

Einführung in Git und Entwicklung mit lwIP-Stack

Martin

October 23, 2024

Agenda

- 1 Projektinitialisierung
- 2 Featureentwicklung
- 3 State Machine Implementierung
- 4 Modulentwicklung
- 5 Projekt abschließen

Projekt von GitHub herunterladen

- Initialisierung der Submodule:

```
git clone https://github.com/...  
git submodule init  
git submodule update --recursive
```

- Alle Bibliotheken werden geladen
- Vorbereitetes lwIP-Projekt beinhaltet alle notwendigen Implementierungen

PBranch für neues Feature anlegen

- Neuen Branch erstellen:
git checkout -b State-Machine
- Neues Projekt erstellen, Beispielprojekte nicht verändern
- Script „create new project“ nutzen

Projekt kompilieren und flashen

- "Clean All" vor dem Kompilieren ausführen
- Projekt auf das Board flashen
- Wichtig: stabile Verbindungen sicherstellen (USB-Verlängerung)

State Machine entwickeln

- State Machine für Task-Management implementieren
- Funktionszeiger und Zeithandling für Tasks
- Code soll erweiterbar sein

Verwendung von ChatGPT für Codegenerierung

- Nutzung von ChatGPT zur Unterstützung bei der Codeerstellung
- Beispiel: State Machine mit Funktionszeigern für Tasks
- Anpassung des generierten Codes für Echtzeitfähigkeit

Anlegen eines neuen Moduls

- Neues Modul für LED-Steuerung erstellen
- Struktur von GPIO-Pins definieren
- Precompiler-Direktiven und Funktionsdeklarationen verwenden

Implementierung der LED-Steuerung

- Implementierung der Funktion ‘ToggleGPIO‘
- Funktionsaufruf im Task-Manager integrieren
- Debugging ohne printf, stattdessen LED-Feedback verwenden

PCodeversionierung

- Änderungen committen:
 `git add .`
 `git commit -m "Implemented LED control and state machine"`
- Branches sauber verwalten
- Projektdateien und neue Module korrekt versionieren

Zusammenfassung

- Projektinitialisierung und Submodule
- State Machine und Taskverwaltung
- Nutzung von ChatGPT zur Codeerstellung
- Modulentwicklung für GPIO-Steuerung
- Commit und Versionierung