

Real-Time Clock and Interrupt Module (RCIM) User's Guide



0898007-1240
Jan 2025

Copyright 2025 by Concurrent Real-Time. All rights reserved.

本書は当社製品を利用する社員、顧客、エンドユーザーを対象とします。
本書に含まれる情報は、本書発行時点での正確な情報ですが、予告なく変更されることがあります。
当社は、明示的、暗示的に関わらず本書に含まれる情報に対して保障できかねます。

誤字・誤記の報告または本書の特定部分への意見は、当該ページをコピーし、コピーに修正またはコメントを記述してコンカレント日本株式会社まで郵送またはメールしてください。

<http://www.concurrent-rt.co.jp/company/>

本書はいかなる理由があろうとも当社の許可なく複製・変更することはできません。

Concurrent Real-TimeおよびそのロゴはConcurrent Real-Timeの登録商標です。
当社のその他すべての製品名はConcurrent Real-Timeの商標であり、同時にその他全ての製品名が各々の所有者の商標または登録商標です。
Linux®は、Linux Mark Institute(LMI)のサブライセンスに従い使用しています。

改定履歴:

Revision History:	Level:	Effective With:
Original Release – August 2002	000	RedHawk Linux Release 1.1
Previous Release -- December 2003	210	RedHawk Linux Release 2.0
Current Release -- May 2005	300	RedHawk Linux Release 2.3
Update -- September 2005	310	RedHawk Linux Release 2.3-4.1
Update -- May 2007	320	RedHawk Linux Release 4.2
Update -- April 2008	330	RedHawk Linux Release 5.1
Update -- June 2008	400	RedHawk Linux Release 5.1
Update -- October 2010	500	RedHawk Linux Release 5.4
Update -- December 2011	600	RedHawk Linux Release 6.0
Update -- February 2013	610	RedHawk Linux Release 6.3
Update -- February 2014	620	RedHawk Linux Release 6.3
Update -- March 2016	700	RedHawk Linux Release 7.2
Update – June 2016	800	RedHawk Linux Release 7.2
Update – June 2017	900	RedHawk Linux Release 7.3
Update – March 2021	1000	RedHawk Linux Release 8.2
Update – June 2021	1100	RedHawk Linux Release 8.2
Update – August 2021	1200	RedHawk Linux Release 8.2
Update – April 2022	1210	RedHawk Linux Release 8.4
Update – June 2023	1220	RedHawk Linux Release 9.2
Update – May 2024	1230	RedHawk Linux Release 9.2

マニュアルの範囲

本マニュアルはConcurrent Real-TimeのRedHawk™ Linux®オペレーティング・システムがインストールされたiHawk™システム上のReal-Time Clock and Interrupt Module (RCIM)を導入および利用するユーザーを対象としています。

NOTE

本ガイドでは2つのRCIMモデル(RCIM IIIとRCIM IV)について説明します。用語「RCIM」の使用は両方のボードに共通する機能性について言及しています。“RCIM III”と“RCIM IV”は特定のボードを指します。各ボードの仕様については1-3ページの「仕様」を参照してください。

NOTE

RCIM IIIとRCIM IVの双方ともにPCI Expressボードです。古いRCIM I(PCI)ボードとRCIM II(PCI-X)ボードを扱う情報については、次の場所にあるConcurrentのDocumentation Libraryに含まれる過去のRCIM User's Guideを参照して下さい：
<https://redhawk.concurrent-rt.com/docs/>

マニュアルの構成

本ガイドは次から構成されます：

- 1章「はじめに」は、RCIMボードに関する概要および仕様を含みます。
- 2章「ハードウェア、取り付け、構成」は、RCIMボードおよびコネクタの詳細な解説、導入および設定の取り扱い説明を提供します。
- 3章「機能説明」は、RCIMで利用可能なクロックと割り込みについての一般的な操作、ユーザー・インターフェース、構成オプションを提供します。
- 4章「GPSクロックの同期」は、オプションのGPSモジュールを取り上げます。
- 5章「IRIGタイムコードの同期」は、オプションのIRIGモジュールの利用を取り上げます。
- 付録A「RCIM IVレジスタ」は、RCIM IVのレジスタについて説明します。
- 付録B「RCIM IIIレジスタ」は、RCIM IIIのレジスタについて説明します。

- 付録C「RCIMケーブル伝送遅延の計算」は、RCIMをチェーン接続した時の伝搬遅延を防ぐための計算式を提供します。

構文記法

本書を通して使用される表記法は次のとおりとなります：

斜体	ユーザーが特定する書類、参照カード、参照項目は、 <i>斜体</i> にて表記します。特殊用語も <i>斜体</i> 形式にて表記します。
太字	ユーザー入力は 太字 形式にて表記され、指示されたとおりに入力する必要があります。ディレクトリ名、ファイル名、コマンド、オプション、manページの引用も 太字 形式にて表記します。
list	プロンプト、メッセージ、ファイルやプログラムのリストのようなオペレーティング・システムおよびプログラムの出力は list 形式にて表記します。
[]	ブラケット(大括弧)はコマンドオプションやオプションの引数を囲みます。もし、これらのオプションまたは引数を入力する場合、ブラケットをタイプする必要はありません。
ハイパーテキスト・リンク	本資料を見ている時に項、図、テーブル・ページ番号照会をクリックすると対応する本文を表示します。 青字 で提供されるインターネットURLをクリックするとWebブラウザを起動してそのWebサイトを表示します。 赤字 の文書名称および番号をクリックすると(アクセス可能であれば)対応するPDFのマニュアルを表示します。

関連図書

Title	Pub No.
RedHawk Linux Release Notes Version x.x	0898003
RedHawk Linux User's Guide	0898004
RedHawk Linux Frequency-Based Scheduler (FBS) User's Guide	0898005

x.x = リリース・バージョン

目次

前書き	iii
1章 はじめに	
概要	1-1
仕様	1-3
2章 ハードウェア、取り付け、構成	
ボードの説明仕様	2-1
RCIM IV仕様	2-2
ボード説明図	2-2
コネクタとLED	2-3
LEDの機能	2-3
入出力ケーブルとコネクタ	2-4
発振器	2-5
GPSアンテナ	2-5
外部割込みI/Oコネクタ	2-5
システムの確認	2-7
デイジー・チェーン・ケーブル	2-7
RCIM III	2-8
ボード説明図	2-8
コネクタとLED	2-9
LEDの機能	2-9
入出力ケーブルとコネクタ	2-10
発振器	2-10
GPSアンテナ	2-11
外部割込みI/Oコネクタ	2-11
システムの確認	2-13
デイジー・チェーン・ケーブル	2-13
接続モード	2-14
RCIMの開梱	2-15
取り付け	2-15
構成	2-16
カーネル構成	2-16
ドライバ構成	2-16
一般論	2-17
MSI割込み構成	2-18
3章 機能説明	
概要	3-1
クロック	3-1
ティック・クロック	3-2
POSIXクロック	3-2
クロックへの直接アクセス	3-3
クロックの同期	3-3

rcim_clocksyncユーティリティ	3-3
ティック・クロックの同期	3-5
RCIMマスタークロックの留意事項	3-5
POSIXクロックの同期	3-6
自動同期	3-6
システム時刻管理のためのGPSの利用	3-6
システム時刻管理のためのIRIGの利用	3-7
割込み処理	3-8
割込み処理ロジック	3-8
DIとETIの実装と有効化	3-9
割込み認識ロジック	3-9
分配割込みの設定	3-10
RCIM値の取得	3-11
エッジ・トリガ割込み	3-12
入力構成	3-12
ETIデバイス・ファイル	3-13
ETIユーザー・インターフェース	3-13
ETIの配信	3-14
リアルタイム・クロック(RTC)	3-14
RTCデバイス・ファイル	3-14
RTCの配信	3-14
RTCユーザー・インターフェース	3-15
外部出力割込み	3-15
出力ソース構成	3-16
MSI-X割込み構成	3-17
MSI-Xソース構成	3-17
MSI-Xの互換性	3-18
プログラマブル割込み発生器(PIG)	3-19
PIGデバイス・ファイル	3-19
PIGの配信	3-19
分配割込み	3-20
DI構成	3-20
DIデバイス・ファイル	3-21
DIユーザー・インターフェース	3-21
マスタークロックの無効化	3-22
外部クロックの入力	3-22
RCIM IVのピン構成	3-23

4章 GPSクロックの同期

概要	4-1
GPSDとChronyd	4-1
gpsdの構成	4-2
chronydの構成	4-3
PPS時刻オフセットの計算	4-5
chronycを使ったGPS機能の確認	4-7
xgpsを使ったGPS衛星の表示	4-8
NTPD	4-9
ntpdの構成	4-9
ntpqを使ったGPS機能の確認	4-10
GPS同期の精度	4-11

5章 IRIGタイムコードの同期

概要	5-1
IRIGマスター	5-1
出力信号	5-1
AM信号	5-2
DCLS信号	5-2
伝搬遅延	5-3
IRIGスレーブ	5-3
構成	5-4
入力信号	5-5
AM信号	5-5
DCLS信号	5-5
同期の精度	5-6
IRIGプログラミング・インターフェース	5-8
IRIGマスターioctl	5-8
IRIG_OUTPUT_ENABLE	5-8
IRIG_OUTPUT_DISABLE	5-9
IRIG_OUTPUT_STATUS	5-9
IRIG_OUTPUT_RESET	5-9
IRIG_OUTPUT_GET_CONTROL_BITS	5-9
IRIG_OUTPUT_SET_CONTROL_BITS	5-9
IRIG_GET_LEAP_SECOND	5-10
IRIG_SET_LEAP_SECOND	5-10
IRIGスレーブioctl	5-10
IRIG_INPUT_ENABLE	5-11
IRIG_INPUT_DISABLE	5-11
IRIG_INPUT_STATUS	5-11
IRIG_GET_TIME	5-11
IRIG_GET_NS64	5-12
IRIG_GET_TIME_RAW	5-12

付録A RCIM IVレジスタ

RCIM IVアドレス・マップ	A-1
RCIM IVレジスタ	A-5

付録B RCIM IIIレジスタ

RCIM IIIアドレス・マップ	B-1
RCIM IIIレジスタ	B-4

付録C RCIMケーブル伝搬遅延の計算

相互接続の詳細	C-1
---------------	-----

1 はじめに

本章はReal-Time Clock and Interrupt Module (RCIM)に関する概要と仕様を提供します。

NOTE

本ガイドでは2つのRCIMモデル(RCIM IIIとRCIM IV)について説明します。用語「RCIM」の使用は両方のボードに共通する機能性について言及しています。“RCIM III”と“RCIM IV”は特定のボードを指します。「仕様」項で各ボードの仕様について解説しています。

NOTE

RCIM IIIとRCIM IVの双方ともにPCI Expressボードです。古いRCIM I(PCI)ボードとRCIM II(PCI-X)ボードを扱う情報については、次の場所にあるConcurrentのDocumentation Libraryに含まれる過去のRCIM User's Guideを参照して下さい：
<https://redhawk.concurrent-rt.com/docs/>

概要

Real-Time Clock and Interrupt Module (RCIM)は、外部イベント、同期クロック、同期割り込みへの急速な応答を必要とするタイム・クリティカル・アプリケーションをサポートするPCIベースのカードです。

個々のシステムのRCIMボードがチェーン接続されている場合、割り込みを接続された全てのRCIMに対して、そしてRCIMから対応する全てのホスト・システムへ一斉に分配することが可能です。

複数のシステムのRCIMチェーン内の全てのRCIMが共通の時間基準を共有できるように同期された高分解能クロックが提供されます。これはローカルPOSIX 1003.1に準拠する高分解能クロックもまた提供します。オプションのGPSモジュールはGPS標準時間へのクロック調整を可能にします。GPSモジュールを搭載する複数のRCIMはケーブル接続無しで共通の時間基準を提供することが可能です。高安定発信器は標準です。オプションの発信器はRCIMで計測される時間の精度を向上させます。

クロックに加え、この多目的PCIベースのカードは次の機能を持っています：

- 外部デバイス割り込みの接続
- システムに割り込むことが可能なリアルタイム・クロック・タイマー

- アプリケーション・プログラムから割り込みの生成を可能にするプログラマブル割り込み発生器

これらの機能はRCIMカードがインストールされたシステム上で全てのローカル割り込みを生成する事が可能です。システムがチェーン接続されている時、複数の入力・出力割り込みはRCIMが接続された他のシステムへ分配する事が可能です。これは1つのタイマー、1つの外部割り込み、1つのアプリケーション・プログラムが同期した動作を引き起こすためにほぼ一斉に複数のRedHawk Linuxシステムに割り込む事を可能にします。

仕様

特徴	RCIM IV	RCIM III
クロック		
POSIX 長さ 分解能 発信器の安定性	64bit (2つの32bitワード) 上位32bit – 1sec 下位32bit – 400nsec +/-2.5 PPM	64bit (2つの32bitワード) 上位32bit – 1sec 下位32bit – 400nsec +/-2.5 PPM
Tick Timer 長さ 分解能	64bit (2つの32bitワード) 400ns/ティックの64bitカウンタ	64bit (2つの32bitワード) 400ns/ティックの64bitカウンタ
リアルタイム・クロック		
数 長さ 分解能 発信器の安定性	8個 32bit 1 microsecond (より大きな値をプログラム可能) ±2.5 PPM	8個 32bit 1 microsecond (より大きな値をプログラム可能) ±2.5 PPM
ローカル割込み		
外部入力割込み 外部出力割込み リアルタイム・クロック	共有入出力12個(3.5Vまたは5V TTL) 8個	12個 (5V TTL) 12個 (5V TTL) 8個
分配割り込み		
入力 出力	12個 12個	12個 12個
割込み応答時間		
ユーザー・プロセスへの 割込み	< 8 microsec	< 8 microsec
実装		
外形 最大ケーブル長 (算出方法については付録 Cを参照) 外部コネクタ PCI性能 オプション	PCIe 30メータ 26pin HD D-SUB x1 マルチGNSSタイミング・モジュール、 恒温槽付き発信器	PCIe 30メータ Molex LFH-60 x1 GPSモジュール、 恒温槽付き発信器
動作環境		
動作温度 保管温度 相対湿度	10～55℃ -40～70℃ 10～90% (非結露)	10～55℃ -40～70℃ 10～90% (非結露)
電力		
消費電力	最大20W	最大20W

ハードウェア、取り付け、構成

本章はPCIベースのボードRCIMの説明に加え、導入や構成情報を提供します。

ボードの説明

本項はRCIM IVおよびRCIM IIIボードの説明図や解説を提供します。

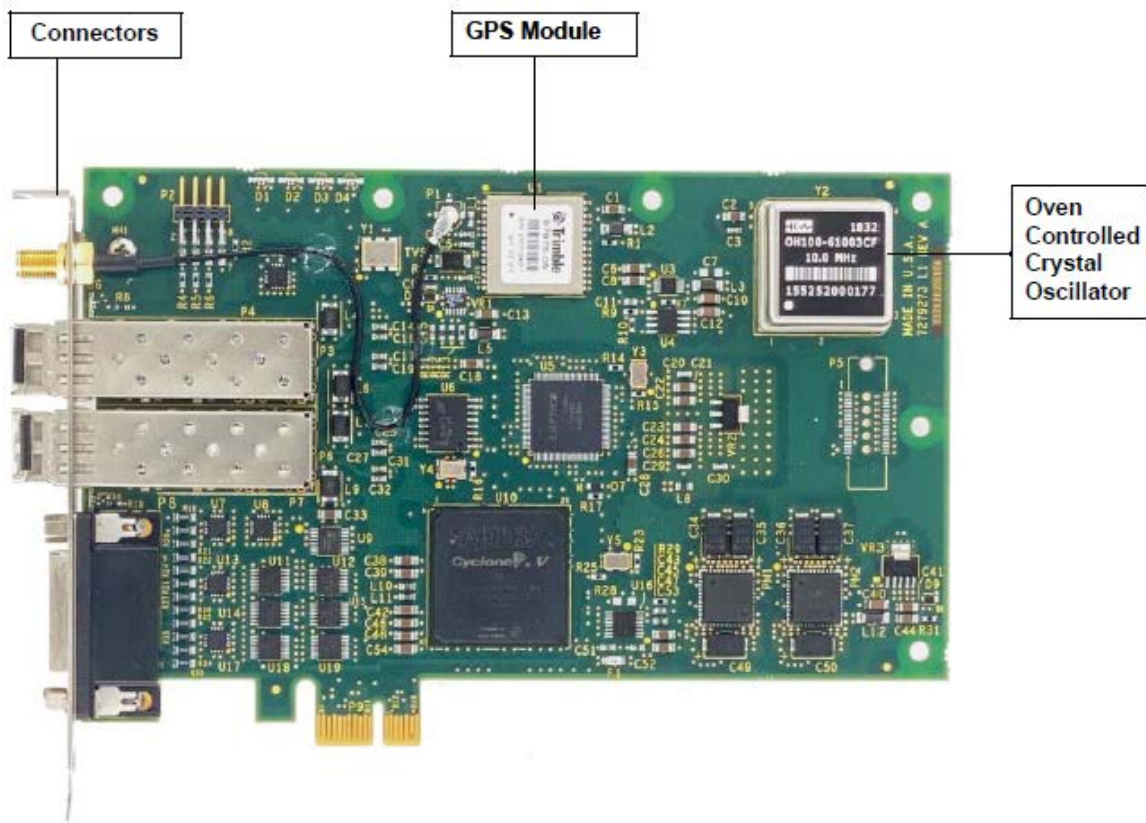
RCIMボードはホスト・システムの標準的なPCI Expressスロットに実装します。外部割込み接続のためにコネクタが各RCIMに実装され、同期ケーブルがマスターRCIMと1つ以上のスレーブRCIMをデージー・チェーン接続するために組み込まれます。

RCIM IV

ボード説明図

図2-1はオプションの高安定性OCXO(Oven Controlled Crystal Oscillator : 高温槽付水晶発信器)とGPSモジュールが組み込まれたRCIM IVボードを示します。

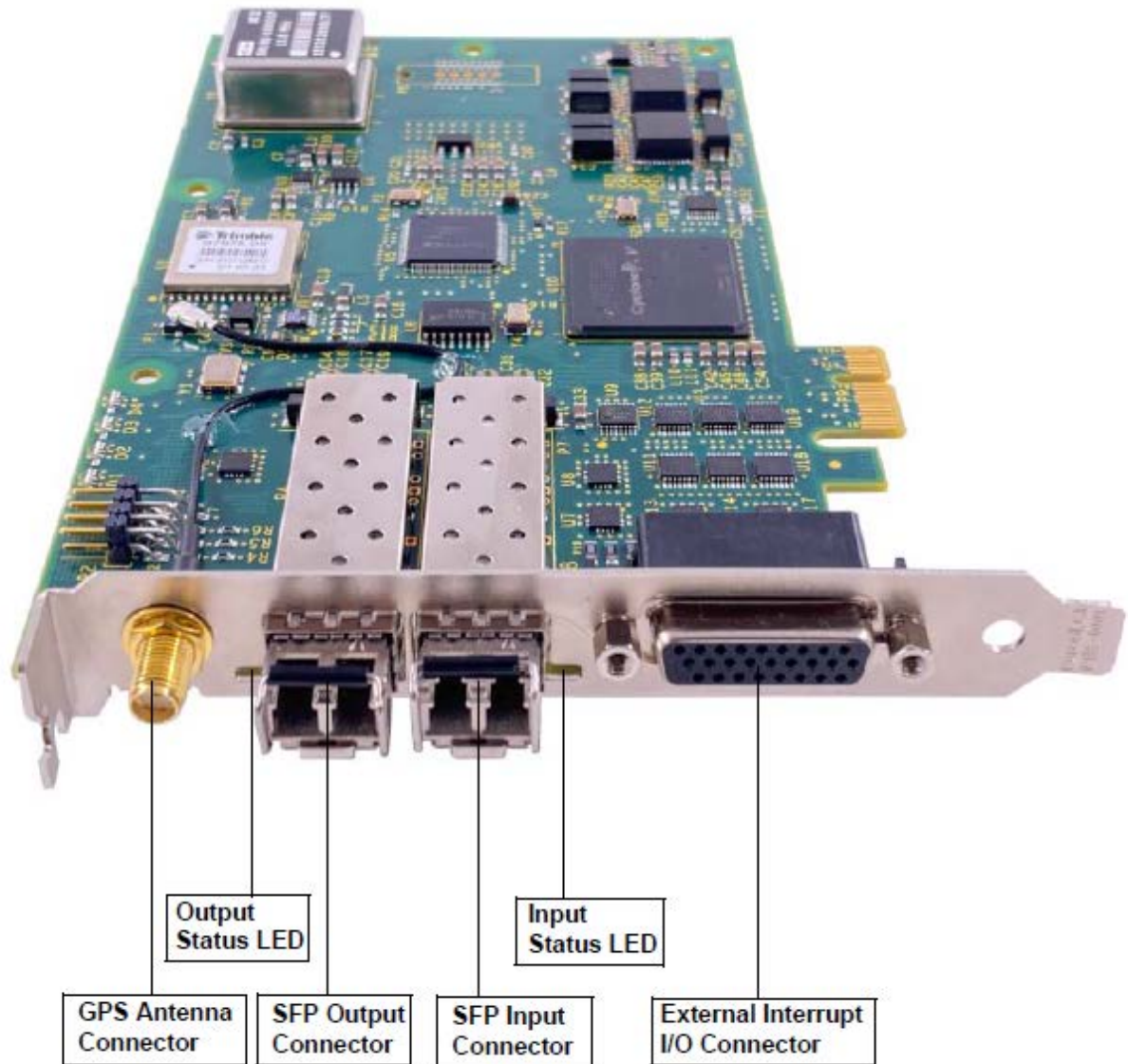
図2-1 RCIM IVボード



コネクタとLED

図2-2はRCIM IVボードの入出力コネクタとLEDを示します。LEDや各々のコネクタに関する詳細な情報は次の項で提供します。

図2-2 RCIM IVのコネクタとLEDの位置



LEDの機能

RCIM IVボードの入力および出力コネクタの近くに2つの2色のステータスLEDがあります。それらはボードがリセット・モードの時にテストとして赤と緑に短い間隔で明るく光り、その後どちらにも微かな赤に光ります。

ボードが正常に動作している間のLED機能は次のようになります：

LED	内容	機能
出力ステータスLED	赤固定	10MHzクロックの故障
	2秒毎に赤が点灯	ケーブル・オプション付きだがケーブルは同期していない、または失敗した状態でPOSIXクロックは停止中
	1秒毎に緑が点灯	ケーブル・オプション無しでPOSIXクロックは動作中
	1秒毎に緑が点滅	ケーブル・オプション付きでケーブルは同期した状態でPOSIXクロックは動作中
	2秒毎に赤と緑が交互に点灯	ケーブル・オプション付きだがケーブルは同期していない、または失敗した状態でPOSIXクロックは動作中
	1秒毎に青が点灯	ケーブル・オプション無しでPOSIXクロックが動作中、かつ適切なGPSパルスを受信中
	1秒毎に青が点滅	ケーブル・オプション付きでPOSIXクロックが動作中、かつ適切なGPSパルスを受信中、かつケーブルは同期した状態
	2秒毎に赤と青が交互に点灯	ケーブル・オプション付きでPOSIXクロックが動作中、かつ適切なGPSパルスを受信中だがケーブルは同期していない、または失敗した状態
入力ステータスLED	2秒毎に赤が点灯	ケーブル・オプション付きだがケーブルは同期していない、または失敗した状態
	緑固定	ケーブル・オプション付きでケーブルが同期した状態
	1秒毎に青が点灯	ケーブル・オプション無しだが適切なIRIGパルスを受信中
	1秒毎に青が点滅	ケーブル・オプション付きで適切なIRIGパルスを受信中、かつケーブルは同期した状態
	2秒毎に赤と青が交互に点灯	ケーブル・オプション付きで適切なIRIGパルスを受信中だがケーブルは同期していない、または失敗した状態

入出力ケーブルとコネクタ

RCIM IVは1組の標準的なSFP (small form-factor pluggable)コネクタをRCIM IVケーブルと接続するためのケージに差し込んで使用します。ケーブルはRCIM IVボード間で割り込み、タイム・スタンプ、基準クロックを伝達するために使用されます。出力ケーブル・コネクタは、RCIMがマスタまたはRCIMチェーン間のスレーブのどちらかの場合に使用されます(RCIMモードの説明については2-14ページを参照)。入力ケーブル・コネクタは、RCIMがスレーブ・モードまたはRCIMチェーン間のスレーブの場合に使用されます。ケーブルの部品番号(HS002-3CBL-xx : xxはメートル長)はLC光ファイバー・ケーブルとRCIMのマスタおよびスレーブの空のケージに差し込む2つのSFPを含みます。ケーブルに関する詳細については「デイジー・チェーン・ケーブル」項を参照してください。

NOTE

SFPのケーブルはRCIMを実装しているシステムの電源が落ちた状態で取り付けおよび取り外しを行ってください。ESD(静電気放電)の注意については「取り付け」項を参照してください。注意としてはSFPモジュールを所定の位置に固定すること、RCIM IVをSFPを取り付けている間にPCIeスロットから押し出されないことを確実に行うことが必要とされます。光ファイバー・ケーブルそのものはRCIM IVにダメージを与えることなくいつでも取り付けおよび取り外しする事が可能です。

発振器

RCIM IVに供給される温度補償水晶発信器(TCXO: Temperature Compensated Crystal Oscillator)は、±2.5PPM(100万分の1)の精度を持っています。

オプションの恒温槽付き水晶発信器(OCXO: Oven Controlled Crystal Oscillator)は、±10PPB(10億分の1)の温度安定性を提供します。

GPSアンテナ

RCIM IVのGPSオプションは有効なGPSアンテナと同軸ケーブルを含みます。

アンテナはGPS衛星信号を受信して受信機にそれを渡します。GPS信号は1575MHzレンジのスペクトラム拡散信号であり導電性または不透明な表面は貫通しません。従って、アンテナは空がきれいに一望できる屋外に設置する必要があります。

もし異なるアンテナまたはケーブルを使用する場合、以下の仕様に一致させる必要があります：

- 50Ωインピーダンス
- 27dBゲイン
- 3.3V DC電源、最大30mA.

外部割込みI/Oコネクタ

RCIM IVの外部割込みI/Oコネクタは、12個の共有出力と入力を提供する業界標準の26ピン高密度D-SUBです。

デフォルトでRCIM IVの外部入出力信号は**正論理(High True)**ですが、これは反転させることも可能です。詳細についてはA-7ページの「RCIM IV Interrupt Enable/Request/Pending/Clear/Arm/Level/Polarity Registers」を参照して下さい。

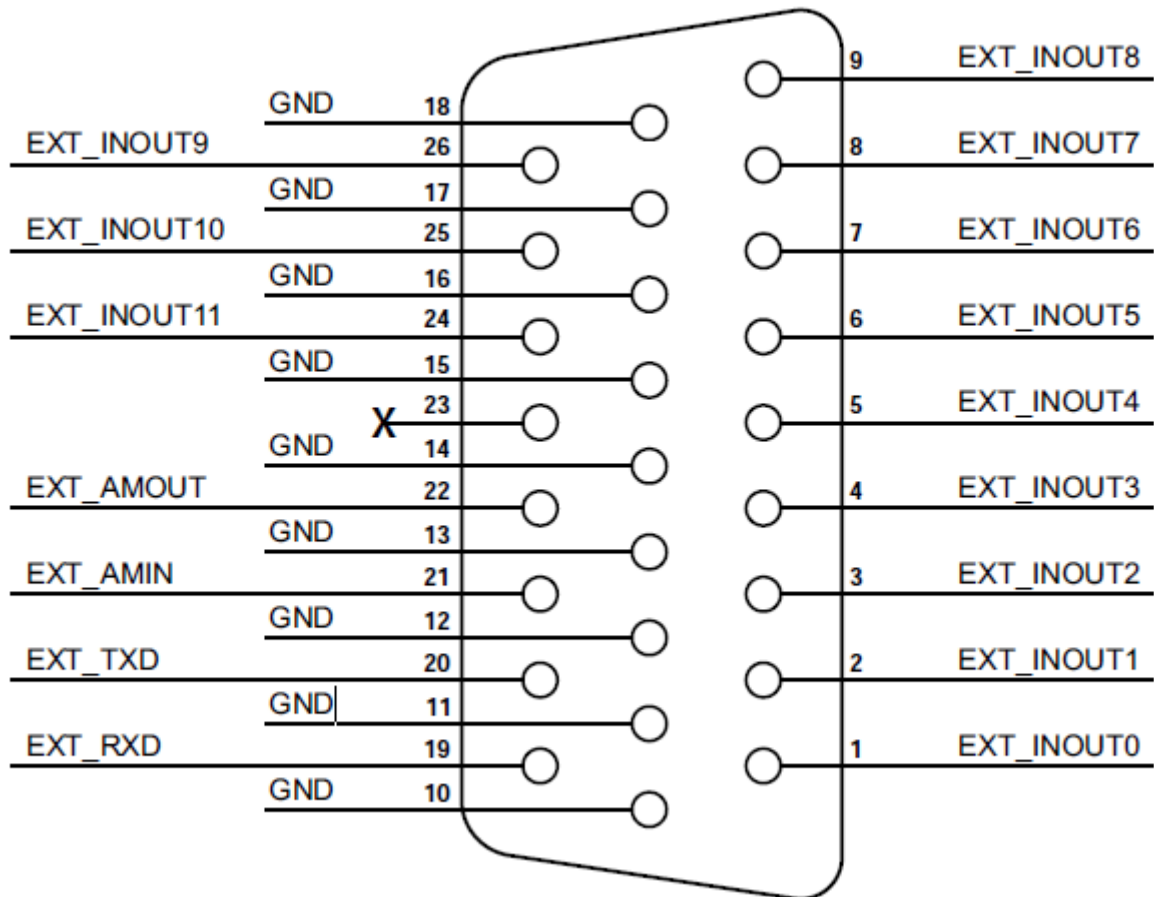
外部EXT_INOUTピンは、切り替え可能な100オーム終端を持つ入力または出力のいずれかをソフトウェアで選択可能なI/O割り込みを介して機器はRCIMと通信することが可能です。各ピンの機能は、入力または出力のいずれかに個々のピンを設定することが可能なマルチプレクサによって駆動されます。出力として設定されたピンは次のRCIMデバイスのいずれかにより駆動されます：プログラマブル割込み発生器(PIG)、リアルタイム・クロック・タイマー(RTC)、エッジ・トリガ割込み(入力ピンまたはETIとも呼ばれます)、分配割込み(DI)。各ピンは出力ゲートを無効にする機能を持つ入力と出力の両方になるように物理的に配線されています。外部配線がRCIMのピンに接続されていない場合、ピンはループバック接続用に入力と出力の両方の機能を同時に果たすことが可能です。

デフォルトで、ピンEXT_INOUT0からEXT_INOUT5は入力と出力の両方になるように構成され、ピンEXT_INOUT6からEXT_INOUT11は入力のみが構成されます。詳細については3-23ページの「RCIM IVのピン構成」を参照して下さい。

外部出力割込みとプログラム可能な割込みの利用に関する情報については3章を参照してください。

外部割込みI/Oコネクタのピン配列を図2-3に示します。

図2-3 RCIM IV外部割込みI/Oコネクタのピン配列



RCIM IVの外部割込み入力を選択可能な100オームの終端を持ち、3.3Vまたは5V TTLレベルの機能があります。推奨する入力信号持続時間は1 μ sであり、終端を無効にすることが可能ですのでライン・ドライバに対して最小電流量の必要条件はありません。

RCIM IVはマスタークロックを制御するために物理ピンに割り当てることができる2つのプログラム可能な入力ソース(DCLS入力と外部10MHzクロック)をサポートします。DCLS入力はデジタルIRIG入力を有効にし、外部10MHzクロック入力は他のクロックソースがRCIM上に存在していたとしても自動的にRCIMのマスタークロックを制御します。

信号EXT_RXDとEXT_TXDはRS-232レベルの信号です。これらは現在デバッグ目的で使用されています。

システムの確認

次の**lspci(8)**の出力は、RCIM IVに関するPCIクラス、ベンダーID、デバイスIDを示しています(1b:00.0 (bus:slot.function)はシステムで異なります)：

```
# lspci -v | grep -i rcim
1b:00.0 System peripheral: Concurrent Real-Time RCIM-IV Real-Time
Clock & Interrupt Module (PCIe) (rev 01)
    Kernel driver in use: rcim
    Kernel modules: rcim

# lspci -ns 1b:00.0
1b:00.0 0880: 1542:9273 (rev 01)
```

デイジー・チェーン・ケーブル

RCIM IVは、RCIMチェーンにRCIM IVを接続するために光ファイバー・シリアル同期ケーブルとSFP(small form-factor pluggable)コネクタ(P/N: HS002-3CBL-xx)を使用します。ケーブル上のシリアル・データはケーブルの問題を検出させる事が可能なパリティとフレミング情報を含みます。ポーリングが連続的に行われ、エラー状態が検出された時にRCIM IVのデイジー・チェーン・ケーブルのステータスを報告するメッセージが出力されます。問題を表しているメッセージは、失敗しているリンクが接続されているシステムに直接表示されます。

シリアル・ケーブルはポイント・トゥ・ポイント接続です。「Input」ケーブルは上流のマスタRCIMに向かっていているケーブルを指します。「Output」ケーブルはマスタから離れる下流の接続です。

```
RCIM: Input cable disconnected.
RCIM: Input cable connected.
RCIM: Input cable connected but not synchronized.
RCIM: Input cable unsynchronized.
RCIM: Input cable O.K.

RCIM: Output cable disconnected.
RCIM: Output cable connected.
RCIM: Output cable connected but not synchronized.
RCIM: Output cable unsynchronized.
RCIM: Output cable O.K.

RCIM: Cable error on input cable.
RCIM: Cable error on output cable.
```

「disconnected」と「connected」のメッセージはSFPがRCIM IVの適切なケージに差し込まれているかどうかに基づいてのみ発生します。

これらは光ケーブルが差し込まれているまたは外されている場合は発生しません。これらはSFPが正しく差し込まれていないまたは故障中の場合を除き、通常の操作中には発生しません。

「not synchronized」と「unsynchronized」のメッセージはケーブルが通信の試みに応答していない事を示します。これらのメッセージは光ケーブルが取り付けられているまたは外されている場合に発生します。これらは接続されたシステムの電源がOFFの場合にも発生します。

最後の2つのメッセージは、ケーブルのパリティ・エラーや一時的なケーブル同期の失敗のような一時的なエラーを示します。一時的なエラーが発生した場合、再同期するためにケーブルのリンクを必要とする可能性があります。分配割込みがケーブル上をブロードキャストしている場合、失敗する可能性があります。

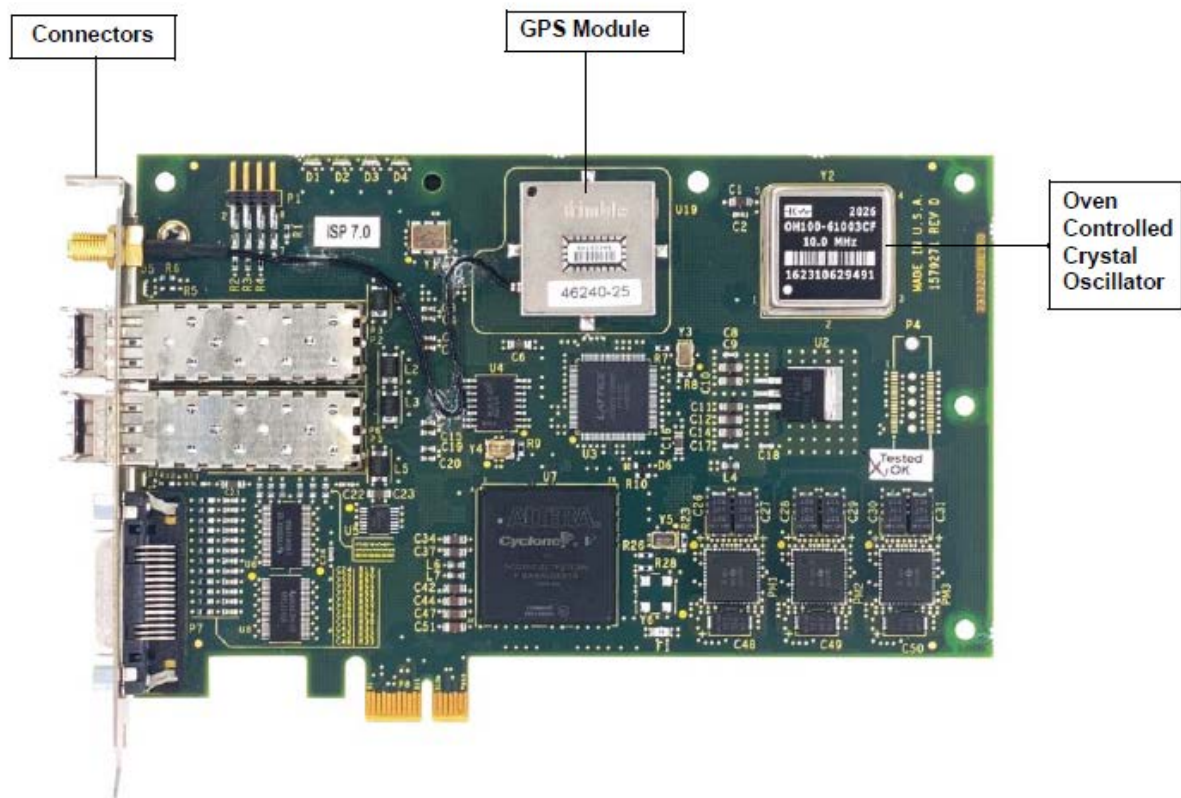
一時的なエラーはケーブル・クロックが全てのシステムに届かなくなるためティック・タイマーの同期にも影響します。クロック同期に関する説明については3章を参照して下さい。

RCIM III

ボード説明図

図2-4はオプションの高安定性OCXO(Oven Controlled Crystal Oscillator：高温槽付水晶発信器)とGPSモジュールが組み込まれたRCIM IIIボードを示します。

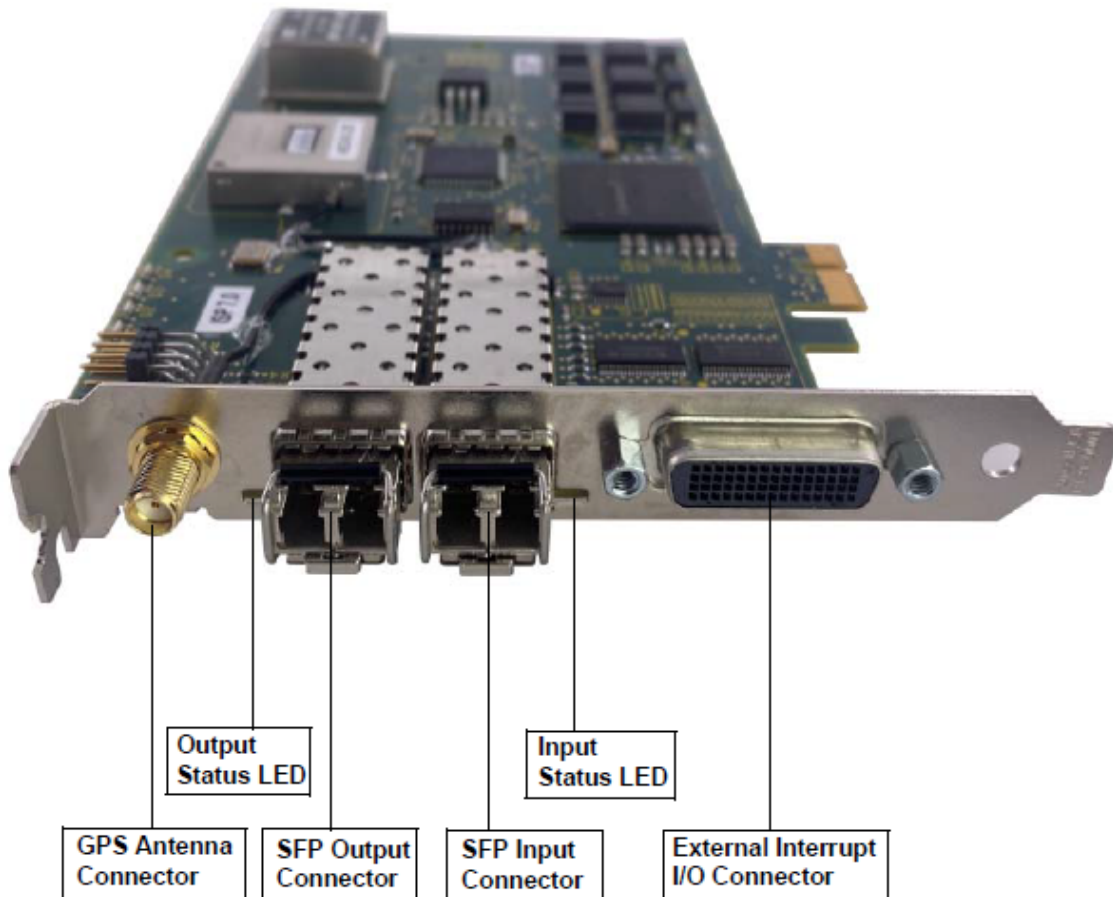
図2-4 RCIM IIIボード



コネクタとLED

図2-5はRCIM IIIボードの入出力コネクタとLEDを示します。LEDや各々のコネクタに関する詳細な情報は次の項で提供します。

図2-5 RCIM IIIのコネクタとLEDの位置



LEDの機能

RCIM IIIボードの入力および出力コネクタの近くに2つの2色のステータスLEDがあります。それらはボードがリセット・モードの時にテストとして赤と緑に短い間隔で明るく光り、その後どちらにも微かな赤に光ります。

ボードが正常に動作している間のLED機能は次のようになります：

LED	内容	機能
出力ステータスLED	赤固定	10MHzクロックの故障
	2秒毎に赤が点灯	ケーブル・オプション付きだがケーブルは同期していない、または失敗した状態でPOSIXクロックは停止中
	1秒毎に緑が点灯	ケーブル・オプション無しでPOSIXクロックは動作中
	1秒毎に緑が点滅	ケーブル・オプション付きでケーブルは同期した状態でPOSIXクロックは動作中
	2秒毎に赤と緑が交互に点灯	ケーブル・オプション付きだがケーブルは同期していない、または失敗した状態でPOSIXクロックは動作中
入力ステータスLED	2秒毎に赤が点灯	ケーブル・オプション付きだがケーブルは同期していない、または失敗した状態
	緑固定	ケーブル・オプション付きでケーブルが同期した状態

入出力ケーブルとコネクタ

RCIM IIIは1組の標準的なSFP (small form-factor pluggable)コネクタをRCIM IIIケーブルと接続するためのケージに差し込んで使用します。ケーブルはRCIM IIIボード間で割り込み、タイム・スタンプ、基準クロックを伝達するために使用されます。出力ケーブル・コネクタは、RCIMがマスタまたはRCIMチェーン間のスレーブのどちらかの場合に使用されます(RCIMモードの説明については2-14ページを参照)。入力ケーブル・コネクタは、RCIMがスレーブ・モードまたはRCIMチェーン間のスレーブの場合に使用されます。ケーブルの部品番号(HS002-3CBL-xx：xxはメートル長)はLC光ファイバー・ケーブルとRCIMのマスタおよびスレーブの空のケージに差し込む2つのSFPを含みます。ケーブルに関する詳細については「デジタイズ・チェーン・ケーブル」項を参照してください。

NOTE

SFPのケーブルはRCIMを実装しているシステムの電源が落ちた状態で取り付けおよび取り外しを行ってください。ESD(静電気放電)の注意については「取り付け」項を参照してください。注意としてはSFPモジュールを所定の位置に固定すること、RCIM IIIをSFPを取り付けている間にPCIeスロットから押し出されないことを確実に行うことが必要とされます。光ファイバー・ケーブルそのものはRCIM IIIにダメージを与えることなくいつでも取り付けおよび取り外しする事が可能です。

発振器

RCIM IIIに与えられた温度補償水晶発信器(TCXO: Temperature Compensated Crystal Oscillator)は±2.5PPM(100万分の1)の精度を持っています。

オプションの恒温槽付き水晶発信器(OCXO: Oven Controlled Crystal Oscillator)は±10PPB (10億分の1)の温度安定性を提供します。

GPSアンテナ

RCIM IIIのGPSオプションは有効なGPSアンテナと同軸ケーブルを含みます。

アンテナはGPS衛星信号を受信して受信機にそれを渡します。GPS信号は1575MHzレンジのスペクトラム拡散信号であり導電性または不透明な表面は貫通しません。従って、アンテナは空がきれいに一望できる屋外に設置する必要があります。

もし異なるアンテナまたはケーブルを使用する場合、以下の仕様に一致させる必要があります：

- 50 Ω インピーダンス
- 27dBゲイン
- 3.3V DC電源、最大30mA.

外部割込みI/Oコネクタ

RCIM IIIの外部割込みI/Oコネクタは、12個の出力と12個の入力を提供するMolex LFH-60 (Low Force Helix)です。

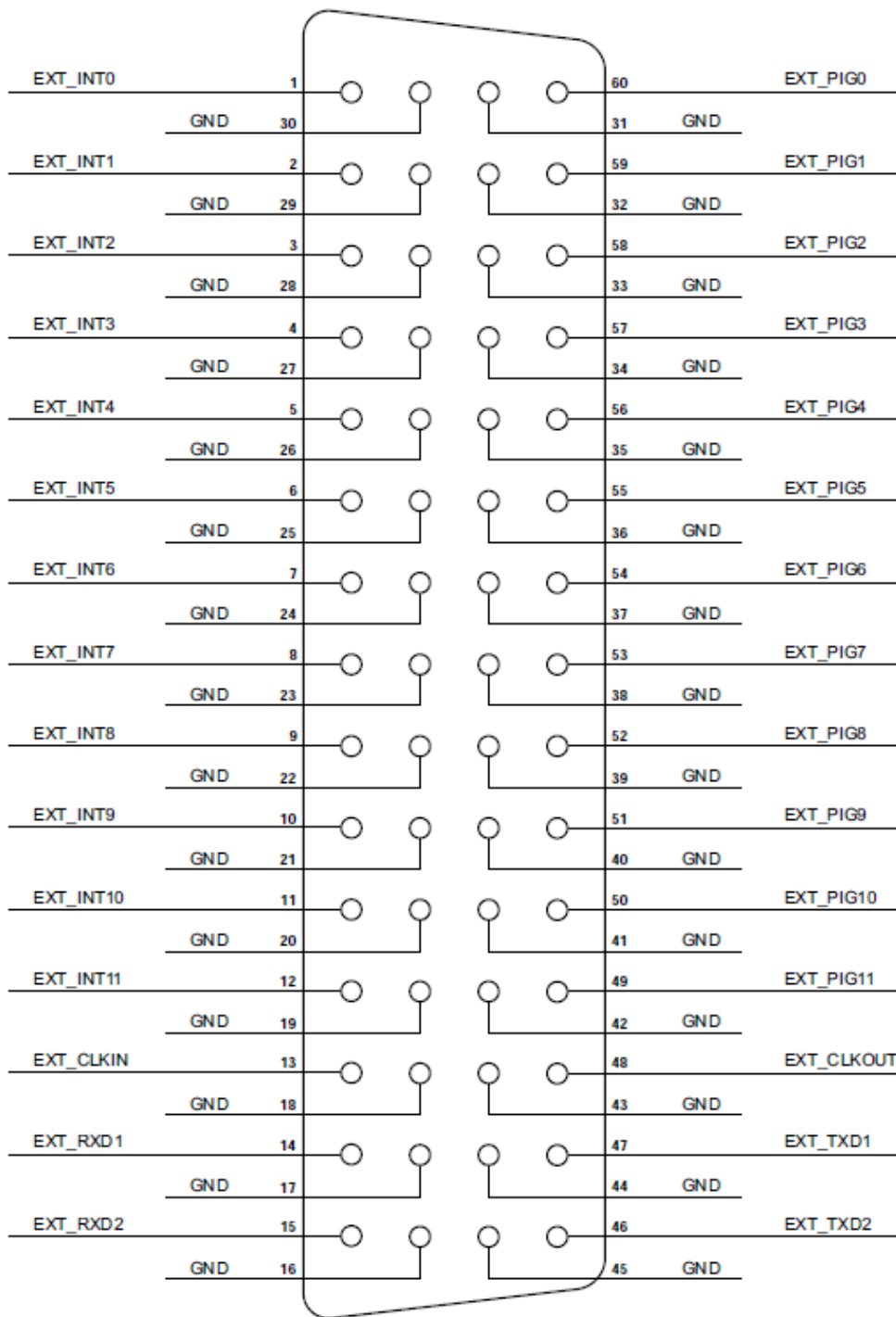
デフォルトでRCIM IIIの外部入出力信号は**負論理(Low True)**ですが、これは反転させることも可能です。詳細についてはB-6ページの「RCIM III Interrupt Enable/Request/Pending/Clear/Arm/Level/Polarity Registers」を参照して下さい。

外部出力は機器に取り付けてRCIMによる制御を可能にします。出力を駆動するためにプログラマブル割込み発生器(PIC)、リアルタイム・クロック・タイマー(RTC)、エッジ・トリガ割込み(ETI)、分配割込み(DI)のどれでも選択が可能なマルチプレクサによって出力が駆動されます。選択は一連の構成レジスタにより制御されます。

外部出力割込みとプログラム可能な割込みの利用に関する情報については3章機能説明を参照してください。

外部割込みI/Oコネクタのピン配列を図2-6に示します。

図2-6 RCIM III外部割込みI/Oコネクタのピン配列



外部割込みの入力信号は5V TTLレベルです。(EXT_PIG[0-11]と表示された)外部割込みの出力は74ABT16240ライン・ドライバを使い駆動されます。外部割込みの入力は+5Vに180オーム、グランドに330Ωと0.1μFが終端されています。この入力を駆動するには最低でも30mAを引き込むこと(シンク)が可能なライン・ドライバが必要となります。入力端子は外部割込み信号の速度を制限し、発生した疑似割込みからノイズを抑制します。殆どのライン・ドライバは流し込むこと(ソース)が可能な電流以上に引き込む事が可能なので、信号の立下りエッジはより速くなります。

EXT_CLKINとEXT_CLKOUTの各信号は外部の10MHzクロックの入力/出力用に使用されます。RCIM IIIを駆動する外部クロックは50オームの負荷に5V TTL信号を駆動する能力を持っている必要があります。RCIM IIIは1つでも存在する場合は自動的に外部クロックの利用に切り替えます。RCIM IIIからの外部クロック出力は74ABT16240ライン・ドライバを使って駆動されます。

EXT_RXD1, EXT_TXD1, EXT_RXD2, EXT_TXD2の各信号はRS-232Cレベルの信号です。これらは今のところデバッグ目的のために使用されます。

システムの確認

下の**lspci(8)**の出力は、RCIM IIIに関するPCIクラス、ベンダーID、デバイスIDを示します(0e:04.0 (bus:slot,function)はシステムで異なります)：

```
# lspci -v | grep -i rcim
0e:04.0 System peripheral: Concurrent Real-Time, Inc. RCIM III
Real-Time Clock & Interrupt Module (PCIe) (rev 01)
# lspci -ns 0e:04.0
0e:04.0 Class 0880: 1542:9271 (rev 01)
```

デジター・チェーン・ケーブル

RCIM IIIは、RCIMチェーンにRCIM IIIを接続するために光ファイバー・シリアル同期ケーブルとSFP(small form-factor pluggable)コネクタ(P/N: HS002-3CBL-xx)を使用します。ケーブル上のシリアル・データはケーブルの問題を検出させる事が可能なパリティとフレミング情報を含みます。ポーリングが連続的に行われ、エラー状態が検出された時にRCIM IIIのデジター・チェーン・ケーブルのステータスを報告するメッセージが出力されます。問題を表しているメッセージは、失敗しているリンクが接続されているシステムに直接表示されます。

シリアル・ケーブルはポイント・トゥ・ポイント接続です。「Input」ケーブルは上流のマスタRCIMに向かっていているケーブルを指します。「Output」ケーブルはマスタから離れる下流の接続です。

```
RCIM: Input cable disconnected.
RCIM: Input cable connected.
RCIM: Input cable connected but not synchronized.
RCIM: Input cable unsynchronized.
RCIM: Input cable O.K.

RCIM: Output cable disconnected.
RCIM: Output cable connected.
RCIM: Output cable connected but not synchronized.
RCIM: Output cable unsynchronized.
RCIM: Output cable O.K.

RCIM: Cable error on input cable.
RCIM: Cable error on output cable.
```

「disconnected」と「connected」のメッセージはSFPがRCIM IIIの適切なケージに差し込まれているかどうかに基づいてのみ発生します。

これらは光ケーブルが差し込まれているまたは外されている場合は発生しません。これらはSFPが正しく差し込まれていないまたは故障中の場合を除き、通常の操作中には発生しません。

「not synchronized」と「unsynchronized」のメッセージはケーブルが通信の試みに応答していない事を示します。これらのメッセージは光ケーブルが取り付けられているまたは外されている場合に発生します。これらは接続されたシステムの電源がOFFの場合にも発生します。

最後の2つのメッセージは、ケーブルのパリティ・エラーや一時的なケーブル同期の失敗のような一時的なエラーを示します。一時的なエラーが発生した場合、再同期するためにケーブルのリンクを必要とする可能性があります。分配割込みがケーブル上をブロードキャストしている場合、失敗する可能性があります。

一時的なエラーはケーブル・クロックが全てのシステムに届かなくなるためティック・タイマーの同期にも影響します。クロック同期に関する説明については3章を参照して下さい。

接続モード

各システムのRCIMボードが互いにチェーン接続されている場合、割込みを接続された全てのRCIMへ、RCIMから対応する全てのホスト・システムへ一斉に分配させる事が可能です。

NOTE

チェーン内のRCIMは全て同じモデル、もしくはRCIM IIIボードとRCIM IVボードの組み合わせにすることも可能です。

ご使用のシステムをRCIMチェーンの一部とする場合、RCIMをインストールする前に望む接続モードを決定することが最適です(RCIMをインストールする前に光ケーブルをケーブル・コネクタへ接続するのは簡単なことです)。

RCIMチェーンを再構成するには、システムの電源をOFFにしてケーブルを移動した後に再起動する必要があることに注意してください。ドライバーはブート時にマスターRCIMであるかどうかを判断し、ケーブル・クロックを制御するためにマスター・システムを構成して有効化します。システムを再起動せずにケーブルを入れ替えるとケーブル・クロックに関連する問題が発生します。

RCIMは次のモードのいずれかで接続する事が可能です：

Isolated mode	他のRCIMとの接続なし
Master mode	このRCIMはRCIMチェーンの先頭にある。このRCIMに入るケーブル接続はなく出て行くケーブル接続のみ。RCIMマスターはクロックを制御するのでその中で一つのみ(説明については3章を参照)。
Pass-through Slave mode	このRCIMは他の2つのRCIMと接続されている。チェーン内の前方のRCIMから来ている入力ケーブル接続、およびチェーン内の次のRCIMに向かう出力ケーブル接続がある。
Final Slave mode	このRCIMは他の1つのRCIMと接続されている。最後のスレーブRCIMに加わる入力ケーブル接続はあるが、外に出る出力ケーブル接続はなし。

RCIMの開梱

輸送箱から機材を開梱する時、梱包リストを見て全ての品目があることを確認してください。機材を保管および再送用に梱包材は保存しておいて下さい。

NOTE

梱包箱が受け取り時に損傷している場合、機材を開梱および検品している間は運送業者に居合わせるよう要求してください。

取り付け

通常、このカードの取り付けや設定はConcurrent Real-Timeで行われます。本情報は製造後の環境にあるシステムにRCIMを追加するようなケースのために提供します。

RCIMを無事導入するため、RCIMを外部割込みの受け取りもしくは配信のどちらで使用し、どのモード(Isolated, Master, Pass-through Slave, Final slave)でRCIMが実行されるのかを知る必要があります。詳細については「接続モード」項を参照してください。

CAUTION

静電放電は回路を損傷する可能性がありますので集積回路の場所を触れる事は差し控えてください。

Concurrent Real-Timeは、システムの導入またはアップグレード時に帯電防止リスト・ストラップや導電性フォーム・パッドの利用を強く推奨します。ディスク・ドライブ、コンピュータ・ボード、メモリ・モジュールのような電子部品は静電放電(ESD: Electrostatic Discharge)に対して極めて敏感である可能性があります。システムまたは保護用カバーからボードを取り外した後、接地した静電気のない面の平らな所に部品側を上にして置いて下さい。どのような表面上でもボードをスライドしないで下さい。

ESD装置が利用できない場合、システム筐体の塗装されていない金属部分に取り付けた(電気店で入手可能な)帯電防止ストラップを身に着けることによりESDに起因する損傷を回避する事が可能です。

システムにRCIMを組み込むには以下の手順を利用してください：

1. システムの電源がOFFであることを確認してください。
2. システムから電源ケーブルを外してください。
3. システムのケースを開けてRCIMを実装したいPCIeスロットを特定してください。一般的に、存在するほかのデバイスが最小もしくは競合しない、かつ可能であればIRQ優先度が最高のスロットにRCIMを設定するのが最善です。

4. RCIMを要望するスロットへ組み込み、ケースが提供する仕組みを使ってカードをスロットに固定してください。
5. もしRCIMチェーンの一部とする場合、必要に応じたケーブルを取り付けてください。このシステムの接続モードに基づくケーブルを接続する方法を確認するには「接続モード」項を参照してください。
6. もしオプションのGPSモジュールを搭載するRCIMボードを組み込む場合、GPSアンテナのリードを取り付けてアンテナを組み込んでください。アンテナは屋上または空地内に取り付ける必要があります。アンテナに関する詳細については「コネクタとLED」項と「GPSアンテナ」項を参照してください。
7. カバーを元に戻します。
8. システムに電源ケーブルを取り付けます。
9. 電源を入れてシステムをブートします。

構成

カーネル構成

下のRedHawk Linuxカーネル・パラメータはRCIMに関連しています。全てがカーネル構成GUIの「Character Devices」選択項目を通して利用可能であり、全てのプレビルトRedHawk Linuxカーネルでデフォルトで有効となっています。

RCIM	本パラメータはRCIMドライバをカーネル内に設定します。望むのであれば、これはモジュールとして構成することが可能です。
RCIM_MASTERCLOCK	本パラメータはRCIMをシステム時刻の監視や調整を行うマスター・クロックとして使用することを有効にし、結果、システムの時間管理が更に正確になります。
RCIM_IRQ_EXTENSIONS	本パラメータは他のドライバが自分自身の割込みルーチンをRCIMドライバに加える事を可能にします。 Frequency-Based Scheduler (FBS)はこのサポートを必要とします。
MULTI_RCIM_MAX	規定値は24。この数値パラメータはシステムに構成することが可能な単一のRCIMカードの最大値を定義します。

カーネル調整パラメータの変更やカーネルの構築に関する詳細については、「RedHawk Linux User's Guide」(文書番号0898004)を参照してください。

ドライバ構成

RCIMボードはアプリケーションが利用するために設定することが可能な多くの機能を持っています。設定は対象となるRCIMボードに関連する構成ファイルに対してカンマで区切られた連続したトークンを反映することによって行われます。各トークンは単一機能の構成オプションを変更する方法で規定します(規定されない機能のトークンは変更されません)。

例えば、以下のコマンドは立下りエッジで始動する**eti1**エッジ・トリガ割り込みに変更します：

```
echo "eti1/f" > /proc/driver/rcim:0/config
```

トークンを囲む引用符は常に望ましく、以下のコマンドの例のように垂直バーを含んだ構成要求を囲むために使用する必要があります：

```
echo "rtc0|di1" > /proc/driver/rcim:0/config
```

上記コマンド実行後、RCIMの**rtc0**リアルタイム・クロックはRCIMの**di1**分配割り込みへ送られます。

次はRCIMスレーブ・システムに対して更に複雑な例です：

```
echo "host/server1.ccur.com, eti1/rising, di3/high, rtc3|di6" > /proc/driver/rcim:0/config
```

本コマンドは以下の動作全てを実行します：

- マスターRCIMのホスト名を**server1.ccur.com**に設定
- **eti1**エッジ・トリガ割り込みを立上りで発生に設定
- **di3**分配割り込みをハイの値で発生に設定
- **rtc3**リアルタイム・クロックを**di6**分配割り込みに転送

全てのRCIM構成オプションは以下のカテゴリの1つに収まることに注意してください：

- 様々な割り込みのどれで発生させるかを設定： 立上り/立下りエッジ、ハイ/ロー・レベル。
- 内部的に生成された割り込み信号、出力ライン、分配割り込みライン間の関連性を設定。
- RCIMチェーン内のマスターRCIMを所有するシステム名称を設定。
- RCIMチェーン内にあるRCIMに対して、ローカルRCIMの発振器もしくはマスターRCIMの発振器のどちらでティックとPOSIXクロックを駆動するのかを決定。

設定の変更はRCIMの構成ファイルへの書き込み許可(例えばrootでのアクセス)が必要であり、RCIMが使用中ではない時にのみ行う必要があります。

設定の修正がシステム再起動の間に自動的に記憶されることはありません。それらを持続させるには、システム起動中に必要なRCIM設定を生成する**/etc/sysconfig/rcim**ファイルを変更してください。

RCIM構成に関する詳細については**rcim(4)**のmanページの「*Available Configuration Options*」項を参照してください。

一般論

RCIMシステムを構成する際は、以下に留意してください：

- RCIMチェーンに関して、マスター内で増加し続けるクロック・シグナルは全てのスレーブにブロードキャストされるため、全てのスレーブRCIMのティック・クロックおよびPOSIXクロックはマスターで同期されます。一旦、全てのRCIMのクロックが最初に同期されると同期されたままとなります。

ティック・クロックを同期するには、全てのシステム間で動作しているTCP/IP接続が必要となります。更に各スレーブRCIMのホスト名の構成にマスターRCIMを設定する必要があり、各スレーブはブート時に一度initスクリプトで**rcim_clocksync**を実行する設定する必要があります。これはアプリケーションが同期用にティック・クロックを使う場合のみ必要となります。

システム・タイムキーピング・デーモンはPOSIXクロックを同期するために使用することが可能ですが、より優れたメカニズム**rcimdate(8)**があります。RCIMマスターは1秒毎に1回、自身のPOSIXクロックをRCIMのケーブルにブロードキャストします (**rcimdate**はスレーブのPOSIXクロックを正確にマスターに一致させるためにこれを利用)。

これはシステム間のTCP/IP接続を必要としない、同期がより高速、同期が非常に正確であるため、システム・タイムキーピング・デーモンよりもいくつもの優位性を持っています。

- 割り込み(ローカルで動作またはRCIMチェーン全体に配信)は各システムで構成された値に従い処理されます。もし設定された規定値とは異なる方法で機能させたい場合、必要な構成オプションを指定する必要があります。
- RCIMチェーンのシステム全体に割り込みを分配する場合、全てのシステムが分配割り込みラインに対して対応する構成を行う必要があります。
- デフォルトでRCIM IVの外部入出力信号は**正論理(High True)**ですが、RCIM IIIの外部入出力信号は**負論理(Low True)**です。必要に応じて極性を反転させる制御レジスタが両方に存在します。例えば、外部割り込み入力信号はレベル割り込みに関しては極性制御、エッジ割り込みに関しては立下り/立上りエッジ制御を持っており、プログラマブル割り込み発生器(PIG)出力は任意の状態にセットまたはリセットすることが可能です。

MSI割り込み構成

RCIM IVおよびRCIM IIIの最新バージョン(リビジョン9以上)はMSI (Message Signaled Interrupts)をサポートします。デフォルトで、RCIMのカーネル・ドライバはPCI INTA割り込みの代わりにMSI割り込みを使用するために可能な時にハードウェアを初期化します。MSI割り込みを使用することにより、RCIMは特有の非共有割り込みを有する事が保証され、従って更に信頼のある割り込み応答時間を提供します。

加えて、RCIM IVはMSI-X規格を完全にサポートしており、互換性のあるハードウェア上で最大16個の異なる割り込み(例えば、**irq8**)を個々のソース(例えば、**rtc3**)に加えることが可能です。詳細については3-17ページの「MSI-X割り込み構成」を参照して下さい。

RCIMドライバは前述の**rcim=**構成オプションとは関係のない**rcim.nomsi=1**オプションを持っています。全てのドライバのバージョンがこのオプションを持っています。指定した場合、MSI機能はそれをサポートする全てのRCIMボードが無効となります。ボードのMSI無効の有無を選ぶためのメカニズムはありません。本オプションが指定された場合、RCIMドライバはPCI INTA割り込み方式の使用に逆戻りします。

性能上の理由により、本オプションはMSI割込みの問題が発生する場合にのみ使用する必要があります。

静的にリンクされたRCIMドライバについては、この調整はGRUBブート・ローダのコマンド行(`rcim.noms=1`)で指定することが可能です。モジュール形式のRCIMについては、この調整は「`options rcim.noms=1`」として`/etc/modprobe.conf`に設定する事が可能です。

本章ではRCIMが提供するクロックと割込みの機能および各々のユーザー・インターフェースについて説明します。

概要

Real-Time Clock and Interrupt Module (RCIM)は2つの非割込みクロックを提供します。そのクロックの1つはシステム全体に共通のタイム・スタンプを提供するためにRCIMチェーンの全てのRCIMを同期させることが可能です。もう一方のクロックはPOSIX 1003.1に準拠しており、RCIMチェーン全体で同期はされませんが、RCIMボードの他のクロックと揃ってインクリメントし特定の時間にセットする事が可能です。

クロックに加え、シグナル処理(割込み)の操作に関する以下の方式が利用可能です：

- エッジ・トリガ割込み Edge-Triggered Interrupt (ETI)
- リアルタイム・クロック Real-Time Clock (RTC)
- 外部出力割込み External Output Interrupt
- プログラマブル割込み発生器 Programmable Interrupt Generator (PIG)
- 分配割込み Distributed Interrupt (DI)

これらの割込みはRCIMシステム上でローカルに機能する、またはRCIMチェーン内の全てのRCIMシステムに分配させる事が可能です。 **open(2)**, **close(2)**, **ioctl(2)** システム・コールは割込みを操作するために使用されます。各々のデバイス・ファイルはそれぞれの割込みに関連付けられています。

クロックと割込みについては本章内で説明します。

クロック

RCIMは2つの非割込みクロックを提供しており、次の項で全て説明します。

tick	共通の400nsクロック・シグナルの各ティックで1ずつ増加する64bitの非割込みクロック。共通のタイム・スタンプを提供するため、このクロックはゼロにリセットされRCIMチェーン全体で同期させることが可能です。
POSIX	POSIX 1003.1形式にエンコードされた64bitの非割込みクロック。上位32bitは秒を含み下位32bitはナノ秒を含んでいます。このクロックは共通のクロック・シグナルの各ティックでインクリメントされます。これは主に高分解能ローカル・クロックとして使用されます。GPSを搭載するボードのGPS標準時間にシステム時刻を同期するために構成することが可能です。

RCIMチェーン内の全てのRCIMの全てのクロックは、マスターRCIMから発する共通のクロック・シグナルで全て駆動するため一斉にインクリメントされます。

ティック・クロック

ティック・クロックは共通のクロック・シグナルの各ティックで1ずつインクリメントする64bitの非割込みカウンタです。特定の時間に設定は出来ませんが、増加させるまたはゼロに設定する事は可能です。従って、ティック・クロックは正確な日時が必要となる場合に現在の日時に近づけるために動的に調整する事ができません。

RCIMボードがRCIMチェーンの一部である場合、全てのスレーブRCIMのティック・クロックはマスターRCIMにあるティック・クロックでインクリメントやクリアが行われる何でも同期してインクリメントおよびクリアされます。

デバイス・ファイル`/dev/rcim:N/sclk` (*N*はゼロから始まるRCIMカードの番号)がプログラムのアドレス空間にマッピングされている場合は、ティック・クロックはダイレクト・アクセスを使いマスターまたはスレーブのどちらのシステムでも読む事が可能です。詳細については後述の「クロックへの直接アクセス」項を参照して下さい。

デフォルトで、ティック・クロックの初期化(ゼロ化)およびRCIMチェーンのほかのティック・クロックとの同期はRCIMマスターのブート時にいつでも自動で行われます。ティック・クロックの初期化と同期は`rcim_clocksync(1)`コマンドを使って達成します。詳細については後述の「ティック・クロックの同期」項および「`rcim_clocksync`ユーティリティ」項を参照して下さい。

POSIXクロック

POSIXクロックはPOSIX 1003.1形式にエンコードされた64bitの非割込みカウンタです。上位32bitは秒を含み下位32bitはナノ秒を含みます。本クロックは共通のクロック・シグナルの各ティックでインクリメントされ、システム・クロックはPOSIXクロックに同期されます。

(例えば`clock_settime(2)`を使った)システム・クロックの設定はシステム・クロックとRCIM POSIXクロックの両方を新しい時刻に設定します。加えて、POSIXクロックは`mmap`によりマッピングし、アプリケーションで他のRCIMのレジスタのように読む事が可能です。しかしながら`mmap`方式を介したRCIM POSIXクロックの変更は、システムがRCIM POSIXクロックに同期している間は推奨しません。

POSIXクロックは希望する時刻を取り込むことが可能ですが、取り込んだ値はRCIMチェーンの他のクロックに同期しません。そのホストに取り付けられたRCIMのPOSIXクロックだけを更新します。POSIXクロックのアクセスに関する詳細については後述の「クロックへの直接アクセス」項を参照して下さい。

マスターにPOSIXクロックを同期させるために各スレーブにて`rcimdate`コマンドを実行する事が可能です。TCP/IP接続または他のソフトウェアは必要ありません。`rcimdate`はRCIMマスターより1秒毎に1回RCIMケーブルに送信されるPOSIXのタイム・スタンプを利用します。

GPSモジュールを持つRCIMシステムでシステム時計管理ソフトウェアが実行中(例えば、`ntpd`単独または`chronyd`を使用する`gpsd`)は、GPS受信機がGPS時刻に所属するRCIM上のPOSIXクロックを同期するために使用されます。RCIMに搭載した1つのGPSがRCIMチェーン内の全てのiHawkシステムを同期することが可能です。GPSモジュールを搭載する複数のRCIMはシステム間のケーブル接続無しで共通の時間基準を提供することが可能です。絶対GPS時刻を基準とするPOSIXタイマーは、物理的に接続されていないシステムにおいて同時にプログラムの実行を開始するために使用することが可能です。

詳細については「システム時刻管理のためのGPSの利用」項を参照して下さい。通常、マスターRCIMだけがGPSを必要とし、スレーブはマスターにPOSIXクロックを同期させるために利用可能なソフトウェアの手法のいずれかを使用します。

クロックへの直接アクセス

デバイス・ファイル`/dev/rcim:N/sclk` (N はゼロから始まるRCIMカード番号)は`mmap(2)`を使って直接RCIMクロックにアクセスするために使用することが可能です。`mmap`で返されたアドレスから、以下のオフセットをクロック領域にアクセスするために使用します。

0x0	ティック・クロックの上位32bit
0x8	ティック・クロックの下位32bit
0x10	ステータスと制御(変更不可)
0x100	POSIXクロック(秒)
0x108	POSIXクロック(ナノ秒)
0x110	ステータスと制御(変更不可)

これらのオフセットはヘッダー・ファイル`/usr/include/linux/rcim.h`内に(RCIM_SYNCCLOCK_で始まる名称で)定義されています。

POSIXクロックの値を設定するには、`rcim_clocksync(1)`ユーティリティを「update」コマンドを使い対話モードで使用することが可能です。

クロックの同期

本項はRCIM上のクロックに同期するために使用されるテクニックとツールについて説明します。

rcim_clocksyncユーティリティ

`rcim_clocksync(1)`ユーティリティは、システム全体に共通のタイム・スタンプを提供するため、接続された全てのRCIMのティック・クロックをゼロにリセットするために使用することが可能です。この同期の操作はRCIMマスター・システムのブート時に自動的に行われます。スレーブ・システムが利用可能になった時、RCIMチェーン全体のティック・クロックを同期するために本コマンドの再発行が必要となりますが、これは自動化することが可能です(後述の「自動同期」項を参照)。`rcim_clocksync`は接続された全てのRCIM上のPOSIXクロックを同期するために使用することも可能です。その手順については「POSIXクロックの同期」項で説明します。

システム・クロックはRCIMに同期するので`rcim_clocksync`がシステム起動時以外で実行される場合、多くの影響があるため慎重に使用する必要があることに注意してください。マスターとスレーブのシステム時刻はPOSIXクロックに同期し、システム時刻の進行が停止した場合はこれらのクロックを使用している時間ベースの機能は停止します。

同期は常にRCIMマスターまたは独立システムでは成功し、RCIMスレーブのシステムで同期が成功しなかった場合にエラーが返されます。

RCIMマスターのシステムでオプション無しで`rcim_clocksync`を指定するとRCIMチェーン内の全てのティック・クロックが同期します。

rcim_clocksyncは次のオプションを受け取ります：

- i** 対話モード(以下参照)
- m** RCIMマスターを指し示す構成されたホスト名称を出力(2章の「構成」を参照)
- s** RCIMの接続状態を出力
- devname** 必要とするRCIMボードのデバイス名称。ブート時に見つけた最初のRCIMボードはデフォルトで/dev/rcim:0/rcim。
/dev/rcim:1/rcim, /dev/rcim:2/rcim,,,はブート時に見つけた2番目,3番目,,,のRCIMボードを指すためにこのツールで使用されます。

対話モードが呼び出された場合、下の例に似た表示を構成およびステータス、同様にコマンドの使用法を2秒毎に更新して提供します。これらの項目について以下説明します。

```

RCIM is isolated                      RCIM version: 3
Configured RCIM master hostname is Not_Configured

Clock status and values ...
cable signal    : ENABLED
tick timer     : CABLE_ENABLE LOCAL_ENABLE
posix clock    : CABLE_ENABLE LOCAL_ENABLE
tick timer     :                10.3361 seconds (        25840213 ticks)
posix clock    :                18665.4696 seconds

operations are:
s            - synchronize clocks
0[tp]       - stop clock ([t]ick/[p]osix)
1[tp]       - start clock ([t]ick/[p]osix)
w[tp]       - update clock value ([t]ick/[p]osix)
i[tp]       - isolate clock ([t]ick/[p]osix)
c[tp]       - connect clock ([t]ick/[p]osix)
d            - disable cable clock signal
e            - enable cable clock signal
q            - quit

enter operation

```

RCIM is このシステムのRCIMのモード: master, pass-through, slave, final slave, isolated

RCIM version RCIMのバージョン番号

Configured RCIM master hostname is
RCIMマスターのホスト名称。これは構成オプションhostを使って構成される必要があります(2章の「構成」を参照して下さい)

cable signal 以下のいずれか：
ENABLED/DISABLED – RCIMマスターにおいては、ケーブル・クロック・シグナルをスレーブに伝達しているかどうかを示す。RCIMスレーブにおいては、クロックがRCIMマスターにより駆動されているまたは(同期なしで)ローカルで時間を刻んでいるかどうかを示す。
CLOCK_MISSING – エラー状況を示しているケーブル・クロック・シグナルがスレーブに正しく伝達されていない。
CLOCK_STOPPED – エラー状況を示しているケーブル・クロック・シグナルは停止された。

Status:	以下のいずれか：
tick timer	CABLE_SYNC – 利用可能である場合にRCIMスレーブ・クロックがRCIMマスター・ケーブル・クロック・シグナルで駆動されていることを示す。
posix clock	CABLE_ENABLE – RCIMマスターが同期を行った時にRCIMクロックをリセットした事を示す。
	LOCAL_ENABLE – クロックが有効化されたことを示す。
	NO_RESET_WHEN_DISABLED – 無効化したときにクロックがリセットされない事を示す。
Values:	各クロックにおける現在のクロック値
tick timer	
posix clock	
operations are:	この部分是对話モードの使用法に関する情報。 enter operation: プロンプトにて、その操作について記載された役割を達成するためにいずれかの操作コードを提供します。前述のとおり、一部の操作は操作するクロック(ティック・クロックは t 、POSIXクロックは p)の指定を必要とします。

ティック・クロックの同期

RCIMマスター・システムが起動する時、**rcim_clocksync**が実行され、RCIMチェーン内の全てのティック・クロックをゼロにリセットします。

マスターが起動した後にRCIMスレーブ・システムが起動した時、自動によるクロック同期が構成されていない限り、RCIMチェーン内のシステムのティック・クロックは同期されません(後述の「自動同期」項を参照)。マスターRCIMシステムで**rcim_clocksync**をオプション無しで呼び出すとRCIMチェーン内の全てのティック・クロックを同期させます。このユーティリティに関する詳細については「**rcim_clocksync**ユーティリティ」項および**rcim_clocksync(1)**のmanページを参照して下さい。

RCIMマスタークロックの留意事項

RCIM POSIXクロックはシステムのマスタークロックです。これはシステム時刻が頻繁にRCIMのPOSIXクロックを監視し一致させるために自身で調整する事を意味します。

RedHawkの以前のリリースでは、壊れたまたは停止したRCIMはシステムが混乱し頻繁に動かなくなっていました、これはもはや事実ではありません。現在のRedHawkのマスタークロック・コードは頻繁にRCIM POSIXクロックの正当性を検査し、問題が検出された場合はクロックの同期を停止します。これはRCIM POSIXクロックとシステム・クロックの2つのクロックが自由に実行している状態である事を意味します。RCIM POSIXクロックが再び有効になり次第クロックの同期が再開されます。

同期を再開させるため、RedHawkはシステム・クロックとRCIM POSIXクロックがインクリメントされている事、各クロックが正確な速度でインクリメントされている事、各クロックが互いに約2秒以内の値である事を検出する必要があります。この後者の状態は**clock_settime(3)**の実行により最も簡単に得られますが、**masterclock(5)**に記述されているようなほかの方法は、特定の状況においてはより適している可能性のある前者で利用可能です。

NOTE

必要ではない場合にRCIMのマスタークロック動作を無効化する情報については3-22ページの「マスタークロックの無効化」を参照して下さい。

POSIXクロックの同期

POSIXクロック・ティックは同期しますが、通常は同じクロック値を持たないので共通のタイム・スタンプとして使用するべきではありません。必要であれば各スレーブで**rcimdate**を実行することで、これらを一致させるためにRCIMチェーン内の他のPOSIXクロックに同期させることが可能です(戻る時にスレーブのRCIM POSIXクロックがマスターに同期されます)。

この操作はRCIMティック・クロックも同期させることに注意してください。この手順を実行している間はシステム時刻は停止し、時間ベースの機能は影響を受ける事を認識してください。

1. RCIMスレーブ・システムの全てのPOSIXクロックが**cp**コマンドを使って接続されている状態を確保して下さい。
2. RCIMマスター・システムのケーブル・クロック・シグナルを(**d**コマンドを使って)無効にして下さい。全システムのPOSIXクロックが時間を刻むのを停止します。
3. 各システムのPOSIXクロックの時刻の値を(**wp**コマンドを使って)同じ値に更新して下さい。
4. RCIMマスター・システムのケーブル・クロック・シグナルを(**e**コマンドを使って)再度有効にして下さい。全てのクロックが時間を刻み始めます。

オプションのGPSモジュールを搭載するRCIMシステムでは、POSIXクロックを標準GPS時刻に同期させることが可能です。詳細は後述の「システム時刻管理のためのGPSの利用」項を参照して下さい。

自動同期

スレーブのPOSIXクロックをブート時にマスターに自動的に同期させるよう各々のRCIMスレーブ・システムを構成させることが可能です。それを行うには、スレーブの **/etc/sysconfig/rcim**内の**RCIMDATE=continuous**行または**RCIMDATE=oneshot**行のどちらかをコメント解除して下さい。その後、スレーブが起動するたびに**rcimdate**が1回または断続的に実行されます。それが1回実行された場合、その後スレーブのPOSIXクロックはマスター上の一部のアプリケーションが**clock_settime(3)**を実行するような時(通常は**ntp**または**ptpd**が行う)までマスターに一致します。**rcimdate**が連続モードで実行される場合、マスター上で生じた**clock_settime(3)**の数秒以内にスレーブは**clock_settime(3)**を実行します。

RCIMスレーブ・システム起動時にティック・クロックを自動的に同期するようにシステムを構成することも可能です。この機能はデフォルトでは無効であり、慎重に必要があります。これはRCIMチェーン内のどのシステムが起動した時にも全システムのティック・クロックをゼロにリセットしてしまい、同期中のティック・クロックを使っているプロセスに好ましくない影響を与える可能性があります。

システム時刻管理のためのGPSの利用

オプションのGPSモジュールが搭載されているRCIMシステムでは、標準GPS時刻にシステム時刻を同期させるためにPOSIXクロックを使用することが可能です。システム・タイムキープینگ・デーモンはタイム・サーバーとしてGPS受信機が利用され処理します(例えば、**ntpd**単独または**chronyd**を使用する**gpsd**)。これをサポートするためにシステム・タイムキープینگ・デーモンを構成する方法についての詳細は、4章の「GPSクロックの同期」項を参照して下さい。

NOTE

RCIMスレーブとして構成されたRCIMは光ケーブルを介して自動的にRCIMマスターとシステム時刻を同期します。この自動同期のためにRCIMスレーブはGPSタイミング・ソースに同期するよう構成することが出来ませんので、ChronydまたはNTPDを決して有効にしないで下さい。

RedHawk Linuxは、GPS PPSにシステム時刻を同期するRFC-2783 Pulse Per Second (PPS)インターフェースを含んでいます。POSIXクロック・レジスタはGPS PPSシグナルのエッジを定期的に取り込みます。これは、システム時刻とGPS時刻との間の誤差を取り込むために割込みが使用された場合に取り込まれるジッターを回避します。

専用のシリアル・インターフェース(`/dev/rcim_uart`)はGPS受信機と通信するためにソフトウェア・デーモン(例えば、**ntpd**単独または**chronyd**を使用する**gpsd**)に使用されます。このシンボリック・リンクはGPSを搭載する最後に見つかったRCIMを指し示します。他のRCIMが使用される場合、管理者はこの専用のデバイスを必要なRCIMの専用のuartデバイス・ファイル(例：`/dev/rcim:3/uart`)にリンクする必要があります。

同様に、GPS PPSを受信するデバイスは`/dev/refclock-0`にリンクされている必要があります。このシンボリック・リンクもまたGPSを搭載する最後に見つかったRCIMにリンクします。それが必要とするGPSではない場合、管理者はこのシンボリック・リンクを(例えば)`/dev/rcim:3/gps`にリンクする必要があります。

ntpq(1), **ntpd(1)**, **chronyc(1)**のようなツールは時刻管理の動きを監視するために使用することが可能です。それ以外の補助機能として必要に応じて`/etc/ntp.conf`または`/etc/chrony.conf`へ加えることが可能な様々なログファイルを含みます。詳細については**ntpd(1)**および**chronyd(1)**のmanページを参照して下さい。

システム時刻管理のためのIRIGの利用

オプションのIRIGモジュールが搭載されているRCIMシステムでは、IRIG時刻にシステム時刻を同期させるためにPOSIXクロックを使用することが可能です。オプションのIRIGモジュールはIRIG Bタイムコード仕様に準拠するIRIGマスターまたはスレーブとしてRCIM IVを機能させます。IRIGサポートを構成する方法についての詳細は、5章の「IRIGタイムコードの同期」項を参照して下さい。

NOTE

RCIMスレーブとして構成されたRCIMは光ケーブルを介して自動的にRCIMマスターとシステム時刻を同期します。この自動同期のためにRCIMスレーブはIRIGタイミング・ソースに同期するよう構成することが出来ませんので、ChronydまたはNTPDを決して有効にしないで下さい。

割込み処理

以下の1つ以上のモジュールがRCIMの割込み処理のために使用されます：

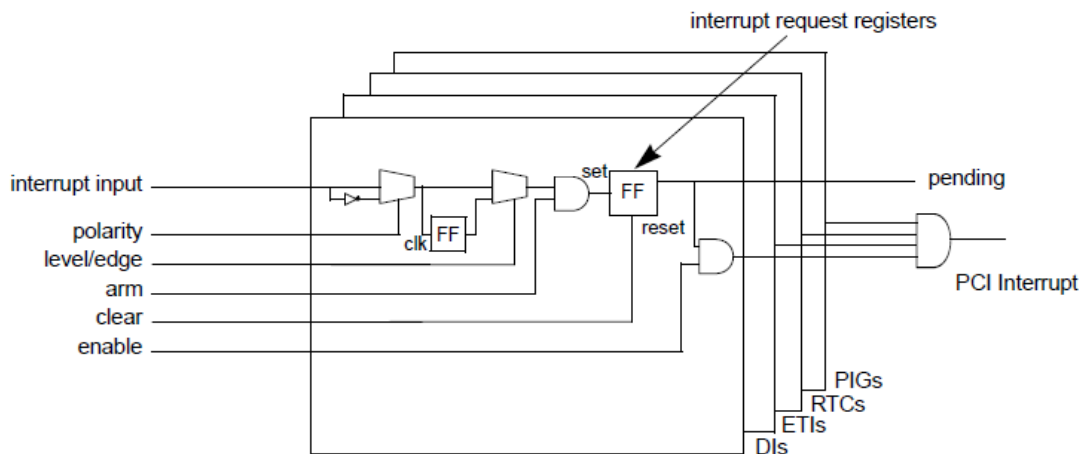
- **エッジ・トリガ割込み Edge-Triggered Interrupt (ETI)** – ETIは割込み発生させるために外部イベントを使用することを可能にします。ETIに関する説明は3-12ページから始まります。
- **リアルタイム・クロック Real-Time Clock (RTC)** – RTCは割込みを発生させるためにカウンタを設定する事を可能にします。RTCに関する説明は3-14ページから始まります。
- **外部出力割込み External Output Interrupt** – 外部出力信号は外部デバイスへの割り込みのためのソースとして他のシグナル処理モジュールのいずれかを使用することを可能にします。外部出力割込みに関する説明は3-15ページから始まります。
- **プログラマブル割込み発生器 Programmable Interrupt Generator (PIG)** – PIGは割込みを発生させるために使用する事が可能なシグナルをプログラムで生成することを可能にします。PIGに関する説明は3-19ページから始まります。
- **分配割込み Distributed Interrupt (DI)** – 分配割込みはRCIMチェーンで接続されている全てのシステム間にシグナルまたは割り込みを分配することを可能にします。DIに関する説明は3-20ページから始まります。

次の項ではRCIMの割込みをどのように処理するかを説明します。

割込み処理ロジック

割込み要求(RCIMボードもしくはソフトウェア要求により生成するかどうか)はエッジ・トリガ割込み(ETI)および分配割込み(DI)で処理されます。図3-1はどのように各割込み要求が処理されるかを説明しています。

図3-1 割込み処理ロジック



DIとETIは割込みを発生させるために実装および有効化する必要があります。各割込みは個別に実装/解除および有効化/無効化することが可能です。電源投入時の初期化後、全ての割込みは解除かつ無効化されています。

割込みが実装された場合、割込み要求を要求ビットに設定します。割込みが解除された場合、どの未処理の要求もオフとなり無視されます。

有効な割込みの要求は割込み優先度解決ロジックに参加する事を可能とします。有効ビットは受け付ける割込み要求を配信するためにRCIMボードの許可の承認について判断します。ETIまたはDIが無効である場合、割込み要求は受け付けますが、再度有効となるまで割込みの配信は遅れます。

割込みブロックへの入力DIを駆動するために使用されます。この出力はRCIMボードが取り付けられているホスト・コンピュータへ、構成することで更にRCIMチェーン内の他のシステムへ届けられます。ホストのETIまたはDIの割込みハンドラーが保留中ビットを解除する度に割込みの処理を決定します。これはRCIMボードがこの割込みの次のインスタンスを出力する方法をクリアします。

DIとETIの実装と有効化

DIおよびETIは次の命令と共に*ioctl(2)*システム・コールを使い実装しますが、これらは互換的に使用することが可能です：

```
DISTRIB_INTR_ARM
ETI_ARM
```

DIおよびETIは次の命令と共に*ioctl(2)*を使い有効化しますが、これらは互換的に使用することが可能です：

```
DISTRIB_INTR_ENABLE
ETI_ENABLE
```

割込み認識ロジック

デフォルトで、RCIMは割込みを発生させる入力シグナルの先頭のエッジを待ち受けます。割込みを認識するとすぐ、新しい割込み要求を誘発するためにデアサートおよび再アサートされる必要があります。任意の方法で、割込みをレベル・センシティブとして構成することが可能です。このモードでは、割込みは割込みシグナルがハイまたはローの時に発生します。構成オプションは本章で後述する各割込みのタイプに関する資料に含まれています。

RCIMが割込みシグナルから確実に割込みを取り出すには、シグナルを生成する設備がリセット値に変化する少なくとも1.5マイクロ秒前に生成された値を保持する必要があります。

分配割込みの設定

RCIMは、RCIMチェーンを使って相互接続されたシステム全体で割込みを共有する能力を提供します。分配割込みは3-20ページで詳細が取り上げられていますが、下図でどのように操作するかを図で提供します。分配割込みを設定するためのガイドラインは次の図に基づいています。

図3-2 分配割込み操作の例

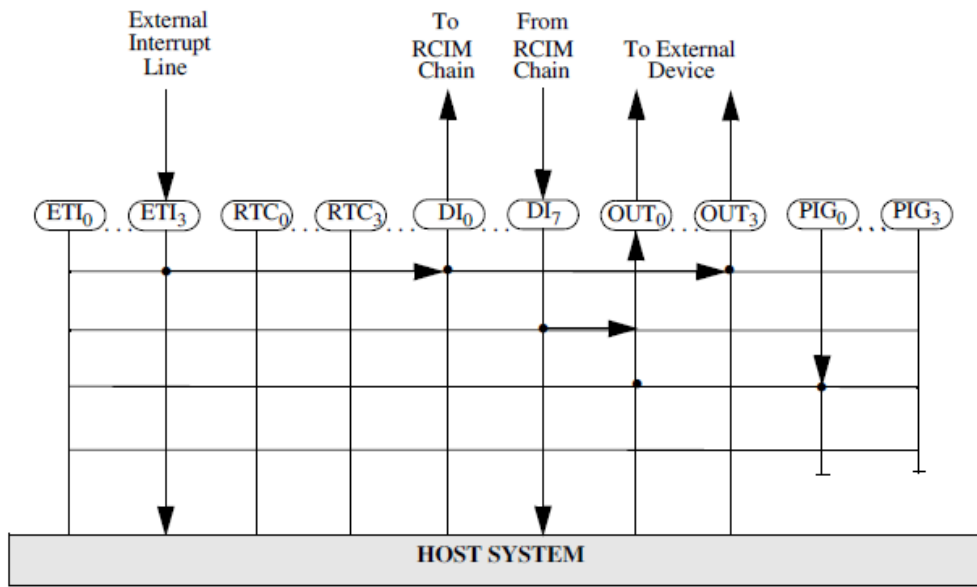


図3-2には3つのシナリオがあります：

- DI₀およびOUT₃で割込みを駆動するシグナルがETI₃で生成されます。その割込みはホスト・システムにも渡されます。
- 割込みをRCIMチェーンのDI₇から受信します。この割込みはホスト・システムに送信され、またOUT₀を介して外部デバイスへ送信します。
- 割込みはOUT₀を介して外部デバイスへ渡されるPIG₀で生成されます。

ローカルの割込みは構成されたDIを駆動しないことに注意して下さい。ioctlのETI_REQUESTはローカル割込みを発生しますが、ETIに関連付けたDIには影響しません。例えば、ETI₀をDI₀が駆動するように構成する場合、ローカルでのETI₀割込み制御ロジックを通らずに外部ETI入力をDI₀に直接接続します。このメリットはローカル割込みがPIGやETIが配信されるたびに発生しない事です。PIGは割込みを制御するプログラム可能なソフトウェア用に使用され、ETIは外部割込みの出力用に使用されることになります。

この例では以下に注意して下さい：

- 1つ以上の割込みモジュールが同じシグナル・ラインを駆動しようとした時に問題が発生する可能性があります。この例では、ETI₃シグナルがDI₀とOUT₃の割込みを駆動しています。

これはDI₀とOUT₃を同時に割込みを駆動するために(RTC₀といった)他のシグナル処理モジュールを構成することが可能です。この状況では、ラインを駆動するシグナルは強力な増幅器となります。この状況を回避するのは管理者次第です。

- どの方向から配信された割込みを受け取るのかを判断して下さい。これは配信されたデバイスのあるシステム上において、2つの割込みを生成する可能性があることを意味します：自身のローカル・デバイス割込み(ETI, PIG, RTC)と分配割込み。これらはそれぞれ個々の割込みベクターがあります。

両方の割込みを受信する事が望ましいのかもしれませんが、通常1つだけで差し支えありません。分配割込みを解除する事により、それがローカル・システムで生成されることからその割込みを避ける事が可能となります。デフォルトで、分配割込みは解除された状態です。

RCIM値の取得

RCIMの値を表示するまたは取得するために利用可能な方法がいくつかあります：

ファイルシステム/**proc/driver/rcim:N** (Nはゼロから開始):

このファイルシステム内の次のファイルは表示する事が可能です(特に断りがない限り読み取り専用)：

config – カット&ペーストに適した様式のRCIM構成(read/write)
interrupts – CPU単位での全てのETI, DI, RTCの割込み回数および合計
status – 様々なRCIMボードのステータスおよび時刻同期
rawregs – 読み取り可能なRCIMボードの全てのレジスタの名前付き16進表示
rtc – RTCのステータス(実行状況、カウント値等)
eti – ETIのステータス(実装済み、有効化等)
di – DIラインのステータス(実装済み、有効化等)

ioctl(2)システム・コール：

特定の割込みタイプに関する情報は、プログラムのアドレス空間に適切にマッピング(**mmap**)されたデバイス・ファイルを使い適切な操作を指定することで取り出すことが可能となります(例：**/dev/rcim:0/eti1**でのETI_INFO)。 **rcim_eti(4)**, **rcim_rtc(4)**, **rcim_distrib_intr(4)**のmanページを参照して下さい。

mmapされたデバイス・ファイル**/dev/rcim:0/rcim**でのRCIM_GET_INFO命令は**/proc/driver/rcim:0/config**と同じ情報を提供します。

mmapされたデバイス・ファイル**/dev/rcim:0/rcim**でのRCIM_GET_ADDR命令はRCIM制御レジスタの仮想および物理アドレスを提供します。

ヘッダ・ファイル**/usr/include/rcim.h**はRCIM_GET_INFOおよびRCIM_GET_ADDRで返される情報のレイアウトを記述しています。**rcim(4)**のmanページを参照して下さい。

mmap(2)システム・コール：

mmapはRCIMボードのデバイス・レジスタの一部または全てをマッピングするために使用することが可能です。このレジスタのレイアウトは**/usr/include/linux/rcim_ctl.h**および本ガイドの付録Bの中にあります。

エッジ・トリガ割込み

各RCIMボードは、これらの最も一般的な動作モードにちなんで名付けられたETIまたはエッジ・トリガ割込みと呼ばれる入力的外部割込みラインを持っています。これらのラインは所有する割込みソースをユーザーに提供する事を許可します。RCIMはこれらの割込みを処理してホスト・システムに配信し、それらが配信された場合はチェーン内の他の全てのRCIMに分配割込みとしてルートを決めて配信します。RCIMは12個のETI(0-11)をサポートします。

各々のETIは単独で構成可能です。ETIはエッジまたはレベル検出割込みとして入力シグナルを扱う事が可能です。エッジ検出の場合、立上りまたは立下りエッジのどちらかで割込みを発生させる事が可能です。レベル検出の場合、ハイまたはローのシグナル値のどちらかに対して割込みを発生させる事が可能です。入力シグナルのパターンを割込み要求に変換する方法を指定するには、「入力構成」に記述されているETI構成オプションの1つを使用して下さい。

これらのアプリケーションの要求に応じて実行中の各システムの各ETIをアプリケーションが順に実装/解除、有効化/無効化します。

RCIMが外部シグナルを生成する機材に課す要件の1つは、出力するシグナルは次の状態に変更する前に少なくとも1マイクロ秒間はローまたはハイのシグナル値を保持する必要があるという事です。より短い幅のパルスは割込みコントローラに認識されない可能性があります。パルスが1マイクロ秒よりも長い限りは、パルスの幅は重要ではありません。

rcim_eti(4)のmanページはETIに関する情報一式を提供します。

入力構成

各入力(ETI)は、**input**構成オプションを使ってシグナルの立上りまたは立下り、ハイまたはローのシグナル値で発生させるように構成することが可能です。本オプションは次の構文となります：

inputN / [rising | falling | high | low]

入力に関するデフォルト設定は「立下り」です。

フラグ・ワード(**rising**, **falling**, **high**, **low**)は単語の最初の文字を使って指定することが可能です。これらの単語は大文字と小文字の区別はしません。

下位互換性のため、元の名前**eti**は引き続き対応しています。

例として：

input0/falling	入力0を入力シグナルの立下りエッジで発生させるよう設定
input1/r	入力1を入力シグナルの立上りエッジで発生させるよう設定
input2/h	入力2をハイのシグナル値で発生させるよう設定

構成オプションの指定で利用可能な様々な方法については、2章の「構成」項または**rcim(4)**のmanページを参照して下さい

デフォルトでRCIM IVの外部入出力信号は**正論理(High True)**ですが、RCIM IIIの外部入出力信号は**負論理(Low True)**です。必要に応じて極性を反転させる制御レジスタが両方に存在します。外部割込み入力信号はレベル割込みに関しては極性制御、エッジ割込みに関しては立下り/立上りエッジ制御を持っています。

ETIデバイス・ファイル

各ETIは各々の専用のデバイス・ファイルから情報を得ます：

/dev/rcim:NetiM

*N*は(ゼロから開始される)RCIMカードの番号、*M*はETIのIDとなります。

これらのファイルは**/etc/init.d/rcim**初期化スクリプトによってシステム起動時に自動で生成されます。

ETIユーザー・インターフェース

ETIは**open(2)**, **close(2)**, **ioctl(2)**システム・コールで制御されます。本デバイスは**read(2)**, **write(2)**, **mmap(2)**システム・コールはサポートしない事に注意して下さい。

open コールは1つのエッジ・トリガ割り込みへのファイル記述子を割り当てて割り込みが現在他のデバイス・ドライバに使用されていないことを確認します。1つのデバイス・ファイルが各エッジ・トリガ割り込み対して存在します。**close** コールはファイル識別子を開放し付随するシグナルを取り除きます。詳細についてはmanページを参照して下さい。

ioctlへの以下の命令はETIを操作するために使用されます。これらの命令はDIに適用する事も可能です。全ての**ioctl**コールは**/usr/include/rcim.h**に定義された定数を使用します。詳細は**rcim_eti(4)**のmanページを参照して下さい。

ETI_ARM	ETIを実装します
ETI_DISARM	ETIを解除します
ETI_ENABLE	ETIを有効にします
ETI_DISABLE	ETIを無効にします
ETI_REQUEST	ソフトウェア要求割り込みを生成します
ETI_INFO	ETIに関する情報を取得します
ETI_WAIT	プロセスをスリープ状態にします
ETI_WAKEUP	全てのスリープ状態のプロセスを起こします
ETI_GETICNT	このETIが発した回数を返します
ETI_KEEPAIVE	キープアライブの状態を設定または解放します
ETI_VECTOR	ETIに関連する割り込みベクターを取得します
IOCTLGETICNT	このETIが発した回数を返します(汎用)
IOCTLKEEPAIVE	キープアライブの状態を設定または解放します(汎用)
IOCTLVECNUM	ETIに関連する割り込みベクターを設定します(汎用)
IOCTLSIGATTACH	RCIMデバイスが割り込みを生成した時にシグナルを要求します

DI同様、ETIは割り込みが受信可能となる前に実装、有効化されている必要があることに注意して下さい。

ETIの配信

RCIM上のいずれかまたは全てのETIをRCIMチェーンで接続された全てのシステムに配信する事が可能です。配信されたETIのソースはチェーン内のいずれのRCIMに定める事が可能です。

指定されたETIが自身の割込みを接続されたシステム全てに送信させるかどうかを判断するには、3-11ページの「RCIM値の取得」項で説明された方法の1つを利用して下さい。分配割込みの設定に関する詳細については3-20ページの「分配割込み」項を参照して下さい。

リアルタイム・クロック (RTC)

RCIMはリアルタイム・クロック・タイマーを提供します。各々のそれらのカウンターは専用のファイルを使ってアクセス可能で、それぞれがいずれのタイミングまたは周波数を制御する機能の殆どで使用することが可能です。8個の32bit RTC(0-7)をサポートします。

リアルタイム・クロック・タイマーは(クロック・カウント値を組み合わせた場合に)様々なタイミング間隔を提供するいくつかの異なる分解能にプログラム可能です。これは与えられた周波数(例えば600Hz)で実行するプロセスまたはタイミング・コード・セグメントにとって理想となります。タイマーはワン・ショットまたは周期的とすることが可能です(周期的の場合、最初にロードした値はゼロに到達する度に自動でカウンターに再ロードされます)。

ホスト・システムで割込みを生成できる事に加え、RCIMリアルタイム・カウンターの出力は対応するホスト・システムへ配信するため、またはRCIMの外部出力割込みラインの1つに取り付けた外部機材へ配信するために他のRCIMボードへ配信する事が可能です。

rcim_rtc(4)のmanページはRTCに関する詳細な情報を提供します。

RTCデバイス・ファイル

各RTCはそれぞれの専用のデバイス・ファイルを介してアクセスされます：

/dev/rcim:NrtcM

Nは(ゼロから開始される)RCIMカードの番号、MはRTCのIDとなります。

これらのファイルは**/etc/init.d/rcim**初期化スクリプトによってシステム起動時に自動で生成されます。

RTCの配信

RCIM上のいずれかまたは全てのRTCをRCIMチェーンで接続された全てのシステムに配信する事が可能です。配信されたRTCのソースはチェーン内のいずれのRCIMに定める事が可能です。

指定されたRTCが自身の割込みを接続されたシステム全てに送信させるかどうかを判断するには、本章で前述した「RCIM値の取得」項で説明された方法の1つを利用して下さい。分配割込みの設定に関する詳細については3-20ページの「分配割込み」項を参照して下さい。

RTCユーザー・インターフェース

リアルタイム・クロック・タイマーは**open(2)**, **close(2)**, **ioctl(2)**システム・コールで制御されます。**close(2)**システム・コールは、(最後にオープンしたデバイスをクローズする場合に)**IOCTLKEEPALIVE ioctl**命令を**close**の前にRTCに対して発行していない限り、RTCを停止しその設定を消去します。

ioctlを介して渡したパラメータはリアルタイム・クロック・タイマーのモード、クロック・カウント値、集計値を制御し、加えてRTCの現在の設定を取得します。本デバイスは**read(2)**, **write(2)**システム・コールはサポートしません。

ioctlへの以下の命令はRTCを操作するために使用されます。全ての**ioctl**コールは**/usr/include/rcim.h**に定義された定数を使用します。詳細は**rcim_rtc(4)**のmanページを参照して下さい。

RTCIOCSETL	RTC値(32bitインターフェース)を初期化します
RTCIOCGETL	RTC値(32bitインターフェース)を取得します
RTCIOCSET	RTC値(16bitインターフェース)を初期化します
RTCIOCGET	RTC値(16bitインターフェース)を取得します
RTCIOCSETCNT	RTCクロック・カウントを設定します
RTCIOCMODCNT	RTCクロック・カウントを変更します
RTCIOCGETCNT	RTCクロック・カウントを取得します
RTCIOCRES	RTCクロックの分解能を取得します
RTCIOCSTART	RTCの集計を開始します
RTCIOCSTOP	RTCの集計を停止します
RTCIOCWAIT	RTCクロック・カウントがゼロに達するまでブロックします
RTCIOCWAKEUP	全てのスリープ状態のプロセスを起こします
RTCIOCINFO	RTCに関する情報を取得します
IOCTLGETICNT	このクロックが発した回数を返します
IOCTLVECNUM	RTCに関する割込みベクターを設定します
IOCTLKEEPALIVE	最後のクローズでタイマーを破棄しません
IOTLSIGATTACH	RTCが割込みを生成した時にシグナルを要求します

外部出力割込み

各RCIMは外部出力シグナルを提供します。これらのシグナルは他の計算機への割込みソースとして使用する、または外部デバイスを制御するためのシグナルとして使用することが可能です。RCIMは12個の外部出力割込み(0-11)をサポートします。

外部出力割込みはRCIMの内部的ないくつかのソースのいずれかから駆動させる事が可能です。最も一般的なソースはプログラマブル割込み発生器(PIG)です。PIGは出力シグナルの生成に関して完全なソフトウェア制御を提供します。

外部割込みコネクタ用のピン配列については2章で説明しています。

出力ソース構成

各外部出力ラインは以下の構成オプションを使って指定したソースで駆動させるように構成することが可能です：

`<source> | outN`

ソースに指定する値は次のいずれかにする事が可能です：

rtcN	リアルタイム・クロック・タイマーを使い出力を駆動
pigN	プログラマブル割込み発生器を使い出力を駆動
inputN	エッジ・トリガ割込みを使い出力を駆動
etiN	入力(ETI)の別名
diN	分配割込み
gps	GPS PPS信号を使い出力を駆動
irig	IRIG PPS信号を使い出力を駆動
dcls_out	IRIG信号を使い出力を駆動
10mhz	10MHzクロックを使い出力を駆動
mclock	マスター時刻信号を使い出力を駆動
none	割り込み出力ラインは浮いた状態

例として：

<code>rtc3 out0</code>	出力ライン0をリアルタイム・クロック3で駆動するよう設定
<code>di5 out2</code>	出力ライン2を分配割込み5で駆動するよう設定

本構成オプションのデフォルトは対応する出力ライン用のソースと同じPIGです。例：

`pig0|out0`
`pig1|out1`
 等

NOTE

RCIM IVではピンの出力を駆動するために同じピンを入力ソースとして指定することは出来ません。例えば、”**input0|out0**”を設定しようとするとなんとも無効となって「Invalid Argument」エラーとなります。

構成オプションの指定で利用可能な様々な方法については、2章の「構成」項または**rcim(4)**のmanページを参照して下さい。

RCIM IVにおいて各EXT_INOUTピンは、ソースを使って駆動する前に最初に出力ラインとして構成する必要があります。詳細については3-23ページの「RCIM IVのピン構成」を参照して下さい。

デフォルトでRCIM IVの外部入出力信号は**正論理(High True)**ですが、RCIM IIIの外部入出力信号は**負論理(Low True)**です。必要に応じて極性を反転させる制御レジスタが両方に存在します。

MSI-X割込み構成

RCIM IVはMSI-X規格を完全にサポートしており、互換性のあるハードウェア上で最大16個の異なる割り込みをソースに加えることが可能です。各MSI-X割込みは特有のユニークなcoreアフィニティを施すことが可能で、RCIMソースからの干渉を排除し、現在のハードウェア上で可能な限り最小の割込み遅延時間で配信します。

NOTE

RedHawkの古いバージョンとの下位互換性のため、デフォルトで1つのMSI-X割込みベクターだけが有効化(`irq0`)され、全てのMSI-X割込みで使用されます。

追加のMSI-X割込みベクターを有効化するには、`rcim.max_msix_irqs=N`起動オプションを付けてRedHawkカーネルを起動して下さい。*N*には1～16の任意の整数を設定することが可能です。

全てのMSI-Xサポートを完全に無効化するには、`rcim.nomsix`起動オプションを付けてRedHawkカーネルを起動して下さい。

MSI-Xソース構成

各MSI-X割込みは以下の構成オプションを使って指定したソースに接続させるように構成することが可能です：

`<source> | irqN`

ソースに指定する値は次のいずれかにする事が可能です：

rtcN	リアルタイム・クロックを指定したIRQに接続
inputN	入力を指定したIRQに接続
etiN	入力(ETI)の別名
diN	分配割込みを指定したIRQに接続
gps	GPS PPS信号を指定したIRQに接続
irig	IRIG PPS信号を指定したIRQに接続
none	指定したIRQに関連付けられたソースがないことを確保

例として：

<code>rtc3 irq3</code>	リアルタイム・クロック3の割込みをIRQ3に接続
<code>input8 irq4</code>	入力8の割込みをIRQ4に接続

MSI-X IRQソースのデフォルトは次のように構成されています：

```
rtc0|irq1
rtc1|irq2
rtc2|irq3
rtc3|irq4
rtc4|irq5
rtc5|irq6
rtc6|irq7
```

```

rtc7|irq8
input6|irq9
input7|irq10
input8|irq11
input9|irq12
input10|irq13
input11|irq14
di0|irq15

```

他のIRQに明示的に接続されていない全てのソース割り込みを提供する多目的IRQであるため、上記ではirq0に言及していないことに注意して下さい。

別のIRQだった以前の接続を削除するためにソースを明示的にirq0に接続することは可能です。加えて、構成変更でソースが明示的に非ゼロのIRQに接続されなくなるといつでもソースは自動的にirq0に接続されます。

MSI-Xの互換性

RCIMの割り込みに関する情報を得るために**/proc/interrupts**ファイルの中身进行处理する既存のシェル・スクリプトやプログラムは、複数のMSI-X割り込みベクターが割り当てられた場合に現れる追加の割り込みライン进行处理するよう修正する必要がある可能性があります。

デフォルトで全てのMSI-X割り込みに割り込みベクター1つのみが割り当てられ、**/proc/interrupts**ファイル内のRCIMの1行は以前のリリースから変更されません。例えば、RCIMの行はクアド・コアのシステムでは次の行のようになります：

```
242: 0 0 0 0 IR-PCI-MSI 49283072-edge rcim
```

一方、MSI-Xが複数の割り込みベクターで構成された場合(例: `rcim.max_msix_irqs=16`を付けてカーネルを起動した場合)、**/proc/interrupts**ファイルの中身は著しく異なることとなり、各割り込みベクターに対して1行となります。例えば、RCIMの行はクアド・コアのシステムでは次の行のようになります：

```

242: 0 0 0 0 IR-PCI-MSI 49283072-edge rcim-irq0
249: 0 0 0 0 IR-PCI-MSI 49283073-edge rcim-irq1
250: 0 0 0 0 IR-PCI-MSI 49283074-edge rcim-irq2
251: 0 0 0 0 IR-PCI-MSI 49283075-edge rcim-irq3
252: 0 0 0 0 IR-PCI-MSI 49283076-edge rcim-irq4
253: 0 0 0 0 IR-PCI-MSI 49283077-edge rcim-irq5
254: 0 0 0 0 IR-PCI-MSI 49283078-edge rcim-irq6
255: 0 0 0 0 IR-PCI-MSI 49283079-edge rcim-irq7
256: 0 0 0 0 IR-PCI-MSI 49283080-edge rcim-irq8
257: 0 0 0 0 IR-PCI-MSI 49283081-edge rcim-irq9
258: 0 0 0 0 IR-PCI-MSI 49283082-edge rcim-irq10
259: 0 0 0 0 IR-PCI-MSI 49283083-edge rcim-irq11
260: 0 0 0 0 IR-PCI-MSI 49283084-edge rcim-irq12
261: 0 0 0 0 IR-PCI-MSI 49283085-edge rcim-irq13
262: 0 0 0 0 IR-PCI-MSI 49283086-edge rcim-irq14
263: 0 0 0 0 IR-PCI-MSI 49283087-edge rcim-irq15

```

NOTE

前述の例は1枚のRCIMカードだけが組み込まれているシステムの典型です。複数のRCIMカードが存在する場合、割込みの名前は**RCIM:R-irqN**として表示され、**R**は(0から始まる)RCIMカードの連続番号、**N**は接続されたMSI-X IRQのベクトル番号となります。

プログラマブル割込み発生器 (PIG)

各RCIMはプログラマブル割込み発生器(PIG)を提供します。PIGは外部出力シグナルの出力を制御するソフトウェア提供するために通常は使用します。更に、PIGは分配割込みへのシグナルを駆動するため、許可されたユーザー・ソフトウェアがRCIMチェーン内の全てのRCIMに同時に配信する分配割込みを生成するために使用することが可能です。RCIM IIIとRCIM IIは12個のPIG(0-11)をサポートし、RCIM Iは4個のPIG(0-3)をサポートします。

rcim_pig(4)のmanページはPIGに関する詳細な情報を提供します。

デフォルトでRCIM IVの外部入出力信号は**正論理(High True)**ですが、RCIM IIIの外部入出力信号は**負論理(Low True)**です。必要に応じて極性を反転させる制御レジスタが両方に存在します。プログラマブル割込み発生器(PIG)出力は任意の状態にセットまたはリセットすることが可能です。

PIGデバイス・ファイル

デバイス・ファイル**/dev/rcim:N/pig**はローカル・システムでPIGレジスタにアクセスするために使用されます。このファイルは**mmap(2)**を使ってプログラムのアドレス空間にマッピングする必要があります。デフォルトで、**root**の特権を持つユーザーだけがそれを行う権利を持っています。

RCIM IIIとRCIM IVでは、PIGレジスタは12bit幅、各PIGに対して1bitです。2つの追加レジスタはマルチプロセッサの安全な方法でPIGのビットを設定および消去することを許可します。**mmap**されたPIGレジスタのページにおいて、レジスタ設定(PIGS)はオフセット0x10およびレジスタ消去(PIGC)はオフセット0x20となります。PIGを設定するとPIGが接続されている方法次第で分配割込みまたは外部出力を生成します。信号の必要な長さは取り付けられたデバイスの要件に依存します。

シグナルがRCIM IVに流れている場合、ローまたはハイの値を次の状態に変わる前に少なくとも1マイクロ秒間保持する必要があります。RCIM IIIに関しては保持される持続時間は1.5マイクロ秒である必要があります。

PIGの配信

RCIM上のPIGのいくつかまたは全てをRCIMチェーンで接続された全てのシステムに配信する事が可能です。配信するPIGのソースはチェーン内のいずれのRCIMに配置させる事が可能です。

指定したPIGがその割込みを接続された全てのシステムに送信させるかどうかを決定するには、本章で前述している「RCIM値の取得」項で説明されている方法の1つを利用して下さい。分配割込みの設定に関する情報については次項「分配割込み」を参照して下さい。

分配割込み

RCIMの本当の核心と能力は分配割込みシステムの中にあります。各RCIMはRCIMチェーンを介して接続された全てのシステムに同時に割込みを配信する事が可能です。RCIMは12個の分配割込み(0-11)をサポートします。本機能の図および分配割込みの設定に関するガイドラインは本章で前述している「分配割込みの設定」項で見る事が可能です。

チェーン内のいずれのRCIMボードでエッジ・トリガ割込み、リアルタイム・クロック・タイマー、GPS PPS(Pulse Per Second)、プログラマブル割込み発生器のいずれかを配信するために構成することが可能です。割当てられたデバイス・ファイルは各々の分配割込みに関連付けられています。

RCIMの分配割込みは、分配割込みを配信するもしくは受信するのいずれかにすることをRCIMが取り付けられた各システムで構成する必要があります。分配割込みは分離したシステムで構成しローカルで使用する事も可能です。構成の詳細は後述の「DI構成」項にあります。構成情報を得るために利用可能な方法については3-11ページの「RCIM値の取得」項を参照して下さい。

rcim_distrib_intr(4)のmanページはDIに関する詳細な情報を提供します。

DI構成

システムに接続された全てのRCIMはRCIMの分配割込みラインに関して互換性のある構成を有している事が重要です。

デフォルトで、分配割込みは構成されていません。

分配割込みは最初にソースを与える必要があり、次にシグナルの立上りまたは立下りエッジ、またはハイかローのシグナル値で発生させるように構成します。

分配割込みのためのソースを定義するには、以下の構成オプションを使用して下さい：

<source> | diN

ソースに対して指定する値は以下のいずれかが可能です：

rtcN	リアルタイム・クロック・タイマー
pigN	プログラマブル割込み発生器
inputN	エッジ・トリガ割込み
etiN	エッジ・トリガ割込みの別名
gps	GPS信号
irig	IRIG信号
none	RCIMはこの分配割込みを駆動しない

例として：

rtc3 di6	分配割込み6がリアルタイム・クロック3で駆動されるように設定
pig1 di3	分配割込み3をプログラマブル割込み発生器1で駆動するように設定
none di0	RCIMは分配割込み0を駆動しない

各分配割込みを**di**構成オプションを使いシグナルの立上りまたは立下りエッジ、またはハイカローのシグナル値で発生させるよう構成させることが可能です。本オプションは以下の構文を有します：

diN / [rising | falling | high | low]

フラグ・ワード(**rising**, **falling**, **high**, **low**)は単語の最初の文字を使って指定することが可能です。これらの単語は大文字と小文字の区別はしません。

例として：

di0/falling	分配割込み0をその入力割込みの立下りエッジで発生させるよう設定
di1/r	分配割込み1をその入力割込みの立上りエッジで発生させるよう設定

構成オプションの指定で利用可能な様々な方法については、2章の「構成」項または**rcim(4)**のmanページを参照して下さい。

DIデバイス・ファイル

各分配割込みはそれぞれの専用のデバイス・ファイルを介してアクセスされます：

/dev/rcim:NdiM

Nは(ゼロから開始される)RCIMカードの番号、**M**は分配割込みのIDとなります。

これらのファイルは**/etc/init.d/rcim**初期化スクリプトによってシステム起動時に自動で生成されます。

DIユーザー・インターフェース

分配割込みは**open(2)**, **close(2)**, **ioctl(2)**システム・コールで制御されます。本デバイスは**read(2)**, **write(2)**, **mmap(2)**システム・コールはサポートしないことに注意して下さい。

openコールは1つの分配割込みに1つのファイル記述子を割当てます。**close**コールはファイル記述子を解放し、(それが最後のcloseである場合) **IOCTLKEEPALIVE**ステータスが解放されていれば割込みを解除します。

ioctlへの以下の命令は分配割込みを操作するために使用されます。これらの命令はETIに適用する事も可能です。全ての**ioctl**コールは**/usr/include/rcim.h**に定義された定数を使用します。詳細は**rcim_distrib_intr(4)**のmanページを参照して下さい。

DISTRIB_INTR_ARM	DIを実装します
DISTRIB_INTR_DISARM	DIを解除します
DISTRIB_INTR_ENABLE	DIを有効にします
DISTRIB_INTR_DISABLE	DIを無効にします
DISTRIB_INTR_REQUEST	ソフトウェア要求割込みを生成します
DISTRIB_INTR_INFO	DIに関する情報を取得します

DISTRIB_INTR_WAIT	次のDIまでスリープ状態にします
DISTRIB_INTR_WAKEUP	全てのスリープ状態のプロセスを起こします
DISTRIB_INTR_KEEPAIVE	キープアライブの状態を設定または解放します
DISTRIB_INTR_GETICNT	このDIが発した回数を返します
DISTRIB_INTR_VECTOR	DIに関する割込みベクターを取得します
IOCTLKEEPAIVE	キープアライブの状態を設定または解放します(汎用)
IOCTLGETICNT	このDIが発した回数を返します(汎用)
IOCTLVECNUM	DIに関する割込みベクターを設定します(汎用)
IOCTLSIGATTACH	RCIMデバイスが割込みを生成した時にシグナルを要求します

ETI同様、分配割込みは割込みが受信可能となる前に実装、有効化されている必要があることに注意して下さい。

マスタークロックの無効化

デフォルトで全てのRCIMカードはRCIMドライバーに検出され初期化された時に自動的にマスタークロックとして登録され、これはRCIMの時刻がシステム時刻の基準として使用されることを意味します。マスタークロックの動作は次の構成構文を使って有効化および無効化することが可能です。

[**clock** | **noclock**]

特定の条件化においてはRCIMの既定のマスタークロック動作を無効にすることが不可欠となる可能性があります。例えば、システム時刻の基準として使用可能なサード・パーティのGPSカードが要求される場合です。

詳細については3-5ページの「RCIMマスタークロックの留意事項」および**masterclock(5)**のmanページを参照して下さい。

外部クロックの入力

RCIMは外部の10MHzクロック信号で駆動することが可能です。RTC、ティック・クロック、POSIXクロック用に外部クロック信号はRCIMのプライマリ・クロック・ソースとして使用されます。この外部クロック・ソースはRCIM内部のクリスタルと置き換わります。

RCIM IIIにおいては、外部クロック入力信号は専用のEXT_CLKIN入力ピン(2-11ページの「外部割込みI/Oコネクタ」を参照)に接続する必要があります。

RCIM IVにおいては、外部クロック入力信号は次の構成構文を使って外部クロック入力信号を受信するように構成した任意のEXT_INOUTピン(2-5ページの「外部割込みI/Oコネクタ」を参照)に接続することが可能です。

pinN | ext_clock_in

下位互換性のため、**mclock_in**は**ext_clock_in**のエイリアスとして受け入れられます。

次項で説明しているように外部クロック入力信号を受信するために使用される特定のピンも入力として構成される必要があることに注意して下さい。

RCIM IVのピン構成

RCIM IV固有の各EXT_INOUTピンは入力または出力、同様に終端または非終端のどちらかに構成することが可能です。

pinN / [out|in] / [non-terminated|terminated]

フラグ・ワード(out, in, non-terminated, terminated)は単語の最初の文字を使って指定することが可能です。これらの単語は大文字と小文字の区別はしません。

例として：

pin0/out/n	非終端の出力となるようEXT_INOUT0を構成
pin1/in/t	終端付きの入力となるようEXT_INOUT1を構成
pin2/i/terminated	終端付きの入力となるようEXT_INOUT2を構成
pin3/o/non-terminated	非終端の出力となるようEXT_INOUT3を構成

デフォルト：EXT_INOUTピン0-5は出力、EXT_INOUTピン6-11は入力、全てのピンが非終端。

ピンはソースとして使用する前に入力として構成する必要があり、ピンはソースで駆動する前に出力として構成する必要があります。

NOTE

RCIM IVではピンの出力を駆動するために同じピンを入力ソースとして指定することは出来ません。例えば、”input0 | out0”を設定しようとするが無効となって「Invalid Argument」エラーとなります。

構成オプションの指定で利用可能な様々な方法については、2章の「構成」項または**rcim(4)**のmanページを参照して下さい。

GPSクロックの同期

本章はオプションのGPSモジュールを含むRCIMを使ってGPSクロック同期を構成する方法について解説します。

概要

オプションのGPSモジュールは、RCIMのPOSIXクロックをGPS時刻に同期するためにシステム・タイムキーピング・デーモンで利用することが可能です。本機能はチェーン内の全てのRCIMがRCIMマスターからのGPS時刻と同期することが可能です。

RCIMにオプションのGPSモジュールが含まれている場合、RCIMのPOSIXクロックをGPS時刻に同期するためシステム・タイムキーピング・デーモンはGPS受信機を使用するようにインストールおよび構成されている必要があります。RedHawkは2つの異なる同期方式をサポートします：

1. GPSDとChronydデーモンの利用
2. NTPDデーモンの利用

両方の方式について以降の主要な項で説明しています。

NOTE

RCIM IVのGPSモジュールは固定アンテナ位置から過剰決定(Overdetermined)クロックモードで動作します。モバイルでの利用が必要である場合、コンカレント日本の営業部(03-3864-5713)に連絡、またはWebサイト(<http://www.concurrent-rt.co.jp/company/>)にアクセスして下さい。

GPSDとChronyd

RedHawk Version 7.xと8.xはgpsdデーモンと一緒にchronydデーモンを利用するGPSクロック同期をサポートします。次項ではこれらのデーモンを使ってGPSクロックの同期を構成、実行、検証するための詳細について説明します。

NOTE

RCIMスレーブとして構成されたRCIMは光ケーブルを介して自動的にRCIMマスターとシステム時刻を同期します。この自動同期のためにRCIMスレーブはGPSタイミング・ソースに同期するよう構成することが出来ませんので、ChronydまたはNTPDを決して有効にしないで下さい。

gpsdの構成

RCIMクロックをGPS時刻に同期するために**gpsd**を構成するには次の手順に従って下さい：

1. システムに**gpsd rpm**がインストールされていることを確認して下さい：

```
# rpm -q gpsd
```

パッケージがインストールされていない場合、必要な**gpsd**パッケージを次の場所で見つけることが可能です。

- RedHawk Linux Installation Discの**gpsd**ディレクトリ内
- NUU更新ユーティリティを利用してConcurrentのオンライン・リポジトリ経由
- Extra Package for Enterprise Linux (EPEL)のオンライン・リポジトリ

NOTE

3.23以前の**gpsd**のバージョンは正確な時間管理に影響を及ぼす可能性のある重大なバグが含まれています。必ずバージョンが3.23以降の**gpsd**のバージョンを使用して下さい。適切なバージョンを探すのに苦労する場合は当社に連絡して下さい。

NOTE

Ubuntuがインストールされたシステムでは、**gpsd**パッケージは標準ビルトイン・ディストリビューション・リポジトリから入手可能です。Ubuntuでの**gpsd**のセットアップはここで説明しているのと同じ構成ファイルが必要です。

RedHawk Linux Installation Discを利用できる場合、単にディスクをマウントしてrootユーザーとしてディスク上の**gpsd**ディレクトリから次のコマンドを実行して下さい：

```
# rpm -Uhv ./*.rpm
```

インストールが完了したら、次の2項へスキップして下さい。

RedHawk Linux Installation Discを利用できない場合、次のURLにあるExtra Package for Enterprise Linux (EPEL)リポジトリの中でソフトウェアを見つけることが可能です：

<https://fedoraproject.org/wiki/EPEL>

まず、自動依存を使う**yum**を介してEPELパッケージを簡単にインストールできるようにRedHawkバージョンに対する**epel-release RPM**をダウンロードしてインストールして下さい。

また、ベース・ディストリビューション・リポジトリ(例えば、BaseOSやAppStream)の依存性が有効であることを確認して下さい。詳細については**yum(8)**のmanページを参照して下さい。

epel-release RPMがインストールされたら、**gpsd**デーモンとクライアントをインストールするため次のコマンドを実行して下さい：

```
# yum install gpsd gpsd-clients
```

2. システムの**/etc/sysconfig/gpsd**ファイルを編集して次の変数が以下に示すように設定されていることを確認して下さい：

```

OPTIONS="-n"
DEVICES="/dev/rcim_uart0"
BAUDRATE="9600"

```

これらの設定変数はRCIMのGPSクロック同期で必要となります。

NOTE

RedHawk 7.Xに関する**gpsd**のバージョンはデバイス名を指定するために**OPTIONS**変数のみが必要で、例えば、**OPTIONS="-n /dev/rcim_uart"**となる。

ファイル内に存在する他の設定変数は変更しないままとすることが可能です(例えば、USBAUTO)。

3. **gpsd**デーモンが有効化され実行されていることを確認して下さい。

systemdを利用するRedHawkの最新のバージョンでは：

gpsdの状態を確認するため次のコマンドを実行して下さい：

```

# systemctl | grep gpsd
gpsd.service
    loaded active running GPS (...) daemon
gpsd.socket
    loaded active running GPS (...) daemon socket

```

gpsdサービスが実行されていない場合、デーモンを有効化し開始するにはrootユーザーとして次のコマンドを実行して下さい：

```

# systemctl enable gpsd
# systemctl start gpsd

```

systemdがないRedHawkの古いバージョンでは：

gpsdの状態を確認するため次のコマンドを実行して下さい：

```

# service gpsd status
gpsd (pid 6031) is running...

```

gpsdサービスが実行されていない場合、デーモンを有効化し開始するにはrootユーザーとして次のコマンドを実行して下さい：

```

# chkconfig gpsd on
# service gpsd start
Starting gpsd: [ OK ]

```

/etc/sysconfig/gpsd構成ファイルの変更を行った後は、常に**gpsd**を再開またはシステムを再起動することを忘れないで下さい。

gpsdの構成は第一段階ですが、GPSクロック同期を実現する前に**chronyd**も構成する必要があります。

chronydの構成

RCIMクロック同期のために**chronyd**を構成するには次の手順に従って下さい：

1. システムに**chrony rpm**がインストールされていることを確認して下さい：

```
# rpm -q chrony
```

インストールされていない場合、システムに付属するベース・ディストリビューションのアップデートDVDからインストール、または次のコマンドを使いInternetを介してインストールすることが可能です：

```
# yum install chrony
```

2. システムの**/etc/chrony.conf**ファイルに次の行を追加して下さい：

```
refclock PPS /dev/pps0 lock NMEA refid RCIM
refclock SHM 0 refid NMEA noselect
```

これらの行はRCIMのGPSクロック同期で必要となります。

複数のGPS対応デバイスが存在する場合、GPSクロック同期で必要となるデバイスの**/dev/pps#**を選択して下さい。

NOTE

上記の**refclock**の2行は**/etc/chrony.conf**ファイルに存在する**refclock**の行のみである必要があり、コメント・アウトされた行のインスタンスは無視されます。加えて、システムはネットワーク・ベースの時間ソースを使って時刻を同期するのに他のNTPまたはPTPデーモンを実行すべきではありません。インストールされた**ntpd**や**ptpd**サービスが無効化され停止されていることを確認して下さい。

NOTE

前述の**refclock SHM**の行は殆どのシステムで動作しますが、一部のシステムはオフセットの追加が必要となる可能性があります。例えば、次の行は±400ミリ秒のオフセットが**chronyd**が正確に同期するために必要であることを指定しています：

```
refclock SHM 0 offset 0.4 refid NMEA noselect
```

chronydがお手持ちのシステムがデフォルトの**refclock SHM**行で同期しない場合、必要となる正確なオフセットを決定するための指示については4-5ページの「PPS時刻オフセットの計算」を参照して下さい。

3. **chronyd**デーモンが有効かつ実行されていることを確認して下さい。

NOTE

GPSクロック同期のための**chronyd**が正常に開始される前に**gpsd**を正しく構成する必要があります。詳細については4-2ページの「**gpsd**の構成」を参照して下さい。

systemdを利用するRedHawkの最新のバージョンでは：

chronydの状態を確認するため次のコマンドを実行して下さい：

```
# systemctl | grep chronyd
chronyd.service
    loaded active running NTP client/server
```

chronydサービスが実行されていない場合、デーモンを有効化し開始するにはrootユーザーとして次のコマンドを実行して下さい：

```
# systemctl enable chronyd
# systemctl start chronyd
```

systemdがないRedHawkの古いバージョンでは：

chronydの状態を確認するため次のコマンドを実行して下さい：

```
# service chronyd status
chronyd (pid 12473) is running...
```

chronydサービスが実行されていない場合、デーモンを有効化し開始するにはrootユーザーとして次のコマンドを実行して下さい：

```
# chkconfig chronyd on
# service chronyd start
```

```
Starting chronyd: [ OK ]
```

/etc/chrony.conf構成ファイルの変更を行った後は、常に**chronyd**を再開またはシステムを再起動することを忘れないで下さい。

システムの電源投入時、RCIMのGPS受信機がGPS衛星にロックし全てのデータを受信したら、正確な時刻が利用可能となります。初回のコールド・ブートでGPS同期プロセスが完了するのに最大15分必要となる可能性があります。RCIM IVのより最新のGPSモジュールおよび最新世代のRCIM IIIを使用時は通常1～2分以内で完了します。全てのケースにおいて、ウォーム・リブート後のGPS同期は1～2分以内で完了するはずです。

PPS時刻オフセットの計算

/etc/chrony.confのrefclock SHM構成の行は**chronyd**が同期する前に手動で構成されたPPS時刻オフセット補正が必要となる可能性があります。例えば、次の行は-300ミリ秒オフセット補正が現在のシステムで正確に同期するために**chronyd**が必要であることを指定しています：

```
refclock SHM 0 offset -0.3 refid NMEA noselect
```

パルス/秒(PPS)基準クロックはどのUTC秒が各パルスに一致するかを決定するために非PPSソースが常に必要であり、相関するには両方の時刻が±200ミリ秒以内であるため、オフセット補正が要求される可能性があります。非PPS時刻はRCIM GPS UART(NMEA)を介して提供され、様々なハードウェア要因が2つの時刻ソースを200ミリ秒以上相殺するため、オフセット補正を手動で指定する必要があります。

NOTE

古いGPSモジュール付きの古いRCIM IIIカードは常に基準クロックから1秒ずれた時間に同期する可能性があります。これらのカードにおいては、補正オフセットを/etc/chrony.confに手動で追加する必要があります。

必要なオフセット補正を決定するには、**refclock SHM**構成行に**offset**キーワードをセットする前に次のコマンドを実行して下さい。

```
# chronyc sources
```

通常、次のような**Last sample**の値が見れるはずです：

```
210 Number of sources = 2
MS Name/IP address    ...    Last sample
=====
#* RCIM                ...    -0ns[  +0ns] +/-  101ns
#? NMEA                ...    +77ms[ +77ms] +/- 1737us
```

上記のNMEA行にある**+77ms**のサンプルはPPSおよび非PPS時刻ソースが±200ミリ秒未満の相関関係にあり、RCIMに示されている*ステータスでわかるように**chronyd**がRCIM GPSに同期しています。

一方、一部のシステムでは次のように±200ミリ秒を超える**Last sample**の値が見られる可能性があります：

```
210 Number of sources = 2
MS Name/IP address    ...    Last sample
=====
#? RCIM                ...    -0ns[  +0ns] +/-  101ns
#? NMEA                ...    +574ms[ +574ms] +/- 1558us
```

上記のNMEA行にある**+574ms**のサンプルはPPSおよび非PPS時刻ソースが±200ミリ秒未満の相関関係がなく、**chronyd**が同期するのを妨げておりRCIMに示される?ステータスの原因になっています。この問題を解決するには、このシステムの**refclock SHM**行に**-500**ミリ秒のオフセット補正をセットして下さい：

```
refclock SHM 0 offset -0.5 refid NMEA noselect
```

続いて**chronyd**を再開するために次のコマンドを呼び出し、調整されたNMWAの最後のサンプル値が±200ミリ秒以内であることを確認するために再度検査して下さい：

```
# systemctl restart chronyd
# chronyc sources
```

NOTE

500ミリ秒以上のオフセットに関しては、表示された最後のサンプル値は時刻相関関係の誤差の真の方向を正確に表していない可能性があります。この場合、**chronyd**を同期させる正確な値を決定するために負のオフセット補正(-0.5)の後に正のオフセット補正(0.5)を追加する必要があります。

オフセットに関する詳細については次のURLでChrony FAQを参照して下さい：
https://chrony.tuxfamily.org/faq.html#_using_a_pps_reference_clock

chronycを使ったGPS機能の確認

GPSが正確なシステム時刻を生産していることを測定するには、下に示すように**chronyc(1)**のソース表示機能を使用して下さい。本例はコマンド・ライン・オプションを利用していますが、**chronyc**は対話形式で実行することも可能です。詳細についてはmanページを参照して下さい。

```
# chronyc sources
210 Number of sources = 3
MS Name/IP address      Stratum  Poll    Reach   LastRx  Last sample
=====
#* RCIM                  0        4      377     14      -15ns[ -16ns] +/- 294ns
#? NMEA                  0        4      377     11      -82ms[ -82ms] +/- 100ms
^ ntp1.sample.com        1        6      377     41      +12us[ +12us] +/- 1455us
```

出力はシステム時刻がどれくらい他の時刻ソースと差が生じているかを示します。これはGPS受信機と他の時刻サーバーを含んでいます。

Name/IP address列は時刻サーバーのホスト名称を示します。RCIM時刻サーバーはGPS受信機が最後にPPS信号を受信した時のPOSIX時刻のスナップショットを示します。

NMEA時刻サーバーは**chronyd**がUARTを介してGPSからPPSを読めた時の遅延時間を示します。他の時刻サーバーはローカルで定義されたか自動的にpool.ntp.orgで割り当てられたかのどちらかです。

M列はソースのモードを示しており、「^」は外部の時刻サーバーを示し、「=」はピアを示し、「#」はローカルで接続された基準クロックを示します。

S列はソースの状態を示しており、「*」は現在同期しているソースを示し、「+」は同期したソースと結合されているソースを示し、「-」は結合アルゴリズムを使って除外されているソースを示し、「?」は受け入れ可能でもアクセス可能でもないソースを示し、「x」および「~」の両方は信頼できないと見做されたソースを示します。

Stratum列は階層番号を示します。階層ゼロのシステムは信頼できるソースに直接接続する必要があります。

Poll列はどのくらい頻繁にこのサーバーがポーリングされているかを示し、間隔が2を底とする対数で表されています。例えば、4はポーリングが16秒間隔で発生していることを示し、6はポーリングが64秒間隔で発生していることを示します。

Reach列は8進数で出力されたソースの到達可能性レジスタを示します。377の値は最後の8回の送信に対して有効な応答を受信したことを示します。

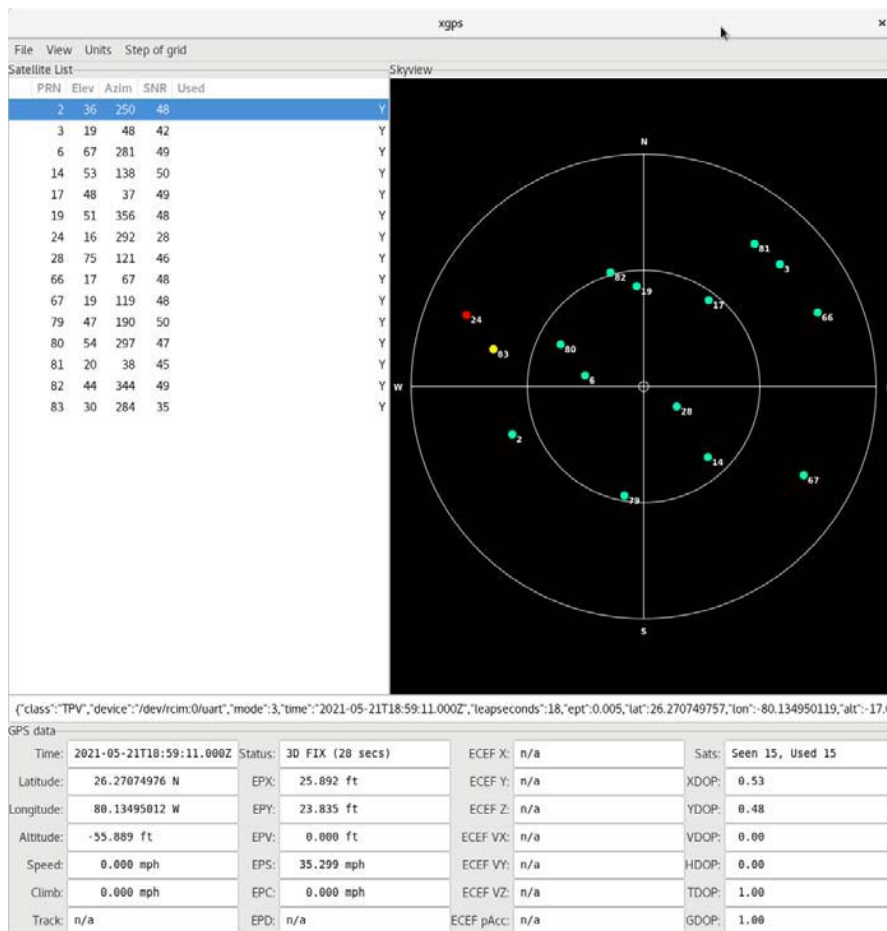
LastRx列はソースから最後の正常なサンプル(次の列に表示される)をどれくらい前に受信したかを示します。

Last sample列は最後の計測時のローカル・クロックとソース間のオフセットを示します。角括弧内の数字は実際に測定されたオフセットを示します。角括弧内の左側の数字は最初の測定を示しており、ローカル・クロックに適用されるslewを考慮して調整されます。+/-に続く数字は測定でのエラーのマーヅンを示します。正数のオフセットはローカル・クロックがソースより進んでいることを示します。

xgpsを使ったGPS衛星の表示

xgpsユーティリティは**gpsd-clients** RPMにより提供されます。**xgps**は現在のGPSの位置と時刻をアクセス可能な衛星の位置と一緒に表示します。

次のスクリーンショットは**xgps**表示の例を示します：



Skyview枠では、信号の品質を示すために衛星は色分けされており、明るい緑は高品質を示し濃い赤は低品質を示しています(dB単位の正確な数値については左側のデータ表示を参考にして下さい)。

円はGPS衛星を示します。塗りつぶされた円は直近のGPS計算中に衛星が使用されたことを示し、輪郭だけの円は使用されていない衛星を示します。

マウス・ポインターをシンボルの上に重ねると、Satellite List枠が提供する衛星の詳細を伴う小さなポップアップ・ウィンドウを一時的に表示します。

NTPD

RedHawkの以前のバージョンおよび最新のバージョンは、レガシーなntpdデーモンを利用するGPSクロック同期をサポートします。次項では構成、実行、ntpdを使ったGPSクロック同期の検証に関する詳細を提供します。

ntpdの構成

GPSクロック同期のためにntpdを構成するには次の手順に従って下さい：

1. システムにccur-ntpのrpmがインストールされていることを確認して下さい：

```
# rpm -q ccur-ntp
```

インストールされていない場合、RedHawk NTP w/RCIM GPSインストール・メディアから本パッケージをインストールするための手順についてRedHawk Linux Release Notesを参照して下さい。

2. ccur-ntpのrpmが提供する/etc/ntp.confファイルは、GPSを使用するために必要となる次の行を含んでいます：

```
server 127.127.8.0 mode 138 prefer #PARSE TSIP (10)+ PPS(128)
fudge 127.127.8.0 flag3 1 #enable PPS signal
```

適切なGPS同期のためにこれらの行の変更は必要ではありませんが、指定するパラメータに関する詳細は次のURLで入手することが可能です：

```
http://doc.ntp.org/4.1.2/clockopt.htm
http://doc.ntp.org/4.1.2/driver8.htm
```

次の3行は、時刻同期のために電源投入時にランダムに選択される世界中のサーバーの集まりを定義しています。本機能はデフォルトのNTP構成として動作し、GPSのバックアップとして役立ちます。最良の結果を得るにはこれらのエントリーの「pool」の前に自国のコードを含めると良いでしょう(例えば、0.us.pool.ntp.org)。詳細についてはwww.pool.ntp.orgを参照して下さい。

```
server 0.pool.ntp.org
server 1.pool.ntp.org
server 2.pool.ntp.org
```

「logfile」で始まるコメント・アウト・エントリのブロックは統計のロギングで使われるファイルを構成するために使用します。それらを有効にしたい場合は、エントリを非コメントとして下さい。

ログ・ファイルに加えて、ntpq(1)とntpd(1)はNTPを監視するために使用されます。NTPに関する詳細な情報はntpq(1)のmanページとwww.ntp.orgを参照して下さい。

3. ntpdデーモンが有効化され実行されていることを確認して下さい。

systemdを利用するRedHawkの最新のバージョンでは：

ntpdの状態を確認するため次のコマンドを実行して下さい：

```
# systemctl | grep ntpd
ntpd.service
loaded active running Network Time Service
```

ntpdサービスが実行されていない場合、デーモンを有効化し開始するにはrootユーザーとして次のコマンドを実行して下さい：

```
# systemctl enable ntpd
# systemctl start ntpd
```

systemdがないRedHawkの古いバージョンでは：

ntpdの状態を確認するため次のコマンドを実行して下さい：

```
# service ntpd status
ntpd (pid 8537) is running...
```

ntpdサービスが実行されていない場合、デーモンを有効化し開始するにはrootユーザーとして次のコマンドを実行して下さい：

```
# chkconfig ntpd on
# service ntpd start

Starting ntpd: [ OK ]
```

/etc/ntp.conf構成ファイルの変更を行った後は、常に**ntpd**を再開またはシステムを再起動することを忘れないで下さい。

システムの電源投入時、RCIMのGPS受信機がGPS衛星にロックし全てのデータを受信したら、正確な時刻が利用可能となります。初回のコールド・ブートでGPS同期プロセスが完了するのに最大15分必要となる可能性があります、RCIM IVのより最新のGPSモジュールおよび最新世代のRCIM IIIを使用時は通常1～2分以内で完了します。全てのケースにおいて、ウォーム・リブート後のGPS同期は1～2分以内で完了するはずです。

NOTE

gpsdと**chronyd**デーモンは**ntpd**よりも常に速くGPS同期を実現します。詳細については4-1ページの「GPSDとChronyd」を参照して下さい。

ntpqを使ったGPS機能の確認

GPSが正確なシステム時刻を生産していることを測定するには、下に示すように**ntpq(1)**のピア表示機能を使用して下さい。本例はコマンド・ライン・オプションを利用していますが、**ntpq**は対話形式で実行することも可能です。詳細についてはmanページを参照して下さい。

```
# /usr/sbin/ntpq -np
remote          refid                      st  t   when poll reach delay  offset jitter
=====
xns2.medbanner.c 192.43.244.18      2  u   33   64   377  72.443 -2.897  11.235
+toshi.keneli.or .GPS.              1  u   43   64   377  27.915  0.938   2.075
```

```

-216.56.81.86      193.131.101.50    3    u    33    64    377    49.388    -0.579    2.710
+new.localdomain  .GPS.                1    u    42    64    377    0.182    0.010    0.020
*GENERIC(0)       .GPS.                0    l    46    64    377    0.000    0.000    0.001

```

出力はシステム時刻がどれくらい他の時刻ソースと差が生じているかを示します。これはGPS受信機と他の時刻サーバーを含んでいます。

remoteのラベルの列は時刻サーバーのホスト名称です。システム**new.localdomain**はローカル・ネットワークの時刻サーバーで、**GENERIC(0)**はRCIMに取り付けられたGPSです。他の行は**pool.ntp.org**に割り当てられた時刻サーバーです。最初の列は同期のためにどのサーバーが選択されているかを示します。**GENERIC(0)**の前の「*」はRCIM GPS受信機がシステム・ピアとして使用されていることを示します。

delay, offset, jitter列は全てミリ秒単位の時間です。オフセットはローカル・システム時刻と時刻ソース間の差です。この場合はマイクロ秒の精度でGPS受信機に同期しています。

delay領域はリモート・サーバーと時間を交換するために計測されたネットワークの遅延です。

jitterは同じソースからのオフセット値間のずれを計測します。

refidはリモート・システムがその時刻を何処で取得したかを示します。

st列は階層番号を示します。階層ゼロのシステムは信頼できるソースに直接接続する必要があります。

poll列はどのくらい頻繁にこのサーバーがポーリングされているかを示します。**when**列は最後のポーリングからの秒単位の時間です。

reach列は最近のポーリングが成功したかどうかを示す8進数のビットマップです。377の値は最後の8回のポーリングが成功したことを示します。

GPS同期の精度

同期したGPSの動作中、RCIM POSIXクロックはGPS PPS信号に絶え間なく同期されている必要があります。精度を確認するため、同期の精度は1秒に1回計測され、直近2時間の動作の値が**/proc/driver/rcim/gps-stats**ファイルに一時的に記録されます。

NOTE

RedHawk 7.5以前では1時間の履歴だけが記録されます。

計測された精度はRCIM POSIXクロック(システム時刻の基準)とGPS PPS信号との間でナノ秒単位で差があります。

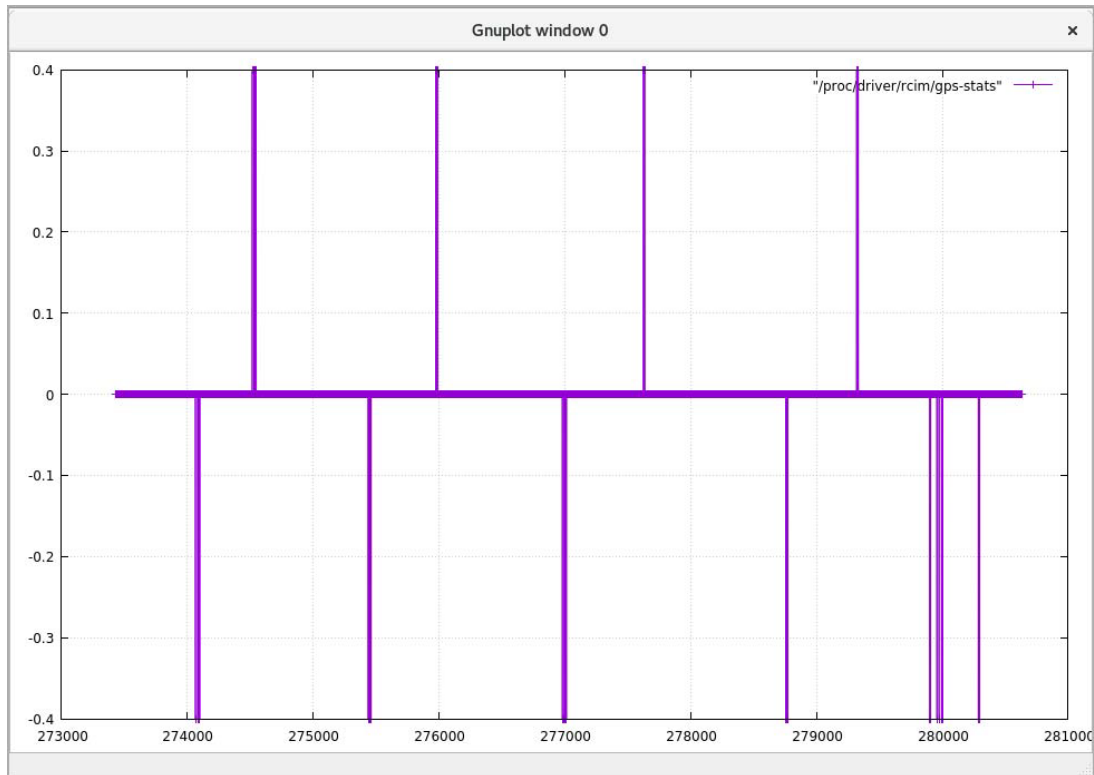
gps-statsファイルは直接見ることは可能ですが、**gnuplot**の利用が理想的で適しています。統計値をグラフ化するために**gnuplot**を使用するには、最初に次の内容を含めたテキスト・ファイル(例えば、名前を「**gps**」)を生成して下さい：

```
set grid;
while (1) {
    plot "/proc/driver/rcim/gps-stats" with linespoints;
    pause 1;
}
```

そして次のように**gnuplot**を起動して下さい：

gnuplot gps

次のスクリーンショットは**gnuplot**表示の例を示しており、Y軸は計測されたマイクロ秒単位の時間差、X軸はシステムが起動されてからの秒数となります。



詳細については**gnuplot(1)**を参照し、**info gnuplot**を起動して下さい。

IRIGタイムコードの同期

本章ではRCIM IVをIRIGマスターまたはスレーブにするための構成方法を説明します。

概要

オプションのIRIGモジュールは、IRIGタイムコードの仕様に準拠するIRIGマスターまたはスレーブとしてRCIM IVを機能させることが可能です。IRIG-B B124(AM)およびIRIG-B B004(DCLS)の両方の形式が、ある範囲の電圧や振幅信号と共にサポートされます。

RCIM IRIGマスターはIRIG信号をIRIGスレーブ(RCIM IRIGスレーブまたはサード・パーティのIRIGスレーブ・カードのどちらでも可能)に送信することが可能です。

RCIM IRIGスレーブはRCIMのPOSIXクロック(システム時刻の基準)をIRIGマスターに同期するために**chronyd**を使用します。RCIM IRIGスレーブ上のシステム時刻を制御する機能は、サード・パーティのIRIGカードに対しRCIM IRIGソリューションの重要な差別化要因となります。

IRIGマスター

IRIGモジュールを搭載するRCIM IVは開始時にデフォルトでIRIGマスターになり、RCIM IRIGマスターは手動での構成は必要なくIRIG時刻をRCIMのPOSIXクロックに自動的に同期します。

(GPSまたはネットワーク・タイムサーバーを介して)制御されるシステム時刻を使用するRCIM IRIGマスター・システムは、RCIMのPOSIXクロックにシステム時刻を自動的に同期し、これはうるう日、うるう秒、そしてその他全てのシステム時刻の変更を自動的に構成するためにIRIG時刻を有効にします。詳細については4章の「GPSクロックの同期」および**chronyd(1)**を参照して下さい。

出力信号

RCIM IRIGマスターはアナログ振幅変調(AM: Amplitude Modulation)信号、デジタルDCレベル・シフト(DCLS)信号、または両方の組み合わせを出力するように構成することが可能です。必要とするIRIG信号タイプの詳細および構成を次のセクションで説明します。

AM信号

RCIM IRIGマスターのAM信号出力はある範囲の電圧を出力することが可能で、殆どのIRIGスレーブと互換であるはずです。詳細なAM信号の出力仕様は次のとおりです：

- 最大振幅2.5V, 5V, 8V (選択可能)
- 50Ωインピーダンス
- 1KHz 3:1レシオ
- ±10マイクロ秒の精度

RCIM IRIGマスターは外部ピンEXT_AMOUTを介してAM信号を出力するように自動的に構成されます。詳細については図2-3「RCIM IV外部割込みI/Oコネクタのピン配列」を参照して下さい。

AM信号出力電圧はデフォルトで2.5Vになります。必要であれば、AM信号出力を出力ソース構成構文を使い手動で2.5V, 5Vまたは8Vのいずれかに設定することが可能です。例えば、AM信号出力を5Vに設定するにはrootユーザーとして次のコマンドを実行して下さい：

```
echo '5V|irig_amp' > /proc/driver/rcim/config
```

使用するAM信号出力電圧を決定したら、**/etc/sysconfig/rcim**の行に追加して電圧を起動時に自動的に構成されるようにすることが可能です。例：

```
RCIMCONFIG_3="5V|irig_amp"
```

詳細については**/etc/sysconfig/rcim**ファイルのコメントを参照して下さい。

RCIM IRIGマスターは1個のAM信号のみを出力することが可能です。AM信号と共に複数の追加のDCLS信号を出力したい場合は次項を参照して下さい。

DCLS信号

RCIM IRIGマスターのDCLS信号出力は複数のTTL電圧をサポートしており、殆どのIRIGスレーブと互換であるはずです。詳細なDCLS信号の出力仕様は次のとおりです：

- ハイ>2Vかつロー<0.5Vの3.5Vおよび5V TTLに対応
- 100Ωインピーダンス
- ±1マイクロ秒の精度

RCIM IRIGマスターは外部ピンEXT_INOUT0からEXT_INOUT11の組み合わせを介して12個ものDCLS信号を出力することが可能です。詳細については図2-3「RCIM IV外部割込みI/Oコネクタのピン配列」を参照して下さい。

DCLS信号を出力するために外部ピンを有効にするには、出力ソース構成構文を使いピンを構成して下さい。例えば、DCLS信号を出力する外部ピンEXT_INOUT2を構成するにはrootユーザーとして次のコマンドを実行して下さい：

```
echo 'dcls_out|out2,pin2/out/n' > /proc/driver/rcim/config
```

状況：

<code>dcls_out</code>	DCLS信号ソースを選択します
<code>out2</code>	ソースが出力であることを指定します
<code>pin2</code>	使用する外部ピン(EXT_INOUT2)を指定します
<code>/out/n</code>	ピンを終端抵抗なしの出力とするよう構成します

どの外部ピン(複数指定可)にDCLS信号を出力するかを決定したら、**`/etc/sysconfig/rcim`**の行に追加してピンを起動時に自動的に構成されるようにすることが可能です。例：

```
RCIMCONFIG_4="dcls_out|out1,pin1/out/n"
RCIMCONFIG_5="dcls_out|out7,pin7/out/n"
```

詳細については**`/etc/sysconfig/rcim`**ファイルのコメントを参照して下さい。

伝搬遅延

上級ユーザーは、信号伝搬遅延を補うためにIRIGマスター信号を早期に出力するようタイミング・オフセットを構成することが可能です。タイミング・オフセットは400ナノ秒間隔で0ナノ秒から65,520ナノ秒(65.52ミリ秒)の間のいずれの値を設定することが可能です。デフォルトでタイミング・オフセットは0 (調整なし)となります。

例えば、IRIG信号伝搬を50マイクロ秒早く出力するよう設定するには、rootユーザーとして次のコマンドを実行して下さい：

```
echo '50000|irig_adj' > /proc/driver/rcim/config
```

使用するIRIG信号伝搬遅延を決定したら、**`/etc/sysconfig/rcim`**の行に追加して伝搬遅延を起動時に自動的に構成されるようにすることが可能です。例：

```
RCIMCONFIG_6="50000|irig_adj"
```

詳細については**`/etc/sysconfig/rcim`**ファイルのコメントを参照して下さい。

IRIGスレーブ

IRIGモジュールを搭載するRCIM IVは、IRIGマスターからIRIG信号を受信するIRIGスレーブとして構成することが可能です。デフォルトでRCIMはIRIGマスターとして構成されますが、IRIG PPS信号でRCIMのPOSIXクロックを同期するために**`chronyd`**を実行すると自動的にRCIMをIRIGスレーブとなるように構成します。

RCIMのPOSIXクロックはシステム時刻の基準であるため、RCIM IRIGスレーブ・アプリケーションは**`clock_gettime(2)`**を呼び出すだけで正確なIRIG時刻を直ぐに得ることが可能です。**`clock_gettime(2)`**の呼び出しは、`ioctl`を使いPCI Expressバスを越えて複数のIRIGレジスタを読むよりも非常に高速となります。

構成

次のrefclockの2行を**/etc/chrony.conf**に追加してRCIM IRIGスレーブを構成して下さい。

```
refclock PPS /dev/pps1 lock PTP refid IRIG
refclock PHC /dev/ptp4 refid PTP noselect
```

NOTE

上の例では**/dev/pps1**と**/dev/ptp4**を使用していますが、異なるハードウェア構成のシステムでは異なる可能性があります。お手持ちのシステムで適切なデバイスを確認するには、起動直後にカーネル・メッセージを検索して下さい。例：

```
# dmesg | grep -m1 -e 'pps.*ptp.*'
[35.205135] pps pps1: new PPS source ptp4
```

上記出力は**/dev/pps1**と**/dev/ptp4**がこのシステムでは適切なデバイスであることを示しています。

NOTE

上記refclockの2行は、**/etc/chrony.conf**ファイルの中に存在する唯一のrefclock行である必要があり、コメントアウトされた行のいずれのインスタンスも無視されます。

NOTE

システムはネットワーク・ベースの時刻ソースを使ってシステム時刻を同期する他のNTPまたはPTPデーモンを実行すべきではありません。インストールされたいずれのntpdやptpdサービスも無効化かつ停止されていることを確認して下さい。

/etc/chrony.confを正しく構成したら、**chronyd**を開始するためrootユーザーとして次のコマンドを実行して下さい：

```
systemctl enable chronyd
systemctl restart chronyd
```

あるいは、**systemd**のない古いシステムでは次のコマンドを実行して下さい：

```
chkconfig chronyd on
service chronyd start
```

IRIG PPS同期が開始されたら、RCIMはIRIGスレーブとして構成されます。

RCIM IRIGスレーブで**chronyd**を実行する主な利点は、システム時刻はIRIG時刻と密接に同期するため、アプリケーションはIRIG時刻を得るのに**clock_gettime(2)**を呼び出すだけであることです。

本機能はRCIM IRIGスレーブ・ソリューション特有のものなのですが、デバイスのioctlを使ってRCIM IRIGとインターフェースを取りたい開発者は、詳細について5-8ページの「IRIGプログラミング・インターフェース」項を参照することが可能です。

入力信号

RCIM IRIGスレーブはアナログ振幅変調(AM: Amplitude Modulation)信号またはデジタルDCレベル・シフト(DCLS)信号を受信することが可能です。各RCIM IRIGスレーブは1つのIRIG信号入力だけ(外部ピンを介してAMまたはDCLSのいずれか)を受信することが可能です。必要なIRIG信号タイプの詳細と構成については次のセクションで説明します。

AM信号

RCIM IRIGスレーブのAM信号入力はデフォルトで有効化されています。AM信号入力はある範囲の電圧を受け入れ、殆どのIRIGマスターと互換であるはずですが。詳細なAM信号入力の仕様は次のとおりです：

- 最大振幅1V～10V (オート・レンジ)
- 1KHz 2:1～6:1レシオ
- AC結合された4K Ω 負荷

AM信号入力を使うには、IRIGマスターのAM信号出力(RCIM IRIGマスター用のEXT_AMOUT)をRCIM IRIGスレーブのAM信号入力(EXT_AMIN)に接続して下さい。前述のとおりで**chrony**dを実行する以外に追加の構成は必要ありません。

RCIM IRIGスレーブがIRIGマスターと正しく同期していることを確認するには、次のコマンドを実行して下さい：

```
grep Slave /proc/driver/rcim/irig
```

次と似たような出力が表示されるはずです：

```
IRIG Slave Input:      enabled: yes, PPS: good,
                       input type: AM
```

更なる確認については5-6ページの「同期の精度」項を参照して下さい。

DCLS信号

RCIM IRIGスレーブのDCLS信号入力はある範囲の電圧を受け入れ、殆どのIRIGマスターと互換であるはずですが。詳細なDCLS信号入力の仕様は次のとおりです：

- ハイ>2V、ロー<0.8Vの3.5Vまたは5V TTL
- 切り替え可能な100 Ω 終端抵抗

DCLS信号入力を使うには、IRIGマスターのDCLS信号出力をRCIM IRIGスレーブの外部ピンに接続し、入力ソース構成構文を使ってピンを構成して下さい。例えば、DCLS信号を入力するように外部ピンEXT_INOUT7を構成するには、rootユーザーとして次のコマンドを実行してください：

```
echo 'pin7|dcls_in,pin7/in/n' > /proc/driver/rcim/config
```

状況：

<code>pin7</code>	ソースが入力であることを指定します
<code>dcls_in</code>	DCLS信号ソースを選択します
<code>pin7</code>	使用する外部ピン(EXT_INOUT7)を指定します
<code>/in/n</code>	ピンを終端抵抗なしの入力とするよう構成します

NOTE

DCLS信号入力を有効にするとAM信号入力は無効となります。
AM信号入力は再起動または次のコマンドを実行することで再度有効にすることが可能です：

```
echo 'none|dcls_in' > /proc/driver/rcim/config
```

RCIM IRIGスレーブがIRIGマスターと正しく同期していることを確認するには、次のコマンドを実行して下さい：

```
grep Slave /proc/driver/rcim/irig
```

次と似たような出力が表示されるはずです：

```
IRIG Slave Input:      enabled: yes, PPS: good,
                      input type: DCLS, input pin: 7
```

更なる確認については5-6ページの「同期の精度」項を参照して下さい。

どの外部ピンがDCLS信号を受信するかを決定したら、**/etc/sysconfig/rcim**に行を追加してピンを起動時に自動的に構成されるようにすることが可能です。例：

```
RCIMCONFIG="pin5|dcls_in,pin5/in/n"
```

詳細については**/etc/sysconfig/rcim**ファイルのコメントを参照して下さい。

同期の精度

同期したIRIGスレーブが動作中、RCIM IRIGスレーブのPOSIXクロック(システム時刻の基準)は、DCLS入力信号を使用している場合はIRIG PPS信号の1マイクロ秒以内、AM入力信号を使用している場合はIRIG PPS信号の10マイクロ秒以内で同期されています。

同期が行われている時、同期の精度は1秒毎に計測され、直近2時間の動作の値が**/proc/driver/rcim/gps-stats**ファイルに一時的に記録されます。

NOTE

RedHawk 7.5以前では1時間の履歴だけが記録されます。

計測された精度はRCIM POSIXクロック(システム時刻の基準)とIRIGマスターからの信号で直接駆動されるIRIG PPS信号との間でナノ秒単位で差があります。

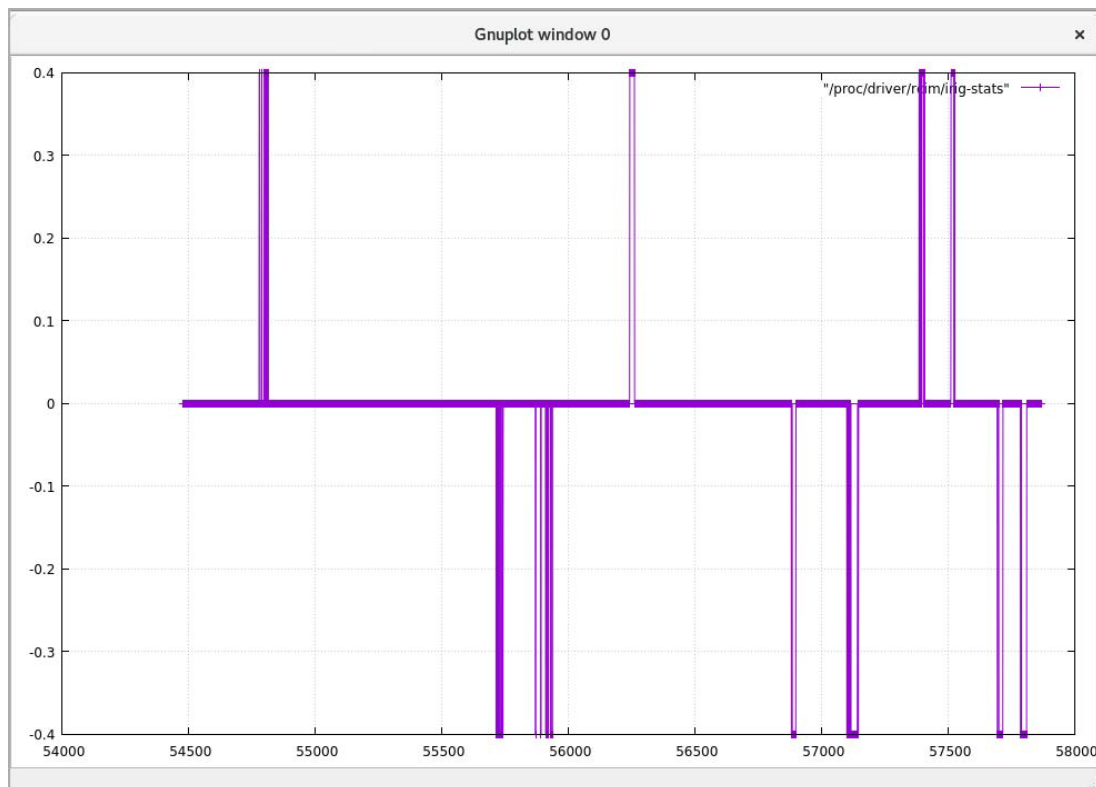
irig-stats ファイルは直接見ることは可能ですが、**gnuplot**の利用が理想的で適しています。統計値をグラフ化するために**gnuplot**を使用するには、最初に次の内容を含めたテキスト・ファイル(例えば、名前を「**irig**」)を生成して下さい：

```
set grid;
while (1) {
    plot "/proc/driver/rcim/irig-stats" with linespoints;
    pause 1;
}
```

そして次のように**gnuplot**を起動して下さい：

```
gnuplot irig
```

次のスクリーンショットは**gnuplot**表示の例を示しており、Y軸は計測されたマイクロ秒単位の時間差、X軸はシステムが起動されてからの秒数となります。



詳細については**gnuplot(1)**を参照し、**info gnuplot**を起動して下さい。

IRIGプログラミング・インターフェース

NOTE

IRIGプログラミング・インターフェースの利用は大抵は不要です。IRIGマスターとスレーブは、5-1ページの「IRIGマスター」項および5-3ページの「IRIGスレーブ」項で説明しているようにプログラミングの必要なしに構成、アクセス、制御することが可能です。

RCIMドライバーは、RCIMカードがIRIGモジュールを搭載する場合に現れるIRIGレジスタや状態の情報へのアクセスに使用可能ないくつかの*ioctl*を提供します。提供された*ioctl*を使用するには、最初にRCIM IRIGモジュールの有効なファイル記述子を得るために **/dev/rcim:0/irig** をオープンし、続いてファイル記述子を使って必要な*ioctl*を呼び出して下さい。

デフォルトで **/dev/rcim:0/irig** は全てのユーザーにデバイスの読み取りから書き込みまでを許可するファイル権限を持っていますが、アクセスを制限するためにudevルールを使用することが可能です。例えばRedHawk 8.Xにおいて、**/etc/udev/rules.d/99-irig.rules**に次を設定するとrootユーザーおよび新しいirigグループに属する他のユーザーにデバイスのアクセスを制限します：

```
KERNEL=="rcim:0/irig", OWNER=="root", GROUP=="irig",
MODE=="0660"
```

再起動で、**/dev/rcim:0/irig** デバイス・ファイルは次の権限を所有します：

```
# ls -l /dev/rcim:0/irig
crw-rw---- 1 root irig 508, 224 Aug 8 15:24 /dev/rcim:0/irig
```

2セットの*ioctl*が提供されます。1セットはIRIGマスターの情報にアクセスするためのもので、もう1セットがIRIGスレーブの情報にアクセスするためのものです。*ioctl*の各セットの詳細は次のセクションで説明します。詳細については***ioctl*(2)**を参照して下さい。

上級ユーザーはRCIM IRIGレジスタに直接アクセスするために**/dev/rcim:0/irig** デバイスを **mmap(2)** することも可能です。詳細については付録Aの「RCIM IVレジスタ」を参照して下さい。

IRIGマスター*ioctl*

次のセクションではRCIM IRIGマスターのIRIGレジスタや状態の情報にアクセスするために提供される*ioctl*について説明します。示されたいずれの引数も**/dev/rcim:0/irig**への最初のファイル記述子引数の後に登場する追加の*ioctl*引数です。

IRIG_OUTPUT_ENABLE

概要：	IRIGマスターの動作を有効にします。
説明：	RCIMはIRIGタイムコード形式でのRCIMのPOSIXクロックの継続的な出力を開始します。
引数：	なし

IRIG_OUTPUT_DISABLE

- 概要： IRIGマスターの動作を無効にします。
- 説明： RCIMはIRIGタイムコード形式でのRCIMのPOSIXクロック時刻の出力を停止します。
- 引数： なし

IRIG_OUTPUT_STATUS

- 概要： IRIGマスターの現在の状態を問い合わせます。
- 説明： 現在の状態を引数で指し示された整数にコピーします。**1**の状態はRCIMがIRIGタイムコード信号を出力していることを示します。**0**の状態はRCIMがIRIGタイムコード信号を出力していないことを示します。
- 引数： `int *`

IRIG_OUTPUT_RESET

- 概要： IRIGマスターをリセットします。
- 説明： 全てのIRIGマスターのプログラム化された情報を消去します(例えば、出力タイプ、出力ピン)。
- 引数： なし

IRIG_OUTPUT_GET_CONTROL_BITS

- 概要： IRIG Output Control Bitsレジスタを問い合わせます。
- 説明： IRIG Output Control Bitsレジスタは、IRIGマスターにより出力され、ユーザー定義領域に含まれる可能性のあるIRIGタイムコード信号内に含めるための情報を保持します。本レジスタを引数で指し示された整数にコピーします。値の下位18ビットだけが重要です。
- 引数： `int *`

IRIG_OUTPUT_SET_CONTROL_BITS

- 概要： IRIG Output Control Bitsレジスタを設定します。
- 説明： IRIG Output Control Bitsレジスタは、IRIGマスターにより出力され、ユーザー定義領域に含まれる可能性のあるIRIGタイムコード信号内に含めるための情報を保持します。引数で指し示された整数を本レジスタにコピーします。値の下位18ビットだけが重要です。
- 引数： `int *`

IRIG_GET_LEAP_SECOND

- 概要： IRIGマスターのうるう秒を問合せします。
- 説明： 次のうるう秒の挿入時刻に関するIRIGマスターのプログラム化された情報を問合せ、引数で指し示された構造体にその時間を書き込みます。書き込まれる時刻は**irig_master_leap_second_args**形式で、**rcim.h**に含まれていて次のように定義されています：

```
struct irig_master_leap_second_args {
    unsigned int year;
    unsigned int day;
    unsigned int hour;
}
```

- 引数： **struct irig_master_leap_second_args ***

IRIG_SET_LEAP_SECOND

- 概要： IRIGマスターのうるう秒を設定します。
- 説明： 引数で指し示された時間の構造体で指定される年、日、時の59分59秒でうるう秒の挿入を実行するようにIRIGマスターをプログラムします。年、日、時は**irig_master_leap_second_args**形式を使い表わされ、**rcim.h**に含まれていて次のように定義されています：

```
struct irig_master_leap_second_args {
    unsigned int year;
    unsigned int day;
    unsigned int hour;
}
```

- 引数： **struct irig_master_leap_second_args ***

IRIGスレーブioctl

RCIM IRIG情報をアクセスするためのioctlの使用は、PCI Expressバスを越えたRCIM IRIGレジスタの読み取りと書き込みが必要です。RCIM IRIGスレーブにおいて、IRIG時刻にアクセスするためのこれらのioctlの使用は**clock_gettime**の呼び出しよりも著しく遅くなります。詳細については**clock_gettime(2)**を参照して下さい。

次のセクションではRCIM IRIGスレーブのIRIGレジスタや状態の情報にアクセスするために提供されるioctlについて説明します。示されたいずれの引数も**/dev/rcim:0/irig**への最初のファイル記述子引数の後に登場する追加のioctl引数です。

IRIG_INPUT_ENABLE

概要 :	IRIGスレーブの動作を有効にします。
説明 :	有効化すると有効なAMまたはDCLS信号が存在する場合、RCIMはRCIMのPOSIXクロックを同期するために使用する chrony dに対してIRIG PPS信号を提供します。
引数 :	なし

IRIG_INPUT_DISABLE

概要 :	IRIGスレーブの動作を無効にします。
説明 :	無効化するとRCIMはIRIG PPS信号を提供しなくなり、リセット状態に入ります。全てのIRIGスレーブのプログラム化情報(例えば、入力タイプ、入力ピン)は消去されます。
引数 :	なし

IRIG_INPUT_STATUS

概要 :	IRIGスレーブの現在の状態を問い合わせます。
説明 :	現在の状態を引数で指し示された整数にコピーします。 1 の状態はIRIGスレーブの動作が現在有効であることを示します。 0 の状態はIRIGスレーブの動作が現在無効であることを示します。
引数 :	<code>int *</code>

IRIG_GET_TIME

概要 :	IRIGスレーブの現在の時刻を問い合わせます。
説明 :	IRIGスレーブの最新時刻のスナップショットを rtc.h に含まれ次のように定義される rtc_time 形式を使う引数で指し示された構造体にコピーします :

NOTE

`tm_sec`, `tm_min`, `tm_hour`, `tm_year`, `tm_yday`の構造体領域のみが更新されます。

```
struct rtc_time {
    int tm_sec;
    int tm_min;
    int tm_hour;
    int tm_mday;
    int tm_mon;
```

```

        int tm_year;
        int tm_wday;
        int tm_yday;
        int tm_isdst;
    }

```

引数 : `struct rtc_time *`

IRIG_GET_NS64

概要 : IRIGスレーブの現在の時刻をナノ秒単位で問い合わせます。

説明 : IRIGスレーブの最新時刻のナノ秒単位のスナップショットを引数で指し示された値にコピーします。スナップショットは、**rcim.h**に含まれ64 bit整数のtypedefとして定義されるns64形式を使う紀元(1970年1月1日午前0時UTC)からのナノ秒です。

引数 : `ns64 *`

IRIG_GET_TIME_RAW

概要 : IRIGスレーブの現在のRAW時刻を問い合わせます。

説明 : IRIGスレーブの最新RAW時刻のスナップショットを**rtc.h**に含まれ次のように定義される**rtc_time**形式を使う引数で指し示された構造体にコピーします :

NOTE

`tm_sec`, `tm_min`, `tm_hour`, `tm_year`, `tm_yday`の構造体領域のみが更新されます。加えて、RAW時刻のスナップショットは`tm_year`については最下位2桁のみを保持し、`tm_sec`の値は常に過去1秒となります。

```

    struct rtc_time {
        int tm_sec;
        int tm_min;
        int tm_hour;
        int tm_mday;
        int tm_mon;
        int tm_year;
        int tm_wday;
        int tm_yday;
        int tm_isdst;
    }

```

引数 : `struct rtc_time *`

A

RCIM IVレジスタ

本項にはRCIM IVボードのアドレス・マップとレジスタが含まれます。

一部のレジスタは物理アドレス空間の2つの場所に現れる事に注意して下さい。これらのレジスタについては、関連する*Xregister*があります。例えば、PCSATとXPCSATです。*Xregister*は古い4kページ・サイズではなく64kのシステムに対応します。

RCIM IVアドレス・マップ

Address	Function	
0xXXX00000	Board Status/Control Register	
0xXXX00004	Firmware Rev/Options Present Register	(Read Only)
0xXXX00008	Board Identity Register	
0xXXX00010	Interrupt Enable Register #1	
0xXXX00014	Interrupt Enable Register #2	
0xXXX00020	Interrupt Request Register #1	(Write Only)
0xXXX00024	Interrupt Request Register #2	(Write Only)
0xXXX00020	Interrupt Pending Register #1	(Read Only)
0xXXX00024	Interrupt Pending Register #2	(Read Only)
0xXXX00028	Interrupt Pending Extra Register #1	(Read Only)
0xXXX0002C	Interrupt Pending Extra Register #2	(Read Only)
0xXXX00030	Interrupt Clear Register #1	
0xXXX00034	Interrupt Clear Register #2	
0xXXX00040	Interrupt Arm Register #1	
0xXXX00044	Interrupt Arm Register #2	
0xXXX00050	Interrupt Select Level Register #1	
0xXXX00054	Interrupt Select Level Register #2	
0xXXX00060	Interrupt Select Polarity Register #1	
0xXXX00064	Interrupt Select Polarity Register #2	
0xXXX00070	External Interrupt Routing Register #1	
0xXXX00074	External Interrupt Routing Register #2	
0xXXX00078	External Interrupt Routing Register #3	
0xXXX00080	Cable Interrupt Routing Register #1	
0xXXX00084	Cable Interrupt Routing Register #2	
0xXXX00088	Cable Interrupt Routing Register #3	
0xXXX00200	GPS PPS Snapshot Register	(Read Only)
0xXXX00204	GPS PPS Seconds Snapshot Register	(Read Only)
0xXXX00210	Cable Snapshot Register	(Read Only)
0xXXX00214	Cable Seconds Snapshot Register	(Read Only)
0xXXX00220	Cable Master Time Register	(Read Only)
0xXXX00230	IRIG PPS Snapshot Register	(Read Only)
0xXXX00234	IRIG PPS Seconds Snapshot Register	(Read Only)

Address	Function	
0xXXX00400	Clear Cable Errors Register	(Write Only)
0xXXX00410	Output Cable Status Register	(Read Only)
0xXXX00420	Input Cable Status Register	(Read Only)
0xXXX01000	Tick Clock Upper Register	
0xXXX10000	Tick Clock Lower Register	
0xXXX01008	Tick Clock Lower Register	
0xXXX10008	Tick Clock Lower Register	
0xXXX01010	Tick Clock Status/Control Register	
0xXXX10010	Tick Clock Status/Control Register	
0xXXX01100	POSIX Clock Seconds Register	
0xXXX10100	POSIX Clock Seconds Register	
0xXXX01108	POSIX Clock Nanoseconds Register	
0xXXX10108	POSIX Clock Nanoseconds Register	
0xXXX01110	POSIX Clock Status/Control Register	
0xXXX10110	POSIX Clock Status/Control Register	
0xXXX01114	POSIX Clock Skip/Add Time Register	(Write Only)
0xXXX10114	POSIX Clock Skip/Add Time Register	(Write Only)
0xXXX01120	Clock Frequency Adjust Register	
0xXXX10120	Clock Frequency Adjust Register	
0xXXX01150	External Clock Input Select	
0xXXX01500- 0xXXX015F0	MSI-X Lower Address N	
0xXXX01504- 0xXXX015F4	MSI-X Upper Address N	
0xXXX01508- 0xXXX015F8	MSI-X Message Data N	
0xXXX0150C- 0xXXX015FC	MSI-X Vector Control Register N	
0xXXX01600	MSI-X Pending Register	(Read Only)
0xXXX01700- 0xXXX01778	MSI-X Select #1 Register N	
0xXXX01704- 0xXXX0177C	MSI-X Select #2 Register N	
0xXXX01800- 0xXXX01878	MSI-X Time #1 Register N	
0xXXX01804- 0xXXX0187C	MSI-X Time #2 Register N	
0xXXX01900- 0xXXX0193C	MSI-X Interrupt Clear Register #1 N	(Write Only)
0xXXX02000	RTC #0 Control Register	
0xXXX02010	RTC #0 Timer Register	
0xXXX02014	RTC #0 Repeat Register	
0xXXX02020	RTC #1 Control Register	
0xXXX02030	RTC #1 Timer Register	
0xXXX02034	RTC #1 Repeat Register	
0xXXX02040	RTC #2 Control Register	
0xXXX02050	RTC #2 Timer Register	
0xXXX02054	RTC #2 Repeat Register	

Address	Function	
0xFFFF02060	RTC #3 Control Register	
0xFFFF02070	RTC #3 Timer Register	
0xFFFF02074	RTC #3 Repeat Register	
0xFFFF02080	RTC #4 Control Register	
0xFFFF02090	RTC #4 Timer Register	
0xFFFF02094	RTC #4 Repeat Register	
0xFFFF020A0	RTC #5 Control Register	
0xFFFF020B0	RTC #5 Timer Register	
0xFFFF020B4	RTC #5 Repeat Register	
0xFFFF020C0	RTC #6 Control Register	
0xFFFF020D0	RTC #6 Timer Register	
0xFFFF020D4	RTC #6 Repeat Register	
0xFFFF020E0	RTC #7 Control Register	
0xFFFF020F0	RTC #7 Timer Register	
0xFFFF020F4	RTC #7 Repeat Register	
0xFFFF03000	Programmable Interrupt Generator Register	
0xFFFF30000	Programmable Interrupt Generator Register	
0xFFFF03010	Programmable INTR Generator Set Register	(Write Only)
0xFFFF30010	Programmable INTR Generator Set Register	(Write Only)
0xFFFF03020	Programmable INTR Generator Clear Register	(Write Only)
0xFFFF30020	Programmable INTR Generator Clear Register	(Write Only)
0xFFFF03040	External I/O Output Enable Register	
0xFFFF03044	External I/O Output Enable Set Register	
0xFFFF03048	External I/O Output Enable Clear Register	
0xFFFF03050	External I/O Terminator On Register	
0xFFFF03054	External I/O Terminator On Set Register	
0xFFFF03058	External I/O Terminator On Clear Register	
0xFFFF03200	GPS Receive Pointers Register	
0xFFFF03204	GPS Transmit Pointers Register	
0xFFFF03208	GPS Debug Control/Status Register	
0xFFFF0320C	GPS Communication Error Register	
0xFFFF03300	Board Information Register	(Read Only)
0xFFFF03304	Firmware Data Register	(Read Only)
0xFFFF03308	Firmware Revision Register	(Read Only)
0xFFFF03800	Firmware SPI Command/Status Register	
0xFFFF03804	Firmware SPI Address Register	
0xFFFF03808	Firmware Reload Register	
0xFFFF0380C	Remote Update Sector/Status Register	
0xFFFF03900- 0xFFFF039FC	Firmware SPI Buffer	
0xFFFF04000- 0xFFFF047FF	GPS Receive Data Buffer	
0xFFFF04800- 0xFFFF04FFF	GPS Transmit Data Buffer	
0xFFFF06000	IRIG Input Enable Register	
0xFFFF06004	IRIG Input Control Register	

Address	Function	
0xFFFF06008	IRIG Input Status Register	(Read Only)
0xFFFF0600C	IRIG Input Error Register	(Read Only)
0xFFFF06020	IRIG Input Seconds Register	(Read Only)
0xFFFF06024	IRIG Input Minutes Register	(Read Only)
0xFFFF06028	IRIG Input Hours Register	(Read Only)
0xFFFF0602C	IRIG Input Days Register	(Read Only)
0xFFFF06030	IRIG Input Years Register	(Read Only)
0xFFFF06034	IRIG Input Control Bits Register	(Read Only)
0xFFFF06038	IRIG Input SBS Register	(Read Only)
0xFFFF06060	IRIG Output Enable Register	
0xFFFF06064	IRIG Output Control Register	
0xFFFF06068	IRIG Output Adjust Register	
0xFFFF0606C	IRIG Output Leap Second Time Register	
0xFFFF06080	IRIG Output Seconds Register	
0xFFFF06084	IRIG Output Minutes Register	
0xFFFF06088	IRIG Output Hours Register	
0xFFFF0608C	IRIG Output Days Register	
0xFFFF06090	IRIG Output Years Register	
0xFFFF06094	IRIG Output Control Bits Register	
0xFFFF06098	IRIG Output SBS Register	
0xFFFF06100	IRIG ADC Data Register (Debug)	(Read Only)
0xFFFF06110- 0xFFFF0614C	IRIG ADC History Registers (Debug)	(Read Only)
0xFFFF06160	IRIG DAC Data Register (Debug)	
0xFFFF06164	IRIG DAC Command/Data Register (Debug)	

RCIM IVレジスタ

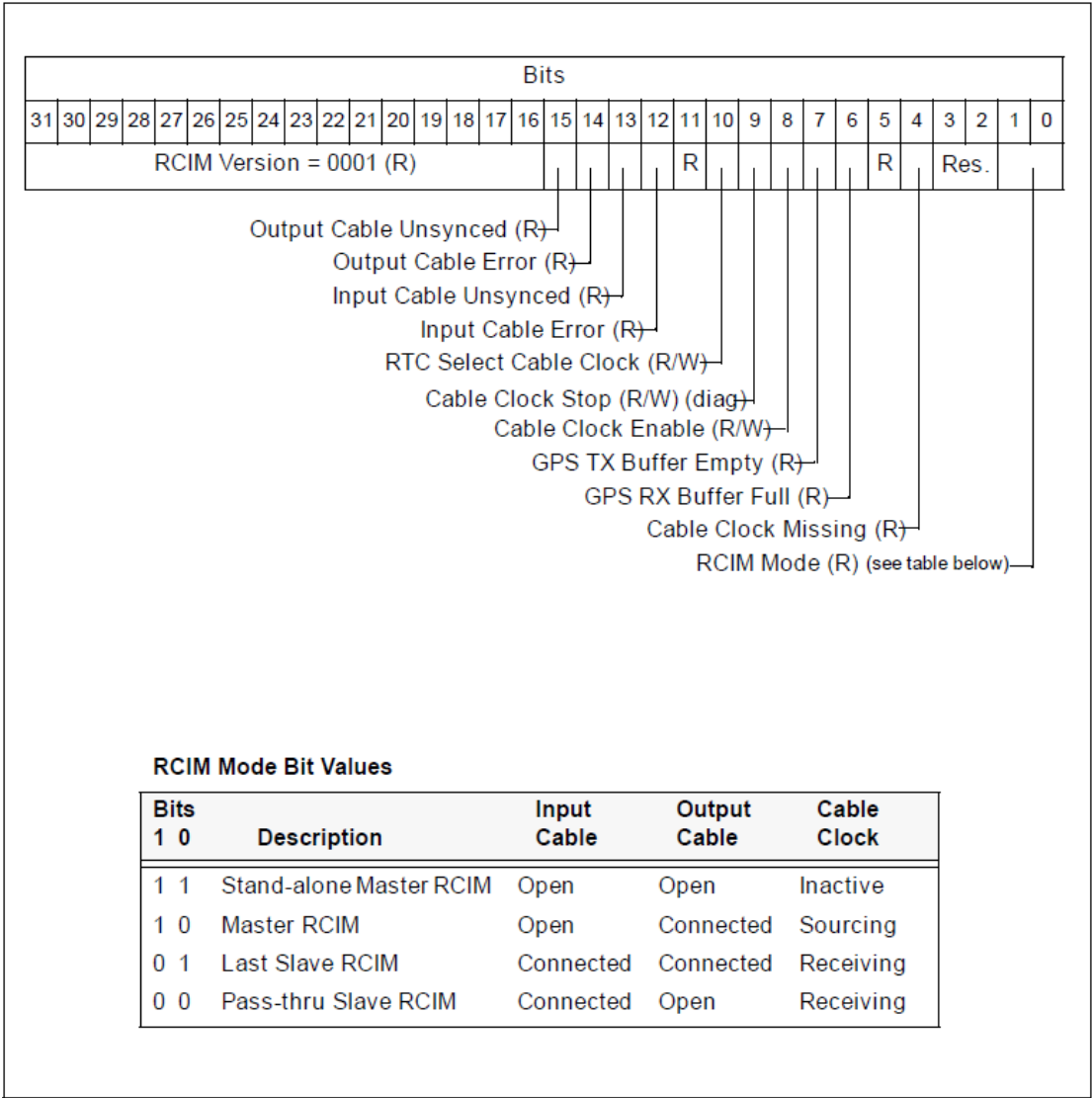
RCIM IVレジスタを本章内で図解します。

NOTE: 特に指定がない限り、ビットの値は1=on; 0=offとなります

図A-1 RCIM IV Board Status/Control Register

本レジスタはRCIM IVボードの特定の機能のステータスと制御を提供します。

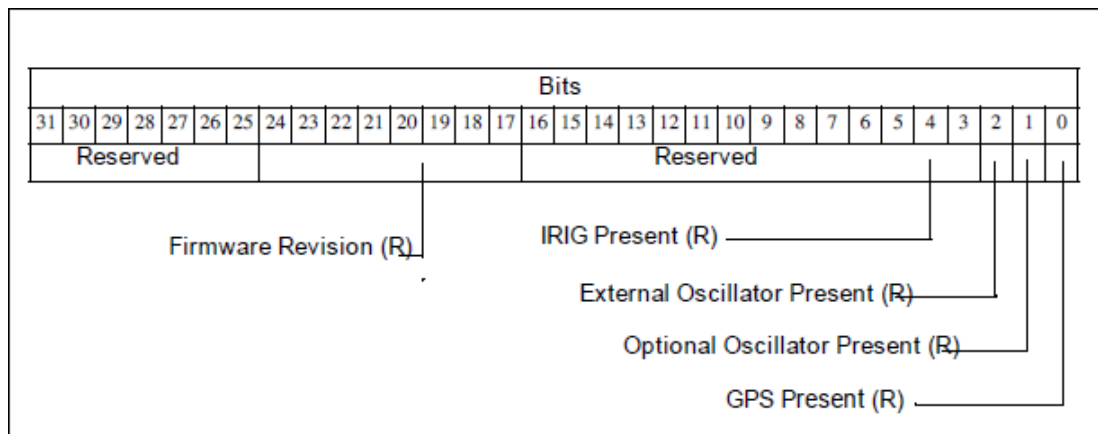
Offset: 00000



図A-2 RCIM IV Firmware Revision/Options Present Register

本レジスタはこのRCIMボードに存在するオプションとファームウェアのリビジョンに関する情報を提供します。

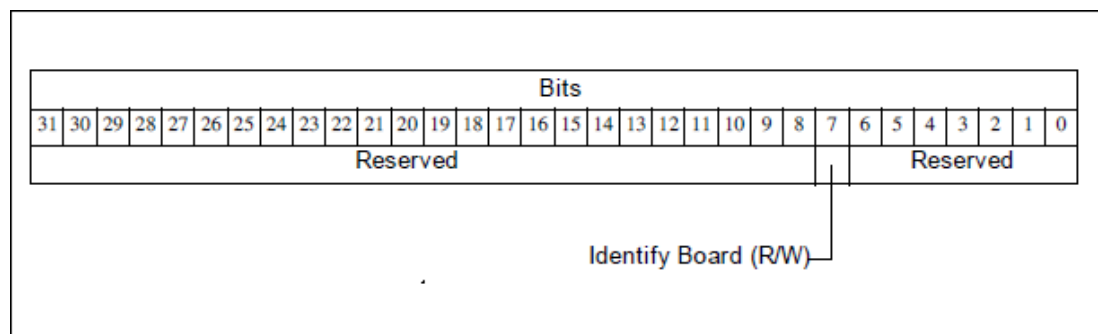
Offset: 00004



図A-3 RCIM IV Board Identity Register

7ビット目を設定すると特定のボードを認識するためにRCIMのLEDを1秒毎に赤と緑を点滅させます。1台のシステムに複数のRCIMがある場合に便利です。7ビット目をクリアすると通常の運用を再開します。

Offset: 00008



図A-4 RCIM IV Interrupt Enable/Request/Pending/Clear/Arm/Level/Polarity Registers

Enable Registersは選択された割込みを有効にします。

Request Registersは選択された割込みのソフトウェア駆動要求です。

Pending Registersは要求を保留します。

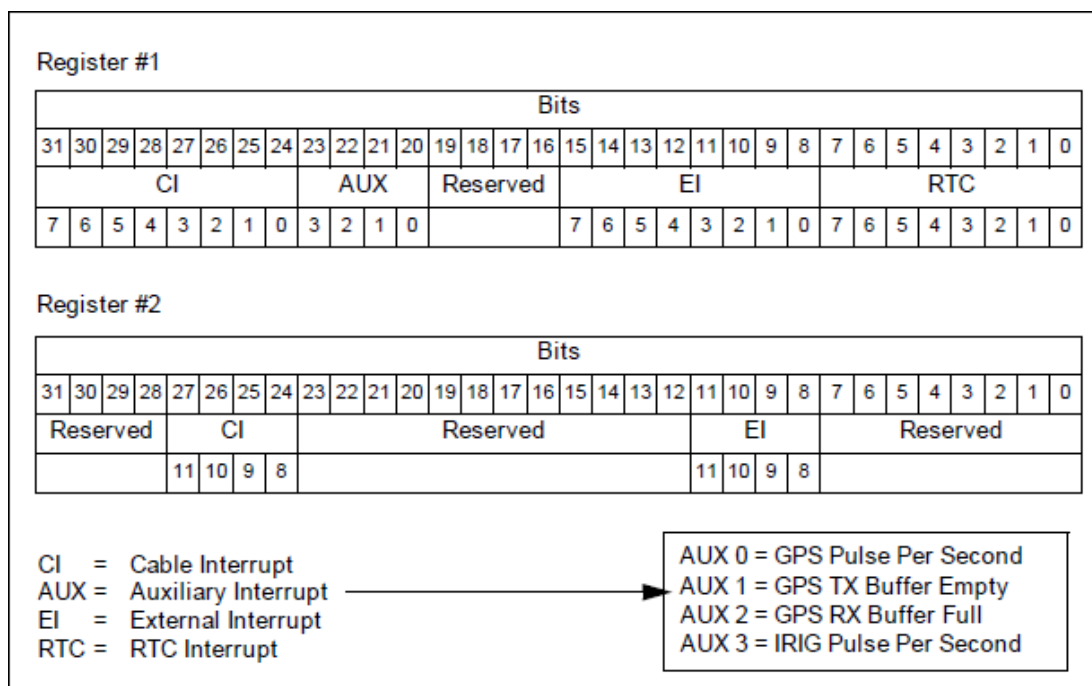
Clear Registersは選択された割込みを解放します。

Arm Registersはエッジ・トリガ用に選択された割込みを実装します。

Level Registersは選択された割込みに対してレベル(1)またはエッジ(0)を設定します。

Polarity Registersは選択された割込みに対してハイ(1)またはロー(0)の極性を設定します。

Offset: 00010, 00014, 00020, 00024, 00030, 00034, 00040, 00044, 00050, 00054, 00060, 00064



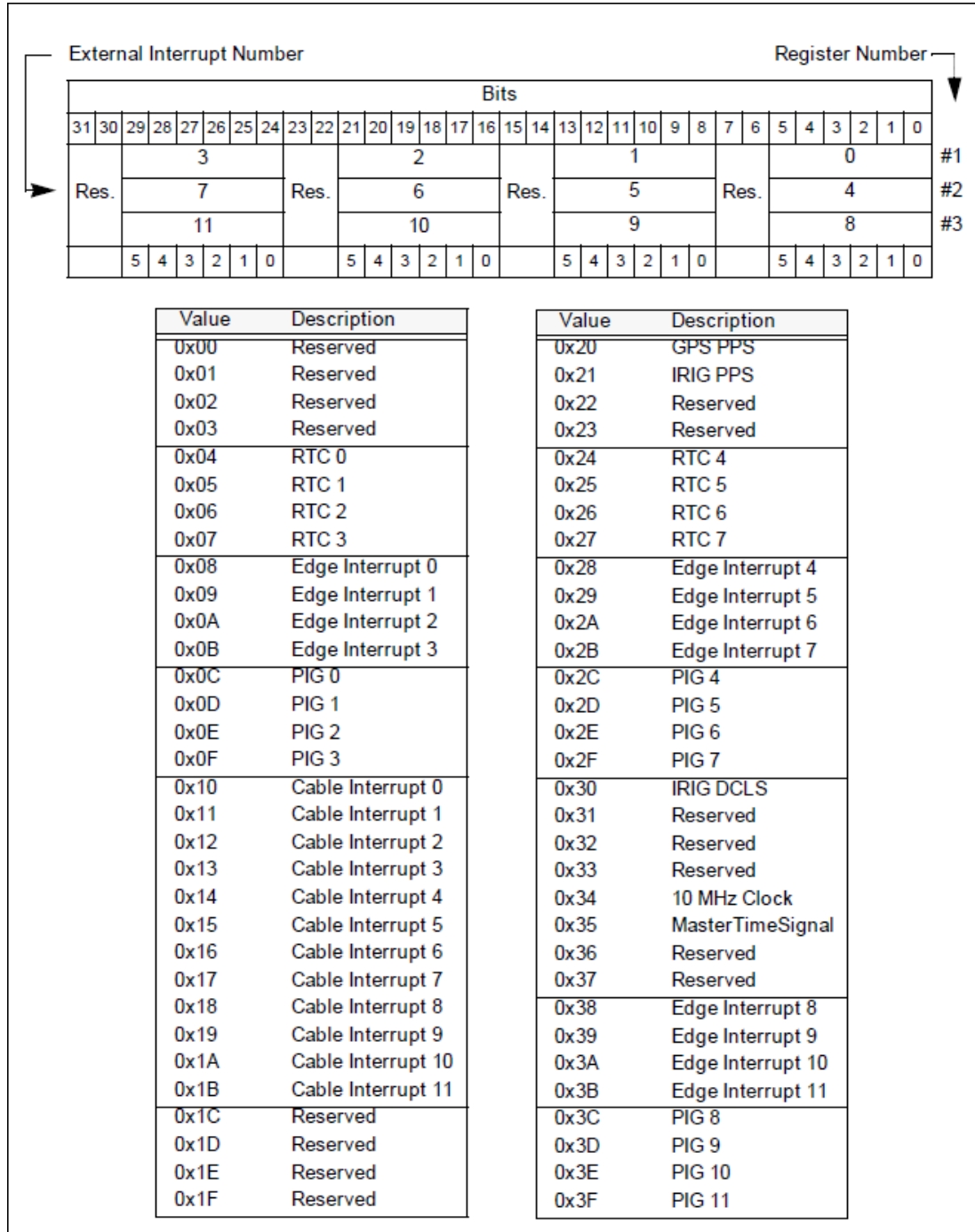
NOTE

上記レジスタ・マップには1つの例外があります。オフセット00020のRCIM IV Interrupt Pending Register #1のビット16は予約されていません。ビット16が「1」として読み取られた場合、オフセット00024のRCIM IV Interrupt Pending Register #2に保留中の割込みがないことを示しています。

図A-5 RCIM IV External Interrupt Routing Registers

External Interrupt Routing Registersは選択された割込みを外部割込みコネクタへのルートに設定します。

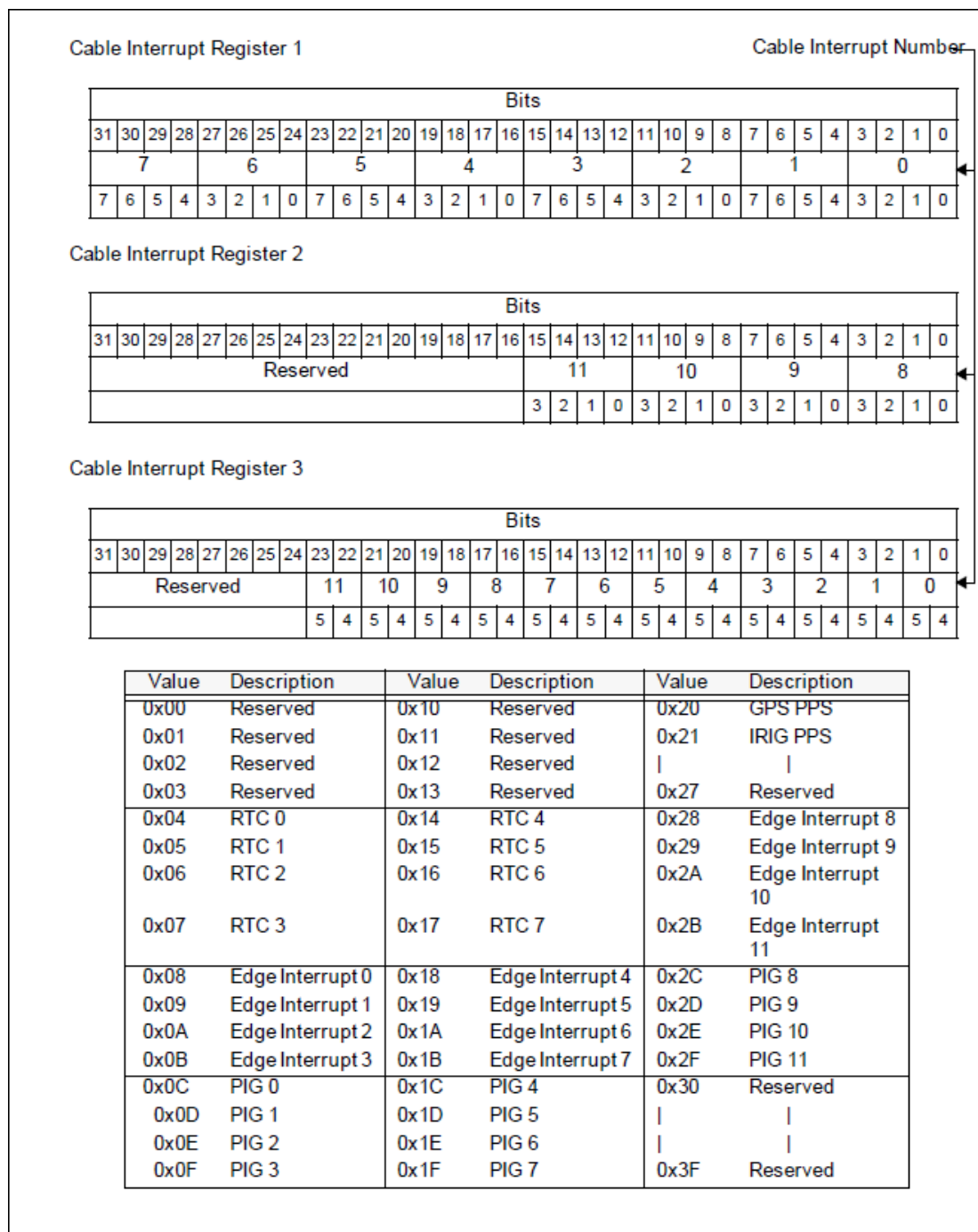
Offset: 00070, 00074, 00078



図A-6 RCIM IV Cable Interrupt Routing Registers

Cable Interrupt Routing Registersは選択された割込みをRCIM相互接続ケーブルへのルートに設定します。

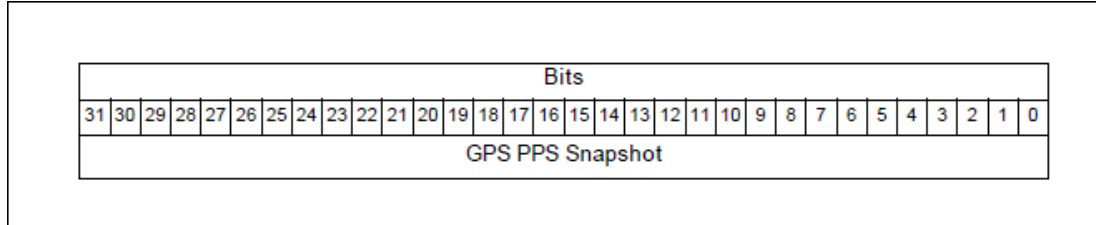
Offset: 00080, 00084, 00088



図A-7 RCIM IV GPS PPS Snapshot Register

GPS PPS Snapshot RegisterはPOSIXクロックのナノ秒領域と秒領域の2bitのスナップショットを含みます。スナップショットはGPS PPSシグナルが発生する度に取り込まれます。

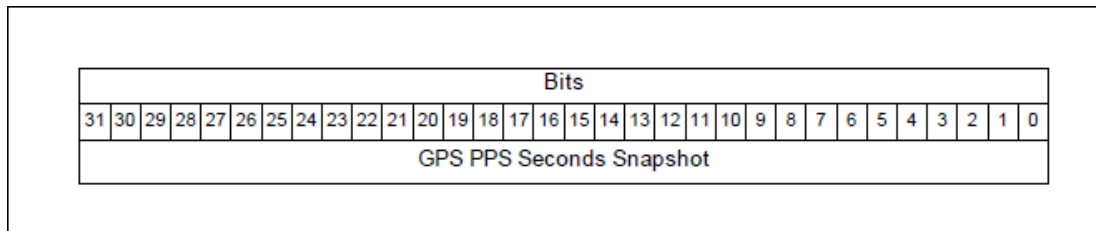
Offset: 00200



図A-8 RCIM IV GPS PPS Seconds Snapshot Register

GPS PPS Snapshot Seconds RegisterはPOSIXクロックの秒領域のスナップショットを含みます。スナップショットはGPS PPSシグナルが発生する度に取り込まれます。

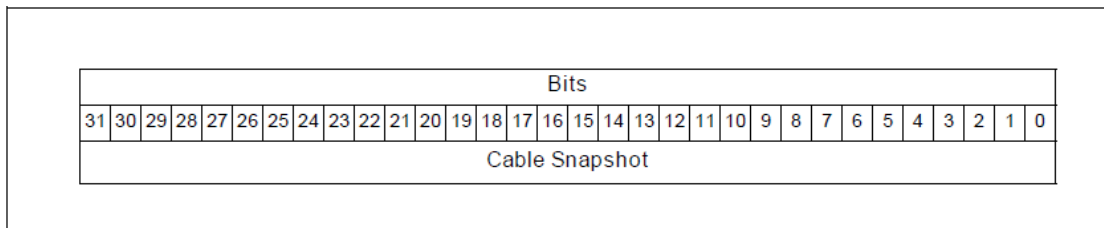
Offset: 00204



図A-9 RCIM IV Cable Snapshot Register

Cable Snapshot RegisterはPOSIXクロックのナノ秒領域と秒領域の2bitのスナップショットを含みます。スナップショットはケーブル・マスター時刻を受信する度に取り込まれます。

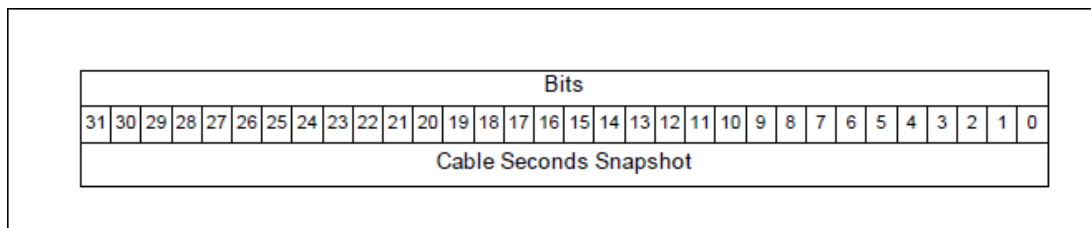
Offset: 00210



図A-10 RCIM IV Cable Seconds Snapshot Register

Cable Seconds Snapshot RegisterはPOSIXクロックの秒領域のスナップショットを含みます。スナップショットはケーブル・マスター時刻を受信する度に取り込まれます。

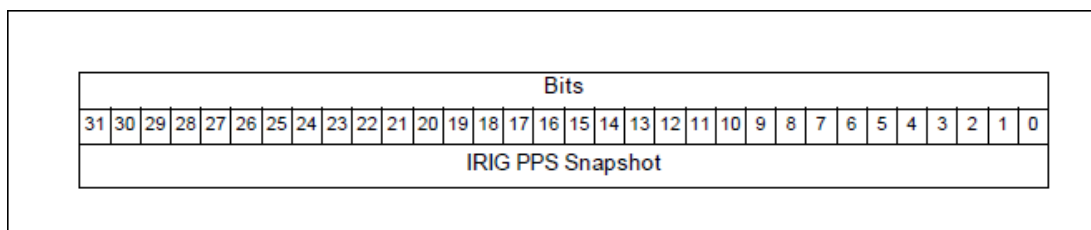
Offset: 00214



図A-11 RCIM IV IRIG PPS Snapshot Register

IRIG PPS Snapshot RegisterはPOSIXクロックのナノ秒領域と秒領域の2bitのスナップショットを含みます。スナップショットはIRIG PPSシグナルが発生する度に取り込まれます。

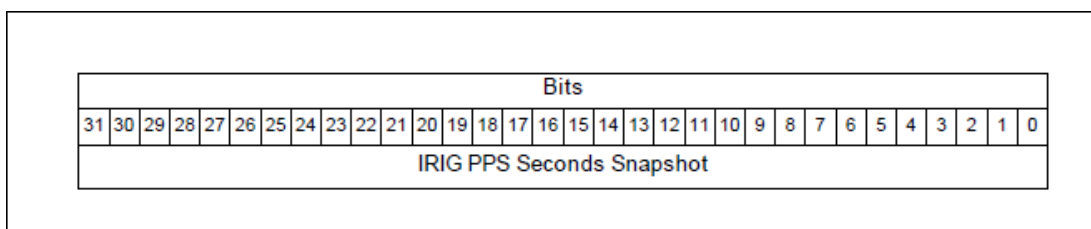
Offset: 00230



図A-12 RCIM IV IRIG PPS Seconds Snapshot Register

IRIG PPS Seconds Snapshot RegisterはPOSIXクロックの秒領域のスナップショットを含みます。スナップショットはIRIG PPSシグナルが発生する度に取り込まれます。

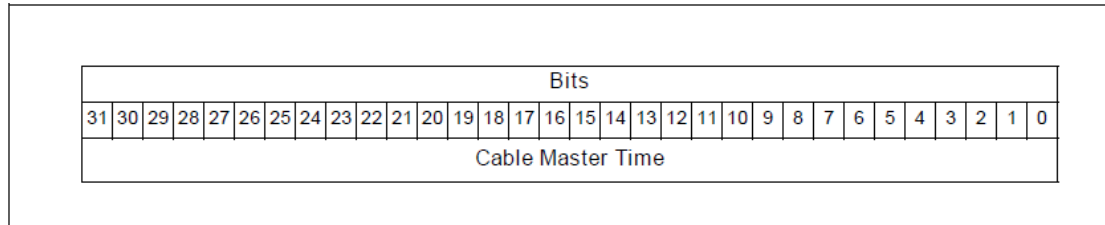
Offset: 00234



図A-13 RCIM IV Cable Master Time Register

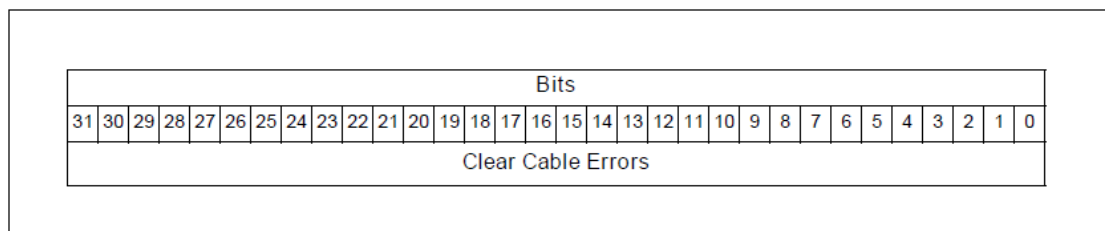
Cable Master Time Registerは秒の境界でのクロックの変化の度にケーブルに送信されたマスターRCIMのPOSIXクロックの秒領域を含みます。

Offset: 00220

**図A-14 RCIM IV Clear Cable Errors Register**

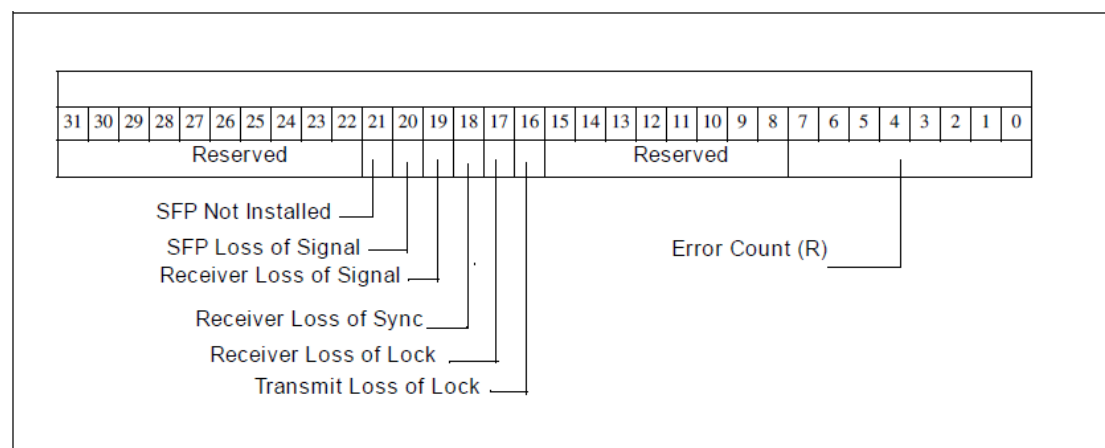
これは報告されたケーブル・エラーを消去する「書き込み専用」のレジスタです。データ領域は気にしないで下さい。

Offset: 00400

**図A-15 RCIM IV Output Cable Status Register**

本レジスタは出力ケーブルに関する詳細なハードウェアのステータス情報を提供します。

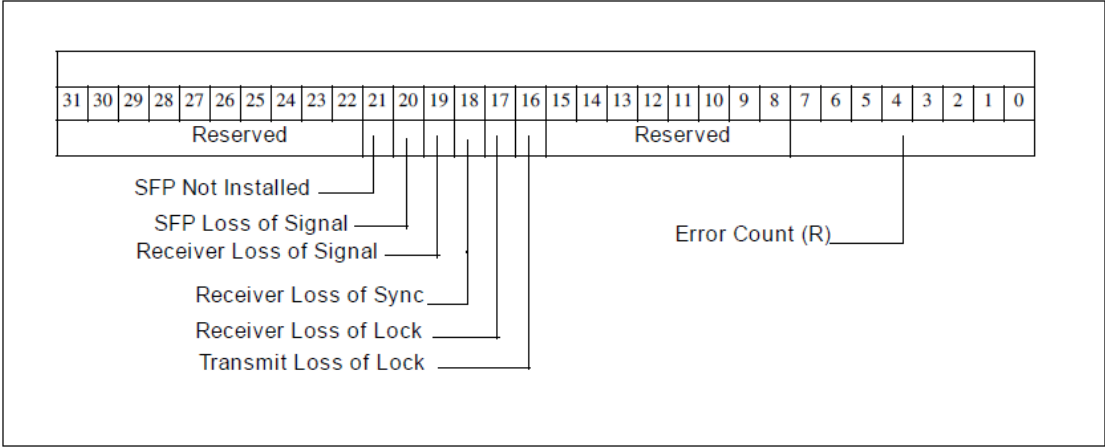
Offset: 00410



図A-16 RCIM IV Input Cable Status Register

本レジスタは入力ケーブルに関する詳細なハードウェアのステータス情報を提供します。

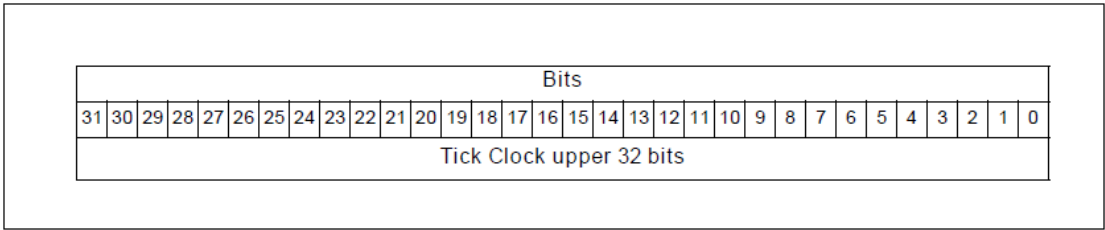
Offset: 00420



図A-17 RCIM IV Tick Clock Upper Register

本レジスタはティック・クロックの上位32bitを含みます。

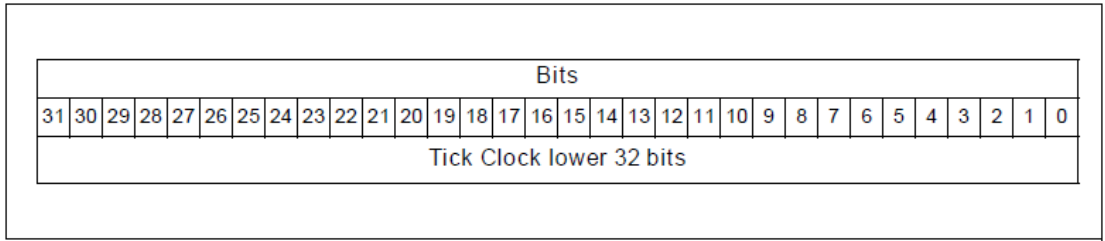
Offset: 01000, 10000



図A-18 RCIM IV Tick Clock Lower Register

本レジスタはティック・クロックの下位32bitを含みます。

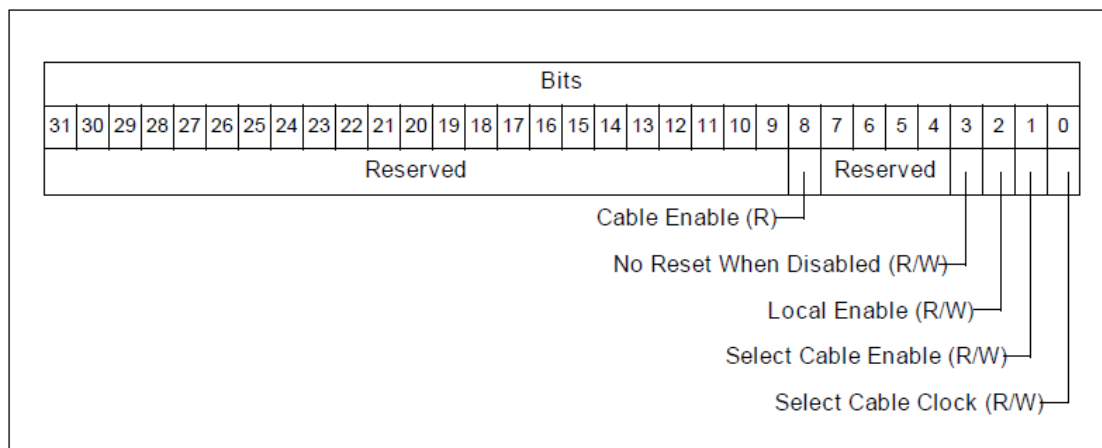
Offset: 01008, 10008



図A-19 RCIM IV Tick Clock Status/Control Register

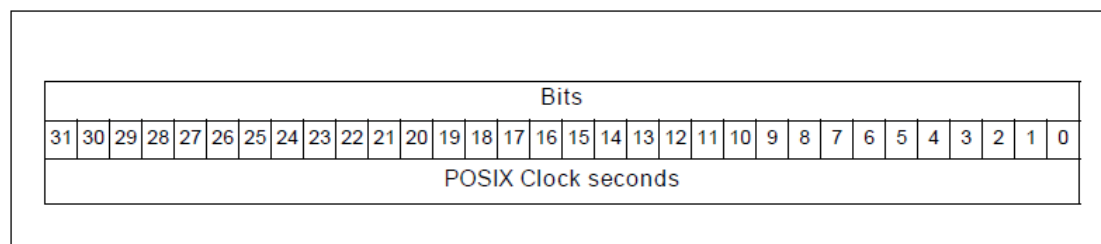
本レジスタはティック・クロックのステータスと制御を提供します。

Offset: 01010, 10010

**図A-20 RCIM IV POSIX Clock Seconds Register**

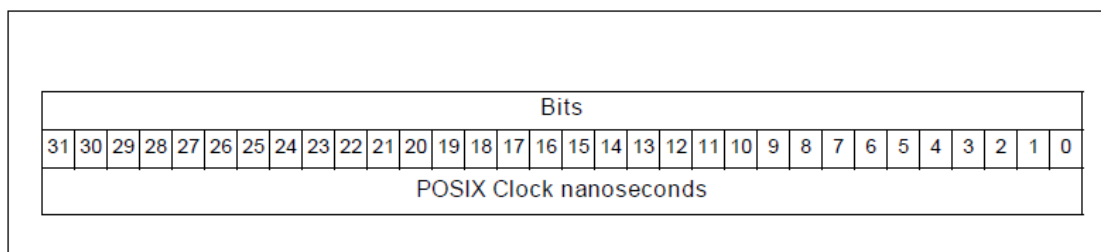
本レジスタはPOSIXクロックの秒を含みます。

Offset: 01100, 10100

**図A-21 RCIM IV POSIX Clock Nanoseconds Register**

本レジスタはPOSIXクロックのナノ秒を含みます。

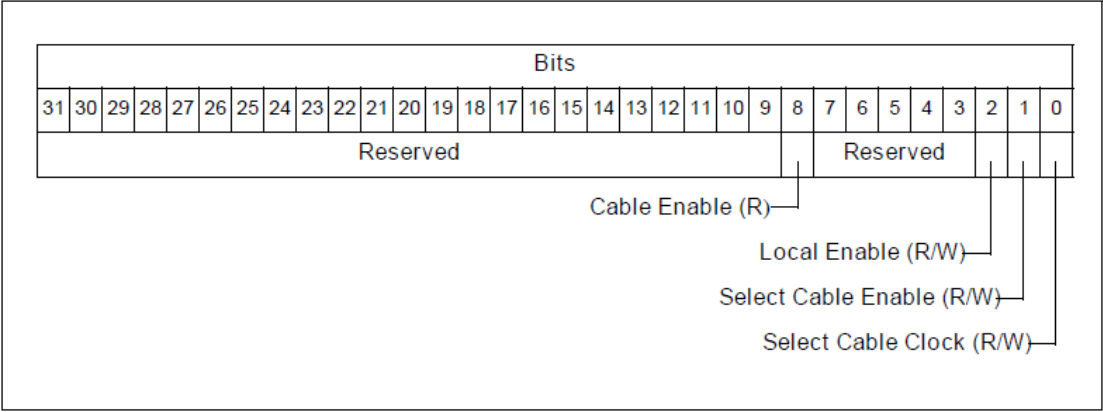
Offset: 01108, 10108



図A-22 RCIM IV POSIX Clock Status/Control Register

本レジスタはPOSIXクロックのステータスと制御を提供します。

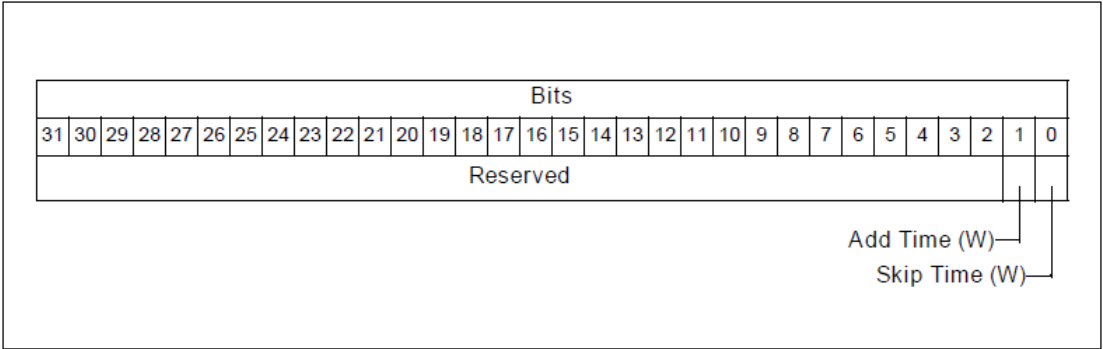
Offset: 01110, 10110



図A-23 RCIM IV POSIX Clock Skip/Add Time Register

本レジスタは400ナノ秒単位でPOSIXクロックの時間をスキップ/追加します。

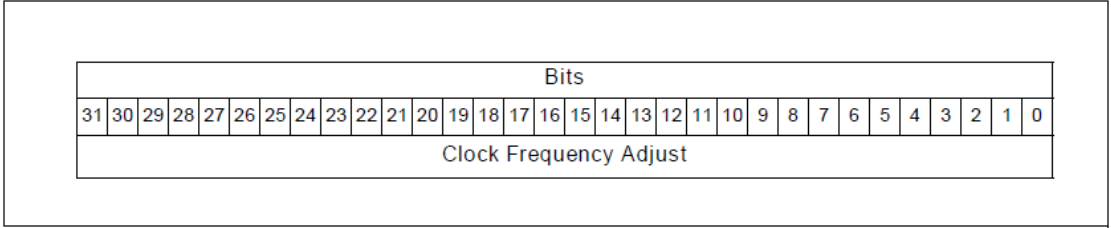
Offset: 01114, 10114



図A-24 RCIM IV Clock Frequency Adjust Register

Clock Frequency Adjust Registerは10 MHzのマスター・クロックの周波数を制御するために使
用します。

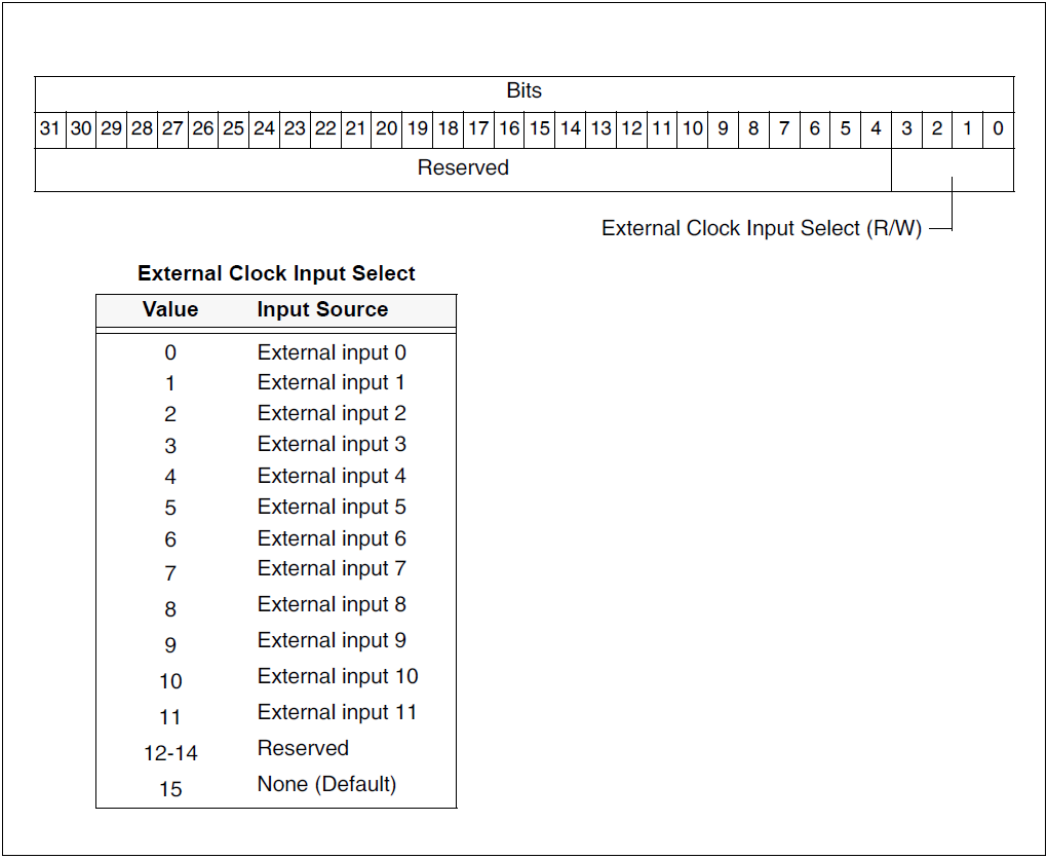
Offset: 01120, 10120



図A-25 RCIM IV External Clock Input Select Register

本レジスタはRCIMクロックを駆動する10MHz信号用に外部クロック入力ソースを選択しま
す。

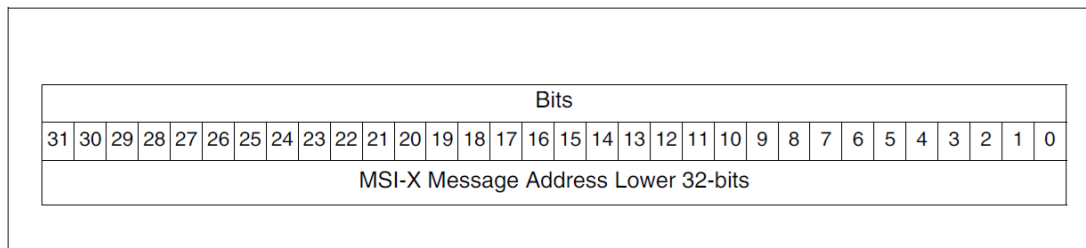
Offset: 01150



図A-26 RCIM IV MSI-X Lower Address N Register

各MSI-X割込みに関するメッセージ・アドレスの下位32bit。

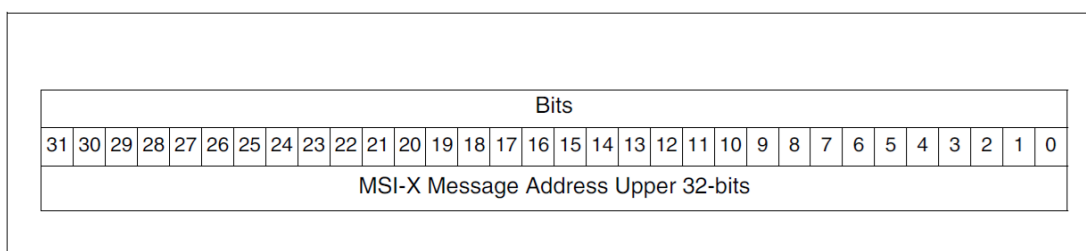
Offset: 1500, 1510, 1520, 1530, 1540, 1550, 1560, 1570, 1580, 1590, 15A0, 15B0, 15C0, 15D0, 15E0, 15F0



図A-27 RCIM IV MSI-X Upper Address N Register

各MSI-X割込みに関するメッセージ・アドレスの上位32bit。

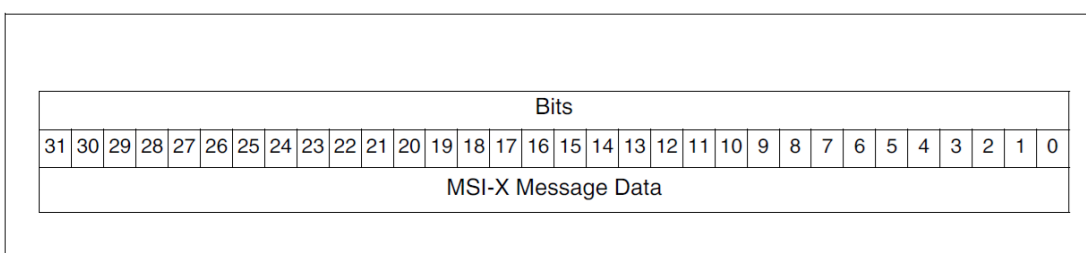
Offset: 1504, 1514, 1524, 1534, 1544, 1554, 1564, 1574, 1584, 1594, 15A4, 15B4, 15C4, 15D4, 15E4, 15F4



図A-28 RCIM IV MSI-X Message Data N Registers

各MSI-X割込みに関するメッセージ・データ。

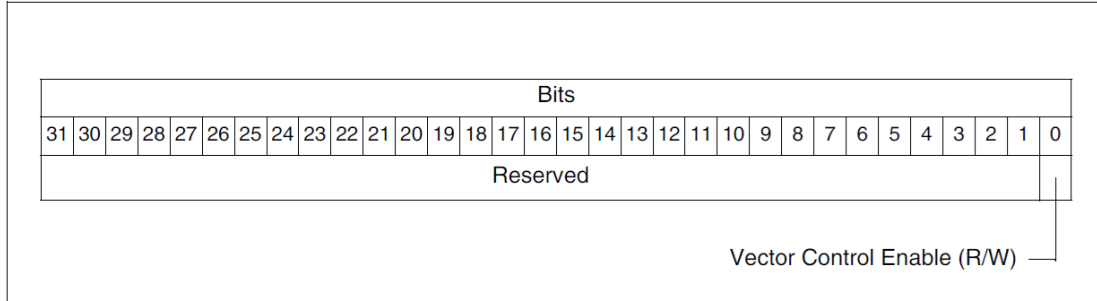
Offset: 1508, 1518, 1528, 1538, 1548, 1558, 1568, 1578, 1588, 1598, 15A8, 15B8, 15C8, 15D8, 15E8, 15F8



図A-29 RCIM IV MSI-X Vector Control N Registers

各MSI-X割込みに関するVector Control Register。

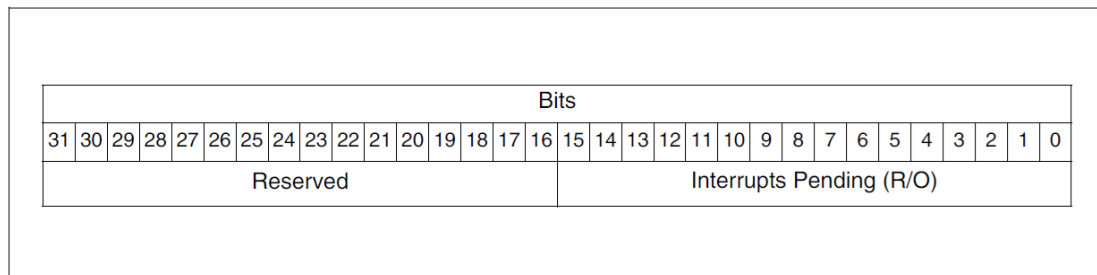
Offset: 150C, 151C, 152C, 153C, 154C, 155C, 156C, 157C, 158C, 159C, 15AC, 15BC, 15CC, 15DC, 15EC, 15FC



図A-30 RCIM IV MSI-X Pending Registers

MSI-X Pending Registerは現在保留中であるMSI-X割込み一式を示します。割込みが有効である場合、これらのビットは長い時間見える状態にはなりません。

Offset: 1600



図A-31 RCIM IV MSI-X Select #1 Register N

これらのレジスタは各MSI-X割込みに対し個々の割り込みを選択するために使用します。

Offset: 1700, 1708, 1710, 1718, 1720, 1728, 1730, 1738, 1740, 1748, 1750, 1758, 1760, 1768, 1770, 1778

これらのレジスタはInterrupt Enable Register #1と同じマッピングとなります。A-7ページにある図A-4「RCIM IV Interrupt Enable/Request/Pending/Clear/Arm/Level/Polarity Registers」を参照して下さい。

図A-32 RCIM IV MSI-X Select #2 Register N

これらのレジスタは各MSI-X割込みに対し個々の割り込みを選択するために使用します。

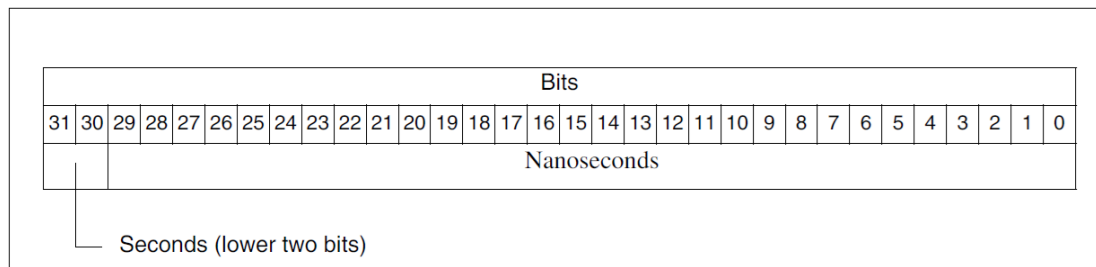
Offset: 1704, 170C, 1714, 171C, 1724, 172C, 1734, 173C, 1744, 174C, 1754, 175C, 1764, 176C, 1774, 177C

これらのレジスタはInterrupt Enable Register #2と同じマッピングとなります。A-7ページにある図A-4「RCIM IV Interrupt Enable/Request/Pending/Clear/Arm/Level/Polarity Registers」を参照して下さい。

図A-33 RCIM IV MSI-X Time #1 Register N

これらのレジスタは各MSI-X割込みが発生する度に取得されるPOSIXクロックのナノ秒領域に加えて秒領域の下位2bitのスナップショットを含みます。

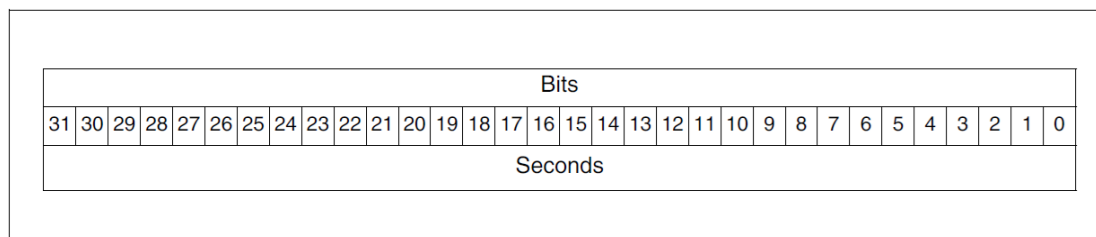
Offset: 1800, 1808, 1810, 1818, 1820, 1828, 1830, 1838, 1840, 1848, 1850, 1858, 1860, 1868, 1870, 1878



図A-34 RCIM IV MSI-X Time #2 Register N

これらのレジスタは各MSI-X割込みが発生する度に取得されるPOSIXクロックの秒領域のスナップショットを含みます。

Offset: 1804, 180C, 1814, 181C, 1824, 182C, 1834, 183C, 1844, 184C, 1854, 185C, 1864, 186C, 1874, 187C



図A-35 RCIM IV MSI-X Interrupt Clear Register #1 N

これらのレジスタは各MSI-X割込みを消去します。

Offset: 1900, 1904, 1908, 190C, 1910, 1914, 1918, 191C, 1920, 1924, 1928, 192C, 1930, 1934, 1938, 193C

これらのレジスタはInterrupt Clear Register #1と同じマッピングとなります。A-7ページにある図A-4「RCIM IV Interrupt Enable/Request/Pending/Clear/Arm/Level/Polarity Registers」を参照して下さい。

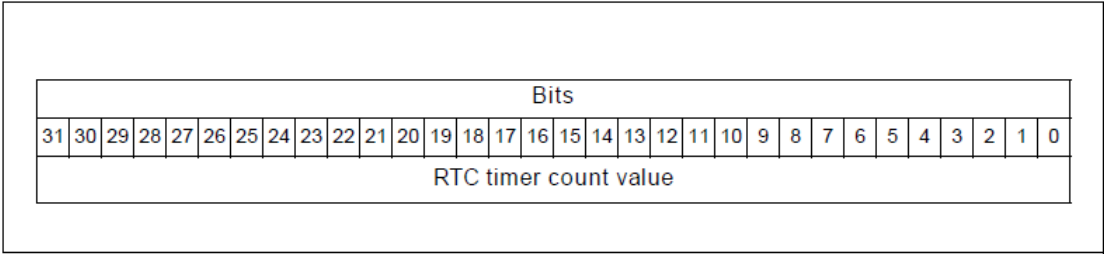
これは書き込み専用のレジスタです。レジスタへの書き込みは選択した割込みを消去します。

Note: これは全ての割り込みに対して使用される同じInterrupt Clear Registerです。個々のMSI-X割込みに関するアドレス・マッピングは、有効/無効bit書き込みを利用せずに新しい割込みを生成するために使用します。MSI-X割込みのためのInterrupt Clear Register #2への書き込みは、例えこの範囲内に選択された割込みがない場合であってもこのレジスタへの書き込み後に行う必要があります。

図A-36 RCIM IV RTC Timer Registers

初期のRTCタイマー値はRTC Timer Registerに取り込まれます。現在のタイマー値は本レジスタから読み込まれます。NOTE: 本レジスタの読み取りはRCIMとの互換性のためにRTC Repeat Registerも読み込まれます。

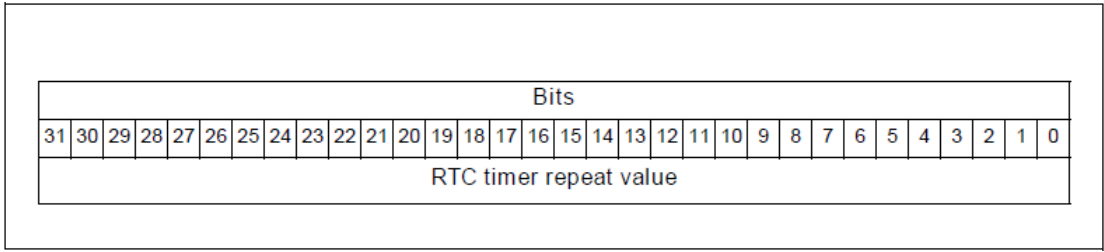
Offset: 02010, 02030, 02050, 02070, 02090, 020B0, 020D0, 020F0



図A-37 RCIM IV RTC Repeat Registers

RTC Repeat Registerはリピートのカウント値を含みます。

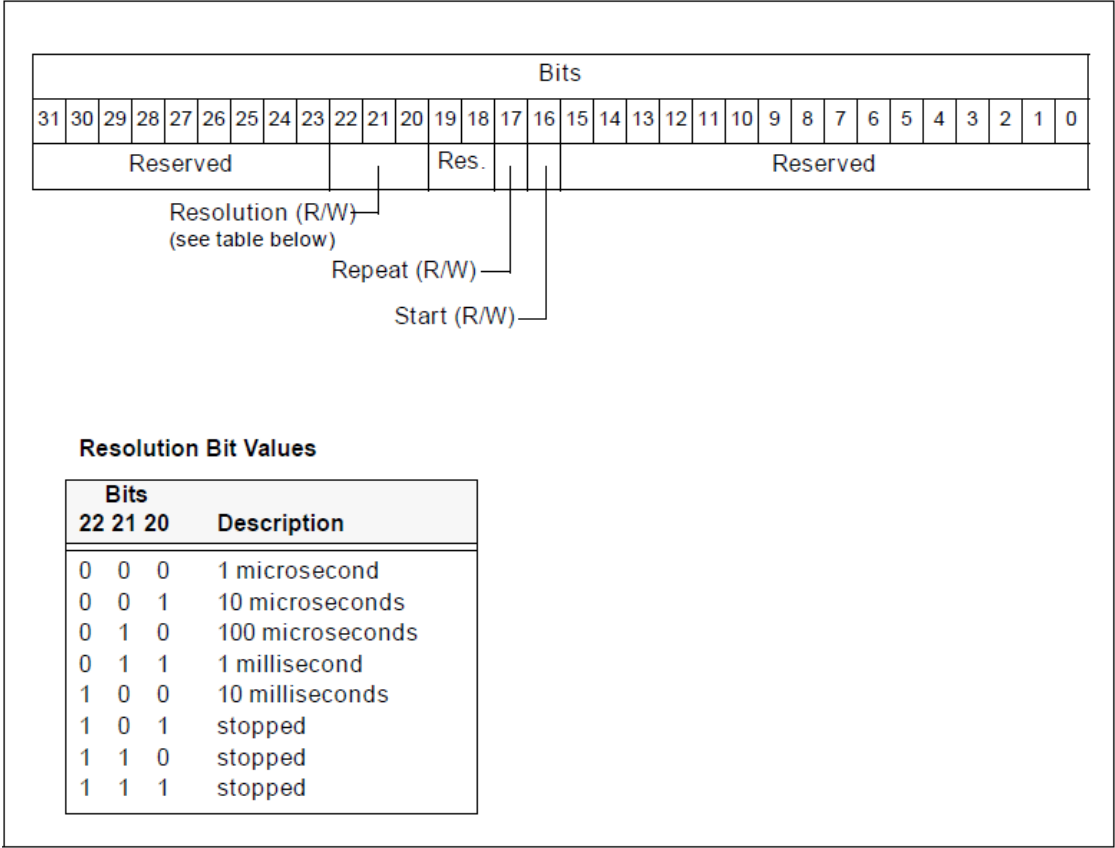
Offset: 02014, 02034, 02054, 02074, 02094, 020B4, 020D4, 020F4



図A-38 RCIM IV RTC Control Registers

本レジスタはRTCの制御を提供します。

Offset: 02000, 02020, 02040, 02060, 02080, 020A0, 020C0, 020E0



図A-39 RCIM IV Programmable Interrupt Generator Register

本レジスタはプログラマブル割込みを確認します。

Offset: 03000, 30000

Bits																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																				PIG											

図A-40 RCIM IV Programmable Interrupt Set and Clear Registers

これらのレジスタへの書き込みは他のビットに影響することなく Programmable Interrupt Registerの単一ビットを設定/消去します。

Offset: 03010, 30010, 03020, 30020

Bits																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																				PIG											

図A-41 RCIM IV External I/O Output Enable Register

本レジスタは選択されたExternal I/O Output Enableを設定またはリセットします。

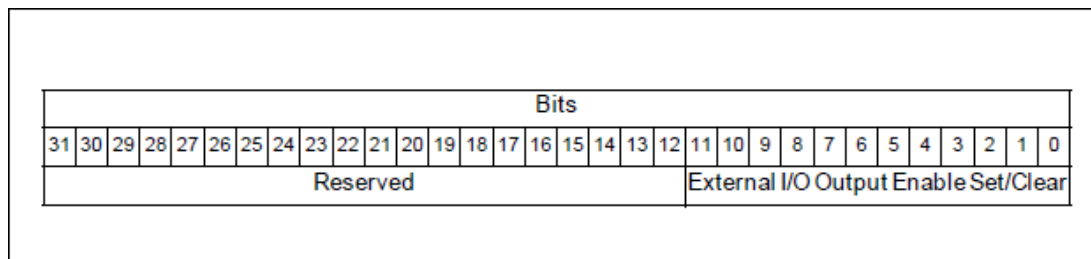
Offset: 03040

Bits																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																				External I/O Output Enable											

図A-42 RCIM IV External I/O Output Enable Set/Clear Registers

これらのレジスタへの書き込みは他のビットに影響を与えることなく External I/O Output Enableの単一ビットを設定/クリアします。

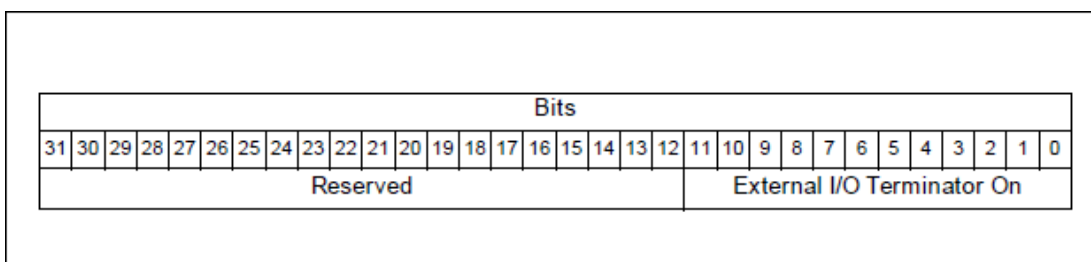
Offset: 03044, 03048



図A-43 RCIM IV External I/O Terminator On Register

本レジスタは選択されたExternal I/O Terminator Onを設定またはリセットします。

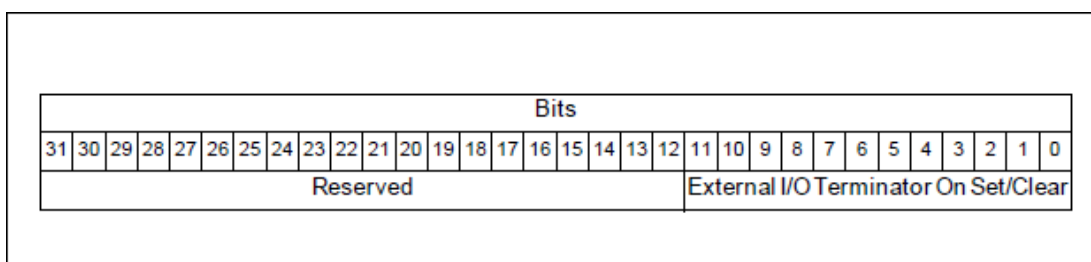
Offset: 03050



図A-44 RCIM IV External I/O Terminator On Set/Clear Registers

これらのレジスタへの書き込みは他のビットに影響を与えることなく External I/O Terminator On Registerの単一ビットを設定/クリアします。

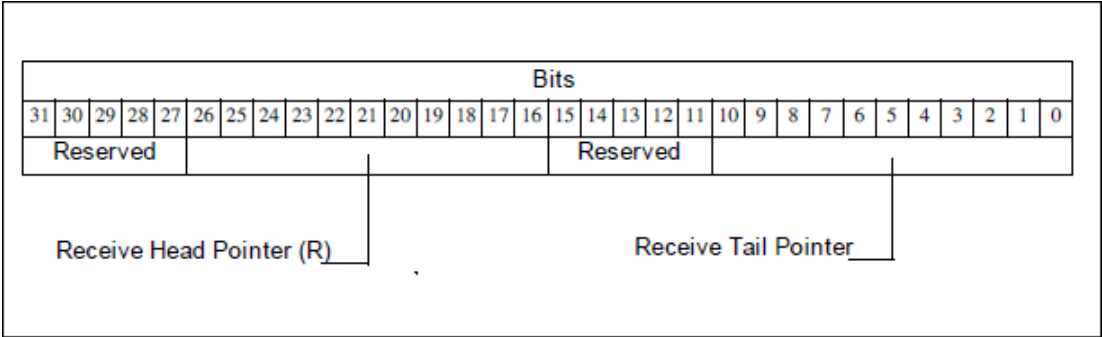
Offset: 03054, 03058



図A-45 RCIM IV GPS Receive Pointers

GPS Receive PointerはオプションのGPSモジュールとの通信で使用されます。

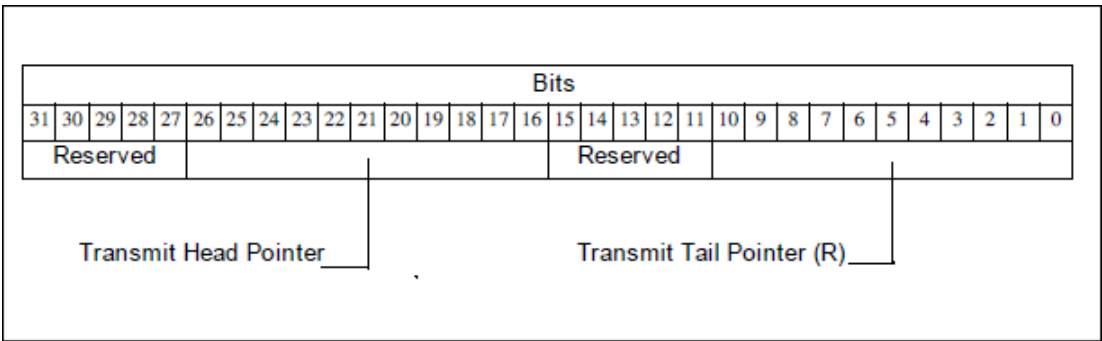
Offset: 03200



図A-46 RCIM IV GPS Transmit Pointers

GPS Transmit PointerはオプションのGPSモジュールとの通信で使用されます。

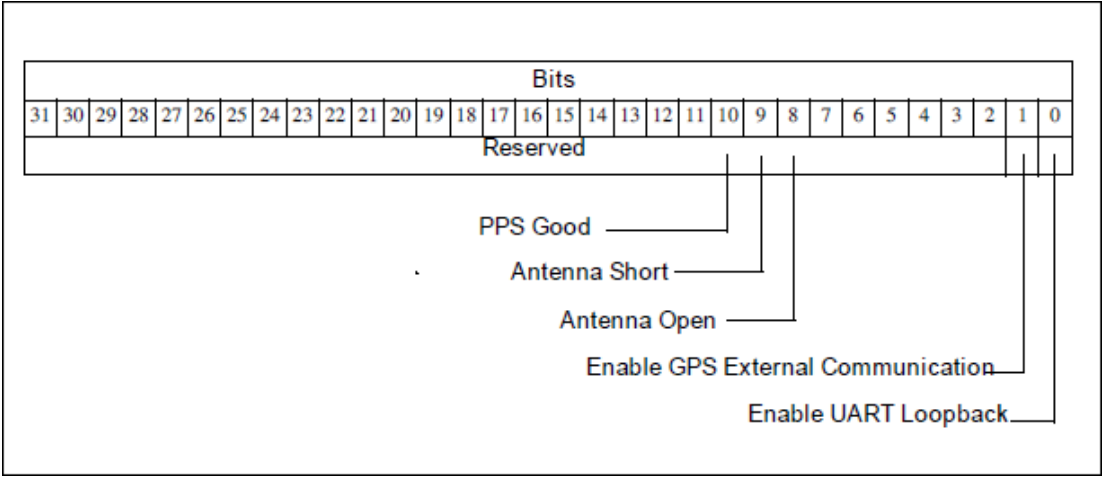
Offset: 03204



図A-47 RCIM IV GPS Debug Control/Status Register

GPS Debug Control/Status Registerはテストおよびデバッグ中に使用されるビットを含みます。これらのビットのいずれかを設定するとGPSモジュールとのRCIM通信が無効となります。

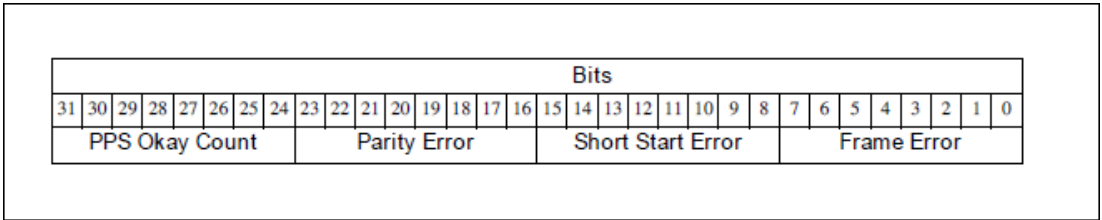
Offset: 03208



図A-48 RCIM IV GPS Communication Error Register

GPS Communication Error RegisterはGPSモジュールとの通信エラーに関する情報を含みます。このレジスタへの書き込みはGPSモジュールへの通信インターフェースをリセットします。

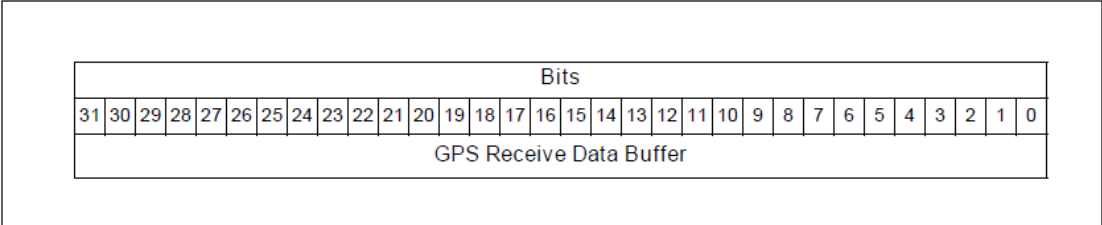
Offset: 0320C



図A-49 RCIM IV GPS Receive Data Buffer

これはGPSの受信データ領域です。

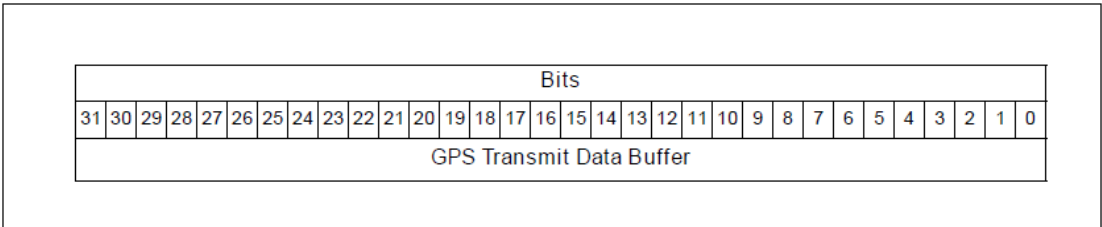
Offset: 04000 to 047FF



図A-50 RCIM IV GPS Transmit Data Buffer

これはGPSの転送データ領域です。

Offset: 04800 to 04FFF

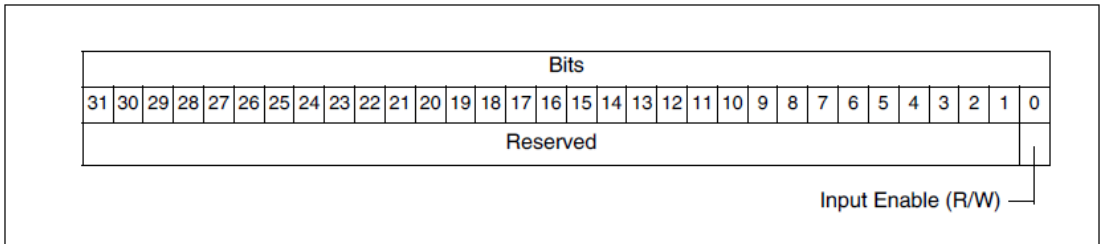


図A-51 RCIM IV IRIG Input Enable Register

IRIG入力信号を有効化するか無効化(無視)するかを制御します。

Enableビットを切り替えると入力インタフェースは全てのプログラム化情報を消去してリセット状態に戻ります。

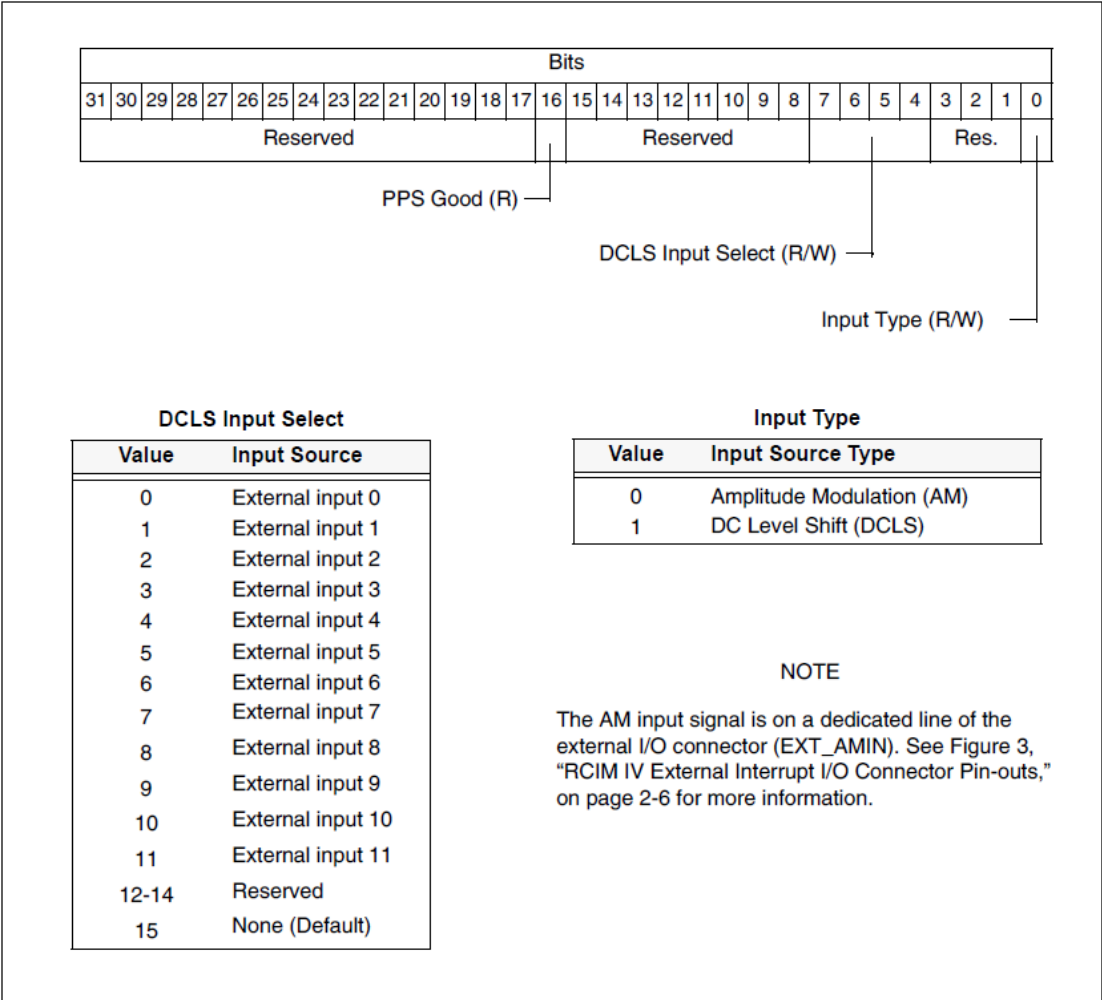
Offset: 06000



図A-52 RCIM IV IRIG Input Control Register

IRIG Input Control RegisterはIRIG入力ソースの属性を構成するために使用します。

Offset: 06004



図A-53 RCIM IV IRIG Input Status Register

IRIG AM入力タイプが選択された場合にIRIG AM波形の最大値を決定します。

Offset: 06008

Bits																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																AM High Maximum (R)															

Multiply the AM High Maximum value by 375 microvolts to determine the peak value of the IRIG AM waveform.

図A-54 RCIM IV IRIG Input Error Register

IRIG Input Error RegisterはIRIGモジュールとの通信エラーに関する情報を含んでいます。IRIG入力が有効化された場合、全てのカウントがゼロにリセットされます。

Offset: 0600C

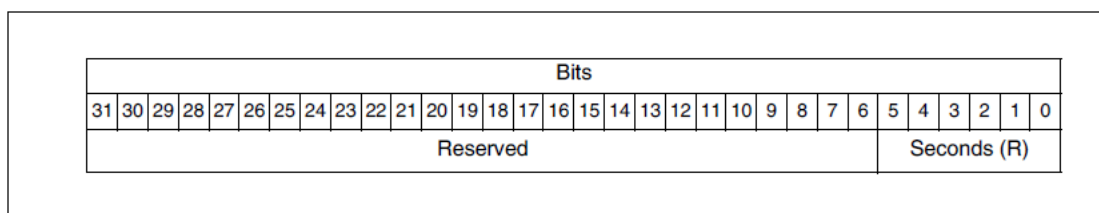
Bits																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PPS Okay Count (R)								AM Frame Error (R)								DCLS Frame Error (R)								Sync Error (R)							

PPS Okay Count	The number of times the IRIG PPS signal has indicated it is OK. This number should not change if the IRIG input data is reliable.
AM Frame Error	The number of AM framing errors that have occurred. Framing errors may occur if the source transmission stops and restarts, but should not occur when transmission is stable.
DCLS Frame Error	The number of DCLS framing errors that have occurred. Framing errors may occur if the source transmission stops and restarts, but should not occur once transmission is stable.
Sync Error	The number of sync errors that have occurred. Sync errors may occur if the source transmission stops and restarts, but should not occur once the transmission is stable.

図A-55 RCIM IV IRIG Input Seconds Register

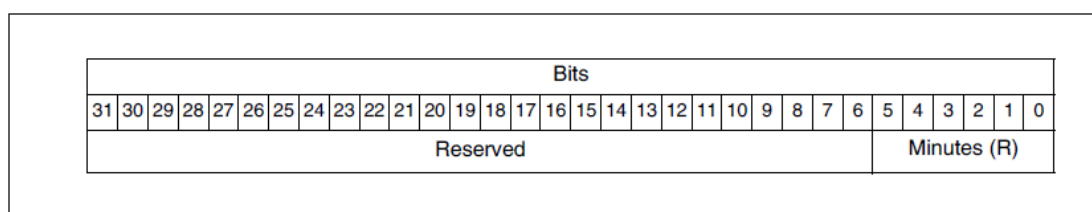
本レジスタはIRIG入力データの秒(second)の領域を含んでいます。

Offset: 06020

**図A-56 RCIM IV IRIG Input Minutes Register**

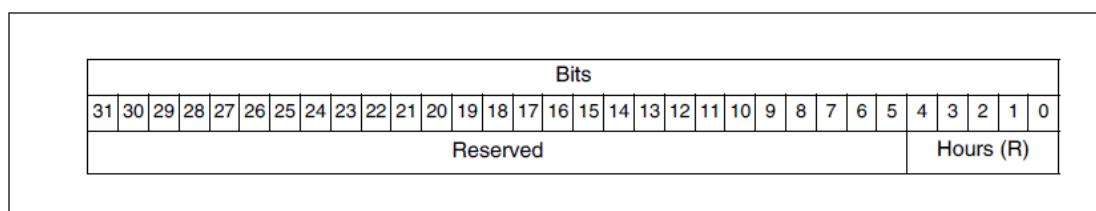
本レジスタはIRIG入力データの分(minute)の領域を含んでいます。

Offset: 06024

**図A-57 RCIM IV IRIG Input Hours Register**

本レジスタはIRIG入力データの時(hour)の領域を含んでいます。

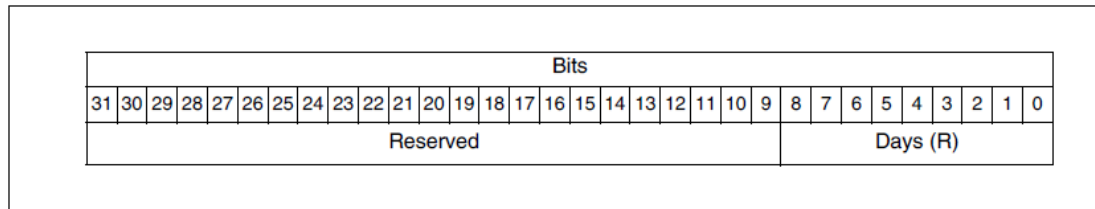
Offset: 06028



図A-58 RCIM IV IRIG Input Days Register

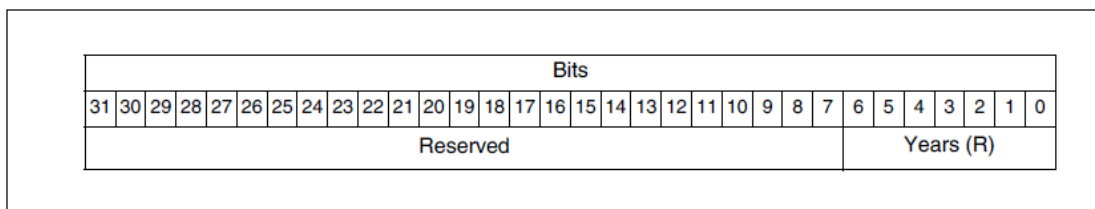
本レジスタはIRIG入力データの日(day)の領域を含んでいます。

Offset: 0602C

**図A-59 RCIM IV IRIG Input Years Register**

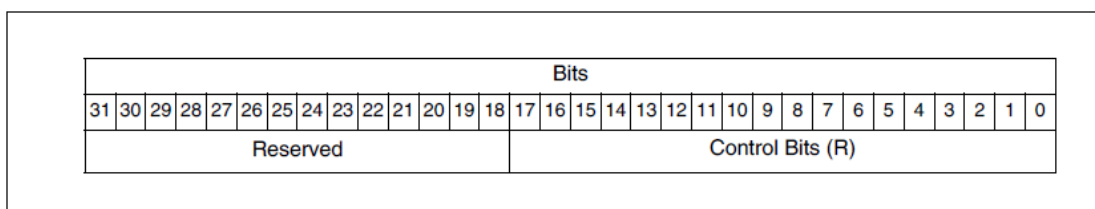
本レジスタはIRIG入力データの年(year)の領域を含んでいます。

Offset: 06030

**図A-60 RCIM IV IRIG Input Control Bits Register**

本レジスタはIRIG入力データの制御ビットの領域を含んでいます。

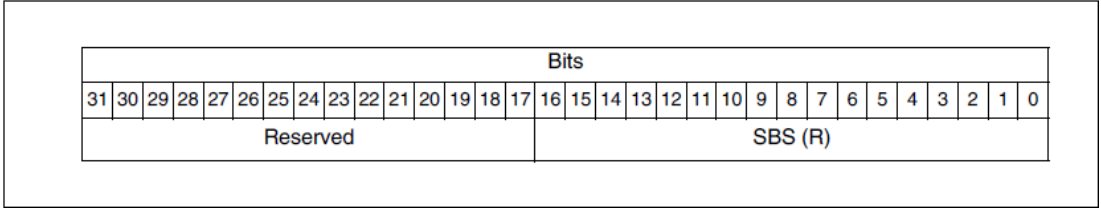
Offset: 06034



図A-61 RCIM IV IRIG Input SBS Register

本レジスタはIRIG入力データのStraight Binary Seconds (SBS)の領域に1日の秒を含んでいます。

Offset: 06038

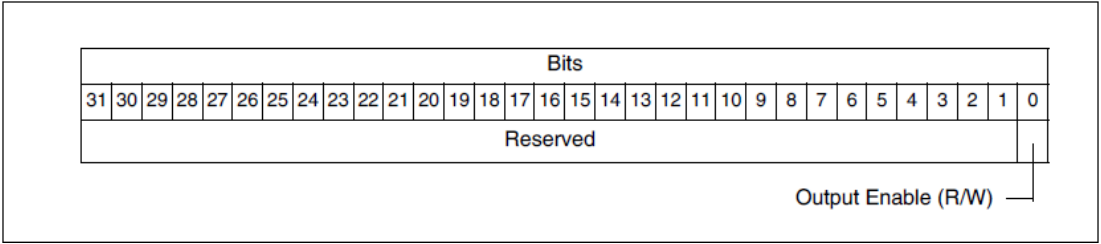


図A-62 RCIM IV IRIG Output Enable Register

IRIG出力信号を有効化するか無効化するかを制御します。

Enableビットを切り替えると出力インタフェースは全てのプログラム化情報を消去してリセット状態に戻ります。

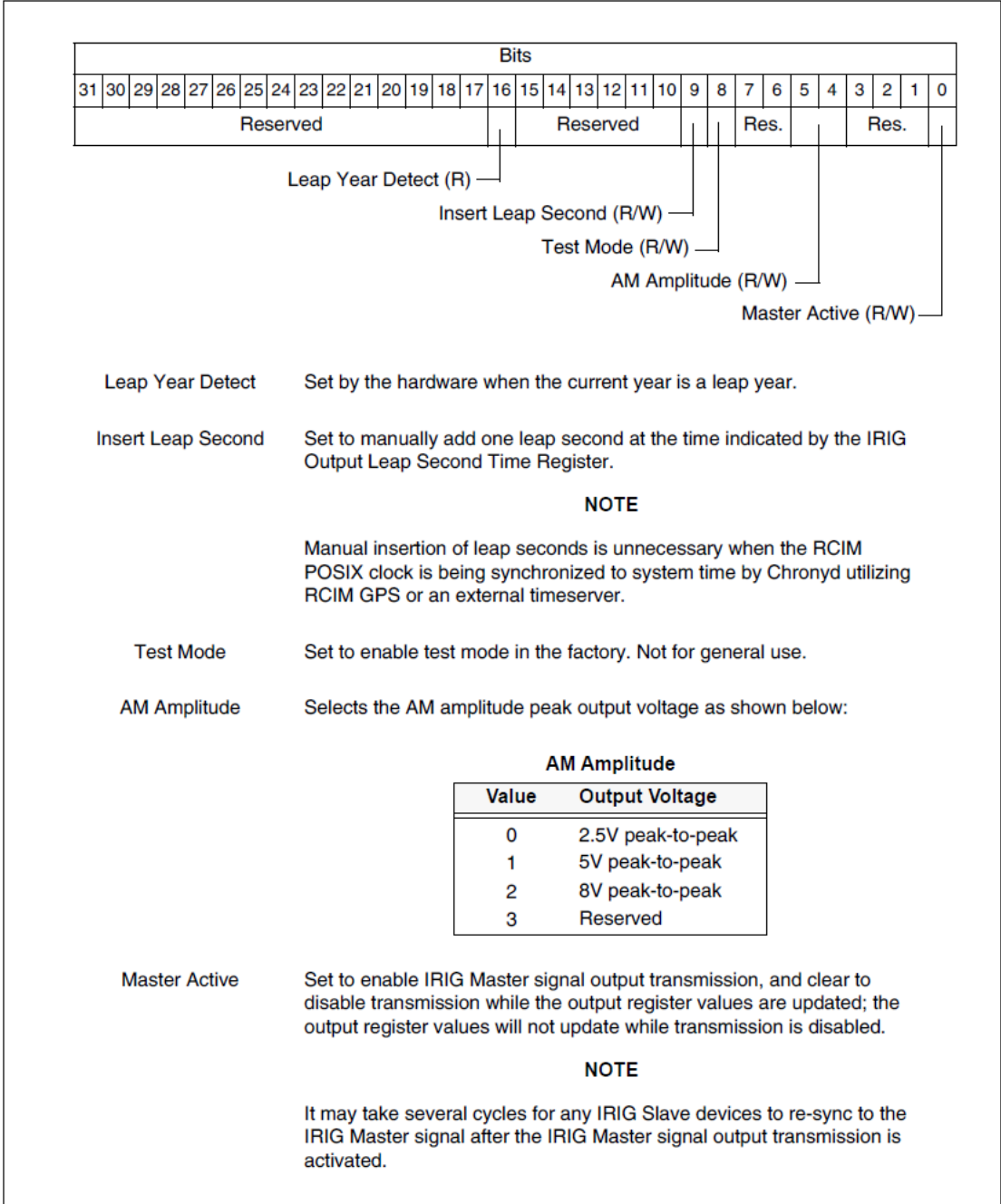
Offset: 06060



図A-63 RCIM IV IRIG Output Control Register

IRIG Output Control RegisterはIRIG出力ソースの属性を構成するために使用します。

Offset: 06064



図A-64 RCIM IV IRIG Output Adjust Register

本レジスタは様々なタイミング遅延を補正するために使用されるIRIG出力オフセット調整を保持します。

Offset: 06068

Bits																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																Output Adjust (R/W)															

Output Adjust

This value can be used to offset the IRIG output timing to compensate for delays due to AM modulation/demodulation and cabling. The offset is in the negative direction causing the IRIG output data to be sent earlier.

The formula for the compensation value is:

$$\text{Output Adjust} = (\text{desired offset in nanoseconds} / 400) * 25.$$

The maximum offset is 65.52 microseconds in 400 nanosecond increments.

図A-65 RCIM IV IRIG Output Leap Second Time Register

本レジスタはIRIG出力データのうるう秒挿入日時の領域を含んでいます。

Offset: 0606C

Bits																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R.	Leap Second Year (R/W)								Reserved								Leap Second Day (R/W)								Res.		Leap Second Hour (R/W)				

Leap Second Year The year that the leap second will be inserted.

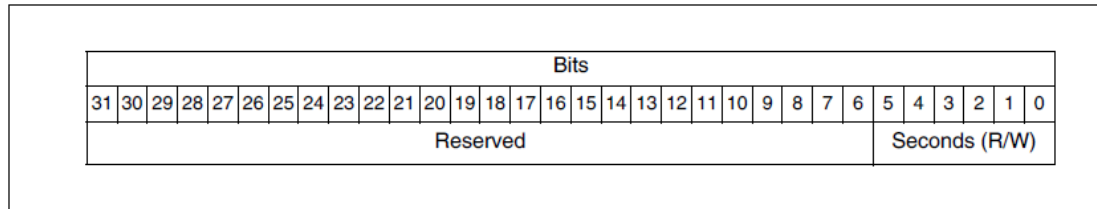
Leap Second Day The day that the leap second will be inserted.

Leap Second Hour The hour that the leap second will be inserted.

図A-66 RCIM IV IRIG Output Seconds Register

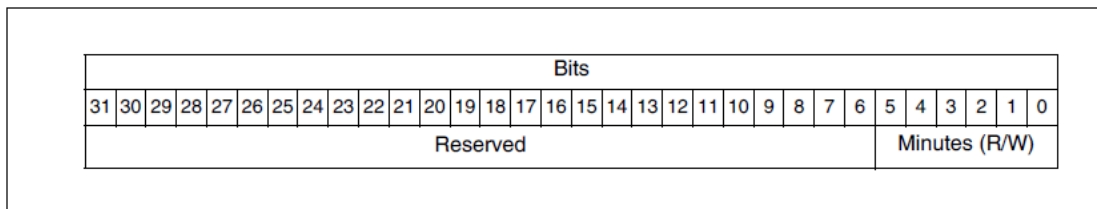
本レジスタはIRIG出力データの秒(second)の領域を含んでいます。

Offset: 06080

**図A-67 RCIM IV IRIG Output Minutes Register**

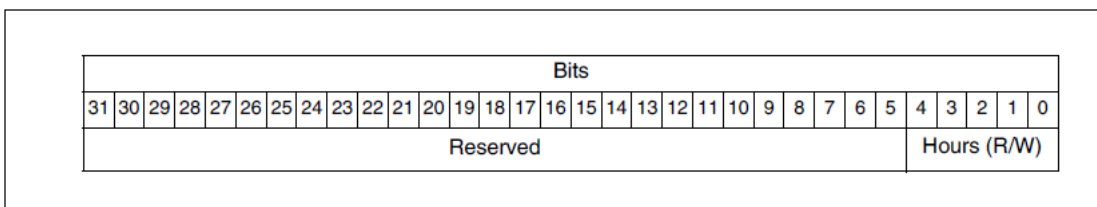
本レジスタはIRIG出力データの分(minute)の領域を含んでいます。

Offset: 06084

**図A-68 RCIM IV IRIG Output Hours Register**

本レジスタはIRIG出力データの時(hour)の領域を含んでいます。

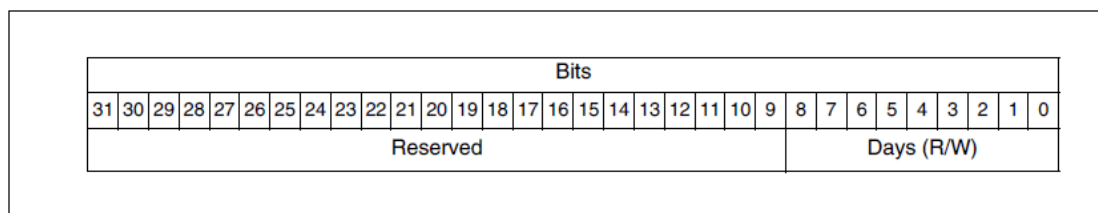
Offset: 06088



図A-69 RCIM IV IRIG Output Days Register

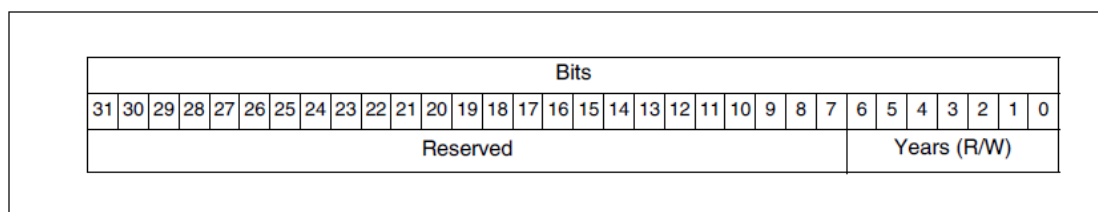
本レジスタはIRIG出力データの日(day)の領域を含んでいます。

Offset: 0608C

**図A-70 RCIM IV IRIG Output Years Register**

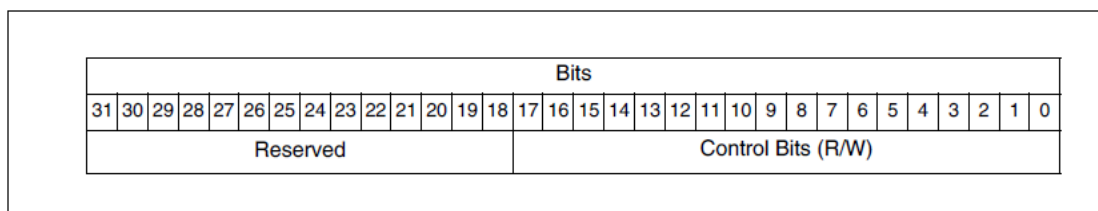
本レジスタはIRIG出力データの年(year)の領域を含んでいます。

Offset: 06090

**図A-71 RCIM IV IRIG Output Control Bits Register**

本レジスタはIRIG出力データの制御ビットの領域を含んでいます。

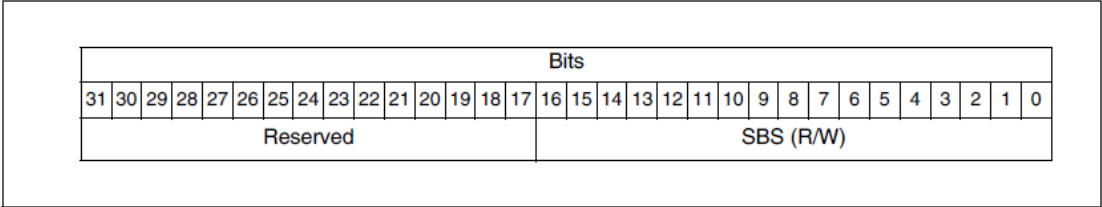
Offset: 06094



図A-72 RCIM IV IRIG Output SBS Register

本レジスタはIRIG出力データのStraight Binary Seconds (SBS)の領域に1日の秒を含んでいます。

Offset: 06098



B RCIM IIIレジスタ

本項にはRCIM IIIボードのアドレス・マップとレジスタが含まれます。

一部のレジスタは物理アドレス空間の2つの場所に現れる事に注意して下さい。これらのレジスタについては、関連する*Xregister*があります。例えば、PCSATとXPCSATです。*Xregister*は古い4kページ・サイズではなく64kのシステムに対応します。

RCIM IIIアドレス・マップ

Address	Function	
0xXXX00000	Board Status/Control Register	
0xXXX00004	Firmware Rev/Options Present Register	
0xXXX00010	Interrupt Enable Register #1	
0xXXX00014	Interrupt Enable Register #2	
0xXXX00020	Interrupt Request Register #1	(Write Only)
0xXXX00024	Interrupt Request Register #2	(Write Only)
0xXXX00020	Interrupt Pending Register #1	(Read Only)
0xXXX00024	Interrupt Pending Register #2	(Read Only)
0xXXX00030	Interrupt Clear Register #1	
0xXXX00034	Interrupt Clear Register #2	
0xXXX00040	Interrupt Arm Register #1	
0xXXX00044	Interrupt Arm Register #2	
0xXXX00050	Interrupt Select Level Register #1	
0xXXX00054	Interrupt Select Level Register #2	
0xXXX00060	Interrupt Select Polarity Register #1	
0xXXX00064	Interrupt Select Polarity Register #2	
0xXXX00070	External Interrupt Routing Register #1	
0xXXX00074	External Interrupt Routing Register #2	
0xXXX00078	External Interrupt Routing Register #3	
0xXXX00080	Cable Interrupt Routing Register #1	
0xXXX00084	Cable Interrupt Routing Register #2	
0xXXX00088	Cable Interrupt Routing Register #3	
0xXXX00200	PPS Snapshot Register	(Read Only)
0xXXX00210	Cable Snapshot Register	(Read Only)
0xXXX00220	Cable Master Time Register	(Read Only)
0xXXX00400	Clear Cable Errors	(Write Only)
0xXXX00410	Output Cable Status Register	(Read Only)
0xXXX00420	Input Cable Status Register	(Read Only)
0xXXX01000	Tick Clock Upper	
0xXXX10000	Tick Clock Lower	
0xXXX01008	Tick Clock Lower	
0xXXX10008	Tick Clock Lower	

Address	Function	
0xFFFF01010	Tick Clock Status/Control	
0xFFFF10010	Tick Clock Status/Control	
0xFFFF01100	POSIX Clock Seconds	
0xFFFF10100	POSIX Clock Seconds	
0xFFFF01108	POSIX Clock Nanoseconds	
0xFFFF10108	POSIX Clock Nanoseconds	
0xFFFF01110	POSIX Clock Status/Control	
0xFFFF10110	POSIX Clock Status/Control	
0xFFFF01114	POSIX Clock Skip/Add Time	(Write Only)
0xFFFF10114	POSIX Clock Skip/Add Time	(Write Only)
0xFFFF01120	Clock Frequency Adjust Register	
0xFFFF10120	Clock Frequency Adjust Register	
0xFFFF02000	RTC #0 Control	
0xFFFF02010	RTC #0 Timer	
0xFFFF02014	RTC #0 Repeat	
0xFFFF02020	RTC #1 Control	
0xFFFF02030	RTC #1 Timer	
0xFFFF02034	RTC #1 Repeat	
0xFFFF02040	RTC #2 Control	
0xFFFF02050	RTC #2 Timer	
0xFFFF02054	RTC #2 Repeat	
0xFFFF02060	RTC #3 Control	
0xFFFF02070	RTC #3 Timer	
0xFFFF02074	RTC #3 Repeat	
0xFFFF02080	RTC #4 Control	
0xFFFF02090	RTC #4 Timer	
0xFFFF02094	RTC #4 Repeat	
0xFFFF020A0	RTC #5 Control	
0xFFFF020B0	RTC #5 Timer	
0xFFFF020B4	RTC #5 Repeat	
0xFFFF020C0	RTC #6 Control	
0xFFFF020D0	RTC #6 Timer	
0xFFFF020D4	RTC #6 Repeat	
0xFFFF020E0	RTC #7 Control	
0xFFFF020F0	RTC #7 Timer	
0xFFFF020F4	RTC #7 Repeat	
0xFFFF03000	Programmable Interrupt Generator	
0xFFFF30000	Programmable Interrupt Generator	
0xFFFF03010	Programmable INTR Generator Set	(Write Only)
0xFFFF30010	Programmable INTR Generator Set	(Write Only)
0xFFFF03020	Programmable INTR Generator Clear	(Write Only)
0xFFFF30020	Programmable INTR Generator Clear	(Write Only)
0xFFFF03100	SPI Count Register	
0xFFFF03200	GPS Receive Pointers	
0xFFFF03204	GPS Transmit Pointers	
0xFFFF03208	GPS Debug Control Register	

Address	Function	
0xFFFF0320C	GPS Communication Error Register	
0xFFFF03800- 0xFFFF03FFF	SPI Data Buffer	
0xFFFF04000- 0xFFFF047FF	GPS Receive Data Buffer	
0xFFFF04800- 0xFFFF04FFF	GPS Transmit Data Buffer	

RCIM IIIレジスタ

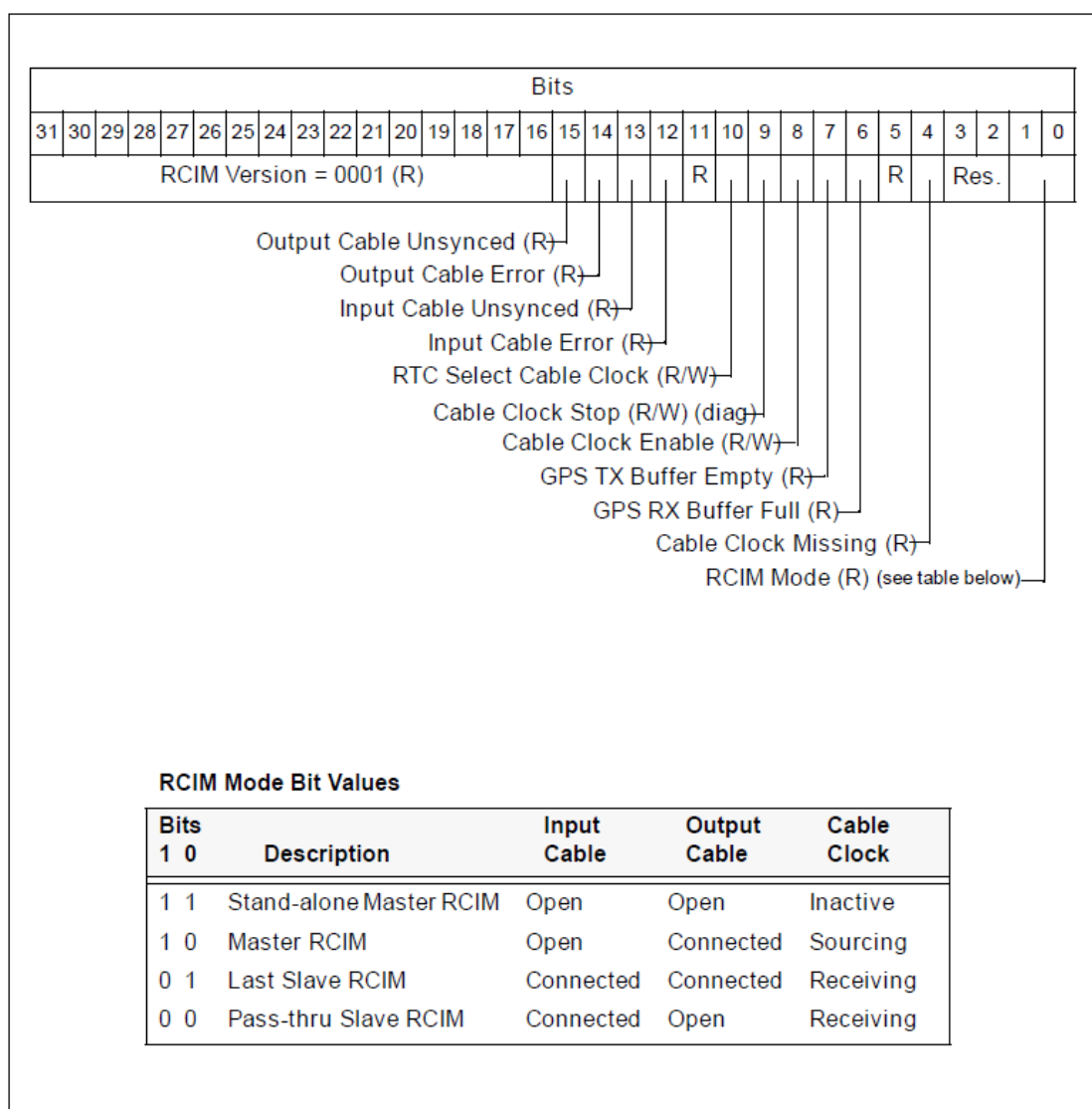
RCIM IIIレジスタを本章内で図解します。

NOTE: 特に指定がない限り、ビットの値は1=on; 0=offとなります

図B-1 RCIM III Board Status/Control Register

本レジスタはRCIM IIIボードの特定の機能のステータスと制御を提供します。

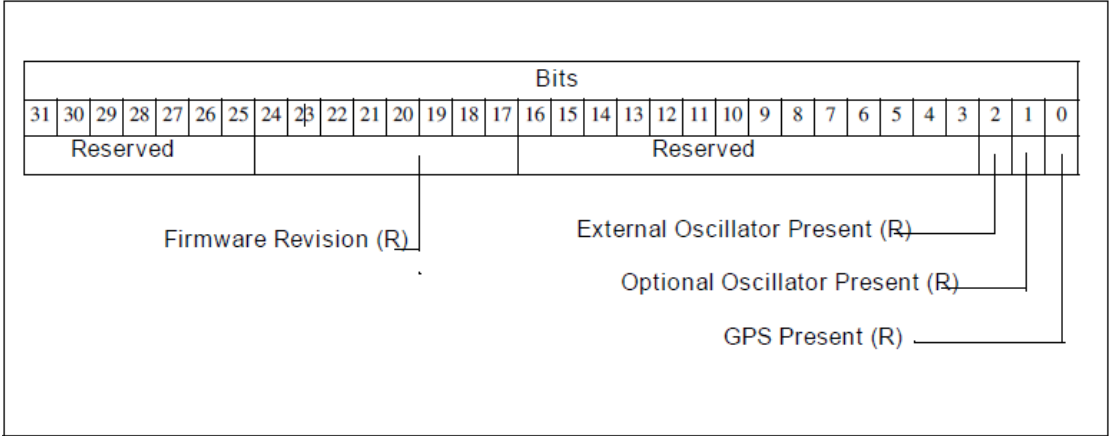
Offset: 00000



図B-2 RCIM III Firmware Revision/Options Present Register

本レジスタはこのRCIMボードに存在するオプションとファームウェアのリビジョンに関する情報を提供します。

Offset: 00004



図B-3 RCIM III Interrupt Enable/Request/Pending/Clear/Arm/Level/Polarity Registers

Enable Registersは選択された割込みを有効にします。

Request Registersは選択された割込みのソフトウェア駆動要求です。

Pending Registersは要求を保留します。

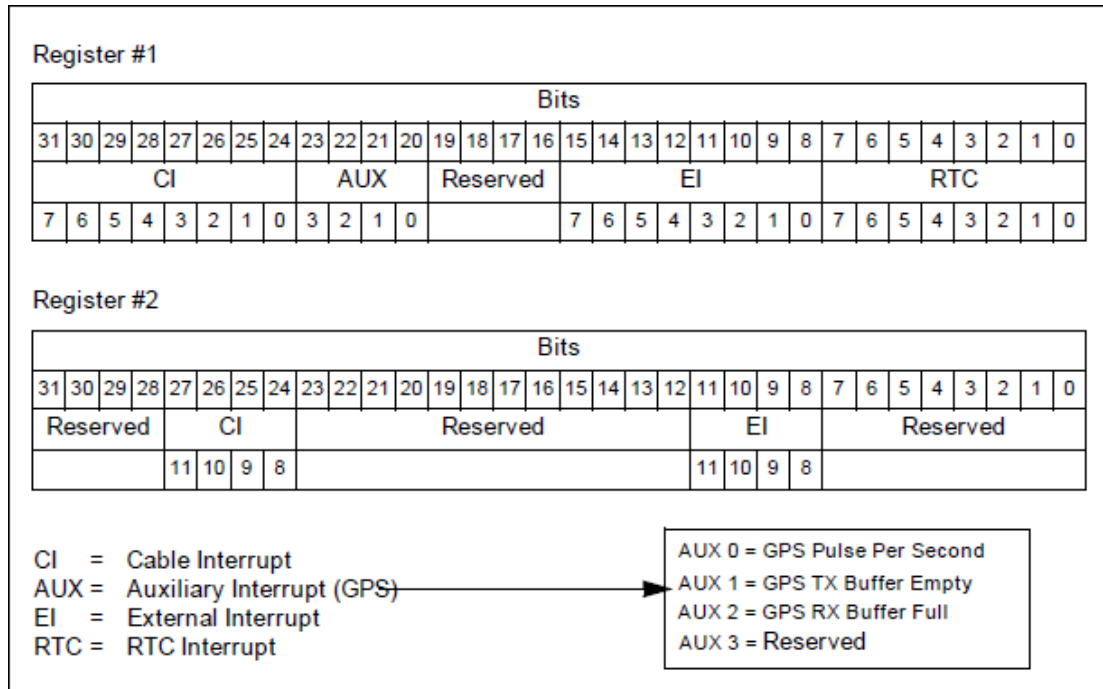
Clear Registersは選択された割込みを解放します。

Arm Registersはエッジ・トリガ用に選択された割込みを実装します。

Level Registersは選択された割込みに対してレベル(1)またはエッジ(0)を設定します。

Polarity Registersは選択された割込みに対してハイ(1)またはロー(0)の極性を設定します。

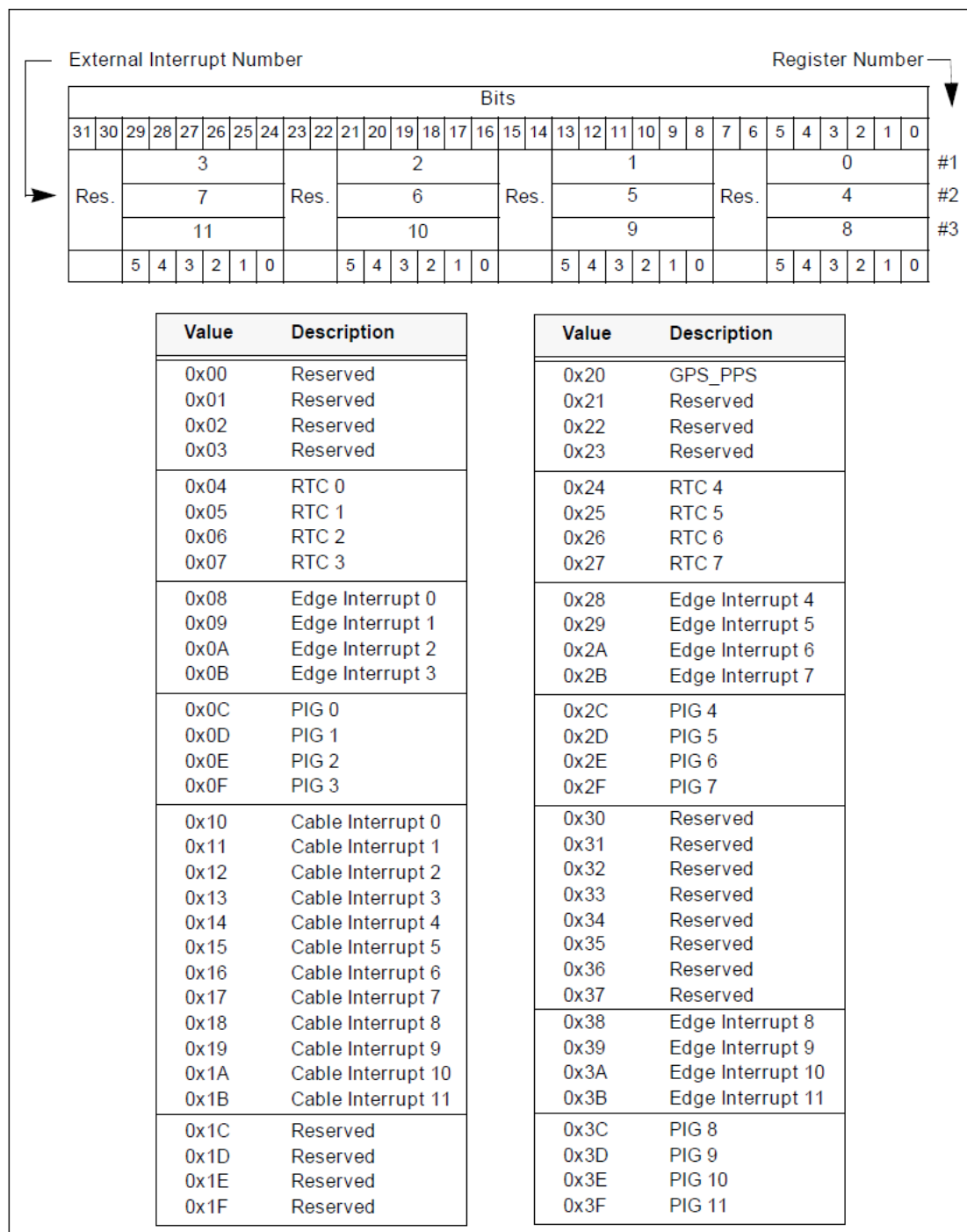
Offset: 00010, 00014, 00020, 00024, 00030, 00034, 00040, 00044, 00050, 00054, 00060, 00064



図B-4 RCIM III External Interrupt Routing Registers

External Interrupt Routing Registersは選択された割込みを外部割込みコネクタへのルートに設定します。

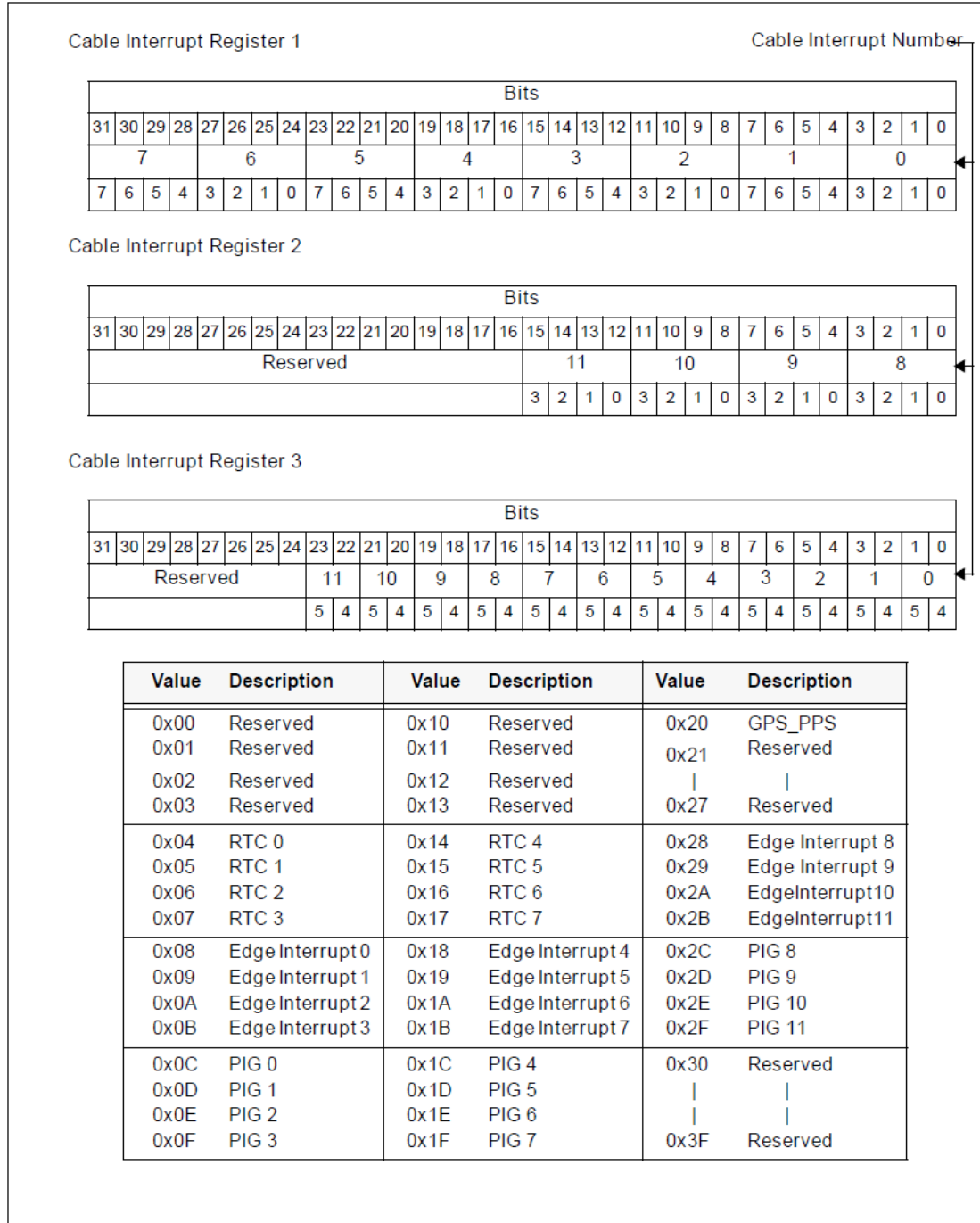
Offset: 00070, 00074, 00078



図B-5 RCIM III Cable Interrupt Routing Registers

Cable Interrupt Routing Registersは選択された割込みをRCIM相互接続ケーブルへのルートに設定します。

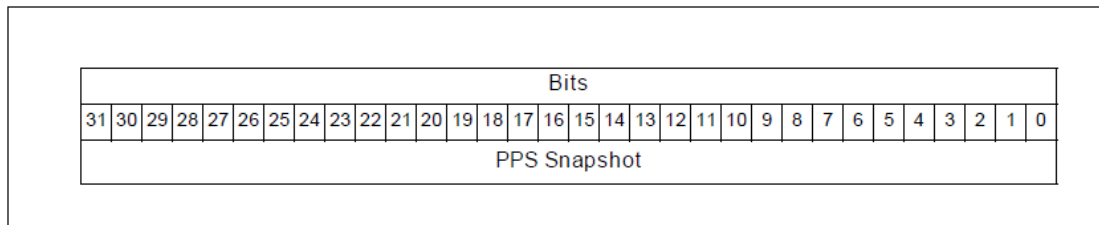
Offset: 00080, 00084, 00088



図B-6 RCIM III PPS Snapshot Register

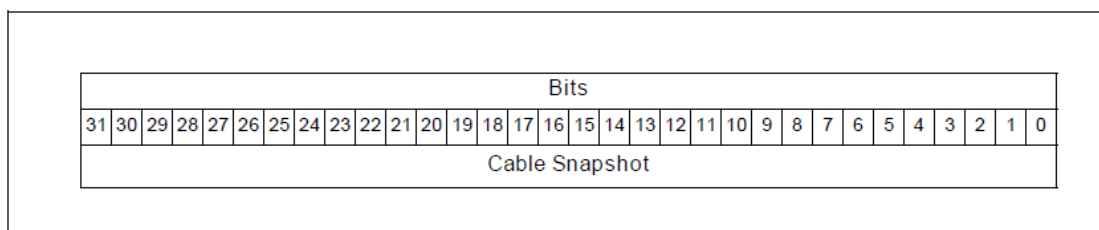
PPS Snapshot RegisterはPOSIXクロックのナノ秒領域と秒領域の2つのbitのスナップショットを含みます。スナップショットはGPS PPSシグナルが発生する度に取り込まれます。

Offset: 00200

**図B-7 RCIM III Cable Snapshot Register**

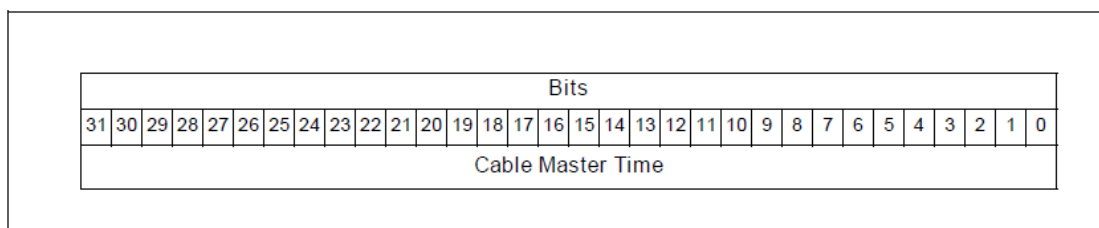
Cable Snapshot RegisterはPOSIXクロックのナノ秒領域と秒領域の2つのbitのスナップショットを含みます。スナップショットはケーブル・マスター時刻を受信する度に取り込まれます。

Offset: 00210

**図B-8 RCIM III Cable Master Time Register**

Cable Master Time Registerは秒の境界でクロックの変化の度にケーブルに送信されたマスターRCIMのPOSIXクロックの秒領域を含みます。

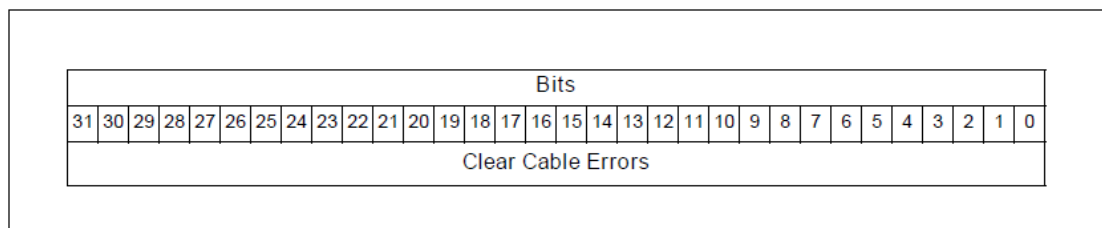
Offset: 00220



図B-9 RCIM III Clear Cable Errors Register

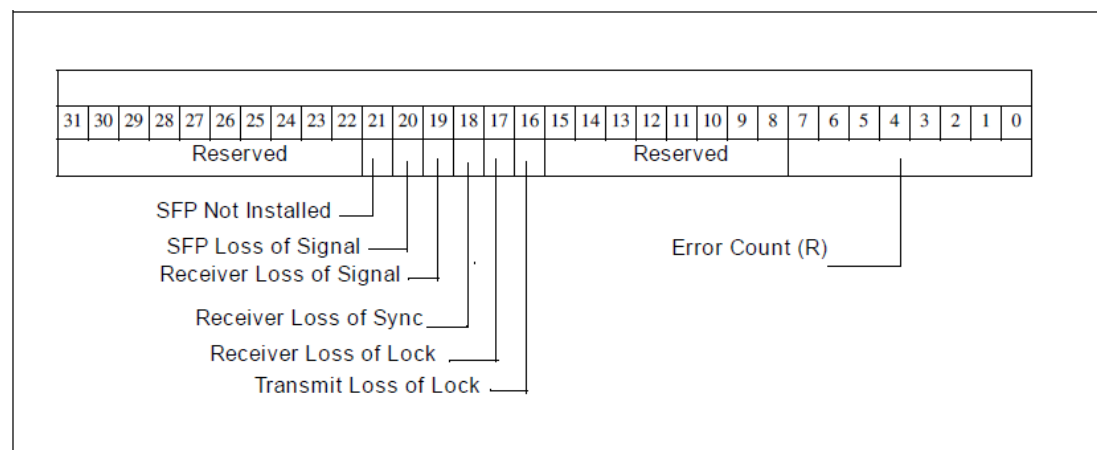
これは報告されたケーブル・エラーを消去する「書き込み専用」のレジスタです。データ領域は気にしないで下さい。

Offset: 00400

**図B-10 RCIM III Output Cable Status Register**

本レジスタは出力ケーブルに関する詳細なハードウェアのステータス情報を提供します。

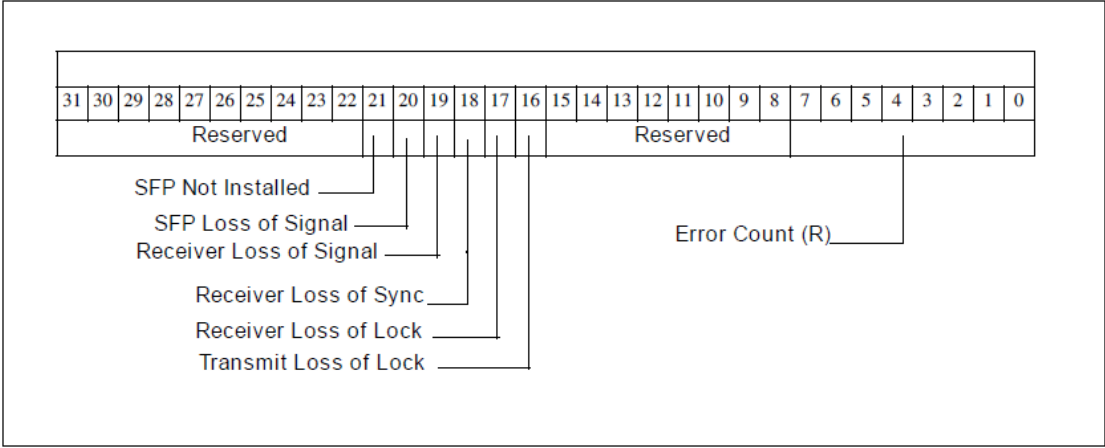
Offset: 00410



図B-11 RCIM III Input Cable Status Register

本レジスタは入力ケーブルに関する詳細なハードウェアのステータス情報を提供します。

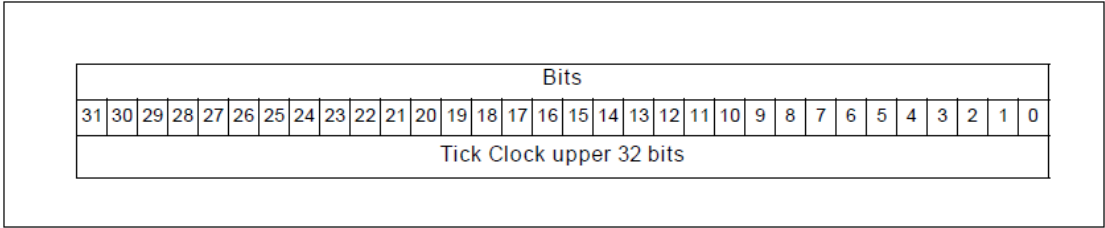
Offset: 00420



図B-12 RCIM III Tick Clock Upper Register

本レジスタはティック・クロックの上位32bitを含みます。

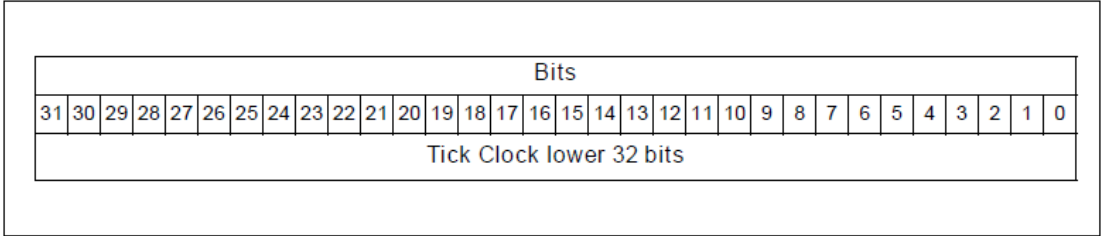
Offset: 01000, 10000



図B-13 RCIM III Tick Clock Lower Register

本レジスタはティック・クロックの下位32bitを含みます。

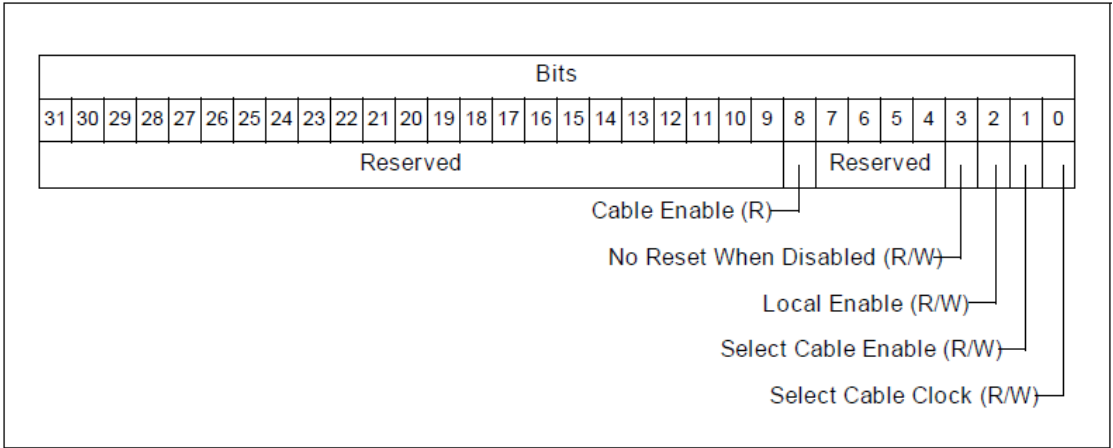
Offset: 01008, 10008



図B-14 RCIM III Tick Clock Status/Control Register

本レジスタはティック・クロックのステータスと制御を提供します。

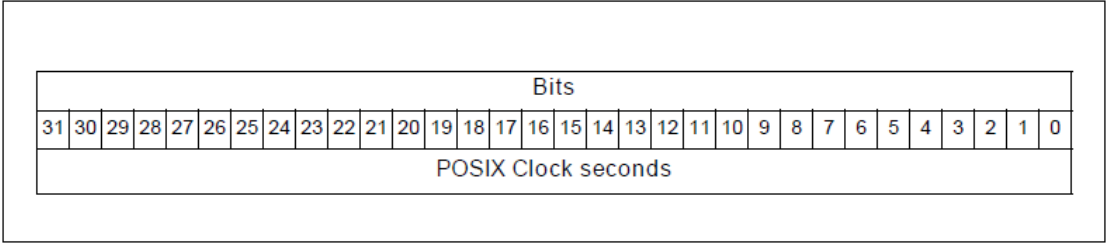
Offset: 01010, 10010



☒B-15 RCIM III POSIX Clock Seconds Register

本レジスタはPOSIXクロックの秒を含みます。

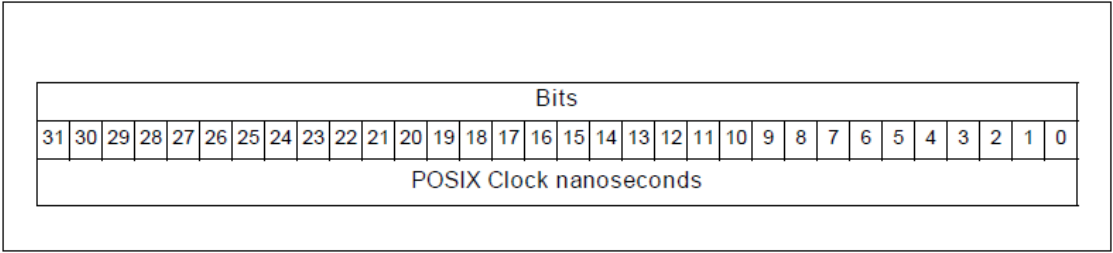
Offset: 01100, 10100



☒B-16 RCIM III POSIX Clock Nanoseconds Register

本レジスタはPOSIXクロックのナノ秒を含みます。

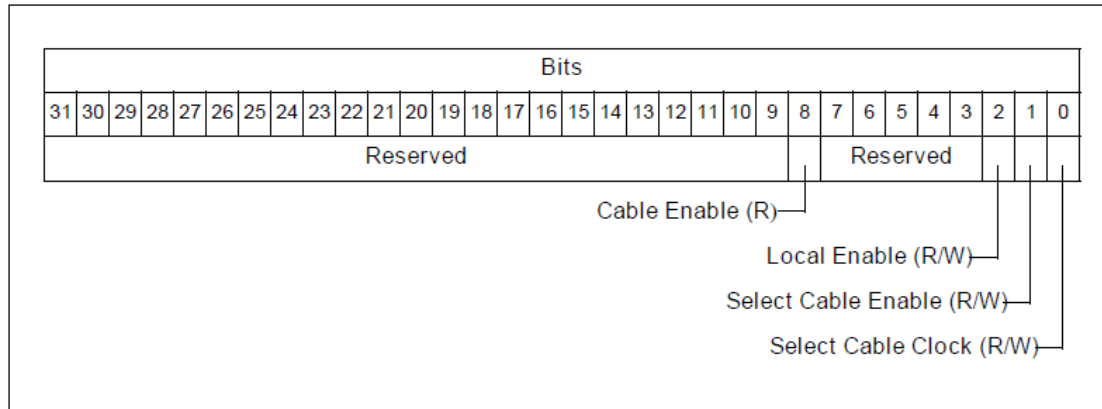
Offset: 01108, 10108



図B-17 RCIM III POSIX Clock Status/Control Register

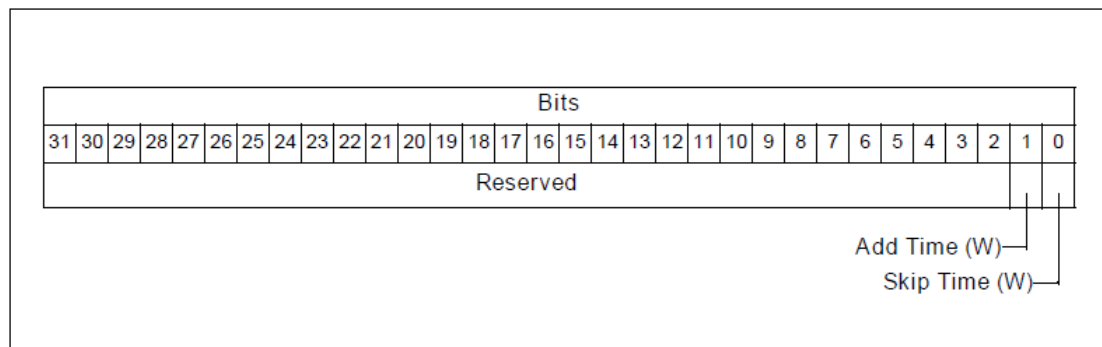
本レジスタはPOSIXクロックのステータスと制御を提供します。

Offset: 01110, 10110

**図B-18 RCIM III POSIX Clock Skip/Add Time Register**

本レジスタは400ナノ秒単位でPOSIXクロックの時間をスキップ/追加します。

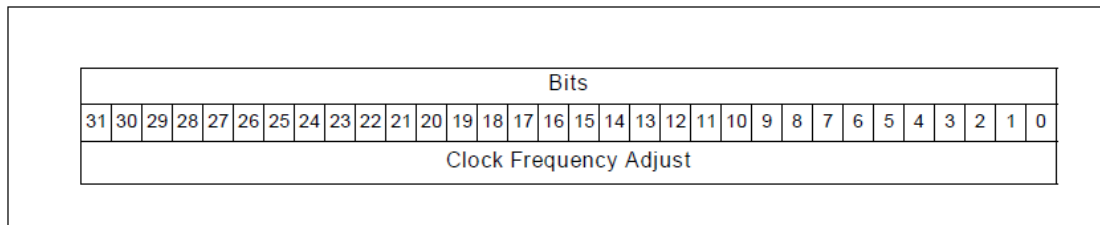
Offset: 01114, 10114



図B-19 RCIM III Clock Frequency Adjust Register

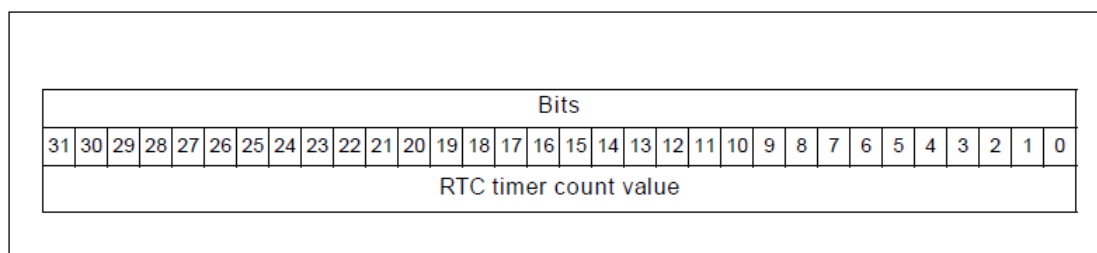
Clock Frequency Adjust Registerは10 MHzのマスター・クロックの周波数を制御するために使用します。

Offset: 01120, 10120

**図B-20 RCIM III RTC Timer Registers**

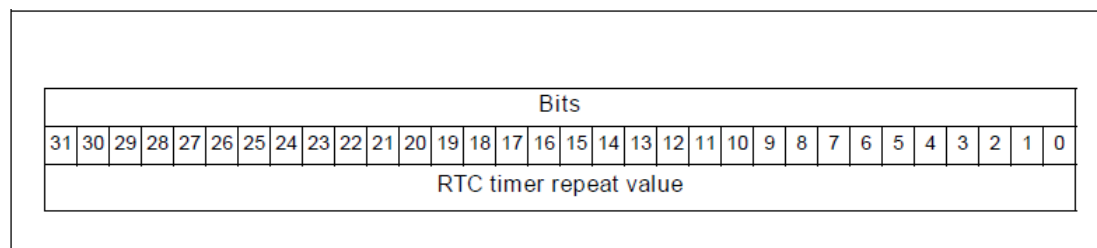
初期のRTCタイマー値はRTC Timer Registerに取り込まれます。現在のタイマー値は本レジスタから読み込まれます。NOTE: 本レジスタの読み取りはRCIMとの互換性のためにRTC Repeat Registerも読み込まれます。

Offset: 02010, 02030, 02050, 02070, 02090, 020B0, 020D0, 020F0

**図B-21 RCIM III RTC Repeat Registers**

RTC Repeat Registerはリピートのカウント値を含みます。

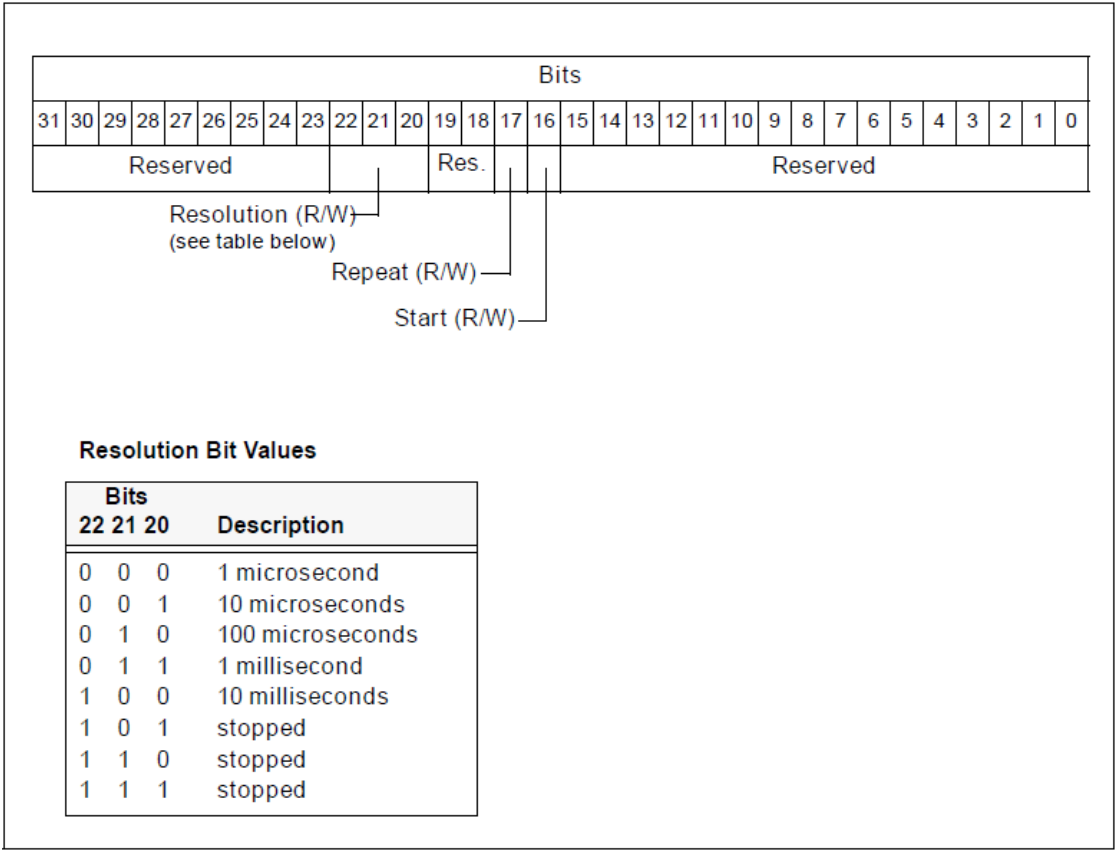
Offset: 02014, 02034, 02054, 02074, 02094, 020B4, 020D4, 020F4



図B-22 RCIM III RTC Control Registers

本レジスタはRTCの制御を提供します。

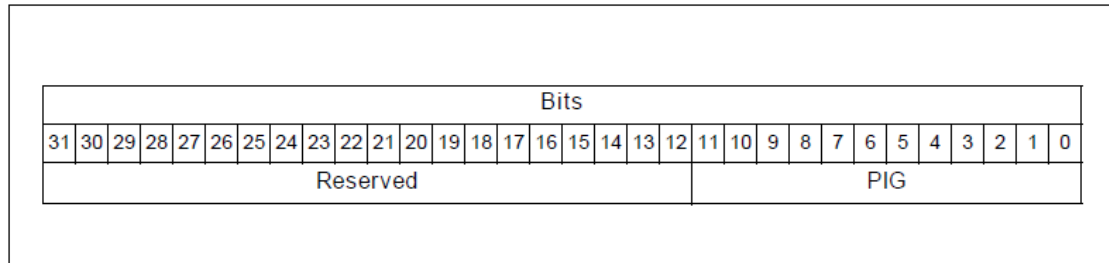
Offset: 02000, 02020, 02040,: 02060, 02080, 020A0, 020C0, 020E0



図B-23 RCIM III Programmable Interrupt Generator Register

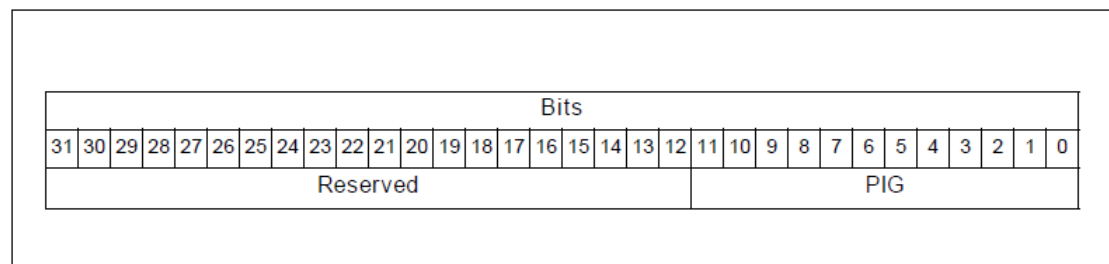
本レジスタはプログラマブル割込みを確認します。

Offset: 03000, 30000

**図B-24 RCIM III Programmable Interrupt Set and Clear Registers**

これらのレジスタへの書き込みは他のビットに影響することなく Programmable Interrupt Registerの単一ビットを設定/消去します。

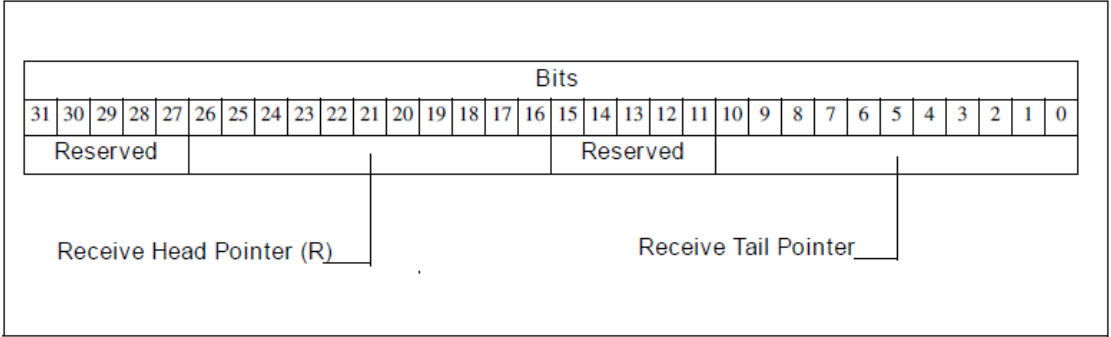
Offset: 03010, 30010, 03020, 30020



図B-25 RCIM III GPS Receive Pointers

GPS Receive PointerはオプションのGPSモジュールとの通信で使用されます。

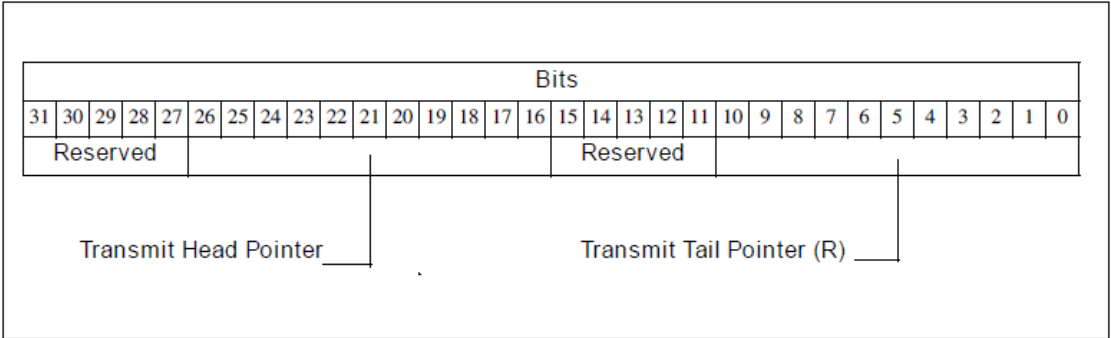
Offset: 03200



図B-26 RCIM III GPS Transmit Pointers

GPS Transmit PointerはオプションのGPSモジュールとの通信で使用されます。

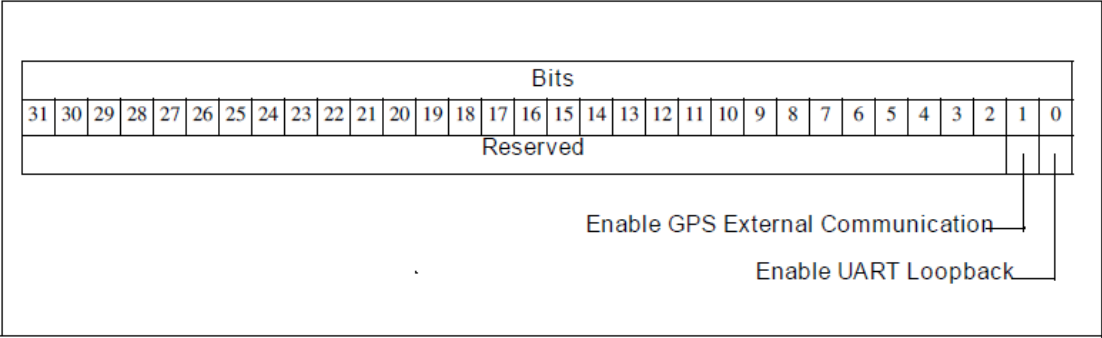
Offset: 03204



図B-27 RCIM III GPS Debug Control Register

GPS Debug Control Registerはテストおよびデバッグ中に使用されるビットを含みます。これらのビットのいずれかを設定するとGPSモジュールとのRCIM通信が無効となります。

Offset: 03208

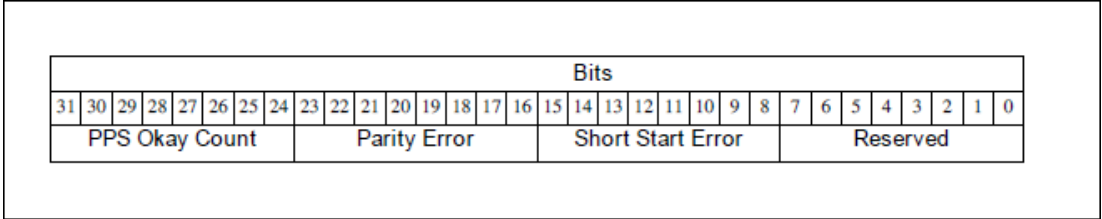


図B-28 RCIM III GPS Communication Error Register

GPS Communication Error RegisterはGPSモジュールとの通信エラーに関する情報を含みます。

このレジスタへの書き込みはGPSモジュールへの通信インターフェースをリセットします。

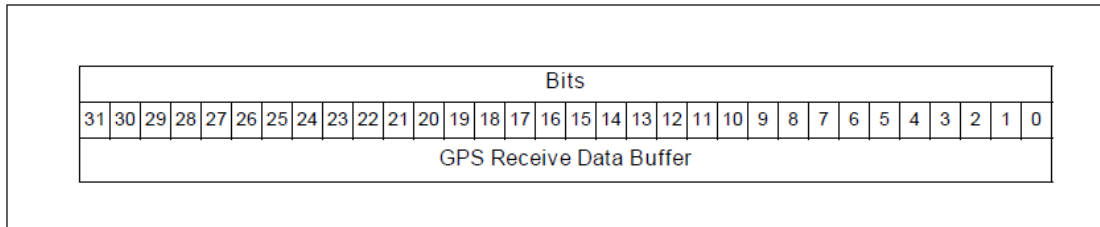
Offset: 0320C



図B-29 RCIM III GPS Receive Data Buffer

これはGPSの受信データ領域です。

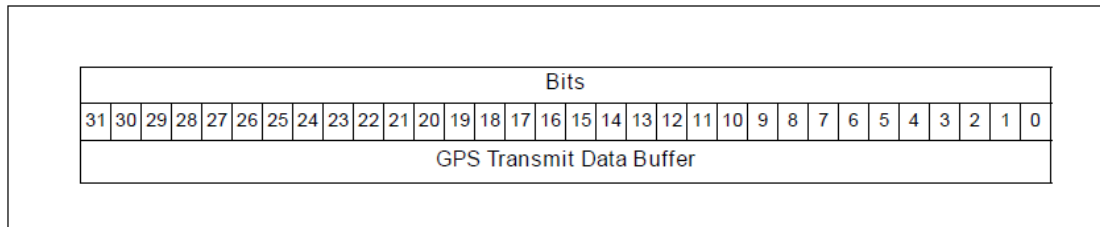
Offset: 04000 to 047FF



図B-30 RCIM III GPS Transmit Data Buffer

これはGPSの転送データ領域です。

Offset: 04800 to 04FFF



RCIMケーブル伝搬遅延の計算

本付録はケーブル接続がシグナルの遅延をもたらすかどうかを割り出すための計算を提供します。

相互接続の詳細

相互接続された各RCIMボード間の最大ケーブル長は30メートル(～100フィート)です。

クロックはティックあたり400ナノ秒で動作します。クロック・シグナルがチェーン内の特定のスレーブに到達するのに400ナノ秒以上掛かる場合、クロックの歪みはその時点から始まります。クロックは各パススルー・スレーブにより再駆動されます。

ケーブル上の連続的なデータを同期および再駆動のため、RCIMチェーンに追加された各RCIMは約200ナノ秒の遅延に加えてケーブル1メートル毎におよそ～7ナノ秒もしくはケーブル30メートル毎に～200ナノ秒が加算され、合計遅延時間は約400ナノ秒となります。

30メートルのケーブルを使ったRCIMチェーン内の2台のシステムは400ナノ秒のクロック・ティック以内で動作します。2台以上とすると厳密な同期からは劣る結果となります。

パススルー・スレーブのシステムが電源OFFである場合、ケーブル・クロックはそこから下流のスレーブへは伝達されないことに注意して下さい。このケースでは、下流のスレーブはケーブル・クロックの代わりにローカルの発信器を使用します。

クロック歪みのレベルがアプリケーションで容認できるかどうかを決定するには、ユーザーはチェーン内の各RCIMおよびケーブルに関連する遅延を認識する必要があります。

