

```

/*-----+
|                                     |
|                               Simple AD Converter                               |
| Name: ADConveter.c               |
|                                     |
| Purpose:                           |
|   Using 2051 as an Analog to digital converter with 64 voltage steps         |
|-----*/

/*_____ I N C L U D E S _____*/

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

sbit ADCMP = P3^6;
sbit Bit5 = P1^7;
sbit Bit4 = P1^6;
sbit Bit3 = P1^5;
sbit Bit2 = P1^4;
sbit Bit1 = P1^3;
sbit Bit0 = P1^2;

// com port with 9600 baud with crystal 11.0592MHz.
void init_uart(void)
{
    SCON = 0x50;
    TMOD |= 0x20;
    TH1 = 253;
    TR1 = 1;
    TI = 1;
}

/*
   convert digit to analogy voltage for 89C2051 P1.0 input
*/
void digit2BinaryInput(int digit)
{
    bit b5,b4,b3,b2,b1,b0;
    b0=(digit&0x0001)/1;    //(LSB), convert interger to binary
    b1=(digit&0x0002)/2;
    b2=(digit&0x0004)/4;
    b3=(digit&0x0008)/8;
    b4=(digit&0x0010)/16;
    b5=(digit&0x0020)/32;  //(MSB)

/*
   printStr("Binary : ");           //print the binary result
   printBit(b5);
   printBit(b4);
   printBit(b3);
   printBit(b2);
   printBit(b1);
   printBit(b0);
*/
    //(MSB)                                     (LSB)
    Bit5=!b5;   Bit4=!b4;   Bit3=!b3;   Bit2=!b2;   Bit1=!b1;   Bit0=!b0;
}

/*
   Main program to implement AD-conversion.
*/
void main(void)
{
    int digit=0;
    char cmd;
    char ch;

    init_uart();    // utilize rs232 port
    printStr("-----AD-Converter-----1");
    printStr("\n d. Digit to analog voltage for P1^0");
    printStr("\n s. Increment or decrement in step of analog reference voltage");
    printStr("\n c. Compare the input & reference");
    printStr("\n g. Get the digit number using analogy comparator");
    while(1)
    {
        printStr("\n[Enter Selection ] : ");
        scanf("%c",&cmd);
    }
}

```



```
/*  
  Name : rprintf.h  
  
  Header file for rprintf.c  
*/  
  
/* _____ P R O T O C O L _____ */  
  
void printHex8(unsigned char hexdata);  
void printHex16(int hex16);  
void printInt(int num);  
void printFloat(float flt);  
void printStr(char str[]);  
void rprintf(char type[],char content[]);void printBit(bit b);
```

```
/*-----+
|
|           Simple Print String
|
| Name: printStr.c
|
| Purpose:
|   Because, the puts function writes string followed by a newline character ('\n')
|   to the output stream using the putchar function. It is no sense to use it for
|   non-RS232 device. So I write a new print string function..
|-----*/
/*_____ I N C L U D E S _____*/
#include <reg51.h>
#include <stdio.h>
/*_____ F U N C T I O N _____*/
/*
   print string
*/
void printStr(char str[])
{
    // send a string stored in RAM
    // check to make sure we have a good pointer
    if (!str) return;

    // print the string until a null-terminator
    while (*str)
        putchar(*str++);
}
```

```
/*-----+
|
|           Print Bit
|
| Name: printBit.c
|
| Purpose:
|   print the bool bit to the streaming output.
|-----*/
/*_____ I N C L U D E S _____*/
#include <reg51.h>
#include <stdio.h>
/*_____ F U N C T I O N _____*/
/*
   print bit
*/
void printBit(bit b)
{
    if(b==1)
        putchar('1');
    else
        putchar('0');
}
```

```

/*-----+
|                                     |
|                               Print Integer                               |
| Name: printInt.c                                                         |
| Purpose:                                                                  |
|   print the integer to the streaming output.                             |
|-----*/

/*_____ I N C L U D E S _____*/

#include <reg51.h>
#include <stdio.h>

/*_____ P R O T O C O L _____*/
void printInt(int num);
unsigned char hex2ascii(unsigned char bits_data);

/*_____ F U N C T I O N _____*/
/*
   print interger
*/
void printInt(int num)
{
    unsigned char a,b,c,d;
    if(num<0)                                // if num is negative
    {                                          // take 2 complement
        putchar('-');
        num=~num+1;
    }
    if(num==0)                                // if num is zero, print
    {                                          // zero
        putchar('0');
    }
    a=num/0x03E8;                              // digit 1(MSB)
    b=(num-a*0x03E8)/0x64;                   // digit 2
    c=(num-a*0x03E8-b*0x64)/0x0A;            // digit 3
    d=num-a*0x03E8-b*0x64-c*0x0A;            // digit 4(LSB)

    if(a!=0x00)                                // no. print out
        putchar(hex2ascii(a));
    if((a+b)!=0x00)
        putchar(hex2ascii(b));
    if((a+b+c)!=0x00)
        putchar(hex2ascii(c));
    if((a+b+c+d)!=0x00)
        putchar(hex2ascii(d));
}
/*
   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;
    }
}

```