

Chapter 8. A/D Conversion and Data Acquisition

A/D conversion (or analog-to-digital conversion) is to read analog values from, for example, temperature or voltage level, into the PIC chip in the form of digital value. PIC 16F877 has an internal built-in module for this A/D conversion. A/D conversion has many applications: reading from a temperature sensor and displaying in on a PC screen of an LCD display; reading from a pressure sensor for blood pressure and acquiring the pressure data into a text file in a PC; reading a current value through an electric wire and alerting a circuit protection device for an above-normal power consumption, etc. In this chapter, we thoroughly study the built-in A/D module and practice of A/D conversion coding with a few practical applications.

1. A/D Conversion Module

A/D conversion is well utilized for external analog signal reading such as voltage, current, temperature, pressure, distance, or even color information. 16F877 has a A/D module. In this chapter, we will study the details of A/D converter module and its application. As need arises, some explanation on general A/D Conversion is discussed occasionally.

There is an 8 channel A-to-D (or A/D) converter module inside a 16F877: AN7 – AN0. These pins are not as well organized as other I/O pins. The lower four channels, AN0 – AN3, are arranged in the pin nos. 2 – 5, and AN4 – AN7 arrange with the pin Nos. of 7 through 10.

The A/D module allows conversion of an analog input signal to a corresponding 10-bit digital number. The output of the sample and hold is the input into the converter, which generates the result via successive approximation.

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/VREF-pins.

There are three types of registers we have to well control for A/D conversion. They are: A/D Result Registers (ADRESH and ADRESL), A/D Control Register0 (ADCON0), and A/D Control Register1 (ADCON1). The ADCON0 register controls the operation of the A/D module. The ADCON1 register configures the functions of the port pins. ADRESH and ADRESL registers contain the 10-bit results. Since each register is 8-bit register, we see that only one of the registers would be fully filled while the other would be partially filled by the A/D conversion result. Which register we configure to be fully filled, and which one to be partially filled is controlled by 'result justification': left- or right- justified.

Let's examine ADCON0 register first for the A/D operation.

The first two bits are assigned to select the A/D conversion clock. For correct A/D conversion, as the electrical specification of 16F877 states, the minimum A/D conversion clock must be selected to ensure a minimum of 1.6 μ s. With 20MHz crystal oscillation, the pre-scaled clock of $F_{osc}/2$ would be 100 ns, while the pre-scaled clock of $F_{osc}/8$ would be 400ns. So either selection would violate the minimum conversion clock of 1.6 μ s. The $F_{osc}/32$ with 1.6 μ s would satisfy the minimum clock. However, the internal RC source has typical 6 μ s of clock pulse. So, in 20MHz oscillator, selection of RC is safer and may be the only safe option for the A/D clock.

ADCON0 Register (1Fh)

ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	0	ADON
						Always 0	

ADCS1:ADCS0: A/D Conversion Clock Select bits

00 = $F_{osc}/2$

01 = $F_{osc}/8$

10 = $F_{osc}/32$

11 = FRC (clock derived from internal A/D RC oscillator)

CHS2:CHS0: Analog Channel Select bits

000 = channel 0, (AN0) 001 = channel 1, (AN1)

010 = channel 2, (AN2) 011 = channel 3, (AN3)

100 = channel 4, (AN4) 101 = channel 5, (AN5)

110 = channel 6, (AN6) 111 = channel 7, (AN7)

GO/DONE: A/D Conversion Status bit

1 = A/D conversion in progress (**Start A/D**)

0 = A/D conversion not in progress

ADON: A/D On bit

1 = A/D converter **On**

0 = A/D converter **Off**

The next three bits select which one channel we use to read analog signal from, external world. Similar channel selection is done in ADCON1 to determine which channels are for analog input and which are for digital I/Os. Anyway, for ADCON0, select one channel you want to read. If you have multiple analog signals, you still have to select one channel for reading and then select another for another reading, etc. CS2:CS0=(000) would select the AN0 for the analog signal reading channel. The second bit (GO/~DONE) indicates the A/D conversion status: 1 indicates the process is still going on and 0 for no process. By setting the bit, we can start the A/D process. This bit is automatically cleared, when a process is finished, there is no need to clear the bit in program code. The last bit ADON works as a switch to turn on/off the A/D module: setting would make the A/D module ready for a conversion process. However, the final say is reserved to the GO/~DONE bit for actually starting the conversion.

For the ADCON1 register, we use only five bits: ADFM and PCFG3:PCFG0.

ADFM is to decide how we store the 10-bit A/D conversion result to the two A/D result registers: ADRESH and ADRESL. When set, the "Right Justification" is selected which stores the 8 LSBs of the result are stored to ADRESL and the 2 MSBs of the results are stored to 2 LSB positions of ADRESH. On the other hand, with its bit cleared, the "Left Justification" is selected which stores the 8 MSBs of the result into ADRESH register and the 2 LSBs of the result to the 2 MSB positions of ADRESL register. See the diagram below for illustration.