External (Off-Chip) Components

Introduction

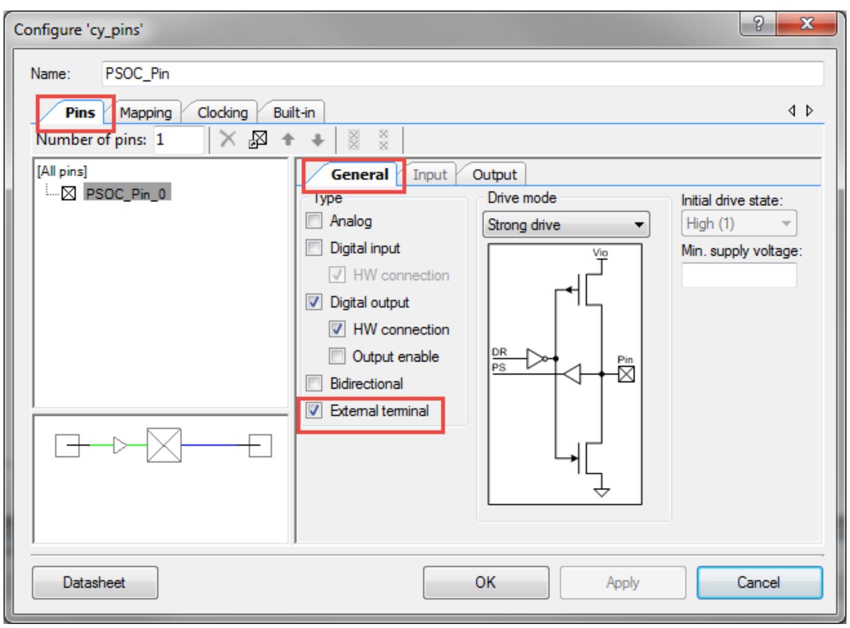
Description: PSoC designs get interesting when you connect them to the outside world. The Pins component can drive LEDs, read switches, connect a voltage source to an ADC, and so on, but all these things are on the board, not in the PSoC. There is no LED inside the PSoC! Because of this, you can make a design that works perfectly but another engineer would have no idea what it does because they don't know what you are connecting to (the rest of the board).

External components are a solution to this problem. They are a collection of components that people commonly use on boards – things like LEDs, sensors, diodes, rectifiers, and so on. You use them to show the off-PSoC connections and they really help people understand the intent of a design. Please keep in mind that these external components are there just to make the design easier to understand. These components are outside the PSoC, not inside. For example, placing an LED component on your schematic doesn't automatically connect it to the PSoC – you need a PSoC pin to drive the LED.

If you are going to use external components, and we recommend that you do, you need to enable the connection to the pin. When you do that, you'll notice that the pin has two connections – one internal and one external. When you use external components you stop thinking of pins as an end point, and more of a portal to the outside world.

Configuration Dialog:

To expose the external terminal on the schematic check the box in the Pins component dialog.



When you do not check on "External terminal" the pin looks like this – there is only an internal connection. The wire is green for digital connections and yellow for analog connections.

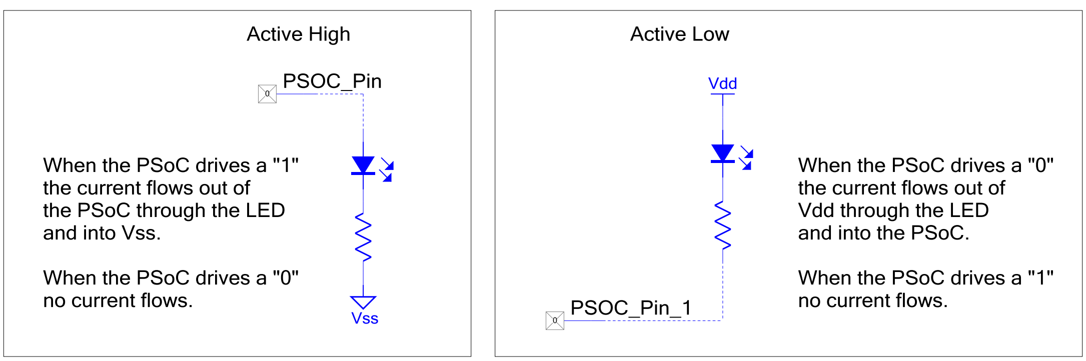
When showing the terminal external components can be connected – making your design easier to understand. The external connection is always shown in blue.

Light Emitting Diode (LED)

Description: An LED is a device that glows when you pass electrical current through it. The brightness of the LED depends on the amount of current that passes through it. If you pass too much current through the LED it will blow up (think fire and smoke). In general, LEDs are connected in series with a resistor that limits the amount of current (remember Ohms law? V=IR … look at the schematic below). You can vary the brightness of the LED by either controlling the input voltage (which limits the current) or by “blinking” the LED faster than the human eye can see (see PWM). An LED is connected to a PSoC in one of two ways:

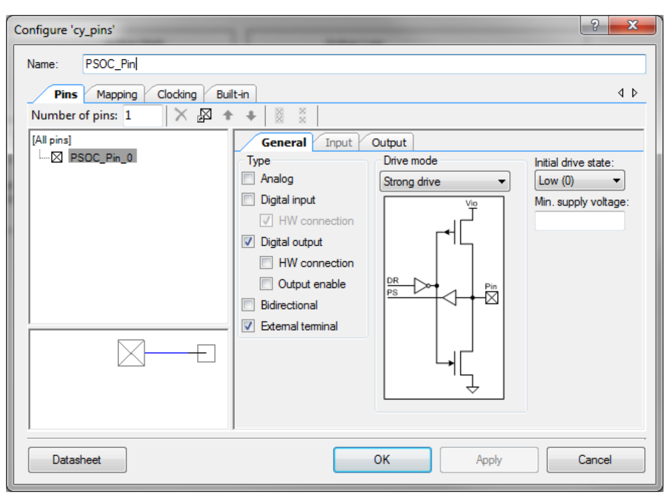
- Active High (driving the LED with a 1 lights it up)
- Active Low (driving the LED with a 0 lights it up)

Schematic



PSoC Interface

In order to connect an LED to a PSoC you should connect it to a “digital pin” (read the pin section and the pin datasheet) setup as “digital output” and “strong drive.”

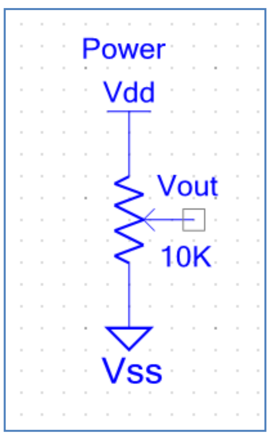


In this case, we have turned off the output's "HW connection" which is the pin's schematic connection inside PSoC since we want to control the pin with the firmware. In the figures above, the green line and box to the pin go away because there is no schematic connection. See the pin section in the internal components chapter for more information about hardware (HW) vs. firmware connections.

Potentiometer (a.k.a. Pot)

Description: A Pot is a 3 terminal electromechanical (meaning that mechanical movements cause electrical actions) analog device connected to power and ground, and having a single output. The output voltage varies between ground and power based on the position of the dial. Mechanically, a pot uses a sliding contact along a resistor to form an adjustable resistor voltage divider. A pot may be thought of as simply an analog voltage reference source.

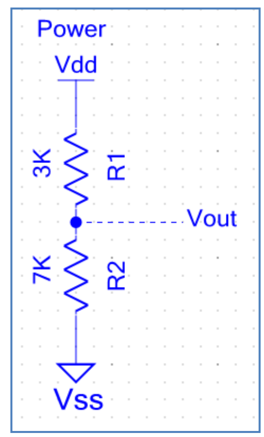
The arrow in the Pot schematic symbol represents the variable contact, controlled by turning the Pot dial. As the contact slides toward the power rail (Vdd), the output voltage (Vout) rises higher. As the contact slides toward ground (Vss), Vout drops.



For example, suppose the dial is set so that the sliding contact is closer to the top power rail. In that case, the resistance between Vout and Vdd will be smaller than the resistance between Vout and Vss, as shown in the figure to the right. Remember the voltage divider equation?

$$Vout = Vdd\left(\frac{R2}{R1+R2}\right) = 3.3\left(\frac{7}{3+7}\right) = 2.31V$$

The output is 7/10ths of the power rail in this example.

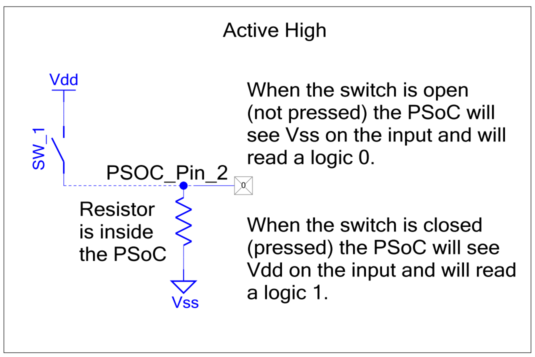


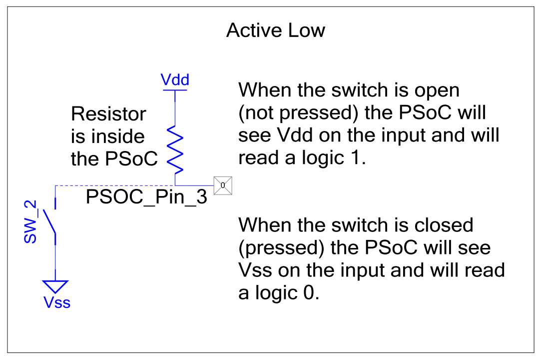
Mechanical Switch

Description: A Mechanical Switch (Switch) is an electromechanical device that electrically connects two terminals when the “button” is pressed. When connecting switches to a PSoC, the pin is typically configured inside the PSoC using a resistor that pulls the PSoC input to either Vdd or Vss. This is done so a separate resistor is not required on the board. On Cypress DVK's switches are typically active low and do not have a separate resistor so the pin should be configured as an input with resistive pullup as shown below.

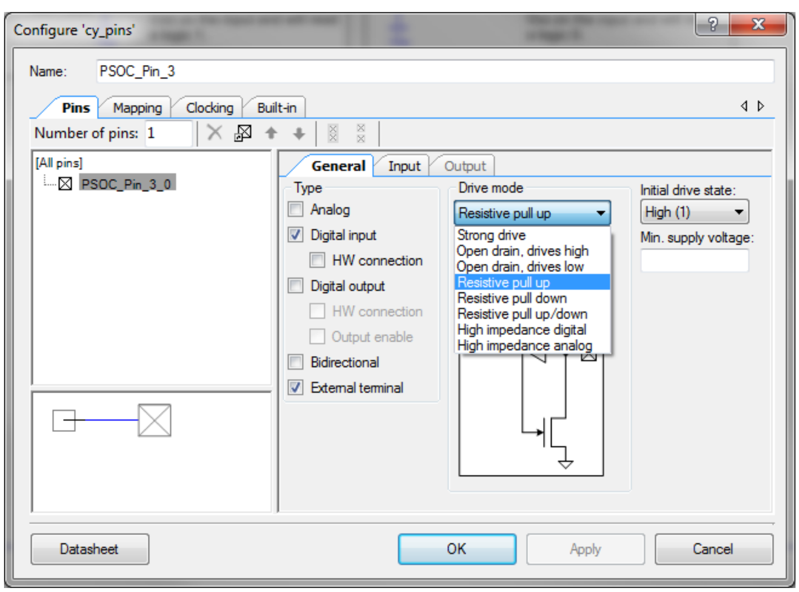
This circuit can be configured as:

- Active High (when the button is pressed the PSoC reads 1)
- Active Low (when the button is pressed the PSoC reads 0)





The circuit inside the PSoC has the ability to provide the pullup or the pulldown resistor. This is selected on the “Drive Mode” menu of the pin configuration wizard. When you use a resistive pullup the initial state must be set to “High(1)” which Creator does for you automatically. When you use resistive pulldown you need to set the initial state to “Low (0)” which Creator also does automatically.



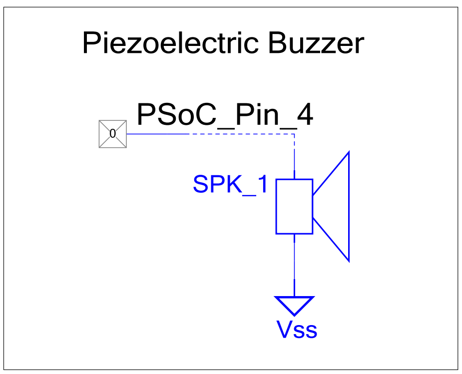
PSoC kits typically use active low switches, so you will need to configure the inputs as resistive pullup.

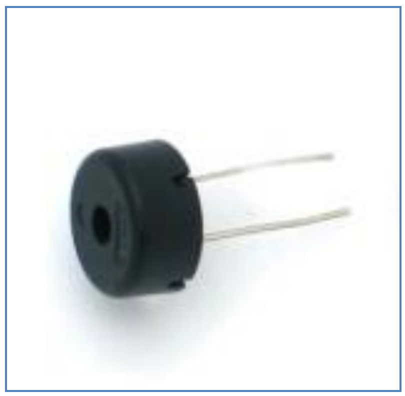
Buzzer (a.k.a. Piezoelectric buzzer)

Description: A buzzer or in our case, a piezoelectric buzzer, is a 2 terminal, electromechanical device that will vibrate (move air) at the same frequency as the electrical input. For instance, if you want to create a “middle C” you would input a 261.626 Hz signal into the buzzer. Piezoelectric buzzers work best if the input signal is a square wave (for instance from a PSoC Clock). Using a PSoC, there are many methods of driving a square wave onto a pin, including:

- Firmware (writing the C program to alternate 1’s and 0’s
- Clock component

The schematic for a buzzer circuit often looks like this (the off-chip component used is “speaker”):





To drive the pin it is probably best to setup the output as a “digital output” and “strong drive” (just like the LED).

Note	Hz	Note	Hz	Note	Hz	Note	Hz	Note	Hz	Note	Hz	Note	Hz
C1	32.7	C2	65.4	C3	130.8	C4	261.6	C5	523.3	C6	1046.5	C7	2093.0
C#1	34.6	C#2	69.3	C#3	138.6	C#4	277.2	C#5	554.4	C#6	1108.7	C#7	2217.5
D1	36.7	D2	73.4	D3	146.8	D4	293.7	D5	587.3	D6	1174.7	D7	2349.3
D#1	38.9	D#2	77.8	D#3	155.6	D#4	311.1	D#5	622.3	D#6	1244.5	D#7	2489.0
E1	41.2	E2	82.4	E3	164.8	E4	329.6	E5	659.3	E6	1318.5	E7	2637.0
F1	43.7	F2	87.3	F3	174.6	F4	349.2	F5	698.5	F6	1396.9	F7	2793.8
F#1	46.2	F#2	92.5	F#3	185.0	F#4	370.0	F#5	740.0	F#6	1480.0	F#7	2960.0
G1	49.0	G2	98.0	G3	196.0	G4	392.0	G5	784.0	G6	1568.0	G7	3136.0
G#1	51.9	G#2	103.8	G#3	207.7	G#4	415.3	G#5	830.6	G#6	1661.2	G#7	3322.4
A1	55.0	A2	110.0	A3	220.0	A4	440.0	A5	880.0	A6	1760.0	A7	3520.0
A#1	58.3	A#2	116.5	A#3	233.1	A#4	466.2	A#5	932.3	A#6	1864.7	A#7	3729.3
B1	61.7	B2	123.5	B3	246.9	B4	493.9	B5	987.8	B6	1975.5	B7	3951.1