

BEND SENSOR-BASED HUMAN POSTURE MONITORING

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Technology degree in
Biomedical Engineering

By

AISHWARYA.J (39240003)

M.HEPZIBAH SUSAN (39240018)



**DEPARTMENT OF BIOMEDICAL ENGINEERING SCHOOL
OF BIO AND CHEMICAL ENGINEERING**

SATHYABAMA

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

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

MAY-2022



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

www.sathyabama.ac.in



DEPARTMENT OF BIOMEDICAL ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Interdisciplinary Project Report is the bonafide work of **AISHWARY.J (Reg. No.39240003)**, **M. HEPZIBAH SUSAN (Reg. No. 39240018)** who carried out the project entitled "**BEND SENSOR - BASED HUMAN POSTURE MONITORING**" under our supervision from **DECEMBER 2021 to MAY 2022**.

Dr. T. SUDHAKAR, M.Sc., Ph.D.,
Guide and Supervisor

Dr. T. SUDHAKAR, M.Sc., Ph.D.,
Head of the Department

Submitted for Viva-voce Examination held on 30Th April 2022.

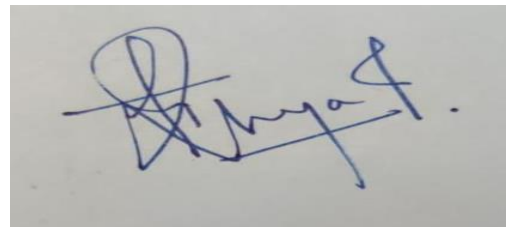
Internal Examiner

External Examiner

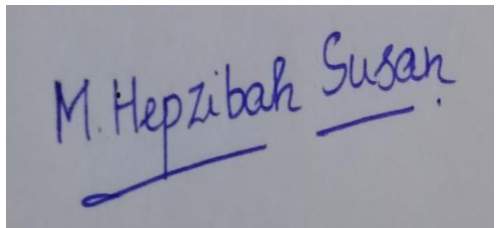
DECLARATION

We, **AISHWARYA.J** (Reg. No. 39240003), **M. HEPZIBAH SUSAN** (Reg. No. 39240018) hereby declare that the Interdisciplinary Project Report entitled “**BEND SENSOR BASED HUMAN POSTURE MONITORING**” done by us under the guidance of **Dr. T. SUDHAKAR**, Department of **Biomedical Engineering** is submitted in partial fulfillment of the requirements for the award of a **Bachelor of Technology** degree in **Biomedical Engineering**.

1.

A handwritten signature in blue ink, appearing to read 'Aishwarya.J.', on a light-colored background.

2.

A handwritten signature in blue ink, reading 'M. Hepzibah Susan' with a horizontal line underneath, on a light-colored background.

DATE:28.04.2022

PLACE: Chennai

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

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

We convey our thanks to **Dr. T. Sudhakar, M.Sc., Ph.D., Head of the Department, Department of Biomedical Engineering** 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 **Dr. T. Sudhakar, M.Sc., Ph.D., Department of Biomedical Engineering** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of our Interdisciplinary project work.

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

ABSTRACT

Recently, sitting posture monitoring has been focused in the fields of health management in order to improve the quality of Sedentary Lifestyle. We propose an unconstraint sitting posture monitoring method using two sensors. The sensor has several advantages for unconstrained human behavior monitoring, such as single-mode transmission usage, flexibility and light-weight. Therefore, sensor enables the sitting posture to be monitored without conscious to the sensor. We successfully demonstrated that the developed sensitive cushion could detect body moving of a subject on a chair. The proposed method could monitor body moving in the case that the subject sat down on several moving patterns on the chair.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	1
	LIST OF ABBREVIATION	4
	LIST OF FIGURES	5
1	INTRODUCTION	6
	1.1 DOMAIN OVERVIEW	6
	1.1.1 OVERVIEW OF EMBEDDED SYSTEM	6
	1.1.2 BLOCK DIAGRAM OF AN EMBEDDED SYSTEM	7
	1.1.3 CHARACTERISTICS OF EMBEDDED SYSTEM	8
	1.1.4 APPLICATION OF SPECIFIC SYSTEM	8
2	LITERATURE SURVEY	9
3	AIM AND SCOPE	12
	3.1 AIM	12
	3.2 SCOPE	12
4	MATERIALS AND METHODS	13
	4.1 METHODOLOGY	13
	4.1.1 BLOCK DIAGRAM WITH EXPLANATION	13
	4.2 HARDWARE USED	14
	4.2.1 ARDUINO	14
	4.2.2 FLEX SENSOR	19
	4.2.3 MEMS SENSOR	24
	4.2.4 BUZZER	30

	4.3 SOFTWARE REQUIREMENTS	32
	4.3.1 ARDUINO SOFTWARE	32
5	RESULT AND DISCUSSION	37
	5.1 RESULT	37
	5.2 DISCUSSION	38
6	SUMMARY AND RESULT	39
	6.1 SUMMARY	39
	6.2 CONCLUSION	39
	REFERENCES	40
	APPENDIXES	41

LIST OF ABBREVIATION

ABBREVIATION

EXPANSION

PWM

Pulse-Width Modulation

MEMS

Micro-electro-mechanical system

LIST OF FIGURES

FIGURE NO	NAME OF THE FIGURE	PAGE NO
1.1.2	block diagram of a typical embedded system	8
4.1.1	block diagram with explanation	13
4.2.1(a)	Arduino	15
4.2.1(b)	Arduino lily pad & Arduino mini	16
4.2.2(a)	flex sensor	19
4.2.2(b)	pin configuration of flex sensor	20
4.2.2(c)	resistance of flex sensor	21
4.2.2(d)	circuit diagram of flex sensor	22
4.2.3(a)	mems sensor	24
4.2.3(b)	setup of mems sensor	26
4.2.3(c)	circuit diagram of mems sensor	27
4.2.4(a)	buzzer	30
4.2.3(b)	buzzer pin configuration	31
4.3.1	command window	33&34
5.1.1	hardware setup	37
5.1.2	software setup	38

CHAPTER-1

INTRODUCTION

Physical activity has an important role in health management which recovers the fatigue of human body and mentality. Additionally, the risk of many diseases can be reduced due to physical activities. Most of people sit more than six hours per day in daily life. This is called as “Sedentary Lifestyle”. Recently, it has been proposed that measuring sitting posture enables the quality of Sedentary Lifestyle to be improved. Here we propose a pose estimation system where a flex sensor and mems sensor is used for monitoring a person’s pose We can implement this system using arduino embedded development platform We use most commonly used arduino board called arduino uno for implementing the project .This system is helpful for people work in office such as IT employees

1.1 DOMAIN OVERVIEW

1.1.1 OVERVIEW OF EMBEDDED SYSTEMS

An embedded system is a special-purpose computer system designed to perform one or a few dedicated functions, often with real-time computing constraints. It is usually embedded as part of a complete device including hardware and mechanical parts. In contrast, a general-purpose computer, such as a personal computer, can do many different tasks depending on programming. Embedded systems have become very important today as they control many of the common devices we use.

Since the embedded system is dedicated to specific tasks, design engineers can optimize it, reducing the size and cost of the product, or increasing the reliability and performance. Some embedded systems are mass-produced, benefiting from economies of scale.

In general, "embedded system" is not an exactly defined term, as many systems have some element of programmability. For example, Handheld computers share some elements with embedded systems — such as the operating systems and microprocessors which power them — but are not truly embedded systems, because they allow different applications to be loaded and peripherals to be connected.

Embedded systems provide several functions

- Monitor the environment; embedded systems read data from input sensors. This data is then processed and the results displayed in some format to a user or users
- Control the environment; embedded systems generate and transmit commands for actuators.
- Transform the information; embedded systems transform the data collected in some meaningful way, such as data compression/decompression

Although interaction with the external world via sensors and actuators is an important aspect of embedded systems, these systems also provide functionality specific to their applications. Embedded systems typically execute applications such as control laws, finite state machines, and signal processing algorithms. These systems must also detect and react to faults in both the internal computing environment as well as the surrounding electromechanical systems.

1.1.2 BLOCK DIAGRAM OF AN EMBEDDED SYSTEM:

An embedded system usually contains an embedded processor. Many appliances that have a digital interface -- microwaves, VCRs, cars -- utilize embedded systems. Some embedded systems include an operating system. Others are very specialized resulting in the entire logic being implemented as a single program. These systems are embedded into some device for some specific purpose other than to provide general purpose computing . A typical embedded system is shown in Fig 1.1.2

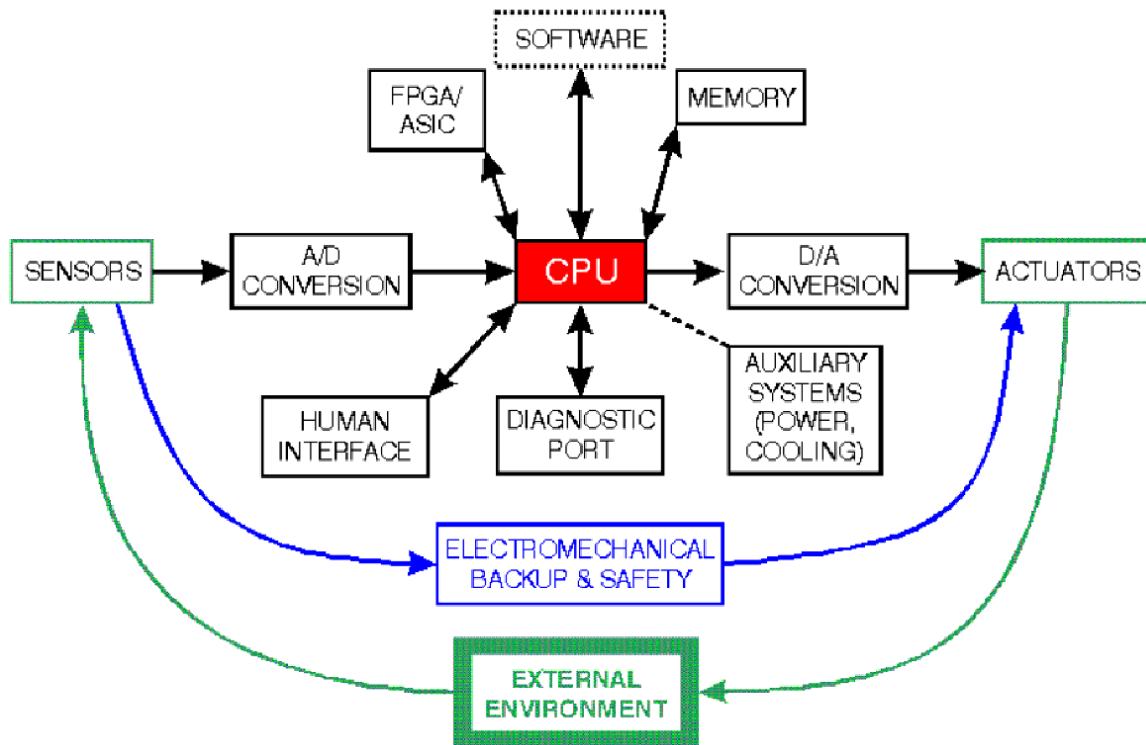


Fig: 1.1.2: BLOCK DIAGRAM OF A TYPICAL EMBEDDED SYSTEM

1.1.3 CHARACTERISTICS OF EMBEDDED SYSTEMS:

Embedded systems are characterized by a unique set of characteristics. Each of these characteristics imposed a specific set of design constraints on embedded systems designers. The challenge to designing embedded systems is to conform to the specific set of constraints for the application.

1.1.4 APPLICATION OF SPECIFIC SYSTEMS:

Embedded systems are not general-purpose computers. Embedded system designs are optimized for a specific application. Many of the job characteristics are known before the hardware is designed. This allows the designer to focus on the specific design constraints of a well-defined application. As such, there is limited user reprogram ability. Some embedded systems, however, require the flexibility of reprogram ability. Programmable DSPs are common for such applications.

CHAPTER 2

LITERATURE SURVEY

[1].Head Posture Estimation by Deep Learning Using 3-D Point Cloud Data From a Depth Sensor (2021)Y.P. Tsang, K.L. Choy*, T.C. Poon, G.T.S. Ho, C.H. Wu, H.Y. Lam, P.S. Koo, H.Y. Ho Seiji Sasaki; Chinthaka Premachandra Seiji Sasaki; Chinthaka Premachandra Through field investigation, it is found that the settlement range and subsidence trend of the mining area are consistent with the actual situation. The working performance is not improved, but also become more stable for work measurement.

[2]. Neural Network Augmented Sensor Fusion for Pose Estimation of Tensegrity Manipulators (2020)Askat Kuzdeuov; Matteo Rubagotti; Huseyin Atakan Varol All Authors In this paper, we present a pose estimation strategy for the end effector of a tensegrity manipulator, based on the use of an extended Kalman filter and a deep feedforward neural network with three hidden layers. Our scheme is based on the fusion of sensor data obtained from an inertial measurement unit and ArUco fiducial markers. The method was implemented on a six bar tensegrity prism manipulator, tested using ground truth acquired from an external vision-based motion capture system, and compared with other estimation methods. Pose estimation, sensor fusion, neural networks, fiducial markers, tensegrity robots, inertial measurement sensors, extended Kalman filters. The pose estimation method for tensegrity manipulators developed in this work will be used in future work for designing advanced feedback control algorithms. They requires improvements in reliability functions to accommodate poor quality of service and low network distribution.

[3]. Field Test Validations of Vision-based Multi-camera Multi-drone Tracking and 3D Localizing with Concurrent Camera Pose Estimation (2021) Niven Junliang Sie; Sutthiphong Srigrarom; Sunan Huang This paper reports the field test validations of the recently proposed vision-based real-time multi-camera setups for detecting, tracking and 3D localizing multiple aerial targets (mainly drones). We also propose the additional concurrent camera pose estimation when the camera poses are not known beforehand. This extra step can be used alongside (in parallel) with the drone

tracking and localizing process. We conducted flight tests using 2 drones flying in 2 specific scenarios, and used 3 cameras to observe, detect, track and locate the positions of both drones in global frame. Vision-based, Multicamera, Multi-object, Aerial target, Drones, Tracking, 3D Localization, Camera Pose Estimation, Field tests. I The nearby police station email id is also configured in the system, then the intrusion mail can be received by police also and necessary action can be taken immediately. Used RF and lora or Bluetooth based communication

[4]. RFID-Pose: Vision-Aided Three-Dimensional Human Pose Estimation With Radio-Frequency Identification 2021 Chao Yang; Xuyu Wang; Shiwen Mao In recent years, human pose tracking has become an important topic in computer vision (CV). To improve the privacy of human pose tracking, there is considerable interest in techniques without using a video camera. To this end, radio-frequency identification (RFID) tags, as a low-cost wearable sensor, provide an effective solution for 3-D human pose tracking. In this article, we propose RFID-Pose, a vision-aided realtime 3-D human pose estimation system, which is based on deep learning assisted by CV. Computer vision (CV), deep learning, high accuracy low rank tensor completion (HaLRTC), human pose estimation, Radio-frequency Identification (RFID). The RFID-pose system was prototyped with commodity RFID devices. Its high accuracy and realtime operation were demonstrated in our experimental study using Kinect 2.0 as a benchmark. That are not in the database will cause so much delay

[5]. Multimodal Sensor Fusion for LowPower Wearable Human Motion Tracking Systems in Sports Applications 2021 Mariusz P. Wilk; Michael Walsh; Brendan O'Flynn This paper presents a prototype human motion tracking system for wearable sports applications. It can be particularly applicable for tracking human motion during executing certain strength training exercises, such as the barbell squat, where an inappropriate technique could result in an injury. The key novelty of the proposed system is twofold. Firstly, it is an inside-out, multimodal, motion tracker that incorporates two complementary sensor modalities, i.e. a camera and an inertial motion sensor, as well as two externally-mounted points of reference. Secondly, it incorporates a novel multimodal sensor fusion algorithm which uses the complementary nature of vision and inertial sensor modalities to perform a

computationally efficient 3- Dimensional (3-D) pose detection of the wearable device. Inertial Motion Sensor, InsideOut Tracking, Pose Detection, Monocular Camera, Multimodal, Sensor Fusion, 3- D Future work can involve the development of a small-formfactor proof-of-concept prototype of the WP which will turn this pre-prototype system into a miniature wearable device. Furthermore, additional experimental evaluation of the miniaturized WP can be carried out with the involvement of human subjects. The individuals can perform real barbell squats with the miniaturized WP attached to their back. Used RF and lora or Bluetooth based communication

[6]. 1.Chengyu Li, Di Liu, Chaoqun Xu, Ziming Wang, Sheng Shu, Zhuoran Sun, Wei Tang & Zhong Lin Wang. Sensing of joint and spinal bending or stretching via a retractable and wearable badge reel. Nature Communications. 2021 Measurement of human body bending inclination: We used Unreal Four Engine developed by Epic Games, the SDK skeleton tracking package was customized by using blueprint and C++ to track and measure the inclination angle of human body bending in real time. Our proposed technology could be beneficial to reduce the risks of spinal disorders, especially for those long-term sitting in the office. It can serve as a rehabilitation brace for real-time recording of patients' joint motions during the long recovering time after injuries. Human motions, such as joint/spinal bending or stretching, often contain information that is useful for orthopaedic/neural disease diagnosis, rehabilitation, and prevention. Using the stretch sensor for real-time monitoring, our spinal motions represent a potential application that may reduce the cervical, vertebra, and back pain diseases in clinical, or decrease the number of hunchbacks.

[7].Ferdews Tlili, Rim Haddad, Ridha Bouallegue & Neila Mezghani A Real-time Posture Monitoring System Towards Bad Posture Detection Wireless Personal Communications 2021 The system is composed of a smart belt equipped with inertial sensors. The sensors collect the posture information and send them to a cloud server via Wi-Fi connection. The cloud server processes the collected data, then, sends the result to mobile applications via Wi-Fi connection. The mobile applications allow the user to monitor his posture over time and receive in Real-time a sound and a visual notification in case of a bad posture detection. Two mobile applications Android and iOS are implemented that can be used for different mobile phone OS.

CHAPTER-3

AIM AND SCOPE

3.1 AIM:

- It will present the posture measurements of good and bad posture using the proposed posture monitoring system.
- Most of people sit more than six hours per day in daily life. This is called as “Sedentary Lifestyle”.
- Recently, it has been proposed that measuring sitting posture enables the quality of Sedentary Lifestyle to be improved.

3.2 SCOPE:

- Human motions, such as joint/spinal bending or stretching, often contain information that is useful for orthopaedic/neural disease diagnosis, rehabilitation, and prevention.
- Using the stretch sensor for real-time monitoring, our spinal motions represent a potential application that may reduce the cervical, vertebra, and back pain diseases in clinical, or decrease the number of hunchbacks.

CHAPTER-4

MATERIALS AND METHODS

4.1 METHODOLOGY

- Here the central part is arduino uno
- Both sensors are connected to arduino uno
- Both flex sensor and mems sensor gives analog output
- Hence we interface these sensors using ADC
- Arduino has a 6 channel ADC
- Buzzer is connected to a digital output pin

4.1.1 BLOCK DIAGRAM WITH EXPLANATION

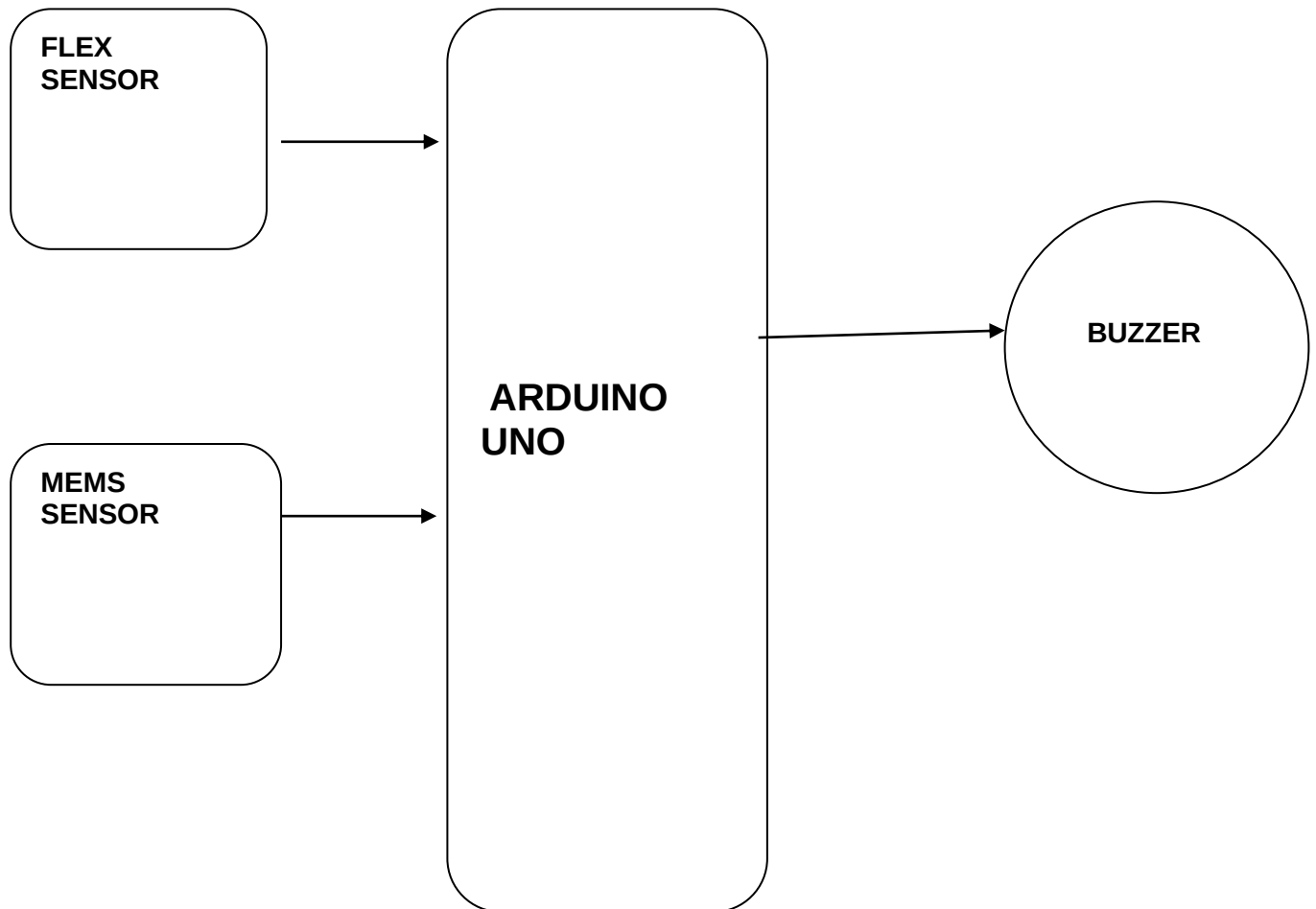


Fig.4.1.1:BLOCK DIAGRAM WITH EXPLANATION

4.2 HARDWARE USED

- ARDUINO
- FLEX SENSOR
- MEMS SENSOR
- BUZZER

4.2.1. ARDUINO

setup(): A function present in every Arduino sketch. Run once before the loop() function.

Often used to set pinmode to input or output. The setup() function looks like:

```
void setup( ){  
  //code goes here  
}
```

loop(): A function present in every single Arduino sketch. This code happens over and over again. The loop() is where (almost) everything happens. The one exception to this is setup() and variable declaration. ModKit uses another type of loop called “forever()” which executes over Serial. The loop() function looks like:

```
void loop( ){  
  //code goes here  
}
```

input: A pin mode that intakes information.

output: A pin mode that sends information.

HIGH: Electrical signal present (5V for Uno). Also ON or True in boolean logic.

LOW: No electrical signal present (0V). Also OFF or False in boolean logic.

digitalRead: Get a HIGH or LOW reading from a pin already declared as an input.

digitalWrite: Assign a HIGH or LOW value to a pin already declared as an output.

analogRead: Get a value between or including 0 (LOW) and 1023 (HIGH). This allows you to get readings from analog sensors or interfaces that have more than two states.

analogWrite: Assign a value between or including 0 (LOW) and 255 (HIGH). This allows you to set output to a PWM value instead of just HIGH or LOW.

PWM: Stands for Pulse-Width Modulation, a method of emulating an analog signal through a digital pin. A value between or including 0 and 255. Used with analogWrite.

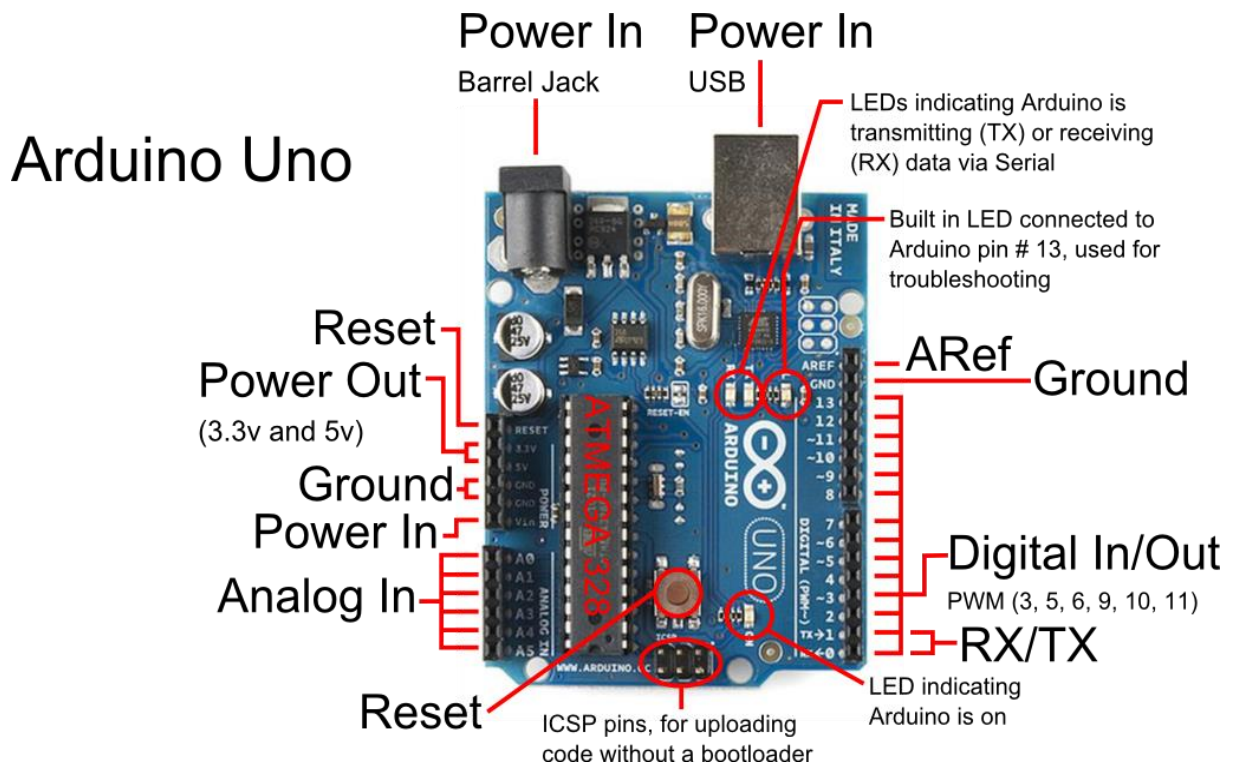


Fig.4.2.1(a): ARDUINO

These boards below use the same micro-controller, just in a different package. The Lilypad is designed for use with conductive thread instead of wire and the Arduino Mini is simply a smaller package without the USB, Barrel Jack and Power Outs. It depends on what you want to do with it really. There are two different purposes outlined above for the voltage divider, we will go over both. All of the electrical signals that the Arduino works with are either Analog or Digital. It is extremely important to understand the difference between these two types of signal and how to manipulate the information these signals represent.

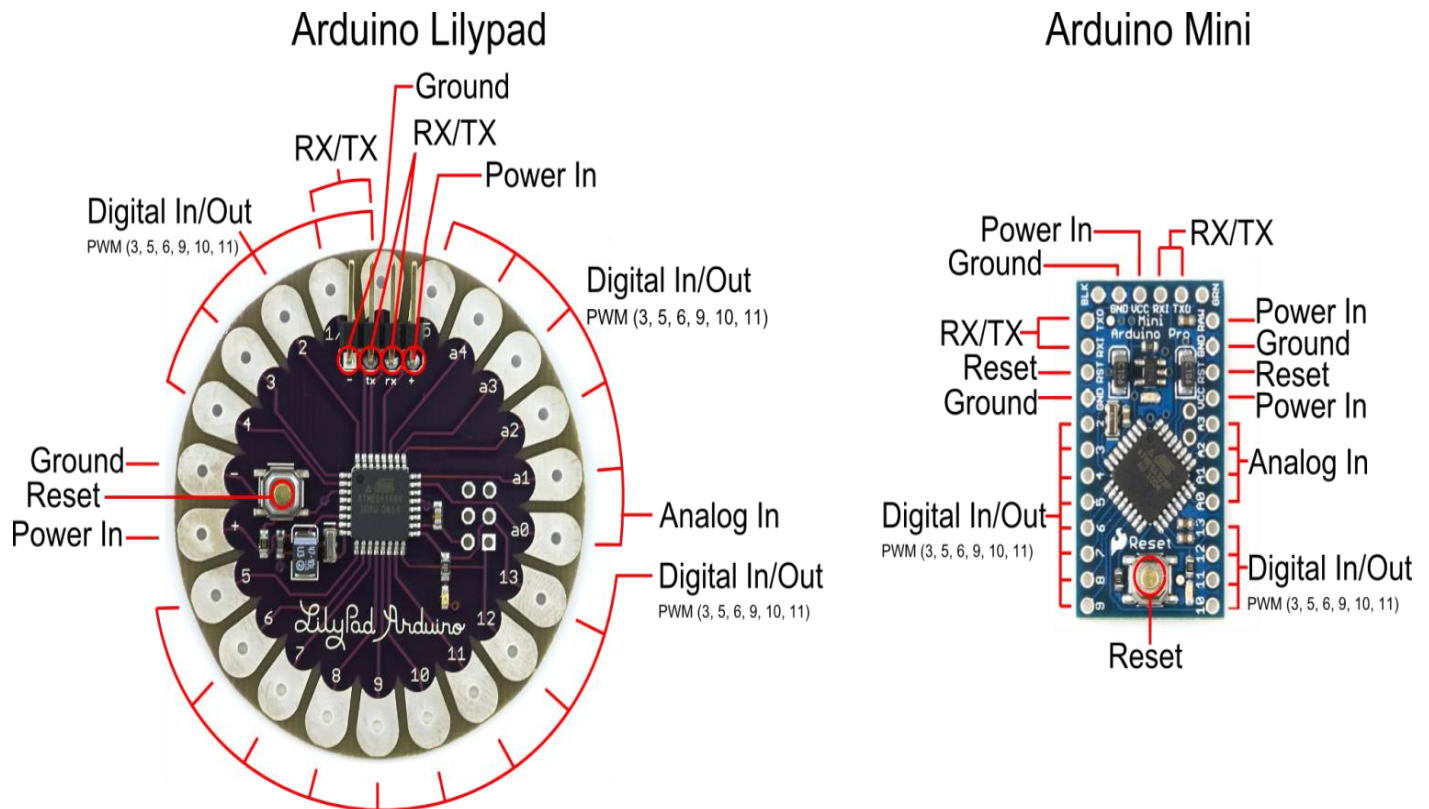


Fig.4.2.1(b) : ARDUINO LILY PAD & ARDUINO MINI

1.DIGITAL

An electronic signal transmitted as binary code that can be either the presence or absence of current, high and low voltages or short pulses at a particular frequency. Humans perceive the world in analog, but robots, computers and circuits use Digital. A digital signal is a signal that has only two states. These states can vary depending on the signal, but simply defined the states are ON or OFF, never in between.

In the world of Arduino, Digital signals are used for everything with the exception of Analog Input. Depending on the voltage of the Arduino the ON or HIGH of the Digital signal will be equal to the system voltage, while the OFF or LOW signal will always equal 0V. This is a fancy way of saying that on a 5V Arduino the HIGH signals will be a little under 5V and on a 3.3V Arduino the HIGH signals will be a little under 3.3V. To receive or send Digital signals the Arduino uses Digital pins # 0 - # 13. You may also set up your Analog In pins to act as Digital pins. To set up Analog In pins as Digital pins use the command:

pinMode(pinNumber, value);

where pinNumber is an Analog pin (A0 – A5) and value is either INPUT or OUTPUT. To setup Digital pins use the same command but reference a Digital pin for pinNumber instead of an Analog In pin. Digital pins default as input, so really you only need to set them to OUTPUT in pinMode. To read these pins use the command:

digitalRead(pinNumber);

where pinNumber is the Digital pin to which the Digital component is connected. The digitalRead command will return either a HIGH or a LOW signal. To send a Digital signal to a pin use the command:

digitalWrite(pinNumber, value);

where pinNumber is the number of the pin sending the signal and value is either HIGH or LOW.

The Arduino also has the capability to output a Digital signal that acts as an Analog signal, this signal is called Pulse Width Modulation (PWM). Digital Pins # 3, # 5, # 6, # 9, # 10 and #11 have PWM capabilities. To output a PWM signal use the command:

analogWrite(pinNumber, value);

where pinNumber is a Digital Pin with PWM capabilities and value is a number between 0 (0%) and 255 (100%). For more information on PWM see the PWM worksheets or S.I.K. circuit 12.

2.ANALOG

Humans perceive the world in analog. Everything we see and hear is a continuous transmission of information to our senses. The temperatures we perceive are never 100% hot or 100% cold, they are constantly changing between our ranges of acceptable temperatures. (And if they are out of our range of acceptable temperatures then what are we doing there?) This continuous stream is what defines analog data. Digital information, the complementary concept to Analog, estimates analog data using only ones and zeros.

In the world of Arduino an Analog signal is simply a signal that can be HIGH (on), LOW (off) or anything in between these two states. This means an Analog signal has a voltage value that can be anything between 0V and 5V (unless you mess with the Analog Reference pin). Analog allows you to send output or receive input about devices that run at percentages as well as on and off. The Arduino does this by sampling the voltage signal sent to these pins and comparing it to a voltage reference signal (5V). Depending on the voltage of the Analog signal when compared to the Analog Reference signal the Arduino then assigns a numerical value to the signal somewhere between 0 (0%) and 1023 (100%). The digital system of the Arduino can then use this number in calculations and sketches.

To receive Analog Input the Arduino uses Analog pins # 0 - # 5. These pins are designed for use with components that output Analog information and can be used for Analog Input. There is no setup necessary, and to read them use the command:

```
analogRead(pinNumber);
```

where pinNumber is the Analog In pin to which the Analog component is connected. The analogRead command will return a number including or between 0 and 1023.

The Arduino also has the capability to output a digital signal that acts as an Analog signal, this signal is called Pulse Width Modulation (PWM). Digital Pins # 3, # 5, # 6, # 9, # 10 and #11 have PWM capabilities. To output a PWM signal use the command:

```
analogWrite(pinNumber, value);
```

where pinNumber is a Digital Pin with PWM capabilities and value is a number between 0 (0%) and 255 (100%). On the Arduino UNO PWM pins are signified by a ~ sign. For more information on PWM see the PWM worksheets or S.I.K. circuit 12.

4.2.2. FLEX SENSOR

A flex sensor is a kind of sensor which is used to measure the amount of deflection otherwise bending. The designing of this sensor can be done by using materials like plastic and carbon. The carbon surface is arranged on a plastic strip as this strip is turned aside then the sensor's resistance will be changed. Thus, it is also named a bend sensor. As its varying resistance can be directly proportional to the quantity of turn thus it can also be employed like a goniometer.



Fig.4.2.2(a).FLEX SENSOR

We know that there are different types of sensors available in the market where each sensor can be used based on the application. Likewise, a bend sensor or flex sensor is one kind of sensor used to measure the quantity of bending otherwise deflection. Generally, this sensor is fixed to the exterior, and the resistance of this sensor can be changed by twisting the exterior. These sensors are applicable in the Nintendo power glove, robot whisker sensors, door sensors, otherwise the main component in making alert stuffed animal toys.

1.TYPES OF FLEX SENSOR

These sensors are classified into two types based on its size namely 2.2-inch flex sensor & 4.5-inch flex sensor. The size, as well as the resistance of these sensors, is dissimilar except the working principle. Therefore the suitable size can be preferred

based on the necessity. Here this article discusses an overview of 2.2-inch flex-sensor. This type of sensor is used in various applications like computer interface, rehabilitation, servo motor control, security system, music interface, intensity control, and wherever the consumer needs to modify the resistance throughout bending.

2.PIN CONFIGURATION

The pin configuration of the flex sensor is shown below. It is a two-terminal device, and the terminals are like p1 & p2. This sensor doesn't contain any polarized terminal such as diode otherwise capacitor, which means there is no positive & negative terminal. The required voltage of this sensor to activate the sensor ranges from 3.3V -5V DC which can be gained from any type of interfacing.

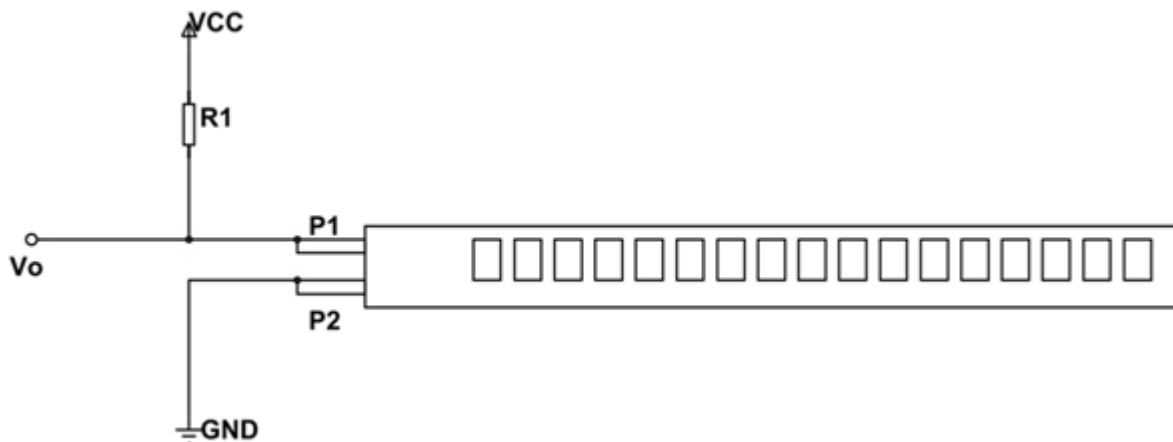


Fig.4.2.2(b).PIN CONFIGURATION OF FLEX SENSOR

- ❖ Pin P1: This pin is generally connected to the +ve terminal of the power source.
- ❖ Pin P2: This pin is generally connected to GND pin of the power source.

3.Where to Use?

The flex-sensor can be used in the following two cases.

Case1: Where you want to check whether the surface of a device or thing is leveled or not. Say you want a device to check whether a window or door is open or not. At that time a Flex sensor could be used. The sensor could be fixed at door edge and when the door opens the Flex sensor gets flexed. With the sensor being flexed its parameters changes which could be designed to provide an alert.

Case2: Where you want to measure the FLEX or BENT or ANGLE change of any instrument or device. The FLEX SENSOR internal resistance changes almost linearly with its flex angle. So by sticking the sensor to the instrument we can have the flex angle in electrical parameter of resistance.

4.How to Use FLEX SENSOR

As mentioned earlier, FLEX SENSOR is basically a VARIABLE RESISTOR whose terminal resistance increases when the sensor is bent. So this sensor resistance increases depends on surface linearity. So it is usually used to sense the changes in linearity.

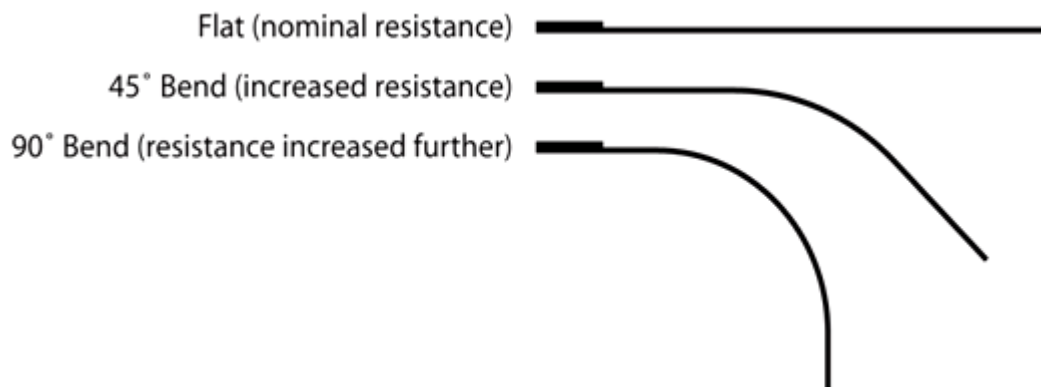


Fig.4.2.2(c). RESISTANCE OF FLEX SENSOR

This sensor works on the bending strip principle which means whenever the strip is twisted then its resistance will be changed. This can be measured with the help of any controller.

This sensor works similar to a variable resistance because when it twists then the resistance will be changed. The resistance change can depend on the linearity of the surface because the resistance will be dissimilar when it is level.

When the sensor is twisted 45° then the resistance would be dissimilar. Similarly, when this sensor is twisted to 90° then the resistance would be dissimilar. These three are the flex sensor's bending conditions.

According to these three cases, the resistance will be normal in the first case, the resistance will be double as contrasted with the first case, and the resistance will be four-time when compared with the first case. So the resistance will be increased when the angle is increased

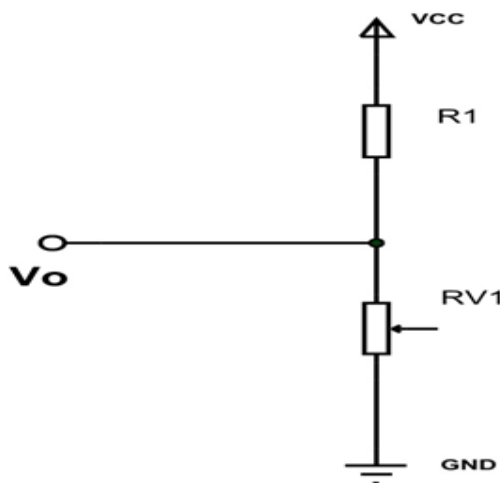


Fig.4.2.2(D).CIRCUIT DIAGRAM OF FLEX SENSOR

5.SPECIFICATIONS & FEATURES

The specifications and features of this sensor include the following.

- ❖ Operating voltage of this sensor ranges from 0V to 5V
- ❖ It can function on low-voltages.
- ❖ Power rating is 1 Watt for peak & 0.5Watt for continuous.
- ❖ Operating temperature ranges from -45°C to +80°C
- ❖ Flat resistance is 25K Ω
- ❖ The tolerance of resistance will be $\pm 30\%$
- ❖ The range of bend resistance will range from 45K -125K Ohm

6.APPLICATIONS

The applications of the flex-sensor include the following.

- ❖ Medical Instruments
- ❖ Peripherals of Computer
- ❖ Robotics
- ❖ Physical Therapy
- ❖ Virtual Motion (Gaming)
- ❖ Musical Instruments

4.2.3.MEMS SENSOR

Micro-electro-mechanical Systems (MEMS) Technology is one of the most advanced technologies that have been applied in the making of most of the modern devices like video projectors, bi-analysis chips and also car crash airbag sensors. This concept was first explained by Professor R. Howe in the year 1989. Since then many prototypes have been released and revised and has thus become an integral part of the latest mechanical products available in the market today.



Fig 4.2.3(a): MEMS SENSOR

During its early stage, the MEMS chip had two parts. One part included the main structure of the chip and the other part included everything needed for signal conditioning. This method was not successful as the total space taken by the device was larger, and thus the different parts of a single chip needed multi-assembling procedures. The output obtained from such a device had less accuracy and the mounting of such a device was difficult.

As the technology became more advanced the idea of integrating multi-chips was applied on to produce a single chip MEMS with high performance and accuracy.

The main idea behind this technology is to use some of the basic mechanical devices like cantilevers and membranes to have the same qualities of electronic circuits. To obtain such a concept, micro-fabrication process must be carried out. Though an electronic process is carried out, an MEMS device cannot be called as an

electronic circuit. MEMS duplicate a mechanical part and have holes, cantilevers, membranes, channels, and so on. But an electronic circuit has a firm and compact structure. To make MEMS from silicon process, the manufacturer must have a deep knowledge in electronics, mechanical and also about the materials used for the process.

1.MEMS Accelerometer

An accelerometer is an electromechanical device that is used to measure acceleration and the force producing it. Many types of accelerometers are available in the market today. They can be divided according to the force (static or dynamic) that is to be measured. Even today, one of the most commonly used one is the piezoelectric accelerometer. But, since they are bulky and cannot be used for all operations, a smaller and highly functional device like the MEMS accelerometer was developed. Though the first of its kind was developed 25 years ago, it was not accepted until lately, when there was need for large volume industrial applications. Due to its small size and robust sensing feature, they are further developed to obtain multi-axis sensing.

2.Working

One of the most commonly used MEMS accelerometers is the capacitive type.

The capacitive MEMS accelerometer is famous for its high sensitivity and its accuracy at high temperatures. The device does not change values depending on the base materials used and depends only on the capacitive value that occurs due to the change in distance between the plates.

If two plates are kept parallel to each other and are separated by a distance 'd', and if 'E' is the permittivity of the separating material, then capacitance produced can be written as

$$C_0 = E_0 \cdot E \cdot A/d = E_A/d$$

$$E_A = E_0 E A$$

A – Area of the electrodes

A change in the values of E, A or d will help in finding the change in capacitance and thus helps in the working of the MEMS transducer. Accelerometer

values mainly depend on the change of values of d or A .

A typical MEMS accelerometer is shown in the figure below. It can also be called a simple one-axis accelerometer. If more sets of capacitors are kept in 90 degrees to each other you can design 2 or 3-axis accelerometer. A simple MEMS transducer mainly consists of a movable microstructure or a proof mass that is connected to a mechanical suspension system and thus on to a reference frame.

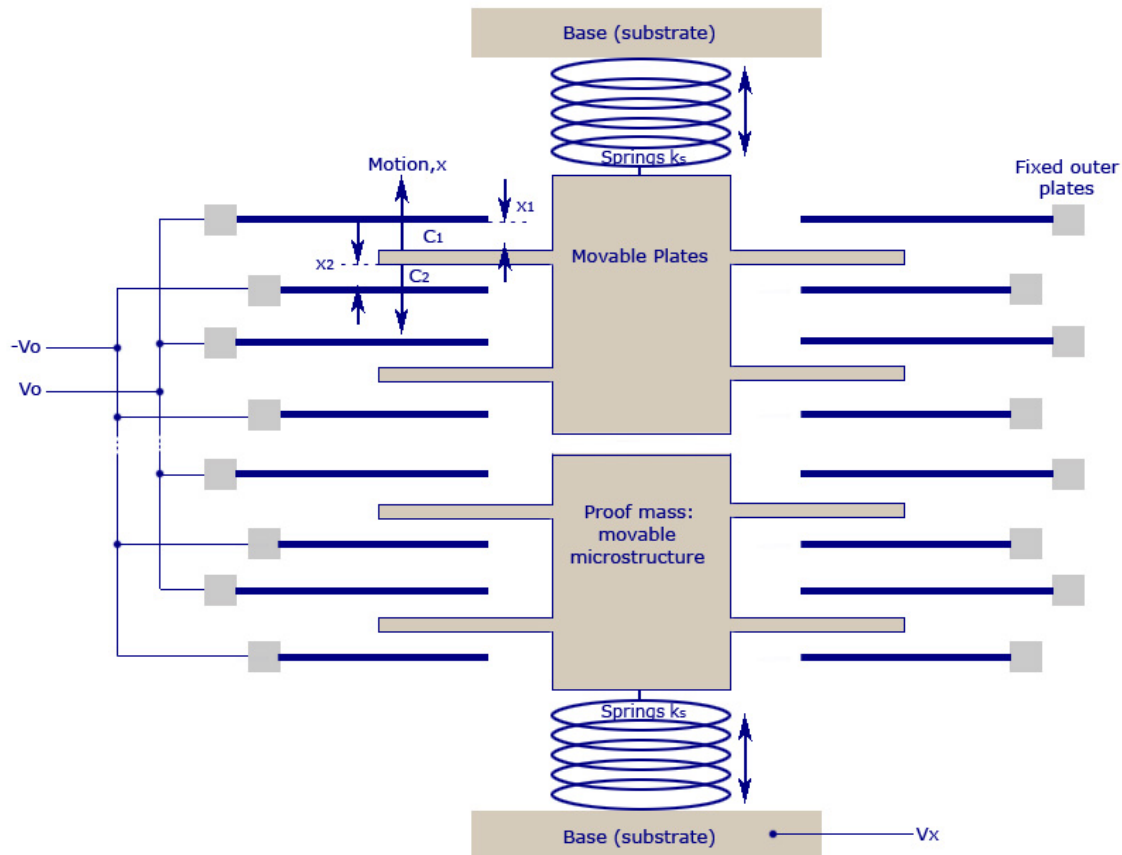


Fig 4.2.3(b):SETUP OF MEMS SENSOR

The movable plates and the fixed outer plates act as the capacitor plates. When acceleration is applied, the proof mass moves accordingly. This produces a capacitance between the movable and the fixed outer plates.

When acceleration is applied, the distance between the two plates displace as x_1 and x_2 , and they turn out to be a function of the capacitance produced.

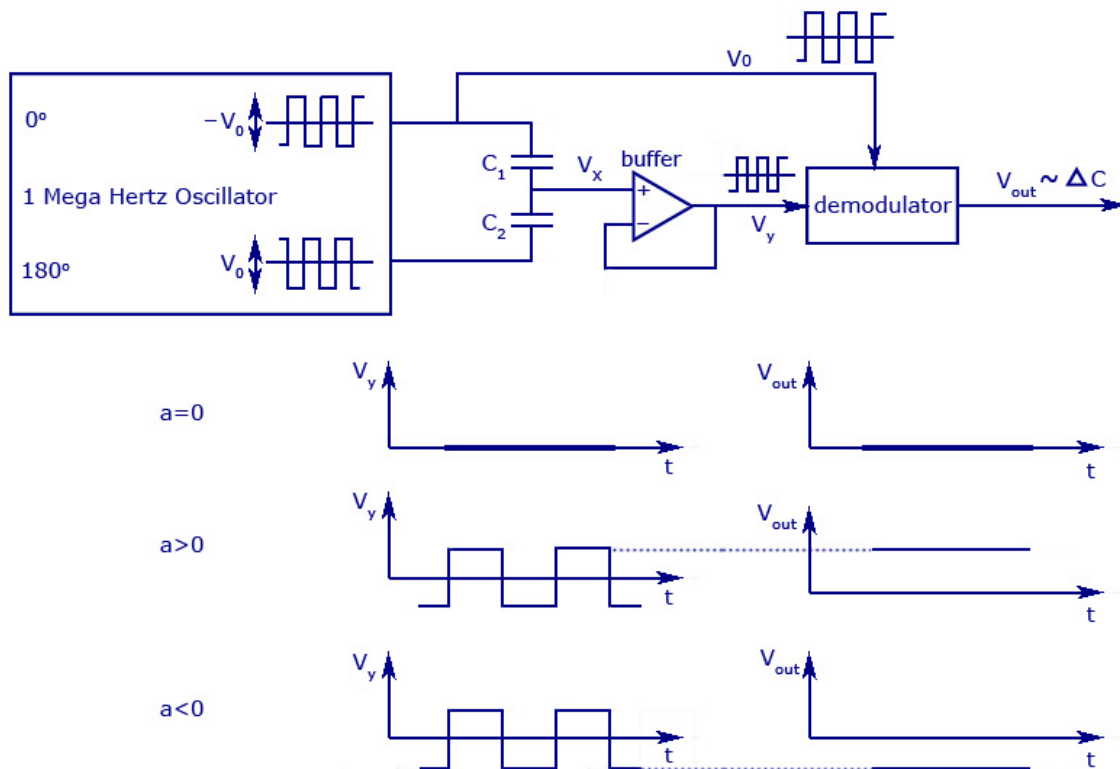
From the image above it is clear that all sensors have multiple capacitor sets. All upper capacitors are wired parallel to produce an overall capacitance C_1 and the lower ones produce an overall capacitance of C_2 . If V_x is the output voltage of the proof mass, and V_0 is the output voltage produced between the plates, then

$$(V_x + V_0) C_1 + (V_x - V_0) C_2 = 0$$

We can also write

$$V_x = V_0 [(C_2 - C_1) / (C_2 + C_1)] = (x/d) V_0$$

The figure below shows the circuit that is used to calculate the acceleration, through change in distance between capacitor plates. The output obtained for different values of acceleration is also shown graphically.



Acceleration Calculation - Capacitor Type MEMS Accelerometer

Fig 4.2.3(C):CIRCUIT DIAGRAM OF MEMS SENSOR

When no acceleration is given ($a=0$), the output voltage will also be zero.

When acceleration is given, such as ($a>0$), the value of value of V_x changes in proportion to the value of V_0 .

When a deceleration is given, such as ($a<0$), the signals V_x and V_y become negative. The demodulator produces an output equal to the sign of the acceleration, as it multiplies both the values of V_y and V_0 to produce V_{OUT} , which has the correct acceleration sign and correct amplitude.

The length of the distance, d and the proof mass weight is surprisingly very small. The proof mass weighs no more than 0.1 microgram and the output capacitance is approximately 20 aF and the plate distance is no more than 1.3 micrometers.

We must select the device in reference to its noise characteristics. If the acceleration value at low gravity condition is to be found out, the noise characteristics could easily affect its accuracy. An MEMS accelerometer is said to have three noise producing parameters – from the signal conditioning circuit, from the vibrations produced by the springs, and from the output measuring system.

3.MEMS Accelerometers – Applications

- ❖ EMS sensors are being used in latest mobile phones and gaming joysticks as step counters, user interface control, and also for switching between different modes.
- ❖ Used in mobile cameras as a tilt sensor so as to tag the orientation of photos taken.
- ❖ To provide stability of images in camcorders and also to rotate the image to and fro when you turn the mobile.
- ❖ A 3D accelerometer is used in Nokia 5500 so as to provide easier tap and change feature by which you can change mp3's by tapping on the phone when it is lying inside the pocket.
- ❖ Used to protect hard disk drives in laptops from getting damaged when the

PC falls to the ground. The device senses the free fall and automatically switches off the hard disk.

- ❖ Used in car crash airbag sensors, where it senses the sudden negative acceleration and determines the correct time to open the airbag.
- ❖ Used in real-time applications like military monitoring, missile launching, projectiles.

4. Advantages

- ❖ MEMS devices are very small and can be applicable for many mechanical purposes where large measurements are needed.
- ❖ The small size of the device has also helped in reducing its cost.
- ❖ If two or three different devices are needed to deploy a particular process, all of them can be easily integrated in an MEMS chip with the help of microelectronics. Thus, data reception, filtering, storing, transfer, interfacing, and all other processes can be carried out with a single chip.

5. Applications

- ❖ The device is highly applicable as an accelerometer, and thus can be deployed as airbag sensors or in digital cameras in order to stabilize the image.
- ❖ Can be used as a pressure sensor so as to calculate the pressure difference in blood, manifold pressure (MAP), and also tire pressure.
- ❖ It is commonly used in a gyroscope, DNA chips and also inkjet printer nozzle.
- ❖ Optical MEMS is used for making projectors, optical fiber switch and so on.
- ❖ RFMEMS is used for making antennas, filters, switches, relays, RAM's microphones, microphones, and so on.

4.2.4.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



Fig 4.2.4(a):BUZZER

1.How to use a Buzzer

A buzzer is a small yet efficient component to add sound features to our project/system. It is very small and compact 2-pin structure hence can be easily used on breadboard, Perf Board and even on PCBs which makes this a widely used component in most electronic applications. There are two types of buzzers that are commonly available. The one shown here is a simple buzzer which when powered will make a Continuous Beeeeeeppp.... sound, the other type is called a readymade buzzer which will look bulkier than this and will produce a Beep. Beep. Beep. Sound due to the internal oscillating circuit present inside it. But, the one shown here is most widely used because it can be customised with help of other circuits to fit easily in our application. This buzzer can be used by simply powering it using a DC power supply ranging from 4V to 9V. A simple 9V battery can also be used, but it is recommended to use a regulated +5V or +6V DC supply. The buzzer is normally associated with a switching circuit to turn ON or turn OFF the buzzer at required time and required interval.

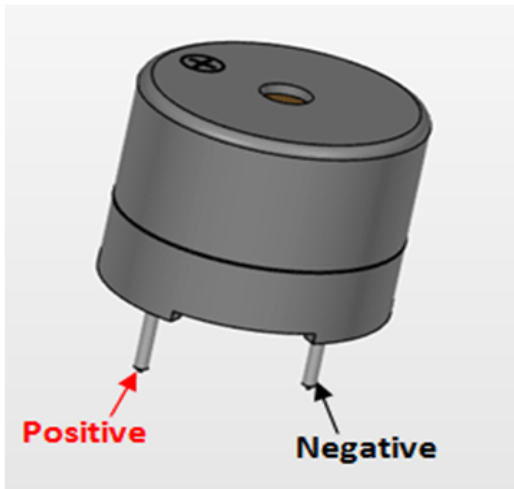


FIG 4.2.3(b): BUZZER PIN CONFIGURATION

Pin Number	Pin Name	Description
1	Positive	Identified by (+) symbol or longer terminal lead. Can be powered by 6V DC
2	Negative	Identified by short terminal lead. Typically connected to the ground of the circuit

2.Specifications

- ❖ Rated Voltage: 6V DC
- ❖ Operating Voltage: 4-8V DC
- ❖ Rated current: <30mA
- ❖ Sound Type: Continuous Beep
- ❖ Resonant Frequency: ~2300 Hz
- ❖ Small and neat sealed package
- ❖ Breadboard and Perf board friendly

3.Application

- ❖ Alarming Circuits, where the user has to be alarmed about something
- ❖ Communication equipments
- ❖ Automobile electronics, Portable equipments, due to its compact size

4.3 SOFTWARE REQUIREMENTS

4.3.1 ARDUINO SOFTWARE (IDE)

- Writing Sketches
 - File
 - Edit
 - Sketch
 - Tools
 - Help
- Sketchbook
- Tabs, Multiple Files, and Compilation
- Uploading
- Libraries
- Third-Party Hardware
- Serial Monitor
- Preferences
- Language Support
- Boards

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuino hardware to upload programs and communicate with them.

1.WRITING SKETCHES

Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension .ino. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom right hand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

NB: Versions of the Arduino Software (IDE) prior to 1.0 saved sketches with the extension .pde. It is possible to open these files with version 1.0, you will be prompted to save the sketch with the .ino extension on save.

	<i>Verify</i> Checks your code for errors compiling it.
	<i>Upload</i> Compiles your code and uploads it to the configured board. See uploading below for details. Note: If you are using an external programmer with your board, you can hold down the "shift" key on your computer when using this icon. The text will change to "Upload using Programmer"
	 <i>New</i> Creates a new sketch.

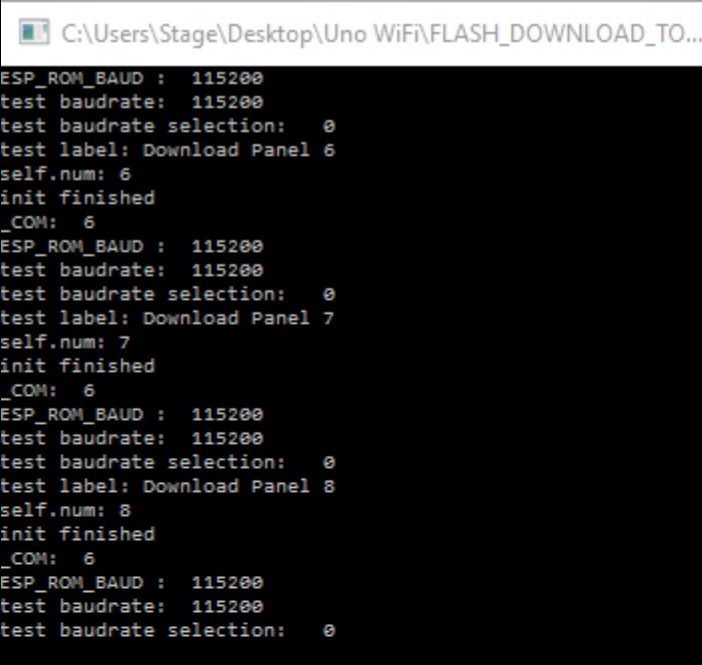
	<i>Open</i> Presents a menu of all the sketches in your sketchbook. Clicking one will open it within the current window overwriting its content. Note: due to a bug in Java, this menu doesn't scroll; if you need to open a sketch late in the list, use the File Sketchbook Menu instead.
	<i>Save</i> Saves your sketch.
	<i>Serial Monitor</i> Opens the serial monitor.

Fig 4.3.1:COMMAND WINDOW

Additional commands are found within the five menus: File, Edit, Sketch, Tools, Help. The menus are context sensitive, which means only those items relevant to the work currently being carried out are available.

2.SKETCH

❖ *Verify/Compile*

Checks your sketch for errors compiling it; it will report memory usage for code and variables in the console area.

❖ *Upload*

Compiles and loads the binary file onto the configured board through the configured Port.

❖ *Upload Using Programmer*

This will overwrite the bootloader on the board; you will need to use Tools > Burn Bootloader to restore it and be able to Upload to USB serial port again. However, it allows you to use the full capacity of the Flash memory for your sketch. Please note that this command will NOT burn the fuses. To do so a *Tools -> Burn Bootloader* command must be executed.

❖ *Export Compiled Binary*

Saves a .hex file that may be kept as archive or sent to the board using other tools.

❖ *Show Sketch Folder*

Opens the current sketch folder.

❖ *Include Library*

Adds a library to your sketch by inserting #include statements at the start of your code. For more details, see libraries below. Additionally, from this menu item you can access the Library Manager and import new libraries from .zip files.

❖ *Add File...*

Adds a source file to the sketch (it will be copied from its current location). The new file appears in a new tab in the sketch window. Files can be removed from the sketch using the tab menu accessible clicking on the small triangle icon below the serial monitor one on the right side of the toolbar.

3.TABS, MULTIPLE FILES, AND COMPILATION

Allows you to manage sketches with more than one file (each of which appears in its own tab). These can be normal Arduino code files (no visible extension), C files (.c extension), C++ files (.cpp), or header files (.h).

4.Serial monitor

Displays serial data being sent from the Arduino or Genuino board (USB or serial board). To send data to the board, enter text and click on the "send" button or press enter. Choose the baud rate from the drop-down that matches the rate passed to Serial.begin in your sketch. Note that on Windows, Mac or Linux, the Arduino or Genuino board will reset (rerun your sketch execution to the beginning) when you connect with the serial monitor. You can also talk to the board from Processing, Flash, MaxMSP, etc (see the interfacing page for details).

CHAPTER 5

RESULT AND DISCUSSION

5.1 RESULT

In this project, we are using a arduino IDE software to detect the human sitting posture to be monitored without being conscious of the sensor and developed sensitive cushion could detect the body moving of a subject on a chair . The output is connected to a Buzzer.

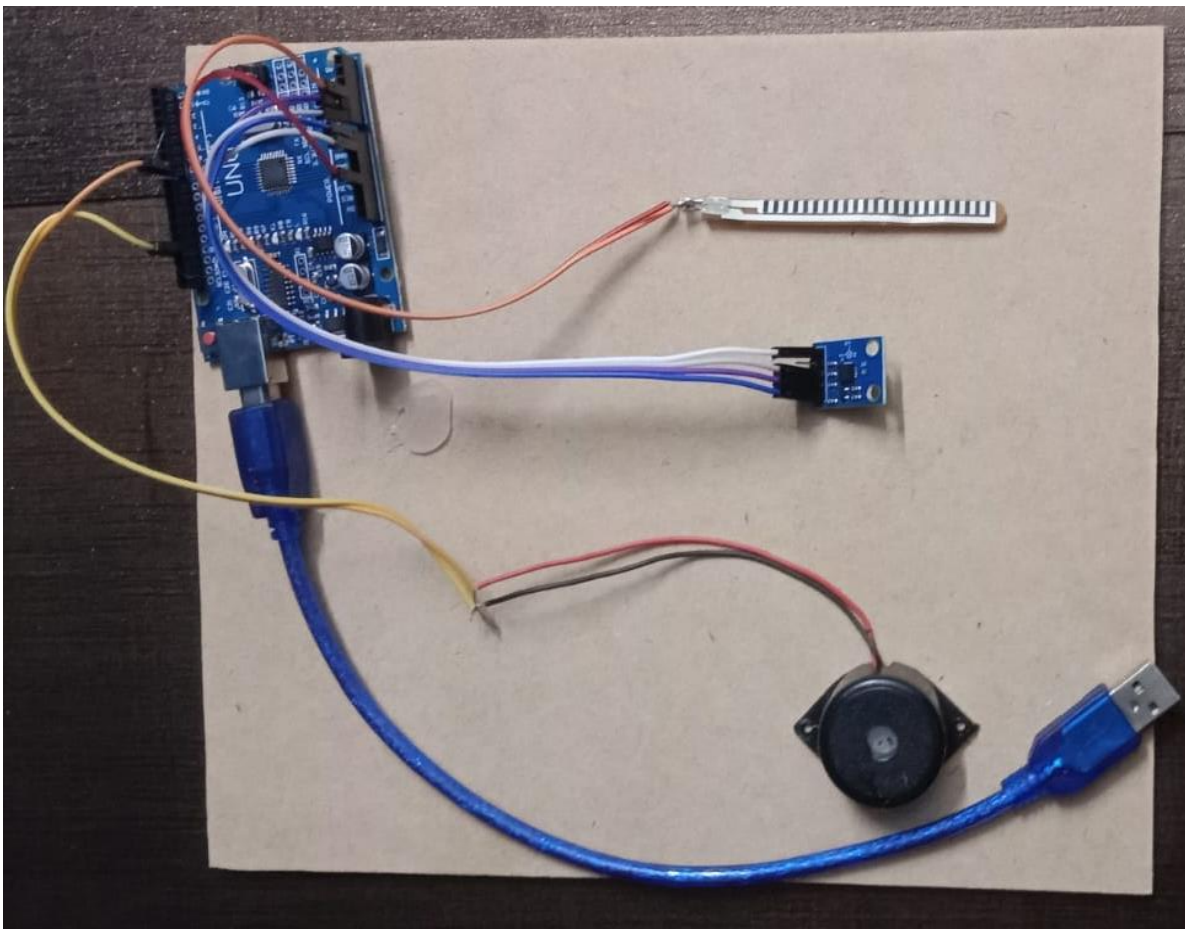
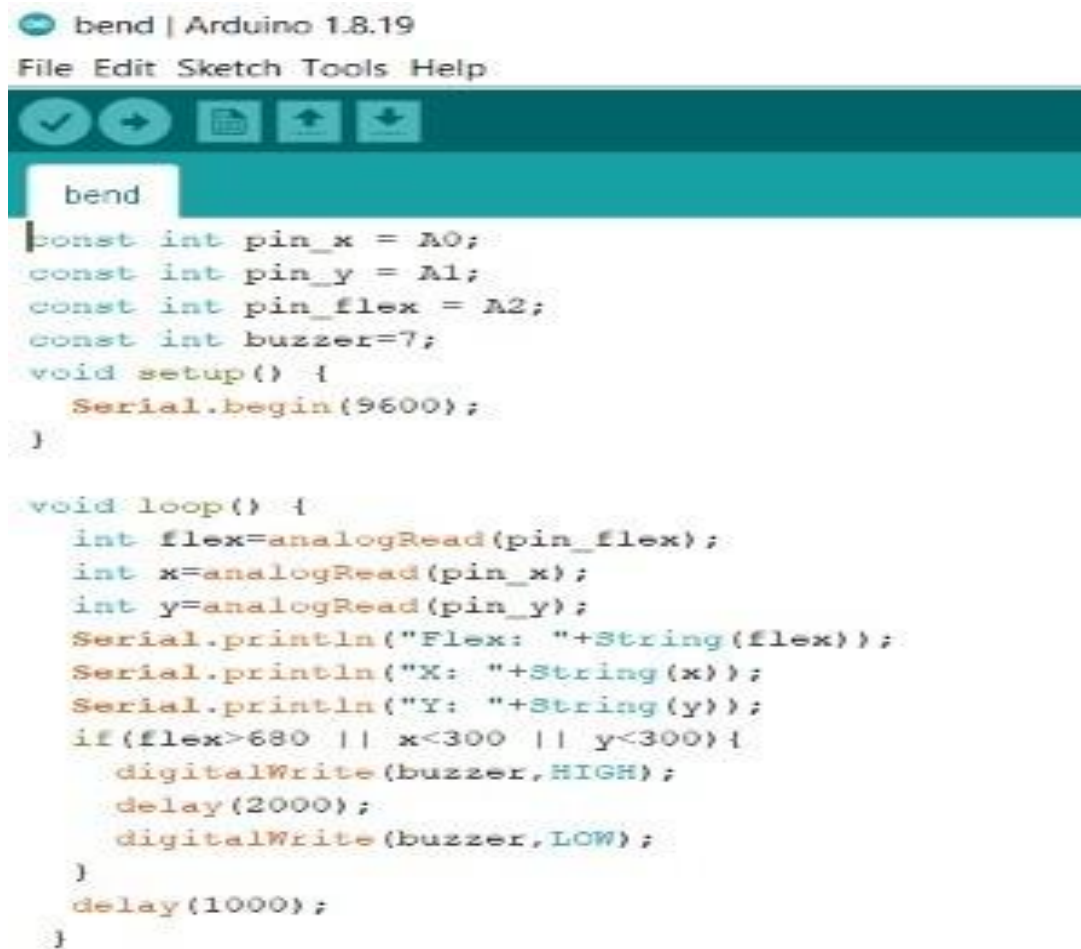


Fig: 5.1.1: HARDWARE SETUP



The screenshot shows the Arduino IDE interface with the file 'bend' open. The code is as follows:

```
bend | Arduino 1.8.19
File Edit Sketch Tools Help

bend
const int pin_x = A0;
const int pin_y = A1;
const int pin_flex = A2;
const int buzzer=7;
void setup() {
  Serial.begin(9600);
}

void loop() {
  int flex=analogRead(pin_flex);
  int x=analogRead(pin_x);
  int y=analogRead(pin_y);
  Serial.println("Flex: "+String(flex));
  Serial.println("X: "+String(x));
  Serial.println("Y: "+String(y));
  if(flex>680 || x<300 || y<300){
    digitalWrite(buzzer,HIGH);
    delay(2000);
    digitalWrite(buzzer,LOW);
  }
  delay(1000);
}
```

Fig: 5.1.2: SOFTWARE SETUP

5.2 DISCUSSION

This section provides about a review of applications of sensors and softwares used in this project. After the successful implementation and testing of coding the following output was achieved. The screenshots of the outputs have been displayed above the figure.

- Hardware setup
- Software setup

CHAPTER – 6

SUMMARY AND CONCLUSION

6.1 SUMMARY:

A Flex and MEMS sensor detects the sitting posture and a Arduino analyses this data to determine the angle ($\text{flex} > 680 \parallel x < 300 \parallel y < 300$) and direction of shoulder and spine movements .

- After that has been determined, the Arduino generates a specific message for that direction, which it will then send to the buzzer to produce alarm .
- This device contain auditory alarms to notify the person's sitting posture to convey whether the posture is between the given level in the sensor.

6.2 CONCLUSION

As an outcome of the pursued research work, the real time wireless bend monitoring system is depicted in this thesis. This prototype of Real time bend monitoring system was designed and built to monitor the people's health wirelessly at any time and make precautionary measures to avoid pain or illness. The wearable technology is implemented with this system for the people to feel comfortable without any add-ons to there daily wear that would not make them feel conscious.

REFERENCE

- [1].Y.P. Tsang, K.L. Choy*, T.C. Poon, G.T.S. Ho, C.H. Wu, H.Y. Lam, P.S. Koo, H.Y. Ho Seiji Sasaki; Chinthaka Premachandra Seiji Sasaki; Chinthaka Premachandra- Head Posture Estimation by Deep Learning Using 3-D Point Cloud Data From a Depth Sensor (2021)
- [2]. Askat Kuzdeuov; Matteo Rubagotti; Huseyin Atakan Varol- Neural Network Augmented Sensor Fusion for Pose Estimation of Tensegrity Manipulators (2020)
- [3]. Niven Junliang Sie; Sutthiphong Srigrarom; Sunan Huang- Field Test Validations of Vision-based Multi-camera Multi-drone Tracking and 3D Localizing with Concurrent Camera Pose Estimation (2021)
- [4].Chao Yang; Xuyu Wang; Shiwen Mao- RFID-Pose: Vision-Aided Three-Dimensional Human Pose Estimation With Radio-Frequency Identification 2021
- [5].Mariusz P. Wilk; Michael Walsh; Brendan O'Flynn-]. Multimodal Sensor Fusion for LowPower Wearable Human Motion Tracking Systems in Sports Applications 2021
- [6].Chengyu Li, Di Liu, Chaoqun Xu, Ziming Wang, Sheng Shu, Zhuoran Sun, Wei Tang & Zhong Lin Wang.-Sensing of joint and spinal bending or stretching via a retractable and wearable badge reel.-Nature Communications.-2021
- [7]. Ferdews Tlili, Rim Haddad, Ridha Bouallegue & Neila Mezghani-A Real-time Posture Monitoring System Towards Bad Posture Detection-Wireless Personal Communications-2021

APPENDIXES

```
const int pin_x = A0;
const int pin_y = A1;
const int pin_flex = A2;
const int buzzer=7;
void setup() {
  Serial.begin(9600);
}

void loop() {
  int flex=analogRead(pin_flex);
  int x=analogRead(pin_x);
  int y=analogRead(pin_y);
  Serial.println("Flex: "+String(flex));
  Serial.println("X: "+String(x));
  Serial.println("Y: "+String(y));
  if(flex>680 || x<300 || y<300){
    digitalWrite(buzzer,HIGH);
    delay(2000);
    digitalWrite(buzzer,LOW);
  }
  delay(1000);
}
```