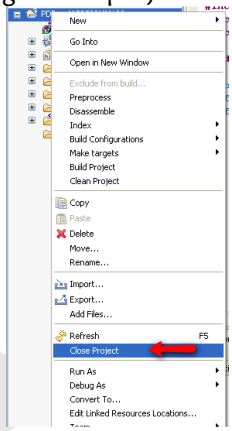
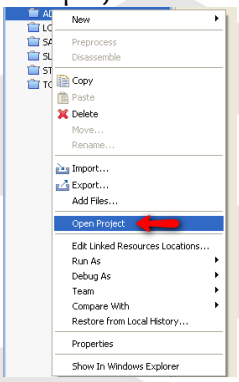
ADC Lab

This lab demonstrates how to use the 16-bit ADC module available at MM devices.

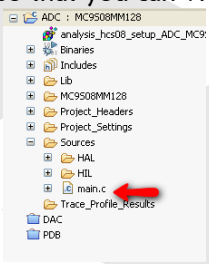
1. Close all other open projects by right clicking on the project name and select “Project Close”



2. Open the “ADC” project by right clicking on the project name and select “Open Project”



Ensure that the Sources folder is expanded so that you can view main.c content



3. Open main.c file by double clicking on it
4. In main .c, find the comment **/** Step1: Init ADC module*/**
5. Open the ADC initialization function, by placing the cursor over the text and pressing **F3** :

   vfnADC_Init ();

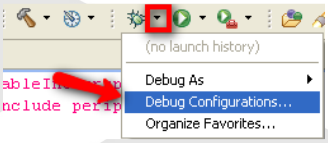
6. Then find the comment **/** Step2: Settings for ADC Channel 0 */**
7. Configure channel 0 with the following settings
   Channel 4, No hardware average, No hardware compare

   `sADCChannels[0].u8Channel` → Select the ADC channel to be converted.
   `sADCChannels[0].u8Control1` → Configures hardware compare
   `sADCChannels[0].u16CompareValue1` → Hardware compare voltage reference
   `sADCChannels[0].u8Control2` → Configures hardware average

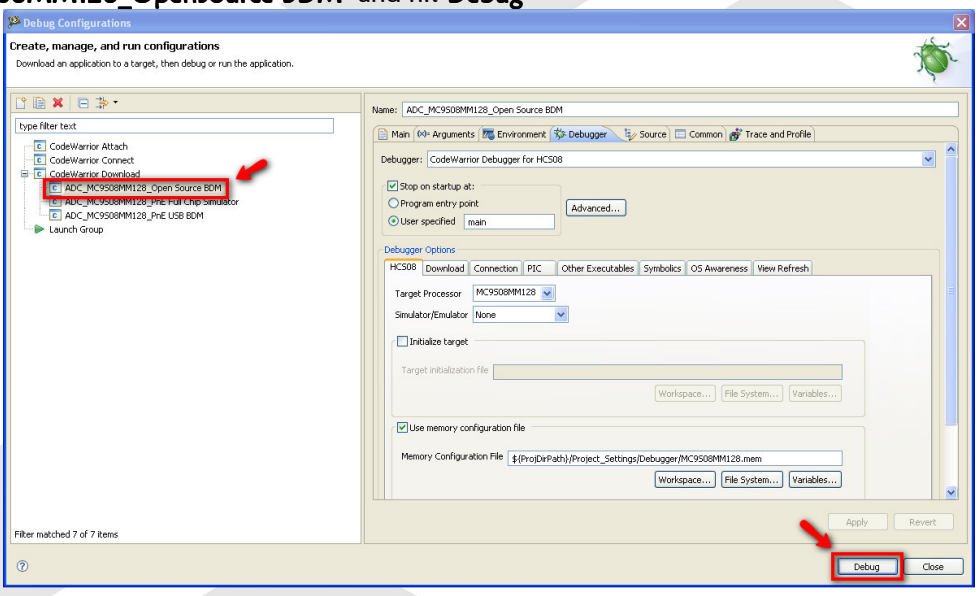
Note: "ADC.h" contains the macros to configure these kinds of settings (lines 97-132)

8. Go back to main.c and find the comment **/** Step3: Start Continuous Conversions */**
9. Convert channel 0 continually by typing :
   ADC_CONVERT_CONTINUOUS_CHANNEL(0)

10. Save the changes by clicking on the Save icon in the toolbar: 
11. Then build the project by clicking on the Hammer icon 
12. Then flash the code by clicking on the *arrow* next to the green bug in the menu bar, select **Debug Configurations**



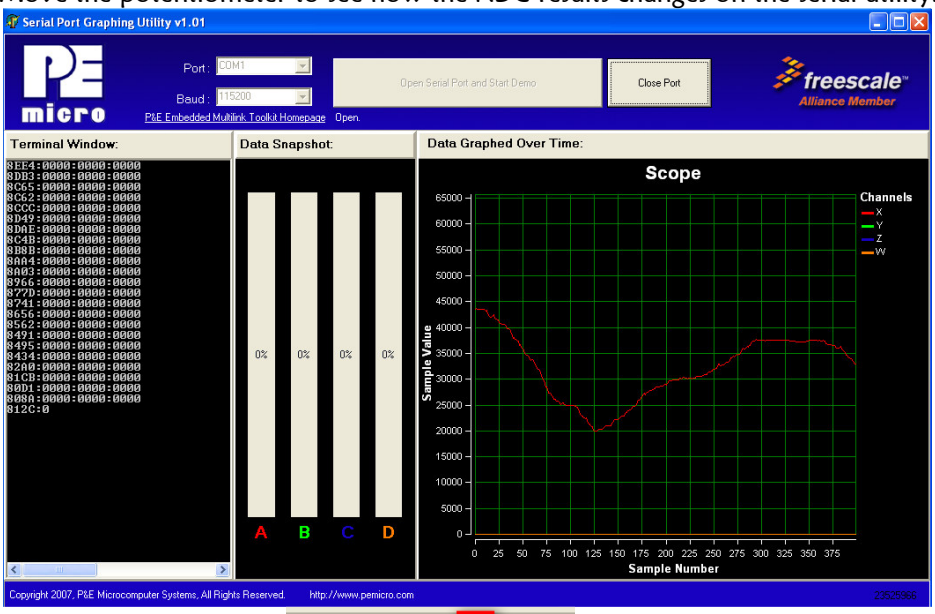
13. The **Debug Configurations** dialog box will pop up. Then in the File Options section, click on **LCD_MC9S08MM128_OpenSource BDM** and hit **Debug**



14. Open the p&e serial grapher utility, connect the serial cable; open the port at COM1 with 115200 of baudrate.

15. Resume by clicking the play button (F8).

16. Move the potentiometer to see how the ADC results changes on the serial utility.



17. Suspend the application 
18. Stop the debugger session 
19. Change to the C/C++ perspective 

20. Find the comment **/** Step2: Settings for ADC Channel Structure 0 */**
21. Configure Channel 1 settings with:

Channel 4, Hardware compare with values higher than one volt, and hardware average with 32 samples

`sADCChannels[0].u8Channel` → Select the ADC channel to be converted.
`sADCChannels[0].u8Control1` → Configures hardware compare
`sADCChannels[0].u16CompareValue1` → Hardware compare voltage reference
`sADCChannels[0].u8Control2` → Configures hardware average

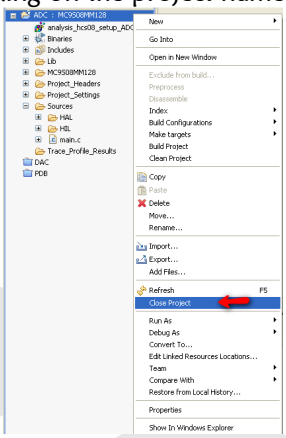
Note: "ADC.h" contains the macros to configure these kinds of settings (lines 97-132).

22. Repeat 10 – 18 steps

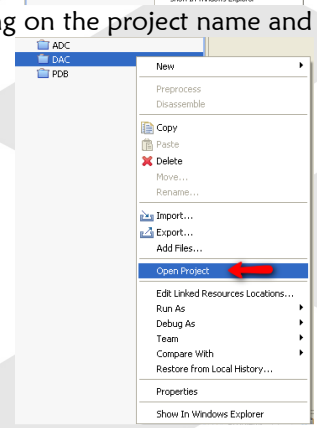
DAC Lab

This lab demonstrates how to build a sine wave, software triggering the DAC module, and using the VREF module like a reference voltage. Once the wave has been created the OPAMP amplifies the DAC output.

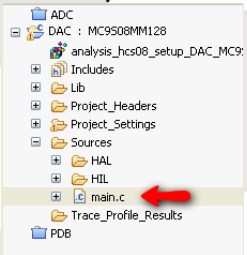
1. Close all other open projects by right clicking on the project name and select “Project Close”



2. Open the **“DAC”** project by right clicking on the project name and select “Open Project”



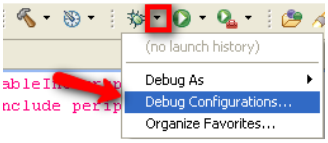
Ensure that the Sources folder is expanded so that you can view main.c content



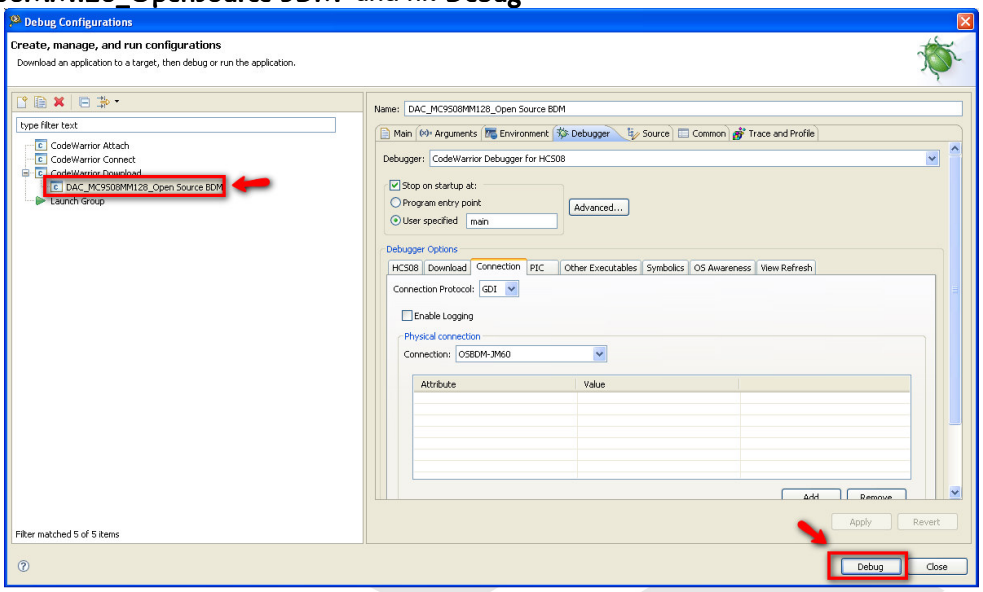
3. Open main.c file by double clicking on it

4. Save the changes by clicking on the Save icon in the toolbar: 

5. Then build the project by clicking on the Hammer icon 
6. Then flash the code by clicking on the *arrow* next to the green bug in the menu bar, select **Debug Configurations**



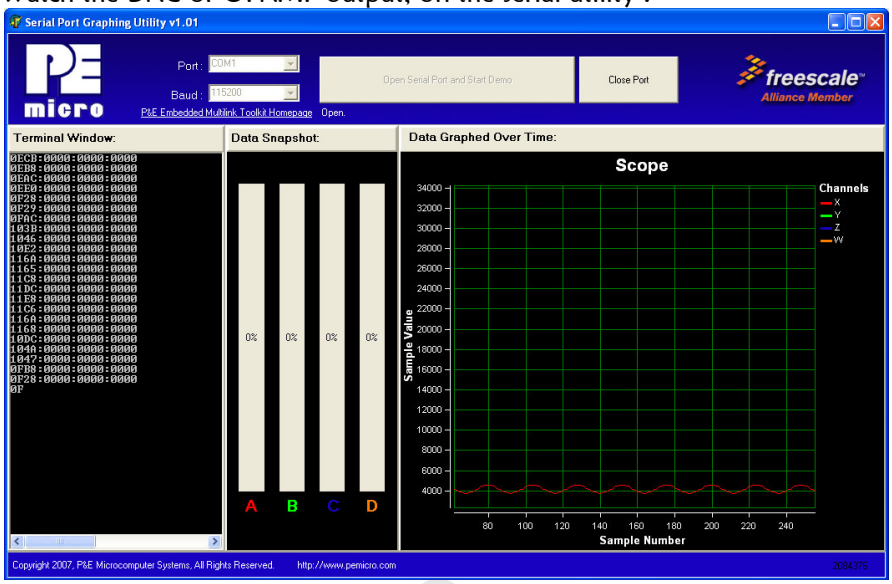
7. The **Debug Configurations** dialog box will pop up. Then in the File Options section, click on **LCD_MC9S08MM128_OpenSource BDM** and hit **Debug**



8. At the p&e serial grapher utility, open the port at COM1 with 115200 of baudrate.

9. Resume by clicking the play button (F8).

10. Watch the DAC or OPAMP output, on the serial utility .



11. Suspend the application 
12. Stop the debugger session 
13. Change to the C/C++ perspective 

14. In main.c, find the comment **/** Step3: Amplify DAC output with the OPAMP */**
15. Open the General Purpose Operational Amplifier initialization function, by placing the cursor over the text and pressing **F3** :

    vfnOPAMPInit ();

16. Then find the comment **/** Step4: OPAMP initialization values */**
    Uncomment all the code below this comment
17. Configure OPAPMP0 with the following settings

    Non-Inverting mode , +13 Gain, VSS like negative input, and DAC like positive input

    `sOPAMP[0].gu8Mode` = → OPAMP mode
    `sOPAMP[0].gu8Gain` = → OPAMP gain
    `sOPAMP[0].gu8NegativeInput` = → Negative input
    `sOPAMP[0].gu8PositiveInput` = → Positive input

NOTE: "OPAMP.h" contains the macros to configure such settings (lines 59-100)

18. Find the comment **/**Step5: Measure sine wave*/**
19. Configure and convert channel 2 (OPAMP OUT0)continually by typing:

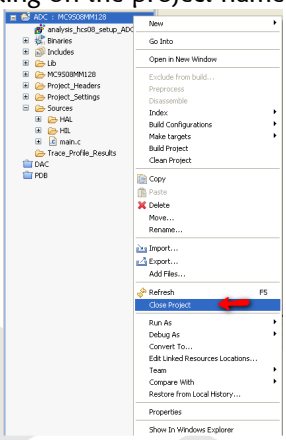
    vfnADConfigChannel(2);
    ADC_CONVERT_CONTINUOUS_CHANNEL(2);

20. Repeat 4 – 13 steps

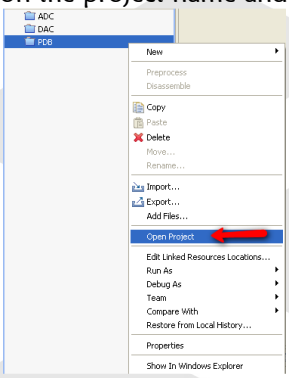
PDB Lab

This lab demonstrates how to hardware trigger the DAC counter, with the Programmable Delay Block (PDB). An also shows the Back-to-Back operation mode (PDB-ADC)

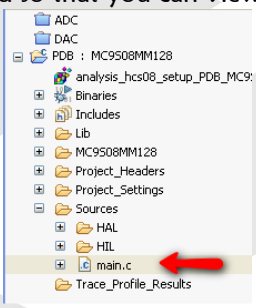
1. Close all other open projects by right clicking on the project name and select "Project Close"



2. Open the **"PDB"** project by right clicking on the project name and select "Open Project"



Ensure that the Sources folder is expanded so that you can view main.c content

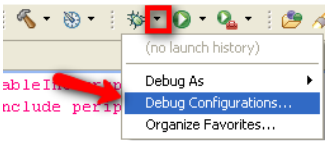


3. Open main.c file by double clicking on it
4. In main.c, find the comment **/**Step2: Sine wave to exit from DAC output*/**
5. Go to the *gu8DACConfig1* declaration by single clicking over the variable name and pressing **F3** :
   vfnDACInit(gu8DACConfig1, gu8DACConfig2);

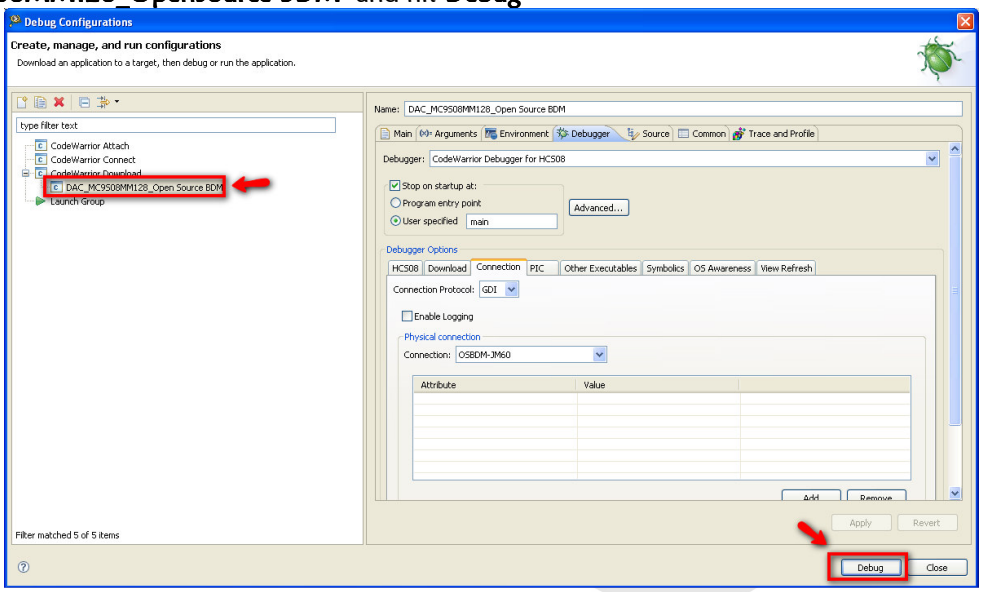
6. Change the DAC configuration to be hardware triggered by typing **DAC_TSEL_HW** instead of **DAC_TSEL_SW**

7. Save the changes by clicking on the Save icon in the toolbar: 

8. Then build the project by clicking on the Hammer icon 
9. Then flash the code by clicking on the *arrow* next to the green bug in the menu bar, select **Debug Configurations**



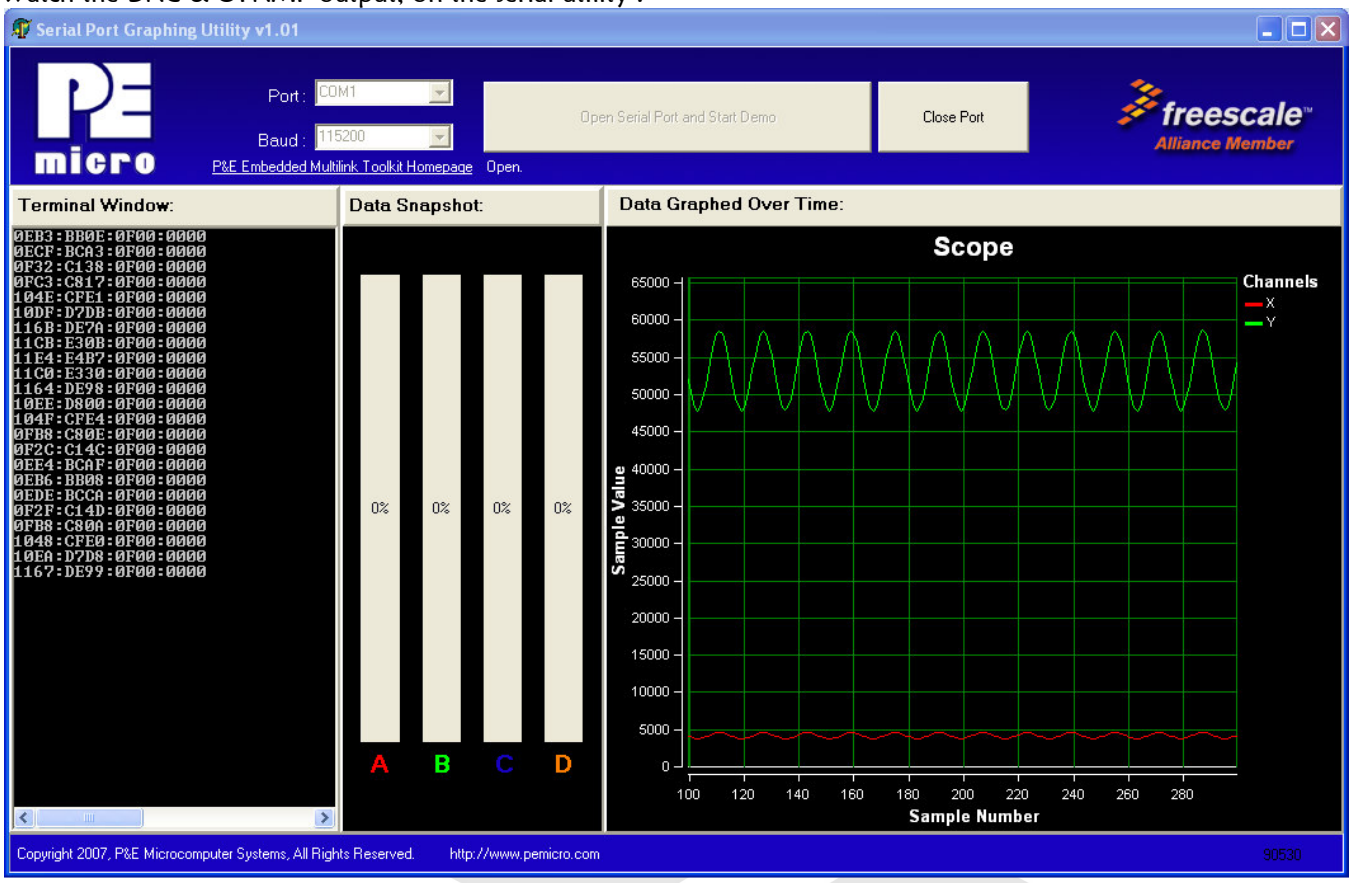
10. The **Debug Configurations** dialog box will pop up. Then in the File Options section, click on **LCD_MC9S08MM128_OpenSource BDM** and hit **Debug**



11. At the p&e serial grapher utility, open the port at COM1 with 115200 of baudrate.

12. Resume by clicking the play button (F8).

13. Watch the DAC & OPAMP output, on the serial utility .



14. Suspend the application 
15. Stop the debugger session 
16. Change to the C/C++ perspective 

17. In main.c, find the comment **/** Step6: Configure back-to-back operation */**
18. Go to the **vfnPDBInit** function by single clicking over the variable name and pressing **F3** :
19. Change the PDB delays and see what happens

    sPDB.gu16Modulus = →PDBMOD delay
    sPDB.gu16DACDelay = →DAC delay
    sPDB.gu16DelayA = →First ADC conversion delay
    sPDB.gu16IntDelay = →Interrupt periodicity

20. Repeat 7 – 16 steps