Hercules™ TMS570LS31x Microcontrollers

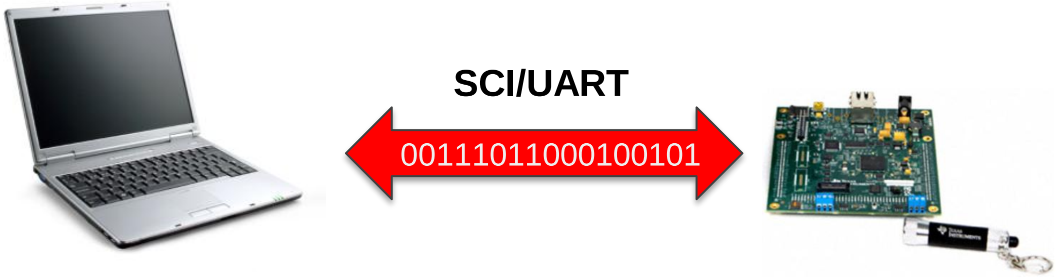
MibADC Example

Overview

- In this example we will:
  - Create a TMS570LS31x HALCoGen Project
  - Generate and import code into Code Composer Studio
  - Write code to take analog to digital (ADC) conversions from the ambient light sensor on the development board (USB Stick or HDK) and display the conversion values to a PC using the SCI (UART) port.
  - Build, deploy and execute the code to the microcontroller

- Required Hardware:
  - Windows Based PC (WinXP, Vista, 7)
  - TMS570LS31x USB Development Stick or TMS570LS31x Hercules Development Kit

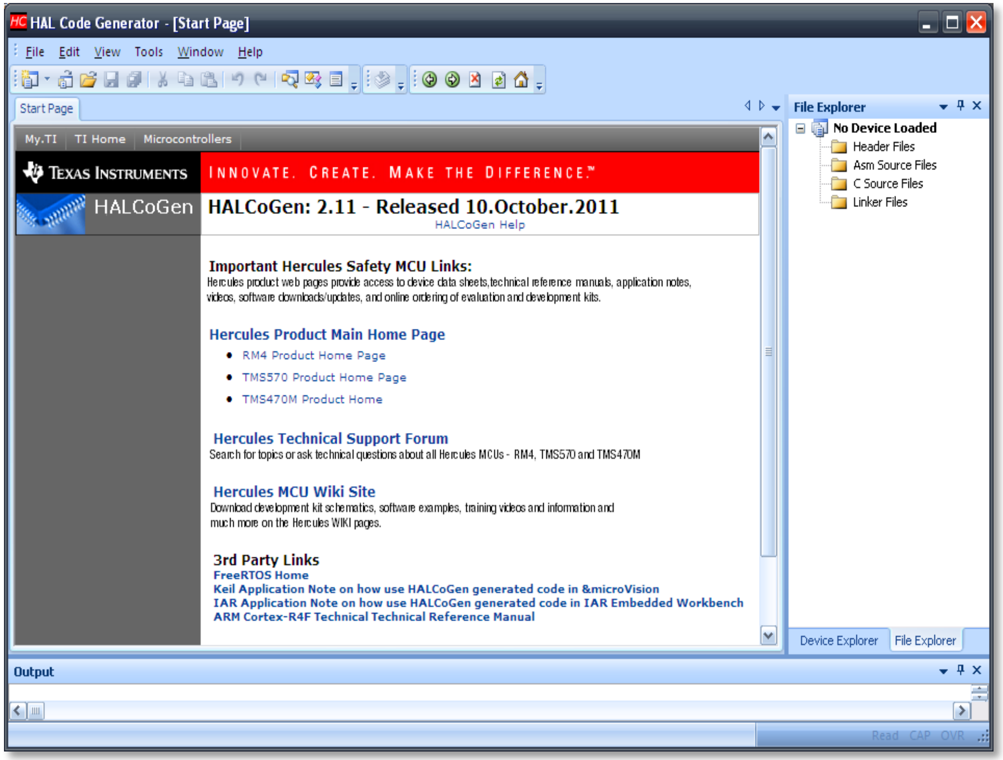
- Required Software:
  - HALCoGen v2.11 *or higher*
  - Code Composer Studio v4.x



Setting up a New HALCoGen Project

- To launch HALCoGen go to:

start → Programs → Texas Instruments → Hercules → HALCoGen



HALCoGen: 2.11 - Released 10.October.2011

HALCoGen Help

Important Hercules Safety MCU Links:

Hercules product web pages provide access to device data sheets,technical reference manuals, application notes, videos, software downloads/updates, and online ordering of evaluation and development kits.

Hercules Product Main Home Page

- RM4 Product Home Page
- TMS570 Product Home Page
- TMS470M Product Home

Hercules Technical Support Forum

Search for topics or ask technical questions about all Hercules MCUs - RM4, TMS570 and TMS470M

Hercules MCU Wiki Site

Download development kit schematics, software examples, training videos and information and much more on the Hercules WIKI pages.

3rd Party Links

FreeRTOS Home
Keil Application Note on how use HALCoGen generated code in &microVision
IAR Application Note on how use HALCoGen generated code in IAR Embedded Workbench
ARM Cortex-R4F Technical Technical Reference Manual

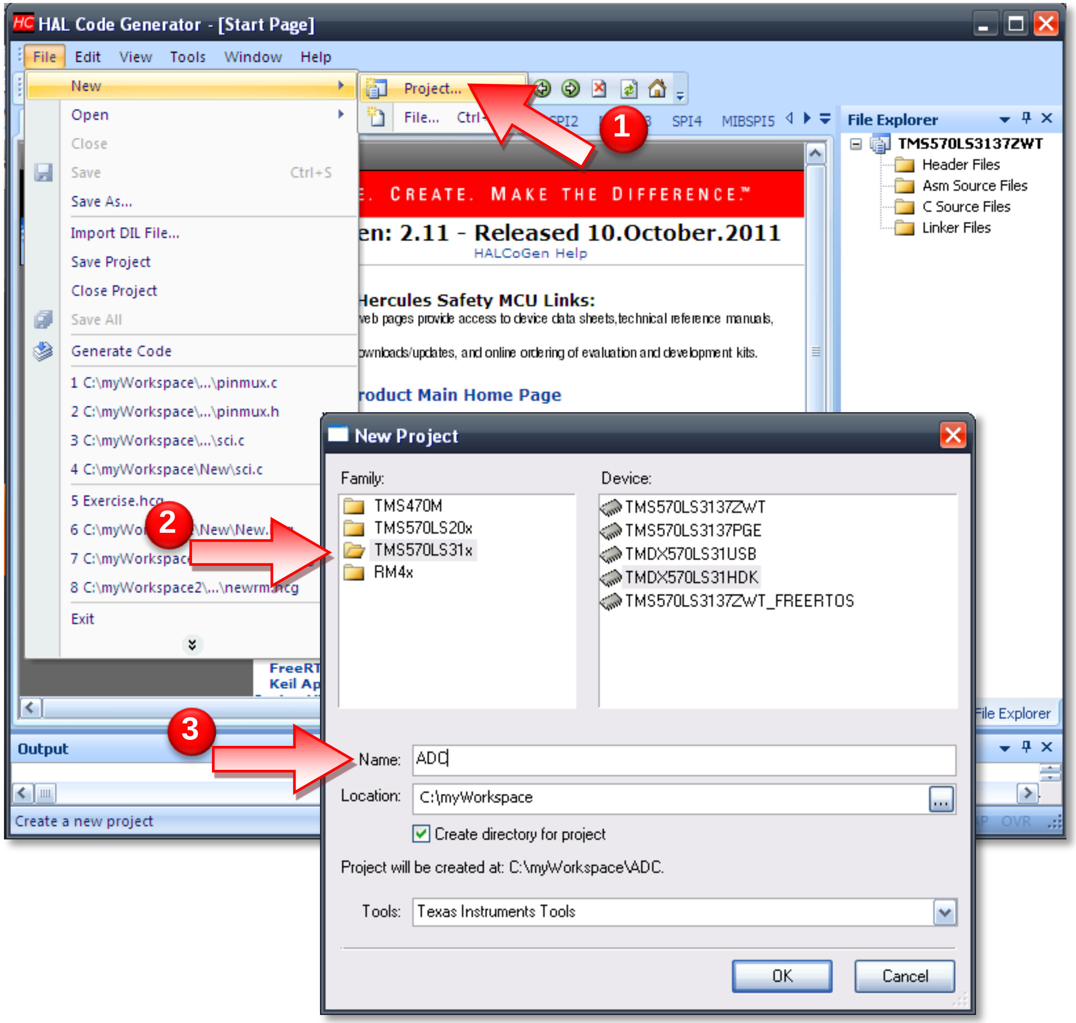
For more help with HALCoGen, see this getting started video: LINK

Setting up a New HALCoGen Project

- To create a new HALCoGen project:
  - File → New → Project

- **For the TMS570 USB Stick:**
  - Choose Family: TMDX570LS31x
  - Device:TMDX570LS31USB

- **For the TMS570 HDK:**
  - Choose Family: TMDX570LS31x
  - Device:TMDXTMS570LS31HDK



HAL Code Generator - [Start Page]
File
Edit
View
Tools
Window
Help
New
Project...
File...
Ctrl+
Open
Close
Save
Ctrl+S
Save As...
Import DIL File...
Save Project
Close Project
Save All
Generate Code
1 C:\myWorkspace\...\pinmux.c
2 C:\myWorkspace\...\pinmux.h
3 C:\myWorkspace\...\sci.c
4 C:\myWorkspace\New\sci.c
5 Exercise.hcg
6 C:\myWo...\New\New...
7 C:\myWorkspac...
8 C:\myWorkspace2\...\newrm...hcg
Exit
1
2
3
SPI2
SPI4
MIBSPI5
E. CREATE. MAKE THE DIFFERENCE.™
en: 2.11 - Released 10.October.2011
HALCoGen Help
Hercules Safety MCU Links:
veb pages provide access to device data sheets,technical reference manuals,
ownloads/updates, and online ordering of evaluation and development kits.
roduct Main Home Page
FreeRT
Keil Ap
Output
Create a new project
File Explorer
TMS570LS31372WT
Header Files
Asm Source Files
C Source Files
Linker Files
OVR
OK
Cancel

New Project

Family:
- TMS470M
- TMS570LS20x
- TMS570LS31x
- RM4x

Device:
- TMS570LS3137ZWT
- TMS570LS3137PGE
- TMDX570LS31USB
- TMDX570LS31HDK
- TMS570LS3137ZWT_FREERTOS

Name: ADC

Location: C:\myWorkspace

☑ Create directory for project

Project will be created at: C:\myWorkspace\ADC.

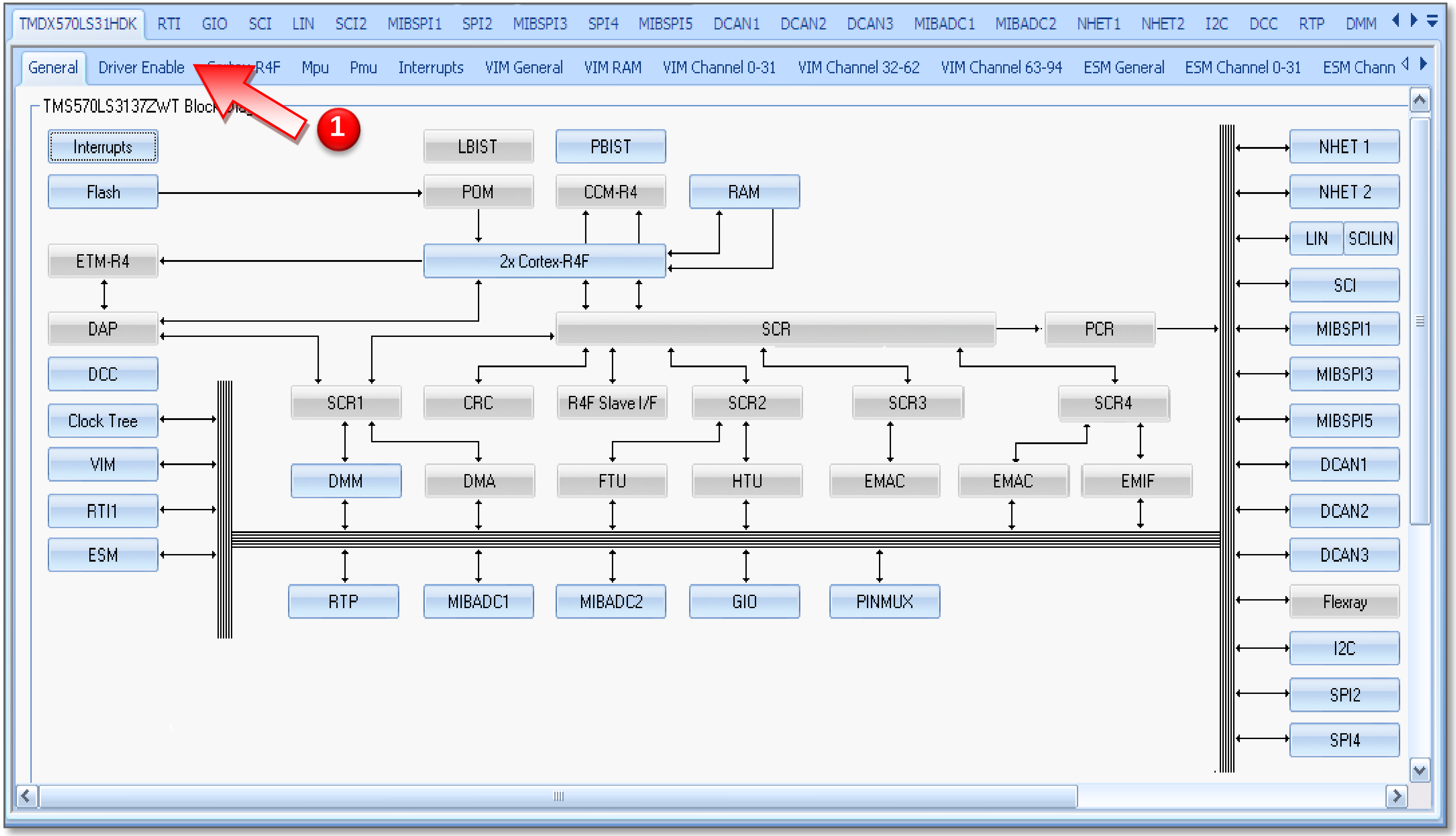
Tools: Texas Instruments Tools

- Then define a name: ‘ADC’
- Location: “C:\myWorkspace”

Module Selection

- Drivers, Modules and Peripherals can be selected for configuration by either the module selection ribbon or by clicking on the module in the block diagram
- The next step is to enable the appropriate drivers for this example

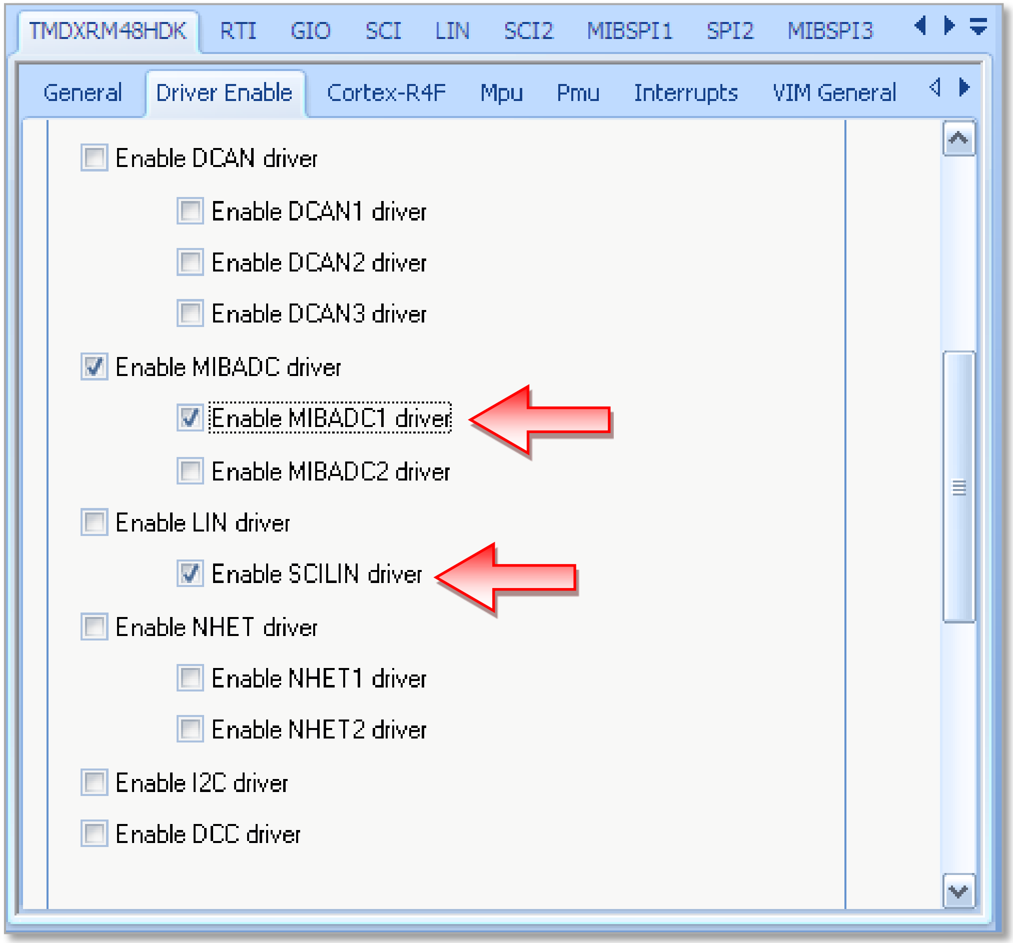
Ribbon



Block Diagram

Enabling Perphierial Drivers

- Enable the SCILIN and MIBADC1 drivers in the ‘Driver Enable’ sub tab of the device



□ Enable DCAN driver
- □ Enable DCAN1 driver
- □ Enable DCAN2 driver
- □ Enable DCAN3 driver

☑ Enable MIBADC driver
- ☑ Enable MIBADC1 driver
- □ Enable MIBADC2 driver

□ Enable LIN driver
- ☑ Enable SCILIN driver

□ Enable NHET driver
- □ Enable NHET1 driver
- □ Enable NHET2 driver

□ Enable I2C driver

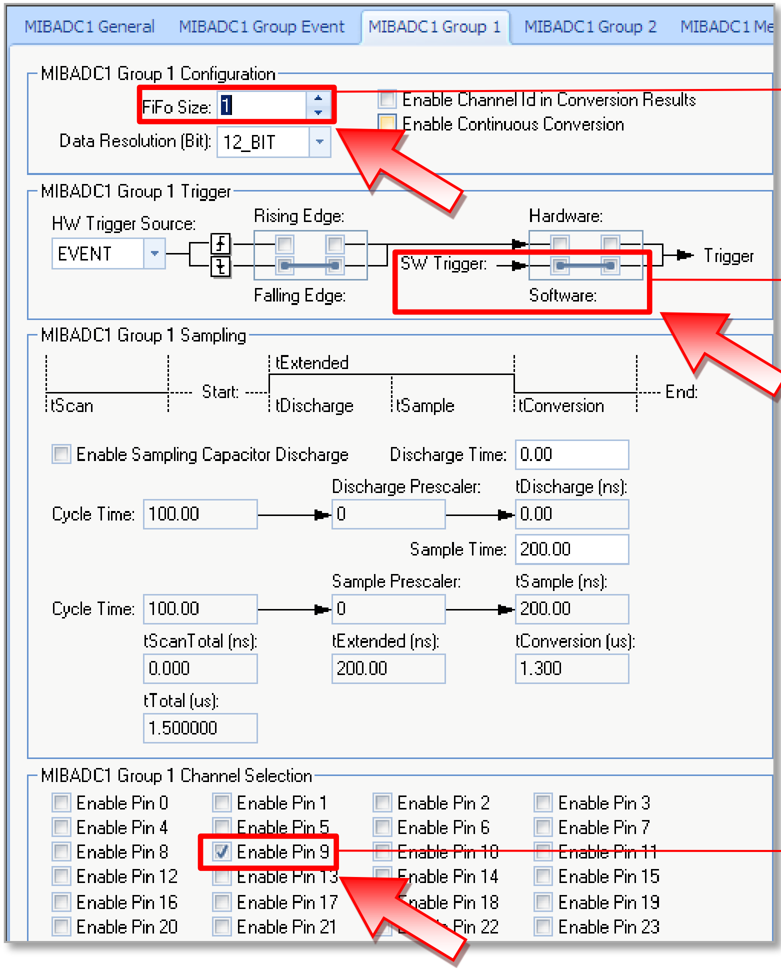
□ Enable DCC driver

WARNING:

Make sure all other peripheral drivers are disabled

ADC Group Configuration

- From the MIBADC1 Group1 tab, configure FIFO Size, Trigger Source and ADC channel as shown below



MIBADC1 Group 1 Configuration

FiFo Size: 1

□ Enable Channel Id in Conversion Results

□ Enable Continuous Conversion

Data Resolution (Bit): 12_BIT

MIBADC1 Group 1 Trigger

HW Trigger Source: EVENT

MIBADC1 Group 1 Sampling

□ Enable Sampling Capacitor Discharge

Discharge Time: 0.00

Discharge Prescaler: 0

tDischarge (ns): 0.00

Cycle Time: 100.00

Sample Time: 200.00

Sample Prescaler: 0

tSample (ns): 200.00

Cycle Time: 100.00

tScanTotal (ns): 0.000

tExtended (ns): 200.00

tConversion (us): 1.300

tTotal (us): 1.500000

MIBADC1 Group 1 Channel Selection

□ Enable Pin 0	□ Enable Pin 1	□ Enable Pin 2	□ Enable Pin 3
□ Enable Pin 4	□ Enable Pin 5	□ Enable Pin 6	□ Enable Pin 7
□ Enable Pin 8	☑ Enable Pin 9	□ Enable Pin 10	□ Enable Pin 11
□ Enable Pin 12	□ Enable Pin 13	□ Enable Pin 14	□ Enable Pin 15
□ Enable Pin 16	□ Enable Pin 17	□ Enable Pin 18	□ Enable Pin 19
□ Enable Pin 20	□ Enable Pin 21	□ Enable Pin 22	□ Enable Pin 23

Only one conversion per software trigger is done

SW Trigger as Trigger Source

Ambient light sensor is connected to MibADC1 Pin9

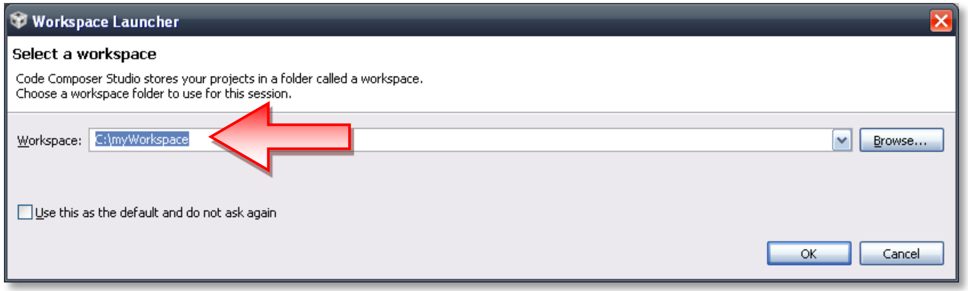
- Generate code: File → Generate Code or click

Setting up Code Composer Studio 4

- Launch Code Composer Studio v4.x (CCS)

→ Programs → Texas Instruments → Code Composer Studio v4.x → Code Composer Studio v4

- When it launches, CCS will ask you to select a workspace, we will chose “C:\myWorkspace”



Workspace Launcher

Select a workspace

Code Composer Studio stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: C:\myWorkspace

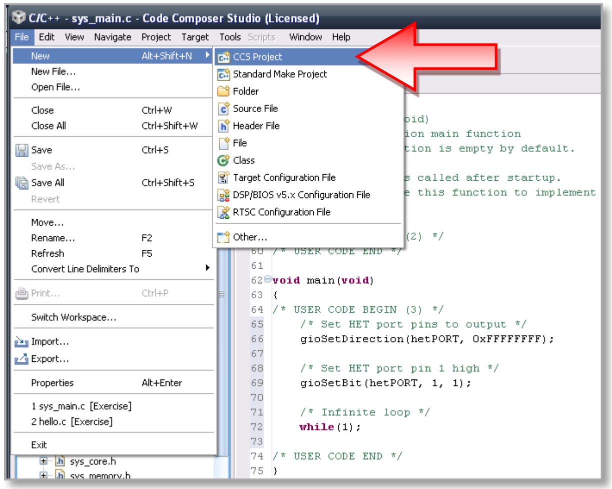
□ Use this as the default and do not ask again

WARNING:

Make sure to define the same ‘Workspace’ as defined in the HALCoGen setup

- Once CCS loads, go to

  File → New → CCS Project



C/C++ - sys_main.c - Code Composer Studio (Licensed)

```
oid)
ion main function
tion is empty by default.

s called after startup.
e this function to implement

(2) */
/* USER CODE END */

void main(void)
{
/* USER CODE BEGIN (3) */
    /* Set HET port pins to output */
    gioSetDirection(hetPORT, 0xFFFFFFFF);

    /* Set HET port pin 1 high */
    gioSetBit(hetPORT, 1, 1);

    /* Infinite loop */
    while(1);

/* USER CODE END */
}
```

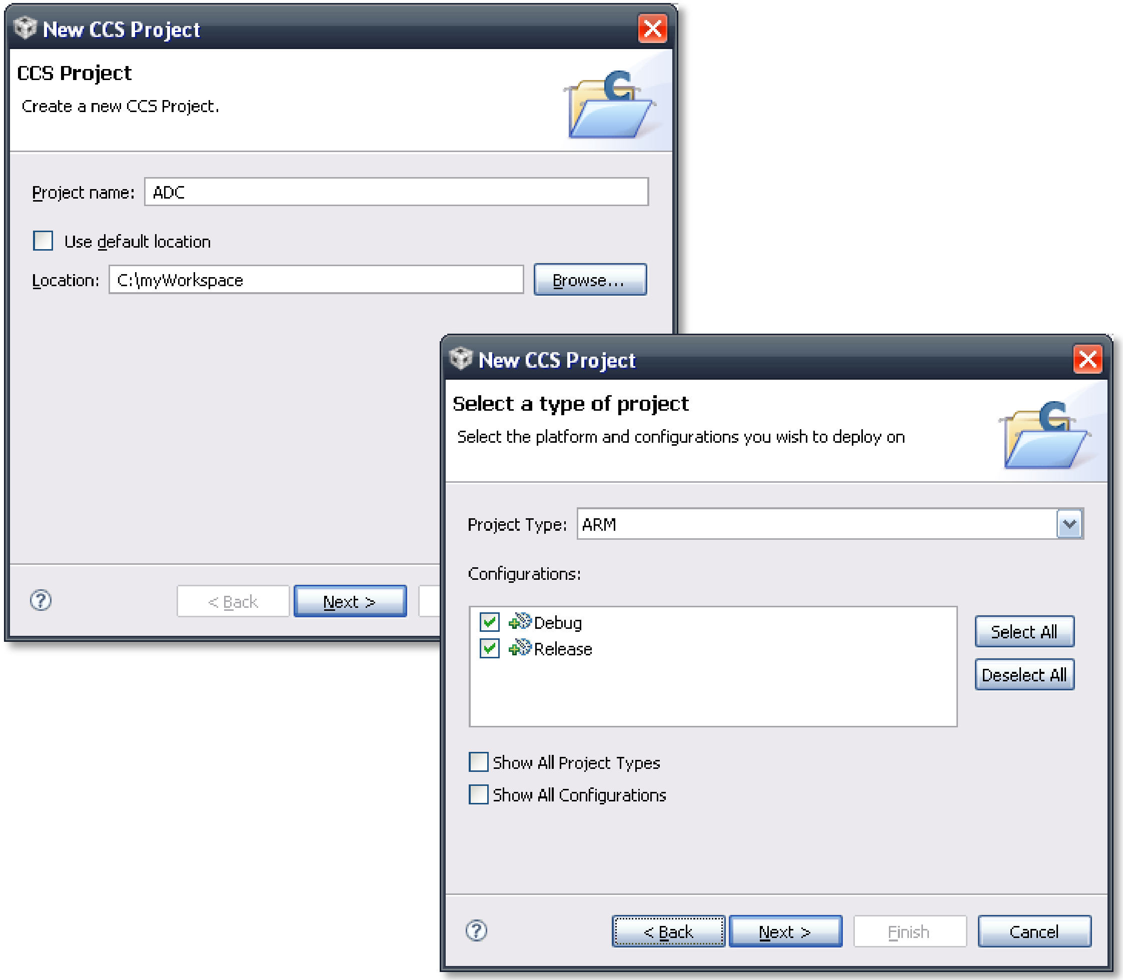
Setting up the CCS Project

- Our project name needs to match the name of our HALCoGen Project: ADC

- Then Click “next”

- On the next page, make sure that your project type is set to ARM and Debug and Release are both checked

- Then Click “next”



New CCS Project

CCS Project

Create a new CCS Project.

Project name: ADC

□ Use default location

Location: C:\myWorkspace

New CCS Project

Select a type of project

Select the platform and configurations you wish to deploy on

Project Type: ARM

Configurations:

☑ Debug

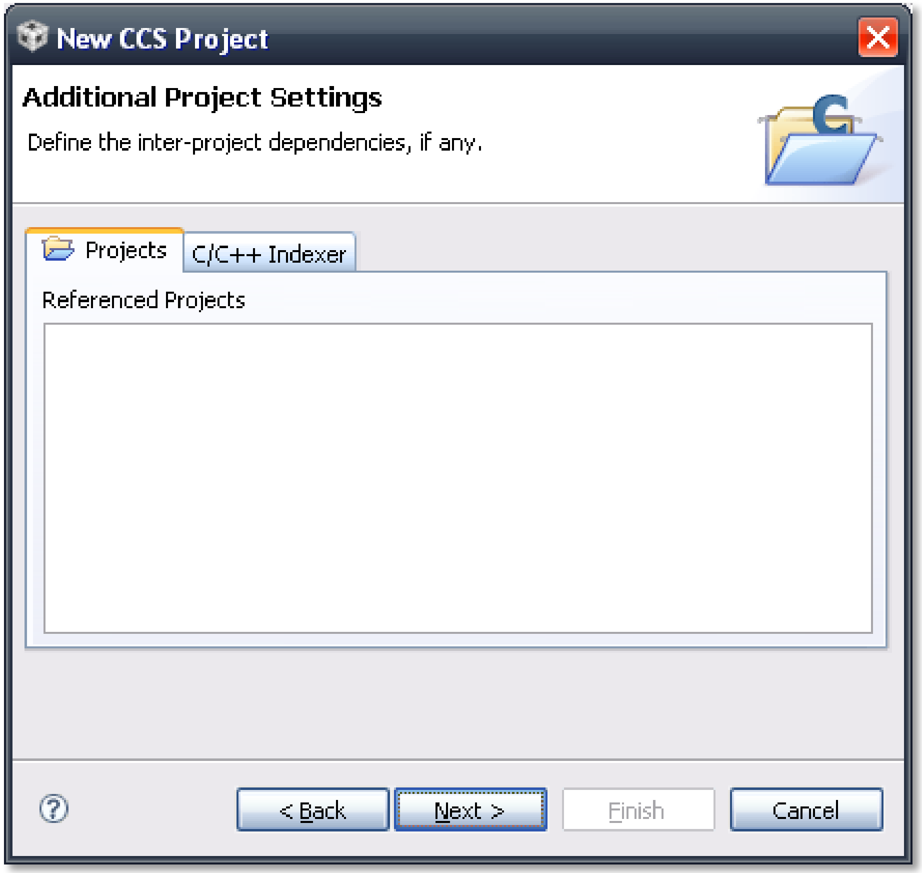
☑ Release

□ Show All Project Types

□ Show All Configurations

Setting up the CCS Project

- We are not using any referenced projects so click “next” again



New CCS Project

Additional Project Settings

Define the inter-project dependencies, if any.

Referenced Projects

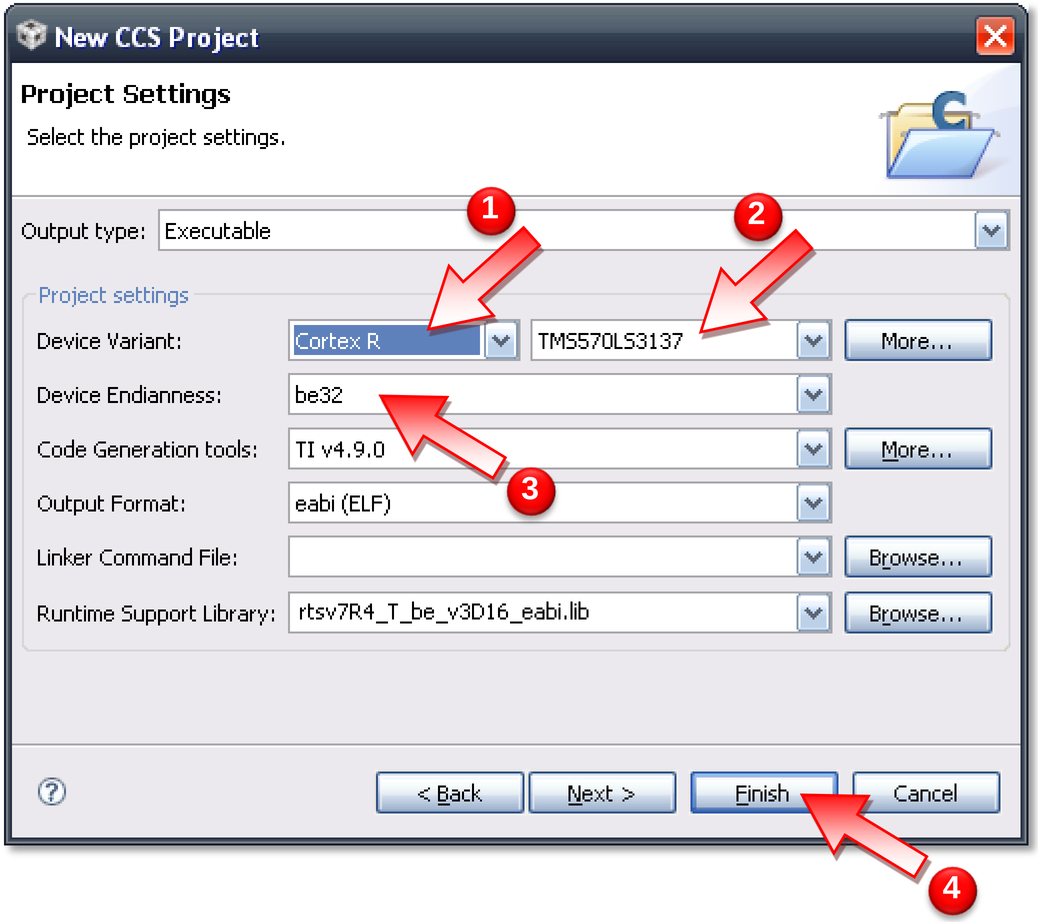
Setting up the CCS Project

- Next, set the Device Variant to “Cortex R”

- Choose: TMS570LS3137

- Device Endianness: be32

- Then click ‘Finish’



New CCS Project

Project Settings

Select the project settings.

Output type: Executable

Project settings

Device Variant: Cortex R | TMS570LS3137

Device Endianness: be32

Code Generation tools: TI v4.9.0

Output Format: eabi (ELF)

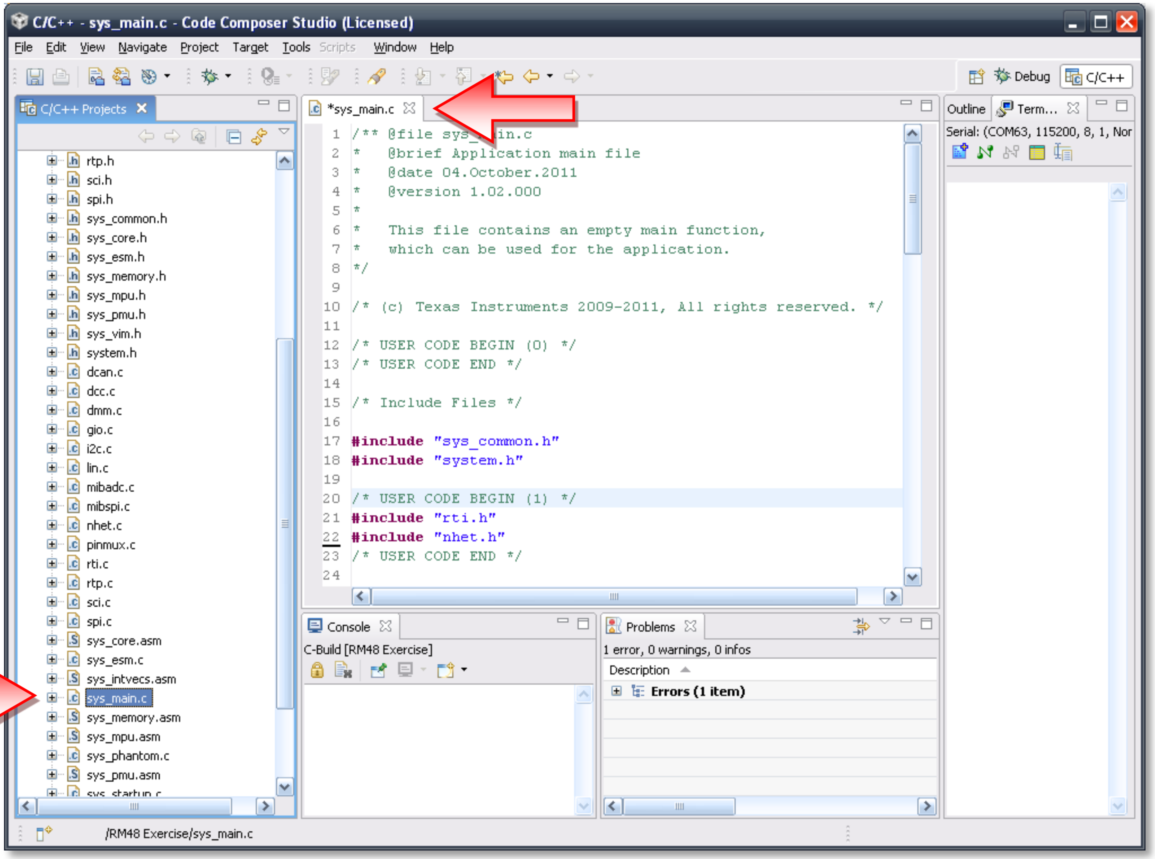
Linker Command File:

Runtime Support Library: rtsv7R4_T_be_v3D16_eabi.lib

Entering Code into the CCS Project

- Expand the project and open the “sys_main.c” file from the CCS “Project Explorer”

Note: Since the HALCoGen and CCS project have the same project name, all the startup and driver files generated by HALCoGen will be automatically included in the CCS project. Use the “sys_main.c” file for your main program code.



```
/** @file sys_main.c
*   @brief Application main file
*   @date 04.October.2011
*   @version 1.02.000
*
*   This file contains an empty main function,
*   which can be used for the application.
*/

/* (c) Texas Instruments 2009-2011, All rights reserved. */

/* USER CODE BEGIN (0) */
/* USER CODE END */

/* Include Files */

#include "sys_common.h"
#include "system.h"

/* USER CODE BEGIN (1) */
#include "rti.h"
#include "nhet.h"
/* USER CODE END */

```

Serial: (COM63, 115200, 8, 1, Nor

C-Build [RM48 Exercise]

1 error, 0 warnings, 0 infos

Errors (1 item)

/RM48 Exercise/sys_main.c

Entering Code into the CCS Project

- Insert the following code in the corresponding sections within the ‘sys_main.c’ file
  - **USER CODE BEGIN (0) - #include header section**

```
/* USER CODE BEGIN (0) */
#include "sci.h"
#include "mibadc.h"
#include "stdlib.h"
unsigned char command[8];
/* USER CODE END */
```

Entering Code into the CCS Project

– USER CODE BEGIN(3) - Main() section

```
void main(void)
{/* USER CODE BEGIN (3) */
  mibadcData_t adc_data; //ADC Data Structure
  mibadcData_t *adc_data_ptr = &adc_data; //ADC Data Pointer
  unsigned int NumberOfChars, value; //Declare variables

  sciInit(); //Initializes the SCI (UART) module
  mibadcInit(); //Initializes the ADC module

  while(1)  // Loop to acquire and send ADC sample data via the SCI (UART)
  {
    mibadcStartConversion(mibadcREG1, 1U);  //Start ADC conversion
    while(!mibadcIsConversionComplete(mibadcREG1, 1U)); //Wait for ADC conversion
    mibadcGetData(mibadcREG1, 1U, adc_data_ptr); //Store conversion into ADC pointer
    value = (unsigned int)adc_data_ptr->value;
    NumberOfChars = ltoa(value,(char *)command);
    sciSend(scilinREG, 2, (unsigned char *)"0x"); //Sends '0x' hex designation chars
    sciSend(scilinREG, NumberOfChars, command); //Sends the ambient light sensor data
    sciSend(scilinREG, 2, (unsigned char *)"\r\n"); //Sends new line character
  }
/* USER CODE END */}
```

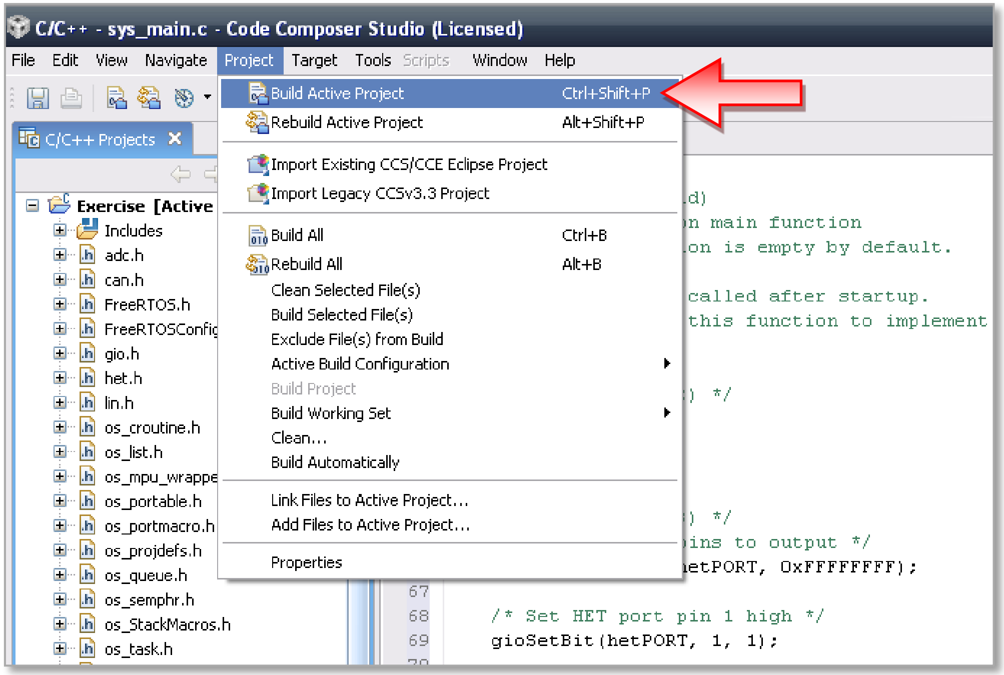
Entering Code into the CCS Project

– USER CODE (4) - Interrupt notification section

```
/* USER CODE BEGIN (4) */
/* Notification functions not used, but required by compiler*/
void mibadcNotification(mibadcBASE_t *adc, unsigned group)
{
    return;
}
void sciNotification(sciBASE_t *sci, unsigned flags)
{
    return;
}
void esmGroup1Notification(int bit)
{
    return;
}
void esmGroup2Notification(int bit)
{
    return;
}
/* USER CODE END */
```

Compiling the Project

- The code is now complete and we are ready to build our project.
  - Go to Project → Build Active Project

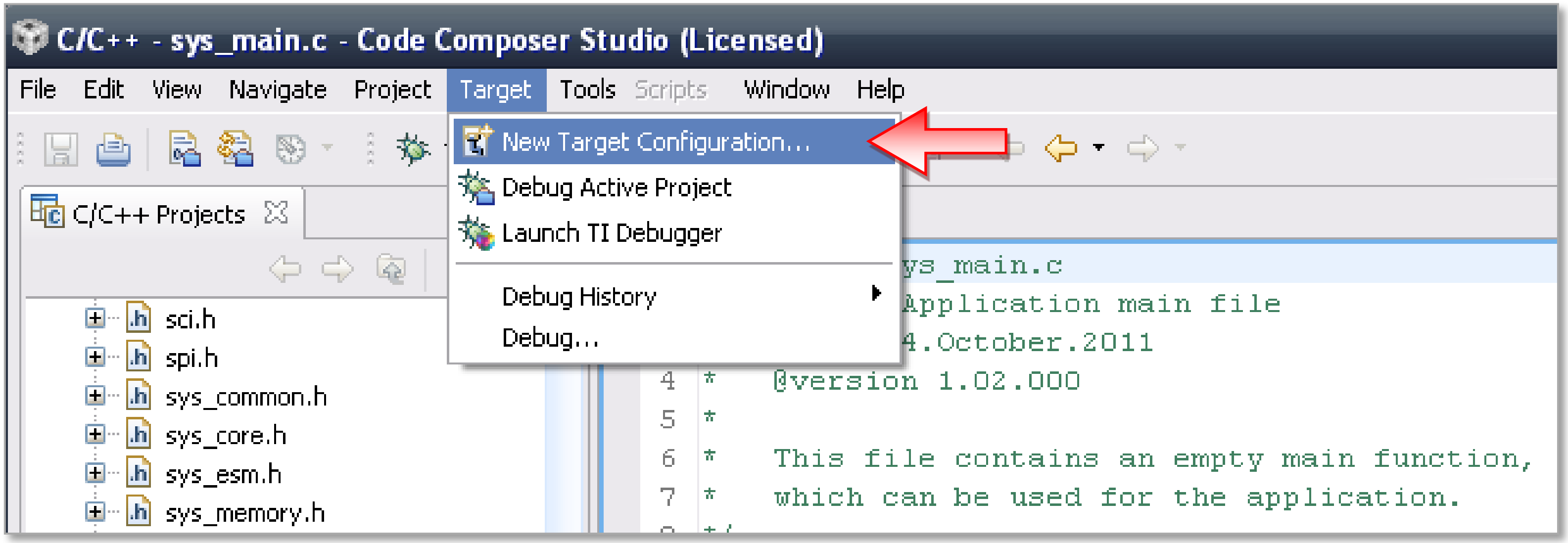


```
d)
n main function
on is empty by default.

called after startup.
this function to implement

) */

) */
ins to output */
etPORT, 0xFFFFFFFF);

    /* Set HET port pin 1 high */
    gioSetBit(hetPORT, 1, 1);
```

- Now that we have our .out file, we need to program the microcontrollers Flash memory.

Creating a Target Configuration

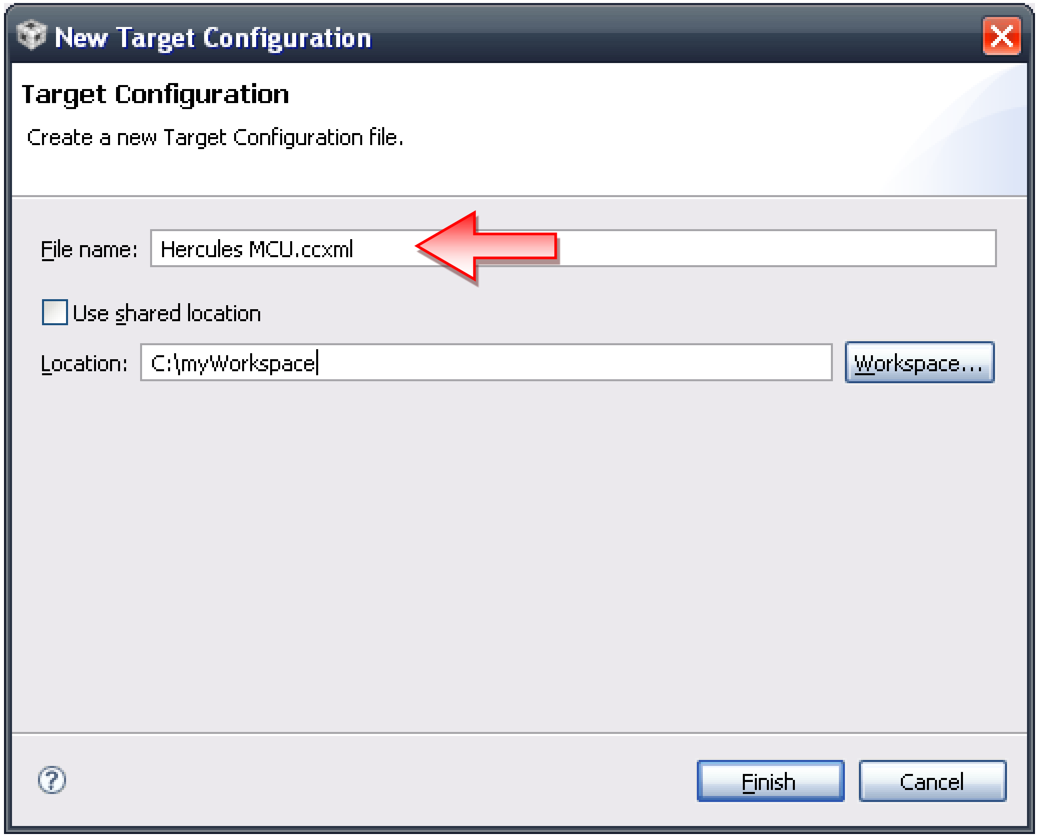
- Before we begin, we must make a new target configuration, this tells CCS4 what device this project is designed for.
  - Target → New Target Configuration


C/C++ - sys_main.c - Code Composer Studio (Licensed)
File Edit View Navigate Project Target Tools Scripts Window Help
New Target Configuration...
Debug Active Project
Launch TI Debugger
Debug History
Debug...
C/C++ Projects
sci.h
spi.h
sys_common.h
sys_core.h
sys_esm.h
sys_memory.h

```
ys_main.c
Application main file
4.October.2011
 *    @version 1.02.000
 *
 *    This file contains an empty main function,
 *    which can be used for the application.
```

Creating a Target Configuration

- A new window will appear, we will make our file name “Hercules MCU.ccxml”
- Click Finish



New Target Configuration

Target Configuration

Create a new Target Configuration file.

File name: Hercules MCU.ccxml

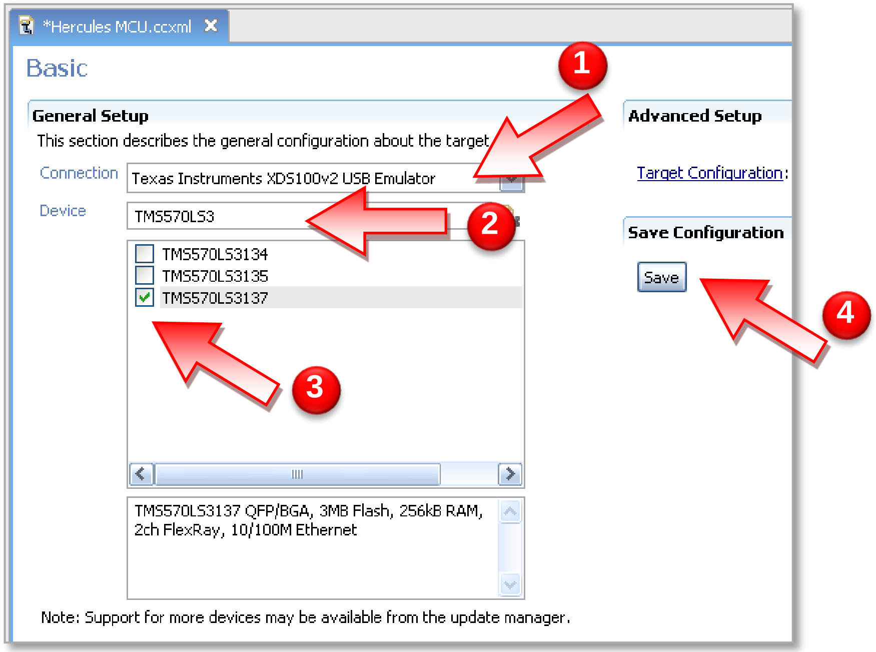
□ Use shared location

Location: C:\myWorkspace

Creating a Target Configuration

- A new tab will appear with a list of emulators and devices.
  - Connection: Texas Instruments XDS100v2 USB Emulator

- Device General Setup:
  - In the 'Device' box start typing TMS570LS3 to filter the device list
  - Select Device: TMS570LS3137



*Hercules MCU.ccxml

Basic

General Setup

This section describes the general configuration about the target.

Connection: Texas Instruments XDS100v2 USB Emulator

Device: TMS570LS3

☐ TMS570LS3134
☐ TMS570LS3135
☑ TMS570LS3137

TMS570LS3137 QFP/BGA, 3MB Flash, 256kB RAM, 2ch FlexRay, 10/100M Ethernet

Note: Support for more devices may be available from the update manager.

Advanced Setup

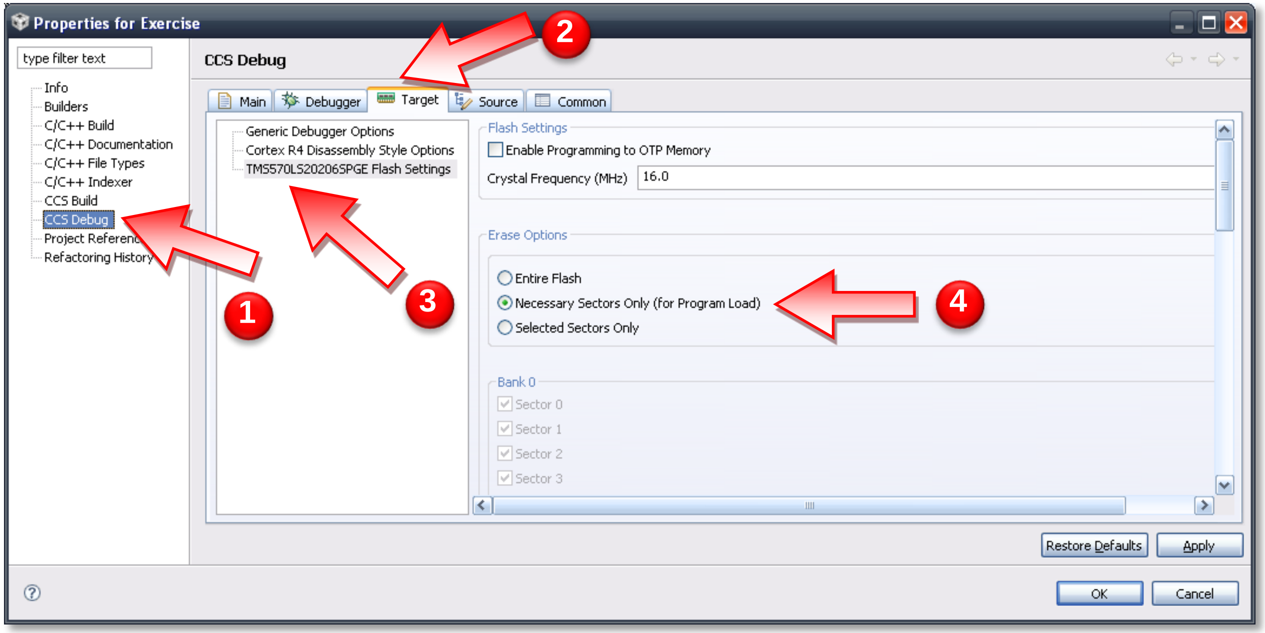
Target Configuration:

Save Configuration

- Click "Save" on the right

Flash Programming Configuration

- It is possible to make the flash programming process much faster by only the necessary erasing and programming the necessary regions of flash memory.
  - To do so go to Project → Properties
  - In the window that appears select ‘CCS Debug’
  - In the CCS Debug window select the TMS570LS3137 Flash Settings option in the ‘Target’ tab.
  - Then select the ‘Necessary Sectors Only’ option in the Erase Options area, then click the ‘Apply’ button.



Flash Settings

☐ Enable Programming to OTP Memory

Crystal Frequency (MHz) 16.0

Erase Options

○ Entire Flash

● Necessary Sectors Only (for Program Load)

○ Selected Sectors Only

Bank 0

☑ Sector 0

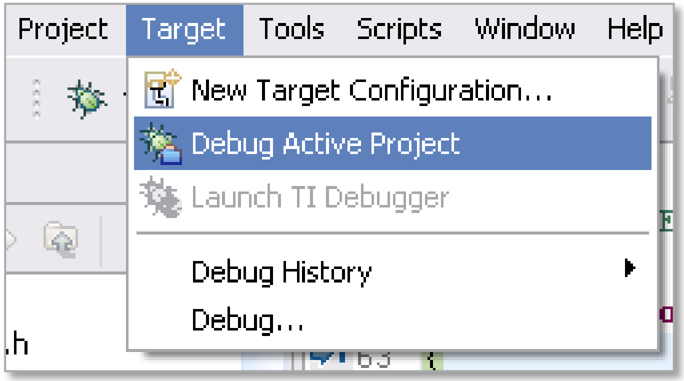
☑ Sector 1

☑ Sector 2

☑ Sector 3

Programming the Flash

- We are now ready to program the flash.
  - Go to Target → Debug Active Project
  - A new window should appear as it programs the flash memory.
    - This may take a few moments.



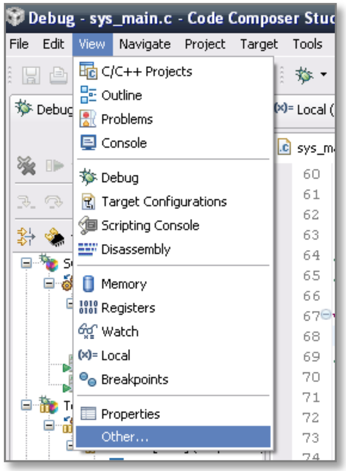
Testing your code

- Upon Completion open your preferred terminal program.
  - Note: A terminal program is included in CCSv4. To enable it go to 'View' -> Other and select 'Terminal' from the 'Show View' menu. *(See the next page for more information).*

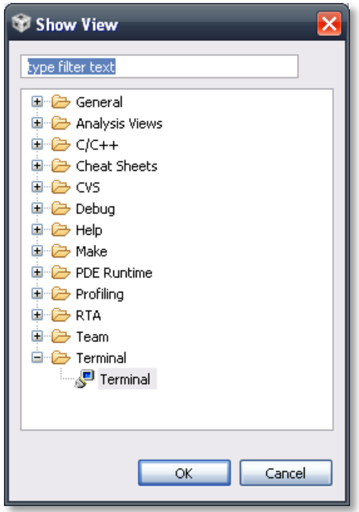
- Setup the terminal program with the following properties:
  - Baud rate: 9600
  - Data bits: 8
  - No parity, 2 Stop bits

Enabling the CCSv4 Terminal

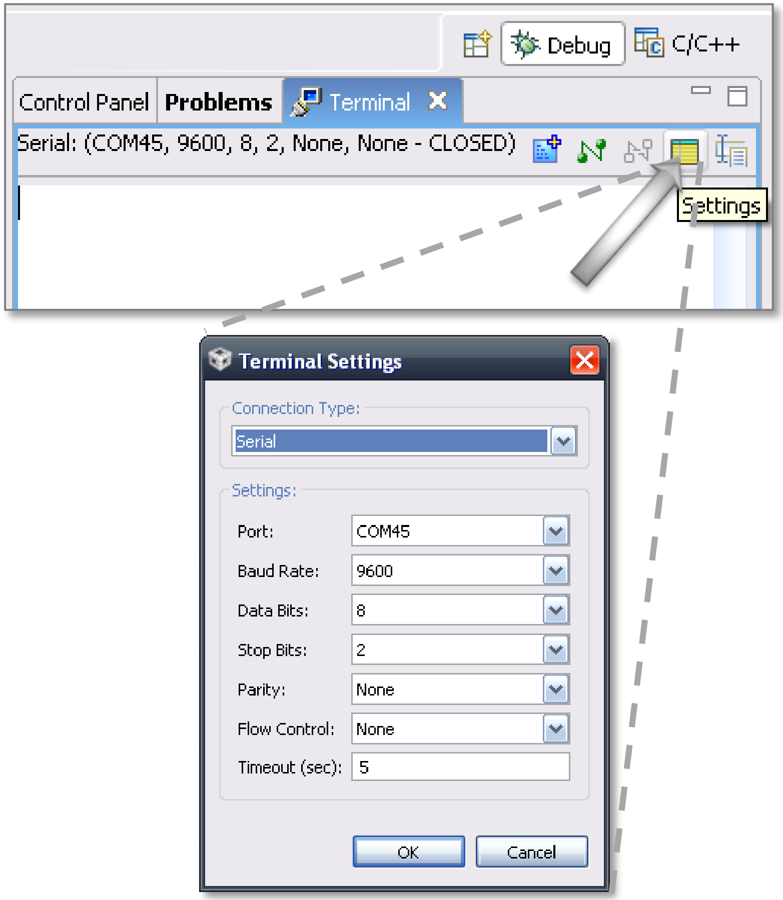
1) Select View → Other



2) Then select 'Terminal' from the 'Show View' menu.



3) Enter the proper communication settings for the 'Terminal Tab'



Serial: (COM45, 9600, 8, 2, None, None - CLOSED)

Connection Type: Serial

Settings:

Port: COM45

Baud Rate: 9600

Data Bits: 8

Stop Bits: 2

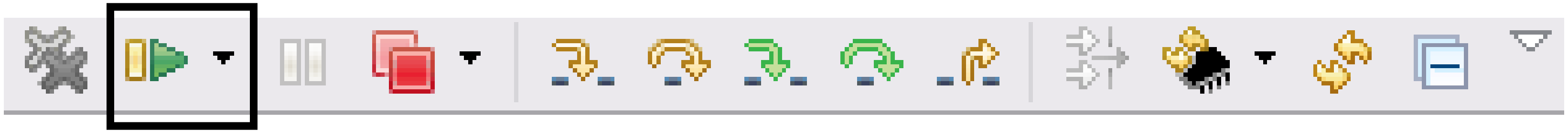
Parity: None

Flow Control: None

Timeout (sec): 5

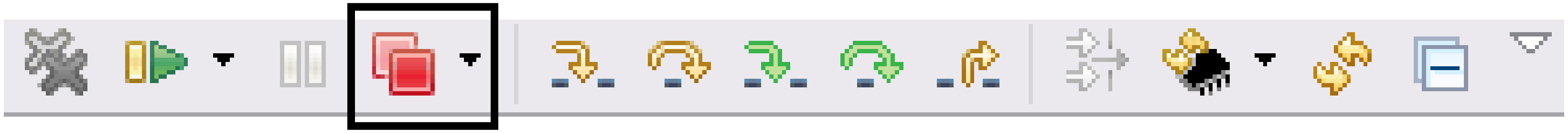
Testing our Program

- Click the green arrow on the debug tab to run our program



  - Alternatively the program can be run without the debugger connected by

- Clicking the red square on the debug tab to terminate the debugger's connection



- Hit the reset button on the board and view the ADC data in the terminal. Use the LED flashlight included with the TMS570 development kit to change the light level supplied to the ambient light sensor on the board.

- Congratulations! You have completed the example.

For more Hercules™ TMS570 Information

TMS570 Web Page: www.ti.com/TMS570

- Data Sheets
- Technical Reference Manual
- Application Notes
- Software & Tools Downloads and Updates
- Order Evaluation and Development Kits

TMS570 Engineer 2 Engineer Forums:
www.ti.com/hercules-support

- News and Announcements
- Useful Links
- Ask Technical Questions
- Search for Technical Content

TI E2E™ Community

TMS570 WIKI:
www.ti.com/hercules-tms570-wiki

- How to guides
- Intro Videos
- General Information

Thank You!

For completing this TMS570 example