

HUMAN FALL DETECTION USING ARDUINO AND BLUETOOTH

Human Fall Detection Using Aurdino And Bluetooth

Submitted in partial fulfillment of the requirements for the award of Bachelor of
Engineering Degree in Electrical and Electronics Engineering

by

NARMADHA E (39140033)

SNEHA D (39140053)

SNOWSHA J (39140054)



DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING
SCHOOL OF ELECTRICAL AND ELECTRONICS ENGINEERING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600119



SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

Ko

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of NARMADHA (Reg. No. 39140033), SNEHA D (Reg. No. 39140053) and SNOWSHA J(Reg.No.39140054)who carried out the project entitled "**HUMAN FALL DETECTION USING ARDUINO AND BLUETOOTH**"under our supervision from November 2021 to May 2022.

Internal Guide

Mrs.RAMYA D

**Dr. V. SIVACHIDAMBARANATHAN
Head of the Department**

Submitted for Viva voce Examination held on_____

Internal examiner.

External Examiner:



SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr. N. M. NANDHITHA, M.E, Ph.D.**, Dean, School of Electrical and Electronics and Head of the Department. for providing us necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our Project Guide **Mrs.RAMYA D**for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of our project work.

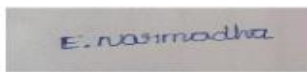
We wish to express our thanks to all Teaching and Non-teaching staff members of the Department of **Electrical and Electronics Engineering** who were helpful in many ways for the completion of the project.

I

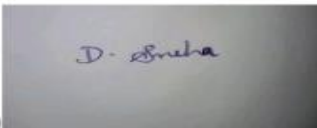
DECLARATION

We, **NARMDHA E (39140033)** , **SNEHA D (39140053)** and **SNOWSHA J(39140054)** hereby declare that the Project Report entitled "**HUMAN FALL DETECTION USING ARDUINO AND BLUETOOTH**" done by under the guidance of **Mrs.RAMYA D** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Electrical and Electronics Engineering.

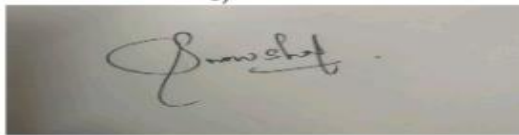
1)



2)



3)



3

DATE:
PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

TABLE OF CONTENTS

S.No.	Contents	Page No
1	Abstract	2
2	Introduction	3
3	Overview	4
4	Objective	5
5	Literature survey	6
6	Existing system	9
7	Proposed system	9
8	Existing block diagram	10
9	Proposed block diagram	11
10	Hardware & Software requirements and Hardware design	12
11	Overview	13

12	Summary	14
13	Arduino software	33
14).	Output.	35
15).	Result.	42

ABSTRACT

Falling down is among the most common causes of medical attention required by the elderly people. Elderly people often injure themselves from falling down more especially when they are living alone. After a fall occurs, medical attention needs to be provided promptly in order to reduce the risk of victim. Several technologies have been developed which utilize webcams to monitor the activities of elderly people. However, the cost of operation and installation is expensive and only applicable for indoors... Current commercialized device required user to wear wireless emergency transmitter in form of wristwatch. This method will restrict the user movement and produce high false alarm due to frequent swinging and movement of the device.

INTRODUCTION

Human falls are considered as dreaded events since they can affect a person physically as well as psychologically. Compared to younger people, older people have a greater risk of falling (Duque, 2016). As the age of a human being advances, his body becomes physically weak which increases the risk to fall. By 2030, the worldwide population having age more than 60 is predicted to grow up to 1.4 billion (WPA, 2017) and by 2050, it is predicted to grow from 962

million to 2.1 billion, relative to the year 2017 (Uddin et al., 2018). Also, according to the report WHO (2008), approx 28% to 35% of people in the age group of more than 65 years fall at least once every year. This number increases 32% to 42% for people having age more than 70 years (WHO, 2008). The major concern here is how to handle an aging population and provide them with effective healthcare services (Singh, 2014; Frieson, 2016) since in the worst cases, the fall of older people can obstruct the independence of their life and make them dependent on other human beings for the rest of their life. One of the solutions includes the presence of a person for assistance and personal care. But physically it is not possible for any person to remain present with an elderly person every time. There may be some scenarios when elderly people are left alone by their relatives. Also, there may be some cases where elderly people are prolonged living alone in their homes. It is dangerous to fall when alone because rescue time may increase significantly. In order to maintain their safety and to detect whether they have fallen or not when they are left alone, automatic fall detection systems are developed.

OVERVIEW

Human falls can occur at all ages, provoked by the loss of balance or the inability to regain balance. The World Health Organization (WHO, www.who.int, accessed on 1 February 2022) and the Prevention of Falls Network Europe (Pro-FaNE) define a fall as an event that inadvertently takes a person to the ground or to a lower level, excluding intentional situations, it was defined that a fall is “an unexpected event in which the participant comes to rest on the ground, floor, or lower without known loss of consciousness”. Although human falls are common at young ages, it is at advanced ages that they can be a significant health risk. Falls can result in dependence, loss of autonomy, confusion, immobilization, and depression, negatively affecting the person’s day-to-day activities. The health issues resulting from human falls affect all countries, but especially countries where there is a significant rate of elderly people. Moreover, the share of elderly people in the population is increasing. In Europe, people aged 65 or above represented 20.2% of the population in 2019 and are predicted to reach 29.4% in 2050. According to, falls can be classified according to their cause, as being accidental falls, unpredictable physiological falls, and predictable physiological falls. Accidental falls are unpredictable falls that result from external factors that affect people with no risk of falling. Unpredictable physiological falls represent about 8% of falls and affect people with no risk of

falling, but they are the result of physiological factors, such as pathological fractures or seizures. Predictable physiological falls are 80% of falls and affect people with a risk of falling, mainly resulting from physiological changes. To identify people with a risk of falling and to evaluate the likelihood of falling, the Morse Falls Scale (NDE) can be applied to patients and elderly persons.

OBJECTIVE

- To develop an intelligent and effective fall detection and alert system.
- To develop a reliable and cost efficient fall detection and alert system.
- To develop a fall detection system that is user friendly and without causing disturbance to activities of daily living of elderly people

LITERATURE SURVEY

IoT-Based Human Fall Detection System Osvaldo Ribeiro, Luis Gomes * and Zita Vale

Human falls are an issue that especially affects elderly people, resulting in permanent disabilities or even in the person's death. Preventing human falls is a social desire, but it is almost impossible to achieve because it is not possible to ensure full prevention. A possible solution is the detection of human falls in near real-time so that help can quickly be provided. This has the potential to greatly reduce the severity of the fall in long-term health consequences. This work proposes a solution based on the internet of things devices installed in people's homes. The proposed non-wearable solution is non-intrusive and can be deployed not only in homes but also in hospitals, rehabilitation facilities, and elderly homes. The solution uses a three-layered computation architecture composed of edge, fog, and cloud. A mathematical model using the Morlet wavelet and an artificial intelligence model using artificial neural networks are used for human fall classification; both approaches are compared. The results showed that the combination of both models is possible and brings benefits to the system, achieving an accuracy of 92.5% without false negatives.

Human Fall Detection Using Machine Learning Methods: A Survey Komal Singh

Department of Computer Science and Engineering, Graphic Era Deemed to be University, Dehradun, Uttarakhand, India. Corresponding author: komalsingh.0492@gmail.com

Akshay Rajput Department of Computer Science and Engineering, Graphic Era Deemed to be University, Dehradun, Uttarakhand, India. E-mail: akshay.rajput.1711@gmail.com

Sachin Sharma Department of Computer Science and Engineering, Graphic Era Deemed to be University, Dehradun, Uttarakhand, India. E-mail: sxsharma88@gmail.com

Human fall due to an accident can cause heavy injuries which may lead to a major medical issue for elderly people. With the introduction of new advanced technologies in the healthcare sector, an alarm system can be developed to detect a human fall. This paper summarizes various human fall detection methods and techniques, through observing people's daily routine activities. A human fall detection system can be designed using one of these technologies: wearable based device, context-aware based and vision based methods. In this paper, we discuss different machine learning models designed to detect human fall using these techniques. These models have already been designed to discriminate fall from activities of daily living (ADL) like walking, moving, sitting, standing, lying and bending. This paper is aimed at analyzing the effectiveness of these machine learning algorithms for the detection of human fall.

Wearable Device based Fall Detection Systems Wearable fall detection devices are electronic devices that are worn by a person under or on top of their clothes (Nizam et al., 2016). These devices comprise different types of sensors such as gyroscopes, accelerometers, tilt-meters, and oscilloscopes to detect a fall. A person has to carry the device with him all the time. Hence the elderly get location independence as they can move with the monitoring device (Doukas et al., 2007; Chen et al., 2010). But in reality, sometimes it is not possible like if the person forgets to wear the device or forgets to charge its battery or feels uncomfortable to wear it. Because of these limitations, developers direct their research towards developing contact-free fall detection systems where the user does not bother to wear anything additional for their safety (Uddin et al., 2018).

Human fall detection on embedded platform using depth maps and wireless accelerometer

Bogdan Kwolek1* and Michal Kepski2

Since falls are a major public health problem in an ageing society, there is considerable demand for low-cost fall detection systems. One of the main reasons for non-acceptance of the currently available solutions by seniors is that the fall detectors using only inertial sensors generate too much false alarms. This means that some daily activities are erroneously signaled as fall, which in turn leads to frustration of the users. In this paper we present how to design and implement a low-cost system for reliable fall detection with very low false alarm ratio. The detection of the fall is done on the basis of accelerometric data and depth maps. A tri-axial accelerometer is used to indicate the potential fall as well as to indicate whether the person is in motion. If the measured acceleration is higher than an assumed threshold value, the algorithm extracts the person, calculates the features and then executes the SVM-based classifier to authenticate the fall alarm. It is a 365/7/24 embedded system permitting unobtrusive fall detection as well as preserving privacy of the user.

Assistive technology or adaptive technology is an umbrella term that encompasses assistive and adaptive devices for people with special needs [1, 2]. Special needs and daily living assistance are often associated with seniors, disabled, overweight and obese, etc. Assistive technology for ageing-at-home has become a hot research topic since it has big social and commercial value. One important aim of assistive technology is to allow elderly people to stay as long as possible in their home without changing their living style. Wearable sensor-based systems for health monitoring are an emerging trend and in the near future they are expected to make possible proactive personal health monitoring along with better medical treatment. Inertial measurement units (IMUs) are low-cost and low power consumption devices with many potential applications. Current miniature inertial sensors can be integrated into clothes or shoes [3]. Inertial tracking technologies are becoming widely accepted for the assessment of human movement in health monitoring applications [4]. Wearable sensors offer several advantages over other sensors in terms of cost, weight, size, power consumption, ease of use and, most importantly, portability. Therefore, in the last decade, many different methods based on inertial sensors were developed to detect human falls. Falls are a major cause of injury for older people and a significant obstacle in independent living of the seniors. They are one of the top causes of injury-related hospital admissions in people aged 65 years and over. The statistical results demonstrate that at least one-

third of people aged 65 years and over fall one or more times a year [5]. An injured elderly may be laying on the ground for several hours or even days after a fall incident has occurred. Therefore, significant attention has been devoted to developing an efficient wearable system for human fall detection.

EXISTING SYSTEM

The drawbacks of the system is the limited range of infrared sensor.thus it is confined to limited application.

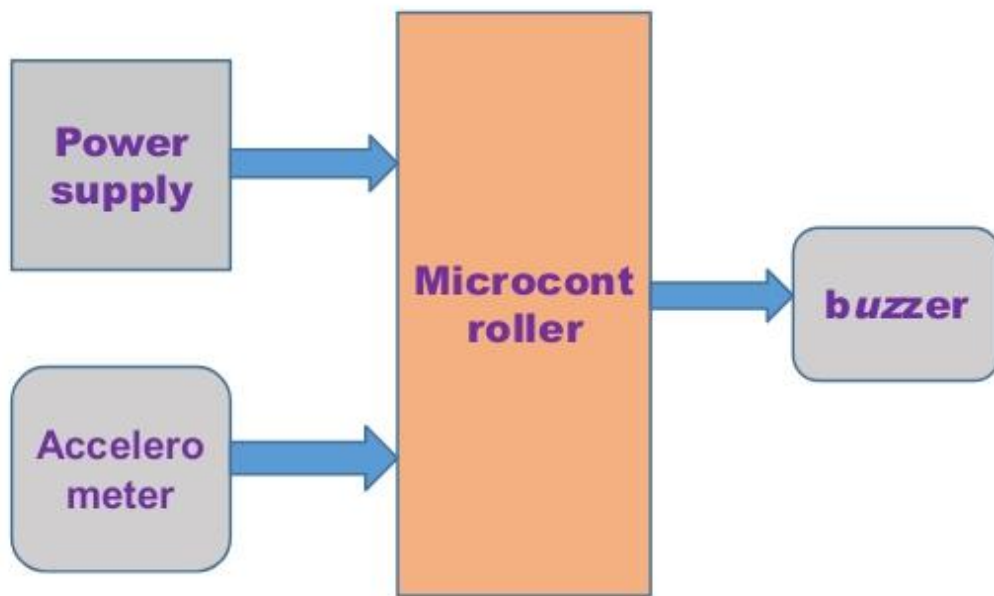
- We just get to know about the emergency situation through the emergency switches, or the person the person themselves informing through call.

PROBLEM STATEMENT

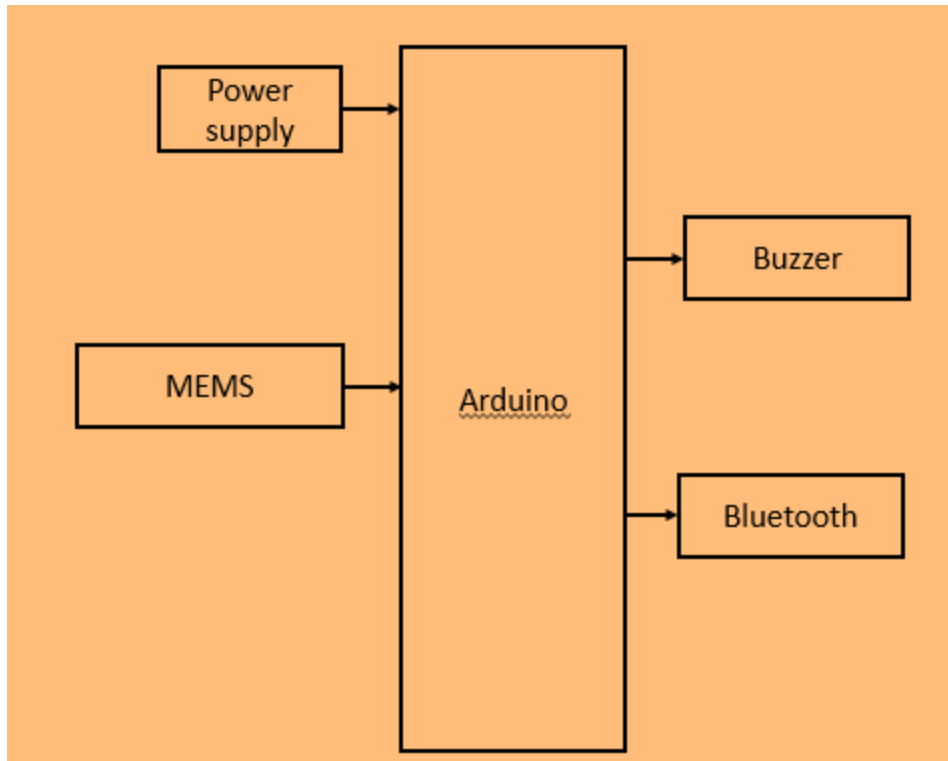
- Hence it is impossible for any people to monitor the emergency people's all the time from nearby situation.
- This creates the panic situation when the person has fallen down and can't able to inform it to the people.

PROPOSED SYSTEM

- This is a system consist of an arduino and MEMS sensor as the major sensors which will be able to detect the fall of a person.
- Then we use the Bluetooth module that can be able to transmit the data to the registered mobile number.
- Hence it will be programmed accordingly to detect the fall and to inform it to the person's guardian immediately.



Proposed BLOCK DIAGRAM



HARDWARE REQUIREMENTS

- Arduino
- Mems sensor
- Bluetooth
- Buzzer

SOFTWARE REQUIREMENTS

- Arduino IDE

HARDWARE DESIGN

ARDUINO UNO AND ITS PROGRAMMING

Arduino is a tool for making computers that can sense and control more of the physical world than your desktop computer. It's an open-source physical computing platform based on a simple microcontroller board, and a development environment for writing software for the board.

Arduino can be used to develop interactive objects, taking inputs from a variety of switches or sensors, and controlling a variety of lights, motors, and other physical outputs. Arduino projects can be stand-alone, or they can be communicate with software running on your computer. The boards can be assembled by hand or purchased preassembled; the open-source IDE can be downloaded for free.

The Arduino programming language is an implementation of Wiring, a similar physical computing platform, which is based on the Processing multimedia programming environment.

Arduino Uno:

2.1 Overview

The Arduino microcontroller is an easy to use yet powerful single board computer that has gained considerable traction in the hobby and professional market. The Arduino is open-source,

which means hardware is reasonably priced and development software is free. This guide is for students in ME 2011, or students anywhere who are confronting the Arduino for the first time. For advanced Arduino users, prowl the web; there are lots of resources.

This guide covers the Arduino Uno board (Spark fun DEV-09950, \$29.95), a good choice for students and educators. With the Arduino board, you can write programs and create interface circuits to read switches and other sensors, and to control motors and lights with very little effort. Many of the pictures and drawings in this guide were taken from the documentation on the

This is what the Arduino board looks like.

The Duemilanove board features an Atmel ATmega328 microcontroller operating at 5 V with 2 Kb of RAM, 32 Kb of flash memory for storing programs and 1 Kb of EEPROM for storing parameters. The clock speed is 16 MHz, which translates to about executing about 300,000 lines of C source code per second. The board has 14 digital I/O pins and 6 analog input pins. There is a USB connector for talking to the host computer and a DC power jack for connecting an external 6-20 V power source, for example a 9 V battery, when running a program while not connected to the host computer. Headers are provided for interfacing to the I/O pins using 22 g solid wire or header connectors.

The Arduino programming language is a simplified version of C/C++. If you know C, programming the Arduino will be familiar. If you do not know C, no need to worry as only a few commands are needed to perform useful functions.

An important feature of the Arduino is that you can create a control program on the host PC, download it to the Arduino and it will run automatically. Remove the USB cable connection to the PC, and the program will still run from the top each time you push the reset button. Remove the battery and put the Arduino board in a closet for six months. When you reconnect the battery, the last program you stored will run. This means that you connect the board to the host PC to develop and debug your program, but once that is done, you no longer need the PC to run the program.

The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

2.2 Summary:

Microcontroller ATmega328

Operating Voltage 5V

Input Voltage (recommended) 7-12V

Input Voltage (limits) 6-20V

Digital I/O Pins 14 (of which 6 provide PWM output)

Analog Input Pins 6

DC Current per I/O Pin 40 mA

DC Current for 3.3V Pin 50 mA

Flash Memory 32 KB (ATmega328) of which 0.5 KB used by bootloader

SRAM 2 KB (ATmega328)

EEPROM 1 KB (ATmega328)

Clock Speed 16 MHz

2.3 Pin Configuration

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall- wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as

opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

- **5V.** this pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.
- **IOREF.** This pin on the Arduino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs for working with the 5V or 3.3V.

Memory

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM.

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using `pinMode()`, `digital`

Write(), and digital Read() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kohms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and Transmit (TX) TTL serial data.

These pins are connected to the corresponding pins of the ATmega8U2 USB-to- TTL Serial chip.

- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attach Interrupt () function for details.

- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the analog Write () function.

- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication using the SPI library.

- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and the analog Reference () function. Additionally, some pins have specialized functionality:

- **I2C: A4 or SDA pin and A5 or SCL pin.** Support I2C communication using the Wire library.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with analog Reference ().

Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

2.4 Communication

Microcontrollers depend on a host computer for developing and compiling programs. The software used on the host computer is known as an integrated development environment, or IDE. For the Arduino, the development environment is based on the open source Processing platform (www.processing.org) which is described by its creators as a “programming language and environment for people who want to program images, animation, and interactions.” The Arduino programming language leverages an open source project known as Wiring (wiring.org.co). The Arduino language is based on good old- fashioned C. If you are unfamiliar with this language, don’t worry; it’s not hard to learn, and the Arduino IDE provides some feedback when you make mistakes in your programs.

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A Software Serial library allows for serial communication on any of the Uno's digital pins. The

ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details. For SPI communication, use the SPI library.

As you go through the list of programming statements available in the Arduino IDE (choose Help->Reference), you might think there isn't much power for doing things like running servos, operating stepper motors, reading potentiometers, or displaying text on an LCD. Like most any language based on C, the Arduino supports the notion of "libraries" code

Repositories that extend core programming functionality. Libraries let you re-use code without having to physically copy and paste it into all your programs. The standard Arduino software installation comes with several libraries you may use, and you can download others from the Arduino support pages and from third-party websites that publish Arduino library code. A good example of a library you'll use with the Robot and likely many other robot projects is

Servo. This library allows you to connect one or more hobby R/C servos to the Arduino's digital I/O pins. The Servo library comes with the standard Arduino installation package Library->Servo. This adds the line
`#include <Servo.h>`

Which tells the Arduino IDE that you wish to include the Servo library in your sketch. With the functionality of the library now available to you, you can use its various functions to control one or more servos. For example, you can use the write function to rotate a servo to a specific position, from 0 to 180 degrees. The following code

```
myServo.write(90);
```

Moves a servo to its midpoint, or 90 degree position. Structurally, Arduino sketches are very straightforward and are pretty easy to read and understand. The Arduino program contains two main parts: setup () and loop (). These are programming functions that do what their names suggest: setup () sets up the Arduino hardware, such as specifying which I/O lines you plan to use, and whether

They are inputs or outputs. The loop () function is repeated endlessly when the Arduino is operating.

Arduino IDE (Integrated development environment) is used to write the program and dump into the Arduino board

MEMS

Micro-electromechanical systems (MEMS) is a technology that combines computers with tiny mechanical devices such as sensors, valves, gears, mirrors, and actuators embedded in semiconductor chips. MEMS or what he calls analogue computing will be "the foundational technology of the next decade." MEMS is also sometimes called *smart matter*.

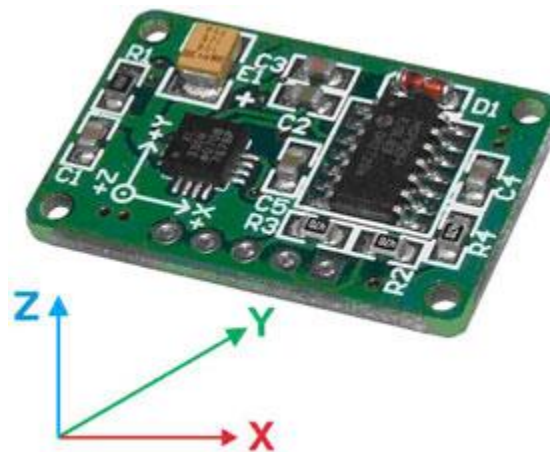
MEMS are already used as accelerometers in automobile air-bags. They've replaced a less reliable device at lower cost and show promise of being able to inflate a bag not only on the basis of sensed deceleration but also on the basis of the size of the person they are protecting. Basically, a MEMS device contains micro-circuitry on a tiny silicon chip into which some mechanical device such as a mirror or a sensor has been manufactured. Potentially, such chips can be built in large quantities at low cost, making them cost-effective for many uses.

Among the presently available uses of MEMS or those under study are:

- Global position system sensors that can be included with courier parcels for constant tracking and that can also sense parcel treatment en route
- Sensors built into the fabric of an airplane wing so that it can sense and react to air flow by changing the wing surface resistance; effectively creating a myriad of tiny wing flaps

- Optical switching devices that can switch light signals over different paths at 20-nanosecond switching speeds
- Sensor-driven heating and cooling systems that dramatically improve energy savings
- Building supports with imbedded sensors that can alter the flexibility properties of material based on atmospheric stress sensing

Visual Diagram:



FUNCTIONAL DIAGRAM:

One of the newer sensors to come about in recent times is the accelerometer. Granted the idea and implementation of a sensor that measures acceleration has been around for a long time the newer technologies available to industry have made them super accurate. The MEMS Accelerometer usually comes in the smallest surface mount package and can detect acceleration in up to 3 axis. This tutorial will cover capturing data for onle one axis. The [PIC](#) will be used to capture the data and make sense of it.

FINGERPRINT SENSOR

the fingerprint sensor we are using is an Optical Type, there exists two more types of sensor like capacitive which can be found in smart phones and ultrasonic ones, which are yet in testing phase, and both these options are expensive, so we will focus on this optical type for this hobby electronics and similar projects.

The way this optical fingerprint sensor works is that it captures a photo of our finger ridges, and then it uses certain algorithm to match it with stored data and displays result of the same.

few *features* of this sensor are as following:

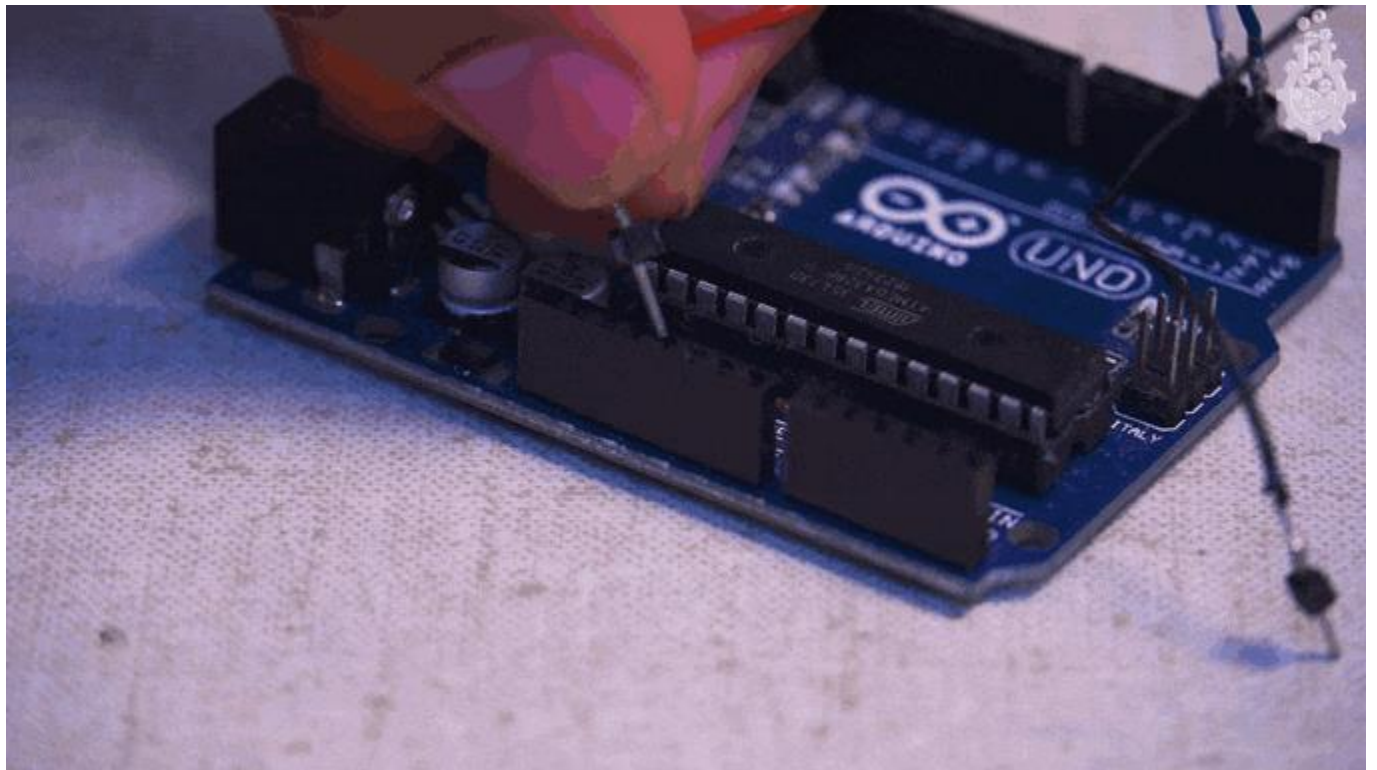
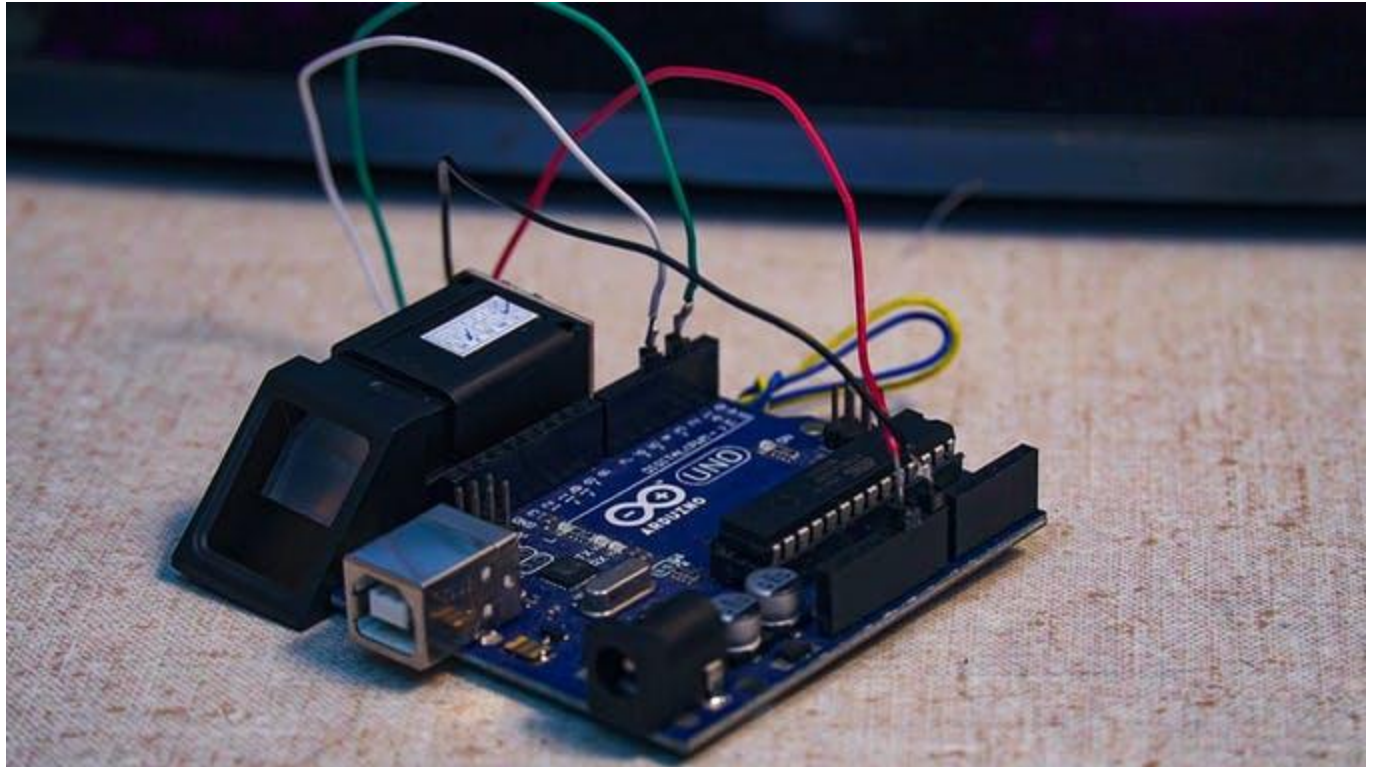
- Power supply: DC 3.8V-7.0V
- Operating current: 65mA (Typical)
- Interface: UART (TTL logical level)
- Average searching time: <1s (1:500, average)
- Security level: 5(1, 2, 3, 4, 5(highest))
- Working environment:Temp: -20°C to +60°
- Touch area dimension: 14.5*19.4 mm
- Outline dimension: 54*20*20.5 mm

Step 3: Solder Male Headers to Sensor

Since the wires that come along with sensor are flexible and the ends of this wire are quite wiggly so we would add male headers (or female headers in case if you use arduino nano) to connect those wires to female headers on arduino board.

You can use a helping hand or something to make your soldering task a lot easier since our header pins are small and we really don't want to risk burning our hands with solder iron.

Step 4: Connect the Sensor With Arduino



The connections part is very very simple, this fingerprint sensor has 6 wires, out of which we only 4 wires are useful, for arduino interfacing, out of which 2 wires will be used for power and 2 for data.

Connections:

- Red to 5v,
- Black to GND,
- Green to pin D2
- White to pin D3

(You can also refer to connection diagram above)

Step 5: Include the Adafruits Fingerprint Library

To complete this project or tutorial, we will need "Adafruit's Fingerprint" library, which you can find attached in this step or Google it.

To install that library, follow these steps:

Add the library by selecting Add ZIP under *SKETCH* menu, *INCLUDE* Library options.

- Open arduino IDE
- Now, select the .zip file from the location where we downloaded the file!

Step 6: Enroll the Finger Print

first we need to enroll the fingers into the EPROM of our Sensor module, so we will follow these steps to enroll our fingerprints.

select the *enroll* example.

- Open Arduino IDE
- Under the FILE menu, and Examples, find the Adafruit's fingerprint library.
- Upload the enroll example.
- Open the Serial Monitor.
- Select the baud rate to be 9600.
- Enter the Finger Print ID number when the serial monitor prompts to enter the fingerprint id number.
- Place the finger you need to enroll on the sensor.
- Place the finger again on the sensor once prompted by serial monitor.

Step 7: Upload the Main Code

Since we enrolled the fingerprints in the last step, now we can test if this sensor works or may be check for match using this following code and steps.

- Open arduino IDE
- Under the FILE menu, and Examples, find the adafruits finger print library.
- Select the fingerprint example.
- Upload the fingerprint example.
- Open the Serial Monitor.
- Select the baud rate to be 9600.
- Place the finger you need to test on the sensor.
- Check for fingerprint id number for OUTPUT.

Step 8: Modify the Main Code to Interface LED (or Solenoid)

Now let's add a led to indicate our Fingerprint match, to do so follow these steps:

use `"digitalWrite(13, HIGH); delay(1000); digitalWrite(13, LOW);"` on line 144 to turn on and off the LED

- Define the LED on pin 13 as output in void setup part
- Find then on line 135 we see a fingerprint id match loop, where in we use if statements along with library definitions to find our fingerprint id.
- We will give a high pulse for 1 second and turn the LED off.
- After this line and our code will turn on the LED every time sensor finds a match.

You can add a solenoid instead of this LED, and make a fingerprint door lock, which could be a brief tutorial for future, but for this tutorial, LED indication is our Eureka!

Step 9: Conclusion

In this tutorial post, we learnt how to set up a fingerprint sensor and how to use it with our Arduino, you can use this knowledge to control a door lock, and LED or any apparatus that you want to control.

BLUETOOTH MODULE

HC-05 Bluetooth Module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup. Its communication is via serial communication which makes an easy way to interface with controller or PC. HC-05 Bluetooth module provides switching mode between master and slave mode which means it able to use neither receiving nor transmitting data. Specification:

- Model: HC-05
- Input Voltage: DC 5V
- Communication Method: Serial Communication
- Master and slave mode can be switched

In communication, the Bluetooth wireless technology has become very popular and it is one of the fastest growing fields in the wireless technology. Hence it is important to learn how the **HC 05 Bluetooth module** interfacing with the microcontroller. Nowadays, demands of mobile phones and personal communication the bandwidth are easy and convenient to use. The Bluetooth technology manages the communication channel of the wireless part. The Bluetooth modules can transmit and receives the data wirelessly by using two devices. The Bluetooth module can receive and transmits the data from a host system with the help of the **host controller interface** (HCI). The UART & USB are the most popular host controller interfaces and in this article, we have discussed the UART. By using the UART connections, the Bluetooth module can be integrated and it gives the best possible solutions for the Bluetooth embedded systems.

HC 05 Bluetooth is a wireless communication protocol; it is used in two devices as a sending and receiving the information. The Bluetooth is free to use in the wireless communication protocol as the range of the Bluetooth is less than the other **wireless communication protocols like WiFi and Zigbee**. The Bluetooth operates at the frequency of the 2.41 GHz and also used in many small ranges of applications.

There are three different types of Bluetooth module which are classified and described below

Class 1: The output range of Class 1 Bluetooth module is about the 100 mW. The distance between the two Bluetooth module devices is about the 100 meters.

Class 2: The output range of this Bluetooth module is about the 2.5 mW and the distance of two Bluetooth devices are about the 10 meters.

Class 3: The output range of this type Bluetooth module is 1mW and the distance is about the 10 centimeters of two Bluetooth devices.

The Bluetooth module requires a communication protocol to interface with the other devices. In this article we are discussing the Bluetooth module is interfaced with the microcontroller and pic microcontroller. The microcontroller can communicate with the Bluetooth devices with the following wired communications to receive and send the information to other Bluetooth devices.

- UART
- SPI
- USB

We can use any one of the above three wired communication for the interfacing to a microcontroller with the Bluetooth module. There are many Bluetooth modules which are available in the market. The Bluetooth modules are following. In this article, we will discuss two modules which are HC 05 Bluetooth Module and RN-42.

- HC-04
- HC-05
- RN-41
- RN-42
- AUBTM
- BLUESMIRF

The RN-42 is a class two type Bluetooth module it has UART and USB port communication interface. The RN-42 can be used for both as a receiver and send the information and it also

works as a master and slave. In audio applications, the RN-42

RN-42 Module Interfacing with PIC-Microcontroller

We need a voltage level shifting circuit with RN-42 to interface this with the pic microcontroller or with any microcontroller. The microchip provides its solution by giving the complete compatible board to interface with the microcontroller. The RN-42 EK board is interfaced with the pic microcontroller by connecting directly, i.e. UART-RX pin of the RN-42 Bluetooth module to the pic microcontroller of Tx pin and UART-Tx pin is connected to the Rx pin of pic microcontroller. By using the USB cable the RN-42 EK board is connected to the computer.

The following diagram shows the interfacing of RN-42 with the **pic microcontroller**. There are two types of modes in the RN-42. They have command mode and data mode and there is no antenna for the RN42. The command modes are used for the set configuration of Bluetooth modules like master or slave and serial port control. The Bluetooth module is kept in a particular form before using the data mode. Hence there is a use of the default values of the commands. We have to remember that there is a specific time for the command mode and we have to use the command mode within the time otherwise the Bluetooth module will reach the data mode behind the time.

HC 05 Bluetooth Module

The HC 05 Bluetooth module is the most popular module in the Indian market and this module is mostly used in the embedded projects. The HC 05 Bluetooth modules are easy to use & simple, its price is low and these types of modules are interfaced with the Arduino, Raspberry Pi, and Microcontroller through the serial UART interface. These modules are designed for the transparent wireless connection setup and it is very easy to use in the Bluetooth serial port protocol.

The serial port, Bluetooth module is highly qualified Bluetooth with the version of the V2.0+Enhanced Data Rate of 3Mbps. The modulation has completely 2.4 GHz radio transceiver and baseband. These serial port, Bluetooth modules use the CSR Bluecore 04-external single chip Bluetooth system by using the COMS technology and also with the Adaptive Frequency Hopping Feature. The footprint of this Bluetooth module is as small as 12.7mmx27mm. Hence it will help in the overall design and development cycle.

The following are some specifications of the H05 Bluetooth module

- The frequency is about the ISM band of 2.4GHz.
 - The Bluetooth specification of the Bluetooth module is V2.0 + EDR
 - The emission power is about the <4dbm and it is a Class 2 Bluetooth Module
 - The input power supply is from the 3.6V To 6v
 - The operating voltage of the HC 05 Bluetooth module is 3.3V
 - The Dimensions are 35.7mm x 15.2mm x 5.6mm
- The HC 05 Bluetooth module works at the temperature of -200C ~ + 750C.
- The security of this Bluetooth module is in authentication and encryption.

The following image shows the HC 05 Bluetooth module

HC 05 Bluetooth Module

This Module has two types of operation modes which are following

- AT Command Mode
- Connection Mode

AT Command Mode

The AT command mode is a module whenever the set of commands is used for the setup and configure the module. In this mode, the modules are not identified by the other Bluetooth devices. This type of mode will take the pin to high and the commands are sent to the module serially like string. The string should be in the capital letters and every command should be ended by the 'rn'. If the command is known by the module then the module will reply with the string which contains the 'OK'. If the command is unknown by the module will reply with the string which contains ERROR (). The error determines the type of error and a specific code.

ARDUINO SOFTWARE:

ARDUINO

Arduino is a tool for making computers that can sense and control more of the physical world

than your desktop computer. It's an open-source physical computing platform based on a simple microcontroller board, and a development environment for writing software for the board.

Arduino can be used to develop interactive objects, taking inputs from a variety of switches or sensors, and controlling a variety of lights, motors, and other physical outputs. Arduino projects can be stand-alone, or they can be communicate with software running on your computer (e.g. Flash, Processing, MaxMSP.) The boards can be assembled by hand or purchased preassembled; the open-source IDE can be downloaded for free.

The Arduino programming language is an implementation of Wiring, a similar physical computing platform, which is based on the Processing multimedia programming environment.

Why Arduino?

There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than \$50

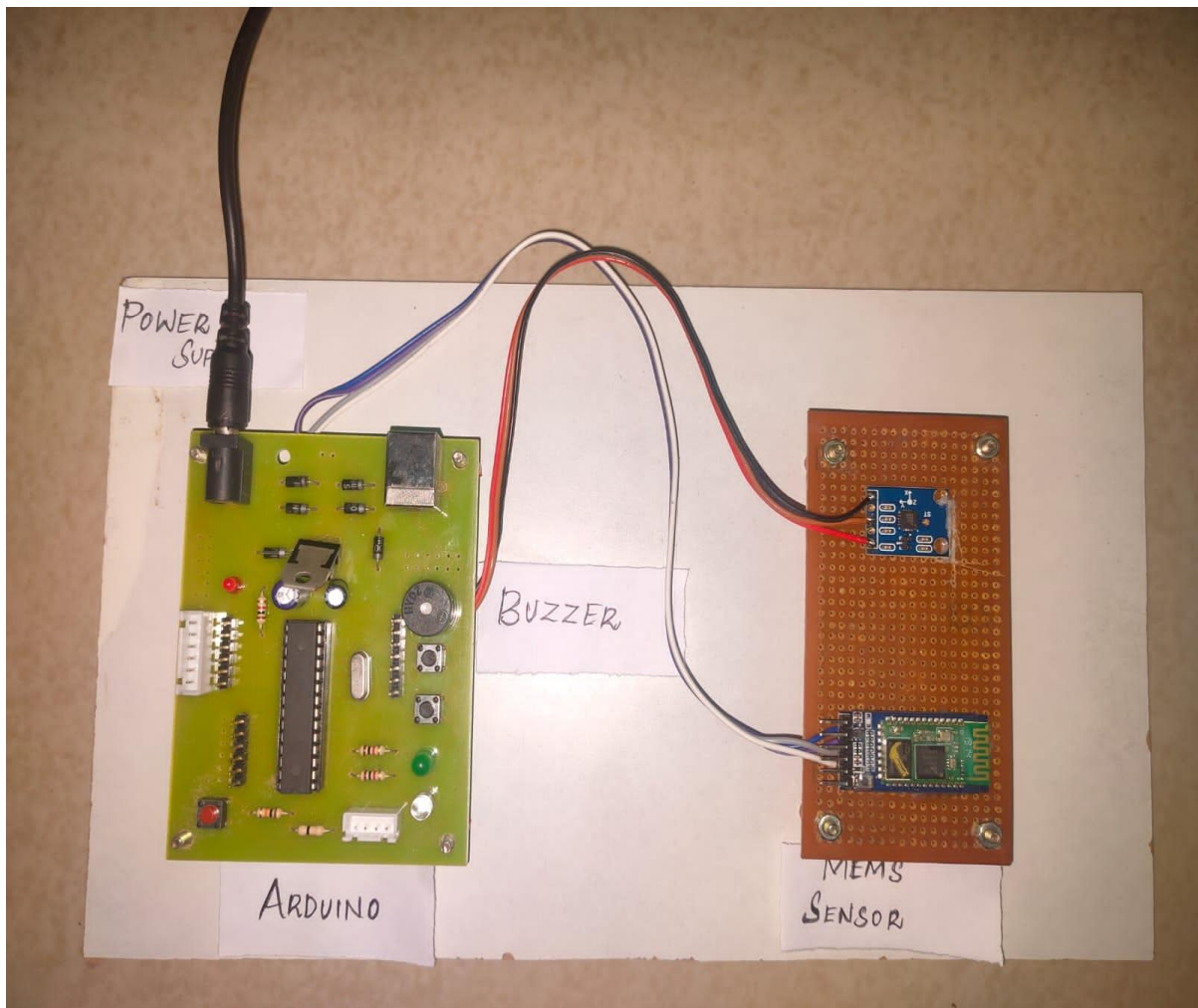
Cross-platform - The Arduino software runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.

Simple, clear programming environment - The Arduino programming environment is easy-

to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with the look and feel of Arduino

Open source and extensible software- The Arduino software and is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based.

Hardware:



Output:



04:03:22 HC-05 > 403
04:03:22 HC-05 > 392
04:03:22 HC-05 > 385
04:03:22 HC-05 > FALL DETECTED
04:03:22 HC-05 > 394
04:03:22 HC-05 > 395
04:03:22 HC-05 > 401
04:03:23 HC-05 > 395
04:03:23 HC-05 > 386
04:03:24 HC-05 > FALL DETECTED
04:03:25 HC-05 > 395
04:03:25 HC-05 > 386
04:03:26 HC-05 > FALL DETECTED
04:03:27 HC-05 > 393
04:03:27 HC-05 > 383
04:03:28 HC-05 > FALL DETECTED
04:03:29 HC-05 > 395
04:03:29 HC-05 > 386
04:03:30 HC-05 > FALL DETECTED
04:03:31 HC-05 > 386
04:03:32 HC-05 > FALL DETECTED
04:03:33 HC-05 > 408
04:03:33 HC-05 > 381
04:03:34 HC-05 > FALL DETECTED
04:03:35 HC-05 > 390
04:03:35 HC-05 > 383
04:03:36 HC-05 > FALL DETECTED
04:03:37 HC-05 > 388
04:03:37 HC-05 > FALL DETECTED

04:03:41 HC-05 > FALL DETECTED
04:03:42 HC-05 > 395
04:03:43 HC-05 > 404
04:03:43 HC-05 > 392
04:03:44 HC-05 > 395
04:03:44 HC-05 > 387
04:03:45 HC-05 > FALL DETECTED
04:03:46 HC-05 > 395
04:03:46 HC-05 > 384
04:03:47 HC-05 > FALL DETECTED
04:03:48 HC-05 > 408
04:03:48 HC-05 > 394
04:03:49 HC-05 > 399
04:03:49 HC-05 > 395
04:03:50 HC-05 > 384
04:03:50 HC-05 > FALL DETECTED
04:03:51 HC-05 > 398
04:03:52 HC-05 > 400
04:03:52 HC-05 > 394
04:03:53 HC-05 > 396
04:03:53 HC-05 > 384
04:03:54 HC-05 > FALL DETECTED
04:03:55 HC-05 > 399
04:03:55 HC-05 > 400
04:03:56 HC-05 > 391
04:03:56 HC-05 > 394
04:03:57 HC-05 > 386
04:03:57 HC-05 > FALL DETECTED
04:03:58 HC-05 > 468
04:03:59 HC-05 > 471
04:03:59 HC-05 > 467

```
04:03:59 HC-05 > 467  
04:04:00 HC-05 > 467  
04:04:00 HC-05 > 462  
04:04:01 HC-05 > 470  
04:04:01 HC-05 > 466  
04:04:02 HC-05 > 462  
04:04:02 HC-05 > 453  
04:04:03 HC-05 > 468  
04:04:03 HC-05 > 462  
04:04:04 HC-05 > 470  
04:04:04 HC-05 > 466  
04:04:05 HC-05 > 462  
04:04:05 HC-05 > 468  
04:04:06 HC-05 > 470  
04:04:06 HC-05 > 468  
04:04:07 HC-05 > 467  
04:04:07 HC-05 > 467  
04:04:08 HC-05 > 467  
04:04:08 HC-05 > 464  
04:04:09 HC-05 > 466  
04:04:09 HC-05 > 466  
04:04:10 HC-05 > 463  
04:04:10 HC-05 > 472  
04:04:11 HC-05 > 466  
04:04:11 HC-05 > 461  
04:04:12 HC-05 > 468  
04:04:12 HC-05 > 468  
04:04:13 HC-05 > 460  
04:04:13 HC-05 > 467  
04:04:14 HC-05 > 469  
04:04:15 HC-05 > 466  
04:04:15 HC-05 > 468
```

RESULT

The several fall-feature parameters of the 6-axes acceleration were introduced and applied according the algorithm. Possible falls were chosen through the simple threshold and then applied to the MPU to solve the problems such as deviation of interpersonal falling behavioral patterns and similar fall actions. The test of the proposed device studied along a

different 350 case studies. The parameters of upper and lower of acceleration and velocity have adjusted to give best fall detection with sensitivity, specificity, and accuracy which were over than 95 %. These results demonstrate the reduction of the computing effort and resources, compared to those of using all the events applied. Then the proposed algorithms were very simple because it depend son a simple sensor (measure the angle) and the program calculates the angular velocity and acceleration.

REFERENCE

1. Chan, M., Esteve, D., Escriba, C., Campo, E.: A review of smart homes - present state and future challenges. *Computer Methods and Programs in Biomedicine* 91(1), 55–81 (2008)
2. Cook, A.M.: The future of assistive technologies: A time of promise and apprehension. In: *Proc. of the 12th Int. ACM SIGACCESS Conf. on Comp. and Accessibility*, pp. 1–2. ACM, New York, USA (2010)
3. Hoflinger, F., Muller, J., Zhang, R., Reindl, L.M., Burgard, W.: A wireless micro inertial measurement unit (imu). *Instrumentation and Measurement, IEEE Transactions on* 62(9), 2583–2595 (2013)
4. Buesching, F., Kulau, U., Gietzelt, M., Wolf, L.: Comparison and validation of capacitive accelerometers for health care applications. *Comp. Methods and Programs in Biomedicine* 106(2), 79–88 (2012)
5. Heinrich, S., Rapp, K., Rissmann, U., Becker, C., K"onig, H.-H.: Cost of falls in old age: a systematic review. *Osteoporosis Int.* 21, 891–902 (2010).