ちょっと、前回の補足

本題の 「H8マイコンで I2C通信　その1」に 入る前に ちょっと、前回の補足を 行っておきます。 前回の 常連さんのコメントを見て、そういえば H8マイコン HEWの Cにて、I/O レジスタの説明は 全然してなかったなと 思いました。

I/Oレジスタの名称と 物理的な I/Oアドレスの 関連付けを 宣言しているのは iodefine.h です。 比較的簡単な I/Oポートの設定に使用するレジスタを 紹介します。

① 各ポートの 入出力方向の設定レジスタ:
P1DDR, P2DDR, P3DDR, P4DDR, P5DDR, P6DDR, P7DDR, P8DDR, P9DDR, PADDR, PBDDR 計 11個 あります。設定内容は、各ポートの bit の入出力方向を指定します。 DDRは データディレクションレジスタの略と思われます。

例として iodefine.h内の P1DDRの 宣言を 表示します。

```
#define P1DDR (*(volatile unsigned char *)
0xFEE000) /* P1DDR Address*/
```

2行に 分かれてしまいましたが、元は 1行です。
一見難しそうに見えますが、単純に表現するとポート1の 入出力レジスタ P1DDRの アドレスは 0xFE E0 00 である という事です。
で、volatile で 宣言されているので コンパイラのオプティマイザにて最適化しないように宣言しています。そして unsigned char * なので、符号なし byteデータの ポインタ変数と なります。

で、P1DDRに 初期値を 設定する例を 示します。
P1DDR = 0xFF; この場合は全bit 出力と なります。P1DDR = 0x0F; この場合 上位 4bitは 入力で、下位 4bitは 出力となります。 但し、1bit 単独の設定は 出来ません。

② 各ポートの 入出力データ レジスタ:
P1DR, P2DR, P3DR, P4DR, P5DR, P6DR, P7DR, P8DR , P9DR, PADR, PBDR 計 11個 あります。
DRは データレジスタと思われます。 指定ポートに データを 出力したり、ポート外部に接続されるデータを 読み込んだりします。
このレジスタは byte単位での入出力 以外に bit単独での入出力も出来ます。

例として iodefine.h内の P1DRの 宣言を 表示します。

```
#define P1DR (*(volatile union  un_p1dr *) 0xFFFFD0) /* P1DR  Address*/
```

2行に 分かれてしまいましたが 元は 1行です。
今回 新たに出て来たのは union un_p1dr です。union 共用体が 出て来ると 急に 難しくなった気がしますが、これは 1つの変数領域に 2つの 異なるデータ型を 宣言するために使用されます。 共用体は、以前 195の動画で 説明しています。この union un_p1dr で 宣言する 2つのデータ型は unsigned char と bit field です。bit field は 例えば 1byte の変数領域内を 1bit 以上の 複数の bit変数として宣言できます。 遥か昔のミニコンとかでは メモリが高価で 0～15を 扱える 4bit整数とか、0～7を 扱える 3bit整数、0～3を 扱える 2bit整数として、bit field を 使用していた様です。現在は そこまでメモリを節約する必要は 無いでしょうが、別の使い方が出て来ました。 そのうちの一つが 今回紹介する bit 単位のI/Oポート並びに 名前を付けて呼び出せる bit field の 使い方が 考案されました。これにより、unsigned char と bit field を 共用体とする事で、同時には使用出来ませんが byte単位で 入出力が出来て 且つ bit 単位でも 入出力が 出来る 柔軟性のある アクセス方法が

考案されました。

#define P1DR (*(volatile union un_p1dr *) 0xFFFFD0) /* P1DR Address*/

内の un_p1drの宣言は、以下の通りです。

```
union un_p1dr {              /* union P1DR   */
    unsigned char BYTE;      /*  Byte Access */
    struct {
        unsigned char B7:1;  /*   Bit 7  */
        unsigned char B6:1;  /*   Bit 6  */
        unsigned char B5:1;  /*   Bit 5  */
        unsigned char B4:1;  /*   Bit 4  */
        unsigned char B3:1;  /*   Bit 3  */
        unsigned char B2:1;  /*   Bit 2  */
        unsigned char B1:1;  /*   Bit 1  */
        unsigned char B0:1;  /*   Bit 0  */
    } BIT;                   /*  Bit  Access */
};
```

補足しておきますと、union un_p1dr は 2つのメンバー変数 unsigned char BYTEと struct{} BIT が同じアドレスに 配置されます。

ポート1を アクセスする時:

バイト値 0x35 を 出力する時:

```
P1DR.BYTE = 0x35;    // と なります。
```

バイト変数 sts に ポート1の値を 読み込む時:

```
sts = P1DR.BYTE;  // と なります。
```

ポート1の bit4 に 1 を 出力する時:

```
P1DR.BIT.B4 = 1;  // と なります。
```

ポート1の bit0 が 1である事を確認したら ポート1の bit7 に 1を 出力する場合。

```
if( P1DR.BIT.B0 ==1 )  P1DR.BIT.B7 = 1;
```

③ I/Oポートに関わるレジスタは あと PCRという レジスタが あります。これは、一部の I/Oポートを 入力に設定した時、プルアップ抵抗を接続するかどうかの 設定です。レジスタ内の各bitを 1 にすると 対応するポートの bitが プルアップされます。PCRレジスタは、P2PCR, P4PCR, P5PCRの 3つしか 有りません。設定のやり方は P1DRと 同じです。

I/Oポートのアクセス方法は　以上です。

I/Oポートは 比較的アクセスが　簡単ですが
　周辺回路は、一つの周辺回路に　レジスタが何本も存在し構造体宣言が　階層構造となりアクセスが　複雑化してます。

　しかし、基本は　前ページの構造体、共用体、ビットフィールドの事を　理解していれば、iodefine.h を 読み説く事は 出来ると思います。

　それと、H8マイコンで　この iodefine.h の レジスタの使い方に慣れておくと、後で　RXマイコンに移行した時も、同じ名前の iodefine.h があるので、少なくとも　I/Oポートの扱い方は　レジスタ名は　多少異なると思いますが、同様に記述する事が 出来ます。

　今回の　I/Oレジスタの説明は　ここまでとしておきます。

H8マイコンで I2C通信 その1

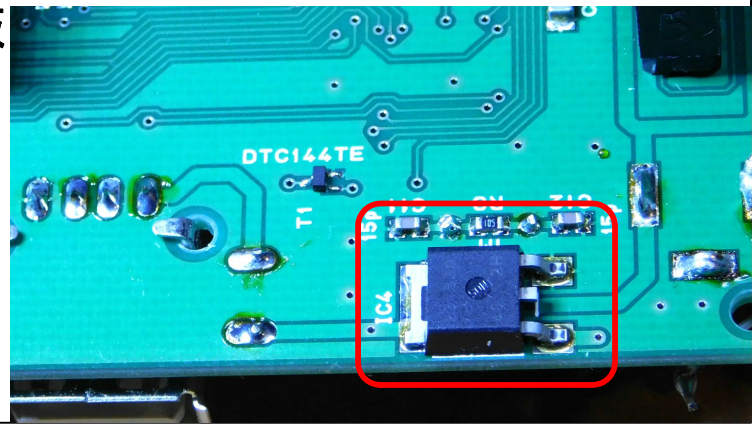
では、本題に 入ります。

ソフトで I2C通信を行うので、まず AE-3069 USB基板の どの足ピンから I2Cの信号を出すかを 決める必要が あります。 で、高速化を実現するために、一つの Byte単位の I/Oポートを 占有します。

I2Cインタフェースは 2線式ですが、I2Cデバイスによっては、割り込みの信号を 出す物（ ADCの変換完了割り込みや RTCの 1秒パルス出力 ）、逆に デバイスに タイミング信号を出す必要がある物が ありません。

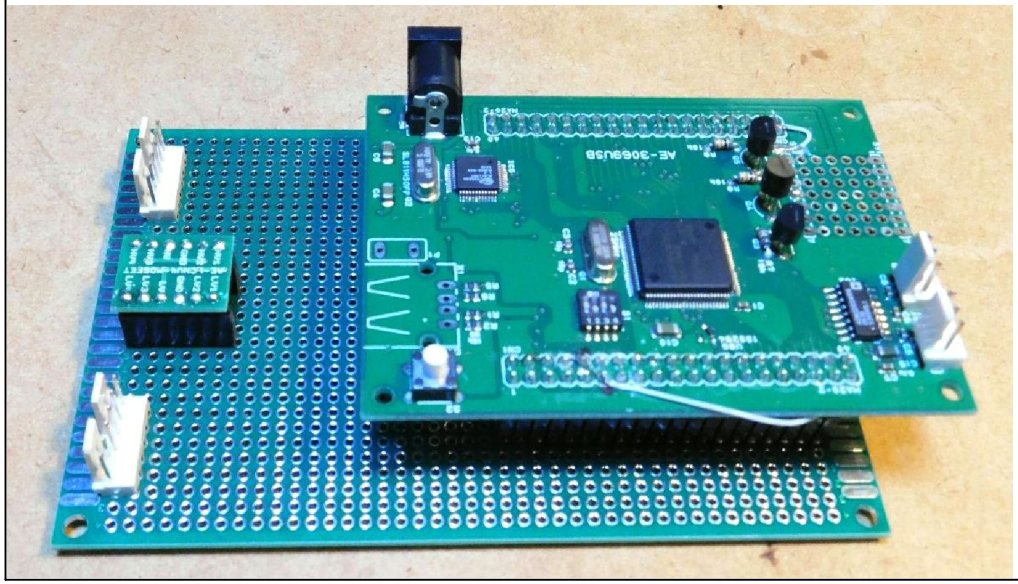
よって今回 ハード的には 4bitの信号線を出す事にします。 で、H8マイコンは 5Vで 動作しますが、I2Cデバイスは 3.3Vの物が 多いので 信号線の電圧変換が 必要です。

信号線の電圧変換は 秋月電子で販売されている「4ビット 双方向ロジックレベル変換モジュール」を 使用します。DIP 12ピンの小基板に なっているので アマチュアにとって扱いやすいです。で、都合のいい事に AE-3069 USB基板の裏側に 3.3Vの3端子電源ICが 実装されています。 これは サイプレスの USBホストLSIが 3.3V駆動の関係で、実装されています。センサー基板ぐらいであれば 十分駆動出来ると思います。



最初、基板を作る事は考えていませんでしたが、外付けの部品が　やや増えそうなので、AE-3069USB基板より、ちょっと大きめの 2.54 mmピッチのユニバーサルボードを、H8マイコンボード下に　ベースボードとして 配置する事にします。 そしてベースボード上に　追加の部品を　実装します。 凡そ　このような構成でハードウェアの追加を 行います。

（ 下の画像は　部品の仮配置です。）



次は、I/Oポートの一覧表を見て　どのポートから　I2Cの信号を取り出すか検討します。

I2Cの信号は SCLと　SDAの2本です。 この2つの信号は 同じポートから取り出したいと考えます。 あと、2本 割り込み入力信号 IRQ と ロード出力信号として LD　計 4bit用意します。

実際に使ってみないと 分からない部分も あるので 暫定的に　I2Cで使用する ポートを　決めておきます。 尚、I/Oポートの一覧表の周りは　空きが　無いので　説明を書き込みにくいので、暫定的に決めたポートを 先に　表示しておきます。

I2C.SCL　:　P6.b0 ＜CN1-1＞
I2C.SDA　:　P6.b1 ＜CN1-2＞
I2C.LD　:　P6.b2 ＜CN1-3＞
I2C.IRQ　:　P9.b4 （ IRQ4 ）＜CN2-3＞
＜ ＞内は 基板の 40Pin コネクタ番号です。

H8/3069F USB Host基板の I/Oポート表

Port	bit	信号名	CN	Setup	On Board	External
P1	b7	P17/ A7	2.28	Abus_7	D-RAM	
	b6	P16/ A6	2.27	Abus_6	D-RAM	
	b5	P15/ A5	2.26	Abus_5	D-RAM	
	b4	P14/ A4	2.25	Abus_4	D-RAM	
	b3	P13/ A3	2.24	Abus_3	D-RAM	
	b2	P12/ A2	2.23	Abus_2	D-RAM	
	b1	P11/ A1	2.22	Abus_1	D-RAM	
	b0	P10/ A0	2.21	Abus_0	USB, D-RAM	
P2	b7	P27/ A15	2.36	Abus_15	?	(In)
	b6	P26/ A14	2.35	Abus_14	?	(In)
	b5	P25/ A13	2.34	Abus_13	?	(In)
	b4	P24/ A12	2.33	Abus_12	?	(In)
	b3	P23/ A11	2.32	Abus_11	?	(In)
	b2	P22/ A10	2.31	Abus_10	D-RAM	
	b1	P21/ A9	2.30	Abus_9	D-RAM	
	b0	P20/ A8	2.29	Abus_8	D-RAM	
P3	b7	P37/ D15	2.20	Dbus_15	USB, D-RAM	
	b6	P36/ D14	2.19	Dbus_14	USB, D-RAM	
	b5	P35/ D13	2.18	Dbus_13	USB, D-RAM	
	b4	P34/ D12	2.17	Dbus_12	USB, D-RAM	
	b3	P33/ D11	2.16	Dbus_11	USB, D-RAM	
	b2	P32/ D10	2.15	Dbus_10	USB, D-RAM	
	b1	P31/ D9	2.14	Dbus_9	USB, D-RAM	
	b0	P30/ D8	2.13	Dbus_8	USB, D-RAM	

Port	bit	信号名	CN	Setup	On Board	External
P4	b7	P47/ D7	2.12			
	b6	P46/ D6	2.11			
	b5	P45/ D5	2.10			
	b4	P44/ D4	2.9			
	b3	P43/ D3	2.8			
	b2	P42/ D2	2.7			
	b1	P41/ D1	2.6			
	b0	P40/ D0	2.5			
P5	b3	P53/ A19	2.40			(In)
	b2	P52/ A18	2.39			(In)
	b1	P51/ A17	2.38			(In)
	b0	P50/ A16	2.37			(In)

Port	bit	信号名	CN	Uses	On Board	External
P6	b7	P67/ ϕ	1.4			(In)
	b6	P66/ LWR	1.9			
	b5	P65/ HWR	1.8	HWR	USB, D-RAM	
	b4	P64/ RD	1.7	RD	USB, D-RAM	
	b3	P63/ AS	1.6			
	b2	P62/ BACK	1.3			I2C.LD
	b1	P61/ BREQ	1.2			I2C.SDA
	b0	P60/ WAIT	1.1			I2C.SCL

P4と P5は 空いてますが、P5 入力専用ポートのようです。P1、P2、P3 及び P6の一部は CPUの Bus Lineを、出すために 使用されています。

Port	bit	信号名	CN	Uses	On Board	External
P7 Inp only	b7	P77/ AN7/DA1	1.19			
	b6	P76/ AN6/DA0	1.18			
	b5	P75/ AN5	1.17			
	b4	P74/ AN4	1.16			
	b3	P73/ AN3	1.15			
	b2	P72/ AN2	1.14			
	b1	P71/ AN1	1.13			
	b0	P71/ AN0	1.12			

Port	bit	信号名	CN	Uses	On Board	External
P8	b4	P84/ CS0	1.24			(In)
	b3	P83/ CS1/IRQ3	1.21	CS1	USB	
	b2	P82/ CS2/IRQ2	1.22	CS2	D-RAM	
	b1	P81/ CS3/IRQ1	1.23			
	b0	P80/ RFSH/IRQ0	1.20	IRQ0	USB	

P9	b5	P95/ IRQ5/SCK1	2.4			
	b4	P94/ IRQ4/SCK0	2.3			I2C.IRQ
	b3	P93/ RxD1		COM1.Rx	COM1.Rx	
	b2	P92/ RxD0		COM0.Rx	COM0.Rx	
	b1	P91/ TxD1		COM1.Tx	COM1.Tx	
	b0	P90/ TxD0		COM0.Tx	COM0.Tx	

Port	bit	信号名	CN	Uses	On Board	External
PA	b7	PA7/ A20/TIOCB2/TP7	1.32			
	b6	PA6/ A21/TIOCA2/TP6	1.31			
	b5	PA5/ A22/TIOCB1/TP5	1.30			
	b4	PA4/ A23/TIOCA1/TP4	1.29			
	b3	PA3/TIOCB0/TCLKD/TP3	1.28			MMC.RxD
	b2	PA2/TIOCA0/TCLKC/TP2	1.27			MMC.TxD
	b1	PA1/TCLKB/TEND1/TP1	1.26			MMC./CS
	b0	PA0/TCLKA/TEND0/TP0	1.25			MMC.CLK

PB	b7	PB7/ RxD2	1.40		ocTr.Q3	確認用 LED1
	b6	PB6/ TxD2	1.39		ocTr.Q4	確認用 LED2
	b5	PB5/ SCK2/LCAS	1.38		ocTr.Q5	確認用 LED3
	b4	PB4/ UCAS	1.37	UCAS	UCAS	
	b3	PB3/ CS4/DREQ1/TMI03	1.36			I2C.LE
	b2	PB2/ CS5/TMO2	1.35			
	b1	PB1/ CS6/DREQ0/TMI01	1.34			
	b0	PB0/ CS7TMO0	1.33			

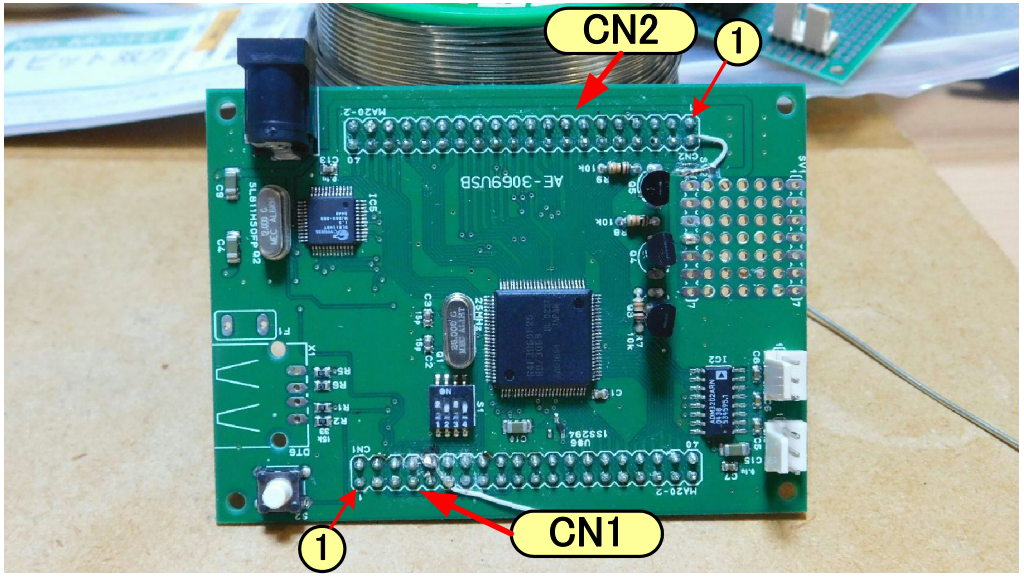
この I/Oポート表の 続きのページにて I2Cで使用する予定の I/Oポートは 緑の I2C.IRQ信号です。

I2C.IRQ : P9.b4 (IRQ4)＜CN2-3＞

コネクタ CN1の ピン アサイン

用途	信号名	CN1		信号名	用途
I2C. SCL	P60	1	2	P61	I2C. SDA
I2C. Ld	P62	3	4	P67	
	NMI	5	6	P63/-AS	
Ex. Rd	P64/-RD	7	8	P65/-HWR	Ex. HiWr
	P66/-LWR	9	10	AVcc	
	Vref	11	12	P70_i/AN0	
	P71_i/AN1	13	14	P72_i/AN2	
	P73_i/AN3	15	16	P74_i/AN4	
	P75_i/AN5	17	18	P76_i/AN6/DA0	
	P77_i/AN7/DA1	19	20	P80/-IRQ0	
	P81/-IRQ1/-CS3	21	22	P82/-IRQ2/-CS2	Ex. D-RAM
Ex. USBif	P83/-IRQ3/-CS1	23	24	P84/-CS0	
MMC. Clk	PA0	25	26	PA1	MMC. /Cs
MMC. Sd	PA2	27	28	PA3	MMC. Rd
	PA4	29	30	PA5	
	PA6	31	32	PA7	

用途	信号名	CN1		信号名	用途
	PB0/-CS7	33	34	PB1/-CS6	
	PB2/-CS5	35	36	PB3/-CS4	
	PB4/-UCAS	37	38	PB5/-LCAS	
	PB6/TxD2	39	40	PB7/RxD2	



基板上の 40Pinコネクタの CN1と CN2 コネクタ 及び1番ピンの位置を 指しています。

コネクタ CN2の ピン アサイン

用途	信号名	CN2		信号名	用途
	Gnd	1	2	+5V	
I2C.IRQ	P94/-IRQ4	3	4	P95/-IRQ5	
	P40/D0	5	6	P41/D1	
	P42/D2	7	8	P43/D3	
	P44/D4	9	10	P45/D5	
	P46/D6	11	12	P47/D7	
Bus_D8	P30/D8	13	14	P31/D9	Bus_D9
Bus_D10	P32/D10	15	16	P33/D11	Bus_D11
Bus_D12	P34/D12	17	18	P35/D13	Bus_D13
Bus_D14	P36/D14	19	20	P37/D15	Bus_D15
Bus_A0	P10/A0	21	22	P11/A1	Bus_A1
Bus_A2	P12/A2	23	24	P13/A3	Bus_A3
Bus_A4	P14/A4	25	26	P15/A5	Bus_A5
Bus_A6	P16/A6	27	28	P17/A7	Bus_A7
Bus_A8	P20/A8	29	30	P21/A9	Bus_A9
Bus_A10	P22/A10	31	32	P23/A11	Bus_A11

用途	信号名	CN2		信号名	用途
Bus_A12	P24/A12	33	34	P25/A13	Bus_A13
Bus_A14	P26/A14	35	36	P27/A15	Bus_A15
	P50/A16	37	38	P51/A17	
	P52/A18	39	40	P53/A19	

コネクタ2では 3番ピンの P94/IRQ4 を I2C.IRQとして 引き出す予定です。

で、このコネクタピンアサイン表を見て、一つ まずい事に 気付きました。 電源線ですが、GNDと 5Vは 出ていますが、3.3Vが 無いです。

コネクタ1側では 電源関係は A/Dコンバータ用のアナログ電源電圧と 基準電圧の入力ピンだけで 3.3V出力は 無いです。

この、CN1、CN2の ピンアサインは 変えたくないのでちょっと冗長な感じは しますが 5Vを使って ベースボード側に 3.3Vの 3端子電源 IC を 付ける事にします。

H8マイコン I2Cインタフェース回路図