MPCH-8CPU -V1

取扱説明書

2011 年 4 月 26 日

株式会社エンベデッドテクノロジー

改訂履歴

改訂日	改訂項目	ページ	改訂内容
2005/8/18	新規作成		第一版発行
2006/03/27	各所		リリース用校正済み
2006/04/23	ポート６		P6 ポート解説修正　　(*3)
2006/05/31	搭載デバイス･拡張バスアドレス	13	ブロック図追加
2006/05/31	システムレジスタ等初期化	37	
2006/11/20	社名変更	表紙	
2011/4/26	リアルタイムクロック	1,3,6,21,25,37,48,60	リアルタイムクロック変更による使用方法
2011/4/26	開発環境		新規追加

はじめに

１．製品の保証について

・ 無償修理
製品ご購入後1年間は無償で修理いたします。
（但し、下記「有償修理」に該当するものを除く）

・ 有償修理
1) 製品ご購入後1年を経過したもの。
2) 製品購入1年以内で故障の原因がお客様の取り扱い上のミスによるもの。
3) 製品購入1年以内で故障の原因がお客様の故意によるもの。

・ 免責事項
当社製品の故障、不具合、誤動作あるいは停電によって生じた損害等の純粋経済損失につきましては、当社は一切その責任を負いかねますので、あらかじめご了承ください。

２．製品について

・ 当社製品はカタログ仕様範囲内において、使用部品、回路図等、予告無く変更することが有ります。
・ 当社製品は部品メーカーの製造中止等によりやむを得ず製品の供給を続けることが出来なくなることが有ります。
・ 当社製品の無断での複製を禁止します。
・ 当社製品は一般商工業用として設計されており生命、財産に関わるような状況下で使用されることを意図して設計、製造されたものではありません．本製品の故障、誤動作が人命を脅かしたり、人体に危害を与えたりする恐れのある用途（生命維持、監視のための医療用）、および高い信頼性が要求される用途（航空・宇宙用、運輸用、海底中継器、原子力制御用、走行制御用、移動体用）にはご利用されないようご注意ください。すべての電子機器はある確率で故障が発生します。当社製品の故障により、人畜や財産が被害を受けたり、火災事故や社会的損害が生じたりしないように安全設計をお願いします。また長時間連続運転や仕様外の環境でのご使用は避けてください。

但し、長時間運転でご使用された場合の故障に付きましては通常どおりの修理保証（1年以内無償、1年以上有償）が受けられます。

３．カタログ、取扱説明書の記載事項について

・ 当社製品のカタログ及び取扱説明書は予告無く変更する場合があります。
・ 取扱説明書に記載されている内容及び回路図の一部又は全部を無断での転載、転用を禁止します。
・ 本資料に記載された情報、回路図は機器の応用例であり動作、性能を保証するものではなく、実際の機器への搭載を目的としたものではありません。またこれらの情報、回路を使用することにより起因する第三者の工業所有権、知的所有権、その他権利侵害に関わる問題が生じた際、当社はその責を負いませんのであらかじめご了承ください。
・ 本製品に搭載されているIC、LSIの使用方法はそれぞれのメーカー発行の取扱説明書、データシートにしたがってください。

４．海外への輸出について

・ 当社製品を使用した機器を海外へ持ち出される場合、当社製品のCOCOMパラメーターシートが必要です。その都度お申しつけ頂ければパラメーターシートを発行いたします。

目　　次

1. 概要

MPCH-8CPU-V1 カード（以下、本カードと記述します）はルネサステクノロジ製高性能フラッシュメモリ内蔵 16 ビットマイコンを核に大容量スタティック RAM、USB コントローラ、LAN コントローラなどを搭載した多機能組み込みマイコンカードです。

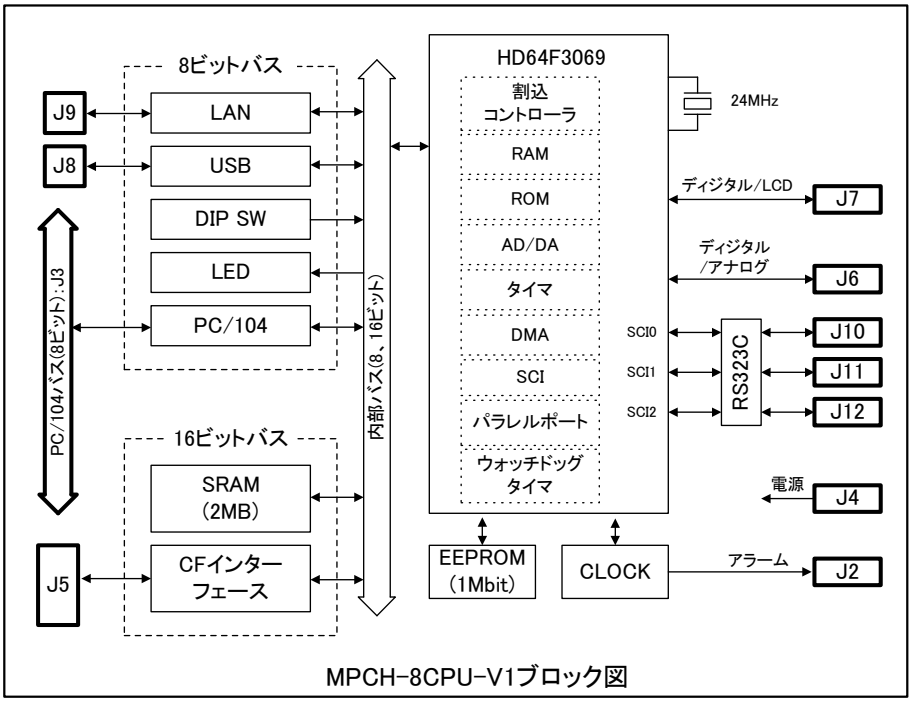
2. 構成

2.1. 仕様

項目	型名・規格	備考
CPU	HD64F3069RF	
クロック	24MHz 水晶発振器	
外部 RAM	2M バイト SRAM	16 ビットバス
汎用 I/O ポート1	PA ポート	LCD モジュール(SUNLIKE 社 SC1602)対応
汎用 I/O ポート2	P7、PA ポート、NMI	ポート1と排他的に使用
シリアルポート	RS232C	3 チャネル・TD、RD のみ
汎用外部バス	PC/104	8ビット I/O のみ、最大4系統割り込み
LAN コントローラ	10BASE-T	RTL8019AS 使用
USB コントローラ	USB1.1 ターゲット	USBN9604 使用
コンパクトフラッシュ	True IDE インターフェース	
ディップスイッチ	8 ビット	汎用
モニタ LED	8 個	汎用
リアルタイムクロック	セイコーエプソン ~~RX4702~~ RX-4571	~~温度センサ内蔵~~
バックアップメモリ	シリアル EEPROM	1M ビット
カードサイズ	145mm×100mm	コネクタ、CF イジェクタ等の突起部分を除く
電源	5V	

2.2. ブロック図

本カードのブロックを示します。



MPCH-8CPU-V1ブロック図

2.3. 接続デバイス

CPU 外部 I/O として、つぎに示すデバイスが接続されております。

2.3.1. 記憶デバイス

種類	容量	備考
SRAM	2M バイト	16 ビットバス接続
シリアル EEPROM	1M ビット	I²C インターフェース
CF（コンパクトフラッシュカード）	カード容量による	True IDE モード 16 ビットバス接続

2.3.2. 通信デバイス

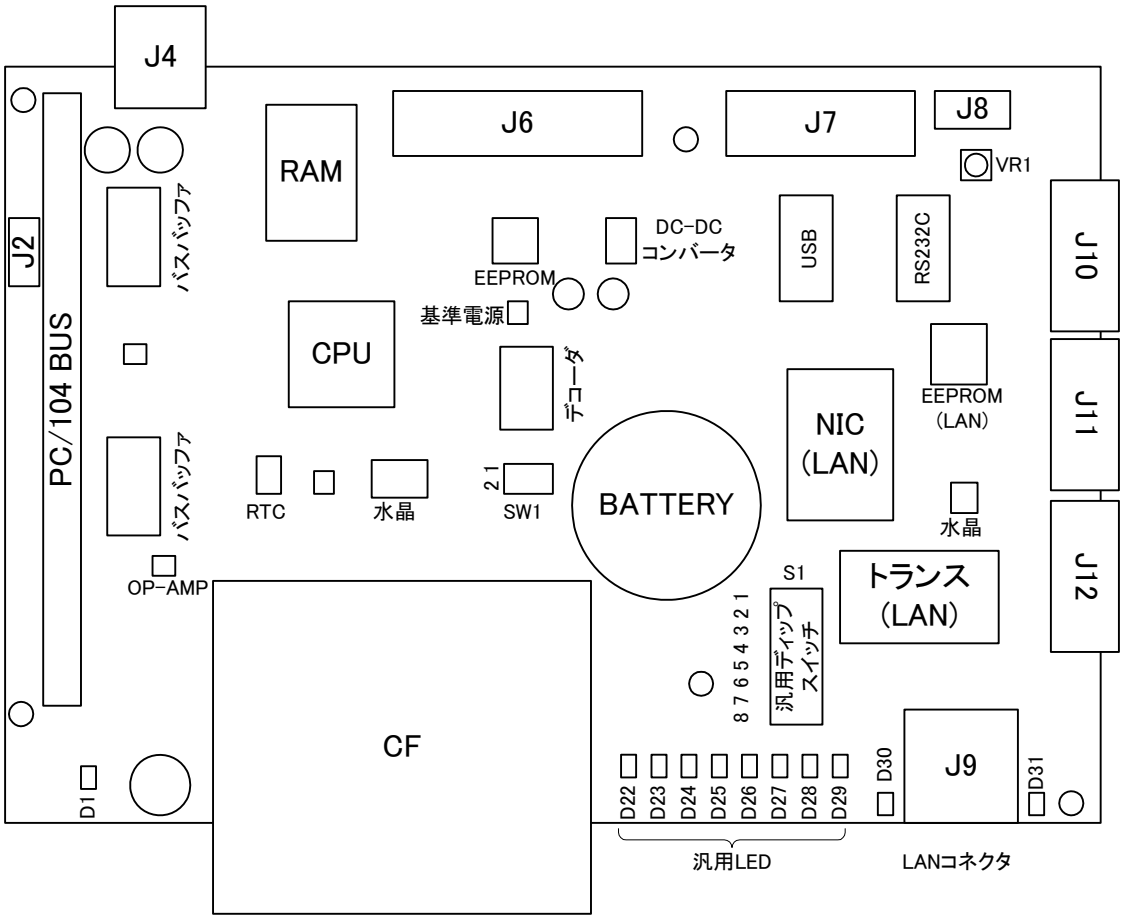
種類	規格	備考
ネットワーク	10BASE-T	REALTEK RTL8019AS
USB	USB1.1	NS USBN9604

2.3.3. その他

種類	規格	備考
外部 I/O バス	PC/104	8 ビットバス
リアルタイムクロック	3 線式シリアル インターフェース	~~温度センサ内蔵~~
ディップスイッチ	8 ビット	負論理（ON=0、OFF=1）
LED	8 ビット	正論理（1=点灯、0=消灯）

2.4. 実装図

主要部品およびコネクタの配置を実装図で示します。



2.5. I/O ポート割当

CPU 内蔵 I/O ポートと各種信号の接続状況を表に示します。

P10～P17

外部バス接続のためアドレスを出力します。

ポート	ピン#	方向	信号名
P10	36	－	アドレス(A0)
P11	37	－	アドレス(A1)
P12	38	－	アドレス(A2)
P13	39	－	アドレス(A3)
P14	40	－	アドレス(A4)
P15	41	－	アドレス(A5)
P16	42	－	アドレス(A6)
P17	43	－	アドレス(A7)

P20～P27

外部バス接続のためアドレスを出力します。

ポート	ピン#	方向	信号名
P20	45	－	アドレス(A8)
P21	46	－	アドレス(A9)
P22	47	－	アドレス(A10)
P23	48	－	アドレス(A11)
P24	49	－	アドレス(A12)
P25	50	－	アドレス(A13)
P26	51	－	アドレス(A14)
P27	52	－	アドレス(A15)

P30～P37

外部バス接続のためデータバスで使用します。

ポート	ピン#	方向	信号名
P30	27	－	データ(D8)
P31	28	－	データ(D9)
P32	29	－	データ(D10)
P33	30	－	データ(D11)
P34	31	－	データ(D12)
P35	32	－	データ(D13)
P36	33	－	データ(D14)
P37	34	－	データ(D15)

P40～P47

外部バス接続のためデータバスで使用します。

ポート	ピン#	方向	信号名
P40	18	－	データ(D0)
P41	19	－	データ(D1)
P42	20	－	データ(D2)
P43	21	－	データ(D3)
P44	23	－	データ(D4)
P45	24	－	データ(D5)
P46	25	－	データ(D6)
P47	26	－	データ(D7)

P50～P53

外部バス接続のためアドレスを出力します。

ポート	ピン#	方向	信号名
P50	53	－	アドレス(A16)
P51	54	－	アドレス(A17)
P52	55	－	アドレス(A18)
P53	56	－	アドレス(A19)

P60～P67

リアルタイムクロック制御信号と CPU バス制御信号に使用しています。

ポート	ピン#	方向	信号名
P60	58	I/O	リアルタイムクロック制御((DATA)
P61	59	OUT	リアルタイムクロック制御(CLK)
P62	60	OUT	リアルタイムクロック制御(CE)
P63	69	－	CPU バス制御信号(AS#)　　(*3)
P64	70	－	CPU バス制御信号(RD#)
P65	71	－	CPU バス制御信号(HWR#)
P66	72	－	CPU バス制御信号(LWR#)
P67	61	IN	汎用入力ポート　　(*3)

P70～P77

汎用 I/O としてコネクタに接続しています。

ポート	ピン#	方向	信号名
P70	78	I/O	AD 変換入力/汎用ポート(コネクタ接続)~~または温度センサ接続~~
P71	79	I/O	AD 変換入力/汎用ポート(コネクタ接続)
P72	80	I/O	AD 変換入力/汎用ポート(コネクタ接続)
P73	81	I/O	AD 変換入力/汎用ポート(コネクタ接続)
P74	82	I/O	AD 変換入力/汎用ポート(コネクタ接続)
P75	83	I/O	AD 変換入力/汎用ポート(コネクタ接続)
P76	84	I/O	AD 変換入力/ DA 変換出力/汎用ポート(コネクタ接続)
P77	85	I/O	AD 変換入力/ DA 変換出力/汎用ポート(コネクタ接続)

P80 から P84

割込入力と、CF 検出に使用しています。

ポート	ピン#	方向	信号名
P80	87	－	USB コントローラ割込
P81	88	－	コンパクトフラッシュカード割込
P82	89	－	PC/104 バス割込(IRQ7)
P83	90	－	PC/104 バス割込(IRQ3)
P84	91	IN	コンパクトフラッシュカード挿入検出(LOW で挿入)

P90～P95

シリアル通信と割込入力に使用しています。

ポート	ピン#	方向	信号名
P90	12	－	シリアルコミュニケーションインターフェース 0(TxD0)J10
P91	13	－	シリアルコミュニケーションインターフェース 1(TxD1)J11
P92	14	－	シリアルコミュニケーションインターフェース 0(RxD0)J10
P93	15	－	シリアルコミュニケーションインターフェース 1(RxD1)J11
P94	16	－	PC/104 バス割込(IRQ4)または LAN コントローラ割込
P95	17	－	PC/104 バス割込(IRQ5)

PA0～PA6

汎用 I/O としてコネクタに接続しています。

ポート	ピン#	方向	信号名
PA0	93	I/O	汎用ポート(コネクタ接続)
PA1	94	I/O	汎用ポート(コネクタ接続)
PA2	95	I/O	汎用ポート(コネクタ接続)
PA3	96	I/O	汎用ポート(コネクタ接続)
PA4	97	I/O	汎用ポート(コネクタ接続)
PA5	98	I/O	汎用ポート(コネクタ接続)
PA6	99	I/O	汎用ポート(コネクタ接続)

PA7

外部バス接続のためアドレスを出力します。

ポート	ピン#	方向	信号名
PA7	100	－	アドレス(A20)

PB0～PB7

シリアル EEPROM、チップセレクト、SCI２などに使用します。

ポート	ピン#	方向	信号名
PB0	2	I/O	シリアル EEPROM(SDA)
PB1	3	－	CPU チップセレクト信号(SRAM 選択)
PB2	4	－	CPU チップセレクト信号(コンパクトフラッシュカード選択)
PB3	5	－	CPU チップセレクト制御信号(PC/104、USB、LAN、DIPSW、LED 選択)
PB4	6	OUT	シリアル EEPROM(SCL)
PB5	7	OUT	内蔵フラッシュメモリ書き込みイネーブル
PB6	8	－	シリアルコミュニケーションインターフェース 2(TxD2)J12
PB7	9	－	シリアルコミュニケーションインターフェース 2(RxD2)J12

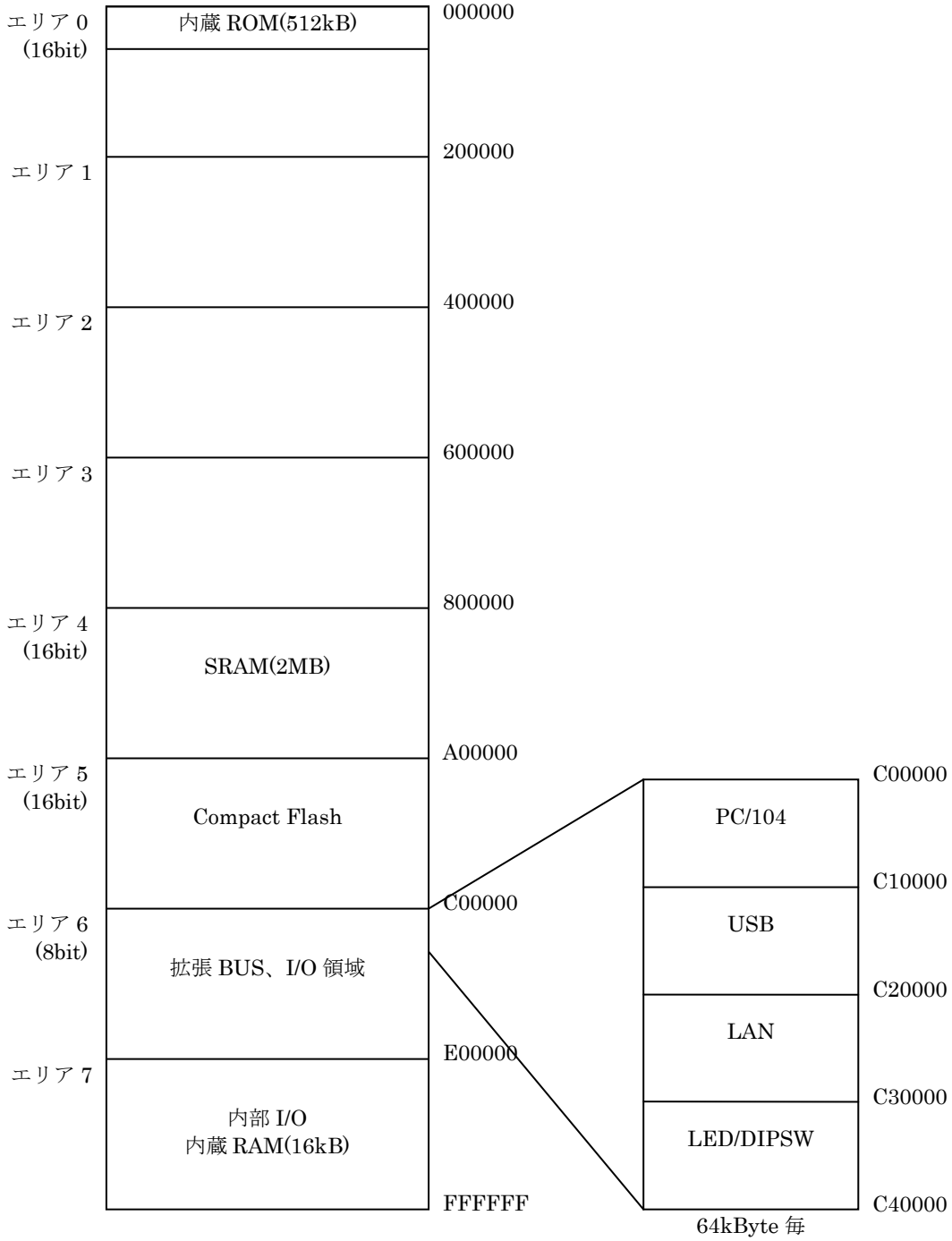
2.6. CPU 端子割当表

PIN#	CPU 端子名	接続先	信号名
1	VCL	バイパスコンデンサ(0.1μF)	
2	PB0/TP8/TMO0/CS7#	シリアル EEPROM	SDA
3	PB1/TP9/TMO1/DREQ0#/CS6#	PC/104,USB,LED,SW,LAN	CS6#
4	PB2/TP10/TMO2/CS5#	コンパクトフラッシュ	CS5#
5	PB3/TP11/TMO3/DREQ4#/CS4#	SRAM	CS4#
6	PB4/TP12/UCAS#	シリアル EEPROM	SCL
7	PB5/TP13/LCAS#/SCK2	フラッシュメモリ書き込み回路	PBFWE
8	PB6/TP14/TxD2	シリアル通信 SCI2	TxD2
9	PB7/TP15/RxD2	シリアル通信 SCI2	RxD2
10	FWE	モード設定回路	FWE
11	VSS		GND
12	P90/TxD0	シリアル通信 SCI0	TxD0
13	P91/TxD1	シリアル通信 SCI1	TxD1
14	P92/RxD0	シリアル通信 SCI0	RxD0
15	P93/RxD1	シリアル通信 SCI1	RxD1
16	P94/SCK0/IRQ4	PC/104-IRQ4 または LAN 割り込み	IRQ4
17	P95/SCK1/IRQ5	PC/104-IRQ5 割込	IRQ
18	P40/D0	データバス	D0
19	P41/D1		D1
20	P42/D2		D2
21	P43/D3		D3
22	VSS		GND
23	P44/D4	データバス	D4
24	P45/D5		D5
25	P46/D6		D6
26	P47/D7		D7
27	P30/D8		D8
28	P31/D9		D9
29	P32/D10		D10
30	P33/D11		D11
31	P34/D12		D12
32	P35/D13		D13
33	P36/D14		D14
34	P37/D15		D15
35	VCC		+5V
36	P10/A0	アドレスバス	A0
37	P11/A1		A1
38	P12/A2		A2
39	P13/A3		A3
40	P14/A4		A4
41	P15/A5		A5
42	P16/A6		A6
43	P17/A7		A7
44	VSS		GND
45	P20/A8	アドレスバス	A8
46	P21/A9		A9
47	P22/A10		A10
48	P23/A11		A11
49	P24/A12		A12
50	P25/A13		A13
51	P26/A14		A14
52	P27/A15		A15
53	P50/A16		A16

54	P51/A17		A17
55	P52/A18		A18
56	P53/A19		A19
57	VSS		GND
58	P60/WAIT#	リアルタイムクロック	DATA
59	P61/BREQ#	リアルタイムクロック	CLK
60	P62/BACK#	リアルタイムクロック	CE#
61	P67/ϕ	汎用入力ポート	P67
62	STBY#	プルアップ	
63	RES#	リセット	
64	NMI	コネクタ	NMI
65	VSS		GND
66	EXTAL	クロック入力	CLK
67	XTAL	オープン	
68	VCC		+5V
69	P63/AS#	CPU 制御信号	AS# (*3)
70	P64/RD#	外部 SRAM、外部 I/O 等	RD#
71	P65/HWR#	外部 SRAM、外部 I/O 等	HWR#
72	P66/LWR#	外部 SRAM、コンパクトフラッシュ	LWR#
73	MD0	モード設定回路	VCC
74	MD1		GND
75	MD2		VCC/GND
76	AVCC		
77	VREF	基準電源	
78	P70/AN0	コネクタ (ディジタル/アナログ)	P70
79	P71/AN1		P72
80	P72/AN2		P73
81	P73/AN3		P73
82	P74/AN4		P74
83	P75/AN5		P75
84	P76/AN6/DA0		P76
85	P77/AN7/DA1		P77
86	AVSS		GND
87	P80/IRQ0#/RFSH#	USB 割込	IRQ0
88	P81/IRQ1#/CS3#	コンパクトフラッシュ割り込み	IRQ1
89	P82/IRQ2#/CS2#	PC/104-IRQ7 割込	IRQ7
90	P83/IRQ3#/CS1#	PC/104-IRQ3 割込	IRQ3
91	P84/CS0#	コンパクトフラッシュカード挿入検出	CF_DET#
92	VSS		GND
93	PA0/TP0/TCLKA/TEND0#	コネクタ (ディジタル/LCD)	PA0
94	PA1/TP1/TCLKB/TEND1#		PA1
95	PA2/TP2/TIOCA0/TCLKC		PA2
96	PA3/TP3/TIOCB0/TCLKD		PA3
97	PA4/TP4/TIOCA1/A23		PA4
98	PA5/TP5/TIOCB1/A22		PA5
99	PA6/TP6/TIOCA2/A21		PA6
100	PA7/TP7/TIOCB2/A20	アドレスバス	A20

3. アドレスマップ

HD64F3069(CPU)はアドバンスモードのみで動作し、1M バイトまたは 16M バイトのアドレス空間をリニアにアクセスすることが可能です。本カードの設定では 16M バイトモードでのみ動作します。本カードは様々なデバイスが接続されておりますが、バス接続されているデバイスのアドレスマップを示します。



4. CPU 内部レジスタ解説

4.1. P1：アドレスバス

アドレスポートとして使用するため DDR は出力設定します。

FEE000	P1DDR							
bit	7	6	5	4	3	2	1	0
名称	P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR
設定値	1	1	1	1	1	1	1	1

4.2. P2：アドレスバス

アドレスポートとして使用するため DDR は出力設定します。

FEE001	P2DDR							
bit	7	6	5	4	3	2	1	0
名称	P27DDR	P26DDR	P25DDR	P24DDR	P23DDR	P22DDR	P21DDR	P20DDR
設定値	1	1	1	1	1	1	1	1

4.3. P3：データバス

FEE002	P3DDR							
bit	7	6	5	4	3	2	1	0
名称	P37DDR	P36DDR	P35DDR	P34DDR	P33DDR	P32DDR	P31DDR	P30DDR
設定値	－	－	－	－	－	－	－	－

4.4. P4：データバス

FEE003	P4DDR							
bit	7	6	5	4	3	2	1	0
名称	P47DDR	P46DDR	P45DDR	P44DDR	P43DDR	P42DDR	P41DDR	P40DDR
設定値	－	－	－	－	－	－	－	－

4.5. P5：アドレスバス

FEE004	P5DDR							
bit	7	6	5	4	3	2	1	0
名称					P53DDR	P52DDR	P51DDR	P50DDR
設定値	－	－	－	－	1	1	1	1

4.6. P6：CPU バス制御信号、ポート

FEE005	P6DDR							
bit	7	6	5	4	3	2	1	0
名称	P67DDR	P66DDR	P65DDR	P64DDR	P63DDR	P62DDR	P61DDR	P60DDR
設定値	0	－	－	－	－	1	1	0/1

4.7. P7：汎用ポート

FFFFD6	P7DR							
bit	7	6	5	4	3	2	1	0
名称	P77	P76	P75	P74	P73	P72	P71	P70
設定値	－	－	－	－	－	－	－	－

4.8. P8：割り込み、出力ポート

FEE007	P8DDR							
bit	7	6	5	4	3	2	1	0
名称				P84DDR	P83DDR	P82DDR	P81DDR	P80DDR
設定値	－	－	－	0	0	0	0	0

FFFFD7	P8DR							
bit	7	6	5	4	3	2	1	0
名称				P84	P83	P82	P81	P80
設定値	－	－	－	0/1	－	－	－	－

4.9. P9：シリアルコミュニケーションインターフェース、割り込み

FEE008	P9DDR							
bit	7	6	5	4	3	2	1	0
名称			P95DDR	P94DDR	P93DDR	P92DDR	P91DDR	P90DDR
設定値	－	－	－	1	0	0	0	0

4.10. PA：汎用ポート、アドレスバス

FEE009	PADDR							
bit	7	6	5	4	3	2	1	0
名称	PA7DDR	PA6DDR	PA5DDR	PA4DDR	PA3DDR	PA2DDR	PA1DDR	PA0DDR
設定値	－	－	－	1	0	0	0	0

4.11. PB：チップセレクト信号、シリアルコミュニケーションインターフェース、ポート

FEE00A	PBDDR							
bit	7	6	5	4	3	2	1	0
名称	PB7DDR	PB6DDR	PB5DDR	PB4DDR	PB3DDR	PB2DDR	PBDDR	PB0DDR
設定値	0	0		1				0/1

FFFFDA	PBDR							
bit	7	6	5	4	3	2	1	0
名称	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
設定値					－	－	－	－

FEE01C	MSTCRH							
bit	7	6	5	4	3	2	1	0
名称	PSTOP	－	－	－	－	MSTPH2	MSTPH1	MSTPH0
設定値	1					0	0	0

FEE013	BRCR							
bit	7	6	5	4	3	2	1	0
名称	A23E	A22E	A21E	A20E	－	－	－	BRLE
設定値					1	1	1	0

FEE024	BCR							
bit	7	6	5	4	3	2	1	0
名称	ICIS1	ICIS0	BROME	BRSTS1	BRSTS0	EMC	RDEA	WAITE
設定値								0

FEE026	DRCRA							
bit	7	6	5	4	3	2	1	0
名称	DRAS2	DRAS1	DRAS0		BE	RDM	SRFMD	RFSHE
設定値	0	0	0	－	0	0	0	0

5. 搭載デバイス・拡張バスアドレス

本カードは様々な規格のインターフェースとデータ交換が容易に行えるよう、多くのデバイスを搭載しております。
各デバイスの特徴と接続アドレスは以下のとおりです。

5.1. コンパクトフラッシュメモリインターフェース

本カードは外部記憶装置としてコンパクトフラッシュメモリ（CF）が搭載可能です。CF は True IDE モードで接続します。
CPU からみたレジスタのアドレスは下記一覧表のとおりです。レジスタ詳細はコンパクトフラッシュメモリカードの仕様書をご参考にしてください。

5.1.1. アドレス割り当て

IDE コマンドブロックレジスタ(CS0 側)

アドレス	レジスタ(Read 時)	レジスタ(Write 時)
A00000,1	Data register	Data register
A00003	Error register	Feature register
A00005	Sector count register	Sector count register
A00007	Sector number register	Sector number register
A00009	Cylinder low register	Cylinder low register
A0000B	Cylinder high register	Cylinder high register
A0000D	Drive head register	Drive head register
A0000F	Status register	Command register

A00000 番地のみ 16 ビットレジスタです。他のアドレスは８ビットレジスタです。例：A00002 番地を 16 ビットアクセスで読み込むと A00003 番地（下位８ビット）のデータが有効となります。

IDE コントロールブロックレジスタ(CS1 側)

アドレス	レジスタ(Read 時)	レジスタ(Write 時)
A00801	無効	無効
A00803	無効	無効
A00805	無効	無効
A00807	無効	無効
A00809	無効	無効
A0080B	無効	無効
A0080D	無効	無効
A0080F	Alternate Status register	Device Control register

このレジスタは８ビットレジスタです。16 ビットでアクセスすると下位８ビットのみ有効です。例：A0080E 番地を読み込むと A0080F 番地（下位８ビット）のデータが有効となります。

電源制御

アドレス	レジスタ(Read 時)	レジスタ(Write 時)
A01000	禁止	CF 電源制御 1 を書き込むと電源 ON、 0 を書き込むと電源 OFF

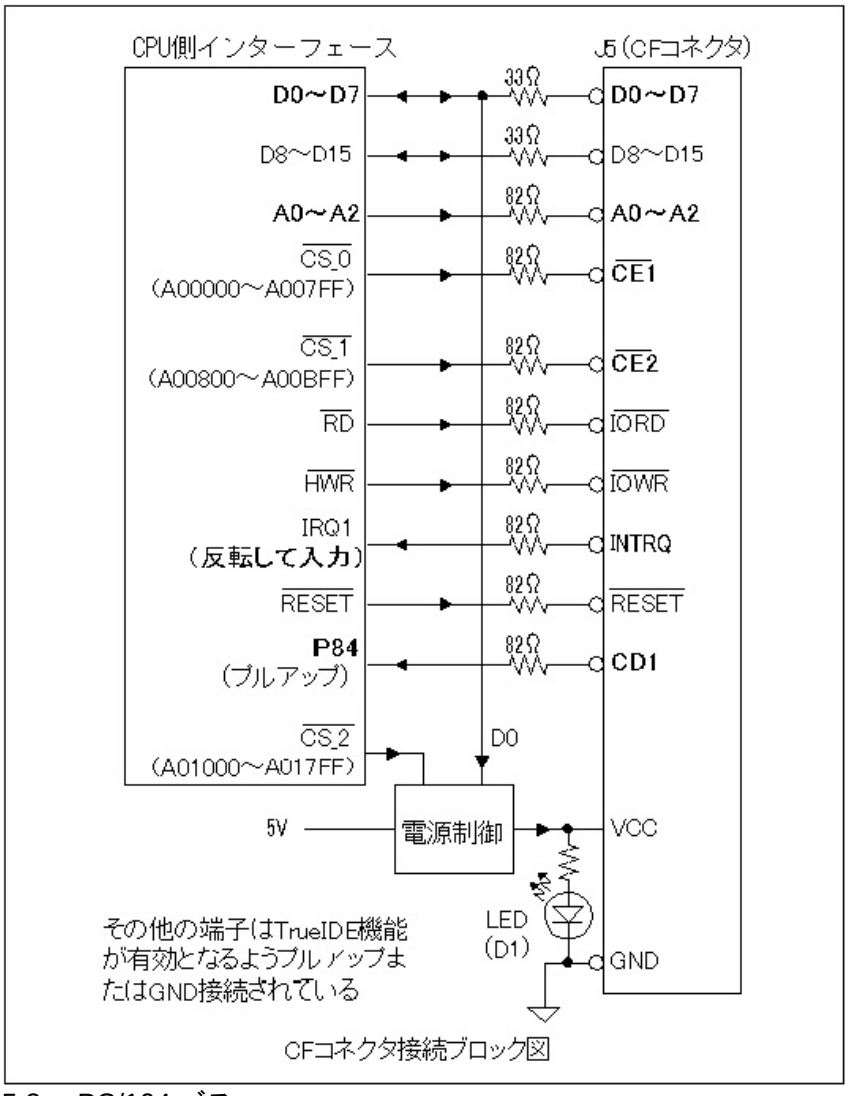
カード挿入検出

アドレス	ビット	レジスタ(Read 時)	レジスタ(Write 時)
FFFFD7	4	0:カード検出、1:未挿入	禁止

5.1.2. 割り込み

カードの INTRQ を反転して CPU の IRQ1 に接続

5.1.3. ブロック図



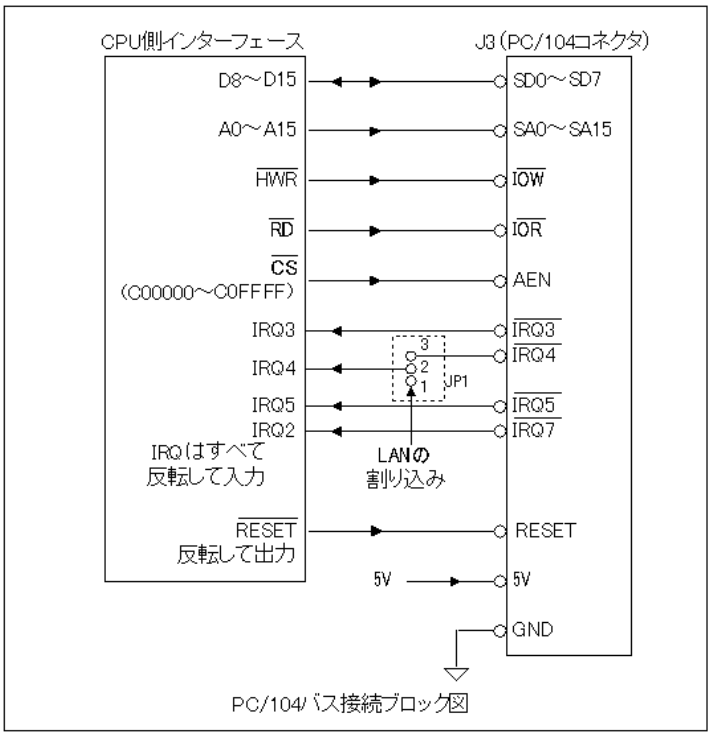
5.2. PC/104 バス

本カードは機能拡張のため、PC/104 規格のカードを搭載可能です。PC/104 規格は約 90mm×90mm のコンパクトな拡張カードでパラレル I/O、AD コンバータ、DA コンバータなど多種類の I/O カードがあります。本カードは 8 ビット I/O バスモードで使用し、割込は最大 4 系統利用できます。

5.2.1. アドレス割り当て

アドレス	レジスタ(Read 時)	レジスタ(Write 時)
C00000～ C0FFFF	PC/104 バス、I/O アドレスの 0～FFFF 番地にマップされます	

5.2.2 ブロック図

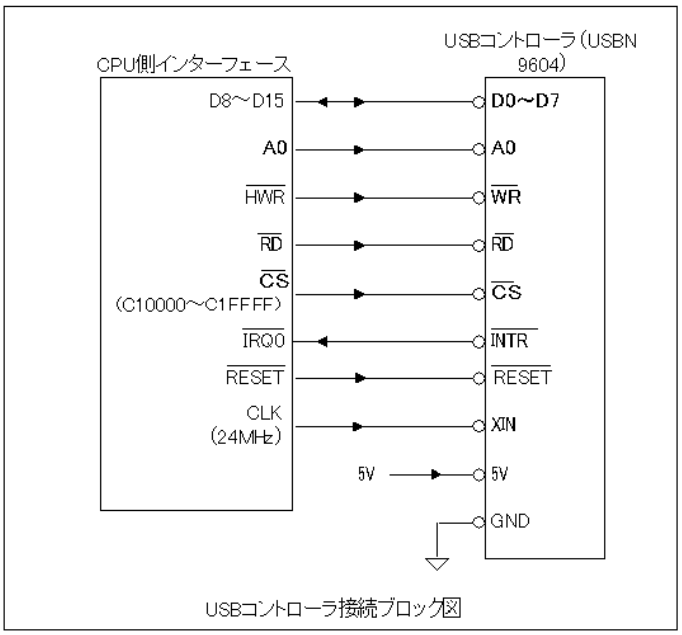


5.3. USB コントローラ

5.3.1. アドレス割り当て

アドレス	レジスタ(Read 時)	レジスタ(Write 時)
C10000、 C10001	USBN9604 のマニュアルを参照してください	

5.3.2 ブロック図

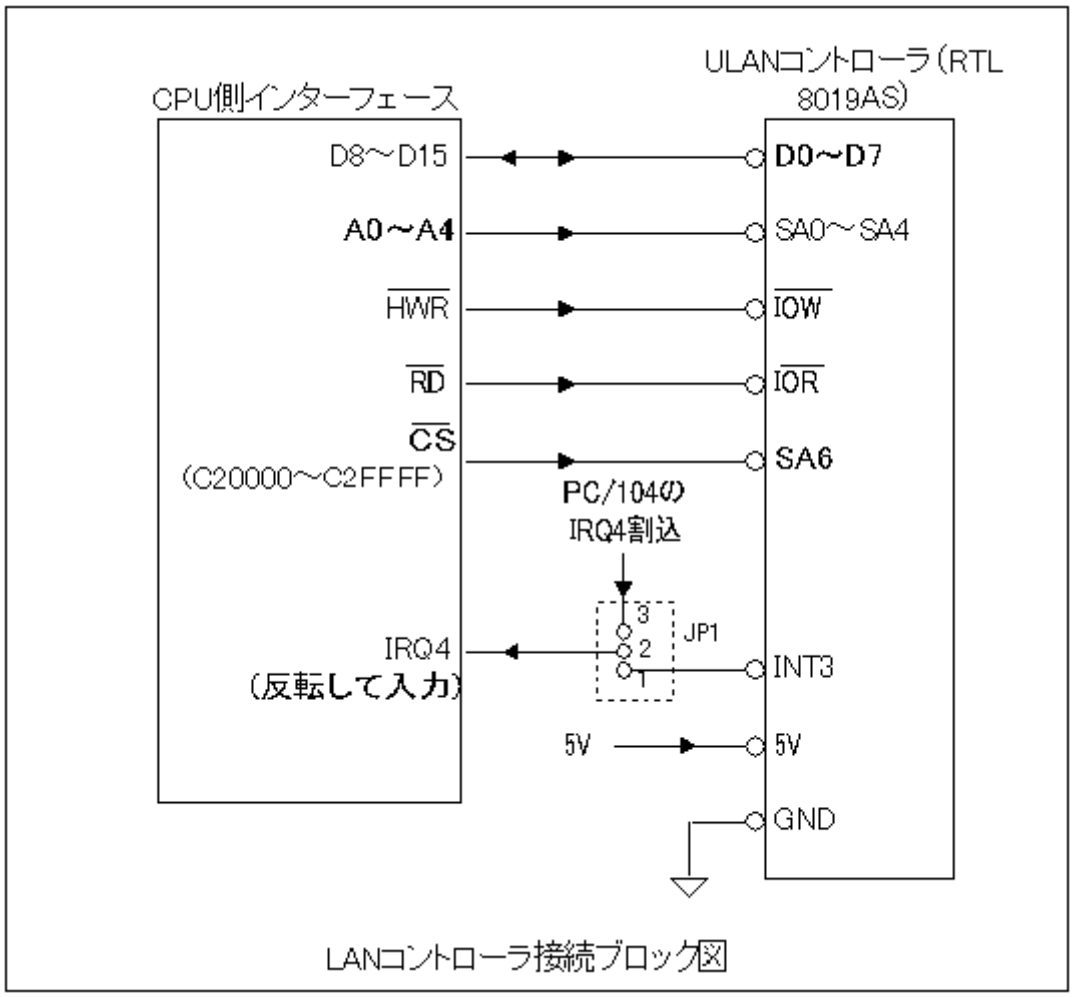


5.4. LAN コントローラ

5.4.1. アドレス割り当て

アドレス	レジスタ(Read 時)	レジスタ(Write 時)
C20000～ C2000F	RTL8019AS のマニュアルを参照してください	

5.4.2 ブロック図



5.5. ディップスイッチ

本カードは動作確認、パラメータ設定、デバッグ等、汎用で利用できるディップスイッチを搭載しています。

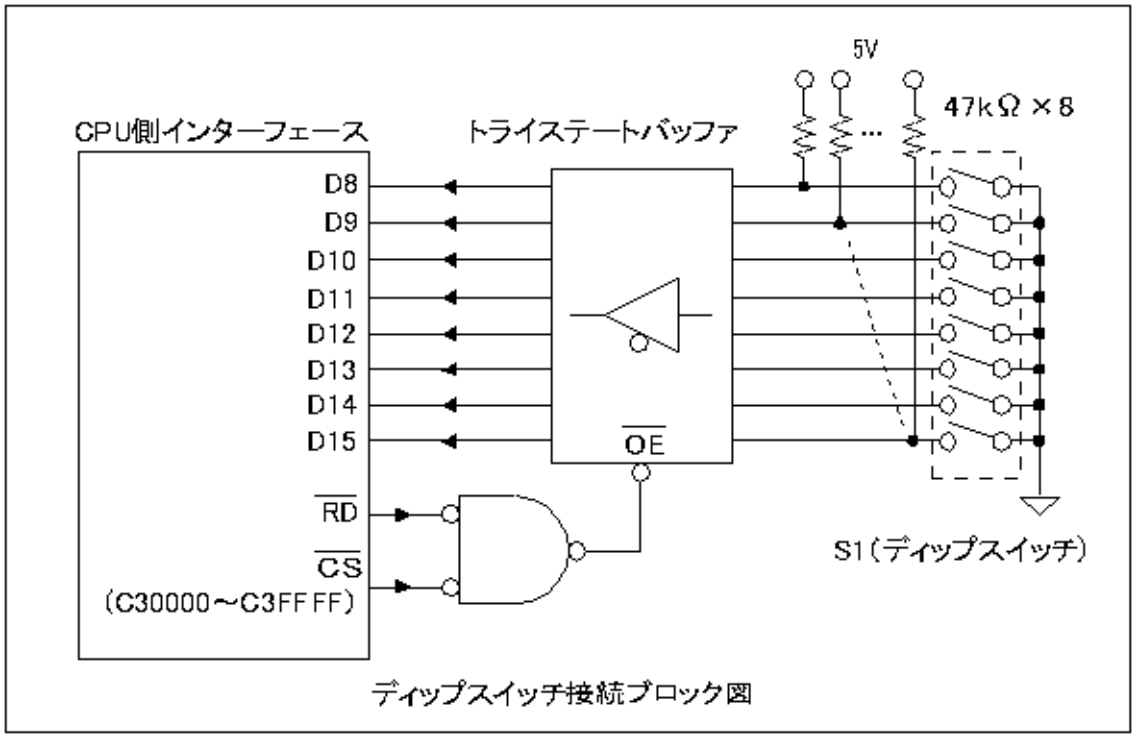
5.5.1. アドレス割り当て

アドレス、スイッチの読み込み論理を次に示します。

アドレス：C30000

ビット	D7	D6	D5	D4	D3	D2	D1	D0
番号	8	7	6	5	4	3	2	1
OFF 状態	1							
ON 状態	0							

5.5.2 ブロック図



ディップスイッチ接続ブロック図

5.6. LED

本カードは動作確認、デバッグ等、汎用で利用できる LED を搭載しています。

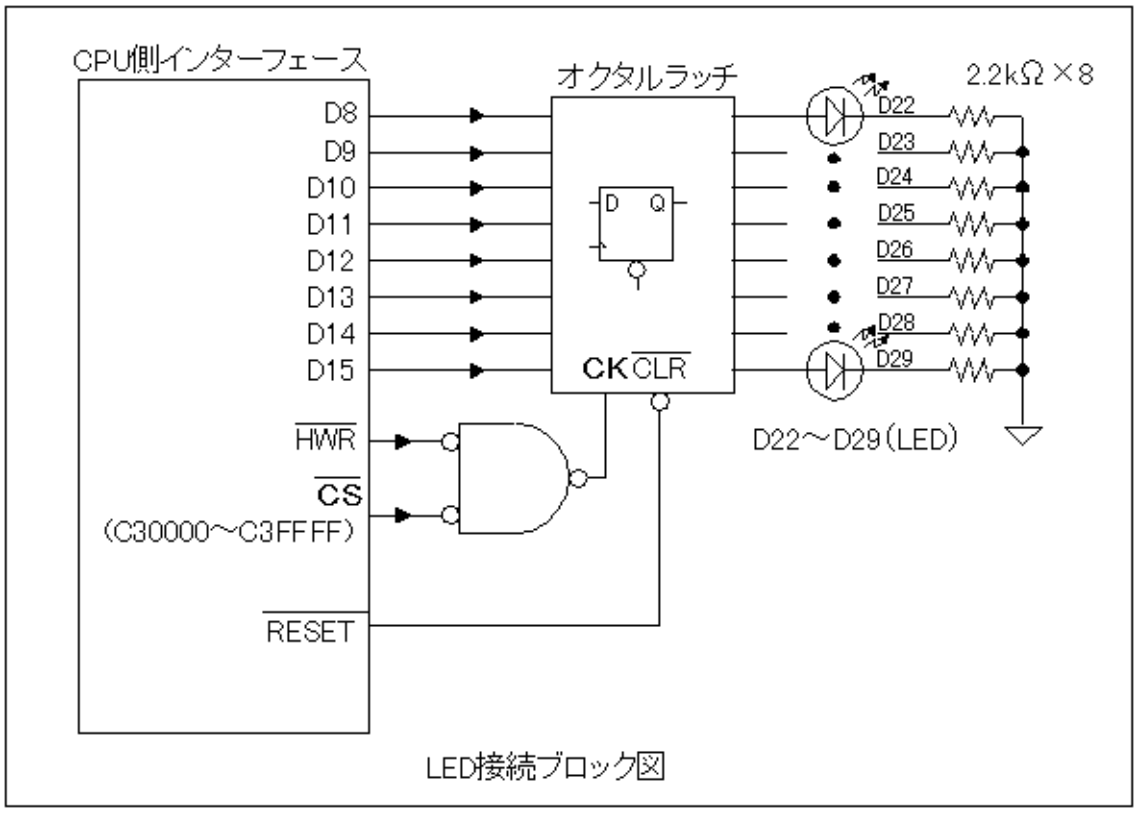
5.6.1. アドレス割り当て

アドレス、LED の点灯論理を次に示します。

アドレス：C30000

ビット	D7	D6	D5	D4	D3	D2	D1	D0
LED 番号	D29	D28	D27	D26	D25	D24	D23	D22
0 を書き込み	消灯							
1 を書き込み	点灯							

5.6.2 ブロック図



LED接続ブロック図

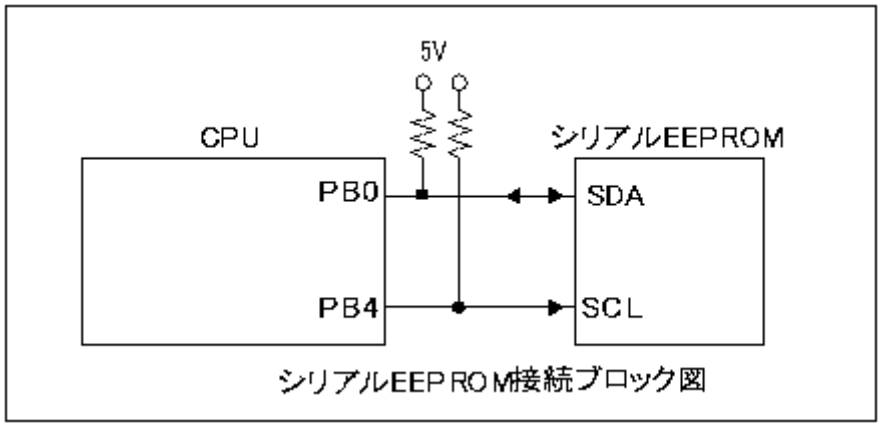
5.7. シリアル EEPROM

最大 1M ビットのシリアル EEPROM を搭載します。インターフェースは I²C バスで、CPU の I/O ポートに接続しています。

5.7.1. ポート割り当て

信号名	CPU 接続先	方向
SCL	PB4(6 番ピン)	CPU→EEPROM
SDA	PB0(2 番ピン)	CPU←→EEPROM

5.7.2. ブロック図



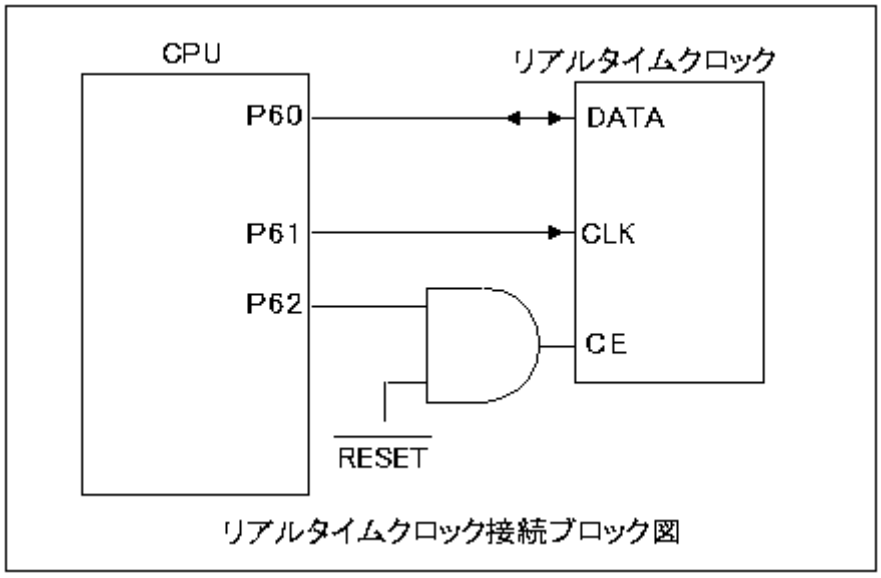
5.8. リアルタイムクロック

バッテリバックアップされたリアルタイムクロック(RTC)を搭載します。

5.8.1. ポート割り当て

信号名	CPU 接続先	方向
DATA	P60(58 番ピン)	CPU←→RTC
CLK	P61(59 番ピン)	CPU→RTC
CE	P62(60 番ピン)	CPU→RTC

5.8.2. ブロック図



リアルタイムクロック接続ブロック図

6. ボード設定

6.1. ジャンパ設定

JP1：IRQ4 入力切り替え

ショート位置	CPU の 16 番ピンの接続先
1-2	LAN コントローラ割り込み
2-3	PC/104 バス IRQ4 割込

JP2：P70 入力切り替え

新機種の MPCH-8CPU-V1 のリアルタイムクロックには温度センサが内蔵されておりません。したがって、JP2 は削除されています。

~~ショート位置~~	~~P70 の接続先~~
~~1-2~~	~~温度センサ入力（P70 を AD 変換器として動作させる）~~
~~2-3~~	~~コネクタ J6、ピン 1 と接続入力~~

JP3：リアルタイムクロック電池

ジャンパ設定	バックアップ電池スイッチ
オープン	バックアップ無効
ショート	内蔵リチウム電池によるバックアップ有効

JP4：LCD コネクタ 2 番ピン

ショート位置	J7 コネクタ 2 番ピン接続先
1	VCC に接続します
3	GND に接続します

JP5：LCD コネクタ 1 番ピン

ショート位置	J7 コネクタ 1 番ピン接続先
1	VCC に接続します
3	GND に接続します

6.2. モード設定ディップスイッチ

SW1-1

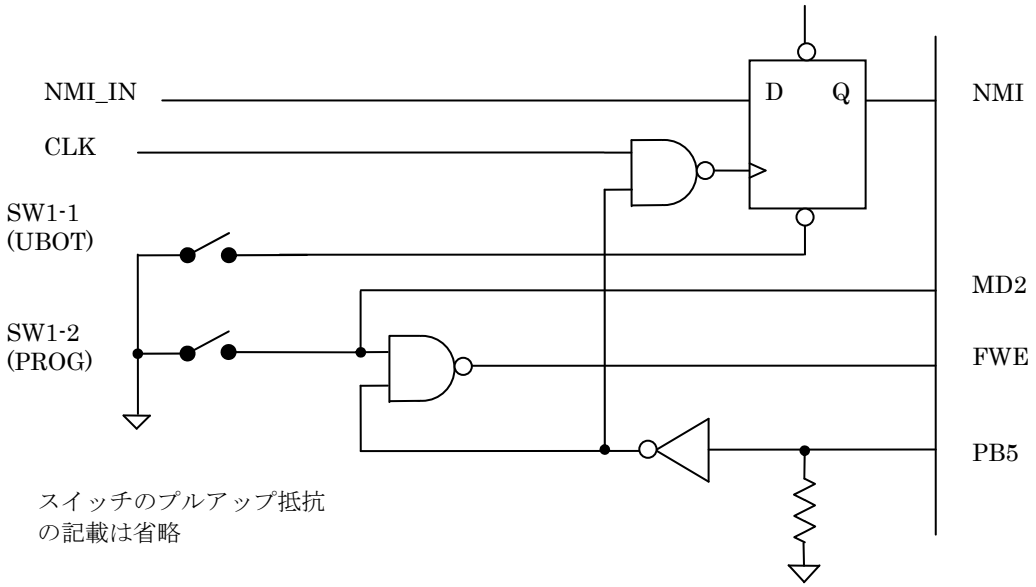
ユーザーブートモードに切り替えます。

SW1-2

ブートモードに切り替えます。

スイッチ位置とモードの関係は CPU 動作モードを参照してください。

6.3. モード設定回路



7. CPU 動作モード

・ ユーザー（通常）モード
内蔵 ROM（ユーザーマット）からプログラムを起動

・ ユーザープログラムモード
ユーザープログラムから内蔵 ROM（ユーザーマット）を書き換える

・ ブートモード
ホストから SCI 経由でユーザーマットまたはユーザーブートマットを書き換える

・ ユーザーブートモード
ユーザーブートマットからプログラムを起動する．緊急時などにユーザーマットを書き換える際使用するモード

スイッチ設定		外部信号			CPU 端子状態			モード
SW1-2 (PRG)	SW1-1 (UBOT)	PB5	NMI_IN		NMI	MD2	FWE	
OFF	OFF	0	0/1		0/1	1	0	ユーザー（通常）モード
OFF	OFF	1	0/1		固定	1	1	ユーザープログラムモード
ON	OFF	0	－		1	0	1	ブートモード
ON	ON	0	－		0	0	1	ユーザーブートモード

ポイント！
・ シリアルポート(J11)から内蔵フラッシュメモリを書き換えるときは SW1-2 を ON に
・ 通常の使用状態では SW1 はどちらも OFF に
・ ユーザープログラム時は PB5 を 1 にすると CPU の FWE 端子が High になります

8. コネクタピン番号解説

8.1. J2：アラーム出力

ピン番号	信号
1	ALARM 出力
2	GND

8.2. J3：PC/104 コネクタ

ピン番号	信号
A1	N.C.
A2	SD7
A3	SD6
A4	SD5
A5	SD4
A6	SD3
A7	SD2
A8	SD1
A9	SD0
A10	N.C.
A11	AEN
A12	プルダウン（4.7k）
A13	プルダウン（4.7k）
A14	プルダウン（4.7k）
A15	プルダウン（4.7k）
A16	SA15
A17	SA14
A18	SA13
A19	SA12
A20	SA11
A21	SA10
A22	SA9
A23	SA8
A24	SA7
A25	SA6
A26	SA5
A27	SA4
A28	SA3
A29	SA2
A30	SA1
A31	SA0
A32	GND

ピン番号	信号
B1	GND
B2	RESET
B3	+5V
B4	N.C.
B5	N.C.
B6	N.C.
B7	N.C.
B8	N.C.
B9	+12V
B10	GND
B11	プルアップ（10k）
B12	プルアップ（10k）
B13	IOW#
B14	IOR#
B15	プルアップ（10k）
B16	N.C.
B17	プルアップ（10k）
B18	N.C.
B19	プルアップ（10k）
B20	N.C.
B21	IRQ7
B22	N.C.
B23	IRQ5
B24	IRQ4
B25	IRQ3
B26	N.C.
B27	N.C.
B28	N.C.
B29	+5V
B30	N.C.
B31	GND
B32	GND

8.3. J4：電源コネクタ

ピン番号	信号
1	+5V
2	GND
3	GND
4	+12V（PC/104 バスにのみ接続）

8.4. J5：CF コネクタ

ピン番号	信号
1	GND
2	D3
3	D4
4	D5
5	D6
6	D7
7	CE1#
8	A10（GND）
9	ATASEL#（GND）
10	A9（GND）
11	A8（GND）
12	A7（GND）
13	+5V
14	A6（GND）
15	A5（GND）
16	A4（GND）
17	A3
18	A2
19	A1
20	A0
21	D0
22	D1
23	D2
24	IOCS16# (N.C.)
25	CD2# (N.C.)

ピン番号	信号
26	CD1#
27	D11
28	D12
29	D13
30	D14
31	D15
32	CE2#
33	VS1 (N.C.)
34	IORD#
35	IOWR#
36	WE （プルアップ;10k）
37	INTRQ
38	+5V
39	CSEL（GND）
40	VS2 (N.C.)
41	RESET#
42	IORDY (N.C.)
43	INPACK (N.C.)
44	REG# （プルアップ;10k）
45	DASP （プルアップ;10k）
46	PDIAG （プルアップ;10k）
47	D8
48	D9
49	D10
50	GND

8.5. J6：ディジタル I/O／アナログ I/O

ピン番号	信号
1	P70~~（JP2 経由）~~
3	P71
5	P72
7	P73
9	P74
11	P75
13	P76
15	P77
17	NMI
19	GND

ピン番号	信号
2	PA0
4	PA1
6	PA2
8	PA3
10	PA4
12	PA5
14	PA6
16	AS#(使用しないでください)
18	P67
20	GND

入力保護回路が挿入されています。

注意！

J7 コネクタと同時使用しないでください。

8.6. J7：ディジタル I/O／LCDI/F

ピン番号	信号
1	JP5-2
3	LCD バイアス

ピン番号	信号
B1	JP4-2
B2	PA5

5	PA6
7	プルダウン(4.7k)
9	プルダウン(4.7k)
11	PA0
13	PA2

B3	PA4
B4	プルダウン(4.7k)
B5	プルダウン(4.7k)
B6	PA1
B7	PA3

JP4、JP5 は VCC または GND に接続可能です。

LCD バイアス調整ポテンショ（10kΩ）は VCC と GND 間に接続され、3 番ピンはスライダと接続されています。LCD のバイアスとして使用します。

注意！

JP4、JP5 のそれぞれの 1、3 番を同時にショートしないでください（基板シルクは 1、3 と印刷されています)。

J6 コネクタと同時使用しないでください。

8.7. J8：USB コネクタ

ピン番号	信号
1	N.C.
2	D−
3	D＋
4	GND

8.8. J9：10BASE-T コネクタ

ピン番号	信号
1	TX＋
2	TX−
3	RX＋
4	N.C.
5	N.C.
6	RX−
7	N.C.
8	N.C.

8.9. J10：SCI（シリアル）ポート 0

ピン番号	信号
1	N.C.
2	RD
3	TD
4	6、8 と接続
5	GND
6	4、8 と接続
7	N.C.
8	4、6 と接続
9	N.C.
10	N.C.

8.10. J11：SCI（シリアル）ポート 1

ピン番号	信号
1	N.C.
2	RD
3	TD
4	6、8 と接続
5	GND
6	4、8 と接続
7	N.C.
8	4、6 と接続

9	N.C.
10	N.C.

8.11. J12：SCI（シリアル）ポート 2

ピン番号	信号
1	N.C.
2	RD
3	TD
4	6、8 と接続
5	GND
6	4、8 と接続
7	N.C.
8	4、6 と接続
9	N.C.
10	N.C.

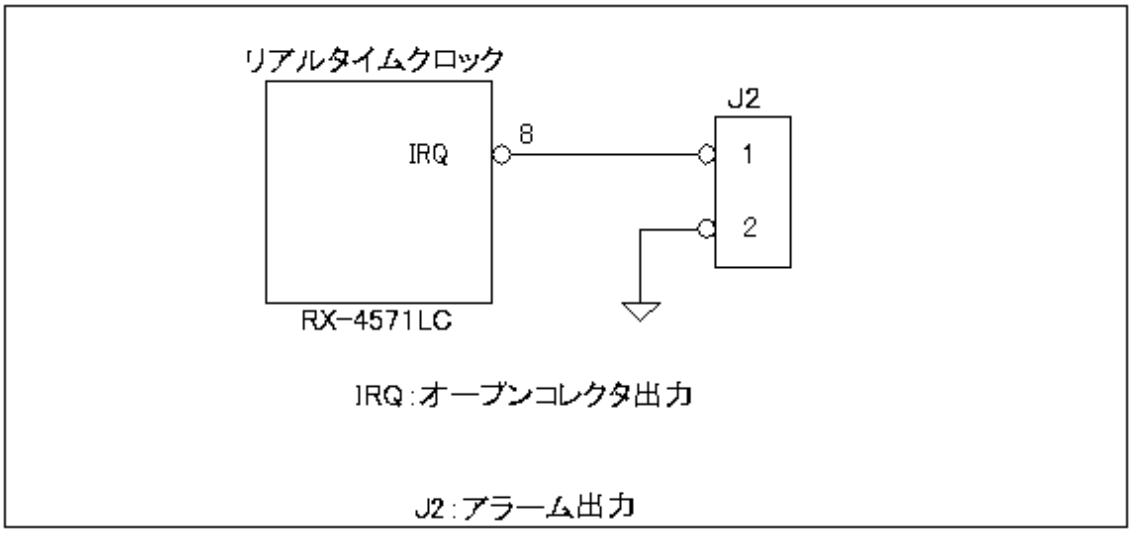
9. 適合コネクタ

コネクタ番号	メーカー	型番
J2	JST	XHP-2
J3		PC/104 コネクタ
J4	Tyco AMP	171822-4
J5		
J6	OMRON	XG4M-2030-T、XG5M-2032-N など
J7	OMRON	XG4M-^1430-T、XG5M-1432-N など
J8	JST	XHP-4
J9		
J10～J12	OMRON	XG4M-1030-T、XG5M1032-N など

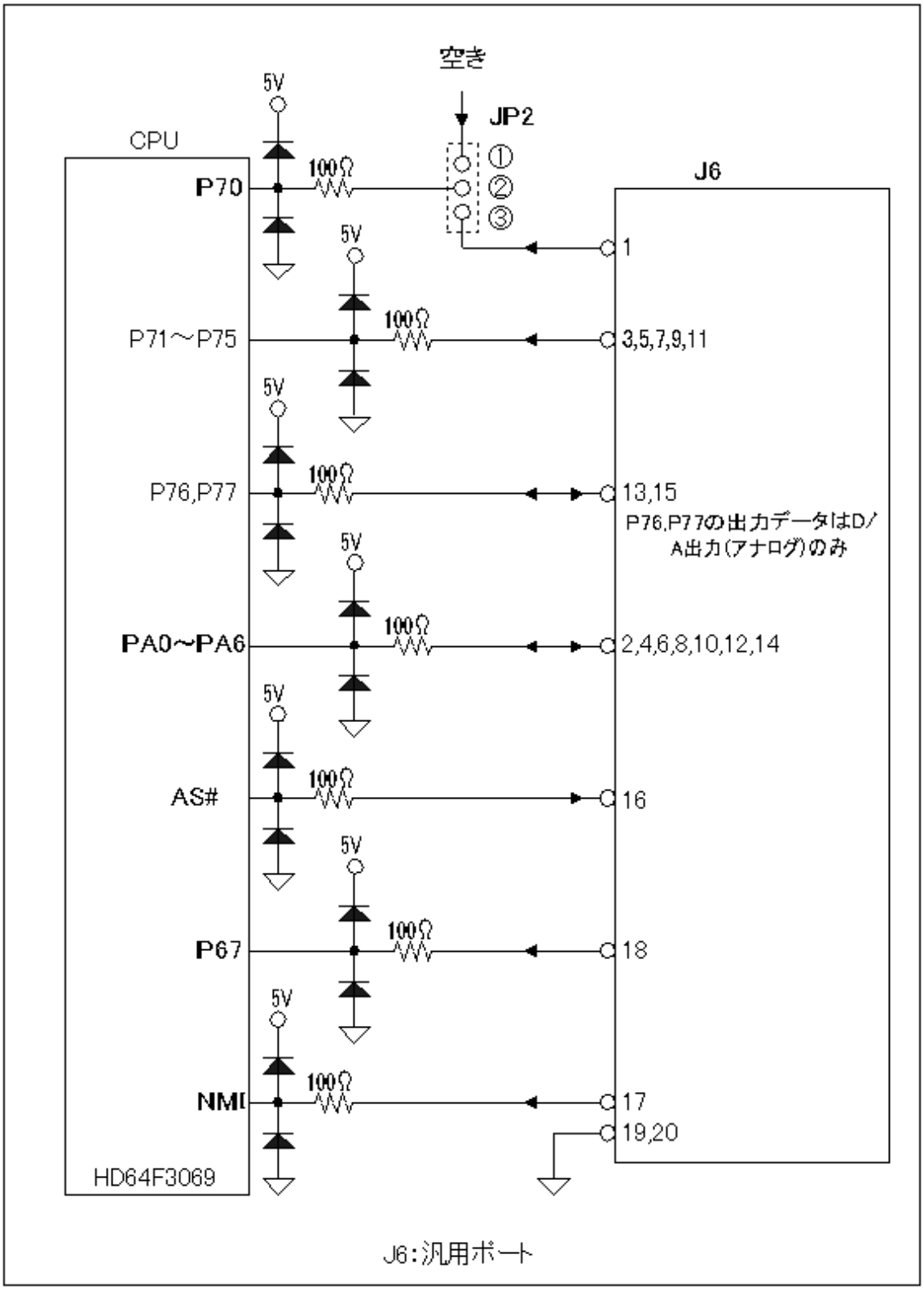
10. 入出力回路説明

10.1. J2：アラーム出力

J2 端子はリアルタイムクロック IC と直結されています。外部から定格を超える電圧、電流を加えると素子が破壊されます。また、接続時には静電気や、グラウンド電位の違いにじゅうぶん注意してください。



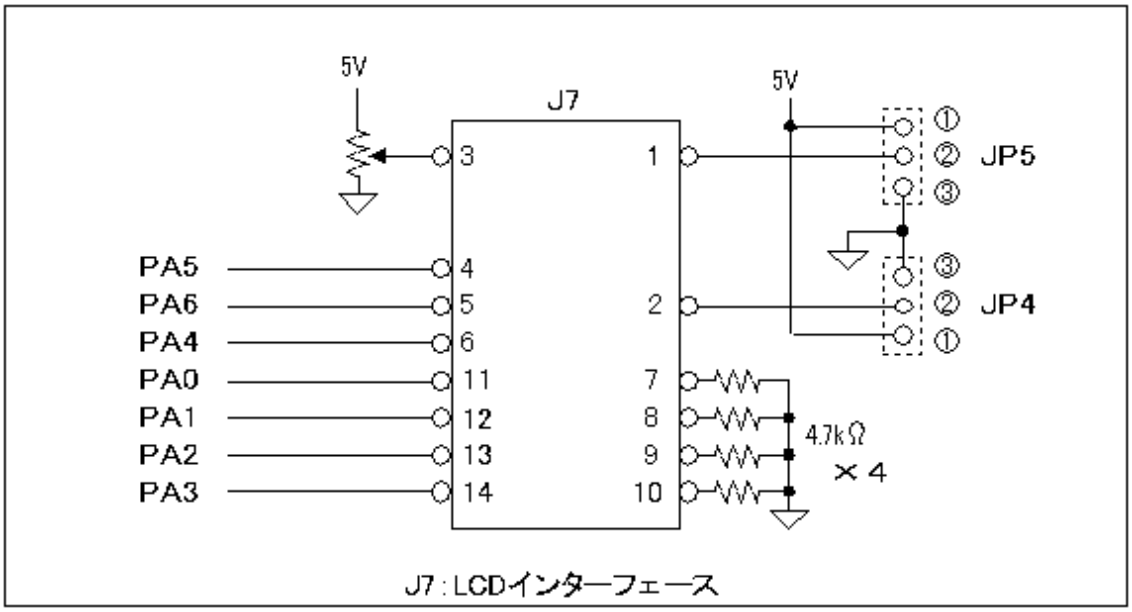
10.2. J6：汎用ポート



J6:汎用ポート

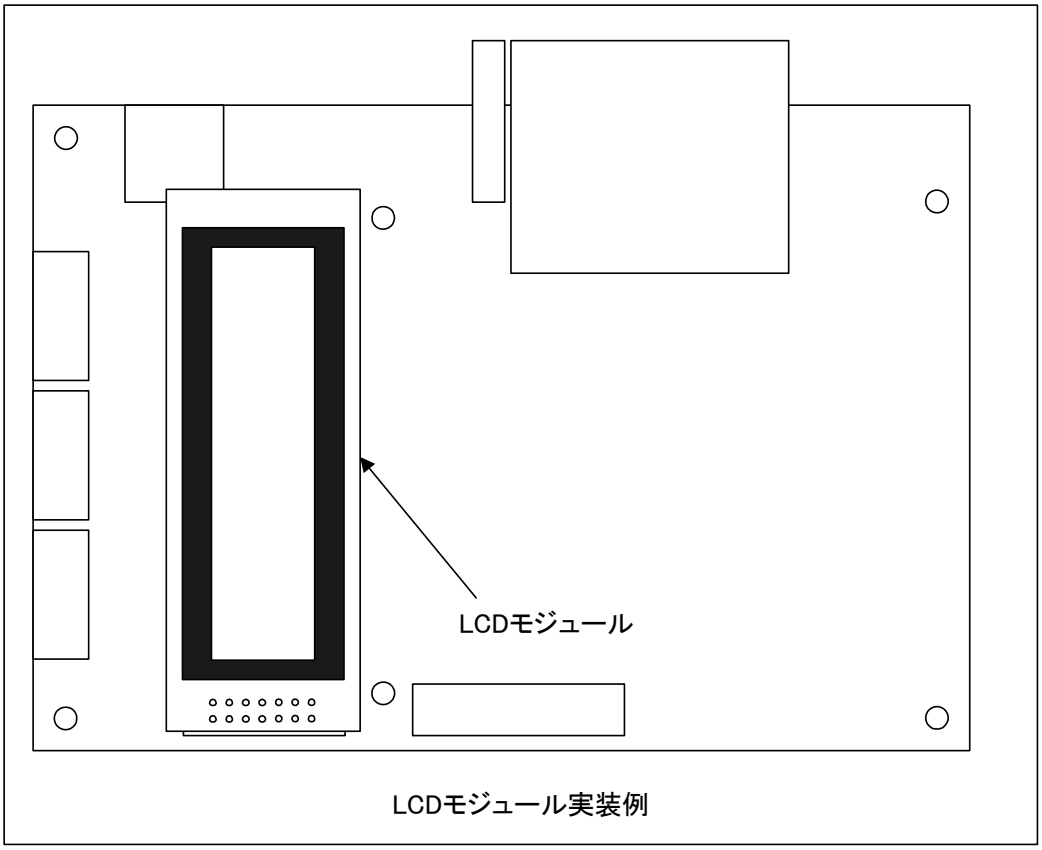
10.3. J7：LCD インターフェース

JP4、JP5 のショート位置番号は１と３がシルク印刷されています。ランドにハンダを盛ってショートしてください。１と３を同時にショートすると、電源とグラウンドが短絡状態になり、危険です。ご注意ください。



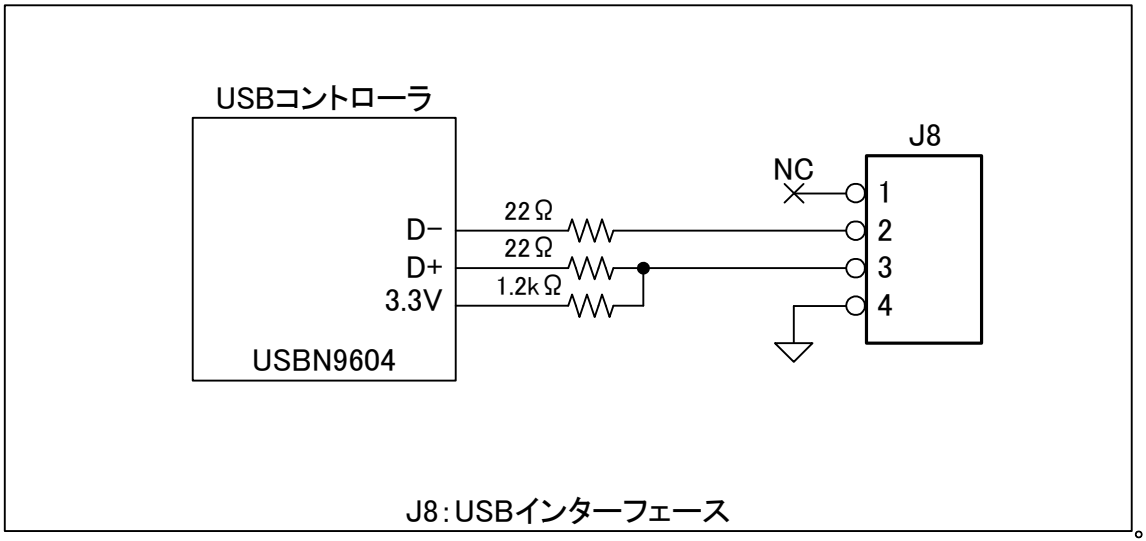
J7：LCDインターフェース

本カードは LCD モジュールを直接実装することができます。



LCDモジュール実装例

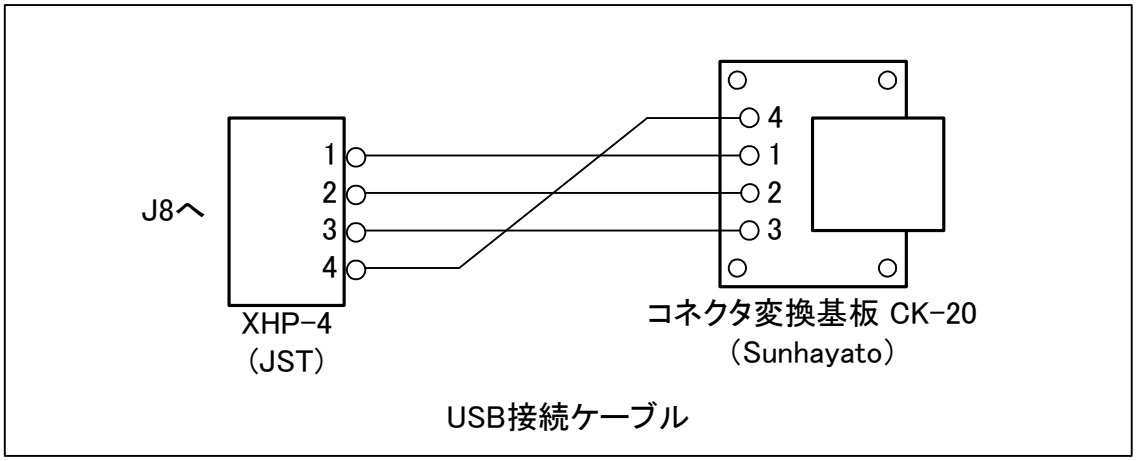
10.4. J8：USB インターフェース



J8：USBインターフェース

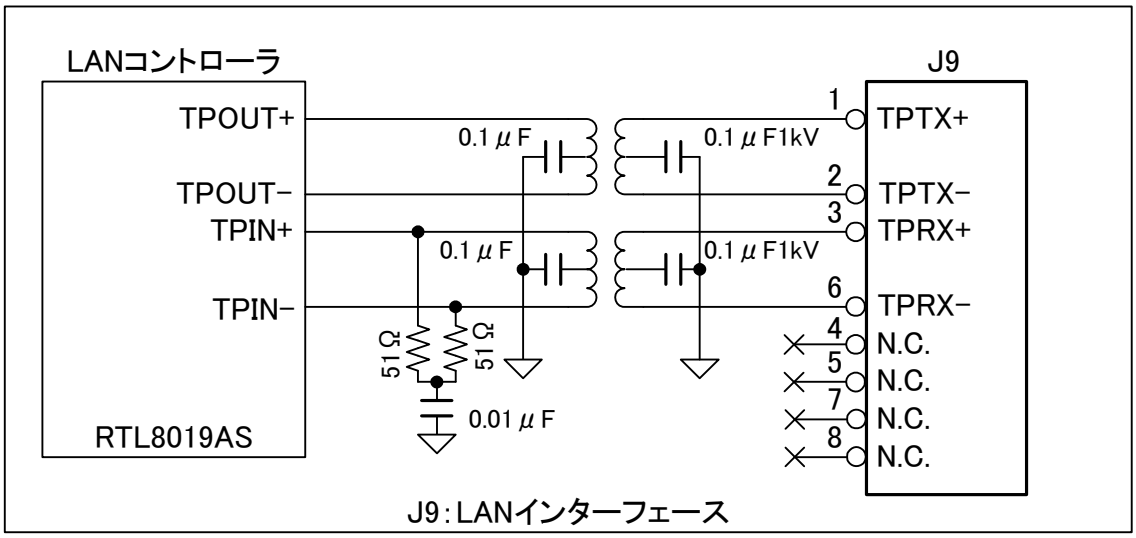
。

USB B タイプコネクタを使用する場合は変換コネクタを利用し、下記の接続ケーブルを作成してください。



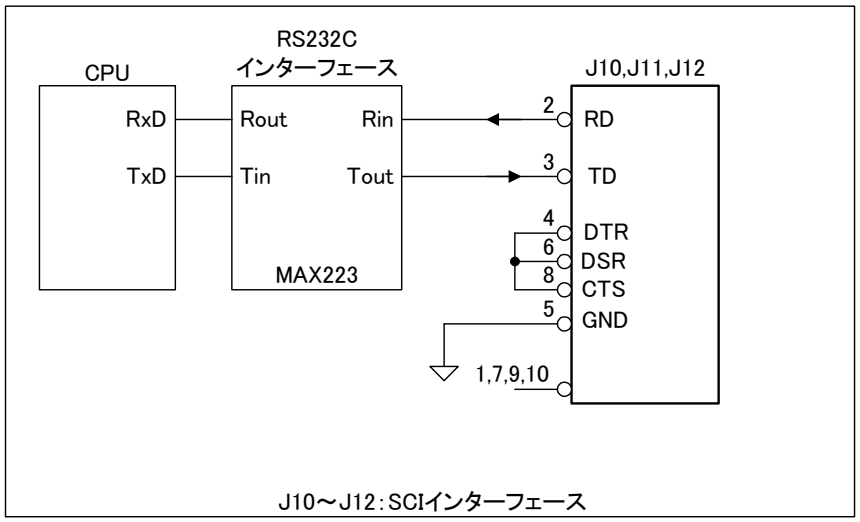
USB接続ケーブル

10.5. J9：LAN インターフェース



J9：LANインターフェース

10.6. J10～J12：SCI インターフェース

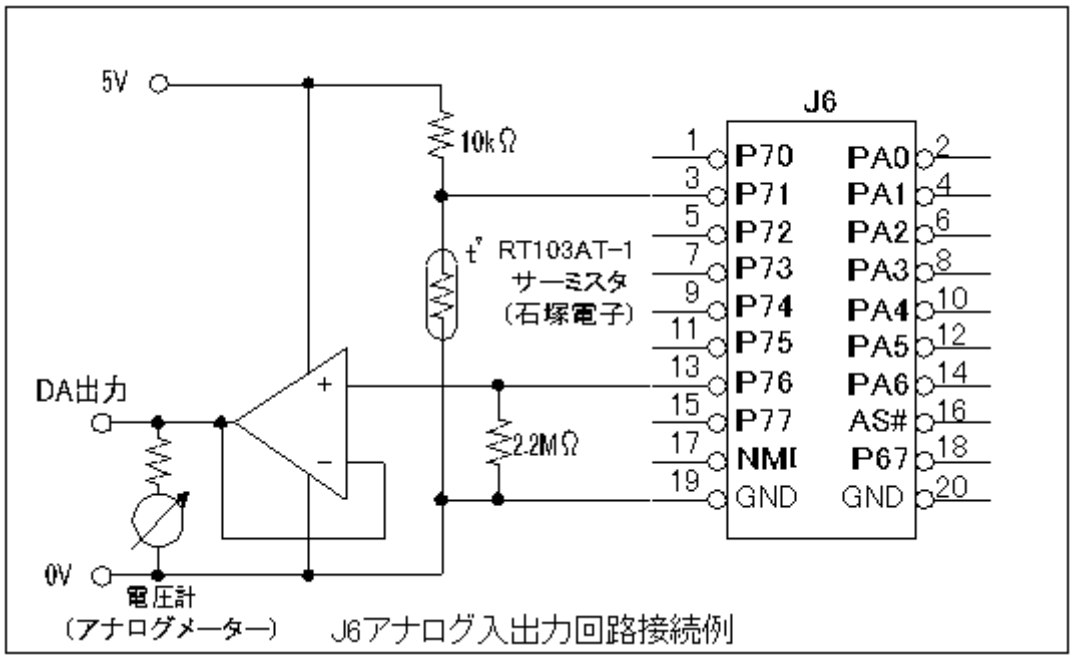


J10～J12：SCIインターフェース

11. I/O 接続例

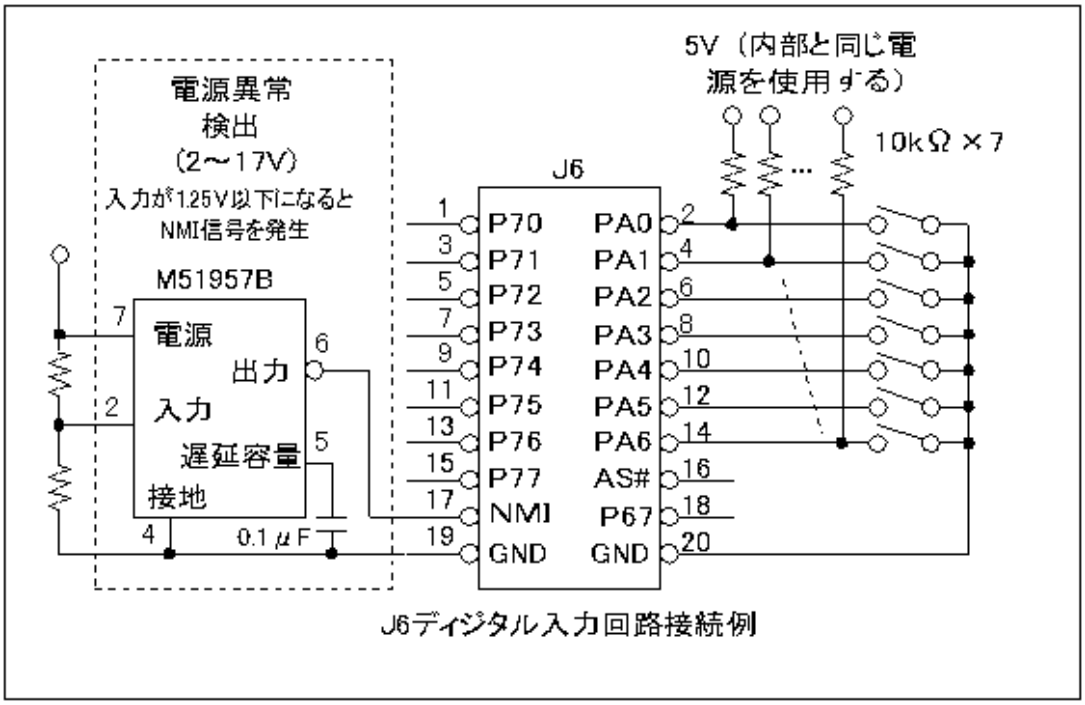
11.1. J6 アナログ入出力

下図は内蔵 AD 変換器、DA 変換器を利用する例です。AD 変換器で温度を計測し、演算処理した値を DA 変換器し、アナログ式の電圧計で表示します。



11.2. J6 ディジタル入力

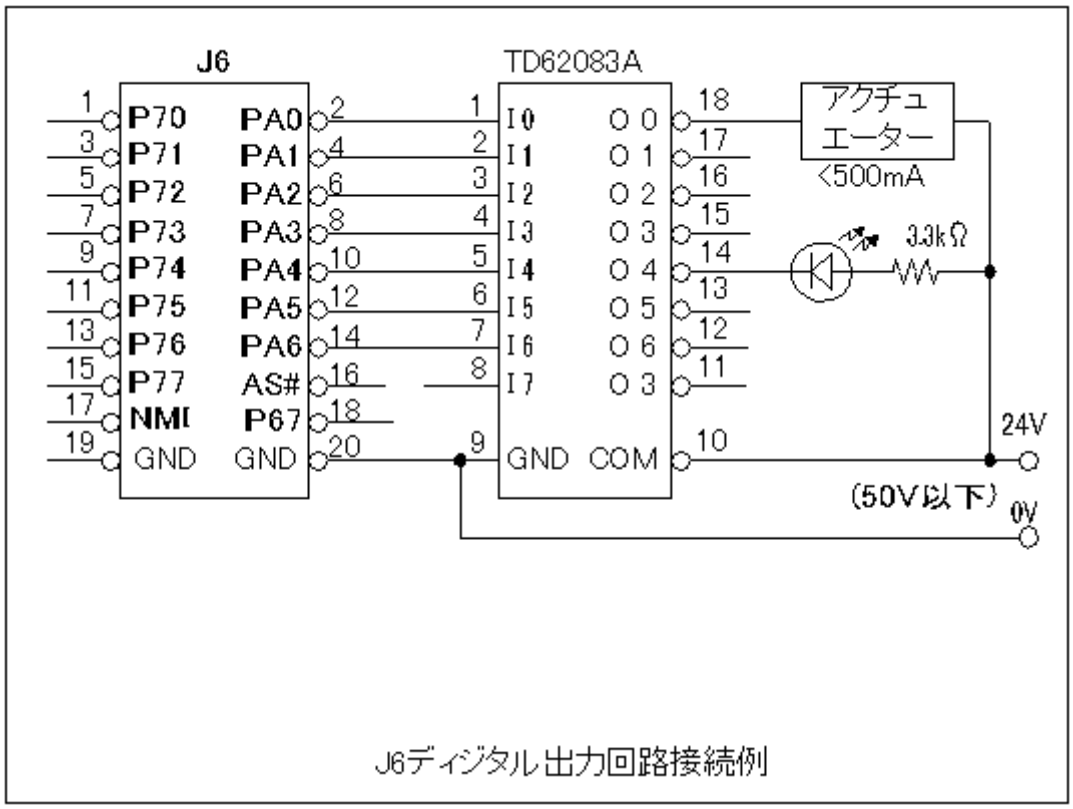
イッチからの入力は PA ポートから読み取ります。NMI 入力は電圧検出素子に接続されており、電源異常（電圧低下、瞬断）時、CPU に割り込みを発生します。



11.3. J6 ディジタル出力

CPU 内部ポートの出力許容電流は 2mA と小さいため LED の駆動程度にとどめてくださ

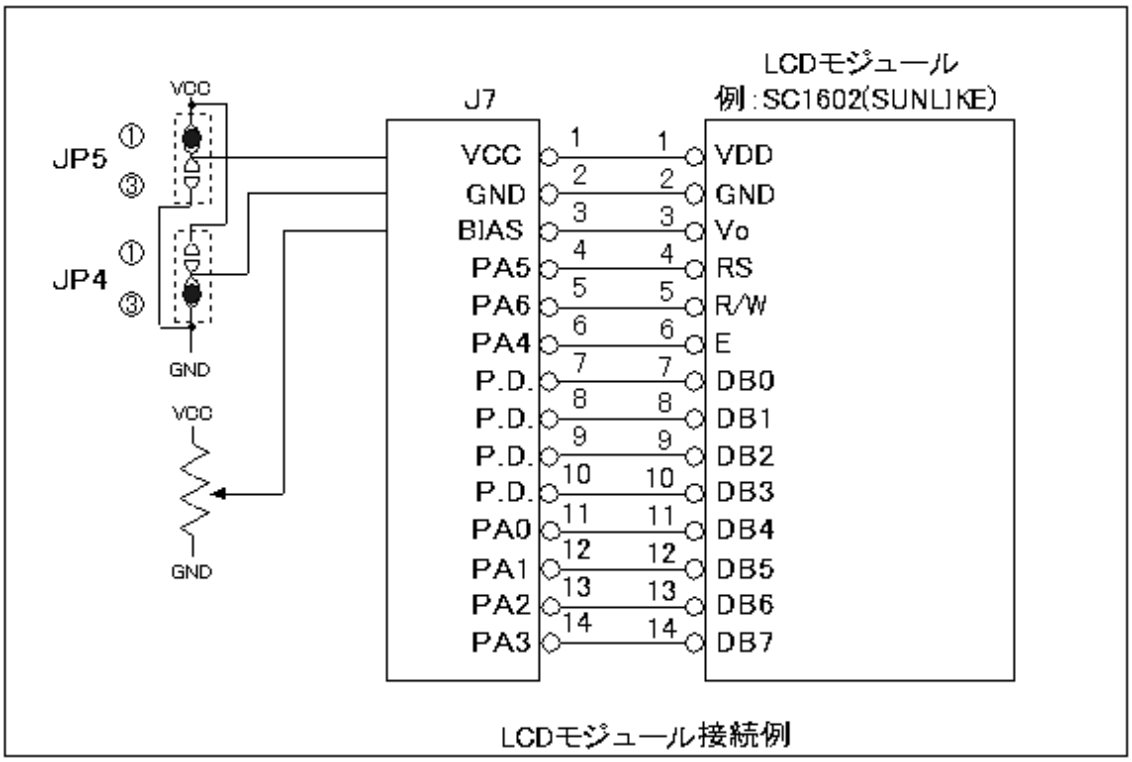
い。大電流を制御する場合はトランジスタアレイや MOS-FET などを介して負荷を接続してください。



J6ディジタル出力回路接続例

11.4. J7 LCD 接続

下図に 16 列×2 行表示の LCD モジュールを使用した回路例を示します。本カードの J7 コネクタは SUNLIKE 社 SC1602 と直結できるように設計されています。JP4 と JP5 で電源、GND のジャンパ接続をしてください。

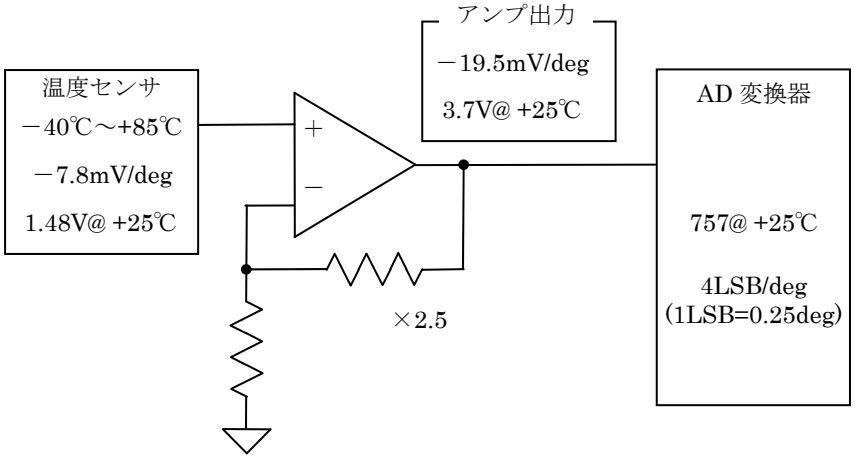


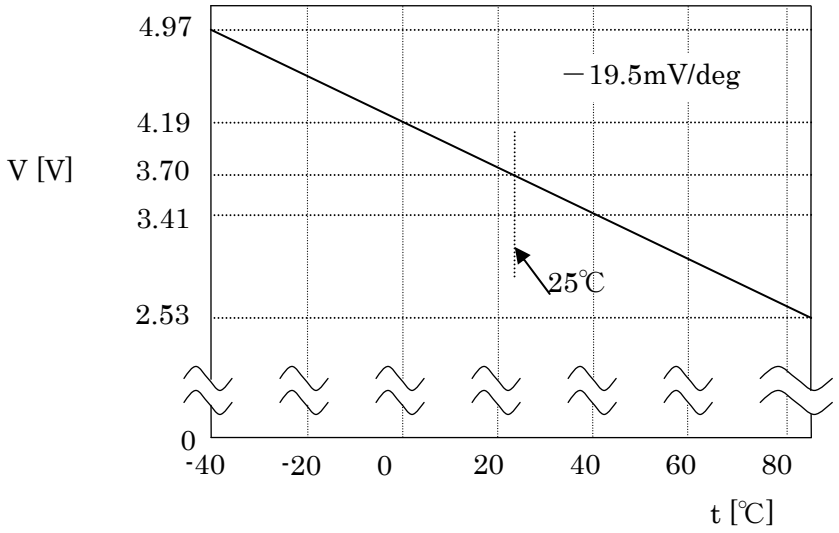
LCDモジュール接続例

12. 温度センサ変換特性

新機種の MPCH-8CPU-V1 のリアルタイムクロックには温度センサが内蔵されておりません。したがって、下記解説は旧機種のみ有効とします。

旧機種のリアルタイムクロックには温度センサが内蔵されており、リアルタイムクロックチップ周辺の温度を計測します。





変換式　：　V = -0.01952t + 4.19

t = (4.19 - V) / 0.01952

V = 5.0 × N / 1024 より(N は AD 変換値：0～1023)

t = 214.65 – 0.2501 × N

AD 変換値はレジスタ読み取り値を 6 ビットシフトした後の値（0～1023）です。

13. プログラム例

13.1. システムレジスタ等初期化

CPU レジスタの初期化の中で、バス、I/O アクセスに関係する部分をアセンブリ言語の例で示します。メモリ配置、バス幅、必要ステート、ポートの入出力方向を適切な状態に設定しております。これらの設定の後、スタートアップルーチンに移行し、変数、I/O の初期化を行ってください。

```
;
;
;---------------------------------------------------------------
;       スタック定義
;---------------------------------------------------------------
STACK_TOP       EQU     H'FFFF20        ;内蔵 RAMTOP の次
;---------------------------------------------------------------
;       システムコントローラアドレス定義
;---------------------------------------------------------------
SYSCR           EQU     H'FEE012
;---------------------------------------------------------------
;       ポートアドレス定義
;---------------------------------------------------------------
P1DDR           EQU     H'FEE000
P2DDR           EQU     H'FEE001
P5DDR           EQU     H'FEE004
P8DDR           EQU     H'FEE007
P9DDR           EQU     H'FEE008
PADDR           EQU     H'FEE009
PBDDR           EQU     H'FEE00A
P6DR            EQU     H'FFFFD5
PBDR            EQU     H'FFFFDA
CSCR            EQU     H'FEE01F
ASTCR           EQU     H'FEE021
;---------------------------------------------------------------
;       バスコントローラ定義
;---------------------------------------------------------------
BRCR            EQU     H'FEE013
ABWCR           EQU     H'FEE020
ASTCR           EQU     H'FEE021
WCRH            EQU     H'FEE022
WCRL            EQU     H'FEE023
;***************************************************************
;       プログラムエントリ
;***************************************************************
START:                                                  ;
;---------------------------------------------------------------
;       スタックポインタ初期化
;---------------------------------------------------------------
        MOV.L   #STACK_TOP,ER7
;---------------------------------------------------------------
```

```
;       割込み制御モード設定
;---------------------------------------------------------------------
        MOV.B   @SYSCR,R0L
        AND.B   #B'11110111,R0L
        MOV.B   R0L,@SYSCR
;---------------------------------------------------------------------
;       バス、外部メモリ、チップセレクトの設定
;---------------------------------------------------------------------
        MOV.B   #H'FF,R0L
        MOV.B   R0L,@P1DDR
        MOV.B   R0L,@P2DDR
        MOV.B   R0L,@P5DDR
;
        MOV.B   #H'40,R0L               ;
        MOV.B   R0L,@ABWCR              ;
;---------------------------------------------------------------------
;       A21-A23、BREQ、BACK はポートで使用、A20 はアドレス出力
;---------------------------------------------------------------------
        MOV.B   #H'EE,R0L               ;
;
        MOV.B   R0L,@BRCR               ;
;---------------------------------------------------------------------
;       CS4-CS6 を有効にする
;
;       CS4(80000H-9FFFFH) : 外部 SRAM(16bit バス、2M バイト)
;       CS5(A0000H-BFFFFH) : コンパクトフラッシュ(16bit バス)
;       CS6(C0000H-DFFFFH) : PC104、USB、LAN、LED、DIPSW(8bit バス、各 64K バイト)
;---------------------------------------------------------------------
        MOV.B   #H'7F,R0L
        MOV.B   R0L,@CSCR               ;
;---------------------------------------------------------------------
;       CS4 はウエイトステート 0 に設定
;---------------------------------------------------------------------
        MOV.B   #B'11111100,R0L         ;
        MOV.B   R0L,@WCRH               ;
;---------------------------------------------------------------------
;        PB0,PB4 を出力ポートに設定
;---------------------------------------------------------------------
        MOV.B   #H'11,R0L               ;
        MOV.B   R0L,@PBDDR
        XOR.B   R0L,R0L                 ;R0L=0
        MOV.B   R0L,@P6DR               ;PB0,PB4 を L にする
;---------------------------------------------------------------------
;       P80～P83 は IRQ0～IRQ3、P84 は入力ポートに設定
;---------------------------------------------------------------------
        MOV.B   #H'00,R0L               ;
        MOV.B   R0L,@P8DDR
;---------------------------------------------------------------------
;       PA0-PA6 を出力ポートに設定
;---------------------------------------------------------------------
```

```
        MOV.B   #H'7F,ROL                   ;
        MOV.B   ROL,@PADDR
;-------------------------------------------------------------------
;       PB4(EEPROM の SCK 信号)を H にする
;-------------------------------------------------------------------
        BSET.B  #4,@PBDR            ;
;-------------------------------------------------------------------
;       P61,P62 を出力ポートに設定
;-------------------------------------------------------------------
        MOV.B   #H'06,ROL                   ;
        MOV.B   ROL,@P6DDR
;-------------------------------------------------------------------
;       リアルタイムクロックのアクセス準備
;-------------------------------------------------------------------
        SET.B   #1,@P6DR
        SET.B   #2,@P6DR
;-------------------------------------------------------------------
;       EEPROM のアクセス準備
;-------------------------------------------------------------------
        SET.B   #1,@PBDR
        SET.B   #2,@P6DR
;-------------------------------------------------------------------
;       モジュールスタンバイレジスタの設定（デフォルト）
;-------------------------------------------------------------------
        MOV.B   #H'F8,ROL                   ;
        MOV.B   ROL,@MSTCRH
;-------------------------------------------------------------------
;       PB0(EEPROM の SDA 信号)を H にする
;-------------------------------------------------------------------
        BSET.B  #4,@PBDR            ;

        ・
        ・
        ・
```

13.2. シリアルコントロールレジスタ設定

シリアルコントロールレジスタの初期化の例を示します。例では割込許可ビットをイネーブルに設定していませんので割り込みを使用する場合、各種初期化が終了した後（割込受付可能になってから）割込許可ビットを設定してください。

13.2.1. アセンブリ言語で記述した場合

```
;=============
; SCIアドレス
;=============
;SCI0
SMR0    EQU     H'FFFFB0
BRR0    EQU     H'FFFFB1
SCR0    EQU     H'FFFFB2
TDR0    EQU     H'FFFFB3
SSR0    EQU     H'FFFFB4
RDR0    EQU     H'FFFFB5
;SCI1
SMR1    EQU     H'FFFFB8
BRR1    EQU     H'FFFFB9
SCR1    EQU     H'FFFFBA
TDR1    EQU     H'FFFFBB
SSR1    EQU     H'FFFFBC
RDR1    EQU     H'FFFFBD
; SCI2
SMR2    EQU     H'FFFFC0
BRR2    EQU     H'FFFFC1
SCR2    EQU     H'FFFFC2
TDR2    EQU     H'FFFFC3
SSR2    EQU     H'FFFFC4
RDR2    EQU     H'FFFFC5
;       SCI0の場合
;       DEFINE  SCI_SMR SMR0
;       DEFINE  SCI_BRR BRR0
;       DEFINE  SCI_SCR SCR0
;       DEFINE  SCI_TDR TDR0
;       DEFINE  SCI_SSR SSR0
;       DEFINE  SCI_RDR RDR0
;       DEFINE  INT_PRI 8
;
;       SCI1の場合
        DEFINE  SCI_SMR SMR1
        DEFINE  SCI_BRR BRR1
        DEFINE  SCI_SCR SCR1
        DEFINE  SCI_TDR TDR1
        DEFINE  SCI_SSR SSR1
        DEFINE  SCI_RDR RDR1
        DEFINE  INT_PRI 4
;
```

```
;       SCI2の場合
;       DEFINE   SCI_SMR  SMR2
;       DEFINE   SCI_BRR  BRR2
;       DEFINE   SCI_SCR  SCR2
;       DEFINE   SCI_TDR  TDR2
;       DEFINE   SCI_SSR  SSR2
;       DEFINE   SCI_RDR  RDR2
;       DEFINE   INT_PRI  2
;
DEFINE  BRR_CNT  19       ;38400bps@24MHz
;
        MOV.B   #BRR_CNT,R0L
        MOV.B   R0L,@SCI_BRR
        MOV.B   #B'00000000,R0L
        MOV.B   R0L,@SCI_SMR
        MOV.B   #B'00110000,R0L   ; 受信データフル割り込み禁止
        MOV.B   R0L,@SCI_SCR
        MOV.B    @IPRB,R0L
        OR.B     #INT_PRI,R0L     ;割込みのレベルをあげる
        MOV.B    R0L,@IPRB
        .
        .
```

13.2.2. C 言語で記述した場合

```
#define SMR0  (*(volatile unsigned char   *)0xFFFFB0)
#define BRR0  (*(volatile unsigned char   *)0xFFFFB1)
#define SCR0  (*(volatile unsigned char   *)0xFFFFB2)

void init_sci(void)
{
  SMR0              = 0;          // Parity None
  BRR0              = 19;         // 38400bps @24MHz
  SCR0              = 0x30;       //
}
```

13.3. タイマコントロールレジスタ設定

この例ではタイマ(チャネル０)をインターバルタイマとして使用します。タイマは 10ms ごとに CPU に割込を発生させます。割込処理内では割込フラグのリセットと、タイマ変数（外部）をインクリメントを行います。

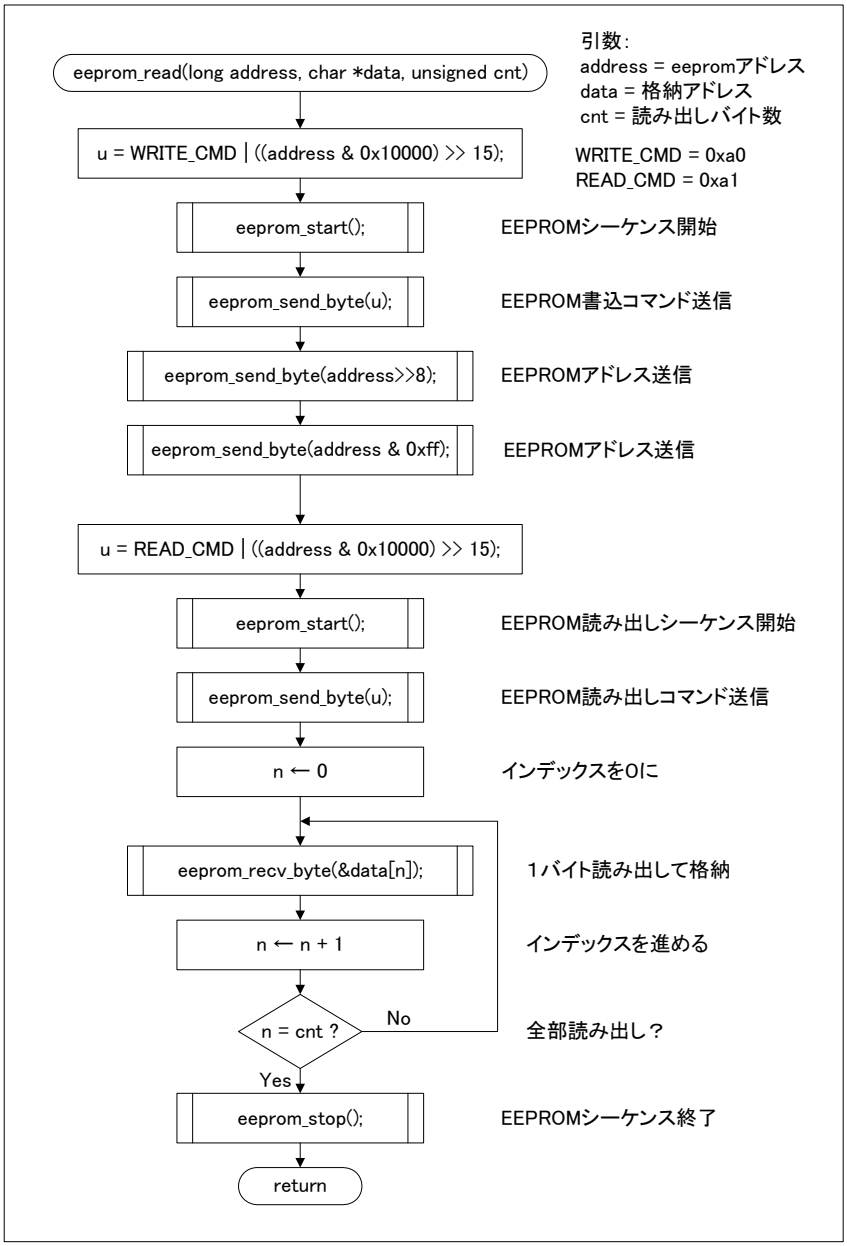
```
#define TSTR            (*((unsigned char *)(0xffff60)))
#define TCR0            (*((unsigned char *)(0xffff68)))
#define TISRA           (*((unsigned char *)(0xffff64)))
#define TGR0A           (*((unsigned short *)(0xffff6c)))

extern long timercount; // 外部で定義されたタイマ変数
//************************************
// タイマ初期化
// 10msインターバル割込として使用する
//************************************
void
init_timer(VOID)
{
  TCR0 = 0x23;          // 動作モード
                        //  GRAのコンペアマッチでTCNTをクリア、
                        //  プリスケーラ=8 (24.000MHz/8 = 3.000MHz)
                        //  IPRF &= 0x0f;
                        //  IPRF |= 0x60;   (プライオリティ= 6)
  TISRA = 0x10;         //  IMFA0による割り込み許可
  TGR0A = (unsigned short)30000; // GRAレジスタ設定  3.000MHz/30000 = 100Hz(10ms)

  TSTR |= 0x01;         // タイマ０スタート
}

//************************************
// タイマ割込処理
// 10ms毎に呼び出される
//       割込ベクタは別に設定するか、
//       開発環境で登録する
//************************************
interrupt void timerIMIA0int(void)
{
  unsigned char int_stat;

  int_stat = TISRA;         // 割込フラグ読み込み
  timercount++;

  TISRA = int_stat & 0xfe;  // 割込フラグリセット
}
```

13.4. シリアル EEPROM アクセス

シリアル EEPROM アクセスの例をフローチャートで示します。

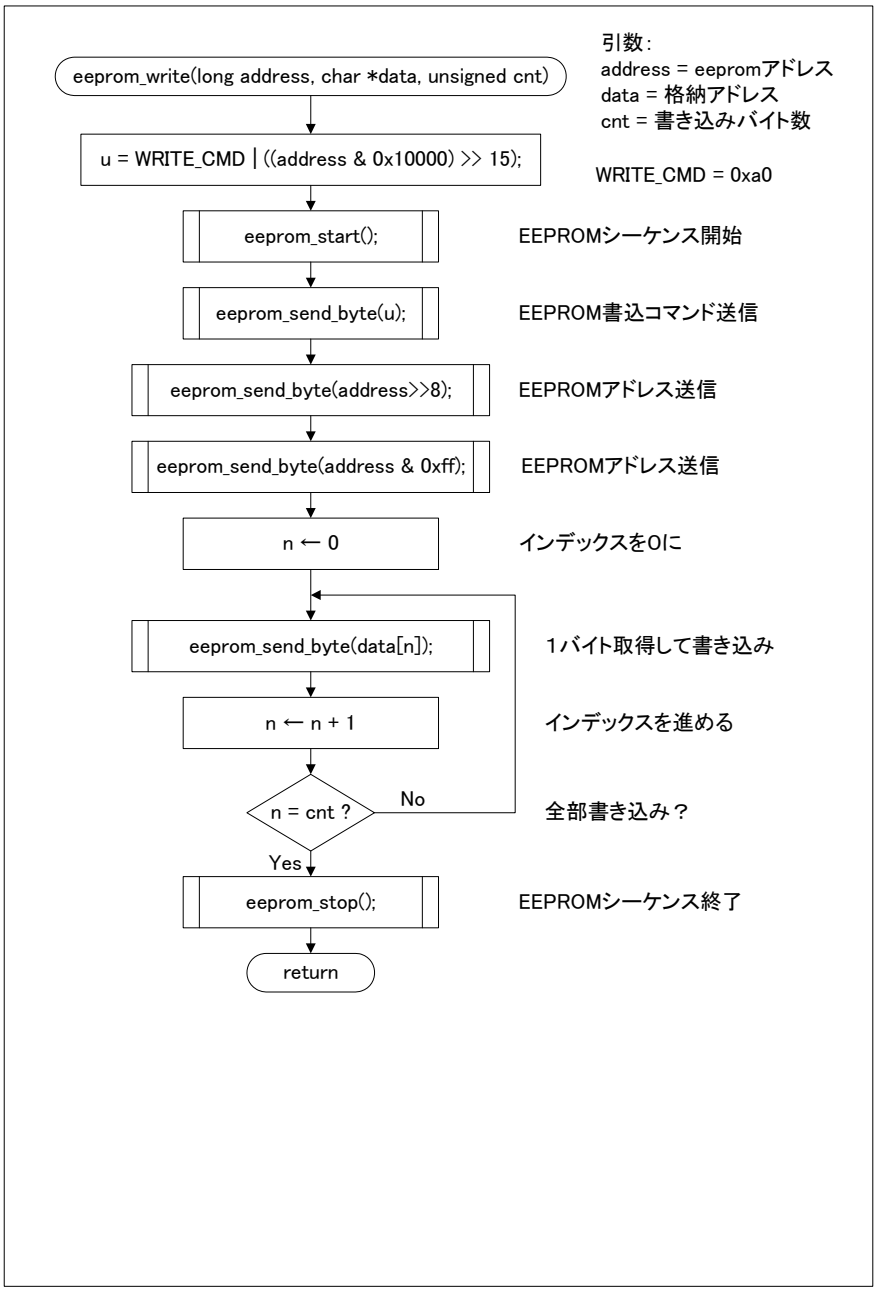
13.4.1. シリアル EEPROM 読み出しサンプル

シリアル EEPROM から任意バイト（256 バイト以下）を読み出します。ただし、ページ境界をまたがないこと。



13.4.2. シリアル EEPROM 書き込みサンプル

シリアル EEPROM へ任意バイト（256 バイト以下）を書き込みます。ただし、ページ境界をまたがないこと。



13.5. リアルタイムクロックアクセス

```
#define CLOCK_HIGH   P_P6DR |= 0x02;       // P61(CLK)  = H
#define CLOCK_LOW    P_P6DR &= 0fd;        // P61(CLK)  = L
#define DATA_HIGH    P_P6DR |= 0x01;       // P60(DATA) = H
#define DATA_LOW     P_P6DR &= 0xfe;       // P60(DATA) = L
#define CHANGE_OUT   P_P6DDR = (p6ddr_temp |= 0x01);       // P60 = OUT
#define CHANGE_IN    P_P6DDR = (p6ddr_temp &= 0xfe);       // P60 = OUT
#define CE_HIGH      P_P6DR |= 0x04;       // P62(DATA) = H
#define CE_LOW       P_P6DR &= 0xfb;       // P62(DATA) = L
#define DEV          RX4571
#if DEV=RX4571
  #define WRITE_CMD    0x10
  #define READ_CMD     0x90
#else
  #define WRITE_CMD    0x3
  #define READ_CMD     0xc
#endif
/*-------------------------------------------------------------*/
/*                  ~~RX-4702~~RX-4571レジスタ書き込み             */
/*      ~~RX-4702~~RX-4571レジスタへ１バイトのデータを書き込む    */
/*-------------------------------------------------------------*/
void
clk_data_out(unsigned char data)
{
    int  i;

    for(i=0;i<8;i++){
#if DEV=RX4571
        if (data & 0x80) DATA_HIGH else DATA_LOW   // RX-4571
        CLOCK_LOW
        data <<= 1;
        CLOCK_HIGH
#else
        if (data & 0x01) DATA_HIGH else DATA_LOW // RX-4702
        CLOCK_LOW
        data >>= 1;
        CLOCK_HIGH
#endif
   }
}

/*-------------------------------------------------------------*/
/*                  ~~RX-4702~~RX-4571レジスタ読み込み             */
/*      ~~RX-4702~~RX-4571レジスタから１バイトのデータを読み出す  */
/*-------------------------------------------------------------*/
void
clk_data_in(unsigned char *data)
{
```

```
    int i;

    *data = 0;
    for(i=0;i<8;i++){
#if DEV=RX4571
        *data <<= 1;   // RX-4571
        CLOCK_LOW
        if (P_P6DR & 0x01)
            *data |= 0x01;  // RX-4571
        CLOCK_HIGH
#else
        *data >>= 1;   // RX-4702
        CLOCK_LOW
        if (P_P6DR & 0x01)
            *data |= 0x80;  // RX-4702
        CLOCK_HIGH
#endif
    }
}

/*-------------------------------------------------------------------*/
/*                RX-4702RX-45712レジスタ列読み込み                  */
/*   regnoで指定されたRX-4702RX-4571レジスタからcntバイトのデータを   */
/*   読み出しdata配列へ格納する                                      */
/*-------------------------------------------------------------------*/
void
clk_read_register(unsigned char regno, unsigned char cnt, unsigned char *data)
{
    int i;

    CHANGE_OUT
    CE_HIGH
#if DEV=RX4571
    ;
#else
    regno <<= 4;  // RX-4702
#endif
    clk_data_out(READ_CMD + regno);

    CHANGE_IN
    for(i=0;i<cnt;i++)
        clk_data_in(data+i);

    CE_LOW
}

/*-----------------------------------------------------------*/
/*                RX-4702RX-4571レジスタ列書き込み           */
/*   data配列からcntバイトのデータをregnoで指定された        */
/*   RX-4702RX-4571レジスタへ書き込む                        */
```

```
/*-----------------------------------------------------------*/
void
clk_write_register(unsigned char regno, unsigned char cnt, unsigned char *data)
{
    int i;

    CHANGE_OUT
    CE_HIGH
#if DEV=RX4571
    ;
#else
    regno <<= 4;  // RX-4702
    clk_data_out(WRITE_CMD + regno);
#endif
    for(i=0;i<cnt;i++)
        clk_data_out(*(data+i));

    CE_LOW
}
```

13.6. USB コントローラレジスタアクセス

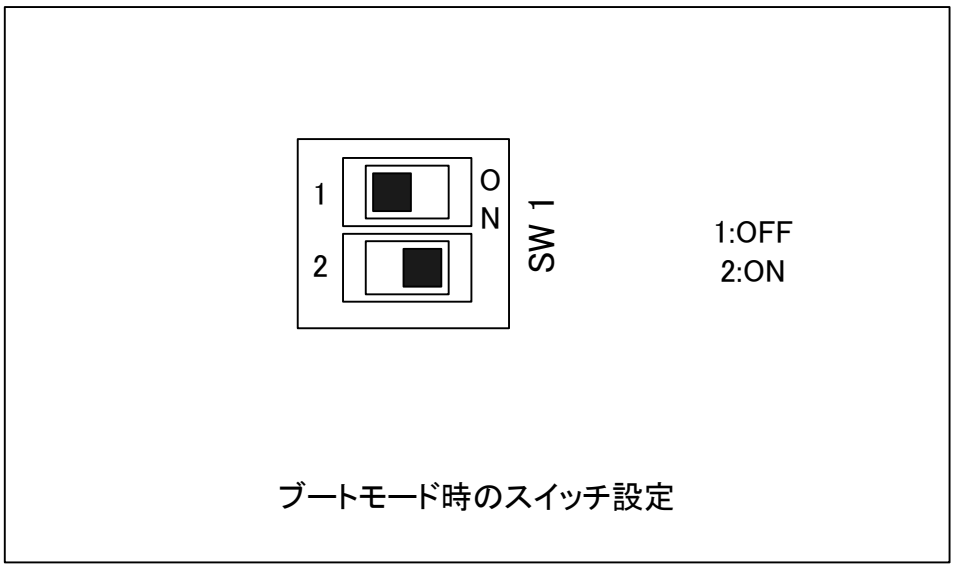
```
#define USB_DATA (*(unsigned char *)0xC10000)
#define USB_CTRL (*(unsigned char *)0xC10001)
#define USB_CLKDIV     0x04       // CLKOUT = 48MHz/4 = 12MHz
#define MCNTRL         0x00       // Main Control
#define CCONF          0x01       // Clock Configuration

/*-------------------------------------------------------------*/
/*                  USBN9604レジスタ読み込み                   */
/*-------------------------------------------------------------*/
unsigned char
ReadUSBreg(unsigned char reg_n)
{
    USB_CTRL = (unsigned char)reg_n;
    return( USB_DATA );
}

/*-------------------------------------------------------------*/
/*                  USBN9604レジスタ書き込み                   */
/*-------------------------------------------------------------*/
void
WriteUSBreg(unsigned char reg_n, unsigned char data)
{
    USB_CTRL = reg_n;
    USB_DATA = data;
}
/*-------------------------------------------------------------*/
/*                      USBN9604リセット                       */
/*-------------------------------------------------------------*/
void
ResetUSB(void)
{
    WriteUSBreg(MCNTRL,     0x01 | 0x04);       // USBリセット,3.3V供給
    wait100ms();                                // 100msec待ち
    WriteUSBreg(MCNTRL,     0x40 | 0x04);       // 割込設定(Active Low OC)
    WriteUSBreg(CCONF,     (USB_CLKDIV-1));     // 24MHz/4 = 12MHz
}
```

14. フラッシュメモリ書き込み方法

本カードの CPU は外部からシリアルポート(SCI1)を通じて受信したプログラム、データを内蔵フラッシュメモリに書き込む機能を持っています。CPU 動作モードをブートモードにすることにより、書き込み可能な状態になります。

14.1. 本カードのモード設定

パソコンの COM ポートと本カードのシリアルポート（J11）を接続し、書き込みモード（ディップスイッチ SW1 の１を OFF、２を ON）にして、本カードの電源を入れます。

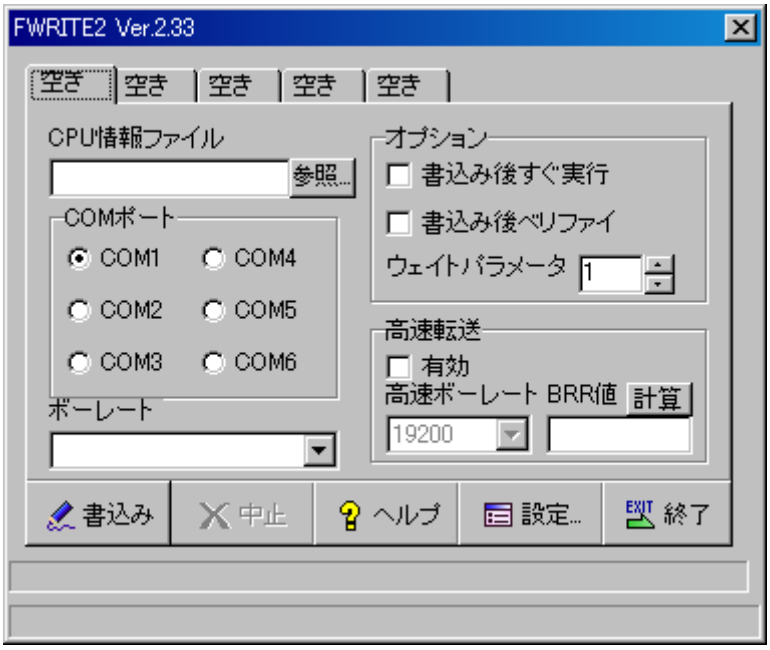


ブートモード時のスイッチ設定

14.2. YellowIDE 付属ツールを使用する

YellowIDE 付属のフラッシュメモリ書き込みツールを使用する場合について解説します。

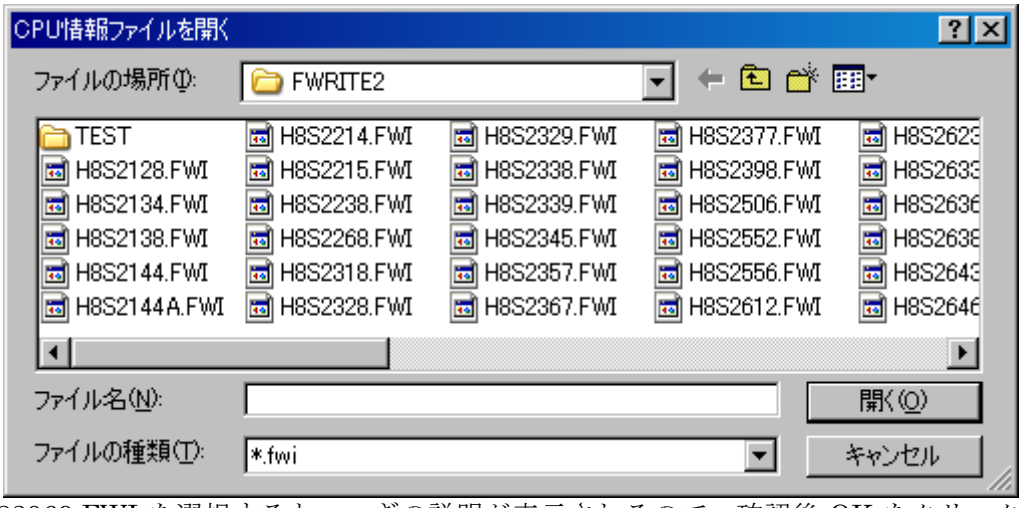
FWRITE2 を起動すると次のウインドウが開きます。初めて起動したときはパラメータが登録されていないため、空白の部分があります。この場合はパラメータを設定します。



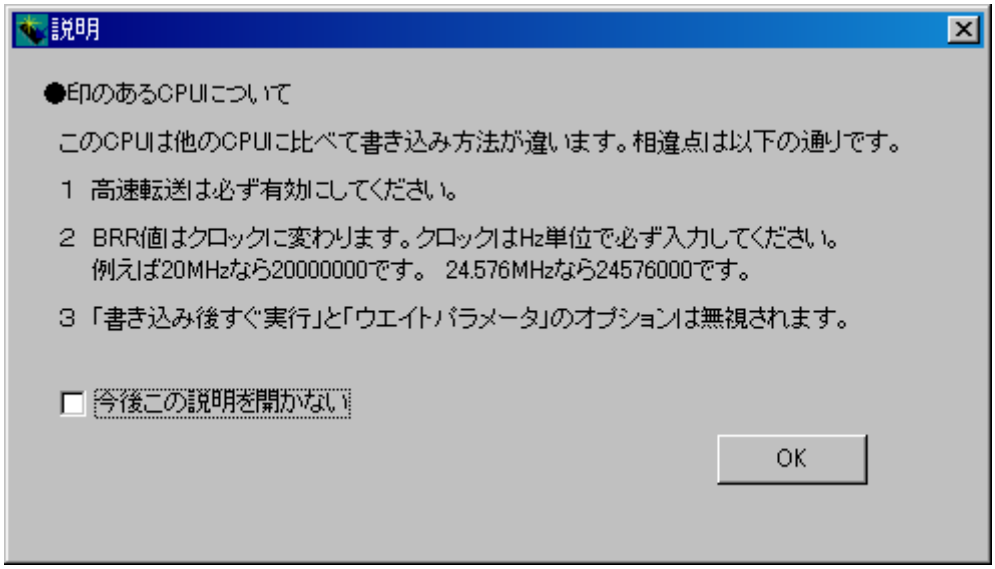
このウインドウではCPUが登録されていないので画面上部のタブはすべて空きになっています。

まず、書き込み対象の CPU を登録します。

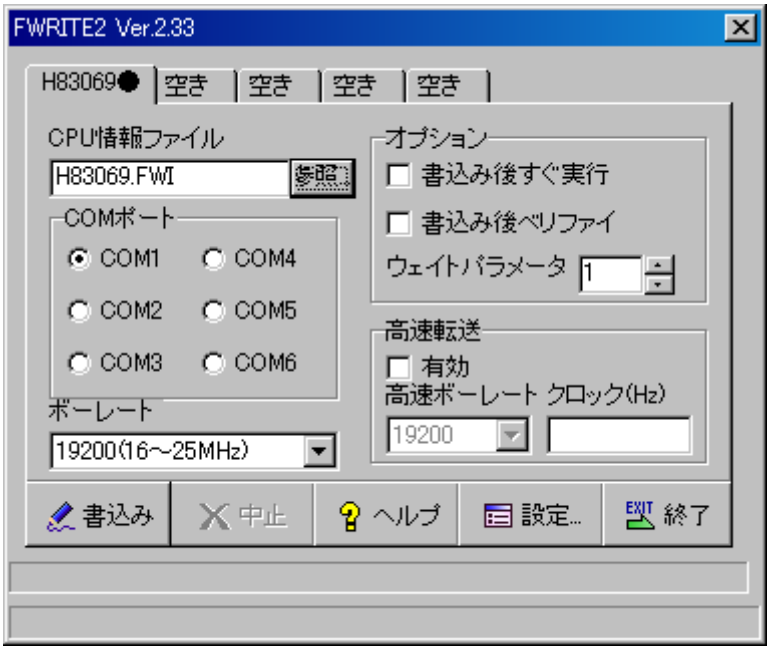
CPU 情報ファイルの参照ボタンをクリックすると、情報ファイルが表示されます。ここで、H83069.FWI を選択し、開く(O)をクリックします。なお、情報ファイルはC:¥YellowIDE6¥FWRITE2 内に格納されています。



H83069.FWI を選択すると、つぎの説明が表示されるので、確認後 OK をクリックします。



CPU が登録されるとタブに CPU の名称（H83069=H8 の 3069）が表示されます。



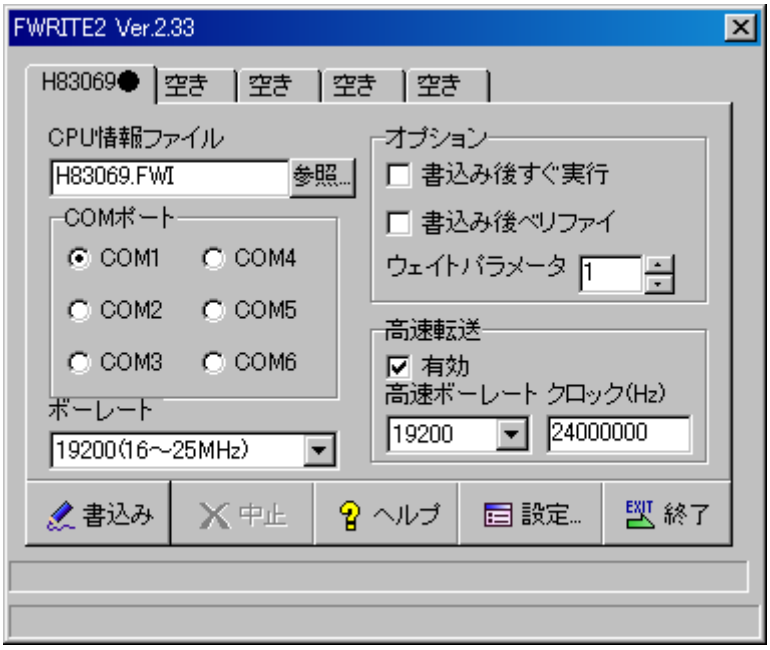
使用する COM ポートを選択します（例では COM1）。

つぎに、高速転送を有効にし（有効のチェックボックスを選択すると”高速ボーレート”と”クロック(Hz)”が有効になります）、クロック(Hz)に本カードのクロックを Hz 単位（24000000）で設定します（24 のあと 0 が 6 個）。

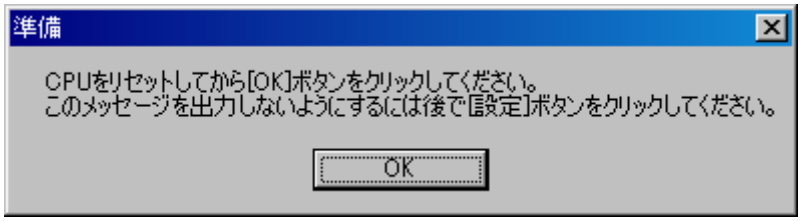
他はデフォルトで使用します。

これで FWRITE2 の準備は完了しました。パラメータは保存されるので、次回以降 FWRITE2 を起動したときはこれまでの、設定作業は不要です。

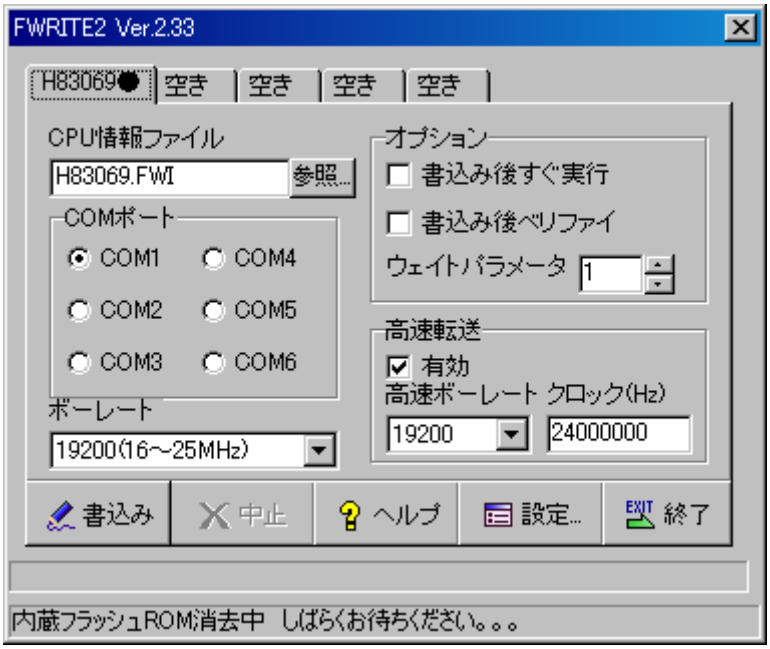
設定後の画面はつぎのとおりです。



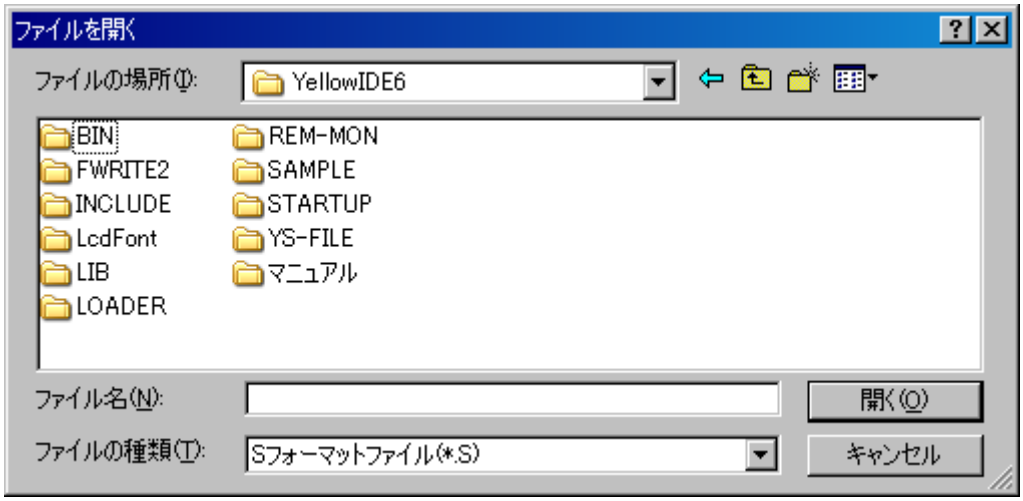
書込みボタンをクリックすると準備メッセージが表示されます。



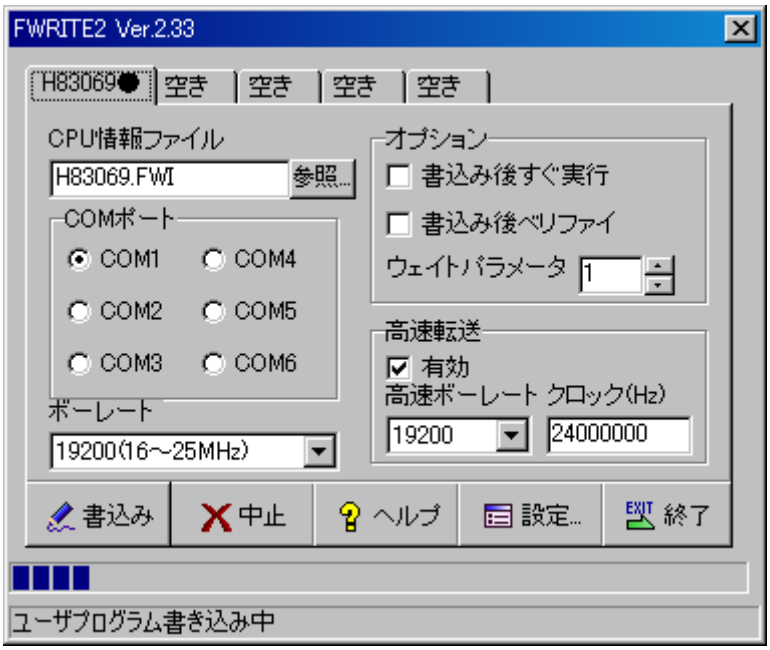
OK ボタンをクリックすると本カードと通信確認をし、フラッシュメモリ消去を実行します。



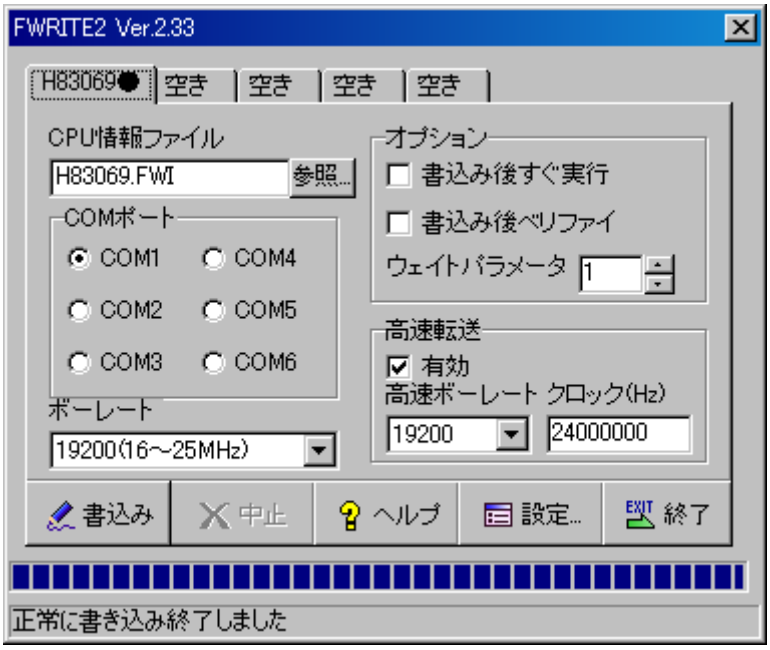
消去が完了すると、書き込みファイルを選択するダイアログウインドウが開きます。



ファイルを選択すると CPU の内蔵フラッシュメモリの消去・書き込みが開始します。



書き込み動作実行中は FWRITE2 ウインドウ下部に進行状況がバーグラフで表示されます。



ウインドウ最下部に終了メッセージが表示されると FWRITE2 を終了し、本カードのディップスイッチ SW1 の 1 を OFF、2 を OFF にし、モードを通常モードに戻してください。

電源を再投入すると、書き込んだプログラムの実行が開始します。

15. 標準設定

15.1. SRAM 設定

アドレス範囲	0x800000～0x9FFFFF
容量	2MByte
バス幅	16 ビット
ステート数	3
ウェイト数	0

15.2. 内蔵 I/O デバイス用バス設定

アドレス範囲	0xC00000～0xDFFFFF
容量	64kByte×4 エリア
バス幅	8 ビット
ステート数	3
ウェイト数	3

15.3. I/O 方向（DDR）設定

ポート	入出力方向								備考
	D7	D6	D5	D4	D3	D2	D1	D0	
P1	1	1	1	1	1	1	1	1	アドレスバス
P2	1	1	1	1	1	1	1	1	アドレスバス
P3	0	0	0	0	0	0	0	0	データバス
P4	0	0	0	0	0	0	0	0	データバス
P5	×	×	×	×	1	1	1	1	アドレスバス
P6	0	0	0	0	0	1	1	0/1	汎用 IN、内蔵時計
P7	×	×	×	×	×	×	×	×	入力専用ポート
P8	×	×	×	0	0	0	0	0	入力、割り込み
P9	×	×	0	0	0	0	0	0	割り込み、SCI
PA	0	0/1	0/1	0/1	0/1	0/1	0/1	0/1	汎用ポート
PB	0	0	1	1	0	0	0	0/1	EEPROM､CS、SCI

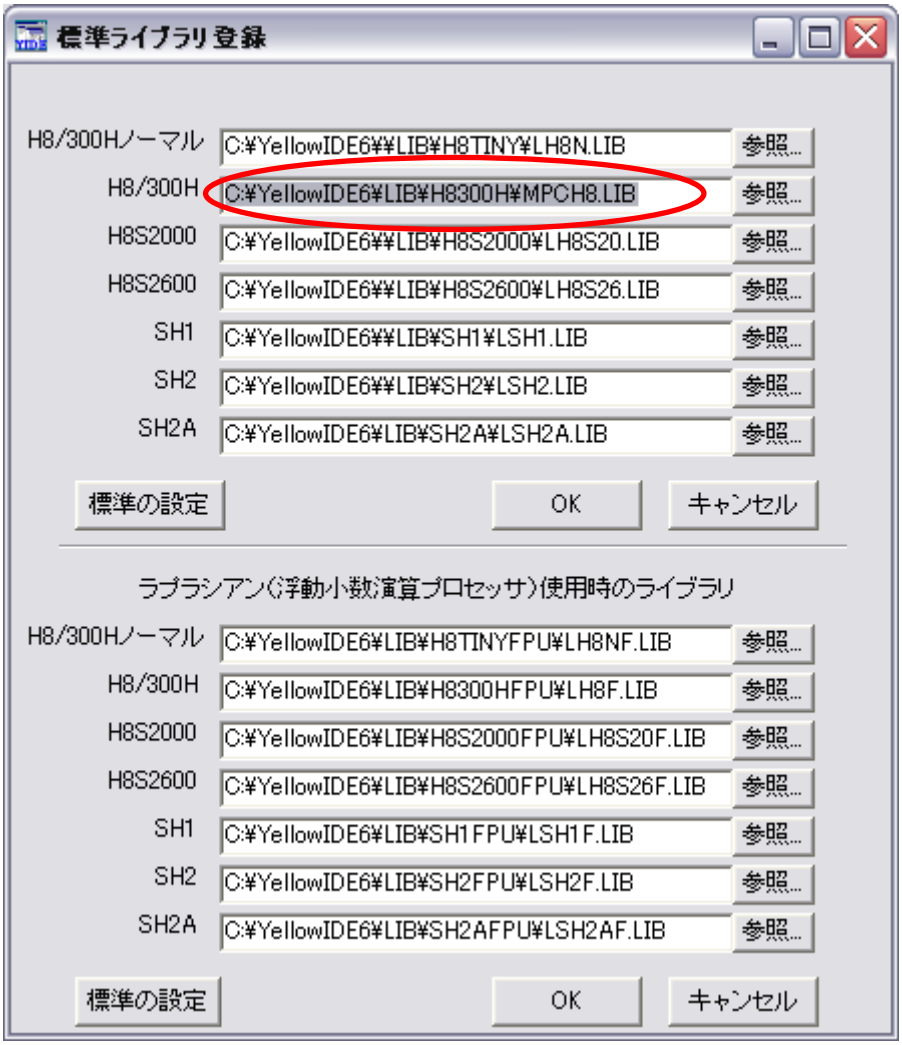
0　　：入力ポートまたは初期値
1　　：出力ポート
0/1　：入出力を切り替えて使用
×　　：存在しない

16. YellowIDE による開発

開発環境に YellowIDE を利用される場合は添付のライブラリをアプリケーションに組み込むことにより、標準入出力および時刻取得が可能です。

16.1. ライブラリ組み込み

添付のライブラリ(MPCH8.LIB)を C:¥YellowIDE6¥LIB¥H8300 に保存します。
YellowIDE のツールバーから設定メニューを開き、標準ライブラリを選択します。
標準ライブラリ登録ウインドウの H8/300H の項目に保存したライブラリを選択します。



16.2. 標準 IO ライブラリ使用例

```
#define    FILE_IO    std_sio1              // SCI0 を標準 IO ポートに

#define    putch(c)              fputc(c,FILE_IO)
#define    getch()               fgetc(FILE_IO)

void test_func(void)
{
```

```
        char c;

        while(!(c=kbchk(FILE_IO)))              // SCI0 からの入力をチェックする。
                ;                               //  データを受信すれば受信コードが返る。
                                                // 0 なら入力はないのでループする。

        if (c=='Q' || c=='q' || c==0x03)         // Q または q、CTRL-C か？
                fprintf(FILE_IO,"    BYE¥n¥n");
        else
                fprintf(FILE_IO,"    MPCH-8CPU TEST¥n¥n");
}
```

16.3. クロックライブラリ使用方法

本ライブラリは旧モデル(V1 のつかない MPCH-8CPU)のリアルタイムクロックモジュールにも対応させるため、自動的にモジュールの種別を判別します。判別させるためには clk_read_register 関数を呼ぶ必要があります。

```
unsigned char c;
char buf[7];                    //6 文字以上の文字列バッファを準備
clk_read_register(0, 1, data);  //ダミーリード ： リアルタイムクロックの種別を判別
c = clk_get_device(buf);        //デバイス名と番号を取得。戻り値は
                                // C=0, buf="------"(判別前)  または
                                // C=1, buf="RX4702"(旧ボード) または
                                // C=2, buf="RX4571"(新ボード)
```

リアルタイムクロックにアクセスする関数のプロトタイプ宣言は INCLUDE¥H8 内の CLK_MOD.H 内に記述されております。

また時刻、日付・時刻呼び出し用につぎの関数が準備されています。

ANSI 標準関数

time	現在時刻の取得
asctime	tm 構造体を文字列に変換
ctime	現在時刻を文字列に変換(ローカル時刻) *1
mktime	tm 構造体を time_t に変換
localtime	tm 構造体への変換(ローカル時刻) *1
gmtime	tm 構造体への変換(UTC 時刻)
difftime	2 つの時間の差(秒)
ctime	ローカル時間を文字列に変換
strftime	tm 構造体を様々な書式の文字列に変換 *2

*1 タイムゾーンは 0 に設定されているので UTC 時刻と同じ。

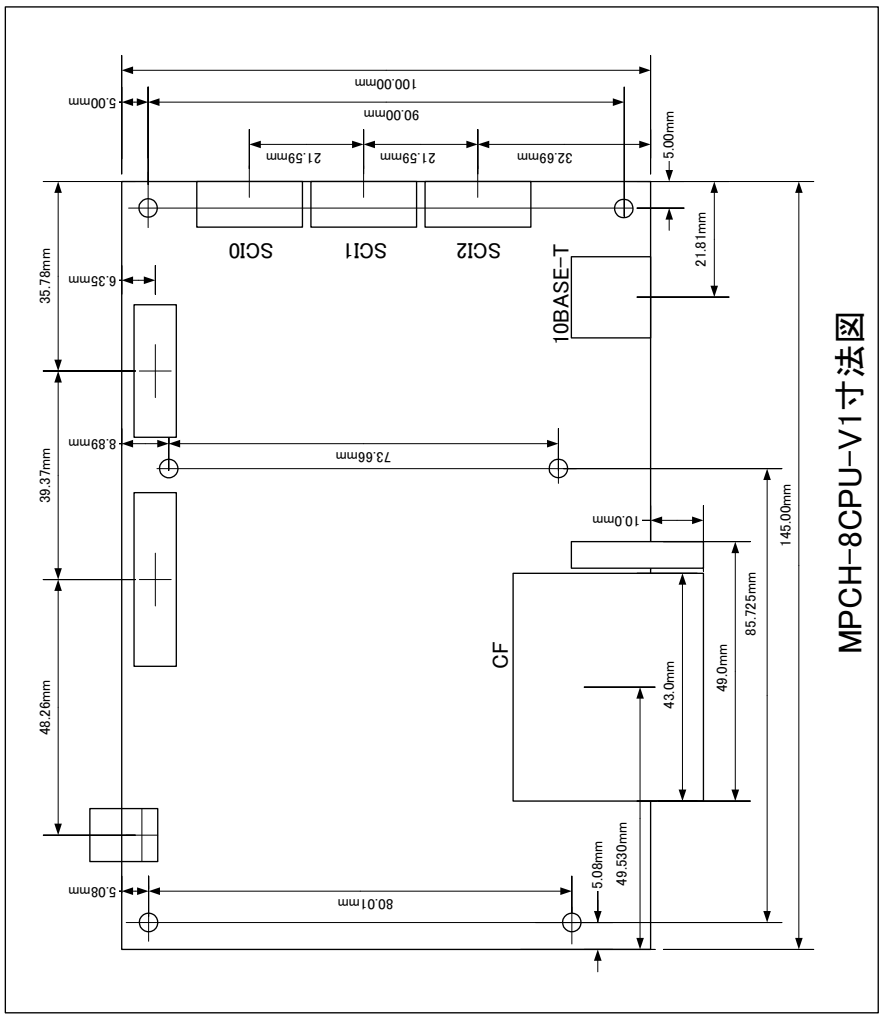
*2 タイムゾーン名(%Z)は無視されます。

独自関数

int	SetNowTime(struct tm *t);	リアルタイムクロックの設定
time_t	GetNowTime(struct tm *t);	リアルタイムクロックから読み出し

上記関数は添付のライブラリ関数 mpctime.c 内に記述されております。

17. 寸法図


MPCH-8CPU-V1寸法図
100.00mm
90.00mm
5.00mm
5.00mm
21.59mm
21.59mm
32.69mm
SCI0
SCI1
SCI2
10BASE-T
21.81mm
35.78mm
6.35mm
8.89mm
73.66mm
39.37mm
10.0mm
145.00mm
85.725mm
49.0mm
43.0mm
CF
48.26mm
5.08mm
49.530mm
5.08mm
80.01mm

18. 参考データシート

CPU

ルネサステクノロジ
H8/3069R F-ZTAT ハードウェアマニュアル

シリアル EEPROM

ATMEL 2-wire Serial EEPROM 1M(131,072 × 8)
AT24C1024

USB コントローラ

National Semiconductor Full Speed Node Controller
USBN9604

LAN コントローラ

REALTEK SEMI-CONDUCTOR
Full-Duplex Ethernet Controller
RTL8019AS

リアルタイムクロック

SEIKO EPSON CORPORATION
Real Time Clock Module
RX-4571

コンパクトフラッシュメモリカード

CompactFlash Association
http://www.compactflash.org/

MPCH8-CPU 取扱説明書

株式会社エンベデッドテクノロジー

〒578-0946
大阪府東大阪市瓜生堂 3 丁目 8-13
奥田ビル 2F
TEL：06-6224-1137
FAX：06-6224-1138

http://www.emb-tech.co.jp/