RPiSoC

Revision C User Guide

Doc. v1.03, Modified 7/21/14

Originally created for:

The Design Lab at Rensselaer
110 8th St
Troy, NY 12180
Phone: 518-276-6626
http://lamp3.server.rpi.edu/soe/mdl

Modified for:

Embedit Electronics
Email: embedit@embeditelectronics.com
http://www.embeditelectronics.com/

Document Revision History

Date	Revisions Made	Name
12/7/2013	Initial Creation of Document	Eric Arntzen
12/9/2013	Added images of Revision B board	Todd Anderson
7/21/2014	Updated to reflect Revision C changes	Robert Barron

Contents

1 Introduction

Thank you for your interest in the RPiSoC. This platform is intended to serve as a development platform for the PSoC 5 family of mixed signal array processors. The board features a variety of headers and connectors for interfacing with external hardware, including the Raspberry Pi. The board features a header for connecting to the Raspberry Pi, Arduino headers, and Pmod (5V) headers. In addition, the board features six 10-pin female headers for connecting external hardware to the various GPIO pins of the PSoC 5 chip (Ports 0, 1, 3, 4, 5, and 12). The board also has two 12-pin right angle headers that connect to the PSoC 5 chip (Ports 2 and 6). Keep in mind the GPIO on the PSoC has a very different meaning than on the Raspberry Pi or similar boards. With this board GPIO means access to both digital and analog functionality, and all communication protocols, routed to nearly any pin as needed by your project. In addition an API has been developed that allows the Raspberry Pi to access specific functionality of the RPiSoC through an easy to use python interface.

One external oscillator is available for use, rated at 32.786 KHz for real-time clock use. The board also has a Micro-USB connection that provides USB communication and power to the PSoC 5LP chip. Creating projects for the RPiSoC is done using the free PSoC Creator software. Programming the Revision C board is completed over USB via the Bootloader that comes pre-installed, or using the 5-pin programming header with the Cypress Mini-Prog 3 programmer. Programming the board over USB is covered in detail in section 6.

1.1 Quick Links

Software

PSoC Creator 3.0 (Windows only. Parallels VM is supported for OSX) found here (Direct link) or from here (PSoC Creator Landing page)

The RPiSoC github page, including bootloader files, found here (If you are unfamiliar with using git, just click the “Download Zip” option in the bottom right hand corner when looking at a project)

2 Known Issues

This section describes the issues that exist with the Revision C PCB that need to be manually corrected for the board to function properly.

1. The Programming slide-switch (SW) is floating on both sides. The PROG side should be connected to GND. (Fixed in revision C1)

3 Component List

Below is a table listing the details of the components that can be used to populate the Revision C board. Replacement of a listed component with an alternative part is possible, but should be done if necessary.

Table 1: Component List

Qty	Reference	Value	Description
1	Y1	32.768kHz	Oscillator
6	P0, P1, P3, P4, P5, P12	10-Pin	Female Header
2	P2, P6	12-Pin	Female 90-Degree Header
7	C1, C2, C4, C5, C7, C8, C10, C21, C22	0.1 uF	Capacitor
8	C3, C6, C9, C11, C17, C18, C19, C20	1.0 uF	Capacitor
1	C16	100 uF	Capacitor
1	PROG	5-Pin	Mini-Prog 3 90-Degree Male Header
2	C14, C15	18pF	Capacitor
2	C12, C13	12pF	Capacitor
2	R3, R4	1M	Resistor
2	R1, R2	22	Resistor
2	R6, R7	1k	Resistor
2	8-Pin	8-Pin	Arduino Shield Female Connector
2	6-Pin	6-Pin	Arduino Shield Female Connector
1	U1	-	Cypress PSoC 5LP 035
2	USB	-	Micro-USB Port
1	RST	Momentary	Pushbutton
1	S2	-	Switch
1	COMMLED	Yellow	Activity LED
1	PWRLED	Red	Power LED
1	JP1	3-Pin	Male Jumper
1	JP1	2-Socket	Female Jumper Shunt
1	F1	400mA	Fuse
1	ICSP	6-Pin	Male 2x3 Arduino ICSP Communication Header
1	IC1	3.3V, 0.2A	Voltage Regulator

4 Safety

4.1 Discarding

This device contains materials that cannot be discarded through traditional waste management means. Please contact your nearest electronic recycler for more details before discarding this board.

4.2 Electrostatic Discharge Protection

This board contains devices that are sensitive to electrostatic discharge. To avoid damaging discharge, wear an antistatic wristband or other antistatic device while handling the board.

4.3 Handling the Board

Hold the board from the edges when handling the board. Place the board on an insulating static free surface when operating or working on the board. Do not slide the board on the surface, as this may damage underside leads.

4.4 Power and Voltage

Power is applied by the Mini-Prog 3 programming header, the Micro USB port, or the Raspberry Pi GPIO power pins of the GPIO header. The white 3-pin jumper next to the Micro USB port allows you to select between USB and Raspberry Pi power, ensuring that neither connection can back feed, and allowing USB communication to be used while still connected to the Raspberry Pi. The Mini-Prog 3 header does not have the same protection, so ONLY APPLY POWER TO A SINGLE CONNECTION AT A TIME. **DO NOT** APPLY POWER VIA THE MINIPROG3 AND THE RASPBERRY PI/ USB AT THE SAME TIME.

Also, make sure the Raspberry Pi is turned off before plugging the RPiSoC into it. Plugging the RPiSoC into an active Raspberry Pi may cause it to restart, which can lead to SD card corruption.

5 Board Layout

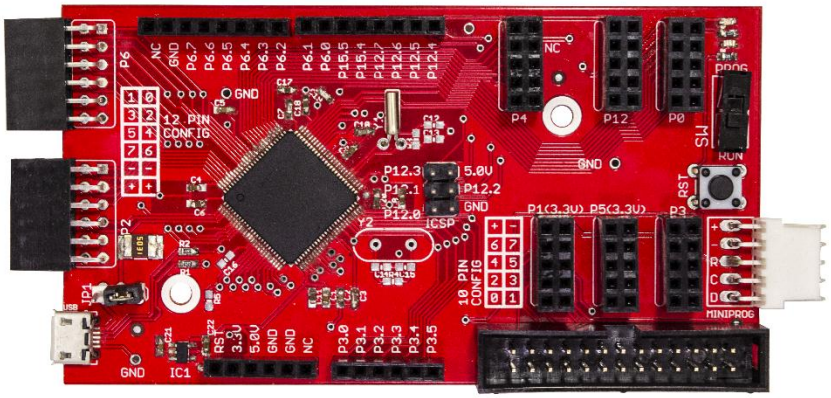


Figure 1: Top View of RPiSoC

5.1 Arduino Headers

The Arduino headers on this board are compatible with a variety of Arduino shields with a 0.1” spacing between adjacent pins. The board features an ICSP header for communication with advanced Arduino.

At the base of each pin is the label of the PSoC 5 port that the pin is connected to, as shown in Figure 2. If a pin is labeled NC, it is not electrically connected to any portion of the board and is intended to provide only physical connectivity. Note that the corresponding Arduino pinout is silkscreened on the bottom of the board.

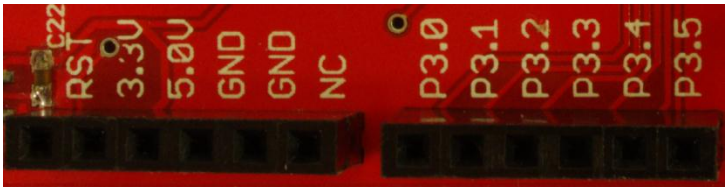


Figure 2: Top-Side Silk Screening Indicates PSoC 5 Port Connections

5.2 Raspberry Pi Connections

This board can connect to the Raspberry Pi via the GPIO pins, utilizing SPI, I2C, or UART communication protocols. When viewing the board from the top side, Pin 1 of the RPi

GPIO header is located in the top-right and Pin 26 is in the bottom-left. Table 2 describes how the Raspberry Pi headers are connected to the PSoC 5 chip.

Table 2: Raspberry Pi Header Pinout

Protocol	Line Name	Raspberry Pi GPIO Pin	PSoC 5 Pin
I2C	SDA	3	27 (P1.6)
	SCL	5	28 (P1.7)
UART	TX	8	22 (P1.2)
	RX	10	25 (P1.5)
SPI	MOSI	19	24 (P1.4)
	MISO	21	23 (P1.3)
	SCLK	23	21 (P1.1)
	CE1 N	26	Not Connected

5.2.1 Raspberry Pi Connection via 2x13 Male Header

A 26-pin ribbon cable (Adafruit link) can be used to connect a Raspberry Pi model A or B to the RPiSoC. For the Raspberry Pi model B+, use this cable instead, which converts the B+'s 40-pin header back to the 26-pin form factor.

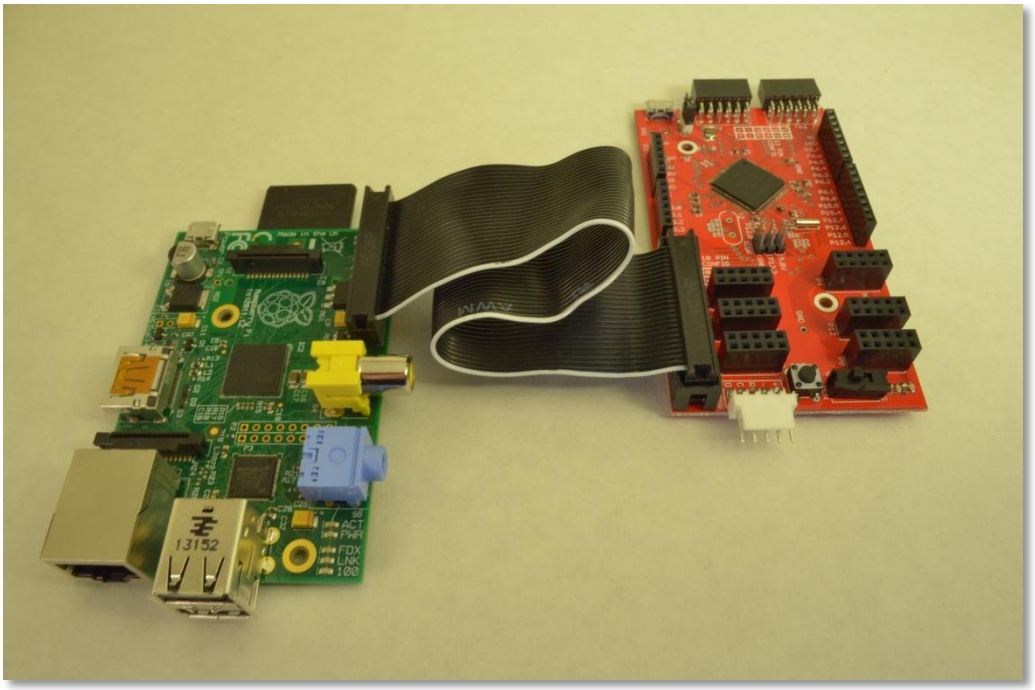


Figure 3: Shrouded Header w/ Ribbon Cable

5.3 RPiSoC GPIO Headers

Six 10-pin PSOC GPIO headers are available in the top right corner of the Revision C board. Each is labeled with the connected PSoC 5 port. The 8 pins of a port occupy the lower 8 connections and the top two connections supply power and ground. The pinout for each of the PSoC GPIOs follows the diagram on the board. Ports 1 and 5 are set to have a 3.3V logic level, and their power (+) headers provide 3.3V. (NOTE: 3.3V on the power (+) headers is as of revision C1. Previous versions provide 5V. You can check your revision by looking at the back of the board under the USB port.) All other ports are set to 5V.

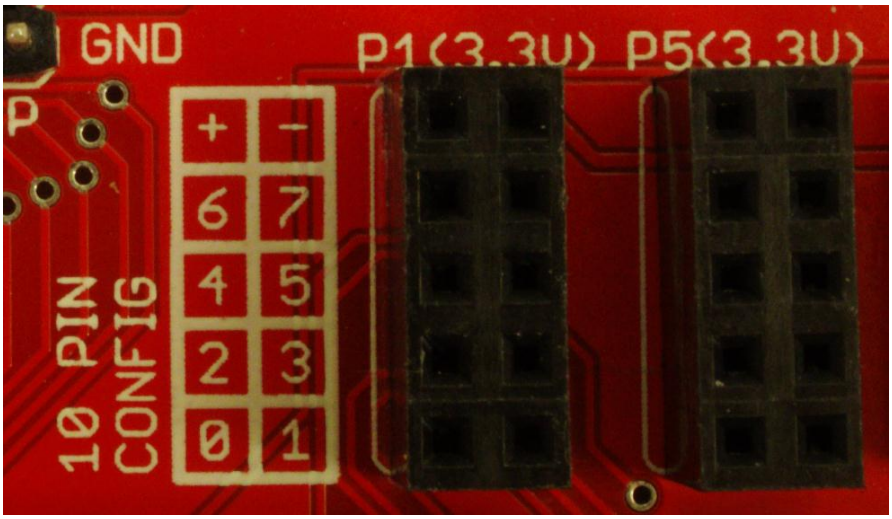


Figure 4: GPIO Pinout

5.3.1 SIO Header

Port 12 of the RPiSoC board has Special I/O pins that are different from normal GPIO. These pins support a higher current of 25mA compared to GPIO's 8mA. They also support hot swap capability and programmable output voltage levels and input thresholds. However, these pins do not support all the functions of GPIO, namely analog connectivity.

5.4 5-Pin Mini-Prog 3 Header

The Revision C board features a 5-pin Mini-Prog 3 header for programming via the Mini-Prog 3 and PSoC Creator software program from Cypress Semiconductor. The programming header is shown below. Use of the Mini-Prog allows for additional functionality, including debugging.

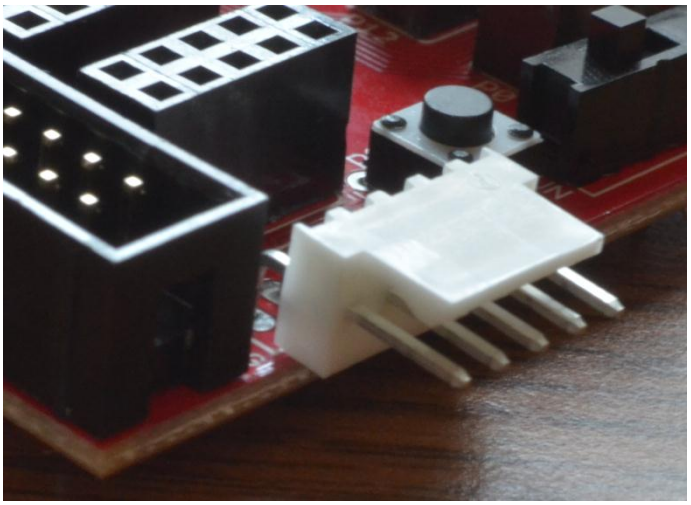


Figure 5: MiniProg3 Connector

5.5 Micro-USB Port

The Micro-USB port is located in the bottom-left corner of the board, as shown in figure 8.

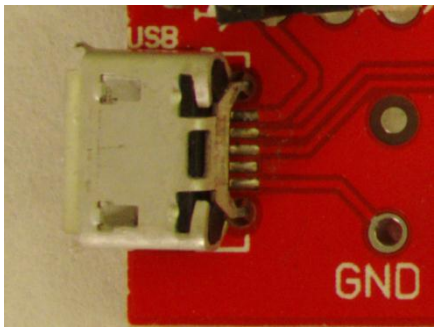
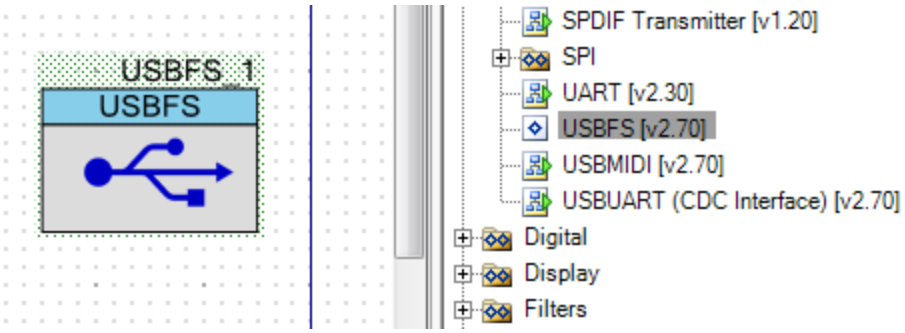


Figure 6: Micro-USB Port

The RPiSoC has full USB HID and Host capabilities. You can use the micro USB port to enable the RPiSoC to act as a keyboard, mouse, joystick, serial communications device and more. You can also program the board over USB from directly within PSoC Creator. Using USB in your design is as simple as dropping in the USBFS component in PSoC Creator, and configuring it using the built in GUI interface.



PSoC Creator has multiple example projects that show the different uses of the USB component, so like most things PSoC, you won't have to start from scratch.

5.6 Pmod Headers

The board features two peripheral modules (Pmod) headers that are turned at right angles and connected to Ports 2 and 6. When viewed from the left looking rightward, the pinout is shown below. You can configure the pins of the RPiSoC to use whatever interface is necessary for communicating with a Pmod device.

COMPONENT SIDE OF BOARD

n.0	n.2	n.4	n.6	GND	5V
n.1	n.3	n.5	n.7	GND	5V

BACK SIDE OF BOARD

Figure 7: Pmod Pinout

5.7 External Oscillators

The Revision C board features one external oscillator clocked at 32.768 KHz, allowing for a real time clock. It can be enabled via the 'Edit Clock…' button on the Clocks page of the project .cydwr file. You can also add another Oscillator at spot Y2 on the PCB, as shown in figure 10. The PSoC 5LP chips supports ranges from 4 to 25 MHz on the connected pins.

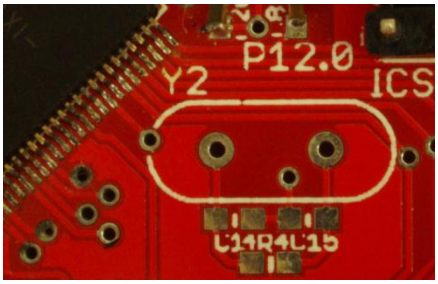


Figure 8: MHz Oscillator can be added here

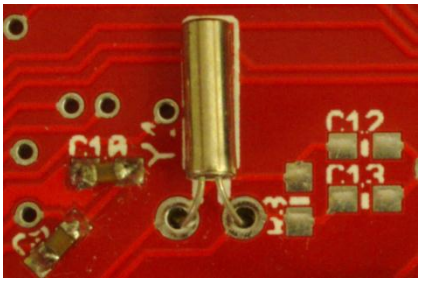


Figure 9: 32.768 KHz Oscillator

5.8 Reset Button

The Revision C board features a pushbutton to reset the PSoC 5 via XRES (Pin 15 on the PSoC 5LP).

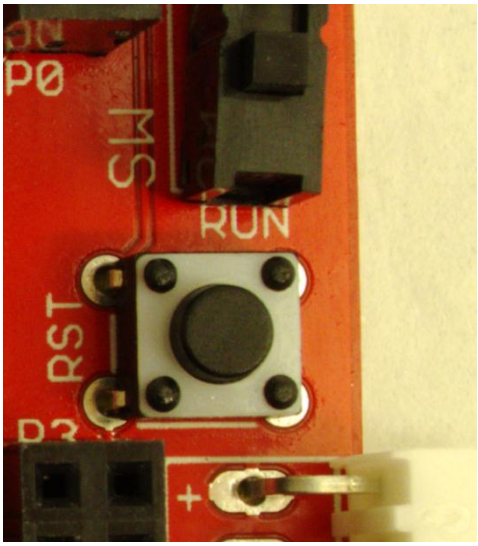


Figure 10: Reset Pushbutton

5.9 Programmer Mode Switch

The Revision C board is programmable using USB via its preinstalled bootloader. To activate the bootloader simply move the slide-switch on the right side of the board to the “PROG” position, and then reset the board. Windows should automatically detect and install the appropriate drivers. To program the board, you can use the Bootloader Host Tool included in the PSoC Creator software. Section 6 of this user guide includes a step-by-step example of creating a project and programming the board with it.

Once a program is on the board, you can move the slide-switch to the “RUN” position, and then reset the board to run your program.

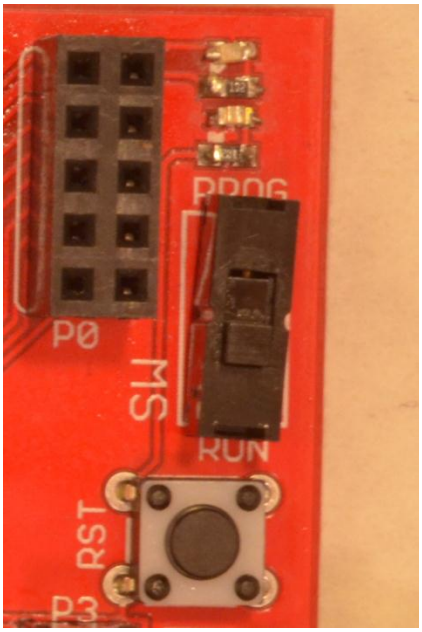


Figure 11: Programmer Mode Switch

5.10 LEDs

Two LEDs are mounted on the surface of the board. One is the red Power LED to indicate if the board is currently receiving power. The other is the yellow Display LED which is connected to Pin 12.0 and can be used by your programs.

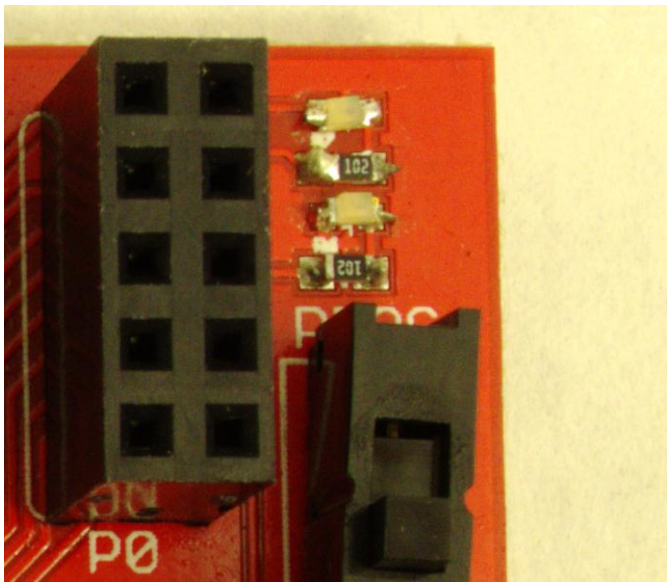


Figure 12: Yellow LED (left) and Red LED (right)

5.11 Power Jumper

The power jumper is used to switch the board between receiving power from the USB port or the Raspberry Pi. Putting the jumper in the position closest to the USB port enables power via USB.

6 Example Program

6.1 Blink LED

6.1.1 Description

This starter program steps you through the process of programming the board using the bootloader. The program will blink an LED at a regular interval utilizing the PSoC's PWM modules.

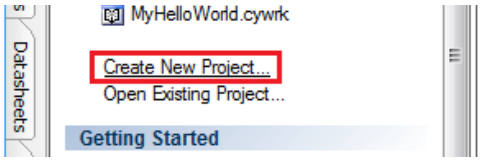
6.1.2 Preparation

For this example, you will need the following items:

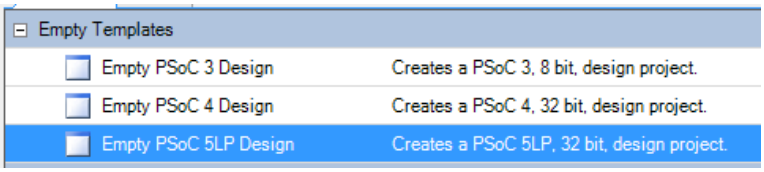
- Revision C board
- Micro USB to USB cable
- A Windows computer with
    - An internet connection (First time use only)
    - PSoC Creator 3.0 installed, found here (Direct link) or from here (PSoC Creator Landing page)
    - The RPiSoC bootloader files, found here (If you are unfamiliar with github, just click the "Download Zip" option in the bottom right hand corner)

6.1.3 Software Side

1. First, open PSoC Creator 3.0 and select 'Create New Project…'



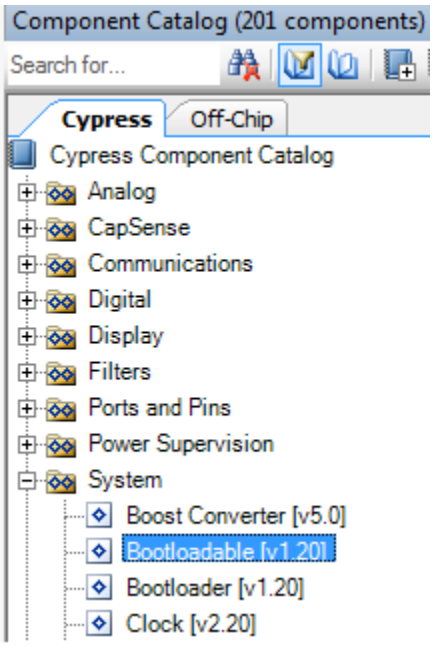
2. Then select 'Empty PSoC 5LP Design' under 'Empty Templates'



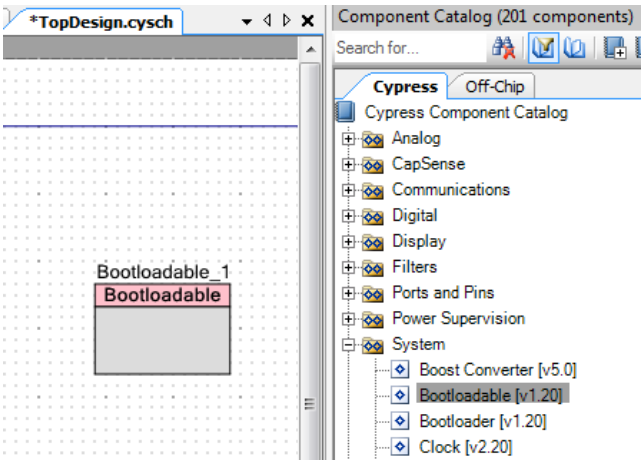
3. Now click the plus icon next to the word ‘Advanced’ at the bottom. Click on the drop down menu next to the ‘Application type’ field, and select the ‘Bootloadable’ option.

4. Choose a name and location for your project. Then click OK. At this point you will see several panes in the interface displaying different information. Let’s focus on the right hand side of the screen at the ‘Component Catalog’ pane.

5. With the ‘Cypress’ tab selected, find the ‘Bootloadable’ component (System > Bootloadable).

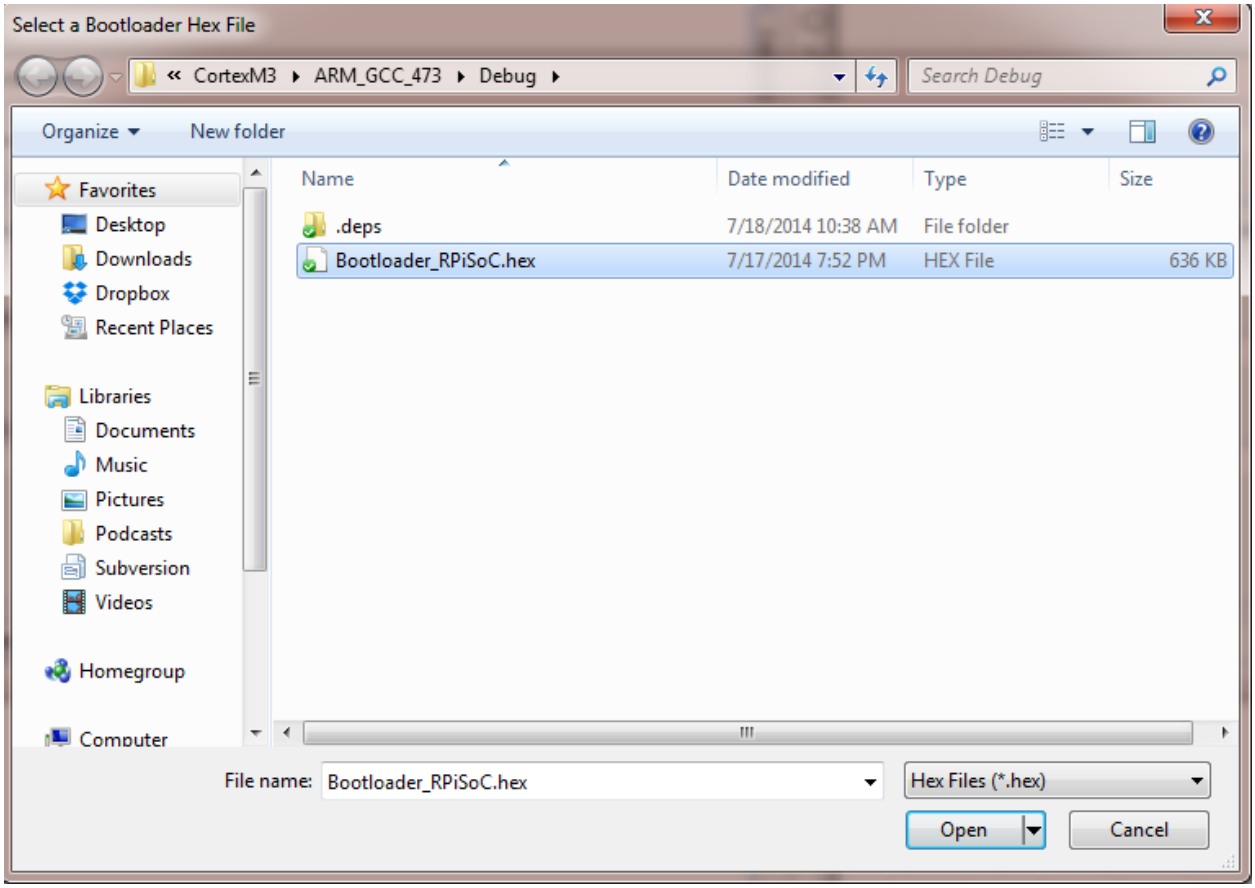


6. Drag and drop the Bootloadable component from the ‘Component Catalog’ to the ‘TopDesign.cysch’ diagram page in the middle of the screen.



7. Now you need to configure the Bootloadable. Double click on it, or right click it and then click ‘configure’. Click on the ‘Dependencies’ tab and then click ‘Browse’. You need to open the file ‘Bootloader_RPiSoC.hex’. From the folder you downloaded RPiSoC_Bootloader in the path should be:

   “your_downloads_folder”\RPiSoC_Bootloader\Bootloader_RPiSoC.cydsn\CortexM3\ARM_GCC_473\Debug

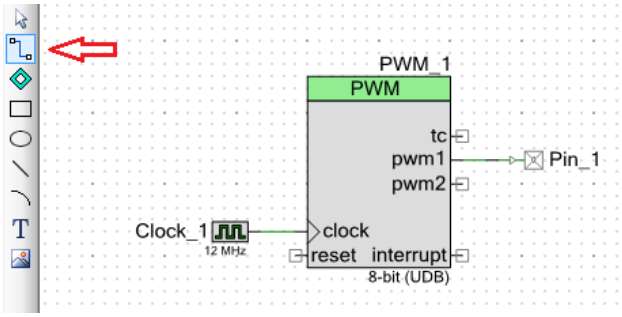


8. Once you have found the hex file, click ‘open’. The path to the Bootloader should now be filled in on the Bootloadable configuration screen. You can now hit ‘OK’ to leave the configuration screen.

9. Next we need to pull in the ‘PWM’ component from the Component Catalog pane. It can be found under Digital > PWM. Add the PWM component to the diagram once you have found it.

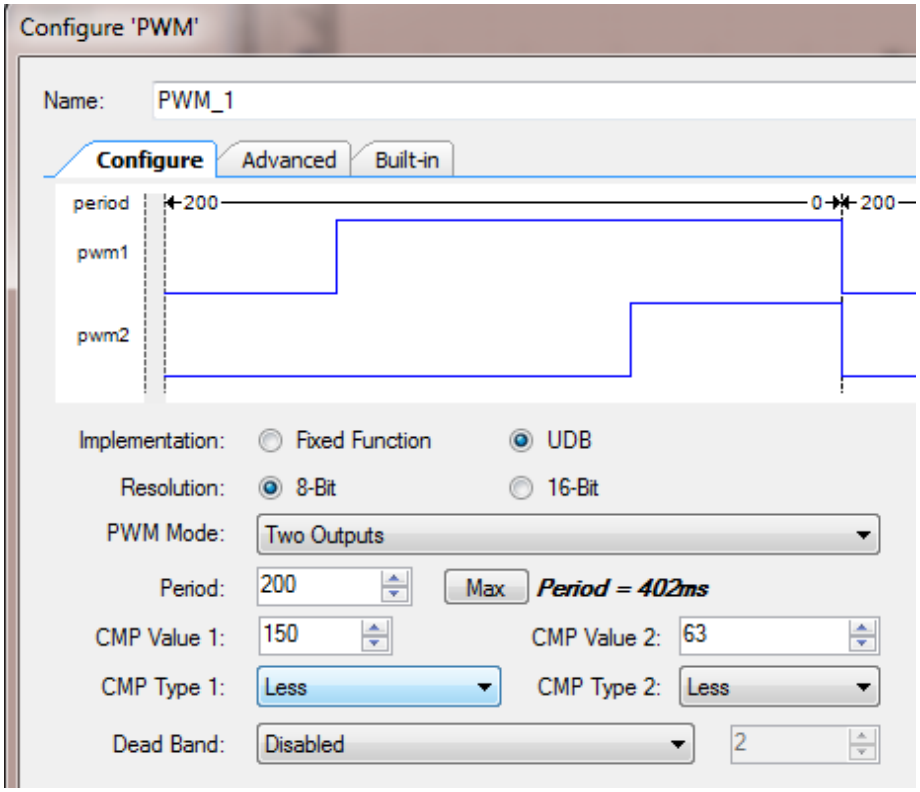
10. The PWM block will drive the pulsing waveform to power the blinking LED. The PWM block needs a ‘Clock’ component to drive it and a ‘Digital Output Pin’ to send the signal to. Add both of these components to the diagram now. ‘Clock’ is found under System > Clock and the ‘Digital Output Pin’ is found under Ports and Pins > Digital Output Pin.

11. Place the Clock near the PWM's clock input and the Pin near the PWM's pwm1 output. Then use the wire tool to connect the components.



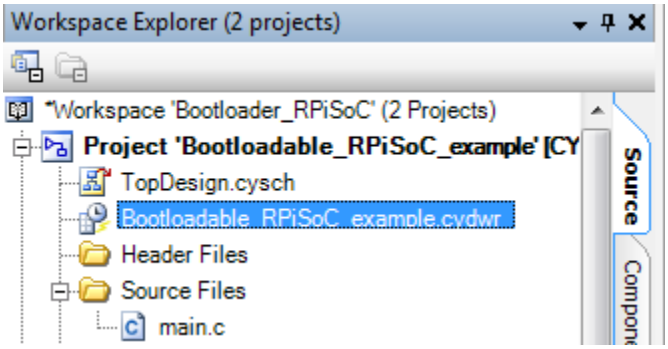
12. We will configure the Clock component frequency next. Double click the Clock component and change the frequency from 12MHz to 500 Hz. This can be done by clicking on the 'Hz' option in the dropdown menu next to frequency, and then typing 500 into the white box.

13. Now we will configure the PWM component. Double click the PWM component and change the 'Period' to 200 and the 'CMP Value 1' to 150. This will provide a period of approximately 402ms and cause the LED to be on for 236ms every cycle.

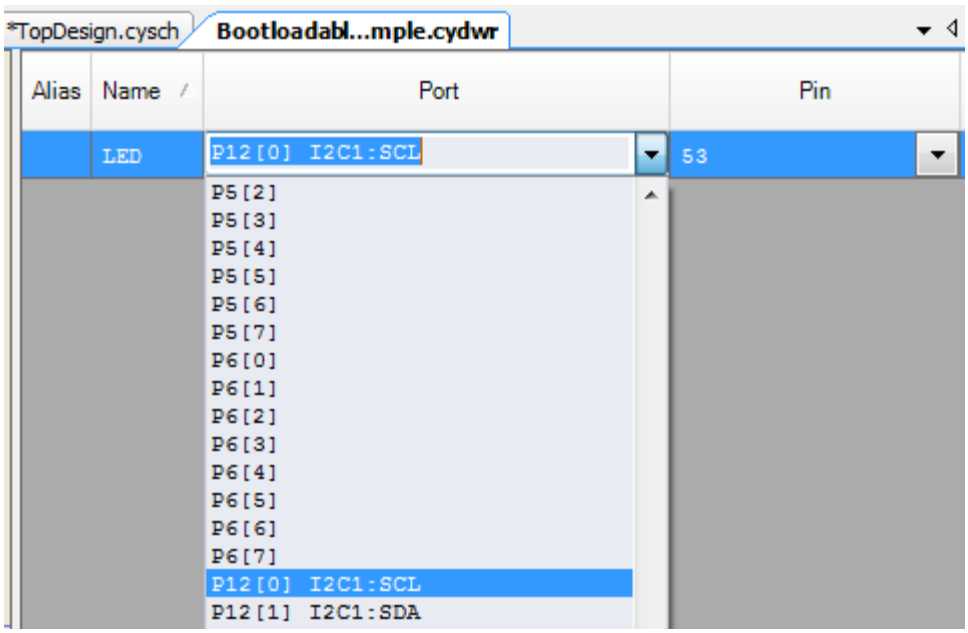


14. Next we want to rename the output pin connected to the PWM output to have a more appropriate name. While not strictly necessary, it will help you keep track of what your inputs and outputs do. Double click on the Pin_1 component and then type “LED” and hit enter.

15. Now we will configure what physical port on the board the Pin component represents. This is necessary because the virtual “LED” pin you attached in the schematic only gets assigned a specific physical pin if you explicitly tell it to. This lets you connect your virtual schematic pins to nearly any physical port on the RPiSoC, allowing for very flexible designs. Navigate to the ‘Workspace Explorer’ pane on the left side of the PSoC Creator, and then double click the “’your_project_name’.cydwr” file.



16. Select the Port dropdown next to “LED” and select P12[0] I2C1:SCL. This means that the output of the PWM connected to pin “LED” will end up being output on P12.0 of the physical board. While you can assign the virtual “LED” pin to nearly any physical pin on the board, we choose P12[0] because it is attached to an actual LED on the RPiSoC.

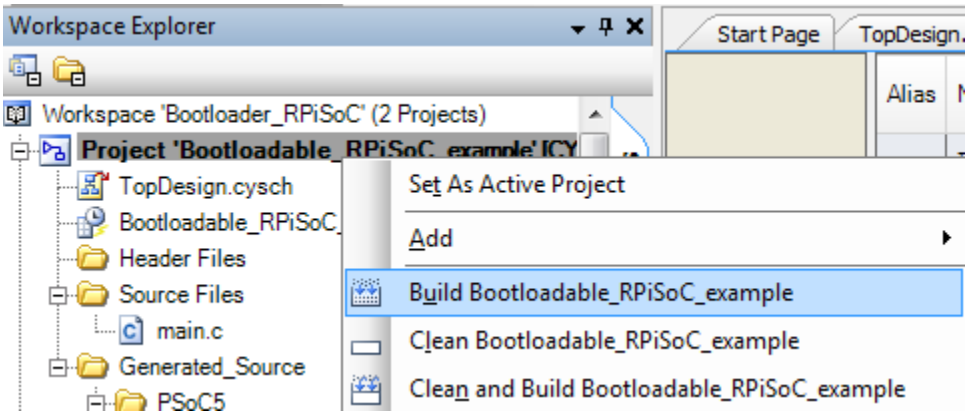

*TopDesign.cysch
Bootloadabl...mple.cydwr
Alias
Name
Port
Pin
LED
P12[0] I2C1:SCL
53
P5[2]
P5[3]
P5[4]
P5[5]
P5[6]
P5[7]
P6[0]
P6[1]
P6[2]
P6[3]
P6[4]
P6[5]
P6[6]
P6[7]
P12[0] I2C1:SCL
P12[1] I2C1:SDA

17. Next, we need to tell the PWM component to start when the RPiSoC is turned on. Open the “main.c” file under the “Source Files” folder in the “Workspace Explorer” pane on the left. Add “PWM_1_Start();” inside the “int main()” function, as shown below.

```
#include <project.h>

int main()
{
    /* Place your initialization/startup code here (e.g. MyInst_Start()) */
    PWM_1_Start();
    /* CyGlobalIntEnable; */ /* Uncomment this line to enable global interrupts. */
    for(;;)
    {
        /* Place your application code here. */
    }
}
```

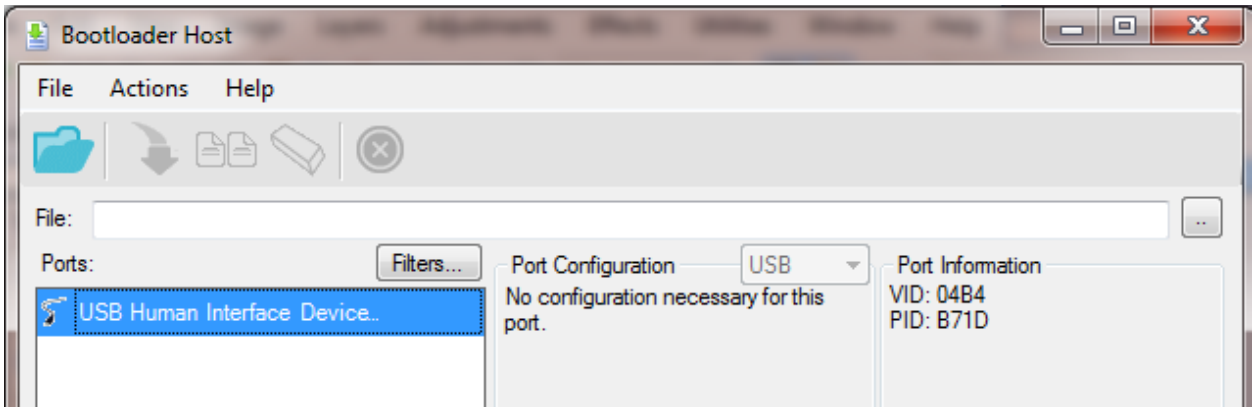
18. Now we are ready to build the project. In the “Workspace Explorer” pane Right-click on “Project ‘your_project_name’” and then left click on “Build ‘your_project_name’”. PSoC Creator will now compile your project. Wait until it says “Ready” in the bottom left hand corner of the PSoC Creator window.



19. If you haven’t already, go ahead and plug your RPiSoC board into the computer with a micro-USB cable. The red power LED should illuminate on the board. If it does not, make sure the jumper next to the USB port on the board is in the correct position. Drivers should automatically install, and then windows should tell you that your device is ready to use.

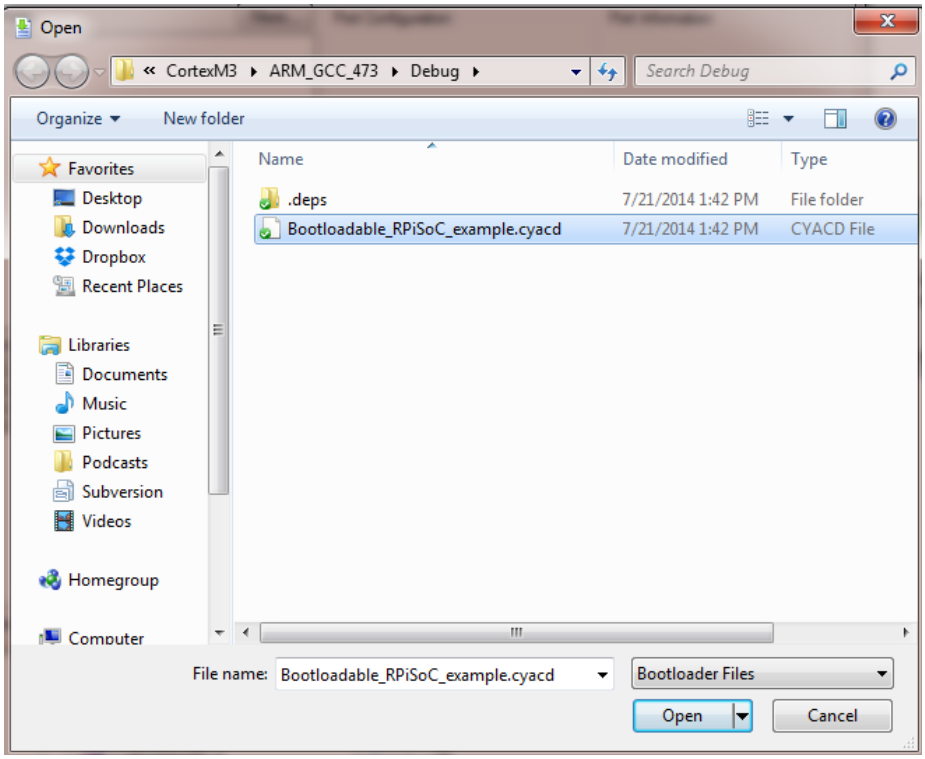
20. On the RPiSoC board, move the slide-switch into the “PROG” position and then press and release the “RST” button.

21. Now we are ready to program the board. First click on “Tools” in the top of PSoC Creator, and then click “Bootloader Host”. The RPiSoC will show up in the Ports section of Bootloader Host as “USB Human Interface Device”



22. Now you need to point the Bootloader Host tool to the project file you created. Click on the “..” button to the right of the file text box. You need to open the “’your_project_name’.cyacd” file. The path should be:

“your_project_folder”\”your_project_name”.cydsn\CortexM3\ARM_GCC_473\Debug\”your_project_name”.cyacd



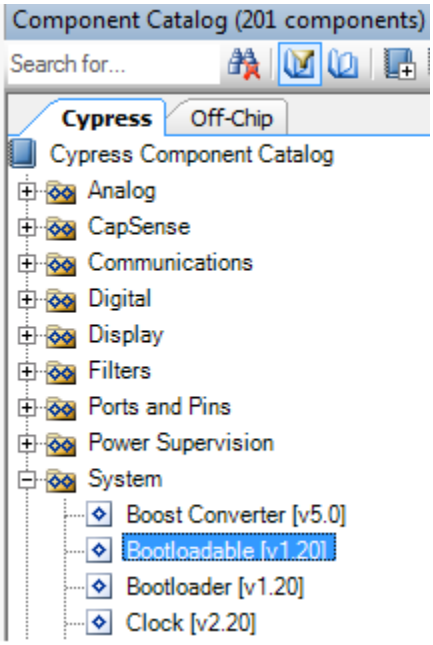
23. Finally, you can program the board by clicking the blue downward facing arrow near the top of the screen, or hit F5. The yellow LED on the board should now be blinking. If you want to program your board again, simply restart it with the slide-switch in the PROG position, which will bring it into programming mode. To run your program again, simply restart it with the slide-switch in the RUN position and wait 2 seconds.

7 Converting an existing project for Bootloading on the RPiSoC

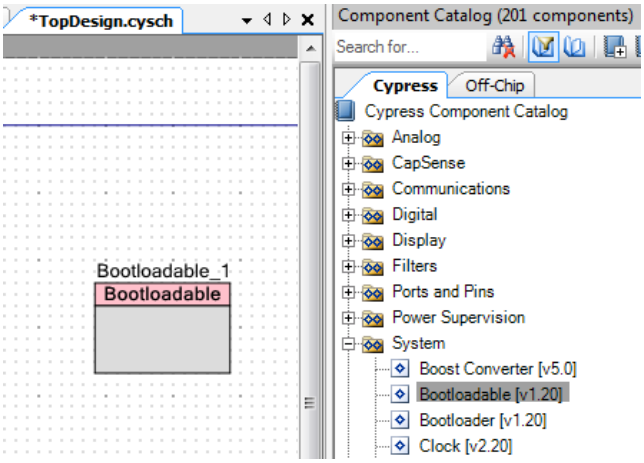
This section will explain how to convert an existing PSoC Creator project, such as a PSoC Creator example project, to a Bootloadable one, allowing you to program the RPiSoC with it. Prerequisites are the same as the example project in section 6.

1. First, open your project. This guide will assume you are using the HelloWorld_Blinky example project. Once in PSoC Creator you can open it by going to File > Example project. Then select “PSoC 5LP” for Architecture. Then scroll down until you find “HelloWorld_Blinky” and open it.

2. Now make sure you are in the “TopDesign.cysch” window. If you aren’t, double click on “TopDesign.cysch” in the Workspace explorer pane on the left. Now navigate to the Component Catalog pane on the right. With the “Cypress” tab selected, find the “Bootloadable” component (System > Bootloadable).

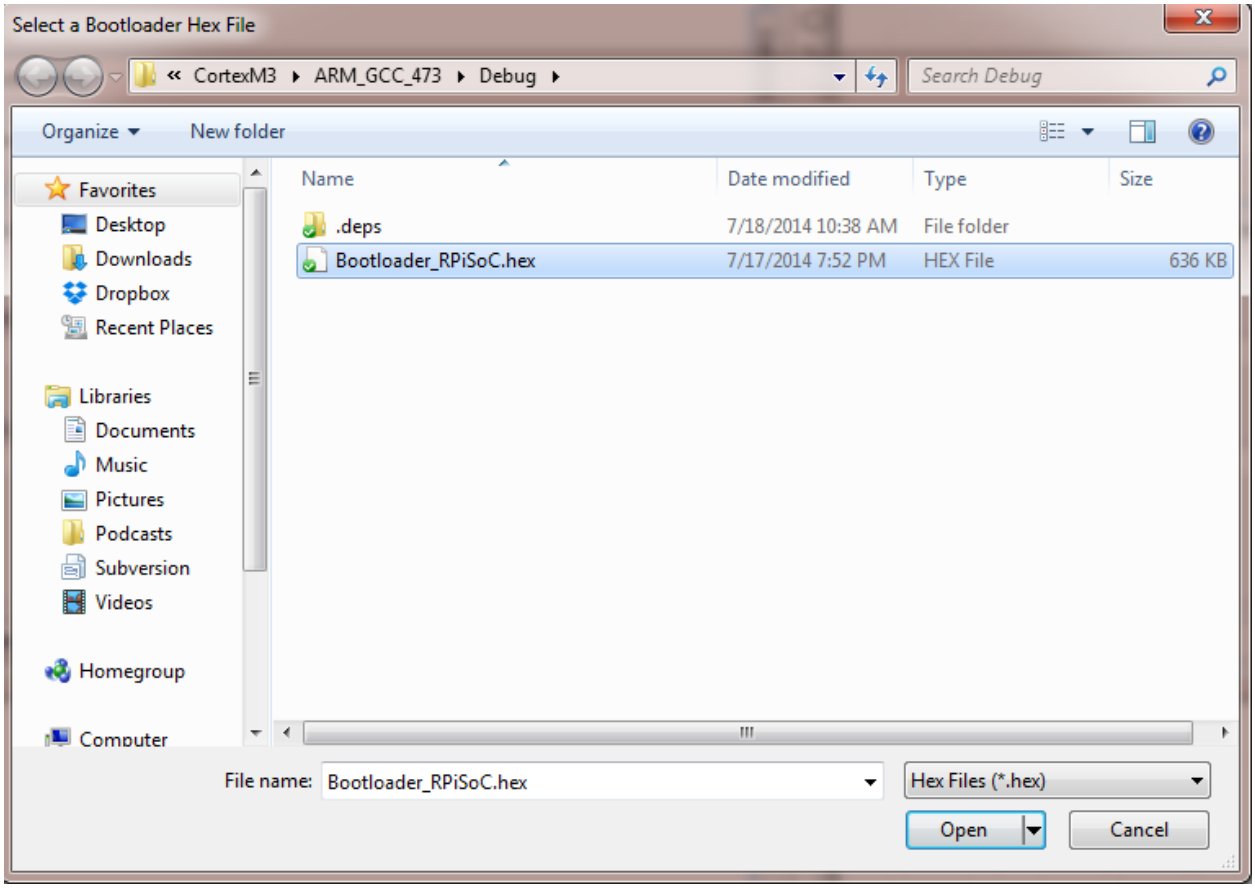


3. Drag and drop the Bootloadable component from the Component Catalog to the TopDesign diagram page in the middle of the screen.



4. Now you need to configure the Bootloadable. Double click on it, or right click it and then click "Configure…" Click on the "Dependencies" tab and then click "Browse". You need to open the file "Bootloader_RPiSoC.hex". From the folder you downloaded RPiSoC_Bootloader in the path should be:

   "your_downloads_folder"\RPiSoC_Bootloader\Bootloader_RPiSoC.cydsn\CortexM3\ARM_GCC_473\Debug

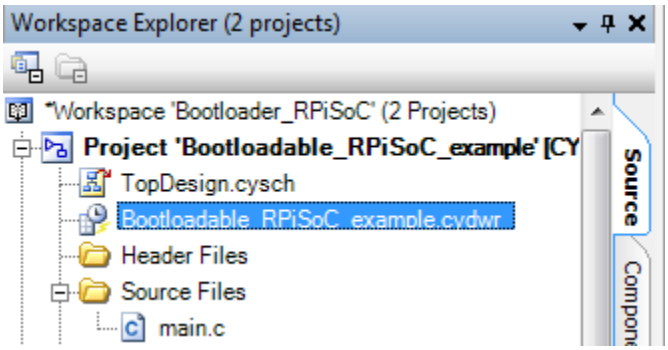


5. Once you have found the hex file, click “Open”. The path to the Bootloader should now be filled in on the Bootloadable configuration screen. You can now hit ‘OK’ to leave the configuration screen.

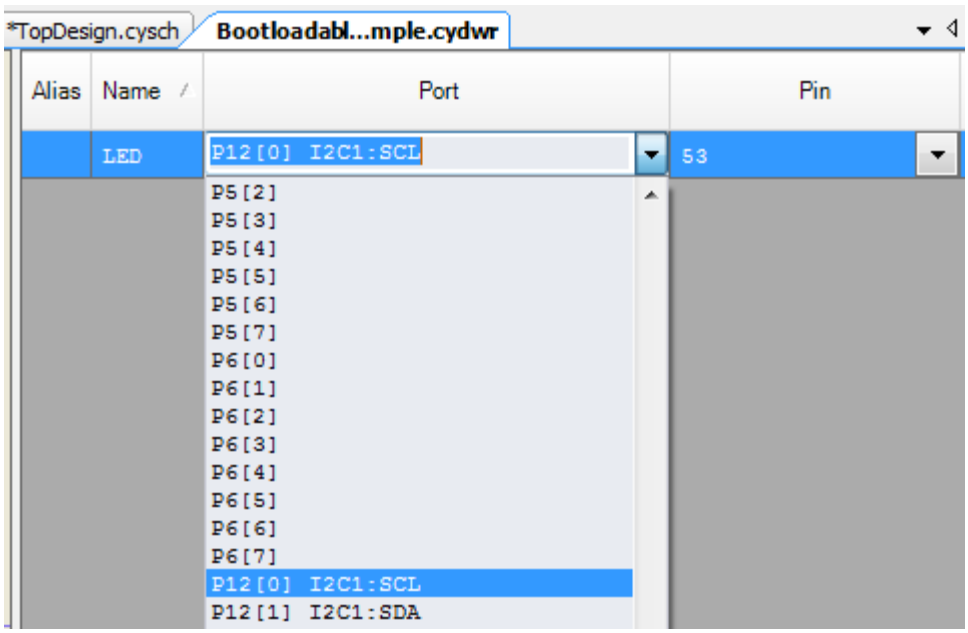
6. Now we need to tell PSoC Creator that this is going to be a Bootloadable application, so it can take advantage of the Bootloadable component. At the top of the PSoC Creator window, go to Project > Build Settings. Then in the Code Generation tab select “Bootloadable” under Application Type.

7. NOTE: Keep in mind that example projects will likely have a pin configuration that is not ideal for your projects. In HelloWorld_Blinky's example, the LED will be mapped to the wrong pin. If you know how to configure PSoC Creator pins, you can skip to step 9. The next two steps show how to configure a pin in the HelloWorld_Blinky example.

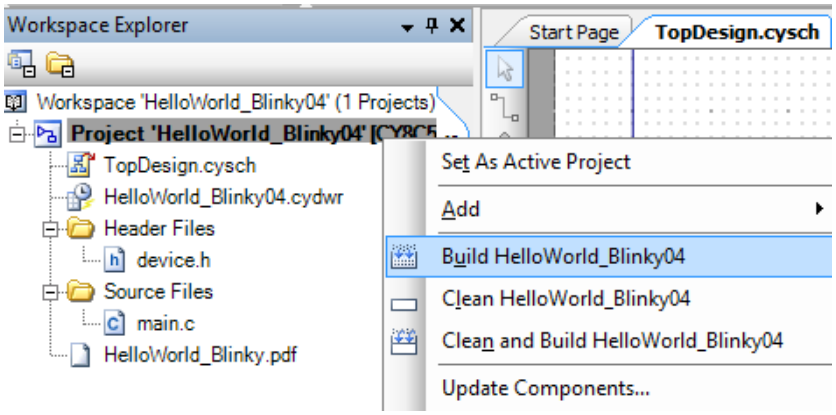
8. We need will configure what physical port on the board the Pin component on the TopDesign schematic represents. This is necessary because the virtual "P0_0" pin attached in the schematic only gets assigned a specific physical pin if you explicitly tell it to. This lets you connect your virtual schematic pins to nearly any physical port on the RPiSoC, allowing for very flexible designs. Navigate to the 'Workspace Explorer' pane on the left side of the PSoC Creator, and then double click the "'your_project_name'.cydwr" file.



9. Select the Port dropdown next to "P0_0" and select P12[0] I2C1:SCL. This means that the output of the PWM connected to pin "P0_0" will end up being output on P12.0 of the physical board. While you can assign the virtual "P0_0" pin to nearly any physical pin on the board, we choose P12[0] because it is attached to an actual LED on the RPiSoC.


*TopDesign.cysch
Bootloadabl...mple.cydwr
Alias
Name
Port
Pin
LED
P12[0] I2C1:SCL
53
P5[2]
P5[3]
P5[4]
P5[5]
P5[6]
P5[7]
P6[0]
P6[1]
P6[2]
P6[3]
P6[4]
P6[5]
P6[6]
P6[7]
P12[0] I2C1:SCL
P12[1] I2C1:SDA

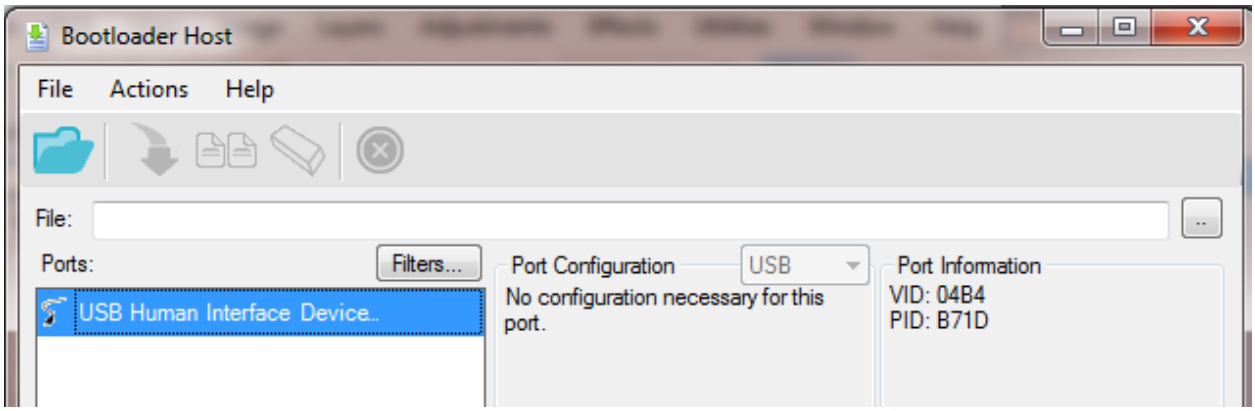
10. Now we are ready to build the project. In the “Workspace Explorer” pane Right-click on “Project ‘your_project_name’” and then left click on “Build ‘your_project_name’”. PSoC Creator will now compile your project. Wait until it says “Ready” in the bottom left hand corner of the PSoC Creator window.



11. If you haven’t already, go ahead and plug your RPiSoC board into the computer with a micro-USB cable. The red power LED should illuminate on the board. If it does not, make sure the jumper next to the USB port on the board is in the correct position. Drivers should automatically install, and then windows should tell you that your device is ready to use.

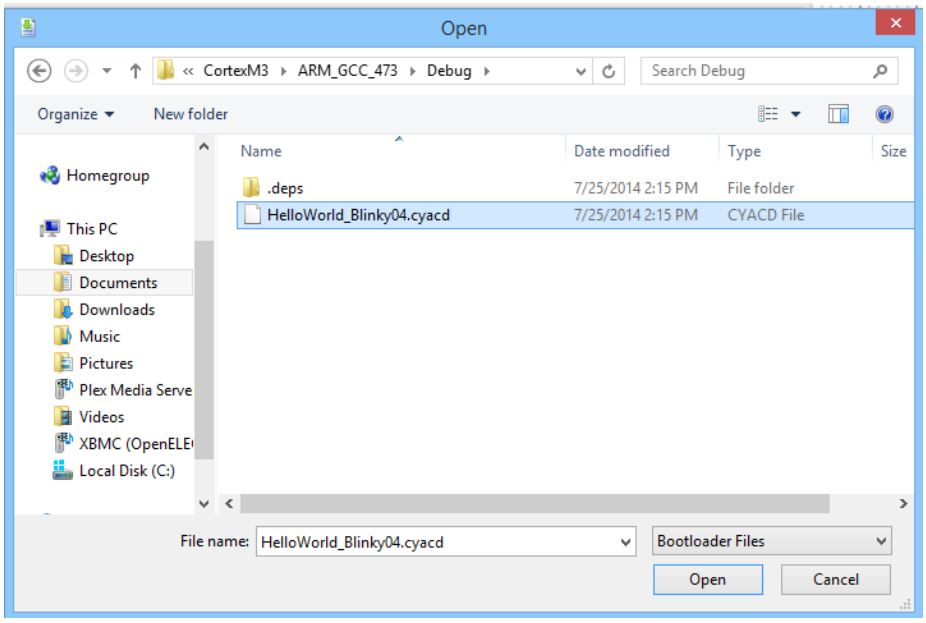
12. On the RPiSoC board, move the slide-switch into the “PROG” position and then press and release the “RST” button.

13. Now we are ready to program the board. First click on “Tools” in the top of PSoC Creator, and then click “Bootloader Host”. The RPiSoC will show up in the Ports section of Bootloader Host as “USB Human Interface Device”



14. Now you need to point the Bootloader Host tool to the project file you created. Click on the ".." button to the right of the file text box. You need to open the "'your_project_name'.cyacd" file. The path should be:

    "your_project_folder"\"your_project_name".cydsn\CortexM3\ARM_GCC_473\Debug\"your_project_name".cyacd



15. Finally, you can program the board by clicking the blue downward facing arrow near the top of the screen, or hitting F5. The yellow LED on the board should now be blinking. If you want to program your board again, simply restart it with the slide-switch in the PROG position, which will bring it into programming mode. To run your program again, simply restart it with the slide-switch in the RUN position and wait 2 seconds.

8 Appendices

8.1 Appendix A: Learning Resources

1. Cypress has created a wealth of videos and tutorials on how to use PSoC Creator, and the PSoC and general. Check them out here.

2. There are dozens of example projects included in PSoC Creator that show you how to use various features of the PSoC. You can access them by clicking on “File” and then “Example project” and then filtering the Architecture option to “PSoC 5LP”. These can be great projects to use a base for more comprehensive ones. Keep in mind you must add the Bootloadable component and follow the steps covered in this user guide for programming in order to make use of these example projects.

3. If you are ever curious about how to use a PSoC Creator component, or what it even does, you can right click it and select “Open Datasheet” to view its full documentation.

4. All Embedit Electronics example projects will be posted on our github page. You can also find the RPiSoC schematic and board layout there. Feel free to modify them to fit your needs.

8.2 Appendix B: Tips and Tricks

1. A few specific analog features of the PSoC chip, specifically Opamps and IDACs, are tied to certain pins. These can be viewed in the pin configuration screen of the .cydwr project file.

2. If you are having a problem with your PSoC Creator project, feel free to search the Cypress forums, Knowledge Base, or your search engine of choice. The RPiSoC uses the PSoC 5LP chip, which has a large existing support base built around it. If you believe you have an RPiSoC specific bug, or just want to discuss your project we will be establishing a forum at EmbeditElectronics.com in the near future. In the meantime we can be contacted by email at: Embedit@embeditelectronics.com