

目 录

第一章：开发板简介.....	3
1 - 1 .SY_07011 开发板的特性简介.....	3
1 - 2 .SY_07011 开发板的构成和工作原理.....	4
第二章：开发板使用说明.....	5
2 - 1 .系统操作软件安装.....	5
2 - 2 .开发板键盘设置.....	9
2 - 3 .开发板连接安装.....	9
2 - 4 .运行调试软件.....	10
第三章：开发板用器件资料及说明.....	15
3—1 .TIMSP430F1121.....	15
3 - 2 .DTLED-6.....	16
第四章：开发板器件表附件清单.....	19
4—1 .调试用源程序.....	19
4 - 2 .原理图.....	附录插页
4 - 2 .包装清单.....	30
第五章：其它 5 1 类实验板简介.....	32
5 - 1 .51DEMO I/O 板简介.....	32
5 - 2 .A/D89C51 数模转换实验板简介.....	23
5 - 3 .流水灯控制器(12 路).....	34
5 - 4 .SY0606 开发板.....	35
5 - 5 .Atmel_ISP 下载线(选配自购件).....	37

5 - 6 . Altera_ISP 下载线 (选配自购件)37

5 - 7 .SY03091 开发板.....38

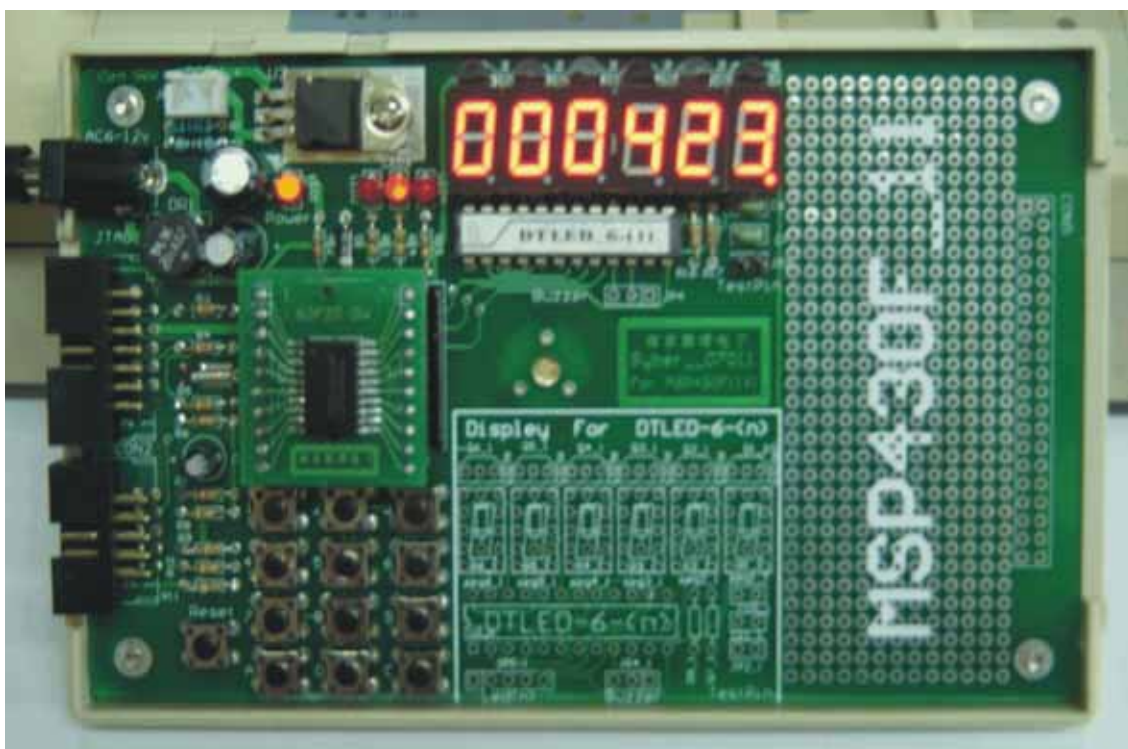
5 - 8 .MSP430Flash Emulation Tool 工具.....39

*****公司其它产品简介见软件盘中电子版文件*****

第一章：MSP430 开发板简介

1 - 1 . SY_07011 开发板的特性简介

标准的 TI 的 JTAG 和 BOOTST 接口 ,适用与 TI 的 MSP430 Flash Enulation Tool 工具配合使用。



1. 电源适应性强，可随意使用无极性 8~15V 电源或 DC+5V 电源供电。
2. 可用 MSP430 Flash Enulation Tool 工具一连串的完成编程 ,调试，程序的在线烧录（自下载），和设计功能的演示等。
3. 自带 3*4 标准键盘输入，便于学习者掌握键盘输入和程序编写。
4. 用串行驱动方式，驱动 6 位数码管显示，大大节省了单片机的接口资源（详见后面“DTLED-6”芯片介绍）。提供数码管字符显示驱动模块的接口，只用三根线就可以驱动 6 个数码

管。并可以根据用户的需要扩展数码管模块为 12 个/18 个/24 个等。

5. 有一对蜂鸣器驱动输出在显示芯片上，不占用单片机的口资源。
6. 预留扩展空间及接口，可直接驱动或控制用户设备，便于用户直接开发品。
7. 弹性空间大，预留扩展空间及接口，可直接驱动或控制用户设备，便于用户直接开发产品。

1 - 2 . SY_07011 开发板的构成和工作原理

本产品集单片机的最小系统应用于一体，在设计其电路的过程中，力求简洁明了，资源共享开放，方便实用；为单片机的二次开发和初学者提供极大的方便。其电路中包含典型的电源整流稳压电路，复位启动电路，晶体振荡时钟电路，键盘输入电路，专用 LED 数码管驱动显示电路等，是单片机初学者的精典范例。也是工程开发人员可随手取及的最佳半成品。

工作原理：SY_07011 开发板是与其它实验板一样具有直观的演示功能，上电复位后进入启动工作状态，程序自动将LED数码管上电显示并且清零，并进入累加计数状态状态进行演示，另有三端口有单个发光管交替闪亮，便于用户的直接用于驱动控制。按键盘可输入 1/2/3/4/5/6/7/8/9/A/C，数字在数码管上显示出来。

第二章：开发板使用说明

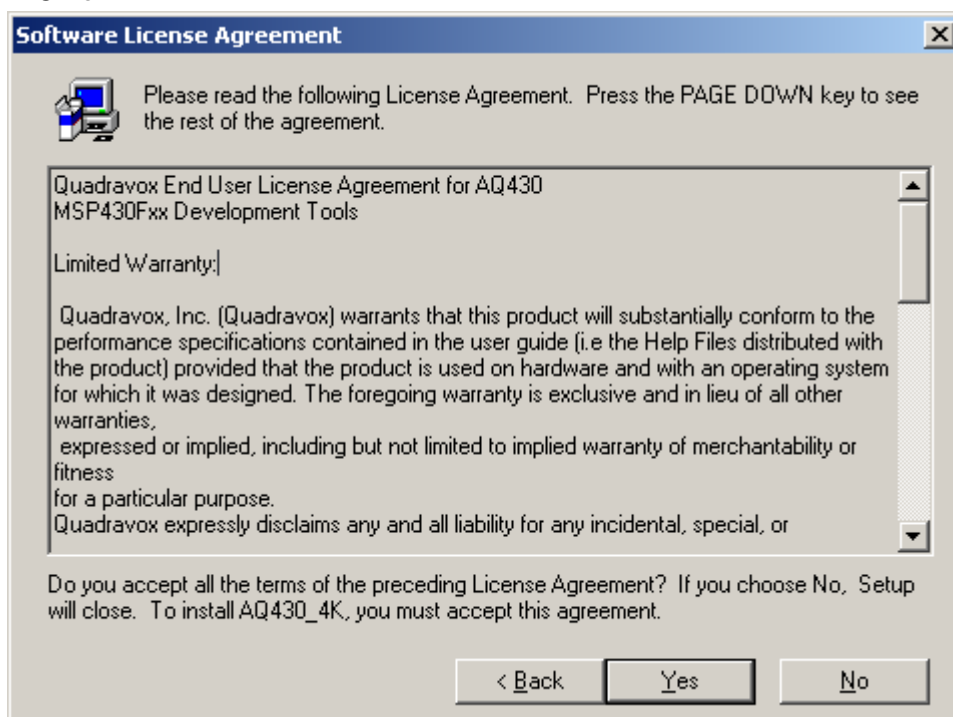
2 - 1 . 操作软件安装：

****建议使用正版软件,请与当地经销商联系****

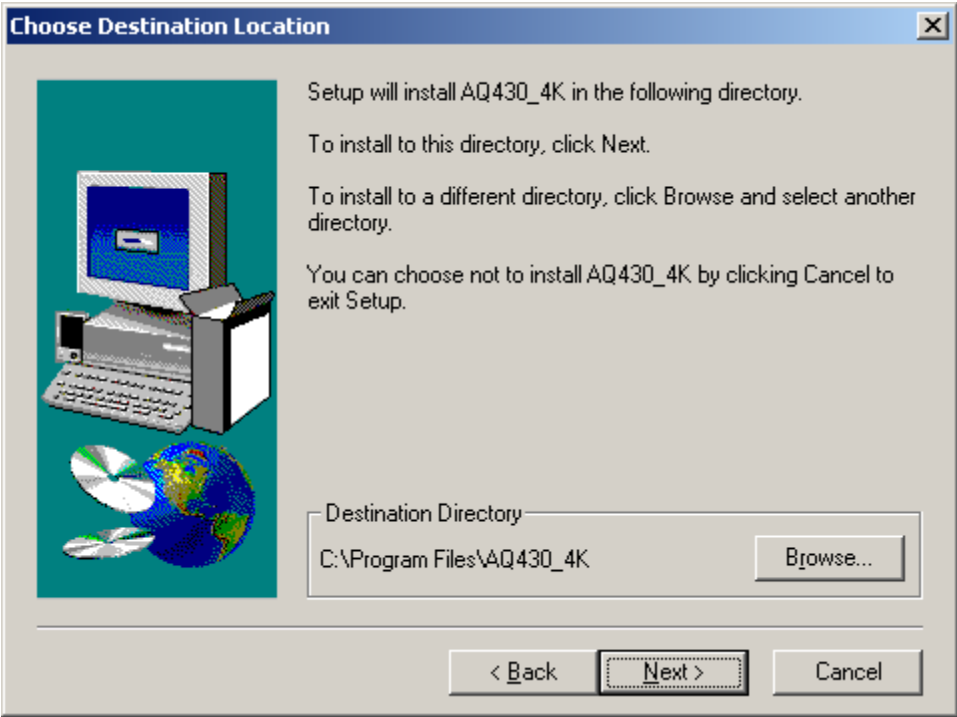
1. 运行光盘上免费提供\AQ4302K_4K.zip 解压缩文件 ,进行安装。如下画面：



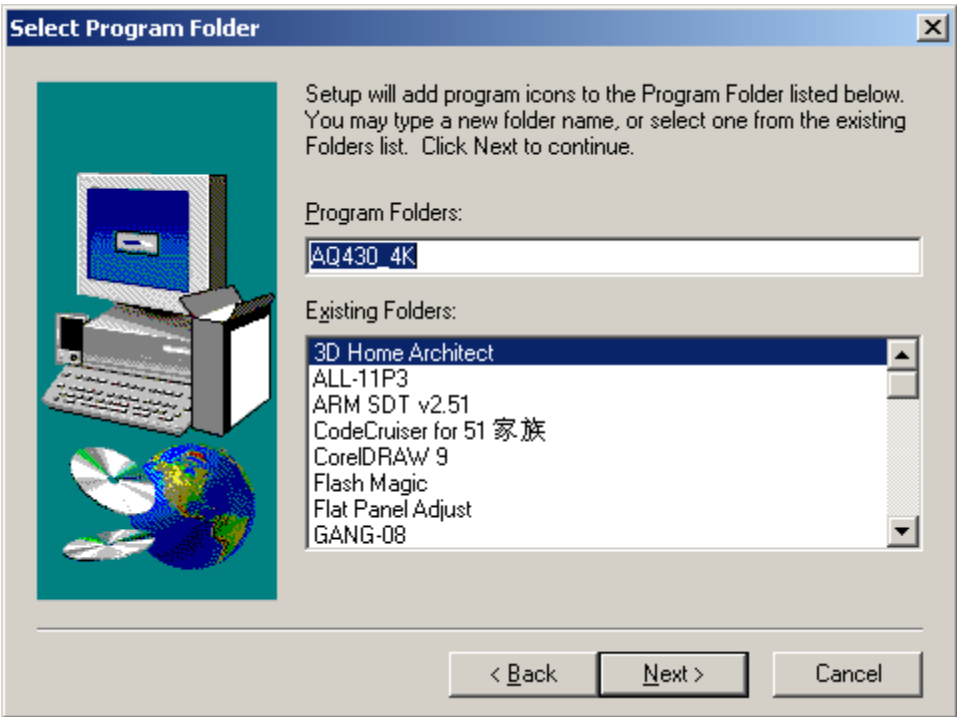
2. 点击“Next”：



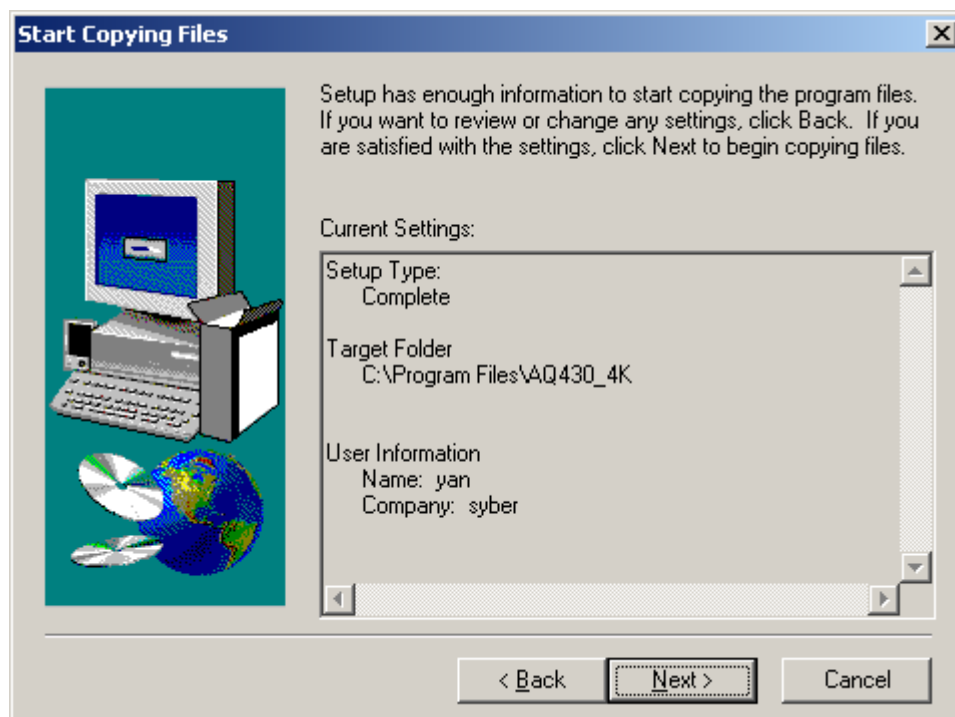
3. 点击 “ Yes ” :



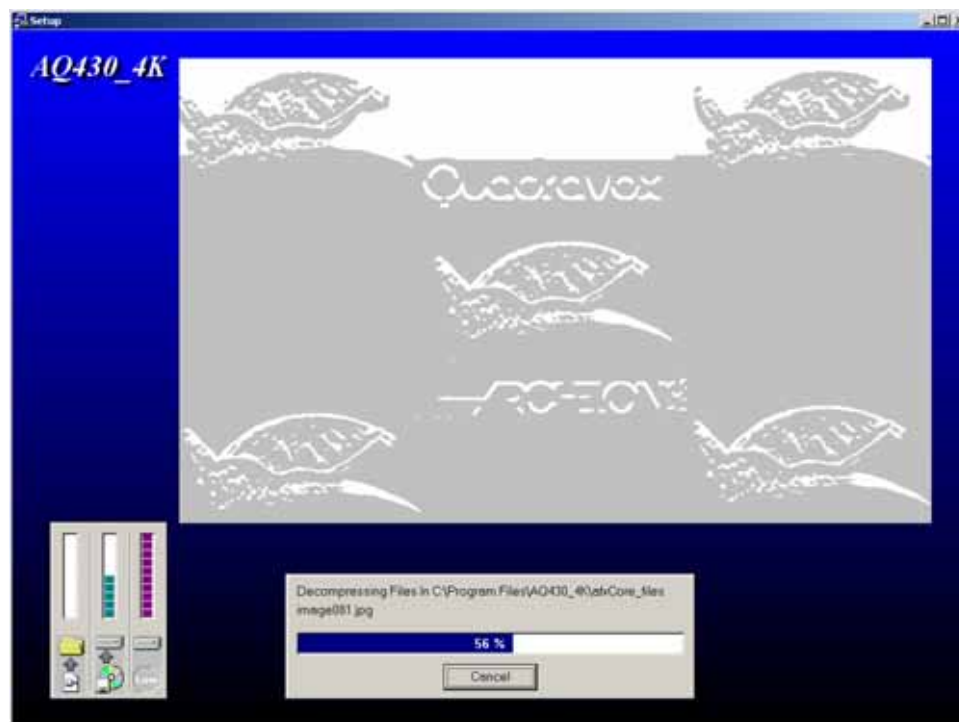
4. 选择路径 (或省略), 点击 “ Next ” :



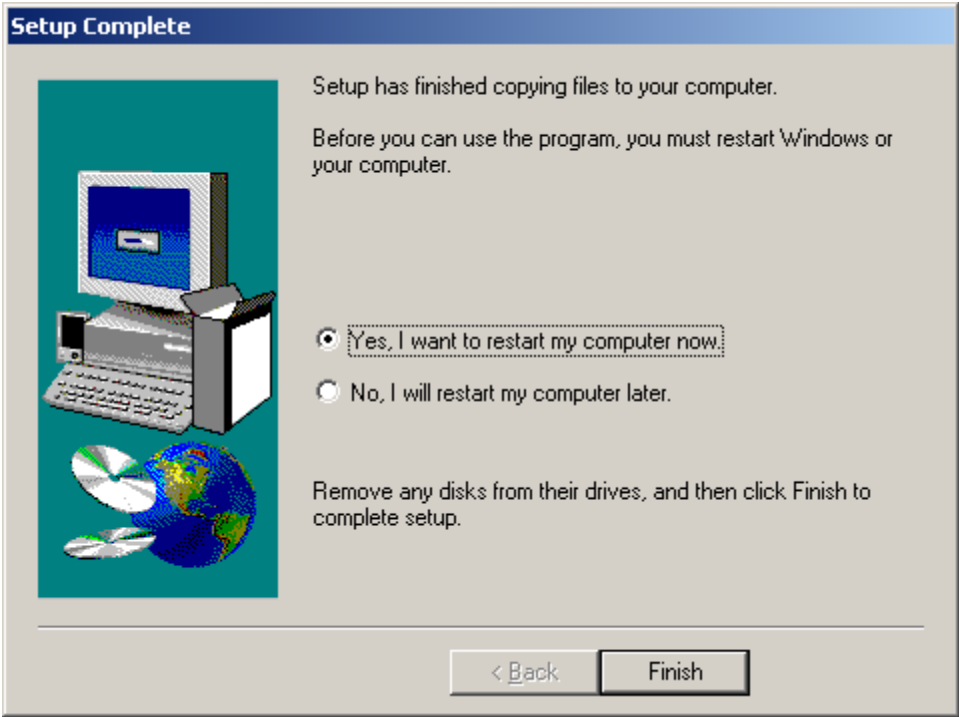
5. 点击“Next”：



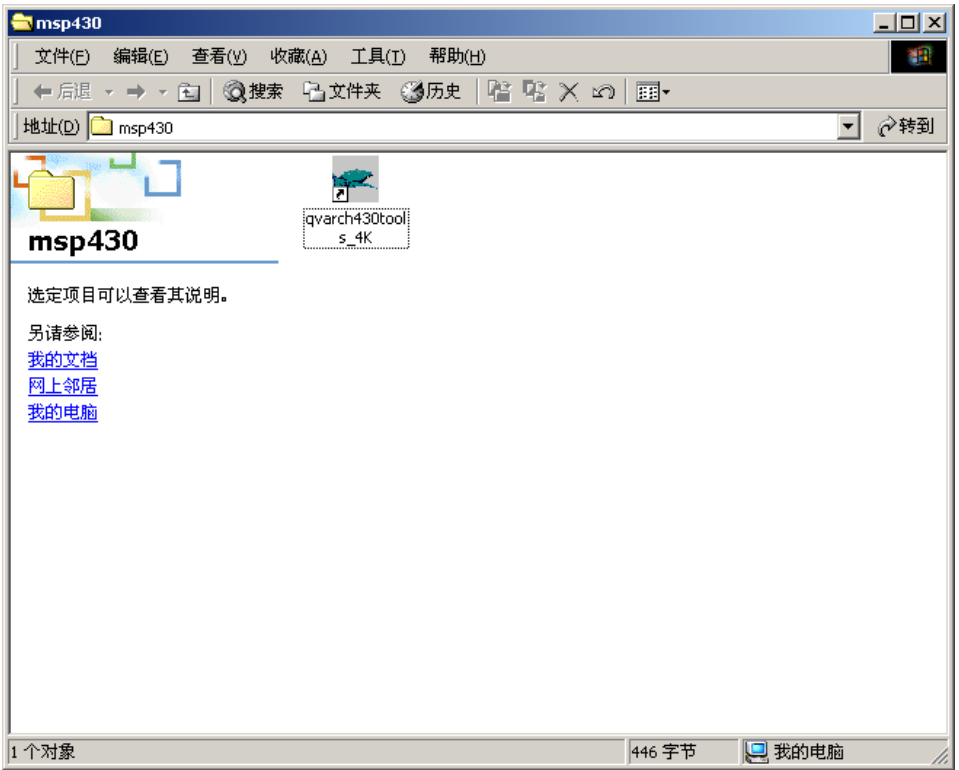
6. 点击“Next”：



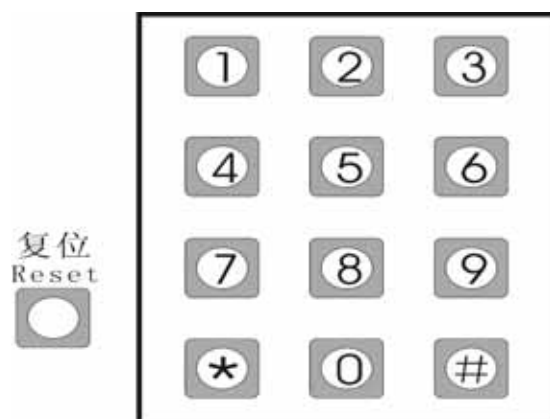
7. 直到出现安装完成提示，如下图：



8. 软件安装完成后，桌面上出现如下图标，即可进行软件调试。



2 - 2 . 开发板键盘设置如下图：

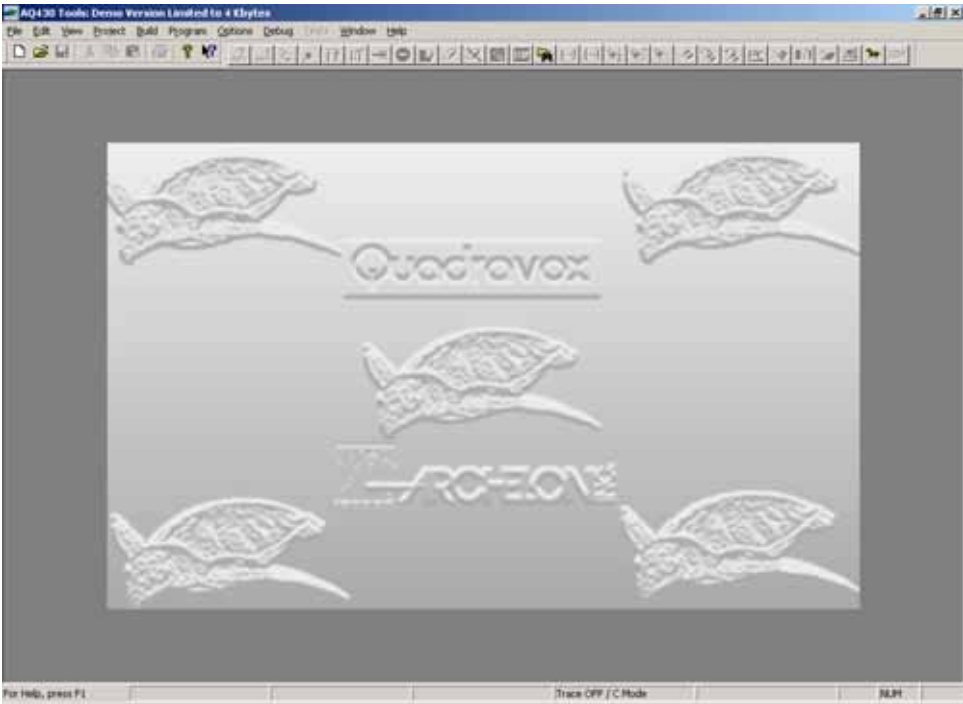


2 - 3 . 开发板连接安装

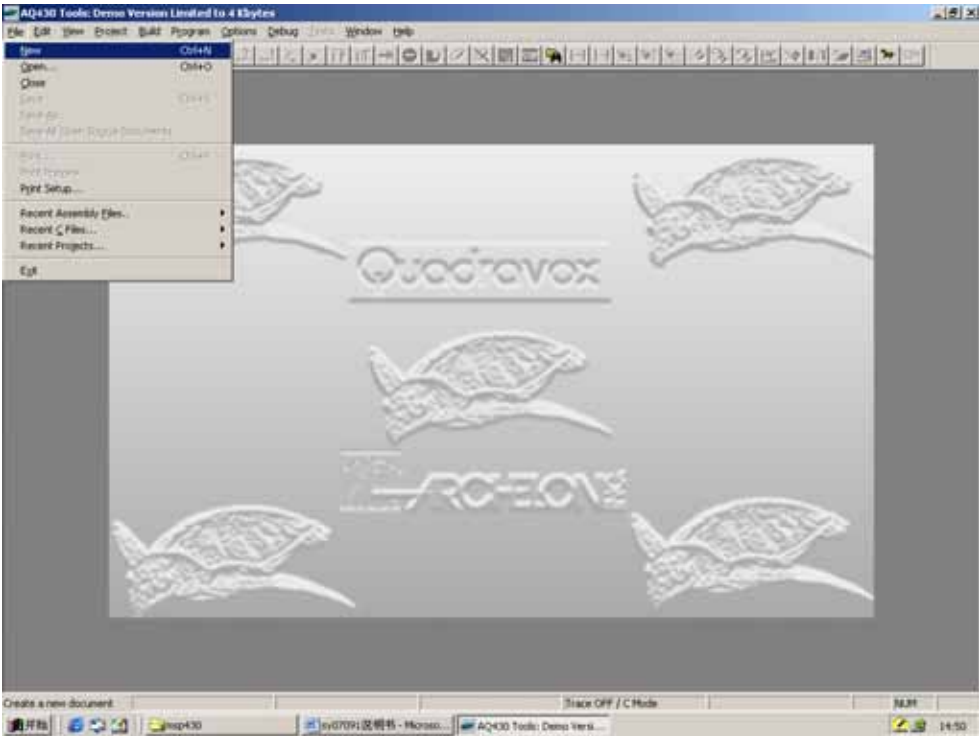
将专用“MSP430 Flash Emulation Tool”下载器的14pin扁平电缆插头与开发板上的ISP_JTAG口相连，下载器的另一端25pin端口直接连接到PC机的打印机输出（Lpt1）端口上，再将SY_07011开发板接上+5V电源（或AC8V~12V）。电源接通后，开发板上红色“Power”指示灯点亮，出厂时芯片内已有程序，会自动进入程式演示状态——程序自动将LED数码管上电显示并且清零，并进入累加计数状态状态进行演示，另有三个红色发光管交替闪亮。说明电路连接正常，可以进行下一步的操作学习。

2 - 4 . 运行调试软件

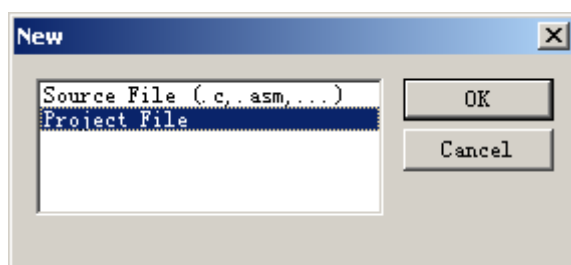
1. 运行桌面上的“  ”图标，进入



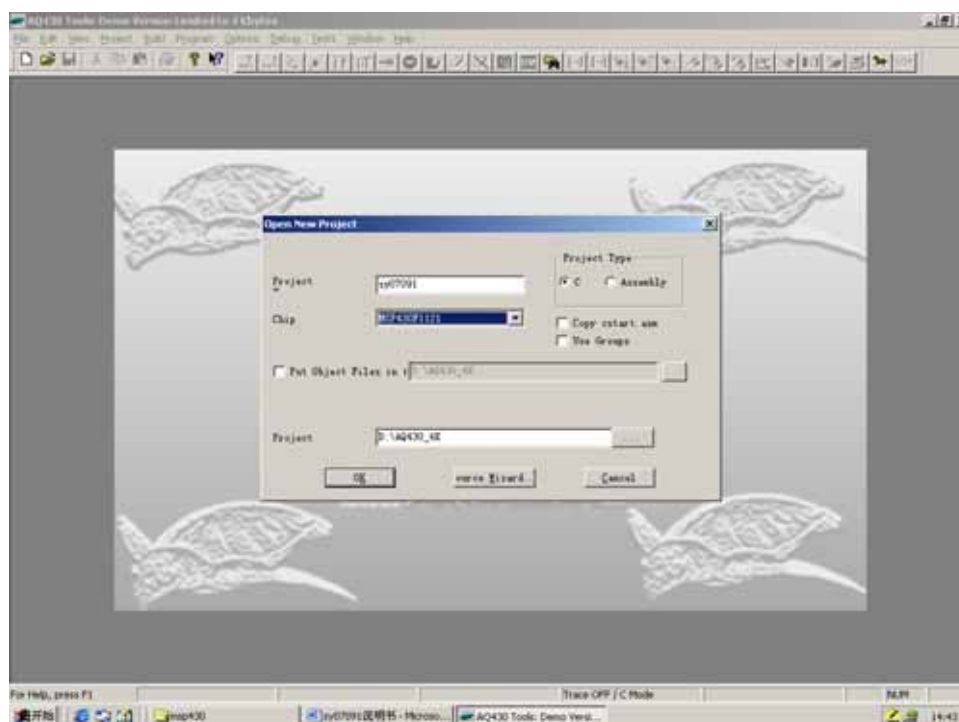
2. 在“ file\new ”下，新建一工程文件，例如“ demo430 ”



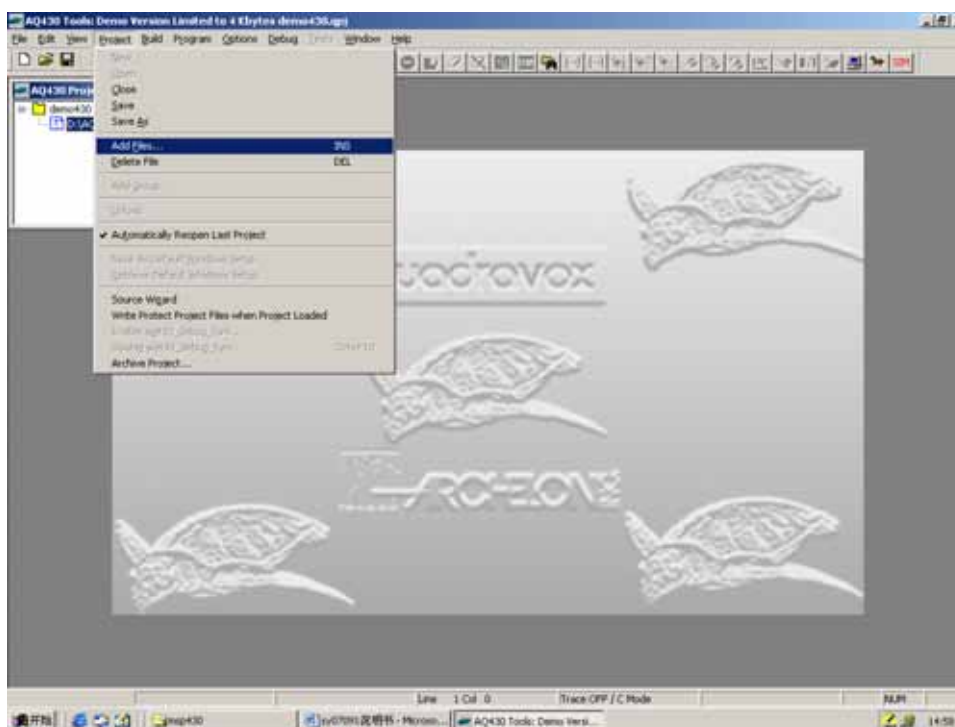
3. 选择 “ project ” ， 点击 “ OK ” 键



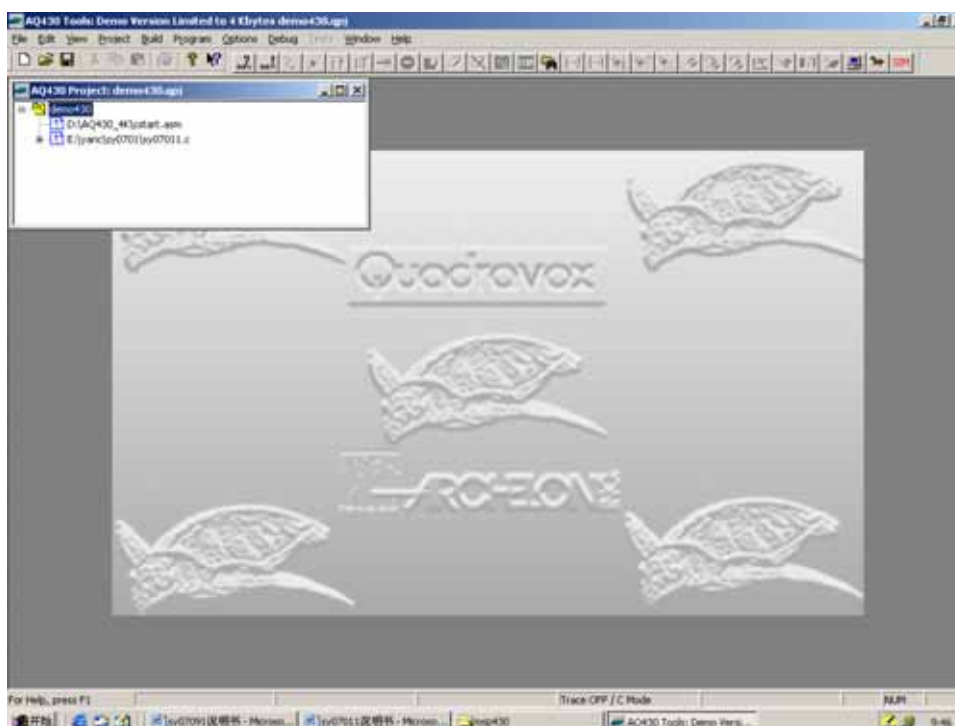
4. 填写项目文件名“ demo430 ”,选择芯片型号“ MSP430F1121 ”， 点击 “ OK ” 键。



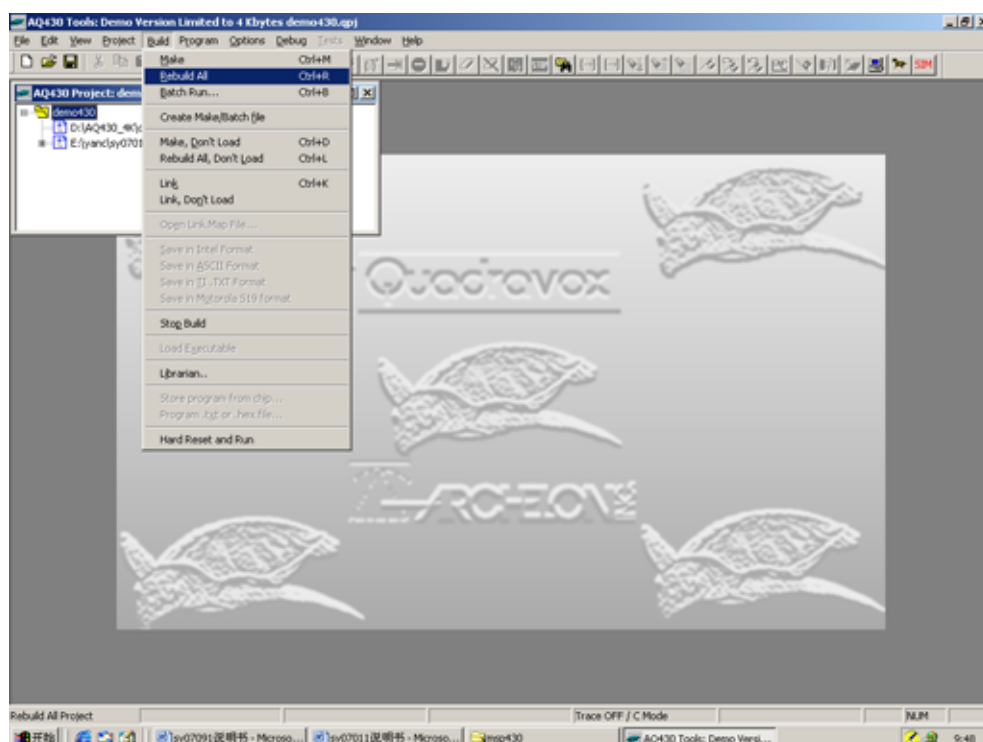
5. 在“Project”下，有选项“Add File”下。添加示范用的源程序。



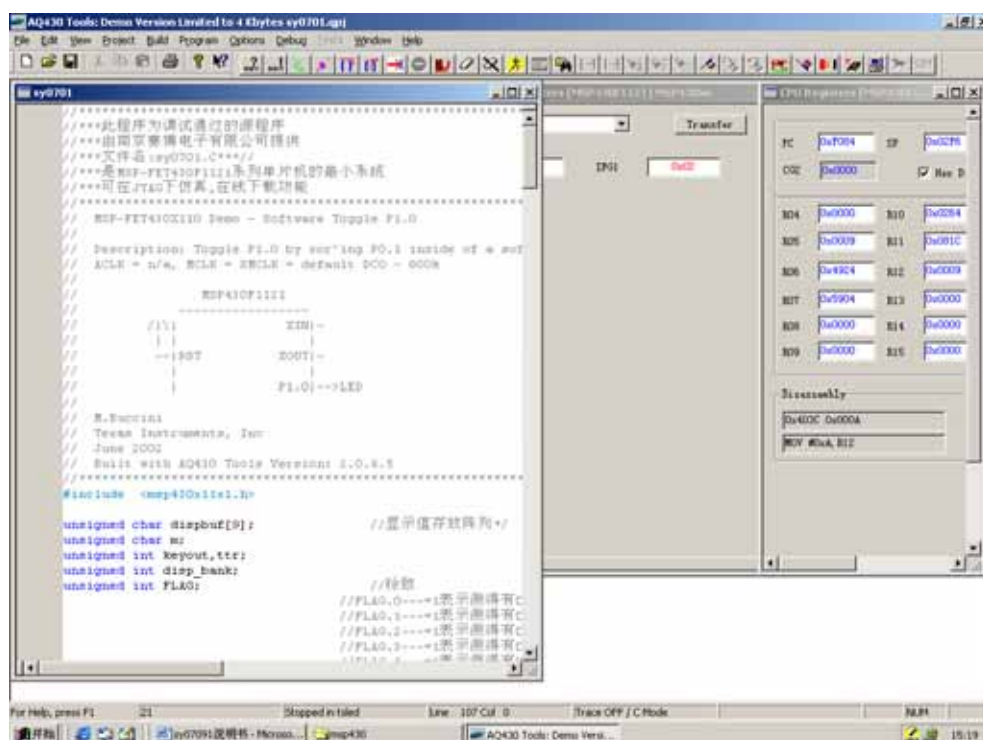
6. 选择已编写好的源程序加入即可。如下图：



7. 在“Build / ReBuild All”下，进行编译/连接/下载。



8. 如编译/连接/下载无错误时，进入下图的仿真调试界面。

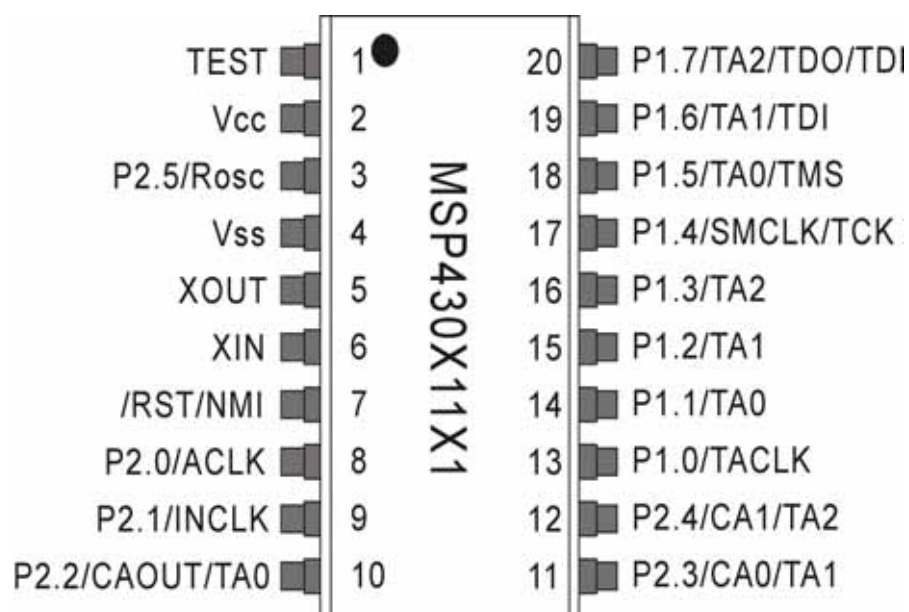


9. 在“ Debug ”下，点击“ Go/Run ”等其它选项，即可进行仿真调试您的源程序。
10. 观察开发板上的运行状态，如与设计目标状态相同——
恭喜您！！已经学会了使用本套开发装置，能轻松自如的在 MSP430 系列单片机的天空中翱翔。

第三章：开发板用器件资料及说明

3—1 . MSP430F1121

芯片平面引脚功能如图：



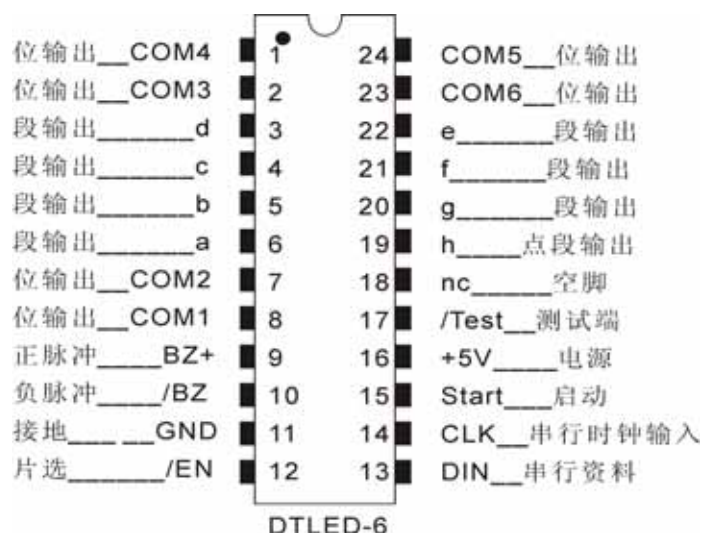
芯片技术特性和接口特性：

- 低电源电压范围 1.8~3.6V 供电。
- 超低功耗:低工作电流，160uA/在 1Mhz/2.2V；1.3uA/在 4Khz/2.2V。同其它单片机相比，可大大延长电池使用寿命。
- 6us 微秒的启动时间可以使启动更加迅速。
- 五个省电模式(备用模式 :0.8uA,RAM 保持关闭模式 :0.1uA)
- 基本时钟系统 (片内 DCO+一个或两个晶体振荡器)
- 16 位精简指令结构，125ns 指令周期时间。
- 有 14 个 I/O 端口可用，均具有中断功能。
- 精确的模拟比较器，配外接器件的斜率 A/D 转换器。

- 看门狗定时器/通用一目的定时器。
- 16 位定时器具有 3 个捕获/比较寄存器和 PWM 输出。
- 串行在线编程，无需外部编程电压，安全熔丝可编程代码保护。

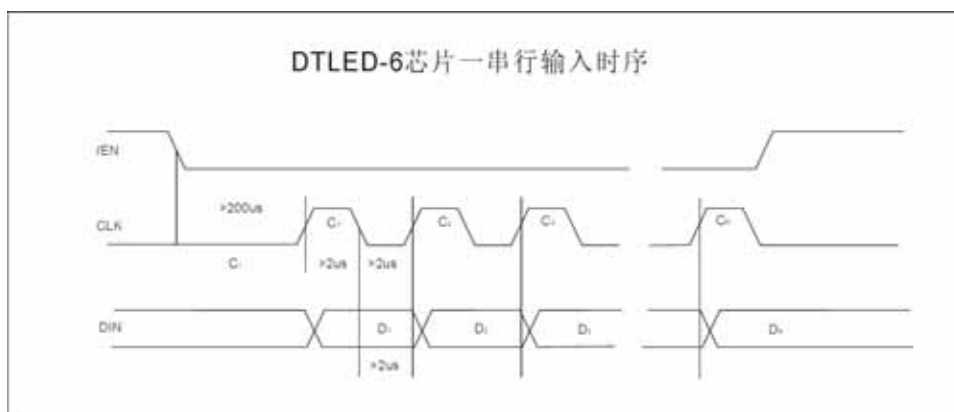
3 - 2 . DTLED-6 串行接口 BCD 译码显示片

芯片平面引脚功能如图：



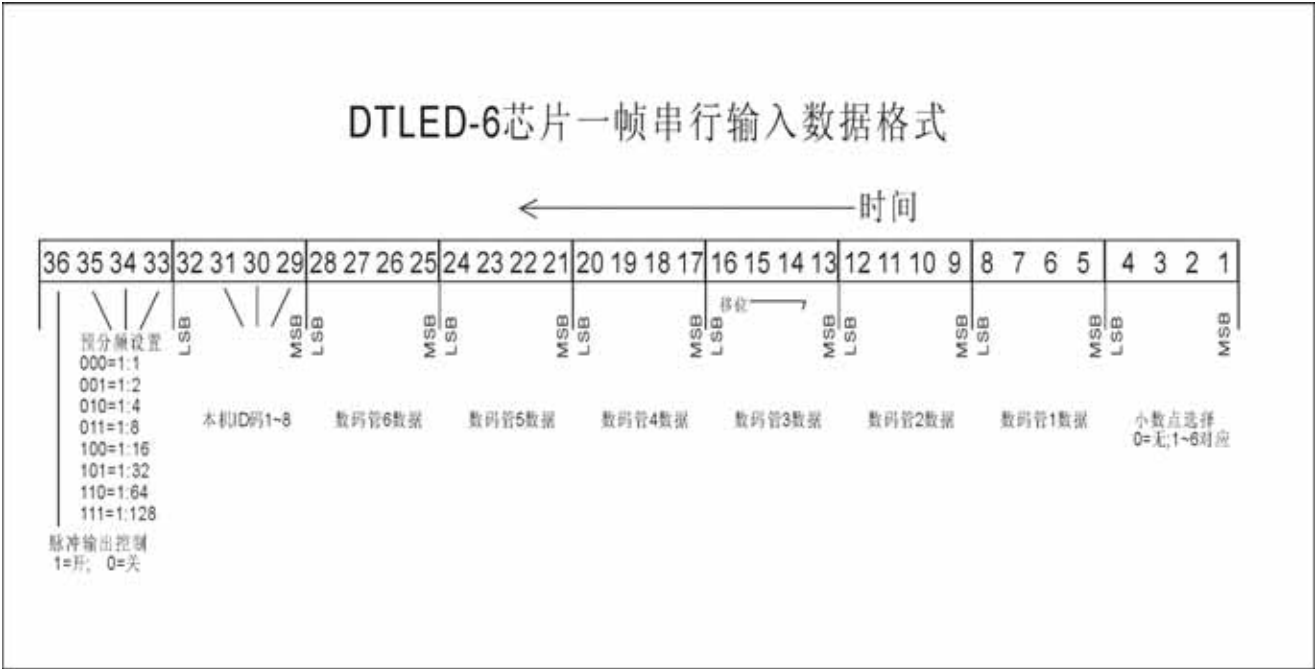
芯片上的 P17 脚为测试端。工作时为“1”电平；

当其为“0”电平时，P5/P4/P3/P2/P22/P21/P20/P19 为零电平，输出端接的数码管以次全亮，P9/P10 端有一对反向脉冲输出（蜂鸣器响）。



“DTLED-6”芯片是在摩托罗拉的“MC14499”芯片上延伸出来的（可参阅MC14499的使用），与“MC14499”芯片的指令相同，并增加更多的功能，一片芯片可显示6个数码管，而且还可N片芯片并在一条线上，同时驱动多组N*6个数码管，自带芯片ID码识别功能，购买时可以选定ID（本机中使用的ID=01），市价只在几元钱，是同产品中的性价比极高的优选器件。由于DTLED-6片内具有BCD译码器和串行接口，所以它可以与任何单片机接口相连。DTLED-6每一次可接收36位串行输入数据，32位串行数据依序提供了6个数码管小数点的位选择，6位数码管的BCD码，本机ID识别码，蜂鸣器分频输出码。

其串行输入的时序如图所示：



前4位为0~6选项，对应控制6个数码管的小数点是否显示，0则不显示小数点。（软件中设为3，对应于第三个数码

管的小数点亮。)

5~28 位 (4*6 位) 是 6 个数码管显示值输入数据其相应的字符如表所示：BCD 码显示字符表

0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

29~32 位是本芯片的识别 ID 码 ,用于在多片芯片并联使用时的识别检查。(软件中设为 1 , 选用 “ DTLED-6 ” 芯片时选用 ID=1 的即可。)

32~36 位是一对蜂鸣器输出控制和预分频输出设置。36 位 =1 时 , 蜂鸣器输出开 ; 36 位=0 时 , 蜂鸣器输出关 ; 35/34/33 组成 0~7 级的预分频数。

第四章：开发板器件表附件清单

4—1．单片机用源程序（仅供参考）

```
//=====

/**此程序为调试通过的源程序

/**由南京赛博电子有限公司提供

/**文件名:sy07011.C

/**是 MSP-FET430F1121 系列单片机的最小系统

/**可在 JTAG 下仿真,在线下载功能

// Built with AQ430 Tools Version: 2.0.6.5

/*******

#include <msp430x11x1.h>

unsigned char dispbuf[9];          //显示值存放阵列*/

unsigned char m;

unsigned int keyout,ttr;

unsigned int disp_bank;

unsigned int FLAG;                //FLAG.0---=1 表示测得有按键

                                   //FLAG.1---=1 表示测得有按键

                                   //FLAG.2---=1 表示测得有按键

                                   //FLAG.3---=1 表示测得有按键

                                   //FLAG.4---=1 表示测得有按键

                                   //FLAG.5---=1 表示测得有按键
```

```
//FLAG.6---=1 表示测得有按键

//FLAG.7---=1 表示测得有按键

//FLAG.8---=1 表示测得有按键

//FLAG.9---=1 表示测得有按键

void delay (unsigned int value)      /*延时副程式*/
{
    while (value!=0) value--;      /*10US 延时*/
}

void tsled(void);      //串行发送 DTLED-6 显示副程式*/

void KeyProess(void);

void scan(void);

void scan_set(void);

void xch(void);      /*显示向右移动副程式*/

void xchl(void);      /*显示向右移动副程式*/

void ledput(void);

//=====

void main(void)

{
    WDTCTL = WDTPW + WDTHOLD;      // Stop watchdog timer

    dispbuf[0]=1;      //显示值存放阵列*/
    dispbuf[1]=0;      //显示值存放阵列*/
    dispbuf[2]=0;      //显示值存放阵列*/
    dispbuf[3]=0;      //显示值存放阵列*/
    dispbuf[4]=0;      //显示值存放阵列*/
    dispbuf[5]=0;      //显示值存放阵列*/
}
```

```
    dispbuf[6]=0;                //显示值存放阵列*/
    dispbuf[7]=1;                //显示值存放阵列*/
    dispbuf[8]=1;                //显示值存放阵列*/
    P2DIR = 0xff;                // Set P2 to output direction
    P2OUT = 0x07;                // Toggle P2.0,p2.1,p2.2 using Hi test
    tsled();                     //串行发送 DTLED-6 显示副程式*/

while(1)
{
    { // delay(3000);

    tsled();                     //串行发送 DTLED-6 显示副程式*/
    KeyProess();                //键位扫描读入
    ledput();

    xch1();                      /*显示向右移动副程式*/
    }
}

//=====
//串行发送 DTLED-6 显示
//dispbuf0-dispbuf1-dispbuf2-dispbuf3-dispbuf4-dispbuf5-dispbuf6-dispbuf7-dispbuf8
// 小数点 - 数码管 1- 数码管 2- 数码管 3- 数码管 4- 数码管 5- 数码管 6-ID 识别码-BZ 识别码
//00000xxx-0000xxxx-0000xxxx-0000xxxx-0000xxxx-0000xxxx-0000xxxx-0000xxxx-0000xxxx-0000xxxx
//小数点=1~6,其它为不显示

//数码管 1-6=0~15 ,
//ID 识别码=0~15
//BZ 识别码=BZ.3 为蜂鸣输出控制位 , 0=关闭 , 1=打开
// BZ0~2 为蜂鸣频率输出设定为 , 000=不分频 , 最高频率输出=4800Hz ;
//
//                                001=一次分频 , 频率输出=4800/ 2 =2400Hz ;
```

```
//          010=二次分频，频率输出=4800/ 4 =1200Hz；
//          011=三次分频，频率输出=4800/ 8 =600Hz；
//          100=四次分频，频率输出=4800/ 16=300Hz；
//          101=五次分频，频率输出=4800/ 32=150Hz；
//          110=六次分频，频率输出=4800/ 64=75Hz；
//          111=七次分频，频率输出=4800/128=38Hz；

//=====

void tsled(void)          /*串行发送 DTLED-6 显示副程式*/
{   char i,j;

    unsigned int ts0,ts1;

    if(displ_bank==(displbuf[6]+displbuf[5]+displbuf[4]+displbuf[3]+displbuf[2]+displbuf[1]))

        delay(10);

    else

        {   P2OUT &=0xdf;          //clk=0;

            P2OUT &=0xf7;          //en=0;

            delay(10);

            for(j=0;j<9;j++)

                { ts0=displbuf[j];

                    for(i=0;i<4;i++)          //发小数点码 8 , 4 , 2 , 1*/

                        {P2OUT |=0x20;          //clk=1;

                            ts0 =ts0<<1;

                            ts1 =ts0&0x0010;

                            if(ts1==0)
```

```
        P2OUT &=0xef;        //dout=0;

    else    P2OUT |=0x10;      //dout=1;

    P2OUT &=0xdf;            //clk=0;

    delay(3);

    }

}

P2OUT |=0x08;              //en=1;

disp_bank=(dispbuf[6]+dispbuf[5]+dispbuf[4]+dispbuf[3]+dispbuf[2]+dispbuf[1]);

}

}    /*返回主程式*/

//=====

void KeyProess(void)        //3*4 键盘矩阵

{    scan();

    if(FLAG&0x01==1)

    {scan_set();

    FLAG &=0xfe;

    }

    delay(10);

}

//=====

void scan(void)              // 扫描键盘副程式*/

{unsigned char a1=0x1,i;    // A1=0XF7 列扫描初值，I 行*/

    for(i=0;i<4;i++)        // 键盘 4 个扫描列
```

```
{  P1DIR =a1;                // 键盘口 P1.0~P1.3,为输出状态

  P1OUT =~a1;                // 键盘口 P1.0~P1.3,为高电平输出

  m =P1IN&0x0f;              // 键盘口 P1.0~P1.3,为高电平输出

  switch(m)

  {

    case 0x03: keyout=m;

        delay(100);

        FLAG |=0x01;        //是则设 FLAG.0=1 表有按键

        break;              //跳出此循环*/

    case 0x05: keyout=m;

        delay(100);

        FLAG |=0x01;        //是则设 FLAG.0=1 表有按键

        break;              //跳出此循环*/

    case 0x06: keyout=m;

        delay(100);

        FLAG |=0x01;        //是则设 FLAG.0=1 表有按键

        break;              //跳出此循环*/

    case 0x09: keyout=m;

        delay(100);

        FLAG |=0x01;        //是则设 FLAG.0=1 表有按键

        break;              //跳出此循环*/

    case 0x0a: keyout=m;

        delay(100);
```



```
                FLAG |=0x01;        //是则设 FLAG.0=1 表有按键

                break;                //跳出此循环*/

        case 0x0c: keyout=m;

                delay(100);

                FLAG |=0x01;        //是则设 FLAG.0=1 表有按键

                break;                //跳出此循环*/

//        default:    break;        //跳出此循环*/

    }

    if(FLAG&0x01==1)

        i=4;

        else a1=a1<<1;        //高位补 1 , 由于 P1.7~P1.4 未接+5V , 而是由指令

加载高电平

    }

    delay(10);

}

//=====

void scan_set(void)                //扫描键盘副程式

{ unsigned char a1=0x1,i,k;        //A1=0XF7 列扫描初值 , I 行

    unsigned char BR7=0,kk;        //行扫周期

    unsigned int q;

    for(q=0;q<500;q++);            //防抖动延迟

        for(i=0;i<4;i++)            //键盘 4 个扫描列

            {    BR7 =a1<<4;
```

```
P1DIR =a1;                // 键盘口 P1.0~P1.3,为输出状态

P1OUT =~a1;               // 键盘口 P1.0~P1.3,为高电平输出

m =(BR7)|(P1IN&0x0f);

delay(10);

switch(m)                  //取行的高 4 位元，侦测那一行被按
{
    case 0x16: keyout=1; //第一行被按否？是则扫描指标=列 X4
        FLAG &=0xfe; //FLAG.0=0 表示已处理过或无按键
        break;      //跳出此循环*/
    case 0x1a: keyout=2; //第一行被按否？是则扫描指标=列 X4
        FLAG &=0xfe; //FLAG.0=0 表示已处理过或无按键
        break;      //跳出此循环
    case 0x1c: keyout=3; //第一行被按否？是则扫描指标=列 X4
        FLAG &=0xfe; //FLAG.0=0 表示已处理过或无按键
        break;      //跳出此循环
    case 0x25: keyout=4; //第一行被按否？是则扫描指标=列 X4
        FLAG &=0xfe; //FLAG.0=0 表示已处理过或无按键
        break;      //跳出此循环
    case 0x29: keyout=5; //第一行被按否？是则扫描指标=列 X4
        FLAG &=0xfe; //FLAG.0=0 表示已处理过或无按键
        break;      //跳出此循环
    case 0x2c: keyout=6; //第一行被按否？是则扫描指标=列 X4
        FLAG &=0xfe; //FLAG.0=0 表示已处理过或无按键
        break;      //跳出此循环
```

```
        case 0x43: keyout=7;      //第一行被按否？是则扫描指标=列 X4
                FLAG &=0xfe;      //FLAG.0=0 表示已处理过或无按键
                break;            //跳出此循环

        case 0x49: keyout=8;      //第一行被按否？是则扫描指标=列 X4
                FLAG &=0xfe;      //FLAG.0=0 表示已处理过或无按键
                break;            //跳出此循环

        case 0x4a: keyout=9;      //第一行被按否？是则扫描指标=列 X4
                FLAG &=0xfe;      //FLAG.0=0 表示已处理过或无按键
                break;            //跳出此循环

        case 0x85: keyout=0;      //第一行被按否？是则扫描指标=列 X4
                FLAG &=0xfe;      //FLAG.0=0 表示已处理过或无按键
                break;            //跳出此循环*/

        case 0x83: keyout=0xa;    //第一行被按否？是则扫描指标=列 X4
                FLAG &=0xfe;      //FLAG.0=0 表示已处理过或无按键
                break;            //跳出此循环*/

        case 0x86: keyout=0xc;    //第一行被按否？是则扫描指标=列 X4
                FLAG &=0xfe;      //FLAG.0=0 表示已处理过或无按键
                break;            //跳出此循环

        default: break;          //跳出此循环

    }

    k=FLAG&0x01;

    if(k==0)

        {do
```

```
    kk =(BR7)|(P1IN&0x0f);

    while(m ==kk);           // 键盘口 P1.0~P1.3,为高电平输出

    xch();                   //显示向右移动副程式*/

    i=4;

    }

    else  a1=a1<<1;          //高位补 1 , 由于 P1.7~P1.4 未接+5V , 而是由
指令加载高电平

    }

}                             /*返回主程式*/

//=====

void ledput(void)    //

{  unsigned char i;

    unsigned char swe;

    ttr++;

    if(ttr==500)

    {ttr=0;

    i =P2OUT&0xf8;

    swe =(P2OUT&0x07);

    if(swe==0)

    {swe =BIT0;

    dispbuf[1]++;           //显示值存放阵列*/

    }

    else  swe=swe<<1;
```

```
P2OUT =swe|i;

}

}

//=====

void xch(void)                //显示向右移动副程式*/
{char c;                     //宣告变数*/

    for(c=6;c>0;c--)

        { dispbuf[c]=dispbuf[c-1];        //将显示器 dispbuf[]移位交换

        }

    dispbuf[0]=0;                //有安键时,小数点不亮

    dispbuf[1]=keyout;           //新的按键值存入 dispbuf[1],最低位

}                                //返回上一层程式

//=====

void xch1(void)               //显示向右移动副程式

{char c;

    if(dispbuf[1]>9)

        { dispbuf[1]=0;dispbuf[2]++;

        if(dispbuf[2]>9)

            { dispbuf[2]=0;dispbuf[3]++;

            if(dispbuf[3]>9)

                { dispbuf[3]=0;dispbuf[4]++;

                if(dispbuf[4]>9)

                    { dispbuf[4]=0;dispbuf[5]++;
```

```
        if(disobuf[5]>9)

            { disobuf[5]=0;disobuf[6]++;

                if(disobuf[6]>9)

                    { disobuf[6]=0;

                        }

                    }

            }

        }

    }

}

//=====
```

4 —2 . 电路原理图

请见附录（一）

4 - 3 . 包装清单

- a . 主机板 * 1 块
- b . Altera ISP 下载线 * 1 只
- c . SOP20 转接板 * 1 块
- d . 使用说明书 * 1 本
- e . 软件光盘 * 1 张（包含软件）
 - a) SY_07011.c 测试用源程序
 - b) SY_07011 开发板电原理图
 - c) PowerLogic 绘制电原理图软件

- d) UltraEdit 编辑软件
- e) AQ4302K_4K 编译调试软件
- f) FET_R306 编译调试软件
- g) 开发板上用芯片资料

第五章：赛博其它 51 类实验板简介

5 - 1 . 51DEMO I/O 板简介



51XXDEMO 板结构简单，使用方便，是检测仿真器端口的最佳工具板。51XXDEMO 板的 P0，P1，P2，P3 各口已直接联接上四个七段 LEO 数码管显示模块，并配有演示示范程序，将板上单片机取下，插上 51 系列仿真器的仿真头后，可以直接监测到仿真器 P0，P1，P2，P3 端口的输出状态。是用户用来检测仿真器输出端口的最好工具，板上并扩展了液晶显示的模式。适用于初学者及数学试验和直接用于产品开发。此产品还提供了方便实用的双电源供电插座，DC+5V 或 AC/8-12V 电源均可。

详情请见该产品的资料说明（型号为：SY-51DEMO）。

5 - 2 . A/D-8X51 数模转换开发实验板简介



A/D-8X51 模数转换板像其它 51 类的开发实验板一样，上电复位后进入启动工作状态，程序自动的控制 AD0804 模数转换器，将 0—5 伏的模拟电压转换成数字电压，供 8X51 单片机处理后送 LED 数码管显示；在本案中是采用了 AD590 的温度传感器将变化的温度转换成变化的电压，再将电压的变化值转换成对应的温度显示出来；如测到的温度在设定的上下限之间，绿色的正常指示灯亮，反之温度大于或小于设定温度的上下限，红色（HI/LOW）指示灯将分别闪亮，表示报警。

板上的配制有：

- a . 标准的 8X51 应用电路设计。
- b . 带 4 位数码管显示 LED , 可显示 3 位数电压或转换成的温度。
- c . 自设 4 个功能设置按键 , 或用于上下 (加减) 调整之用。在板上分别为是 MODE , SET , UP , DOWN。
- d . 有方便实用的双电源供电插座 , DC+5V 或 AC9V-12V 电源均可。
- e . 有三路 LED 显示的控制输出指示 , 并可直接在板上再扩展输出控制。
- f . 预留 8X51 所有端口的连接插口 , 便于用户二次开发成成品。

详情请见该产品的资料说明 (型号为 : SY-1032)。

5 - 3 . 流水灯控制器 (12 路)

外型如下图 :



流水灯控制器是用 89C2051 单片机为机芯 , 可直接使用于

广告灯的循环时序控制以及工业现场控制等。将控制器接上电源 DC-+5V 或 DC-+12V 后,面板上侧的绿色指示灯亮,12 路指示灯自动会象流水似的移动闪烁;如需提高或降低流水指示灯的变换速度,只需按住加速(减速)钮,这时可以看到指示灯在做二进制的减法(加法),松开按钮后指示灯的变换速度则会有相应的变化;按住正/反向钮一次,则会自动变换一次流水灯的循环方向一次。

详情请见该产品的资料说明(型号为:SY-0729)。

5 - 4 . SY_0606 开发板



本产品集单片机的最小系统应用于一体,在设计其电路的过程中,力求简洁明了,资源共享开放,方便实用;为单片机

的二次开发和初学者提供极大的方便。其电路中包含典型的电源整流稳压电路，复位启动电路，晶体振荡时钟电路，键盘输入电路，RS232 串行通讯电路，串行 EPROM 的在线擦写（可随时更改信息密码），LED 数码管扫描显示电路等，精确时钟读写等是单片机初学者的精典范例。也是工程开发人员可随手取及的最佳半成品。

1. 标准的 8X5X 应用电路设计，电源/时基/复位/端口驱动。
2. 自带程序的在线烧录（自下载）功能。
3. 提供 16*2 字符显示液晶板接口(液晶模块需另选配)。
4. 带标准 RS232 接口连接电路。
5. 有掉电密码不消失之功能串行 EPROM 应用。
6. 自带 4*4 标准键盘输入，便于学习者掌握键盘输入和程序编写。
7. 精确标准的时钟电路，（另加备用电源）可直接作为时钟计时用。
8. 用串行驱动方式，驱动 6 位数码管显示，大大节省了单片机的接口资源（详见后面“DTLED-6”芯片介绍）。
9. 有一对蜂鸣器驱动输出在显示芯片上，不占用单片机的口资源。
10. 预留扩展空间及接口，可直接驱动或控制用户设备，便于用户直接开发品。

详情请见该产品的资料说明（型号为：SY-0606）。

5 - 5 . Atmel_ISP 下载线



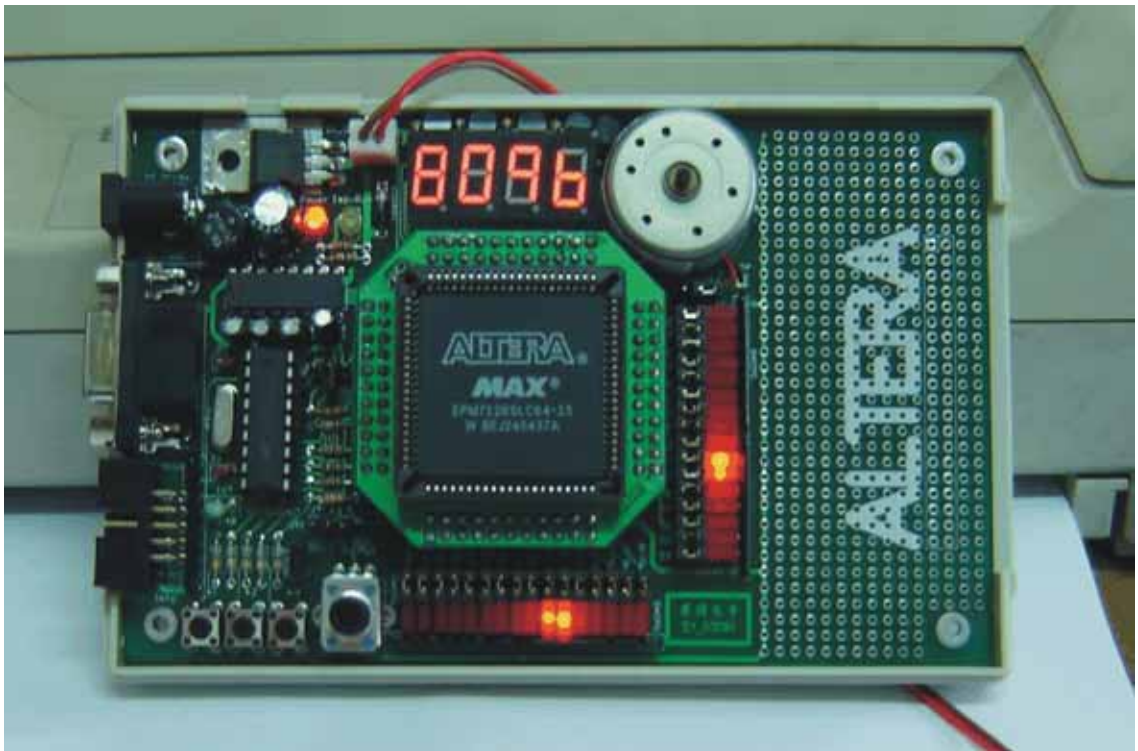
将 ISP 下载线 25pin 接口端连接在 PC 机的并口（打印口）上，10pin 排线端通过排线插在本开发板上的“ Atmel-ISP ”插口座上，开发板接上+5 伏电源，下载线上有红色指示灯亮即可启动下载软件进行下载（烧录）ATMEL 的 AT89S52 系列芯片。

5 - 6 . ALTERA_ISP 下载线（选配自购件）



支持 ALTERA_7XXX 系列，可在 MAX+plusII 操作环境下运行，将 ISP 下载线 25pin 接口端连接在 PC 机的并口(打印口)上，10pin 排线端通过排线插在本开发板上的“ISP-JATG”插口座上，开发板接上+5 伏电源，下载线上有红色指示灯亮即可启动下载软件进行下载（烧录）ALTERA 的 EPMXX 系列芯片。

5 - 7 . ALTERA_CPLD 开发板的特性简介



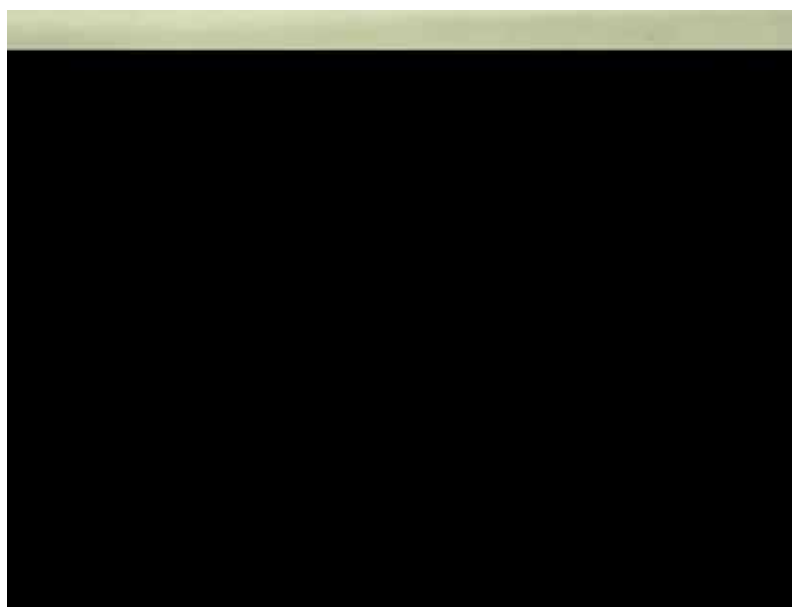
8. 电源适应性强，可随意使用无极性 8~15V 电源或 DC+5V 电源供电。
9. CPLD 时钟是由单片机“HT46R47”产生，并且可随意改变其时钟频率。
10. 本开发板可直接支持 ALTERA 的 EPM7128SLC (PLCC84)；EPM7064SLC (PLCC68)；EPM7064SLC/EPM7032 (PLCC44) 等芯片为主控元件。
11. 有一组四字节数码管显示器硬件电路。
12. 有 2 组 16 位 LED 发光管连接电路，可作 LED 跑马灯实习，

演示。

13. 一路电机驱动输出，可通过电压的调整进行脉宽调制来达到电机的调速，用 JP1 跳线来选择 MCU/CPLD 驱动。
14. 有一路无源蜂鸣器驱动输出，可通过 JP2 跳线来选择 MCU/CPLD 驱动。
15. 有三路按键连接硬件电路。
16. 一路模拟电压调节输入，在单片机的 PA 口上有一组对应的数字电压输出。相当于 9 位分辨率的模数转换提供给 CPLD 供学习用。
17. 弹性空间大，预留扩展空间及接口，可直接驱动或控制用户设备，便于用户直接开发产品。

详情请见该产品的资料说明（型号为：SY-03091）。

5 - 8 . MSP430 Flash Emulation Tool 下载调试器



此“MSP430 Flash Emulation Tool”下载调试器是由利尔达公司生产的专门用于MSP430系列的开发工具，具有价廉物美，适用性强的特点；使用简单可靠，将“MSP430 Flash Emulation Tool”下载调试器的25pin接口端，连接在PC机的并口（打印口）上，14pin排线端通过排线插在本开发板上的“JTAG”插口座上，开发板接上电源即可。



*****公司其它产品简介见软件盘中电子版文件*****