Summary

Proboard 21J systems are Arduino[1] compatible circuit boards based on the Microchip SAMD21 microcontroller. The design allows breadboard pin access to all functions of the SAMD21 ARM Cortex processor and supports easy interfacing with external devices.

21Js are ideal to use in educational or hobby electronics projects, or can be used as components in precision battery powered devices such as test instruments. System functions include software controlled disconnection of power from the onboard RGB LED and power indicator LED while operating in low-power applications.

See the Programming section for details about how to program a 21J using Arduino sketches, or by using CircuitPython.

Features of the Proboard21J:

- ATSAMD21J18A microcontroller
- USB Flashing Format (UF2) bootloader preinstalled
- Quartz crystal timing, ±10 parts per million accuracy
- 10 pin Serial Wire Debug (SWD) connector
- SWD connections accessible via breadboard pins
- USB-C connection for device programming
- Onboard 4MB SPI flash memory for use as a mini hard disk while running CircuitPython
- An onboard addressable RGB LED
- Fully compatible with the Arduino[1] Integrated Development Environment (IDE)
- 20 digital I/O pins, 6 analog inputs, 12 PWM outputs
- Onboard +3.3 VDC voltage regulator accepts +5VDC to12VDC input
- 256KB flash memory - 8K used for bootloader
- Dimensions: 46.99mm x 12.19mm (1.85in x 0.48in)
- Weight: 4.7g

Contents

Figures

Tables

Note #1: The name “Arduino” is a trademark of the Arduino Company. Heron Circuits, LLC is not affiliated with the Arduino Company. “Proboard” is a trademark of Heron Circuits, LLC. Proboards are not made by the Arduino Company, but are compatible with the Arduino Integrated Development Environment (IDE).

1. Pinouts

Figures 1-3 are pinout diagrams for P1 (36 header pin plug), J1 (USB-C jack) and J2 (SWD jack).

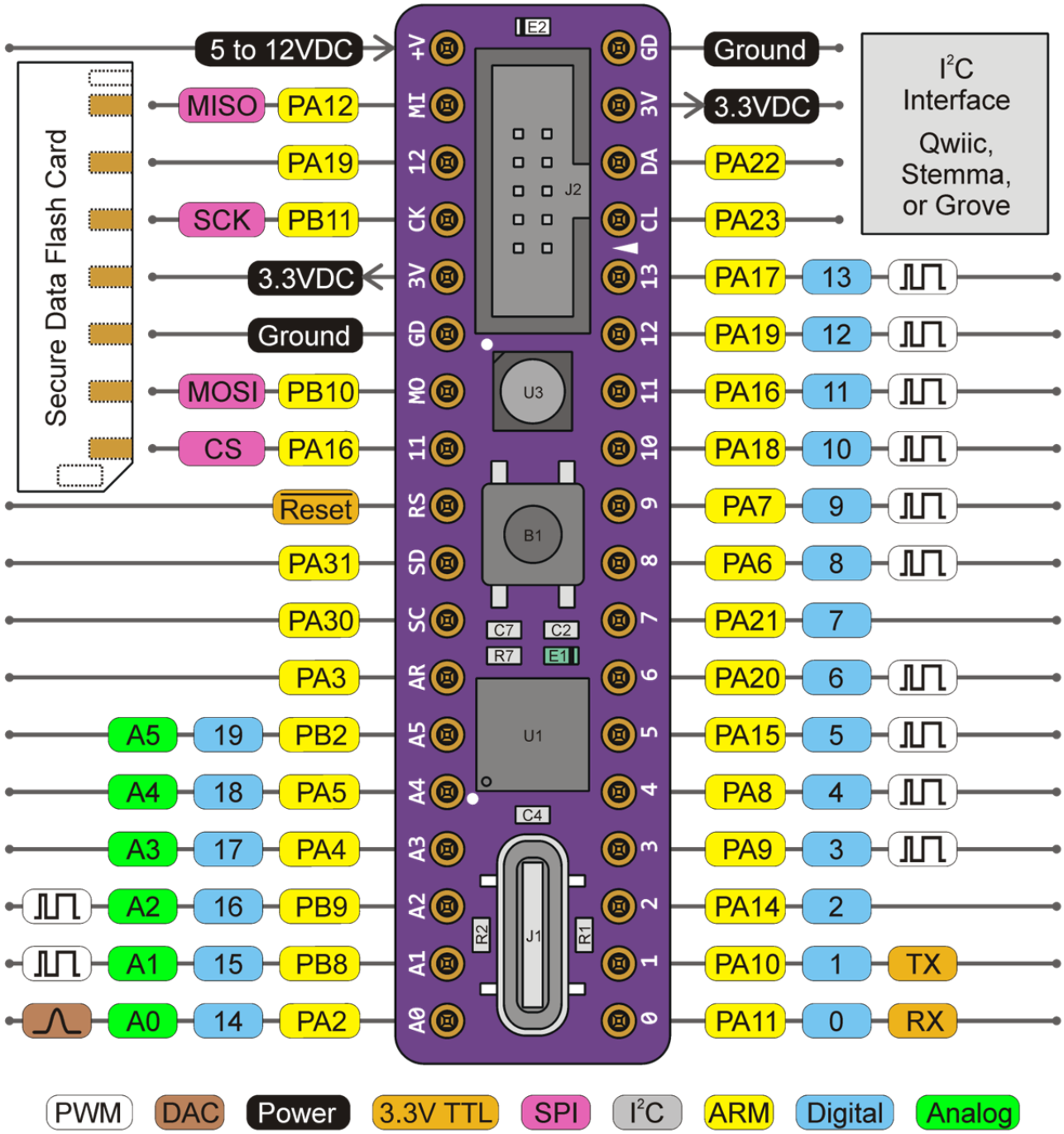

Figure 1 P1 Header Pins

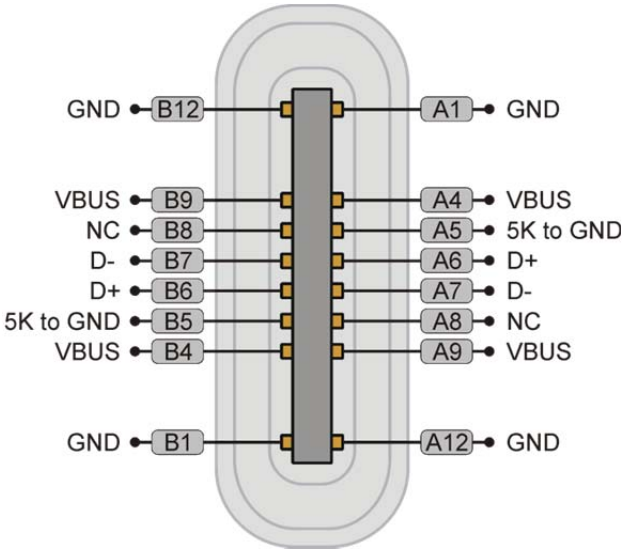

Figure 2 J1 USB-C Interface

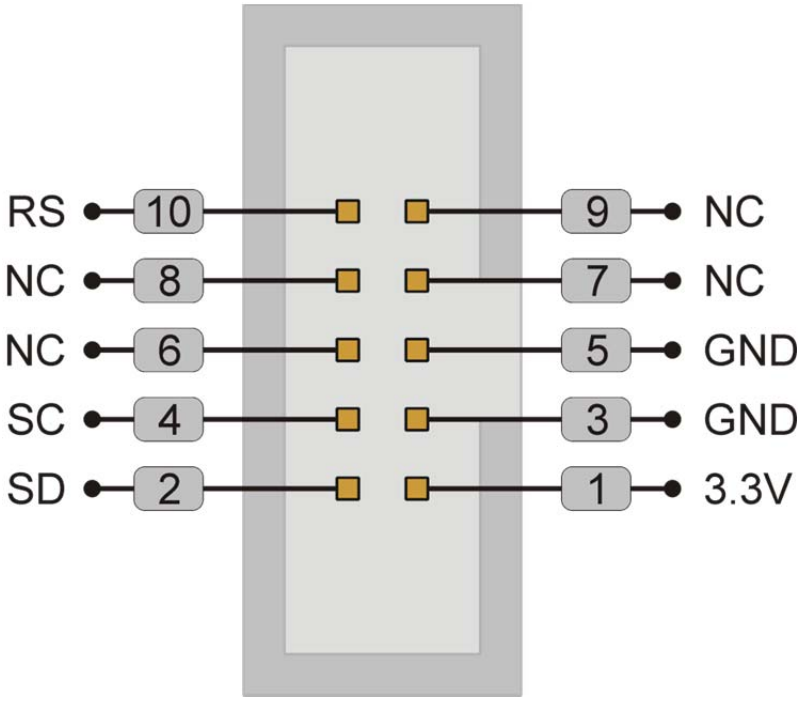

Figure 3 J2 SWD Interface

2. Schematic

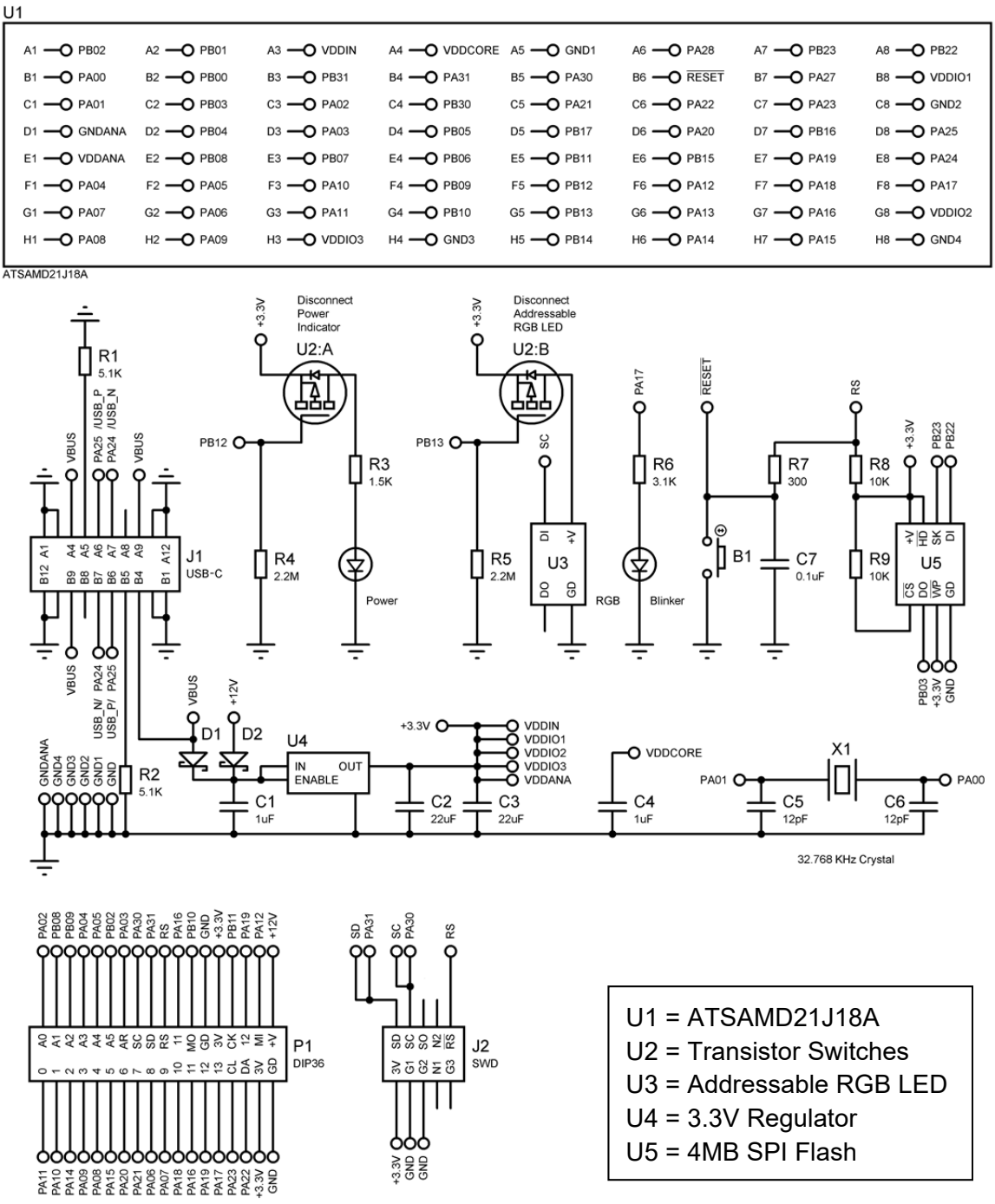

Figure 4 Proboard 21J Schematic

3. Signal Descriptions

Table 1 Header Pins

Pin Name	Description
+V	Input into the onboard voltage regulator, +5VDC to+12VDC
3V	Regulated +3.3VDC output from the onboard voltage regulator
GD	Ground pin
0	Digital Input / Output (I/O) pin, or receive serial data (RX)
1	Digital I/O pin, or transmit serial data (TX)
2 to 13	Digital I/O pins
CL	Serial Clock (SCL) pin for the Inter-integrated Circuit (I^2C) interface
DA	Serial Data (SDA) pin for the I^2C interface
A0 to A5	Combination analog or digital I/O pins, also known as digital pins 14 to 19
AR	Analog Reference
SC	Serial Wire (SW) interface Clock (CLK) pin (SWCLK)
SD	Serial Wire (SW) interface Data Input / Output (DIO) pin (SWDIO)
RS	External input to /RESET
Others	A convenient interface for breadboard connecting a Secure Data (SD) flash card

4. Powering Proboard 21J

Users have three options for supplying power.

- Connect the positive terminal of a +5VDC to +12VDC power supply to the pin marked +V, and connect the negative terminal of the power supply to either of the GD connections.
- Connect a regulated +3.3VDC power source to either of the pins marked 3V and ground. The output of the Proboard's onboard voltage regulator is protected by an internal output blocking diode. An externally applied voltage will not feed current back into the unused onboard voltage regulator. Just be certain the external source is a regulated supply that does not exceed +3.3VDC.
- The easiest option is to plug one end of a USB-C to USB-A data cable into the onboard USB-C jack, and plug the other end into the USB-A port of a PC. The Proboard 21J onboard voltage regulator will convert the +5V USB bus power into +3.3V for powering the SAMD processor.

For pins other than +V, avoid connecting any of the pins to voltages greater than +3.3VDC!

5. Connecting an SD Card

Use an SD card to micro SD adapter as shown in Figure 5, below. This works well for data logging. See examples in the Programming section for how to write data to PC formatted memory cards.

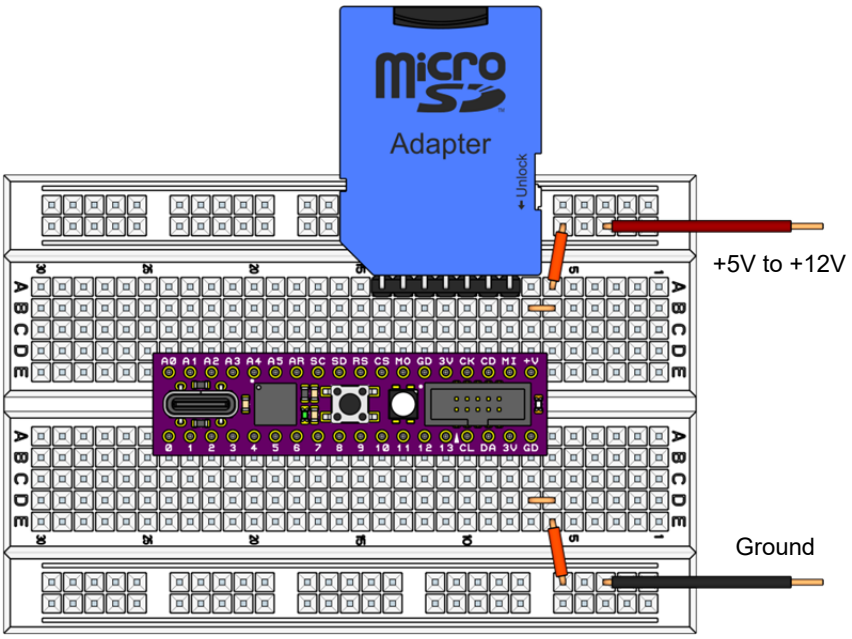

Figure 5 Installing a micro SD Card Adapter

Seven of the pads on the back of an SD card are spaced at 2.50 mm intervals. This is similar to the 0.1 inch (2.54 mm) pin spacing for a standard breadboard, so users can install an SD card as a memory module by soldering seven pins with 0.1 inch pin spacing to the back of the card. See the positioning in Figure 6. Note that pads 7 and 8 must not be shorted. Be careful not to plug any pin into the +V connection. Avoid plugging the module in backwards when installing it into the breadboard as shown in Figure 5.

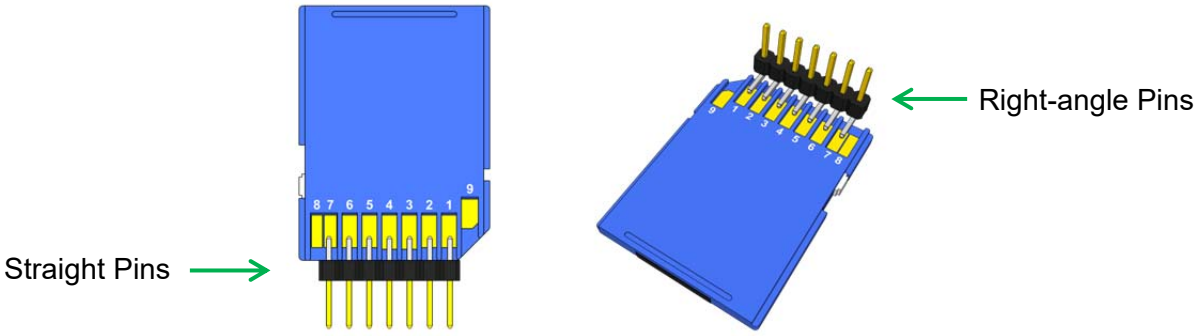

Figure 6 Pins for a micro SD Card Adapter

6. Programming

Navigate to https://www.arduino.cc/en/main/software where you can download the Arduino IDE to your computer. It will be necessary to make a few modifications to the Arduino IDE configuration so it can support all of the features of a Proboard 21J. Setup is as follows:

First, install Arduino IDE version 1.8.12 or later onto an Internet connected computer.

The Arduino IDE software will need a Heron Circuits link added to its configuration. Select "**File>Preferences**." In the text box marked "**Additional Boards Manager URLs**" add this text:

https://files.heroncircuits.com/proboard-index/package_heroncircuits_index.json

If any other installed json files are already listed in the same text box, keep them installed. Separate each Uniform Resource Locator (URL) with a comma.

Select "**OK**."

Choose "**Tools>Board:>Boards Manager…**" In the "**Type**" drop-down menu, select "**All**." Feed "Heron Circuits" into the search box, and press "**Enter**."

Point to "**Heron Circuits SAMD Boards**." Click "**Install**" and then "**Close**" when finished.

Choose "**Tools>Board:>Heron Circuits Proboard 21J**

In a moment, you will choose "**Tools>Port**" for selecting the serial port. To identify the correct port number for the Proboard 21J, first unplug the Proboard 21J's USB data cable from the PC and note any serial devices already present on the "**Tools>Port**" list. Then, plug the Proboard 21J's USB cable back into the PC's USB port and double press the Proboard 21J's "/RESET" button. This makes certain the Proboard 21J is in bootloader mode. In later operations uploading Arduino sketches, always remember to double press the reset button on the Proboard 21J to establish a com link with the system in bootloader mode before uploading any new sketch.

With the cable reconnected, now look for the extra serial port that appears on the "**Tools>Port**" menu. The new option that appears will be the one for the Proboard 21J. Other than being new, it should also have "(Heron Circuits Proboard 21J)" written in as part of the name. Select it.

Open the "**File>Examples>01.Basics>Blink**" sketch to test the system. Upload the blink sketch into the Proboard 21J by choosing "**Sketch>Upload**." You will see a "**Done Uploading**" message appear in the IDE status bar. The blink sketch flashes a small white LED at the end of the board.

Please complete this part of the setup first. Confirm the Arduino IDE is fully installed before doing other operations such as updating the bootloader or installing CircuitPython.

7. Updating the UF2 Bootloader

Each Proboard 21J is manufactured with a USB Flashing Format (UFF or UF2) bootloader resident in the lowest 8K of the processor memory. When using the Arduino IDE or CircuitPython it should not be necessary to erase the UF2 bootloader, but it can be modified in two possible ways. First, by using nothing more than a Windows PC and a USB-C data cable, the existing bootloader can update itself to a later version, thus providing an easy path for updates.

In the second case, it is possible to use a hardware debug probe such as a Segger J-Link to erase the UF2 bootloader. This is done if using Microchip Studio software to compile and install programs designed to take up all memory of the processor – including the bit of memory normally populated by the bootloader. Memory in the SAMD21 processor is huge, and all of its functions are accessible while keeping a bootloader installed, so programming it without a bootloader is not covered in this datasheet. For users who need that ability, the Heron Circuits website will maintain files showing how to program a Proboard 21J in a variety of ways without a bootloader.

The preinstalled UF2 bootloader is version 3.12.0. It should perform all functions without ever needing to be replaced, but if required the newest update will always be here:

https://files.heroncircuits.com/archives/Newest_Bootloader_Proboard_21J.zip

In the file name, the under bar character (“_”) is between words. The zip file contains two formats, bin and uf2. Updating is accomplished by using the uf2 file. The process is astonishingly easy. Simply download and open the zip archive using any unzip program. Connect the Proboard 21J to the PC using a USB-C data cable. Double press the reset button to put the Proboard into bootloader mode. In Windows File Explorer, the Proboard 21J will appear as a new virtual disk named “PROBOOT.” Now, use the mouse to drag the uf2 file from the archive and drop it onto the new virtual disk name. Windows shows a progress bar indicating a file transfer. Proboard 21J reboots itself using the newest bootloader. That’s it. The old bootloader installs the new bootloader by using the Windows graphical user interface and nothing more than a data cable.

Before performing the update, you can check the current version number for the bootloader that is already installed on your Proboard 21J. Double press the reset button. When the PROBOOT disk name appears in Windows File Explorer, double click on the disk name. The file inside named “INFO_UF2.TXT” contains the newest version information. Files available inside the virtual disk are returned to their factory default configurations by the bootloader when it is rebooted, so don’t bother editing or deleting them. They are quite indestructible, like Captain Scarlet.

The bin file version of the bootloader is included in the zip archive in case Microchip Studio and a debug probe are used to completely erase an installed bootloader. In that case, Microchip Studio would need to use the bin file for installing a fresh bootloader back into a totally erased system.

8. Installing CircuitPython

Before installing CircuitPython, please read and perform the steps in the previous section for updating the bootloader. The steps for installing CircuitPython are quite similar and will be much easier to follow after first warming up with the bootloader update.

Now, in order to install CircuitPython, download the CircuitPython installation zip file:

https://files.heroncircuits.com/archives/Newest_CircuitPython_Proboard_21J.zip

Open the zip file using any unzip program. Connect the Proboard 21J to the PC using a USB-C data cable. Double press the /RESET button to put the Proboard into bootloader mode. In Windows File Explorer, the Proboard 21J will appear as a virtual disk named “PROBOOT.” Use the mouse to drag the uf2 file from the CircuitPython archive and drop it onto the virtual disk name. Windows shows a progress bar indicating a file transfer. CircuitPython starts running on the Proboard 21J.

While CircuitPython is running, it changes the name of the virtual disk seen in Windows File Explorer from “PROBOOT” to “CIRCUITPY.” The Proboard 21J has a Serial Peripheral Interface (SPI) flash memory chip onboard. Double clicking on the “CIRCUITPY” disk name opens a new virtual disk and shows files that are stored in the SPI memory chip.

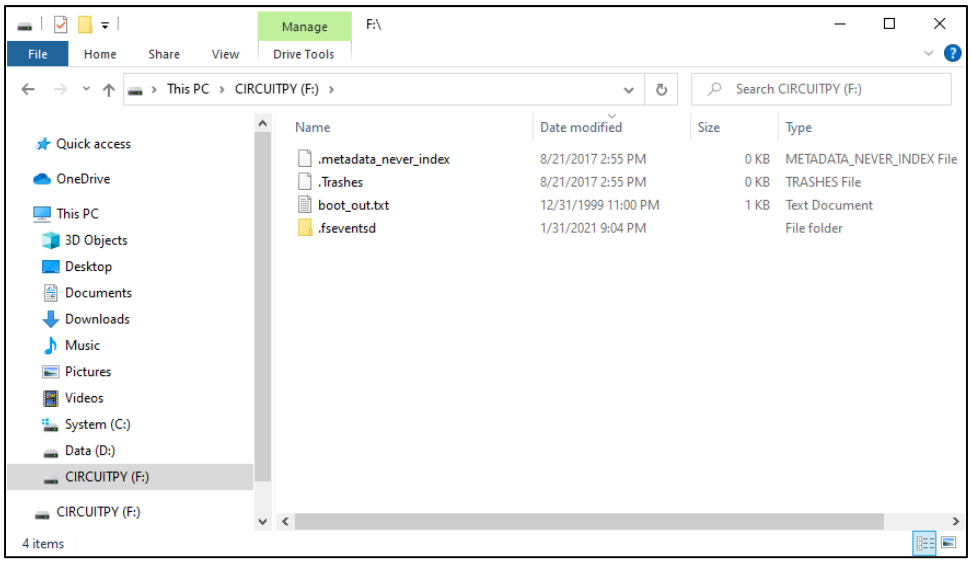

Figure 7 CIRCUITPY Initial Files

If CircuitPython was never installed previously, the content of the CIRCUITPY disk will be similar to that shown in Figure 7. It will have no Python scripts. The “boot_out.txt” file contains version information. Other files that begin with a dot “.” in their file names are used by the system to keep track of metadata, and should not be modified or deleted by the user. They are not indestructible.

https://files.heroncircuits.com/archives/CircuitPython_Flash_Chip_Content_Proboard_21J.zip

The file above contains Python script examples and a “lib” folder with libraries made open source by Adafruit Industries. Adafruit people know more about CircuitPython than any other beings in the universe, so be sure to read from their canon. For now, unzip the above content into the root directory of the CIRCUITPY virtual disk. The resulting root directory will look similar to the set of files shown below. Make sure the “lib” folder highlighted in blue is also copied to CIRCUITPY.

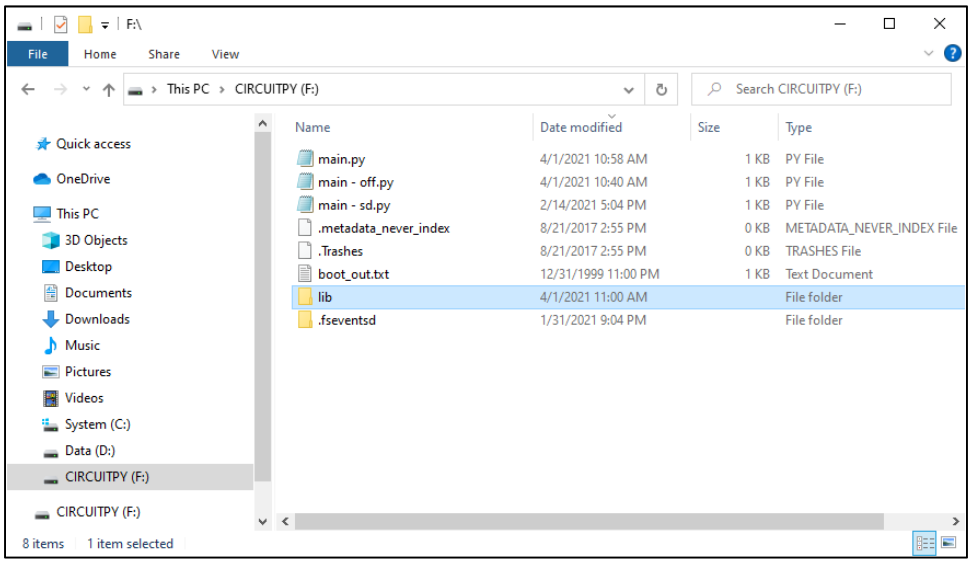

Figure 8 Example Python Scripts

CircuitPython is an interpreter. It immediately executes instructions in the “main.py” file any time the content of the CIRCUITPY virtual disk is changed. This has advantages over compiling C++ code.

The default “main.py” file is a copy of the “main - off.py” file. It shows how CircuitPython statements can turn off the three colors of an onboard Red Green Blue (RGB) Light Emitting Diode (LED). The code sets the RGB color brightness values to (0, 0, 0) which means “dark.” Get some cheap sun glasses, and change (0, 0, 0) to (0, 255, 0) if you wish to see the brightest green.

The file “main - sd.py” contains an example that shows how CircuitPython can write data onto an external secure data card as pictured in Figure 5. To switch between different examples, simply open a desired example file. Select all text. Copy the text into the text editor’s copy paste buffer. Close the example source file. Open the “main.py” script file. Select all text in the “main.py” file and hit “Paste” then “Save.” This replaces “main.py” without changing the source files.

Pressing the /RESET button once will cause CircuitPython to restart the “main.py” script.
Pressing /RESET twice will cause the bootloader to take over instead of CircuitPython.
Always press /RESET twice to enter bootloader mode before uploading Arduino IDE sketches.
Uploading Arduino sketches wipes out CircuitPython, every time, but it is easy to put back.

9. Low Power Applications

We solved a small problem we noticed when placing the SAMD21 processor into low power sleep modes. Various library functions are available for placing a SAMD21 into sleep modes where the processor can be awakened by external events such as button presses, or by the cycling of a watch dog timer. It is easy to find a variety of different third party code examples that work fine for placing the processor into such modes. As such, the processor consumes very little current while waiting between wake-up events.

The problem is that the onboard RGB LED includes a small integrated circuit inside the device which continues to consume about 1.0 mA of current – even if all three colors of the RGB LED are set for zero output. Translation; even in ultra-low-power standby modes, if operating on battery power, the system would kill batteries way too quickly.

The RGB pixel is useful when running CircuitPython code because the colors and the blinking of the RGB pixel have special meanings that help with code debugging. We did not want to get rid of it, or replace it with a very tiny RGB pixel as a compromise.

The solution is visible in the schematic in Figure 4 (on page 4). The hardware of a Proboard 21J takes advantage of two extra pins that are not available on similar boards such as a Metro M0 or other devices which use a 48 pin version of the SAMD21 processor. Proboard 21J uses a 64 pin version of the same processor. Two switching transistors are wired into two extra pins of the processor, and allow for disconnecting current from the onboard LEDs prior to placing the processor into sleep modes.
Click the image in Figure 9 to download the example code.

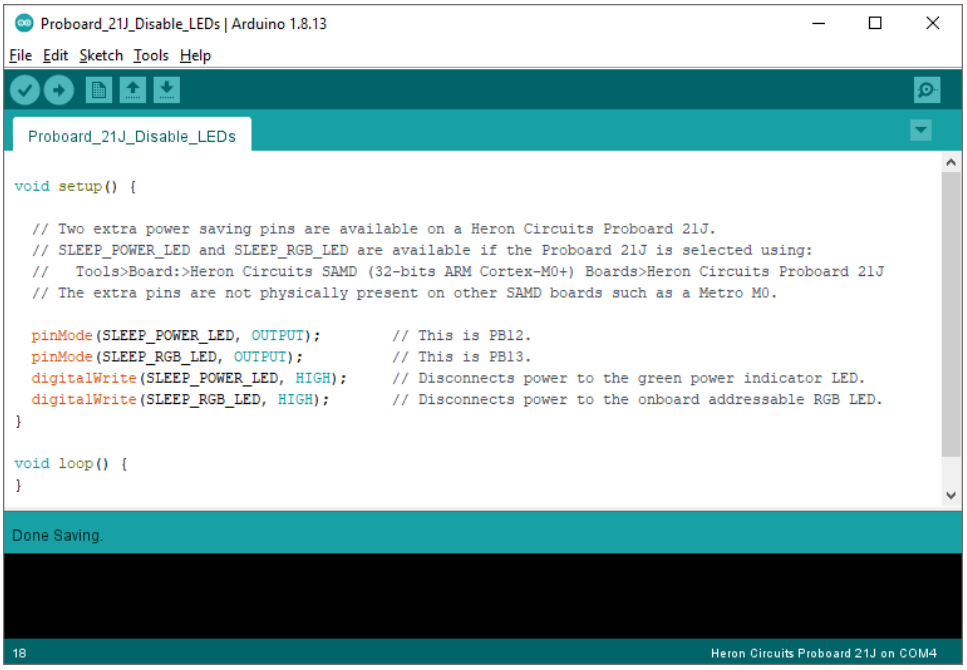

Figure 9 Disconnecting LED Power

10. Notice

IMPORTANT NOTICE – PLEASE READ CAREFULLY

Heron Circuits reserves the right to make changes, corrections, enhancements, modifications and improvements to Heron Circuits products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on Heron Circuits products before placing orders. Heron Circuits products are sold pursuant to Heron Circuits' terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of Heron Circuits products and Heron Circuits assumes no liability for application assistance or the design of the Purchaser's products.

No license, express or implied, to any intellectual property rights is granted by Heron Circuits herein.

Resale of Heron Circuits products with provisions different from the information set forth herein shall void any warranty granted by Heron Circuits for such product.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.