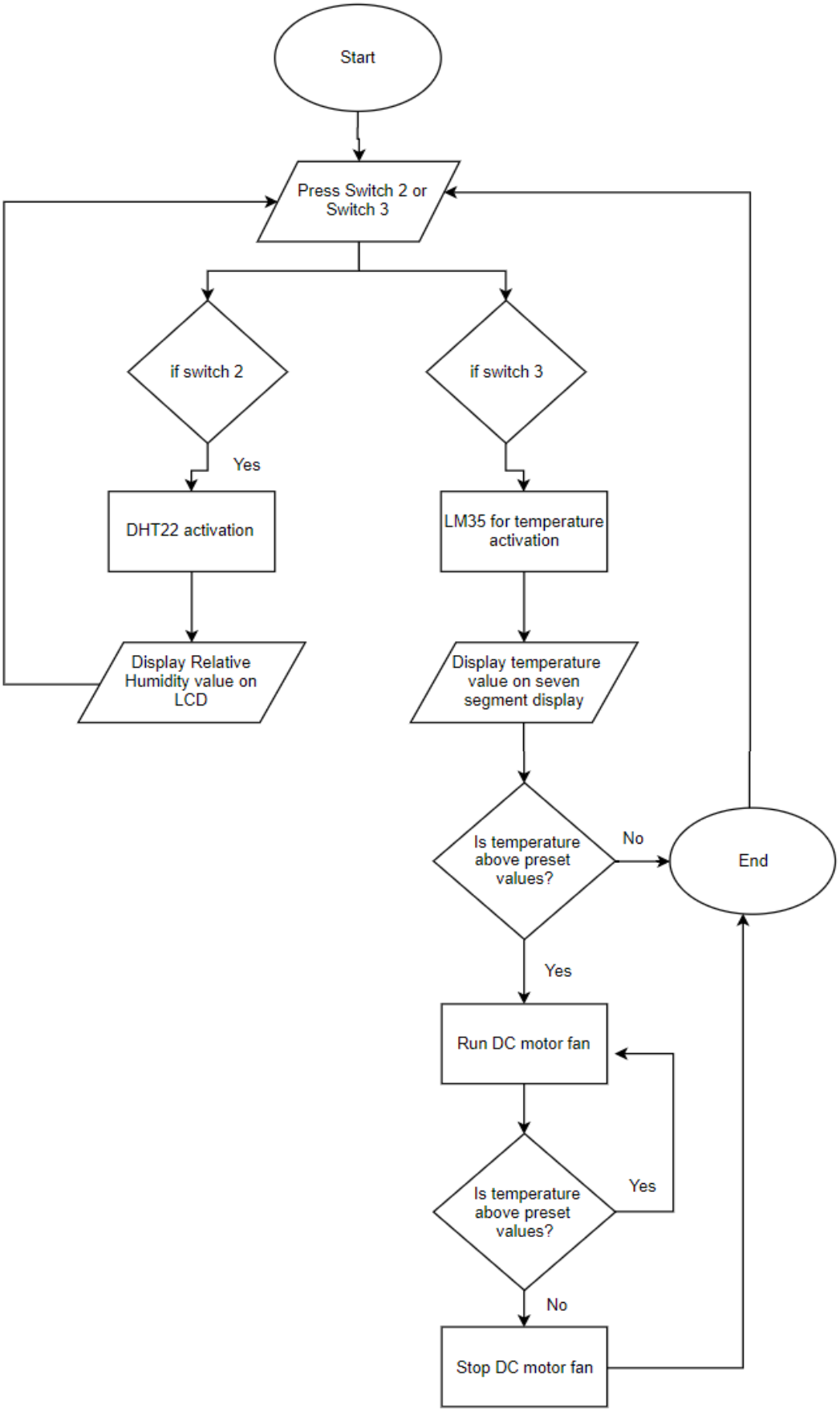


Figure 5 Flow Diagram for Temperature Control and Humidity

4.3 Risk Assessment

Risks	Potential Damage	Risk Ranking
Incorrect Power Supply to PIC board	Damage to CQU microcontroller and sensor board	Medium
Incorrect Pin Connection	Incorrect values from sensors/actuators	Low
Mishandling of physical prototype	Damage to house prototype	High
Selecting wrong microcontroller programmer through MPLAB X IDE	Damage to CQU Pic Board	High

4.4 Gantt Chart

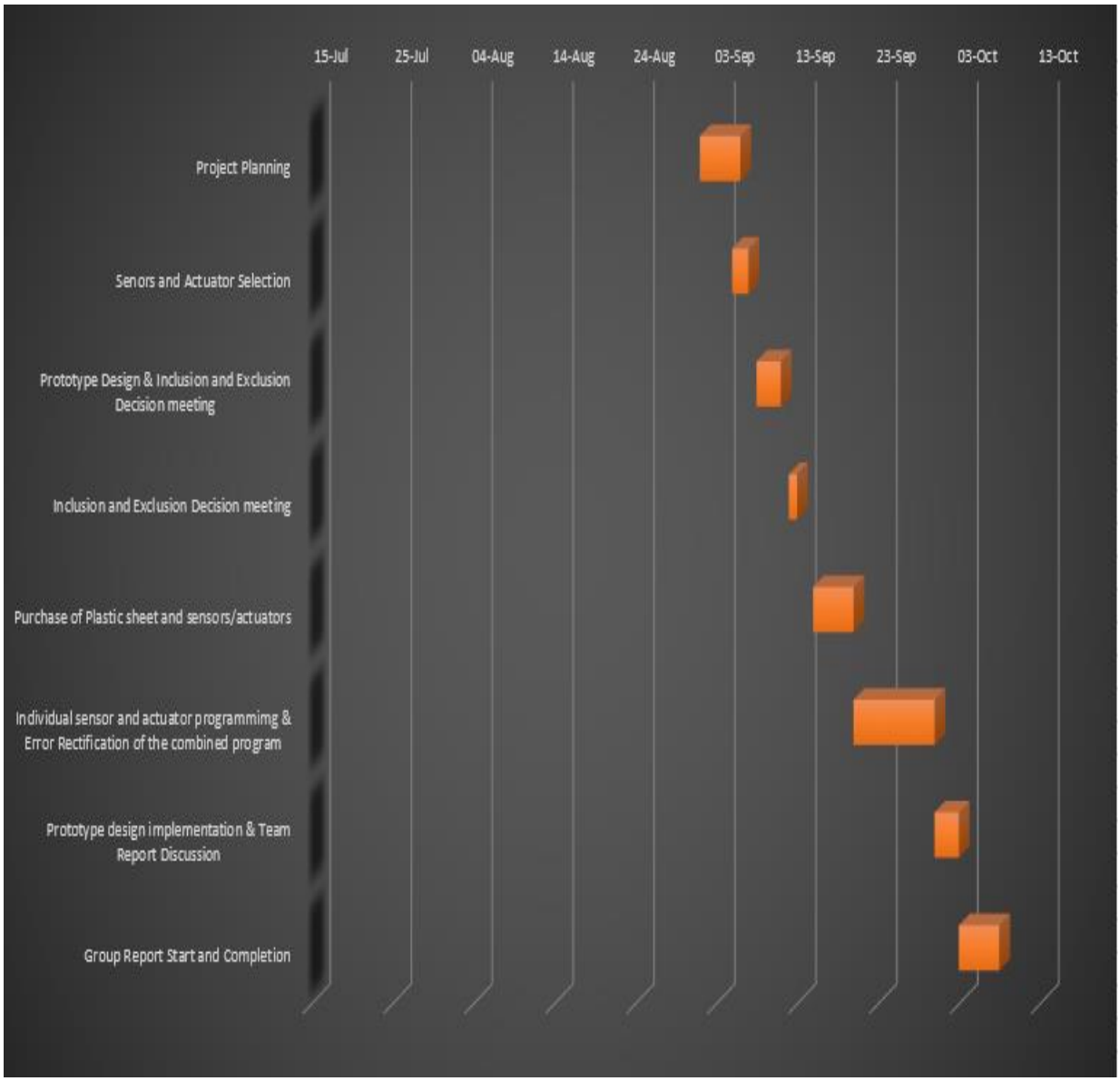


Figure 6 Gantt Chart

5.0 Project Development

The Home Automation project is developed by step by step manner and this helped us to maintain the time frame within which we have completed the project. The main step is the research about the relevant information available to develop the project.

5.1 Research

The idea of setting up our project came from the home automation products of different companies like Philips (www2.meethue.com, 2019), Chubb Home Security by United Technologies (Chubbhomesecurity.com.au, 2019), Samsung and many more. The basic idea is of Home Automation and Security project is secure collection of parameters which is to be altered according to the convenience of human and provide secure entry to the front door of the home using password.

The essential part of this project is the microcontroller and we have used PIC18F4321 microcontroller. The main advantage of this microcontroller is that it can provide a range of control features, such as a real-time clock, motor control and power supply, counters and capture/compare. It also has built in A/D converters, comparators, D/A converters and op amps. In comparison to other microcontrollers, the PIC microcontroller is much more reliable, as it is less likely to malfunction when built into a device. It also offers a powerful performance thanks to the use of RISC architecture. (Delmar, 2019)

In accordance to the research we found out that PIC18F4321 is the exact tool which acts as the brain of this project.

Figure 7 CQU PIC18F4321 Development Board

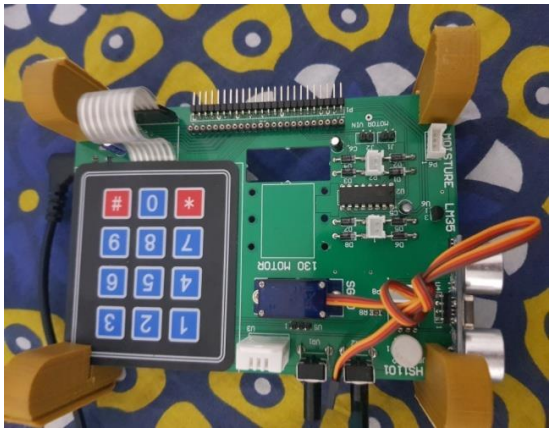


Figure 8 CQU Sensor Board

The PIC18f4321 incorporate a range of features that can significantly reduce power consumption during operation. The main reason behind which we have this microcontroller is its high computational performance at an economical price – with the addition of high endurance, and enhanced flash program memory.

The features of PIC18F4321 is given below (Microchip.com, 2019):

Operating Frequency	DC – 40 MHz
Program Memory (Bytes)	8192
Program Memory (Instructions	4096
Data Memory (Bytes)	512
I/O Ports	Ports A, B, C, D, E
Timers	4
Serial Communications	MSSP, Enhanced USART
10-bit Analog-to-Digital Module	13 Input Channels

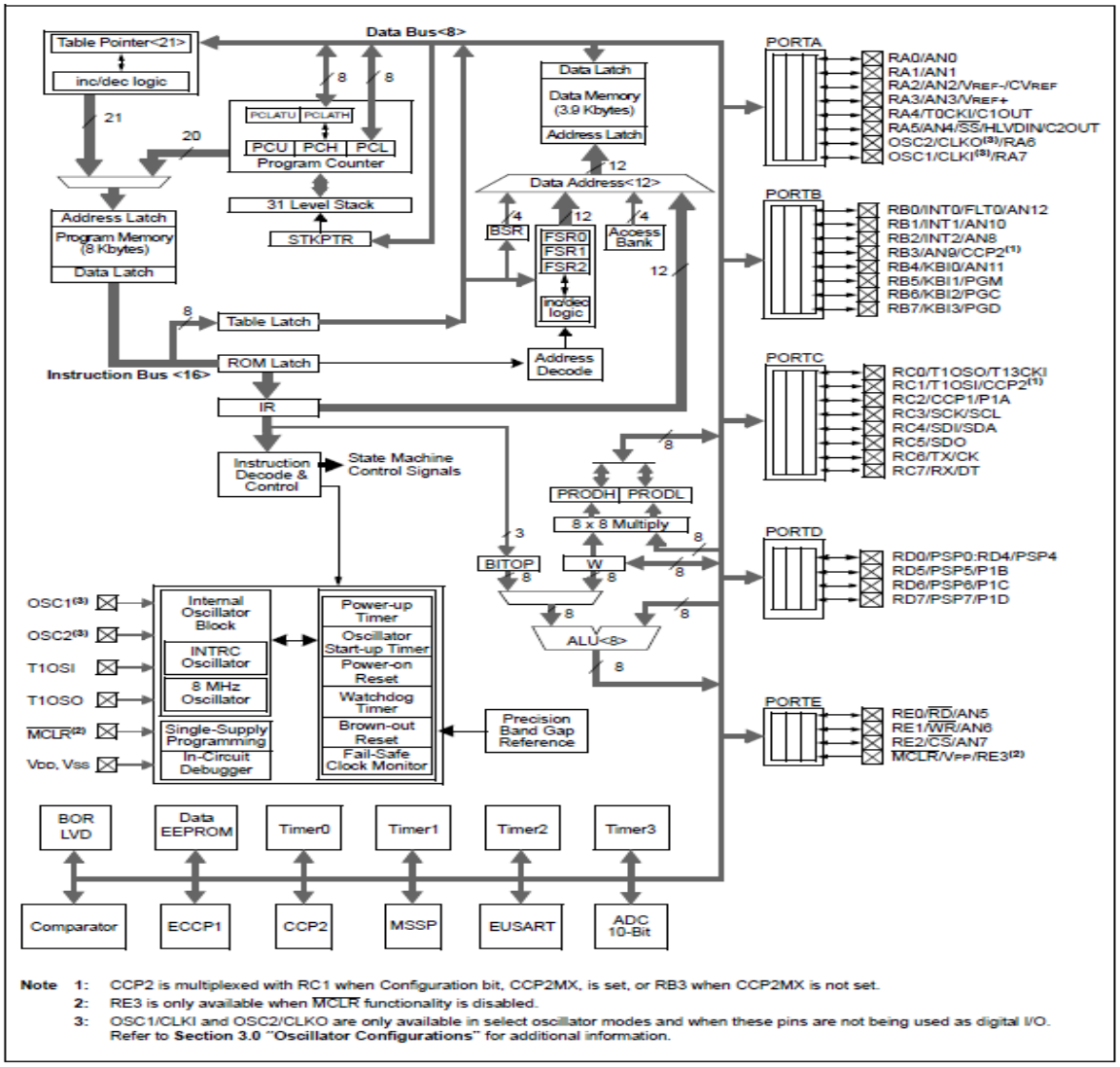


Figure 9 PIC18F4321 Diagram

5.2 Sensor and Actuator Selection

The three essential parts of a home automation system: sensors, controllers, and actuators.

- Sensors can monitor changes in temperature and motion detection. Home automation systems can then adjust those settings to your preferences. We planned to use Humidity sensor, Temperature sensor, Keypad and PIR sensor.
- Controllers are microcontrollers used to send and receive messages about the status of automated features in your home and control the actuators. We have planned to use PIC18F4321.

- Actuator includes the servo that controls the actual mechanism of a door and DC motor that start rotating when the temperature exceeds the preset value. They are programmed to be activated by a remote command from a controller. We have used 5v motor and servo motor.

5.2.1 Humidity Sensor

The humidity sensor DHT22 is used to get the relative humidity of the room. DHT22 output calibrated digital signal. It uses exclusive digital-signal-collecting-technique and humidity sensing technology, assuring its reliability and stability.

Model	DHT22/AM2302
Power supply	3.3-5.5V DC
Output signal	digital signal via 1-wire bus
Sensing element	Polymer humidity capacitor
Operating range	humidity 0-100%RH
Accuracy	humidity +-2%RH

Availability

1.Jaycar : https://www.jaycar.com.au/humidity-temperature-sensor-module-for-arduino/p/XC4246

2.ebay : https://www.ebay.com.au/i/201355561311?chn=ps&norover=1&mkevt=1&mkrid=705-139619-5960-

0&mkcid=2&itemid=201355561311&targetid=467329303426&device=c&mktype=pla&googleloc=9069261&poi=&campaignid=6472420988&mkgroupid=77077696065&rlsatarget=pla-467329303426&abcId=1141706&merchantid=116292819&gclid=Cj0KCQjwoebsBRCHARIsAC3JP0KjNG2zpCaoZZIIWvgW7oYrCFiEZR2qsKSwiJ6FoLnxQg8rmFPlvRcaAkgQEALw_wcB

3. CQU Sensor Board

The humidity sensor DHT22 was already available on the sensor board of CQU and we decided to use it instead of analogue humidity sensor. Analog signals are much affected by external noise and create errors in the output signal. But digital signals are susceptible to noisy environments and hence digital sensor is preferred over analogue ones.

Operation

The DHT22 is powered and do not send any signal to the sensor within one second initially. After that a start signal is send to the sensor by MCU through the DATA pin. This will change the mode of sensor from low-power-consumption-mode to running-mode. After receiving the start signal the DHT22 will send a response of 40-bit data that contains the information of Relative Humidity and Temperature.

One start signal will give one response data that reflect the relative humidity and temperature information from DHT22. DHT22 will change to low-power-consumption-mode when data collecting finish if it does not receive start signal from MCU again. (Sparkfun.com, 2019)

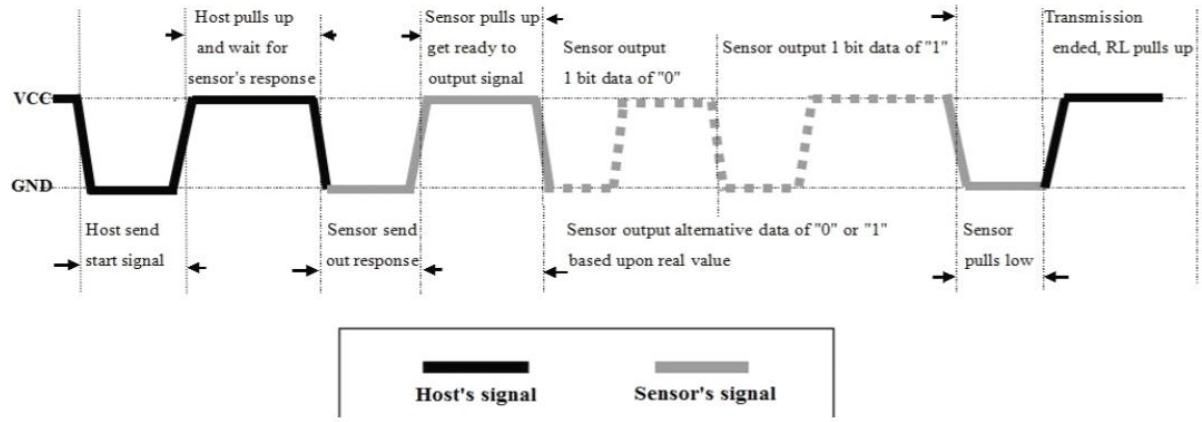


Figure 10 Signal and DATA transmission between MCU and DHT22

Calculation:

DATA=16 bits Relative Humidity data+16 bits Temperature data+8 bits check-sum

Example: MCU has received 40-bit data from DHT22 as

0000 0010 1000 1100 0000 0001 0101 1111 1110 1110

16 bits RH data 16 bits T data check sum

Here we convert 16 bits RH data from binary system to decimal system,

0000 0010 1000 1100 → 652

Binary system Decimal system

RH=652/10=65.2%

Here we convert 16 bits T data from binary system to decimal system,

0000 0001 0101 1111 → 351

Binary system Decimal system

T=351/10 =35.1 T = 35.1℃

When highest bit of temperature is 1, it means the temperature is below 0 degree Celsius. Example: 1000 0000 0110 0101, T= minus 10.1℃

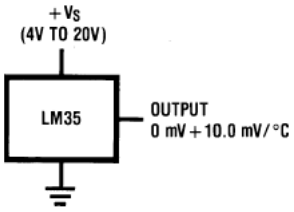
16 bits T data

Sum=0000 0010+1000 1100+0000 0001+0101 1111=1110 1110 Check-sum

Check-sum=the last 8 bits of Sum=1110 1110

5.2.2 Temperature Sensor – LM35

The LM35 is precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius temperature. The LM35 was already available on the CQU sensor board.



Features:

- Calibrated Directly in Celsius (Centigrade)
- Linear + 10-mV/°C Scale Factor
- 0.5°C Ensured Accuracy (at 25°C)
- Rated for Full −55°C to 150°C Range
- Suitable for Remote Applications •
- Low-Cost Due to Wafer-Level Trimming
- Operates From 4 V to 30 V
- Less Than 60-µA Current Drain
- Low Self-Heating, 0.08°C in Still Air

Advantages of LM35 over other Sensors:

The LM35 device has an advantage over linear temperature sensors calibrated in Kelvin, as the user is not required to subtract a large constant voltage from the output to obtain convenient Centigrade scaling. The LM35 device does not require any external calibration or trimming to provide typical accuracies of ±¼°C at room temperature and ±¾°C over a full −55°C to 150°C temperature range. The low-output impedance, linear output, and precise inherent calibration of the LM35 device makes interfacing to readout or control circuitry especially easy. As the LM35 device draws only 60 µA from the supply, it has very low self-heating of less than 0.1°C in still air. (TEXAS INSTRUMENT, 2019)

Availability

1. core electronics : https://core-electronics.com.au/lm35-analog-linear-temperature-sensor.html

2. CQU Sensor Board

5.2.3 Servo Motor- SG90

SG90 is a tiny and lightweight servo with high output power. Servo can rotate approximately 180 degrees (90 in each direction). We used servo motor for the door opening and closing.

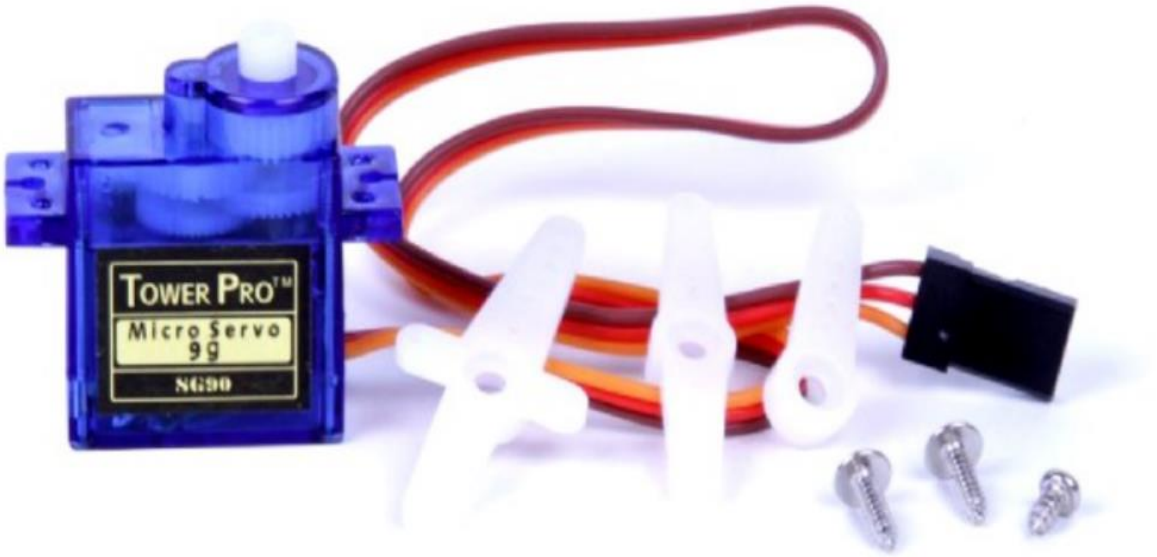


Figure 11 SG90 Servo Motor

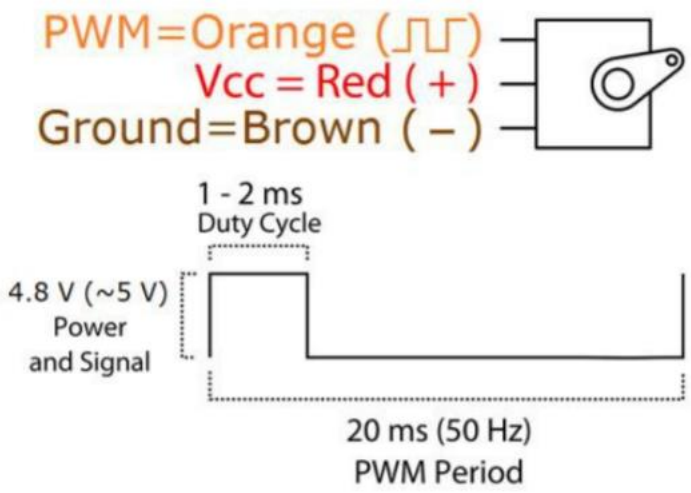
Availability

1. CQU Sensor Board

2 Jaycar : https://www.jaycar.com.au/arduino-compatible-9g-micro-servo-motor/p/YM2758

We have purchased the servo from jaycar as we need to fix the servo inside the prototype.

Operation



The sg90 has 3 pins: Orange is the pin through which the signal is sent, Red is the pin for supply and Brown is for Ground. To get the required position of the servo motor, the pulse width of the signal is the essential thing.

Middle position (0 degree) : 1.5 ms pulse

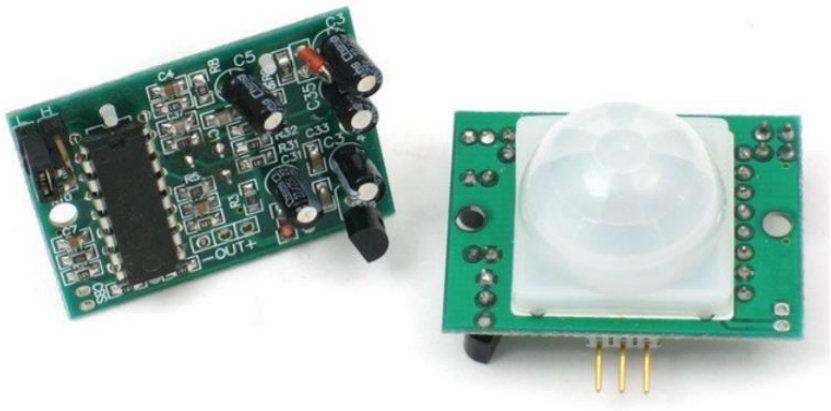
Leftmost position (-90 degree) : 1 ms pulse

Rightmost position (90 degree) : 2ms pulse

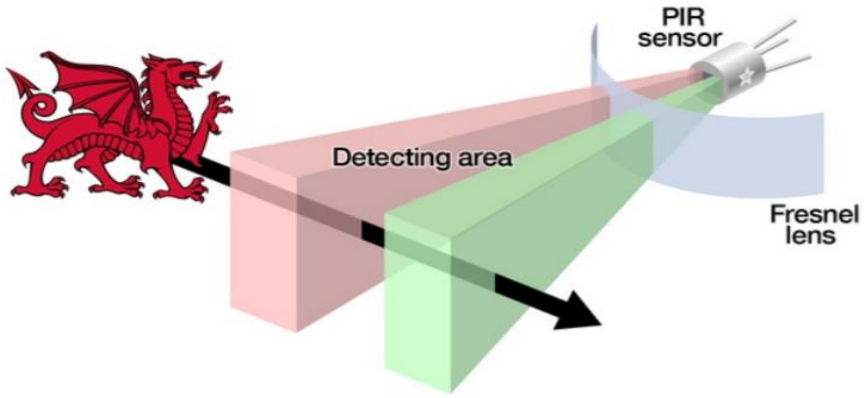
The total pulse width should be 20 ms of time period.

5.2.4 PIR Sensor

PIR is short for Passive Infra Rodration. PIR sensors allow you to sense motion. It is used to detect whether a human has moved in or out of the particular area. They are small, inexpensive, low-power, easy to use. So, they are commonly found in appliances and gadgets used in homes or businesses.



PIRs consists of a pyro-electric sensor which can detect levels of infrared radiation. Every object emits some amount of radiation. 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.



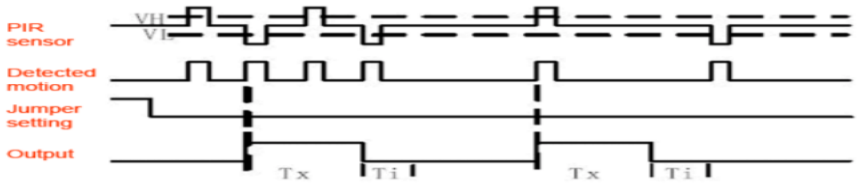
The two halves are wired up so that they cancel each other out. If one half sees more or less IR radiation than the other, the output will swing high or low. Note that PIRs won't tell you how many people are around or how close they are to the sensor. (Adafruit Learning System, 2019)

Specification

Delay Time	0.3s - 18s
Logic Voltage	3.3VDC(5V Tolerant)
Operating Voltage%%	5VDC - 20VDC
Dimensions	32(L) x 24(W) x 25(H)
Additional Features	Adjustable sensistivity & delay time
Sensitivity Distance	3~4m and 5~7m
Sensitivity Angle	100 degrees

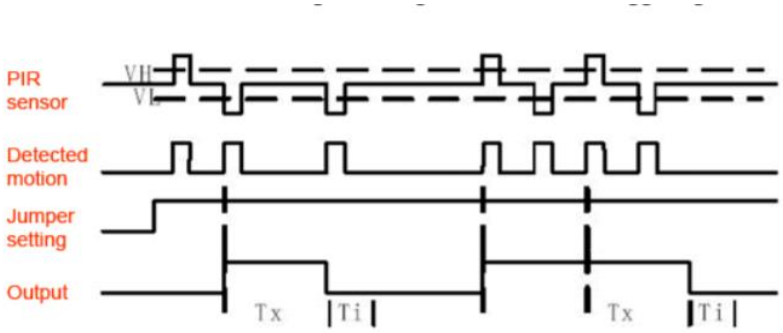
Retriggering

L position



The LED does not stay on when moving in front of it but actually turns on and off every second or so. That is called "non-retriggering".

H position



The LED does stay on the entire time when something is moving. That is called "retriggering.

Availability

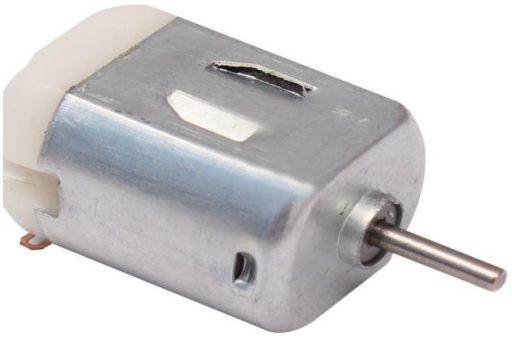
1. Jaycar : https://www.jaycar.com.au/arduino-compatible-pir-motion-detector-module/p/XC4444

2. Amazon : https://www.amazon.com/Envistia-HC-SR501-Passive-Infrared-Detector/dp/B01N7E4UA0/ref=alp_dpwidget_a_w_?th=1&psc=1&smid=A1CV2ETGSPQEB3

We have selected the jaycar store as it is near to our location and amazon delivery is not fast.

5.2.5 DC Motor- 5V

The 5v DC motor is used for the control of temperature and it turns ON when the temperature exceeds a particular value.



AVAILABILITY

1. Jaycar : https://www.jaycar.com.au/hobby-motor-medium-torque/p/YM2707

2. CQU Sensor Board

We have selected the motor from CQU Sensor Board, since it is more reliable and to cut short the budget we have eliminated the Jaycar product.

5.2.6 LCD 16X2

The Liquid Crystal Display can display 2-lines X 16-characters. It has 5 x 8 dots with cursor each character and commonly-used HD44780 controller is built in this 1602 LCD module.

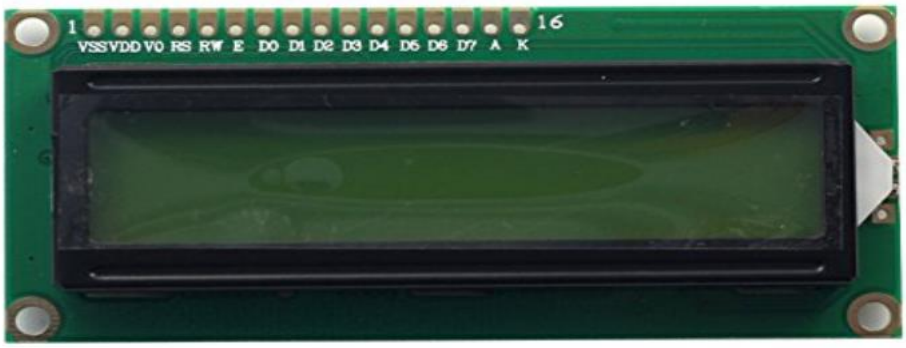


Figure 12 16x2 LCD Display

Availability

1. Amazon : https://www.amazon.com/Arducam-Display-Controller-Character-Backlight/dp/B01N5I8KCS/ref=sr_1_17?keywords=lcd+16&qid=1570430095&s=electronics&sr=1-17

2. Jaycar : https://www.jaycar.com.au/dot-matrix-white-on-blue-lcd-16x2-character/p/QP5521

3. CQU Sensor Board

We have selected to use the CQU Sensor Board LCD as it was the more cost effective way to reduce the budget.

5.2.7 Segment Display

A seven-segment display is a form of electronic display device for displaying decimal numerals. We used the 7-segment display to show the temperature measured by the LM35. It consists of 8 LEDs connected in parallel that can be lit in different combinations to display the numbers. Each segment (LED) is denoted by letters A to G.

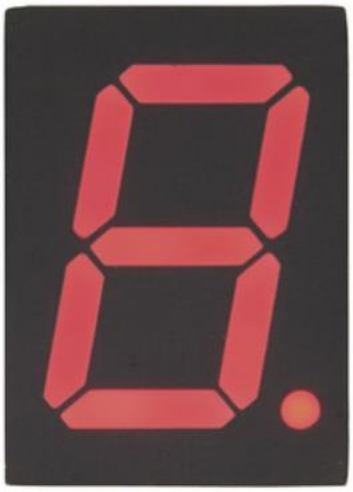


Figure 13 7 Segment Display

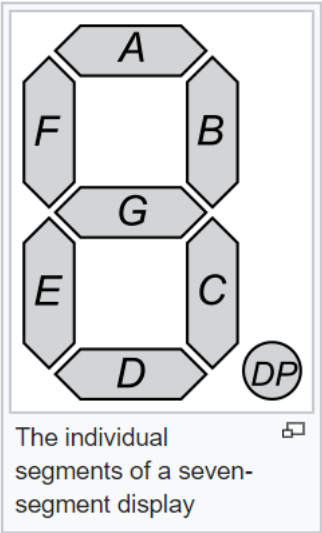


Figure 14 Seven Segment Display Diode Position

Availability

1. Jaycar : https://www.jaycar.com.au/fnd507-lts542r-s50-rwb-common-anode-7-segment-display/p/ZD1857

2. CQU Sensor Board

We have selected the 7-segment display on the CQU Sensor Board as it is suitable to display a 2-digit temperature and to reduce the budget.

OPERATION

In order to display a particular digit, we should give appropriate data to the data bus.

Digit	Display	gfedcba	abcdefg	a	b	c	d	e	f	g
0	0	0x3F	0x7E	on	on	on	on	on	on	off
1	1	0x06	0x30	off	on	on	off	off	off	off
2	2	0x5B	0x6D	on	on	off	on	on	off	on
3	3	0x4F	0x79	on	on	on	on	off	off	on
4	4	0x66	0x33	off	on	on	off	off	on	on
5	5	0x6D	0x5B	on	off	on	on	off	on	on
6	6	0x7D	0x5F	on	off	on	on	on	on	on
7	7	0x07	0x70	on	on	on	off	off	off	off
8	8	0x7F	0x7F	on	on	on	on	on	on	on
9	9	0x6F	0x7B	on	on	on	on	off	on	on
A	A	0x77	0x77	on	on	on	off	on	on	on
b	b	0x7C	0x1F	off	off	on	on	on	on	on
C	C	0x39	0x4E	on	off	off	on	on	on	off
d	d	0x5E	0x3D	off	on	on	on	on	off	on
E	E	0x79	0x4F	on	off	off	on	on	on	on
F	F	0x71	0x47	on	off	off	off	on	on	on

Figure 14 Hexadecimal encodings for displaying the digits 0 to F

5.2.8 Keypad

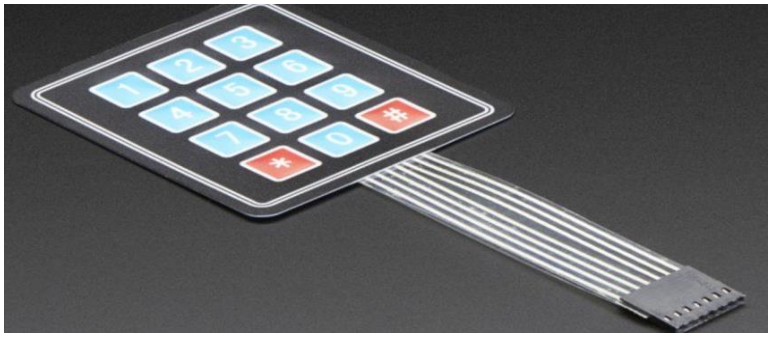


Figure 15 Keypad

A membrane keypad is a matrix consisting of rows and columns. Each key is assigned to a certain row and column. This keypad has 12 buttons, arranged in a telephone-line 3x4 grid. It's made of a thin, flexible membrane material with an adhesive backing (just remove the paper) so you can attach it to nearly anything. The keys are connected into a matrix, so you only need 7 microcontroller pins (3-columns and 4-rows) to scan through the pad.

Operation

On a 12-button keypad you have 4 rows and 3 columns. Set the rows as high voltage and scan the columns for the appropriate pressing of the key or vice versa.

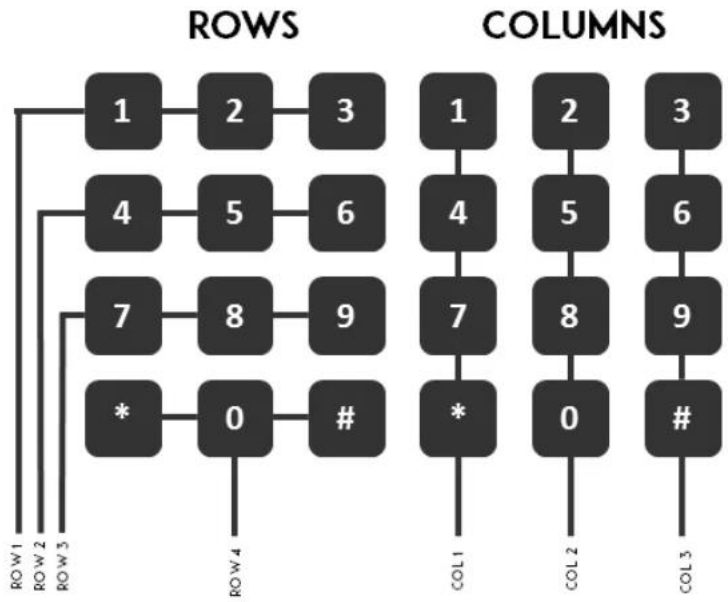


Figure 16 Keypad Rows and Columns

Availability

1. core electronics : https://core-electronics.com.au/membrane-3x4-matrix-keypad-3x4.html?utm_source=google_shopping&gclid=CjwKCAjwxOvsBRAjEiwAuY7L8jQ4_FNYVkrci1W_vMYU0f4Z1wV9gviKFESFwZJ5sg_4L4pcr6BgqRoCle4QAvD_BwE

2. CQU Sensor Board or from Lab.

We have selected the 3x4 keypad obtained from the lab.

5.3 Programming Individual Sensors and Actuators

The sensors and actuators are individually programmed before putting it all together. We have divided the whole workload into 4 parts:

1. The password entry through keypad and display it into the LCD screen and servo motion

2. The PIR sensor senses the motion and blink an LED

3. DHT22 sensor for displaying relative humidity on LCD

4. Display the temperature measured by LM35 to 7 segment display and run the motor

5.3.1 Password on LCD Through Keypad and Servo Motion

```
// PASSWORD IS 123123
#include <stdio.h>
#include <stdlib.h>
#include<string.h>
#include <p18f4321.h>
#include <delays.h>
#include<CQU_LCD_Ver_3.h>

#define SERVO PORTAbits.RA1
# pragma config WDT = OFF
# pragma config LVP = OFF
# pragma config BOR = OFF
# pragma config OSC = INTIO2
int i;
char key;
char password[7]={'1','2','3','1','2','3'};
char input[7];
void main(void)
{
    int flag=0;
    ADCON1 = 0x0F; //Configuring port A as digital

    TRISAbits.TRISA1 = 0; //Trig as output
    LCDInit();              // Initialize the LCD
    LCDClear();             // Clear the LCD
    CursorHome();           // Bring the cursor to top left position
    CursorMode();
    DispControl();
     PStrToLCD("Enter 6 digit PWD"); //Printing on first line
     WriteCmd(0xC0);                 // Bring the cursor to 2nd line left corner
     CursorMode();
     i=0;
    while(i<6)                   // Check upto 6 digits only and give the status
    {
        key='\0';

        // Assign the pins of keypad to port pins
```

```
        TRISDbits.TRISD0=1;    // Make RD0 as row 1 input
        TRISDbits.TRISD1=1;    // Make RD1 as row 2 input
        TRISDbits.TRISD2=1;    // Make RD2 as row 3 input
        TRISDbits.TRISD3=1;    // Make RD3 as row 4 input
        TRISDbits.TRISD4=0;    // Make RD4 as COL 1 OUTPUT
        TRISDbits.TRISD5=0;    // Make RD5 as COL 2 OUTPUT
        TRISDbits.TRISD6=0;    // Make RD6 as COL 3 OUTPUT

       // Start Scanning the first column
        // Set Col 1=0, Col 2=1, Col 3=1
        LATDbits.LATD4=0;
        LATDbits.LATD5=1;
        LATDbits.LATD6=1;

        // CHECKING THE FIRST COLUMN KEYS
        if(PORTDbits.RD0==0)
            key='1';
        if(PORTDbits.RD1==0)
            key='4';
        if(PORTDbits.RD2==0)
            key='7';
        if(PORTDbits.RD3==0)
            key='*';


       // Start scanning the second column
        // set Col 1=1, Col 2=0, Col 3=1
        LATDbits.LATD4=1;
        LATDbits.LATD5=0;
        LATDbits.LATD6=1;
        // Checking the Second Column keys

        if(PORTDbits.RD0==0)
            key='2';
        if(PORTDbits.RD1==0)
            key='5';
        if(PORTDbits.RD2==0)
            key='8';
        if(PORTDbits.RD3==0)
            key='0';

        // Start scanning the third column
        // Set Col 1=1, Col 2=1, Col 3=0
        LATDbits.LATD4=1;
        LATDbits.LATD5=1;
        LATDbits.LATD6=0;
        // Checking the Third Column keys

        if(PORTDbits.RD0==0)
            key='3';
        if(PORTDbits.RD1==0)
            key='6';
        if(PORTDbits.RD2==0)
            key='9';
        if(PORTDbits.RD3==0)
            key='#';

        // printing the pressed key

        Delay1KTCYx(50);    // delay for key de-bouncing
        if(key!=('\0'))
                {
                 WriteDataToLCD(key); // print the pressed key
                 Delay1KTCYx(5);
```

```
                    WriteCmd(0x10);   // move cursor to left to overwrite the pressed
    key
                    Delay10KTCYx(5);
                    PStrToLCD("*");   // overwrite the pressed key with *
                    input[i]=key;     // insert the pressed character into input char
   acter array
                    i++;
                   }

           }

           for(i=0;i<6;i++)
           {
               if(input[i]!=password[i])   // Checking each letter of password
                   flag=1;                 //  if incorrect set flag as 1
           }
           if(flag==0)
           {
               PStrToLCD("WELCOME");
                for(i=0;i<248;i++)    // Giving the pulse untill servo reaches the
   position
                {
                  SERVO  = 1;
                   Delay10TCYx(25); //On for 1ms
                   SERVO = 0;
                   Delay100TCYx(48); //OFF for 19ms approx
                }
           }
           else
           {
               PStrToLCD("INCORRECT");

           }
           Delay10KTCYx(50);

           return;
       }
```

5.3.2 PIR Senses Motion and Blinks Led

```
/*
 * File:   PIRsensor.c
 *
 * Created on September 27, 2019, 7:18 AM
 */

#include <stdio.h>
#include <stdlib.h>
#include <p18f4321.h>
#include <usart.h>
#include <delays.h>
#include <CQU_LCD_Ver_2.h>


//Default Configurations
#pragma config WDT = OFF    //Watch Dog Timer OFF
#pragma config LVP = OFF    //Low VOltage Option OFF
#pragma config BOR = OFF    //Brown Out Reset
#pragma config OSC = INTIO2 //Select Internal Oscillator with I/O on RA6 and RA7

#define Pir_Port_Dir TRISAbits.TRISA2
```

```
#define Pir_Data_In PORTAbits.RA2
//#define Led_Dir TRISCbits.TRISC4
//#define Led_On LATCbits.LATC4
#define Led_Dir TRISDbits.TRISD7
#define Led_On LATDbits.LATD7


void main(void) {
    OSCCON = 0x60;   //4MHz internal clock oscillator
    ADCON1 = 0x0F;


    Pir_Port_Dir = 1; //as input
    Led_Dir = 0; //as output
    Led_On = 0; //initially led D3 off

    while(1){
        while(Pir_Data_In == 1){
            Led_On = 1;
            Delay10KTCYx(2);
        }

        Led_On = 0;
    }

}
```

5.3.3 Relative Humidity on LCD Through DHT22

```
#include <stdio.h>
#include <stdlib.h>
# include <p18f4321.h>
# include <delays.h>
# include <CQU_LCD_Ver_2.h>

# pragma config WDT = OFF               //Watchdog timer OFF
# pragma config LVP = OFF               // Low Voltage Programming OFF
# pragma config BOR = OFF               // Brown Out Reset OFF
# pragma config OSC = INTIO2            // Select internal oscillator

#define Data_Out LATAbits.LATA1
#define Data_In PORTAbits.RA1
#define DataDir TRISAbits.TRISA1

char value[10];
unsigned char CheckSum;
unsigned char T_Byte1, T_Byte2, RH_Byte1, RH_Byte2;
unsigned short RH, Temp;

unsigned short StartSignal(){
    DataDir = 0;
    Data_Out = 0;
    Delay1KTCYx(25);   //Sending logic low for at least 18ms we have set up 25 ms
    Data_Out = 1;
    Delay10TCYx(3);    // Sending logic high for 20-40us here we have set up 20us
    DataDir = 1;       //Data port as input
    Delay1TCY();
    Delay1TCY();//2us delay
    return 1;
}

unsigned short CheckResponse(){
```

```
     if (!Data_In){
     while (!Data_In){}  //waiting till sensor starts sending ones
     if (Data_In){
       while (Data_In){} //waiting till 80us high period is over
       Delay10TCYx(2);      //just in case delay 20us
       return 1;
     }
  }
  else if (Data_In){      // if data pin is still high, then something is wrong
        PStrToLCD("No response");
        return 0;
  }
}

unsigned char ReadByte(){
  unsigned char data = 0, i;

  for (i=0; i<8; i++){
   while(Data_In == 0);
   Delay10TCYx(5); // 50us delay
   if(Data_In == 1){             /* check whether data is 1 or 0 */
                data = ((data<<1) | 1);
                while(Data_In == 1);
    }else{
        data = (data<<1);
    }
  }
  return data;
}

void main(void) {

    unsigned short start, check;
    OSCCON = 0x60;   //4MHz internal clock oscillator
    ADCON1 = 0x0F;   //Making analog port A as digital
    TRISCbits.TRISC0 = 1; //Switch 2 as input

    TRISD = 0;
   //LCD initialization
    LCDInit();
    LCDClear();
    CursorHome();
    CursorMode();
    DispControl();

    PStrToLCD("Press Switch 2");
    while(1){
    if(PORTCbits.RC0 == 0){
    start = StartSignal();
    if(start){
        check = CheckResponse();
    }
    if(!check){
        PStrToLCD("Sensor fail");
        Delay1KTCYx(200); //Delay 200ms
    }else{

        RH_Byte1 = ReadByte();
        RH_Byte2 = ReadByte();
        T_Byte1 = ReadByte();
        T_Byte2 = ReadByte();
        CheckSum = ReadByte();

        Temp = T_Byte1;
        Temp = (Temp << 8) | T_Byte2;

```

```
                RH = RH_Byte1;
                RH = (RH << 8) | RH_Byte2;

                sprintf(value,"RH %d Tmpr %d   ", RH/10, Temp/10);
                LCDClear();
                DStrToLCD(value);
                Delay1KTCYx(20); //Delay 2ms
            }
           }
          }
        }

```

5.3.4 Display LM35 Temperature Reading on 7 Segment And Run A Motor

```
#include <stdio.h>
#include <stdlib.h>
#include <adc.h>
#include<CQU_LCD_Ver_3.h>
#include <delays.h>
#include<p18f4321.h>


#pragma config WDT = OFF
#pragma config LVP = OFF
#pragma config BOR = OFF
#pragma config OSC = INTIO2

void main(void)
{
int div1,div2,digit;
int input_value,reading_i;
TRISD =0;
ADCON1=0x0A;  // disable the analog function and setting the pins as Digital
    TRISBbits.TRISB2=0;
    TRISBbits.TRISB3=0;
        //configure ADCON register and set channel 2 or AN2 for input
    OpenADC( ADC_FOSC_2 &
    ADC_RIGHT_JUST &
    ADC_2_TAD ,
    ADC_CH2 &
    ADC_INT_OFF &
    ADC_REF_VDD_VSS, 15);
```

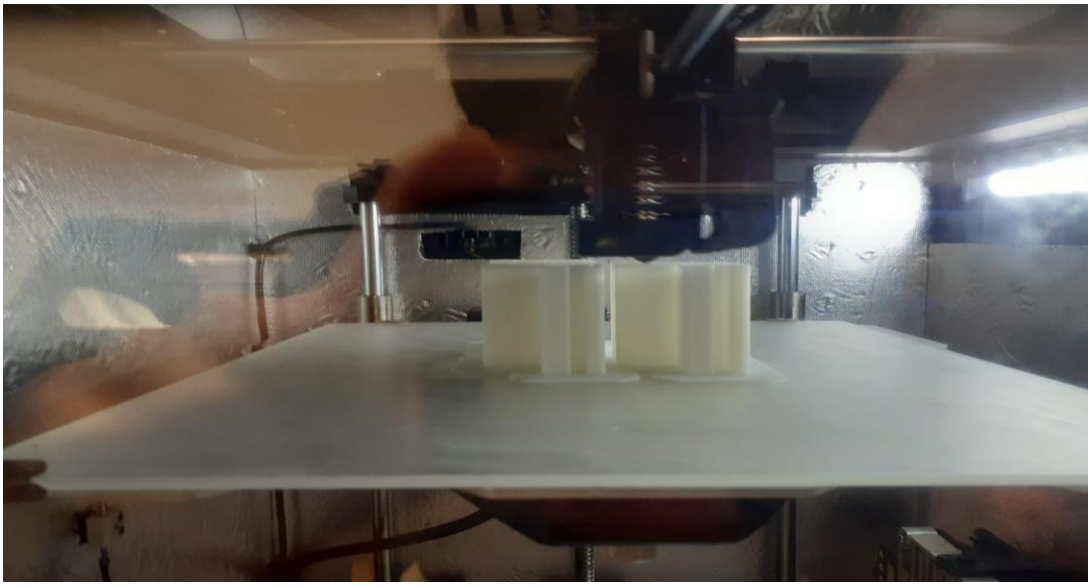
```
29.     Delay10TCYx(5); // delay of 50 Clock cycles
30.     ConvertADC(); // start conversion
31.     while (BusyADC()); //wait for the completion communication
32.     input_value = ReadADC(); // store the ADC value ranging from 0 to 1023
33.     CloseADC(); // disable ADC
34.     reading_i = (input_value*4.88)/10; // conversion to 5 V scale without decimal p
    art
35.     div1=reading_i/10;
36.     div2=reading_i%10;
37.     digit=div2|div1<<4;
38.      LATD=digit;
39.      if(reading_i>23)
40.      {
41.          LATBbits.LATB3=0; //ENABLE A=0  runs motor
42.          LATBbits.LATB2=1;
43.      }
44.      else
45.      {
46.          LATBbits.LATB3=1;    // ENABLE A=1 stops motor
47.          LATBbits.LATB2=1;
48.      }
49.      Delay10KTCYx(15);
50.
51.   return;
52.
53.  }
```

5.4 Developing Physical Prototype

The home automation project is demonstrated in a miniature house like physical prototype made from plastic sheet of 5mm thickness. The step by step process involved in making the prototype in listed below.

5.4.1 3D Printing the Pillar

The pillar that supports the wall structure is made by 3D printing and we have used the lab facilities to 3D print the pillar. We made 4 pillars to support the 4 walls.



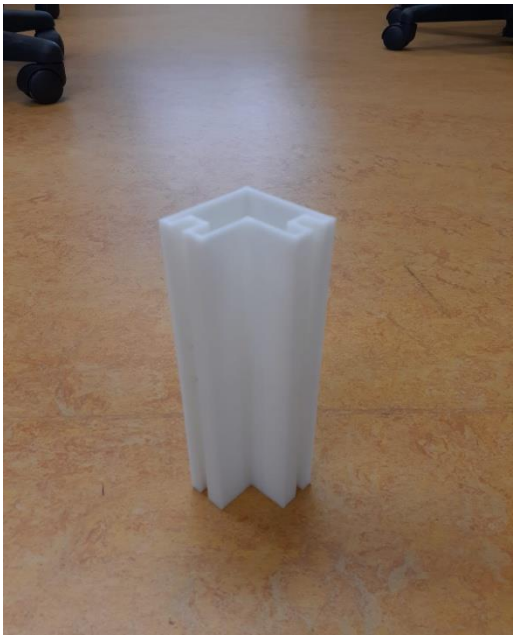


Figure 16 3D Printed Pillar

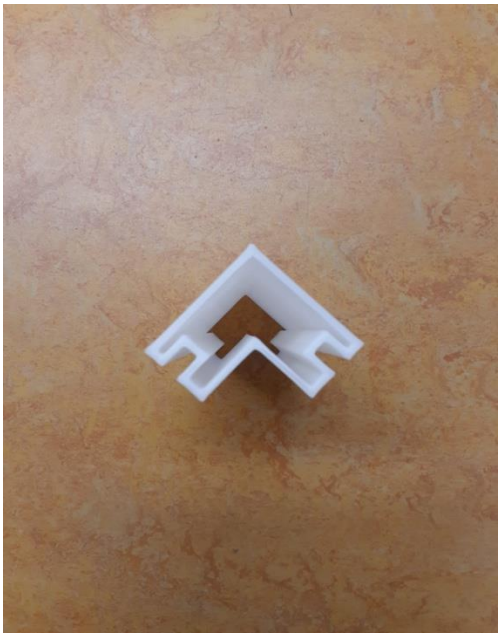


Figure 17 3D Printed Pillar Topview

5.4.2 Fixing Keypad and Door and Door Opening on the Wall

The plastic sheet was cut into pieces of size 15cm X 25cm and 15cm X 30cm, two each for the wall structure and one 25cm X 30cm for the ceiling structure. The keypad is fixed on to a wall

structure and its wire is running through a hole. A square was cut to resemble the door space and a small door was cut to fix with servo to act like a door.

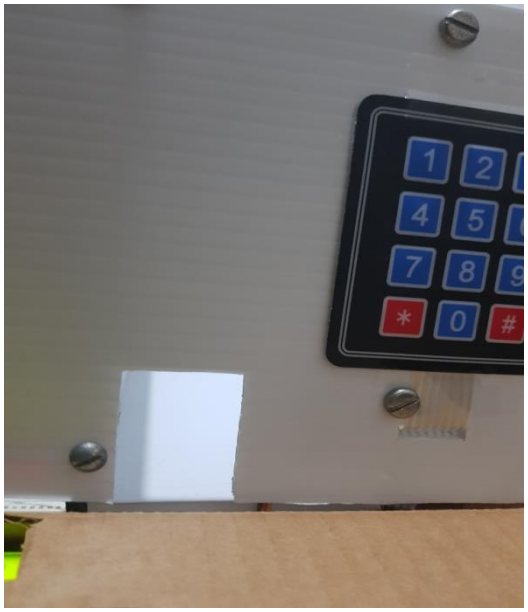


Figure 17 Door Opening And Keypad

Figure 18 Door

5.4.3 Fixing Servo Motor and PIR Sensor on The Passage

The servo motor is fixed on the plastic to facilitate the door opening action. The PIR sensor is fixed on the end of the passage to ensure the door closing after the visitor enters through the door and passes through the door.

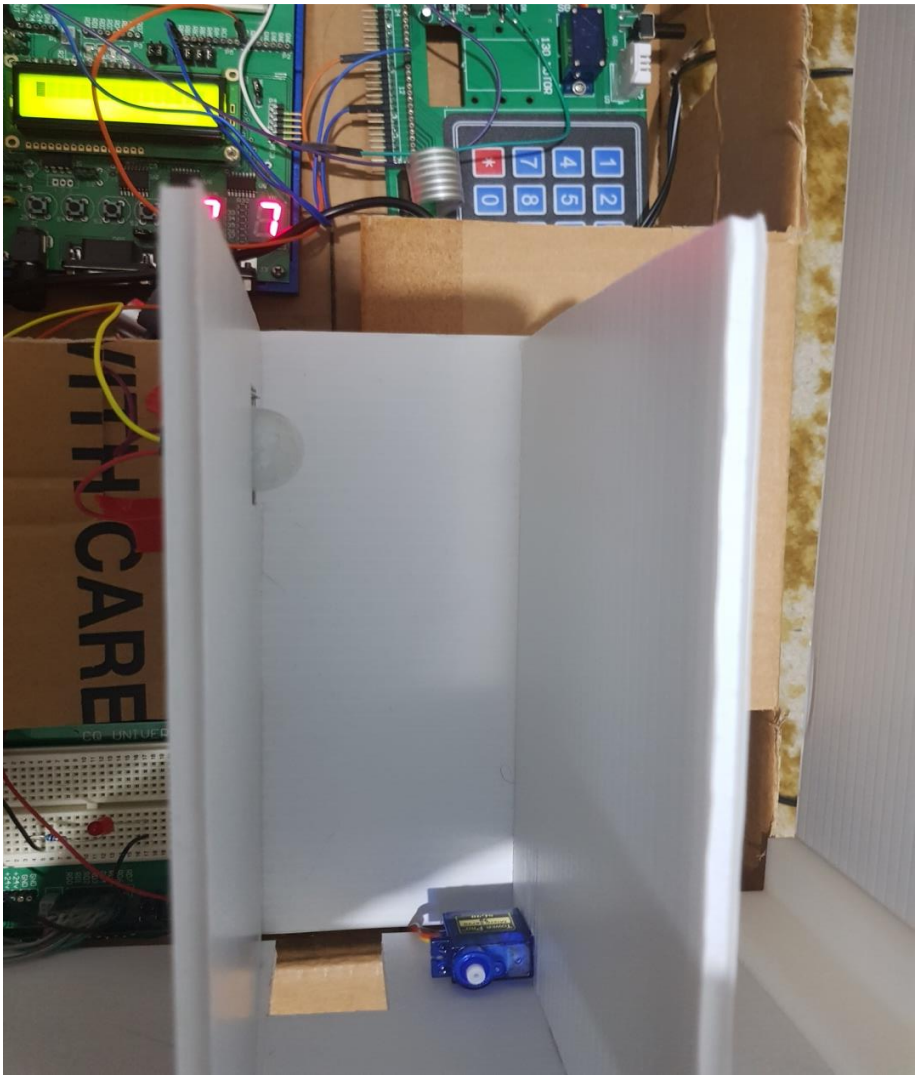


Figure 19 The Passage Topview

5.4.4 Fixing the Motor to Maintain the Temperature

The motor is fixed on the wall and this motor runs when the temperature exceeds the pre-set value.

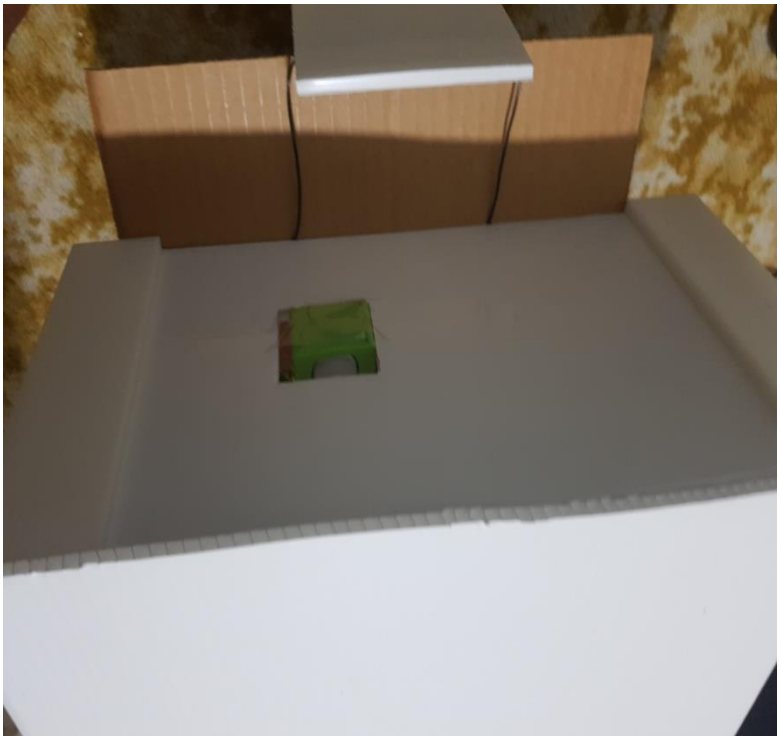


Figure 20 Motor Fixed On Wall

5.4.5 Fixing the LCD On the Cardboard

The LCD on the PIC development board is fixed on the card board to display the WELCOME message and facilitate PASSWORD entry.

Figure 21 LCD portion of the Board is visible through Cardboard

5.5 Putting Them All Together

The whole part is integrated, and all the connection are made, and all the program is loaded to the microcontroller.

Figure 22 Front View

Figure 23 Back View

5.6 Explanation of All Code Segment

The whole project program is fed into two microcontroller and the actuator command is given by both of them individually.

5.6.1 Microcontroller – 1

This microcontroller is loaded with program that interface the Keypad Entry , Password Display, Servo Control, and PIR sensor input.

Step 1: The LCD will always give a prompt to “Enter the 6 digit passcode” and it is made possible by using CQU_LCD_Ver_3 library.

Step 2: The Keypad number entry is obtained by setting HIGH on the Column1, 2, 3 sequentially and checking the input from ROW 1, 2, 3, 4 to get the appropriate key pressed by user. This pressed key is displayed on the LCD and checked whether correct or not

Step 3: If the password entered is not correct display “INCORRECT” message on display or if the password is correct print “WELCOME” and give 1ms pulse for a time period to the servo motor to open the door.

Step 4: If the PIR sensor has sensed any motion through the passage , the microcontroller will sent 2ms pulse to the servo to close the door.

5.6.2 Microcontroller - 2

This microcontroller gets input from DHT22 and displays it on the LCD. The temperature from LM35 is fed into this microcontroller and it controls the motor.

Step 1: The LCD displays press “SW2 or SW3 “which is a prompt to select between two modes.

Step 2: The first mode is to display the Relative Humidity on LCD. In order to get the RH value, the microcontroller first sent a start signal to wake the DHT22 sensor. Then it waits to receive a 40-bit data, which is decoded to obtain the RH value.

Step 3: The second mode displays the temperature on the 7 segment display. The analog values from LM35 is read and checked whether it exceeds the pre-set value. If the temperature increases the pre-set value, the microcontroller turns ON the motor.

6.0 Results and Discussion

The individual sensors and actuators are tested individually before making sure all of them will work together. We have successfully conducted all the testing of the individual sensors and actuators and it worked perfectly alright.

6.1 Testing Individual Sensors and Actuators

The sensors and actuators are tested individually. The step by step testing is given below:

6.1.1 The LCD Display of Password and Door Opening

Figure 24 LCD Display of Password Entered

6.1.2 Relative Humidity Display on LCD

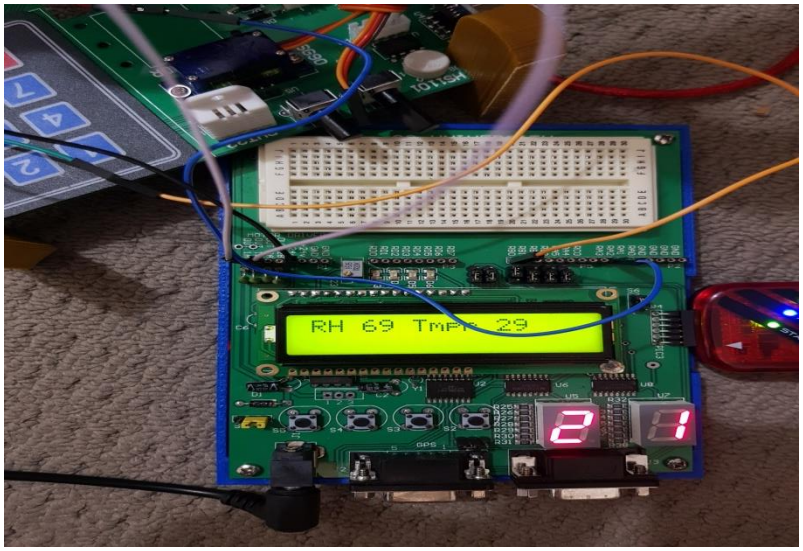


Figure 25 RH Displayed on LCD

6.1.3 The Motor Turns ON after Exceeding the Preset Temperature

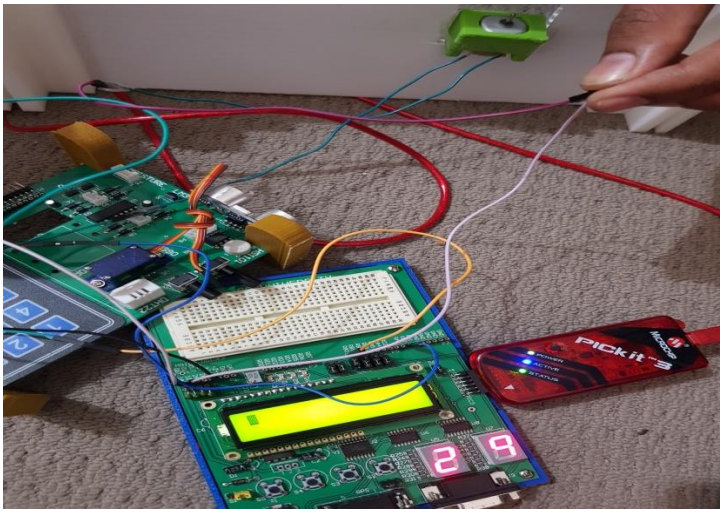


Figure 26 Motor ON when temp exceeds pre-set value

6.2 Testing the Complete Prototype

The whole project is presented sequentially and the below given images is shown in the order in which a visitor enters the house.

6.2.1 Password Entry

Figure 27 LCD Prompt for Password

Figure 28 Password Typing

6.2.2 Door Opening

Figure 29 Door Open by Entering Correct Password

6.2.3 Relative Humidity Measure

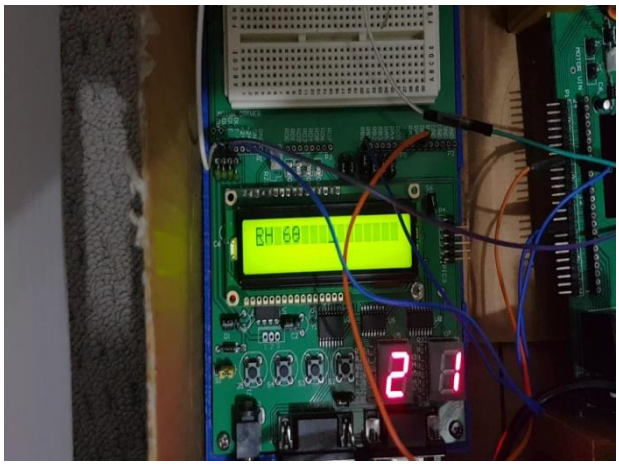


Figure 30 Initial RH Value

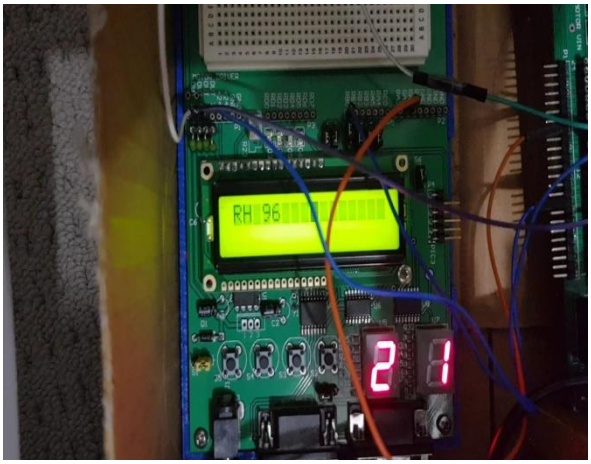


Figure 31 RH Value After Blowing Manually

6.2.3 Temperature Display On 7 Segment Display and Motor Turns On

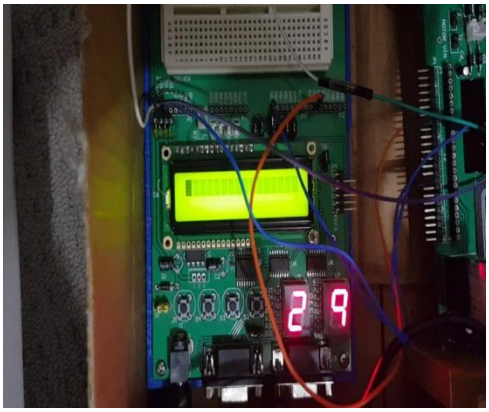


Figure 32 Temperature On 7 Segment Display

Figure 33 Motor On: Temp Exceeds Pre-set Values

7.0 Recommendations for Future Development

As the project has been completed for this term, other major additions of sensors and actuators can be done for future development. For example, the data from temperature and humidity sensors can be collected and logged for data analysis, and graphs and information can be presented to the user in an interactive manner. Different IoT (Internet of things) concepts can be implemented using the Wi-Fi module to connect the microcontroller to the internet, and sensors and actuators can be activated accordingly through the use of mobile apps or web apps. For example, an ESP8266, which is a low-cost Wi-Fi microchip with full TCP/IP stack and microcontroller capability, can be interfaced with the microcontroller and information from sensors can be accessed from the microcontroller through internet and actuators can be activated as desired from smartphone or computer by the user/client using some web app or smartphone app.

Concepts of advanced home automation can be achieved by connecting every electrical appliance, sensors even clocks, lights, and doorbells as well as computers and phones to the internet and making them communicate with each other for the desired task.

8.0 Conclusion

The world is evolving with new technologies and it is being reflected even in our house. The significance of home automation arises very rapidly, as the demand for convenience of humans always increases. This project is a scale down version of the capabilities of home automation and the real fact is that home automation can make even more unbelievable convenience in the human lifestyle. We believe that our home automation project is a stepping stone to our journey to the mesmerizing automation technologies.

This project is our humble effort to demonstrate the current home automation technologies and we are thankful to all the support by the unit coordinator Prof. Preethi Preethichandra and our lab facilitator Troy.

9.0 References

Xfinity. (2019). *What Is Home Automation and How Does it Work? | Xfinity*. [online] Available at: https://www.xfinity.com/hub/smart-home/home-automation

[Accessed 7 Oct. 2019].

Apm.org.uk. (2019). *What is project management?*. [online] Available at: https://www.apm.org.uk/resources/what-is-project-management/

[Accessed 29 Sep. 2019].

Projects, S. (2019). *DHT22 Sensor interface with PIC microcontroller - mikroC Projects*. [online] Simple Projects. Available at: https://simple-circuit.com/mikroc-dht22-sensor-pic-microcontroller/

[Accessed 7 Oct. 2019].

Electronicwings.com. (2019). *PIR Motion Sensor Interface with PIC18F4550 | PIC Controllers*. [online] Available at: https://www.electronicwings.com/pic/pir-motion-sensor-interface-with-pic18f4550

[Accessed 7 Oct. 2019].

Www2.meethue.com. (2019). *Philips Hue Smart lighting* | Philips Hue. [online] Available at: https://www2.meethue.com/en-au/philips-hue-benefits

[Accessed 5 Oct. 2019].

Chubbhomesecurity.com.au. (2019). *Chubb Home Security*. [online] Available at: https://www.chubbhomesecurity.com.au/

[Accessed 5 Oct. 2019].

Anon, (2019). [online] Available at: http://www.ti.com/lit/ds/symlink/lm35.pdf

[Accessed 7 Oct. 2019].

Adafruit Learning System. (2019). *PIR Motion Sensor*. [online] Available at: https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/

[Accessed 7 Oct. 2019].

Sparkfun.com. (2019). [online] Available at: https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf

[Accessed 7 Oct. 2019].

10.0 Appendices

10.1 User Manual of the Prototype

The user manual for this project prototype is straight forward. The project can be divided into two parts, the first one is Automated door and Security, and second one is Temperature control and Humidity.

Let's discuss the use of Automated door and Security first.

Usage of Automated door and Security

1. As the user comes near to the door, he/she will be prompted to enter 6 digits pin password through LCD display.
2. The user has to enter a password through the keypad just beside the main door.
3. If the user enters the correct password, "Welcome" message is displayed on the LCD and the main door opens, if the password is incorrect, a message "Incorrect" will be displayed on the LCD and door remains closed.
4. Now, as the door opens the user walks through the alleyway, in the alleyway there is a PIR sensor that detects for movements, and as it detects user movements the main door is closed.

Now, let's discuss the use of Temperature Control and Humidity module.

Usage of Temperature Control and Humidity Module

1. The user will be prompted with a message to press switch 2 or switch 3 from the CQU pic microcontroller board.
2. If the user presses switch 2 the humidity sensor gets activated and the relative humidity value is displayed in the LCD display.
3. If the user presses switch 3 the LM35 temperature sensor gets activated and temperature value is displayed on the seven-segment display.
4. The temperature control is done with the help of DC motor fan, if the temperature read from LM35 exceeds the pre-set temperature value stored in microcontroller the DC motor fan starts to run and runs until the temperature is under control.
5. The program loops continuously in a sense that every time the user presses either switch 2 or switch 3 the humidity sensor or LM35 activates respectively.

Here is a detailed pictorial representation of the project prototype with sensors and actuators in respective places.

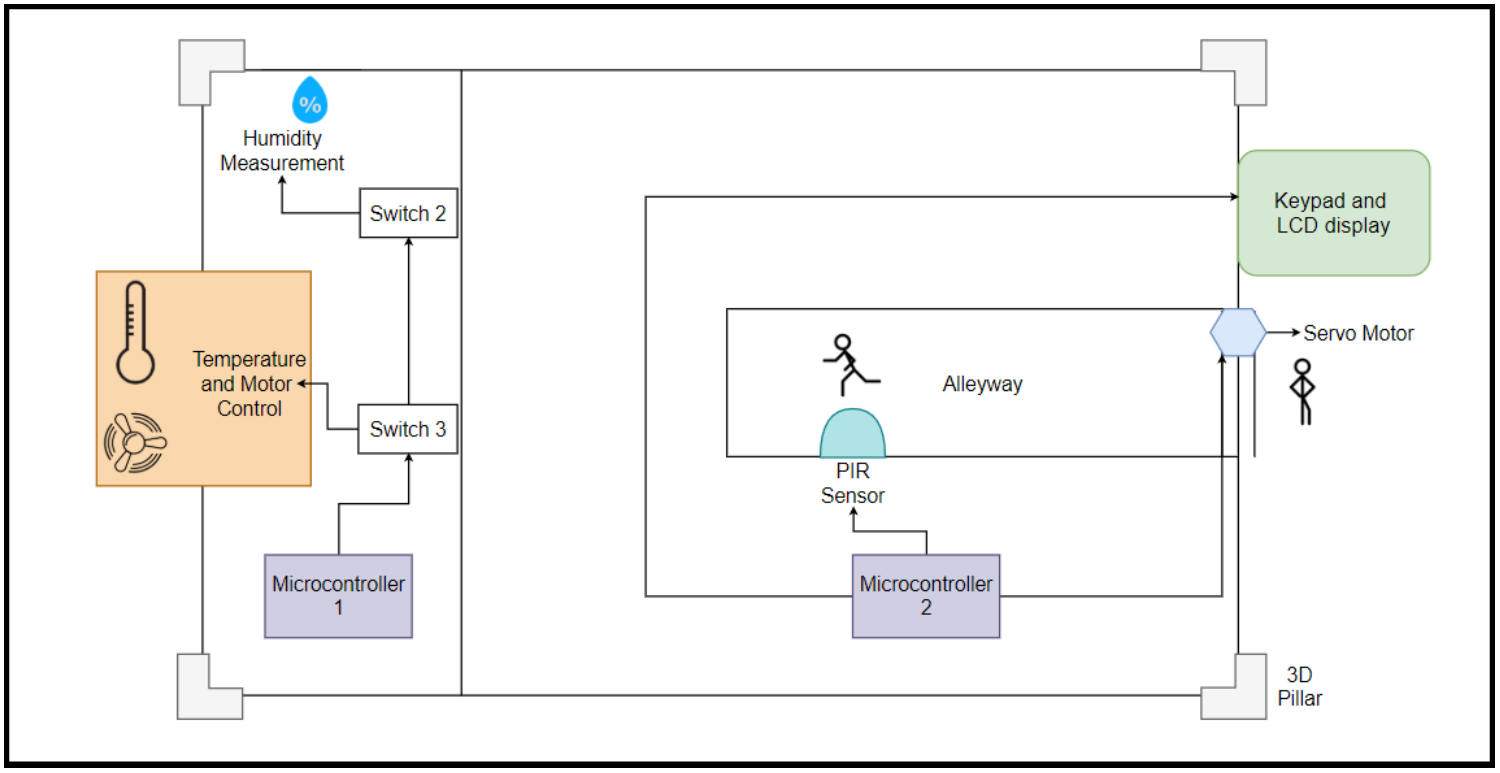


Figure 34 Detailed Pictorial Representation of Project Prototype

10.2 Service Manual of the Prototype

So, before troubleshooting any errors it is necessary to understand the datasheets of every sensors and actuators.

10.2.1 Wiring Troubleshoot

The hardware connections problems are rectified by properly checking the connection of each sensors and actuators with Microcontroller. The connection problem arises whenever there is a huge vibration or shake happens to the whole prototype. The connectivity of both the microcontrollers are given below.

Microcontroller 1	Sensor/Actuator
RD0	ROW1 SENSOR BOARD (KEYPAD)
RD1	ROW2 SENSOR BOARD (KEYPAD)
RD2	ROW3 SENSOR BOARD (KEYPAD)
RD3	ROW4 SENSOR BOARD (KEYPAD)
RD4	COL1 SENSOR BOARD (KEYPAD)
RD5	COL2 SENSOR BOARD (KEYPAD)
RD6	COL3 SENSOR BOARD (KEYPAD)
RD7	LED ON BREADBOARD
RA1	SERVO DATA PIN
RA2	PIR DATA PIN

Microcontroller 2	Sensor/Actuator
RB0	DHT22 DATA PIN
AN2	LM35 PIN
1A,1B	MOTOR TERMINALS
24 V	VCC SENSOR BOARD

10.3.2 Software Troubleshoot

The PIC18f4321 microcontroller is programmed through MPLAB x IDE in C programming language with the compiler MCC18. So, any problem with the software code should be consulted with MCC18 compiler guidelines and instructions and respective sensor/actuator protocol.

10.2.2 Actuator and Sensor Position Error

The placement of every sensors and actuators is presented in the detailed pictorial representation of project prototype. Check the position of sensors and actuators and make sure that they are placed properly in their respective slots.

10.2.3 Microcontroller Fault and Sensor Fault

- The faulty PIC18f4321 can only be rectified by replacing with a new microcontroller board
- If the sensor is giving incorrect values even after the proper connection, then the sensor should be replaced with new sensors. The correct version of the sensors/actuators should be purchased.

10.3 Complete Codes

10.3.1 Complete Code of Automated Door and Security

```
#include <stdio.h>

#include <stdlib.h>

#include <p18f4321.h>

#include<string.h>

#include <usart.h>

#include <delays.h>

#include <CQU_LCD_Ver_3.h>


//Default Configurations

#pragma config WDT = OFF    //Watch Dog Timer OFF

#pragma config LVP = OFF    //Low VOltage Option OFF

#pragma config BOR = OFF    //Brown Out Reset
```

```
#pragma config OSC = INTIO2 //Select Internal Oscillator with I/O on RA6 and RA7

#define ROW1 PORTDbits.RD0
#define ROW2 PORTDbits.RD1
#define ROW3 PORTDbits.RD2
#define ROW4 PORTDbits.RD3

#define Pir_Port_Dir TRISAbits.TRISA2
#define Pir_Data_In PORTAbits.RA2
#define Led_Dir TRISDbits.TRISD7
#define Led_On LATDbits.LATD7
#define SERVO PORTAbits.RA1
int i;
char key;
char password[7]={'1','1','1','1','1','1'};
char input[7];

void main(void){
  int flag=0;
   ADCON1 = 0x0F;
   Pir_Port_Dir = 1; //as input
   Led_Dir = 0; //as output
   Led_On = 0; //initially led D3 off
```

```
 // OSCCON = 0x60;
//  ADCON1 = 0x0F; //Configuring port A as digital


  TRISAbits.TRISA1 = 0; //Trig as output
  LCDInit();            // Initialize the LCD
  LCDClear();           // Clear the LCD
  CursorHome();         // Bring the cursor to top left position
  CursorMode();
  DispControl();
   PStrToLCD("Enter 6 digit Pass"); //Printing on first line
   WriteCmd(0xC0);               // Bring the cursor to 2nd line left corner
   CursorMode();
   i=0;
  while(i<6)                // Check upto 6 digits only and give the status
  {
          key='\0';


     // Assign the pins of keypad to port pins
     TRISDbits.TRISD0=1;    // Make RD0 as row 1 input
     TRISDbits.TRISD1=1;    // Make RD1 as row 2 input
     TRISDbits.TRISD2=1;    // Make RD2 as row 3 input
     TRISDbits.TRISD3=1;    // Make RD3 as row 4 input
     TRISDbits.TRISD4=0;    // Make RD4 as COL 1 OUTPUT
     TRISDbits.TRISD5=0;    // Make RD5 as COL 2 OUTPUT
```

```
 TRISDbits.TRISD6=0;    // Make RD6 as COL 3 OUTPUT


// Start Scanning the first column
 // Set Col 1=0, Col 2=1, Col 3=1
 LATDbits.LATD4=0;
 LATDbits.LATD5=1;
 LATDbits.LATD6=1;

 // CHECKING THE FIRST COLUMN KEYS
 if(ROW1==0)
    key='1';
 if(ROW2==0)
    key='4';
 if(ROW3==0)
    key='7';
 if(ROW4==0)
    key='*';

// Start scanning the second column
 // set Col 1=1, Col 2=0, Col 3=1
 LATDbits.LATD4=1;
 LATDbits.LATD5=0;
 LATDbits.LATD6=1;
 // Checking the Second Column keys
```

```
if(ROW1==0)
   key='2';
if(ROW2==0)
   key='5';
if(ROW3==0)
   key='8';
if(ROW4==0)
   key='0';

// Start scanning the third column
// Set Col 1=1, Col 2=1, Col 3=0
LATDbits.LATD4=1;
LATDbits.LATD5=1;
LATDbits.LATD6=0;
// Checking the Third Column keys

if(ROW1==0)
   key='3';
if(ROW2==0)
   key='6';
if(ROW3==0)
   key='9';
if(ROW4==0)
```

```
        key='#';


    // printing the pressed key


    Delay1KTCYx(50);    // delay for key de-bouncing
    if(key!=('\0'))
      {
       WriteDataToLCD(key); // print the pressed key
       Delay1KTCYx(5);
       WriteCmd(0x10);   // move cursor to left to overwrite the pressed key
       Delay10KTCYx(5);
       PStrToLCD("*");   // overwrite the pressed key with *
       input[i]=key;     // insert the pressed character into input character array
       i++;
      }



  }


  for(i=0;i<6;i++)
  {
    if(input[i]!=password[i])   // Checking each letter of password
        flag=1;                 //  if incorrect set flag as 1
  }
```

```
if(flag==0)
{
   PStrToLCD("WELCOME");
   // OPENING DOOR
   for(i=0;i<248;i++)
    {
      SERVO  = 1;
      Delay10TCYx(25); //On for 1ms
      SERVO = 0;
      Delay100TCYx(48); //OFF for 19ms approx
    }

   while(Pir_Data_In == 0)
   {} //Wait till PIR data is high

   if(Pir_Data_In == 1){
         Led_On = 1;
          // CLOSING DOOR
         for(i=0;i<248;i++)
            {
               SERVO  = 1;
               Delay10TCYx(50); //On for 2ms
               SERVO = 0;
               Delay100TCYx(45); //OFF for 18ms approx
```

```
            }
            Delay10KTCYx(200);
            Led_On = 0;
        }


    }else
    {
        PStrToLCD("INCORRECT");

    }

     Delay10KTCYx(50);
}
```

10.3.2 Complete Code of Temperature Control and Humidity

```
#include <stdio.h>

#include <stdlib.h>

# include <p18f4321.h>

# include <delays.h>

# include <CQU_LCD_Ver_2.h>

#include <adc.h>


# pragma config WDT = OFF              //Watchdog timer OFF

# pragma config LVP = OFF              // Low Voltage Programming OFF

# pragma config BOR = OFF              // Brown Out Reset OFF

# pragma config OSC = INTIO2           // Select internal oscillator


#define Data_Out LATBbits.LATB0

#define Data_In PORTBbits.RB0

#define DataDir TRISBbits.TRISB0


char value[10];

unsigned char CheckSum;

unsigned char T_Byte1, T_Byte2, RH_Byte1, RH_Byte2;

unsigned short RH, Temp;

int div1,div2,digit;

int input_value,reading_i;
```

```
void TempandHumi(void);
void LMandMotorControl(void);


unsigned short StartSignal(void){
   DataDir = 0;
   Data_Out = 0;
   Delay1KTCYx(25);   //Sending logic low for at least 18ms we have set up 25 ms
   Data_Out = 1;
   Delay10TCYx(3);    // Sending logic high for 20-40us here we have set up 20us
   DataDir = 1;       //Data port as input
   Delay1TCY();
   Delay1TCY();//2us delay
   return 1;
}


unsigned short CheckResponse(void){
    if (!Data_In){
    while (!Data_In){}  //waiting till sensor starts sending ones
    if (Data_In){
      while (Data_In){} //waiting till 80us high period is over
      Delay10TCYx(2);      //just in case delay 20us
      return 1;
    }
  }
```

```
  else if (Data_In){      // if data pin is still high, then something is wrong
        PStrToLCD("No response");
        return 0;
  }
}

unsigned char ReadByte(void){
  unsigned char data = 0, i;

  for (i=0; i<8; i++){
   while(Data_In == 0);
   Delay10TCYx(5); // 50us delay
   if(Data_In == 1){            /* check whether data is 1 or 0 */
              data = ((data<<1) | 1);
              while(Data_In == 1);
    }else{
       data = (data<<1);
    }
  }
  return data;
}

void LMandMotorControl(void){
    while(1){
```

```
OpenADC( ADC_FOSC_2 &
ADC_RIGHT_JUST &
ADC_2_TAD ,
ADC_CH2 &
ADC_INT_OFF &
ADC_REF_VDD_VSS, 15);
Delay10TCYx(5);                 // delay of 50 Clock cycles
ConvertADC();                   // start conversion
while (BusyADC());              //wait for the completion communication
input_value = ReadADC();        // store the ADC value ranging from 0 to 1023
CloseADC();                     // disable ADC
reading_i = (input_value*4.88)/10;  // conversion to 5 V scale without decimal part
div1=reading_i/10;
div2=reading_i%10;
digit=div2|div1<<4;
 LATD=digit;
 if(reading_i>27)
 {
    LATBbits.LATB3=0; //ENABLE A=0  runs motor
    LATBbits.LATB2=1;
 }
 else
 {
```

```
            LATBbits.LATB3=1;    // ENABLE A=1 stops motor
            LATBbits.LATB2=1;
        }
        Delay10KTCYx(15);
        if(PORTCbits.RC0 == 0){    //if switch 2 pressed goto humidity program
            TempandHumi();
        }
    }
    return;
}
void TempandHumi(void){
    unsigned short start, check;
    while(1){

        Delay10KTCYx(200); //Delay of 2seconds
        start = StartSignal();
        if(start){
            check = CheckResponse();
        }
        if(!check){
            PStrToLCD("Sensor fail");
            Delay1KTCYx(200); //Delay 200ms
        }else{
```

```
        RH_Byte1 = ReadByte();
        RH_Byte2 = ReadByte();
        T_Byte1 = ReadByte();
        T_Byte2 = ReadByte();
        CheckSum = ReadByte();

        Temp = T_Byte1;
        Temp = (Temp << 8) | T_Byte2;

        RH = RH_Byte1;
        RH = (RH << 8) | RH_Byte2;

        sprintf(value,"RH %d   ", RH/10);
        LCDClear();
        DStrToLCD(value);
        Delay1KTCYx(20); //Delay 2ms
      }
      Delay1KTCYx(1);
      if(PORTCbits.RC1 == 0){    //if switch 3 pressed goto LM35 program
        LCDClear();
        LMandMotorControl();
      }
  }
   return;
```

```
}
void main(void) {



    OSCCON = 0x60;   //4MHz internal clock oscillator
    ADCON1 = 0x0A;   //Making analog port A as digital
    TRISCbits.TRISC0 = 1; //Switch 2 as input
    TRISCbits.TRISC1 = 1; //Switch 3 as input
    TRISBbits.TRISB2=0;   //For Motor control A3966
    TRISBbits.TRISB3=0;   //For Motor Control A3966


    TRISD = 0;
   //LCD initialization
    LCDInit();
    LCDClear();
    CursorHome();
    CursorMode();
    DispControl();


    PStrToLCD("SW2 or SW3");
    Delay1KTCYx(100);
    //WriteCmd(0XC0);
    //CursorMode();
    //PStrToLCD("SW 3 for TempControl");
```

```
    while(1){
        if(PORTCbits.RC0 == 0){        //if switch 2 pressed
            LCDClear();
            TempandHumi();
        }else if(PORTCbits.RC1 == 0){  //if switch 3 pressed
            LCDClear();
            LMandMotorControl();
        }
    }

}
```

10.4 Datasheets Used

The list of datasheets of individual sensors, actuators and microcontroller is given below:

- PIC 18F4321 family datasheet

  http://ww1.microchip.com/downloads/en/DeviceDoc/39689b.pdf
- MPLAB C18 C Compiler User's Guide

  http://ww1.microchip.com/downloads/en/DeviceDoc/MPLAB_C18_Users_Guide_51288d.pdf
- LM35

  http://www.ti.com/lit/ds/symlink/lm35.pdf
- PIR

  https://cdn-learn.adafruit.com/downloads/pdf/pir-passive-infrared-proximity-motion-sensor.pdf
- DHT22

  https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf
- LCD 16X2

  http://www.picaxe.com/docs/led008.pdf
- SERVO (SG90)

  http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf
- 7 SEG DISPLAY

  https://e-radionica.com/productdata/LD3361BS.pdf
- DC motor driver A3966h

  https://www.allegromicro.com/~/media/Files/Datasheets/A3966-Datasheet.ashx