π^2 Manual

Matthias Schmid

1 February 2021

This document serves as a manual for the current software of the π^2 detector. For more detailed information (e.g. on normalized intensity or perspective correction), refer to my Master thesis[1] or contact me by email[2].

1 Download Software

Download the source code and copy it to some location on the Raspberry Pi. The current version of the pi2 software is available here.

2 Setup

If the IP address of the Raspberry Pi is unknown, connect it to the screen in the physics lab. Assign it a static IP address by editing the `dhcpcd` file[3]. Example: the Raspberry Pi with an orange sticker on one connector port has the address `192.168.110.70`. **Note:** this particular computer will lose all new files after shutdown, so make sure to save your data locally[4].
Once the IP address is known install the Raspberry Pi in the BTL bunker: connect the power and Ethernet cables to the Raspberry Pi as well as the camera ribbon cable from the detector. Connect to the Raspberry Pi from the control room via some graphical remote operation software (for example Microsoft Remote Desktop). Username and password for the SSH connection to the Raspberry Pi are the default (user: pi, pw: raspberry).

3 Start

On the Raspberry Pi, use the terminal to navigate to the software location. Start the application with the command `python3 pi2App.py`. **Note:** the program

[1] M. Schmid: *Development and Optimization of 2D and 3D Beam Detectors at the Bern Medical Cyclotron Laboratory*, University of Bern, 2021.

[2] matthias.schmid@lhep.unibe.ch

[3] Follow this page or one of many other online guides.

[4] Better: acquire some new computers

uses a bunch of libraries which need to be installed[5] on the Raspberry Pi if not yet installed.

4 Software Functions

Start Live View

The image frame gets updated continually with a new image. This is useful to observe a changing beam, e.g. during focusing. Note that the images are in color and not perspective corrected, unlike the images taken as measurement.

Auto / Manual Mode

Choose if the exposure time is set by the camera (auto), or set the desired exposure time in microseconds (manual). The text field "Exposure Time: -" will update to the desired time. **Note:** the text field will update to the actual time used in the capture process, which should be close to the desired time.

Capture Image

This will capture an image. The image is then converted to grayscale, perspective corrected and displayed on the interface. The software tries to fit a Gaussian to the x and y profile of the image. If the fit does not converge, `No optimal parameters found` will be printed in the linux terminal. If the fit converges, the fit parameters are printed. Note that the fit sometimes converges to a very flat Gaussian (i.e. very large sigma). In this case the beam parameters are set to 0.
The profiles and fits are displayed in the graph section. The user should check by eye if the fits are plausible.
Using the fit parameters, the beam center (mean) μ and the $\mathrm{FWHM} = 2\sqrt{2\ln 2}\sigma$ are displayed in mm.
The total intensity and the normalized intensity (which compensates for the gain values) are calculated and displayed. The normalized intensity is used to calculate and display the beam current in nA.

Examine the Image

The image can be examined, e.g. to check for saturation. After a single-click on the image, the brightness value of the clicked pixel is printed in the terminal. By double-clicking, a cross-hair is shown on the image. The brightness values along the cross-hairs are shown on the graph (note that this is different from a beam profile). The brightness of each pixel can range from 0 (black) to 255 (white).

[5] Example guide here. Note that `pip3` has to be used, since the software runs on python 3.x.

Show Fit

If different graphs have been displayed (by clicking on the image), the beam profiles with fits can be shown again by clicking this button.

Save Image

This opens a file dialog where the image currently displayed will be saved. **Note:** If it is not saved, the current image will be lost upon capturing a new image. The saved image will be in grayscale but not perspective corrected. If many images are to be saved in the same folder, the default folder can be changed in line 320 of the source code (`initialdir="*desired filepath*"`) to save the trouble of selecting the folder each time.

5 After a Measurement

Copy the measurement data to your device. This can be done remotely via SSH connection, for example using secure file transfer protocol (sftp).