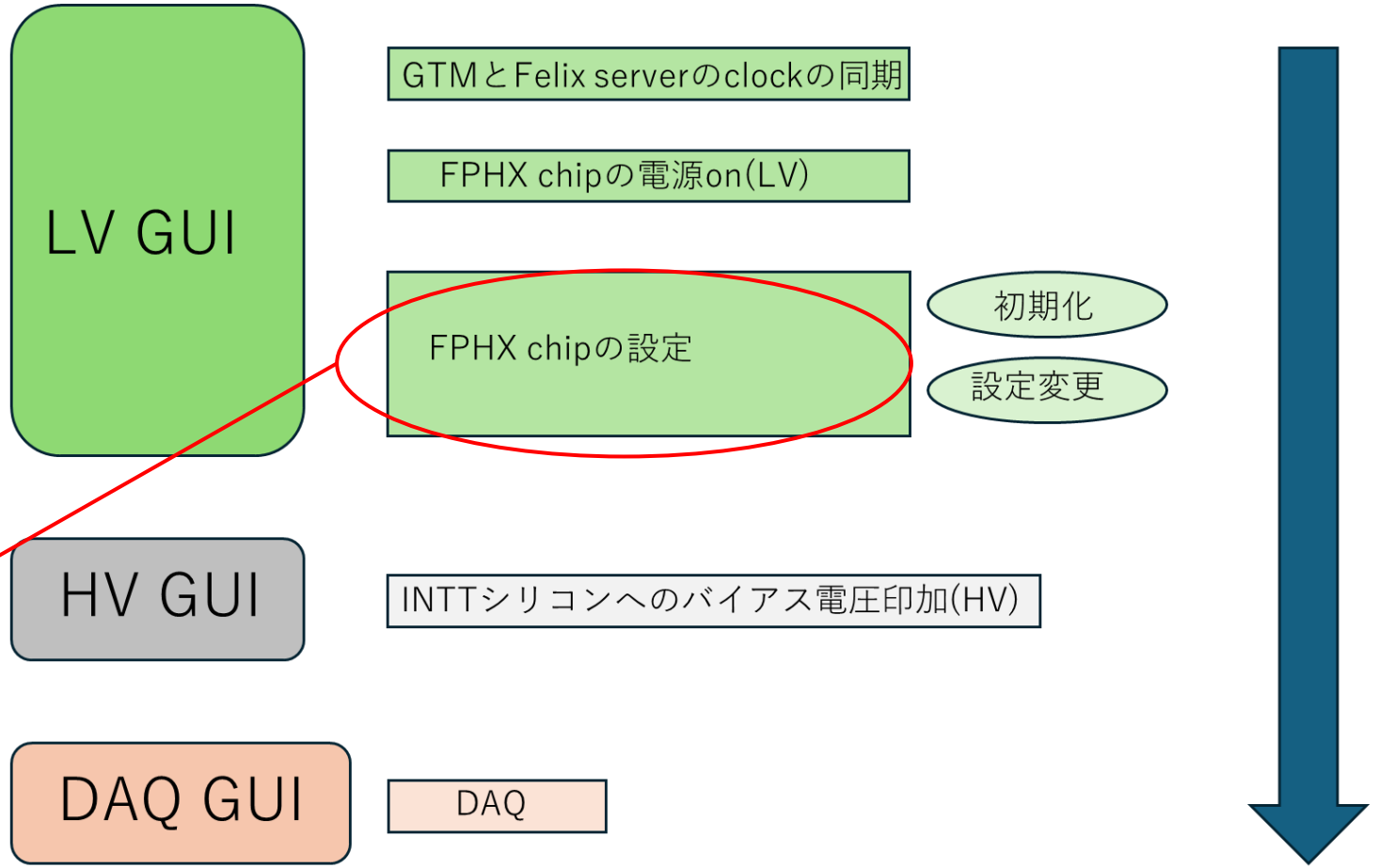
DigConのデバッグ

2025/03/25 菊池陸大

INTTの
オペレーション
手順

- 簡単に

- 設定を変更するときは基本的にLV GUIのfphx chipの設定で走るスクリプトを書き換える。



Fphx chipの設定

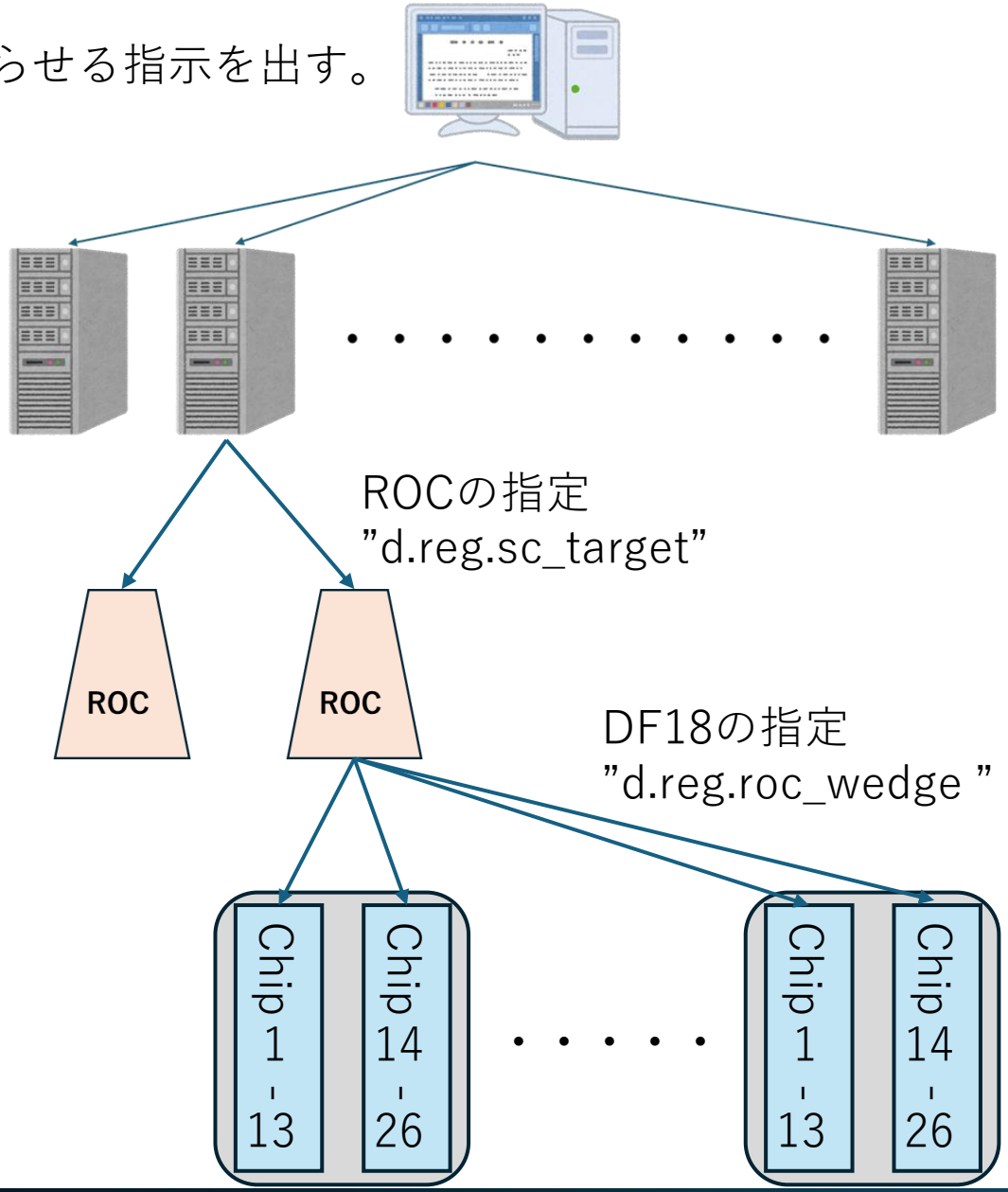
- この設定にはn collision, open time, DAC値, Digital control, ChipやChannelのマスクがある。
- このうちDAC値, Digital control, マスクはSlow Controlというコマンドによって管理されている。

- Slow controlの送信は各felix serverでpythonスクリプトを実行することで行われる。

Slow control送信

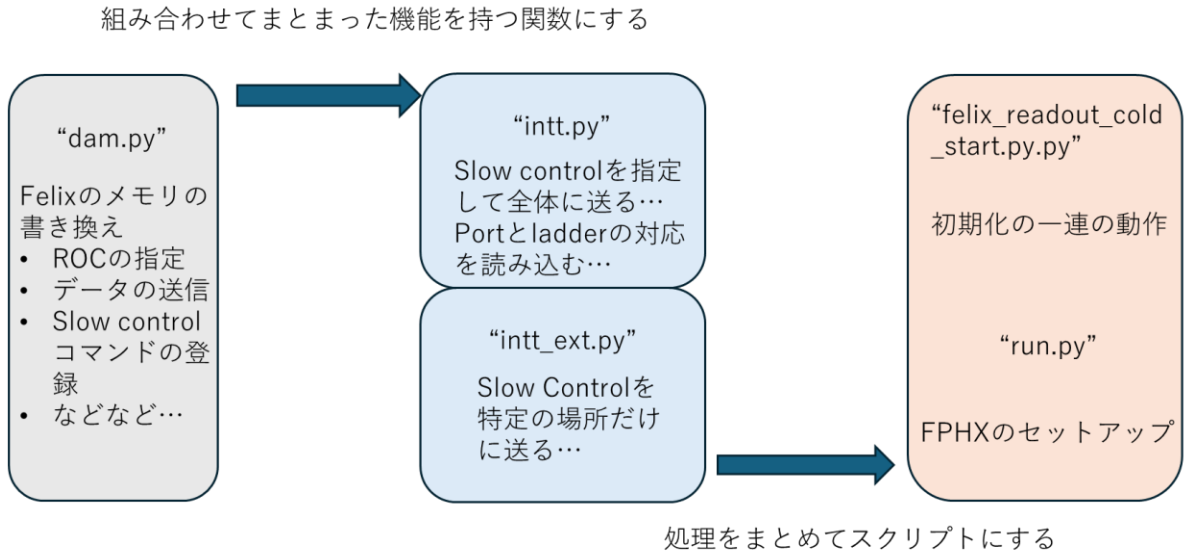
- 8台あるfelix serverのそれぞれから送られる。
- felix1台につき2つのROC
- ROC1つにつき14台のハーフラダー
- ハーフラダー1台につき2つのDF18(Chip1-13 & Chip14-26)

- FelixではROCとDF18によって管理されている。
- “dam.py”を通してこの2つの値を指定することでSlow controlを送る。



Felix serverにおけるスクリプトの実行

- “dam.py”, “intt.py”, “intt_ext.py”をimportして作成する。
  - “dam.py”…felix serverのメモリを書き換えるためのモジュール。
  - “intt.py”…dam.pyを用いて、データの送受信やコマンドの設定など機能ごとに関数にまとめたもの。
  - “intt_ext.py”…intt.pyはRaulが作ったものであり、それ以外の人がdamを用いて機能を付け足すために作ったものの置き場。
- 現在はintt.pyにあるcold startとintt_ext.pyにあるtake dataが順に実行されている。
- 機能を追加する場合はintt_ext.pyに関数を追加し、take dataの中にそれを加える。



- “intt.cold_start()”…fphx chipの初期化を行っている。
- “intt_ext.take_data()”…fphx chipの設定を行っている。

- Fphx chipの設定をいじるならintt_ext.take_data()に機能を追加する。

“intt_ext.take_data()”

- やっていることは、おおまかに
  - N_collisionの指定
  - Open_timeの指定
  - Generalなfphx chipの設定
  - Fphx chipの個別設定
- となっている。

- 今回はfphx chipの個別設定の場所にDigital Controlの個別設定の機能を追加する。

Slow controlの個別送信

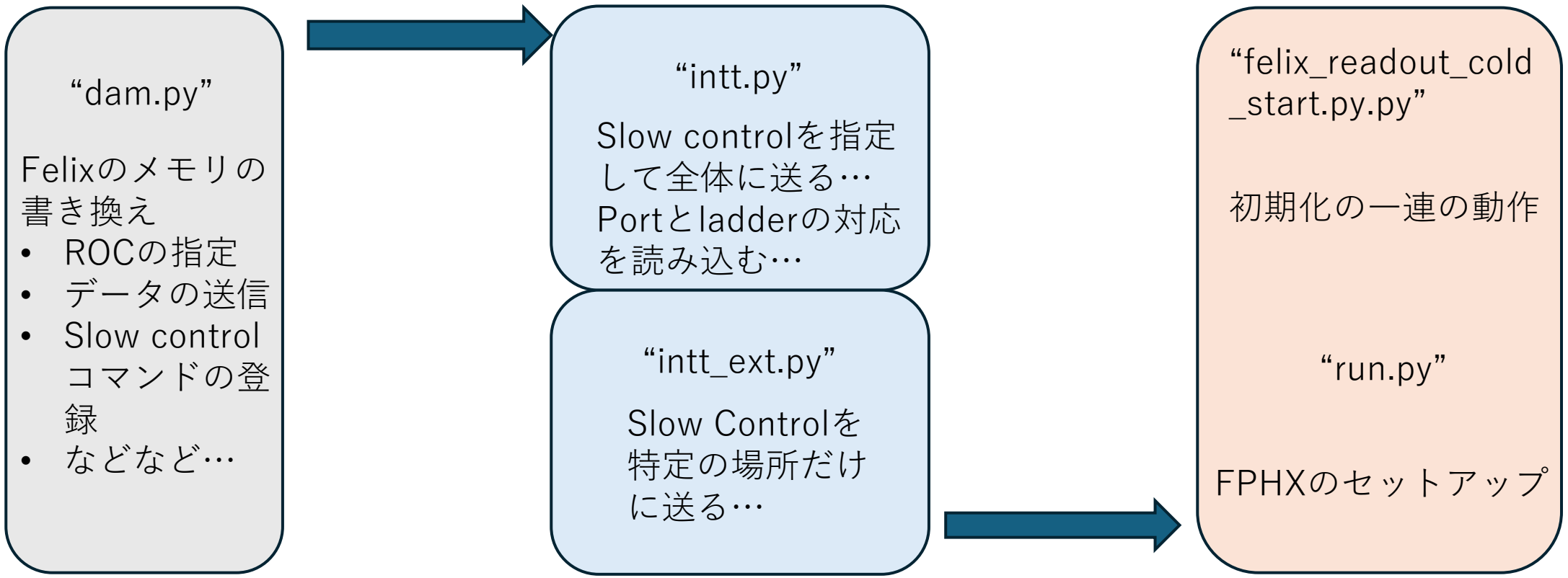
- 既にDAC0をchip単位で個別指定する機能があり、それのパラメータを変えればdigital controlに流用できる。
  - Slow controlを特定のchipに送信する関数は
  - ”intt.write_fphx(d, chip=chip_0_13, register, wedge=wedge, value)”
  - であり、registerをdigital controlを示す値に、valueをdigital controlの設定値にすればよい。

- Custom DAC0 valueを設定する際に作るchipを指定するtxtファイルと同じ様式のものを用意する。

Custom_DigCon_param()

- 個別のchipにslow controlを送る他に、送る値と現在の設定値が同じになるまでslow controlを送り続ける機能がある。
  - これをonにする？
  - それともchipから現在の設定値を読み出す機能を分けたほうが良い？

組み合わせてまとまった機能を持つ関数にする



処理をまとめてスクリプトにする