```
/*-------------------------------------------------------------------------------------------+
|                              Get AT Keyboard scan codes                                   |
|                                    (MPU AT89C51)                                          |
| Name : ATKeyboard.c                                                                       |
| Purpose:                                                                                  |
| Provide the interface for the ATKeyboard, print the keyboard scancode on screen through the |
| transmition of RS232 port. Hence, we can see the scancode of each keyboard key.           |
+-------------------------------------------------------------------------------------------*/
#include "config.h"

void main(void)
{
    static unsigned char rcvdata;
    init_uart();                            // 9600 baud @ 11.0592MHz
    init_int0();                            // enable INT0

    while(1)                                // use polling method to handle AT keyboard
    {                                       // interface.
        rcvdata=getrcvdata();
        print_hexdata(rcvdata);
    }
}
```

```
/*---------------------------------------------------------------------------------------+
| Name : uart.c                                                                          |
| Purpose:                                                                               |
| Initalize the com port interface in 9600 baud with crystal 11.0592MHz                  |
| Initalize the external interrupt 0 (P3.2).                                             |
+---------------------------------------------------------------------------------------*/
#include "config.h"

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

// initalize external interrupt 0 (P3.2)
void init_int0(void)
{
    PX0=1;  //Define Int0 high priority
    IE0=0;  //External Interrupt 0 edge flag, set when external interrupt detected,
            //cleared when interrupt is processed.
    IT0=1;  //set to specific falling edge produce interrupt
    EX0=1;  //enable External Interrupt 0
    EA=1;   //enable all interrupt
}
```

```
/*-------------------------------------------------------------------------------------------+
| Name : config.h                                                                            |
| Purpose:                                                                                   |
| Define the common include files for the program, hardware pin connection and function      |
| protocol.                                                                                  |
+-------------------------------------------------------------------------------------------*/
#include <reg51.h>
#include <stdio.h>

// define hardware pin connection
sbit CLK = P3^2;            // keyboard clock input pin
sbit KBDATA = P1^0;         // keyboard data input pin

// uart.c
void init_uart(void);
void init_int0(void);

//hex.c
void print_hexdata(unsigned char hexdata);
unsigned char hex2ascii(unsigned char bits_data);

//kbdata.c
unsigned char getrcvdata(void);
```

```
/*-------------------------------------------------------------------------------------------+
| Name : hex.c                                                                               |
| Purpose:                                                                                   |
| Provide the function that will print the hexdata on screen through RS232 port.             |
+-------------------------------------------------------------------------------------------*/
#include"config.h"

/*
    print the value of hex data on screen
*/
void print_hexdata(unsigned char hexdata)
{
    static unsigned char hex_h;
    static unsigned char hex_l;
    hex_l = hex2ascii(hexdata&0x0f);                    // convert low byte
    hex_h = hex2ascii((hexdata&0xf0)/0x0f);             // convert high byte
    printf("%c%c",hex_h,hex_l);                         // print out
}

/*
    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;
    }
}
```

```
/*-------------------------------------------------------------------------------------------+
| Name : kbdata.c                                                                            |
| Purpose:                                                                                   |
| Get the keyboard scancode form the keyboard, and export it to other function.              |
| Ps: I have try to export and implement convertion directly when receive a newer scancode,  |
|     the method, however, is failed. The reason, i guess is the time is not enough before   |
|     another interrupt occur. So, i use a variable to store the scancode, process it and use |
|     the buffer that provided by the standard function printf().                            |
+-------------------------------------------------------------------------------------------*/
#include"config.h"

static unsigned char rcvdata;                   // keyboard scancode data
static bit ok;                                  // scancode data validlity

/*
    export the keyboard scancode, wait until a new scancode is received.
*/
unsigned char getrcvdata(void)
{
    while(!ok);                                 // wait until a scancode received.
    ok=0;                                       // when get the scancode, clear the ok state
    return rcvdata;
}

/*
    External interrupt 0 will triger the clock pin, as the bit count continues, the scancode
    data can be obtained and stored in variable rcvdata. Flag ok indicate the validable of the
    scancode data.
*/
void getkbData(void) interrupt 0
{
    static unsigned char maskdata,bitcount;
    bitcount++;
    // bit 2 to 9 is data, 10 is parity, 11 is stop bit and 1 is start bit
    switch(bitcount)
    {
        case 1:
                break;
        case 2:
                if (KBDATA==1)
                maskdata |= 0x01;
                break;
        case 3:
                if (KBDATA==1)
                maskdata |= 0x02;
                break;
        case 4:
                if (KBDATA==1)
                maskdata |= 0x04;
                break;
        case 5:
                if (KBDATA==1)
                maskdata |= 0x08;
                break;
        case 6:
                if (KBDATA==1)
                maskdata |= 0x10;
                break;
        case 7:
                if (KBDATA==1)
                maskdata |= 0x20;
                break;
        case 8:
                if (KBDATA==1)
                maskdata |= 0x40;
                break;
        case 9:
                if (KBDATA==1)
                maskdata |= 0x80;
                rcvdata=maskdata;
                ok=1;
                break;
        case 10:
                break;
        case 11:
```

```
                bitcount=0;
                maskdata=0x00;
                break;
    }// end of switch
}
```