

PSI Center for Neutron and
Muon Sciences

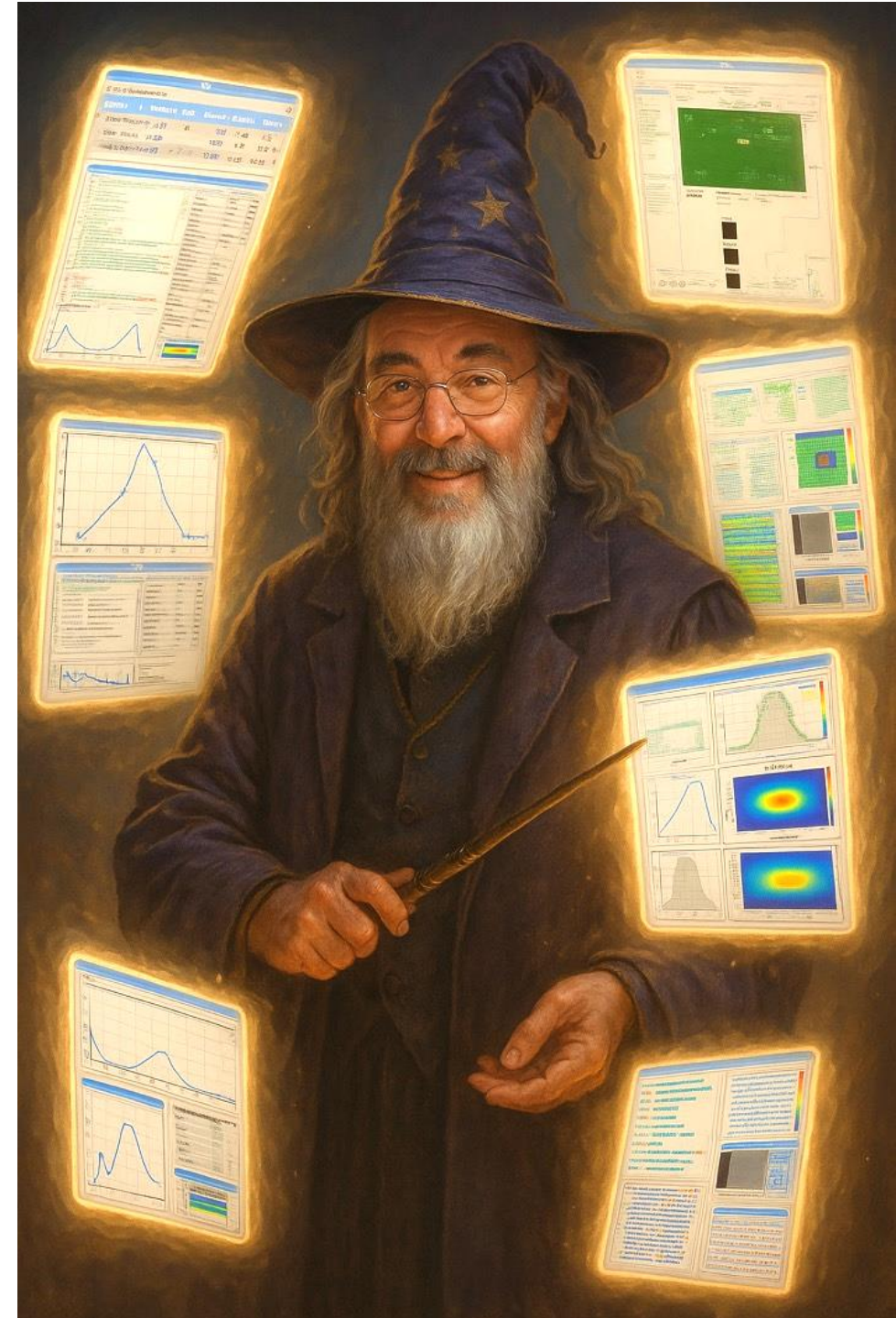
PSI General Beam Profile Monitor

Kick-off Meeting Jan 30th, 2026

Stefan Ritt
Muon Physics Group, LTP

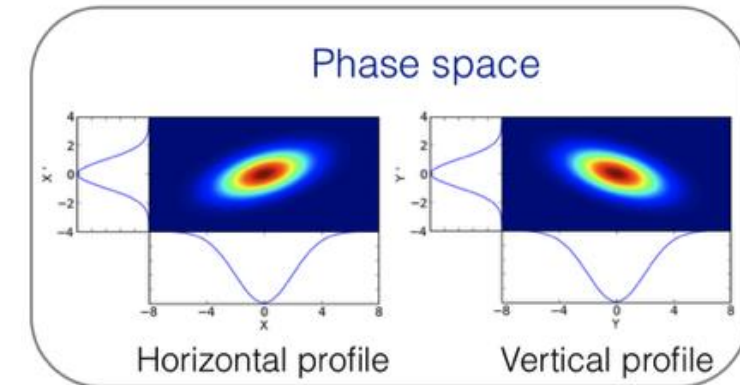
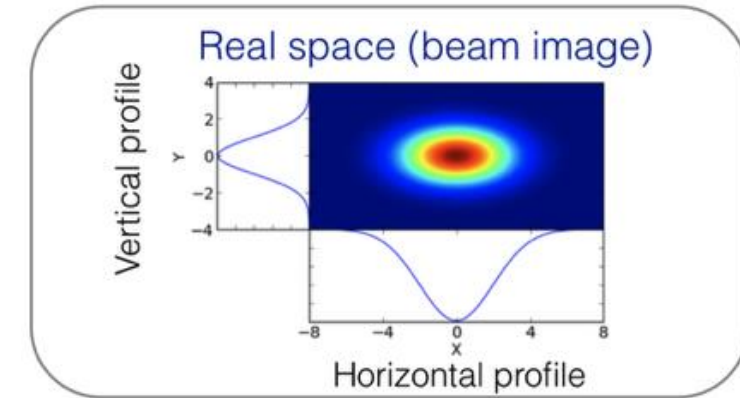
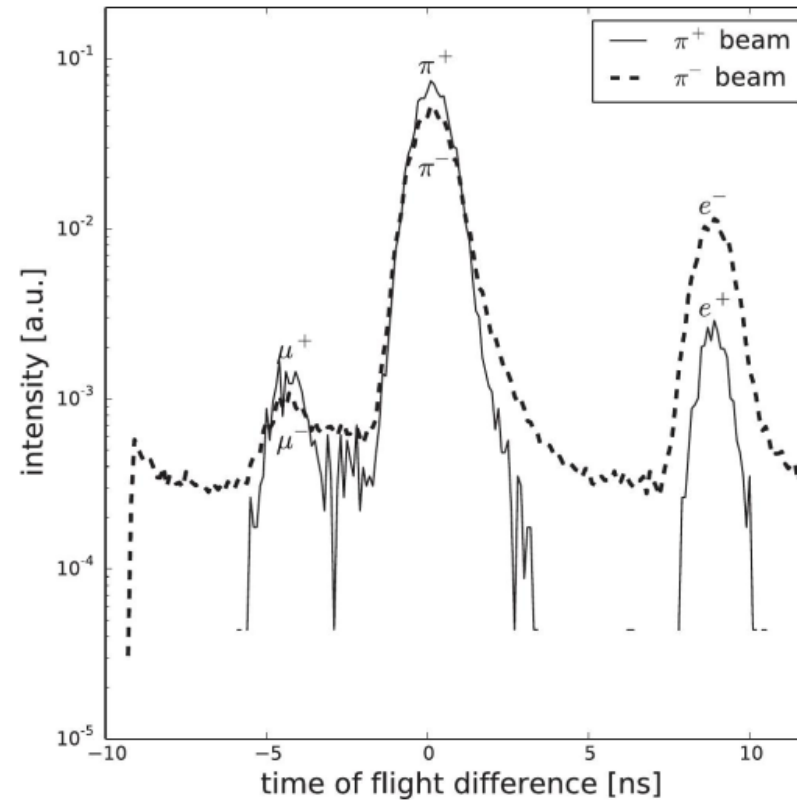
Idea of this meeting

- 1 **Collect requirements** for a general-purpose beam profile monitor
- 2 **Review** existing solutions
- 3 Discussion about a **path forward**



General Requirements

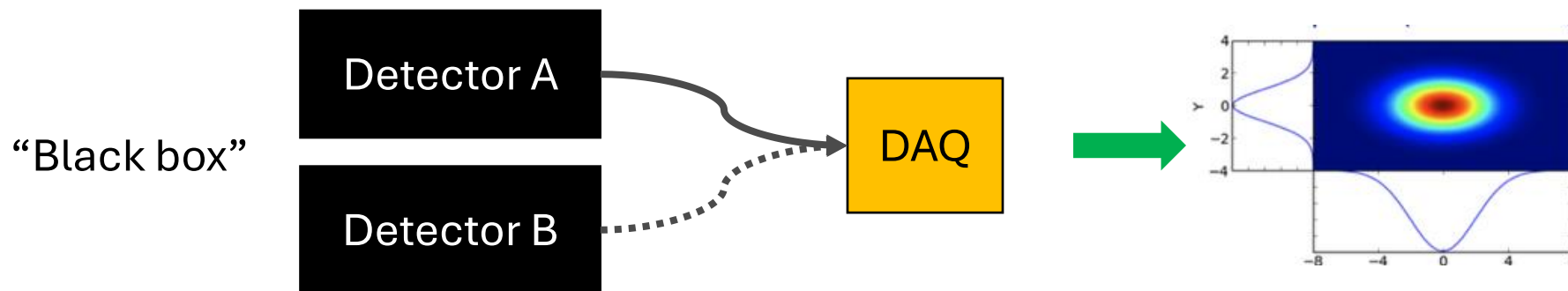
- Beam profile **X/Y**
 - Position
 - Width
 - **Phase space**
- Beam **intensity**
 - Hz to GHz (HIMB)
- Particle ID
 - **ToF** on \sim ns level to distinguish e, μ , π , (p)
 - **dE/dx**



Consolidation of existing solutions

- **Many** beam profile monitor **exists**: “**Pill**”-based scanners, **Scintillator**-based, **Pixel**-based, **GEM**-based, ... → will review them today
- Each solution has **its own** hardware / software / documentation (?) and typically needs an **expert** to install and run it → waste of work-power

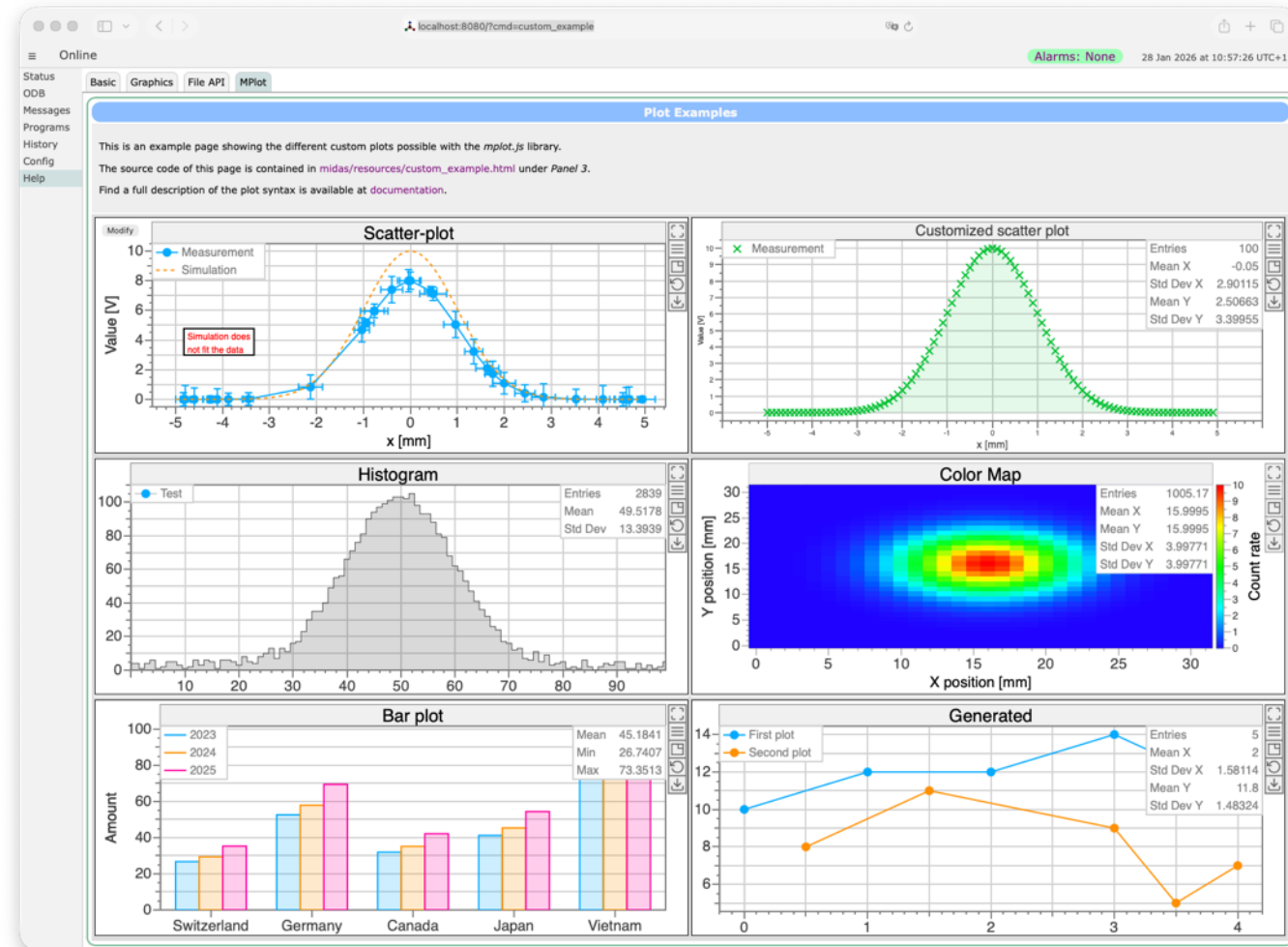
What about consolidating solutions for an easy-to-use, general-purpose device?



My Vision: Software



- **MIDAS** for detector readout
- **Established** PSI system (MSetPoint)
- Modern **web interface**
- **Sequencer** (Optima, ...)
- Software should come **pre-installed** on the beam profile monitor
- Easy to use, **plug-and-play**
- User Interface should be **independent** of actual detector → quick learning
- How to achieve that?



My Vision: Hardware

- **Avoid** Windows & Labview
- **Cheap** & Compact
- Detector readout & **Slow Control** (HV, Temperature, ...)
- Proposal: **RevolutionPi**



RevolutionPi

<https://revolutionpi.com> (KUNBUS, Denkendorf, Germany)

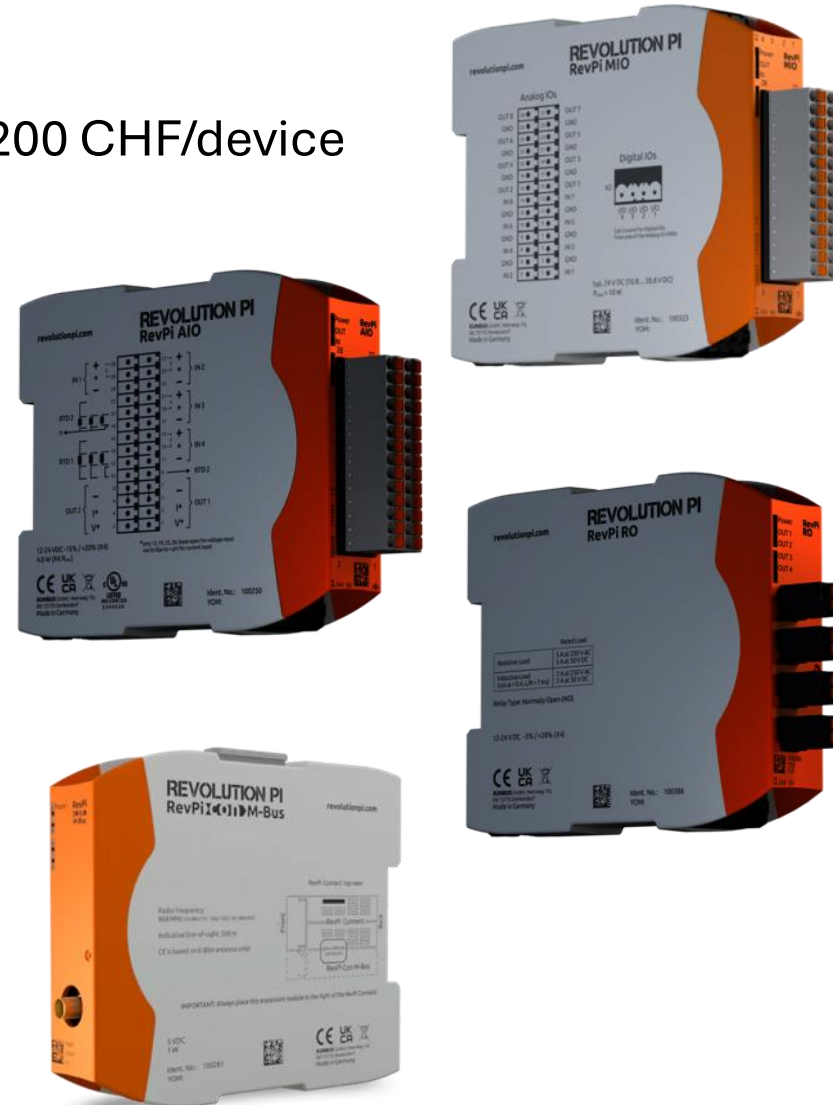
- **Raspberry Pi** based extendable control units
- **Digital, Analog** (+-10V, 4-20mA, PT1000), **Relais** expansion units (up to 80 IOs), **USB**, Ethernet
- Standard Linux with Real Time extension (cycle time <math><10\text{ ms}</math>) “Soft Real Time”
- Can run **MIDAS** natively, can read out **WaveDREAM** boards
- **Open-Source** philosophy (schematics can be downloaded, software under github)
- KUNBUS founded in 2008, **1 million devices** produced, including nuclear power plants



RevolutionPi Devicec

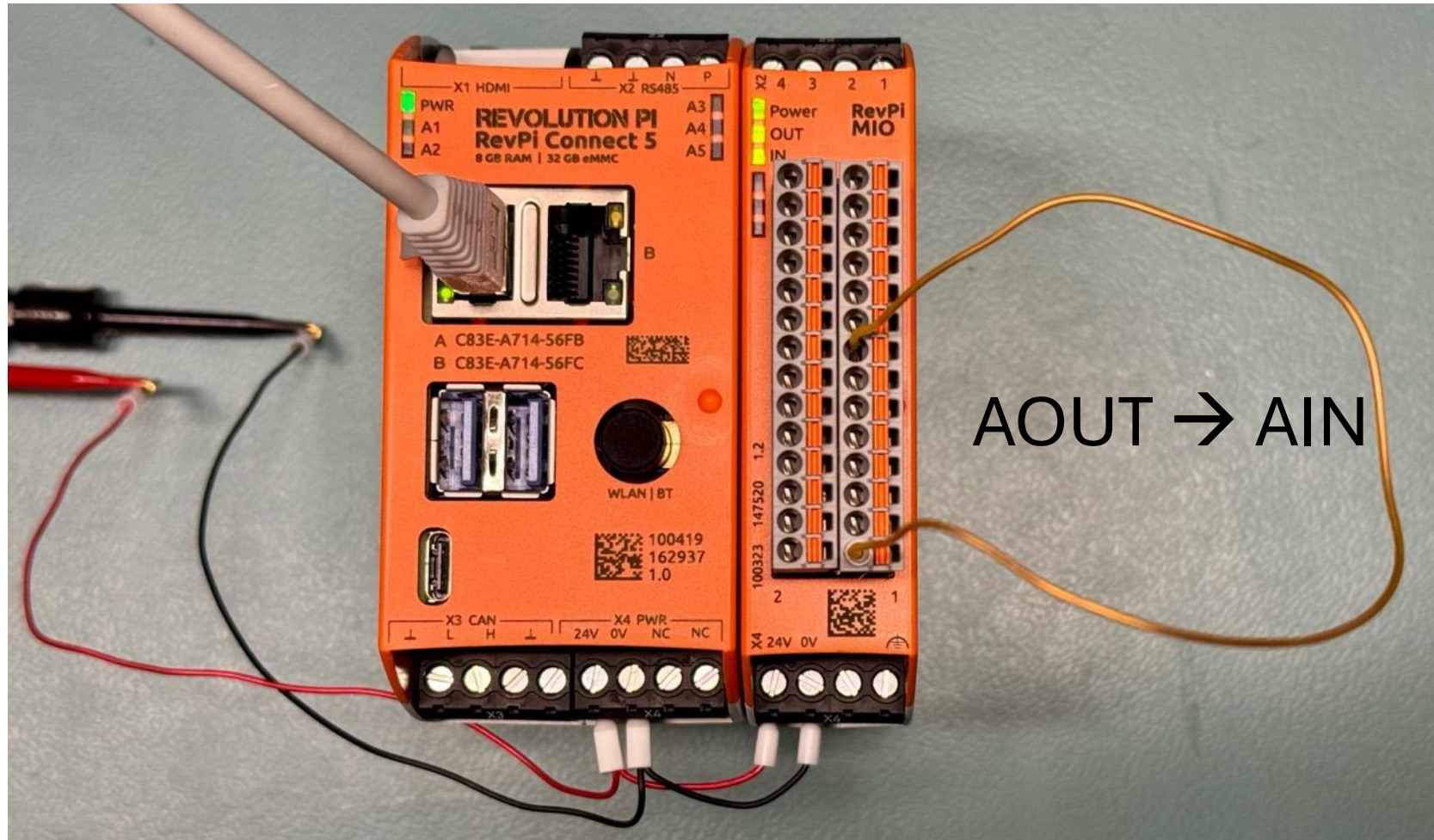
- **Mixed Analog / Digital IO**
 - 0...10V In/Out
 - Digital In/Out
 - PWM / Frequency In/Out
- **Analog IO**
 - +-10V, +-5V, 0...20mA, PT1000
- **Relais 4x**
- **Gateway**
 - Profinet
 - EtherCAT
 - Profibus
 - CAN
 - M-Bus

~200 CHF/device



Controller I/O

24V



AOUT → AIN

Node-RED

<https://nodered.org>



The screenshot displays the Node-RED web interface in a browser window. The address bar shows the URL `mu3erevpi02.psi.ch:41880/#flow/0f200966f2888eb5`. The interface includes a left sidebar with node categories: storage, Revolution PI, and dashboard 2. The main workspace shows a flow with the following components:

- A button labeled "Slider generates analog output" connected to an "Analog Output" node.
- An "Analog Output" node connected to a function node labeled "V -> mV".
- The "V -> mV" function node connected to a "Pin:AnalogOutput_1" node.
- A "Pin:AnalogInput_1" node connected to a function node labeled "mv -> V".
- The "mv -> V" function node connected to three output nodes: "debug 1", "gauge", and "chart".

An "Edit function node" dialog is open for the "V -> mV" node, showing the following code:

```
1 msg.payload = Number(msg.payload) * 1000.0;  
2 return msg;
```

The right sidebar shows a "debug" console with a list of messages:

```
1/6/2026, 8:34:06 AM node: debug 1  
AnalogInput_1 : msg.payload : number  
5.01  
1/6/2026, 8:34:06 AM node: debug 1  
AnalogInput_1 : msg.payload : number  
5.011  
1/6/2026, 8:34:06 AM node: debug 1  
AnalogInput_1 : msg.payload : number  
5.01  
1/6/2026, 8:34:06 AM node: debug 1  
AnalogInput_1 : msg.payload : number  
5.011  
1/6/2026, 8:34:06 AM node: debug 1  
AnalogInput_1 : msg.payload : number  
5.01  
1/6/2026, 8:34:06 AM node: debug 1  
AnalogInput_1 : msg.payload : number  
5.009  
1/6/2026, 8:34:06 AM node: debug 1  
AnalogInput_1 : msg.payload : number  
5.011  
1/6/2026, 8:34:06 AM node: debug 1  
AnalogInput_1 : msg.payload : number  
5.009  
1/6/2026, 8:34:06 AM node: debug 1  
AnalogInput_1 : msg.payload : number  
5.01
```

Graphical Output



mdev_revpi MIDAS device driver

Slow Control:

```
#include "mdev_revpi.h"

...
auto revpi = new mdev_revpi("RevPi");

revpi->add_output("RevPiLED", "LED");
revpi->add_output("AnalogOutput_1", "DAC1", RPI_ANALOG, "Volt");
revpi->add_input("AnalogInput_1", "Voltage1", RPI_ANALOG, "Volt",
    [](float x) -> float {return x - 0.01;});

mdev_table.push_back(revpi);
```

WaveDREAM readout via Ethernet

The screenshot shows the MIDAS online interface with the following sections:

- Run Status:** Run 5, Start: Tue Sep 09 15:04:42 1997, Stop: Tue Sep 09 15:04:42 1997, Alarms: On, runStatusSequencer, Data dir: /home/pi/online/
- Equipment:** RevPi (Status: RevPi Frontend@localhost, Events: 1134, Events[/s]: 0.3, Data[MB/s]: 0.000)
- Logging Channels:** Channel #0: Lazy Label, Events: 0, MB written: 0.000, Compr.: 0.0%, Disk Level: 0.0%
- Clients:** mhttpd [localhost], RevPi Frontend [localhost], Logger [localhost]

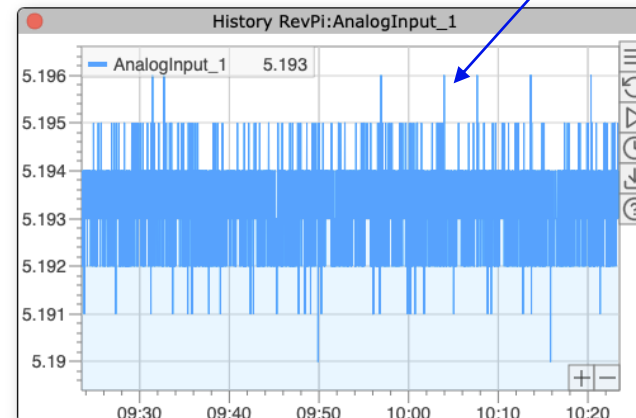


Stand-alone MIDAS running on RevPi

RevPi

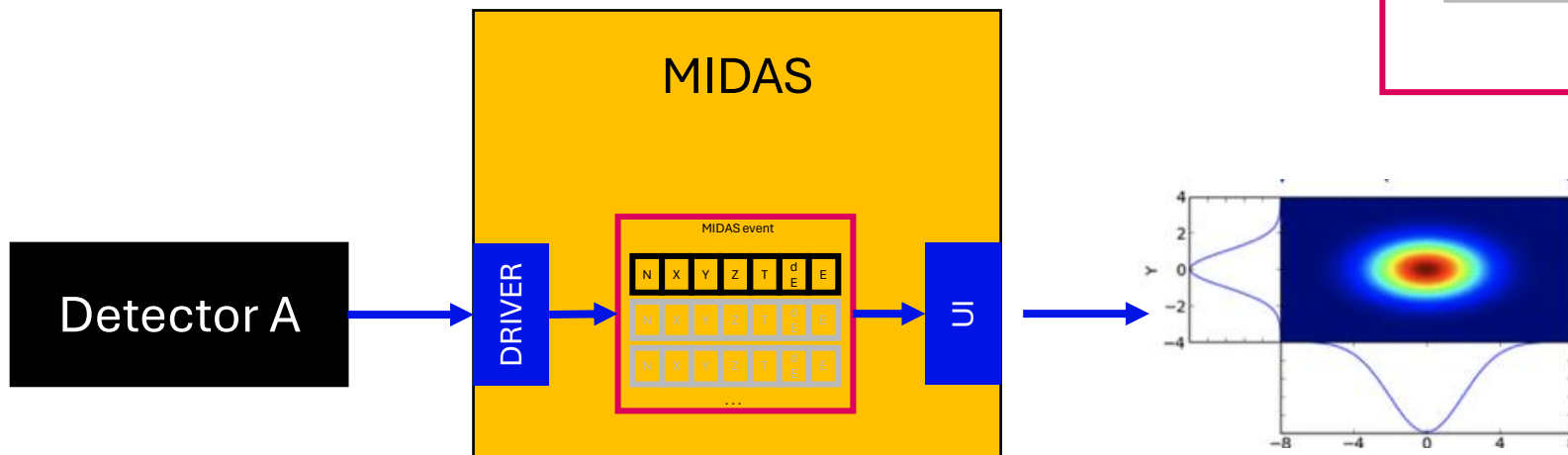
Save Load Export Import ODB

#	Name	RevPi	HST
0	RevPiLED	0	~
1	AnalogOutput_1	5.192 Volt	~
2	AnalogInput_1	5.193 Volt	~



My vision: Detector integration through common data format

- N: Number of particles
- X,Y,Z: Position in **m**
- T: Time in **s**
- dE/E: Energy in **MeV/c²**



Mapping of different detectors

- **Pixel Detector**

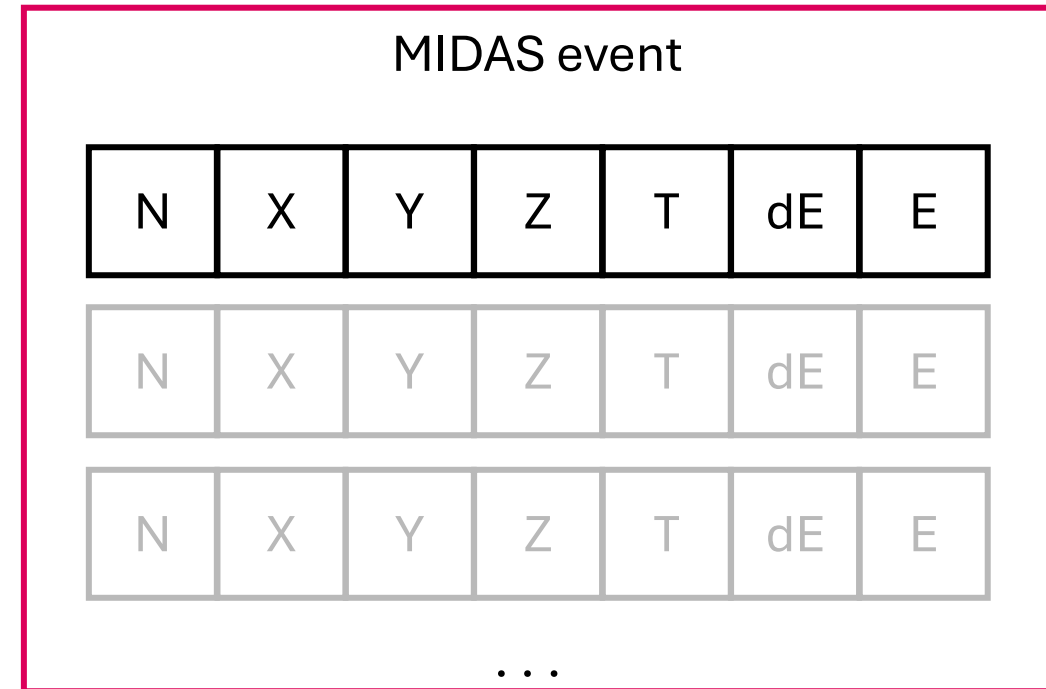
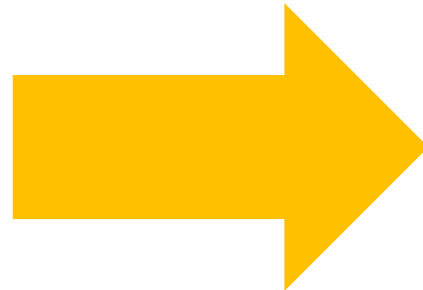
- $N = 1$
- $X/Y = \text{Pixel } X/Y + \text{offset}$
- $Z = \text{position(s)}$

- **Beam Scanner**

- $N = \text{count rate}$
- $T = \text{unused}$
- $X/Y = \text{motor}$
- $Z = \text{position}$

- **Sandwich Detector**

- $X/Y = \text{Pixel } X/Y + \text{offset}$
- $T = \text{scintillator time}$
- $dE/E = \text{scintillator light}$



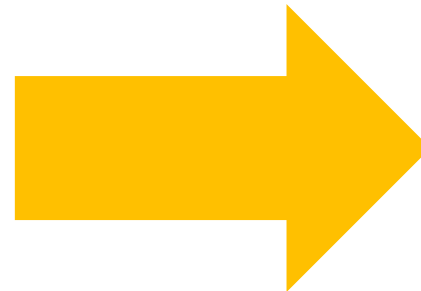
MIDAS event

N	X	Y	Z	T	dE	E
---	---	---	---	---	----	---

N	X	Y	Z	T	dE	E
---	---	---	---	---	----	---

N	X	Y	Z	T	dE	E
---	---	---	---	---	----	---

...



- **Profile plots**
 - X/Y
 - Cut on Particle ID
- **Phase Space / Divergence**
 - $X1-X2 / X$
 - $X1-X2 / Y1-Y2$
- **Beam rate**
 - $\Sigma N / \text{normalization}$
- **Particle ID**
 - T: ToF
 - dE/E: Energy loss

Analysis and Data presentation can be the same of all detectors

Conclusions

- **Combining** our **efforts** will benefit our community and save double work
- A **general** tool is only useful if **easy-to-use** and well **documented**
- I'm happy to bring in my **experience with MIDAS** and various **hardware**, but **rely on others** for the detector parts

