Summary

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

21Js are ideal for use in educational or hobby electronics projects, or can be used as components in precision battery powered devices such as test instruments. Software functions include an ability to disconnect power from the onboard power indicator LED and the addressable RGB LED when used in low-power battery operated 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, ±20 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. “Proboard” is a trademark of Heron Circuits, LLC.

Note #2: Heron Circuits, LLC is not affiliated with the Arduino Company, Adafruit, OSEPP or SparkFun, but we recommend their products.

1. Pinouts

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

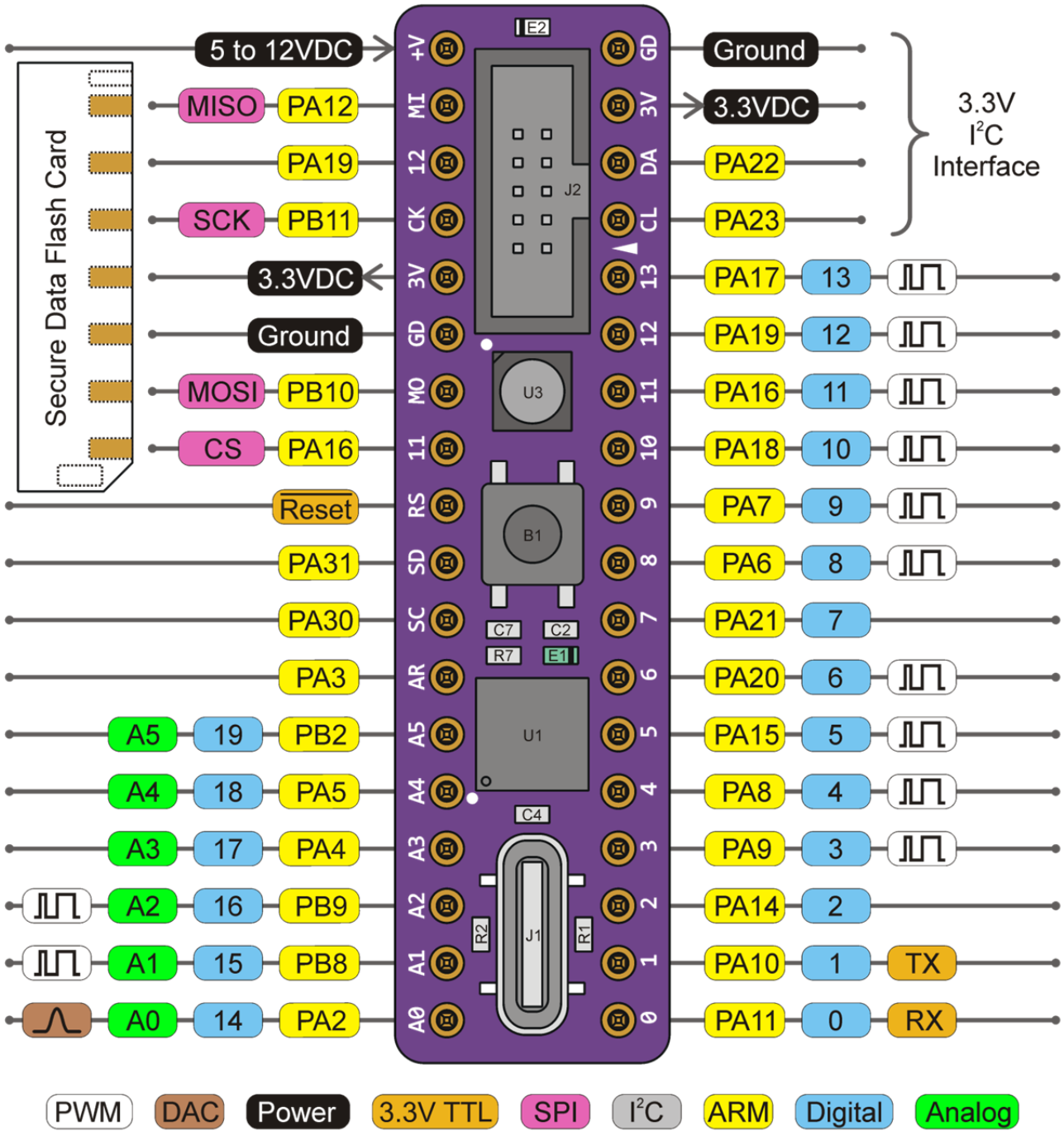

Figure 1 P1 Header Pins

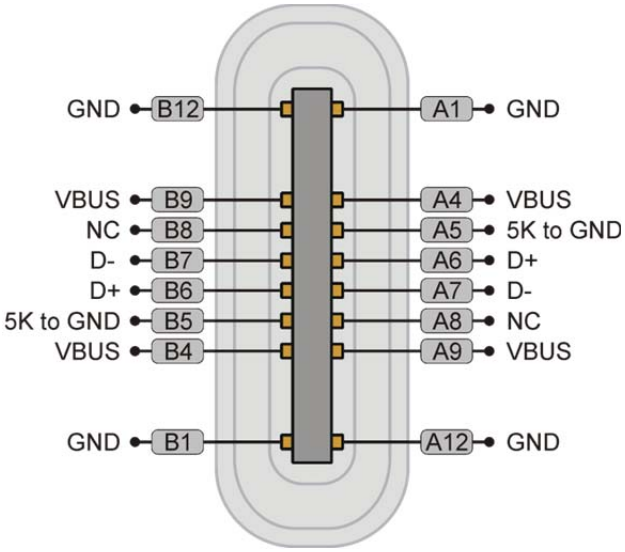

Figure 2 J1 USB-C Interface

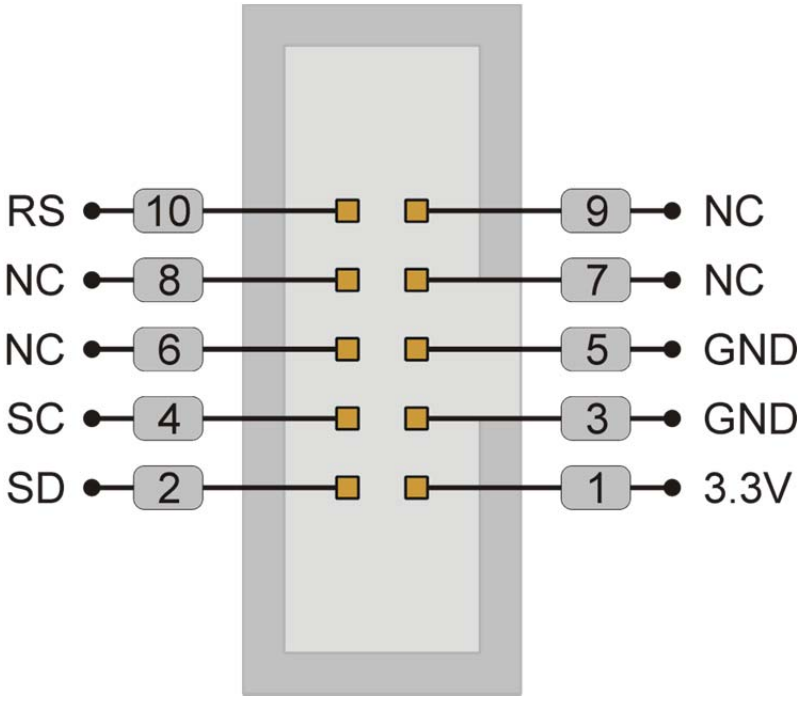

Figure 3 J2 SWD Interface

2. Schematic

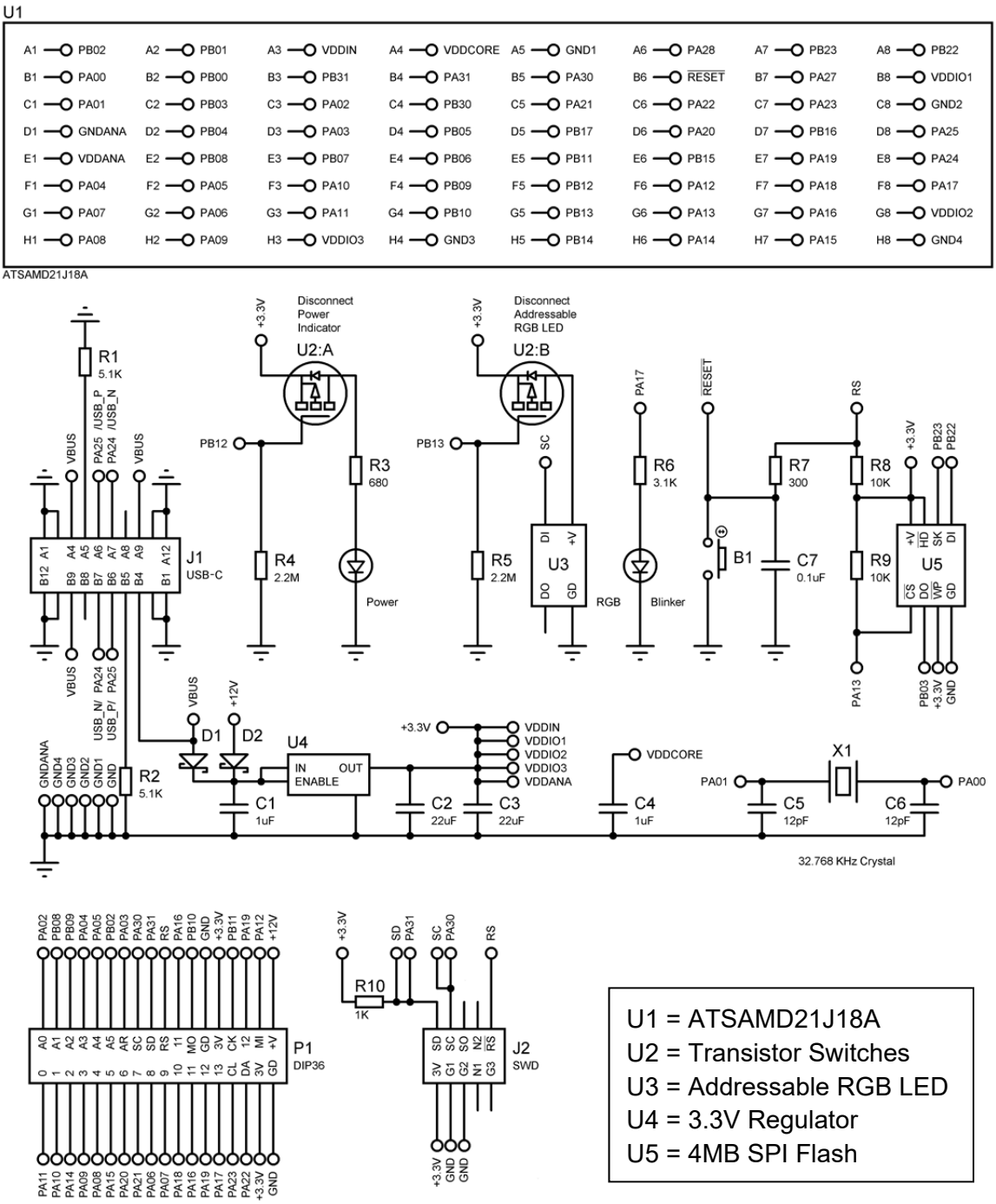

Figure 4 Proboard 21J Schematic

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 Debug (SWD) interface Clock (CLK) pin (SWCLK)
SD	Serial Wire Debug (SWD) interface Data Input / Output (DIO) pin (SWDIO)
RS	External input to /RESET
Others	The remaining pins form an interface for connecting an SD memory card. See Figure 1.

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 SD memory cards.

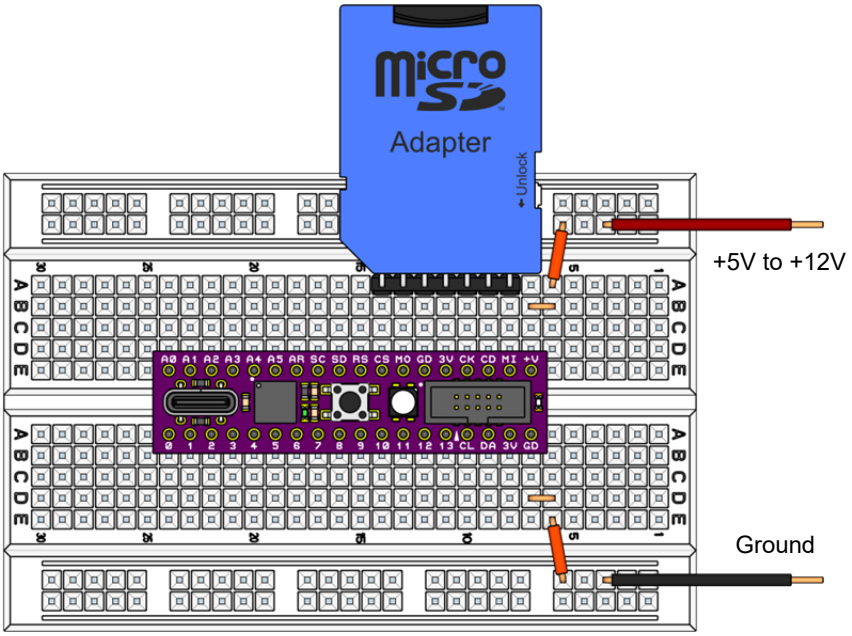

Figure 5 Installing a micro SD Card Adapter

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.

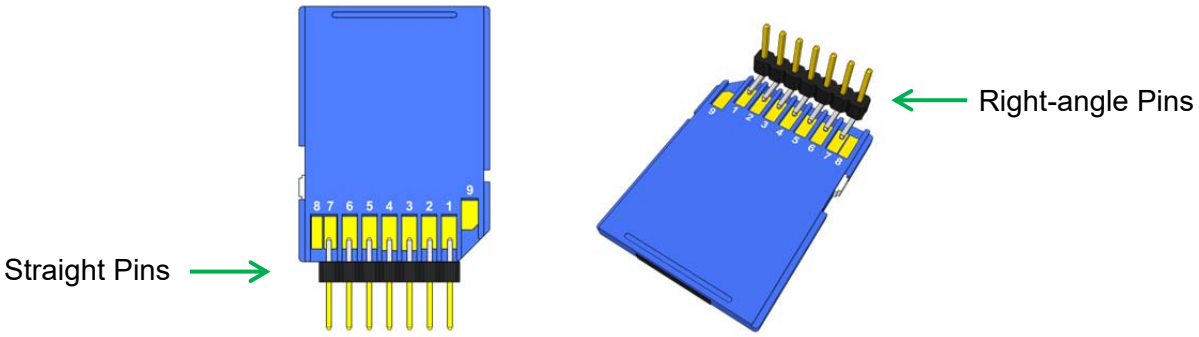

Figure 6 Pins for a micro SD Card Adapter

6. Programming

Navigate to https://www.arduino.cc/en/main/software where you can download the Arduino IDE to your computer. Choose the option pictured in Figure 7 if you need the Windows 10 EXE installer.

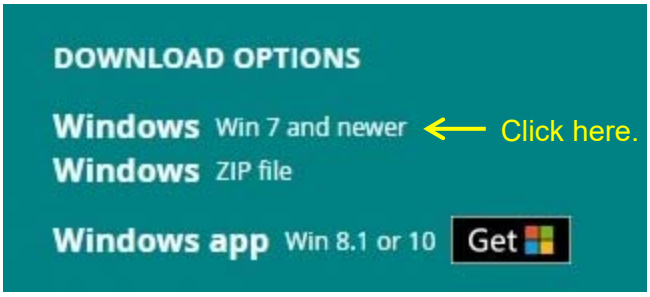

Figure 7 Downloading the Arduino IDE

You will need to provide information to the Arduino IDE so it can support using all features and functions of a SAMD processor. The order of the installation steps is important.

In a later step, you will need to install "Arduino SAMD Boards (32-bits ARM Cortex-M0+)."
In tests, we noticed that installing Arduino SAMD Boards can generate error messages that do not happen if a related Adafruit support package is installed first.

So, install the Adafruit support package first. Then, install the Arduino SAMD Boards package. Then, lastly, install the SAMD board files from Heron Circuits. The installation can sound complicated, but it is short and easy. Do and check off each step as listed here.

First, the Arduino IDE needs two links added to its configuration. Select "**File>Preferences**."
In the text box marked "Additional Boards Manager URLs" add the following text:

> https://adafruit.github.io/arduino-board-index/package_adafruit_index.json,
> https://files.heroncircuits.com/proboard-index/package_heroncircuits_index.json

If you already have other .json files linked in this same text box, keep them installed. Just remember to 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 "Adafruit SAMD" into the search box. Hits will start appearing before you type all of it.

Type all of it. You will see two hits. Begin with the second item on the list.

Point to "**Adafruit SAMD Boards**." Click "**Install**."
While this installation runs, the Arduino IDE can display a warning that it ignored a script.
Ignore the warning.

If “Adafruit SAMD Boards” is not listed, check your firewall and make sure your firewall allows the Arduino IDE full access to the internet connection so the Boards Manager can succeed.

The progress bar moves slowly, but this is normal. After the Adafruit SAMD Boards installation is done, next point to “**Arduino SAMD Boards (32-bits ARM Cortex-M0+)**.” Click “**Install**.” During installation, depending on preferences set in each Windows computer, the Windows operating system can interrupt several times asking for reassurances that it is OK to make changes. Click yes each time until you see a message confirming that the installation is finished. Do not click “**Close**” yet.

Next, feed "Heron Circuits" into the Boards Manager search box, and press “**Enter**.”

Point to “**Heron Circuits SAMD Boards**” and click “**Install.**”

If you also have a Proboard PB system based on the ATmega328PB microcontroller, you can optionally install Heron Circuits AVR boards support here.

Select “**Close**” when finished. Now, the Arduino IDE has new menu items available.

For using a Proboard 21J, select “**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 21J is in bootloader mode. In later operations while uploading Arduino sketches, always remember to double press the reset button on the 21J to place it into bootloader mode before uploading any new sketch. This makes the com ports match up automatically.

With the USB cable reconnected, 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 beside it as part of the name. Select it.

Open the “**File>Examples>01.Basics>Blink**” sketch to test your system. Upload Blink 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 and working 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 never 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 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 with 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 version 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 smart phone 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 version information. Files in the PROBOOT 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 also 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. The uf2 version is only useful when replacing a working bootloader that is already installed.

8. Installing CircuitPython

Before installing CircuitPython, please read and perform the steps in the previous section for updating the bootloader. Installing CircuitPython is quite similar and will be much easier to accomplish after first warming up with the bootloader update. In order to install CircuitPython, download the CircuitPython installation zip file at the following link:

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

Open this zip file using any unzip program. Connect the Proboard 21J to the PC using a USB-C data cable. Double press the /RESET button of the Proboard to put it into bootloader mode. In Windows File Explorer, the Proboard 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. 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 non-volatile memory chip onboard just for storing CircuitPython scripts and related library files. Double clicking on the “CIRCUITPY” disk name opens a new virtual disk showing files stored in the memory chip.

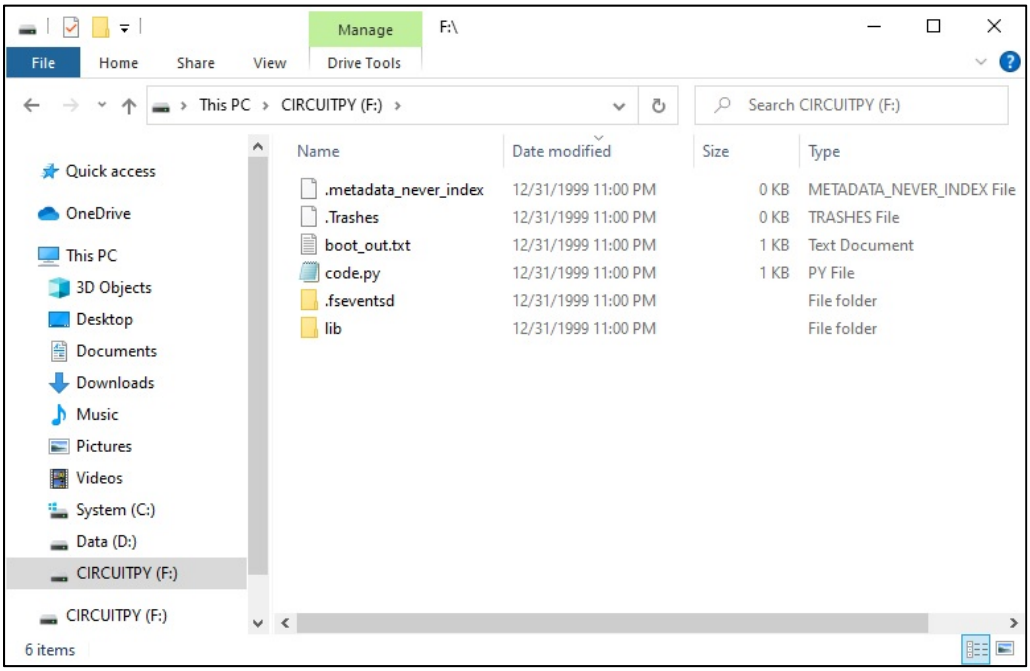

Figure 8 CIRCUITPY Initial Files

If CircuitPython was never previously installed, it will set up the CIRCUITPY virtual disk’s content as shown in Figure 8. Note the disk has a single “code.py” example script. The file “boot_out.txt” contains version information. Files that start with a dot “.” in their names are used by the system to keep track of metadata, and should not be modified or deleted. The “lib” folder is empty at this point.

In the zip archive, along with the CircuitPython UF2 installer, the files include extra Python script examples and a “lib” folder full of libraries created by Adafruit Industries. Delete the empty “lib” folder and then copy all zip file content except the UF2 file into CIRCUITPY virtual disk. The result will look like Figure 9. The full set of libraries needs about seven minutes to copy. The collection is slightly too big to fit into a typical 2MB Serial Peripheral Interface (SPI) memory chip, but this is not a problem for the Proboard 21J. It has a 4MB chip and can fit the entire set of libraries.

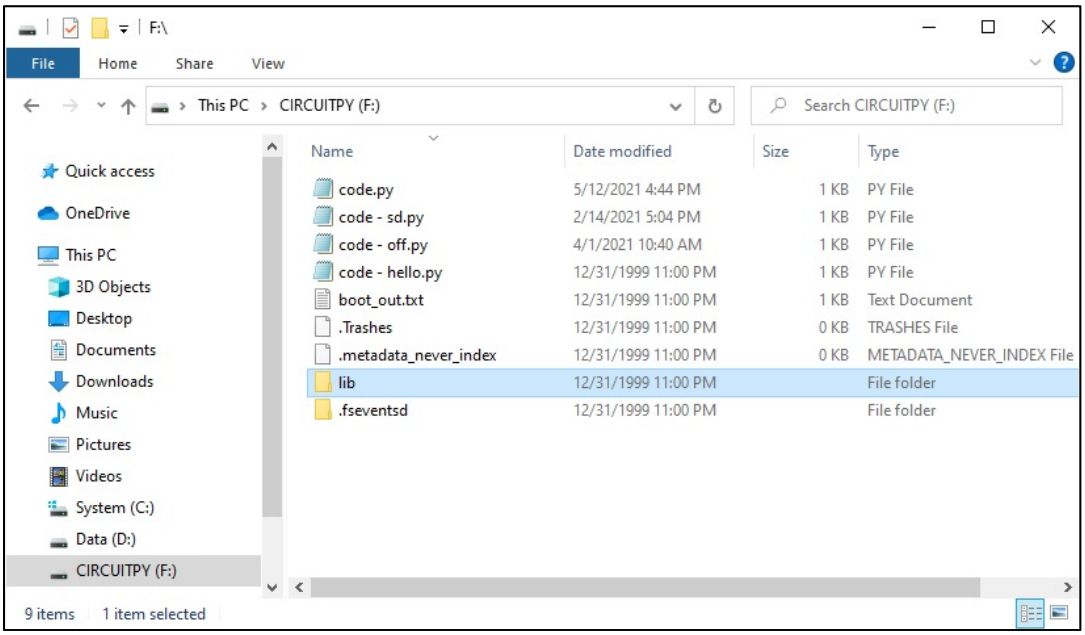

Figure 9 Example Python Scripts

CircuitPython is an interpreter. When it starts up it immediately executes the Python script instructions in the “code.py” file. To run any of the other example scripts, delete the existing code.py file, then copy and rename a copy of the example file to make a new code.py file. Any time the content of the CIRCUITPY virtual disk changes, CircuitPython reloads the code.py file and runs the Python program stored inside. There is no waiting for compiling. It runs NOW.

The example file “code - off.py” shows how CircuitPython can control the Red, Green, Blue (RGB) colors of an onboard three-color lamp. The code sets the RGB colors to (0, 0, 0) which means “dark.” Change (0, 0, 0) to (0, 0, 255) and hit “Save” to see the brightest of blue. Try (255, 0, 0) to see RED. Smaller numbers are less bright, so use numbers such as “1” to avoid dazzling.

The example file named “code - sd.py” contains code that shows how CircuitPython can write data onto an external secure data (SD) card as pictured in Figure 5. If you run this SD example with no data card connected, it will turn the multicolor LED red. If it finds a valid SD card, it will turn the light green. The SD card should be formatted as FAT32 with a single partition no bigger than 32GB.

Pressing the /RESET button once will cause CircuitPython to restart the “code.py” script. Pressing /RESET twice will cause the bootloader to take over instead of CircuitPython.

9. Low Power Applications

We solved a problem we noticed when placing the SAMD21 processor into low power sleep modes. Various library functions allow 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; it kills batteries way too quickly.

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

The solution is visible in the schematic in Figure 4 (on page 4). The hardware of the 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. The Proboard 21J uses a 64 pin version of the same processor. Two MOSFET switching transistors are connected to two extra processor pins. These allow disconnecting current from the onboard LEDs prior to placing the processor into sleep modes.
Click on the image in Figure 10 to download the example code.

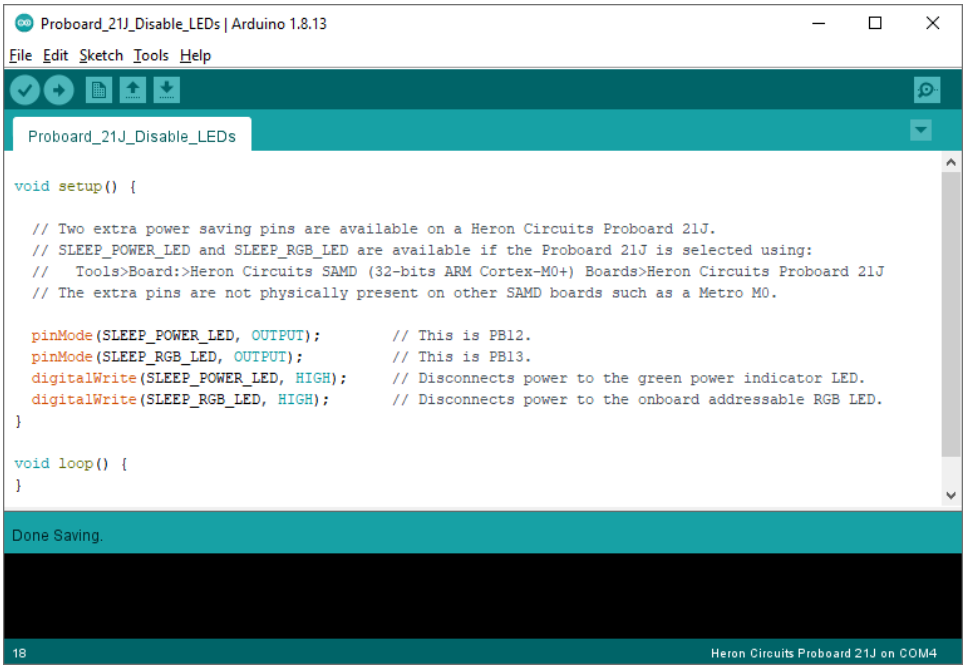

Figure 10 Disconnecting LED Power

10. Additional Resources

Related Products:

https://files.heroncircuits.com/documents/Proboard_328P_Datasheet.pdf

https://files.heroncircuits.com/documents/Proboard_PB_Datasheet.pdf

https://files.heroncircuits.com/documents/Timeport_Datasheet.pdf

https://files.heroncircuits.com/documents/Polaron_Datasheet.pdf

11. 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.