Morse Encoder Arduino Project Documentation

Project Overview

The Morse Encoder project uses an Arduino board to encode text input into Morse code and display it on an LCD module. It also generates audible signals by activating a buzzer to represent the Morse code symbols.

Components Used

Arduino board (UNO, Nano, or similar)

Bluetooth module (HC-05, HC-06, or similar)

LiquidCrystal_I2C library and compatible I2C LCD module

SoftwareSerial library for communication with the Bluetooth module

Buzzer

Jumper wires

Circuit Connections

Make the following connections between the components and the Arduino board:

Bluetooth module:

Connect the VCC pin to +5V on the Arduino

Connect the GND pin to GND on the Arduino

Connect the RXD pin to pin 2 (Tx) on the Arduino

Connect the TXD pin to pin 3 (Rx) on the Arduino

LCD module:

Connect the SDA pin to the SDA (A4) pin on the Arduino

Connect the SCL pin to the SCL (A5) pin on the Arduino

Connect the VCC pin to +5V on the Arduino

Connect the GND pin to GND on the Arduino

Buzzer:

Connect the positive (anode) pin to pin 11 on the Arduino

Connect the negative (cathode) pin to GND on the Arduino

Software Dependencies

This project relies on the following libraries:

Wire.h (built-in library for I2C communication)

LiquidCrystal_I2C.h (I2C library for LCD module)

SoftwareSerial.h (library for software-based serial communication)

Make sure these libraries are properly installed in your Arduino IDE.

Morse Code Encoding

The project uses a lookup table (morseTable) to map each uppercase letter from A to Z to its corresponding Morse code representation. The Morse code timings are defined as constants (dotDuration, dashDuration, etc.) in milliseconds.

The encodeChar() function takes a character as input, checks if it is a letter or space, converts it to uppercase, and retrieves the Morse code representation from the lookup table. It then displays the character and Morse code on the LCD module. It also produces sound by activating the buzzer according to the Morse code symbols.

Bluetooth Communication

The project uses the SoftwareSerial library to communicate with the Bluetooth module. The bt object is created with the Rx and Tx pins defined as pin 2 and pin 3, respectively. In the loop() function, incoming characters from the Bluetooth module are

read using bt.available() and passed to the encodeChar() function for Morse code encoding.

LCD Display

The project uses the LiquidCrystal_I2C library to control the LCD module. The library simplifies the communication between the Arduino and the LCD through I2C. The LCD is initialized in the setup() function, and the lcd object is created with the I2C address and the number of columns and rows of the LCD module.

The LCD is used to display the current character being encoded, its Morse code representation, and the final message once it is complete. The display is updated using the lcd.print() function.

Complete Message Display

Once the complete message is received and there is no input for a certain period (2 seconds in this case), the message is considered complete. The LCD is then updated to display the complete message for 5 seconds before clearing the screen.

Initialization and Main Loop

In the setup() function, the LCD is initialized, the Bluetooth communication is started, and the buzzer pin is set as an output.

The main functionality of the project is implemented in the loop() function. It continuously checks for incoming data from the Bluetooth module and encodes each character using encodeChar(). It also checks if the message is complete and displays it on the LCD.

Summary

The Morse Encoder Arduino project encodes text input into Morse code, displays it on an LCD module, and generates audible signals using a buzzer. It can be used as a learning tool for Morse code or as a standalone communication device. The project can be modified and expanded to include additional features or integrated into other projects.