

BNLでの活動報告

RBRC/Rikkyo

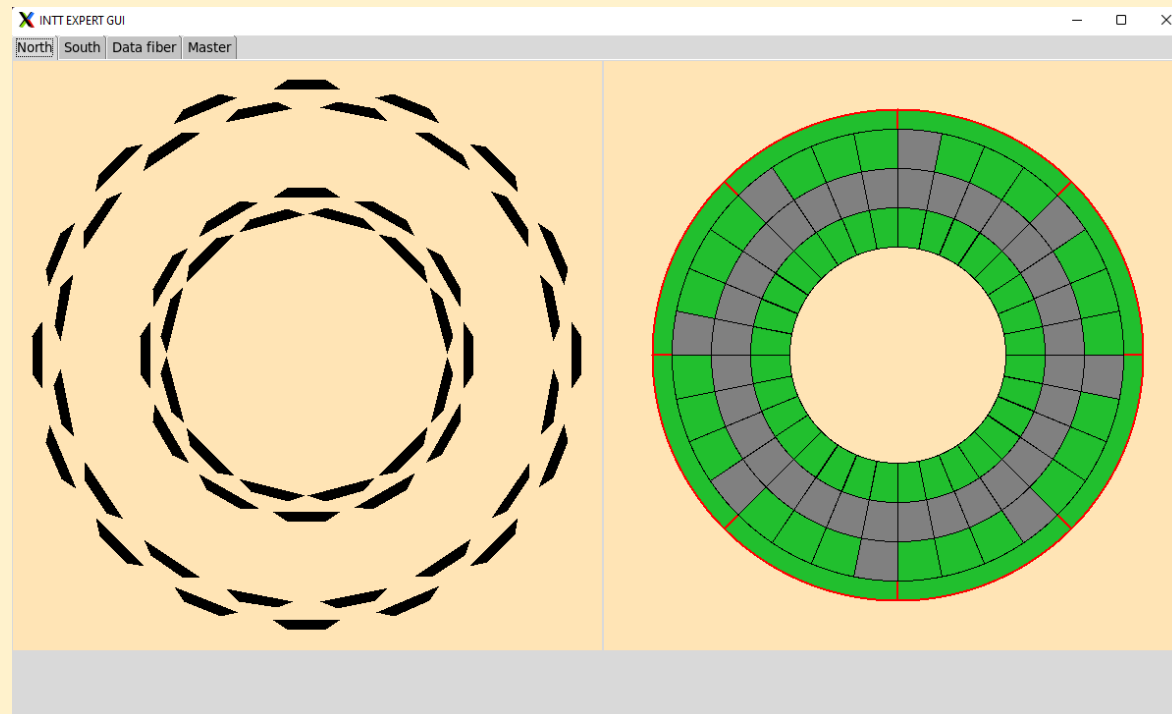
今井ひかる



主に取り組んだこと

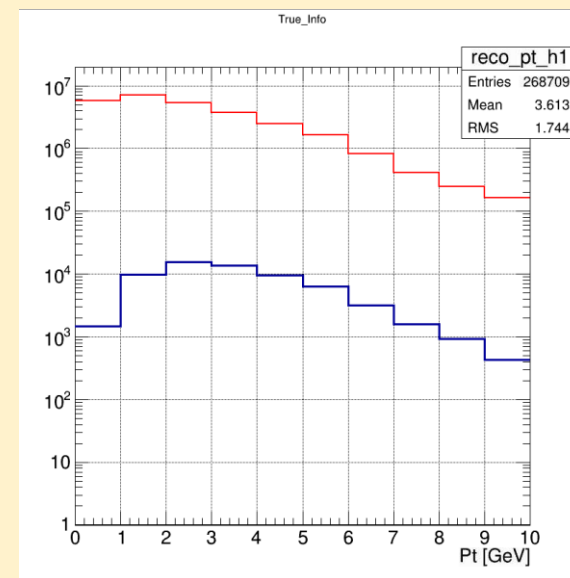
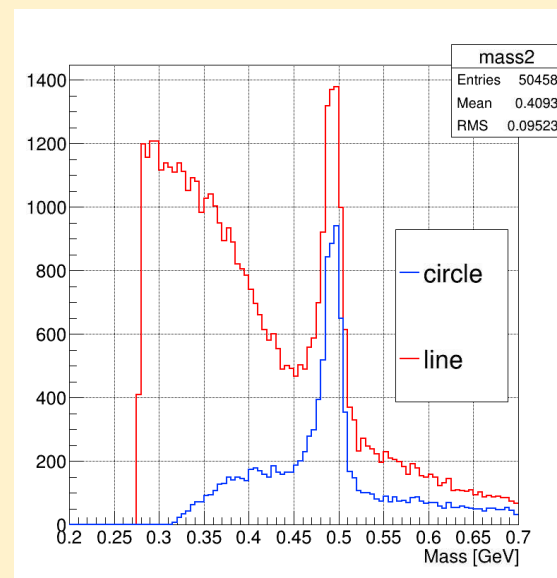
1. Expert-GUIの開発

すべてのラダーに対して、
パラメタの変更などの操作ができるGUIを作成する。



2. PHENIX解析

蜂谷さんありがとうございました。



ExpertGUI開発

- 開発してきたGUIは、ある程度、形を成してきて**試作タイプ**のものが完成した。
- しかし、2ndFelixと通信を行い、ラダーを操作できるようにする機能を**実装出来なかった**。(2nd Felix調整中だった。)

The screenshot shows a window titled "N_B0L011" with a grid of 13 chips (Chip1 to Chip13) at the top. Below the grid is a table with columns for each chip and rows for various parameters. The table is as follows:

| | chip_1 | chip_2 | chip_3 | chip_4 | chip_5 | chip_6 | chip_7 | chip_8 | chip_9 | chip_10 | chip_11 | chip_12 | chip_13 | |
|---------|--------|--------|--------|--------|--------|--------|--------|--------|--------|---------|---------|---------|---------|------|
| R & B | send | back | send | back | send | back | send | back | send | back | send | back | send | back |
| Vref | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| DAC0 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | |
| DAC1 | 25 | 25 | 25 | 25 | 25 | 25 | 25 | 25 | 25 | 25 | 25 | 25 | 25 | |
| DAC2 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | |
| DAC3 | 35 | 35 | 35 | 35 | 35 | 35 | 35 | 35 | 35 | 35 | 35 | 35 | 35 | |
| DAC4 | 40 | 40 | 40 | 40 | 40 | 40 | 40 | 40 | 40 | 40 | 40 | 40 | 40 | |
| DAC5 | 45 | 45 | 45 | 45 | 45 | 45 | 45 | 45 | 45 | 45 | 45 | 45 | 45 | |
| DAC6 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | |
| DAC7 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | |
| N1Sel | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| N2Sel | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| FB1Sel | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| LeakSel | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| P3Sel | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| P2Sel | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| GSel | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| BWSel | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| P1Sel | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| injSel | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| LVDS | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | |

Buttons "Send" and "read" are located at the bottom of the table.

The screenshot shows a window titled "Chip menu N_B0L011_chip9" with a "Channel map" table and a list of control parameters. The channel map is as follows:

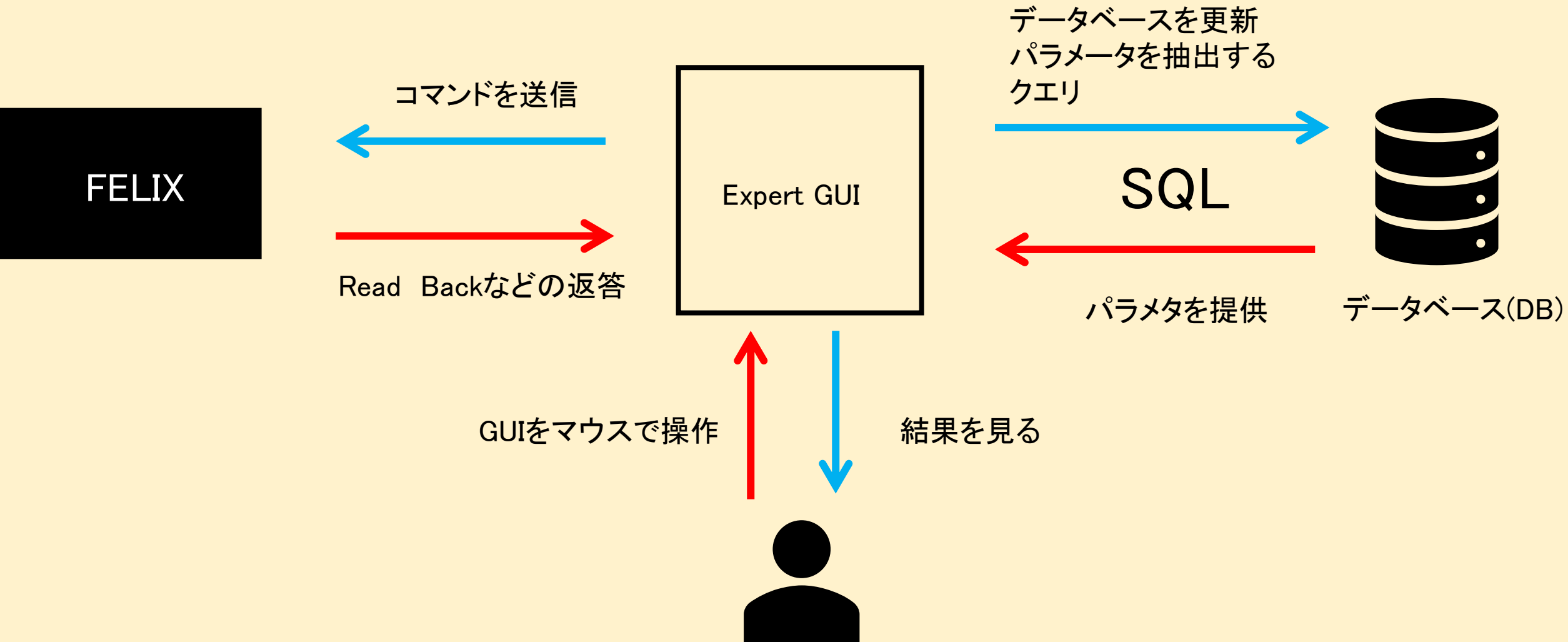
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---------|-------|------|------|---|---|---|---|---|---|
| chip 9 | R & B | send | back | | | | | | |
| Vref | 0 | | | | | | | | |
| DAC0 | 20 | | | | | | | | |
| DAC1 | 25 | | | | | | | | |
| DAC2 | 30 | | | | | | | | |
| DAC3 | 35 | | | | | | | | |
| DAC4 | 40 | | | | | | | | |
| DAC5 | 45 | | | | | | | | |
| DAC6 | 50 | | | | | | | | |
| DAC7 | 55 | | | | | | | | |
| N1Sel | 0 | | | | | | | | |
| N2Sel | 0 | | | | | | | | |
| FB1Sel | 0 | | | | | | | | |
| LeakSel | 0 | | | | | | | | |
| P3Sel | 0 | | | | | | | | |
| P2Sel | 0 | | | | | | | | |
| GSel | 0 | | | | | | | | |
| BWSel | 0 | | | | | | | | |
| P1Sel | 0 | | | | | | | | |
| injSel | 0 | | | | | | | | |
| LVDS | 3 | | | | | | | | |

Buttons "mask all" (red), "unmask all" (green), and "send mask" (cyan) are located below the channel map. Buttons "Send" and "read" are at the bottom of the window.

The screenshot shows a window titled "NE3" with a control panel containing the following buttons:

- Init (cyan)
- FFR (blue)
- FO Sync (cyan)
- Enable RO (blue)
- Latch (cyan)
- BCO Start (blue)
- calib (cyan)
- global start (blue)
- Send Reset (cyan)

やるべきこと



やるべきこと

1

Pythonを用いてGUIを
デザインする
(ある程度終了)



コマンドを送信



Read Backなどの返答



Expert GUI

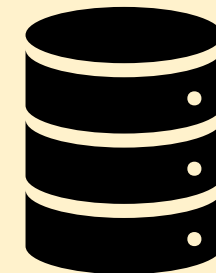
データベースを更新
パラメータを抽出する
クエリ



SQL



パラメータを提供

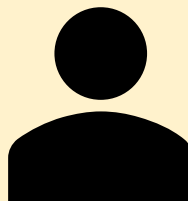


データベース(DB)

GUIをマウスで操作



結果を見る



やるべきこと

1

Pythonを用いてGUIを
デザインする
(ある程度終了)



コマンドを送信

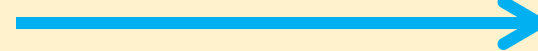


Read Backなどの返答



Expert GUI

データベースを更新
パラメータを抽出する
クエリ



SQL



パラメータを提供



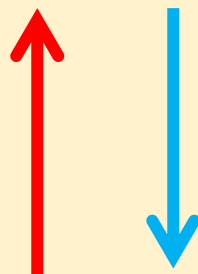
データベース(DB)

2

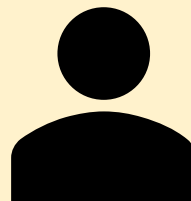
ラウルが用意してくれた関数を用いてFelix、ラダーを操作

- 1.まずは1つのFelixと通信
- 2.次に複数台のFelixと

GUIをマウスで操作



結果を見る



やるべきこと

1

Pythonを用いてGUIを
デザインする
(ある程度終了)



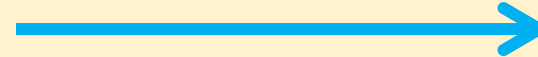
コマンドを送信



Read Backなどの返答

Expert GUI

データベースを更新
パラメータを抽出する
クエリ



SQL



パラメータを提供

データベース(DB)

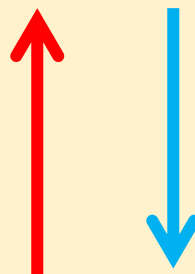


2

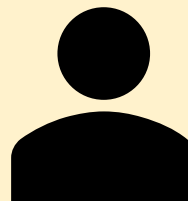
ラウルが用意してくれた関数を用いてFelix、ラダーを操作

1. まずは1つのFelixと通信
2. 次に複数台のFelixと

GUIをマウスで操作



結果を見る



3

DAC値などのパラメータを
保存するデータベースづくり

1. SQLの勉強
2. どのようなテーブル、DBをデザインするか？

やるべきこと

1

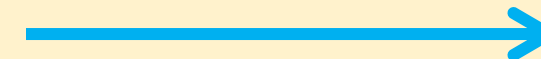
Pythonを用いてGUIを
デザインする
(ある程度終了)

4

ExpertGUIのドキュメント



データベースを更新
パラメータを抽出する
クエリ

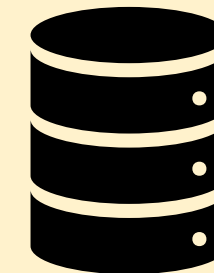


SQL



パラメータを提供

データベース(DB)



FELIX

コマンドを送信



Read Backなどの返答



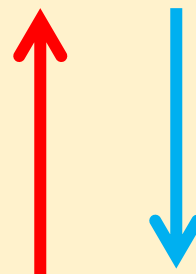
Expert GUI

2

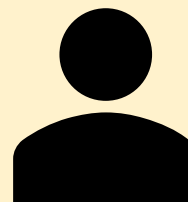
ラウルが用意してくれた関数を用いてFelix、ラダーを操作

1. まずは1つのFelixと通信
2. 次に複数台のFelixと

GUIをマウスで操作



結果を見る



3

DAC値などのパラメータを
保存するデータベースづくり

1. SQLの勉強
2. どのようなテーブル、DBをデザインするか？

まとめ

- Expert GUIのフロントエンドは、ある程度形を成していた。
- 重要なインプリケーションは2nd-Felixの都合上出来なかった。
- 次回のBNL出張ではバックエンドに手を付けて、GUIでFelix、ラダーを操作できるようにできることを目的とする。