デジタルシステムの各サブグループへの配布

2010/10/25

Digital system group:

宮川、三代木、大石、斎藤(紀)、麻生、和泉

1. 目的

　LCGTにはデジタルシステムが導入され、制御やモニタ等に使用される[1]が、その段階的な導入の準備の一環として、各サブグループにできるだけデジタルシステムを組み込んだ状態で要素技術を開発してもらう。これはデジタルシステムに慣れてもらうことと、開発した物がLCGTにほぼそのまま組み込むことができるので、LCGTへの要素技術の導入をスムーズにするためである。

2. 概要

　LCGTの予算化が認められたが、LCGTサイトへの実際のデジタルシステムのインストールまでに、デジタルシステムを組み込んでの要素技術開発を必要とする各サブグループに対し、デジタルシステムサブグループからCLIOに組み込んだ試験システムと同等のものを提供する。初年度の予算規模にもよるが、5台程度、すなわち5サブグループ程度への配布を考えている。各サブグループで構築されたシステムは、LCGT本体のデジタルシステムの一部としてそのまま導入される。

2.1 デジタルグループからの提供範囲

* リアルタイム用計算機、及び信号入出力ハード一式
* 信号フィルタリング用アナログ回路一式
* リアルタイム制御に必要な基本ソフト(データモニタ、スイッチング等のソフトも含む)
* Client用計算機一式

これらを5セット程度製作し、ラックに入れた状態で配布する予定

2.2 各サブグループで用意してもらうもの

* 各実験装置とシステムを接続するドライバー
* オシロスコープ等の測定器や、各グループで個別に必要な実験装置や回路

3. 仕様

3.1 Hardware

　配布される予定のシステムの詳細を下記に示す。また、その主な性能を表1にあげておくが、これらの値はCLIOに試験システムを導入したときに測定したデータ等を元にしている。

　実際に各サブグループに導入されるシステムは図1に示された写真のようなラックに入れられた形で配布される予定である。

* PC: SUPERMICRO社製 1DIN server, 2x4core 1台
  + CentOS + Real Time Linux (OSがGentooに変更になる可能性有り)
* PCIe Expansion chassis: One Stop System社製、OSS-PCIe-4U-EXP-2001-700 1台
* Analog Digital Converter (ADC): Genaral Standards社製、16bit 64ch(32ch for diff.)、PMC66-16AI64SSA-64-50MHz-MEM 1 or 2枚
* Digital Analog Converter (DAC): Genaral Standards社製、16bit 32ch(16ch for diff)、PMC66-16AO16-16-F0-DF-MEM 1 or 2枚
* Binary Output (BO): CONTEC社製、32ch、DO-32L-PE 1 or 2枚
* Timing system(Columbia大学を通して手配済み) 1台
* 付随するアナログ回路:
  + Interface box for ADC/DAC/BO 必要なチャンネル数
  + Anti Aliasing/Imaging filters 必要なチャンネル数
  + Whitening/dewhitening filter + Variable gain amplifier 必要なチャンネル数

|  |  |  |
| --- | --- | --- |
|  | **Item** | **Requirements** |
| \* | Sampling rate | 16384Hz(sampled at 65536Hz and decimated to 16384Hz)  option:　up to 65536Hz for limited number of channels |
|  | Number of channels | 32ch / 1 ADC card  16ch / 1 DAC card  32ch / 1 BO card |
|  | Maximum card number | Total number of ADC, DAC, BO = 14 |
| \* | ADC bit resolution | 16bit = 65536 |
| \* | ADC dynamic range | Full differential +/-10V = effective range +/-20V |
| \* | DAC dynamic range | Full differential +/-5V = effective range +/-10V |
| \* | Through delay | 80usec |
| \* | ADC noise level | 2uV/rHz |
| \* | DAC noise level | 1.5uV/rHz |
| \* | AA filter noise level | 0.1u V/rHz |
| \* | AI filter noise level | 0.1u V/rHz |
| \* | Whitening filter noise level | 1n V/rHz |
| \* | Dewhitening filter noise level | 1n V/rHz |
|  | Connector shape | D-SUB9 @ ADC in, DAC out, BO, AA in, AI out |

表1:Hardwareの性能

図1:上からReal time計算機、AI filters(D-sub 1段), AAfilters（1段）, DAC interface（1段）, ADC interface（2段）, BO interface（1段）, Expansion chassis(この中にADCやDACが入っている)

3.2 Software

　基本的なソフトは全て導入された状態でシステムが配布される。システム全体のための資料[10]にも示したが、以下のようなソフトが走っている。

* Matlab and Sumulink: real time code generator
* CDS software
  + RTFE: リアルタイムコード
  + DAQS: データ取得サービス
  + EPICS: channel name 等の割当
  + MEDM: EPICSを利用したヒューマンインターフェース
  + AWG: Arbitrary Wave form Generator
  + DTT: Diagnostic Test Tool, FFT, Sine response, Swept sine, Trigger time response等
  + foton: digital filter生成ツール
  + dataviewer: 多チャンネル、簡易オシロスコープ
  + ezca: スクリプト用自動制御コマンド群、ezcaread, ezcawrite, ezcastep, ezcaservo, ezcademod等
  + burt: 干渉計パラメータ記録、再現ツール

　導入してからすぐに触る必要があるものは、一番最上のMatlabとSimulinkを利用してのリアルタイムコードを実際に生成する部分であろう。ここはGUIであるので、サンプルファイル等を見ればすぐにビルドまでできる。リアルタイムコードは変更のたびにシステムの再スタートを必要とするので、毎日変更するという類いの物ではない。

　そのかわりにMEDMと呼ばれるヒューマンインターフェース用のソフトを通して干渉計を操作する。具体的には、各コンポーネントのMEDMスクリーン(GUI)を作る/もしくは自動生成され、そこにゲインの調整や、フィルターのon/offなどのスイッチを組み込む。

　例えば、一つのデジタルフィルターモジュールに一つのMEDMスクリーンが存在して、図3のようになっている。フィルターモジュールは10のフィルターバンクを持ち、それぞれのフィルターバンクに複数のフィルターを設定できるので、非常に多くのフィルターを持つことができる。

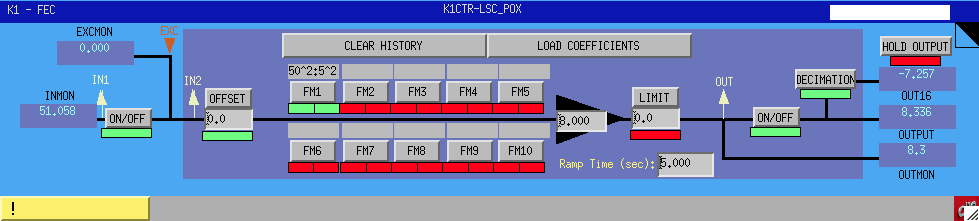


図3: デジタルフィルターモジュールのMEDMスクリーン。

　フィルターの設計にはfotonと呼ばれるソフトが使われ、GUIを用い見通しの良いフィルター設計が非常に簡単にできる。

　ADCからの信号はオシロスコープに相当するDataviewerというソフトと、スペアナに相当するDTTと呼ばれるソフトで見る。DTTは伝達関数測定等も可能である。

　最初に使うソフトはこれくらいである。少し使い込んでいくとスクリプト等でルーチンワークを操作したいという欲求が出てくるので、そのときはezcaなどのコマンドをUNIXのshell上で使うことができるが、ここでは詳しくは述べない。

4 デジタルシステムにつなぐためのドライバーの製作

　デジタル装置の出力は基本的にD-sub 9pinコネクターの形で提供され、その中の8本を差動で使うために1コネクター当たり4つの信号が扱えることになる。D-subケーブル内の信号は差動信号であるため、数10m程度までは延ばすことができる。

　そこから先の実験装置に接続するためのドライバーの製作は基本的に各サブグループの担当としてもらう。一つの箱に、D-subの端子と、実験装置を結ぶ回路を組み込む等する。

　差動信号を受け取り、シングルエンドに変換するための回路図などは、デジタルグループから回路サブグループを通して提供する。また、ゲインを調整するためのアンプや、whitning/dewhitening filterは汎用の物を用意する予定であるが、ドライバーの中に組み込みたという要請も考えられるために、その場合はデジタルグループが回路図を提供し、各グループで組み込んでもらう。

　各サブグループがデジタル機器と各自の実験装置をつなぐときに、コネクター部分として直接関わる回路は

* Binary Output receiver
* Differential signal driver for ADC
* Differential to single receiver for DAC

の3つである。回路図等の情報はCLIOで製作した物が、上記3つも含めて全て

<http://gw.icrr.u-tokyo.ac.jp/JGWwiki/CLIO/Tasks/DigitalControl/pban_files>

にアップロードされている。また、ゲインを調整するためのアンプや、whitning/dewhitening filter等を独自にドライバーに組み込みたい場合も上記ページに回路図等がある。なおこれらはLCGT用に表面実装回路を使う等して、もう少しコンパクト化され、改良される予定である。

5. System構築例

5.1 Hardware

　ハード部分は基本的にセットアップが完了された状態で配布される。

5.2 Software

　基本ソフトも各グループに共通な部分は全てインストールされた状態で配布される。各グループ固有のリアルタイムコード、MEDMと呼ばれるヒューマンインターフェース部分のスクリーン、各種スクリプト等は配布されてから各グループで書いてもらうことになる。最初の基本的なリアルタイムコードは、雛形のような物をデジタルグループで用意することになると考えている。その際、各グループにも基本を理解してもらい、その後各自で拡張できるようになってもらう。実際に引き渡すまでに、詳細なマニュアルを用意する予定である。

5.3 Network構成

　デジタルシステムはnetwork routerを用いたプライベートネットワーク内で動作する。ルータにはReal time用のPCとClient用がぶら下がる。Client PCは複数台つなぐことができ、遠隔操作のための無線LANでの運用もサポートする。Client用PCは一台はおそらくデジタルサブグループ側で用意することになる予定であるが、複数台の場合は配布先のサブグループで用意してもらうこともある。Client用の計算機は図4のような形で運用される。小さいが画面の中にMEDMやDTT等のスクリーンが示されている。写真にあるように3台程度のモニタがあると操作性が上がるので、予算が許す限り多数モニタを提供する方向で行きたいと考えている。



図4:CLIOで実装されたClient systemの例

6. 各サブグループに置けるデジタルシステムの使用例(案)

* 防振系: ローカルコントロール（何らかのセンサー、コイルマグネットアクチュエータ）
* 入射/出射光学系: MC制御、周波数安定化キャビティーの温度制御、周波数安定化サーボ、PMC, OMCなどの制御のスイッチ
* 干渉計グループ:PD, QPD, I&Q demodulator
* 真空系: 真空ポンプのスイッチ
* 冷却系: 温度計モニタ、冷凍機のスイッチ
* Baseline: 各種環境モニタ
* レーザー: Thermal control

　天文台、三尾研、KEK、本郷、柏など、場所で配布するという手もある。予算、台数が限られているので各サブグループからの希望を聞きながら優先順位を考えて調整してきたいと考えている。

7. 有用なドキュメント

　LCGTのデジタルシステム全体を見渡した文書が、本文書に先立って書かれているので、参考にして欲しい。

[1] “LCGT digital system -システム全体- ” [JGW-T1000208](http://gw.icrr.u-tokyo.ac.jp/cgi-bin/DocDB/ShowDocument?docid=208)

　本デジタルシステムはLIGOとの共同開発であたっているので、詳細の理解に当たってはLIGOの文書が役に立つことも多い。ここに、有用な文書をいくつか挙げておく。中にはパスワードがかかっているものもあるが、MOUが進めばこれら文書は公開されるということである。

Real time codeをMatlab上のGUIで構築するためのReference Manual

[2] AdvLigo CDS Realtime Code Generator (RCG) Application Developer’s Guide, [LIGO-T080135](https://dcc.ligo.org/cgi-bin/private/DocDB/ShowDocument?docid=1033)

Real time systemを構築する場合の説明(開発者向け)

[3] AdvLigo CDS Realtime Software SysAdmin Guide, [LIGO-T080136](https://dcc.ligo.org/cgi-bin/DocDB/ShowDocument?docid=1032)

タイミングシステムに付いての説明

[4] Quick Start Guide Optical Timing Distribution System Advanced LIGO, [LIGO-E080541](https://dcc.ligo.org/cgi-bin/DocDB/ShowDocument?docid=485)

LIGOでデジタルシステムを使ってSuspention systemを構築した際の具体例

[5] BSC Suspension Test Stand Electronics and Controls Quick Start Guide, [LIGO-T080321](https://dcc.ligo.org/cgi-bin/DocDB/ShowDocument?docid=8950)

デジタルシステムをCLIOでセットアップしたまとめが

[6] <http://gw.icrr.u-tokyo.ac.jp/JGWwiki/CLIO/Tasks/DigitalControl>

にある。

特に各種回路関連:

[7] <http://gw.icrr.u-tokyo.ac.jp/JGWwiki/CLIO/Tasks/DigitalControl/AnalogFrontEnd>

<http://gw.icrr.u-tokyo.ac.jp/JGWwiki/CLIO/Tasks/DigitalControl/AnalogFrontEnd2>

回路図:

[8] <http://gw.icrr.u-tokyo.ac.jp/JGWwiki/CLIO/Tasks/DigitalControl/pban_files>

また、リアルタイムサーバについての細かい設定手順:

[9] <http://gw.icrr.u-tokyo.ac.jp/JGWwiki/CLIO/Tasks/DigitalControl/RTCSetupMemo>

各種測定データ:

[10] <http://gw.icrr.u-tokyo.ac.jp/JGWwiki/CLIO/Tasks/DigitalControl/PerformanceTest>

などがまとめてあるので、各サブグループのメンバーが実際にシステムを組むわけではないが、いざというときのトラブルシューティングの際などには参考になると思われる。

なお、デジタルに関わる場合、大きく分けて4つの段階があると考えている。

a. システム開発者(Digital subsystem subgroup member):

デジタルシステムについて深く理解し、システムを一から構築できる。最低限文書[2-5]を理解し、実際の機器の作業を通じで学ぶ必要がある。計算機にも長けていることが必要。

b. 要素技術開発においてデジタルシステムを組み込む人(各サブグループの開発者):

Matlab上のGUIを用いてリアルタイムコードを組む必要があり、コード生成時にシステムを停止、再起動などしたりできるが、システムの構築までは関わらない。文書[2]を理解する必要がある。

c. 干渉計や要素技術でデジタルシステムにユーザーとして関わる人:

リアルタイムコードを書く必要は無いが、MEDMと呼ばれるGUIや、各種スクリプトを用いて、実際の実験装置に携わる。このレベルの適した文書があまり存在しない。むしろ、実際に装置を触り、これまでに書かれた多数のスクリプト等を読むことが重要。

d. 解析等、デジタルシステムと直接は関係ないが、データ等に直接触れる人:

実際に実験装置に触ることはないが、データのフォーマット、解析ソフト等で生データを扱う。

　この文書で想定しているのは主にbの人である。配布されたデジタルシステムを、最初はデジタルグループのメンバー一緒に要素技術に組み込み、そこから各グループで独自に拡張していくことになるはずである。そして、LCGT本体にその要素技術が組み込まれる際は、それまでに培われたリアルタイムコード、GUIのスクリーン、スクリプト等がほぼそのまま、LCGT本体の大きなシステムの一部として使われていくことになる。