Open Panzer TCB - Bootloader & Firmware Instructions

Product Profile
The Open Panzer Tank Control Board (TCB) is a micro-controller board based on the ATmega2560 processor. It is used to control RC models, specifically RC tanks. Although it is Arduino-compatible we use a custom bootloader to provide additional functionality.

General test procedure
There are two pieces of software that must be loaded onto the chip after production, and each uses a different process.

1) The first step is to flash a custom **bootloader** to the chip using an In-System Programmer. This provides the processor with the basic ability to communicate and receiver commands.
2) Next we flash the **firmware**, which is the operating system of the board. This can be done either through the USB connector or the alternate "SER. 1" port on the TCB board.

Required Tools

Device	**Description**
Computer	Running Windows system
USB Micro-B Cable, ***OR*** 6-16 volt battery/power supply with 2-pin JST-XH plug	You can power the TCB either through USB or from a battery/power supply (not to exceed 16 volts). USB is via Micro-B cable. Battery/DC power supply requires a 2-pin JST-XH plug.
- AVIRSP mkII, ***OR*** - USBasp, ***OR*** - Pololu USB AVR Programmer v2.x	Used to flash bootloader
ISP Pogo adapter	For example the SparkFun ISP Pogo Adapter or similar, used to flash bootloader
5 volt FTDI Adapter	Optional, is faster when flashing firmware to large number of boards. Many options, including: - Adafruit FTDI Friend - SparkFun FTDI Basic Breakout 5V - the above mentioned Pololu Programmer

USB Mini B cable	If using the optional FTDI adapter, this cable will be required.

Software tools	**Description**
Zadig	USB drivers for the AVRISP mkII or the USBasp (not needed if using the Pololu programmer below)
Pololu USB AVR Programmer V2.x Drivers	USB drivers for the Pololu USB AVR Programmer V2.x (not needed if using the AVRISP mkII or USBasp programmers above)
OP Config	Uploading firmware HEX file

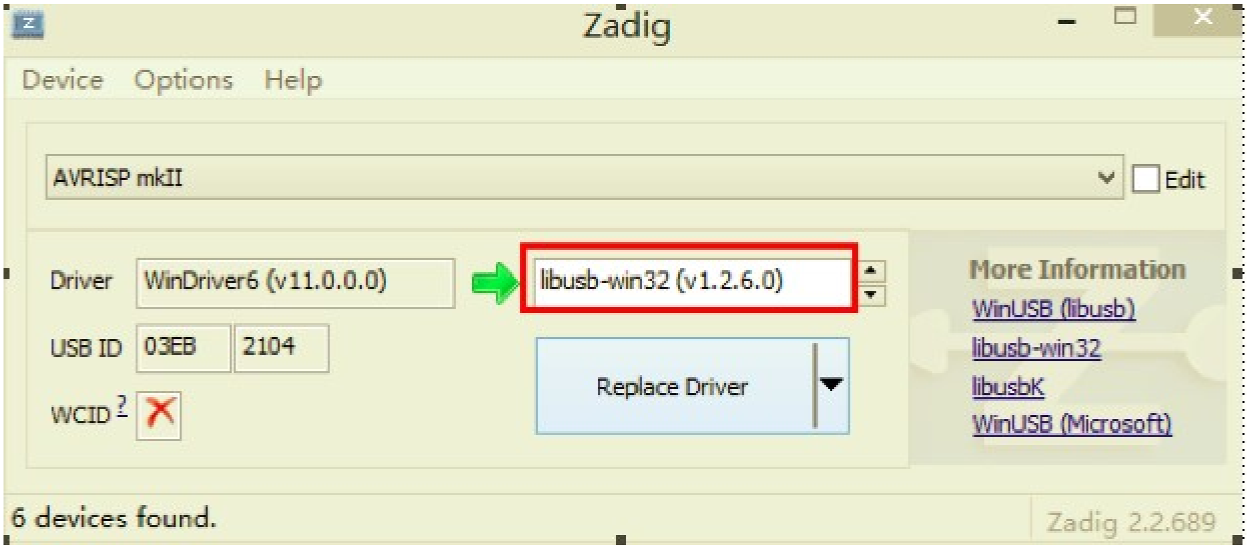
Program Files Included

1)	avrdude.exe	(flash bootloader)
2)	avrdude.conf	(flash bootloader)
3)	optcb2560_boot.hex	(bootloader hex)
4)	bootloader_batch_script.bat	(batch file to automate bootloader flash)

1. Flash Bootloader Procedure

The bootloader is a small program placed on the chip that will give it the ability to communicate and accept firmware updates. At this point the chip is blank and so it is unable to read the serial/USB port, therefore, we must flash the bootloader using an ISP (“In-System Programmer”) device.

1) Connect your ISP device (AVRISP mkII / USBasp / Pololu USB AVR Programmer) to computer.
2) If you are using the AVRISP mkII or USBasp, run the Zadig program and select your device from the list.
3) Install or Replace driver to libusb-win32. NOTE – if you encounter problems with flashing, you can also try libusbK driver.



AVRISP mkII

Driver WinDriver6 (v11.0.0.0) → libusb-win32 (v1.2.6.0)

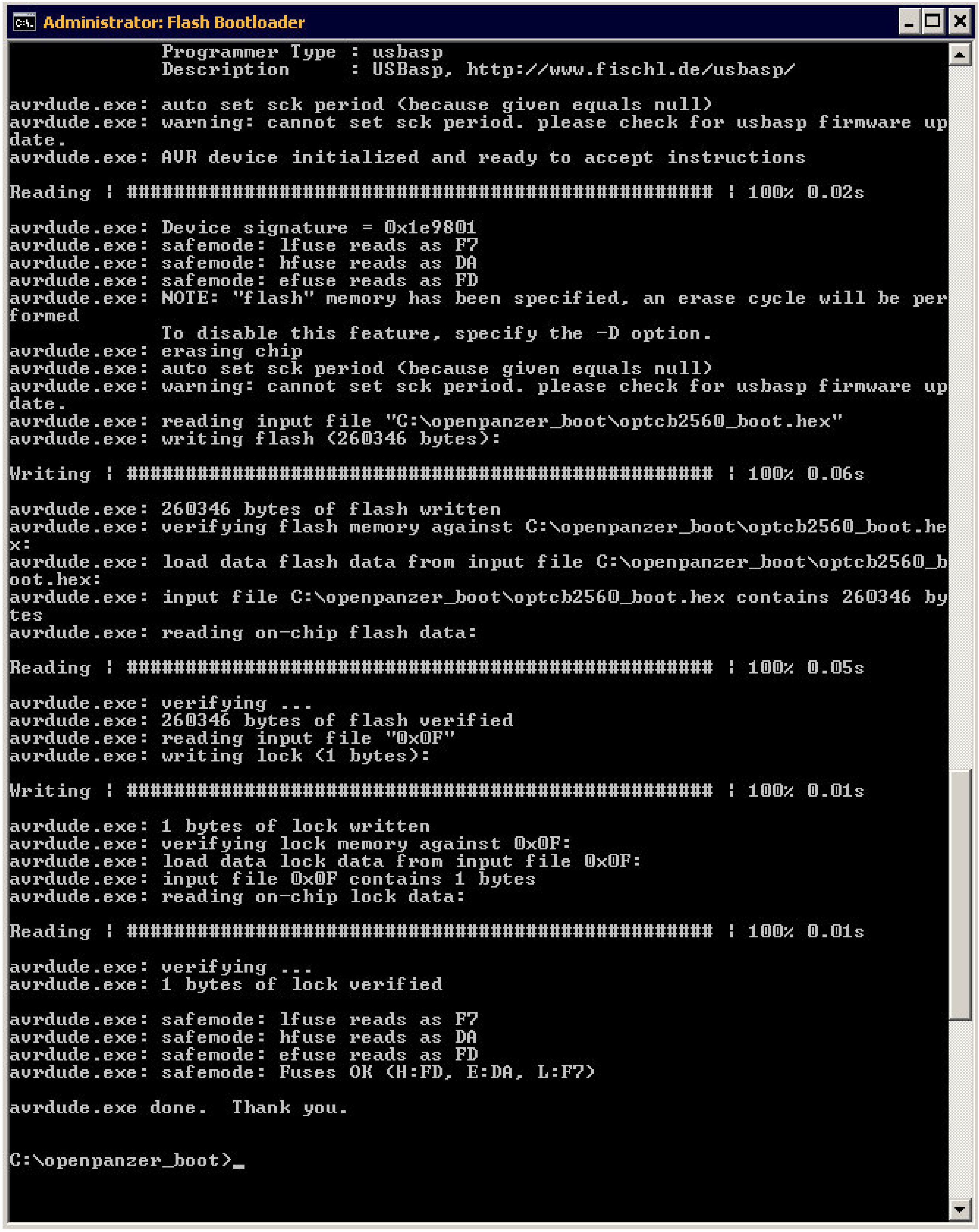
USB ID 03EB 2104

6 devices found.

4) If you are using the Pololu USB AVR Programmer v2.x, install the Windows drivers as described on Pololu’s resource page: https://www.pololu.com/docs/0J67/4.1
5) Open ”`bootloader_batch_script.bat`” file with a text editor such as Notepad. There are three sections that you need to modify:
    a. Paths to avrdude.exe, avrdude.conf, and optcb2560_boot.hex
    b. The type of programmer you will be using: AVRISP mkII, USBasp, or the Pololu
    c. The port that your programmer will be communicating on.

    These sections are clearly marked in the file. Make any adjustments necessary for your environment then save and close the batch file.
6) Power the TCB either with a Micro-B USB cable or with a battery/DC power supply using a 2-pin JST-XH plugged into the BATT terminal of the TCB.
7) Run ”`bootloader_batch_script.bat`” file (you can just double-click it to execute it). The screenshot on the next page should be the result.
8) You can also confirm successful bootloader install by observing the board. The **red** LED will be flashing very slowly (once per second).

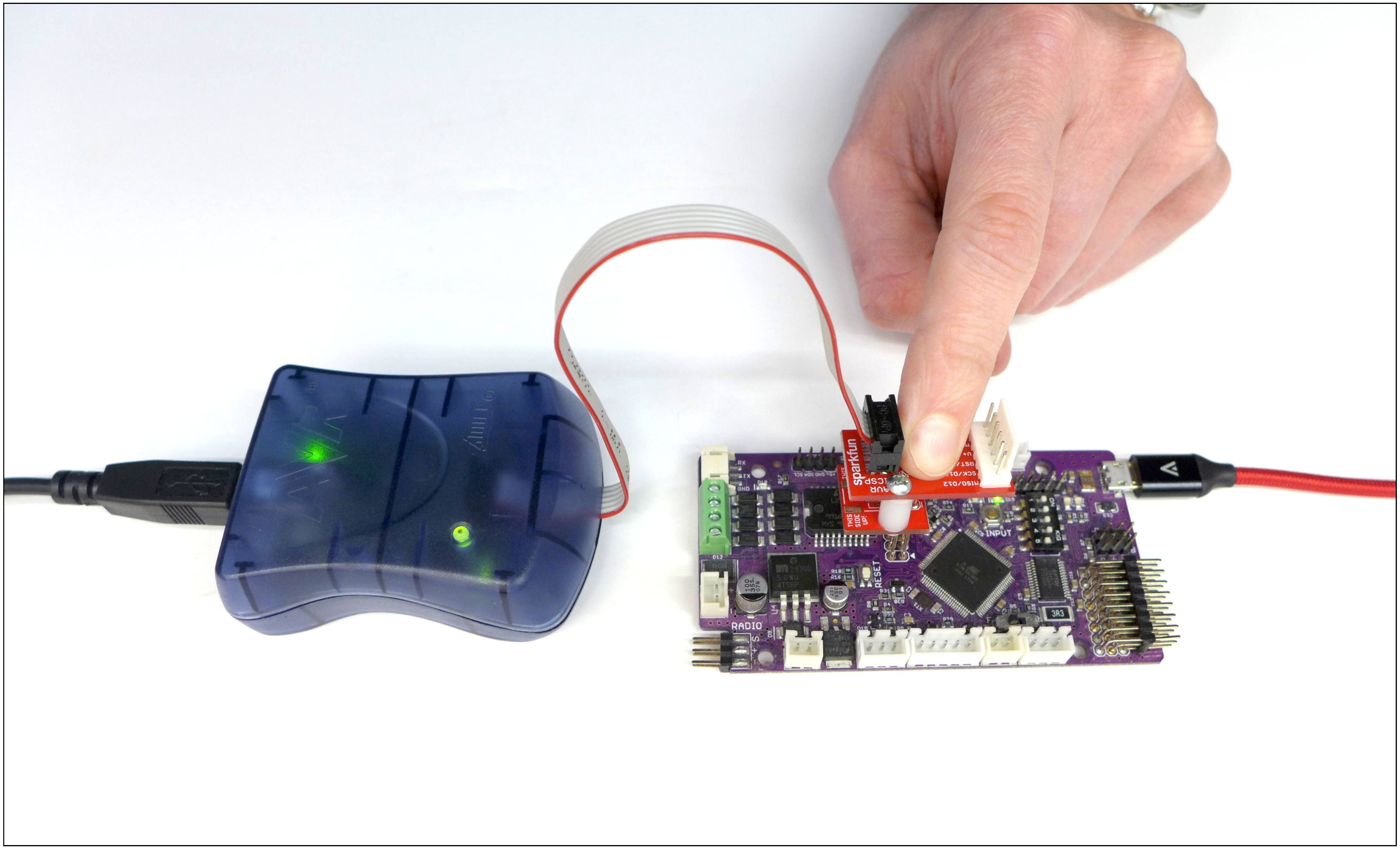
Successful Bootloader flash



```
                Programmer Type : usbasp
                Description     : USBasp, http://www.fischl.de/usbasp/

avrdude.exe: auto set sck period (because given equals null)
avrdude.exe: warning: cannot set sck period. please check for usbasp firmware up
date.
avrdude.exe: AVR device initialized and ready to accept instructions

Reading | ################################################## | 100% 0.02s

avrdude.exe: Device signature = 0x1e9801
avrdude.exe: safemode: lfuse reads as F7
avrdude.exe: safemode: hfuse reads as DA
avrdude.exe: safemode: efuse reads as FD
avrdude.exe: NOTE: "flash" memory has been specified, an erase cycle will be per
formed
             To disable this feature, specify the -D option.
avrdude.exe: erasing chip
avrdude.exe: auto set sck period (because given equals null)
avrdude.exe: warning: cannot set sck period. please check for usbasp firmware up
date.
avrdude.exe: reading input file "C:\openpanzer_boot\optcb2560_boot.hex"
avrdude.exe: writing flash (260346 bytes):

Writing | ################################################## | 100% 0.06s

avrdude.exe: 260346 bytes of flash written
avrdude.exe: verifying flash memory against C:\openpanzer_boot\optcb2560_boot.he
x:
avrdude.exe: load data flash data from input file C:\openpanzer_boot\optcb2560_b
oot.hex:
avrdude.exe: input file C:\openpanzer_boot\optcb2560_boot.hex contains 260346 by
tes
avrdude.exe: reading on-chip flash data:

Reading | ################################################## | 100% 0.05s

avrdude.exe: verifying ...
avrdude.exe: 260346 bytes of flash verified
avrdude.exe: reading input file "0x0F"
avrdude.exe: writing lock (1 bytes):

Writing | ################################################## | 100% 0.01s

avrdude.exe: 1 bytes of lock written
avrdude.exe: verifying lock memory against 0x0F:
avrdude.exe: load data lock data from input file 0x0F:
avrdude.exe: input file 0x0F contains 1 bytes
avrdude.exe: reading on-chip lock data:

Reading | ################################################## | 100% 0.01s

avrdude.exe: verifying ...
avrdude.exe: 1 bytes of lock verified

avrdude.exe: safemode: lfuse reads as F7
avrdude.exe: safemode: hfuse reads as DA
avrdude.exe: safemode: efuse reads as FD
avrdude.exe: safemode: Fuses OK (H:FD, E:DA, L:F7)

avrdude.exe done.  Thank you.


C:\openpanzer_boot>_
```

Example of flashing bootloader with AVRISP mkII and SparkFun ISP Pogo Adapter.

(Red USB cable at right is only being used to provide power, it is not communicating)



2. Flash Firmware Procedure

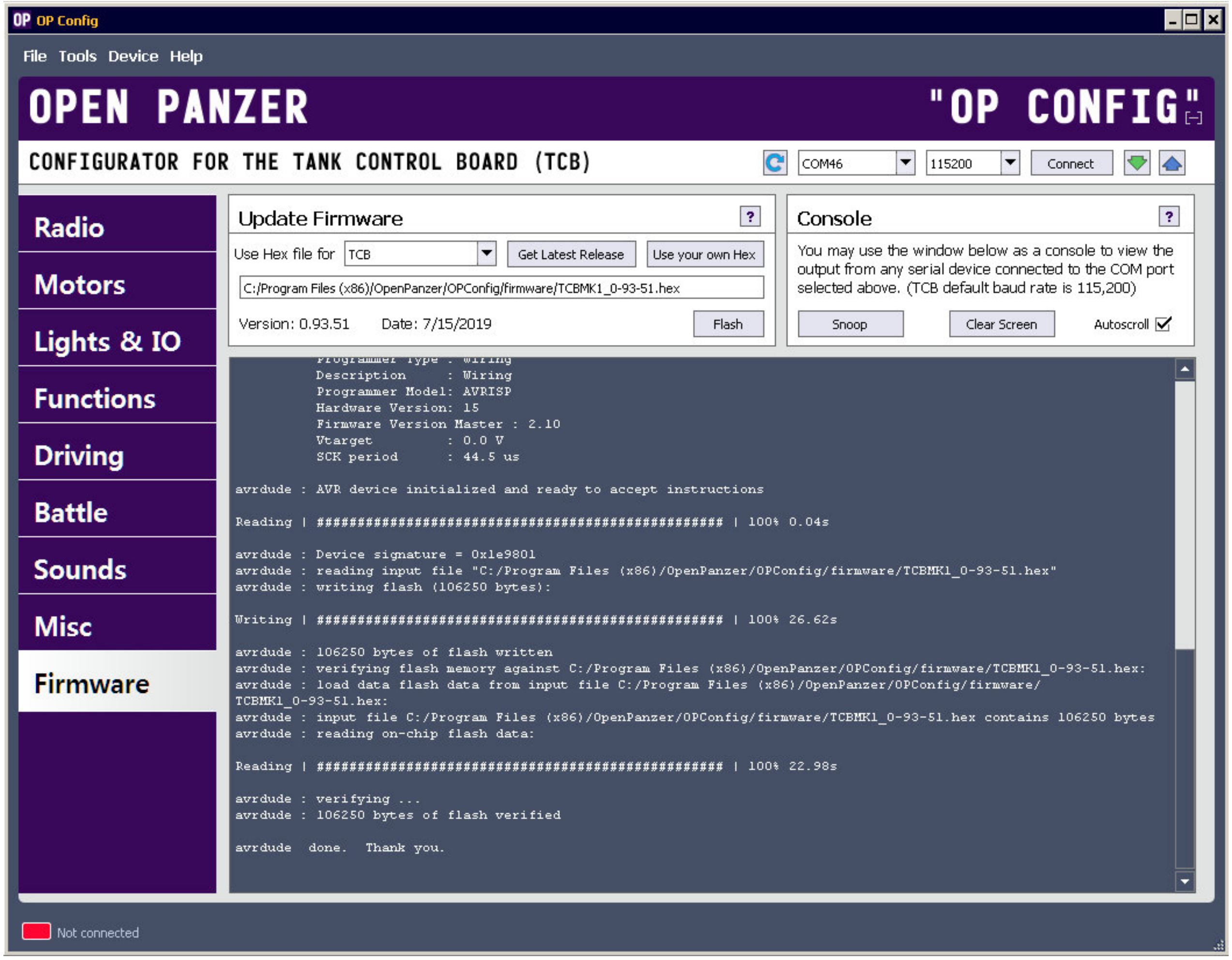
After the bootloader is installed we are able to communicate with the board through normal means, and this allows us to load the TCB Firmware (the actual Open Panzer program).

1) Open the OP Config program and go to the Firmware tab. The “Use Hex file for” list should already default to “TCB” (*not* “TCB – DIY Version”!)

2) Click the “Get Latest Release” button and the latest firmware will be downloaded. Now click the “Flash” button to load the firmware on to the board.

3) There are two ways to connect to the TCB for loading firmware – first, you can connect the board directly to your computer with the USB cable (make sure dipswitch #5 on the TCB is in the “ON” position). This is the normal way that most users will prefer. However, if you are loading firmware onto hundreds of boards, Windows will create a new COM port for each board and this will take an extremely long time.

4) In the case of programming large numbers of boards, the alternate method is faster. Here we use a serial adapter connected to the “SER. 1” port on the TCB. Various adapters are listed at the beginning of this document from Adafruit, SparkFun or Pololu. Windows will create a single COM port for this adapter, and then you will use the same adapter to program as many boards as you want. In order for the TCB to accept programming commands through the “SER.1” port instead of through the USB port as usual, we need to set dipswitch #5 to the ”OFF” position. Now plug the serial adapter into the “SER. 1” connector on the TCB. Make sure to match “black” and “green” on adapter and TCB connectors so polarity is correct (your adapter may also say “B” and “GND” which matches to “black” and “green” respectively). Now you can load firmware from OP Config by selecting the COM port for your adapter and clicking the Flash button. If you are using the Pololu adapter note that it creates two COM ports, you want to use the “TTL Port” shown in the Pololu Configuration Utility. But be sure to disconnect and close the Pololu Configuration Utility before trying to flash firmware with OP Config, otherwise they will conflict with each other.



Use Hex file for TCB

C:/Program Files (x86)/OpenPanzer/OPConfig/firmware/TCBMK1_0-93-51.hex

Version: 0.93.51 Date: 7/15/2019

You may use the window below as a console to view the output from any serial device connected to the COM port selected above. (TCB default baud rate is 115,200)

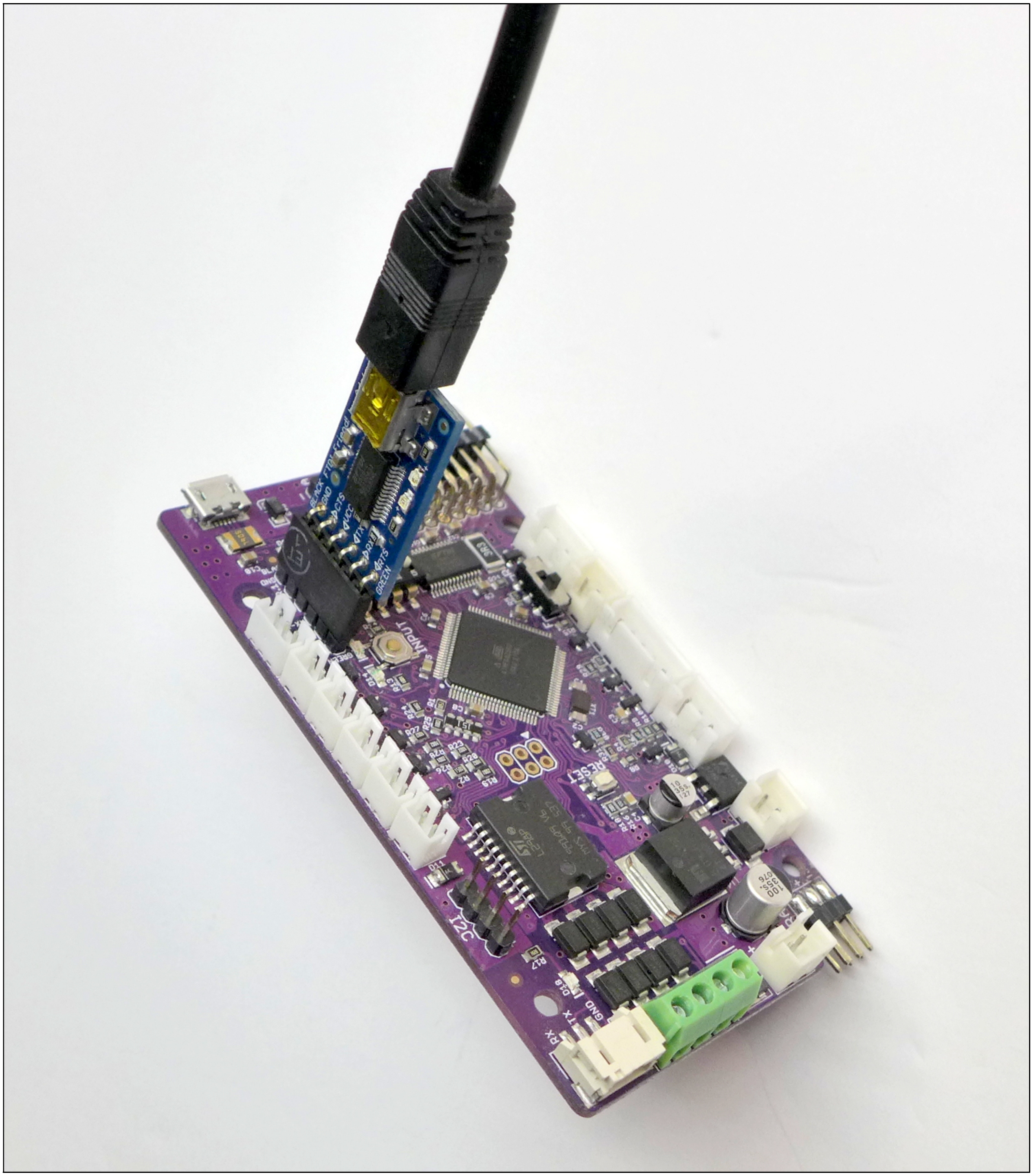
```
         Programmer Type : Wiring
         Description     : Wiring
         Programmer Model: AVRISP
         Hardware Version: 15
         Firmware Version Master : 2.10
         Vtarget         : 0.0 V
         SCK period      : 44.5 us

avrdude : AVR device initialized and ready to accept instructions

Reading | ################################################## | 100% 0.04s

avrdude : Device signature = 0x1e9801
avrdude : reading input file "C:/Program Files (x86)/OpenPanzer/OPConfig/firmware/TCBMK1_0-93-51.hex"
avrdude : writing flash (106250 bytes):

Writing | ################################################## | 100% 26.62s

avrdude : 106250 bytes of flash written
avrdude : verifying flash memory against C:/Program Files (x86)/OpenPanzer/OPConfig/firmware/TCBMK1_0-93-51.hex:
avrdude : load data flash data from input file C:/Program Files (x86)/OpenPanzer/OPConfig/firmware/
TCBMK1_0-93-51.hex:
avrdude : input file C:/Program Files (x86)/OpenPanzer/OPConfig/firmware/TCBMK1_0-93-51.hex contains 106250 bytes
avrdude : reading on-chip flash data:

Reading | ################################################## | 100% 22.98s

avrdude : verifying ...
avrdude : 106250 bytes of flash verified

avrdude  done.  Thank you.
```

5) If firmware flash is successful you will see the results shown above. You can also confirm by looking at the board. The **red** and **green** LEDs will be flashing in alternating pattern very quickly. The **blue** LED will be flashing at medium rate.

6) **NOTE:** Before sending the board to customers, put dipswitch #5 back to the “ON” position since they will be using USB communications. This will save you many support requests!

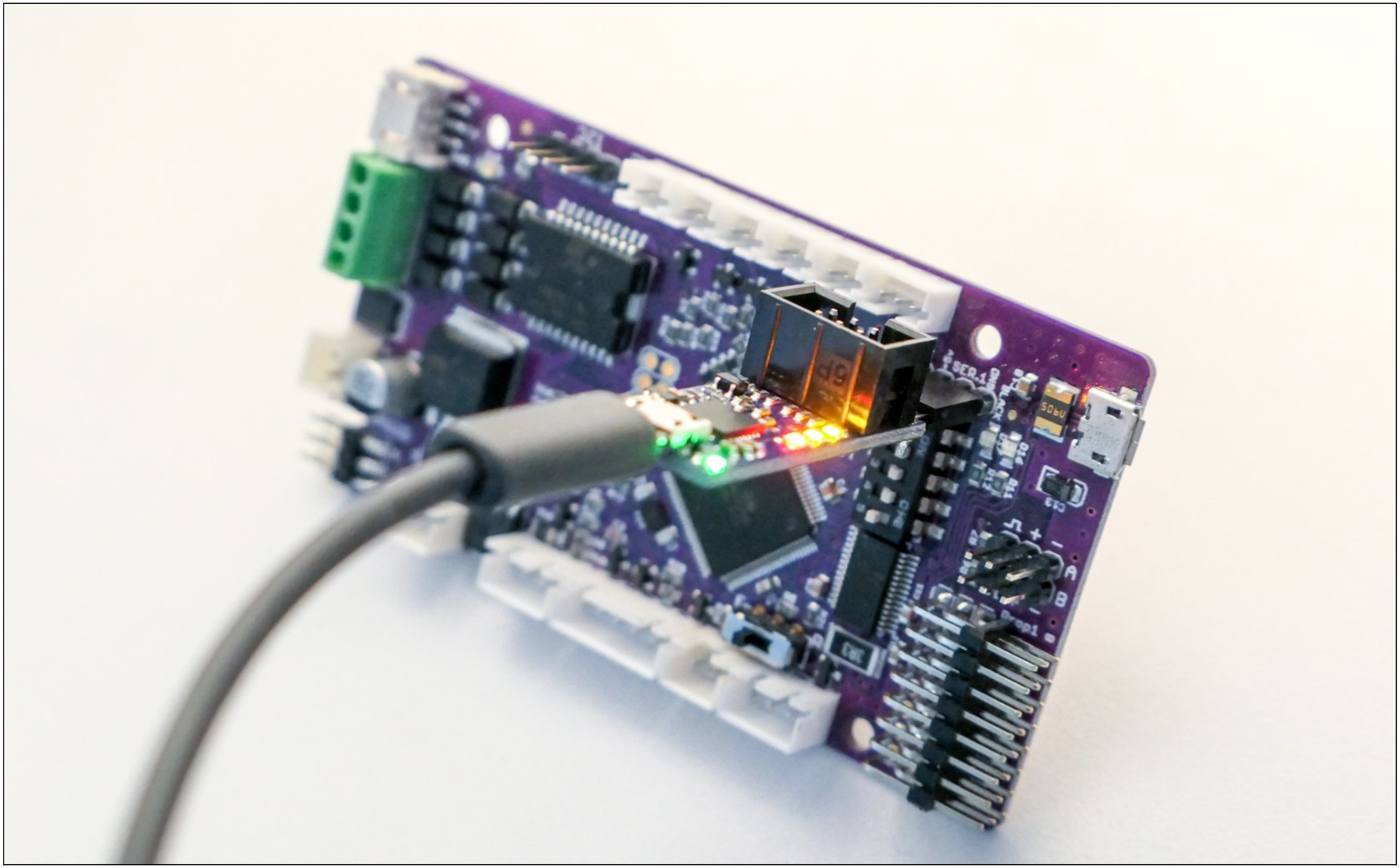
Example using the Adafruit FTDI Friend to flash firmware over the alternate "SER. 1" port

(you also need to provide power, not shown)



Example using the Pololu Programmer to flash firmware over the alternate "SER. 1" port

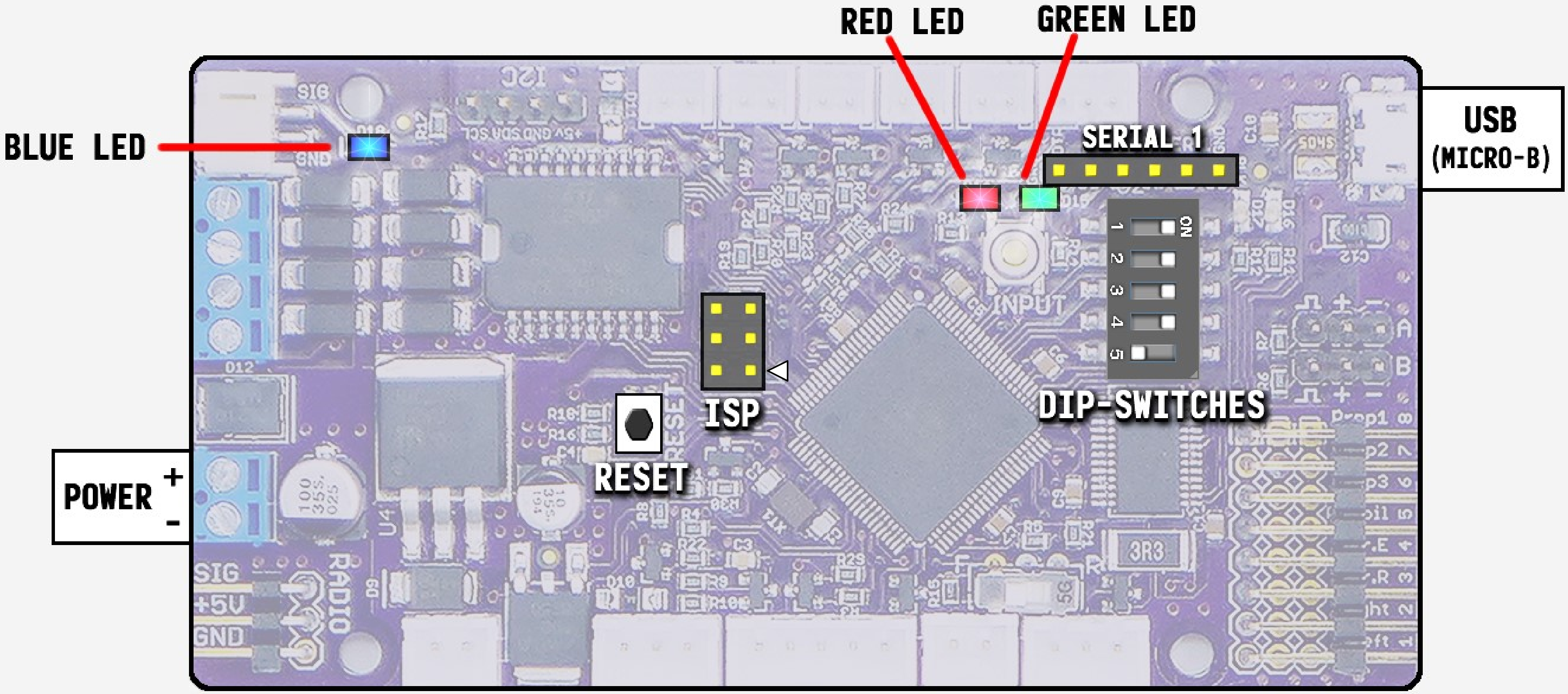
(you also need to provide power, not shown)



TCB Board Layout

USB/Power - Use to provide power during bootloader flash
ISP - Use pogo adapter to flash bootloader
Red LED - Will flash slowly (once per second) after bootloader flash

SERIAL 1 - Use FTDI Friend to flash firmware
Dipswitch - Position #5 must be in "OFF" position to be able to flash with SERIAL 1
Red/Green - Will flash alternating very fast after firmware flash
Blue LED - Will flash medium after firmware flash



Note: place dipswitch #5 back to the "ON" position before sending the board to customers, this will save you many support requests!