SkinKonnect

By

Daman Ahmed
01-133172-021
Hadeed-ur-Rehman
01-133172-028
Mubashar Mansoor
01-133172-055

Supervised by

Dr. Imran Fareed Nizami

{Session 2017-21}

A Report is submitted to the Department of Electrical Engineering, Bahria University, Islamabad.
In partial fulfillment of requirement for the degree of BS(EE) .

Certificate

We accept the work contained in this report as a confirmation to the required standard for the partial fulfillment of the degree of BS(EE).

_____	_____
Head of Department	Supervisor
_____	_____
Internal Examiner	External Examiner

Dedication

This project dedicated to over loving parents who are, our biggest support system, our teachers, and many friends.

DAMAN AHMED
HADEED UR REHMAN
MUBASHAR MANSOOR

Acknowledgements

With the grace of Almighty Allah, this project has been completed. Many people have played a pivotal role in the completion of this project. Deepest gratitude to the supervisor of this project Dr. Imran Fareed whose constant support, encouragement and efforts helped a lot to achieve success. A Great thanks to all the teachers that provided us with knowledge, confidence, and a platform to strive hard for what we want. Thanks to Dr. Junaid Imtiaz for giving the motivation to complete this project properly. Furthermore, a profound appreciation to parents whose prayers helped a lot in this journey.

Abstract

SkinKonnect, an innovation that appropriates the human body for acoustic transmission, permitting the skin to be used as an input surface. It resolves the location of finger taps on the arm and hand by analyzing mechanical vibrations that propagate through the body. It collects these signals using a novel array of sensors worn a band. This approach provides an always available, naturally portable, and on-body finger input system. To further outline the utility of our methodology, it is finished up with a few proof- of-idea applications we created. The arm band is a crude prototype. The next generation could be made considerably smaller – likely easily fitting into a wristwatch, from there it's fairly simple to associate those tappable areas with different commands in an interface, just as different keystrokes and mouse clicks perform different functions on a computer.

Table of Contents

List of Figures

Chapter # 1

Introduction

1.1 Project Background

Devices with significant computational power and capabilities can now be easily carried on our bodies. However, their small size typically leads to limited interaction space and consequently diminishes their usability and functionality. Since we cannot simply make buttons and screens larger without losing the primary benefit of small size, we consider alternative approaches that enhance interactions with small mobile systems. One option is to opportunistically appropriate surface area from the environment for interactive purposes. For example, a technique that allows a small mobile device to turn tables on which it rests into a gestural finger input canvas. However, tables are not always present, and are not usable in a mobile context, However, there is one surface that has been previous overlooked as an input canvas, and one that happens to always travel with us: our skin. Furthermore, proprioception

Our sense of how our body is configured in three-dimensional space – allows us to accurately interact with our bodies in an eyes-free manner. For example, we can readily flick each of our fingers, touch the tip of our nose, and clap our hands together without visual assistance. Few external input devices can claim this accurate, eyes-free input characteristic and provide such a large interaction area.

SkinKonnect is a method that allows the body to be appropriated for finger input using a novel, non-invasive, wearable bio-acoustic sensor. In SkinKonnect, a console, menu, or different designs are radiated onto a client's palm and lower arm from a Pico projector inserted in an armband. An acoustic indicator in the armband at that point figures out which a piece of the presentation is initiated by the client's touch. As the specialists clarify, varieties in bone thickness, size, and mass, just as separating impacts from delicate tissues and joints,

mean diverse skin areas are acoustically particular. Their product matches sound frequencies to explicit skin areas, permitting the framework to figure out which "skin button" the client squeezed.

The model framework at that point utilizes remote innovation like Bluetooth to send the orders to the gadget being controlled, for example, a telephone, iPod, or PC. Twenty volunteers who have tried the framework have given positive input on the simplicity of route. The specialists state the framework likewise functions admirably when the client is strolling or running. It is an interesting idea to use human body as an input device. This idea will minimize and create effectiveness in using spaces. This way what we need is just all the possible skin in our body to be used as the input device, from head until the tip toe.

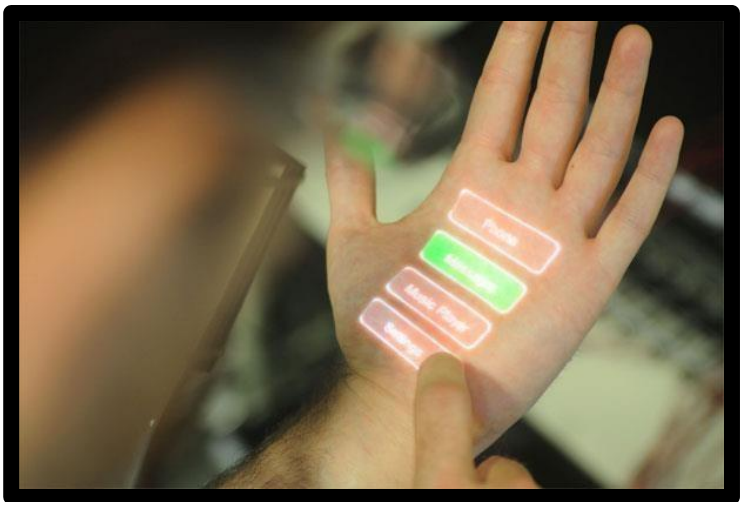


Figure 1.1: Skin as a touchscreen interface.

1.2 Problem Description

The reason of the exploration was prodded from the struggles between the scaling down of versatile gadgets and the excess requirement for an agreeable method of cooperating with said items. As the gadgets gets more modest, so does the territory to control them, for example, fastens and contact screens.

The quest for an outside surface zeroed in on the one thing everyone consistently hefts around with them, their skin. With the utilization of the sensors, one can handle an electronic gadget essentially by tapping their skin in predestinated places. The clearest utilization of this innovation is convenient buyer hardware, like light, fan, motor or tv.

1.3 Project Objectives

SkinKonnect uses a series of sensors to track where a user taps on his arm. Previous attempts at using projected interfaces used motion-tracking to determine where a person tap. SkinKonnect uses a different and novel technique: Its "listens" to the vibrations in our body. Tapping on different parts of your arm create different kinds of vibrations depending on the amount and shape of bones, tendons, and muscle in that specific area. SkinKonnect sensors can track those vibrations using an armband and discern where the user tapped. The arm band is a crude prototype. The next generation could be made considerably smaller – likely easily fitting into a wristwatch. From there it is simple to associate those tappable areas with different commands in an interface, just as different keystrokes and mouse clicks perform different functions on a computer.

When coupled with a small projector, SkinKonnect can simulate a menu interface like the ones used in other kinds of electronics. Tapping on different areas of the arm and hand allow users to control the electrical appliances around them. SkinKonnect could also be used without a visual interface. Different areas on the arm and fingers simulate common commands for these tasks, and a user could tap them without even needing to look.

1.4 Project Scope

The SkinKonnect technology has a lot of future implementations right now the technology is just controlling only four devices around the user. The idea of using skin as an input is

so attractive that in future it can have immense applications like using a whole mobile phone on your arm.

1.5 Thesis structure

This thesis further contains the following chapters:

1. Literature Review
2. Requirement Specifications
3. System Design
4. System Implementation
5. System Testing and Evaluation
6. Conclusion
7. References
8. Appendices

Chapter # 2

Literature Review

2.1 Motivation from the Technology

There exists a technology named as Skinput, that helps you to use your mobile phone functions on your skin. Skinput is a technology that appropriates the human body for acoustic transmission, allowing the skin to be used as an input surface. The location of finger taps on the arm and hand is resolved by analyzing mechanical vibrations that propagate through the body. These signals are Seminar Report 2011 Skinput Technology collected using a novel array of sensors worn as an armband. This approach provides an always available, naturally portable, and on-body finger input system. To expand the range of sensing modalities for always available input systems, we introduce Skinput, a novel input technique that allows the skin to be used as a finger input surface. In our prototype system, we choose to focus on the arm (although the technique could be applied elsewhere). This is an attractive area to appropriate as it provides considerable surface area for interaction, including a contiguous and flat area for projection. Furthermore, the forearm and hands contain a complex assemblage of bones that increases acoustic distinctiveness of different locations. To capture this acoustic information, we developed a wearable armband that is non-invasive and easily removable.

2.2 Bio-Acoustics

When a finger taps the skin, several distinct forms of acoustic energy are produced. Some energy is radiated into the air as sound waves; this energy is not captured by the Skinput system. Seminar Report 2011 Skinput Technology Among the acoustic energy transmitted through the arm, the most readily visible are transverse waves, created by the displacement of the skin from a finger impact (Figure 2.1)

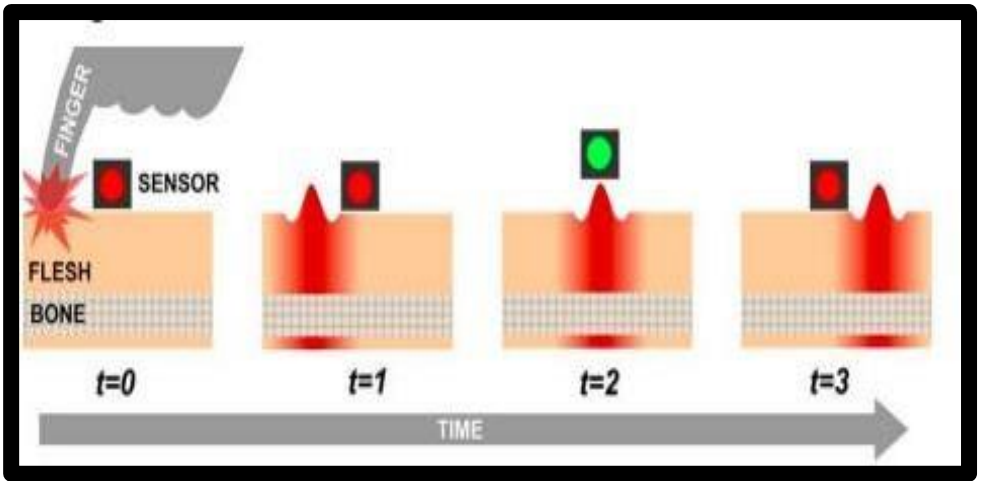


Figure 2.1: Tapping vibrations.

Transverse wave propagation: Finger impacts displace the skin, creating transverse waves (ripples). The sensor is activated as the wave passes underneath it. When shot with a high-speed camera, these appear as ripples, which propagate outward from the point of contact. The amplitude of these ripples is correlated to both the tapping force and to the volume and compliance of soft tissues under the impact area. In general, tapping on soft regions of the arm creates higher amplitude transverse waves than tapping on boney areas which have negligible compliance. In addition to the energy that propagates on the surface of the arm, some energy is transmitted inward, toward the skeleton (Figure 2.2)

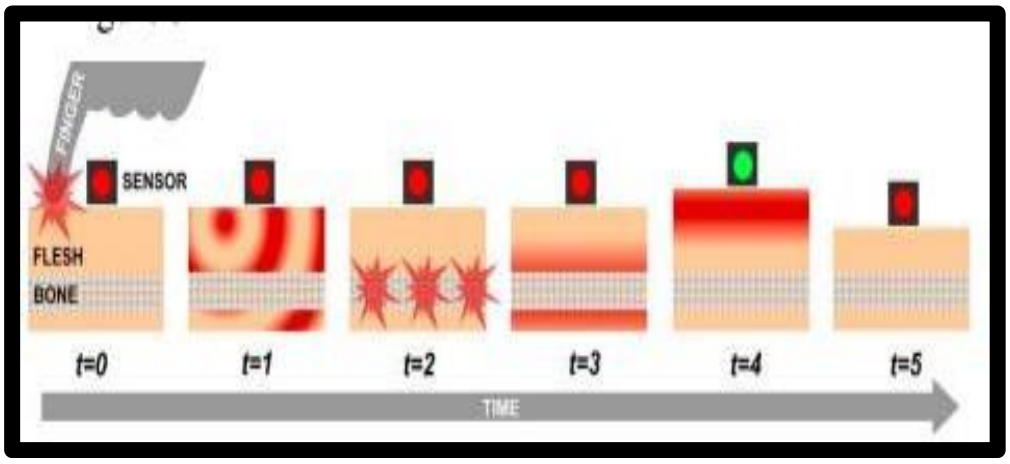


Figure 2.2 : Energy Transmission

Longitudinal wave propagation: Finger impacts create longitudinal (compressive) waves that cause internal skeletal structures to vibrate. This, in turn, creates longitudinal waves that emanate outwards from the bone (along its entire length) toward the skin.

These longitudinal (compressive) waves travel through the soft tissues of the arm, exciting the bone, which is much less deformable than the soft tissue but can respond to mechanical excitation by rotating and translating as a rigid body. This excitation vibrates soft tissues surrounding the entire length of the bone, resulting in new longitudinal waves that Seminar Report 2011 Skinput Technology propagate outward to the skin. We highlight these two separate forms of conduction – transverse waves moving directly along the arm surface, and longitudinal waves moving into and out of the bone through soft tissues – because these mechanisms carry energy at different frequencies and over different distances. Higher frequencies propagate more readily through bone than through soft tissue, and bone conduction carries energy over larger distances than soft tissue conduction. While we do not explicitly model the specific mechanisms of conduction, or depend on these mechanisms for our analysis, we do believe the success of our technique depends on the complex acoustic patterns that result from mixtures of these modalities.

Similarly joints play an important role in making tapped locations acoustically distinct. Bones are held together by ligaments, and joints often include additional biological structures such as fluid cavities. This makes joints behave as acoustic filters. In some cases, these may simply dampen acoustics; in other cases, these will selectively attenuate specific frequencies, creating location specific acoustic signatures.

2.3 Sensing

To capture the rich variety of acoustic information we evaluated many sensing technologies, including bone conduction microphones, conventional microphones coupled

with stethoscopes, piezo contact microphones and accelerometers. However, these transducers were engineered for very different applications than measuring acoustics transmitted through the human body. Most of the mechanical sensors are engineered to provide relatively flat response curves over the range of frequencies that is relevant to our signal. This is a desirable property for most applications where a faithful representation of an input signal – uncolored by the properties of the transducer – is desired. However, because only a specific set of frequencies is conducted through the arm in response to tap input, a flat response curve leads to the capture of irrelevant frequencies and thus to a high signal- to-noise ratio. While bone conduction microphones might seem a suitable choice for Skinput, these devices are typically engineered for capturing human voice, and filter out energy below the range of human speech (whose lowest frequency is around 85Hz). Thus, most sensors in this category were not especially sensitive to lower-frequency signals (e.g., 25Hz.) To overcome these challenges, we moved away from a single sensing element with a flat response curve, to an array of highly tuned vibration sensors. Specifically, we employ small piezo film.

By adding small weights to the end of the cantilever, we can alter the resonant frequency, allowing the sensing element to be responsive to a unique, narrow, low-frequency band of the acoustic spectrum. Adding more mass lowers the range of excitation to which a sensor responds. Additionally, the cantilevered sensors were naturally insensitive to forces parallel to the skin (e.g., shearing motions caused by stretching). Thus, the skin stretches induced by many routine movements (e.g., reaching for a doorknob) tends to be attenuated. However, the sensors are highly responsive to motion perpendicular to the skin plane – perfect for capturing transverse surface waves and longitudinal waves emanating from interior structures . Finally, our sensor design is relatively inexpensive and can be manufactured in a very small form factor, rendering it suitable for inclusion in future

mobile devices (e.g., an arm mounted audio player).

2.4 Processing

In our prototype system, we employ a Mackie Onyx 1200F audio interface to digitally capture data from the ten sensors. This was connected via Fire wire to a conventional desktop computer, where a thin client written in C interfaced with the device using the Audio Stream Input/ Output (ASIO) protocol. Each channel was sampled at 5.5kHz, a sampling rate that would be considered too low for speech or environmental audio but was able to represent the relevant spectrum of frequencies transmitted through the arm. This reduced sample rate (and consequently low processing bandwidth) makes our technique readily portable to embedded processors. Data was then sent from our thin client over a local socket to our primary application, written in Java. This program performed three key functions. First, it provided a live visualization of the data from our ten sensors, which was useful in identifying acoustic features (Figure 6Second, its segmented inputs from the data stream into independent instances). (taps). Third, it classified these input instances.

The audio stream was segmented into individual taps using an absolute exponential average of all ten channels (Figure 6, red waveform). When an intensity threshold was exceeded (Figure 6, upper blue line), the program recorded the timestamp as a potential start of a tap. If the intensity did not fall below a second, independent "closing" threshold (Figure 6, lower purple line) between 100ms and 700ms after the onset crossing (a duration we found to be the common for finger impacts), the event was discarded. If start and end crossings were detected that satisfied these criteria, the acoustic data in that period (plus a 60ms buffer on either end) was considered an input event.

Although simple, this heuristic proves to be highly robust, mainly due to the extreme noise suppression provided by our sensing approach. After an input has been segmented, the waveforms are analyzed. The highly discrete nature of taps (i.e., point impacts) meant

acoustic signals were not particularly expressive over time. Signals simply diminished in intensity overtime. Thus, features are computed over the entire input window and do not capture any temporal dynamics. We employ a brute force machine learning approach, computing features in total, many of which are derived combinatorically.

These features are passed to a Support Vector Machine (SVM) classifier. Our software uses the implementation provided in the Weka machine learning toolkit. Other, more sophisticated classification techniques and features could be employed. Before the SVM can classify input instances, it must first be trained to the user and the sensor position. This stage requires the collection of several examples for each input location of interest. When using Skinput to recognize live input, the same acoustic features are computed on-the fly for each segmented input. These are fed into the trained SVM for classification. We use an event model in our software – once an input is classified, an event associated with that location is instantiated. Any interactive features bound to that event are fired. We readily achieve interactive speeds.

2.5 Accuracy

The classification accuracies for the test phases in the five different conditions. Overall, classification rates were high, with an average accuracy across conditions of 87.6%. Additionally, we present preliminary results exploring the correlation between classification accuracy and factors such as BMI, age, and sex.

i. Five Fingers

Despite multiple joint crossings and ~40cm of separation between the input targets and sensors, classification accuracy remained high for the five-finger condition, averaging 87.7% (SD=10.0%, chance=20%) across participants. Segmentation, as in other conditions, was essentially perfect. Inspection of the confusion matrices showed no systematic errors in the classification, with errors tending to be evenly distributed over the other digits. When classification was incorrect, the system believed the input to be an adjacent finger 60.5% of the time; only marginally above prior probability (40%). This suggests there are only limited acoustic continuities between the fingers. The only potential exception to this was in the case of the pinky, where the ring finger constituted 63.3% percent of the misclassifications.

ii. Whole Arm

Participants performed three conditions with the whole-arm location configuration. The below-elbow placement performed the best, posting a 95.5% (SD=5.1%, chance=20%) average accuracy. This is not surprising, as this condition placed the sensors closer to the input targets than the other conditions. Moving the sensor above the elbow reduced accuracy to 88.3% (SD=7.8%, chance=20%), a drop of 7.2%. This is almost certainly related to the acoustic loss at the elbow joint and the additional 10cm of distance between the sensor and input targets. Figure 8 shows these results. The eyes-free input condition

yielded lower accuracies than other conditions, averaging 85.0% (SD=9.4%, chance=20%). This represents a 10.5% drop from its vision assisted, but otherwise identical counterpart condition. It was apparent from watching participants complete this condition that targeting precision was reduced. In sighted conditions, participants appeared to be able to tap locations with perhaps a 2cm radius of error. Although not formally captured, this margin of error appeared to double or triple when the eyes were closed. We believe that additional training data, which better covers the increased input variability, would remove much of this deficit. We would also caution designers developing eyes-free, on-body interfaces to carefully consider the locations participants can tap accurately.

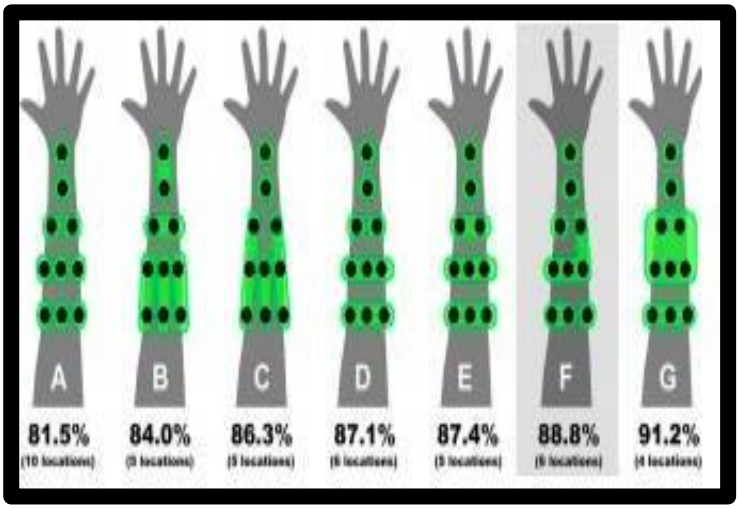


Figure 2.3 : Tapping Locations On Arm

iii. Forearm

Classification accuracy for the ten-location forearm condition stood at 81.5% (SD=10.5%, chance=10%), a surprisingly strong result for an input set we devised to push our system's sensing limit (K=0.72, considered very strong). Following the experiment, we considered different ways to improve accuracy by collapsing the ten locations into larger input groupings. The goal of this exercise was to explore the tradeoff between classification accuracy and number of input locations on the forearm, which represents a particularly

valuable input surface for application designers. We grouped targets into sets based on what we believed to be logical spatial groupings (Figure 2.2, A-E and G). In addition to exploring classification accuracies for layouts that we considered to be intuitive, we also performed an exhaustive search (programmatically) over all possible groupings. For most location counts, this search confirmed that our intuitive groupings were optimal; however, this search revealed one plausible, although irregular, layout with high accuracy at six input locations (Figure 2.2, F). Unlike in the five-fingers condition, there appeared to be shared acoustic traits that led to a higher likelihood of confusion with adjacent targets than distant ones. This effect was more prominent laterally than longitudinally.

iv. BMI Effects

Early on, we suspected that our acoustic approach was susceptible to variations in body composition. This included, most notably, the prevalence of fatty tissues and the density/mass of bones. These, respectively, tend to dampen or facilitate the transmission of acoustic energy in the body. To assess how these variations affected our sensing accuracy, we calculated each participant's body mass index (BMI) from self-reported weight and height. Data and observations from the experiment suggest that high BMI is correlated with decreased accuracies.

The participants with the three highest BMIs (29.2, 29.6, and 31.9 – representing borderline

obese to obese) produced the three lowest average accuracies. Figure 10 illustrates this significant disparity - here participants are separated into two groups, those with BMI

greater and less than the US national median, age and sex adjusted ($F_{1,12}=8.65$, $p=.013$). Other factors such as age and sex, which may be correlated to BMI in specific populations, might also exhibit a correlation with classification accuracy. For example, in our participant pool, males yielded higher classification accuracies than females, but we expect

that this is an artifact of BMI correlation in our sample, and probably not an effect of sex directly.

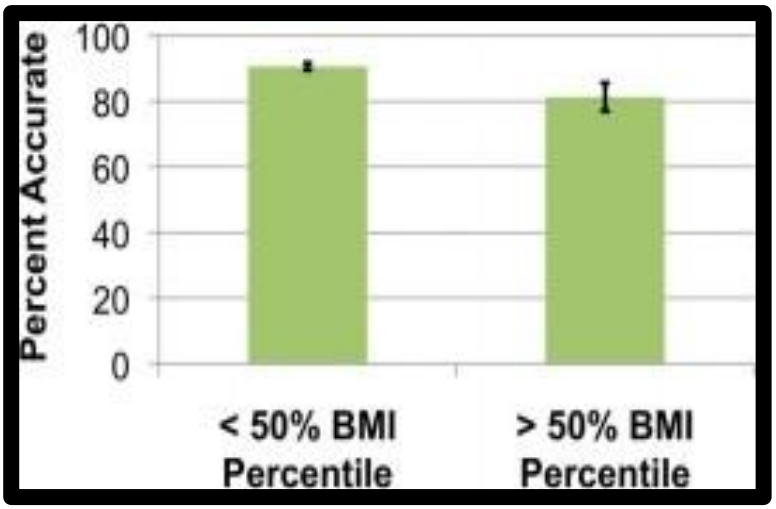


Figure 2.4 : Body Mass Index

Chapter # 3

Requirement Specifications

3.1 Existing System

Skinput is a technology that appropriates the human body for acoustic transmission, allowing the skin to be used as an input surface. The location of finger taps on the arm and hand is resolved by analyzing mechanical vibrations that propagate through the body. These signals are collected using a novel array of sensors worn as an armband. This approach provides an always available, naturally portable, and on-body finger input system. To expand the range of sensing modalities for always available input systems, we introduce Skinput, a novel input technique that allows the skin to be used as a finger input surface. In our prototype system, we choose to focus on the arm (although the technique could be applied elsewhere). This is an attractive area to appropriate as it provides considerable surface area for interaction, including a contiguous and flat area for projection. Furthermore, the forearm and hands contain a complex assemblage of bones that increases acoustic distinctiveness of different locations. To capture this acoustic information, we developed a wearable armband that is non-invasive and easily removable.

3.2 Proposed System

SkinKonnect uses a series of sensors to track where a user taps on his arm. Previous attempts at using projected interfaces used motion-tracking to determine where a person tap. SkinKonnect uses a different and novel technique: Its "listens" to the vibrations in our body. Tapping on different parts of your arm create different kinds of vibrations depending on the amount and shape of bones, tendons, and muscle in that specific area. SkinKonnect sensors can track those vibrations using an armband and discern where the user tapped. The arm band is a crude prototype. The next generation could be made considerably smaller – likely easily fitting into a wristwatch. From there it is simple to associate those tappable areas with different commands in an interface, just as different keystrokes and mouse clicks perform different functions on a computer.

When coupled with a small projector, SkinKonnect can simulate a menu interface like the ones used in other kinds of electronics. Tapping on different areas of the arm and hand allow users to control the electrical appliances around them.

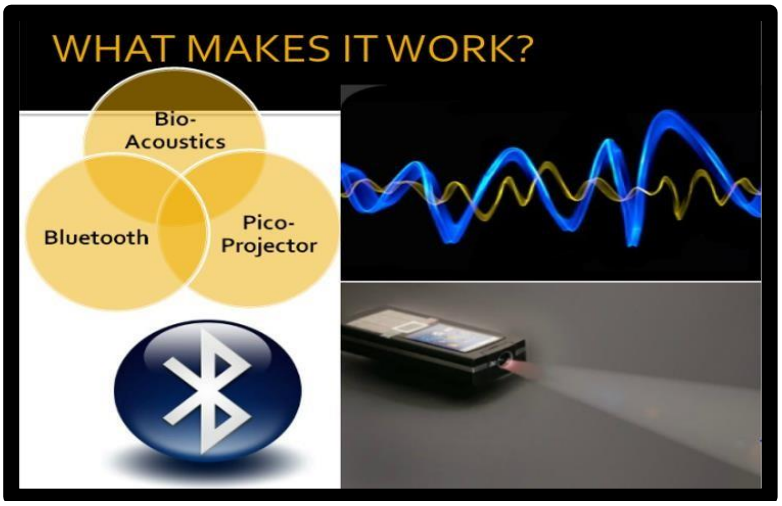


Figure 3.1 : Connection Diagram

In SkinKonnect, a keyboard, menu, or other graphics are beamed onto a user's palm and forearm from a Pico projector embedded in an armband. An acoustic detector in the armband then determines which part of the display is activated by the user's touch. As the researchers explain, variations in bone density, size, and mass, as well as filtering effects from soft tissues and joints, mean different skin locations are acoustically distinct. Their software matches sound frequencies to specific skin locations, allowing the system to determine which “skin button” the user pressed.

3.3 Requirement Specifications

The project consists of three main components.

i. Raspberry pi
ii. Pico Projector
iii. Sensors

The sensors used in the project are minisense 100, and it is considered to be the ears of the project. This sensor is very fragile and can get effected by the external environment very easily. So, to improve the efficiency of the project several sensors are used and together they are called as array of sensors.

The raspberry pi module is considered to be the brain of the project because it is capable to make decision on the basis of the input from the sensors.

The pico projector is considered to be the eyes of the project and is used to project the GUI design of the remote control of the hand of the user.

3.4 Use Cases

The project idea has the capability to work on any area of your body but due to the time and other resources limitations we have designed it to work on your forearm only.

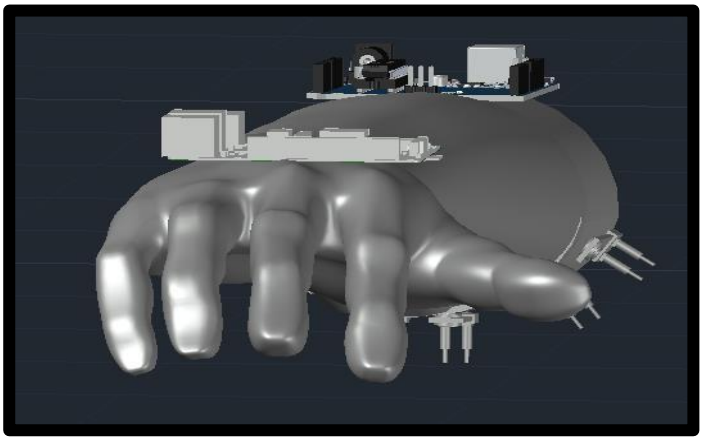
The process of taking input from the skin includes tapping frequency on which the microcontroller can differentiate that exact point of the tap because our body produce different tapping frequency on different places. So, to make it work on other parts of the body as well a very well-done research was required based truly on experiments that cannot be done due to limited time and limited resources.

Chapter # 4

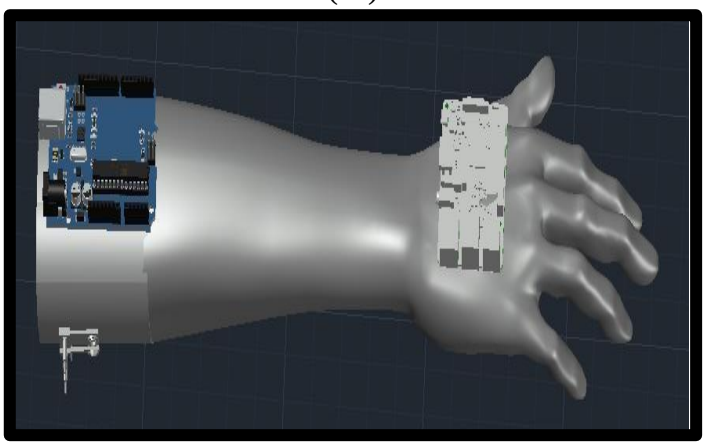
System Design

4.1 System Architecture

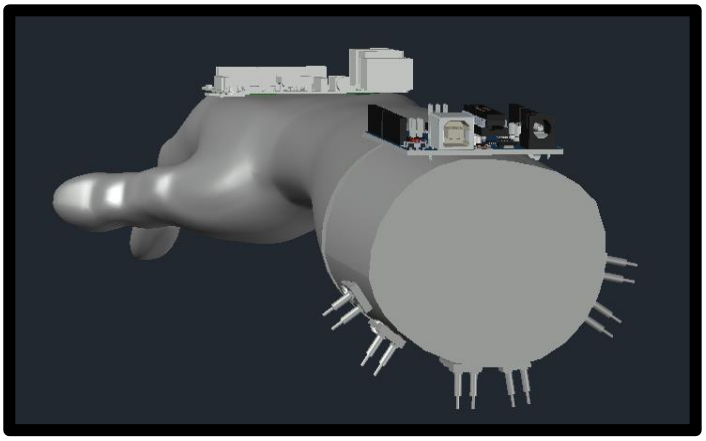
Keeping the image of the final design of the project Figure 4.1 illustrated the AutoCAD design of the project.



(a)



(b)



(c)

Figure 4.1: (a) Front View, (b) Upper View, (c) Back View

The system is mainly composed of sensors, arduino nano and a relay-based circuit. Minisense 100 Sensors are used in an array that is being used to capture acoustic vibrations from the skin. Minisense 100 has Up to 40 Hz (2,400 rpm) Operation Below Resonance that is quite useful to collect low frequency vibrations produced by the tap on the skin. Arduino Nano is used to integrate the sensors output to the remaining system. Arduino nano is further integrated with Raspberry pi module. Lm2596 buck converter is used to provide the required voltages to the subsystems being used in the project. Raspberry Pi3 b+ is the micro controller of the project that contains the GUI of the remote controller. The output from the sensors array is collected by Arduino nano and then transferred to Raspberry pi module. In raspberry pi module the output is looked in to the look up table based on that; results are being performed. 32 GB Memory Card Sand Disk is used for storing data, a 32 GB memory card of sand disk is being used. Pico Projector is also used. The main purpose of using the Pico projector is to make the product portable. The Pico projector is used to display the GUI of the remote controller on the arm of the user. Bluetooth Module is used to link the raspberry pi module to the Pico projector.

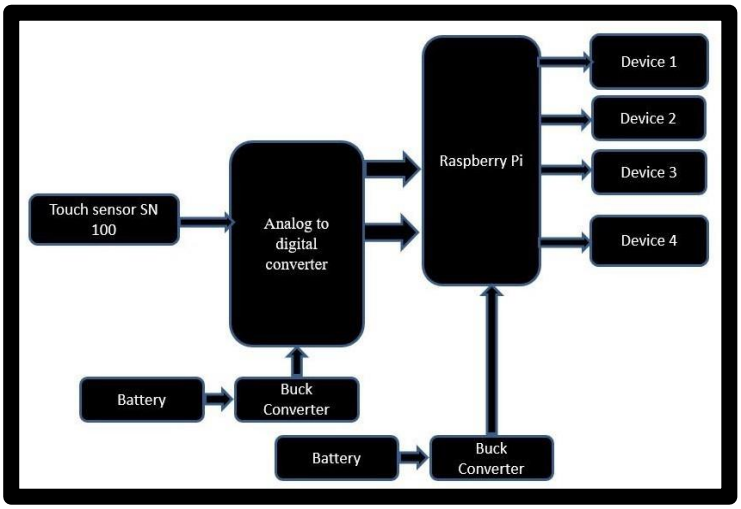


Fig 4.2: Block Diagram

4.2 Design Constraints

This section describes any constraints in the system design (reference any trade-off analyses conducted such, as resource use versus productivity, or conflicts with other systems) and includes any assumptions made during the developing the system design.

4.3 Design Methodology

The brain of the product is considered to be Raspberry pi module because it holds the decision ability of the whole design. The Raspberry pi module is capable of generating the commands on the basis of the output from the sensors array and it is then wirelessly integrated with relays that turns the load ON and OFF.

The output from the sensor is amplified and integrated with raspberry pi through Arduino nano. The GUI of the remote controller is designed in Raspberry pi module and displayed on the arm of the user through Pico projector.

4.4 High Level Design

The Design is typically divided into three parts:

i. Sensors

The sensors are used to capture the bio-acoustics vibrations from the skin. Sensor arrays are used to determine the finger tap positions on the skin. According to the features of human body, the finger tapping on the muscles, bones and tendons creates different forms of vibrations. A bone conduction microphone is a suitable choice for SkinKonnect. This sensor is perfect for sensing the longitudinal waves and the transverse waves that generated due to the finger tapping. This bio acoustic and sensors captures and converts the acoustic energy (in the form of transverse and longitudinal waves) formed by the finger taps on the

skin into a digital signal information. The accuracy of the project depends on the number of sensors used in the array. The greater the number of sensors the greater will be the accuracy.

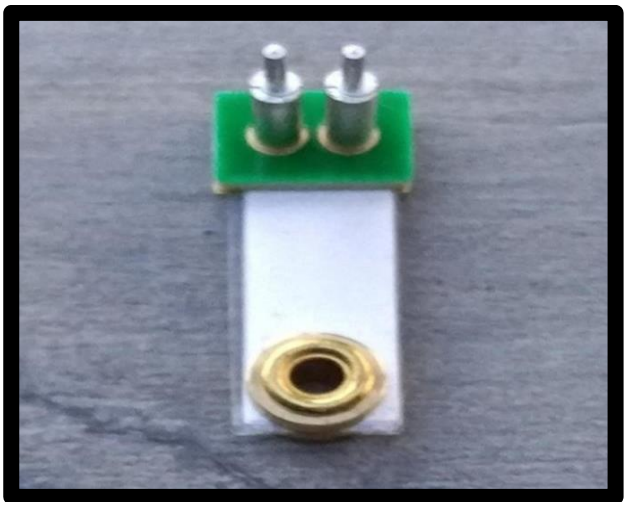


Figure 4.3 : Minisense 100 Sensor

ii. Pi B+

The raspberry pi module is used to decide which task has to be performed on the basis of sensors output. Then the output is decided by finding the output from the sensors in the look up table. Also, the GUI of the remote controller is designed in this module.

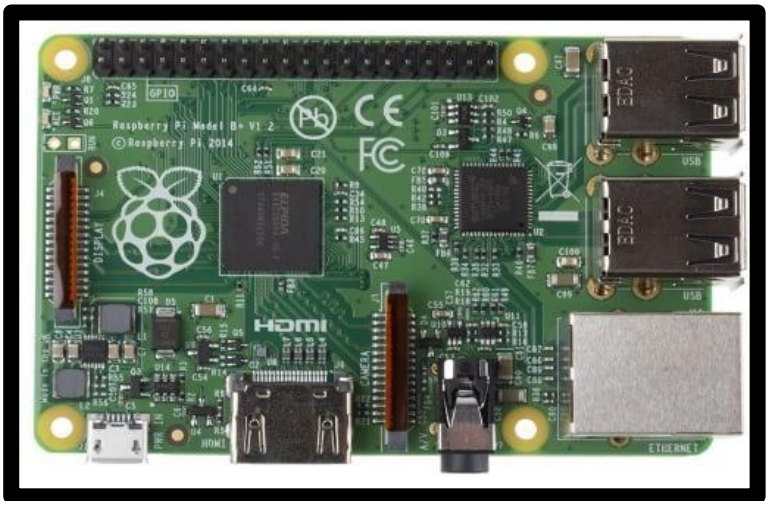


Figure 4.4: Raspberry pi B+

iii. Relay-Based Circuit

The relay-based circuit is capable of turning the outputs ON and OFF on the basis of the command generated by the raspberry pi module.

4.5 Low Level Design

The fig:4.5 shows the basic simulations of project design. The output from the raspberry pi is fed into the relays through transistors and then the relays are turning the devices ON and OFF accordingly. The whole process of conveying the signal from raspberry pi to relays is done wirelessly in the project but not in the simulations.

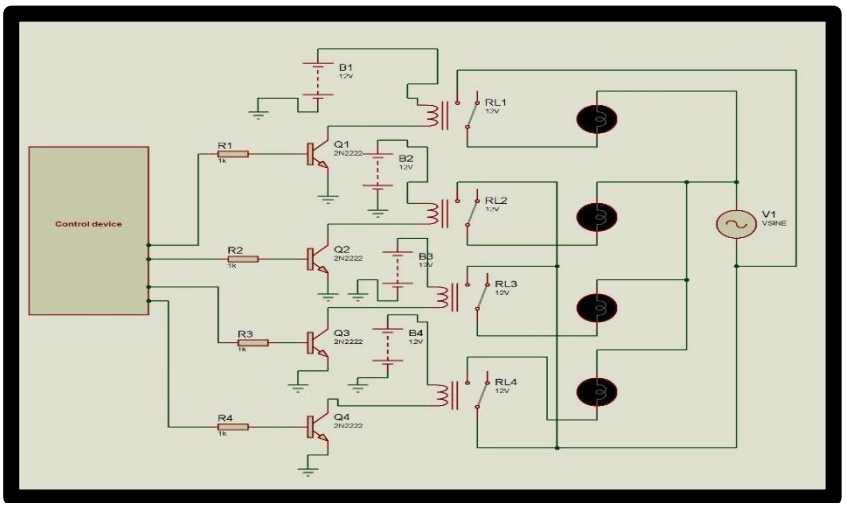


Fig:4.5: Simulation on Proteus

Chapter # 5

System Implementation

5.1 System Architecture

The system is typically composed of an array of sensors, the sensors used are: Minisense 100. The output of the sensors is analog with is converted to digital by using an analog to digital converter. The digital output is amplified and fed to Arduino nano. Arduino nano is further integrated with raspberry pi 3 b+ module that is a microcontroller. In microcontroller there are mainly two functions being performed. Firstly, there is a look up table that generated the commands of which function has to be performed based on the output from the sensors array. Secondly, a GUI of a remote controller is designed in microcontroller. The output from the microcontroller is then integrated with relay-based circuit wirelessly that drive the devices being operated.

5.2 Tools and Technology Used

i. Minisense 100

The MiniSense 100 acts as a cantilever-beam accelerometer. When the beam is mounted horizontally, acceleration in the vertical plane creates bending in the beam, due to the inertia of the mass at the tip of the beam. Strain in the beam creates a piezoelectric response, which may be detected as a charge or voltage output across the electrodes of the sensor.

ii. Arduino Nano

The Arduino Nano is a small, complete, and breadboard-friendly board based on the ATmega328 (Arduino Nano 3. x). It has more or less the same functionality of the Arduino Duemilanove, but in a different package. It lacks only a DC power jack and works with a Mini-B USB cable instead of a standard one.

iii. Raspberry Pi3 b+

The Raspberry Pi 3 Model B+ is the latest product in the Raspberry Pi 3 range, boasting a 64- bit quad core processor running at 1.4GHz, dual-band 2.4GHz. and 5GHz wireless LAN, Bluetooth 4.2/BLE, faster Ethernet, and PoE capability. via a separate PoE HAT.

iv. Buck Converter

A buck converter (step-down converter) is a DC-to-DC power converter which steps down voltage (while stepping up current) from its input (supply) to its output (load).

v. Pico Projector

Pico projector is a small, yet powerful mini projector. It's an extremely portable projector which, at its smallest, is pocket-sized and comparable in weight and shape to a cell phone.

vi. 32 GB Memory Card

A memory card or memory cartridge is an electronic data storage device used for storing digital information, typically using flash memory.

vii. Internet of things

The Internet of things describes the network of physical objects— “things”—that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the Internet.

5.3 Development Environment/Languages Used

i. Embedded C

Embedded C is a generic term given to a programming language written in C, which is associated with a particular hardware architecture. Embedded C is an extension to the C language with some additional header files. It is a set of language extensions for the C programming language by the C Standards Committee to address commonality issues that exist between C extensions for different embedded systems.

ii. Python

Python is an interpreted, high-level and general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. It is dynamically typed, and garbage collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

5.4 Processing Logic/Algorithms:

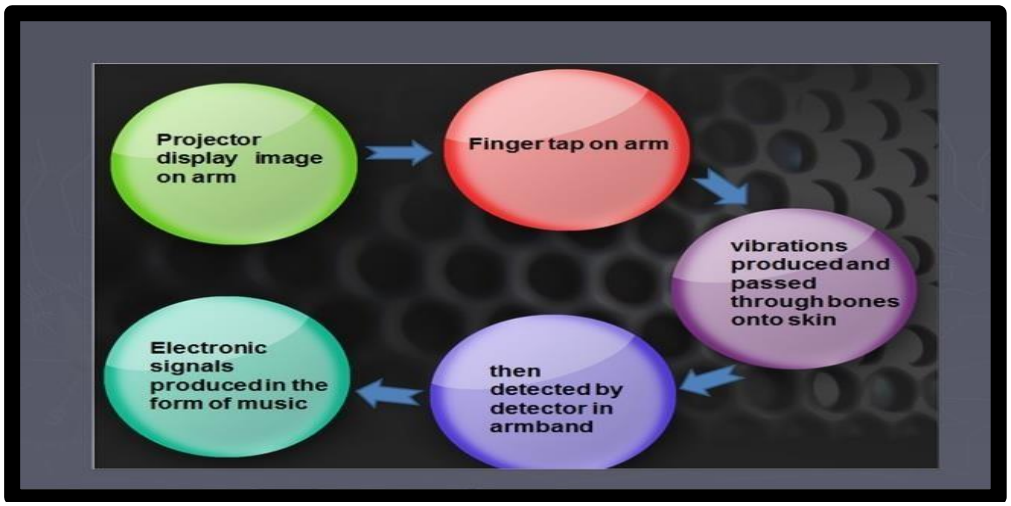


Figure 5.1: Algorithm

Figure 5.1 shows the algorithm for taking the input from the skin. Firstly, the array of the vibrational sensors captures the vibrations.

The output from the sensors is linked with Arduino nano which is converting the analog signal into digital. Then the digital signal is fed into raspberry pi module that is considered to be the brain of our project. The Raspberry pi module is generating the command to be performed on the basis of the output from the sensors. The generated command is then sent to the relay-based circuit that is turning ON the required device. The GUI of the remote control is designed in Raspberry pi module and is projected on the arm of the user by using pico projector.

Chapter # 6

System Testing and Evaluation

6.1 Graphical User Interface Testing

The input from the user is taken through skin by projecting the GUI of the remote control on the forearm of the user. The GUI is actually the depiction of a remote control that is capable to control 4 devices. The GUI is designed in the micro controller. The Raspberry pi is used as a micro controller so, the GUI is designed is Raspberry pi. Figure 6.1 shows the GUI design that can 4 options.

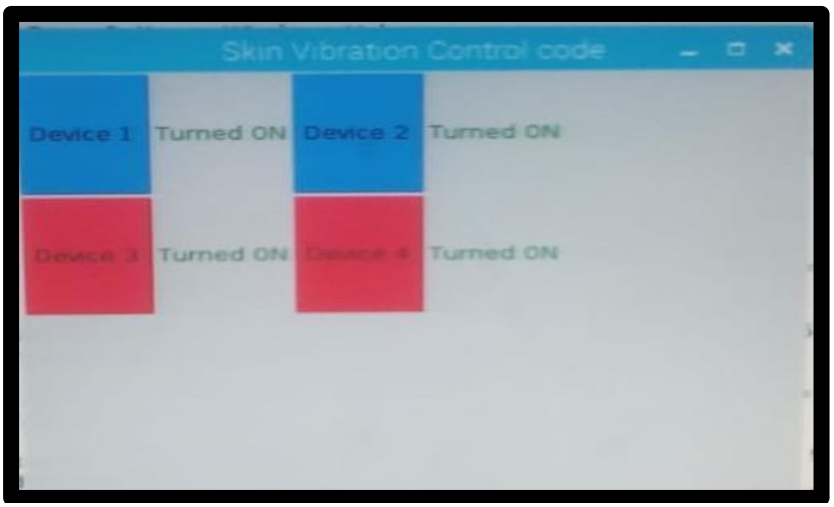


Figure 6.1: Graphical User Interface Design

6.2 Usability Testing

The actual product will be so embedded hence small in size, but right the input side of the project is big in size. The input side contains 4 sensors that are taking the input from the skin. The user while using the project must hold it across the forearm. The system is just a prototype right now so, the components are not so very secured. The project contains sensors that are versy fragile, so the input side of the system requires a lot of care.

i. Software Performance Testing

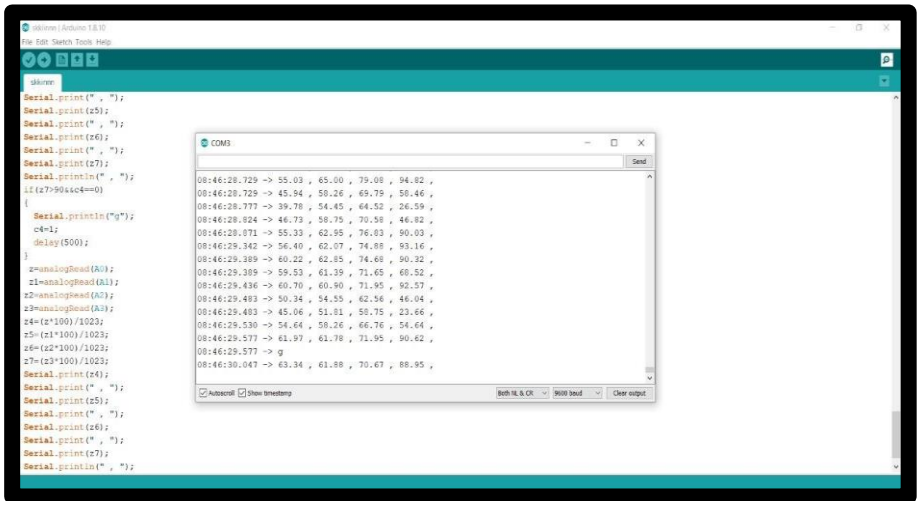


Figure 6.2: Sensors Output Serial

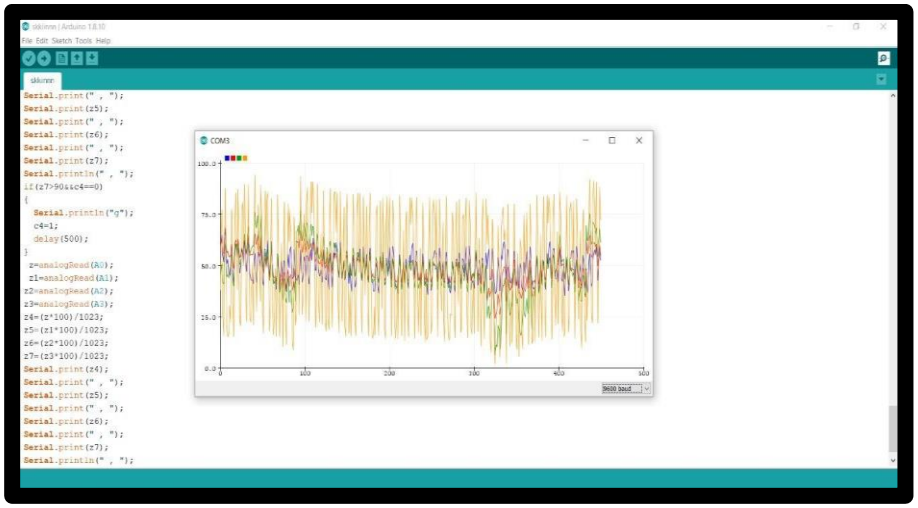


Figure 6.3: Sensor Output

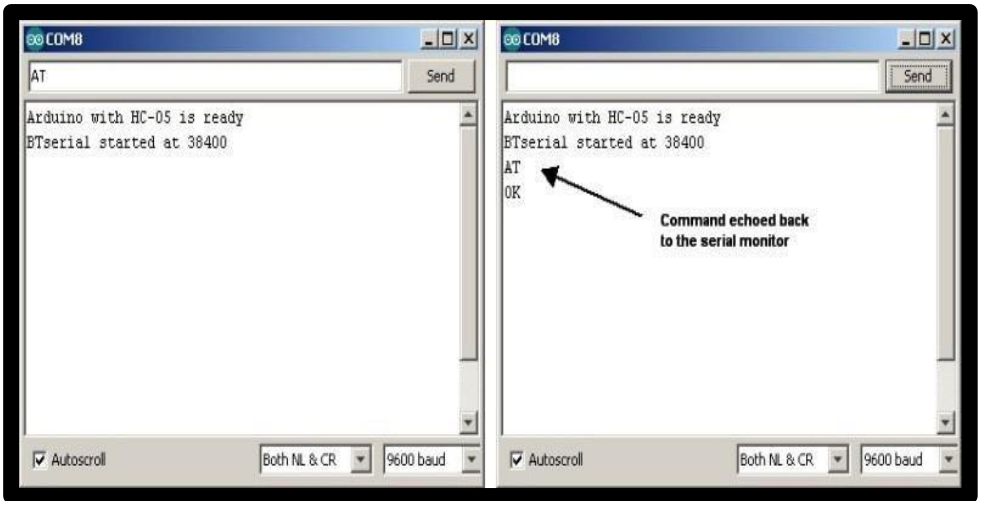


Figure 6.4: Bluetooth Communication

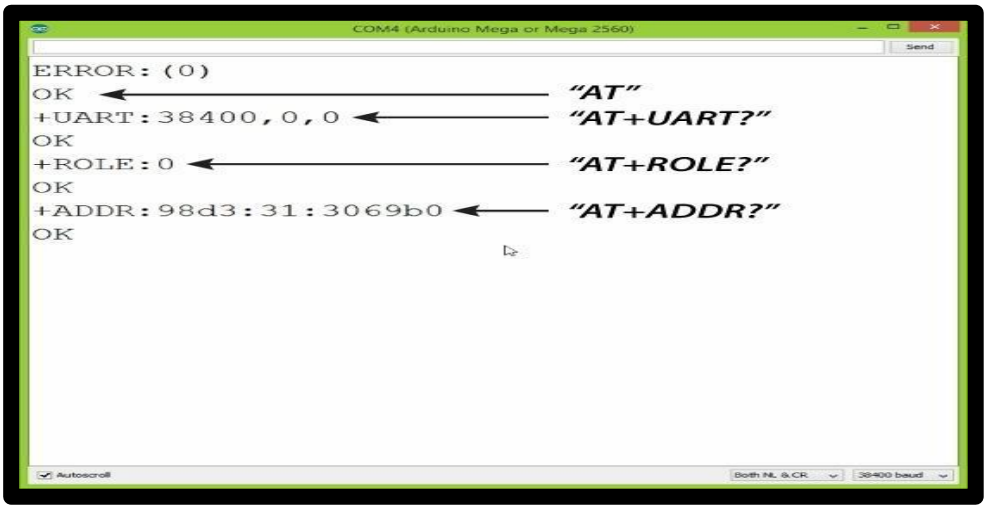
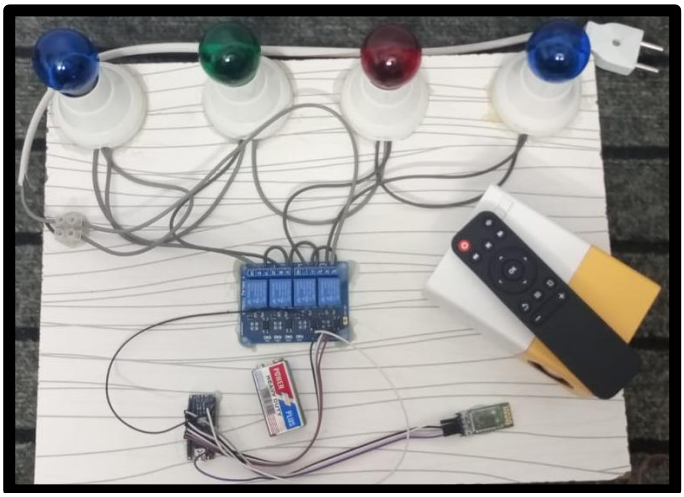


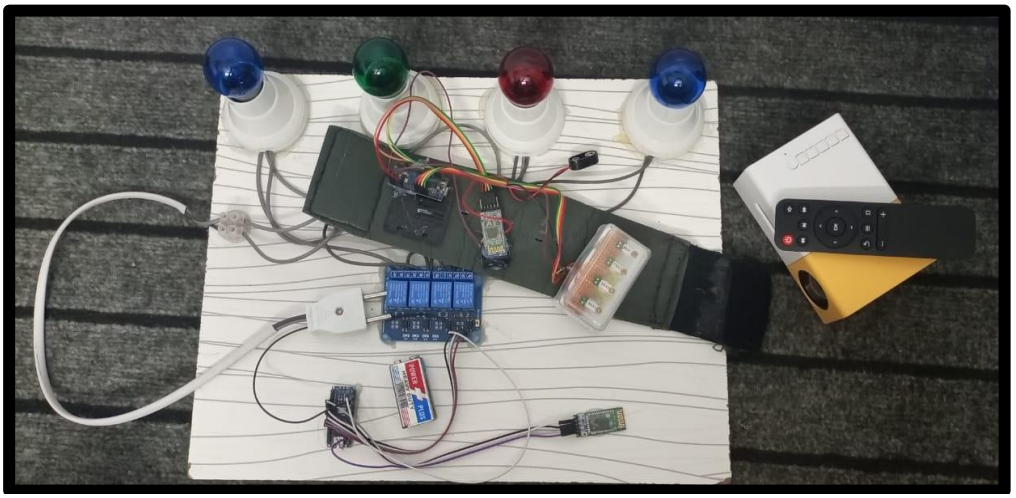
Figure 6.5: Communicatio

6.3 Load Testing

The output side of the project is shown in below figures



(a)



(b)

Figure 6.6 : (a) Output module, (b) Input module

Figure 6.6 (a) shows 4 loads that can be operated through our project. The output is controlled by a relay-based circuit that is responsible to power ON one device at a time. Figure 6.6 (b) shows complete hardware input and output side with sensors attached to the arm band .

Chapter # 7

Conclusion

The first and foremost task was to generate the output from the tap on the skin. For that purpose, minisense 100 sensors were used. By linking the output from the sensors to Arduino nano the output from the skin tap was generated. The output was then processed and fed into raspberry pi module that generates the command based on that output. Since due to very limited resources the project is working on the forearm only but in future it has many applications. The GUI designed in raspberry pi module is projected on the arm of the user with the help of pico projector. After the selection of the used, the generated command was sent to the relay-based circuit that is driving the load. All the 4 devices as proposed were operation properly.

It is designed to detect and localize finger taps on the forearm and hand. Our project performs very well for a series of gestures, even when the body is in motion. Additionally, it has presented initial results demonstrating other potential uses of our approach, which we hope to further explore in future work. In this report we also discussed the major components through which the SkinKonnect Technology works that are: Bio-acoustic sensing, Pico-projector, Bluetooth. This technology has a great future scope as it uses our body as the input devices.

We present SkinKonnect, a technology that appropriates the human body for acoustic transmission, allowing the skin to be used as an input surface. In particular, we resolve the location of finger taps on the arm and hand by analyzing mechanical vibrations that propagate through the body. We collect these signals using a novel array of sensors.

This approach provides an always available, naturally portable, and on-body finger input system. We assess the capabilities, accuracy, and limitations of our technique through a two-part, twenty-participant user study. To further illustrate the utility of our approach, we conclude with several proof-of-concept applications we developed.

References

[1] R. Lawanya, "BIO-ACOUSTIC SENSING OF SKINPUT," *International Journal of Scientific & Engineering Research,* vol. 6, no. 7, 2015.

[2] Riya Gupta, "Skinput Technology- A Technique to act Skin as," *International Journal of Engineering Research & Technology,* no. special, 2017.

[3] M. Chris Harrison, "Skinput: Appropriating the Body as an Input Surface," *Computing on the Body,* 2010.

[4] Ravindra K. Patil, "Design of Touch-screen by Human Skin for Appliances," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering,* vol. 5, no. 9, 2016.

[5] Abhijeet S. Kapse, "Arm as a Touch-Screen," *ijera,* vol. 4, no. 2, 2014.

[6] Martin Weigel, "iSkin: Flexible, Stretchable and Visually Customizable".

[7] S. D.Srinath, "Human Body as a Touch Screen," *Panimalar institute of technology.*

[8] Richa, "PROPOSED NEW TECHNIQUE FOR EVALUATION OF SKIN AS TOUCH SCREEN IN COMPUTER ENGINEERING," *International Journal of Advanced Science, Engineering and Technology.* , vol. 3, no. 2, 2014.

[9] S. Mandlik, "The Skinput Technology - A Technology using Skin as an Input," *International Journal for Scientific Research & Development*|, vol. 7, no. 3, 2019.

[10] M. Monisha, "Skinput the Human Body as Touch Screen," *International Journal of Engineering Research & Technology,* no. Special, 2019.

[11] C. HARRISON, "*The Human Body as an Interactive Computing Platform,"* 2013.

Appendices

Appendix A
User manual

Skinkonnect uses serial communication to communicate with controller and then the signal is sent wirelessly to the output module controller and relay circuit which drives load.

Step 1: Wearing of Wrist Band

Wear the wrist band on your arm sensors should be below the band touching the skin.

Step 2: Turn on the Modules

Turn the input and output modules on by giving them power supply wait for the Connections between transmitter and receiver side.

Step 3: Check for the Blinking Lights

Check for the blinking lights in following components.

i. Hc-05 module transmitter
ii. Hc-05 module receiver

Appendix B
Code

C.1 For Low Pass Filter (Arduino Tx Side)

```
#include <SoftwareSerial.h>

SoftwareSerial BTSerial(10, 11); // RX | TX

const float alpha = 0.3;
double data_filtered[] = {0, 0};
double data1_filtered[] = {0, 0};
double data2_filtered[] = {0, 0};
double data3_filtered[] = {0, 0};
const int n = 1;
const int analog_pin = 3;
double data = 0;
double data1 = 0;
double data2= 0;
double data3= 0;
double z1=500;
double z=0;
double z4=0;
double z5=0;
double z6=0;
double z7=0;
void setup(){
   Serial.begin(9600);
   BTSerial.begin(38400);
}

void loop(){
   // Retrieve Data
    data = analogRead(A0);
   data_filtered[n] = alpha * data + (1 - alpha) * data_filtered[n-1];
   data_filtered[n-1] = data_filtered[n];
   data1 = analogRead(A1);
   data1_filtered[n] = alpha * data1 + (1 - alpha) * data1_filtered[n-1];
   data1_filtered[n-1] = data1_filtered[n];
```

```
    data2 = analogRead(A2);
    data2_filtered[n] = alpha * data2 + (1 - alpha) * data2_filtered[n-1];
    data2_filtered[n-1] = data2_filtered[n];
    data3 = analogRead(A3);
    data3_filtered[n] = alpha * data3 + (1 - alpha) * data3_filtered[n-1];
    data3_filtered[n-1] = data3_filtered[n];

    z=(data_filtered[n]+data1_filtered[n]+data2_filtered[n]+data2_filtered[n])/4;
   // Serial.println(z);

data = analogRead(A0);
    data_filtered[n] = alpha * data + (1 - alpha) * data_filtered[n-1];
    data_filtered[n-1] = data_filtered[n];
    data1 = analogRead(A1);
    data1_filtered[n] = alpha * data1 + (1 - alpha) * data1_filtered[n-1];
    data1_filtered[n-1] = data1_filtered[n];

    data2 = analogRead(A2);
    data2_filtered[n] = alpha * data2 + (1 - alpha) * data2_filtered[n-1];
    data2_filtered[n-1] = data2_filtered[n];
    data3 = analogRead(A3);
    data3_filtered[n] = alpha * data3 + (1 - alpha) * data3_filtered[n-1];
    data3_filtered[n-1] = data3_filtered[n];

    z=(data_filtered[n]+data1_filtered[n]+data2_filtered[n]+data2_filtered[n])/4;
   // Serial.println(z);
   //if(z>50)
   //{
 Serial.println(z);
 //  }
 //delay(1000);
      delay(5);
      z1=z;
 if(z>450)
  {
     // Serial.println("a");
   // z4=1;
    //Serial.print
    while(z1>190)
    {
    //  Serial.println("a");
      BTSerial.println("a");
    z4=1;
     data = analogRead(A0);
    data_filtered[n] = alpha * data + (1 - alpha) * data_filtered[n-1];
    data_filtered[n-1] = data_filtered[n];
    data1 = analogRead(A1);

    data1_filtered[n] = alpha * data1 + (1 - alpha) * data1_filtered[n-1];
    data1_filtered[n-1] = data1_filtered[n];
```

```
 data2 = analogRead(A2);
 data2_filtered[n] = alpha * data2 + (1 - alpha) * data2_filtered[n-1];
 data2_filtered[n-1] = data2_filtered[n];
 data3 = analogRead(A3);
 data3_filtered[n] = alpha * data3 + (1 - alpha) * data3_filtered[n-1];
 data3_filtered[n-1] = data3_filtered[n];

 z1=(data_filtered[n]+data1_filtered[n]+data2_filtered[n]+data2_filtered[n])/4;
// Serial.print(",");
 Serial.println(z1);
 }
}
if(z>450&&z4==1)
{
// Serial.println("b");
// z4=0;
 while(z1>190)
 {
  //  Serial.println("b");
    BTSerial.println("b");
 z4=0;
/*  data = analogRead(analog_pin);
 data_filtered[n] = alpha * data + (1 - alpha) * data_filtered[n-1];
 data_filtered[n-1] = data_filtered[n];*/

 data = analogRead(A0);
 data_filtered[n] = alpha * data + (1 - alpha) * data_filtered[n-1];
 data_filtered[n-1] = data_filtered[n];
 data1 = analogRead(A1);
 data1_filtered[n] = alpha * data1 + (1 - alpha) * data1_filtered[n-1];
 data1_filtered[n-1] = data1_filtered[n];

 data2 = analogRead(A2);
 data2_filtered[n] = alpha * data2 + (1 - alpha) * data2_filtered[n-1];
 data2_filtered[n-1] = data2_filtered[n];
 data3 = analogRead(A3);
 data3_filtered[n] = alpha * data3 + (1 - alpha) * data3_filtered[n-1];
 data3_filtered[n-1] = data3_filtered[n];
```

```
   z1=(data_filtered[n]+data1_filtered[n]+data2_filtered[n]+data2_filtered[n])/4;
  // Serial.print("K=");
  // Serial.println(z1);
   }
 }
  /**********************************Device 2
*******************************/
  if(z>360&&z<450&&z5==0)
 {
  // Serial.println("b");
  // z4=0;
    z5=1;




 while(z1>190)
   {
     //Serial.println("c");
     BTSerial.println("c");

   data = analogRead(A0);
   data_filtered[n] = alpha * data + (1 - alpha) * data_filtered[n-1];
   data_filtered[n-1] = data_filtered[n];
   data1 = analogRead(A1);
   data1_filtered[n] = alpha * data1 + (1 - alpha) * data1_filtered[n-1];
   data1_filtered[n-1] = data1_filtered[n];

   data2 = analogRead(A2);
   data2_filtered[n] = alpha * data2 + (1 - alpha) * data2_filtered[n-1];
   data2_filtered[n-1] = data2_filtered[n];
   data3 = analogRead(A3);
   data3_filtered[n] = alpha * data3 + (1 - alpha) * data3_filtered[n-1];
   data3_filtered[n-1] = data3_filtered[n];

   z1=(data_filtered[n]+data1_filtered[n]+data2_filtered[n]+data2_filtered[n])/4;
  // Serial.print("K=");
  // Serial.println(z1);
   }
 }

  if(z>360&&z<450&&z5==1)
 {

   while(z1>190)
   {
```

```
   // Serial.println("d");
      BTSerial.println("d");
    z5=0;
   data = analogRead(A0);
    data_filtered[n] = alpha * data + (1 - alpha) * data_filtered[n-1];
    data_filtered[n-1] = data_filtered[n];
    data1 = analogRead(A1);
    data1_filtered[n] = alpha * data1 + (1 - alpha) * data1_filtered[n-1];
    data1_filtered[n-1] = data1_filtered[n];

    data2 = analogRead(A2);
    data2_filtered[n] = alpha * data2 + (1 - alpha) * data2_filtered[n-1];
    data2_filtered[n-1] = data2_filtered[n];
    data3 = analogRead(A3);
    data3_filtered[n] = alpha * data3 + (1 - alpha) * data3_filtered[n-1];
    data3_filtered[n-1] = data3_filtered[n];

    z1=(data_filtered[n]+data1_filtered[n]+data2_filtered[n]+data2_filtered[n])/4;
    //Serial.print("K=");
   // Serial.println(z1);
     }
  }



  /************************************** Device 3
************************************/
    if(z>280&&z<360&&z6==0)
   {
//BTSerial.println("e");
   //  z6=1;
     while(z1>190)
     {
       // Serial.println("e");
        BTSerial.println("e");
     z6=1;
     data = analogRead(A0);
     data_filtered[n] = alpha * data + (1 - alpha) * data_filtered[n-1];
     data_filtered[n-1] = data_filtered[n];
     data1 = analogRead(A1);
     data1_filtered[n] = alpha * data1 + (1 - alpha) * data1_filtered[n-1];
     data1_filtered[n-1] = data1_filtered[n];

     data2 = analogRead(A2);
     data2_filtered[n] = alpha * data2 + (1 - alpha) * data2_filtered[n-1];
     data2_filtered[n-1] = data2_filtered[n];
```

```
  data3 = analogRead(A3);
  data3_filtered[n] = alpha * data3 + (1 - alpha) * data3_filtered[n-1];
  data3_filtered[n-1] = data3_filtered[n];

  z1=(data_filtered[n]+data1_filtered[n]+data2_filtered[n]+data2_filtered[n])/4;
//  Serial.print("K=");
 // Serial.println(z1);
  }
}

 if(z>280&&z<360&&z6==1)
{
 // BTSerial.println("f");
 // z6=0;

  while(z1>190)
  {
    // Serial.println("f");
     BTSerial.println("f");
  z6=0;
  data = analogRead(A0);
  data_filtered[n] = alpha * data + (1 - alpha) * data_filtered[n-1];
  data_filtered[n-1] = data_filtered[n];
  data1 = analogRead(A1);
  data1_filtered[n] = alpha * data1 + (1 - alpha) * data1_filtered[n-1];
  data1_filtered[n-1] = data1_filtered[n];

  data2 = analogRead(A2);
  data2_filtered[n] = alpha * data2 + (1 - alpha) * data2_filtered[n-1];
  data2_filtered[n-1] = data2_filtered[n];
  data3 = analogRead(A3);
  data3_filtered[n] = alpha * data3 + (1 - alpha) * data3_filtered[n-1];
  data3_filtered[n-1] = data3_filtered[n];

  z1=(data_filtered[n]+data1_filtered[n]+data2_filtered[n]+data2_filtered[n])/4;
//  Serial.print("K=");
//  Serial.println(z1);
  }
 }
 /************************************** Device 4
**********************************/
  if(z>210&&z<280&&z7==0)
 {
```

```
 while(z1>190)
  {
    // Serial.println("g");
     BTSerial.println("g");
  z7=1;
   data = analogRead(A0);
  data_filtered[n] = alpha * data + (1 - alpha) * data_filtered[n-1];
  data_filtered[n-1] = data_filtered[n];
  data1 = analogRead(A1);
  data1_filtered[n] = alpha * data1 + (1 - alpha) * data1_filtered[n-1];
  data1_filtered[n-1] = data1_filtered[n];

  data2 = analogRead(A2);
  data2_filtered[n] = alpha * data2 + (1 - alpha) * data2_filtered[n-1];
  data2_filtered[n-1] = data2_filtered[n];
  data3 = analogRead(A3);
  data3_filtered[n] = alpha * data3 + (1 - alpha) * data3_filtered[n-1];
  data3_filtered[n-1] = data3_filtered[n];

  z1=(data_filtered[n]+data1_filtered[n]+data2_filtered[n]+data2_filtered[n])/4;
 // Serial.print("K=");
//  Serial.println(z1);
   }
}

  if(z>210&&z<280&&z7==1)
{

  while(z1>190)
  {
    // Serial.println("h");
     BTSerial.println("h");
  z7=0;


data = analogRead(A0);
  data_filtered[n] = alpha * data + (1 - alpha) * data_filtered[n-1];
  data_filtered[n-1] = data_filtered[n];
  data1 = analogRead(A1);
  data1_filtered[n] = alpha * data1 + (1 - alpha) * data1_filtered[n-1];
  data1_filtered[n-1] = data1_filtered[n];
```

```
  data2 = analogRead(A2);
  data2_filtered[n] = alpha * data2 + (1 - alpha) * data2_filtered[n-1];
  data2_filtered[n-1] = data2_filtered[n];
  data3 = analogRead(A3);
  data3_filtered[n] = alpha * data3 + (1 - alpha) * data3_filtered[n-1];
  data3_filtered[n-1] = data3_filtered[n];

  z1=(data_filtered[n]+data1_filtered[n]+data2_filtered[n]+data2_filtered[n])/4;
 // Serial.print("K=");
//  Serial.println(z1);
  }
}
```

C.2 For Skin Receiver (ARDUINO)

```
#include <SoftwareSerial.h>

SoftwareSerial BTSerial(2, 3); // RX | TX
char z=0;
void setup()
{
  pinMode(4, OUTPUT);  // this pin will pull the HC-05 pin 34 (key pin) HIGH to
switch module to AT mode
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);

  digitalWrite(4, HIGH);
  digitalWrite(5, HIGH);
  digitalWrite(6, HIGH);
  digitalWrite(7, HIGH);
  Serial.begin(9600);
 // Serial.println("Enter AT commands:");
  BTSerial.begin(38400);  // HC-05 default speed in AT command more
  delay(500);
  digitalWrite(4, HIGH);
  digitalWrite(5, HIGH);
  digitalWrite(6, HIGH);
  digitalWrite(7, HIGH);
}

void loop()
{
  if (BTSerial.available()>0)
  {
    z=BTSerial.read();
   // Serial.println(z);
    delay(2);

if(z=='a')
{
  digitalWrite(4,LOW);
  delay(1);
}
```

```
if(z=='b')
{
  digitalWrite(4,HIGH);
  delay(1);
}
if(z=='c')
{
  digitalWrite(5,LOW);
  delay(1);
}
if(z=='d')
{
  digitalWrite(5,HIGH);
  delay(1);
}
if(z=='e')
{
  digitalWrite(6,LOW);
  delay(1);
}
if(z=='f')
{
  digitalWrite(6,HIGH);
  delay(1);
}
if(z=='g')
{
  digitalWrite(7,LOW);
  delay(1);
}
if(z=='h')
{
  digitalWrite(7,HIGH);
  delay(1);
}
}
}
```