



Test- und Validierung für Sicherheitssysteme von Teilchenbeschleuniger

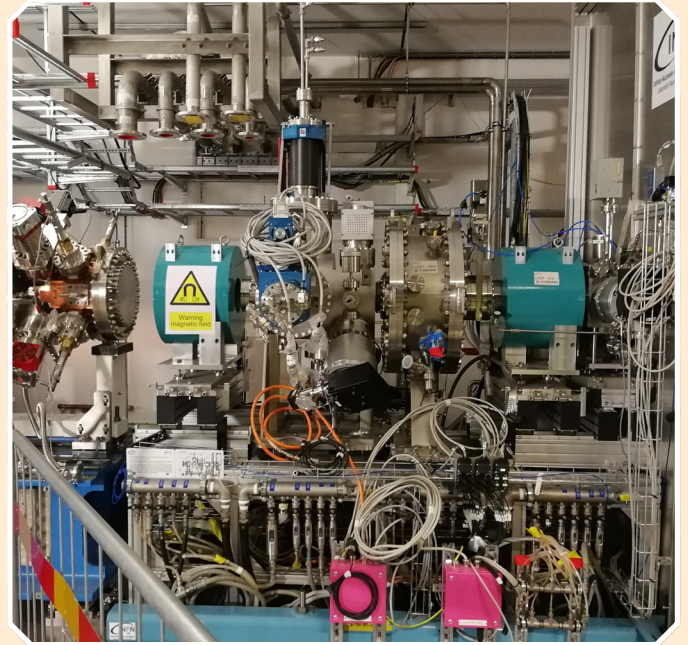
Kunde: ZHAW

Lieferumfang: VeriStand Custom Devices, VeriStand FPGA Personality, LabVIEW APIs, TestStand APIs, TestStand Unit Test Sequenzen,

Technologien: NI VeriStand 2018, PXI-System von National Instruments (NI)

Programmiersprachen: LabVIEW 2018, NI TestStand 2017

Speziell: Das Resultat dieses Projektes ist ein Framework bestehend aus Bausteinen, mit welchen die ESS die implementierten Tests einfach und ohne externe Hilfestellung ändern und erweitern kann. Ebenfalls wird die ESS mit diesem Framework nebst dem Fast Beam Interlock System auch andere Systeme testen.



Einleitung

Die European Spallation Source ESS in Lund (Schweden) ist eine im Bau befindliche Forschungsanlage, die lange Pulse von Neutronenstrahlung mit sehr hoher Intensität erzeugen wird. Diese Anlage ermöglicht Struktur- und Dynamikuntersuchungen von Materialien auf molekularer und atomarer Skala. Anders als in klassischen Neutronenquellen werden die Neutronen an der ESS nicht durch Kernspaltung in einem Reaktor, sondern durch Beschuss von Metall (Wolfram) mit Protonen erzeugt (Kernzertrümmerung bzw. «Spallation»). Sowohl mit der verwendeten Technologie als auch mit den Leistungsdaten betritt die ESS dabei Neuland und wird die weltweit leistungsfähigste Neutronenquelle sein.



Darstellung der ESS mit dem Beschleuniger (überdeckter Tunnel) der Spallationquelle (im Zentrum) und den umgebenden Forschungsarealen

Abbildung: «Erstes Stück der Strahlführung (im Aufbau) nach der Ionen Quelle. Eine Vielzahl von Sensoren überwachen die Extraktion und Beschleunigung der Protonen.»

Die ZHAW ist damit beauftragt, für die ESS das sogenannte Fast Beam Interlock System (FBIS) zu konzeptionieren, entwickeln, integrieren und testen. Das FBIS dient dem Schutz der Anlage und überwacht dazu eine Vielzahl von Sensor-Systemen. Wird eine Abweichung vom Soll-Zustand festgestellt, so aktiviert das FBIS innert wenigen Mikro-Sekunden eine Serie von Aktuatoren. Die Latenzanforderungen gepaart mit den Vorgaben zur Verfügbarkeit der ESS (Verhindern eines Ausfalls) stellen eine besondere Herausforderung dar. Um die Nachweise diesbezüglich zu erbringen, wurde für das FBIS ein Test- und Validierungssystem entwickelt. Das Test- und Validierungssystem des FBIS wird mit eigens entwickelter Hardware (Signal Adapter und Routing) wie auch mit Produkten von National Instruments (NI-PXI, LabVIEW, VeriStand, TestStand) realisiert. Die Tests umfassen sowohl Modultests (Unit Tests) von einzelnen FBIS Komponenten wie auch Integrationstests, bei welchen das Testsystem dynamisch, mittels LabVIEW FPGA Code und Matlab Simulink Modellen auf Ausgaben des Prüflings reagiert (Hardware In The Loop – HIL).

Test- und Validierung für Sicherheitssysteme von Teilchenbeschleuniger



NI TestStand

TestStand ist eine Entwicklungsumgebung für automatisierte Mess- und Prüfsysteme. Mithilfe eines Sequenzeditors können Prüfsequenzen konfiguriert werden, welche die Ausführung von Codemodulen automatisieren, ganz gleich in welcher Programmiersprache diese verfasst wurden (LabVIEW, C/C++ DLL, .NET, ...). Im FBIS Testsystem werden die Codemodule ausschliesslich in wiederverwendbaren LabVIEW Bibliotheken entwickelt. Prüfsequenzen, welche in mehreren Tests vorkommen, sind in eigenen «Sequenz-Files» organisiert und bilden TestStand Bibliotheken für das FBIS Test System.

NI VeriStand

Die Software VeriStand ermöglicht eine effiziente Erstellung von Echtzeit-Prüfanwendungen mithilfe einer konfigurationsbasierten Entwicklungsumgebung. Sie bietet eine vorgefertigte Architektur für Echtzeit Tests (Multicore-fähige Echtzeit-Engine). Ein und Ausgänge von NI Hardware können sehr einfach durch Konfiguration eingebunden werden. Ergänzend zu den bestehenden Möglichkeiten können auch benutzerdefinierte «Custom Devices» mit LabVIEW programmiert werden, welche eine standardisierte Schnittstelle für die Integration in VeriStand bereitstellen. Diese «Custom Devices» können zum Beispiel komplexere Messaufgaben mit beliebiger Logik implementieren. Äquivalent kann benutzer-

definierter FPGA Code als «FPGA Personality» erstellt und integriert werden. Steuerungsalgorithmen und Systemmodelle aus LabVIEW und Modellierungsumgebungen wie MATLAB Simulink können in VeriStand importiert werden.

TestStand PC

Je nach Testsituation sind der VeriStand PC sowie das NI PXI System schwer zugänglich. Die Architektur sieht deshalb vor, dass die Testbedienung mit TestStand auf einem separaten Rechner ausgeführt werden kann. Ein Test wird in «Sequence Files» definiert, welche von der TestStand Engine interpretiert und ausgeführt werden. In vielen Fällen sind jedoch der VeriStand- und der TestStand PC das gleiche Gerät.

VeriStand PC

Auf diesem Rechner wird das Test-System mit NI VeriStand definiert und die Konfiguration mit allen zugehörigen Modulen (z.B. «Custom Devices», Modelle, ...) auf die Zielsysteme geladen und ausgeführt.

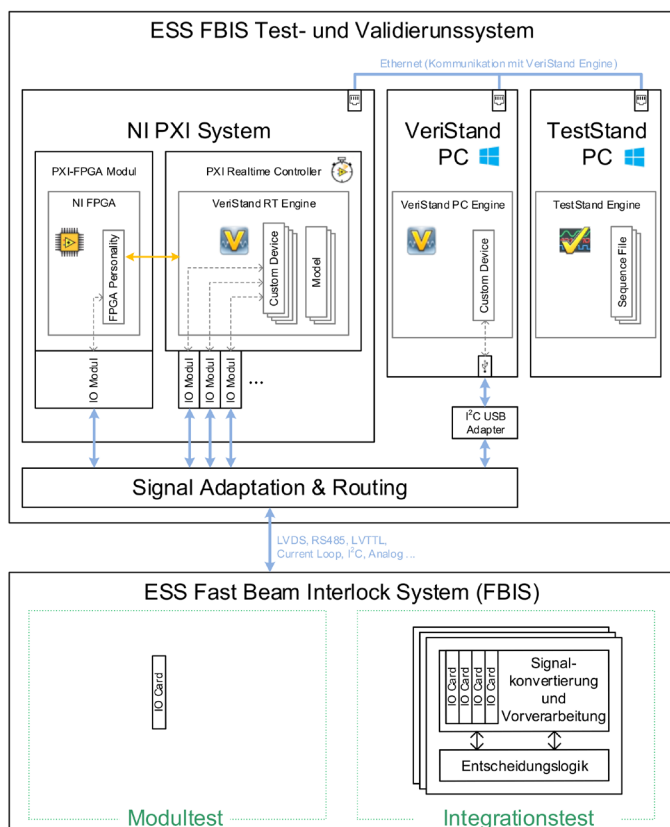
Weiter ist dieser Rechner mit dem I2C USB Adapter für die Kommunikation mit den I2C Slaves des FBIS verbunden. Die Schnittstelle zu diesem I2C Master (Adapter) und allen Slaves bildet ein dazu erstelltes «Custom Device».

NI PXI System

Die VeriStand Echtzeit-Testanwendung läuft auf dem Realtime Controller des PXI Systems. Hier werden unter anderem die «Custom Devices» und Modelle zyklisch ausgeführt. Ein Custom Device bildet in der Regel eine benutzerdefinierte Schnittstelle zu einem PXI IO Modul (mit integrierter Intelligenz). Das PXI System enthält nebst verschiedensten IO Modulen für analoge und digitale Signalerfassung und Generierung eine FPGA Karte. Auf dieser läuft das sogenannte «FPGA Personality», eine nebst den Modellen und «Custom Devices» weitere Möglichkeit, benutzerdefinierten Code in VeriStand zu integrieren.

Signal Adapter & Routing

Die Anbindung des ESS Fast Beam Interlock Systems an das Testsystem wurde mit von der ZHAW entwickelte Hardware realisiert.



Die Abbildung zeigt eine grobe Übersicht des FBIS Test- und Validierungssystems.

Test- und Validierung für Sicherheitssysteme von Teilchenbeschleuniger