Filipino Sign Language to Text Converter using K-Nearest Neighbor Algorithm

by

John Vincent G. Calites

Jean Tristan L. Reyes

Glenn Christian D. Sioson

A Design Report Submitted to the School of Electrical Engineering,
Electronics Engineering, and Computer Engineering in Partial
Fulfilment of the Requirements for the Degree

Bachelor of Science in Computer Engineering

Mapúa University

September 2020

Approval Sheet

Mapua University

School of EECE

This is to certify that I have supervised the preparation of and read the design report prepared by **John Vincent G. Calites, Jean Tristan L. Reyes, Glenn Christian D. Sioson** entitled **Filipino Sign Language to Text Converter using K-Nearest Neighbor Algorithm** and that the said report has been submitted for final examination on **September 23, 2020** by the Oral Examination Committee.

Noel B. Linsangan
Adviser

As members of the Oral Examination Committee, we certify that we have examined this design report, presented before this committee **September 23, 2020,** and hereby recommended that it be accepted in fulfillment of the design requirements for the degree in **Bachelor of Science in Computer Engineering**.

Rafael G. Maramba
Panel Member 1

Meo Vincent C. Caya
Panel Member 2

Jumelyn L. Torres
Committee Chair

This design report is hereby approved and accepted by the School of Electrical, Electronics and Computer Engineering in partial fulfillment of the requirements for the degree in **Bachelor of Science in Computer Engineering.**

Alejandro H. Ballado, Jr.
Dean, School of EECE

AKNOWLEDGEMENT

This design project would not be possible without the support of our adviser Engr. Noel Linsangan, his guidance kept our work on track from the initial phase to the finalization of the design project.

We would also like to give thanks first to our parents and family members, for being the support that was given to us.

Our Design coordinator, Engr. Jumelyn L. Torres, for being considerate and understanding to our group.

We also would like to show our appreciation towards the client for this design project which is Kaisahan ng Magulang at Anak na may Kapansanan Inc. (KAISAKA), for giving us the opportunity to be of service to the PWD community.

Lastly, to our friends and classmates who gives related information and insights towards the development of this design project.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

The Philippines has recognized Filipino Sign Language as the national sign language of the Filipino deaf and the official sign language of the government (RA 11106), this involves all the transactions involving the deaf, its usage in schools, broadcast media, and workplaces. This design project develops a glove that will translate Filipino Sign Language (FSL) to text using the K-Nearest Neighbor (KNN) algorithm. The device will only translate the alphabet instead of words. The device, while worn by the user, will do a real-time language translation through the means of capturing hand movements and gestures. The device is portable in the sense that it will only need an android phone in viewing the translated gesture while connected to the gloves through Bluetooth. After conducting several tests, it shows an accuracy of 89.63% when translating FSL to text.

Keywords: Android, Bluetooth, FSL, Gestures, Hand movements

Chapter 1

BACKGROUND AND INTRODUCTION

Hearing impairment is the decreased ability to hear a sound which can occur at any age. Hearing disability can be acquired due to some genetic factors which can make a newly born child deaf, or people at any age can become deaf due to complications, diseases, nerve damages, or injuries.

Communication plays a vital role in society; it employs the transfer of information that is essential in developing relations. People who are deaf and has other forms of hearing impairment hinders them to properly communicate. Although other forms of communication for them exists, such as sign language, it is not understood by many. Thus, the communication for those who have little to no knowledge of sign language and people with hearing disabilities must be addressed.

On October 30, 2018, the Philippines has recognized Filipino Sign Language as the national sign language of the Filipino deaf and the official sign language of the government (RA 11106), this involves all the transactions involving the deaf, its usage in schools, broadcast media, and workplaces. Since most of the schools use American Sign Language (ASL), various government offices such as the DepEd, CHED, and TESDA and all national and local government authorities involving the deaf are tasked to use the Filipino Sign Language (FSL) as the medium of instruction for education.

A. Customer

The customer for this design project is Kaisahan ng Magulang at Anak na may Kapansanan Inc. (KAISAKA), a Community Based Rehabilitation (CBR) program for children and young adults with disability and their families based in the urban poor community of Manila; and the contact person of the said organization is Ms. Jeannifer Villanueva, the Program Coordinator. The organization started as a parish outreach program in 1989, KAISAKA, Inc. was registered in the Securities and Exchange Commission (SEC) as a People's Organization (PO). On August 6, 1998, with mothers and people with disabilities themselves as the frontrunner of the program.

B. Needs

After the initial interview with Ms. Jeannifer Villanueva, some of the officers of the said organization which directly deals with the PWD's have low proficiency regarding Filipino sign language, thus they encounter difficulty when communicating with deaf people. Since they are a non-profit organization, they do not have sufficient funds to be able to hire translators all the time, it will also take time to fully learn and study Filipino Sign Language. As one of the officers said, they barely understand Filipino sign language, and most of the time they have a hard time understanding it; this is because their organization is not limited in accommodating the deaf, but a variety of PWDs. The need of the client is a device that can aid the officers in understanding deaf people. The device must also be easy to use, making it usable even by children. It must also be portable in such a way that it can be used in everyday situations.

C. Solution

As a response to the need of the client, the group intends to create a device that will help by providing a means of training officers, volunteers, or those persons with disabilities in Filipino Sign Language.

The proposed solution is a glove that will translate Filipino Sign Language (FSL) to text, which can be viewed on an android phone. The device is portable in the sense that it will only need an android phone in viewing the translated gesture. The glove, while worn by the user, will do a real-time language translation through the means of capturing hand movements and gestures. The movement and gesture of the hand will serve as the input data, these data could be gathered by providing data to the microcontroller coming from the flex sensors, gyroscope, and accelerometer. The flex sensors will be responsible for taking the bending motion of the fingers. The gyroscope and accelerometer will take the position and speed of the hand in motion. Combining these three data inputs, the interpretation of a specific gesture could be possible.

The data coming from the sensors will then be sent by the microcontroller to the connected Android phone using a Bluetooth module. In the android phone, the hand movement and gesture are interpreted through the K-nearest neighbor algorithm that predicts the output using the created mobile application. The force-sensitive resistor will be pressed for the device to start translating the sign language to text. The device will be powered by a rechargeable battery with a charging module.

In terms of how the device will be made, the group plans to design it as small as possible for the user to be able to move his hand and make gestures comfortably. The flex sensors will be placed along with the fingers. The gyro sensor is placed on the back of the hand while the force sensitive resistor will be placed at the thenar space (between the thumb and index finger). The microcontroller, Bluetooth module, and the battery along with the charging module will be placed on top of the wrist. A small and compact PCB will be designed to connect all these components.

1. Objectives

In this design project, the group intends to develop a real-time language translation of Filipino Sign Language (FSL) to text. Specifically, the group wants to achieve the following:

1. To use a flex sensor, gyroscope, and accelerometer to capture hand movements and gestures.
2. To create a portable device utilizing an Android application capable of translating Filipino Sign Language (FSL) to text.
3. To use the K-Nearest Neighbor Algorithm for classifying gestures.

2. Scopes and Delimitation

The android device only translates the alphabets in Filipino Sign Language (FSL) to text. Only the right-hand glove will be designed and built for gesture recognition. The output text will be based only on the data gathered from the glove. Facial expressions, head movement, body positions, and proximity of the hands to each other are not translatable by the device. The microcontroller will only send the data coming from the sensor to the android phone once the force sensitive resistor is pressed, this is done so that it will only send data to the android phone when the user wants to fingerspell a letter and to avoid transmitting data when it is idle. Only one android phone can be paired to the glove to view the translated texts. The android phone will be used to process the data and convert the signals to text rather than the ATmega328p microcontroller since KNN uses large amounts of resources.

D. Differentiation

A thesis published from Mapua Institute of Technology Manila: MIT School of Graduate Studies, "A heuristic decision tree algorithm for Filipino Sign Language (FSL) translator implemented using accelerometers with Bluetooth connectivity to a host PC" (Piscasio,2014) was found to have a similar problem to solve, to bridge the communication gap between signers and non-signers. There are however some differences in the implementation of the solution to the problem. The design to be implemented aims to make the device portable for it to be usable in an everyday situation as compared to the work of Piscasio that uses a PC as the output and processing device. There are also differences in the components, method, and technology used as seen in Table 1.1. Our group will use the ATmega328p microcontroller and translate Filipino Sign Language (FSL) to text and will have a rechargeable battery. Lastly, the machine learning algorithm that the group will use is the K-Nearest Neighbor algorithm, unlike the similar device which uses a Heuristic Decision Tree.

Table 1.1: Differentiation from Previous Work

	Design Solution	**Nearest Similarity**
Technology	**Microcontroller:** ATmega382p **Sensors:** Flex Sensors Force Sensitive Resistors MPU6050 Sensor **Algorithm:** K-Nearest Neighbor	**Microcontroller:** ATmega2560 **Sensors:** Accelerometer **Algorithm** Heuristic Decision Tree
Functionality	Converts Filipino Sign Language to Text	Converts Filipino Sign Language to Text
Features	Portable Rechargeable Uses Android phone	Uses PC

E. Benefits

The product is made specifically to benefit the officers of the organization who have little knowledge about Filipino Sign Language (FSL) and have a hard time understanding it. This product will also benefit the deaf people of whom the organization helps, since they can now train by themselves and their knowledge will be reinforced by showing them if their gesture is correct. The product is being designed and built as a compact and portable device, making it easier to be carried around, thus it can be used anywhere when needed. The device can help those who are beginners in terms of FSL for them to learn the basics of sign language which is fingerspelling.

Chapter 2

REVIEW OF RELATED LITERATURE AND STUDIES

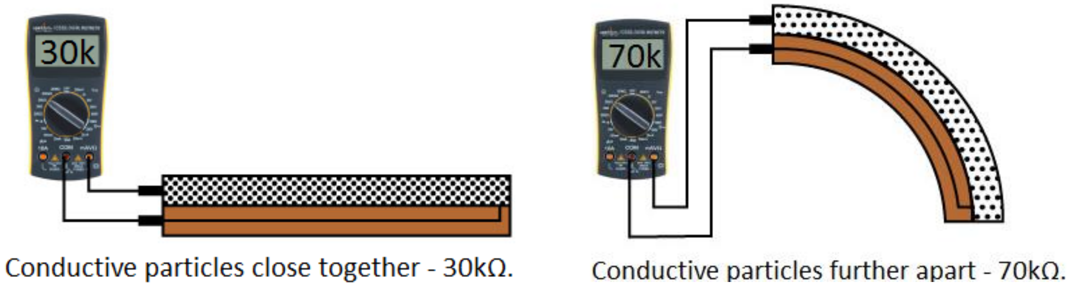
This chapter involves related studies and developments of the design of the Filipino Sign Language (FSL) to text converter. The researchers will give an overview to its readers how this study's design works and develop a new concept.

Technology for Capturing Gestures

Language translation can be done through the use of electromagnetic sensors that have direct contact with the skin along with accelerometers, gyroscopes, and vision through means of a supervised machine learning algorithm requiring the user up to 30 minutes of training per gesture. The algorithm consists of different type of machine learning algorithm which includes Naïve Bayes, Nearest Neighbor, decision trees, and vector machines which enables the comparison for accuracy of 40 common gestures. An important consideration in creating a communication device is by following the safety standards. To bring a device as close as a commercial product, it must follow the safety standard like prevention of shorting wires that might cause injuries. Also, electrocution must be prevented through means of having a total voltage lower than 25V which was referred to as the lowest lethal voltage by a research paper completed at the City University of New York.

Flex Sensor

A flex sensor is a variable resistor, its resistance increases as the component bends. One side of the Flex Sensor is made out of polymer ink which has conductive particles. When the flex sensor is straight, it gives the polymer ink resistance of approximately 30kΩ, when bent, it gives around 50kΩ~70kΩ depending on the degrees it is bent.



Conductive particles close together - 30kΩ.

Conductive particles further apart - 70kΩ.

Figure 2.1: Comparison of Flex Sensor resistance

A voltage divider will be required in incorporating the sensor to a project, it must be connected in the Arduino's A0 port and the other end at 5V. When developing the program, the straight and bend resistance must be declared. And then calculate for the resistance using various formulas.

The mechanism for Filipino Sign Language (FSL)

Sign Language is not an international language, but those who use sign language may understand fellow sign language users quickly because there are universal features in sign languages as stated by the World Federation of the Deaf. Filipino Sign Language is used by deaf people in the Philippines, this is distinct in comparison to the American Sign language (ASL) because Filipino sign language poses unique structural features in terms of its phonology, morphology, syntax, and discourse.

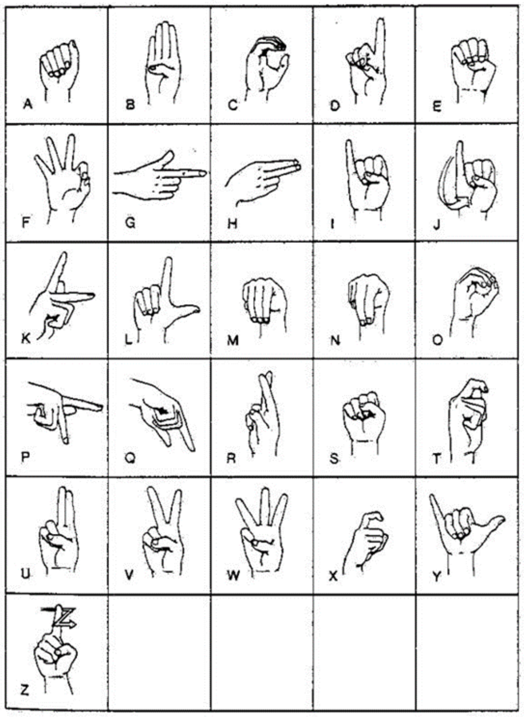


Figure 2.2: Filipino Sign language alphabet

MPU6050 Gyro Sensor

The MPU6050 is an 8-pin with 6 axis chip that combines the accelerometer and a gyro. To use the Gyro Sensor, pin 18 (GND) must be connected to the ground, 3.3V must be connected to pin 13 (VDD). Pin 23 (SCL) and Pin 24 (SDA) of the MPU6050 must be connected to the A4 pin and A5 pin of the microcontroller respectively. The data pin is directly connected to the digital PINs 10-13. To program using the MPU6050 Sensor, a library must be imported from GitHub and install it in the IDE. This library supports reading and displaying gyro and accelerometer readings. The Gyro sensor uses the earth's gravity to determine the x-axis, y-axis, and z-axis position of the sensor while the accelerometer detects based on the movement.

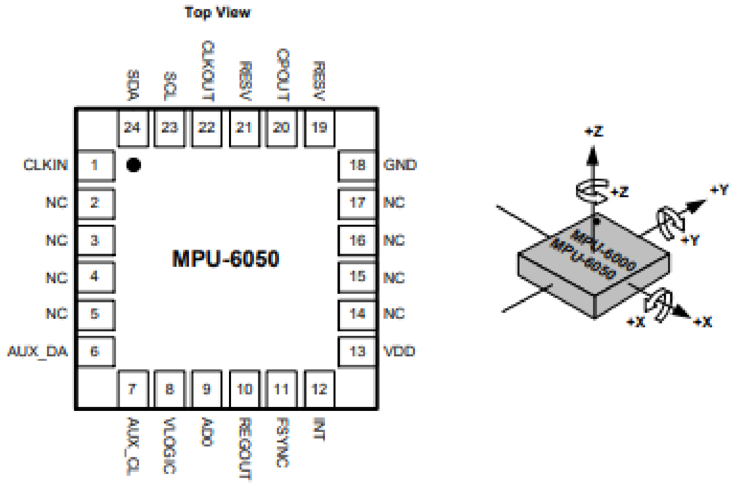


Figure 2.3: MPU6050 Pin diagram and orientation

Force Sensitive Resistors

FSRs are simple and low-cost sensors that allow the detection of pressure and weight. It is made of 2 layers which are the flexible substrate with printed semiconductor and a flexible substrate with printed interdigitating electrodes which are separated by a spacer. The resistance of the FSR changes as pressure is applied. When no pressure is applied, the FSR acts as a resistor, as pressure is applied the resistance decreases. One end must be connected to a 5V supply while the other end is connected to the analog input.

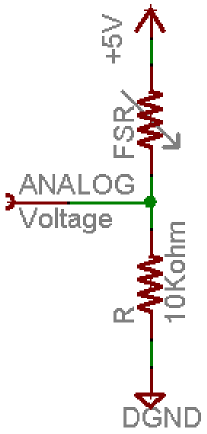


Figure 2.4: FSR Connection

The voltage of the Force Sensitive Resistor can be computed as $v_o = v_{cc}\left(\frac{R}{R+FSR}\right)$, whereas V_o is the output voltage, V_{cc} being equal to 5V, FSR being the resistance of the force-sensitive resistor when 0 pressure is applied and lastly R being the value of the pull-up resistor.

Interfacing an Android Phone

Interfacing an Android phone with Arduino via Bluetooth is possible through the use of a Bluetooth module. Connect first the Pin 0 (RX) to TX, TX(Pin 1) to RX, VCC, and ground accordingly. Turning on the microcontroller also powers up the HC 05/06 Bluetooth module, the android phone needs to be paired via Bluetooth by entering the default password which can either be 1234 or 0000. After a successful connection, data transfers via Bluetooth can now be used.

K-Nearest Neighbor Algorithm

It is a supervised machine learning algorithm that is dependent on labeled data inputs to learn and will produce an output based on the newly given input unlabeled data. This is commonly used to solve classification and regression problems. This algorithm assumes that similar things exist in proximity.

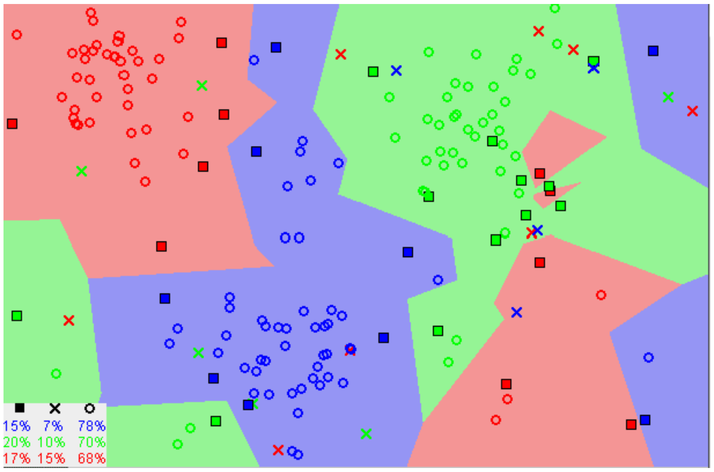


Figure 2.5: Similar datapoints existing close to each other

Figure 2.5 shows that most of the time, similar data points are located close enough from each other. The KNN algorithm bases on this assumption being true, it captures the idea of similarity. Several steps are required in using the K-Nearest Neighbor algorithm, these are as follows;

1. Load the data
2. Initialize the value of K to the number of chosen neighbors
3. For each sample data, compare the distance between the query example and current example then add the distance along with its index to an ordered collection

4. Sort the ordered collection of distances and indices in ascending order
5. Pick the first K entry
6. Get the labels of the selected entry
7. Return the mean or the mode of the K labels if it is a regression (mean) or classification (mode)

K-Nearest Neighbor Data Pre-Processing

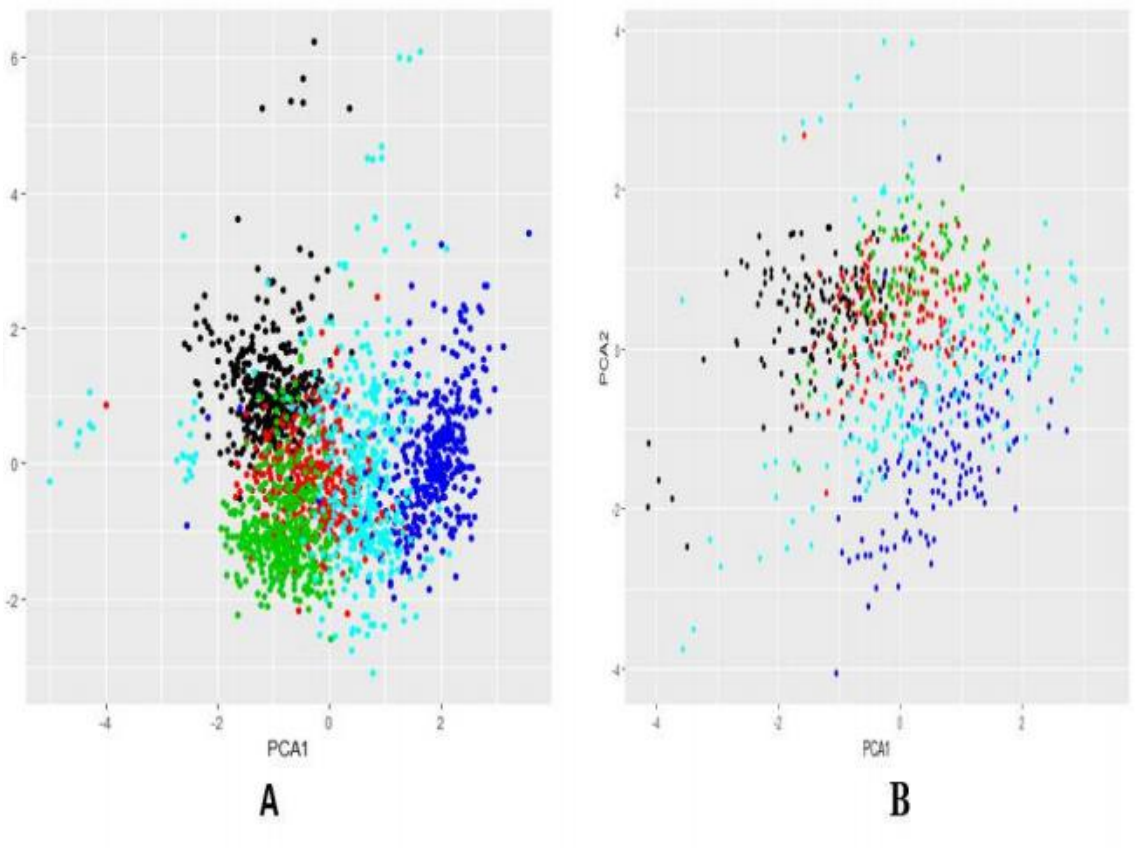


Figure 2.6: Sign Language Database

When implementing a complex prototype selection algorithm with regards to a large scale of data, the execution time needed to accomplish the task is long due to the repeated iterations in the training set. The gloves acquire the data from the flexion of the sensors when performing different sign languages, thus a large number of data are stored, which needs to be pre-processed. The data which are like the noise to the database are then removed. For data acquisition, the gesture must be done for several minutes for one minute in each position. These data are stored in the new matrix **T**, of **m** x **n** order wherein the value of m is the number of samples and the variable n is the number of attributes that represent the data.

Figure 2.6 illustrates in part A the matrix T and in part B the matrix U, colors: black (number 1), red (number 2), green (number 3), blue (number 4), and cyan (number 5).

Chapter 3

DESIGN PROCEDURES

In compliance with the client's needs, the researchers have designed the Filipino Sign Language (FSL) to text converter. In this chapter, the stages of designing the system that will comply with the needs of the client are discussed.

A. Design Process Flow

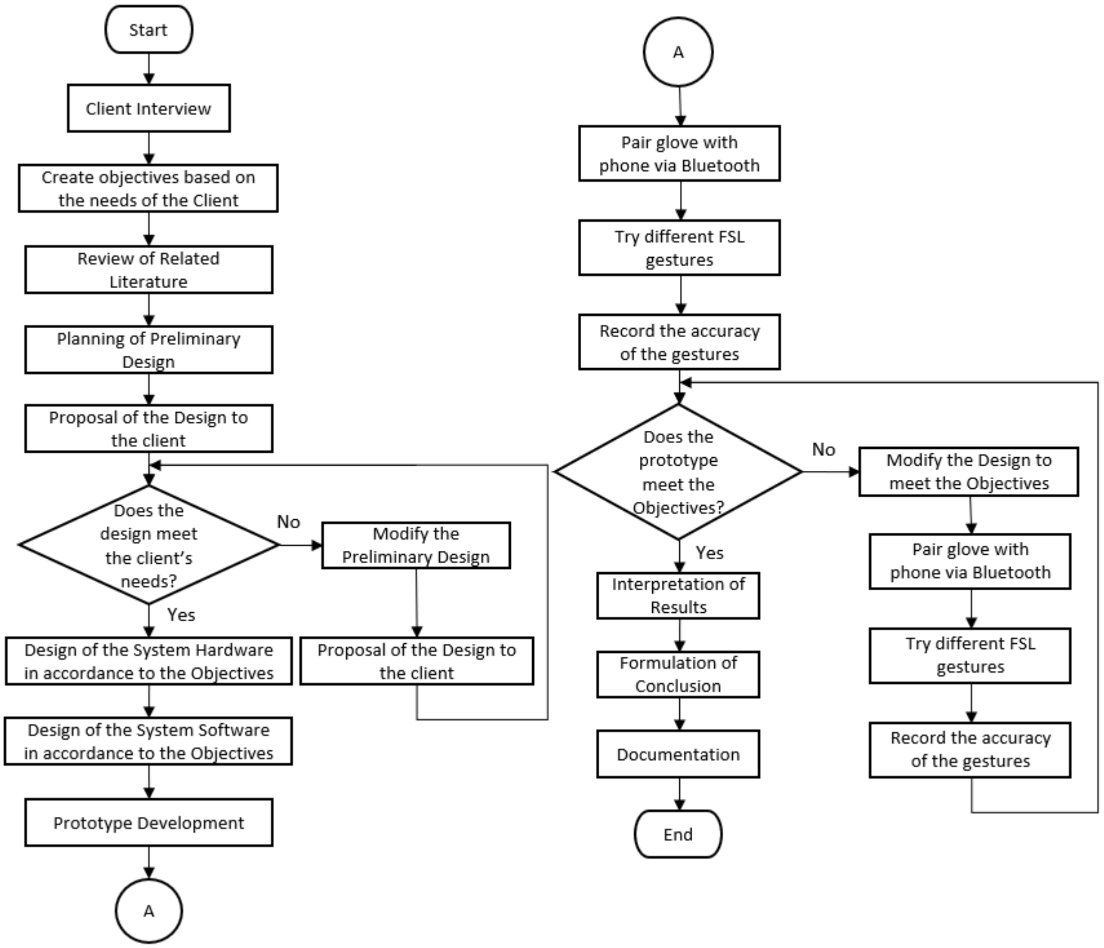

Start
Client Interview
Create objectives based on the needs of the Client
Review of Related Literature
Planning of Preliminary Design
Proposal of the Design to the client
Does the design meet the client's needs?
No
Modify the Preliminary Design
Proposal of the Design to the client
Yes
Design of the System Hardware in accordance to the Objectives
Design of the System Software in accordance to the Objectives
Prototype Development
A
A
Pair glove with phone via Bluetooth
Try different FSL gestures
Record the accuracy of the gestures
Does the prototype meet the Objectives?
No
Modify the Design to meet the Objectives
Pair glove with phone via Bluetooth
Try different FSL gestures
Record the accuracy of the gestures
Yes
Interpretation of Results
Formulation of Conclusion
Documentation
End

Figure 3.1: Design Process Flow

Figure 3.1 illustrates the flow of design development. First, the client was interviewed to know their needs with regard to the existing problems of persons with physical disabilities. Then the objectives were established under the needs of the client. The researchers then read documents, journals, and previous studies to have an idea of how the problem will be solved. After formulating concepts for the solution, the next step is to meet and discuss with the clients if the proposed solution meets the client's needs, if not, the design is modified and is discussed again with the client. If the design is approved by the client, the next step is to design the hardware part of the design which includes the listing of components needed and creating the schematic diagram. Next is to design the software part of the system, this will describe the logical flow of the system using programming. Then the hardware design will be implemented to form the actual hardware as well as the software to be uploaded into the microcontroller to build the prototype of the system. Testing and data gathering will be conducted to see if the prototype functions in accordance with the objectives. The glove and the app are tested with different gestures, and the accuracy for each gesture is recorded. If it meets the objectives, the results are interpreted and conclusions are formed, if not then the current prototype is modified and tested until the objectives are met. The last step of the design process flow is the documentation of the project.

B. Hardware Development

Step-by-Step Procedure of Hardware Development

1. Identify the needed components to be used; the major materials are:
    a. ATmega328p – The microcontroller that will be used.

b. HC05 Bluetooth Module – Used in transmitting converted FSL to an android phone.

c. Flex Sensor – Used to measure the flexion of the fingers.

d. FSR 400 – Used to identify whether the device should translate or not.

e. MPU6050 – Used to measure the orientation of the hand and the speed at which the hand is moving.

f. TP4056 Module – Used to charge the batteries.

g. Gloves – Used to hold the components.

h. 14500 Li-Ion Battery – Used to power the device.

2. Construct a block diagram that will show the hardware design flow.
3. Create a schematic diagram for the FSL Translator Circuit.

a. Conceptual Framework

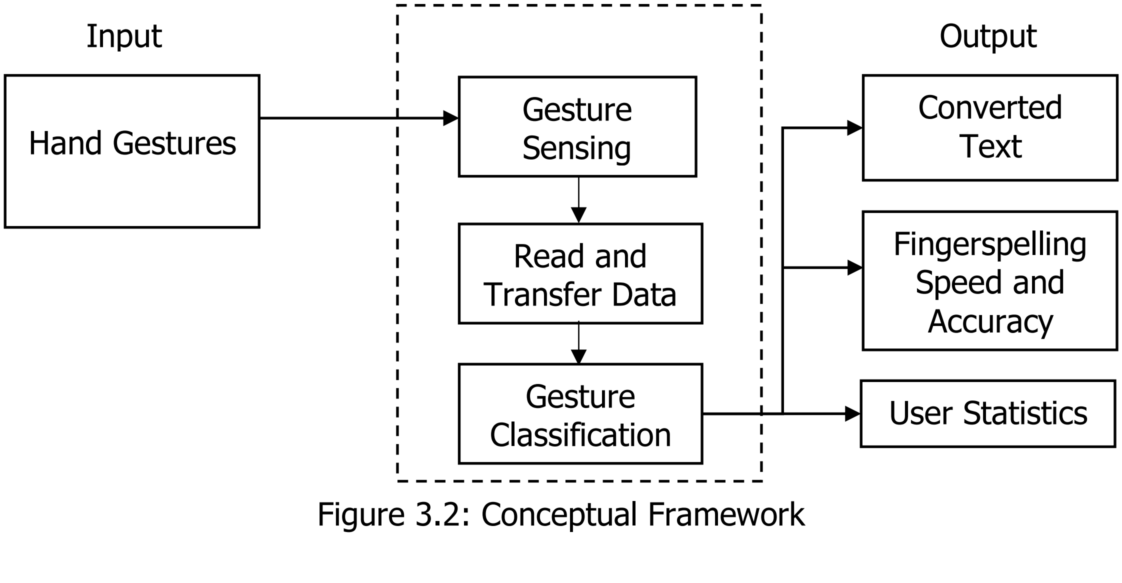


Figure 3.2: Conceptual Framework

Figure 3.2 shows the conceptual framework of the system. The input will come from the hand gestures of the user that corresponds to the Filipino Sign Language (FSL). There will be sensors providing the data from the hand gestures. The data will be read by the microcontroller and will be sent to the android phone via Bluetooth connection. It will only be sent to the android application once the FSR is pressed. In the android phone, the data will be processed using the Machine Learning Model that is trained using the K-Nearest Neighbor Algorithm which finds the most frequent label for a given value of K on the trained data. Lastly, the label's equivalent text will be displayed on the android phone. The user can also see how fast and accurate his/her fingerspelling is after fingerspelling a sentence, this contains the words translated per minute and accuracy. The user can also view their progress through a line graph displaying their records which includes the games played, speed record, average speed, and accuracy.

b. Block Diagram

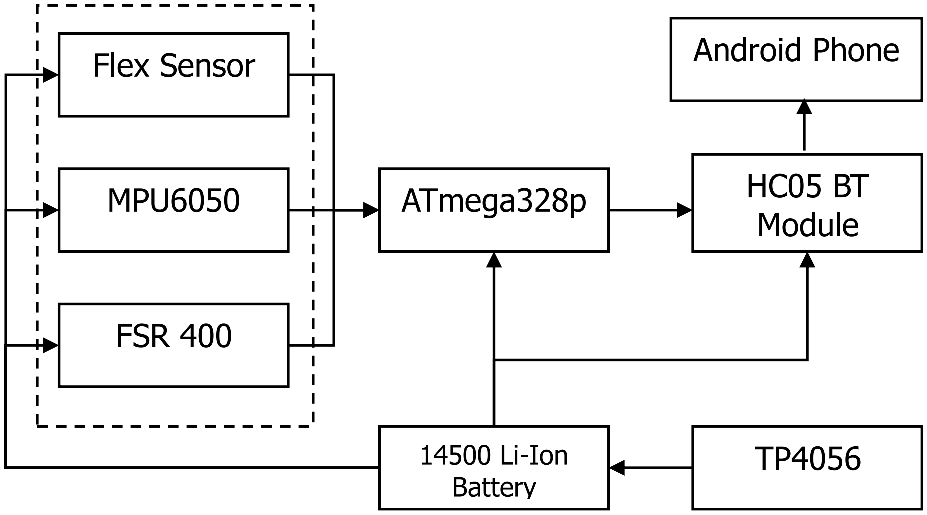


Figure 3.3: Block Diagram for the Design Project

Figure 3.3 illustrates the interaction among the components of the device. The microcontroller ATmega328p will serve as the controller for the system in which it controls and reads data from the different components. The ATmega328p reads the resistance coming from the flex sensor, the acceleration, and position from the MPU6050, and if the device is ready to translate by pressing the FSR. The microcontroller then sends it to the android phone through means of Bluetooth connection. Then, the smartphone uses the data gathered by the sensors to interpret the gesture through a given set of algorithms and translates it accordingly. The translated text will be displayed on the smartphone. The power will come from the battery that is rechargeable through means of the TP4056.

c. Schematic Diagram

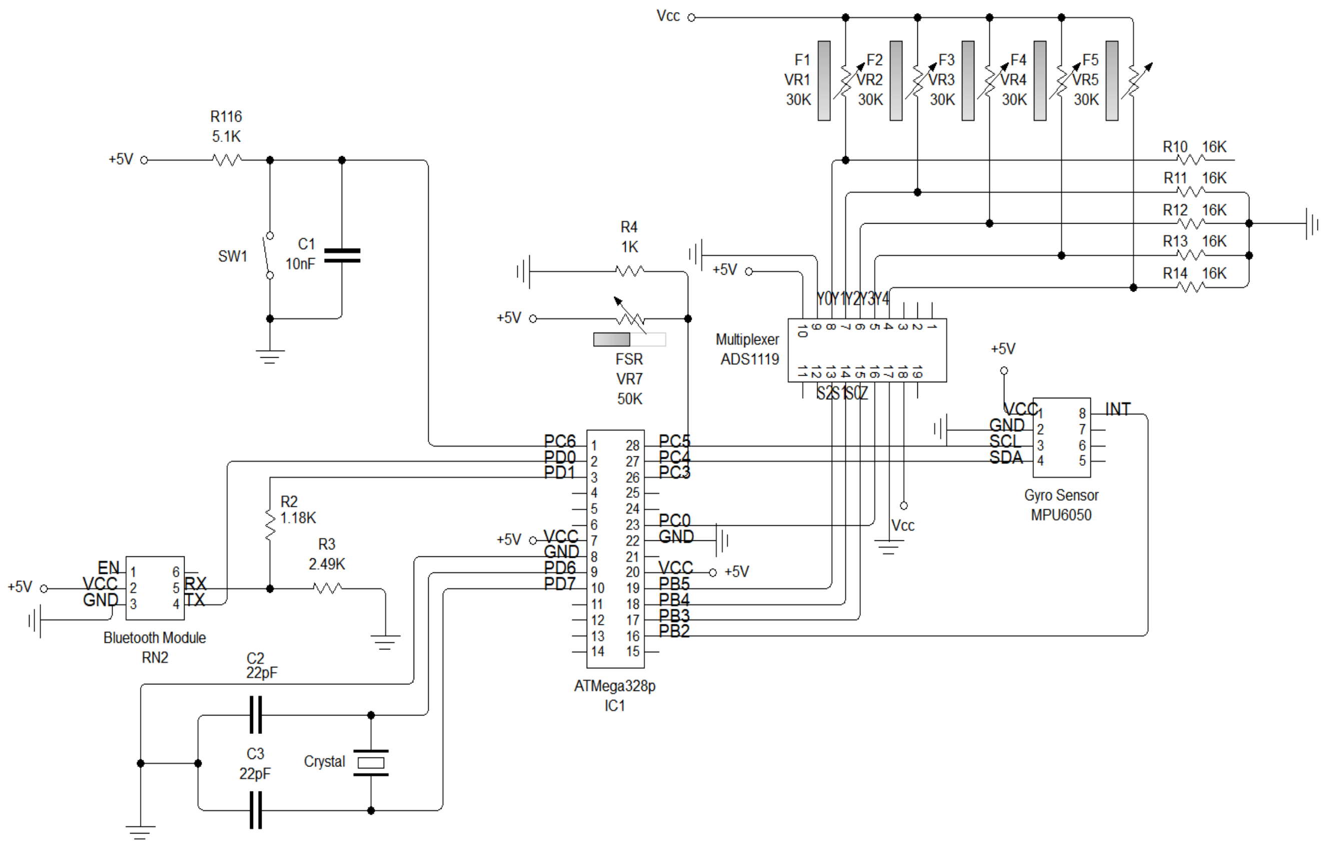


Figure 3.4.1: Schematic Diagram for the Main Module

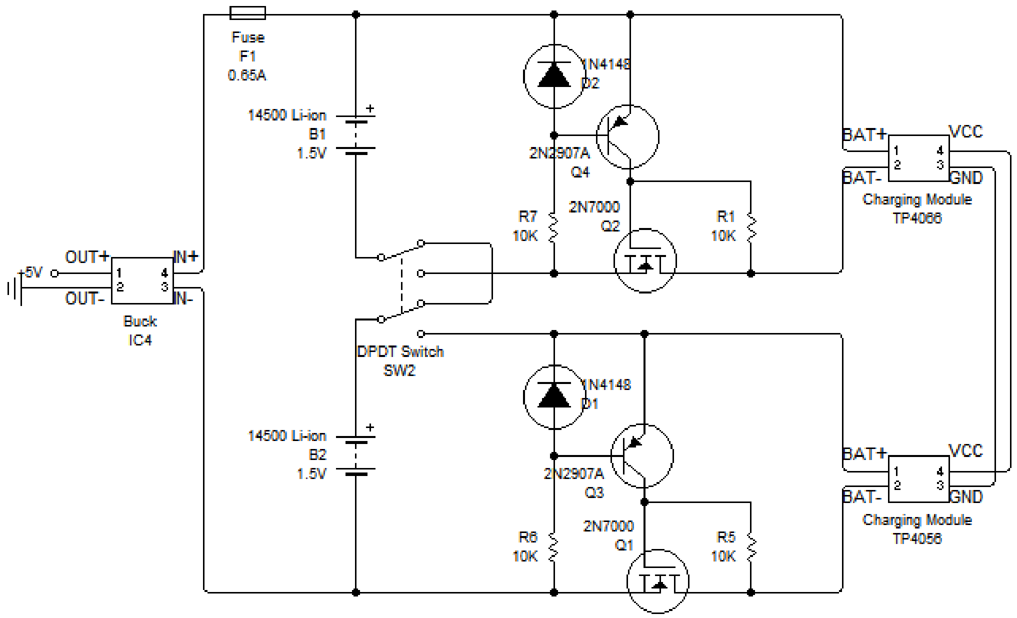


Figure 3.4.2: Schematic Diagram for the Battery Module

The schematic diagram is illustrated in figure 3.4.1 and figure 3.4.2. The microcontroller is powered by 2 pieces of 3.7V Li-ion battery which uses the buck converter to step down the output to 5V, it is then connected at the DPDT switch. The FSR400 is connected to pin 26 of the microcontroller, once pressed the FSR lowers its resistance, thus the FSR enables the sign language translation when it is pressed. The pin 7 of the microcontroller which is the VCC is connected to the output voltage of the buck converter while 8 and 22 of the microcontroller are connected to the ground.

The microcontroller is short of analog input pins thus an extender is used, this multiplexer is used for the connection of the flex sensors. The RX and TX pins of the HC05 Bluetooth module are connected in pins 2 and 3 of the ATMega328p which are the RXD and the TXD pins respectively, this is used for communication between the ATmega328p board and the other devices serially. Pins 9 and 10 of the ATmega328P microcontroller is also connected to a 16Mhz Crystal Oscillator with both ends being connected to a 22pF capacitor.

The data signal pin of the pin 24 of the MPU6050 is the SDA, this is connected to the pin 27 of the microcontroller which permits the data to be transferred from the gyro sensor to the microcontroller, the pin 28 of the microcontroller is connected to the SCL or the clock of the MPU6050.

A TP4056 is connected to the battery which enables the lithium-ion battery to charge. Although, in the case of inserting the battery incorrectly, it will cause reverse polarity, and will end up damaging the circuit. To prevent this, a circuit was created for reverse polarity protection. A MOSFET was placed between the battery and charging module. When off, the MOSFET does not let any current through, effectively opening the circuit like a reverse-polarized diode. When ON, it behaves more like a resistor with very low resistance, causing a voltage drop that decreases as the cell becomes full and the current slows down. A resistor was placed from the source to the gate to make sure the MOSFET is off by default and a transistor is added to override the resistor and flip the voltage on the gate when the battery is correctly positioned. The values for the limiting resistors are computed using the formula $V = IR$, whereas a two 3.7V batteries in series have a current of I=0.0011A and a voltage of V=3.4V, the value for R can be computed. This resulted in the formula $R = \frac{Vt-Vi}{I}$ wherein V_T is the total input voltage which is equal to 7.4V and V_o is the output voltage that is equal to 3.4V, by substituting the values will result in an approximate value of 3650 Ω. The fuse rating that will be used for the circuit is 0.65A, from the formula Fuse Rating = Wattage / Voltage x 1.25, where the values for the power and voltage are based on the datasheet of 14500 Li-ion battery.

d. Isometric Model

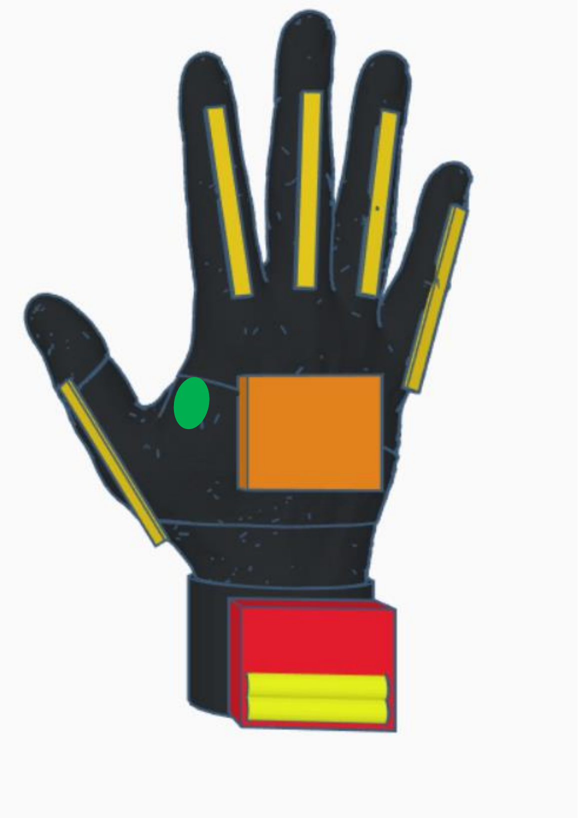


Figure 3.5: Isometric Model of the Prototype

Figure 3.5 illustrates the isometric model of the design. The design will be compact for the user to freely move the hand and make gestures with ease. The components will be placed at the top of the wrist for simple wiring, this includes the ATmega328p, TP4056 as well as the battery that will be contained in a compact case that is placed along the top of the wrist. The flex sensors will be placed on top of the fingers. The force-sensitive resistor is placed on the thenar space, while the gyro sensor is placed on the back of the hand. To provide safety for the user, the case for the ATmega328P together with the other components will both be properly insulated.

C. Software Development

The device requires a program that will take the input data coming from the flex sensors, accelerometer, and gyroscope, and output the equivalent text in the android application. For this to happen, the group needs to work on the program for the Microcontroller, the Android Application, and the Training Application.

Step-by-Step Procedure of Software Development

1. Identify the needs of the program that must be satisfied.
2. Plan the program's logic and create a system flowchart that contains the steps of the program.
3. Prepare the software to be used in building the program such as IDE and programming language to be used.
4. Construct the code of the program.
5. Test the program for syntax and logical errors.
6. Deploy the program to the device.

ATmega328p Microcontroller Programming

Needs of the Program that must be satisfied

The program must be able to let the ATmega328p microcontroller to receive data from the flex sensors, accelerometer, gyroscope, and force resistive resistor and send it to the Android phone via Bluetooth.

ATmega328p Flowchart

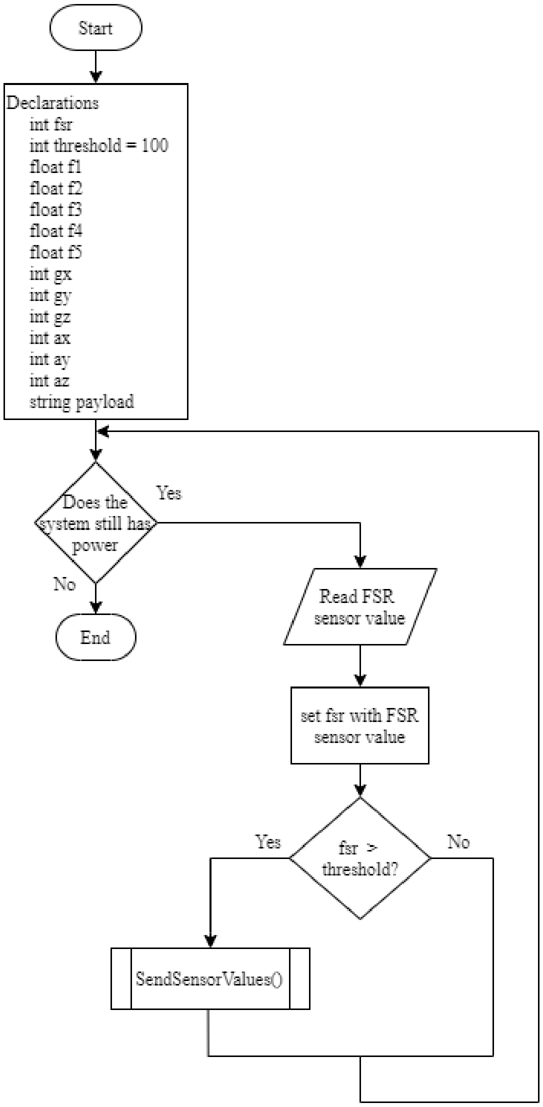
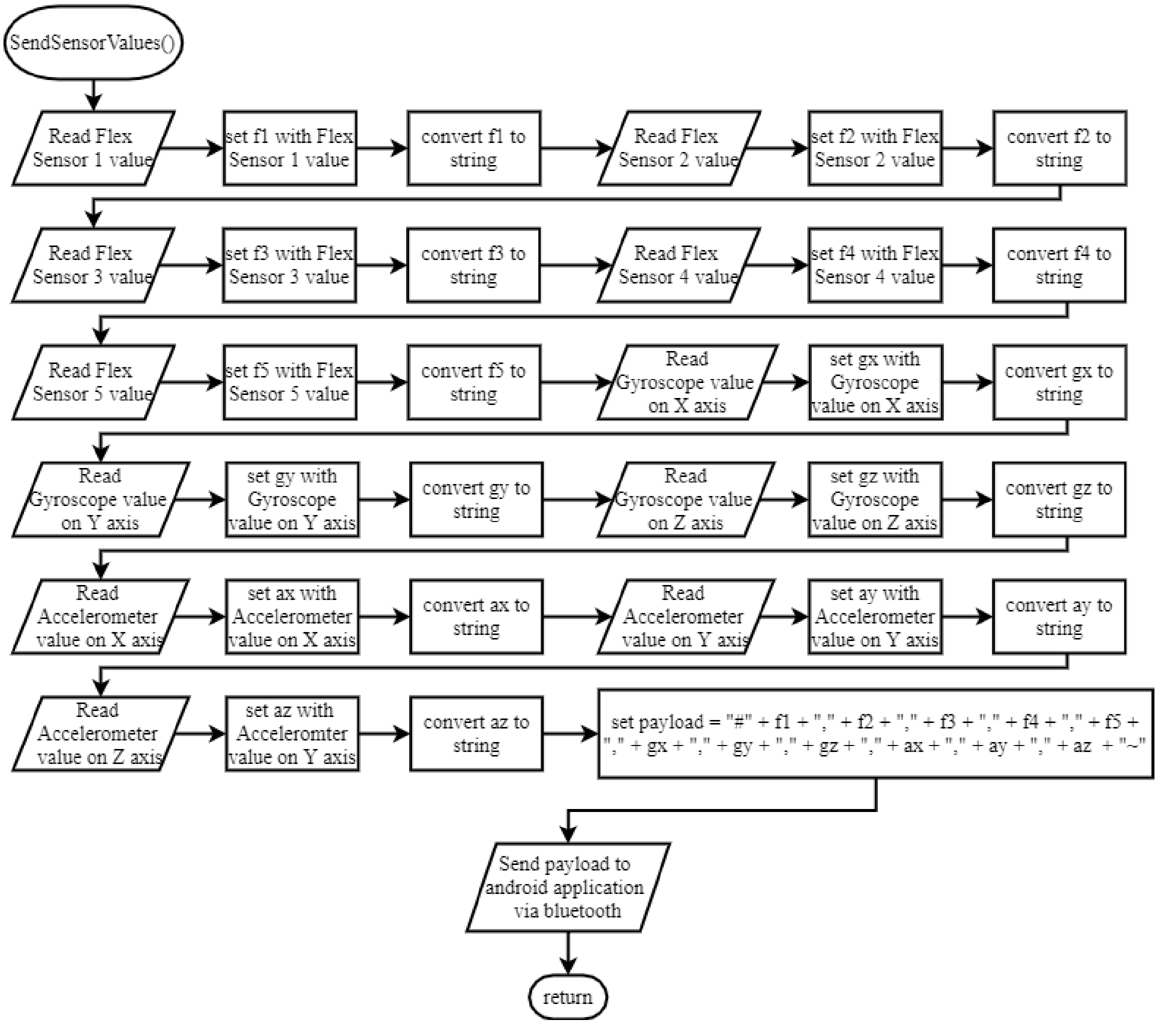


Figure 3.6: ATmega328p Program Flowchart

Figure 3.6 shows the flow of the program that is being implemented on the microcontroller. All the variables to be used are declared at the start of the program. From here, the program will loop until the system gets powered down. While the system has power, it will read the Force Sensitive Resistor (FSR) Sensor value. If the FSR value is greater than the given threshold, the module in Figure 3.7 will be used.



SendSensorValues()
Read Flex Sensor 1 value
set f1 with Flex Sensor 1 value
convert f1 to string
Read Flex Sensor 2 value
set f2 with Flex Sensor 2 value
convert f2 to string
Read Flex Sensor 3 value
set f3 with Flex Sensor 3 value
convert f3 to string
Read Flex Sensor 4 value
set f4 with Flex Sensor 4 value
convert f4 to string
Read Flex Sensor 5 value
set f5 with Flex Sensor 5 value
convert f5 to string
Read Gyroscope value on X axis
set gx with Gyroscope value on X axis
convert gx to string
Read Gyroscope value on Y axis
set gy with Gyroscope value on Y axis
convert gy to string
Read Gyroscope value on Z axis
set gz with Gyroscope value on Z axis
convert gz to string
Read Accelerometer value on X axis
set ax with Accelerometer value on X axis
convert ax to string
Read Accelerometer value on Y axis
set ay with Accelerometer value on Y axis
convert ay to string
Read Accelerometer value on Z axis
set az with Accelerometer value on Y axis
convert az to string
set payload = "#" + f1 + "," + f2 + "," + f3 + "," + f4 + "," + f5 + "," + gx + "," + gy + "," + gz + "," + ax + "," + ay + "," + az + "~"
Send payload to android application via bluetooth
return

Figure 3.7: SendSensorValue module flowchart

Figure 3.7 shows the module to be used when the FSR reached the threshold value. This module reads all the value of Flex Sensors, Gyroscope, and Accelerometer, assign each sensor readings to variables, and converts all the variables to a string. The variables are then concatenated with delimiters to produce a specific format that will be able to parse by the android application.

Preparing the Software to be used

In writing and uploading the code to the microcontroller, the open-source Arduino Software (IDE) will be used. It is an IDE especially made for Arduino and it uses C/C++ as its programming language.

Android Application Development

Needs of the Program that must be satisfied

The main purpose of developing an Android application is to display the equivalent text of a corresponding gesture. To implement this, the application should be able to let the Android phone to connect and receive sensor data from the microcontroller, classify the input data using K-Nearest Neighbor Algorithm Model, and display the output letter as text on the screen of the android phone. It must also provide a simple and friendly user interface, for easier use.

Android Application Flowchart

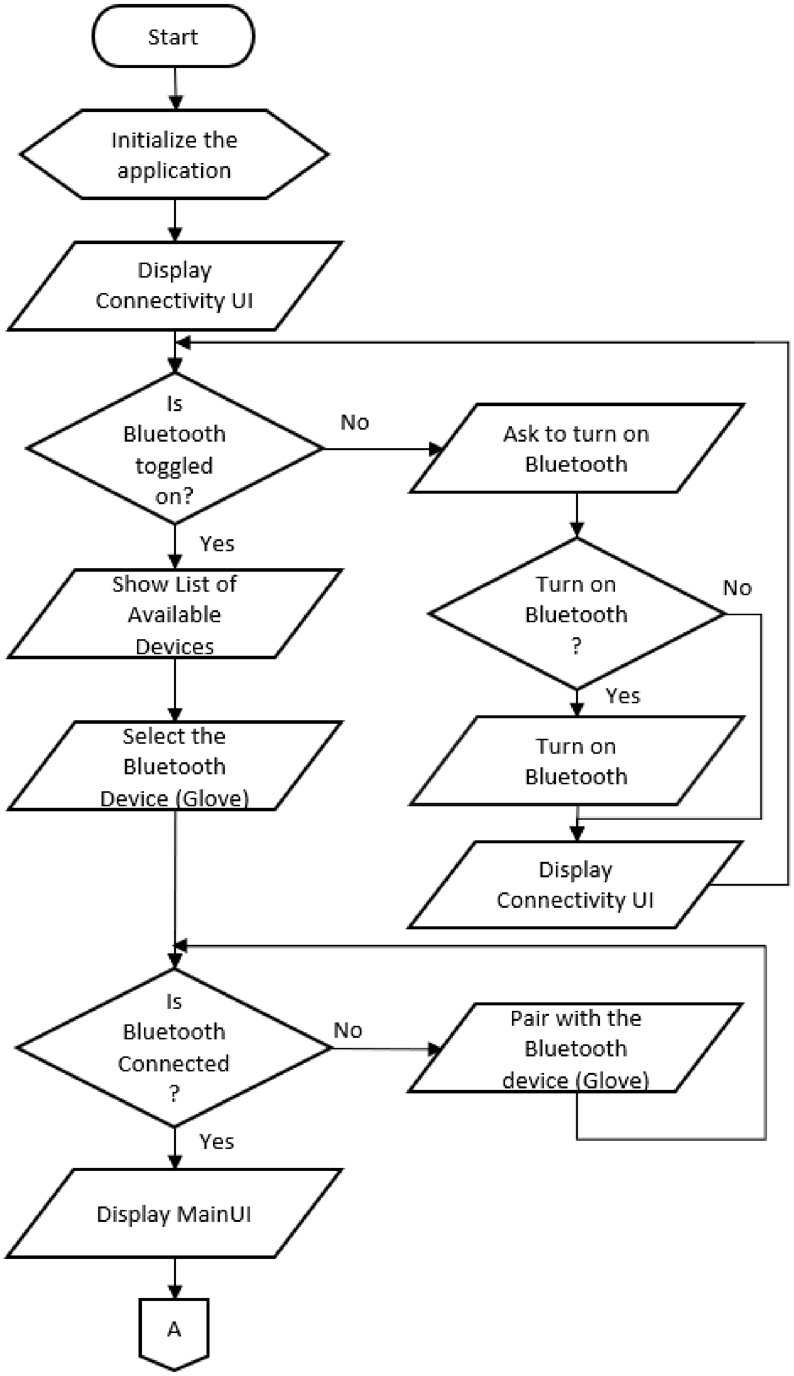


Figure 3.8.1: Android Development Flowchart

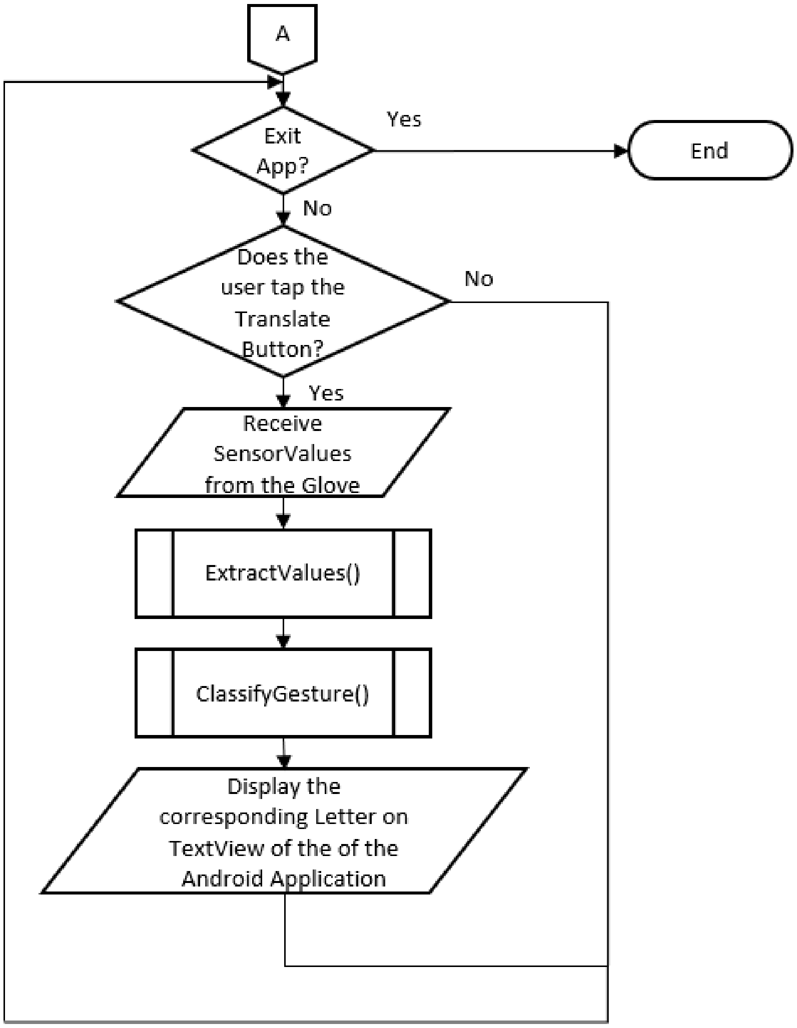

A
Exit App?
Yes
End
No
Does the user tap the Translate Button?
No
Yes
Receive SensorValues from the Glove
ExtractValues()
ClassifyGesture()
Display the corresponding Letter on TextView of the of the Android Application

Figure 3.8.2: Android Development Flowchart

Figure 3.8.1, 3.8.2, and 3.8.3 illustrates the flow of the program of the android application. As the application starts, the application will initialize and show the user interface for connectivity. A toggle switch is available for activating the phone’s Bluetooth. Once activated, the application will show a list of available devices to connect. The user must choose the glove as the device to connect to. Once connected, the main user interface will appear. The main user interface contains a toggle switch for the translate function and a textbox for viewing of text.

While the toggle switch is on, the input data from the microcontroller is read and will be used for classifying the hand gestures' equivalent letter. The letters will be displayed as text and can be viewed in the textbox, one letter at a time. The program will end when the user exits from the application.

Figure 3.9 illustrates the function to classify gestures. Variables to be used within the function are declared at the start of the function. The content of the totalTrainedData variable will be initialized by the total number of trained data of our data set. Then get the computed distance from our testInput to each trainedInput in the data set and store it in the list of distance, this is computed using equation 3.1.

$$distance = \sqrt{\sum_{i=1}^{k}(x_i - y_i)^2} \qquad (3.1)$$

After getting all the distances, sort the list of distances in increasing order. In this way, we can get the K-nearest distances from the list of distance and get all their trainedLabels. All the trainedLabels will be stored in the list of targets. Now that we have the list of trainedLabels within the range of K, we can get the most common trainedLabel in the list and set it as the outputLabel of the testInput.

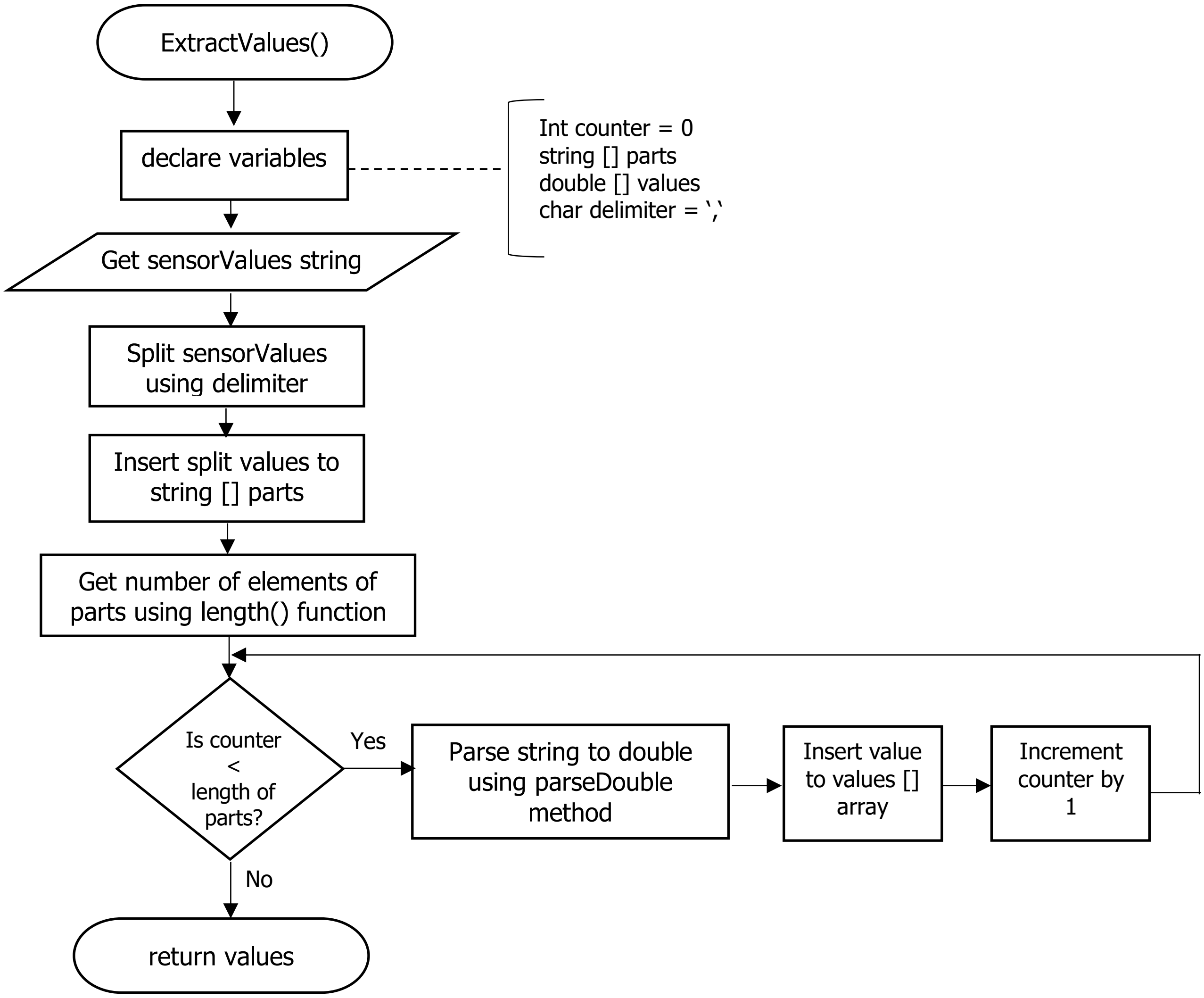

ExtractValues()
declare variables
Int counter = 0
string [] parts
double [] values
char delimiter = ‘,’
Get sensorValues string
Split sensorValues using delimiter
Insert split values to string [] parts
Get number of elements of parts using length() function
Is counter < length of parts?
Yes
Parse string to double using parseDouble method
Insert value to values [] array
Increment counter by 1
No
return values

Figure 3.8.3: Extracting Sensor Values from a single string

To extract the compiled sensor values from a string with a delimiter, Figure 3.8.3 will be implemented. The string will be split using a delimiter of ‘,’, the split parts will then be placed to an array named string [] parts that will contain all of the sensor values but with a string data type, to be able to process this data it needs to be converted to a double data type. The length of the string [] parts array, each element of this array will be parsed to double and will be placed in a new array named double [] values array.

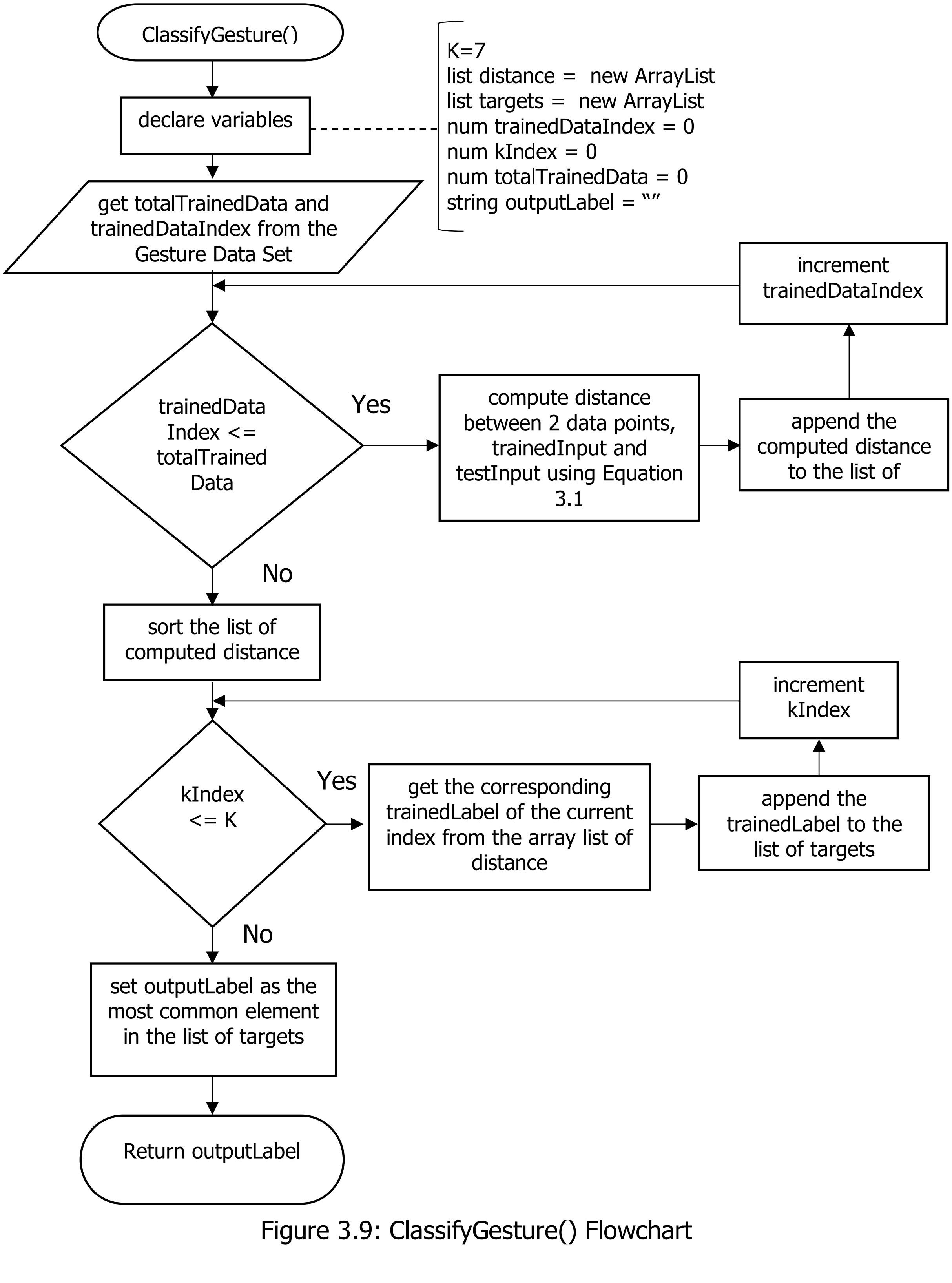

ClassifyGesture()
declare variables
K=7
list distance = new ArrayList
list targets = new ArrayList
num trainedDataIndex = 0
num kIndex = 0
num totalTrainedData = 0
string outputLabel = “”
get totalTrainedData and trainedDataIndex from the Gesture Data Set
increment trainedDataIndex
trainedData Index <= totalTrained Data
Yes
compute distance between 2 data points, trainedInput and testInput using Equation 3.1
append the computed distance to the list of
No
sort the list of computed distance
increment kIndex
kIndex <= K
Yes
get the corresponding trainedLabel of the current index from the array list of distance
append the trainedLabel to the list of targets
No
set outputLabel as the most common element in the list of targets
Return outputLabel

Figure 3.9: ClassifyGesture() Flowchart

Preparing the Software to be used

In writing the program and creating the installer for the android application, the Android Studio will be used. This is the official IDE for android development, and it uses Java as its main programming language.

Training Application Development

Needs of the Program that must be satisfied

A training application is needed to create a dataset that will be used in the android application. The training application should be able to receive sensor data from the microcontroller of the gloves and classify it to a specific label using the K-Nearest Neighbor (KNN). Since KNN is a supervised learning algorithm, the training application should be able to ask the user if the classification is correct or not. In this way, every time the classification is wrong, it could be corrected. Lastly, when the user responds if the classification is correct or not, the application should be able to collect the received sensor data and add the proper label with it.

System Flowchart

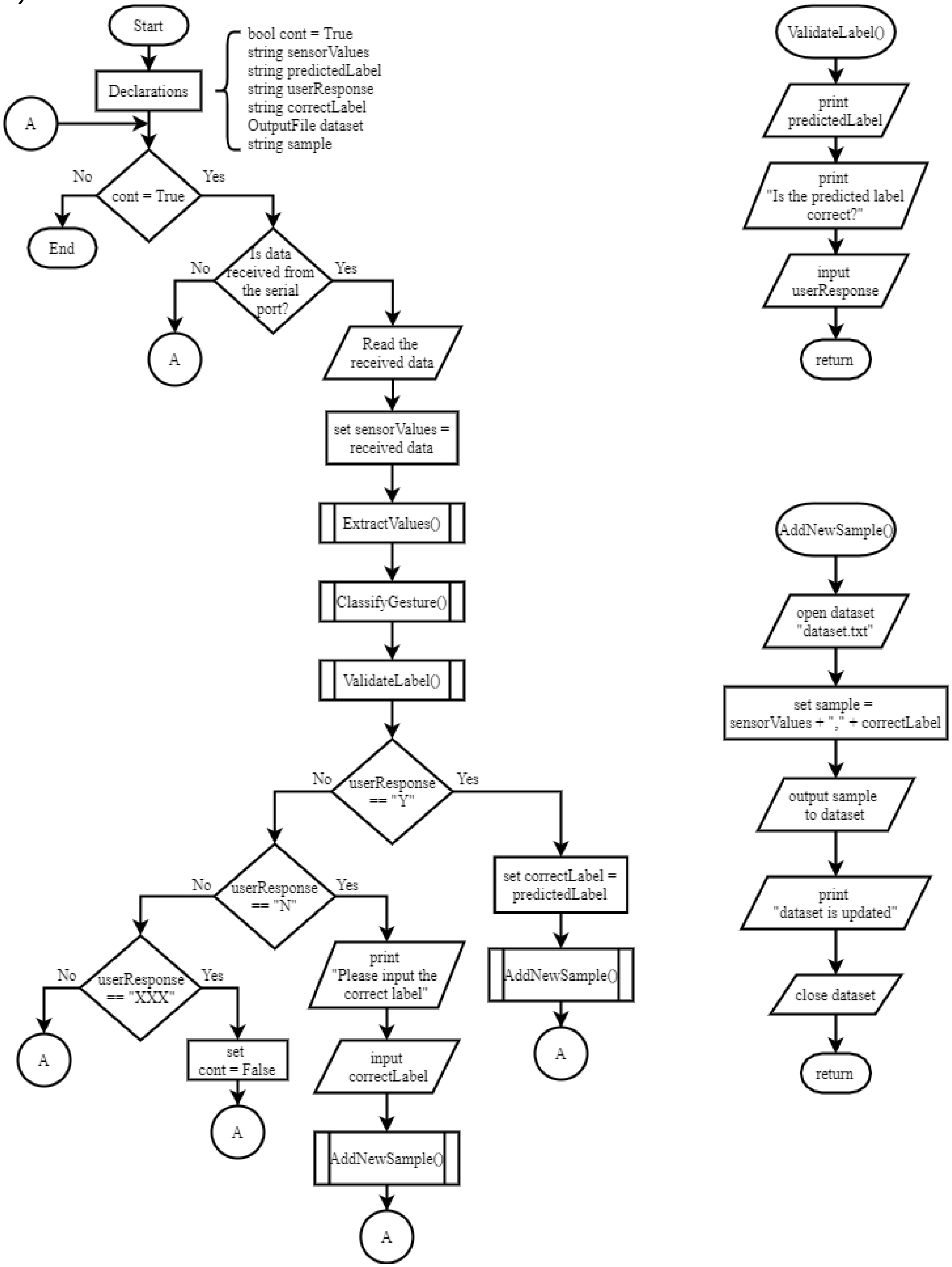


Figure 3.10: System Flowchart for Training Application

Figure 3.10 shows the flowchart to be followed in developing the training application. All variables to be used are declared at the start of the program. From here, the program will loop while the bool variable *cont* is set to true. Inside the loop, the program will check whether there is data received from the gloves via Bluetooth through the serial port. If data is received, the data will be set to a variable and will be used in the ExtractValues module, which can be seen in Figure 3.8.3, to get each sensor's values. The extracted sensor values will then be used in the ClassifyGesture module, which can be seen in Figure 3.9, to get the predicted label using K-Nearest Neighbors (KNN). Once the predicted label is acquired, it will then be validated by the user before it will be saved in the dataset. If the user input is "Y", the user implies that the predicted label is correct and the AddNewSample module will be used to save the sensor values and its corresponding label in the dataset. If the user input is "N", the user implies that the label is wrong and will then be asked for the correct label and the AddNewSample module will be called. If the user input is "XXX", the bool variable will be set to false and the program will end.

Preparing the Software to be used

A simple Python script will be created and will be used to create the dataset for the K-Nearest Neighbor algorithm. The chosen IDE to be used is PyCharm, as it is mainly used for Python programming language.

D. Prototype Development

Step-by-Step Procedure of Prototype Development

1. Identify the needs of the prototype that must be satisfied.
2. Upload the code to the ATmega328P.
3. Plan the layout and design of the PCB of the main module and the charging module of the prototype.
4. Prepare the components to be used according to the schematic diagram of the prototype.
5. Develop the PCB and solder the components in their proper location.
6. Plan and create the casing for the prototype.
7. Place and secure the PCB on the casing created.
8. Test the Prototype

The step by step procedure above shows the process for the development of the prototype. First, the needs of the prototype is identified. Then, the created code that will collect all sensor values will be uploaded to the ATmega328p. Then, the development of the PCB will be based on the materials that were used in the schematic diagram. After the PCB has been developed and all components are placed, the casing will then be created to protect all the components. The final step in the prototype development is to test it after all the components are in place and secured inside the casing.

a. PCB Design

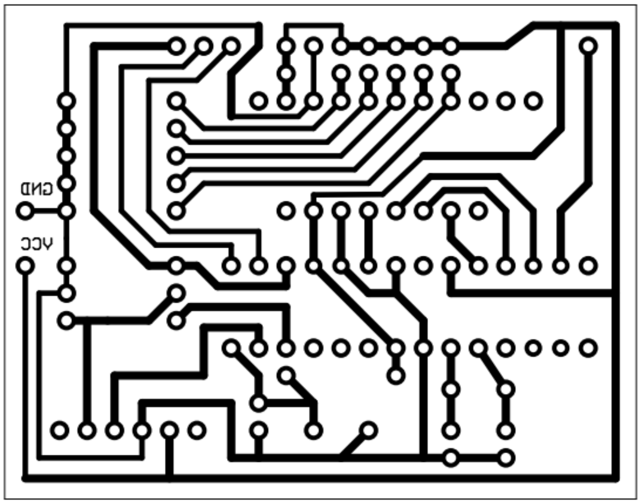


Figure 3.11: Main Module PCB Design

Figure 3.11 Illustrates the PCB design that was used for the main module. This module contains all the sensors that will send data to the mobile application.

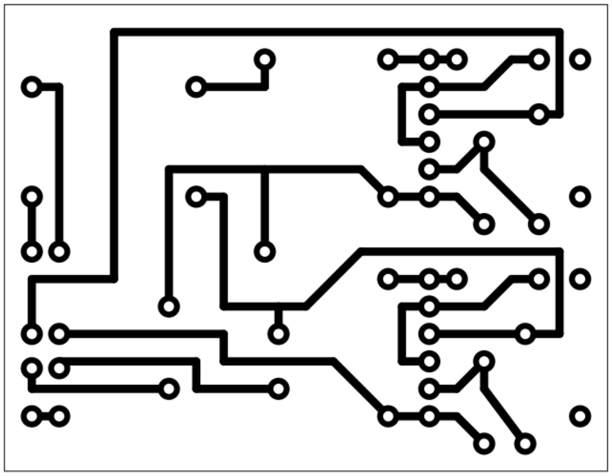


Figure 3.12: Battery Module PCB Design

Figure 3.12 shows the PCB design for the battery module. This module will be responsible for powering and charging the system.

b. PCB Component Diagram

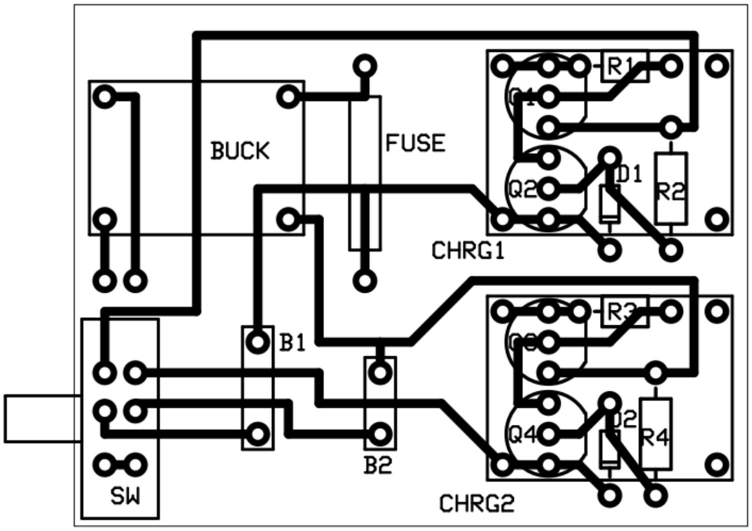


Figure 3.13: Main Module PCB Component Diagram

Figure 3.13 Illustrates the PCB component diagram that was used for the main module. This figure shows the placement of all the sensors and other components on the PCB.

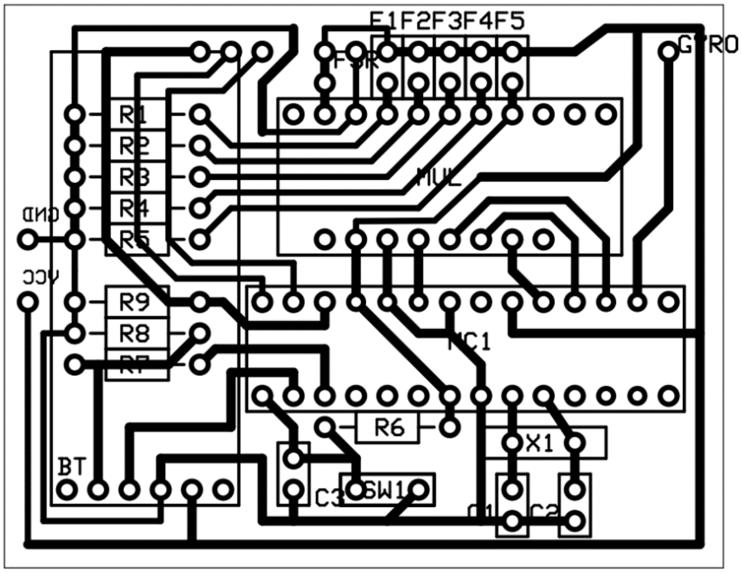


Figure 3.14: Battery Module PCB Component Diagram

Figure 3.14 shows the PCB component diagram for the battery module. This figure shows which components are soldered on the holes of the PCB.

c. Casing Model of the Prototype

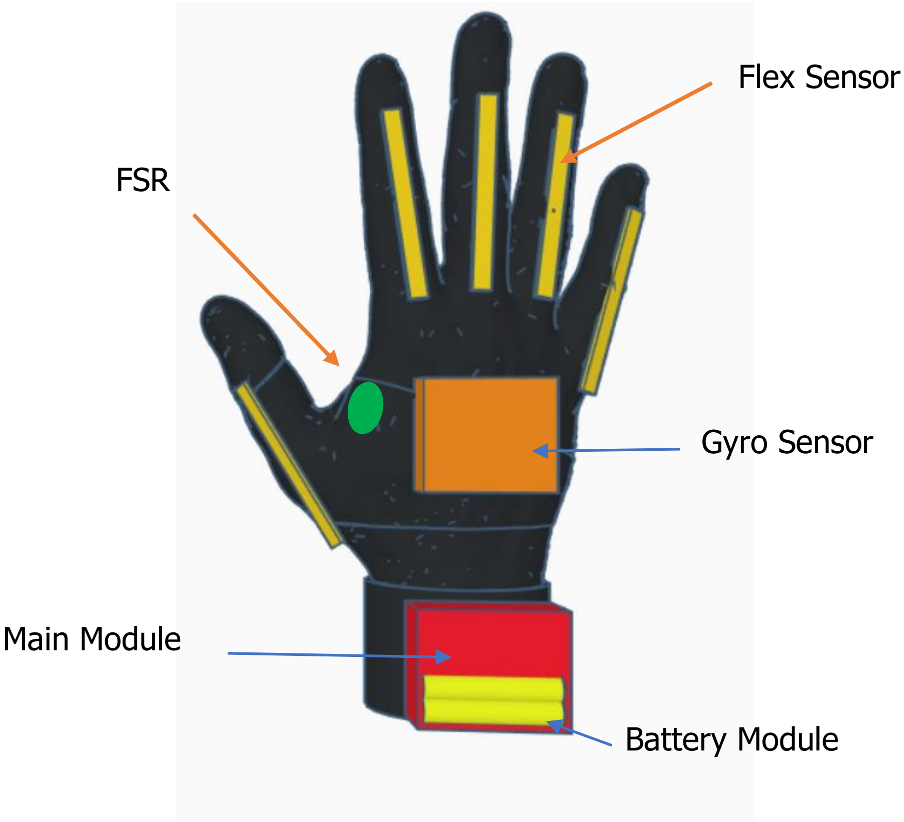


Figure 3.15: Components used in the Prototype

Figure 3.15 shows the casing for the components. The main module is composed of the components namely the ATmega328p and the Bluetooth module, the battery module which is made up of the TP4056 as well as the battery will be contained in a compact case will be placed in a case along the top of the wrist. The flex sensors will be placed on top of the fingers. The force-sensitive resistor is placed on the thenar space while the gyro sensor is placed on the back of the hand.

d. Bill of Materials

Table 3.1: Tabulated Bill of Materials

Material	Price (Php)	Quantity	Total (Php)
ATMega328P	126.00	1	126.00
Flex Sensor (4.5”)	999.75	3	2999.25
Flex Sensor (2.2”)	549.75	2	1099.5
14500 Li-Ion Battery	79.75	2	159.5
Force Sensitive Resistor	450.00	1	450.00
MPU6050	169.75	1	169.75
HC05 BT Module	299.75	1	299.75
TP4056	39.75	2	79.50
DPDT Switch	12.00	1	12.00
Fuse	10.00	1	10.00
Buck Converter Step Down Module	79.75	1	79.75
PCB (4x6)	50.00	1	50.00
74HC4051 Multiplexer	99.75	1	99.75
Battery Holder	15.00	1	15.00
		Grand Total	**Php 5,649.75**

Table 3.1 shows the list of materials and its corresponding price and quantity. The grand total for the equipment costs Php 5,649.75.

E. Multiple Design Constraints

Table 3.2: Decision Matrix of Design Constraints

Decision Evaluation Criteria							
Weight	Portability		Operation Time		Accuracy		Total
	0.3		0.2		0.5		1
	Score	Total Weight	Score	Total Weight	Score	Total Weight	
Ergonomics	**5**	**1.5**	**3**	**0.6**	**3**	**1.5**	**3.6**
Accessibility	3	0.9	3	0.6	1	0.5	2
Safety	3	0.9	1	0.2	1	0.5	1.6
Constructability	4	1.2	0	0	0	0	1.2
Usability	**3**	**0.9**	**4**	**0.8**	**5**	**2.5**	**4.2**
Sustainability	**4**	**1.2**	**3**	**0.6**	**1**	**0.5**	**2.3**

The scores for the multiple design constraints will be from 1 to 5, 1 being the lowest and 5 being the highest. There are multiple design constraints that the researchers have encountered, the most relevant of which are discussed below.

Usability

The machine learning algorithm that is used in performing the translation of sign language to text makes extensive use of computing power. A computer or a laptop could be used to perform computations. Another alternative would be to use an Android phone in the translation, but its computing power is sub-par compared to a computer.

Sustainability

The battery life of a Lithium-Ion based battery decreases performance overtime. An alternative would be to use a power cord; its usability is longer compared to a rechargeable battery.

Ergonomics

In transferring the data from the microcontroller to the computing device that will be used, a data cable from the microcontroller could be connected to the computing device. An alternative would be to make use of a Bluetooth connection between the two devices.

F. Trade-offs

Machine learning needs large amounts of memory to perform its tasks, a computer or a laptop is the most suitable platform to use since it has the power to perform computations even with a large data set, but it lacks the portability. On the other hand, an Android phone has less computing power compared to a computer or a laptop but offers great portability since it is a handheld device.

In implementing the design, either Li-Ion batteries or a simple power cord connection can be used to power the device. On one hand, the Li-Ion batteries' performance degrades depending on how many times it has been used, but it offers great portability for the device. On the other hand, a simple power cord connecting the device would give the device more time for usability, since it provides continuous power, and its life doesn't degrade with each use.

Using a data cable is viable in transferring data from the microcontroller to the Android phone; its transfer speed is faster compared to Bluetooth, but at the cost of the device's ergonomics since the microcontroller is on the glove, it would be difficult to perform some of the gestures while the user is tied down to the phone. Bluetooth, on the other hand, has a slower transfer speed but it does not use any wire to connect to the phone giving the user more freedom for the movement of their hand.

G. Selection of Alternative Solution

The scores for the decision matrix will be from 1 to 5, 1 being the lowest and 5 being the highest.

Table 3.3: Decision Matrix for Usability Constraint

Usability	Weight	Alternative Solution			
		A computer for performing computations		Android phone in the translation	
		Score	Weighted Score	Score	Weighted Score
Reliability	0.3	5	1.5	4	1.2
Portability	0.7	2	1.4	5	3.5
Total Weighted Score		2.9		4.7	

Table 3.3 shows the decision matrix for the usability constraint. This constraint is about the capability of the system in terms of providing portability while being used, as well as providing reliable results while using the device. Two solutions were formulated, one is the use of a computer in performing computation and another is the use of an android device. Both devices are capable of performing the K-Nearest Neighbor Algorithm, but the computer will outperform the android device in terms of raw performance. Since one of the objectives that are supposed to be met is the portability of the device, the Android phone was decided to be used in the solution. It may not have the same computing power as a computer or a laptop, but it still has sufficient memory to translate the sign language to text.

Table 3.4: Decision Matrix for Sustainability Constraint

Sustainability	Weight	Alternative Solution			
		Lithium-Ion based battery		Power Cord	
		Score	Weighted Score	Score	Weighted Score
Portability	0.6	5	3	3	1.8
Reliability	0.4	3	1.2	4	1.6
Total Weighted Score		4.2		3.4	

Table 3.4 shows the decision matrix for the sustainability constraint. This constraint is all about how long the device can be used while giving out good results in terms of gesture classification while maintaining its portability. There were two alternatives formulated, this is through the use of a Lithium-Ion based battery and another is through the use of a power cord to provide power to the system. The use of a power cord will give a more stable power source for the system but since one of the objectives is to make the device portable, the Lithium-Ion based battery was selected, that is why a charging module was then added to make the device last longer. Again on the portability of the device, using a Li-Ion battery would enable the user to use the device anywhere, unlike the power cable where the user needs to plug into a socket, it would also need a power brick to lower the voltage, and to convert AC to DC coming from a socket, it will also hinder the user to do hand gestures freely because of the dangling power cord that is connected to the power supply.

Table 3.5: Decision Matrix for Ergonomics Constraint

Ergonomics	Weight	Alternative Solution			
		Use of a data cable from the microcontroller		Use of a Bluetooth connection	
		Score	Weighted Score	Score	Weighted Score
Reliability	0.5	5	2.5	4	2
Convenience	0.5	2	1	4	2
Total Weighted Score		3.5		4	

Table 3.5 shows the decision matrix for the ergonomics constraint. This constraint is about the convenience and the reliability of data transmission from the gloves to the computing device which is the android phone. Two solutions were formulated, first is the use of a data cable that will connect to the android device for data transmission and another is the use of a Bluetooth connection for wireless data transmission. In considering the ergonomics of the device, Bluetooth is the alternative that is used, with the use of Bluetooth, no wire will impede the user in making their hand gesture. The use of a wireless Bluetooth connection will free up the movements the users can take unlike with the data cable, the user should be mindful of their hand's movements because the wire may be tangled up.

H. Impact of Design Solution

The design will impact the livelihood of the deaf and their relatives. It can be very hard for a person to not understand what the other person is saying, what more when you can't communicate entirely. With the help of the device, more people will be able to communicate with the deaf directly by starting to learn sign language. Below is the impact of the design solution in different categories.

Global

In 2018, there are 466 million people in the world who are experiencing a disabling hearing loss [11]. The design would help start people's understanding of sign language. It will help newly deaf and their relatives to start learning the language and ease their way into more advanced sign language.

Societal

Most of the families in the urban poor sector in the Philippines prefer to let their children without disabilities go to school instead of those with disabilities [8]. Having this device where they can train by themselves. It would lead to more deaf children getting the education that they deserve. By lowering the barrier to which people can learn Filipino Sign Language, more people will be able to understand the deaf. It would break the stigma of deaf people not having the same capability to work as people without disabilities.

Economic

The design would also lead to the increase of employment for these people with disabilities, not only because more people with disabilities will have a chance of education, but also because more people will be able to understand sign language. Deaf people employed, especially in services, will have an easier time interacting with people.

Environmental

The design is environmentally friendly, in which the power supply used is a rechargeable battery which is far less waste compared to non-rechargeable ones that are binned after one usage.

I. Engineering Principles

To build a device capable of translating Filipino Sign Language to text, engineering principles were applied.

Electrical

Basic electrical principles were used to create the schematics of the device, and to determine the correct amount of power the device needs. The amount of voltage needed by the components of the device was determined using the appropriate datasheets.

Electronics

Basic electronics principles are used to make the devices work properly. It is used in the interconnections of the components, and the knowledge about transistors is used especially in the charging circuit to prevent any damage to the circuit in the event of placing the batteries incorrectly.

Programming

Programming is used in programming the microcontroller for it to be able to gather the values of the sensors and to send those values to the phone. It is also used in creating the Android app for translating the gestures into text.

J. Modern Engineering Tools

Arduino IDE

The Arduino IDE was used in programming the instructions for the ATmega328p IC. It is also used as a debugging tool to calibrate the sensors and to see the values that the sensors are getting.

Android Studio

The Android Studio is used to develop the app where the translated text is being displayed.

PyCharm

PyCharm is an IDE that enables users to code using the Python Language. It is used in developing the machine learning algorithm.

Livewire

Livewire is a software that provides the required tools to create the schematic diagram of the device. It is also used in simulating the circuit.

ExpressPCB

ExpressPCB is a program that provides tools to create the PCB layout of the main module and the charging module of the glove.

Chapter 4

TESTING, PRESENTATION, AND INTERPRETATION OF DATA

A. Picking the K value

It is important to pick the K-value to be used in the K-Nearest Neighbor (KNN) algorithm since it determines the classification of the gesture.

B. Purpose of the Test

Picking the value of K that will be used for the K Nearest Neighbor algorithm is important as it affects the result of each classification. A small value of K will result in a noise having a large effect on the results, while a large value of K will make it computationally expensive for the device it will be applied to.

C. Initial Assumptions

The initial assumptions for this test are that the optimal K value for the design is greater than 3, since that value is too small to give an accurate classification, but is less than 11 since the dataset that is used is not very big, the data points that have similar gestures will have a greater impact to the output.

D. Procedures in picking the K Value

To pick the K value that will be used for the KNN, the following steps are done.

1. A sample of sensor values from the gloves for each label was acquired through serial communication between the gloves and the computer using PuTTY.
2. The samples were prepared by placing it on a text file.
3. The text file containing the samples was used as an input to a simple python script that uses K Nearest Neighbor classification using the created dataset and applying different values of K.
4. The result of the classification is recorded and the score for each value of K is counted.
5. There are 5 trials. Repeat steps 1 to 4 for each trial.

E. Data Table for the value of K

Table 4.1: Picking the value of K (Trial 1)

Label	**Value of K**									
	1	**3**	**5**	**7**	**9**	**11**	**13**	**15**	**17**	**19**
A	A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C	Y
D	L	L	L	L	L	L	L	L	D	L
E	E	E	E	E	E	E	M	M	M	M
F	F	F	F	F	F	F	F	F	F	F
G	G	G	G	G	G	G	G	G	G	G
H	U	U	U	U	U	U	U	U	U	U
I	I	I	I	I	I	I	I	I	I	I
J	J	J	J	J	J	J	J	J	J	J
K	U	U	U	U	U	U	U	U	U	U
L	L	L	L	L	L	L	L	L	L	L
M	M	M	M	M	M	M	M	M	M	M
N	N	N	N	N	N	N	N	N	N	N
O	O	O	O	O	O	O	O	O	O	O
P	P	P	P	P	H	H	H	H	H	H
Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
R	R	R	R	R	R	R	R	R	R	R
S	S	S	S	S	S	S	S	S	S	S
T	X	T	N	A	A	T	T	T	A	A
U	U	U	U	U	U	U	U	U	U	U
V	V	V	V	V	V	V	V	V	V	V
W	W	W	W	W	W	W	W	W	W	W
X	X	X	X	X	T	X	X	X	A	A
Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z
Space	__	__	__	__	__	__	__	__	__	__
Score	23/27	24/27	23/27	23/27	21/27	23/27	22/27	22/27	21/27	20/27

Table 4.2: Picking the value of K (Trial 2)

Label	**Value of K**									
	1	**3**	**5**	**7**	**9**	**11**	**13**	**15**	**17**	**19**
A	A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B	B
C	O	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D	D
E	E	E	M	M	M	M	M	M	M	M
F	F	F	F	F	F	F	F	F	F	F
G	G	G	G	G	G	G	G	G	G	G
H	H	H	H	H	H	H	H	H	H	H
I	I	I	I	I	I	I	I	I	I	I
J	I	I	I	I	I	I	I	I	I	I
K	K	K	K	K	K	K	K	K	K	K
L	L	L	L	L	L	L	L	L	L	L
M	M	M	M	M	M	M	M	M	M	M
N	N	N	N	N	N	N	N	N	N	N
O	O	O	O	O	O	O	O	O	O	O
P	P	P	P	P	P	P	P	P	P	P
Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
R	R	R	K	R	R	R	R	R	R	R
S	S	S	S	S	S	S	S	S	S	S
T	T	T	T	T	T	T	T	T	T	T
U	U	U	U	U	U	U	U	U	U	U
V	V	M	M	M	V	V	V	V	V	V
W	W	W	W	W	W	W	W	W	W	W
X	T	T	T	T	T	T	T	T	T	T
Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Z	Z	Z	Z	Z	T	T	T	T	T	T
Space	___	___	___	___	___	___	___	___	___	___
Score	24/27	24/27	22/27	23/27	23/27	23/27	23/27	23/27	23/27	23/27

Table 4.3: Picking the value of K (Trial 3)

Label	**Value of K**									
	1	**3**	**5**	**7**	**9**	**11**	**13**	**15**	**17**	**19**
A	A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C	C
D	D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	M	M	M	M	M
F	F	F	F	F	F	F	F	F	F	F
G	G	G	G	G	G	G	G	G	G	G
H	H	H	H	H	H	H	H	H	H	H
I	I	I	I	I	I	I	I	I	I	I
J	J	I	I	I	I	I	I	I	I	I
K	U	K	U	K	K	U	K	K	U	K
L	L	L	L	L	L	L	L	L	L	L
M	M	M	M	M	M	M	M	M	M	M
N	N	N	N	N	N	N	N	N	N	N
O	O	O	O	O	O	O	O	O	O	O
P	P	P	P	P	P	P	P	P	P	P
Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
R	U	R	R	R	U	U	R	R	U	U
S	S	S	S	S	S	S	S	S	S	S
T	X	T	T	T	T	T	T	T	T	T
U	U	U	U	U	U	U	U	U	U	U
V	V	V	V	V	V	V	V	V	V	V
W	W	W	W	W	W	W	W	W	W	W
X	T	X	X	X	X	X	X	X	X	X
Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Z	Z	Z	T	Z	T	T	T	T	T	T
Space	—	—	—	—	—	—	—	—	—	—
Score	23/27	26/27	24/27	26/27	24/27	22/27	24/27	24/27	22/27	23/27

Table 4.4: Picking the value of K (Trial 4)

Label	**Value of K**									
	1	**3**	**5**	**7**	**9**	**11**	**13**	**15**	**17**	**19**
A	A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	O	O
D	D	D	D	D	D	D	D	D	D	D
E	E	E	E	E	E	E	E	E	M	M
F	F	F	F	F	F	F	F	F	F	F
G	G	G	G	G	G	G	G	G	G	G
H	H	H	H	H	H	H	H	H	H	H
I	I	I	I	I	I	I	I	I	I	I
J	I	I	I	I	I	I	I	I	I	I
K	U	U	U	U	U	U	U	K	K	K
L	L	L	L	L	L	L	L	L	L	L
M	M	M	M	M	M	M	M	M	M	M
N	N	N	N	N	N	N	N	N	N	N
O	O	O	O	O	C	O	C	C	C	C
P	P	P	P	P	P	P	P	P	P	P
Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
R	R	K	K	K	K	K	K	K	K	K
S	S	S	S	S	S	S	S	S	S	S
T	X	X	X	X	X	X	X	X	X	X
U	U	U	U	U	U	U	U	U	U	U
V	V	V	V	V	V	V	V	V	V	V
W	W	W	W	W	W	W	W	W	W	W
X	X	X	X	X	X	X	X	X	X	X
Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z
Space	—	—	—	—	—	—	—	—	—	—
Score	24/27	23/27	23/27	23/27	22/27	23/27	22/27	23/27	21/27	21/27

Table 4.5: Picking the value of K (Trial 5)

Label	**Value of K**									
	1	**3**	**5**	**7**	**9**	**11**	**13**	**15**	**17**	**19**
A	A	A	A	A	A	A	A	A	A	A
B	B	B	B	B	B	B	B	B	B	B
C	C	C	C	C	C	C	C	C	C	Y
D	D	D	D	D	D	D	D	D	D	D
E	E	E	M	M	M	M	M	M	M	M
F	F	F	F	F	F	F	F	F	F	F
G	G	G	G	G	G	G	G	G	G	G
H	H	H	H	H	H	H	H	H	H	H
I	I	I	I	I	I	I	I	I	I	I
J	J	J	J	J	J	J	J	J	J	J
K	U	U	U	U	U	U	U	U	U	U
L	L	L	L	L	L	L	L	L	L	L
M	M	M	M	M	M	M	M	M	M	M
N	N	N	N	N	N	N	N	N	N	N
O	O	O	O	O	O	O	O	O	O	Y
P	P	P	P	P	P	H	H	H	H	H
Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
R	R	K	K	K	K	K	K	K	K	K
S	S	S	S	S	S	S	S	S	S	S
T	T	T	T	T	T	T	T	T	T	T
U	U	K	K	K	K	K	K	K	K	K
V	R	V	R	R	V	V	V	V	V	V
W	W	W	W	W	W	W	W	W	W	W
X	X	X	X	X	X	X	X	X	X	X
Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Z	R	T	T	T	T	T	T	T	T	T
Space	___	___	___	___	___	___	___	___	___	___
Score	24/27	23/27	21/27	21/27	22/27	21/27	21/27	21/27	21/27	20/27

Table 4.6: Picking the value of K (Summary of 5 Trials)

Trial	**Value of K**									
	1	**3**	**5**	**7**	**9**	**11**	**13**	**15**	**17**	**19**
1	23/27	24/27	23/27	23/27	21/27	23/27	22/27	22/27	21/27	20/27
2	24/27	24/27	22/27	23/27	23/27	23/27	23/27	23/27	23/27	23/27
3	23/27	26/27	24/27	26/27	24/27	22/27	24/27	24/27	22/27	23/27
4	24/27	23/27	23/27	23/27	22/27	23/27	22/27	23/27	21/27	21/27
5	24/27	23/27	21/27	21/27	22/27	21/27	21/27	21/27	21/27	20/27
Total	118	120	113	116	112	112	112	113	108	107

F. Analysis and interpretations of collected data

The table shows that the K value of 3 gives the highest total of correct classification. However, a small value of K means that noise will have a higher influence on the result. Thus, the K used for the KNN algorithm is 7.

A. Testing the Android Application

The application will be tested to measure its accuracy in translating FSL to text, this will be done by connecting the gloves to the created application to begin translating. The accuracy will be computed by dividing the number of successful attempts to the total attempts multiplied by 100.

B. Purpose of the Test

This test will measure the accuracy of the system in translating Filipino sign language to text to determine how many times the gestures being translated is correct with the use of the created android application.

C. Procedures in Testing the Application

To test the functionality of the system, the following steps are done.

1. Install and open the application on an Android Phone.
2. Turn on the gloves and wear it properly.
3. Tap the gesture classification button on the android application and connect to the gloves via Bluetooth.
4. Finger spell a letter and tap the Force Sensitive Resistor (FSR) for the gloves to be able to send data to the application.
5. The label of the translated sign language will then be displayed to the application.
6. If the displayed letter is the expected output, mark as a check, otherwise cross it out. Repeat steps 4 to 6 until the column is filled.
7. There will be 10 trials, the number of successful attempts will be recorded, and the accuracy will be computed. This is shown in Table 4.7

D. Initial Assumption

The researchers assume that the application will be able to successfully translate the sign language to text when a letter does not require movement as its input. It can also have some difficulty in translating letters with almost the same input such as I and J, S and T, and lastly U and V.

E. Data Table for the Android Application

Table 4:7: Android Application Testing

Label	Number of Successful Attempts	Number of Attempts	Accuracy (%)
A	10	10	100
B	10	10	100
C	10	10	100
D	9	10	90
E	7	10	70
F	10	10	100
G	10	10	100
H	9	10	90
I	10	10	100
J	6	10	60
K	10	10	100
L	10	10	100
M	9	10	90
N	10	10	100
O	10	10	100
P	10	10	100
Q	10	10	100
R	4	10	40
S	9	10	90
T	6	10	60
U	9	10	90
V	6	10	60
W	10	10	100
X	8	10	80
Y	10	10	100
Z	10	10	100
Space	10	10	100

F. Analysis and interpretations of collected data

Based on the collected data, the system shows an average of 89.63% accuracy for translating FSL to text using the KNN algorithm. Most of the letters in the alphabet have a 100% accuracy in translation except for D, E, H, J, M, R, S, T, U, and X, with R having the lowest accuracy of 40%.

Chapter 5
CONCLUSION AND RECOMMENDATION

Conclusion

The design project of creating a Filipino Sign Language Translator was developed for the client which helps their operation by aiding them to understand FSL. The system uses a flex sensor, gyroscope, and accelerometer which are used to capture hand movements and gestures that are used to interpret a sign language gesture. To be portable, the researchers utilized the portability of an android device and created an application that is capable of translating Filipino Sign language (FSL) to text. The K-Nearest Neighbor falls in the family of supervised learning, a labeled dataset consisting of the sensor values alongside with the corresponding label was given to the model as its training data which was used to classify an input from the user, the system shows an average of 89.63% accuracy for translating FSL to text using the KNN algorithm.

Recommendation

The prediction of the KNN algorithm is already acceptable; however, further improvements of the system could be done through the use of the following.

(1) The researchers recommend the use of a larger dataset for a better prediction.

(2) The researchers also recommend the collection of data with different hand sizes for much better accuracy.

Bibliography

[1] Ada, L. (2015, November 19). Force Sensitive Resistor (FSR). Retrieved March 1, 2019, from https://learn.adafruit.com/force-sensitive-resistor-fsr/using-an-fsr

[2] Cloutier, P., Helderman, A., & Mehilli, R. (2016). Prototyping a Portable, Affordable Sign Language Glove.

[3] Esposa, J. I., Jr. (2017, August 05). Difference between ASL and FSL. Retrieved February 27, 2019, from https://deafphilippines.wordpress.com/2017/08/03/difference-between-asl-and-fsl

[4] Girish, M. (2018, March 23). Bluetooth Basics: How to Control an LED Using an Android phone and Arduino. Retrieved March 1, 2019, from https://maker.pro/arduino/tutorial/bluetooth-basics-how-to-control-led-using-android phone-Arduino

[5] Harrison, O. (2018, September 10). Machine Learning Basics with the K-Nearest Neighbors Algorithm. Retrieved March 5, 2019, from https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761

[6] Inquirer, P. D. (2016, November 6). In The Know: Sign languages. Retrieved February 27, 2019, from https://newsinfo.inquirer.net/302184/in-the-know-sign-languages

[7] Lom, J. B. (2016). Flex Sensor Hookup Guide. Retrieved March 3, 2019, from https://learn.sparkfun.com/tutorials/flex-sensor-hookup-guide/all

[8] Mori, S. (2008). Economic Livelihood of the Deaf in the Philippines. *Poverty Reduction for the Disabled in the Philippines*, 159-164.

[9] Rosero-Montalvo, P. D., Godoy-Trujillo, P., Flores-Bosmediano, E., Carrascal-Garcia, J., Otero-Potosi, S., Benitez-Pereira, H., & Peluffo-Ordonez, D. H. (2018). Sign Language Recognition Based on Intelligent Glove Using Machine Learning Techniques. 2018 IEEE Third Ecuador Technical Chapters Meeting (ETCM).

[10] Sadam (2018, March 9). MPU6050 Gyro Sensor Interfacing with Arduino. Retrieved March 2, 2018, from https://circuitdigest.com/microcontroller-projects/mpu6050-gyro-sensor-interfacing-with-arduino

[11] World Health Organization (WHO). (2018, March 15). *Deafness*. Retrieved from World Health Organization: https://www.who.int/news-room/facts-in-pictures/detail/deafness#:~:text=There%20are%20466%20million%20people,rise%20to%20over%20900%20million.

Appendix

A. Article

B. Operation's Manual

1. System Requirements

Arduino IDE

- Microsoft Windows 7, Windows 8/8.1 and Windows 10 operating system.
- Microsoft .NET Framework 4.0 or higher. (Download from Microsoft Webpage)
- Intel Pentium / AMD Athlon processor or equivalent running at 1 GHz or more.
- 512 MB RAM (1 GB RAM recommended).
- 10MB free hard drive space or more (only for PROGRAMMING IDE for Arduino).

Android Studio

- Microsoft® Windows® 7/8/10 (64-bit)
- 4 GB RAM minimum, 8 GB RAM recommended.
- 2 GB of available disk space minimum, 4 GB Recommended (500 MB for IDE + 1.5 GB for Android SDK and emulator system image)
- 1280 x 800 minimum screen resolution.

Android Application

- Android Version:6.0.0 or above

- Capable of using Bluetooth Connection

2. Installation Procedures for the Android Application

1. Download the Android Package (APK) file.
2. Allow installation for external sources.
3. Run and install the APK application.

3. User's Manual

1. Enable Bluetooth connectivity of the Android Phone.
2. Open the Application.
3. Connect the Android Application to the device via Bluetooth.
4. Once connected the Device is ready to translate Sign Language to Text.

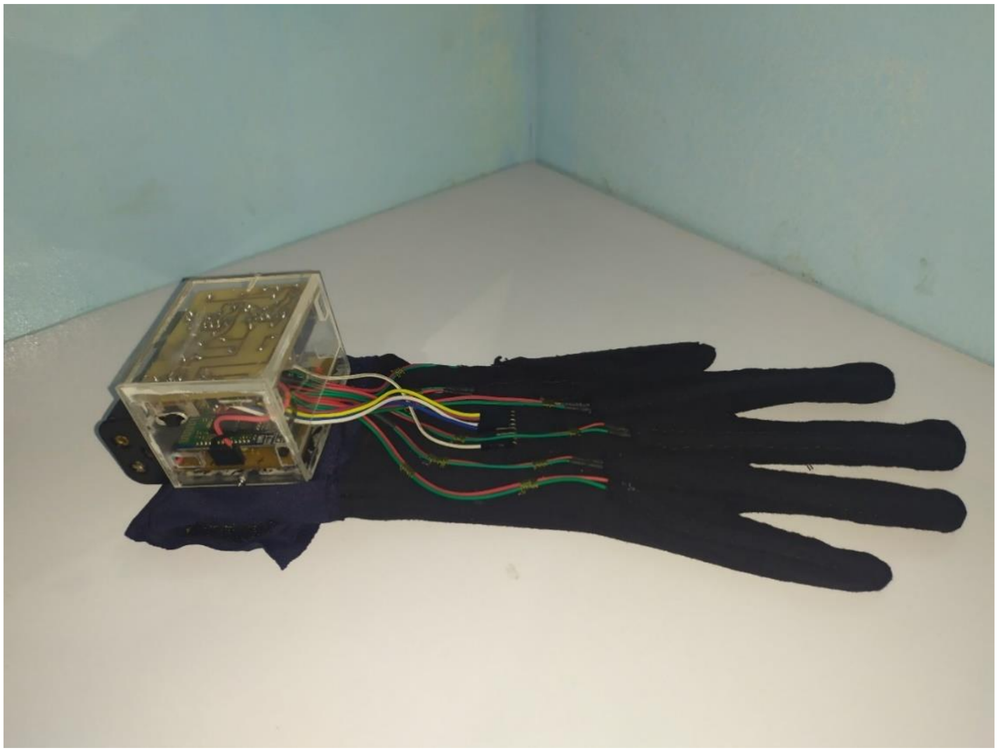
4. Troubleshooting Guides and Procedures

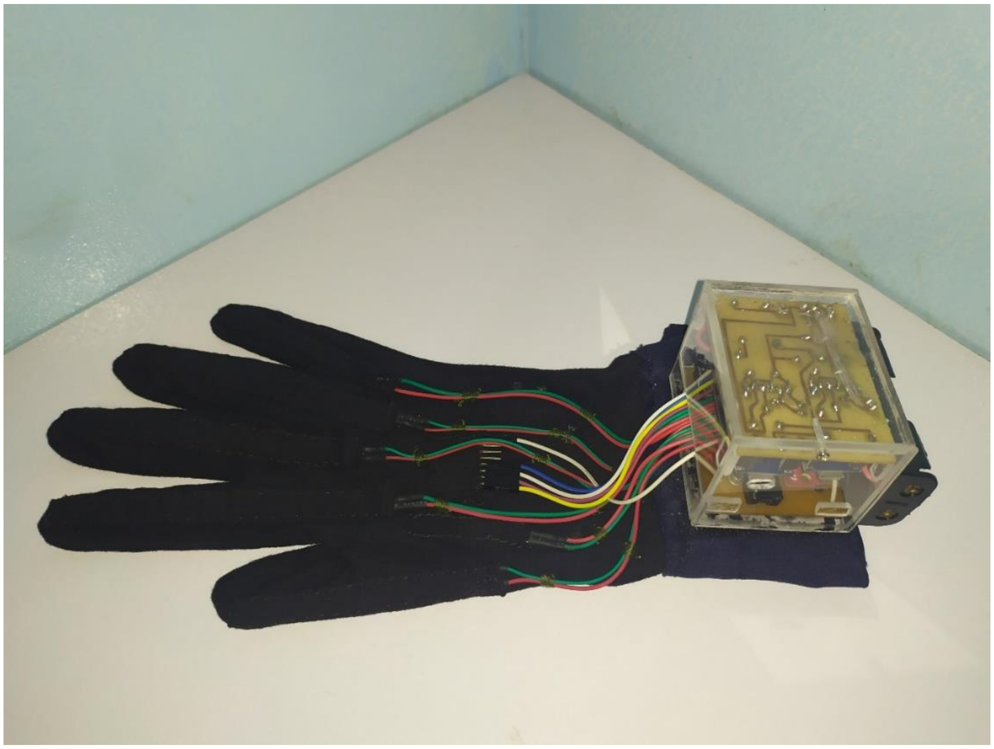
The device won't connect to the Application

1. Turn off the device and wait for at least 10 seconds.
2. Turn on the device.
3. Reconnect using the android application.

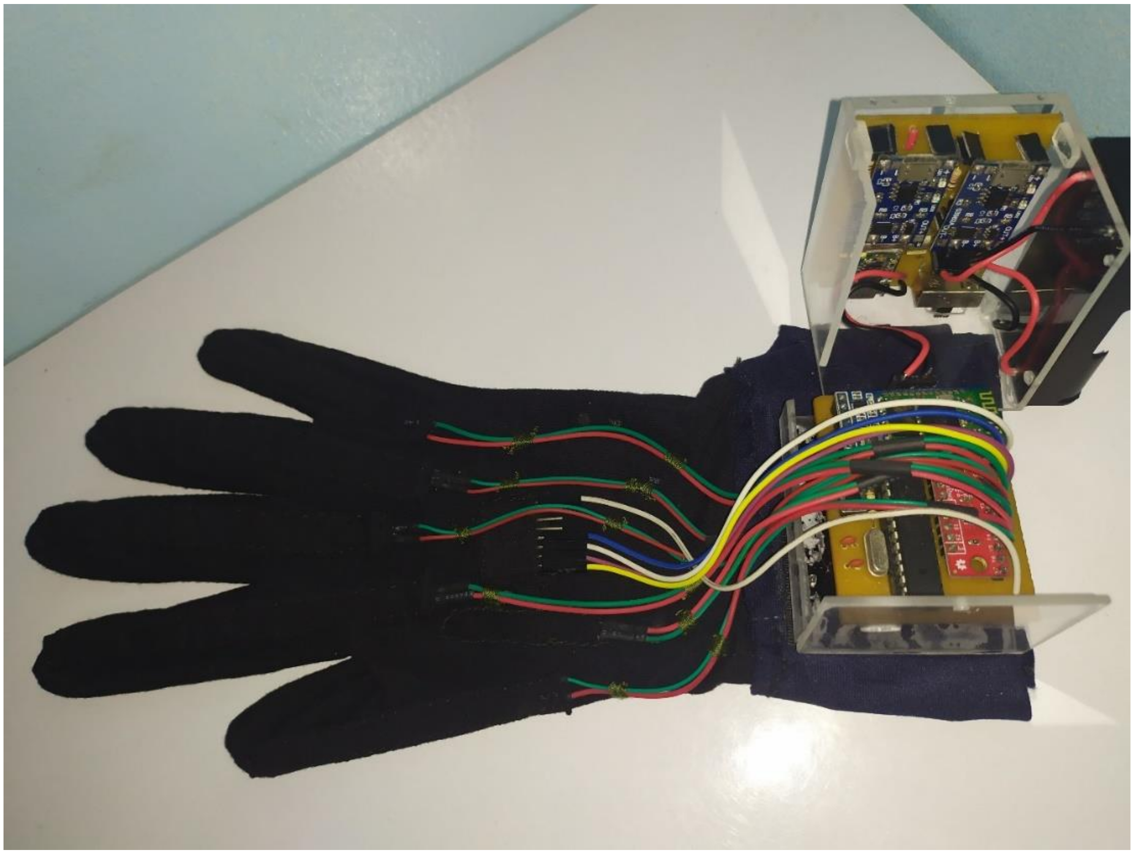
C. Pictures of Prototype

Side View



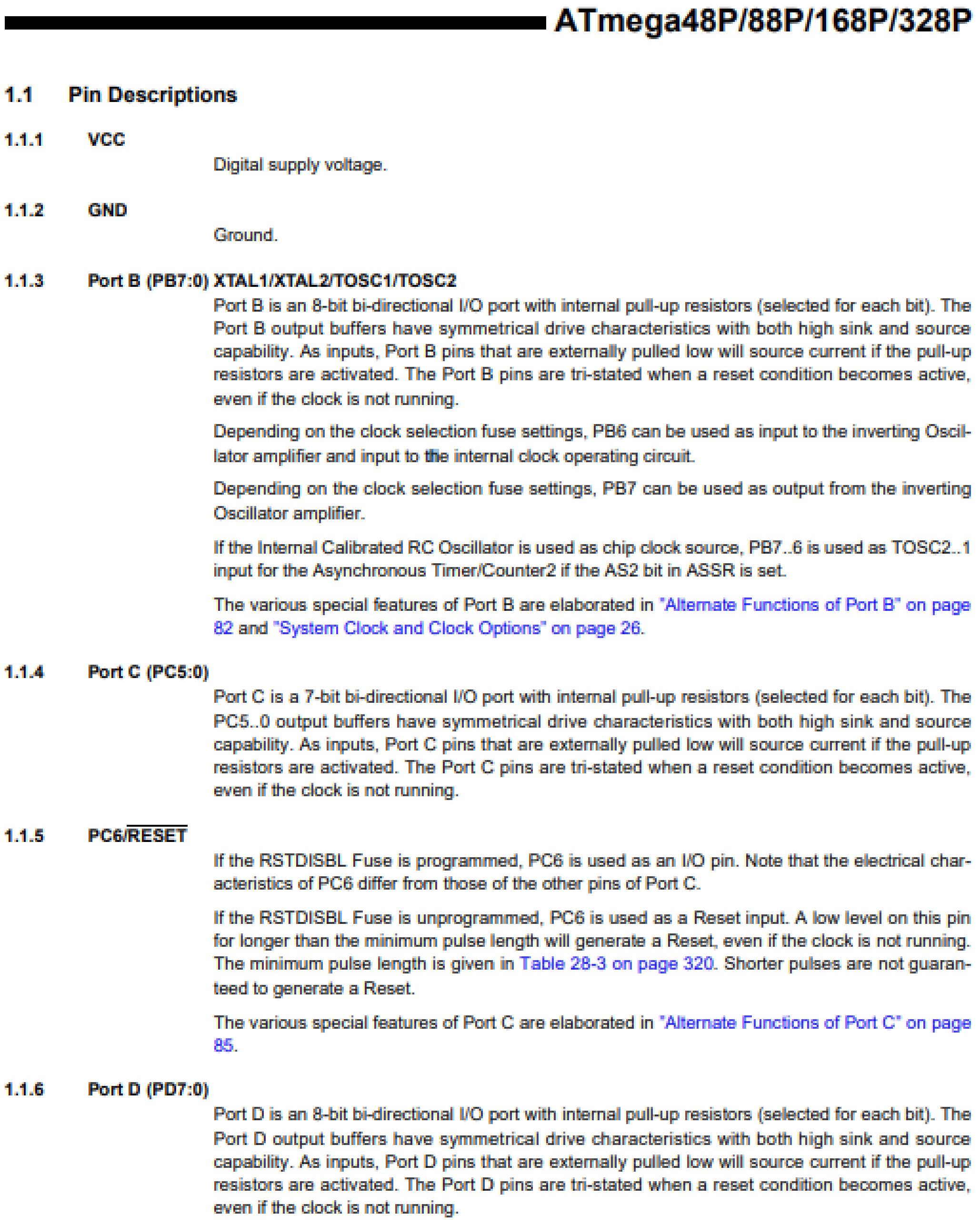


Top View



D. Data Sheets

ATMega328P



1.1 Pin Descriptions

1.1.1 VCC

Digital supply voltage.

1.1.2 GND

Ground.

1.1.3 Port B (PB7:0) XTAL1/XTAL2/TOSC1/TOSC2

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Depending on the clock selection fuse settings, PB6 can be used as input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

Depending on the clock selection fuse settings, PB7 can be used as output from the inverting Oscillator amplifier.

If the Internal Calibrated RC Oscillator is used as chip clock source, PB7..6 is used as TOSC2..1 input for the Asynchronous Timer/Counter2 if the AS2 bit in ASSR is set.

The various special features of Port B are elaborated in "Alternate Functions of Port B" on page 82 and "System Clock and Clock Options" on page 26.

1.1.4 Port C (PC5:0)

Port C is a 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The PC5..0 output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

1.1.5 PC6/$\overline{\text{RESET}}$

If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C.

If the RSTDISBL Fuse is unprogrammed, PC6 is used as a Reset input. A low level on this pin for longer than the minimum pulse length will generate a Reset, even if the clock is not running. The minimum pulse length is given in Table 28-3 on page 320. Shorter pulses are not guaranteed to generate a Reset.

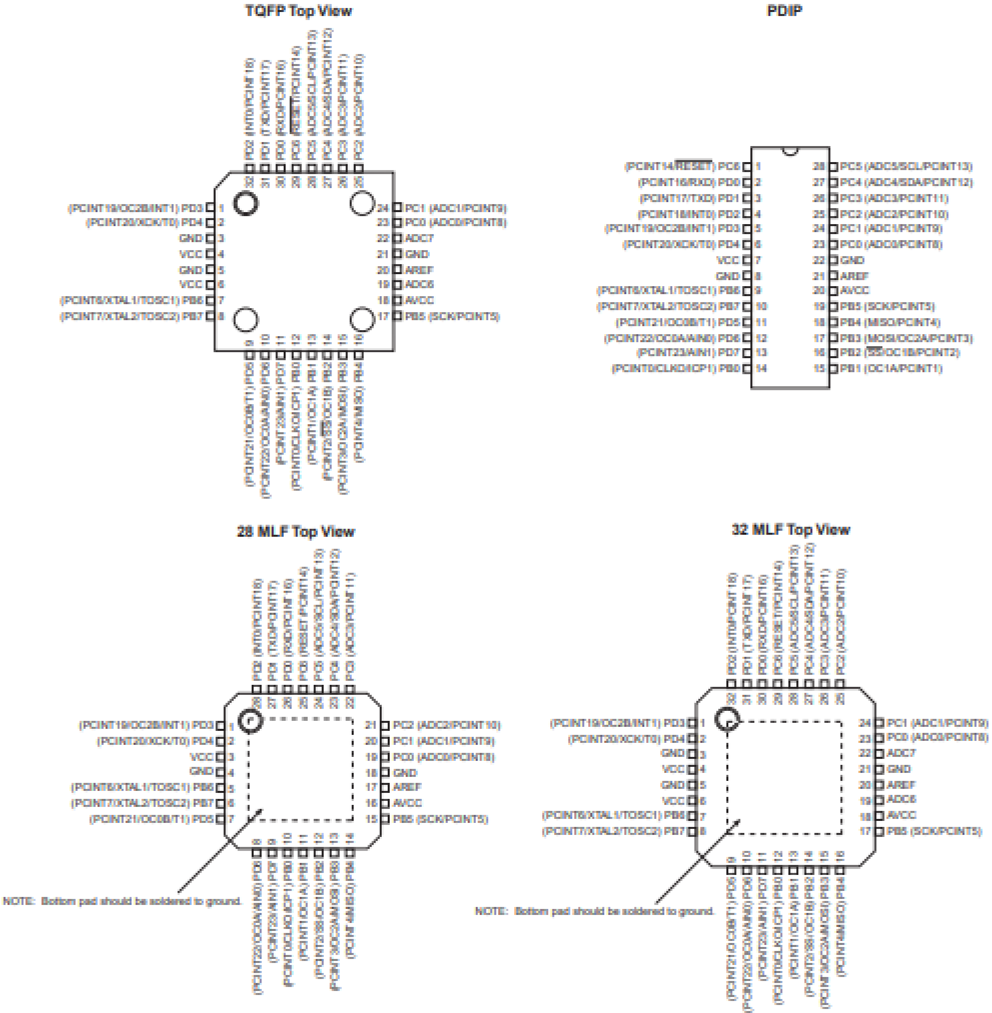
The various special features of Port C are elaborated in "Alternate Functions of Port C" on page 85.

1.1.6 Port D (PD7:0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

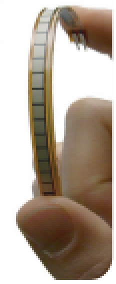
1. Pin Configurations

Figure 1-1. Pinout ATmega48P/88P/168P/328P



Flex Sensor Data Sheet

spectrasymbol



FLEX SENSOR FS

Special Edition Length

Features

- Angle Displacement Measurement
- Bends and Flexes physically with motion device
- Possible Uses
  - Robotics
  - Gaming (Virtual Motion)
  - Medical Devices
  - Computer Peripherals
  - Musical Instruments
  - Physical Therapy
- Simple Construction
- Low Profile

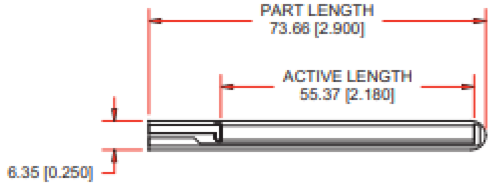
Mechanical Specifications

-Life Cycle: >1 million
-Height: ≤0.43mm (0.017")
-Temperature Range: -35°C to +80°C

Electrical Specifications

-Flat Resistance: 25K Ohms
-Resistance Tolerance: ±30%
-Bend Resistance Range: 45K to 125K Ohms (depending on bend radius)
-Power Rating : 0.50 Watts continuous. 1 Watt Peak

Dimensional Diagram - Stock Flex Sensor

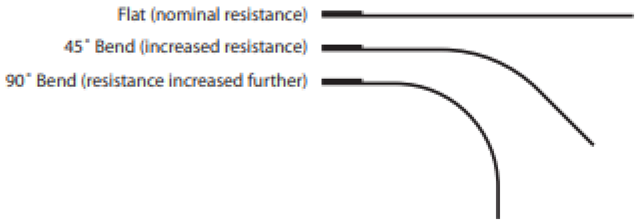


How to Order - Stock Flex Sensor

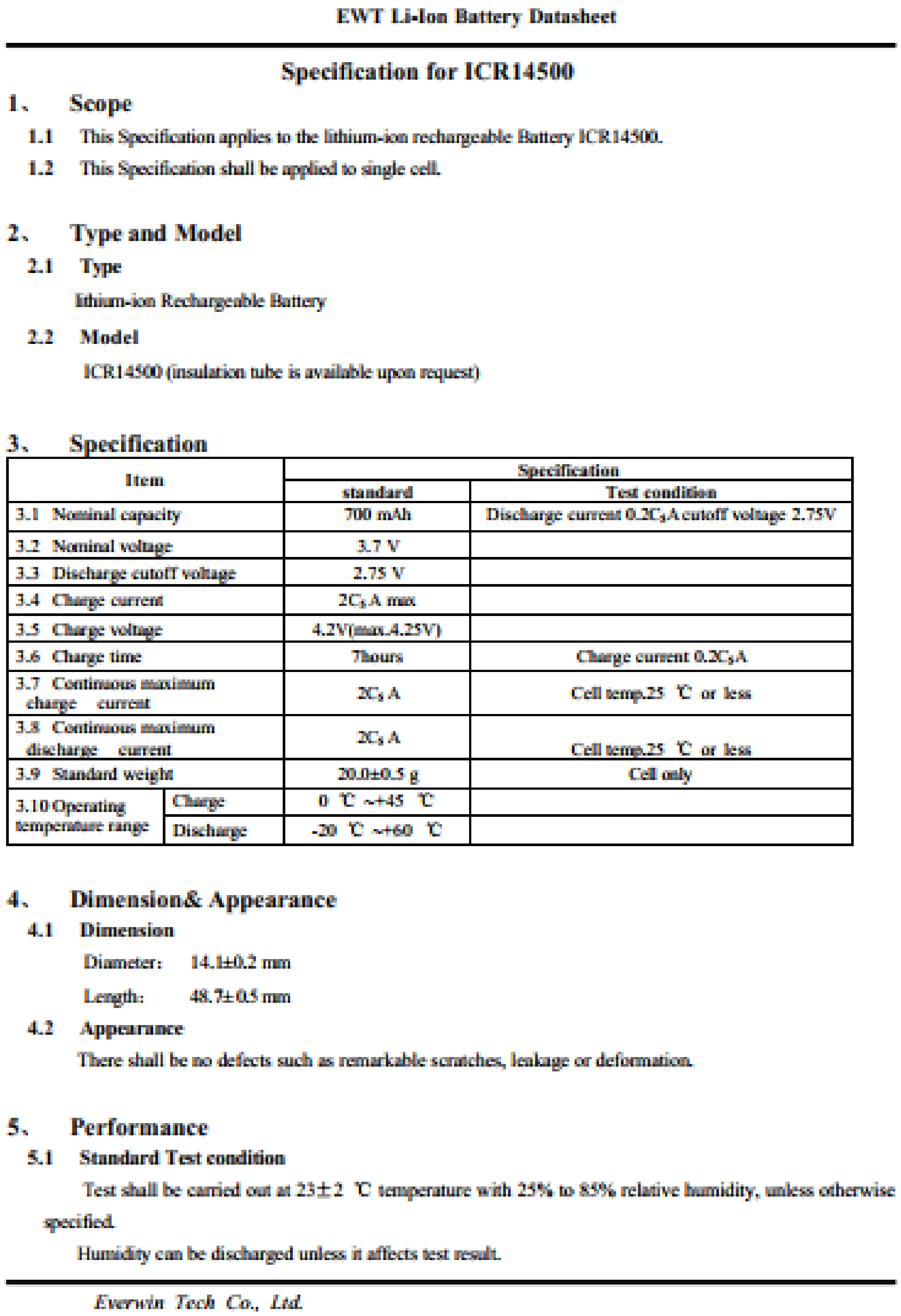
FS — L — 0055 — 253 — ST

Series	Model	Active Length	Resistance	Connectors
FS = Flex Sensor	L = Linear	0055 = 55.37mm	253 = 25K Ohms	ST = Solder Tab

How It Works


Flat (nominal resistance)
45˚ Bend (increased resistance)
90˚ Bend (resistance increased further)

14500 Li-Ion battery Data Sheet



EWT Li-Ion Battery Datasheet

Specification for ICR14500

1、 Scope

1.1 This Specification applies to the lithium-ion rechargeable Battery ICR14500.

1.2 This Specification shall be applied to single cell.

2、 Type and Model

2.1 Type

lithium-ion Rechargeable Battery

2.2 Model

ICR14500 (insulation tube is available upon request)

3、 Specification

Item		Specification	
		standard	Test condition
3.1 Nominal capacity		700 mAh	Discharge current 0.2C_5A cutoff voltage 2.75V
3.2 Nominal voltage		3.7 V	
3.3 Discharge cutoff voltage		2.75 V	
3.4 Charge current		2C_5 A max	
3.5 Charge voltage		4.2V(max.4.25V)	
3.6 Charge time		7hours	Charge current 0.2C_5A
3.7 Continuous maximum charge current		2C_5 A	Cell temp.25 ℃ or less
3.8 Continuous maximum discharge current		2C_5 A	Cell temp.25 ℃ or less
3.9 Standard weight		20.0±0.5 g	Cell only
3.10 Operating temperature range	Charge	0 ℃ ~+45 ℃	
	Discharge	-20 ℃ ~+60 ℃	

4、 Dimension& Appearance

4.1 Dimension

Diameter: 14.1±0.2 mm

Length: 48.7±0.5 mm

4.2 Appearance

There shall be no defects such as remarkable scratches, leakage or deformation.

5、 Performance

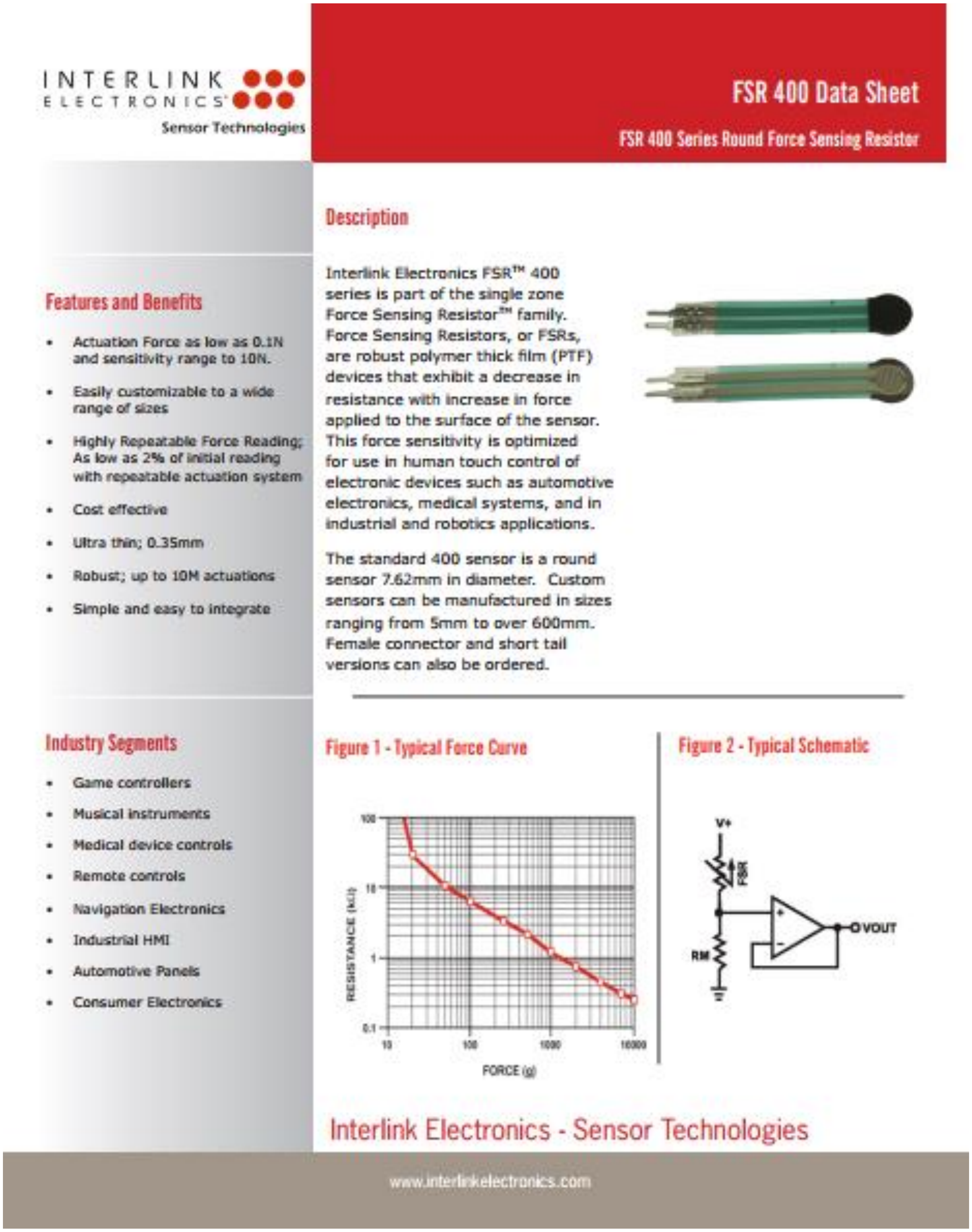
5.1 Standard Test condition

Test shall be carried out at 23±2 ℃ temperature with 25% to 85% relative humidity, unless otherwise specified.

Humidity can be discharged unless it affects test result.

Everwin Tech Co., Ltd.

FSR400 Data Sheet



INTERLINK ELECTRONICS

Sensor Technologies

FSR 400 Data Sheet

FSR 400 Series Round Force Sensing Resistor

Description

Interlink Electronics FSR™ 400 series is part of the single zone Force Sensing Resistor™ family. Force Sensing Resistors, or FSRs, are robust polymer thick film (PTF) devices that exhibit a decrease in resistance with increase in force applied to the surface of the sensor. This force sensitivity is optimized for use in human touch control of electronic devices such as automotive electronics, medical systems, and in industrial and robotics applications.

The standard 400 sensor is a round sensor 7.62mm in diameter. Custom sensors can be manufactured in sizes ranging from 5mm to over 600mm. Female connector and short tail versions can also be ordered.

Features and Benefits

- Actuation Force as low as 0.1N and sensitivity range to 10N.
- Easily customizable to a wide range of sizes
- Highly Repeatable Force Reading; As low as 2% of initial reading with repeatable actuation system
- Cost effective
- Ultra thin; 0.35mm
- Robust; up to 10M actuations
- Simple and easy to integrate

Industry Segments

- Game controllers
- Musical instruments
- Medical device controls
- Remote controls
- Navigation Electronics
- Industrial HMI
- Automotive Panels
- Consumer Electronics

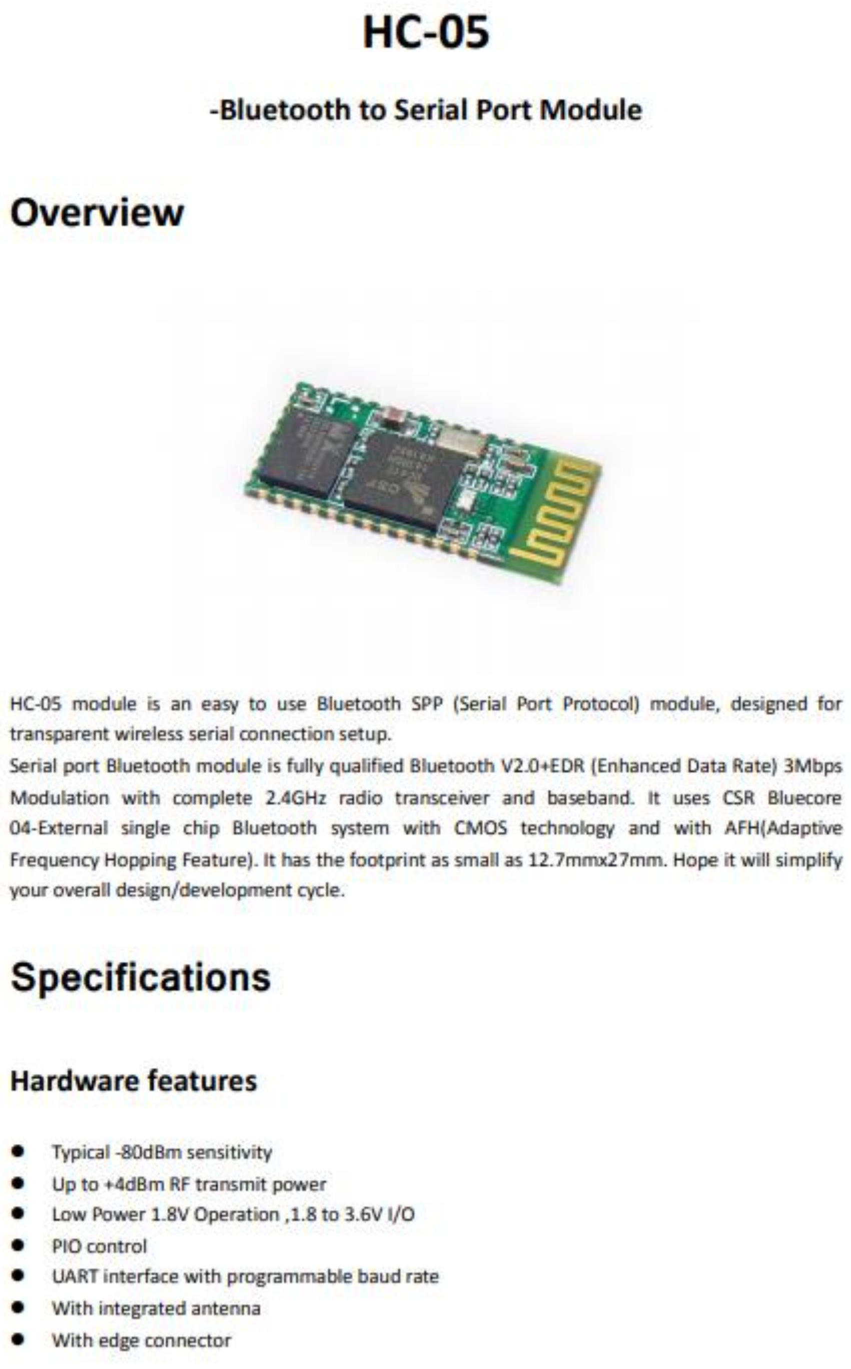
Figure 1 - Typical Force Curve

Figure 2 - Typical Schematic

Interlink Electronics - Sensor Technologies

www.interlinkelectronics.com

HC05 Bluetooth Module



HC-05

-Bluetooth to Serial Port Module

Overview

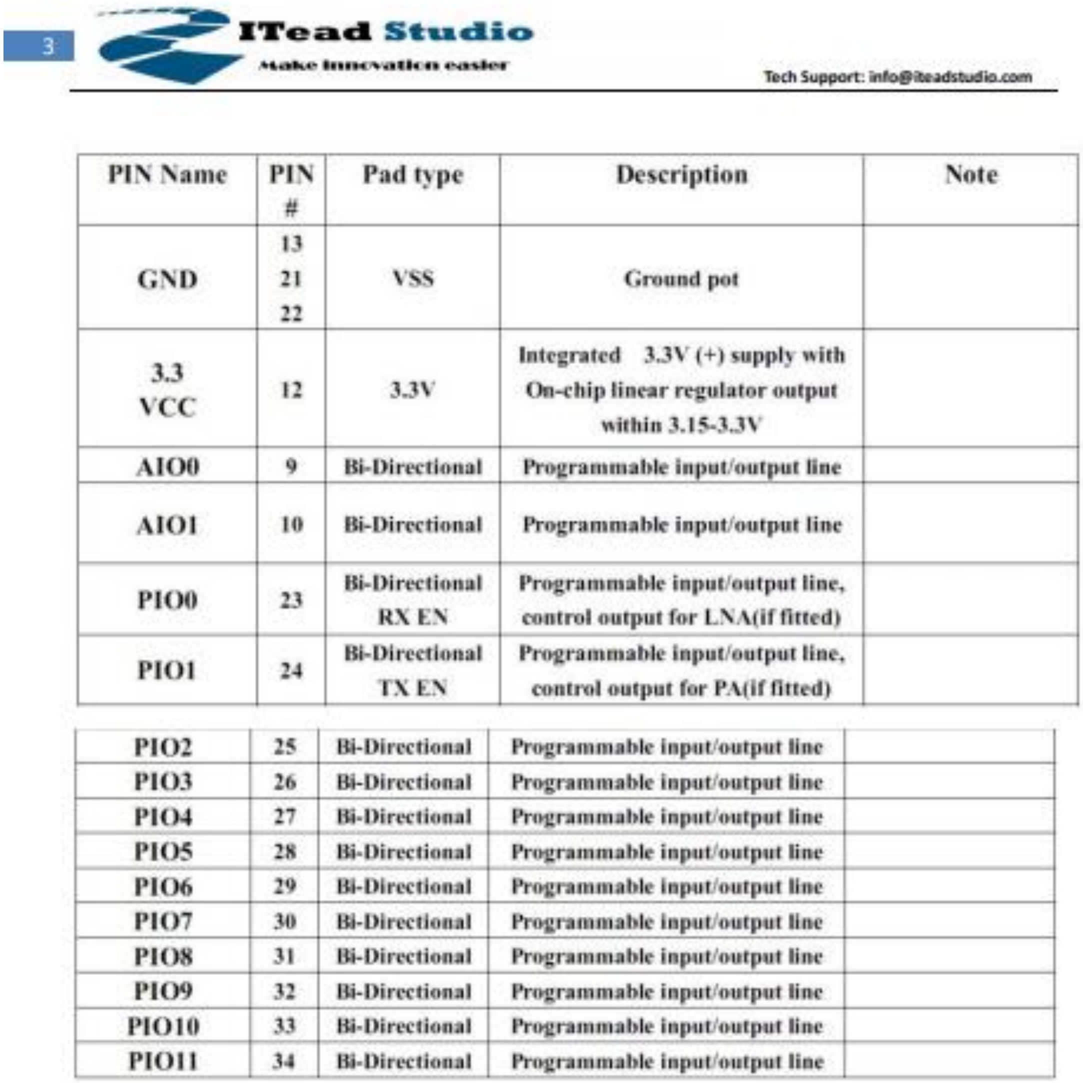
HC-05 module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup.

Serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses CSR Bluecore 04-External single chip Bluetooth system with CMOS technology and with AFH(Adaptive Frequency Hopping Feature). It has the footprint as small as 12.7mmx27mm. Hope it will simplify your overall design/development cycle.

Specifications

Hardware features

- Typical -80dBm sensitivity
- Up to +4dBm RF transmit power
- Low Power 1.8V Operation ,1.8 to 3.6V I/O
- PIO control
- UART interface with programmable baud rate
- With integrated antenna
- With edge connector

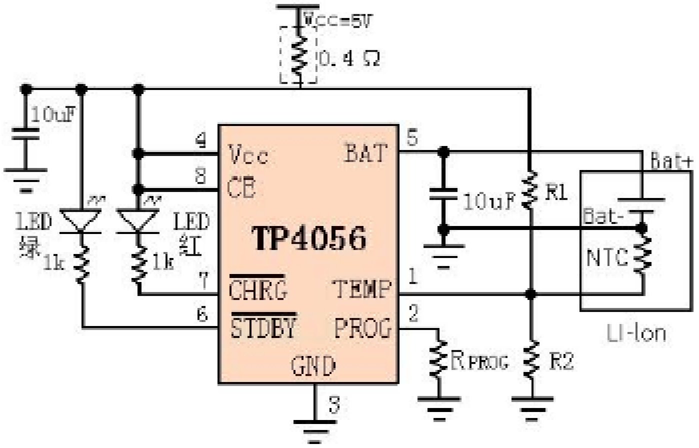


PIN Name	PIN #	Pad type	Description	Note
GND	13 21 22	VSS	Ground pot	
3.3 VCC	12	3.3V	Integrated 3.3V (+) supply with On-chip linear regulator output within 3.15-3.3V	
AIO0	9	Bi-Directional	Programmable input/output line	
AIO1	10	Bi-Directional	Programmable input/output line	
PIO0	23	Bi-Directional RX EN	Programmable input/output line, control output for LNA(if fitted)	
PIO1	24	Bi-Directional TX EN	Programmable input/output line, control output for PA(if fitted)	
PIO2	25	Bi-Directional	Programmable input/output line	
PIO3	26	Bi-Directional	Programmable input/output line	
PIO4	27	Bi-Directional	Programmable input/output line	
PIO5	28	Bi-Directional	Programmable input/output line	
PIO6	29	Bi-Directional	Programmable input/output line	
PIO7	30	Bi-Directional	Programmable input/output line	
PIO8	31	Bi-Directional	Programmable input/output line	
PIO9	32	Bi-Directional	Programmable input/output line	
PIO10	33	Bi-Directional	Programmable input/output line	
PIO11	34	Bi-Directional	Programmable input/output line	

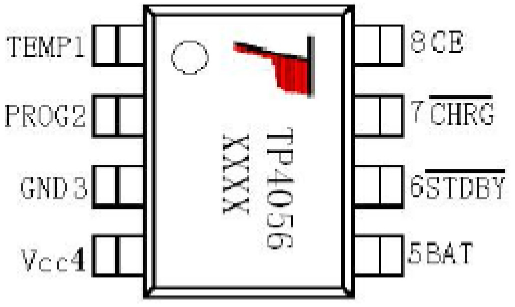
TP4056 Data Sheet

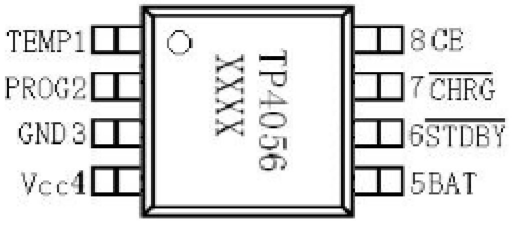
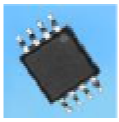
南京拓微集成电路有限公司 TP4056

典型应用



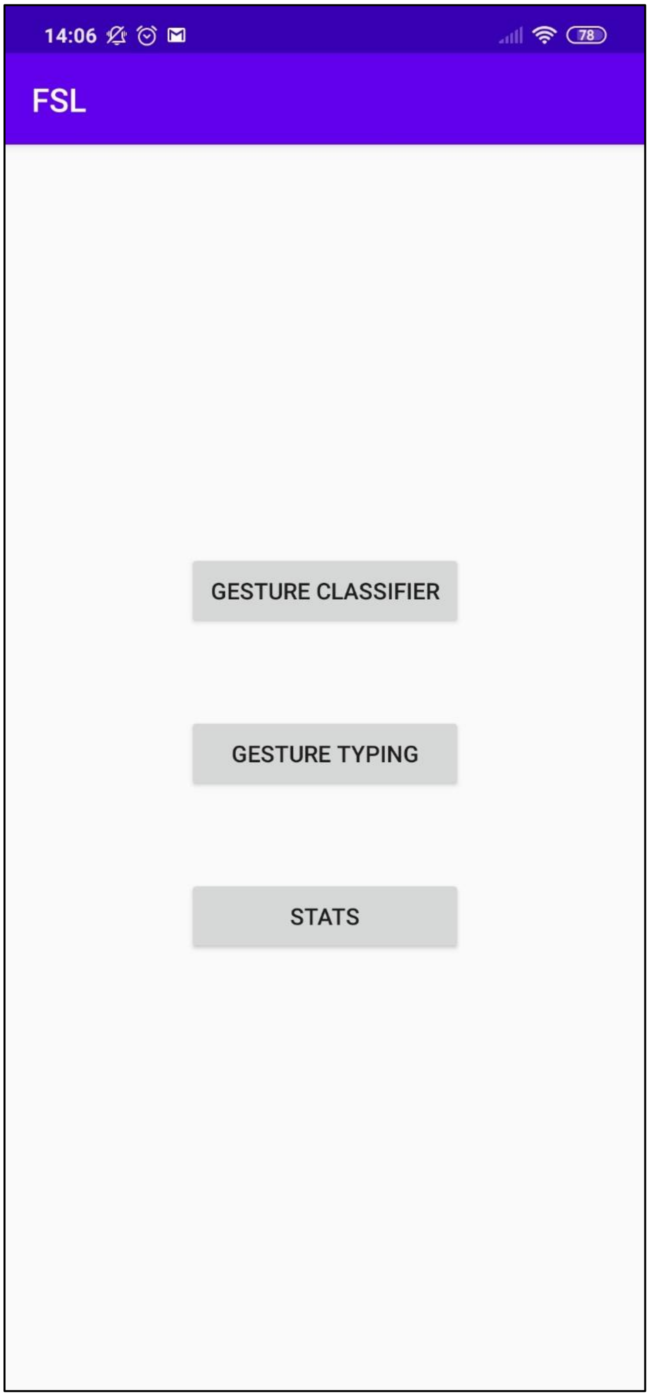
封装/订购信息

 TEMP1 PROG2 GND 3 Vcc4 8CE 7 CHRG 6STDBY 5BAT TP4056 XXXX 8 引脚 ESOP 封装(底部带有散热片)	订单型号
	TP4056-42-ESOP8
	器件标记
	TP4056
	实物图片

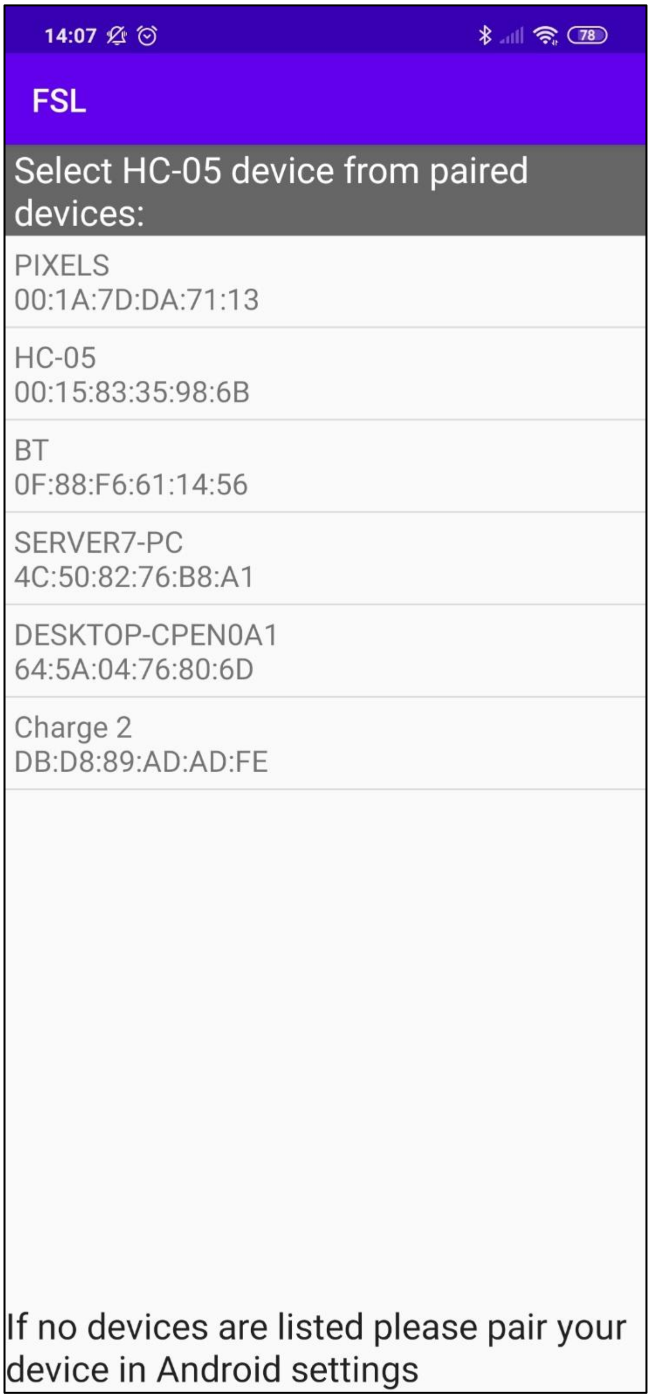
 TEMP1 PROG2 GND 3 Vcc4 8CE 7 CHRG 6STDBY 5BAT TP4056 XXXX 8 引脚 EMSOP 封装(底部带有散热片)	订单型号
	TP4056-42-EMSOP8
	器件标记
	TP4056
	实物图片
	

E. Android Application

Main Menu Activity Layout



Device Connectivity Layout



14:07

FSL

Select HC-05 device from paired devices:

PIXELS
00:1A:7D:DA:71:13

HC-05
00:15:83:35:98:6B

BT
0F:88:F6:61:14:56

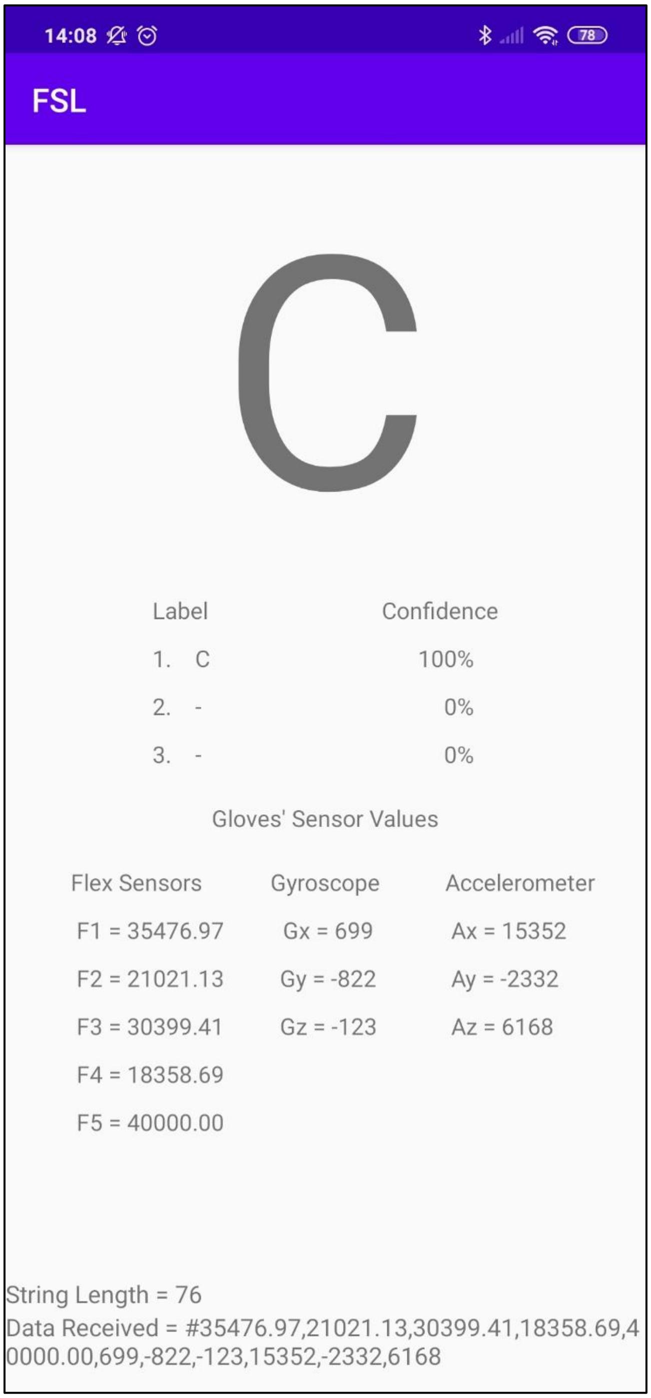
SERVER7-PC
4C:50:82:76:B8:A1

DESKTOP-CPEN0A1
64:5A:04:76:80:6D

Charge 2
DB:D8:89:AD:AD:FE

If no devices are listed please pair your device in Android settings

Gesture Classification Activity Layout



14:08 78

FSL

C

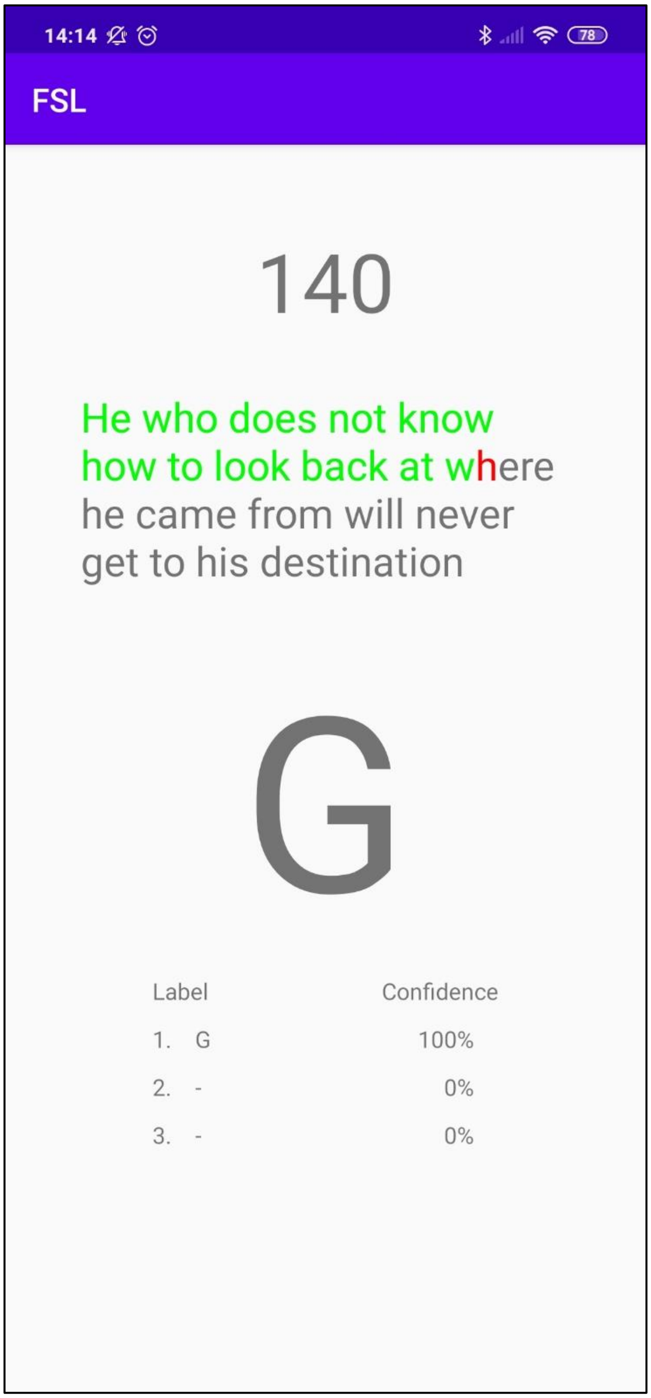
Label	Confidence
1. C	100%
2. -	0%
3. -	0%

Gloves' Sensor Values

Flex Sensors	Gyroscope	Accelerometer
F1 = 35476.97	Gx = 699	Ax = 15352
F2 = 21021.13	Gy = -822	Ay = -2332
F3 = 30399.41	Gz = -123	Az = 6168
F4 = 18358.69		
F5 = 40000.00		

String Length = 76
Data Received = #35476.97,21021.13,30399.41,18358.69,40000.00,699,-822,-123,15352,-2332,6168

Gesture Typing Activity Layout



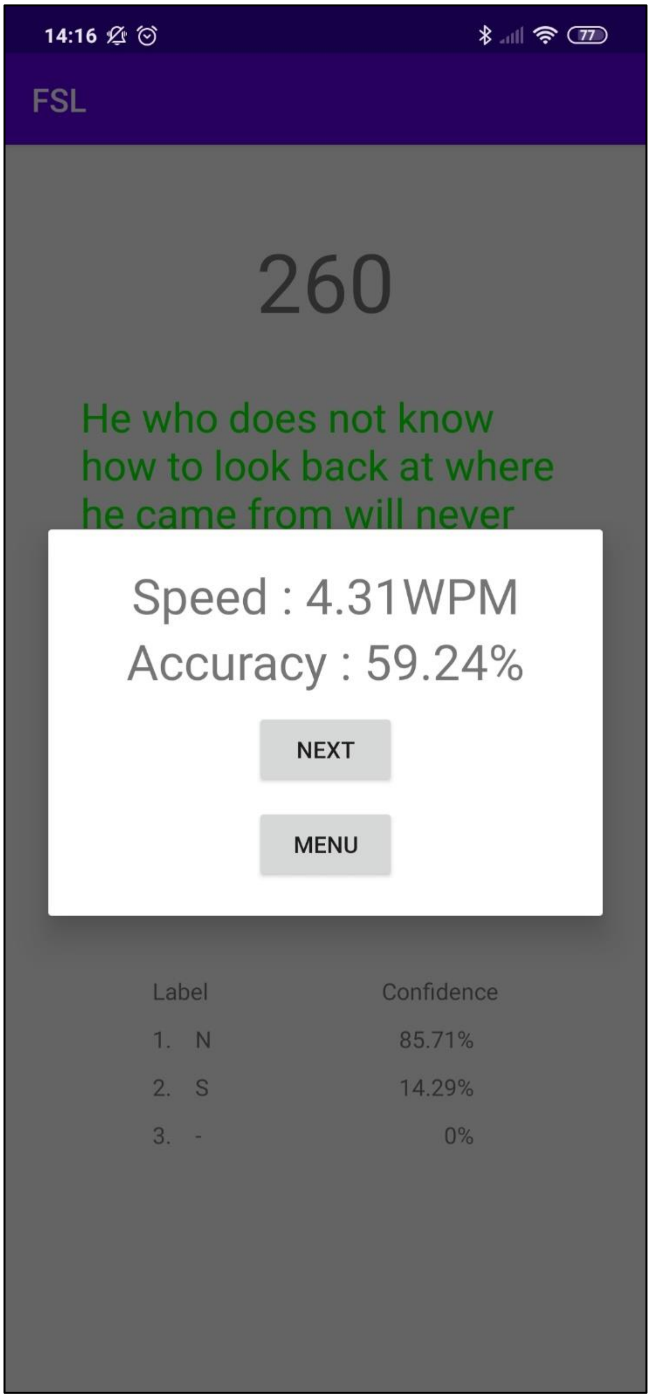
140

He who does not know how to look back at where he came from will never get to his destination

G

Label	Confidence
1. G	100%
2. -	0%
3. -	0%

Complete Dialog



260

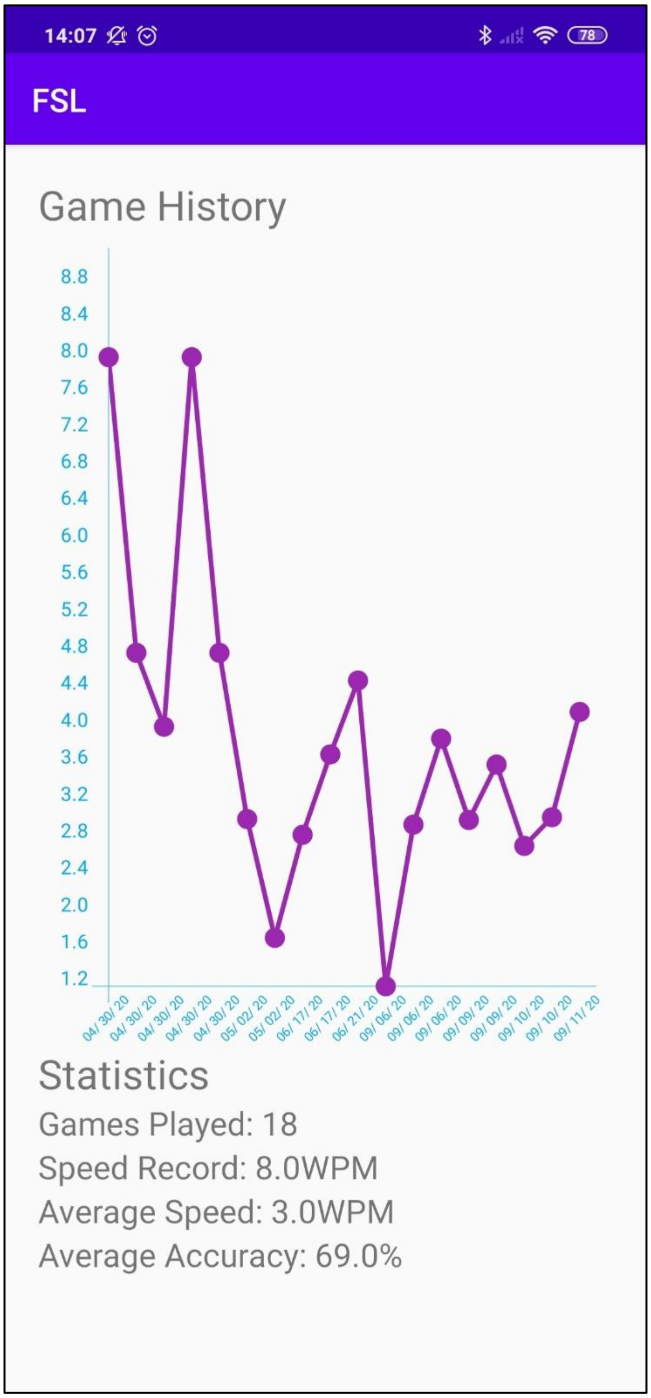
He who does not know how to look back at where he came from will never

Speed : 4.31WPM

Accuracy : 59.24%

Label	Confidence
1. N	85.71%
2. S	14.29%
3. -	0%

Statistics Activity Layout



Game History

Statistics

Games Played: 18

Speed Record: 8.0WPM

Average Speed: 3.0WPM

Average Accuracy: 69.0%

MainMenu.java

```java
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;

public class MainMenu extends AppCompatActivity {

    public static String KEY;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main_menu);
    }
    /** Called when the user taps the GestureClassifying button */
    public void gestureClassifer(View view) {
        // Do something in response to button
        Intent intent = new Intent(this, DeviceListActivity.class);

        // Attach the key value pair using putExtra to this intent
        String key = "classify";
        intent.putExtra(KEY, key);

        // Starting the activity
        startActivity(intent);
    }

    /** Called when the user taps the GestureTyping button */
    public void gestureTyping(View view) {
        // Creating and intializing Intent object
        Intent intent = new Intent(this, DeviceListActivity.class);

        // Attach the key value pair using putExtra to this intent
        String key= "type";
        intent.putExtra(KEY, key);

        // Starting the Activity
        startActivity(intent);
    }

    /** Called when the user taps the ShowStats button */
    public void showStats(View view) {
         Intent intent = new Intent(this, StatsActivity.class);
         startActivity(intent);
    }
}
```

MainMenu.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin" >


    <Button
        android:id="@+id/btn_gestureClassifier"
        android:layout_width="170dp"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="250dp"
        android:onClick="gestureClassifer"
        android:text="Gesture Classifier" />

    <Button
        android:id="@+id/btn_gestureTyping"
        android:layout_width="170dp"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="350dp"
        android:onClick="gestureTyping"
        android:text="Gesture Typing" />

    <Button
        android:id="@+id/btn_stats"
        android:layout_width="170dp"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="450dp"
        android:onClick="showStats"
        android:text="Stats" />
</RelativeLayout>
```

DeviceListActivity.java

```
import androidx.appcompat.app.AppCompatActivity;
import java.util.Set;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;
import android.widget.AdapterView.OnItemClickListener;

public class DeviceListActivity extends AppCompatActivity {

    private static final String TAG = "DeviceListActivity";

    //An EXTRA to take the device MAC to the next activity
    public static String EXTRA_DEVICE_ADDRESS;

    // textview for connection status
    TextView textConnectionStatus;
    ListView pairedListView;

    // Member fields
    private BluetoothAdapter mBtAdapter;
    private ArrayAdapter<String> mPairedDevicesArrayAdapter;

    // For determining where activity to go
    private static String key;

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        textConnectionStatus = (TextView) findViewById(R.id.connecting);
        textConnectionStatus.setTextSize(40);

        // Initialize array adapter for paired devices
        mPairedDevicesArrayAdapter = new ArrayAdapter<String>(this, R.layout.device_name);

        // Find and set up the ListView for paired devices
        pairedListView = (ListView) findViewById(R.id.paired_devices);
        pairedListView.setAdapter(mPairedDevicesArrayAdapter);

        // Find and set up the ListView for paired devices
        ListView pairedListView = (ListView) findViewById(R.id.paired_devices);
        pairedListView.setAdapter(mPairedDevicesArrayAdapter);
        pairedListView.setOnItemClickListener(mDeviceClickListener);

    }

    @Override
    public void onResume()
    {
        super.onResume();
        //It is best to check BT status at onResume in case something has changed while app
was paused etc
```

```
        checkBTState();

        // clears the array so items aren't duplicated when resuming from onPause
        mPairedDevicesArrayAdapter.clear();

        //makes the textview blank
        textConnectionStatus.setText(" ");

        // Get the local Bluetooth adapter
        mBtAdapter = BluetoothAdapter.getDefaultAdapter();

        // Get a set of currently paired devices and append to pairedDevices list
        Set<BluetoothDevice> pairedDevices = mBtAdapter.getBondedDevices();

        // Add previously paired devices to the array
        if (pairedDevices.size() > 0) {
            //make title viewable
            findViewById(R.id.title_paired_devices).setVisibility(View.VISIBLE);
            for (BluetoothDevice device : pairedDevices) {
                mPairedDevicesArrayAdapter.add(device.getName() + "\n" +
device.getAddress());
            }
        } else {
            mPairedDevicesArrayAdapter.add("No devices paired");
        }
    }

    //method to check if the device has Bluetooth and if it is on.
    //Prompts the user to turn it on if it is off
    private void checkBTState()
    {
        // Check device has Bluetooth and that it is turned on
        mBtAdapter=BluetoothAdapter.getDefaultAdapter(); // CHECK THIS OUT THAT IT WORKS!!!
        if(mBtAdapter==null) {
            Toast.makeText(getBaseContext(),
                    "Device does not support Bluetooth", Toast.LENGTH_SHORT).show();
            finish();
        } else {
            if (!mBtAdapter.isEnabled()) {
                //Prompt user to turn on Bluetooth
                Intent enableBtIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
                startActivityForResult(enableBtIntent, 1);
            }
        }
    }

    private OnItemClickListener mDeviceClickListener = new OnItemClickListener()
    {
        public void onItemClick(AdapterView<?> av, View v, int arg2, long arg3)
        {
            //textConnectionStatus.setText("Connecting...");
            Toast.makeText(getBaseContext(),
                    "Connecting....", Toast.LENGTH_SHORT).show();

            // Get the device MAC address, which is the last 17 chars in the View
            String info = ((TextView) v).getText().toString();
            String address = info.substring(info.length() - 17);

            // Get the current intent
            Intent intent = getIntent();

            // Get the attached extras from the intent
            key = intent.getStringExtra(MainMenu.KEY);

            // Variables for key comparison
            String classify = "classify";
            String type  = "type";

            // The key determines where will be the next activity
            if(classify.equals(key)){
                // Make an intent to start next activity while taking an extra which is the
                   MAC address.
                Intent intent1 = new Intent(DeviceListActivity.this, MainActivity.class);
                intent1.putExtra(EXTRA_DEVICE_ADDRESS, address);
                startActivity(intent1);
            } else if (type.equals(key)){
                // Make an intent to start next activity while taking an extra which is the
                   MAC address.
                Intent intent2 = new Intent(DeviceListActivity.this, GestureTyping.class);
                intent2.putExtra(EXTRA_DEVICE_ADDRESS, address);
                startActivity(intent2);
            } else {
                Toast.makeText(getBaseContext(), key, Toast.LENGTH_SHORT).show();
            }
        }
    };
}
```

DeviceListActivity.xml

```
<?xml version="1.0" encoding="utf-8"?>
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="18sp"
    android:padding="5dp">

</TextView>
```

MainActivity.java;

```java
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import android.annotation.SuppressLint;
import android.os.Bundle;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.text.DecimalFormat;
import java.util.UUID;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothSocket;
import android.content.Intent;
import android.os.Handler;
import android.util.Log;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    private static final String TAG = "MainActivity";
    private static final String FILE_NAME = "dataset.txt";

    // Text Views
    TextView letter, label1, label2, label3, confidence1, confidence2, confidence3;
    TextView flex1, flex2, flex3, flex4, flex5, gyroX, gyroY, gyroZ, accX, accY, accZ;
    TextView txtString, txtStringLength;

    Handler bluetoothIn;
    final int handlerState = 0;                        //used to identify handler message

    private BluetoothAdapter bluetoothAdapter = null;
    private BluetoothSocket btSocket = null;

    // Contains the string received from Bluetooth
    private StringBuilder recDataString = new StringBuilder();

    // Create Thread
    private ConnectedThread mConnectedThread;

    // SPP UUID service - this should work for most devices
    private static final UUID BTMODULEUUID = UUID.fromString("00001101-0000-1000-8000-
00805F9B34FB");

    // String for MAC address
    private static String address;
    public int counter = 0;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main2);

        // Link the buttons and textViews to respective views
        txtString = findViewById(R.id.txtString);
        txtStringLength = findViewById(R.id.testView1);
        flex1 = findViewById(R.id.FlexSensor1);
        flex2 = findViewById(R.id.FlexSensor2);
        flex3 = findViewById(R.id.FlexSensor3);
        flex4 = findViewById(R.id.FlexSensor4);
        flex5 = findViewById(R.id.FlexSensor5);
        gyroX = findViewById(R.id.GyroX);
        gyroY = findViewById(R.id.GyroY);
        gyroZ = findViewById(R.id.GyroZ);
        accX = findViewById(R.id.AccelerometerX);
        accY = findViewById(R.id.AccelerometerY);
        accZ = findViewById(R.id.AccelerometerZ);
        letter = findViewById(R.id.text_letter);
        label1 = findViewById(R.id.text_label1);
        label2 = findViewById(R.id.text_label2);
        label3 = findViewById(R.id.text_label3);
        confidence1 = findViewById(R.id.text_confidence1);
        confidence2 = findViewById(R.id.text_confidence2);
        confidence3 = findViewById(R.id.text_confidence3);

        bluetoothIn = new Handler() {
            @SuppressLint("HandlerLeak")
            public void handleMessage(android.os.Message msg) {
                if (msg.what == handlerState) {
                    String readMessage = (String) msg.obj;
                    recDataString.append(readMessage);
                    int endOfLineIndex = recDataString.indexOf("~");
                    if (endOfLineIndex > 0) {
                        String dataInPrint = recDataString.substring(0, endOfLineIndex);
                        txtString.setText("Data Received = " + dataInPrint);
                        int dataLength = dataInPrint.length();
                        txtStringLength.setText("String Length = " +
String.valueOf(dataLength));

                        //if it starts with # we know it is what we are looking for
                        if (recDataString.charAt(0) == '#')
                        {
                            // Parse the packet String into array of characters
                            String packet = recDataString.substring(1, endOfLineIndex);
                            String[] values = packet.split(",");

                            // Store the values in the array into variables
                            String f1 = values[0];
                            String f2 = values[1];
                            String f3 = values[2];
                            String f4 = values[3];
                            String f5 = values[4];
                            String gx = values[5];
                            String gy = values[6];
                            String gz = values[7];
                            String ax = values[8];
                            String ay = values[9];
                            String az = values[10];

                            // Set the textViews and show their values
                            flex1.setText(" F1 = " + f1);
                            flex2.setText(" F2 = " + f2);
                            flex3.setText(" F3 = " + f3);
                            flex4.setText(" F4 = " + f4);
                            flex5.setText(" F5 = " + f5);
                            gyroX.setText(" Gx = " + gx);
                            gyroY.setText(" Gy = " + gy);
                            gyroZ.setText(" Gz = " + gz);
                            accX.setText(" Ax = " + ax);
                            accY.setText(" Ay = " + ay);
                            accZ.setText(" Az = " + az);

                            // Create a classifier object
                            Classifier classifier = new Classifier();
                            // Prepare testFeatures inputs in the object
                            classifier.setFlex1(f1);
                            classifier.setFlex2(f2);
                            classifier.setFlex3(f3);
                            classifier.setFlex4(f4);
                            classifier.setFlex5(f5);
                            classifier.setGyroX(gx);
                            classifier.setGyroY(gy);
                            classifier.setGyroZ(gz);
                            classifier.setAccX(ax);
                            classifier.setAccY(ay);
                            classifier.setAccZ(az);

                            // Prepare the trained features and label values
                            BufferedReader reader = null;
                            try {
                                String line;
                                // Get it from the text file and store it in a List
                                reader = new BufferedReader(
                                        new InputStreamReader(
                                                getAssets().open(FILE_NAME), "UTF-8"));

                                // Do reading, usually loop until end of file reading
                                while ((line = reader.readLine()) != null) {
                                    String[] split = line.split(",");
                                    double[] feature = new double[split.length - 1];

                                    // Store each line in an array
                                    for (int i = 0; i < split.length - 1; i++)
                                        feature[i] = Double.parseDouble(split[i]);

                                    // Add the trained features and labels in the object
                                    classifier.AddTrainedFeatures(feature);
                                    classifier.AddTrainedLabel(split[feature.length]);
                                }
                            } catch (IOException e) {
                                Log.e(TAG, "Error: " + e);
                            } finally {
                                if (reader != null) {
                                    try {
                                        reader.close();
                                    } catch (IOException e) {
                                        Log.e(TAG, "Error: " + e);
                                    }
                                }
                            }

                            // Process the test inputs and trained inputs in the object
                            classifier.Classify();

                            // Get the label and confidence output from the object
                            // Print the text to UI
                            letter.setText(classifier.GetLabel(0));
                            label1.setText("1.    " + classifier.GetLabel(0));
                            label2.setText("2.    " + classifier.GetLabel(1));
                            label3.setText("3.    " + classifier.GetLabel(2));
                            DecimalFormat df2 = new DecimalFormat("#.##");
                            confidence1.setText(df2.format(classifier.GetConfidence(0)) +
"%");
                            confidence2.setText(df2.format(classifier.GetConfidence(1)) +
"%");
                            confidence3.setText(df2.format(classifier.GetConfidence(2)) +
"%");
```

```
                            Log.i(TAG, "tag_handler");

                        }
                        //clear all string data
                        recDataString.delete(0, recDataString.length());
                    }
                }
            }
        };

        bluetoothAdapter = BluetoothAdapter.getDefaultAdapter();   // get Bluetooth adapter
        checkBTState();
    }

    @Override
    public void onResume() {
        super.onResume();
        Log.i(TAG, "tag_onResume");
        //Get MAC address from DeviceListActivity via intent
        Intent intent = getIntent();

        //Get the MAC address from the DeviceListActivity via EXTRA
        address = intent.getStringExtra(DeviceListActivity.EXTRA_DEVICE_ADDRESS);

        //create device and set the MAC address
        BluetoothDevice device = bluetoothAdapter.getRemoteDevice(address);

        try {
            btSocket = createBluetoothSocket(device);
        } catch (IOException e) {
            Toast.makeText(getBaseContext(), "Socket creation failed",
Toast.LENGTH_LONG).show();
        }
        // Establish the Bluetooth socket connection.
        try
        {
            btSocket.connect();
        } catch (IOException e) {
            try
            {
                btSocket.close();
            } catch (IOException e2)
            {
                //insert code to deal with this
            }
        }
        mConnectedThread = new ConnectedThread(btSocket);
        mConnectedThread.start();

        //I send a character when resuming.beginning transmission to check device is
connected
        //If it is not an exception will be thrown in the write method and finish() will be
called
        mConnectedThread.write("x");
    }

    @Override
    public void onPause()
    {
        Log.i(TAG, "tag_onPause");
        super.onPause();
        try
        {
            //Don't leave Bluetooth sockets open when leaving activity
            btSocket.close();
        } catch (IOException e2) {
            //insert code to deal with this
        }
    }

    private BluetoothSocket createBluetoothSocket(BluetoothDevice device) throws IOException
{
        //creates secure outgoing connection with BT device using UUID
        return  device.createRfcommSocketToServiceRecord(BTMODULEUUID);
    }

    //Checks that the Android device Bluetooth is available and prompts to be turned on if
off
    private void checkBTState() {

        if(bluetoothAdapter==null) {
            Toast.makeText(getBaseContext(),
                    "Device does not support bluetooth", Toast.LENGTH_LONG).show();
        } else {
            if (bluetoothAdapter.isEnabled()) {
            } else {
                Intent enableBtIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
                startActivityForResult(enableBtIntent, 1);
            }
        }
    }

    //create new class for connect thread
    private class ConnectedThread extends Thread {
        private final InputStream mmInStream;
        private final OutputStream mmOutStream;

        //creation of the connect thread
        public ConnectedThread(BluetoothSocket socket) {
            InputStream tmpIn = null;
            OutputStream tmpOut = null;

            try {
                //Create I/O streams for connection
                tmpIn = socket.getInputStream();
                tmpOut = socket.getOutputStream();
            } catch (IOException e) { }

            mmInStream = tmpIn;
            mmOutStream = tmpOut;
        }

        public void run() {
            byte[] buffer = new byte[256];
            int bytes;

            // Keep looping to listen for received messages
            while (true) {
                try {
                    Log.i(TAG, "tag_firstThread");
                    //read bytes from input buffer
                    bytes = mmInStream.read(buffer);
                    String readMessage = new String(buffer, 0, bytes);
                    // Send the obtained bytes to the UI Activity via handler
                    bluetoothIn.obtainMessage(handlerState, bytes, -1,
readMessage).sendToTarget();
                    Log.i(TAG, "tag_secondThread");
                } catch (IOException e) {
                    break;
                }
            }
        }
        //write method
        public void write(String input) {
            //converts entered String into bytes
            byte[] msgBuffer = input.getBytes();
            try {
                //write bytes over BT connection via outstream
                mmOutStream.write(msgBuffer);
            } catch (IOException e) {
                //if you cannot write, close the application
                Toast.makeText(getBaseContext(), "Connection Failure",
Toast.LENGTH_LONG).show();
                finish();

            }
        }
    }
}
```

MainActivity.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin" >


    <TextView
        android:id="@+id/text_letter"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:text="A"
        android:textSize="200dp"
        />
    <!--Labels-->
    <TextView
        android:id="@+id/text_label"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/text_letter"
        android:layout_marginTop="10dp"
        android:layout_marginLeft="90dp"
        android:text="Label" />

    <TextView
        android:id="@+id/text_label1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/text_label"
        android:layout_marginTop="10dp"
        android:layout_marginLeft="90dp"
        android:text="1.    ?" />

    <TextView
        android:id="@+id/text_label2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/text_label1"
        android:layout_marginTop="10dp"
        android:layout_marginLeft="90dp"
        android:text="2.    ?" />

    <TextView
        android:id="@+id/text_label3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/text_label2"
        android:layout_marginTop="10dp"
        android:layout_marginLeft="90dp"
        android:text="3.    ?" />


    <!--Confidence-->
    <TextView
        android:id="@+id/text_confidence"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/text_letter"
        android:layout_marginTop="10dp"
        android:layout_alignParentRight="true"
        android:layout_marginRight="90dp"
        android:text="Confidence" />

    <TextView
        android:id="@+id/text_confidence1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/text_confidence"
        android:layout_marginTop="10dp"
        android:layout_alignParentRight="true"
        android:layout_marginRight="105dp"
        android:text="33.33%" />

    <TextView
        android:id="@+id/text_confidence2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/text_label1"
        android:layout_marginTop="10dp"
        android:layout_alignParentRight="true"
        android:layout_marginRight="105dp"
        android:text="33.33%" />

    <TextView
        android:id="@+id/text_confidence3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/text_label2"
        android:layout_marginTop="10dp"
        android:layout_alignParentRight="true"
        android:layout_marginRight="105dp"
        android:text="33.33%" />
    <!--Gloves' Sensor Values-->
    <TextView
        android:id="@+id/GloveSensorValues"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/text_label3"
        android:layout_marginTop="20dp"
        android:layout_centerHorizontal="true"
        android:text="Gloves' Sensor Values" />
    <!--Flex Sensor-->
    <TextView
        android:id="@+id/FlexSensor"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/GloveSensorValues"
        android:layout_marginTop="20dp"
        android:layout_marginLeft="40dp"
        android:text="Flex Sensors" />

    <TextView
        android:id="@+id/FlexSensor1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/FlexSensor"
        android:layout_marginTop="10dp"
        android:layout_marginLeft="40dp"
        android:text="F1 = ??????.??" />

    <TextView
        android:id="@+id/FlexSensor2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/FlexSensor1"
        android:layout_marginTop="10dp"
        android:layout_marginLeft="40dp"
        android:text="F2 = ??????.??" />

    <TextView
        android:id="@+id/FlexSensor3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/FlexSensor2"
        android:layout_marginTop="10dp"
        android:layout_marginLeft="40dp"
        android:text="F3 = ??????.??" />

    <TextView
        android:id="@+id/FlexSensor4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/FlexSensor3"
        android:layout_marginTop="10dp"
        android:layout_marginLeft="40dp"
        android:text="F4 = ??????.??" />

    <TextView
        android:id="@+id/FlexSensor5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/FlexSensor4"
        android:layout_marginTop="10dp"
        android:layout_marginLeft="40dp"
        android:text="F5 = ??????.??" />
    <!--GyroScope-->
    <TextView
        android:id="@+id/Gyroscope"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/GloveSensorValues"
        android:layout_marginTop="20dp"
        android:layout_centerHorizontal="true"
        android:text="Gyroscope" />

    <TextView
        android:id="@+id/GyroX"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/Gyroscope"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="10dp"
        android:text="Gx = ?????" />

    <TextView
        android:id="@+id/GyroY"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/GyroX"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="10dp"
        android:text="Gy = ?????" />

    <TextView
        android:id="@+id/GyroZ"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/GyroY"
        android:layout_marginTop="10dp"
        android:layout_centerHorizontal="true"
        android:text="Gz = ?????" />

<!--Accelerometer-->

    <TextView
        android:id="@+id/Accelerometer"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/GloveSensorValues"
        android:layout_toRightOf="@id/Gyroscope"
        android:layout_marginTop="20dp"
        android:layout_marginLeft="40dp"
        android:layout_centerHorizontal="true"
        android:text="Accelerometer" />

    <TextView
        android:id="@+id/AccelerometerX"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/Accelerometer"
        android:layout_toRightOf="@id/Gyroscope"
        android:layout_marginTop="10dp"
        android:layout_marginLeft="40dp"
        android:text="Ax = ?????" />
    <TextView
        android:id="@id/AccelerometerY"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/AccelerometerX"
        android:layout_toRightOf="@id/Gyroscope"
        android:layout_marginTop="10dp"
        android:layout_marginLeft="40dp"
        android:text="Ay = ?????" />

    <TextView
        android:id="@+id/AcceleromterZ"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/AccelerometerY"
        android:layout_toRightOf="@id/Gyroscope"
        android:layout_marginTop="10dp"
        android:layout_marginLeft="40dp"
        android:text="Az = ?????" />
    <TextView
        android:id="@+id/testView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@+id/txtString"
        android:text=""
        android:textSize="15sp" />

    <TextView
        android:id="@+id/txtString"
        android:layout_width="wrap_content"
        android:layout_height="50dp"
        android:layout_alignLeft="@+id/testView1"
        android:layout_alignParentBottom="true"
        android:text=""
        android:textSize="15sp" />

</RelativeLayout>
```

GestureTyping.java

```
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.annotation.SuppressLint;
import android.app.Dialog;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothSocket;
import android.content.Intent;
import android.graphics.Color;
import android.os.Bundle;
import android.os.Handler;
import android.text.SpannableString;
import android.text.Spanned;
import android.text.style.ForegroundColorSpan;
import android.util.Log;
import android.widget.TextView;
import android.widget.Toast;
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.text.DecimalFormat;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;
import java.util.Locale;
import java.util.Random;
import java.util.UUID;

public class GestureTyping extends AppCompatActivity implements
CompleteDialog.ExampleDialogListener {

    // Log Tag
    private static final String TAG = "GestureTyping";
    private static final String FILE_NAME = "gameData.txt";
    private static final String DATASET = "dataset.txt";
    // Text Views
    TextView letter, label1, label2, label3;
    TextView confidence1, confidence2, confidence3;
    TextView sentence, timer;

    // Handler
    Handler bluetoothIn;

    // Use to identify handler message
    final int handlerState = 0;

    // Bluetooth
    private BluetoothAdapter btAdapter = null;
    private BluetoothSocket btSocket = null;

    // Contains the string received from Bluetooth
    private StringBuilder recDataString = new StringBuilder();

    // Threads
    private GestureTyping.ConnectedThread mConnectedThread;
    private Timer timerThread;

    // SPP UUID service - this should work for most devices
    private static final UUID BTMODULEUUID = UUID.fromString("00001101-0000-1000-8000-
00805F9B34FB");

    // String for MAC address
    private static String address;

    // Serves as pointer of the current character the user is typing
    private int counter;

    // Game timer
    private int seconds;

    // Checks if the game is complete
    private boolean isComplete;

    // Sentence to type on
    private String givenSentence;

    // TypeSpeed and Accuracy
    private double speed, accuracy;

    // User's total character input
    private int totalEntries;

    @Override
    protected void onSaveInstanceState(@NonNull Bundle outState) {
        super.onSaveInstanceState(outState);
        outState.putInt("counter", counter);
        outState.putInt("seconds", seconds);
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_gesture_typing);

        if (savedInstanceState != null) {
            counter = savedInstanceState.getInt("counter");
            seconds = savedInstanceState.getInt("seconds");
        }

        // Link the buttons and textViews to respective views
```

```
timer = findViewById(R.id.timer);
sentence = findViewById(R.id.sentence);
letter = findViewById(R.id.text_letter);
label1 = findViewById(R.id.text_label1);
label2 = findViewById(R.id.text_label2);
label3 = findViewById(R.id.text_label3);
confidence1 = findViewById(R.id.text_confidence1);
confidence2 = findViewById(R.id.text_confidence2);
confidence3 = findViewById(R.id.text_confidence3);

initializeActivity();

bluetoothIn = new Handler() {
    @SuppressLint("HandlerLeak")
    public void handleMessage(android.os.Message msg) {
        if (msg.what == handlerState) {
            String readMessage = (String) msg.obj;
            recDataString.append(readMessage);
            int endOfLineIndex = recDataString.indexOf("~");
            if (endOfLineIndex > 0) {
                //if it starts with # we know it is what we are looking for
                if (recDataString.charAt(0) == '#')
                {
                    // Parse the packet String into array of characters
                    String packet = recDataString.substring(1, endOfLineIndex);
                    String[] values = packet.split(",");

                    // Store the values in the array into variables
                    String f1 = values[0];
                    String f2 = values[1];
                    String f3 = values[2];
                    String f4 = values[3];
                    String f5 = values[4];
                    String gx = values[5];
                    String gy = values[6];
                    String gz = values[7];
                    String ax = values[8];
                    String ay = values[9];
                    String az = values[10];

                    // Create a classifier object
                    Classifier classifier = new Classifier();

                    // Prepare testFeatures inputs in the object
                    classifier.setFlex1(f1);
                    classifier.setFlex2(f2);
                    classifier.setFlex3(f3);
                    classifier.setFlex4(f4);
                    classifier.setFlex5(f5);
                    classifier.setGyroX(gx);
                    classifier.setGyroY(gy);
                    classifier.setGyroZ(gz);
                    classifier.setAccX(ax);
                    classifier.setAccY(ay);
                    classifier.setAccZ(az);

                    // Prepare the trained features and label values
                    BufferedReader reader = null;
                    try {
                        String line;
                        // Get it from the text file and store it in a List
                        reader = new BufferedReader(
                                new InputStreamReader(
                                        getAssets().open(DATASET), "UTF-8"));
                                // Do reading, usually loop until end of file reading
                                while ((line = reader.readLine()) != null) {
                                    String[] split = line.split(",");
                                    double[] feature = new double[split.length - 1];

                                    // Store each line in an array
                                    for (int i = 0; i < split.length - 1; i++)
                                        feature[i] = Double.parseDouble(split[i]);

                                    // Add the trained features and labels in the object
                                    classifier.AddTrainedFeatures(feature);
                                    classifier.AddTrainedLabel(split[feature.length]);
                                }
                            } catch (IOException e) {
                                Log.e(TAG, "Error: " + e);
                            } finally {
                                if (reader != null) {
                                    try {
                                        reader.close();
                                    } catch (IOException e) {
                                        Log.e(TAG, "Error: " + e);
                                    }
                                }
                            }

                            // Process the test inputs and trained inputs in the object
                            classifier.Classify();

                            // Get the label and confidence output from the object
                            // Print the text to UI
                            letter.setText(classifier.GetLabel(0));
                            label1.setText("1.    " + classifier.GetLabel(0));
                            label2.setText("2.    " + classifier.GetLabel(1));
                            label3.setText("3.    " + classifier.GetLabel(2));
                            DecimalFormat df2 = new DecimalFormat("#.##");
                            confidence1.setText(df2.format(classifier.GetConfidence(0)) +
"%");

                            confidence2.setText(df2.format(classifier.GetConfidence(1)) +
"%");

                            confidence3.setText(df2.format(classifier.GetConfidence(2)) +
"%");


                            //======================FOR TYPING GAME=====================//
                            Log.i(TAG, "tag_typingGame");
                            String text = givenSentence;
                            String letter = classifier.GetLabel(0);
                            Log.i(TAG, "letter" + letter);
                            String space = "Space";
                            if(letter.length() > 1){
                                Log.i(TAG, "SPACE");
                                letter = " ";
                            }

                            //Convert Text to List of Characters
                            List<Character> charList = new ArrayList<>();
                            for (char ch : text.toCharArray()) {
                                charList.add(ch);
                            }

                            // This is used for coloring characters
                            SpannableString ss = new SpannableString(text);
                            ForegroundColorSpan fcsRed = new ForegroundColorSpan(Color.RED);
                            ForegroundColorSpan fcsGreen = new
ForegroundColorSpan(Color.GREEN);
```

```
                            // The users input
                            char input = letter.charAt(0);

                            // Compare the users input to the specific character
                            char currentCharacter = 
Character.toUpperCase(charList.get(counter));
                            if(input == currentCharacter){
                                // Change the color to green
                                ss.setSpan(fcsGreen, 0, counter + 1, 
Spanned.SPAN_EXCLUSIVE_EXCLUSIVE);
                                // Increment counter
                                counter++;
                                //Check if it is the last letter
                                if(counter == charList.size()){
                                    isComplete = true;
                                }
                            } else {
                                if(counter == 0){
                                    // Change the color to red
                                    ss.setSpan(fcsRed, counter, counter + 1, 
Spanned.SPAN_EXCLUSIVE_EXCLUSIVE);
                                } else{
                                    // Change the color of the previous letters to green and 
current letter to red
                                    ss.setSpan(fcsGreen, 0, counter, 
Spanned.SPAN_EXCLUSIVE_EXCLUSIVE);
                                    ss.setSpan(fcsRed, counter, counter + 1, 
Spanned.SPAN_EXCLUSIVE_EXCLUSIVE);
                                }
                            }

                            totalEntries++;
                            sentence.setText(ss);
                            Log.i(TAG, "tag_handler");

                        }
                        //clear all string data
                        recDataString.delete(0, recDataString.length());
                    }
                }
            }
        };

        btAdapter = BluetoothAdapter.getDefaultAdapter();       // get Bluetooth adapter
        checkBTState();
    }

    public void initializeActivity(){
        Log.i(TAG, "tag_initActivity");
        // initialize variables
        counter = 0;
        seconds = 0;
        isComplete = false;
        try {
            givenSentence = generateSentence();
        } catch (IOException e) {
            e.printStackTrace();
        }
        totalEntries = 0;

        // create new timer thread
        timerThread = new Timer();
        new Thread(timerThread).start();
        // prepare textviews
        timer.setText("0");
        sentence.setText(givenSentence);
        letter.setText("-");
        label1.setText("-");
        label2.setText("-");
        label3.setText("-");
        confidence1.setText("-");
        confidence2.setText("-");
        confidence3.setText("-");
    }

    private String generateSentence() throws IOException {
        String generatedSentence = "" ;

        // Generate random number from 0 to 5
        Random random = new Random();
        int fileNum = random.nextInt(5);
        String fileName = Integer.toString(fileNum) + ".txt";

        BufferedReader reader = null;
        try {
            String line;
            reader = new BufferedReader(new InputStreamReader(
                            getAssets().open(fileName), "UTF-8"));
            while ((line = reader.readLine()) != null) {
                generatedSentence = line;
            }
        } catch (IOException e) {
            Log.e(TAG, "Error: " + e);
        } finally {
            if (reader != null) {
                try {
                    reader.close();
                } catch (IOException e) {
                    Log.e(TAG, "Error: " + e);
                }
            }
        }
        return generatedSentence;
    }

    public void save(double speed, double accuracy, String date) {
        FileOutputStream fos = null;
        try {
            // to delete, use MODE_PRIVATE, then change to MODE_APPEND
            fos = openFileOutput(FILE_NAME, MODE_APPEND);
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
        String content =
                String.format("%.2f",speed) + "," + String.format("%.2f",accuracy) + "," +
date + "\n";
        byte[] bytesArray = content.getBytes();
        try {
            fos.write(bytesArray);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    public void openDialog(){
        CompleteDialog completeDialog = new CompleteDialog();
        completeDialog.setSpeed(speed);
        completeDialog.setAccuracy(accuracy);
        completeDialog.show(getSupportFragmentManager(), "complete dialog");
        Log.i(TAG, "tag_openDialog");
    }

    @Override
    public void onResume() {
        super.onResume();
        Log.i(TAG, "tag_onResume");
        //Get MAC address from DeviceListActivity via intent
        Intent intent = getIntent();

        //Get the MAC address from the DeviceListActivity via EXTRA
        address = intent.getStringExtra(DeviceListActivity.EXTRA_DEVICE_ADDRESS);

        //create device and set the MAC address
        BluetoothDevice device = btAdapter.getRemoteDevice(address);

        try {
            btSocket = createBluetoothSocket(device);
        } catch (IOException e) {
            Toast.makeText(getBaseContext(), "Socket creation failed",
Toast.LENGTH_LONG).show();
        }
        // Establish the Bluetooth socket connection.
        try
        {
            btSocket.connect();
        } catch (IOException e) {
            try
            {
                btSocket.close();
            } catch (IOException e2)
            {
                //insert code to deal with this
            }
        }

        mConnectedThread = new GestureTyping.ConnectedThread(btSocket);
        mConnectedThread.start();

        //I send a character when resuming.beginning transmission to check device is
connected
        //If it is not an exception will be thrown in the write method and finish() will be
called
        mConnectedThread.write("x");
    }

    @Override
    public void onPause() {
        Log.i(TAG, "tag_onPause");
        super.onPause();
        try
        {
            //Don't leave Bluetooth sockets open when leaving activity
            btSocket.close();
        } catch (IOException e2) {
            //insert code to deal with this
        }
    }

    private BluetoothSocket createBluetoothSocket(BluetoothDevice device) throws IOException
{
        //creates secure outgoing connection with BT device using UUID
        return  device.createRfcommSocketToServiceRecord(BTMODULEUUID);
    }
    //Checks that the Android device Bluetooth is available and prompts to be turned on if
off
    private void checkBTState() {
        if(btAdapter==null) {
            Toast.makeText(getBaseContext(), "Device does not support bluetooth",
Toast.LENGTH_LONG).show();
        } else {
            if (btAdapter.isEnabled()) {
            } else {
                Intent enableBtIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
                startActivityForResult(enableBtIntent, 1);
            }
        }
    }

    //create new class for connect thread
    private class ConnectedThread extends Thread {
        private final InputStream mmInStream;
        private final OutputStream mmOutStream;

        //creation of the connect thread
        public ConnectedThread(BluetoothSocket socket) {
            InputStream tmpIn = null;
            OutputStream tmpOut = null;

            try {
                //Create I/O streams for connection
                tmpIn = socket.getInputStream();
                tmpOut = socket.getOutputStream();
            } catch (IOException e) { }

            mmInStream = tmpIn;
            mmOutStream = tmpOut;
        }

        public void run() {
            byte[] buffer = new byte[256];
            int bytes;

            // Keep looping to listen for received messages
            while (true) {
                try {
                    Log.i(TAG, "tag_firstThread");
                    //read bytes from input buffer
                    bytes = mmInStream.read(buffer);
                    String readMessage = new String(buffer, 0, bytes);
                    // Send the obtained bytes to the UI Activity via handler
                    bluetoothIn.obtainMessage(handlerState, bytes, -1,
readMessage).sendToTarget();
                    Log.i(TAG, "tag_secondThread");
                } catch (IOException e) {
                    break;
                }
            }
        }
        //write method
        public void write(String input) {
            //converts entered String into bytes
            byte[] msgBuffer = input.getBytes();
            try {
                //write bytes over BT connection via outstream
                mmOutStream.write(msgBuffer);
            } catch (IOException e) {
                //if you cannot write, close the application
                Toast.makeText(getBaseContext(), "Connection Failure",
Toast.LENGTH_LONG).show();
                finish();

            }
        }
    }

    private class Timer implements Runnable{
        @Override
        public void run() {
            while(!isComplete){
                try {
                    Thread.sleep(1000);  //1000ms = 1 sec
                    runOnUiThread(new Runnable() {
                        @Override
                        public void run() {
                            seconds++;
                            timer.setText(String.valueOf(seconds));
                        }
                    });
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }

            // compute for speed and accuracy
            double wordChar = 5.00;
            double minute = 60.00;
            double wordsNum = (double) counter/wordChar;
            double time = (double) seconds/minute;
            speed = wordsNum/time;

            // compute for accuracy
            double acc = (double) counter/totalEntries;
            accuracy = acc*100;

            Log.i(TAG, "speed = " + speed);
            Log.i(TAG, "accuracy = " + accuracy);
            Log.i(TAG, "counter = " + counter);
            Log.i(TAG, "totalEntries = " + totalEntries);

            // save speed and accuracy
            String currentDate;
            currentDate = new SimpleDateFormat("MM/ dd/ yy", Locale.getDefault()).format(new
Date());
            Log.i(TAG, currentDate);
            save(speed,accuracy,currentDate);
            openDialog();
        }
    }
}
```

GestureTyping.xml

```xml
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin" >

    <TextView
        android:id="@+id/timer"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:text="0"
        android:textSize="50sp"
        android:layout_marginTop="50dp"
        />

    <TextView
        android:id="@+id/sentence"
        android:layout_width="300dp"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="150dp"
        android:textSize="25sp"
        android:text="THE QUICK BROWN FOX JUMP OVER THE LAZY DOG"/>

    <TextView
        android:id="@+id/text_letter"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:text="A"
        android:textSize="150dp"
        android:layout_marginTop="300dp"
        />
    <!--Labels-->
    <TextView
        android:id="@+id/text_label"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/text_letter"
        android:layout_marginTop="10dp"
        android:layout_marginLeft="90dp"
        android:text="Label" />

    <TextView
        android:id="@+id/text_label1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/text_label"
        android:layout_marginTop="10dp"
        android:layout_marginLeft="90dp"
        android:text="1.    ?" />

    <TextView
        android:id="@+id/text_label2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/text_label1"
        android:layout_marginTop="10dp"
        android:layout_marginLeft="90dp"
        android:text="2.    ?" />

    <TextView
        android:id="@+id/text_label3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/text_label2"
        android:layout_marginTop="10dp"
        android:layout_marginLeft="90dp"
        android:text="3.    ?" />


    <!--Confidence-->
    <TextView
        android:id="@+id/text_confidence"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/text_letter"
        android:layout_marginTop="10dp"
        android:layout_alignParentRight="true"
        android:layout_marginRight="90dp"
        android:text="Confidence" />

    <TextView
        android:id="@+id/text_confidence1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/text_confidence"
        android:layout_marginTop="10dp"
        android:layout_alignParentRight="true"
        android:layout_marginRight="105dp"
        android:text="33.33%" />

    <TextView
        android:id="@+id/text_confidence2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/text_label1"
        android:layout_marginTop="10dp"
        android:layout_alignParentRight="true"
        android:layout_marginRight="105dp"
        android:text="33.33%" />

    <TextView
        android:id="@+id/text_confidence3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/text_label2"
        android:layout_marginTop="10dp"
        android:layout_alignParentRight="true"
        android:layout_marginRight="105dp"
        android:text="33.33%" />

</RelativeLayout>
```

CompleteDialog.java

```java
import android.app.AlertDialog;
import android.app.Dialog;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatDialogFragment;
import java.text.DecimalFormat;

public class CompleteDialog extends AppCompatDialogFragment {

    private TextView txtSpeed, txtAccuracy;
    private Button btnMenu, btnNext;
    private ExampleDialogListener listener;
    private double speed, accuracy;

    @NonNull
    @Override
    public Dialog onCreateDialog(@Nullable Bundle savedInstanceState) {
        AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());
        LayoutInflater inflater = getActivity().getLayoutInflater();
        View view = inflater.inflate(R.layout.layout_dialog, null);

        // Build the dialog with the builder
        builder.setView(view);

        // link textViews and buttons
        txtSpeed = view.findViewById(R.id.txtSpeed);
        txtAccuracy = view.findViewById(R.id.txtAccuracy);
        btnMenu = view.findViewById(R.id.btnMenu);
        btnNext = view.findViewById(R.id.btnNext);

        // setup the dialog box
        DecimalFormat df2 = new DecimalFormat("#.##");
        txtSpeed.setText("Speed : " + df2.format(speed) + "WPM");
        txtAccuracy.setText("Accuracy : " + df2.format(accuracy) + "%");

        btnNext.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // refresh the GestureTyping activity
                listener.initializeActivity();
                // close the dialog
                getDialog().dismiss();
            }
        });
        btnMenu.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // return to menu
                Intent intent = new Intent(v.getContext(), MainMenu.class);
                startActivity(intent);
            }
        });

        return builder.create();
    }
    public void setSpeed(double speed){
        this.speed = speed;
    }

    public void setAccuracy(double accuracy){
        this.accuracy = accuracy;
    }

    @Override
    public void onAttach(Context context) {
        super.onAttach(context);
        try {
            listener = (ExampleDialogListener) context;
        } catch (ClassCastException e) {
            throw new ClassCastException(context.toString() +
                    "must implement ExampleDialogListener");
        }
    }

    public interface ExampleDialogListener {
        void initializeActivity();
    }
}
```

CompleteDialog.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="20dp">

    <TextView
        android:id="@+id/txtSpeed"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:text="Speed : 30 WPM"
        android:layout_centerHorizontal="true"
        android:textSize="30sp"
        />
    <TextView
        android:id="@+id/txtAccuracy"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:text="Accuracy : 65.35%"
        android:layout_below="@+id/txtSpeed"
        android:textSize="30sp"
        />
    <Button
        android:id="@+id/btnMenu"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="MENU"
        android:layout_below="@+id/btnNext"
        android:layout_marginTop="10dp"
        android:layout_centerHorizontal="true"
        />
    <Button
        android:id="@+id/btnNext"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="NEXT"
        android:layout_below="@+id/txtAccuracy"
        android:layout_marginTop="10dp"
        android:layout_centerHorizontal="true"
        />
</RelativeLayout>
```

StatsActivity.java

```
import androidx.appcompat.app.AppCompatActivity;

import android.graphics.Color;
import android.os.Bundle;
import android.widget.TextView;

import java.io.BufferedReader;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.ArrayList;
import java.util.List;

import lecho.lib.hellocharts.model.Axis;
import lecho.lib.hellocharts.model.AxisValue;
import lecho.lib.hellocharts.model.Line;
import lecho.lib.hellocharts.model.LineChartData;
import lecho.lib.hellocharts.model.PointValue;
import lecho.lib.hellocharts.model.Viewport;
import lecho.lib.hellocharts.view.LineChartView;

public class StatsActivity extends AppCompatActivity{

    private static final String TAG = "StatsActivity";
    TextView gamesPlayed, speedRecord, averageSpeed, averageAccuracy;
    private static final String FILE_NAME = "gameData.txt";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_stats);

        gamesPlayed = findViewById(R.id.gamesPlayed);
        speedRecord = findViewById(R.id.speedRecord);
        averageSpeed = findViewById(R.id.averageSpeed);
        averageAccuracy = findViewById(R.id.averageAccuracy);
        try {
            load();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    public void load() throws IOException {
        // Open the file
        FileInputStream fileInputStream = null;
        try {
            fileInputStream = openFileInput(FILE_NAME);
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
        // Get the data from the file
        InputStreamReader inputStreamReader = new InputStreamReader(fileInputStream);
        BufferedReader bufferedReader = new BufferedReader(inputStreamReader);
        List<String[]> gameDataList = new ArrayList<>();
        String string;
        while ((string = bufferedReader.readLine()) != null) {
            String[] split = string.split(",");
            String[] gameData = new String[split.length];
            for (int index = 0; index < split.length; index++) {
                gameData[index] = split[index];
            }
            gameDataList.add(gameData);
        }

        // Use the data to show statistics
        Statistics statistics = new Statistics();
        statistics.setSpeedAccuracyList(gameDataList);
        gamesPlayed.setText("Games Played: " + statistics.getGamesPlayedCount());
        speedRecord.setText("Speed Record: " + statistics.getSpeedRecord() + "WPM");
        averageSpeed.setText("Average Speed: " + statistics.getAverageSpeed() + "WPM");
        averageAccuracy.setText("Average Accuracy: " + statistics.getAverageAccuracy() +
"%");

        // Prepare data for x and y axis
        List<String> xData = new ArrayList<>();
        List<Float> yData = new ArrayList<>();
        for(int i = 0; i < gameDataList.size(); i++){
            String[] gameData = gameDataList.get(i);
            // store the date
            xData.add(gameData[2]);
            // store the speed
            yData.add(Float.parseFloat(gameData[0]));
        }

        // Declare a list which will hold the values in x and y axis
        List yAxisValues = new ArrayList();
        List xAxisValues = new ArrayList();

        // Declare and initialize the line that will show in the graph with set color
        Line line = new Line(yAxisValues).setColor(Color.parseColor("#9C27B0"));

        // Initialize the X and Y axis values lists
        for(int i = 0; i < xData.size(); i++){
            xAxisValues.add(i, new AxisValue(i).setLabel(xData.get(i)));
        }
        for (int i = 0; i < yData.size(); i++){
            yAxisValues.add(new PointValue(i, yData.get(i)));
        }

        // Declare a list that will hold the line of the graph chart
        List lines = new ArrayList();
        lines.add(line);

        // Add the graph line to the overall chart data
        LineChartData data = new LineChartData();
        data.setLines(lines);

        // View the android line chart
        LineChartView lineChartView = findViewById(R.id.graph);
        lineChartView.setLineChartData(data);

        // Show x and y axis
        Axis axis = new Axis();
        axis.setValues(xAxisValues);
        data.setAxisXBottom(axis);

        // Show y axis values
        Axis yAxis = new Axis();
        data.setAxisYLeft(yAxis);

        // change x axis values text size
        axis.setTextSize(8);

        // change text color of axis data
        axis.setTextColor(Color.parseColor("#03A9F4"));
        yAxis.setTextColor(Color.parseColor("#03A9F4"));

        // adjust y axis maximum value
        Viewport viewport = new Viewport(lineChartView.getMaximumViewport());
        Float top = (float) statistics.getSpeedRecord() + 1;
        viewport.top = top;
        lineChartView.setMaximumViewport(viewport);
        lineChartView.setCurrentViewport(viewport);

        // make the X axis labels tilt
        axis.setHasTiltedLabels(true);

    }
}
```

StatsActivity.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="20dp"
    tools:context=".StatsActivity">


    <lecho.lib.hellocharts.view.LineChartView
        android:id="@+id/graph"
        android:layout_width="match_parent"
        android:layout_height="500dip"
        android:layout_below="@+id/gameHistory"
        android:padding="10dp"
        />
    <TextView
        android:id="@+id/gameHistory"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Game History"
        android:textSize="25sp"
        />
    <TextView
        android:id="@+id/statistics"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Statistics"
        android:layout_below="@+id/graph"
        android:textSize="25sp"/>
    <TextView
        android:id="@+id/gamesPlayed"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Games Played : ?"
        android:layout_below="@+id/statistics"
        android:textSize="20sp"/>
    <TextView
        android:id="@+id/speedRecord"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Speed Record : ?"
        android:layout_below="@+id/gamesPlayed"
        android:textSize="20sp"/>
    <TextView
        android:id="@+id/averageSpeed"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Average Speed : ?"
        android:layout_below="@+id/speedRecord"
        android:textSize="20sp"/>
    <TextView
        android:id="@+id/averageAccuracy"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Average Accuracy : ?"
        android:layout_below="@+id/averageSpeed"
        android:textSize="20sp"/>



</RelativeLayout>
```

Classifier.java

```
package com.example.wingoodharry;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

public class Classifier {
    private static final String TAG = "Classifier";
    private double flex1;
    private double flex2;
    private double flex3;
    private double flex4;
    private double flex5;
    private double gyroX;
    private double gyroY;
    private double gyroZ;
    private double accX;
    private double accY;
    private double accZ;
    private double[] testFeatures;
    private List<double[]> trainedFeatures;
    private List<String> trainedLabel;
    private double[][] distanceList;
    private int k;
    private List<String> targetLabels;
    private String[] topThreeLabel;
    private int[] topThreeLabelCount;
    private double[] topThreeConfidence;

    public Classifier() {
        this.k = 7;
        this.trainedFeatures = new ArrayList<>();
        this.trainedLabel = new ArrayList<>();
        this.targetLabels = new ArrayList<>();
        this.topThreeLabel = new String[3];
        this.topThreeLabelCount = new int[3];
        this.topThreeConfidence = new double[3];
    }


    public void setFlex1(String flex1) {
        this.flex1 = Double.parseDouble(flex1);
    }

    public void setFlex2(String flex2) {
        this.flex2 = Double.parseDouble(flex2);
    }

    public void setFlex3(String flex3) {
        this.flex3 = Double.parseDouble(flex3);
    }

    public void setFlex4(String flex4) {
        this.flex4 = Double.parseDouble(flex4);
    }

    public void setFlex5(String flex5) {
        this.flex5 = Double.parseDouble(flex5);
    }

    public void setGyroX(String gyroX) {
        this.gyroX = Double.parseDouble(gyroX);
    }

    public void setGyroY(String gyroY) {
        this.gyroY = Double.parseDouble(gyroY);
    }

    public void setGyroZ(String gyroZ) {
        this.gyroZ = Double.parseDouble(gyroZ);
    }

    public void setAccX(String accX) {
        this.accX = Double.parseDouble(accX);
    }

    public void setAccY(String accY) {
        this.accY = Double.parseDouble(accY);
    }

    public void setAccZ(String accZ) {
        this.accZ = Double.parseDouble(accZ);
    }

    private void AddTestFeatures () {
        this.testFeatures = new double[]{
                flex1, flex2, flex3, flex4, flex5,
                gyroX, gyroY, gyroZ,
                accX, accY, accZ};
    }

    public void AddTrainedFeatures(double[] trainedFeatures) {
        this.trainedFeatures.add(trainedFeatures);
    }

    public void AddTrainedLabel(String trainedLabel) {
        this.trainedLabel.add(trainedLabel);
    }
```

```java
private void ComputeDistance() {
    int totalTrainedFeatures = this.trainedFeatures.size();
    this.distanceList = new double[totalTrainedFeatures][2];

    double[] indexTrainedFeature;
    double sum;
    double distance;

    for(int i = 0; i < totalTrainedFeatures; i++) {
        indexTrainedFeature = this.trainedFeatures.get(i);
        sum = 0;
        for(int j = 0; j < indexTrainedFeature.length; j++) {
            sum += Math.pow(indexTrainedFeature[j] - testFeatures[j], 2);
        }
        distance = Math.sqrt(sum);
        this.distanceList[i][0] = distance;
        this.distanceList[i][1] = i;
    }
}

private void SortDistanceList() {
    int distanceListLength = distanceList.length;

    for(int i = 0; i < distanceListLength; i++) {
        for(int j = 0; j < distanceListLength - 1; j++) {
            if(distanceList[j][0] > distanceList[j+1][0]) {
                //swap distance
                double temp = distanceList[j][0];
                distanceList[j][0] = distanceList[j+1][0];
                distanceList[j+1][0] = temp;
                //as well as label
                double temp2 = distanceList[j][1];
                distanceList[j][1] = distanceList[j+1][1];
                distanceList[j+1][1] = temp2;
            }
        }
    }
}

private void GetTopLabels() {
    for(int i = 0; i < this.k; i++){
        targetLabels.add(trainedLabel.get((int) distanceList[i][1]));
    }
}

private void GetMostNumberOfLabel() {
    Map<String, Integer> hashmap = new HashMap<String, Integer>();

    for(String label : targetLabels){
        if (hashmap.keySet().contains(label))
            hashmap.put(label, hashmap.get(label) + 1);
        else
            hashmap.put(label, 1);
    }

    for(int counter = 0; counter < 3; counter++) {
        String highestLabelCount = "";
        int labelCount = 0;

        for(Map.Entry<String, Integer> entry : hashmap.entrySet()){
            String key = entry.getKey();
            Integer value = entry.getValue();
            if (value > labelCount){
                highestLabelCount = key;
                labelCount = value;
                }
                else if(value == labelCount && highestLabelCount.compareTo(key) > 0)
                    highestLabelCount = key;
            }

            if(labelCount == 0){
                topThreeLabel[counter] = "-";
                topThreeLabelCount[counter] = 0;
            } else {
                topThreeLabel[counter] = highestLabelCount;
                topThreeLabelCount[counter] = labelCount;
            }

            hashmap.remove(highestLabelCount);
        }
    }

    private void ComputeConfidence() {
        for(int i = 0; i < 3; i++) {
            topThreeConfidence[i] = (topThreeLabelCount[i] / (double) this.k) * 100;
        }
    }

    public void Classify(){
        AddTestFeatures();
        ComputeDistance();
        SortDistanceList();
        GetTopLabels();
        GetMostNumberOfLabel();
        ComputeConfidence();
    }

    public double GetConfidence(int i) {
        return topThreeConfidence[i];
    }

    public String GetLabel(int i) {
        return topThreeLabel[i];
    }
}
```

Statistics.java

```java
import java.util.ArrayList;
import java.util.List;

public class Statistics {
    private List<Double> speedList;
    private List<Double> accuracyList;

    public Statistics() {
        this.speedList = new ArrayList<>();
        this.accuracyList = new ArrayList<>();
    }

    public void setSpeedAccuracyList(List<String[]> gameDataList) {
        String[] gameRecord;
        for(int i = 0; i < gameDataList.size(); i++) {
            gameRecord = gameDataList.get(i);
            speedList.add(Double.parseDouble(gameRecord[0]));
            accuracyList.add(Double.parseDouble(gameRecord[1]));
        }
    }

    public int getGamesPlayedCount() {
        return speedList.size();
    }

    public double getSpeedRecord() {
        double speedRecord = 0;
        for(int i = 0; i < speedList.size(); i++) {
            if (speedRecord < speedList.get(i)) {
                speedRecord = speedList.get(i);
            }
        }
        return speedRecord;
    }

    public double getAverageSpeed() {
        double averageSpeed = 0;
        int totalSum = 0;
        for(int i = 0; i < speedList.size(); i++) {
            totalSum += speedList.get(i);
        }
        averageSpeed = totalSum/speedList.size();
        return averageSpeed;
    }

    public double getAverageAccuracy() {
        double averageAccuracy = 0;
        int totalSum = 0;
        for(int i = 0; i < accuracyList.size(); i++) {
            totalSum += accuracyList.get(i);
        }
        averageAccuracy = totalSum/accuracyList.size();
        return averageAccuracy;
    }
}
```

dataset.txt

```
74736.84,25702.92,24857.14,23652.39,69313.19,408,-197,-368,14552,-2908,9084,N
45892.86,24548.97,14970.70,13155.96,66622.34,-1912,297,631,15604,-1576,7924,K
81509.44,26250.00,27863.13,14796.12,66622.34,-1225,-47,-47,15608,-3104,7156,R
60220.59,25595.24,13735.96,13001.82,61343.28,354,722,-482,15916,-3568,3964,U
57725.12,25275.59,19100.00,17579.62,70726.26,150,-337,52,15820,-2848,4876,V
71207.87,24960.94,27983.19,26250.00,65763.16,383,-125,-209,15236,-2620,7832,N
35311.48,22888.89,15386.14,13155.96,65763.16,1551,-535,129,15452,-3564,5144,K
56041.67,26029.41,26472.97,14623.55,55714.29,54,-155,389,15484,-3804,6996,R
52898.23,25275.59,12798.91,12648.65,58071.43,-445,29,-219,16136,-1224,3496,U
48937.50,24857.14,20936.77,18874.17,54750.00,141,167,-139,16384,-984,2372,V
75798.82,24346.15,28225.35,26585.37,67058.82,-142,-322,-367,15736,-2032,6800,N
49746.84,22610.29,14680.85,12748.64,67058.82,1022,548,862,16096,-1964,1816,K
50858.37,25702.92,26250.00,15147.35,71207.87,1861,-1543,837,16404,440,5636,R
63290.82,25170.16,13259.67,12900.00,49205.02,-321,-125,-78,15544,-2872,5980,U
62500.00,24651.16,21975.90,20438.80,50021.19,1711,-868,72,16208,-688,4908,V
74215.13,22982.67,26029.41,25920.00,70726.26,533,-545,-86,14876,-4004,7404,N
37551.38,21887.02,14396.55,13001.82,68852.46,2539,-313,1588,15592,-3828,4128,K
58071.43,25702.92,26250.00,14854.09,69313.19,-261,-529,-300,15976,-3280,2924,R
55714.29,25170.16,13155.96,13207.72,61343.28,395,-1482,-17,16024,-3848,800,U
58071.43,25920.00,21975.90,20194.95,73699.42,130,-1,-362,16200,-2908,1948,V
81509.44,22795.56,22065.22,21622.91,63290.82,-955,316,-17,15616,-1712,6968,N
44019.23,21362.56,14396.55,12748.64,65763.16,-74,-260,-100,16408,-2012,4428,K
78567.07,24346.15,24960.94,15029.35,57382.08,-59,13,-426,15508,-5356,3100,R
57382.08,25488.13,13311.81,12500.00,52302.63,93,-137,-197,15724,-4272,3296,U
48937.50,24548.97,21622.91,19175.95,53504.46,638,-141,38,15916,-4064,2324,V
73699.42,23076.92,26029.41,26472.97,67058.82,253,-320,-298,15920,-1680,5764,N
38466.90,21622.91,14340.34,13259.67,65763.16,234,-165,-300,16048,-2624,3724,K
46626.50,25702.92,26139.41,14912.28,71207.87,438,-969,516,15948,-3808,3448,R
47889.34,24857.14,13843.99,12698.56,59853.66,180,-217,-316,16372,-3244,2496,U
44476.74,24346.15,21710.53,19483.14,53504.46,-883,-81,77,16156,-3172,2976,V
73189.65,23076.92,28225.35,27272.73,64507.77,838,350,-433,15880,-3312,5564,N
35980.07,21887.02,13898.31,13053.01,66622.34,243,-248,-108,15900,-3580,3408,K
79722.22,26361.18,25381.58,14680.85,69313.19,-65,-153,-206,15852,-4440,3788,R
59853.66,23171.64,13575.42,12598.92,59853.66,303,-189,-303,15944,-4072,3232,U
56705.61,22888.89,21448.93,21535.71,53504.46,411,-559,-288,15652,-3460,2192,V
74736.84,22518.34,27041.10,26029.41,68852.46,-204,128,-255,15392,-2852,6824,N
35147.06,22335.77,12500.00,12304.27,65763.16,-3228,1796,16,16016,-2148,5416,K
45892.86,25170.16,26698.37,15206.69,70726.26,-128,47,-721,15416,-4412,5444,R
67945.95,24548.97,13682.25,12698.56,46135.45,261,-204,-223,15596,-4308,4572,U
59853.66,24346.15,21276.60,21276.60,50297.87,1554,-1447,-270,15536,-3560,3980,V
73189.65,22518.34,27041.10,27156.59,69313.19,620,-231,-284,15560,-2352,5292,N
36666.67,21887.02,13735.96,13416.67,64507.77,826,182,-375,16632,-2060,3288,K
56705.61,26585.37,27272.73,14796.12,77439.76,484,-378,7,16132,-3120,3560,R
61725.00,25702.92,13259.67,12950.82,70250.00,220,-534,52,16160,-2540,3568,U
60220.59,24753.88,20357.14,19483.14,64921.88,1310,-325,-755,15844,-2268,2872,V
77439.76,22702.70,26585.37,26585.37,63290.82,137,421,-182,16196,-2076,5372,N
39608.54,21975.90,12849.36,12401.79,65763.16,163,-326,-419,15976,-1664,4980,K
49746.84,25595.24,24651.16,15751.50,66622.34,14,259,-538,16056,-1980,4728,R
52898.23,23652.39,14117.65,13053.01,58071.43,436,-84,-248,16200,-2940,3608,U
44019.23,22982.67,18874.17,19328.86,58071.43,-1644,841,327,16464,-2544,3356,V
79722.22,25065.28,24548.97,22888.89,62500.00,1485,-80,293,16024,364,5072,N
49474.79,22888.89,13843.99,14228.57,68852.46,869,-658,404,15932,2268,6108,K
65763.16,24753.88,23171.64,13682.25,60591.13,715,-192,-634,15908,1024,3540,R
67058.82,24045.80,13735.96,12950.82,63290.82,432,82,-106,15236,3252,7744,U
61343.28,22610.29,20603.25,19560.81,63290.82,-55,216,-61,15268,4876,5460,V
72187.50,24651.16,24857.14,25811.17,57042.25,858,-277,-380,16108,-436,5448,N
46875.00,21191.04,13575.42,14340.34,58774.04,-547,125,116,14704,-2340,9136,K
73699.42,25065.28,21105.88,13682.25,62500.00,310,160,-183,15960,-4160,2796,R
59853.66,25170.16,13001.82,11826.92,59853.66,362,197,-196,16628,-1200,3788,U
59130.43,24346.15,21021.13,18431.37,61343.28,-206,182,338,16340,116,3724,V
64921.88,21887.02,25488.13,26361.18,61343.28,1484,-1645,-823,15236,-288,8244,N
42044.61,20194.95,12950.82,13207.72,57725.12,644,-1026,-354,15768,-1072,5500,K
```

```
55714.29,22426.83,13104.40,12500.00,60591.13,848,-913,-265,15972,-2736,5032,U
52302.63,21710.53,19954.44,17510.59,61343.28,60,-584,178,16156,-472,4744,V
46875.00,23458.65,22245.15,14796.12,64507.77,-77,-346,461,16652,-244,3536,R
71207.87,23266.83,24651.16,25488.13,59853.66,665,-615,-935,15780,-2124,5556,N
41415.44,22065.22,13053.01,13469.39,56705.61,283,-1237,-313,16272,-1732,2440,K
56041.67,23171.64,12849.36,12798.91,67058.82,547,-1326,-254,16364,-1964,2916,U
50021.19,24045.80,19483.14,18000.00,60220.59,4446,-1598,-1692,16636,-1192,3556,V
77439.76,24245.52,15326.08,13311.81,60220.59,196,-765,-148,14896,-4744,7156,R
61725.00,22518.34,22245.15,23555.27,55389.91,83,122,-302,13704,-1432,11048,N
40800.00,20438.80,12849.36,13469.39,56705.61,274,-141,66,16340,-1524,2916,K
59130.43,23171.64,13104.40,12698.56,55714.29,-1354,426,-240,15736,-2024,4768,U
54121.62,21710.53,21276.60,19483.14,59853.66,-101,-7,-139,16080,-788,5592,V
72685.71,25920.00,23076.92,15147.35,63290.82,-191,-155,-355,15964,-2608,496,R
64921.88,22426.83,24651.16,23652.39,55714.29,345,-653,-218,15104,-3784,7708,N
38281.25,21887.02,12798.91,13628.73,57725.12,2831,-1649,-73,16544,200,5520,K
59853.66,23848.10,13259.67,12950.82,48409.09,-129,305,-86,16056,-2160,3716,U
55714.29,22982.67,19405.83,18504.37,56705.61,88,-873,-240,16084,-1916,3652,V
46626.50,23171.64,19954.44,16836.10,61725.00,135,-189,-268,16292,-952,4496,R
68852.46,21448.93,24045.80,23750.00,59853.66,98,90,-70,15792,-1648,6388,N
42905.66,20769.23,12748.64,12798.91,57725.12,-1683,-366,-313,16232,484,5596,K
57382.08,22518.34,19795.92,15029.35,53504.46,303,-203,-94,16488,-184,3812,R
55714.29,22065.22,12598.92,13575.42,47378.05,398,-1222,37,16140,960,2672,U
56372.09,21622.91,19024.39,20769.23,48937.50,251,-344,-261,16532,1428,3316,V
67058.82,23266.83,24960.94,25275.59,53504.46,130,-269,-207,15336,-3244,7140,N
46626.50,21191.04,12950.82,13416.67,61343.28,970,-116,-283,16080,-1180,5152,K
62500.00,23362.50,12798.91,13053.01,47889.34,-693,-1372,-782,16092,-1488,4356,U
62500.00,23076.92,20603.25,18286.34,53504.46,68,-298,-336,16192,-564,4932,V
40197.84,23171.64,22518.34,15326.08,60591.13,180,35,-16,15864,-2392,5824,R
61343.28,22518.34,25381.58,24753.88,58071.43,1387,-339,96,15504,-3428,8400,N
43568.71,21276.60,14117.65,14284.35,60220.59,275,-3718,346,15352,-2096,7800,K
64507.77,23750.00,13259.67,12798.91,62110.55,520,-1219,92,16352,-2064,3432,U
63692.31,21798.56,18286.34,17579.62,56372.09,494,-342,379,16628,176,3188,V
40800.00,23458.65,19252.23,12450.81,64507.77,-253,-1169,179,16620,-2100,3996,R
64507.77,22154.96,25275.59,25381.58,59490.29,42,91,-335,16352,-1304,4612,N
40597.83,21276.60,12748.64,13001.82,63692.31,-1921,664,-14,16268,516,5708,K
42905.66,22888.89,17858.67,12648.65,63290.82,-3117,1424,-380,16688,152,3724,R
60220.59,23171.64,12598.92,12900.00,50021.19,386,-1632,-317,16408,-384,1772,U
65763.16,22795.56,18725.27,18286.34,54434.39,373,-443,267,16352,464,2324,V
62500.00,20686.05,26361.18,26585.37,60220.59,-186,-20,151,16064,-736,5380,N
37731.96,20114.41,13001.82,13952.83,60220.59,-207,-121,-465,16204,596,5100,K
68396.74,21887.02,12698.56,12849.36,52008.73,101,-181,50,16036,-352,2528,U
70726.26,21021.13,18949.11,18358.69,57725.12,519,-1165,-475,16360,716,3336,V
56705.61,22518.34,19175.95,14680.85,56705.61,1506,182,-50,15656,-1240,7144,R
78567.07,26361.18,24045.80,22610.29,55714.29,-3972,1163,595,15664,716,6524,N
46626.50,22795.56,13952.83,12798.91,55714.29,-7,-180,-300,15560,1924,6416,K
83365.39,26029.41,12900.00,12500.00,56041.67,349,-177,-469,16176,1316,5944,U
78567.07,24548.97,16574.07,16902.29,61343.28,-275,-316,-13,15800,520,6080,V
77439.76,23848.10,21448.93,13053.01,56705.61,-938,-370,-200,15916,1344,7292,R
86622.52,25488.13,25702.92,25065.28,52302.63,-496,417,100,15364,-432,8764,N
40800.00,21798.56,13898.31,12549.37,62110.55,407,-702,-813,16160,2284,5048,K
72685.71,24960.94,12950.82,12401.79,58071.43,435,-281,-419,16692,972,4340,U
75798.82,24753.88,19638.82,20438.80,61343.28,329,-150,-383,15996,1396,5980,V
73699.42,24548.97,21362.56,13104.40,61343.28,141,-760,-160,16504,1008,4672,R
73189.65,24145.41,22245.15,22245.15,48409.09,194,-318,-254,16012,152,5952,N
48409.09,22518.34,14340.34,13789.87,56705.61,1796,-3047,932,15452,1544,4260,K
75798.82,24346.15,13364.14,11640.63,57725.12,36,-490,-163,16128,-876,3840,U
61343.28,24447.30,19405.83,19100.00,62500.00,32,-359,-111,15788,1092,6356,V
73699.42,24447.30,22154.96,13104.40,52302.63,1143,-358,-286,15348,-12,6844,R
70726.26,22702.70,23076.92,23266.83,45892.86,-709,-1486,-124,15864,488,5608,N
35980.07,21362.56,14396.55,12450.81,50021.19,1353,-287,-358,16476,1108,3268,K
87299.99,23171.64,12950.82,11826.92,55714.29,-184,-463,243,16684,-588,3436,U
64507.77,23652.39,20438.80,19024.39,57382.08,-494,-450,-430,16088,640,5464,V
71207.87,23848.10,17788.46,11968.37,62500.00,-641,-784,87,15840,80,6696,R
68396.74,21362.56,24045.80,25702.92,55714.29,447,-113,-418,16124,-416,5004,N
37193.88,20357.14,12798.91,15029.35,52898.23,582,640,-532,15876,-244,5140,K
```

```
67945.95,24346.15,13259.67,12207.45,61343.28,-1233,-272,-274,16188,-792,5084,U
59130.43,23362.50,22518.34,19638.82,55714.29,-339,19,-186,16164,-524,3708,V
51717.39,24045.80,21535.71,14623.55,68852.46,-1543,-2248,-207,16076,-1912,4540,R
51717.39,23848.10,20936.77,14062.50,63290.82,460,-531,-67,16212,-2440,2884,R
71207.87,21975.90,24045.80,26139.41,53504.46,-2600,-627,41,16312,-1864,4724,N
47632.65,20357.14,12207.45,13735.96,60591.13,136,-590,-29,16396,532,3752,K
74215.13,23555.27,12598.92,13104.40,60591.13,443,211,-924,16484,-1084,3148,U
57042.25,23076.92,18577.68,19795.92,60591.13,-188,81,91,16376,-252,3480,V
84000.00,25381.58,29868.42,27389.50,71694.91,-935,-1487,1094,16072,-356,4688,N
84000.00,26250.00,13469.39,12159.29,65340.31,412,-1326,-1407,16320,-1532,2672,U
74736.84,25811.17,20034.25,18874.17,66622.34,696,-710,-176,16048,-140,4792,V
53811.66,22518.34,14173.00,12500.00,64921.88,217,-357,-366,15836,1092,6352,K
50021.19,23848.10,19795.92,12111.31,63290.82,292,325,-110,16136,452,6280,R
82120.26,24857.14,25702.92,24145.41,67945.95,1058,-1547,-787,15276,-1532,7924,N
76886.23,26361.18,13104.40,11733.45,63290.82,789,-636,-21,16552,-1956,4724,U
63290.82,24960.94,19024.39,18000.00,64507.77,-1079,-74,313,16720,-232,4852,V
43568.71,21362.56,12500.00,11780.10,65763.16,925,-29,-140,16564,972,4028,K
45892.86,23555.27,22065.22,13898.31,59853.66,70,-38,-115,17064,-1060,2128,R
77439.76,24245.52,27272.73,25702.92,64507.77,304,-209,-331,15456,-1260,7344,N
61725.00,25381.58,12798.91,11968.37,61343.28,1332,-493,-1720,15948,-1716,4268,U
60591.13,24651.16,21535.71,19483.14,62500.00,1091,-2198,-170,15724,-1084,6104,V
41623.61,21535.71,13469.39,12352.94,65763.16,-274,604,409,15360,520,5972,K
44247.11,24651.16,21105.88,13843.99,63290.82,302,638,-234,15940,-1668,4596,R
74215.13,23946.70,27743.73,25488.13,67945.95,393,-62,-169,16200,-1860,2280,N
66622.34,25170.16,13207.72,12207.45,56041.67,870,-707,-43,16328,-2448,2784,U
62500.00,23946.70,20275.86,19405.83,62500.00,42,-579,-143,16296,-804,2744,V
40397.11,21191.04,13898.31,12304.27,63290.82,-790,-1196,94,16724,-1072,5120,K
48937.50,24145.41,22982.67,12950.82,67058.82,-325,-668,-169,16396,-1932,408,R
71207.87,24651.16,27272.73,25702.92,68396.74,1708,-1913,-1044,16228,-2396,2756,N
74215.13,24753.88,12849.36,11826.92,55714.29,1135,-2725,-609,16356,-1900,2748,U
67945.95,24346.15,19100.00,17929.18,60591.13,96,154,-497,16292,-952,2664,V
40597.83,21021.13,13207.72,12450.81,67058.82,1239,-1945,-2377,16652,-1080,4948,K
44019.23,23555.27,21021.13,13952.83,59853.66,575,36,-185,16420,-1752,2920,R
64921.88,22982.67,23750.00,23458.65,65763.16,759,-479,117,16316,-1312,3844,N
48937.50,20603.25,13364.14,13001.82,60591.13,1192,-562,-121,16484,-440,2904,K
53811.66,23555.27,20357.14,12698.56,64507.77,-187,-318,-427,16300,-1168,3652,R
70250.00,23076.92,13053.01,12063.49,52302.63,235,-292,394,16464,-1232,1408,U
62893.40,22245.15,20520.83,19328.86,57725.12,-2475,487,-484,16112,-1016,4832,V
71207.87,23946.70,25811.17,25488.13,64921.88,925,-359,-439,16492,-928,3812,N
42905.66,21105.88,13207.72,12950.82,65340.31,552,-2572,435,16184,-460,4072,K
42905.66,24548.97,23750.00,14680.85,65340.31,111,417,-233,16280,-1348,1068,R
54750.00,23171.64,12900.00,11921.05,52302.63,-571,-1903,94,16180,-836,3236,U
51717.39,23076.92,18651.32,18577.68,51717.39,540,-1268,42,16256,-76,5620,V
71207.87,23848.10,23362.50,24045.80,64507.77,1376,-1663,-318,16044,-480,5268,N
42044.61,21710.53,13952.83,13575.42,61343.28,132,693,-378,16140,-2300,3524,K
46135.45,22795.56,20034.25,13843.99,63692.31,-487,52,-44,16360,-1312,3320,R
56041.67,23171.64,13416.67,12549.37,56705.61,79,-150,-108,16260,-1020,3488,U
55714.29,22795.56,19717.19,18431.37,54434.39,396,-1319,-185,16232,-384,4984,V
73699.42,25381.58,26029.41,26029.41,65763.16,324,-2021,1383,15744,-1516,5568,N
40000.00,22795.56,13628.73,13311.81,61725.00,3125,637,-1232,16896,-2020,2180,K
62893.40,24548.97,13053.01,12207.45,48148.15,-160,-982,-1123,16104,-1788,3628,U
40597.83,25170.16,22335.77,13259.67,63290.82,1365,-1080,-1016,15544,-1160,4300,R
55714.29,23750.00,21710.53,19795.92,64921.88,-960,-743,1301,16116,760,4912,V
73699.42,25381.58,26250.00,26029.41,70726.26,432,-1469,-521,16068,-1748,5100,N
69313.19,22610.29,12798.91,11873.91,53504.46,1536,156,-1323,15176,-2808,2332,U
63290.82,25381.58,22245.15,20194.95,56705.61,1981,-455,-889,15492,-1252,5180,V
46135.45,21798.56,12798.91,12549.37,65763.16,556,-773,-622,15448,-1052,4400,K
50858.37,25170.16,21710.53,14566.47,66622.34,-647,1753,-167,17332,-680,3584,R
74736.84,25065.28,26585.37,26139.41,66190.48,-551,-874,-97,16148,-1916,5556,N
43793.11,23362.50,13053.01,12698.56,61343.28,308,-577,-723,15480,-1264,5484,K
57725.12,23458.65,12648.65,13053.01,57042.25,-212,-918,-1297,15300,-2640,4180,U
52302.63,22610.29,17718.55,17788.46,54121.62,-1425,522,1310,17300,-852,6816,V
39031.69,23362.50,20357.14,13522.30,57725.12,-1930,223,1266,17688,-1376,5588,R
70250.00,24346.15,26472.97,26585.37,64507.77,879,294,-722,15888,-1076,5932,N
39608.54,22888.89,14117.65,13416.67,63692.31,679,-866,-531,14928,-96,6812,K
```

```
58774.04,23750.00,12950.82,11968.37,53504.46,123,-1355,-1222,14068,-2444,5968,U
50021.19,23362.50,19328.86,18000.00,54434.39,-2262,-795,962,14796,-1076,8488,V
42471.91,24651.16,22245.15,14509.62,58071.43,-228,815,151,16088,-1556,6364,R
74736.84,24857.14,25065.28,25170.16,63290.82,-19,54,435,18244,-1580,5128,N
39222.61,22335.77,13364.14,13155.96,67058.82,307,-310,638,15984,204,6776,K
51717.39,23362.50,12598.92,12063.49,52008.73,-998,-208,-948,15920,-2052,4144,U
47889.34,22888.89,19954.44,18000.00,55714.29,-1024,-397,406,16880,-340,7112,V
38653.85,23946.70,22065.22,14738.37,59853.66,-13,-60,439,15604,-2668,1668,R
72685.71,24857.14,26585.37,25702.92,64507.77,-939,-707,954,17768,-1444,4680,N
37731.96,22888.89,13469.39,13155.96,65763.16,340,718,-358,16324,-152,5464,K
55068.49,23652.39,12950.82,12015.85,58421.05,-183,-714,-524,15696,-2024,5928,U
52898.23,23266.83,18142.55,17102.51,55068.49,237,825,-146,15864,-1384,6212,V
42905.66,24045.80,19954.44,13628.73,71207.87,-532,-486,186,15688,-716,4728,R
73699.42,24245.52,26029.41,25488.13,64507.77,-457,-949,240,15980,-2552,5060,N
41208.79,22702.70,13311.81,12950.82,62500.00,-374,-32,-844,15368,-528,4836,K
55714.29,23555.27,12798.91,12111.31,53811.66,347,-1028,-262,16388,-1440,4564,U
53811.66,22610.29,18577.68,17510.59,52898.23,409,367,-132,16320,-1592,4872,V
40800.00,23076.92,22154.96,14452.98,53504.46,699,-556,-185,15512,-1924,4348,R
72685.71,25170.16,24857.14,25702.92,67058.82,-453,-897,1114,18104,-784,4752,N
48672.20,22245.15,13952.83,13259.67,60220.59,-221,-617,-44,16068,868,4800,K
49746.84,22795.56,12798.91,12500.00,54750.00,-258,-950,-192,16720,-1696,3620,U
48409.09,21975.90,18286.34,17237.39,57382.08,471,-812,-838,15236,-696,6368,V
41003.65,24045.80,20936.77,14854.09,63290.82,-560,-51,194,16568,-672,5648,R
74736.84,24857.14,26250.00,27743.73,68852.46,1386,-1477,-649,15224,-1388,6556,N
41833.33,23171.64,13155.96,13001.82,61343.28,1082,-1363,-108,15840,308,5504,K
55068.49,23458.65,12798.91,12748.64,58071.43,132,-896,-370,15560,-1748,4996,U
57725.12,22245.15,17718.55,16000.00,56705.61,544,-64,-343,15940,-1272,7640,V
44247.11,23848.10,23652.39,15446.43,65340.31,1319,188,-1445,16888,-412,4816,R
72685.71,24753.88,26698.37,27506.93,63290.82,505,-306,-312,16308,-2544,3288,N
42905.66,23266.83,13682.25,14452.98,62500.00,-3046,-1066,1928,15596,1728,5092,K
50858.37,23171.64,12648.65,12159.29,51717.39,279,-1369,-545,15412,-1660,4596,U
50858.37,22610.29,17102.51,16639.18,50021.19,76,214,-466,16184,-608,6544,V
34025.56,23458.65,22982.67,15813.25,65763.16,-250,-740,-139,15456,-1432,8244,R
28225.35,8644.07,12207.45,11733.45,63290.82,192,-188,266,15804,-3616,4872,B
28968.48,8753.87,12304.27,11686.96,63692.31,-119,-209,-596,15692,-4280,4788,B
28593.75,8827.64,12255.77,11733.45,64921.88,-117,-139,-357,15612,-4352,4748,B
28225.35,8901.87,12255.77,11733.45,64921.88,136,-201,-284,15892,-4056,4992,B
28470.25,8864.7,12255.77,11873.91,64507.77,46,-202,-370,15732,-3728,4928,B
27743.73,8939.16,12304.27,11873.91,68852.46,340,-277,-678,15812,-3364,5364,B
27983.19,8827.64,12401.79,11780.1,68396.74,447,-196,-78,15672,-3608,5484,B
28225.35,8864.7,12500,11780.1,64921.88,101,136,-123,15576,-4048,5548,B
28717.95,8827.64,12401.79,11780.1,66190.48,-79,-131,-77,15564,-3940,5544,B
28225.35,8864.7,12450.81,11780.1,66622.34,64,-158,-187,15632,-3964,5600,B
28593.75,8901.87,12798.91,12015.85,71207.87,420,68,-133,15776,-3624,5308,B
29478.26,9089.48,12798.91,12015.85,64921.88,-81,16,-113,15592,-3720,5376,B
28593.75,8790.7,12748.64,12015.85,67500,110,-497,-430,15740,-3744,5452,B
29478.26,8976.56,12798.91,11968.37,67500,-98,-405,-501,15724,-3784,5648,B
28842.86,8901.87,12798.91,12063.49,67945.95,355,-171,-383,15684,-3644,5816,B
29737.61,8864.7,13053.01,12159.29,73189.65,2,108,131,15780,-3848,5292,B
29221.9,8939.16,13001.82,12304.27,71207.87,67,-162,-205,15600,-4352,5192,B
29737.61,9014.09,13053.01,12207.45,72685.71,130,-284,-225,15616,-4112,5072,B
28842.86,8901.87,13053.01,12255.77,72685.71,167,-107,-265,15616,-4228,5172,B
29478.26,8901.87,13001.82,12207.45,69313.19,2,16,-240,15656,-4080,5092,B
41623.61,16902.29,23848.1,18286.34,44941.41,-83,-968,-54,15368,-4848,3452,C
41003.65,16444.67,23652.39,18725.27,51428.57,1535,-477,-457,15828,-3776,4576,C
39803.57,16252.54,22702.7,18214.29,47125.5,4398,-2235,306,15688,-3524,5220,C
41208.79,16252.54,22426.83,18142.55,45176.47,3343,-775,-525,15824,-3344,5216,C
43346.01,16639.18,22888.89,18504.37,44941.41,-1621,4,-834,15788,-4208,5076,C
40800,16509.24,22795.56,18214.29,46135.45,-1491,1185,-943,15744,-3104,5000,C
41415.44,16444.67,22518.34,18142.55,43793.11,-36,-507,-111,15760,-3736,5512,C
41623.61,16509.24,22795.56,18358.69,43568.71,1692,-1349,420,15556,-3540,6008,C
38281.25,16380.37,22426.83,18286.34,49205.02,-1819,-348,-835,15616,-2932,6380,C
42471.91,16380.37,22518.34,18142.55,45413.38,978,-721,-25,15848,-2932,6124,C
42471.91,16444.67,22702.7,18142.55,45652.17,812,-339,25,15936,-3632,4676,C
46135.45,17102.51,23458.65,18799.56,41003.65,797,-444,106,16004,-3240,4924,C
```

```
44941.41,17510.59,24145.41,19328.86,41208.79,-1087,-154,-439,15800,-3392,5668,C
43568.71,17102.51,24145.41,18799.56,43346.01,81,92,-441,15696,-3720,5464,C
43346.01,16968.75,23652.39,18651.32,42044.61,1601,357,-153,15544,-3368,6076,C
41623.61,16444.67,22518.34,17788.46,24346.15,-520,-155,-317,15688,-3656,5652,C
41208.79,16316.32,22518.34,18071.12,47632.65,172,-362,-209,15752,-3628,5336,C
41415.44,16316.32,22518.34,18214.29,47378.05,-193,-525,-188,15616,-3536,5480,C
40197.84,16316.32,22610.29,18071.12,45413.38,2426,-1527,-162,15832,-3812,5732,C
38281.25,16000,21975.9,17718.55,45892.86,531,-559,-181,15232,-4908,6112,C
43125.00,16639.18,22982.67,11873.91,43125.00,-4610,-267,541,16196,-1612,4520,D
43346.01,16252.54,22795.56,12255.77,45652.17,978,-130,386,16552,-784,2664,D
40597.83,16189.02,21975.90,12698.56,43793.11,-207,-452,246,16512,-836,3084,D
41415.44,16189.02,22245.15,12598.92,44247.11,-258,78,484,16360,-1640,3096,D
41415.44,16189.02,21975.90,12598.92,42257.46,-89,-350,-293,16320,-1540,3152,D
42687.97,16189.02,21975.90,11921.05,44247.11,351,211,11,16492,-1360,3108,D
45652.17,16380.37,22610.29,12598.92,43793.11,140,351,-72,16072,-1936,2016,D
47632.65,17035.49,22245.15,11873.91,45413.38,-69,-293,-170,16128,-1800,4772,D
48148.15,16770.19,22518.34,11780.10,45652.17,21,-562,496,16344,-1736,4084,D
47632.65,16836.10,22888.89,11502.59,44708.17,-317,-475,-324,16296,-2128,3420,D
46135.45,16704.55,22245.15,12015.85,43793.11,-145,113,-721,15832,-2764,4260,D
44708.17,16639.18,22154.96,12352.94,42257.46,777,-1127,1499,15856,-3396,3568,D
46135.45,16509.24,22335.77,12255.77,41208.79,740,-76,-292,16260,-2140,3188,D
46135.45,16380.37,21887.02,12255.77,45892.86,-110,-261,-174,16124,-2948,3240,D
45892.86,16444.67,22065.22,12207.45,45176.47,-68,-62,-335,16104,-2700,3180,D
43125.00,16509.24,22154.96,11873.91,45892.86,1708,-662,-1182,15952,-3076,2880,D
42905.66,16252.54,21021.13,12748.64,44941.41,-512,-68,50,16220,-2780,3736,D
41415.44,16189.02,21191.04,12352.94,46135.45,-335,-446,461,16892,728,5548,D
41623.61,16125.76,21105.88,12352.94,46380.00,222,-130,-405,16020,-2332,3824,D
43346.01,16062.75,21105.88,12401.79,45892.86,-367,-451,-666,16404,-1320,4112,D
57042.25,21362.56,31081.08,24045.80,69779.01,68,-258,-445,15440,-5380,2024,E
62893.40,21276.60,30534.13,24346.15,72685.71,129,961,59,16016,-3500,3236,E
60965.34,20936.77,29737.61,23750.00,79722.22,719,-518,-31,15756,-4168,4132,E
63290.82,21021.13,30132.36,23848.10,79141.10,240,220,-72,15840,-4336,4392,E
62500.00,21021.13,30534.13,23848.10,72187.50,257,-453,-258,15904,-4196,4240,E
60220.59,21276.60,30805.97,23555.27,67945.95,258,-314,-92,15740,-3708,4340,E
64097.94,21105.88,28347.46,23555.27,68396.74,-120,130,-356,15616,-4228,4444,E
65340.31,21105.88,30132.36,23458.65,69779.01,235,-517,-141,15720,-4476,4096,E
58774.04,20936.77,30000.00,23458.65,64097.94,182,-667,-178,15740,-4072,3836,E
62110.55,21105.88,30132.36,23555.27,67945.95,-281,-72,-237,15796,-4192,4740,E
63290.82,20852.80,29868.42,23266.83,65763.16,155,-53,-355,15624,-4488,4168,E
61343.28,20852.80,30265.49,23362.50,63290.82,3526,-1581,-315,15760,-3516,3864,E
59490.29,20769.23,30000.00,23362.50,60220.59,-136,-281,224,15780,-3940,4428,E
64921.88,22610.29,31500.00,25920.00,71207.87,322,-495,-181,16032,-3472,4068,E
64507.77,22245.15,30805.97,24960.94,65340.31,1175,51,-509,15980,-3504,3572,E
65340.31,22335.77,30669.64,24651.16,64921.88,-49,134,-241,16040,-3396,3264,E
64507.77,21975.90,30132.36,24548.97,68396.74,-144,-195,-20,16000,-3764,3596,E
63692.31,21710.53,30669.64,24245.52,62500.00,216,-278,-163,16076,-3532,3584,E
66622.34,21887.02,30669.64,24145.41,62500.00,-88,-453,-187,15940,-3640,3560,E
61725.00,21535.71,32507.74,24651.16,56705.61,196,-452,-18,15900,-4116,3164,E
33714.29,9710.15,13053.01,18577.68,44019.23,777,-819,-187,15660,-3212,5900,F
32953.12,9951.22,13628.73,18504.37,42257.46,-201,-739,-103,15496,-3532,6196,F
32953.12,9750.00,13735.96,18504.37,44247.11,1538,-423,-24,15496,-3592,5924,F
32507.74,9951.22,13682.25,18431.37,44941.41,-84,-258,-82,15800,-3196,5832,F
33103.45,9951.22,13735.96,18504.37,46135.45,195,-407,-70,15720,-3244,6288,F
33406.94,9870.34,13682.25,18504.37,45652.17,124,-269,-150,15700,-3316,5716,F
33254.72,9750.00,13469.39,19483.14,41623.61,164,-73,55,15840,-3176,5404,F
34340.84,9830.10,13789.87,19024.39,42044.61,650,-585,-48,15836,-3372,4944,F
32803.74,9670.42,13898.31,18874.17,41833.33,130,-213,-236,15920,-3600,4852,F
33560.13,9951.22,13898.31,19024.39,44247.11,-36,-279,-176,15796,-3512,5072,F
32953.12,9991.86,13789.87,19024.39,42471.91,-48,-181,-199,15956,-3068,5404,F
33714.29,9870.34,14228.57,19100.00,43793.11,332,-297,-92,15724,-3788,4932,F
33254.72,10073.53,14173.00,19024.39,42905.66,134,-670,179,15684,-3992,4624,F
33714.29,9951.22,14173.00,19175.95,42257.46,247,-111,-320,15732,-3388,5176,F
33103.45,10155.74,14228.57,19024.39,42905.66,34,-385,-86,15684,-3588,5336,F
32803.74,10155.74,14228.57,19175.95,43125.00,-34,-290,-278,15716,-3608,5032,F
32361.11,9830.10,14007.56,19328.86,35147.06,426,394,-41,15852,-3612,4448,F
```

```
32215.38,10280.06,14228.57,19175.95,43346.01,394,-601,-116,15728,-4012,4284,F
32803.74,10155.74,14173.00,19252.23,41415.44,-300,-441,-236,15752,-3904,4736,F
33103.45,10280.06,14117.65,19175.95,42257.46,540,-252,96,15924,-3840,4524,F
51142.25,23946.70,31219.88,12352.94,33103.45,54,165,-299,5448,15156,5728,G
62500.00,23171.64,29607.56,12352.94,33714.29,1097,80,-348,5548,16020,5136,G
69779.01,23171.64,30534.13,12401.79,33103.45,-121,-115,-129,5404,15424,5340,G
72187.50,22518.34,28842.86,12450.81,32953.12,-113,-198,-193,5344,15188,5340,G
72685.71,22702.70,30669.64,14284.35,33560.13,54,-65,-165,5284,15436,5524,G
68396.74,23266.83,29221.90,13735.96,32215.38,401,-1272,-633,5376,15456,5352,G
68852.46,22982.67,30534.13,13898.31,33103.45,190,-223,-222,5340,15440,5092,G
72685.71,22702.70,31926.61,13522.30,32070.55,158,-256,-263,5544,15428,5284,G
64921.88,22888.89,30805.97,13682.25,32953.12,108,-358,-266,5232,15192,5276,G
64507.77,22245.15,31081.08,13789.87,32655.28,157,41,-86,5296,15268,5476,G
60965.34,22610.29,29868.42,13001.82,29478.26,499,93,-235,5356,15180,5272,G
69313.19,22245.15,30000.00,12401.79,27983.19,-425,304,289,5212,15876,4876,G
71207.87,22518.34,29868.42,12900.00,28225.35,479,-244,-347,5444,15528,5300,G
65763.16,22154.96,30805.97,13001.82,28593.75,-31,-611,-320,5288,15468,5344,G
71207.87,22610.29,29868.42,14340.34,28225.35,228,224,-187,5736,14956,5636,G
72685.71,22610.29,30000.00,12950.82,28470.25,83,-613,-390,5360,15420,5584,G
67945.95,22245.15,30000.00,14062.50,28842.86,71,131,-72,5276,15316,5564,G
67500.00,22245.15,29868.42,13522.30,28470.25,311,-416,-414,5368,15324,5624,G
69313.19,21887.02,29607.56,12500.00,28470.25,184,-255,-272,5196,15224,5676,G
68396.74,22245.15,29737.61,12748.64,28593.75,64,-45,-133,5524,15132,5684,G
60591.13,19795.92,12111.31,11921.05,67945.95,643,-1047,-665,3740,16008,4044,H
58071.43,19717.19,12255.77,12015.85,67500.00,-64,-238,-162,3784,15832,4128,H
58421.05,19638.82,12159.29,12063.49,67500.00,-253,-33,-113,3720,16028,4024,H
58774.04,19795.92,12304.27,12159.29,68852.46,353,4,-394,3948,15916,4356,H
60220.59,19560.81,12450.81,12111.31,64921.88,21,-413,-266,4212,15756,4284,H
59130.43,19483.14,12500.00,12207.45,58774.04,-166,-46,5,3836,16212,4244,H
57382.08,19560.81,12450.81,12063.49,63692.31,-4,-28,-312,3868,16124,4168,H
61343.28,19405.83,12304.27,11968.37,69779.01,-437,439,50,4124,15792,4372,H
61343.28,19483.14,12500.00,12015.85,65763.16,164,-537,-224,3732,15880,4180,H
62500.00,19560.81,11968.37,11921.05,63692.31,21,-477,-186,3728,15856,4144,H
62500.00,19638.82,12304.27,12015.85,68852.46,-70,-150,-2,3748,16044,4152,H
62110.55,19717.19,12111.31,11921.05,68396.74,-355,-207,-194,3648,15548,4340,H
62110.55,19560.81,12255.77,11826.92,64921.88,116,-10,-225,3760,16128,4084,H
58421.05,19483.14,12304.27,11968.37,74215.13,790,-599,-978,4064,16388,4120,H
56372.09,19560.81,12352.94,11826.92,64921.88,-248,113,-131,3860,16276,4100,H
57042.25,19405.83,12401.79,11968.37,62110.55,514,-686,-442,3724,16080,4384,H
54434.39,19024.39,13364.14,12111.31,64921.88,214,-331,-295,3912,15832,4224,H
51428.57,19175.95,12900.00,12159.29,68852.46,-739,-517,-384,4188,15928,4752,H
55389.91,19175.95,12549.37,12015.85,71694.91,52,674,-87,4140,15532,5048,H
53811.66,18949.11,12648.65,12063.49,70726.26,-1069,912,586,4292,15528,5844,H
33560.13,19483.14,26811.99,31500.00,47125.50,-3756,-1187,-389,13196,-6816,6972,I
32507.74,20603.25,26585.37,30000.00,48148.15,-851,943,-682,14412,-6856,5704,I
30669.64,20275.86,26585.37,30265.49,50297.87,-1569,2437,106,15672,-5512,4204,I
31641.34,20114.41,26361.18,30399.41,58071.43,348,-192,-199,15276,-5448,4264,I
33254.72,19954.44,25595.24,29737.61,51142.25,8,-96,92,15268,-5560,4460,I
32361.11,20034.25,26139.41,28225.35,50021.19,92,231,91,15212,-6100,5328,I
38653.85,19175.95,26250.00,28717.95,41208.79,2093,2735,-1714,15912,-5140,3596,I
40397.11,20114.41,26811.99,28968.48,43568.71,-358,-1662,-632,15100,-5748,3660,I
31926.61,19638.82,25488.13,28842.86,48937.50,-828,-2792,-929,15396,-5072,3912,I
31641.34,19252.23,25275.59,28842.86,51142.25,630,-341,-167,15492,-6432,4876,I
32507.74,19405.83,24651.16,28593.75,51428.57,76,1953,1161,14780,-7500,5096,I
31926.61,19405.83,25488.13,28470.25,49474.79,46,210,-159,14980,-6032,4044,I
31500.00,19328.86,24651.16,28717.95,49205.02,154,-275,-101,15392,-5372,3452,I
31641.34,19252.23,24651.16,28717.95,48409.09,-982,841,118,15212,-6320,5244,I
32070.55,19252.23,24346.15,28103.93,51428.57,-10,107,-4,14944,-6328,5468,I
31926.61,19328.86,24651.16,28470.25,49205.02,395,178,327,15024,-6048,5328,I
31783.54,19328.86,24447.30,28225.35,51142.25,-171,-920,-382,14884,-6264,4380,I
31783.54,19483.14,24548.97,29737.61,51428.57,171,-150,-247,15188,-6408,4768,I
31641.34,19483.14,23946.70,28225.35,51142.25,111,136,-205,15168,-6012,4160,I
32215.38,19252.23,24145.41,27506.93,49205.02,364,-35,-146,15020,-6080,3900,I
32507.74,19717.19,27041.10,25170.16,47378.05,32767,13207,12154,5300,1944,16284,J
30534.13,17102.51,22702.70,21105.88,42471.91,32767,1506,-7433,9792,-5908,13940,J
```

```
30943.12,14970.70,19875.00,19175.95,47889.34,-4890,7769,475,11732,-6340,7140,J
28593.75,15690.00,21535.71,20852.80,46626.50,32767,5758,-3272,10552,3544,9484,J
30265.49,15446.43,21276.60,20686.05,45652.17,32767,20526,-6634,10348,1736,9144,J
31926.61,14228.57,19100.00,18431.37,45176.47,19950,1309,-7493,10888,-8020,7016,J
29737.61,14340.34,19560.81,19175.95,47125.50,-7480,1416,45,15312,6160,4452,J
28842.86,14117.65,19252.23,18651.32,43568.71,-3651,5118,-3199,12380,5872,7120,J
29094.83,13735.96,19328.86,18577.68,43793.11,-1052,-1583,3227,12604,3144,5428,J
30399.41,14228.57,19483.14,19100.00,44941.41,-13576,8667,-4152,13408,5696,9688,J
29221.90,17648.94,23946.70,21622.91,46875.00,1153,-24182,17597,14696,9584,6424,J
31081.08,13259.67,19252.23,18577.68,42044.61,32767,9525,4371,6672,2768,18128,J
30000.00,12648.65,17648.94,17441.86,42905.66,579,-1330,29,7264,-9572,7752,J
30000.00,13104.40,18071.12,17441.86,43125.00,-3849,-724,542,9120,-11836,10528,J
30805.97,13469.39,18142.55,17718.55,44941.41,1332,-143,-99,9780,-12004,9936,J
29478.26,13735.96,18874.17,18431.37,42687.97,-19026,13345,-8878,11100,4584,14952,J
27863.13,13628.73,18577.68,17929.18,44941.41,-4335,4793,-2872,15576,5336,3336,J
28225.35,13207.72,19405.83,18651.32,55068.49,14470,-8707,10640,17812,6168,-228,J
28842.86,13898.31,19328.86,18504.37,45176.47,-10145,2349,88,15244,7844,4716,J
30399.41,14509.62,20034.25,18431.37,44019.23,-10638,7771,7747,11060,5996,1256,J
70250.00,23266.83,31641.34,12598.92,38096.89,648,-507,1312,15772,-3328,2168,L
66190.48,23362.50,31500.00,12648.65,35147.06,1042,-2255,-1112,15576,-4892,4980,L
64507.77,22610.29,31926.61,12598.92,35311.48,-40,-330,-247,15728,-4300,3916,L
73699.42,22610.29,33103.45,12549.37,36493.29,194,-464,-474,15696,-4276,3904,L
65340.31,22982.67,30669.64,12598.92,34821.43,-2,2,-218,15872,-4156,3588,L
65340.31,22610.29,30669.64,12900.00,36841.21,-1135,854,62,15604,-3328,96,L
65340.31,22610.29,30669.64,13104.40,36841.21,508,-1056,-247,16256,-2848,1352,L
61725.00,22154.96,30132.36,13207.72,36666.67,-9510,2232,1285,15792,-3476,804,L
64097.94,22154.96,30265.49,13207.72,40000.00,2604,-526,-2405,15528,-4332,2256,L
60220.59,21975.90,30265.49,13259.67,39414.89,1628,-1246,-478,15672,-4180,1252,L
64097.94,21887.02,29607.56,13259.67,40397.11,107,-217,-257,15836,-4268,1272,L
58774.04,21975.90,30534.13,13155.96,38653.85,701,-763,-253,15872,-4056,1528,L
61343.28,21887.02,28470.25,13053.01,36841.21,-53,-133,-251,16000,-4124,1444,L
60965.34,21798.56,29607.56,12849.36,26585.37,256,-640,-393,15764,-4608,1876,L
57382.08,21975.90,29094.83,12352.94,28470.25,-481,-541,27,15716,-4720,1628,L
60220.59,22154.96,28347.46,12549.37,33560.13,126,-1018,-375,15024,-5088,-1584,L
64507.77,21362.56,30265.49,13469.39,39803.57,121,-268,-127,15180,-4656,-1496,L
60220.59,21191.04,30399.41,12698.56,35311.48,482,2886,-1073,15376,-4724,-1352,L
60220.59,21021.13,29221.90,12598.92,35643.57,-337,-112,-130,15536,-4312,0,L
62110.55,21448.93,29349.71,12698.56,35147.06,336,-241,-289,15524,-4584,-56,L
41415.44,15628.74,21710.53,18431.37,51428.57,2772,-400,-601,15176,-6172,4172,O
41415.44,15690.00,21887.02,18358.69,54434.39,1005,-1228,-69,15444,-5228,1964,O
39608.54,15628.74,21798.56,18358.69,49205.02,-255,-48,-345,15504,-5324,2088,O
40000.00,15628.74,21887.02,18142.55,49474.79,-76,-451,-11,15548,-5600,1436,O
40397.11,15628.74,21105.88,18504.37,53200.00,397,-106,-305,15172,-6068,1188,O
38281.25,15690.00,21622.91,18214.29,55714.29,1132,-2045,761,15628,-4740,1944,O
39803.57,15628.74,21622.91,18214.29,54750.00,457,-615,136,14976,-6092,-288,O
40597.83,15628.74,21622.91,18214.29,52898.23,-143,-377,226,15456,-5428,-588,O
40800.00,15567.73,21975.90,18142.55,53200.00,84,-302,-53,15312,-5576,232,O
40197.84,15628.74,21798.56,18142.55,51428.57,246,-335,-32,15372,-5372,292,O
39031.69,15567.73,21710.53,18000.00,52008.73,-130,-163,-16,15284,-5752,224,O
40000.00,15628.74,21535.71,18214.29,52008.73,136,-113,-308,15336,-5636,340,O
38842.11,15628.74,21710.53,18286.34,50021.19,-48,-249,-155,15324,-5436,376,O
41415.44,15628.74,21710.53,18071.12,50021.19,46,-257,-199,15228,-5816,388,O
38096.89,15690.00,21710.53,18214.29,49205.02,107,-245,-238,15304,-5592,576,O
39803.57,15628.74,21887.02,18142.55,48937.50,121,-201,-262,15292,-5592,484,O
40397.11,15690.00,21362.56,18431.37,52008.73,319,78,-518,15324,-5624,500,O
39608.54,15690.00,22335.77,18504.37,49746.84,980,-320,-732,15624,-4652,1072,O
40197.84,15690.00,21798.56,18142.55,53811.66,754,463,-341,15752,-4740,1248,O
39031.69,15751.50,21105.88,18214.29,50297.87,630,-443,318,15644,-5460,956,O
53504.46,18286.34,13416.67,13843.99,52302.63,-281,-80,-537,-14576,1516,8228,P
53504.46,18358.69,13104.40,13628.73,53811.66,1029,-1397,-406,-14440,2156,10016,P
52008.73,18504.37,13364.14,14007.56,50858.37,3478,-1561,231,-13480,2908,10572,P
53811.66,18577.68,13416.67,14228.57,50858.37,-405,-425,-476,-14152,2768,10560,P
51142.25,18577.68,13575.42,14062.50,53811.66,-85,-318,257,-13332,3192,10492,P
52599.12,17718.55,12648.65,13207.72,50858.37,466,-19,-119,-14864,2468,9100,P
49205.02,17718.55,12304.27,13155.96,54434.39,93,-529,-855,-14828,2228,9468,P
```

```
47632.65,17788.46,12450.81,13364.14,53200.00,-2136,-3805,-1729,-12280,1500,14020,P
48937.50,17648.94,12648.65,13628.73,51717.39,263,2911,2121,-10932,3984,13812,P
44941.41,17169.81,12450.81,13311.81,49746.84,718,-56,-466,-15444,4136,7344,P
48148.15,17929.18,13155.96,13416.67,57382.08,921,-241,263,-13784,6988,7452,P
46380.00,17718.55,13155.96,13469.39,56372.09,-2809,-722,-4725,-14128,6948,6848,P
49205.02,17648.94,13682.25,13843.99,54750.00,98,-773,-524,-14284,3352,9948,P
49205.02,17237.39,12798.91,14173.00,52898.23,274,-141,145,-13612,3288,10240,P
48937.50,17237.39,13207.72,14284.35,53811.66,352,-78,-667,-13796,4184,10128,P
50021.19,17788.46,12950.82,14228.57,56372.09,1106,843,-88,-13728,3896,10084,P
49205.02,17510.59,12500.00,14452.98,56372.09,-31,-458,-301,-13892,3348,10612,P
55068.49,17237.39,12450.81,14228.57,53504.46,-58,-291,-520,-13760,3184,10484,P
48672.20,17305.27,12698.56,14228.57,51428.57,-143,-577,-649,-14616,576,11580,P
54121.62,17237.39,12598.92,14117.65,51717.39,3647,986,3358,-12352,1720,9480,P
59490.29,22795.56,32803.74,13843.99,30943.12,637,215,-115,-8156,2056,16420,Q
55068.49,22335.77,32215.38,13843.99,31926.61,280,35,-183,-8736,1900,15920,Q
55068.49,22245.15,31926.61,13898.31,31500.00,36,-45,-67,-9000,2080,15972,Q
56041.67,22245.15,31081.08,13843.99,31359.52,73,-71,-224,-9088,2180,16000,Q
52008.73,22335.77,31926.61,13898.31,30943.12,355,-264,-3,-9056,2116,15904,Q
56372.09,22335.77,30805.97,14228.57,31219.88,187,-87,320,-12708,1916,15756,Q
54121.62,21887.02,31926.61,14007.56,30669.64,41,-134,-210,-9756,2004,15212,Q
59130.43,21710.53,29478.26,14340.34,25702.92,303,-361,-239,-9224,1856,15668,Q
53811.66,21622.91,30534.13,14062.50,26250.00,-42,-109,-326,-9344,1736,16136,Q
54434.39,21622.91,29478.26,14062.50,27041.10,256,1,-85,-9160,1884,15580,Q
59130.43,21191.04,28968.48,14117.65,27506.93,22,9,-206,-9264,2036,15684,Q
53200.00,21021.13,29868.42,14173.00,27743.73,425,-221,-169,-9608,2016,15604,Q
50297.87,21276.60,29737.61,14117.65,27625.00,346,-198,-146,-9620,2256,15292,Q
52599.12,21276.60,28968.48,14173.00,28347.46,-92,-297,-426,-9456,1964,15516,Q
54750.00,21191.04,29221.90,14117.65,27389.50,357,-278,21,-9532,2144,15488,Q
50858.37,21975.90,27863.13,14117.65,28103.93,-235,-421,-220,-9920,2272,15052,Q
52898.23,20852.80,28717.95,14062.50,29607.56,-279,-497,-467,-9544,2264,15088,Q
54121.62,21105.88,28593.75,13155.96,29737.61,-217,-245,-251,-9000,1480,15868,Q
52302.63,21191.04,28470.25,13364.14,30132.36,-159,-163,-303,-9080,1804,15980,Q
55068.49,20936.77,28347.46,13416.67,30265.49,-438,-193,-297,-8832,1780,15796,Q
58071.43,9280.06,12748.64,11921.05,59853.66,67,-192,-341,15212,-5248,5548,W
59490.29,9165.35,12748.64,11968.37,57382.08,-244,-375,-190,15136,-5200,5512,W
55389.91,9280.06,12698.56,12015.85,58071.43,477,-706,-27,15184,-5816,4824,W
64507.77,9165.35,12748.64,12015.85,62110.55,262,-650,66,15056,-6072,4464,W
59130.43,9357.14,12748.64,12015.85,58774.04,190,-86,92,15108,-5776,4808,W
56041.67,9089.48,12798.91,12015.85,56041.67,151,-270,-222,15076,-6224,4756,W
61725.00,9318.54,12798.91,12015.85,60220.59,140,-179,-304,14876,-6084,5100,W
59130.43,9630.82,12900.00,12255.77,58421.05,-27,406,-709,14956,-6072,5528,W
58071.43,9670.42,12950.82,12063.49,57382.08,214,-301,-223,15136,-6044,4720,W
58071.43,9089.48,12798.91,12111.31,54750.00,-329,-138,-215,15040,-5688,5284,W
60965.34,9395.87,12798.91,12255.77,54750.00,42,-554,-208,15040,-5320,5192,W
62110.55,9630.82,12849.36,12111.31,59130.43,445,-462,-155,15080,-6320,4316,W
53504.46,9395.87,12849.36,11921.05,58071.43,-603,-123,-163,15180,-5748,5128,W
56041.67,9473.68,12950.82,12063.49,55714.29,3,-393,-185,15028,-6084,4772,W
59853.66,9512.78,12849.36,12015.85,54121.62,66,-355,-189,15012,-5852,4812,W
57042.25,9552.00,12849.36,12063.49,56041.67,-35,-318,-162,15024,-6108,4720,W
60591.13,9630.82,13001.82,12159.29,53504.46,244,-298,-185,15064,-5756,4992,W
65340.31,9552.00,12849.36,12063.49,56372.09,304,-319,-322,15072,-6368,4536,W
58421.05,9710.15,12950.82,12207.45,56372.09,7,-177,-309,14984,-6072,4816,W
69313.19,10114.57,13001.82,12159.29,56705.61,51,-110,-340,15096,-5932,5212,W
53504.46,22065.22,31926.61,24548.97,47889.34,736,-388,501,15644,-4292,3524,X
53504.46,22154.96,30669.64,24753.88,46875.00,116,-472,-15,15772,-4548,3744,X
56372.09,21710.53,29868.42,24548.97,47125.50,105,-360,-180,15504,-4904,3508,X
59130.43,21975.90,28225.35,22888.89,50297.87,-93,-810,-245,15724,-4840,2624,X
56372.09,21710.53,30132.36,23946.70,48937.50,-281,-270,-503,15628,-4984,2672,X
56041.67,21622.91,30265.49,23848.10,46875.00,195,-642,-365,15696,-5008,2628,X
52898.23,21535.71,30265.49,22982.67,47125.50,334,-155,-396,15608,-5012,2616,X
55389.91,21887.02,28593.75,23652.39,47632.65,151,-204,-241,15680,-4892,2776,X
57382.08,21535.71,29478.26,23555.27,46626.50,34,-64,-336,15820,-4716,2724,X
53504.46,21362.56,29737.61,23266.83,48672.20,-47,149,-275,15660,-5064,2980,X
50858.37,22335.77,27983.19,23848.10,44476.74,150,-170,-195,15640,-4860,2488,X
53200.00,21535.71,29478.26,23266.83,42905.66,10,-315,-346,15636,-4712,3040,X
```

```
52302.63,20936.77,29737.61,22888.89,43125.00,255,-173,-284,15552,-5076,2992,X
51428.57,21362.56,29221.90,22982.67,48148.15,259,-189,-84,15524,-5524,1308,X
53811.66,21276.60,29094.83,22795.56,49205.02,95,-190,-359,15656,-5108,2976,X
53504.46,21276.60,28968.48,22610.29,48672.20,-252,-436,-194,15692,-5280,2912,X
60591.13,21276.60,27506.93,22154.96,48409.09,-126,-142,-251,15524,-5192,3744,X
56705.61,21105.88,28593.75,22610.29,46135.45,-228,-327,-260,15524,-5068,2892,X
55068.49,21021.13,29478.26,22245.15,48409.09,-234,-849,-400,15580,-5032,3044,X
65340.31,21191.04,28593.75,22610.29,47125.50,-138,-438,-262,15556,-5088,2440,X
35643.57,18725.27,25920.00,29221.90,27156.59,18,-133,-307,15356,-3824,6836,Y
32070.55,18651.32,25065.28,26139.41,28103.93,-445,-491,-816,15508,-4952,5640,Y
34025.56,18504.37,25065.28,25595.24,26698.37,351,-112,-317,15732,-3800,5228,Y
34821.43,18286.34,26472.97,24960.94,27041.10,633,-1014,-772,15640,-3344,4724,Y
33869.43,18577.68,26139.41,27272.73,27863.13,-748,-568,-958,15344,-4812,4744,Y
34983.71,18358.69,25920.00,26926.23,26472.97,341,1093,440,15728,-3960,3636,Y
32507.74,18358.69,25488.13,26585.37,31359.52,683,-167,136,15540,-4868,4852,Y
33254.72,18358.69,24960.94,25381.58,30399.41,-456,-692,-485,15212,-5380,5812,Y
33714.29,18358.69,25381.58,26472.97,30805.97,-274,-847,-599,15492,-5124,4416,Y
33103.45,18358.69,25702.92,26811.99,31783.54,409,-67,-232,15464,-4804,3880,Y
33869.43,18358.69,25702.92,25811.17,31359.52,-21,-217,-86,15404,-5340,4516,Y
33560.13,18286.34,25811.17,26250.00,31081.08,-375,-365,-358,15360,-5024,4948,Y
35811.26,18286.34,25065.28,25920.00,32507.74,168,-445,-83,15372,-4924,4936,Y
32361.11,18431.37,25595.24,25920.00,31500.00,104,-1135,-326,15540,-5032,4636,Y
35643.57,18286.34,25595.24,25920.00,31500.00,-842,-958,-512,15448,-4696,5232,Y
33103.45,18431.37,25595.24,25170.16,33103.45,1053,-785,-325,15624,-5140,4144,Y
34340.84,18504.37,26361.18,25920.00,32361.11,-757,-482,-531,15368,-5012,5808,Y
33254.72,18286.34,25170.16,25170.16,31926.61,171,-327,-155,15388,-4928,4720,Y
34983.71,18358.69,25275.59,25702.92,31500.00,76,-263,-259,15520,-4844,4784,Y
39222.61,18000.00,25381.58,25595.24,32655.28,140,-202,-282,15524,-4812,4876,Y
64507.77,22518.34,30000.00,12798.91,53504.46,3551,-2354,3629,13136,4252,9872,Z
61343.28,22245.15,28593.75,12900.00,54121.62,-12664,7518,-16364,11244,-4536,4008,Z
58071.43,22795.56,30399.41,12648.65,56705.61,-558,-4560,9399,14636,8932,7132,Z
65340.31,21710.53,27983.19,12648.65,52302.63,3657,-4032,7131,11596,-88,15800,Z
58071.43,21276.60,27743.73,12598.92,53504.46,-9251,9885,-16296,14216,9888,10828,Z
56041.67,21105.88,27156.59,12598.92,53200.00,11305,7062,15698,11180,2080,10044,Z
59130.43,21021.13,27389.50,12549.37,46875.00,1374,-4945,9287,8648,-8844,9160,Z
53504.46,21105.88,26585.37,12748.64,53811.66,-11870,2227,-20770,12908,6272,13460,Z
56041.67,20686.05,26926.23,12598.92,57725.12,1728,-603,4262,14212,8936,9160,Z
53504.46,20852.80,27272.73,12598.92,57725.12,4271,854,-12450,6444,-15492,23068,Z
54434.39,20603.25,27863.13,12549.37,44019.23,-7196,6925,-17207,11956,-3940,9344,Z
61343.28,20603.25,27863.13,12598.92,44941.41,-4637,247,5659,14484,8356,9016,Z
51428.57,20852.80,26472.97,12401.79,58071.43,14533,-774,14676,6680,-16636,17672,Z
56372.09,20603.25,26585.37,12401.79,51428.57,-4073,1678,1317,5008,-16572,14520,Z
54750.00,20357.14,26698.37,12549.37,56372.09,-11693,9151,-23150,9516,-5340,9904,Z
51428.57,20520.83,25920.00,12352.94,53504.46,-64,-1559,3695,16988,12968,8160,Z
48937.50,20520.83,26585.37,12549.37,53811.66,14508,5190,11922,12272,-9392,14004,Z
53504.46,20520.83,26472.97,12255.77,50297.87,-15485,-3577,-21926,10432,-1372,15232,Z
56041.67,20114.41,26811.99,12352.94,51428.57,-12125,-1680,-24441,11380,4184,9608,Z
54121.62,20275.86,25381.58,12401.79,50858.37,9005,6348,9033,9132,-5908,16500,Z
31081.08,9051.73,12159.29,12748.64,42687.97,318,946,2467,5856,-3188,18224,__
30805.97,8976.56,12401.79,12401.79,39222.61,-6566,15607,-32768,-1756,6320,22008,__
28103.93,9014.09,12401.79,12798.91,42905.66,3623,-3886,18748,4348,3460,19180,__
29868.42,8753.87,12401.79,12748.64,28593.75,-3139,10556,-6283,-1632,12692,28472,__
30265.49,9127.36,12500.00,13952.83,45413.38,-31660,27342,-32768,5280,-8836,13880,__
29868.42,9241.71,12648.65,13682.25,48672.20,4007,-4296,7995,6972,-1792,17336,__
30132.36,8976.56,12304.27,13628.73,38281.25,-9344,19368,-17721,-1112,2792,22084,__
29094.83,9318.54,12598.92,13522.30,57725.12,-1648,-3451,2602,4132,-5852,19532,__
31926.61,9127.36,12450.81,13364.14,47125.50,-12410,19117,-32435,5096,-1108,8736,__
30399.41,9318.54,12500.00,13789.87,45176.47,5713,-5354,10556,4792,648,18700,__
28968.48,9318.54,12648.65,13682.25,44708.17,-386,-1925,1810,2800,-11556,16120,__
29478.26,9051.73,12450.81,13469.39,43793.11,2501,-5514,5163,4720,3380,19060,__
30265.49,9127.36,12500.00,13575.42,49205.02,3752,-5135,10664,3312,-5600,19016,__
29868.42,9089.48,12450.81,13469.39,36666.67,11024,-2864,2121,-1108,6580,25852,__
27506.93,9670.42,12450.81,13522.30,48937.50,5805,-4427,9435,2284,-7200,19364,__
29607.56,9089.48,12549.37,14062.50,43793.11,4951,-2257,1502,7032,4752,16768,__
30132.36,9434.71,12698.56,14117.65,44941.41,4834,-1654,7314,2648,-8184,19456,__
```

```
30399.41,9512.78,12748.64,14284.35,50858.37,1900,-2371,6550,8876,2460,14160,__
28842.86,9280.06,12900.00,14396.55,48937.50,-2310,-5823,2810,2980,-10980,17680,__
30132.36,9127.36,12748.64,14173.00,44941.41,4799,3087,12471,5248,752,18704,__
27506.93,8864.70,12352.94,11186.01,61343.28,-271,-347,60,15532,-2844,5940,B
33714.29,16639.18,24857.14,16509.24,35476.97,-942,205,158,14632,-4768,7420,C
38281.25,17788.46,23362.50,11096.94,32953.12,718,-1695,-135,14868,-3192,8036,D
64507.77,21887.02,31219.88,25275.59,77439.76,-274,-263,-462,14508,-3392,9184,E
33103.45,9127.36,12159.29,20034.25,37551.38,242,410,-210,15264,-3612,6692,F
58774.04,21448.93,30943.12,11456.90,37551.38,105,-482,68,740,15448,7488,G
53811.66,20194.95,12159.29,11141.40,60965.34,184,-37,-150,6664,14220,7228,H
29868.42,23848.10,28347.46,32361.11,57042.25,778,-464,-145,15568,-5488,3552,I
30399.41,21105.88,27743.73,30399.41,42905.66,23380,17482,1300,6504,-1488,14792,J
57725.12,21535.71,36150.00,11921.05,30534.13,-234,500,-259,14844,-3612,8000,L
42471.91,15813.25,21887.02,17305.27,33560.13,553,-64,-234,15048,-3652,7700,O
45413.38,17102.51,13311.81,13469.39,52302.63,-1205,666,-2492,-8656,4648,14448,P
54750.00,23848.10,33714.29,13311.81,26811.99,-580,-300,-359,-4148,224,17524,Q
45892.86,8680.55,11826.92,11733.45,58071.43,-476,-926,14,16036,-1612,5116,W
60220.59,23555.27,32803.74,20357.14,47889.34,-283,-916,-16,16236,-2084,4792,X
28842.86,18577.68,27272.73,24651.16,27983.19,303,138,-269,15104,-1780,8340,Y
72685.71,22518.34,28103.93,11780.10,47378.05,3727,3874,4705,12848,3912,10304,Z
28103.93,8391.77,11640.63,11780.10,38281.25,-5146,4166,-7494,4200,-4200,16912,__
28842.86,9014.09,12500.00,11502.59,62500.00,130,-510,168,16084,-3116,3872,B
32655.28,14912.28,23458.65,14452.98,30000.00,-550,-1671,524,14328,-5204,8092,C
38842.11,17237.39,23076.92,11320.75,39414.89,-1703,640,-354,14440,-5332,7724,D
60220.59,21362.56,33406.94,25920.00,73699.42,285,18,-105,13796,-4936,9240,E
34660.20,9089.48,11594.46,21798.56,34983.71,-4755,1533,132,13752,-4644,8900,F
60220.59,23652.39,31219.88,11548.44,32507.74,399,82,-218,3704,15612,5320,G
46875.00,18725.27,12255.77,11640.63,81509.44,278,-255,-97,3812,14576,8680,H
29221.90,23171.64,29221.90,30669.64,56372.09,68,101,-121,15492,-5012,4064,I
30805.97,21448.93,26472.97,27041.10,46135.45,25191,2570,7358,2136,16308,19472,J
58071.43,21887.02,35980.07,11640.63,28103.93,-164,-700,-172,14792,-3876,7760,L
40197.84,14970.70,21362.56,16770.19,35147.06,94,-903,-196,14876,-4164,7884,O
54750.00,19175.95,14396.55,13628.73,52302.63,193,599,-723,-5904,4268,16528,P
58071.43,22795.56,31783.54,13522.30,26029.41,-777,-5401,-392,-6184,2836,19020,Q
43568.71,8680.55,12111.31,11594.46,56041.67,372,-416,-441,15656,-2028,6336,W
59130.43,23266.83,32070.55,21191.04,46135.45,-1899,521,595,16048,-1828,4992,X
29094.83,18214.29,25065.28,23946.70,29607.56,808,-756,-215,15312,-2016,7848,Y
50021.19,20275.86,30399.41,11733.45,44941.41,5931,5276,8581,5660,-2204,17944,Z
27041.10,8427.48,11275.69,11548.44,34340.84,2362,-2955,2441,3820,-1536,19196,__
29868.42,8864.70,12304.27,11230.77,72187.50,-353,112,-157,15424,-2912,6904,B
30399.41,14970.70,24753.88,17035.49,29094.83,80,-204,-441,14344,-4924,8424,C
34025.56,17035.49,24045.80,11411.36,37193.88,4714,-1737,-500,14588,-3500,8164,D
60591.13,20852.80,35147.06,24045.80,70250.00,608,-275,-318,14180,-5184,8576,E
34500.00,8901.87,11594.46,20769.23,34500.00,-935,367,-223,14644,-5064,8316,F
64921.88,23946.70,32953.12,12063.49,34660.20,-287,-481,85,4000,14624,8544,G
47889.34,19875.00,11968.37,11456.90,64507.77,484,-402,-190,6648,11876,11072,H
29737.61,23652.39,28103.93,32507.74,60591.13,-1366,-99,-477,15460,-4280,5196,I
29737.61,21535.71,25595.24,27272.73,39031.69,-349,-2753,4388,15576,2280,4,J
54434.39,21021.13,35311.48,11826.92,26811.99,162,-490,14,15092,-3924,7180,L
41003.65,15206.69,22065.22,18577.68,41003.65,-315,-159,-313,15472,-4104,5748,O
49205.02,17035.49,14452.98,13364.14,45176.47,-124,-108,-241,-8036,3496,16036,P
53504.46,23652.39,31359.52,13952.83,25488.13,-36,-275,-239,2640,1560,18244,Q
44019.23,8499.23,11594.46,11640.63,60591.13,135,297,-25,15636,-2392,6236,W
64921.88,23362.50,31219.88,19405.83,46875.00,261,-570,-58,15728,-2424,6312,X
29737.61,18214.29,24753.88,23750.00,30669.64,-860,-654,-480,14556,-2392,10088,Y
65340.31,21887.02,32070.55,11640.63,52599.12,660,-348,-2090,11296,-4804,12940,Z
28103.93,9241.71,11502.59,12304.27,40800.00,457,-2053,-4200,8188,5556,17356,__
28470.25,9051.73,12500.00,11365.98,65340.31,-1175,-1365,-497,14624,-4872,8784,B
28470.25,14623.55,23652.39,15506.96,32361.11,-645,-62,124,14804,-4192,7104,C
34182.70,17648.94,23848.10,11230.77,41415.44,206,-654,118,14932,-3864,7596,D
72187.50,22518.34,33254.72,26811.99,77439.76,471,-443,388,13476,-4712,10848,E
67945.95,22426.83,31500.00,11594.46,28470.25,1542,-433,23,6576,11064,12272,G
33714.29,9127.36,12207.45,20520.83,44708.17,-316,1116,-75,15232,-4920,6024,F
54750.00,19175.95,12450.81,11320.75,66190.48,70,-332,-50,6572,12016,11196,H
30534.13,20852.80,25381.58,29607.56,59130.43,458,3823,-2923,14288,-4152,7480,I
```

```
30399.41,21105.88,26472.97,27272.73,52008.73,32287,21346,2644,-484,3236,16292,J
51428.57,23076.92,37372.02,11733.45,28347.46,500,-55,-709,13472,-4904,9972,L
46135.45,16902.29,22065.22,19024.39,47889.34,-280,-45,-132,15344,-4284,6160,O
44708.17,18504.37,13207.72,14228.57,55389.91,46,-385,-226,-8064,3076,16196,P
44941.41,23458.65,31081.08,13469.39,25381.58,130,26,-152,-6852,1484,16896,Q
48148.15,8285.28,11502.59,11411.36,56372.09,-135,-1988,243,15236,-2884,7164,W
61725.00,23266.83,31926.61,19328.86,48148.15,-175,86,-188,15652,-2248,6540,X
30132.36,18504.37,25275.59,24346.15,29221.90,-234,-469,4,14632,-2896,8968,Y
44019.23,20520.83,27625.00,11873.91,46135.45,-802,64,1649,4276,-2364,17052,Z
28842.86,8499.23,11733.45,11968.37,37016.95,-1261,2900,-9967,3708,3528,18248,__
30265.49,8827.64,12549.37,11502.59,69313.19,58,-535,-349,14604,-3636,7556,B
35476.97,16189.02,24753.88,16639.18,35147.06,-406,-153,-278,14840,-3884,8708,C
37913.79,16704.55,23266.83,12015.85,42905.66,846,-851,-567,14416,-3552,7260,D
59490.29,22154.96,30399.41,24346.15,72187.50,-201,61,49,13424,-4464,10208,E
34340.84,9203.47,12698.56,18799.56,36666.67,-502,129,-370,13704,-3820,10628,F
65763.16,23458.65,31641.34,12352.94,28593.75,-797,-287,-280,7052,6812,15284,G
51428.57,18949.11,12255.77,11594.46,61343.28,169,-119,-231,8424,5284,15044,H
32953.12,21975.90,23458.65,25275.59,47889.34,153,-325,-315,15440,-4552,4952,I
30534.13,20114.41,21710.53,23848.10,41623.61,16161,-23165,18475,10928,9552,8260,J
56705.61,21887.02,32803.74,11921.05,34983.71,383,-303,75,14964,-2604,8360,L
42687.97,16509.24,22795.56,18874.17,39803.57,1086,-131,-216,13900,-5712,8980,O
53811.66,18949.11,12748.64,13364.14,51717.39,-1093,-174,-475,-6188,3348,17004,P
47889.34,23652.39,29094.83,13364.14,27156.59,-893,-63,-424,-6460,368,16796,Q
43568.71,8499.23,12304.27,12063.49,56705.61,-631,-1126,301,15156,-2020,7876,W
62110.55,22610.29,31783.54,18651.32,49746.84,-166,181,-86,15092,-1604,8628,X
29478.26,17718.55,25065.28,23555.27,30265.49,1152,-1962,259,13864,-2000,9976,Y
59130.43,19875.00,26926.23,12063.49,45176.47,7940,5537,15803,5584,-292,15192,Z
29607.56,8864.70,11873.91,11826.92,39222.61,-7384,4798,-16680,512,-6452,18140,__
28968.48,8901.87,12648.65,11456.90,59853.66,-19,-995,90,15968,-3088,4744,B
35476.97,16770.19,24651.16,17441.86,41003.65,354,43,-31,15440,-4460,5360,C
39031.69,18000.00,24548.97,11411.36,39608.54,286,-180,-80,15120,-4184,7240,D
61343.28,22795.56,33714.29,26250.00,75798.82,410,-214,-102,15640,-2848,5848,E
31081.08,9165.35,12304.27,20520.83,36321.07,386,-401,-492,15668,-3832,5476,F
58774.04,25065.28,31926.61,11548.44,27506.93,-679,-504,202,7680,11452,11096,G
50021.19,19328.86,12207.45,11640.63,61343.28,482,-810,237,8180,10680,11408,H
31081.08,21535.71,27625.00,28717.95,54750.00,624,-269,-74,15516,-4392,4432,I
29868.42,19024.39,23362.50,24753.88,44019.23,11399,27905,8961,7040,-6240,9432,J
52008.73,21975.90,33560.13,11873.91,29094.83,-10,-1849,1,14836,-3200,8080,L
38653.85,14796.12,20357.14,18651.32,46875.00,288,-309,-165,15560,-3184,6252,O
47632.65,17237.39,12401.79,13522.30,38842.11,-441,198,-1096,-8348,3196,15204,P
52599.12,22610.29,30534.13,13364.14,25595.24,401,-8,-301,-5548,1556,17220,Q
44019.23,8463.30,11733.45,11594.46,52599.12,-191,-739,8,16176,-1008,5460,W
66190.48,22518.34,33103.45,20034.25,46135.45,337,-16,-31,16028,-1632,5112,X
29737.61,18358.69,25488.13,24447.30,29349.71,1209,-850,-243,14984,-2052,8456,Y
49746.84,21887.02,29221.90,12304.27,48409.09,-3570,3178,-2917,6360,7336,17144,Z
27625.00,8320.67,11320.75,11873.91,39414.89,-13617,8548,-10687,3092,-8528,16800,__
37016.95,8976.56,12063.49,12255.77,64097.94,527,-291,-756,16268,-528,3396,B
35980.07,14970.70,21362.56,17305.27,40197.84,188,89,-190,15648,-4652,4532,C
44476.74,19175.95,26811.99,12207.45,44476.74,-392,-835,565,16944,-1544,4388,D
44019.23,18142.55,25920.00,22888.89,64097.94,-174,-1197,-261,16112,-2192,3708,E
27983.19,9014.09,12304.27,22335.77,48409.09,992,-405,34,16172,-392,3148,F
68852.46,20438.80,28717.95,11780.10,29094.83,-91,-460,-264,11300,10164,7608,G
28593.75,20114.41,22610.29,30000.00,65340.31,558,-103,-538,16384,836,296,I
65340.31,20852.80,27506.93,12111.31,28470.25,811,-643,645,15468,2540,3368,L
50021.19,17237.39,23555.27,24045.80,49205.02,-1791,-1036,230,16628,-896,4404,O
53811.66,17929.18,14007.56,14396.55,43346.01,1208,-1087,462,-596,-2184,18840,P
58071.43,20194.95,30265.49,14228.57,30669.64,-256,110,-243,-684,-708,18296,Q
37016.95,8827.64,12111.31,12015.85,42905.66,376,27,218,15948,4300,1988,W
70726.26,21105.88,26472.97,20520.83,59130.43,-91,168,-198,16172,2592,2596,X
26698.37,19405.83,27041.10,26472.97,38281.25,1321,27,-803,15392,4744,-576,Y
50858.37,20769.23,24145.41,13001.82,49746.84,289,-1486,-48,11220,5956,12900,Z
31359.52,8790.70,11686.96,12849.36,38466.90,11595,4187,15001,8980,-1780,14288,__
29868.42,19795.92,24960.94,27863.13,41415.44,16166,7122,4152,4628,3208,16816,J
41623.61,19328.86,12698.56,12549.37,52302.63,-733,-274,141,12896,7716,8400,H
33406.94,8644.07,12015.85,11275.69,70250.00,-20,-89,-401,16408,-172,2072,B
```

```
38466.90,16062.75,23848.10,20438.80,48937.50,190,-239,-365,15716,-2432,6300,C
53811.66,19560.81,27506.93,12015.85,44941.41,5321,137,-642,16620,-520,7140,D
37193.88,16770.19,24245.52,24651.16,69313.19,498,-725,-656,15696,-740,6852,E
33869.43,8571.43,12159.29,26139.41,61343.28,-966,-72,117,16244,-208,5556,F
58071.43,23266.83,30399.41,11733.45,35811.26,2121,-549,-3001,8796,13592,6216,G
42471.91,20034.25,12159.29,13522.30,52008.73,-159,18,1399,11504,10016,7832,H
28103.93,19638.82,28593.75,29478.26,65340.31,1203,-162,-257,16228,132,-420,I
30399.41,20852.80,24857.14,27983.19,58774.04,11067,26554,-9996,10780,-3536,3080,J
54434.39,20936.77,27625.00,11921.05,30534.13,525,-155,-223,14904,3080,7304,L
39803.57,17373.42,22982.67,21276.60,52302.63,2880,-257,-930,13612,-4656,10108,O
44941.41,18949.11,12255.77,13628.73,50858.37,203,11,-1052,-7088,-316,17072,P
51142.25,20520.83,29478.26,14623.55,28225.35,3190,-1416,778,-5796,-212,17496,Q
35980.07,8320.67,12950.82,12111.31,34821.43,245,-814,-768,15456,3316,6232,W
52302.63,21276.60,30000.00,20194.95,50858.37,450,-132,-187,15040,2568,7296,X
31219.88,17788.46,27041.10,28968.48,51142.25,319,-97,-45,15092,5892,-620,Y
56372.09,20769.23,26361.18,11548.44,39803.57,101,631,1876,14072,6352,11156,Z
32953.12,9014.09,11873.91,11921.05,39222.61,-6140,8217,-13601,324,-3416,20868,__
29607.56,9318.54,12207.45,11780.10,92307.70,-968,-339,28,16188,900,3616,B
38466.90,13522.30,19560.81,17035.49,52599.12,2564,-456,-486,15452,-3476,6396,C
35643.57,16770.19,26472.97,11968.37,44019.23,-50,272,-474,15320,2604,7716,D
37731.96,14796.12,24145.41,22982.67,77439.76,-1219,-167,291,15124,-2904,7636,E
36321.07,9630.82,13575.42,26585.37,50021.19,-1259,-432,-84,16248,2852,3972,F
58071.43,23652.39,31081.08,11456.90,28717.95,-196,-765,-614,11280,10976,7952,G
43125.00,18949.11,13104.40,12063.49,59130.43,2219,-63,-573,12576,8372,8588,H
32953.12,18949.11,28842.86,28103.93,46380.00,430,-33,-217,16184,2344,704,I
28593.75,19638.82,27863.13,28470.25,48672.20,11122,32767,-18168,7568,1688,7472,J
60220.59,21975.90,29737.61,12207.45,31219.88,3436,-1444,-2033,15800,2420,4876,L
48148.15,18071.12,24145.41,23652.39,61725.00,902,-557,-316,16044,-3264,4272,O
44476.74,20520.83,14173.00,13682.25,35643.57,-4728,-1453,-400,-4080,-3208,18712,P
52302.63,19795.92,26472.97,14566.47,40197.84,339,-739,54,-6156,164,17400,Q
39031.69,8864.70,12500.00,11921.05,42471.91,-41,-32,-157,15760,3516,4292,W
57382.08,21191.04,28968.48,18949.11,58421.05,-155,-814,156,15248,3920,4360,X
31081.08,17441.86,23362.50,24045.80,32655.28,-1112,246,309,14232,5656,-2776,Y
50858.37,21975.90,28842.86,12111.31,53200.00,-1213,-1021,-3084,12932,3156,11712,Z
31926.61,8827.64,11548.44,11780.10,30265.49,-5442,8953,-20264,572,-4696,19492,__
31359.52,9203.47,11968.37,11968.37,65763.16,4829,1137,-3147,17172,804,1232,B
34025.56,12598.92,18949.11,19328.86,55714.29,-1266,-10,-122,15388,-3288,6996,C
48409.09,20520.83,26139.41,11594.46,42905.66,-1068,-691,-36,15384,-36,7760,D
46626.50,17648.94,25811.17,25170.16,80906.24,-502,-423,-214,14792,-3160,8456,E
36321.07,9357.14,14396.55,27272.73,55714.29,706,-583,-326,16104,1080,4424,F
43793.11,20034.25,25065.28,12255.77,32215.38,-23,-127,104,7756,13276,7600,G
32953.12,18504.37,12748.64,12304.27,62110.55,-462,273,-105,10052,12224,6820,H
31783.54,19328.86,27983.19,28717.95,64921.88,-847,-715,68,15908,2624,-1220,I
31641.34,17510.59,24753.88,29868.42,60220.59,2018,25307,-15245,11032,-4012,-2248,J
62893.40,22982.67,31500.00,11921.05,28968.48,469,-383,204,15216,2508,4896,L
34821.43,15029.35,21276.60,21021.13,58421.05,1266,-296,-744,13056,-9680,40,O
46626.50,19100.00,15147.35,12950.82,41415.44,179,-478,-461,-7644,1212,16320,P
55714.29,20194.95,26139.41,12352.94,38466.90,-611,80,-836,-5052,2312,17592,Q
36321.07,8214.83,12500.00,12352.94,44476.74,164,67,-202,15540,2524,6768,W
57382.08,21021.13,32215.38,19717.19,49205.02,417,-1468,-546,15904,2620,5516,X
31500.00,16836.10,26472.97,25275.59,33869.43,-2490,-1455,1273,15624,4424,-1636,Y
52898.23,20769.23,25381.58,11968.37,44019.23,6005,12316,8643,-604,4008,20444,Z
33714.29,8571.43,11968.37,12352.94,39803.57,2477,7289,11849,6224,2852,16036,__
31641.34,8864.70,12549.37,11780.10,63290.82,436,-311,-129,16388,-56,2756,B
36841.21,15751.50,22518.34,21887.02,46135.45,-255,-248,90,14788,-3464,8428,C
40000.00,19175.95,23171.64,12111.31,46875.00,-1122,-776,927,15104,-1348,8364,D
43568.71,16902.29,25065.28,25488.13,78000.00,905,-766,-182,14824,-852,9372,E
33869.43,9473.68,12450.81,28470.25,53504.46,156,-120,-440,16064,-360,4856,F
46135.45,22065.22,29221.90,11733.45,31641.34,-15,144,93,14152,7488,5228,G
35811.26,18286.34,12648.65,12748.64,57725.12,1161,-861,-807,11020,6736,11364,H
35643.57,19405.83,26472.97,30132.36,55068.49,1286,-217,-716,16356,1272,2844,I
31926.61,17237.39,23458.65,29478.26,55714.29,5563,-7872,19564,9392,14328,10884,J
57042.25,22335.77,28347.46,12015.85,29478.26,141,-246,-285,15988,2744,4336,L
42257.46,17035.49,22426.83,22426.83,54434.39,941,-647,-505,14524,-5132,7868,O
53811.66,17858.67,12207.45,11921.05,47125.50,1536,-445,-59,-13876,3436,10812,P
```

```
47125.50,20357.14,28717.95,14396.55,34182.70,117,27,-319,-8396,1288,16176,Q
45176.47,8976.56,12748.64,11826.92,39222.61,-282,-13,-177,15648,3792,5024,W
64507.77,21710.53,24651.16,18725.27,60591.13,-438,-230,-220,15800,1276,6256,X
30943.12,17102.51,23652.39,24857.14,41415.44,-1234,314,88,15152,6416,248,Y
54121.62,20686.05,26029.41,12063.49,52302.63,3793,4008,613,17428,3748,3576,Z
33560.13,8753.87,11873.91,11502.59,31081.08,-4389,-5807,-12280,-4144,-2484,20516,__
29349.71,8717.16,11780.10,11968.37,60965.34,637,-313,-898,15764,-4040,32,B
44708.17,15326.08,23171.64,18504.37,47889.34,-260,-244,132,14224,-5216,8180,C
52599.12,18504.37,21535.71,11780.10,50576.93,671,-46,-35,15260,-4916,6140,D
69313.19,21362.56,32507.74,24753.88,67500.00,633,-801,59,15040,-6808,4732,E
37731.96,9203.47,12598.92,20357.14,61343.28,1850,-1701,-510,15496,-5996,4756,F
81509.44,23946.70,27625.00,12015.85,33560.13,345,407,-317,7404,13952,3812,G
52599.12,20438.80,11921.05,12063.49,58421.05,-1133,-1022,-1638,2484,15744,7216,H
28842.86,20034.25,24548.97,28842.86,62500.00,2057,506,-669,14996,-6668,-2768,I
31081.08,19717.19,23848.10,27156.59,56372.09,11513,14339,-4368,13684,-6280,-2800,J
84000.00,20769.23,27506.93,12950.82,34821.43,859,-1966,123,14788,-6064,-3220,L
38281.25,15386.14,21105.88,19328.86,39608.54,19,-464,-790,14968,-1296,6824,O
58071.43,19024.39,11594.46,14228.57,36666.67,-1572,-191,-209,-12276,3600,13400,P
52302.63,21191.04,26585.37,14796.12,35980.07,1959,60,-123,-10756,3684,14284,Q
50297.87,8499.23,11640.63,11594.46,54434.39,139,-257,-451,15620,-4364,4524,W
65763.16,22426.83,31500.00,23362.50,48409.09,61,-24,533,16268,180,5600,X
27389.50,20194.95,25702.92,25065.28,30265.49,937,1138,27,16504,2444,888,Y
50021.19,18214.29,26250.00,12111.31,49746.84,1009,2940,-10936,4616,-4724,14104,Z
34340.84,8320.67,11275.69,11968.37,37193.88,270,957,-7551,1276,-1720,18416,__
28968.48,9014.09,11968.37,11275.69,68852.46,-128,-431,-2267,15744,3128,-360,B
35980.07,14117.65,24245.52,20275.86,46380.00,-2377,-606,-363,15652,-3012,6272,C
44019.23,18651.32,26472.97,11186.01,51717.39,1254,-531,212,15884,-712,6884,D
63692.31,19954.44,30000.00,27506.93,75798.82,1633,-1006,-995,16220,160,5208,E
32803.74,8680.55,11826.92,24960.94,48409.09,-198,-1798,985,15904,-3436,6336,F
72685.71,25811.17,27625.00,12255.77,26472.97,766,-2637,14,6588,13500,7764,G
50858.37,20686.05,11780.10,11640.63,58774.04,-23,-220,-1451,4372,15020,9032,H
31359.52,20603.25,23750.00,29221.90,53811.66,58,-2694,964,14636,-7368,368,I
26472.97,21622.91,27625.00,24753.88,41833.33,12872,6843,1253,9564,-5532,10296,J
82120.26,22702.70,33714.29,13682.25,34500.00,-1271,2070,1472,12936,-7856,-4028,L
35811.26,13898.31,19100.00,18142.55,39803.57,-1324,496,113,12904,-9956,-404,O
43568.71,17237.39,12159.29,14062.50,37016.95,-2333,-730,-1910,-11880,-2612,11004,P
52302.63,23076.92,30534.13,16000.00,37193.88,1851,960,-1006,-10788,-3988,11448,Q
49205.02,8179.76,11640.63,11502.59,34983.71,-2758,-1109,-2280,14684,-3556,8488,W
68396.74,23362.50,33254.72,16639.18,43125.00,-8415,-791,628,14592,-3272,9036,X
29221.90,19100.00,23362.50,25811.17,32655.28,-2345,-3825,-1080,15548,-5224,976,Y
41208.79,22245.15,28225.35,12798.91,46380.00,3213,5793,9596,7856,8304,24928,Z
30805.97,8535.28,11411.36,11873.91,42257.46,-4039,2114,-6352,2264,-6064,17404,__
29094.83,8976.56,12015.85,11921.05,58071.43,-1956,-273,799,16376,-508,2328,B
35980.07,14566.47,23750.00,18358.69,40000.00,-6062,647,-1814,14180,-6912,6660,C
44941.41,18799.56,23750.00,11686.96,52898.23,2293,-1150,24,16176,-2364,3172,D
70250.00,22518.34,34983.71,28593.75,79722.22,4710,172,1391,16552,-468,3912,E
34182.70,8976.56,12748.64,22518.34,53811.66,-6091,-2543,281,14540,-1888,6232,F
62500.00,22610.29,28225.35,12401.79,37372.02,1151,882,1138,8600,13784,6148,G
45892.86,22335.77,12015.85,11502.59,65763.16,1195,459,-908,5772,14180,6524,H
28717.95,20852.80,25065.28,25065.28,54750.00,-612,-1588,827,14080,-6716,-3208,I
28347.46,19483.14,22982.67,25275.59,44941.41,12539,7715,866,8920,-7792,13716,J
67945.95,22518.34,30399.41,13104.40,28103.93,-509,-5236,483,14396,-8508,1576,L
42257.46,16000.00,22335.77,20852.80,52302.63,3146,1544,-1241,16384,-1608,3680,O
46135.45,17788.46,11365.98,12450.81,30265.49,27,-109,75,-14392,1656,9400,P
52302.63,21975.90,34660.20,16125.76,38281.25,8158,-3575,1360,-8748,-3472,16740,Q
50858.37,8827.64,11968.37,11594.46,45652.17,-1881,-903,220,16216,-1536,4436,W
75798.82,22982.67,28968.48,12450.81,43125.00,1931,1236,382,15064,-2088,7812,X
29094.83,18949.11,21887.02,24145.41,34340.84,-1389,-2474,444,14800,-7832,808,Y
38842.11,20357.14,26698.37,11780.10,33103.45,8748,281,14402,40,-3996,16432,Z
30943.12,8427.48,10920.61,11320.75,30399.41,8813,-3101,-6732,116,1460,20740,__
28347.46,9434.71,12255.77,11826.92,61343.28,-636,-62,195,14768,-5824,-1352,B
37913.79,18799.56,29221.90,20438.80,40197.84,-1547,-733,-597,14216,-7752,4552,C
49205.02,17788.46,24753.88,11733.45,38281.25,2595,494,1137,15048,-5004,5696,D
66190.48,21191.04,30669.64,27863.13,82120.26,7475,-2363,1516,15200,-3440,3156,E
34660.20,8391.77,11733.45,23652.39,56041.67,3779,-1510,-2607,15708,-3324,4580,F
```

```
54121.62,21798.56,28593.75,12450.81,29868.42,6568,-2580,-5004,7636,11508,10312,G
37551.38,20194.95,12255.77,11968.37,53200.00,-3266,-249,1058,5764,12484,12612,H
28968.48,20357.14,23555.27,23848.10,71207.87,2454,1683,-405,14416,-7004,-4612,I
28225.35,16574.07,23848.10,24245.52,46626.50,11215,10669,11743,4892,4440,32736,J
64921.88,21710.53,30000.00,13628.73,32507.74,-2425,-300,560,13652,-7716,-2560,L
37193.88,13575.42,18358.69,20194.95,41208.79,-451,-902,104,14608,-7432,1536,O
55068.49,17305.27,11365.98,12111.31,31926.61,395,119,-976,-13684,-700,11120,P
69313.19,22610.29,32507.74,16380.37,35811.26,-3708,-135,-540,-8308,-5744,15676,Q
58421.05,8901.87,11873.91,12648.65,49746.84,-3769,-286,-123,15412,-5000,-868,W
79722.22,24548.97,33406.94,19100.00,42471.91,1731,859,-1047,14928,-7032,5028,X
29221.90,19560.81,26811.99,26250.00,34340.84,1004,-350,-995,16204,-2372,820,Y
66622.34,21535.71,28225.35,12698.56,43568.71,4765,5410,14716,4260,-12600,18288,Z
30669.64,8427.48,11826.92,12450.81,41833.33,4263,-827,-1621,5432,-11516,13212,__
33103.45,9127.36,11921.05,11780.10,56372.09,-147,-914,-1298,14396,-7072,2048,B
38653.85,16252.54,24346.15,18725.27,43346.01,150,-298,-74,13764,-7260,6836,C
38281.25,18000.00,23362.50,11548.44,47125.50,-155,-405,-273,14848,-6368,4688,D
66190.48,23076.92,32507.74,27743.73,73189.65,-1575,409,-352,15588,-5292,4000,E
35476.97,8391.77,11686.96,24651.16,46875.00,256,-438,-240,15020,-5392,4180,F
68852.46,24548.97,29094.83,12111.31,30132.36,263,-74,-640,11180,10016,8044,G
59853.66,21710.53,12255.77,12849.36,47632.65,-1189,-950,-113,6464,13604,8116,H
30669.64,19175.95,23458.65,24447.30,76886.23,4026,2659,-405,15740,-3816,1564,I
29349.71,21362.56,28968.48,26698.37,52302.63,6531,188,1731,1660,472,-5812,J
64507.77,24651.16,30805.97,12950.82,32361.11,1589,-198,-162,13896,-7224,824,L
46135.45,14796.12,21276.60,19638.82,47378.05,1742,2063,-686,14308,-5540,6684,O
61343.28,19024.39,11594.46,12798.91,25702.92,130,-1534,-553,-14356,-468,9620,P
63290.82,24045.80,34983.71,16189.02,38466.90,-1789,1809,-830,-10628,-3172,14572,Q
60591.13,8976.56,11780.10,11686.96,48937.50,-1275,-340,-1198,14836,-1100,8916,W
61343.28,24346.15,30669.64,19638.82,54750.00,-423,80,146,16372,-68,4908,X
27156.59,18949.11,25488.13,25065.28,29094.83,-4935,3282,301,13440,4492,6984,Y
51428.57,21622.91,27743.73,12748.64,39803.57,-1238,715,-10386,7896,-848,13064,Z
32655.28,8463.30,10833.33,11780.10,34821.43,-3103,-1962,-13500,-5080,-17240,24568,__
28347.46,8790.70,12255.77,11275.69,81509.44,938,148,-20,14800,-4656,6816,B
36666.67,18071.12,28103.93,17858.67,36841.21,949,-223,-97,13772,-6892,8100,C
39803.57,20114.41,25275.59,11052.63,36666.67,-370,-752,285,14168,-5396,8012,D
76339.29,26029.41,37731.96,28470.25,72187.50,650,-455,743,13688,-5908,8716,E
31359.52,8827.64,11780.10,21021.13,39803.57,996,-957,-555,14228,-5648,7672,F
72187.50,27743.73,34025.56,12159.29,31219.88,1622,-559,-229,5488,12424,11616,G
55068.49,18725.27,12111.31,11873.91,58774.04,1246,77,-956,8796,11380,9256,H
31359.52,25065.28,29221.90,32507.74,58774.04,233,-76,-550,14932,-6572,4328,I
29737.61,23750.00,26361.18,28470.25,53811.66,11118,14943,-473,7252,-5352,-588,J
58421.05,24346.15,38466.90,12401.79,26250.00,354,-1190,-131,13324,-4756,10456,L
41415.44,16444.67,22982.67,18286.34,38466.90,1146,-1158,578,14604,-6404,6692,O
51428.57,18949.11,12549.37,13735.96,46626.50,-729,-814,-772,-9068,6012,14160,P
66190.48,26811.99,31641.34,13575.42,24045.80,207,-199,-355,-4644,2156,17444,Q
44476.74,8391.77,11733.45,11320.75,59853.66,-395,1291,-152,15000,-4096,7952,W
80906.24,26029.41,36841.21,21710.53,47125.50,201,110,-149,15160,-3640,7760,X
29607.56,21535.71,28717.95,27156.59,27625.00,-252,-89,-221,13552,-3920,10820,Y
69313.19,23848.10,28593.75,12207.45,48409.09,-11577,-486,-17412,3332,-5380,18632,Z
28103.93,8644.07,11594.46,12111.31,38096.89,782,-395,723,6196,-1324,18124,__
30943.12,9951.22,13364.14,11921.05,65340.31,-586,-1284,718,14400,-5428,7692,B
37731.96,19175.95,29094.83,18142.55,35476.97,-1836,-627,-2468,14432,-6728,8960,C
48148.15,20194.95,25381.58,11968.37,47125.50,-1708,240,-641,13892,-6544,8720,D
67500.00,24857.14,37372.02,30265.49,72685.71,384,-146,-152,13200,-6716,9148,E
30132.36,8680.55,12063.49,21975.90,38096.89,480,-265,-255,14588,-6088,6544,F
70250.00,27041.10,38653.85,12159.29,27506.93,1882,-431,-567,6752,10828,13676,G
51142.25,21710.53,11968.37,11873.91,70250.00,-231,-615,342,8072,10284,11440,H
29607.56,24548.97,28225.35,30943.12,59490.29,283,-107,-145,14816,-6776,4896,I
28470.25,19405.83,24346.15,24145.41,38466.90,32767,1889,17617,6000,31956,18676,J
61343.28,23848.10,39031.69,13104.40,26698.37,272,906,40,13832,-4704,11472,L
38096.89,16189.02,24447.30,19252.23,46626.50,1823,-516,-626,14348,-7196,4816,O
38281.25,16316.32,11968.37,13522.30,60591.13,-1291,-1436,-1468,-10624,3864,13296,P
63692.31,24548.97,34025.56,13469.39,25811.17,-118,-628,-420,-6848,3488,16804,Q
45176.47,8753.87,11686.96,11733.45,68852.46,-2881,1303,-204,14268,-2752,10036,W
62500.00,25170.16,34983.71,20852.80,46380.00,552,-313,-70,14976,-4004,7596,X
29221.90,20275.86,27743.73,28225.35,27506.93,349,66,-550,13284,-3872,11116,Y
```

```
51142.25,22154.96,31219.88,12401.79,50021.19,-16280,4451,-22751,1824,-4952,15688,Z
29221.90,8463.30,11275.69,12401.79,39803.57,-6451,2309,-8333,4016,-5692,18364,__
29737.61,8976.56,13104.40,11686.96,65340.31,393,-431,-17,15032,-5424,5796,B
31500.00,16574.07,26926.23,16770.19,34182.70,492,-626,-53,14028,-6848,7428,C
41833.33,18799.56,25595.24,11594.46,39608.54,-5,-450,-100,15056,-4968,6000,D
62110.55,22065.22,32953.12,25381.58,78567.07,-19,-17,-270,14956,-4324,7420,E
31359.52,9165.35,11921.05,20438.80,39031.69,160,816,-857,15444,-4920,4812,F
75798.82,25702.92,35476.97,12849.36,30132.36,2223,-1449,-94,5844,14024,8916,G
50858.37,19560.81,12015.85,11780.10,73189.65,-917,214,67,4672,13448,9388,H
30805.97,25488.13,29221.90,33560.13,62500.00,212,-456,351,15052,-6932,2616,I
27041.10,20034.25,22702.70,28470.25,47125.50,-32768,3160,6019,14556,-2044,7228,J
52302.63,22518.34,40000.00,13104.40,28225.35,1748,-644,167,13480,-6488,9716,L
40800.00,15937.50,22335.77,19175.95,40800.00,469,-124,-344,15564,-4240,4996,O
41623.61,15813.25,12748.64,13682.25,52599.12,73,-693,-387,-6944,5488,15844,P
69313.19,24045.80,34500.00,14452.98,23652.39,180,-386,-380,3348,708,18120,Q
45176.47,8391.77,12255.77,11826.92,54121.62,1054,-1657,-883,15396,-2712,7292,W
62500.00,24857.14,38096.89,21975.90,44019.23,-6169,407,-184,16132,-1292,5488,X
27983.19,19024.39,28103.93,26250.00,28470.25,-760,-666,320,15060,-2324,8376,Y
60220.59,21622.91,31641.34,12648.65,49474.79,-11590,2808,-7023,11280,-8236,14108,Z
28103.93,8790.70,12352.94,13155.96,41415.44,-6825,1888,-12027,-1080,-5324,20244,__
28103.93,8499.23,12450.81,11456.90,62500.00,-72,-50,-390,15140,-5096,6172,B
40197.84,18577.68,27272.73,18725.27,38842.11,187,-418,-252,14764,-4716,7584,C
48148.15,19252.23,25595.24,11594.46,38842.11,-1328,-377,340,13804,-4752,9348,D
61343.28,24346.15,35311.48,26139.41,87299.99,355,-347,-257,14744,-4028,8224,E
28717.95,8901.87,11640.63,23362.50,42905.66,581,-1527,-406,15220,-3248,7868,F
81509.44,26361.18,36666.67,12207.45,28225.35,-5057,-1289,-526,7576,12956,8688,G
54750.00,21535.71,11968.37,11640.63,84000.00,727,97,-1298,7304,11820,11488,H
30805.97,26361.18,29737.61,33869.43,82120.26,973,-2025,-648,15388,-4764,5288,I
30399.41,22154.96,28347.46,30265.49,50297.87,32767,7797,15118,8924,11576,13288,J
60220.59,24447.30,36841.21,12401.79,27506.93,-319,-293,-126,15136,-2652,7496,L
38653.85,16062.75,25488.13,19175.95,46875.00,248,-575,-138,15672,-3540,4340,O
42044.61,17718.55,12207.45,16125.76,59853.66,-76,-220,-308,-1728,3588,18052,P
66622.34,24548.97,34500.00,13843.99,24651.16,368,101,-126,-1748,3040,18328,Q
48148.15,8717.16,12207.45,11826.92,56041.67,175,264,-153,15652,-2096,6664,W
61343.28,24548.97,33714.29,20852.80,47378.05,179,-241,-217,15656,-2400,6248,X
28717.95,19252.23,25488.13,25170.16,27272.73,-378,-744,-191,15048,-1988,8556,Y
52302.63,23652.39,30669.64,12500.00,44941.41,-3841,1701,-4771,9848,-6916,15988,Z
28470.25,8427.48,10833.33,11968.37,43346.01,-1195,139,-223,5712,-2156,16956,__
29737.61,9014.09,12849.36,11411.36,72187.50,-46,772,-513,15528,-5056,3836,B
36841.21,19024.39,29868.42,18504.37,45413.38,-239,-1056,557,14512,-5732,7424,C
44476.74,19405.83,27156.59,11686.96,41415.44,1787,-2098,-913,14796,-4336,8280,D
66190.48,24753.88,36493.29,27041.10,78567.07,473,151,-215,14132,-5324,8764,E
30669.64,9014.09,11968.37,21191.04,41415.44,25,-958,286,15520,-4972,4452,F
81509.44,28347.46,36493.29,12255.77,26250.00,-279,-778,-198,7116,11684,11296,G
61343.28,21710.53,11921.05,11733.45,86622.52,-1309,361,-629,6756,10220,11680,H
30805.97,26250.00,29094.83,33103.45,68396.74,820,-789,-1260,15960,-5188,4184,I
28103.93,22335.77,23266.83,24651.16,46135.45,16869,14700,-11696,9596,-5968,1808,J
58774.04,23266.83,40197.84,12401.79,27041.10,473,-339,-280,15300,-3128,6940,L
40000.00,15690.00,22610.29,18799.56,41623.61,-57,294,-287,15308,-4784,5192,O
43125.00,17169.81,12900.00,14396.55,59490.29,-246,579,-1246,-9792,5324,15704,P
53504.46,24651.16,32803.74,13789.87,24548.97,-2,-79,-117,-6508,2372,16996,Q
53811.66,8571.43,11826.92,11548.44,52898.23,127,-1049,166,15300,-1276,7688,W
65340.31,25065.28,35980.07,21276.60,53200.00,152,-1584,1685,15944,-3068,3120,X
26698.37,19175.95,27863.13,25811.17,28225.35,112,-2469,-293,14980,-1984,8900,Y
50021.19,22518.34,32070.55,11826.92,49746.84,436,354,-773,10936,-4112,13964,Z
28103.93,7834.82,12111.31,12748.64,39608.54,-270,-1351,2962,6752,1684,17260,__
28103.93,9051.73,13001.82,11826.92,71694.91,260,-2862,1160,15000,-2996,9012,B
31783.54,16509.24,26698.37,16316.32,34821.43,614,-368,-664,15196,-5232,6208,C
36666.67,19328.86,25595.24,12159.29,48148.15,-450,-762,146,15248,-4392,6196,D
54434.39,22335.77,31359.52,24651.16,82120.26,374,-356,-137,15044,-4240,7400,E
35980.07,10032.63,13053.01,21191.04,46135.45,-904,-232,103,15276,-4596,7048,F
73699.42,28593.75,34340.84,12401.79,33714.29,2169,1545,896,2912,15884,6332,G
57382.08,21975.90,12401.79,11921.05,67500.00,1059,-267,-69,2240,14616,9192,H
30534.13,25702.92,31081.08,32361.11,58774.04,-567,-2519,-522,14844,-6548,5184,I
29737.61,23076.92,27272.73,26698.37,44476.74,16175,-19412,14346,15464,20564,14344,J
```

```
62110.55,23848.10,39031.69,12900.00,31219.88,-76,-2308,564,14652,-3556,9432,L
39414.89,16836.10,23076.92,20686.05,32361.11,-121,720,-402,15404,-5824,4116,O
49205.02,19100.00,14623.55,15506.96,66622.34,466,298,-151,-7768,3724,16812,P
72685.71,25381.58,34182.70,14396.55,26585.37,-301,458,103,-7568,-636,16712,Q
57725.12,8499.23,12900.00,12450.81,50297.87,303,-262,-331,15952,-1016,6328,W
57725.12,25488.13,36493.29,20520.83,47889.34,215,-179,21,16120,-1252,4576,X
29478.26,20936.77,31783.54,28225.35,29094.83,2067,-1199,-666,15036,-1600,8904,Y
46135.45,21105.88,29478.26,12500.00,58071.43,-5823,3429,-18179,4188,-732,17716,Z
28968.48,8644.07,12255.77,12698.56,38466.90,616,-409,295,6800,-44,15112,__
29221.90,9473.68,13364.14,12207.45,73189.65,590,-1983,-432,15476,-5868,2724,B
32070.55,16639.18,26361.18,17373.42,29094.83,-696,-3266,-1569,14556,-6984,4200,C
42044.61,17718.55,23362.50,12159.29,36150.00,101,-229,-203,15736,-5964,3408,D
37551.38,21276.60,28717.95,24753.88,92307.70,661,84,776,15596,-5192,2948,E
32070.55,9203.47,12111.31,20686.05,37193.88,216,-408,-93,15348,-5808,1816,F
53504.46,24245.52,30534.13,12255.77,36150.00,3234,-1100,1425,704,15320,7684,G
62110.55,21798.56,12159.29,11502.59,72685.71,2011,-522,1437,3836,12644,11376,H
30534.13,24857.14,27506.93,29607.56,55714.29,102,-520,-377,14652,-6080,-2312,I
28593.75,22335.77,22888.89,24548.97,33869.43,-3347,-926,-1344,12892,-1520,788,J
63692.31,24045.80,38653.85,13053.01,33103.45,179,-173,165,16164,-3652,3756,L
41415.44,16704.55,21887.02,19405.83,34983.71,476,-455,-560,15524,-5496,1788,O
51428.57,20520.83,12598.92,13053.01,64921.88,3534,172,1233,-6864,-1088,14772,P
68852.46,26361.18,35643.57,13053.01,24245.52,72,-157,-111,-5920,-1916,16964,Q
57382.08,8607.69,12500.00,12255.77,58421.05,378,-1489,-719,16124,-4144,2056,W
53200.00,24145.41,33560.13,22426.83,58421.05,408,-53,-170,15504,-4200,-704,X
28593.75,19795.92,27625.00,25170.16,28103.93,-156,-927,-444,15756,-4276,4920,Y
57725.12,23266.83,32803.74,12549.37,63692.31,-2004,8902,9457,7024,13960,12916,Z
28103.93,8717.16,12450.81,12549.37,44476.74,-3572,10773,-16519,4340,-10588,17516,__
28103.93,8499.23,12063.49,12304.27,76886.23,517,-626,-147,15056,-6112,-460,B
33560.13,18286.34,30534.13,19175.95,35811.26,716,3860,1632,14500,-7344,944,C
44019.23,19954.44,25488.13,13001.82,37731.96,-1127,-3237,-407,14492,-8160,-696,D
37193.88,18577.68,29094.83,22245.15,84642.86,-3750,3718,1166,14304,-5220,1156,E
30805.97,9051.73,12159.29,24447.30,39031.69,198,-828,-411,15400,-5668,884,F
56705.61,23458.65,35147.06,12401.79,33254.72,-161,-243,23,5432,13904,10076,G
70250.00,23362.50,15326.08,13053.01,85294.12,973,-1161,-2528,9000,8752,5208,H
33103.45,24960.94,29737.61,34983.71,70726.26,441,-209,-538,14100,-5540,-4048,I
29607.56,24045.80,26250.00,28347.46,43346.01,16549,-10002,24399,19048,11872,6420,J
78000.00,24045.80,38466.90,13001.82,34660.20,-430,-2060,228,15440,-5296,1348,L
45176.47,17035.49,22335.77,20114.41,32803.74,-218,-3039,193,14452,-7132,-1024,O
57382.08,23458.65,13469.39,13416.67,62500.00,-206,7862,-2810,-6140,2252,20772,P
72187.50,24960.94,35980.07,13364.14,23458.65,154,4441,65,-5164,600,21216,Q
52599.12,8644.07,12111.31,12304.27,58421.05,1060,-1248,-4,15636,-4880,1040,W
61343.28,29737.61,37551.38,15813.25,52302.63,-1993,-2392,639,18640,-2420,3744,X
28347.46,21362.56,28103.93,28470.25,33406.94,97,-441,-31,15332,-5760,1392,Y
54121.62,22888.89,29094.83,12598.92,55068.49,2718,1921,6179,5360,11308,17276,Z
29607.56,8535.28,12063.49,12549.37,41415.44,706,-225,1585,4612,1240,17936,__
30399.41,9089.48,13259.67,12500.00,72187.50,-955,-326,-330,15624,-5008,292,B
28842.86,16062.75,26811.99,16252.54,35643.57,-1,-1212,-19,14580,-7264,4564,C
44708.17,19483.14,25381.58,12015.85,35311.48,5291,1275,1704,15848,-4308,2568,D
42044.61,20438.80,34500.00,21798.56,125779.81,450,-565,162,14944,-6256,3832,E
28968.48,9127.36,11640.63,22702.70,32953.12,111,-96,-95,15536,-5876,1952,F
75798.82,26029.41,37731.96,13207.72,30132.36,-292,-372,12,1992,15048,8812,G
72685.71,22426.83,12698.56,11640.63,70726.26,-318,30,36,5804,14076,9288,H
33406.94,25488.13,31926.61,30805.97,69313.19,1486,1253,-1003,13708,-7900,-2524,I
28103.93,24753.88,27506.93,34500.00,42905.66,32767,-15995,16661,18528,23700,13788,J
59853.66,22245.15,38466.90,13575.42,27863.13,474,-699,91,15404,-4984,2648,L
47378.05,17237.39,23171.64,19795.92,47378.05,292,-405,-610,14428,-7684,1160,O
59853.66,22335.77,12015.85,14173.00,65340.31,-1147,1605,-1802,-9876,1008,23400,P
57382.08,25920.00,34983.71,14062.50,23848.10,-2261,-10004,432,-216,-924,19232,Q
58421.05,8680.55,12207.45,11826.92,58071.43,425,-390,-66,15620,-4800,1592,W
72685.71,24045.80,34983.71,20194.95,47889.34,96,-360,305,16268,-3716,1040,X
27506.93,18799.56,24346.15,24145.41,32953.12,564,-132,-223,15392,-4112,5120,Y
48937.50,21362.56,27983.19,12698.56,50021.19,3585,9599,18142,6860,-6040,17440,Z
28593.75,8644.07,11733.45,11826.92,29737.61,744,1367,3090,1076,-2072,18288,__
74215.13,28103.93,34660.20,32070.55,41003.65,544,-843,66,15856,-2264,6472,A
73699.42,25381.58,29868.42,21021.13,39222.61,-404,-1716,264,16176,3180,3824,T
```

```
67945.95,21710.53,28225.35,28225.35,64921.88,-289,-517,590,16216,-880,6140,M
67945.95,24346.15,33254.72,33714.29,64921.88,1024,-422,-949,15672,-2984,5960,S
69313.19,24651.16,31359.52,28968.48,40197.84,165,-890,-820,15460,-1116,8608,A
65340.31,24145.41,30132.36,17788.46,37193.88,81,-71,-104,15796,3600,4464,T
66622.34,26361.18,32803.74,34983.71,66622.34,317,-785,-470,14912,-3856,7236,S
65763.16,20034.25,24045.80,24651.16,63290.82,396,250,-216,15496,-1780,7692,M
77439.76,28347.46,34983.71,31081.08,38842.11,763,75,-109,15692,-3296,5528,A
62893.40,24548.97,31926.61,19638.82,54434.39,459,-322,-68,16284,-2004,1704,T
71207.87,24857.14,32953.12,34500.00,65763.16,21,-221,28,16348,-204,1720,S
65763.16,21710.53,23848.10,24045.80,72685.71,267,208,-420,15992,-3928,2776,M
89387.77,27625.00,32953.12,30132.36,44476.74,2580,46,-405,15248,-3800,3152,A
71207.87,24857.14,30534.13,20034.25,56705.61,-627,112,42,16340,-416,1756,T
62500.00,20520.83,25702.92,25811.17,81509.44,-417,-870,-34,16184,-428,2860,M
63290.82,24753.88,34340.84,33560.13,56705.61,251,68,57,16112,-1948,2916,S
64921.88,24960.94,31641.34,29094.83,33560.13,959,-530,-402,16260,-2880,3176,A
59490.29,23848.10,29221.90,19175.95,59853.66,133,-284,-55,16488,64,1736,T
67945.95,19795.92,25381.58,24960.94,66622.34,299,-8,-187,16236,288,4012,M
62893.40,24346.15,32655.28,33103.45,61343.28,-22,-110,-167,16396,260,3976,S
69313.19,24651.16,31641.34,28103.93,37551.38,302,-355,-139,16212,-2544,2844,A
60220.59,22795.56,28103.93,19954.44,58774.04,412,-429,98,16460,-480,1468,T
63290.82,19483.14,23750.00,24346.15,79141.10,191,-189,-281,16356,488,2412,M
66622.34,24245.52,31783.54,32803.74,74736.84,870,-1149,739,16308,-260,2620,S
75798.82,24548.97,30669.64,28717.95,36321.07,612,-568,13,16228,-1156,4652,A
63290.82,23555.27,30265.49,18874.17,59853.66,-694,-90,126,16604,328,2868,T
64921.88,22795.56,32655.28,34340.84,60220.59,275,12,-39,16260,-592,3220,S
64507.77,18142.55,21021.13,22610.29,73699.42,256,-59,-204,16348,420,4104,M
73189.65,24045.80,30805.97,28593.75,38281.25,467,-312,-150,16180,-2496,4416,A
69313.19,22888.89,28593.75,18799.56,62500.00,819,-436,-77,16268,-916,2872,T
69313.19,23652.39,31081.08,32953.12,62500.00,176,-416,-346,16324,-1284,2428,S
55068.49,20034.25,22065.22,23266.83,73699.42,181,-41,38,16428,-1628,2612,M
67945.95,24548.97,32070.55,27389.50,36666.67,123,-373,-261,15920,-1800,6016,A
56705.61,22518.34,29607.56,17102.51,57042.25,18,-94,-156,16248,-1184,3792,T
64921.88,23362.50,31641.34,32803.74,56041.67,1474,-960,-86,16432,-528,3300,S
61343.28,18431.37,23652.39,23171.64,77439.76,-355,900,-16,16272,-4,4280,M
84000.00,26139.41,32361.11,29868.42,36321.07,-1627,-722,612,16156,-2188,4044,A
70726.26,23266.83,27743.73,17237.39,56372.09,1010,-102,-131,16292,-784,1884,T
63290.82,24346.15,32070.55,33254.72,56705.61,1698,-573,112,16348,-1556,1828,S
58774.04,17718.55,23458.65,23171.64,71207.87,-1429,75,197,16356,4,3612,M
77439.76,26585.37,32361.11,30132.36,33254.72,157,-1378,812,15360,-3648,5244,A
67945.95,24548.97,27272.73,18000.00,56041.67,195,-1122,180,16184,-1104,2876,T
66622.34,24651.16,31783.54,33560.13,60220.59,892,-645,113,16480,-148,3168,S
59853.66,18431.37,22610.29,22610.29,72685.71,631,-1488,1079,16228,864,4388,M
77439.76,24651.16,30132.36,27863.13,43568.71,-489,-2132,703,15740,-3000,5904,A
58071.43,23076.92,29478.26,17648.94,62500.00,-400,-928,1222,16512,-468,2872,T
58421.05,19328.86,23171.64,24346.15,77439.76,-731,-1832,-109,16488,-432,4360,M
63290.82,24346.15,31500.00,33103.45,67058.82,-81,-511,78,16296,-1288,3528,S
89387.77,27506.93,32361.11,30943.12,43568.71,648,-1076,126,15744,-2336,7212,A
70250.00,24045.80,30000.00,19328.86,62500.00,-1012,-1036,726,16220,-364,4016,T
60591.13,19175.95,22245.15,23946.70,81509.44,-141,339,-159,16392,-476,3760,M
67945.95,24346.15,32507.74,32803.74,60220.59,342,-818,-69,16220,-2040,3272,S
79722.22,25275.59,32070.55,29868.42,36666.67,-2389,-1933,1151,16140,-2148,6528,A
64921.88,23555.27,29221.90,19024.39,62893.40,-1528,-910,194,16376,-236,3732,T
70250.00,24651.16,32215.38,32361.11,65763.16,651,-589,-43,16300,-1748,2188,S
60591.13,17510.59,22982.67,23555.27,79722.22,-370,-1843,-330,16268,84,6392,M
72685.71,24651.16,31359.52,27156.59,37551.38,572,-388,-56,15764,-3152,6124,A
52302.63,22426.83,28593.75,17929.18,54121.62,-47,-587,250,16324,-1268,3760,T
58774.04,18214.29,22426.83,23076.92,75798.82,357,255,-91,16248,-120,4868,M
71694.91,23652.39,31219.88,32070.55,66622.34,158,-448,-336,16144,-1248,4304,S
71207.87,23652.39,30000.00,31359.52,61343.28,-2908,117,105,15888,-1424,6452,S
72685.71,23076.92,29478.26,26250.00,44941.41,111,-795,-38,15712,-1272,6316,A
62500.00,23076.92,27041.10,17788.46,59490.29,141,-409,-536,16236,-112,3756,T
55714.29,19100.00,22335.77,22982.67,77439.76,-660,-280,-11,16404,-124,3960,M
79141.10,26250.00,33560.13,30399.41,39222.61,887,-1369,-40,15752,-3808,2744,A
65763.16,23362.50,29478.26,18874.17,60220.59,-937,-652,909,16028,-552,2552,T
63290.82,24753.88,31500.00,34025.56,68396.74,-2341,-590,187,15768,-3496,2420,S
```

```
59853.66,19717.19,20852.80,21276.60,71207.87,408,-795,324,15864,-2856,4792,M
67945.95,25811.17,32953.12,28842.86,41003.65,-1131,-724,316,15748,-3428,4720,A
61343.28,23652.39,28593.75,17858.67,57725.12,-547,-719,283,16396,-656,2436,T
77439.76,24651.16,32803.74,33406.94,69313.19,956,-1185,-422,16456,-1044,2804,S
52302.63,18874.17,21887.02,22795.56,77439.76,517,-1097,757,16112,-700,3952,M
84000.00,26585.37,33560.13,29868.42,37913.79,382,-619,-110,16112,-2896,4088,A
62110.55,22795.56,30399.41,18799.56,55714.29,-243,1065,-193,16020,-2996,3988,T
71694.91,24857.14,31219.88,35147.06,63290.82,687,-1523,-422,15912,-3612,2820,S
62500.00,18874.17,22518.34,22982.67,77439.76,112,-223,450,16652,-1648,3280,M
73189.65,25488.13,32361.11,28103.93,40197.84,617,-570,-298,15816,-3540,4780,A
71207.87,23362.50,29478.26,18358.69,57382.08,-143,-84,123,16268,-760,4660,T
69313.19,24045.80,31081.08,33406.94,52898.23,228,-358,126,15692,-4608,2144,S
54121.62,18358.69,21887.02,22335.77,77439.76,-143,-216,97,16284,-1872,4440,M
77439.76,24960.94,31219.88,28593.75,40597.83,-1034,-939,313,15364,-4048,5964,A
58071.43,23458.65,29737.61,17648.94,60591.13,416,-1970,961,16416,-1304,3136,T
61343.28,18651.32,22245.15,23171.64,81509.44,-374,-647,369,15996,-2796,3124,M
67945.95,24346.15,31359.52,32070.55,64507.77,900,-1097,-563,15932,-3812,3780,S
71207.87,23362.50,29221.90,27156.59,40597.83,-608,-808,-70,15616,-3944,4940,A
51717.39,22335.77,27863.13,17102.51,55714.29,21,-226,-350,16072,-3260,3168,T
55714.29,17929.18,23076.92,23266.83,74215.13,383,1168,-928,16092,-1732,3976,M
67945.95,24753.88,31500.00,33714.29,61725.00,2034,-1205,114,16172,-2792,3184,S
75798.82,23750.00,31219.88,27983.19,47378.05,-112,-2039,347,15568,-4624,3836,A
64507.77,22888.89,27743.73,17648.94,57725.12,3159,-466,-507,16016,-2260,4384,T
62110.55,19638.82,23458.65,23750.00,77439.76,5000,-666,-106,16332,-1524,2404,M
69313.19,24651.16,31359.52,32361.11,60220.59,1130,-875,39,15972,-3360,3520,S
74736.84,23750.00,30399.41,28842.86,37193.88,-156,-1402,627,15404,-3528,6432,A
62500.00,22518.34,28103.93,17169.81,62500.00,-630,-63,-99,16084,-1684,4452,T
58071.43,18286.34,21887.02,23266.83,74215.13,1027,-753,-17,16040,-2196,5368,M
67945.95,23171.64,33714.29,32803.74,64507.77,142,-417,-203,16324,-1972,3928,S
81509.44,27389.50,33254.72,30265.49,39608.54,624,-2830,504,15056,-5052,1632,A
67945.95,24651.16,29478.26,16125.76,61343.28,-901,101,29,16036,-1616,2252,T
71207.87,24045.80,30805.97,33560.13,69313.19,1605,-879,-183,15696,-5104,1820,S
55714.29,17929.18,20686.05,20603.25,74736.84,1186,-326,39,15684,-3332,1660,M
81509.44,25381.58,33103.45,29478.26,45652.17,45,-528,675,16084,-3904,3272,A
67945.95,24346.15,29349.71,16836.10,53811.66,-1922,478,81,16316,-1300,2920,T
75798.82,23652.39,33103.45,34983.71,63290.82,896,-1089,-89,16080,-4336,2000,S
54434.39,19405.83,21975.90,21798.56,74215.13,-335,-789,132,16812,-808,4428,M
68396.74,24753.88,31359.52,27625.00,38466.90,515,-122,70,16076,-3280,3948,A
57042.25,22795.56,29221.90,16770.19,53504.46,-1225,-771,-277,16292,-1444,4828,T
67945.95,23750.00,31219.88,33254.72,65763.16,621,-628,389,16264,-2480,2960,S
55714.29,19954.44,22065.22,22154.96,75798.82,4743,-1797,372,16432,-1448,4260,M
73699.42,25275.59,32361.11,28347.46,36666.67,-756,-143,-73,15836,-3640,3300,A
52302.63,22518.34,29094.83,18214.29,65763.16,-969,-1332,-130,16400,-812,3160,T
55068.49,18358.69,21975.90,23848.10,73699.42,-234,188,-161,16268,-1680,4396,M
72685.71,24960.94,31081.08,33254.72,58774.04,305,-481,-15,15856,-3932,3144,S
76339.29,24346.15,31359.52,28717.95,42905.66,1115,-1993,111,16088,-2516,3660,A
60220.59,22982.67,27863.13,16770.19,55714.29,153,-33,-596,16264,-1920,2856,T
58071.43,18000.00,21622.91,22702.70,79722.22,489,-720,457,15880,-3488,4692,M
64507.77,24245.52,32070.55,32361.11,56705.61,155,-613,-69,16016,-3576,3068,S
71694.91,23458.65,29094.83,26926.23,44476.74,-91,446,-302,15704,-4072,4536,A
61343.28,21887.02,26472.97,17035.49,56041.67,40,-70,-315,16096,-3104,3528,T
53504.46,17648.94,23076.92,23076.92,74215.13,-1507,525,208,16040,-2228,3952,M
69313.19,24045.80,32215.38,32070.55,56705.61,552,-137,-127,15968,-3556,3756,S
67945.95,23171.64,30265.49,27041.10,43793.11,-296,-222,-125,15764,-4252,4324,A
65763.16,22335.77,27041.10,16574.07,62110.55,-3499,619,161,16380,-1120,4800,T
52898.23,18286.34,21535.71,24245.52,82120.26,-140,203,-184,16088,-1672,4700,M
73699.42,25065.28,33254.72,33714.29,60591.13,-349,-472,-546,15288,-5456,4664,S
69779.01,23171.64,30265.49,27506.93,45176.47,-725,-1082,-264,15920,-4584,3844,A
64507.77,22982.67,28347.46,17579.62,55714.29,41,216,-641,16168,-2308,1676,T
56705.61,18358.69,22795.56,22795.56,79141.10,-302,348,-390,16340,-1932,4120,M
63692.31,23652.39,31219.88,31926.61,55714.29,-428,-312,-166,16176,-2340,3424,S
63692.31,23266.83,30132.36,27389.50,41003.65,101,-76,-88,15616,-4816,3864,A
55714.29,21887.02,26585.37,16189.02,55714.29,2386,-3949,-280,16204,-1788,3756,T
55714.29,19175.95,22065.22,22335.77,79722.22,258,-80,-278,16324,-1596,4124,M
64921.88,23555.27,33714.29,32803.74,59130.43,391,-669,130,16252,-2896,2680,S
```

```
77439.76,27863.13,33714.29,30805.97,44019.23,-950,-929,974,15872,-3828,2220,A
71694.91,24651.16,28593.75,17648.94,48937.50,-1887,140,833,16108,-3552,1968,T
54121.62,19175.95,23750.00,25170.16,78000.00,-247,-70,-129,16216,-2444,2752,M
73699.42,25381.58,33103.45,34500.00,63692.31,-1362,-334,417,16328,-2472,2796,S
65763.16,24346.15,31926.61,28225.35,43125.00,85,-901,652,15948,-3844,3700,A
69779.01,24346.15,29221.90,18286.34,52302.63,-730,-1396,198,15984,-2128,3992,T
58071.43,19560.81,22702.70,24045.80,81509.44,-738,-584,603,16112,-2104,2436,M
64507.77,24447.30,30943.12,33714.29,55714.29,-91,-209,9,16120,-3204,2204,S
71207.87,24651.16,30534.13,28593.75,43568.71,-705,-475,601,15844,-4024,3776,A
74736.84,24145.41,28593.75,17929.18,50297.87,1240,-706,264,16204,-2648,1728,T
58071.43,19328.86,22245.15,23171.64,84000.00,-101,-1030,344,16296,-2372,2608,M
66622.34,25488.13,32803.74,33560.13,59490.29,-2255,174,-740,16020,-3444,3204,S
73189.65,24651.16,31081.08,28103.93,43568.71,-565,-883,648,15848,-3680,4948,A
61343.28,23266.83,29094.83,17510.59,57725.12,-481,-182,584,16164,-2644,2996,T
54750.00,19328.86,21975.90,23458.65,77439.76,-5994,1558,-336,16112,-2892,4024,M
72685.71,24960.94,32070.55,33714.29,59853.66,738,-1127,-133,16624,-1268,3608,S
65763.16,23848.10,30534.13,27983.19,40597.83,-524,-367,-14,15492,-4896,4344,A
56705.61,23076.92,28717.95,16968.75,60220.59,471,-307,-933,16100,-1808,4036,T
52302.63,18725.27,21448.93,23750.00,77439.76,764,-1197,-208,16192,-1824,4200,M
64507.77,24245.52,32361.11,33254.72,58774.04,-1770,296,652,16244,-2504,2140,S
63290.82,23555.27,30399.41,28225.35,42905.66,423,-243,125,15604,-4240,4548,A
60591.13,23458.65,28103.93,17237.39,52898.23,1079,-910,717,16316,-1988,3596,T
52898.23,18799.56,21105.88,23750.00,77439.76,795,-988,-74,16332,-1380,4476,M
79141.10,25065.28,32361.11,34500.00,61343.28,-640,-1137,-384,16228,-2396,3360,S
68852.46,23946.70,30805.97,27156.59,41833.33,-57,-477,293,15940,-3392,3732,A
61343.28,22888.89,27156.59,16968.75,58071.43,1138,-313,289,16124,-2284,3800,T
50858.37,17102.51,19875.00,23652.39,78567.07,239,-831,-274,16520,-2144,3072,M
73699.42,24346.15,33714.29,32953.12,60220.59,215,-514,-50,16152,-2668,3264,S
78567.07,27272.73,32953.12,30805.97,40197.84,1137,488,579,16036,-5352,4160,A
66622.34,25170.16,30399.41,18431.37,46626.50,289,49,284,16512,-2800,1796,T
59130.43,19717.19,22888.89,22518.34,74215.13,-1797,-313,73,15812,-4268,2912,M
74736.84,26250.00,34340.84,34025.56,63290.82,120,-406,154,15760,-4528,1352,S
65763.16,23946.70,30534.13,27506.93,43568.71,40,-197,-53,15704,-4560,3488,A
61343.28,23652.39,29478.26,18000.00,52302.63,668,-1312,610,16244,-2284,3032,T
56705.61,19175.95,21535.71,21535.71,79722.22,-1707,-401,699,16088,-2604,4676,M
66190.48,24753.88,33103.45,32803.74,50297.87,-660,-992,768,16132,-3600,3036,S
69313.19,25702.92,32361.11,28842.86,46135.45,324,-480,-558,15648,-4972,3320,A
61343.28,23266.83,29221.90,18286.34,59130.43,351,-709,-110,15604,-5164,2472,T
55714.29,19328.86,22426.83,24045.80,73699.42,121,-101,219,16092,-4492,2724,M
69313.19,24651.16,32361.11,33254.72,65340.31,-168,-628,-169,15876,-4344,2624,S
66622.34,23750.00,30943.12,28225.35,46626.50,-1392,-1081,609,15652,-4164,4452,A
65340.31,23848.10,30265.49,17718.55,57042.25,162,-1002,199,15748,-4108,2300,T
52302.63,18214.29,21105.88,22982.67,74215.13,-128,-554,174,15948,-3328,4632,M
69313.19,25488.13,31359.52,33406.94,71207.87,30,-294,-48,15868,-4476,2676,S
62500.00,24245.52,30534.13,27625.00,44019.23,-310,-870,554,15888,-3716,4444,A
59490.29,23652.39,28968.48,17718.55,52302.63,-2062,-49,582,15596,-4384,1616,T
54434.39,18651.32,21448.93,22518.34,75798.82,14,-479,113,15916,-4156,2304,M
74736.84,25065.28,31081.08,33254.72,65340.31,346,-144,-234,15932,-4240,1708,S
```

Glove_final.ino

```
#include <SoftwareSerial.h>

#include <math.h>

#include<Wire.h>


SoftwareSerial BTserial(0, 1); // RX | TX

// Connect the HC-05 TX to Arduino pin 2 RX.

// Connect the HC-05 RX to Arduino pin 3 TX through a voltage divider.


// Flex Sensor

const float VCC = 5.0; // Measured voltage of Ardunio 5V line 4.98 daw

const float R_DIV = 15000.0; //  15k ohms

const float DIV_4 = 1023.0; //?

const float DIV_2 = 512.0; //?


// Multiplexer

const int selectPins[3] = {11, 12, 13}; // S0~2, S1~3, S2~4

const int zOutput = 5;

const int zInput = A0; // Connect common (Z) to A0 (analog input)


// MPU6050

const int MPU6050_addr=0x68;

int16_t AccX,AccY,AccZ,Temp,GyroX,GyroY,GyroZ;


// FSR

const int fsrAnalogPin = A3; // FSR is connected to analog 3
```

```
void setup()
{
    // Multiplexer
    for (int i = 0; i < 3; i++)
    {
      pinMode(selectPins[i], OUTPUT);
      digitalWrite(selectPins[i], HIGH);
    }
    pinMode(zInput, INPUT); // Set up Z as an input
    pinMode(fsrAnalogPin, INPUT); // Set up fsr as input

    Wire.begin();
    Wire.beginTransmission(MPU6050_addr);
    Wire.write(0x6B);
    Wire.write(0);
    Wire.endTransmission(true);
    Serial.begin(9600);
    BTserial.begin(9600);
}

void loop()
{
  // Flex Sensor
  float flex[7];
  // Loop through five pins that are used (Y0 - Y4)
  for (byte pin = 0; pin <= 4; pin++)
  {
    if (pin == 0 && pin == 4)
    {
      selectMuxPin(pin); // Select one at a time
      int inputValue = analogRead(A0); // and read Z
      float FlexVoltage = inputValue * VCC / DIV_2;
      float FlexResistance = R_DIV * (VCC / FlexVoltage - 1.0);
      flex[pin] = FlexResistance;
    }
    else
    {
      selectMuxPin(pin); // Select one at a time
      int inputValue = analogRead(A0); // and read Z
      float FlexVoltage = inputValue * VCC / DIV_4;
      float FlexResistance = R_DIV * (VCC / FlexVoltage - 1.0);
      flex[pin] = FlexResistance;
    }
  }

  // MPU6050
  Wire.beginTransmission(MPU6050_addr);
  Wire.write(0x3B);
  Wire.endTransmission(false);
  Wire.requestFrom(MPU6050_addr,14,true);
  AccX=Wire.read()<<8|Wire.read();
  AccY=Wire.read()<<8|Wire.read();
  AccZ=Wire.read()<<8|Wire.read();
  Temp=Wire.read()<<8|Wire.read();
  GyroX=Wire.read()<<8|Wire.read();
  GyroY=Wire.read()<<8|Wire.read();
  GyroZ=Wire.read()<<8|Wire.read();

  // FSR
  int fsr = analogRead(fsrAnalogPin);

  if(fsr > 100) //if fsr is pressed
  {
    // Send all the data
    Serial.println("#" +
                  String(flex[0]) + "," +
                  String(flex[1]) + "," +
                  String(flex[2]) + "," +
                  String(flex[3]) + "," +
                  String(flex[4]) + "," +
                  String(GyroX) + "," +
                  String(GyroY) + "," +
                  String(GyroZ) + "," +
                  String(AccX) + "," +
                  String(AccY) + "," +
                  String(AccZ) + "~");

    BTserial.println("#" +
                  String(flex[0]) + "," +
                  String(flex[1]) + "," +
                  String(flex[2]) + "," +
                  String(flex[3]) + "," +
                  String(flex[4]) + "," +
                  String(GyroX) + "," +
                  String(GyroY) + "," +
                  String(GyroZ) + "," +
                  String(AccX) + "," +
                  String(AccY) + "," +
                  String(AccZ) + "~");
  }

  delay(1000);
}
// The selectMuxPin function sets the S0, S1, and S2 pins
// accordingly, given a pin from 0-7.
void selectMuxPin(byte pin)
{
  for (int i = 0; i < 3; i++)
  {
    if (pin & (1 << i))
      digitalWrite(selectPins[i], HIGH);
    else
      digitalWrite(selectPins[i], LOW);
  }
}
```