



Programming Open-Storm Board

A guide to uploading open-storm firmware to a board.

Written By: Jacquelyn

The screenshot shows a Windows File Explorer window titled 'Perfect Cell' with the address bar showing the path: This PC > Documents > PSoC Creator > Perfect Cell. The main pane displays a single folder named 'perfect-cell' with a date modified of 9/17/2019 3:18 PM and a type of File folder. An overlaid terminal window titled 'MINGW64/c/Users/hail/Documents/PSoC Creator/Perfect Cell/perfect-cell' shows the following commands and output:

```
hail@DESKTOP-PINTUGV MINGW64 ~
$ cd ~/Documents/PSoC Creator/Perfect Cell/

hail@DESKTOP-PINTUGV MINGW64 ~/Documents/PSoC Creator/Perfect Cell
$ git clone https://github.com/bpwong-um/perfect-cell.git
Cloning into 'perfect-cell'...
remote: Enumerating objects: 1126, done.
remote: Total 1126 (delta 0), reused 0 (delta 0), pack-reused 1126
Receiving objects: 100% (1126/1126), 8.74 MiB | 22.61 MiB/s, done.
Resolving deltas: 100% (745/745), done.

hail@DESKTOP-PINTUGV MINGW64 ~/Documents/PSoC Creator/Perfect Cell
$ cd perfect-cell/

hail@DESKTOP-PINTUGV MINGW64 ~/Documents/PSoC Creator/Perfect Cell/perfect-cell
(master)
$ git checkout sd_logger
Switched to a new branch 'sd_logger'
Branch 'sd_logger' set up to track remote branch 'sd_logger' from 'origin'.

hail@DESKTOP-PINTUGV MINGW64 ~/Documents/PSoC Creator/Perfect Cell/perfect-cell
(sd_logger)
$
```



TOOLS:

- [PSoC Creator v 4.2 \(Software\)](#) (1)

- [Git Bash \(Software\)](#) (1)

This is a Command Line Interface (CLI).

- [Windows Computer](#) (1)

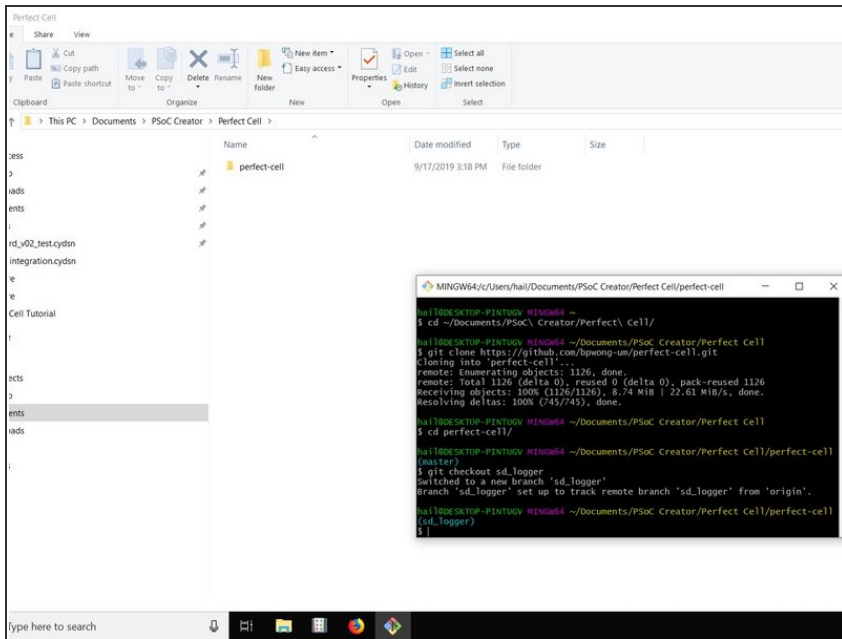
- [USB Programmer](#) (1)



PARTS:

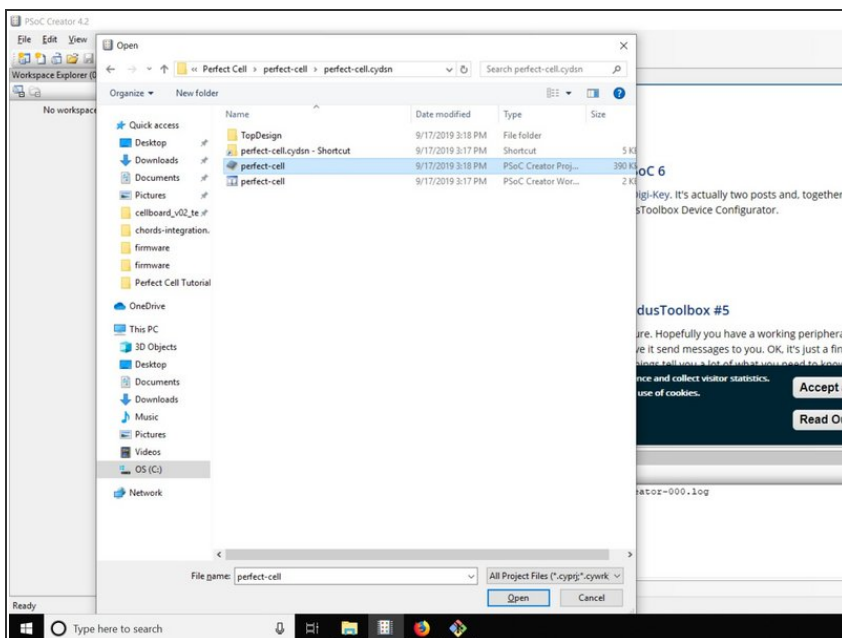
- [Node Board](#) (1)

Step 1 — Clone the most recent version of the firmware



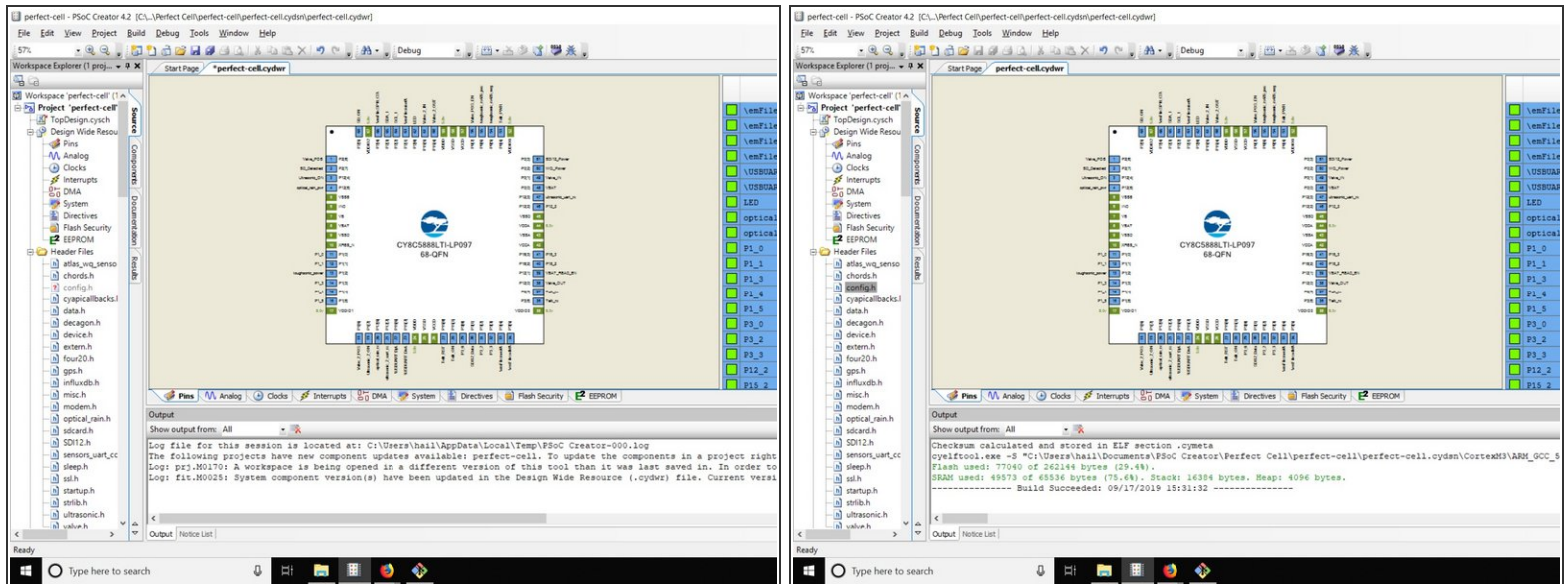
- Open Git Bash and type the following commands:
 - `cd ~/Documents`
 - `mkdir Perfect_Cell`
 - `cd ~/Documents/Perfect_Cell`
 - `git clone https://github.com/bpwong-un/perfect-cell...`
 - `cd perfect-cell`
 - `git checkout sd_logger`

Step 2 — Open file “perfect-cell.cyprj” in PSoC Creator



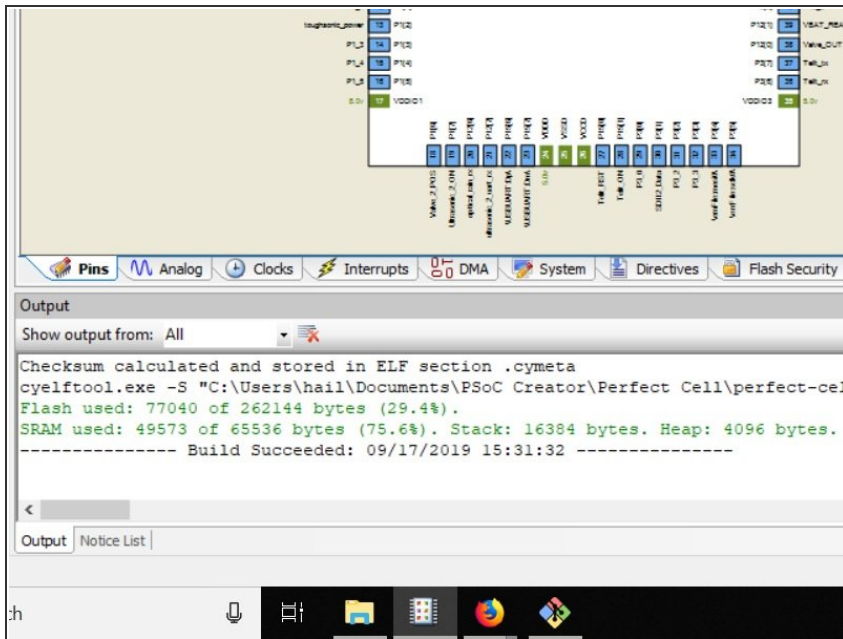
- You need to open the file using the Project/Workspace tab, not the File tab.
- It's file path should be: “Perfect-cell\perfect-cell.cydsn\perfect-cell.cyprj” (you might need to show “All files (*.*)”)

Step 3 — Find and copy the config.h file into the Perfect_Cell directory



- If you are using a lab computer, it should be on the Desktop somewhere. If you don't see it go find a grad student and ask.
- We are coming up with a better way of doing this. The config.h file contains sensitive information like node passwords, so we don't have it on our github repo, and we don't want to spread it around.
- You might have to add config.h file to filepath (right click "Headers" click add > add existing, click on config.h file)

Step 4 — Build the project



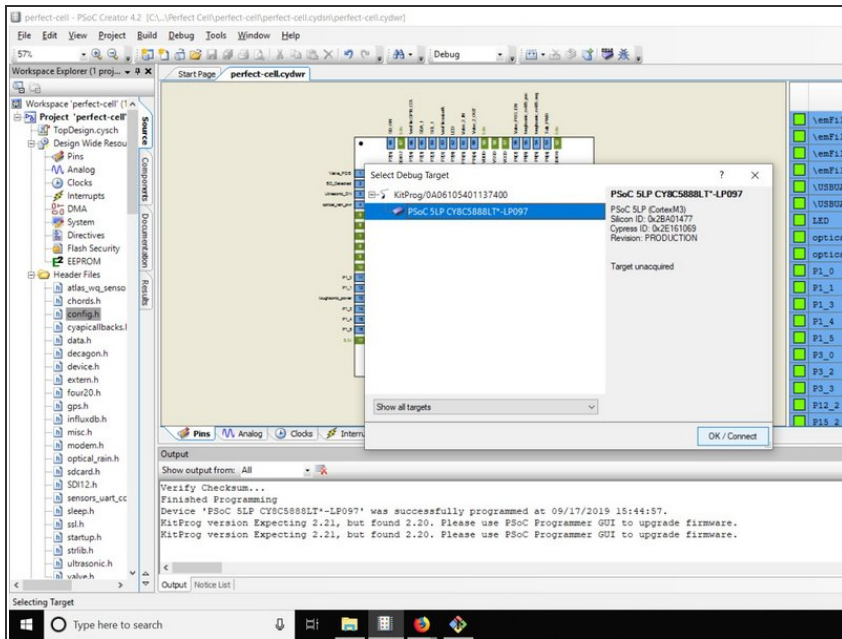
- Go to “Build” → “Build perfect-cell” (this’ll take a lil’ bit)
- If you get any “FLAG” errors, you probably have an old config.h file. That’s alright, you just need to add that variable into the config file code.
 - Eg, `#define V420_FLAG 0u`
- When your build is complete, you will see a “Build Succeeded!” message.

Step 5 — Add jumpers to the board



- Before you can program the board, make sure all the necessary pins have jumpers. Jumpers allow you to toggle between different board settings.
- The jumpers you need are:
 - JP1 - Power the board directly from the battery (not the voltage regulator) JP1 (the bottom two pins are jumped)
 - JP2 - top two pins - controls how we power the modem
 - ST and XB - jump all of the status LEDs

Step 6 — Program the Board



- Before you get any further, make sure the board has an SD card and a modem (with activated SIM card) attached. Make sure the modem is correctly connected to the board!
- Using the board programming cable, connect your board to your computer via USB port
- The pin end connects to the white port on the board.
- If you have a problem connecting, check that the programmer is using the correct port.
- Once you have a connection, debug the project.