

マルチタスク OS を用いたイメージングリオメータ制御部の開発

菊池雅行¹・山岸久雄¹

Development of the imaging riometer control system on a multi-task operating system

Masayuki Kikuchi¹ and Hisao Yamagishi¹

Abstract: We have developed a controller for the imaging riometer running on a multi-task operating system. Beam scanning and data sampling of the system are controlled by the interrupt handler triggered by an external cyclic signal. We can monitor the condition of the system and obtain data from a remote station even while the system is busy for observation, because the system runs on a multi-task operating system. In this paper, we describe the performance of the system and some results of task delay measurement when the CPU is under a heavy load. The measured task delay for cyclic signals was within 30 μ s, which is negligible for data sampling period (12.5 ms) and beam scanning (125 ms) period. These results show that the system can control beam scanning and data sampling under a heavy CPU load.

要旨: 我々はマルチタスク OS 上で動作するイメージングリオメータコントローラを開発した。このコントローラは 125 ms の外部入力周期信号を基準として、割込ハンドラ内のソフトウェア処理によってビーム掃引、キャリブレーション、データサンプリングの計測動作を実行する。システムはマルチタスク OS の特性を生かし、計測と同時に TCP/IP 経由による動作モード制御、監視が可能である。また、冷却ファン・ディスクモーターを排した設計としたため、対障害性・低温動作特性に優れている。本論文ではこのデータコントローラの概要と、マルチタスク OS 使用時に問題となる、高負荷時に発生する動作遅延の評価結果について述べる。この計測の結果、我々のシステムにおける高負荷時の動作遅延は 30 μ s 以下であることが分かった。この値は基準周期信号 125 ms 及びサンプリング周期 12.5 ms に比較して十分小さい値であり、本システムはマルチタスク環境下においても他のタスクに影響されることなく、高精度なビーム制御が可能であることが確認された。

1. はじめに

銀河電波雑音吸収 (Cosmic Noise Absorption; CNA) を測定するリオメータ (Riometer; Relative Ionospheric Opacity Meter) 装置は、古くから電離層への降下粒子の観測手段として用いられてきた。1970 年代末、Nielsen は、それまで天頂付近一点の観測であった CNA 観測を、アレイアンテナによって作り出した 4 本のビームを持つリオメータ (マルチビームリオメータ) で行った。彼はオーロラ粒子に伴う数 10 keV の粒子の降下領域が、電離層高度にお

¹国立極地研究所。National Institute of Polar Research, Kaga 1-chome, Itabashi-ku, Tokyo 173-8515.

いて数 10 km 以下に局在化する場合があることをこの観測により示した (Nielsen, 1980). その後、山岸らは、ビームパターンを一次元方向に掃引する走査型リオメータを開発し、この装置によって 1985 年から昭和基地において観測を開始した (山岸・菊池, 1989). その後 Rosenberg らにより、アレイアンテナとバトラーマトリクス (Butler and Lowe, 1961) を採用したビーム掃引の二次元化が試みられた. Rosenberg らはこの装置をイメージングリオメータ (略称 IRIS) と名付けている (Detrick and Rosenburg, 1990). その後山岸らは Rosenberg らの IRIS (7×7 array) に比較して細かい空間分解能 (8×8 array) を持つ IRIS を開発し、地磁気共役点であるアイランド (1990 年) と昭和基地 (1992 年) に設置し観測を開始した (山岸ら, 1992; 佐藤ら, 1992). この観測は現在も継続している.

昭和基地及びアイランドで稼働している IRIS は非常に安定に稼働している. しかし現行システムは観測中にネットワーク経由でデータを配信することが出来ない. このため宇宙天気予報に代表される近年のデータ流通の即時化要求に対応できない. また使用しているコントロール PC が現在入手困難となりつつあるタイプであるため、予備品の供給が難しいという問題がある. 我々はこの問題を解決するために、現時点で一般的な PC-AT 互換機上で動作するイメージングリオメータ制御装置を開発した. この制御装置はマルチタスク・リアルタイム OS である RT-Linux (Barabanov and Yodaiken, 1996) 上で動作するため、ビーム掃引のような実時間動作に制限のある処理 (リアルタイム処理) とデータ表示・伝送のタスクを同時に独立なプロセスとして行うことが可能である. さらに筐体からモータを排した設計となっているため、機械故障の要因が無い. 本論文ではこの装置の概要と、本システムで採用したアルゴリズムの限界使用速度の評価を行った結果について述べる.

2. システム設計の方針と実装

部品供給の期間が限られる極地観測において、機器の耐久性は安定なデータを取得する上で本質的に重要である. 今回のコントローラ作成において、我々はハードディスク、電源ファン、CPU クーラーなどのモータ可動部を排除することを方針とした. これら可動部を持つ部品は、過去の昭和基地の観測においてしばしば故障の原因となっている. 本システムでは工業機器の内部に組み込んで使用するファンレスタイプの筐体 (Contec 社製 IPC-BX/M600(PCW)CP200) と、フラッシュロムを使用したシリコンディスク (Contec 社製 PC-SDD 1000H) の組み合わせを使用する. ハードウェア構成を図 1 に、スキャンコントロール用パラレルポートピンアサインを表 1 に示す. またアナログデータ入力ピンアサインを表 2 に示す.

ビーム掃引及びデータサンプルのタイミングは、AD ボードの上に設けられたタイマーを用いて周期的に生成された信号を基準とする. この周期信号を割込信号に用い、割込ハンドラーの中で AD 変換のスタート、ビームステップ出力、AD ボードのリセットのハードウ

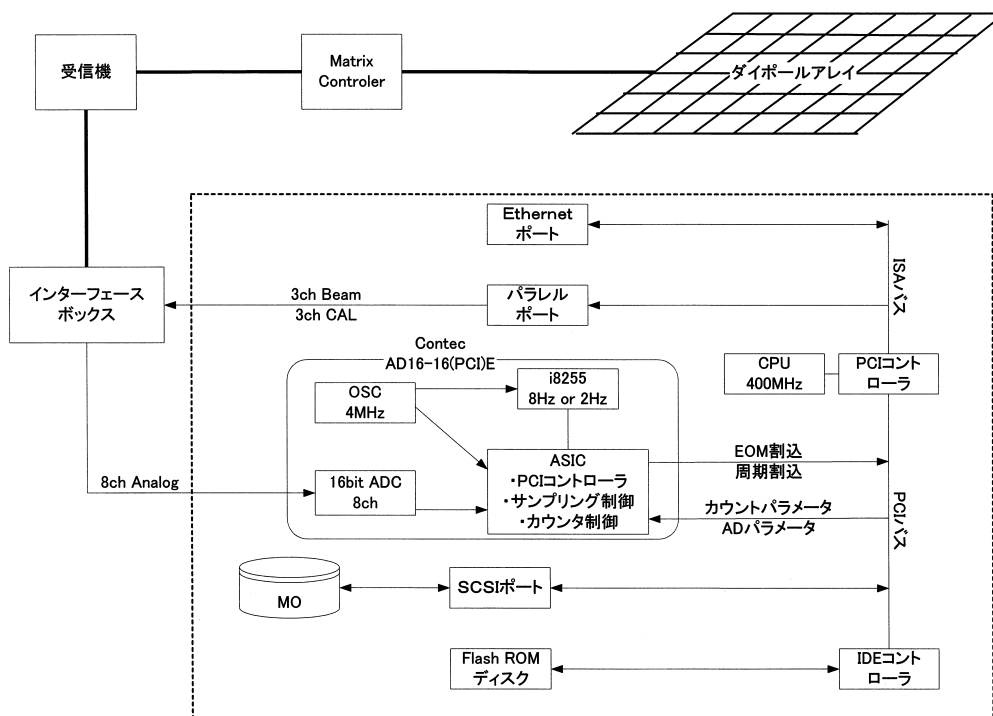


図1 ハードウェア構成図. 点線で囲んだ部分が今回開発した部分となる.

Fig. 1. Block diagram of hardware. We have developed the section in the dashed box.

表1 パラレルポートのピンアサイン

Table 1. Pin assignment of parallel output port.

1 CRS Bit 3 [IN]	7 Cal Bit 1
2 Scan Bit 0 [OUT]	8 Cal Bit 2
3 Scan Bit 1 [OUT]	9 Handler status (H: Hadler active) [OUT]
4 Scan Bit 2 [OUT]	10-17 N/A [IN]
5 Cal status (H: On L: Off) [OUT]	18-25 GND
6 Cal Bit 0	

表2 アナログ入力端子のピンアサイン

Table 2. Pin assignment of AD input port.

1 Analog Input 0+	11 Analog Input 5+
2 Analog Input 0-	12 Analog Input 5-
3 Analog Input 1+	13 Analog Input 6+
4 Analog Input 1-	14 Analog Input 6-
5 Analog Input 2+	15 Analog Input 7+
6 Analog Input 2-	16 Analog Input 7-
7 Analog Input 3+	17 N/A
8 Analog Input 3-	18 N/A
9 Analog Input 4+	19 +5V from PC
10 Analog Input 4-	20-36 Analog ground, 37 digital ground

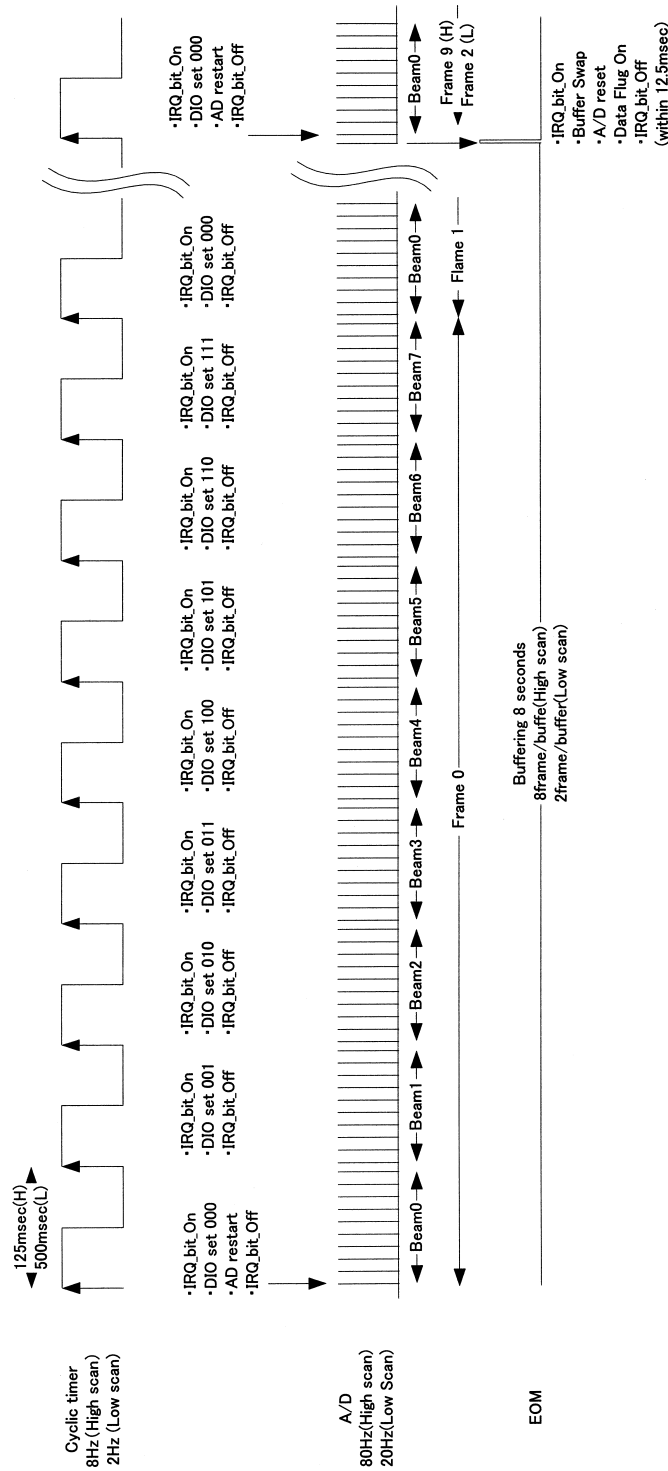


図 2 動作タイミング図. 125 ms, あるいは 0.5 ms 周期の外部クロックを元に, ビーム掃引とデータサンプリングを割込ハンドラー内で制御する.
 Fig. 2. Timing chart of the system. Beam scanning and data sampling of the system are controlled by the interrupt handler triggered by an external cyclic signal.

ア処理を行う。ハードウェア制御部はこのように外部信号を基準として動作するため、CPUが他のタスクに時間を割いている間でも周期的な動作が可能となる。我々のIRISは全点を1秒、または4秒で掃引する。データは各ビームに対して10点のサンプリングを行い（1秒掃引の場合、サンプリング周期12.5ms）、このうち中心部分の5点の平均値を各ビームに対する出力としている。ADボードの中にはトグルバッファがあり、それぞれ8秒間のデータをバッファリングする。バッファがフルの状態になった際にはADボードよりメッセージ終了割込（End of Message interrupt: EOM 割込）が発生する。この時、次のサンプリング開始までの12.5ms以内にハンドラー内部でデータバッファ切り替えを行う必要がある。これらの動作タイミングを図2に示す。

多くの観測において、データに対する時刻付けは十分な検討が必要となる点である。データの時刻間隔が今回の場合最低1秒であることから、本システムにおいて時刻はバッファフラッシュ時のEOM割込の生じた時刻を、OSの時刻情報をms単位まで使用してデータに付与する。より高精度な、1PPSに同期した動作を行うことも可能であるが今回そのような制御は行わない。OSの時刻は、ローカルエリアネットワーク上で時刻を配信するプロトコル（Network Time Protocol: NTP）を利用して校正する。この時刻自体は、ネットワーク上にあるGPSを用いた通常複数のタイムサーバによって管理されている。プログラム起動直後と正時には、8段階のキャリブレーションコントロール信号を出力する。現在キャリブレーション中であることと、そのキャリブレーションレベルはデータ中のステータス領域に記録される。

今回使用したOSであるRT-Linuxは、Linus Torvaldsが1991年に開発したUnix系OSであるLinuxに、Victor Yodaikenがリアルタイムスケジューラ機能を付加したOSである（Barabanov and Yodaiken, 1996）。Linuxのカーネルソースにパッチを当て、タスクスケジューラの一部を書き換えることで通常のLinuxには無いリアルタイムAPI（Application Program Interface）が使えるようになっている。拡張APIの中には、「割り込みハンドラの登録、削除」「リアルタイムスレッドの作成、削除」「ユーザプロセスとの通信に用いるFIFO（first-in-first-out）論理デバイスの作成、削除」が含まれる。本システムではマルチタスクOSの特徴を生かし、データ取得部と表示、通信制御部を分離している（図3）。データ取得部はデバイスドライバとデータ取得バックグラウンドプロセスからなり、このバックグラウンドプロセス（デーモンプロセス；Daemon process）は電源投入後自動起動される。デーモンプロセスは一時間ごとに観測データをWekHHMMSS.datという名前で作成する。このため一週間後には同じデータ名のデータが作成されることとなり、最大一週間分のデータのみが循環的に保存される。この仕組みでハードディスクのデータあふれを防止する。作成中のファイル名は、名前の固定されたファイル/home2/DATA/Active（アナウンスファイル）の中に書かれており、他のプロセスはこのファイルにアクセスすることで現在作成されているファ

イルを知ることが出来る。表 3 にデータ取得デーモンの作成するデータ形式を示す。データはすべてリトルエンディアン (Intel 形式) である。AD データの入力レンジとデジタル値は -10.000 V [0000]~ 9.99970 V [FFFF] に対応し、電圧換算式は

$$V_{\text{oltage}} (V) = \frac{\text{Digit} \times 20}{65536} - 10$$

である。作成されたデータは、TCP/IP (Transmission Control Protocol/Internet Protocol) あるいは UUCP (Unix to Unix Copy) などの適切なプロトコルを用いて国内に転送する。通常は一日に一度サマリーファイルを作成・送付するように運用を行う。

本システムに付属するクイックルック (QL) システムは、原則として現地のオペレーターが機器の正常な動作を確認するためのものである。QL プロセスは、アナウンスファイルに書かれた現在作成中のファイルをオープンして図形表示を行う。このように収録部と表示部をファイルで分離しているため、QL タスクを制御 PC と同じ筐体、あるいは NFS (Network File System) などにより別の PC からネットワーク経由で動作する、のどちらでも実装することが出来る。表示部は使用者の好み反映される部分であるが、このようにファイル構造でインターフェースを切っておくことで、QL プログラムをハードウェア制御の開発とは別のチームでの開発することが容易となる。

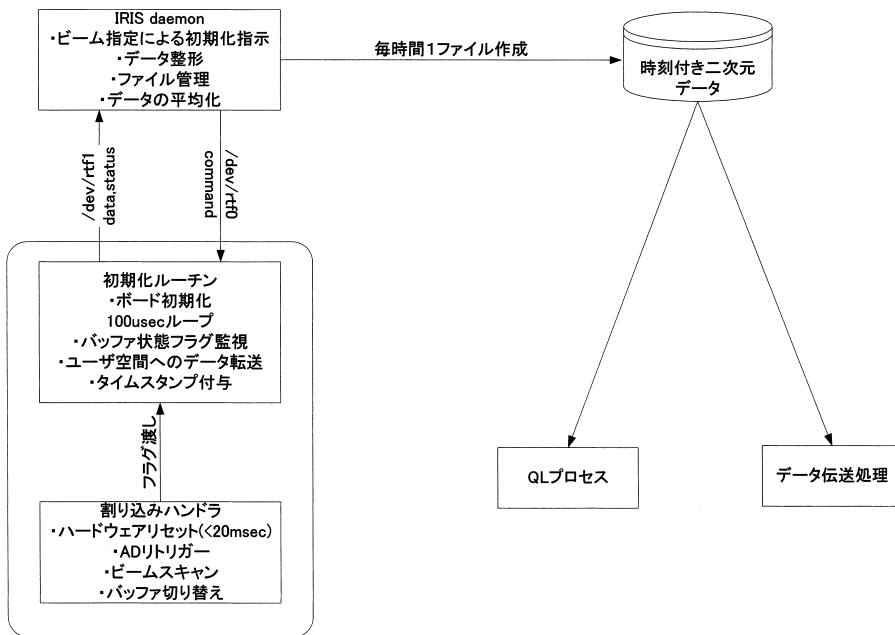


図 3 ソフトウェア構成図。データ取得、表示、伝送をそれぞれ別プロセスとして分離している。
Fig. 3. Block diagram of software. A set of control processes, a data display process and communication processes are designed independently on the multi-task operating system.

表3 ファイルデータフォーマット. このファイルは8秒ごとに追記更新される. クイックルックプロセスはファイルのデータを表示する.

Table 3. Data format on the file system. This file is renewed every 8 s adding new data and the QL process reads the data of the file.

UT0	UT1	UT2	UT3	MS0	MS1	CAL1	CAL2
Pad0	Pad1	Pad2	Pad3	Pad4	Pad5	Pad6	Pad7
CH0#0	CH1#0	CH2#0	CH3#0	CH4#0	CH5#0	CH6#0	CH7#0
CH0#1	CH1#1	CH2#1	CH3#1	CH4#1	CH5#1	CH6#1	CH7#1
CH0#2	CH1#2	CH2#2	CH3#2	CH4#2	CH5#2	CH6#2	CH7#2
CH0#3	CH1#3	CH2#3	CH3#3	CH4#3	CH5#3	CH6#3	CH7#3
CH0#4	CH1#4	CH2#4	CH3#4	CH4#4	CH5#4	CH6#4	CH7#4
CH0#5	CH1#5	CH2#5	CH3#5	CH4#5	CH5#5	CH6#5	CH7#5
CH0#6	CH1#6	CH2#6	CH3#6	CH4#6	CH5#6	CH6#6	CH7#6
CH0#7	CH1#7	CH2#7	CH3#7	CH4#7	CH5#7	CH6#7	CH7#7
UT1	UT2	UT2	UT4	MS1	MS2	CAL1	CAL2
...

UT0-UT3	観測ユニックス時刻 (4 byte, unsigned long)
MS0-MS1	観測ミリ秒 (2 byte, unsigned short)
CAL1, 2	校正ステータス (2 byte, unsigned short 0: cal off, 1: cal on)
Pad0-Pad7	Padding (reserved)
CHn#m	AD 入力チャンネル n の m 番目のデータ

3. ビーム掃引クロックに対する動作遅延量の評価

本システムは、ビーム掃引と AD ボードのバッファリング制御を、外部周期割込とデータバッファからの EOM 割込によって制御している。マルチタスク OS の場合、ディスクへのアクセスあるいは無限ループ処理などの負荷の高い処理を行っている最中に、しばしば実時間に制約のある処理（リアルタイム処理）が中断してしてしまう場合がある。たとえば本システムの場合、EOM 割込処理に対処する時間がサンプリング周波数 12.5 ms を越えると次の周期のサンプリングを開始することが出来なくなるためデータの取りこぼしが生じる。また、この遅延が周期割り込みのタイミングに比較して無視できない量の場合、外部クロックによって指定される 125 ms の周期に対して、実際のビーム掃引はジッタを持つことになる。これらの理由から、外部割込信号に対して実際の動作が実行されるまでの遅延時間の評価は、周期的動作を持つマルチタスクシステムを構築する際には、重要なポイントとなる。今回、開発環境での動作遅延時間の評価を以下の方法で行った。なお、今回測定に使用した開発環境では、データの記録にシリコンディスクではなく通常の回転式ハードディスクを用いている（図 4）。

- 1) 周期割り込み信号を取り出し、オシロスコープのトリガとして使用する。

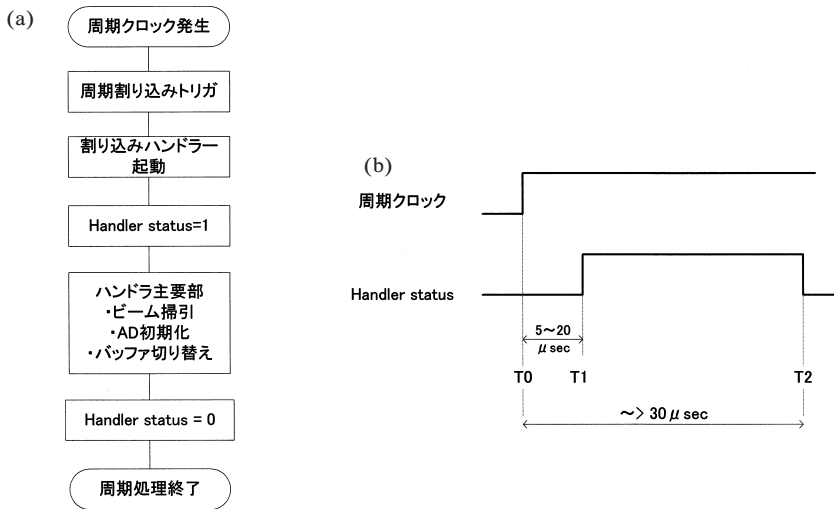


図 4 遅延測定方法 (a) と観測波形の模式図 (b)。周期信号の開始基準時刻を T_0 、割込ハンドラーが動作開始する時刻が T_1 、ハンドラーの終了時刻が T_2 となる。CPU に負荷を掛けた状態であっても、 $T_2 - T_0$ は $30 \mu\text{s}$ 以下であった。

Fig. 4. Measurement method of the interrupt latency (a) and a schematic waveform from the interrupt handler (b). T_0 is the starting point of the external cyclic signal. T_1 is the starting point of the interrupt handler. T_2 is the end of the interrupt handler. $T_2 - T_0$ was within $30 \mu\text{s}$ even if the CPU was under a heavy load.

- 2) 周期割り込み処理ルーチンの冒頭で、パラレルポートの特定ビットを 1 にする。
- 3) 処理ルーチンの終了時に、そのビットを 0 にする。

これらを行うことで割り込み信号に対する処理ルーチンの開始までの遅延時間 T_1 と、実際の処理が終了するまでの時間 T_2 を可視化することが出来る。 $T_0 \sim T_1$ は割り込み発生時での CPU への負荷と OS 固有のハンドラー起動の実装方法、及び CPU クロックに大きく依存する。また、 $T_1 \sim T_2$ は割り込みハンドラー内のステップ数と CPU クロックに大きく依存する。

無限ループ計算処理とディスク書き込みの 2 種類の負荷をかけた場合に、それぞれ上記遅延量の計測を行った。この結果、どちらの場合でも $T_0 \sim T_1$ は $5 \mu\text{s} \sim 20 \mu\text{s}$ の範囲、また $T_0 \sim T_2$ も $30 \mu\text{s}$ 以内であった。このことから CPU に過大な負荷が生じている場合でも、今回の制御方式の元での動作遅延は、ビーム周期 125ms やサンプリング周波数 12.5ms に比較して全く影響が無いレベルであることが分かる。

4. より安定なシステムのために

4.1. ハードウェア

観測装置の制御部の設計において、家電、工業製品、あるいは人工衛星などで用いられているいわゆる組み込み機器の設計方針は非常に参考となる。ハードウェア的に見た組込機器

の特長として、実装部品点数が少なく一般に低消費電力である点が挙げられる。また組込用低速 CPU は消費電力が少ないため、自己発熱や電源の負担が小さい。これは CPU と電源の冷却装置を省略することが容易になることを意味し、故障率の低下に直結する。この低消費電力化を進めることでバッテリー駆動が可能なシステムに到達する。バッテリー駆動が可能なシステムは、電源供給を前提としない無人観測装置への応用が可能となるのみならず、停電の起こることを前提とする昭和基地のような設置環境においても大変有利となる。

しかし観測機器制御は、制御すべき対象が多岐にわたり、一般に市販されている汎用組込 CPU ボード単体には無い AD 変換機能や CCD (Charge Coupled Diode) 駆動回路などの特殊な周辺回路が必要となることが多い。このため観測機器に対して組込用 CPU を単純には流用出来ないという事情がある。特殊な周辺回路は PC-AT 互換機に使用されている PCI バス (Peripheral Components Interconnect bus) あるいは ISA バス (Industrial Standard Architecture bus) で動作するものは多数見受けられる。今回我々が PC-AT 互換機をベースとしてシステムを構築したのも、十分な量のバッファが搭載された 16 bit AD コンバータを選択する必要があったためである。

ところが 1990 年代後半からプログラマブルゲートアレイ上に焼き込んで使用する、ハードウェア記述言語 (HDL) で記述された CPU、及びその上で動作するリアルタイム OS が登場するようになった。現時点での HDL で記述された典型的な 16 bit CPU の動作速度は 43 MHz-75 MHz となっており、ms オーダーの制御が必要とされる CPU としては十分な速度に達している。このような HDL で記述された CPU と、同様に HDL で記述した観測器用周辺回路を一つのプログラマブルゲートアレイに実装することで、組込 CPU に専用周辺機器を搭載したカスタム CPU が実験室レベルで開発可能となってきた。このことで周辺機器のバス依存性の問題が解消することとなり、我々は PCI あるいは ISA バスを持つ PC-AT 互換機以外の筐体を、より柔軟に選択することが出来るようになる。PC-AT 互換機が将来入手不困難となっても HDL (hardware description language) で記述したシステムであれば独自の開発環境がソフトウェア資産として維持できるであろう。今後 HDL で記述された、周辺機器を含んだプログラマブルゲートアレイ上の CPU は、我々が必要とするより信頼性の高い安定した制御・収録システムを柔軟に実現するための有力な手段となる。ゲートアレイを用いたシステムの試作は、実験室レベルで使用できる基盤加工装置が入手可能な環境であれば比較的容易である。開発に際しては、研究室で十分に評価を行い、温度膨張・電磁干渉を考慮した実装委託を行う方法が望ましい。

4.2. ソフトウェア

ソフトウェア的な観点から見た組込システムの特長は、メモリサイズの小さな OS・機能を限定したプログラムが選択される点である。これは、組込機器はコスト・サイズ両面から

メモリが厳しく制限されることに起因する。しかしプログラムサイズに制限があるということで、盛り込むことの出来る機能を必要最低限にする必要に迫られ、検証すべき項目が減少する。こうした汎用コンピュータとは異なる設計思想の、動作モードを限定したコンパクトなプログラムは、検証がより容易になるため品質が向上し、結果的に観測データの安定取得に繋がる。

今回我々が Linux 系の OS を利用した理由は、開発環境が充実していること、無料であること、QL を作成するための GUI (graphics user interface) 環境が整っていること、開発人口が多いため外注が容易であること、である。このような開発環境では、クイックルックのようなディスク上のデータを処理する部分は外注し、デバイスをコントロールする部分は研究室レベルで機器に応じて開発・検証を行うという柔軟な開発スタイルをとることが出来る。

しかし、今回比較的サイズの小さな OS とソフトウェアの組み合わせを使用したにも関わらず、インストーラの指示に従った必要ディスク容量は約 300 Mbyte であった。本システムで最もコストが必要な部品は、容量の大きなフラッシュメモリディスクであり、コードサイズ削減はコスト削減に直結する。観測器のコスト削減は、単に支出を減らすという意味だけではなく、国内における訓練・障害検証用バックアップシステムの準備、あるいは十分な予備品の調達を可能とするという意味がある。この二つは極地観測における安定運用のためには本質的に重要である。必要な機能を過不足無く備え、ディスクスペースを小さくするような Linux のパッケージを作成するためには共有ライブラリの取捨選択を含む多くの試行錯誤が必要となる。フラッシュROMディスクの価格対労力の比較になるが、今後組込用に特化されたコードサイズの小さな Linux ディストリビューション、あるいは他の組込用 OS の選択は十分検討すべき事項である。

5. 結 び

今回我々は、マルチタスク OS 上で動作するイメージングリオメータコントローラの開発を行った。このシステムはイメージングリオメータが計測中であっても TCP/IP などの OS 標準搭載の通信プロトコルを使うことが出来るため、観測装置の遠隔監視及びデータの即時配信が可能となった。マルチタスク OS 上でのリアルタイム性を確保するため、ビーム掃引基準信号をハードウェア的に生成し、これを元にした割込動作をリアルタイム処理の根本アルゴリズムとして実装した。この結果、負荷の高い処理が行われている最中であっても負荷によって発生する動作遅延は $30 \mu\text{s}$ 以下となることが分かった。これはビーム掃引周期 125 ms あるいはデータサンプリング周波数 12.5 ms に比較して全く問題が無いレベルである。RT-Linux と本論で使用した実装方法の組み合わせによれば、クロック 166 MHz 級の CPU であっても、数 $100 \mu\text{s}$ 周期のハードウェア制御は十分可能であることが分かった。この方式はイメージングリオメータのみならず、この時間スケールを持つ計測一般に応用が可能である。

文 献

- Barabanov, M. and Yodaiken, V. (1996): Realtime Linux, *Linux Journal*, March.
- Butler, J. and Lowe, R. (1961): Beam-forming matrix simplifies design of electrically scanned antennas. *Electron. Des.*, **12**, 170–173.
- Detrick, D. L. and Rosenberg, T. J. (1990): A phased-array radiowave imager for studies of cosmic noise absorption. *Radio Sci.*, **25**, 325–338.
- Nielsen, E. (1980): Dynamics and spatial scale of auroral absorption spikes associated with the substorm expansion phase. *J. Geophys. Res.*, **85**, 2092–2098.
- 山岸久雄・菊池 崇 (1989): 掃天型リオメータの開発. *南極資料*, **33**, 17–32.
- 山岸久雄・西野正徳・佐藤 貢・加藤泰男・小島正宣・佐藤夏雄・菊池 崇 (1992): イメージングリオメータの開発. *南極資料*, **36**, 227–250.
- 佐藤 貢・山岸久雄・加藤泰男・西野正徳 (1992): イメージングリオメータ吸収画像 QL システムの開発. *南極資料*, **33**, 251–267.

(2002年6月10日受付; 2002年10月25日改訂稿受理)