



TMS570LS Microcontrollers: MibADC Example



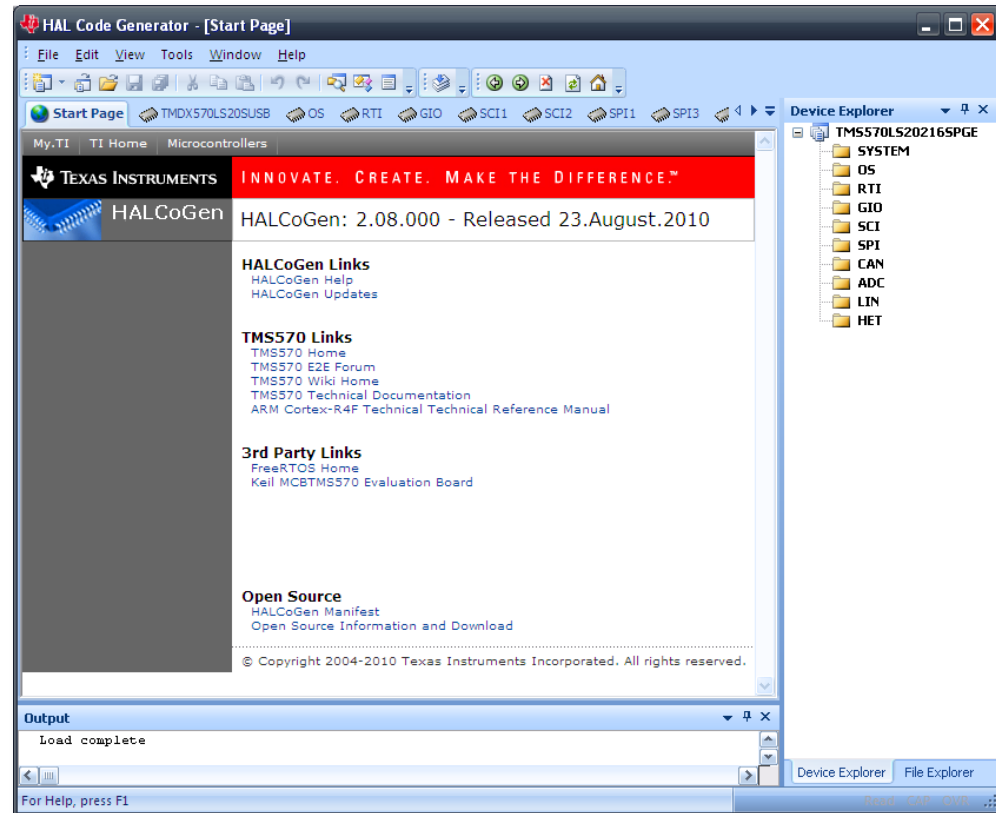
Overview



- In this example we will:
 - Create a TMS570 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 MDK) 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)
 - TMS570LS2x USB Development Stick or Microcontroller Development Kit
- Required Software:
 - TMS570 HALCoGen
 - Code Composer Studio v4.x

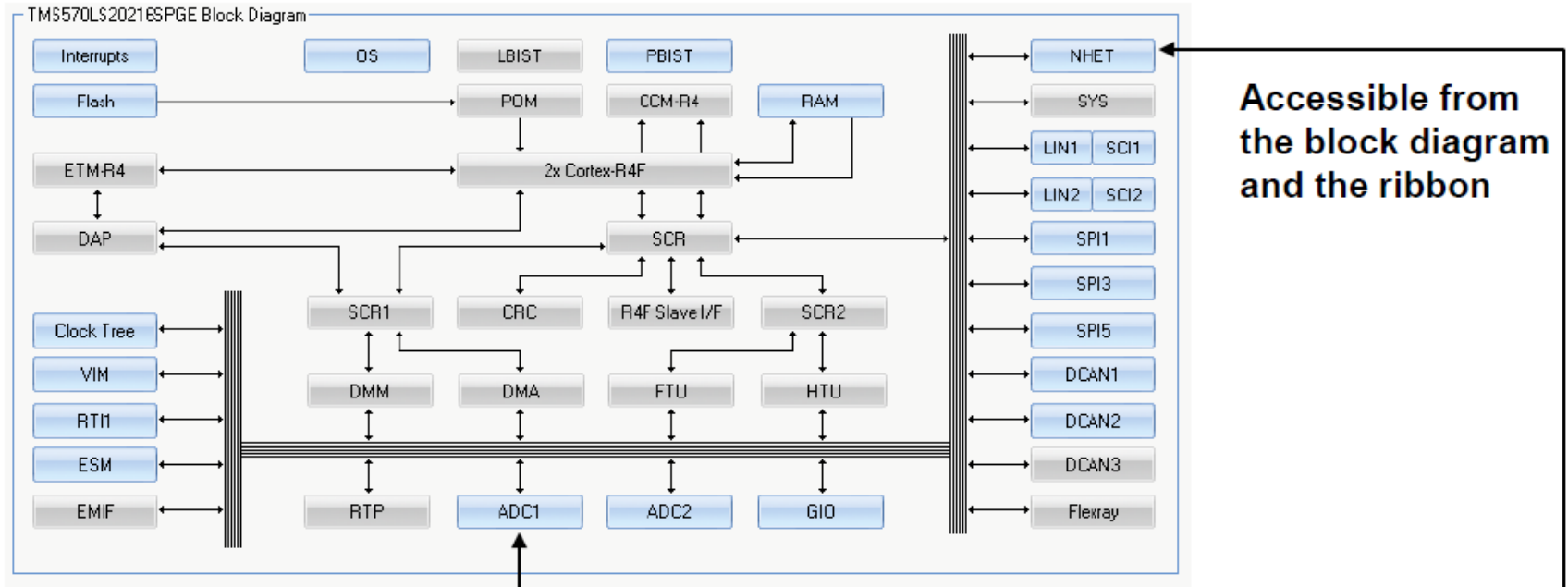
Setting up a New HALCoGen Project

- Launch HALCoGen
 - Start → Programs → Texas Instruments → HALCoGen
- File > New > Project
- Family:
 - TMDX570
- Device:
 - TMDX570LS20USB (for USB Stick)
 - OR**
 - TMDX570LS20MDK (for MDK)
- Name: TMS570 MibADC
- Location: “C:\myWorkspace”

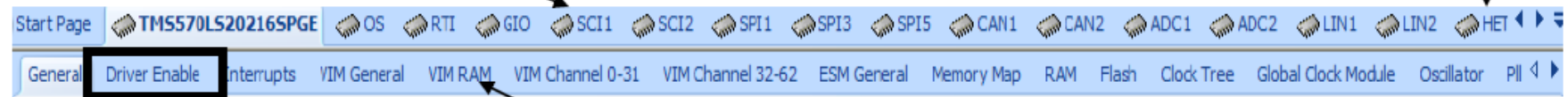


For more help with HALCoGen, see this getting started video: [LINK](#)

The HALCoGen Interface



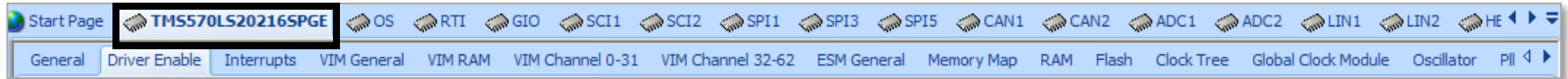
Top Level: Peripherals on the TMS570



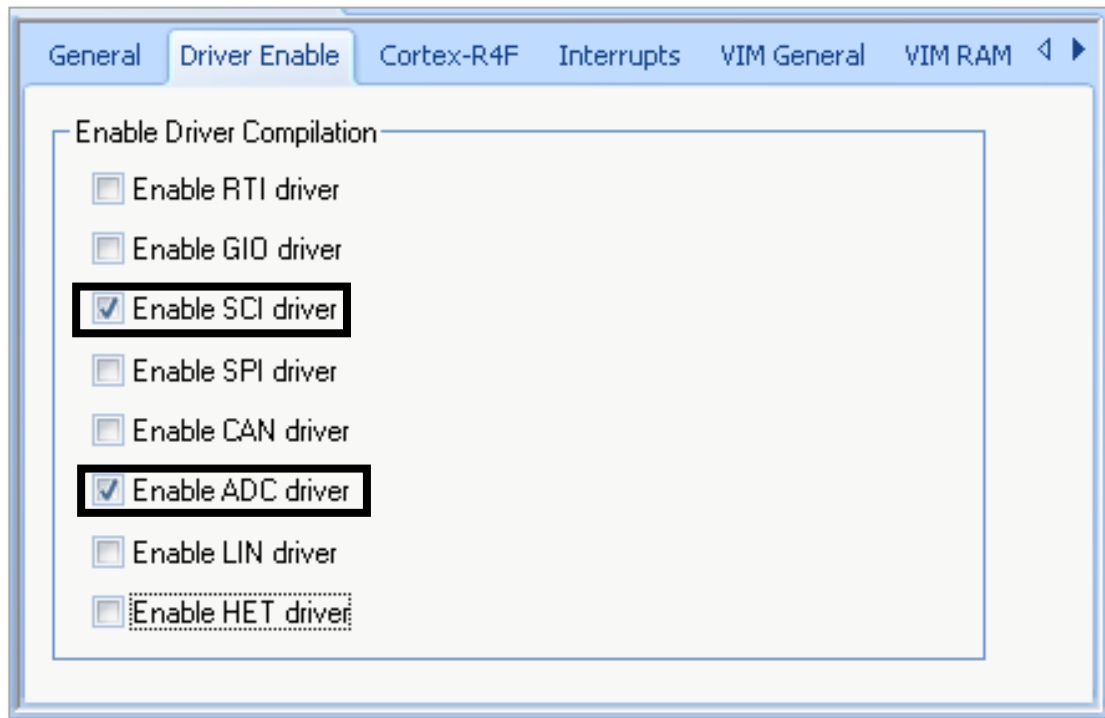
Click "Driver Enable"

Bottom Level: Settings for each peripheral

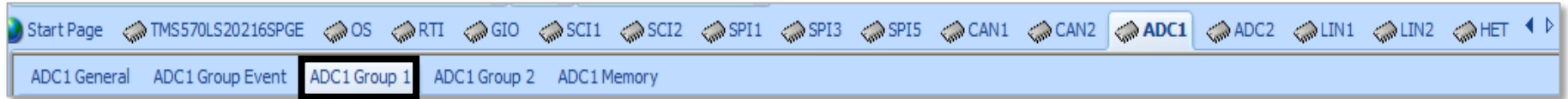
Configuring the Peripheral Drivers



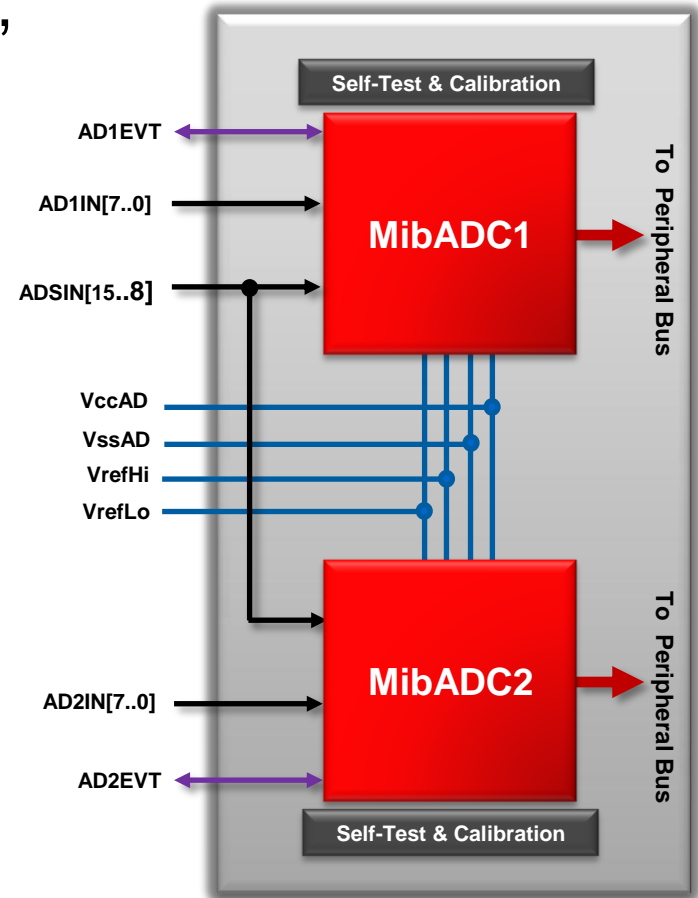
- Select the peripherals that are required for this project.
 - In this example we need to enable the SCI and ADC drivers



Setting up an ADC Conversion Group



- The TMS570 has two ADC Modules, each with 3 conversion groups.
 - The 'ADC Group Event' can only be hardware triggered
 - The other groups can use either a software or hardware trigger.
- We will be using a software trigger.
 - Select the 'ADC1 Group 1' tab.



ADC Group 1 Settings

- We are only making one conversion so set the FiFo Size to 1
- At the bottom, in the 'ADC1 Group 1 Channel Selection' area, check the "Enable Pin 9" box
 - The ambient light sensor is on ADC 1 Pin 9

ADC1 General ADC1 Group Event **ADC1 Group 1** ADC1 Group 2 ADC1 Memory

ADC1 Group 1 Configuration

FiFo Size: 1 Enable Channel Id in Conversion Results
Data Resolution (Bit): 12_B1T Enable Continuous Conversion

ADC1 Group 1 Trigger

HW Trigger Source: EVENT Rising Edge: Falling Edge: Hardware: Software: Trigger

ADC1 Group 1 Sampling

tScan Start tExtended tDischarge tSample tConversion End

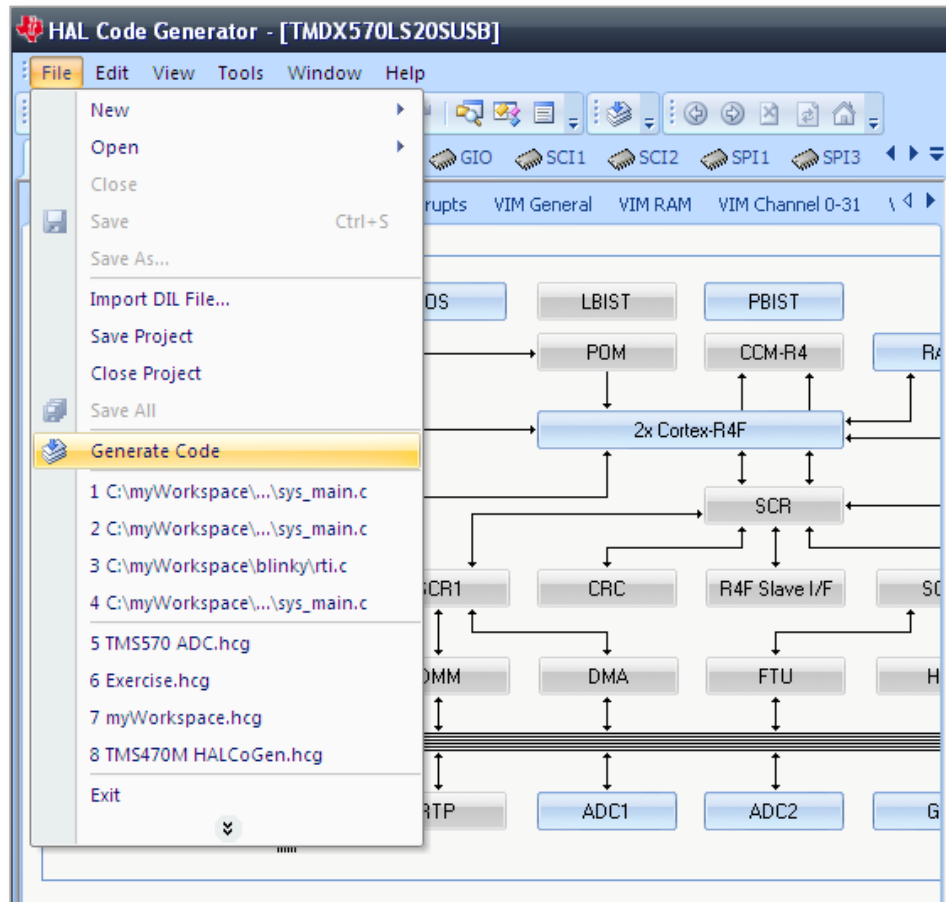
Enable Sampling Capacitor Discharge Discharge Time: 0.00
Cycle Time: 100.00 Discharge Prescaler: 0 tDischarge (ns): 0.00
Sample Time: 200.00
Cycle Time: 100.00 Sample Prescaler: 0 tSample (ns): 200.00
tScanTotal (ns): 0.000 tExtended (ns): 200.00 tConversion (us): 1.300
tTotal (us): 1.500000

ADC1 Group 1 Channel Selection

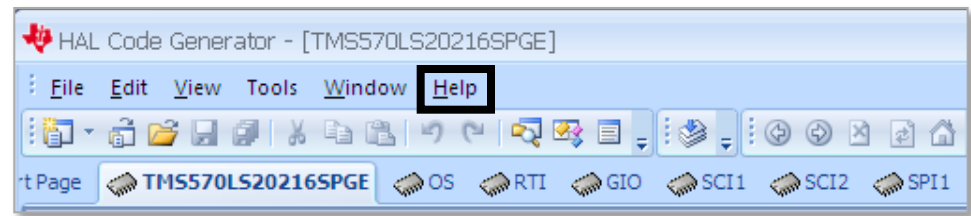
Enable Pin 0 Enable Pin 1 Enable Pin 2 Enable Pin 3
 Enable Pin 4 Enable Pin 5 Enable Pin 6 Enable Pin 7
 Enable Pin 8 Enable Pin 9 Enable Pin 10 Enable Pin 11
 Enable Pin 12 Enable Pin 13 Enable Pin 14 Enable Pin 15

Generate Code

- The HALCoGen Settings are now complete,
- The next step is to generate the code using the File → Generate Code menu.



HALCoGen Help



- Information about the files and functions that HALCoGen creates can be found in the HALCoGen 'Help' menu
- Help can be launched from the main title bar under Help → Help Topics

C:/HALCoGen/adc.c File Reference

ADC Driver Source File. More...

```
#include "adc.h"
```

Include dependency graph for adc.c:

```
graph TD
    A[C:/HALCoGen/adc.c] --> B[adc.h]
```

Go to the source code of this file.

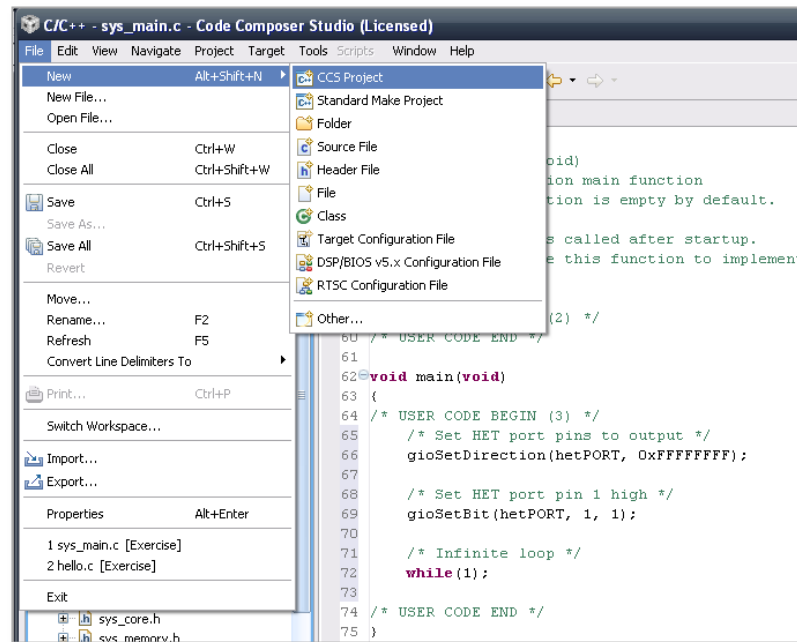
Functions

| | | |
|------|---|-------------------------------------|
| void | adcInit (void) | Initializes ADC Driver. |
| void | adcStartConversion (adcBASE_t *adc, unsigned group) | Starts an ADC conversion. |
| void | adcStopConversion (adcBASE_t *adc, unsigned group) | Stops an ADC conversion. |
| void | adcResetFifo (adcBASE_t *adc, unsigned group) | Resets FIFO read and write pointer. |
| int | adcGetData (adcBASE_t *adc, unsigned group, adcData_t *data) | Gets converted a ADC values. |

Setting up Code Composer Studio 4



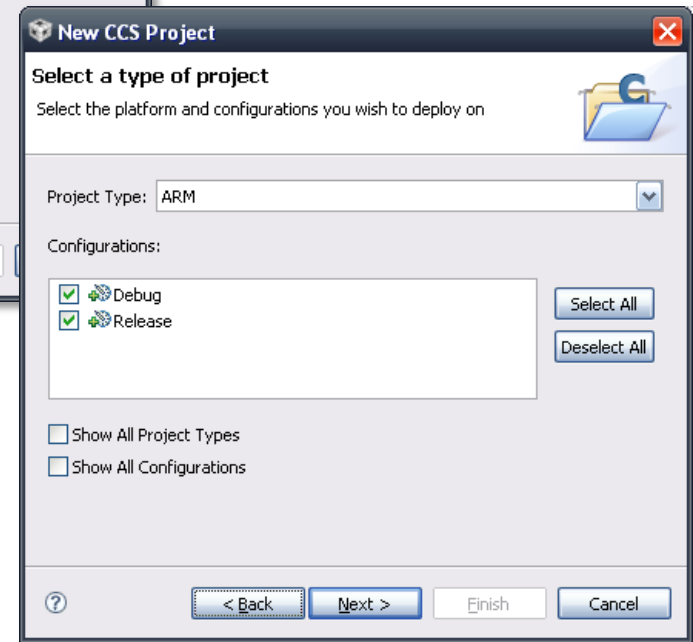
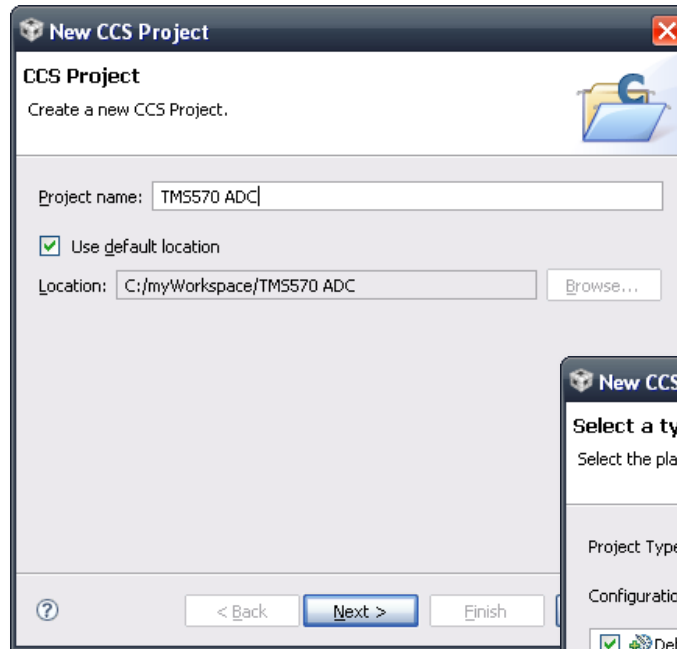
- Launch Code Composer Studio v4.x (CCS)
 - Start → Programs → Texas Instruments → Code Composer Studio v4 → Code Composer Studio v4
- When it launches, CCS will ask you to select a workspace, we will chose “C:\myWorkspace”
- Once it loads, go to File → New → CCS Project



For more help with Code Composer Studio, see this getting started video: [LINK](#)

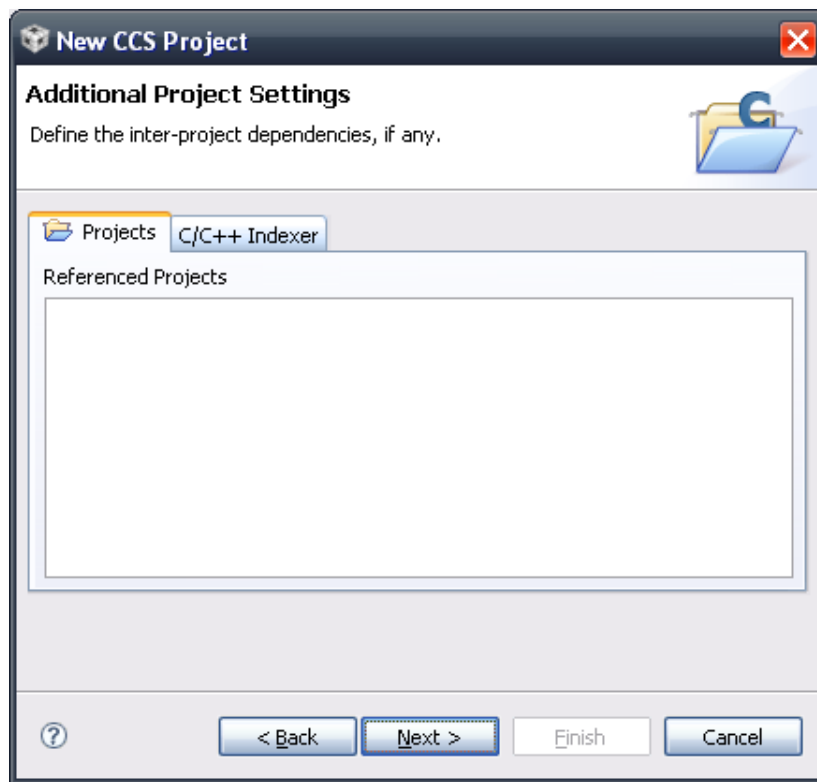
Setting up our Project

- Our project name needs to match the name of our HALCoGen Project, TMS570 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”



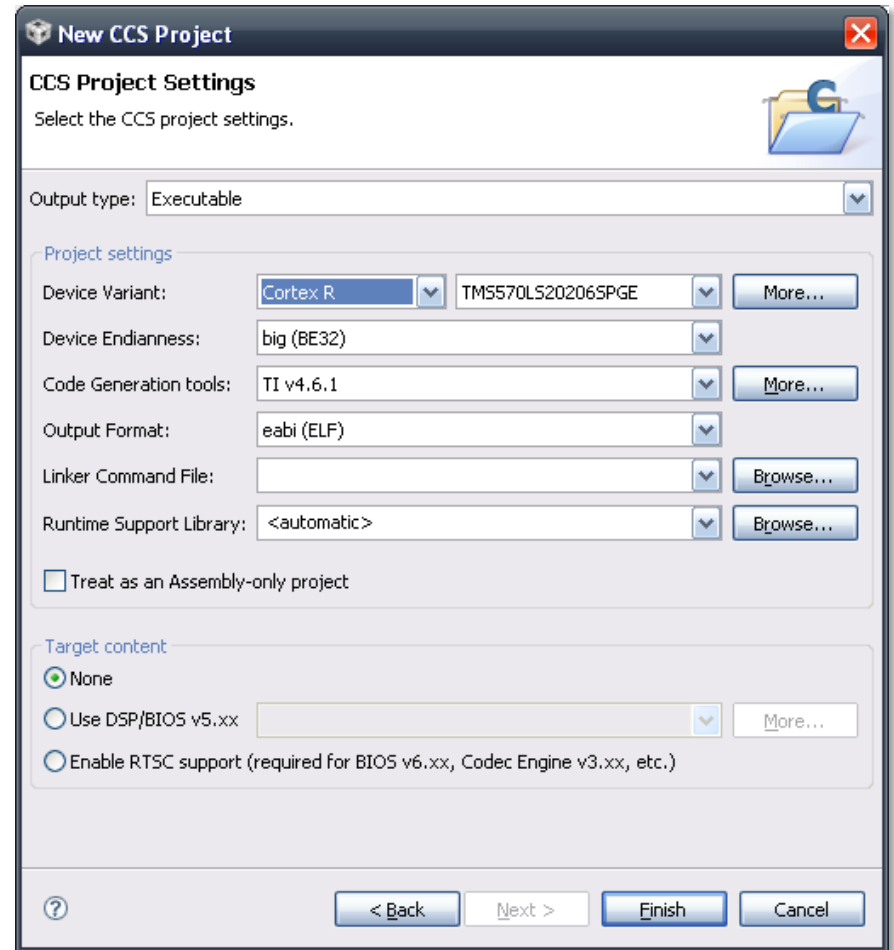
Setting up our Project (cont.)

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



Setting up the Project (cont.)

- Lastly, set the Device Variant to “Cortex R” and TMS570LS20216SPGE
- Click “Finish”



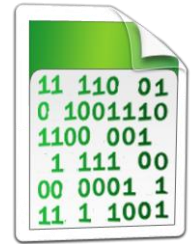
Getting Started



- On the left hand side in the “C/C++ Projects” explorer, open “sys_main.c”
- When ever you generate code in HALCoGen, the program overwrites user code, except specific sections marked by “USER CODE BEGIN (x)” and “USER CODE END”
 - For code placement we will be referring to the number within the User Code block

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

Writing the Code



- Inside User Code 1, copy the code below to include the necessary header files for the project.

```
/* USER CODE BEGIN (1) */  
#include "sci.h"  
#include "adc.h"  
unsigned char command[8];  
/* USER CODE END */
```

Writing the Code cont...

- Then in User Code 3, copy the code below.

```
/* USER CODE BEGIN (3) */
adcData_t adc_data; //ADC Data Structure
adcData_t *adc_data_ptr = &adc_data; //ADC Data Pointer
unsigned int i, value; //Declare variables

scilnit(); //Initializes the SCI (UART) module
adclnit(); //Initializes the ADC module

while(1) // Loop to acquire and send ADC sample data via the SCI (UART)
{
    adcStartConversion(adcREG1, 1U); //Start ADC conversion
    while(!adclConversionComplete(adcREG1, 1U)); //Wait for ADC conversion
    adcGetData(adcREG1, 1U, adc_data_ptr); //Store conversion into ADC pointer
    value = (unsigned int)adc_data_ptr->value;
    for(i=0; i<8; i++) //Convert ADC value into a char array
    {
        command[i] = (value>>28)&15;
        if (command[i] > 9)
            command[i] +=7;
        command[i] +=48;
        value=value<<4;
    }
    sciSend(sciREG1, 2, (unsigned char *)"0x"); //Sends '0x' hex designation chars
    sciSend(sciREG1, 8, command); //Sends the ambient light sensor data on the SCI/UART
    sciSend(sciREG1, 2, (unsigned char *)"\r\n"); //Sends new line character
}

/* USER CODE END */
```



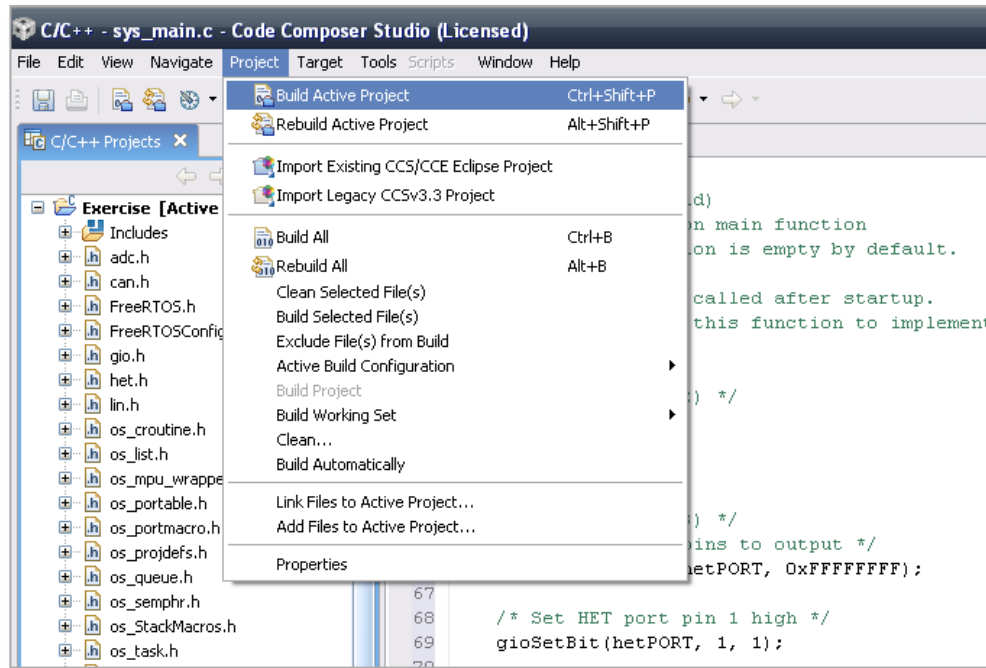
Notifications

- Lastly we must insert a function that would be called if interrupts were enabled. Without these, the code will fail to build

```
/* USER CODE BEGIN (4) */  
/* Notification functions not used, but required by compiler */  
void adcNotification(adcBASE_t *adc, unsigned group)  
{  
    return;  
}  
  
void sciNotification(sciBASE_t *sci, unsigned flags)  
{  
    return;  
}  
  
/* USER CODE END */
```

Compiling the Project

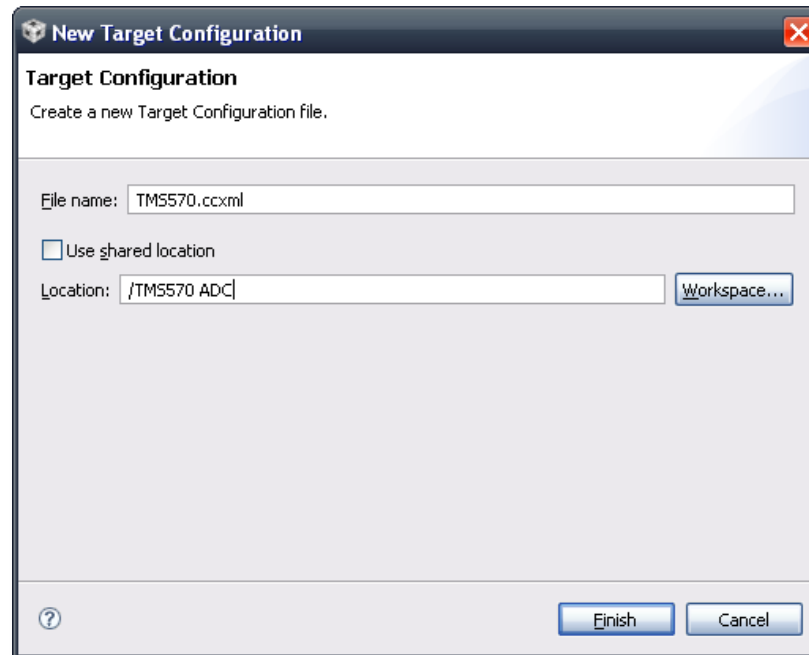
- The code is now complete and we are ready to build our project.
 - Go to Project → Build Active Project



- 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
- A new window will appear, we will make our file name “TMS570.ccxml”
- Click Finish



Creating a Target Configuration...

- A new tab will appear with a list of emulators and devices.
 - Connection: Texas Instruments XDS100v2 USB Emulator
 - In the text box labeled “Type Filter Text”, type “TMS570”.
 - This will narrow the search down to just TMS570 devices, select TMS570LS20216SPGE
 - Click “Save” on the right

Connection: Texas Instruments XDS100v2 USB Emulator

Device: TMS570

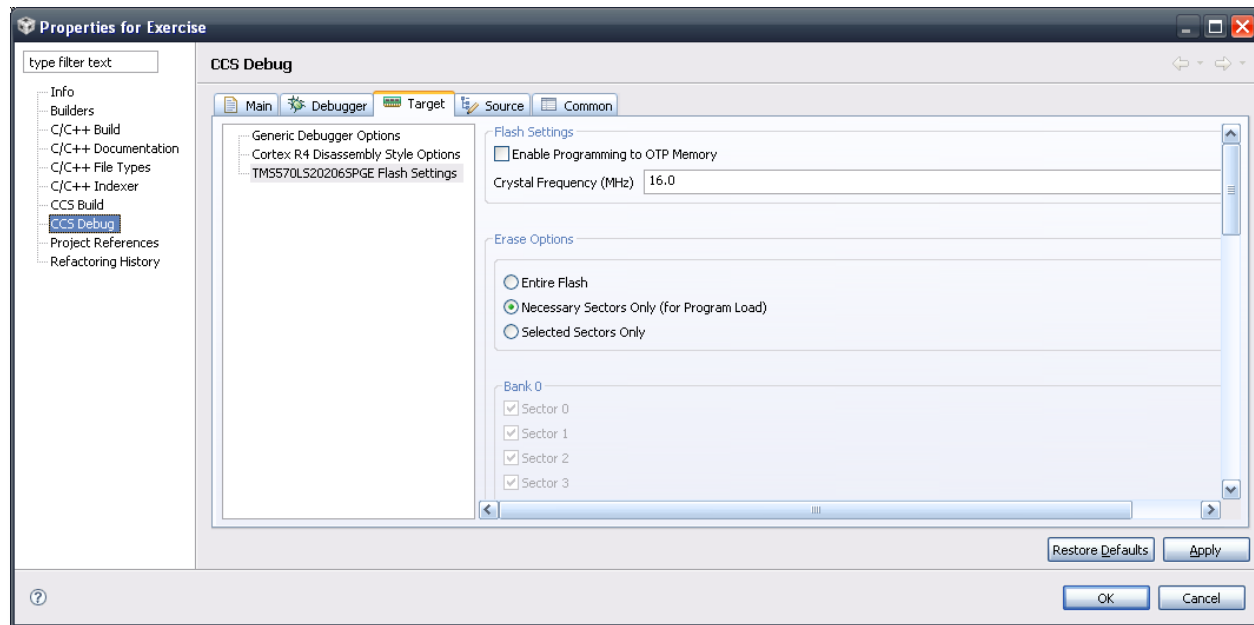
- TMS570LS20206SPGE
- TMS570LS20206SZWT
- TMS570LS20216SPGE
- TMS570LS20216SZWT

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 TMS570LS20216SPGE Flash Settings option in the 'Target' tab.
 - Then select the 'Necessary Sectors Only' option in the Erase Options area, then click the 'Apply' button.

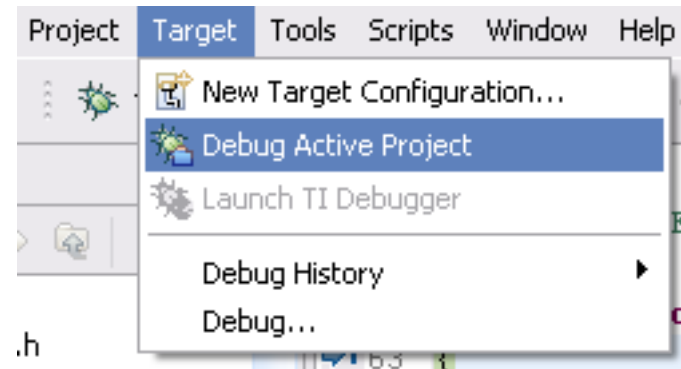
NOTE:

This option is only available in CCSv4.2 and newer



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.

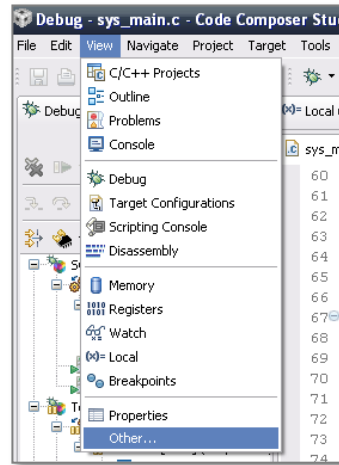


Testing your code

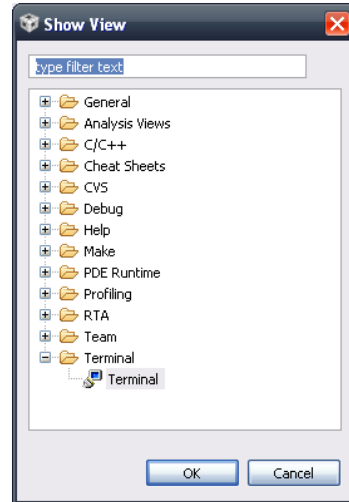
- Upon Completion open your preferred terminal program.
 - Note: A terminal program is included in CCSv4. To enable it go to 'View' -> Other and select 'Terminal' from the 'Show View' menu. *(See the next page for more information).*
- Setup the terminal program with the following properties:
 - Baud rate: 9600
 - Data bits: 8
 - No parity, 2 Stop bits

Enabling the CCSv4 Terminal

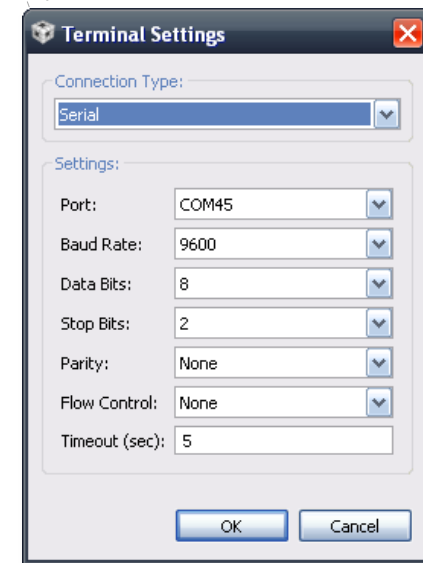
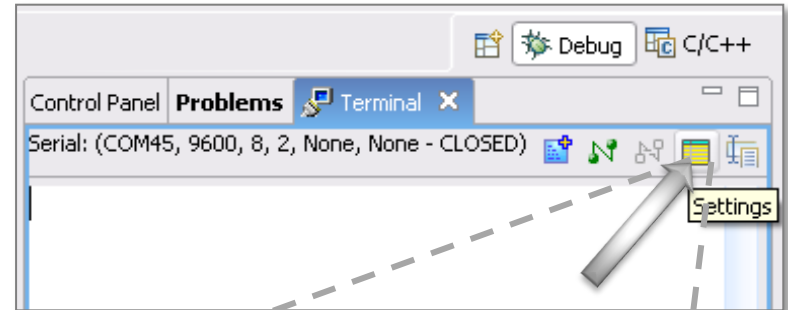
1) Select View → Other



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

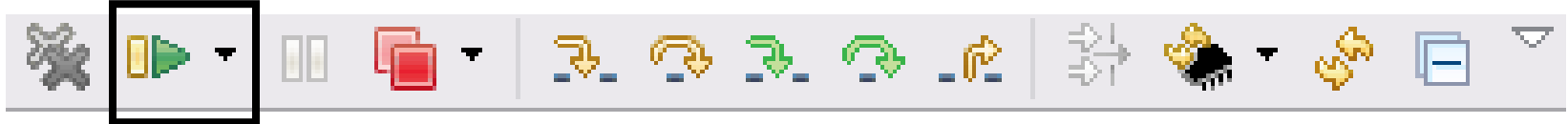


3) Enter the proper communication settings for the 'Terminal Tab'



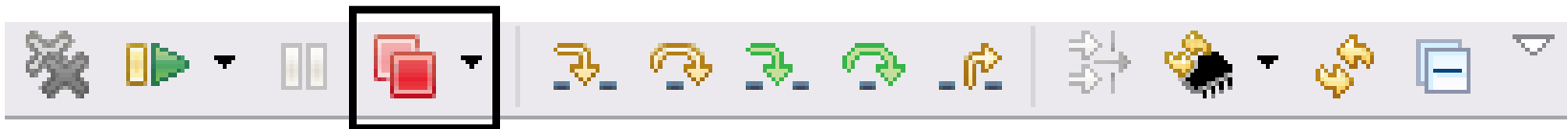
Testing our Program

- 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 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 Forum:
<http://e2e.ti.com/support/microcontrollers/tms570/default.aspx>
 - News and Announcements
 - Useful Links
 - Ask Technical Questions
 - Search for Technical Content
- TMS570 WIKI:
<http://processors.wiki.ti.com/index.php/Category:TMS570>
 - How to guides
 - Intro Videos
 - General Information



Thank You!

For completing this TMS570 example

