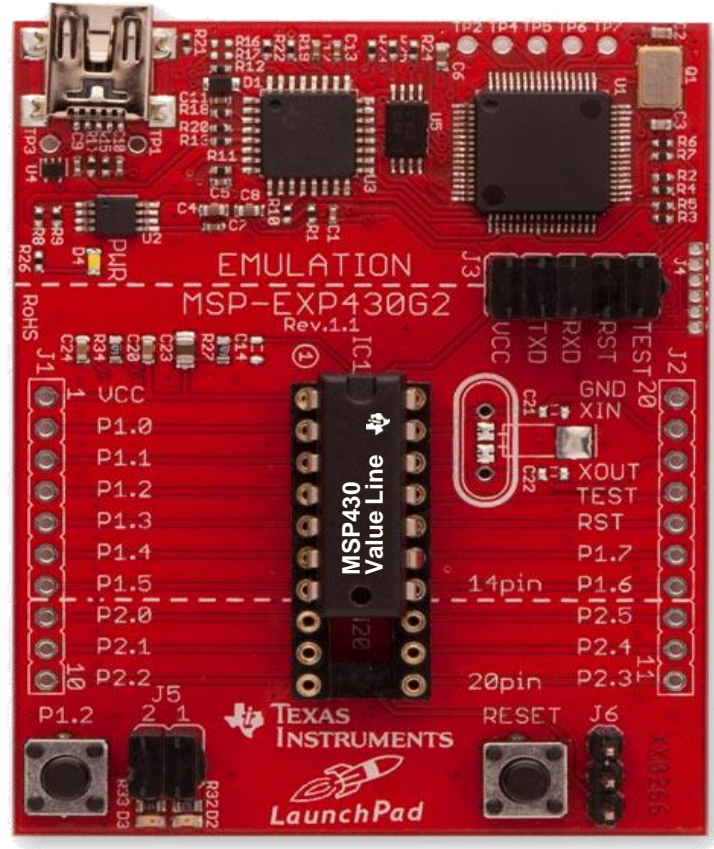
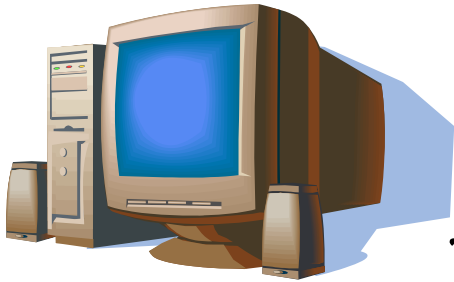


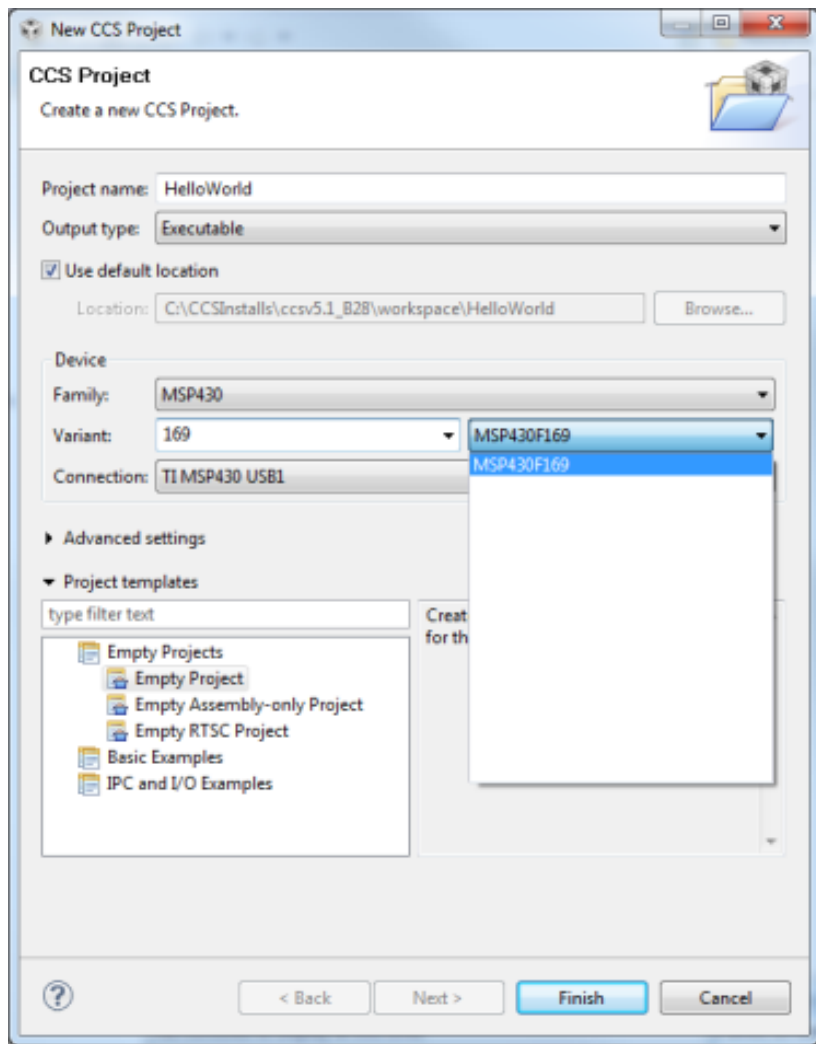
Lab_0: 闪动LED

- 使用在MSP430F5529上运行的例程项目，实现LED闪动



LED1 闪烁!

Project 创建向导



- 一步完成工程的创建，满足大多数情况的需求
 - 当NEXT按钮可用时，提醒需要进一步的选择
- 包括Debugger的设置
 - 在选择了芯片后，可以选择所用的连接方式，环境会自动生成一个 ccxml 文件，在后续可以在该文件中对配置进行修改
- 使用默认配置
 - 大多数情况下，使用默认配置可以满足要求。其余，如Compiler version, endianness... 在 advanced 中可进行配置

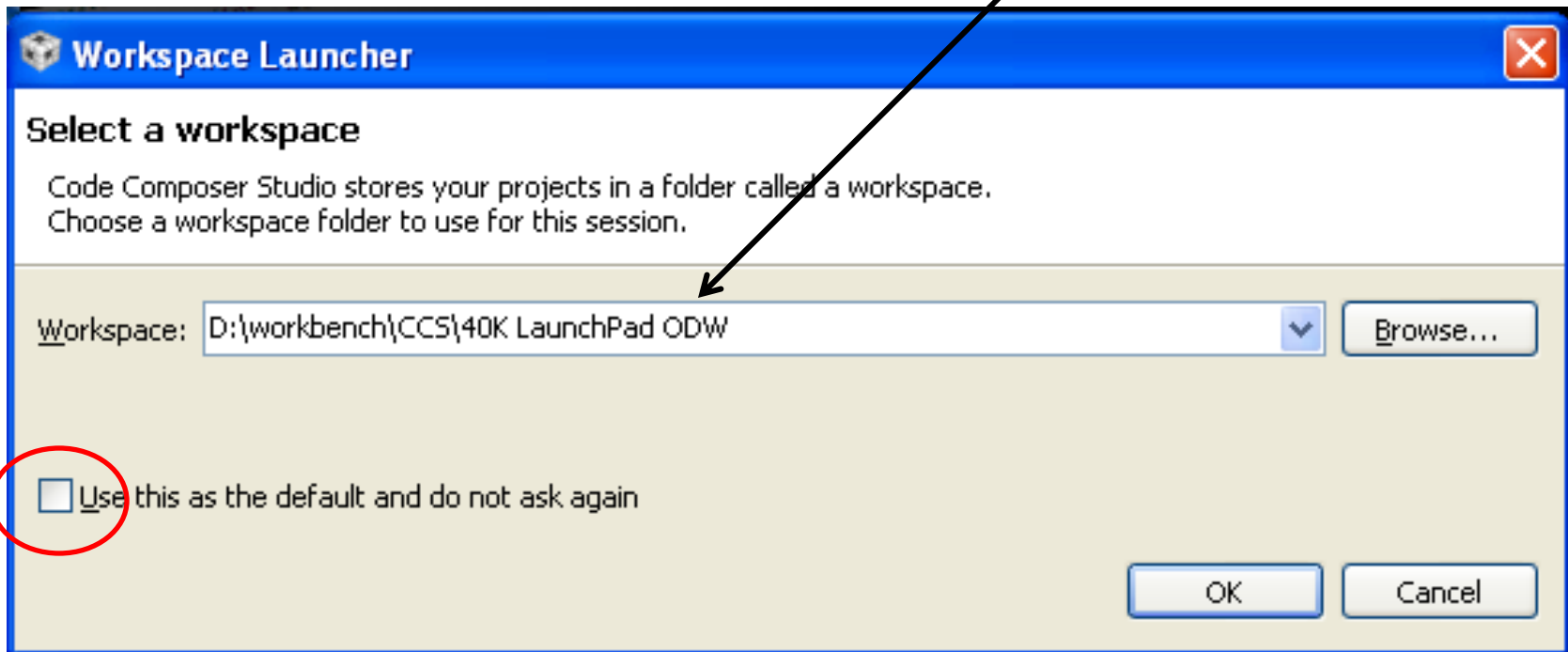


Step 1: 新建 CCS workspace

- 双击图标，启动 CCS v5
- 若未选择默认路径，设置 “Workspace” 路径

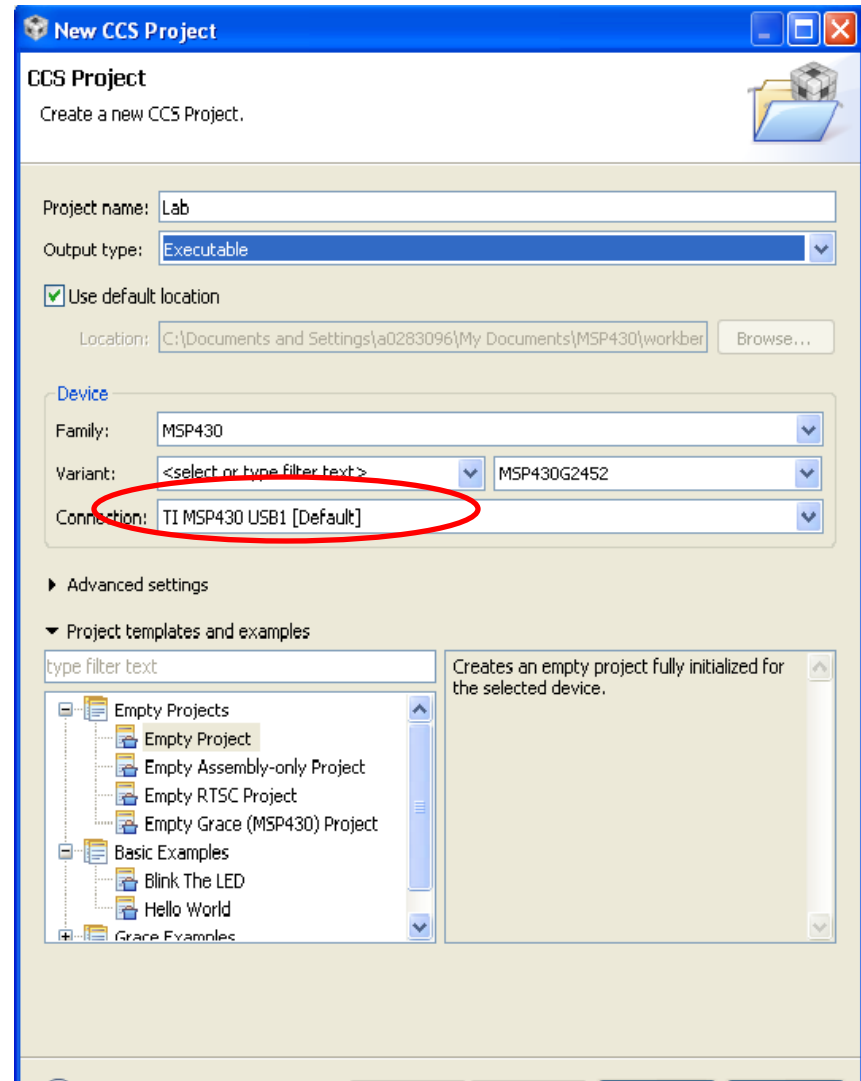
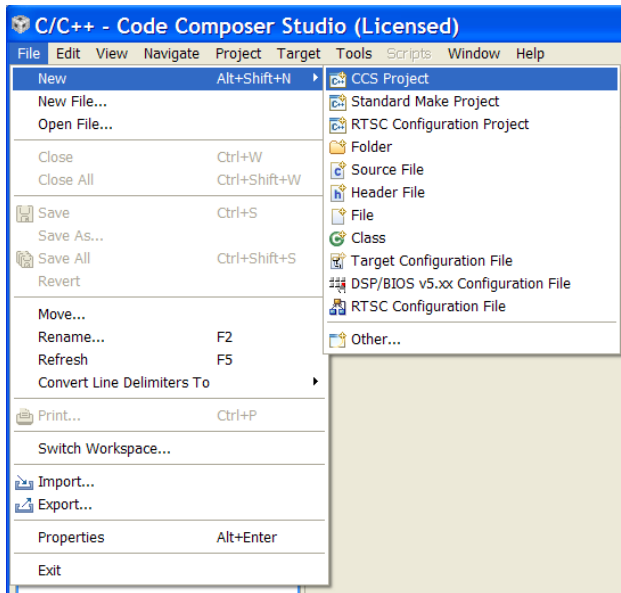


*尽量避免中文字符
(桌面?)



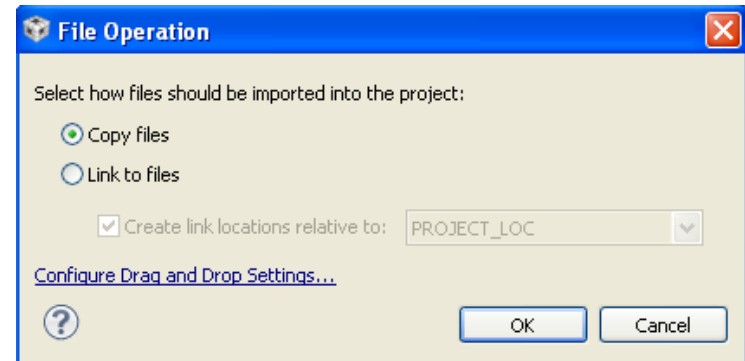
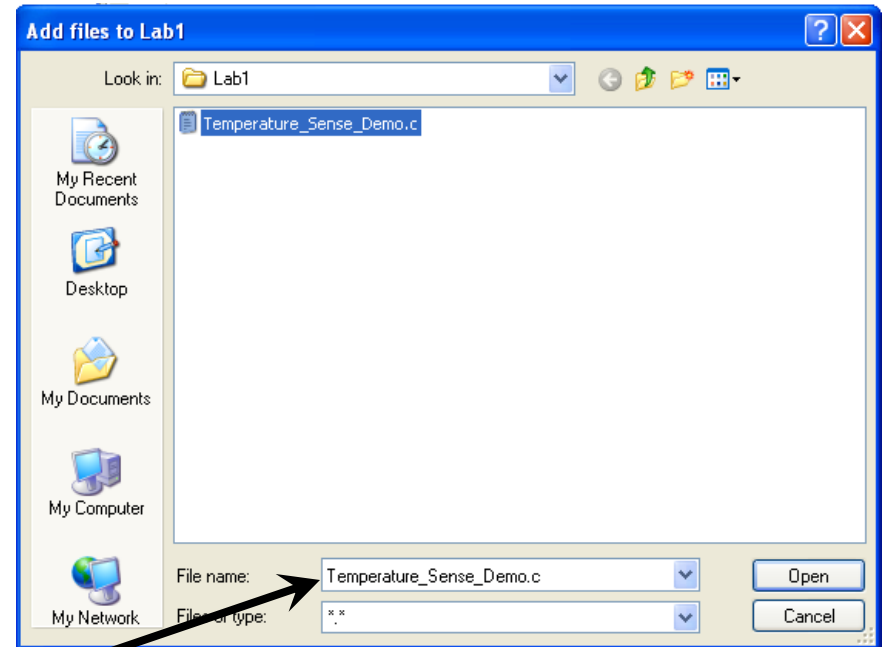
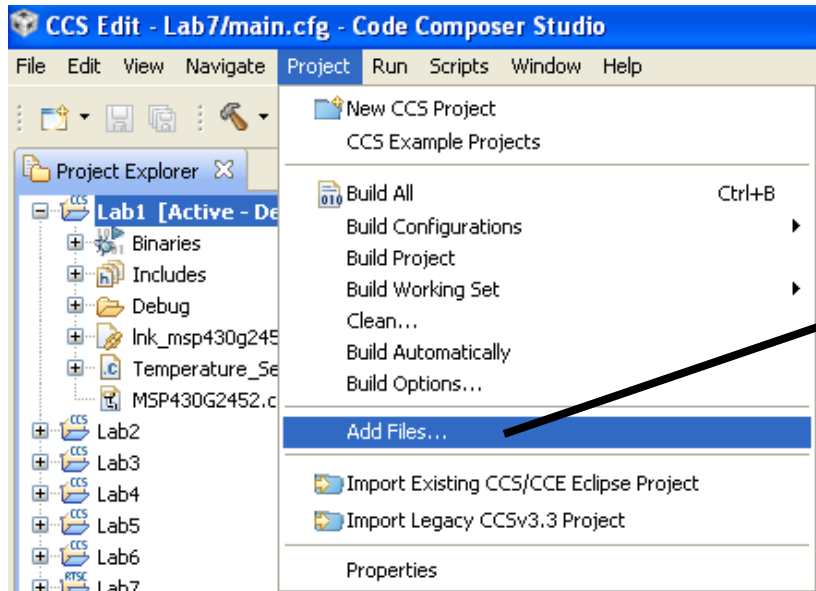
Step 2: 创建 CCS Project

- File > New > CCS Project
- Project 名称: Lab1
- Device>Family: MSP430
- Variant: MSP430F5529
- Project templates and example



Step 3: 在CCS Project中添加文件

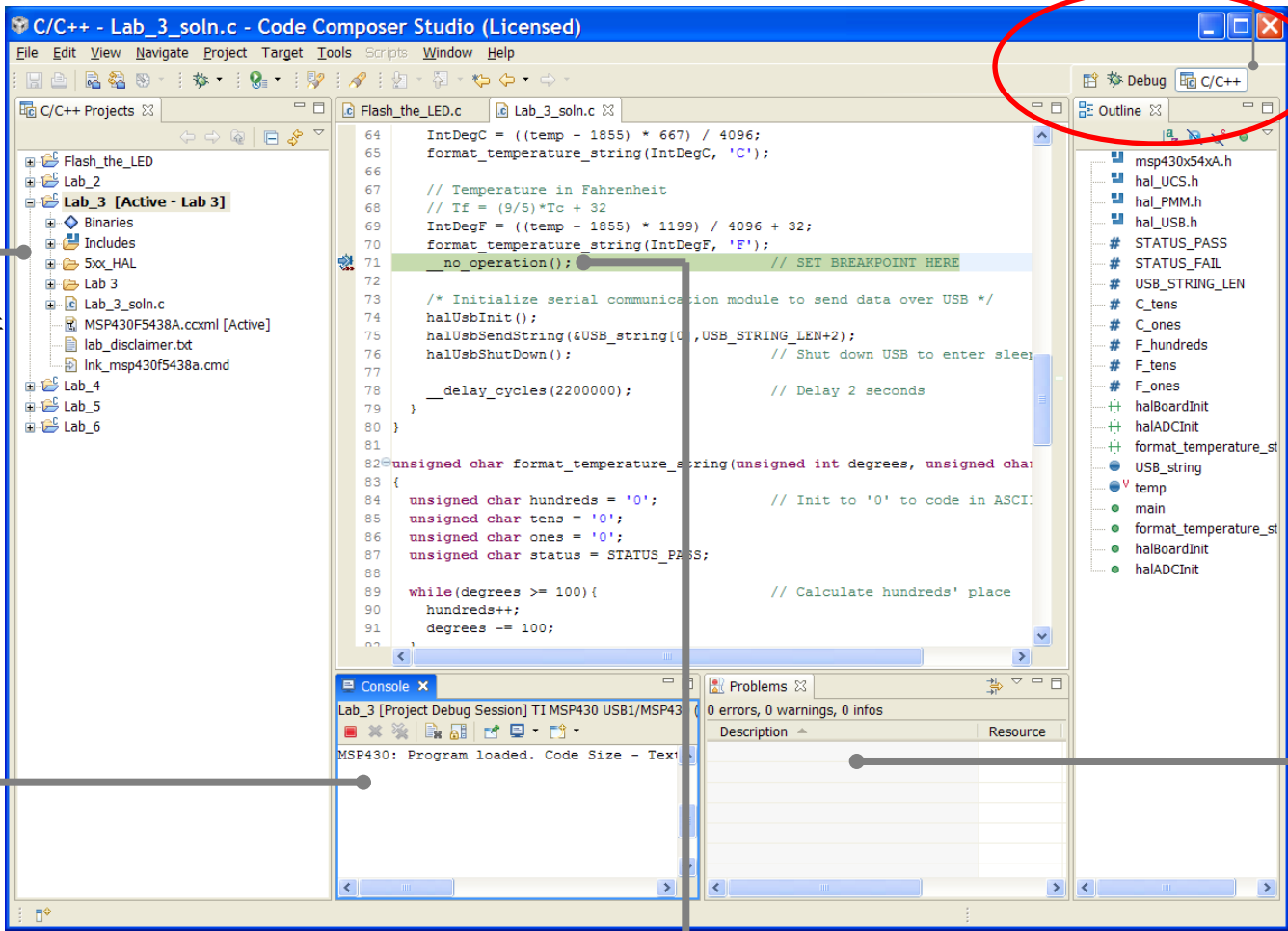
- Project > Add Files
- Navigate to Lab source folder
- And select :
Temperature_Sense_Demo.c



CCS 界面 – C/C++ Perspective Overview

C/C++ 编辑界面

显示当前所有Projects



Console
• 显示Build信息

Code Window

Problems View
• 显示Build结果，
包括error，warning
等

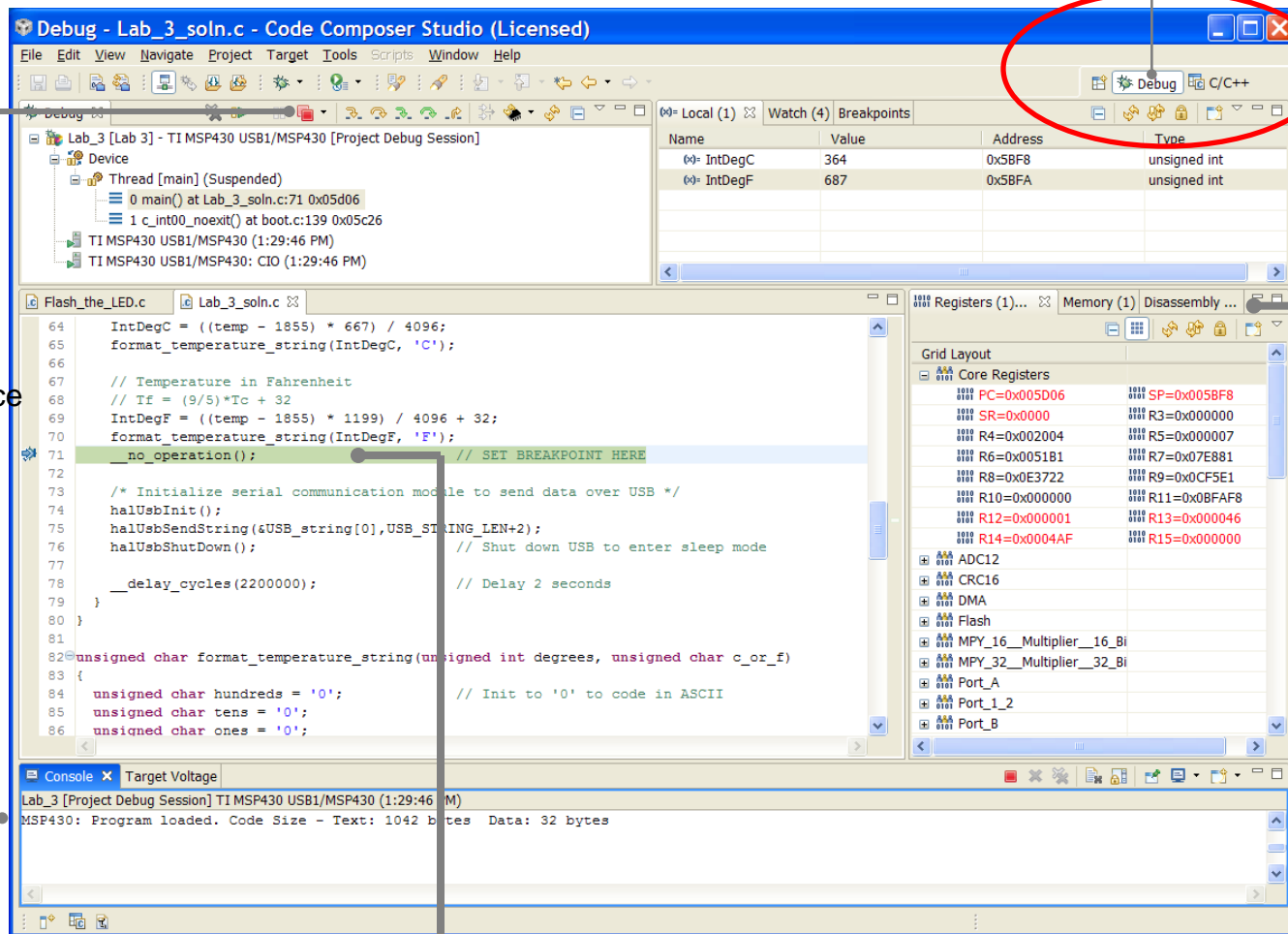


CCS 界面 – Debug Perspective Overview

Debug界面

Target 控制

- Start
- Stop
- Halt
- Stepping
- Stack Trace



实时, 430信息显示

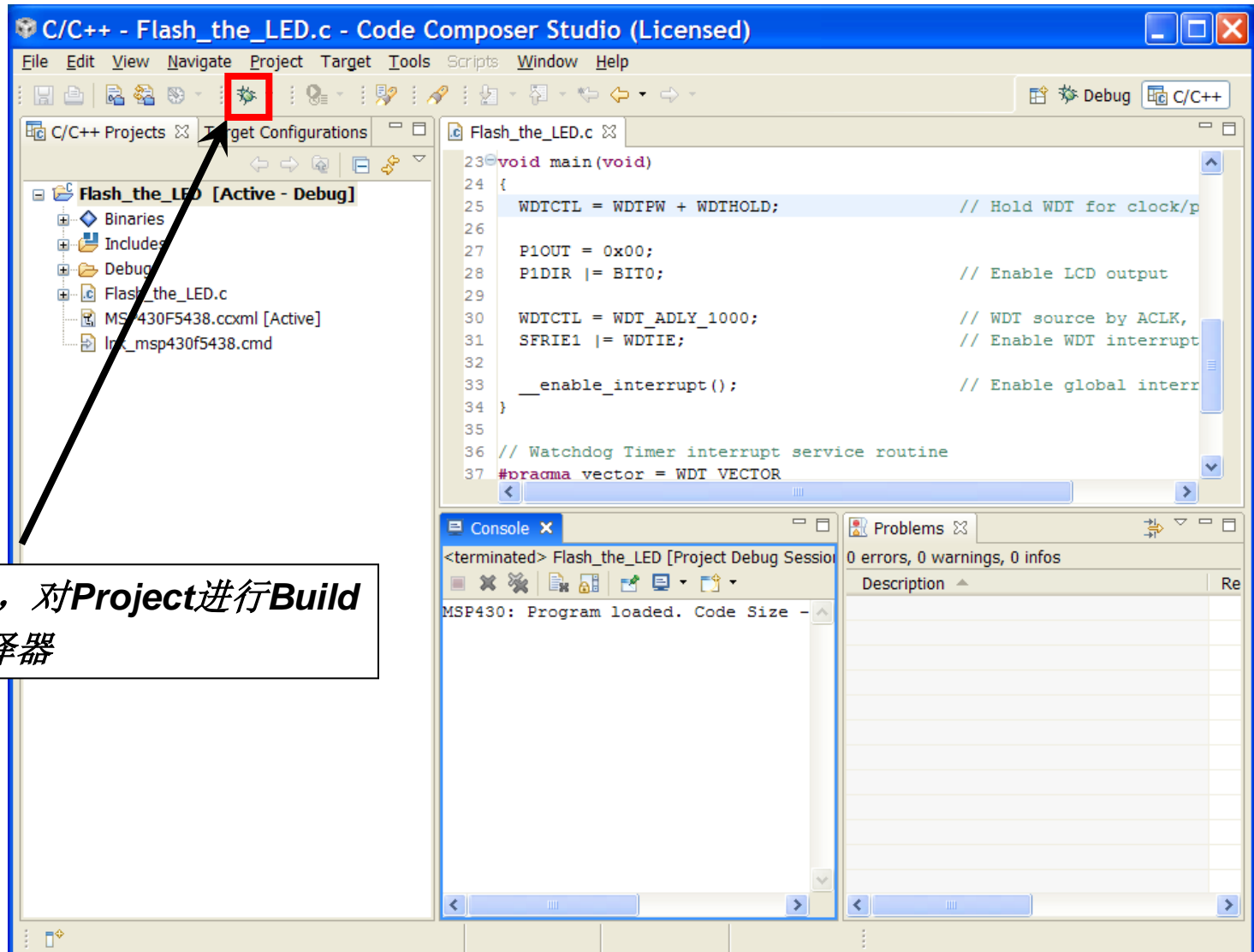
- 寄存器值
- Flash, RAM, Info segment
- 汇编代码显示

程序大小信息CodeSize

Code Window

- 实时断点, 运行状况显示

Step 4: Build & Debug a CCS Project



单击“BUG”，对Project进行Build
并连接到编译器

Step 5: 运行, 终止 CCS Project

“运行”

“停止”

Debug - Flash_the_LED.c - Code Composer Studio (Licensed)

File Edit View Navigate Project Target Tools Scripts Window Help

Flash_the_LED [Debug] - TI MSP430 USB1/MSP430 [Project Debug Session]

Device

Thread [main] (Suspended)

- 0 main() at Flash_the_LED.c:25 0x05c2c
- 1 c_int00_noinit_noexit() at boot.c:154 0x05c1e

TI MSP430 USB1/MSP430 (11:38:47 AM)

TI MSP430 USB1/MSP430: CIO (11:38:47 AM)

Flash_the_LED.c

```
23 void main(void)
24 {
25     WDTCTL = WDTPW + WDTHOLD; //
26
27     P1OUT = 0x00;
28     P1DIR |= BIT0; //
29
30     WDTCTL = WDT_ADLY_1000; //
31     SFRIE1 |= WDTIE; //
32
33     __enable_interrupt(); //
34 }
35
36 // Watchdog Timer interrupt service routine
37 #pragma vector = WDT_VECTOR
38 __interrupt void WDT_ISR(void)
```

Disassembly (main)

Enter location here

main:

- 0x05c2c: 40B2 5A80 015C MOV.W #0x5a80, &Watchdog_Timer
- 0x05c32: 43C2 0202 CLR.B &Port_1_2_P1OUT
- 0x05c36: D3D2 0204 BIS.B #1, &Port_1_2_P1DIR
- 0x05c3a: 40B2 5A3C 015C MOV.W #0x5a3c, &Watchdog_Timer
- 0x05c40: D392 0100 BIS.W #1, &SFR_Special_Functi
- 0x05c44: D232 EINT
- 0x05c46: 0110 RETA

_system_pre_init:

- 0x05c48: 431C MOV.W #1, R12
- 0x05c4a: 0110 RETA

C\$EXIT, abort:

- 0x05c4c: 4303 NOP

C\$L1:

- 0x05c4e: 3FFF JMP (C\$L1)

nop:

Console

Flash_the_LED [Project Debug Session] TI MSP430 USB1/MSP430 (11:38:47 AM)

MSP430: Program loaded. Code Size - Text: 82 bytes Data: 4 bytes

I/O引脚寄存器

1. 方向寄存器PxDIR

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
PxDIR.7	PxDIR.6	PxDIR.5	PxDIR.4	PxDIR.3	PxDIR.2	PxDIR.1	PxDIR.0

Bit = 0: The port pin is switched to input direction

Bit = 1: The port pin is switched to output direction

2. 输出寄存器PxOUT

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
PxOUT.7	PxOUT.6	PxOUT.5	PxOUT.4	PxOUT.3	PxOUT.2	PxOUT.1	PxOUT.0

Bit = 0: The output is low

Bit = 1: The output is high

其他常用寄存器包括P1IN、P1SEL、P1REN、P1IFG、P1IE等，详见User's Guide第8章节。

I/O口常用操作

1. 将特定bit置1

```
PxOUT |= BIT1 + BIT6;
```

2. 将特定bit置0

```
PxOUT &= ~(BIT1 + BIT6);
```

3. 特定bit取反

```
PxOUT ^= BIT1 + BIT6;
```

Lab_0: 闪动LED _code

```
#include <msp430.h>

int main(void)
{
    WDTCTL = WDTPW + WDTHOLD;           // Stop watchdog timer
    P1DIR |= 0x01;                       // Set P1.0 to output direction

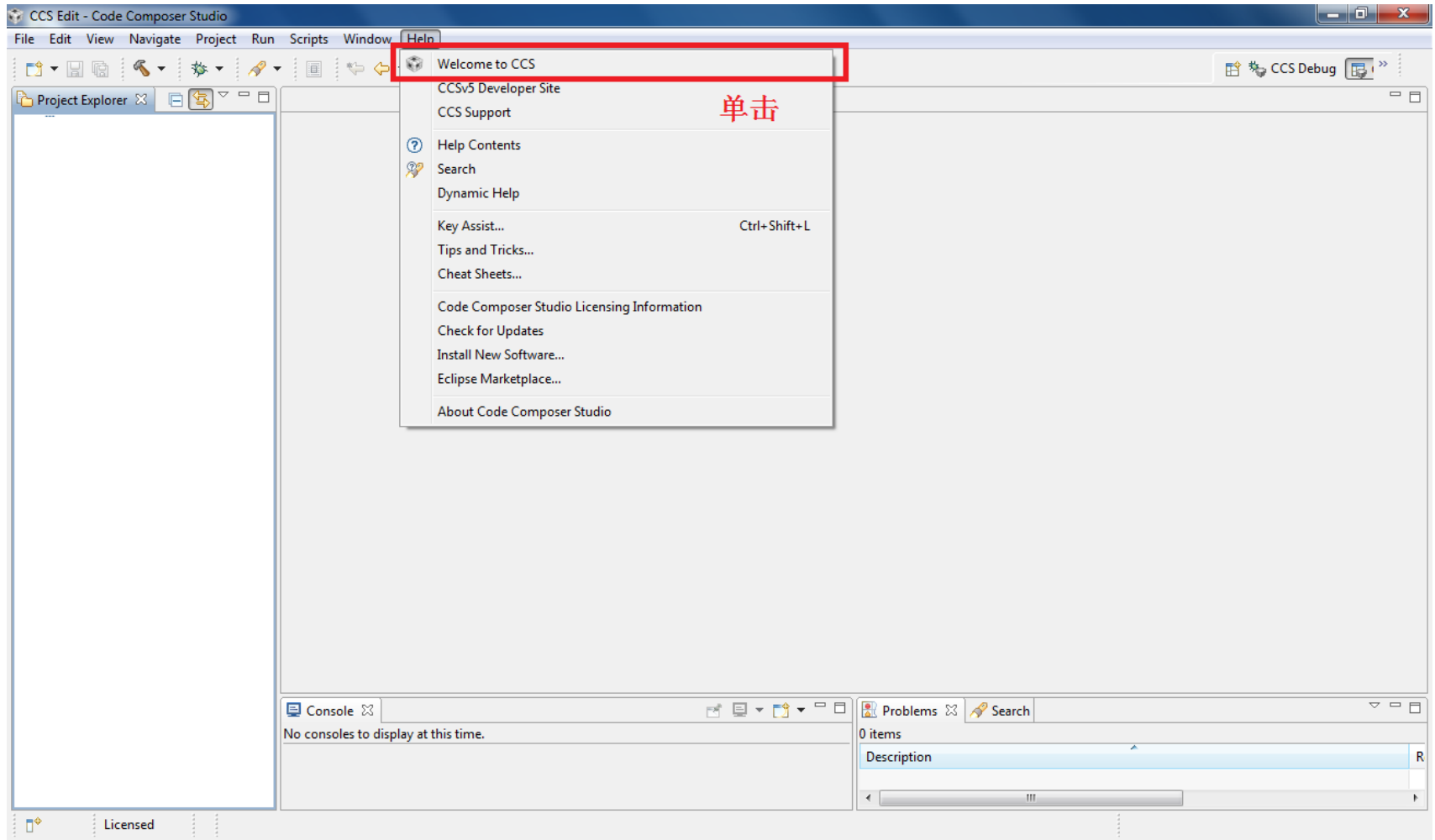
    for (;;)
    {
        volatile unsigned int i;        // volatile to prevent optimization

        P1OUT ^= 0x01;                  // Toggle P1.0 using exclusive-OR

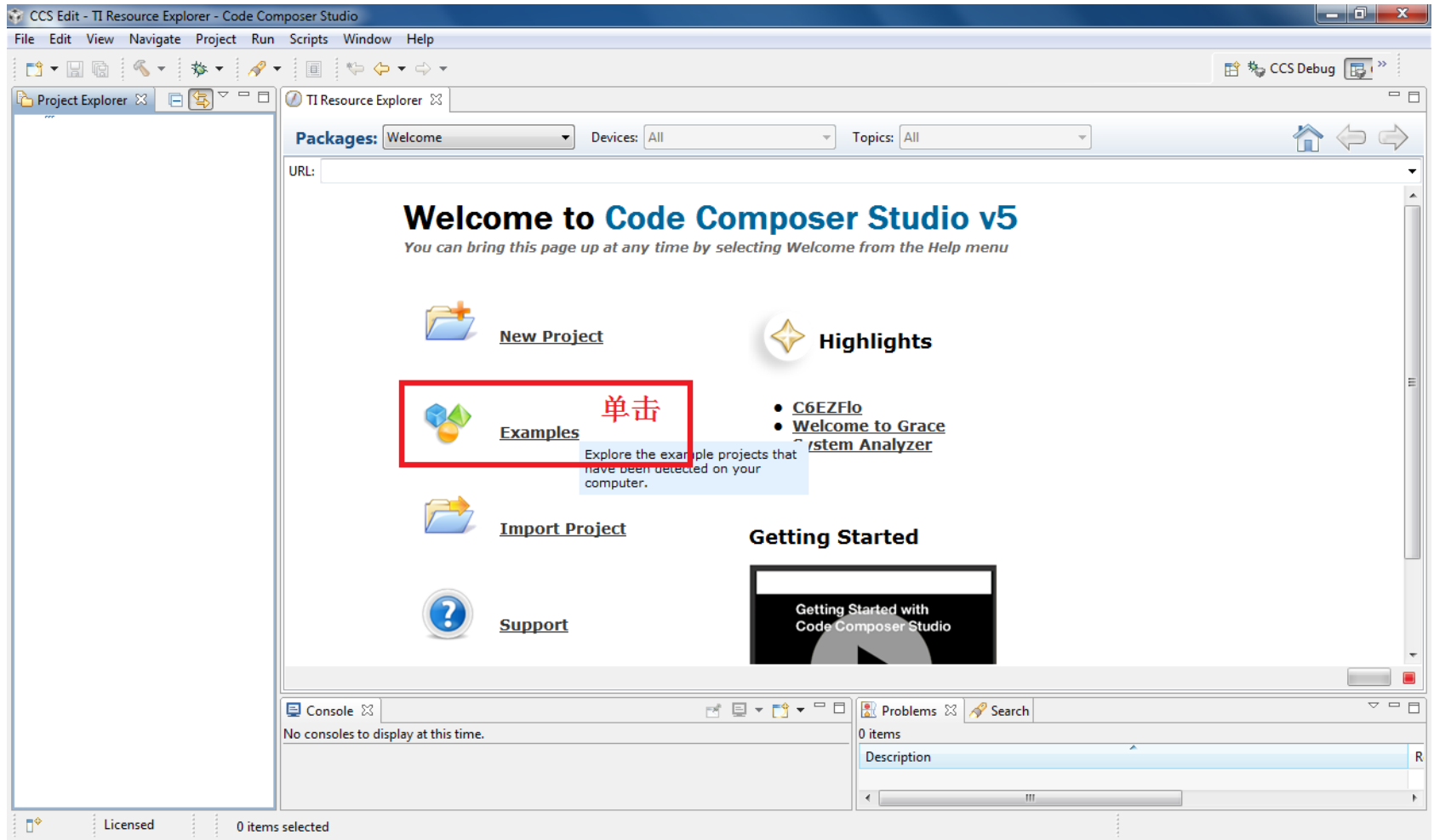
        i = 10000;                      // SW Delay
        do i--;
        while (i != 0);
    }
}
```

这段代码为什么不够好？

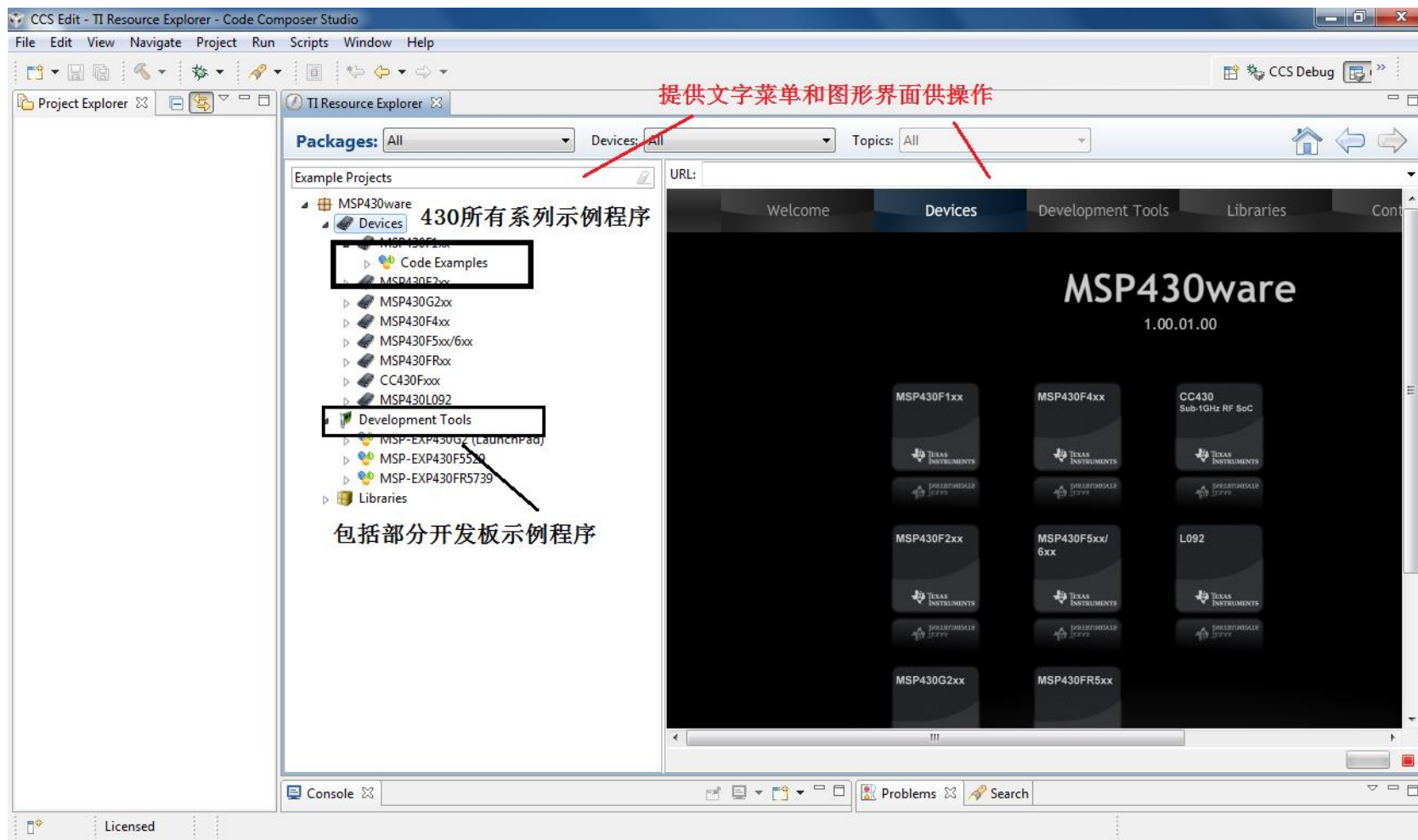
430Ware—快速程序开发工具



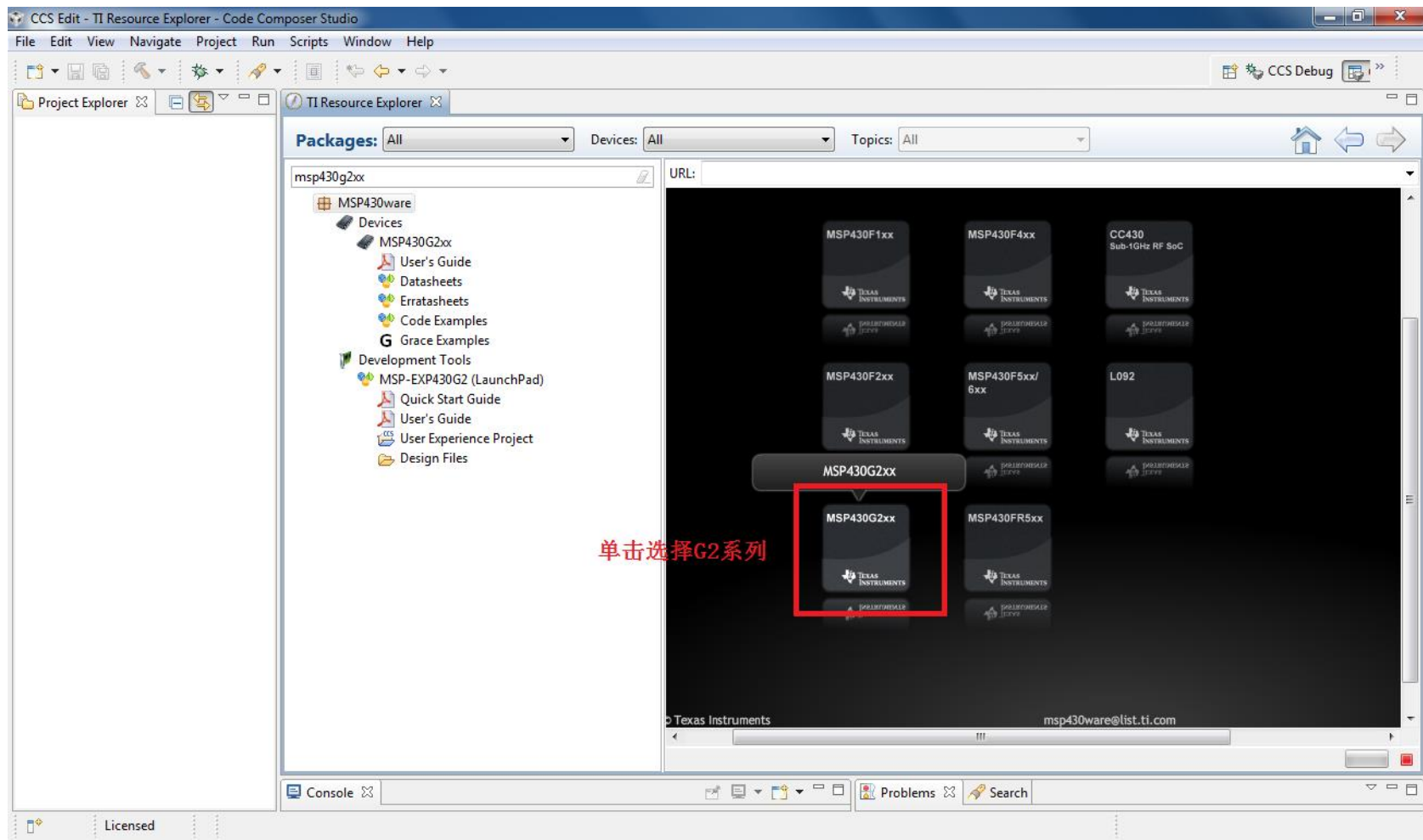
430Ware—快速程序开发工具



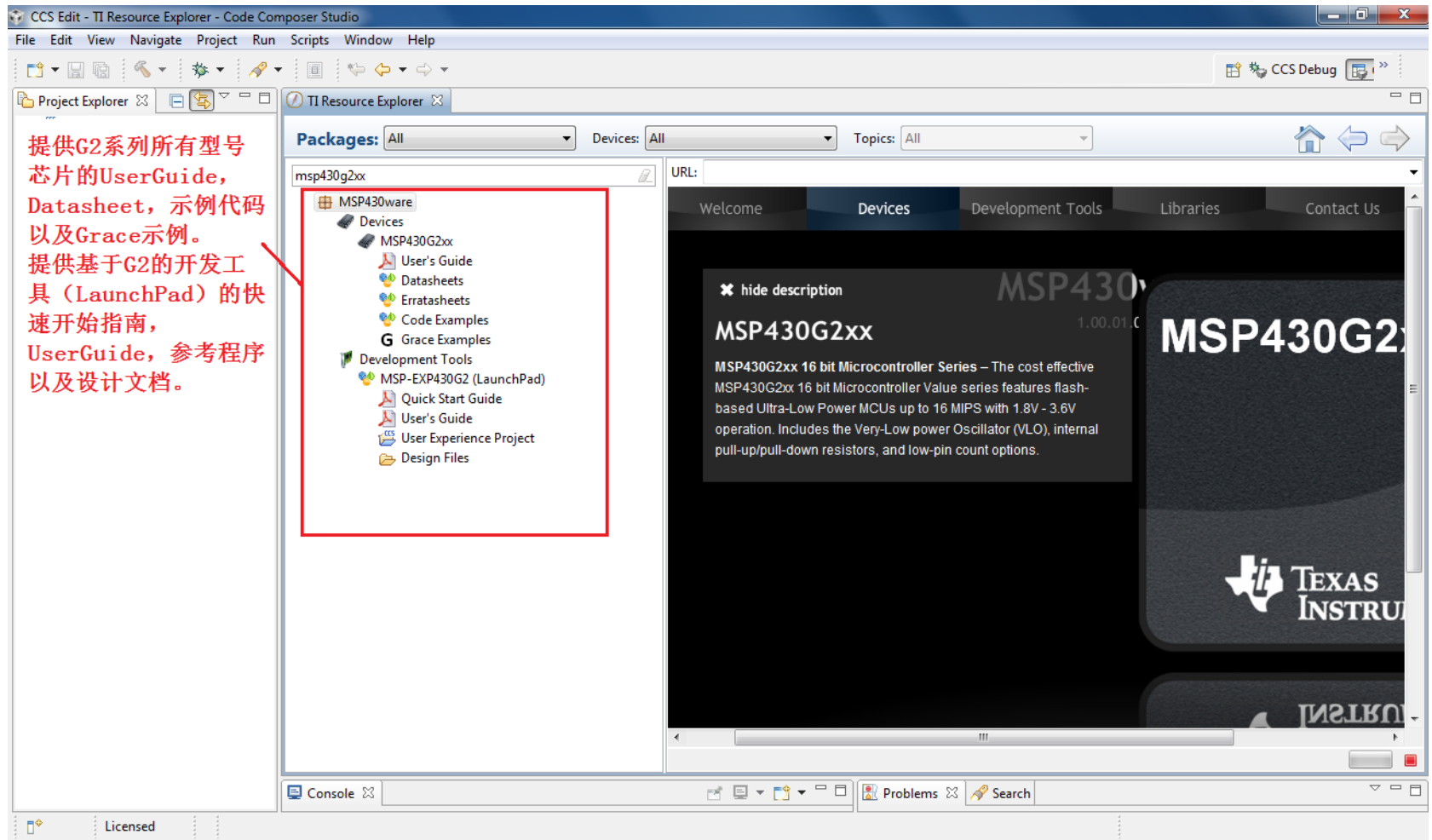
430Ware—快速程序开发工具



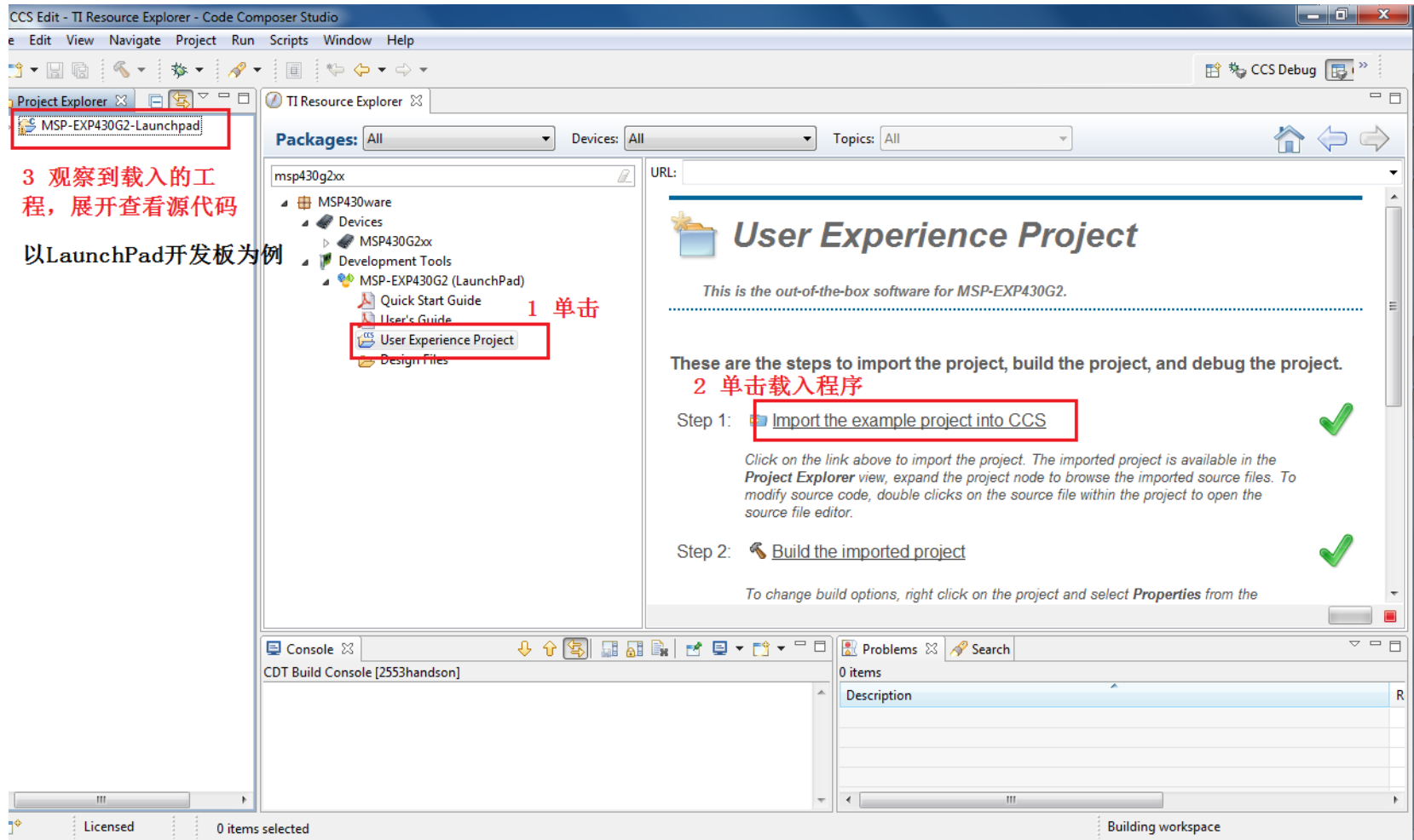
430Ware—快速程序开发工具



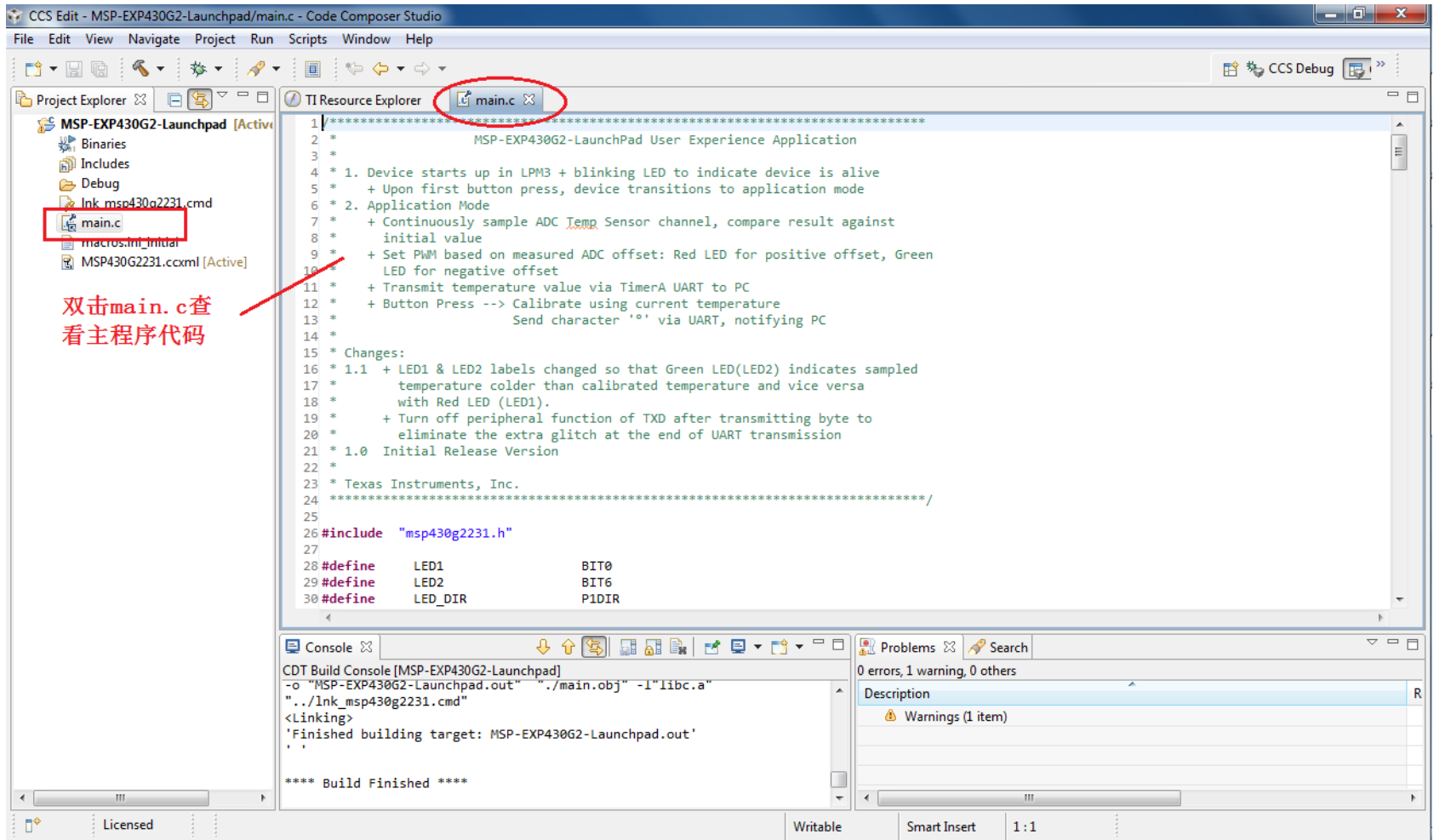
430Ware—快速程序开发工具



430Ware—快速程序开发工具

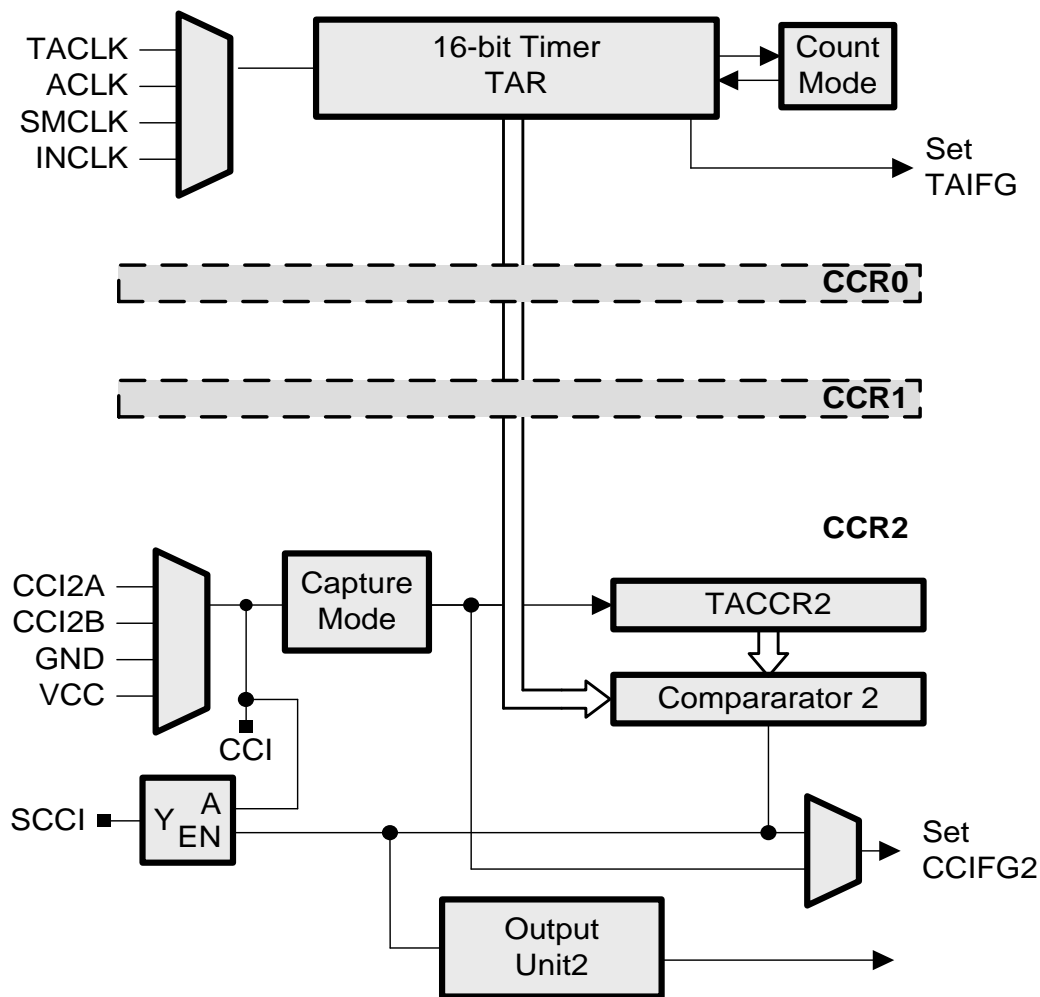


430Ware—快速程序开发工具



计时器_A

- 异步**16-Bit**计时器/计数器
- 连续的、自上而下的up count模式
- 多次捕获/比较寄存器
- **PWM**输出
- 中断矢量寄存器，以便快速解码
- 可以触发**DMA**转移
- 可用于所有**MSP430**产品



计时器_A 计数模式

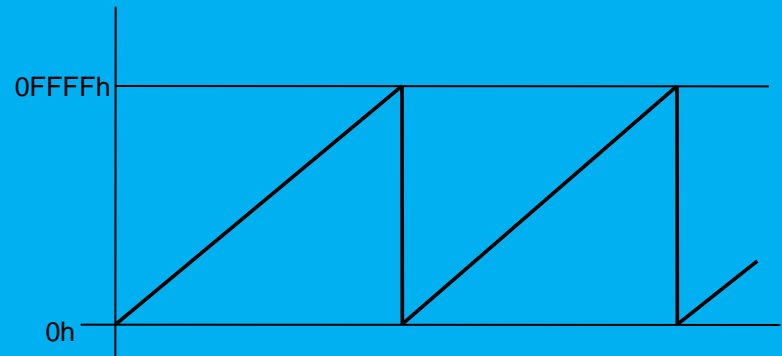
Stop/Halt

计时器停止



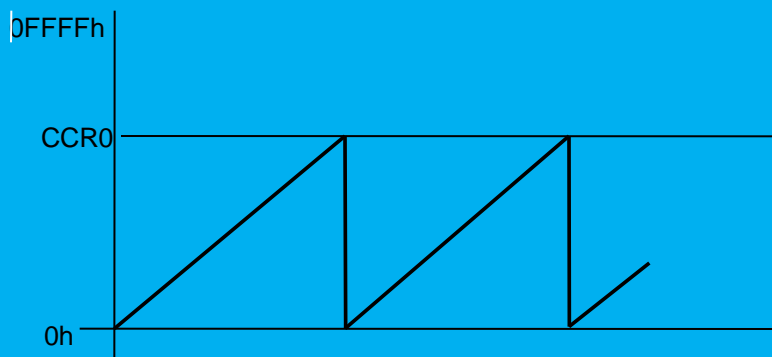
Continuous

计时器连续加总



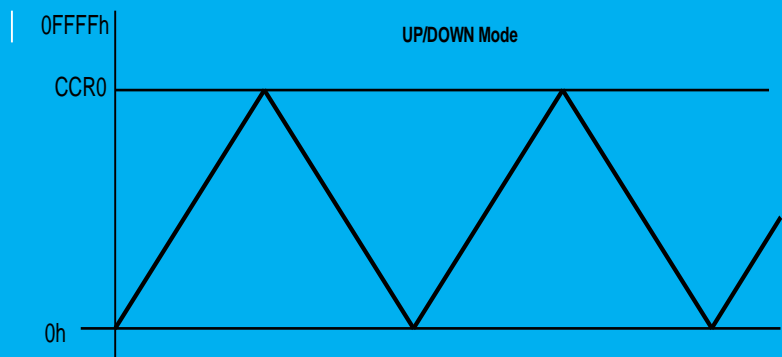
Up

计时器在0及CCR0之间计数



Up/Down

计时器在0及CCR0及0之间计数



CCR – 计数比较寄存器

时钟系统



- Very Low Power/Low Frequency Oscillator (VLO)

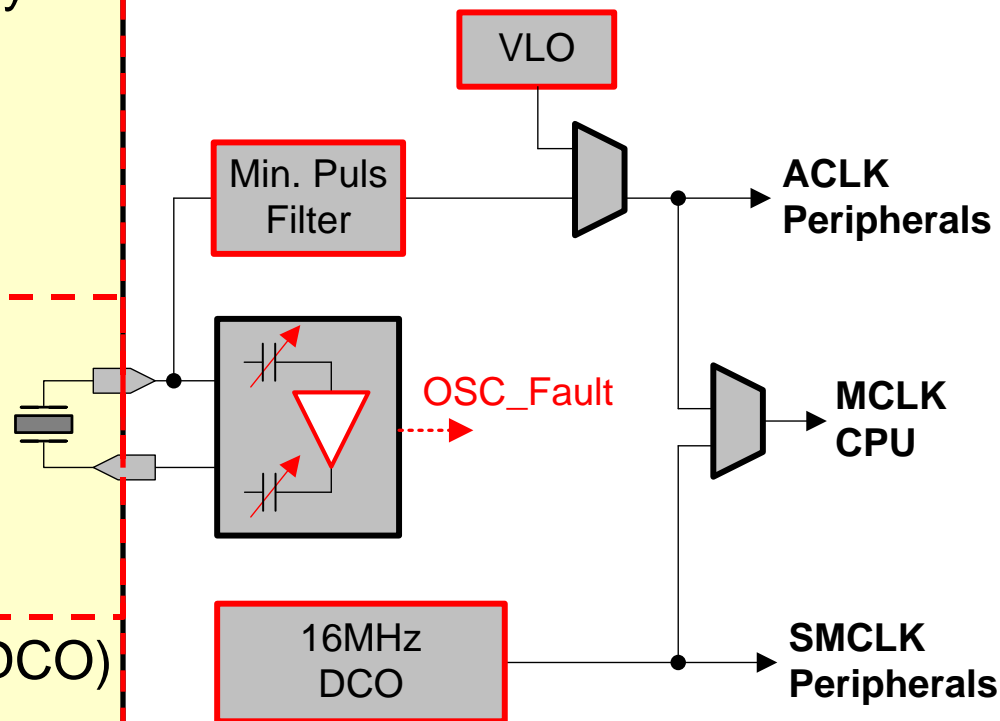
- 4 – 20kHz (典型值12kHz)
- 500nA 待机流耗
- 0.5%/° C and 4%/V 漂移

- 外接晶体振荡器(LFXT1, 低频)

- 片内可编程电容
- 故障保护 OSC_Fault
- 脉冲滤波器

- Digitally Controlled Oscillator (DCO)

- 0-to-16MHz
- $\pm 3\%$ 容差
- 出厂校准 (Flash I.M.)



上电后:

- MCLK 和 SMCLK 由 DCOCLK 提供 (~1.1 MHz)
- ACLK 由 LFXT1CLK 提供 (LF 模式, 6pF 内部负载电容)

基本时钟控制寄存器



5.3.3 BCSCTL2, Basic Clock System Control Register 2

7	6	5	4	3	2	1	0
SELMx		DIVMx		SELS	DIVSx		DCOR ⁽¹⁾⁽²⁾
rw-0		rw-0		rw-0	rw-0		rw-0
SELMx	Bits 7-6	Select MCLK. These bits select the MCLK source.					
		00 DCOCLK					
		01 DCOCLK					
		10 XT2CLK when XT2 oscillator present on-chip. LFXT1CLK or VLOCLK when XT2 oscillator not present on-chip.					
		11 LFXT1CLK or VLOCLK					
DIVMx	Bits 5-4	Divider for MCLK					
		00 /1					
		01 /2					
		10 /4					
		11 /8					
SELS	Bit 3	Select SMCLK. This bit selects the SMCLK source.					
		0 DCOCLK					
		1 XT2CLK when XT2 oscillator present. LFXT1CLK or VLOCLK when XT2 oscillator not present					
DIVSx	Bits 2-1	Divider for SMCLK					
		00 /1					
		01 /2					
		10 /4					
		11 /8					
DCOR	Bit 0	DCO resistor select. Not available in all devices. See the device-specific data sheet.					
		0 Internal resistor					
		1 External resistor					

TimerA输出模式

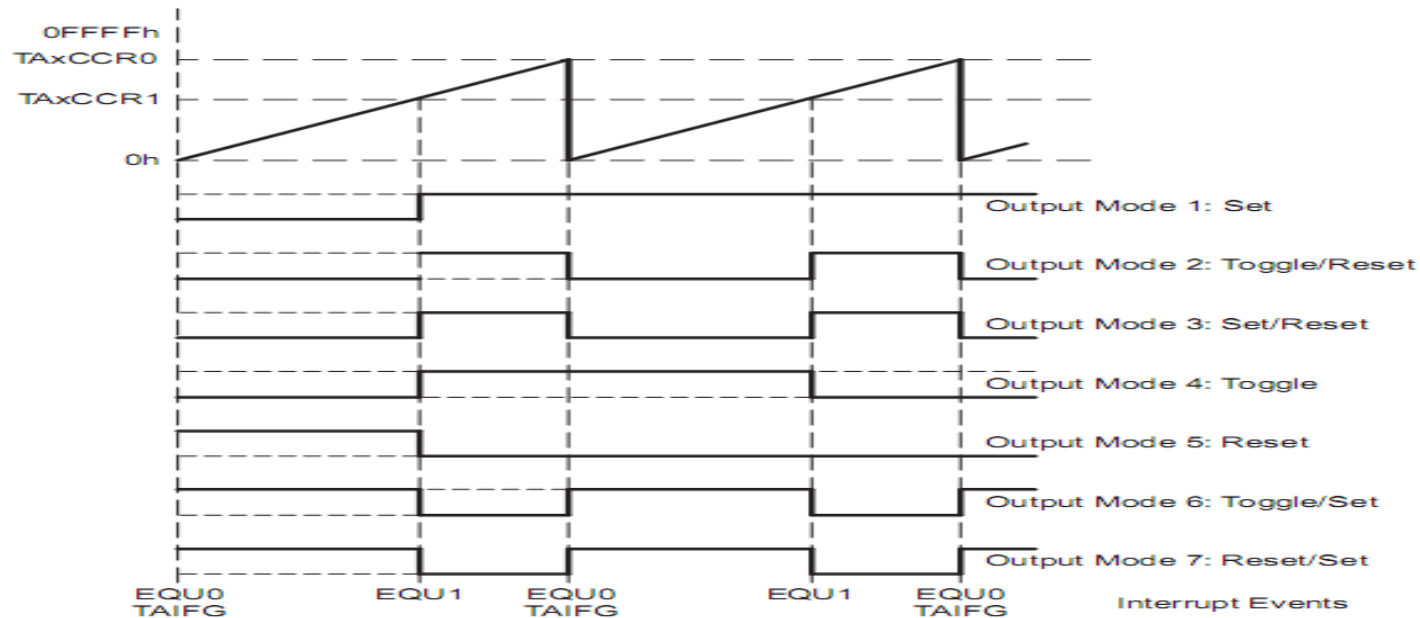
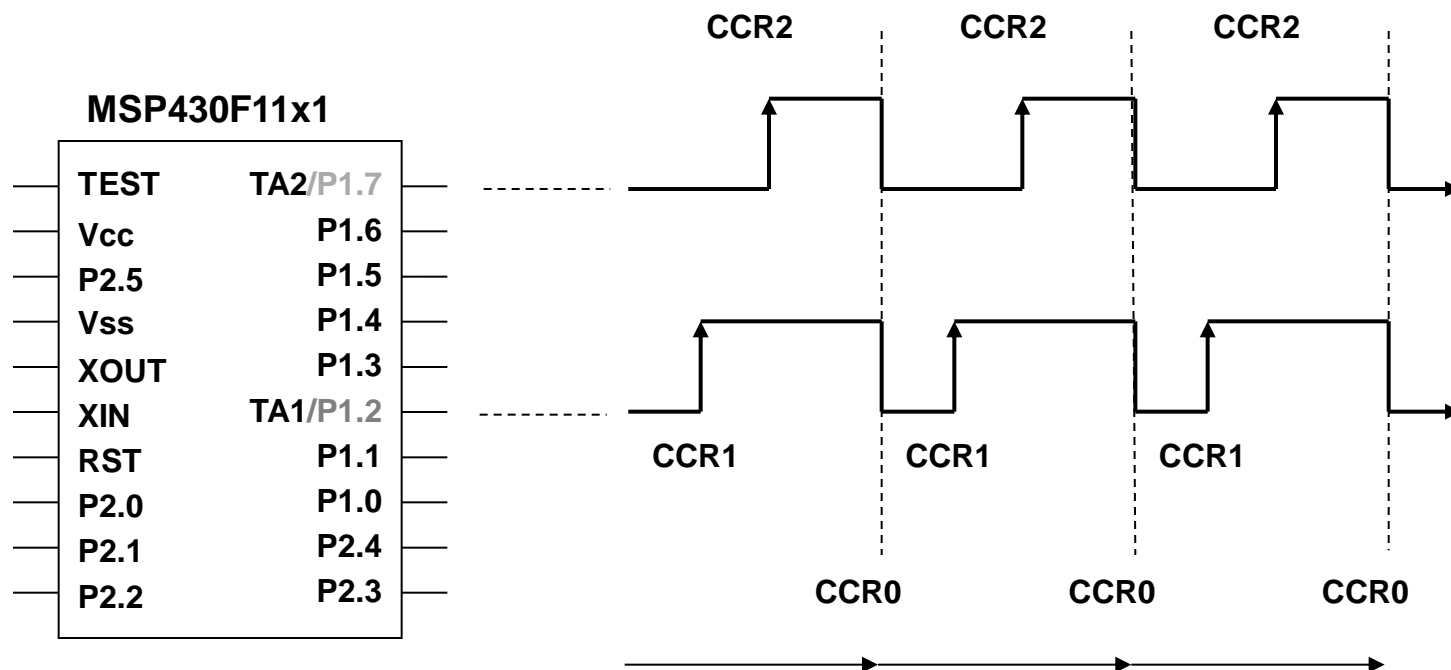


Figure 15-12. Output Example – Timer in Up Mode

Table 15-2. Output Modes

OUTMODx	Mode	Description
000	Output	The output signal OUTn is defined by the OUT bit. The OUTn signal updates immediately when OUT is updated.
001	Set	The output is set when the timer <i>counts</i> to the TAXCCRn value. It remains set until a reset of the timer, or until another output mode is selected and affects the output.
010	Toggle/Reset	The output is toggled when the timer <i>counts</i> to the TAXCCRn value. It is reset when the timer <i>counts</i> to the TAXCCR0 value.
011	Set/Reset	The output is set when the timer <i>counts</i> to the TAXCCRn value. It is reset when the timer <i>counts</i> to the TAXCCR0 value.
100	Toggle	The output is toggled when the timer <i>counts</i> to the TAXCCRn value. The output period is double the timer period.
101	Reset	The output is reset when the timer <i>counts</i> to the TAXCCRn value. It remains reset until another output mode is selected and affects the output.
110	Toggle/Set	The output is toggled when the timer <i>counts</i> to the TAXCCRn value. It is set when the timer <i>counts</i> to the TAXCCR0 value.
111	Reset/Set	The output is reset when the timer <i>counts</i> to the TAXCCRn value. It is set when the timer <i>counts</i> to the TAXCCR0 value.

计时器_A PWM示例



- ◆ 完全自动
- ◆ 可以为每一个CCR生成不同工作周期的独立频率
- ◆ MSP430网站提供代码示例

完全自动

产生PWM



**TEXAS
INSTRUMENTS**