

Manual for the detector control software

Ken Ueberholz and V. Hannen
Institut für Kernphysik, Uni Münster

March 23, 2022

This is the manual for the software that controls the XUV detector and copper mirror at ESR. This includes control for: 1. the motion of the cathode of the XUV-detector, driven by a Trinamic stepper motor, 2. the motion of the copper mirror, driven by a Trinamic stepper motor, 3. the power supply for the voltages of the cathode and shields of the XUV-detector, 4. the power supplies of the main and secondary coils of the XUV-detector.

Components

Raspberry Pi4 with a Pixtend V2S extension board

All Programs are run from a command shell on a Raspberry Pi4. A Pixtend V2S extension board is used to interface with analog and digital inputs and outputs and offers the possibility to switch relays.

Trinamic stepper motors

Two Trinamic stepper motor (TMCM-1161 V1.0) are used to move the cathode and the copper mirror from the park position towards the beam and offer a set of parameters handling speed, acceleration, end switches and more.

Coil power supplies

Two EA power supplies are controlled via the XUV detector software allowing to apply currents up to 10 A (primary, cooled coil) and 5 A (secondary coil).

The Mesytec power supply

A Mesytec 4 channel power supply provides the potentials to the shields and cathode and is controlled via the XUV detector software.

Connection of the hardware and ring signals

The stepper motors are connected via USB and XLR ports to the detector control box. The drive of the parabolic mirror has to be connected to the **motor1** ports while the drive of the XUV detector has to be connected to the **motor2** ports.

The connection diagram of the detector control box is shown in figure 1. Trigger signals for movement of the detectors need to be positive TTL on DI_0 (Trigger IN) and on DI_1 (Trigger OUT). The underlying handling of these signals requires the pulses to be at least 100 ms long. The digital outputs DO_0 and DO_1 are set high whenever the XUV or the copper mirrors are not in their parking position. The digital IO signals are transferred to the laser lab via a DSUB9 cable. To interface the signals with the NIM electronics in the lab, an adapter box is provided.

Software

The software consists of two Python scripts. The XUV-control program sets up a PyQt interface and starts most of the sub processes: The stepper motor control, the pixtend board control and also the potentials control. The later two are connected via an MQTT eclipse Mosquitto broker. The copper mirror control program manages the movement of the parabolic copper mirror.

Starting the software

Usually, the user will have to connect to the Raspberry Pi4 first. To do so open a terminal on any Linux GSI computer or on the Linux laptop provided (Password: 2020gwmx). Connect to the Raspberry Pi4 using `ssh -X pi@140.181.91.15` and enter the password: **SPARCcpu**. Subsequently change into the control folder with `cd detector_control_v3` and start the program with: `./XUV-control.py` and afterwards `./copper-control.py`.

The copper control is a secondary program for the copper mirror detector and must always be started after the XUV-detector control software as it is depended on some of its features. While the XUV-control software can operated standalone, the copper mirror control software must always be used in conjunction with the former.

If the motors are not connected correctly the GUI wont start but fail with a connection error.

The GUI

Both programs contain a section to control the respective stepper motors (left side in the XUV-control program). The functionality is described in the following.

- If the motor loses connection during a measurement it will be displayed in the interface and a **Connect** button offers the possibility to try a reconnect.
- The current position of either cathode or mirror is always displayed in mm together with information if the detector is in the parking position or active position. As soon as the motor leaves the parking position the interface displays the label **Active**, otherwise **Parked**.
- The buttons **Active position** and **Park position** move the detector to the designated positions. The default active position is set to move the cathode / mirror completely into the beam position.

- **Set new active-pos** offers the possibility to set a new position the detector moves to when activated.
- **Automation** Toggles if the cathode / mirror movement is controlled manually or by signals from the ESR. If this is turned on the detectors will move in **30 s** after a **Trigger In** signal and move out immediately after a **Trigger Out**.
- **Set new Acceleration** and **Set new speed** set the acceleration and final speed of the stepper motors. The default values are 100 and 2000.
- **left stop reached** and **right stop reached** are shown when ever a end switch is pressed.
- **STOP** stops the movement of the detector immediately. Next to the stop button the current status of the motor is displayed.
- **Ring signals** are displayed at the bottom of the GUI.

The Voltages of the cathodes and the shields can be turned on/off and set individually in the central panel of the XUV control program. The Mesyttec HV module is rather slow and changes might take a few seconds before being correctly set and displayed.

The coils are controlled on the right side of the interface and only offer limited settings:

- The inner coil closer to the beam is labeled **secondary coil** while the outer, cooled coil is labeled **main coil**. Both allow to set the desired current with **Set new current** buttons next to which the suggested default values shown: 2.8 A for the secondary and 9.2 A for the main coil.
- The measured current read back from the power supplies is displayed for both coils. Notable here is that the set values and the actual currents of the power supplies may deviate by ≈ 0.1 A.

Generally the interface should be closed with the **Exit** button or the **x** in the upper right corner. This will cause both the cathode and the mirror to move back to their parking positions. Closing the interface with strg-c might lead to errors and should be avoided. If the GUI crashes the detectors are set to move to the park position but to be sure the GUI should be restarted in such a case.

If a software crash happens the Mesyttec module tends to become unresponsive. In such a case rebooting the Raspberry Pi most of the times fixes the problem. Should this not be successfull, the NIM crate inside the ESR, containing the Mesyttec module should be power cycled.

Additional Buttons like a possibility to define the Park position can be easily integrated. Other parameter like the time delay of the inwards movements have to be set in the .py files but could be easily integrated into the GUI if necessary.

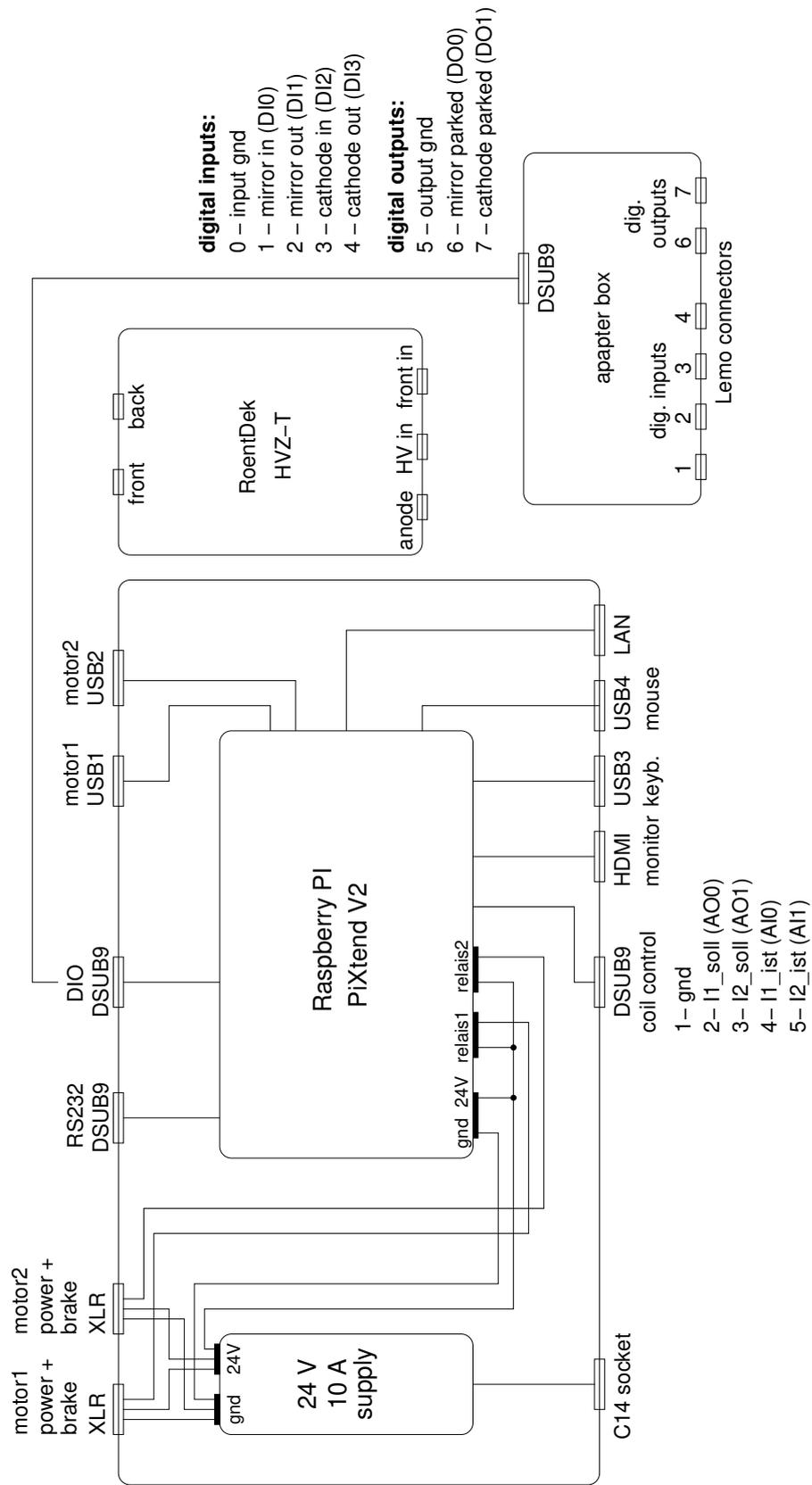


Figure 1: Layout of the detector control box