CHAPTER 1

INTRODUCTION

We live in the 21st century where everything is preferred to be fast contactless and automated. During this covid situation, we are taking a lot of precautions to save ourselves from coronavirus, but there are high chances of carelessness which might lead to consequences. The idea behind this project was basically to reduce the spread of the virus, when someone enters our home the first thing that encounters the person is our doorbell, and it's not always possible to sanitize everything around us. So why not transform our traditional bell into a contactless bell, which eliminates the contact and hence reduces the chances of spread.

1.1 Social Distancing & Covid-19 Pandemic

Social distancing is the one of the best methods to escape from COVID-19. But we can't avoid some emergency visits to some homes. When we arrived at in front of a house, first we search the doorbell button/ calling bell button. And press the button. But in this special situation this doorbell button can cause the virus to spread.

1.2 Contactless belling

The contactless doorbell uses a raspberry pi controller to work along with a speaker module and buzzer to perform automatic operations. This system will help homeowner know who has arrived at his/her door as well as it will act as a security system when the owner is not at home to alert on any robbery.

1.3 Brief history on intercom system

The history of the intercom system is started a long time ago, already hundred years ago people want to notice the guests arrive at their house. it all started when the doorbell and electricity are introduced. In the middle of the twentieth century, the doorbell system introduced a button, transformer and wiring which allows us to place the bell to a remote location. in World War 2 the doorbell system is accelerated. New functions were added and even became wireless. The most important thing is to allow the guest communication without going there. It began with utilizing the analogue communication system.

Then the development went fast forward and nowadays we call the new system door phones instead of doorbells.

1.4 Role of the sensors in Covid-19 pandemic

The Covid-19 pandemic has proven to be the most disastrous pandemic in the history. Millions of people have lost their lives sending nations into lockdown and economic slowdowns. Given the fact that no specific anti-viral treatment is yet suggested for treating Covid-19 infection, “Social distancing” is probably the most effective tool so far in stopping the virus spread. This paper has proposed an IoT based doorbell which alerts the house owner about arrival of a visitor having fever and who could be a Covid-19 patient. The system uses NodeMCU and MLX90614 non-contact infrared temperature sensor. FireBase online database is used to log all the readings of the system and a companion mobile App is also provided. The system was extensively tested using an experimental set up under various conditions. The system works with 99% average accuracy of body temperature measurement.

During this covid situation, we are taking a lot of precautions to save ourselves from coronavirus, but there are high chances of carelessness which might lead to consequences. The idea behind this project was basically to reduce the spread of the virus, when someone enters our home the first thing that meets that person is our doorbell, and it's not always possible to sanitize everything around us. So why not transform our traditional bell into a contactless bell, which eliminates the contact and hence reduces the chances of spread.

Contactless door-bell system with sensor-based technology to reduce the spread of contagious diseases like Covid-19.

1.5 Objectives

1. To find an appropriate sensor for detection of human.
2. To design the touchless doorbell using the selected sensor.
3. To deploy the touchless doorbell.
4. To test the working of the touchless doorbell.

Chapter 2

Software Requirement Specification

A System Requirements Specification (SRS)) is a document or set of documentation that describes the features and behaviour of a system or software application. It includes a variety of elements that attempts to define the intended functionality required.

FUNCTIONAL REQUIREMENTS

A functional requirement defines a function of a system or its component. Where a function is described as a specification of behaviour between inputs and outputs.

NON-FUNCTIONAL REQUIREMENTS

Software requirement can be non-functional and be a performance requirement. Non-functional requirements are the characteristics or attributes of the system that can judge its operation.

Hardware requirements

1. Raspberry Pi PCB.
2. Infrared (IR) module.
3. Ultrasonic sensor.
4. LDR sensor.
5. Breadboard.
6. Jumper wires.
7. Buzzer

Software requirements

1. Raspberry Pi OS.
2. Python3
3. Thonny (Python IDE)

2.1 Raspberry Pi Board

Fig 2.1 Raspberry Pi Board

The Raspberry pi is a single computer board with credit card size, that can be used for many tasks that your computer does, like games, word processing, spreadsheets and to play HD video. It was established by the Raspberry pi foundation from the UK. It has been ready for public consumption since 2012 with the idea of making a low-cost educational microcomputer for students and children. The main purpose of designing the raspberry pi board is, to encourage learning, experimentation, and innovation for school level students. The raspberry pi board is a portable and low cost. Maximum of the raspberry pi computers is used in mobile phones. In the 20th century, the growth of mobile computing technologies is very high, a huge segment of this being driven by the mobile industries. The 98% of the mobile phones were using ARM technology.

The raspberry pi comes in two models, they are model A and model B. The main difference between model A and model B is USB port. Model A board will consume less power and that does not include an Ethernet port. But the model B board includes an Ethernet port and designed in China. The raspberry pi comes with a set of open-source technologies, i.e., communication and multimedia web technologies. In the year 2014, the foundation of the raspberry pi board launched the computer module, that packages a model B raspberry pi board into module for use as a part of embedded systems, to encourage their use.

2.2 IR sensor Working

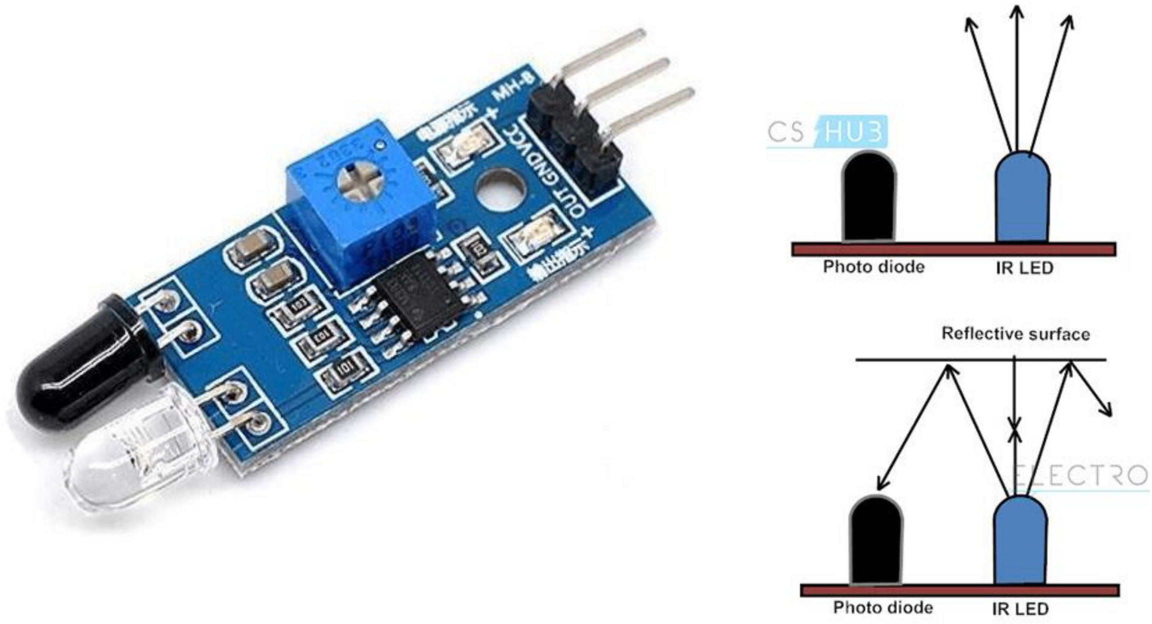

Fig 2.2 IR Sensor

IR sensor is an electronic device, that emits the light to sense some object of the surroundings. An IR sensor can measure the heat of an object as well as detects the motion. Usually, in the infrared spectrum, all the objects radiate some form of thermal radiation. These types of radiations are invisible to our eyes, but infrared sensor can detect these radiations.

The emitter is simply an IR LED (Light Emitting Diode) and the detector is simply an IR photodiode. Photodiode is sensitive to IR light of the same wavelength which is emitted by the IR LED. When IR light falls on the photodiode, the resistances and the output voltages will change in proportion to the magnitude of the IR light received.

There are five basic elements used in a typical infrared detection system: an infrared source, a transmission medium, optical component, infrared detectors or receivers and signal processing. Infrared lasers and Infrared LEDs of specific wavelength used as infrared sources.

The three main types of media used for infrared transmission are vacuum, atmosphere and optical fibres. Optical components are used to focus the infrared radiation or to limit the spectral response.

It is universal that black colour absorbs the entire radiation incident on it and white colour reflects the entire radiation incident on it. Based on this principle, the second positioning of the sensor couple can be made. The IR LED and the photodiode are placed side by side. When the IR transmitter emits infrared radiation, since there is no direct line of contact between the transmitter and receiver, the emitted radiation must reflect to the photodiode after hitting any object. The surface of the object can be divided into two types: reflective surface and non-reflective surface. If the surface of the object is reflective in nature i.e., it is white or other light colour, most of the radiation incident on it will get reflected and reaches the photodiode. Depending on the intensity of the radiation reflected, current flows in the photodiode.

If the surface of the object is non-reflective in nature i.e., it is black or other dark colour, it absorbs almost all the radiation incident on it. As there is no reflected radiation, there is no radiation incident on the photodiode and the resistance of the photodiode remains higher allowing no current to flow. This situation is like there being no object at all.

2.3 Ultrasonic sensor

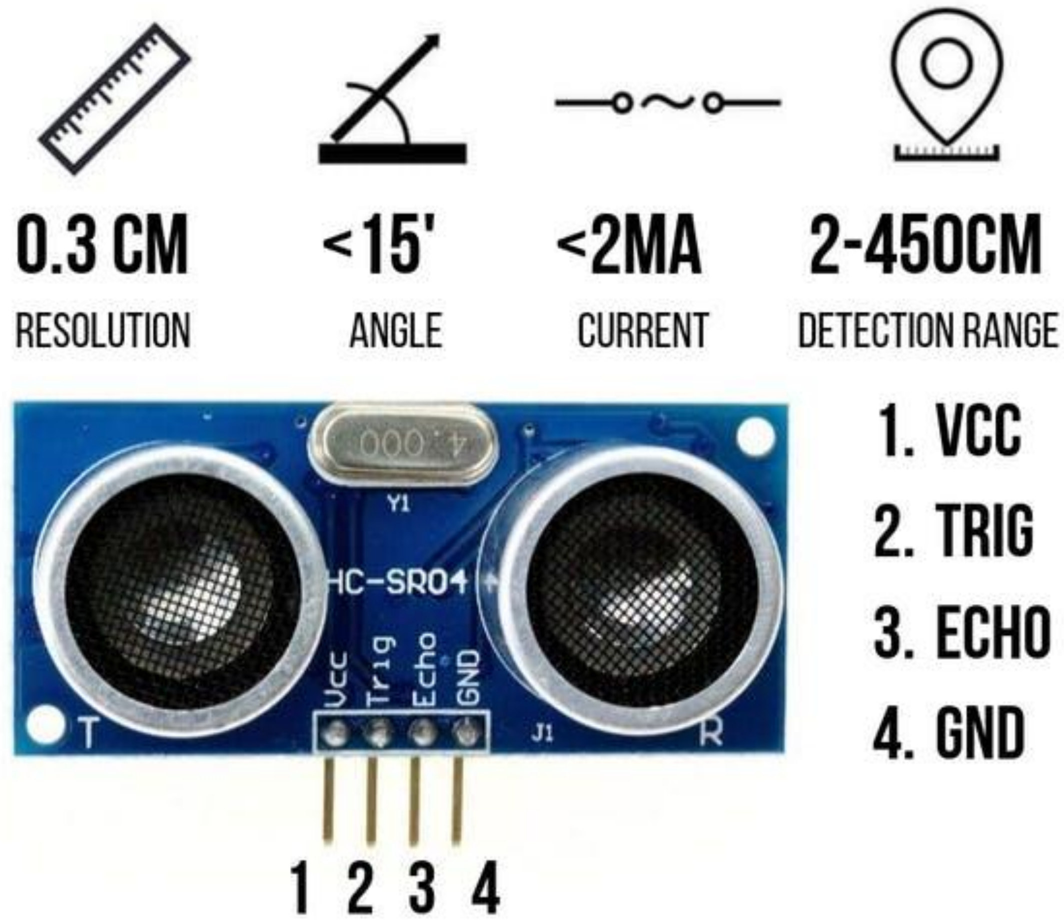

Fig 2.3 Ultrasonic sensor

An ultrasonic sensor is an instrument that measures the distance to an object using ultrasonic sound waves.

An ultrasonic sensor uses a transducer to send and receive ultrasonic pulses that relay back information about an object's proximity.

High-frequency sound waves reflect from boundaries to produce distinct echo patterns.

Ultrasonic sensors work by sending out a sound wave at a frequency above the range of human hearing. The transducer of the sensor acts as a microphone to receive and send the ultrasonic sound. ultrasonic sensors, like many others, use a single transducer to send a pulse and to receive the echo. The sensor determines the distance to a target by measuring time lapses between the sending and receiving of the ultrasonic pulse.

The working principle of this module is simple. It sends an ultrasonic pulse out at 40kHz which travels through the air and if there is an obstacle or object, it will bounce back to the sensor. By calculating the travel time and the speed of sound, the distance can be calculated.

2.4 LDR sensor

Fig 2.4 LDR sensor

The **Light Dependent Resistor** (LDR) is made from a piece of exposed semiconductor material such as cadmium sulphide that changes its electrical resistance from several thousand Ohms in the dark to only a few hundred Ohms when light falls upon it by creating hole-electron pairs in the material.

The net effect is an improvement in its conductivity with a decrease in resistance for an increase in illumination. Also, photo resistive cells have a long response time requiring many seconds to respond to a change in the light intensity.

Materials used as the semiconductor substrate include, lead sulphide (PbS), lead selenide (PbSe), indium antimonide (InSb) which detect light in the infra-red range with the most used of all photo-resistive light sensors being **Cadmium Sulphide** (Cds).

Cadmium sulphide is used in the manufacture of photoconductive cells because its spectral response curve closely matches that of the human eye and can even be controlled using a simple torch as a light source. Typically, then, it has a peak sensitivity wavelength (λp) of about 560nm to 600nm in the visible spectral range.

The most used photo resistive light sensor is the **ORP12** Cadmium Sulphide photoconductive cell. This light dependent resistor has a spectral response of about 610nm in the yellow to orange region of light. The resistance of the cell when unilluminated (dark resistance) is very high at about 10MΩ's which falls to about 100Ω's when fully illuminated (lit resistance).

2.5 Breadboard

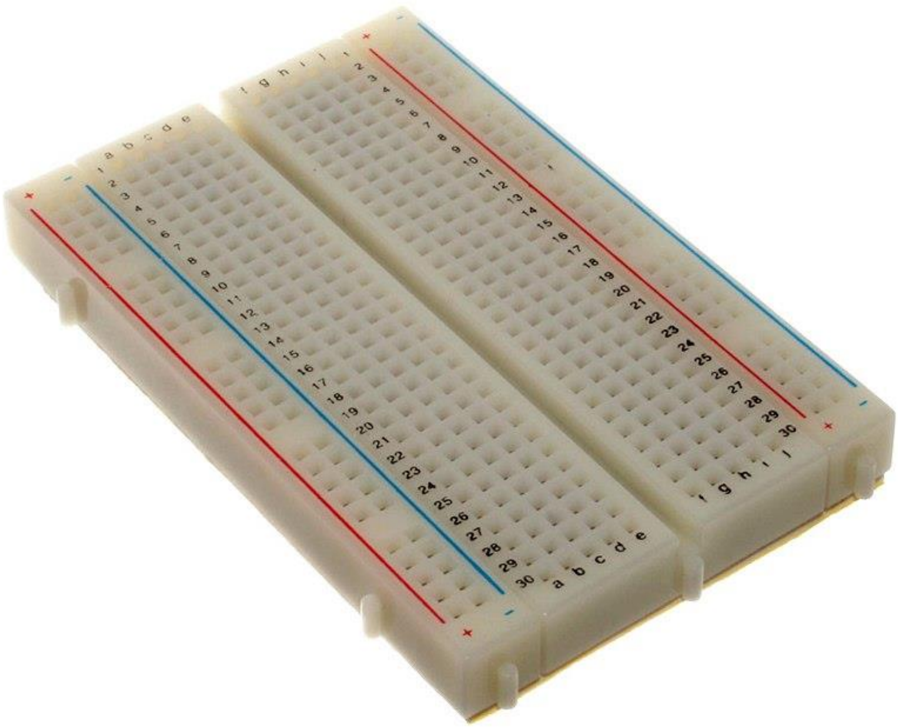

Fig 2.5 Breadboard

A breadboard, or protoboard, is a construction base for prototyping of electronics. Originally the word referred to a literal bread board, a polished piece of wood used when slicing bread. In the 1970s the solderless breadboard became available and nowadays the term "breadboard" is commonly used to refer to these.

Because the solderless breadboard does not require soldering, it is reusable. This makes it easy to use for creating temporary prototypes and experimenting with circuit design. For this reason, solderless breadboards are also popular with students and in technological education. Older breadboard types did not have this property. A stripboard (Veroboard) and similar prototyping printed circuit boards, which are used to build semi-permanent soldered prototypes or one-offs, cannot easily be reused. A variety of electronic systems may be prototyped by using breadboards, from small analogue and digital circuits to complete central processing units (CPUs).

2.6 Jumper wires

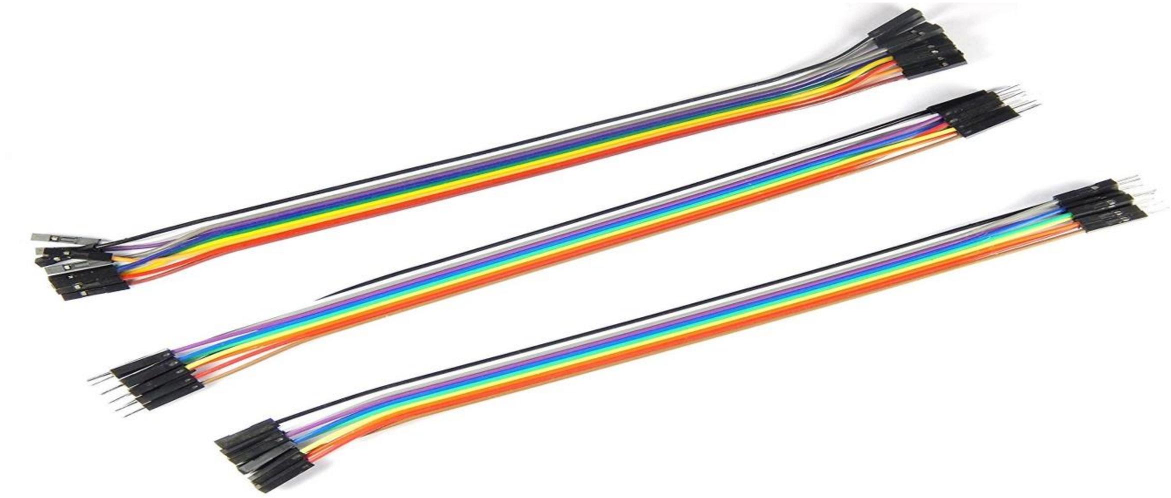

Fig 2.6 Jumper wires

A jump wire (also known as jumper, jumper wire, jumper cable, DuPont wire or cable) is an electrical wire, or group of them in a cable, with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.

There are different types of jumper wires. Some have the same type of electrical connector at both ends, while others have different connectors.

Some common connectors are:

- Solid tips – are used to connect on/with a breadboard or female header connector. The arrangement of the elements and ease of insertion on a breadboard allows increasing the mounting density of both components and jump wires without fear of short-circuits. The jump wires vary in size and colour to distinguish the different working signals.
- Crocodile clips – are used, among other applications, to temporarily bridge sensors, buttons and other elements of prototypes with components or equipment that have arbitrary connectors, wires, screw terminals, etc.
- Banana connectors – are commonly used on test equipment for DC and low-frequency AC signals
- RF connectors – are used to carry radio frequency signals between circuits, test equipment, and antennas

RF jumper cables - Jumper cables are a smaller and more bendable corrugated cable which is used to connect antennas and other components to network cabling. Jumpers are also used in base stations to connect antennas to radio units. Usually, the most bendable jumper cable diameter is 1/2"

2.7 Buzzer

Fig 2.7 Buzzer

A **buzzer** or **beeper** is an audio signalling device, which may be mechanical, electromechanical, or piezoelectric (*piezo* for short). Typical uses of buzzers and beepers include alarm devices, timers, and confirmation of user input such as a mouse click or keystroke.

An audio signalling device like a beeper or buzzer may be electromechanical or piezoelectric or mechanical type. The main function of this is to convert the signal from audio to sound. Generally, it is powered through DC voltage and used in timers, alarm devices, printers, alarms, computers, etc. Based on the various designs, it can generate different sounds like alarm, music, bell & siren.

2.8 Thonny Python IDE

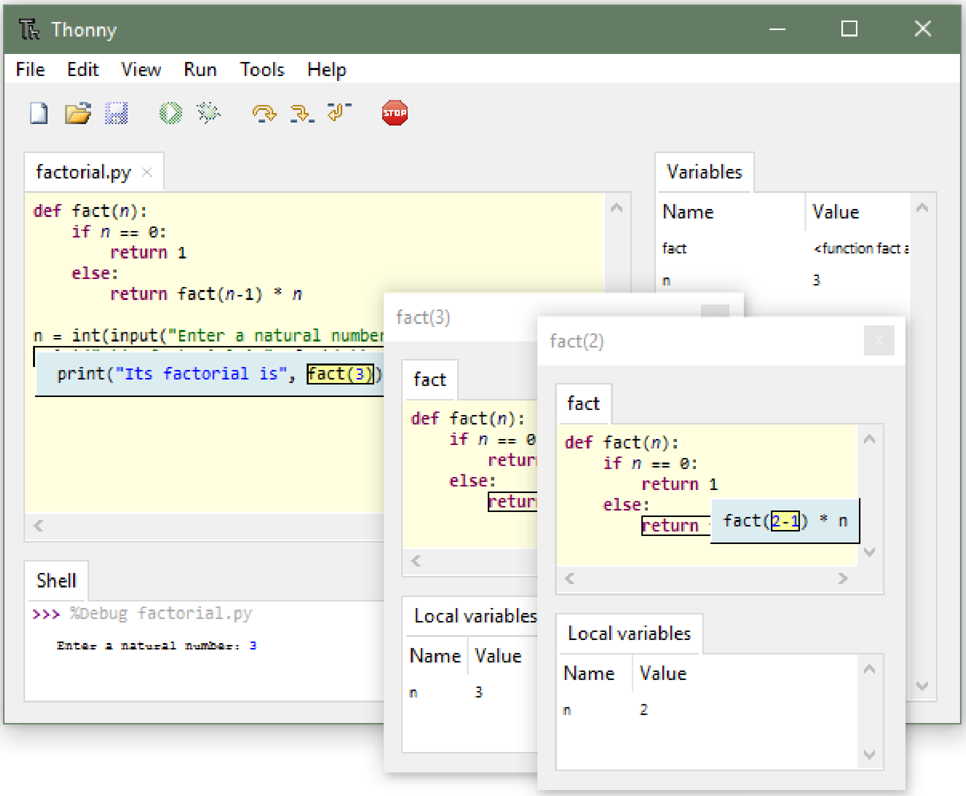

Fig 2.8 ThonnyPython IDE

Thonny is an integrated development environment for Python that is designed for beginners. It supports different ways of stepping through the code, step-by-step expression evaluation, detailed visualization of the call stack and a mode for explaining the concepts of references and heap.

The program works on Windows, macOS and Linux. It is available as binary bundle including the recent Python interpreter or pip-installable package. It can be installed via the operating-system package manager on Debian, Raspberry Pi, Ubuntu, and Fedora.

Chapter 3

Analysis and Design

3.1 Flowchart:

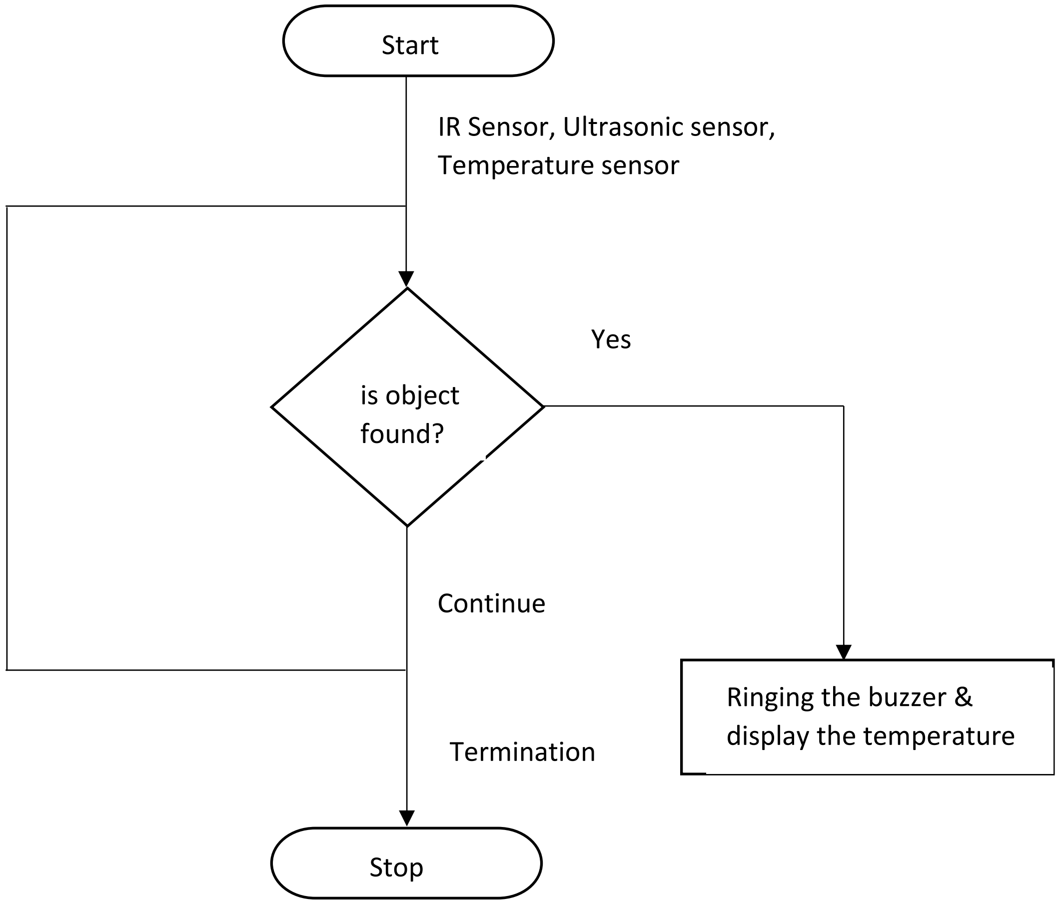

Fig 3.1 Flowchart

Ultrasonic Sensor senses the object surrounded by it. If an object is found, it sends the signal to the Raspberry Pi to turn on the IR as well as temperature sensors and then, the if an object comes nearby IR sensor (i.e., sensing area of IR) then it sends a signal to the board to trigger the Buzzer. But, if the object is not found then, the sensing process continues until we terminate the execution explicitly. Hence, providing an environment for the application as a contactless doorbell.

Chapter 4
Implementation

4.1 Packages used:

Raspberry Pi has lots of applications already installed, like Minecraft, Mathematica, and the browser. And there are a lot of 'libraries'. Linux programmers don't like writing code when they can use code that other people have already written and tested, so there are lots and lots of 'libraries' that have been written and shared by applications.

The Raspberry Pi Foundation has created the 'Pi Store', like the Play Store or App Store on your phone. You'll find it under the Internet Menu when you're running the GUI on the Raspberry Pi. There are a few applications that have been written by the Raspberry Pi community, but this is only a small selection of what is available.

There are two ways to install more applications, or 'packages' as they are called in the Linux world. One is via the command line ('apt-get', in a terminal window), while the other has a user interface. We're going to do the latter using 'synaptic'.

Some of the important libraries used in our application are as follows:

- **GPIO:**

  A simple interface to GPIO devices with Raspberry Pi, developed and maintained by Ben Nuttall and Dave Jones. A general-purpose input/output (GPIO) is an uncommitted digital signal pin on an integrated circuit or electronic circuit board which may be used as an input or output, or both, and is controllable by the user at runtime.

GPIOs have no predefined purpose and are unused by default. If used, the purpose and behaviour of a GPIO is defined and implemented by the designer of higher assembly-level circuitry: the circuit board designer in the case of integrated circuit GPIOs, or system integrator in the case of board-level GPIOs.

- **w1thermsensor:**

Get the temperature from your w1therm sensor in a single line of code! It's designed to be used with the Raspberry Pi hardware but also works on a Beagle Bone and others. Raspberry Pi: this package is available in Raspbian as python-w1thermsensor and python3-w1thermsensor. Python 2 drop: all w1thermsensor releases from 2.0 are Python 3.5+.

- **time:**

This module provides various time-related functions. For related functionality, see also the datetime and calendar modules.

Although this module is always available, not all functions are available on all platforms. Most of the functions defined in this module call platform C library functions with the same name. It may sometimes be helpful to consult the platform documentation because the semantics of these functions varies among platforms.

4.2 Source code:

```
#Libraries
import RPi.GPIO as GPIO
import time

#GPIO Mode (BOARD / BCM)
GPIO.setmode(GPIO.BCM)

#Set Buzzer
GPIO.setup(3,GPIO.OUT,initial=0)

#set GPIO Pins
GPIO_TRIGGER = 18
GPIO_ECHO = 24
GPIO.setup(23, GPIO.IN) #PIR
#set GPIO direction (IN / OUT)
GPIO.setup(GPIO_TRIGGER, GPIO.OUT)
GPIO.setup(GPIO_ECHO, GPIO.IN)

def distance():
    # set Trigger to HIGH
    GPIO.output(GPIO_TRIGGER, True)

    # set Trigger after 0.01ms to LOW
    time.sleep(0.00001)
    GPIO.output(GPIO_TRIGGER, False)

    StartTime = time.time()
    StopTime = time.time()

    # save StartTime
    while GPIO.input(GPIO_ECHO) == 0:
```

```
        StartTime = time.time()

    # save time of arrival
    while GPIO.input(GPIO_ECHO) == 1:
        StopTime = time.time()

    # time difference between start and arrival
    TimeElapsed = StopTime - StartTime
    # multiply with the sonic speed (34300 cm/s)
    # and divide by 2, because there and back
    distance = (TimeElapsed * 34300) / 2

    return distance

if __name__ == '__main__':
    try:
        while True:
            dist = distance()
            print ("Person is at Distance of = %.1f cm" % dist)
            time.sleep(1)
            while dist<50:
                    if GPIO.input(23):
                        print('Buzzer is off')
                        GPIO.output(3,GPIO.LOW)
                        time.sleep(1)
                    else:
                        print('Buzzer is on ')
                        GPIO.output(3,GPIO.HIGH)
                        time.sleep(1)

        # Reset by pressing CTRL + C
    except KeyboardInterrupt:
        print("Stopped by User")
        GPIO.cleanup()
```

#Temperature sensor

```
import time

from w1thermsensor import W1ThermSensor

sensor = W1ThermSensor()

while True:

    temperature = sensor.get_temperature()

    print("The temperature is %s celsius" % temperature)

    time.sleep(1)
```

4.3 Snapshots

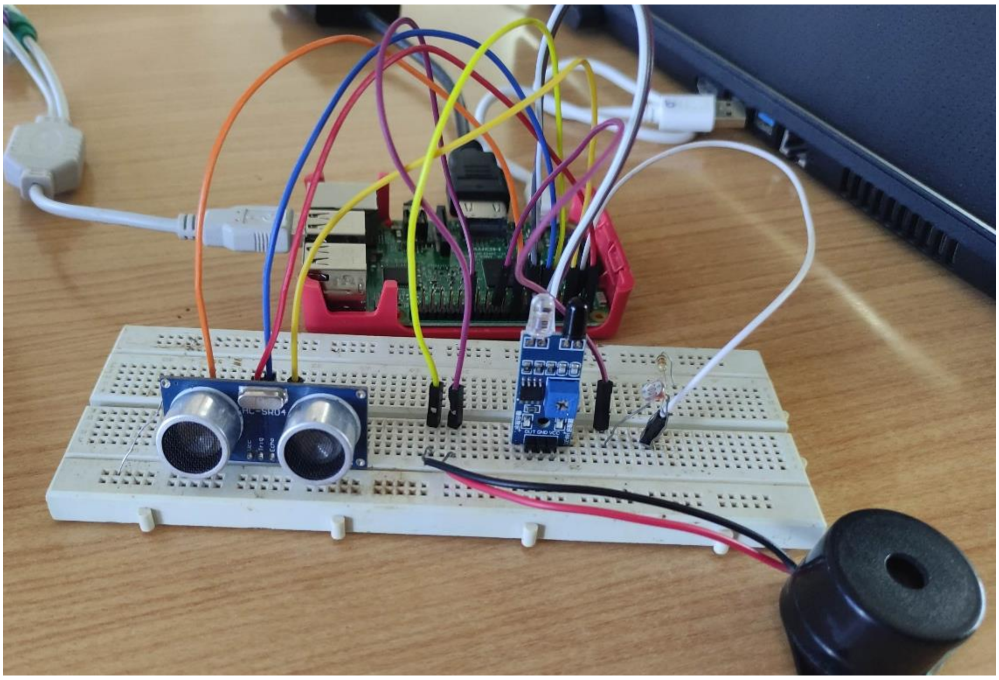

Fig 4.3 Snapshots(1)

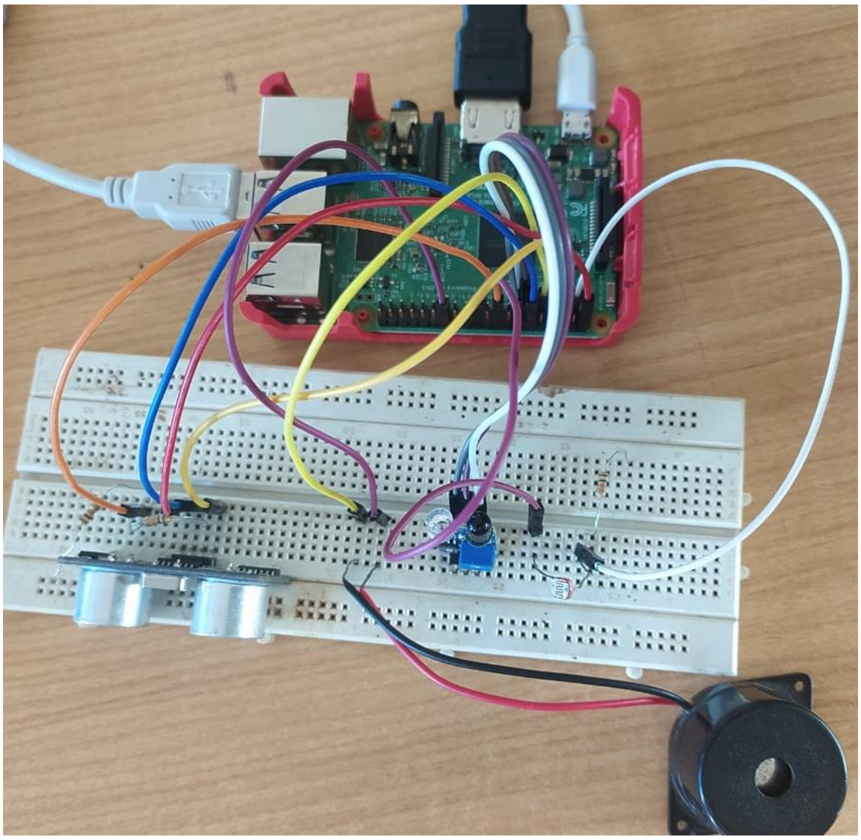

Fig 4.3 Snapshots(2)

```
Distance: 5 cm
Ringing the bell!
The temperature is 39 celsius
```

Fig 4.3 Snapshots(3)

Chapter 5

Testing

The developer has performed testing under various cases and the outcomes of test cases are recorded and listed below:

Scenario	Test Type	Outcome
Under different weather conditions	Sensor test	Works perfectly in all weather conditions
When visitor come closer to the sensor	Visitor Arrival Test	IR and Temperature sensor turns on
When visitor places hand in front of the IR and temperature sensor	Visitor temperature test	Buzzer rings and the temperature is displayed

Table 5.1 Test cases

Every module has been implemented with a structured workflow using optimized coding. First the doorbell is connected to the bread board using cables; the doorbell acts a push button. After the doorbell is connected the connectivity of the device is checked using raspberry pi functions. Once the doorbell is connected, scenarios and actions are added to the push box services, the scenarios and actions help in pushing the buzzer notification to the device.

Chapter 6
Conclusion

The project named " Contactless Doorbell" has been designed with the domain as Internet of Things. The basic concepts and working of IOT has been displayed in the running of the project. The project uses mainly a raspberry pi Board and python programming concept. Since, today, in a technologically enhancing environment, security issues are of at most concern, this project shows how technology can be used to enhance the security features of people's homes. A doorbell is constructed which has the feature to ring a buzzer to the owner.