



Andreas Christen



Dominik Maurer

Diplomanden	Andreas Christen, Dominik Maurer
Examinator	Prof. Erwin Brändle
Experte	Theo Scheidegger, Swens GmbH, Schänis, SG
Themengebiet	Embedded Systems

Embedded GUI via WiFi-Link



Abb. 1: Gesamtsystem
Eigene Darstellung

Ausgangslage: Zum Lieferumfang handelsüblicher Servomotor und Brushed DC-Motor Controller gehört oft eine vom Hersteller massgeschneiderte grafische Bedieneroberfläche. Je nach Modell können so beispielsweise die Konfigurations- und Regelparameter eines Motors eingestellt oder seine dynamischen Regelgrößen wie Position, Geschwindigkeit oder Motorenstrom grafisch dargestellt werden. Da solche GUIs häufig für Windows entwickelt und über USB-Interfaces betrieben werden, ist deren Einsatz jedoch meist auf PC-Umgebungen beschränkt. Oft wäre es aber wünschenswert, eine plattformunabhängige Bedienung greifbar zu haben, welche drahtlos mit den Motor-Controllern kommunizieren kann.

Ziel der Arbeit: Im Rahmen dieser Bachelorarbeit wurde eine plattformunabhängige Bedieneroberfläche für bestehende Brushed DC-Motor Controller entwickelt (Abb. 1). Dabei ist das GUI vollständig auf HTML, CSS und Javascript aufgebaut. Für die drahtlose Verbindung zwischen dem GUI und den Controller-Modulen wird ein Lolin ESP 32 WiFi-Modul verwendet. Die Kommunikation zwischen dem WiFi-Modul und den Motor-Controllern erfolgt über die jeweils verfügbaren seriellen Schnittstellen (UART, SPI, I2C).



Abb. 2: Systemübersicht
Eigene Darstellung

Vorgehen: Gemäss der Systemübersicht (Abb. 2) wurde die Aufgabenstellung in einer ersten Phase für den Mini Maestro 12-Channel Servo Controller umgesetzt. Die Software wurde mit dem Code Editor Visual Studio Code von Microsoft geschrieben. Um Plattformunabhängigkeit zu garantieren wurde das GUI vollständig als HTML-Client umgesetzt. Die erforderlichen HTML-, CSS- und Javascript-Files werden über eine SD-Card eingelesen. Es wurde lediglich ein Subset der Funktionalitäten des vom Hersteller mitgelieferten GUI implementiert. Über einen Slider kann ein Servomotor u. a. an verschiedene Anfangs- und Endpositionen gefahren werden (Abb. 1). Aus dieser ersten Phase konnten wichtige Erkenntnisse über den Workflow gewonnen werden. Als besonders neuralgische Stellen in der Umsetzung sind u. a. das Filehandling via HTTP Protocol oder die Kommunikation zwischen HTML Client und Webserver über Websockets zu erwähnen.

In einer zweiten Phase wurde ein GUI für den RoboClaw 2x7A Motor Controller implementiert. Auch hier stand nicht die vollständige Umsetzung aller Funktionen im Vordergrund. Die Erkenntnisse aus der ersten Phase erleichterten die Implementation des GUI für den RoboClaw Controller wesentlich, da für einige kritischen Stellen bereits Lösungsansätze erarbeitet wurden. Das Hersteller-GUI zum RoboClaw ist sehr umfangreich und bietet beispielsweise die Möglichkeit dynamische Regelgrößen wie Position, Geschwindigkeit oder Motorenstrom grafisch darzustellen. Die grafische Darstellung einzelner Regelgrößen lässt sich auch mit HTML, CSS und Javascript realisieren (Abb. 3).

Für den hardwaremässigen Aufbau wurde ein passendes PCB entworfen und hergestellt (Abb. 1). Der so entstandene Demonstrator ermöglicht eine einfache Montage der verwendeten Motor-Controller und erleichtert die Inbetriebnahme des gesamten Systems.



Abb. 3: GUI RoboClaw
Eigene Darstellung