## 実践!自動車組込み技術者入門

# FPGAとマイコンの連携システム

## (ソフトウェア編)

~The first step is the only difficulty.



東北の復興を担う自動車組込みエンジニア育成支援プロジェクト

#### Content

0		はじめに	3
1		マイコンボードの仕様	<b>5</b>
	1.1	概要	<b>5</b>
	1.2	MCU 回路	17
	1.3	基本 I / O 回路	17
<b>2</b>		HEW による開発環境	27
	2.1	概要	27
	2.2	E8a エミュレータパッケージ付属 CD-ROM のインストール手順	29
	2.3	HEW の起動	33
	2.4	新規プロジェクトの作成	34
	2.5	ワークスペースとプロジェクト	42
	2.6	コーディング、ビルド(ロードモジュールの生成)	43
	2.7	E8a を使用したデバッグの準備	49
	2.8	プログラムの実行	57
3		シリアルインターフェース	61
	3.1	概要	61
4		距離データの応用	64
	4.1	概要	64
	4.2	ソフトウェア詳細仕様	65
	4.3	マイコンのピンアサイン(ピン配置)	67
	4.4	システム構成上の問題点	68
	4.5	距離計システムを動作させる	72
<b>5</b>		おわりに	85
6		Appendix	86
	表1	部品表	86
		参考文献・引用文献	87



# 

本編(ソフトウェア編)は、前編(ハードウェア編)で設計・製作したFPGAボード が計測した距離データをマイコンボード側で受け取り、実際に障害物までの遠近感が人間 の五感で分かるような近接センサーシステムを完成させることを目標としています。

「完成」までの過程で、どのような技術が要求されるのか、組込みソフトウェアエンジ ニアとしてなにさらに何を修得すべきかが理解できるように、組込みソフトウェア開発に 求められる技術世界を一端を一目で俯瞰できるような体験型入門書です。

特徴をまとめると

- 1. 言語については触れません。
- 2. プログラムの設計・製作はしません。
- 3. 設計論、コード書法、情報理論、工程管理、品質保証・管理、プロジェクト管理、 業務知識(自動車の各種ECU、カメラ、モバイル機器の構造・構成要素など)に は触れません。
- 4. マイコンボード基板の電子部品・電子回路が正常動作するか、測定器を使って確認 します。
- 5. マイコンハードウェアマニュアルの電気的特性や電子部品のデータシートを読みま す。ポートに流れる許容電流を把握する必要性があることを理解します。
- マイコンの開発環境(本編ではルネサスエレクトロニクス社の Hew(High performance Embedded Workshop)を構築し・機能・操作方法を学びます。
- ポート制御を行う簡単なプログラムを作成することで、開発環境を活用したソース コードの生成・コンパイル・アセンブル・リンク・実行といった一連の手順を理解 します。
- 8. 入力/出力の意味を理解します。
- 9. 代表的なシリアルインタフェース通信規約を理解し、活用します。
- 10.ソフトウェア仕様書を理解します。

11. FPGAとマイコンボード間の通信の電圧レベルが揃っていないための対策を検 討します。 12. 電圧レベル変換回路を通して、プルアップ抵抗、High レベル、Low レベル、ハイ インピーダンスの意味を理解します。

13. マイコンボードにサンプルプログラムをダウンロードして、デバッガを使ったテ ストを実施します。

14. FPGAボードとマイコンボードを接続して実際に距離データをブザーやLED に表示して、近接センサーとしての動きを確認します。

組込みソフトウェア技術の世界では、自分の立ち位置で修得すべき技術が大きく変化しま す。誤解を恐れずに言えば「言語」を知っていればプログラムは作れるかもしれませんが、 システムは完成しません。

本編は組み込みソフトウェア技術者の入門者向け講座ですのでハードルは低く設定して あります。ハードルを全て越えたとき組込みソフトウェア技術の深さと広がりを認識して いただければ幸いです。

2013年2月



### マイコンボードの仕様

#### 1.1 概要

マイコンボード (Seamark-1: Basic board) は、MCU回路と基本 I/O回路で構成されます。MCU回路と基本 I/O回路はGND以外は電気的に接続されていません。マイコン H8/3694F のポートから入出力制御を行うには入出力ポートの端子割付を行い、MCU回路の I/O拡張端子 (Term8) と基本 I/O回路の各ターミナルをピンクリップ等で結線する必要があります。

写真1にマイコンボード (Seamark-I: Basic board))の写真、図1にマイコンボードの ブロック図を示します。





図1 マイコンボード (Seamark-I) ブロック図

#### 1.2 MCU回路

MCU回路は下記より構成されています。

- ・H8/3694F 搭載マイコンボード MB-H8A(Sunhayato 製)
- ・I/0 拡張端子
- ・リセット回路
- ・電源回路
- ・H-UDI 接続回路

#### ■ H8/3694F 搭載マイコンボード MB-H8A (Sunhayato 製)

H8/3694F 搭載マイコンボード MB-H8A の回路図を図2に、端子割付を表1に示します。



#### 図2 MB-H8A 回路図

CN1 番号	3694F 端子番号	機能	備考	CN2	3694F 端子番号	機能	備考
1	31	P85	ブートモード時H	1	30	P84/FTIOD	
2	32	P86		2	29	P83/FTIOC	
3	33	P87		3	28	P82/FTIOB	
4	34	P20/SCK3		4	27	P81/FTIOA	
5	35	P21/RXD		5	26	P80/FTCI	
6	36	P22/TXD		6	25	NMI	ブートモード時L
7	37	P14/IRQ0		7	24	P76/TMOV	
8	38	P15/IRQ1		8	23	P75/TMCIV	
9	39	P16/IRQ2		9	22	P74/TMRIV	
10	40	P17/IRQ3/TRGV		10	21	P57/SCL	
11	41	PB4/AN4		11	20	P56/SDA	
12	42	PB5/AN5		12	19	P12	
13	43	PB6/AN6		13	18	P11	
14	44	PB7/AN7		14	17	P10/TMOW	
15	45	PB3/AN3		15	16	P55/WKP5/ADTRG	
16	46	PB2/AN2		16	15	P54/WKP4	
17	47	PB1/AN1		17	14	P53/WKP3	
18	48	PBO/ANO		18	13	P52/WKP2	
19	5	RES	しでリセット	19	12	P51/WKP1	
20	4	VCL	内部降圧電源端子	20	11	P50/WKP0	
21,23,25,26	7	VSS	グランド端子	21,22,24,26	7	VSS	グランド端子
22,24	1	AVCC	A/D変換用電源	23,25	10	VCC	電源端子

表1 MB-H8A 端子配列

H8/3694F 搭載マイコンボード MB-H8A には、㈱ルネサスエレクトロニクス社の16 ビット CISC マイコン H8/3694F が搭載されています。図3に H8/3694F の内部ブロック図を示します。

8

#### ① H8/3694F

H8/3694Fの特長

- ・ 16bitCISCマイコン
- ・ 32kバイトのフラッシュメモリ
- ・ 2kバイトのRAM
- ・ 10ビット8チャンネルADコンバータ
- ・ ウオッチドッグタイマ
- ・ 多機能タイマの他にシリアル
- I2Cバスインターフェイスを内蔵
- エミュレータインターフェイスを内蔵
- 高速20MHz動作
- ・ サブクロック



図3 F-ZTATTM 版、マスクROM 版 H8/3694 グループ内部ブロック図

マイコン内蔵フラッシュメモリは、最大書き込み回数を超えるとプログラムのダウンロード等ができなくなる可能性があります。消耗品として扱います。図4にピン配置、図7.5 に H8/3694F のメモリマップを示します。





図5 メモリマップ

#### ■ I/O拡張端子&回路

MCU H8/3694Fは48ピンパッケージ(64ピン版もあります)ですが、48ピンのうち 40ピンを基本I/O回路部の各種電子デバイスに接続できるようにI/O拡張用端子まで 引き出しています。表2にマイコンH8/3694Fのピン番号とマイコンボードI/O 拡張端子(Term8)番号の関係を示します。

マイコンボードの I /O 拡張端子 (Term8) 番号 H8/3694F のピン番号					
1/0耳振	山8/2604日建	_	MC	1#-6	
<u>Din番号</u>	端子名称	端子番号	CN番号	pin番号	備老
1	*RESET	5	1	19	F8接続
2	PB1/AN1	47	i	17	201247
3	PB3/AN3	45	1	15	
4	PB6/AN6	43	1	13	
5	PB4/AN4	41	1	11	
6	P16/*IRQ2	39	1	9	
7	P14/*IRQ0	37	1	7	
8	P21/RxD	35	1	5	
9	P87	33	1	3	E8接続
10	P85	31	1	1	E8接続
11	P83/FTIOC	29	2	2	
12	P81/FTIOA	27	2	4	F a + * 4 +
13	*NMI	25	2	6	E8接続
14	P/5/TMCIV	23	2	8	
15	P57/SCL	21	2	10	
10	PIZ	19	2	12	
1/		15	2	14	
10		10	2	10	
20		11	2	20	
20	P50/*₩KP0	12	2	19	
22	P53/*WKP3	14	2	17	
23	P55/*WKP5/*ADTRG	16	2	15	
24	P11	18	2	13	
25	P56/SDA	20	2	11	
26	P74/TMRIV	22	2	9	
27	P76/TMOV	24	2	7	
28	P80/FTCI	26	2	5	
29	P82/FTIOB	28	2	3	
30	P84/FTIOD	30	2	1	
31	P86	32	1	2	E8接続
32	P20/SCK3	34	1	4	
33	P22/TXD	36	1	6	
34	P15/*IRQ1	38	1	8	
35	P1//*IRQ3	40		10	
36	PB5/AN5	42		12	
37	PB//AN/	44		14	
38	PB2/ANZ	40		10	
39	PB0/ANU VCI	48		18	
40	VUL	4		20	1

表2 H8/3694F ピン番号とマイコンボード拡張端子(term 8)番号



図6 MB-H8A 接続コネクタと I / O拡張端子

#### ■ リセット回路

リセットスイッチ (SW21) を押下するとマイコンのリセット端子が Low になりMCUに 外部リセットがかかります。



図7 リセット回路

◆参考:リセット動作の実際

図7のリセット回路を例に動作を説明します。

① パワーオンリセット

電源が投入されるとR22(47k)を通じてC21(10 $\mu$ F)を充電し、Reset端 子電圧を徐々に上がっていきます。電源電圧が十分に上昇してからマイコンが安定 動作する十分な時間が経過するとリセットは解除されます。

回路中のD21は、電源がオフになったとき速やかにC21の電圧を下げて、次 に電源がオンになったときに備える<u>放電促進ダイオード</u>です。

② マニュアルリセット

リセットスイッチ(Sw21)を押すとReset電圧端子はOvになります。パワー オンリセット時の「マイコンが安定動作する十分な時間を待って」は、リセットス イッチを押すことで実現します。リセットスイッチ(Sw21)を離すとReset端子 は電源電圧まで上昇する過程で電源電圧のある一定レベルを超えるとリセットが解 除されます。

R21(10kΩ)はC21から大きな電流が流れて回路が損傷しないように電制 限の役目をする電流制限抵抗です。

#### ■ 電源回路

ACアダプタを使用してAC100vをDC5vに変換したものがDCジャック(J11) を通してマイコンボードに供給されます。

Power SW (SW11) をONにすると赤色LED (LED11) が点灯しDC5VがMCU回路に供給されたことを表示します。

Power SW (SW12) をONにすると赤色LED (LED12) が点灯しDC5Vが基本 I / O回 路に供給されたことを表示します。



#### ■ H-UD | 接続回路

E8a(またはE8)エミュレータ本体とユーザーシステム(マイコンボード:Seamark-I)を接続するコネクタ(CN3)とその周辺回路、プルアップ抵抗から構成されます。



図9 H-UDI接続回路



図10 E8a接続コネクタのピン配置

#### 1.3 基本 | / O回路

基本 I / O部は、下記電子部品より構成されています。

- LED:4
- 圧電ブザー(他砺式):1
- 半固定抵抗:1
- ・ タクトスイッチ(プルアップ抵抗のオン/オフ可能なジャンパーSW 付き):2
- DIPスイッチ(4回路):1
- 7セグメントLED:1
- トランジスタ用ソケット:1(トランジスタは未実装)
- 14ピンICソケット:1 (今回の講座用に 74HC07AP が搭載済みです)

LEDのを点灯/消灯、SWのオン/オフ操作時の入力端子の信号レベルの確認、TT L ICの使う実験する場合は、マイコンボードの I/O Power SW(SW12)をオンにします。

#### ■ LED

基本 I /OボードのLED1~4は、基本 I /O回路の Term1 の1番から4番(/LED1~/LED4)を Low にすると点灯します。このときLEDには約10mAの電流が流れます。マ イコンのポート8(大電流出力ポート)を使用するとLED1~4を直接ドライブできま

す(Low アクティブ)。それ以外のポートを使用してLED1~4をドライブするとマイコンの出力ポート許容電流の最大定格を超えてしまい、当該ポートが破壊される可能性があります。





図 11の回路では 330 Ωの抵抗が入っています。

項目	記号	適用端子	測定条件		規格値		単位
				Min	Тур	Max	
出力 Low レベル 許容電流	loL	ポート 8、SCL、SDA 以外の出力端子	Vcc=4.0~5.5V	—	—	2.0	mA
(1 端子あたり)		ポート 8		_		20.0	mA
		ポート 8、SCL、SDA 以外の出力端子		—	/	0.5	mA
		ポート 8		_	/ [	10.0	mA
		SCL, SDA		_	- / [	6.0	mA
出力 Low レベル 許容電流	Σlol	ポート 8、SCL、SDA 以外の出力端子	Vcc=4.0~5.5V	_	$7 \lceil$	40.0	mA
(総和)		ポート 8、SCL、SDA		_	7	80.0	mA
		ポート 8、SCL、SDA 以外の出力端子		/		20.0	mA
		ポート 8、SCL、SDA		/	E	40.0	mA
出力 High レベル 許容電流	—Іон	全出力端子	Vcc=4.0~5.5V	1		2.0	mA
(1 端子あたり)				7		0.2	mA
出力 High レベル 許容電流	— Σ Іон	全出力端子	Vcc=4.0~5.5V	7		30.0	mA
<mark>(総</mark> 和)					_	8.0	mA

表3 H8/3694F DC特性

ポート8は、1端子あたりの出力 Low レベルの 許容電流(シンク許容電流)が 20mA あり、他 のポート端子に比べて許容電流値は大きい。

#### ■ 圧電ブザー

圧電ブザーは他励式ブザーです。外部から Term1 の6番(Bz) に一定周波数のパルス を入力すると圧電ブザーが鳴動します。



図12 圧電ブザー回路

#### ■ 半固定抵抗

半固定抵抗(VR41)を廻すことで5vを分圧した電圧が Term1の5番(Volume)に 出力されます。



図13 半固定抵抗回路

#### ■ タクトスイッチ

タクトスイッチ1 (SW71) とタクトスイッチ2 (SW72) にはプルアップ抵抗を 接続する/しないを設定できるジャンパSW (JP71、JP72) が取り付けてありま す。プルアップ抵抗を接続した状態でタクトスイッチを押すと、Term4 の1番と2番には、 Low レベルの信号が出力されます。



図14 タクトスイッチ回路

#### ■ DIPスイッチ

4回路を内臓したDIPスイッチ(DIPSw)はプルアップ用の集合抵抗が接続されて います。4回路のそれぞれのスライドスイッチを「on」側へセットするとTerm5の1番か ら4番の各端子はLowになります。



図15 DIPスイッチ回路

#### ■ 7セグメントLED

7セグメントLED(7 seg101)はアノードコモンのLEDです。シンクドライバIC (IC3: TD62083)を使ってドライブしています。TD62083はインバータ 方式のシンクドライバなので、各セグメントの制御端子 Term7の1番から8番(Seg.a~ Seg. dp)をHighにすると7セグメントLEDの各セグメントが点灯します。



図16 IC3: TD62083 (IC3) ピン接続図と基本回路図



図17 7セグメントLED回路

#### ■ トランジスタ用ソケット

トランジスタやFETを接続するためのソケットで、電気的にはどことも接続されていません。

#### ■ 14 ピン | Cソケット: | C2

14 ピン I C ソケットの 14 ピンが 5 v 電源(V c c)へ、14 ピンソケットの 7 ピンは 0 v (GND)に接続されています。



図18 14ピンICソケット回路図

#### 【実習1】基本 | / 〇 ボードの動作確認

■ 手動操作(マイコンから制御しない)で次の回路動作を確認します。

①LED1~4
②7セグメントLED
③タクトスイッチ1,タクトスイッチ2

#### ■ 手順

LED1~4の端子、Term1 の1番(/LED1)から4番(/LED4)を順次GNDへ接続します。

⇒ LED1からLED4は点灯しますか。

②7セグメントLEDの端子、Term7の1番(Seg.a)から8番(Seg.dp)を順次Vcc に接続します。

⇒ 7セグメントLEDのセグメント a~dp は順次点灯しますか。

③SW1、SW2のプルアップ抵抗が「ある」場合と「ない」場合について動作確認 しましょう。

#### ・プルアップ抵抗を接続した場合

プルアップ抵抗用ジャンパ (JP71, JP72) の2番と3番をショートした場合、SW71、SW72を押したときは0v、離したときは大よそ5vになりますか?

◆ 解説1:プルアップ抵抗の機能

電流は電圧の高いところから低いところへ流れます。水の流れと一緒です。

スイッチ1 (SW71)をオンするとほとんどの電流が電源 Vcc からプルアップ 抵抗 R71、そして短絡しているスイッチ1 (SW71)を通して GND に向かって流れ ます。電流がマイコン入力ポート側 (Term41) へはほとんど流れないのはマイコン の入力ポートの内部抵抗が非常に大きい(約 1MΩ程度)ためで、プルアップ抵抗を 通った電流はマイコンの入力ポート側へはほとんど流れずにGND側へ流れます。 このときマイコンポート側 (Term4 の/SW1、/SW2)の端子電圧は0 v です。

スイッチ1 (SW71)をオフすると、電流は電源 Vcc からプルアップ抵抗 R71 を介 して全てマイコンポート側 (Term41) へ流れます。スイッチ1 (SW71)が切れてい るためで、このとき、Vcc (5 v) をプルアップ抵抗 (4.7 k  $\Omega$ ) と入力ポートの内 部抵抗 (約 1M $\Omega$ ) で分圧するのでほとんど5 v に近い電圧となります。

◆ 解説2:プルアップ抵抗の値は?

プルアップ抵抗を小さくすれば、限りなく Vcc に近い電圧になり High レベルと しては良いプルアップ抵抗値なのでしょうか?たとえば1 $\Omega$ のプルアップ抵抗を付 けた場合はスイッチを押したときは 5A (5v/1 $\Omega$  = 5A) も流れます。回路パターン が焼けて煙が出るでしょう。

大きな抵抗を付ければいいのでしょうか?たとえば  $1M\Omega$ の抵抗を付けると、スイ ッチを押したときはほんのわずかな電流が流れて0v、スイッチを離したときはプ ルアップ抵抗の  $1M\Omega$ とマイコン入力ポートの内部抵抗( $1M\Omega$ とすると)で Vcc の 5vを分圧しますから、2.5vとなる計算です。スイッチをオフしたときの電圧レ ベルが 2.5vでは High とも Low とも判定しない「不定」レベルです。

ここから、プルアップ抵抗値は High レベル電圧がマイコンのDC特性で許容して いる電圧より十分高く、かつスイッチをオンしたときに流れる電流があまり大きく ない(すくなくとも数ミリアンペア程度)ような値に設定する必要があります。

#### ・プルアップ抵抗を外した場合

プルアップ抵抗用ジャンパ (JP71, JP72) の1番と2番をショートした場合、SW 71、SW72を押したときは0v、離したときも0vになりますか?

#### ◆ 解説

プルアップ抵抗がない状態でスイッチ(SW71、SW72)を押すとプルアップ抵抗がある場合と同様の電気回路が形成されますから0vになります。

プルアップ抵抗がない状態でスイッチを離したとき、オシロスコープでマイコンボードのターミナル (/SW1、/SW2) 観測するとスイッチを押したときと同様に0 v に見えます。そこで 1div あたりの電圧レンジを拡大してみましょう。発振していることがわかりますか?

プルアップ抵抗を外した状態では電気的な回路が形成されていません。すなわちオ ープン状態の電圧を測っていることになります。マイコン内部のポートデータレジス タは0 (Low) または1 (High)を格納していますが、実際は「不定」ということです。



【課題1】 圧電ブサー回路の動作確認をする方法を検討しましょう。

#### ◆ 解説

マイコンボードに搭載されている圧電ブザーは他励式ですから、Term1 の Bz 端 子にパルス発信器などで外部から一定周波数のパルスを与えると当該周波数で鳴動 します。

パルス発信器がない場合は、CMOSのシュミット・トリガ・インバータIC (74HC14 など)をマイコンボードの14ピンICソケットに挿入して下記の発振 回路を作成して鳴動させることができます。



## **2** HEWによる開発環境



#### 2.1 概要

マイコンのソフトウェア開発では、エディタ、コンパイラ、エミュレータといったツール が必要です。さらに開発ニーズや各種マイコン特性に合わせて多種多様な組み合わせが必 要となります。統合開発環境 High-performance Embedded Workshop は、それらのツールを 使いやすく統合するフレームワーク です。

HEWによる開発環境構築例を図1に示します。



図1 HEWによる開発環境構築例

E8aエミュレータパッケージ付属のCD-ROMをインストールすると図2に示すHEW (High-performance Embedded system Workshop)の環境が提供されます。

参考 ルネサスエレクトロニクス社のホームページから無償評価版の 「H8SX, H8S, H8 ファミリ用 C/C++コンパイラパッケージ」 (シミュレータデ バッガ付属)をダウンロードすることができます。このパッケージにはルネ サス統合開発環境 HEW (High-performance-Embedded Workshop) が同包 されていますが、エミュレータ・デバッガは付属しておりません。 したがってターゲットボードでのエミュレーションデバッグはできませ んが、エディタ、ビルド(コンパイル/リンク)、シミュレータ機能を使う ことはできます。



図2 HEW (High-performance Embedded system Workshop) の環境

#### 2. 2 E8a エミュレータパッケージ付属CD-ROMのインストール手順

① CD-ROM を開くと下記のような各種ファイルが見えます。



- ② HewInstMan.exe をクリックして起動します。
- ③ 1 of 3 Installer:「E8a Emulator Software」のインストール 表示された画面の指示に従って「次へ (N)」をクリックします。 最後に下記画面が表示されます。「完了」をクリックします。

E	a Emulator Software V.1.05	Release 01
		InstallShield Wizard の完了
		セットアップは、コンピュータへE8a Emulator Software のインストールを終了しました。
	RENESAS	
		✓ 注意事項を表示する。
	High-performance Embedded	
	Workshop <sup>4</sup>	
Γ	InstallShield	< 戻る(B) <b>完了</b> キャンセル キャンセル

「E8a Emulator Software」のインストールが完了すると、自動的に「**2 of 3 Installer**」 が実行されます。

 ④ 2 of 3 Installer: 「Renesas AutoUpdate Utility」のインストール 表示された画面の指示に従って「次へ(N)」をクリックします。



「Renesas AutoUpdate Utility」のインストールが完了すると、自動的に「**3 of 3 Installer**」 が実行されます。

#### ⑤ **3 of 3 Installer :** [Renesas Flash Development Tool] のインストール

表示された画面の指示に従って「次へ(N)」をクリックします。言語選択画面では 「Asia(Jananese)」にチェックをいれて、「Next >」をクリックします。

Renesas Flash Development Toolkit (v4.07) – InstallShield Wizard 🛛 🛛 🔀
Select Language
Language selection will determine the language of the installed help and user documentation. International (English) Sisia (Japanese) Asia (English)
Release Notes
<u>Kanonicia</u> <u>Rack</u> <u>N</u> ext > Cancel

⑥ ライセンスの認証

License 承認画面では「accept the terms of the license agreement」にチェックを入れて「Next >」をクリックします。

Renesas Flash Development Toolkit (v4.07) – InstallShield Wizard 🛛 🔀
License Agreement Please read the following license agreement carefully.
「ソフトウェア使用許諾契約書」 お客様(以下、「甲」といいます)とルネサス エレクトロニクス株式会社( 以下、「乙」といいます)とは、この「ソフトウェア使用許諾契約書」(以下 、「本契約」といいます)とともに提供されるソフトウェア及びそのマニュア ルにつき、以下の通り契約するものとします。 1. 用語の定義 (1) 「許諾ソフトウェア」とは、本製品に含まれるコンパイラ、アセンブラお よび関連する実行ブログラム及び「ライブラリ」並びにそのマニュアルをいい ます。 (2) 「指定システム」とは、甲が許諾ソフトウェアをインストールし、使用す るコンピュータシステムをいいます。
OI do not accept the terms of the license agreement
< <u>Back</u> <u>Next</u> Cancel

- ⑦ オプション選択
  - オプション選択画面では「.mot」にチェックを入れて「Next>」をクリックします。

Renesas Flash De	velopment Toolkit	(v4.07) – InstallShield Wizard 🛛 🔀
Select Options		RENESAS
Setup requires :	you to make the follo	owing choices:
Clean up old	settings. This optio	n removes any existing settings before
Associate data <sup>.</sup>	files. Data files wil	II open in FDT.
🔽 . a20	✓.fpr	
🔽.a37	.mot	
🗹.bin	.rec	
🔽 . cde	🔽 . s2	
🔽 . dd i		
		< <u>B</u> ack <u>N</u> ext > Cancel

⑧ インストール終了

下記画面が表示されたら、「終了」をクリックします。

💊 High-performance Embedded Workshopインストールマネージャ 🛛 🔀				
CONCESSOR Multi installation Active High-performance Embedded Workshop Switch over!	インストールが終了しました。 ■ツール製品のユーザ登録のお願い ルネサスではツール製品をご購入されたお客様にユーザ 登録をお願いしております。 ご登録いただくと、以下のようなメリットがあります。 ・ツールニュース配信サービス ・技術サポートサービス 詳細は以下の「ツール製品のユーザ登録のご案内」ボタ ンをクリック!			
Non-active	(ツール製品のフーザ登録のご室内) 終了	-		

#### 2.3 HEWの起動

インストールが全て完了したら、HEW を起動します。

「すべてのプログラム (P)」> 「Renesas」 > 「High-performance Embedded Worksop」 > 「High-performance Embedded Worksop」を選択します。





次に示す「ようこそ!」画面が表示されます。下記の手順にしたがって新規プロジェクト の生成を行います。

#### 2. 4 新規プロジェクトの作成

新規プロジェクトを作成する手順を示します。

「ようこそ」画面で「新規プロジェクトワークスペースの作成」にチェックをいれて「O K」をクリックします。

ようこそ!		? 🛛
	● 新規プロジェクトワークスペースの作成(□)	ОК
×	<b>Samual and a set of the set of t</b>	キャンセル
	○ 最近使用したブロジェクトワークスペースを開く(□):	(
	○ 別のプロジェクトワークスペースを参照する(B)	

ワークスペース名は「養成講座」、プロジェクト名は「project\_010」とします。

新規プロジェクトワークスペース			? 🗙	
プロジェクト プロジェクトタイプ Pappication Assembly Application Demonstrati	ワークスペース名(W): 養成講座 プロジェクト名(P): project_010 ディレクトリ(D): C:¥WorkSpace¥養成講座 CPLI種別(C): H85 H8/300	↓ 「養成講座」を入力 「project_010」を入力 参照(B)		
<b>く )</b> プロパティ…	ツールチェイン(T): Renesas H8S,H8/300 Standard	■ H	8S,H8/300を選 キャンセル	択

新規プロジェクトー1/9ーCPU	? 🛛
	ツールチェインパージョン: 7.0.0.0
	この7泊ジェクトで使うCPUのシリーズとタイプを選択して下さい。 CPUシリーズ: RS4 2600 2000 300H 300
	CPU\$/7*: 「3694F」を選択 36902 36911 36912 3694F 30976R 39086B
PP PP PP	選択したいCPUタイブがない場合は、ハードウェア 仕様の近いCPUタイプまたは"Other"を選択してく ださい。 
< 戻る(B)	次へ(N) > 完了 キャンセル

新規プロジェクトー2/9ーオブション	? 🔀
	<ul> <li>ケローハルオフ*ションを指定します。</li> <li>動作モード: Normal マ</li> <li>アト・レス空間: マ</li> <li>無除算器指定: マ</li> <li>テイフ*ラリ作成方針: コードサイズ、優先 マ</li> <li>スタック計算サイズ: ミディアム(2byte) マ</li> <li>8bit絶対領域指定: マ</li> <li>Pefault マ</li> <li>H0</li> <li>引数格納レジスタを2つから3つに変更 ▲</li> <li>double型の変数/引数をfloat型として扱う</li> <li>構造体パラメタ、リターン値をレジスタに書的付</li> <li>▲byteパラメタ、リターン値をレジスタに書的付</li> <li>▲byteパラメタ、リターン値をレジスタに書的付</li> <li>▲byteパラメタ、リターン値をレジスタに書的付</li> </ul>
< 戻る( <u>B</u> )	次へ(N) >完了キャンセル



## ◆ 参考 //Oライブラリ使用: チェックを入れると標準入出力ライブラリを活用できます。 ヒープメモリ使用: ヒープメモリとはライブラリ関数 malloc,realloc,calloc,new で使用する領域のこ とで、チェックを入れるとヒープ領域管理用の関数 sbrk()を使用できます。 I/Oレジス夕定義ファイル: チェックするとC言語で記述した I / O定義ファイル ("iodef.h") を生成します。
新規プロジェクトー4/9ー標準ライブラリ	? 🛛
	<ul> <li>ライク・ラリ:</li> <li>C言語の選択(L):</li> <li>C (C89)</li> <li>✓ untime: 実行時ルーチン</li> <li>✓ new: メモリ操作用ライブラリ</li> <li>Ctype.h: 文字操作用ライブラリ</li> <li>math.h: 数値計算用ライブラリ</li> <li>math.f.h: 数値計算用ライブラリ(各関数に stdarg.h: 可変個の実引数アクセス用ライ</li> <li>Stdio.h: 入出力用ライブラリ</li> <li>Stdlib.h: 標準処理用ライブラリ</li> <li>Stdlib.h: 標準処理用ライブラリ</li> <li>String.h: 文字列操作用ライブラリ</li> <li>ins(FC++): ストリーム入出力用クラスライ</li> <li>▲ 全て有効</li> <li>全て有効</li> <li>全て無効</li> </ul>
	バヘヘロノン 元了 キャンセル

新規プロジェクトー5/9ースタック領域	? 🔀
	スタックの設定を行って下さい。 スタックホ°インタアト <sup>®</sup> レス: (power-on reset) F <mark>IFF80</mark> スタックサイス: F <sup>I</sup> 100
< 戻る( <u>B</u> )	次へ(N) > 完了 キャンセル

新規プロジェクトー6/9ーベクタ	? 🗙
・ウタの設定を行います。         ・ウタの設定を行います。         ・ウタハンドラ:         ・ハンドラ ヘウタ         PowerON_Reset 0 Power On Rese	
〈戻る(B) 次へ(N) > 完了 キャンセ	μ

新規プロジェクトー7/9ーデバゥガ	? 🔀
	ターゲット:         ● ガット:         ● B Tiny/Super Low Power E8a SYSTEM 30         ● H8/300HA Simulator         ● H8/300HN Simulator         ● H8/300HN Simulator         ● チェックを入れる         ● ケットタイプ*:         300H         ターケットOPU:
< 戻る( <u>B</u> )	次へ(N) > 完了 キャンセル

新規プロジェクトー8/9ーデバッガオブション	? 🔀
	ターゲット名: H8 Tiny/Super Low Power E8a SYSTEM 300H コア: くsingle core> コンフィグシーション名: Debug_H8_Tiny_Super_Low_Power_E8 詳細オフ <sup>®</sup> ション: Item Setting 変更(M)
< 戻る( <u>B</u> )	次へ(N) > 完了 キャンセル



「概要」画面で「OK」をクリックします。



新規プロジェクトの生成が完了です。次のプロジェクト project\_010 のHEW初期画面が 表示されます

🛞 project_010 - High-performance Embedded Workshop				
ファイル(E) 編集(E) 表示(V) プロジェクト(E) ビルド(B) デバッグ(D)	基本設定(U) ツール(T) テスト(S) ウィンドウ(W) ヘルプ(H)			
	● 約 約 済 □ == 参  ● 約 約 済 □ == 参  ○ ● ● ● ○ ● ○ ● ● ● ● ● ● ● ● ● ● ● ● ●	DefaultSession		
Cource file Cource				
Build ( Debug ) Find in Files 1 ) Find in Files 2 ) Macro Ready	/ Lest / Version Control / 原 図 図	函 [Default1 desktop		INS NUM
🛃 スタート 🐚 R0E00008AKCE00SR. 🚺 受信トレイ -	Micro f 🔄 文書 1 - Microsoft W 🏾 🦉 無題 - ペイント	opproject_010 - High-p	📄 🖷 🌏 A般	😫 🥔 🧷 😨 😰 🔆 🕏 💐 🔤 💽 🏀 1513 -
	1			
	🙆 project 010 – High-p	erformance	5	
	ファイル(E) 編集(E) 表示(V)	) プロジェクト(P)	)	ワークスペース名
	□ ☞ ■ ❹   ቆ   ₺	<b>b C</b> ()		
				プロジェクト名
	□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □			
	E G source file			





PC上のフォルダ構成

新規プロジェクト「project 010」の生成が完了するとパソコンの「WorkSpace」フォル ダの中に「養成講座」というワークスペースが、その下部に「project\_010」というプロジ ェクトが生成されます。

# 2.5 ワークスペースとプロジェクト

High-performance Embedded Workshopでは、ワークスペースとプロジェクトという概念 があります。

・ワークスペース

High-performance Embedded Workshopを用いてプログラムを作成する場合の、一番大き な管理単位です。ワークスペースには、最低1つのプロジェクトが必要で、この1つのプロ ジェクトは、ワークスペース作成時に自動的に作成されます。また、ワークスペースは複 数のプロジェクトを持つことができます。

#### ・プロジェクト

プログラムを作成する場合、特定機能をライブラリ化してモジュールを階層化する場合 があります。このような場合、ライブラリ用のプロジェクトを作成、追加することができ ます。



# 2.6 コーディング、ビルド(ロードモジュールの生成)

図 4 にHEWのメインウィンドウを示します。メインウィンドウには3つのウィンドウ があります。それぞれワークスペースウィンドウ、エディタウィンドウ、アウトプットウ ィンドウと称します。このメインウィンドウでコード生成、ビルド(コンパイル/アセン ブル/リンク)を行います。



図4 HEWのメインウィンドウ

# 【実習2】スイッチ1(SW1)の操作でLED1 が点灯/消灯するプログラムを作成しましょう。

ワークスペースウィンドウでソースコードの生成・ビルドを行い、ダウンロードモジュ ールを作成します。

①LED1:ポート8の第0ビット(P80)に出力ポートとして割り付けます。
 ②SW1:ポート1の第0ビット(P10)に入力ポートとして割り付けます。

ポートを入力として使用するか、出力として使用するかはポートコントロールレジス タ(PCR)で指定します。ポートコントロールレジスタは、読み書き専用ポート以 外は各ポートごとに用意されており、1で出力ポート、0で入力ポートに切り替わり ます。使用に先立って入力/出力の設定が必要です。一般には「hwsetup.c」ファイル の中で設定します。

#### ポートコントロールレジスタ8 (PCR8)

PCR8 はポート8 の汎用入出力ポートとして使用する端子の入出力をビットごとに選択します。

ビット	ビット名	初期値	R/W	説 明
7	PCR87	0	w	汎用入出力ポートの機能が選択されているとき、このビットを1にセットす
6	PCR86	0	w	ると対応する端子は出力ポートとなり、0にクリアすると入力ポートとなりま
5	PCR85	0	w	す。
4	PCR84	o	w	P80(LED1)は出力ポートなので0を1に書き換える。
3	PCR83	0	w	コーディング例
2	PCR82	0	w	IO DOD = 0.01
1	PCR81	0	w	10.PCR8 – 0x01,
0	PCR80	0	w	

#### ポートコントロールレジスタ1(PCR1)

PCR1 はポート 1 の汎用入出力ポートとして使用する端子の入出力をビットごとに 選択します。

ビット	ビット名	初期値	R/W	説明
7	PCR17	0	w	PMR1 により汎用入出力ポートの機能が選択されているとき、このビットを 1
6	PCR16	0	w	にセットすると対応する端子は出力ポートとなり、0にクリアすると入力ポー
5	PCR15	0	w	トとなります。
4	PCR14	o	w	ビット3はリザーブビットです。
3	_	_	_	
2	PCR12	o	w	<b>P10(SW1)</b> は入力ポートなので0を0に書き換える。
1	PCR11	0	w	<u>コーディング例</u>
0	PCR10	0	W	IO.PCR1 = 0x00;

# ポートデータレジスタ8 (PDR8)

PDR8 はポート8 の汎用入出力ポートデータレジスタです。

ビット	ビット名	初期値	R/W	説明
7	P87	0	R/W	汎用出力ポートの出力値を格納します。
6	P86	0	R/W	このレジスタをリードすると、PCR8 がセットされているビットはこのレジス
5	P85	0	R/W	タの値が読み出されます。PCR8 がクリアされているビットはこのレジスタの
4	P84	0	R/W	値にかかわらず端子の状態が読み出されます。
3	P83	0	R/W	LED1(P80)が点灯のとき0 (Low)、消灯のとき1
2	P82	0	R/W	
1	P81	0	R/W	(High)を書きより。
0	P80	0	R/W	コーディング例
		•		IO.PDR8.BIT.B0 = 0; /* LED1 点灯 */

# ポートデータレジスタ1(PDR1)

PDR1 はポート1 の汎用入出力ポートデータレジスタです。

ビット	ビット名	初期値	R/W	説明
7	P17	0	R/W	PDR1 はポート 1 の出力値を格納するレジスタです。
6	P16	0	R/W	このレジスタをリードすると、PCR1 がセットされているビットはこのレジス
5	P15	o	R/W	タの値が読み出されます。PCR1 がクリアされているビットはこのレジスタの
4	P14	o	R/W	値にかかわらず端子の状態が読み出されます。
3	—	1	_	ビット3はリザーブビットです。リードすると常に1が読み出されます。
2	P12	o	R/W	
1	P11	0	R/W	
0	P10	_0	R/W	
			$\overline{}$	

SW1を押すとP10には0 (Low)、離すと1 (High)が入力されます。 P10のc言語での記述: IO.PDR1.BIT.B0

#### ■ 手順

- ① hwsetup.c内にLED1のP80を出力ポートに設定するコードを書きます。
- ② hwsetup.c内にSW1のP10を入力ポートに設定するコードを書きます。



③ project\_010.c 内の関数 main()内にSW1がON(押す)ならLED1点灯、SW1がOFF(離す)ならLED1消灯するように動作するプログラムコードを書きます。これらのコードが繰り返し実行されるように繰り返し文で構成します。



④「全てビルド」を実行します。

⑤正常に終了するとアウトプットウィンドウに下記メッセージが表示されます。 ⑥エラーが発生したらエラー発生要因(コーディングミス等)を取り除いてから、再度 ④に戻って実行します。



#### ◆ 参考:ビルドについて

ビルド処理の一般的な流れを以下の図に示します。ビルドの各フェーズにおいて、1 セットのプロジェクトファイルについてビルド処理を行います。それが完了すると、次 のフェーズに移ります。

上記の例では、第一のフェーズがコンパイラ、第二のフェーズがアセンブラ、そして 最後のフェーズがリンケージエディタです。コンパイラのフェーズでは、プロジェクト のC/C++ソースファイルを順次コンパイルします。アセンブラのフェーズでは、アセンブ リ言語のソースファイルを順次アセンブルします。リンケージエディタのフェーズでは、 すべてのライブラリファイルと、コンパイラフェーズとアセンブラフェーズからの出力 ファイルをリンクして、ロードモジュールを作成します。



#### 2.7 E8a を使用したデバッグの準備

### ① ケーブル接続

E8a 本体をパソコンとマイコンボードに接続します。パソコンとE8a本体間は付属のUSBケーブルで、E8a本体とマイコンボード間は付属の14ピンユーザーインタフェースケーブルで接続します。



パソコンとE8a本体をUSBで接続したとき、E8aドライバをインストールするため に下記の画面が表示されることがあります。画面の手順とおりE8a付属のCD-ROM を挿入して「次へ(N)」をクリックします。

新しいハードウェアの検出ウィザード				
	新しいハードウェアの検索ウィザードの開始			
	このウィザードでは、 次のハードウェアに必要なソフトウェアをインストールします: E8a			
	🥹 ハードウェアに付属のインストール CD またはフロッピー ディ スクがある場合は、挿入してください。			
	インストール方法を選んでください。			
	○ソフトウェアを自動的にインストールする(推奨)① ○ 一覧または特定の場所からインストールする (詳細)(S)			
	続行するには、D欠へ] をクリックしてください。			
	〈 戻る(8) 【 次へ(10) > 】 ( キャンセル			

ドライバのインストールが終了すると下記画面が表示されます。



# ② エミュレータとの接続

デバッグセッションを「Debug」から「SessionH8\_Tiny\_Super\_Low\_Power\_E8」に切り替 えると「エミュレータ設定」画面が表示されます



デバッグセッションの切り替え

「エミュレータ設定」画面ではデバイスが「H8/3694F」であることを確認のうえ下記の モードを選択し「OK」をクリックします。



#### ♦ 参考

#### エミュレータ起動モードの説明

#### 【フラッシュメモリデータを消去して起動】

本モードは、ターゲットデバイスのフラッシュメモリ内にE8エミュレータ用プロ グラムが存在しない場合に使用します。E8aをはじめて起動するときや、E8aエ ミュレータソフトウェアバージョンアップの際、またIDコードを変更したい場合、 こちらを選んでください。

#### 【フラッシュメモリデータを保持して起動】

本モードは、ターゲットデバイスのフラッシュメモリ内にE8エミュレータ用プログ ラムが存在している場合に使用します。入力するIDコードは上記で設定したコード を入力してください。間違ったコードを入力すると、フラッシュメモリ上のプログラ ムを全て消去します。

#### 【フラッシュメモリデータの書込みのみ】

本モードは、フラッシュメモリのライタとしてE8aエミュレータを使用します。プ ログラムのデバッグはできません。

ダウンロードするロードモジュールをワークスペースに登録し、ダウンロードしてく ださい。 「電源」画面が表示されます。ACアダプタから電源を供給しますので、チェックをいれないまま「OK」を選択します。

電源	? 🛛
🗆 エミュレータから電源	供給( <u>P)(最大 300 m A)</u>
■ 電圧選択 <b>③</b> <u>3</u> .3 V	<u>с 5</u> 0 л
ОК	キャンセル
	<ul> <li>◆ 参考</li> <li>マイコンボードの電源供給を</li> </ul>
	をACアダプタではなくエミ ュレータ(パソコンのUSB)
	から供給する場合は、チェック を入れます。

「E8AH8」画面が表示されます。



マイコンボードのDCジャック(J11)にACアダプタが挿入されていることを確認のうえ、 MCU回路のパワースイッチ(SW11)、基本I/O回路のパワースイッチ(SW12) をONにして Power LED が点灯したことを確認してから「OK」をクリックします。 「システムクロック」画面で「OK」をクリックします。

システムクロック	? 🗙
システムクロック( <u>S</u> )	L. C.
10.00	MHz
ОК	キャンセル

「IDコード」画面で「OK」をクリックします。

IDコード	? 🛛
ID⊐ードΦ	
DESA	
ОК	キャンセル

マイコンボードとE8 aの接続が成功するとアウトプットウィンドウに「Connected」が表示されます。



# ③ 接続完了時のメインウィンドウ

マイコンボードとE8aの接続が完了するとアウトプットウィンドウに「connect」の表示とともに下記のようにターゲットと接続完了したときにのみ表示されるデバッグツールバーやデバッグランツールバーと「Download module」フォルダが見えるようになります。



主なツールバーの説明



# ④ ダウンロード

ツールバーの「デバッグ」から下記のようにダウンロード対象モジュール名 (project\_010.abs)をクリックします。ロードモジュールがMCUへダウンロードされま す。



ダウンロードが完了するとエディタウィンドウの「ソースアドレス」欄にアドレスが見えます。





#### 2.8 プログラムの実行

#### ① CPUの初期化

デバッグランツールバーの「CPUリセット」をクリックすると、プログラムカンタ(PC)は初期化されます。すなわちプログラムカウンタは実行するプログラムの先頭アドレスを示します。エディタウィンドウに黄色のバーが表示されますが、この黄色のバーが示すアドレスの命令コードが次回実行される命令コードになります。

#### ② プログラムの実行

デバッグランツールバーの「実行」をクリックすると、プログラムは実行されます。

# ③ プログラム停止

プログラム実行中にだけ表示されるデバッグランツールバーの赤い「STOP」をクリック するとプログラムは停止します。プログラムが停止すると、停止した命令コードの次回実 行アドレスにある命令コードを黄色のバーが表示します。

# 【実習3】実際にロードモジュールをダウンロードしてプログラムを実行し、プログラム 動作を確認しましょう。

実習2で作成したロードモジュール「project\_010. abs」をダウンロードしてプログラム を実行します。

#### ■ 手順

① マイコンボード上で回路を作る。



- ② 「project\_010. abs」をダウンロードします。  $\Rightarrow$  「8.4 ④ダウンロード」参照
- ③ エディタウィンドウの「ソースアドレス」欄にアドレスが表示されていることを 確認します。
- ④ [標準]ツールバーの「CPUリセット」をクリックします。0x0400番地のソース コードに黄色のバーが表示されます。

🛷 resetpre.c		
行番 ソースアドレス	E., S., ソース	
44 45 46 47 48 43 50 51 52 53 53 54 55 54	<pre>//#ifdefcplusplus //extern "C" { //#endif //extern void _CALL_INIT(void); //extern void _CALL_END(void); //#ifdefcplusplus //} //#endif #pragma section ResetPRG</pre>	// Remove the comment when you use globa // Sections C\$INIT and C\$END will be gen
57 58 0404 59 0406 60 62 63 64 65 66 66 67 88	<pre>set_imask_corr((_UBYTE)1);     _INITSCT();  // _CALL_INIT();  // _INIT_IOLIB();  // errno=0;  // srand((_UINT)1);  // _s1ptr=NULL;</pre>	// Remove the comment when you use s // Enable I/O in the application(bot // Remove the comment when you use e // Remove the comment when you u // Remove the comment when you use s

- ⑤ [標準ツールバー]の「実行」をクリックします。プログラムが実行されます。RU
   N状態になります。[標準ツールバー]の「STOP」が赤で表示されます。
- ⑥ RUN状態でSW1の操作に対するLED1の点灯/消灯状態を確認します。
  - ・SW1を押したとき、LED1は点灯しますか?
  - ・SW1を離したとき、LED1は消灯しますか?
  - ・リセットスタート後のLED1の初期表示はどうなっていますか?

【課題2】CPUリセット後の「実行」でLED2が点灯、SW2を押すとLED2は消灯、SW2を離すとLED2は点灯プログラムを作成してデバッグして動作を確認しましょう。

# 3

シリアルインターフェース



#### 3.1 概要

シリアルインターフェースとはHigh(1)/Low(0)の信号を1ビットずつ送信または受信 する通信方式です。通信を行うためには同期を取らなければなりませんが、この同期を取 る方式の違いによって同期式と調歩式(非同期式)に分類できます。

同期式は、同期対象をクロックとします。クロックの立ち上がり、立下りタイミングでデ ータの送信・受信をおこないます。全二重通信を想定した場合、信号線はデータ線2本と クロック線1本の合計3本が必要です。1バイトデータの送信・受信には連続した8個の クロックで行います。クロックに同期するという意味でクロック同期式通信と呼ばれます。 クロック同期式通信の一般的なフォーマットを図1に示す。



図1 クロック同期式通信

特徴

・3線(送信線、受信線、クロック線)

・調歩同期式通信方式に比べて、スタートビット、ストップビットがない分だけ伝送効率 がよい。 調歩式(非同期式)は「歩みを調える」方式ですから、同期対象をクロックに求めずそれ ぞれのデータの直前に1ビット分のスペース(スタートビット)を付加して、このスペー スで通信開始タイミング計り、事前に決めた通信速度でスペースに続くデータをサンプリ ングします。通信速度で決定される1ビット長を厳守する必要があります。1文字(7ビ ットまたは8ビット)の最後に1~2ビットのマーク(ストップビット)を入れます。クロ ックに同期しないという意味で非同期式通信また調歩同期式通信(Universal Asynchronus Receiver/Transmitter (UART))と呼ばれます。

#### 特徴

- ·2線式(送信線、受信線)
- クロック線が不要
- ・ 同期式に比べると、スタート、ストップのタイミングビット分だけ伝送効率は悪い。

調歩同期式通信を実現するためには、必ず下記プロトコルを決定しておく必要があります。

必須通信プロトコル

- ・通信速度 (ボーレート): 4800bps, 9600bps, 384000bps など
- ・データ長 :7bit/8bit
- ・パリティビット (1bit): 偶数パリティ/奇数パリティ/なし
- ・ストップビット : 1bit/2bit
- ・フロー制御 : Xon/Xoff 制御/なし

調歩同期式通信の通信データの一般的なフォーマットを図2に示す。



図2 調歩同期式通信の通信フォーマット

「10. 距離データの応用」では、FPGAボードとマイコンボード間の距離データの送受 信にこの調歩同期式通信を使用します。

# ◆参考

調歩同期式通信と RS-232-C は同義と捕らえがちですが、それは間違いです。 RS-232-C はOSI参照モデルの物理層にあたる部分の仕様です。コネクタの形状、 Low(0)/High(1)の電気的表現方法、信号の論理的特性などが決まっています。 現在では多くのパソコンで標準の入出力インターフェースとして RS-232-C を採用し ており、さらに通信方式として調歩同期式通信を使っているため、「調歩同期式通信 = RS-232-C」と誤解されているようです。

# 4 距離データの応用



#### 4.1 概要

FPGAで計測した距離データをマイコン側で受信して、当該距離に応じた警報を発するシステムを構築しましょう。

#### ■ システムブロック図

図1にシステムブロック図を示します。FPGA ボードとマイコンボードの通信距離が長い場合はFPGA 基板のRS-232-Cコネクタ(J3)からRS-232-Cケーブルを使用してマイコンボードと通信を行うという方法もありますが、今回は近接したボード同士での通信なので、RS-232-Cケーブルを介さずに、FPGA ボードの送信ポート(P70)とマイコンボードのRxD 端子(P21)を直接接続して通信を行います。



図1 システムブロック図

# ■ システムフューチャー

①通信:調歩同期式 UART (FPGA からマイコンへ一方向通信)
 ②マンマシンインタフェース:

距離データに応じて

- ・警報(圧電ブザー)を鳴動
- ・LED を点滅
- ・7セgメントLED1桁に距離を表示

#### 4.2 ソフトウェア詳細仕様

#### ■ FPGAとマイコンボード間の通信

FPGAとマイコンボード間は図2に示す 6 バイトのデータを調歩同期式通信方式でデ ータの受け渡しを行います。

BYTE	1	2	3	4	5	6
	SOD	1000	100	10	1	EOD
	StartOfData	mm	mm	mm	mm	EndOfData
	0x42(B)	•••	0x00~0;	x09 × 4	•••	0x45(E)
	BinaryData					
	0x44(D)		0x30~0;	x39 × 4		0x45(E)
	DecimalData					

図2 FPGA・マイコンボード間データフレーム

通信の例:42.8cmのとき

Binaary Data :0x420x000x040x020x080x45Decimal Data :0x300x300x340x320x380x45

# ■ 通信プロトコル

- Baud Rate : 38400bps
- Data Bits : 8
- Stop Bit : 1
- Parity Bit : none
- Xon/Xoff : none

#### ■ 制御周期タイミングチャート

設定する t<sub>1</sub>のタイミングチャートを図3に示します。



t<sub>1</sub>を表1に示します。

入力值範囲(cm)	入力値<10	10≦入力値<20	20≦入力値<30	))	90≦入力値<100	100≦入力値
t <sub>1</sub> 設定値(ms)	0	100	200	$\langle \langle$	900	1000

#### 表1 t<sub>1</sub>設定値

#### ■ 7 segment LED

・距離単位(cmとm)を区別するためにセンチメートル表記のときは dp を点灯、メートル表記のときは dp を消灯します。

・入力値と7seg. LED に表示する数字の関係を表2に示します。

入力值範囲(cm)	入力値<10	10≦入力値<20	/	90≦入力値<100	100≦入力値<200	)	\ 800≦入力値<900	900≦入力値
7seg. LED表示	8		$\langle \langle$	8.			8	8

表2 入力値と7seg. LED に表示する数字の関係

#### ■ 圧電ブザー

t<sub>1</sub>の周期で圧電ブザーが鳴動します。

# ■ LED

t<sub>1</sub>の周期でLEDが点滅します。

# 4.3 マイコンのピンアサイン(ピン配置)

表3にFPGAからの距離データを受信して、LED,ブザー等に表示するためのピン アサイン表を示します。

端	子	定	義	表
---	---	---	---	---

ターゲット:H8/3694F

Pin No	5	25	1/0	Reset時 の状 <del>え</del>	<b>右</b> 子記号	端子名称	1/0	機能	INZ時の 状態	備考
1	A	VCC	Ι	н						
2		X2	0							
3		X1	1							
4	۱ ۱	/CL	1	н						
5	/	RES	1	L		/RES	Ι	リセット	Н	
6	Т	EST	1	L						
7	)	/ss	1	L						
8	0	SC2	0							
9	0	SC1	1							
10	۱	/cc	1	н						
11	P50	/WKP0	1/0	X	P50	汎用入出力ポート	0	7 seg. LED [a] 対応	L	
12	P51	/WKP1	1/0	X	P51	汎用入出力ポート	0	7 seg. LED[b] 対応	L	
13	P52	/WKP2	1/0	X	P52	汎用入出力ボート	0	7seg.LED[c]対応	L	
14	P53	/WKP3	1/0	X	P53	汎用入出力ポート	0	7seg. LED[d]対応	L	
15	P54	/WKP4	1/0	X	P54	汎用入出力ポート	0	7seg, LED[e] 対広	L	
16	P55	/WKP5	1/0	×	P55	汎用入出力ポート	0	7seg. LED「引対応	L	
17	P10	TMOW	1/0	X						
18	F	211	1/0	X						
19	F	12	1/0	X						
20	P56	SDA	1/0	Ŷ	P56	汎用入出力ポート	0	7see LED[e]耕広		
21	P57	SOL	1/0	Ŷ	P57	汎用入出力ポート		7 see LED do b tr		
22	P74	TMRLV	1/0	Ŷ	1.01			7 See. CCD1 dp1 X1MS	-	
23	P75	TMOLV	1/0	⊢ ≎ –			+			
2.0	P76	TMOV	1/0	÷	TMOVA	カイフソ史市標子	0	ゴモーモーモニ	L	
25	170	NMI	100	<u> </u>	THOU			- ノリー山ノ」市(14)		FS接続体田
26	 	FTO	1/0	×	Pag	NH IN Y-H				CO199 #/Life /H1
27	P81	ETICA	1/0	- ÷	100		Ľ			
28	P82	FTIOR	1/0	⊢ ≎ –	$\vdash$		<u>``</u>			
20	P92	FTIOC	1/0	⊢ ≎ –	· · · · ·		rm	8 の 27 番ピン		
20	D04	FTIOD	1/0	- ÷						
21	F04	PHOD	1/0	÷		$\Box  \diagdown$				口袋的休用
20		00	1/0	- ÷		Term8 $O$ 28	悉と	~~~		この接続体用
32		007	1/0	<u></u>						CO按勒(使用
33		07 0000	1/0	X			-			□□按範便用
34	P20		1/0	X	D-D			いた母に招き		
30	P21	TYD	1/0	<u>×</u>	TOD 1			uar (3216 9m T		
27	P14	/IPO0	1/0	⊢ ÷ −			-		+	
20	D16	/IP01	1/0	⊢ ÷ −			+		+	
38	P10	/IRQ1	1/0	<u>×</u>			$\leftarrow$			
39	P10	ADCA	1/0	<u> </u>		`		<u> </u>		
40	P17	/TRQ3 /TRGV	1/0	×						
41	PB4	AN4		X				$\vdash$		
42	PB5	AN5		X					I	<u> </u>
43	PB6	AN6		X				Term8	の8番と	シレー
44	PB7	AN7		X				1011110		
45	PB3	AN3		X						
46	PB2	AN2		X						
47	PB1	AN1	1	X						
48	PB0	ANO	1	X						
Ж H:H	ish L	:Low >	C: 77-	イインビーク	<b>シ</b> ス					

表3 マイコンポートのピンアサイン表

# 4. 4 システム構成上の問題点

FPGA ボードの TxD 端子の信号レベルは High のとき 3.3vです。マイコンボードは 5.0v で動作しているため、FPGA ボードの TxD 端子とマイコンボードの RxD 端子を直接接続する とマイコンボード側では最低でも 3.5v (Vcc×0.7 = 3.5v) 以上ないと High レベルと認識 しません (表4参照)。

項目	記号	適用端子	測定条件		規格値		単位	備考
				Min	Тур	Max		
入力 High	Ин	RES, NMI	Vcc=4.0~5.5V	Vcc×0.8	_	Vcc+0.3	V	
レベル電圧		WKP0~WKP5						
		IRQ0~IRQ3						
		ADTRG						
		TMRIV, TMCIV						
		FTCI, FTIOA		Vcc×0.9		Vcc+0.3	V	
		FTIOB, FTIOC						
		FTIOD						
		SCK3						
		TRGV						
		RXD, SCL, SDA	Vcc=4.0~5.5V	Vcc×0.7	—	Vcc+0.3	V	
		P10~P12						
		P14~P17						
		P20~P22						
		P50~P57		Vcc×0.8	—	Vcc+0.3	V	
		P74~P76						
		P80~P87						
		PB0~PB7	AVcc=4.0~5.5V	AVcc×0.7	—	AVcc+0.3	V	

表4 H8/3694FのDC特性

図4からマイコンボード側では、1.0vから3.5v間は「不定」であることがわかります。FPGA の送信データの3.3v(Highレベル)を何とかして5v程度までレベル変換する必要があり ます。



図4 FPGAの出力信号レベルとマイコンボードの入力信号レベル

■ 対策

FPGA ボードの TxD 端子の 3.3 v を 5.0v 信号に変換するために、電圧レベル変換回路を介してデータの送受信を行います。



図5 電圧レベル変換

今回の電圧レベル変換回路には、汎用ロジック IC「74HC07」を使用します。FPGA ボードの TxD 端子には 0/1 のデータとして 0.0v/3.3v の電圧レベルの信号が出力されます。

74HC07 の出力は、図 10.6 の Truth Table (真理値表) および System Diagram から Nch オープンドレイン回路になっていることがわります。したがって出力端子にはプルアップ 抵抗が必要です。3.3v を 5.0v にレベル変換しますのでプルアップ抵抗は 5 v に吊ります。

74HC07のVccをいくらにするかを検討します。マイコンボードの14ピンソケット(IC2) は、14番ピンは5v、7番ピンがGNDにつながっています。表10.6から、Vccが4.5vの とき入力信号レベルが3.15v以上ないとHighレベルとは認識しないことがわかります。こ のことから Vcc を 5 v にした場合は入力信号レベルは 3.5 v 以上程度は必要と推測できます。 74HC07 の Vcc を 5 v にした場合は、FPGA の High のレベル 3.3 v を High と認識できない 可能性があります。このような理由からここでは 74HC07 の Vcc を F P G A ボードから供 給するとして 3.3 v にします。汎用ロジック IC 74HC07 の電源(74HC07 の 14 番ピン)に は 3.3 v を印加します。





Vcc が 4.5 v のとき入力信号レベルが 3.15 v 以上な いと High レベルとは認識しないことがわかる。

Characteristics	Symbol	n l			٦	Ta = 25°C			Ta = -40 to 85°C		
Characteristics	Symbol		$\langle \rangle$		V <sub>CC</sub> (V)	Min	Тур.	Max	Min	Max	Unit
				~	2.0	1.50	_	_	1.50	-	
High-level input voltage	VIH		_		4.5	3.15	—	—	3.15	—	V
					6.0	4.20	—	—	4.20	—	
Low-level input voltage					2.0	—	_	0.50	—	0.50	
	VIL	—			4.5	—	—	1.35	—	1.35	V
					6.0	—	_	1.80	—	1.80	
	V <sub>OL</sub>		I <sub>OL</sub> = 20 μA		2.0	_	0.0	0.1	_	0.1	
					4.5	_	0.0	0.1	_	0.1	
Low-level output voltage		VIN = VII			6.0	—	0.0	0.1	—	0.1	V
-			I <sub>OL</sub> = 4 mA		4.5	_	0.17	0.26	- 0.33		
			I <sub>OL</sub> = 5.2 mA		6.0	_	0.18	0.26	_	0.33	
Output off-state	lez	VIN = VIH	l or VIL		6.0		_	+0.5		+5.0	Δ
current	102	Vout = \	/cc		0.0			10.5		10.0	μα
Input leakage current	IIN	VIN = V <sub>CC</sub> or GND			6.0	_	_	±0.1	_	±1.0	μА
Quiescent supply current	Icc	VIN = VC	<sub>C</sub> or GND		6.0	_	_	1.0	_	10.0	μА

表6 74HC07のDC特性の一例

#### 【実習4】 電圧レベル変換回路の設計

マイコンボードの基本 I/O 部の 14pin I Cソケットには 74HC07 が実装されています。 0v/3.3v を入力すると出力側で 0v/5.0v にレベル変換される回路を 74HC07 を使って設計 しましょう。

#### ■ 手順

- 回路設計
  - ・74HC07の14番ピン(Vcc)を3.3Vに接続します。

⇒ 74HC07 を I C ソケットから取り出し、74HC07 の 14番ピンだけを横に伸ばす ようにして、74HC07 を再度 IC ソケットに挿入したときに 74HC07 の 14番ピンが マイコンボードの IC ソケットに接触しないような配慮・工夫が必要です。

⇒ FPGA ボードに 3.3v があるので、74HC07 の 14番ピン (Vcc) にはそこから Vcc を供給するようにします。

・74HC07の出力端子(2番ピン)は Nch オープンドレインですから、4.7 k  $\Omega$ 程度のプルアップ抵抗で 5v 電源に吊ります。

⇒ マイコンボードに 5.0 v があります。

・マイコンボード、FPGAボード、汎用ロジック IC 74HC07 のそれぞれのGNDを接続します。



図7 74HC07による電圧レベル変換回路

② 動作確認

・74HC07 の1番ピン(入力)に 3.3 vを入力したときの2番ピン(出力)には何ボルトが出力されますか。  $\Rightarrow$  5.0 vになってますか。

・74HC07 の1番ピン(入力)に 0v を入力したときの2番ピン(出力)には何ボルトが出力されますか。  $\Rightarrow$  0.0v になってますか。

#### 4.5 距離計システムを動作させる

#### ■ システム全体の回路

電圧レベル変換回路を挿入したシステム全体の結線図を下記に示します。



図8 距離計測システム結線図

#### ■ マイコンプログラムの実装

#### ① マイコン側プログラムの基本動作およびプログラム構成

FPGAから1バイト受信するごとにシリアル通信割り込み処理が動作し、1バイト単位 に受信データを解析します。SODからEODまでのデータが揃ったところで割り込み処
理で作成した距離データをメイン関数(main())で取り込み、距離データに応じたマンマ シンインターフェース出力(ブザー、LED、7segment LED)します。

これを実現するために、マイコン側プログラムではFPGAからのデータをベクタ番号 23 の SCI3 割り込み処理で1バイト単位に受信します。EODまで正常に受信したとき、「距 離データ更新要求フラグ」をセットします。メインプログラム中ではこの「距離データ更 新要求フラグ」をポーリングしており、フラッグがセットされているときに最新の距離デ ータに更新します。メインプログラムでは最新距離データに応じたブザー、LED等の点 滅等を実施します。

図9にメイン関数(main())のフローチャート、図10にシリアル通信割り込み処理 (INT\_SCI())のフローチャートを示します。





図10 シリアル通信割り込み処理のフローチャート

### ② 新規プロジェクトの挿入

いままで使用してきた「養成講座」というワークスペースのなかのに「contact\_senser」 という名称のプロジェクトを新たに挿入して、そのプロジェクトの中に完成済みのソース ファイル(rx\_main.cとintprg.c)を追加するという手順でプログラムを作成します。

ふたつの完成済みソースコードファイルは、次のとおりです。

・rx\_main.c:新規プロジェクト挿入時に自動生成される「contact\_senser.c」の代わ りとなるソースコードファイル。ポートの初期設定等は一般には hwsetup.c のなかで 実施しますが、今回は rx\_main.c のなかで実施しています。したがって hwsetup.c フ ァイルは追加しません。プロジェクト「contact\_senser」生成時に生成された hwsetup.c をそのまま使います。

・int\_prg.c: FPGA からのデータを受信する割り込みルーチンを組み込んだ割り込み 処理プログラムのソースコードファイル

# ③ 「contact\_senser」プロジェクトの作成

「養成講座」ワークスペースの中には、すでに実習2で作成したプロジェクト「project\_010」があります。この「養成講座」ワークスペースの中に新規に「contact\_senser」という名称のプロジェクトを作成します。手順を示します。



プロジェクトの挿入	? 🛛
挿入 <ul> <li>         ・         新規プロジェクト(N)         ・         ・         既存プロジェクト(E):         ・         ・         ・</li></ul>	OK キャンセル
参照(B)	

新規プロジェクトの挿入		? 🔀
プロジェクト クロジェクトタイプ Application Assembly Application Demonstration Empty Application Empty Application Import Makefile Library Debugger only - H8 Tiny/Super L Debugger only - H8S/Tiny E8a E	ワークスペース名(W): 養成講座 プロジェクト名(P): contact_senser ディレクトリ(D): C:¥WorkSpace¥養成講座¥contact_senser CPU種別(C): H8S,H8/300 ▼ ツールチェイン(T): Renesas H8S,H8/300 Standard ▼	参照( <u>B</u> )
	ОК	キャンセル

プロジェクト名に「contact\_senser」と入力して OK」をクリックすると「8.2 新規プ ロジェクトの作成」で説明した一連の画面が表示されます。同じ手順で新規プロジェクト 「contact\_senser」生成時の初期画面が下記のように表示されます。



ワークスペースウィンドウに「contact\_senser」プロジェクトが生成されたことを確認し て「はい」をクリックします。

# ③ rx\_main.cとintprg.cの「contact\_senser」フォルダへのコピー

添付 CD-ROM 中の「距離計システムソースコード」からファイル rx\_main.c と intprg.c を WorkSpace>養成講座>contact\_senser フォルダの中へコピーします。



ワークスペースウィンドウに表示されたプロジェクト「contact\_senser」のファイルは HEWが自動生成したものです。⑤でファイルコピーした「rx\_main.c」はワークスペース ウィンドウには見えません。「intprg.c」は⑤でコピーしたものです。「intprg.c」のソー スコードを開くと「vector 23 SCI3」の割り込み処理のところに FPGA からのデータを受信 するためのコードが見えます。プロジェクト「contact\_senser」を生成したときに自動生 成されたものではなく、⑤でコピーしたものであることが分かります。

E Contact senser	🛷 intpre.c	
E G source file	行乗 5 パーフ	
Contact senser (		
T dboot o	58 // vector 18 WKP 59 // teterovet(uset=10) used INT WKD(used) [/w stars(), w/]	
	53 Interrupt(vect-18) void INT_WKP(void) {/* sieep(); */} 50 77 vector 19 Timer & Overflow	
i nwsetup.c	61 interrupt(vect=19) void INT TimerA(void) {/* sleep(): */}	
intpre.c	62 77 vector 20 Reserved	
i resetprg.c	63	
🖃 🔄 Dependencies	64 // vector 21 Timer W	
iodefine.h	65interrupt(vect=21) void INT_TimerW(void) {/* sleep(); */}	
🖃 stacksct.h	66 // vector 22 limer V 87 // interpret(upst-22) upid INT TimerV(upid) //w place(): w/b	
🔤 typedefine.h	67 Thterrupt(vect-22) void INT_Thmerv(void) (/* steep(), */)	
🔤 💽 project_010	69 interrupt(vect=23) void INT SCI3(void)	
	70 [	
	71 extern _UWORD g_input_buf;	
	72 extern _UBYTE g_input_update_req;	
		/m = = = = = = m/
	74 Extern _UDITE stage,	7● ステニン ●/ 7# 受信コュール
	76 extern IIWORD input cal:	/* 文信フォーム /* 入力データ計1
	77	in start start
	78	
		4. <u>.</u>
	80 If ((SU13.SSR.BYIE & U×38) == U){	/* オーハーラン、
	82 switch(SCI3_RDR){	
	83 case SOD BIN:	/* RDR = binarv(
	84 stage = 0;	/* yes ⇒ ステ <sup>、</sup>
	<pre></pre>	· · · · · · · · · -

# ⑤ 「rx\_main.c」のプロジェクトへの追加

「rx\_main.c」ファイルを当該プロジェクトに追加します。HEWの画面から「プロジェクト(P)」>「ファイルの追加(A)」を選択します。



左クリックすると「プロジェクトにファイルを追加」の画面が表示されます。 「rx\_main.c」を選択して「追加」をクリックします。

'contact_senser'	プロジェクトにファイル	を追加				? 🗙
ファイルの場所の:	Contact_senser		•	← 🗈	d 🏹	•
🛅 Debug		🗐 intprg.c				
🛅 Debug_H8_Tiny_	Super_Low_Power_E8	🗐 iodefine.h				
🛅 Release		🗐 resetprg.c				
🗐 contact_senser.	c	🗐 rx_main.c				
🗐 dbsct.c		🗐 stacksct.h				
🗐 hwsetup.c		🗐 typedefine.h				
ファイル名(N):	rx_main.c					追加
ファイルの種類(工):	Project Files			-	*	キャンセル
	☑ 相対パス(B)	□ 登録済みつ	ァイル	を非表示	( <u>P</u> )	

ワークスペースウィンドウに「rx\_main.c」が表示されます。



この状態でビルドを行うとメイン関数(main())が「contact\_senser.c」と「rx\_main.c」 の両ファイルに存在することになりコンパイルエラーになりますので「contact\_senser.c」 は削除します。

# ⑥ 「contact\_senser.c」のプロジェクトからの削除

「HEW」の画面から「プロジェクト (P)」>「ファイルの削除 (R)」を選択すると 「プロジェクトファイルの削除」画面が表示されます。



「contact\_senser.c」を選択し「削除(R)」をクリックします。

プロジェクトファイルの削除		? 🔀
プロジェクトファイル(P): contact_senser.c dbsct.c hwsetup.c intprg.c resetprg.c rx_main.c	C:¥WorkSpace¥義成講座 C:¥WorkSpace¥義成講座 C:¥WorkSpace¥義成講座 C:¥WorkSpace¥義成講座 C:¥WorkSpace¥義成講座 C:¥WorkSpace¥義成講座	OK キャンセル 削除(R) すべて削除( <u>A</u> )



「プロジェクトファイルの削除」画面から「contact\_senser.c」ファイルが消えます。

「OK」をクリックします。

ワークスペースウィンドウから「contact\_senser.c」が消えます。



以上で、「contact\_senser」プロジェクトに最終ソースコードのファイルが揃いました。「ビ ルド」を行い「no error」を確認後、マイコンへロードモジュールをダウンロードします。

# ■ 距離計システムの動作

ロードモジュールの書き込みが終了したら、マイコン側プログラムを実行します。FP GAボードの距離センサーの前に手のひらをかざして前後させてみましょう。「10.2 ソフ トウェア詳細仕様」に記述されているように、手のひらと距離センサーとの距離に応じて マイコンボード上のブザー鳴動間隔が変化、LEDの点滅周期が変化、7セグメントLED に距離に応じた数字が表示されます。

注意:受信側の処理について

FPGAボードには送信データ形式を切り替えるスイッチがあります。スイッチを押す ごとに、Binary Data 形式(SOD=0x42)と Decimal Data 形式(SOD=0x44)を切り替え ます。

マイコン側のプログラムは Binary Data (SOD =0x42) に限って受信した距離データを 変換するように設計されています。Binary Data (SOD=0x44) は無視されます。

# 【実習5】マイコンボード側の RxD 端子と FPGA の送信端子の波形を観測してみましょう。

■ 手順

オシロスコープでマイコン側受信データと FPGA 送信データを2 チャンネルで観測します。

⇒ FPGA 側では約 3.3v で出力しているがマイコンボード側の RxD 端子では約5 v に レベル変換されていますか。

⇒ SOD1バイト、データ4バイト、EOD1バイトの全体で6バイトのデータを送受信 していることがわかりますか?

⇒ 先頭の1バイトデータをスタートビットからストップビットまで1ビット単位にデ ータを確認しましょう。

⇒ FPGA 送信データ波形とマイコンボード側受信データ波形の波形の立ち上がり時間 は同じですか。



【実習6】74HCO7の出力端子(マイコンボードのRxD端子)のプルアップ抵抗値を替 えたときの波形を観測して、比較してみましょう。

### ■ 手順

- ① プルアップ抵抗4.7 k  $\Omega$ のとき、FPGA 送信端子とマイコンボード受信端子の2チャネルで観測します。
- ④ プルアップ抵抗15kΩのとき、FPGA送信端子とマイコンボード受信端子の2チャネ ルで観測します。
- ⑤ ①と②で観測した波形を比較しましょう。
   ⇒ プルアップ抵抗が 4.7 k Ω時の通信波形と 15 k Ω時の通信波形を観測すると、4. 7 k Ωよりも15 k Ωの方が、信号の立ち上がりが鈍っていることが分かります。



送信側の方形波(a.入力波形)は受信側では回路常数によって立ち上がりが多少鈍った波形になります(b.出力波形)。63%まで立ち上がる時間を時定数といいτで表します。 この時定数はコンデンサ容量と抵抗値で決まります(式(1))。



 $\tau = CR \cdot \cdot \cdot \cdot \cdot ( \vec{\mathfrak{T}} \mathbf{1} )$ 

 $\tau$ :時定数(単位:SEC) / C :コンデンサ容量(単位:F) / R:抵抗値(単位: $\Omega$ )

プルアップ抵抗を小さくすればτが小さくなることが分かります。プルアップ抵抗により 波形整形が可能です。

# **5** おわりに



本編を通して、自分がどの立ち位置で組込みソフトウェアエンジニアとして関わってい きたいのかを考えたとき、これから修得すべき技術が自分の足元におぼろげながら見えて きたようであれば講座の目的は達成できたと思います。

本編の発展的学習としては、次のようなことが考えられます。

1. F P G Aボードとマイコンボードを RS-232-C ケーブルで接続した通信形態としてみる。 2. 近接状況を通知するマンマシンインタフェースを自分なりに改良してソフトウェア仕 様書を作成、仕様にしたがった新たなプログラム開発を試みる。

3. FPGAボードからの受信において、SODコードが 0x44 でも受信できるように改修 してみる。

4. ソースコードをより見やすくする工夫、いわゆるコード書法について学ぶ。

さらに、専門的に技術要素ごとにより深く学ぶこと、すなわち次に記載するような事項を 修得することをお勧めします。

- 1. マイコンのハードウェア (マニュアルの精読)
- 2. 測定技術の向上
- 3. 開発環境のより詳細機能
- 4. 電子デバイス
- 5. 電子回路・デジタル回路
- 6. 開発言語
- $7. \ OS$
- 8. 設計論
- 9. コード書法
- 10. 規格

# 6 Appendix



# 表1 マイコンボード部品表

Na	回路部名称	部品番号	日本	型番 仕様	数量	レメーカ	備考
1	電源回路部	J11	DCジャック	21 ゆセンター+	1		
		swl1	MCU Power スイッチ	スライドスイッチ	1		
		sw12	I/O Power スイッチ	スライドスイッチ	1		
		LED11	MCUB Power LED	砲運形	1		MCU部 Power(Voot)供給確認用
		LED12	I/O部 Power LED	砲運形	1		I/O部 Power(Voc2)供給確認用
		R11	炭素皮膜抵抗	1.2K	1		
		F12	炭素皮膜抵抗	1.2K	1		
		C11	電解コンデンサ	25ν47μF	1		
		C12	電解コンデンサ	25ν 47 μ F	1		
2	リセット回路	D21	整流用ダイオード		1		
		C21	電解コンデンサ	35v10µF	1		
		F21	炭素皮膜抵抗	10K 1/8W	1		
		sv21	<u>タクト スイッチ</u>	小	1		
3		LED31	小型LED(赤色)	SLP-881 A	1	SANYO	
		LED32	小型LED(赤色)	SLP-881 A	1	SANYO	
		LED33	小型LED(赤色)	SLP-881 A	1	SANYO	
		LED34	小型LED(赤色)	SLP-881 A	1	SANYO	
		F31~F34	<u> 炭素皮膜抵抗</u>	3301/8W	4		
4	Volumeleitä	R41	炭素皮膜抵抗	1.2K 178W	1		
		VR41	半固定抵抗	20K	1		10k~20kΩ 柱度
		041	セラミックコンテンサ	01 µ F	1		
5	圧電フサー凹路	F61	灰素皮肤抵抗	1.2K 1/8W			
		F62	灰素皮肤透抗	4/K 1/8W	11		
		BZD1	住電/サー	PKM11-4A11(他奶买)	1		
0	基本ケート回路	102	ロンクテット レートニアー・クラント・アントサー	14P1N			
7		001	セフミックコンテンリ				
'	SW//JUBB	SW/1	メンキ スイツチ  カクト フィッチ				
		D71	13:21:24:27 歴史成時46位	71 77 / 1 /03 /			
		ID70	灰条灰肤抱机	471/ 1 /01/			
		1071	201番1X1米12511。  そうらう いくした いくうら 二人 しょうがせき	4/1、1/017  2-1			
			シャンハビノ(ショードビンド)の おみつい(ビン)(ショートレン)(オオ	opin ビンベンスを加用 Dain ビンヘッダが変用	I i		
•		D01 72	ビュンバビスショー・ビンドは		1		
Ů		su81	末 custu dioスイッチ				
9	Tr(FET)回路	FFT1	「近日」用フケット		<u> </u>		未実装(2/ケットのみ用意)
10	7ses. 08	7ser101	小型未色7set.LED	G 9A040G(アノードコモル)	1	SHARP	IE=20mA
<u> </u>		103	シアノトライバ(8回路)	TD62083APG	li	TOSHBA	
		R101~R106	炭素皮膜抵抗	270	8		
		0101	ヤラミックコンデンサ		Ĭ		
11	MCUEBB	ONI	ピンヘッダノケット	25pin(2列)	1		MB+H8A挿入コネクタ
		ON2	ピンヘッダノケット	25pin(2列)	1		MB+H8A挿入コネクタ
			マイコンボード	MBHBA	1	Sunhayato	H8/3694F搭載
12	B8接続回路	ON3	E8用コネクタ	H+UDI(14pin)	1		
		R1 21	集合抵抗	コモン 47K 9端子	1		ブルアップ用

◇ 参考・引用 文献 ◇

- 1. E8a エミュレータ ユーザーズマニュアル 別冊 (H8/300H Tiny シリーズ接続時の注 意事項)、(㈱ルネサスエレクトロニクス
- 2. E8a エミュレータ ユーザーズマニュアル、㈱ルネサスエレクトロニクス
- 3. H8/3694 グループ ハードウェアマニュアル、㈱ルネサスエレクトロニクス
- 4. High-performance Embedded Workshop V4.09 ユーザーズマニュアル、㈱ルネサスエ レクトロニクス
- 5. TD62083 データシート、㈱東芝
- 6. TC74HC07 データシート、(㈱東芝
- 7. 新プロトコルハンドブック、朝日新聞社
- 8. MB-H8A 取り扱い説明書、サンハヤト㈱

### 平成 24 年度文部科学省委託 「東日本大震災からの復興を担う専門人材育成支援事業」 東北の復興を担う自動車組込みエンジニア育成支援プロジェクト

#### ■推進協議会

C

)	佐藤 公一	東北電子専門学校 校長
	今野 幸信	東北電子専門学校 総務部 部長
	與那嶺 尚弘	仙台高等専門学校 知能エレクトロニクス工学科 准教授
	岩渕 喜悦	公益財団法人仙台市産業振興事業団
		地域産業振興部 新事業推進課 産学連携担当 ビジネス開発ディレクター
	伊藤俊	宮城県黒川高等学校 教頭
	木村 康弘	宮城県米谷工業高等学校 情報技術科 科長
	森 武彦	宮城県工業高等学校 校長
	白田 正樹	アベールジャパン仙台支店 システム開発センター
	渋谷 義博	トライポッドワークス株式会社
		技術本部 プロジェクトマネージメントグループ プロジェクトマネージャ
	羽曽部 恭美	カストマシステム株式会社 エンベデソドシステムソリューション事業部 事業部長
	三浦 卓	宮城県経済商工観光部 産業人材対策課 課長
	工藤拓	宫城県自動車産業振興室 技術支援班 主事
	今井 和彦	宮城県震災復興・企画部情報産業振興室/ 産業技術総合センター
		機械電子情報技術部 情報技術開発班 副主任研究員
	吉岡 正勝	有限会社ザ・ライスマウンド マーケティングマネージャー

#### ■開発分科会

■៣元刀竹去	
〕坂藤 健	東北電子専門学校 自動車組込みシステム科 学科主任
高橋 敬	東北電子専門学校 CAD設計製図科 学科主任
小野寺 敬司	花壇自動車大学校 教頭
白田 正樹	アベールジャパン仙台支店 システム開発センター
渡辺 登	株式会社アフレル エデュケーション・プランナー/事業企画室 室長
柴原 健次	エキスパートプロモーション 代表
吉岡 正勝	有限会社ザ・ライスマウンド マーケティングマネージャー

### ■講座運営分科会

■舑座連呂刀枰云	
〕坂藤 健	東北電子専門学校 自動車組込みシステム科 学科主任
小野寺 敬司	花壇自動車大学校 教頭
與那嶺 尚弘	仙台高等専門学校 知能エレクトロニクス工学科 准教授
岩渕 喜悦	公益財団法人仙台市産業振興事業団
	地域産業振興部 新事業推進課 産学連携担当 ビジネス開発ディレクター
伊藤 政光	株式会社エスワイシステム 執行役員 中部事業部 事業部長
吉岡 正勝	有限会社ザ・ライスマウンド マーケティングマネージャー

### ■事業実施協力専修学校・企業・団体等

)	古賀	稔邦	日本電子専門学校 校長
	石川	浩	日本工学院専門学校 テクノロジーカレッジ ロボット科
	岡田	靖志	浜松情報専門学校 教務課長
	村岡	好久	名古屋工学院専門学校 テクノロジー学部 部長
	村上	登昭	大阪工業技術専門学校 教員
	服部	博行	株式会社ヴィッツ 取締役 組込みソフトウェア開発部部長
	伊藤	政光	株式会社エスワイシステム 執行役員 中部事業部 事業部長
	小林	靖英	株式会社アフレル 代表取締役社長
	柴原	健次	エキスパートプロモーション 代表
	飯塚	久仁子	有限会社ザ・ライスマウンド 取締役
	飯塚	正成	一般社団法人全国専門学校情報教育協会 專務理事

### 平成 24 年度文部科学省委託 「東日本大震災からの復興を担う専門人材育成支援事業」 東北の復興を担う自動車組込みエンジニア育成支援プロジェクト

実践!自動車組込み技術者入門 FPGAとマイコンの連携システム ソフトウェア編

### 平成 25 年 3 月

学校法人日本コンピュータ学園(東北電子専門学校) 〒980-0013 宮城県仙台市青葉区花京院一丁目3番1号

問合せ先 有限会社ザ・ライスマウンド 〒164-0003 東京都中野区東中野 1-57-8 辻沢ビル 3F 電話:03-5332-5080 FAX 03-5332-5083

●本書の内容を無断で転記、掲載することは禁じます