Getting started – blinking LED

# Overview

This lab introduces you how to use ARM Keil MDK for programming and debugging an application on the PSoC 4 BLE chip. It also introduces you to the PSoC 4 BLE Bluetooth Low Energy Pioneer Kit.

KitProg

P3 [7] port

LEDs

ARM Cortex-m0

PSoC4 BLE chip

BLE Pioneer Kit

Keil MDK Development tool

USB

USB

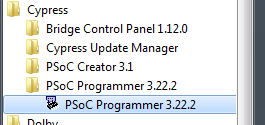
Figure 1 System Diagram

# Targets

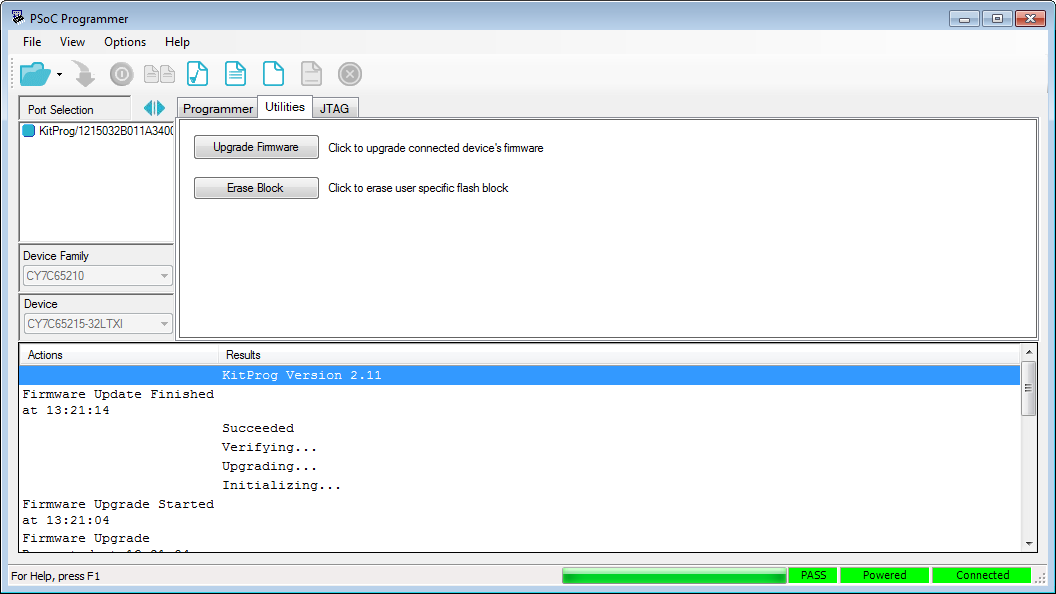
* Learn how to use ARM Keil MDK development tool to write a program
* Design a blinking LED program, download and debug on the PSoC4 BLE chip

# Procedure

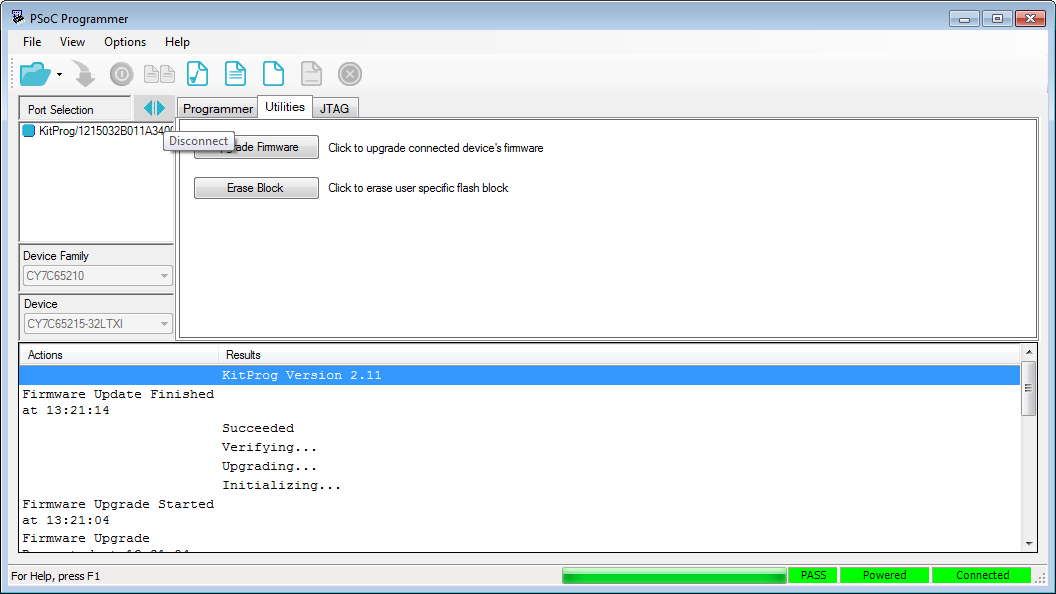
1. Getting started
   1. Install ARM Keil MDK software. You can download it at <http://www2.keil.com/mdk5/>, or open the installer file provided in this package.
   2. Install Cypress PSoC creator at <http://www.cypress.com/psoccreator/>, or use the installer file. The PSoC creator will provide us with the necessary tools to install the driver and update the hardware firmware
   3. Update hardware firmware
      1. Connect the board using a mini USB cable
      2. Open PSoC programmer



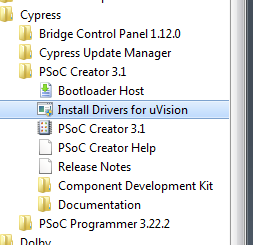
* + 1. Click “update firmware” under “utilities”. If it is successful, it will show “pass” on the bottom right



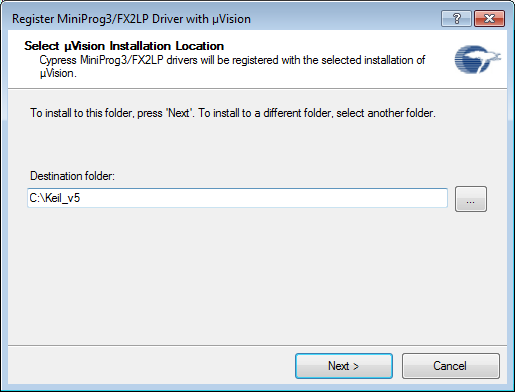
* + 1. Click “disconnect” to disconnect the programmer, then close programer



* 1. Install PSoC driver for ARM Keil MDK uVision
     1. Under “Cypress” -> “PSoC Creator”, Select “Install drivers for uVision”



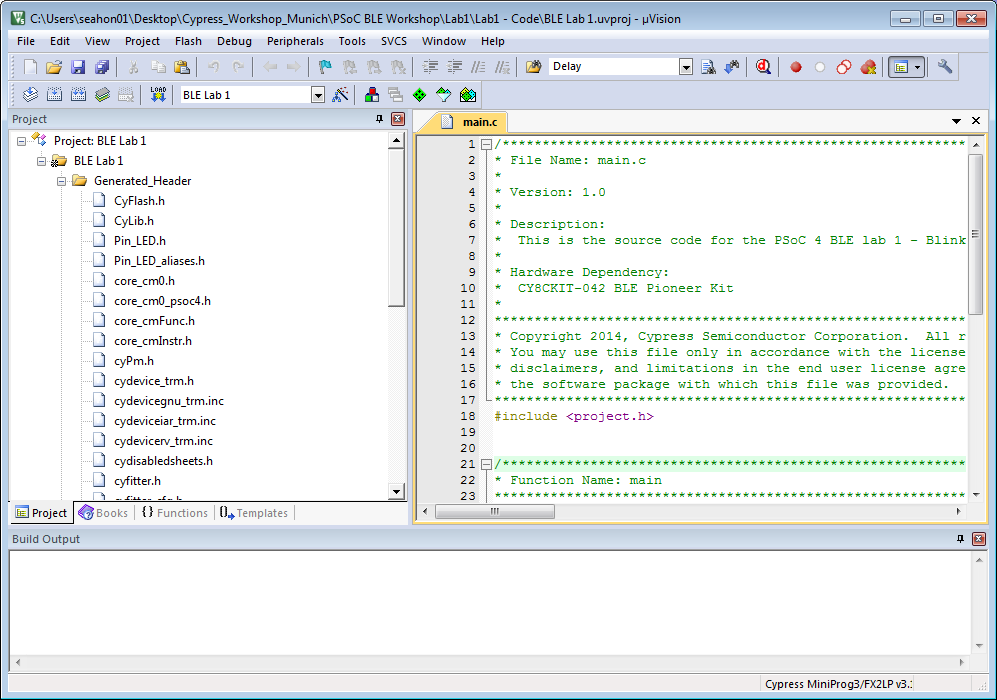
* + 1. Use the default directory, click “next”. This will finish the installation.



1. blinking led program
   1. Open “Lab Code - LED”. In this folder we have already created a Keil MDK project for you.
   2. Double click to open project file “BLE Lab 1.uvproj”

*Note: the program may pop up a message indicating no device library found, this is because the device support is installed through 3rd party and it hasn’t been recognized by the software, we can ignore the message and click OK*

* 1. This opens the standard IDE of Keil MDK. Below are some highlights of the main functions.



* 1. Open the main file, copy and paste the below code into the for loop, then save it.

for(;;)

{

Pin\_LED\_Write(~Pin\_LED\_Read()); // Toggle Pin state

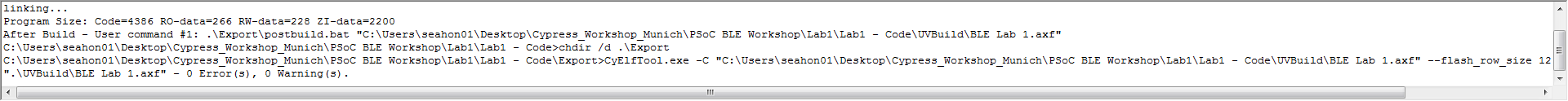
CyDelay(1000); // System delay of 1 second

}

* 1. Build the program
     1. Click the build button



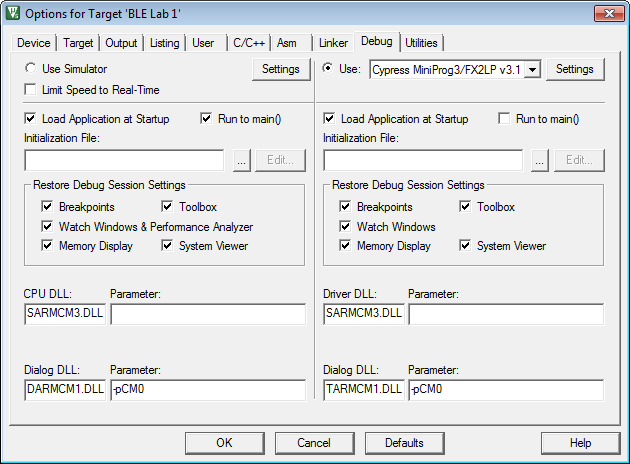
* + 1. Make sure there is no error found



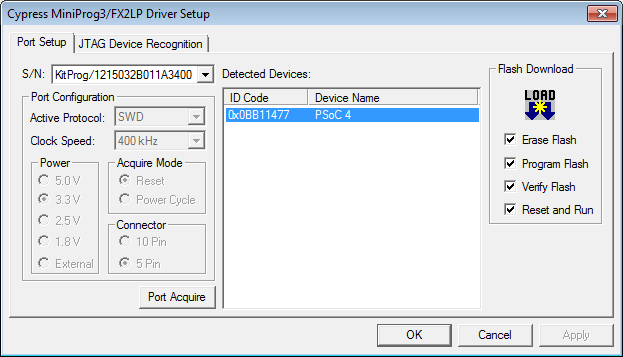
* 1. Download the program. To do that we need to configure the download and debug settings. Note that we only need to do it once.
     1. Click the “option” button at the top bar



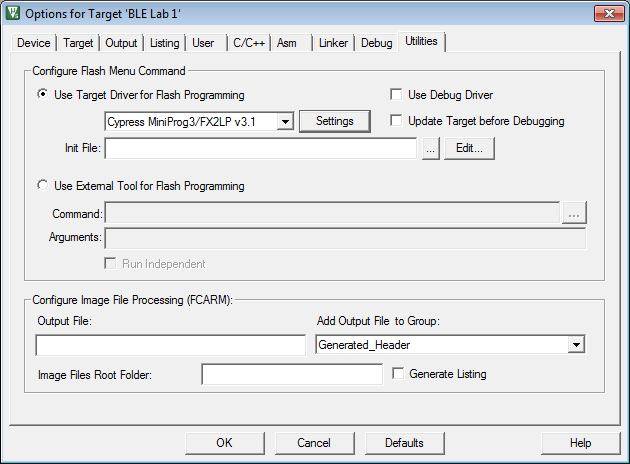
* + 1. Go to tab “debug”, make sure “Cypress MiniProg3” debugger is selected, then click “settings”



* + 1. Click “Port acquire”, then select “PSoC 4”, then click “OK”. This will finish the configuration



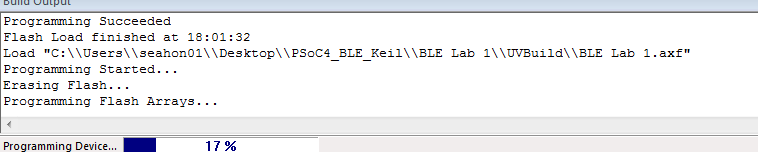
* + 1. Go to Utility tab, and repeat the above 2 steps. Make sure “*Cypress MiniProg3*” is selected.



* + 1. Click the download button



You should see the download starts and the progress bar is showing at the bottom



* + 1. Once the download is finished, you should see the LED starts blinking
  1. Debug the program
* Using the Debugger
* Begin or end a debugger session using one of these methods:
* Toolbar button: 
* Menu: Debug->Start/Stop Debug Session
* Accelerator keys: ctrl+F5
* Control the target program execution with the following toolbar buttons (shown from left to right above): 
* Reset MCU
* Run program execution (F5)
* Stop program execution
* Step one line in program, entering a subroutine (F11)
* Step one line in program, executing and returning from a subroutine (F10)
* Step out of current function (ctrl+F11)
* Run to cursor (ctrl+F10)
* Right-clicking on a line of code will bring up a context menu with various options, including:
* Setting and clearing breakpoints
* Adding a variable to a watch window
* Navigating to definitions or uses of symbols (functions and variables)
* The View menu can be used to open different windows to help in debugging, including:
* Disassembly
* Symbols
* CPU registers
* Call stack
* Variable watch windows
* Memory windows
* System viewer (MCU and peripheral control registers)

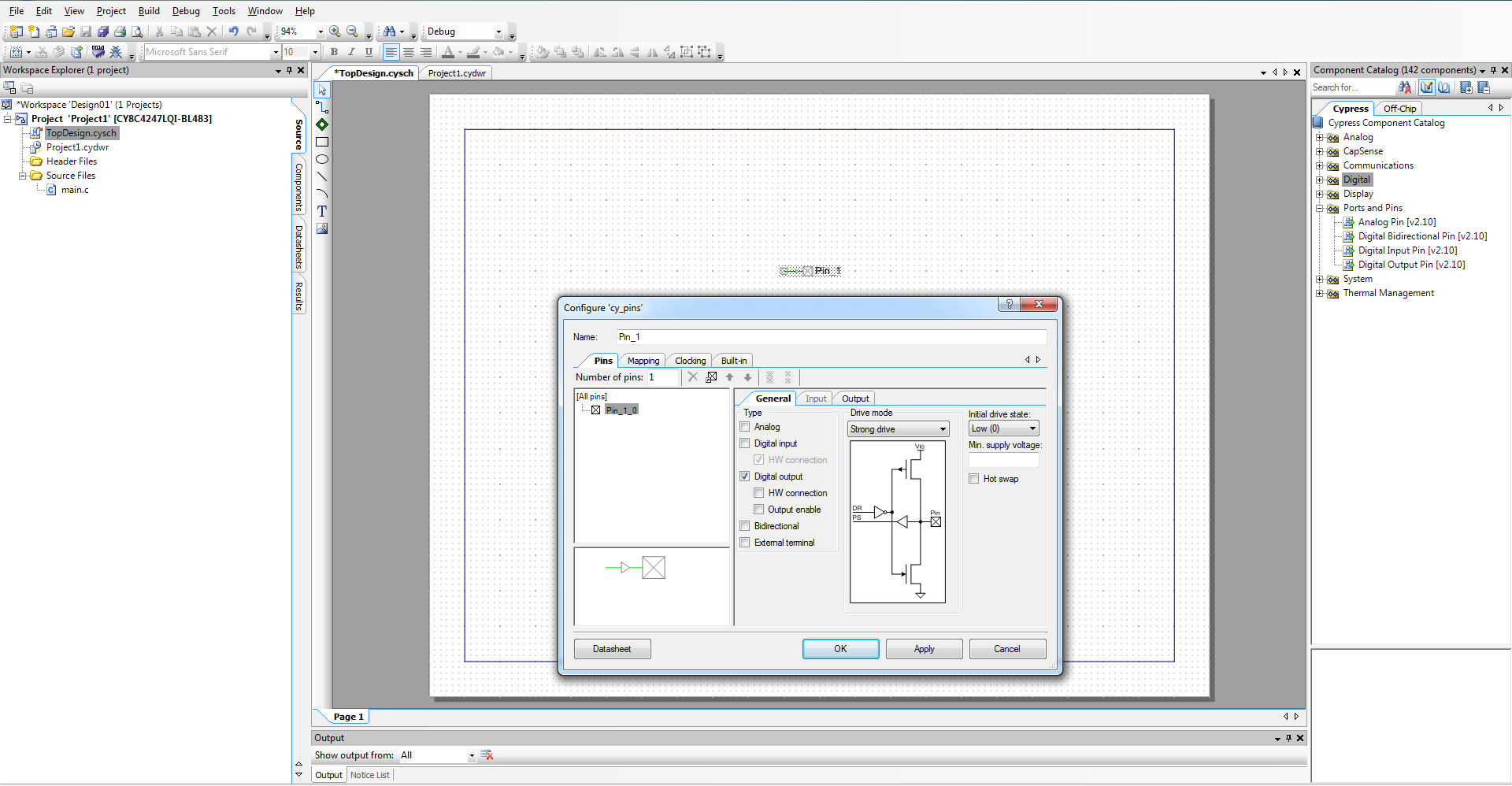
1. using psoc creator

The provided Keil MDK project contains all the configuration files needed for this project. These files are exported using PSoC Creator from Cypress. The PSoC Creator is capable of generating the hardware configuration file as well as developing software applications on PSoC platforms. Click [here](http://www.cypress.com/psoccreator/) to learn more about PSoC Creator.

To create a new project using PSoC Creator:

* Open PSoC Creator . You can find it at Start -> all Programs -> Cypress -> PSoC Creator 3.1
* On the top menu select File -> New -> Project to create a new project.
* A New Project widow will appear. Select your board using the Default Templates. In this case it is PSoC 4100 BLE / PSoC 4200 BLE Design. At the bottom give a name to your project and the location where you want to store it. Click OK.

The workspace will open. On the left side of the window it is the Workspace Explorer where you can see the project structure and all the files. At the central part of the screen there are current files you are editing. Notice that you can switch between the open files using the tabs on the top. On the right side those are the tools or properties of the file you are editing.



Workspace: 1. Workspace Explorer, 2. Editor, 3. Tools, 4 Configuration window

The new project has 3 files: TopDesign.cysch, [project\_name].cydwr and main.c.

### add components

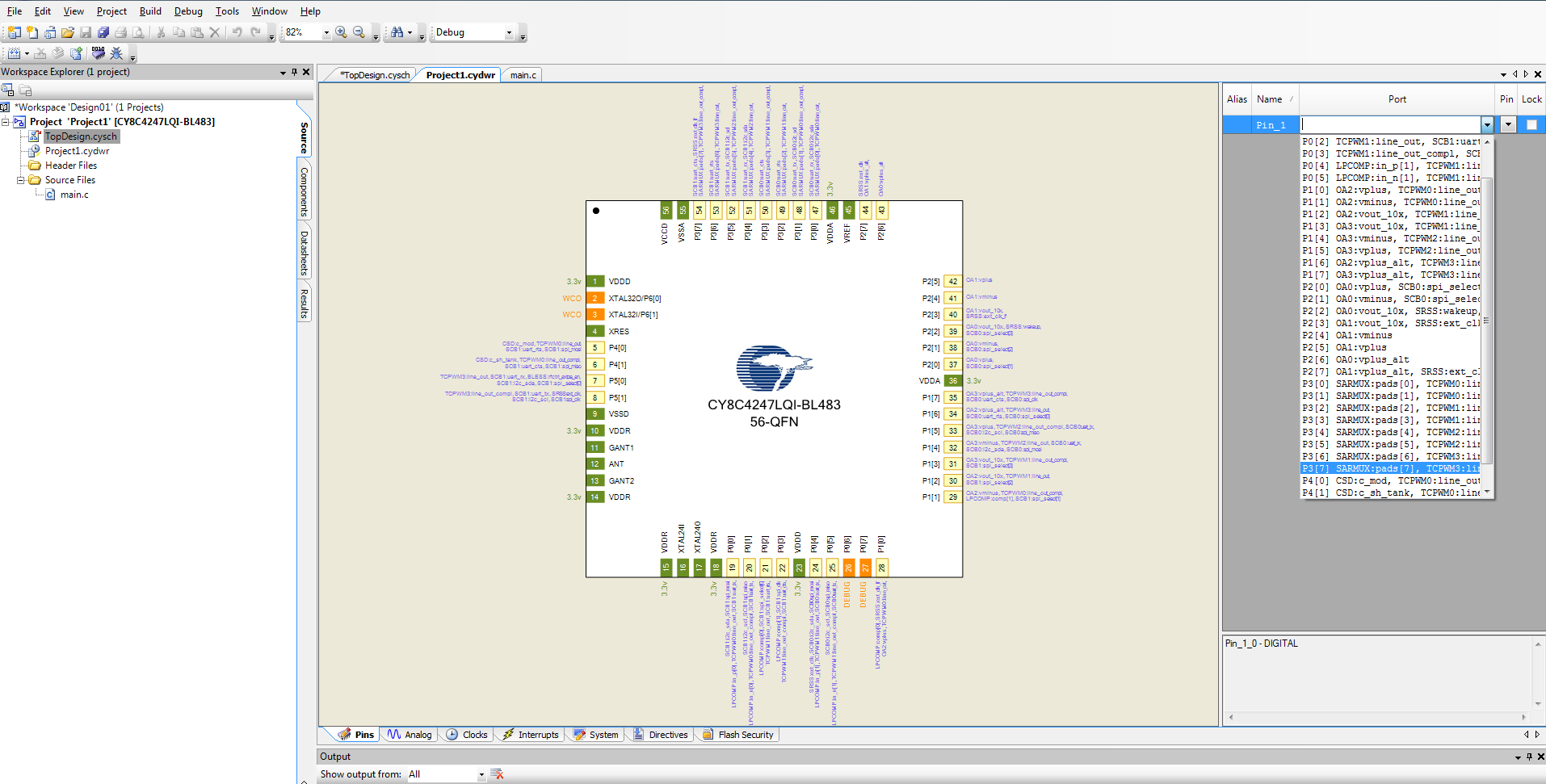
The TopDesign.cysch is the schematic window where you can add and configure components on the programmable hardware. On the right side there is the list of components. You can drag and drop them into your project. In this lab we need a *Dgital Output Pin* object to control a LED. We can add more objects and connect them using the connector tool.

Double-click on an object and open the configuration window. Here we can change the object name and configure the properties. Deselect the *HW connection* and keep all the other parameters as default.

### configuration

Once we have added the object to the schematic we can configure the Design Wide Resources (DWR). Open the .cydwr file. Here we can configure overall setting such as pin assignments, clock configurations, system debugging options.

At the bottom of this page there are tabs to navigate between the different types of settings. On the Pins tab we can wire the pin objects to MCU pins. For example to control a blue LED of the onboard RGB LED we have to link a *Digital Out object* to pin P3[7]. All the pin objects we have on your schematic will be listed on the right side of the window. Click the drop-down list under the Port column and assign the pin.



### write software program

The last step is to program the processor. Before that we need to build the project to generate the components source files. To do that, click Build -> Build your project.

Open the main.c source file. Here we can edit the software program. For example to blink an LED we can write;

For(;;)

{

Pin\_1\_Write(~Pin\_1\_Read());

CyDelay(1000);

}

Build the project again and then press Debug (F5 Key) to test the program.

To export the project to Keil MDK, select *Project -> Export to IDE* and then click *Export to µVision*.