Qatar University

College of Engineering

Department of Computer Science and Engineering

CMPE 364 Microprocessor based Applications
Course Project Report
Spring 2022

PROJECT TITLE

HOME MONITORING SYSTEM

(FINAL REPORT)

Submitted By:

1. Marim Elhanafy 201803468
2. Hagar Elsayed 201805123
3. Asma Al-nuaimi 201608057
4. Fatima Awad 201308262

Table of Contents

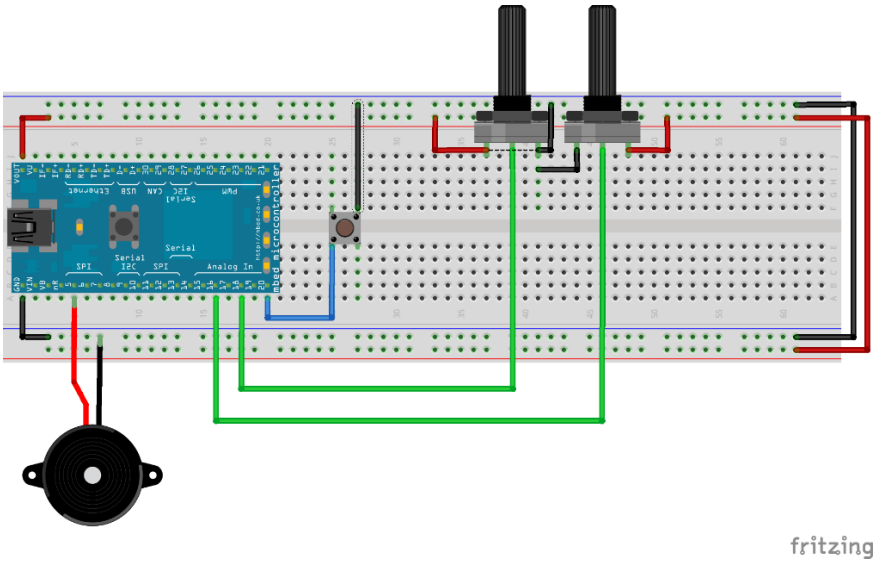
Introduction

In this project we will implement a simple home monitoring system that will monitor a couple of parameters that will do some actions, which are:

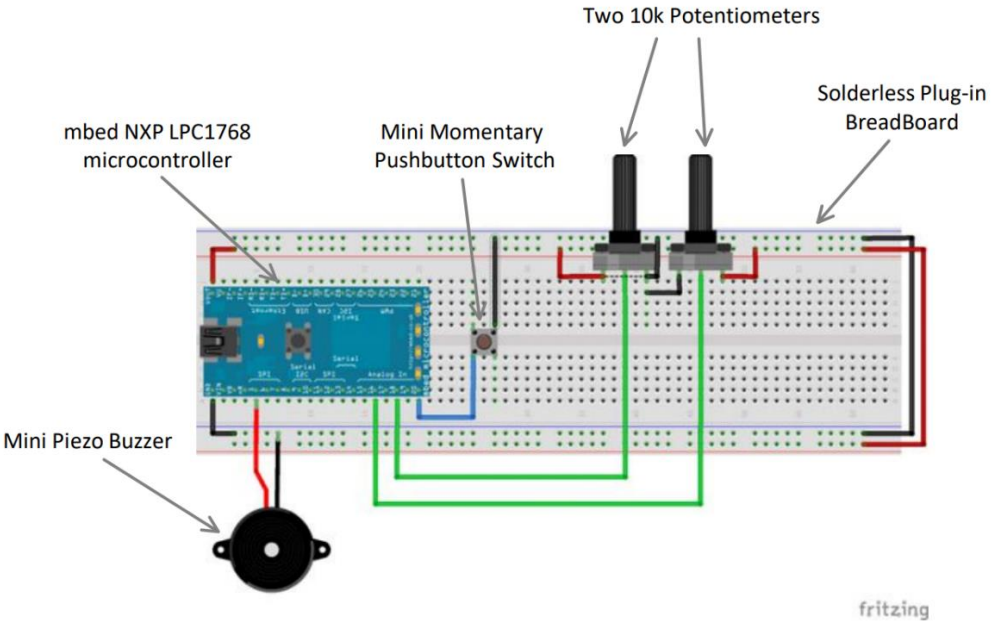
1. Measure the level of light intensity outside the house, and switch automatically (on and off) of the lights depending on it.
2. Measure the level of temperature inside the house and adjust the speed of the fan automatically in response to the monitored temperature.
3. Check the status of front door whether it is opened or closed and starting an alarm if the front door is kept opened for a certain period (5 sec).

Task 1: Hardware Design

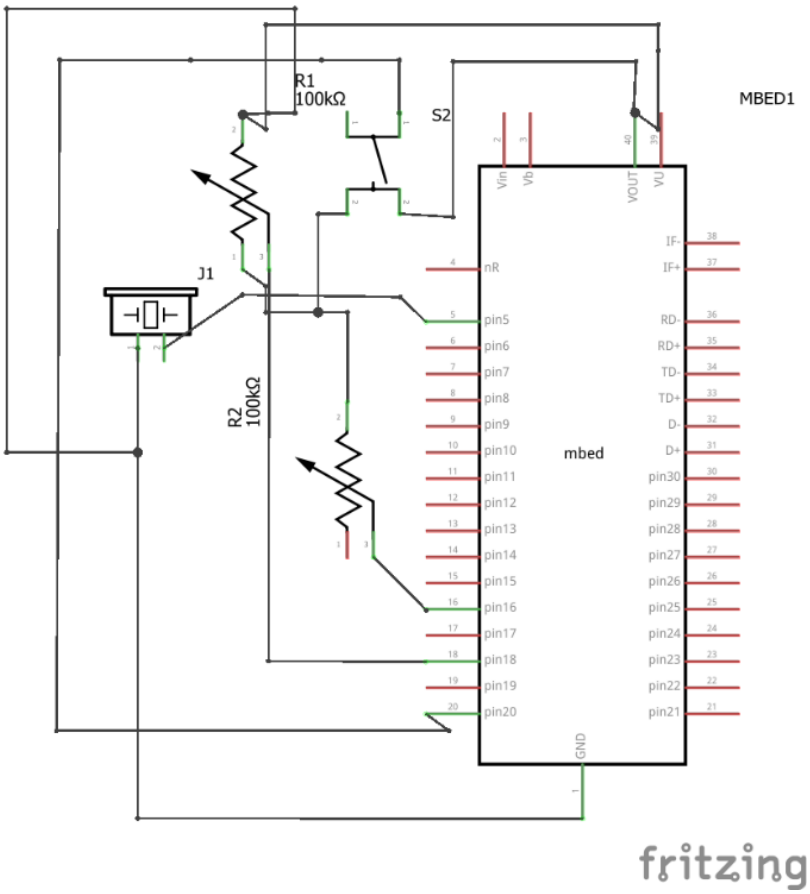
- The hardware design:



- The components used:



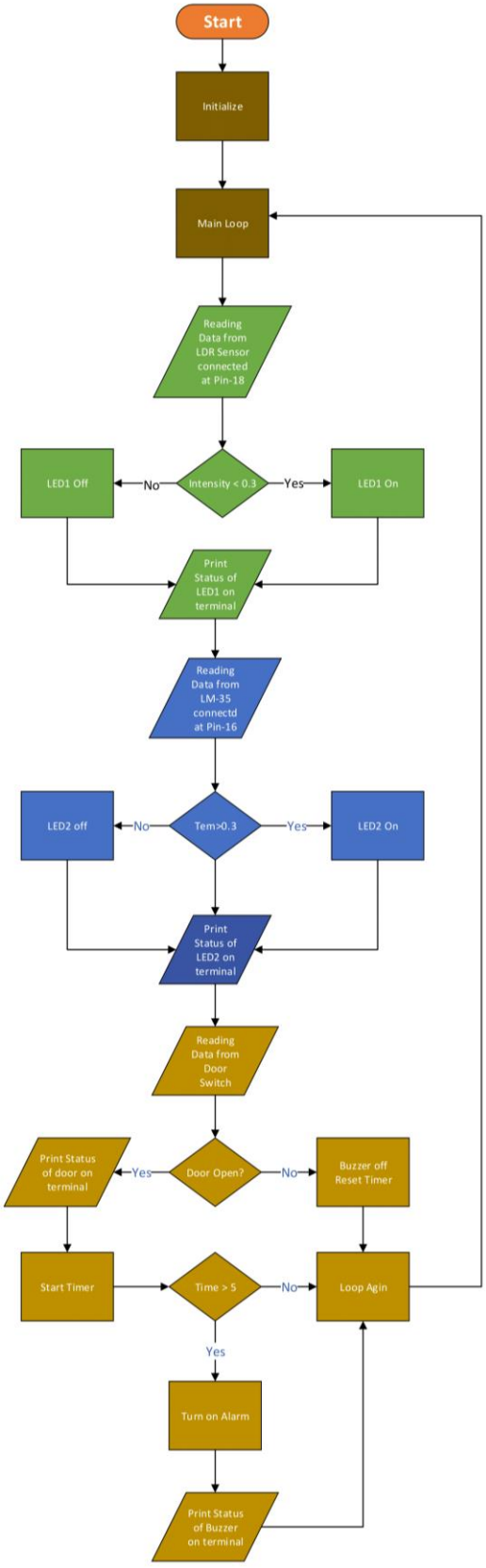
- The Schematic diagram of the design:



- Explanation of the hardware design:
  The home monitoring system is implemented using the Breadboard with mbed NXP LPC1768 microcontroller that connected to:
  - Two potentiometers to simulate analog inputs (instead of temperature sensor and the light sensor).
  - Pushbutton switch to simulate the front door status (opened or closed).
  - Buzzer to produce sound alarm.

  Remark: the LEDs on the LPC 1768 mbed board are used to simulate the switching on and off of lights as well as simulating the fan speed.

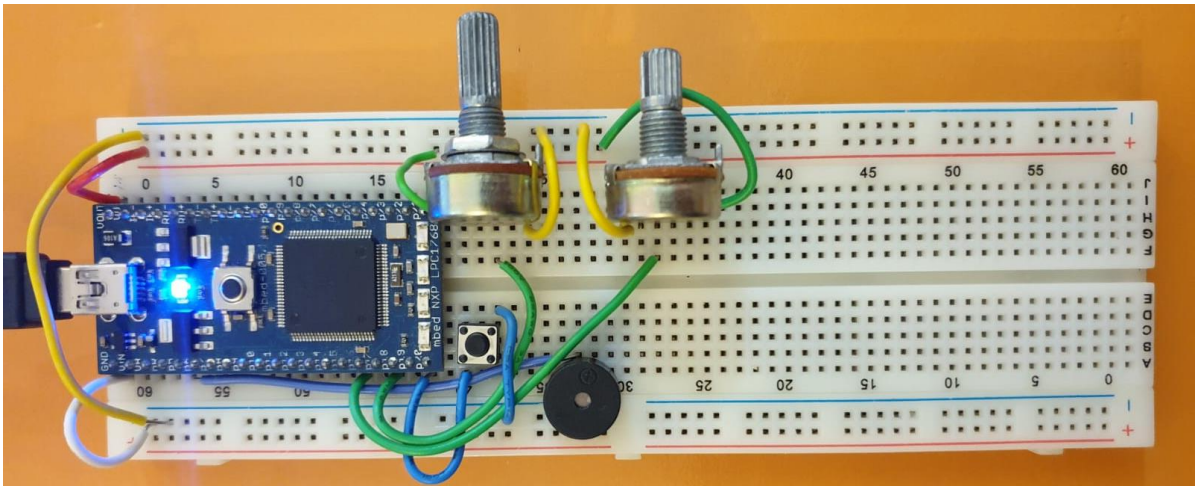
Task 2: Software Design



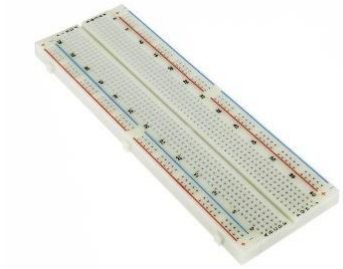
Task 3: Implementation

Hardware Implementation

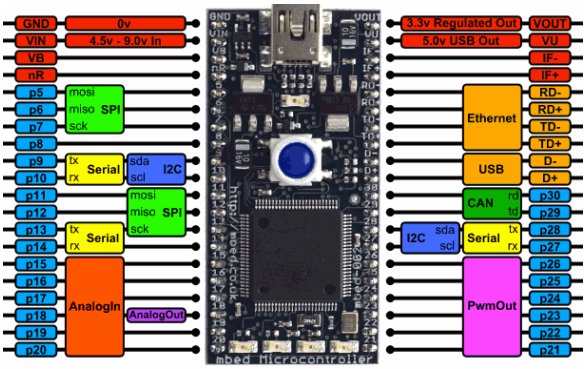
- The implemented hardware system:



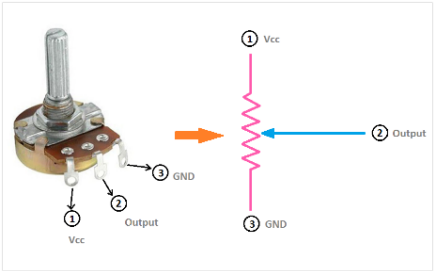
- The diagram used is simple and implemented in efficient way with less components needed.
- The hardware equipment and components used in the implementation:
  1. Solderless Plug-in Breadboard



  2. mbed NXP LPC1768 microcontroller



  3. Two 10k Potentiometers



4. Mini Momentary Pushbutton Switch

5. Mini Piezo Buzzer

6. Wires to connect the components

- The tool used in the implementation:
  - Fritzing

Software Implementation

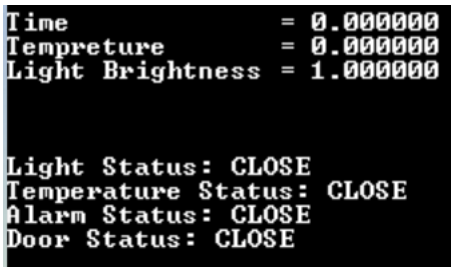
```
#include "mbed.h"

DigitalOut L1(LED1);
DigitalOut L2(LED2);
DigitalOut Buz(p5);

AnalogIn temp(p16);
AnalogIn light(p18);
DigitalIn B(p20);

Serial pc(USBTX,USBRX);

int main() {
    float t=0, data;
    bool DoorStillOpen=false;
    B.mode(PullUp);

    while(1) {

      pc.printf("Time             = %f\n\r",t);
      data = temp.read();
      pc.printf("Tempreture       = %f\n\r",data);
      data= light.read();
      pc.printf("Light Brightness = %f\n\n\n\n\r",data);

      wait(1);

    // Light status
      if(light.read()<=0.3){
          pc.printf("Light Status: OPEN \n\r");
          L1=1;
      }else{
          pc.printf("Light Status: CLOSE \n\r");
          L1=0;
        }

    // Temp status
      if(temp.read()>=0.3){
          pc.printf("Temperature Status: OPEN \n\r");
          L2=1;
      }else{
          pc.printf("Temperature Status: CLOSE \n\r");
          L2=0;
        }

    // Door status
    // 1) Check if door's open
      if(!B && !DoorStillOpen)
          DoorStillOpen=true;

    // 2) Count time while door's open (increment 1 sec)
      if(DoorStillOpen){
          t++;
          wait(1);
      }

    // 3) Check if door's been closed
      if(B) {
```

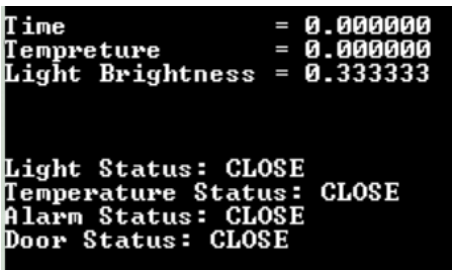
```
            Buz=0;
            DoorStillOpen=false;
            t=0; //reset timer
        }

    // 4) If door's been open for t>5
        if(!B && t>=5) {
            Buz=1;
        }

        if(Buz==1)
          pc.printf("Alarm Status: OPEN \n\r");
        else
          pc.printf("Alarm Status: CLOSE \n\r");

        if(DoorStillOpen)
          pc.printf("Door Status: OPEN \n\n\n\n\r");
        else
          pc.printf("Door Status: CLOSE \n\n\n\n\r");
    }
}
```

- The tools used in developing the code:
  1. Computer running Windows operating system.
  2. Web browser, such as Google chrome or similar.
  3. NXP LPC1768 platform (https://os.mbed.com/platforms/mbed-LPC1768/) using mbed online compiler (https://os.mbed.com/).
  4. TERA Term terminal emulator to facilitate the serial communication between the mbed MCU and the host PC (useful to debug the program on the mbed MCU and to send parameters to the program running on the mbed MCU).
  5. The programming language is C++

Task 4: Testing

Testing of monitoring light intensity





Testing of monitoring temperature level

```
Time              = 0.000000
Tempreture        = 0.000000
Light Brightness  = 1.000000


Light Status: CLOSE
Temperature Status: CLOSE
Alarm Status: CLOSE
Door Status: CLOSE
```

```
Time              = 0.000000
Tempreture        = 0.292796
Light Brightness  = 1.000000


Light Status: CLOSE
Temperature Status: CLOSE
Alarm Status: CLOSE
Door Status: CLOSE
```

```
Time              = 0.000000
Tempreture        = 0.300366
Light Brightness  = 1.000000


Light Status: CLOSE
Temperature Status: OPEN
Alarm Status: CLOSE
Door Status: CLOSE
```

```
Time              = 0.000000
Tempreture        = 1.000000
Light Brightness  = 1.000000


Light Status: CLOSE
Temperature Status: OPEN
Alarm Status: CLOSE
Door Status: CLOSE
```

Testing of monitoring status of front door

```
Time              = 0.000000
Tempreture        = 0.000000
Light Brightness  = 1.000000


Light Status: CLOSE
Temperature Status: CLOSE
Alarm Status: CLOSE
Door Status: CLOSE
```

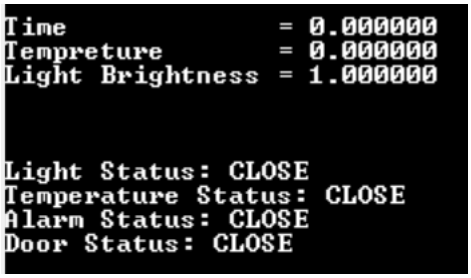
```
Time              = 0.000000
Tempreture        = 0.000000
Light Brightness  = 1.000000


Light Status: CLOSE
Temperature Status: CLOSE
Alarm Status: CLOSE
Door Status: OPEN
```

```
Time              = 1.000000
Tempreture        = 0.000000
Light Brightness  = 1.000000


Light Status: CLOSE
Temperature Status: CLOSE
Alarm Status: CLOSE
Door Status: OPEN
```

```
Time              = 2.000000
Tempreture        = 0.000000
Light Brightness  = 1.000000


Light Status: CLOSE
Temperature Status: CLOSE
Alarm Status: CLOSE
Door Status: OPEN
```

```
Time              = 3.000000
Tempreture        = 0.000000
Light Brightness  = 1.000000


Light Status: CLOSE
Temperature Status: CLOSE
Alarm Status: CLOSE
Door Status: OPEN
```

```
Time              = 4.000000
Tempreture        = 0.000000
Light Brightness  = 1.000000


Light Status: CLOSE
Temperature Status: CLOSE
Alarm Status: OPEN
Door Status: OPEN
```



Conclusion

The simple home monitoring system that will save money and electricity, as the lights will not always be on, as well as make the lifespan of lighting lamps and fans longer, in addition to increasing the safety in the house.