Kinetis高速ADC应用

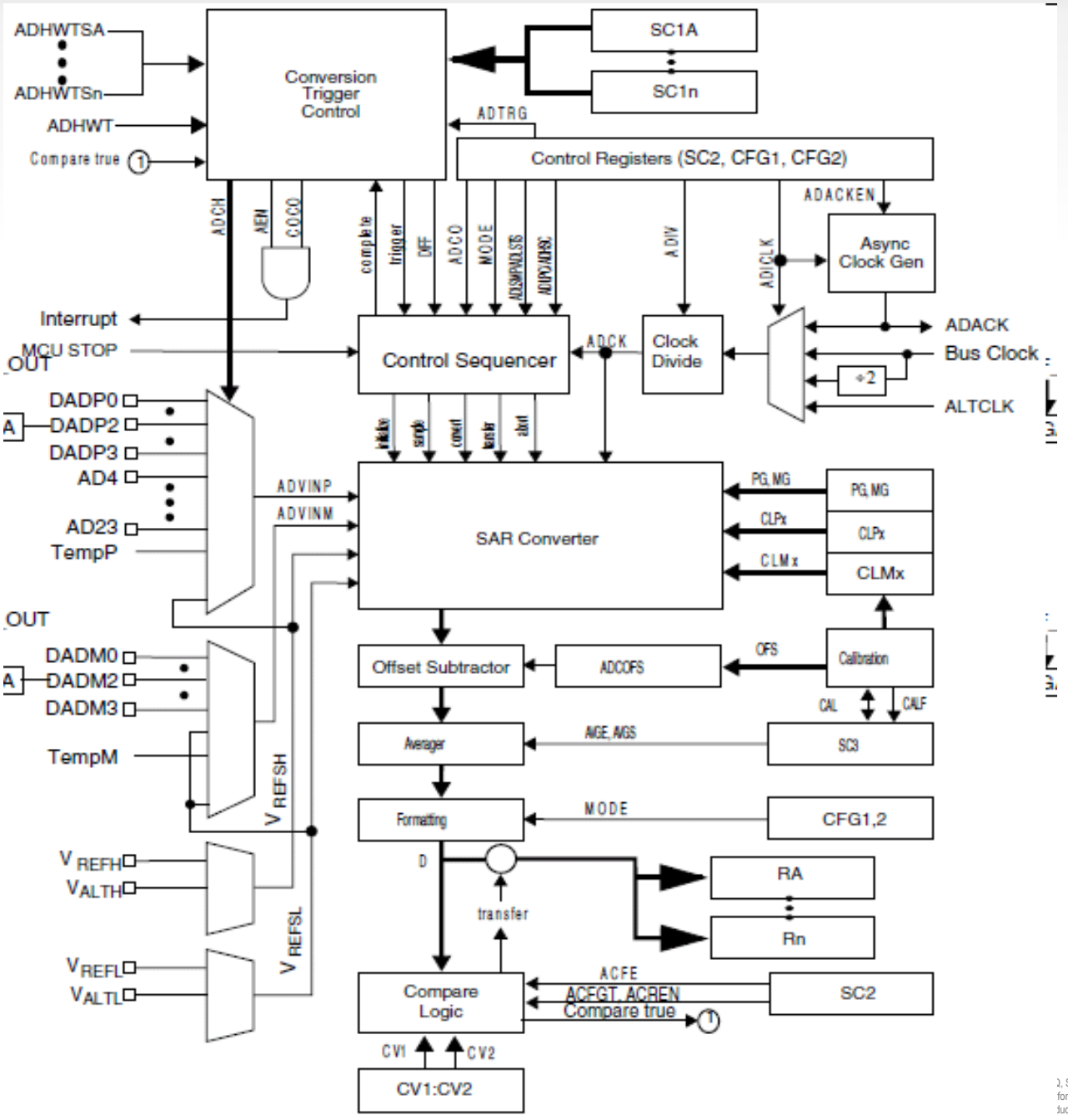
更多飞思卡尔技术资源、技术支持欢迎访问：

飞思卡尔中文支持社区：http://www.eefocus.com/bbs/forumall_171.html
飞思卡尔英文支持社区：https://community.freescale.com/

题　目

- Kinetis ADC、PDB 模块技术介绍
- ADC采样设计应用
- 独立ProcessorExpert与IAR结合使用技巧
- 两种软件方案设计
- 上手实验

ADC方块图



freescale™

ADC 模块特点

- 线性逐次逼近算法16位精度
- 4对差分、24个单端输入通道
- 输出结果：差分16位、13位、11位、9为模式，或者单端16位、12位、10位、8位模式
- 差分输出为2次补充16位有符号数
- 单端输出为无符号数，右侧对齐
- 单次采样或连续采样(单次采样完成自动返回空闲)
- 可配置采样时间、转换速度、功率
- AD转换完成、硬件平均完成标志和中断标志
- 四种可选时钟源（总线时钟、总线时钟/2、ALTCLK、异步时钟）
- 可在低功耗模式下做低噪声操作
- 可选的多个硬件触发源
- 可做数值自动比较(小于、大于、等于、范围内、超范围、或者编程的数值)
- 带温度传感器，可做温度补偿
- 硬件平均功能
- 可选内部或外部参考电压
- 自校验模式
- 带有最大64倍增益的可编程放大器

ADC模块管脚特征

1. 模拟电源 (VDDA)
   需要滤波后的干净电源。有的封装VADDA连接到VDD。

2. 模拟地 (VSSA)
   外接专用VSSA，或者接芯片内部的Vss。

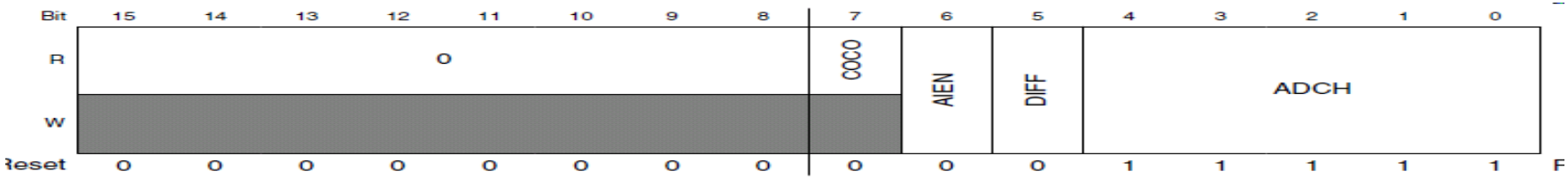
3. 电压参考Vref
   VREFSH 和 VREFSL电压对，是ADC转换的高压与低压参考。参考电压可以是内部或外部参考，由SC2的REFSEL决定。

4. 模拟通道输入 (ADx)
   24路单端模拟输入。每一路输入的选择由SC1寄存器中的ADSH、DIFF=0来决定。

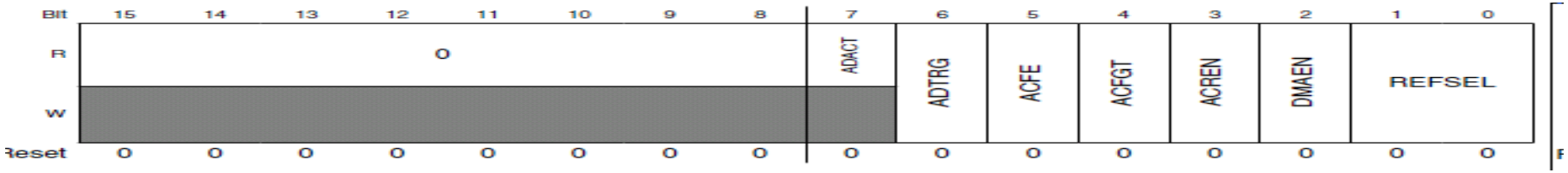
5. 差分模拟通道输入 (DADx)
   ADC 支持4对差分模拟通道，管脚对应 (DADPx and DADMx)。通道选择由SC1的ADCH、DIFF=1决定。差分管脚可以作为单端输入的管脚来使用。

ADC模块主要寄存器

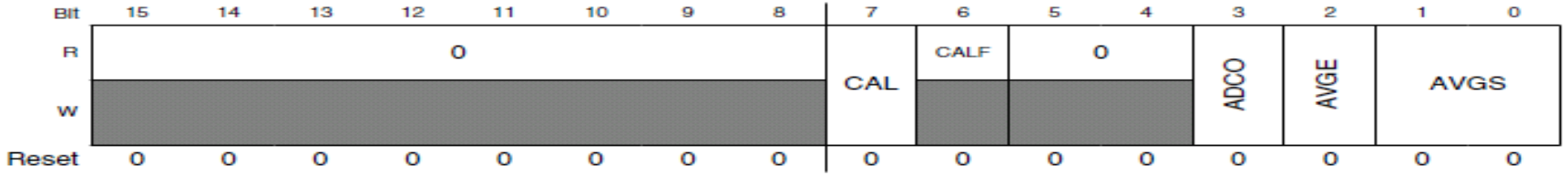
状态与控制寄存器 1 (ADCx_SC1)



状态与控制寄存器 2 (ADCx_SC2)

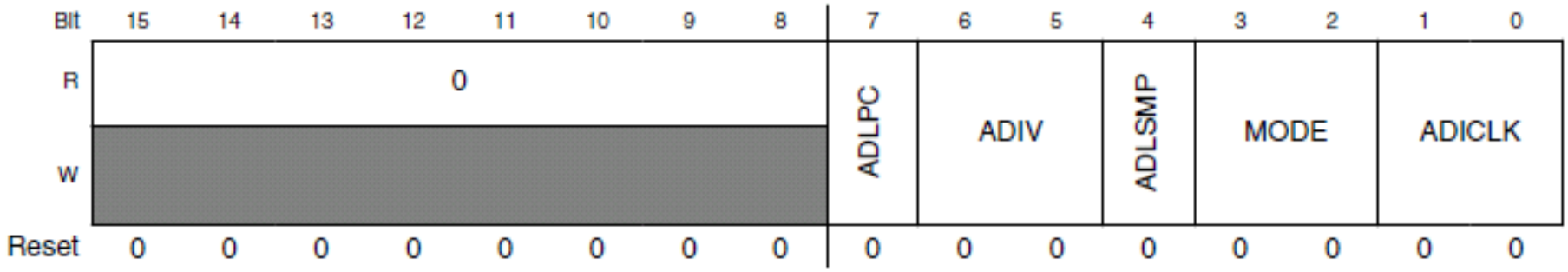


状态与控制寄存器3 (ADCx_SC3)

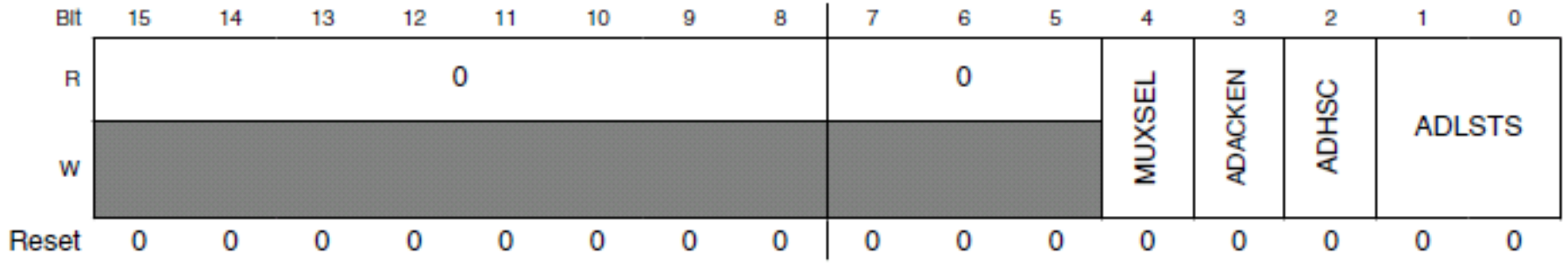


ADC模块主要寄存器（续）

ADC 配置寄存器1 (ADC*x*_CFG1)



ADC配置寄存器2 (ADC*x*_CFG2)



ADC模块主要寄存器（续）

ADC 结果寄存器 (ADC*x*_R*n*)

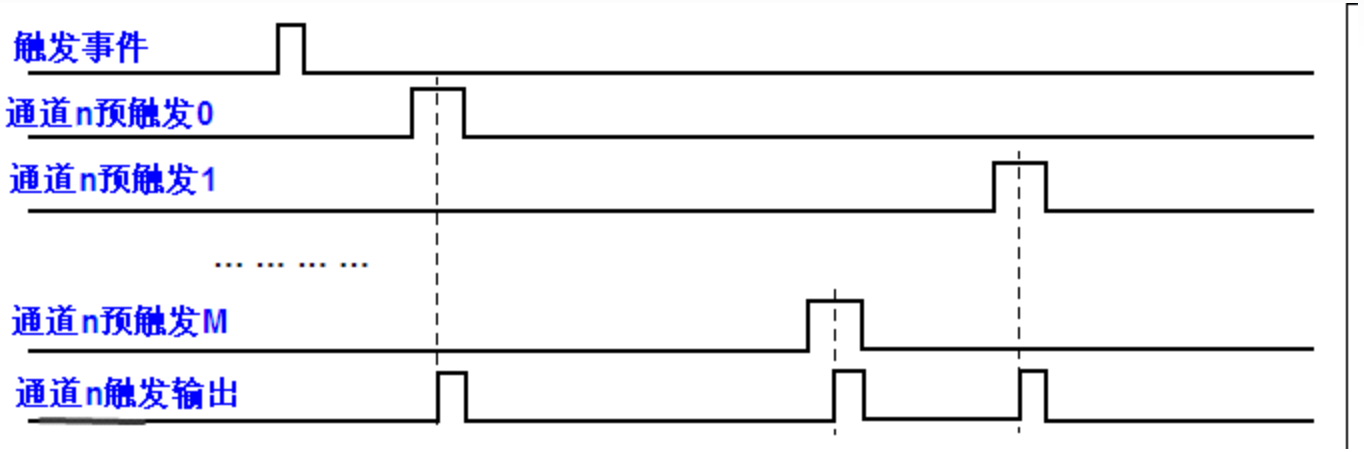
Conversion mode	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Format
16-bit differential	S	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	Signed 2's complement
16-bit single-ended	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	Unsigned right justified
13-bit differential	S	S	S	S	D	D	D	D	D	D	D	D	D	D	D	D	Sign extended 2's complement
12-bit single-ended	0	0	0	0	D	D	D	D	D	D	D	D	D	D	D	D	Unsigned right justified

Conversion mode	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Format
11-bit differential	S	S	S	S	S	S	D	D	D	D	D	D	D	D	D	D	Sign extended 2's complement
10-bit single-ended	0	0	0	0	0	0	D	D	D	D	D	D	D	D	D	D	Unsigned right justified
9-bit differential	S	S	S	S	S	S	S	S	D	D	D	D	D	D	D	D	Sign extended 2's complement
8-bit single-ended	0	0	0	0	0	0	0	0	D	D	D	D	D	D	D	D	Unsigned right justified

ADC硬件触发源PDB

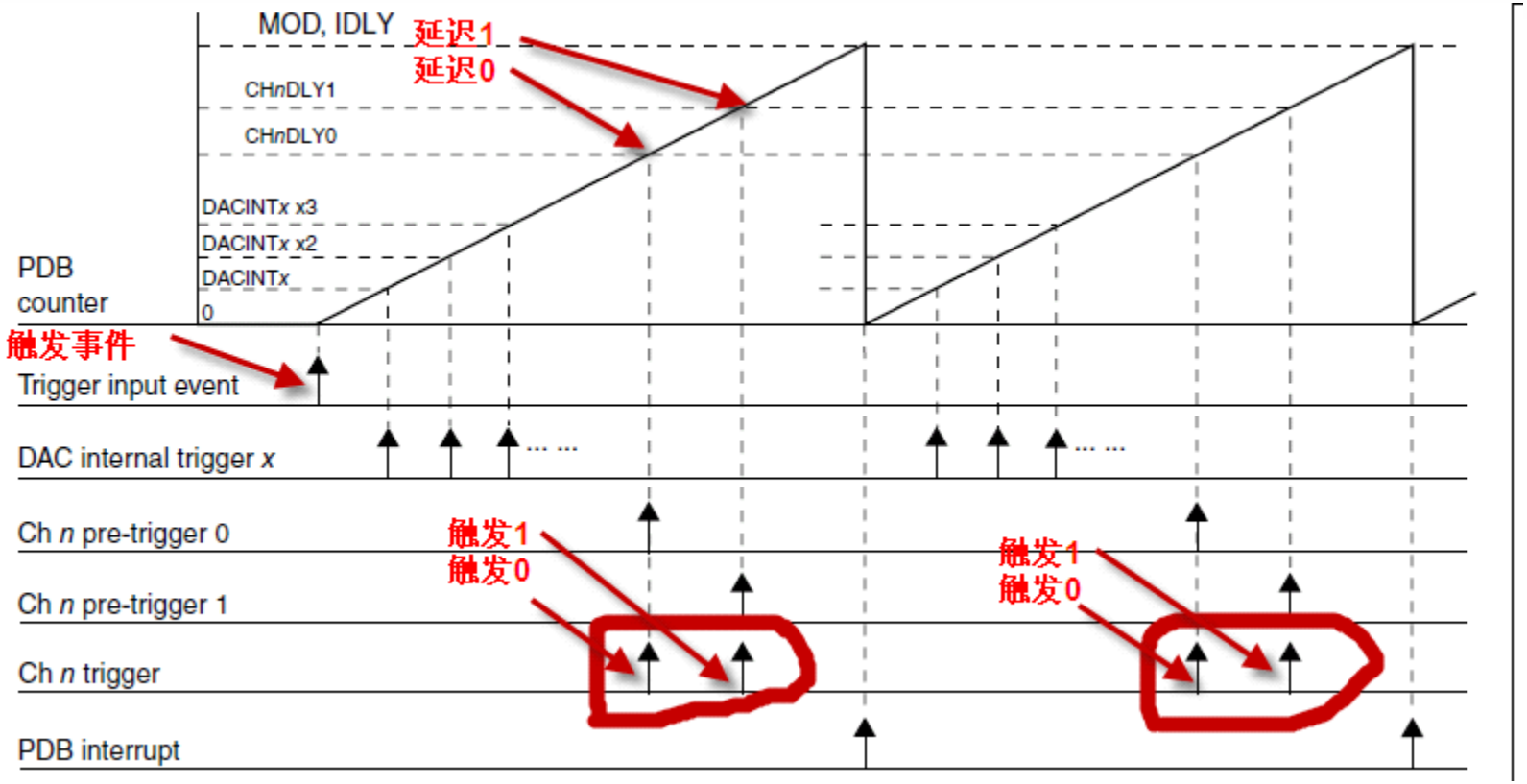
- PDB 特点：
  - 多达15个硬件输入源，一个软件触发源
  - 多达8个可配置通道，用于ADC硬件触发
  - 每个PDB通道与一个ADC模块相关联
  - 每个触发传输对应一个ADC硬件触发，共有8个预触发输出
  - 每个触发输出可以被独立使能或禁止
  - 每个预触发输出带一个16位延迟寄存器
  - 可选择旁路延迟寄存器
  - 可选择单次或连续模式工作模式
  - 支持背对背模式，可在ADC转换完成后触发下一个PDB通道
  - 一个可编程的延迟中断
  - 一个时序错误中断
  - 每个预触发带通道标志和时序错误标志
  - 支持DMA
  - 多达8路DAC内部触发
  - 每路内部触发对应一个DAC模块
  - 可选择旁路延迟寄存器
  - 可选择外部触发源
  - 八个脉冲输出
  - 脉冲输出使能或禁止
  - 可编程脉宽

PDB 预触发和触发输出



每个PDB通道与一个ADC模块相关联。PDB通道n的预输出0~M，与触发输出、ADC的硬件触发选择和硬件触发输入相连接。

PDB ADC 触发和 DAC 内部触发



ADC时钟与转换速率

符号	描述	条件	最小值	典型值	最大值	单位	注释
fADCK	ADC转换时钟频率	≤13 bit modes	1.0	NC	18.0	Mhz	
fADCK	ADC转换时钟频率	16 bit modes	2.0	NC	12.0	Mhz	
Crate	ADC转换率	≤13bit 模式 无硬件平均 连续转换 外设时钟频率 =50MHz	18.484	NC	818.330	Ksps	
Crate	ADC转换率	16 bit模式 无硬件平均 连续转换 外设时钟频率 =50MHz	37.037	NC	361.402	Ksps	

ADC校准

- ADC校准的目的是：大幅减小因内部电容器组的变化而造成的准精度误差。在程序初始化阶段，ADC模块采样之前，先对ADC进行校准，之后再进行采样。若在环境发生变化时(如电压、温度变化)，也可以做一下校准。

- 设置正、反偏校准值。偏移值自动存储到ADC偏移修正寄存器OFS和正反偏移校准寄存器(CLPD, CLPS, CLP4, CLP3, CLP2, CLP1, CLP0, and CLMD, CLMS, CLP4, CLM3, CLM2, CLM1, CLM0).
- 在进行校准之前，必须先正确配置ADC，并且必须产生正反偏校准结果，将结果存储到ADC正增益寄存器PG。

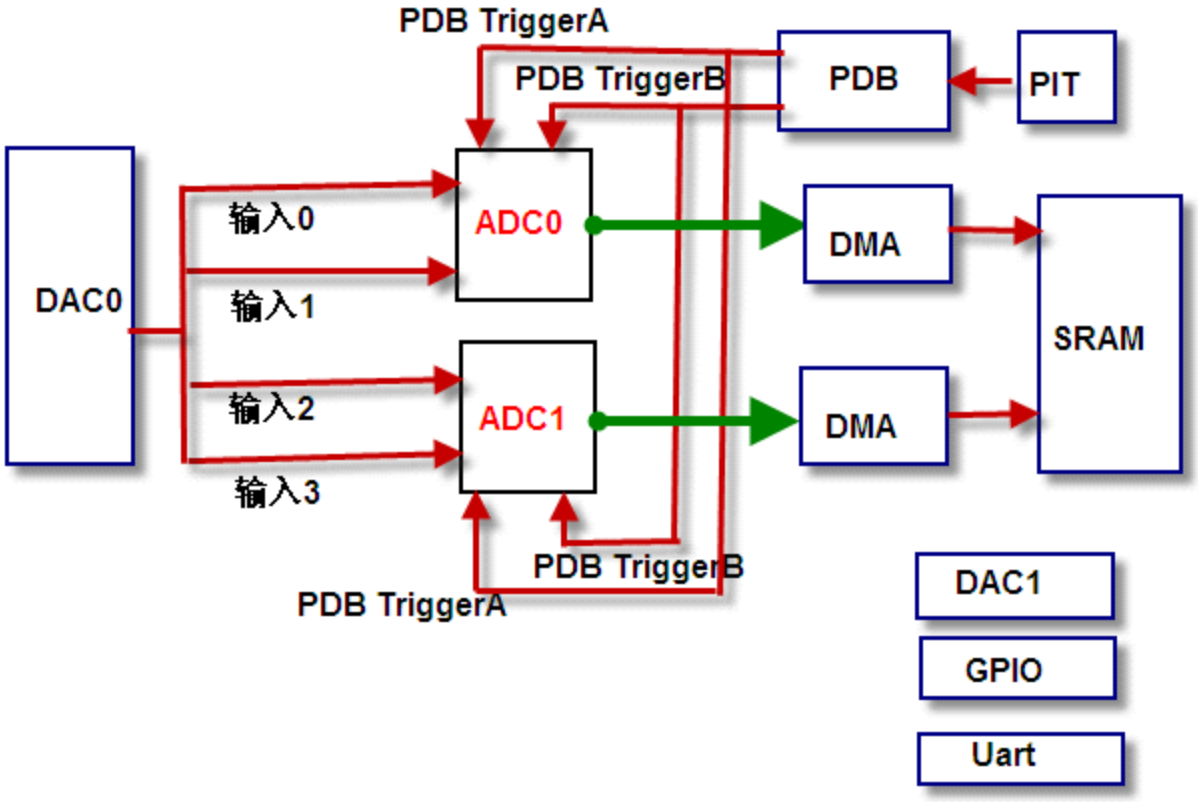
ADC应用设计

方案1： 在100Mhz K60 上采用两个ADC模块完成4路ADC输入采集，涉及下列模块

1. PIT：产生定长的触发脉冲，作为PDB输入源
2. PDB：通过两个时钟延迟，产生2路触发脉冲，用于两个ADC模块转换
3. DMA：用于ADC的转换结果传输
4. DAC： 1路作为ADC的输入信号源，另一路用于输出ADC的计算结果。输入还可换成恒压输入，范围0~3.3V.
5. ADC：完成校准、数据采集、DMA请求
6. 串口、IO：用于输出结果和指示状态
7. 每个通道的数据源信号频率50Hz,每周波采集1024点。在两个ADC模块上同时采集，采集一个点需要的时长为1/50/1024/4*2=9.7us.（注：在K60 120\150Mhz的芯片上有4个ADC模块，分布在四个ADC模块同时采集，采集一个点需19.5us）。

系统功能框图

方案1功能结构图：



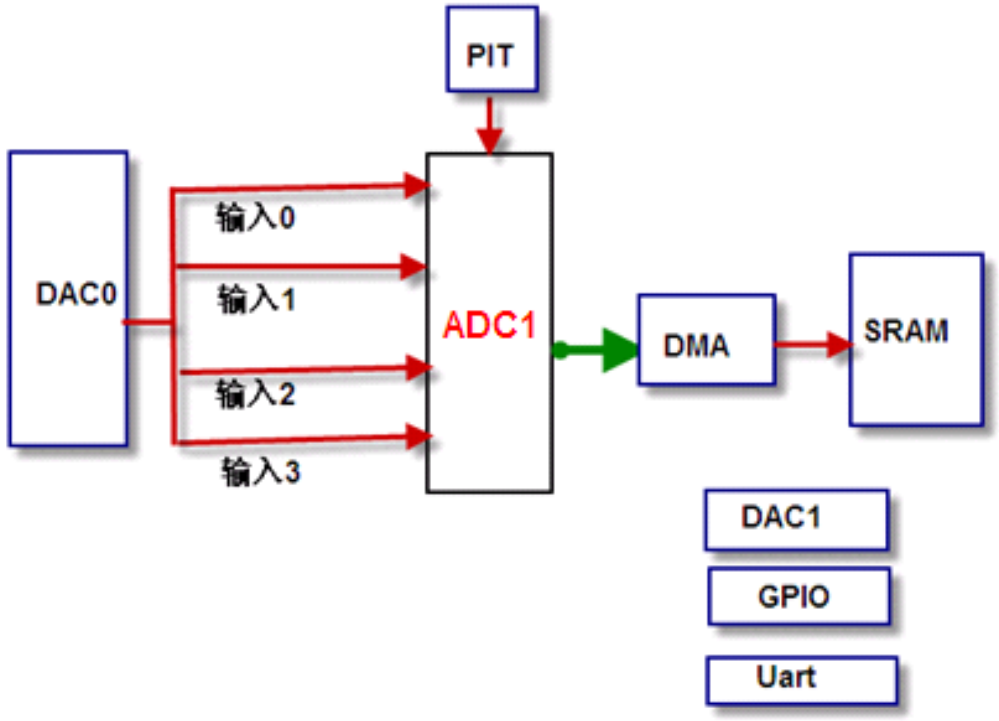
ADC应用设计

方案2： 采用一个ADC模块完成4路以上ADC输入的采集，涉及下列模块

1. PIT：产生定长的触发脉冲，直接触发ADC
2. DMA：用于ADC结果传输，并做通道切换
3. DAC：1路作为ADC的输入信号源，另一路用于输出ADC的计算结果。输入还可换成恒压输入，范围0~3.3V
4. ADC：完成校准、数据采集、DMA请求
5. 串口、IO：用于输出结果和指示状态
6. 每个通道的数据源信号频率50Hz,每周波采集1024点。在一个ADC模块上4路同时采集，采集一个点需要的时长为1/50/1024/4=4.8us。与方案1 的对比可看出，ADC的采样周期缩短一半，与ADC的最高转换速率接近。

功能框图

方案2功能结构图：



硬件配置

使用100Mhz MK60DN512 Tower板作为实验用的硬件， primary塔式侧面板作为电压信号输入端口。A和B端口上分布了MCU的管脚连接，下面的A13、B10指的是塔式侧面板接口。

方案1：GPIO、Uart管脚配置

DAC0、DAC1管脚配置：DAC0->A32　DAC1->B32
ADC0 输入管脚配置: PTB0(ADC0 ch8)->A14
PTB1(ADC0 ch9)->A13
ADC1输入管脚配置： PTE1(ADC1 ch5)->B10
PTE2 (ADC1 ch6)->B7

方案2：GPIO、Uart管脚配置
DAC0、DAC1管脚配置：DAC0->A32　DAC1->B32
ADC1输入管脚配置：
PTE0(ADC1 ch4)->B22
PTE1(ADC1 ch5)->B10
PTE2 (ADC1 ch6)->B7
PTE3(ADC1 ch7)->B11

芯片资源配置

方案1：
1）PIT通道1定时10ms 一个周期，驱动DAC0输出一路50Hz的方波信号。
2）PIT通道0定时19.5us，驱动PDB产生两路触发，两路触发间隔9.7us，分别驱动ADC0和ADC1。
3）DMA通道0 传输ADC0 RA和ADC0 RB数据到内存； DMA通道1 传输ADC1 RA和ADC1 RB数据到内存；各个通道独立使用，每次DMA请求只传输一个数据。DMA各个通道分别采用2048 字节的buffer 作为数据存储区。

方案2：
1）PIT通道0定时10ms 一个周期，驱动DAC0输出一路50Hz的方波信号。
2）PIT通道1定时4.8us，驱动ADC1。
3）DMA通道1 传输ADC1 RA数据到内存，DMA通道0传输存放了通道号码的变量到ADC1_SC1寄存器， 通道1链接到通道0，用于触发AD通道的轮换。
还可以用DMA完成中断或者ADC完成中断来切换通道。本实验用两种方式实现通道切换。

注意：实验发现，用DMA 通道0传输ADC的采样通道，可能有时传不成功。若传输不成功，会造成ADC通道与采样值的错位问题。所以为了保证通道与数值的一致性，最好使用DMA或ADC中断程序来切换通道，不过这要加大CPU的运行负荷。

开发工具介绍 — 独立ProcessorExpert使用技巧

- **创建项目**

  项目类型：IAR、Keil、Codewarrior、GNU c

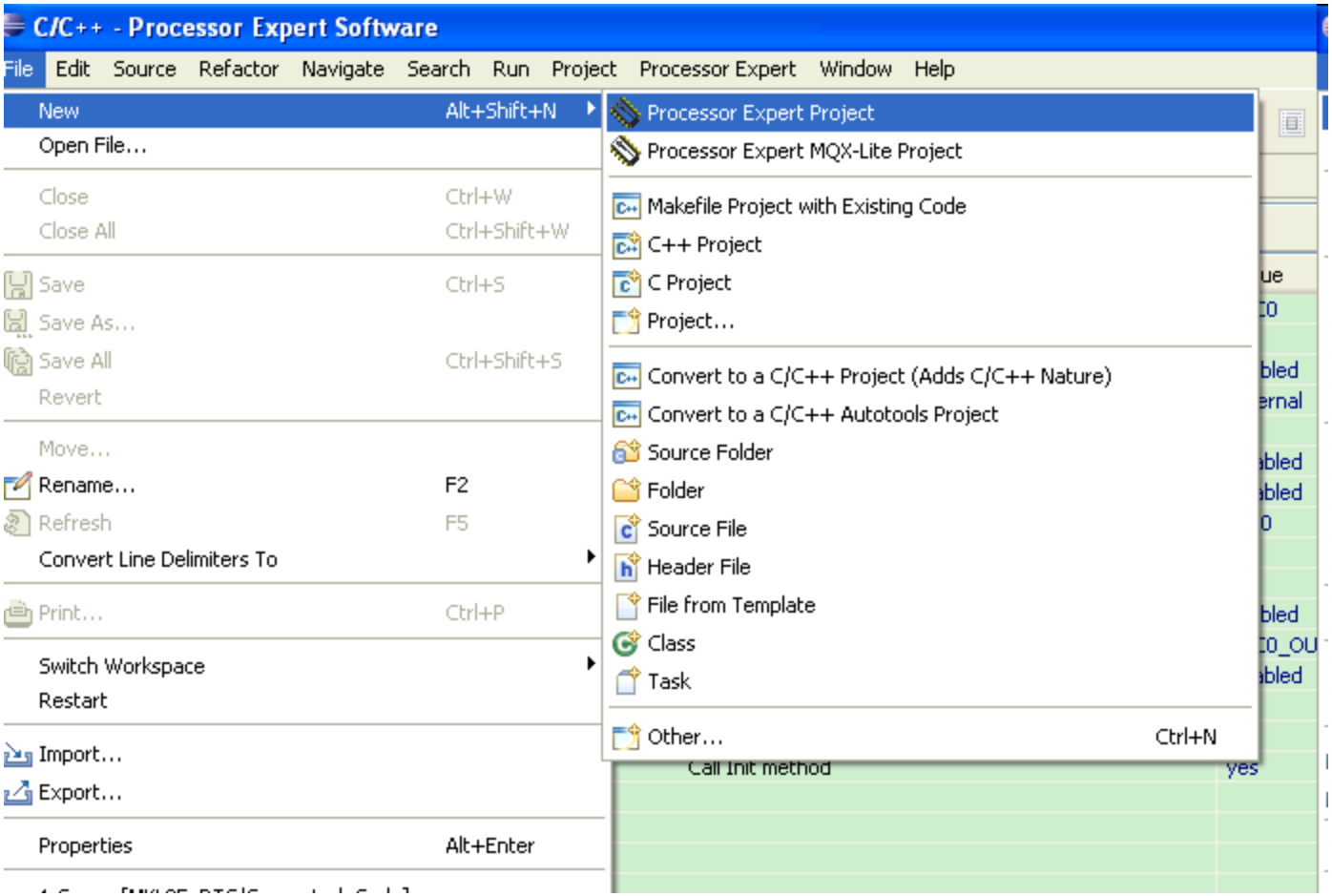
- **帮助文件**

  Bean文件的使用说明、例程

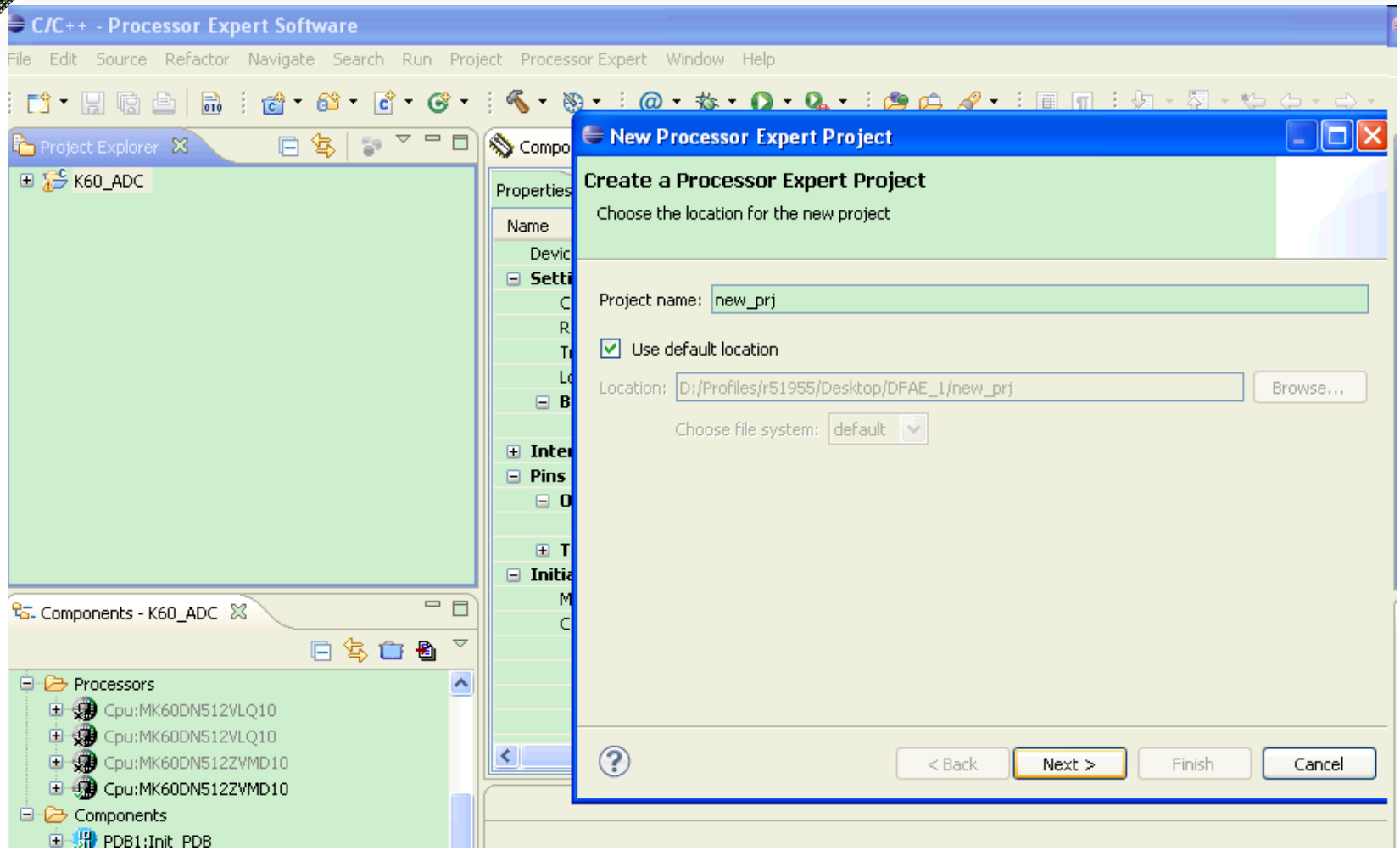
- **代码保存、移植、重用**
  - 工程的整体保存与输出
  - Bean文件的保存与输出

独立ProcessorExpert使用技巧（续）

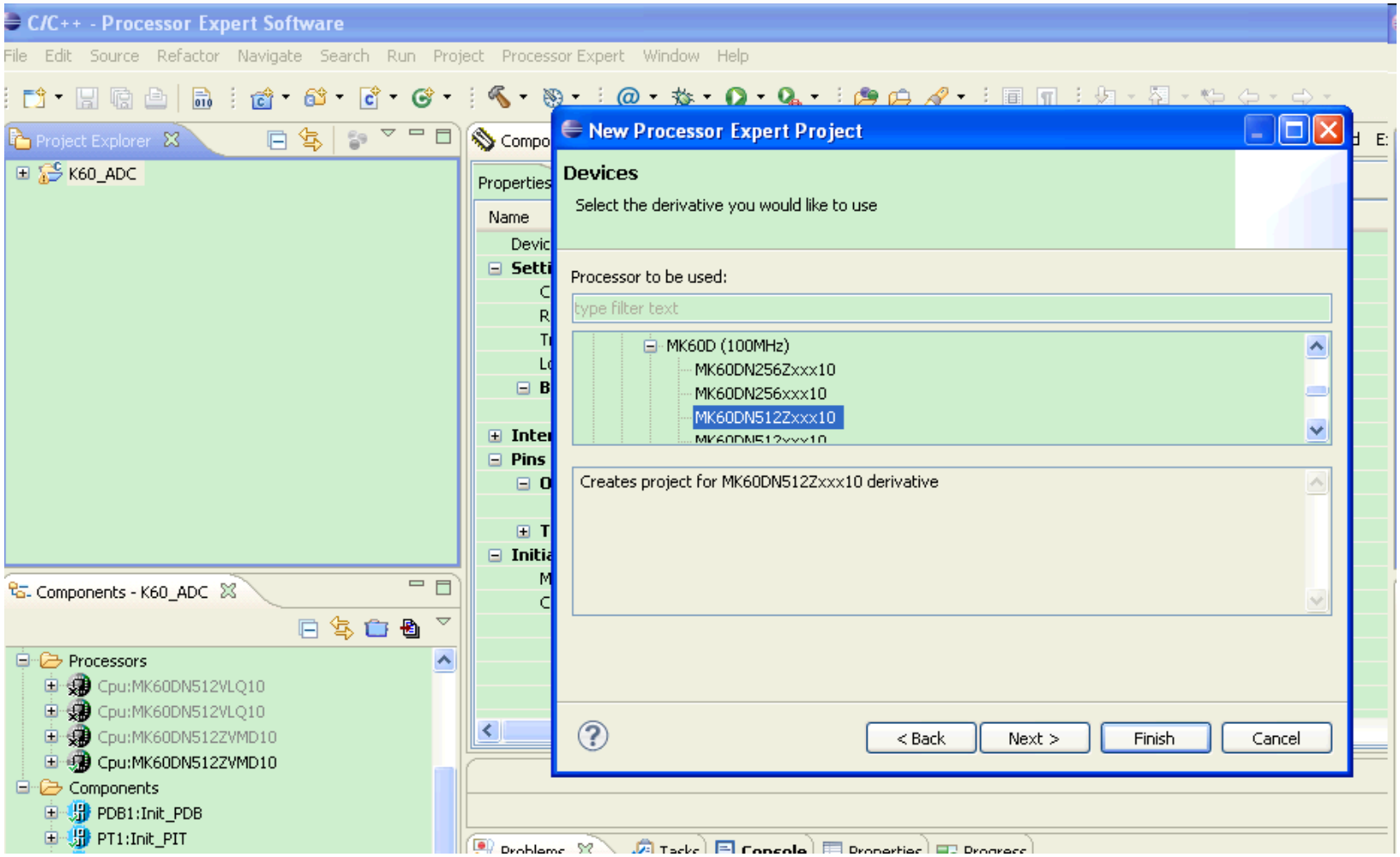
- 项目创建



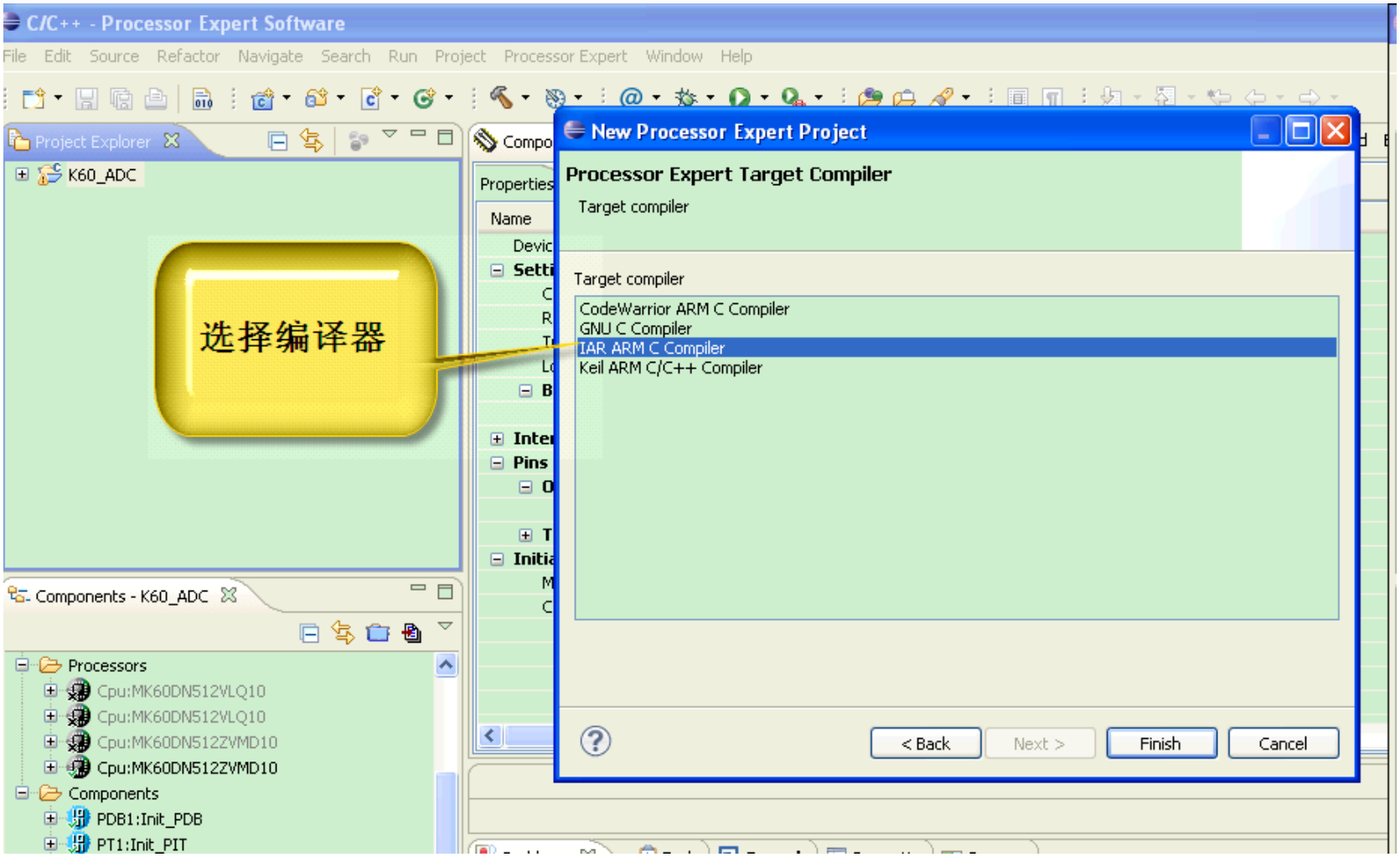
项目创建（续）



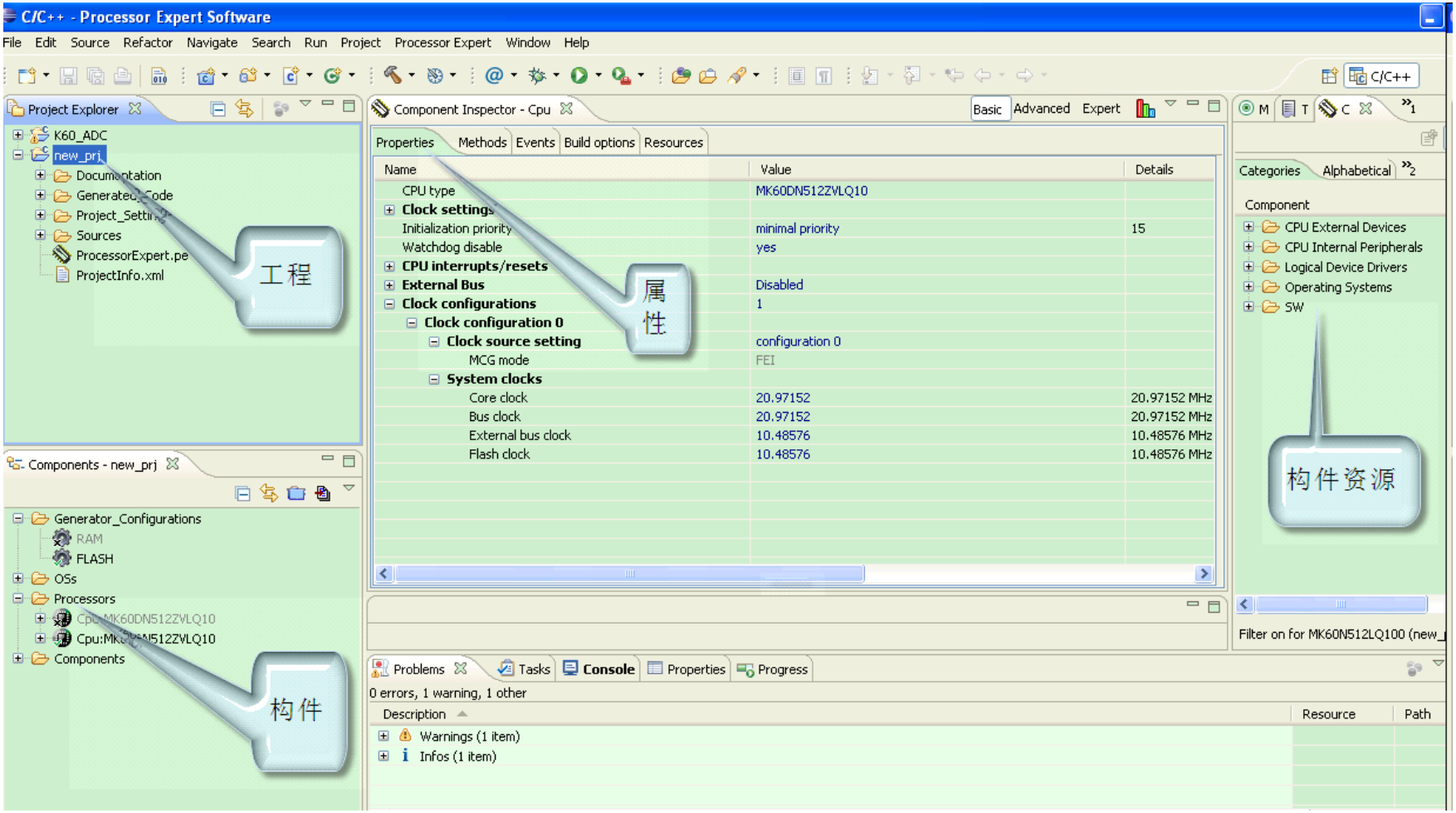
项目创建（续）



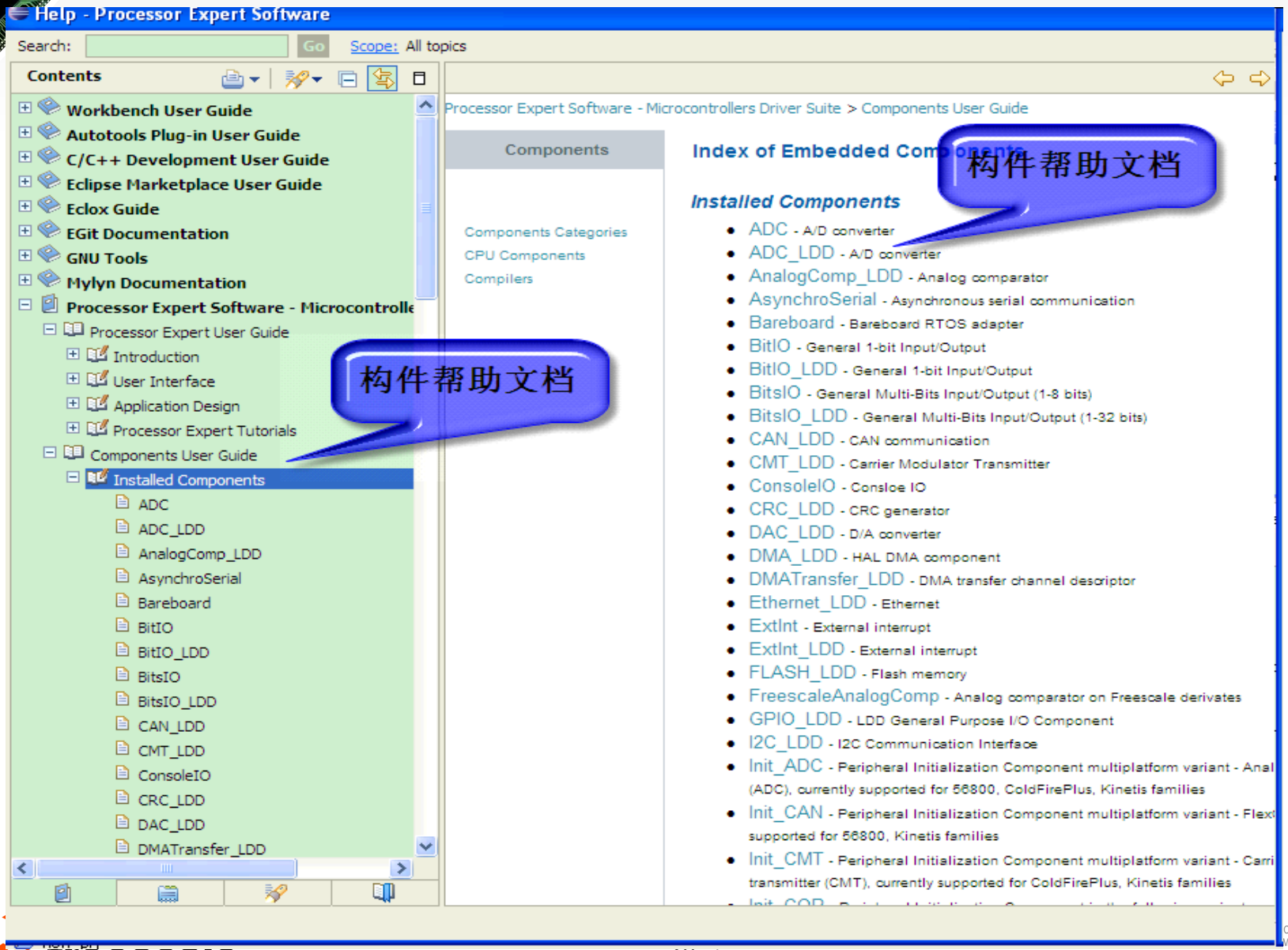
项目创建（续）



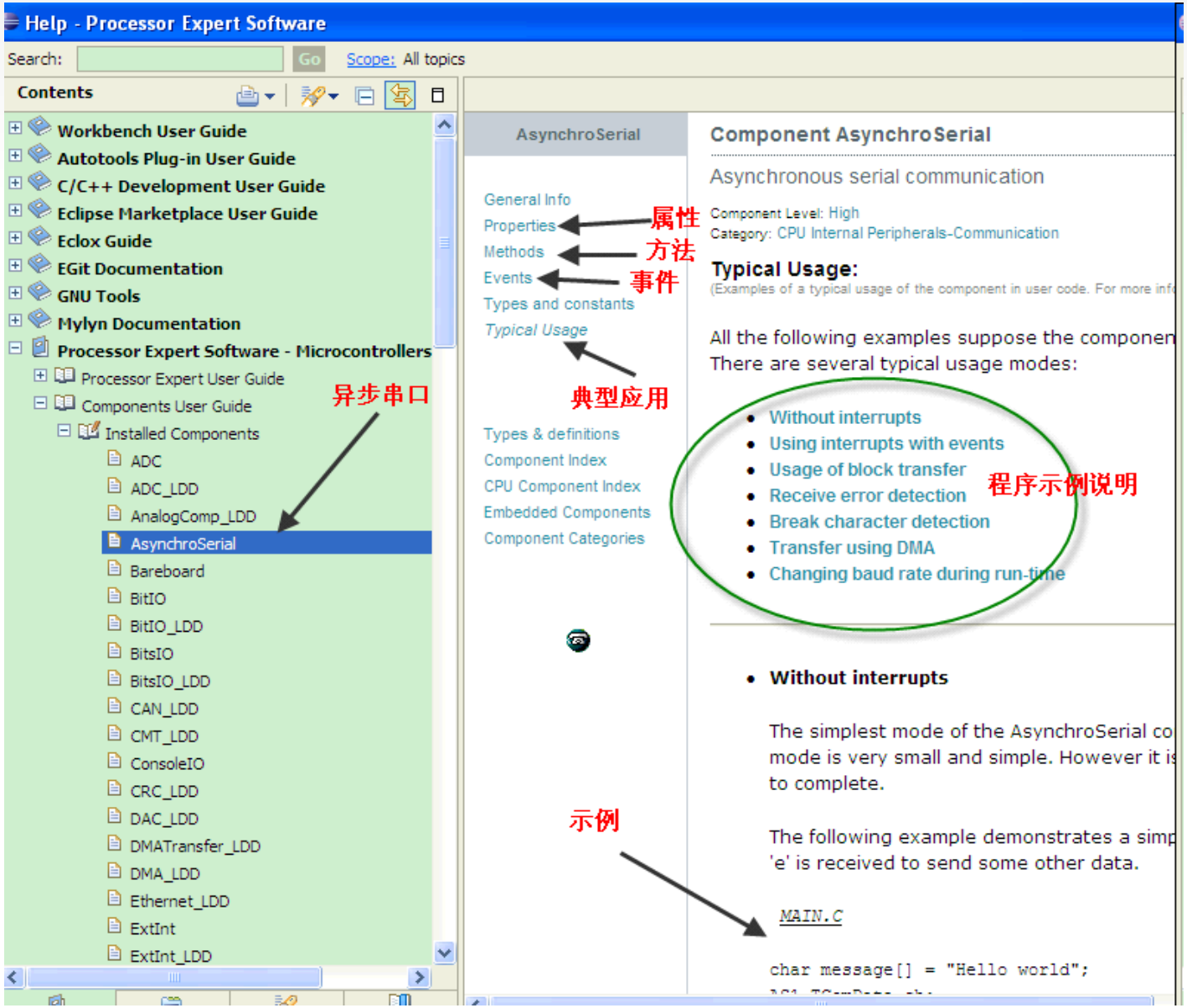
项目创建（续）



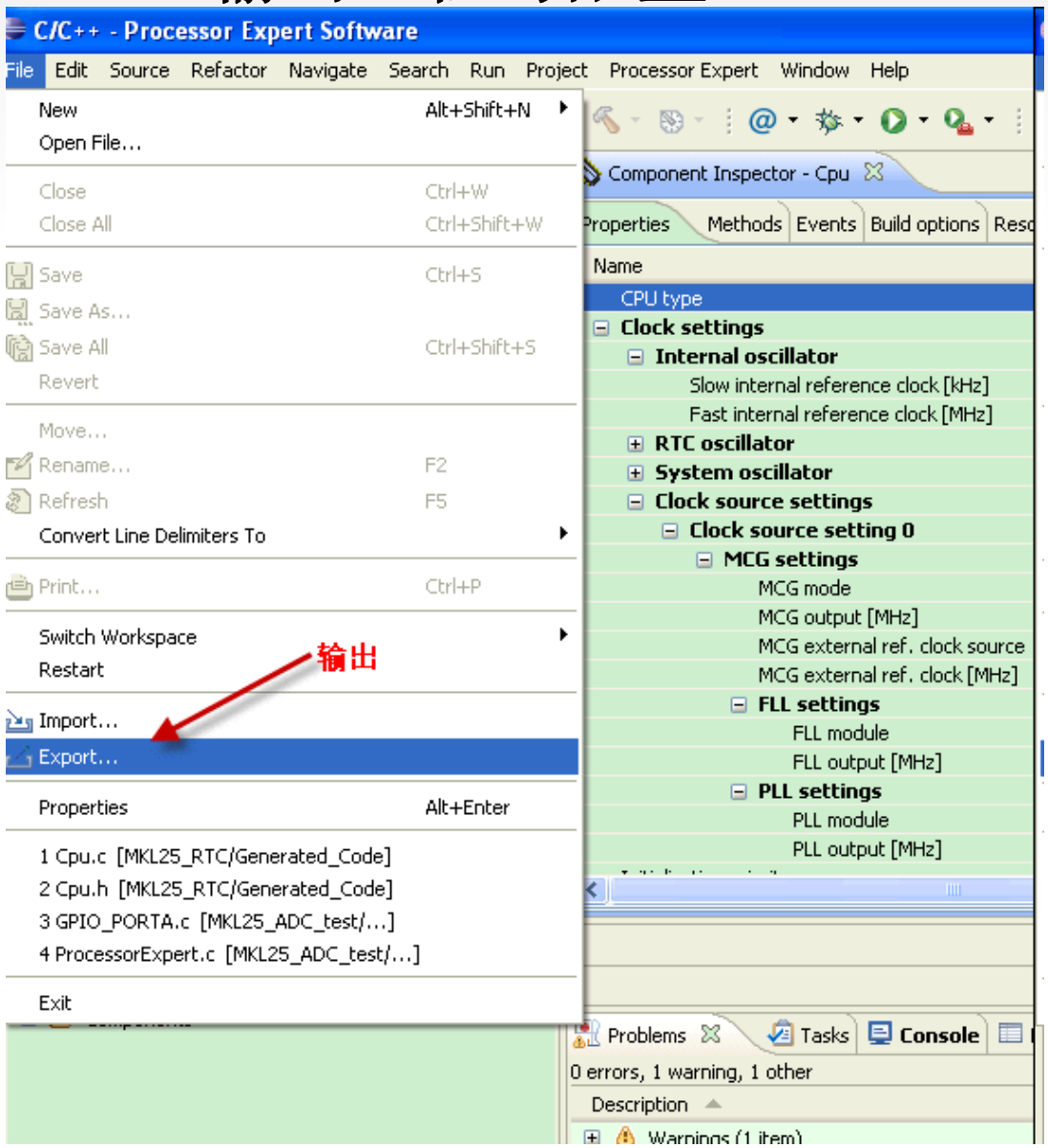
构件帮助文档



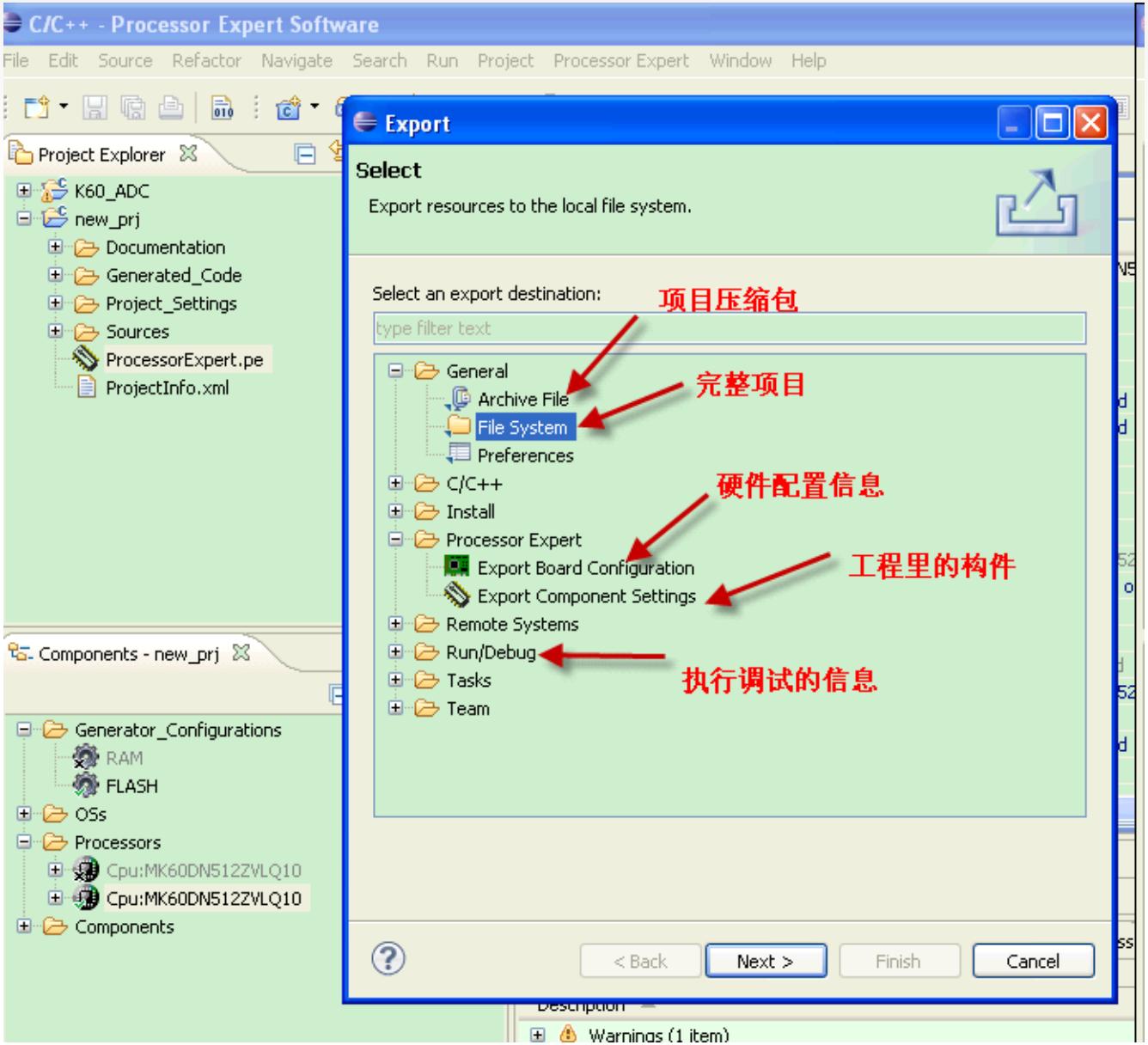
构件帮助文档（续）



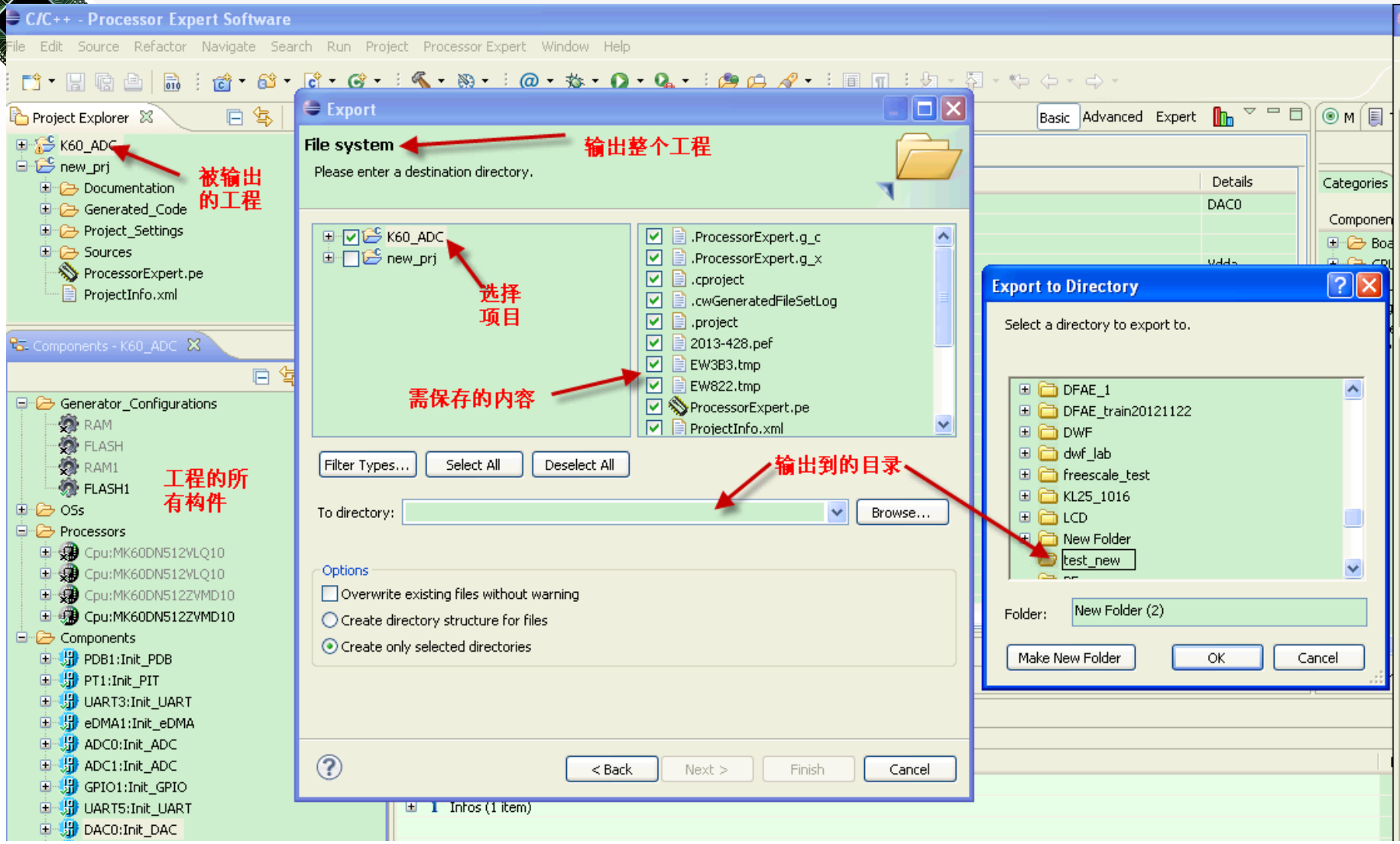
输出工程与配置



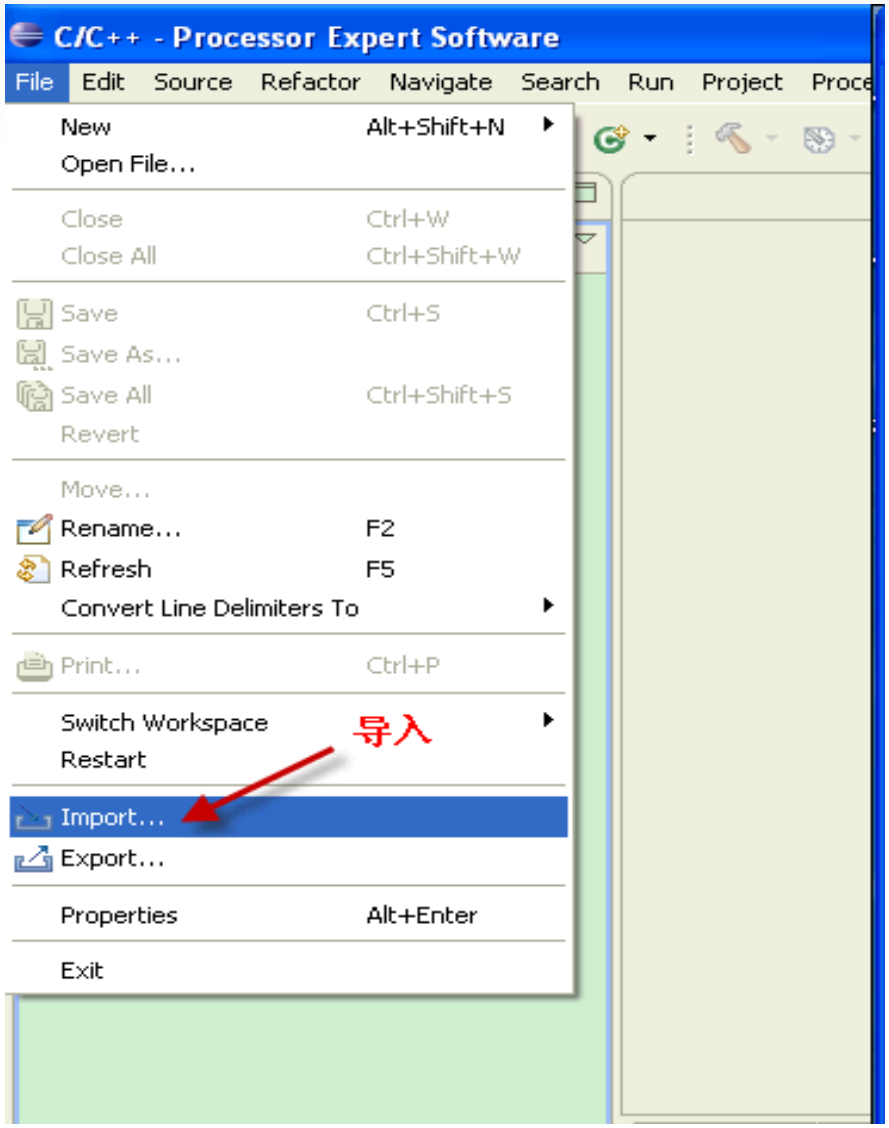
输出工程与配置



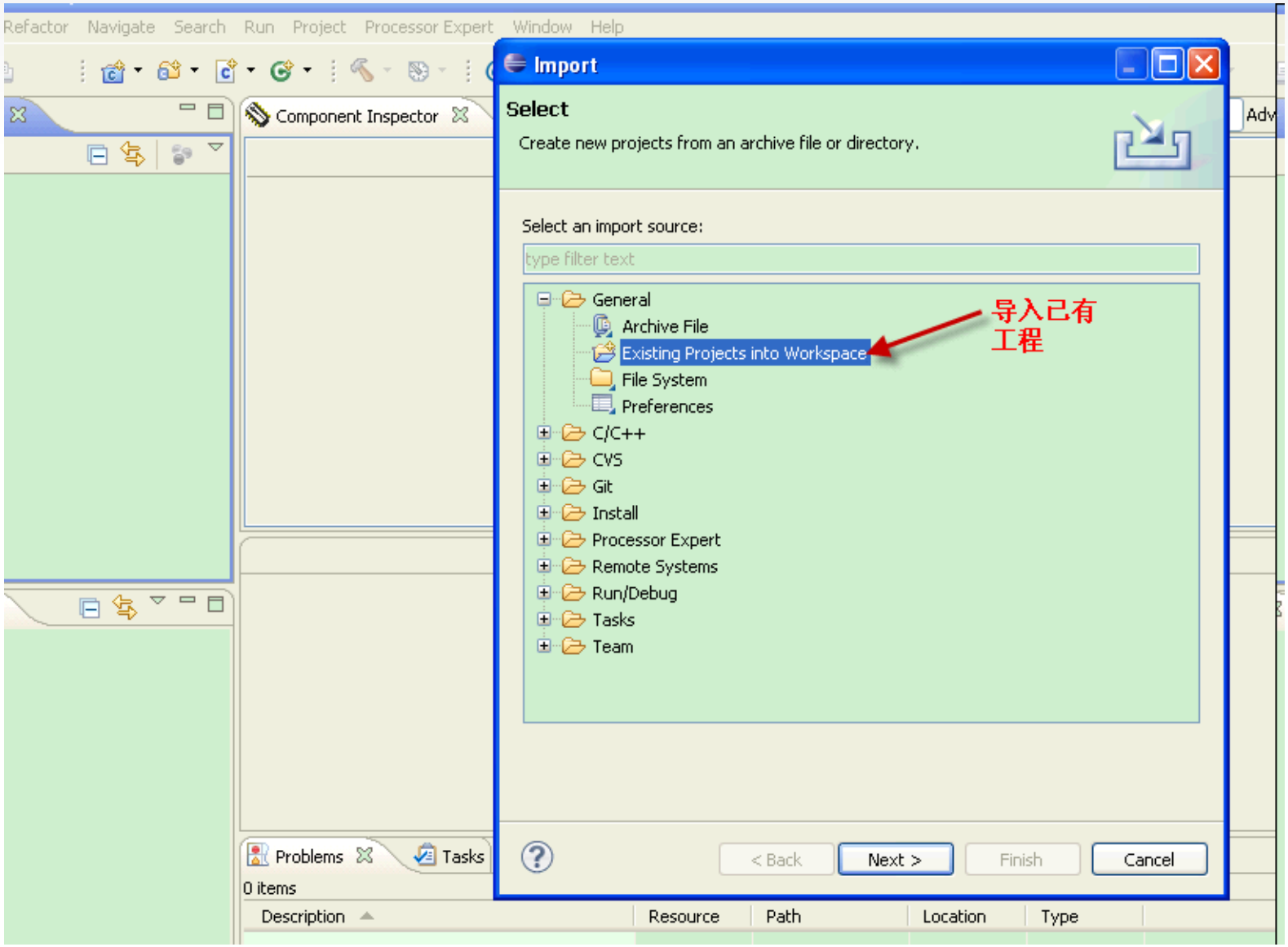
保存输出一个工程



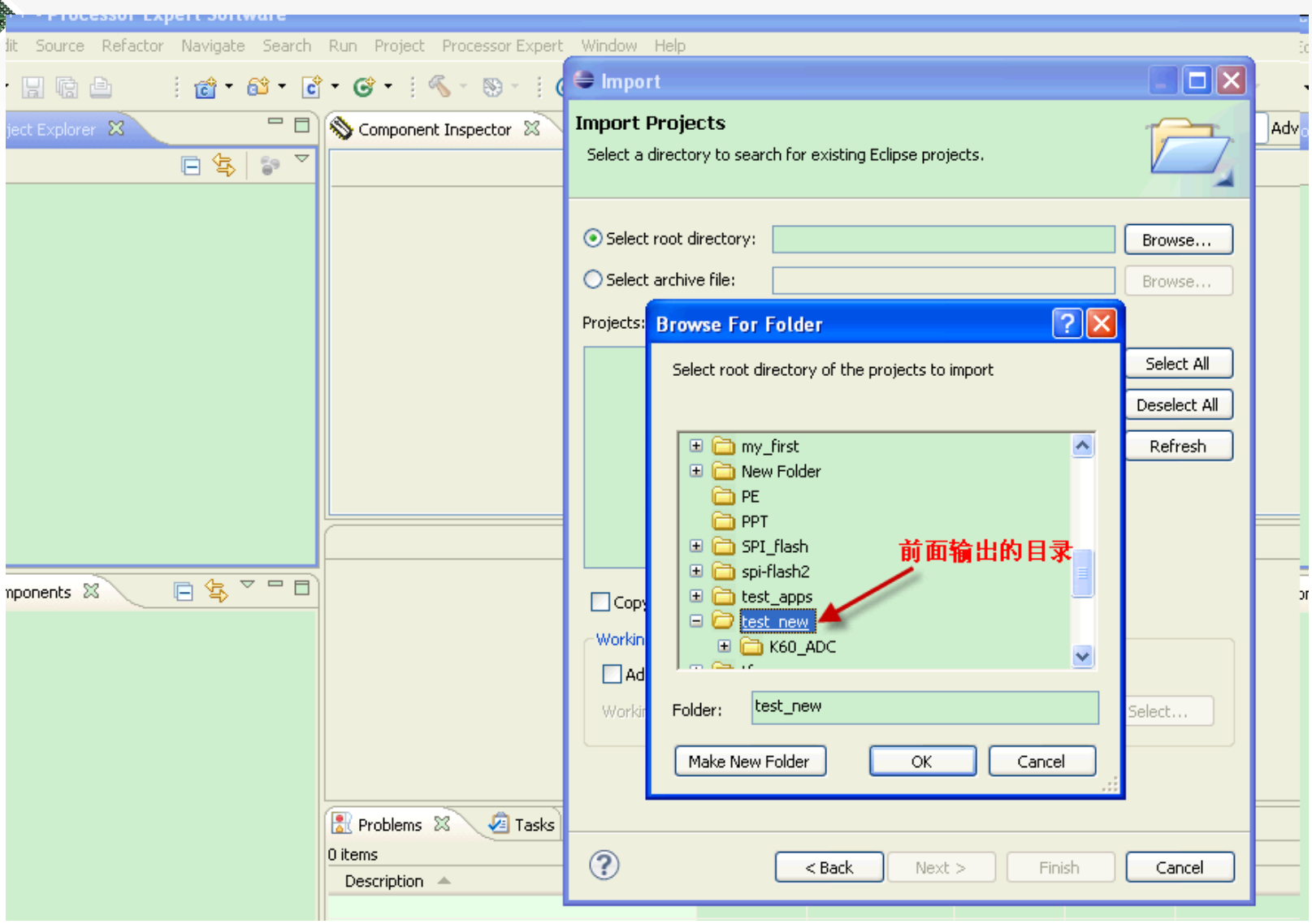
工程导入



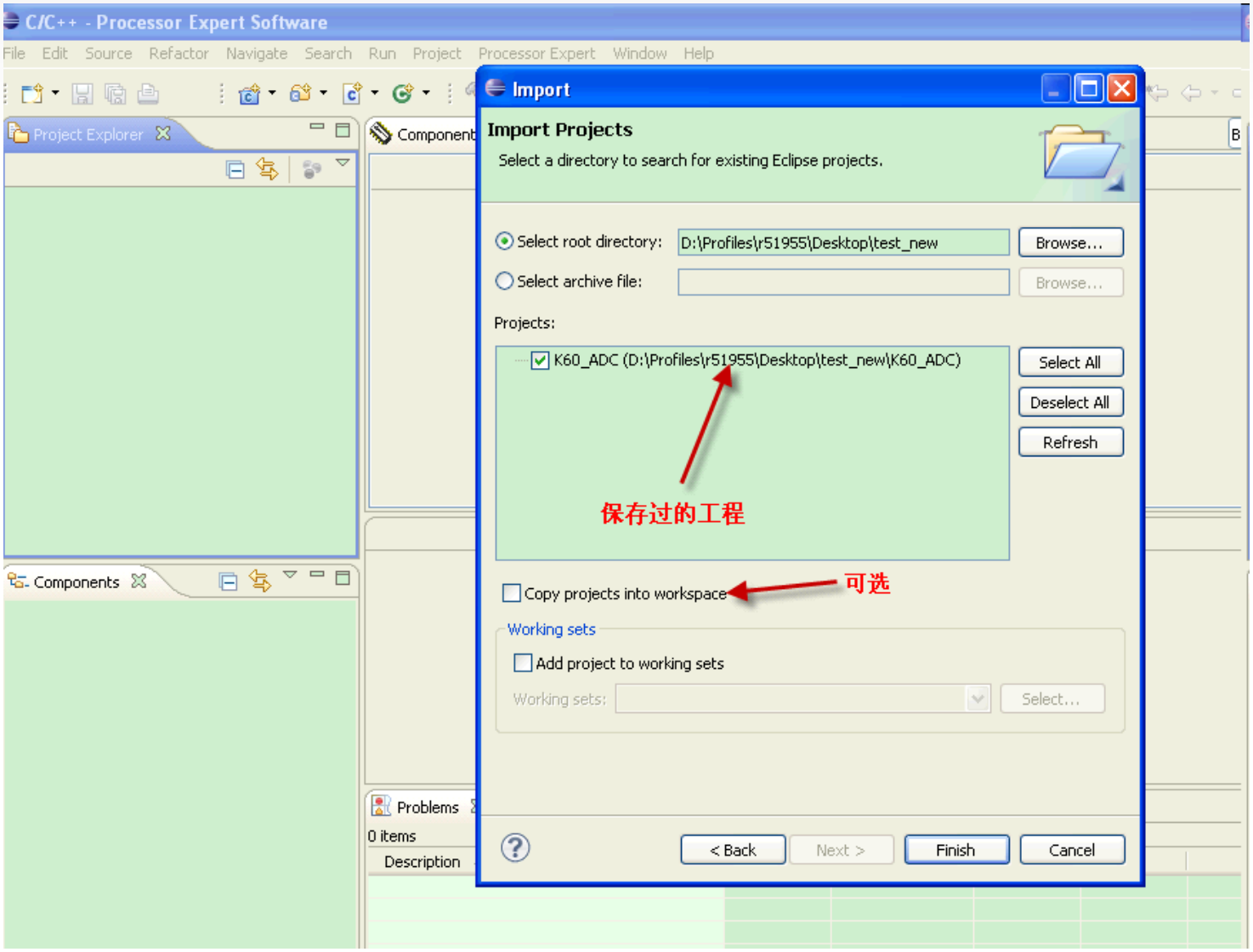
工程导入（续）



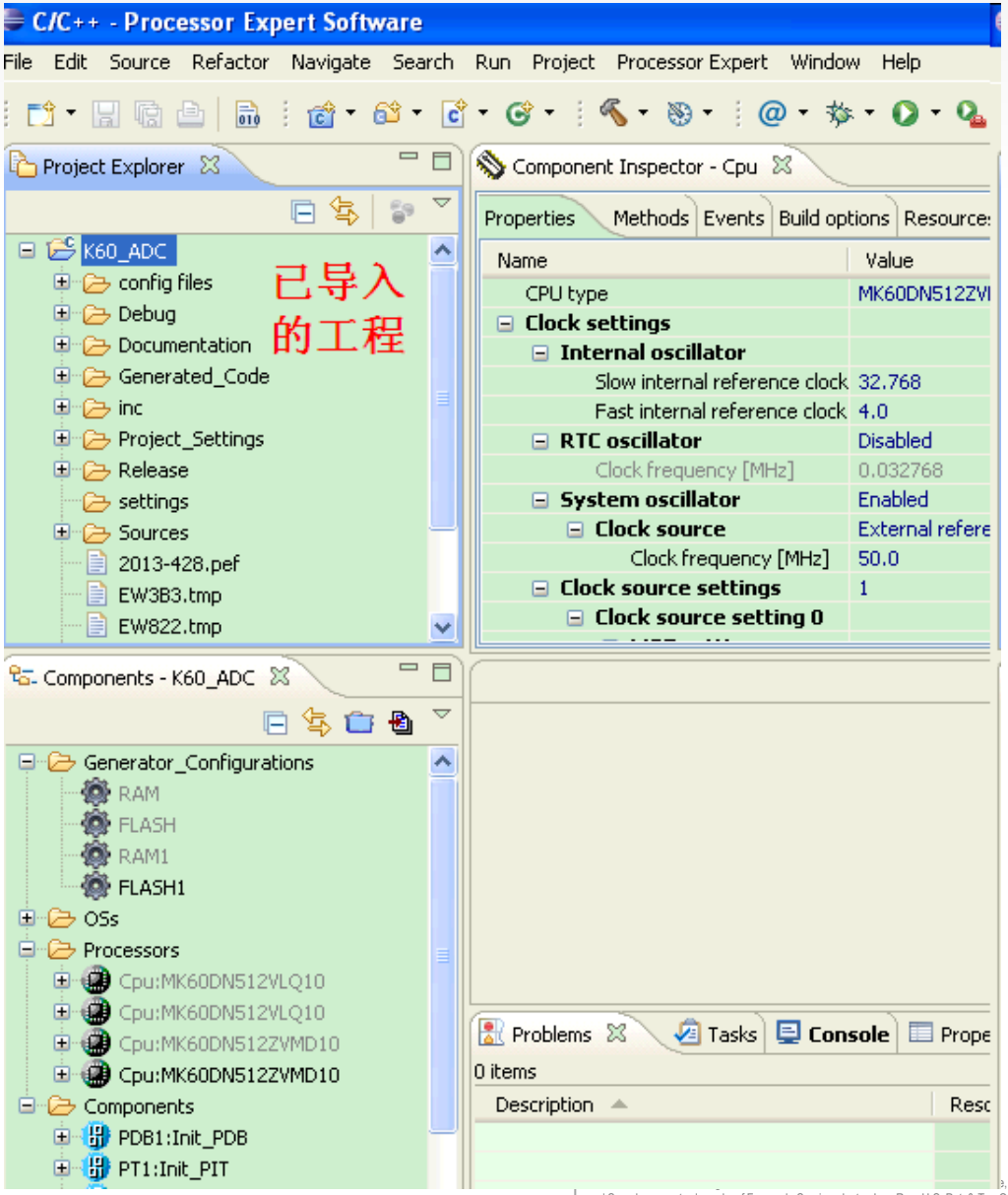
工程导入（续）



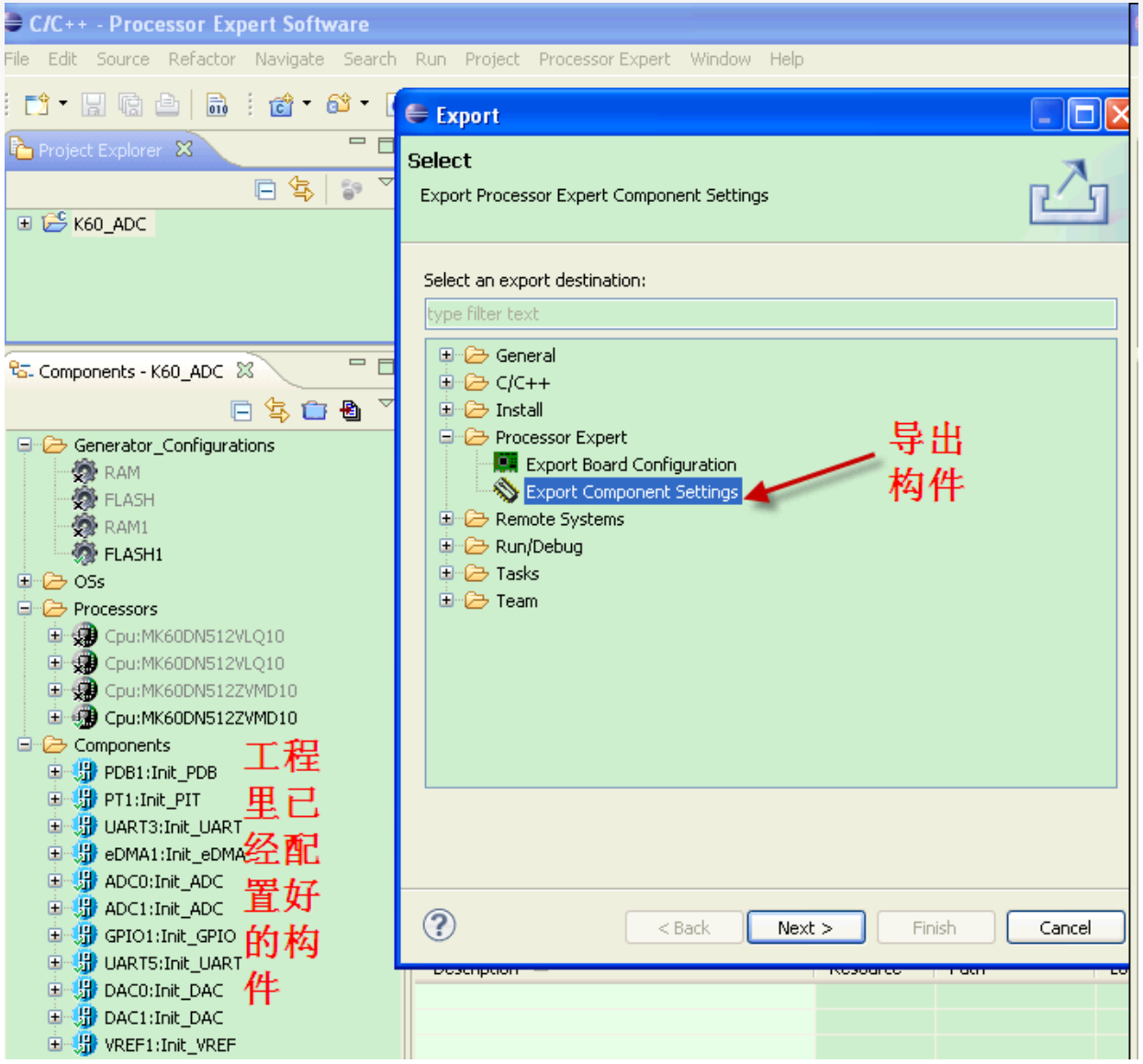
工程导入（续）



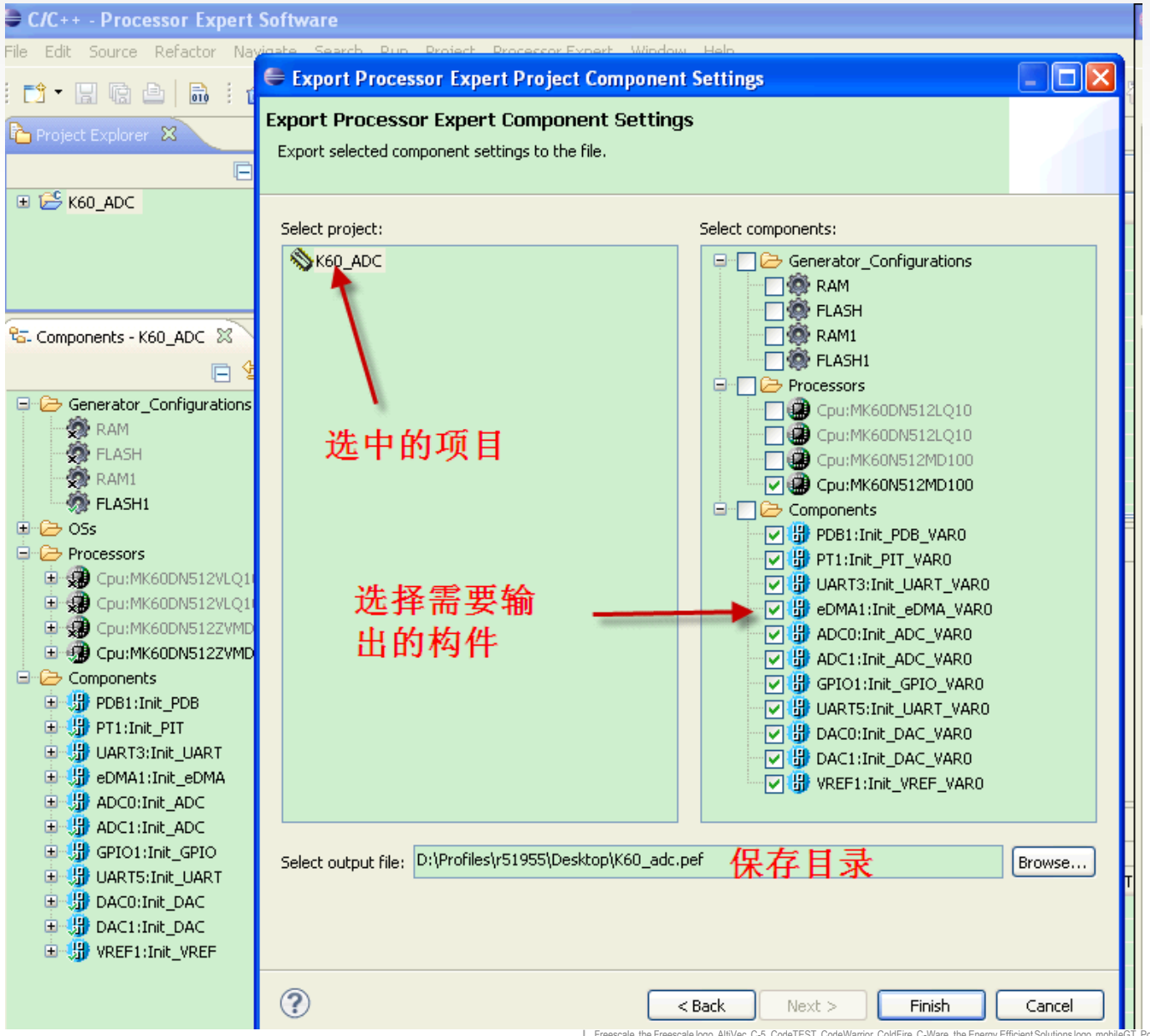
工程导入（续）



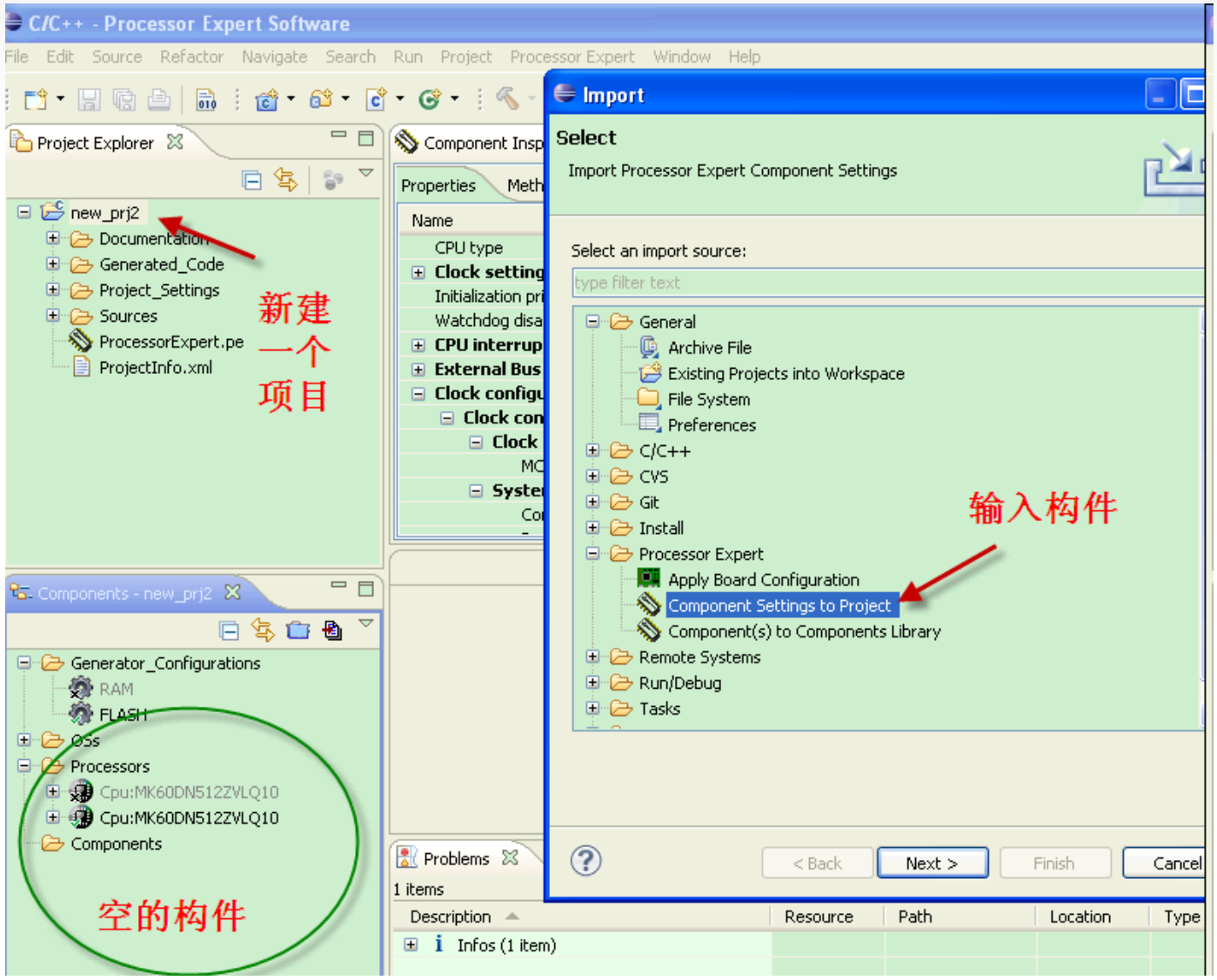
可视化构件的输出（续）



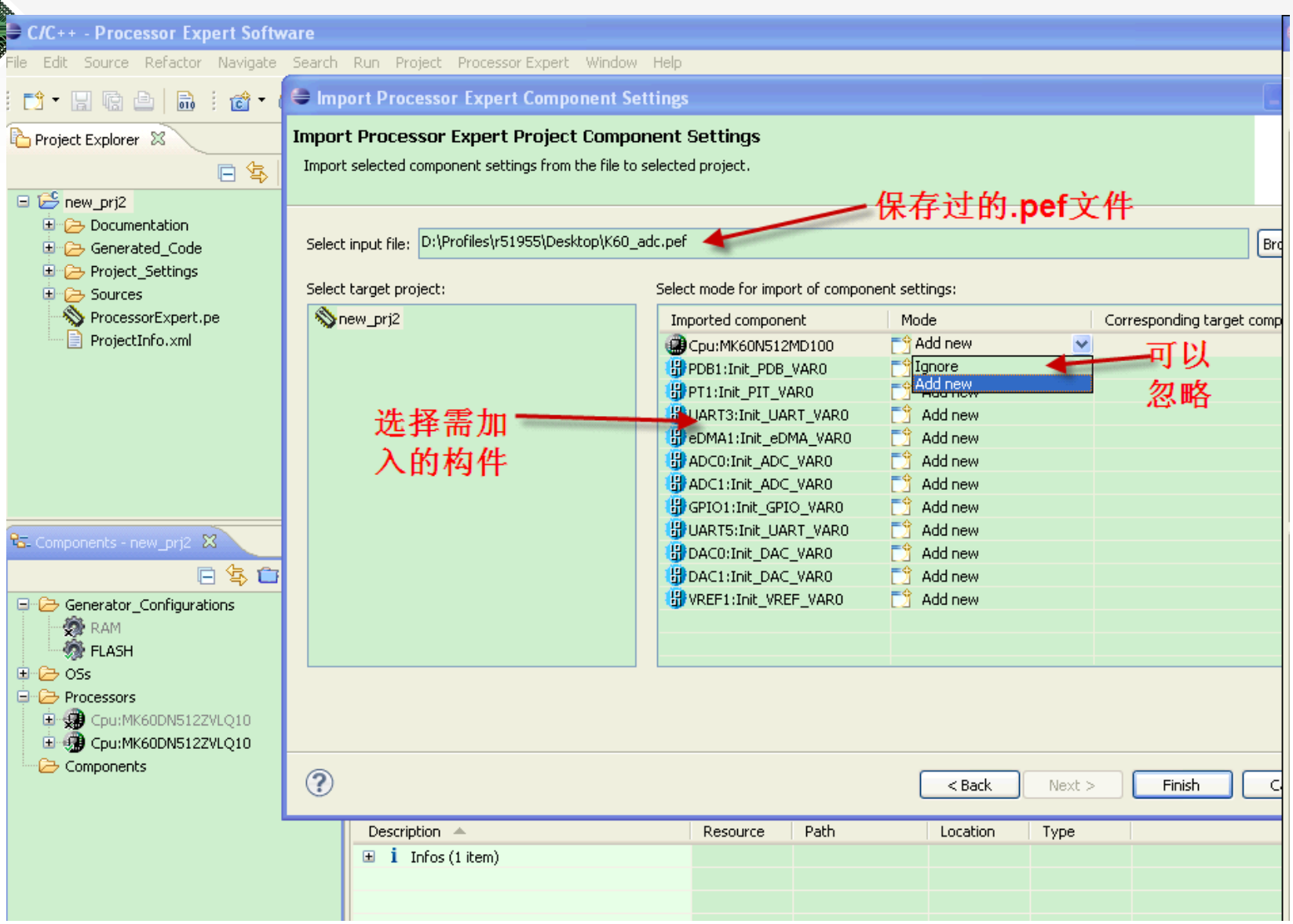
可视化构件的输出（续）



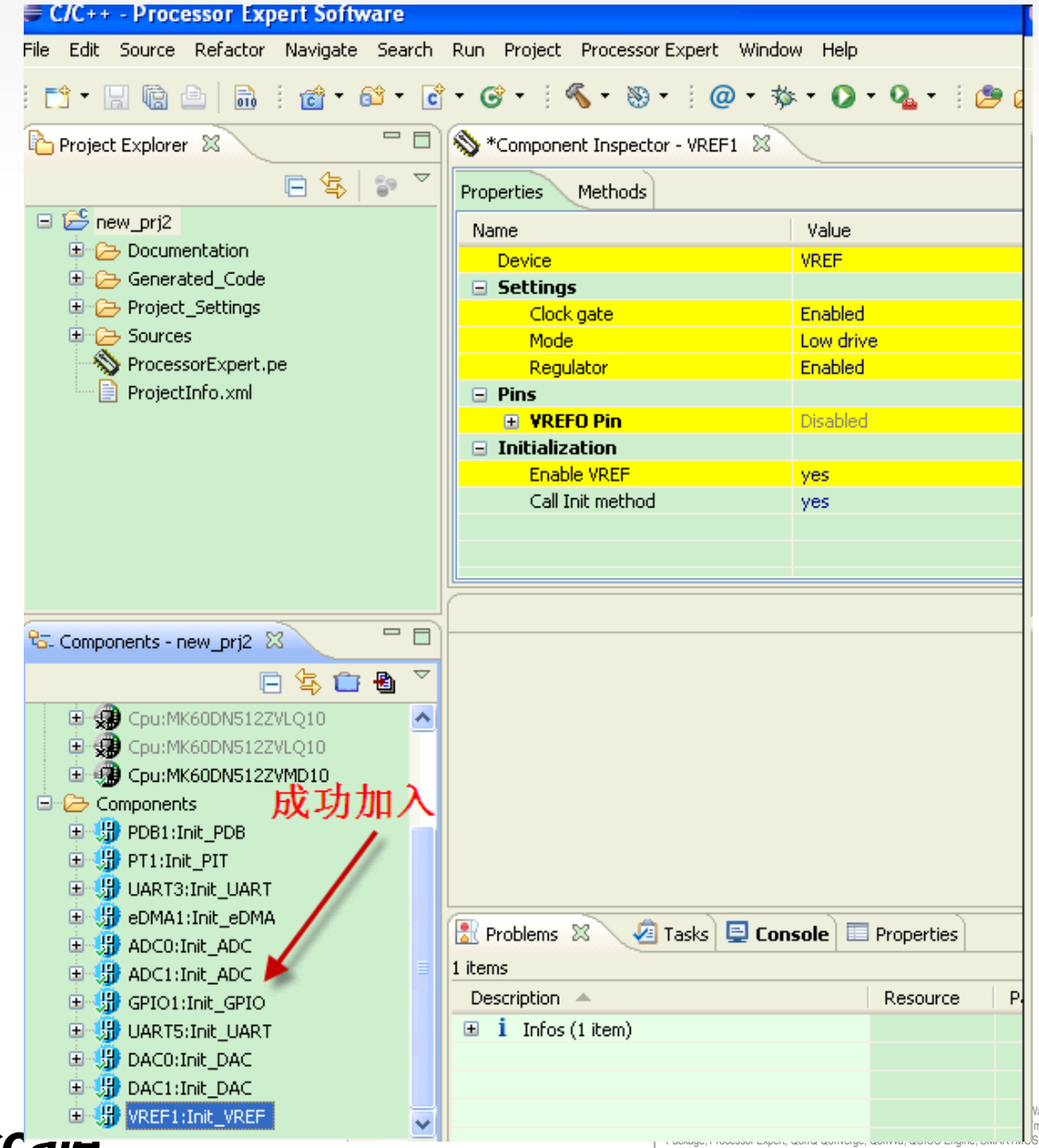
可视化构件的导入



可视化构件的导入（续）



可视化构件的导入（续）



freescale

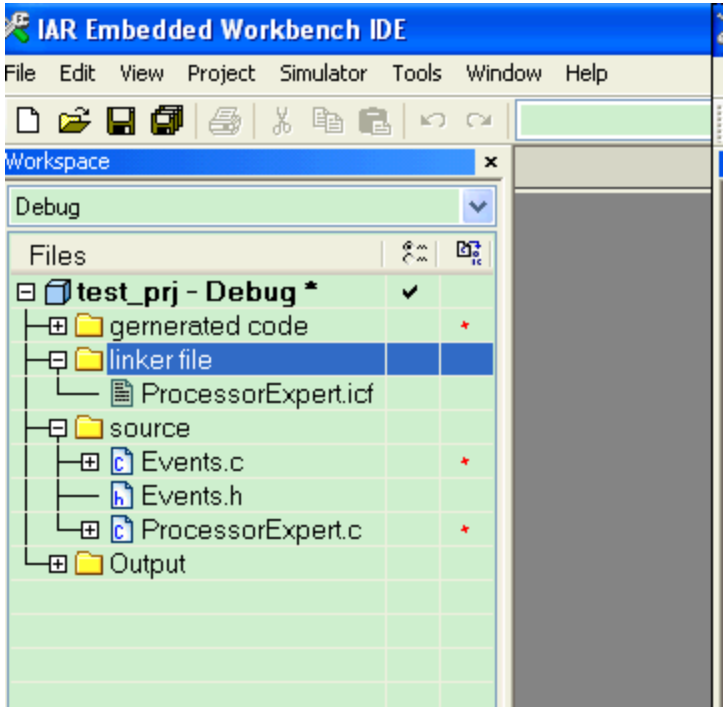
独立ProcessorExpert与IAR的结合

第一步、在PE下建立一个新工程，命名为new_prj2，选择IAR编译器。

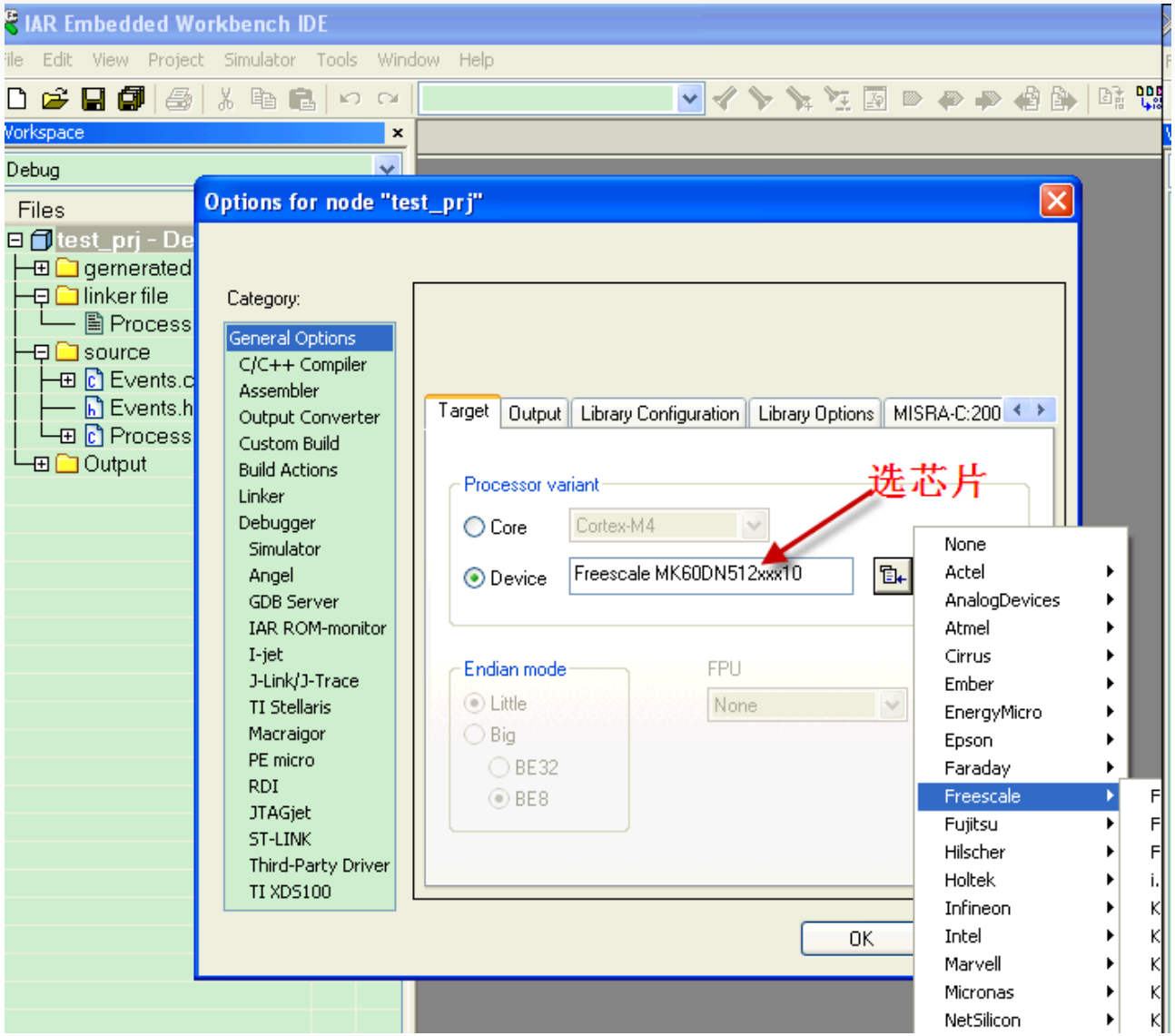
第二步、加入一些可视化构件，配置以后，生成代码。

第三步、打开IAR，在IAR下建立一个新的Workspace。然后再建立一个ARM的空工程，命名为test_prj.ewp，工程文件保存到PE生成代码的目录下，便于代码管理，如：..\new_prj2 目录。保存workspace 到同一目录下，名为为test_prj.eww。

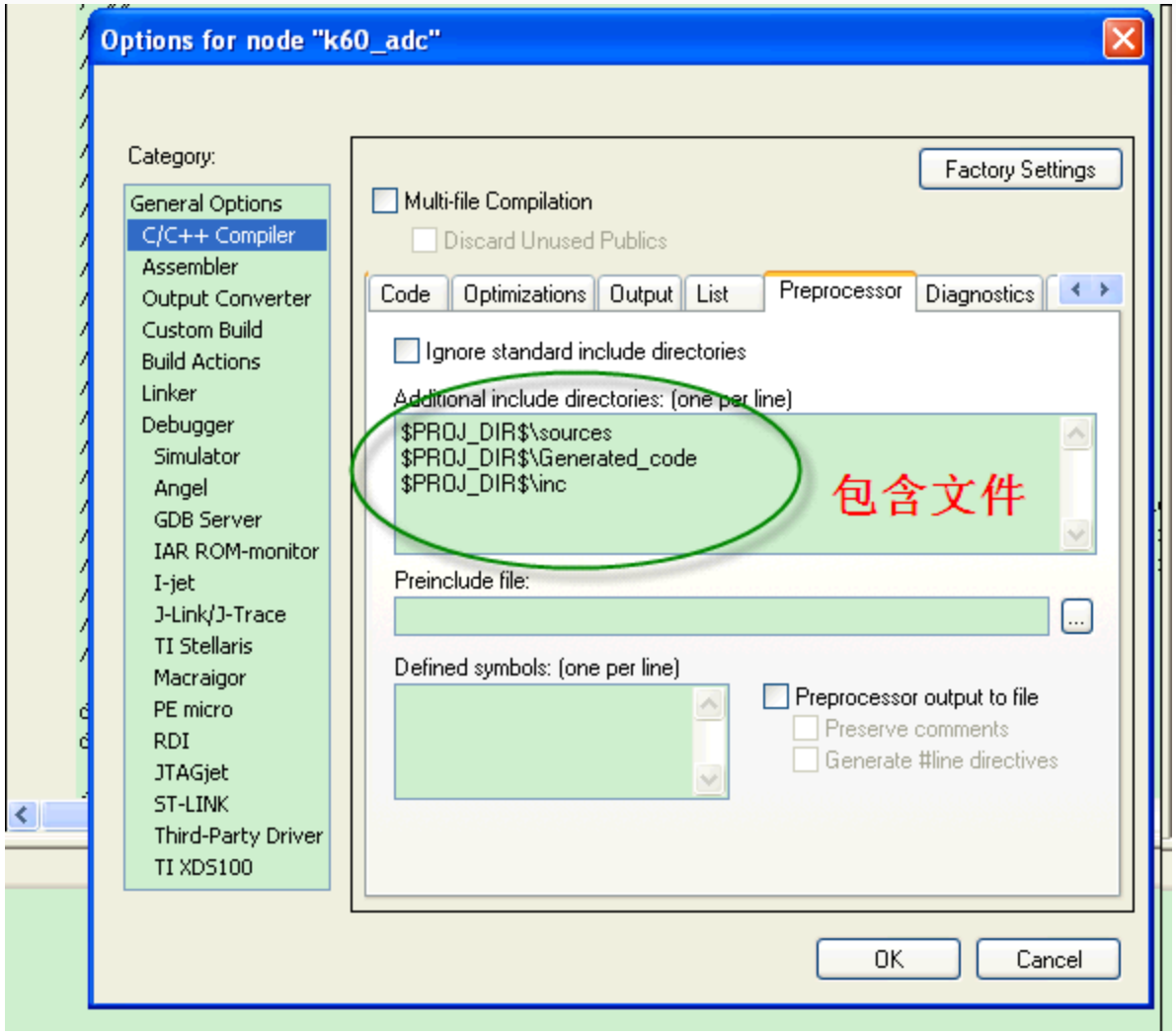
第四步、在新的项目里建立几个代码组，如：gernerated code 、source code、linker file 等。将PE生成的代码分别加入到各个代码组。



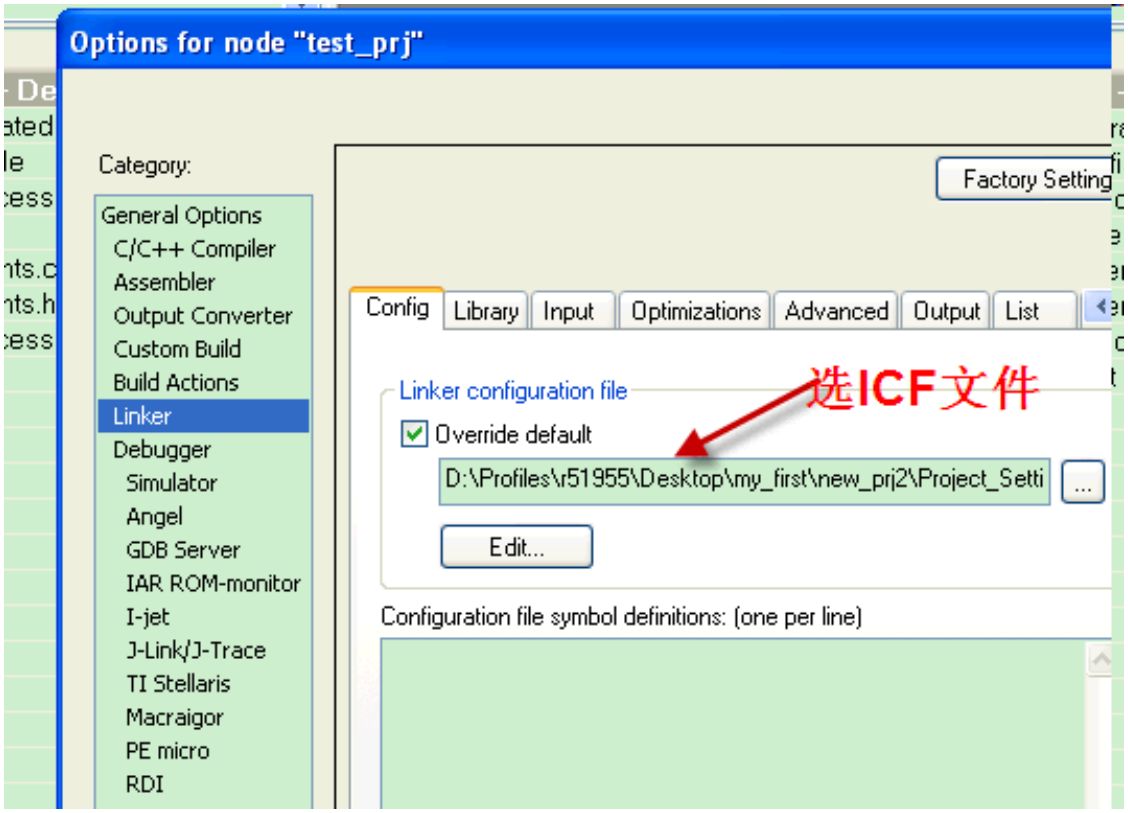
独立ProcessorExpert与IAR的结合（续）



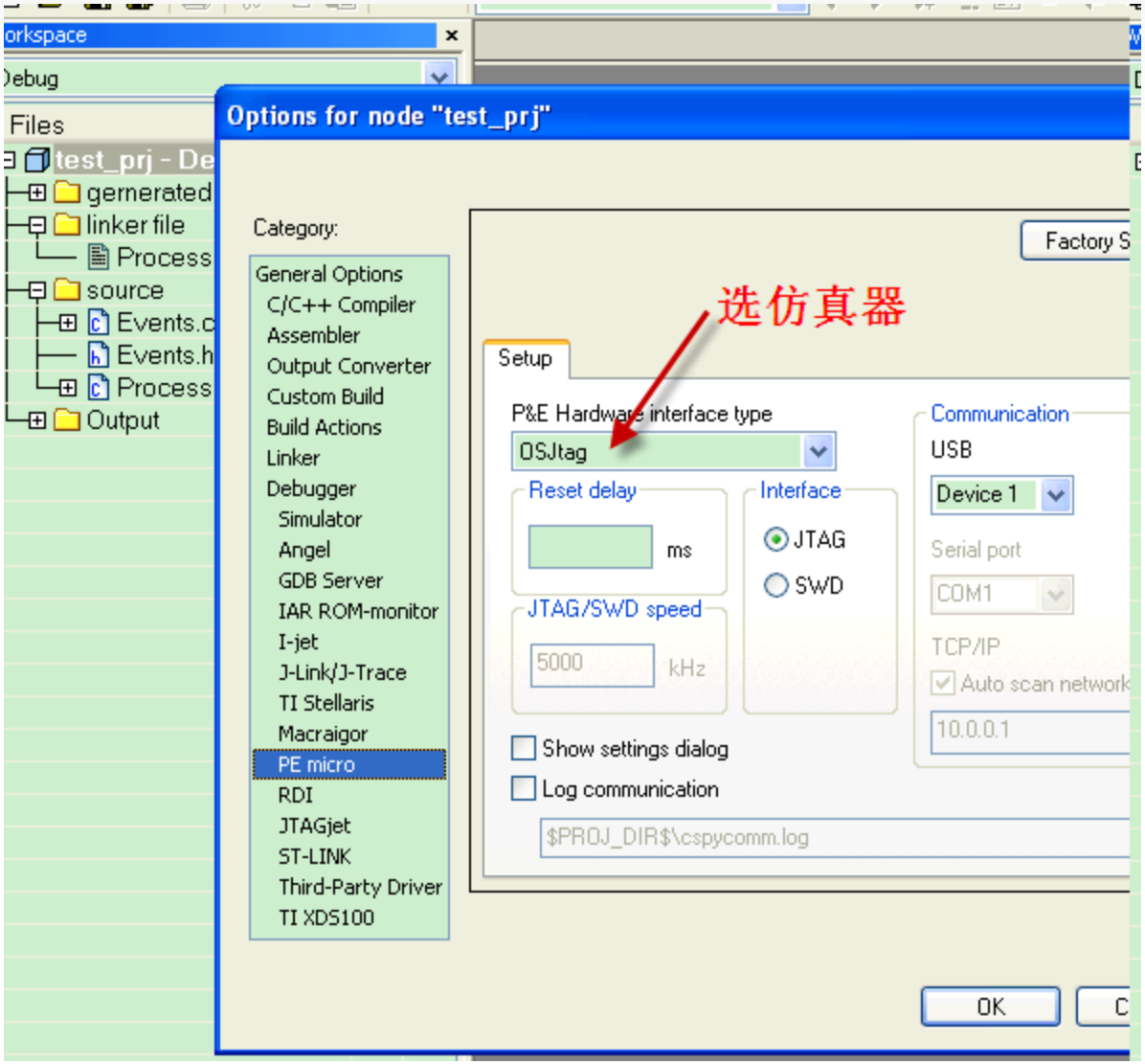
独立ProcessorExpert与IAR的结合（续）



独立ProcessorExpert与IAR的结合（续）



独立ProcessorExpert与IAR的结合（续）

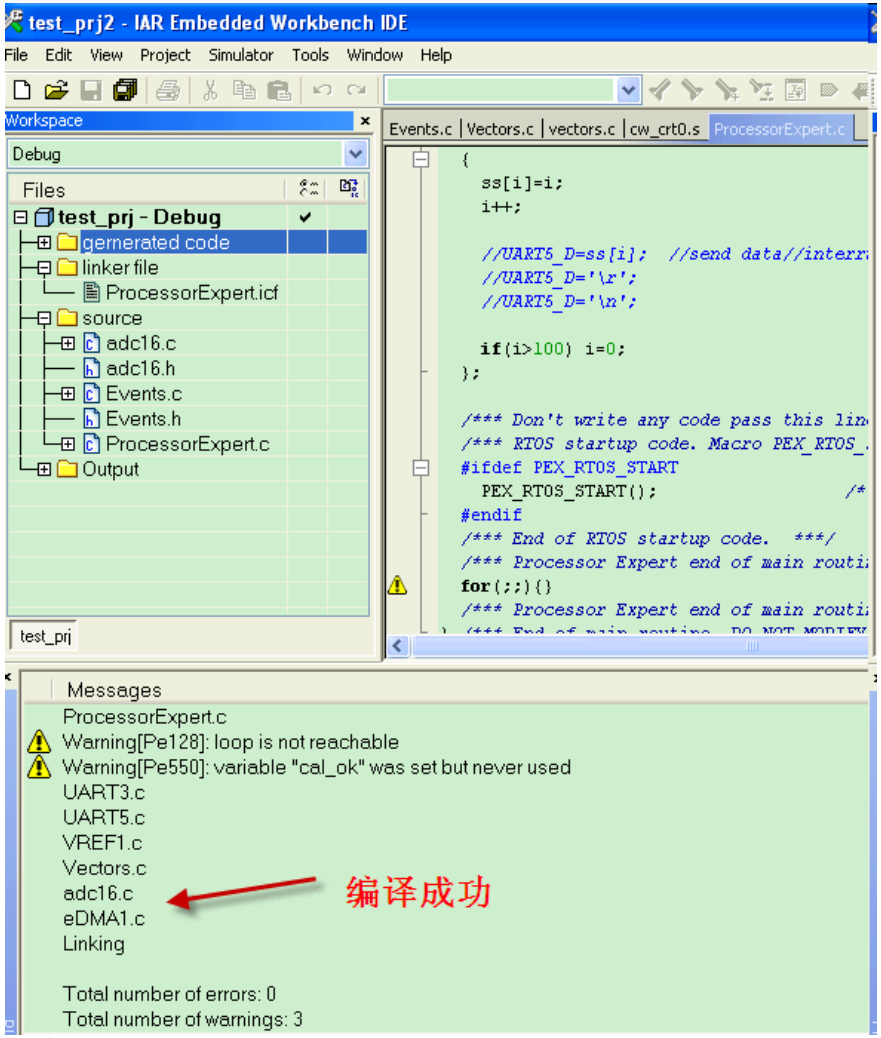


独立ProcessorExpert与IAR的结合（续）

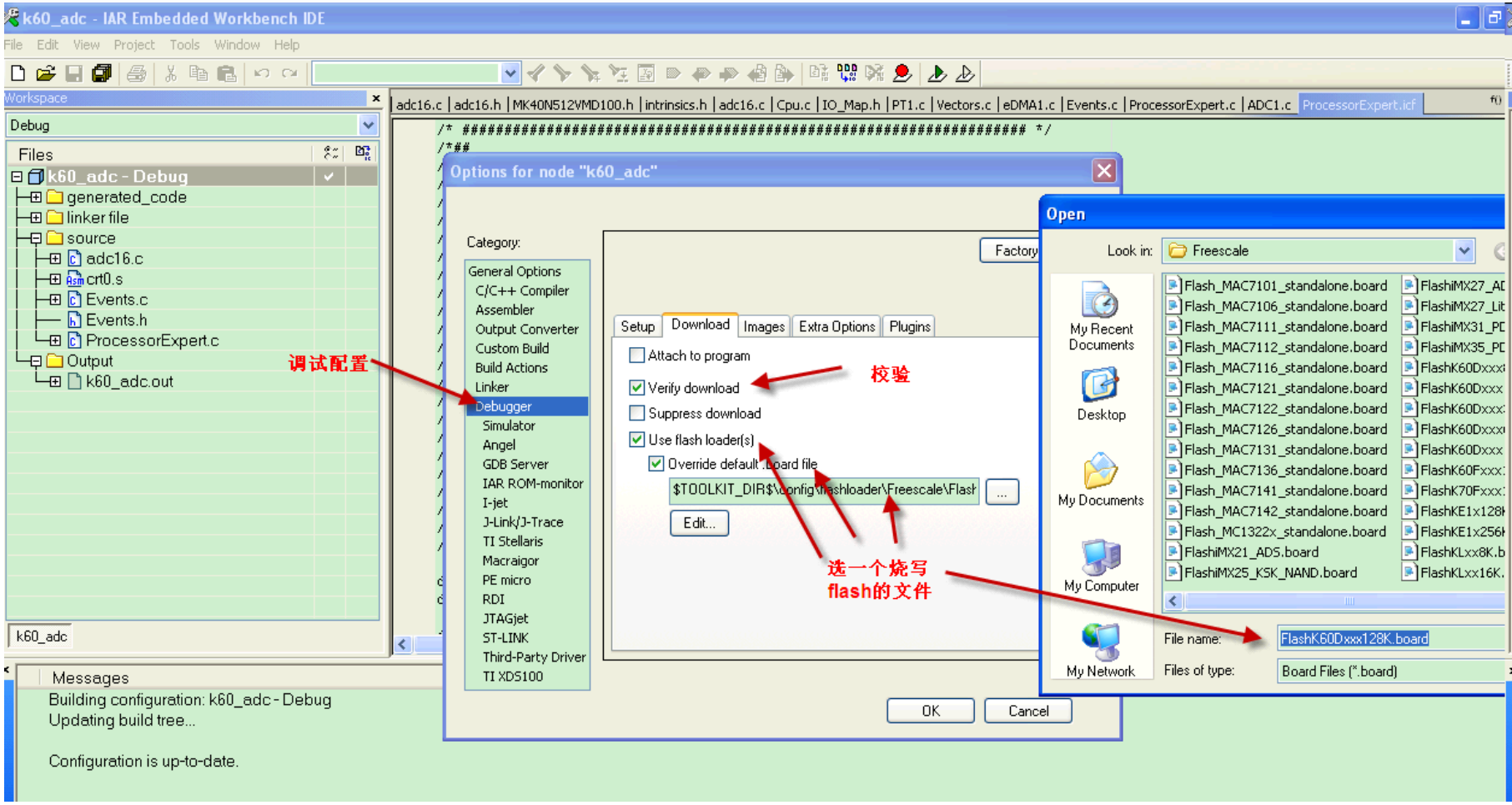
最后一步：把PE下的包含文件拷贝到当前工程下，以便工程文件的调用。

C:\PExDrv v10.0\eclipse\ProcessorExpert\lib\Kinetis\pdd\inc 下的文件拷到当前工程下，如：D:\Profiles\r51955\Desktop\my_first\new_prj2\inc

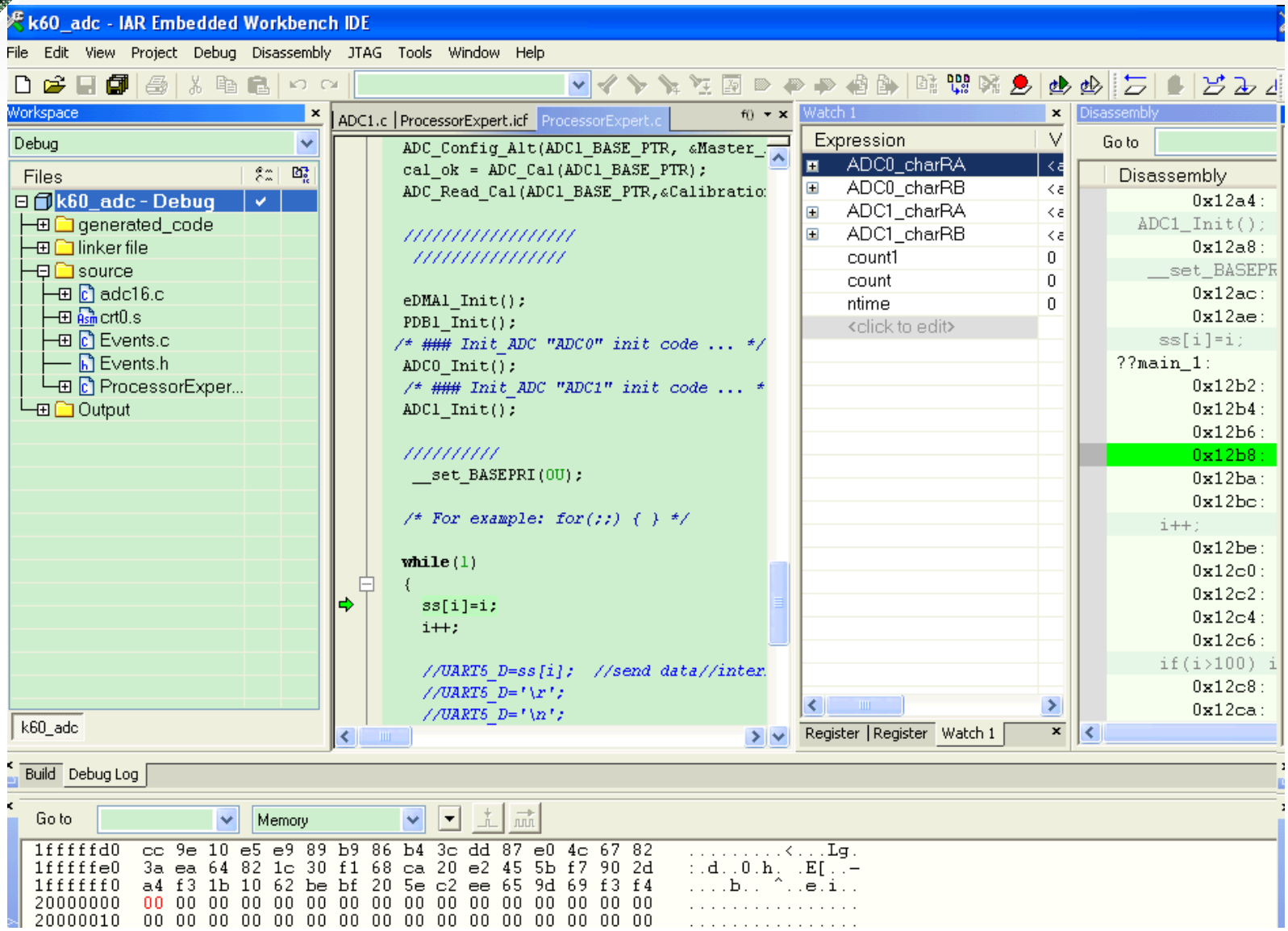
然后就可以编译连接了。



烧写调试



烧写调试（续）



方案1软件设计与调试

1、采用独立ProcessorExpert （简称PE）来生成工程的结构

可先创建一个新目录， 如d:\project, 用于存放建立的工程。建立一个新工程的过程如下：

1）File -> new ->Processor expert project
2）Project name: K60_ADC1_PDB
3）芯片名称：K60→MK60DN512Zxxx10
4）编译器：IAR ARM C complier
5）先不做任何配置，直接生成代码。

2、在IAR下建立一个新的空工程，将工程文件 .ewp存到目录d:\project\ K60_ADC1_PDB， 该目录就是宏$PROJ_DIR$\所代表的目录； 工作空间文件.eww也放到该目录。完成新工程以后，需要在IAR下设置工程属性，添加源文件，编译、链接、调试。请参照前文的内容-----独立ProcessorExpert与IAR的结合所述过程，一步步进行配置。

方案1软件设计与调试（续）

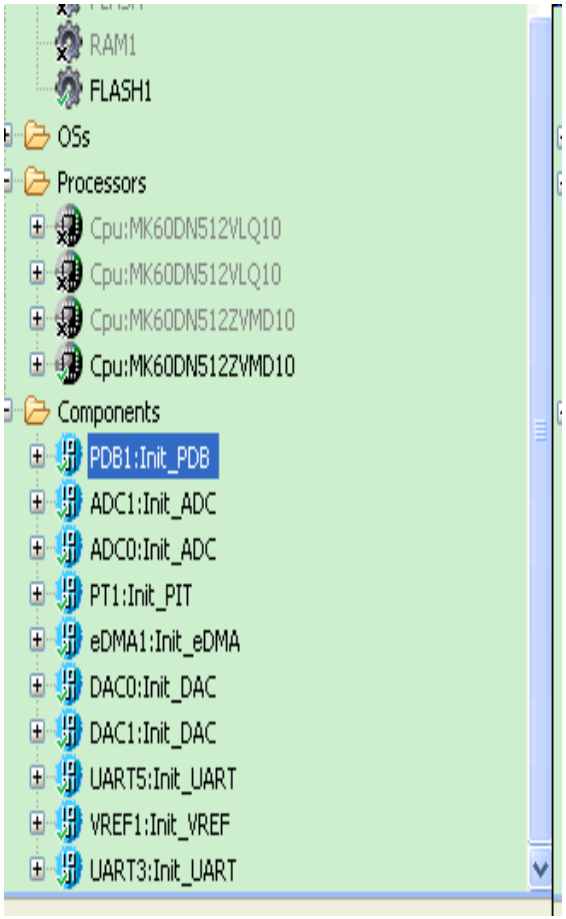
3、新模块增加

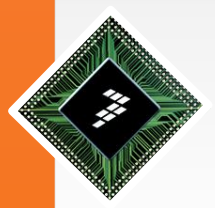
1） 由前两个步骤已经完成了一个由Processor Expert作为辅助工具产生代码，在IAR下工程开发的过程，工程结构是组成该项目的基础构架。

接下来还需要继续增加新模块到PE工程，生成C语言代码后，在IAR下将源代码加入到IAR工程。

2）方案一模块：PIT、PDB、ADC0、ADC1、eDMA、DAC0、DAC1、Uart3、Uart5

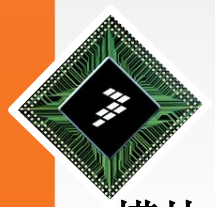
从PE的外设列表里找到Init_PIT、Init_PDB、Init_ADC0、 Init_ADC1、Init_eDMA、Init_DAC0、Init_DAC1、Init_Uart3、Init_Uart5，分别加入到PE工程里。





模块属性设置： PIT

Device	PIT	设备名称
Setttings		
Clock gate	Enabled	时钟开关
Clock settings		
Base clock frequency	48 MHz	系统时钟
Freeze in debug mode	yes	
Channels		
Channel 0		
Timer 0	Enabled	通道0使能
Timer 0 load value	945	
Timer 0 Period	19.708 us	时钟周期
Interrupt		
Interrupt	INT_PIT0	
Interrupt request	Disabled	中断禁止
Interrupt priority	0 (Highest)	
ISR Name	pit0_int	
Timer interrupt	Disabled	
Channel 1		
Timer 1	Enabled	通道1使能
Timer 1 load value	480000	
Timer 1 Period	10.000 ms	时钟周期
Interrupt		
Interrupt	INT_PIT1	
Interrupt request	Enabled	中断使能
Interrupt priority	4	
ISR Name	pit1_int	中断函数
Timer interrupt	Enabled	
Channel 2		
Timer 2	Enabled	通道2使能
Timer 2 load value	96000	
Timer 2 Period	2.000 ms	
Interrupt		
Interrupt	INT_PIT2	
Interrupt request	Enabled	中断使能
Interrupt priority	4	
ISR Name	pit2_int	
Timer interrupt	Enabled	
Channel 3		
Initialization		
Module	Enabled	模块使能
Call Init method	no	初始化不调用



模块属性设置： PDB

Device	PDB0	
Settings		
Clock gate	Enabled	
Clock settings		
Prescaler	divide by 1	
Divider	divide by 1	
Counter frequency	48 MHz	
Modulus	945	
Counter period	19.687 us	PDB时钟周期
Continuous mode enable	One-shot	单个时钟周期
Interrupt delay value	1	
Interrupt delay	0.021 us	
Load Mode	Synchronized with input trigger	
Channels		
Channel triggers		
Channel 0 trigger		
Trigger A		
Trigger A enable	Enabled	通道0触发A使能
Output select A	Delay only	
Back to Back	Disabled	
Trigger delay A value	48	
Trigger A delay	1.000 us	触发A延迟
Trigger B		
Trigger B enable	Enabled	通道0触发B使能
Output select B	Delay only	
Back to Back	Disabled	
Trigger delay B value	514	
Trigger B delay	10.708 us	触发B延迟
Channel 1 trigger		
Trigger A		
Trigger A enable	Enabled	通道0触发A使能
Output select A	Delay only	
Back to Back	Disabled	
Trigger delay A value	48	
Trigger A delay	1.000 us	触发A延迟
Trigger B		
Trigger B enable	Enabled	通道0触发B使能
Output select B	Delay only	
Back to Back	Disabled	
Trigger delay B value	514	
Trigger B delay	10.708 us	触发B延迟
Channel for DAC		
Pulse outs		
Pins/Signals		
Input trigger select	PIT_trigger_0	硬件触发
Interrupts/DMA		
Initialization		
Load OK	yes	PDB使能
Enable PDB	yes	
Call Init method	no	初始化不调用

freescale™

模块属性设置： ADC1

Device	ADC1	
Settings		
Clock gate	Enabled	时钟开关使能
Clock settings		
Input clock select	Bus clock	时钟选择
Prescaler	4	
Frequency	12000 kHz	ADC时钟
High-speed conversion	Enabled	高速转换使能
Asynchro clock output	Disabled	
Long sample time	Disabled	
Long sample time length	20 ADCK cycles	
Conversion mode	Single	单次转换
Result data format	16-bit right	16位模式
Low power mode	Disabled	
Conversion trigger	HW	硬件触发
HW average settings		
HW average	Disabled	
HW average length	4 samples	
Single conversion time - Single-ended	2.77 us	单次转换时间
Single conversion time - Differential	3.52 us	
Additional conversion time - Single-ended	2.25 us	
Additional conversion time - Differential	3.00 us	
Compare settings		
Compare	Disabled	
Compare value 1	0	
Compare value 2	0	
Relation of the CV1 to CV2	Less than or equal	
Compare function	Result < CV1	
Offset	4	
Voltage reference	Default pin pair	缺省参考电压
Pins/Signals		
Channel 0	Disabled	

模块属性设置： ADC1

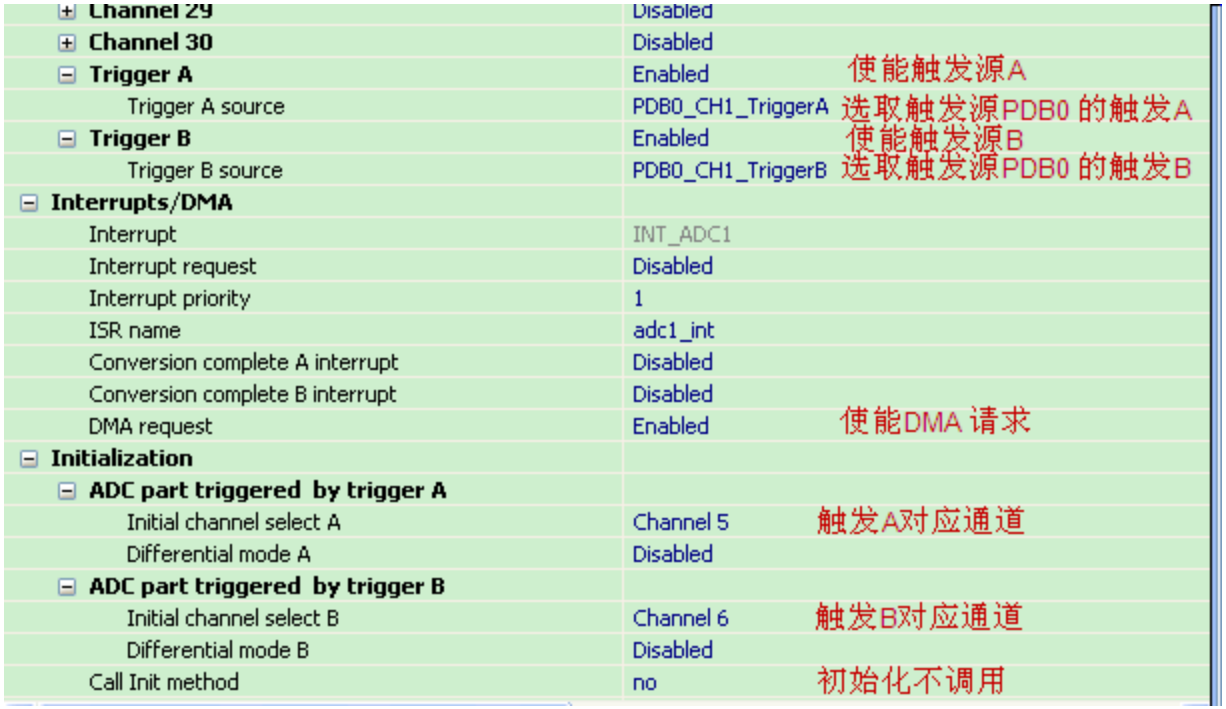
Pins/Signals		
Channel 0		Disabled
Channel 1		Disabled
Channel 2		Disabled
Channel 3		Disabled
Channel 4		Disabled
Channel 5		Enabled
Channel 5 single input	通道5	PTE1/SPI1_SOUT/UART1_RX/SDHC0_D0/I2C1_SCL/ADC1_S
Channel 6		Enabled
Channel 6 single input	通道6	PTE2/SPI1_SCK/UART1_CTS_b/SDHC0_DCLK/ADC1_SE6a
Channel 7		Disabled

Channel 29		Disabled
Channel 30		Disabled
Trigger A		Enabled 使能触发源A
Trigger A source		PDB0_CH1_TriggerA 选取触发源PDB0 的触发A
Trigger B		Enabled 使能触发源B
Trigger B source		PDB0_CH1_TriggerB 选取触发源PDB0 的触发B
Interrupts/DMA		
Interrupt		INT_ADC1
Interrupt request		Disabled
Interrupt priority		1
ISR name		adc1_int
Conversion complete A interrupt		Disabled
Conversion complete B interrupt		Disabled
DMA request		Enabled 使能DMA 请求
Initialization		
ADC part triggered by trigger A		
Initial channel select A		Channel 5 触发A对应通道
Differential mode A		Disabled
ADC part triggered by trigger B		
Initial channel select B		Channel 6 触发B对应通道
Differential mode B		Disabled
Call Init method		no 初始化不调用

模块属性设置： ADC0

Device	ADC0	
Settings		
Clock gate	Enabled	时钟开关使能
Clock settings		
Input clock select	Bus clock	时钟选择
Prescaler	4	
Frequency	12000 kHz	ADC时钟
High-speed conversion	Enabled	高速转换使能
Asynchro clock output	Disabled	
Long sample time	Disabled	
Long sample time length	20 ADCK cycles	
Conversion mode	Single	单次转换
Result data format	16-bit right	16位模式
Low power mode	Disabled	
Conversion trigger	HW	硬件触发
HW average settings		
HW average	Disabled	
HW average length	4 samples	
Single conversion time - Single-ended	2.77 us	单次转换时间
Single conversion time - Differential	3.52 us	
Additional conversion time - Single-ended	2.25 us	
Additional conversion time - Differential	3.00 us	
Compare settings		
Compare	Disabled	
Compare value 1	0	
Compare value 2	0	
Relation of the CV1 to CV2	Less than or equal	
Compare function	Result < CV1	
Offset	4	
Voltage reference	Default pin pair	缺省参考电压
Pins/Signals		
Channel 0	Disabled	
Channel 1	Disabled	

模块属性设置： ADC0



模块属性设置： eDMA

Device	DMA	
Settings		
Clock gate for DMA	Enabled	打开DMA时钟开关
Clock gate for DMA multiplexor	Enabled	打开DMA复用器时钟开关
Peripheral access control	Peripheral access contro settings in the CPU component	
Master privilege level	Masters privilege settings in the CPU component	
Channel arbitration	Fixed priority	固定优先级
Minor loop mapping	Enabled	
Continuous link mode	Enabled	连续链接模式
Halt on error	Disabled	
Enable debug	yes	
Channels		
Channel 0	Initialize	初始化使能
Settings		
Auto disable external request	Disabled	
Preemption	Disabled	
Preempt ability	Enabled	
Channel arbitration priority	Default	通道竞争优先级
Bandwidth control	No stalls	
DMA source muxing		
Channel	Enabled	通道复用使能
Channel trigger	Disabled	
Channel source	ADC0	通道请求源为ADC0
Data source		
External object declaration		
Address	0x4003b010	源地址为ADC0数据寄存器
Transfer size	16-bit	16位数据
Address offset	2	地址偏移为2字节
Address adjustment	0	
Address modulo	8 Bytes	源地址为3个字节，即
Data destination		0x4003b010~0x4003b018，刚好是
External object declaration		RA、RE寄存器地址空间
Address	0x20000000	目的地址

模块属性设置： eDMA（续）

Data destination		
External object declaration		
Address	0x20000000	目的地址放置ADC采样数据
Transfer size	16-bit	16位数据
Address offset	2	地址偏移+2
Address adjustment	0	不使用SLAST
Address modulo	2 Kbytes	使用2K字节数据缓冲区
Minor loop maping	Destination address offset	
Minor loop offset value	2	minor Loop 为2，每次
Block length	2	传两个16位数据
Minor linking	Enabled	
Major iteration count	1	major loop 为1
Minor link channel number	0	通道链接到本通道
Major linking	Disabled	
Major link channel number	0	
Scatter/gather processing	Disabled	
Interrupts		
Transfer done		
Half transfer interrupt	Disabled	
Transfer complete interrupt	Disabled	无中断
Error		
Pins		
DMA trigger	Disabled	
Initialization		
External request	Enabled	
Start transfer	yes	
Channel 1	Initialize	通道1初始化
Settings		
Auto disable external request	Disabled	
Preemption	Enabled	
Preempt ability	Enabled	

模块属性设置： eDMA （续）

Channel 1	Initialize	
Settings		
Auto disable external request	Disabled	
Preemption	Enabled	优先级使能
Preempt ability	Enabled	
Channel arbitration priority	Default	
Bandwidth control	No stalls	
DMA source muxing		
Channel	Enabled	通道使能
Channel trigger	Disabled	
Channel source	ADC1	DMA通道请求源
Data source		
External object declaration		
Address	0x400bb010	ADC1 RA寄存器地址
Transfer size	16-bit	16位数据
Address offset	2	地址偏移+2， 指向RB地址
Address adjustment	0	
Address modulo	8 Bytes	源地址为8个字节，即 0x4003b010~0x4003b018, 刚好是 RA、RB寄存器地址空间
Data destination		
External object declaration		
Address	0x20004000	放ADC1的采集数据地址
Transfer size	16-bit	16位数据
Address offset	2	地址偏移+2
Address adjustment	0	
Address modulo	2 Kbytes	DMA缓冲区
Minor loop maping	Destination address offset	
Minor loop offset value	2	
Block length	2	minor Loop 为2，每次传两个16位数据
Minor linking	Enabled	major loop 为1
Major iteration count	1	
Minor link channel number	1	通道链接到本通道
Major linking	Disabled	不使用major link

模块属性设置： eDMA （续）

Minor linking	Enabled	
Major iteration count	1	
Minor link channel number	1	
Major linking	Disabled	
Major link channel number	0	
Scatter/gather processing	Disabled	
Interrupts		
Transfer done		
Half transfer interrupt	Disabled	
Transfer complete interrupt	Disabled	无中断
Error		
Pins		
Initialization		
External request	Enabled	请求使能
Start transfer	yes	初始化后立即传输
Channel 2	Do not initialize	
Channel 3	Do not initialize	
Channel 4	Do not initialize	
Channel 5	Do not initialize	
Channel 6	Do not initialize	
Channel 7	Do not initialize	
Channel 8	Do not initialize	
Channel 9	Do not initialize	
Channel 10	Do not initialize	
Channel 11	Do not initialize	
Channel 12	Do not initialize	
Channel 13	Do not initialize	
Channel 14	Do not initialize	
Channel 15	Initialize	
Interrupts		
Initialization		
Call Init method	yes	初始化调用

模块属性设置： uart5

Device	UART5	
Settings		
Clock gate	Enabled	时钟使能
Clock settings		
Baud rate divisor	26	
Baud rate fine adjust	1	
Baud rate	115246.098 baud	波特率115200
Transfer settings		
Data format	8bit	8位模式
Bits ordering	LSB first	
Parity	Off	
Parity placement	Parity in last data bit	
Idle character counting	After start bit	
Break character generation length	Short	
LIN Break detection	Disabled	
Stop in Wait mode	Disabled	
Receiver wakeup settings		
Modem settings		
Infrared settings		
FIFOs settings		
Loops and Single wire settings		
Receiver input	Not inverted	
Transmitter output	Not inverted	
Pins/Signals		
Interrupts/DMA		
Common Tx/Rx interrupt		
Interrupt	INT_UART5_RX_TX	
Interrupt request	Disabled	
Interrupt priority	4	优先级
ISR name	uart5_int	
Transmit empty request	Disabled	
Transmit empty request type	DMA	
Transmit complete request	Disabled	

模块属性设置： uart5 （续）

Interrupts/DMA	
Common Tx/Rx interrupt	
Interrupt	INT_UART5_RX_TX
Interrupt request	Disabled
Interrupt priority	4
ISR name	uart5_int
Transmit empty request	Disabled
Transmit empty request type	DMA
Transmit complete request	Disabled
Receiver full request	Disabled 无中断
Receiver full request type	IRQ
Idle line request	Disabled
LIN break detect request	Disabled
Rx active edge interrupt	Disabled
Error Interrupt	
Error interrupt	INT_UART5_ERR
Interrupt request	Disabled
Interrupt priority	0 (Highest)
ISR name	
Overrun error interrupt	Disabled
Noise error interrupt	Disabled
Framing error interrupt	Disabled
Parity error interrupt	Disabled
FIFOs interrupts	
Tx FIFO overflow interrupt	Disabled
Rx FIFO underflow interrupt	Disabled
Initialization	
Send break	Disabled
Enable transmitter	Enabled 打开发、收驱动器
Enable receiver	Enabled
Call Init method	no 初始化不调用

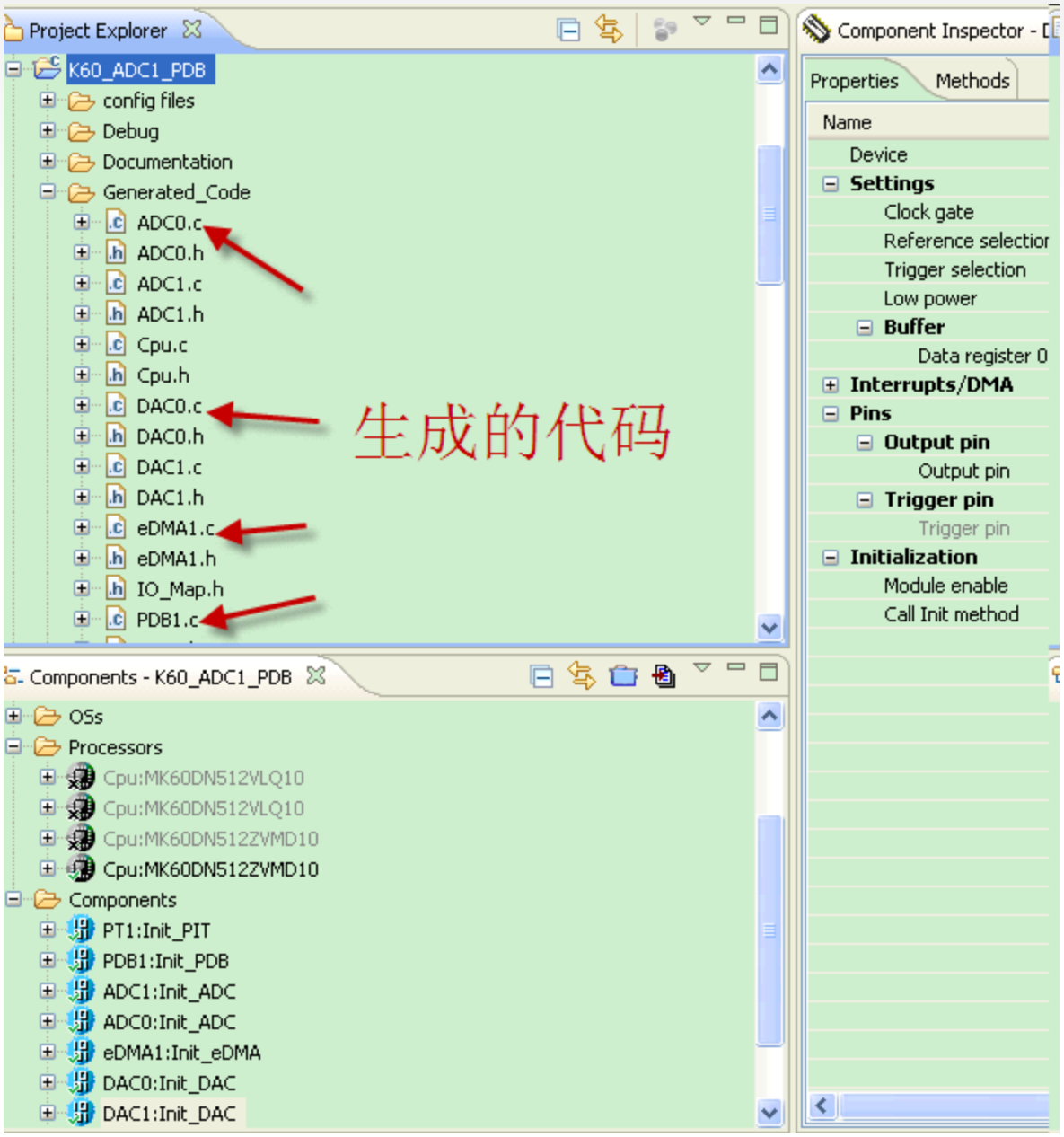
模块属性设置： DAC0

Device	DAC0	
Settings		
Clock gate	Enabled	时钟使能
Reference selection	External	
Trigger selection	SW	
Low power	Disabled	
Buffer	Disabled	
Data register 0	1000	数据初始值
Interrupts/DMA		
Pins		
Output pin	Enabled	打开管脚输出
Output pin	DAC0_OUT	
Trigger pin	Disabled	
Initialization		
Module enable	yes	模块使能
Call Init method	no	初始化不调用

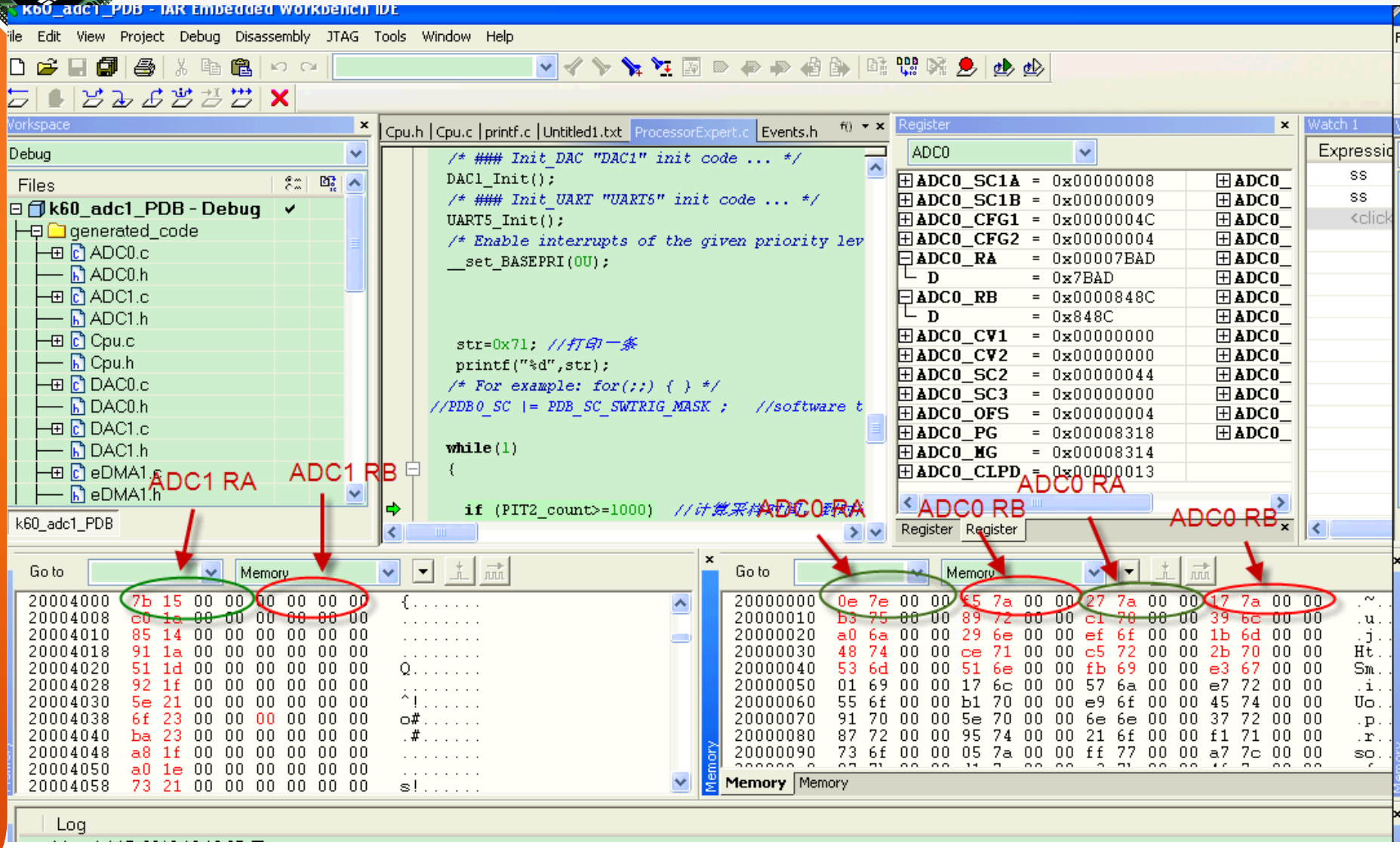
模块属性设置： DAC1

Device	DAC1	
Settings		
Clock gate	Enabled	时钟使能
Reference selection	External	
Trigger selection	SW	
Low power	Disabled	
Buffer	Disabled	
Data register 0	750	数据初始值
Interrupts/DMA		
Pins		
Output pin	Enabled	打开管脚输出
Output pin	DAC1_OUT	
Trigger pin	Disabled	
Trigger pin	PDB0_CH2_Output	
Initialization		
Module enable	yes	模块使能
Call Init method	no	初始化不调用

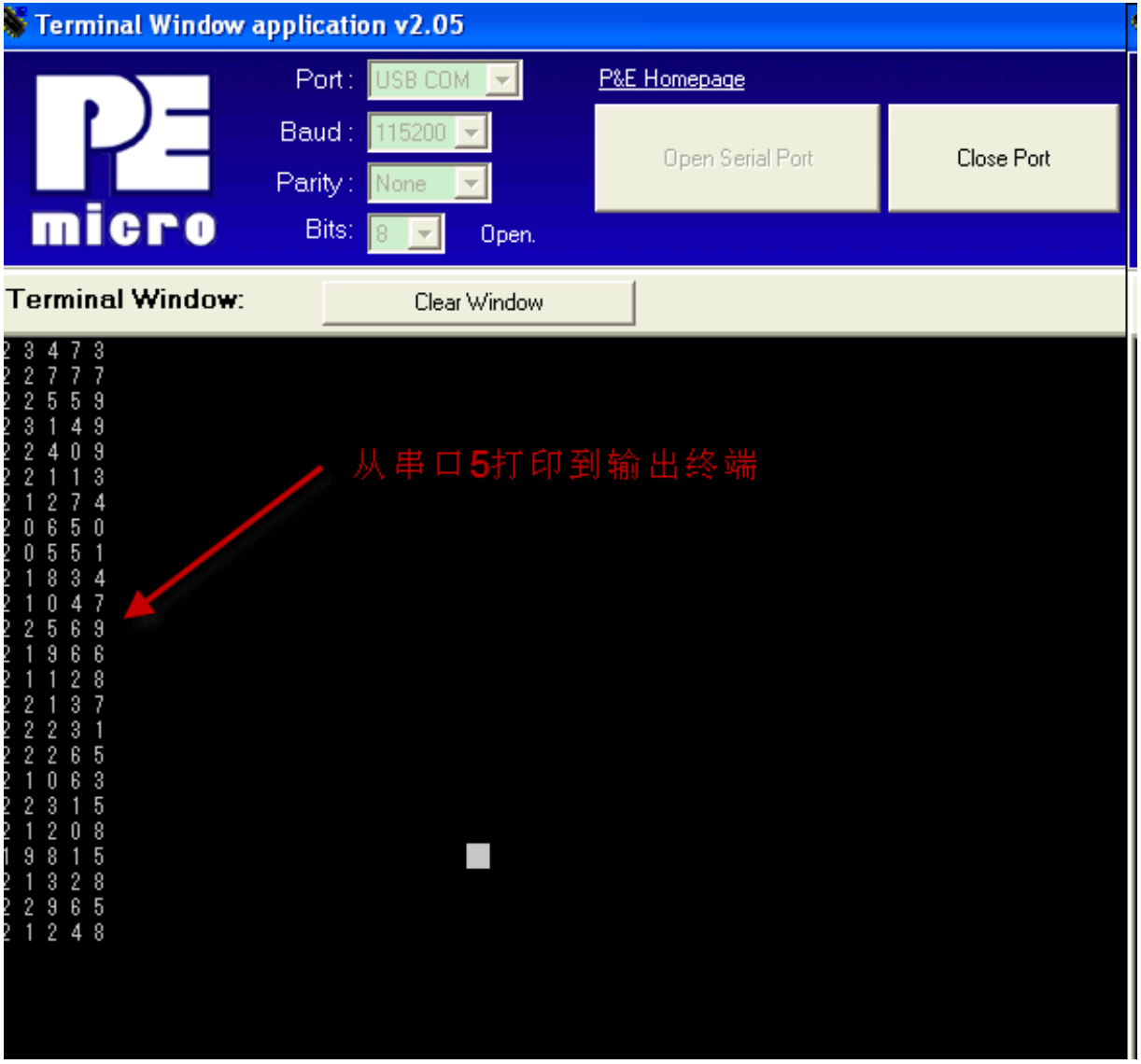
代码生成



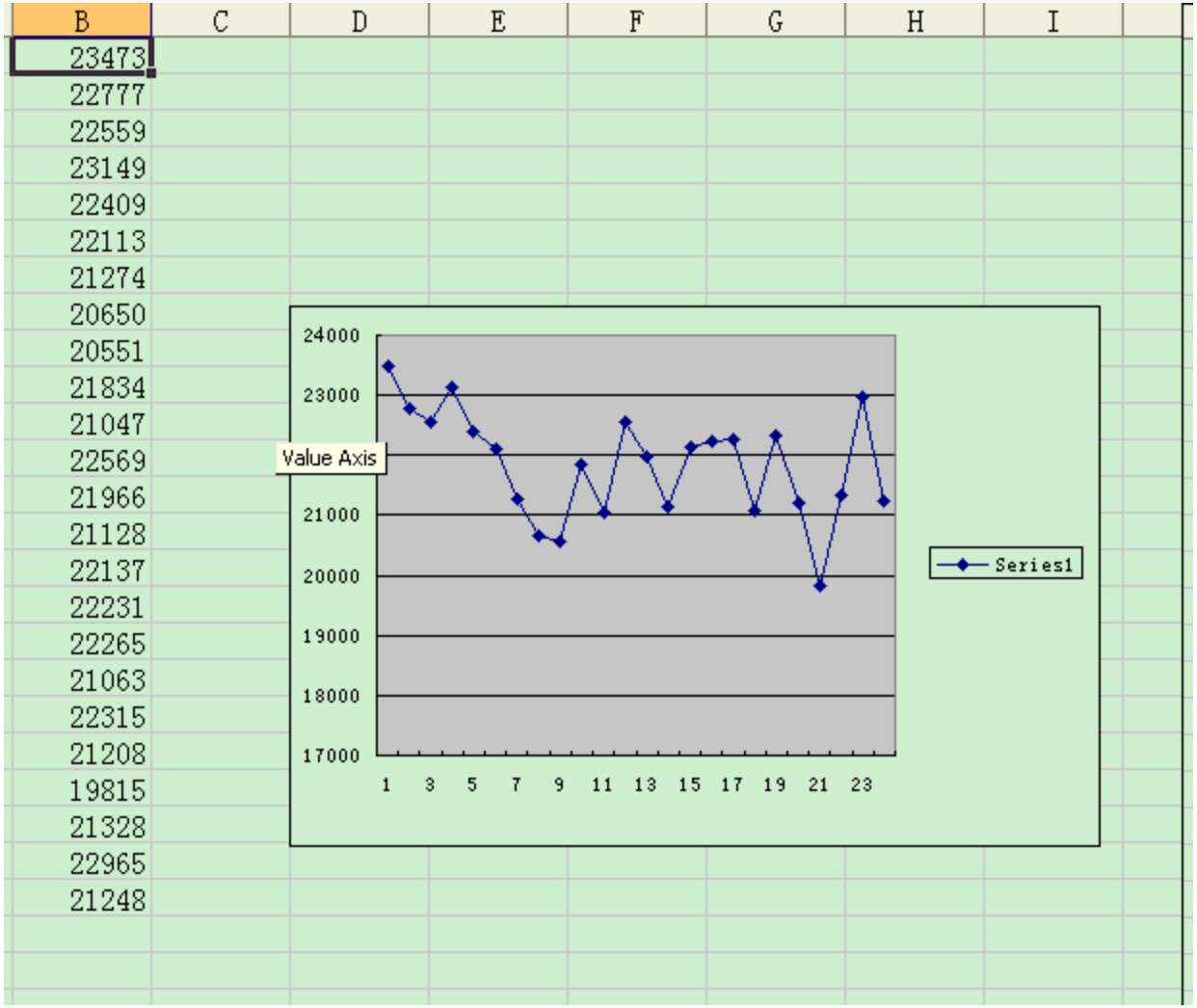
在IAR下编译、调试、链接



打印输出结果



在excel表中做曲线



小结：
方案1是采用两个ADC模块，采集四路信号，每个ADC模块负责采集两路信号。由于ADC模块有两个结果寄存器RA和RB，分别对应不同的通道，就可以用PDB的两个延迟触发脉冲对ADC两次触发，完成对两个通道数据的采集。

实验运用了PIT、PDB、DMA联动方式，将采集数据放到内存里，整个过程中没有使用中断函数，所以采集过程不增加CPU的负荷，节省了CPU后期处理的时间，从而降低了系统整体负荷。

方案2软件设计与调试

方案2与方案1的区别：

1. 采用一个ADC模块，4路通道都属于ADC1；
2. 需要切换通道，用两种方法来完成通道切换工作（DMA或中断）；
3. 只使用RA结果寄存器；
4. ADC触发源采用PIT；

项目建立过程同方案1所描述：

1. 在PE下建立新项目K60_ADC1_4_channels_test， 增加可视化构件，PIT、ADC1、eDMA、Uart5等。
2. 在IAR下建立新工程K60_ADC1_4_channels_test, 并配置相关选项
3. 在PE下配置模块属性，并生成代码。
4. 在IAR下，添加代码，编译、链接、调试。

模块属性设置： PIT

Device	PIT
Setttings	
Clock gate	Enabled
Clock settings	
Base clock frequency	48 MHz
Freeze in debug mode	yes
Channels	
Channel 0	
Timer 0	Enabled
Timer 0 load value	460
Timer 0 Period	9.604 us
Interrupt	
Interrupt	INT_PIT0
Interrupt request	Disabled
Interrupt priority	0 (Highest)
ISR Name	pit0_int
Timer interrupt	Disabled
Channel 1	
Timer 1	Enabled
Timer 1 load value	480000
Timer 1 Period	10.000 ms
Interrupt	
Interrupt	INT_PIT1
Interrupt request	Disabled
Interrupt priority	4
ISR Name	pit1_int
Timer interrupt	Enabled
Channel 2	
Channel 3	
Initialization	
Module	Enabled
Call Init method	yes

采样触发时间

用于定时

初始化调用

模块属性设置： ADC1

Device	ADC1
Settings	
Clock gate	Enabled
Clock settings	
Input clock select	Bus clock
Prescaler	4
Frequency	12000 kHz
High-speed conversion	Enabled
Asynchro clock output	Disabled
Long sample time	Disabled
Long sample time length	20 ADCK cycles
Conversion mode	Single
Result data format	16-bit right
Low power mode	Disabled
Conversion trigger	HW
HW average settings	
HW average	Disabled
HW average length	4 samples
Single conversion time - Single-ended	2.77 us
Single conversion time - Differential	3.52 us
Additional conversion time - Single-ended	2.25 us
Additional conversion time - Differential	3.00 us
Compare settings	
Compare	Disabled
Compare value 1	0 D
Compare value 2	0 D
Relation of the CV1 to CV2	Less than or equal
Compare function	Result < CV1
Offset	4 D
Voltage reference	Default pin pair
Pins/Signals	
Channel 0	Disabled
Channel 1	Disabled

模块属性设置： ADC1 （续）

Channel 4	Enabled
Channel 4 single input	PTE0/SPI1_PCS1/UART1_TX/SDHC0_D1/I2C1_SDA/ADC1_SE4a
Channel 5	Enabled
Channel 5 single input	PTE1/SPI1_SOUT/UART1_RX/SDHC0_D0/I2C1_SCL/ADC1_SE5a
Channel 6	Enabled
Channel 6 single input	PTE2/SPI1_SCK/UART1_CTS_b/SDHC0_DCLK/ADC1_SE6a
Channel 7	Enabled
Channel 7 single input	PTE3/SPI1_SIN/UART1_RTS_b/SDHC0_CMD/ADC1_SE7a
Channel 8	Disabled
Channel 9	Disabled
Channel 10	Disabled
Channel 11	Disabled
Channel 12	Disabled
Channel 27	Disabled
Channel 28	Disabled
Channel 29	Disabled
Channel 30	Disabled
Trigger A	Enabled
Trigger A source	PIT_trigger_0
Trigger B	Disabled
Interrupts/DMA	
Interrupt	INT_ADC1
Interrupt request	Disabled
Interrupt priority	1
ISR name	adc1_int
Conversion complete A interrupt	Disabled
Conversion complete B interrupt	Disabled
DMA request	Enabled
Initialization	
ADC part triggered by trigger A	
Initial channel select A	Channel 4
Differential mode A	Disabled
ADC part triggered by trigger B	
Initial channel select B	ADC disabled
Differential mode B	Disabled
Call Init method	yes

ADC1的4~7 通道

采用PIT通道0
作为触发源

打开DMA请求

初始通道4

初始化调用

模块属性设置：eDMA

Device	DMA
Settings	
Clock gate for DMA	Enabled
Clock gate for DMA multiplexor	Enabled
Peripheral access control	Peripheral access control settings in the CPU component
Master privilege level	Masters privilege settings in the CPU component
Channel arbitration	Fixed priority
Minor loop mapping	Enabled
Continuous link mode	Enabled
Halt on error	Disabled
Enable debug	yes
Channels	
Channel 0	Initialize
Settings	
Auto disable external request	Disabled
Preemption	Enabled
Preempt ability	Enabled
Channel arbitration priority	Default
Bandwidth control	No stalls
DMA source muxing	
Channel	Enabled
Channel trigger	Disabled
Channel source	Disabled
Data source	
External object declaration	unsigned char str[4]={4,5,6,7};
Address	&str[0]
Transfer size	8-bit
Address offset	1
Address adjustment	0
Address modulo	4 Bytes
Data destination	
External object declaration	
Address	(uint32) &ADC1_SC1A

模块属性设置： eDMA （续）

Address adjustment	0	D
Address modulo	4 Bytes	
Data destination		
External object declaration		
Address	(uint32) &ADC1_SC1A	ADC1 RA 寄存器地址
Transfer size	8-bit	8位数据，是指ADC1-SC1A低8位
Address offset	0	D
Address adjustment	0	D
Address modulo	Buffer disabled	
Minor loop maping	Disabled	
Block length	1	每次请求传输一个字节数据 D
Minor linking	Disabled	
Major iteration count	1	主循环为1 D
Major linking	Disabled	
Major link channel number	0	D
Scatter/gather processing	Disabled	
Interrupts		
Transfer done		
Half transfer interrupt	Disabled	
Transfer complete interrupt	Disabled	
Error		
Pins		
DMA trigger	Disabled	
Initialization		
External request	Enabled	使能并立即传输
Start transfer	yes	
Channel 1	Initialize	通道1初始化
Settings		
Auto disable external request	Disabled	
Preemption	Enabled	
Preempt ability	Enabled	优先级使能
Channel arbitration priority	Default	
Bandwidth control	No stalls	

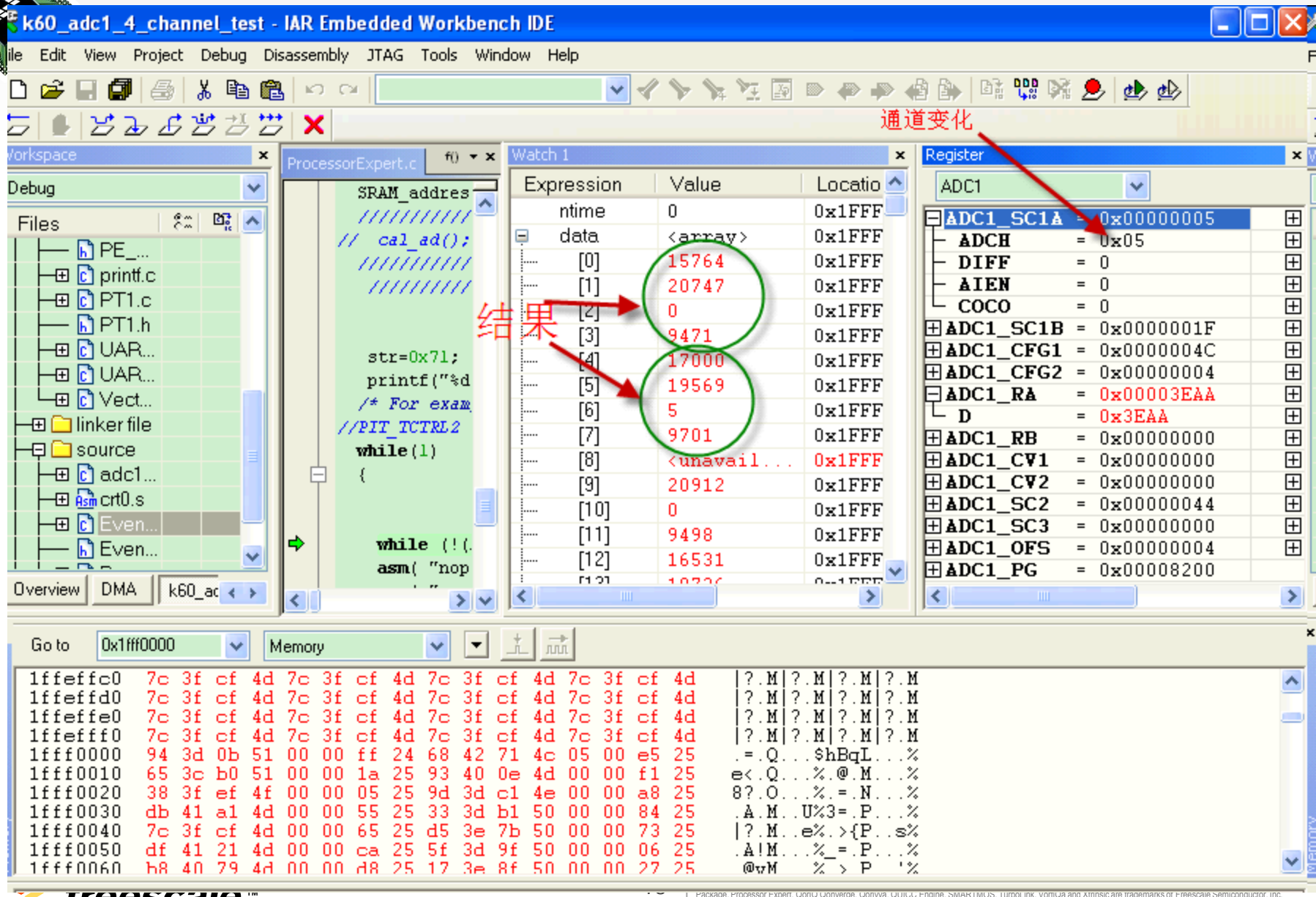
模块属性设置： eDMA （续）

Preempt ability	Enabled	
Channel arbitration priority	Default	
Bandwidth control	No stalls	
DMA source muxing		
Channel	Enabled	通道复用器使能
Channel trigger	Disabled	
Channel source	ADC1	请求源为ADC1
Data source		
External object declaration		
Address	0x400bb010	ADC1的RA地址
Transfer size	16-bit	16位数据
Address offset	0	D
Address adjustment	0	D
Address modulo	Buffer disabled	
Data destination		
External object declaration	unsigned short data[256]={1,1,1,1};//0x20000000	存放数据数组
Address	&data[0]	数组内存的起始地址
Transfer size	16-bit	16位数据
Address offset	2	地址偏移+2 D
Address adjustment	0	D
Address modulo	512 Bytes	DMA缓冲区，大小和data数组相同
Minor loop maping	Disabled	
Block length	2	D
Minor linking	Enabled	
Major iteration count	0	D
Minor link channel number	0	D
Major linking	Disabled	
Major link channel number	0	D
Scatter/gather processing	Disabled	
Interrupts		
Pins		
DMA trigger	Disabled	
Initialization		

模块属性设置： eDMA （续）

⊞ **DMA trigger**	Disabled
⊟ **Initialization**	
External request	Enabled
Start transfer	yes 开始传输
⊞ **Channel 2**	Do not initialize
⊞ **Channel 3**	Do not initialize
⊞ **Channel 4**	Do not initialize
⊞ **Channel 5**	Do not initialize
⊞ **Channel 6**	Do not initialize
⊞ **Channel 7**	Do not initialize
⊞ **Channel 8**	Do not initialize
⊞ **Channel 9**	Do not initialize
⊞ **Channel 10**	Do not initialize
⊞ **Channel 11**	Do not initialize
⊞ **Channel 12**	Do not initialize
⊞ **Channel 13**	Do not initialize
⊞ **Channel 14**	Do not initialize
⊞ **Channel 15**	Do not initialize
⊞ **Interrupts**	
⊟ **Initialization**	
Call Init method	yes 初始化调用

生成代码并调试



注意：由于DMA传输通道号不知何原因有可能会传输失败，通道号与采集的数据就错位了。所以，换通道可以采用DMA传输完成中断来改变通道号。

如：在DMA 通道1 的中断函数里采用下面语句

```
char ch_count=0x0; //通道计数
unsigned char channel[4]={0x4,0x5,0x6,0x7};//通道数组
PE_ISR(DMA1_int)
{

DMA_CINT=0x1;  //清除ch1中断标志

 ch_count++;  //通道计数加1
 if (ch_count>=0x4){ch_count=0x0;} //数组大于3，则置为0
 ADC1_SC1A = ADC_SC1_ADCH(channel[ch_count]);//给ADCH赋值

}
```

- 方案2小结

  采用了单个ADC完成4路信号采集，换通道的过程可以采用DMA通道来帮助切换通道，如：通道0用于切换通道，方法是把内存数组存放的通道号传输给通道寄存器，这种方法不需要中断函数的介入，同样可以减轻CPU的负荷。但实验发现，这种方法有时会造成通道与数据的错位，原因还需分析查找。

  另一种切换通道的方法是需要引入一个中断函数，可以采用DMA传输完成中断来换通道，在高速采集的环境下，引入中断会加大CPU的负荷，提供给CPU做其它处理的时间就减少了。

  方案2适合于多个通道在同一ADC模块的情况，在大多数低速采集的情况下都可以采用。如果需要很快的采集速度，则要按方案1的方法设计。

上手实验

- 方案一

  参考例子工程， 解压文件 K60_ADC1_PDB.rar 到一个目录，在PE下导入工程，可以修改可视化构件的属性，生成代码；在IAR下直接打开对应的工程，可以直接编译。

- 方案二

  参考例子工程，解压文件 K60_ADC1_4_channels_int_test.rar，用于测试DMA传输通道号和采集数据； 解压文件 K60_ADC1_4_channels_test.rar， 用于测试DMA传输采集数据，用DMA中断切换通道。

更多飞思卡尔技术资源、技术支持欢迎访问：

飞思卡尔中文支持社区：
http://www.eefocus.com/bbs/forumall_171.html
飞思卡尔英文支持社区：https://community.freescale.com/