Programming Assignment #1 - Variable Blinking LED
CSE30 - Computer Organization and Systems Programming
Winter 2011

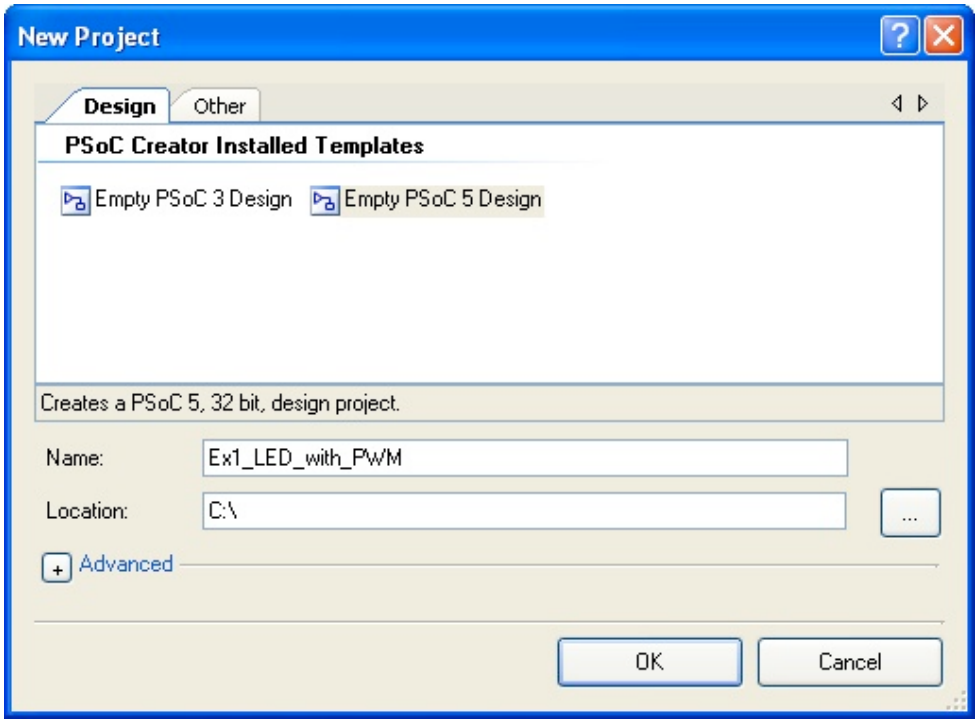
Overview: The goal of this assignment is to get you started on the Cypress PSoC platform. You will step through two given example projects; one that flashes LEDs using hardware and software methods to do it, and one that displays the voltage from the variable resistor on the LCD. After that, you will write code that changes the blinking rate of an LED based upon the position of the variable resistor.

Before you start, you should have installed all of the software for PSoC 5. Using the CD that comes with the kit is the easiest way to do this, as it will insure that all of the software you need is installed. You should run the Cypress Update Manager to make sure all your software is the most recent version.

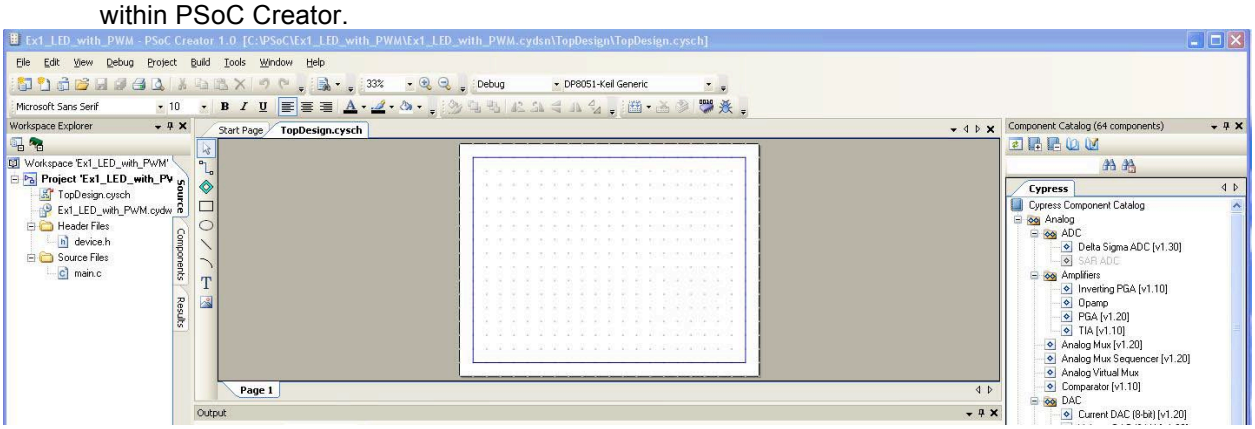
Example Project #1: Blinking LEDs

Overview: This project demonstrates basic hardware and software functionality with the PSoC 5 device. It flashes two LEDs independently, one using hardware, the other with software. The hardware LED uses a hardware enabled digital port and a PWM to generate a duty cycle and flash the LED. The software LED uses a software enabled digital port and a simple delay in the *main.c* to flash the LED at a known rate.

1. Open PSoC Creator
2. Create a new project by clicking **Create New Project…** in the **Start Page** of PSoC Creator.
3. In the **New Project** window, select the **Empty PSoC5 Design** template for a PSoC 5 design and name the project **Ex1_LED_with_PWM**.
4. In **Location**, type the path where you want to save the project, or click the  button and navigate to the appropriate directory.

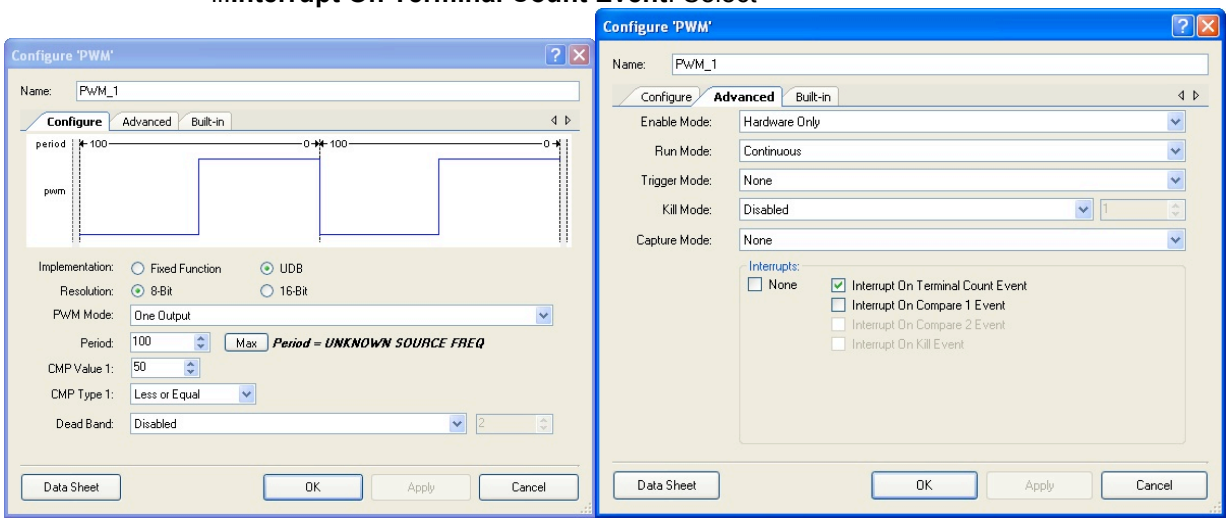


5. By default, the design window opens *TopDesign.cysch*. This is the project's schematic entry file within PSoC Creator.



Placing and Configuring the PWM

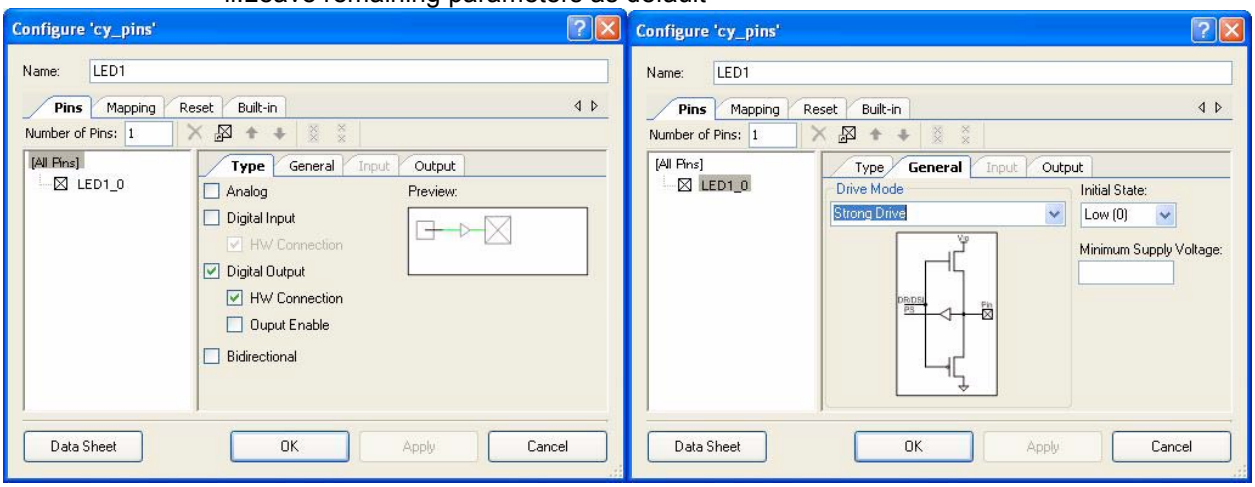
6. Drag-and-drop the **PWM** component (**Cypress Component Catalog → Digital → Functions → PWM**) toworkspace.
7. Double click the **PWM_1** component in the schematic to open the configuration window.
8. Configure the PWM in this manner:
    a. **Configure** Tab
        i. **Name**: PWM_1
        ii. **Implementation:** UDB
        iii. **Resolution**: 8-Bit
        iv. **PWM Mode**: One Output
        v. **Period**: 100
        vi. **CMP Value 1**: 50
        vii. **CMP Value Type 1**: Less or Equal
        viii. **Dead Band:** Disabled
    b. **Advanced** Tab
        i. **Enable Mode**: Hardware Only
        ii. **Interrupt On Terminal Count Event**: Select



9. For more information about what the parameters mean, click the **Data Sheet** button in the configuration window. This can be done for any component.

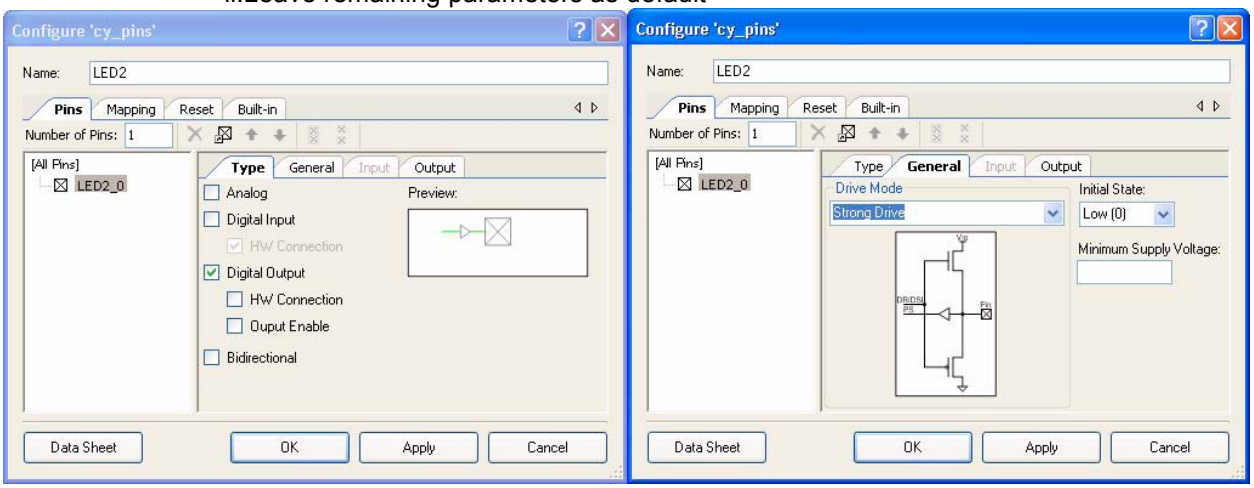
Placing and Configuring Digital Output Pin Hardware

10. Drag-and-drop the **Digital Output Pin** component (**Component Catalog → Ports and Pins → Digital Output Pin**).
11. Double click the **Pin_1** component in the schematic to open the configuration window.
12. Configure the digital output pin:
    a. **Type** Tab
        i. **Name**: LED1
        ii. Select **Digital Output** Check box
        iii. Select **HW Connection** Check box
    b. **General** Tab
        i. **Drive Mode**: Strong Drive
        ii. Leave remaining parameters as default



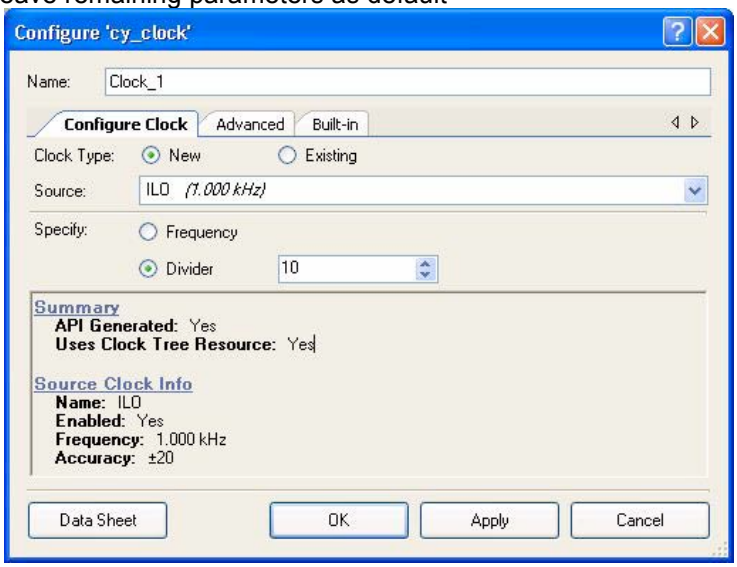
Placing and Configuring the Software Digital Output Pin

13. Drag-and-drop the **Digital Output Pin** component (**Component Catalog → Ports and Pins → Digital Output Pin**).
14. Double click the **Pin_1** component in the schematic to open the configuration window.
15. Configure the digital output pin:
    a. **Type** Tab
        i. **Name**: LED2
        ii. Select **Digital Output** Check box (Note all other check boxes are not selected).
    b. **General** Tab
        i. **Drive Mode**: Strong Drive
        ii. Leave remaining parameters as default

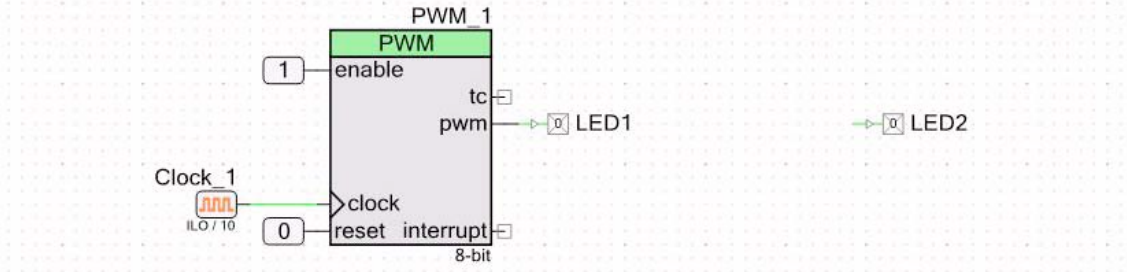


Connecting the Components Together

16. Using the Wire Tool  , connect **pwm** (in the PWM component) to hardware connection point of
17. LED1.
18. Connect a Logic High component (**Component Catalog → Digital → Logic → Logic High**) to the **enable** on the PWM
19. Connect a Logic Low component (**Component Catalog → Digital → Logic → Logic Low**) to the **reset** on the PWM
20. Connect a Clock component (**Component Catalog → System → Clock**) to the **clock** on the PWM.
21. Double click the **Clock_1** component to configure.
22. Configure the clock:
    a. **Configure Clock** Tab
        i. **Name**: Clock_1
        ii. **Source**: ILO (1.000 kHz)
        iii. Select **Divider** and set the value as 10
        iv. Leave remaining parameters as default

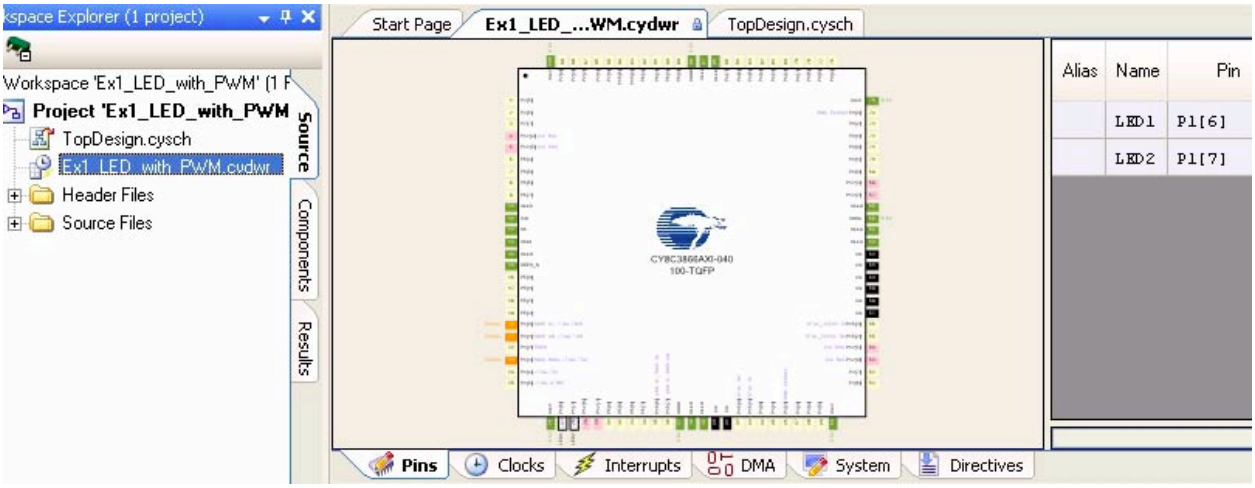


23. When complete the schematic looks similar to:



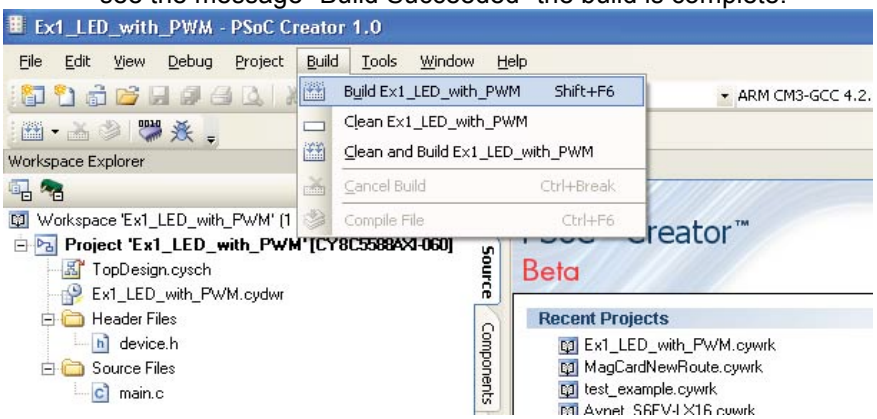
Configuring the Pins

24. From the **Workspace Explorer**, double click the *Ex1_LED_with_PWM.cydwr* file
25. Click the **Pins** tab.
26. Select pin P1[6] for LED1.
27. Select pin P1[7] for LED2.



Creating the main.c File

28. Open the existing *main.c* file within **Workspace Explorer**.
29. Replace the existing *main.c* content with the content of the embedded *PA1-LED.c* file, which can be found at: http://www.cse.ucsd.edu/~kastner/cse30/PA1-LED.c
30. From the **Build** menu, select **Build Ex1_LED_with_PWM**.
31. PSoC Creator builds the project and displays the comments in the **Output** dialog box. When you see the message "Build Succeeded" the build is complete.

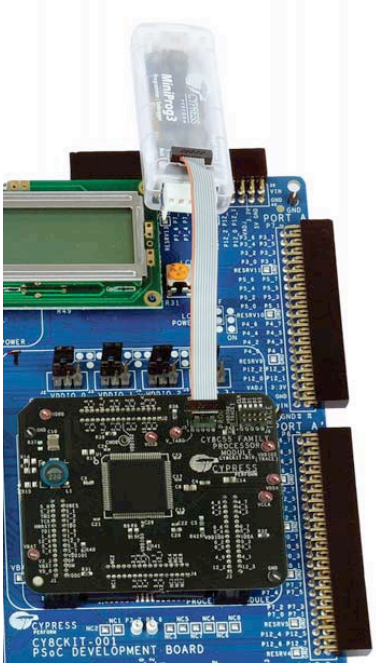


Configuring the PSoC Development Board

32. Disconnect the power to the board.
33. Configure the DVK SW3 to 3.3V.
34. Configure the following on the PSoC Development Board's prototyping area using the included
35. jumper wires
    a. P1[6] to LED1
    b. P1[7] to LED2

Programming the PSoC Development Board

36. Connect MiniProg3 to J5 (the pins that read PROG) on the CY8C55 Family Processor Module (the board cannot be programmed directly via the USB port on board)

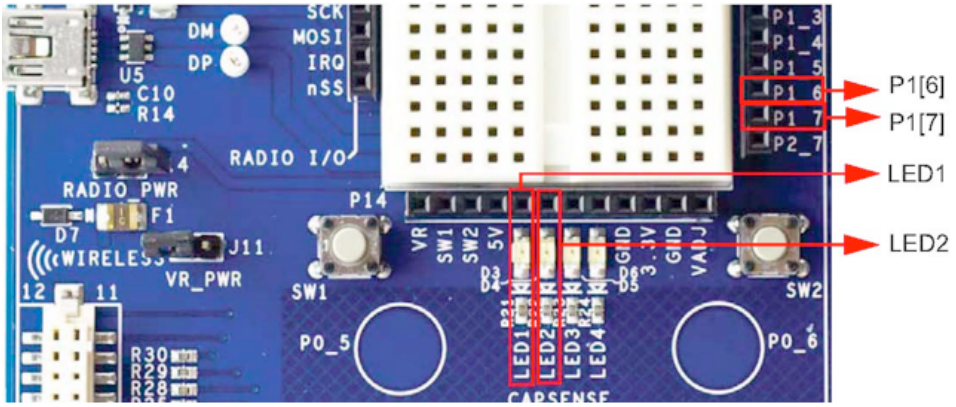


If this is your first time running PSoC Creator, follow these steps to configure the MiniProg3 device for these PSoC Development Kit projects. If these configurations are set, skip to the next step below and begin programming.

Note VTARG of the MiniProg3 is wired exclusively to VDDIO1 of the chip on the PSoC CY8C55 Family Processor Module. Because of this, you cannot perform power cycle mode programming.

37. From the **Tools** menu in PSoC Creator, click **Options**.
38. In the Options window, select **Programmer/Debugging** -**MiniProg3** from the list.
    a. Set **Applied Voltage** to **3.3V**
    b. Set **Transfer Mode** to **SWD**
    c. Set **Active Port** to **10 Pin**
    d. Set **Acquire Mode** to **Reset**
    e. Set **Debug Clock Speed** to **3.2 MHz**
    f. Click **OK**
39. From the **Debug** menu, select **Select Debug Target**.
    a. Expand the tree under **MiniProg3** and click **Port Acquire**.
    b. Select the appropriate device and click **Connect**.
    c. Click **Close**.

40. In PSoC Creator, from the **Debug** menu, click **Program**.
41. The PSoC Creator Status Bar indicates that the device is programming.
42. Wait until programming is complete before continuing.
43. Unplug the DVK board, switch SW3 to 3.3V and then reapply power to the DVK board.
44. Connect P1[6] to LED1 and P1[7] to LED2. Verify that LED1 and LED2 are blinking based on the project's use of the PWMs.
45. LED1 blinks approximately once every second and LED 2 blinks about three times a second.



Now that you have completed your first project, let's move on to a small introduction to using the other useful components on this board. In the following guided exercise, you will be using the potentiometer (the small black dial to the left of the touch-sensing sliders) to control the brightness of the LCD display.

Example Project #2: ADC to LCD Project

This project demonstrates the Delta Sigma ADC by measuring the voltage of the potentiometer on the board and displays the result on the Character LCD of the PSoC Development Board. The ADC is clocked by the internal clock of 3 MHz and the sampling rate is set to 10,000 sps. Connect the voltage potentiometer (labeled "VR" on the PSoC Development Board) to the ADC input (programmed to P0[7] for this example). The program reads the ADC result and prints it to the LCD.

This example project uses these components:

Delta Sigma ADC (**Component Catalog → Analog → ADC → Delta Sigma ADC**)
Character LCD (**Component Catalog → Display → Character LCD**)
Analog Pin (**Component Catalog → Ports and Pins → Analog Pin**)

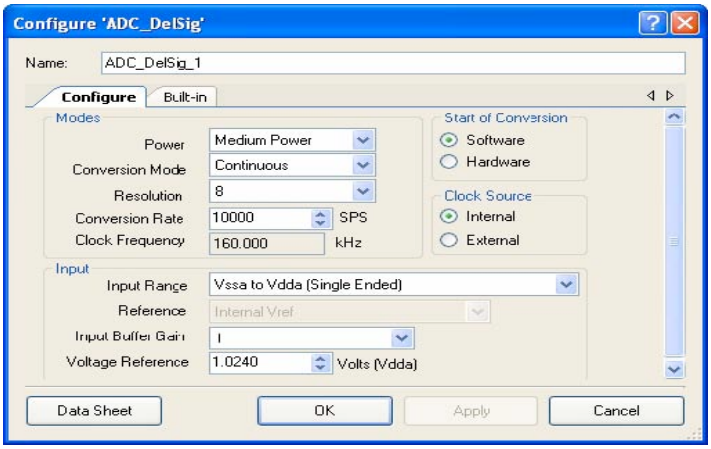
1. Open PSoC Creator.
2. Create a new project by clicking **Create New Project…** in the **Start Page** of PSoC Creator.
3. In the **New Project** window, select **Empty PSoC5 Design** template for a PSoC 5 design and **Name** the project **Ex2_ADC_to_LCD**.
4. In **Location**, type the path where you want to save the project, or click and navigate to the appropriate directory. By default, the design window opens *TopDesign.cysch*. This is the project's schematic entry file within PSoC Creator.

Placing and Configuring Delta Sigma ADC

5. Drag and drop the Delta Sigma ADC component (**Component Catalog → Analog → ADC → Delta Sigma ADC**)
6. Double click the **ADC_DelSig_1** component in the schematic to open the configuration window.
7. Configure the Delta Sigma ADC in this manner:
    a. **Configure** Tab
    b. **Name**: ADC_DelSig_1
    c. **Power**: Medium Power
    d. **Conversion Mode**: Continuous
    e. **Resolution**: 8
    f. **Conversion Rate**: 10000
    g. **Input Range**: Vssa to Vdda (Single Ended)
    h. **Input Buffer Gain**: 1
    i. **Reference**: Internal Vref

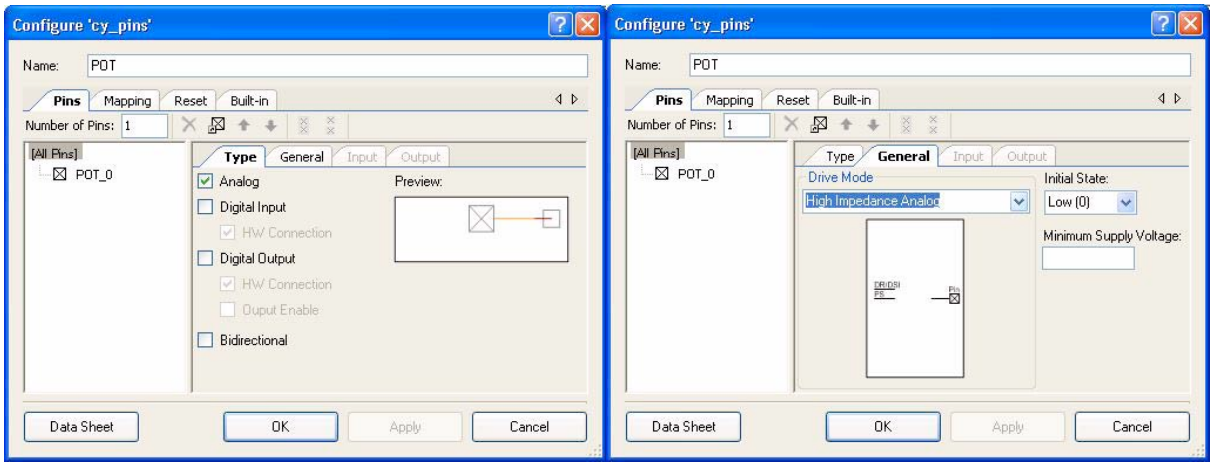
j. **Clock Source**: Internal
k. **Start of Conversion**: Software

For more information about what the parameters mean, click the **Data Sheet** button in the configuration window.



Placing and Configuring an Analog Pin

8. Drag-and-drop the analog pin component (**Component Catalog → Ports and Pins → Analog Pin**).
9. Double click on the **Pin_1** component in the schematic to open the configuration window.
10. Configure the analog pin in this manner
    a. **Type** Tab
        i. **Name**: POT
        ii. Select **Analog** check box only For more information about what the parameters mean, click the **Data Sheet** button in the configuration window.
    b. **General** Tab
        i. **Drive Mode**: High Impedance Analog
        ii. Leave remaining parameters as default.

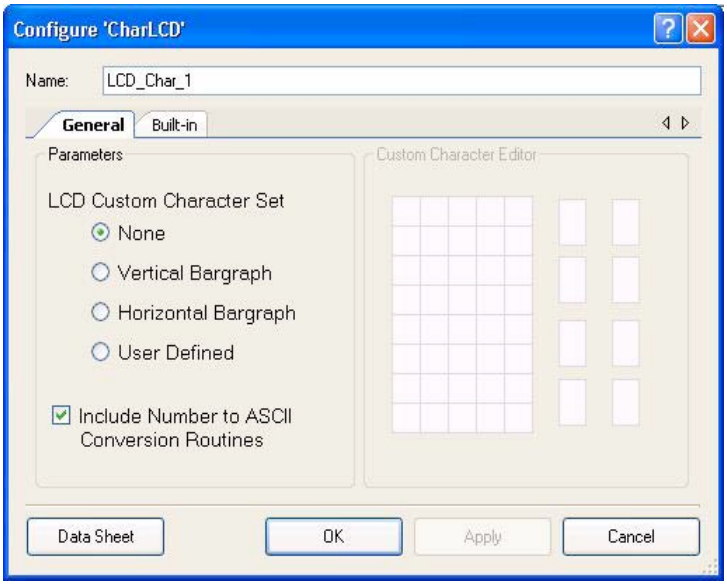


Placing and Configuring Character LCD

11. Drag-and-drop the Character LCD component (**Component Catalog → Display → Character LCD**)

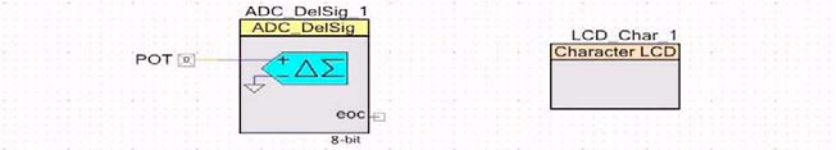
12. Double click the **LCD_Char_1** component in the schematic to open the configuration window.
13. Configure the Character LCD
    a. **LCD Custom Character Set**: None
    b. **LCD Custom Character Set**: None
    c. **Include ASCII to Number Conversion Routines**: Check box

For more information about what the parameters mean, click the **Data Sheet** button in the configuration window.



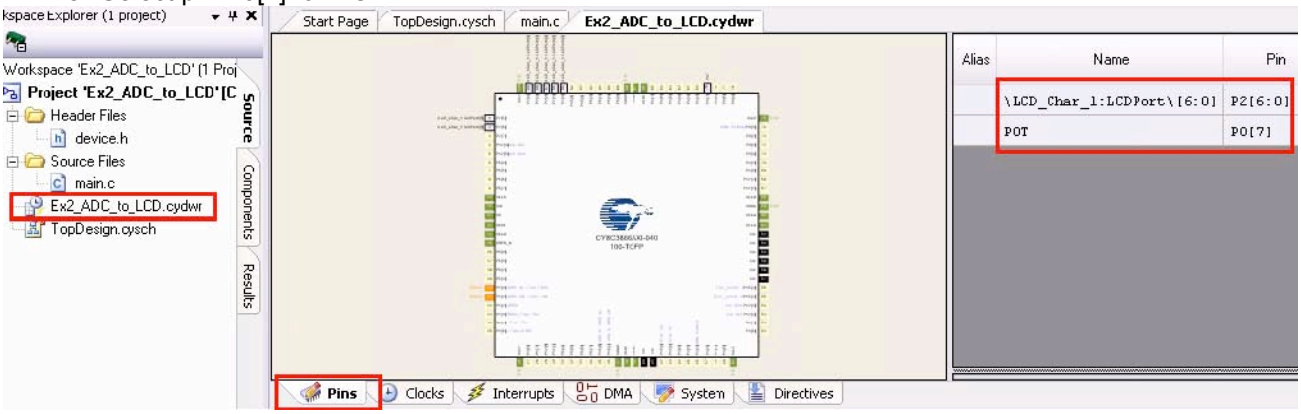
Connecting the Components Together

14. When complete, the schematic looks like the figure below.



Configuring the Pins

15. From the **Workspace Explorer**, double click the *Ex2_ADC_to_LCD.cydwr* file.
16. Click the **Pins** tab.
17. Select pins P2[6:0] for LCD_Char_1.
18. Select pin P0[7] for POT



Creating the main.c File

19. Open the existing *main.c* file within **Workspace Explorer**.
20. Replace the existing *main.c* content with the content of the PA1-VR.*c* file, which can be found at http://cseweb.ucsd.edu/~kastner/cse30/PA1-VR.c
21. From the **Build** menu, select **Build Ex2_ADC_to_LCD**. PSoC Creator builds the project and displays the comments in the **Output** dialog box. When you see the message "Build Succeeded" the build is complete.

Configuring and Programming the PSoC Development Board

22. Disconnect power to the board.
23. Using the jumper wires included, configure the PSoC Development Board's prototyping. Connect P0[7] to VR
24. Verify that VR_PWR (J11) is jumpered to ON.
25. Apply power to the board
26. Use PSoC Creator as described in the previous project to program the device.
27. After programming the device, press the **Reset** button on the PSoC Development Board to see the output of the ADC displayed on the LCD. Turning the potentiometer results in the LCD value changing.

Finally, the assignment

Now that you have done these two examples, combine the two projects to toggle the rate of one LED blinking based upon the position of the potentiometer. This means that as you turn the potentiometer, the LED blinks at a faster/slower rate. Twisting the potentiometer clockwise makes the LED blink slower (this is when the ADC has a higher value), and twisting it counter clockwise increases the rate of the LED blinking (when the ADC has a lower value).

Check the forum for a video demonstration of what we expect.