

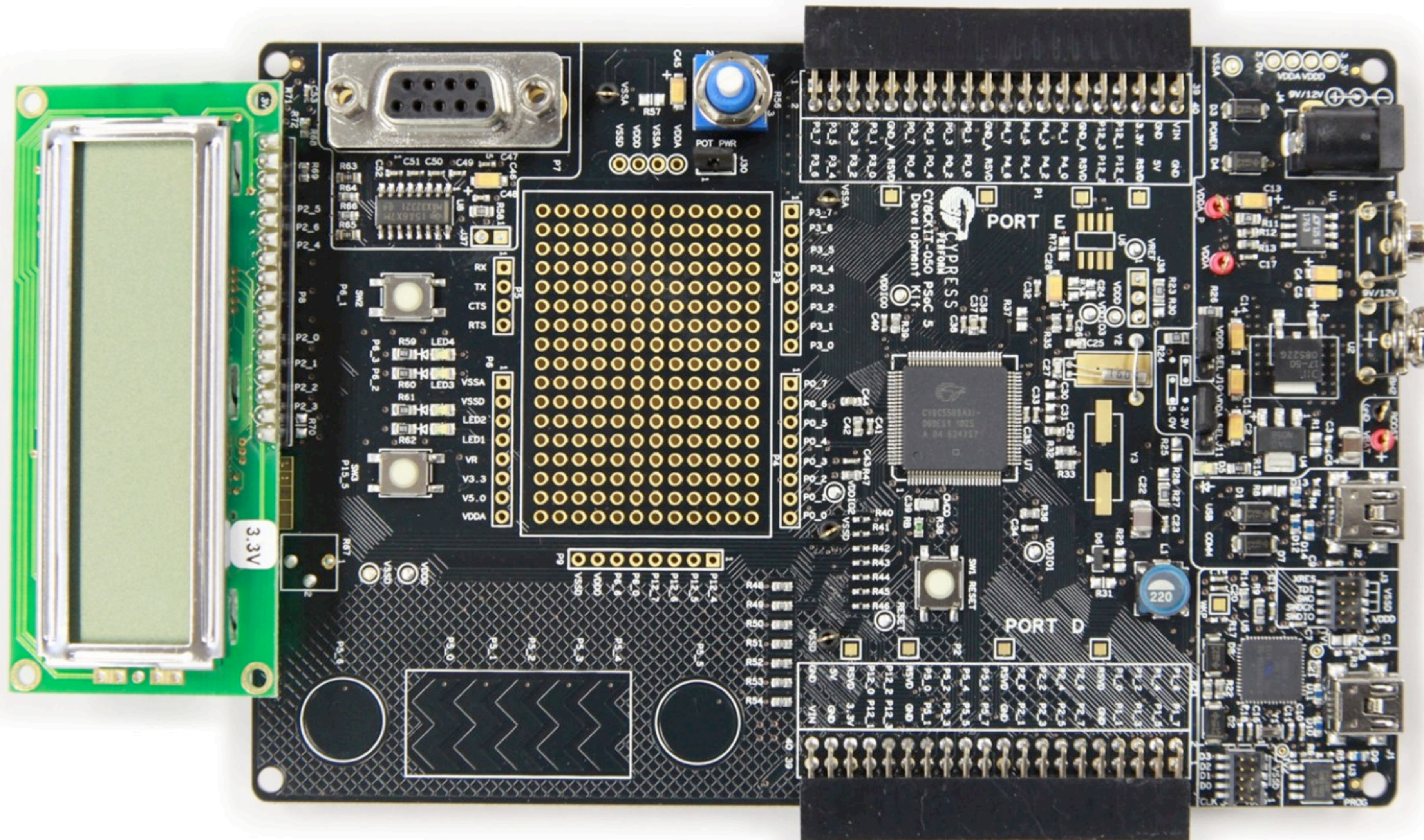
Introduction to PSoC

CSCE 689 - Sensory Interfaces

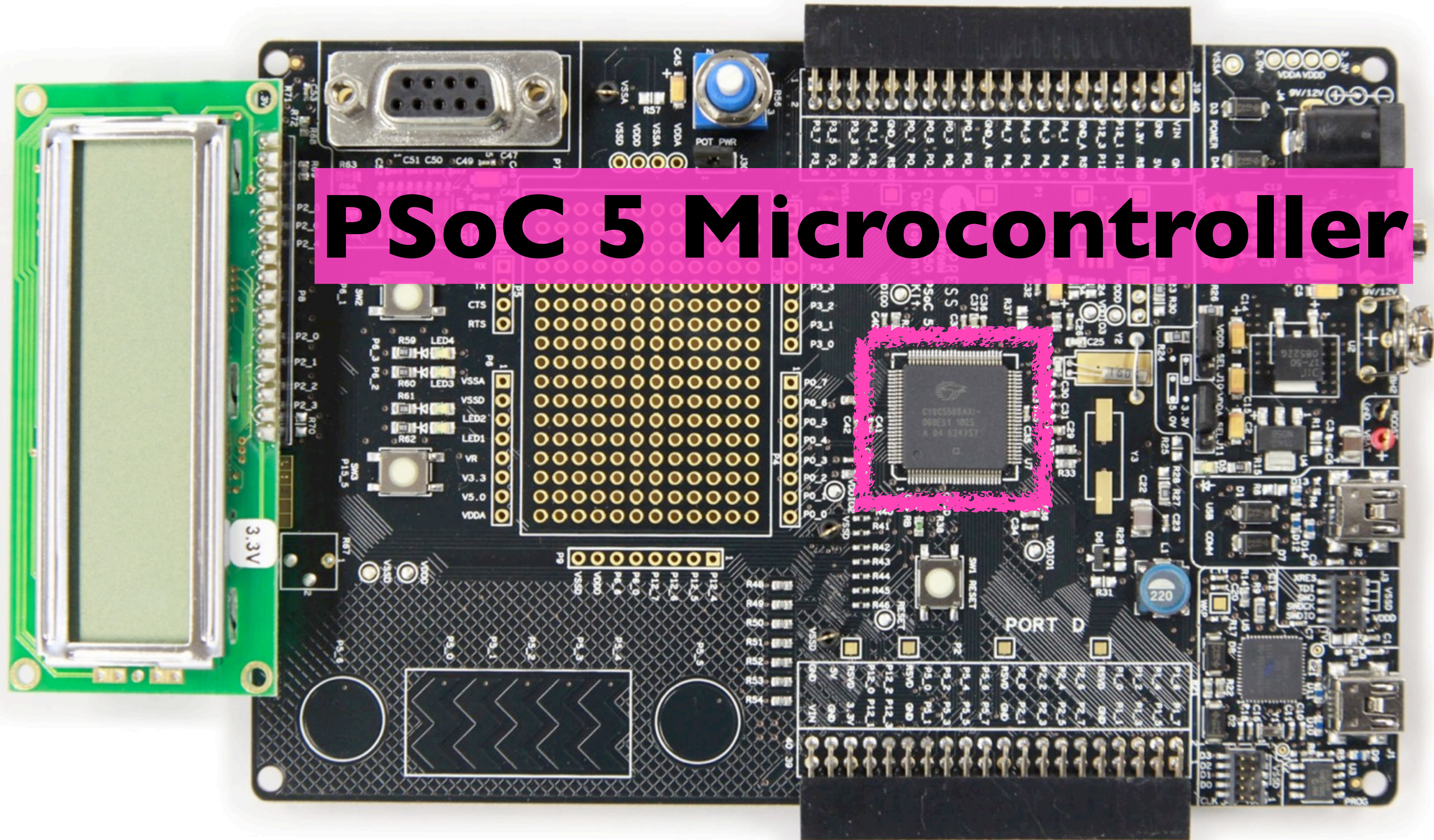
What is a PSoC?

- **Programmable System on a Chip**
- 8051 or ARM Microcontroller
- Digital Block Array
 - 24 configurable PLDs, can take on almost any digital functionality up to 67 MHz
- Configurable Analog Subsystems
- Extremely reconfigurable inter-chip routing network, allowing any internal component to connect to nearly any external pin
- **Easy-to-use APIs for all of the above, enabling you to use complex functionality without needing to understand the hardware complexities underneath.**

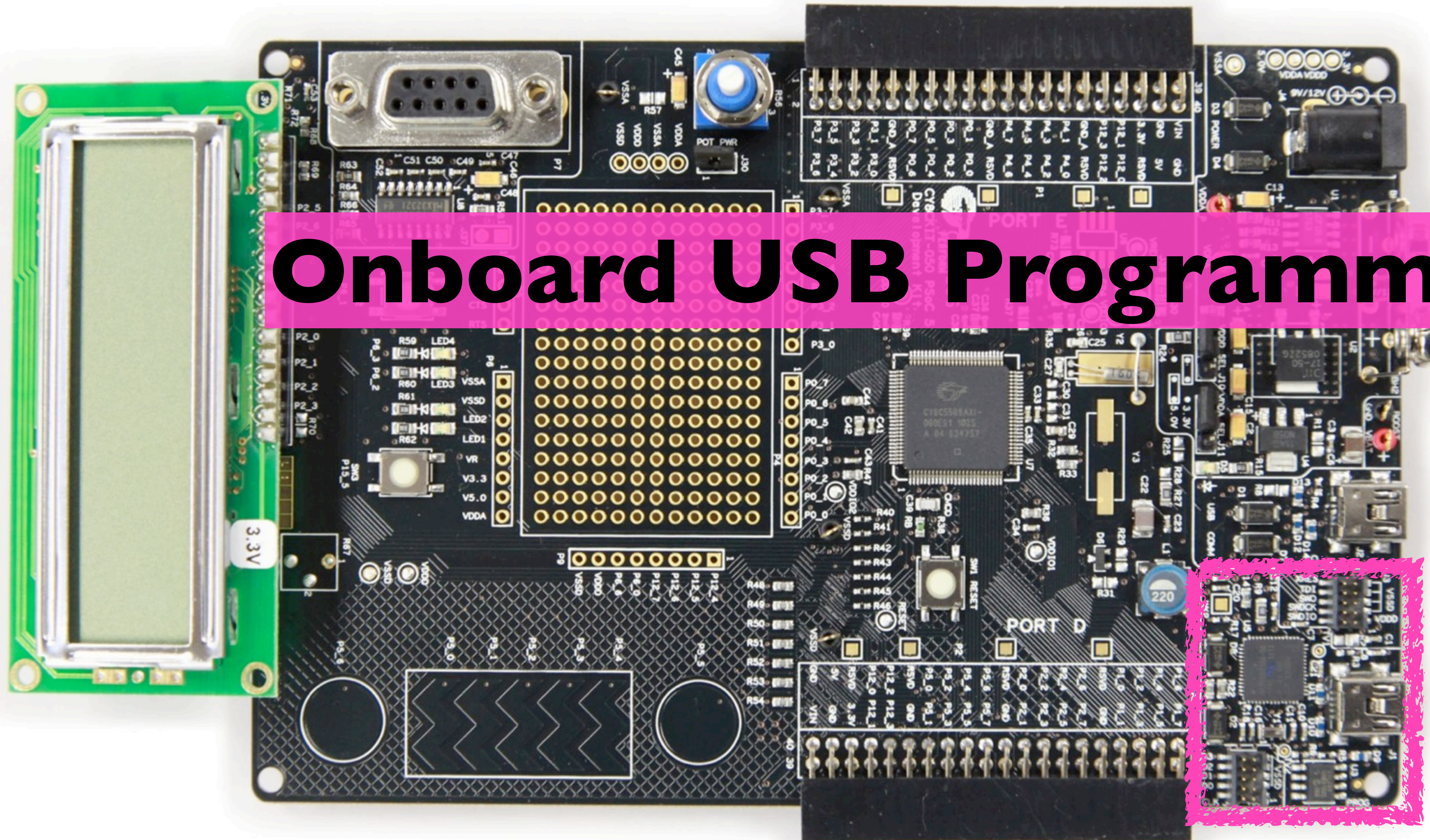
CY8CKIT-050



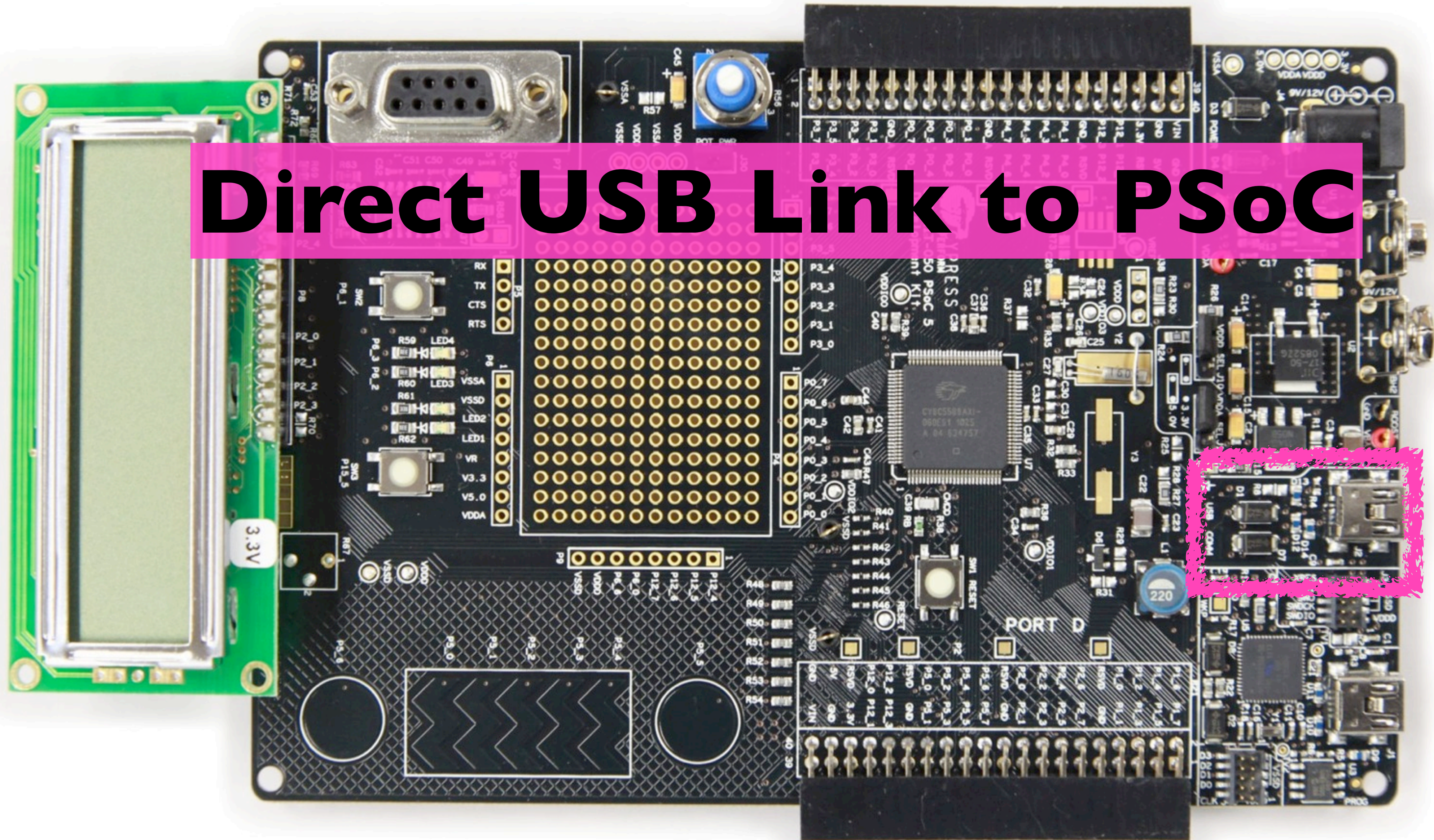
PSoC 5 Microcontroller

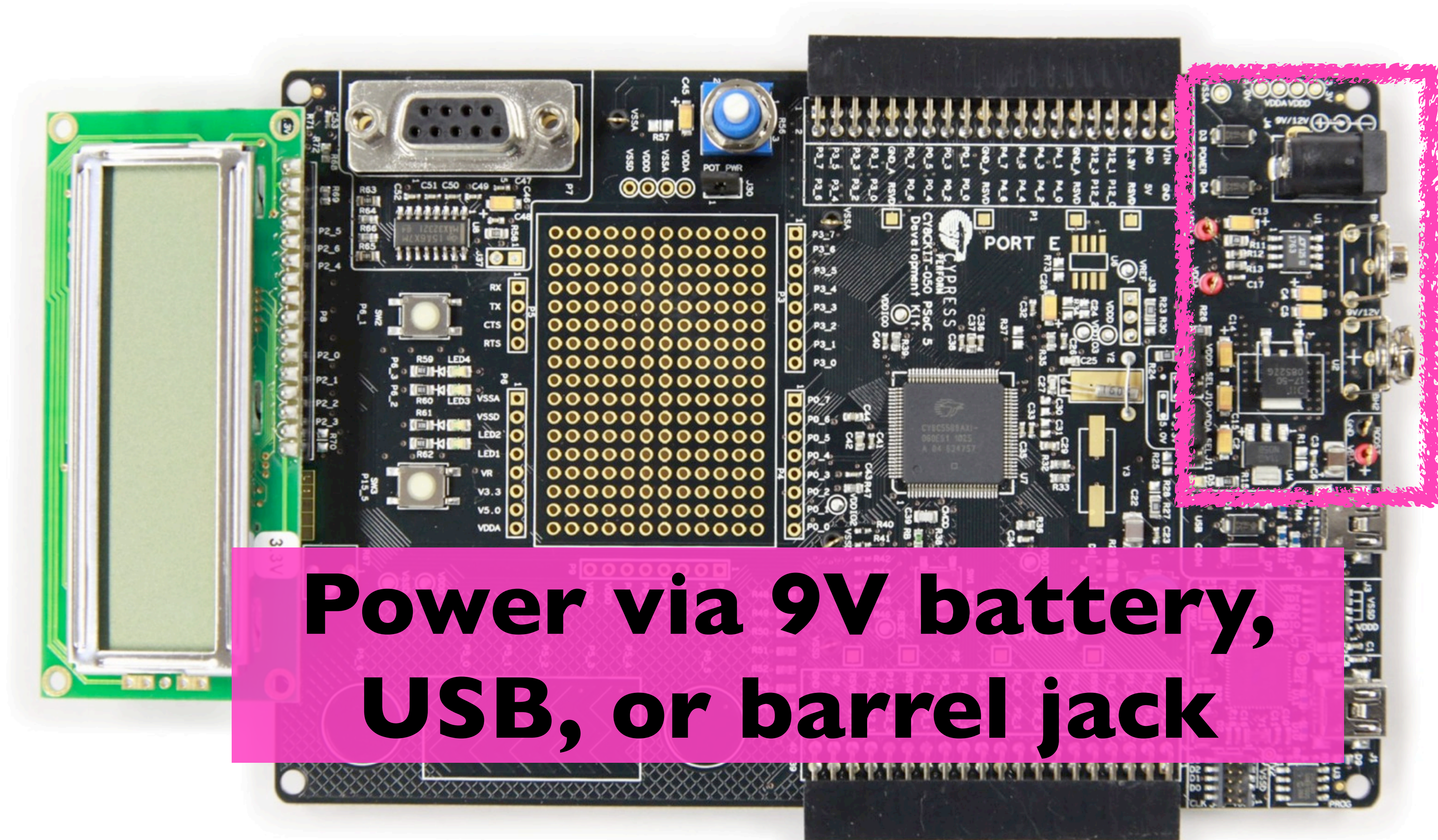


Onboard USB Programmer



Direct USB Link to PSoC







Expansion Headers for Connecting External Components (52 pins total)

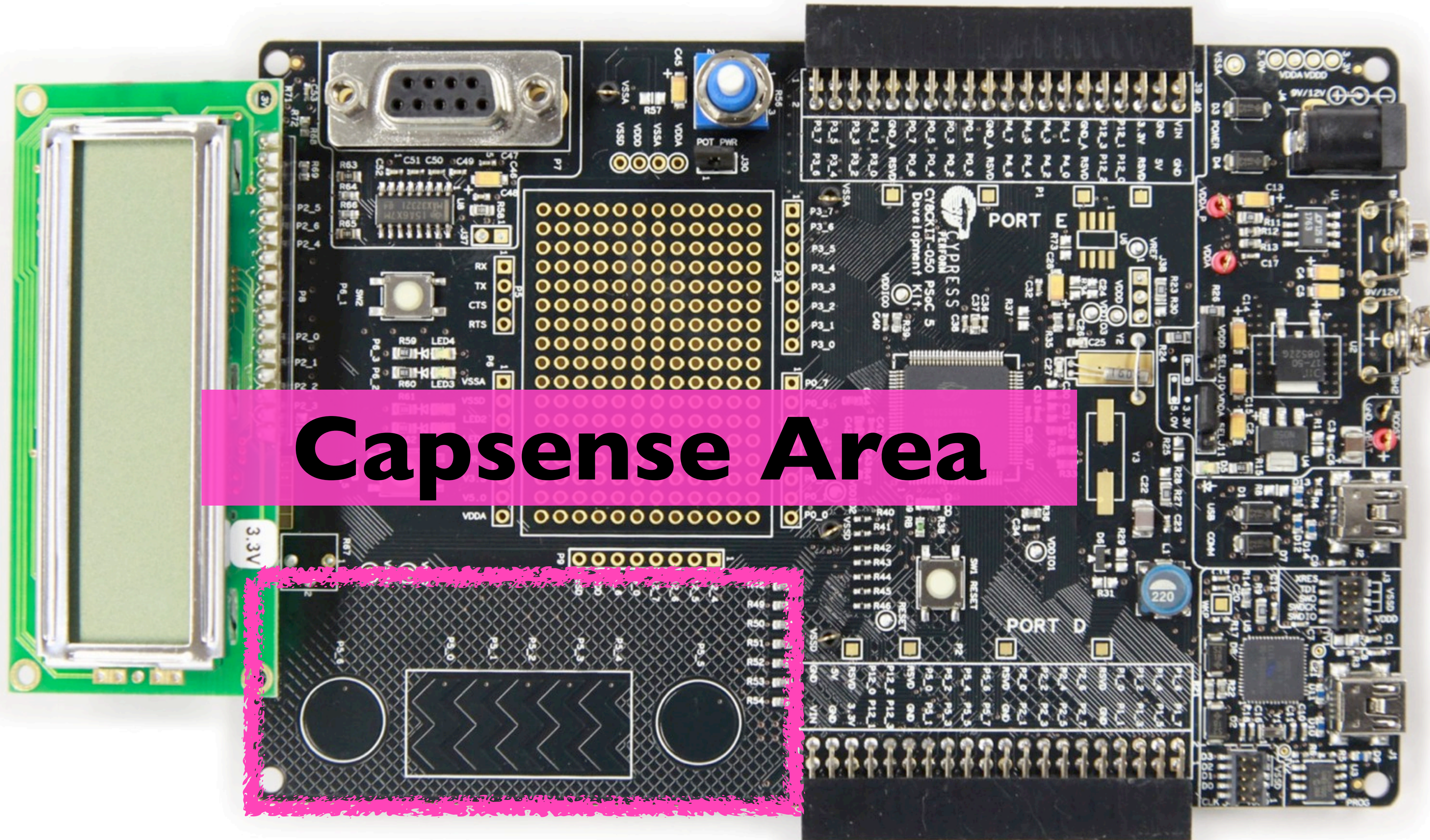


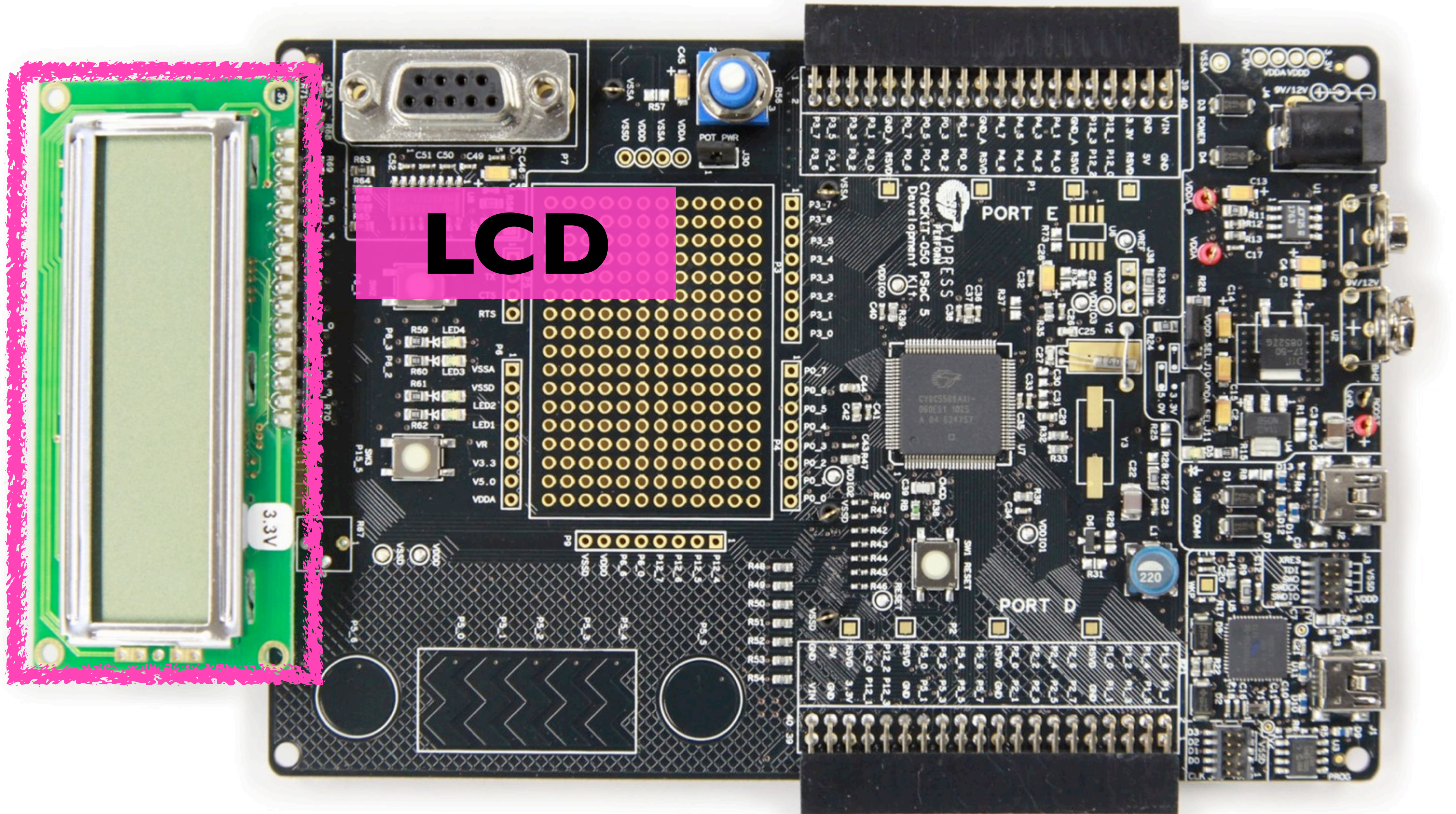
The image shows two Arduino Uno boards. The top board is a standard black PCB with a green USB Type-B connector, a DC power jack, and a pin header. The bottom board is a modified version with a green PCB, a USB Type-A connector, and a pin header. Both boards have their pin headers highlighted with a green border. The text "Analog Optimized Pins" is overlaid on the top board's header, and "Digital Pins" is overlaid on the bottom board's header.

Analog Optimized Pins

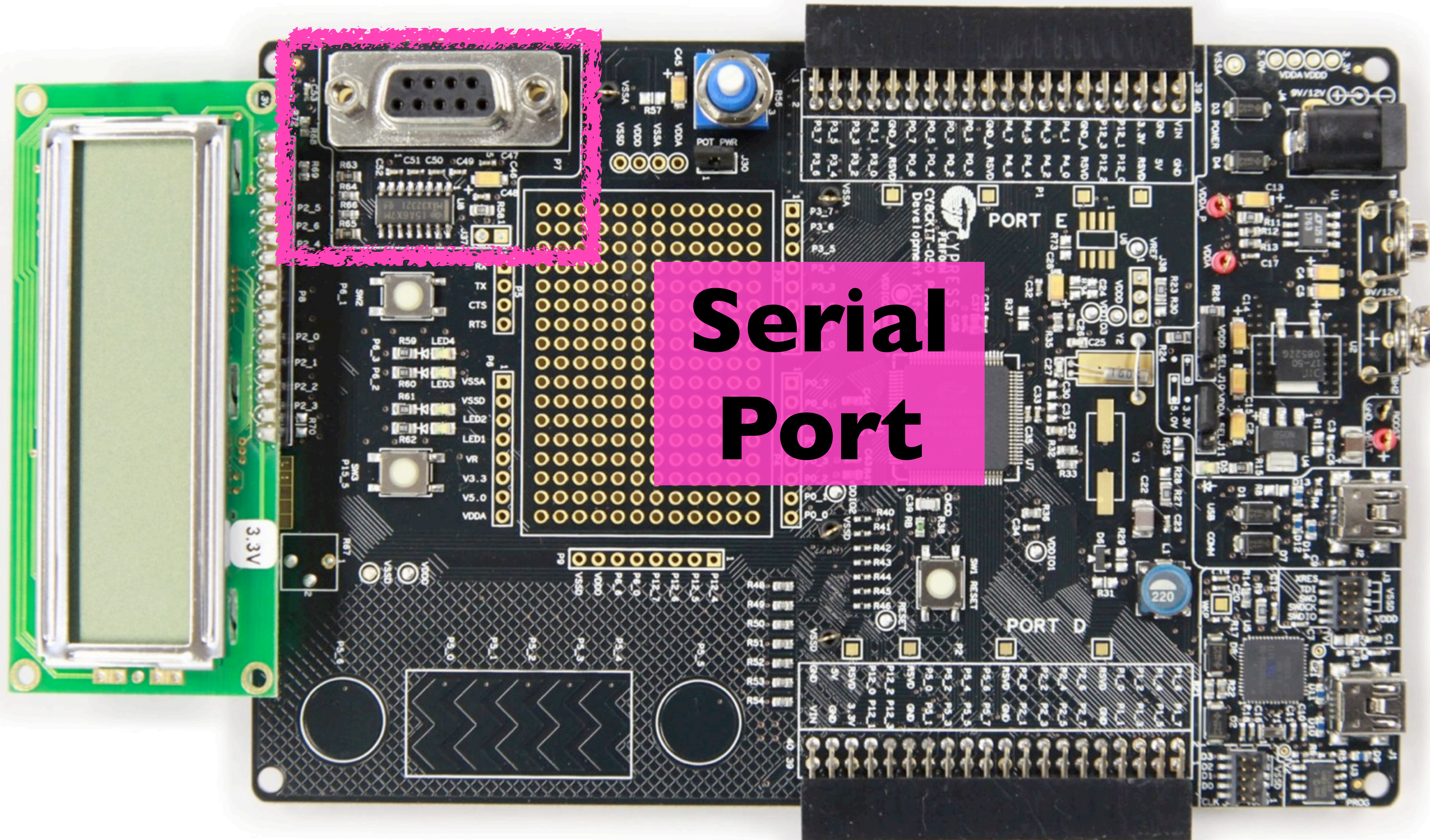
Digital Pins

Capsense Area

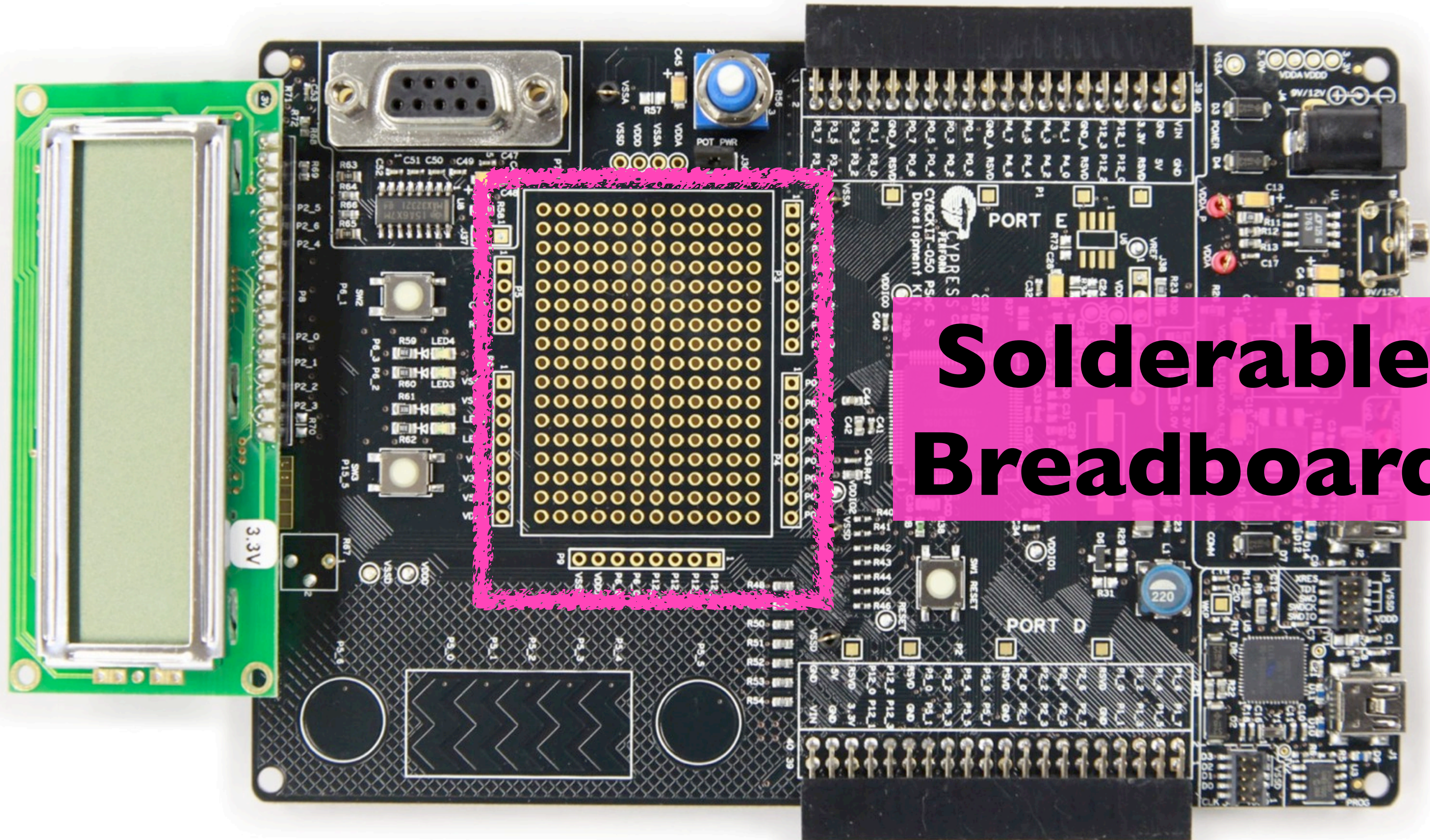




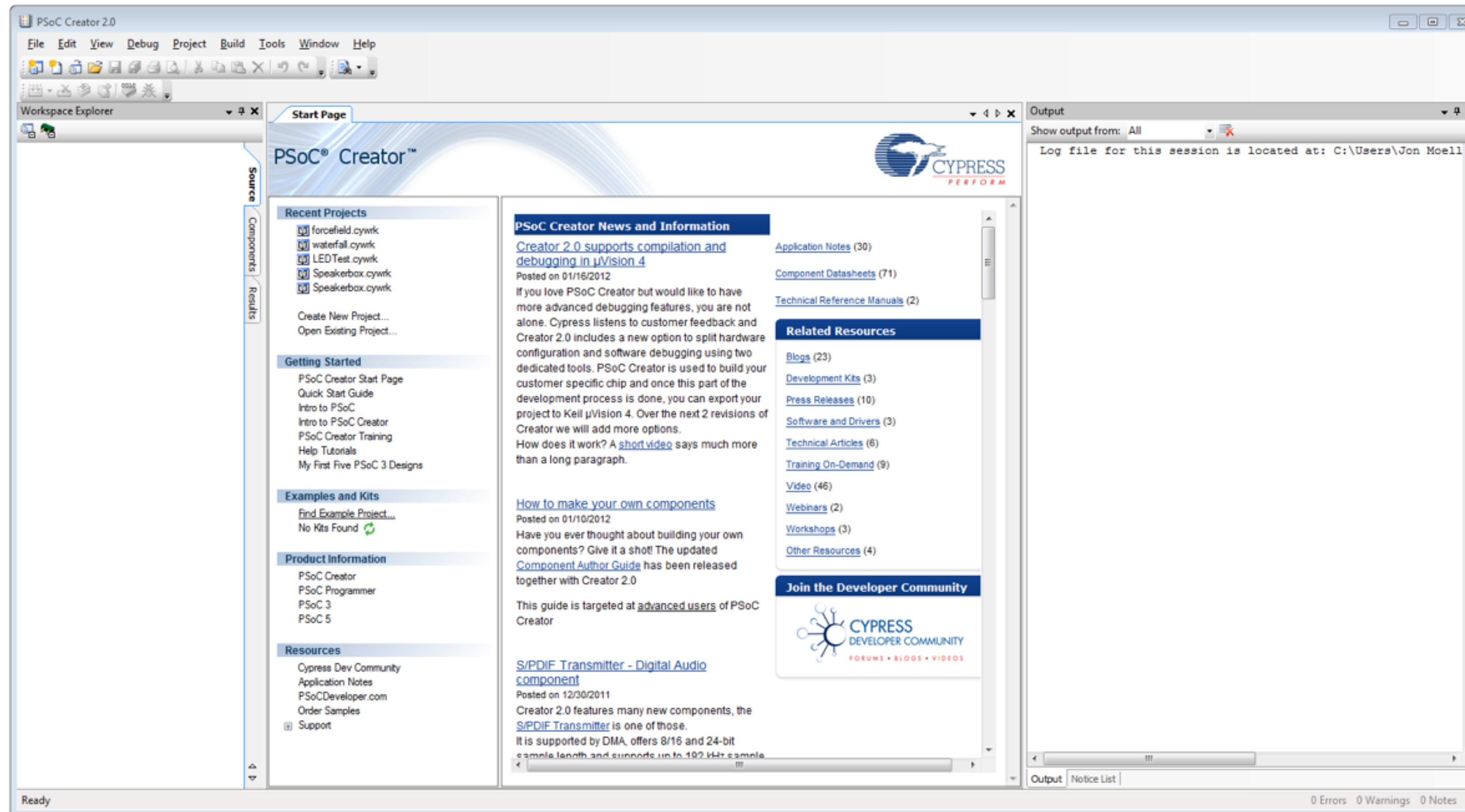
Serial Port



Solderable Breadboard



PSoC Creator 2.0



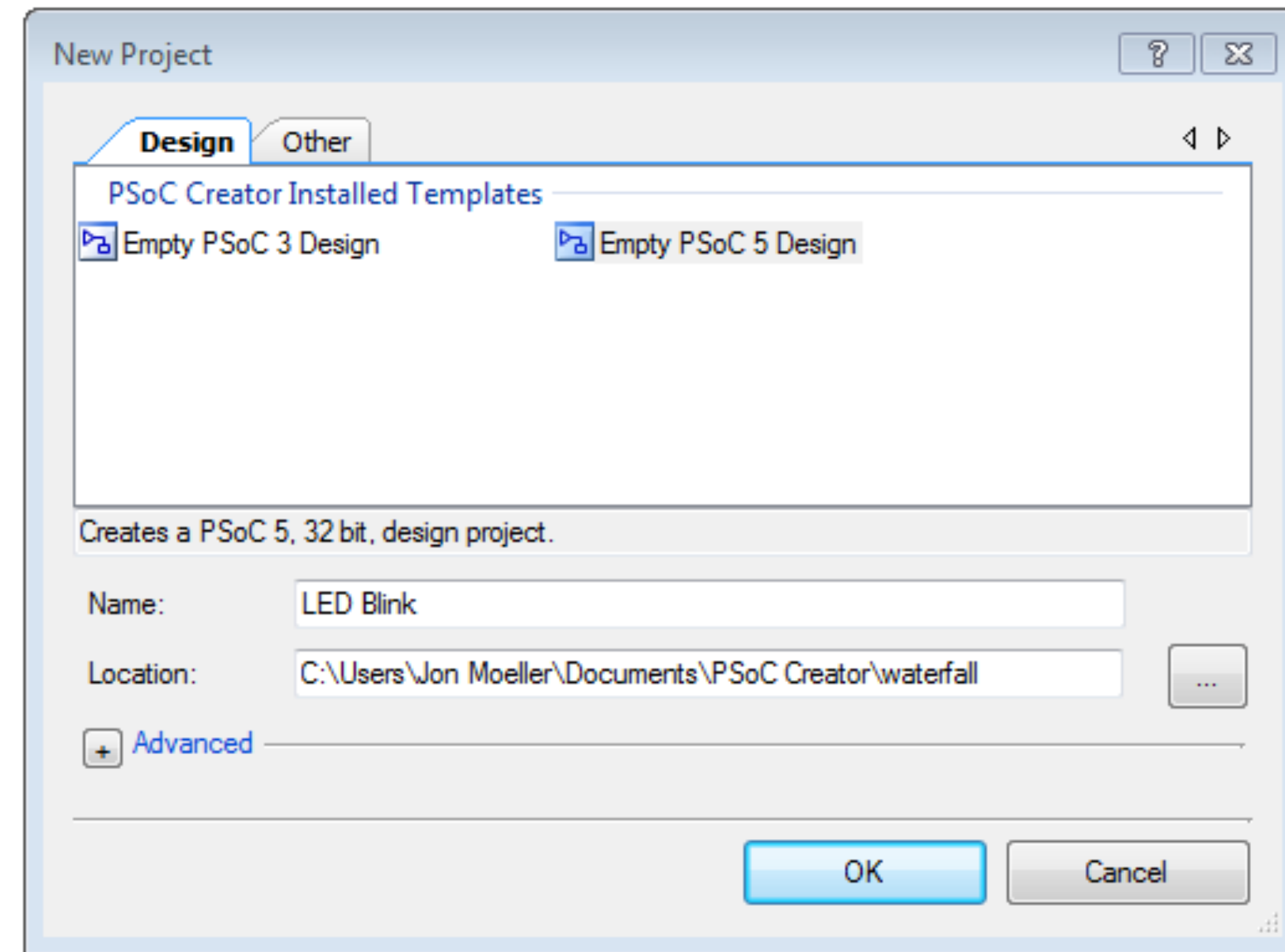
Download at <http://cypress.com/> (registration required)

Or, get the ISO file from \\chevron\installs\psoc

First Homework Assignment

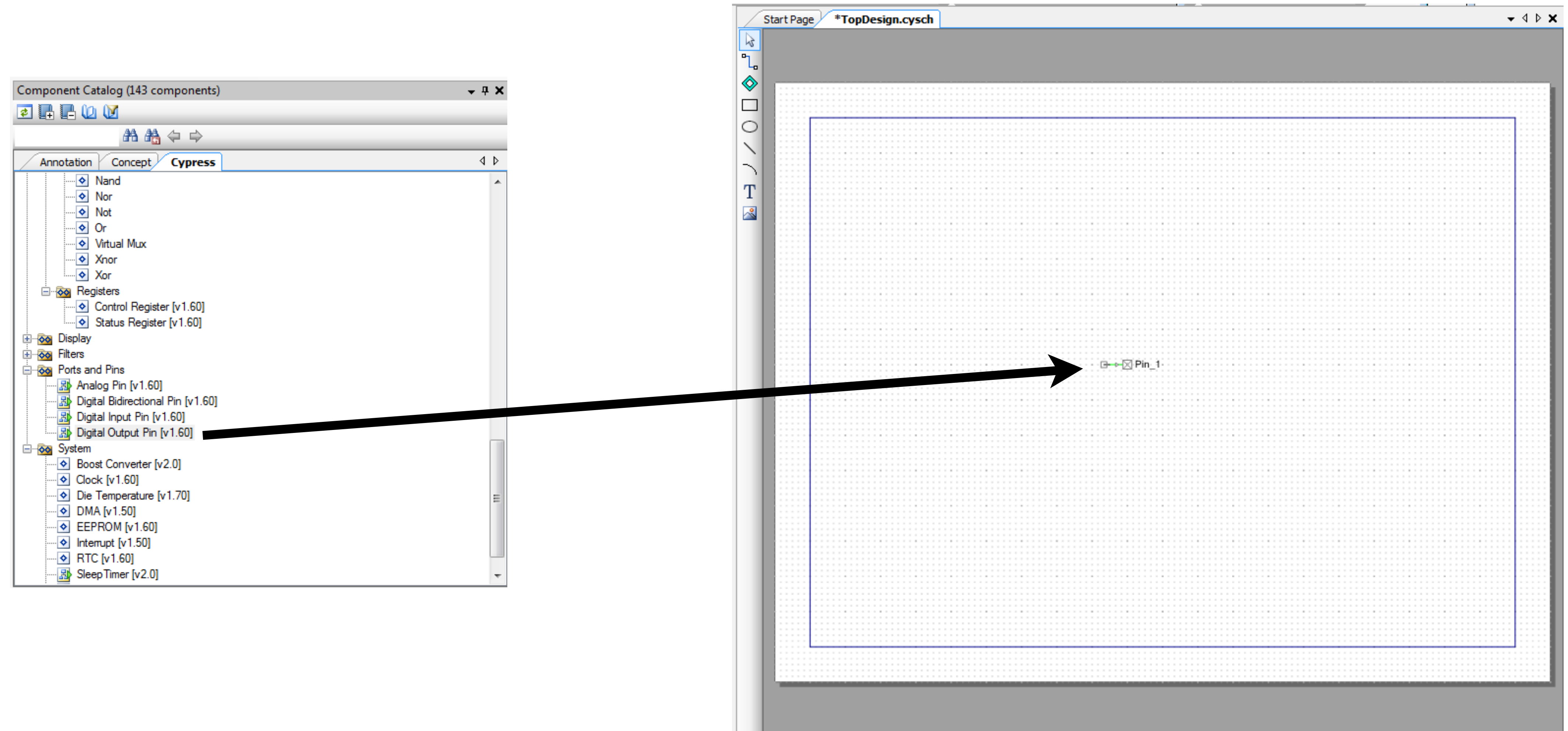
- Download and Install PSoC Creator 2.0
- Watch this Introduction to PSoC Creator Video:
 - <http://www.youtube.com/watch?v=UrpdbefYg-0>
- **Create a simple program**
 - Make one of the LEDs on the PSoC blink at 1 Hz. (500ms on, 500ms off)
 - HINT: There are two LEDs hardwired to PSoC Pins P6[3] and P6[2].
 - HINT: Setting a pin HIGH will turn an LED on.
 - HINT: Setting a pin LOW will turn an LED off.

Getting Started

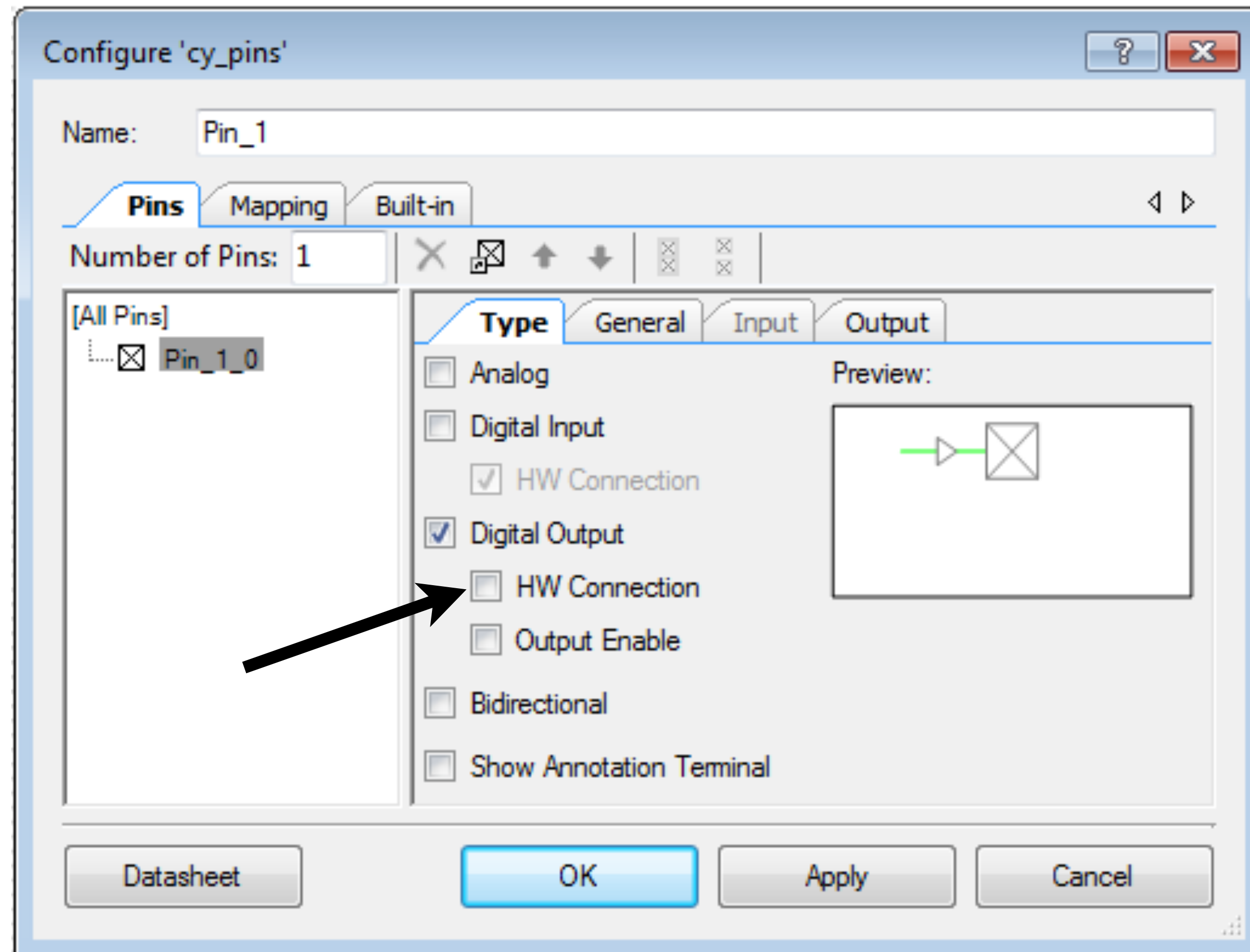


File > New > Project
Select Empty PSoC 5 Design

Drag a Digital Output Pin to Schematic Window

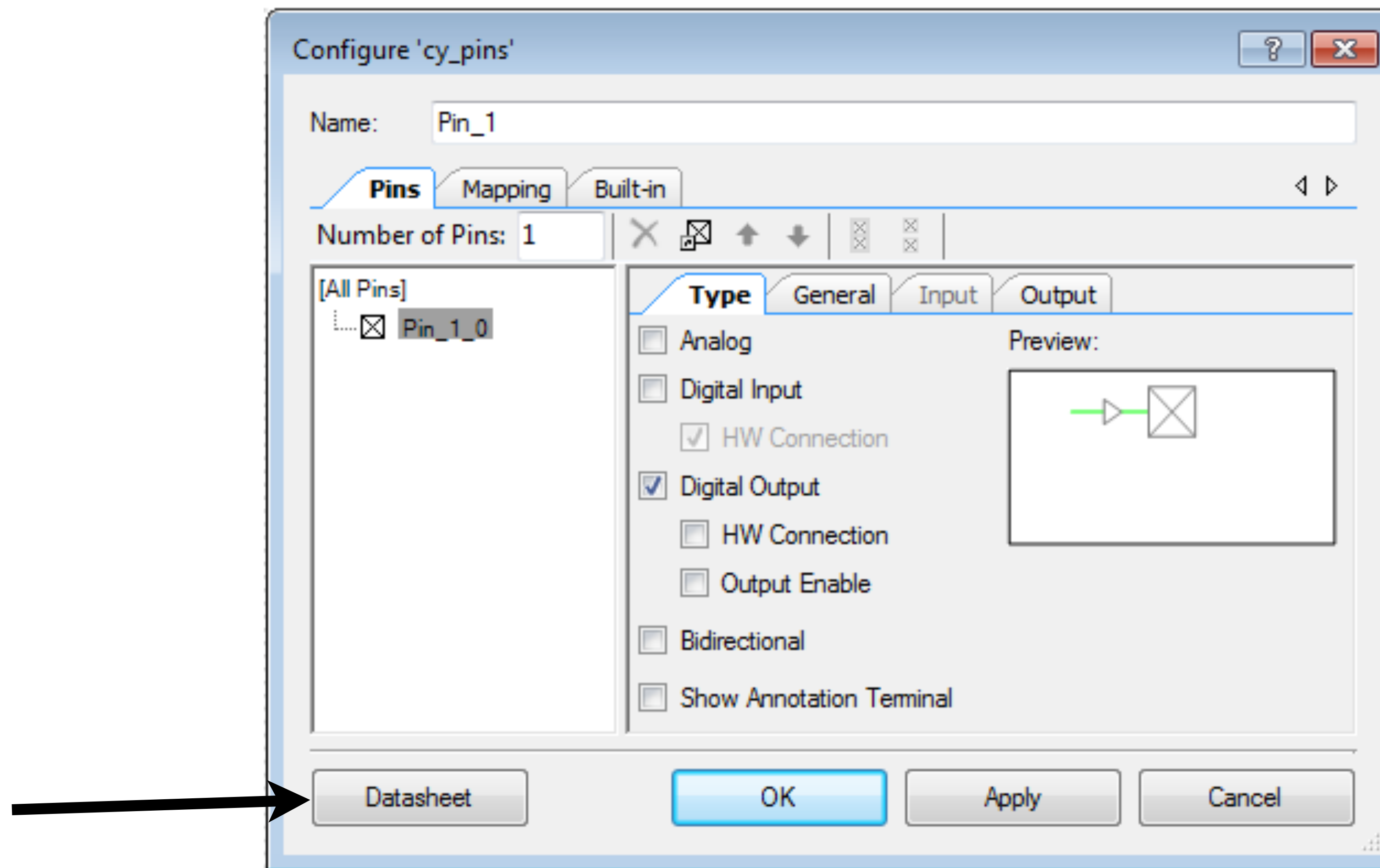


Double-Click Pin to Disable HW Connection (enables Pin to be controlled by firmware)

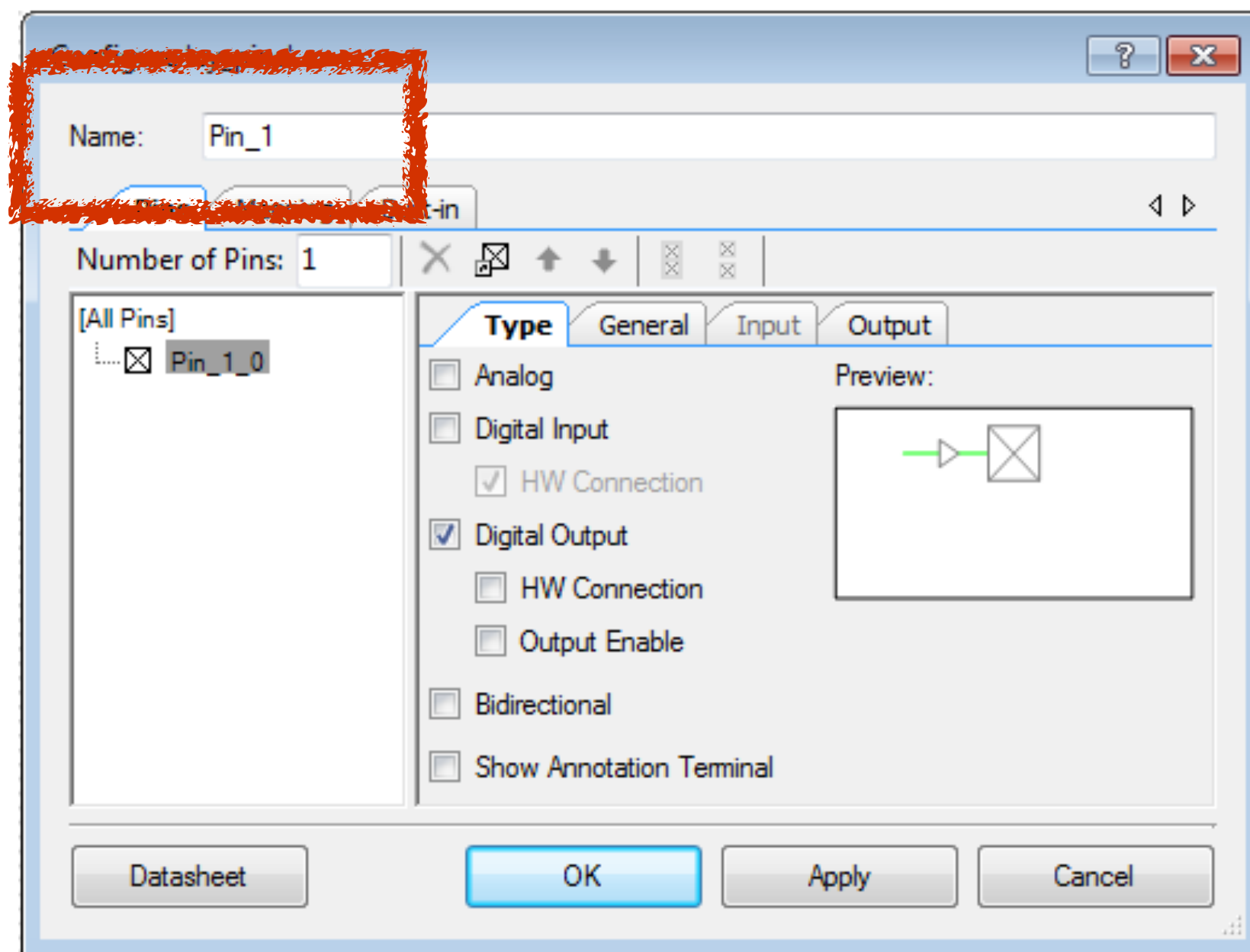


If HW Connection is checked, it will allow you to connect the pin to other components on the block diagram.

Read Datasheet to Learn API.



Functions listed in API use a generic prefix.
To control a specific component, replace the
prefix with the component's name.



+

uint8 Pin_Read(void)

Description:

Reads the associated physical port (pin status register) and masks the required bits according to the width and bit position of the component instance. The pin's status register returns the current logic level present on the physical pin.

Parameters:

None

Return Value:

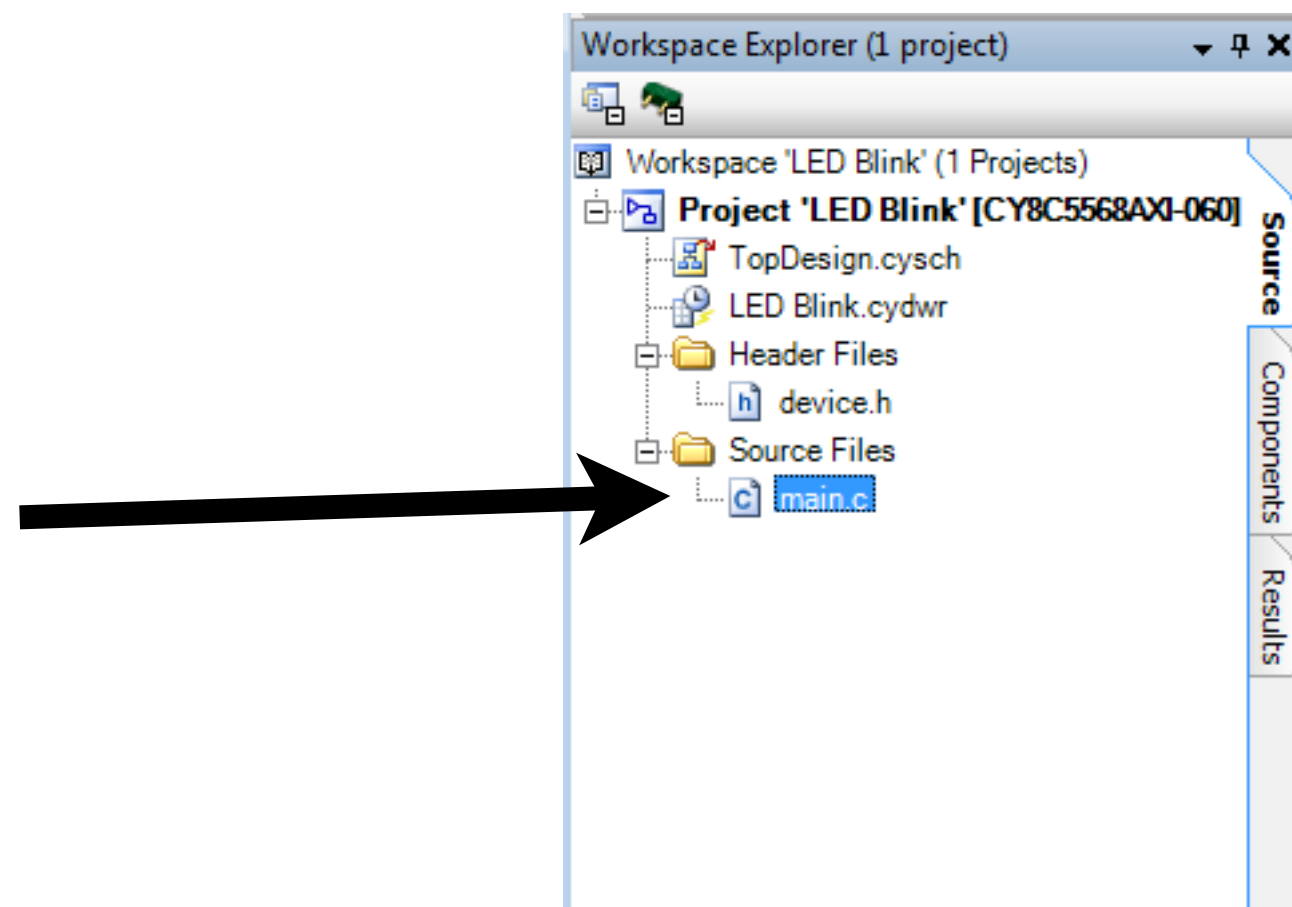
The current value for the pins in the component as a right justified number.

Side Effects:

None

= `Pin_1_Read(); // Reads value from Pin_1`

Write your code in main.c



```
1  /* =====
2  *
3  * Copyright YOUR COMPANY, THE YEAR
4  * All Rights Reserved
5  * UNPUBLISHED, LICENSED SOFTWARE.
6  *
7  * CONFIDENTIAL AND PROPRIETARY INFORMATION
8  * WHICH IS THE PROPERTY OF your company.
9  *
10 * =====
11 */
12 #include <device.h>
13
14 void main()
15 {
16     /* Place your initialization/startup code here (e.g. MyInst_Start()) */
17
18     /* CyGlobalIntEnable; */ /* Uncomment this line to enable global interrupts. */
19     for(;;)
20     {
21         /* Place your application code here. */
22     }
23 }
24
25 /* [] END OF FILE */
26
```

A screenshot of the 'main.c' file in a code editor. The file contains a standard C program template with copyright information, a header inclusion, and a main function. A black arrow points to the 'for(;;)' loop, indicating where the user should place their application code.

One last hint.

```
CyDelay(123); // Delays execution for 123 ms.
```

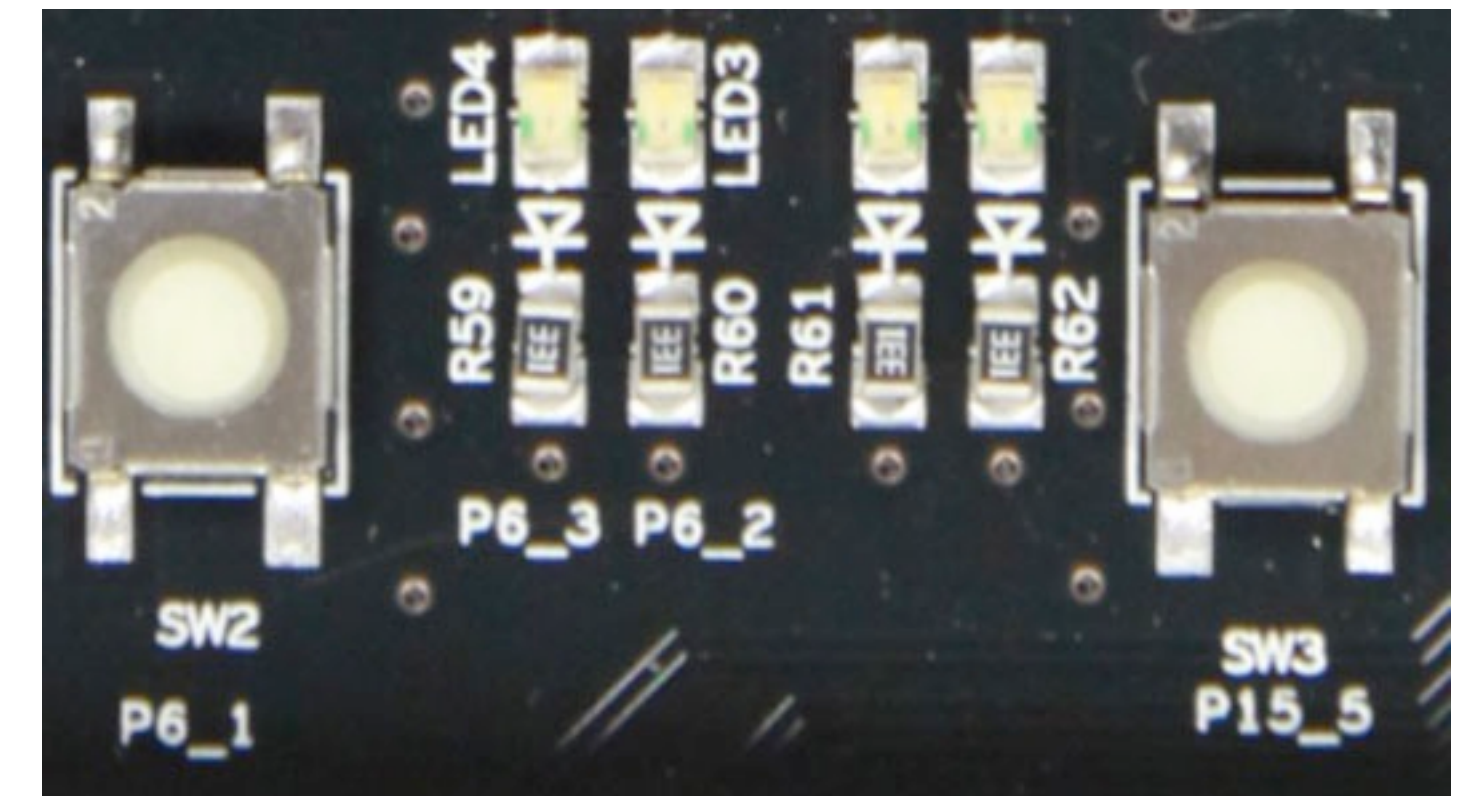
BONUS:

Blink one LED at 1 Hz using the Pin API.
Blink the other LED at 2 Hz using no code.

HINT: Look at the Clock component.

Reminders:

LEDs are located at P6[3] and P6[2]



USB cable must be plugged into the USB programming port to program the board.
It's the one closest to the corner.

