

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	3
Grundlegende zentrale Anforderungen (Stichwortliste)	4
Generelle Hardware-Komponenten-Architekturüberlegungen	4
Komponentengruppen	4
Inkrementignalübertragung über Netzwerk oder USB	5
Noch in Evaluierung stehende alternative System-Architekturen und Technologien	7
Fazit	8

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*.

Project FCC Digital Potiboard

Project Lead S. Reimann

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 -> open • technology decision for incremental encoder -> in progress ↗ • technology decision software stack -> in progress (FESA, LSA or else) • general decision on UI -> open (existing or new app + developing group) ↗ • build hardware prototype -> open • build software prototype -> open • FAT → serial production -> open 																							
Project Goals		Ressource Profile																							
<ol style="list-style-type: none"> 1. Replace the outdated UNILAC Potiboard control with a modern version 2. compatible with fully digital control room (FCC) 3. Full control system integration 4. Be ready on time for the move to FCC 5. Production of relevant spare parts 		<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Year</th><th>Estimated costs [k€]</th><th>Personnel [person months]</th></tr> </thead> <tbody> <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> </tbody> </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: - turnaround time too slow for adequate UNILAC control Concerned departments: - ACO (FE, AP, IN) - OPE (APS)																							
Q4/2022	technology decision – incremental encoder hardware and communication protocol	Boundary Condition: - must be functional before move from HKR to FCC - existing potiboard must kept functional until move to FCC is complete Comments: - to be clarified: Responsible group for hardware maintenance																							
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																								
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 wurde beim [alten Potiboard-Prototyp](#) über eine hohe Integration der beteiligten Komponenten gelöst. Drehgeber wie Potiboard-App, gesteuert über einen Touchscreen, befanden sich in einem Gehäuse mit Netzwerkanschluss.



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, Stückpreis: 100 \$
 - Bourns ENA1J-B28-L00128L, Stückpreis: 60 \$

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 Endcoder-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, Stückpreis: 70 \$
 - STM32H7, STM32F7, Stückpreis: 70 \$
 - Teensy 4.1 (Arduino kompatibel), Stückpreis: 40 \$
 - Raspberry Pi Pico, Stückpreis: 8 \$

Bitte Bilder anhängen

Alle drei Systeme stellen nachbaubare Hardware dar (Ersatzteilversorgung scheint gesichert) und lassen sich mit Open-Source Software betreiben. Pro Einheit bewegen sie sich in einem Kostenrahmen von 6-80 US \$. Der Mikrocontroller-Code zur Weiterverarbeitung der Encoder-Inkremeante muss bzw. musste in C, Python oder Assembler geschrieben und gewartet werden.

Grundlegende zentrale Anforderungen (Stichwortliste)

- Lebensdauer und Wartbarkeit Hardware

Die Funktion der eingesetzten Drehgeber und Mikrocontroller muss durch Verfügbarkeit am Markt oder durch Reserveiteile-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.

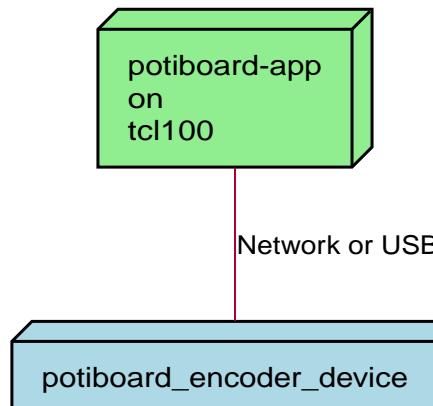
- Geschwindigkeit Signalübertragung der Inkremeante der Encoder

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.

Generelle Hardware-Komponenten-Architekturüberlegungen

Komponentengruppen

General topology for connecting a potiboard encoder device

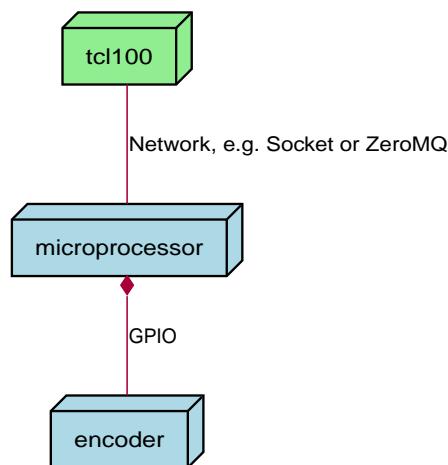


General Deployment Diagram, A. Bloch-Späth, M. Stein

Inkrementalsignalübertragung über Netzwerk oder USB

Network based encoder counter transmission

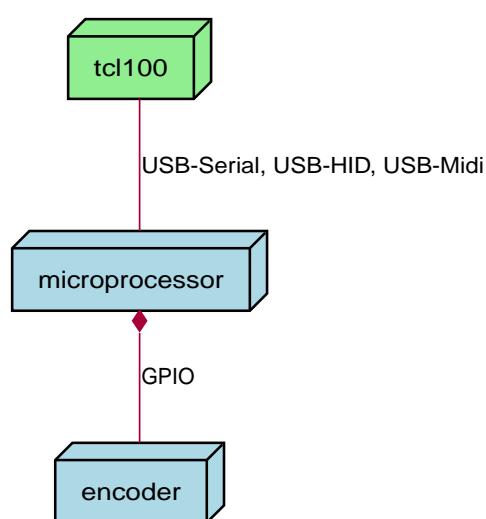
GSI/FAIR 03.11.2022



Deployment Diagram Potiboard Network, A. Bloch-Späth, M. Stein

USB based encoder counter transmission

GSI/FAIR 03.11.2022



Deployment Diagram Potiboard USB, A. Bloch-Späth, M. Stein

USB Nachteil



USB ist auf eine **maximale Kabellänge** von 5m spezifiziert. Mit guten Kabel und/oder Repeatern sind vielleicht bis zu 10m möglich.

USB Vorteil



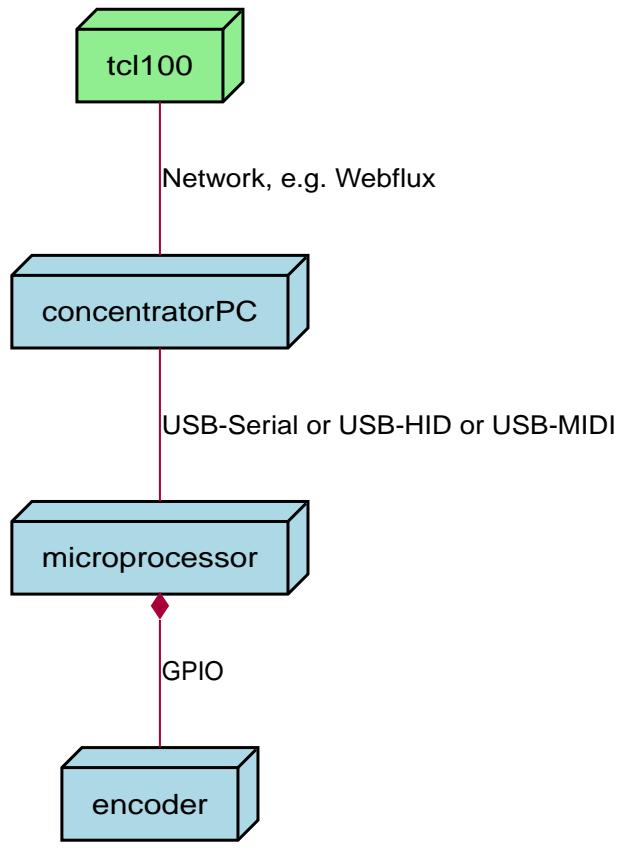
USB ist prinzipiell schneller, die Protokolle haben keinen Adressierungs-Overhead. Vieles ist dadurch einfacher, z.B. müssen Sender (Potiboard) und Empfängeradresse (Potiboard-App) nicht konfigurierbar programmiert werden.

USB Vorteil



USB liefert out of the box ausreichend Strom für Drehgeber und Mikroprozessor. Beim Netzwerk müßte zusätzliche Hardware (z.B. PoE) hinzugefügt werden, wenn 220V Anschluss vermieden werden soll.

Reference Implementation with Spring WebFlux

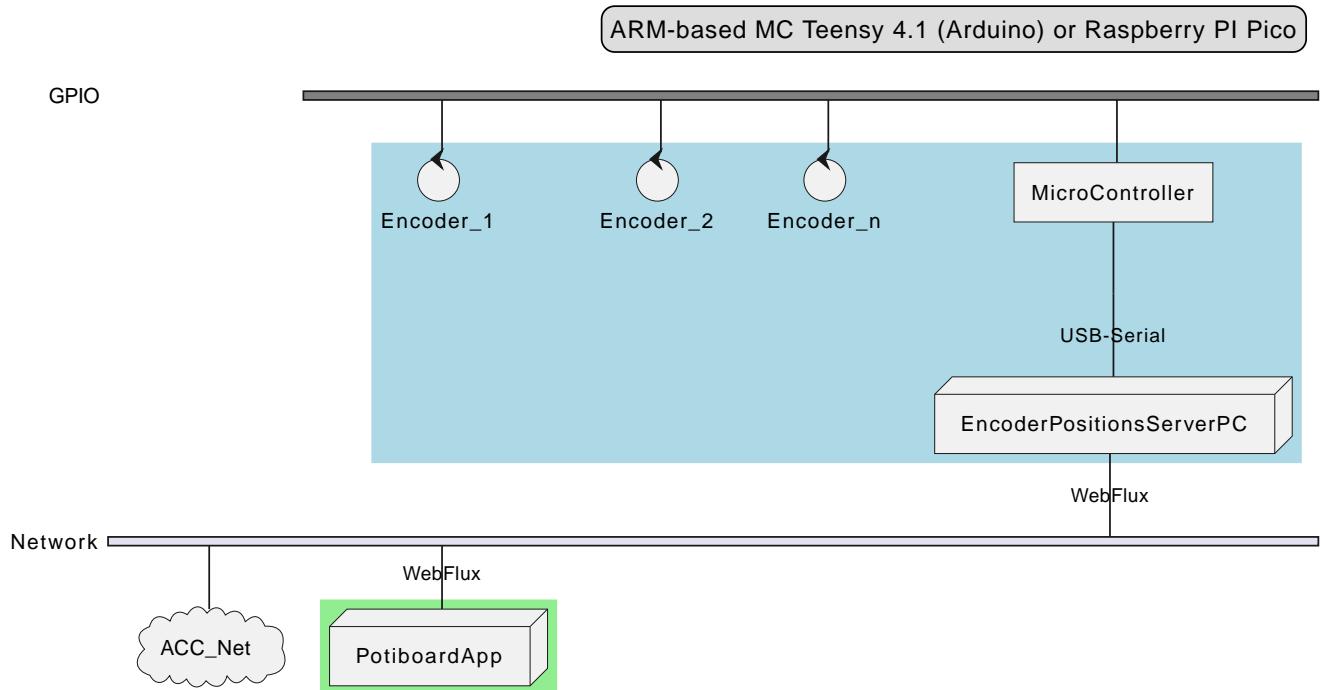


Deployment Diagram Potboard Webflux, A. Bloch-Späth, M. Stein

Figure 2. UML-Komponenten Diagram Network Connection with Spring Webflux

GSI/FAIR 07.11.2022

Reference Implementation with USB-Serial and Spring WebFlux



System/Network Diagram Potboard 1., A. Bloch-Späth, M. Stein

Figure 3. 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 Datenübertragungstechnologie 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.

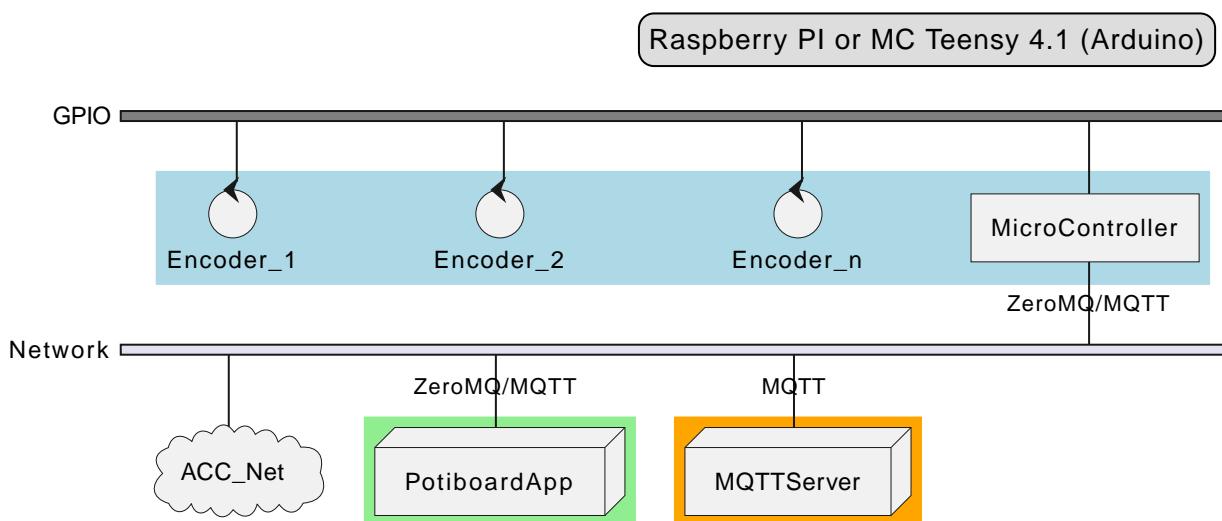
Auf der Habenseite dieser Architektur steht die Anpassbarkeit und Wartbarkeit nach den Richtlinien der Controls Abteilung und damit eine sichere, kontrollierbare Netzwerkkommunikation im ACC-Netzwerk auf lange Sicht und keine Insellösung im ACC-Netz.

Noch in Evaluierung stehende alternative System-Architekturen und Technologien

Test-Implementation 2

GSI/FAIR 06.11.2022

Implementation with ZeroMQ or other messaging libraries



System/Network Diagram Potiboard 2., A. Bloch-Späth, M. Stein

Eine vereinfachte Architektur könnte der Einsatz einer "leichteren Netzwerk-Übertragungstechnologie" (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 lässt.

Fazit

Das Ziel dieser Technologie-Evaluierungen und Grundlage für eine neue Potiboard-Generation ist die Gegenüberstellung von mind. zwei möglichen Lösungen und deren Vergleich in Bezug auf Faktoren wie Kosten, Laufzeitverhalten, Wartbarkeit, System-Lebenserwartung, usw..