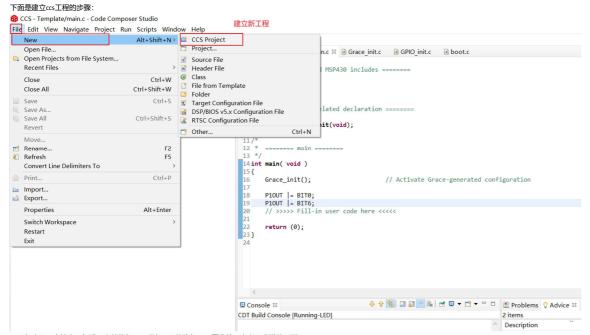


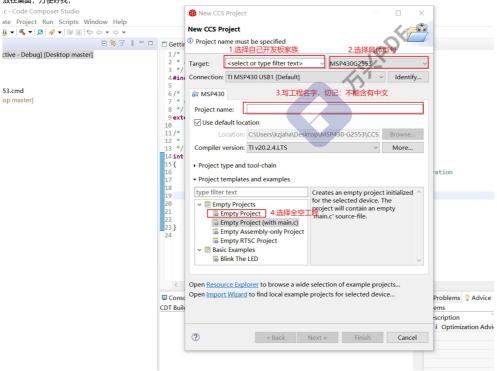
MSP430点灯和简单流水灯

2021年7月8日 11:06

先打开cs的时候会选择自己的workshop,这时也选桌面上自己建立的文件夹;先建立css的工程,注意路径不要乱放,在桌面建立一个文件夹MSP430G2553,里面分别建立CCS和Grace,然后以后的工程都放在这文件夹中,然后配置的过程不要出现任何失误;

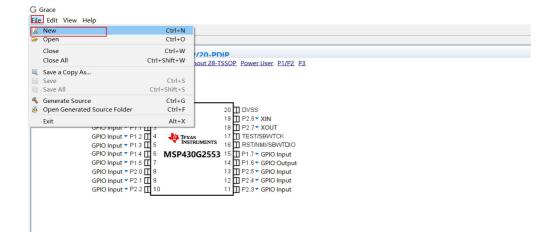


下面忽略了一个地方,切记:文件路径(工程名下面的路径)不要乱放,向上面我说的那样放在桌面。方便好找:



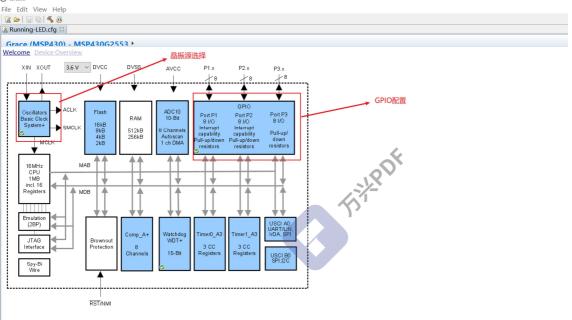
ccs工程建好了,下面开始配置Grace,打开grace,按照下面步骤走;





选择device overview,出现下面的界面,按照图上的信息配置;

G Grace



按照自己开发板的引脚去选择,我用的是20个引脚的;

File Edit View Help

R = | R | 4 @ 🙎 Running-LED.cfg 🖾

Grace (MSP430) > GPIO - Overview
Overview Pinout 32-QFN Pinout 20-TSSOP/20-PDIP Pinout 28-TSSOP Power User P1/P2 P3

MSP430 devices have up to eight digital I/O ports implemented, P1 to Px. Each port has eight I/O pins. Every I/O pin is individually configurable for input or output direction, and each I/O line can be indinterrupt capability. More interrupt capable pins are available on several devices. Each interrupt for the P1 and P2 I/O lines can be individually enabled and configured to provide an interrupt on a rising enable. source a single interrupt vector, and all P2 I/O lines source a different, single interrupt vector.

Note that in Grace the pin configuration by directly manipulating the GPIO module should be done last since Grace-managed peripheral modules will perform the needed pin setup automatically.

▼ Use Case: Port Pins Configured in Output Direction

With the pins configured as I/O function and output direction, individual bits in each PXOUT register represent the output value on the corresponding I/O pin. Internal pull-up/down resistor is disabled in Bit = 1 the output is set high.

Grace Configuration:

- 1. In the GPIO module select Pinout View of applicable device package
- 2. From the drop-down box of P1.5 and P1.7, select GPIO Output option
- 3. To select GPIO Output high or low state, navigate to ${\bf Power\ User\ }$ view to configure

User Code:

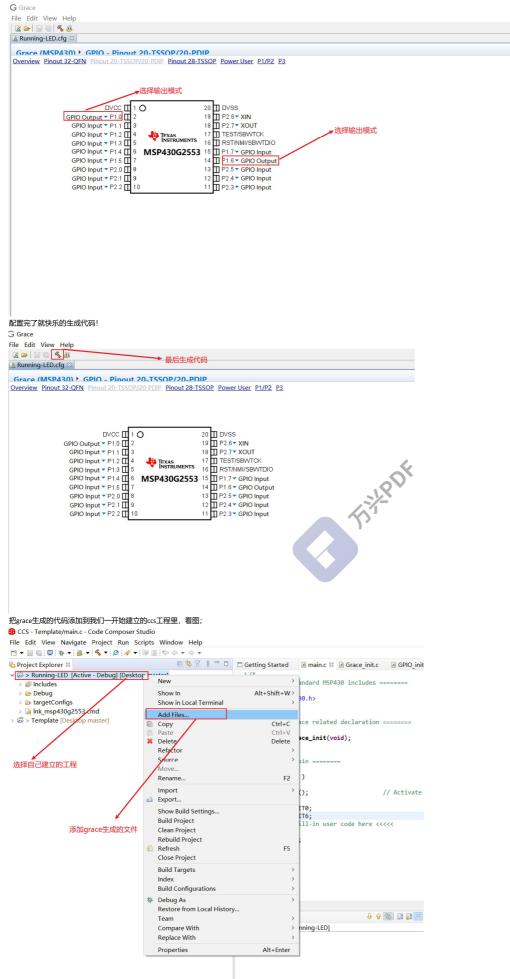
// Drive P1.5, P1.7 output high P10UT |= (BIT5 + BIT7);

// Drive P1.5, P1.7 output low P10UT &= ~(BIT5 + BIT7); ▶ Use Case: Port Pins Configured in Input Direction

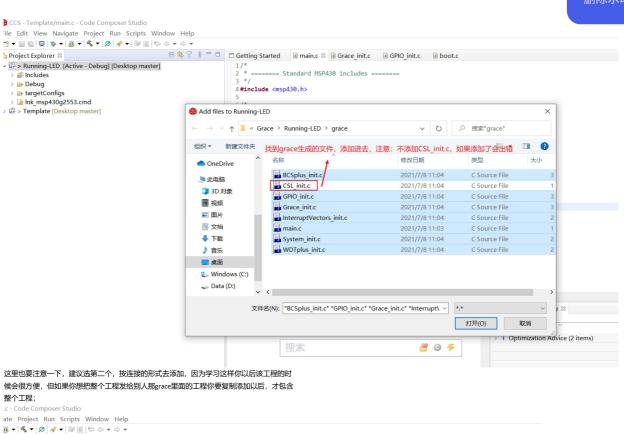
> Use Case: Event Capture on Port Input Pin using Polling

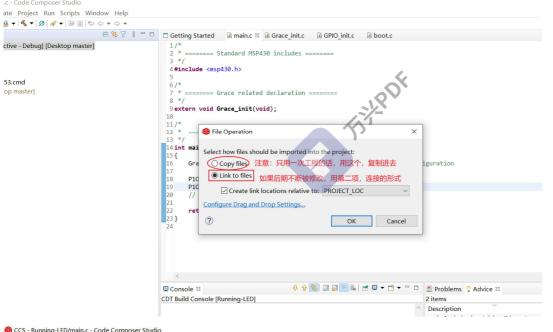
> Use Case: Event Capture on Port Input Pin using Interrupts

配置GPIO,查看开发板原理图发现,p1.0和p1.6引脚连着LED正极,所以配成输出模式;



这里注意一下:选择文件的时候,切记不要选择CSL_init.c文件,因为添加了就会报错;





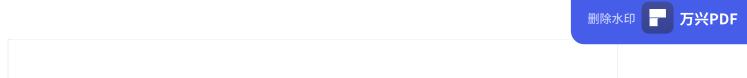
```
CCS - Running-LED/main.c - Code Composer Studio
File Edit View Navigate Project Run Scripts Window Help
□ 🕏 🔻 🖟 🗆 🗆 Getting Started 🗓 main.c 🚨 Grace_init.c 🚨 GPIO_init.c 🚨 boot.c 🗋 main.c 🕮
Project Explorer ⋈
                                                                1/*
2 * ====== Standard MSP430 includes =======

√ ¼ > Running-LED [Active - Debug] [Desktop master]

  > 🔊 Includes
   > 🗁 Debug
                                                                 4#include <msp430.h>
   > 📴 targetConfigs
   > 🖟 BCSplus_init.c
                                                                6/*
7 * ======= Grace related declaration =======
    Grace init.c
    9 extern void Grace_init(void);
                           发现我们建立的ccs工程下出现了
                                                               9 extern void Grace
10
11/*
12 * ---- main
13 */
14 int main( void )
                           grace生成的文件,说明文件已经添加进去了!
    lnk msp430q2553.cmd
   main.c

Rystem_init.c

    ₩DTplus_init.c
    > Template [Desktop m
                                                                    Grace_init();
                                                                                                      // Activate Grace-generated configuration
                                                                     // >>>> Fill-in user code here <<<<
                                                                     return (0);
```

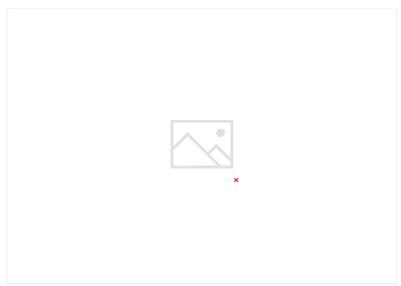








Filed



这就是入门MSP430的第一课,距电赛还有20多天的时间,希望自己能快点熟练msp430,加油!

