

Expert GUI 進捗

今井ひかる

データベースの設計

- データベースを **inttdev@inttdaq** に設立した。

```
[inttdev@inttdaq 18:16:24 INTT_ExpertGUI] $ sh start_postgreSQL.sh
psql (13.8 (Debian 13.8-0+deb11u1))
Type "help" for help.

INTT_DB=# SELECT * FROM ladder_roc_felix_relationship_present WHERE roc_position = 'RC_0' and direction = 'N';
 update_time      | direction | barrel | layer | position_number | roc_position | column_name | station | felix_channel
-----+-----+-----+-----+-----+-----+-----+-----+-----
 1999-03-02 10:00:00 | N         | B0     | L0    | 00              | RC_0        | C           | 2       | 1
 1999-03-02 10:00:00 | N         | B0     | L1    | 00              | RC_0        | D           | 1       | 2
 1999-03-02 10:00:00 | N         | B0     | L1    | 01              | RC_0        | B           | 2       | 4
 1999-03-02 10:00:00 | N         | B1     | L0    | 00              | RC_0        | C           | 3       | 5
 1999-03-02 10:00:00 | N         | B1     | L0    | 01              | RC_0        | A           | 2       | 3
 1999-03-02 10:00:00 | N         | B1     | L1    | 00              | RC_0        | D           | 2       | 6
 1999-03-02 10:00:00 | N         | B1     | L1    | 01              | RC_0        | B           | 1       | 0
(7 rows)
```

データベースの設立方法

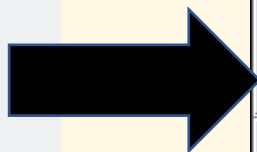
初期値が入っているデータベースを作成するPython スクリプトがある。
\$ python3 create_init_DB.py と実行するだけでテーブルが作られる。

```
614
615 connector = psycopg2.connect(user='inttdev', database = 'INTT_DB')
616
617 cursor = connector.cursor()
618
619
620 # ----- FPHX parameters database -----
621 #create_tabel_and_insert_for_fphxparameters_present(connector,cursor)
622 #
623
624
625 # ----- channel mask status -----
626 #create_tabel_and_insert_for_channel_mask(connector , cursor )
627 #
628
629 #----- felix_info -----
630 #create_tabel_and_insert_for_felix_info( connector , cursor )
631 #
632
633
634 #----- ladder_roc_felix_relationship
635 #create_tabel_and_insert_for_ladder_roc_felix_relationship_present(connector ,cursor )
636 #
637
638
639
640 print("close database !")
641
642 connector.close()
```

create_init_DB.py

GUIのアップデート (Felix menu)

Felix1		
IP address	localhost	
port	8050	
Cold Start		
Slow	Control # 0	True
Slow	Control # 1	False
Ladder Channel # 0		True
Ladder Channel # 1		True
Ladder Channel # 2		True
Ladder Channel # 3		True
Ladder Channel # 4		False
Ladder Channel # 5		True
Ladder Channel # 6		True
Ladder Channel # 7		False
Ladder Channel # 8		False
Ladder Channel # 9		False
Ladder Channel #10		False
Ladder Channel #11		False
Ladder Channel #12		False
Ladder Channel #13		False



felix_1N		
IP address	127.0.0.1	
port	50051	
Slow Control	RC_2N	None
Ladder Channel # 0	C1	None
Ladder Channel # 1	B1	None
Ladder Channel # 2	D2	None
Ladder Channel # 3	D1	None
Ladder Channel # 4	A2	None
Ladder Channel # 5	C2	None
Ladder Channel # 6	C3	None
Slow Control	RC_3N	None
Ladder Channel # 0	C3	None
Ladder Channel # 1	B1	None
Ladder Channel # 2	C1	None
Ladder Channel # 3	D1	None
Ladder Channel # 4	A1	None
Ladder Channel # 5	C2	None
Ladder Channel # 6	D2	None
Cold Start		reload
update		

Ladder channel(Felix channel)とport(A1~D3)の関係は先程のデータベースから取得している。

SC・データファイバのラッチ確認

糠塚さんにcold start関数のテストをBNLで行ってもらった。

データファイバのラッチ確認は別の関数であるverify_latch関数で行うように変更された。

→ Ladder channelがFalseになっている。

The screenshot shows the INTT EXPERT GUI interface. On the left, there are two circular fiber diagrams. The right side displays configuration tables for RC_2N and RC_3N, and control buttons.

Slow Control		RC_2N	True
Ladder Channel # 0	C1		False
Ladder Channel # 1	B1		False
Ladder Channel # 2	D2		False
Ladder Channel # 3	D1		False
Ladder Channel # 4	A2		False
Ladder Channel # 5	C2		False
Ladder Channel # 6	C3		False

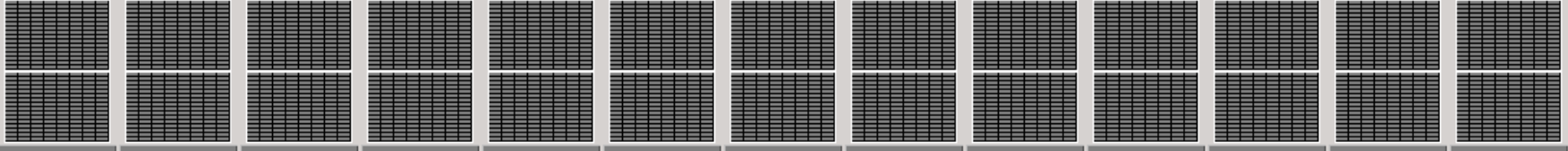
Slow Control		RC_3N	True
Ladder Channel # 0	C3		False
Ladder Channel # 1	B1		False
Ladder Channel # 2	C1		False
Ladder Channel # 3	D1		False
Ladder Channel # 4	A1		False
Ladder Channel # 5	C2		False
Ladder Channel # 6	D2		False

Control buttons: Cold Start, reload, update.

ラダーメニュー

N_BOL000

Chip26Chip25Chip24Chip23Chip22Chip21Chip20Chip19Chip18Chip17Chip16Chip15Chip14



Chip13Chip12Chip11Chip10Chip9Chip8Chip7Chip6Chip5Chip4Chip3Chip2Chip1

side0
side1

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	side0:WILD	
name	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	0
DAC0	15	0	15	0	15	0	15	0	15	0	15	0	15	0
DAC1	30	0	30	0	30	0	30	0	30	0	30	0	30	0
DAC2	60	0	60	0	60	0	60	0	60	0	60	0	60	0
DAC3	90	0	90	0	90	0	90	0	90	0	90	0	90	0
DAC4	120	0	120	0	120	0	120	0	120	0	120	0	120	0
DAC5	150	0	150	0	150	0	150	0	150	0	150	0	150	0
DAC6	180	0	180	0	180	0	180	0	180	0	180	0	180	0
DAC7	210	0	210	0	210	0	210	0	210	0	210	0	210	0
N1sel	6	0	6	0	6	0	6	0	6	0	6	0	6	0
N2sel	4	0	4	0	4	0	4	0	4	0	4	0	4	0
FB1sel	4	0	4	0	4	0	4	0	4	0	4	0	4	0
Leaksel	0	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	0
P2sel	4	0	4	0	4	0	4	0	4	0	4	0	4	0
Gsel	2	0	2	0	2	0	2	0	2	0	2	0	2	0
BWsel	8	0	8	0	8	0	8	0	8	0	8	0	8	0
P1sel	5	0	5	0	5	0	5	0	5	0	5	0	5	0
injsel	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LVDS	3	0	3	0	3	0	3	0	3	0	3	0	3	0

side0:wild read
side0:wild send
side1:wild read
side1:wild send
side0:unmask_all
side0:mask_all
side1:unmask_all
side1:mask_all

FPHXパラメタ

side0:WILD	send back
0	
20	
25	
30	
35	
40	
45	
50	
55	
6	
4	
4	
0	
0	
4	
2	
8	
5	
0	
255	

side0:wild send

side0:WILDの入力欄に値を入れて、side0:wild sendボタンを押すと、side0、chip1~13に値を送信する。

しかし、ラウルのスクリプトにて、どのようにしてsideを指定するか不明だったので今はおそらく、side0とside1の両方、つまり全chip26に値を送信してると思われる。

```
def write_fphx(d, chip, register, wedge, value, cmd=1):
    fphx_word = make_fphx_cmd(chip, register, cmd, value)
    send_fphx_cmd(d, fphx_word, wedge)
    time.sleep(0.01)

def make_fphx_cmd(chipId, regId, cmd, data):
    header = 0x67
    trailer = 0x0
    word = int(0)
    word = header << (32-7)
    word |= (0x1f & chipId) << (32-12)
    word |= (0x1f & regId) << (32-17)
    word |= (0x7 & cmd) << (32-20)
    word |= (0xff & data) << (32-28)
    word |= (0xf & trailer)
    return word
```

まとめ

- 仮のデータベースを作成、データベースをすぐ作れるPythonコードを用意した。
- データベースを反映させたFelixmenuの改良。ラッチの状態をパネルの色に反映させることができた。
- FPHXパラメタの値をワイルドカードを用いて送信できるようにした。(しかし、テストはまでしていない。)

最後に



立教大学 Rikkyo University's Science
理学部 / 大学院理学研究科

🔍 検索

☰ メニュー

2023/02/18 (SAT)

2022年度物理学専攻修士論文発表 会最優秀・優秀発表賞

物理学科・物理学専攻

OBJECTIVE.

2022年度修士論文発表会最優秀・優秀発表賞が
下記の通り決まりました。

最優秀発表賞：高寺俊希さん
優秀発表賞：今井 皓さん，齊藤聖悟さん

表彰は修了式において行われる予定です。おめで
とうございます。

2023/3/29