

PIC プログラム解説書(超音波センサボード)

1. 本解説書の目的

この解説書は超音波センサボードに搭載されている PIC の使い方について書かれたものです。超音波センサ上で PIC のプログラムがどのように動いているのかを理解し、超音波センサの修正、改良などを行ってもらうためにこの解説書は作成されました。

2. プログラムの構成

超音波センサに搭載されている PIC は、超音波による距離の測定から CPU への距離データ送信までを行っています。プログラムは主に初期設定・識別コード受信部・識別コード判別部・送信パルス発生部・距離計測部・距離送信部の 6 つに分けられます。

2-1. 初期設定

Program 1 では PIC の機能を使用するためにレジスタの設定を行っています。次にそれぞれの行で何が行われているか説明します。

```
*****
; 初期設定
; 入力: (TRISA)
;     PORTA, 1 (反射波検出部)
;     PORTA, 3 (コンパレータ入力)
;     PORTA, 5 (シリアル通信受信部)
; 出力: (TRISA)
;     PORTA, 0 (超音波発生部)
;     PORTA, 2 (超音波発生部)
;     PORTA, 4 (シリアル通信送信部)
; 内蔵オシレータの誤差修正 (OSCCAL)
; PORTA, 3以外のPORTAをプルアップ (WPUA, OPTION_REG)
; コンパレータの比較電圧を約0.2[V]に設定 (VRCON)
*****
INIT
    BSF     STATUS, RPO    ...①
    CALL   3FFH           ...②
    MOVWF  OSCCAL         ...②
    MOVLW  02AH           ...③
    MOVWF  TRISA          ...③
    MOVLW  037H           ...④
    MOVWF  WPUA           ...④
    CLRF   TRISC          ...⑤
    MOVLW  0A1H           ...⑥
    MOVWF  VRCON          ...⑥
    BCF    STATUS, RPO    ...⑦
    MOVLW  014H           ...⑧
    MOVWF  CMCON          ...⑧
    CLRF   PORTA
```

Program1 :レジスタの初期化

①バンク 1 に切り替える。

② 3FF 番地の内容を呼び出しワーキングレジスタにセット。その後、ワーキングレジスタの値を OSCCAL レジスタにセット。

(②の解説)

超音波センサの PIC は内蔵発振回路を利用して動いています。本来、PIC は外部に振動子を設置し、そこから動作用のクロックを得ています。しかし、PIC16F630 は内部に発振回路を内蔵しており、この装置から動作用クロックを得ることもできます。

PIC に内蔵されている発振回路には誤差があります。その誤差を修正する値がバンク 1 の 3FF 番地に格納されており、この値を OSCCAL レジスタに格納することで誤差を修正することができます。

③ PORTA の入出力設定。

④ WPUA レジスタに 037H をセット。

(④ の解説)

WPUA レジスタでは PORTA の各ピンをプルアップするかしらないかを設定できます。
WPUA レジスタの概要は次のようになっています。

U-0	U-0	R/W-1	R/W-1	U-0	R/W-1	R/W-1	R/W-1
—	—	WPUA5	WPUA4	—	WPUA2	WPUA1	WPUA0
bit 7				bit 0			

- bit7-6 常に 0。
- bit5-4 PORTA,5・PORTA,4 をプルアップするかしらないか。
する場合は 1 を、しない場合は 0 をセット。
- bit3 PORTA,3 は常に入力である為、設定を行えない。常に 0。
- bit2-0 PORTA,2・PORTA,1・PORTA,0 をプルアップするかしらないか。
する場合は 1 を、しない場合は 0 をセット。

WPUA レジスタに 037H を代入することにより、PORTA をすべてプルアップすることになります。

⑤ PORTC をすべて出力に設定。

⑥ VRCON に 0A1H をセット。

(⑥ の解説)

VRCON レジスタではコンパレータ用の比較電圧を発生させる、電圧リファレンス(CV_{REF} module)の設定を行うことができます。

VRCON レジスタの概要は次のようになっています。

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
VREN	—	VRR	—	VR3	VR2	VR1	VR0
bit 7				bit 0			

- bit7 電圧リファレンスを使用するかしらないか。
する場合は 1 を、しない場合は 0 をセット。
- bit6 常に 0。
- bit5 電圧リファレンスのレンジを設定する。
1 なら low range、0 なら high range。
- bit4 常に 0。
- bit3-0 比較電圧の大きさを設定。
VRR=1 なら、比較電圧の値は $(VR3:VR0/24)*V_{DD}$
VRR=0 なら、比較電圧の値は $V_{DD}/4+(VR3:VR0/32)*V_{DD}$

VRCON レジスタに 0A1H を代入することにより、電圧リファレンス(CV_{REF} module)を用いてコンパレータの比較電圧を 0.2V にします。

⑦ バンク 0 に切り替える。

⑧ CMCON レジスタに 014H をセット。

(⑧ の解説)

CMCON レジスタではコンパレータの設定を行えます。
CMCON レジスタの概要は次のようになっています。

U-0	R-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	COUT	—	CINV	CIS	CM2	CM1	CM0
bit 7							bit 0

- bit7 常に0。
- bit6 コンパレータの出力 bit。
CINV=0 なら $V_{IN+} > V_{IN-}$ のときに1。 $V_{IN+} < V_{IN-}$ ときは0。
CINV=1 なら $V_{IN+} < V_{IN-}$ のときに1。 $V_{IN+} > V_{IN-}$ ときは0。
- bit5 常に0。
- bit4 出力の設定。
- bit3 コンパレータの入力設定。(CM2:CM0=110 or 101 の時のみ)
1 なら V_{IN-} と C_{IN+} をつなげる。
0 なら V_{IN-} と C_{IN-} をつなげる。
- bit2-0 コンパレータのモードを設定。図1を参照。

<p>Comparator Reset (POR Default Value - low power) CM2:CM0 = 000</p>	<p>Comparator Off (Lowest power) CM2:CM0 = 111</p>
<p>Comparator without Output CM2:CM0 = 010</p>	<p>Comparator w/o Output and with Internal Reference CM2:CM0 = 100</p>
<p>Comparator with Output and Internal Reference CM2:CM0 = 011</p>	<p>Multiplexed Input with Internal Reference and Output CM2:CM0 = 101</p>
<p>Comparator with Output CM2:CM0 = 001</p>	<p>Multiplexed Input with Internal Reference CM2:CM0 = 110</p>
<p>A = Analog Input, ports always reads '0' D = Digital Input CIS = Comparator Input Switch (CMCON<3>)</p>	

図1 :コンパレータのモード

CMCON レジスタに 014H をセットすることで、リファレンス電圧(CVREF module)と PORTA,1 の電圧を比較することができます。

Program 2 では使用する変数の初期化を行っています。次にそれぞれの行で何が行われているか説明します。

```
*****
; 変数の初期化
;   PECNUM:超音波センサボードの固有識別番号
;   MCODE:超音波センサボードのマスターコード
*****
MAIN
    BSF      PORTA, 4      ...①
    CLRF    PORTC         ...②
    CLRF    COUNT0       ...②
    CLRF    COUNT1       ...②
    CLRF    COUNT2       ...②
    CLRF    COUNT3       ...②
    CLRF    COUNT4       ...②
    CLRF    COUNT5       ...②
    CLRF    COUNT6       ...②
    CLRF    TCOUNT1     ...②
    CLRF    TCOUNT2     ...②
    CLRF    DISCNUM0     ...②
    CLRF    DISCNUM1     ...②
    CLRF    DISCNUM2     ...②
    CLRF    DISCNUM      ...②
    MOVLW   01H          ...③
    MOVWF   PECNUM       ...③
    MOVLW   0F0H         ...④
    MOVWF   MCODE        ...④
```

Program2 :変数の初期化

① PORTA,4 を 1 にセット。

(① の解説)

PORTA,4 はシリアル通信の送信を行っているピンです。シリアル通信ではデータが送信されていない時、常に high の状態になっています。そのため、PORTA,4 の初期値は 1 とします。

② 各変数の初期化。

③ PECNUM に 01H をセット。

(③ の解説)

PECNUM の値が固有識別番号となります。この値を変えることにより、超音波センサの識別コードを設定することができます。

④ MCODE に 0F0H をセット。

(④ の解説)

PECNUM の値がマスターコードとなります。この値を変えることにより、超音波センサのマスターコードを設定することができます。

2-2. 識別番号受信部

Program 3 ではシリアル通信で送られていく識別コードの受信を行っています。次にそれぞれの行で何が行われているか説明します。

```
*****
; 識別番号受信部
*****
LOOP
    BTFSC    PORTA, 5    ...①
    GOTO     LOOP       ...①
    MOVLW    08H        ...②
    MOVWF    COUNT4     ...②
    GOTO     $+1
    NOP
DISCO
    CALL     DUMMYBIT   ...③
    BTFSC    PORTA, 5    ...④
    BSF      STATUS, C  ...④
    BTFSS    PORTA, 5    ...④
    BCF      STATUS, C  ...④
    RRF      DISCNUM     ...⑤
    DECFSZ   COUNT4, F  ...⑥
    GOTO     DISCO      ...⑥
*****
```

Program3 : 識別コード受信部

①シリアル通信で start bit が送られてきたら、ループを抜ける。

(①の解説)

PORTA,5 はシリアル通信の受信を行っているピンです。シリアル通信ではデータが送信されていない時、常に high の状態になっています。start bit は low であるため、PORTA,5 が low になったら識別コードの受信を始めます。

② 8bit の識別コードを受信するために、変数 COUNT4 に 08H をセット。

③ 1bit 分待機する。(104μsec 待機)

(③の解説)

シリアル通信の通信速度は 9600bps を採用しています。この場合、1bit の時間幅は 104μsec となるので、1bit 分の待機は 104μsec となります。

④シリアル通信で送られてきた識別コードを受信。

⑤ DISCNUM に識別コードを保存。

(⑤の解説)

④で受信した識別コードを carry bit にセットしました。その後、RRF を行うことで DISCNUM の 1bit 目に受信した識別コードが格納されます。これを 8 回行うことで、8bit の識別コードを受信することができます。

⑥④～⑤を 8 回繰り返す。

2-3. 識別コード判別部

Program 4 では受信した識別コードを、自分の持っている固有の識別コードと比較して計測を行うか行わないかを判断します。受信した識別コードが自分の識別コードと同じならば計測を開始します。次にそれぞれの行で何が行われているか説明します。

```
*****  
; 識別番号判定部  
*****  
DISC1  
    MOVF    PECNUM,W    ...①  
    MOVWF   NUM        ...①  
    MOVF    DISCNUM,W  ...②  
    XORWF   NUM,F      ...③  
    INCF    NUM,F      ...④  
    DECFSZ  NUM,F      ...⑤  
    GOTO    MASTER     ...⑥
```

Program4 : 識別コード判別部

- ①変数 PECNUM の値(自分の識別コード)を変数 NUM にセット。
- ②変数 DISCNUM の値(受信した識別コード)をワーキングレジスタにセット。
- ③ワーキングレジスタの値と、変数 NUM の値の排他的論理和をとる。

(③ の解説)

排他的論理和を行った場合、それぞれのビットで値が同じならば 0 を、値が異なるならば 1 を返します。ワーキングレジスタと変数 NUM の値が等しければ、それぞれの値の排他的論理和をとると結果は 0 となります。

- ④変数 NUM の値を一つ増やす。
- ⑤変数 NUM の値から 1 を引き 0 なら計測を始める。
- ⑥ 0 でなかったらマスターコードかどうか調べる。マスターコードでもなかったら待機する。

2-4. 送信パルス発生部

Program 5 では送信パルスの発生を行っています。送信パルスの周波数は 40kHz です。次にそれぞれの行で何が行われているか説明します。

```
*****  
; 送信パルス発生部  
*****  
IPULSE  
    BSF     PORTC,1    ...①  
    MOVLW  06H        ...②  
    MOVWF  COUNT4     ...②  
  
PULSE  
    CALL   F40K       ...②  
    DECFSZ COUNT4,F  ...②  
    GOTO   PULSE     ...②
```

Program5 : 送信パルス発生部

- ①送信パルスの発生を LED で確認。
- ② COUNT4 にパルス数をセット(ここでは 06H)。その後パルスを発生させます。

Program 6 では超音波の回り込み回避を行っています。超音波による距離測定では、超音波を発生させ、物体に当たって跳ね返ってきた超音波を検出し、距離の測定を行っています。しかし、超音波の送信器から受信器へ、直接入っていく超音波も存在します。この超音波による誤動作を防ぐために、回り込み回避を行います。次にそれぞれの行で何が行われているか説明します。

```
*****  
; 回り込み回避  
*****  
    MOVLW  015H      ...①  
    MOVWF  COUNT4    ...①  
  
WAIT15  
    CALL   WAIT58    ...②  
    DECFSZ COUNT4,F  ...②  
    GOTO   WAIT15    ...②
```

Program6 : 送信パルス発生部

- ①変数 COUNT4 に 015H をセット。
- ② 58μsec のプログラムを 21 回繰り返す。

2-5. 距離計測部

Program 7 では距離計測が行われています。超音波を送信してから受信するまでの時間を計り、距離を計算しています。次にそれぞれの行で何が行われているか説明します。

```
*****
; 距離計測部
; 1mの距離を往復するのにかかる時間(音波)=5.82[msec]
; 1cmの距離を往復するのにかかる時間(音波)=58.2[usec]
*****
MOV LW    02H      ...①
MOV WF    COUNT5   ...①
DIS
MOV LW    0FFH     ...①
MOV WF    COUNT4   ...①

DISTANCE
BTFSC    CMCON, COUT ...②
GOTO     ANALYZE   ...②
CALL     WAIT58    ...③
INCF     TCOUNT1, F ...③
GOTO     DISTANCE  ...③
INCF     TCOUNT2  ...③

DISTANCEL
DECFSZ   COUNT4, F ...④
GOTO     DISTANCE  ...④
DECFSZ   COUNT5, F ...④
GOTO     DIS       ...④
MOV LW    0FBH     ...⑤
MOV WF    TCOUNT1 ...⑤
MOV LW    026H     ...⑤
MOV WF    TCOUNT2 ...⑤

ANALYZE
```

Program7 : 距離計測部

- ①変数 COUNT5 に 02H を、変数 COUNT4 に 0FFH をセット。
- ② CMCON レジスタの COUT ビットが 1 なら計測を終わる。

(②の解説)

CMCON レジスタの COUT ビットはコンパレータの出力となっています。超音波を受信するとコンパレータは 1 を出力するので、COUT ビットが 1 になった時に計測を完了します。

- ③ 58 μ sec ごとに変数 TCOUNT1 に 1 を加える。TCOUNT1 があふれたら変数 TCOUNT2 に 1 を加える。

(③の解説)

超音波が 1cm の距離を往復するのにかかる時間は約 58 μ sec です。58 μ sec ごと TCOUNT1 に 1 を加えることで、TCOUNT1 の値は計測距離(cm)となります。しかし、1つの変数で蓄えておけるデータ量は 255 までであるため、変数が 1 つだけでは 2m55cm までしか測ることができません。そこで、あふれた分を TCOUNT2 に入れることで、655m35cm まで対応することができます。

- ④変数 COUNT4、COUNT5 が 0 になったら計測を終わる。

(④の解説)

変数 COUNT4 と COUNT5 が 0 になるときは、超音波がいくら待っても戻ってこない時です。この場合は計測エラーとなります。

- ⑤ TCOUNT1 に 0FBH を、TCOUNT2 に 026H をセット。

2-6. 距離送信部

Program 8 ではシリアル通信を用いて、CPU ボードへ距離データを送信しています。送信に使用するポートは PORTA,4 です。次にそれぞれの行で何が行われているか説明します。

```
*****
; 距離送信部 (シリアル通信)
*****
    MOVLW    08H      ...①
    MOVWF   COUNT4   ...①
    MOVWF   COUNT5   ...①
    MOVWF   COUNT6   ...①
    MOVLW   00H      ...②
    CALL    DATABIT  ...②
    NOP
    GOTO    $+1

CATCH1
    RRF     TCOUNT1 ...③
    BTFSS  STATUS,C  ...③
    MOVLW  01H      ...③
    BTFSC  STATUS,C  ...③
    MOVLW  011H     ...③
    CALL   DATABIT  ...③
    DECFSZ COUNT5,F  ...④
    GOTO   CATCH1   ...④
    NOP
    BSF   PORTA,4   ...⑤
    CALL  DUMMYBIT  ...⑤
    CALL  DUMMYBIT  ...⑤
    CALL  DUMMYBIT  ...⑤
    MOVLW 00H      ...⑥
    CALL  DATABIT  ...⑥
    NOP
    GOTO  $+1

CATCH2
    RRF     TCOUNT2 ...⑦
    BTFSS  STATUS,C  ...⑦
    MOVLW  01H      ...⑦
    BTFSC  STATUS,C  ...⑦
    MOVLW  011H     ...⑦
    CALL   DATABIT  ...⑦
    DECFSZ COUNT6,F  ...⑧
    GOTO   CATCH2   ...⑧
    NOP
    BSF   PORTA,4   ...⑨
    GOTO  MAIN
```

Program8 : 距離送信部

- ①変数 COUNT4、COUNT5、COUNT6 に 08H をセット。
- ② start bit を出力。
- ③ TCOUNT1 の値をシリアル通信で送信。
- ④③を 8 回繰り返す。
- ⑤ PORTA,4 を high にし、3bit 分待機。
- ⑥ start bit を出力。
- ⑦ TCOUNT2 の値をシリアル通信で送信。
- ⑧⑦を 8 回繰り返す。
- ⑨ PORTA,4 を high にし、送信を終了する。