福井大学リポジトリ
UNIVERSITY OF FUKUI REPOSITORY

μTCA 及びDAQ-Middlewareを用いたデータ収集機構の開発

メタデータ	言語: ja
	出版者:
	福井大学工学部技術部技術部活動報告集編集委員会
	公開日: 2023-08-18
	キーワード (Ja):
	キーワード (En):
	作成者: 戸澤, 理詞, 伊藤, 雅基
	メールアドレス:
	所属:
URL	http://hdl.handle.net/10098/0002000033

μTCA 及び DAQ-Middleware を用いたデータ収集機構の開発

戸澤 理詞* 伊藤 雅基*

1. 研修背景・目的

μTCA は，コンピュータの物理的・論理的仕様を定めた規格である ATCA 規格から派生した，小型のボードの仕様を定めた規格であり，高エネルギー物理分野で広く使用されている．派遣先研究室である素粒子・原子核実験グループでは，これまでも μTCA 規格に準拠したハードウェアによるデータ収集が行われてきたが，セットアップの一部を変更したことにより，プログラムの更新が必要となった．

本研修では，ソフトウェアとして DAQ-Middleware を利用し，放射線検出器からのデータ収集機構を開発することを目的として，プログラムの作成を行った．

2. データ収集環境

2.1 ハードウェア

2.1.1 ATCA と μTCA

通信事業者向けコンピュータの物理的・論理的な仕様を定めた規格である ATCA（Advanced Telecommunications Computing Architecture）規格は，データ転送の並列化や回路の多チャンネル化が可能といった特長がある．一方で，サイズや消費電力の大きさのため，導入に係る初期コストが大きいという問題があった．そこで，小型のボードのみでシステムを構成できるように，μTCA（Micro Telecommunications Computing Architecture）規格が策定された．

ATCA と μTCA の構成を図 1 に示す．ATCA シェルフは，ATCA ブレードを挿入できるように設計されており，ATCA 向けメザニンカードの規格である AMC（Advanced Mezzanine Card）を使用する場合は，キャリアブレードに装着した上で挿入する．μTCA は，この AMC を，キャリアブレードを介さずに直接シェルフに挿入できる規格である．

2.1.2 ハードウェアセットアップ

今回，開発を行ったハードウェアのセットアップを図 2 に示す．μTCA シェルフは，MCH（MicroTCA Carrier Hub）1 台と AMC モジュールを最大 6 台まで挿入できる．MCH は，Shelf Management Controller 機能と LAN スイッチ機能を備えたもので，これにより，シェルフの制

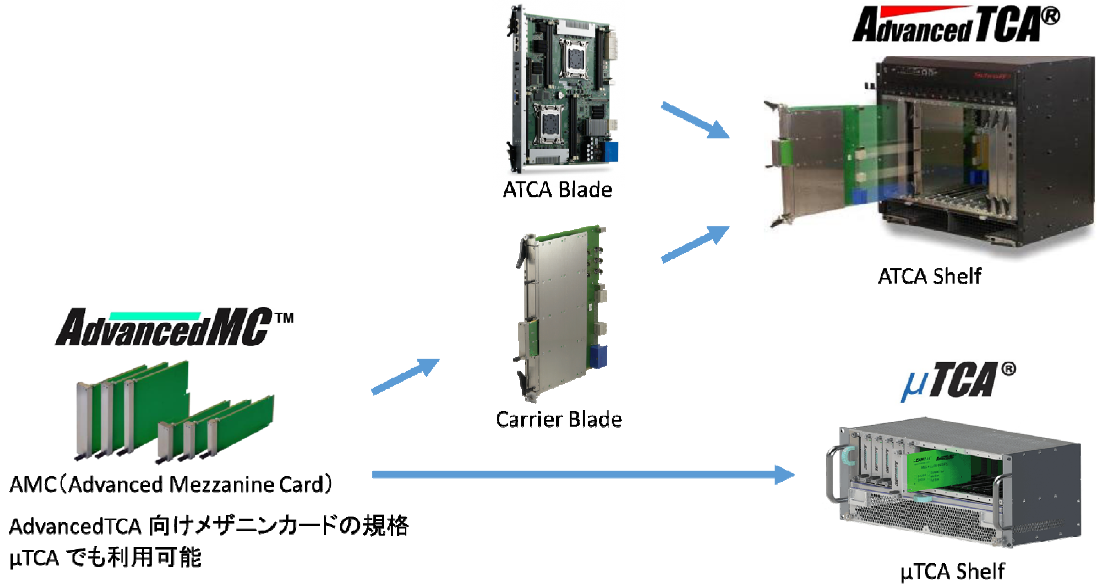

図 1 ATCA と μTCA の構成

* 第 2 技術室 物理計測班

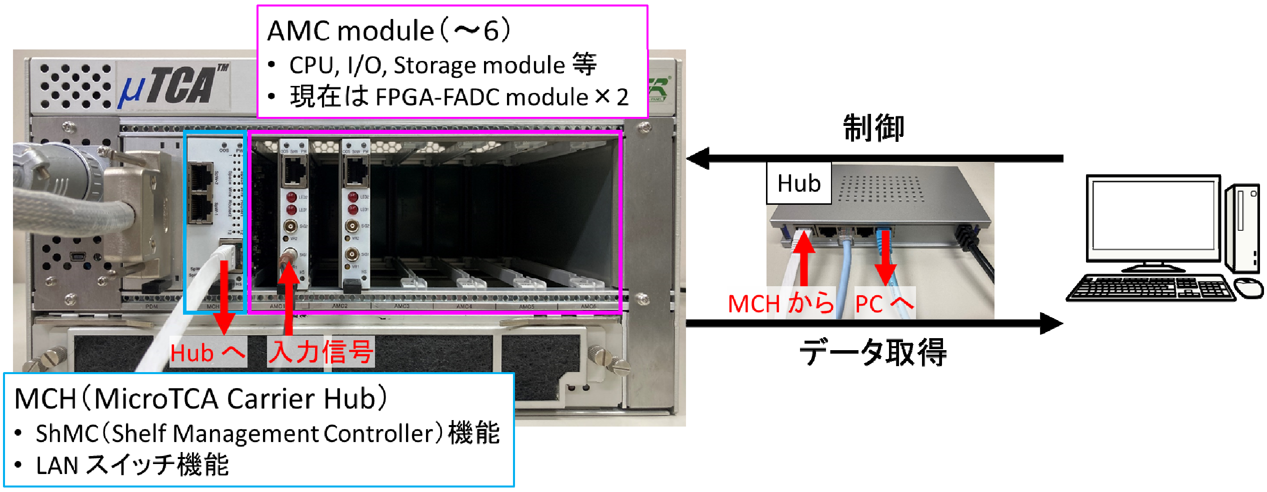

図 2　ハードウェアセットアップ

御及び PC との通信を行う．AMC モジュールとしては，現在は FPGA-FADC モジュールを 2 台挿して，最大 4ch の信号を同時に取得できるようになっている．

2.2 ソフトウェア

2.2.1 DAQ-Middleware の概要

DAQ-Middleware は，DAQ コンポーネントと呼ばれるソフトウェア・コンポーネントを組み合わせることで，容易にシステムを構築することができるソフトウェア・フレームワークであり，起動後に取る状態や状態間の遷移で呼び出される関数を，各コンポーネントに実装することで構築していく．

DAQ-Middleware の概要図を図 3 に示す．DAQ-Middleware では，複数の DAQ コンポーネントでデータ収集を行い，DaqOperator でコンポーネントを統括する．DaqOperator は，XML 形式で書かれたシステムコンフィギュレーションファイルを読み込むことで，使用するコンポーネントやコンポーネント間の接続情報を把握し，他の上位システムからの指示によって動作する．ラン毎に変わるパラメータは，コンディションファイルとして XML 形式で記述した後，コンポーネントにかかる構文解析処理の負荷を軽減するため，JSON 形式に変換しておく．各コンポーネントは，JSON 形式で書かれたファイルを読み込み，パラメータを取得する．

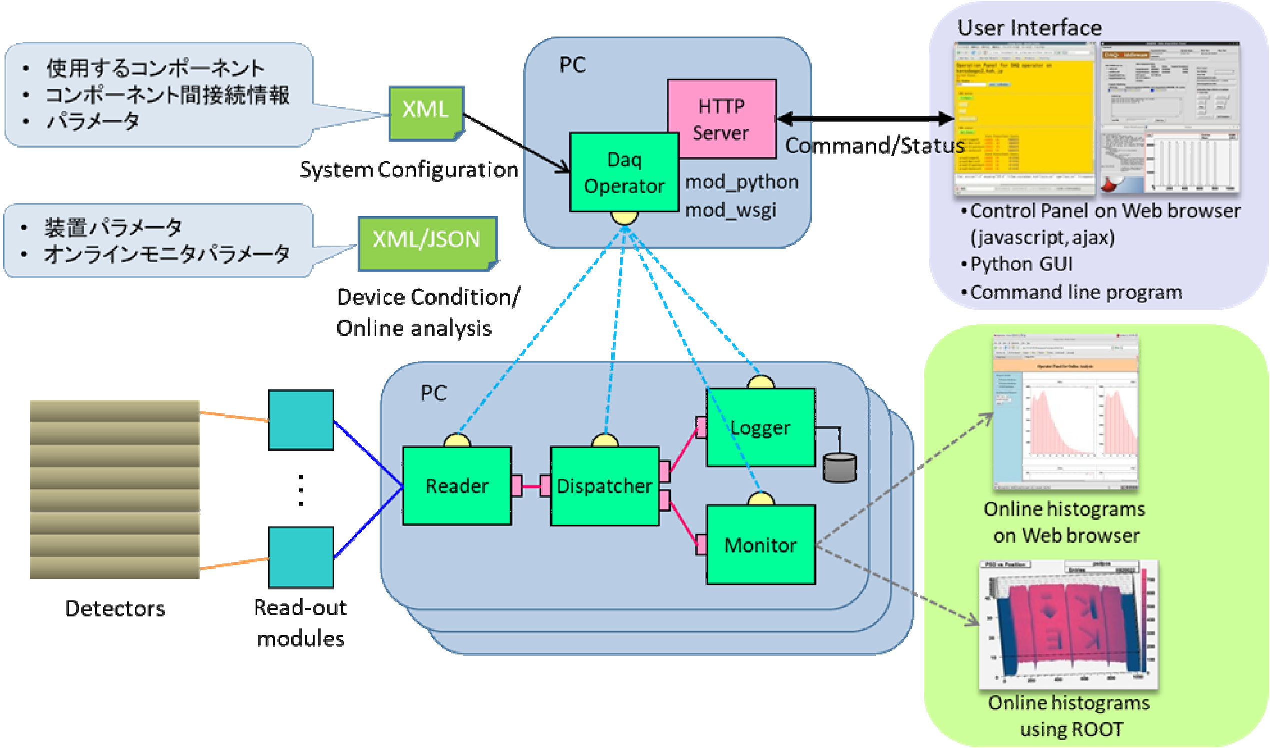

図 3　DAQ-Middleware 概要図

2.2.2 DAQ コンポーネントの構成

現在，派遣先研究室で組んでいる DAQ コンポーネントの構成図を図 4 に示す．

FADC で取得したデータを読み取る MtcaReader，データを Logger（全データ）と Monitor（best effort）へ送る MtcaDispatcher，データをファイルに書き込む MtcaLogger，データから図を作りオンラインモニターに表示する MtcaMonitor の 4 つのコンポーネントで構成されており，各コンポーネントに接続された DaqOperator から，データ収集開始，終了等のコマンドを送ることで, ランコントロールを行う．

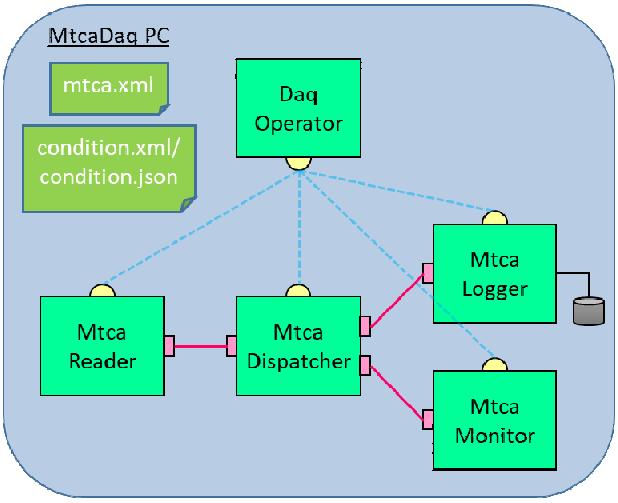

図 4　DAQ コンポーネント構成図

2.2.3 コンポーネントの状態及び遷移

DAQ-Middleware における状態遷移図を，MtcaReader で呼び出す関数とともに図 5 に示す．各コンポーネントは起動している間，図 5 中央に示した状態（LOADED，CONFIGURED，RUNNING，PAUSED）の内のいずれかの状態にあり, DaqOperator からコマンドを送ることにより，状態間を遷移する．

各状態及び状態間の遷移には関数が定義されている．一つの状態にある間は，その状態における関数が繰り返し実行され，遷移する際は，その遷移に定義された関数が一度だけ実行される．

例えば，コンポーネントがプロセスとして走り出した直後は LOADED 状態であり，CONFIGURE コマンドを送ると daq_configure() 関数が一度だけ実行され，CONFIGURED 状態に遷移する．START コマンドを送ると RUNNING 状態に遷移し，次に PAUSE または STOP コマンドを送るまで，daq_run()関数が繰り返し実行される．

DAQ-Middleware では，各コンポーネントにこれらの関数を実装することで，プログラムを構築していく．

3. プログラムの編集

図 4 に示した 4 つのコンポーネントの内，今回のセットアップ変更に関係するのは MtcaReader と MtcaMonitor であり，プログラムの編集は主にこの 2 つのコンポーネントについて行った．

3.1 MtcaReader コンポーネント

FADC で取得したデータを読み取る MtcaReader については，データ取得を行うサンプルプログラムの内容を関数の形にして，図 5 の吹き出しで示す位置で呼び出すようにプログラムを作成した．

サンプルプログラムの内容は，①ルーターの設定，②FADC の設定，③トリガーモジュールの設定，④データ収集，⑤データ収集の停止で

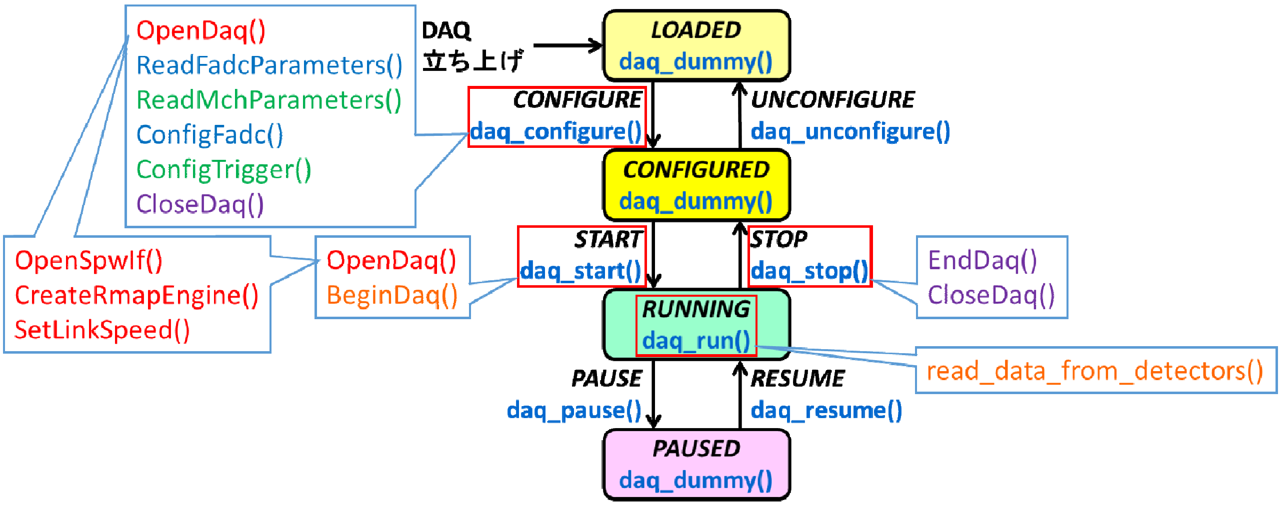

図 5　DAQ-Middleware 状態遷移図

あり，それぞれで行っている内容を細分化した上で関数化し，各呼び出し位置に振り分けた．

プログラムとしては，一つの波形データに対して，DAQ コンポーネントのシーケンス番号，Request Trigger，Accept Trigger，タイムスタンプ等の情報をヘッダーに入れ，後段のコンポーネントへ送るようにした．限られたデータ量の中で必要な情報を効率的に送るため，波形データに関して，前半は 1 点 2 nsec のデータ，後半は複数点の和のデータとして取得している．前半の点数及び後半の足し合わせる点数については，パラメータとして変更可能であり，現在は，前半を 256 点 (512 nsec) 分，後半を 32 点 (64 nsec) の和のデータとして取得するように設定している．

3.2 MtcaMonitor コンポーネント

データから図を作りオンラインモニターに表示する MtcaMonitor については，以前のセットアップで使用していたプログラムの内容解読と計算方法の確認を行い，オンラインモニターで表示していたものを再現するようにプログラムを作成した．オンラインモニターの描画には，CERN によって開発が行われている，データ解析環境及び関連するライブラリ群である ROOT を採用した．

以前のプログラムについて内容解読を行ったところ，オンラインモニターには，①波形データの前半（細かい波形データ（2 nsec/点）），②波形データの全体（細かい波形データ+粗い波形データ（複数点の和）），③トリガーレート，④イベント間の時間差，⑤ADC 値を表示していたことがわかった．それぞれについて計算方法の確認を行い，図 6 に示すようにオンラインモニター画面を再現した．

①波形データの前半及び②波形データの全体については，2 nsec/bin のヒストグラムに，取得した波形データの値を詰めていくことで再現した．複数点の和としてデータを取得している後半については，足し合わせている点数で割ることで平均値を求め，その値を点数分詰めるようにした．

③トリガーレート及び④イベント間の時間差については，ヘッダーに入れた Request Trigger, Accept Trigger，タイムスタンプの情報を使用し，それぞれ一つ前のイベントとの差を計算することで値を求め，トリガーレートは時間変化に対する値，時間差はヒストグラムに詰めた結果として表示した．

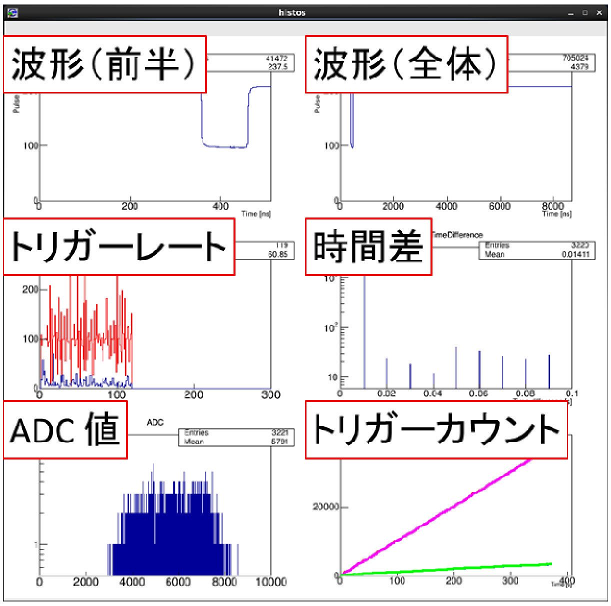

図 6　オンラインモニター画面

⑤ADC 値については，波形の取り始めから 50 点目までの平均値をペデスタルとして求めた上で，波形の前半と後半それぞれで，ペデスタルとの差の積分値を計算し，その和をヒストグラムに詰めることで表示した．

また，③トリガーレートについて，より詳細に調査できるようにするため，トリガーカウント（Request Trigger，Accept Trigger）の時間変化も表示するようにした．

4.　まとめ

μTCA 規格に準拠したハードウェアによるデータ収集機構において，新しいセットアップに対応したプログラムの開発を行った．ソフトウェアとしては DAQ-Middleware を利用し，主に MtcaReader コンポーネントと MtcaMonitor コンポーネントについて，以前のプログラムに編集を加えることで，新しいセットアップでのデータ収集を実現することができた．

謝辞

本研修を実施するにあたり，データ取得を行うサンプルプログラムを作成いただいた，大阪大学放射線科学基盤機構　能町正治特任教授に深く感謝申し上げます．

参考文献

[1] 千代浩司，DAQ-Middleware 1.4.4 開発マニュアル，2019 年 7 月 https://daqmw.kek.jp/docs/DAQ-Middleware-1.4.4-DevManual.pdf