For example, to enable channels 1 and 5, either of the following statements could be included:

```
ADC10AE0 = 1 + 32;
ADC10AE0 = 33;
```

**Enable ADC** :

The actual conversion(s) will occur when the following constants are added to **ADC10CTL0**:

```
+ ADC10ON; (last two letters spell "on")
+ ENC;     (ENC stands for ENable Conversion)
```

**Reading The Digital Value After Conversion** *:*

The ADC core converts an analog input to its 10 bit digital representation and stores the result in the **ADC10MEM** register. Any program variable can be set to the momentary value of **ADC10MEM** (i.e., the result of the most recent conversion). For example:

```
X = ADC10MEM;
```

where x can now be used to perform operations in C code.

**Disable the Watchdog Timer (If Necessary)**

The following code statement will disable the Watchdog Timer if you want ADC performed according to the chosen clock. Note: It is permissible to use the timing of the Watchdog to decide when ADC will occur.

```
WDTCTL = WDTPW + WDTHOLD;
```

**Some Simple Sample Code**

(This code would be included in a program loop to set the variable *n* to the instantaneous value of `ADC10MEM` for each execution of the loop.)

```
// SETUP COMMANDS:

// Turn off the WatchDog Timer
      WDTCTL = WDTPW + WDTHOLD;


// Set conversion to single channel and continuous-sampling mode
      ADC10CTL1 |= CONSEQ1;

//Set S/H time, 10-bit converter, and continuous sampling:
      ADC10CTL0 |= ADC10SHT_2 + ADC10ON + MSC;

//Choose P1.0 (channel 1) as an analog input pin:
      ADC10AE0  |= 1;

// Start A/D conversion; The result will appear in the memory variable "ADC10MEM"
      ADC10CTL0 |= ADC10SC + ENC;

// SET VARIABLE n TO THE RESULT OF THE A/D CONVERSION:
      n = ADC10MEM;
```