****

**MEDICAL PRESCRIPTION WRITING ROBOT**

## A PROJECT REPORT

***Submitted by***

**SUBITHA VARSHINI.V [REGISTER NO : 211417104271]**

**VENNILA.D [REGISTER NO : 211417104297]**

**HARITHA.V [REGISTER NO : 211417104319]**

***in partial fulfillment for the award of the degree of***

# BACHELOR OF ENGINEERING

**IN**

## COMPUTER SCIENCE AND ENGINEERING

**PANIMALAR ENGINEERING COLLEGE, CHENNAI-600123**

# ANNA UNIVERSITY : CHENNAI 600 025

**AUGUST 2021**

# BONAFIDE CERTIFICATE

**Certified that this project report “ MEDICAL PRESCRIPTION WRITING**

**ROBOT” is the bonafide work of “ SUBITHA VARSHINI.V [211417104271],**

**VENNILA.D [211417104297], HARITHA.V [211417104319] ” who carried out**

**the project work under my supervision.**

## SIGNATURE SIGNATURE

**Dr. S. MURUGAVALLI, M.E., Ph.D., Mrs. P. DEEPA**

**HEAD OF THE DEPARTMENT ASSOCIATE PROFESSOR**

**DEPARTMENT OF CSE, DEPARTMENT OF CSE,**

**PANIMALAR ENGINEERING PANIMALAR ENGINEERING**

**COLLEGE, NAZARATHPETTAI, COLLEGE, NAZARATHPETTAI,**

**POONAMALLEE, POONAMALLEE,**

**CHENNAI-600 123. CHENNAI-600 123.**

**Certified that the above candidate(s) was/ were examined in the Anna University Project Viva-Voce Examination held on 06/08/2021.**

## INTERNAL EXAMINER EXTERNAL EXAMINER

**ACKNOWLEDGEMENT**

We express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We would like to extend our heartfelt and sincere thanks to our Directors **Tmt.C.VIJAYARAJESWARI, Dr.C.SAKTHIKUMAR, M.E., Ph.D.** and **Tmt.SARANYASREE SAKTHIKUMAR B.E., M.B.A.** for providing us with the necessary facilities for completion of this project.

We also express our gratitude to our Principal **Dr.K.Mani, M.E., Ph.D.** for his timely concern and encouragement provided to us throughout the course.

We thank the HOD of CSE Department, **Dr. S.MURUGAVALLI,** **M.E., Ph.D.** for the support extended throughout the project.

We would like to thank our **Project Guide Mrs.P.Deepa, M.E.** and all the faculty members of the Department of CSE for their advice and suggestions for the successful completion of the project.

**SUBITHA VARSHINI.V**

**VENNILA.D**

**HARITHA.V**

**ABSTRACT**

This Project presents an approach to design rapid and fluid movements of a Universal robot to perform robot writing tasks. Reading a doctor’s handwritten prescription is a challenge that most patients and some pharmacists face; an issue that, in some cases, this lead to negative consequences due to wrong deciphering of the prescription. Part of the reason why doctor's prescriptions are so difficult to decipher is that doctors make use of Latin abbreviations and medical terminology that most people don't understand. The design of the proposed robot comprises both hardware and software. The hardware parts consist of the mechanical design of the robot, the adequate choice of the motors, and the electronic devices to properly drive the robot joints. The software part contains the high-level algorithms that convert the desired word to a sequence of target points, and the control algorithms that ultimately make the robot move according to the specifications. Here the writing mechanism is made by speech recognition technique. This speech recognition can be provided through either by using microphone or by using android applications. Thereby the robot can make the writing mechanism according to the user’s input.

# TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | **ABSTRACT** | **iii** |
|  | **LIST OF FIGURES** | **vi** |
|  | **LIST OF SYMBOLS, ABBREVIATIONS** | **vii** |
| **1.** | **INTRODUCTION** | **1** |
|  | **1.1 Overview** | **1** |
|  | **1.2 Problem Definition** | **1** |
| **2.** | **LITERATURE SURVEY** | **3** |
| **3.** | **SYSTEM ANALYSIS** | **10** |
|  | **3.1 Existing System** | **10** |
|  | **3.2 Proposed system** | **11** |
|  | **3.3 Requirement Analysis and Specification** | **11** |
|  | **3.3.1 Input Requirements** | **11** |
|  | **3.3.2 Output Requirements** | **13** |
|  | **3.3.3 Functional Requirements** | **14** |
|  | **3.4 Technology Stack** | **14** |

|  |  |  |  |
| --- | --- | --- | --- |
| **CHAPTER NO.** | | **TITLE** | **PAGE NO.** |
| **4.** | | **SYSTEM DESIGN** | **15** |
|  | | **4.1 Flow diagram** | **15** |
|  | **4.2 UML diagram** | | **16** |
|  | **4.3 Block diagram** | | **17** |
|  | **4.4 Process Flow** | | **17** |
| **5.** | | **SYSTEM ARCHITECTURE** | **18** |
|  | | **5.1 Architecture Overview** | **18** |
|  | | **5.2 Module Design Specification** | **20** |
| **6.** | | **SYSTEM IMPLEMENTATION** | **34** |
|  | | **6.1 Source Code** | **34** |
| **7.** | | **CONCLUSION** | **48** |
|  | | **7.1 Conclusion and Future Enhancements** | **48** |
|  | | **APPENDICES** |  |
|  | | **A.1 Sample Screens** | **49** |
|  | | **A.2 Publications** | **51** |

**REFERENCES 56**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | **LIST OF FIGURES** | |  | |
|  |  | | |  | |
| **FIGURE NO.** | | **TITLE** | | **PAGE NO.** | |
| **3.3.1** | | **BLUETOOTH COMMUNICATION** | | **12** | |
|  | |  | |  | |
| **4.1** | | **FLOW DIAGRAM** | | **15** | |
|  | |  | |  | |
| **4.2** | | **UML DIAGRAM** | | **16** | |
|  | |  | |  | |
| **4.3** | | **BLOCK DIAGRAM** | | **17** | |
|  | |  | |  | |
| **4.4** | | **PROCESS FLOW** | | **17** | |
|  | |  | |  | |
| **5.1** | | **TRANSMITTER AND RECEIVER SECTION** | | **18** | |
|  | |  | |  | |
| **5.2.1** | | **BLUETOOTH MODULE** | | **20** | |
|  | |  | |  | |
| **5.2.2** | | **MICROCONTROLLER PIN DIAGRAM** | | **22** | |
|  | |  | |  | |
| **5.2.3** | | **LCD DISPLAY** | | **22** | |
|  | |  | |  | |
| **5.2.4** | | **STEPPER MOTOR** | | **24** | |
|  | |  | |  | |
| **5.2.5** | | **SERVO MOTOR** | | **25** | |
|  | |  | |  | |
| **5.2.6** | | **BRIDGE RECTIFIER** | | **26** | |
|  | |  | |  | |
|  | | |  | |  | |
|  | | | **LIST OF ABBREVIATIONS** | |  | |
| **SCM** | | | **Spatial Covariance Matrix** | |  | |
| **TF** | | | **Time Frequency** | |  | |
| **DNN** | | | **Deep Neutral Network** | |  | |
| **IRM** | | | **Ideal Ratio Masks** | |  | |
| **HMM** | | | **Hidden Markav Models** | |  | |
| **A-ASR** | | | **Audio-Based Automatic Speed Recognition** | |  | |
| **CNN** | | | **Convolutional Neural Networks** | |  | |
| **LCD** | | | **Liquid Crystal Display** | |  | |
|  | | |  | |  | |
| **PS** | | | **Power Supply** | |  | |
|  | | |  | |  | |
| **PDA** | | | **Personal Digital Assistants** | |  | |
|  | | |  | |  | |
|  | | |  | |  | |
|  | | |  | |  | |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | |  |

**CHAPTER 1**

**INTRODUCTION**

* 1. **OVERVIEW**

Education system plays a major role in describing the innovative ideas to the students. In recent years, the definition of robot is generally used to mean an unmanned system or automation, as often seen in industrial applications. Generally, a robot is used to be shaped like humans, and referred to as machines and electric systems were capable of performing similar actions as humans. With the technological advancements in robotics field, efforts are being taken in researching, designing and development of robots for different practical purposes. Robots designed to assist human in their work and reduced human efforts. Nowadays, robots are designed to mimic human behaviour and perform tasks similar to human. Many research companies are developing robotic arm for performing basic functions like human arm. Among different functions, writing skills are one of function. The proposed robotic arm can be used by physically challenged person for writing operation. The main aim of developing the proposed system is to facilitate the physically challenged persons to write what they speak and also this design can be used in many other applications such as data accounting in industries can be done through wireless communication from one place to another. This reduces time and efforts of the workers.

* 1. **PROBLEM DEFINITION**

Medical prescription are the instructions which is given to the pharmacist for indicating what are the medicines that the patient should have to take but due to the handwriting of the doctors, some patients and pharmacists face many problems, they can’t understand the name of the written medicine. Due to this several life’s that could have been saved have been lost. In order to overcome this in our project we have come out with a robot prescription writing. Robot is a machine that designed to work automatically and it performs one or more tasks with speed. In our project we have a Bluetooth sensor which takes the data that is being given by the doctor. With the help of the Bluetooth receiver, it receives the data and the robot starts writing. With the help of the stepper motors the x and y axis can be controlled so that the text can be in order, and a server motor is being used in order to in order to rotate the pen. A prescription order should clearly communicate with a pharmacist/dispenser what therapy a particular patient is to get: how much of a specific medicine should be taken, how often and for how long. Children have a three-fold greater risk of experiencing a medication error than adults and are more likely to be harmed. Hence the components of a prescription should be clearly written, free from writing errors, non-official abbreviation, and fulfil the legal requirements of a prescription.

**CHAPTER -2**

**LITERATURE SURVEY**

**Title: Universal robot employment to mimic human writing.**

The paper presents an approach to design rapid and fluid movements of an universal robot to perform robot writing mimicking both the human kinematics when signing and the trajectory. To perform the task, on-line human signing standards are created first. Robot writing task is performed using these standards after that and robot signatures are acquired as a result. Next, both human and robot signatures are analyzed and compared, and the degree of signatures correlation is defined. Finally, recommendations of robot motion improvement are given.

**Title: Unsupervised Speech Enhancement Based on Multichannel NMF-Informed Beamforming for Noise-Robust Automatic Speech Recognition.**

Recently, the minimum variance distortion less response (MVDR) beam forming has widely been used because it works well if the steering vector of speech and the spatial covariance matrix (SCM) of noise are given. To estimating such spatial information, conventional studies take a supervised approach that classic search time-frequency(TF) bin in to noise or speech by training a deep neural network (DNN). The performance of ASR, however, is degraded in an unknown noisy environment. To solve this problem, we take an unsupervised approach that decompose search TF bin into the sum of speech and noise by using multichannel nonnegative matrix factorization (MNMF). This enables us to accurately estimate the SCMs of speech and noise not from observed noisy mixtures but from separated speech and noise components. In this paper, we propose online MVDR beam forming by effectively initializing and incrementally updating the parameters of MNMF. Another main contribution is to comprehensively investigate the performances of ASR obtained by various types of spatial ﬁlters, i.e., time-invariant and variant versions of MVDR beam formers and those of rank-1 and full-rank multichannel Wiener ﬁlters, in combination with MNMF. The experimental results showed that the proposed method out performed the state-of-the-art DNN-based beam forming method in unknown environments that did not match training data. This paper describes multichannel speech enhancement for improving automatic speech recognition (ASR) in noisy environments.

**Title: Speech Enhancement Based on Teacher-Student Deep Learning Using Improved Speech Presence Probability for Noise-Robust Speech Recognition.**

A teacher model with deep architectures is built to learn the target of ideal ratio masks (IRMs) using simulated training pairs of clean and noisy speech data. Next, a student model is trained to learn an improved speech presence probability by incorporating the estimated IRMs from the teacher model into the IMCRA approach. The student model can be compactly designed in a causal processing mode having no latency with the guidance of a complex and non-causal teacher model. Moreover, the clean speech requirement, which is difﬁcult to meet in real-world adverse environments, can be relaxed for training the student model, implying that noisy speech data can be directly used to adapt the regression-based enhancement model to further improve speech recognition accuracies for noisy speech collected in such conditions. Experiments on the CHiME-4 challenge task show that our best student model with bidirectional gated recurrent units (BGRUs) can achieve a relative word error rate (WER) reduction of 18.85% for the real test set when compared to unprocessed system without acoustic model retraining. However, the traditional teacher model degrades the performance of the unprocessed system in this case. In addition, the student model with a deep neural network (DNN) in causal mode having no latency yields a relative WER reduction of 7.94% over the unprocessed system with 670 times less computing cycles when compared to the BGRU-equipped student model. In this paper, we propose a novel teacher-student learning framework for the pre-processing of a speech recognizer, leveraging the online noise tracking capabilities of improved minima controlled recursive averaging (IMCRA) and deep learning of nonlinear interactions between speech and noise.

**Title: Spontaneous Speech Emotion Recognition Using Multiscale Deep Convolutional LSTM.**

Motivated by the diverse effects of different lengths of audio spectrograms on emotion identification, this paper proposes a multiscale deep convolutional long short-term memory (LSTM) framework for spontaneous speech emotion recognition. Initially, a deep convolutional neural network (CNN) model is used to learn deep segment-level features on the basis of the created image-like three channels of spectrograms. Then, a deep LSTM model is adopted on the basis of the learned segment-level CNN features to capture the temporal dependency among all divided segments in an utterance for utterance-level emotion recognition. Finally, different emotion recognition results, obtained by combining CNN with LSTM at multiple lengths of segment-level spectrograms, are integrated by using a score-level fusion strategy.Emotion recognition in real sceneries such as in the wild has attracted extensive attention in affective computing, because existing spontaneous emotions in real sceneries are more challenging and difficult to identify than other emotions. Experimental results on two challenging spontaneous emotional datasets, i.e., the AFEW5.0 and BAUM-1s databases, demonstrate the promising performance of the proposed method, outperforming state-of-the-art methods.

**Title: Deep Audio-visual Speech Recognition.**

Unlike previous works that have focused on recognizing a limited number of words or phrases, we tackle lip reading as an open-world problem – unconstrained natural language sentences, and in the wild videos. Our key contributions are: (1) we compare two models for lip reading, one using a CTC loss, and the other using a sequence-to-sequence loss. Both models are built on top of the transformer self-attention architecture; (2) we investigate to what extent lip reading is complementary to audio speech recognition, especially when the audio signal is noisy; (3) we introduce and publicly release two new datasets for audio-visual speech recognition: LRS2-BBC, consisting of thousands of natural sentences from British television; and LRS3-TED, consisting of hundreds of hours of TED and TEDx talks obtained from YouTube. The models that we train surpass the performance of all previous work on lip reading benchmark datasets by a signiﬁcant margin. The goal of this work is to recognize phrases and sentences being spoken by a talking face, with or without the audio.

**Title: Gating Neural Network for Large Vocabulary Audio visual Speech Recognition.**

Audio-based automatic speech recognition (A-ASR) systems are affected by noisy conditions in real-world applications. Adding visual cues to the ASR system is an appealing alternative to improve the robustness of the system, replicating the audio visual perception process used during human interactions. A common problem observed when using audio visual automatic speech recognition (AV-ASR) is the drop in performance when speech is clean. In this case, visual features may not provide complementary information, introducing variability that negatively affects the performance of the system. The experimental evaluation in this study clearly demonstrates this problem when we train an audio visual state-of-the-art hybrid system with a deep neural network (DNN) and hidden Markov models (HMMs). This study proposes a framework that addresses this problem, improving, or at least, maintaining the performance when visual features are used. The proposed approach is a deep learning solution with a gating layer that diminishes the effect of noisy or uninformative visual features, keeping only useful information. The framework is implemented with a subset of audio visual CRSS-4ENGLISH-14 corpus which consists of 61 hours of speech from 105 subjects simultaneously collected with multiple cameras and microphones. The proposed framework is compared with conventional HMMs with observation models implemented with either a Gaussian mixture model (GMM) or DNNs. We also compare the system with a multi-stream hidden Markov model (MS-HMM) system.

**Title: Speech Emotion Recognition Using Deep Convolutional Neural Network and Discriminant Temporal Pyramid Matching.**

Deep Convolutional Neural Networks (DCNN) has exhibited remarkable success in bridging the semantic gap in visual tasks like image classiﬁcation, object detection. This paper explores how to utilize a DCNN to bridge the affective gap in speech signals. To this end, we ﬁrstly extract three channels of log Mel-spectrograms (static, delta and delta-delta) similar to the RGB image representation as the DCNN input. Then the Alex Net DCNN model pre-trained on the large ImageNet dataset is employed to learn high-level feature representations on each segment divided from an utterance. The learned segment-level features are aggregated by a Discriminant Temporal Pyramid Matching (DTPM) strategy. DTPM combines temporal pyramid matching and optimal Lp-norm pooling to form a global utterance-level feature representation, followed by the linear Support Vector Machines (SVM) for emotion classiﬁcation. Experimental results on four public datasets, i.e., EMO-DB, RML, eNTERFACE05 and BAUM-1s, show the promising performance of our DCNN model and the DTPM strategy. Another interesting ﬁnding is that the DCNN model pre-trained for image applications performs reasonably good in affective speech feature extraction.

**Title: Neural Network based Multi-Factor Aware Joint Training for Robust Speech Recognition.**

Neural Network based Multi-Factor Aware Joint Training, is proposed to improve the recognition accuracy for noise robust speech recognition. This approach is a structured model which integrates several different functional modules into one computational deep model. We explore and extract speaker, phone and environment factor representations using deep neural networks(DNNs), which are integrated into the main ASR DNN to improve classiﬁcation accuracy. In addition, the hidden activations in the main ASR DNN are used to improve factor extraction, which in turn helps the ASR DNN. All the model parameters, including those in the ASR DNN and factor extraction DNNs, are jointly optimized under the multitask learning framework. Unlike prior traditional techniques for the factor-aware training, our approach requires no explicit separate stages for factor extraction and adaptation. Moreover, the proposed neural network based multi-factor aware joint training can be easily combined with the conventional factor aware training which uses the explicit factors, such as i-vector, noise energy and T60 value to obtain additional improvement. The proposed method is evaluated on two main noise robust tasks: the AMI single distant microphone (SDM) task in which reverberation is the main concern, and the Aurora4 task in which multiple noise types exist.

**Title: Novel Unsupervised Auditory Filter bank Learning Using Convolutional RBM for Speech Recognition.**

we have proposed an unsupervised learning model based on convolutional RBM with rectiﬁed linear units. In this paper, theory, training algorithm of our proposed model and detailed analysis of learned ﬁlter bank is being presented. Learning of the model with different databases shows that model is able to learn cochlear-like impulse responses which are localized in frequency-domain. Auditory-like scale obtained from ﬁlter banks learned from clean and noisy data sets resemble Mel scale which is known to mimicperceptually relevant aspect of speech. We have experimented with both cepstral (denoted as ConvRBM-CC) as well as ﬁlter bank features (denoted as ConvRBM-BANK). On LVCSR task, we achieved relative improvement of 7.21-17.8 % in WER compared to MFCC features and 1.35-6.82 % compared to FBANK features. On AURORA 4 multi-condition training database, the relative improvement in WER by 4.8-13.65 % was achieved using DNNHMM system with ConvRBM-CC features. Using ConvRBMBANK features, we achieve absolute reduction of 1.25-3.85 % in WER on AURORA 4 test sets compared to FBANK features. Context-dependent DNN-HMM system further improves performance with a relative improvement of 3.6-4.6 % and on an averageforbi-gram5Kandtri-gram5Klanguagemodels.Hence, our proposed learned ﬁlter-bank perform better than traditional MFCC and Mel-ﬁlter bank features for both clean and multi condition ASR task. System combination of ConvRBM-BANK and FBANK features further improve performance in all ASR tasks. We have also performed cross-domain experiments where sub-band ﬁlters trained on one database is used for ASR task of another database. In this we have shown that our model learns generalized representations of speech signals.

**Title: A Joint Training Framework for Robust Automatic Speech Recognition.**

A novel joint training framework for speech presents separation and recognition. The key idea is to concatenate a deep neural network (DNN) based speech separation frontend and a DNN-based acoustic model to build a larger neural network, and jointly adjust the weights in each module. This way, the separation frontend is able to provide enhanced speech desired by the acoustic model and the acoustic model can guide the separation frontend to produce more discriminative enhancement. In addition, we apply sequence training to the jointly trained DNN so that the linguistic information contained in the acoustic and language models can be back-propagated to influence the separation frontend at the training stage. To further improve the robustness, we add more noise- and reverberation-robust features for acoustic modelling. At the test stage, utterance-level unsupervised adaptation is performed to adapt the jointly trained network by learning a linear transformation of the input of the separation frontend. The resulting sequence-discriminative jointly-trained multi-stream system with run-time adaptation achieves 10.63% average word error rate (WER) on the test set of the reverberant and noisy CHiME-2 dataset (task-2), which represents the best performance on this dataset and a 22.75% error reduction over the best existing method.

**CHAPTER 3**

**SYSTEM ANALYSIS**

**3.1 EXISTING SYSTEM**

The system will be divided into two sections one will be transmitter section and other will be receiver section. The transmitter section will consist of one Arduino Uno, one 3-axis accelerometer and one RF transmitter module. The receiver section consists of one RF receiver module, one motor driver IC, two PMDC motor, two wheels. Here we will require two separate 5 Volt power supply which will be applied to both the sections. The robot moves forward, backward, right and left when there is tilt in the palm of user in forward, backward, right and left respectively directions. A gesture-controlled robot can be controlled by using hand in place of any other method like buttons or joystick. Here one only needs to move hand to control the robot. A transmitting device is used in your hand which contains RF Transmitter and accelerometer. This will transmit command to robot so that it can do the required task like moving forward, reverse, turning left, turning right and stop. All these tasks will be performed by using hand gesture. Here the most important component is accelerometer.

**DISADVANTAGES**

1. Time Consumption high
2. Manual Work
3. Mechanism little much difficult to implement.

**3.2 PROPOSED SYSTEM**

The writing robot makes the written prescription chit about the patients with the help of wireless communication. The movement G-Code file created by the help of Inkscape software then the processing software is used to send the G-Code file to the microcontroller. Then the CNC shield drive sends the controlling signals to the stepper motors and servo motor. Now the XY axis which operates as follows by the instructions given to the controller unit. The corresponding code is sending the data to controller block is interfaced with motor driver unit along the DAC provides the pulse width signal to motor unit where it has been processed and final output is written and displayed on the paper from the output unit.

**ADVANTAGES**

1. Accuracy
2. Optimized character axis
3. Less delay.
4. Low cost

**3.3 REQUIREMENT ANALYSIS AND SPECIFICATION**

**3.3.1 INPUT REQUIREMENTS**

Send data from Smartphone terminal to HC-05 Bluetooth module and see this data on PC serial terminal and vice versa.

To communicate smartphone with HC-05 Bluetooth module, smartphone requires Bluetooth terminal application for transmitting and receiving data. You can find Bluetooth terminal applications for android and windows in respective app. store.

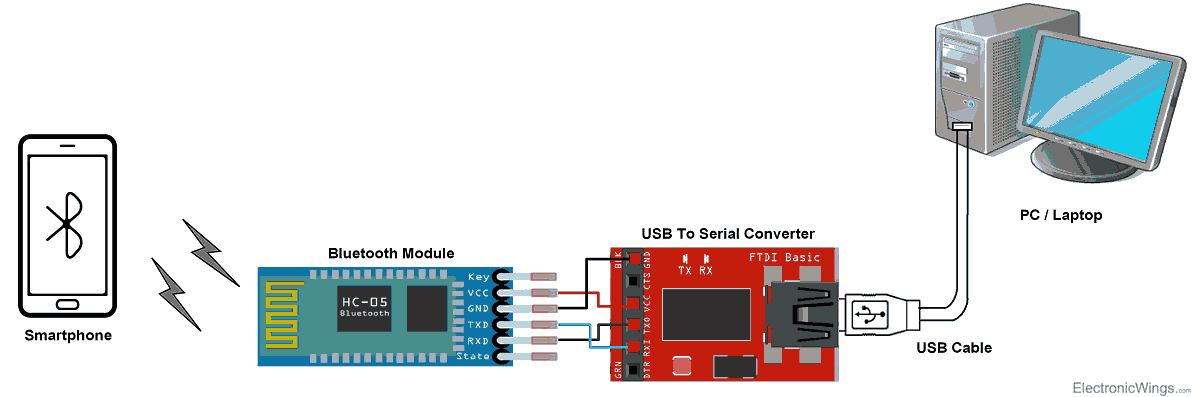


Figure 4.5: Bluetooth communication

**Bluetooth Module Serial Interface**

So, when we want to communicate through smartphone with HC-05 Bluetooth module, connect this HC-05 module to the PC via serial to USB converter.

Before establishing communication between two Bluetooth devices, 1st we need to pair HC-05 module to smartphone for communication.

**Pair HC-05 and smartphone**:

1. Search for new Bluetooth device from your phone. You will find Bluetooth device with “HC-05” name.
2. Click on connect/pair device option; default pin for HC-05 is 1234 or 0000.

After pairing two Bluetooth devices, open terminal software (e.g. Teraterm, Realterm etc.) in PC, and select the port where we have connected USB to serial module. Also select default baud rate of 9600 bps.

In smart phone, open Bluetooth terminal application and connect to paired device HC-05.

It is simple to communicate; we just have to type in the Bluetooth terminal application of smartphone. Characters will get sent wirelessly to Bluetooth module HC-05. HC-05 will automatically transmit it serially to the PC, which will appear on terminal. Same way we can send data from PC to smartphone.

**3.3.2 OUTPUT REQUIREMENTS**

**Servo Mechanism:**

It consists of three parts:

1. Controlled device
2. Output sensor
3. Feedback system

### **Types of Stepper Motor**:

There are three main types of stepper motors, they are:

1. Permanent magnet stepper
2. Hybrid synchronous stepper
3. Variable reluctance stepper

**Permanent Magnet Stepper Motor:**Permanent magnet motors use a permanent magnet (PM) in the rotor and operate on the attraction or repulsion between the rotor PM and the stator electromagnets.

**Variable Reluctance Stepper Motor:**Variable reluctance (VR) motors have a plain iron rotor and operate based on the principle that minimum reluctance occurs with minimum gap, hence the rotor points are attracted toward the stator magnet poles.

**Hybrid Synchronous Stepper Motor:**Hybrid stepper motors are named because they use a combination of permanent magnet (PM) and variable reluctance (VR) techniques to achieve maximum power in a small package size.

**3.3.3 FUNCTIONAL REQUIREMENTS**

The power supply circuit consists of step-down transformer which is 230v step down to 12v.In this circuit 4diodes are used to form bridge rectifier which delivers pulsating dc voltage & then fed to capacitor filter the output voltage from rectifier is fed to filter to eliminate any a.c. components present even after rectification.

The filtered DC voltage is given to regulator to produce 12v constant DC voltage. 230V AC power is converted into 12V AC (12V RMS value wherein the peak value is around 17V), but the required power is 5V DC; for this purpose, 17V AC power must be primarily converted into DC power then it can be stepped down to the 5V DC. AC power can be converted into DC using one of the power electronic converters called as Rectifier. There are different types of rectifiers, such as half-wave rectifier, full-wave rectifier and bridge rectifier. Due to the advantages of the bridge rectifier over the half and full wave rectifier, the bridge rectifier is frequently used for converting AC to DC.

**3.4 TECHNOLOGY STACK**

**HARDWARE REQUIREMETS**

1. ATMega328 MICRO CONTROLLER
2. Bluetooth Module
3. Liquid Crystal Display
4. Stepper Motor Driver
5. Stepper Motors and Servo Motor
6. Gear and Belt Setups

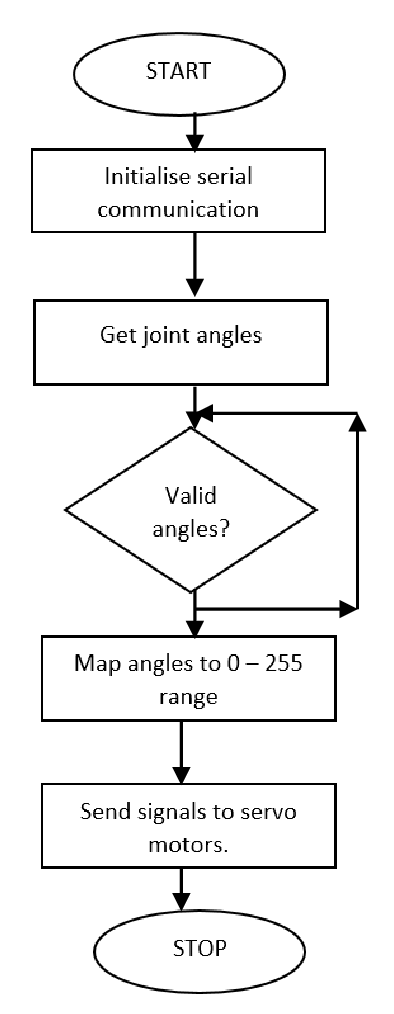
**SOFTWARE REQUIREMENTS**

1. EMBEDDED C
2. Android Application
3. Android IDE

**CHAPTER 4**

**SYSTEM DESIGN**

**4.1 FLOW DIAGRAM**

****

**4.2 UML DIAGRAM**

**Writes prescription**

**Bluetooth Module**

**Transfer via Bluetooth**

**Recognize**

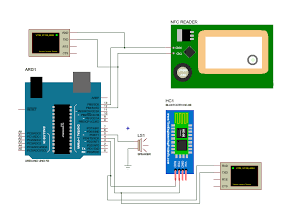
**User Voice**

**LOGIN**

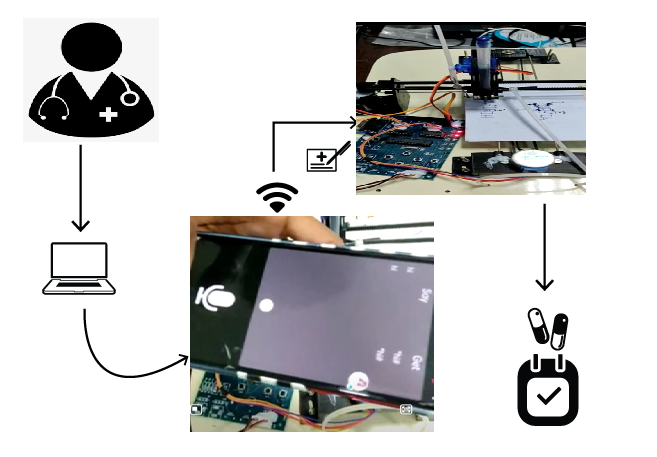
**USER**

**ROBOT**

**4.3 BLOCK DIAGRAM**



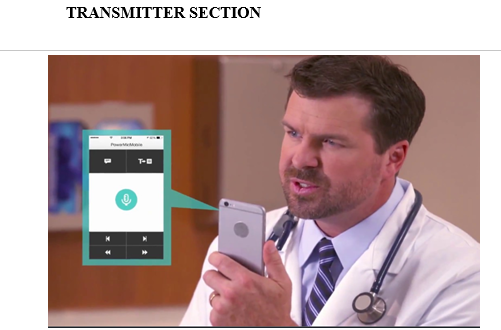
**4.4 PROCESS FLOW**

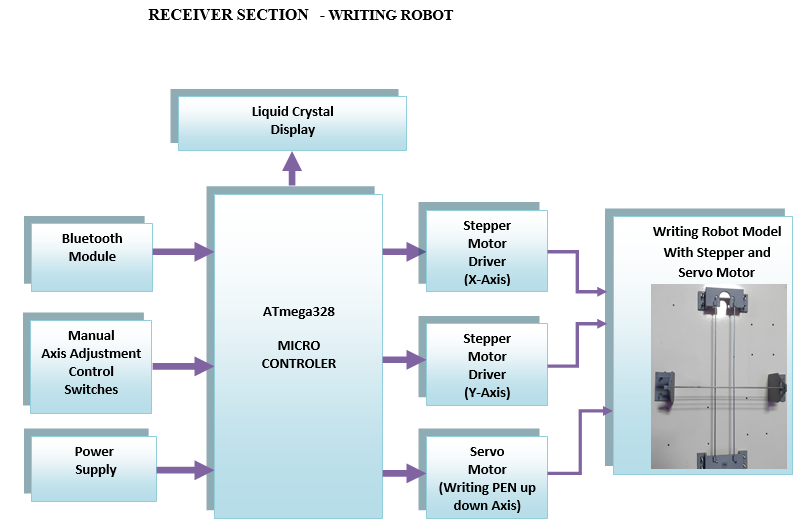


**CHAPTER 5**

**SYSTEM ARCHITECTURE**

**5.1 ARCHITECTURE OVERVIEW**





The power supply required for Arduino and stepper motor as well as servo motor is 5V and 12V.Hence stepdown transformer, bridge rectifier, filter and voltage regulators are used. The voice or speech is recognized by the Bluetooth module through android application or microphone. Then by using Inkscape software and Arduino IDE software the analog signal is converted into digital signal and then it is interfaced with the Arduino Uno. The Arduino Uno sends the signals to the respective x and y axis stepper motor and servo motor. According to the Arduino commands, the motors move and rotates along the required axis. Finally, the writing operation is done according to the user’s input

**1. TRANSMISSION SECTION**

In the Transmitter section, user gives the data to the receiver with the help of the Bluetooth that is present in the mobile app (Bluetooth voice). In transmitter section it contains a Bluetooth device named HC-05 that helps to connect to the receiver Bluetooth for transmitting the data, First the user has to store the required data in Bluetooth voice app. Then the receiver receives the data from the transmitter section. Thus, it recognizes the data and sends it to the receiver section. Then the receiver starts doing its work.

**2. RECEIVER SECTION**

In the receiver section it receives the data from the transmitter and then with the help of ATmega328P the audio is converted into text. The manual axis adjustment is used to adjust the direction. The power supply helps to transmit power to the kit. The Stepper motor is used to drive through the x & y direction and the Servo motor is used for writing the pen in up-down axis. Then the writing robot starts writing data which has been received from the transmitter.

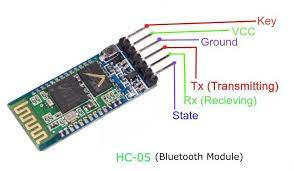
Our project proposes a robot that is used for prescription writing. It designs a rapid fluid movement of a universal robot to perform dual axis control robot for writing medical prescription. The robot is connecting to 5V power supply. Then it connects with then Bluetooth device. Thus, the user gives the data and then the Bluetooth transmitter gives the data to the Arduino (ATmega328P), then it starts interfacing with the program in kit. The both ULN2003 then drive the stepper motors that are connected with the dual axis that is the X and Y axis. The stepper and servo motor are moving with help of gear and belt setups. And the stepper motor starts working to rotate the writing pad. Once the data is received the server motor helps the pen to write the text (in order to convert the audio into text). Hence the text is written.

**5.2 MODULE DESIGN SPECIFICATION**

**HARDWARE DESCRIPTION:**

1. **BLUETOOTH MODULE**

Bluetooth is a telecommunications industry specification that describes how mobile phones, computers, and personal digital assistants (PDAs) can be easily interconnected using a short-range wireless connection. Using this technology, users of phones, pagers, and personal digital assistants can buy a three-in-one phone that can double as a portable phone at home or in the office, get quickly synchronized with information in a desktop or notebook computer, initiate the sending or receiving of a fax, initiate a print-out, and, in general, have all mobile and fixed computer devices be totally coordinated. The maximum range is 10 meters. Data can be exchanged at a rate of 1 megabit per second (up to 2 Mbps in the second generation of the technology). A frequency hop scheme allows devices to communicate even in areas with a great deal of electromagnetic interference. Built-in encryption and verification are provided.



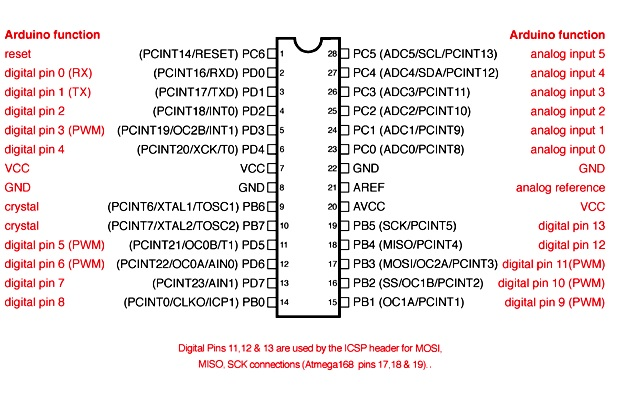
1. **ATMega328 MICRO CONTROLLER**

The Atmel AVR® core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in a single instruction executed in one clock cycle.

The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers. The ATmega328/P provides the following features: 32Kbytes of In-System Programmable Flash with Read-While-Write capabilities.

* 1Kbytes EEPROM
* 2Kbytes SRAM
* 23 general purpose I/O lines
* 32 general purpose working registers
* Real Time Counter (RTC)
* three flexible Timer/Counters with compare modes and PWM
* 1 serial programmable USARTs
* 1 byte-oriented 2-wire Serial Interface (I2C)
* 6- channel 10-bit ADC (8 channels in TQFP and QFN/MLF packages)
* programmable Watchdog Timer with internal Oscillator,
* SPI serial port, and six software selectable power saving modes.

The Boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega328/P is a powerful microcontroller that provides a highly flexible and cost-effective solution to many embedded control applications.



1. **LCD**

LCD stands for Liquid Crystal Display. LCD is finding wide spread use replacing LEDs (seven segment LEDs or other multi segment LEDs) because of the following reasons:

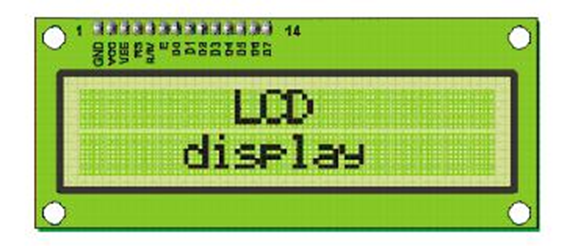
The declining prices of LCDs.

The ability to display numbers, characters and graphics. This is in contrast to LEDs, which are limited to numbers and a few characters.

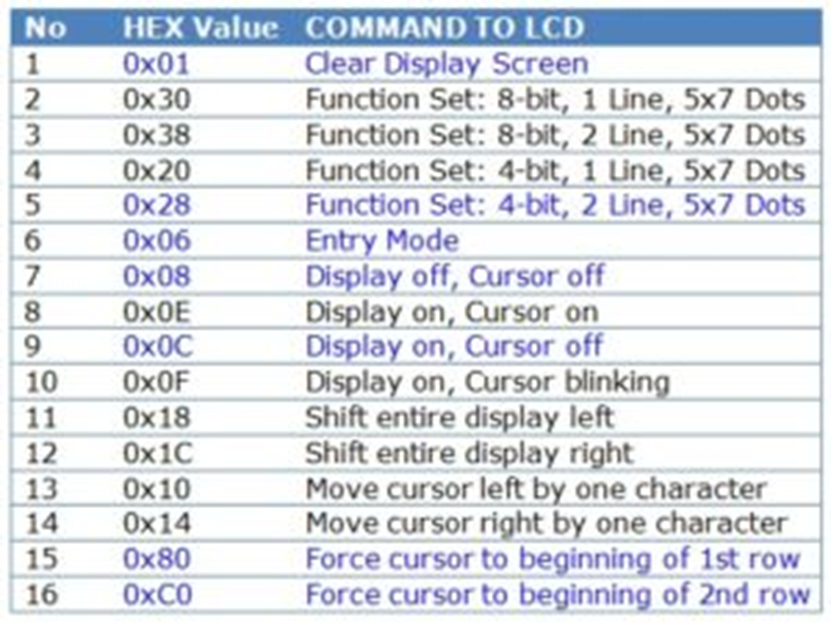
Incorporation of a refreshing controller into the LCD, thereby relieving the CPU of the task of refreshing the LCD. In contrast, the LED must be refreshed by the CPU to keep displaying the data.

Ease of programming for characters and graphics.

These components are “specialized” for being used with the microcontrollers, which means that they cannot be activated by standard IC circuits. They are used for writing different messages on a miniature LCD.

****

**LCD COMMANDS:**

****

All data transferred to LCD through outputs D0-D7 will be interpreted as commands or as data, which depends on logic state on pin RS:

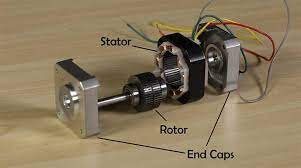
RS = 1 - Bits D0 - D7 are addresses of characters that should be displayed. Built in processor addresses built in “map of characters” and displays corresponding symbols.

Displaying position is determined by DDRAM address. This address is either previously defined or the address of previously transferred character is automatically incremented.

RS = 0 - Bits D0 - D7 are commands which determine display mode.

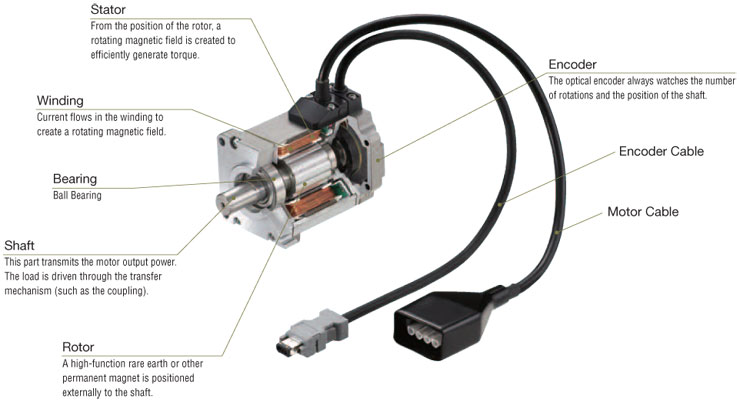
1. **STEPPER MOTOR**

Stepper motor (also called as step motor) is basically a brushless DC motor, whose rotor rotates through a fixed angular step in response to input current pulse. That means, the full rotation of the rotor is divided into equal number of steps, and rotor rotates through one step for each current pulse. Stepper motors are becoming very popular due to the fact that they can be controlled directly by computers, microprocessor. A stepper motor is a type of DC motor which has a full rotation divided in an equal number of steps. It is a type of actuator highly compatible with numerical control means, as it is essentially an electromechanical converter of digital impulses into proportional movement of its shaft, providing precise speed, position and direction control in an open-loop fashion, without requiring encoders, end-of-line switches or other types of sensors as conventional electric motors require.



**5. SERVO MOTOR**

A servo motor allows precise control of linear or angular position, speed. And it is a rotary actuator. It involves a sensor which is coupled to a motor for location feedback. It also requires a servo driver to send the feedback about the PWM signals. It refers to a motor appropriate for use in a closed-loop control system. Applications of Servomotors are Metal Cutting and Metal Forming Machines and solar tracking system. When the beam of the motor at the preferred location, power supply to the motor is stopped. If not, the motor is twisted in the suitable direction.



1. **POWER SUPPLY**

A bridge rectifier circuit is a common part of the electronic power supplies. Many electronic circuits require rectified DC power supply for powering the various electronic basic components from available AC mains supply. We can find this rectifier in a wide variety of electronic AC power devices like home appliances, motor controllers, modulation process, welding applications, etc.

What is a Bridge Rectifier?

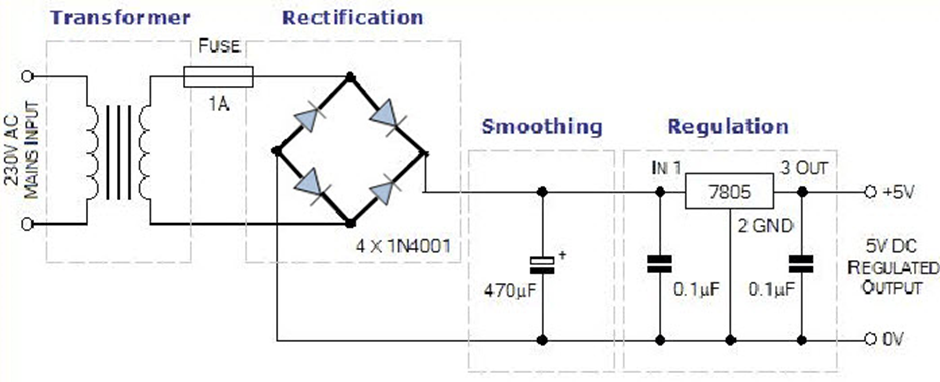
A Bridge rectifier is an Alternating Current (AC) to Direct Current (DC) converter that rectifies mains AC input to DC output. Bridge Rectifiers are widely used in power supplies that provide necessary DC voltage for the electronic components or devices. They can be constructed with four or more diodes or any other controlled solid-state switches.

Depending on the load current requirements, a proper bridge rectifier is selected. Components’ ratings and specifications, breakdown voltage, temperature ranges, transient current rating, forward current rating, mounting requirements and other considerations are taken into account while selecting a rectifier power supply for an appropriate electronic circuit’s application.

The main advantage of bridge rectifier is that it produces almost double the output voltage as with the case of a full wave rectifier using center-tapped transformer. But this circuit doesn’t need center tapped transformer so it resembles low-cost rectifier.

The bridge rectifier circuit diagram consists of various stages of devices like transformer, Diode Bridge, filtering and regulators. Generally, all these blocks combination is called as regulated DC power supply that powers various electronic appliances.

The first stage of the circuit is a transformer which is a step-down type that changes the amplitude of the input voltage. Most of the electronic projects uses 230/12V transformer to step-down the AC mains 230V to 12V AC supply.



Next stage is a diode-bridge rectifier which uses four or more diodes depending on the type of bridge rectifier. Choosing a particular diode or any other switching device for a corresponding rectifier needs some considerations of the device like Peak Inverse Voltage (PIV), forward current If, voltage ratings, etc. It is responsible for producing unidirectional or DC current at the load by conducting a set of diodes for every half cycle of the input signal.

Since the output after the diode bridge rectifiers is of pulsating nature, and for producing it as a pure DC, filtering is necessary. Filtering is normally performed with one or more capacitors attached across the load, as you can observe in the below figure wherein smoothing of wave is performed. This capacitor rating also depends on the output voltage.

The last stage of this regulated DC supply is a voltage regulator that maintains the output voltage to a constant level. Suppose the microcontroller works at 5V DC, but the output after the bridge rectifier is around 16V, so to reduce this voltage, and to maintain a constant level – no matter voltage changes in input side – a voltage regulator is necessary.

**VOLTAGE REGULATOR:**

The name says it all: voltage regulator. The battery in your car that gets charged from the alternator, the outlet in your home that provides all the electricity you desire, the cell phone you likely keep on-hand every minute of the day they all require a specific voltage in order to function. Fluctuating outputs that jump from ±2V can cause inefficient operation and possibly even damage to your charging devices. There's a variety of reasons why a voltage fluctuation may occur: condition of the power grid, other appliances turning off and on, time of day, environmental factors, etc. Due to the need for a steady, constant voltage, enter the voltage regulator.

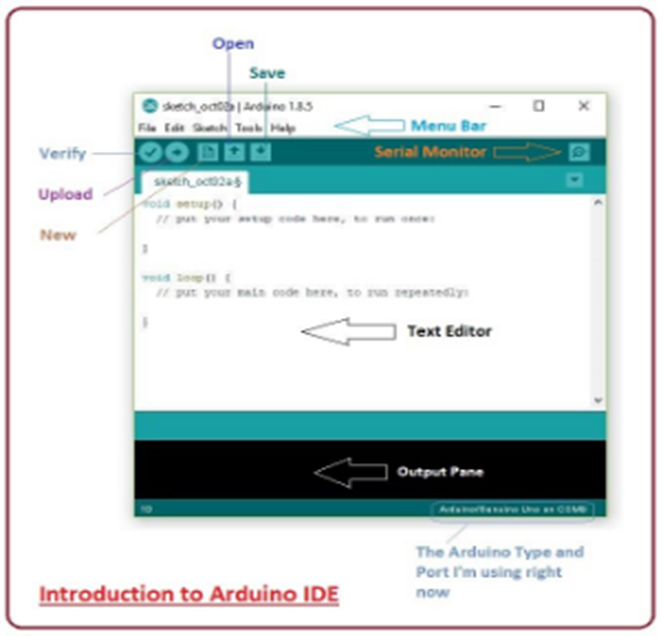
A voltage regulator is an integrated circuit (IC) that provides a constant fixed output voltage regardless of a change in the load or input voltage. It can do this many ways depending on the topology of the circuit within, but for the purpose of keeping this project basic, we will mainly focus on the linear regulator. A linear voltage regulator works by automatically adjusting the resistance via a feedback loop, accounting for changes in both load and input, all while keeping the output voltage constant.

**FILTER**

In power supplies, capacitors are used to smooth (filter) the pulsating DC output after rectification so that a nearly constant DC voltage is supplied to the load. The pulsating output of the rectifiers has an average DC value and an AC portion that is called ripple voltage. Filter capacitors reduce the amount of ripple voltage to a level that is acceptable. It should be noted that resistors and inductors can be combined with the capacitors to form filter networks. Here we will concentrate on capacitive filters only. In a filter circuit the capacitor is charged to the peak of the rectified input voltage during the positive portion of the input. When the input goes negative, the capacitor begins to discharge into the load. The rate of discharge is determined by the RC time constant formed by the capacitor and the load's resistance. Filters is used to removes ripples and noise.

**SOFTWARE DESCRIPTION:**

Introduction to Arduino IDE where IDE stands for Integrated Development Environment – An official software introduced by Arduino.cc, that is mainly used for writing, compiling and uploading the code in the Arduino Device. Almost all Arduino modules are compatible with this software that is an open source and is readily available to install and start compiling the code on the go.

****

Arduino IDE is an open-source software that is mainly used for writing and compiling the code into the Arduino Module.

It is an official Arduino software, making code compilation too easy that even a common person with no prior technical knowledge can get their feet wet with the learning process.

It is easily available for operating systems like MAC, Windows, Linux and runs on the Java Platform that comes with inbuilt functions and commands that play a vital role for debugging, editing and compiling the code in the environment.

A range of Arduino modules available including Arduino Uno, Arduino Mega, Arduino Leonardo, Arduino Micro and many more.

Each of them contains a microcontroller on the board that is actually programmed and accepts the information in the form of code.

The main code, also known as a sketch, created on the IDE platform will ultimately generate a Hex File which is then transferred and uploaded in the controller on the board.

The IDE environment mainly contains two basic parts: Editor and Compiler where former is used for writing the required code and later is used for compiling and uploading the code into the given Arduino Module.

This environment supports both C and C++ languages.

The IDE environment is mainly distributed into three sections

1. Menu Bar

2. Text Editor

3. Output Pane

The bar appearing on the top is called Menu Bar that comes with five different options as follow

File – You can open a new window for writing the code or open an existing one. Following table shows the number of further subdivisions the file option is categorized into.

As you go to the preference section and check the compilation section, the Output Pane will show the code compilation as you click the upload button.

And at the end of compilation, it will show you the hex file it has generated for the recent sketch that will send to the Arduino Board for the specific task you aim to achieve.

Edit – Used for copying and pasting the code with further modification for font

Sketch – For compiling and programming

Tools – Mainly used for testing projects. The Programmer section in this panel is used for burning a bootloader to the new microcontroller.

Help – In case you are feeling skeptical about software, complete help is available from getting started to troubleshooting.

The Six Buttons appearing under the Menu tab are connected with the running program as follow.

The check mark appearing in the circular button is used to verify the code. Click this once you have written your code.

The arrow key will upload and transfer the required code to the Arduino board.

The dotted paper is used for creating a new file.

The upward arrow is reserved for opening an existing Arduino project.

The downward arrow is used to save the current running code.

The button appearing on the top right corner is a Serial Monitor – A separate pop-up window that acts as an independent terminal and plays a vital role for sending and receiving the Serial Data. You can also go to the Tools panel and select Serial Monitor, or pressing Ctrl+Shift+M all at once will open it instantly. The Serial Monitor will actually help to debug the written Sketches where you can get a hold of how your program is operating. Your Arduino Module should be connected to your computer by USB cable in order to activate the Serial Monitor.

You need to select the baud rate of the Arduino Board you are using right now. For my Arduino Uno Baud Rate is 9600, as you write the following code and click the Serial Monitor.

Libraries

Libraries are very useful for adding the extra functionality into the Arduino Module. There is a list of libraries you can add by clicking the Sketch button in the menu bar and going to Include Library.

As you click the Include Library and Add the respective library it will on the top of the sketch with a #include sign. Suppose, I Include the EEPROM library, it will appear on the text editor as

#include <EEPROM.h>.

Most of the libraries are preinstalled and come with the Arduino software. However, you can also download them from the external sources.

Making Pins Input or Output

The digital Read and digital Write commands are used for addressing and making the Arduino pins as an input and output respectively.

These commands are text sensitive i.e., you need to write them down the exact way they are given like digital Write starting with small “d” and write with capital “W”. Writing it down with Digital write or digital write won’t be calling or addressing any function.

How to Select the Board

In order to upload the sketch, you need to select the relevant board you are using and the ports for that operating system. As you click the Tools on the Menu, it will open like the figure below.

Just go to the “Board” section and select the board you aim to work on. Similarly, COM1, COM2, COM4, COM5, COM7 or higher are reserved for the serial and USB board. You can look for the USB serial device in the ports section of the Windows Device Manager.

Following figure shows the COM4 that I have used for my project, indicating the Arduino Uno with COM4 port at the right bottom corner of the screen.

After correct selection of both Board and Serial Port, click the verify and then upload button appearing in the upper left corner of the six button section or you can go to the Sketch section and press verify/compile and then upload.

The sketch is written in the text editor and is then saved with the file extension. ino.

It is important to note that the recent Arduino Modules will reset automatically as you compile and press the upload button the IDE software, however, older version may require the physical reset on the board.

Once you upload the code, TX and RX LEDs will blink on the board, indicating the desired program is running successfully.

Note: The port selection criteria mentioned above is dedicated for Windows operating system only, you can check this Guide if you are using MAC or Linux.

The amazing thing about this software is that no prior arrangement or bulk of mess is required to install this software, you will be writing your first program within 2 minutes after the installation of the IDE environment.

Bootloader

As you go to the Tools section, you will find a bootloader at the end. It is very helpful to burn the code directly into the controller, setting you free from buying the external burner to burn the required code.

When you buy the new Arduino Module, the bootloader is already installed inside the controller. However, if you intend to buy a controller and put in the Arduino module, you need to burn the bootloader again inside the controller by going to the Tools section and selecting the burn bootloader.

**CHAPTER 6**

**SYSTEM IMPLEMENTATION**

**6.1 SOURCE CODE**

MultiStepper.0 steppers;

Servo penServo;

const int penServoPin =7 ;

const int penZUp = 125;

const int penZDown = 95;

String inputString = ""; // a string to hold incoming data

boolean stringComplete = false; // whether the string is complete

long posx,posy;

long positions[2];

void mow(long x,long y)

{

posx=posx+x;

posy=posy+y;

positions[0] = posx;

positions[1] = posy;

steppers.moveTo(positions);

steppers.runSpeedToPosition(); // Blocks until all are in position

Serial.print("X=");

Serial.print(posx);

Serial.print(" Y=");

Serial.println(posy);

}

int c,j=1;

char inChar;

long count;

int i;

void main()

{

Serial.begin(9600);

inputString.reserve(200);

Serial.println("Writing Robot");

stepper1.setMaxSpeed(300);

stepper2.setMaxSpeed(300);

penServo.attach(penServoPin);

penServo.write(penZUp);

delay(100);

steppers.addStepper(stepper1);

steppers.addStepper(stepper2);

if(digitalRead(8)==0)

{

mow(0,10);

}

if(digitalRead(9)==0)

{

mow(0,10);

}

if(digitalRead(12)==0)

{

mow(10,0);

}

if(digitalRead(10)==0)

{

mow(10,0);

}

if(j==0)

{

count++;

if(count==12)

{

count=0;

penServo.write(penZUp);

mow(0,1000);

mow(-posx,0);

}

if(inChar=='\*')

{

mow(0,100);

}

if(inChar=='#')

mow(0,100);

}

if(inChar=='$')

{

mow(100,0);

}

if(inChar=='@')

{

mow(100,0);

}

if((inChar=='A')||(inChar=='a'))

{

penServo.write(penZDown);

mow(0,500);

mow(50,50);

mow(100,0);

mow(50,50);

mow(0,500);

}

if((inChar=='B')||(inChar=='b'))

{

penServo.write(penZDown);

mow(0,538);

mow(150,0);

mow(50,50);

mow(0,169);

mow(0,169);

mow(-50,50);

}

if((inChar=='C')||(inChar=='c'))

{

mow(300,0);

penServo.write(penZDown);

mow(150,0);

mow(50,50);

mow(0,438);

}

if((inChar=='D')||(inChar=='d'))

{

penServo.write(penZDown);

mow(0,-538);

mow(150,0);

}

if((inChar=='E')||(inChar=='e'))

{

penServo.write(penZDown);

mow(0,-538);

mow(150,0);

penServo.write(penZUp);

mow(100,0);

penServo.write(penZUp);

mow(-100,0);

}

if((inChar=='F')||(inChar=='f')){

mow(-150,0);

mow(0,269);

penServo.write(penZDown);

mow(100,0);

penServo.write(penZUp);

}

if((inChar=='G')||(inChar=='g'))

{

mow(0,-538);

mow(200,0);

mow(50,50);

penServo.write(penZDown);

mow(-50,-50);

}

if((inChar=='H')||(inChar=='h'))

{

penServo.write(penZDown);

mow(0,-538);

mow(250,0);

penServo.write(penZUp);

mow(0,-279);

penServo.write(penZDown);

}

if((inChar=='I')||(inChar=='i'))

{

penServo.write(penZUp);

mow(0,-538);

penServo.write(penZDown);

mow(250,0);

penServo.write(penZUp);

mow(-125,0);

penServo.write(penZDown);

mow(0,538);

}

if((inChar=='J')||(inChar=='j'))

{

penServo.write(penZUp);

mow(0,-538);

penServo.write(penZDown);

mow(250,0);

penServo.write(penZUp);

mow(-125,0);

penServo.write(penZDown);

}

if((inChar=='K')||(inChar=='k'))

{

penServo.write(penZDown);

mow(0,-538);

penServo.write(penZUp);

mow(0,279);

penServo.write(penZDown);

mow(279,-279);

}

if((inChar=='L')||(inChar=='l'))

{

penServo.write(penZUp);

mow(0,-538);

penServo.write(penZDown);

mow(0,538);

mow(250,0);

penServo.write(penZUp);

mow(125,0);

j=1;

}

if((inChar=='M')||(inChar=='m'))

{

penServo.write(penZDown);

mow(0,-538);

mow(200,379);

mow(200,-379);

mow(0,538);

penServo.write(penZUp);

mow(125,0);

j=1;

}

if((inChar=='N')||(inChar=='n'))

{

penServo.write(penZDown);

mow(0,-538);

mow(250,538);

mow(0,-538);

}

if((inChar=='O')||(inChar=='o'))

{

mow(0,-79);

penServo.write(penZDown);

mow(0,-380);

mow(100,-100);

mow(100,0);

mow(100,100);

}

if((inChar=='p')||(inChar=='P'))

{

penServo.write(penZDown);

mow(0,-536);

mow(250,0);

mow(50,50);

}

if((inChar=='Q')||(inChar=='q'))

{

mow(0,-79);

penServo.write(penZDown);

mow(0,-380);

mow(100,-100);

mow(200,0);

mow(100,100);

mow(0,380);

mow(-100,100);

}

if((inChar=='R')||(inChar=='r'))

{

penServo.write(penZDown);

mow(0,-538);

mow(150,0);

mow(50,50);

mow(0,169);

}

if((inChar=='S')||(inChar=='s'))

{

mow(0,-536);

mow(250,0);

penServo.write(penZDown);

mow(-250,0);

mow(0,279);

}

if((inChar=='T')||(inChar=='t'))

{

mow(0,-536);

mow(300,0);

penServo.write(penZDown);

mow(-300,0);

penServo.write(penZUp);

}

if((inChar=='U')||(inChar=='u'))

{

mow(100,0);

mow(100,-100);

mow(0,-462);

penServo.write(penZUp);

mow(0,536);

mow(125,0);

j=1;

}

if((inChar=='V')||(inChar=='v'))

{

mow(175,-538);

penServo.write(penZUp);

mow(0,538);

mow(125,0);

j=1;

}

if((inChar=='W')||(inChar=='w'))

{

mow(150,538);

mow(150,-538);

penServo.write(penZUp);

mow(0,538);

mow(125,0);

j=1;

}

if((inChar=='X')||(inChar=='x'))

{

penServo.write(penZDown);

mow(-250,538);

penServo.write(penZUp);

mow(375,0);

j=1;

}

if((inChar=='Y')||(inChar=='y'))

{

mow(0,-538);

penServo.write(penZDown);

mow(125,279);

penServo.write(penZUp);

mow(125,279);

}

if((inChar=='Z')||(inChar=='z'))

{

mow(0,-538);

penServo.write(penZDown);

mow(250,0);

mow(-250,538);

}

if(inChar==' ')

{

penServo.write(penZUp);

mow(250,0);

j=1;

}

if(inChar=='1')

{

penServo.write(penZDown);

mow(0,-538);

penServo.write(penZUp);

}

if(inChar=='2')

{

mow(0,-536);

penServo.write(penZDown);

mow(250,0);

mow(0,279);

}

if(inChar=='3')

{

mow(0,-536);

penServo.write(penZDown);

mow(250,0);

mow(0,279);

mow(-250,0);

penServo.write(penZUp);

}

if(inChar=='4')

{

mow(0,-536);

penServo.write(penZDown);

mow(0,400);

mow(250,0);

penServo.write(penZUp);

}

if(inChar=='5')

{

mow(0,-536);

mow(250,0);

penServo.write(penZDown);

mow(-250,0);

mow(0,379);

}

if(inChar=='6')

{

mow(0,-536);

penServo.write(penZDown);

mow(0,536);

mow(250,0);

}

if(inChar=='7')

{

mow(0,-536);

penServo.write(penZDown);

mow(250,0);

}

if(inChar=='8')

{

penServo.write(penZDown);

mow(0,-536);

mow(250,0);

mow(0,536);

mow(-250,0);

}

if(inChar=='9')

{

mow(250,0);

penServo.write(penZDown);

mow(0,-536);

mow(-250,0);

}

if(inChar=='0')

{

mow(0,-79);

penServo.write(penZDown);

mow(0,-380);

}

if(inChar=='=')

{

mow(0,-379);

penServo.write(penZDown);

mow(250,0);

penServo.write(penZUp);

mow(0,200);

j=1;

}

if(inChar=='.')

{

penServo.write(penZDown);

mow(10,0);

mow(0,-10);

mow(-10,0);

j=1;

}

if(inChar==',')

{

mow(0,-50);

mow(-50,0);

penServo.write(penZDown);

mow(50,50);

penServo.write(penZUp);

mow(100,0);

j=1;

}

}

}

**CHAPTER – 7**

**CONCLUSION**

**7.1 CONCLUSION AND FUTURE ENHANCEMENTS**

**CONCLUSION**

In this project, our approach is to discover a robot that can write a medical prescription what doctor says in mobile app, so that the patients and the pharmacists can understand what is written in the prescription. now the patients can take the correct medicine and the pharmacist can give the right medicine. we present a method to design rapid and fluid movements of a universal robot to perform robot writing mimicking the kinematics and trajectory of human handwritten signatures. The handwriting specimen acquisition, writing specimen processing, robot writing and comparison of on-line human and robot signatures are the phases of the experimental research performed.

**FUTURE ENHANCEMENT**

In our project the writing speed is slow since we use the servo motor. So, In order to increase the speed we can use the DSP (Digital Signal Processing) in future if needed. Even it can be used for handicapped people during their examination. The people who can’t write can use this instead of depending on some other people.

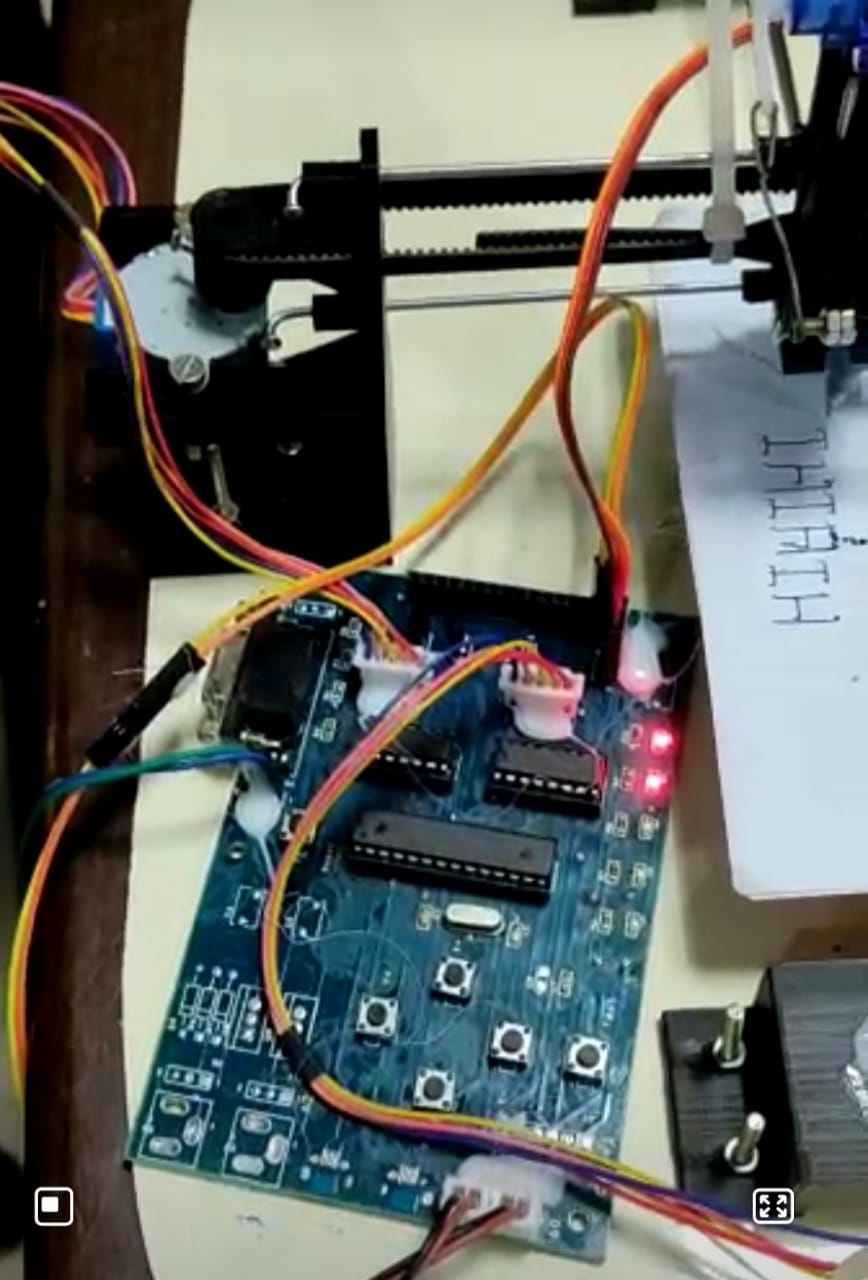
**APPENDICES**

**A.1 SAMPLE SCREENS**

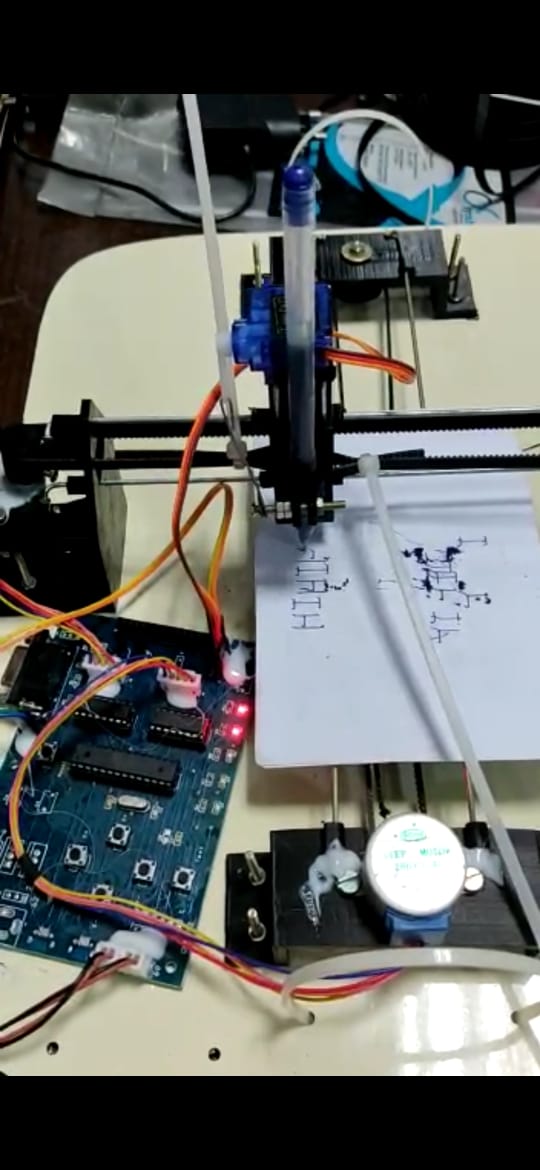
**A.1.1 VOICE RECOGNITION**



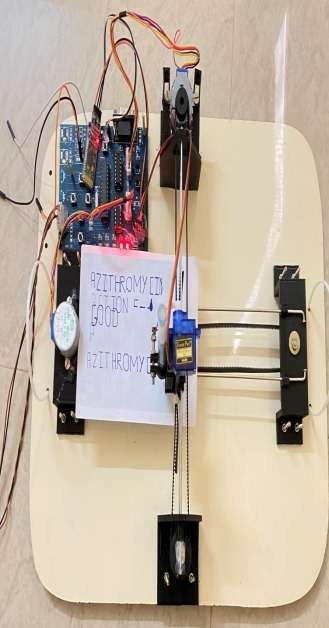
**A.1.2 CONTROL SWITCHES**



**A.1.3 PRESCRIPTION WRITING**

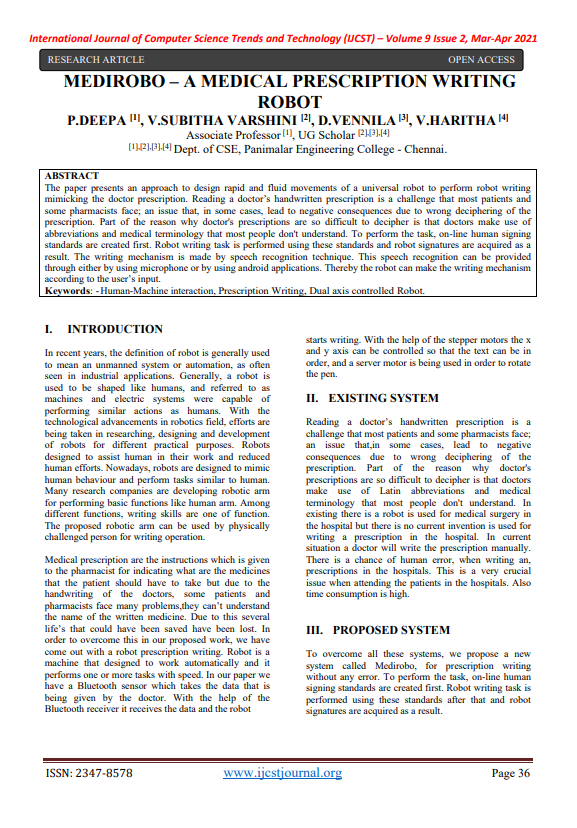


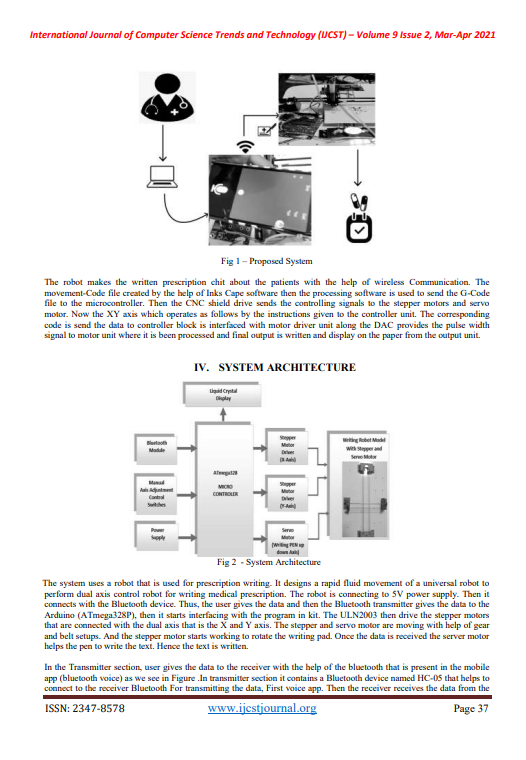
**A.1.4 FULL HARDWARE MODULE**

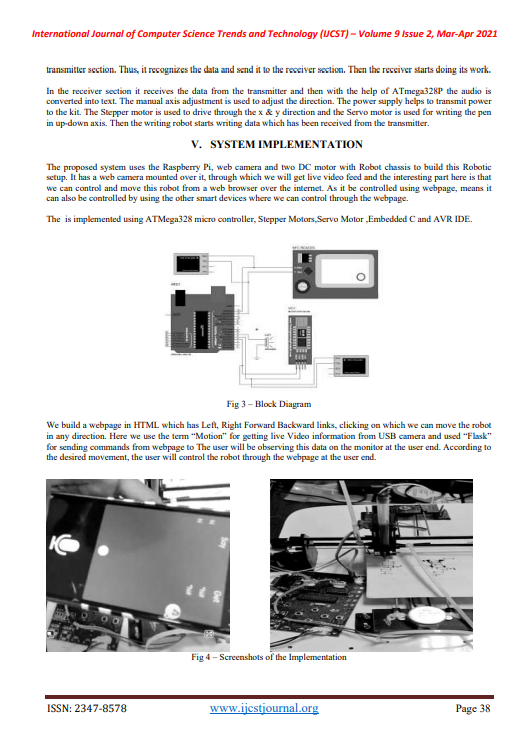


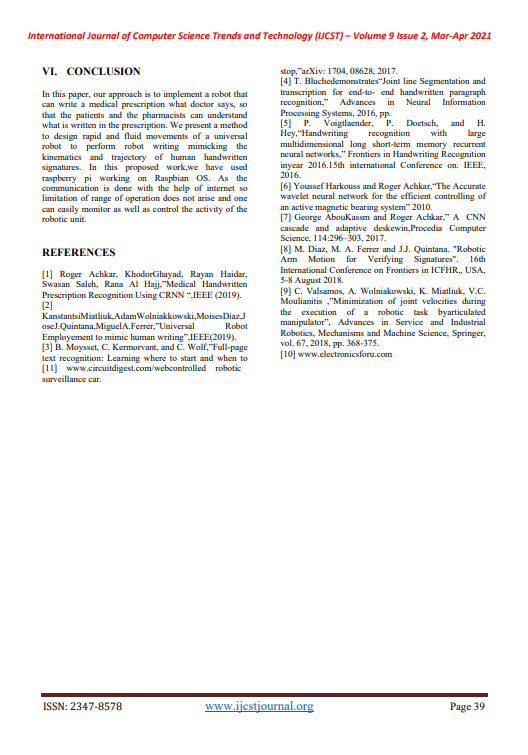
**A.2 PUBLICATIONS**

**A.2.1 PUBLISHED JOURNAL PAPER**









**A.2.2 PUBLISHED CERTIFICATES**







**REFERENCES**

[1] Roger Achkar, Khodor Ghayad, Rayan Haidar, Swasan Saleh, Rana Al Hajj,” Medical Handwritten Prescription Recognition Using CRNN “, IEEE (2019).

[2] Kanstantsi Miatliuk, Adam Wolniakkowski, Moises Diaz, Jose J.Quintana, Miguel A.Ferrer,” Universal Robot Employement to mimic human writing”, IEEE(2019).

[3] B. Moysset, C. Kermorvant, and C. Wolf proposes “Full-page text recognition: Learning where to start and when to stop,” arXiv: 1704, 08628, 2017.

[4] T. Bluche demonstrates “Joint line Segmentation and transcription for end-to- end handwritten paragraph recognition,” in Advances in Neural Information Processing Systems, 2016, pp.

[5] P. Voigtlaender, P. Doetsch, and H.Hey publish about “Handwriting recognition with large multidimensional long short-term memory recurrent neural networks,” in Frontiers in Handwriting Recognition in year 2016.15th international Conference on. IEEE, 2016.

[6] Youssef Harkouss and Roger Achkar have published a paper “The Accurate wavelet neural network for the efficient controlling of an active magnetic bearing system” 2010.

[7] George Abou Kassm and Roger Achkar demonstrates Apr cnn cascade and adaptive deskewing. Procedia Computer Science, 114:296–303, 2017.

[8] M. Diaz, M. A. Ferrer and J.J. Quintana. "Robotic Arm Motion for Verifying Signatures". 16th International Conference on Frontiers in ICFHR, USA, 5-8 August 2018.

[9]C.Valsamos, A.Wolniakowski, K.Miatliuk, V.C.Moulianitis proposed the Minimization of joint velocities during the execution of a robotic task by articulated manipulator, Advances in Service and Industrial Robotics, Mechanisms and Machine Science, Springer, vol. 67, 2018, pp. 368-375.

[10] Dr. Nomal Chandra Borah, Chairman & Managing Director published there paper in Guwahati Neurological Research Centre. Doctor’s illegible handwriting creating confusion leads to death on April 20, 2015.