Morse Code Converter Using Arduino for Real-Time Communication

Saksham Thakur, Gurrapu Pruthvik Sai
Maturi Venkata Subba Rao (MVSR) Engineering College (A) ,
Nadergul(P.O.) ,Telangana ,India.

Abstract: Morse code is a simple communication method which encodes letters and numbers using dashes and dots. Below is the creation and construction of a Morse code converter from an Arduino microcontroller, an input push button, an LED and buzzer for display and audio output, and the Arduino IDE Serial Monitor to output the decoded message. The system's code is designed for modification, and users are allowed to customize timing and cue feedback. The open nature allows for it to suit various learning speeds and sensory preferences. Besides that, the project encourages open-source design to support community contribution and development. The system provides an easy and cheap solution for converting Morse code, and hence, it is suited for educational and assistive communication purposes. Its minimal power consumption and mobility also make it suitable for field demonstrations and emergency use.

Keywords: Arduino Morse Code Converter, Real-time Processing, Visual Feedback, Assistive Technology.

I. INTRODUCTION

Morse code has also been a very important form of communication since the 19th century, more so in telegraphy and emergency signaling. Morse code is applicable today in the use of microcontrollers for educational and assistive purposes. This project will illustrate the implementation of a Morse code converter using an Arduino, which will allow users to input Morse signals through a push button, obtain real-time output through an LED and buzzer, and display the decoded message on the Serial Monitor. This project is more than mere translation; it is a real-world demonstration of embedded systems and signal processing. Also, integration of wireless communication modules, i.e., Bluetooth or Wi-Fi, might make it possible to transmit and receive Morse code remotely. Next-generation devices may be designed to incorporate voice synthesis so that the device speaks the messages after they have been decoded, making them more accessible to blind people. Also, a display screen might be added to the system to dispense with the requirement for an attached computer. The project also seeks to offer a modular design, which will allow users to easily modify the system to accommodate various input and output means. By opening up the project, it promotes collaboration and further development, which will ensure that Morse code remains relevant in the digital era.

II. LITERATURE REVIEW

Morse code, an artifact of 19th-century telegraphy, has made a comeback in the 21st century, particularly with Arduino-based converters. Although no longer used for large-scale communication, its use persists in niche fields like education, accessibility, and embedded systems. This discussion is focused on applications of Arduino-based Morse code converters, highlighting their relevance in today's times. Despite its old-fashioned reputation, Morse code is still applied in assistive technology as a method of

ISSN [Online]: 2583-2654

communication for those with speech or motor disabilities, typically facilitated by text-to-speech (TTS) engines. Arduino microcontrollers are also being widely used to construct such conversion systems due to their affordability, simplicity, and versatility.

These systems read in, translate, and convert Morse code input to text or speech, and the Arduino open-source platform can be continuously enhanced and tuned. Arduino Morse code programs are used in interactive learning of STEM courses by students to demonstrate embedded systems, signal processing, and hardware-software integration while introducing them to the basic elements of communication protocols, microcontroller programming, and HCI.

One of the major uses is in accessibility, and the converters offer a means for disabled people to gain communication so that the system becomes more flexible and usable with HCI-based interfaces so that maximum efficiency in communication and independence can be attained. The operation is further enriched with the addition of wireless components such as Bluetooth, Wi-Fi, and RF transceivers to facilitate long-distance communications where other normal practices do not work, for example, emergency rescue from disasters and emergency signaling. Visual feedback is important with LCD or LED display delivering instant visual feedback of input Morse code for improved accuracy and usability, while others utilize graphical user interfaces (GUIs) for more interactive use. Current research targets enhancing input devices, noise interference elimination, and enhancing usability in real life by using sophisticated signal processing and machine learning with the objective of developing adaptive systems to enhance accessibility and usability for various users.

Finally, Arduino-based Morse code translators bridge a historic method of communication to contemporary usage, most notably assistive technology, STEM learning, and wireless applications, with further development on the way toward even greater efficiency and usability, re-affirming the sustainability of Morse code as an operational method of communication today.

This literature review provides the current trends and context for the discussions concerning the design, implementation, and real-world application of Arduino-based Morse code converters.

2.1 Hardware Components:

Arduino Uno: It processes input and produces output.
Push Button: It acts as a user input, detects short and long pushes corresponding to dots and dashes.
LED & Buzzer: It gives a visual and audio feedback towards Morse code signals.
Resistors & Connecting Wires: These are used for proper connections in the circuit.

2.2 *Software Components:*

Arduino IDE: This is used to program and monitor the Morse code translation.
Embedded C/C++: The programming language for developing the Morse code logic.

2.3 Embedded C/C++ Code:

```
// Morse Code Decoder/Encoder
const int piezoPin = 2, ledPin = 13, buttonPin = 8, debounce = 30;
 int dotDuration = 240, dashDuration = dotDuration * 3, letterPause = dotDuration * 3, wordPause = dotDuration * 7;
float wpm = 1200.0 / dotDuration;
const char PROGMEM morseChars[] = {'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z', '0', '1', '2', '3', '4', '5', '6', '7', '8', '9'};
const char *PROGMEM morseCodes[] = {".-", "-...", "-.-.", "-..", ".", "..-.", "--.", "....", "..", ".---", "-.-", ".-..", "--", "-.", "---", ".--.", "--.-", ".-.", "...", "-", "..-", "...-", ".--", "-..-", "-.--", "--..", "-
```

```
----", ".----", "..---", "...--", "....-", ".....", "-....",
"--...", "---..", "----."};
String currentSequence = "";
bool newCharacter = false, newWord = false,
readingKeyboard = false;

void setup()
{
pinMode(ledPin, OUTPUT);
pinMode(piezoPin, OUTPUT);
pinMode(buttonPin, INPUT_PULLUP);
Serial.begin(9600); Serial.println("Morse Code
Decoder/Encoder\nSpeed: " + String(wpm) + "
wpm, Dot: " + String(dotDuration) + "
ms\nEnter text or Key in Morse:\n---------------
-----------------");
tone(piezoPin, 1000); digitalWrite(ledPin,
HIGH); delay(2000); digitalWrite(ledPin,
LOW); noTone(piezoPin); delay(500);
}

void loop()
{
if (Serial.available() > 0)
{
if (!readingKeyboard) Serial.println("\n---------
----------------------");
readingKeyboard = true;
char inputChar = toupper(Serial.read());
 if (inputChar >= 'A' && inputChar <= 'Z' ||
inputChar >= '0' && inputChar <= '9') {
Serial.print(inputChar);
Serial.print(" "); printMorse(inputChar);
 }
else if (inputChar == ' ') { Serial.println("_");
delay(wordPause);
}
if (Serial.available() <= 0) { Serial.println("\n--
------------------------------\nEnter text or Key in
Morse:\n--------------------------------");
readingKeyboard = false;
} }

if (digitalRead(buttonPin) == LOW) {
newCharacter = newWord = true;
 unsigned long startTime = millis();
digitalWrite(ledPin, HIGH);
tone(piezoPin, 1000);
delay(debounce);
while (digitalRead(buttonPin) == LOW)
delay(debounce);
delay(debounce);
unsigned long pressTime = millis() - startTime;
 digitalWrite(ledPin, LOW);
noTone(piezoPin);
currentSequence += (pressTime <=
dotDuration * 1.5) ? "." : "-";
}
unsignedlongtimeSinceLastInput =millis()-
(digitalRead(buttonPin) == LOW ? millis() :
(newCharacter || newWord ? millis() : 0));
if (newCharacter && timeSinceLastInput >=
letterPause) { decodeAndPrint();
currentSequence = ""; newCharacter = false;
}
if (newWord && timeSinceLastInput >=
wordPause * 1.5) { Serial.print("_"); newWord
= false; digitalWrite(ledPin, HIGH); delay(25);
digitalWrite(ledPin, LOW);
} }

void printMorse(char character)
{
 for (int i = 0; i < sizeof(morseChars) /
sizeof(morseChars[0]); i++)
{
if(pgm_read_byte(&morseChars[i])==characte
r)
{
Serial.println((constchar*)pgm_read_word(&
morseCodes[i]));
flashSequence((const
char*)pgm_read_word(&morseCodes[i]));
delay(letterPause);
return;
 } } }

void decodeAndPrint()
{
bool found = false;
for (int i = 0; i < sizeof(morseChars) /
sizeof(morseChars[0]); i++)
{
if (strcmp_P(currentSequence.c_str(), (const
```

```
char*)pgm_read_word(&morseCodes[i])) ==
0)
{
Serial.print(pgm_read_byte(&morseChars[i]));
found = true; break;
} }
if (!found) { Serial.print("?"); tone(piezoPin,
100, 500);
} }
void flashSequence(const char* sequence)
{
for (int i = 0; sequence[i] != '\0'; i++) {
digitalWrite(ledPin, HIGH); tone(piezoPin,
1000); delay(sequence[i] == '.' ? dotDuration :
dashDuration); digitalWrite(ledPin, LOW);
noTone(piezoPin); delay(dotDuration);
} }
```

III. WORKING PRINCIPLE

3.1 System Design and Component Choice

The project begins with a very detailed designing phase, where structuring of hardware and software is emphasized. The processor employed is the Arduino microcontroller with a push button as a secondary input source in the form of Morse code. Visual output and audio output are given through an LED and piezo buzzer, respectively. Debugging and output are achieved using the Serial Monitor of the Arduino IDE. It is modular in design to future-proof it for adaptability. It uses Morse timing conventions as user-selectable parameters, and PROGMEM-resident lookup tables translate strings of Morse characters to strings of alphanumeric characters.

3.2 Hardware Design and Software Coding

Hardware design is just designing the electronic circuit on a breadboard and controlling it while putting the buzzer, LED, and push button into the Arduino microcontroller. Software design in the Arduino IDE is for coding the simple functionality of the Morse code converter. These are correct identification of the input signal, real-time timing measurement, and effective algorithms for decoding the Morse code. Both visual and sound feedback systems are employed for the provision of feedback in real-time. Both accuracy and reliability of the system are also provided through extensive testing and debugging, wherein Serial Monitor is utilized to debug input and output data of the program and detect and rectify any bugs. The software is written in modular form for any future expansion and customization.

3.3 Development towards Improvement and Expansion (Future Stages of Development):

The wireless communication modules of Bluetooth or Wi-Fi would be used for wirelessly transmitting the Morse code, and voice synthesis integration is being proposed for synthesizing the audio output of decoded messages, in addition to ease of use offered by the present device for visually impaired people. An integration of a display screen is also under consideration, making the device self-contained, obviating the requirement for an attached computer. The project emphasizes modular design, enabling easy integration of different input and output modes, improving flexibility. Testing the usability integration of multiple peripherals will evaluate this modularity. These improvements are intended to expand the device's functionality and widen its area of application.

3.4 Evaluation and Documentation:

Effectiveness of the project is strenuously tested through controlled testing for accuracy testing, speed, and reliability of Morse code decryption. User feedback usability testing and user observation is utilized to allow for ease of use as well as ease of understanding of the system.Open-source comprehensive documentation in schematics, code, and instructions form is ready in an attempt to provide opportunities for collaborative development as well as information sharing. When applied to an academic setting, the learning outcomes are monitored through

ISSN [Online]: 2583-2654

survey and examination, including the educational contribution of the project. The measuring system ensures the success of the project and permits its application and dissemination to other purposes.

Fig 1: Morse Code

IV. CONCLUSION

This project can effectively employ an efficient Morse code decoder based on the functions of an Arduino microcontroller. A minimal push button input allows for the input of Morse code signals, with immediate feedback through an LED and buzzer. The decoded text is viewed using the Arduino Serial Monitor, with an easily readable and distinct output.

The device is cheap and simple to install and is therefore very well-adapted for educational settings and assistive communication.

Future expansion can also support the addition of wireless transmission modules for enabling far-end communication and the inclusion of speech-based Morse code interpretation, which would add further visually impaired user access. Furthermore, the system would also be able to accept input from other sources, such as a touch screen or voice input, to make it even more accessible. Additionally, the code can be made to consume less power, making it even more suitable for handheld and battery-powered uses.

REFERENCES

[1] Banzi, M., & Shiloh, M. (2014). Getting started with Arduino. Maker Media, Inc.

[2] IJARCCE. (2024). Morse Code Decoder Using Arduino. International Journal of Advanced Research in Computer and Communication Engineering.

[3] Monk, S. (2016). Programming Arduino: Getting Started with Sketches. McGraw-Hill Education.

[4] Raymond, E. S. (1999). The cathedral and the bazaar. O'Reilly Media, Inc.

[5] Arduino Project Hub. (n.d.). Morse Code Converter. Arduino Project Hub.

[6] Hobbyist electronics websites and blogs. (n.d.). Various Morse code projects using Arduino.

[7] Hackaday.com. (n.d.). Arduino Morse code projects.

[8] "Morse code to Text Convertor for Marine Communication", International Journal for Research in Applied Science & Engineering Technology by Pranali Suryavanshi, Vaishnavi Dhavale, Nishigandha Shinde, Prof. K.V. Jadhav, Mr. Nitin Ainapure.

[9] "Morse code translator using the Arduino platform: Crafting the future of microcontrollers", António Valente, Sérgio Silva, Salviano P. Soares, M. J. C. S. Reis, Jean Paiva, Paulo C. Bartolomeu.

[10] M. V. Bhalerao, S. S. Sonavane and V. Kumar, "A survey of wireless communication using visible light", IJAET, Jan 2013.

[11] G. Pang et al., "Visible Light Communication for Audio Systems", IEEE Trans. Consumer Elect., vol. 45, no. 4, pp. 1112-18, Nov. 1999.

[12] Muskon, M. T., Ismail, N., Shafie, R., Ab. Ghani, M. A. H., & Fadzal, N. (2017). Communication approach to handheld devices for disabled patients based on Morse code. ARPN Journal of Engineering and Applied Sciences, 12(9), 3456–3460.

[13] Rajpure, R., Pandekar, P., Shinde, S., Phadtare, S., & Nalagae, S. (2024). Morse Code Communication System Using Microcontroller. International Journal of Microelectronics and Digital Integrated Circuits, 10(1), 10-17p

[14] K.C.T Swamy., Vineeth Kumar Soma., V. Naresh., S. Mohammed Aquib., S. Mohammad Wahaz. (2023). The Encryption and Decryption of Morse Code Using Arduino. Journal of Electronics and Communication Systems. Vol. 8 No. 1

ISSN [Online]: 2583-2654

AUTHOR’S PROFILE

Saksham Thakur pursuing his B. Tech from MVSR Engineering College Nadergul, Hyderabad. Interested in learning Verilog Systems and Satellite Technology.

Gurrapu Pruthvik Sai pursuing his B. Tech from MVSR Engineering College Nadergul, Hyderabad. Interested in learning Robotics, Embedded System and Applications.