Konzept für ein Python-Framework für die Verwendung von NI-Hardware

Konzeptpapier zur Seminararbeit von Zakaria Bara

Inhaltsverzeichnis

Einleitung

Am *Institut für Schweiß- und Fügetechnik* (ISF) der RWTH Aachen University werden im Rahmen von Schweißversuchsreihen für diverse Forschungsprojekte Mess- und Steuerungssysteme der Firma National Instruments (NI) verwendet. In diesem Kontext werden Treiber der Firma wie auch ihre grafische Programmierumgebung Laboratory Virtual Instrument Engineering Workbench (LabVIEW) genutzt. Die LabVIEW-Programmierung ist daher ein essenzieller Bestandteil für praktische Versuche am ISF, einerseits im Rahmen der Prozesssteuerung und andererseits bei der Datenerfassung verschiedener Versuchsparameter. Momentan existiert bereits ein in LabVIEW erstelltes, objektorientiertes Framework (DAQmxOOP), das es ermöglicht, neue Programme für wiederkehrende, ähnliche Aufgabenstellungen mit wenig Aufwand erstellen zu können. In der geplanten Seminararbeit soll ein Konzept für ein Python-Framework entwickelt werden, welches Mess- und Steuerungssysteme von NI samt ihrem Treiber für Forschungsprojekte nutzbar macht. Dabei wird DAQmxOOP als Vorlage genutzt, um eine vergleichbare Funktionalität abzubilden.

Softwaretechnische Randbedingungen

Das bisherige Framework DAQmxOOP basiert auf LabVIEW. Im Konzept wird davon ausgegangen, dass in Python 3 programmiert und eine virtuelle Python-Umgebung über Package Installer for Python (pip) verwaltet wird. Welche konkreten Python-Module am besten geeignet sind, wird sich im Rahmen der Seminararbeit ergeben.

Motivation

In der Vergangenheit sind die Preise für die LabVIEW-Lizenzen stetig gestiegen. Zur selben Zeit hat sich Python als kostengünstige und flexible Alternative etabliert. Mittlerweile sind viele Anwendungsfälle der am ISF entwickelten LabVIEW-Programme mit Python und entsprechenden Python-Bibliotheken realisierbar. Im Rahmen dieser Seminararbeit wird zunächst das bestehende DAQmxOOP-Framework analysiert und ein Konzept für eine Umsetzung mit vergleichbarerer Funktionalität in Python erstellt. Darauffolgend können Klassen und Methoden iterativ übersetzt und mit Test-Unit Programmen getestet werden.

Ziel und Fragestellungen

Das Ergebnis dieser Seminararbeit soll folgende Ziele adressieren:

[Z1] Schriftliche Erörterung der Funktionalität und der Features des DAQmxOOP-Frameworks, wie auch der Klassen und Methoden.

[Z2] Erstellung eines Klassendiagrammes,eines Aktivitätsdiagrammes des DAQmxOOP-Frameworks, und des Konzepts.

[Z3] Ausarbeitung und Analyse, welche Features durch die von NI gestellten Python-Module abgedeckt werden.

[Z4] Schriftliche Erörterung der Definitionen von Test-Unit Klassen als Teil des Konzepts.

[Z5] Schriftliche Erörterung über mögliche Erweiterungen und Zusammenstellung des Konzepts.

[F1.1] In welche Komponenten lässt sich DAQmxOOP unterteilen?

[F1.2] Welche Features sind essenziell für das zu erstellende Konzept?

[F2.1] Welche Klassenstruktur ist, notwendig und welche Abläufe soll das Framework ermöglichen?

[F2.2] Welche Architekturmuster bieten sich an?

[F3.1] Welche Feature aus [F.1.2] werden von NIs Python-Modulen gedeckt?

[F3.2] Welche bereits existierenden Python-Bibliotheken und Module sind für die Umsetzung geeignet?

[F4.1] Welche Möglichkeiten gibt es zur Automatisierung von Tests?

[F5.1] Wie sieht der Workflow zum Hinzufügen neuerer NI-Module und damit die Erweiterung des Frameworks aus?

Arbeitspakete

AP1 Beschreibung des bereits vorhandenen DAQmxOOP

- AP1.1 Definition der DAQmxOOP-Features
- AP1.2 Klassen
- AP1.3 Methoden
- AP1.4 Funktionalität
- AP1.5 Features

AP2 Konzeptionierung des neuen Python-Framework

- AP2.1 Definition der entsprechenden Python- Framework-Features
- AP2.2 Identifizierung der übernehmbaren DAQmxOOP-Features
- AP2.3 Abwägung der nutzbaren Architekturmuster
- AP2.4 Klassendiagramme
- AP2.5 Aktivitätsdiagramme

AP3 Test-Unit Klassen

- AP3.1 Definition einzelner Test-Unit Klassen

Zeitplan

Woche	1	2	3	4	5	6	7	8
Paket 1								Puffer
Paket 2								Puffer
Paket 3								Puffer