A/D CONVERTER FUNCTIONS

The information in this document is obtained from the following Microchip manuals:

- PIC18Fxx2 Datasheet
- PICmicro® 18C MCU Family Reference Manual
- MPLAB C18 C Compiler Libraries

Function Prototypes:
For a detailed description of these functions, please see:
Section 2.2 A/D Converter Functions, in MPLAB C18 C Compiler Libraries manual.

```
#include <adc.h>

void OpenADC    ( unsigned char config, unsigned char config2 );
                                                      // Configure the A/D converter.
void SetChanADC ( unsigned char channel );            // Select A/D channel to be used.

void ConvertADC ( void );                             // Start an A/D conversion.

char BusyADC    ( void );   // Is A/D converter currently performing a conversion?

int  ReadADC    ( void );                    // Read the results of an A/D conversion.

void CloseADC   ( void );                             // Disable the A/D converter.
```

Notes:

1. The Analog-to-Digital (A/D) converter module has five input channels for the PIC18F2X2 devices (only four channels are available on C_STAMP) and eight for the PIC18F4X2 devices.

2. The analog input charges a sample and hold capacitor. The output of the sample and hold capacitor is the input into the converter. The converter then generates a digital result of this analog level via successive approximation. This A/D conversion of the analog input signal results in a corresponding 10-bit digital number.

3. The analog reference voltages (positive and negative supply) are software selectable to either the device's supply voltages (AVDD, AVss) or the voltage level on the AN3/VREF+ and AN2/VREFpins.

4. The A/D converter has the unique feature of being able to convert while the device is in SLEEP mode.

5. On any device RESET, the port pins that are multiplexed with analog functions (ANx) are forced to be an analog input.

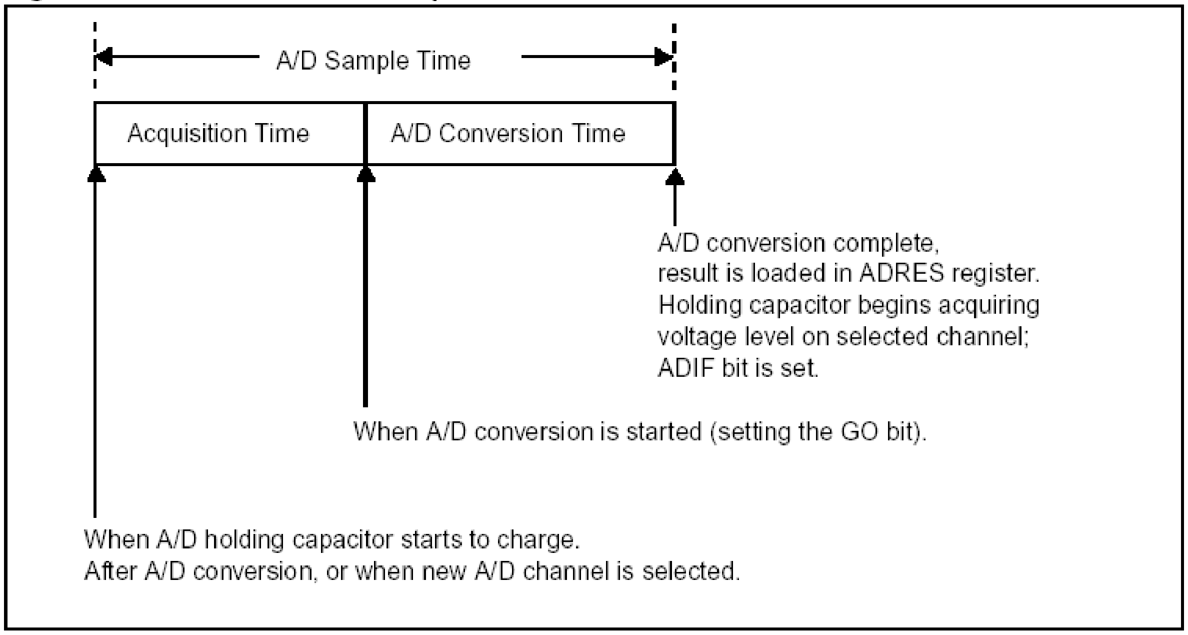
6. The ADCON1, TRISA and TRISE registers control the operation of the A/D port pins. The port pins that are desired as analog inputs, must have their corresponding TRIS bits set (input). If the TRIS bit is cleared (output), the digital output level (VOH or VOL) will be converted.

7. When reading the port register, all pins configured as analog input channels will read as cleared (a low level). Pins configured as digital inputs will convert an analog input. Analog levels on a digitally configured input will not affect the conversion accuracy.

8. Analog levels on any pin that is defined as a digital input (including the AN4:AN0 pins) may cause the input buffer to consume current that is out of the device's specification.

9. **The maximum recommended output impedance for analog sources is 2.5 kΩ.**

10. After the A/D module has been configured, the signal on the selected channel must be acquired before the conversion is started. The analog input channels must have their corresponding TRIS bits selected as inputs.

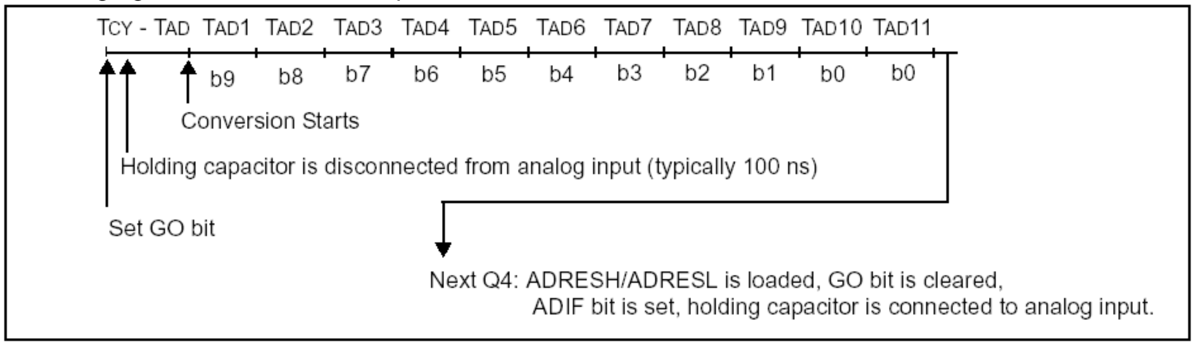

A/D Sample Time
Acquisition Time
A/D Conversion Time
A/D conversion complete, result is loaded in ADRES register. Holding capacitor begins acquiring voltage level on selected channel; ADIF bit is set.
When A/D conversion is started (setting the GO bit).
When A/D holding capacitor starts to charge. After A/D conversion, or when new A/D channel is selected.

11. The worst case acquisition time is 12.86 µs (micro-second). After this acquisition time has elapsed, the A/D conversion can be started.

12. The A/D conversion time per bit is defined as TAD. The A/D conversion requires 12 TAD per 10-bit conversion. For correct A/D conversions, the A/D conversion clock (TAD) must be selected to ensure a minimum TAD time of 1.6 µs. Following table shows the resultant TAD times derived from the device operating frequencies and the A/D clock source selected.

AD Clock Source (TAD)		**Maximum Device Frequency**	
Operation	**ADCS2:ADCS0**	**PIC18FXX2**	**PIC18LFXX2**
2 TOSC	000	1.25 MHz	666 kHz
4 TOSC	100	2.50 MHz	1.33 MHz
8 TOSC	001	5.00 MHz	2.67 MHz
16 TOSC	101	10.00 MHz	5.33 MHz
32 TOSC	010	20.00 MHz	10.67 MHz
64 TOSC	110	40.00 MHz	21.33 MHz
RC	011	—	—

13. Following figure 17-3 shows the operation of the A/D converter after the GO bit has been set.


TCY - TAD TAD1 TAD2 TAD3 TAD4 TAD5 TAD6 TAD7 TAD8 TAD9 TAD10 TAD11
b9 b8 b7 b6 b5 b4 b3 b2 b1 b0 b0
Conversion Starts
Holding capacitor is disconnected from analog input (typically 100 ns)
Set GO bit
Next Q4: ADRESH/ADRESL is loaded, GO bit is cleared, ADIF bit is set, holding capacitor is connected to analog input.

14. For a 20MHz PIC **18LF252/452**, the A/D Clock Source should be **64 T_{OSC}**, and the conversion time will be 3.2 µs x 12 = 38.4 µs. Therefore the total A/D sample time will be 12.86 µs + 38.4 µs = 51.26 µs. If this value is rounded up to 52 µs, the corresponding **maximum** sampling frequency will be 1/52 = **19.23 kHz**.

15. The same calculations for a 20MHz PIC **18F252/452** will be:
    A/D Clock Source = **32 T_{OSC}**
    Conversion Time = 1.6 μs x 12 = 19.2 μs
    Total A/D sample time = 12.86 μs + 19.2 μs = 32.06 μs rounded to 32 μs
    M**aximum** sampling frequency = 1/32 = **31kHz**

16. Clearing the GO/DONE bit during a conversion will abort the current conversion. The A/D result register pair will NOT be updated with the partially completed A/D conversion sample. That is, the ADRESH:ADRESL registers will continue to contain the value of the last completed conversion (or the last value written to the ADRESH:ADRESL registers). After the A/D conversion is aborted, a 2 TAD wait is required before the next acquisition is started. After this 2 TAD wait, acquisition on the selected channel is automatically started. The GO/DONE bit can then be set to start the conversion.

17. The GO/DONE bit should NOT be set in the same instruction that turns on the A/D.

18. The following steps should be followed for doing an A/D conversion:
    1. Configure the A/D module:
        • Configure analog pins, Voltage Reference, and digital I/O (ADCON1)
        • Select A/D input channel (ADCON0)
        • Select A/D conversion clock (ADCON0)
        • Turn on A/D module (ADCON0)
    2. Configure A/D interrupt (if desired):
        • Clear the ADIF bit
        • Set the ADIE bit
        • Set/Clear the ADIP bit
        • Set the GIE/GIEH or PEIE/GIEL bit
    3. Wait the required acquisition time.
    4. Start conversion:
        • Set the GO/DONE bit (ADCON0)
    5. Wait for the A/D conversion to complete, by either:
        • Polling for the GO/DONE bit to be cleared or the ADIF bit to be set, or
        • Waiting for the A/D interrupt
    6. Read A/D Result register pair (ADRESH:ADRESL): clear the ADIF bit, if required.
    7. For next conversion, go to step 1 or step 2 as required.

19. Example Use of the A/D Converter Routines

```
#include <p18f452.h>
#include <adc.h>
#include <delays.h>

int result;
void main (void){

      // configure A/D convertor
      OpenADC( ADC_FOSC_32 &
               ADC_RIGHT_JUST &
               ADC_8ANA_0REF,
               ADC_CH0 &
               ADC_INT_OFF );

      Delay10TCYx(5);       // Delay 50TCY for acquisition time
      ConvertADC();         // Start conversion
      while( BusyADC() );   // Wait for completion
      result = ReadADC();   // Read result
      CloseADC();           // Disable A/D converter
}
```