

Technologie-Evaluierung von Drehgebern und deren Anbindung an eine Java-basierte Potiboard-App

Table of Contents

Einbindung dieser Arbeit in das laufende Projekt <i>FCC Digital Potiboard</i>	1
Rückblick und Motivation	2
Vom alten Potiboard-Prototyp übernommene Technologieentscheidungen und neue Wege.....	2
Sammlung von diskutierten Anforderungen (Stichwortliste).....	3
Noch in Evaluierung stehende alternative System-Architekturen und Technologien	4
Fazit	5

Einbindung dieser Arbeit in das laufende Projekt *FCC Digital Potiboard*

Die hier beschriebene Technologie-Evaluierung dient als Grundlage zur Entscheidungsfindung zu den in der [Abbildung: Status FCC Digital Potiboard](#) markierten *Major Milestones* und *Subprojects / Tasks*.

Status Date 08.07.2022

Project Description		Subprojects / Tasks																						
Development of a digital potentiometer board control for FCC - Focus: UNILAC operation - But also generally as an option for other linear accelerators or any parameters, where a rotating controller makes sense. The project includes the incremental encoder hardware and the software application.		<ul style="list-style-type: none">Specify the system ✓Specification approval-> opentechnology decision for incremental encoder -> in progresstechnology decision software stack-> in progress (FESA, LSA or else)general decision on UI-> open (existing or new app + developing group)build hardware prototype -> openbuild software prototype -> openFAT -> serial production-> open																						
Project Goals		Resource Profile																						
<ol style="list-style-type: none">Replace the outdated UNILAC Potiboard control with a modern versioncompatible with fully digital control room (FCC)Full control system integrationBe ready on time for the move to FCCProduction of relevant spare parts		<table><tr><th>Year</th><th>Estimated costs [k€]</th><th>Personnel [person months]</th></tr><tr><td>2022</td><td>1</td><td>2</td></tr><tr><td>2023</td><td>4</td><td>12</td></tr><tr><td>2024</td><td>15 (initial equipment)</td><td>3</td></tr><tr><td>2025</td><td>10 (spare parts)</td><td>3</td></tr><tr><td>2026</td><td>0</td><td>0</td></tr><tr><td>Sum</td><td>30.000 Euro</td><td>20 person months</td></tr></table>		Year	Estimated costs [k€]	Personnel [person months]	2022	1	2	2023	4	12	2024	15 (initial equipment)	3	2025	10 (spare parts)	3	2026	0	0	Sum	30.000 Euro	20 person months
Year	Estimated costs [k€]	Personnel [person months]																						
2022	1	2																						
2023	4	12																						
2024	15 (initial equipment)	3																						
2025	10 (spare parts)	3																						
2026	0	0																						
Sum	30.000 Euro	20 person months																						
Major Milestones		Risks, Boundary Conditions and Comments																						
Q3/2022	Specification revision and approval	Major Risks: <ul style="list-style-type: none">turnaround time too slow for adequate UNILAC control	Concerned departments: <ul style="list-style-type: none">ACO (FE, AP, IN)OPE (APS)																					
Q4/2022	technology decision – incremental encoder hardware and communication protocol																							
Q4/2022	decision regarding use of control system stack (FESA or LSA or else) to be able to ensure the required performance																							
Q3/2023	Hardware prototype ready	Boundary Condition: <ul style="list-style-type: none">must be functional before move from HKR to FCCexisting potiboard must kept functional until move to FCC is complete	Comments: <ul style="list-style-type: none">to be clarified: Responsible group for hardware maintenance																					
Q1/2024	Software prototype ready																							
Q3/2024	Live test in HKR (e.g. for HEST magnets)																							
GSI Helmholtzzentrum für Schwerionenforschung GmbH																								

Figure 1. Status FCC Digital Potiboard

Rückblick und Motivation

Der vor ca. 6 Jahren entwickelte Potiboard-Prototyp wurde in einem der vergangenen Potiboard-Meetings als unzureichend in seiner Technologieauswahl eingestuft. Insbesondere der Einsatz einer "closed source library" des benutzten Phidget-Mikrocontroller, der zur Verarbeitung der Encodersignale dient, wurde kritisiert. Dessen Treiber läuft zudem nicht im "Userspace" auf dem zu benutzenden Linuxderivat, was aus system-administrativer Sicht ein Nachteil ist.

Eine weitere Anforderung der Zukunft (FCC) **könnte sein**, dass die Drehgeber ihre Inkremente an eine Potiboard-App über eine nicht unerhebliche Entfernung übermittelt werden müssen. Diese Anforderung ist bis zum jetzigen Zeitpunkt noch nicht festgelegt und wurde beim alten Potiboard-Prototyp gar nicht diskutiert.

Vom alten Potiboard-Prototyp übernommene Technologieentscheidungen und neue Wege

- Optische "Rotary Quadrature Encoder" wurden wieder wegen ihrer Signalqualität, Zuverlässigkeit und Verfügbarkeit eingesetzt. Auf kugelgelagerte Modelle wurde diesmal verzichtet (Haptikgründe wegen zu hoher Leichtgängigkeit). Merkmale sind 16-128 Pulse pro 360-Rotation, keine Zahnung, 5 V. 3.3V Modelle waren auf dem Markt nicht erhältlich.

- Beispiel-Encoder sind:
 - Grayhill 63R128
 - Bourns ENA1J-B28-L00128L

Durch den Einsatz von modernen Mikrocontrollern, deren Spannung an ihren I/O Kanälen häufig auf 3,3 V limitiert ist (anstatt 5V), schränkt sich die Auswahl der möglichen Encoder-Modelle deutlich ein. Eventuell müssten die Encoder-Ausgangsspannungen an den Eingängen der Mikrocontroller mit Pegelumsetzern (Level-Shifter) angepasst werden, wenn 5 V Encoder-Modelle eingesetzt werden müssen.

- Es wurden wieder Mikrocontroller evaluiert, die die Inkremente der bis zu acht (8!) Encoder, ohne spürbare Zeitverzögerung, weiterverarbeiten können. Statt des im alten Prototypen verwendeten Phidget-Mikrocontroller (1047) wurden folgende Mikrocontroller stattdessen betrachtet:
 - Raspberry Pi 4
 - STM32H7, STM32F7
 - Teensy 4.1 (Arduino kompatibel)
 - Raspberry Pi Pico

Alle drei Systeme stellen nachbaubare Hardware dar (Ersatzteilversorgung scheint gesichert) und lassen sich mit Open-Source Hardware betreiben. Pro Einheit bewegen sie sich in einem Kostenrahmen von 30-100 US \$. Der Mikrocontroller-Code zur Weiterverarbeitung der Encoder-Inkremente muss bzw. musste in C oder Python geschrieben und gewartet werden.

Sammlung von diskutierten Anforderungen (Stichwortliste)

Lebensdauer und Wartbarkeit Hardware

Die Funktion der eingesetzten Drehgeber und Mikrocontroller muss durch Verfügbarkeit am Markt oder durch Reserveteile-Einlagerung für möglichst mehrere Jahrzehnte mit finanziell überschaubarem Aufwand absicherbar sein.

Lebensdauer und Wartbarkeit Software

Die eingesetzte Software auf Mikrocontroller und auf PotiBoard-App Seite sollte aus möglichst gut gepflegten und verständlichen Open-Source Projekten mit hoher Verbreitung stammen. Dies kann auch Auswirkungen auf die Wahl des Mikrocontrollers haben. Der notwendige selbst geschriebene Source-Code sollte möglichst einfach wartbar sein.

Geschwindigkeit Signalübertragung

Die vom Nutzer über den Drehgeber zum Mikrocontroller und dann in das Java-Programm sollte mind. zwischen 1 ms bis 10 ms (0.1 Hz - 1 Hz) liegen.

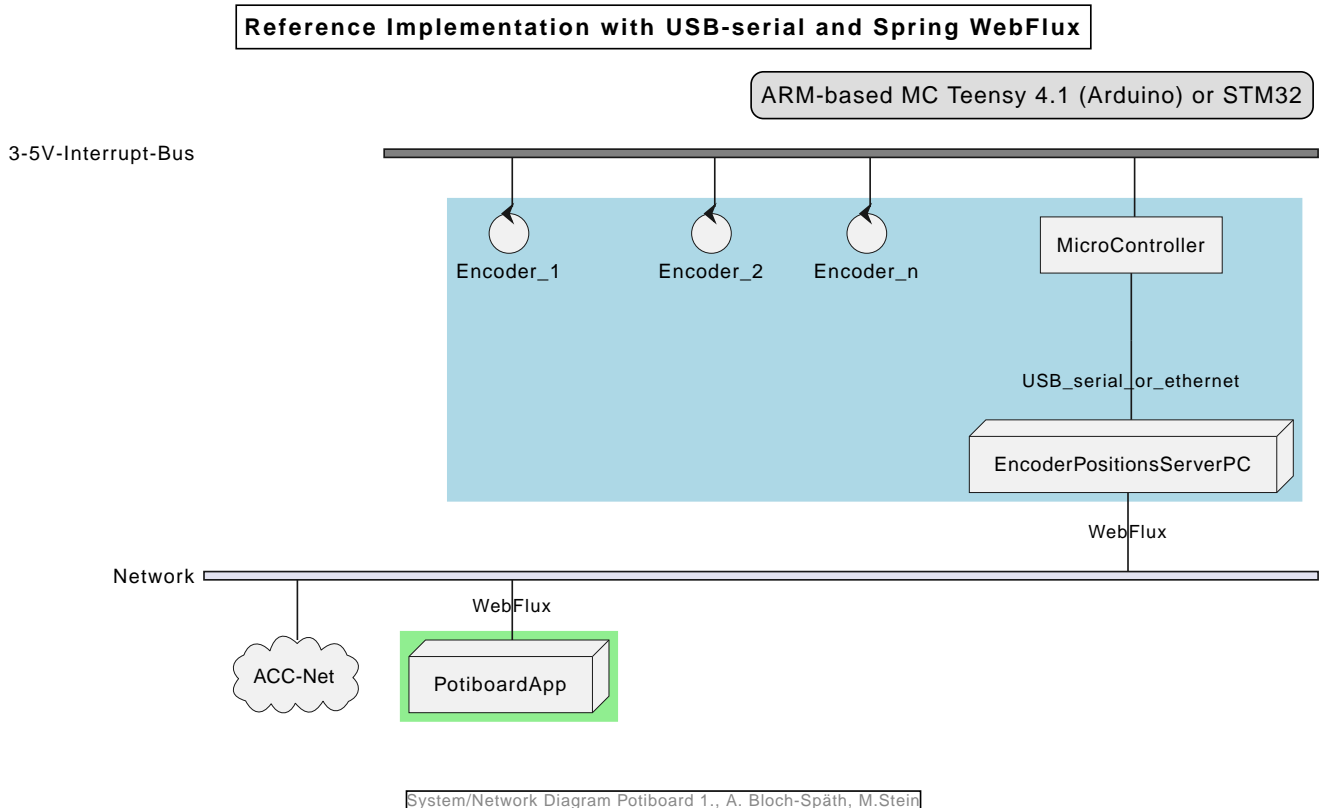


Figure 2. Test-Implementation 1

Es wurde ein Referenzsystem, wie im oberen Bild dargestellt, auf Basis eines Teensy 4.1 Mikrocontrollers entwickelt, der die Inkremente der Encoder in hoher Geschwindigkeit bis in eine Beispiel-JavaFX-Applikation weiterreicht. Das Referenzsystem kann in einem späteren Meeting genauer vorgestellt werden.

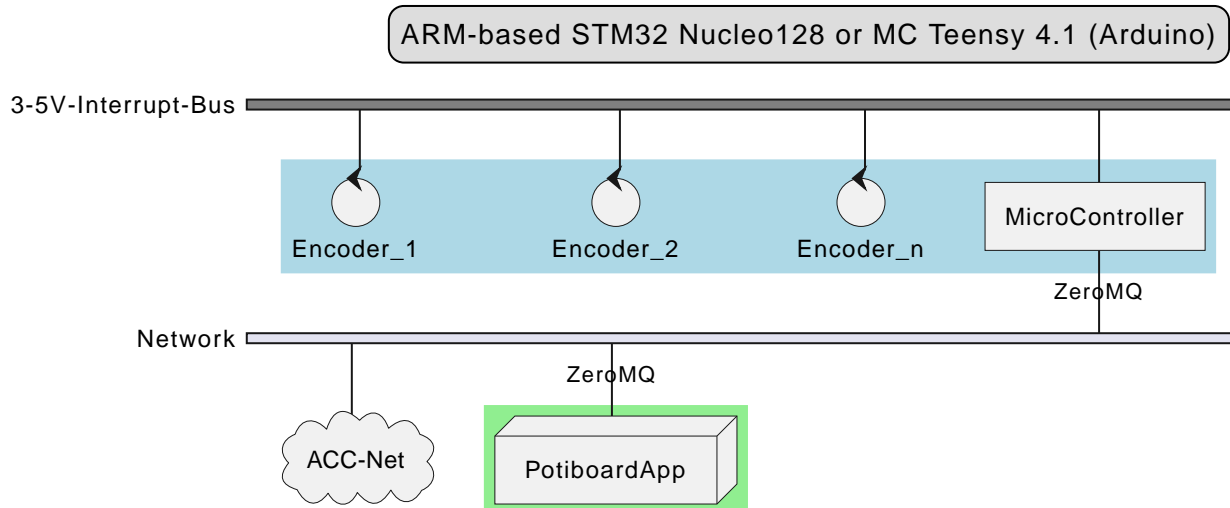
Die im Referenzsystem eingesetzte Datenübermittlungstechnologie basiert auf der Technologie **Spring Webflux** und dem "Reactive Toolkit" **Project Reactor**. Sie wurde ausgewählt, da sie der "GSI Controls Server-Technologie" entspricht, die für die Operating-Applikationen im FCC und HKR eingesetzt werden soll und teilweise schon eingesetzt wird.

Ein Nachteil und in mancherlei Hinsicht sicher auch Vorteil dieser Architektur ist die Einführung eines Webflux-Servers (siehe Bild **EncoderPositionsServerPC**), der ein PC-System mit Controls-konformen OS sein sollte. Es ist also eine Schicht (**Tier**) notwendig, um die Inkremente der verschiedenen Encoder (z.B. im **WebFlux**-Format) zu versenden.

Auf der Habenseite dieser Architektur steht die Anpassbarkeit und Wartbarkeit nach den Richtlinien der Controls Abteilung und damit eine sichere, kontrollierbare Netzwerkkommunikation im ACC-Netzwerk auf lange Sicht und keine Insellösung im ACC-Netz.

Noch in Evaluierung stehende alternative System-Architekturen und Technologien

Implementation with ZeroMQ or similar messaging library



System/Network Diagram Potiboard 2., A. Bloch-Sp  th, M.Stein

Eine vereinfachte Architektur k  nnte der Einsatz einer "leichteren Netzwerk-  bertragungstechnologie" (im Vergleich mit **Spring WebFlux**) mit sich bringen. Ein Kandidat ist zum Beispiel die Technologie **ZeroMQ**, die sich von einem Raspberry Pi 4 aus leicht einsetzen l  sst. Ausstehend sind noch Evaluierungen der Netzwerkkommunikationsm  glichkeiten mit dem leistungsf  higen STM32 Micro-Controller.

Fazit

Das Ziel dieser Technologie-Evaluierungen und Grundlage f  r eine neue Potiboard-Generation ist die Gegen  berstellung von mind. zwei m  glichen L  sungen und deren Vergleich in Bezug auf Faktoren wie Kosten, Laufzeitverhalten, Wartbarkeit, System-Lebenserwartung, usw..