

Configuration of SAMPA i2c?

Tools: dam.py, plot.py, and plot_zs.py (python scripts)

Data decoding example for now: ReadDat.cc (rewrite of parsing part of plot.py)

In the plot.py, there is a function for configuring SAMPA:

`Fee_reg_write (fee, address, data)`

Usage of this is shown below. When you connect just one FEE, the fee is always "0".

1, To write values to the global registers in SAMPA, what you need to do is:

- 1), Write SAMPA register address to be configure as the data to address 0x0601
- 2), Write data to be sent to the SAMPA register address as the data to address 0x0602
- 3), Write (0x050 & (4 bits chip address)) to address 0x0600

Example here. If you want to write the value of 0x0a to the SAMPA global register address of 0x0e on the 2nd SAMPA chip on the FEE (number runs from 0-7), here is the code:

```
Fee_reg_write (0, 0x0601, 0x0e)
```

```
Fee_reg_write (0, 0x0602, 0x0a)
```

```
Fee_reg_write (0, 0x0600, 0x051)
```

For writing the channel-specific registers of the SAMPA, the procedure has one more step, which is to write channel specific register address to the SAMPA global register address of 0x17, and its value to 0x18 (lower 8 bits) and 0x19 (higher 5 bits), and then write configuration scheme to 0x1a. You should simply do this using the `fee_reg_write()` function. You can see an example in `plot_zs.py` (applying zero-suppression threshold, but it makes decoding almost impossible. Guess what. So please don't do this.)

2, Some specific address associated to specific configuration.

Fee_reg_write (fee, 0x200, data) : This is to enable/disable elinks from SAMPA, effectively set whether or not you receive data from a particular SAMPA. This address is for SAMPA 4-7. Current firmware sets 4-elinks readout per chip. You have to turn on all or neither of 4-elinks of a chip, otherwise, you will lose some channels.

Fee_reg_write (fee, 0x201, data) : Same above but this is for SAMPA 0-3.

Fee_reg_write (fee, 0x300, data): This is to change gain/shaping time. See description in plot.py

Fee_reg_write (fee, 0xa001, 0x01): Issue a software trigger.

The register address used in fee_reg_write() can be found in "register_io.vhd" under tpc_fee_testing_fw/hdl/phy. You could figure out how to read i2c config out from the hdl code.

By Takao Sakaguchi, Mar 11. 2022