



# Introduction to PSoC Creator for 6.131 and 6.115

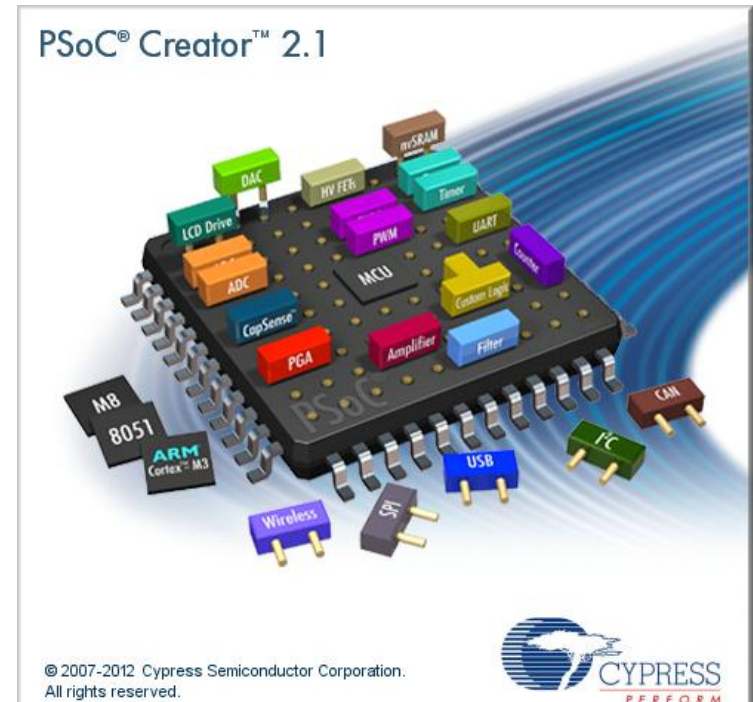
---

Massachusetts Institute of Technology  
Department of Electrical Engineering and  
Computer Science

# Section Objectives

You will be able to:

- Follow the PSoC Creator Design Flow and develop projects
- Find and use the tools available within the software IDE
- Compile, build and program PSoC 3/5 applications
- Debug PSoC 3/5 applications



# PSoC Creator Design Flow

---

## Configure

- Start a new project
- Place components
- Configure components
- Connect components

## Develop

- Build hardware design and generate component APIs
- Write application code utilizing component APIs
- Compile, build and program

## Debug

- Perform in-circuit debug using the MiniProg3 and PSoC Creator

## Reuse

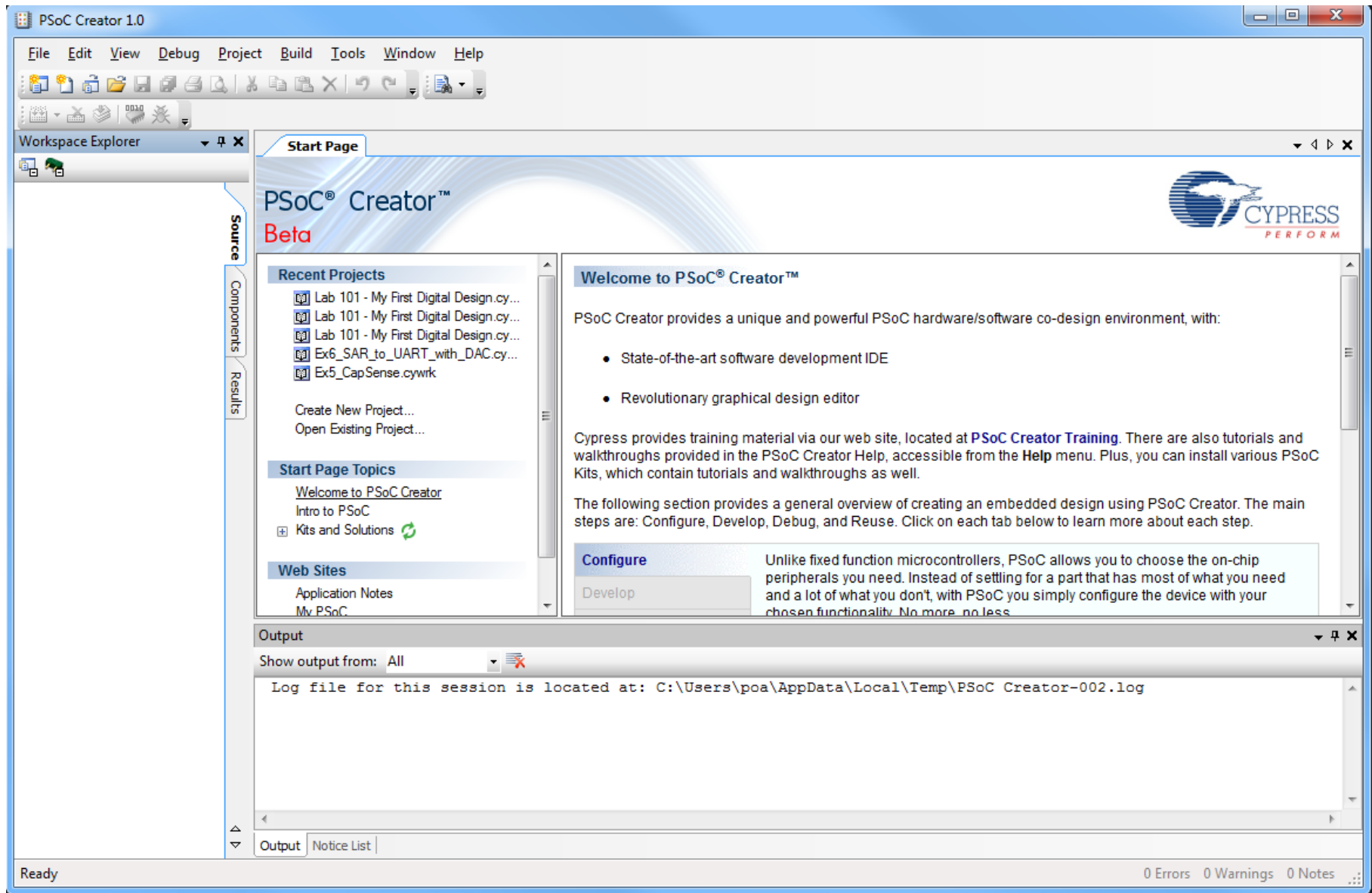
- Capture working hardware/software designs as your own components for future use

# Step 1: Download PSoC Creator

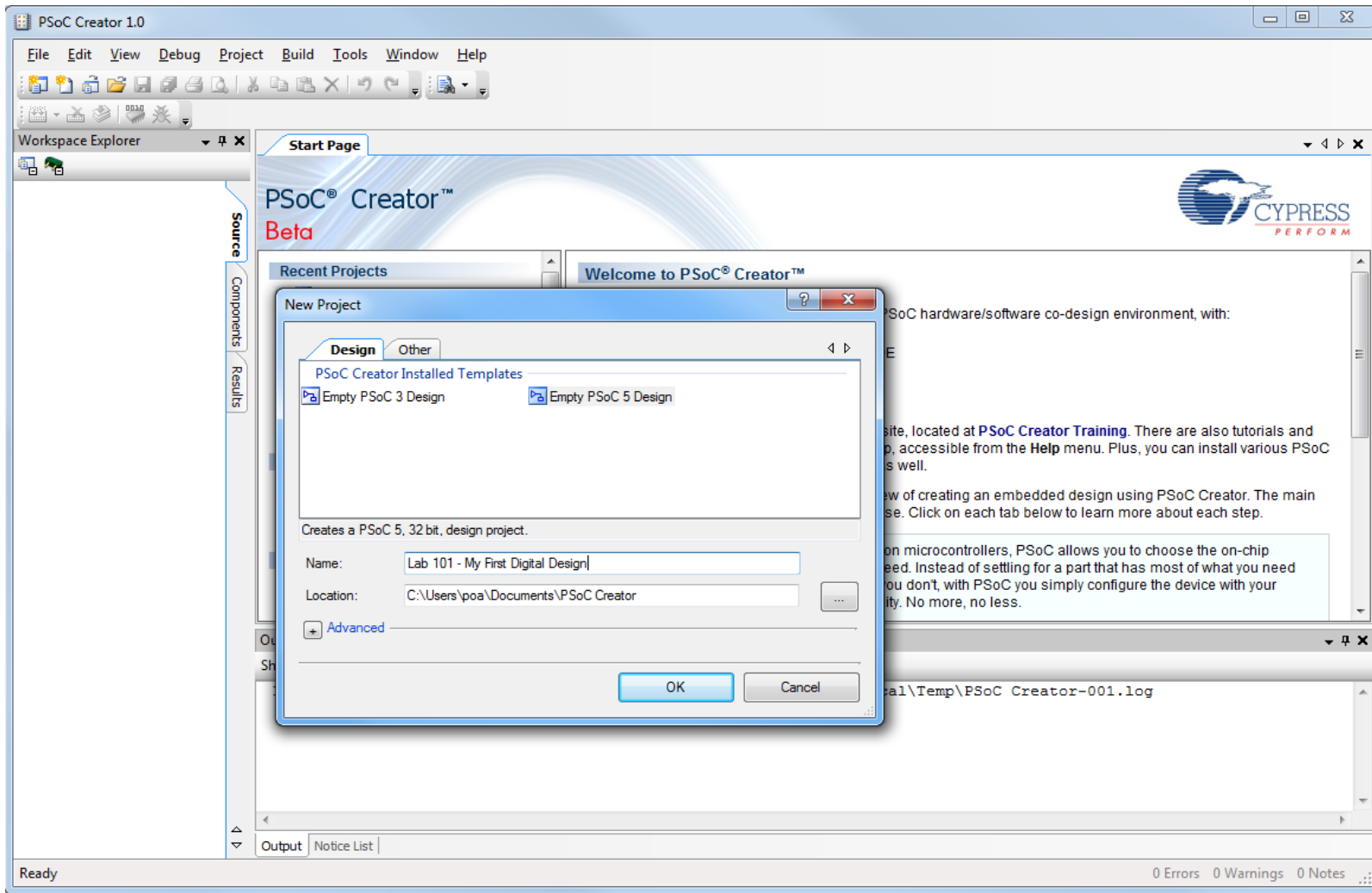
---

1. Go to [www.cypress.com](http://www.cypress.com)
2. Go to Products → Programmable System-on-chip → PSoC Software
3. Click on PSoC Creator. (PSoC Creator is the IDE for PSoC 3 and 5 designs, PSoC Designer is for PSoC 1 designs)
4. Click download. Scroll to the bottom of the page and click on the first link.
5. You will be prompted to login. Create an account to do so.
6. Use the download manager. Click “launch.” Install the typical version of Creator. Run the update manager.

# Step 2: Start PSoC Creator



# Step 3: Create a New Project



File→New→New Project

# PSoC Creator Design Canvas

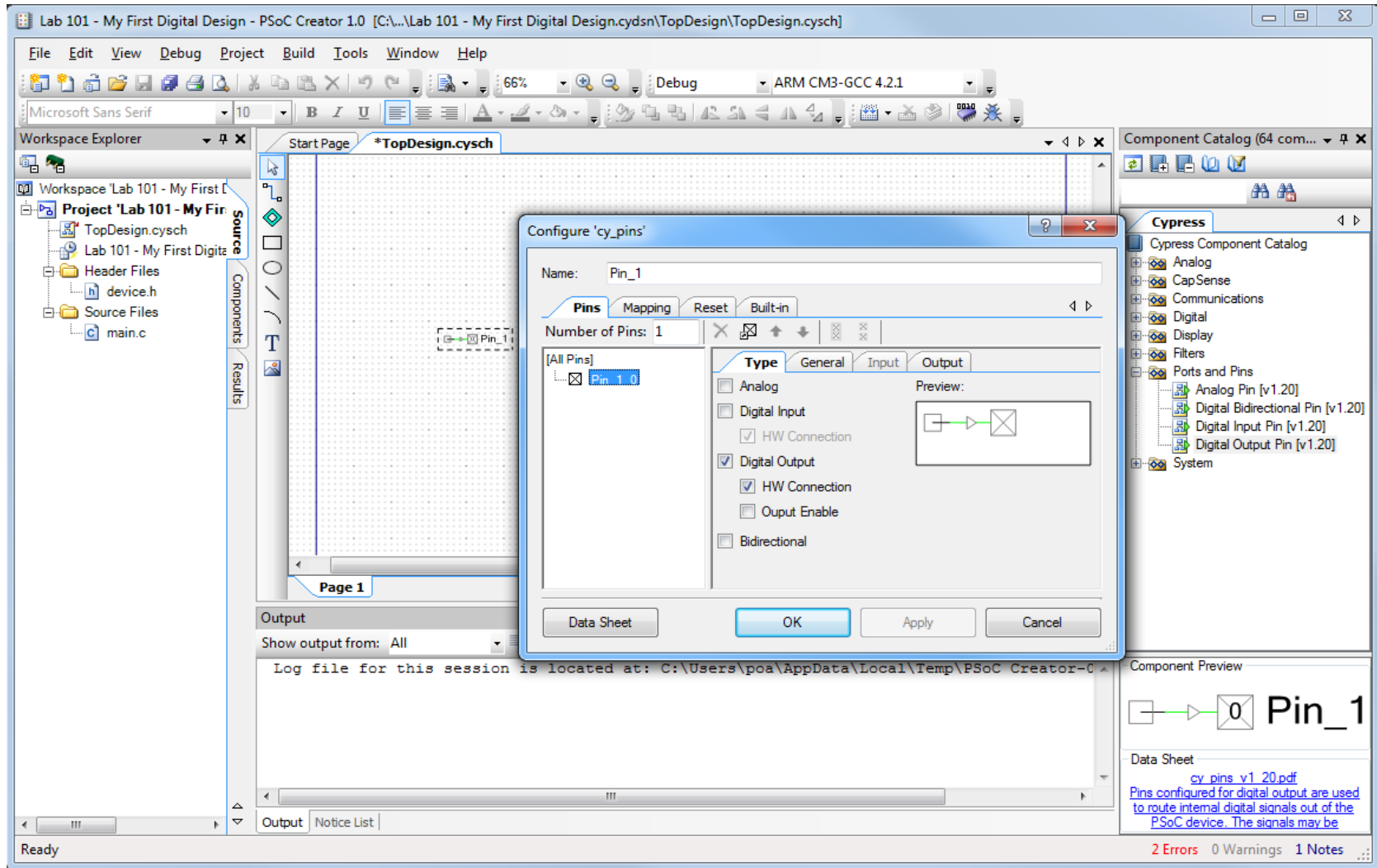
The screenshot displays the PSoC Creator Design Canvas interface. The main window is titled "Design02 - PSoC Creator 1.0" and shows a blank design canvas with a grid pattern. The canvas is currently empty, with a blue border indicating the design area. The interface includes a menu bar (File, Edit, View, Debug, Project, Build, Tools, Window, Help), a toolbar, and a status bar at the bottom.

On the left side, the "Workspace Explorer" shows the project structure for "Project 'Design02' [CY8C5588AXI-060]". The project files include "TopDesign.cysch", "Design02.cydwr", "Header Files", "device.h", "Source Files", and "main.c".

On the right side, the "Component Catalog" is visible, showing a list of components under the "Cypress" category. The components are organized into folders: Analog, CapSense, Communications, Digital, Functions, Logic, Registers, Display, Filters, Ports and Pins, and System. The "Ports and Pins" folder is expanded, showing components like Analog Pin [v1.20], Digital Bidirectional Pin [v1.20], Digital Input Pin [v1.20], and Digital Output Pin [v1.20].

At the bottom, the "Output" window is visible, showing "Show output from: All" and a "Notice List" tab. The status bar at the bottom right indicates "0 Errors 0 Warnings 0 Notes".

# Step 4: Place/Configure Digital Pin



Drag pin from Component Catalog on right

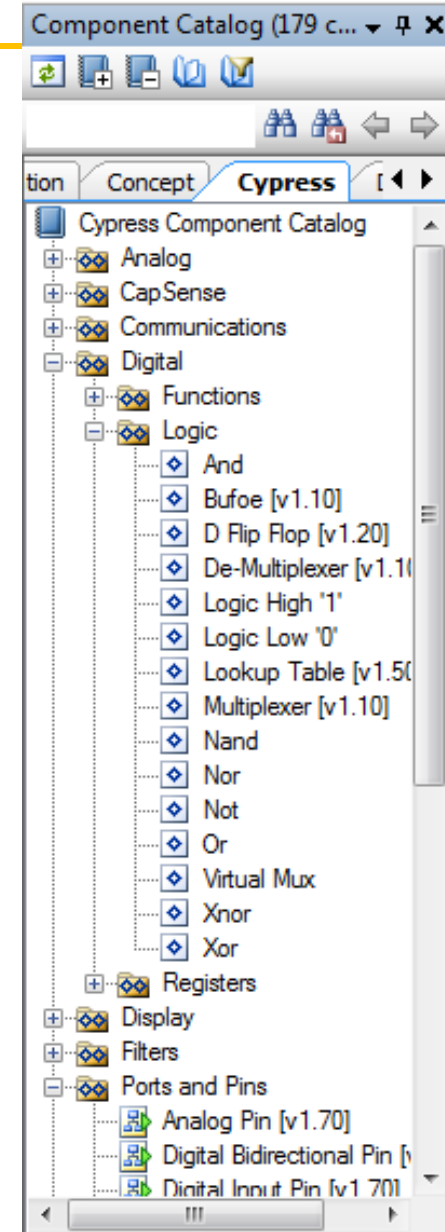


# Component Catalog

## Catalog Folders

- Analog
  - ADC
  - Amplifier
  - DAC
- Digital
  - Registers
  - Functions
  - Logic
- Communication
  - UART
  - Display
  - System

## Datasheet access



# Step 5: Adding Other Components

The screenshot displays the PSoC Creator 1.0 software interface. The main workspace shows a circuit diagram with the following components and connections:

- Clock\_1**: A 24 MHz clock source connected to the **clock** input of the **PWM\_1** component.
- 0**: A logic zero component connected to the **reset** input of the **PWM\_1** component.
- PWM\_1**: A PWM component with outputs **tc**, **pwm1**, and **pwm2**.
- Pin\_1**: A digital output pin component connected to the **interrupt** output of the **PWM\_1** component.

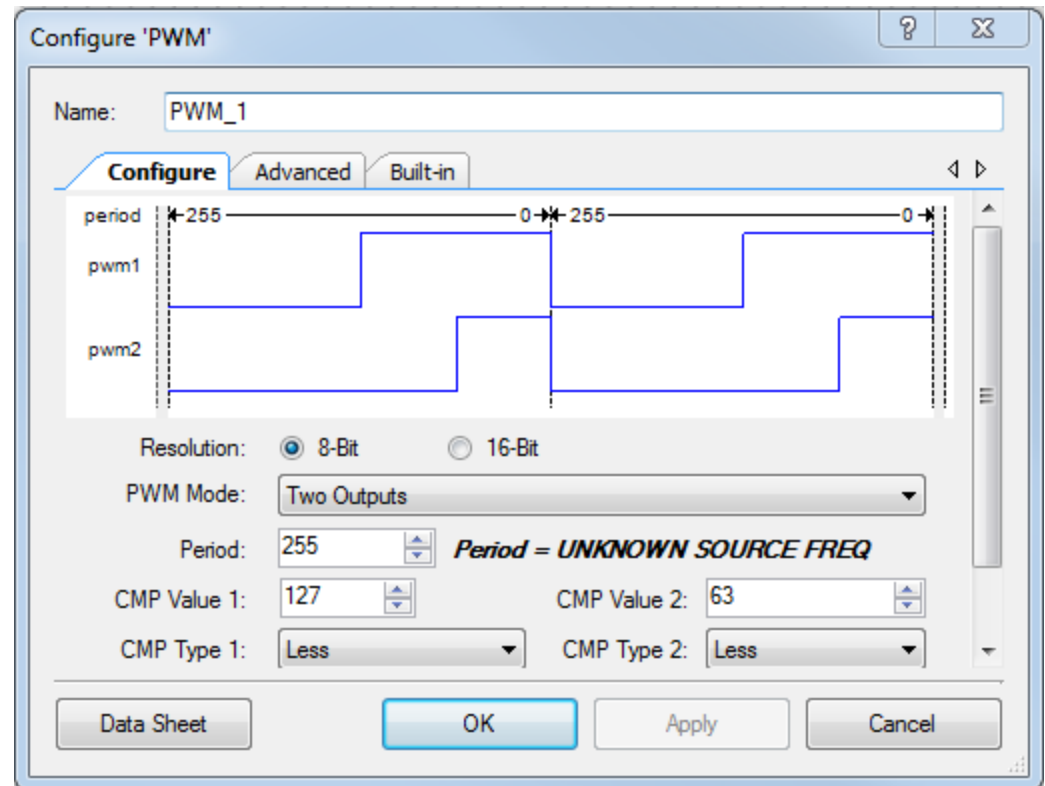
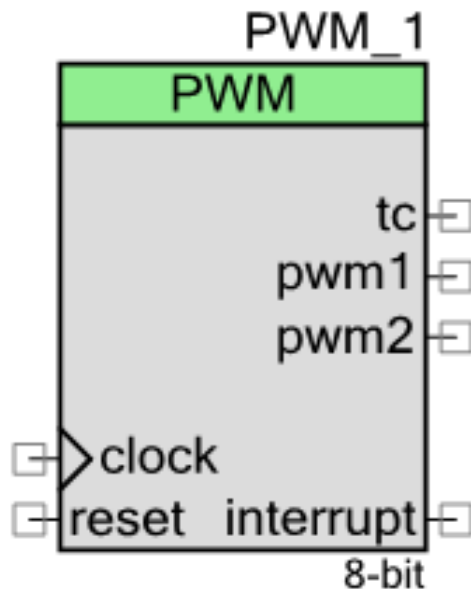
The **Component Catalog** on the right side of the interface shows the **Cypress** library with various components. The **System** category is expanded, showing components like **Boost Converter**, **Clock**, **Die Temperature**, **DMA**, and **EEPROM**. The **Component Preview** window at the bottom right shows a detailed view of the **Pin\_1** component, including a data sheet link: [cy\\_pins\\_v1\\_20.pdf](#). The data sheet link text reads: "Pins configured for digital output are used to route internal digital signals out of the PSoC device. The signals may be".

At the bottom of the interface, the status bar indicates "6 Errors 0 Warnings 1 Notes".

Drag needed components from Component Catalog

# Step 6: Component Configuration

- Double-click on a component to open its component configuration dialog box
- Change configuration
- Click on Datasheet in bottom left corner for component description and APIs



# Component Datasheets

## Contents:

- Features
- General description of component
- When to use component
- Input/Output connections
- Parameters and setup
- Application Programming Interface (API)
- Sample firmware source code
- Functional description
- DC and AC electrical characteristics



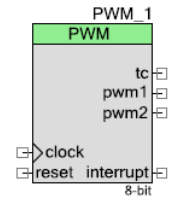
PSoC Creator Component Data Sheet

## Pulse Width Modulator (PWM)

1.10

### Features

- 8 or 16-Bit Resolution
- Multiple Pulse Width Output Modes
- Configurable Trigger
- Configurable Capture
- Configurable Hardware/Software Enable
- Configurable Dead-Band
- Multiple Configurable Kill Modes
- Customized Configuration tool



### General Description

The PWM component provides compare outputs to generate single or continuous timing and control signals in hardware. The PWM is designed to provide an easy method of generating complex real time events accurately with minimal CPU intervention. The PWM features may be combined with other analog and digital components to create custom peripherals.

The PWM generates up to 2 left or right aligned PWM outputs or 1 center aligned or dual edged PWM output. The PWM outputs are double buffered to avoid glitches due to duty cycle changes while running. Left aligned PWMs are used for most general purpose PWM uses. Right aligned PWMs are typically only used in special cases which require alignment opposite of left aligned PWMs. Center aligned PWMs are most often used in AC motor control to maintain phase alignment. Dual edge PWMs are optimized for power conversion where phase alignment must be adjusted.

The optional deadband provides complementary outputs with adjustable dead time where both outputs are low between each transition. The complementary outputs and dead time are most often used to drive power devices in half bridge configurations to avoid shoot through currents and resulting damage. A kill input is also available that immediately disables the deadband outputs when enabled. Three kill modes are available to support multiple use scenarios.

Two hardware dither modes are provided to increase PWM flexibility. The first dither mode increases effective resolution by 2-bits when resources or clock frequency preclude a standard implementation in the PWM counter. The second dither mode uses a digital input to select one of the two PWM outputs on a cycle by cycle basis typically used to provide fast transient response in power converts.

PRELIMINARY

# Step 7: Connecting Components

The screenshot displays the PSoC Creator 1.0 interface. The main workspace shows a circuit diagram with a component labeled "PWM\_1" (PWM) connected to a "Clock\_1" (24 MHz) and a "Pin\_1". A red arrow points to the wire tool in the components palette on the left. The Component Catalog on the right shows the "Cypress" component library, and the Component Preview at the bottom right shows the selected "Pin\_1" component.

Wire tool

Output

Show output from: All

Output Notice List

0 Errors 0 Warnings 0 Notes

Use wire tool (or shortcut key “w”) to connect components

# Design-Wide Resource Manager (.cydwr)

## Clocks

## Interrupts

- Set priority and vector

## DMA

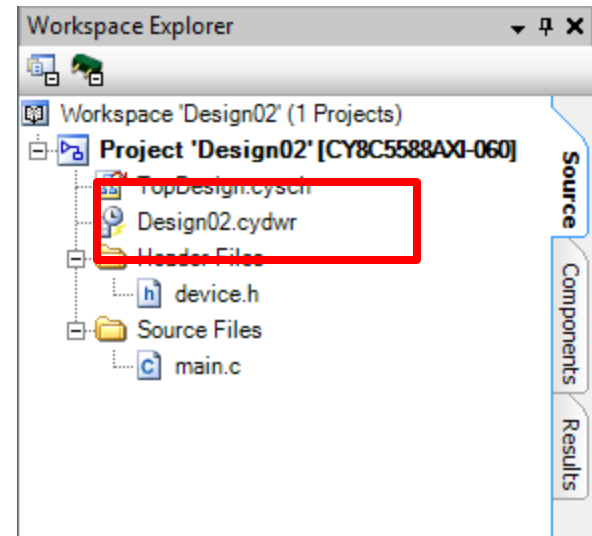
- Manage DMA channels

## System

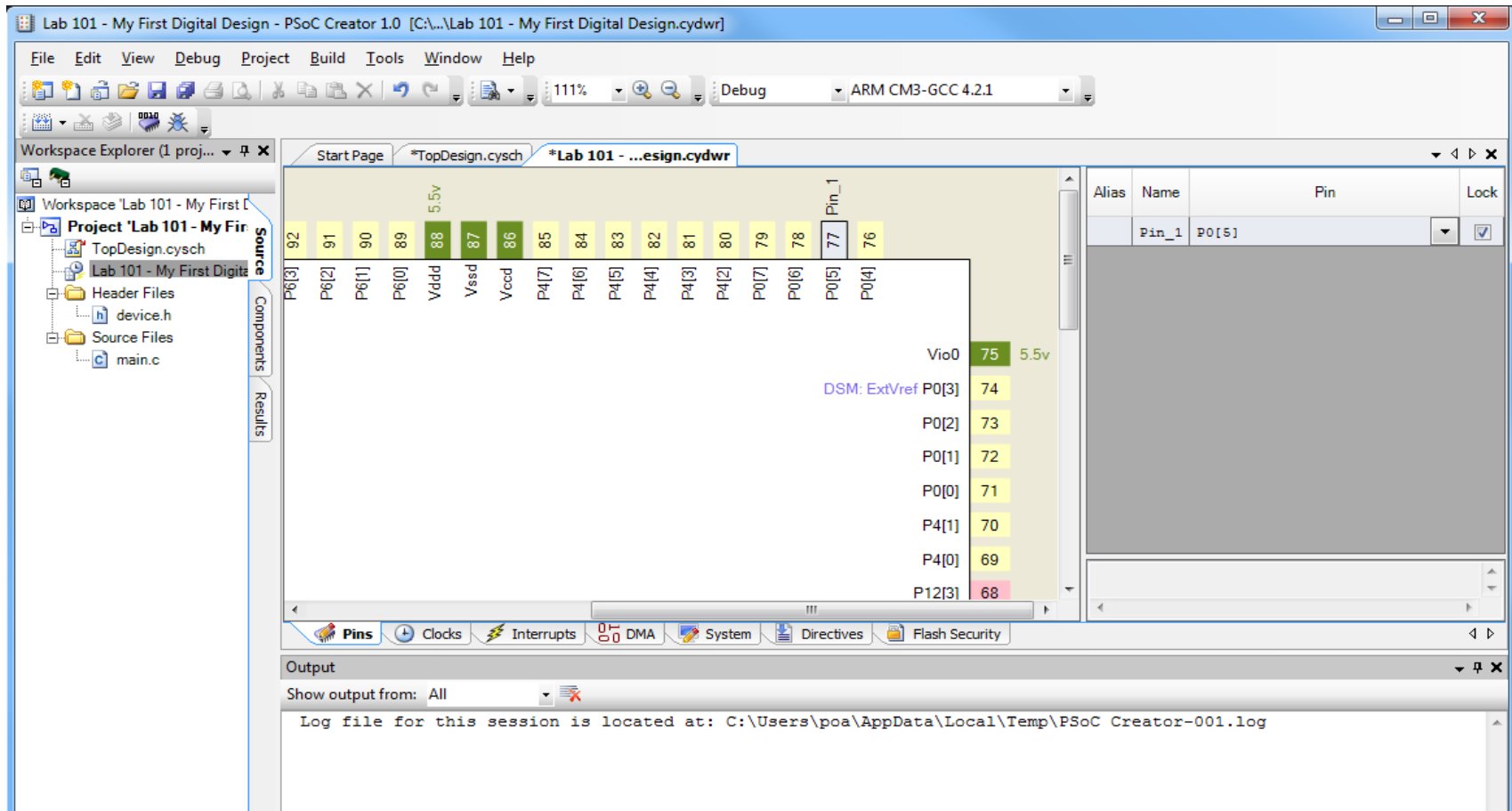
- Debug, boot parameters, sleep mode API generation, etc.

## Pins

- Map I/O to physical pins and ports
- Over-ride default selections



# Step 8: Configure PSoC I/O



The screenshot shows the PSoC Creator 1.0 interface. The main workspace displays a grid of pins with their physical pin numbers. The pins are arranged in a grid with the following labels and values:

Label	Value
P6[3]	92
P6[2]	91
P6[1]	90
P6[0]	89
Vddd	88
Vssd	87
Vccd	86
P4[7]	85
P4[6]	84
P4[5]	83
P4[4]	82
P4[3]	81
P4[2]	80
P0[7]	79
P0[6]	78
P0[5]	77
P0[4]	76

Additional pins shown in the bottom right of the grid:

Label	Value
Vio0	75
DSM: ExtVref	P0[3] 74
P0[2]	73
P0[1]	72
P0[0]	71
P4[1]	70
P4[0]	69
P12[3]	68

The table on the right side of the window shows the mapping from schematic pin names to physical pin numbers:

Alias	Name	Pin	Lock
	Pin_1	P0[5]	<input checked="" type="checkbox"/>

The Output window at the bottom shows the log file path: C:\Users\poa\AppData\Local\Temp\PSoC Creator-001.log

In .cydwr file, route schematic pins to physical pins on the PSoC chip (on right: name to pin mapping)

# Interrupts

- Priority may be changed: defaults to 7 (lowest priority)
- Edit ISR code in interrupt Creator-generated .c file

The screenshot shows the PSoC Creator IDE interface. The main window displays the interrupt configuration table for the ADC\_DelSig\_1\_IRQ. The table has three columns: Instance Name, Priority, and Vector. The priority is set to 'Default <7>' and the vector is 29. A dropdown menu is open over the priority field, showing a list of priority values from 0 to 7. The workspace explorer on the left shows the project structure, with a red arrow pointing to the ADC\_DelSig\_1.c file under the Generated\_Source directory. The output window at the bottom shows 'Build Succeeded'.

Instance Name	Priority	Vector
ADC_DelSig_1_IRQ	Default <7>	29

Creator-generated ISR



# System

- System settings
- Debug settings
- Voltage Configuration

The screenshot displays the PSoc Creator 1.0 software interface. The main window is titled "Design02 - PSoc Creator 1.0" and shows the "System" configuration window. The "System" window is divided into several sections: "Configuration", "Debugging", and "Voltage Configuration".

Option	Type	Value
Configuration		
Mode	ENUM	Compressed
Store Configuration Data in ECC Memory	BOOL	<input type="checkbox"/>
Instruction Cache Enabled	BOOL	<input checked="" type="checkbox"/>
Data Cache Enabled	BOOL	<input checked="" type="checkbox"/>
Enable Error Correcting Code (ECC)	BOOL	<input type="checkbox"/>
Enable Fast IMO	BOOL	<input type="checkbox"/>
Debugging		
Enable	BOOL	<input checked="" type="checkbox"/>
Require XRES Pin	BOOL	<input checked="" type="checkbox"/>
Use Optional XRES	BOOL	<input type="checkbox"/>
Debug Port Select (DPS)	ENUM	SWD (serial wire...)
Enable Trace	BOOL	<input type="checkbox"/>
Voltage Configuration		
Vddd	FLOAT	5.5
Vdda	FLOAT	5.5
Vio0	FLOAT	5.5
Vio1	FLOAT	5.5
Vio2	FLOAT	5.5
Vio3	FLOAT	5.5

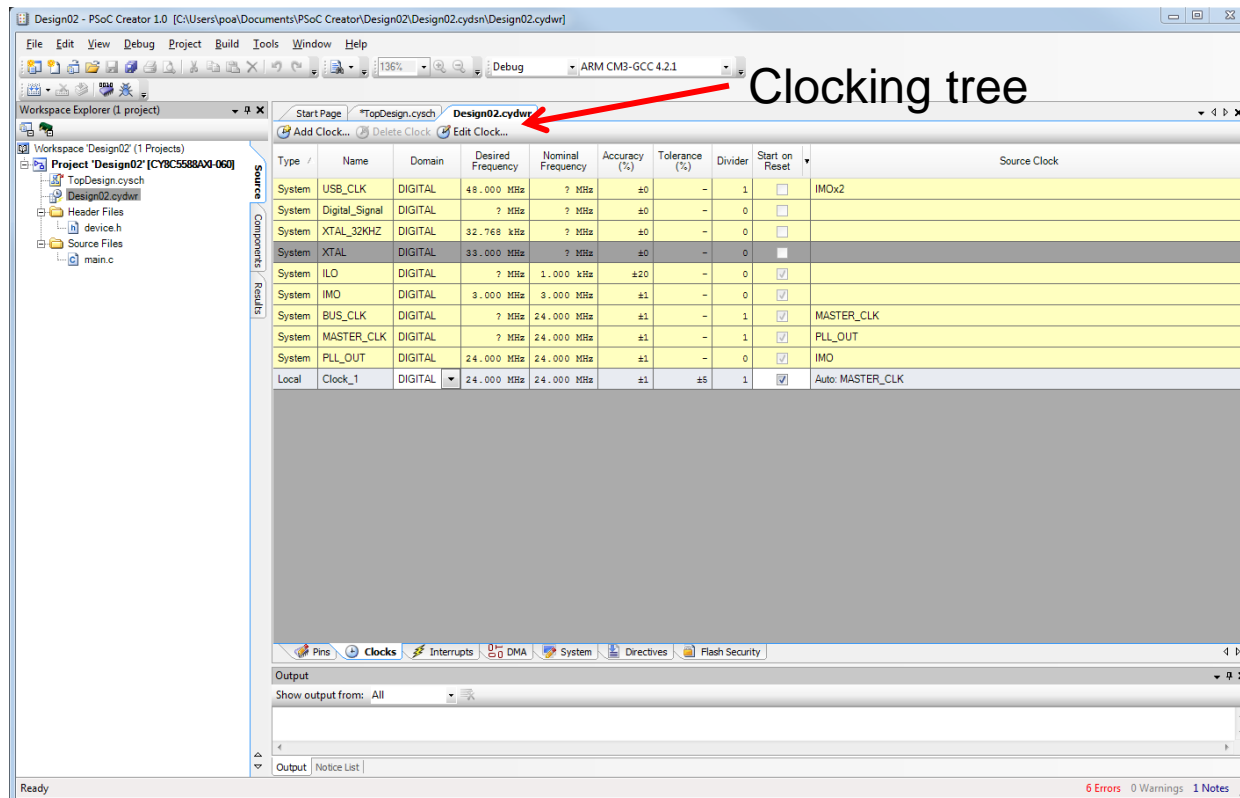
The "System" window also includes a "Voltage Configuration" section with the following settings:

Option	Type	Value
Vddd	FLOAT	5.5
Vdda	FLOAT	5.5
Vio0	FLOAT	5.5
Vio1	FLOAT	5.5
Vio2	FLOAT	5.5
Vio3	FLOAT	5.5

The interface also shows a "Workspace Explorer" on the left, a "Source" view, and an "Output" window at the bottom. The status bar at the bottom indicates "Ready" and "6 Errors 0 Warnings 1 Notes".

# Clock Configurations

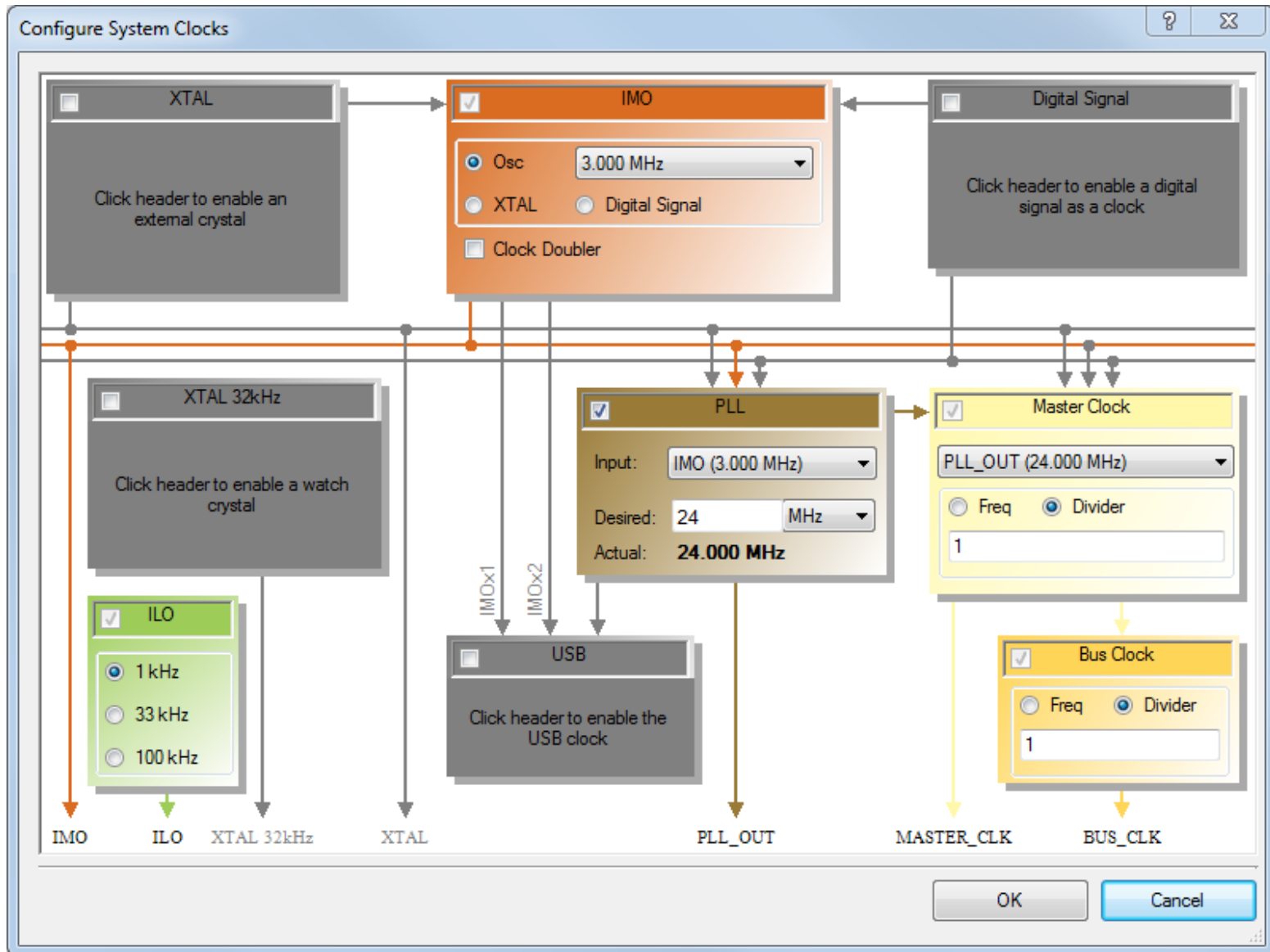
- Clocks are allocated to slots in the clock tree
- 8 digital, 4 analog
- Clocks have APIs
- Reuse existing clocks to preserve resources, if possible



The screenshot displays the PSoC Creator software interface. The main window shows a table of clock configurations. A red arrow points to the 'Add Clock...' button in the top toolbar, which is labeled 'Clocking tree'. The table lists various clock sources and their parameters.

Type	Name	Domain	Desired Frequency	Nominal Frequency	Accuracy (%)	Tolerance (%)	Divider	Start on Reset	Source Clock
System	USB_CLK	DIGITAL	48.000 MHz	? MHz	±0	-	1	<input type="checkbox"/>	IM0x2
System	Digital_Signal	DIGITAL	? MHz	? MHz	±0	-	0	<input type="checkbox"/>	
System	XTAL_32KHZ	DIGITAL	32.768 kHz	? MHz	±0	-	0	<input type="checkbox"/>	
System	XTAL	DIGITAL	33.000 MHz	? MHz	±0	-	0	<input type="checkbox"/>	
System	ILO	DIGITAL	? MHz	1.000 kHz	±20	-	0	<input checked="" type="checkbox"/>	
System	IMO	DIGITAL	3.000 MHz	3.000 MHz	±1	-	0	<input type="checkbox"/>	
System	BUS_CLK	DIGITAL	? MHz	24.000 MHz	±1	-	1	<input checked="" type="checkbox"/>	MASTER_CLK
System	MASTER_CLK	DIGITAL	? MHz	24.000 MHz	±1	-	1	<input checked="" type="checkbox"/>	PLL_OUT
System	PLL_OUT	DIGITAL	24.000 MHz	24.000 MHz	±1	-	0	<input checked="" type="checkbox"/>	IMO
Local	Clock_1	DIGITAL	24.000 MHz	24.000 MHz	±1	±5	1	<input checked="" type="checkbox"/>	Auto: MASTER_CLK

# System Clocking Tree



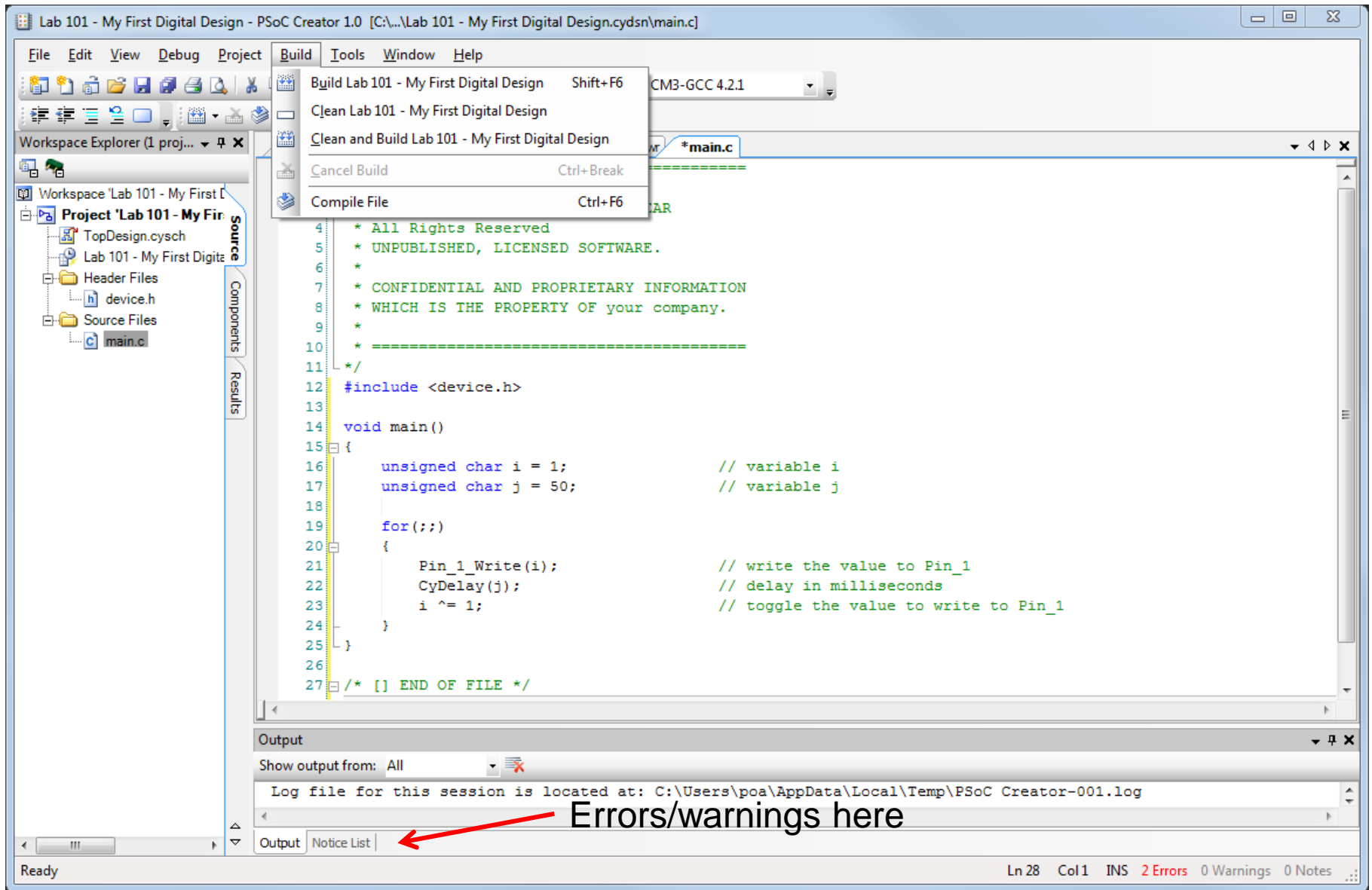
# Step 8: Add main.c Code

The screenshot shows the PSoC Creator 1.0 IDE with the following components:

- Workspace Explorer:** Shows the project structure for 'Lab 101 - My First Digital Design'. The 'Source Files' folder contains 'main.c'.
- Code Editor:** Displays the content of 'main.c' with line numbers 1 through 27. The code is as follows:

```
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     unsigned char i = 1;           // variable i  
17     unsigned char j = 50;         // variable j  
18  
19     for(;;)  
20     {  
21         Pin_1_Write(i);           // write the value to Pin_1  
22         CyDelay(j);               // delay in milliseconds  
23         i ^= 1;                   // toggle the value to write to Pin_1  
24     }  
25 }  
26  
27 /* [] END OF FILE */
```
- Output Window:** Shows the message: 'Log file for this session is located at: C:\Users\poa\AppData\Local\Temp\PSoC Creator-001.log'.
- Status Bar:** Displays 'Ready' on the left and 'Ln 28 Col 1 INS 2 Errors 0 Warnings 0 Notes' on the right.

# Step 9: Build Project



The screenshot displays the PSoC Creator 1.0 interface. The 'Build' menu is open, showing options: 'Build Lab 101 - My First Digital Design' (Shift+F6), 'Clean Lab 101 - My First Digital Design', 'Clean and Build Lab 101 - My First Digital Design', 'Cancel Build' (Ctrl+Break), and 'Compile File' (Ctrl+F6). The compiler is set to 'CM3-GCC 4.2.1'. The workspace explorer on the left shows the project structure: 'Project 'Lab 101 - My First Digital Design' containing 'TopDesign.cysch', 'Lab 101 - My First Digital Design' (with subfolders 'Header Files' containing 'device.h' and 'Source Files' containing 'main.c'). The main editor shows the content of 'main.c':

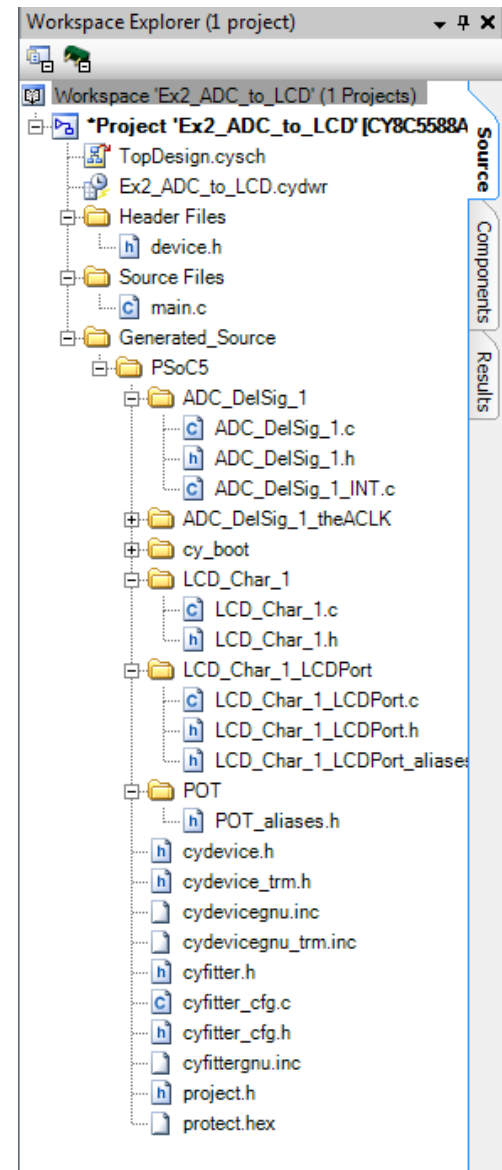
```
4  * All Rights Reserved
5  * UNPUBLISHED, LICENSED SOFTWARE.
6  *
7  * CONFIDENTIAL AND PROPRIETARY INFORMATION
8  * WHICH IS THE PROPERTY OF your company.
9  *
10 * =====
11 */
12 #include <device.h>
13
14 void main()
15 {
16     unsigned char i = 1;           // variable i
17     unsigned char j = 50;         // variable j
18
19     for(;;)
20     {
21         Pin_1_Write(i);           // write the value to Pin_1
22         CyDelay(j);               // delay in milliseconds
23         i ^= 1;                   // toggle the value to write to Pin_1
24     }
25 }
26
27 /* [] END OF FILE */
```

The 'Output' window at the bottom shows the message: 'Log file for this session is located at: C:\Users\poa\AppData\Local\Temp\PSoC Creator-001.log'. A red arrow points to the 'Notice List' tab in the Output window, with the text 'Errors/warnings here' overlaid. The status bar at the bottom right indicates 'Ready' and 'Ln 28 Col 1 INS 2 Errors 0 Warnings 0 Notes'.

# Step 9: Build Process

- API Generation
- Compilation
- Configuration Generation
- Configuration Verification

Generated APIs



# Step 10: Program Device

The screenshot displays the PSoC Creator 1.0 interface. The 'Debug' menu is open, showing options like 'Program' (Ctrl+F5), 'Execute Code' (F5), and 'Attach to Running Target...'. The main editor shows the C code for 'main.c', which includes a loop that writes to Pin\_1 with a delay. The 'Output' window at the bottom shows the build process, with a red box highlighting the memory usage: 'Flash used: 2006 of 262144 bytes (0.8 %)' and 'SRAM used: 152 of 65536 bytes (0.2 %)'.

```
Lab 101 - My First Digital Design - PSoC Creator 1.0 [C:\...\Solution\Lab 101 - My First Digital Design.cysdn\main.c]
```

File Edit View **Debug** Project Build Tools Window Help

Debug ARM CM3-GCC 4.2.1

Workspace Explorer

Workspace 'Lab 101'

Project 'Lab 101 - My First Digital Design'

TopDesign

Lab 101 - My First Digital Design

Header Files

device

Source Files

main.c

Generated\_Source

PSoC5

cy\_boot

cm3gcc.ld

Cm3RealView.scat

Cm3Start.c

CyBootAsmGnu.s

CyDmac.c

CyDmac.h

CyFlash.c

CyFlash.h

CyLib.c

CyLib.h

CyLibNoOS.c

cypins.h

CySp.c

CySp.h

cytypes.h

cyutils.c

Pin\_1

Pin\_1.c

Pin\_1.h

Pin\_1\_aliases.h

```
CONFIDENTIAL AND PROPRIETARY INFORMATION  
WHICH IS THE PROPERTY OF your company.
```

```
=====
```

```
#include <device.h>
```

```
main()
```

```
16 unsigned char i = 1; // variable i
```

```
17 unsigned char j = 50; // variable j
```

```
18
```

```
19 for(;;)
```

```
20 {
```

```
21     Pin_1_Write(i); // write the value to Pin_1
```

```
22     CyDelay(j); // delay in milliseconds
```

```
23     i ^= 1; // toggle the value to write to Pin_1
```

```
24 }
```

```
25 }
```

```
26
```

```
27 /* [] END OF FILE */
```

```
28
```

Output

Show output from: All

```
arm-none-eabi-gcc.exe "-nostartfiles" "-mcpu=cortex-m3" "-mthumb" "-T" ".\cm3gcc.pld" "-g" "-Wl,-Map,C:/U:\n\ncynextool" "-o" "C:/Users/poa/Desktop/101 - Overview and Design Flow/Solution/Lab 101 - My First Digital D\nFlash used: 2006 of 262144 bytes (0.8 %).  
SRAM used: 152 of 65536 bytes (0.2 %).
```

Build Succeeded

Output Notice List

Ready Ln1 Col1 INS 0 Errors 0 Warnings 0 Notes

# Step 11: Debug

The screenshot shows the PSoc Creator 1.0 debugger interface. The main window displays the source code for `main.c` with the following content:

```
8  * WHICH IS THE PROPERTY OF your company.
9  *
10 * =====
11 */
12 #include <device.h>
13
14 void main()
15 {
16     unsigned char i = 1; // variable i
17     unsigned char j = 50; // variable j
18
19     for(;;)
20     {
21         Pin_1_Write(i); // write the value to Pin_1
22         CyDelay(j); // delay in milliseconds
23         i ^= 1; // toggle the value to write
24     }
25 }
26
27 /* [] END OF FILE */
28
```

The debugger is halted at line 16. The Registers window on the left shows the state of the processor registers, and the Call Stack window shows the current function `main()` at `./main.c`. A context menu is open over the code, listing various actions such as `Insert Breakpoint`, `Break Here Once`, `Add WatchPoint`, `Add Watch`, `Run To Cursor`, `Set Next Instruction`, `Undo` (Ctrl+Z), `Redo` (Ctrl+Y), `Cut` (Ctrl+X), `Copy` (Ctrl+C), `Paste` (Ctrl+V), `Delete` (Del), and `Select All` (Ctrl+A).

A red arrow points to the `Memory` window at the bottom, which displays the memory contents at address `0x00000000`:

Address	00	80	00	20	fd	00	00	00	...
0x00000000	00	80	00	20	fd	00	00	00	...
0x00000008	f5	00	00	00	f5	00	00	00	...
0x00000010	f5	00	00	00	f5	00	00	00	...
0x00000018	f5	00	00	00	00	00	00	00	...
0x00000020	00	00	00	00	00	00	00	00	...
0x00000028	00	00	00	00	f5	00	00	00	...

The Output window at the bottom right shows the following text:

```
Erasing...
Programming of Flash Start...
Protecting...
Verify Checksum...
Device 'PSOC5 CY8C5588AX*-060' was successfully p
Continuing target program
The target program has stopped at: file: main.c 1
```

The status bar at the bottom indicates the debugger is `Halted` at `Ln 16 Col 1` with `INS` instruction and `0 Errors 0 Warnings 0 Notes`.