

Projektablaufplan

Einarbeitung	
Geplante Anzahl an Stunden: 10 Stunden Zeitraum: 10/2022	Inhalt: Einarbeitung in das FreeRTOS, welches auf dem ESP32 läuft. Besonderes Augenmerk dabei, die Erstellung von Tasks, die Kommunikation zwischen Tasks, die Priorisierung von Tasks und die Unterbrechung von Tasks durch Hardwareinterrupts. Dabei soll auch das ESP IoT Development Framework (ESP-IDF) betrachtet werden, welches für die Programmierung des ESP verwendet wird. Ebenso soll die Bluetooth Low Energie (BLE) Technologie betrachtet werden und die Human Interface Devices (HID) welche per USB oder Bluetooth bereitgestellt werden können.
BLE HID Joystick implementieren	
Geplante Anzahl an Stunden: 30 Stunden Zeitraum: 10/2022 - 11/2022	Inhalt: Mittels des ESP32-Entwicklerboards die Kommunikation zwischen Windows, Linux, iOS / iPadOS und Android herstellen (MacOS kann nicht überprüft werden, da kein Gerät mit MacOS vorhanden ist). Die Kommunikation soll mittels BLE stattfinden und das Entwicklerboard soll sich als HID authentifizieren. In diesen Schritt sollen alle analogen und digitale Knöpfe, sowie die analogen Steuerknüppel übertragen werden. Auch soll / muss (für Apples Betriebssysteme) der aktuelle Akkustand des Geräts übertragen werden. In diesen Stadium soll nur ein Demo-Akkustand übertragen werden. Überprüft werden soll die Funktionsweise mittels eines Programms welche alle Übermittelten Tastensignale darstellt.
Kommunikation zwischen Mikrocontroller und Fernbedienung implementieren	
Geplante Anzahl an Stunden: 20 Stunden Zeitraum: 11/2022 – 12/2022	Inhalt: Die Fernbedienung bietet verschiedene Übertragungstechniken zum Modulschacht. Einige Übertragungstechniken sind für bestimmte Module bereits implementiert, auch werden die universellen Protokolle SBus und PPM unterstützt. Für die Datenübertragung in diesem Projekt soll das Protokoll CRSF verwendet werden.
Implementierung weiterer Eingabe- und Ausgabe-Elemente	
Geplante Anzahl an Stunden: 10 Stunden Zeitraum: 01/2023	Inhalt: Für die Bedienung des Moduls (setzen von Einstellungen) sollen Taster verwendet werden. Durch ein 0,91“ großes OLED-Display sollen kurze Statusnachrichten gezeigt werden. Zusätzlich sollen

	<p>noch ein bis zwei Status-LEDs vorhanden sein, welche ebenfalls für die Interaktion mit dem Modul dienen sollen.</p> <p>Für die Bestimmung des Akkustands der Fernbedienung soll ein integrierter ADC verwendet werden, mit welchem die Versorgungsspannung des Modulschachts ermittelt werden kann. Der Akkustand wird für die Bereitstellung des HID benötigt.</p>
Zusammenführung Software	
<p>Geplante Anzahl an Stunden: 20 Stunden</p> <p>Zeitraum: 01/2023 - 02/2023</p>	<p>Inhalt:</p> <p>Alle bis zu diesem Schritt vorhandenen Komponenten sollen in diesen Schritt zusammengeführt werden. Dabei muss beachtet werden, dass das ESP-Modul zwei Kerne besitzt welche durch das vorhandene FreeRTOS angesteuert werden können. Dafür müssen alle Komponenten in Tasks oder ISRs ausgelagert werden und die richtigen Prioritäten für das Scheduling gesetzt werden, damit die Übertragung der Daten über Bluetooth mit einer möglichst geringen Latenz erfolgt.</p>
Platinenentwurf und -bestückung	
<p>Geplante Anzahl an Stunden: 30 Stunden</p> <p>Zeitraum: 02/2023 - 03/2023</p>	<p>Inhalt:</p> <p>In diesem Schritt soll der vorhandene Aufbau auf Steckbrettern in eine Platine umgewandelt werden, welche in Auftrag gegeben und danach bestückt werden soll. Dabei sollen noch weitere kleine Schaltungen integriert werden, um die Benutzung des Moduls zu verbessern.</p> <p>Die Bestandteile der Platine sind:</p> <ul style="list-style-type: none"> • ESP-Modul • OLED-Display • Taster mit einer Schaltung zum entprellen der Taster • Status-LEDs • Schaltung zur Spannungsregulierung der Leitungen die vom Modulschacht kommen • Stiftheile für die Kommunikation zwischen der Fernbedienung und des Moduls <p>Es sollen zwei Platinen entworfen werden. Die erste Platine soll dabei das ESP-Modul und die Spannungsregulierung enthalten. Die zweite Platine sollen die LEDs, Taster und das OLED-Display enthalten. Dadurch wird das Bestücken der Platine und der Einbau in das Gehäuse vereinfacht.</p> <p>Als Software hierfür soll KiCad verwendet werden.</p>
Gehäuseentwurf und -erstellung	
<p>Geplante Anzahl an Stunden: 10 Stunden</p> <p>Zeitraum:</p>	<p>Inhalt:</p> <p>Es soll ein Gehäuse entworfen werden, in dem die vorhandene Platine im JR-Modulschacht der Fernbedienung sicher befestigt</p>

04/2023	werden kann. Beachtet werden muss dabei, dass das Gehäuse so entworfen werden soll, dass das Gehäuse möglichst ohne Stützstrukturen 3D gedruckt werden kann. Dadurch ist nur eine geringe bis keine Nachbearbeitung des Gehäuses nötig.
Test / Vergleich zwischen USB- und BLE-Verbindung	
Geplante Anzahl an Stunden: 20 Stunden Zeitraum: 05/2023 - 06/2023	Inhalt: Mittels eines Einplatinencomputers soll die durchschnittliche Übertragungszeit zwischen USB- und BLE-Verbindung ermittelt werden. Dafür wird an einem Taster der Fernsteuerung ein Optokoppler angelötet, womit der Taster via dem Einplatinencomputer gesteuert werden kann. Die Zeit zwischen der Ansteuerung des Tasters und dem Empfang der Daten der Fernbedienung am Einplatinencomputer soll dabei gemessen werden. Die Versuchssoftware soll dabei in C geschrieben werden. Die Erkennung der Joystick-Ereignisse, unter Linux, soll durch die evdev-Schnittstelle erfolgen. Durch diese Schnittstelle erhalten alle Ereignisse einen Zeitstempel, wann diese im Kernel aufgetreten sind. Mittels mehreren Durchläufen soll die mittlere Laufzeit ermittelt werden, da Jitter durch das Betriebssystem Linux auftreten können. Alle Durchläufe sollen dabei in einer CSV-Datei für eine spätere Auswertung gespeichert werden.
Dokumentation	
Geplante Anzahl an Stunden: 150 Stunden Zeitraum: 10/2022 – 06/2023	Studienarbeit schreiben.
Optional: Aktualisierung des Moduls mittels OTA oder einer Desktopsoftware	
Zeitraum: Optional	Inhalt: Das ESP-Modul bietet die Möglichkeit Softwareaktualisierungen via OTA zu laden. Dafür müssen mehrere Partitionen im Speicher erstellt werden, welche zwei verschiedene Stände der Software enthalten. Der Bootloader entscheidet durch eine gesetzte Marke, welche Software von welcher Partition geladen werden soll. Die Übertragung der Daten kann entweder über Bluetooth oder WLAN erfolgen.