

Smart Glove: Gesture Translator

By

Angela Mukulu Mwanzia

19ZAD103899

Supervisor: Mr. Matthew Thiong’o

A Project Proposal Submitted in Partial Fulfilment of the Requirements for the Degree/Diploma of Computer Science, School of Computing Sciences Riara University

Nairobi, Kenya

Date Submitted: 24th /03th/2021

# **Declaration**

I declare that this or any other University has not previously submitted this work for the awarding of the course marks. To the best of my knowledge and belief, this wok contains no material previously published or written by another person except where due reference is made.

Student Name:

.................................................

Signature:

.................................................

Date:

.................................................

.................................................

**APPROVAL**

The project proposal of student name was reviewed and approved by the following:

Supervisor Name:

Signature: .....................

# **Dedication**

I dedicate this project to my parents due to allowing me to pursue this diploma by contributing to my education. In addition, I dedicate this work to my Lecturers: Mr. Matthew and Mr. Paul, who’ve helped me with this proposal. I also dedicate this project proposal to God who provides me with all the energy, willpower and good health to go about my day.

# **Acknowledgement**

I cannot express enough thanks to my Professors: Mr. Matthew and Mr. Paul for their help in finishing this proposal. I offer my sincere appreciation to my parents who have enabled me to take this course.

# **ABSTRACT**

There are people who live their own lives with no ability to speak, sign language can demonstrate their points of view. Sign language is based on hand gestures that have a certain movement to symbolize the “language” they’re using to communicate. In sign language, a gesture is a particular motion of the hands that forms a specific shape from the fingers and whole palm of the hand. Sign language, on the other hand, can lose its meaning in the nonverbal world. This concept is for a Smart Glove: Gesture Interpreter for the nonverbal who interact with others by sign language.

The Smart Glove: Gesture Translator will be able to track hand signals using sensors and then translate sign language into text and speech. The object of this proposal is to translate sign language into text and speech in order to make communication effective between the non-verbal and verbal.

The Smart Glove will be worn by the non-verbal, so when they perform their hand gestures the signs are detected and deciphered thus converted to text then speech in order to enable the verbal to understand the non-verbal easily. The Smart Glove is connected to a smart phone via Bluetooth Module. The smart phone will have an Android Application that will convert the data received from the glove to text based on the data received and then convert the text to speech. The Smart Glove will have the English alphabets so as to give a representation of the possibility of representing words.

Table of Contents

[**Declaration** I](#_Toc68025174)

[**Dedication** II](#_Toc68025175)

[**Acknowledgement** III](#_Toc68025176)

[**ABSTRACT** IV](#_Toc68025177)

[**List of Figures** VII](#_Toc68025178)

[**List of Tables** VIII](#_Toc68025179)

[**CHAPTER 1: INTRODUCTION** 9](#_Toc68025180)

[**1.1: Background** 9](#_Toc68025181)

[**1.2: Problem Statement** 10](#_Toc68025182)

[**1.3: Objectives** 11](#_Toc68025183)

[**1.4: Justification** 11](#_Toc68025184)

[**1.5: Scope** 12](#_Toc68025185)

[**CHAPTER 2: LITERATURE REVIEW** 13](#_Toc68025186)

[**2.1: Introduction** 13](#_Toc68025187)

[**2.2: The background of Sign Language** 13](#_Toc68025188)

[**2.3: Gesture Recognition Methods** 13](#_Toc68025189)

[**2.3.1: Data Glove Approach** 13](#_Toc68025190)

[**2.3.2: Visual-Based Approach** 14](#_Toc68025191)

[**2.3.3: Virtual-Based Button Approach** 15](#_Toc68025192)

[**2.3.4: Selection of the Approach** 16](#_Toc68025193)

[**2.4: Communication between verbal and non-verbal people** 16](#_Toc68025194)

[**2.5: Concepts of Gestures** 17](#_Toc68025195)

[**2.5.1: CMOS Camera Use** 17](#_Toc68025196)

[**2.5.2: Glove Based Leaf Switches** 17](#_Toc68025197)

[**2.5.3: Copper Glove Based Plate** 18](#_Toc68025198)

[**2.5.4: Glove Based on Flex Sensor** 18](#_Toc68025199)

[**2.5.5: Selection of the type of material to be use on the data glove** 18](#_Toc68025200)

[**2.6: Conclusion** 18](#_Toc68025201)

[**CHAPTER 3: METHODOLOGY** 19](#_Toc68025202)

[**3.1: Introduction** 19](#_Toc68025203)

[**3.2: System Analysis** 19](#_Toc68025204)

[**3.2.1: Stages of System Analysis** 19](#_Toc68025205)

[**3.3: Feasibility Study** 20](#_Toc68025206)

[**3.3.1: Types of feasibilities** 20](#_Toc68025207)

[**3.4: Requirement Elicitation** 23](#_Toc68025208)

[**3.4.1: Method used** 23](#_Toc68025209)

[**3.5: Requirements** 23](#_Toc68025210)

[**3.5.1: Hardware Components** 23](#_Toc68025211)

[**3.5.2: Software** 28](#_Toc68025212)

[**3.6: System Design** 28](#_Toc68025213)

[**3.6.1: Types of designs** 29](#_Toc68025214)

[**CHAPTER 4: IMPLEMENTATION, TESTING AND RESULT** 32](#_Toc68025215)

[**4.1: Implementation** 32](#_Toc68025216)

[**4.2: Testing** 32](#_Toc68025217)

[**4.2.1: Collecting data** 32](#_Toc68025218)

[**4.2.2: Data analysis** 33](#_Toc68025219)

[**4.3: Connecting components** 34](#_Toc68025220)

[**4.3.1: Detector module** 34](#_Toc68025221)

[**4.3.2: Translator module** 35](#_Toc68025222)

[**4.4: Results** 35](#_Toc68025223)

[**CHAPTER: FIVE** 36](#_Toc68025224)

[**5.1: Objectives attained** 36](#_Toc68025225)

[**5.1.1: Developed a user-friendly interface** 36](#_Toc68025226)

[**5.1.2: Raised the efficiency of the past gloves by increasing the letters.** 36](#_Toc68025227)

[**5.1.3: Decreased the glove dimensions so that it becomes easier to use effectively.** 36](#_Toc68025228)

[**5.2: Conclusion of the study** 36](#_Toc68025229)

[**5.3: Challenges faced during development** 36](#_Toc68025230)

[**5.4: Further work** 36](#_Toc68025231)

[**REFERENCES** 38](#_Toc68025232)

[**APPENDIX** 40](#_Toc68025233)

[**CODE** 45](#_Toc68025234)

# **List of Figures**

Figure 1-1: Basic Sign Language Gestures

Figure 3-2: Arduino Uno R3

Figure 3-3: Flex sensor

Figure 3-4: Flex bend levels

Figure 3-5: (Flex-sensor Interfacing with Arduino) (Jimb0,2017)

Figure 3-6: (GY-61 Accelerometer with Arduino) (Arduino d, 2015)

Figure 3-7: (Contact sensor basic circuit)

Figure 3-8: (HC-05 interfacing with Arduino) (Currey, 2014)

# **List of Tables**

Table 2-1: Comparison of different approached for sign language detection

Table 3-2: Resources table

Table 3-3: Gant Chart

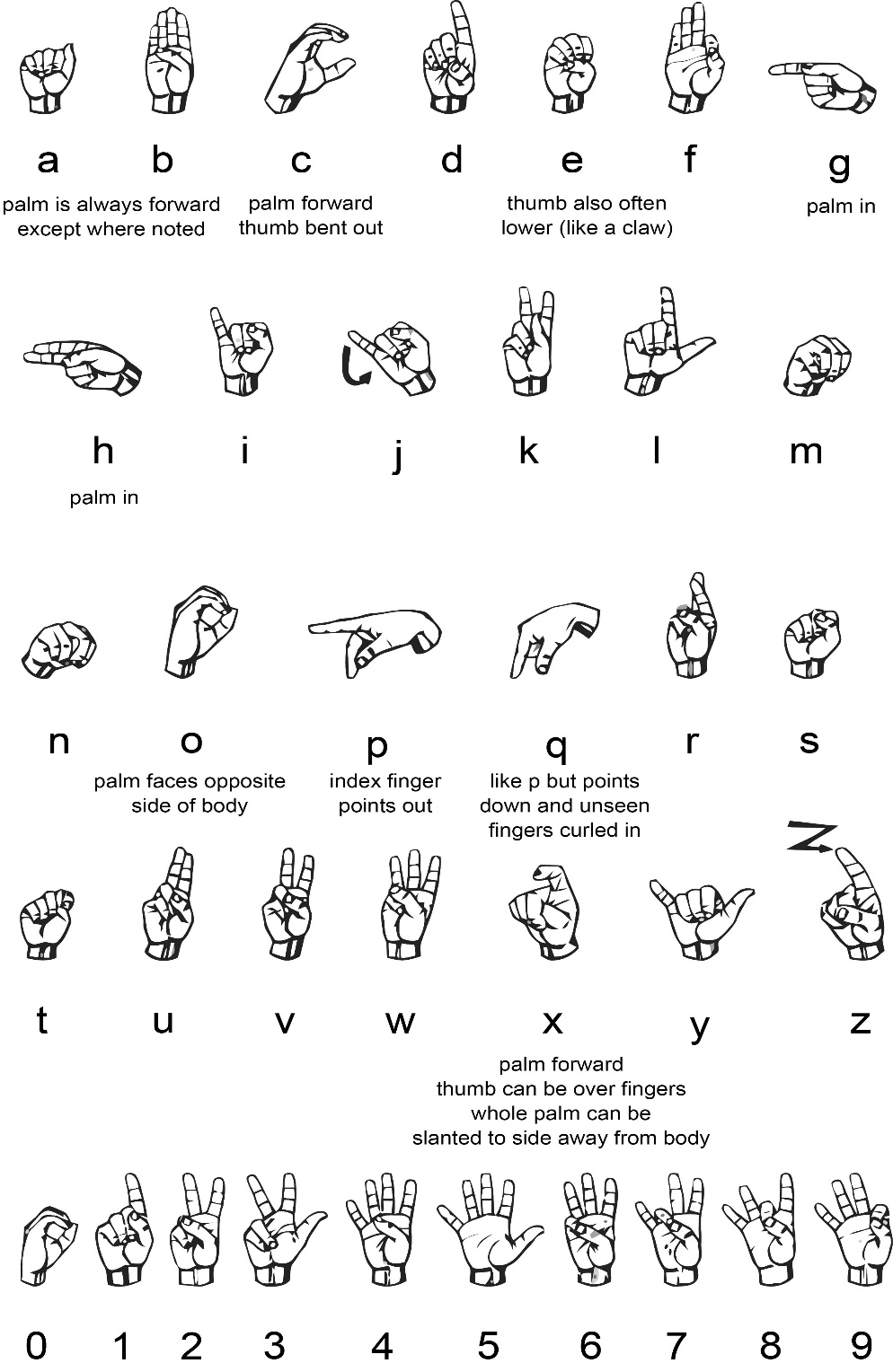
# **CHAPTER 1: INTRODUCTION**

**1.1: Background**

Communication is essential in one’s everyday life. Things like asking for directions, shopping and the preparation of events rely on efficient communication. For verbal people, carrying out one's day with no capacity to speak which assists in conveying thoughts would be quite difficult. Therefore, people require a means of communication even including the non-verbal people even though they can’t communicate verbally.

Every single individual uses a language to have the ability to communicate with someone else. Sign language which was developed by the hearing and speech impaired is used by the developers themselves to communicate with one another. Communication that is done by signing is an extremely organized non-verbal language which uses both non-manual and manual correspondence. Non-manual signs are basically visible like head movements, stance and the orientation of the human body. As for manual signs they include movements and orientations of the hand that passes on average significance (R. Akmeliawati, 2007). “Sign language may be delineated as a way of human communication composed of the usage of one's hand, body and arm movements in addition to facial expressions while not using one's voice.” (signsoflifeasl, 2013)

Sign language was purposely made to allow the non-verbal to have the ability to convey messages and to also connect with society. Below are the fundamental sign language gestures used by the non-verbal.



*Figure 1: Basic Sign Language Gestures*

## **1.2: Problem Statement**

For the hearing and speech impaired sign language is the main means of communication, for them to have the flexibility to communicate with efficiency with one other. The majority of the time, verbal people do not know sign language consequently they are unable to communicate with the non-verbal people who use sign language for communication at all times, developing a large communication barrier between non-verbal and most verbal people.

Thus, studying sign language is needed to reduce the barrier in communication between the verbal and non-verbal people.” (signsoflifeasl, 2013) Sign language on the other hand is also fairly difficult to learn due to its difference in sentence structure and grammar.

## **1.3: Objectives**

* To develop a system which may translate sign language into text and speech with a high level of accuracy for recognizing gestures.
* To raise the efficiency of the past gloves by increasing the letters.
* To decrease the glove dimensions so that it becomes easier to use effectively.

## **1.4: Justification**

In this proposal a system that can help in translating sign language into text and speech to guarantee that effective communication can happen between verbal and non-verbal people will be developed.

The system to be developed is a Smart Glove: Gesture Translator. The Smart Glove is made to target the deaf and mute people so they are able to use it to communicate with normal people. The aim of the Smart Glove is to design a system which can effectively interact with the user so that when worn by a deaf or mute individual it will translate sign language gestures into text and speech with accuracy in real time (with the assistance of single-handed sign language), to ensure that a verbal individual will have the ability to comprehend a deaf and mute person.

The Smart glove will consist of two modes. The first mode is the detection mode. This mode detects the hand gestures. The second mode is the translator mode. This mode translates the hand gestures that have been detected in the initial mode into English text and speech. The translation mode aims to help communication to take place between verbal and non-verbal people. The detection mode is in the Smart Glove and the translator mode is in a phone application (the phone which has the application displays the interpreted signs in the form of text onto its own screen in addition it converts the text into speech). The two modules are connected via Bluetooth Module.

The Smart Glove will support the 26 letters of the English alphabet which the non-verbal people will need to use to communicate in the future. A set of words will later be developed after thorough research and data collection on sign language so as to help the non-verbal introduce and express themselves as well as greet others.

The main reason why this solution is proposed is because sign language mainly involves hand motions. The Smart Glove can use different methods for recognizing gestures. The facts of the various methods will be in chapter 2 of this proposal.

The Smart Glove: Gesture Translator can help non-verbal people make presentations before an audience since the sign language is translated into text and speech.

Moreover, the Smart Glove could be used to aid in the education field, this is because a teacher will have the capability to easily teach students sign language. This helps increase the amount of people in society that understand sign language.

## **1.5: Scope**

The Smart Glove: Gesture Translator is specifically for the non-verbal. Since Sign language for verbal people can be rather difficult to learn due to its difference in sentence structure and grammar. For that reason, the Smart Glove will help with the communication barrier between non-verbal and verbal people. Though the Smart Glove is mainly for the non-verbal, the verbal people can use the translator mode to help them comprehend sign language better.

# **CHAPTER 2: LITERATURE REVIEW**

## **2.1: Introduction**

A review of literature is a compilation of earlier studies on a topic. The literature review analyses scholarly articles, books and alternative resources related to a specific field of study. The purpose of the analysis is to enumerate, define, outline, critically analyze and explain the previous study. The history of sign language, the techniques used in gesture recognition, communication between verbal and non-verbal individuals and the conclusion will be included in this literature review.

**2.2: The background of Sign Language**

There are three entirely distinct primary types of sign language currently being used: American Sign Language (ASL), Pidgin Signed English (PSE), and Signed Exact English (SEE). Assimilation to the Deaf Community for many deaf persons is facilitated by using ASL. To learn ASL fluently, speech-reading or listening abilities are not needed since ASL is a visual language. Among the non-verbal, the most commonly used communication mode is potentially PSE. PSE or Signed English are used by many educators. Since the vocabulary for PSE is taken from ASL, it fits the word order in English. SEE relies on signs taken from ASL and augmented with words, prefixes, tenses, and endings to provide a simple and complete visual demonstration of English. In this proposal, the type of sign language to be used is ASL (signsoflifeasl, 2013).

**2.3: Gesture Recognition Methods**

Currently, systems that translate sign language automatically use two different sensor-based techniques, which will be the data glove and the visual based approach (R. Akmeliawati, 2017). However, a modern approach to understanding hand signs, known as the virtual button approach, has been developed (J. Lim, D. Lee, & B. Kim, 2010). The facts of each approach will be explained in this section and a contrast between the approaches will be shown.

**2.3.1: Data Glove Approach**

In this method, a special assembled electronic glove is used, which includes sensors that detect hand gestures and identify them. The data glove process is used by many commercial sign language translation systems, so it is possible to collect details about a finger's bending along with the hand's 3D orientation using the gloves (R. Akmeliawati, 2007). Less computing power is needed in the system and it is much less difficult to achieve constant understanding. Ten flex sensors, two on each finger, outline the data glove (C. Preetham, G. Ramakrishnan & S. Kumar, 2013). The role of a flex sensor as a variable resistor is to change the resistance when used in conjunction with the bending of the sensor (T.T. Swee, A.K. Ariff, S.-H Salleh, S.K. Seng & L.S. Huat, 2007). The sensors can feel each finger joint's bending point and send the information to the microcontroller. Sensors are located from the association joints of the fingers and palm to the fingertips on the outer layer of the data glove. Furthermore, in order to boost the accuracy of differentiating hand movements, a 3-axis accelerometer is used to detect the change in the acceleration of hand movement in distinctive bearings (W. Jingqiu, & Z. Ting, 2014). The accelerometer is placed on the back of the glove for info. The data glove is very compatible when perceiving both finger-spelling and hand gestures, which include motion and static signals. The data glove may however be costly. Although less costly data gloves can be made, they are more susceptible to noise. In the event that the number of sensors to be used is decreased, the lack of substantial hand signal data will occur. This will then contribute to the loss in precision in the understanding of sign language (R. Akmeliawati, 2007). As wearing, the data glove can also be painful for the signer.

**2.3.2: Visual-Based Approach**

On account of the evolution of technologies, the use of the visual-based approach has expanded. In this technique, a camera captures the signer and then video editing is performed to recognize the signer's hand gestures in order to decode them. This technique is distinct from the approach of the data glove and its advantage over the approach of the data glove is its adaptability of the framework. In addition to lip-perusing, other aspects that may be integrated into the process include the identification of facial gestures and head motions. This technique would be broken into two, utilization hand-crafted shading gloves for use and also in light of the identification of skin color. The signer comes with color-coded gloves for the hand-crafted shading glove technique. Via color segmentation, the data from the signer's photos will be derived from the gloves' color. The gloves are a standard pair of gloves whereby on each fingertip and palm there are some shadings. Color-coded gloves are less expensive as compared to computerized data gloves. This technique utilizes inexpensive devices such as webcams and simple gloves (color-shaded). In RGB (red-green-blue) coloring, the webcams are used in the form of still images and video streams to get images from the signer.

As for the technique of in light of skin-color detection in order to record the signer's moving images with no external technology, the only thing the framework needs is a camera. It turned out to be more widespread and useful for consistent applications. This technique is simple since the user communicates directly with the platform by using an exposed hand to focus the information needed for identification (K. Sharma, & N. Kumar, 2014). The topic of interest can be detected using the color threshold technique that will segment the skin color area in order to monitor the position of the hand. The image acquisition is continuous until a stop sign is displayed by the signer (Y. Madhuri, G. Anitha & M. Anburajan, 2013). The segmented images are then analyzed, after the threshold, to acquire the distinct characteristics of each symbol.

The equipment needs and expenses are reduced by this technique. As compared to discerning symbol gestures, these mechanisms are only sufficient and usable for interpreting alphabets and numbers. The system's accuracy is diminished because it is possible to misinterpret signs with a similar stance to another symbol (Y. Madhuri, G. Anitha & M. Anburajan, 2013). In addition, the method of image acquisition is subject to many ecological issues, such as camera location, background condition, and lighting effects. The distinct height of the signer is another thing to remember. In order to allow adequate light to be seen and examined, appropriate illumination is often needed (R. Akmeliawati, 2007).

**2.3.3: Virtual-Based Button Approach**

A virtual button has the purpose of creating a series of button events by pressing and freeing them. The virtual button will distinguish various kinds of gestures and build appropriate functions. This technique utilizes wrist patterns. A twist action can be discerned. By bending one finger, miniature IR optic sensors may recognize the patterns of the finger flexor tendons on the wrist (the patterns are used to detect finger or hand movements). The IR emitter and IR optic sensor are attached to the bottom of the wrist as the region consists of finger flexor tendons that respond sensitively to finger movements. Voltage values per IR radiation are produced by these sensors. This sensor is used in the framework to track various wrist shape patterns triggered by the displacement of finger flexor tendons on the wrist as fingers move (J. Lim, D. Lee, & B. Kim, 2010).

Unfortunately, this approach is inadequate for sign language interpretation because it needs more intricate finger positions. Sign language movements also use more than just a wrist shape and finger movement. Therefore, for perceiving communication through gestures, this technique is not wise.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Device/components** | **Gestures** | **Environment** | **Accuracy** |
| Data-glove approach | Flex sensor, Accelerometer, Microcontroller. | Fingerspelling (alphabets and numbers), sign gestures. | Practical, no environmental concern. | Higher accuracy if more sensors are used. |
| Visual-based approach | Custom-made glove, computer’s webcam. | Effective for fingerspelling only (alphabets and numbers) | position of the camera, background condition and lighting sensitivity | can be misinterpreted |
| Virtual button | IR optic sensor | Finger and hand movement, not suitable for gestures | no environmental concern. | Overall correctness 88.82% |

*Table 1: Comparison of different approaches for sign language detection*

### **2.3.4: Selection of the Approach**

The data glove technique is favored when referring to the above table, since it is a more suitable tool for identifying sign language movements. The data glove is more practical and is not inhibited by any environmental aspects, such as illumination. In comparison, this technique is known to have greater accuracy over visual-based methods.

## **2.4: Communication between verbal and non-verbal people**

The creation of technology to ease communication between the verbal and non-verbal has been pursued by research programs. The first effort in 1999 to solve the problem was a foundational glove device that could learn to decipher sign language gestures and output the correct terms. Since then, numerous entities, often undergraduate or graduate university students, have each created their own concept projects, each with its own benefits and design options (Princesa Cloutier, 2016).

**2.5: Concepts of Gestures**

There have been several efforts in the past to create a device that could understand and interpret hand movements in sign language through the use of the notion of gestures. In addition to a long time to decipher the movements, these few attempts had a few drawbacks, such as a poor gesture recognition rate. The few attempts which have been made include:

1. CMOS Camera Use.

2. Glove Based Leaf Switches.

3. Copper Glove Based Plate.

4. Glove Based on Flex Sensor (Kanika Rastogi, 2016).

**2.5.1: CMOS Camera Use**

Via the UART serial terminal, the CMOS camera transmits image data. The UART conducts serial-to-parallel conversions on peripheral computer data (in this case, the CMOS camera) and parallel-to-serial conversions on CPU data (Microcontroller in this case). Three measures were used to track hand movements using the CMOS camera:

1. Capturing the gesture's picture

2. Detection of the edge of a picture

3. The image's optimum detection

A few limitations of the CMOS camera are that it is incredibly expensive, highly latent, and each image has 50 KB of memory (Kanika Rastogi, 2016).

### **2.5.2: Glove Based Leaf Switches**

These are like normal switches, except that they are built in such a way that when force is applied to the switch, the switch is locked when two ends come into contact. These leaf switches are located on the fingertips of the glove so that the two terminals of the switch come into contact when the finger is bent.

A downside of the leaf switch-based glove after prolonged use is that the switch would be closed rather than opened until the finger is straight, resulting in inaccurate movement transmission. (Kanika Rastogi, 2016).

**2.5.3: Copper Glove Based Plate**

A copper plate is placed on the palm as a foundation in this prototype. The copper strips display a logic voltage level of one in the resting position. However, the voltage is drained as copper strips touch the ground plate and they show a logic voltage standard of 0.

A downside to using a copper plate is that it makes the glove heavy, making it inappropriate to wear for a long period of time (Kanika Rastogi, 2016).

**2.5.4: Glove Based on Flex Sensor**

"Flex means to "bend" or "curve". The sensor refers to an electronic device which transforms physical energy into power. The Flex Sensor is a resistive sensor that adjusts its resistance to an analog voltage due to the difference in its bend or curvature. This may be a haptic solution consisting of the use of flex sensors to involve physical values for processing (Kanika Rastogi, 2016).

### **2.5.5: Selection of the type of material to be use on the data glove**

This plan would use flex sensors on the data glove on the basis of the various materials because they are more powerful in the long run.

**2.6: Conclusion**

Sign language tends to ease non-verbal and verbal communication. There is also a barrier between the verbal and the non-verbal, though. This initiative attempts to resolve the gap in conversations between the verbal and the non-verbal. The Smart Glove will be used by the non-verbal to perform sign language and will translate the signals to speech in response in order to assist with the interaction barrier. The value of this Smart Glove is that it is perfect for a person's everyday life.

# **CHAPTER 3: METHODOLOGY**

## **3.1: Introduction**

A methodology is a variety of methods, protocols, procedures, techniques, systems, and concepts. Methodologies are precise, strict in project management, which usually include a sequence of steps and tasks for each point in the life cycle of the project. This chapter will decide how during the continuous execution process, the project will be designed, created, managed and distributed before complete completion and termination.

## **3.2: System Analysis**

The word System originated from the Greek word Systema, which means a coordinated relationship to accomplish any mutual purpose or goal between any pair of elements. A structure is an orderly grouping of mutually beneficial components united on the basis of a concept to achieve a particular goal."

System Analysis is a method in which details are gathered and decoded, problems are differentiated, and a system is decomposed into its components. System analysis is done with the aim of observing a system or its elements in order to understand its aims. It is a problem-solving strategy that strengthens the system and ensures that any single system aspect performs successfully to achieve its function. What the machine can do is defined in the word analysis.

System analysis is important for the production of the project plan in that it offers a pathway across the different activities involved in the process for solutions in the system. Via these multiple functions, a system's average output can be quickly adjusted or increased and error happenings will eventually be minimized.

### **3.2.1: Stages of System Analysis**

There are four levels of system analysis, which are:

1. To assess and analyze the new business or operational system
2. To assess the demands of the framework
3. To configure the specifications and structure them
4. To pick the right approach for alternative design

For the Smart Glove: Gesture Translator, system analysis involves the following steps:

1. Identify the concern as verbal and non-verbal individuals seek to engage with each other.
2. Identify the demands of the system.
3. Prioritize the needs.
4. Perform an elicitation and feasibility review of specifications.
5. To represent the events that will take place, draw templates.

## **3.3: Feasibility Study**

The Feasibility Analysis can be used as a preliminary appraisal that allows managers to determine whether or not a framework study should be viable for development or not.

For the advancement of the project, this is important in that:

* It recognizes the opportunity for an existing system to be strengthened, for a new system to be developed and for refined projections to be generated for further system improvement.
* It is used to gain the outline of the issue of a project to determine whether or not there is a suitable or acceptable solution.
* Instead of addressing the dilemma, the primary goal of a feasibility project will be to gain a problem spectrum.
* A structured framework proposal act may be the result of a feasibility review as a decision statement that comprises the whole design and complexity of the proposed system.

### **3.3.1: Types of feasibilities**

In this plan, there are three types of feasibilities that will be used:

1. Resource feasibility
2. Economic feasibility
3. Schedule feasibility

#### **3.3.1.1: Resource feasibility**

This requires the assets to be included in planning the layout. Both hardware and software components are required for the system. The software specifications then include: IDE for Android Studio and Arduino IDE. As for the hardware specs, they include: Arduino nano, Flex Sensors, Jumper wires, 47K Resistors. Long cables, Glove, Gy-61 DXL335 3-axis accelerometer module, contact sensors, Emic 2 Text to Speech Module, HC-05 Bluetooth module, MT3608 step up converter, USB and breadboard converter.

#### **3.3.1.2: Economic feasibility**

This concerns the financial calculation of the project's gains, which can be real or intangible, and the money that would be used to build up the project. Open and free tools are both resources that can be used during development; thus, not a significant amount of money will be spent (BrightHubProjectManagement, N.D).

**Resource Table**

|  |  |
| --- | --- |
| **RESOURCES** | **COST** |
| **GY-61ADXL335-3AXIS ACCELEROMETER** | KSH 500 |
| **ARDUINO UNO R3** | KSH 1,200 |
| **BLUETOOTH MODULE HC05** | KSH 600 |
| **BREADBOARD MINI** | KSH 100 |
| **5 FLEX SENSOR 4.5”** | KSH 2,500 x 5 |
| **GLOVE** | KSH |
| **JUMPER WIRES MALE-FEMALE 40PC** | KSH 100 |
| **5 47K RESISTORS** | KSH 3 x 5 |
| **TOTAL** | KSH 12,715.00 |

*Table 2: Resources table*

#### **3.3.1.3: Schedule feasibility**

Schedule Feasibility is defined as the possibility of a project being finished by a projected due date within its scheduled time limits. If a project has a high probability of being finished on time, so the viability of its timeline is measured as relevant. This means that the project can be finished within the agreed time period or timetable. It also verifies and validates whether or not the project's deadline is fair.

A table illustrating the timeframe in which the project will be finished is below.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Task | Duration (Weeks) | | | | | | | | | | | | | |
| First semester | | | | | | | Second semester | | | | | | |
| 2 | 4 | 6 | 8 | 10 | 12 | 14 | 2 | 4 | 6 | 8 | 10 | 12 | 14 |
| Planning |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Project Requirements |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Analyzing and design |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Project development |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Project testing  and maintenance |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Documentation |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

*Table 3: Gant Chart*

## **3.4: Requirement Elicitation**

Requirement Elicitation is the method of finding the specifications for a given system by engaging with consumers, end users, users of the system and others who are interested in system development.

### **3.4.1: Method used**

The first step in the elicitation of the criteria for these proposals is to gather evidence from related projects in the same sector and to figure out which elements have been used in similar projects. Each variable is then independently analyzed to learn how each function and the importance each provides. In order to be able to select the right component that matches the specifications, a standard circuit was designed for each component and the output obtained was recorded and saved in order to be evaluated and compared with accurate components.

The second step is to learn ASL and how to represent the alphabets of English (Refer to Figure 1). Upon learning the alphabets, they are classified into groups, each group would have alphabets that are normally identical in the representation of signs. Each group is then separated into sub-groups of identical alphabets. The alphabets are differentiated within the sub-groups according to the sensors that may be used by wearing the smart glove (Refer to Appendix).

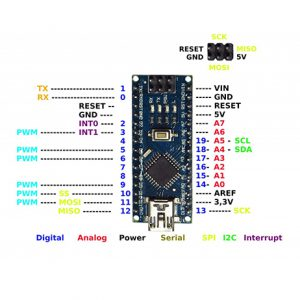
## **3.5: Requirements**

### **3.5.1: Hardware Components**

The following is a list with their respective descriptions of the hardware components needed for the proposal:

Arduino Nano:

A compact, absolute, and breadboard-friendly board based on the ATmega3288 is the Arduino Nano (Arduino Nano 3. x). It has the same features as the Arduino Duemilanove, more or less, but in a different compartment. It just lacks a DC power port, which operates with a Mini-B USB cable instead of a normal one.



*Figure 2: Arduino Nano*

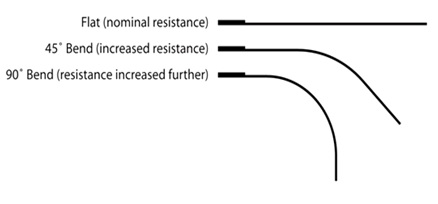
5 Flex sensors:

This is a resistor for a variable. The resistance depends on how far the sensor is bent, because it is a sensor that senses the rise and decrease of resistance when bent. It is used for the identification of gestures and it is placed on one's finger. It has two dimensions:

* 2.2” (5.588cm) long
* 4.5” (11.43cm) long

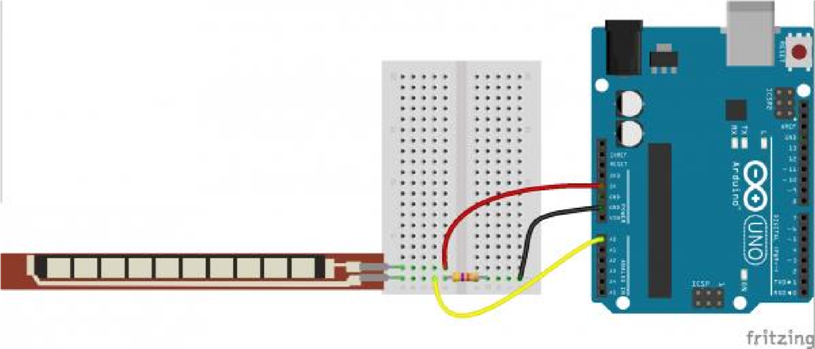
In the proposal, I choose to use the size of 4.5 to be acceptable for the length of the finger. The easiest way to implement this sensor into the proposal is to use it in a voltage divider circuit that allows one to know when the sensor is being flexed and when it is not by attaching it to the comparator circuit. One resistor is required for this circuit (voltage divider circuit) (Jimb0, 2017).

As the sensors are mounted in a glove, when bent It generates a resistance output correlated to the radius of the bend-the smaller the radius, the greater the value of the resistance. For the flex sensor, the bending resistance frequency ranges from around 30Kohm to 125Kohms. Usually, a value between the minimum and maximum resistance values is a safe alternative. In the proposal, I used a 47Kohms resistor (Electronics, 2015).



*Figure 3: Flex sensor* *Figure 4: Flex bend levels*

An example circuit representing the interfacing flex sensor with the Arduino is the following design:



*Figure 5: (Flex-sensor Interfacing with Arduino) (Jimb0,2017)*

10K Resistors (5 Resistors):

Each flex in the circuit is attached to this. Each resistor has its own value, which is 10000, and the code describes and uses this value.

Jumper wires:

This is used to connect all parts together.

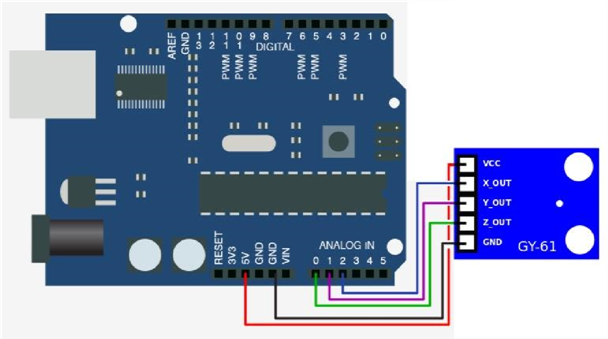
Glove:

It's connected to the circuit.

GY-61 ADXL335 3-axis accelerometer module:

It's a three-axis sensor accelerometer module built on an optimized ADXL335 circuit. With remarkably low noise and power consumption, the ADXL335 is a triple axis accelerometer. (Robot Park, 2015) The accelerometer acts as a resistor whose resistance is proportional to the acceleration of the corresponding pin and axis. This is used to identify the direction of one's hand.

The following figure illustrates the design circuit that represents an accelerometer sensor interfacing with the Arduino:



*Figure 6: (GY-61 Accelerometer with Arduino) (Arduino d, 2015)*

Contact Sensor:

It's a basic 10k resistor circuit with an Arduino that you close/open by attaching or disconnecting two pieces of silver to give the Arduino digital input 0/1.



*Figure 7: (Contact sensor basic circuit)*

TheHC-05 Bluetooth module:

It functions as a transmitter. The HC-05 has the firmware 'full' on it. So, it may act as a module master or slave. In order to enter the configuration mode and set the slave and master option, the figure below shows howHC-05 is interfaced with the Arduino, the State pin is connected to VCC. (Currey, 2014)

I referred to a tutorial by Martyn Currey that very well explains this process in order to make the two modules communicate with each other and pass data (Currey, 2014).



*Figure 8: (HC-05 interfacing with Arduino) (Currey, 2014)*

Breadboard Mini:

The function of the breadboard is to make fast electrical connections between parts, such as resistors, LEDs and capacitors, so that you can evaluate your circuit before welding it together securely.

### **3.5.2: Software**

A list of the software needed for the proposal is given below:

Android studio IDE:

For Android application development, Android Studio is the official integrated development environment (IDE). It is based on the IntelliJ IDEA, an advanced software development environment for Java, which integrates its code editing and developer tools. The mobile application will be built with this.

Arduino IDE:

The Arduino IDE (Integrated Development Environment) is a java-written cross-platform framework which is derived from the IDE for processing and writing this prototype programming language. It provides a code editor with features such as syntax highlighting, brace matching, and 25 indentation automation, and can also compile and upload programs with a single click to the board (Kanika Rastogi, 2016). The programming language to be used is C# in this project.

## **3.6: System Design**

It is a procedure by which a new business structure is designed or an existing system is replaced by specifying its parts or components to fulfill particular specifications. You ought to fully consider the old system before preparing, and decide how devices can better be used to work efficiently. System Architecture depends on how to meet the system's target.

For the development of the project proposal, system design is critical in that it provides ample comprehensive data and information about the system and its system elements to allow the application to be compatible with architectural entities as specified in the system architecture models and views.

### **3.6.1: Types of designs**

There are two kinds of designs for the system:

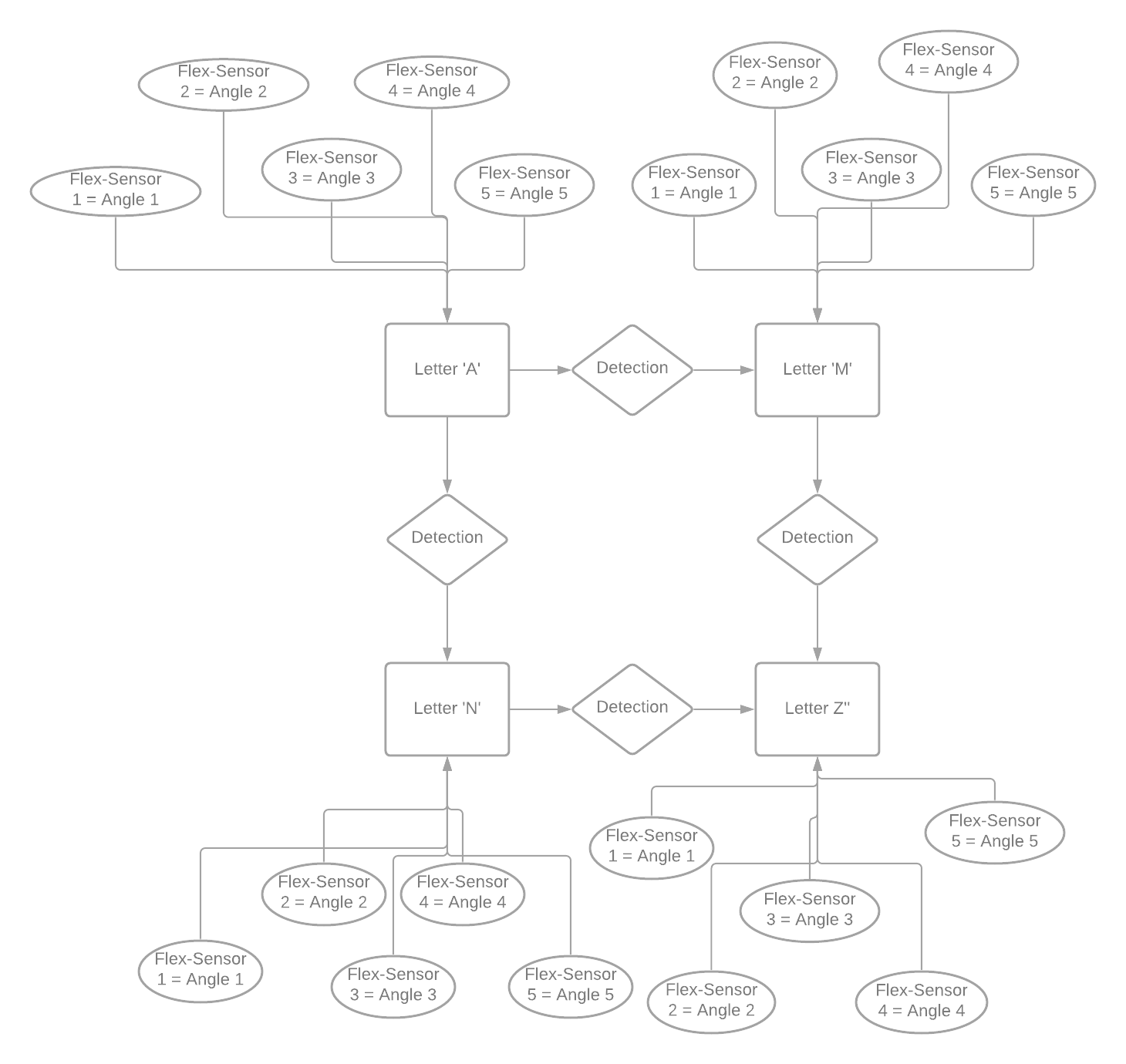
* Logical design
* Physical design

#### **3.6.1.1: Logical design**

This is an abstract conceptual style. In which the physical implementation specifics are not yet discussed, but only the description of the categories of information required is addressed. The logical design method involves the organization of knowledge into a set of logical associations called entities and attributes. It is used to represent the system's data flow, inputs and outputs.

##### **3.6.1.1.1: Entity Relationship Diagram**

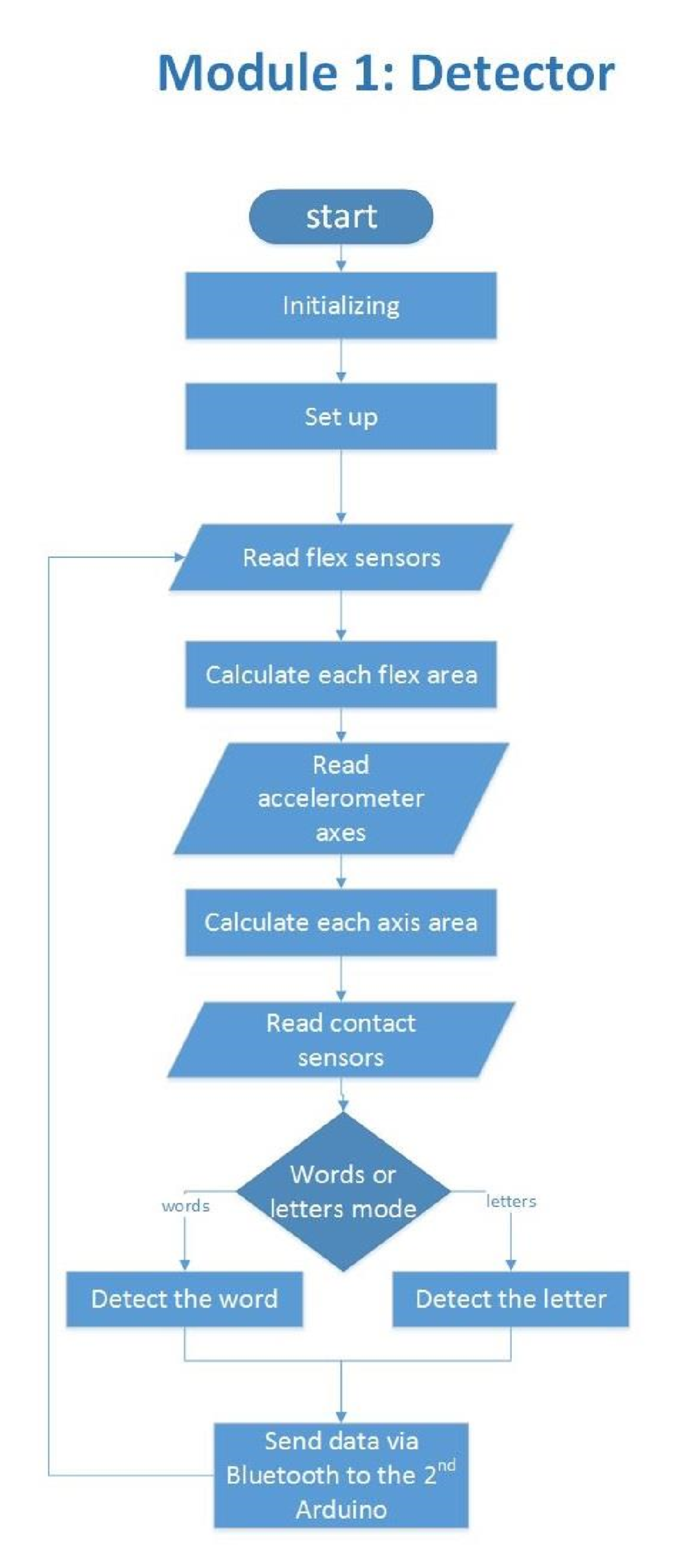
A type of structural diagram for use in database design is the Entity Relationship Diagram, also known as the ERD, ER Diagram, or ER model. There are numerous symbols and connectors in an ERD that visualize two essential data. It is essential to systematically aid one to evaluate data requirements to create a well-designed database.



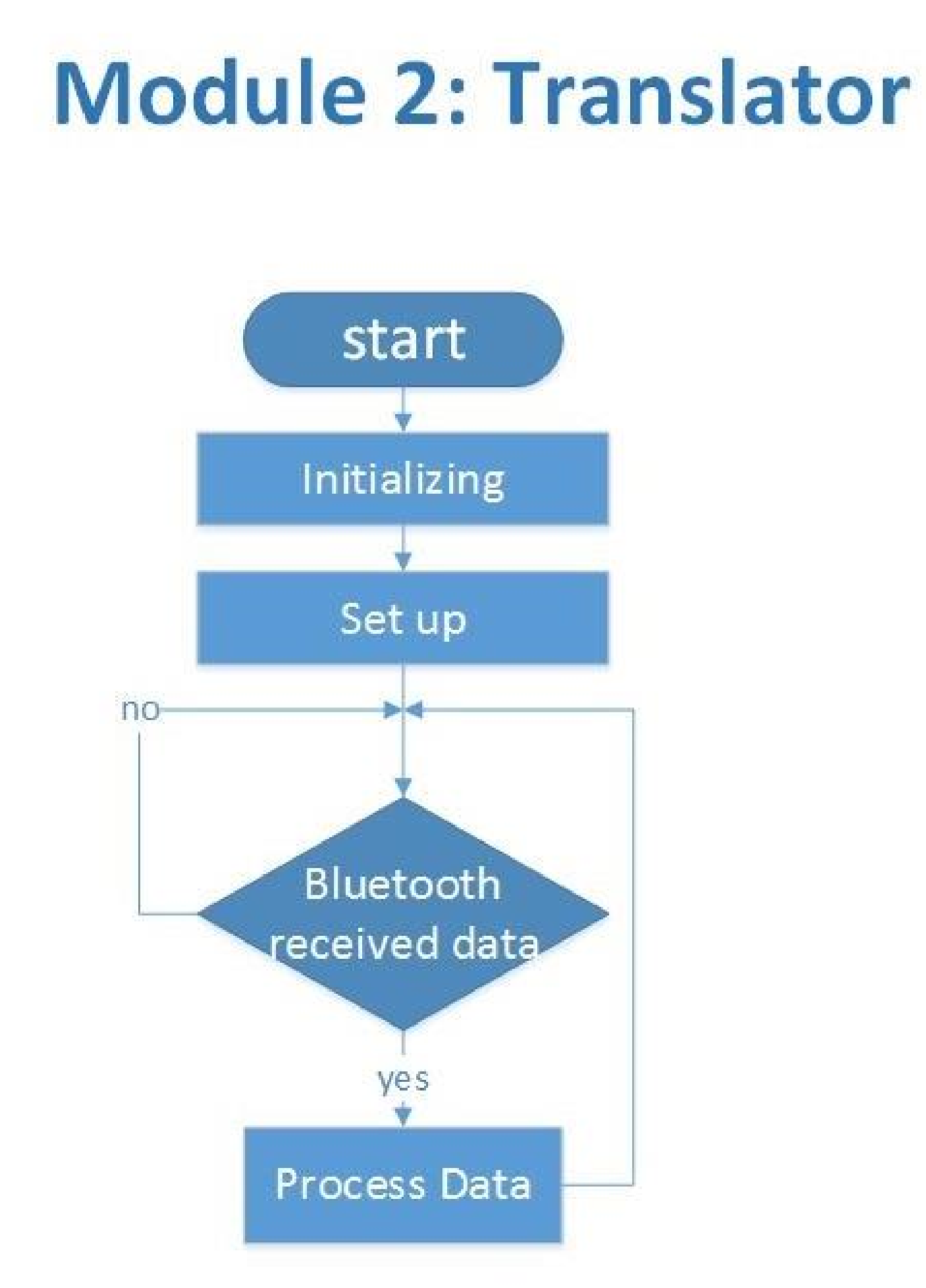
*Figure 9: (Entity Relationship diagram)*

##### **3.6.1.1.2: Data Flow Diagram**

A data flow diagram is a schematic representation of the data flow through an information system, modeling the facets of its operation.



*Figure 10: (Detector module code flow chart)*



*Figure 11: (Translator module code flow chart)*

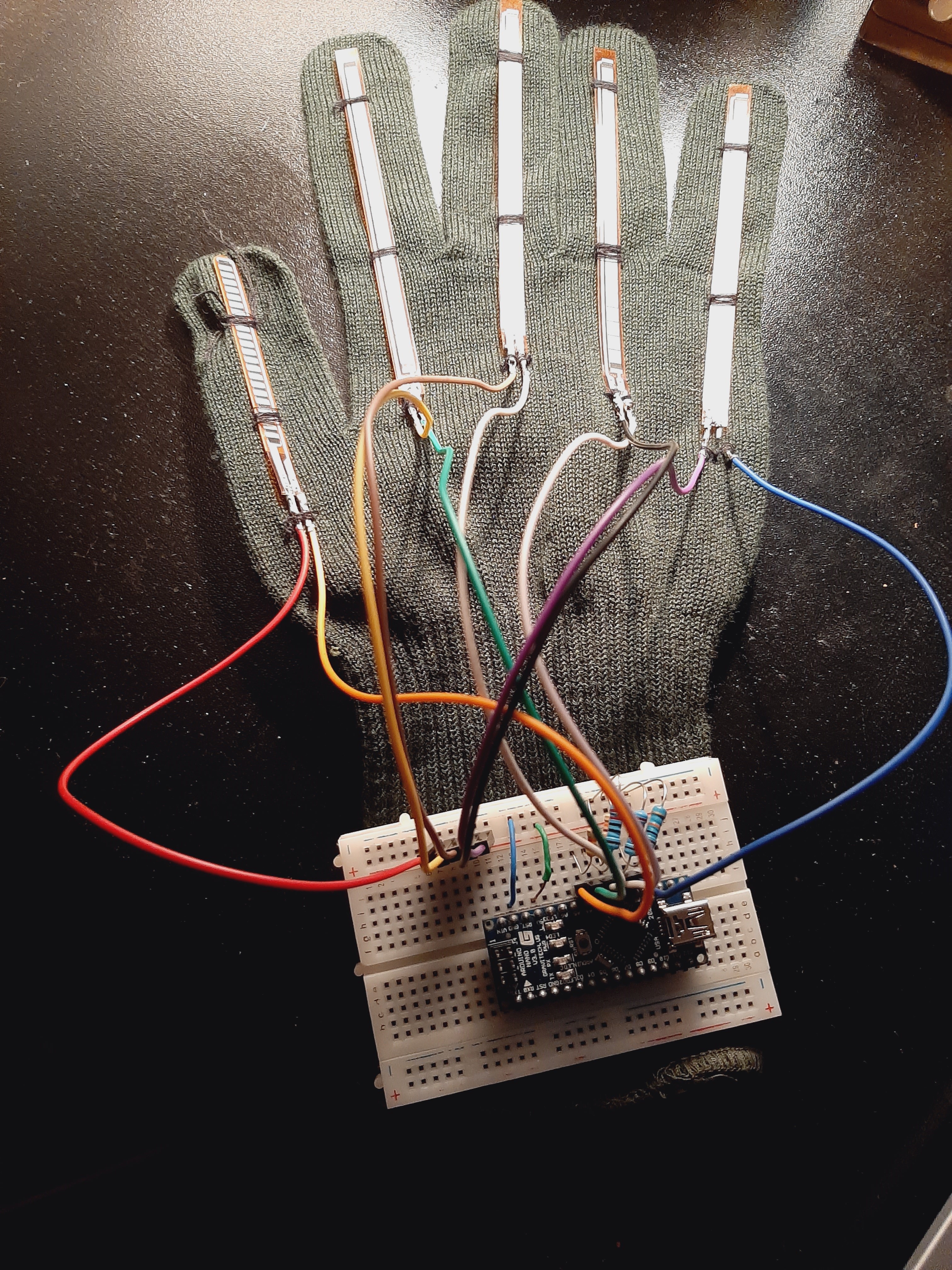
#### **3.6.1.2: Physical design**

The physical design is the method of converting a summary of the circuit into a physical layout that specifies the location of the cells and the routes between them for interconnections.

# **CHAPTER 4: IMPLEMENTATION, TESTING AND RESULT**

## **4.1: Implementation**

The project prototype includes 5 flex sensors, an Arduino nano and an ADC circuit comprised of a comparator and resistors, as seen in Figure below.



*Figure 12: (Glove prototype)*

## **4.2: Testing**

### **4.2.1: Collecting data**

The first move was to gather information about related projects in this area, as well as the materials used in these types of projects. Following that, I analyzed each part independently to learn how it functions and what values it provides. Each component's basic circuit was designed, and its findings and assessment results were captured and recorded in order to be evaluated and correlated with consistent components so that I could choose the right one for our needs.

The second move was to read about American Sign Language (ASL) and how the English alphabets are interpreted (Refer to Appendix). To begin, I divided the alphabets into groups, with each group comprising alphabets with sign language representations that are broadly identical. Then, for each group, I created a table in which I assigned it to alphabet sub-groups. Finally, I determined the sensors that will be used to distinguish between identical alphabets in each sub-group. (For more information, see the Appendix.)

### **4.2.2: Data analysis**

Following the conclusion of the data collection process, it was crucial to review the gathered data in order to determine what I ultimately required to create the Smart Glove: Gesture translator. I discovered that the glove must be able to detect hand movements, finger compressions, and touches between certain fingers after researching other projects and mastering ASL.

#### **4.2.2.1: Hand compressions**

Flex-sensors are used in the vast majority of programs in this field to sense hand compression. As a result, I opted to depict the five hand fingers with five flex sensors. Since a Flex-sensor is basically a variable resistor, the resistor value as a function of compression angle is the output data. In the Arduino IDE, I wrote code to help see and analyze the actions of flex sensors. (Refer to Appendix) I attached the flex sensors to the Arduino Nano's analog inputs. For each flex, the code reads the analog value from the analog inputs and describes them as an analog signal. As I found, the signal varies over time based on how often the flex sensor is squeezed.

It is difficult to analyze these values and use them to symbolize a specific hand shape. Since it is almost impossible to generate the same alphabet representation and float values for the flex sensors resistors. To solve this problem, I used the following procedure:

I ran through all of the alphabets for each flex-sensor and recorded the flex value for each one. Based on this, I specified the range for each flex sensor between its minimum and maximum observed values. Finally, I separated these ranges into areas so that the flex value for a certain alphabet could differ marginally without conflicting with another region. The flex sensor value corresponds to the flex sensor bent angle, which is determined using three parameters: the flex resistance value for this angle, the flex resistance value when the flex is straight, and the flex resistance value when the flex is bended to 90o. (For more information, see the Appendix.)

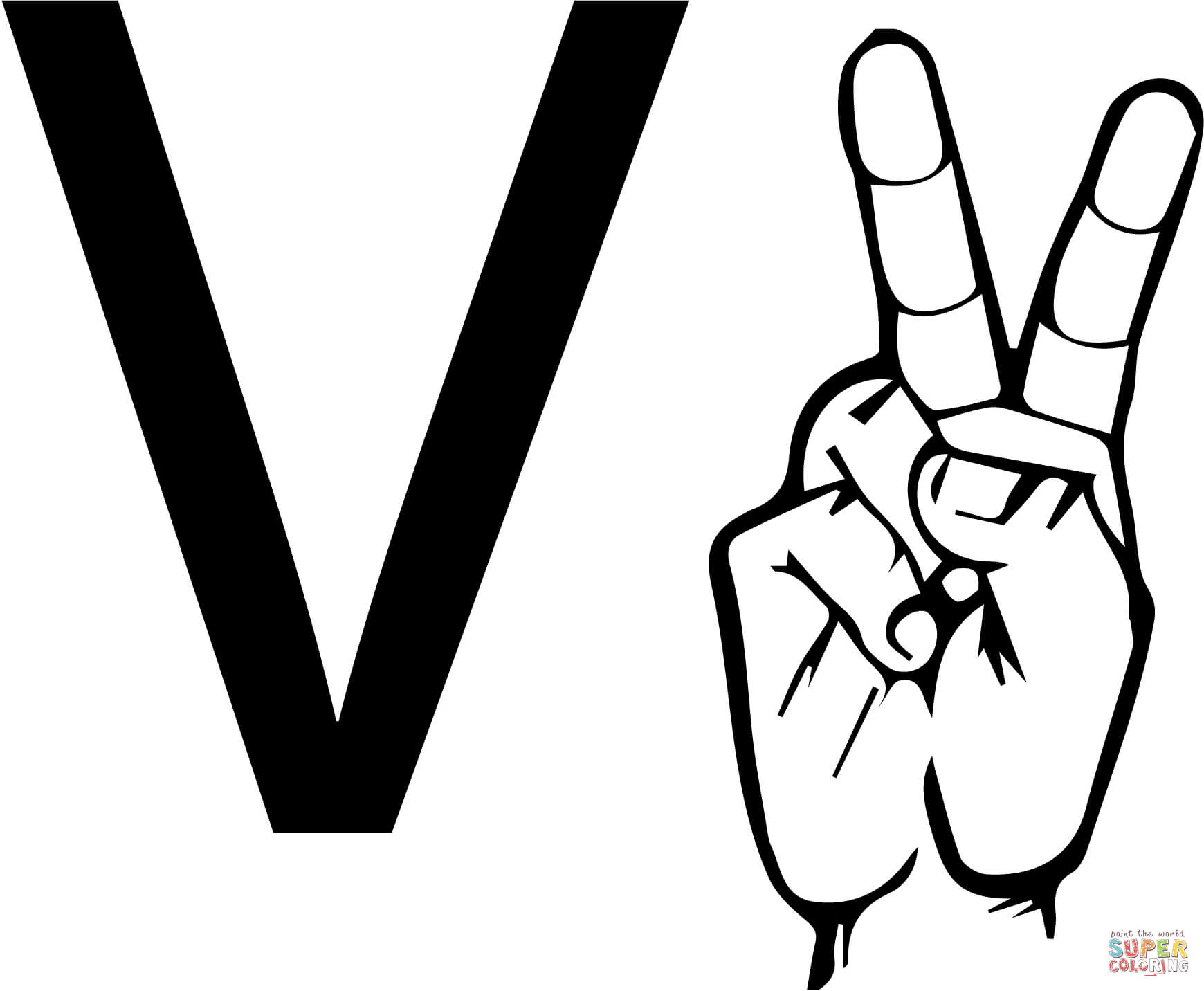
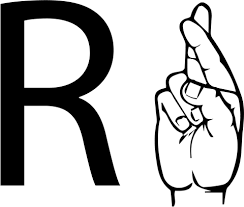
#### **4.2.2.2: Hand gestures**

Particular sensors are capable of detecting hand signals. Initially, I used the performance date obtained by attaching a gyroscope to Arduino, but I discovered that it generates erratic values and that the outcome varies based on hand motions and location. Analyzing the performance data from attaching the GY-61 Accelerometer, on the other hand, guided me to the correct part to use because it senses hand movements in XYZ axes. So, I took the XYZ values for all alphabet representations and dispersed them into areas similarly to how I distributed flex sensors into areas. (See Appendix for more information.)

In this way I am able to distinguish between alphabets that had the same flex sensor region values in this manner.

**4.2.2.3: Contacts between fingers**

I found that the alphabets "V" and "R" have the same flex sensors and accelerometer field values after extracting the alphabets from the same flex values using the accelerometer regions. The only distinction was that in the alphabet "R" representation, the index and middle fingers were crossed, but not in the alphabet "V" representation, as seen in the figure below.



*Figure 14: (V and R representation problem)*

To fix this, I built contact sensors that sense the interaction between two fingers by forming a closed circuit when they are crossed and an open circuit when they are not crossed. (The circuit architecture was previously specified in Hardware Part-2.)

**4.3: Connecting components**

Two components make up the Smart Deaf-Mute Glove. Bluetooth is used to communicate with the Detector and Translator units. I developed the project in this manner to provide a safe haven for deaf-mutes when conversing with others. Furthermore, the glove circuit will be light enough to be carried. To highlight this point even more, I designed it as a PCB to be as light as possible by hand.

**4.3.1: Detector module**

The aim of this module is to detect hand compressions, gestures, and contact between fingers. The data processor was the Arduino Nano, which was controlled by five 4.5-inch Flex sensors, and a GY-61 accelerometer. In addition to the contact sensor circuits. The output of the data processing is an alphabet or a letter, which is sent to the second module through the HC-05 Bluetooth module. The circuit for this module was built in Eagle IDE in order to generate a PCB. A Lithium battery is connected to a step-up converter as the power supply. To obtain the lightest weight, I chose this method of providing voltage.

### **4.3.2: Translator module**

The translator model is made up of an application, a Text-To-Speech module, a speaker, and an HC-06 Bluetooth module linked to an Arduino Nano. This module collects data from the detector module and displays it on the serial monitor before translating it to audio, which is output via a speaker. By adding a 9V battery to the Arduino's Vin pin, I was able to supply voltage to this board.

Since all of the components in both modules need 5V to operate, I powered the Arduinos via the Vin pins and the components via the Arduino VCC pins. Considering the maximum current each module absorbs from the Arduino and the maximum current the Arduino can have on the VCC pins according to the Nano and Mega Arduino documentation. (LearnArduino, 2017)

## **4.4: Results**

*Figure 13: (“” Gesture result at smart phone application)*

# **CHAPTER: FIVE**

## **5.1: Objectives attained**

I set out some targets that I wanted to achieve by the conclusion of the device creation phase at the start of the project; in the end, I was able to meet the majority of the goals I set out.

### **5.1.1: Developed a user-friendly interface**

The framework has a user-friendly GUI from which non-verbal users can quickly use to communicate. The user can easily use the glove to convert sign language into text and speech.

### **5.1.2: Raised the efficiency of the past gloves by increasing the letters.**

All the letters of the alphabet have been integrated into the system compared to past systems. There for it is slightly more advanced in terms of quantity.

### **5.1.3: Decreased the glove dimensions so that it becomes easier to use effectively.**

The size of the glove was reduced due to ensuring there was no unnecessary wiring. Therefore it is more simple and easier to understand.

## **5.2: Conclusion of the study**

To conclude, society is increasingly evolving and gravitating toward technological know-how. Technology is not a static or obsolete field; rather, one that is rapidly changing, with new trends emerging on a daily basis. It is beyond time for us to shift towards trends that evolve and improve rapidly. The use of the Smart Glove: Gesture Translator is important for fast communication between non-verbal and verbal people. Therefore, this will help in decreasing the communication barrier.

Also, this project focuses on the ASL alphabets as a means to enable the use of words that form sentences based on signs through measuring the bend of a person's fingers.

## **5.3: Challenges faced during development**

I.) Solving errors in code

II) Connecting jump wires on breadboard

III.) Converting text to speech

IV.) Integrating the system to work

V.) Soldering the wires without affecting the resistance of the sensors.

VI.) Calculating the angles was difficult because the resistance is not stable.

## **5.4: Further work**

This tool may be:

1) Further integrated with other programs that will continue to create opportunities for deaf and hard of hearing people.

2) Geared up with the controller to provide home automation on finger tips.

3) Used in combination with a fitness sensor to track the individual’s health.

4) Added words that only need one hand to translate sign language.

5) Paired with another glove together so as to improve the variety of signs that can be done by two hands.

6) Further integrated to be used as a remote control like for a TV, toy car, helicopter, drone and many more.

# **REFERENCES**

Arduino, d. (2015, August 23). index.php. Retrieved from https://forum.arduino.cc: https://forum.arduino.cc/index.php?topic=343426.0

ARDUINOWEBSITE. What is Arduino? [Online]. Available: https://www.arduino.cc/en/Guide/Introduction [Accessed 20-10-2017 2017].

(BrightHubProjectManagement, N.D) Explaining the different types of feasibility study

Retrieved from https://www.brighthubpm.com/project-planning/56372-types-of-feasibility-

studies/

C. Preetham, G. Ramakrishnan & S. Kumar, “Hand Talk- Implementation of a Gesture Recognizing Glove”, 2013 Texas Instruments India Educators’ Conference (TIIEC), Bangalore, April 2013, pp 328-331

Currey, M. (2014, October 28). /arduino-with-hc-05-bluetooth-module-at-mode/. Retrieved from http://www.martyncurrey.com: http://www.martyncurrey.com/arduino-with-hc-05- bluetooth-module-at-mode/

Curry, M. (2014, November 19). /connecting-2-arduinos-by-bluetooth-using-a-hc-05-and-a-hc06-easy-method-using-cmode/. Retrieved from http://www.martyncurrey.com: http://www.martyncurrey.com/connecting-2-arduinos-by-bluetooth-using-a-hc-05-and-ahc-06-easy-method-using-cmode/

EELCTRONICS, L. A. How to Build a Flex Sensor Circuit with a Voltage Comparator [Online]. Available: http://www.learningaboutelectronics.com/Articles/Flex-sensorcircuit-with-a-voltage-comparator.php [Accessed 20-10-2017 2015].

Jimb0. (2017). flex-sensor-hookup-guide. Retrieved 2017, from https://learn.sparkfun.com: <https://learn.sparkfun.com/tutorials/flex-sensor-hookup-guide>

J. Lim, D. Lee, & B. Kim, “Recognizing Hand Gesture using Wrist Shapes”, 2010 Digest of Technical Papers of the International Conference on Consumer Electronics (ICCE), Las Vegas, 2010, pp 197-198

KANIKA RASTOGI , P. B. May 2016. A Review Paper on Smart Glove - Converts Gestures into Speech and Text. International Journal on Recent and Innovation Trends in Computing and Communication 4, 92 - 94.

K. Sharma, & N. Kumar, “Hand Gestures Recognition for Deaf and Dumb”, International Journal of Computer Application and Technology (s), May 2014, pp. 10-13

LearnArduino. (2017). /Main/ArduinoPinCurrentLimitations. Retrieved from

http://playground.arduino.cc:http://playground.arduino.cc/Main/ArduinoPinCurrentLimitations

PRINCESA CLOUTIER , A. H. R. M. 2016. Prototyping a Portable, Affordable Sign Language Glove. Bachelor of Science, Faculty of Worcester Polytechnic Institute.

R. Akmeliawati, “Real-Time Malaysian Sign Language Translation using Colour Segmentation and Neural Network”, Proc. of the IEEE International Conference on Instrumentation and Measurement Technology 2007, Warsaw, 2007, pp 1-6.

Robot Park, D. (2015). GY-61-DXL335-3-Axis-Accelerometer-Module. Retrieved from http://www.robotpark.com: http://www.robotpark.com/GY-61-DXL335-3-AxisAccelerometer-Module

SIGNSOFLIFEASL. February 27, 2013. 3 Forms of Sign Language: ASL vs. PSE vs. SEE [Online]. Available: https://signsoflifeasl.wordpress.com/2013/02/27/3-forms-of-signlanguage-asl-vs-pse-vs-see/ [Accessed 5-4-2017 2017].

T.T. Swee, A.K. Ariff, S.-H Salleh, S.K. Seng & L.S. Huat, “Wireless Data Gloves Malay Sign Language Recognition System”, 2007 6th International Conference on Information, Communications & Signal Processing, 2007, pp 1-4

W. Jingqiu, & Z. Ting, “An ARM-Based Embedded Gesture Recognition System Using a Data Glove”, The 26th Chinese Control and Decision Conference (2014 CCDC), 2014, pp 1580-1584

Y. Madhuri, G. Anitha & M. Anburajan “ Vision-based Sign Language Translation Device”, 2013 International Conference on Information Communication and Embedded Systems (ICICES), 2013, pp 565-568

# **APPENDIX**

* The distribution of ASL alphabets to groups and sub-groups is illustrated in the following table:

|  |  |
| --- | --- |
|  | Group 1: A, E, M, N, S, T |
| A | Thumb flex differs (not bended) |
| S, T, E | Same acc. Reading but M, N differs in the acc. reading from them |
| S, T | Differs in thumb flex reading with using contact sensors to differentiate them from A |
| E | Differs in flex sensors readings and contact sensors |
| M, N | Differs from each other using contact sensors |
|  | Group 2: B |
| B | The four fingers are not bended and the contact sensors contacted with each other |
|  | Group 3: C, O |
| C, O | Differs in acc. Reading and contacts between thumb and index fingers |
|  | Group 4: D, G, Z, X |
|  | D, X, Z |
| D, X | Differs in flex reading and acc. |
| D, Z | Differs in acc. Reading but they have similar flex reading |
| Z, X | Differs in flex sensors readings but they have similar acc. reading |
| G | Differs from them all in acc. Reading and thumb flex reading |
|  | Group 5: F |
| F | Three fingers are not bended |
|  | Group 6: H |
| H | Similar to G in acc. Reading and number of not bended fingers |
|  | Group 7: I, J |
| I, J | Same movement but differs in acc. reading |
|  | Group 8: Y |
| Y | Differs from I and J in thumb flex reading |
|  | Group 9: K, U, V, R |
| U, V | Differs in contacts sensors |
| V, K | Differs in thumb flex reading |
| U, R | Differs in Ring flex reading |
|  | Group 10:L |
| L | Two fingers are not bended |
|  | Group 11: P |
| P | Three fingers are not bended and the contact sensors and acc. reading |
|  | Group 12: Q |
| Q | Two fingers are not bended and the acc. reading |
|  | Group 13: W |
| W | Three fingers are not bended but they are different from F fingers |

Table 4: (Groups of ASL alphabets)

* The flex sensors area distribution for alphabets:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Letter | Flex0 | Flex1 | Flex2 | Flex3 | Flex4 |
| A | 2 | 2 | 3 | 4 | 1 |
| B | 1 | 1 | 1 | 1 | 2 |
| C | 1 | 2 | 2 | 3 | 1 |
| D | 2 | 3 | 3 | 1 | 2 |
| E | 2 | 3 | 3 | 4 | 3 |
| F | 1 | 1 | 1 | 3 | 1 |
| G | 2 | 3 | 3 | 1 | 1 |
| H | 2 | 2 | 1 | 1 | 2 |
| I | 1 | 2 | 2 | 4 | 2 |
| J | 1 | 2 | 2 | 4 | 2 |
| K | 2 | 2 | 1 | 1 | 1 |
| L | 2 | 3 | 3 | 1 | 1 |
| M | 2 | 2 | 2 | 4 | 3 |
| N | 2 | 2 | 2 | 3 | 2 |
| O | 1 | 2 | 2 | 3 | 2 |
| P | 1 | 1 | 1 | 1 | 1 |
| Q | 2 | 3 | 3 | 1 | 1 |
| R | 2 | 2 | 1 | 1 | 2 |
| S | 2 | 3 | 3 | 4 | 2 |
| T | 2 | 2 | 3 | 2 | 1 |
| U | 2 | 2 | 1 | 1 | 2 |
| V | 2 | 2 | 1 | 1 | 2 |
| W | 2 | 1 | 1 | 1 | 2 |
| X | 2 | 3 | 3 | 2 | 3 |
| Y | 1 | 2 | 2 | 4 | 1 |
| Z | 2 | 3 | 3 | 1 | 2 |

Table 5: (Accelerometer areas for alphabets)

* The distribution of accelerometer areas for alphabets:

|  |  |  |  |
| --- | --- | --- | --- |
| Letter | X-axis | Y-axis | Z-axis |
| A | 2 | 1 | 3 |
| B | 2 | 1 | 3 |
| C | 2 | 4 | 3 |
| D | 2 | 5 | 3 |
| E | 2 | 1 | 3 |
| F | 2 | 1 | 3 |
| G | 2 | 4 | 2 |
| H | 2 | 4 | 2 |
| I | 2 | 1 | 3 |
| J | 2 | 4 | 3 |
| K | 2 | 1 | 3 |
| L | 2 | 5 to 1 | 3 |
| M | 2 | 4 | 3 |
| N | 2 | 4 | 3 |
| O | 2 | 1 | 3 |
| P | 1 | 4 | 2 |
| Q | 1 | 3 | 1 |
| R | 2 | 1 | 3 |
| S | 2 | 1 | 3 |
| T | 2 | 1 | 3 |
| U | 2 | 2 | 3 |
| V | 2 | 1 | 3 |
| W | 2 | 5 | 3 |
| X | 2 | 4 | 3 |
| Y | 2 | 5 | 3 |
| Z | 2 | 4 | 3 |

Table 6: (Accelerometer areas for alphabets)

## **CODE**

[code]

#include <SoftwareSerial.h>

SoftwareSerial BTSerial(2,3); //RX / TX

const int FLEX\_PIN1 = A0;

const int FLEX\_PIN2 = A1;

const int FLEX\_PIN3 = A2;

const int FLEX\_PIN4 = A3;

const int FLEX\_PIN5 = A4;

const int xpin = A5;

const int ypin =A6;

const float VCC1 = 5;

const float VCC2 = 5;

const float VCC3 = 5;

const float VCC4 = 5;

const float VCC5 = 5;

const float R\_DIV1 = 10000.0;

const float R\_DIV2 = 10000.0;

const float R\_DIV3 = 10000.0;

const float R\_DIV4 = 10000.0;

const float R\_DIV5 = 10000.0;

const float flatResistance1 = 25000.0;

const float flatResistance2 = 25000.0;

const float flatResistance3 = 25000.0;

const float flatResistance4 = 25000.0;

const float flatResistance5 = 25000.0;

const float bendResistance1 = 100000.0;

const float bendResistance2 = 100000.0;

const float bendResistance3 = 100000.0;

const float bendResistance4 = 100000.0;

const float bendResistance5 = 100000.0;

void setup() {

Serial.begin(9600);

pinMode(FLEX\_PIN1, INPUT);

pinMode(FLEX\_PIN2, INPUT);

pinMode(FLEX\_PIN3, INPUT);

pinMode(FLEX\_PIN4, INPUT);

pinMode(FLEX\_PIN5, INPUT);

pinMode(xpin, INPUT);

pinMode(ypin, INPUT);

BTSerial.begin(9600);

}

void loop() {

if (BTSerial.available())

Serial.write(BTSerial.read());

if (Serial.available())

BTSerial.write(Serial.read());

int flexADC1 = analogRead(FLEX\_PIN1);

float Vflex1 = flexADC1 \* VCC1 / 1023.0;

float Rflex1 = R\_DIV1 \* (VCC1 / Vflex1 - 1.0);

//Serial.println("Resistance: " + String(Rflex1) + " ohms");

float angle1 = map(Rflex1, flatResistance1, bendResistance1, 0, 90.0);

//Serial.println("Bend: " + String(angle1) + " degrees");

//Serial.println();

int flexADC2 = analogRead(FLEX\_PIN2);

float Vflex2 = flexADC2 \* VCC2 / 1023.0;

float Rflex2 = R\_DIV2 \* (VCC2 / Vflex2 - 1.0);

//Serial.println("Resistance: " + String(Rflex2) + " ohms");

float angle2 = map(Rflex2, flatResistance2, bendResistance2, 0, 90.0);

//Serial.println("Bend: " + String(angle2) + " degrees");

//Serial.println();

int flexADC3 = analogRead(FLEX\_PIN3);

float Vflex3 = flexADC3 \* VCC3 / 1023.0;

float Rflex3 = R\_DIV3 \* (VCC3 / Vflex3 - 1.0);

//Serial.println("Resistance: " + String(Rflex3) + " ohms");

float angle3 = map(Rflex3, flatResistance3, bendResistance3, 0, 90.0);

//Serial.println("Bend: " + String(angle3) + " degrees");

//Serial.println();

int flexADC4 = analogRead(FLEX\_PIN4);

float Vflex4 = flexADC4 \* VCC4 / 1023.0;

float Rflex4 = R\_DIV4 \* (VCC4 / Vflex4 - 1.0);

//Serial.println("Resistance: " + String(Rflex4) + " ohms");

float angle4 = map(Rflex4, flatResistance4, bendResistance4, 0, 90.0);

//Serial.println("Bend: " + String(angle4) + " degrees");

//Serial.println();

int flexADC5 = analogRead(FLEX\_PIN5);

float Vflex5 = flexADC5 \* VCC5 / 1023.0;

float Rflex5 = R\_DIV5 \* (VCC5 / Vflex5 - 1.0);

//Serial.println("Resistance: " + String(Rflex5) + " ohms");

float angle5 = map(Rflex5, flatResistance5, bendResistance5, 0, 90.0);

//Serial.println("Bend: " + String(angle5) + " degrees");

//Serial.println();

int xadc = analogRead(xpin);

int yadc = analogRead(ypin);

if(((angle1>=70)&&(angle1<=82))&&((angle2>=77)&&(angle2<=90))&&((angle3>=70)&&(angle3<=86))&&((angle4>=73)&&(angle4<=85))&&((angle5>=0)&&(angle5<=45)))

Serial.println('A');

if(((angle1>=0)&&(angle1<=10))&&((angle2>=0)&&(angle2<=10))&&((angle3>=0)&&(angle3<=12))&&((angle4>=0)&&(angle4<=10))&&((angle5>=65)&&(angle5<=80)))

Serial.println('B');

if(((angle1>=40)&&(angle1<=72))&&((angle2>=50)&&(angle2<=90))&&((angle3>=51)&&(angle3<=75))&&((angle4>=42)&&(angle4<=66))&&((angle5>=34)&&(angle5<=50)))

Serial.println('C');

if(((angle1>=50)&&(angle1<=72))&&((angle2>=45)&&(angle2<=90))&&((angle3>=35)&&(angle3<=75))&&((angle4>=0)&&(angle4<=10))&&((angle5>=45)&&(angle5<=80))&&!(((xadc>=412)&&(xadc<=418))&&((yadc>=340)&&(yadc<=360))))

Serial.println('D');

if(((angle1>=68)&&(angle1<=88))&&((angle2>=68)&&(angle2<=90))&&((angle3>=50)&&(angle3<=80))&&((angle4>=54)&&(angle4<=80))&&((angle5>=58)&&(angle5<=88)))

Serial.println('E');

if(((angle1>=0)&&(angle1<=10))&&((angle2>=0)&&(angle2<=10))&&((angle3>=0)&&(angle3<=10))&&((angle4>=15)&&(angle4<=45))&&((angle5>=34)&&(angle5<=65)))

Serial.println('F');

if(((angle1>=75)&&(angle1<=90))&&((angle2>=75)&&(angle2<=90))&&((angle3>=65)&&(angle3<=90))&&((angle4>=0)&&(angle4<=15))&&((angle5>=0)&&(angle5<=30))&&(((xadc>=400)&&(xadc<=420))&&((yadc>=340)&&(yadc<=360))))

Serial.println('G');

if(((angle1>=70)&&(angle1<=85))&&((angle2>=75)&&(angle2<=90))&&((angle3>=0)&&(angle3<=10))&&((angle4>=0)&&(angle4<=10))&&((angle5>=50)&&(angle5<=65))&&!(((xadc>=410)&&(xadc<=420))&&((yadc>=368)&&(yadc<=380))))

Serial.println('H');

if(((angle1>=0)&&(angle1<=10))&&((angle2>=50)&&(angle2<=70))&&((angle3>=50)&&(angle3<=70))&&((angle4>=50)&&(angle4<=70))&&((angle5>=50)&&(angle5<=85)&&((xadc>=410)&&(xadc<=420))&&((yadc>=330)&&(yadc<=370))))

Serial.println('I');

if(((angle1>=0)&&(angle1<=10))&&((angle2>=50)&&(angle2<=70))&&((angle3>=50)&&(angle3<=70))&&((angle4>=50)&&(angle4<=70))&&((angle5>=50)&&(angle5<=85))&&(!((xadc>=410)&&(xadc<=420))&&((yadc>=355)&&(yadc<=370))))

Serial.println('J');

if(((angle1>=60)&&(angle1<=75))&&((angle2>=60)&&(angle2<=85))&&((angle3>=0)&&(angle3<=10))&&((angle4>=0)&&(angle4<=15))&&((angle5>=30)&&(angle5<=55))&&(((xadc>=404)&&(xadc<=415))&&((yadc>=368)&&(yadc<=380))))

Serial.println('K');

if(((angle1>=75)&&(angle1<=90))&&((angle2>=75)&&(angle2<=90))&&((angle3>=70)&&(angle3<=90))&&((angle4>=0)&&(angle4<=15))&&((angle5>=0)&&(angle5<=30))&&(((xadc>=390)&&(xadc<=405))&&((yadc>=360)&&(yadc<=380)))&&!((xadc>=270)&&(xadc<=300))&&((yadc>=360)&&(yadc<=390)))

Serial.println('L');

if(((angle1>=40)&&(angle1<=61))&&((angle2>=72)&&(angle2<=84))&&((angle3>=45)&&(angle3<=65))&&((angle4>=62)&&(angle4<=75))&&((angle5>=65)&&(angle5<=86)))

Serial.println('M');

if(((angle1>=54)&&(angle1<=70))&&((angle2>=50)&&(angle2<=61))&&((angle3>=48)&&(angle3<=66))&&((angle4>=60)&&(angle4<=76))&&((angle5>=50)&&(angle5<=65))&&(((xadc>=400)&&(xadc<=435))&&((yadc>=350)&&(yadc<=390))))

Serial.println('N');

if(((angle1>=68)&&(angle1<=88))&&((angle2>=68)&&(angle2<=90))&&((angle3>=50)&&(angle3<=80))&&((angle4>=54)&&(angle4<=80))&&((angle5>=0)&&(angle5<=30)))

Serial.println('O');

if(((angle1>=60)&&(angle1<=75))&&((angle2>=60)&&(angle2<=85))&&((angle3>=0)&&(angle3<=10))&&((angle4>=0)&&(angle4<=15))&&((angle5>=30)&&(angle5<=55))&&(((xadc>=270)&&(xadc<=290))&&((yadc>=360)&&(yadc<=380))))

Serial.println('P');

if(((angle1>=75)&&(angle1<=90))&&((angle2>=75)&&(angle2<=90))&&((angle3>=65)&&(angle3<=90))&&((angle4>=0)&&(angle4<=15))&&((angle5>=0)&&(angle5<=30))&&(((xadc>=270)&&(xadc<=300))&&((yadc>=360)&&(yadc<=390))))

Serial.println('Q');

if(((angle1>=40)&&(angle1<=72))&&((angle2>=45)&&(angle2<=90))&&((angle3>=20)&&(angle3<=45))&&((angle4>=0)&&(angle4<=10))&&((angle5>=45)&&(angle5<=80))&&(((xadc>=412)&&(xadc<=418))&&((yadc>=340)&&(yadc<=360))))

Serial.println('R');

if(((angle1>=70)&&(angle1<=90))&&((angle2>=80)&&(angle2<=90))&&((angle3>=80)&&(angle3<=90))&&((angle4>=80)&&(angle4<=90))&&((angle5>=60)&&(angle5<=80)))

Serial.println('S');

if(((angle1>=40)&&(angle1<=61))&&((angle2>=72)&&(angle2<=84))&&((angle3>=45)&&(angle3<=65))&&((angle4>=44)&&(angle4<=63))&&((angle5>=65)&&(angle5<=86))&&(digitalRead(6)==HIGH))

Serial.println('T');

if(((angle1>=70)&&(angle1<=90))&&((angle2>=80)&&(angle2<=90))&&((angle3>=0)&&(angle3<=10))&&((angle4>=0)&&(angle4<=10))&&((angle5>=60)&&(angle5<=80)))

Serial.println('U');

if(((angle1>=70)&&(angle1<=90))&&((angle2>=80)&&(angle2<=90))&&((angle3>=0)&&(angle3<=10))&&((angle4>=0)&&(angle4<=10))&&((angle5>=60)&&(angle5<=80))&&(digitalRead(6)==HIGH))

Serial.println('V');

if(((angle1>=70)&&(angle1<=90))&&((angle2>=0)&&(angle2<=10))&&((angle3>=0)&&(angle3<=10))&&((angle4>=0)&&(angle4<=10))&&((angle5>=60)&&(angle5<=80)))

Serial.println('W');

if(((angle1>=50)&&(angle1<=72))&&((angle2>=45)&&(angle2<=90))&&((angle3>=35)&&(angle3<=75))&&((angle4>=80)&&(angle4<=89))&&((angle5>=45)&&(angle5<=80)))//&&!(((xadc>=412)&&(xadc<=418))&&((yadc>=340)&&(yadc<=360))))

Serial.println('X');

if(((angle1>=0)&&(angle1<=10))&&((angle2>=70)&&(angle2<=90))&&((angle3>=60)&&(angle3<=80))&&((angle4>=80)&&(angle4<=90))&&((angle5>=15)&&(angle5<=35)))

Serial.println('Y');

if(((angle1>=50)&&(angle1<=72))&&((angle2>=45)&&(angle2<=90))&&((angle3>=35)&&(angle3<=75))&&((angle4>=0)&&(angle4<=10))&&((angle5>=45)&&(angle5<=80))&&(((xadc>=412)&&(xadc<=418))&&((yadc>=340)&&(yadc<=360))))

Serial.println('Z');

delay (10000);

}

[/code]