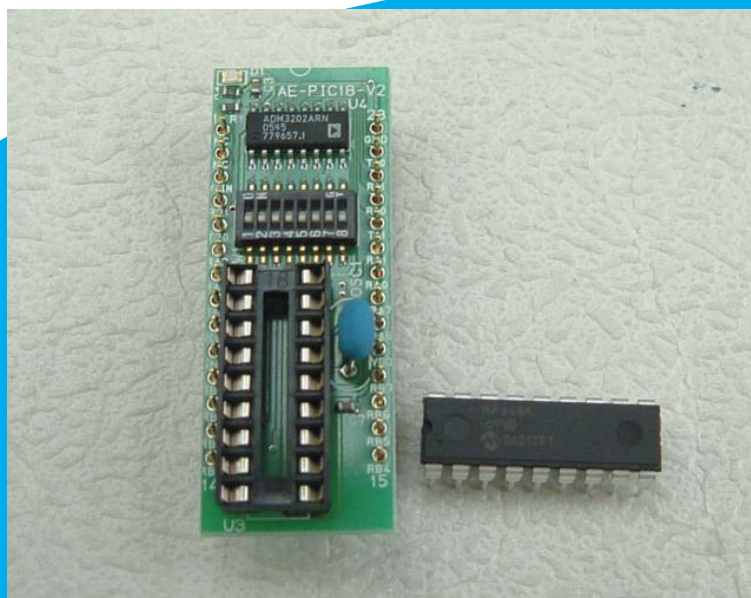


PIC16F648A CPUボードモジュールキット Ver.2

小型マイコンボードがさらに小さく、そしてRS232信号の切り替えDIPスイッチを搭載してリニューアルしました。
この基板をそのままAKI - PICプログラムの28ピンソケットにセットして、書き込みが行えます。
5VレギュレータIC、セラミック発振子搭載で電源(6 ~ 16V)を供給するのみで動作



AKI-PIC16Fxxx CPU ボードモジュールキット

PIC16F84A/627A/628A/648A/818/819/88 対応

47 × 18 mm超小型マイコンボード

RS232、レギュレータ (5V/150mA) 搭載、20MHz 動作

■超小型マイコンボードキットが、さらに小さく、そして RS232 信号の切り替え用 DIP スイッチを搭載し、リニューアルしました。ほとんどの PIC16 シリーズの18ピンパッケージタイプが搭載可能です。

(18ピン SOP パッケージの PIC マイコンが実装済みのキット商品は、マイコンを交換する事ができませんのでご注意ください。 18ピン DIP パッケージの PIC マイコンがセットになっている商品は、別途 PIC マイコンを入手することで交換可能です。)

■ユーザーコードのプログラミングは、この基板をそのまま AKI-PIC プログラマーの 28 ピン ZIP ソケットにセットし、単体のマイコンをプログラミングするときと同じ操作で可能。(プログラミングするときは、この基板の DIP スイッチ (SW1) を全て OFF にします。)

■ PIC16F シリーズは Flash プログラムメモリ内蔵で、約 1000 回以上の書換え可能でプログラムの開発が容易。

■ 5V レギュレータ IC、セラミック発振子が付属、この基板に電源 (6 ~ 16V) を供給するのみで動作。

■ RS232 ドライバー IC を搭載しているので、パソコンなどとのデータ通信に便利。

	プログラムメモリ (word)	データメモリ (byte)	UART モジュール		アナログコンパレータ	A/D	PWM	内蔵 RC オシレータ
			TX	RX				
PIC16F84A	1K	68	-	-	-	-	-	-
PIC16F627A	1K	224	8(RB2)pin	7(RB1)pin	2ch	-	あり	あり
PIC16F628A	2K	224	8(RB2)pin	7(RB1)pin	2ch	-	あり	あり
PIC16F648A	4K	256	8(RB2)pin	7(RB1)pin	2ch	-	あり	あり
PIC16F818	1K	128	-	-	-	10bit-5ch	あり	あり
PIC16F819	2K	256	-	-	-	10bit-5ch	あり	あり
PIC16F88	4K	368	11(RB5)pin	8(RB2)pin	2ch	10bit-7ch	あり	あり

	UART モジュール	DIP スイッチの設定	
		TX	RX
PIC16F84A	なし	5pin	1pin
PIC16F627A	有り	6pin	2pin
PIC16F628A	有り	6pin	2pin
PIC16F648A	有り	6pin	2pin
PIC16F818	なし	5pin	4pin
PIC16F819	なし	5pin	4pin
PIC16F88	有り	7pin	3pin

■ RS232 を使用するときの DIP スイッチ設定 ■

PIC16F627A/628A/648A/88 はマイコン内部に UART モジュールを内蔵しているため、表で指定された DIP スイッチを ON にして RS232 を使用します。

PIC16F84A/818/819 はマイコン内部に UART モジュールが内蔵されていないため RS232 はソフトウェア処理で実現します。この場合、どの端子を使用するかはユーザーの選択となります。表の DIP スイッチの設定は参考例です。

RS232 を使用しない場合は、DIP スイッチの 1 ~ 7 番をオフします。

DIP スイッチの 8 番を ON で、マイコンに電源を供給します。

■ 部品表 ■

番号	名称	数	備考	パッケージ
C1,C2	コンデンサ 100uF	2	積層セラミックコンデンサ 16V/100uF	3225
C3,C4,C5,C6,C7	コンデンサ 0.1uF	5	積層セラミックコンデンサ	1608
U1	レギュレータ 5V	1	XC6202P502P	SOT-89
U2	SOPパッケージ PIC マイコン	1	U2,U3 はどちらか一方のみ使用	SOP-18
U3 未実装	DIPパッケージ PIC マイコン	1	U2,U3 はどちらか一方のみ使用	DIP-18
	ICソケット 18ピン	1	U3 を使用するときは、IC ソケットを取付ける。	
D1	LED	1	チップ LED	1608
OSC1 未実装	セラミック発振子 20MHz	1		
SW1	DIP スイッチ 8ビット	1	CHS-08TB	SOP
R1	抵抗 33K Ω	1	チップ抵抗	1608
R2	抵抗 1K Ω	1	チップ抵抗	1608
CN1,CN2 未実装	連結ソケット または 1x14 ヘッダーピンが 2 本	1	連結ソケットの場合は CN1,CN2 の両方で使用	

■製作■

あらかじめ部品表と部品を照らし合わせ、数量などを確認してから製作してください。ほとんどの部品は実装済みです。セラミック発振子 (OSC1)、18ピン DIP ソケット (U3)、連結ソケット (CN1,CN2) の3個の部品のみ取り付けする必要があります。(18ピン DIP ソケットは、DIP タイプの PIC マイコンがセットになっているキット商品のみ含まれています。)

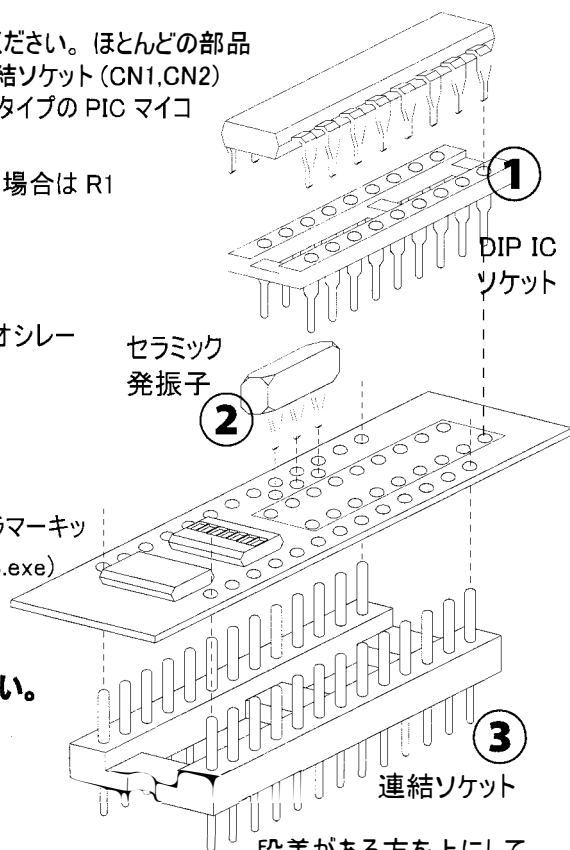
● MCLR を RA5 入力端子として使用する場合で、プルアップ抵抗が不要な場合は R1 (33K Ω) を取り外してください。

- ① 基板の表面に DIP IC ソケットを U3 に取付け、マイコンを挿入します。
(DIP タイプのマイコンがセットのキットのみ)
- ② 基板の表面にセラミック発振子を OSC1 に取り付けます。内蔵4MHzRC オシレータを使用するときは、セラミック発振子 (OSC1) を取り付けないでください。
セラミック発振子は、3本足の中央がコモン端子で極性はありません。
- ③ 最後に基板の裏面に連結ソケットを取り付けます。
1x14 ヘッダーピンを CN1、CN2 に取り付ける場合もあります。

■プログラミング■ ユーザコードの書き込みは、この基板のまま AKI-PIC プログラマーキットの 28ピン ZIP ソケットで行います。プログラマーソフト (picpgm.exe,picpgm6.exe) の画面で「デバイス選択」は、U2 または U3 に実装されているマイコンの型番を指定します。

★プログラミングを行う時は、SW1 のスイッチを全て「OFF」にしてください。

AKI-PIC プログラマー Ver3/3.5 サポートホームページ：
http://akizukidenshi.com/down/tk/pic_pgm/pic_pgm.htm
AKI-PIC プログラマー Ver4 サポートホームページ：
http://akizukidenshi.com/down/tk/picpgm_v4/



段差がある方を上にして最後に基板を取り付けます。

■ CN1,CN2 接続表 ■

CN1	PIC16F84A	PIC16F627A /628A/648A	PIC16F818/819	PIC16F88	RS232 ドライバー IC と DIP スイッチ	
1			NC (未接続)			
2			NC (未接続)			
3			V-IN 電源入力 +6 ~ 16V			
4			RX2IN (RS232:PIC-RX)			
5			TX2OUT (RS232:PIC-TX)			
6	RA2	RA2/AN2/VREF	RA2/AN2/VREF-	RA2/AN2/VREF/VREF-	-	-
7	RA3	RA3/AN3/CMP1	RA3/AN3/VREF+	RA3/AN3/VREF+/C1OUT	-	T2IN-DIP SW-5
8	RA4	RA4/T0CKI/CMP2	RA4/AN4/T0CKI	RA4/AN4/T0CKI/C2OUT	R2OUT-DIP SW-1	-
9	MCLR	RA5/MCLR	RA5/MCLR	RA5/MCLR	-	-
10			GND		-	-
11	RB0/INT	RB0/INT	RB0/INT	RB0/INT/CCP1	R2OUT-DIP SW-4	-
12	RB1	RB1/RX/DT	RB1/SD/SDA	RB1/SDI/SDA	R2OUT-DIP SW-2	-
13	RB2	RB2/TX/CK	RB2/SDO/CCP1	RB2/SDO/RX/DT	R2OUT-DIP SW-3	T2IN-DIP SW-6
14	RB3	RB3/CCP1	RB3/CCP1/PGM	RB3/CCP1/PGM	-	-

CN2	PIC16F84A	PIC16F627A /628A/648A	PIC16F818/819	PIC16F88	RS232 ドライバー IC	
28			GND			
27			TX1OUT (RS232)			
26			RX1IN (RS232)			
25			RX1OUT (LOGIC)			
24			TX1IN (LOGIC)			
23	RA1	RA1/AN1	RA1/AN1	RA1/AN1	-	-
22	RA0	RA0/AN0	RA0/AN0	RA0/AN0	-	-
21	OSC1	RA7/OSC1/CLKIN	RA7/OSC1/CLKI	RA7/OSC1/CLKI	-	-
20	OSC2	RA6/OSC2/CLKOUT	RA6/OSC2/CLKO	RA6/OSC2/CLKO	-	-
19			VDD (+5VOUT)		-	-
18	RB7/PGD	RB7/T1OSI/PGD	RB7/T1OSI/PGD	RB7/AN6/T1OSI/PGD	-	-
17	RB6/PGC	RB6/T1OSO/T1CKI/PGC	RB6/T1OSO/T1CKI/PGC	RB6/AN5/T1OSO/T1CKI/PGC	-	-
16	RB5	RB5	RB5/SS	RB5/SS/TX/CK	-	T2IN-DIP SW-7
15	RB4	RB4/PGM	RB4/SCK/SCL	RB4/SCK/SCL	-	-

■回路の説明■ 回路図を参考にお読みください。

① CN1,CN2

外部に接続するコネクタです。28ピンICの形をしています。それぞれの機能は【CN1,CN2 接続表】をご覧ください。

② PIC マイコン

PIC16Fxxx は全てのピンが CN1, CN2 に接続されています。U2 または U3 には、PIC マイコンが取り付けられています。このモジュール使用した商品には2種類あります。一つは、U2 に SOP タイプのマイコンを実装したタイプで、もう一つは、U3 に DIP タイプのマイコンを取り付けるタイプです。U2 と U3 には同時にマイコンを取り付けることはできません。(マイコンの種類によって使用できる機能が異なります。違いは【CN1,CN2 接続表】をご参考して下さい。)

★ RA7(OSC1)、RA6(OSC2) は、OSC 端子として使用するか、I/O 端子として使用するか選択できます。OSC 端子として使用するときはセラミック発振子を取り付けます。内蔵 RC-OSC を使用し、RA6、RA7 を I/O 端子として使用するときは、セラミック発振子を取り付けしないでください。(内蔵 RC-OSC は 4MHz 動作)

★ RA5(MCLR) は、MCLR 端子として使用するか、RA5(入力専用)端子として使用するか選択できます。この端子は、R1:33K Ω でプルアップされています。この抵抗が不要な場合は、R1 を取り外してください。

③ 電源

CN1-3:V-INピンが電源入力。入力電圧は5～16Vです。CN1-10ピンとCN2-28ピンがGND。CN2-19ピンからは5Vが出力されます。PICマイコンに電源を供給する時は、SW1:DIPスイッチの8番をONにセットします。

④ 発振子

20MHz セラミック発振子が付いています。コンデンサ内蔵なので、このままで 20MHz 動作します。内蔵 RC オシレータを使用するときは 4MHz 動作となります。

⑤ RS232

RS232 IC は、送信 :2ch/ 受信 :2ch 内蔵。1ch 分は SW1:DIP スイッチを経由しマイコンの TX/RX 端子に接続されます。

■ DIP スイッチ設定 ■

■ DIP スイッチ設定 - RS232 の使用時 ■

RS232 を使用しないときは、DIP スイッチ の1～7番を OFF します。PIC16F627A/628A/648A/88 は UART モジュールがマイコンに内蔵されているので、使用するときの TX,RX 端子は固定されます。RS232 を 使用するときには表で指定された DIP スイッチを ON にセットして使用します。

しかし、PIC16F84A/818/819 はマイコンに UART モジュールが内蔵されていないので RS232 はソフトウェア処理で実現します。この場合、どの端子を使用するかはユーザの判断となります。

■ DIP スイッチ設定 - 電源 ■

8番のスイッチを ON にセットすると PIC マイコンに電源が供給されます。

■ サンプルプログラムの説明 ■

RS232 データ送受信フォーマットは最初が Low レベルのスタートビット、次が LSB ビットから8ビット分のデータビット、最後が Hi レベルのストップビットとなります。

受信プログラム

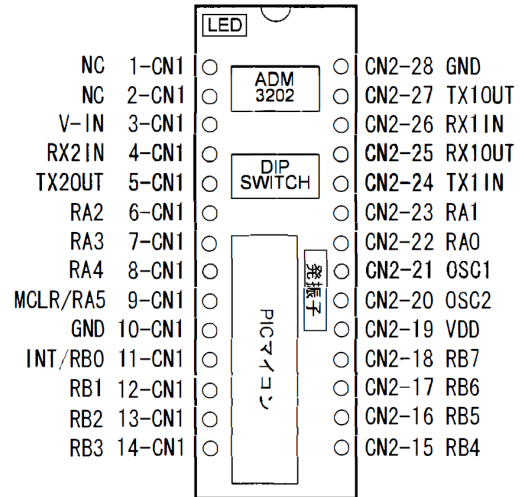
サブルーチン名は、SCI_RX_DATA です。

通常、RS232 信号ラインは、Hi レベルです。この信号が Low になった時から、データ受信処理を開始します。最初は、1ビット周期+1ビット周期/2の時間ウエイトを行い、その後、1ビット周期時間でレベルのサンプリングを8回行います。最後のデータビットサンプリング後、1ビット周期時間ウエイトしてからサブルーチンを抜けます。データを格納するレジスタ名は、SCI_BUFFER で、MSB ビットに受信ビットをセットし右シフトで格納します。

送信プログラム

サブルーチン名は、SCI_TX_DATA です。

送信サブルーチンの最初で、BCF INTCON, GIE 命令、INTCON レジスタの GIE ビットをクリアすることで、これ以降割り込みを禁止します。データ送信中に割り込みが発生すると、1ビット周期にずれが発生してしまうので、これを防止するためです。最初に、スタートビットのレベルである Low を出力し、その後、SCI_BUFFER レジスタの内容を0ビット目から8ビット分出力します。最後にストップビットである Hi レベルを出力し1ビット周期分のウエイトします。BSF INTCON, GIE 命令によって割り込みを有効に設定しサブルーチンを抜けます。



	UART モジュール	DIP スイッチの設定	
		TX	RX
PIC16F84A	なし	いずれか一つを選択 5pin - RA3 6pin - RB2 7pin - RB5	いずれか一つを選択 1pin - RA4 2pin - RB1 3pin - RB2 4pin - RB0/INT
PIC16F627A	有り	6pin - RB2:TX	2pin - RB1:RX
PIC16F628A	有り	6pin - RB2:TX	2pin - RB1:RX
PIC16F648A	有り	6pin - RB2:TX	2pin - RB1:RX
PIC16F818	なし	いずれか一つを選択 5pin - RA3 6pin - RB2 7pin - RB5	いずれか一つを選択 1pin - RA4 2pin - RB1 3pin - RB2 4pin - RB0/INT
PIC16F819	なし	いずれか一つを選択 5pin - RA3 6pin - RB2 7pin - RB5	いずれか一つを選択 1pin - RA4 2pin - RB1 3pin - RB2 4pin - RB0/INT
PIC16F88	有り	7pin - RB5:TX	3pin - RB2:RX

1. サンプルプログラムで使用している指示言語 (Directive Language) の説明

LIST P=16F84A,ST=OFF,R=DEC,F=INHX8M

P=16F84A マイコンの種類を指定します。
 ST=OFF シンボルテーブル情報をリストファイルに出力しないようにします。
 R=DEC 基数を10進数に設定します。この指定を行わなかった時、movlw 10 などの '10' は16進数になります。
 F=INHX8M hex ファイル、書き込みデータファイルのフォーマットの指定を行います。

INCLUDE P16F84A.INC

インクルードファイルの指定を行います。ここで使用するインクルードファイル P16F84A.INC は、MPLAB をインストール時に、インストールされたファイルをそのまま使用しています。内容は、スペシャルファンクションレジスタの定義です。

ERRORLEVEL -207,-302,-306

ソースコードのアセンブル時、エラーや、警告 (warning)、メッセージ (Messages) があつたときアセンブラは、それに対応する番号を出力します。エラーのときは修正しない限り hex ファイルが生成されません。しかし、エラー以外は、ただ単に注意を促すものなので hex ファイルは生成されます。この警告やメッセージの中には、あまり意味をなさないものもあり、これらを出さないようにするのが "ERRLEVEL" です。

201 スペースまたはタブの後にラベル名が見つかった教えています。(warning)
 302 BANK0 以外のレジスタをアクセスしようとしているのでバンク切り替えビットを確認してください。(messages)
 306 異なるページへアクセスしようとしているのでページ切り替えビットを確認してください。(messages)

_CONFIG _HS_OSC & _PWRTE_ON & _WDTRDIFC & _CP_OFF & _WDT_OFF & _CP_OFF

コンフィギュレーションワードの設定をソースコード内で行うとき、_config を使用します。_config 以降に続く文字列は、インクルードファイル P16F84A.INC 内で宣言されています。

_HS_OSC オシレータモードを HS に指定します。4MHz のときは、_XT_OSC、8MHz,10MHz,20MHz の時は、_HS_OSC を指定します。
 _PWRTE_ON ON のとき、電源の投入時に 72ms 待った後、命令実行を開始します。電源の立ち上がりが遅い時や、電源投入時に電圧がふらつく可能性があるときは ON にします。電源投入と同時にマイコンを動作させる必要があるとき意外は ON を推奨します。
 _WDT_OFF 今回のサンプルプログラムでは、ウオッチドックタイマーを使用しませんので OFF です。このタイマーを使用するときは、ウオッチドックタイマーがオーバーフローする前に clrwdt 命令を実行させる必要があります。
 _CP_OFF マイコンにコードを書き込んだ後、書き込まれたコードの内容を読み出せないようにします。今回は、コードプロテクトをかける必要が無いので OFF です。PIC16F シリーズに限りコードプロテクトを ON にしても、再プログラミングをすることができます。

#define BANK0 BCF STATUS,RP0

#define BANK1 BSF STATUS,RP0

デファインを使用することで、よく使う命令を、別な名称で置き換えることができます。ここでは、バンク0への切り替え命令 BCF STATUS,RP0 を BANK0 に置き換えています。

END

アセンブルの終了を指定しています。これ以降に記述されている内容はアセンブル時には無視されます。

備考 (EEPROM データメモリの初期データ)

マイコンにコードをプログラミングする際に、EEPROM データメモリにもプログラミングすることができます。この場合は、ソースコード内に以下の記述を行います。

```
ORG 0x2100
DE 0x41,0x42,0x43
```

0x2100 番地が、EEPROM データメモリアドレス0に対応しています。上記の場合、アドレス0に 0x41、アドレス1に 0x42、アドレス2に 0x43 がプログラミングされます。上記の記述以降にプログラムコードを記述する場合は、ORG "コードを記述を開始するプログラムメモリアドレス" を記述してからプログラムコードの記述を行います。

2. 割り込み

PIC16F84A の割り込みは、4種類用意されています。

- a.RB0/INT 端子：外部割り込み。立上り、立下りエッジを OPTION レジスタで選択できます。サンプルプログラムは立下りエッジを選択しています。
- b.タイマー 0 のオーバーフロー割り込み。アンダーフロー割り込みはサポートしていません。
- c.PORTB 変化割り込み。RB4-7 端子の追加機能です。4x4 キーパッドとのインターフェイスなどのために用意されています。
- d.EEPROM 書き込み終了割り込み。EEPROM データメモリへの書き込みが完了した時に割り込みが発生します。

電源投入時や、ウオッチドックタイマーが動作した時は、プログラムメモリアドレス0番地から、命令実行を開始しますが、割り込み発生時はイベントの種類に関わらずプログラムメモリアドレス4番地をサブルーチンコールし命令を実行します。

このとき重要になるのが、ワーキングレジスタ (W)、スペシャルファンクションレジスタの STATUS、FSR、PCLATH の内容の一時退避と、割り込み処理終了後の元の処理に戻る前の一時退避データの復元です。この部分のコードを以下に記します。(ソフトウェアの作り方によっては、ユーザ定義レジスタの退避も必要になります。) 割り込み使用時の注意点は、割り込みフラグは、自動的にクリアされないのでユーザーコード内でクリアする必要がある点で

```

=====
; interrupt vector
=====
PUSH MOVWF    ORG    0x004
               TEMP_W
               SWAPF  STATUS,W
               CLRF   STATUS
               MOVWF  TEMP_S
               MOVF   FSR,W
               MOVWF  TEMP_F
               MOVF   PCLATH,W
               MOVWF  TEMP_P
               MOVLW  HIGH  PUSH
               MOVWF  PCLATH
               ;=====
               CALL   SCI_RX_DATA
               BSF   FLAG,RCIF
               BCF   INTCON,INTF
               ;=====
POP            CLRF   STATUS
               MOVF  TEMP_F,W
               MOVWF FSR
               MOVF  TEMP_P,W
               MOVWF PCLATH
               SWAPF TEMP_S,W
               MOVWF STATUS
               SWAPF TEMP_W,F
               SWAPF TEMP_W,W
               RETFIE
               ;=====
               ; BANK change
               ; GIE=1
=====

```

す。以下の例では、BCF INTCON,INTF 命令がこれにあたります。割り込みから元のルーチンに戻る際は、RIEFIE 命令を使用します。RETURN 命令との違いは、INTCON.GIE ビットを自動的にセットする点です。(割り込み発生時は、自動的に INTCON.GIE ビットがクリアされます。) 全ての割り込みの有効、無効の設定は、INTCON.GIE ビットにて制御します。

3. アセンブル

ソースコードのアセンブルは、MPASMWIN.EXE を使用します。MPLAB (www.microchip.com から無償ダウンロードできる統合開発環境) で、” Build1” を行う時も、MPASMWIN.EXE を呼び出しアセンブルを行っています。デフォルト状態で MPASMWIN.EXE を使用したとき、ソースコード内の大文字、小文字の区別はされません。

```

■ I/O 端子ポーリングによる RS232 ■ PIC16F84A 20MHz@9600,8,N,1 DIP-SW1,5,8-ON

LIST          P=16F84A,ST=OFF,R=DEC,F=INHX8M
INCLUDE      P16F84A.INC

ERRORLEVEL -207,-302,-306
_CONFIG    _HS_OSC & _PWRTE_ON & _WDT_OFF & _CP_OFF

;=====
;#define    BANK0    BCF STATUS,RPO
;#define    BANK1    BSF  STATUS,RPO
;=====
;レジスタ設定
;=====
SCI_BUFFER   EQU 0x20
SCI_LOOP_CNT EQU 0x21
SCI_WAIT_CNT EQU 0x22

;=====
RS232_OFFSET EQU 190
RS232_WAIT   EQU 126
TX_PIN       EQU 3
RX_PIN       EQU 4
;=====
;リセットベクター
;=====
ORG          0x0000
CLRF        STATUS
CLRF        PCLATH

;=====
;イニシャライズルーチン
;=====
INITIALIZEBANK0 MOVW  B'00001000' PORTA
MOVWF       PORTA
BANK1      MOVW  B'11110111' TRISA
MOVWF       TRISA

;=====
;メインルーチン
;=====
MAIN        BANK0  PORTA,_RX_PIN
           BTFSC  MAIN
           GOTO   MAIN

           CALL   SCI_RX_DATA
           CALL   SCI_TX_DATA
           GOTO   MAIN

;=====
;RS232 データ送信サブルーチン
;input
SCI_BUFFER
;=====
SCI_TX_DATA BCF  PORTA,_TX_PIN
           MOVLW RS232_WAIT+1
           Wait For START

           CALL   SCI_WAIT
           MOVLW 8
           MOVWF SCI_LOOP_CNT
           RRF   SCI_BUFFER,F
           BTFSC STATUS,C
           GOTO SCI_TX_HI
           PORTA,_TX_PIN
           GOTO $+3
           PORTA,_TX_PIN
           BSF  SCI_TX_HI
           NOP
           GOTO $+1
           MOVW  RS232_WAIT
           CALL  SCI_WAIT
           DECFSZ SCI_LOOP_CNT,F
           GOTO SCI_TX_LOOP
           BSF  PORTA,_TX_PIN
           MOVLW RS232_WAIT
           CALL  SCI_WAIT
           RETURN
           Wait For STOP bit
           Wait

;=====
;RS232 データ受信ルーチン
;output SCI_BUFFER
;=====
SCI_RX_DATA MOVW  RS232_OFFSET
           CALL  SCI_WAIT
           MOVW  8
           MOVWF SCI_LOOP_CNT
           RRF   SCI_BUFFER,F
           BTFSC PORTA,_RX_PIN
           GOTO SCI_RX_HI
           BCF  SCI_BUFFER,7
           GOTO $+3
           BSF  SCI_BUFFER,7
           NOP
           GOTO $+1
           MOVW  RS232_WAIT
           CALL  SCI_WAIT
           DECFSZ SCI_LOOP_CNT,F
           GOTO SCI_RX_LOOP
           RETURN
           Wait

;=====
;RS232 ウェイトルーチン
;=====
SCI_WAIT   MOVWF  SCI_WAIT_CNT
           CLRWDT
           DECFSZ SCI_WAIT_CNT,F
           GOTO  SCI_WAIT+1
           RETURN
           bit

           bit

```

```

■ 割り込みによる RS232 ■ PIC16F819 20MHz@9600,8,N,1 DIP-SW4,5,8-ON

LIST          P=16F819,ST=OFF,R=DEC,F=INHX8M
INCLUDE      P16F819.INC

ERRORLEVEL -207,-302,-306
_CONFIG    _HS_OSC & _PWRTE_ON & _WDT_OFF & _CP_OFF & _DEBUG_OFF
           & _WRT_ENABLE_OFF & _CPD_OFF & _LVP_OFF & _BODEN_OFF & _MCLR_OFF

;#define    BANK0    BCF STATUS,RPO
;#define    BANK1    BSF  STATUS,RPO
;=====
;レジスタ設定
;=====
FLAG        EQU 0x20
RCIF        EQU 0x00
SCI_BUFFER   EQU 0x21
SCI_LOOP_CNT EQU 0x22
SCI_WAIT_CNT EQU 0x23
TEMP_W      EQU 0xF0
TEMP_S      EQU 0xF1
TEMP_F      EQU 0xF2
TEMP_P      EQU 0xF3

;=====
RS232_OFFSET EQU 190
RS232_WAIT   EQU 126
TX_PIN       EQU 3
RX_PIN       EQU 0
;=====
;リセットベクター
;=====
ORG          0x0000
CLRF        STATUS

;=====
;interrupt vector
;=====
PUSH        ORG          0x004
           MOVWF  TEMP_W
           SWAPF  STATUS,W
           CLRF  STATUS
           MOVWF  TEMP_S
           MOVF  FSR,W
           MOVWF TEMP_F
           MOVF  PCLATH,W
           MOVWF TEMP_P
           MOVLW HIGH
           PUSH  PUSH
           MOVWF PCLATH

           CALL   SCI_RX_DATA
           BSF   FLAG,RCIF
           BCF   INTCON,INTF

;=====
POP         CLRF   STATUS
           MOVF  TEMP_F,W
           MOVWF FSR
           MOVF  TEMP_P,W
           MOVWF PCLATH
           SWAPF TEMP_S,W
           MOVWF STATUS
           SWAPF TEMP_W,F
           SWAPF TEMP_W,W
           RETFIE
           BANK0
           BANK change
           GIE=1

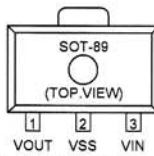
```

: イニシャライズルーチン		
INITIALIZE	BANK0 CLRF MOVLW MOVWF CLRF	FLAG B' 00001000' PORTA PORTB
	BANK1 MOVLW MOVWF MOVLW MOVWF MOVLW MOVWF	B' 00000110' ADCON1 B' 11110110' TRISA B' 11111111' TRISB
	BANK0 MOVLW MOVWF MOVLW MOVWF	B' 10000000' OPTION_REG B' 10010000' INTCON
: メインルーチン		
MAIN	BTFSZ GOTO GOTO	FLAG, RCIF DATA_GET MAIN
DATA_GET	CALL BCF GOTO	SCI_TX_DATA FLAG, RCIF MAIN
: RS232 データ送信サブルーチン		
input	SCI_BUFFER	
SCI_TX_DATA	BCF BTFSZ GOTO BCF MOVLW CALL MOVLW MOVWF RRF BTFSZ GOTO	INTCON, GIE INTCON, GIE \$-2 PORTA, TX_PIN RS232_WAIT+1 SCI_WAIT 8 SCI_LOOP_CNT SCI_BUFFER, F STATUS, C SCI_TX_HI
SCI_TX_LOOP		For START bit Wait
: RS232 データ受信ルーチン		
output	SCI_BUFFER	
SCI_RX_DATA	MOVLW CALL MOVLW MOVWF RRF BTFSZ GOTO	RS232_OFFSET SCI_WAIT 8 SCI_LOOP_CNT SCI_BUFFER, F PORTB, RX_PIN SCI_RX_HI SCI_BUFFER, 7 \$+3 SCI_BUFFER, 7
SCI_RX_LOOP		Wait
SCI_RX_LO	BCF GOTO	\$+1 RS232_WAIT SCI_WAIT
SCI_RX_HI	BSF NOP GOTO MOVLW CALL DECFSZ GOTO RETURN	\$+3 SCI_BUFFER, 7 \$+1 RS232_WAIT SCI_WAIT SCI_LOOP_CNT, F SCI_RX_LOOP
: SCI Wait subroutine		
SCI_WAIT	MOVWF CLRWDI DECFSZ GOTO RETURN END	SCI_WAIT_CNT SCI_WAIT_CNT, F SCI_WAIT+1

■ UART による RS232 ■ PIC16F88 20MHz@9600,8,N,1 DIP-SW3,7,8-ON		
LIST P=16F88, S1=OFF, R=DEC, F=INHX8M, N=130 INCLUDE P16F88.INC		
ERRORLEVEL -207, -302, -306 _CONFIG _CONFIG1, _HS_OSC & _MCLR_OFF & _LVP_OFF & _WDT_OFF & _PWRTE_ON & _BODEN_OFF & _CPD_OFF & _CP_OFF & _WRT_PROTECT_OFF _CONFIG _CONFIG2, _IESO_OFF & _FCMEN_OFF		
#define	BANK0	BCF STATUS, RPO
#define	BANK1	BSF STATUS, RPO
: レジスタの宣言		
TEMP_W	EQU	0x70
TEMP_S	EQU	0x71
TEMP_F	EQU	0x72
TEMP_P	EQU	0x73
: リセットベクター		
ORG	0x0000	
CLRF	STATUS	
CLRF	PCLATH	
GOTO	INITIALIZE	
: 割り込みベクター		
PUSH	ORG	0x0004
	MOVWF	TEMP_W
	SWAPF	STATUS, W
	CLRF	STATUS
	MOVWF	TEMP_S
	MOVF	FSR, W
	MOVWF	TEMP_F
	MOVF	PCLATH, W
	MOVWF	TEMP_P
	BTFSZ	PIR1, RCIF
	CALL	UART_GET
	BCF	PIR1, RCIF
POP	CLRF	STATUS
	MOVF	TEMP_P, W
	MOVWF	PCLATH
	MOVF	TEMP_F, W
	MOVWF	FSR
	SWAPF	TEMP_S, W
	MOVWF	STATUS
	SWAPF	TEMP_W, F
	SWAPF	TEMP_W, W
	RETFIE	
: イニシャライズルーチン		
INITIALIZE	BANK1 CLRF BANK0 MOVLW MOVWF BANK1 MOVLW MOVWF	ANSEL B' 00100000' PORTB B' 11011111' TRISB
	BANK1 MOVLW MOVWF BANK0 MOVLW MOVWF	129 SPBRG B' 00100100' TXSTA B' 10010000' RCSTA
INT_UART_ERR	MOVF MOVF ANDLW BTFSZ GOTO	RCREG, W RCSTA, W B' 00000110' STATUS, Z INT_UART_ERR
	BANK1 MOVLW MOVWF BANK0 MOVLW MOVWF MOVWF	B' 00100000' PIE1 B' 00000000' PIR1 B' 11000000' INTCON
: メインルーチン		
MAIN	BANK0 GOTO	MAIN
: データ受信と送信 サブルーチン		
UART_GET	MOVF ANDLW BTFSZ GOTO	RCSTA, W B' 00000110' STATUS, Z UART_ERROR
	BTFSZ GOTO BANK0 MOVF MOVWF GOTO	PIR1, TXIF : 送信バッファの空きを確認 \$-1 RCREG, W TXREG UART_END RCREG, W
UART_ERROR	MOVF	UART_ERROR
UART_END	RETURN END	UART エラー処理

XC6202シリーズ 20V耐圧正電圧レギュレータ

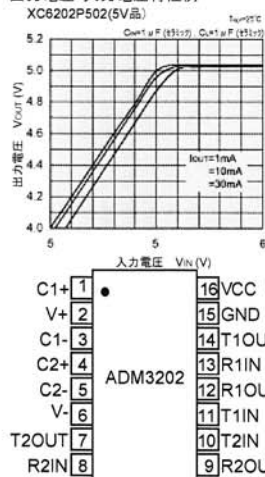
- ◆CMOS構成 : 低消費電流
- ◆動作電圧範囲 : ~20V
- ◆出力電圧精度 : ±2%
- ◆出力電圧範囲 : 1.8~18V
- ◆入出力電位差 : 200mV @ 30mA
- ◆消費電流 : 670mV @ 100mA
- ◆電流制限回路内蔵
- ◆最大出力電流 : 150mA以上



■電気的特性

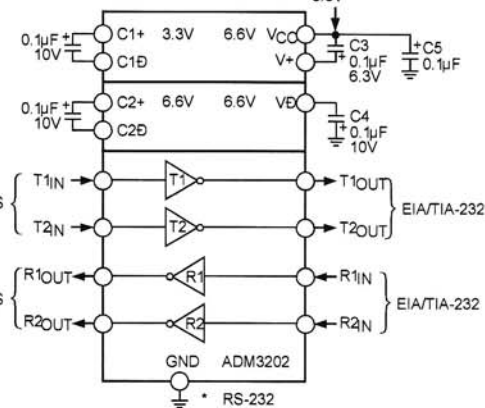
項目	記号	測定条件	MIN.	TYP.	MAX.	単位	測定回路
出力電圧	$V_{OUT}(E)^{(2)}$	$V_{IN}=8V$ $I_{OUT}=30mA$	4.900	5.000	5100	V	②
最大出力電流	I_{OUTmax}	$V_{IN}=8V$ $V_{OUT} \geq V_{OUT}(E) \times 0.9$	200	-	-	mA	②
負荷安定度	ΔV_{OUT}	$V_{IN}=8V$ $1mA \leq I_{OUT} \leq 100mA$	-	30	100	mV	②
入出力電圧差 ⁽¹⁾	V_{dIF1}	$I_{OUT}=30mA$	-	130	190	mV	②
	V_{dIF2}	$I_{OUT}=100mA$	-	440	550	mV	
消費電流	ISS	$V_{IN}=8V$	-	10	24	μA	①
入力安定度	$\frac{\Delta V_{OUT}}{\Delta V_{IN} \cdot \Delta V_{OUT}}$	$I_{OUT}=1mA$ $8V \leq V_{IN} \leq 20V$	-	0.01	0.20	%/V	②
入力電圧	V_{IN}	-	-	-	20	V	-
出力電圧温度特性	$\frac{\Delta V_{OUT}}{\Delta T_{opr} \cdot \Delta V_{OUT}}$	$I_{OUT}=30mA$ $-40^\circ C \leq T_{opr} \leq 85^\circ C$	-	±100	-	ppm/ $^\circ C$	②
短絡電流	I_{lim}	$V_{IN}=7V$	-	40	-	mA	②

出力電圧-入力電圧特性例



ADM3202ARN

ローパワー、RS-232ライン
ドライバ/レシーバ



BANK0		BANK1		BANK2		BANK3	
00h	INDF	80h	INDF	100h	INDF	180h	INDF
01h	TMR0	81h	OPTION	101h	TMR0	181h	OPTION
02h	PCL	82h	PCL	102h	PCL	182h	PCL
03h	STATUS	83h	STATUS	103h	STATUS	183h	STATUS
04h	FSR	84h	FSR	104h	FSR	184h	FSR
05h	PORTA	85h	TRISA	105h		185h	
06h	PORTB	86h	TRISB	106h	PORTB	186h	TRISB
07h		87h		107h		187h	
08h		88h		108h		188h	
09h		89h		109h		189h	
0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah	PCLATH
0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh	INTCON
0Ch	PIR1	8Ch	PIE1	10Ch	EEADATA	18Ch	EECON1
0Dh	PIR2	8Dh	PIE2	10Dh	EEADR	18Dh	EECON2
0Eh	TMR1L	8Eh	PCON	10Eh	EEDATH	18Eh	Reserved
0Fh	TMR1H	8Fh	OSCCON	10Fh	EEADRH	18Fh	Reserved
10h	T1CON	90h	OSCTUNE	110h		190h	
11h	TMR2	91h		111h		191h	
12h	T2CON	92h	PR2	112h		192h	
13h	SSPBUF	93h	SSPADD	113h		193h	
14h	SSPCON	94h	SSPSTAT	114h		194h	
15h	CCPR1L	95h		115h		195h	
16h	CCPR1H	96h		116h		196h	
17h	CCP1CON	97h		117h		197h	
18h		98h		118h		198h	
19h		99h		119h		199h	
1Ah		9Ah		11Ah		19Ah	
1Bh		9Bh		11Bh		19Bh	
1Ch		9Ch		11Ch		19Ch	
1Dh		9Dh		11Dh		19Dh	
1Eh	ADRESH	9Eh	ADRESL	11Eh		19Eh	
1Fh	ADCON0	9Fh	ADCON1	11Fh		19Fh	
20h	汎用レジスタ 80Byte	A0h	汎用レジスタ 80Byte	120h	汎用レジスタ 80Byte	1A0h	accesses 20h-7Fh
7Fh	16Byte	F0h	accesses 70h-7Fh	17Fh	accesses 70h-7Fh	1FFh	

BANK0		BANK1	
00h	INDF	80h	INDF
01h	TMR0	81h	OPTION
02h	PCL	82h	PCL
03h	STATUS	83h	STATUS
04h	FSR	84h	FSR
05h	PORTA	85h	TRISA
06h	PORTB	86h	TRISB
07h		87h	
08h	EEDATA	88h	EECON1
09h	EEADR	89h	EECON2
0Ah	PCLATH	8Ah	PCLATH
0Bh	INTCON	8Bh	INTCON
0Ch		8Ch	
0Dh		8Dh	
0Eh		8Eh	
0Fh		8Fh	
10h		90h	
11h		91h	
12h	汎用レジスタ 68Byte	92h	Bank0にマッピング
13h		93h	
14h		94h	
15h		95h	
16h		96h	
17h		97h	
18h		98h	
19h		99h	
1Ah		9Ah	
1Bh		9Bh	
1Ch		9Ch	
1Dh		9Dh	
1Eh		9Eh	
1Fh		9Fh	
20h		A0h	
4Fh		CFh	
50h		D0h	
7Fh		FFh	

BANK0		BANK1		BANK2		BANK3	
00h	INDF	80h	INDF	100h	INDF	180h	INDF
01h	TMR0	81h	OPTION	101h	TMR0	181h	OPTION
02h	PCL	82h	PCL	102h	PCL	182h	PCL
03h	STATUS	83h	STATUS	103h	STATUS	183h	STATUS
04h	FSR	84h	FSR	104h	FSR	184h	FSR
05h	PORTA	85h	TRISA	105h		185h	
06h	PORTB	86h	TRISB	106h	PORTB	186h	TRISB
07h		87h		107h		187h	
08h		88h		108h		188h	
09h		89h		109h		189h	
0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah	PCLATH
0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh	INTCON
0Ch	PIR1	8Ch	PIE1	10Ch		18Ch	
0Dh		8Dh		10Dh		18Dh	
0Eh	TMR1L	8Eh	PCON	10Eh		18Eh	
0Fh	TMR1H	8Fh		10Fh		18Fh	
10h	T1CON	90h		110h		190h	
11h	TMR2	91h		111h		191h	
12h	T2CON	92h	PR2	112h		192h	
13h		93h		113h		193h	
14h		94h		114h		194h	
15h	CCPR1L	95h		115h		195h	
16h	CCPR1H	96h		116h		196h	
17h	CCP1CON	97h		117h		197h	
18h	RCSTA	98h	TXSTA	118h		198h	
19h	TXREG	99h	SPBRG	119h		199h	
1Ah	RCREG	9Ah	EEDATA	11Ah		19Ah	
1Bh		9Bh	EEADR	11Bh		19Bh	
1Ch		9Ch	EECON1	11Ch		19Ch	
1Dh		9Dh	EECON2	11Dh		19Dh	
1Eh		9Eh		11Eh		19Eh	
1Fh	CMCON	9Fh	VRCON	11Fh		19Fh	
20h	汎用レジスタ 80Byte	A0h	汎用レジスタ 80Byte	120h	汎用レジスタ 80Byte		
	16Byte	F0h	accesses 70h-7Fh	17Fh	accesses 70h-7Fh	1FFh	accesses 70h-7Fh

BANK0		BANK1		BANK2		BANK3	
00h	INDF	80h	INDF	100h	INDF	180h	INDF
01h	TMR0	81h	OPTION	101h	TMR0	181h	OPTION
02h	PCL	82h	PCL	102h	PCL	182h	PCL
03h	STATUS	83h	STATUS	103h	STATUS	183h	STATUS
04h	FSR	84h	FSR	104h	FSR	184h	FSR
05h	PORTA	85h	TRISA	105h	WDTCON	185h	
06h	PORTB	86h	TRISB	106h	PORTB	186h	TRISB
07h		87h		107h		187h	
08h		88h		108h		188h	
09h		89h		109h		189h	
0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah	PCLATH
0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh	INTCON
0Ch	PIR1	8Ch	PIE1	10Ch	EEDATA	18Ch	EECON1
0Dh	PIR2	8Dh	PIE2	10Dh	EEADR	18Dh	EECON2
0Eh	TMR1L	8Eh	PCON	10Eh	EEDATH	18Eh	Reserved
0Fh	TMR1H	8Fh	OSCCON	10Fh	EEADRH	18Fh	Reserved
10h	T1CON	90h	OSCTUNE	110h		190h	
11h	TMR2	91h		111h	汎用レジスタ 16Byte	191h	汎用レジスタ 16Byte
12h	T2CON	92h	PR2	112h		192h	
13h	SSPBUF	93h	SSPADD	113h		193h	
14h	SSPCON1	94h	SSPSTAT	114h		194h	
15h	CCPR1L	95h		115h		195h	
16h	CCPR1H	96h		116h		196h	
17h	CCP1CON	97h		117h		197h	
18h	RCSTA	98h	TXSTA	118h		198h	
19h	TXREG	99h	SPBRG	119h		199h	
1Ah	RCREG	9Ah		11Ah		19Ah	
1Bh		9Bh	ANSEL	11Bh		19Bh	
1Ch		9Ch	CMCON	11Ch		19Ch	
1Dh		9Dh	CVRCON	11Dh		19Dh	
1Eh	ADRESH	9Eh	ADRESL	11Eh		19Eh	
1Fh	ADCON0	9Fh	VRCON1	11Fh		19Fh	
20h	汎用レジスタ 96Byte	A0h	汎用レジスタ 80Byte	120h	汎用レジスタ 80Byte		
	16Byte	F0h	accesses 70h-7Fh	17Fh	accesses 70h-7Fh	1FFh	accesses 70h-7Fh