Morse Code Transmitter and Receiver with LCD Display

This project is an Arduino-based Morse Code transmitter and receiver system using a 20x4 LCD display. It includes features like real-time Morse code encoding/decoding, button debouncing logic, and visual/audio feedback via LEDs and a buzzer. The system allows two-way communication with distinct transmit modes.

Table of Contents

Overview

This project allows users to communicate via Morse code using buttons. It displays the transmitted and received characters on a 20x4 LCD screen. It provides two distinct modes for transmitting Morse code signals, ensuring only one mode is active at a time.

Features

- Two distinct transmitter modes (Transmit 1 and Transmit 2).
- Real-time Morse code encoding and decoding.
- Visual and auditory feedback via LEDs and a buzzer.
- 20x4 LCD display to show the transmitted or received characters.
- Automatic line wrapping and display clearing for continuous use.
- Robust button debouncing logic to handle noisy inputs.

Hardware Requirements

1. Arduino Uno
2. 20x4 LCD with I2C interface
3. Push buttons (x2 for Morse code input)
4. Switches (x2 for transmit control)
5. LEDs (x2 for transmit mode indicators, x1 for input feedback)
6. Buzzer
7. Jumper wires

Circuit Diagram

How It Works

Debouncing Logic

Debouncing is used to eliminate false triggers caused by mechanical noise in button presses. The code tracks the button's state using the millis() function:

Debouncing a Button

```
if ((millis() - lastDebounceTime) > debounceDelay) {
// Button state change logic
}
```

- **Debounce Delay:** A constant debounceDelay (50ms) ensures that the button's state stabilizes before processing.
- **Logic:** When a button state changes, the code records the current time (millis()). It only considers the new state valid if the state remains unchanged for the debounce delay.

Transmit and Receive Modes

- **Modes:** The system has two transmit modes (Transmit 1 and Transmit 2). Only one can be active at a time.
- **Mode Switching:** Activating one mode disables the other and clears the LCD screen.
- **LEDs:** LEDs indicate the currently active mode.

Morse Code Decoding

- **Input Timing:** The duration of a button press determines whether the input is a dot (.) or a dash (-).

    - Dot: Press duration ≤ 1.5 times the dotLength.
    - Dash: Press duration > 1.5 times the dotLength.
- **Word and Letter Gaps:** Spaces between letters and words are determined by specific time intervals:
    - Letter Gap: 3x dotLength
    - Word Gap: 7x dotLength
- **Display:** Characters are displayed in real time on the LCD.

Code Explanation

Key Sections

1. **Setup Function:**
    - Initializes the LCD and pins.
    - Prints an introductory message on the Serial Monitor.
2. **Loop Function:**
    - Manages debouncing for transmitter buttons.
    - Handles Morse code input and decoding.
    - Updates the LCD display.
3. **Helper Functions:**
    - handleTransmitSwitch(): Handles activation and deactivation of transmit modes.
    - handleMorseCode(): Captures and processes Morse code input.
    - decodeMorse(): Decodes the Morse code sequence into a character.
    - displayDecodedChar(): Displays characters on the LCD.

Timing Using millis()

The millis() function is used to manage non-blocking timing operations:

- **Debouncing:** Ensures reliable button press detection.
- **Morse Code Timing:** Differentiates between dots, dashes, letters, and words based on time gaps.
- **Display Updates:** Controls when to clear or update the LCD.

Uniqueness of the Project

1. **Two-Way Communication:** Allows seamless switching between two transmitters.
2. **Real-Time Decoding:** Characters are displayed immediately upon decoding.
3. **Debouncing Logic:** Ensures accuracy in button inputs.
4. **Dynamic LCD Handling:** Automatically wraps text and clears the display when full.
5. **Error Handling:** Plays a tone for invalid Morse code sequences.

How to Use

1. Connect the components as per the circuit diagram.
2. Upload the provided code to the Arduino using the Arduino IDE.
3. Power on the system.
4. Use the transmit buttons to select a mode and input Morse code using the respective button.
5. View the decoded characters on the LCD.

Done By

Mohammed Riyas N

Bannari Amman Institute Of Technology