Voron Skirt Button PCB Klipper Firmware

Compile Firmware

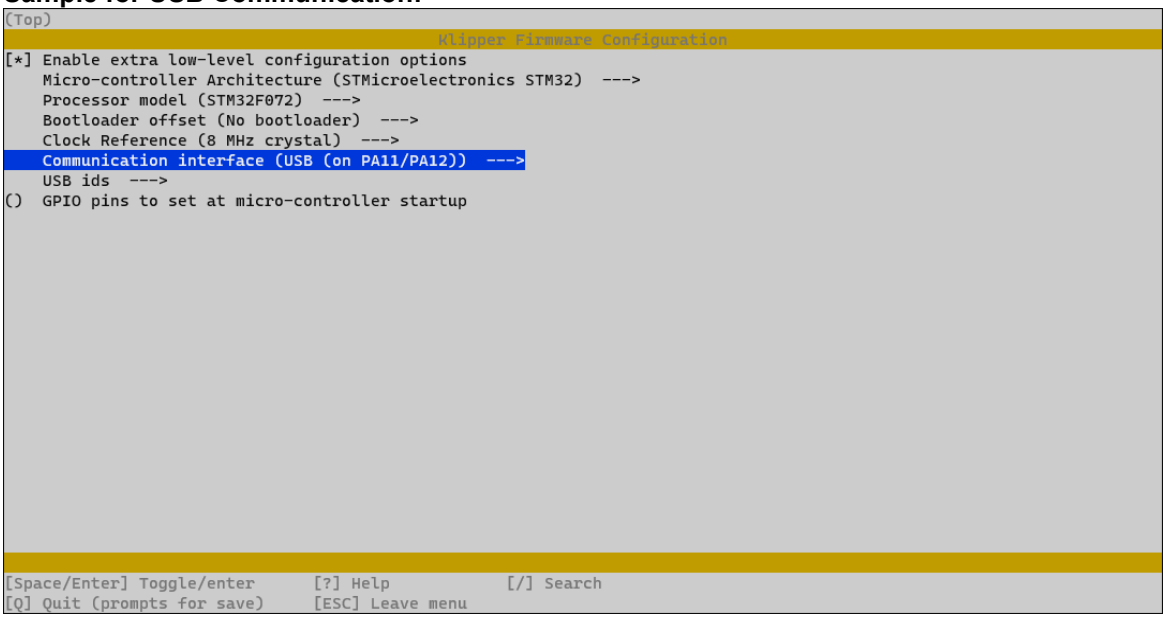
1. Connect the Voron Skirt Button PCB to your Raspberry Pi
2. SSH to your Raspberry Pi
3. Type the following command line:

```
cd ~/klipper/
make menuconfig
```

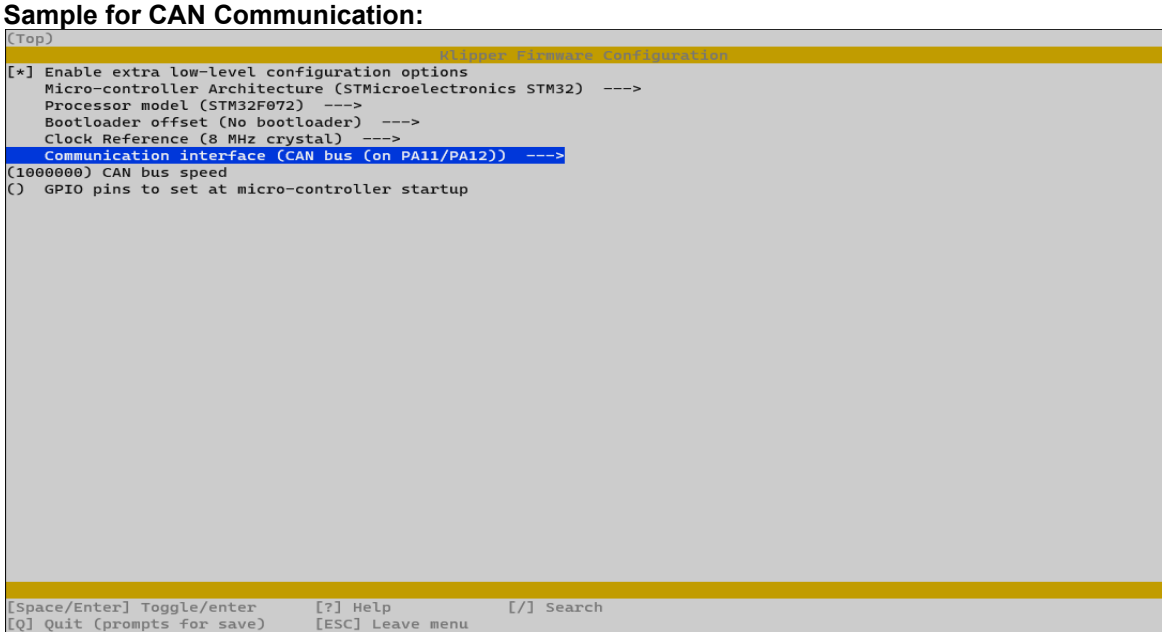
4. Compile the firmware with the following configuration:

[*] Enable extra low-level configuration options
    Micro-controller Architecture (STMicroelectronics STM32) --->
    Processor model (STM32F072) --->
    Bootloader offset (No bootloader) --->
    Clock Reference (8 MHz crystal) --->
If using USB communication over Type-C Communication interface (USB (on PA11/PA12)) --->
If using CANBus communication Communication interface (CAN bus (on PA11/PA12)) ---> (1000000) CAN bus speed (set the speed to your CAN speed configuration)

Sample for USB Communication:



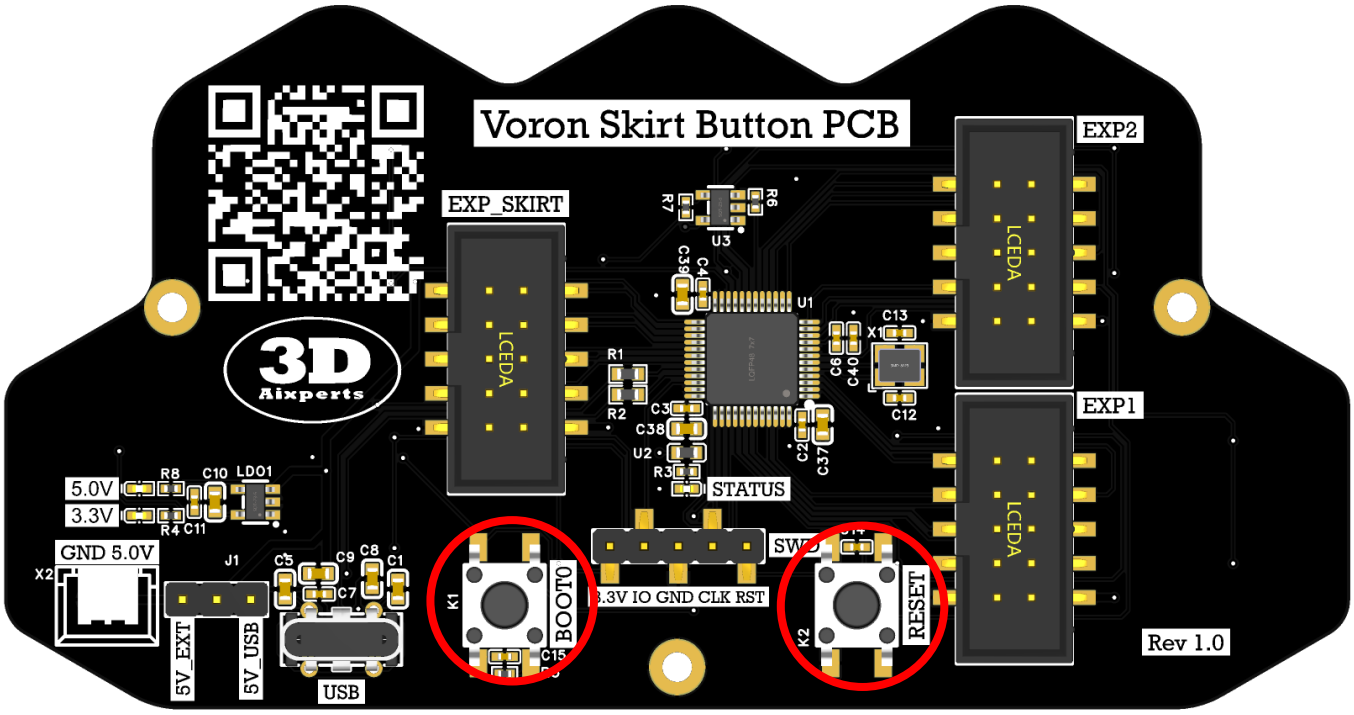
Sample for CAN Communication:



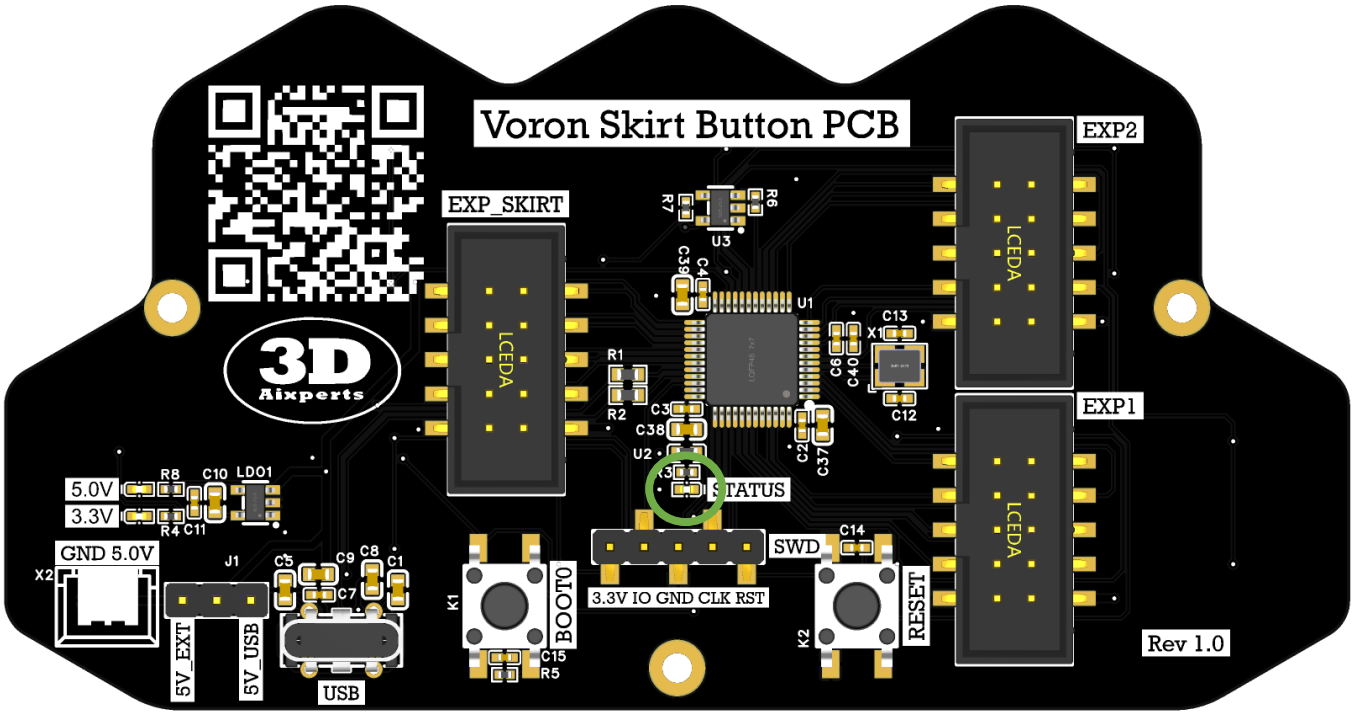
5. When completed the configuration selection, type „q“ to exit the configuration interface. Select „Yes“ when asked wheter to save the configuration.
6. Enter „make clean“ to remove older firmware compilation files.
7. Enter „make“ to compile the new firmware. The firmware file „klipper.bin“ would be located in home/pi/klipper/out

Updating Firmware

1. Make sure the Skirt Button PCB is connected to your Raspberry Pi
2. Press the „BOOT0“ button, then press and release the „RESET“ button to entert he DFU mode.



3. Make sure that the „STATUS“ LED is green. This indicates that the Board is in DFU mode



4. Type „lsusb“ in the SSH terminal command line to query the ID oft he DFU device

```
pi@testbench:~/klipper $ lsusb
Bus 001 Device 004: ID 0483:df11 STMicroelectronics STM Device in DFU Mode
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp. SMSC9512/9514 Fast Ethernet Adapter
Bus 001 Device 002: ID 0424:9514 Standard Microsystems Corp. SMC9514 Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

5. Type „make flash FLASH_DEVICE=0483:df11“ to download the firmware to the board (Notice: Replace 0483:df11 with the actual ID oft he DFU device queried in the previous step).

```
pi@testbench:~/klipper $ make flash FLASH_DEVICE=0483:df11
  Flashing out/klipper.bin to 0483:df11
sudo dfu-util -d ,0483:df11 -R -a 0 -s 0x8000000:leave -D out/klipper.bin

dfu-util 0.9

Copyright 2005-2009 Weston Schmidt, Harald Welte and OpenMoko Inc.
Copyright 2010-2016 Tormod Volden and Stefan Schmidt
This program is Free Software and has ABSOLUTELY NO WARRANTY
Please report bugs to http://sourceforge.net/p/dfu-util/tickets/

dfu-util: Invalid DFU suffix signature
dfu-util: A valid DFU suffix will be required in a future dfu-util release!!!
Opening DFU capable USB device...
ID 0483:df11
Run-time device DFU version 011a
Claiming USB DFU Interface...
Setting Alternate Setting #0 ...
Determining device status: state = dfuERROR, status = 10
dfuERROR, clearing status
Determining device status: state = dfuIDLE, status = 0
dfuIDLE, continuing
DFU mode device DFU version 011a
Device returned transfer size 2048
DfuSe interface name: "Internal Flash  "
Downloading to address = 0x08000000, size = 24504
Download        [=========================] 100%        24504 bytes
Download done.
File downloaded successfully
Transitioning to dfuMANIFEST state
dfu-util: can't detach
Resetting USB to switch back to runtime mode
```

6. Type „ls /dev/serial/by-id/“ to query about the serial ID of the device when finished downloading firmware (The serial ID will only exists when communicating via USB. Ignore this step when communicating via CAN Bus).

```
pi@testbench:~/klipper $ ls /dev/serial/by-id/
usb-Klipper_stm32f072xb_210021001057434831393420-if00
```

Enter the result of the command in the skirt_button_pcb.cfg

```
[mcu skirt]
#canbus_uuid: f382e92cfe29
serial: /dev/serial/by-id/usb-Klipper_stm32f072xb_210021001057434831393420-if00
restart_method: command
```

Note: Your ID may vary