LPC82X 培训资料

状态可配置定时器SCT 动手实验

MAY, 2016

动手实验1
SCT产生PWM输出信号

内容

- 实验简介（目的，内容，结果）
- 软/硬件环境搭建
- 实验步骤

实验简介

- **目的**：通过本实验，理解和掌握LPC82x SCT实现PWM输出信号：
  -设置输出不同周期的PWM信号

- **描述**：计数匹配产生2个事件，分别控制输出信号的高电平输出和低电平输出

- **结果**:LED灯闪烁

软/硬件环境搭建

- 硬件
  - 评估板：LPC824Lite-V1.0

- 工程位置
  - ..\peri_example\sctimer\sct_blinky\project_sct_blinky.uvprojx

实验步骤

- 第一步 – 更改计数器匹配条件，输出不同周期的PWM信号
- 第二步 – 根据连接指示，搭建好硬件环境
- 第三步 – 编译下载程序。运行，LED1灯闪烁，表明输出PWM信号

动手实验2
SCT实现计数器捕获

内容

- 实验简介（目的，内容，结果）
- 软/硬件环境搭建
- 实验步骤

实验简介

- **目的**：通过本实验，理解和掌握LPC82x SCT实现计数器捕获功能

- **描述**：利用P0_1做为SCT的SCT_INPUT0，按S4键产生下降沿的输入信号，该信号做为SCT事件0的触发信号捕获当前计数器值到捕获寄存器0，用DEBUG UART打印出捕获值

- **结果**:上位机串口打印出捕获计数器值

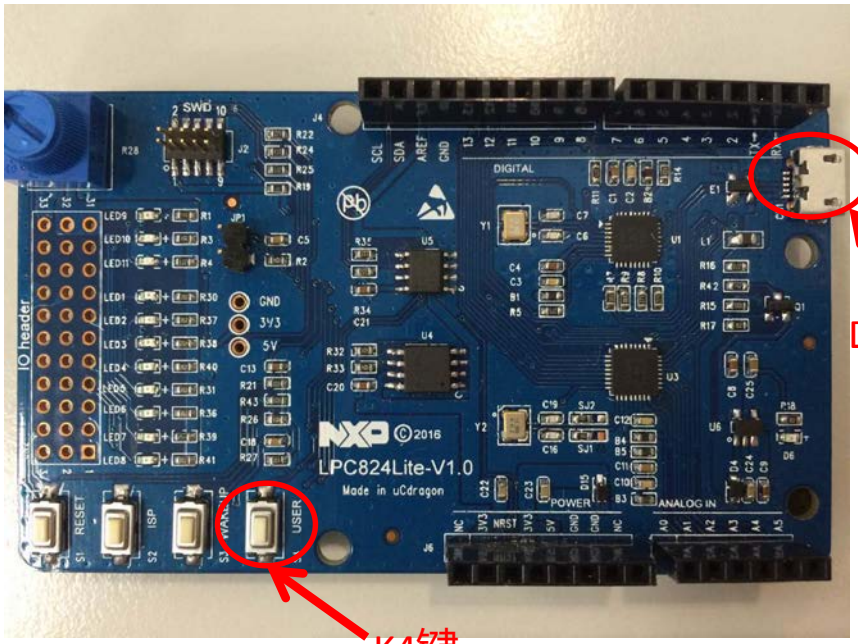
软/硬件环境搭建

- 硬件
  - 评估板：LPC824Lite-V1.0

- 工程位置
  - ..\peri_example\sctimer\sct_capture\project_sct_capture.uvprojx

硬件配置

- 无需特别的硬件配置
  - 使用DEBUG UART来输出调试字符串



实验步骤

- 第一步 – 根据连接指示，搭建好硬件环境
- 第二步 – 上位机设置串口助手为115200波特率，数据位8 - 校验0 - 停止位1
- 第三步 – 编译下载程序，运行
- 第四步 – 按下S4键，串口打印计数器捕获值

动手实验 3
SCT实现交通信号灯

内容

- 实验简介（目的，内容，结果）
- 软/硬件环境搭建
- 实验步骤

实验简介

- **目的：**通过本实验，理解和掌握LPC82x SCT硬件状态机实现交通信号灯

- **描述**：SCT实现4个状态分别代表红灯，绿灯，黄灯和黄灯闪烁，板子上的LED4，LED2和LED6分别表示红灯，黄灯和绿灯

- **结果**:LED模拟交通信号灯闪烁

软/硬件环境搭建

- 硬件
  - 评估板：LPC824Lite-V1.0

- 工程位置
  - ..\peri_example\sctimer\sct_tlight\project_sct_tlight.uvprojx

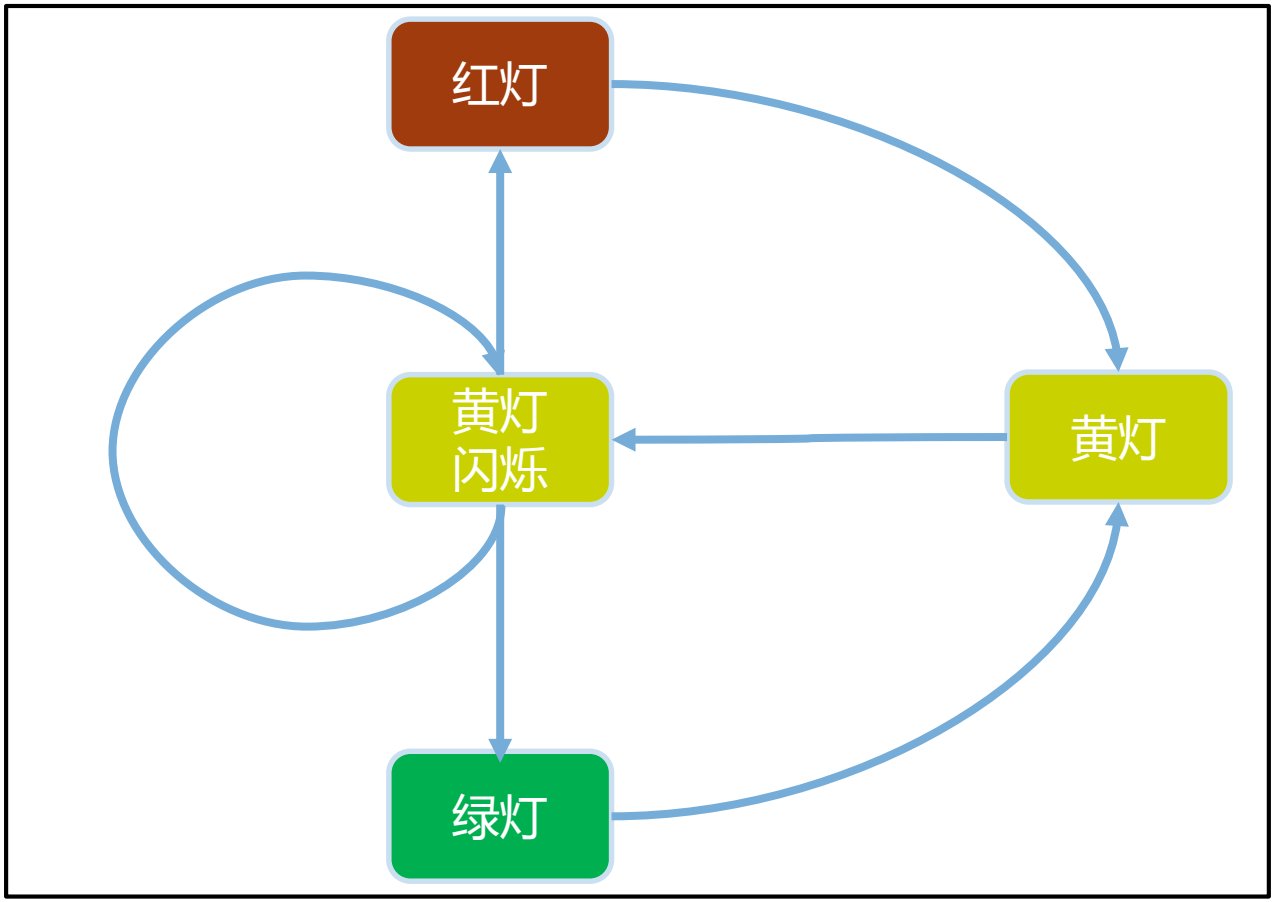
实验步骤

- 第一步 – 根据连接指示，搭建好硬件环境
- 第二步 – 编译下载程序，运行
- 第三步 – 盖板上三个LED2，4，6灯模拟交通灯交替闪烁

SCT状态机实例 - 交通灯

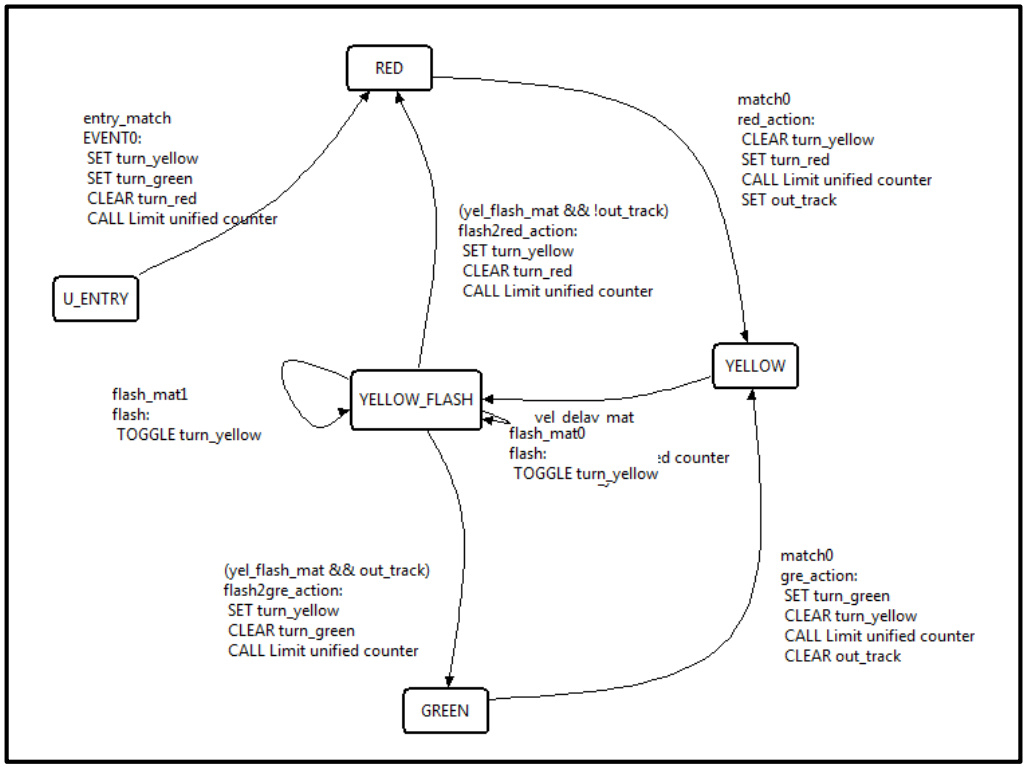
- 交通灯状态变换图

  –红灯-> 黄灯-> 黄灯闪烁-> 绿灯-> 黄灯-> 黄灯闪烁-> 红灯



SCT状态机开发工具

- LPCXpresso IDE Red State
- Red State交通灯状态图及产生代码
- 启动后不需要任何软件干涉，解放CPU




```
void sct_fsm_init (void)
{
LPC_SCT->MATCH[0].U = entry_mat;            /* entry_match */
LPC_SCT->MATCHREL[0].U = entry_mat;
LPC_SCT->MATCH[1].U = flash;              /* flash_mat0 */
LPC_SCT->MATCHREL[1].U = flash;
LPC_SCT->MATCH[2].U = flash1;             /* flash_mat1 */
LPC_SCT->MATCHREL[2].U = flash1;
LPC_SCT->MATCH[3].U = delay;              /* match0 */
LPC_SCT->MATCHREL[3].U = delay;
LPC_SCT->MATCH[4].U = yel_delay;            /* yel_delay_mat */
LPC_SCT->MATCHREL[4].U = yel_delay;
LPC_SCT->MATCH[5].U = yel_flash;            /* yel_flash_mat */
LPC_SCT->MATCHREL[5].U = yel_flash;

/* OUTPUT registers */
LPC_SCT->OUT[5].SET = 0x00000002;       /* out_track */
LPC_SCT->OUT[5].CLR = 0x00000008;
LPC_SCT->OUT[2].SET = 0x00000009;       /* turn_green */
LPC_SCT->OUT[2].CLR = 0x00000080;
LPC_SCT->OUT[0].SET = 0x00000002;       /* turn_red */
LPC_SCT->OUT[0].CLR = 0x00000041;
LPC_SCT->OUT[1].SET = 0x000000F5;       /* turn_yellow */
LPC_SCT->OUT[1].CLR = 0x0000003A;

/* Conflict resolution register */
LPC_SCT->RES = (LPC_SCT->RES & ~0x0000000C) | 0x0000000C;

/* EVENT registers */
LPC_SCT->EV[0].CTRL = 0x0000D000;     /* U: --> state RED */
LPC_SCT->EV[0].STATE = 0x00000001;
LPC_SCT->EV[1].CTRL = 0x00015003;     /* U: --> state YELLOW */
LPC_SCT->EV[1].STATE = 0x00000002;
LPC_SCT->EV[2].CTRL = 0x00025004;     /* U: --> state
YELLOW_FLASH */
LPC_SCT->EV[2].STATE = 0x00000004;
LPC_SCT->EV[3].CTRL = 0x00015003;     /* U: --> state YELLOW */
LPC_SCT->EV[3].STATE = 0x00000008;
LPC_SCT->EV[6].CTRL = 0x0000F165;     /* U: --> state RED */
LPC_SCT->EV[6].STATE = 0x00000010;
LPC_SCT->EV[7].CTRL = 0x0001FD65;     /* U: --> state GREEN */
LPC_SCT->EV[7].STATE = 0x00000010;
LPC_SCT->EV[4].CTRL = 0x00025001;     /* U: --> state
YELLOW_FLASH */
LPC_SCT->EV[4].STATE = 0x00000010;
LPC_SCT->EV[5].CTRL = 0x00025002;     /* U: --> state
YELLOW_FLASH */
LPC_SCT->EV[5].STATE = 0x00000010;

/* STATE registers */
LPC_SCT->STATE_L = 0;
LPC_SCT->LIMIT_L = 0x000000CF;
}
```

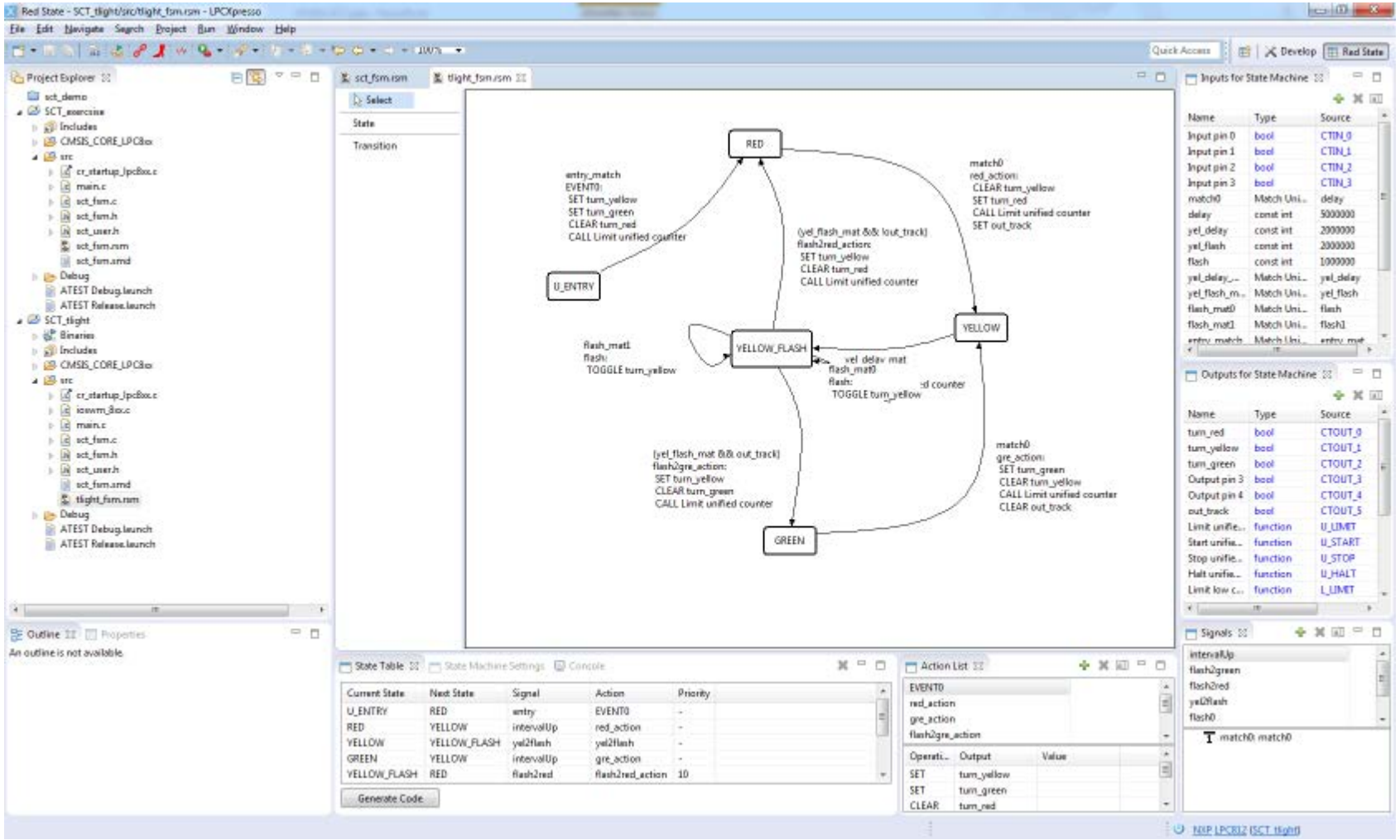
普通Timer实现交通灯对比

- 状态机需要软件实现
- 高度CPU消耗

```
init()
{
red_led = out;
yellow_led = out;
green_led = out;

match = red_interval;
match_interupt = enable;
state = red;
red_led = up;
counter = start;
}
```

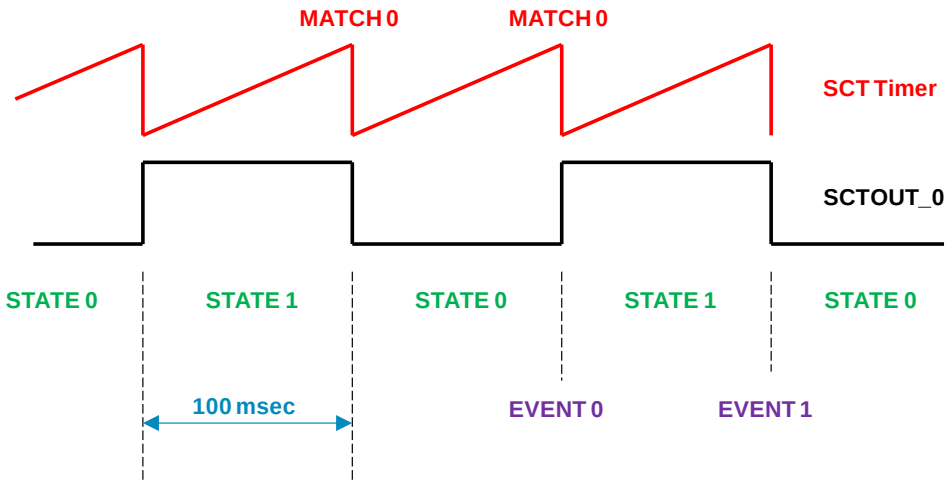
```
match_state_isr()
{
  switch(state)
  {
  case red:
  red_led = out;
  yellow_led = up;
  red_flag = 1;
  match = yellow_interval;
  state = yellow;
  counter = start;
  case yellow:
  yellow_led = flash;
  match = yellowFlash_interval;
  state = yellow_flash;
  counter = start;

  case yellow_flash:
  yellow_led = out;
  if(red_flag == 1)
  {
    green_led = up;
    state = green;
    match = green_interval;
    counter = start;
  }
  else
  {
    red_led = up;
    state = red;
    match = red_interval;
    counter = start;
  }
  case green:
  green_led = out;
  yellow_led = up;
  red_flag = 0;
  match_interval = yellow_interval;
  state = yellow;
  counter = start;
  }
}
```

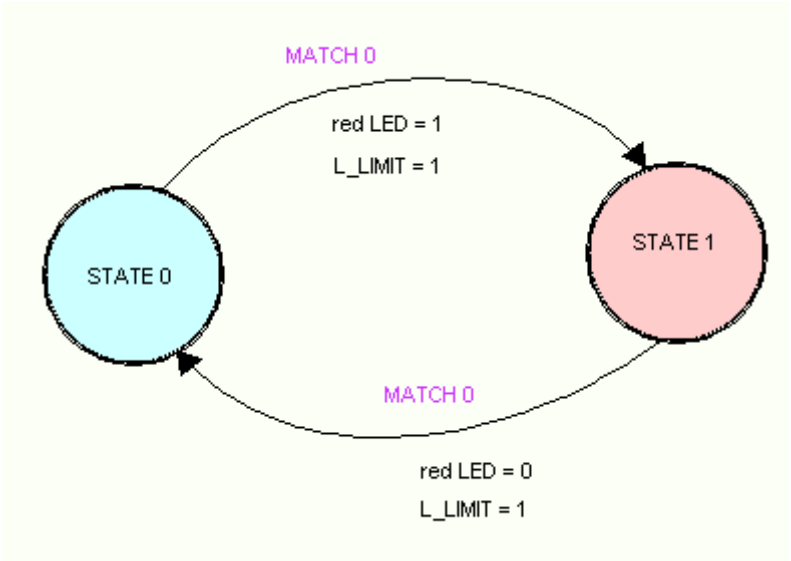
Red State介绍 – 图形化的SCT配置工具



Red State实验

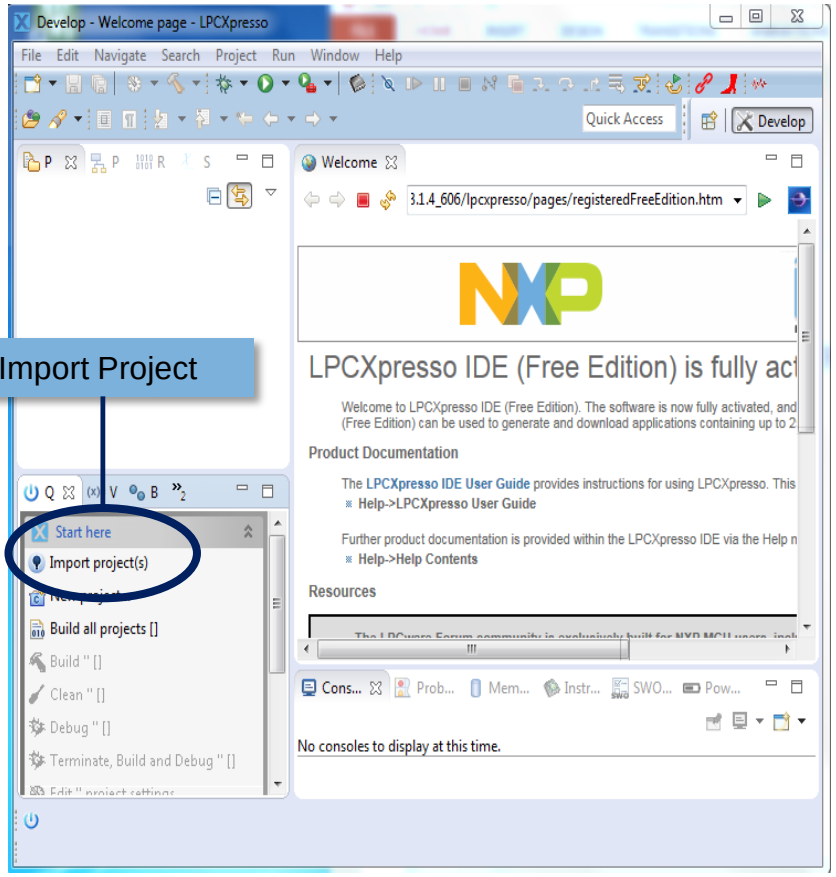
- 使用 SCT 和 Red State 来做前述的实验
- 连接 RED led (P0_17) 到 SCTOUT_0

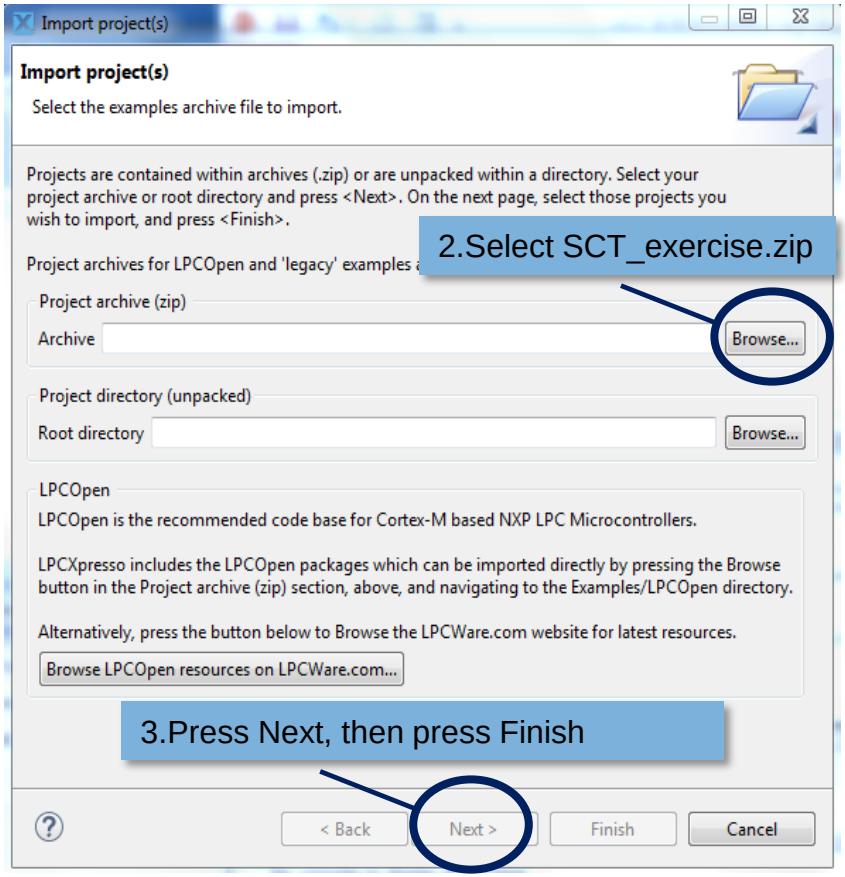


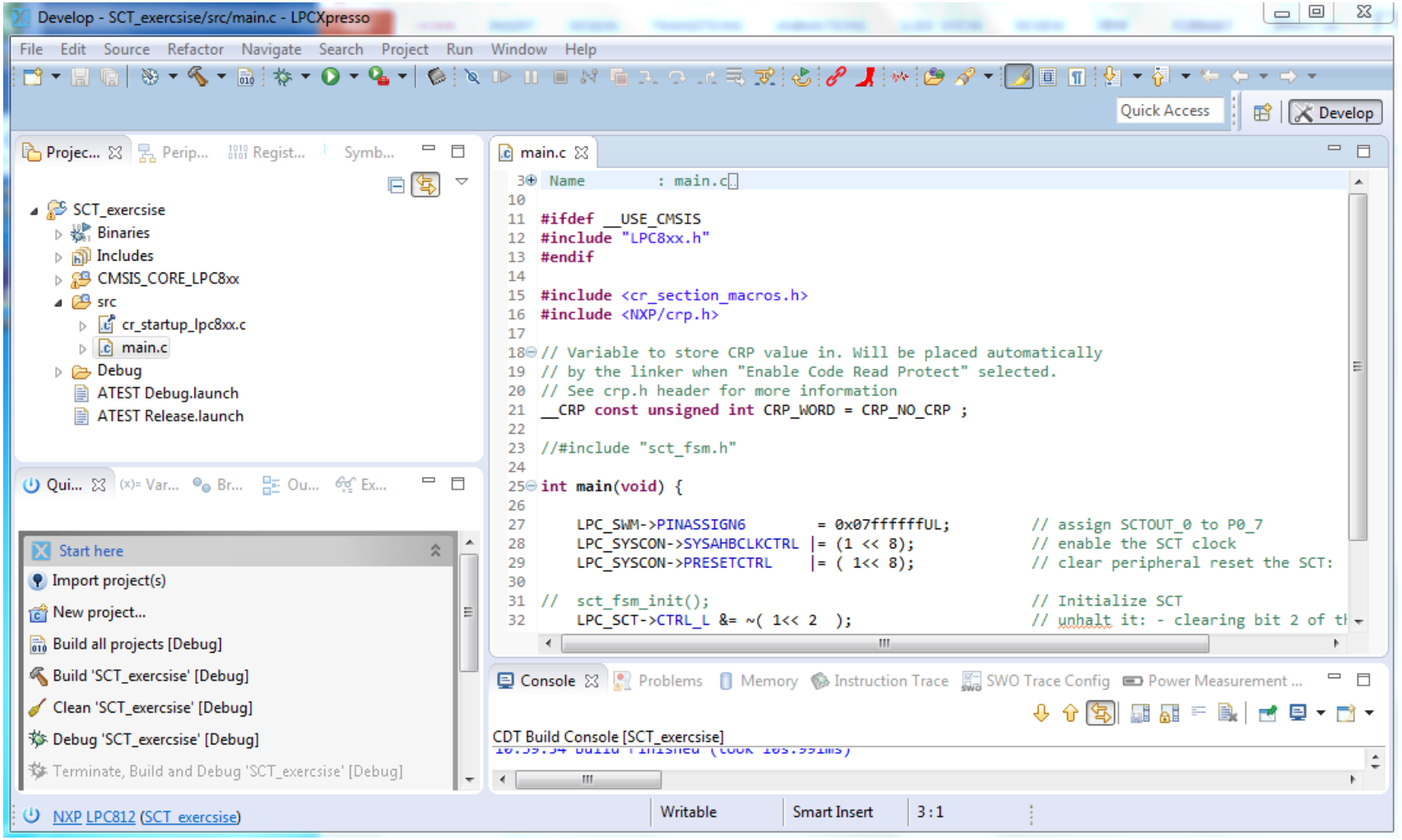


步骤 - 1

- 打开 LPCXpresso， 打开工程 SCT_exercise
  - ..\peri_example\sctimer\SCT_exercise.zip

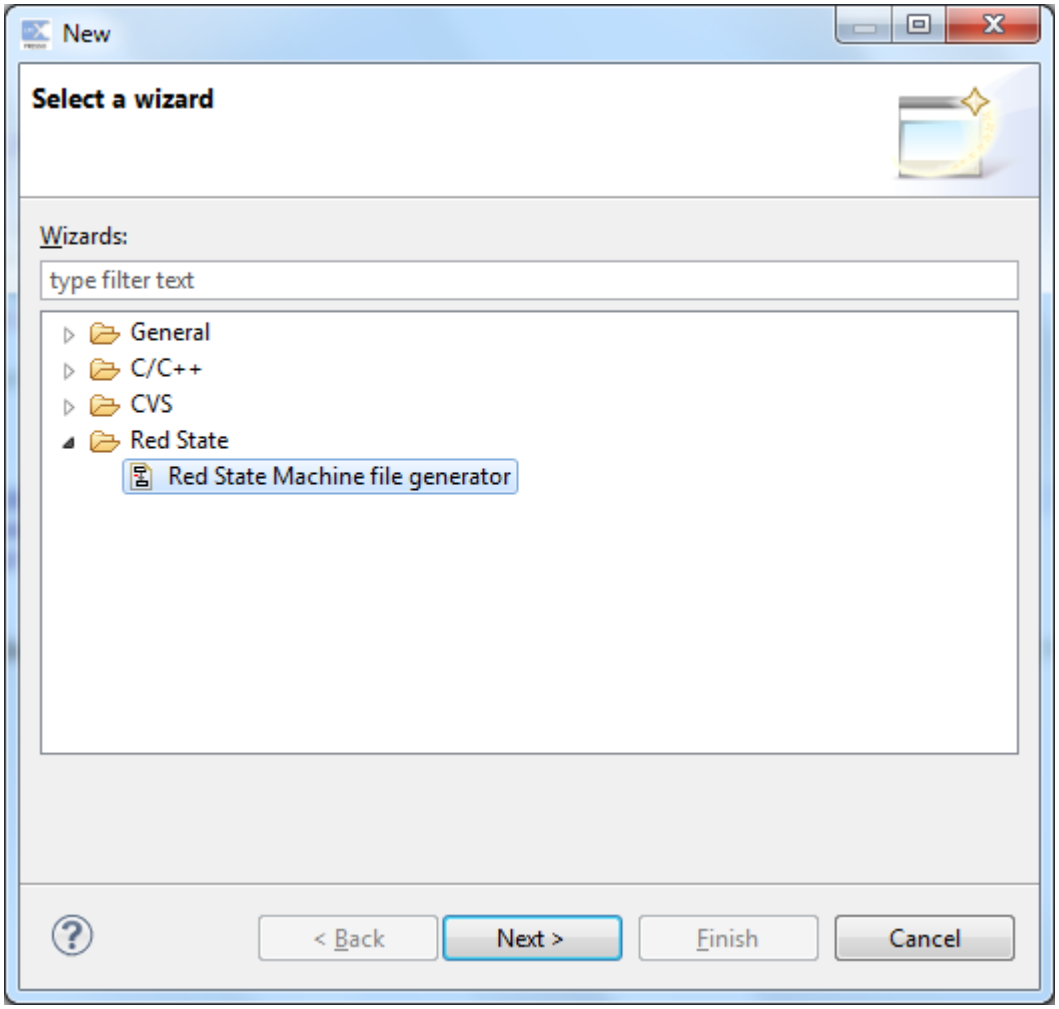





Develop - SCT_exercsise/src/main.c - LPCXpresso
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Develop
Projec... Perip... Regist... Symb...
SCT_exercsise
Binaries
Includes
CMSIS_CORE_LPC8xx
src
cr_startup_lpc8xx.c
main.c
Debug
ATEST Debug.launch
ATEST Release.launch
Qui... (x)= Var... Br... Ou... Ex...
Start here
Import project(s)
New project...
Build all projects [Debug]
Build 'SCT_exercsise' [Debug]
Clean 'SCT_exercsise' [Debug]
Debug 'SCT_exercsise' [Debug]
Terminate, Build and Debug 'SCT_exercsise' [Debug]
main.c
3 Name : main.c
10
11 #ifdef __USE_CMSIS
12 #include "LPC8xx.h"
13 #endif
14
15 #include <cr_section_macros.h>
16 #include <NXP/crp.h>
17
18 // Variable to store CRP value in. Will be placed automatically
19 // by the linker when "Enable Code Read Protect" selected.
20 // See crp.h header for more information
21 __CRP const unsigned int CRP_WORD = CRP_NO_CRP ;
22
23 //#include "sct_fsm.h"
24
25 int main(void) {
26
27 LPC_SWM->PINASSIGN6 = 0x07ffffffUL; // assign SCTOUT_0 to P0_7
28 LPC_SYSCON->SYSAHBCLKCTRL |= (1 << 8); // enable the SCT clock
29 LPC_SYSCON->PRESETCTRL |= (1<< 8); // clear peripheral reset the SCT:
30
31 // sct_fsm_init(); // Initialize SCT
32 LPC_SCT->CTRL_L &= ~(1<< 2); // unhalt it: - clearing bit 2 of th
Console Problems Memory Instruction Trace SWO Trace Config Power Measurement ...
CDT Build Console [SCT_exercsise]
NXP LPC812 (SCT_exercsise)
Writable
Smart Insert
3 : 1

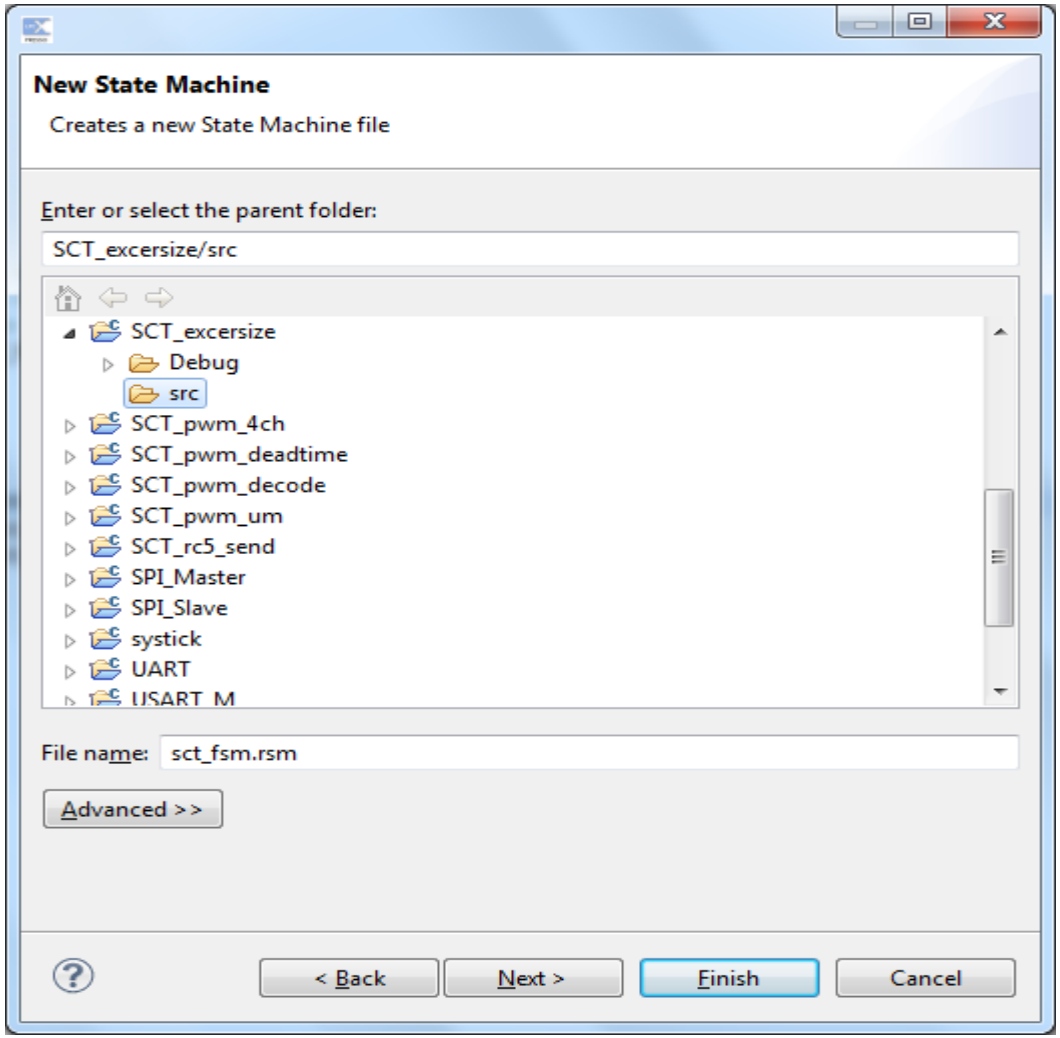
步骤 - 2

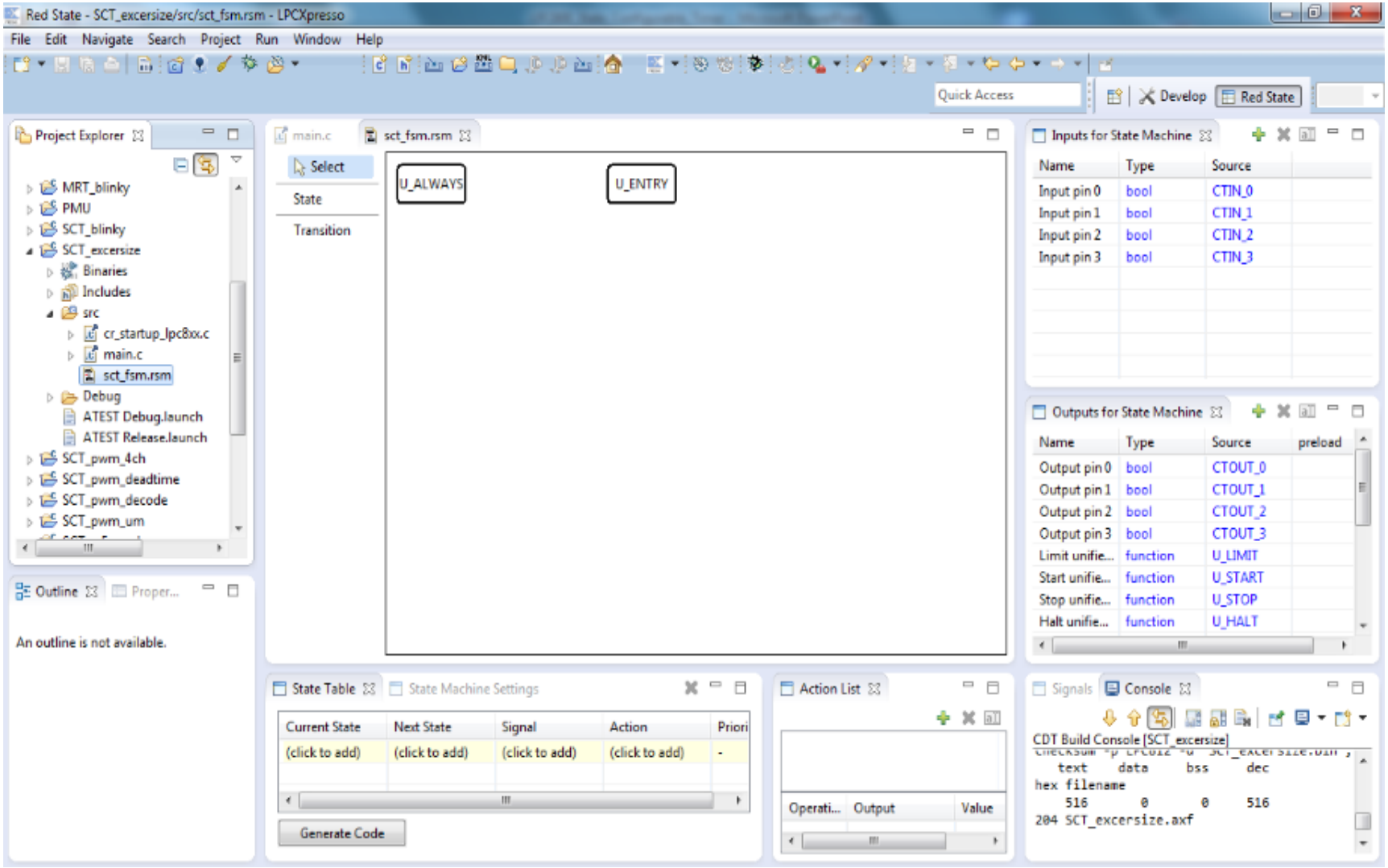
- 选中 file/new/other
- 选中 Red State . . .



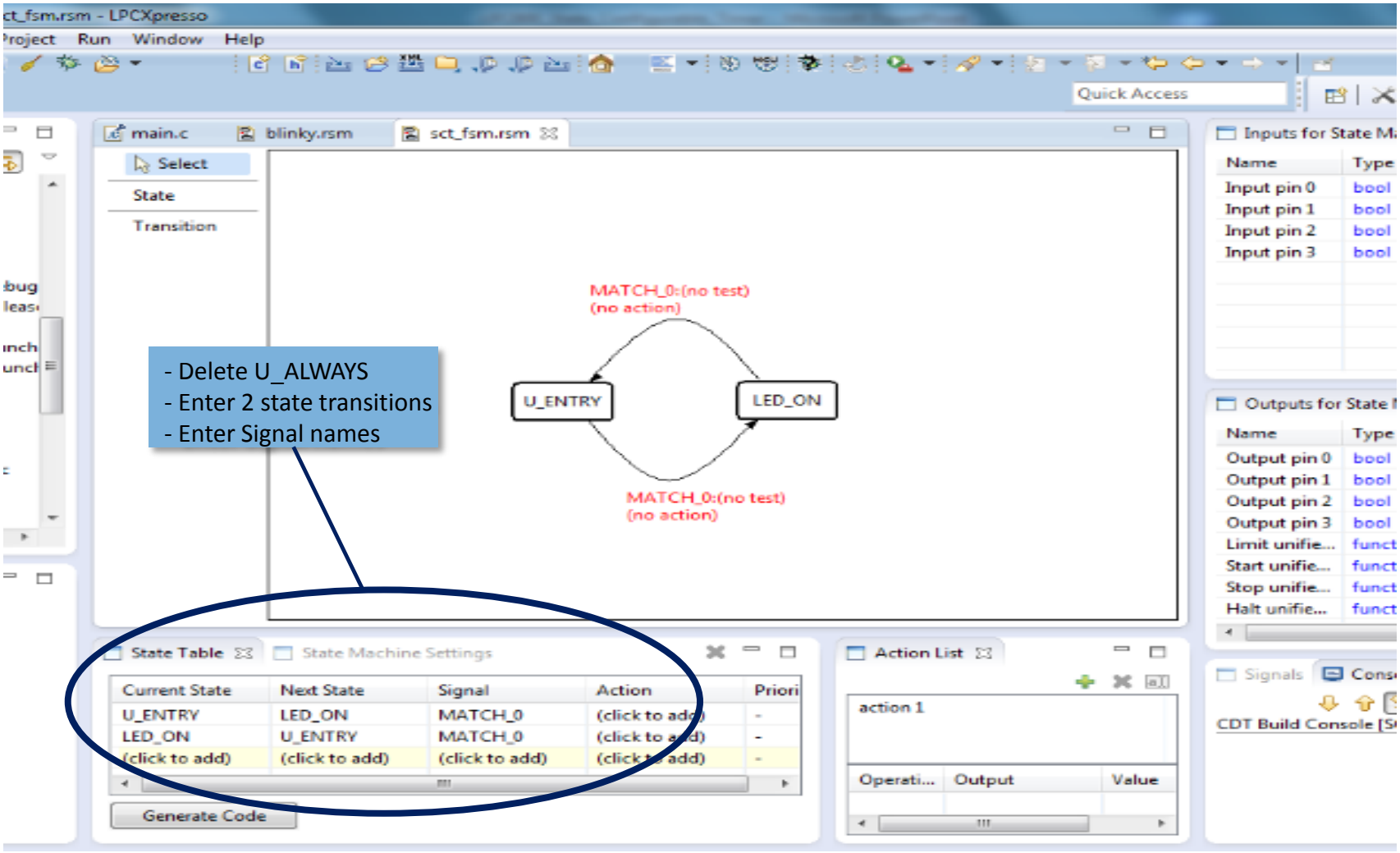
步骤 - 3

- 选中上一级目录
- 输入文件名
- 点击 Next
- 点击 Finish

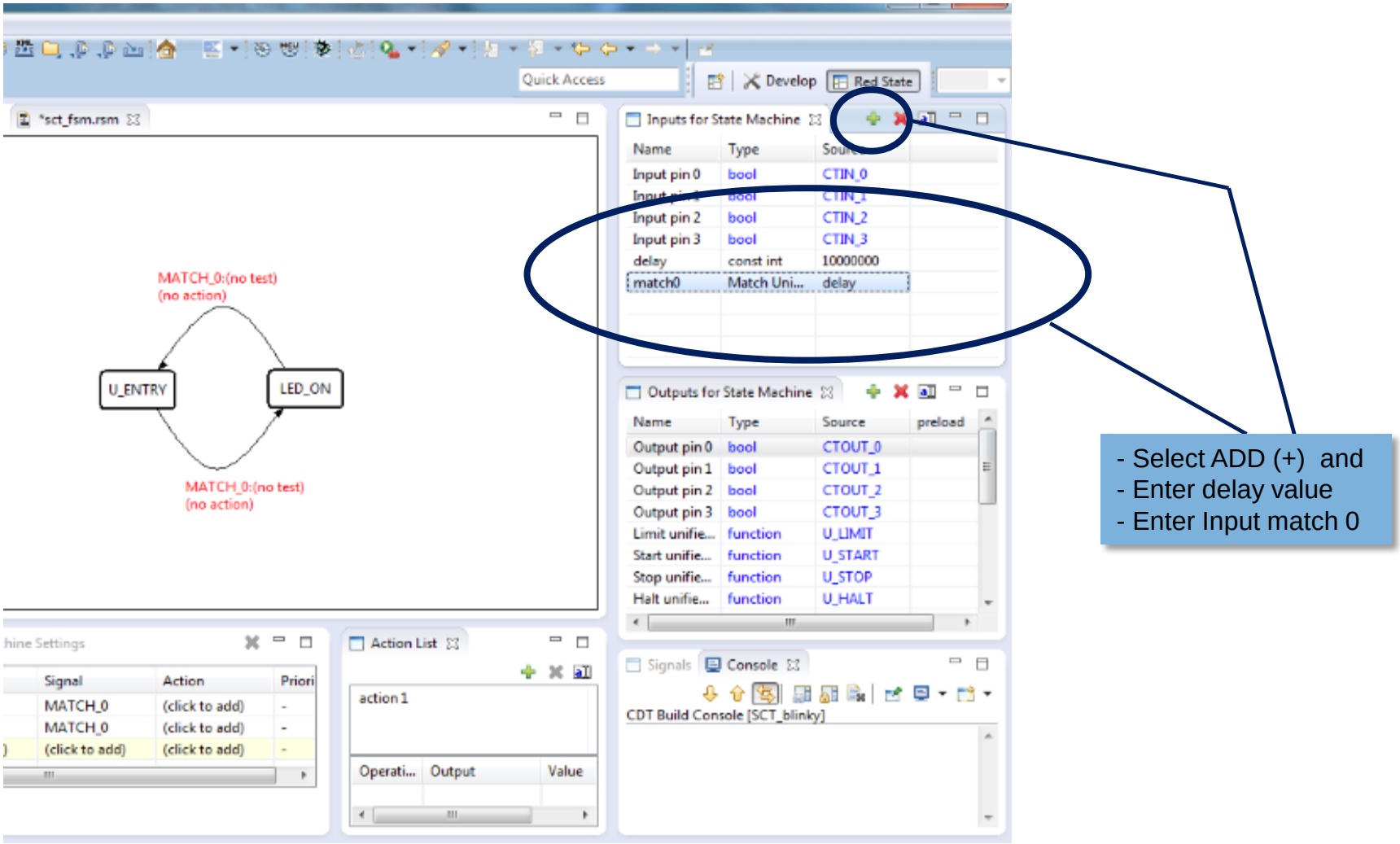



Red State - SCT_excersize/src/sct_fsm.rsm - LPCXpresso
File Edit Navigate Search Project Run Window Help
Quick Access
Develop
Red State
Project Explorer
MRT_blinky
PMU
SCT_blinky
SCT_excersize
Binaries
Includes
src
cr_startup_lpc8xx.c
main.c
sct_fsm.rsm
Debug
ATEST Debug.launch
ATEST Release.launch
SCT_pwm_4ch
SCT_pwm_deadtime
SCT_pwm_decode
SCT_pwm_um
Outline
Proper...
An outline is not available.
main.c
sct_fsm.rsm
Select
State
Transition
U_ALWAYS
U_ENTRY
Inputs for State Machine
Name Type Source
Input pin 0 bool CTIN_0
Input pin 1 bool CTIN_1
Input pin 2 bool CTIN_2
Input pin 3 bool CTIN_3
Outputs for State Machine
Name Type Source preload
Output pin 0 bool CTOUT_0
Output pin 1 bool CTOUT_1
Output pin 2 bool CTOUT_2
Output pin 3 bool CTOUT_3
Limit unifie... function U_LIMIT
Start unifie... function U_START
Stop unifie... function U_STOP
Halt unifie... function U_HALT
State Table
State Machine Settings
Current State Next State Signal Action Priori
(click to add) (click to add) (click to add) (click to add) -
Generate Code
Action List
Operati... Output Value
Signals
Console
CDT Build Console [SCT_excersize]
text data bss dec
hex filename
516 0 0 516
204 SCT_excersize.axf

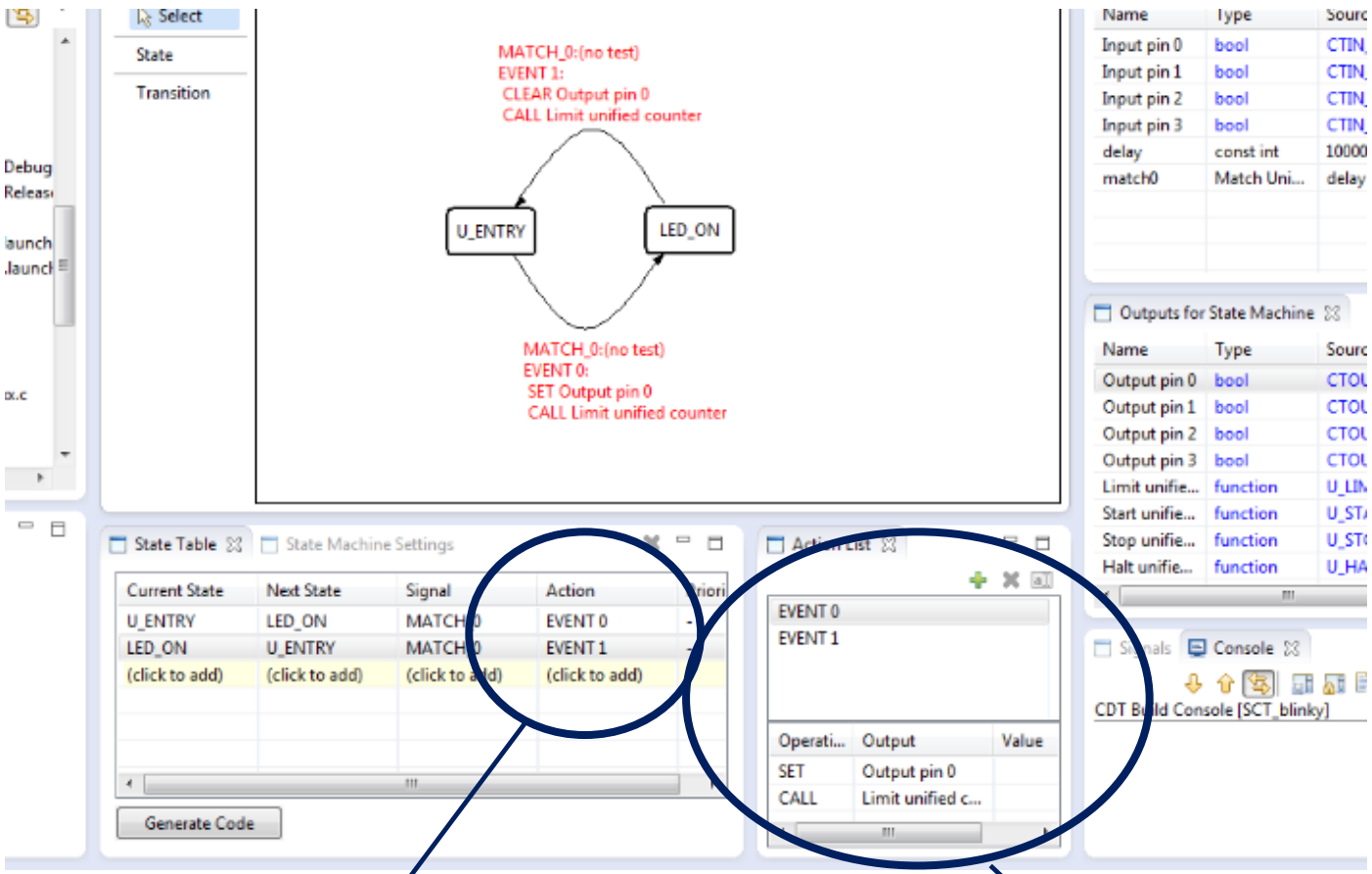
步驟 - 4



步骤 - 5



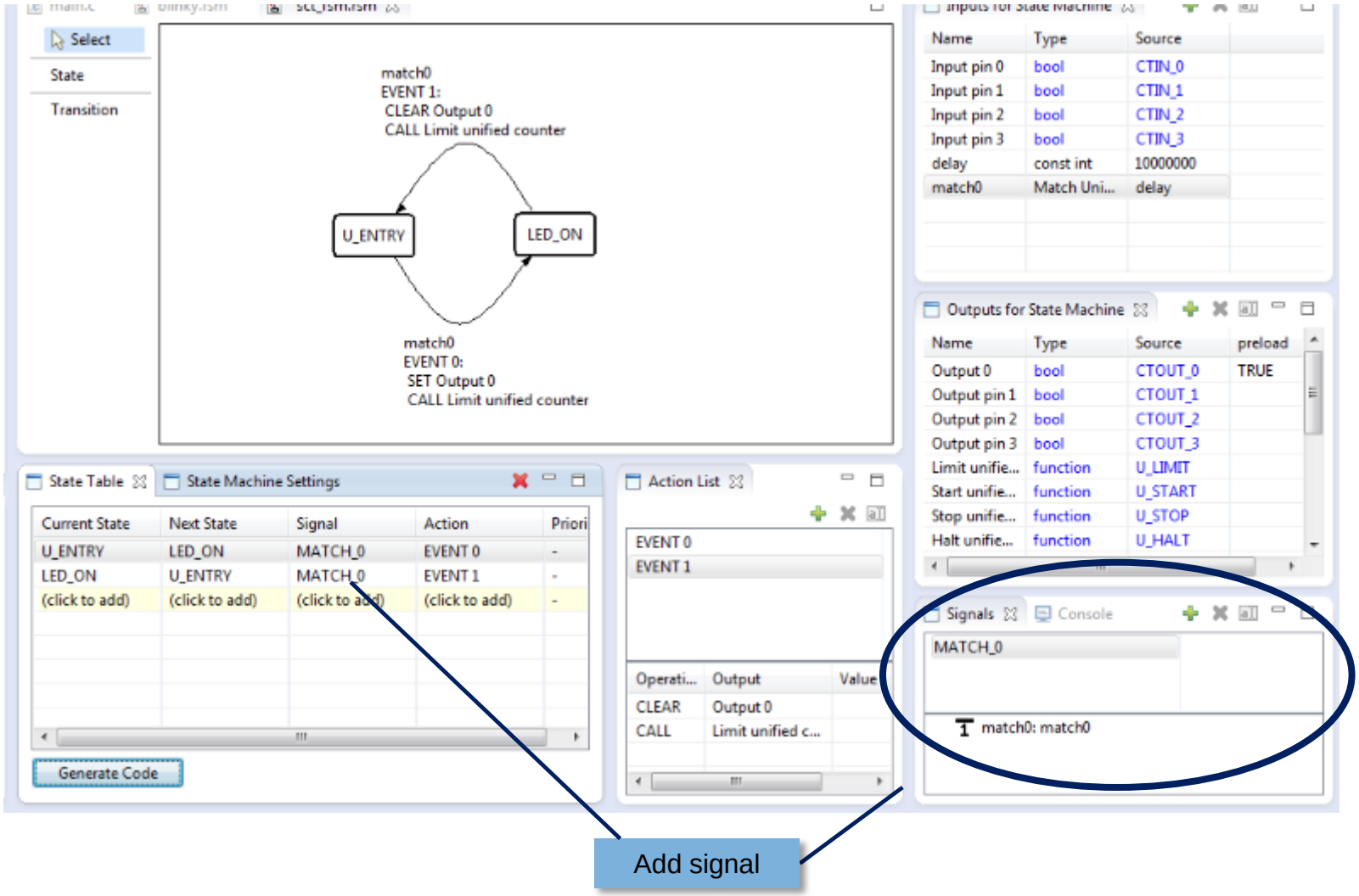
步骤 - 6



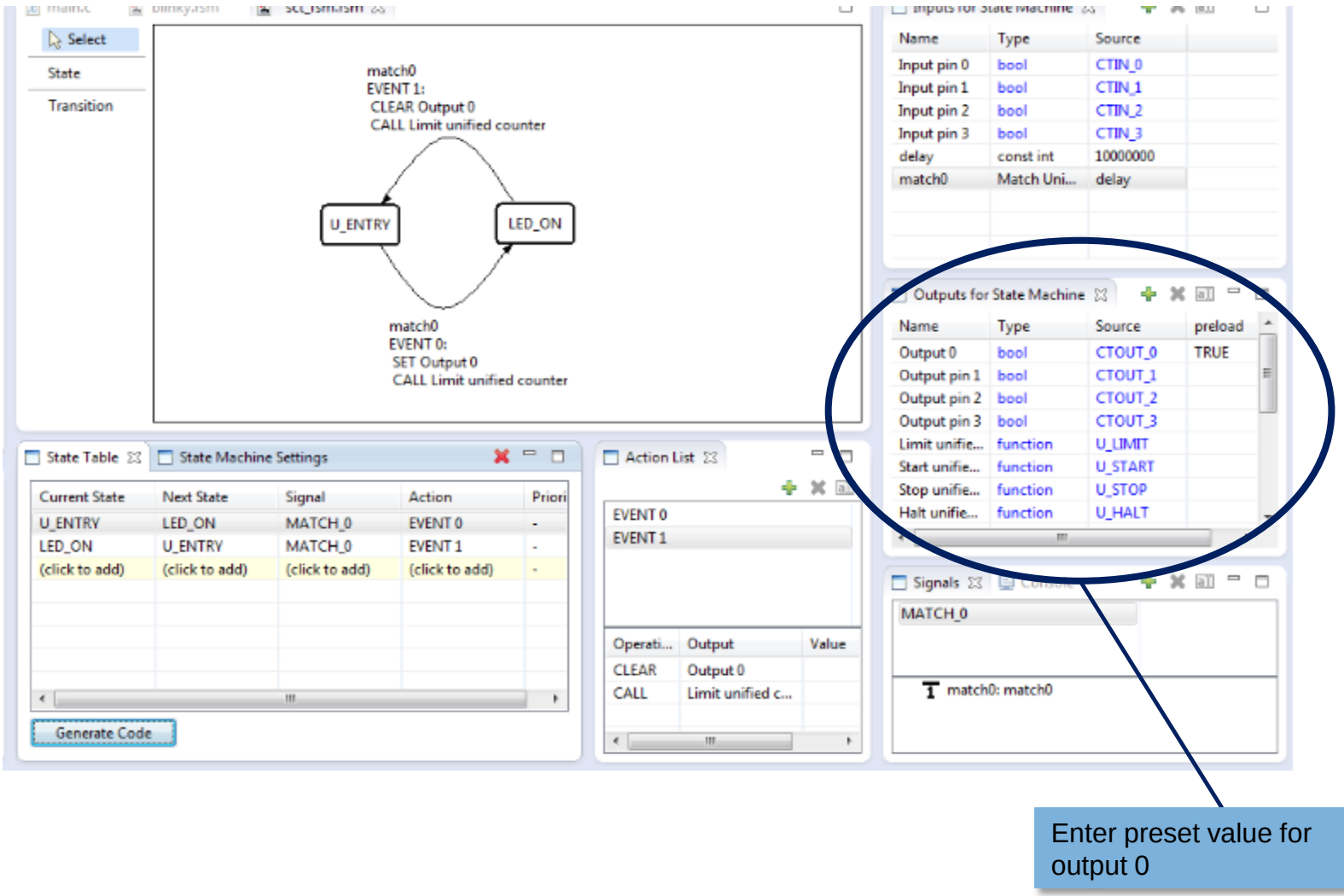
Add actions to the state transitions

- Add 2 Actions (EVENTS) and for
- EV0 operations SET and LIMIT
- EV1 operations CLEAR and LIMIT

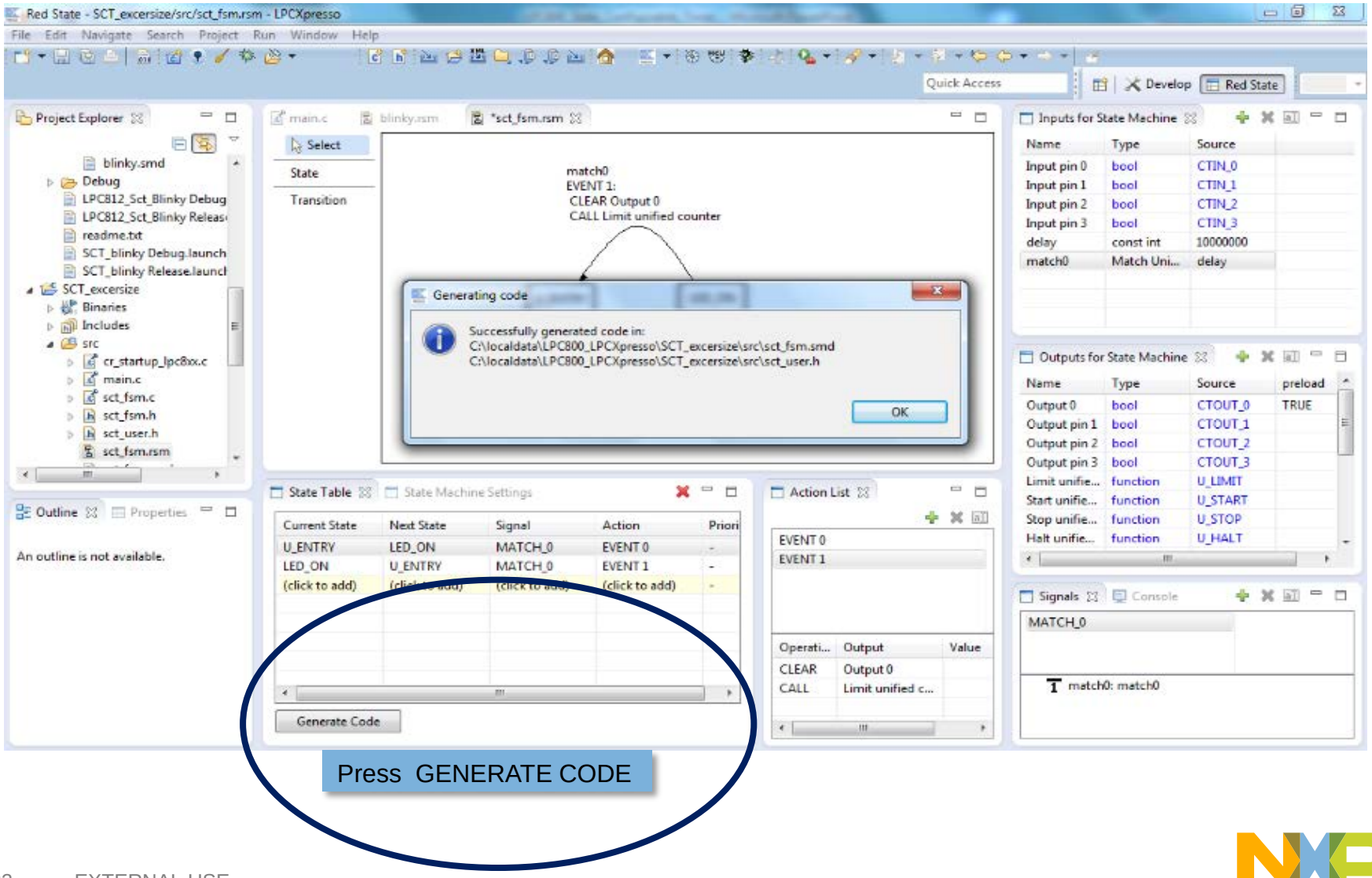
步骤 - 7



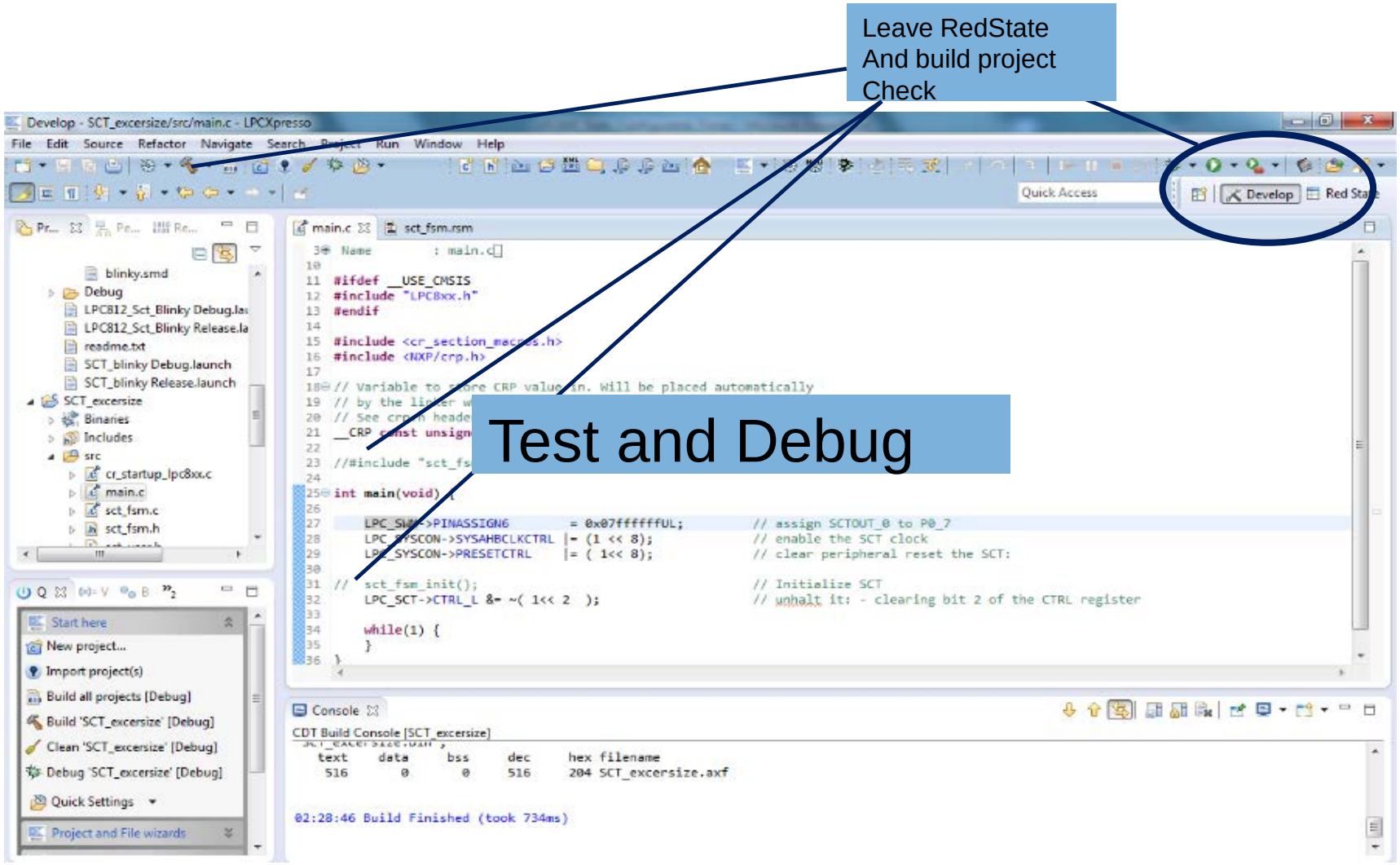
步骤 - 8



步驟 - 9


Red State - SCT_excersize/src/sct_fsm.rsm - LPCXpresso
File Edit Navigate Search Project Run Window Help
Quick Access
Develop
Red State
Project Explorer
blinky.smd
Debug
LPC812_Sct_Blinky Debug
LPC812_Sct_Blinky Releas
readme.txt
SCT_blinky Debug.launch
SCT_blinky Release.launcl
SCT_excersize
Binaries
Includes
src
cr_startup_lpc8xx.c
main.c
sct_fsm.c
sct_fsm.h
sct_user.h
sct_fsm.rsm
main.c
blinky.rsm
*sct_fsm.rsm
Select
State
Transition
match0
EVENT 1:
CLEAR Output 0
CALL Limit unified counter
Generating code
Successfully generated code in:
C:\localdata\LPC800_LPCXpresso\SCT_excersize\src\sct_fsm.smd
C:\localdata\LPC800_LPCXpresso\SCT_excersize\src\sct_user.h
OK
Inputs for State Machine
Name Type Source
Input pin 0 bool CTIN_0
Input pin 1 bool CTIN_1
Input pin 2 bool CTIN_2
Input pin 3 bool CTIN_3
delay const int 10000000
match0 Match Uni... delay
Outputs for State Machine
Name Type Source preload
Output 0 bool CTOUT_0 TRUE
Output pin 1 bool CTOUT_1
Output pin 2 bool CTOUT_2
Output pin 3 bool CTOUT_3
Limit unifie... function U_LIMIT
Start unifie... function U_START
Stop unifie... function U_STOP
Halt unifie... function U_HALT
Outline
Properties
An outline is not available.
State Table
State Machine Settings
Current State Next State Signal Action Priori
U_ENTRY LED_ON MATCH_0 EVENT 0
LED_ON U_ENTRY MATCH_0 EVENT 1
(click to add) (click to add)
Generate Code
Action List
EVENT 0
EVENT 1
Operati... Output Value
CLEAR Output 0
CALL Limit unified c...
Signals
Console
MATCH_0
match0: match0
Press GENERATE CODE

步骤 - 10


Leave RedState
And build project
Check
Test and Debug
Develop - SCT_excersize/src/main.c - LPCXpresso
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Develop
main.c
sct_fsm.rsm
blinky.smd
Debug
LPC812_Sct_Blinky Debug.la
LPC812_Sct_Blinky Release.la
readme.txt
SCT_blinky Debug.launch
SCT_blinky Release.launch
SCT_excersize
Binaries
Includes
src
cr_startup_lpc8xx.c
main.c
sct_fsm.c
sct_fsm.h
Start here
New project...
Import project(s)
Build all projects [Debug]
Build 'SCT_excersize' [Debug]
Clean 'SCT_excersize' [Debug]
Debug 'SCT_excersize' [Debug]
Quick Settings
Project and File wizards
Name : main.c
#ifdef __USE_CMSIS
#include "LPC8xx.h"
#endif
#include <cr_section_macros.h>
#include <NXP/crp.h>
// Variable to store CRP value in. Will be placed automatically
__CRP const unsign
//#include "sct_fs
int main(void) {
LPC_SWM->PINASSIGN6 = 0x07ffffffUL; // assign SCTOUT_0 to P0_7
LPC_SYSCON->SYSAHBCLKCTRL |= (1 << 8); // enable the SCT clock
LPC_SYSCON->PRESETCTRL |= (1<< 8); // clear peripheral reset the SCT:
// sct_fsm_init(); // Initialize SCT
LPC_SCT->CTRL_L &= ~(1<< 2); // unhalt it: - clearing bit 2 of the CTRL register
while(1) {
}
}
Console
CDT Build Console [SCT_excersize]
text data bss dec hex filename
516 0 0 516 204 SCT_excersize.axf
02:28:46 Build Finished (took 734ms)

SECURE CONNECTIONS
FOR A SMARTER WORLD