University Of Regina

CS 807

Interactive Hardware

Smart Home Management

Author:
Sai Keerthi Mettu

Supervisor:
Trevor Tomesh

Abstract

These days a great number of people prefer having a smart home since this has a wide range of benefits. To illustrate a few, having a smart home can provide better security and safety, efficient handling of regular household activities, make living more entertaining, save energy and money. Today there are many Smart Home technology advancements like gadgets, entertainment systems, security control systems available in the market. These are plug and play devices that get connected with the internet and provide the sensor information through actuators. Smart home management is a project that came up with a need to design a low-priced device that meets the minimum requirements for an individual home, which is an amalgamation of a couple of security measures along with music play options. The basic requirements of an individual home include a monitoring system for intrusions, fire identification, and safety alarms. Management of the music system in the house should be restricted only for specific users through an authentication process. Different colours light representation module of the device aids to communicate its information even for the unerudite people.

1 INTRODUCTION

This document describes the creation and stages of development of a Smart House Management using Arduino. The project consists of five main modules, which includes a combination of safety measures and entertainment aspects for the users of the system. The modules in the system are:

- Intruder detection
- Fire detection
- Music system for authenticated users
- Display the room environment conditions
- Display on the LCD screen and glow the light in different colors according to each of the working module.

The first two modules are designed to alert the people in the house. This is done through an alarm when there is any person entering the house and in the case of a fire break out in the house surroundings respectively. Ultrasonic sensor is used for the intruder detection and the flame sensor is used for the fire detection module. To play the music, user has the choice of selecting a song from the remote control, only if he/she gets authenticated through the finger print match process on a Touch sensor. Alongside, there will be appropriate messages displayed on the LCD screen and the RGB LED that indicates different colors for different sensor functioning in the system. There is also a provision for authenticated users to turn off or on RBG LED according to the need. When these key sensors don't have any significant information to show on the LCD screen, DTH11 sensor shows the temperature and humidity of the room, where it is placed.

2 BACKGROUND

The main aim of this project is to use some of the easily available sensors and create a simple design that is advantageous in our daily routine in terms of safety, protection and sometimes for entertainment needs. Standard devices like thermostats, fire alarm units, does seem costly to buy for the people living in rural areas. This is also another factor to design a low-priced device

that addresses their problem and helps them in their daily life.
There are basic projects available on the internet in different websites like create.arduino.cc, circuitbasics.com, circuitdigest.com, instructables.com and randomnerdtutorials.com on how to use the components in the Arduino kit. There are projects in Arduino.cc website has shown the usage of ultrasonic sensor and applications like distance measurement and monitoring system of the house for any intruders [2][3]. Combination of these two ideas gives a module to identify the height of the intruder when the ultrasonic sensor is placed on top of the door entrance of the house. The fire detection module was originally developed by Shashi Kumar on circuitdigest.com [4], that actually detects the fire with the flame sensor and signal through a sounding alarm. There are some song notes available in create.arduino.cc and josephmilla websites that helped to the idea of playing music through remote control. To make this module little exclusive, it was combined with the authentication process of users with their fingerprint on the touch sensor [17]. The touch sensor provided in the Arduino kit was not reliable in sensing the fingerprints of users, so it was the only component that was purchased from an online store. RGB LED was implemented as a stretch goal along with LCD display so that the system would become more adaptable and understandable even for uneducated users.

3 DESIGN DESCRIPTION

This section details the connections with each of the sensors used in a specific module. On a whole, all the 12 digital pins and 4 for the analog pins are used on the Arduino board, where the majority of them are with the LCD screen connections. Here goes the design of each module in the system:

3.1 Intruder detection

The components used in this module are:

- Ultrasonic sensor
- Buzzer

The main function of an HC-SR04 ultrasonic sensor is to detect the distance of an object similar to bats [6]. It does a magnificent non-contact sonar

wave run identification with high precision and comes in a simple-to-utilize package. The transmitter within (trigger pin) is responsible for emitting the high-frequency sounds and if those signals hit any object, they reflect back. and receiver (echo pin) modules within, that helps in the distance determination of those reflected waves.
In order to identify any intrusion through a doorway, the ultrasonic sensor has to be placed on a wall above the entrance of a door, so that it detects anything that passes through it. Measure the length of the door to the point where the sensor is placed above the floor. Keep a record of this length as 'Entrance_height'so that it helps in the height identification of the intruder. The buzzer output is read high if the echoes reflect back when the distance length is less than the Entrance_height. So, if any human or object passes through the entrance, the echoes transmitting between the ground surface to its location height would get interrupted. It would let us calculate from which point, the echoes are reflected back and help to identify the height of the object that passed through the entrance.

3.2 Fire detection

The components used in this module are:

- Flame sensor
- Buzzer

The flame sensor is an IR diode, that is a key item used in the fire source detection module. It is capable of reading the fire sources and other sources that are with a wavelength range from 760nm to 1100nm [7]. So, when a fire signal is detected beyond a threshold level from the flame sensor, the buzzer will ring high until the sensor reading goes down below a limit.

3.3 Music system

The components used in this module are:

- Buzzer
- IR receiver
- Remote control

– Touch sensor

This module is designed in a way to play music for different options chosen on the remote control and permit its access only for limited users (owners) of the system. To achieve this stipulation, the fingerprints of the owners have to be enrolled in the system through the fingerprint touch sensor. Therefore, each time when the users want to play the music has to go through this authentication process where the given user's fingerprint on the touch sensor is matched with the enlisted fingerprints. Once the user gets authenticated, they get access to view the list of songs available and to play the songs that are preloaded for selected buttons on the remote control. User can play these music notes, until the time they want to exit from the authenticated mode to normal mode in the system.

3.4 Display room temperature

The only component required in this module:

– DTH11

This sensor is also called the temperature and humidity sensor as it is embedded with both the sensor components of humidity sensing and an NTC (Negative Temperature Coefficient means the inverse proportionality relation the resistance and temperature) temperature sensor Thermistor respectively [8]. When there is an interminable power supply to the DTH11 sensor, it keeps providing to the temperature and humidity values of the room and keeps displaying them to the users when there is no information from the other sensors.

3.5 LCD and RGB lights Integration

The components used in this module are:

– LCD screen

– Potentiometer

– RGB LED

– Touch sensor

Sensor	Color
Ultrasonic	Blue
Flame	Red
Touch	Green
DHT11	White
IR receiver	Purple
Music	aqua
Off authenticate mode	yellow

Table 1: LED colour for each sensor.

The LCD screen is used for the display of each sensor information. Any sensor that has information to show on a serial monitor, will be done through the LCD screen. RGB LED will glow in different colours according to the sensor that is displaying information on the LCD screen. Therefore, the display logic of the LCD and RDB colour setting is called in every module of the respective sensors. The colours that are chosen for the operation of each sensor are mentioned in Table 1.

DTH11 sensor information is displayed continuously when others are idle. So, a white light will be lit throughout the time when the device is given with the power supply. In case if we want to turn off this white light, there is a provision to do so by following the authentication process, which follows a couple of steps on the remote control and the finger match on the touch sensor.

3.6 STEP-BY-STEP Instructions for circuit connection

1. Connect the 5V power supply pin to a positive rail (Vcc rail) and GND pin to the ground rail (GND rail) from Arduino board to the breadboard respectively.

2. The buzzer is the system is connected from the digital pin 10 on Arduino to the anode end and the cathode is wired to the GND rail.

3. An ultrasonic sensor has four pins i.e., Vcc, Trigger, Echo and GND. The trigger and echo are connected to the Arduino digital pins 8 and 9 respectively. The power pin is connected to the Vcc rail and the ground pin to the GND rail on the breadboard.

4. The flame sensor is connected with an analog pin A0 on Arduino board on the anode end and the cathode to the GND rail on the breadboard.

5. DTH11 sensor has four pins of which only three of them are used for connections. First one is the Vcc pin that is joined to the Vcc rail and the last pin of the sensor to the GND rail respectively. The second pin in between is the output pin and is connected to the digital pin 11 on Arduino.

6. IR receiver sensor has got three pins in the order of Vcc pin followed by a ground pin and an output pin. The first two pins are connected to the Vcc rail and GND rail on breadboard respectively. The last pin is connected to the 12th digital pin on the Arduino board.

7. RGB LED has four pins for R, G, B and GND pins. The R, G, B pins are connected with the analog pins on Arduino to A3, A4 and A5 respectively. The GND pin is connected to the GND rail on the breadboard.

8. An Arduino LCD will have 16 pins of which 8 of them are data pins from D0 to D7. The data pins D4, D5, D6 and D7 are only used for connection to the Arduino digital pins 4, 5, 6 and 7 respectively. Vo pin is used for display contrast and gets connected to the potentiometer for variable resistance. RS (register select) and R/W (read/write) pins are also connected to the digital pins 0 and 1 respectively on Arduino board. The remaining pins A (anode), Vcc (5V) pins are connected to the positive rail and GND, E (Enable), K (cathode) are connected to the GND rail on the breadboard respectively.

9. A potentiometer has three pins for Vcc, output and GND. The output pin is connected with the Vo pin on LCD and the rest to their respective rails on the breadboard.

10. A touch sensor has six pins where only four of them are used. The Vcc and GND pins are connected to the positive and negative rails on board. The Tx (transmitter) pin is joined to the digital pin 2 (Software serial, Rx) and the Rx pin on touch sensor to the digital pin 3 (Software serial, Tx) on the Arduino respectively.

4 BUILD PROCESS AND IMPLEMENTATION

4.1 Intruder detection

n the first module, the ultrasonic sensor is installed on the door top to monitor for passers and determine their height. So, the average length of door entrance is considered as 180 centimetres, to use for the intruder's height calculation. The ultrasonic sensor's trigger pin is defined as the output pin and the echo as input pin respectively. When any obstacle passes between the sensor and the ground, the control goes to the IntruderAlarm() function where the buzzer pin is given a digital high signal to start buzzing. Also, it would let us calculate from which point, the echoes are triggered back (by subtracting the echo distance from the entrance height) from the ultrasonic sensor and help to identify the height of the object. The buzzer keeps ringing with constant beep sounds until the trigger waves hit the intermediate object between the sensor and the floor. Along with the buzzing sound, the height of the intruder is displayed on the LCD screen and the RGB LED keeps blinking in blue colour.

4.2 Fire detection

In the second module, the flame sensor's analog data is read through the Arduino pin A0. As a threshold value, 700 is the number chosen on seeing a series of outputs obtained from the experiments done with fire and flame sensor's sample program codes. When the flame sensor exceeds this threshold reading, the buzzer keeps sounding until the sensor reading goes down the limit value. Simultaneously, there will be a message displayed on the LCD screen and a red-light blinking is put on the RGB LED.

4.3 Music system

In the third module, there are four subtasks to implement.

1. IR receiver sensor to sense the button clicks on the remote control.
2. Gather and execute the music notes of different songs.

3. Registration and insertion of authentication functions from built-in libraries.

4. Associate all the above subtasks in a step by step manner.

Firstly, to let the IR receiver sensor allow the transmission and receiving of Infrared remote-control code, IRremote library is included in the code. The built-in IRrecv object is responsible for reading the remote-control input through the data received from the IR sensor's digital pin 12 [5]. The functions enableIRIn() and decode(results) are declared in the setup to begin the receiving process and attempts to receive a code from the IR sensor respectively [11]. Code is written in a way to show the received code for the button clicks on the remote control. Thus, the received codes are noted for all the buttons present on the remote control.
In the second subtask, the song notes are gathered from different sources in the internet. Two song notes i.e., Smoke on the Water and Jingle Bells are taken from the website create.Arduino.cc projects hub [9]. 'Star Wars imperial march' is the third song for which the notes are taken from a public website [10]. All these songs are played using a workaround function implemented for the buzzer instead of using the built-in Tone() function.
To do the authentication process with fingerprint touch sensor, Adafruit Fingerprint library is used. As soon this library is imported to the Arduino libraries set, the existing code file namely 'Enroll' is run, to register (create models) the user's fingerprints using the optical touch sensor. Each fingerprint image is stored with an ID value in its library space in the unsigned single byte format (uint8_t) [17]. 'Fingerprint' is another Adafruit library file, when deployed does the fingerprint match process, for the fingers placed on the touch sensor to the stored byte code fingerprints and displays the confidence percentage. Once the registration of user's fingerprints is complete in the library's code file, include the Adafruit_Fingerprint library into the current working project. A software serial port object is initialized with the transmitter and receiver pins on Arduino. This object is used to set up a data rate for the touch sensor serial port (57600). Create a function getfingerName() in this project, to assign a name for the ID's given for corresponding user fingerprint. Also take the function getFingerprintIDez() from 'Fingerprint' file, which is used to do the fingerprint match process when a fingertip is placed on the touch sensor.

Integration of the above sub-tasks code along with the LCD screen display is executed by following a series of button clicks on the remote control are

shown in Fig. 1. In order to emulate the processes of authentication followed by music play options, there is a need for storing the button clicks that were pressed one after the other. After clicking the CH button, the order of the next steps has to be followed accordingly. If the above line up pattern is not followed (if any other buttons are clicked in between), then it would exit this flow process and restricts the user to start from the beginning. RGB LED alters its colours (purple, green, aqua and yellow) subsequently with the changing button choices on remote and according to the current working sensor.

4.4 Display room temperature

An object created for the library DHT class that has built-in functions is invoked to read the values from the sensor. A Boolean type of variable is created to determine whether the system is in the idle or non-idle state. Therefore, when the ultrasonic sensor or flame sensor or touch sensor or IR receiver sensor is working and presenting information on the LCD screen at any given moment, the flag variable (idleFlag) is set to zero. Only these sensors don't have anything to do, then the idle flag is set to one and displays the Temperature and Humidity values on the LCD screen provided by the DHT11 sensor. RGB LED light glows in white colour in the system's idle time. This LED light can be turned on/off by pressing a couple of buttons on remote control followed by an authentication process and is represented in Fig. 2. This operation in code is handled through a flag variable called turnOffLED, where zero is set when the white light is off and set to one when turned on.

4.5 LCD and RGB lights Integration

lthough LCD application in the project was proposed to integrate with the last module, on understanding the complexity and insufficiency of digital pins on a single Arduino board, it was concurrently implemented with each module. LiquidCrystal is the built-in library that has essential pre-determined functions to the display on the LCD screen component. An LC object is created with the parameters of digital pins on Arduino having the wired connections of RS, Enable, D4, D5, D6 and D7 [1]. This object initializes the interface to the screen dimensions (width and height sized are 16 and 2 respectively) of the display in the program Setup () function. Clear() and

print() are the two main handling functions that are called in every module to clear the text on the LCD screen and print a provided text to present on the screen respectively. SetCursor() is a function that is invoked to change the starting point of the cursor so that printing the text on display is changed from the given point on the screen. RGB (Red, Green, Blue) Light emitting diode is a combination of three LED's and different colours can be produced with a varying intensity level configuration (range is 0-255) for each colour from the connected analog pins A3, A4 and A5. A method called setLED-Color() is written to read an analog high value for the three colours by taking the configured intensity values as parameters. This function is called in the other modules, to change the colours on the RGB LED and TurnOffRGB() is another function that reads an analog low from the pins to switch off the light. Either of these two functions is called based on the turnOffLED flag variable set value.

4.6 Problems faced during code Implementation

1. A potentiometer has used to control for contrast display on the LCD screen. When a basic program code of LCD was uploaded on to the Arduino board, there was no text visible on the screen except a glowing light in the background, which is because of power supply.
   Reason: It took some time to understand that, a really very slow and steady movement on the rotatory shaft of the potentiometer has to applied, to have a noticeable text message display on the LCD screen. A minute shift in the wiper arm of the potentiometer can disturb the display with a maximum brightness of texts or total contrast of the screen.

2. When the intruder detection code module was initially coordinated with the LCD display, the LCD showed the texts in some unreadable format that looks like chopped letters in the text message. Sometimes LCD.clear() function did not work and continuously printed the messages on the screen and finally left the LCD screen with no display. Below approaches are followed to identify the cause of the problem.
   a. The addition and deletion of delay functions within the sensor trigger signals nor buzzer sounds did not solve the problem.
   b. Every single line of code in the program was commented to identify the cause of this problem but it was of no benefit.

c. The basic program code of LCD was verified against this combined code of LCD and ultrasonic sensor's module logic.
Solution: The code comparison in the third approach resolved the problem. Through this, it is identified that other than the ultrasonic sensor's logic calculations and the texts used for printing on serial monitor, everything matched in their codes including the delay function implementations. For some reason, when the Serial Monitor display is disabled, then the messages with respect to the ultrasonic sensor are seen on the LCD screen. Thus, resisting the printing of messages to the serial monitor has clarified the problem.

3. Initially, the IR receiver sensor module code was not displaying any received input code from the remote control, after the correct circuit setup. Despite the good working condition and a good battery inside, the remote-control codes were not detected by the IR sensor.
Reason: In the attempts of doing the circuit connection with the IR sensor, the power pin and the ground pin were misinterpreted for one another and connected wrongly. And when the circuit is powered up, the IR sensor got fried inside. All these steps were realized only this IR sensor was replaced with a new one and the button codes of remote-control started displaying on the serial monitor.

4. When the IR sensor code is integrated with the music play submodule, there occurred a compile-time error and it took a long time to understand the problem. The error says, 'In function timer0_pin_port: multiple libraries for IRheader.h and multiple definition of __vector_7'.
Reason: This is because the tone() function is used to play the sounds through the buzzer. This function is invoked from tone.cpp file, which uses a timer variable from its library file that is also present in IRremote file and throws an incompatibility error in the compile time of the program [12] [13]. Below are the alternatives that are found online to resolve the problem.

    - Reinstalled the IRremote library [14]. This approach did not solve the problem.
    - Comment the duplicate timer definition in IRremoteInt.h library [14]. To modify the code, there was no duplicate timer present in the IRremoteInt.h library and commenting the other timer values resulted in multiple compile errors.

- Replace the IRremote header library with the IRremoteControl header library [14]. The functions in the alternate header file looked too complicated to implement in the existing project.
- Use a New Tone standard library/ ToneAC library that has NewTone() function in the place of tone() function [15]. There were different music notes available with these functions and for some reason, they were not at all playing from the buzzer. So, couldn't use this as a working solution.
- Use a workaround function that does the same job as tone() function[15].

Solution: Implementing a workaround function for the tone() function, solved the timer conflicting problem with the IRremote header file. As this solution worked only for three songs, only they were included under the remote-control options leaving the incompatible song notes.

5 REACHING THE MILESTONES

To keep a track of the project implementation, the following milestones were determined are represented in Fig. 3

Milestones 1 and 2 were achieved as per the proposed time. Due to the problems (3 and 4) faced with the IR sensor and music play integration, more time (around 5 days) was utilized to overcome them and the milestones 3 and 4 were extended. During problem resolution time of 3rd milestone, the 5th milestone was met partially by the end of March month and its code was committed to Github before its due time. the 6th milestone was sectionally completed on time leaving the authentication process which is related to the touch sensor. Because, the fingerprint touch sensor was ordered from a distant vendor in an online store (as this is a rare item, it was not available under prime subscription in Amazon.ca) and took longer delivery time i.e., on 8th April. By the stretch goal time, i.e., April 9th, the touch sensor's authentication and RGB LED was integrated and tested with the developed modules and met the milestone 7 on time.

6 FUTURE WORK

There can be the inclusion of more music notes for different songs, to have wide choices for multiple authenticated users in the system. The existing Arduino system benefits only the users who are present in the house where the system is equipped. In order to extend its accessibility, a Wi-Fi module can be implemented, so that the sensor information can be sent to a remote server and make the information accessible from anywhere around the world for permitted users in the future.

7 CONCLUSION

The basic necessities in terms of protection and entertainment are well addressed by this Arduino based smart home device. The block architecture of this system followed the principle of information system. As proposed, a moderate level of protection and safety can be guaranteed for a house that cannot afford a costly surveillance and alarm system.

A Diagrams and Pictures

B Source Code

Link to Smart Home Management Git hub repository:
https://github.com/KeerthiMettu/Smart-Home-Management

References

[1] "Arduino LCD Tutorial — How To Connect an LCD to Arduino." HowToMechatronics, 4 Aug. 2018, howtomechatronics.com/tutorials/arduino/lcd-tutorial/. Accessed 1 Apr. 2019

[2] "Intrusion Detection System Using Arduino." Arduino Project Hub, Arduino Project Hub, 2018,

create.arduino.cc/projecthub/theSTEMpedia/intrusion-detection-system-using-arduino-e31caa?ref=tagref$_id$ = $securityoffset$ = 132.$Accessed$13Apr.2019.

[3] "How To Use An Ultrasonic Sensor With An Arduino." Arduino Project Hub, Arduino Project Hub, 2017, create.arduino.cc/projecthub/jake/how-to-use-an-ultrasonic-sensor-with-an-arduino-63527b. Accessed 13 Apr. 2019.

[4] Kumar, Shashi. "Arduino Flame Sensor Interfacing to Build a Fire Alarm System." Circuitdigest.Com, 2 Aug. 2018, circuitdigest.com/microcontroller-projects/arduino-flame-sensor-interfacing. Accessed 29 Mar. 2019.

[5] Instructables. "Arduino Infrared Remote Tutorial." Instructables, Instructables, 5 Nov. 2013, www.instructables.com/id/Arduino-Infrared-Remote-tutorial/. Accessed 1 Apr. 2019.

[6] "Complete Guide for Ultrasonic Sensor HC - SR04." Random Nerd Tutorials, 2019, randomnerdtutorials.com/complete-guide-for-ultrasonic-sensor-hc-sr04/. Accessed 12 Apr. 2019.

[7] Team, Rhydo. "Flame Sensor - 5mm IR Diode[U+202F]: RhydoLABZ INDIA." RhydoLABZ.Com, 2016, www.rhydolabz.com/sensors-ir-pir-sensors-c-137$_1$50/$flame - sensor - 5mm - ir - diode - p -$ 2263.$html$.$Accessed$13Apr.2019.

[8] Dejan. "DHT11 DHT22 Sensor Temperature and Humidity Tutorial." HowToMechatronics, 21 Oct. 2018, howtomechatronics.com/tutorials/arduino/dht11-dht22-sensors-temperature-and-humidity-tutorial-using-arduino/. Accessed 14 Apr. 2019.

[9] https://josephmilla.com/about. "Playing Tones With Piezoelectric Speakers." Joseph Milla, 27 May 2013, josephmilla.com/2013/05/27/playing-tones-with-piezoelectric-speakers/. Accessed 17 Apr. 2019.

[10] Marinis, Michael. "Newbie Music Player." Arduino Project Hub, Arduino Project Hub, 4 Sept. 2016,

create.arduino.cc/projecthub/Michael$_M arinis/newbie-music-player-372dca.Accessed17Apr.2019.

[11] Shirriff, Ken. "IRremote Library, Send Receive Infrared Remote Control." Pjrc.Com, 2019, www.pjrc.com/teensy/td$_l ibs_I Rremote.html.Accessed2Apr.2019.

[12] Grant, Bruce. "Arduino Timer Conflicts." Wbga.Ca, 21 Aug. 2016, www.wbga.ca/19/. Accessed 5 Apr. 2019.

[13] c) 2004-2019, Computences. "Arduino NewPing, IRremote and Timers." Computences.Com, 5 Aug. 2014, www.computences.com/blog/2014-08-05-Arduino-tricks. Accessed 5 Apr. 2019.

[14] "IRRemote Library Conflicts with Tone() Function." Arduino.Cc, 2012, forum.arduino.cc/index.php?topic=120955.0. Accessed 12 Apr. 2019.

[15] "NewTone Library - Plug-in Replacement for Tone Library - Better, Smaller, Faster." Arduino.Cc, 2013, forum.arduino.cc/index.php?topic=143940.0. Accessed 12 Apr. 2019.

[16] "Adafruit Customer Service Forums • View Topic – Fingerprint Question." Adafruit.Com, 2016, forums.adafruit.com/viewtopic.php?f=22t=33229start=195. Accessed 17 Apr. 2019.

[17] "Fingerprint Sensor Module with Arduino — Random Nerd Tutorials." Random Nerd Tutorials, 26 Jan. 2016, randomnerdtutorials.com/fingerprint-sensor-module-with-arduino/. Accessed 15 Apr. 2019.

Button on remote control	**Action specified in the background**	**LCD Display**
CH		Loading... Playing
PLAY/PAUSE	Fingerprint match process to authenticate the user operating and it	*Before Authentication:* Get ready... Place your finger *After Authenctication:* Hello...<User name>/Unauthorized access
VOL +	Display the list of songs available on 1,2 and 3 buttons in scrolling pattern for authenticated users.	*For authenticated users:* Smoke on the water 1 Jingle Bells 2 Star Wars 3 *For non-authenticated users:* Button pressed is VOL +
1	Plays the 'Smoke on the water' song	*For authenticated users:* Playing track 1 *For non-authenticated users:* Button pressed is 1
2	Plays the 'Jingle bells' song	*For authenticated users:* Playing track 2 *For non-authenticated users:* Button pressed is 2
3	Plays the 'Star Wars' song	*For authenticated users:* Playing track 3 *For non-authenticated users:* Button pressed is 3
EQ	Exits from the authenticated mode	*For authenticated users:* Music track Exiting . . . *For non-authenticated users:* Button pressed is EQ

Figure 1: Actions performed for the remote control buttons to play music

Button on the remote control	**The action specified in the background**	**LCD Display when light is on**	**LCD Display when light is off**
CH+	checks the state of LED	Turn off LED? Yes: NEXT	Turn on LED? Yes: PREV
NEXT	Only when CH+ button is pressed before: Fingerprint match process to authenticate the user operating and if authenticated turns OFF the LED	*Before Authentication:* Get ready… Place your finger *After Authentication:* Hello…<User name> Turning off the LED /Unauthorized access	Button pressed is NEXT
PREV	Only when CH+ button is pressed before: Fingerprint match process to authenticate the user operating and if authenticated turns ON the LED	Button pressed is PREV	*Before Authentication:* Get ready… Place your finger *After Authentication:* Hello…<User name> Turning on the LED /Unauthorized access

Figure 2: Actions performed for the remote control buttons to turn of RGB light

Milestone 1	March 8	Gathering and assembling required components
Milestone 2	March 13	Complete user intrusion detection module and identifying the humidity and temperature info from sensor
Milestone 3	March 21	Coding to use the IR Receiver and play sounds using remote control
Milestone 4	March 26	Complete alarm system for fire detection module
Milestone 5	April 2	Integrate an LCD that connects with all modules
Milestone 6	April 5	Complete demo of the working system
Milestone 7	April 9	Stretch goal: Integrate an RGB LED to Arduino board to identifying the working sensors at any given moment

Figure 3: List of proposed milestones

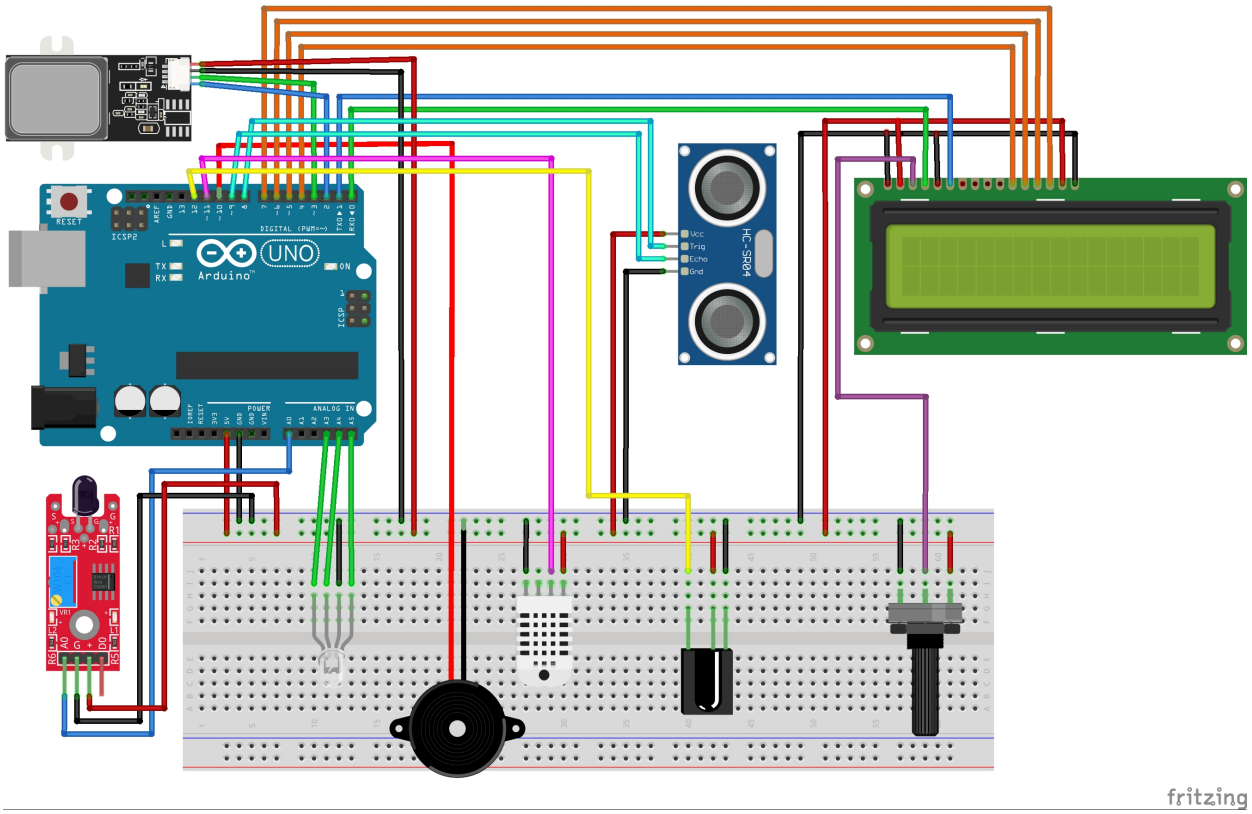


Figure 4: Circuit diagram of the project

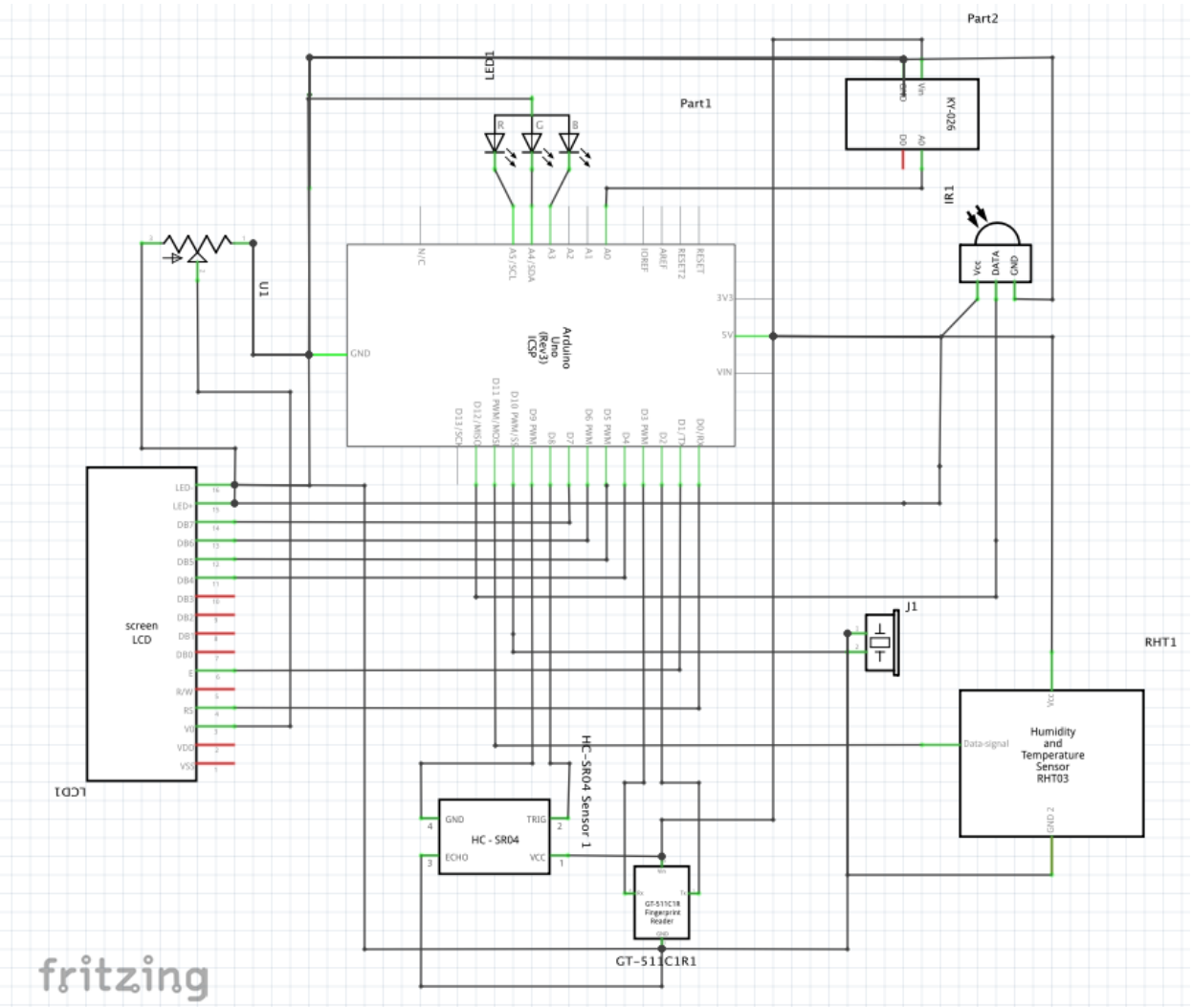


Figure 5: Schematic connections of the project

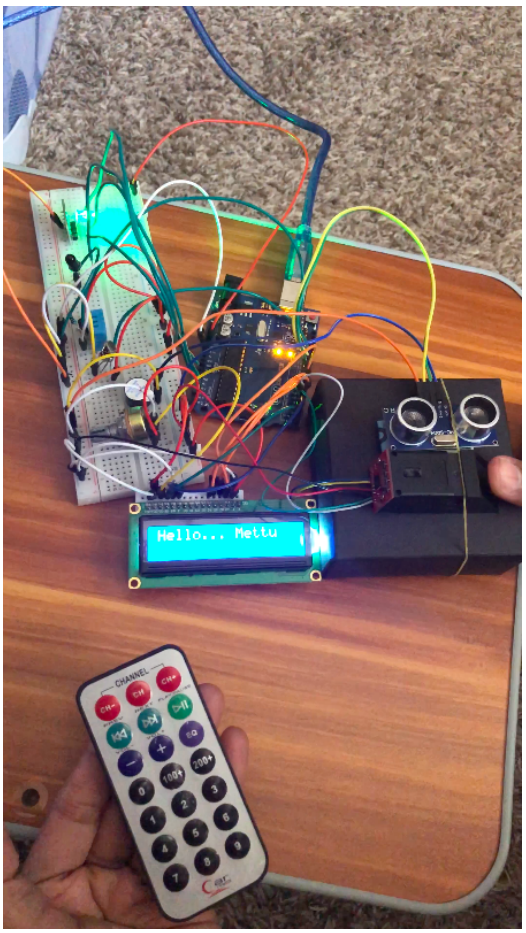


Figure 6: Shows a message on LCD

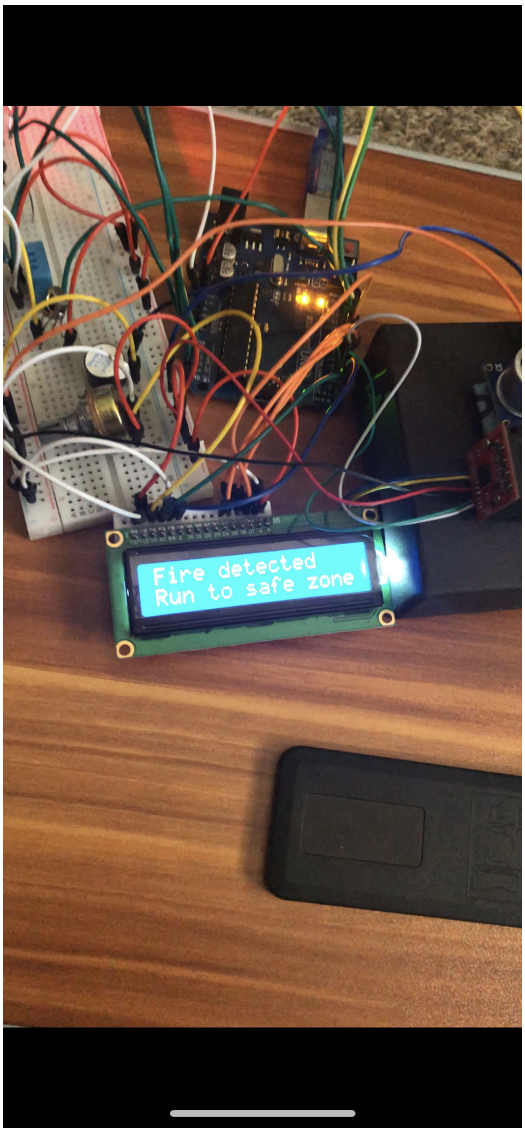


Figure 7: Shows a message on LCD

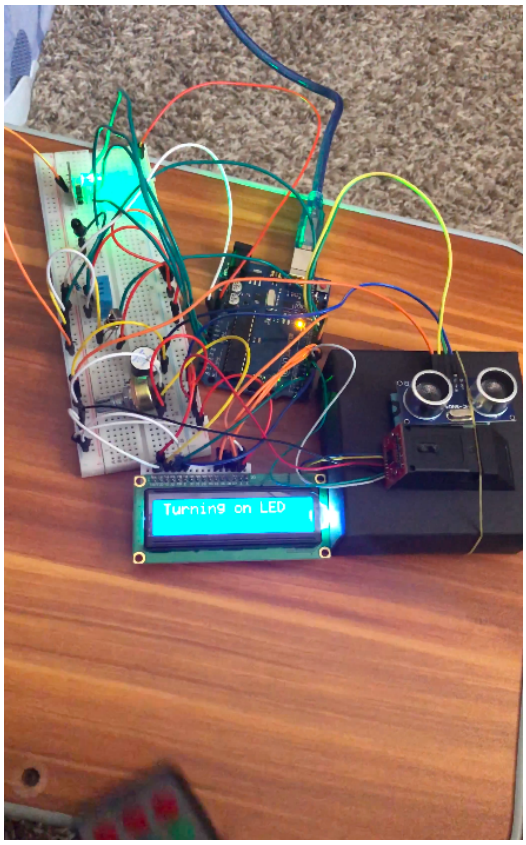
on LED.png 

Figure 8: Shows a message on LCD