SIDDAGANGA INSTITUTE OF TECHNOLOGY, TUMAKURU-572103

(An Autonomous Institute under Visvesvaraya Technological University, Belagavi)

Project Report on

"Decoding Imagined Speech"

submitted in partial fulfillment of the requirement for the completion of VI semester of

BACHELOR OF ENGINEERING

in

ELECTRONICS & COMMUNICATION ENGINEERING

Submitted by

Aditya Keshav Harikantra	(1SI21EC004)
Harsha M	(1SI21EC037)
Rahul Jain S V	(1SI21EC074)
Rohith Ingaleshwar	(1SI21EC078)

under the guidance of

Dr.Veena Karjigi

Associate Professor

Department of E&CE

SIT, Tumakuru-03

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

2023-24

SIDDAGANGA INSTITUTE OF TECHNOLOGY, TUMAKURU-572103

(An Autonomous Institute under Visvesvaraya Technological University, Belagavi)

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

CERTIFICATE

Certified that the mini project work entitled "Decoding Imagined Speech" is a bonafide work carried out by Aditya Keshav Harikantra (1SI21EC004), Harsha M (1SI21EC037), Rahul Jain S V (1SI21EC074) and Rohith Ingaleshwar (1SI21EC078) in partial fulfillment for the completion of VI Semester of Bachelor of Engineering in Electronics & Communication Engineering from Siddaganga Institute of Technology, an autonomous institute under Visvesvaraya Technological University, Belagavi during the academic year 2023-24. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report deposited in the department library. The Mini project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the Bachelor of Engineering degree.

Dr.Veena Karjigi	Head of the Department
Associate Professor	Dept. of E&CE
Dept. of E&CE	SIT,Tumakuru-03
SIT,Tumakuru-03	

External viva:

Names of the Examiners	**Signature with date**
1.	
2.	

ACKNOWLEDGEMENT

We offer our humble pranams at the lotus feet of **His** Holiness, **Dr. Sree Sree Sivakumara Swamigalu**, Founder President and **His** Holiness, **Sree Sree Siddalinga Swamigalu**, President, Sree Siddaganga Education Society, Sree Siddaganga Math for bestowing upon their blessings.

We deem it as a privilege to thank **Dr. M N Channabasappa**, Director, SIT, Tumakuru, **Dr. Shivakumaraiah**, CEO, SIT, Tumakuru, and **Dr. S V Dinesh**, Principal, SIT, Tumakuru for fostering an excellent academic environment in this institution, which made this endeavor fruitful.

We would like to express our sincere gratitude to **Dr. K V Suresh**, Professor and Head, Department of E&CE, SIT, Tumakuru for his encouragement and valuable suggestions.

We thank our guide **Dr. Veena Karjigi**, Associate Professor, Department of E&CE, SIT, Tumakuru for the valuable guidance, advice and encouragement.

Aditya Keshav Harikantra	(1SI21EC004)
Harsha M	(1SI21EC037)
Rahul Jain S V	(1SI21EC074)
Rohith Ingaleshwar	(1SI21EC078)

Course Outcomes

CO1: To identify a problem through literature survey and knowledge of contemporary engineering technology.

CO2: To consolidate the literature search to identify issues/gaps and formulate the engineering problem

CO3: To prepare project schedule for the identified design methodology and engage in budget analysis, and share responsibility for every member in the team

CO4: To provide sustainable engineering solution considering health, safety, legal, cultural issues and also demonstrate concern for environment

CO5: To identify and apply the mathematical concepts, science concepts, engineering and management concepts necessary to implement the identified engineering problem

CO6: To select the engineering tools/components required to implement the proposed solution for the identified engineering problem

CO7: To analyze, design, and implement optimal design solution, interpret results of experiments and draw valid conclusion

CO8: To demonstrate effective written communication through the project report, the one-page poster presentation, and preparation of the video about the project and the four page IEEE/Springer/ paper format of the work

CO9: To engage in effective oral communication through power point presentation and demonstration of the project work

CO10:To demonstrate compliance to the prescribed standards/ safety norms and abide by the norms of professional ethics

CO11: To perform in the team, contribute to the team and mentor/lead the team

Attainment level: - 1: Slight (low) 2: Moderate (medium) 3: Substantial (high)

POs: PO1: Engineering Knowledge, PO2: Problem analysis, PO3: Design/Development of solutions, PO4: Conduct investigations of complex problems, PO5: Modern tool usage, PO6: Engineer and society, PO7: Environment and sustainability, PO8: Ethics, PO9: Individual and team work, PO10: Communication, PO11: Project management and finance, PO12: Lifelong learning

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO-1												3		3
CO-2		3											3	
CO-3											3			3
CO-4						3	3							3
CO-5	3	3											3	
CO-6					3									3
CO-7			3	3									3	
CO-8										3				3
CO-9										3				3
CO-10								3						3
CO-11									3					3

Abstract

Speech in both overt and covert forms are very natural to human beings. During covert speech, human imagine speaking without any intentional movement of any articulators. Brain cells show different electrical activities depending on people's imagination. Electroencephalogram (EEG) is a noninvasive technique that records electrical activity in the human brain of imagined speech. These signals can be used under a Brain-Computer Interface (BCI) framework, for possible decoding of imagined speech.

This project aims to build an assistive technology that provides a new communication channel for people who are unable to speak due to motor disabilities, such as Locked-in Syndrome in which they are not capable to communicate due to the complete loss of motor control. BCI systems using EEG are still facing several challenges that have to be addressed in order to be applied to solve real life problems.

From an accessible source, the EEG signals corresponding to three words are collected. Subsequently, The signals undergo pre-processing, feature extraction, and a support vector classifier is built using training samples. The trained model is deployed on a Raspberry Pi. When tested using test samples the system is able to decode the imagined word correctly with an accuracy of 35%. The imagined word is displayed using LCD. A robotic arm is also built that performs a gesture of the imagines word.

Keywords: Electroencephalogram , Brain-Computer Interface , Assistive technologies, Raspberry Pi.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Humans rely on communication to interact and behave appropriately in social, family, and work environments. Communication can be verbal, using linguistic signs, or non-verbal, using gestures. However, due to conditions like locked-in syndrome or advanced amyotrophic lateral sclerosis (ALS), some individuals cannot produce speech. Imagined speech or the mental simulation of words without physical articulation, can potentially aid these individuals in communication.

Imagined speech involves the brain processing words without moving the lips or tongue. Brain signals, reflecting various tasks and behaviors, can be analyzed to recognize the word a person intends to speak. This recognition could significantly aid individuals with disabilities in communicating effectively. A brain-computer interface is a technology that enables the brain to communicate with external devices without muscle movements. BCIs can use the acquired signals to monitor brain activity during imagined speech, converting brain impulses into spoken words.

BCIs hold promise not only for medical applications but also for enhancing general human-computer interaction. Decoding imagined speech can greatly improve the quality of life for individuals with disabilities by allowing them to express themselves and engage in social interactions. Recent research focuses on creating intuitive BCI systems that utilize the power of imagination, moving beyond traditional paradigms like event-related potentials and motor imagery, which have limitations in diverse communication setups.

To capture brain activity, BCIs use either invasive or non-invasive methods. Invasive techniques, like electrocorticography (ECoG) and functional magnetic resonance imaging (fMRI), provide high-resolution data but require surgery. Non-invasive methods, such as EEG, are preferred for their practicality, good temporal resolution, and cost-effectiveness. EEG does not require surgery, making it more broadly applicable.

Exploring imagined speech recognition is crucial for addressing communication challenges faced by individuals with severe paralysis. Integrating neuroscience with advanced technology in BCIs offers the potential to redefine human interaction with digital systems.

Ongoing research and innovation are essential to improve the performance and accessibility of these systems, empowering individuals with disabilities and opening new possibilities for human-computer interaction.

1.1 Motivation

1. **Developing a surrogate communication for people suffering from neuromotor disorders.**

   Developing a surrogate communication system for individuals suffering from neuromotor disorders involves creating alternative methods and technologies to facilitate effective communication for those who experience difficulties with traditional speech and motor functions. This can include the use of brain-computer interfaces as assistive devices.

2. **Recognition and intervention for depressed people with suicidal/self-harm thoughts.**

   To recognize and intervene in cases of depression with suicidal or self-harm thoughts involves developing a system that can detect specific brain patterns associated with these mental states and provide immediate actions.

3. **A novel approach for covert communication in defence sector.**

   The duty of the soldiers is to defend the territory whenever there is an attack. Signalling their troops to defend becomes challenging. They can warn the enemy by signaling their troops with gestures. In order to get around this problem, using this method which allows the troops to receive messages without giving the enemy any information.

1.2 Objective of the project

The key objectives of the project are:

1. To develop a classifier to classify three imagined words from EEG data.

2. To deploy the model on Raspberry Pi .

3. To design a robotic arm for gesturing the imagined word.

1.3 Organisation of the report

This report is systematically structured into several chapters to provide a clear and detailed understanding of the project. The introduction chapter covers the motivation, objectives and organisation of the report. The literature survey reviews relevant existing research and technologies. The system overview chapter discusses the dataset, pre-processing steps, implementation on Raspberry Pi, model testing and output visualization. The Design of Robotic Arm chapter details of robotic arm components used servo motors, Arduino Uno and Raspberry Pi 4 Model B, along with their functionalities. The system software chapter covers the software aspects, including the Raspberry Pi, Arduino software, and the implemented algorithms. The results chapter presents the outcomes and the conclusion summarizes the findings and suggests future work directions.

Chapter 2

Literature Survey

Understanding the neural and physiological correlates of facial expressions and speech is a critical area of research in neuroscience and cognitive science. EEG and facial expression analysis are commonly used to study brain activity and its relationship with verbal and non-verbal communication. This body of research has numerous applications, from improving human-computer interaction to developing assistive technologies for individuals with communication impairments. [1]

In this study, right-handed individuals without sensory or neurological abnormalities from the University of Toronto recorded their facial expressions and EEG responses while speaking. Participants (eight men and four women, mean age = 27.4) took part in controlled office sessions, responding to word and phonemic cues. A Microsoft Kinect camera tracked six distinct facial animation units (AUs), while EEG data was simultaneously recorded using 64-channel caps with the 10-20 method, ensuring low impedance and minimal eye movement artifacts. The KARA ONE dataset includes EEG, face, and auditory modalities, capturing features during single-word and imagined/vocalized phonemic prompts. It features EEG recordings from twelve individuals responding to four phonetically similar word pairs ("pat," "pot," "know," "gnaw") and seven phonemic/syllabic cues ("iy," "uw," "piy," "tiy," "diy," "m," "n"). Preprocessing with EEGLAB involves band-pass filtering (1-50 Hz), mean subtraction to remove ocular artifacts, and more. Segmented EEG data are windowed and analyzed for features like mean, median, variance, kurtosis, skewness, max, min, spectral entropy, and energy, creating a high-dimensional feature space. Combining these modalities enables binary classification of phonological categories. Deep-belief network (DBN) and Support vector machine (SVM) classifiers are used, with DBN achieving higher performance, classifying ±/uw/ and C/V with 80%-91% accuracy across tasks. [2]

EEG signal preprocessing typically includes windowing, band-passing, downsampling, and filtering. Here, trials were chopped and downsampled to 512 Hz across 64 channels, followed by high-pass filtering. Independent component analysis (ICA) and channel crosscovariance (CCV) have been used in different studies, while the common average reference (CAR) method improves signal-to-noise ratio by eliminating common data across electrodes. Feature extraction can be done in the time, frequency, and space domains. In the time domain, common features include standard deviation (SD), root mean square (RMS), mean, variance, sum, maximum, and minimum. For the frequency domain, techniques like Mel Frequency Cepstral Coefficients (MFCC), Short-Time Fourier Transform (STFT), Fast Fourier Transform (FFT), Wavelet Transform (WT), Discrete Wavelet Transform (DWT), and Continuous Wavelet Transform (CWT) are widely used. [3]

Fifteen participants imagined 15 words ("YES," "NO," "RIGHT," "LEFT," "THANK YOU," "BACKWARD," "DOWN," "TOILET," "TELEVISION," "WATER," "HELP," "LIGHT," "PAIN," "STOP") in about 15 sessions each, with words presented randomly to avoid memorization. The preprocessed EEG signal spanned frequency bands from 0.5 to 100 Hz: delta (0.5–4 Hz), theta (4–8 Hz), mu (8–12 Hz), alpha (8-13 Hz), beta (13–30 Hz), and gamma (30–100 Hz). [4]

Using EEG data from an EMOTIV Epoc device (sampling rate: 128 Hz), "Yes" and "No" were classified. Filters were applied to remove interference and focus on a 0.16-43 Hz bandwidth. Mel Frequency Cepstral Coefficients (MFCCs) were extracted for k-NN classification, using Euclidean, Cosine, and Correlation distance metrics. Cosine distance achieved the highest accuracy at 63%, with 57.5% for "No" and 51.25% for "Yes," highlighting Wernicke's and Broca's area activation during imagined speech. [5] Fifteen participants (S1-S15), aged 20-30, performed 350 EEG trials, with 70 trials per class for the phrases: "hello," "help me," "stop," "thank you," and "yes." For each phrase, 60 trials were designated for training and 10 for validation. Cross-validation across the entire dataset is an option for validating the training data. The test data includes 10 trials for each phrase. Using the supplied validation set is optional. [6]

The study "Classification of Imagined and Heard Speech Using Amplitude Spectrum and Relative Phase of EEG" explores EEG-based speech recognition for imagined and heard speech as part of brain-computer interface technology. Using the FEIS dataset, EEG data from participants hearing and imagining phonemes were analyzed. The study extracted statistical features, amplitude spectrum, and relative phase from EEG signals, testing these with Support Vector Machine and Neural Network classifiers. The results showed that combining amplitude spectrum and relative phase information improved classification accuracy, highlighting the effectiveness of relative phase in imagined speech recognition. Future research aims to refine these models for better feature capture in short time intervals, enhancing BCI applications in medical and welfare fields. [7]

The study investigates Brain-Computer Interfaces for identifying Spanish vowels through imagined speech, addressing challenges in communication for individuals with motor disabilities affecting speech production. It details methods for EEG signal acquisition and processing using spectral analysis, employing Symbolic Aggregate Approximation (SAX) for feature extraction. Classification tasks are handled by Support Vector Machines, achieving an average recognition rate of 85.29% for distinguishing between the vowels 'A' and 'E'. The research underscores the potential of affordable BCI devices in enhancing communication for disabled individuals. [8]

To improve the performance of the system in the absence of large data, for each trial of EEG recording, 11 labels are predicted for each of the 11 channels under consideration. The final decision for each trial is determined through a majority voting process involving the 11 outputs. [9]

2.1 Summary of Literature Survey

The literature review explores the neural and physiological correlates of facial expressions and speech using EEG and facial expression analysis, focusing on studies involving right-handed individuals who recorded facial expressions and EEG responses with Microsoft Kinect and 64-channel EEG caps. The KARA ONE dataset, capturing EEG, facial and auditory data, underwent preprocessing techniques such as filtering and feature extraction (mean, median, variance, kurtosis, skewness, max, min, spectral entropy, and energy), enabling deep-belief networks (DBNs) and support vector machines (SVMs) to classify phonological categories with up to 91% accuracy. Common preprocessing methods like windowing, band-passing, and feature extraction using MFCC, STFT, FFT, and wavelet transforms were applied. Imagined speech studies showed 63% accuracy for classifying “yes” and “no” using MFCCs and k-NN, while phrase classification trials with cross-validation further improved accuracy. These findings have significant applications in human-computer interaction and assistive technologies, enhancing understanding of the neural basis of speech and facial expressions. For every EEG recording trial, 11 labels are predicted for each of the 11 channels that are taken into consideration in the absence of large data. The final trial decision is derived using a majority voting method that aims to enhance system performance.

Chapter 3

System Overview

Decoding imagined speech from EEG involves the discrimination between a fixed set of imagined words from the EEG captured during covert speech.

The fig 3.1 represents a system integrating an EEG signal, a trained Support Vector Machine model, a Raspberry Pi 4 Model B, an Arduino Uno, a robotic arm, an LCD screen and a speaker.

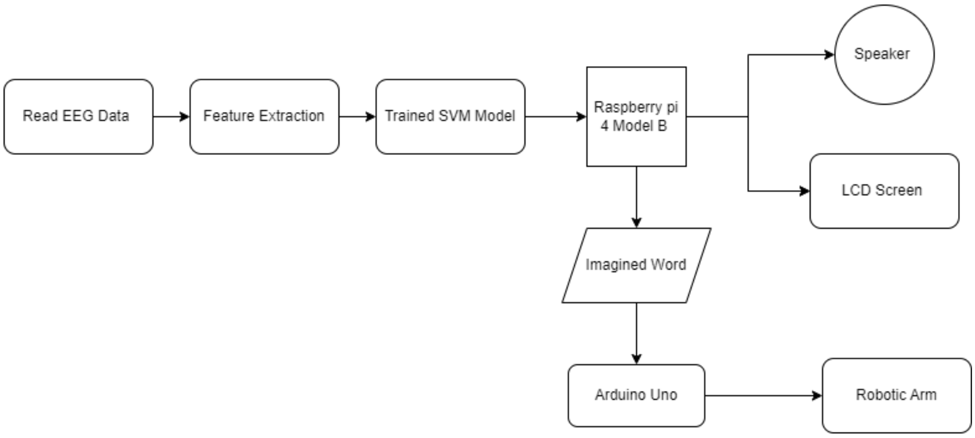


Figure 3.1: Block diagram of system overview

1. EEG Signal: Electroencephalography signals are used to measure electrical activity in the brain. In the system, EEG signals are captured and processed to serve as input data for the trained SVM model. A dataset from a Brain-Computer Interface repository that includes raw EEG data was obtained. The dataset contains recordings of brain activity corresponding to different imagined speech. EEG data is recorded from multiple electrodes placed on the scalp. Out of the 64 available channels, 10 specific channels were selected: FC6, FT8, T7, C5, C3, C4, CP5, CP3, CP1, and P3. These channels are chosen based on their relevance to speech and motor-related brain regions.. This raw data is then processed before being fed into the SVM model.

2. Feature Extraction: FFT is used to convert the time-domain EEG signals into the

frequency domain. This transformation reveals the frequency components present in the signals, which can be more informative for certain types of analysis.
Feature Calculation: Extracting statistical features from the frequency domain signals.

(a) Mean: The average value of the frequency components, indicating the central tendency.

(b) Median: The middle value of the frequency components, providing a robust measure of central tendency.

(c) Skewness: A measure of the asymmetry of the frequency distribution.

(d) Kurtosis: A measure of the 'tailedness' of the frequency distribution.

A feature vector was created by combining the extracted features from all 10 channels. This results in a comprehensive representation of the signal characteristics for each class.
Data Splitting: The dataset was split in 80:20 ratio into training and testing sets to validate the performance of the SVM model. The training set was used to train the model, while the testing set evaluated its performance on unseen data. Model Training: The SVM classifier was trained using the training set. The SVM algorithm found a hyperplane that best separated the classes in the feature space. Model Evaluation: The trained model's performance was evaluated using metrics such as accuracy, precision, recall, and F1-score.

3. LCD Display and Speaker: The SVM model output is interfaced with an LCD to display the predicted word. The system sends the appropriate signal to the LCD and a speaker is used to play pre-recorded audio corresponding to the predicted word. Audio files for each word were stored, and the file matching the prediction is played.

4. Robotic Arm: In a robotic arm, servo motors are used to provide precise control over the arm's movements. Each servo motor controls a different joint or axis of the arm, enabling it to perform complex tasks that require accurate positioning and movement. The shoulder, elbow, and wrist movements are all managed by servo motors, allowing the robotic arm to mimic the natural movements of a human

arm. A program was written for the Arduino to control the servo motors based on the predicted word. The program received the prediction from the SVM model (transmitted via serial communication) and executed the corresponding gesture.

3.1 Dataset Description

A 64-channel Neuroscan Quick-cap was used to collect EEG signals. The EEG data were recorded with an EEG signal amplifier (BrainAmp, BrainProduct GmbH, Germany). The raw data were recorded using BrainVision (BrainProduct GmbH, Germany) with MATLAB 2019a (The MathWorks Inc., USA). 64 EEG electrodes following a 10-20 international configuration were used for the recording as shown in fig 3.2. Ground and reference channels were placed on the Fpz and FCz. The impedance of all electrodes between the sensors and the skin of the scalp were maintained below 15kΩ [10]. The subjects (S1-S15; aged 20-30 years) were asked to perform imagined speech of five different words/phrases that can be utilized for basic conversation: ‘hello’, ‘help me’, ‘stop’, ‘thank you’ and ‘yes’. EEG of five-class imagined speech words/phrases were recorded. 70 trials per class (70×5 = 350 trials) are released for training (60 trials per class) and validation

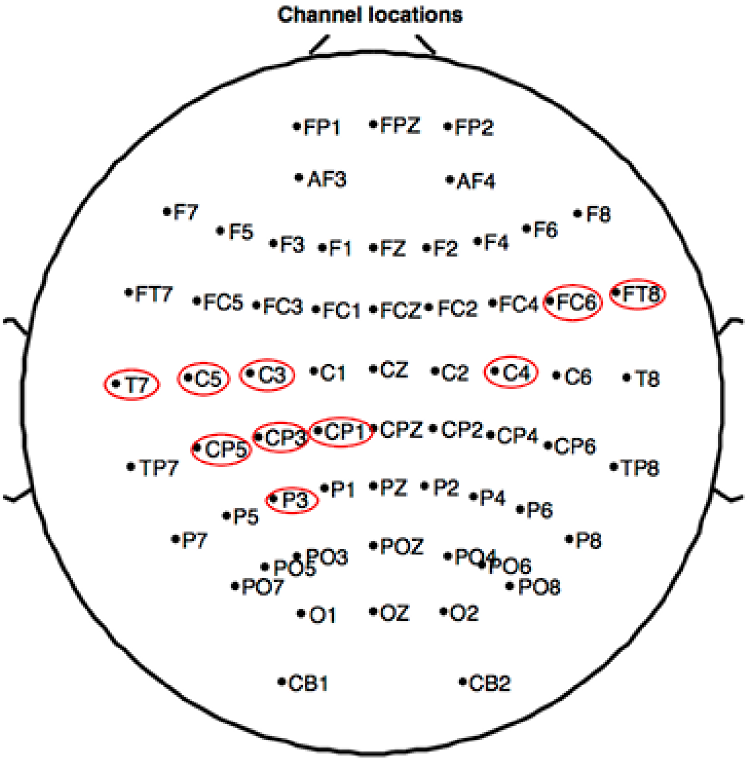


Figure 3.2: Electrode placement locations [10]

(10 trials per class) purpose. Using the given validation set is not obligated. Validation for the training data can be performed not only by the given validation set but also with the competitors' choice (example: N-fold cross validation over the whole data). The test data (10 trials per class) will be released later. The dataset was divided into epochs based on cue information (event codes).

Table 3.1: Classes in Dataset

Class	**Hello**	**Help me**	**Stop**	**Thank you**	**Yes**
code	1	2	3	4	5

The subjects were seated in a comfortable chair in front of a 24-inch LCD monitor screen. The subjects were instructed to imagine the silent pronunciation of the given word as if they were performing real speech, without moving any articulators nor making the sound. Subjects were strictly directed not to perform any other brain activity but the given task. They were noted not to move and avoid dye blinks while imagining or receiving the cue. All imagination trials were performed while a black screen was provided, without receiving any stimulus in order to avoid any other factors affecting the brain activity. An auditory cue of the five words/phrases was randomly presented for 2 s, followed by 0.8-1.2 s of a cross mark. The subjects were instructed to perform imagined speech of the given cue as soon as the cross mark disappears on the screen. Four times of cross mark (0.8-1.2 s) and imagined speech phase (2 s) were followed in a row per random cue as shown in fig 3.3. After performing four times of imagined speech, 3 s of the relaxation phase was given to clear up the mind for the next word/phrase.

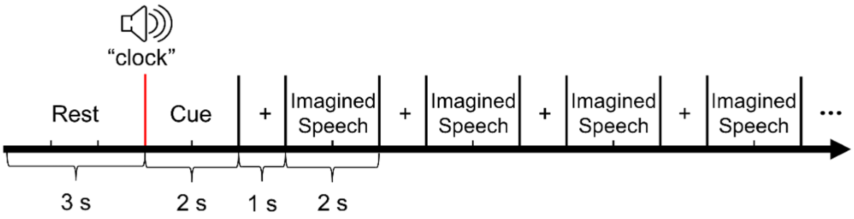


Figure 3.3: Illustration of the Experimental Protocol [10]

3.2 Pre-processing

The collected EEG signals undergo pre-processing to enhance their quality and extract relevant features. It includes taking Fast Fourier Transform subsequent analysis and feature extraction.

3.3 Implementation on Raspberry Pi

The trained machine learning model is implemented on a Raspberry Pi, demonstrating the portability and potential real-world application of the system.

3.4 Testing the model

The system is tested using additional data to evaluate its accuracy and performance.

3.5 Output Visualization

Upon successful testing, the output of the trained model, representing the decoded imagined speech, as illustrated in fig 3.4 output is displayed in the form of image and speech. The output will be displayed with the help of LCD screen and and relayed audibly through a speaker. The robotic arm executes a corresponding gesture of the imagined word.

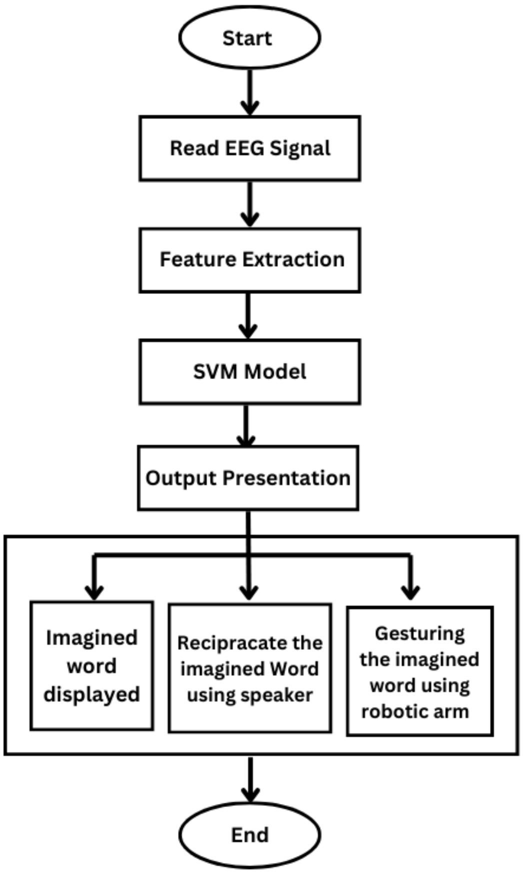


Figure 3.4: Output Visualization Flow Chart

Chapter 4

Design of Robotic Arm

The Arduino-based robotic arm controlled by four servo motors, 3 MG996R servomotors and a SG90 servomotor. The setup uses an Arduino UNO as the central controller interfaced with the servo motors to facilitate various movements of the robotic arm.

4.1 Robotic Arm

The circuit fig 4.1 illustrates an Arduino-based robotic arm controlled by four servo motors with command inputs from a Raspberry Pi. The power supply, connected at P1, distributes voltage across the entire circuit, powering both the Arduino and the servo motors. The Arduino UNO acts as the central controller, receiving commands from the Raspberry Pi, which processes input data or runs pre-programmed sequences. These commands are then translated into pulse-width modulation (PWM) signals sent from the Arduino's digital pins (D8, D9, D10, and D11) to the servo motors signal pins. These PWM signals determine the angle of rotation for each servo, enabling the following movements:

1. The shoulder servo (Servo01, MG996R) is connected to digital pin D8 and controls the arm's up-and-down movement.

2. Two elbow servos (Servo02 and Servo03, MG996R) are connected to digital pins D9 and D10, respectively, enabling the forearm to move up and down and rotate right and left.

3. The wrist servo (Servo04, SG90) is connected to digital pin D11 and allows for precise wrist movements.

Each servo motor is connected to the power supply and the common ground. When powered on, the Raspberry Pi sends commands to the Arduino UNO, which processes these commands and generates corresponding PWM signals to control the servo motors rotation angles. The coordinated actions of these servo motors, driven by the Arduino, allow the robotic arm to perform the natural movements of a human arm.

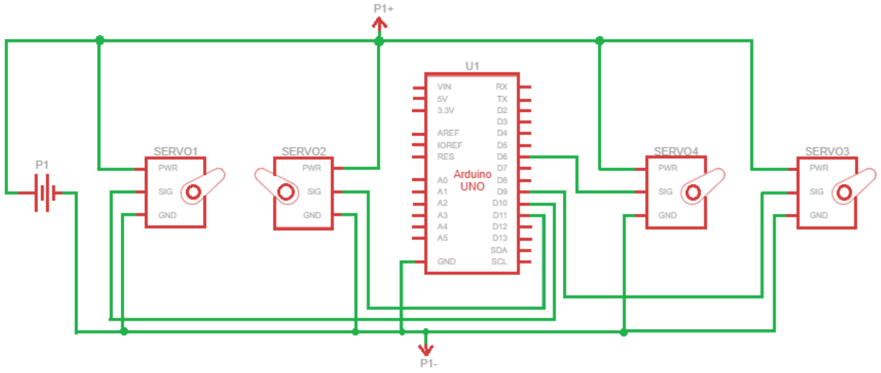


Figure 4.1: Circuit connection of Robotic Arm

4.1.1 Servo-Motors

The Arduino Uno R3 is connected to three MG996R and two SG90 servo motors, creating a setup for precise control of these motors as shown in fig 4.1. In the robotic arm, servo motors are used to provide precise control over the arm's movements. Each servo motor controls a different joint or axis of the arm, enabling it to perform complex tasks that require accurate positioning and movement. The shoulder, elbow, and wrist movements are all managed by servo motors, allowing the robotic arm to mimic human-like movements and perform tasks such as assembly, picking and placing objects, and other precise manipulation tasks.

The SG90 and MG996R are two commonly used servo motors in such applications, each with distinct characteristics. The MG996R is particularly well-suited for a robotic arm due to its high torque and robust metal gears. It can handle heavier loads, which is essential for the shoulder and elbow joints that need to support and move the weight of the arm and any objects it manipulates. The high torque of the MG996R, typically around 10 kg-cm, ensures that the arm can move smoothly and steadily without losing accuracy or control. On the other hand, the SG90, with its plastic gears and lower torque, is better suited for lighter applications. Its typical torque of around 1.8 kg-cm makes it ideal for less demanding joints like the wrist, where precision is more critical than strength. The lightweight and low-cost nature of the SG90 also contribute to its suitability for positions where the load is not as significant.

SG90

The SG90 servo motor, depicted in fig 4.2, is a compact and lightweight motor known for its precise control and wide application in robotics, remote control systems and automation projects. It operates on the principle of pulse width modulation to accurately position its shaft at specific angles within a 180-degree range. This makes it suitable for tasks requiring precise angular control, such as steering mechanisms in robots, camera gimbal stabilization and remote-controlled aircraft.

The SG90 servo motor is typically powered by a DC supply voltage ranging from 4.8V to 6V [Appendix A], making it compatible with standard power sources commonly used in electronics projects. It features a three-wire interface: one for VCC, one for GND and one for control signal. The control signal, usually generated by a microcontroller or PWM signal generator, determines the position of the motor shaft based on the duty cycle of the PWM signal. With its lightweight plastic gears and small size (approximately 23 mm × 12.2 mm × 29 mm), the SG90 servo motor is easy to integrate into compact designs while providing adequate torque for its intended applications. Its affordability, ease of use and availability in hobbyist electronics markets make it a popular choice among makers and enthusiasts for various mechanical control tasks.

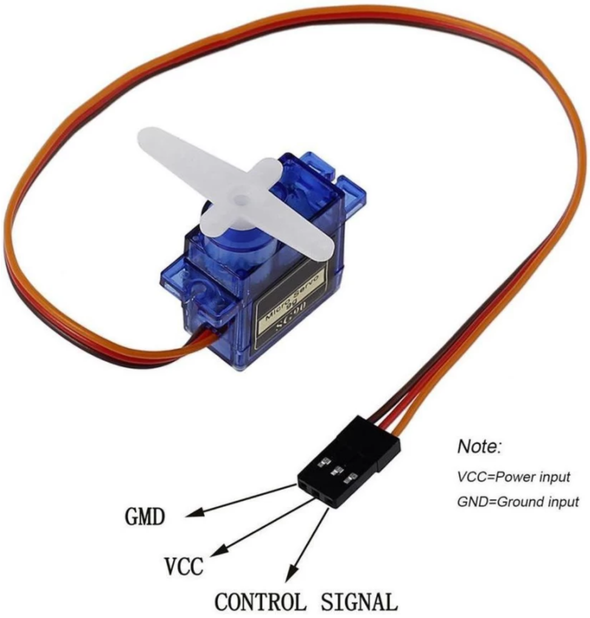


Figure 4.2: Pin details of servo motor SG90 [11]

MG996R

The MG996R servo motor, depicted in fig 4.3, is a robust and powerful motor widely used in robotics, remote control systems and industrial automation due to its high torque and durability. It operates on the principle of pulse width modulation to accurately position its shaft within a 180-degree range, making it suitable for applications requiring precise angular control and strong mechanical force.

The MG996R servo motor is designed to operate with a DC supply voltage typically ranging from 4.8V to 7.2V [Appendix B], making it compatible with various power sources commonly used in electronics and robotics projects. It features metal gears that contribute to its durability and ability to handle higher loads compared to its plastic-gear counterparts like the SG90. The motor's torque output ranges from 10 kg/cm to 15 kg/cm, depending on the operating voltage, enabling it to manipulate larger mechanisms and heavier loads effectively. Similar to other servo motors, the MG996R utilizes a three-wire interface: one wire for VCC, one for GND and one for control signal. The control signal dictates the position of the servo shaft based on the width of the PWM signal it receives from a microcontroller or PWM signal generator.

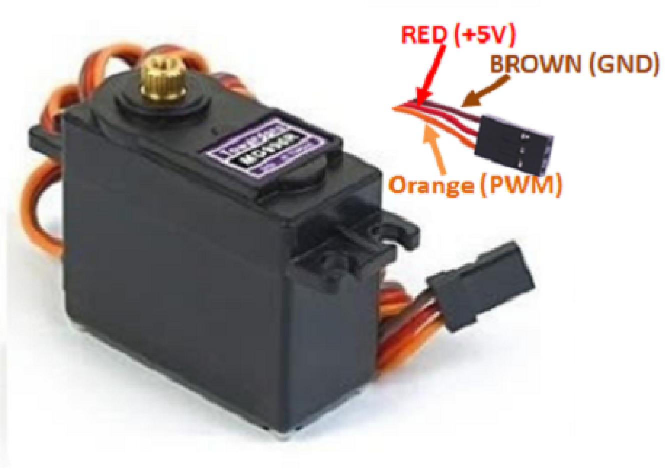


Figure 4.3: Pin details of servo motor MG996R [12]

4.1.2 Arduino-Uno

The Arduino Uno R3, as depicted in fig 4.4, is a widely used microcontroller board known for its simplicity and versatility in electronics projects. Powered by the ATmega328 microcontroller running at 16 MHz, it offers sufficient processing power for a variety of applications, including sensor interfacing, data logging and control systems [Appendix C]. The board features 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a USB connection for programming and power, an ICSP header and a reset button. These components allow users to easily connect and control sensors, actuators and other electronic devices.

The Arduino Uno R3 is favored for its ease of use and extensive community support, making it an excellent choice for both beginners and experienced developers. It operates on a simple open-source platform, supported by a vast ecosystem of libraries and example code, enabling rapid prototyping and develpment of projects in various domains such as robotics, automation and interactive installations. Its compact design and straightforward programming environment via the Arduino IDE make it an ideal tool for learning electronics and exploring the world of embedded systems.

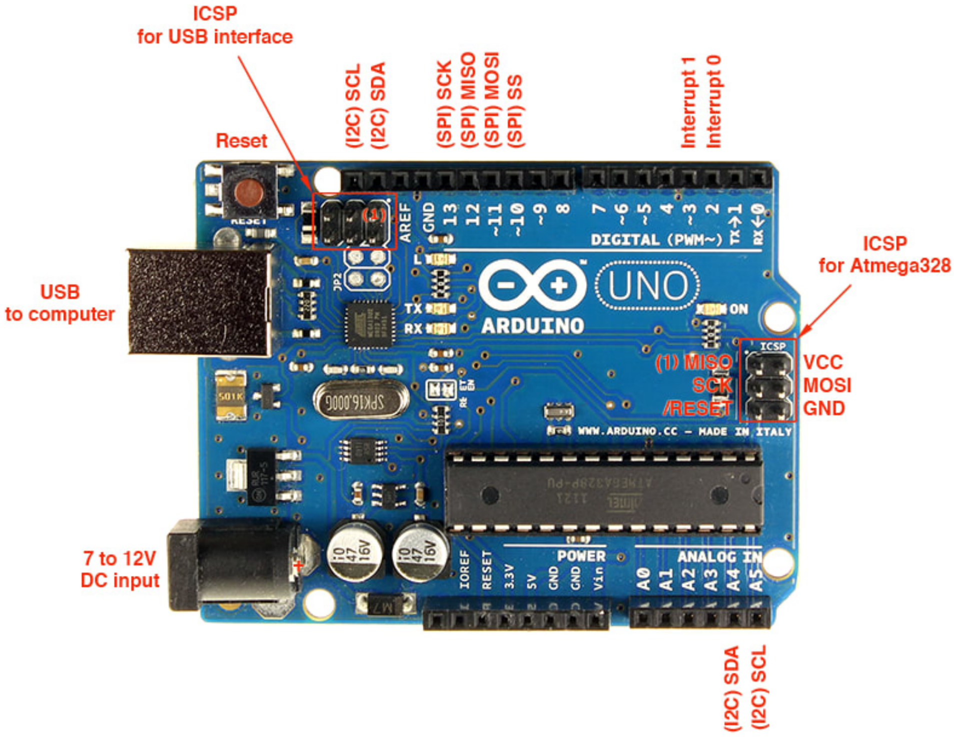


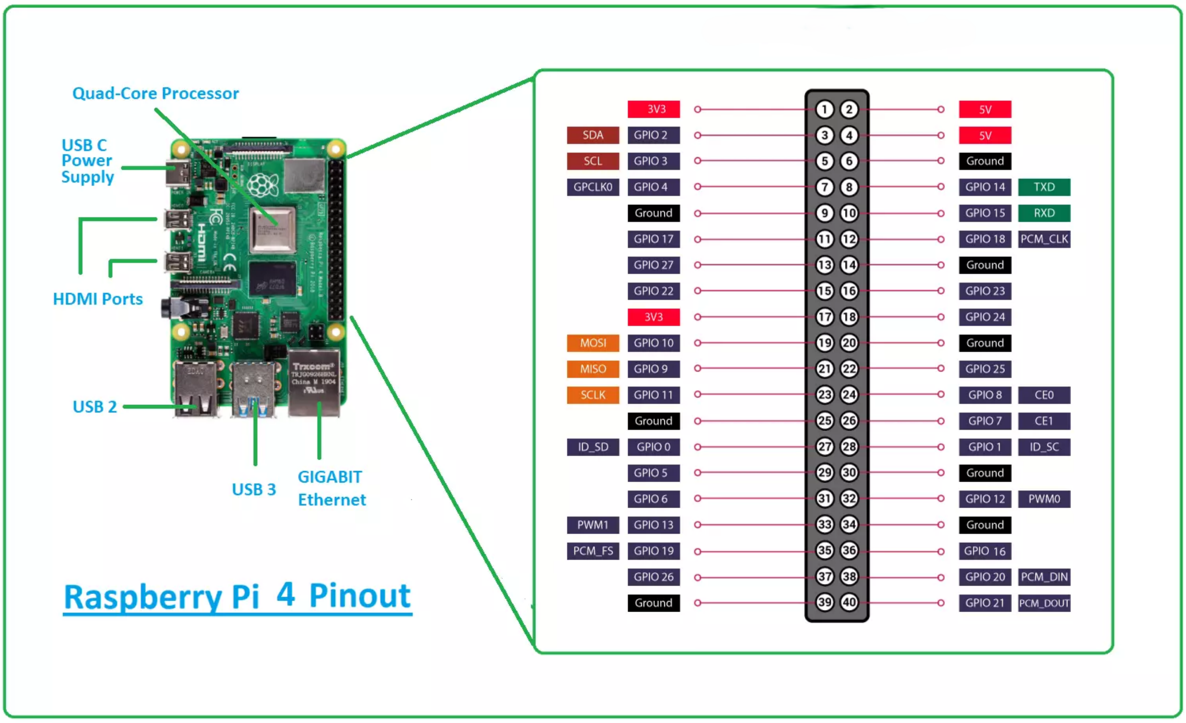
Figure 4.4: Pin details of Arduino-Uno [13]

4.2 Raspberry Pi 4 Model B

The Raspberry Pi 4 Model B, featured in fig 4.5, stands out as a powerful single-board computer renowned for its versatility and robust performance. Key features of the Raspberry Pi 4 Model B include dual-band 2.4/5.0 GHz wireless LAN, Bluetooth 5.0, Gigabit Ethernet and multiple USB ports, ensuring comprehensive connectivity options for networking and peripheral devices [Appendix D]. It supports dual micro-HDMI ports with 4K resolution output, making it an excellent choice for high-definition display projects and digital signage.

The board maintains compatibility with previous Raspberry Pi models through its 40-pin GPIO header, facilitating easy integration with sensors, actuators and expansion boards for customized applications. Its compact form factor and affordability make it ideal for educational purposes, prototyping and IoT deployments.

Fig 4.5 provides a visual representation of the Raspberry Pi 4 Model B's layout and components, emphasizing its role as a versatile platform empowering enthusiasts, educators and professionals to innovate and create with ease in the realm of embedded computing and digital technology.



	3V3	1	2	5V	
SDA	GPIO 2	3	4	5V	
SCL	GPIO 3	5	6	Ground	
GPCLK0	GPIO 4	7	8	GPIO 14	TXD
	Ground	9	10	GPIO 15	RXD
	GPIO 17	11	12	GPIO 18	PCM_CLK
	GPIO 27	13	14	Ground	
	GPIO 22	15	16	GPIO 23	
	3V3	17	18	GPIO 24	
MOSI	GPIO 10	19	20	Ground	
MISO	GPIO 9	21	22	GPIO 25	
SCLK	GPIO 11	23	24	GPIO 8	CE0
	Ground	25	26	GPIO 7	CE1
ID_SD	GPIO 0	27	28	GPIO 1	ID_SC
	GPIO 5	29	30	Ground	
	GPIO 6	31	32	GPIO 12	PWM0
PWM1	GPIO 13	33	34	Ground	
PCM_FS	GPIO 19	35	36	GPIO 16	
	GPIO 26	37	38	GPIO 20	PCM_DIN
	Ground	39	40	GPIO 21	PCM_DOUT

Figure 4.5: Pin details of Raspberry-pi-4 [14]

Chapter 5

System Software

The system software is an integral part of ensuring the seamless interaction between the hardware components and the data processing algorithms. This section includes the details about the Raspberry Pi, Arduino Software, and the algorithms used for processing the EEG data and controlling the robotic arm. The code utilized a variety of software tools and libraries to effectively process and visualize EEG data. Python served as the primary programming language due to its readability and the extensive scientific and data processing libraries. Key libraries used included 'os' for interacting with the operating system, 'matplotlib.pyplot' for creating visualizations, and 'numpy' for handling large multidimensional arrays and performing mathematical operations.

The MNE library was used for working with EEG data, providing tools for reading, preprocessing, analyzing, and visualizing EEG data. The 'scipy.io.loadmat' function was used to read MATLAB files (.mat), which was essential for loading the EEG data. Within the code, the MNE library's functions like 'mne.create_info' and 'mne.io.RawArray' were used to create metadata structures and raw data objects from the preprocessed EEG data. The 'mne.io.read_raw_fif' function read the saved FIF files, a standard format for storing raw EEG data. The visualization of EEG data was handled by 'raw.plot' from the MNE library, which provided interactive plots for inspecting different channels and time points.

5.1 Raspberry Pi

The Raspberry Pi operates on a versatile and user-friendly software ecosystem designed to maximize its hardware capabilities. The primary operating system for Raspberry Pi is Raspberry Pi OS (formerly Raspbian), a Debian-based Linux distribution optimized for the Pi's architecture. Raspberry Pi OS comes pre-installed with a comprehensive suite of development tools, educational software and general-purpose applications, making it an ideal choice for both beginners and advanced users.

Raspberry Pi OS supports a wide range of programming languages such as Python, C, C++, Java and Scratch, enabling users to develop software for various applications, from simple scripts to complex projects. The operating system also includes a graphical user interface (GUI) based on the Lightweight X11 Desktop Environment (LXDE), providing an intuitive desktop experience similar to that of traditional computers. For headless setups, users can opt for the Raspberry Pi OS Lite version, which is a minimal installation without a desktop environment, ideal for server and IoT applications.

The Raspberry Pi ecosystem is further enhanced by an extensive repository of open-source software and community support. Users can easily access software libraries, tutorials and forums, facilitating troubleshooting and project development. This robust software environment, combined with the Raspberry Pi's hardware capabilities, makes it a powerful platform for innovation, education and experimentation in computing and electronics.

5.2 Arduino Software

The Arduino Integrated Development Environment (IDE) is a powerful and user-friendly software platform used for programming and developing Arduino microcontroller boards. Compatible with Windows, macOS and Linux, the Arduino IDE is a cross-platform application that provides an intuitive interface for writing, compiling and uploading code to Arduino boards. It features a simplified programming language based on C++ that is easy to learn, making it accessible for beginners and efficient for advanced users.

Key features of the Arduino IDE include a text editor with syntax highlighting, autocompletion and error highlighting, which enhance the coding experience. Additionally, it comes with a built-in serial monitor, enabling users to send and receive data between the computer and the Arduino board, facilitating real-time debugging and interaction with the hardware. The IDE supports a wide range of Arduino boards, including popular models like the Arduino Uno, Mega, Leonardo and Nano, as well as many third-party boards based on the Arduino platform.

The Arduino IDE's functionality can be extended through plug-ins and libraries, allowing developers to add new features and capabilities to their projects. This flexibility, combined with the IDE's free and open-source nature, encourages continuous improvement and innovation. Users can download the Arduino IDE from the official Arduino website, where it is regularly updated with bug fixes and new features. A large and active community of developers and users supports the IDE, contributing to its development and providing assistance through forums and tutorials. This vibrant ecosystem makes the Arduino IDE an essential tool for anyone working with Arduino microcontroller boards.

5.3 Imagined Speech Recognition

Raspberry Pi

fig 5.1 shows the Flowchart of Imagined Speech Recognition, outlining the key steps involved.

Setup and Initialization

Import necessary libraries. Initialize the serial communication with Arduino. Set up sound and LCD modules. Display messages indicating that the system is waiting for the file to be loaded.

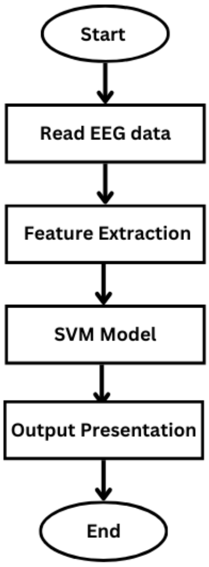


Figure 5.1: Flowchart of Imagined Speech Recognition

Read EEG Data

Change the current directory to where the .fif file is located.Read the EEG data from the '.fif' file using mne.io.read_raw_fif.

Feature Extraction

Remove the mean from the data. Perform FFT on the mean-removed data. Threshold the FFT results to limit magnitudes. Extract statistical features (mean, median, skewness, kurtosis) from the magnitude of FFT results.

Data Preparation

Flatten the extracted features. Scale the flattened features using StandardScaler.

Model Prediction

Load the pre-trained model using 'joblib'. Make predictions using the model.

Output and Actions

Display the predicted word on the LCD. Play the corresponding audio file and send a command to Arduino based on the prediction.

Arduino Code of Robotic Arm

Initialization:

Include Libraries: The 'Servo.h' library is included to facilitate control of the servo motors. Declare Servo Objects: Four servo objects are declared: shoulder, elbow, elbow_pivot and wrist.

Setup Function:

The serial communication is initialized at a baud rate of 9600 using Serial.begin(9600). The servo objects are attached to specific PWM-capable digital pins on the Arduino. Set Initial Servo Positions: The servos are set to their initial positions, shoulder to 0 degrees, elbow to 90 degrees, elbow_pivot to 0 degrees, wrist to 90 degrees. The program prints "Input" to the serial monitor and waits for user input. The program enters a while loop that pauses execution until data is available on the serial input. The user input is read and parsed as an integer using Serial.parseInt().

Switch Case Statement:

Based on the value of the parsed input from Raspberry pi, the program executes different blocks of code to control the servo positions:

Case 1: Sets specific positions for the servos, holds the position for 6 seconds and then breaks out of the switch statement.

Case 2: Sets specific positions for the servos, performs a shaking motion of the elbow servo by moving it between 90 and 105 degrees three times, then holds the position for 1.5 seconds and breaks out of the switch statement.

Case 3: Sets specific positions for the servos, holds the position for 4 seconds and then breaks out of the switch statement.

Default Case: Sets all servos to initial positions.

Chapter 6

Results

This chapter presents the analysis results of EEG signals through both raw data visualization and Fast Fourier Transform (FFT) analysis.

The fig 6.1 represents a plot of raw EEG data visualized using the MNE library in Python. The y-axis represents different EEG channels, each corresponding to a specific electrode placed on the scalp, with labels like FC6, FT8, C3, C4, C5, T7, CP3, CP1, CP5 and P3 indicating these channels. The x-axis represents time in seconds, showing that the EEG data covers a duration of 3 seconds. Each channel displays continuous signals over the 3-second window, with varying amplitude and frequency characteristics. EEG data is inherently noisy and the raw signals often contain artifacts from eye blinks, muscle movements and other external sources. Despite this, distinct oscillatory patterns can be identified, which are crucial for various analyses, such as detecting neural rhythms and identifying potential anomalies.

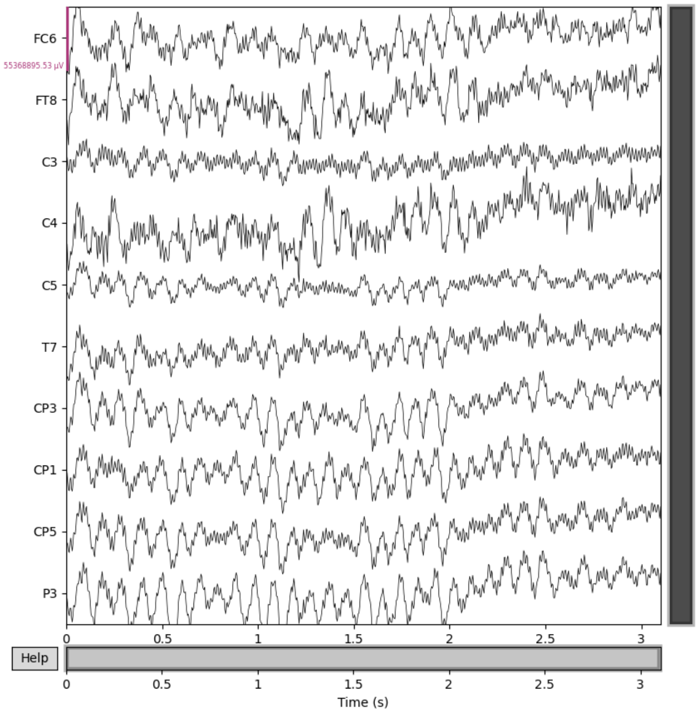


Figure 6.1: Raw EEG data from multiple channels for Class 1 during Trial 167

The fig 6.2 represents magnitude spectrum of the Fast Fourier Transform (FFT) of EEG data from ten channels, visualized using the MNE library in Python for Trial 167, Word 1. Each subplot corresponds to one EEG channel, labeled as Channel 1 through Channel 10. The x-axis represents frequency in Hertz (Hz), ranging from -100 Hz to 100 Hz, while the y-axis represents the magnitude of the FFT, indicating the signal strength at each frequency. Most channels show prominent peaks around the 0 Hz frequency, indicating significant low-frequency components, which are typical of EEG signals due to slow cortical potentials. The FFT magnitude spectra are symmetric about the 0 Hz frequency, which is expected for real-valued signals like EEG data. The magnitude scale varies across channels, with some channels having a maximum magnitude of 1000 and others up to 2000, reflecting variability in the power of frequency components across different EEG channels. Channel 1 shows a complex signal with multiple peaks and significant power across various frequencies. Channels 2, 4, 6, 8 and 10 display prominent peaks around 0 Hz, indicating strong low-frequency activity, with Channel 5 showing sharp peaks at specific frequencies, suggesting strong periodic components. Channels 3, 5, 7 and 9 have more distinct peaks around 0 Hz but also show other frequency components. The presence of smaller peaks and noise

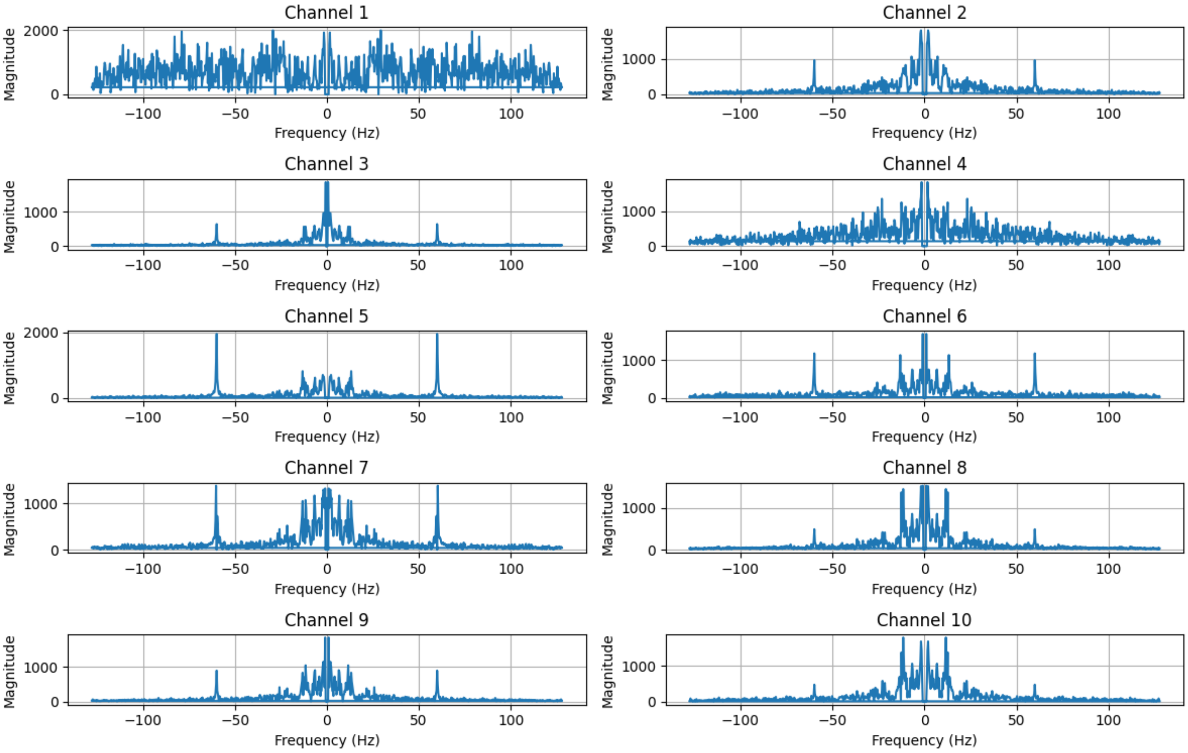


Figure 6.2: Magnitude Spectrum of FFT - Class 1 of Trial 167

components at higher frequencies indicates harmonics and other signal characteristics. This analysis highlights the spatial differences in brain activity captured by different electrodes and provides insights into brain rhythms, anomalies and overall brain function during the trial.

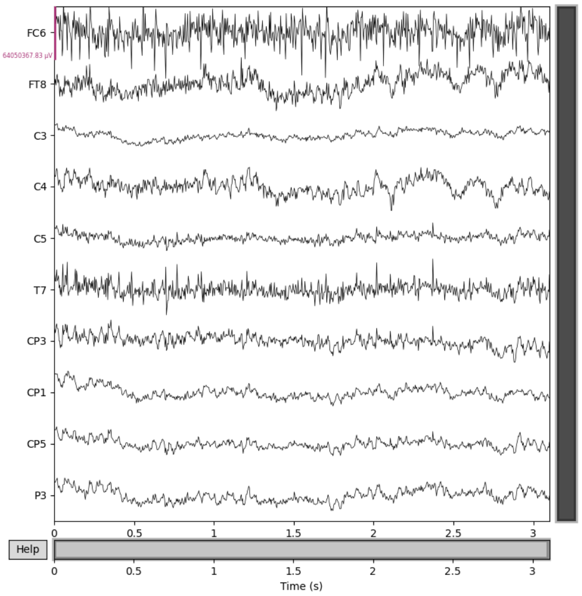


Figure 6.3: Plot of raw EEG data for Class 3 of Trial 102

The fig 6.3 represents Plot of raw EEG data for Class 3 of Trial 102 in which FC6 channel shows high amplitude and frequent oscillations, indicating significant brain activity or potential artifacts. Other channels display lower amplitude signals with more defined rhythms and patterns typical of EEG data. The high amplitude and noise in the FC6 channel could be due to artifacts such as eye blinks, muscle movements, or poor electrode contact. This plot provides a visualization of raw EEG data, highlighting the need for further processing and analysis to extract meaningful information about neural activity.

The FC6 channel shows high amplitude and frequent oscillations, indicating significant brain activity or potential artifacts. Other channels display lower amplitude signals with more defined rhythms and patterns typical of EEG data. The high amplitude and noise in the FC6 channel could be due to artifacts such as eye blinks, muscle movements, or poor electrode contact. This plot provides a visualization of raw EEG data, highlighting the need for further processing and analysis to extract meaningful information about neural activity.

Channels exhibit less prominent high-frequency components, typically above 50 Hz, which

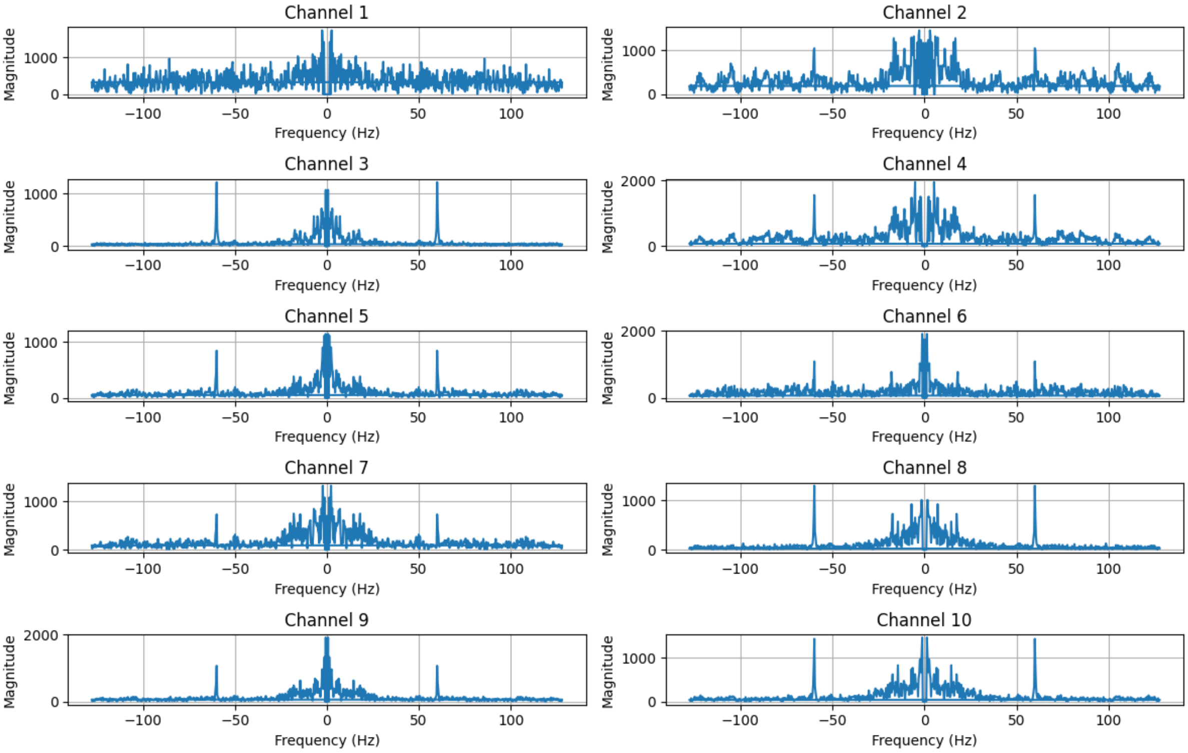


Figure 6.4: Magnitude Spectrum of FFT - Class 3 of Trial 102

could be associated with muscle artifacts or noise. Each channel displays unique frequency distribution patterns, with distinct peaks in certain channels suggesting specific neural or artifact-driven activities. This analysis of the FFT magnitude spectrum reveals the frequency characteristics of the EEG signals, highlighting the predominance of low-frequency components and the variability of spectral power across different channels as show in fig 6.4.

The signals exhibit varying amplitude and frequency characteristics across different channels. For example, the FC6 and FT8 channels display higher amplitude oscillations, indicating significant neural activity or potential artifacts. Other channels, such as C3 and CP1, show more regular and lower amplitude waveforms, typical of EEG signals. This visualization provides an overview of the brain's electrical activity across different regions, with distinct patterns observable in each channel. Further processing, such as filtering and artifact removal, would be necessary to extract meaningful neural information from this raw EEG data as shown in fig 6.5.

Most channels exhibit prominent peaks around 0 Hz, indicating dominant low-frequency components. These peaks suggest strong steady or slow-varying brain activity, which is

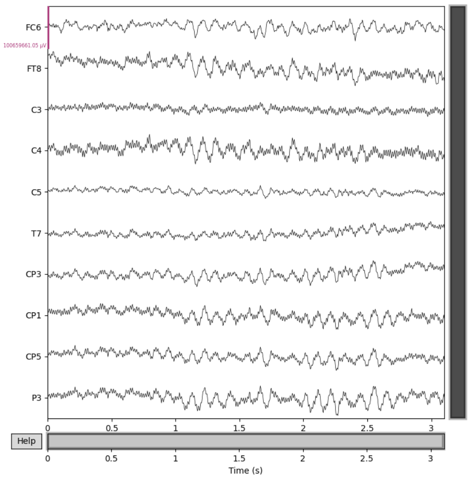


Figure 6.5: Plot of raw EEG data for Class 4 of Trial 52

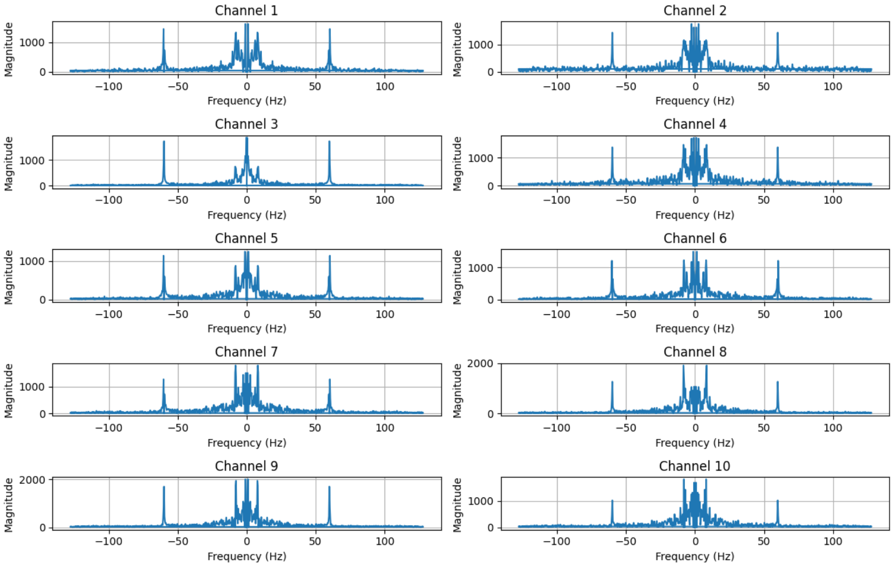


Figure 6.6: Magnitude Spectrum of FFT - Class 4 of Trial 52

common in EEG signals. The variations across channels reflect the spatial differences in brain activity captured by different electrodes. Additionally, the presence of smaller peaks and noise components at higher frequencies indicates harmonics and other signal components. This FFT analysis of EEG signals helps in identifying brain rhythms, detecting anomalies and understanding brain function and its dynamic changes during specific tasks or stimuli as shown fig 6.5.

The Support Vector Machine model for predicting the imagined word was trained using a polynomial kernel with a regularization parameter C=1 and a gamma value of 0.1. A total of 2700 trials were conducted, comprising 2160 trials for training and 540 trials for testing. This experimental design aimed to evaluate the model's performance under varied conditions, ensuring robustness and generalization capability. The parameters selected for the SVM model were chosen to optimize predictive accuracy while maintaining computational efficiency. The division of trials into training and testing sets allowed for a comprehensive assessment of the model's ability to classify data points effectively across different datasets.

The table 6.1 displays the performance metrics of the classification model, evaluated using LibSVM. The model's overall accuracy is approximately 36.67%, indicating that it correctly predicts the class of an instance about one-third of the time. For class 1 (Hello), the precision is 38%, meaning that out of all instances predicted as class 1, 38% were correct. The recall for class 1 is 33%, indicating that the model correctly identifies 33% of all actual class 1 instances, leading to an F1-score of 0.35. Class 1 has 189 actual instances. For class 3 (Stop), the precision is 35%, the recall is 37% and the F1-score is 0.36, with 175 actual instances. Class 4 (Thank You) shows a precision of 37%, recall of 41% and an F1-score of 0.39, with 176 actual instances.

Table 6.1: Performance Metrics

class	precision	recall	f1-score	support
1	0.38	0.33	0.35	189
3	0.35	0.37	0.36	175
4	0.37	0.41	0.39	176
Accuracy	-	-	0.37	540
Macro avg	0.37	0.37	0.37	540
Weighted avg	0.37	0.37	0.37	540

Overall, the accuracy remains at 0.37, consistent with the header. The macro average, which treats all classes equally, shows a precision, recall and F1-score of 0.37 each. The weighted average, which takes the number of instances in each class into account, also shows a precision, recall and F1-score of 0.37 each, reflecting the model's balanced but low performance across different classes. The total number of instances is 540. In summary, the model's performance is fairly consistent across different classes but is generally low, suggesting a need for further improvement or tuning.

The confusion matrix provides a detailed visual representation of a classification model's performance, showing how actual instances of each class were classified by the model. As shown in table 6.2 the result matrix is a 3x3 grid, with rows representing the actual labels ("Hello," "Stop," and "Thank You") and columns representing the predicted labels. The diagonal cells indicate correct classifications: 62 instances of "Hello," 64 of "Stop," and 72 of "Thank You" were correctly identified by the model. However, the off-diagonal cells reveal misclassifications: many "Hello" instances were misclassified as "Stop" (60) and "Thank You" (67), while "Stop" instances were often predicted as "Hello" (53) and "Thank You" (58). Similarly, "Thank You" instances were misclassified as "Hello" (47) and "Stop" (57). This pattern suggests that the model struggles particularly with distinguishing between "Hello" and "Thank You", leading to a notable number of errors. Overall, while the model shows some ability to correctly classify instances, there is a significant level of misclassification, highlighting areas where the model's accuracy could be improved through further refinement or additional data.

	Predicted Hello	**Predicted Stop**	**Predicted Thank You**
Actual Hello	62	60	67
Actual Stop	53	64	58
Actual Thank You	47	57	72

Table 6.2: Confusion Matrix for three class classification

The table 6.3 outlines the specific angles for various joints of a robotic arm to perform different gestures: Default, Hello, Thank You, and Stop. In the Default position, the shoulder angle is 0°, the elbow is bent at 90°, the elbow pivot is 0°, and the wrist is at 90°, representing a neutral, resting state. For the Hello gesture, the shoulder angle is raised to 15°, the elbow is extended to 160°, the elbow pivot remains at 0°, and the wrist stays at 90°, mimicking a waving motion. The Thank You gesture involves raising the shoulder to 45°, adjusting the elbow angle between 90° and 105°, setting the elbow pivot to 90°, and maintaining the wrist at 90°, indicating a gesture of gratitude. The Stop gesture requires the shoulder to be fully raised to 90°, the elbow fully extended to 0°, with both the elbow pivot and wrist at 0°, signaling to halt. These gestures are controlled by specific codes sent from a Raspberry Pi to an Arduino Uno, which interprets the codes to adjust the angles of the shoulder, elbow, elbow pivot, and wrist joints accordingly. This precise and automated control system enables the robotic arm to perform complex gestures accurately, enhancing its functionality in human-robot interactions, demonstrations, and other gesture-based communication applications.

Gesture	**Shoulder Angle**	**Elbow Angle**	**Elbow Pivot**	**Wrist Angle**
Default	0°	90°	0°	90°
Hello	15°	160°	0°	90°
Thank You	45°	90° to 105°	90°	90°
Stop	90°	0°	0°	0°

Table 6.3: Servomotor angles for each gesture

Chapter 7

Conclusion

The analysis compared imagined speech using EEG signals obtained from 15 different participants who were imagining specific words. Time-domain analysis revealed that while there are similarities, imagined speech signals exhibit distinct patterns. This suggests that imagined speech may not precisely replicate the neural patterns of actual speech production, necessitating further research to understand these differences and their implications. Focusing on imagined speech, frequency-domain analysis, specifically using Fast Fourier Transform, provided significant insights into the spectral components of imagined speech signals. Certain EEG channels were identified as crucial for extracting relevant features related to imagined speech. The consistent FFT patterns across these channels suggest common neural processes underlying imagined speech production, which could aid in developing effective classification algorithms for distinguishing between different imagined words.

However, in three-class classification tasks, the classification accuracy was 37%. The analysis of imagined speech signals provides valuable insights into the neural processes underlying speech production. While imagined speech differs from actual speech, frequency-domain analysis allows for the extraction of discriminative features for classification tasks.

7.1 Future Work

Implementing EEG cap to the developed system for data acquisition is envisioned to significantly improve real-time capabilities of decoding imagined speech. The aim of implementing improved pre-processing approaches and advanced feature extraction techniques to further refine improve the accuracy of SVM classifier. Furthermore, the goal is to make these discoveries more effective and significant by broadening the applications of BCI technologies, especially in the areas of assistive technologies.

Bibliography

[1] C. Herff, and T. Schultz. “Automatic speech recognition from neural signals: a focused review”, Frontiers in Human Neuroscience, vol. 10, Art no. 429, 2016.

[2] S. Zhao and F. Rudzicz. “Classifying phonological categories in imagined and articulated speech”. Proc. International Conference on Acoustics, Speech and Signal Processing, Brisbane Australia, 2015.

[3] D. Lopez-Bernal, D. Balderas, P. Ponce and A. Molina. “A State-of-the-Art Review of EEG-Based Imagined Speech Decoding”. Frontiers in Human Neuroscience, vol. 16, Art no. 867281, 2022.

[4] A. Kamble, P. Ghare, V. Kumar, A. Kothari and A. G. Keskar. “Spectral Analysis of EEG Signals for Automatic Imagined Speech Recognition”. Transactions on Instrumentation and Measurement, vol. 72, Art no. 4009409, 2023.

[5] N. Hashim, A. Ali and W. Noorshahida. “Word-Based Classification of Imagined Speech Using EEG”. Proc. Computational Science and Technology, vol 488, Springer, Singapore, 2018.

[6] D. Pawar and S. Dhage, “Imagined Speech Classification using EEG based Brain-Computer Interface”. Proc. 11th International Conference on Communication Systems and Network Technologies (CSNT), pp. 662-666, Indore, India, 2022.

[7] R. Sakai, A. Kai and S. Nakagawa, “Classification of Imagined and Heard Speech Using Amplitude Spectrum and Relative Phase of EEG”. Proc. 3rd Global Conference on Life Sciences and Technologies (LifeTech), pp. 373-375, Nara, Japan, 2021.

[8] D. Rojas, O. Ramos, and J. Saby. “Recognition of Spanish Vowels through Imagined Speech by using Spectral Analysis and SVM.” Journal of Information Hiding and Multimedia Signal Processing, vol. 7, pp. 889-897, 2016.

[9] J. Panachakel, A. Ramakrishnan and T. Ananthapadmanabha, “Decoding Imagined Speech using Wavelet Features and Deep Neural Networks,” 16th India Council International Conference (INDICON), Rajkot, India, pp. 1-4, 2019.

[10] https://osf.io/pq7vb/ 2020 International BCI Competition, https://osf.io/pq7vb/ , Accessed on 11-07-2024.

[11] https://components101.com/motors/servo-motor-basics-pinout-datasheet
SG-90Servo motor .
Available:https://components101.com/motors/servo-motor-basics-pinout-datasheet, September 2017.

[12] https://components101.com/motors/mg996r-servo-motor-datasheet
MG996R Servo Motor
Available:https://components101.com/motors/mg996r-servo-motor-datasheet, April 2019.

[13] https://forum.arduino.cc/t/can-i-reassign-spi-pins-on-an-arduino-uno/510012
Arduino UNO R3.
Available:https://www.pinterest.com/pin/612278511861545941/megha, February 2018.

[14] https://www.theengineeringprojects.com/2021/03/what-is-raspberry-pi-4-pinout-specs-projects-datasheet.html
Raspberry pi Model B
Available:https://www.pinterest.com/pin/612278511861545941/megha, March 2021.

Appendices

Appendix A

Data Sheet of Servo-Motor SG90

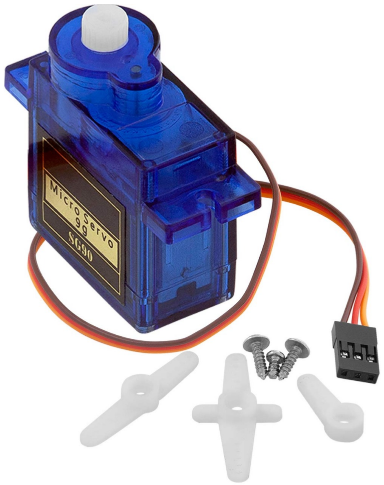


Figure A.1: Servo motor SG90

- Operating Voltage: is +5V
- Torque: 2.5kg/cm
- Operating speed: 0.1s/60°
- Gear Type: Plastic
- Rotation : 0°-180°
- Weight of motor : 9gm
- Pulse Width Range: 500 µs to 2400 µs
- PWM Frequency: 50 Hz
- Dimensions: 22.8 x 12.6 x 22.5 mm
- Connector Type: 3-wire JR connector

Appendix B

Data Sheet of Servo-Motor MG996

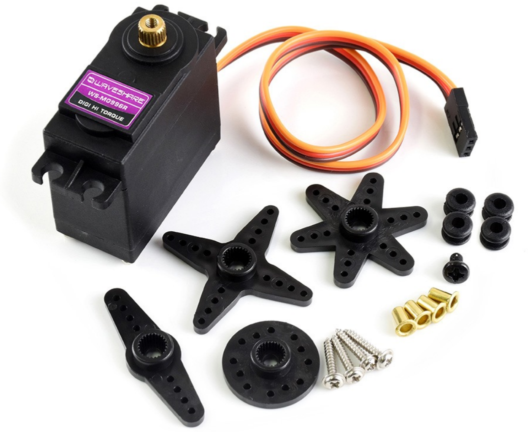


Figure B.1: Servo motor MG996R

- Operating Voltage: is +5V
- Torque: 11 kg/cm
- Operating speed: 0.15s/60°
- Gear Type: Metal gears
- Rotation : 0°-180°
- Weight of motor : 55gm
- Pulse Width Range: 500 µs to 2500 µs
- PWM Frequency: 50 Hz
- Dimensions: 40.7 x 19.7 x 42.9 mm
- Connector Type: 3-wire JR connector

Appendix C

Data Sheet of Arduino Uno R3

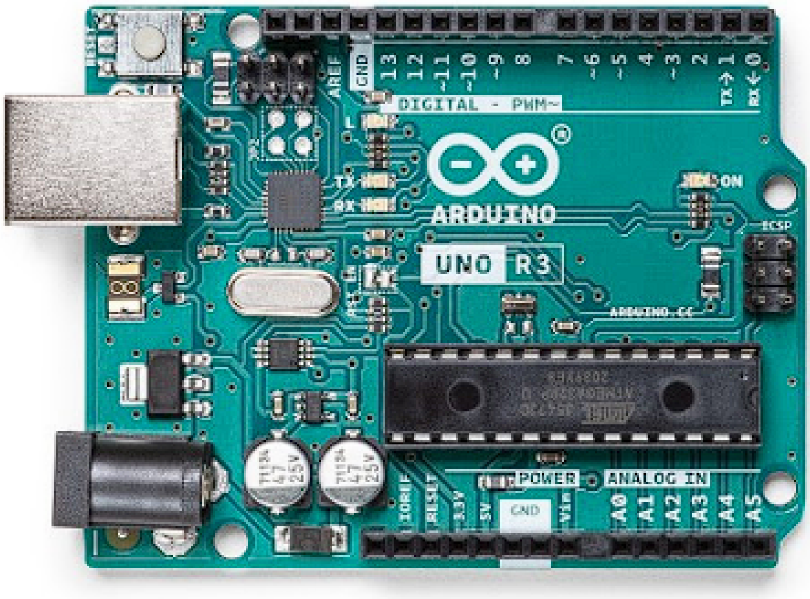


Figure C.1: Arduino Uno R3

ATMega328P Processor

- Memory
  - 32 kB Flash
  - 2 kB SRAM
  - 1 kB EEPROM
- CPU: AVR CPU running at up to 16 MHz
- Analog: 1x Analog Comparator (AC) with a scalable reference input
- Power: Operating voltage: 2.7-5.5 volts

- Security:
  - Power On Reset (POR)
  - Brown Out Detection (BOD)
- Timers/Counters:2x 8-bit Timer/Counter with a dedicated period register and compare channels. 1x 16-bit Timer/Counter with a dedicated period register, input capture and compare channels
- Communication:
  - 1x USART with fractional baud rate generator and start-of-frame detection
  - 1x controller/peripheral Serial Peripheral Interface (SPI)
  - 1x Dual mode controller/peripheral I2C

Appendix D

Data Sheet of Raspberry Pi 4 Model B

Figure D.1: Raspberry Pi 4 Model B

- SoC: Broadcom BCM2711
- CPU: 64-bit ARM Cortex-A72 (4x 1.5 GHz)
- GPU: Broadcom VideoCore VI
- RAM:8GB LPDDR4-3200 SDRAM
- Wireless: 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless
- Bluetooth: Bluetooth 5.0, BLE

- Ethernet: Gigabit Ethernet
- USB Ports:
  - 2 USB 3.0 ports
  - 2 USB 2.0 ports
- GPIO: Raspberry Pi standard 40 pin GPIO header
- HDMI Ports: 2 × micro-HDMI ports
- Display Port: 2-lane MIPI DSI display port
- Camera Port: 2-lane MIPI CSI camera port
- Audio/Video Port: 4-pole stereo audio and composite video port
- Storage: Micro-SD card slot for loading operating system and data storage
- Power Supply:
  - 5V DC via USB-C connector (minimum 3A)
  - 5V DC via GPIO header (minimum 3A)
  - Power over Ethernet (PoE) enabled

Appendix E

Data Sheet of Liquid-Crystal Display

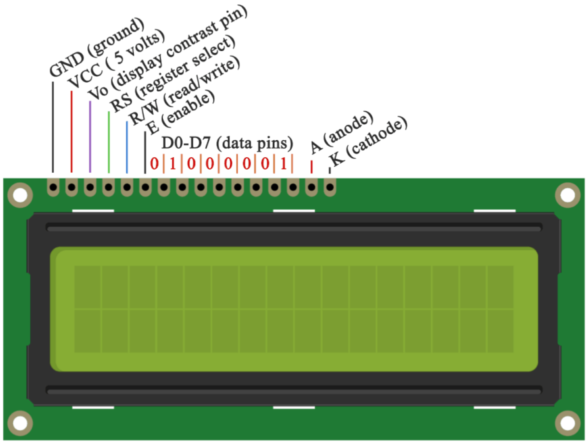


Figure E.1: Liquid-Crystal Display

- Controller IC: Usually HD44780 or equivalent
- Operating Voltage: 4.7V to 5.3V
- Display Format: 16 characters x 2 lines
- Character Size: 5×8 dot matrix
- Current Consumption: 1mA without backlight
- Backlight Options: Available in blue and green
- Viewing Area: Approx. 64.5mm x 16.4mm
- Operating Temperature: -20°C to 70°C
- Dot Size: Approx. 0.56mm x 0.66mm
- Backlight Current Consumption: Typically around 20mA to 40mA, depending on the backlight type