

F&eITシリーズ

RS-232C 2ch
コミュニケーションモジュール
COM-2(FIT)GY
解説書

株式会社コンテック

梱包内容をご確認ください

このたびは、本製品をご購入いただきまして、ありがとうございます。

本製品は次の構成となっています。

構成リストで構成品を確認してください。万一、構成品が足りない場合や破損している場合は、お買い求めの販売店、または総合インフォメーションにご連絡ください。

登録カードは、新製品情報などをお客様にお知らせする際に必要なカードです。ご記入の上、必ずご返送くださいますようお願いいたします。

構成リスト

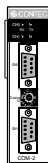
COM-2(FIT)GYモジュール...1

解説書(本書)...1

登録カード&保証書...1

登録カード返信用封筒...1

Question用紙...1



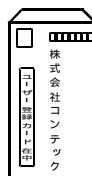
モジュール本体



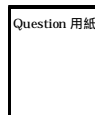
解説書



登録カード & 保証書



登録カード返信用封筒



Question用紙

-
- ・ 本書の内容の全部または一部を無断で転載することは、禁止されています。
 - ・ 本書の内容に関しては、将来予告なしに変更することがあります。
 - ・ 本書の内容については万全を期しておりますが、万一ご不審な点や記載もれなどお気づきのことがありましたら、お買い求めの販売店、または総合インフォメーションへご連絡ください。
 - ・ F&eITは、株式会社コンテックの登録商標です。その他、本書中に使用している会社名および製品名は、一般に各社の商標または登録商標です。

目次

梱包内容をご確認ください.....	i
目次.....	iii

第1章 ご使用になる前に 1

概要.....	1
特長.....	1
サポートのご案内.....	2
ホームページ.....	2
総合インフォメーション(お問い合わせ窓口).....	2
修理窓口.....	3
製品貸出サービス.....	3
各種セミナー.....	3
FA/LA無料相談コーナー.....	3
システム受託開発、OEM受託.....	3
安全にご使用いただくために.....	4
安全情報の表記.....	4
取り扱い上の注意事項.....	5
環境.....	6
点検.....	6
保管.....	6
廃棄.....	6

第2章 モジュールの各部の名称と設定 7

モジュール各部の名称.....	7
Device IDの設定.....	8
設定方法.....	8
LEDインジケータ.....	8

第3章 外部機器との接続 9

インターフェイスコネクタ.....	9
インターフェイスコネクタの接続方法.....	9
ケーブルの接続.....	10

第4章 機能の説明 11

通信機能.....	11
シリアルデータ転送.....	11
送受信用データバッファ.....	11
ボーレートの設定.....	12

第5章 モジュールの使用方法	15
-----------------------	-----------

コンパチブルモードとエンハンスモード	15
I/Oポートとレジスタ	16
I/Oポート、レジスタ詳細	17
割り込みベクタレジスタ	21
割り込みの使用方法	22
使用例	23
コンパチブルモード	24
エンハンスモード	27

第6章 システムリファレンス	31
-----------------------	-----------

ブロック図	31
等価回路	31
仕様	32
外形寸法	33

第 1 章 ご使用になる前に

本章では、本製品をご使用になる前に知っていただくべき情報について説明しています。

概要

COM-2(FIT)GYは、外部装置とシリアル通信を行い、F&eITシリーズのマイクロコントローラモジュール[CPU-SBxx(FIT)GY]の拡張COMポートとして使用できます。

この解説書をよく読んで、アプリケーションプログラムの作成やスイッチの設定および外部機器との接続など、システム構成を行ってください。

特長

- ・ RS-232C準拠のシリアル入出力ポートを2チャンネル備えています。
- ・ 最高921,600bpsの高速通信に対応しています(コンパチブルモードでは115,200bpsまで対応)。
- ・ ボーレートは、ソフトウェアによって個別に設定することができます。
- ・ 送信128byte 受信128byteのFIFOバッファを搭載しています。
- ・ Windows、Linuxの標準COMポートとして使用できます。
- ・ コンパチブルモードとエンハンスモードに対応しており、エンハンスモードを選択した場合、1ユニット当たり1割り込みレベルの占有で使用できます。
- ・ 他のF&eITシリーズ製品と同様、35mmDINレール取り付け機構はモジュール本体に標準で装備されています。また、コントローラユニットとの接続は側面でスタック接続するユニークな構成になっているため、バックプレーン基板などの接続機器を使用せず、簡単、かつスマートにシステムを構成することができます。

サポートのご案内

当社製品をより良く、より快適にご使用いただくために、次のサポートを行っております。

ホームページ

日本語 <http://www.contec.co.jp/>
英語 <http://www.contec.com/>
中国語 <http://www.contec.com.cn/>

最新製品情報

製品の最新情報を提供しています。
また、PDFファイル形式の製品マニュアル、各種技術資料なども提供しています。

資料請求

カタログの請求が行えます。

製品貸出サービス

製品貸出の依頼が行えます。

イベント情報

当社主催/参加のセミナーおよび展示会の紹介を行っております。

総合インフォメーション(お問い合わせ窓口)

技術的なお問い合わせ

当社製品に関する技術的なお問い合わせは、総合インフォメーションで受け付けています。
E-mail(tsc@contec.co.jp)またはFAXでお問い合わせください。専門のスタッフが対応します。
E-mailの場合は、製品添付のQuestion用紙と同様の内容をお送りください。
FAXの場合は、製品添付のQuestion用紙に必要事項を記入の上、お送りください。

FAX番号はQuestion用紙に記載されています。

その他の製品情報のお問い合わせ

製品の価格・納期・見積もり依頼などのお問い合わせは、販売店または当社各支社・営業所までお問い合わせください。

修理窓口

修理の依頼は、お買い求めの販売店経由で受け付けています。

保証書に記載の条件のもとで、保証期間中に製品自体に不具合が認められた場合は、その製品を無償で修理または交換いたします。

保証期間終了後、または保証条件外での修理は、有償修理となりますのであらかじめご了承ください。

なお、対象は製品のハードウェア部分の修理に限らせていただきます。

製品貸出サービス

製品を評価・理解していただくため、製品の貸出サービスを行っております。

詳細は、当社ホームページをご覧ください。

各種セミナー

新製品の紹介・活用方法、システム構築のための技術習得など、各種セミナーを行っております。

出張プライベートセミナーも承ります。詳細は、当社ホームページをご覧ください。

FA/LA無料相談コーナー

「FA/LA無料相談コーナー」は、お客様がシステムを構築する際に当社製品の選定の相談をお受けする窓口です。面談によるシステム相談を専門スタッフが担当いたします。

お問い合わせは、当社各支社・営業所までご連絡ください。

システム受託開発、OEM受託

ソフトウェア/ハードウェアの導入方法やシステム構築のご相談、お客様オリジナル・デザインのシステムを製品化し供給するODMやOEMのご提案を行います。




詳しくは、E-mail(sales@contec.co.jp)または当社各支社・営業所までお問い合わせください。

安全にご使用いただくために

次の内容をご理解の上、本製品を安全にご使用ください。

安全情報の表記

本書では、人身事故や機器の破壊をさけるため、次のシンボルで安全に関する情報を提供しています。内容をよく理解し、安全に機器を操作してください。

 危険	この表示を無視して、誤った取り扱いをすると、人が死亡または重傷を負う危険が差し迫って生じることが想定される内容を示しています。
 警告	この表示を無視して、誤った取り扱いをすると、人が死亡または重傷を負う可能性が想定される内容を示しています。
 注意	この表示を無視して、誤った取り扱いをすると、人が損害を負う可能性が想定される内容および物的損害のみの発生が想定される内容を示しています。

取り扱い上の注意事項

⚠ 注意

- ・ モジュールを改造しないでください。改造したものに対しては、当社は一切の責任を負いません。
- ・ 高温や低温の場所、また温度変化の激しい場所での使用、保存は避けてください。
(動作温度範囲：0～50)
- ・ 直射日光のあたる場所や、ストーブなどの発熱する器具の近くでの使用、保存は避けてください。
- ・ ほこり、湿気の多い場所での使用、保存は避けてください。
(動作湿度範囲：10～90%RH ただし、結露しないこと)
- ・ 本製品は、精密な電子部品でできていますので、衝撃、振動の加わる場所での使用、保存は避けてください。
- ・ 強い磁界、電波を発生する機器の近くでの使用、保存は避けてください。
- ・ 異常な臭いや過熱に気がつきましたら、ただちに電源ケーブルを抜いてください。
- ・ 異常や故障が発生しましたら、お買い上げいただいた販売店にご相談ください。
- ・ 濡れた手で本製品に触らないでください。感電する危険があります。
- ・ モジュールの筐体はあけないでください。あけたものに対しては、当社は一切の責任を負いません。
- ・ モジュールに衝撃を与えたり、曲げたりしないでください。破損の原因となります。
- ・ モジュールの外部接続コネクタ金属端子部には手を触れないでください。接触不良の原因となります。
- ・ モジュール上には、あらかじめ設定を必要とするスイッチがあります。使用する前に必ず確認してください。
- ・ モジュール上のスイッチなどは指定以外の設定にしないでください。故障の原因となります。
- ・ コントローラモジュールの電源が入った状態で、デバイスモジュールの接続操作は実施しないでください。故障の原因になります。必ずコントローラモジュールの電源を切ってから行ってください。

環境

本製品は下記の環境でご使用ください。範囲外の環境で使用した場合、発熱、誤動作、故障の原因になります。

周囲温度

0～50

周囲湿度

10%～90%RH(ただし、結露しないこと)

腐食性ガス

ないこと

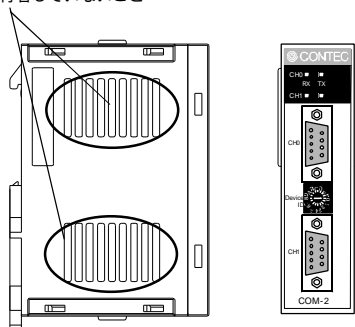
浮遊粉塵

特にひどくないこと

点検

本製品を安全に使用していただくために、定期的に点検を行ってください。

通風スリットをふさいだり、ほこりや異物が
付着していないこと



保管

本製品を保管する際には、購入時の状態で保管してください。

- (1) モジュールを保管袋に入れます。
- (2) 梱包材で包み、箱に入れます。
- (3) 直射日光や湿気、衝撃や振動、磁気や静電気を避けて、常温で保管してください。

廃棄

本製品を廃棄される場合、法律や市町村の条令に定める廃棄方法に従って、廃棄してください。

第 2 章 モジュールの各部の名称と設定

モジュール各部の名称

モジュール各部の名称を図2.1に示しています。

なお、図中のスイッチの状態は、出荷時の設定を示しています。

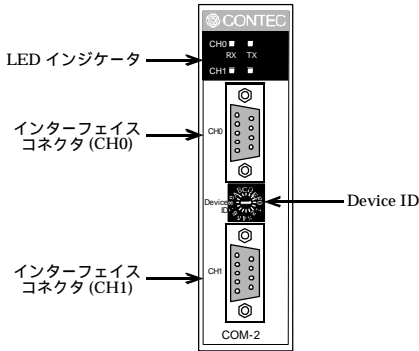


図2.1 モジュール各部の名称

Device IDの設定

Device IDの設定により、コンパチブルモードとエンハンストモードの切り替えを行います。

Device IDと各モードの関係は表2.1に示しています。

コンパチブルモードとエンハンストモードについては、「第4章 モジュールの使用方法」を参照してください。

出荷時設定は「0」でCOM3・COM4の標準ポートとして使用できます。

設定方法

Device IDの設定は、モジュールフェイス上のロータリスイッチで設定します。

スイッチをまわして設定してください。



図2.2 Device IDの設定

表2.1 I/Oアドレス

ID No.	Ch0		Ch1		モード
	I/Oアドレス(h)	割り込みレベル	I/Oアドレス(h)	割り込みレベル	
0	03E8h ~ 03EFh	IRQ5	02E8h ~ 02EFh	IRQ7	コンパチ(COM3/4)
1	03E8h ~ 03EFh	IRQ5	02E8h ~ 02EFh	未使用	コンパチ(COM3/4)
2	03E8h ~ 03EFh	未使用	02E8h ~ 02EFh	IRQ7	コンパチ(COM3/4)
3	03E8h ~ 03EFh	未使用	02E8h ~ 02EFh	未使用	コンパチ(COM3/4)
4	01A0h ~ 01A7h	IRQ5	01A8h ~ 01AFh	IRQ5	エンハンスト
5	01A0h ~ 01A7h	IRQ7	01A8h ~ 01AFh	IRQ7	エンハンスト
6	02A0h ~ 02A7h	IRQ5	02A8h ~ 02AFh	IRQ5	エンハンスト
7	02A0h ~ 02A7h	IRQ7	02A8h ~ 02AFh	IRQ7	エンハンスト
8	9800h ~ 9807h	IRQ9	9808h ~ 980Fh	IRQ9	エンハンスト
9	9820h ~ 9827h	IRQ5	9828h ~ 982Fh	IRQ7	エンハンスト
A	9840h ~ 9847h	IRQ7	9848h ~ 984Fh	IRQ9	エンハンスト
B	9860h ~ 9867h	未使用	9868h ~ 986Fh	未使用	エンハンスト
C	700h ~ 707h	IRQ9	708h ~ 70Fh	IRQ9	エンハンスト
D	720h ~ 727h	IRQ7	728h ~ 72Fh	IRQ7	エンハンスト
E	予約	予約	予約	予約	予約
F	予約	予約	予約	予約	予約

LEDインジケータ

RX: シリアルデータ転送の受信状態を示します。(GREEN)

TX: シリアルデータ転送の転送状態を示します。(GREEN)

第3章 外部機器との接続

インターフェイスコネクタ

インターフェイスコネクタの接続方法

モジュール上のインターフェイスコネクタを用いて、外部機器と接続します。

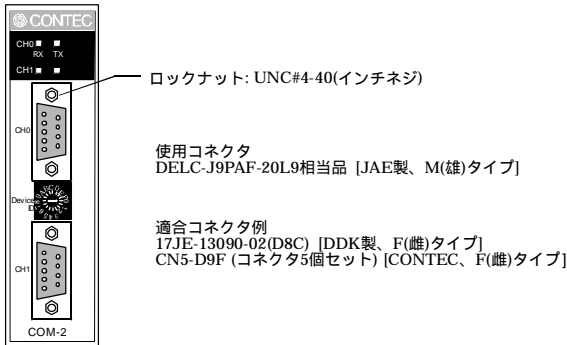


図3.1 インターフェイスコネクタと使用コネクタ

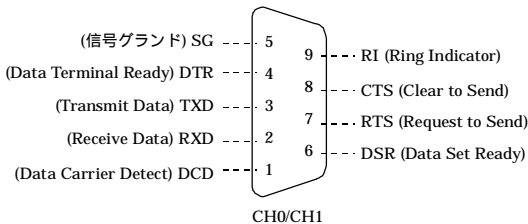


図3.2 インターフェイスコネクタの信号配置

ケーブルの接続

RS-232Cインターフェイスの接続は、モデムやコンピュータ(パソコン)などのように、接続する装置によって使用するケーブルが異なる場合があります。したがって、ケーブルは接続する外部装置の仕様を確認の上、その種別(仕様)によってストレートタイプ、あるいはクロス(リバーシブル)タイプを用意してください。さらに、コネクタ内で信号線処理の必要がある場合には、仕様に合わせ適切に処理を行ってください。

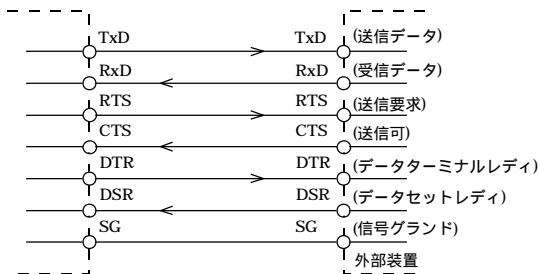


図3.3 モデムとの接続例

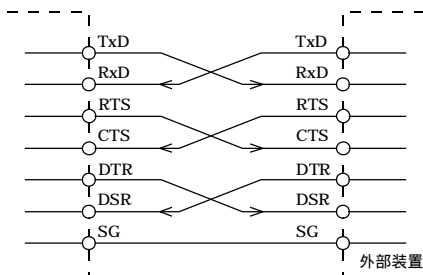


図3.4 パソコンとの接続例

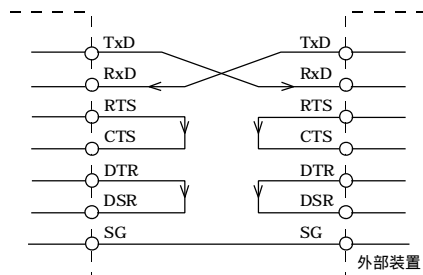


図3.5 機器との接続例

⚠ 注意

誤った結線で接続すると、接続機器や本モジュールの故障原因になります。

第4章 機能の説明

本章では、ボードに搭載されている機能について説明しています。

通信機能

シリアルデータ転送

RS-232C規格に沿ったデータを送受信することが可能です。

2～921,600bpsまでのボーレートをソフトウェアで各チャンネル個別に設定できます。

送受信用データバッファ

送信専用データバッファ128byte、受信専用データバッファ128byteを備えています。

このデータバッファはFIFO形式で、高速な通信やシステムでのCPUの負荷調整に役立ちます。

FIFOトリガサイズは、ハードウェア割り込みの発生タイミングを決定する、データバッファのサイズです。

FIFOトリガサイズが大きいと、通信データの送受信によるハードウェア割り込みの発生回数が減るため、CPUへの負荷軽減になります。

しかし、通信データがある程度たまってから実際に送受信するので、送信データが相手機器に届くタイミングが遅れたり、受信データが届いてからアプリケーションに通知するタイミングが遅れたりします。

逆に、FIFOトリガサイズを小さくしたりFIFOの設定を無効にすれば、データの送受信のタイミングは早くなりますが、CPUの負荷が高くなったり、受信データの取りこぼしにつながる可能性があります。

FIFOトリガサイズが可変であることから、システムに応じた調整をすることで、最適なパフォーマンスを実現できます。

ボーレートの設定

分周レジスタへの設定値により各種ボーレートが設定可能です。クロックモードにより設定可能な範囲が異なります。

クロック3モード：15～921,600bps

クロック2モード：8～460,800bps

クロック1モード：4～230,400bps

クロック0モード：2～115,200bps

ただし、分周レジスタの設定値により設定できないボーレートもあります。以下の計算式で整数値になればそのボーレートは設定可能ですが、小数点がでる場合は設定できません。

クロック3モードの場合

921600 ÷ 設定するボーレート = 分周レジスタに設定する値

例) 921600 ÷ 9600bps = 96 (計算の結果、整数値なので設定可能)

921600 ÷ 128000bps = 7.2 (計算の結果、小数点がでたので設定できません)

クロック2モードの場合

460800 ÷ 設定するボーレート = 分周レジスタに設定する値

例) 460800 ÷ 9600bps = 48 (計算の結果、整数値なので設定可能)

460800 ÷ 128000bps = 3.6 (計算の結果、小数点がでたので設定できません)

クロック1モードの場合

230400 ÷ 設定するボーレート = 分周レジスタに設定する値

例) 230400 ÷ 9600bps = 24 (計算の結果、整数値なので設定可能)

230400 ÷ 128000bps = 1.8 (計算の結果、小数点がでたので設定できません)

クロック0モードの場合

115200 ÷ 設定するボーレート = 分周レジスタに設定する値

例) 115200 ÷ 9600bps = 12 (計算の結果、整数値なので設定可能)

115200 ÷ 76800bps = 1.5 (計算の結果、小数点がでたので設定できません)

注意

クロックの選択(1.8432MHz, 3.6864MHz, 7.3728MHz, または14.7456MHz)は、「第5章 割り込みベクタレジスタ」を参照してください。

コンパチブルモードで使用される場合は、クロック0(1.8432MHz)固定となります。

次にボーレートの設定例を紹介します。これ以外のボーレートでも前記の計算式にあてはまれば設定可能です。

表4.1 ボーレートの設定例

設定する ボーレート	クロック0モード (1.8432MHz)		クロック1モード (3.6864MHz)		クロック2モード (7.3728MHz)		クロック3モード (14.7456MHz)	
	分周レジスタ に設定する値	設定誤差 (%)	分周レジスタ に設定する値	設定誤差 (%)	分周レジスタ に設定する値	設定誤差 (%)	分周レジスタ に設定する値	設定誤差 (%)
2	57600	-						
4	28800	-	57600	-				
5	23040	-	46080	-				
8	14400	-	28800	-	57600	-		
15	7680	-	15360	-	30720	-	61440	-
50	2304	-	4608	-	9216	-	18432	-
75	1536	-	3072	-	6144	-	12288	-
110	1047	0.026	2094	0.026	4189	0.0022	8378	0.0022
134.5	857	0.058	1713	0.0006	3426	0.0006	6852	0.0006
150	768	-	1536	-	3072	-	6144	-
300	384	-	768	-	1536	-	3072	-
600	192	-	384	-	768	-	1536	-
1200	96	-	192	-	384	-	768	-
1800	64	-	128	-	256	-	512	-
2000	58	0.68	115	0.17	230	0.17	461	0.04
2400	48	-	96	-	192	-	384	-
3600	32	-	64	-	128	-	256	-
4800	24	-	48	-	96	-	192	-
7200	16	-	32	-	64	-	128	-
9600	12	-	24	-	48	-	96	-
14400	8	-	16	-	32	-	64	-
19200	6	-	12	-	24	-	48	-
28800	4	-	8	-	16	-	32	-
38400	3	-	6	-	12	-	24	-
57600	2	-	4	-	8	-	16	-
76800			3	-	6	-	12	-
115200	1	-	2	-	4	-	8	-
153600					3	-	6	-
230400			1	-	2	-	4	-
460800					1	-	2	-
921600							1	-

第5章 モジュールの使用法

コンパチブルモードとエンハンストモード

COM-2(FIT)GYには、CPU-SBxx(FIT)GYと接続することにより標準シリアルポートとして動作するコンパチブルモードと、当社独自の制御方法で操作するエンハンストモードがあります。それぞれの長を理解した上で、構築するシステムに合ったモードを使用してください。

Windows環境で4CH以上のCOMをご使用になる場合、エンハンストモードに設定の上、COM-DRV(W32)ドライバでご使用ください。(例:Device ID-No. 04h, 07h, 0Ch)

コンパチブルモード

COM-2(FIT)GYの2つのチャンネルを標準シリアルポートとして使用することができます。COM-2(FIT)GYのCH0はCOM3に、CH1はCOM4に割り当てることができます。

コンパチブルモードでは、チャンネルごとに1つの割り込みレベルを使用します。

2チャンネル増設する場合、割り込みレベルが2つ以上あることが必要です。また、I/Oアドレスは標準シリアルポートに割り当てられているアドレスを使用します。標準シリアルポートとして扱われるため、標準シリアルポートを操作できるその他のドライバソフトを使用して動作させることができます。

エンハンストモード

COM-2(FIT)GYを当社独自の制御方法で操作します。

エンハンストモードでは、1つの割り込みレベルで2つのチャンネルを制御することができます。利用できる割り込みレベルが1つしかない場合でも動作が可能です。また、I/Oアドレスは当社独自のアドレスを使用します。

COM-DRV(W32)ドライバを使用した場合、標準シリアルポートと同等の使い方が出来ます。

表5.1 I/Oアドレス

ID No.	CH0			CH1		
	機能	I/Oアドレス(h)	割り込みレベル	機能	I/Oアドレス(h)	割り込みレベル
0	COM3	03E8 ~ 03EF	IRQ5	COM4	02E8 ~ 02EF	IRQ7
1	COM3	03E8 ~ 03EF	IRQ5	COM4	02E8 ~ 02EF	未使用
2	COM3	03E8 ~ 03EF	未使用	COM4	02E8 ~ 02EF	IRQ7
3	COM3	03E8 ~ 03EF	未使用	COM4	02E8 ~ 02EF	未使用
4	エンハンスト	01A0 ~ 01A7	IRQ5	エンハンスト	01A8 ~ 01AF	IRQ5
5	エンハンスト	01A0 ~ 01A7	IRQ7	エンハンスト	01A8 ~ 01AF	IRQ7
6	エンハンスト	02A0 ~ 02A7	IRQ5	エンハンスト	02A8 ~ 02AF	IRQ5
7	エンハンスト	02A0 ~ 02A7	IRQ7	エンハンスト	02A8 ~ 02AF	IRQ7
8	エンハンスト	9800 ~ 9807	IRQ9	エンハンスト	9808 ~ 980F	IRQ9
9	エンハンスト	9820 ~ 9827	IRQ5	エンハンスト	9828 ~ 982F	IRQ7
A	エンハンスト	9840 ~ 9847	IRQ7	エンハンスト	9848 ~ 984F	IRQ9
B	エンハンスト	9860 ~ 9867	未使用	エンハンスト	9868 ~ 986F	未使用
C	エンハンスト	700 ~ 707	IRQ9	エンハンスト	708 ~ 70F	IRQ9
D	エンハンスト	720 ~ 727	IRQ7	エンハンスト	728 ~ 72F	IRQ7
E	予約	予約	予約	予約	予約	予約
F	予約	予約	予約	予約	予約	予約

I/Oポートとレジスタ

このモジュールでは、UART 16550の上位互換品であるXR16C2850(EXAR社)を使用しています。XR16C2850の内部レジスタと制御の詳細は、XR16C2850データシートを参照してください。また、エンハンスモードでは「先頭I/Oアドレス+1Fh」が割り込みベクタレジスタとして使われます。

入出力ポート

先頭I/O アドレス	D7	D6	D5	D4	D3	D2	D1	D0
+0h	チャンネル0レジスタ							
+1h								
+2h								
+3h								
+4h								
+5h								
+6h								
+7h								
+8h	チャンネル1レジスタ							
+9h								
+Ah								
+Bh								
+Ch								
+Dh								
+Eh								
+Fh								
+1Fh	割り込みベクタレジスタ							

図5.1 入出力ポート

⚠ 注意

各ポートへのアクセスはデバイスの仕様上、バイト単位で行ってください。バイトアクセス以外(ワードアクセスやダブルワードアクセスなど)で、各ポートにアクセスすることはできません。

I/Oポート、レジスタ詳細

XR16C2850内部レジスタは、Line Control Register(LCR)に設定する値によりポートマップが変化します。

起動時およびLCR bit 7 = 0、LCR = 0xBF以外のときGeneral Registersが有効になります。

LCR bit 7 = 1のときBaud Rate Registersが有効になります。

LCR = 0xBFのときEnhanced Registersが有効になります。

入力ポート1(General Registers)

先頭I/Oアドレス	D7	D6	D5	D4	D3	D2	D1	D0
+0h/+8h	Receive Holding Register (RHR)							
	Data Bit 7	Data Bit 6	Data Bit 5	Data Bit 4	Data Bit 3	Data Bit 2	Data Bit 1	Data Bit 0
+1h/+9h	Interrupt Enable Register (IER)							
	CTS Interrupt (*2)	RTS Interrupt (*2)	Xoff Interrupt (*2)	Sleep Mode (*2)	MODEM Status Interrupt	Receive Line Status Interrupt	Transmit Holding Register	Receive Holding Register
+2h/+Ah	Interrupt Status Register (ISR)							
	FIFOs Enable	FIFOs Enable	-RTS, -CTS	Xoff	Int Priority Bit 2	Int Priority Bit 1	Int Priority Bit 0	Int Status
+3h/+Bh	Line Control Register (LCR)							
	Divisor Latch Enable	Set Break	Set Parity	Even Parity	Parity Enable	Stop Bits	Word Length Bit 1	Word Length Bit 0
+4h/+Ch	Modem Control Register (MCR)							
	Clock Prescaler Select (*2)	IR Mode Enable (*2)	Xon Any (*2)	Loop Back Enable	-OP2 and INT Enable	Out 0:INT Enable 1:INT Disable (*1)	-RTS	-DTR
+5h/+Dh	Line Status Register (LSR)							
	FIFO Error	THR& TSR Empty	THR Empty	Break Interrupt	Framing Error	Parity Error	Overrun Error	Receive Data Ready
+6h/+Eh	Modem Status Register (MSR)							
	-CD	-RI	-DSR	-CTS	Delta -CD	Delta -RI	Delta -DSR	Data -CTS
+7h/+Fh	Scratch Pad Register (SPR)							
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
+7h/+Fh	FIFO Level Counter (FLVL) (*3)							
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

*1 詳細については割り込み使用方法を参照してください。

*2 このビットはEFRビット4 = 1のとき有効になります。

*3 このレジスタはFCTRビット6 = 1のとき有効になります。

図5.2 入力ポート(General Registers)

入力ポート2(Baud Rate Registers)

先頭I/Oアドレス	D7	D6	D5	D4	D3	D2	D1	D0
+0h/+8h	LSB of Divisor Latch (DLL)							
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
+1h/+9h	MSB of Divisor Latch (DLM)							
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8

図5.3 入力ポート(Baud Rate Registers)

入力ポート3(Enhanced Registers)

先頭I/Oアドレス	D7	D6	D5	D4	D3	D2	D1	D0
+0h/+8h	FIFO Level Counter							
	Trig/FC	Trig/FC	Trig/FC	Trig/FC	Trig/FC	Trig/FC	Trig/FC	Trig/FC
+1h/+9h	Feature Control Register (FCTR)							
	Rx/Tx Mode	SCPAD Swap	Trig Bit 1	Trig Bit 0	RS485 Auto control	IrRx Inv.	-RTS Delay Bit 1	-RTS Delay Bit 0
+2h/+Ah	Enhanced Feature Register (EFR)							
	Auto -CTS	Auto -RTS	Special Char. select	Enable IER Bits 4-7, ISR, FCR Bits 4-5, MCR Bits 5-7	Cont-3 Tx, Rx Control	Cont-2 Tx, Rx Control	Cont-1 Tx, Rx Control	Cont-0 Tx, Rx Control
+4h/+Ch	Xon-1 Word							
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
+5h/+Dh	Xon-2 Word							
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
+6h/+Eh	Xoff-1 Word							
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
+7h/+Fh	Xoff-2 Word							
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8

図5.4 入力ポート(Enhanced Registers)

入力ポート4(割り込みベクタレジスタ)

先頭I/Oアドレス	D7	D6	D5	D4	D3	D2	D1	D0	
+1Fh	割り込みベクタレジスタ								
	"0"固定	(予約)					CH1 割り込み	CH0 割り込み	

図5.5 入力ポート(割り込みベクタ)

出力ポート1(General Registers)

先頭I/Oアドレス	D7	D6	D5	D4	D3	D2	D1	D0
+0h/+8h	Transmit Holding Register (THR)							
	Data Bit 7	Data Bit 6	Data Bit 5	Data Bit 4	Data Bit 3	Data Bit 2	Data Bit 1	Data Bit 0
+1h/+9h	Interrupt Enable Register (IER)							
	CTS Interrupt (*2)	RTS Interrupt (*2)	Xoff Interrupt (*2)	Sleep Mode (*2)	MODEM Status Interrupt	Receive Line Status Interrupt	Transmit Holding Register	Receive Holding Register
+2h/+Ah	FIFO Control Register (FCR)							
	RCVR	RCVR	TX Trigger (MSB)	TX Trigger (LSB)	DMA Mode Select	XMIT FIFO Reset	RCVR FIFO Reset	FIFO Enable
+3h/+Bh	Line Control Register (LCR)							
	Divisor Latch Enable	Set Break	Set Parity	Even Parity	Parity Enable	Stop Bits	Word Length Bit 1	Word Length Bit 0
+4h/+Ch	Modem Control Register (MCR)							
	Clock Prescaler Select (*2)	IR Mode Enable (*2)	Xon Any (*2)	Loop Back Enable	-OP2 and INT Enable	Out 1 0:INT Enable 1:INT Disable (*1)	-RTS	-DTR
+5h/+Dh	使用不可							
+6h/+Eh	使用不可							
+7h/+Fh	Scratch Pad Register (SPR)							
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
+7h/+Fh	Enhanced Mode Select Registers (EMSR) (*3)							
	Reserved	Reserved	RTS Hyst Bit 3	RTS Hyst Bit 2	Reserved	Reserved	Alt.RX/TX FIFO Count	RX/TX FIFO Count

*1 詳細については割り込み使用方法を参照してください。

*2 このビットはEFRビット4=1のとき有効になります。

*3 このレジスタはFCTRビット6=1のとき有効になります。

図5.6 出力ポート(General Registers)

出力ポート2(Baud Rate Registers)

先頭I/Oアドレス	D7	D6	D5	D4	D3	D2	D1	D0
+0h/+8h	LSB of Divisor Latch (DLL)							
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
+1h/+9h	MSB of Divisor Latch (DLM)							
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8

図5.7 出力ポート(Baud Rate Registers)

出力ポート3(Enhanced Registers)

先頭I/Oアドレス	D7	D6	D5	D4	D3	D2	D1	D0
+0h/+8h	FIFO Level Counter							
	Trig/FC	Trig/FC	Trig/FC	Trig/FC	Trig/FC	Trig/FC	Trig/FC	Trig/FC
+1h/+9h	Feature Control Register (FCR)							
	Rx/Tx Mode	SCPAD Swap	Trig Bit 1	Trig Bit 0	RS485 Auto control	IrRx Inv.	-RTS Delay Bit 1	-RTS Delay Bit 0
+2h/+Ah	Enhanced Feature Register (EFR)							
	Auto -CTS	Auto -RTS	Special Char. select	Enable IER Bits 4-7, ISR, FCR Bits 4-5, MCR Bits 5-7	Cont-3 Tx, Rx Control	Cont-2 Tx, Rx Control	Cont-1 Tx, Rx Control	Cont-0 Tx, Rx Control
+4h/+Ch	Xon-1 Word							
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
+5h/+Dh	Xon-2 Word							
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
+6h/+Eh	Xoff-1 Word							
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
+7h/+Fh	Xoff-2 Word							
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8

図5.8 出力ポート(Enhanced Register)

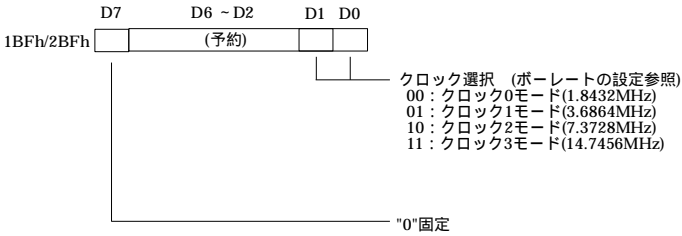
出力ポート4(割り込みベクタレジスタ)

先頭I/Oアドレス	D7	D6	D5	D4	D3	D2	D1	D0
+1Fh	割り込みベクタレジスタ							
	"0"固定	(予約)					クロック 選択 Bit 1	クロック 選択 Bit 0

図5.9 出力ポート(割り込みベクタレジスタ)

割り込みベクタレジスタ

出力ポート



入力ポート

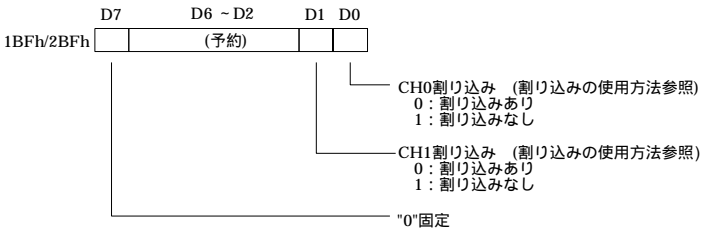


図5.10 割り込みベクタレジスタ

割り込みの使用法

コンパチブルモード

このモジュールをコンパチブルモードで使用する場合、CH0はCOM3と、CH1はCOM4の標準ポートのI/Oアドレスに設定されます。割り込みレベルはそれぞれCH0がIRQ5、CH1がIRQ7となります。

エンハンストモード

このモジュールをエンハンストモードで使用する時、個々のチャンネルの割り込みは、割り込みが発生すると割り込みベクタレジスタ(IVR)で確認できます。どの割り込みラインを使用するかは、Device IDスイッチで設定します。

割り込みを受け付けたとき、割り込みサービスルーチン内で、どのチャンネルから割り込みが発生したか、IVRを読み込むことで調べることができます。割り込みサービスを実行後、割り込みサービスを抜ける前に再度IVRを読み込み、ペンディングになっている割り込みがないか調べる必要があります。

XR16C2850は、それ自身で割り込みをイネーブル(使用可能)にする、または割り込みを発生したか見分けるために内部にレジスタを持っています。このレジスタの使用法については、EXAR社のXR16C2850のデータシートを参照してください。

割り込みを使用する場合は、モデムコントロールレジスタ(MCR)のD2ビットで設定します。

MCRのD2ビット 「0」を書き込み：イネーブル(電源投入時)
 「1」を書き込み：ディセーブル

コンパチブルモード

フローチャート

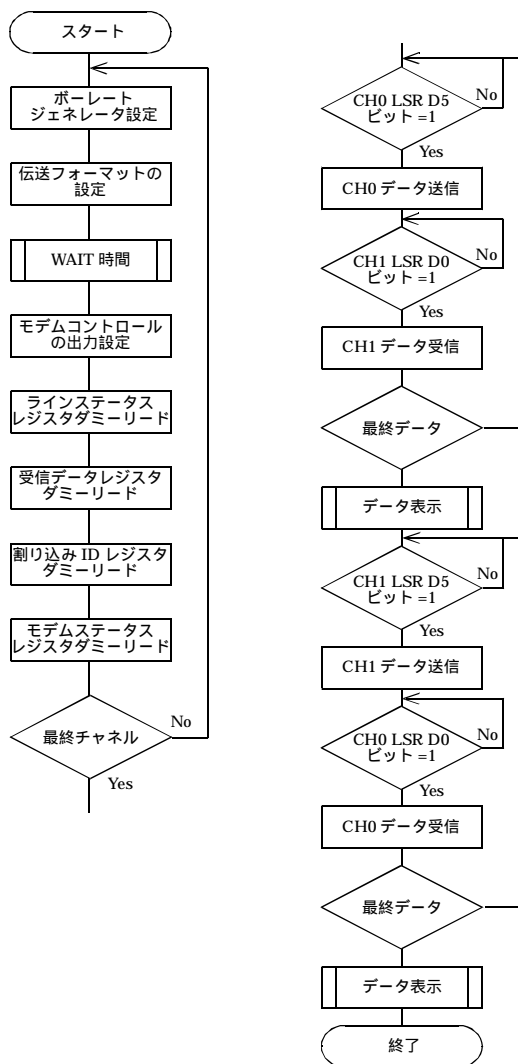


図5.12 フローチャート(コンパチブルモード)

サンプルプログラム例

```

/*=====
Sample program 1  -- CH0 <=> CH1 Loop Test --

DEVICE ID:      0
Mode:           Compatible Mode
Channel 0:      03E8H (COM3)
Channel 1:      02E8H (COM4)
Baud Rate:      115,200 bps
Interrupt:      N/A

===== */
#include <stdio.h>
#include <conio.h>

/* ----- Constant ----- */
#define CH      2          /* no. of channels */
#define NUM     10         /* data of total number */
#define BPS     1          /* baud rate */

unsigned charbuf[CH][NUM]; /* receive buffer */
unsigned int  ADR[CH] = {0x3e8, 0x2e8}; /* I/O address */
unsigned int  rcnt[CH] = {0, 0}; /* receive count */

/* ----- Prototype ----- */
void main( void );

/* ----- Main ----- */
void main( void )
{
    unsigned chari, j;
    int          dummy;

    for(i = 0; i < CH; i ++ ) { /* RS-232C initialize */
        outp( ADR[i]+0x3, 0x80 ); /* Line Control Register (DLAB SET) */
        outp( ADR[i]+0x0, BPS & 0xff ); /* Divisor Latch (Baud Rate SET) */
        outp( ADR[i]+0x1, (BPS >> 8) & 0xff );
        outp( ADR[i]+0x3, 0x0 ); /* Line Control Register (DLAB RESET) */

        outp( ADR[i]+0x3, 0x7 ); /* Line Control Register */
        /* ( Bit 0,1 ... Word Length = 8 bit ) */
        /* ( Bit 2 ..... 2 stop bit ) */

        for(j = 0; j < 10; j++) /* hardware wait */
            dummy = inp( 0x2ef );

        outp( ADR[i]+0x4, 0x3 ); /* MODEM Control Register (DTR . RTS) */
        dummy = inp( ADR[i]+0x5 ); /* Line Status Register(Dummy Read) */
        dummy = inp( ADR[i]+0x0 ); /* Receiver Buffer Register (Dummy Read) */
        dummy = inp( ADR[i]+0x2 ); /* Interrupt ID. Register (Dummy Read) */
        dummy = inp( ADR[i]+0x6 ); /* Modem Status Register (Dummy Read) */
        outp( ADR[i]+0x1, 0x1 ); /* Interrupt Enable Register */
    }

    for(j = 0; j < NUM; j++) { /* send CH0 to CH1 */
        while( !(inp( ADR[0]+0x5 ) & 0x20) ); /* transmitter empty */
        outp( ADR[0]+0x0, j );
        while( !(inp( ADR[1]+0x5 ) & 0x1) ); /* data ready */
        rbuf[1][rcnt[1]] = (unsigned char)inp( ADR[1]+0x0 );
        rcnt[1]++;
    }
}

```

```
for(j = 0; j < NUM; j++) {
    printf(" CH0:%02x -> CH1:%02x  ", j, rbuf[1][j]);
    if(rbuf[1][j] != j) printf("Verify Error %n");      /* verify data */
    else printf("%n");
}
printf("%n");

for(j = 0; j < NUM; j++) {
    while( !(inp( ADR[1]+0x5 ) & 0x20) );      /* transmitter empty */
    outp( ADR[1]+0x0, j );
    while( !(inp( ADR[0]+0x5 ) & 0x1) );      /* data ready */
    rbuf[0][rcnt[0]] = (unsigned char)inp( ADR[0]+0x0 );
    rcnt[0]++;
}
for(j = 0; j < NUM; j++) {
    printf(" CH1:%02x -> CH0:%02x  ", j, rbuf[0][j]);
    if(rbuf[0][j] != j) printf("Verify Error %n");      /* verify data */
    else printf("%n");
}
}

/* ----- End of file --- */
```

エンハンストモード

フローチャート

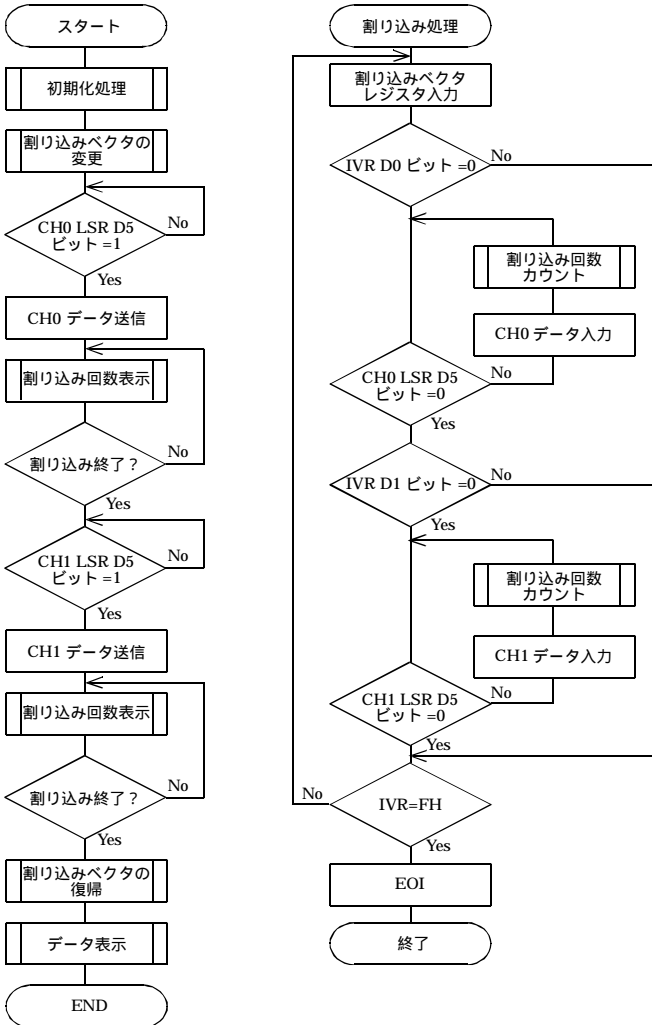


図5.13 フローチャート(エンハンストモード)

サンプルプログラム例

```

/* =====
Sample program 2    -- CH0 <=> CH1 Loop Test --

DEVICE ID:      4
Mode:           Enhanced Mode
Channel 0:      01A0H
Channel 1:      01A8H
Baud Rate:      115,200 bps
Interrupt:      IRQ5

===== */
#include <stdio.h>
#include <conio.h>
#include <dos.h>

/* ----- Constant ----- */
#define CH      2                /* no. of channels */
#define NUM     10               /* data of total number */
#define BPS     1                /* baud rate */
#define IRQ5    0                /* IRQ5 */
#define IRQ7    1                /* IRQ7 */

volatile unsigned char rbuf[CH][NUM];      /* receive buffer */
volatile unsigned int  ADRA[CH] = {0x1a0, 0x1a8}; /* I/O address */
volatile unsigned int  intcnt[CH] = {0, 0};  /* interrupt counter */
volatile int           IrqLevel = IRQ5;      /* interrupt level */
int                  OrgMasterImr;         /* original IMR */
unsigned char         IntVector[2] = { 0x0d, 0x0f }; /* interrupt vector */
/* interrupt vector */
unsigned char         PicMask[2] = { 0xdf, 0x7f }; /* mask bit */
/* mask bit */
unsigned char         IsrClear[3] = { 0x65, 0x67, 0x61 }; /* ISR clear */
/* ISR clear */

/* ----- Prototype ----- */
void main( void );
void Initialize( void ); /* initialize */
void ChgVect( void ); /* change vector */
void ResVect( void ); /* restore vector */
void _interrupt_far inthandler( void ); /* interrupt handler */
void ( _interrupt_far *OrgVect )(); /* original vector */

/* ----- Initialize ----- */
void Initialize( void )
{
    unsigned int i, j;
    int dummy;

    for(i = 0; i < CH; i++) {
        outp( ADRA[i]+0x3, 0x80 ); /* Line Control Register (DLAB SET) */
        outp( ADRA[i]+0x0, BPS & 0xff ); /* Divisor Latch (Baud Rate SET) */
        outp( ADRA[i]+0x1, (BPS >> 8) & 0xff );
        outp( ADRA[i]+0x3, 0x0 ); /* Line Control Register (DLAB RESET) */

        outp( ADRA[i]+0x3, 0x7 ); /* Line Control Register */
        /* ( Bit 0,1 ... Word Length = 8 bit ) */
        /* ( Bit 2 ..... 2 stop bit ) */
        /* hardware wait */

        for(j = 0; j < 10; j++)
            dummy = inp( 0x2ef );
    }
}

```

```

        outp( ADR[i]+0x4, 0x3 ); /* MODEM Control Register (DTR . RTS) */
        dummy = inp( ADR[i]+0x5 ); /* Line Status Register (Dummy Read) */
        dummy = inp( ADR[i]+0x0 ); /* Receiver Buffer Register (Dummy Read) */
        dummy = inp( ADR[i]+0x2 ); /* Interrupt ID Register (Dummy Read) */
        dummy = inp( ADR[i]+0x6 ); /* Modem Status Register (Dummy Read) */
        outp( ADR[i]+0x1, 0x1 ); /* Interrupt Enable Register */
    }
    outp( ADR[0]+0x1f, 0x80 ); /* Select Clock Mode */
}

/* ----- change vector ----- */
void ChgVect( void )
{
    OrgVect = _dos_getvect( IntVector[IrqLevel] );
    _disable();
    _dos_setvect( IntVector[IrqLevel], inthandler );
    outp( 0x21, ( OrgMasterImr = inp( 0x21 ) ) & PicMask[IrqLevel] );
    outp( 0x20, IsrClear[IrqLevel] ); /* ISR clear */
    _enable(); /* enable */
}

/* ----- restore vector ----- */
void ResVect( void )
{
    _disable(); /* disable */
    outp( 0x21, OrgMasterImr );
    _dos_setvect( IntVector[IrqLevel], OrgVect ); /* restore orgvect */
    _enable(); /* enable */
}

/* ----- interrupt handler ----- */
void _interrupt_far inthandler( void )
{
    unsigned charsts;

    _enable(); /* enable */
    do {
        sts = (unsigned char)inp( ADR[0]+0x1f ); /* Int Vector Register */
        if ( !(sts & 0x1) ) { /* CH0 status */
            while( inp( ADR[0]+0x05 ) & 0x1 ) {
                rbuf[0][intcnt[0]] = (unsigned char)inp( ADR[0]+0x0 );
                intcnt[0]++; /* count times of an interrupt */
            }
        }
        if ( !(sts & 0x2) ) { /* CH1 status */
            while( inp( ADR[1]+0x05 ) & 0x1 ) {
                rbuf[1][intcnt[1]] = (unsigned char)inp( ADR[1]+0x0 );
                intcnt[1]++; /* count times of an interrupt */
            }
        }
    } while( !(sts & 0xf) );

    _disable(); /* disable */
    outp( 0x20, 0x20 ); /* EOI */
}

```

```
/* ----- main ----- */
void main( void )
{
    unsigned int i, j;

    Initialize();                /* initialize */
    ChgVect();                   /* change vector */

    for(j = 0; j < NUM; j++) {   /* send CH0 to CH1 */
        while( !(inp( ADR[0]+0x5 ) & 0x20) ); /* transmitter empty */
        outp( ADR[0]+0x0, j );
    }
    while(intcnt[1] < 10)
        printf("Interrupt CH0:%02d CH1:%02d¥n", intcnt[0], intcnt[1]);
    printf("¥n");

    for(j = 0; j < NUM; j++) {   /* send CH1 to CH0 */
        while( !(inp( ADR[1]+0x5 ) & 0x20) ); /* transmitter empty */
        outp( ADR[1]+0x0, j );
    }
    while(intcnt[0] < 10)
        printf("Interrupt CH0:%02d CH1:%02d¥n", intcnt[0], intcnt[1]);
    printf("¥n");

    ResVect();                  /* restore vector */
    for(j = 0; j < NUM; j++) {
        printf(" CH0:%02x -> CH1:%02x ", j, rbuf[1][j]);
        if(rbuf[1][j] != j) printf("Verify Error ¥n"); /* verify data */
        else printf("¥n");
    }
    printf("¥n");
    for(j = 0; j < NUM; j++) {
        printf(" CH1:%02x -> CH0:%02x ", j, rbuf[0][j]);
        if(rbuf[0][j] != j) printf("Verify Error ¥n");
        else printf("¥n");
    }
    printf("¥n");
    printf("Interrupt CH0:%02d CH1:%02d¥n", intcnt[0], intcnt[1]);
}

/* ----- End of file --- */
```

第6章 システムリファレンス

ブロック図

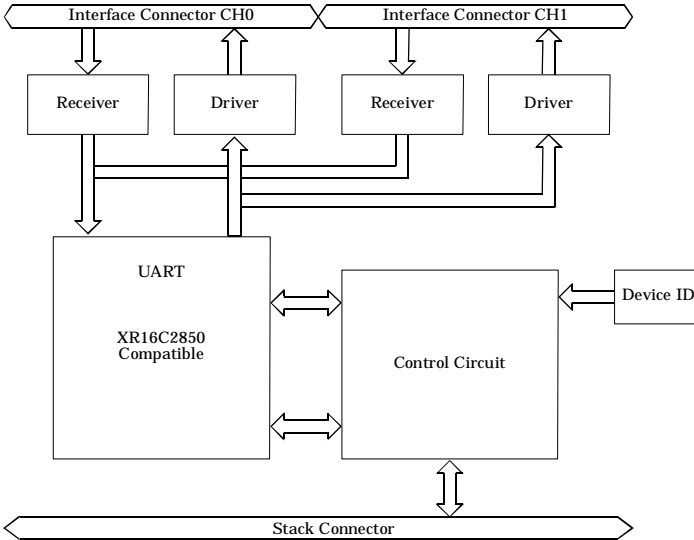


図6.1 ブロック図

等価回路

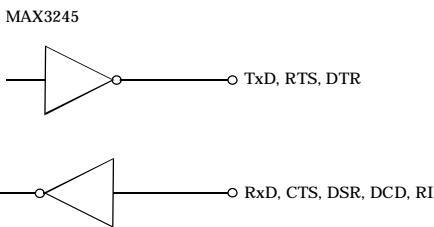


図6.2 等価回路


仕様

表6.1 仕様

項目	仕様
チャンネル数	2ch
入出力仕様	RS-232C
伝送方式	非同期シリアル伝送
ボーレート	2 ~ 921,600bps *1 *2
データ長	5、6、7、8ビット 1、1.5、2ストップビット *1
パリティチェック	イーブン、オッド、ノーパリティ *1
搭載LSI	162850相当品 (FIFO/バッファは、送信用128byte、受信用128byte)
割り込み	1点または2点使用
内部消費電流	5VDC±5% 100mA (Max.)
信号延長可能距離	15m
外形寸法(mm)	25.2(W)×64.7(D)×94.0(H) (ただし、突起物は含まない)
モジュール本体の質量	100g
モジュール接続方法	本体に標準装備されている連結機構によるスタック接続
モジュール設置方法	35mmDINレールにワンタッチ取り付け(取り付け機構は本体に標準装備)

*1 ソフトウェアによって設定することができます。

*2 高速でデータ伝送する場合、外部機器やケーブル長などの環境によって正常な伝送ができない場合があります。

 注意

コントローラモジュールと接続する際は、内部消費電流を考慮してください。総和が電源ユニットの容量を越える場合は、動作を保証できません。詳細はコントローラモジュールの解説書をご覧ください。

表6.2 設置環境条件

条件項目		条件内容
使用周囲温度		0 ~ 50°C
保存周囲温度		-10 ~ 60°C
周囲湿度		10 ~ 90% RH(ただし、結露しないこと)
浮遊粉塵		特にひどくないこと
腐食性ガス		ないこと
耐ノイズ性	ラインノイズ	ACライン/2kV、信号ライン/1kV (IEC1000-4-4Level 3、EN61000-4-4Level 3)
	静電耐圧	接触/4kV (IEC1000-4-2Level 2、EN61000-4-2Level 2)
		気中/8kV (IEC1000-4-2Level 3、EN61000-4-2Level 3)
耐振動性	掃引耐久	10 ~ 57Hz/片振幅0.15mm、57 ~ 150Hz/2.0G
		X、Y、Z方向各80分(JIS C0040準拠、IEC68-2-6準拠)
耐衝撃性		15G X、Y、Z方向各11ms正弦半波 (JIS C0041準拠、IEC68-2-27準拠)

外形寸法

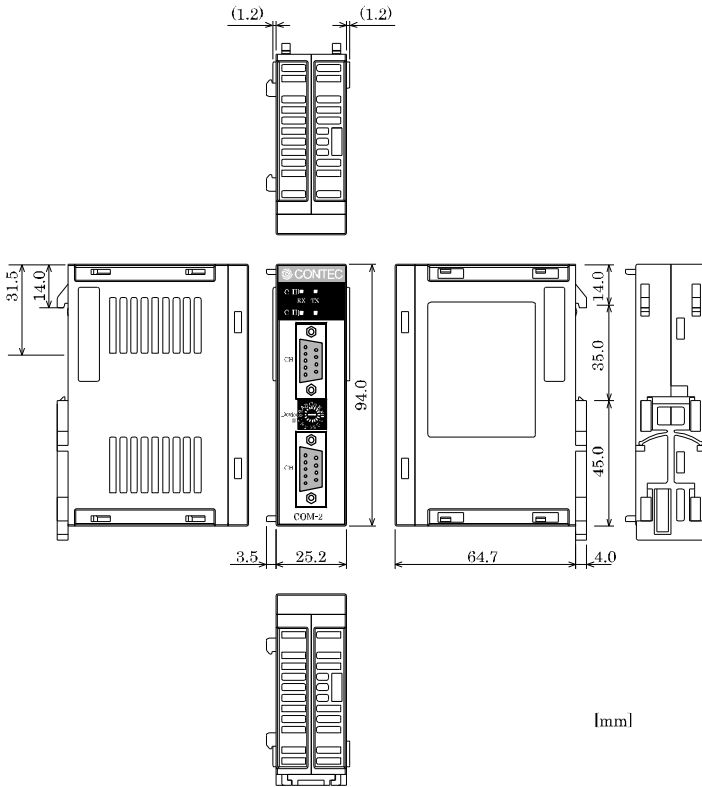


図6.3 外形寸法

COM-2(FIT)GY

解説書

発行 株式会社コンテック

2004年4月改訂

大阪市西淀川区姫里3-9-31 〒555-0025

日本語 <http://www.contec.co.jp/>

英語 <http://www.contec.com/>

中国語 <http://www.contec.com.cn/>

本製品および本書は著作権法によって保護されていますので無断で複写、複製、転載、改変することは禁じられています。

[03282001]

[04132004_rev5]

分類番号

A-40-613

コード番号

LZV6451