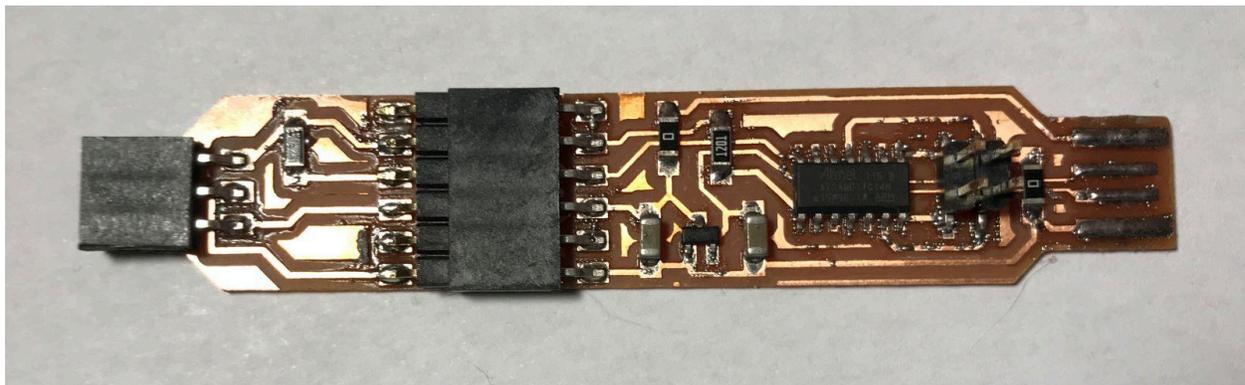


# Building a USB-to-Serial/UPDI Adapter

Jason Goodman, Wheaton College

These instructions describe how to build a USB-to-serial adapter that will communicate with a wide range of microcontrollers using the common UART serial protocol. It can also program Atmel microcontrollers such as the ATTiny 0- and 1-series using the UPDI protocol. It is a minor variation on Quentin Bolsee's [SAMD11C14 UART/UPDI programmer](#), and uses Quentin's [SAMD11C\\_serial](#) code. This version is designed so you can snap it in half to create a USB-to-Serial adapter, and a Serial-to-UPDI adapter, that can be used separately or joined together. You'll probably use this adapter for most of your electronics projects this semester.

This document describes the basic version of this adapter. There is also one with LEDs that light up to show when the boards are ready to communicate, and blink when data is sent. If you would prefer to make this version, the steps are almost the same, you just need to solder in a few more parts. More information about this ["blinking lights" version](#) can be found on my Github.



UPDI Adapter

Serial Output

Power

SAMD11C14A

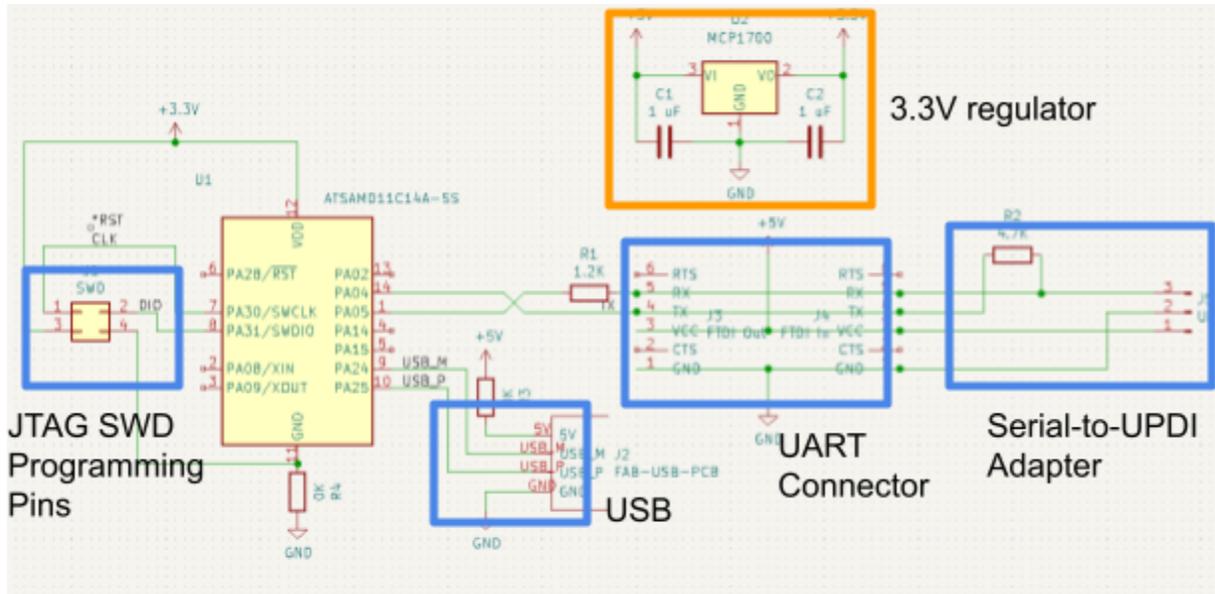
USB

## Parts List

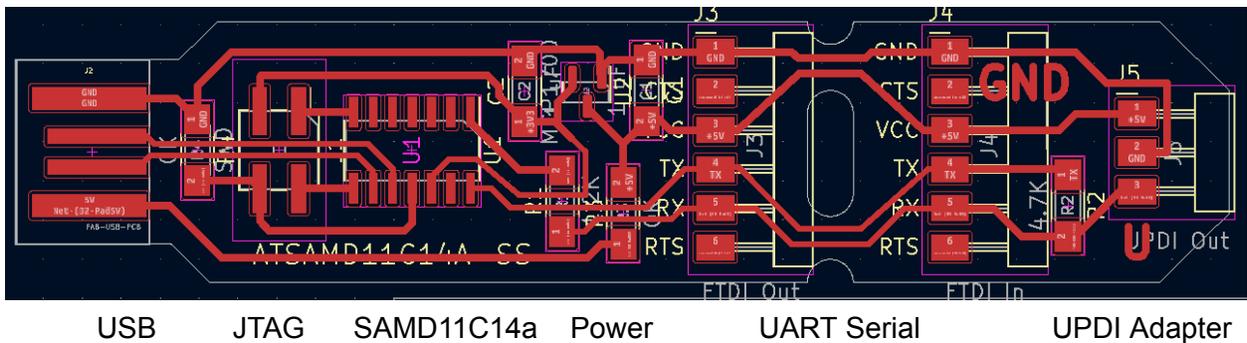
- [MCP1700T-3302E/TT LDO Voltage Regulator 5v->3.3v](#)
- [ATSAMD11C14A-SSNT SAMD microcontroller](#)
- [M20-8750242 2.54mm 2+2 DIL SMT header](#)
- [M20-8890645 2.54mm 6 way horizontal male SMT header](#)
- [M20-7910642R 2.54mm 6 way horizontal female SMT header Qty: 2](#)
- 1206 SMD resistor 4.7k
- 1206 SMD resistor 1.2k
- 1206 SMD zero-ohm SMD jumper
- 1206 SMD 1uF ceramic capacitors Qty: 2

## Schematic

The board is built in KiCAD: if you want to modify the board, project files can be found on [Github](#).



## The Board

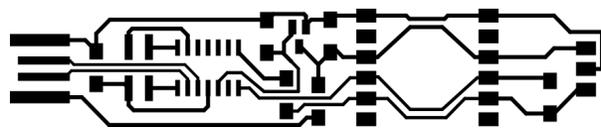


## PCB Fabrication

SVG files showing the board outline and traces files can be downloaded from the links below:



[outline.svg](#)



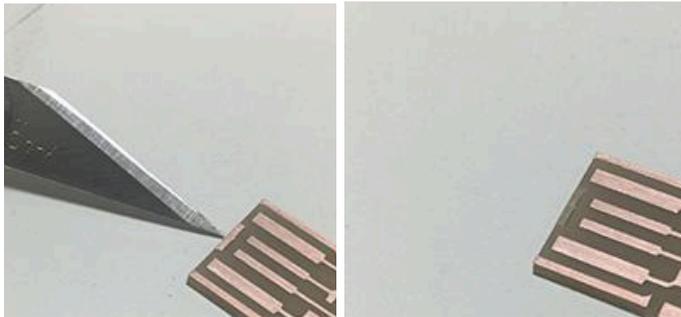
[traces.svg](#)

There are different processes for converting these files to .NC files and milling them on different machines. Here are tutorial documents for the PCB mills at Wheaton College:

- [PCB Design and Manufacturing Workflows](#)
- [Milling Custom Circuit Boards on the Roland Monofab SRM-20 with KiCAD and MODS](#)
- [Milling Custom Circuit Boards on the Sainsmart 3018 ProVer with MODS](#)

Use a 1/64" end mill to cut the traces. Use a 1/32" end mill to cut the board outline.

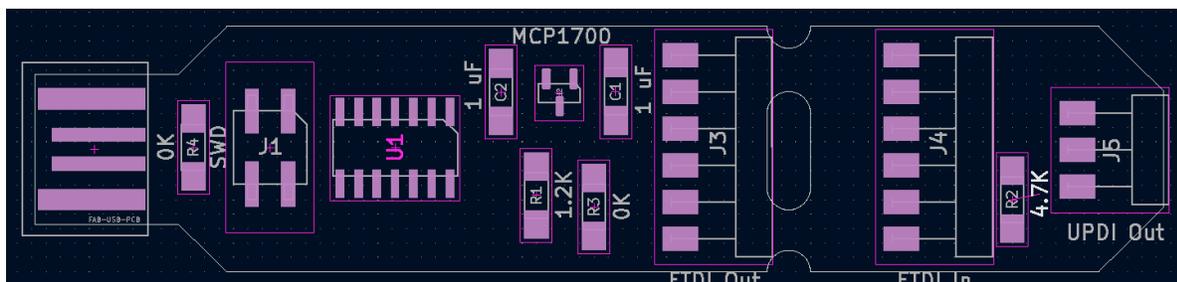
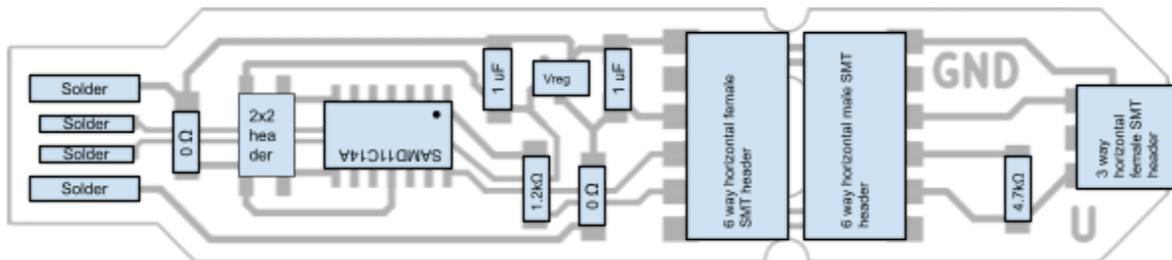
Depending on the number of offsets you milled, there might be a tiny bit of copper left at the edge of the board in front of the USB contacts. The extra copper should be removed with a knife. Only the copper in front of the pads needs to be removed; the copper left on the sides is fine.



(Photo from "[Building the FabTinyISP](#)")

## Soldering

Here is a map showing where all the components of the board should go.



- Note the ● indicating Pin 1 of the SAMD11C14A microcontroller. Look for a matching dot on the chip.
- The 3 way female SMT header can be created by cutting a 6 way female header in half with flush cutters.

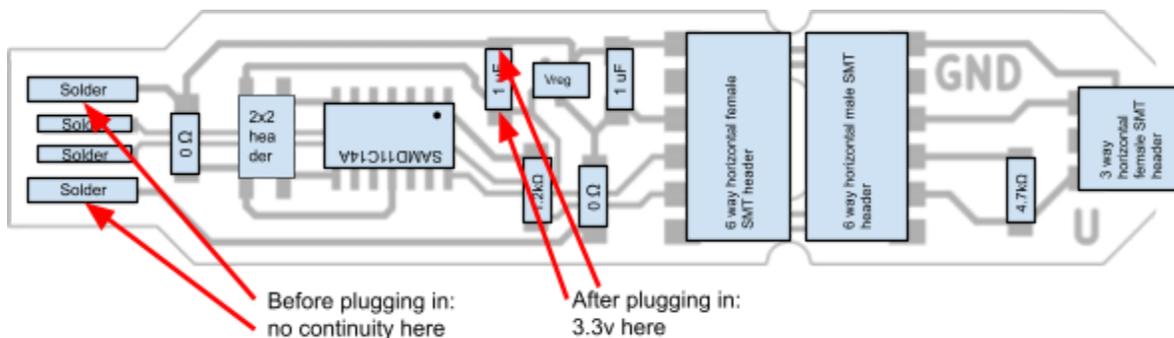
- Apply extra solder to the USB contacts to make them thick enough to fit securely into a USB slot.
- The 2x2 programming header has an annoying plastic peg on the bottom that keeps it from sitting nicely on the board. Cut that off with flush cutters.

## Back Plate

You may find that the USB port is a little too skinny to fit nicely into some USB ports. You can optionally 3-d print a [back plate](#) and glue or double-sided-tape it to the board to make it fit better.

## Smoke Test

**This step is not optional!** If you skip it, best case you may waste hours of time solving weird problems during programming. Worst case you may destroy your board! Before plugging your board in, use a multimeter to ensure that there is no continuity (no beep) between the outer contacts of the USB connector (VCC and GND). Then plug it into a USB adapter. (It might be best to plug it into a wall charger rather than your computer, unless you're sure you did a good job soldering!) Then check for 3.3 volts across capacitor C2. Make sure nothing smokes or overheats.



## Programming

Your board is designed to program other microcontroller boards, but it needs to be programmed itself first! The procedure described here is unique to the SAMD family of microcontrollers. See [Quentin's SAMD11C\\_serial repo](#) for more details.

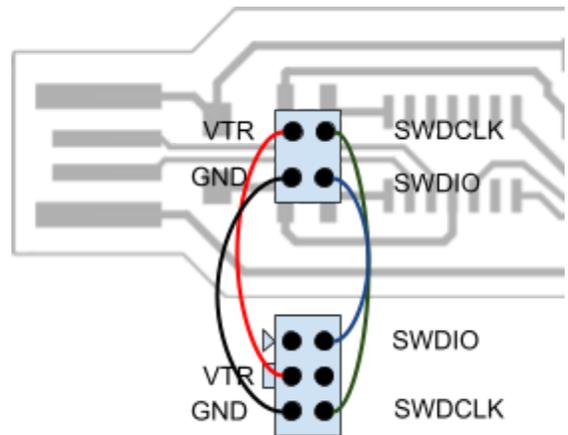
**Step 1:** Add support for the SAMD11C chip to the Arduino IDE. These steps only need to be done once on each computer:

- Start the Arduino IDE on a PC or Mac computer.
- Add the following URL to "Additional Boards Manager URLs" in the Arduino IDE Preferences:  
[https://raw.githubusercontent.com/qbolsee/ArduinoCore-fab-sam/master/json/package\\_Fab\\_SAM\\_index.json](https://raw.githubusercontent.com/qbolsee/ArduinoCore-fab-sam/master/json/package_Fab_SAM_index.json)
- In Tools / Board / Board Manager..., find "Fab SAM Core for Arduino" and click Install.

**Step 2:** Burn an Arduino bootloader onto the microcontroller. This step needs to be done once for each board to be programmed. It teaches the chip how to speak USB, so it can communicate with the Arduino IDE for further programming. This requires special hardware: an ATMEL Ice in-circuit programmer.



- Plug an ATMEL Ice in-circuit programmer into your PC (tested) or Mac (not tested but should work). Plug the gray patch cable into the “SAM” side of the programmer. Don’t force it, it only goes in one way.
- Use Dupont jumper wires to connect the programmer to the board you just made, as shown in the diagram at right. The rectangle in the diagram is the plastic protrusion on the programmer’s 6-pin socket, the triangle indicates “Pin 1”, and is marked on the socket.
- Plug the new board into a USB port, so the chip gets power. The red and green lights on the programmer should now both be on.
- In the Arduino IDE, do the following:
  - In Tools/Board, choose “Fab SAM Core for Arduino” / Generic SAMD11C14A.
  - Other settings to check in Tools:
    - USB config: CDC\_only
    - Serial config: TWO\_UART\_NO\_WIRE\_NO\_SPI
    - Bootloader size: 4KB\_BOOTLOADER
  - In Tools, choose Programmer: ATMEL-Ice.
  - In Tools, choose “Burn Bootloader”. You should see a message at the bottom of the window indicating that the chip was programmed and verified.
- Unplug the dupont wires, remove the board from the USB, and plug it back in. Bootloading was successful if a new communications port, probably labeled “Generic SAMD11C14A”, appears in the Tools / Port dropdown menu.



**Step 3:** Upload the Arduino serial code to the board. Your new board needs additional code so it can talk to other boards using the UART serial and UPDI sockets. This is written by Quentin in Arduino, a good language for beginner microcontroller programmers.

- Download Quentin’s SAMD11C\_serial Arduino code: [https://github.com/qbolsee/SAMD11C\\_serial/blob/main/SAMD11C\\_serial/SAMD11C\\_serial.ino](https://github.com/qbolsee/SAMD11C_serial/blob/main/SAMD11C_serial/SAMD11C_serial.ino)
- Keep all the settings from Step 2 in the Tools menu. In the Port menu, make sure the new “Generic SAMD11C14A” is selected.

- Open it, and click the “Upload” button. It should take a bit to compile and send to the new board. The result should look like the screenshot below.

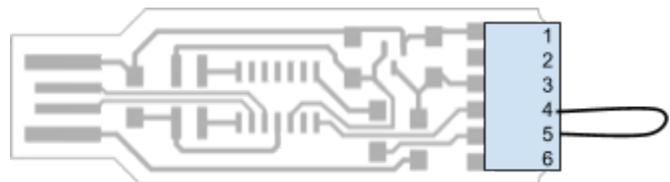
```
Sketch uses 10160 bytes (82%) of program storage space. Maximum is 12288
Atmel SMART device 0x10030006 found
Device      : ATSAM11C14A
Chip ID    : 10030006
Version    : v2.0 Aug 31 2021 14:06:56
Address    : 4096
Pages      : 192
Page Size  : 64 bytes
Total Size : 12KB
Planes     : 1
Lock Regions : 16
Locked     : none
Security   : false
Boot Flash : true
BOD        : true
BOR        : true
Erase flash
done in 2.277 seconds

Write 10512 bytes to flash (165 pages)
[=====] 100% (165/165 pages)
done in 26.131 seconds

Verify 10512 bytes of flash
[=====] 100% (165/165 pages)
Verify successful
done in 0.051 seconds
CPU reset.
```

## Testing

- Unplug your board from the USB.
- Use a pair of flush cutters to cut the PCB at the oval hole, separating the board into its two halves. Remove the UPDI adapter section.
- Plug it in again. It should show in the Port list in the Arduino IDE’s Tools menu as “Generic SAMD11C14A”. Select it.
- Open the Serial Monitor: this lets you use your new board to send text messages out to other microcontroller boards.
- Set the Baud Rate (communications speed) to 115200.
- Type some text in the box at the top and hit “Send”. Nothing should happen: your new board is sending your message out on the 6-pin serial UART header, but without another board connected to it, there’s no one there to answer!
- Connect a Dupont wire between pins 4 (RX) and 5 (TX) on the 6-pin header, like this:
- Now, type some more text and hit “Send”. You should see your text echoed back at you: your message goes out on the wire you added and comes right back in. Your new serial adapter is successfully talking to itself!



CONGRATULATIONS!

## It doesn’t work, now what?

Here are some suggestions for solving problems.

- Look for blobs of solder bridging between two pins, or two traces. Usually you can wipe these away with a hot soldering iron.
- Failed Smoke Test: Follow the GND and +5V traces around the outside of the board and over the 0-ohm resistors, looking for somewhere they might be connected by solder blobs, or not connected at all. The smoke test is not optional!

- Check for continuity between all adjacent pins on the SAMD microcontroller. There should be no continuity (no beep) between any of them. Also make sure there's no continuity between any of the nearby pins on each of the connectors: USB, 2x2 header pins, or the UART serial 6-pin header.
- Don't forget to scrape off the copper in front of the USB pads! (Last step of "PCB fabrication")
- Programming, Step 2: Bootloader won't install (Programming Step 2): make sure to plug the board into a USB to power it up for programming. Make sure the programmer shows both a red and green light. Double-check the traces connecting to the 2x2 header pins for short circuits: trace continuity from pins to chip.
- Bootloader won't install: Check all settings in the Arduino / Tools menu (see Programming, Step 2). Also make sure "ATMEL-ICE" is selected in Arduino / Tools / Programmer.
- Bootloader won't install and you're very sure your soldering is perfect: contact Prof. Goodman, I have a different bootloader installation procedure you can try, but it's more complicated.
- Bootloader installs, but when you plug in the board the computer says the USB device malfunctioned or not recognized: Double-check the traces leading from the inner two USB pads to the chip. These go under a lot of stuff, so make sure they're not shorted to each other or to the 2x2 header.
- Computer recognizes USB device when board is plugged in, but programming failed: check settings in Programming Step 3. Also, be patient, it can take 30 seconds for uploading to start. You should see "Verify Successful" in the red log text when it finishes.
- Programming works, but "loopback" test fails: check for soldering problems on the traces leading to pins 4 and 5 on the 6-pin serial connector.