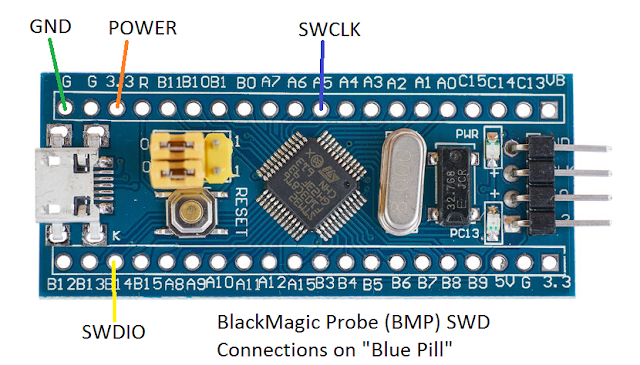
The Black Magic Probe for flashing the SAM D20 MCU’s has been made from a STM32F103 Blue Pill board ~5$ on EBay.

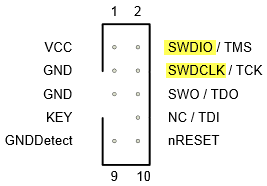
Link to create: <https://hackaday.io/project/28180-bluetooth-gamepad-phone-case/log/71316-stm32-black-magic-probe-flashing>

Once created the STM32 is visible as 2 separate com ports.

1) Usb - > TTL UART converter on pins (PA2 TX, PA3 RX)

2) BMP gdb debugger via SWD or JTAG. SWD pins shown below:





Note: If powered by another means, only the SWDIO, SWDCLK and GND may be required.

Script for auto flashing board found here, or see instructions below: <https://github.com/stuckatmarine/Black-Magic-Probe-Flashing-Script>

**Windows :**

* Install GNU Embedded Toolchain (.exe works well)

<https://developer.arm.com/open-source/gnu-toolchain/gnu-rm/downloads>

* Official cli docs: <https://github.com/blacksphere/blackmagic/wiki/Useful-GDB-commands>

Put these cmds in a .bat file or run individually to simply load program on device.

‘’‘

"**C:\Program Files (x86)\GNU Tools Arm Embedded\7 18-q2-update\bin\arm-none-eabi-gdb.exe**" \

-ex "target extended-remote **COM5**" \

-ex "monitor swdp\_scan" \

-ex "att 1" \

-ex "file **C:/ARM/LED\_TOGGLE1.elf**"\

-ex "load"\

‘’’

Notes:

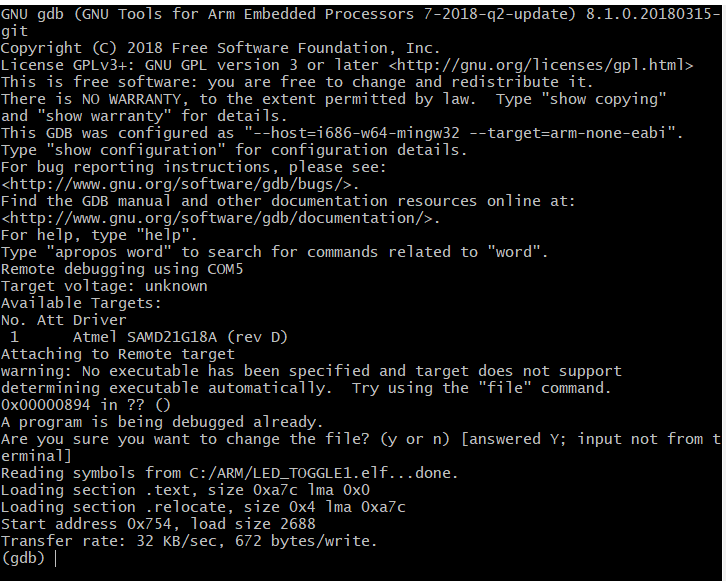
“” to remove windows \ and spaces issues, also note space before each -ex

\ -ex “” not required if running manually but be weary about spaces and \, (add \ before each issue to fix)

**Bold** = use your arm-none-eabi-gdb.exe path, port BMP is on, filename.elf with dir info

q to exit safely

## this also works for flashing a bootloader.hex or program.hex file



**Linux/Mac (unconfirmed) :**

* Same as windows except .bat script, write linux version/go manual
* Install GNU Embedded Toolchain (pip, xpm, link above?)
* Instead of COMX use /Dev/ttyxxx