

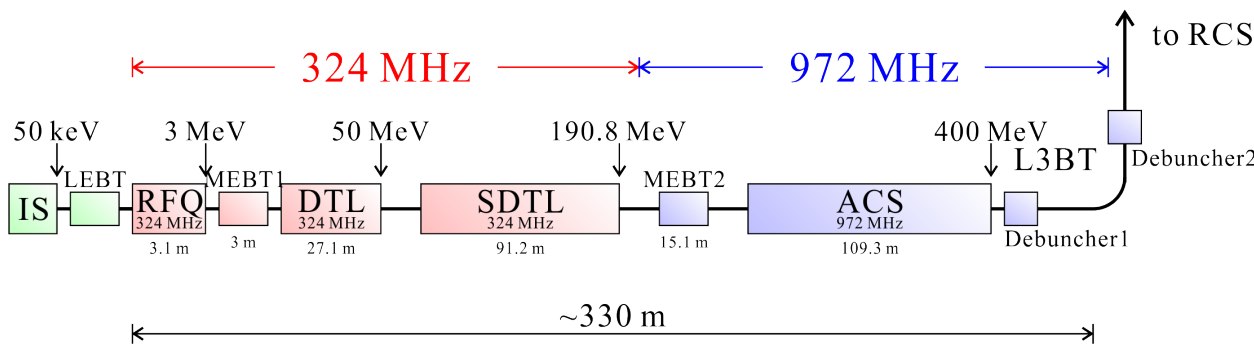
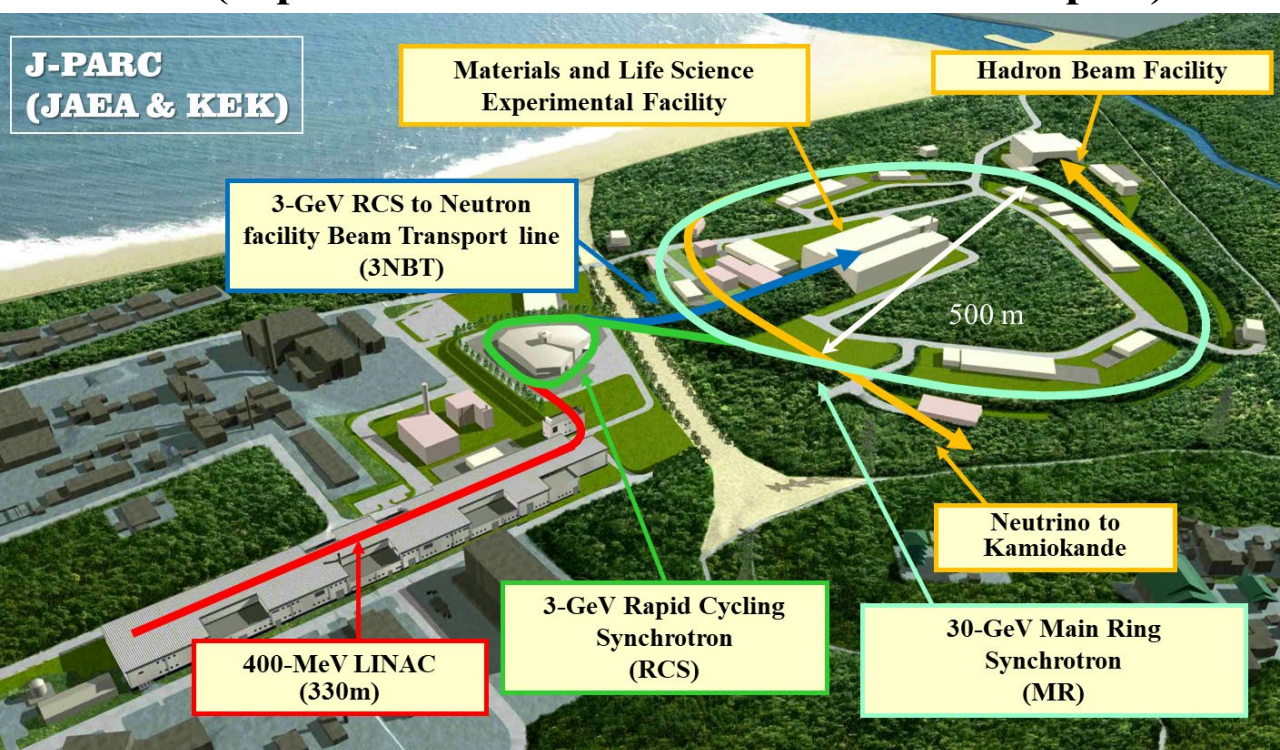


J-PARCリニアックLLRFシステムの高度化

高エネルギー加速器研究機構 ニツ川 健太

リニアック LLRFメンバー 高エネルギー加速器研究機構 Ersin Cicek, 方志高, 福井佑治, 溝端仁志,
NAT 佐藤福克, 日本原子力研究開発機構 篠崎信一

J-PARC (Japan Proton Accelerator Research Complex)

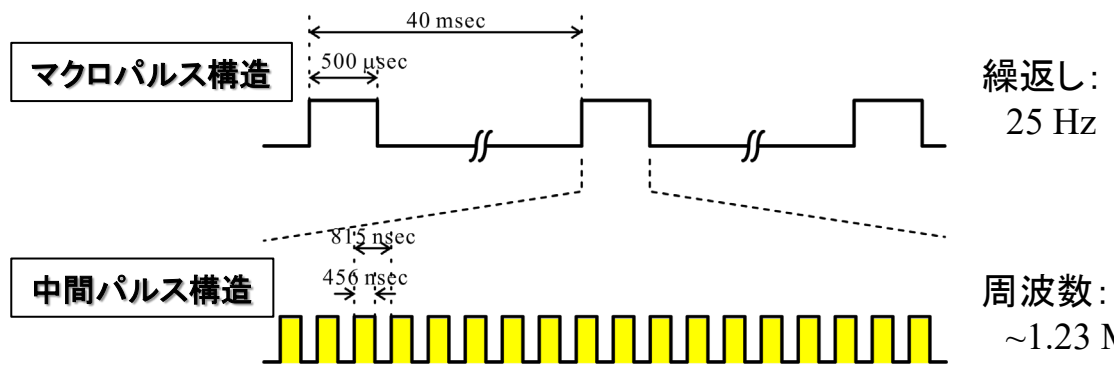


IS: H⁻ Ion Source
 RFQ: Radio Frequency Quadrupole Linac
 DTL: Drift Tube Linac (3 cavity modules)
 SDTL: Separate-type Drift Tube Linac (16 cavity stations)
 ACS: Annual Coupled Structure Linac (21 cavity stations)

J-PARC LINAC ビーム条件

- 加速粒子: H⁻ (negative hydrogen)
- 運動エネルギー: 400 MeV
- **ピークビーム電流:** 50 mA
- 加速周波数: 324 MHz / 972 MHz
- **パルス:** 500 μs (ビーム), ~650 μs (RF)
- 繰返し: 25 Hz (ADSを除く)

→ 大電流ビームを加速するため空洞内電界へのビーム負荷は大きく、ビーム負荷補償システムの性能が空洞内電界の安定度へ大きく寄与する。



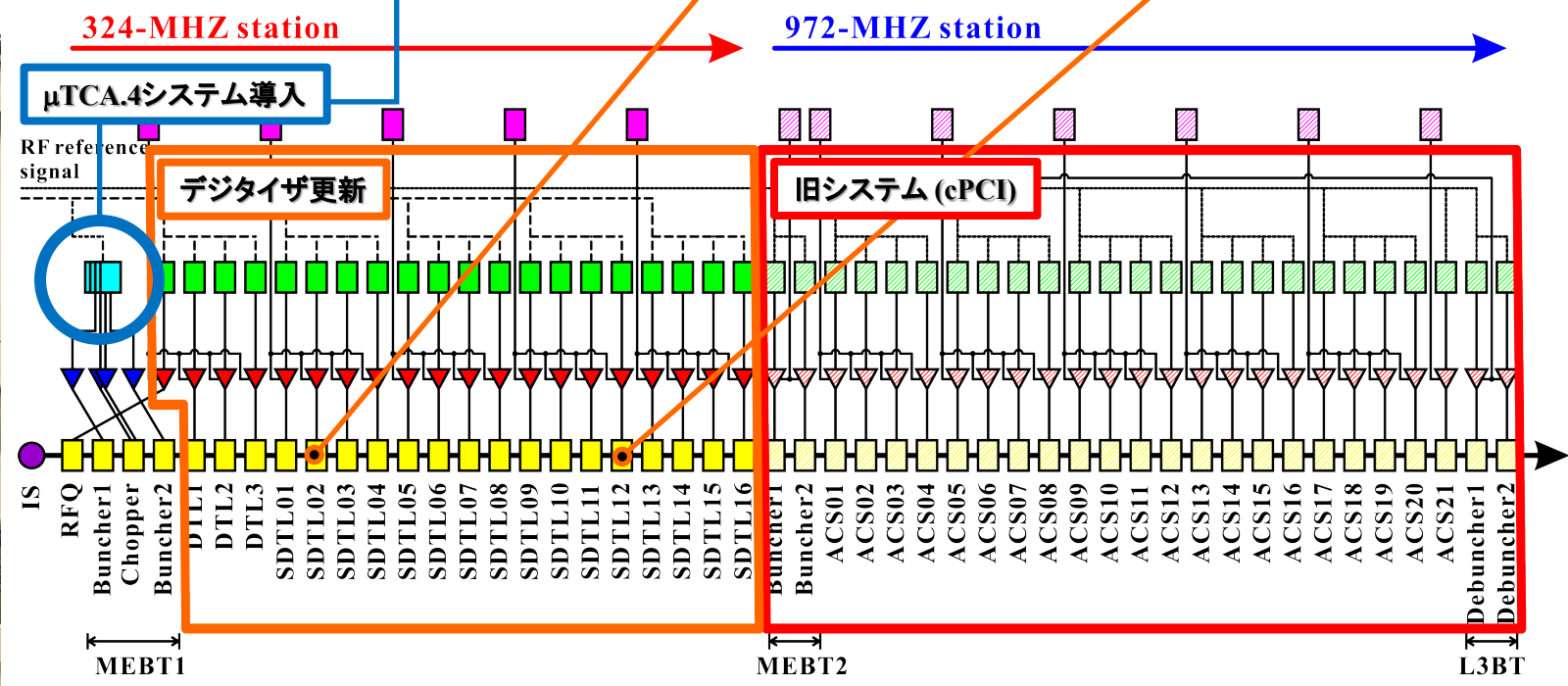
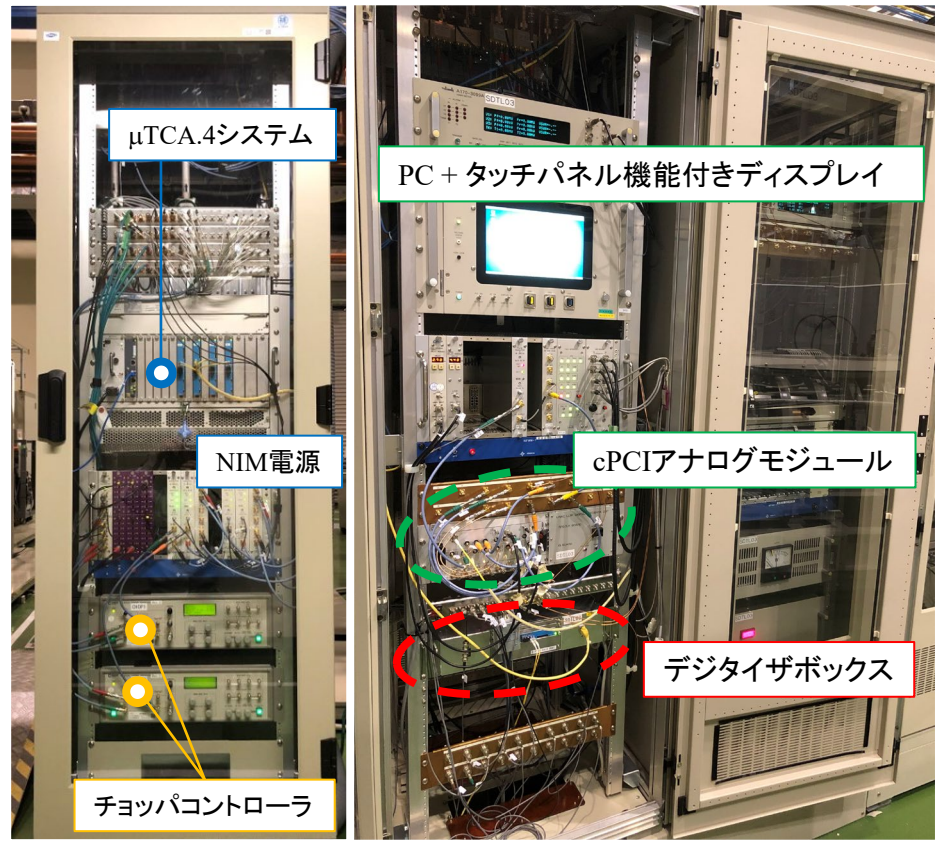
周波数: ~1.23 MHz → マクロパルス内で、ビームの有り無しが繰り返される。言い換えると、ビーム負荷の有り無しが繰り返され、空洞内電界が揺さぶられる。²

μTCA.4システム導入 **デジタイザのみ更新**
 for MEBT1 B1-2, C1-2 @ RFQ, DTL1-3, SDTL01-16

μTCA.4システム導入: 2021年夏期シャットダウン @ MEBT1

デジタイザのみ更新 2020年5月 @ SDTL02

デジタイザのみ更新 2020年1月 @ SDTL12



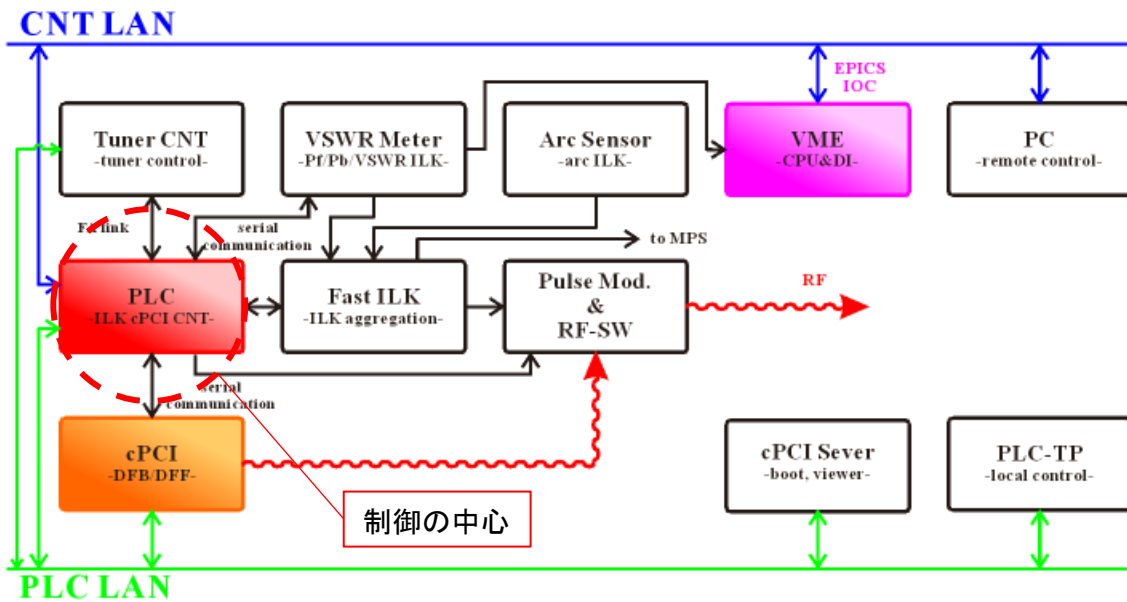
#components	HVDC PS	LLRF (DFB and DFF)	RF source	cavity station
324 MHz	5	4(μTCA.4) 20(digitizer box)	20(klystron) 4(SSA)	24
972 MHz	7	25(cPCI)	25(klystron)	25

2021年の夏期シャットダウンで、324-MHz LLRFシステムのデジタル系を更新を完了した。この新規導入したデジタル系を使用して、ビーム負荷補償システムを開発した。

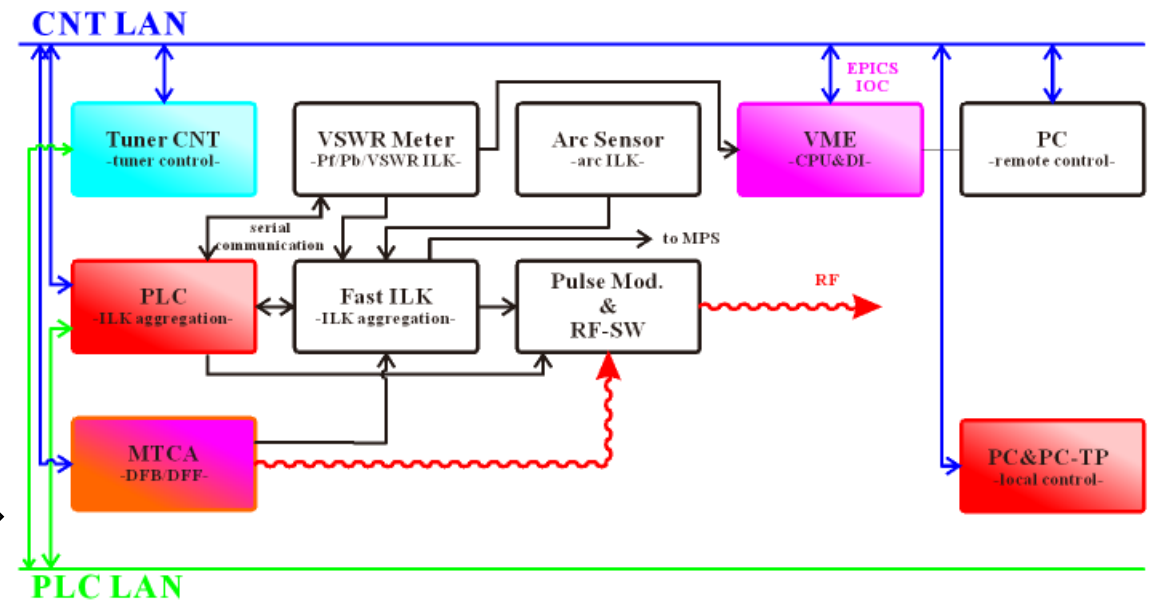
2021年夏期シャットダウン @ RFQ, DTL1-2

2020年夏期シャットダウン @ DTL2, SDTL01-16

cPCIを使用したシステムの模式図



デジタイザを使用したシステムの模式図



cPCIのDFB・DFFを使用する既存のシステムでは、制御の起点は、PLCであり、全ての系統の信号はPLCを経由していた。

cPCIシステムの老朽化のため、維持は困難
→ 持論: デジタルは20年も維持させるものではない!!!

デジタイザ/MTCAのDFB・DFFを使用するシステムでは、配列の設定・モニタなど、PLCを起点とすることは困難であった。

機能移行:

- DFB・DFF用のEPICS IOC
 - 自動調整, 各機器に制御がまたがるシーケンス
 - 電源のオフ/オンでの設定値の保持
 - タッチパネルの機能を他で代用することが必要
- ローカルの設定・モニタ: PC + タッチパネル付きディスプレイ
→ ILK発報履歴: 別途システム開発
→ トレンド監視: local archiver + css

対応力を確保するために、基本的には
自営での開発

J-PARCリニアックDFB/DFEのハードウェア
RF&CLKボード: 一部のICが廃版
FPGAボード: 廃版
DSPボード: 廃版
→ 予備品を大量に購入して、現状を維持

J-PARCリニアックLLRFシステムの開発ツール
ホストプログラム: RedHat8.0 gccコンパイラ Ver3.2
DSP: TI社 Code Composer Studio Ver2.1
FPGA: Xilinx社 ISE Ver6.2i
表示アプリ: python2.4, wxPython2.6

→ 開発環境を整備することが既に困難な状況

Pro-face製のタッチパネルは、PLCは制御できても、デジタイザの制御は不可
 → Pro-faceのタッチパネルの代わりにPC (TP-PC)を設置

無線のマウスとキーボードも用意

- remote/localはEPICSのASG
- ローカルの設定・モニタ: PC + タッチパネル(TP)機能付きディスプレイ
 メイン: cssベースでの開発

当初は、javaベースで開発してほぼ完成まで至ったが、ディスプレイの解像度が異なるとwidgetの配置が崩れて絶望した。
 → cssベースで再作成

チューナの手动制御には、javaでの開発の画面が残る。



トレンド: cssのdata browser活用

現在のJ-PARCのJAEA側のアーカイバは、ユーザからは自由に使い難い。自営で、トレンドを表示するプログラムを作成するか、ローカルアーカイバを設けてdata browserを活用するかで検討

- 後者を選択 (ローカルアーカイバ: channel archiver → archiver appliance)
- この組み合わせは閲覧が便利なので、監視頻度が高いレコードは、ローカルアーカイバでデータ収集

インターロック発報履歴: MySQLデータベース + python

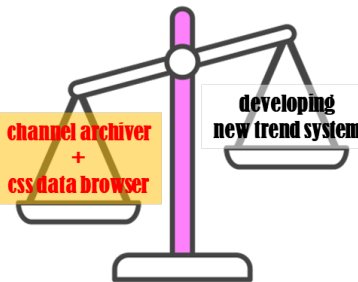
テキストで保存するか、データベースを使用するかで検討
 → 後者を選択

H2データベース + javaの組合せで開発したが、H2データベースの普及率?
 → MySQLデータベース + python

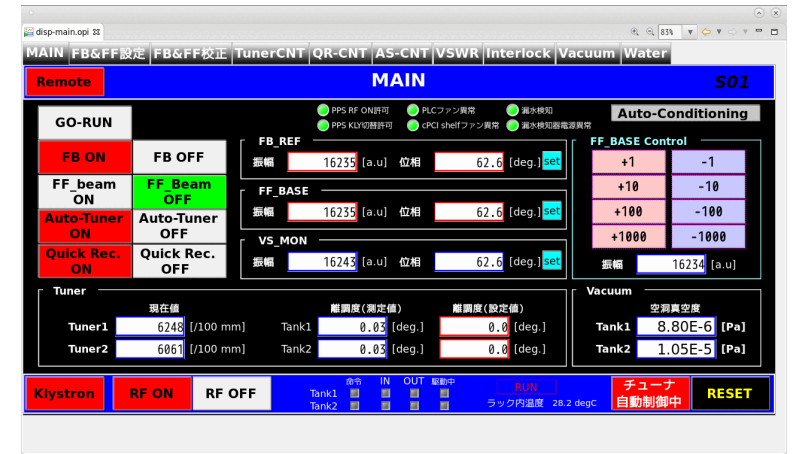
PLCのDレジスタをリングメモリのように用いている
 → 高速ポーリングの必要がない

自動コンディショニング機能: javaで開発

PLCラダーからjavaプログラムに移行されたことは、インターロックと完全分離と、間違いのない停止が可能となり、安全性の向上につながった...はず。



内容	発生時刻	復帰時刻
FAV1 アーク検出	2022/05/13 09:27:05.5	2022/05/13 09:28:28.5
クライストロン電源異常	2022/05/16 14:04:06	2022/05/16 15:25:52
クライストロン電源異常	2022/05/17 09:59:55	2022/05/17 10:04:38
クライストロン電源異常	2022/05/09 23:11:25	2022/05/09 23:15:00
クライストロン電源異常	2022/05/07 16:39:16	2022/05/07 16:42:22
クライストロン電源異常	2022/05/04 20:04:22	2022/05/04 20:07:41
クライストロン電源異常	2022/05/02 16:22:27	2022/05/02 16:25:26
CL冷却水流量異常	2022/05/02 18:18:26	2022/05/02 18:24:41
CL冷却水流量異常	2022/05/02 18:18:16	2022/05/02 18:24:41
TANK1 OK AND検出	2022/05/02 18:07:10	2022/05/02 18:21:03
TANK2 OK AND検出	2022/05/02 18:07:17	2022/05/02 18:21:03
クライストロン電源異常	2022/05/02 09:18:29	2022/05/02 16:21:03
V5 VSWアーム	2022/02/27 16:39:48	2022/02/27 16:40:06
V5 VSWアーム	2022/02/27 16:39:48	2022/02/27 16:40:06
V5 ペルベアアーム	2022/02/27 16:39:48	2022/02/27 16:40:06
V2 ペルベアアーム	2022/02/27 16:39:48	2022/02/27 16:40:06
SRAアンパ出力下層	2022/02/27 16:31:29	2022/02/27 16:39:15
SRAアンパ出力上層	2022/02/27 16:31:29	2022/02/27 16:39:15
SRAアンパ異常	2022/02/27 16:26:44	2022/02/27 16:39:15
コンパニオン異常	2022/02/27 16:12:47	2022/02/27 16:29:14



Pro-face製のタッチパネルは、PLCは制御できても、デジタイザの制御は不可
 → Pro-faceのタッチパネルの代わりにPC(TP-PC)を設置

- remote/localはEPICSのASG
- ローカルの設定・モニタ: PC + タッチパネル(TP)機能付きディスプレイ
- バックグラウンド ジョブ(screen コマンド)

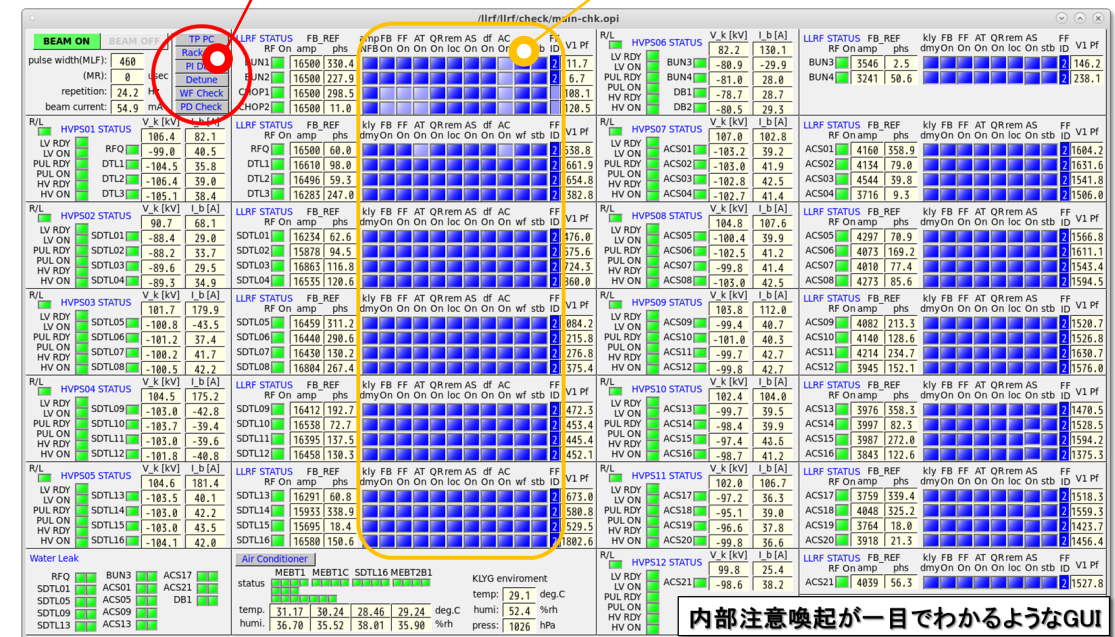
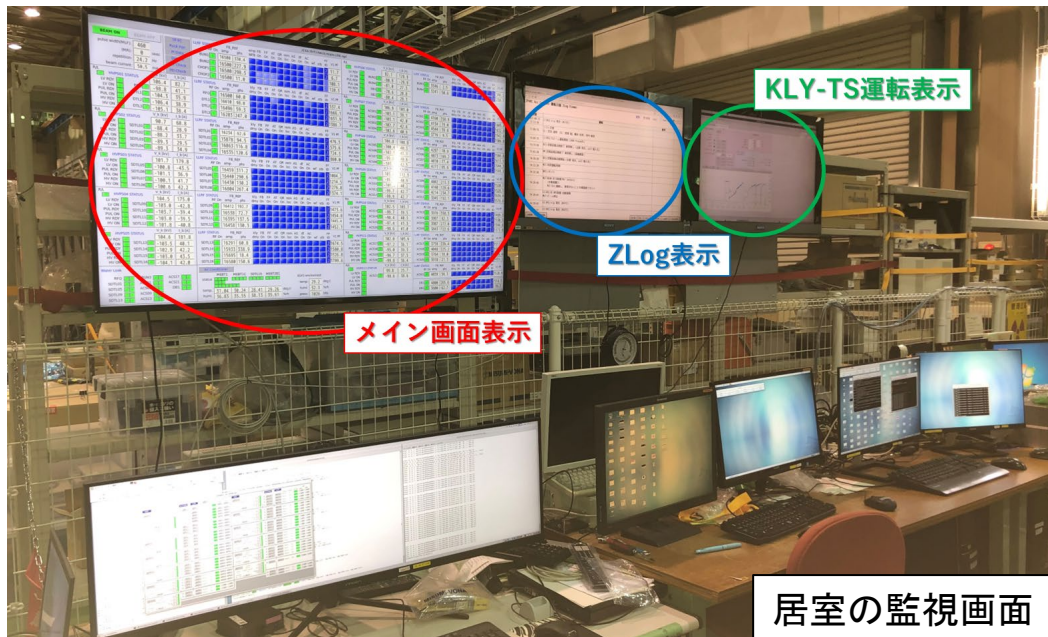
下記の機能は、LLRF PLCラダーが担っていたが、screenバックグラウンドジョブとして実行

- 自動RF立上げシーケンス
- 自動チューナ制御のシーケンス
- FF_BEAMのオン/オフ

更に、上記とインターロック発報履歴のプログラムには、ウォッチドッグス機能を組み込み、居室から動作監視(実績上、プログラムはデバック時以外に停止したことはないが、しばしばPCはフリーズする。)

- PC状態(watch dogs)
- ラックファンの状態
- FBの補正值
- 離調度
- 波形保存
- 位相ドリフト補正

- KLY/DMY
- FB/FF ON/OFF
- 自動チューナ制御 ON/OFF
- 自動復帰 ON/OFF
- remote/local
- Auto-Start ON
- 出力変調 ON/OFF
- Auto-Conditioning ON/OFF
- 自動波形監視 OK or Not
- Stable OK or Not
- 使用中のFF Table番号



居室の監視画面

内部注意喚起が一目でわかるようなGUI

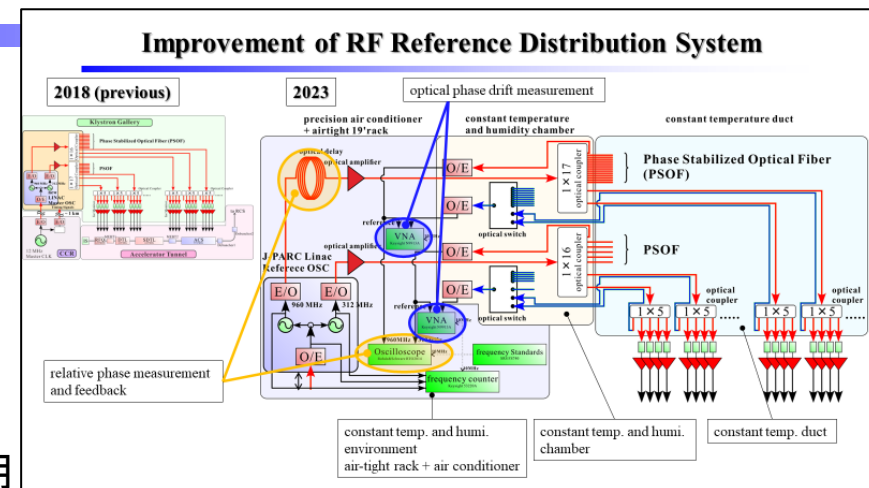
サーバPCのバックグラウンド ジョブ (screen)

- ✓ EPICSのsoft-IOC
- ✓ 高周波基準信号分配システムのデータ収集
- ✓ 位相ドリフト補正
- ✓ ビーム情報の取得 → インターロック統括監視システム用 (後述)

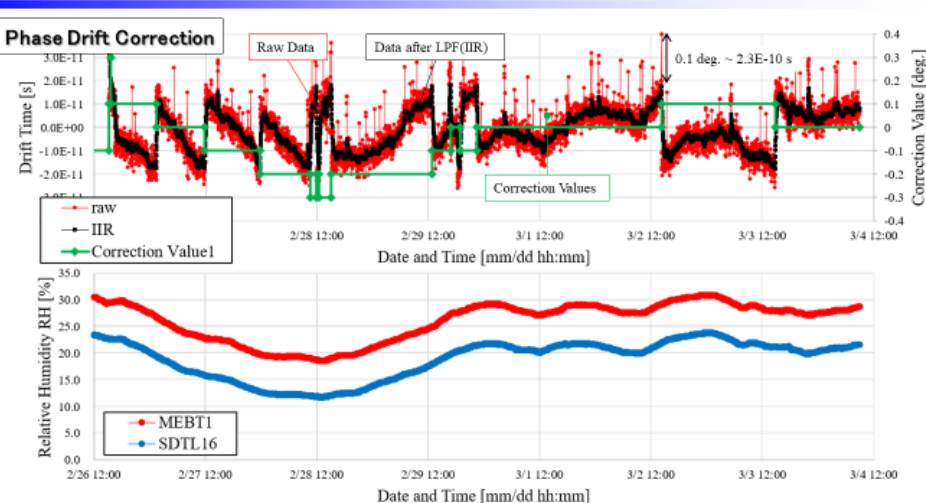
オシロスコープ/VVM
光スイッチ制御
光遅延器制御

居室PCのバックグラウンド ジョブ (screen)

- ✓ RFダウン時のオシロスコープ波形の取得 → インターロック統括監視システム用
- ✓ ウォッチドックス: PCフリーズ, および他のバックグラウンドジョブの動作状態の監視



位相ドリフト補正: 位相差 Drift Compensation

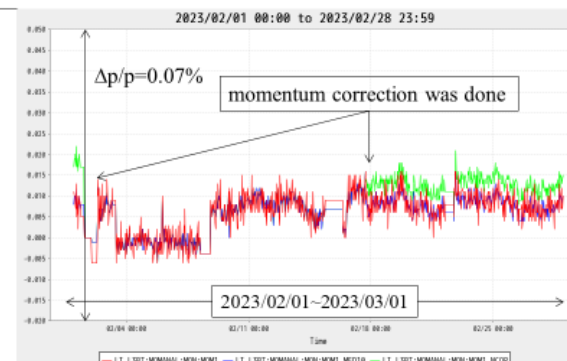
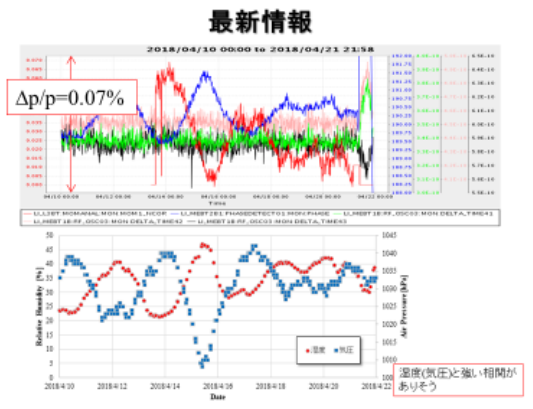


The LLRF systems in the klystron gallery environment are assumed to drift in phase by the same amount of the phase drift, and the phase drift compensations are done when they vary by more than 0.075 deg. from the reference.

位相ドリフト補正の効果: リニアック出射ビームの運動量

2018 (first time long-term measurement)

2023/02/01-03/01



2月の運動量補正: 2回だけ

As the results of the efforts of the RF group, the stability of the output momentum has improved enormously.

サーバPCの定時動作プログラム (cron)

- ✓ RF波形の確認: 1回/時間
 - 安定度が悪いと, 波形を保存して, 居室の監視ディスプレイに表示
- ✓ RF波形の保存: 3回/日
 - 波形を自動保存 (波形の安定度に関わらず)
- ✓ cPCIのインターロック情報をMySQLデータベースに保存: 2回/時間

居室PCの定時動作プログラム (cron)

- ✓ LLRF PLCシーケンスCPUの時刻合せ: 1回/日
- ✓ VSWRメータのパラメータの確認/記録: 1回/時間
 - 変更に気が付かなかったことがあり, 導入
- ✓ インターロック統括システムのアーカイブ: 1回/日

古い画面：
新しい画面：

medm, java (制御Gr., 自作)

css (自作) → 制御LAN上のPCにcssがインストールされているとは限らない

- ステーション毎に表示する画面：各ステーションで5枚以上
→ 全てのステーションで、一斉にパラメータ変更するときに、変更洩れなどが発生
- RFの機能毎に表示する画面：~30枚
- その他：~5枚 (ビーム条件, 位相ドリフト補正, ...)
- トレンド表示(ローカルアーカイバ)

50台近いステーションを確認すると、どこまで確認したのか分からなくなる。

ゲシュタルト崩壊

< FB Gain Parameter Check >

324 MHz 972 MHz

	P Gain				I Gain				UPDATE
	Peak	Start	Stop	Risetime	Peak	Start	Stop	Risetime	
MEBT1B1	30	5	1024	80	6000	5	1024	80	UPDATE
MEBT1B2	16	5	1024	80	10000	5	1024	80	UPDATE
MEBT1C1	0	0	0	0	0	0	0	0	UPDATE
MEBT1C2	0	0	0	0	0	0	0	0	UPDATE
RFQ	7	5	1024	50	9800	5	1024	50	UPDATE
DTL1	40	0	1024	100	10000	0	1024	100	UPDATE
DTL2	50	0	1024	100	12000	0	1024	100	UPDATE
DTL3	50	0	1024	100	12000	0	1024	100	UPDATE
SDTL01	30	5	1024	80	7000	5	1024	80	UPDATE
SDTL02	30	5	1024	80	7000	5	1024	80	UPDATE
SDTL03	30	5	1024	80	10000	5	1024	80	UPDATE
SDTL04	60	5	1024	80	16000	5	1024	80	UPDATE
SDTL05	40	5	1024	80	13000	5	1024	80	UPDATE
SDTL06	30	5	1024	80	8000	5	1024	80	UPDATE
SDTL07	50	5	1024	80	11000	5	1024	80	UPDATE
SDTL08	30	5	1024	80	13500	5	1024	80	UPDATE
SDTL09	40	5	1024	80	15000	5	1024	80	UPDATE
SDTL10	45	5	1024	80	13000	5	1024	80	UPDATE
SDTL11	50	5	1024	80	13000	5	1024	80	UPDATE
SDTL12	45	5	1024	80	18000	5	1024	80	UPDATE
SDTL13	70	5	1024	80	21000	5	1024	80	UPDATE
SDTL14	40	5	1024	80	14000	5	1024	80	UPDATE
SDTL15	50	5	1024	80	13000	5	1024	80	UPDATE
SDTL16	45	5	1024	80	16000	5	1024	80	UPDATE

ステーション毎に表示する画面

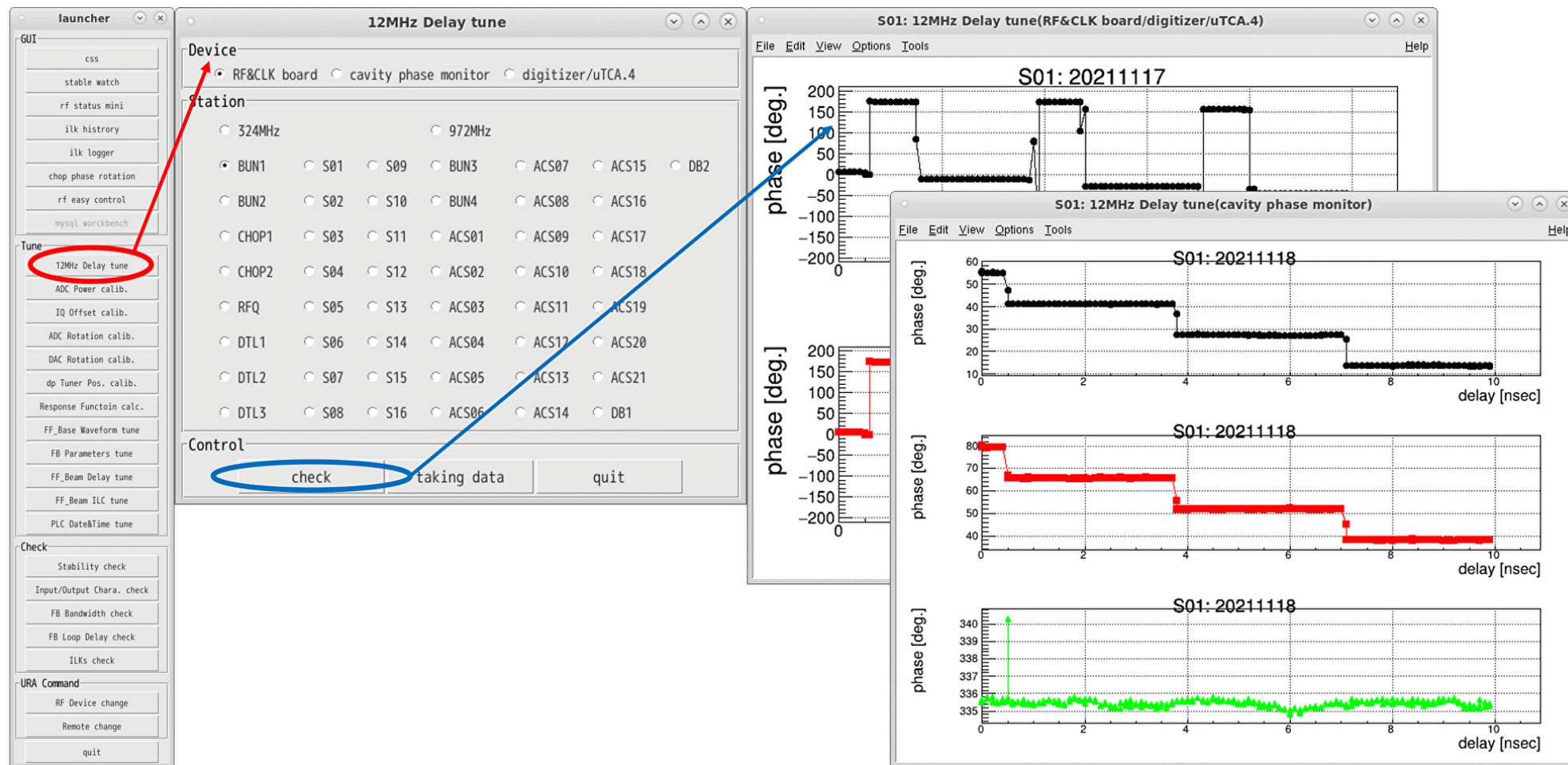
RF機能毎に表示する画面の例：FBパラメータ

pythonでパラメータの自動調整を実施：

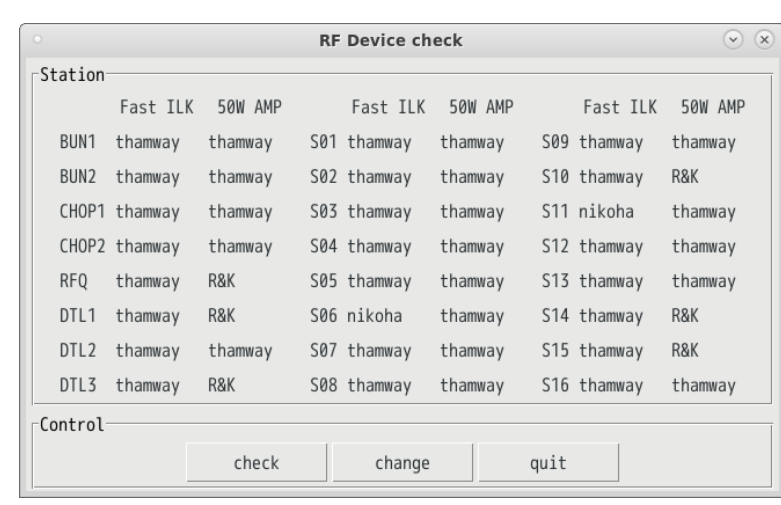
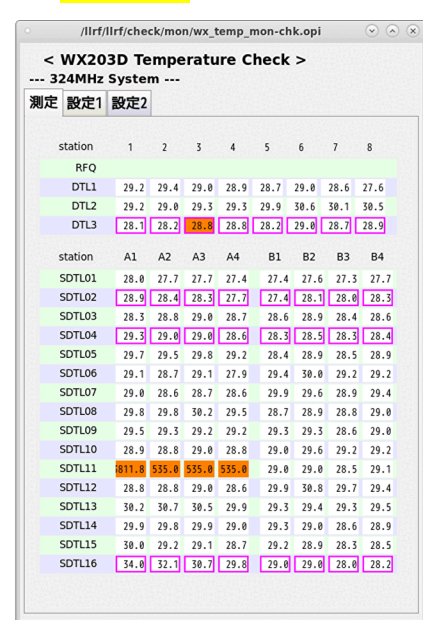
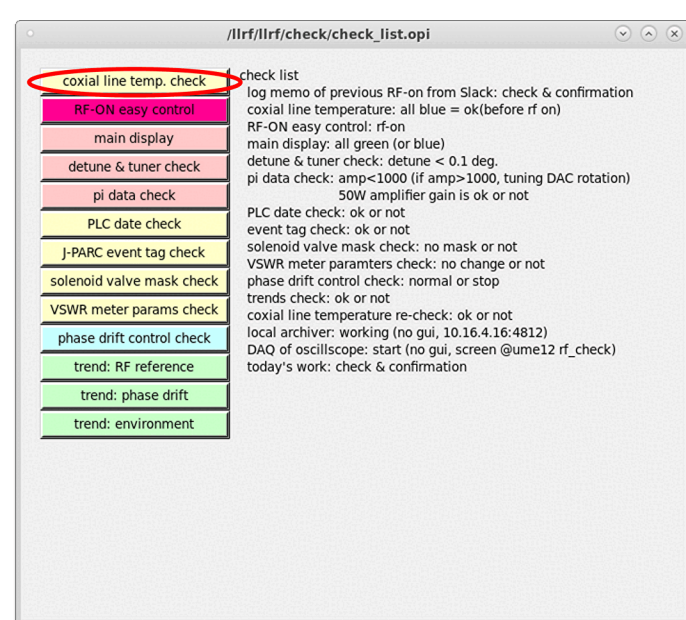
調整用のデータ測定時の表示はなし：同時に複数ステーションで測定を実施するため

→ 必要に応じて, cssの画面で確認

→ 測定結果を確認して, 問題がなければ設定に反映



- メンテナンスからの立上げ: チェックリスト & 操作画面を作成
 - 立上げ担当&責任者を決め, RF立上げを実施
 - メンテナンスでオフにした機能の復帰
 - 必要に応じてDFB&DFFシステムの微調整
 - 機器の動作状況の確認
 - 1週間のトレンドの確認
 - ...
 - グループ内の全員がシステムに触れるように...半強制的...
- RF機器の交換: 予備品は完全互換ではない(より安全に: N.O. → N.C.)
 - プログラムから誰でもPLCラダーを用意に変更可能な状況



PLCの特定のレジスタを変更することで変更可能
但し, 敢えてEPICSレコード化はしていない。

- インターロック発報履歴システムの拡張
 デジタイザ更新時にTP-PCに置いたインターロック発報履歴システムが非常に有用
 → デジタイザを更新していないステーションにも応用
 アーカイバへの自動保存は30分に1回、任意のタイミングでの手動保存も可

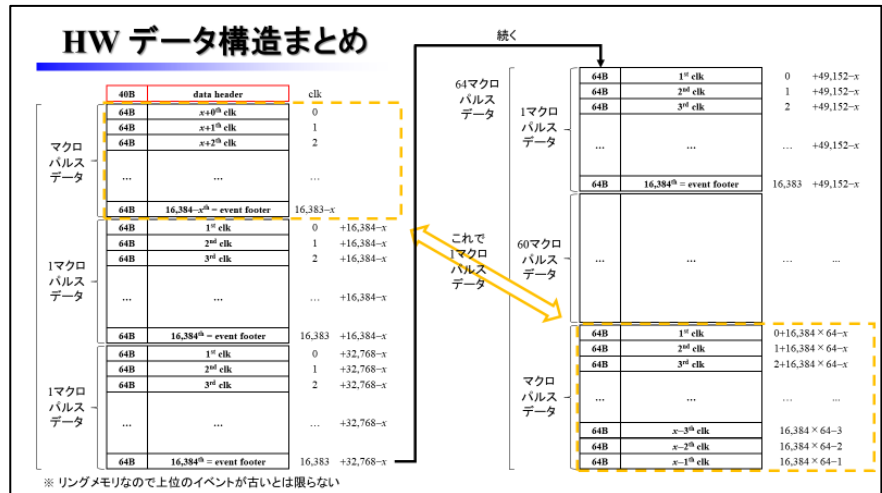
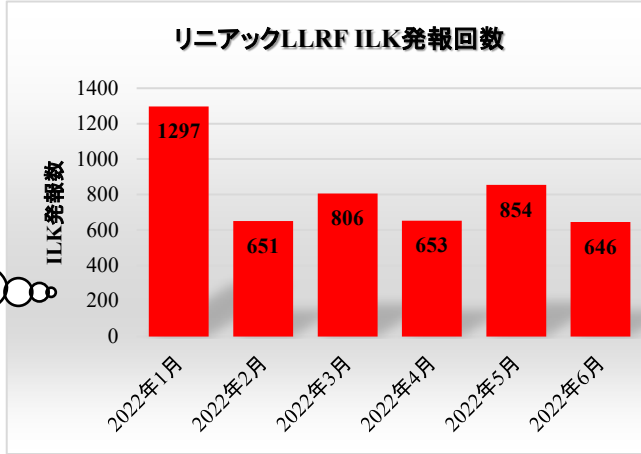
- RFダウン波形の保存 → 空洞から反射データ有り
 RFダウン波形のオシロスコープの画面を保存

- RFダウン情報の保存 → 前のパルス情報有り
 更新されたデジタル系の高速モニタ(波形とJ-PARCのイベントタグの取得)

→ 高速モニタの実装:

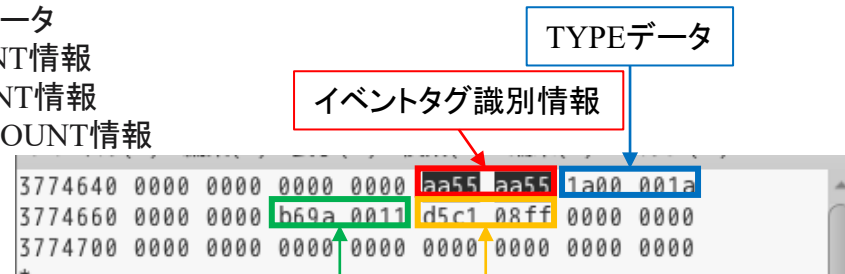
- 64pulses/bank, 16banks, 16383smps/pulse, 64bytes/smpのリングメモリで構成された高速モニタを実装
- ADC1~8,DAC,VS,IIR,REF,PI,FFの各IQ→FB中の計算過程も含めて、メモリに書き込む
- パルスデータのeventfooterには、J-PARCイベントタグ情報を付加

一ヶ月に1000件弱の
インターロック事象
がある。



event footer

- 0-3: イベントタグ識別情報(0xaa55aa55)
- 4-7: TYPEデータ
- 8-11: S_EVENT情報
- 12-15: S_COUNT情報
- 16-19: TRIG_COUNT情報
- 20-63: 0固定



※ リングメモリなので上位のイベントが古いとは限らない

インターロック事象の**半自動解析**でデータベースに保存

- デジタイザ/MTCAステーション:
ILK発報事象 → MySQLサーバに保存
- cPCIステーション:
ILK発報事象 → MySQLサーバに保存

自動: 数秒毎
“DIGI”データベース

自動: 1回/日
“CPCI”データベース

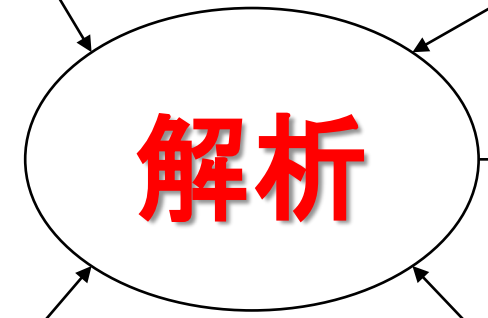


beam ON/OFF

ビーム運転時 or それ以外
ビーム停止時刻, 復旧時刻

ILK内容
ILK発報時刻
復旧時刻

解析データを
“MPS”データベースに保存



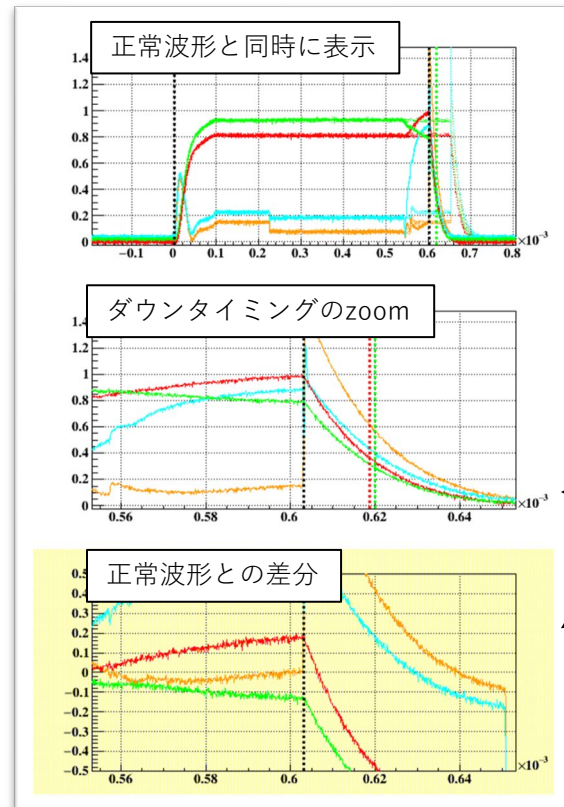
RFダウン波形のpngファイル名
前のパルスが正常かの判断
→目視の確認が必要かどうか
J-PARCイベントタグ (ビーム行先)

RFダウン情報の取得

発報タイミング, QR成功 or 失敗
時定数 → A or B
推定ビーム幅(MLF or MR)
推定発報原因
RFダウン波形のpngファイル名
目視の確認が必要かどうか

自動: RFダウン毎
RFダウン波形の取得

目視確認が必要そうな事象は黄色表示



pngファイルは/jkdata/jkpublic上に置かれてJLAN上から確認できると共に, 制御LAN上にrootのcanvasも保存して後々に詳細確認も可能

自動化できない箇所: “道連れ”MPS事象(Zlogから引き抜き), プログラムが目視を推奨したイベント
 → mysql-workbenchで用意にデータの編集が可能

“道連れ”MPS事象
 自動入力を確立すると管理が容易になる。

The screenshot shows the MySQL Workbench interface with a query result table. The table has columns including ID, ORG, REG_ID, STATION, ILK_CONTENTS, ILK_OCCUR_DATE, ILK_RESET_DATE, QUICK_RECOVERY, BEAM, BEAM_WIDTH, OTHER_ILKS, MPS_OCCUR_DATE, BEAM_RESTART_DATE, DOWN_TIMING_A_OR_B, CAUSE, DECAY_TIME_A, DECAY_TIME_B, CHECK_FLAG, CHECKED_FLAG, FILE_OSC_NAME, FILE_HW_NAME, TRIG_COUNT, S_COUNT, TYPE_CODE, MR_CYCLE_ID, DESTINATION, MAINTENANCE, and REMARKS. The table contains numerous rows of data representing different events and beam operations.

将来は、大きな故障前の予兆検出が可能か？ (1例)

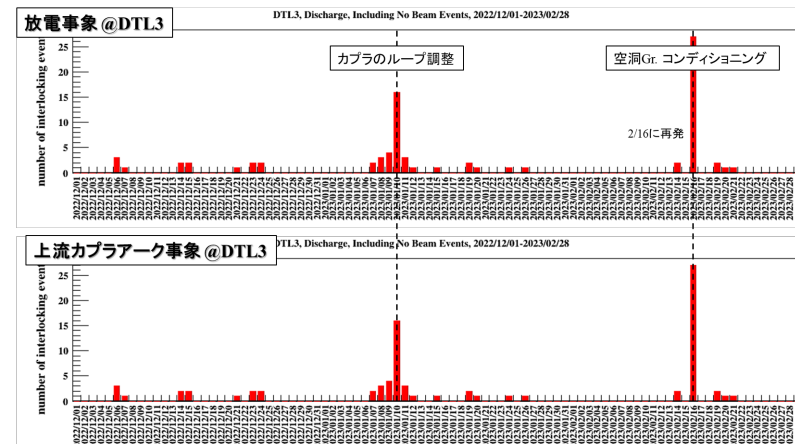
2022年2月に、RF Gr.から空洞Gr.にDTL3上流カプラ近辺(5事象)とSDTL11Bカプラ近辺で不可思議なインターロック発報現象を報告

2022年夏期シャットダウン時
 SDTL11Bカプラの交換を実施 (空洞Gr.)
 DTL3の立体回路の健全性の確認 (RF Gr.)

2023年2~3月にDTL3で放電が頻発
 → 急遽、DTL3のカプラの交換を実施

真空悪化が伴っていなかったため
 立体回路の可能性もあったため

2022年の夏期シャットダウン中に
 交換できていれば、避けられた
 可能性が大きい。

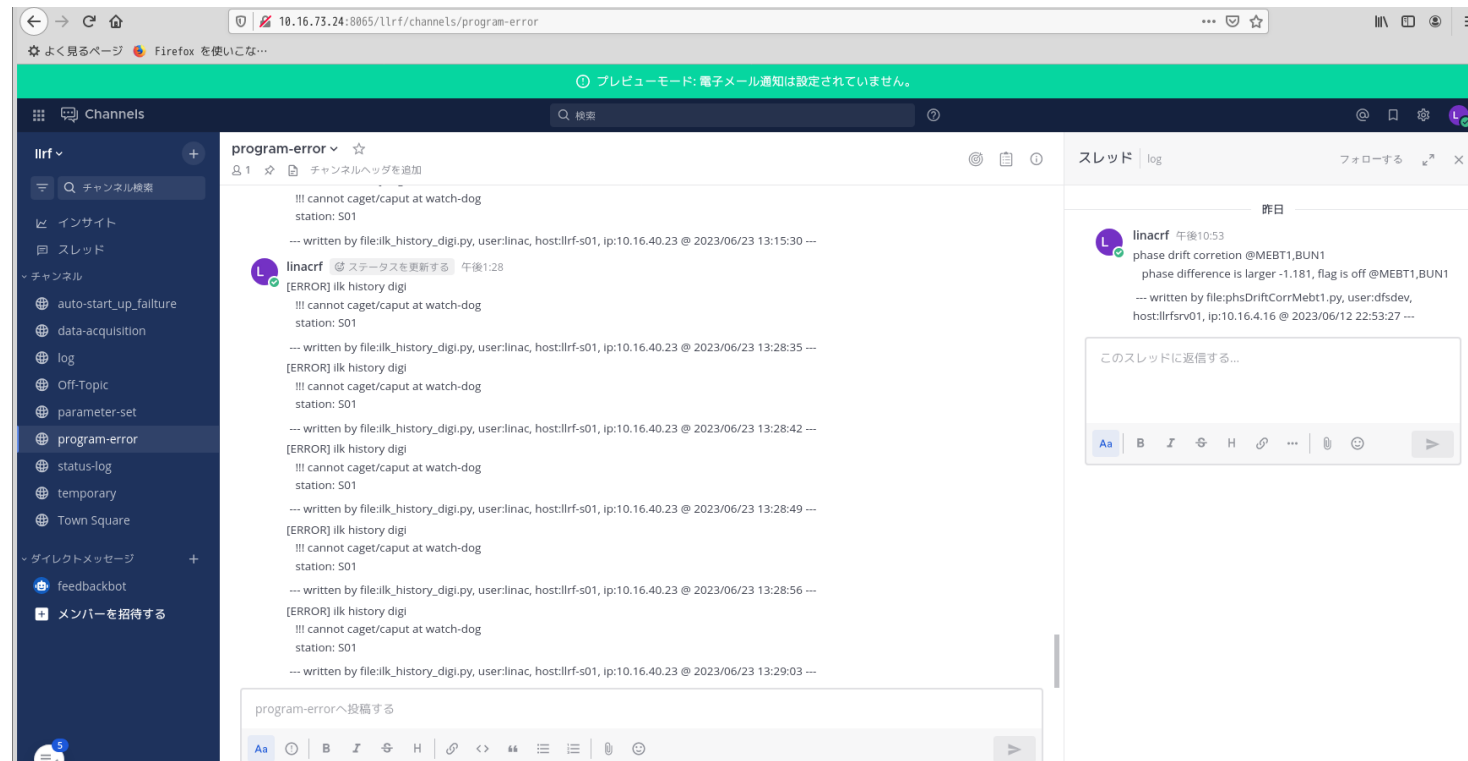


“Slack”のような“Mattermost”のローカルサーバを制御LAN上に設置

→ pythonプログラム上から、記録が可能 → **プログラムからの自動書記を組み込み**

下記のチャンネルを用意して、重要項目は自動で記録ができるシステムを実現

- data-acquisition: データ収集のログ ← 基本見る必要なし, 気になったら見る
- parameter-set: パラメータ設定のログ ← 基本見る必要なし, 気になったら見る
- program-error: プログラムのエラー ← 基本的に見る, デバックに役立つ
- status-log: 状態の記録 ← 絶対見る, 基本的には悪いときに書く
- temporary: テンポラリ ← 見る, テンポラリの状態に設定したときに書く, ex. HWの停止とか



- J-PARCリニアックのLLRFでは, 人手不足&予算不足をシステムの高度化で補うように努力をしている。今回は, 現状を報告したが, 機能の追加のための開発は常日頃から行っている。
- 今回の発表にはありませんでしたが, 同時に不必要になった機器の取外しや機能の削除も実施して, システムを単純化させる努力も行っている。
- 自営でのシステムの導入や改良を進めていることもあり, 開発スピードは決して早くはない。しかし, その分, 対応力&適応能力は高く, 後発開発が活きるLLRFでは重要だと考えている。