

# SuperKEKB 用アボート・トリガー・システムのタイムスタンプ記録システムの開発 DEVELOPMENT OF TIME STAMP RECORDING SYSTEM FOR SUPERKEKB ABORT TRIGGER SYSTEM

佐々木信哉<sup>#,A)</sup>, 秋山篤美<sup>A)</sup>, 内藤孝<sup>A)</sup>, 中村達郎<sup>A)</sup>

Shinya Sasaki<sup>#,A)</sup>, Atsuyoshi Akiyama<sup>A)</sup>, Takashi Naito<sup>A)</sup>, Tatsuro Nakamura<sup>A)</sup>

<sup>A)</sup> High Energy Accelerator Research Organization (KEK)

## Abstract

The abort trigger system, which aggregates abort request signals and sends a trigger signal to the abort kicker, has been upgraded for SuperKEKB. The response time of the system has improved because of optical transmission of the request signals. In addition, timestamps that is the time when the system received request signals are recorded as EPICS records. The timestamp recording system collects the timestamps via channel access protocol and stores them in a file for each beam abort. It is expected that the cause of beam abort will be analyzed efficiently by the timestamps recorded by the system. Information on the system configuration is managed by the configuration management system, and it generates the configuration files. This paper describes the design of the timestamp recording system and the configuration management system.

## 1. はじめに

SuperKEKB 加速器[1]は KEKB 加速器の 40 倍のルミノシティの実現を目指す電子・陽電子衝突型加速器である。2016 年 2 月から 6 月まで行われた Phase 1 運転[2]では、アボート要求信号を集約して、アボートキッカーへトリガーを送信する役割を持つアボート・トリガー・システム[3]は、SuperKEKB に向けて開発された新システムの一部が更新されて運用された。現在は、Phase 2 以降の運転に向けてシステム全体の更新を行っており、すべてのアボート要求信号が光信号によって伝送されることで、信号を集約してから送信するまでの応答速度が高速化される。

新システムでは応答速度が改善されるほかに、アボート要求信号を受信した時間を、受信した信号ごとにタイムスタンプとして記録する機能を備えている。タイムスタンプによって、アボート要求信号を受信した順序関係を明らかにすることが出来るようになるため、ビームアボートが起こった原因の究明に役立てることができる。

本稿では、アボート・トリガー・システムにおけるタイムスタンプの発行方法とその記録システムについて報告する。また、システムを構成するモジュールの設定情報を管理して、設定ファイルなどを自動生成する管理システムについても併せて報告する。

## 2. アボート・トリガー・システムの構成

アボート・トリガー・システムの構成は Figure 1 に示す通りである。アボート・トリガー・システムはアボート要求信号を集約する部分と、イベントシステムによってアボート要求信号入力モジュール (18K15) のタイミング同期をとる部分から構成される。分散されて設置された機器は EPICS (Experimental Physics and Industrial Control System) [4]によって制御される。

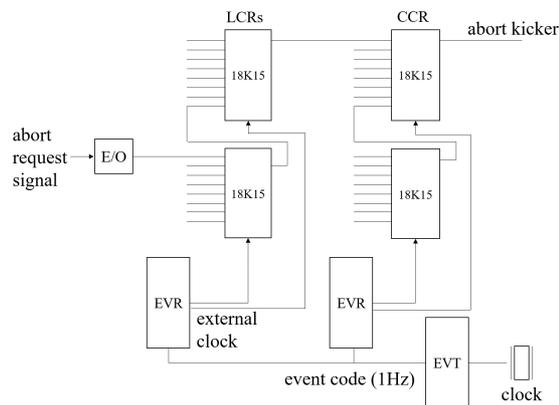


Figure 1: Schematic of the abort trigger system.

アボート要求信号の集約部分では、まず機器から送信されるアボート要求信号が E/O モジュールによって電気信号から光信号に変換されて送信される。送信された信号は、周長約 3 km の加速器リングに沿って点在する Local Control Rooms (LCRs) に設置された 18K15 で一度集約される。各 LCR で集約されたアボート要求信号は再び光信号として Central Control Room (CCR) に送信される。そして CCR に設置された 18K15 で集約され、アボートキッカーへトリガー信号が送信される。LCR から CCR までの伝送距離は最長で約 2 km であり、その伝送時間は約 10  $\mu$ s である。

18K15 間のタイミング同期には KEKB の頃から使用されているイベントシステム[5]を使用している。このイベントシステムは各地に点在する機器や IOC (Input/Output Controller) の同期に利用されている。ここでは、タイミング同期をとるための外部クロック分配として利用している。

イベントシステムから外部クロックとして信号を受け取る 18K15 は、内部に 2 つのカウンタを持っている。1 つは 10 MHz の内部クロックでカウントアップする 10 MHz カウンタである。このカウンタは外部クロックを受信すると 0 で

<sup>#</sup> shinya.sasaki@kek.jp

リセットされ、再びカウントアップを行う。もう 1 つは外部クロックを受信するとカウントアップする外部クロックカウンタである。モジュールがリセットされると、2 つのカウンタは 0 にリセットされる。

18K15 はアボート要求信号を受信すると、受信したチャンネルごとに 2 つのカウンタの値をモジュール内のレジスタに記録する。チャンネルごとに 2 つのカウンタの値を比較することによって、信号を受信した順序関係を明らかにすることが出来る。

### 3. タイムスタンプの発行方法

アボート要求信号を受信した相対的な順序関係は、チャンネルごとに記録される 10 MHz カウンタと外部クロックカウンタの値を比較することで明らかにすることができる。しかし、カウンタの値だけでは、信号を受信した日時を割り出すことはできない。ビームアボートが発生した原因の解析に利用することを考えると、カウンタの値だけでなく、日時形式のタイムスタンプに置き換えた方が理解しやすい。ここでは、モジュールのレジスタに記録された 2 つのカウンタの値から、日時形式のタイムスタンプを発行する方法について説明する。

日時形式のタイムスタンプを発行するには、イベントシステムから外部クロックを受信した日時が必要となる。外部クロックの受信日時が分かれば、その日時に 10 MHz カウンタの値を足した日時が信号を受信したタイムスタンプとなる。しかし、各地に分散設置された IOC ごとに外部クロックの受信日時を算出する場合、IOC ごとの時計のずれがそのままタイムスタンプのずれに反映されてしまう。そのため、受信日時の代わりに CCR から外部クロックが送信された日時を記録することで、単独の IOC の時計のみが使用されるようにした。実際には、CCR 内で外部クロックを送受信する時間は無視できるほど小さいため、CCR の IOC が外部クロックを受信した日時を外部クロックの送信日時として使用している。また、外部クロックを CCR から送信して LCRs で受信するまでの時間をあらかじめ測定・記録しておき、タイムスタンプ発行の際にその時間を足すようにした。これにより、イベント送信にかかる時間を補間できる。

上述のタイムスタンプの発行は EPICS レコードを用いて実装した。CCR に設置された IOC では、外部クロックを受信した際の割り込み処理で、外部クロックを受信した日時と、これまで受信した外部クロックの数をレコード化する。一方、LCRs に設置された IOC では CCR の IOC でレコード化された外部クロック受信日時、および外部クロックの数を Channel Access (CA) プロトコル経由で参照し、アボート要求信号の受信チャンネルごとにタイムスタンプの計算を行う。CA プロトコルでは複数のレコードを参照した場合のアトミック性が保証されていないため、CCR の IOC では 1 つの waveform レコードの array の中に外部クロック受信日時と外部クロック数を格納するようにして、LCRs の IOC からアトミックに参照できるようにしている。

### 4. タイムスタンプ記録システム

アボート要求信号の受信チャンネルごとのタイムスタンプは、EPICS レコードとして発行される。そのため、過去

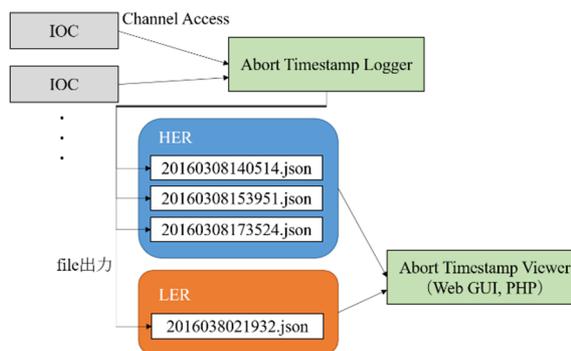


Figure 2: Schematic of the timestamp recording system.

のタイムスタンプの情報を参照するためには、レコードの値を記録しておく必要がある。

EPICS レコードの値を記録するためのアーカイバはいくつか開発・運用が行われている。例えば、SuperKEKB では KEKBLog[6]をメインのアーカイバとして使用しているほか、Channel Archiver[7]や CSS (Control System Studio)[8] に付属する Archiver (BEAUTY) を試験的に運用した実績[9]もある。しかし、上記のアーカイバはレコード毎にその値を時系列的に保存するため、発生したビームアボート毎にタイムスタンプを閲覧したいような場合には、アーカイブされたデータからレコード間の情報を比較して、情報の再構築が必要となる。そのため、発生したビームアボート毎にタイムスタンプの情報を記録するタイムスタンプ記録システムを開発した。

タイムスタンプ記録システムの構成は Figure 2 に示す通りである。システムはタイムスタンプ情報を収集し、ファイルに記録する Abort Timestamp Logger と、ファイルに記録されたタイムスタンプ情報を Web から閲覧するための Abort Timestamp Viewer で構成される。

Abort Timestamp Logger は Python によって実装されたタイムスタンプ収集・記録システムである。PythonCA を使用して、各 IOC の持つタイムスタンプのレコードや、アボート要求信号の入力状況を示すレコードなどを CA プロトコル経由でモニターする。アボート要求信号を受信してから、すべてのアボート要求信号が解除されるまでを 1 回のビームアボートとして、ビームアボート毎にタイムスタンプやその信号名などの情報をファイルに記録する。

#### Abort Timestamp Log Viewer (Manager mode)

2016/05/29

RING	MESSAGE	DATE	DELTA
LER	RF D8-A	2016-05-29 01:04:02.429921000	0.000000000
LER	COLSAFE:CCC:ABORT:CCC-6	2016-05-29 01:04:02.429922400	0.000001400
LER	COLSAFE:CCC:ABORT:D8	2016-05-29 01:04:02.429922400	0.000001400
LER	COLSAFE:CCC:ABORT:D7	2016-05-29 01:04:02.430041700	0.000120700
HER	COHSAFE:CCC:ABORT:D11	2016-05-29 01:04:02.432430800	0.002509800
HER	COHSAFE:CCC:ABORT:CCC-2	2016-05-29 01:04:02.432431100	0.002510100
HER	COHSAFE:CCC:ABORT:D7	2016-05-29 01:04:02.432523600	0.002602600
HER	COHSAFE:CCC:ABORT:D10	2016-05-29 01:04:02.434897100	0.004976100
LER	Soft Abort	2016-05-29 01:04:04.383276700	1.953355700
LER	Soft Abort	2016-05-29 01:04:04.516567000	2.086646000

Figure 3: Screenshot of the user interface for timestamp recording system.

ファイルへの出力フォーマットは JSON (JavaScript Object Notation) を採用し、タイムスタンプの時間が早い順に情報をソートしてファイルに出力する。ファイルの出力先はリングごとに別々にしている。

Abort Timestamp Viewer は PHP によって実装された、web から過去のタイムスタンプ情報を閲覧するためのユーザインタフェースである。Abort Timestamp Logger によって記録されたファイルを読み込み、要求信号ごとのタイムスタンプと、始めに受信した信号からの時間差を表示する。表示画面のスクリーンショットを Figure 3 に示す。

## 5. 設定情報管理システム

アボート・トリガー・システム全体の構成情報を一元管理し、必要となる設定ファイルなどを自動生成するために、設定情報管理システムを開発した。システムの構成は Figure 4 に示す通りである。

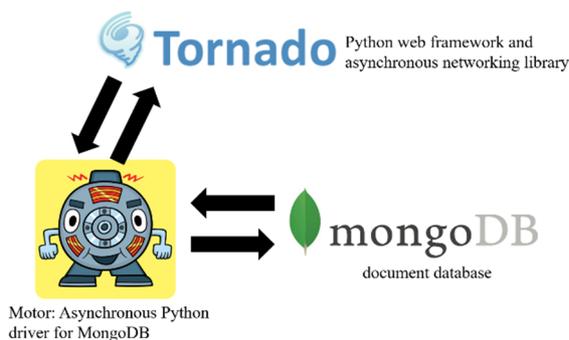


Figure 4: Schematic of the configuration management system.

バックエンドデータベースには MongoDB[10]を採用し、モジュールごとにその設定情報が記録される。フロントエンドには Python の web フレームワークの 1 つである Tornado[11]を利用して、web 経由で設定を行うことができるようにした。管理画面のスクリーンショットを Figure 5 に示す。管理画面から、アボート要求信号受信モジュールごとに信号名や、信号の簡単な説明、アラームシステム[12]で表示するメッセージ、縦続接続しているモジュールに関して確認・変更ができる。

設定ファイルの生成はフロントエンドとは別の Python スクリプトから MongoDB にアクセスして行う。生成する設定ファイルは EPICS データベースやアラーム登録リスト、タイムスタンプ記録システムに必要な情報などである。

CCC

id	CCC-1	id	CCC-2								
Ring	HER	Ring	HER								
Delay	0	Delay	0								
Node	3	Node	4								
CH	Name	ENB	child	Description	Message	CH	Name	ENB	child	Description	Message
ch1	D1	✓	D1-1	D1		ch1	D8	✓	D8-1	D8	
ch2	D2	✓	D2-1	D2		ch2	D10	✓	D10-1	D10	
ch3	D4	✓	D4-1	D4		ch3	D11	✓	D11-1	D11	
ch4	D5	✓	D5-1	D5		ch4	D3_VA1	✓	None	D3 Vacuum	Vacuum D3-1
ch5	D7	✓	D7-1	D7		ch5	D6_VA1	✓	None	D6 Vacuum	Vacuum D6-1
ch6	MANUAL	✓	None	manual abort	Manual Abort	ch6	D9_VA1	✓	None	D9 Vacuum	Vacuum D9-1
ch7	SOFT	✓	None	soft abort	Soft Abort	ch7	D12_VA1	✓	None	D12 Vacuum	Vacuum D12-1
ch8	CCC-2	✓	CCC-2	CCC-2		ch8	CCC-3	✓	CCC-3	CCC-3	

Buttons: Edit, Delete

Figure 5: Screenshot of configuration management system.

## 6. まとめ

SuperKEKB のアボート・トリガー・システムは Phase 2 以降の運転に向けて、システム全体がタイムスタンプ機能を持つ新しいシステムに更新された。アボート要求信号受信時に EPICS レコードとして発行されるタイムスタンプは、タイムスタンプ記録システムによって記録され、web からの閲覧が可能となった。システムを構成するために必要となる情報は設定情報管理システムで一元管理され、必要な設定ファイルが生成される。本システムによって記録されたタイムスタンプによって、今後の SuperKEKB 加速器の運転においてビームアボートの原因解析がより効率的に行われることが期待される。

## 参考文献

- [1] Y. Ohnishi *et al.*, “Accelerator design at SuperKEKB”, Prog.Theor. Exp. Phys. (2013) 03A011; <http://ptep.oxfordjournals.org/content/2013/3/03A011.full.pdf>
- [2] T. Kamitani *et al.*, “SuperKEKB Phase1 (Injector+Ring) Status Report”, Proceedings of the 13th Annual Meeting of Particle Accelerator Society of Japan, Chiba, Aug. 8-10, 2016, MOOLP02; [http://www.pasj.jp/web\\_publish/pasj2016/proceedings/PDF/MOOL/MOOLP02.pdf](http://www.pasj.jp/web_publish/pasj2016/proceedings/PDF/MOOL/MOOLP02.pdf)
- [3] S. Sasaki *et al.*, “Development of Abort Trigger System for SuperKEKB”, Proceedings of the 11th Annual Meeting of Particle Accelerator Society of Japan, Aomori, Aug. 9-11, 2014, SUP093; [http://www.pasj.jp/web\\_publish/pasj2014/proceedings/PDF/SUP0/SUP093.pdf](http://www.pasj.jp/web_publish/pasj2014/proceedings/PDF/SUP0/SUP093.pdf)
- [4] <http://www.aps.anl.gov/epics/>
- [5] T. Naito *et al.*, “Performance of the Timing System for KEKB”, Proceedings of the ICALEPCS’99, Italy, Oct. 4-8, 1999, pp. 214-126; <http://accelconf.web.cern.ch/AccelConf/ica99/papers/mc1p17.pdf>
- [6] T. T. Nakamura *et al.*, “Data Archiving System in KEKB Accelerators Control System”, Proceedings of the 10th ICALEPCS, Geneva, Oct. 2005, PO1.077-7; [http://accelconf.web.cern.ch/AccelConf/ica05/proceedings/pdf/P1\\_077.pdf](http://accelconf.web.cern.ch/AccelConf/ica05/proceedings/pdf/P1_077.pdf)
- [7] K.U. Kasemir *et al.*, “Overview of the Experimental Physics and Industrial Control System (EPICS) Channel Archiver”, Proceedings of the 8th ICALEPCS, San Jose, Nov. 2001; <http://accelconf.web.cern.ch/AccelConf/ica01/papers/THAP019.pdf>
- [8] <http://controlsystemstudio.org/>
- [9] M. Hirose *et al.*, “Present status of CSS Archiver and Channel Archiver at SuperKEKB”, Proceedings of the 13th Annual Meeting of Particle Accelerator Society of Japan, Chiba, Aug. 8-10, 2016, TUP091; [http://www.pasj.jp/web\\_publish/pasj2016/proceedings/PDF/TUP0/TUP091.pdf](http://www.pasj.jp/web_publish/pasj2016/proceedings/PDF/TUP0/TUP091.pdf)
- [10] <https://www.mongodb.com/>
- [11] <http://www.tornadoweb.org/en/stable/>
- [12] T. Nakamura *et al.*, “Operation Status of CSS Alarm System for SuperKEKB”, Proceedings of the 13th Annual Meeting of Particle Accelerator Society of Japan, Chiba, Aug. 8-10, 2016, TUP095; [http://www.pasj.jp/web\\_publish/pasj2016/proceedings/PDF/TUP0/TUP095.pdf](http://www.pasj.jp/web_publish/pasj2016/proceedings/PDF/TUP0/TUP095.pdf)