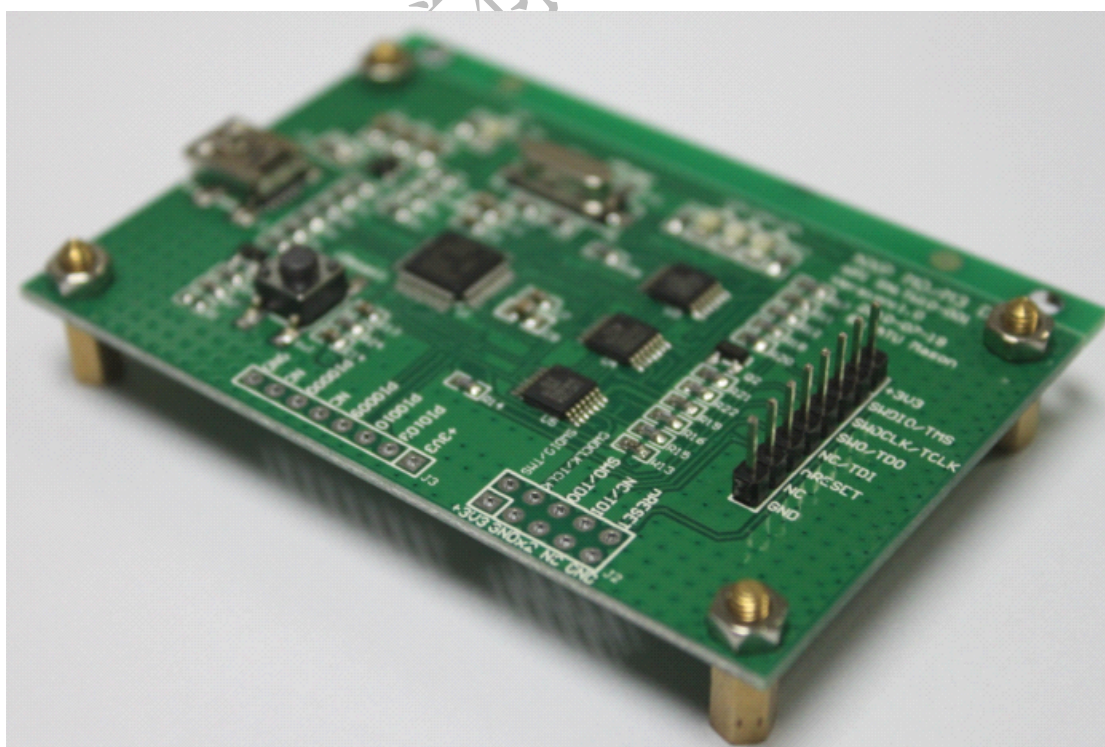
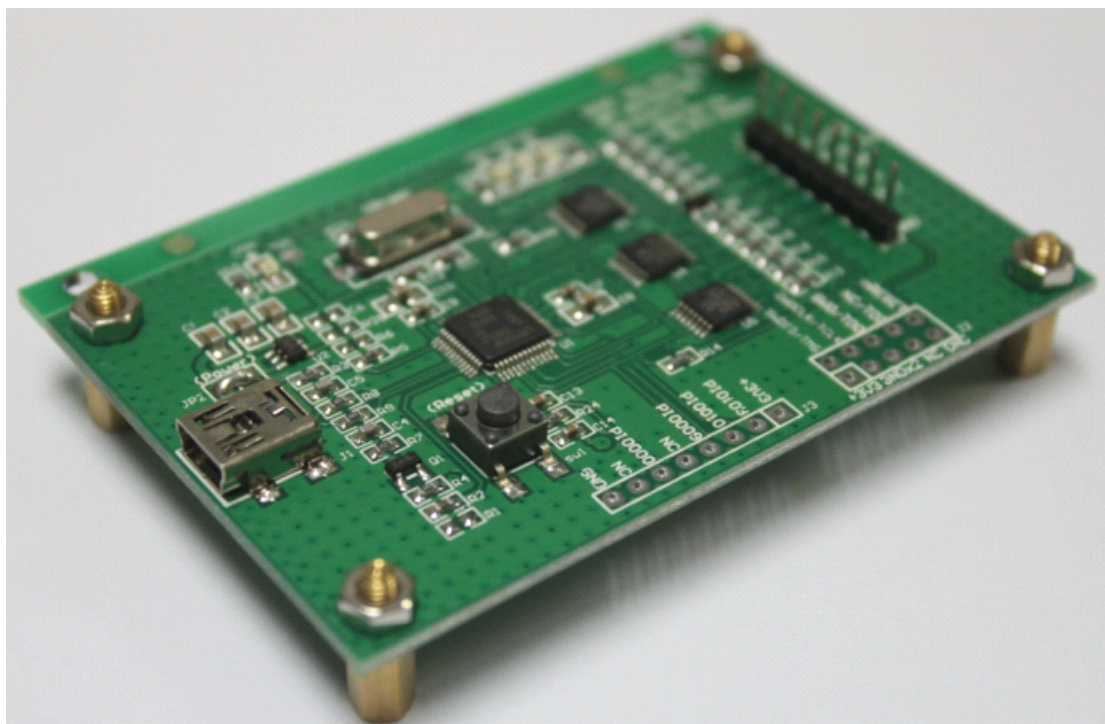
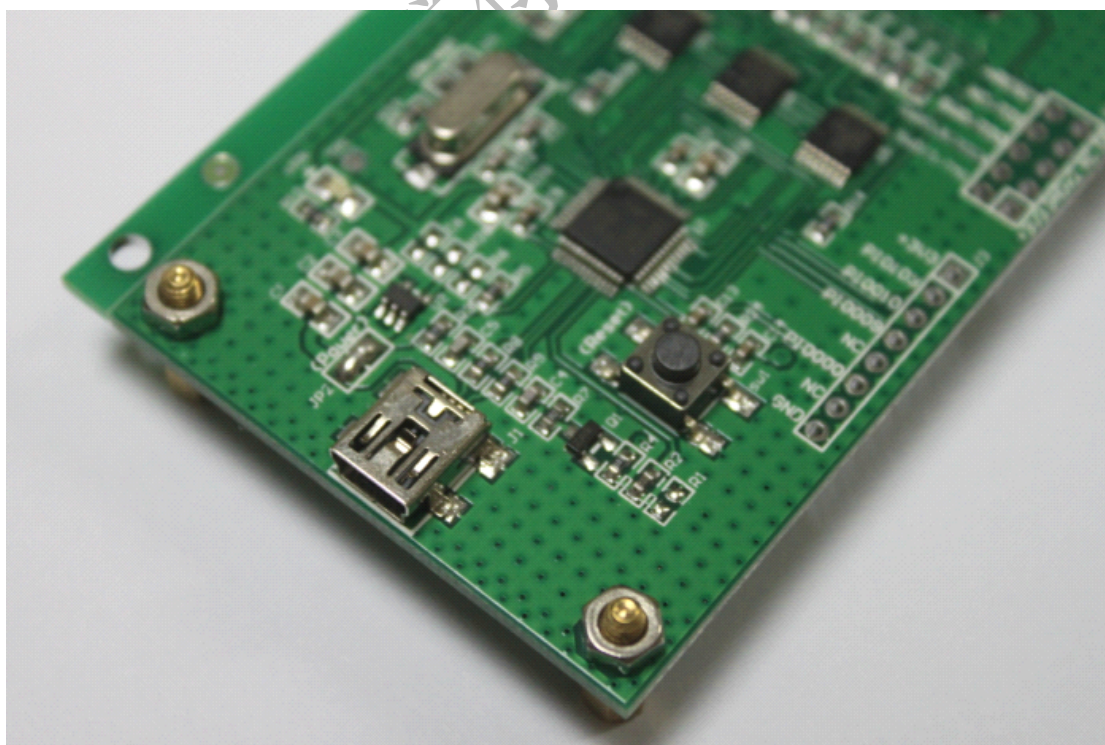
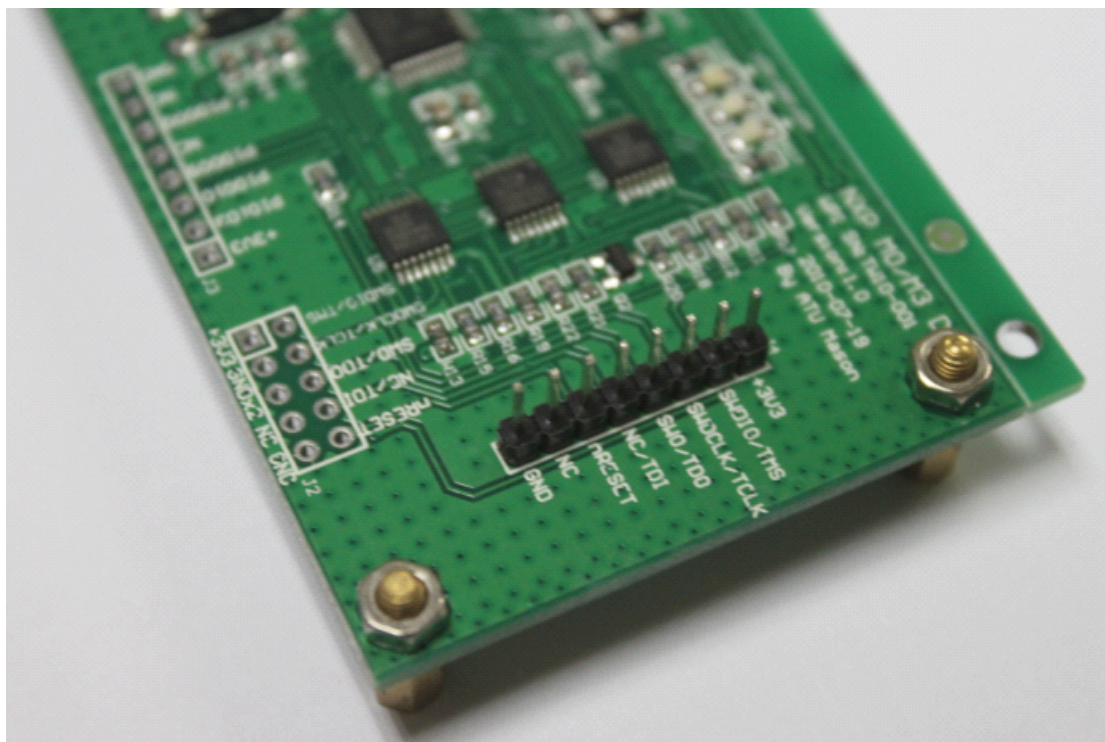
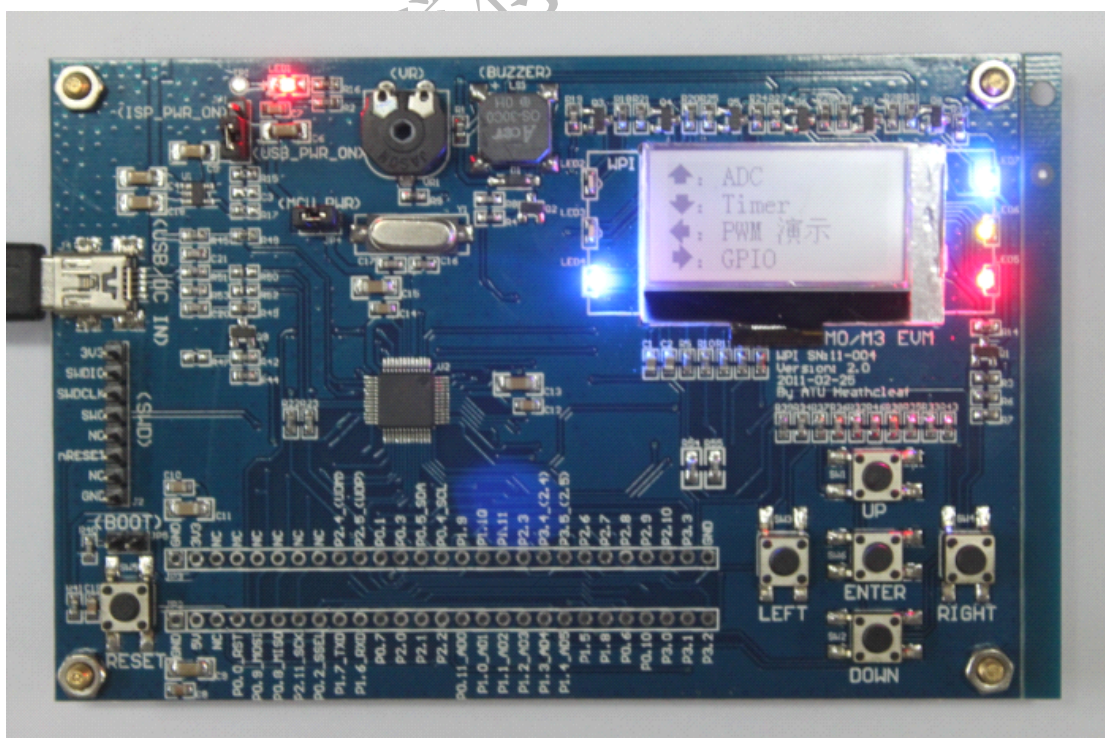
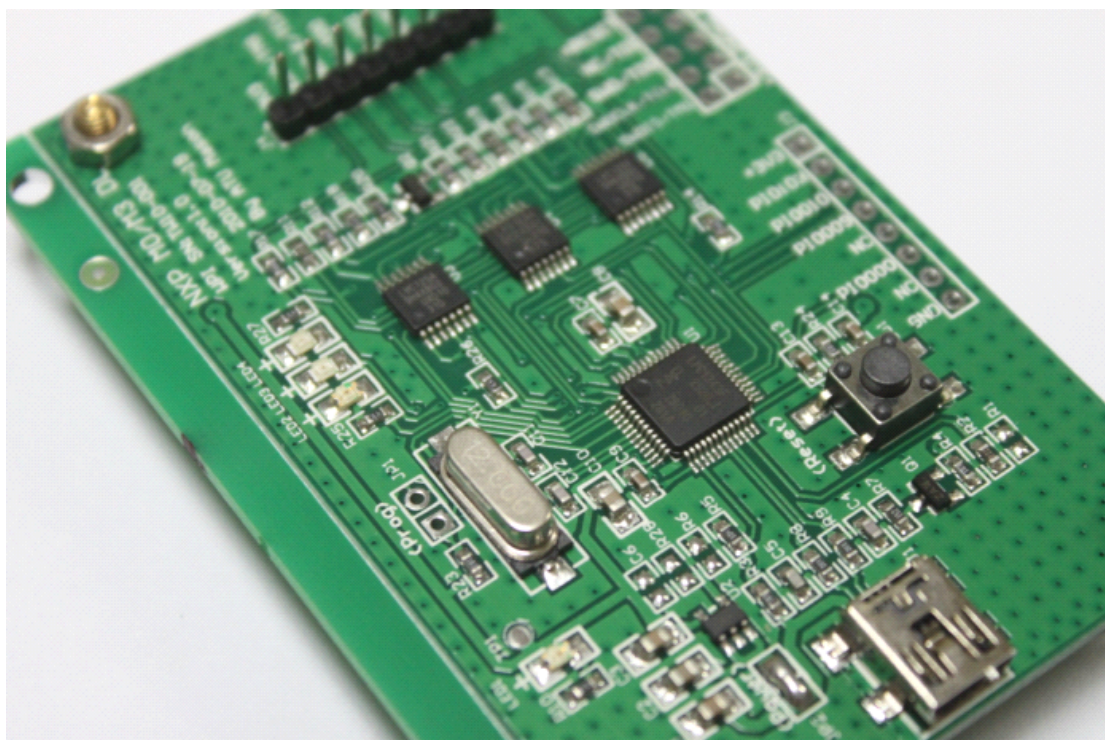
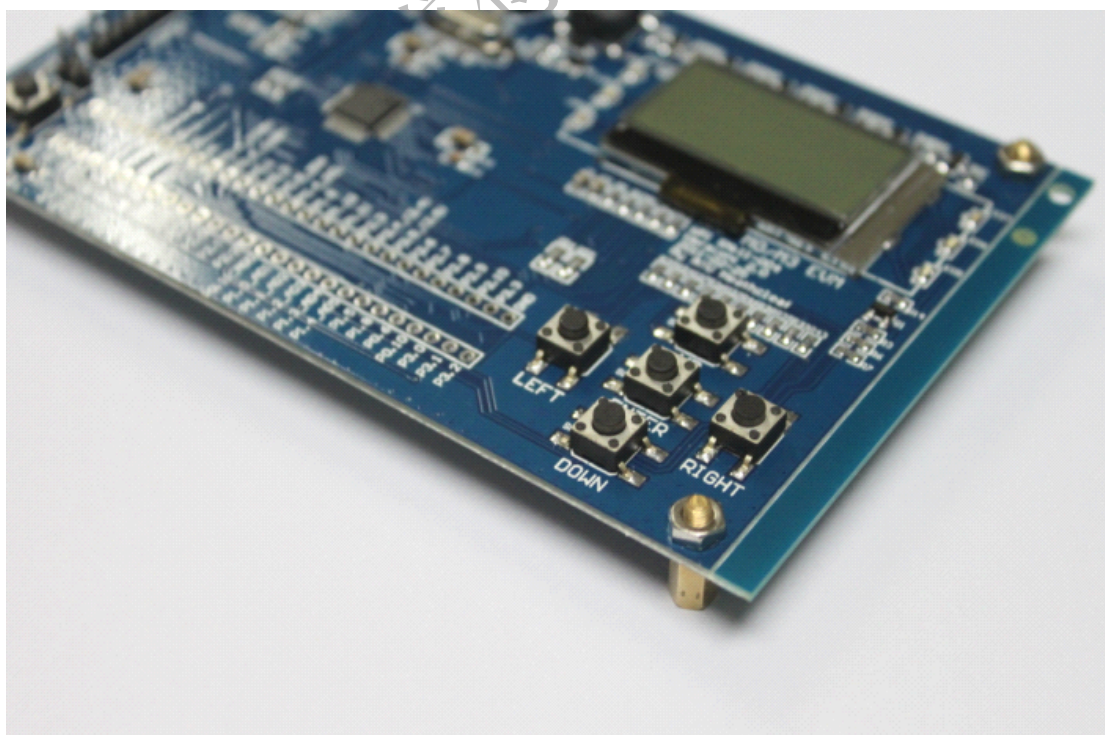
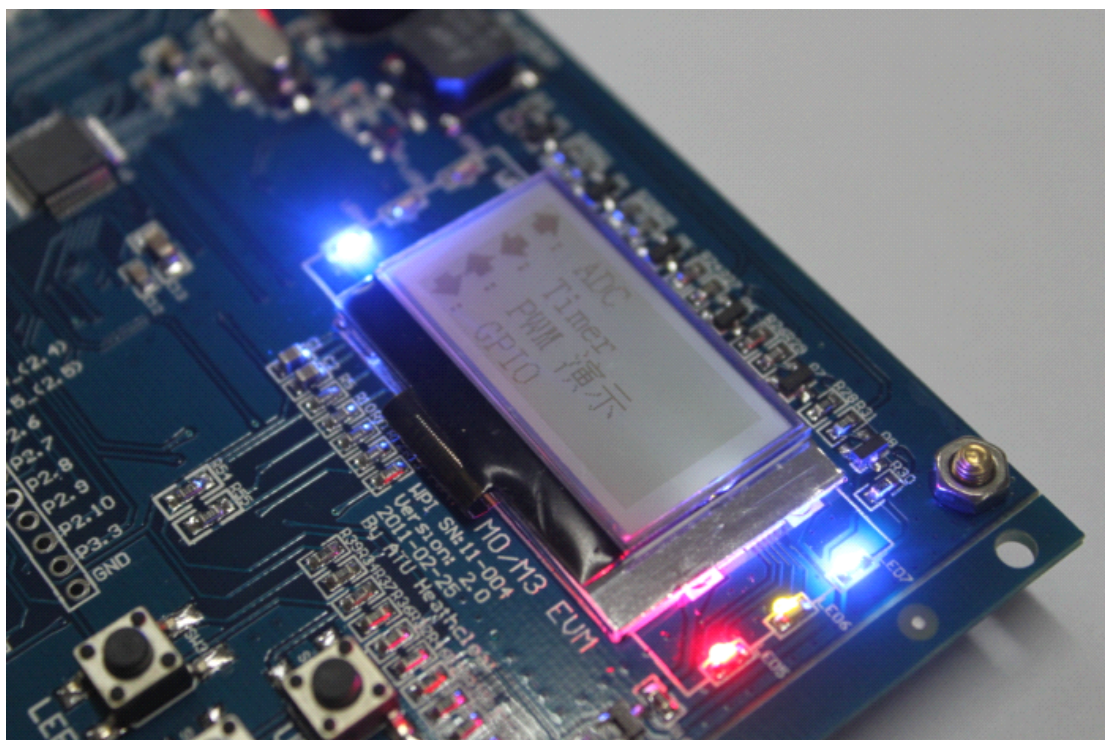


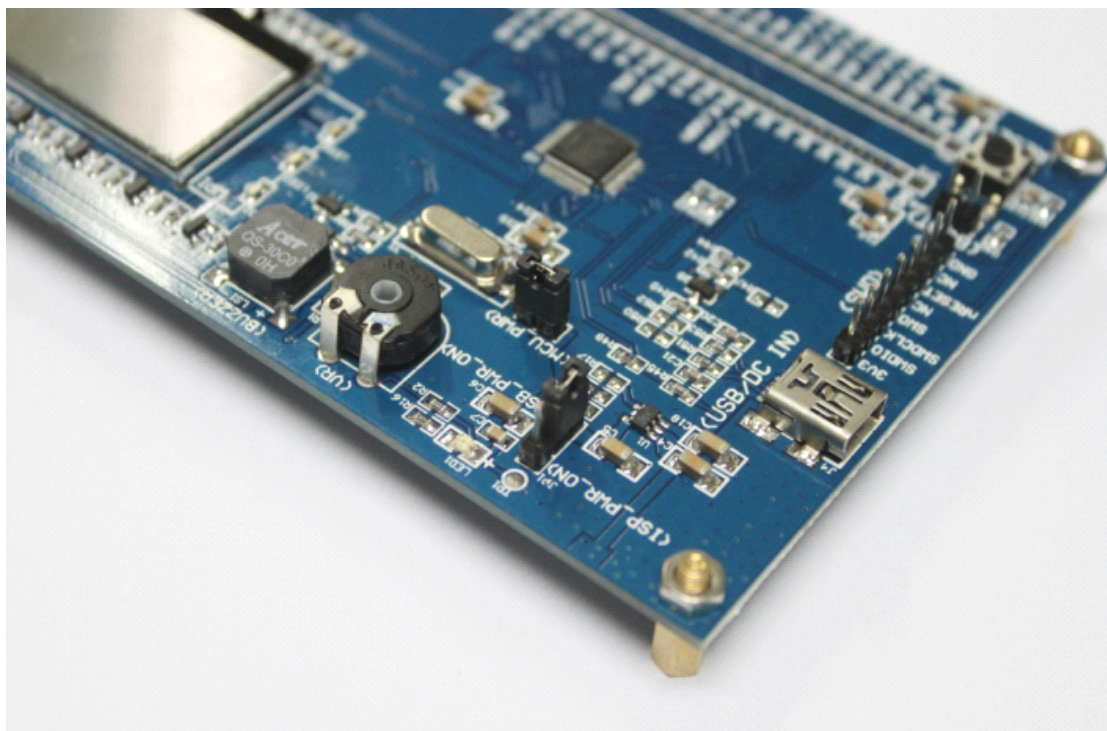
NXP M0 Demo 板



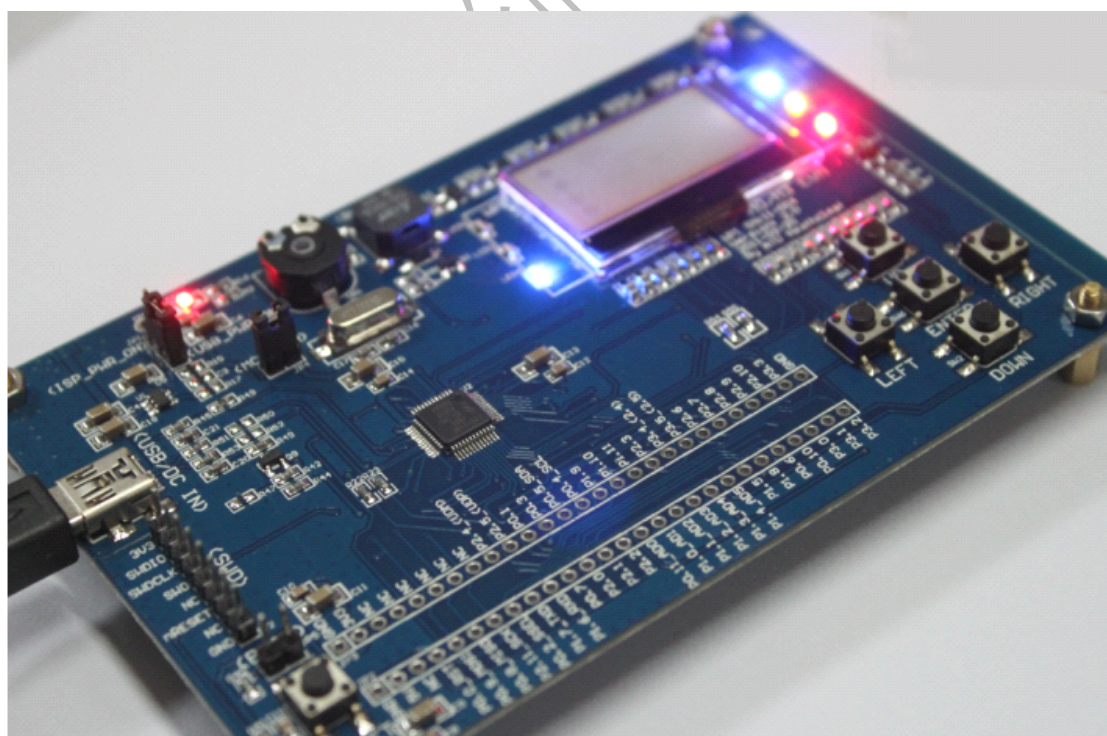


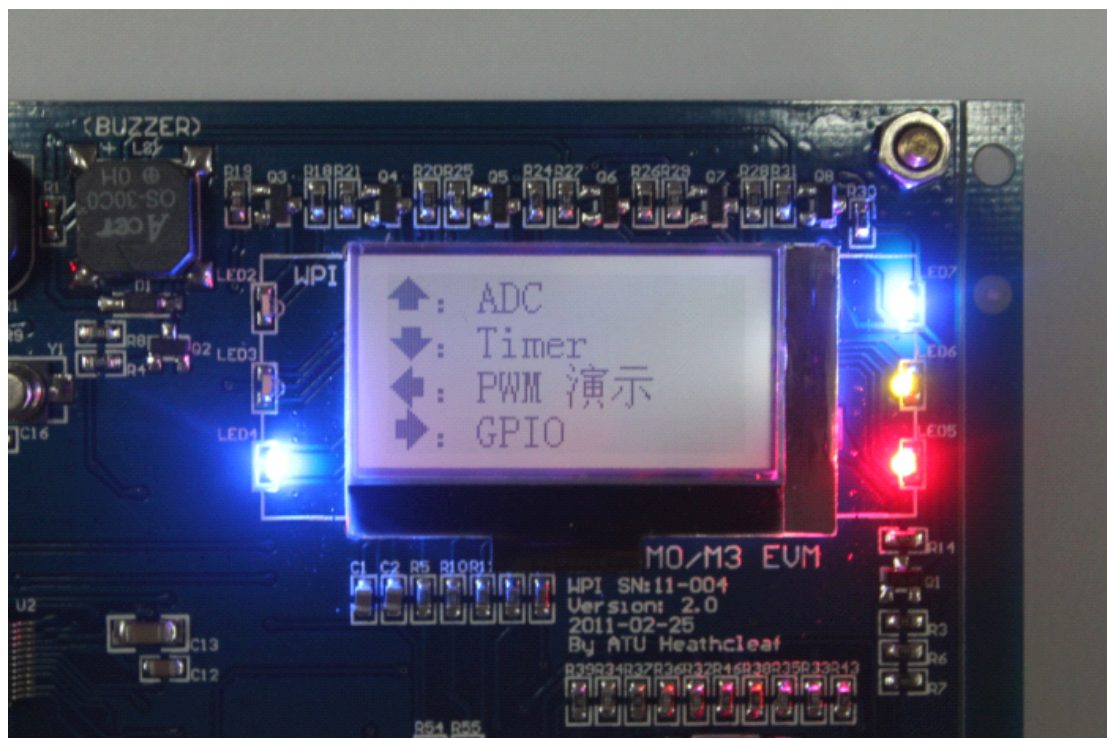






伟博创





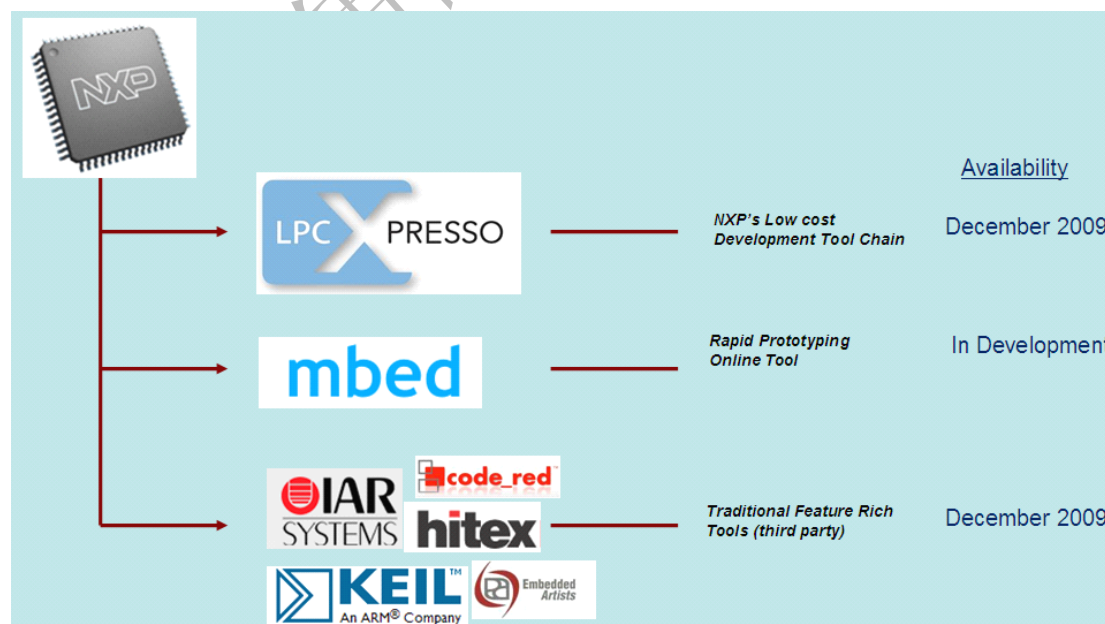
深圳市伟博创科技

Demo Code 的介绍

- 1、“UP” 进入 ADC 演示模式
 通过电位器 VR1 改变 AD 的值并在 LCD 上显示
- 2、“DWON” 进入 Timer 演示模式
 通过 SW1~4 改变 LED 闪烁方式
- 3、“RIGHT” 进入 GPIO 演示模式
 检测 SW1~4 的状态并在 LCD 上显示
- 4、“LEFT” 进入 PWM 演示模式
 通过电位器改变蜂鸣器的发声频率
- 5、“Enter” 返回主菜单界面

安装 Keil 及 M0 EVM 的驱动

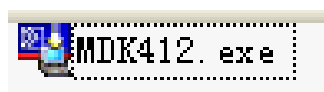
LPC1100 Tool Solutions



Preparations

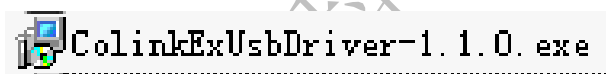
1, Install Keil uVision4

MDK412.exe

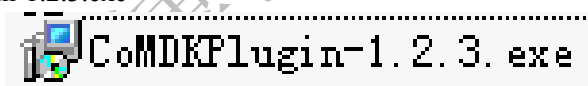


2, Install Colink driver

ColinkExUsbDriver-1.1.0.exe

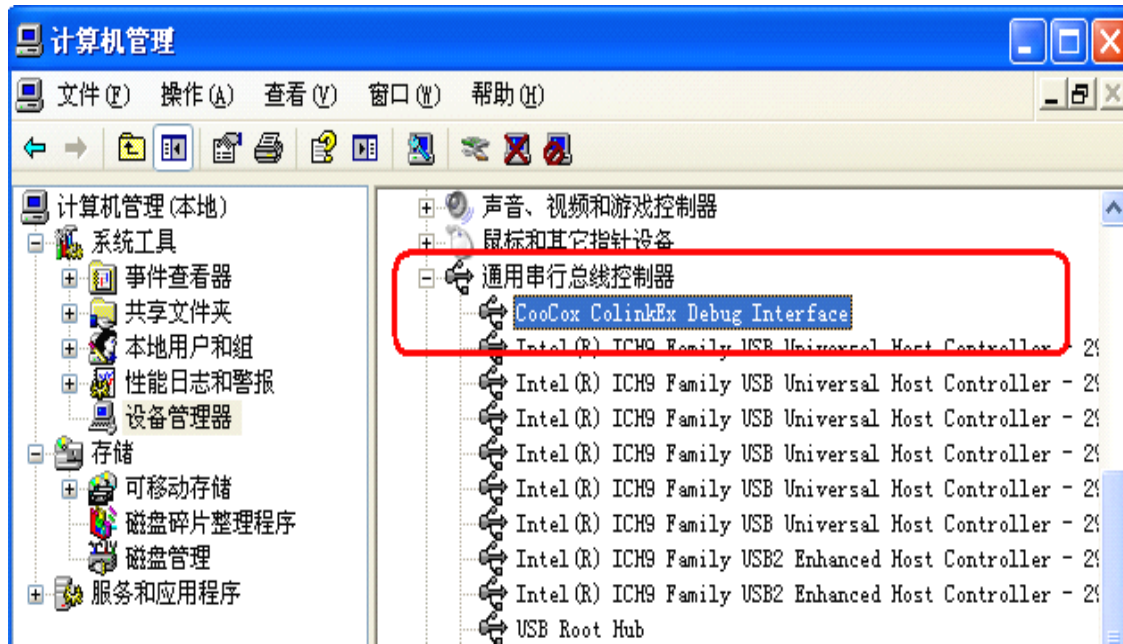


CoMDKPlugin-1.2.3.exe



3, DownCable 和 EVM board 分别连上 USB, 板间用排线连接起来

You can find “CooCox ColinkEx Debug Interface” in DeviceManager



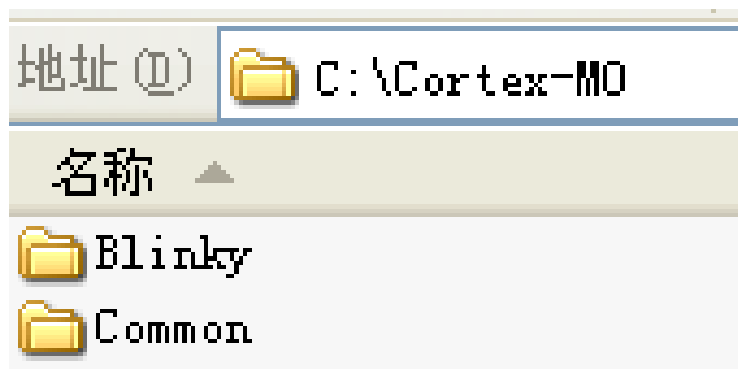
Keil 环境下工程的建立

创建工程

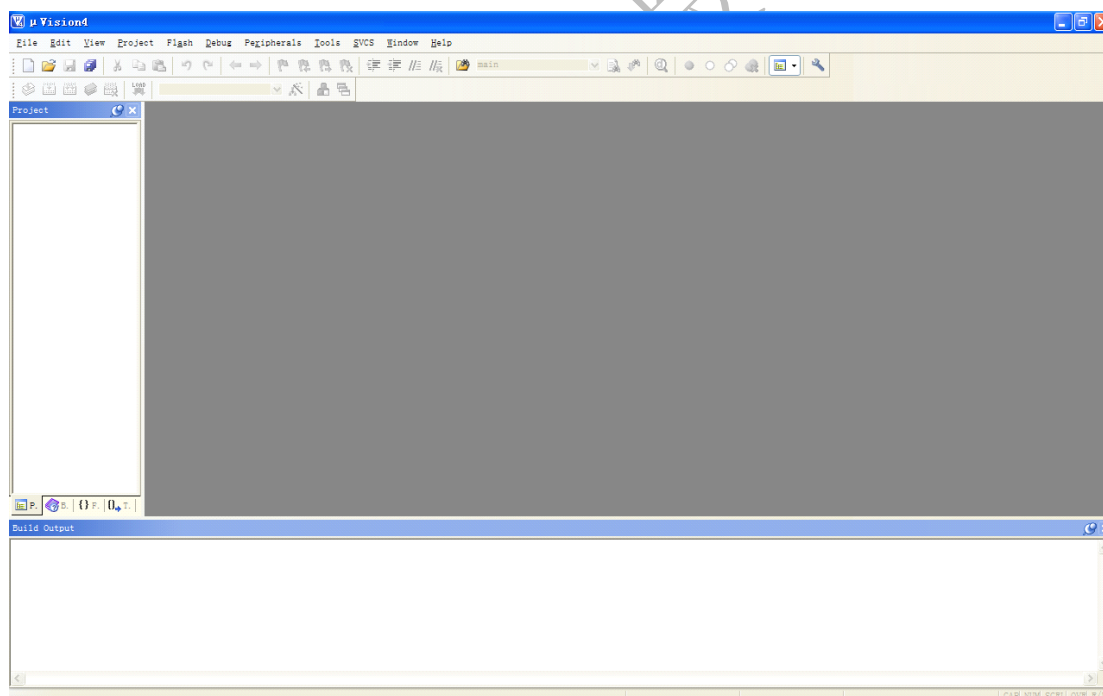
首先，新建一个文件夹用于存放工程文件，此处文件名为“Cortex-M0”，放在 C 盘（盘符可自行选择）；

然后，把“Common”文件夹拷贝到“Cortex-M0”目录下（Common 文件夹在拷给各位的 U 盘文件中的“05 Hands On”文件夹下）

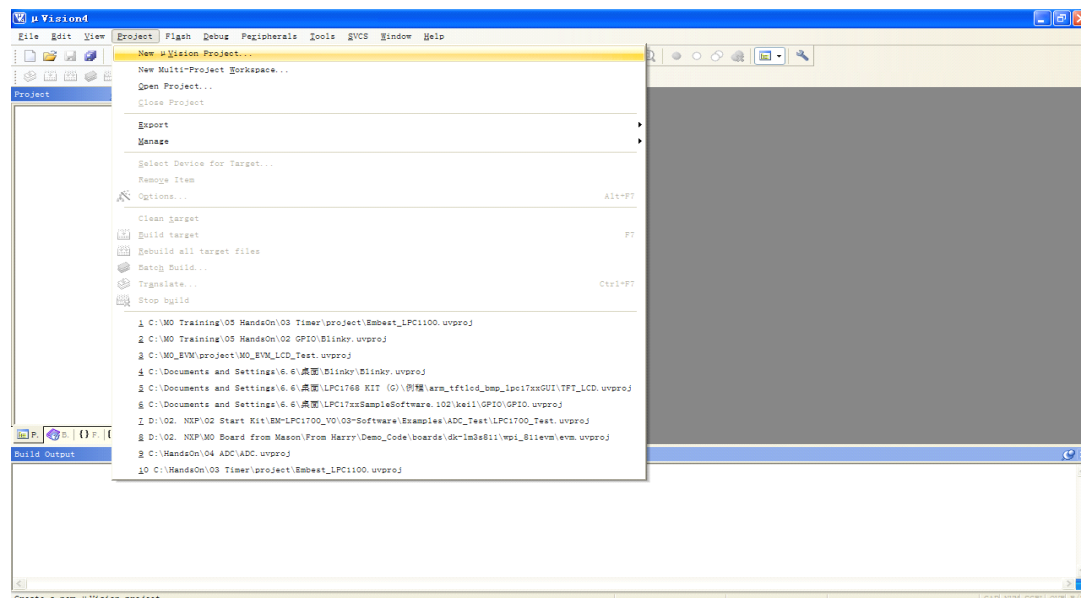
接着，在 Cortex-M0 文件夹下新建名为“Blinky”的文件夹（此文件夹用来存放我们的工程文件，工程路径中建议不要有中文和空格）



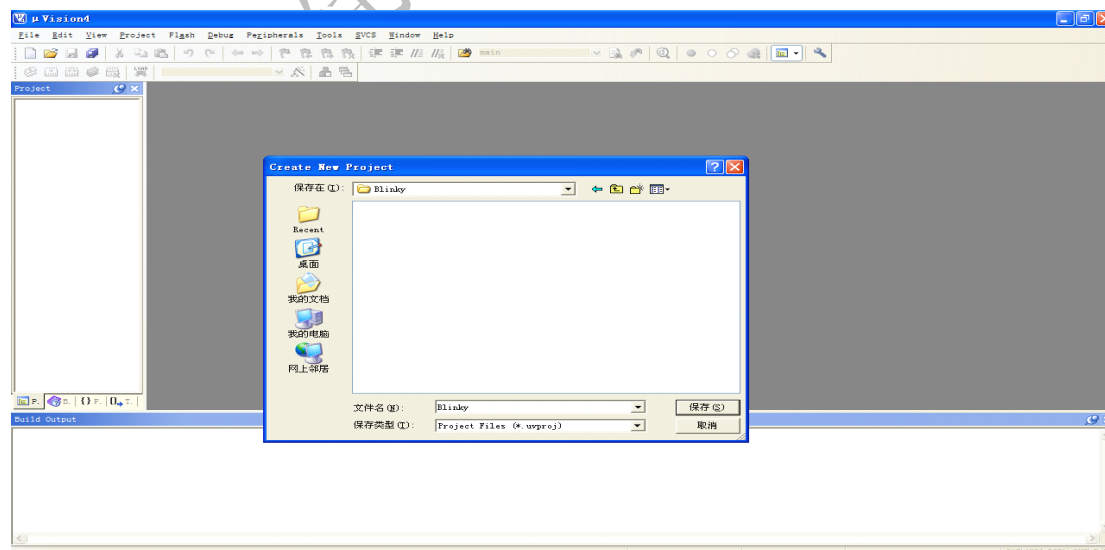
双击桌面图标 Keil uVision4 打开软件环境 Keil uVision4



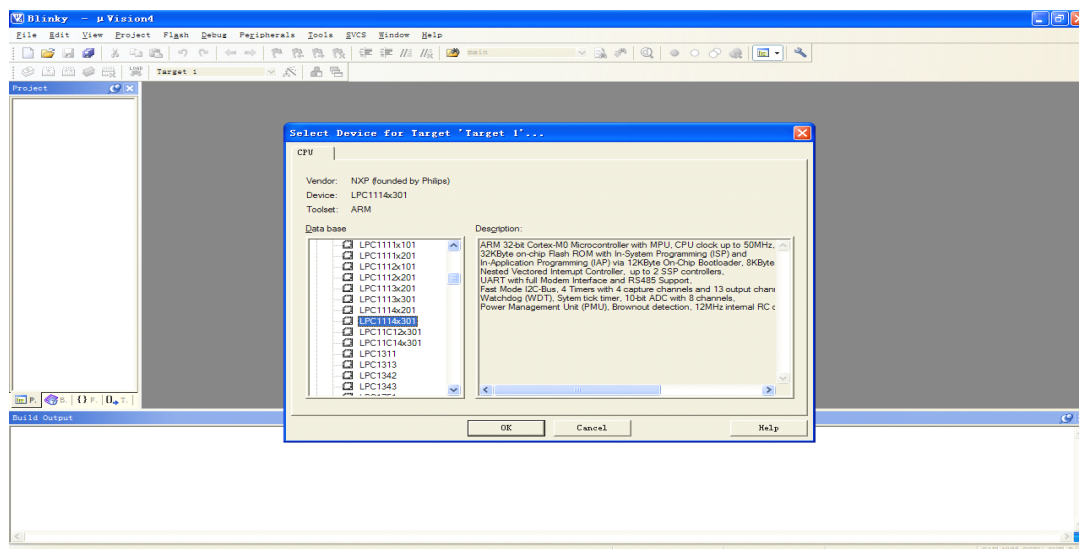
点选 “Project” ——> “New uVision Project”



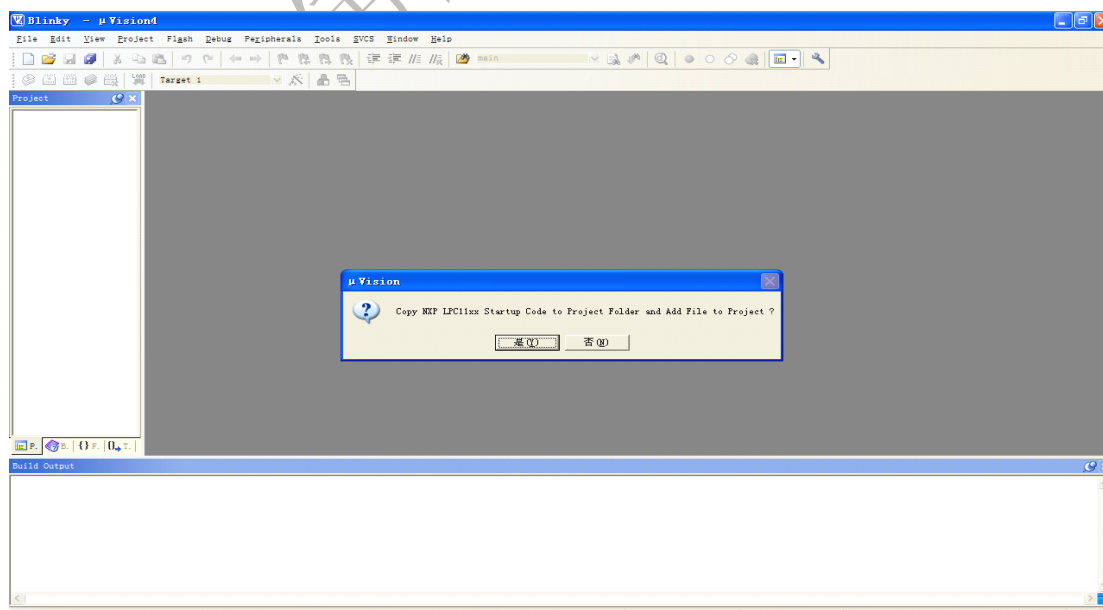
找到保存工程的路径 /Cortex-M0/Blinky
工程名保存为 Blinky， 单击保存



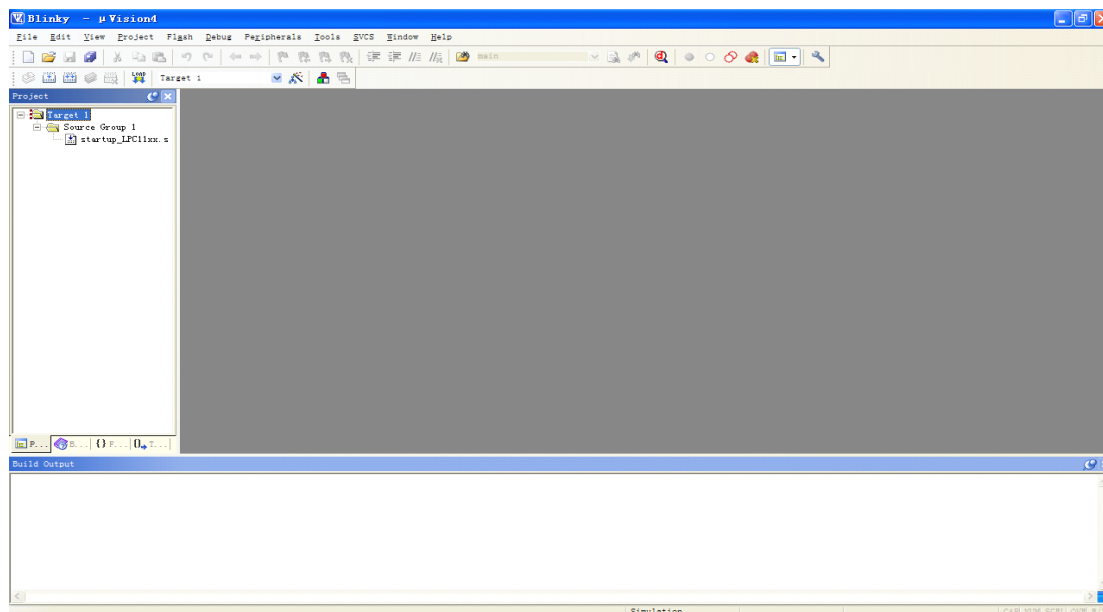
选择 Device, 在 NXP (founded by Philips) 中找到 LPC1114x301, 单击 ok



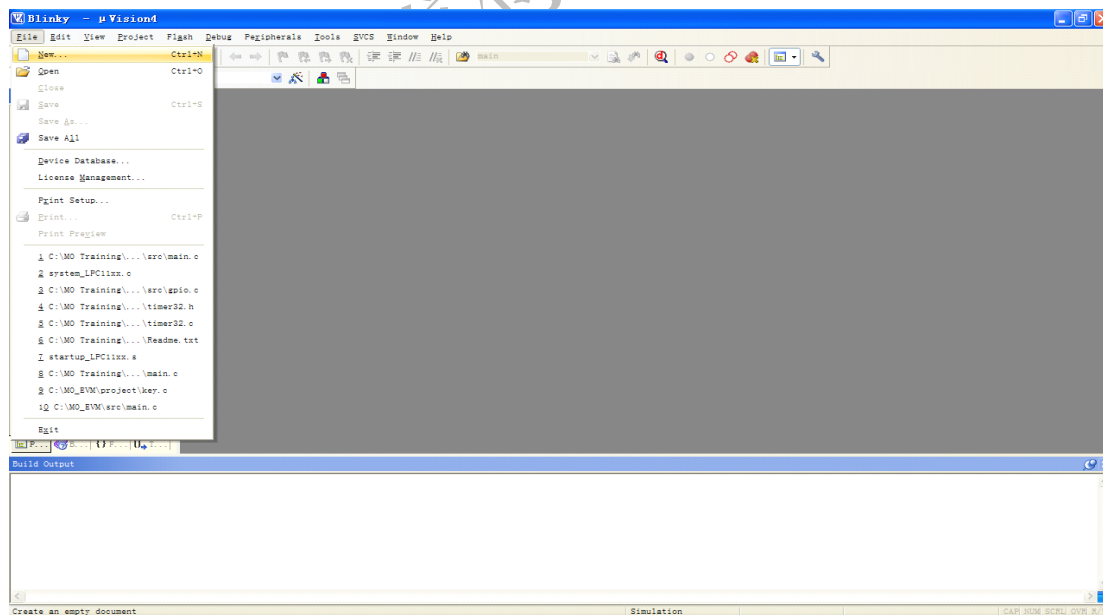
单击 “是” 选择 Keil 自动为工程添加启动代码



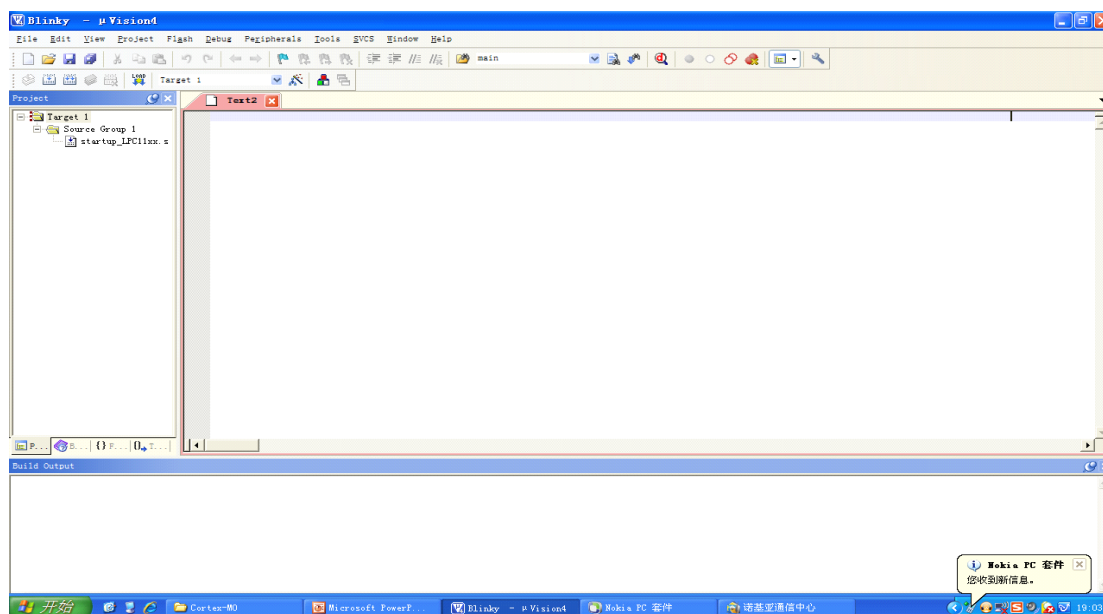
看到初始化的工程界面



Files ——> New



在 Text1 中输入需要编写的代码（复制下页中的代码）



把代码复制到 **Text1** 中

```
#include "LPC11xx.h"
#include "clkconfig.h"
#include "gpio.h"

void delay(uint32_t ms){
    uint32_t i;
    for( i=0; i<ms; i++);
}

int main (void) {

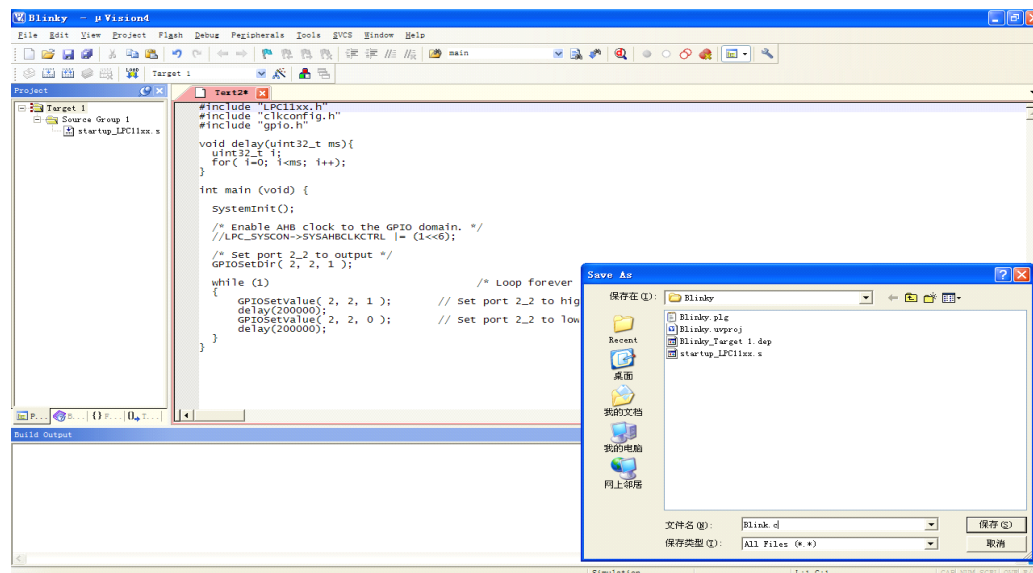
    SystemInit();

    /* Enable AHB clock to the GPIO domain. */
    //LPC_SYSCON->SYSAHBCLKCTRL |= (1<<6);

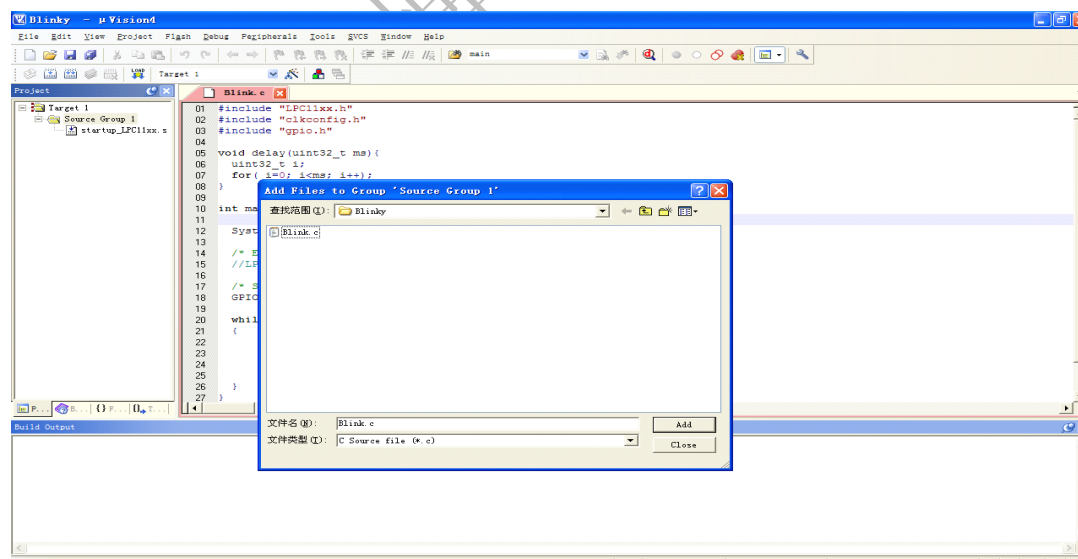
    /* Set port 2_2 to output */
    GPIOSetDir( 2, 2, 1 );

    while (1)                                /* Loop forever */
    {
        GPIOSetValue( 2, 2, 1 );              // Set port 2_2 to high
        delay(200000);
        GPIOSetValue( 2, 2, 0 );              // Set port 2_2 to low
        delay(200000);
    }
}
```


点击 Files ——> Save, 输入文件名 “Blinky.c” 保存为 .C 文件

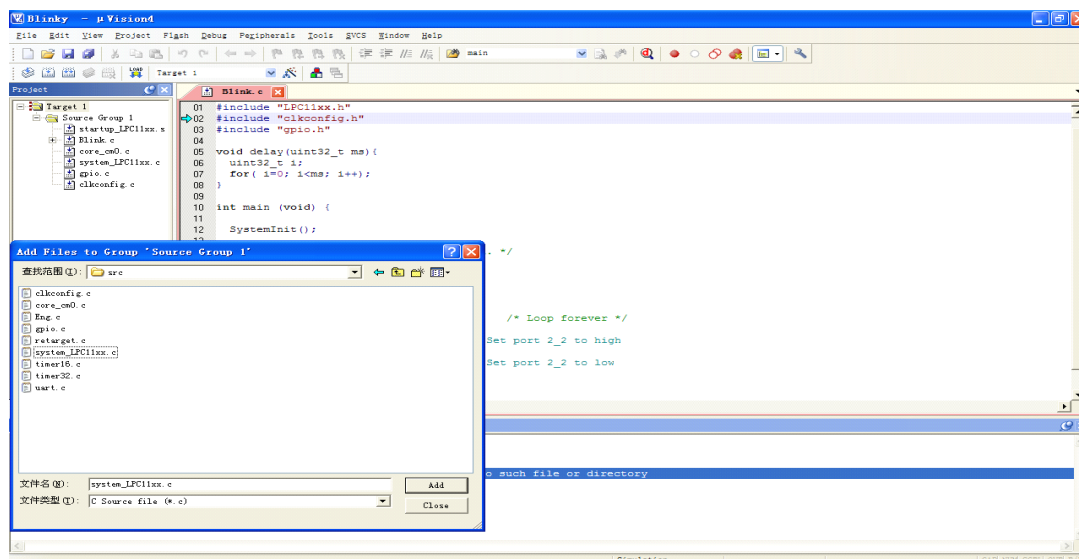


双击 Source Group 1, 添加选中文件 “Blink.C” , 点击 Add 添加到工程中



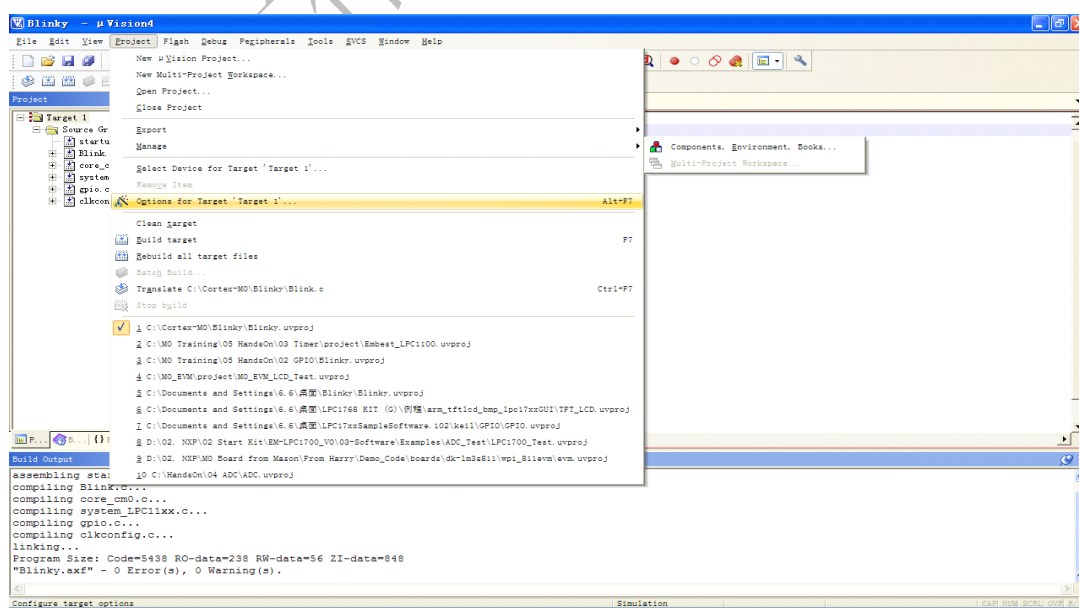
双击 Source Group 1, 继续添加文件到工程中

- 1, clkconfig.c;
- 2, core_cm0.c
- 3, gpio.c
- 4, system_LPC11xx.c

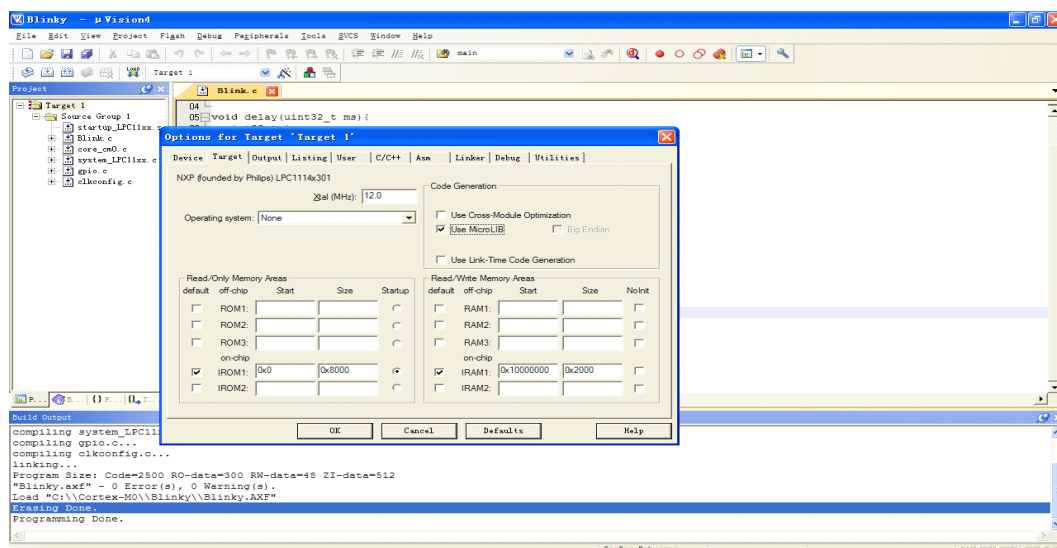


配置工程

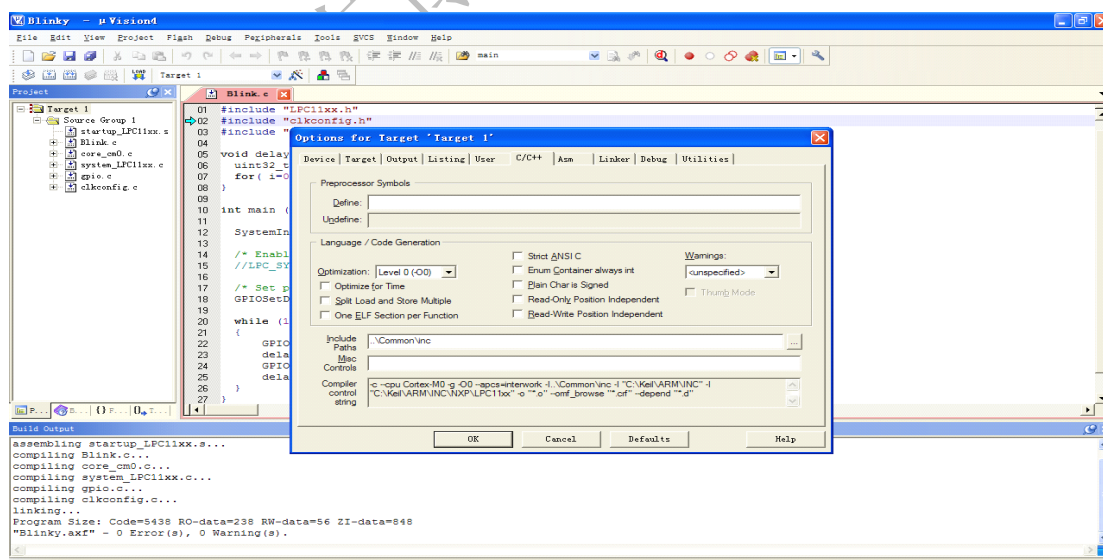
Project à Options for Target "Target 1"



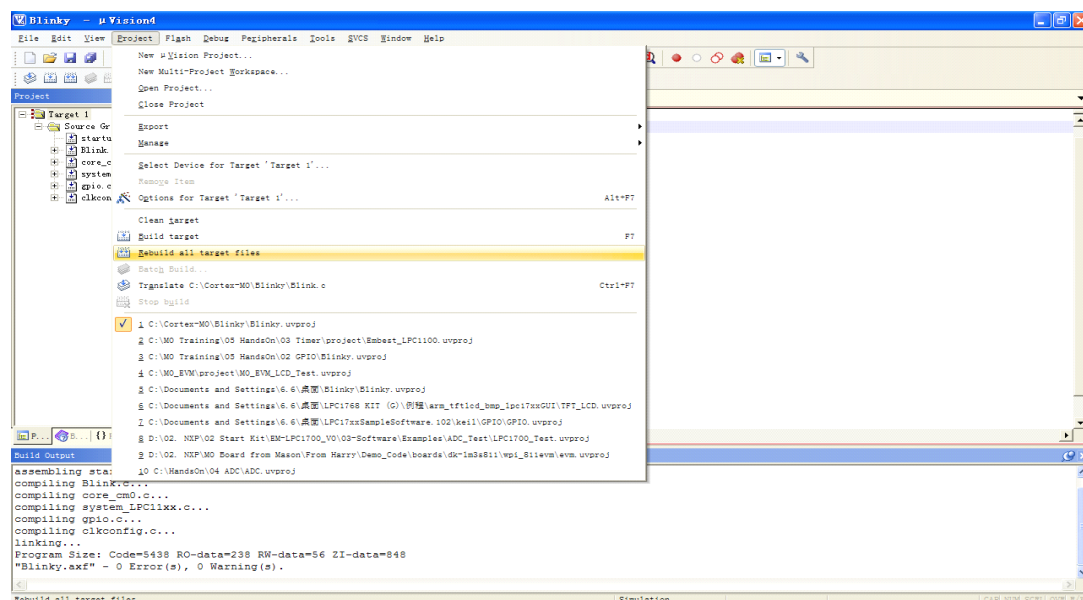
在 Project ——> Options for Target “Target 1” ——> Target ——> 勾选 Use Miro LIB, 点击 ok



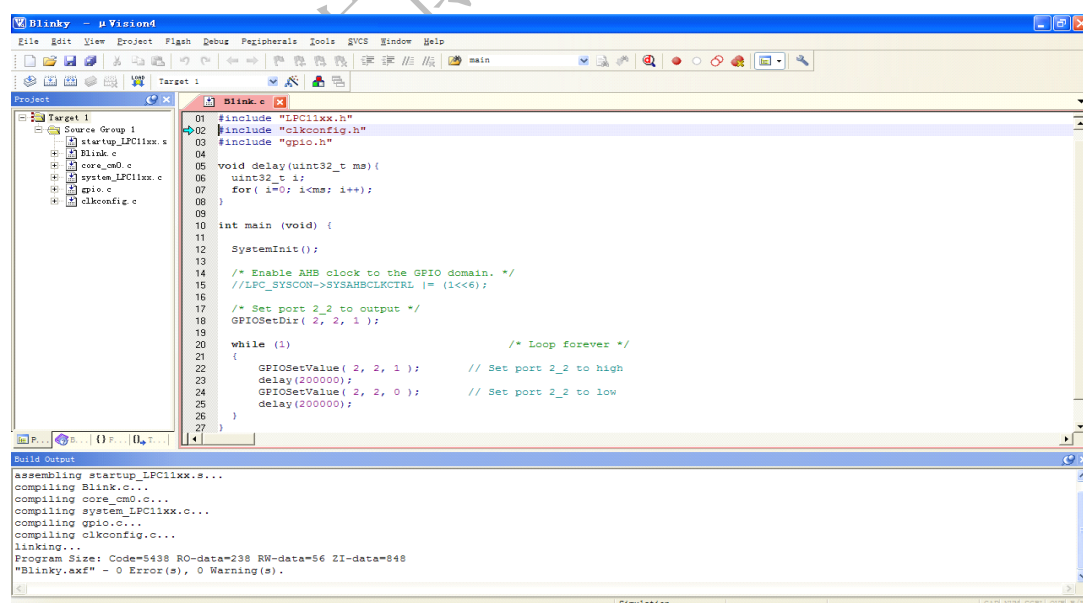
在 Project ——> Options for Target “Target 1” ——> C/C++ ——> include Paths 中输入 ..\Common\inc, 点击 ok



在 Project ——> Rebuild all target files

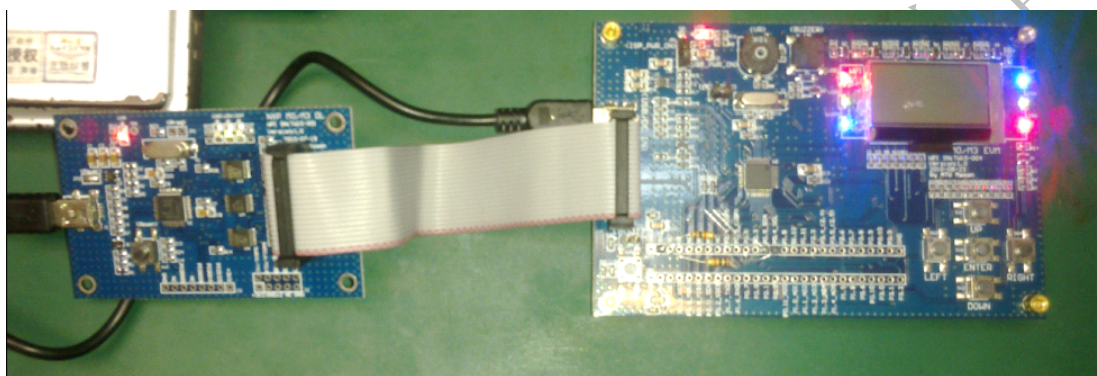


编译工程
看到工程编译完成



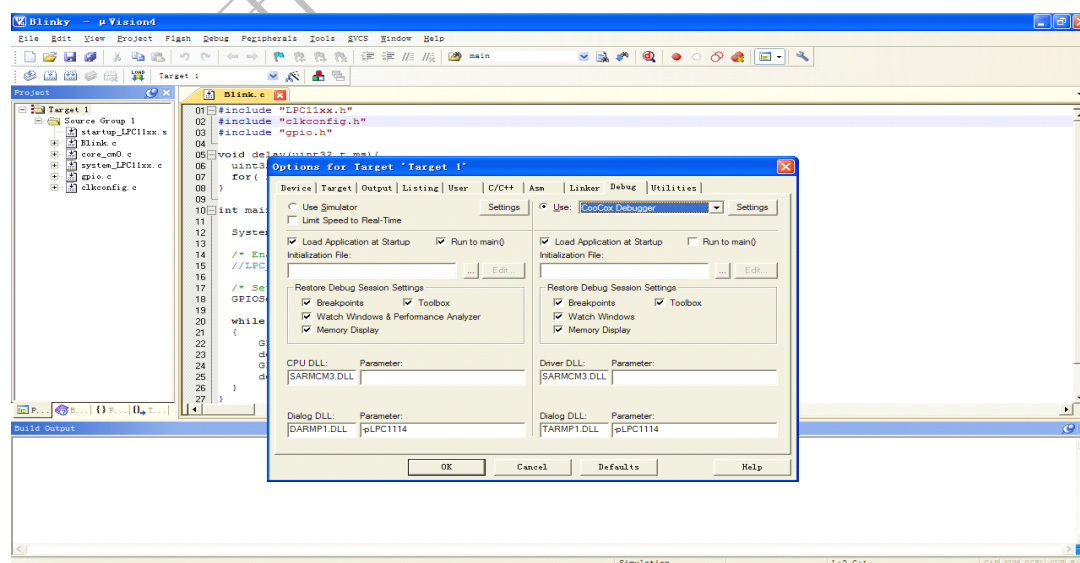
确认硬件连接

- 使用 USB 线给 Download Cable 供电
- 使用 USB 线给 M0 EVM Board 供电
- 使用排线连接 Download Cable & M0 EVM Board

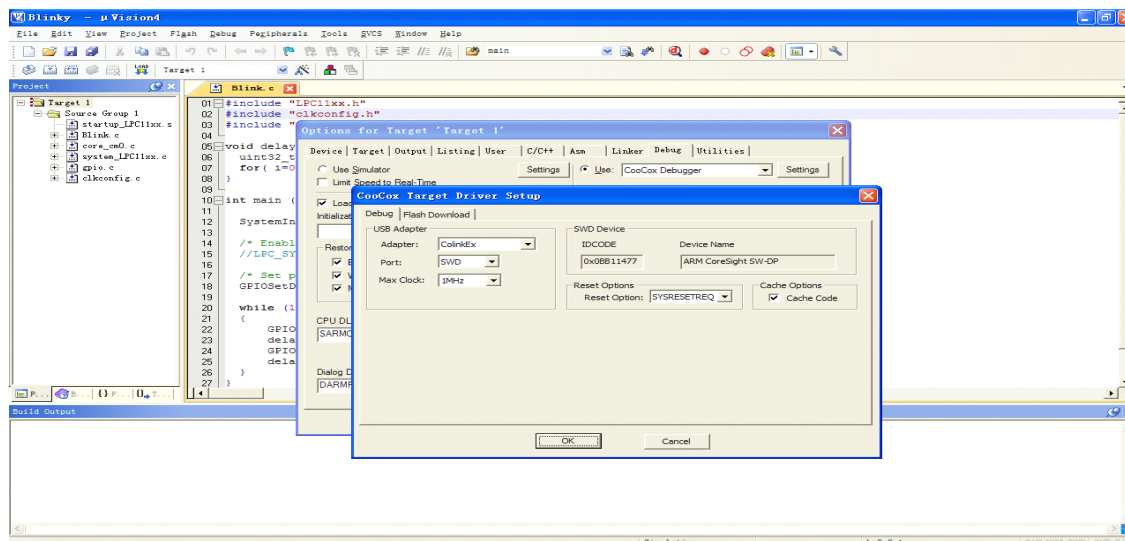


选择仿真器 Coocox Debugger

Project ——> Options for Target “Target 1” ——> Debug ——> Use Coocox Debugger

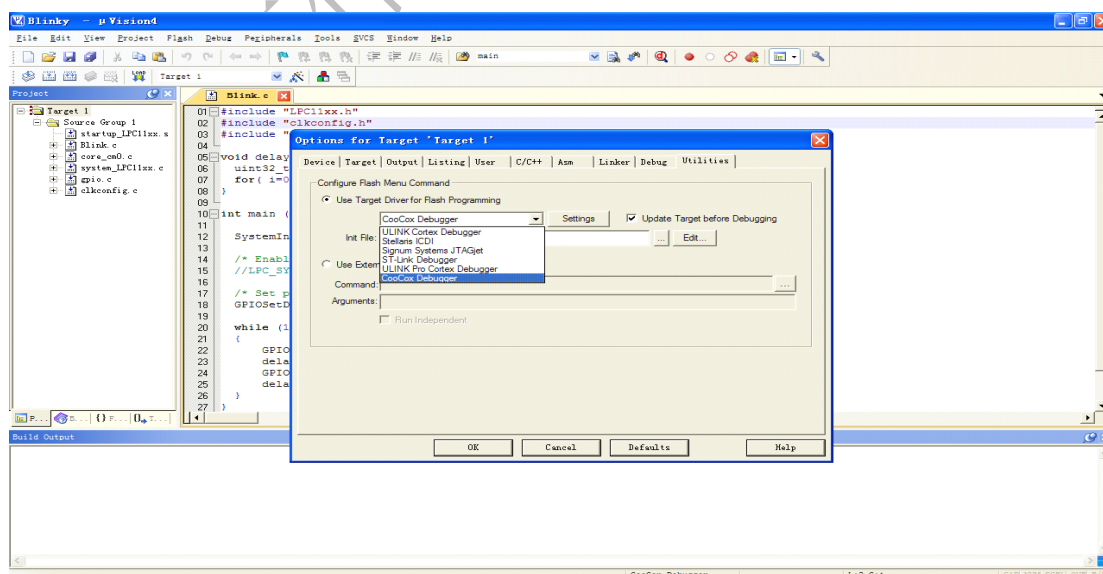


Project ——> Options for Target “Target 1” ——> Debug ——> Use CooCox Debugger ——> 点击 Settings，可识别 IDCODE 0x0BB11477，表示硬件已连接，“Reset Options” 下拉选中 “SYSRESETREQ” 点击 ok

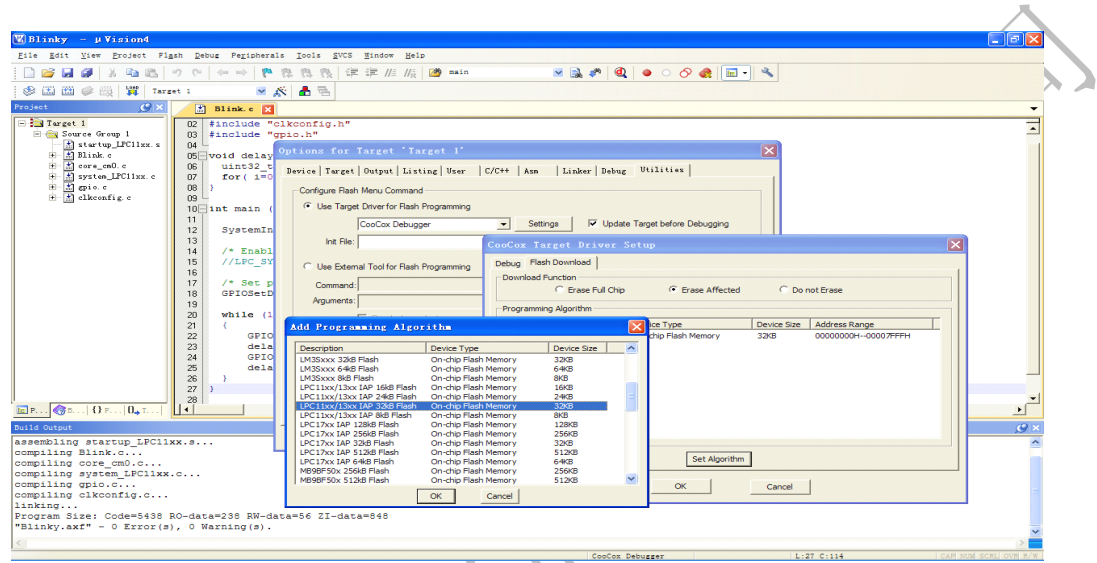


选择 Flash 编程算法

Project ——> Options for Target “Target 1” ——> Utilities ——> Use Target Driver for Flash Programming ——> CooCox Debugger



Project ——> Options for Target “Target 1” ——> Utilities ——> Settings ——> Set Algorithm ——> LPC11xx/13xx IAP 32kB Flash



深圳市伟博创

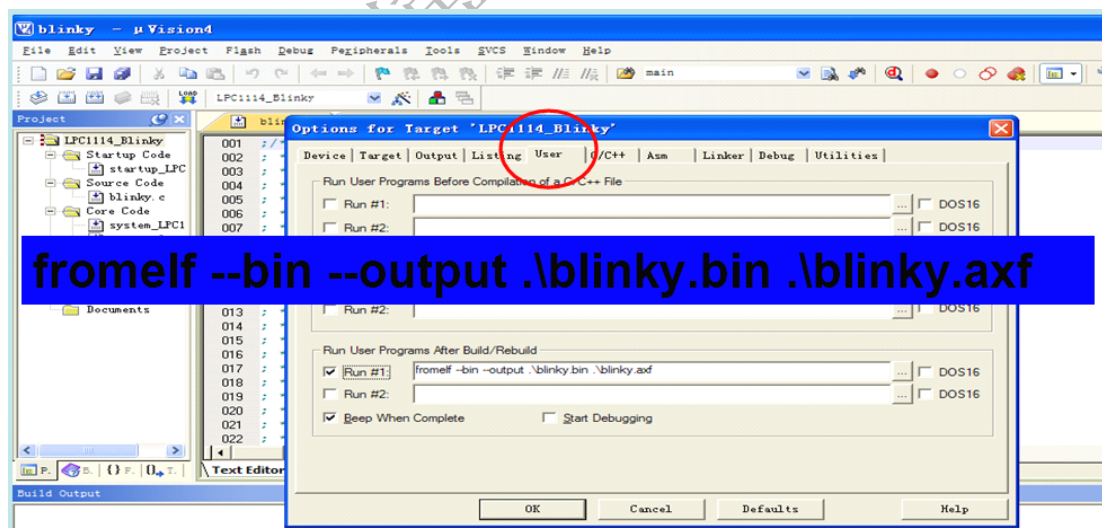
下载，仿真

- 编译程序 Project ——> Rebuild all target files
- 下载程序 Flash ——> à Download
- 仿真 Debug ——> à Start/Stop Debug Session

现象： 红色侧光 LED2 闪烁

生成 bin 文件

“user” -->> “Run User Programs After Build /Rebuild”



常用的 GPIO 库函数

```
void GPIOInit( void );
```

```
void GPIOSetDir( uint32_t portNum, uint32_t bitPosi, uint32_t dir );
```

```
void GPIOSetValue( uint32_t portNum, uint32_t bitPosi, uint32_t bitVal );
```

```
void LEDToggle( uint32_t portNum, uint32_t bitPosi);
```

GPIO 程序示例

Pin: PIO2_2

```
int main (void) {  
  
    SystemInit();  
  
    /* Enable AHB clock to the GPIO domain. */  
    LPC_SYSCON->SYSAHBCLKCTRL |= (1<<6);  
  
    /* Set port 2_2 to output */  
    GPIOSetDir( 2, 2, 1 );  
  
    while (1) /* Loop forever */  
    {  
        GPIOSetValue( 2, 2, 1 ); /* Set port 2_2 to high  
        delay(1000);  
        GPIOSetValue( 2, 2, 0 ); /* Set port 2_2 to low  
        delay(1000);  
    }  
}
```

修改程序点亮 **LED7** (**PIO1_11**)

常用的 **Timer** 库函数

```
void delay32Ms(uint8_t timer_num, uint32_t delayInMs);
```

```
void TIMER32_0_IRQHandler(void);
```

```
void TIMER32_1_IRQHandler(void);
```

```
void enable_timer32(uint8_t timer_num);
```

```
void disable_timer32(uint8_t timer_num);
```

```
void reset_timer32(uint8_t timer_num);
```

```
void init_timer32(uint8_t timer_num, uint32_t timerInterval);
```

Timer 程序示例

```
int main (void) {
    SystemInit();

    init_timer32(1, TIME_INTERVAL);
    enable_timer32(1);

    LPC_SYSCON->SYSAHBCLKCTRL |= (1<<6);    /* Enable AHB clock to the GPIO domain. */
    GPIOSetDir( 2, 2, 1 );                    /* Set port 2_0 to output */

    while (1)                                 /* Loop forever */
    {
        /* I/O configuration and LED setting pending. */
        if ( (timer32_1_counter > 0) && (timer32_1_counter <= 50) )
        {
            GPIOSetValue( 2, 2, 0 );
        }
        if ( (timer32_1_counter > 50) && (timer32_1_counter <= 100) )
        {
            GPIOSetValue( 2, 2, 1 );
        }
        else if ( timer32_1_counter > 100 )
        {
            timer32_1_counter = 0;
        }
    }
}

void TIMER32_1_IRQHandler(void)
{
    LPC_TMR32B1->IR = 1;                      /* clear interrupt flag */
    timer32_1_counter++;
    return;
}
```

```
void init_timer32(uint8_t timer_num, uint32_t TimerInterval)
{
    if ( timer_num == 0 )
    {
        /* Some of the I/O pins need to be clearly planned if
        you use below module because JTAG and TIMER CAP/MAT pins are muxed. */
        LPC_SYSCON->SYSAHBCLKCTRL |= (1<<9);
        /* Timer0_32 I/O config */
        /* Timer0_32 CAP0 */
        LPC_IOCON->PIO1_5 &= ~0x07;
        LPC_IOCON->PIO1_5 |= 0x02;
        /* Timer0_32 MAT0 */
        LPC_IOCON->PIO1_6 &= ~0x07;
        LPC_IOCON->PIO1_6 |= 0x02;
        /* Timer0_32 MAT1 */
        LPC_IOCON->PIO1_7 &= ~0x07;
        LPC_IOCON->PIO1_7 |= 0x02;
        /* Timer0_32 MAT2 */
        LPC_IOCON->PIO0_1 &= ~0x07;
        LPC_IOCON->PIO0_1 |= 0x02;
#ifdef __JTAG_DISABLED
        /* Timer0_32 MAT3 */
        LPC_IOCON->JTAG_TDI_PIO0_11 &= ~0x07;
        LPC_IOCON->JTAG_TDI_PIO0_11 |= 0x03;
#endif
        timer32_0_counter = 0;
        LPC_TMR32B0->MR0 = TimerInterval;
        // LPC_TMR32B0->EMR &= ~(0xFF<<4);
        // LPC_TMR32B0->EMR |= ((0x3<<4)|(0x3<<6)|(0x3<<8)|(0x3<<10)); /* MR0/1/2/3 Toggle */
        /* Interrupt and Reset on MR0 */
        LPC_TMR32B0->MCR = 3;
        /* Enable the TIMER0 Interrupt */
        NVIC_EnableIRQ(TIMER_32_0_IRQn);
    }
}
```

深圳市伟博创科技有限公司

使用 **Timer32_1** 控制 **LED7** 闪烁频率

常用的 ADC 库函数

```
void ADC_IRQHandler( void );
```

```
void ADCInit( uint32_t ADC_Clk );
```

```
uint32_t ADCRead( uint8_t channelNum );
```

```
void ADCBurstRead( void );
```

```
int main(void) {
    uint32_t ADC_value;
    unsigned char buf[4];

    SystemInit();
    ini_lcd();
    CLR(1);
    Open_LCD_Light();
    CLR(1);
    ADCInit( ADC_CLK );

    while (1){
        ADCGetValue_Average();
        ADC_value = ADCGetValue_Average();

        buf[0] = ADC_value/1000%10+'0';
        buf[1] = ADC_value/100%10+'0';
        buf[2] = ADC_value/10%10+'0';
        buf[3] = ADC_value%10+'0';

        Show_English(" ADC Demo ", 0, 0);
        Show_English(" Voltage is ", 0, 4);
        Show_English( &buf[0], 3, 6);
        Show_English(" mV ", 7, 6);
    }
}

void ADCInit ( uint32_t ADC_Clk)
{
    LPC_SYSCON->PDRUNCFG &= ~(0x1<<4);
    LPC_SYSCON->SYSAHBCLKCTRL |= (1<<13);

    LPC_IOCON->PIO1_4 &= ~0x8F;
    LPC_IOCON->PIO1_4 |= 0x01;

    LPC_ADC->CR = ( 0x01 << 5 ) | /* SEL=1,select channel 0~7 on ADC0 */
        ( ( SystemAHBFrequency / ADC_Clk - 1 ) << 8 ) | /* CLKDIV = Fpclk / 1000000 - 1 */
        ( 0 << 16 ) | /* BURST = 0, no BURST, software controlled */
        ( 0 << 17 ) | /* CLKS = 0, 11 clocks/10 bits */
        ( 1 << 21 ) | /* PDN = 1, normal operation */
        ( 0 << 22 ) | /* TEST1:0 = 00 */
        ( 0 << 24 ) | /* START = 0 A/D conversion stops */
        ( 0 << 27 ); /* EDGE = 0 (CAP/MAT singal falling,trigger A/D conversion) */
}
```

```
uint32_t ADCGetValue_Average ( void )
{
    uint32_t ADC_Data_5_Average = 0;
    uint32_t ADC_Data_5_Sum = 0;
    uint32_t Sampling_times= 0;
    for (Sampling_times = 0; Sampling_times < 5; Sampling_times++)
    {
        // ADCGetValue();
        LPC_ADC->CR |= (1 << 24) | (1 << 5);
        while (( LPC_ADC->DR[5] & 0x80000000) == 0);
        ADC_Data_5 = LPC_ADC->DR[5];
        ADC_Data_5 = (ADC_Data_5 >> 6) & 0x3fe;
        //ADC_Data_5_Sum += ADC_Data_5;
        //ADC_Data_5_Sum += ADCGetValue();
        ADC_Data_5_Sum += ADC_Data_5;

        //Delay(200);
    }
    ADC_Data_5_Average = ADC_Data_5_Sum / 5;
    ADC_Data_5_Average = (ADC_Data_5_Average * 3300) / 1024;
    //Delay(200);
    return ( ADC_Data_5_Average );
}
```

Flash Magic 工具的使用

