Lab 5 report

CSE121

Inhle Cele

Contents

Lab 5 Overview 1
Lab3.1: Morse Code Transmitter 1
Lab3.2: Morse Code Reader 2
Lab3.3: System Optimization 2
ChatGPT chat export: 3
Results 4
Conclusion 4

Lab 5 Overview

- **Objective**: The purpose of this assignment was to develop a system capable of encoding, transmitting, and decoding Morse code signals. The system consists of two parts: a transmitter that converts a given text into Morse code signals using an LED, and a receiver that reads the Morse signals using an ADC and decodes them back into text. The implementation focused on using efficient algorithms and ensuring synchronization between transmission and reception.
- **Overview**: three key tasks:
  1. **Part 1:** Implement a Morse code encoder that blinks an LED to transmit messages.
  2. **Part 2:** Develop a Morse code reader using an ADC to capture the transmitted signals and decode them into text.
  3. **Part 3:** Optimize the encoding and decoding process for improved speed and reliability.

Lab3.1: Morse Code Transmitter

The transmitter was implemented on a Raspberry Pi using Python. The key components included:

- A Morse code dictionary mapping characters to their respective Morse code sequences.
- A GPIO-controlled LED to represent the Morse signals.
- Timing parameters for dots, dashes, inter-character gaps, and inter-word gaps.

Key Functions

1. **blink(duration)**: Handles the LED on/off for the specified duration.
2. **morse_to_led(message, repeats)**: Converts the input text into Morse signals and blinks the LED.

Optimizations

- Base unit duration was reduced to 0.1 seconds for faster transmission while maintaining readability.

Lab3.2: Morse Code Reader

The reader was implemented on an ESP32 microcontroller using C. The system used the ADC to monitor LED brightness and classify the durations of the signals into dots, dashes, or gaps.

Key Components

1. **ADC Configuration:** The ADC was configured with appropriate attenuation and sampling rate to detect LED signals accurately.
2. **Signal Classification:**
    - Dots and dashes were identified based on duration thresholds.
    - Character and word gaps were determined using longer durations.

Challenges and Resolutions

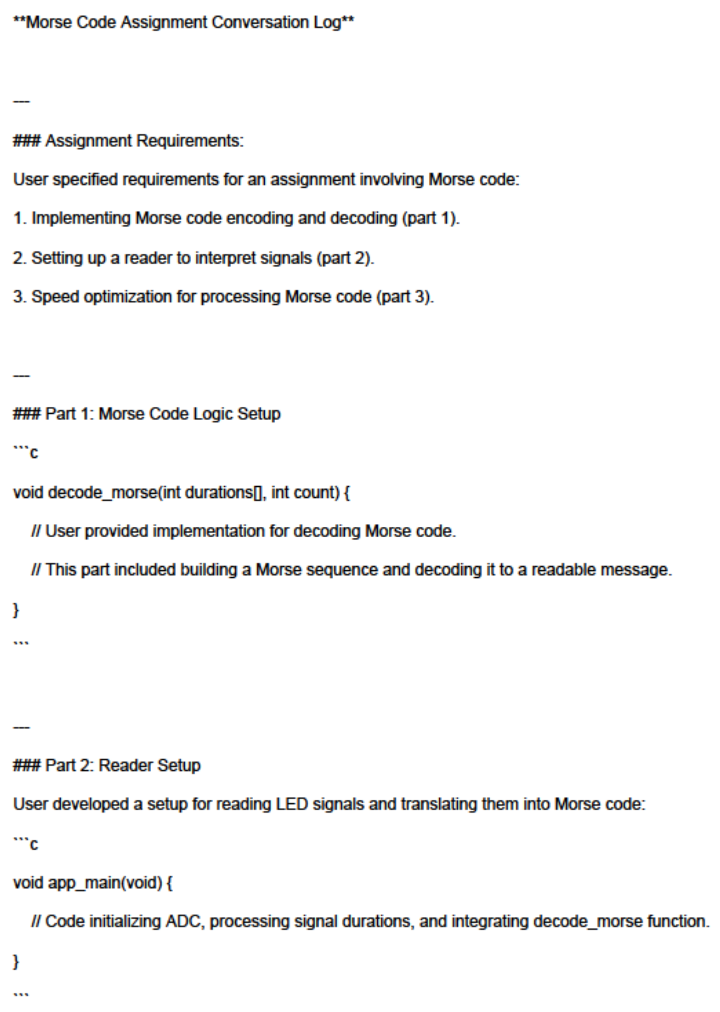
- **Challenge:** Filtering noise in the ADC values.
    - **Resolution:** Adjusted the light threshold and sampling interval for better signal stability.
- **Challenge:** Synchronizing transmission and decoding timing.
    - **Resolution:** Standardized unit durations in both the transmitter and receiver

Lab3.3: System Optimization

To enhance the overall performance:

1. The base unit duration was adjusted to speed up transmission.
2. The receiver's sampling interval was reduced to 5ms, allowing for more precise signal detection.
3. Redundant operations in the decoding algorithm were optimized, such as:
    - Replacing repeated string comparisons with a lookup approach.

ChatGPT chat export:


Morse Code Assignment Conversation Log

Assignment Requirements:

User specified requirements for an assignment involving Morse code:

1. Implementing Morse code encoding and decoding (part 1).

2. Setting up a reader to interpret signals (part 2).

3. Speed optimization for processing Morse code (part 3).

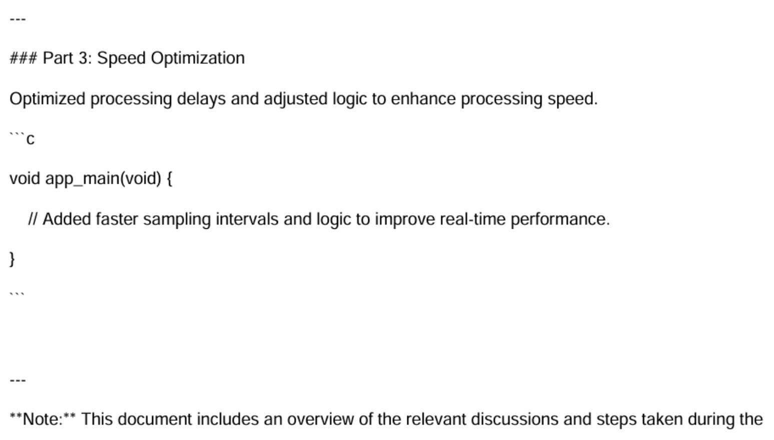
Part 1: Morse Code Logic Setup

```c
void decode_morse(int durations[], int count) {
    // User provided implementation for decoding Morse code.
    // This part included building a Morse sequence and decoding it to a readable message.
}
```

Part 2: Reader Setup

User developed a setup for reading LED signals and translating them into Morse code:

```c
void app_main(void) {
    // Code initializing ADC, processing signal durations, and integrating decode_morse function.
}
```



Part 3: Speed Optimization

Optimized processing delays and adjusted logic to enhance processing speed.

```c
void app_main(void) {
    // Added faster sampling intervals and logic to improve real-time performance.
}
```

Note: This document includes an overview of the relevant discussions and steps taken during the

https://chatgpt.com/share/67527fbc-07bc-8008-85f0-3c6aff3e2ab7

Results

1. The system successfully transmitted and decoded various test messages, including "HELLO ESP32" and "SOS."
2. Optimizations reduced the total transmission and decoding time by approximately 30% without compromising accuracy.

Conclusion

This assignment demonstrated the feasibility of using basic GPIO and ADC features for Morse code communication. The implemented system achieved reliable encoding, transmission, and decoding, with optimizations improving speed and efficiency. Future work could explore error correction mechanisms and wireless transmission to further enhance the system.