

Laborautomatisierung mit MUSCLE



Posterbeitrag zur 20. Nationalen PV-Tagung 2022 von David Joss, Mischa Müller, Peter Wüthrich, Sina Spring, Marco Zaugg, Matthias Burri und Prof. Dr. Christof Bucher, PV-Labor der Berner Fachhochschule

Ein Prüflabor an einer Fachhochschule soll einerseits korrekte und wiederholbare Prüfergebnisse erzielen, andererseits Studierenden Kenntnisse im Bereich der Prüfung und Laborautomation vermitteln können. An der Berner Fachhochschule wurde dafür das MUSCLE-Konzept entwickelt: In einer Embedded Linux Umgebung wird die Kommunikation zwischen Benutzer:in, den zu steuernden Simulatoren und den Messgeräten sichergestellt. Die browserbasierte Bedienoberfläche ermöglicht es, typische Prüfungen am PV-Labor wie die Kennlinienmessung an PV-Modulen oder automatisierte Wechselrichtertests durchzuführen.

Wirkungsfeld PV-Labor

Das PV-Labor der Berner Fachhochschule ist in der Photovoltaik-System tätig und führt regelmässig Messungen an PV-Modulen und Wechselrichtern durch.

Das Rückgrat der modernen Messeinrichtungen bildet MUSCLE. MUSCLE ist ein Open Source Projekt zur Laborautomatisierung, welches an der Berner Fachhochschule entwickelt wurde. Es ermöglicht eine gut strukturierte Ansteuerung der unterschiedlichsten Geräte im Prüflabor.

Konzept von MUSCLE

Das Konzept von MUSCLE basiert auf drei Hauptelementen.

Frontend - Mittels einer Single Page Webapplikation, welche über Smartphone, Tablet oder Computer aufgerufen wird, werden verschiedene Laborinstrumente/systeme gesteuert und ausgelesen.

Device - Zur direkten Verbindung mit dem Laborinstrument oder Laborsystem wird ein Embedded Linux Gerät eingesetzt, welches anhand eines unterstützten Protokolls (gängig sind TCP/MODBUS/GPIB/CAN) mit dem Instrument/System kommuniziert.

Backend - Im Hintergrund verwaltet ein Backend-Server sämtliche Benutzer und Geräte.

Die Daten oder Befehle werden über AJAX-HTTP- und JSON RPC Protokolle zwischen dem Device und dem Frontend ausgetauscht. Die Implementation der Peripheriegeräte am PV-Labor ist in Python umgesetzt und für die APIs kommt Flask zum Einsatz.

