

```

//Description:
//
//In the PIC18F4550, there are 13 channels which be used as A/D conversion channels. See the
schematic of
//PIC18F4550, the channels with ANx (x is 0~12), that means that channel can be used as ADC.
//
//This example shows how the ADC configs by using the library fucntion of <adc.h>
//Before starting the conversion, some settings must be defined firstly.(A/D clock,
//result justified from right or left,acquistion time, input channel and reference voltage.)
//
//
//One important point is the conversion result (10 bit) will be written into register
ADRESH:ADRESL, which has
//16bit. You need to config it for wirting the result from left or right. There are only 10 bits
out of 16 bits will
//be used in ADRESH:ADRESL.
//
//For the conversion time, it is beter to choose slowest conversion clock and at least 12TAD for
making sure
//correct conversion.
//
//
//A/D conversion needs to config reference voltage, it can be chosen to use controller supply
voltage or external
//voltage.
//
//Last, when the conversion started, the flag will be set. Then if the conversion is complete,
the flag will
//be clear. By using ReadADC(), you can read the conversion result(ADRESH:ADRESL).

#include <p18f4550.h>
#include <adc.h> // include the ADC library.
#include <delays.h>

//Always include this code, it is necessary when using a bootloader
extern void _startup (void);
#pragma code _RESET_INTERRUPT_VECTOR = 0x000800
void _reset (void)
{
    _asm goto _startup _endasm
}
#pragma code
#pragma code _HIGH_INTERRUPT_VECTOR = 0x000808
void high_ISR (void)
{
}
#pragma code
#pragma code _LOW_INTERRUPT_VECTOR = 0x000818
void low_ISR (void)
{
    ;
}
#pragma code
//End bootloader code

int result;

void main(void)
{ //Pre: The library of ADC is included.One integer is defined to contain the conversion result.

    //Post: The A/D conversion is done, and the conversion result is written in the integer.

    OpenADC( ADC_FOSC_32 & // A/D clock source set to 32Tosc
             ADC_RIGHT_JUST& // write the Digital result(10bits) from right, into ADRESH:ADRESL
             (16bits).
             ADC_20_TAD, // A/D acquisition time: 20TAD (for 10bit conversion at least
             12TAD)
             ADC_CH0 & //choose input channel (0), AN0
             ADC_INT_OFF& //ADC interrupt off

```

E:\Internship project PIC18F4550\standard library examples\ADC\ADC.c

```
ADC_VREFPLUS_VDD& // choose the supply voltage VDD as reference voltage, V+
ADC_VREFMINUS_VSS, // choose the supply voltage VSS as reference voltage, V-
7 // this value is used for setting the Analog and Digital I/O. Make
sure that AN0 is chosen as analog input.
);
```

```
while(1)
{ //SetChanADC (ADC_CH1); // choose the channel for ADC. You can choose any channel(0-12)
before you start conversion // This means you can change the A/D conversion channel at any
time.

Delay10TCYx(5); //Delay for 50TCY. Because within this delay, the holding capacitor
should be disconnected from the analog input.
ConvertADC (); // Start an A/D conversion.
while( BusyADC()); // Wait for completion. when BusyADC is cleared, the conversion is
finished.
result = ReadADC(); // Read result

}
CloseADC(); // Disable A/D converter
}
```