INTRODUCTION

This application note is intended for PIC16C7X users with some degree of familiarity with analog system design. The various sections discuss the following topics:

- Commonly used A/D terminology
- How to configure and use the PIC16C71 A/D
- Various ways to generate external reference voltage (VREF)
- Configuring the RA3:RA0 pins

COMMONLY USED A/D TERMINOLOGY

The Ideal Transfer Function

In an A/D converter, an analog voltage is mapped into an N-bit digital value. This mapping function is defined as the transfer function. An ideal transfer is one in which there are no errors or non-linearity. It describes the "ideal" or intended behavior of the A/D. Figure 1 shows the ideal transfer function for the PIC16C7X A/D.

Note that the digital output value is 00h for the analog input voltage range of 0 to 1LSb. In some converters, the first transition point is at 0.5LSb and not at 1LSb as shown in Figure 2. Either way, by knowing the transfer function the user can appropriately interpret the data.

Transition Point

The analog input voltage at which the digital output switches from one code to the next is called the "Transition Point." The transition point is typically not a single threshold, but rather a small region of uncertainty (Figure 3). The transition point is therefore defined as the statistical average of many conversions. Stated differently, it is the voltage input at which the uncertainty of the conversion is 50%.

Code Width

The distance (voltage differential) between two transition points is called the "Code Width." Ideally the Code Width should be 1LSb (Figure 1).
Center of Code Width
The midpoint between two transition points is called the “Center of Code Width” (Figure 3).

**FIGURE 3: TRANSITION POINTS**

Differential Non-Linearity (DNL)
It is the deviation in code-width from 1LSb (Figure 4). The difference is calculated for each and every transition. The largest difference is reported as DNL.

It is important to note that the DNL is measured after the transfer function is normalized to match offset error and gain error.

Note that the DNL cannot be any less than -1LSb. In the other direction, DNL can be >1LSb.

**FIGURE 4: DIFFERENTIAL NON-LINEARITY**

Absolute Error
The maximum deviation between any transition point from the corresponding ideal transfer function is defined as the absolute error. This is how it is measured and reported in the PIC16C7X (Figure 5). The notable difference between absolute error and integral non-linearity (INL) is that the measured data is not normalized for full scale and offset errors in absolute error.

Absolute Error is probably the first parameter the user will review to evaluate an A/D. Sometimes absolute error is reported as the sum of offset, full-scale and integral non-linearity errors.

**Total Unadjusted Error**
Total Unadjusted Error is the same as absolute error. Again, sometimes it is reported as the sum of offset, full-scale and integral non-linearity errors.

**No Missing Code**
No missing code implies that as the analog input voltage is gradually increased from zero to full scale (or vice versa), all digital codes are produced. Stated otherwise, changing analog input voltage from one quantum of the analog range to the next adjacent range will not produce a change in the digital output by more than one code count.

**Monotonic**
Monotonicity guarantees that an increase (or decrease) in the analog input value will result in an equal or greater digital code (or less). Monotonicity does not guarantee that there are no missing codes. However, it is an important criterion for feedback control systems. Non-monotonicity may cause oscillations in such systems.

The first derivative of a monotonic function always has the same sign.
Ratiometric Conversion

Ratiometric Conversion is the A/D conversion process in which the binary result is a ratio of the supply voltage or reference voltage, the latter being equal to full-scale value by default. The PIC16C7X is a ratiometric A/D converter where the result depends on VDD or VREF.

In some A/Ds, an absolute reference is provided resulting in “absolute conversion”.

Sample and Hold

In sample and hold type A/D converters, the analog input has a switch (typically a FET switch in CMOS) which is opened for a short duration to capture the analog input voltage onto an on-chip capacitor. Conversion is typically started after the sampling switch is closed.

Track and Hold

Track and Hold is basically the same as sample and hold, except the sampling switch is typically left on. Therefore the voltage on the on-chip holding capacitor “tracks” the analog input voltage. To begin a conversion, the sampling switch is closed.

The PIC16C7X A/D falls in this category.

Sampling Time

Sampling Time is the time required to charge the on-chip holding capacitor to the same value as is on the analog input pin. The sampling time depends on the magnitude of the holding capacitor and the source impedance of the analog voltage input.

Offset Error (or Zero Error)

Offset Error is the difference between the first actual (measured) transition point and the first ideal transition point as shown in Figure 6. It can be corrected (by the user) by subtracting the offset error from each conversion result.

Full Scale Error (or Gain Error)

Full Scale Error is the difference between the ideal full scale and the actual (measured) full scale range (Figure 7). It is also called gain error, because the error changes the slope of the ideal transfer function creating a gain factor. It can be corrected (by the user) by multiplying each conversion result by the inverse of the gain.

Integral Non-Linearity (INL), or Relative Error

The deviation of a transition point from its corresponding point on the ideal transfer curve is called “Integral Non-Linearity” (Figure 8). The maximum difference is reported as the INL of the converter.

It is important to note that Full Scale Error and the Offset Error are normalized to match end transition points before measuring the INL.
HOW TO USE THE PIC16C71 A/D

The A/D in the PIC16C71 is easy to set up and use. There are a few considerations:

1. Select either VDD or VREF as reference voltage. (More on using VREF input later)
   Select A/D conversion clock (TAD): 2TOSC, 8TOSC, TOSC or TRC (internal RC clock). For the first three options, make sure that TAD ≥ 2.0 µs. If deterministic conversion time is required, select TOSC time-base. If conversion during SLEEP is required, select TRC.

2. Channel Selection: If only one A/D channel is required, program the ADCON1 register to 03h. This configures the A/D pins as digital I/O. If multiple channels are required, prior to each conversion the new channel must be selected.

3. Sampling and Conversion: After a new channel is selected, a minimum amount of sampling time must be allowed before the GO/DONE bit in ADCON0 is set to begin conversion. Once conversion begins, it is OK to select the next channel, but sampling does not begin until current conversion is complete. Therefore, it is always necessary to ensure the minimum sampling time is provided for:
   i) after a conversion
   ii) after a new channel is selected
   iii) after A/D is turned on (bit ADON = 1)

4. Reading Result: Completion of a conversion can be determined by polling the GO/DONE bit (cleared), or polling flag bit ADIF (set), or waiting for an ADIF interrupt.

Additional tips:

a) Do not set bits GO/DONE and ADON in the same instruction. First, turn the A/D on by setting bit ADON. Then allow at least 5 µs before conversion begins (setting the GO/DONE bit), longer if sampling time requirement is not met within 5 µs.

b) Aborting a conversion: A conversion can be aborted by clearing bit GO/DONE. The A/D converter will stop conversion and revert back to sampling state.

c) Using the ADRES register as a normal register: The A/D only writes to the ADRES register at the end of a conversion. Therefore, it is possible to use the ADRES register as a normal file register between conversions and when A/D is off.

The following four examples provide sample code on using the A/D module.

EXAMPLE 1: HOW TO DO A SAMPLE A/D CONVERSION

; InitializeAD, initializes and sets up the A/D hardware.
; Always ch2, internal RC OSC.
InitializeAD
  bsf STATUS, 5       ; select Bank1
  movlw b'00000000'     ; select RA3-RA0
  movwf ADCON1          ; as analog inputs
  bcf STATUS, 5       ; select Bank0
  movlw b'11010001'     ; select: RC osc, ch2...
  movwf ADCON0          ; turn on A/D
Convert   call sample-delay    ; provide necessary sampling time
;  bsf ADCON0, 2       ; start new A/D conversion
loop
  btfsc ADCON0, 2       ; A/D over?
  goto loop            ; no then loop
;  movf adres, w        ; yes then get A/D value

A detailed code listing is provided in Appendix A.
EXAMPLE 2: SEQUENTIAL CHANNEL CONVERSIONS

; InitializeAD, initializes and sets up the A/D hardware.
; Select ch0 to ch3 in a round robin fashion, internal RC OSC.
; Load results in 4 consecutive addresses starting at ADTABLE (10h)

; InitializeAD
bsf STATUS, RP0 ; select Bank1
movlw b'00000000' ; select RA3-RA0
movwf ADCON1 ; as analog inputs
bcf STATUS, RP0 ; select Bank0
movlw b'11000001' ; select: RC osc, ch0...
movwf ADCON0 ; turn on A/D
movlw ADTABLE ; point fsr to top of...
movwf FSR ; table

; new_ad call sample_delay ; provide necessary sampling time
bsf ADCON0, GO ; start new A/D conversion

loop
btfsc ADCON0, GO ; A/D over?
goto loop ; no then loop

; movf adres, w ; yes then get A/D value
movwf 0 ; load indirectly
movlw 4 ; select next channel
addwf ADCON0 ;
bcf ADCON0, ADIF ; reset interrupt flag bit.

; increment pointer to correct table offset.
crflw temp ; clear temp register
btfsc ADCON0, CH50 ; test lsb of channel select
bsf temp, 0 ; set if ch1 selected
btfsc ADCON0, CH51 ; test msb of channel select
bsf temp, 1 ;
movlw ADTABLE ; get table address
addwf temp, w ; add with temp
movwf FSR ; move into indirect

; goto new_ad

; A detailed code listing is provided in Appendix B.
EXAMPLE 3: SAMPLE INTERRUPT HANDLER FOR THE A/D

```
org 0x00
goto start
org 0x04
goto service_ad ; interrupt vector 

start
movlw b'00000000' ;init I/O ports
movwf PORT_B
tris PORT_B

call InitializeAD
update
bcf flag, adover ; reset software A/D flag

call SetupDelay ; setup delay >= 10uS.
bcf ADCON0, adif ; reset A/D int flag (ADIF
bsf ADCON0, go ; start new A/D conversion
bsf INTCON, gle ; enable global interrupt

loop
btfsc flag, adover ; A/D over?
goto update ; yes start new conv.
goto loop ; no then keep checking

; InitializeAD, initializes and sets up the A/D hardware.
; select ch0 to ch3, RC OSC., a/d interrupt.
InitializeAD
bsf STATUS, RP0 ; select Bank1
movlw b'00000000' ; select RA0-RA3...
movwf ADCON1 ; as analog inputs
bcf STATUS, RP0 ; select Bank0
clf INTCON ; clr all interrupts
bsf INTCON, ADIE ; enable A/D int.
movlw b'11010001' ; select: RC osc, ch2...
movwf ADCON0 ; turn on A/D
return

; service_ad
btfsc ADCON0, ADIF ; A/D interrupt?
retfie ; no then ignore
movf ADRES, W ; get A/D value
return ; do not enable int
```

A detailed code listing is provided in Appendix C.
EXAMPLE 4: CONVERSIONS DURING SLEEP MODE

; InitializeAD, initializes and sets up the A/D hardware.
; Select ch0 to ch3, internal RC OSC.
; While doing the conversion put unit to sleep. This will
; minimize digital noise interference.
; Note that A/D's RC osc. has to be selected in this instance.

InitializeAD
bsf STATUS, RP0 ; select Bank1
movlw b'00000000' ; select RA0-RA3...
movwf ADCON1 ; as analog inputs
bcf STATUS, RP0 ; select Bank0
movlw b'11000001' ; select: RC osc, ch0...
movwf ADCON0 ; turn on A/D & ADIE
movlw ADTABLE ; point fsr to top of...
movwf FSR ; table

; new_ad
bsf ADCON0, GO ; start new A/D conversion
sleep ; goto sleep
; when A/D is over program will continue from here
; movf ADRES, w ; get A/D value

A detailed code listing is provided in Appendix D.
USING EXTERNAL REFERENCE VOLTAGE

When using the external reference voltage, keep in mind that any analog input voltage must not exceed VREF.

An inexpensive way to generate VREF is by employing a zener diode (Figure 9). Most common zener diodes offer 5% accuracy. Reverse bias current may be as low as 10 μA. However, larger currents (1 mA - 20 mA) are recommended for stability, as well as lower impedance of the VREF source.

FIGURE 9: LOW COST VOLTAGE REFERENCE

A simple and cost-effective method to generate VREF is by using a zener diode (Figure 9). Most common zener diodes offer 5% accuracy. Reverse bias current may be as low as 10 μA. However, larger currents (1 mA - 20 mA) are recommended for stability, as well as lower impedance of the VREF source.

POWER MANAGEMENT IN USING VREF

In power sensitive applications, the user may turn on a VREF generator using another I/O pin (Figure 10). Drive a ‘1’ on pin RB1, in this example, when using the A/D. Drive a ‘0’ on pin RB1 when not using the A/D converter.

Note that this way RB1 is not floating. Even if VREF decays to some intermediate voltage, it will not cause the input buffer on RB1 to draw current.

Alternately, use RA0, RA1 or RA2 pin to supply the current instead of RB1. Configure the RA pin as analog (this will turn off its input buffer). Then use it as a digital output (Figure 11).

FIGURE 10: POWER-SENSITIVE APPLICATIONS #1

Finally, various reference voltage generator chips (typically using on-chip band-gap reference) are available. They are more accurate.

TABLE 1: ZENERS AND REFERENCE GENERATORS

<table>
<thead>
<tr>
<th>Zeners</th>
<th>Vz</th>
<th>Tolerance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1N746</td>
<td>3.3V</td>
<td>±5%</td>
</tr>
<tr>
<td>1N747</td>
<td>3.6V</td>
<td>±5%</td>
</tr>
<tr>
<td>1N748</td>
<td>3.9V</td>
<td>±5%</td>
</tr>
<tr>
<td>1N749</td>
<td>4.3V</td>
<td>±5%</td>
</tr>
<tr>
<td>1N750</td>
<td>4.7V</td>
<td>±5%</td>
</tr>
<tr>
<td>1N751</td>
<td>5.1V</td>
<td>±5%</td>
</tr>
<tr>
<td>1N752</td>
<td>5.6V</td>
<td>±5%</td>
</tr>
<tr>
<td>AD580 (Maxim)</td>
<td>2.5V</td>
<td>±3% to ±0.4%</td>
</tr>
<tr>
<td>LM385</td>
<td>2.5V</td>
<td>±1.5%</td>
</tr>
<tr>
<td>LM1004</td>
<td>2.5V</td>
<td>±1.2%</td>
</tr>
<tr>
<td>LT1009 (LIN. Tech.)</td>
<td>2.5V</td>
<td>±0.2%</td>
</tr>
<tr>
<td>LT1019 (LIN. Tech.)</td>
<td>5.0V</td>
<td>±0.2%</td>
</tr>
<tr>
<td>LT1021 (LIN. Tech.)</td>
<td>5.0V</td>
<td>±0.05% to ±1%</td>
</tr>
<tr>
<td>LT1029 (LIN. Tech.)</td>
<td>5.0V</td>
<td>±0.2% to ±1%</td>
</tr>
</tbody>
</table>
VREF IMPEDANCE AND CURRENT SUPPLY REQUIREMENTS

Ideally, VREF should have as low a source impedance as possible. Referring to Figure 9, VREF source impedance \( R \). However, smaller \( R \) increases current consumption. Since VREF is used to charge capacitor arrays inside the A/D converter and the holding capacitor, CHOOLD = 51 pF, the following guideline should be met:

\[
T_{AD} = 6(1k + R)51.2\, pF + 1.677\, \mu s
\]

\( T_{AD} \) = conversion clock. For \( T_{AD} = 2\, \mu s \) and for \( CHOOLD = 50\, pF, VREF = 50\, \Omega \).
For VREF impedance higher than this, the conversion clock (\( T_{AD} \)) should be increased appropriately.

**FIGURE 11: POWER-SENSITIVE APPLICATIONS #2**

To achieve a low source impedance when using a Zener diode, a voltage follower circuit is recommended. This is shown in Figure 12.

**FIGURE 12: VOLTAGE FOLLOWER CIRCUIT**

CONFIGURING PORTA INPUTS AS ANALOG OR DIGITAL

Two bits in the ADCON1 register, PCFG1 and PCFG0, control how pins RA3:RA0 are configured.

When any of these pins are selected as analog:
- The digital input buffer is turned off to save current (Figure 13). Reading the port will read this pin as '0'.
- The TRIS bit still controls the output buffer on this pin. So, normally the TRIS bit will be set (input).
- However, if the TRIS bit is cleared, then the pin will output whatever is in the data latch.

When any of these pins are selected as digital:
- The analog input still directly connects to the A/D and therefore the pin can be used as analog input.
- The digital input buffer is not disabled.
The user has, therefore, great flexibility in configuring these pins.

**TABLE 2: MAXIMUM RATE OF CONVERSION / BIT**

<table>
<thead>
<tr>
<th>RVREF</th>
<th>( T_{AD} ) (Max)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1k</td>
<td>2.29 ( \mu s )</td>
</tr>
<tr>
<td>5k</td>
<td>3.52 ( \mu s )</td>
</tr>
<tr>
<td>10k</td>
<td>5.056 ( \mu s )</td>
</tr>
<tr>
<td>50k</td>
<td>16.66 ( \mu s )</td>
</tr>
<tr>
<td>100k</td>
<td>32.70 ( \mu s )</td>
</tr>
</tbody>
</table>

Assumes no external capacitors
CURRENT CONSUMPTION THROUGH INPUT BUFFER

A CMOS input buffer will draw current when the input voltage is near its threshold (Figure 14).

In power-sensitive applications, the RA pins, when used as analog inputs, should be configured as “analog” to avoid unintended power drain.

Other considerations and tips:
1. If possible, avoid any digital output next to analog inputs.
2. Avoid digital inputs that switch frequently (e.g., clocks) next to analog inputs.
3. If VREF is used, then ensure that no analog pin being sampled exceeds VREF.

SUMMARY

The PIC16C71 A/D converter is simple to use. It is versatile and has low power consumption.

VTH = Threshold of the inverter
VTN = Device threshold of NMOS pull-down
-VTP = Device threshold of PMOS pull-up
I = On-current (or through current) of the inverter
IMAX = Maximum on-current occurs when VIN = VTH. Value of IMAX depends on the sizes of the devices. The larger the devices, the faster the input buffer, and the larger the value of IMAX. Typically, IMAX is 0.2 mA – 1 mA.
APPENDIX A: SINGLE CHANNEL A/D (SAD)

MPASM 01.40 Released    SAD.ASM   1-16-1997   15:22:04    PAGE 1

LOC  OBJECT CODE  LINE  SOURCE TEXT
    VALUE

00001 ;TITLE   "Single channel A/D (SAD)"
00002 ;This program is a simple implementation of the PIC16C71's
00003 ;A/D. 1 Channel is selected (CHO).
00004 ;The A/D is configured as follows:
00005 ; Vref = +5V internal.
00006 ; A/D Osc. = internal RC
00007 ; A/D Channel = CH0
00008 ;Hardware for this program is the PICDEM1 board.
00009 ;
00010 ;
00011 ;       Program:          SAD.ASM
00012 ;       Revision Date:
00013 ;                         1-14-97      Compatibility with MPASMWIN 1.40
00014 ;
00015 ;
00016 LIST P=16C71
00017 ERRORLEVEL -302
00018 ;
00019 include "p16c71.inc"
00020 LIST
00021 TEMP    EQU     10h
00022 adif    equ     1
00023 adgo    equ     2
00024 ;
00025 ORG     0x00
00026 ;
00027 ;
00028 start
00029 ;
00030 org     0x04
00031 goto    service_int   ;interrupt vector
00032 ;
00033 ;
00034 org     0x10
00035 start
00036 movlw B'00000000'   ;set port b as
00037 movwf PORTB          ;all outputs
00038 ;
00039 BSF     STATUS, RP0     ; Bank1
00040 MOVWF   TRISB           ; PortB as outputs
00041 BCF     STATUS, RP0     ; Bank0
00042 ;
00043 call    InitializeAD
00044 update
00045 movf  ADRES,W         ;get a/d value
00046 movwf PORTB          ;output to port b
00047 call    SetupDelay    ;setup time >= 10uS.
00048 bcf    ADCON0,adif    ;clear int flag
00049 bsf    ADCON0,adgo    ;start new conversion
00050 loop
00051 btfsb  ADCON0,adif    ;a/d done?
00052 goto    update          ;yes then update new value.
001D 281B 00053 goto loop ;no then keep checking
00054 ;
00055 ;no interrupts are enabled, so if the program ever reaches here,
00056 ;it should be returned with the global interrupts disabled.
001E 00057 service_int
001E 0008 00058 return ;do not enable global.
00059 ;
00060 ;
00061 ;
00062 ;InitializeAD, initializes and sets up the A/D hardware.
00063 ;Select ch0 to ch3 as analog inputs, fosc/2 and read ch3.
00064 ;
001F 00065 InitializeAD
001F 1683 00066 bsf STATUS,5 ;select Bank1
0020 3000 00067 movlw B'00000000' ;select ch0-ch3...
0021 0088 00068 movwf ADCON1 ;as analog inputs
0022 1283 00069 bcf STATUS,5 ;select Bank0
0023 30C1 00070 movlw B'11000001' ;select:RC,ch0..
0024 0088 00071 movwf ADCON0 ;turn on A/D.
0025 0189 00072 clr ADRES ;clr result reg.
0026 0008 00073 return
00074 ;
00075 ;This routine is a software delay of 10uS for the a/d setup.
00076 ;At 4Mhz clock, the loop takes 3uS, so initialize TMPP with
00077 ;a value of 3 to give 9uS, plus the move etc should result in
00078 ;a total time of > 10uS.
0027 00079 SetupDelay
0027 3003 00080 movlw .3
0028 0090 00081 movwf TEMP
0029 00082 SD
0029 0B90 00083 decfsz TEMP, F
002A 2829 00084 goto SD
002B 0008 00085 return
00086
00087
00088 END

MEMORY USAGE MAP ('X' = Used, '-' = Unused)

0000 : X---X--------- XXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXX----- ---------------

All other memory blocks unused.

Program Memory Words Used:    30
Program Memory Words Free:   994

Errors : 0
Warnings : 0 reported, 0 suppressed
Messages : 0 reported, 2 suppressed
APPENDIX B: SLPAD.ASM

MPASM 01.40 Released  SLPAD.ASM  1-16-1997  15:22:32  PAGE 1

LOC  OBJECT CODE     LINE SOURCE TEXT
VALUE

00001
00002 ;TITLE   "A/D in Sleep Mode"
00003 ;This program is a simple implementation of the PIC16C71's
00004 ;A/D feature. This program demonstrates
00005 ;how to do a a/d in sleep mode on the PIC16C71.
00006 ;The A/D is configured as follows:
00007 ;       Vref = +5V internal.
00008 ;       A/D Osc. = internal RC
00009 ;       A/D Interrupt = OFF
00010 ;       A/D Channels = ch 0
00011 ;
00012 ;The ch0 A/D result is displayed as a 8 bit binary value
00013 ;on 8 leds connected to port b. Hardware used is that of
00014 ;the PICDEMO board.
00015 ;
00016 ;
00017 ; Program:          SLPAD.ASM
00018 ; Revision Date: 1-14-97   Compatibility with MPASMWIN 1.40
00019 ;
00020 ;
00021 ;
00022 LIST P=16C71
00023 ERRORLEVEL -302
00024 ;
00025 include "p16c71.inc"
00026 ;
00027 LIST P16C71.INC  Standard Header File, Version 1.00 Microchip Technology
00028 LIST
00029 LIST
00030 ;
00031 ;
00032 ORG 0x00
00033 ;
00034 ;
00035 goto start
00036 ;
00037 org 0x04
00038 goto service_int ;interrupt vector
00039 ;
00040 ;
00041 org 0x10
00042 start
00043 movlw B'00000000' ;make port b all
00044 movwf PORTB ;outputs.
00045 ; triw PORTB ; /
00046 BSF STATUS, RPO ; Bank1
00047 MOVWF TRISB ; PortB as outputs
00048 BCF STATUS, RPO ; Bank0
00049 ;
00050 call InitializeAD
00051 update

Please check the Microchip BBS for the latest version of the source code. Microchip's Worldwide Web Address: www.microchip.com; Bulletin Board Support: MCHIPBBS using CompuServe® (CompuServe membership not required).
0016 0809 00052  movf ADRES,W
0017 0086 00053  movwf PORTB ;save in table
0018 2027 00054  call SetupDelay ;
0019 1088 00055  bcf ADCON0,adif ;clr a/d flag
001A 1508 00056  bsf ADCON0,adgo ;start new a/d conversion
00057 ;
001B 0063 00058  sleep
001C 2816 00059  goto update ;wake up and update
00060 ;
001D 0061 service_int
001D 0008 00062  return ;do not enable int
00063 ;
00064 ;InitializeAD, initializes and sets up the A/D hardware.
001E 00065 InitializeAD
001E 1683 00066  bsf STATUS,5 ;select Bank1
001F 3000 00067  movlw B'00000000' ;select ch0-ch3...
0020 0088 00068  movwf ADCON1 ;as analog inputs
0021 1283 00069  bcf STATUS,5 ;select Bank0
0022 30C1 00070  movlw B'11000001' ;select:internal RC, ch0.
0023 0088 00071  movwf ADCON0 ;turn on a/d
0024 018B 00072  clrf INTCON ;clear all interrupts
0025 170B 00073  bsf INTCON,ADIE ;enable a/d
0026 0008 00074  return
00075 ;
00076 ;This routine is a software delay of 10uS for the a/d setup.
00077 ;At 4Mhz clock, the loop takes 3uS, so initialize TEMP with
00078 ;a value of 3 to give 9uS, plus the move should result in
00079 ;a total time of > 10uS.
0027 00080 SetupDelay
0027 3003 00081  movlw .3
0028 0090 00082  movwf TEMP
0029 0083 SD
0029 0B90 00084  decfsz TEMP, F
002A 2829 00085  goto SD
002B 0008 00086  return
00087
00088 ;
00089
00090 END

MEMORY USAGE MAP ('X' = Used, '-' = Unused)

0000 : X---X--------- XXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXX----- -----------

All other memory blocks unused.

Program Memory Words Used:    30
Program Memory Words Free:   994

Errors :  0
Warnings :  0 reported,  0 suppressed
Messages :  0 reported,  2 suppressed
APPENDIX C: INTAD.ASM

MPASM 01.40 Released
INTAD.ASM 1-16-1997 15:21:10 PAGE 1

LOC OBJECT CODE LINE SOURCE TEXT
VALUE

00001
00002 ;TITLE "Single channel A/D with interrupts"
00003 ;This program is a simple implementation of the PIC16C71's
00004 ;A/D. 1 Channel is selected (CH0). A/D interrupt is turned on,
00005 ;hence on completion of a/d conversion, an interrupt is generated.
00006 ;The A/D is configured as follows:
00007 ;  Vref = +5V internal.
00008 ;  A/D Osc. = internal RC Osc.
00009 ;  A/D Interrupt = On
00010 ;  A/D Channel = CH0
00011 ;
00012 ;The A/D result is displayed as a 8 bit value on 8 leds connected
00013 ;to port b. Hardware setup is the PICDEMO board.
00014 ;
00015 ;
00016 ;  Program:          INTAD.ASM
00017 ;  Revision Date:    1-14-97      Compatibility with MPASMWIN 1.40
00018 ;
00019 ;
00020 ;
00021 LIST P=16C71
00022 ERRORLEVEL -302
00023 ;
00024 include "p16c71.inc"
00025 LIST
00026 ; P16C71.INC  Standard Header File, Version 1.00  Microchip Technology
00027 LIST
00028 ;
00029 00030 flag    equ     10
00031 00032 TEMP    equ     11
00032 00033 adover  equ     0
00033 00034 adif    equ     1
00034 00035 adgo    equ     2
00035 00036 adie    equ     6
00036 00037 gie     equ     7
00037 00038 rp0     equ     5
00038 00039
00039 00040 org     0x00
00040 00041 goto    start
00041 00042 service_ad      ;interrupt vector
00042 00043 ;
00043 00044 org     0x10
00044 00045 start
00045 00046 movlw  B'000000000' ;init i/o ports
00046 00047 movwf  PORTB
00047 00048 tris  PORTB
00048 00049 BSF  STATUS, RP0 ; Bank1
00049 00050 MOVWF  TRISB ; PortB as outputs
00050 00051 BCF  STATUS, RP0 ; Bank0
00051 00052 ;

Please check the Microchip BBS for the latest version of the source code. Microchip's Worldwide Web Address: www.microchip.com; Bulletin Board Support: MCHIPBBS using CompuServe® (CompuServe membership not required).
InitializeAD, initializes and sets up the A/D hardware.

select ch0 to ch3, RC OSC., a/d interrupt.

InitializeAD

bsf STATUS, rp0 ; select Bank1
movlw B'00000000' ; select ch0-ch3...
movwf ADCON1 ; as analog inputs
bcf STATUS, rp0 ; select Bank0
clrf INTCON ; clear all interrupts
bsf INTCON, adie ; enable a/d int.
movlw B'11000001' ; select: RC osc, ch0...
movwf ADCON0 ; turn on a/d

This routine is a software delay of 10µS for the a/d setup.
At 4MHz clock, the loop takes 3µS, so initialize TEMP with
a value of 3 to give 9µS, plus the move should result in
a total time of > 10µS.

SetupDelay

movlw .3
movwf TEMP

decfsz TEMP, F

goto SD

return

return

return

END

MEMORY USAGE MAP ('X' = Used, '-' = Unused)

0000 : X---X---------- XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XX----------

All other memory blocks unused.

Program Memory Words Used: 36
Program Memory Words Free: 988

Errors : 0
Warnings : 0 reported, 0 suppressed
Messages : 0 reported, 2 suppressed
APPENDIX D: MULTAD.ASM

MPASM 01.40 Released MULTAD.ASM 1-16-1997 15:21:41 PAGE 1

LOC OBJECT CODE LINE SOURCE TEXT
VALUE

00001 ;TITLE "A/D using Multiple Channels"
00002 ;This program is a simple implementation of the PIC16C71's
00003 ;A/D feature. This program demonstrates
00004 ;how to select multiple channels on the PIC16C71.
00005 ;The A/D is configured as follows:
00006 ; Vref = +5V internal.
00007 ; A/D Osc. = internal RC osc.
00008 ; A/D Interrupt = Off
00009 ; A/D Channels = all in a "Round Robin" format.
00010 ; A/D results are stored in ram locations as follows:
00011 ; ch0 --> ADTABLE + 0
00012 ; ch1 --> ADTABLE + 1
00013 ; ch2 --> ADTABLE + 2
00014 ; ch3 --> ADTABLE + 3
00015 ;
00016 ;The ch0 A/D result is displayed as a 8 bit value on 8 leds
00017 ;connected to port b.
00018 ;Hardware: PICDEMO board.
00019 ; Stan D'Souza 7/6/93.
00020 ;
00021 ; Program: MULTAD.ASM
00022 ; Revision Date: 1-14-97 Compatibility with MPASMWIN 1.40
00023 ;
00024 ;
00025 ;
00026 LIST P=16C71
00027 ERRORLEVEL -302
00028 ;
00029 include "p16c71.inc"
00030 ;
00031 TEMP EQU 10h
00032 adif equ 1
00033 adgo equ 2
00034 ;
00035 ch2 equ 6
00036 ch3 equ 7
00037 flag equ 0C
00038 ADTABLE equ 20
00039 ;
00040 ORG 0x00
00041 ;
00042 goto start
00043 ;
00045 org 0x04
00046 goto service_int ;interrupt vector
00047 ;
00048 ;
00049 org 0x10
00050 start
00051 movlw B'00000000' ;make port b
; as all outputs
movwf PORTB

;       
tris PORTB

;       /
BSF STATUS, RP0

; PortB as outputs
BCF STATUS, RP0

//
call InitializeAD

;save in table
movf ADRES,W

;chk if ch0
movlw ADTABLE

; Bank1
subwf FSR,W

; Bank0
btfss STATUS,Z

;else do next channel
goto NextAd

;get a/d value
movf ADRES,W

;output to port b
call NextChannel

call SetupDelay

;clear flag
bsf ADCON0,adgo

;start new a/d conversion
subwf FSR,W

; a/d done?
addwf FSR,W

;allocate new address
movwf PORTB

;do not enable int
return

; select next channel
call NextChannel

;set up > - 10uS
bcf ADCON0,adif

; clear flag
bsf ADCON0,adgo

; Bank0
subwf FSR,W

; select:fosc/2, ch0.
movlw B'11000001'

;get top of table address
addwf TEMP,W

;wait till done
goto loop

;get channel offset
addwf ADCON0, F

; add to conf. reg.
clf

;clear any carry over
movlw 0x08

; Bank1
addwf ADCON0, F

; Bank0
bcf ADCON0,5

; Bank0
bcf ADCON0,4

; Bank0
bcf ADCON0,3

; Bank0
bcf ADCON0,2

; Test lsb of chnl select
bsf TEMP,5

; Test ch0 or ch3
bsf TEMP,3

; Test msb of chnl select
bsf TEMP,1

; Test ch0 or ch2
bsf TEMP,0

; get top of table
movlw ADTABLE

; load into indirect reg
addwf TEMP,W

; allocate new address
movwf FSR

; Bank3
return

; Bank3
addwf TEMP,W

; allocate new address
movwf PORTB

; Bank3
addwf TEMP,W

; allocate new address
movwf PORTB

; Bank3
addwf TEMP,W

; allocate new address
movlw .3
003D 0090  00118  movwf  TEMP
003E  00119  SD
003E  0B90  00120  decfsz  TEMP,  F
003F  283E  00121  goto  SD
0040  0008  00122  return
       00123
       00124  ;
       00125
       00126  END

MEMORY USAGE MAP ('X' = Used, '-' = Unused)

0000 : X---X------------ XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX
0040 : X-------------------------- ------------------ ------------------

All other memory blocks unused.

Program Memory Words Used: 51
Program Memory Words Free: 973

Errors : 0
Warnings : 0 reported, 0 suppressed
Messages : 0 reported, 2 suppressed
Note the following details of the code protection feature on PICmicro® MCUs.

- The PICmicro family meets the specifications contained in the Microchip Data Sheet.
- Microchip believes that its family of PICmicro microcontrollers is one of the most secure products of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the PICmicro microcontroller in a manner outside the operating specifications contained in the data sheet. The person doing so may be engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable”.
- Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our product.

If you have any further questions about this matter, please contact the local sales office nearest to you.

Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchip’s products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights.

Trademarks

The Microchip name and logo, the Microchip logo, FilterLab, KEELoq, microID, MPLAB, PIC, PICmicro, PICMASTER, PICSTART, PRO MATE, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

dsPIC, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, microPort, Migratable Memory, MPASM, MPLIB, MPSIM, MXDEV, PICC, PICDEM, PICDEM.net, rIPIC, Select Mode and Total Endurance are trademarks of Microchip Technology Incorporated in the U.S.A.

Serialized Quick Turn Programming (SQTP) is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2002, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

Microchip received QS-9000 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona in July 1999. The Company’s quality system processes and procedures are QS-9000 compliant for its PICmicro® 8-bit MCUs, KEELoq® code hopping devices, Serial EEPROMs and microperipheral products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001 certified.
Worldwide Sales and Service

**AMERICAS**

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200 Fax: 480-792-7277
Technical Support: 480-792-7627
Web Address: http://www.microchip.com

**Rocky Mountain**
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7966 Fax: 480-792-7456

**Atlanta**
500 Sugar Mill Road, Suite 200B
Atlanta, GA 30350
Tel: 770-640-0034 Fax: 770-640-0307

**Boston**
2 Lan Drive, Suite 120
Westford, MA 01886
Tel: 978-692-3848 Fax: 978-692-3821

**Chicago**
333 Pierce Road, Suite 180
Itasca, IL 60143
Tel: 630-285-0071 Fax: 630-285-0075

**Dallas**
4570 Westgrove Drive, Suite 160
Addison, TX 75001
Tel: 972-818-7924 Fax: 972-818-2924

**Detroit**
Tri-Atria Office Building
32255 Northwestern Highway, Suite 190
Farmington Hills, MI 48334
Tel: 248-538-2250 Fax: 248-538-2260

**Kokomo**
2767 S. Albright Road
Kokomo, Indiana 46902
Tel: 765-864-8360 Fax: 765-864-8387

**Los Angeles**
18201 Von Karman, Suite 1090
Irvine, CA 92612
Tel: 949-263-1888 Fax: 949-263-1338

**New York**
150 Motor Parkway, Suite 202
Hauppauge, NY 11788
Tel: 631-273-5305 Fax: 631-273-5335

**San Jose**
Microchip Technology Inc.
2107 North First Street, Suite 590
San Jose, CA 95131
Tel: 408-436-7950 Fax: 408-436-7955

**Toronto**
6285 Northaven Drive, Suite 108
Mississauga, Ontario L4V 1X5, Canada
Tel: 905-673-0699 Fax: 905-673-6509

**ASIA/PACIFIC**

**Australia**
Microchip Technology Australia Pty Ltd
Suite 22, 41 Rawson Street
Epping 2121, NSW
Australia
Tel: 61-2-9868-6733 Fax: 61-2-9868-6755

**China - Beijing**
Microchip Technology Consulting (Shanghai) Co., Ltd., Beijing Liaison Office
Unit 915
Beihai Wan Tai Bldg., No. 6 Chaoyangmen Beidajie
Beijing, 100027, No. China
Tel: 86-10-85282100 Fax: 86-10-85282104

**China - Chengdu**
Microchip Technology Consulting (Shanghai) Co., Ltd., Chengdu Liaison Office
Rm. 2401, 24th Floor, Ming Xing Financial Tower
No. 88 TIDU Street
Chengdu 610016, China
Tel: 86-28-6766200 Fax: 86-28-6766599

**China - Fuzhou**
Microchip Technology Consulting (Shanghai) Co., Ltd., Fuzhou Liaison Office
No. 71 Wuli Road
Fuzhou 350001, China
Tel: 86-591-7503506 Fax: 86-591-7503521

**China - Shanghai**
Microchip Technology Consulting (Shanghai) Co., Ltd., Shanghai Liaison Office
Room 701, Bldg. B
Far East International Plaza
No. 317 Xian Xing Road
Shanghai, 200005
Tel: 86-21-6275-5700 Fax: 86-21-6275-5060

**China - Shenzhen**
Microchip Technology Consulting (Shanghai) Co., Ltd., Shenzhen Liaison Office
Rm. 1315, 13/F, Shenzhen Kerry Centre,
Renminnan Lu
Shenzhen 518001, China
Tel: 86-755-2350361 Fax: 86-755-2366086

**Hong Kong**
Microchip Technology Hong Kong Ltd.
Unit 901-6, Tower 2, Metroplaza
223 Hing Fong Road
Kwai Fong, N.T., Hong Kong
Tel: 852-2401-1200 Fax: 852-2401-3431

**India**
Microchip Technology Inc.
India Liaison Office
Divyasree Chambers
1 Floor, Wing A (A3/A4)
No. 11, O’Shaugnesssey Road
Bangalore, 560 025, India
Tel: 91-80-2290061 Fax: 91-80-2290062

**Japan**
Microchip Technology Japan K.K.
Benex S-1 6F
3-18-20, Shinyokohama
Kohoku-Ku, Yokohama-shi
Kanagawa, 222-0033, Japan
Tel: 81-45-471-6166 Fax: 81-45-471-6122

**Korea**
Microchip Technology Korea
168-1, Youngbo Bldg. 3 Floor
Samsung-Dong, Kangnam-Ku
Seoul, Korea 135-882
Tel: 82-2-554-7200 Fax: 82-2-558-5934

**Singapore**
Microchip Technology Singapore Pte Ltd.
200 Middle Road
#07-02 Prime Centre
Singapore, 189980
Tel: 65-6334-8870 Fax: 65-6334-8850

**Taiwan**
Microchip Technology Taiwan
11F-3, No. 207
Tung Hua North Road
Taipei, 105, Taiwan
Tel: 886-2-2717-7175 Fax: 886-2-2545-0139

**EUROPE**

**Denmark**
Microchip Technology Nordic ApS
Regus Business Centre
Lautrup høj 1-3
Ballrup DK-2750 Denmark
Tel: 45 4420 9895 Fax: 45 4420 9910

**France**
Microchip Technology SARL
Parc d’Activite du Moulin de Massy
43 Rue du Saule Trapu
Batiment A - 1er Etage
91300 Massy, France
Tel: 33-1-69-33-63-20 Fax: 33-1-69-30-90-79

**Germany**
Microchip Technology GmbH
Gustav-Heinemann Ring 125
D-81739 Munich, Germany
Tel: 49-89-6275-144 0 Fax: 49-89-6275-144-44

**Italy**
Microchip Technology SRL
Centro Direzionale Colleoni
Palazzo Taurus 1 V. Le Colleoni 1
20041 Agrate Brianza
Milan, Italy
Tel: 39-039-65791-1 Fax: 39-039-6899883

**United Kingdom**
Microchip Technology Ltd.
505 Eskdale Road
Winnersh Triangle
Wokingham
Berkshire, England RG41 5TU
Tel: 44 118 921 5869 Fax: 44-118 921-5820

© 2002 Microchip Technology Inc.