**Example 18-1: Converting 1 Channel, Manual Sample Start (ASAM=0):**

**18.2 Control Registers**

The A/D module has six Control and Status registers. These registers are:

• ADCON1: A/D Control Register 1

• ADCON2: A/D Control Register 2

• ADCON3: A/D Control Register 3

• ADCHS: A/D Input Channel Select Register

• ADPCFG: A/D Port Configuration Register

• ADCSSL: A/D Input Scan Selection Register

The ADCON1, ADCON2 and ADCON3 registers control the operation of the A/D module. The

ADCHS register selects the input pins to be connected to the S/H amplifiers. The ADPCFG

register configures the analog input pins as analog inputs or as digital I/O. The ADCSSL register

selects inputs to be sequentially scanned.

**18.3 A/D Result Buffer**

The module contains a 16-word dual port RAM, called ADCBUF, to buffer the A/D results. The

16 buffer locations are referred to as ADCBUF0, ADCBUF1, ADCBUF2, ..., ADCBUFE,

ADCBUFF.

**Note:** The A/D result buffer is a read only buffer.

**Manual Conversion Start Code Example**

ADPCFG = 0xFFFB; // Register 18-5: (oB1111111111111011) all PORTB = Digital; // RB2=analog

ADCON1 = 0x0000; // Register 18-1:

// ADON=0

// ADSIDL=0 Continue module operation in Idle mode

// FORM 00 = Integer (DOUT = 0000 dddd dddd dddd)

// SSRC=000 000 = Clearing SAMP bit ends sampling and starts conversión

// ASAM=0 Sampling begins when SAMP bit set

// SAM= 0 A/D sample/hold amplifiers are holding

// DONE=0 Clearing this bit will not effect any operation in progress.

// o sea:SAMP bit = 0 ends sampling ...

// and starts converting y luego esta en continue mode….

ADCHS = 0x0002; // Register 18-4:

// CHOSA=0010 Channel 0 positive input is AN2

// CHONA=0 Channel 0 negative input is VREF-

// aquí se usó el channel A CHOXB todo 0

ADCSSL = 0; // Register 18-6: ADCSSL: A/D Input Scan Select Register

// 0 = Skip ANx for input scan aquí se usa un solo canal

ADCON3 = 0x0002; // Register 18-3:

// SAMC=0 0 Tad

// ADRC=0Clock derived from system clock

// ADCS= 2 A/D conversión clock daría 3/2 TCY… ver….

ADCON2 = 0; // **Register 18-2:**

// VCFG=0 usara AVDD y AVSS como referencias + y –

// **CSCNA=0** Do not scan inputs

// BUFS =0 0 A/D is currently filling buffer 0x0-0x7, user should access

// data in 0x8-0xF

// **SMPI<3:0>:** 0000 = Interrupts at the completion of conversion for each // sample/convert sequence ( no se usan interrupciones en este ejemplo)

// **ALTS=0** Always use MUX A input multiplexer settings

ADCON1bits.ADON = 1; // turn ADC ON,el bit de un registro se maneja individualmente

// como un struct

while (1) // repeat continuously

{

ADCON1bits.SAMP = 1; // start sampling ...

DelayNmSec(100); // for 100 mS El tiempo de adquisición se maneja manual en // este ejemplo. Este Tad es enorme !!!!

ADCON1bits.SAMP = 0; // start Converting

while (!ADCON1bits.DONE); // conversion done? Espera a que DONE=1

ADCValue = ADCBUF0; // yes then descargo en la varible ADCValue el ADCBUF0

}