A Real Time Research Project Report

On

MORSE CODE TRANSLATOR USING PYTHON

Submitted by,

ADURI GOPI CHANDRA REDDY - 22J41A0586
KOTHAVADLA SOHAN - 22J41A0599
GUGULOTH MAHESHWARI - 23J45AO507
PENDEM SHIVA SAGAR - 22J41A05B8

in partial fulfilment of the academic requirements II BTech.

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Under the Guidance of

MS. B. SRAVANI

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MALLA REDDY ENGINEERING COLLEGE(A)
Maisammaguda, Secunderabad, Telangana, India 500100

July -2024

MALLA REDDY ENGINEERING COLLEGE

Misammaguda , Secunderabad, Telangana, India 500100

BONAFIDE CERTIFICATE

This is to certify that this **Real Time Research Project** work entitled "**MORSE CODE TRANSLATOR USING PYTHON**", submitted by **ADURI GOPI CHANDRA REDDY (22J41A0586), KOTHAVADLA SOHAN (22J41A0599), GUGULOTH MAHESHWARI (23J45AO507), PENDEM SHIVA SAGAR (22J41A05B8)** to Malla Reddy Engineering College affiliated to **JNTUH, Hyderabad** in academic requirements II BTech for the award of **Bachelor of Technology** in **Computer Science and Engineering** is a bonafide record of project work carried out under my supervision during the academic year 2023– 2024 and that this work has not been submitted elsewhere for a degree.

SIGNATURE

MS. B. SRAVANI

Assistant Professor

Department of CSE

SIGNATURE

DR.P.S.R.C MURTHY

Head of the Department

Department of CSE

Malla Reddy Engineering College Malla Reddy Engineering College
Secunderabad, 500100

Submitted for Real Time Research Project viva-voce examination held on _____

INTERNAL EXAMINAR

EXTERNAL EXAMINER

TABLE OF CONTENTS

ABSTRACT

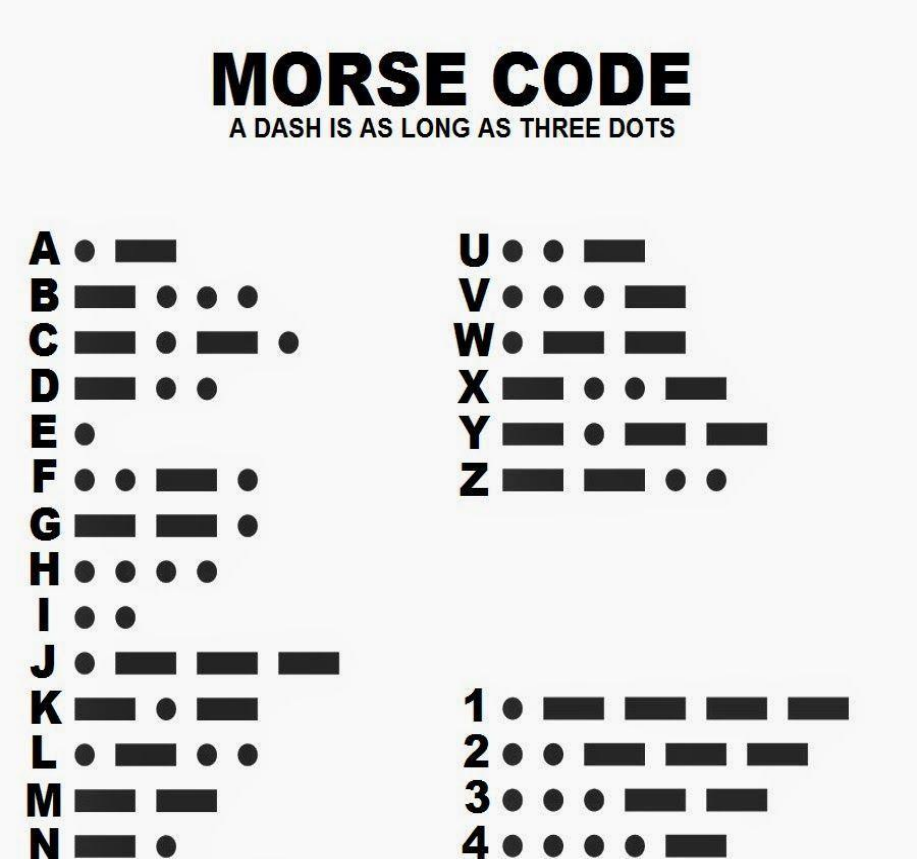
This project aims to design and develop a Morse Code Translator, a software application that enables real-time communication between individuals using Morse code and plain text. The translator will convert Morse code sequences into corresponding text messages and vice versa, facilitating efficient and accurate communication. The application will feature a user-friendly interface, allowing users to input Morse code sequences or text messages and receive instant translations. The Morse Code Translator will be developed using [programming language/technology] and will incorporate a Morse code dictionary and translation algorithm to ensure accurate and efficient translations. The application will also include error handling mechanisms to address invalid input or translation errors. The project's objectives include: Designing and developing a user-friendly Morse Code Translator application Implementing a efficient translation algorithm for Morse code to text and vice versa Ensuring accuracy and reliability in translations Providing a real-time communication tool for individuals using Morse code.The Morse Code Translator has potential applications in various fields, including emergency communication, navigation, and education. This project aims to contribute to the development of innovative communication tools and promote the use of Morse code in modern communication systems.

Note: This abstract is a sample, please make sure to adjust it according to your project's specific goals, objectives and technologies used.

CHAPTER 1

INTRODUCTION

In this Project, we point to create an instinctive however capable application outlined to encode and translate incognito Morse code messages without the need of any tapping device. The application will highlight a user-friendly graphical interface made utilizing Python, permitting clients to easily encode content into Morse code, translate Morse code messages, and indeed tune in to the Morse code through a committed "Play Sound" button.



Morse code translator apps are incredibly useful tools for individuals who require Morse code communication. These apps efficiently convert the dots and dashes of Morse code into readable text and vice versa, facilitating seamless communication. Whether you're a Morse code learner,

looking to practice, or even seeking a reliable means of emergency communication, these apps cater to a wide range of needs. With their user-friendly interfaces and versatile functionality, Morse code translator apps have become an indispensable resource in the modern world. They not only provide a convenient way to translate and communicate in Morse code but also offer additional features such as customizable speed settings, historical message logs, and even interactive learning modules to enhance your Morse code skills.

Imagine being able to effortlessly communicate with others using this timeless and fascinating form of communication, whether it's for fun, educational purposes, or even in emergency situations where traditional means of communication may not be available. Morse code translator apps empower individuals to tap into this rich history and engage in meaningful connections with others who share an appreciation for this unique language.

Objectives :

A. Allow paralysis victims to communicate independently:

Many paralysis victims already use eye blinks as a form of communication. It is common for nurses and caretakers to read a patients eye blinks and decode the pattern. The ALS association even offers a communication guide that relies on eye blinks. Blink To Speak automates this task. The software reads a persons eye blinks and converts them into text. A key feature of the software is that it can be started, paused, and operated entirely with eye blinks. This allows patients to record their thoughts with complete independence. No nurses or caretakers are required to help patients express themselves. Not only does this reduce the financial burden on paralysis patients, but this form of independence can be morally uplifting as well.

B. Be accessible to people with financial constraints:

Many companies are developing technologies that are controlled by eye movement. These technologies rely on expensive hardware to track a user`s eyes. While these devices can absolutely help LIS victims, they are only available to people.

CHAPTER 2
LITERATURE SURVEY

Sourya Dey, Keith M. Chugg and Peter A. Beerel presented a paper on Morse code datasets for machine learning [1].

They have presented an algorithm to generate synthetic datasets of tenable difficulty on the classification of morse code symbols for supervised machine learning problems. They completely used machine learning for this. They have done this process in four steps frame portioning, assigning values for intensity levels, noising, mass generation. Paparao Nalajala Bhavana Godavarth, M Lakshmi Raviteja, Deepthi Simhadri presented a paper on Morse code generator using Microcontroller using Alphanumeric keyboard [2]. They presented the paper using radiotelegraphy .they used keypads etc., the clock signal is generated by matrix keypad.it uses an LCD, led, buzzer, most importantly transmitter.it uses a microcontroller for processing the morse code to display on the LCD screen. Dr A. Murugan, R. Thilagavathy presented a paper on Cloud Storage Security Scheme using DNA Computing with Morse code and Zigzag Pattern [3].

Storage and exchange of data are done in cloud computing, DNA computing, zigzag chippers. It uses a DNA of an organism to process an input and by computing this they take initially a stream of characters and converted to binary data, to DNA sequence and then to morse code and then to zigzag pattern the resultant is the chipper text. Zhimeng Yin, Wenchao Jiang' Song Min Kim, Tian He presented a paper on C-Morse: Crosstechnology Communication with Transparent Morse.Coding [4].it uses a Wi-Fi to zig bee and viceversa it uses a transport protocol like TCP, UDP. degredations are very low nearly 4%. The Wi-Fi___33 sender modulates its symbols by only perturbing the transmission timing of through data packets and control frames within a negligible delay. In this way, C-Morse constructs the statistically recognizable energy patterns in the ISM band in a transparent way. On top of the modulator,

CMorse has its multiple access control, and a packet buffer to opportunistically utilize the existing data packets, avoiding the need for generating dummy packets. maintaining transparency. With 100 packets/s, CMorse achieves a throughput of 12bps and manages to boost its CTC throughput to 137bps at 800 packets/s.

CHAPTER 3
SYSTEM ANALYSIS

3.1 Existing System

1. Input Mechanism

Text Input: Users can type the text they want to convert to Morse code.

Audio Input: Users can speak or play Morse code sounds, which the system then converts to text.

Visual Input: Users can input Morse code via a graphical interface or camera, often used in mobile apps.

2. Processing Unit

Encoding Algorithm: Converts alphanumeric characters into corresponding Morse code symbols (dots and dashes).

Decoding Algorithm: Converts Morse code back into alphanumeric characters.

Error Handling: Identifies and corrects errors in the input, ensuring accurate translation.

3. Output Mechanism

Text Output: Displays the translated Morse code or plain text.

Audio Output: Plays the Morse code as beeps or tones.

Visual Output: Shows the Morse code using visual signals (like blinking lights).

4. User Interface

Web-Based Interfaces: Online tools where users can input text and get Morse code output, or vice versa.

Mobile Applications: Apps that provide text-to-Morse and Morse-to-text translations, often with additional features like training tools.

Desktop Software: Standalone programs for translating Morse code.

5. Additional Features

Real-time Translation: Some systems offer real-time translation for input via typing, audio, or visual signals.

Learning Mode: Tools and apps that help users learn and practice Morse code.

Customization: Allows users to adjust the speed of audio output or the appearance of visual signals.

3.2 Existing System Disadvantages

Limited Accuracy: Existing Morse code translators might struggle with accurately translating messages, especially if the input is noisy or has timing variations.

User Interface Issues: Many existing systems have outdated or non-intuitive user interfaces, making them difficult to use for beginners.

Lack of Support for Audio Input: Some translators do not support audio input, limiting their functionality to text-based Morse code.

Limited Language Support: Existing translators often support only English, neglecting Morse code adaptations for other languages or characters.

No Real-time Translation: Many systems do not offer real-time translation, which is crucial for live communication.

Dependency on Internet Connection: Some Morse code translators are web-based and require an internet connection, which can be a limitation in areas with poor connectivity.

Insufficient Educational Tools: Many translators lack integrated educational tools or tutorials, making it difficult for users to learn Morse code effectively.

3.3 Proposed System

1. **Enhanced Input Mechanism**

- Text Input: A user-friendly interface for typing text to be converted to Morse code.
- Audio Input: Enhanced speech-to-text technology for converting spoken language to text, which is then translated to Morse code.
- Visual Input: Optical character recognition (OCR) for recognizing handwritten or printed Morse code. Additionally, a camera-based input to detect and translate blinking lights or hand signals.

2. **Advanced Processing Unit**

- Encoding and Decoding Algorithms:
    - Robust algorithms for accurate and fast conversions between text and Morse code.
    - Machine learning models to improve the accuracy of audio and visual inputs.
- Error Handling and Correction:
    - Advanced error detection mechanisms using machine learning to identify and correct errors in Morse code input.
    - Contextual error correction to predict the most likely correct translations.

3. Versatile Output Mechanism

- Text Output: Clear and customizable text display for translated Morse code.
- Audio Output: High-quality audio signals with adjustable speed and tone for learning and communication purposes.
- Visual Output:
    - Blinking light signals or visual representations of Morse code.
    - Augmented reality (AR) displays for immersive learning experiences.

4. User-Friendly Interface

- Web-Based Platform: A responsive web interface accessible on all devices.
- Mobile Application:
    - Intuitive and feature-rich apps for both Android and iOS.
    - Integration with wearable devices for real-time Morse code translation.
- Desktop Software: Comprehensive software with additional tools for advanced users, such as amateur radio enthusiasts.

5. Innovative Features

- Real-Time Translation: Instantaneous conversion for all input types with real-time feedback.
- Learning and Practice Mode:
    - Interactive tutorials and quizzes to help users learn Morse code.
    - Gamification elements to make learning engaging and fun.

3.4 Proposed System Advantages

- **Accessibility**: The system's diverse input and output methods make it accessible to a wide range of users, including those with different preferences and needs.

- **Accuracy and Reliability**: Advanced algorithms and error-handling mechanisms ensure accurate translations, making the system reliable for both casual use and serious applications.

- **User Engagement**: Innovative features like AR displays, gamified learning, and real-time feedback keep users engaged and motivated to learn Morse code.

- **Global Reach**: Multi-language support and integration with communication tools make the system valuable to users worldwide, expanding its potential user base.

- **Trust and Security**: Strong security measures and privacy controls build user trust, making the system a safe choice for all users.

CHAPTER 4

SYSTEM DESIGN

4.1 System Architecture

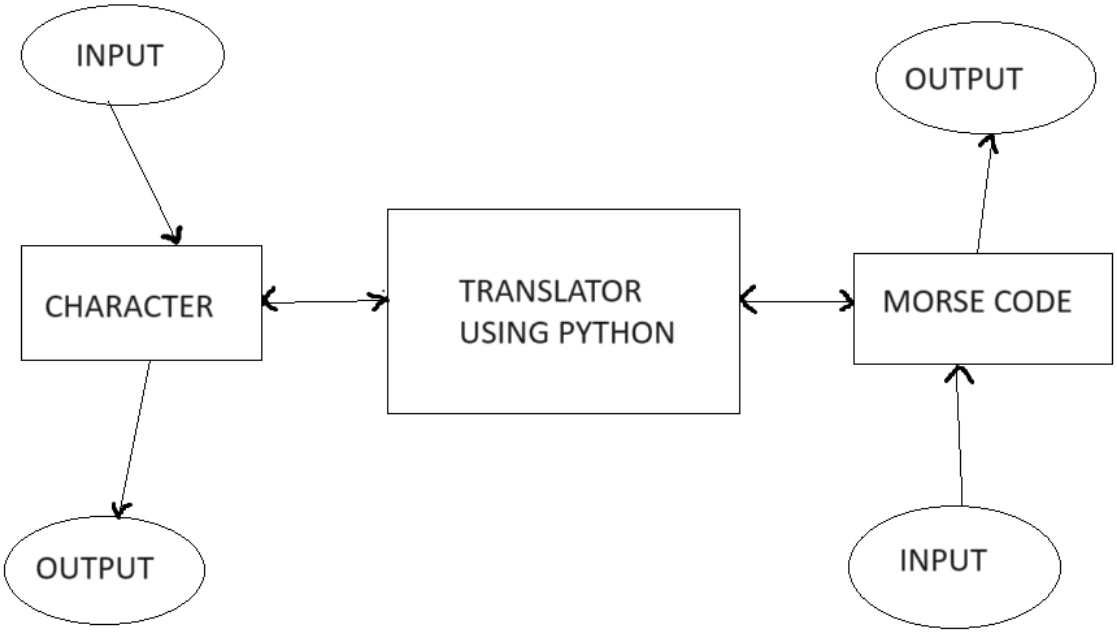


Fig:4.1 Morse Code Translator

During this process of translation from the above diagram, we can say that when a character is given as input by translating it we get a morse code from the hashmaps as output. When Morse code is given as input then by translating it we get a character as output. The use of the hashmaps is most predominant in python for accessing the data structure .so every key of the character has its value in the database we need to access through a function called itertools.

Because we choose python as it is efficient in both fast and accessing a database. Cipher stores the morse translated the form of the English string. Decipher stores the English translated form of the morse string. Context stores morse code of a single character. It keeps count of the spaces between morse characters.

ER-Diagram

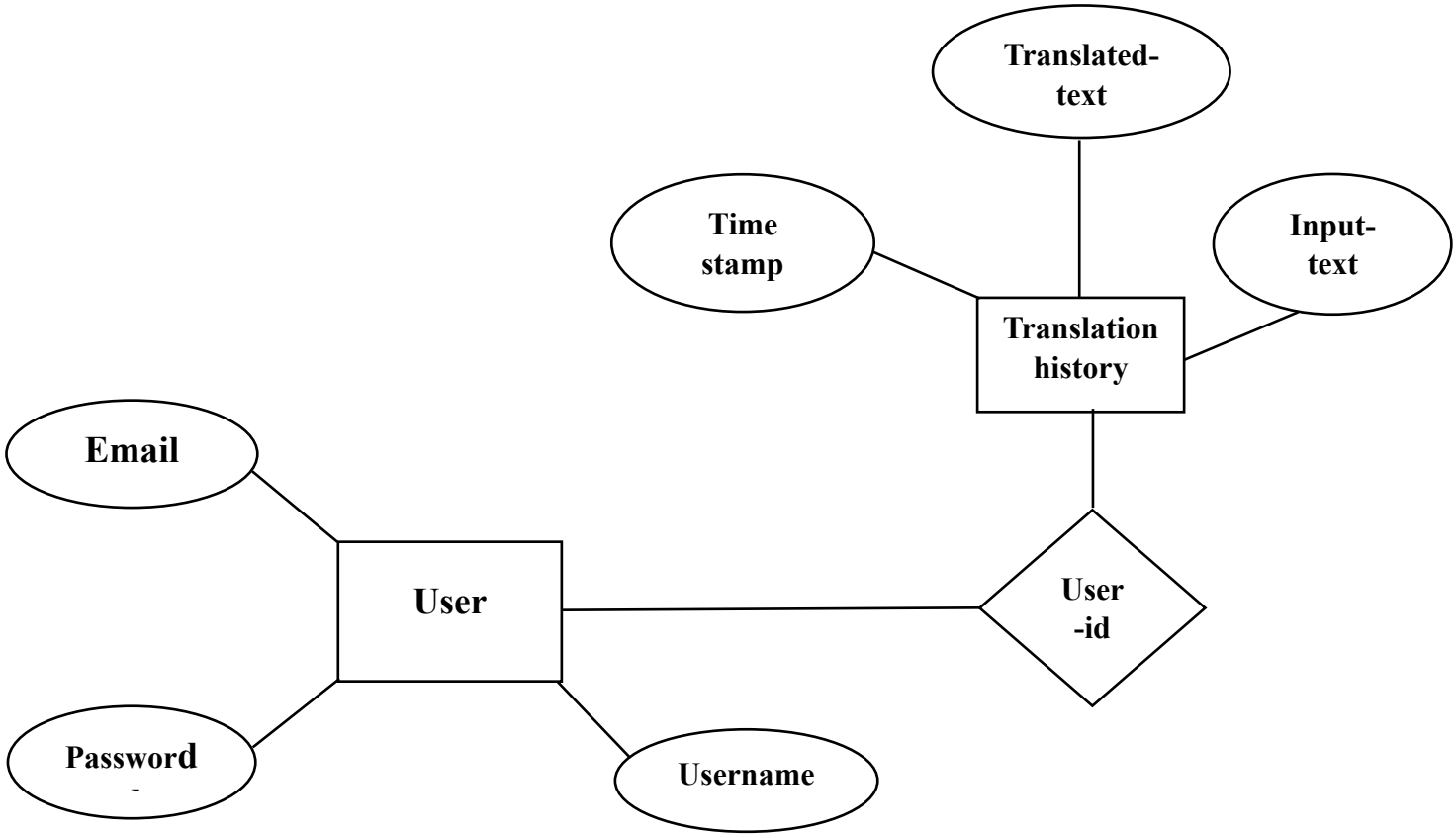


Fig:4.2

Block Diagram

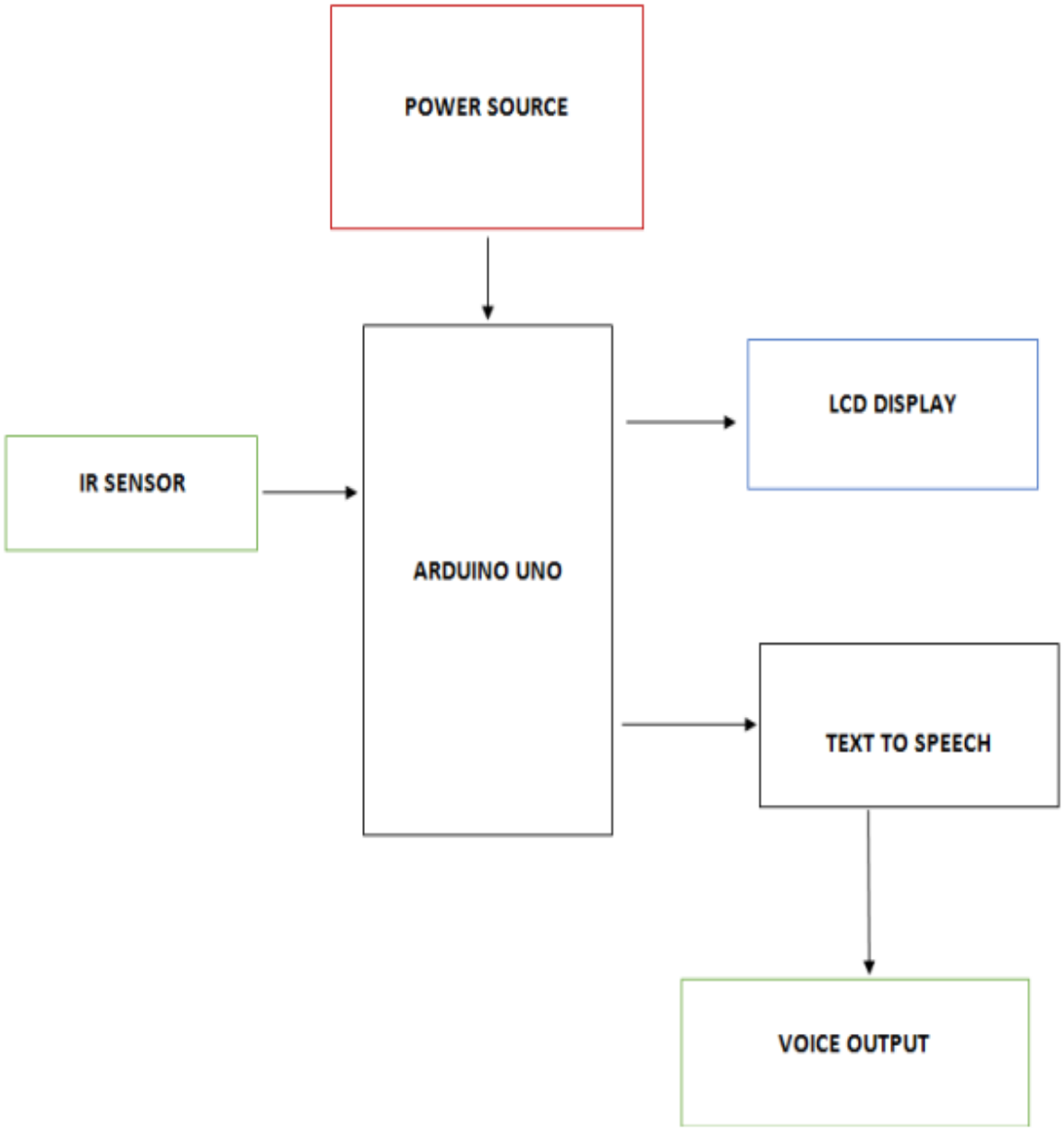


Fig:4.3

Flow chart

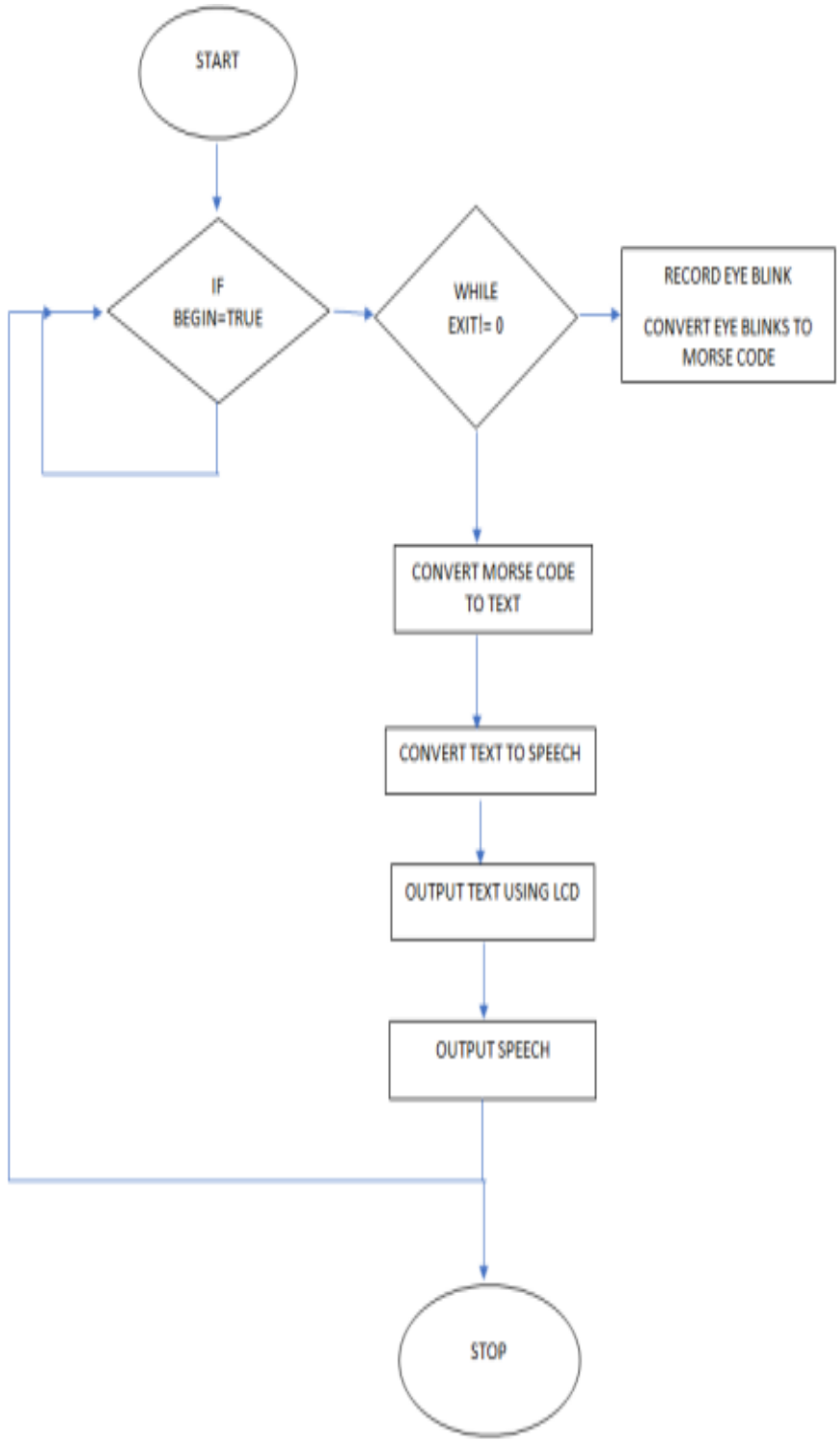


Fig:4.4

CHAPTER 5
IMPLEMENTATION

Python provides a data structure called dictionary which stores information in the form of key-value pairs which is very convenient for implementing a cipher such as a Morse code. We can save the morse code chart in a dictionary where (key -value pairs) => (English character – morse code). The plaintext (English characters) take the place of keys and the ciphertext (Morse code) form the values of the corresponding keys. The values of keys can be accessed from the dictionary in the same way we access the values of an array through their index and vice versa.

We know that we have hashmaps in python which is a form of data used to find the frequency of characters in a string. () Basically a dictionary is like a list instead of integer index it can be of any type.to define a dictionary we use a key value pair with a colon between them iter () method returns an iterator for the given object .in case of sentinel provided, it returns the iterator object that calls the callable object until the sentinel character isn't found. We need to import the dictionary by calling the function MORSE_CODE_DICT (). We can use any of the python compilers like python 3.7(32bit). or any other online python compilers .so for fast output we can use online python compilers.

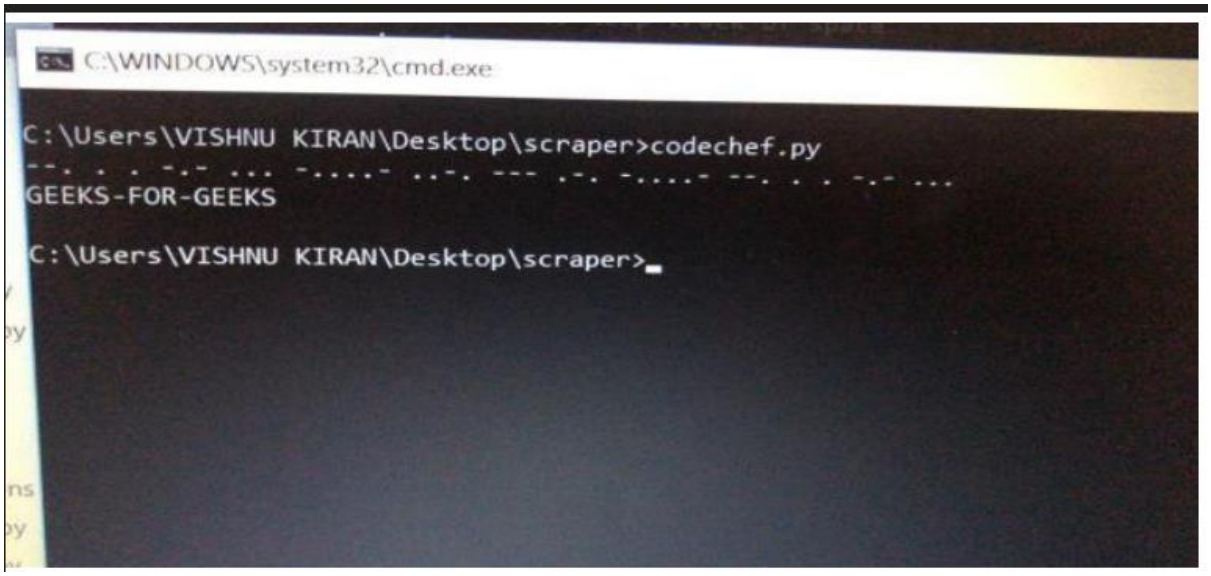


Fig:5.1 Morse Code to Character

5.1 MODULES

- **dictionary.py :-** This module consists of a morse code dictionary having keys and values for the conversion.

- **encryption.py :-** This module consists of a function which is used to convert normal text into morse code.

- **decryption.py :-** This module consists of a function which is used to convert morse code into plain text.

- **gui.py :-** This is the module where you can run the code with its simple GUI interface.

- **gui_support.py :-** This is module with try and except for importing the right library and for displaying GUI
- **run.py :-** This module deals with running of code with GUI. It is terminal based for the sake of simplicity.

CHAPTER 6
SYSTEM REQUIREMENTS

6.1 Hardware Requirements

1. Computer or Device: A computer, laptop, tablet, or smartphone with a compatible operating system (Windows, macOS, Linux, Android, or iOS).

2. Input Device: A keyboard or a button/input device to input Morse code signals (dots and dashes).

3. Output Device: A display screen or speaker to output the translated text or audio.

4. Optional: A radio or communication device to receive Morse code signals (for a more advanced project).

6.2 Software Requirements

1**. Programming Language:** A programming language like Python, Java, C++, or JavaScript to develop the translator software.

2. Development Environment: An Integrated Development Environment (IDE) like PyCharm, Eclipse, or Visual Studio Code to write, debug, and test the code.

3. Morse Code Library: A library or module that provides Morse code encoding and decoding functionality (e.g., Python's morse library).

4. User Interface: A graphical user interface (GUI) library like Tkinter, PyQt, or React to create a user-friendly interface for input and output.

5. Audio Library: An audio library like PyAudio or Web Audio API to generate audio output (if desired).

6. Operating System: A compatible operating system (Windows, macOS, Linux, Android, or iOS) to run the translator software.

CHAPTER 7

EXPECTED OUTPUT

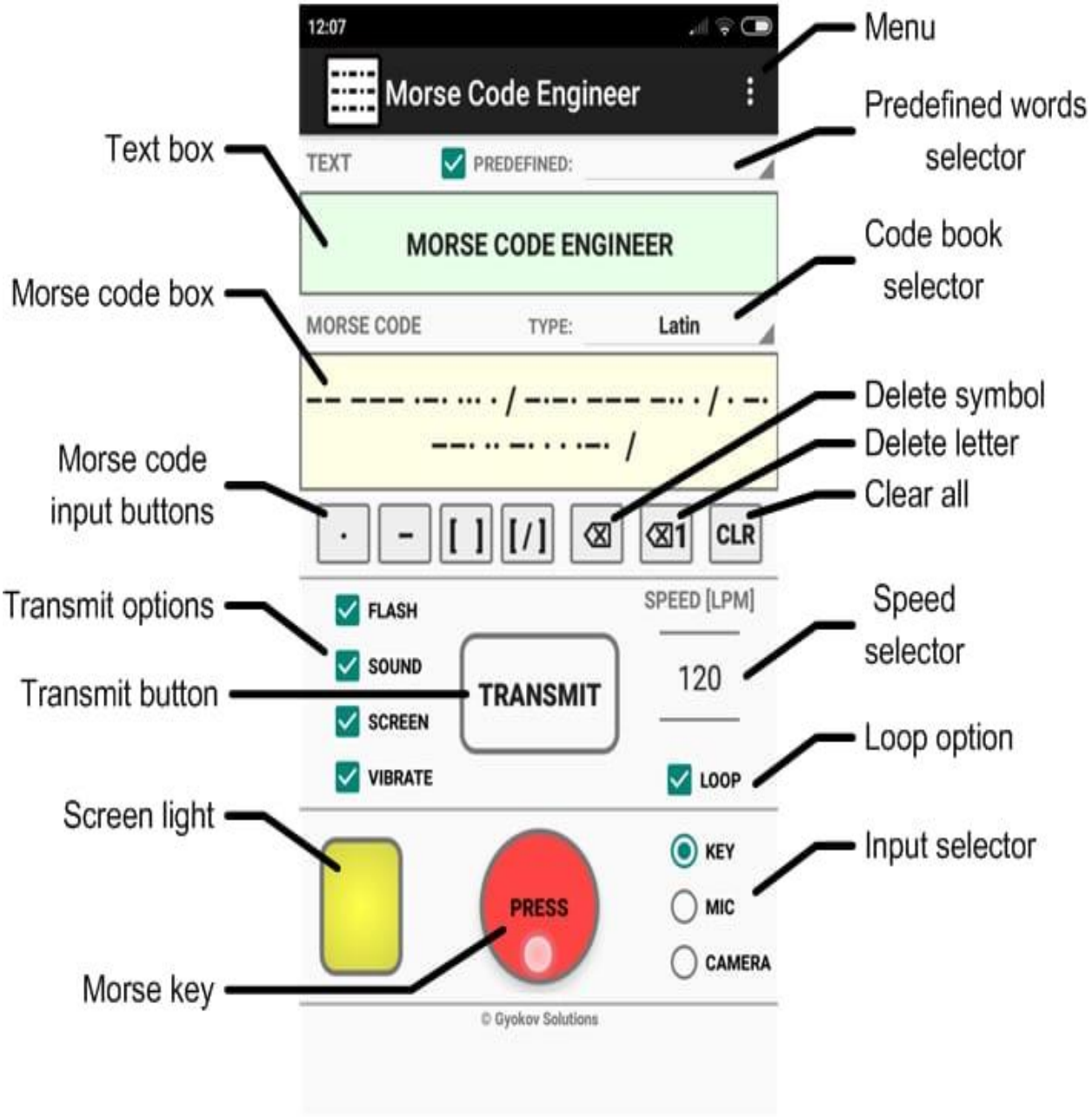


Fig:7.1 The translation of text to Morse code and vice versa, as well as error handling for invalid input**.**

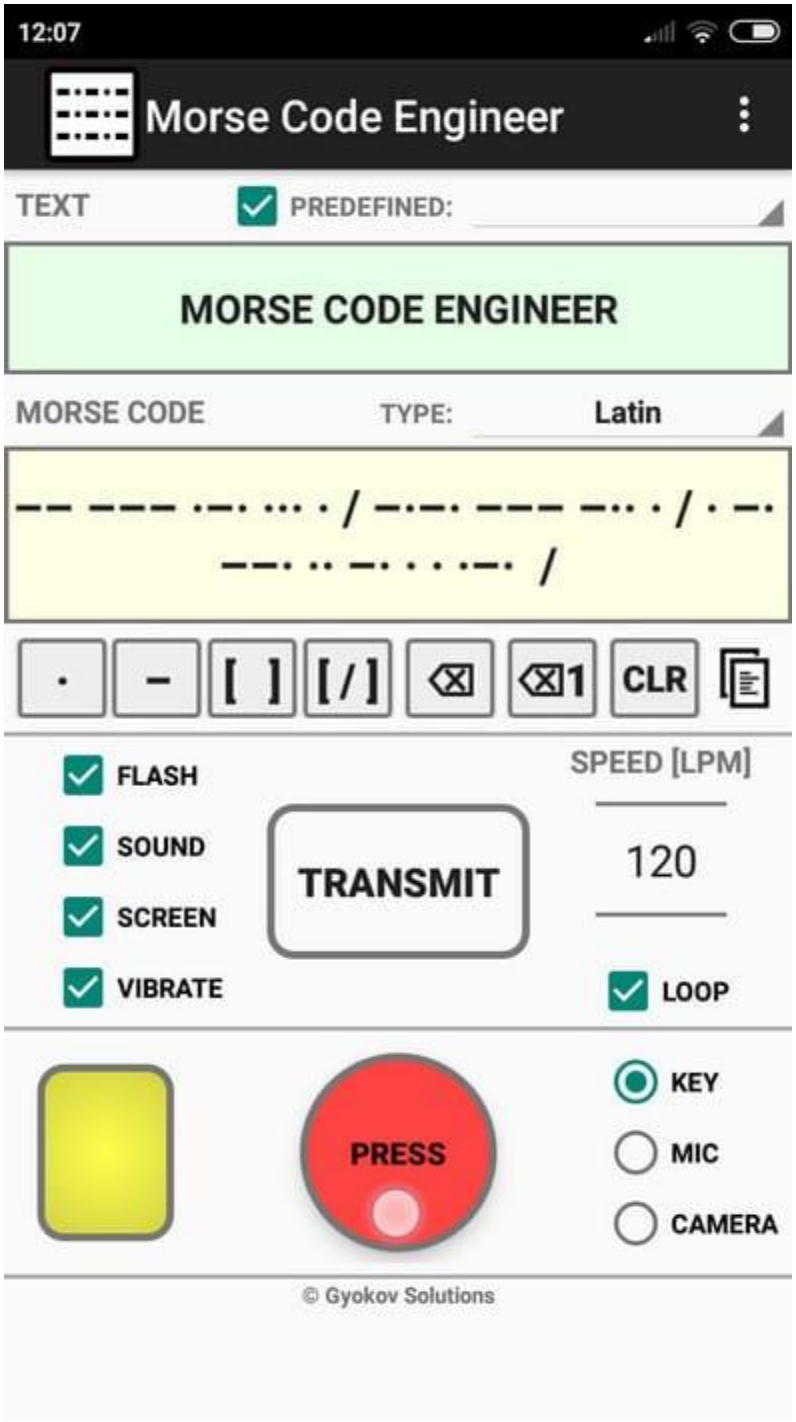


Fig:7.2 Alternative symbols for dot, dash, word delimiter and space can be used.

Future Scope

1. Advanced Audio Processing

- Improved Noise Reduction: Enhance algorithms to better filter out background noise and improve the accuracy of audio translations.
- Real-time Audio Processing: Develop capabilities for real-time translation of live audio signals, enabling seamless communication.

2. Machine Learning Integration

- Adaptive Learning Models: Use machine learning to improve translation accuracy by learning from user inputs and correcting errors over time.
- Pattern Recognition: Implement advanced pattern recognition to distinguish between closely spaced signals and varied Morse code speeds.

3. Multi-language Support

- International Morse Code Variants: Expand support to include different Morse code variants used globally, accommodating non-English languages and special characters.

4. Mobile and Wearable Device Integration

- Mobile Apps: Develop mobile applications for iOS and Android, allowing users to translate Morse code on the go.
- Wearable Technology: Integrate with smartwatches or other wearable devices to provide vibration-based Morse code alerts and inputs.

5. IoT and Smart Home Integration

- Home Automation: Integrate with smart home devices to enable Morse code-based commands for home automation systems.
- IoT Devices: Use Morse code as a communication protocol for Internet of Things (IoT) devices, particularly in low-power or limited-bandwidth scenarios.

CHAPTER 8
CONCLUSION

To conclude, we can say that creating a Morse code app is a challenging task, but if you are well versed in ‘How to develop Morse code translator app?’ you can excel in the development like a pro. Following all the steps mentioned in this blog above will help you develop a successful business application that will meet the users’ needs. While the demand to create a mobile app to convert Morse code to text isn’t that high, the application comes in handy in many cases. So, by working with a top mobile application development company like Dev Technosys, you can remain assured of getting a revenue-generating mobile app. Hire dedicated developers, and the experts will help you craft a perfect application.

CHAPTER 9

REFERENCES

1. Wikipedia : https://en.wikipedia.org/wiki/Morse_code

2. MorseCode.World: https://morsecode.world/international/translator.html

3. IEEE Xplore : httpss://ieeexplore.ieSee.org/document/7556055