

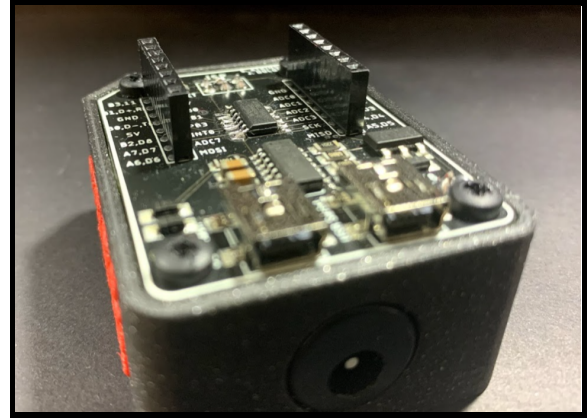
DDB V8+ Manual



Technical Overview

Goals:

Previous iterations of the DDB required the sparkfun pocket programmer to flash the onboard microcontroller with code. Simply, the DDB V8+ uses a USB port to flash code directly to the microcontroller. It achieves this using a bootloader that bit bangs USB. A USB compatible programmer then writes to the ATtiny84a over this USB to flash the ATtiny84a.

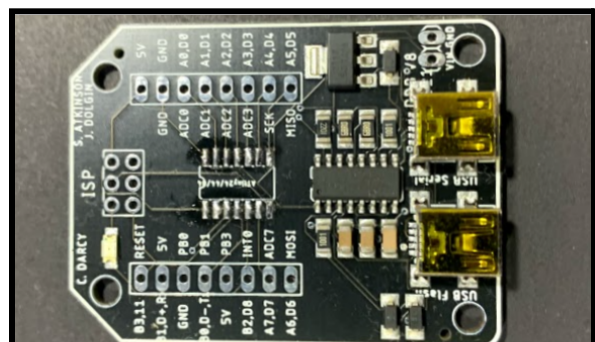


The DDB V8 also aims to make serial debugging easier and more simple. It does this using a serial to USB IC that converts serial to USB. This is a similar IC to the USB to serial adapter used in previous iterations of the DDB. The difference on the V8 is that it is on board.

Device Manufacturing

Assembled Device:

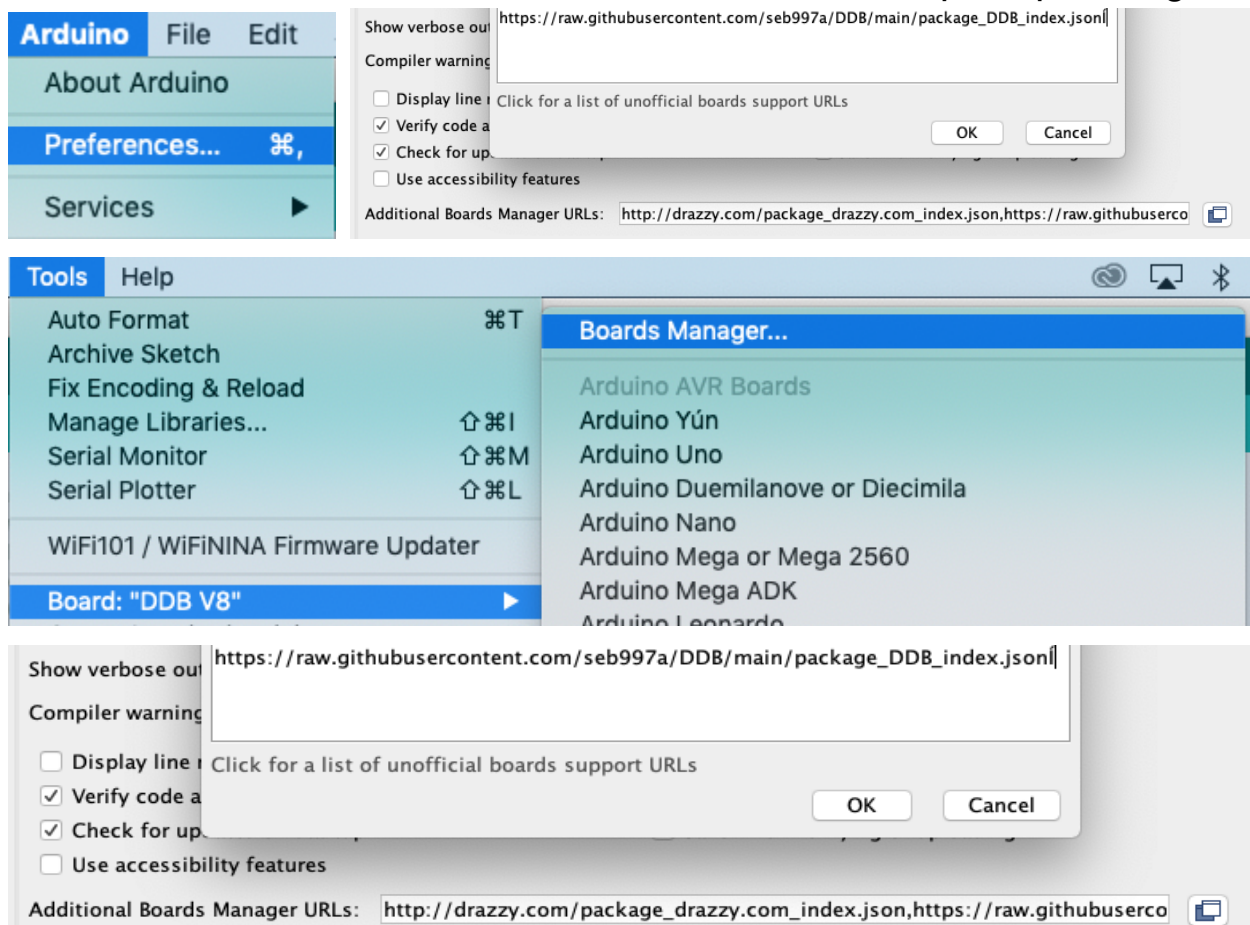
The board is manufactured at a PCB manufacturing facility. The PCB comes with everything but the ATtiny84a. Once delivered, the ATtiny84a is soldered on. Then, the bootloader is burnt making the device ready for USB uploading.



Arduino Mac Setup

To set up the DDB V8 on a mac, the first step is adding the DDB core to the arduino IDE. As shown below, in the Arduino IDE navigate to preferences and then to the additional boards manager urls. Add the following url to the boards manager URL:

https://raw.githubusercontent.com/seb997a/DDB/main/package_DDB_index.json. After adding the URL, navigate to tools, boards manager and install the DDB core. Once installed, the next step is uploading.



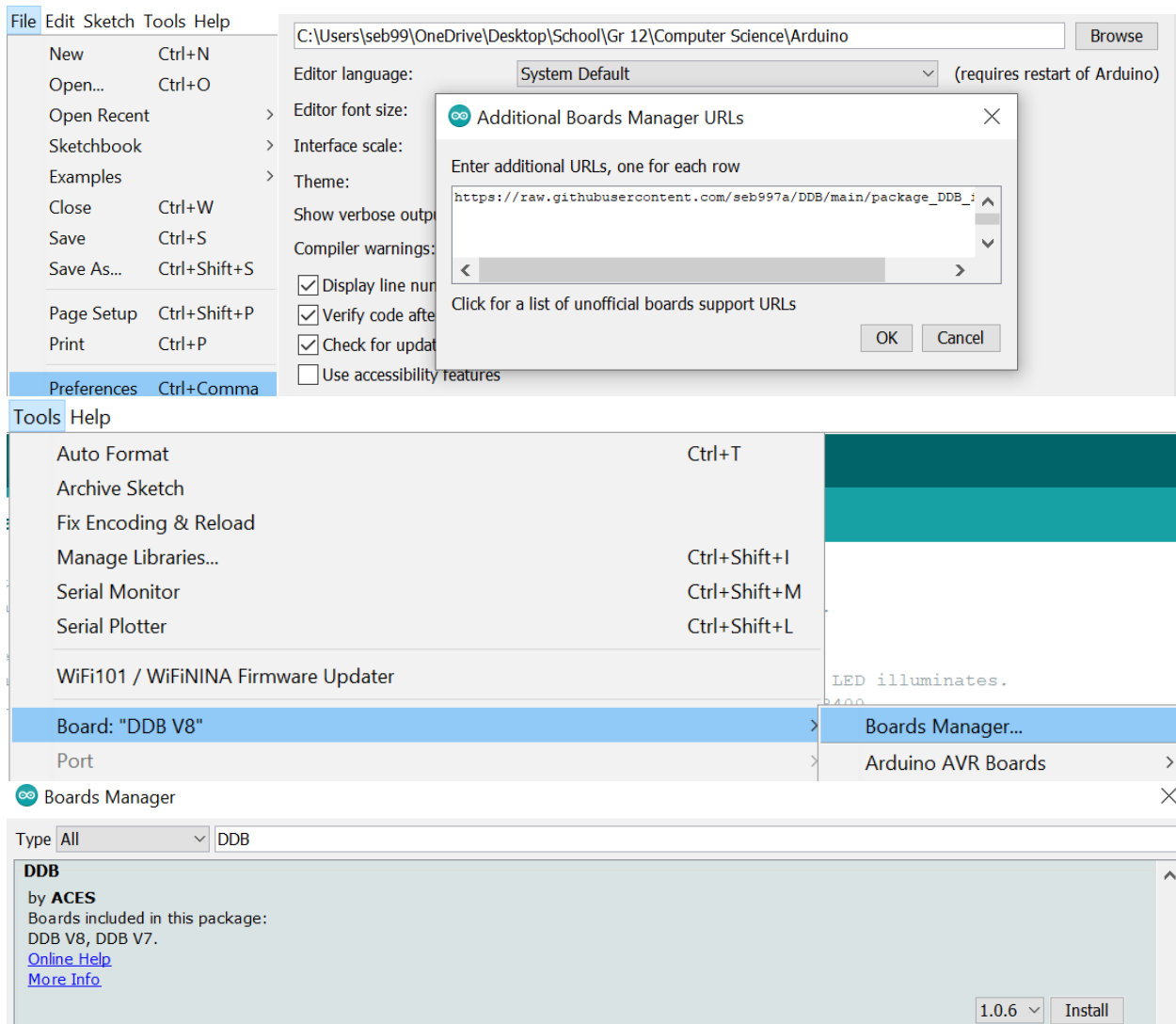
Once installed, the DDB V8 can be selected. To upload, select the V8 and then press upload with nothing plugged in. The message below will appear and then the DDB V8 should be plugged in using the "USB Flash" port. Once plugged in, the sketch will upload.

```
> Please plug in the device (will time out in 60 seconds) ...
```

Arduino Windows Setup

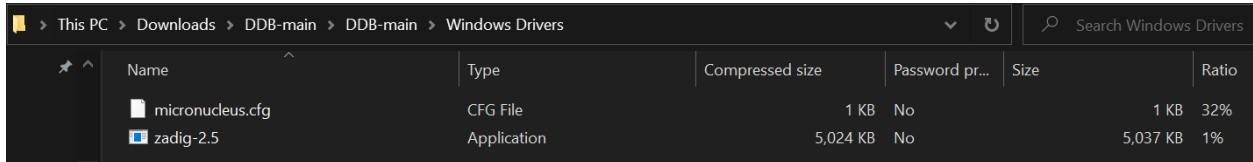
Uploading to the DDB through the Arduino IDE in Windows is very similar to mac except for requiring the manual installation of drivers. The first step is opening the Arduino IDE and then navigating to preferences then additional boards manager and then adding the following link:

https://raw.githubusercontent.com/seb997a/DDB/main/package_DDB_index.json After adding the boards manager URL navigate to tools, board, boards manager and then search DDB and click install.

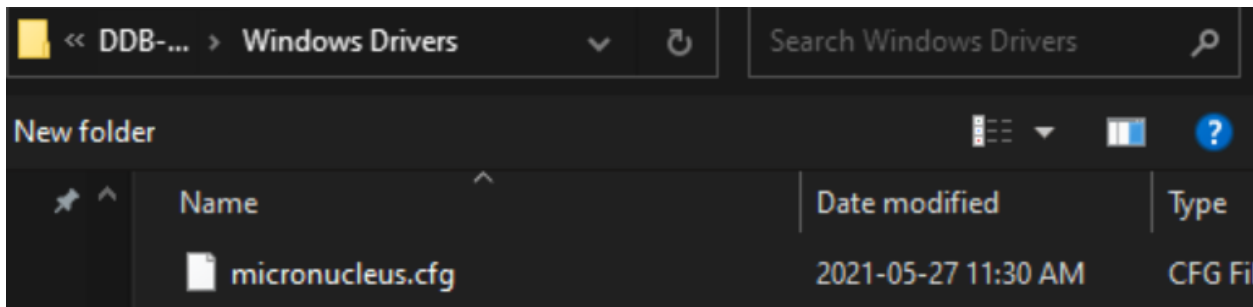
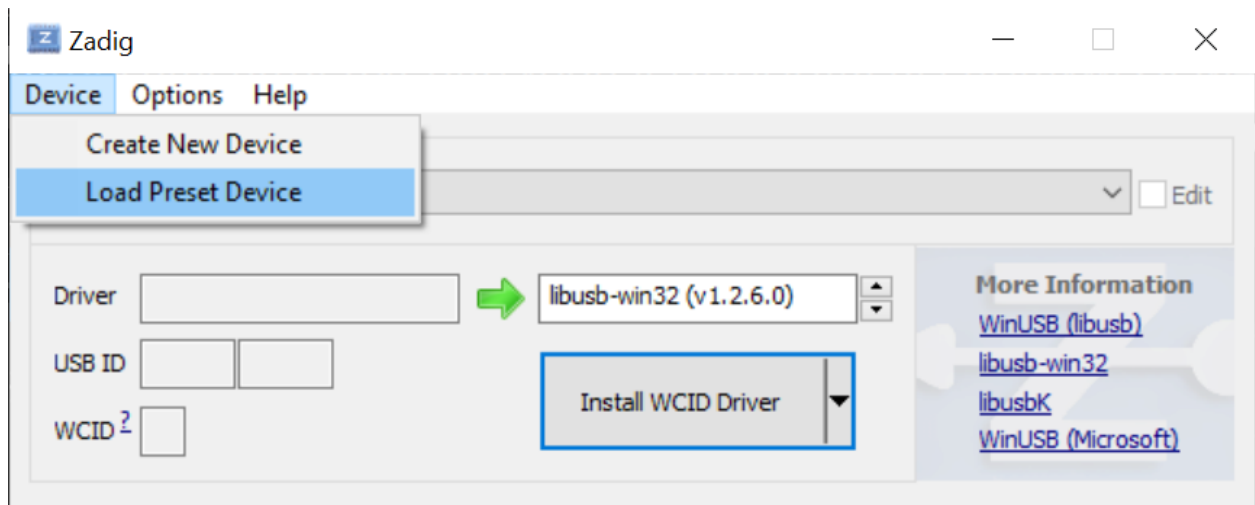


After installing the DDB core, close the Arduino IDE.

Next navigate to the github and download the repository at the following link: <https://github.com/seb997a/DDB>. After installing, open the zadig executable. In the zadig app, go to device, load preset device, and then load the “micronucleus.cfg” file. Next, make sure “libusb-win32” is selected as the driver. Once done, plug in the DDB using the flash USB and then click install. Once installed, close the driver installer and open the IDE again.



Name	Type	Compressed size	Password pr...	Size	Ratio
micronucleus.cfg	CFG File	1 KB	No	1 KB	32%
zadig-2.5	Application	5,024 KB	No	5,037 KB	1%



Name	Date modified	Type
micronucleus.cfg	2021-05-27 11:30 AM	CFG Fi

To upload through the IDE and then follow the same instructions as in mac: To upload, select the V8 and then press upload with nothing plugged in. The message below will appear and then the DDB V8 should be plugged in using the “USB Flash” port. Once plugged in, the sketch will upload.

```
> Please plug in the device (will time out in 60 seconds) ...
```

Atmel Studio Setup

To upload through Atmel studio, navigate to tools, and then external tools. Once opened, select add. Name it DDB V8 and then add the necessary parameters. In command, add the following and replace USERNAME with the correct username.

“C:\Users\USERNAME\AppData\Local\Arduino15\packages\DDB\tools\micronucleus\2.5-azd1b\micronucleus.exe”

Then in arguments add the following: “--run --run --timeout 60 \$(TargetDir)\$(TargetName).hex”

Once added, the device can be flashed through Atmel Studio.

