

トリガ付きスケーラのソフトウェア環境整備: EPICS db 開発

DEVELOPMENT OF EPICS-BASED SOFTWARE OF TRIGGERED SCALER

田島佑斗^{#, A)}, 上窪田紀彦^{B)}, 佐藤健一^{B)}, 吉田奨^{A)}, 山本昇^{B)}

Yuuto Tajima^{#, A)}, Norihiko Kamikubota^{B)}, Kenichi Sato^{B)}, Susumu Yoshida^{A)}, Noboru Yamamoto^{B)}

^{A)} Kanto Information Service

^{B)} High Energy Accelerator Research Organization (KEK) /J-PARC Center

Abstract

A "triggered scaler" has been developed for J-PARC accelerators. It is a scaler to count pulsed signals, which are provided for accelerator components. It stores counts in a memory array, and each external trigger (typically 25Hz, from J-PARC Timing System) shifts the pointer to the array during a machine cycle (2.48s or 5.20s). Then, stored data (waveform-type EPICS record) is expected to verify no miss-trigger (or, to find a miss-trigger) in the previous machine cycle. In order to realize such functionalities, we have made basic design of database definitions (EPICS db) for triggered scaler. Three-layered design has been discussed. Development status of EPICS db, early experiences with real MR signals, are reported.

1. はじめに

J-PARC(Japan Proton Accelerator Research Complex) は、日本原子力研究開発機構(JAEA)と高エネルギー加速器研究機構(KEK)が共同で計画及び運営を行っている大強度陽子加速器施設である[1,2]。

J-PARC では、イベントベースのタイミングシステムを使用している。J-PARC タイミングシステムでは、Linac(LI)/RCS/MR の 3 加速器で、それぞれ約 540/220/300 点のトリガ信号を管理している[3]。

大量の信号が機器現場で生成されるが、何らかの伝送系のトラブル(ケーブル断線やモジュール fuse 切れなど)で信号が途切れ、かつ中央制御棟からの遠隔監視でそれがわからない、という事例があった。このようなトラブルを検知するため、生成したトリガ信号を末端の機器側で検出する新しいモジュール(トリガ付きスケーラ)を開発した[4]。

いるので省略する。ここでは MR の入射信号(通称 K1-K4)を例に、期待される動作を説明する。

K1-K4 信号は Fig. 1 の Normal に示す通り、4 連続トリガ信号である。K1-K4 信号をトリガ付きスケーラで計測することで、3 パターンの異常を判別できる。1) トリガ抜け: K1-K4 信号の K3 が抜けた例、2) 不規則なトリガ: 来てはいけない時にトリガが来た例、3) ダブルトリガ: 本来 1 つのはずのトリガを 2 回計測した例、である。

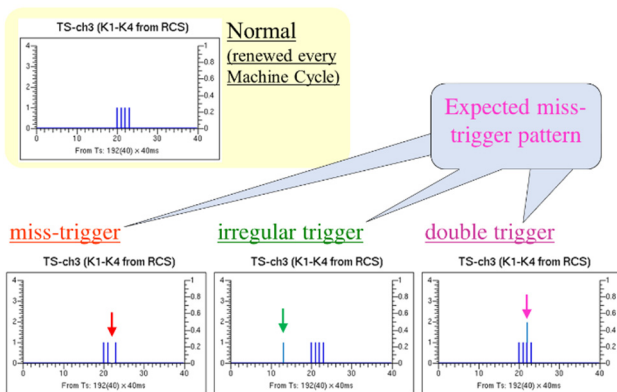


Figure 1: miss-triggers.

トリガ付きスケーラの内部仕様は[4]で詳細に解説して

[#]z-tajima@post.j-parc.jp

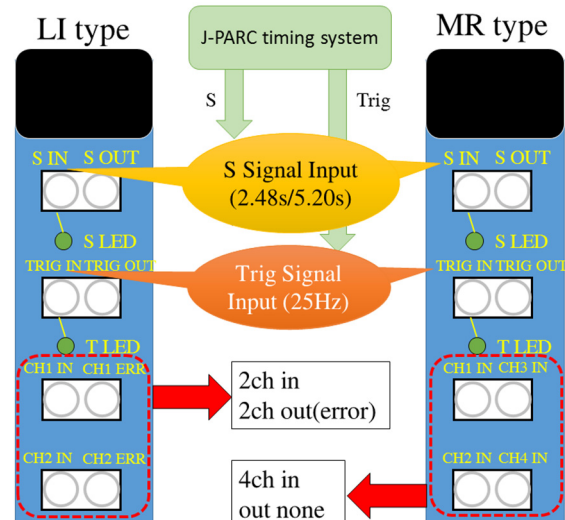


Figure 2: Two triggered scalers: LI-type and MR-type.

Figure 2 に示すように、トリガ付きスケーラには LI タイプと MR タイプの 2 種類がある。前者は Rapid-cycle(25Hz)で動作する LI と RCS 向け、後者は Slow-cycle(2.48s or 5.20s)の MR 向け、である。両タイプとも、J-PARC タイミングシステムが S signal (加速器サイクルの始まり信号)と Trigger (rapid-cycle 25Hz)を供給する。

LIタイプは、計測信号(2chある)で抜けを検知すると、直ちに out(error)を出力する。一方 MRタイプでは out 出

力は無く、入力を倍の 4ch にした設計である。

2. ソフトウェア設計

2.1 概要

J-PARC 加速器は、EPICS toolkit を用いた加速器制御システムで運用されている[5,6]。トリガ付きスケーラを EPICS 環境で利用するために、ソフトウェア環境整備として EPICS db (db は database の略、EPICS の信号定義) の設計・開発を行った。

開発した epics db は、3 階層に分かれている。第 1 層では、トリガ付きスケーラモジュールの hardware を定義する。例えば、使用中の double buffer のメモリ番号(0 or 1) を持つ register や、計測したデータを格納する配列が定義される。第 2 層では、MR 向けの運用パラメータを定義する。第 3 層では、計測する加速器信号に応じた部分を定義する。別の報告[7]で、入射信号(K1-K4)計測、RF 信号計測、MPS 検知の実例が示されているが、それぞれの信号は別の定義 db に対応する (Fig. 8 も参照)。

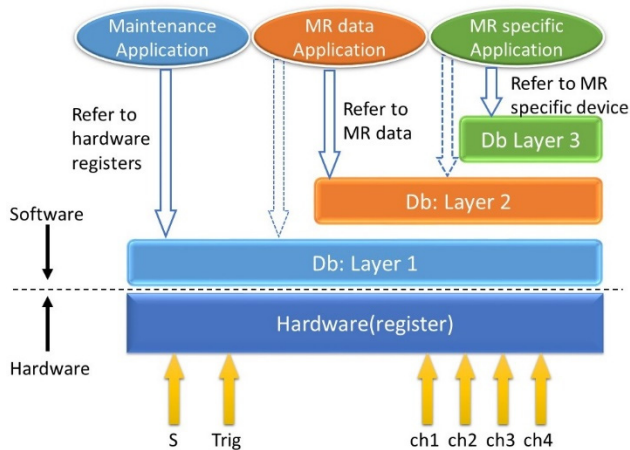


Figure 3: Relationships between db layers and applications.

Figure 3 に、3 階層の db とアプリケーションの参照関係を示した。第 1 層 db が hardware を記述するので、第 2 層・第 3 層は直接 hardware(register) にアクセスしない。第 2 層・第 3 層の db 定義は、第 1 層を参照する。また、第 2 層・第 3 層に対応するアプリケーション (Fig. 3 では「MR data Application」と「MR specific Application」) は、主に対応する層の db を参照する。

2.2 第 1 層:モジュールの hardware を定義

第 1 層は、トリガ付きスケーラを持つ hardware register を、機械的に Record (EPICS の信号単位) に対応づける。Table 1 に、その一部を示す。計測データが入るメモリは MR タイプでは 192(LI タイプでは 448)である。

Table 1: Part of Record Definitions

Record	Register address	Description
pageNow	9	Page number, 0 or 1, which points to the active memory area
triggerNow	10	Position of a memory, 0 to 191, where a counter value is written
pageSet	11	Switch a page number, 0 or 1, which memory to be accessed from CPU
errStatus	14	Error status in counting process
triggerInCycle	16	Number of triggers in the previous S-signal period
wf:ch1	33~224	Array data (192) for 1ch [MR-type]
wf:ch2	289~480	Array data (192) for 2ch [MR-type]
wf:ch3	545~736	Array data (192) for 3ch [MR-type]
wf:ch4	801~992	Array data (192) for 4ch [MR-type]

Figure 4 は、トリガ付きスケーラの第 1 層の EPICS db 定義の一部である。Table1 と対応していることが分かる。

ts_ref_li.db

```
record(longin, "$(host):TS$(n):$(name)")
{
  field(DTYP, "F3RP61")
  field(INP, "@U$(U),S$(S),A$(C)")
  field(SCAN, ".2 second")
}
```

ts_common.substitutions

```
file db/ts_ref_li.db {
  pattern { host, n, name, U, S, C }
  [{"test08", "0", "pageNow", "0", "3", "009"}]
  [{"test08", "0", "triggerNow", "0", "3", "010"}]
  [{"test08", "0", "pageSet", "0", "3", "011"}]
  [{"test08", "0", "trigSel", "0", "3", "012"}]
  [{"test08", "0", "errSet", "0", "3", "013"}]
  [{"test08", "0", "errClear", "0", "3", "015"}]
  [{"test08", "0", "errStatus", "0", "3", "014"}]
  [{"test08", "0", "triggerInCycle", "0", "3", "016"}]
}
```

Figure 4: Part of db definitions for Layer1.

2.3 第 2 層:MR 向けの運用パラメータを定義

LI は 25Hz 運転なので、どの機器も 40ms に 1 回トリガを受ける。したがって、40ms の中で抜けがあれば問題になる。一方、MR は遅い周期 (2.48s or 5.20s) なので、その周期の 1 サイクルに 1 回のトリガ抜けが問題になる。ここでは、MR 向けの運用で必要な Record を検討する。



Figure 5: Behavior of double buffer for MR signals.

Figure 5 は、MR の信号を計測・記録している double buffer を示したものである。pageNow は 0 を指し(上側の memory 0 を使用中)、memory 0 内の書き込み位置は triggerNow (Fig. 5 では 0 から数えて 5) で示される。

MR 運用上は、memory 0・memory 1 という区別より、現在書き込み中のメモリ、または前回のサイクルの書き込みが終了したメモリ、という読み出し方が便利である。しかし、CPU からは pageSet で指定した片方のメモリしか読み出せないため、第 1 層で定義した Record だけではこの要望は満たせない。このため第 2 層で、double buffer から適時メモリ内容を copy する配列 Record を別途定義する (Fig. 3 の wf:ch1_n が現在書き込み中、wf:ch1_x が前回分)。

2.4 第 3 層: 計測する信号に応じた実装の定義

MR で、3 種類の加速器信号 (RF, K1-K4, MPS) を計測した ([7] 参照)。ここでは、例として MPS 向け実装定義について説明する。

MPS は加速器機器保護のためのシステムで、MPS が起こると直ちに加速器全体を安全に停止し、他の機器が損傷する可能性を減らす [8]。MPS 発生時の実信号をトリガ付きスケーラで計測し、取得した前サイクルのメモリ (wf:ch1_x) に 0 で無い値が現れたか判断して MPS の発報を知ることができる。

Figure 6 は、MPS 向け EPICS db 定義の一部である。上部は wf:ch1_x のどこを切り出すか (加速器周期のどの部分を調べるか)、下部は切り出した wf:ch1_x に 0 で無い値があるか調べる (あれば MPS が発報と判断できる)。よって、この db 定義は MPS 発報の処理の開始シグナルとして用いることが出来る。また、Fig. 7 は、MPS 発報した時の関連情報 (Shot-number とその時刻、その shot の DCCT 波形、加速器周期のどのタイミングで発報したか、など) を統合して確認する画面である。

```
File Edit Setup Control Window KanjiCode Help
### MPS in MR cycle
record(subArray, "$(host):TS$(n):$(name):$(ch)") {
  field(INP, "$(host):TS$(n):MPS_WF:$(ch) CP NMS")
  field(NELM, "$(N)")
  field(INDX, "$(I)")
  field(MALM, "192")
  field(FLNK, "$(host):TS$(n):VAL:$(name):$(ch)")
}

record(compress, "$(host):TS$(n):VAL:$(name):$(ch)") {
  field(INP, "$(host):TS$(n):$(name):$(ch) CP NMS")
  field(ALG, "N to 1 High Value")
  field(NSAM, "1")
  field(N, "$(N)")
}
```

Figure 6: Part of db definitions for MR-MPS.

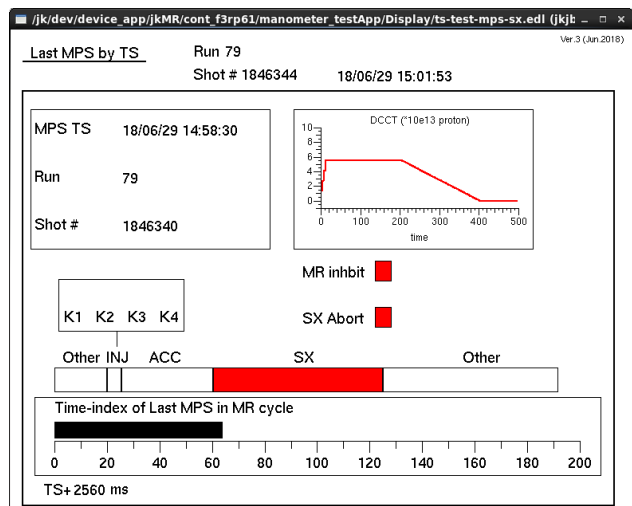


Figure 7: Application screen for MR-MPS.

2.5 開発・整備状況

現時点では特に第 2 層 (MR 向けの運用パラメータ定義) の整備がまだ不十分で、また第 2 層と第 3 層が混合した状態である。今後、各階層を明確に分離した db file に整理していく予定である (Fig. 8 参照)。

トリガ付きスケーラの試験は MR タイプで先行させており、LI タイプは未着手である。上記 db の整備が完了次第、LI タイプでの運用パラメータの定義などを進めたい。

Db: Layer 3 (MR)	A)	ts_L3_rf.db
	B)	ts_L3_kn.db
	C)	ts_L3_mps.db ts_L3_mps_timing.db
Db: Layer 2	MR	ts_L2_MR_wf_setup.db & substitute
	LI	ts_L2_LI_wf_setup.db & substitute
Db: Layer 1		ts_common.db & substitute ts_common_err.db

Figure 8: Plan of db filenames for three layers.

3. まとめ

トリガ付きスケーラを EPICS 環境で使用するために、EPICS db(信号定義)を3階層に分けて設計した。MRタイプのモジュールで現場試験を行いつつ、第2層と第3層のdb整備が進行中である。今後、db整備を継続し、またトリガ抜け検知など本番環境への実装を行っていく。

参考文献

- [1] J-PARC ホームページ; <http://j-parc.jp/>
- [2] K. Hasegawa *et al.*, “Performance and Status of the J-PARC Accelerators”, Proceedings of IPAC2017 in Copenhagen, Denmark, May 14-19, 2017, pp. 2290-2293.
- [3] N. Kamikubota *et al.*, “Operation Status of J-PARC Timing System and Futur Plan”, Proceedings of ICALEPCS in Melbourne, Australia, Oct. 17-23, 2015, pp. 988-991.
- [4] K.C. Sato *et al.*, “トリガ付きスケーラの開発とトリガ抜け検知”, Proc. of the 14th Annual Meeting of Particle Accelerator Society of Japan, Sapporo, Japan, Aug. 1-3, 2017, pp. 627-630.
- [5] EPICS ホームページ; <https://epics.anl.gov/>
- [6] N. Kamikubota *et al.*, “J-PARC Control toward Future Reliable Operation”, Proc. of ICALEPCS 2011, Grenoble, France, October 2011, pp.278-381.
- [7] K.C. Sato *et al.*, “トリガ付きスケーラの応用: J-PARC MR 加速器信号の計測”, Proc. of the 15th Annual Meeting of Particle Accelerator Society of Japan, Nagaoka, Japan, Aug. 8-10, 2018, WEP095, in press.
- [8] T. Kimura *et al.*, “J-PARC MR-MPS の評価と新 MR-MPS の開発計画”, Proc. of the 14th Annual Meeting of Particle Accelerator Society of Japan, Sapporo, Japan, Aug. 1-3, 2017, pp. 1148-1150.