

LM3S811 使用心得—从零开始学 ARM Cortex-M3 内核单片机

申请到211IC的EK - LM3S811评估板，非常开心。接下来几天开始搜集相关的资料文档，回到家开始了LM3S的初步相识。有了开发板，代码调试当然是第一步。
开发环境：

Keil uvision4

从最初学习51单片机时开始，接触的就是Keil，所以还是支持Keil。

下载工具：LM Flash Programmer

Keil 软件中的Load 工具

遇到的问题：1、关于编译器的视频，视频里讲解了 mdk 上每个选项的意思及用法，适合初学 mdk 的同学们

地址：http://v.youku.com/v_show/id_XMjQwMDA5MDIO.html

2、LM3S811 提供的 driver.lib 库，想要弄明白每个函数是不可能的事情，最好的方式是一面看例程的程序流程，一面对应着 Stellaris® Peripheral Driver Library USER'S GUIDE.PDF 文档来看，前期指导函数的作用就可以了。

3、LM3S811 防锁死，翻阅 LM3S 基本例程.PDF，有个底。有什么办法可以防止 LM3S811 锁死？

对于 Sandstorm 家族(即 LM3S100、300、600、800 等)，一旦 JTAG 接口因为误将其无防护地配置为 GPIO 接口而被锁定，则有可能再也无法恢复。为了防患于未然，我们建议在编写每一个应用程序时都必须要在 main()函数的开始处插入一段能够预防 JTAG 失效的代码。

参考：《LM3S 基本例程》

4、如何添加驱动driver.lib库？

浏览StellarisWare \ driverlib \ rvmdk driverlib选择库文件。注意，你需要把文件浏览器更改为寻找库文件类型，所以改变“文件类型的”下拉列表从“C源文件”到“库文件(*.lib)”或“所有文件”。

一、实验一：LED流水灯

EK_LM3S811评估板硬件图的简要说明：

LED：

PD0/PWM0 - - - - - LED3, 蓝
PD1/PWM1 - - - - - LED4, 绿
PB0/PWM2 - - - - - LED5, 红
PB1/PWM3 - - - - - LED6, 蓝
PE0/PWM4 - - - - - LED7, 绿
PE1/PWM5 - - - - - LED8, 红

与JTAG接口复用的GPIO是：PB7 (/TRST)、PC0 (TCK)、PC1 (TMS)、PC2 (TDI)、PC3 (TD0)

```

/*****引用文件*****/
#include "hw_ints.h"
#include "hw_memmap.h"
#include "hw_types.h"
#include "gpio.h"
#include "sysctl.h"
#define uint unsigned int
#define uchar unsigned char
/*****宏定义LED控制引脚*****/
#define LED_PERIPBH SYSCTL_PERIPH_GPIOB //LED外围 B端口作为通用输入/
输出 宏定义
#define LED_PERIPDH SYSCTL_PERIPH_GPIOD //LED外围 D端口作为通用输入/
输出 宏定义
#define LED_PERIPEH SYSCTL_PERIPH_GPIOE //LED外围 E端口作为通用输入/
输出 宏定义
#define BLUE1_L  GPIOPinWrite(GPIO_PORTB_BASE,
GPIO_PIN_0,~GPIO_PIN_0) //BLUE1管脚为低
#define GREEN1_L GPIOPinWrite(GPIO_PORTD_BASE,
GPIO_PIN_1,~GPIO_PIN_1) //GREEN1管脚为低
#define RED1_L  GPIOPinWrite(GPIO_PORTD_BASE,
GPIO_PIN_0,~GPIO_PIN_0) //RED1管脚为低
#define BLUE2_L  GPIOPinWrite(GPIO_PORTB_BASE,
GPIO_PIN_1,~GPIO_PIN_1) //BLUE2管脚为低
#define GREEN2_L GPIOPinWrite(GPIO_PORTE_BASE,
GPIO_PIN_0,~GPIO_PIN_0) //GREEN2管脚为低
#define RED2_L  GPIOPinWrite(GPIO_PORTE_BASE,
GPIO_PIN_1,~GPIO_PIN_1) //RED2管脚为低
#define BLUE1_H  GPIOPinWrite(GPIO_PORTB_BASE,
GPIO_PIN_0,GPIO_PIN_0) //BLUE1管脚为高
#define GREEN1_H GPIOPinWrite(GPIO_PORTD_BASE,
GPIO_PIN_1,GPIO_PIN_1) //GREEN1管脚为高
#define RED1_H  GPIOPinWrite(GPIO_PORTD_BASE,
GPIO_PIN_0,GPIO_PIN_0) //RED1管脚为高
#define BLUE2_H  GPIOPinWrite(GPIO_PORTB_BASE,
GPIO_PIN_1,GPIO_PIN_1) //BLUE2管脚为高
#define GREEN2_H GPIOPinWrite(GPIO_PORTE_BASE,
GPIO_PIN_0,GPIO_PIN_0) //GREEN2管脚为高
#define RED2_H  GPIOPinWrite(GPIO_PORTE_BASE,
GPIO_PIN_1,GPIO_PIN_1) //RED2管脚为高
/***** 如无外部按键设计，可省略*****/
// 定义KEY
#define KEY_PERIPH SYSCTL_PERIPH_GPIOC
#define KEY_PORT GPIO_PORTC_BASE
#define KEY_PIN GPIO_PIN_4

```

```

// 防止JTAG失效
void jtagWait(void)
{
SysCtlPeripheralEnable(KEY_PERIPH);           // 使能KEY所在的GPIO端口
GPIOPinTypeGPIOInput(KEY_PORT, KEY_PIN);     // 设置KEY所在管脚为输入
if (GPIOPinRead(KEY_PORT, KEY_PIN) == 0x00) // 若复位时按下KEY, 则进入
    {
        while(1);                             // 死循环, 以等待JTAG连接
    }
GPIOPinIntDisable(KEY_PERIPH,KEY_PIN);       // 禁止KEY所在的GPIO端口
}
//延迟函数
void delay (uint a)
{
uint i,j;
for(i=256;i>0;i--)
for(j=256;j>0;j--)
while(a>0)
a--;
}

int main(void)
{
jtagWait();           // 防止JTAG失效
SysCtlClockSet(SYSCTL_SYSDIV_1 | SYSCTL_USE_OSC | SYSCTL_OSC_MAIN
|SYSCTL_XTAL_6MHZ);
//SysCtlClockSet函数 () 系统时钟设置 采用主振荡器 外接
6MHz晶振 不分频
//使能LED所在引脚
SysCtlPeripheralEnable(LED_PERIPBH);
SysCtlPeripheralEnable(LED_PERIPDH);
SysCtlPeripheralEnable(LED_PERIPEH);
//GPIO输出型的
GPIOPinTypeGPIOOutput(GPIO_PORTB_BASE, GPIO_PIN_1);
GPIOPinTypeGPIOOutput(GPIO_PORTE_BASE, GPIO_PIN_0);
GPIOPinTypeGPIOOutput(GPIO_PORTE_BASE, GPIO_PIN_1);
GPIOPinTypeGPIOOutput(GPIO_PORTD_BASE, GPIO_PIN_0);
GPIOPinTypeGPIOOutput(GPIO_PORTD_BASE, GPIO_PIN_1);
GPIOPinTypeGPIOOutput(GPIO_PORTB_BASE, GPIO_PIN_0);
//初始化过程拉低全部LED控制脚
BLUE1_L;
BLUE2_L;
RED1_L;
RED2_L;

```

```
GREEN1_L;
GREEN2_L;
while(1)
{
    delay(500000);
    BLUE1_H;
    BLUE2_H;
    RED1_L;
    RED2_L;
    GREEN1_L;
    GREEN2_L;
    delay(500000);
    BLUE1_L;
    BLUE2_L;
    RED1_H;
    RED2_H;
    GREEN1_L;
    GREEN2_L;
    delay(500000);
    BLUE1_L;
    BLUE2_L;
    RED1_L;
    RED2_L;
    GREEN1_H;
    GREEN2_H;
}
}
```