Projektgruppe: 4 Studierende

Bearbeitungszeitraum: 28.4.2025 - 16.6.2025

Gesamtpunkte: 13 + 2 Bonuspunkte

1. Hintergrund

Sie haben Zugang zu einem Hardware-in-the-Loop (HIL) Simulator, der auf einem Raspberry Pi Pico läuft. Dieser Simulator emuliert das physikalische Verhalten eines 3-Achsen-Roboters (X, Y, Z) mit einem Greifer. Die detaillierte Dokumentation des HIL-Simulators (Funktionsweise, Pinbelegung, I2C-Protokoll) liegt Ihnen vor und ist Grundlage für dieses Projekt.

https:/dev.azure.com/mscheifinger/3BHWII/_git/hil_simulation

2. Zielsetzung

Ihre Aufgabe ist es, mit einem eigenen Raspberry Pi Pico eine Steuerung für den simulierten Roboter zu entwickeln. Diese Steuerung soll über GPIOs und I2C mit dem HIL-Simulator kommunizieren, um den Roboter manuell zu verfahren und einfache Bewegungsprogramme abzuarbeiten.

3. Anforderungen und Aufgaben

Teil A: Manuelle Steuerung und Soft-Limits (3 Punkte)

- Implementieren Sie eine manuelle Steuerung, mit der jede der drei Achsen (X, Y, Z) über Taster (oder andere geeignete Eingabeelemente) in positive und negative Richtung verfahren werden kann.
- **Soft-Limits:** Die Steuerung muss sicherstellen, dass die Achsen automatisch gestoppt werden, *bevor* sie ihre physikalischen Endlagen erreichen. Nutzen Sie die über I2C ausgelesene Position vom HIL-Simulator, um die Bewegung rechtzeitig abzubremsen und anzuhalten. Die simulierten Endschalter (PIN_X/Y/Z_END_MIN/MAX) sollen dabei **nicht** aktiviert werden (d.h. nicht auf 1 gehen).

Teil B: Programm-Abarbeitung via Serieller Schnittstelle (4 Punkte)

- Ermöglichen Sie das Empfangen und Speichern von einfachen Bewegungsprogrammen über die USB-Serielle-Schnittstelle Ihres Steuerungs-Picos.
- Das Format für die Eingabe soll dem folgenden Beispiel ähneln (pro Zeile ein Befehlssatz, mehrere Sätze durch Semikolon getrennt oder pro Zeile ein Satz):
  x 50.0, y 100.0, z 10.0, gripper open
  x 20.0, y 70.0, z 30.0, gripper closed

(Das genaue Parsing-Format können Sie definieren, es muss aber klar dokumentiert sein).

- Ihre Steuerung muss mindestens 10 solcher Positionssätze speichern können.
- Implementieren Sie eine Funktion, um das gespeicherte Programm sequenziell (Position für Position) abzufahren.

Teil C: Dokumentation und Entwurf (insgesamt 6 Punkte)

- **Verschaltungsplan (2 Punkte):** Erstellen Sie einen klaren Schaltplan (z.B. mit Fritzing, draw.io oder von Hand gezeichnet und gescannt), der zeigt, wie Ihr Steuerungs-Pico mit dem HIL-Simulator-Pico verbunden ist (alle notwendigen GPIOs und I2C-Leitungen).
- **Pin-Belegung (1 Punkt):** Erstellen Sie eine Liste aller GPIO-Pins, die Sie auf Ihrem Steuerungs-Pico verwenden, und beschreiben Sie deren Funktion (z.B. GPIO 10: Taster Achse X+, GPIO 2: PWM Achse Y, GPIO 4: I2C SDA, etc.).
- **Zustandsautomaten (State Machines) (3 Punkte):** Entwerfen Sie mindestens zwei Zustandsautomaten (State Machines) für relevante Teile Ihrer Steuerung (z.B. für das Abbremsen vor Erreichen der Soft-Limits, für die Abarbeitung der seriellen Befehle, für die Greifersteuerung). Zeichnen Sie diese Automaten (z.B. als UML State Diagram) und implementieren Sie deren Logik in mindestens einem FreeRTOS-Task Ihrer Steuerungssoftware.

Teil D: Bonusaufgabe - Koordinierte Bewegung (2 Bonuspunkte)

- Erweitern Sie die Programm-Abarbeitung aus Teil B so, dass bei einer Bewegung zu einem neuen Zielpunkt (X, Y, Z) alle beteiligten Achsen ihre Bewegung so koordinieren, dass sie **gleichzeitig** am Ziel ankommen. Dies erfordert eine Berechnung der Bewegungszeiten und ggf. eine Anpassung der Geschwindigkeiten der einzelnen Achsen.

4. Abgabe

Folgende Unterlagen und Ergebnisse sind fristgerecht digital einzureichen:

1. **Source Code:** Der vollständige, gut kommentierte C/C++ Quellcode Ihrer Steuerungssoftware für den Raspberry Pi Pico (inkl. CMakeLists.txt etc.).
2. **Verschaltungsplan:** Die erstellte Zeichnung (siehe Teil C).
3. **Pin-Belegung:** Die erstellte Liste (siehe Teil C).
4. **Zustandsautomaten:** Die Zeichnungen und eine kurze Beschreibung, in welchem Teil des Codes die Automaten implementiert wurden (siehe Teil C).

5. **Kurze Dokumentation/Bedienungsanleitung:** Beschreibung, wie Ihre Steuerung bedient wird (manuelle Steuerung, serielles Protokoll für Teil B).
6. **Demonstration:** Eine kurze Live-Demonstration der Funktionsfähigkeit Ihrer Steuerung mit dem HIL-Simulator

Bewertung: Die Punkteverteilung ist bei den einzelnen Aufgabenbereichen angegeben. Die Funktionalität, die Codequalität (Lesbarkeit, Kommentierung, Struktur), die Qualität der Entwurfsdokumente und die erfolgreiche Demonstration fließen in die Bewertung ein.

Viel Erfolg!