```
/*-------------------------------------------------------------------------------------------+
|                   Demonstration program for interfacing ADC0804 & MC145051B                |
|                                                                                            |
| FileName: DemonUsageADMultiplex.c                                                          |
| Purpose:                                                                                   |
|         This program demonstrate the usage of ADC0804 (AD converter) and MC14051B (analog |
|         switch). Using function "int getADout (char sw)" to extract the analog voltage    |
|         in digit 10-base form.                                                             |
|                                                                                            |
| Dillian Wong, 4/04                                                                         |
+-------------------------------------------------------------------------------------------*/
/*__________ I N C L U D E S _____________________________________________________________*/

#include <reg51.h>
#include <stdio.h>
#include <ctype.h>                  //use toint()
#include "ADMultiplexConfig.h"

// 9600 baud, 11.0598MHz crytral
void init_uart(void)
{
    SCON  = 0x50;
    TMOD |= 0x20;
    TH1   = 253;
    TR1   = 1;
    TI    = 1;
}


/*_____________ M A I N   F U N C T I O N _____________________________________________*/

void main (void)
{
    char cmd;
    int intData;
    init_uart();
    while(1)
    {
        printf( "\nPress 0-7 to select analog input and start conversion  : " );
        scanf("%c", &cmd);
        switch (cmd)
        {
            case '0':   intData=getADout('0');
                        printf("\nswitch 0 : %d",intData);
                        break;
            case '1':   intData=getADout('1');
                        printf("\nswitch 1 : %d",intData);
                        break;
            case '2':   intData=getADout('2');
                        printf("\nswitch 2 : %d",intData);
                        break;
            case '3':   intData=getADout('3');
                        printf("\nswitch 3 : %d",intData);
                        break;
            case '4':   intData=getADout('4');
                        printf("\nswitch 4 : %d",intData);
                        break;
            case '5':   intData=getADout('5');
                        printf("\nswitch 5 : %d",intData);
                        break;
            case '6':   intData=getADout('6');
                        printf("\nswitch 6 : %d",intData);
                        break;
            case '7':   intData=getADout('7');
                        printf("\nswitch 7 : %d",intData);
                        break;
        }
    }
}
```

```
/*-------------------------------------------------------------------------------------------+
| FileName: ADMultiplex.c                                                                    |
| Purpose:                                                                                   |
|         Interfacing the analog switch MC14051B, then use AD converter to get the anlogy    |
|         voltage reading. The function switch the multiplex input first, then start AD      |
|         converstion and return a 10-base int value varies from 0-255, 8 bits.             |
| Example:                                                                                   |
|         "x0IntData = getADout('0')" , switch X0                                            |
|         "x1IntData = getADout('1')" , switch X1                                            |
|         "x2IntData = getADout('7')" , switch X7                                            |
+-------------------------------------------------------------------------------------------*/
/*__________ I N C L U D E S _____________________________________________________________*/
#include <reg51.h>
#include <stdio.h>
#include <ctype.h>                 //use toint()

/*__________ H A R D W A R E _____________________________________________________________*/

// ADC0804
sbit ADWr = P3^3;
sbit DBData = P1;
sfr  DBDATA= 0x90;                //0x80 P0, 0x90 P1, 0xA0 p2,  0xB0, p3

// MC14051B
sbit MultiplexA = P3^4;
sbit MultiplexB = P3^5;
sbit MultiplexC = P3^2;

/*__________ M A C R O S _____________________________________________________________*/
#define setADWr ADWr= 1;while (ADWr!=1);
#define clrADWr ADWr=0;

/*__________ P R O T O T Y P E _____________________________________________________________*/
int getADout (char sw);
void switchMultiplexer(char sw);
int hex2Int(unsigned char hex8);
unsigned char hex2ascii(unsigned char bits_data);


/*____________ F U N C T I O N _____________________________________________________________*/
/*__________ ADC0804 & MC14051B _____________________________________________________________*/

/*
    select different analog input first, then return the converted int value
    that represent the analogy digit output.
    sw: 0-7, analog switch no.
    return value: 10-base value, varies from 0-255
*/
int getADout (char sw)
{
    int i;
    /* SWITCH APPROPRIATE ANALOG INPUT  */
    switchMultiplexer(sw);

    /* START AD CONVERSION */
    DBDATA=0xff;                //prepare for input
    setADWr;                    //generate a pulse for AD start conversion
    for(i=0;i<1000;i++);
    clrADWr;
    for(i=0;i<1000;i++);
    return hex2Int(DBDATA);     //return the 10base int value that represent the hex data of DBDATA
                                //port
}


/*____________ M U L T I P L E X _____________________________________________________________*/

//  select different anlog input, interfacing MC14051B
void switchMultiplexer(char sw)
{
    switch (sw)
    {
        case '0': MultiplexC = 0; MultiplexB = 0; MultiplexA=0;printf("\nswitch 0");break;
        case '1': MultiplexC = 0; MultiplexB = 0; MultiplexA=1;printf("\nswitch 1");break;
        case '2': MultiplexC = 0; MultiplexB = 1; MultiplexA=0;printf("\nswitch 2");break;
        case '3': MultiplexC = 0; MultiplexB = 1; MultiplexA=1;printf("\nswitch 3");break;
        case '4': MultiplexC = 1; MultiplexB = 0; MultiplexA=0;printf("\nswitch 4");break;
        case '5': MultiplexC = 1; MultiplexB = 0; MultiplexA=1;printf("\nswitch 5");break;
        case '6': MultiplexC = 1; MultiplexB = 1; MultiplexA=0;printf("\nswitch 6");break;
        case '7': MultiplexC = 1; MultiplexB = 1; MultiplexA=1;printf("\nswitch 7");break;
```

```
        case 'a': MultiplexC = 0; MultiplexB = 0; MultiplexA=1;printf("\nswitch a");break;
        case 'b': MultiplexC = 0; MultiplexB = 1; MultiplexA=0;printf("\nswitch b");break;
        case 'c': MultiplexC = 1; MultiplexB = 0; MultiplexA=0;printf("\nswitch c");break;
    }
}

/*_____________ H E X 2 I N T _____________________________________________________________*/

//convert the hex data (16-base)  into digit form (10-base)
int hex2Int(unsigned char hex8)
{
    char L, H;
    int i;
    L = hex2ascii(hex8&0x0f);       // convert low byte
    H = hex2ascii((hex8&0xf0)/0x0f); // convert high byte
    i=toint(H)*16+toint(L);
    return i;
}

//convert the hex data(4 bits) to ascii code(8 bits)
unsigned char hex2ascii(unsigned char bits_data)
{
    if(bits_data >=0x0f) // all invalid data return 'F'
    {
        return 0x46;
    }
    if (bits_data >=0x00 & bits_data <=0x09) // return '0'-'9'
    {
        bits_data +=0x30;
        return bits_data;
    }
    if (bits_data >=0x0a & bits_data <=0x0f); // return 'A'-'F'
    {
        bits_data +=0x37;
        return bits_data;
    }
}
```

```
/*-------------------------------------------------------------------------------------------+
|                              Header file of ADmultiplex.c                                 |
|                                                                                           |
| Function that provided....                                                                |
|         Interfacing the analog switch MC14051B, then use AD converter to get the anlogy   |
|         voltage reading. The function switch the multiplex input first, then start AD     |
|         converstion and return a 10-base int value varies from 0-255, 8 bits.            |
| Example:                                                                                  |
|         "x0IntData = getADout('0')" , switch X0                                           |
|         "x1IntData = getADout('1')" , switch X1                                           |
|         "x2IntData = getADout('7')" , switch X7                                           |
+-------------------------------------------------------------------------------------------*/

/*__________ P R O T O T Y P E _____________________________________________________________*/

int getADout (char sw);
```