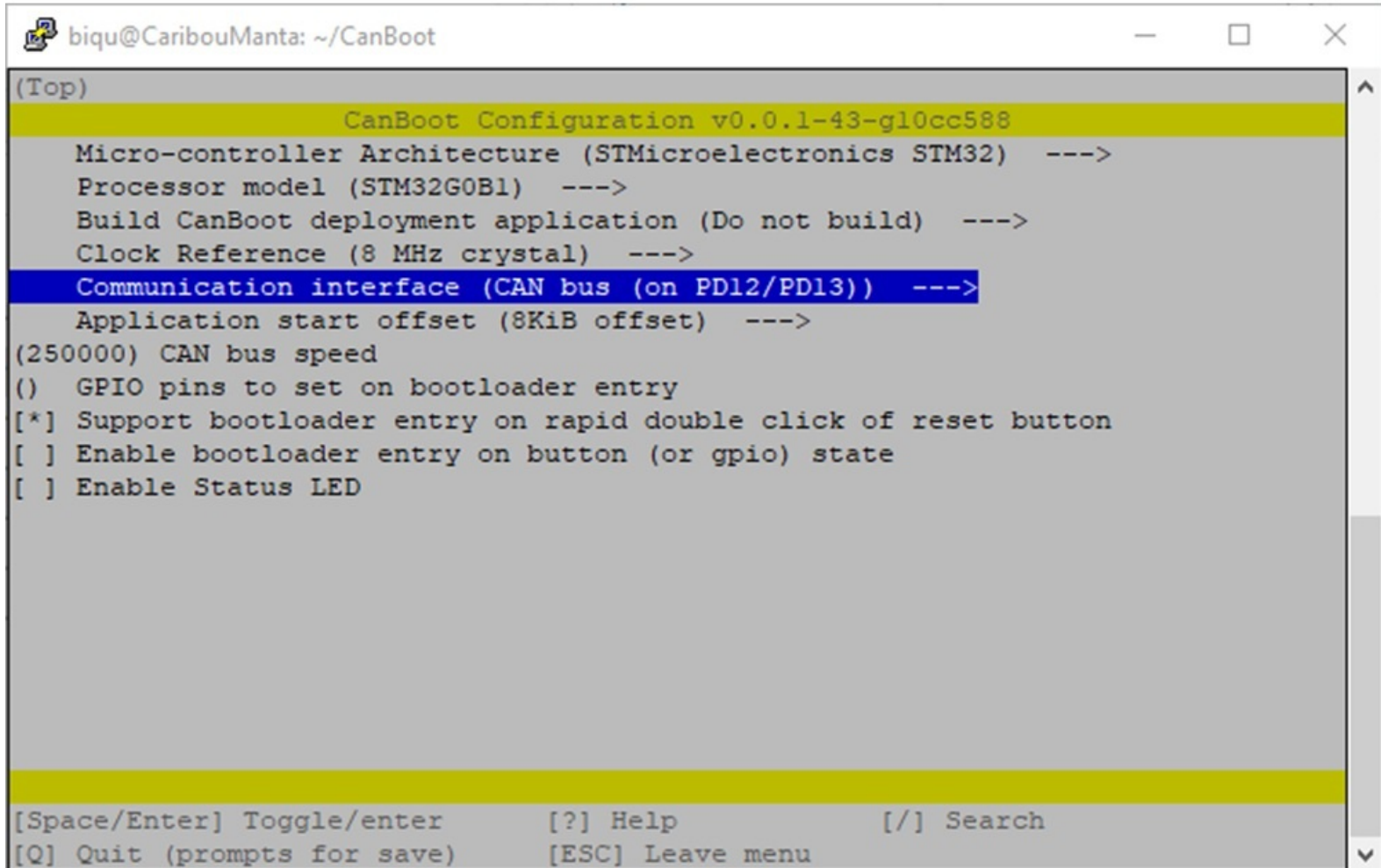


caribou3d

Manta M8P Klipper Installation

Written By: Caribou3d



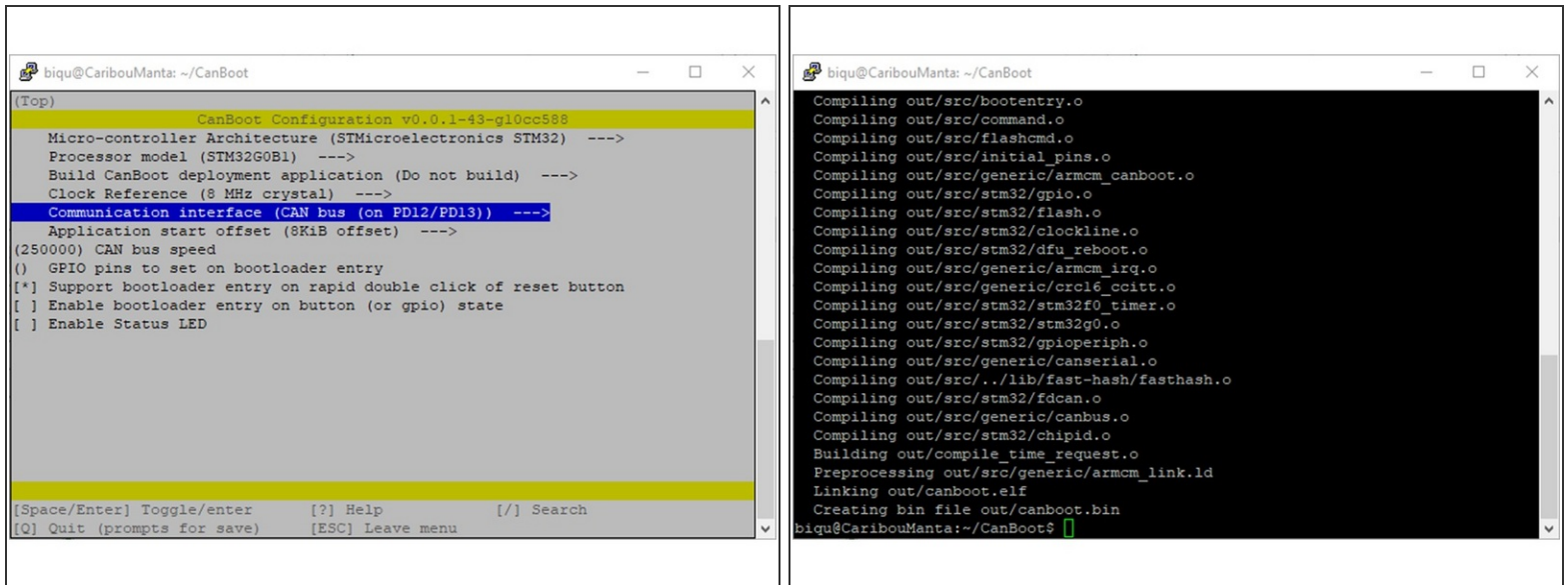
The screenshot shows a terminal window titled "biqu@CaribouManta: ~/CanBoot". The main content is the "CanBoot Configuration v0.0.1-43-gl0cc588" menu. The menu items are as follows:

- (Top)
- CanBoot Configuration v0.0.1-43-gl0cc588
- Micro-controller Architecture (STMicroelectronics STM32) --->
- Processor model (STM32G0B1) --->
- Build CanBoot deployment application (Do not build) --->
- Clock Reference (8 MHz crystal) --->
- Communication interface (CAN bus (on PD12/PD13)) --->
- Application start offset (8KiB offset) --->
- (250000) CAN bus speed
- () GPIO pins to set on bootloader entry
- [*] Support bootloader entry on rapid double click of reset button
- [] Enable bootloader entry on button (or gpio) state
- [] Enable Status LED

At the bottom, there is a footer bar with the following options:

- [Space/Enter] Toggle/enter
- [?] Help
- [/] Search
- [Q] Quit (prompts for save)
- [ESC] Leave menu

Step 1 — Manta M8P Klipper Installation



The image shows two terminal windows side-by-side. The left window displays the CanBoot Configuration v0.0.1-43-g10cc588 menu. The right window shows the compilation process for the CanBoot firmware.

```

biqu@CaribouManta: ~/CanBoot
(Top)
CanBoot Configuration v0.0.1-43-g10cc588
Micro-controller Architecture (STMicroelectronics STM32) --->
Processor model (STM32G0B1) --->
Build CanBoot deployment application (Do not build) --->
Clock Reference (8 MHz crystal) --->
Communication interface (CAN bus (on PD12/PD13)) --->
Application start offset (8KiB offset) --->
(250000) CAN bus speed
() GPIO pins to set on bootloader entry
[*] Support bootloader entry on rapid double click of reset button
[ ] Enable bootloader entry on button (or gpio) state
[ ] Enable Status LED

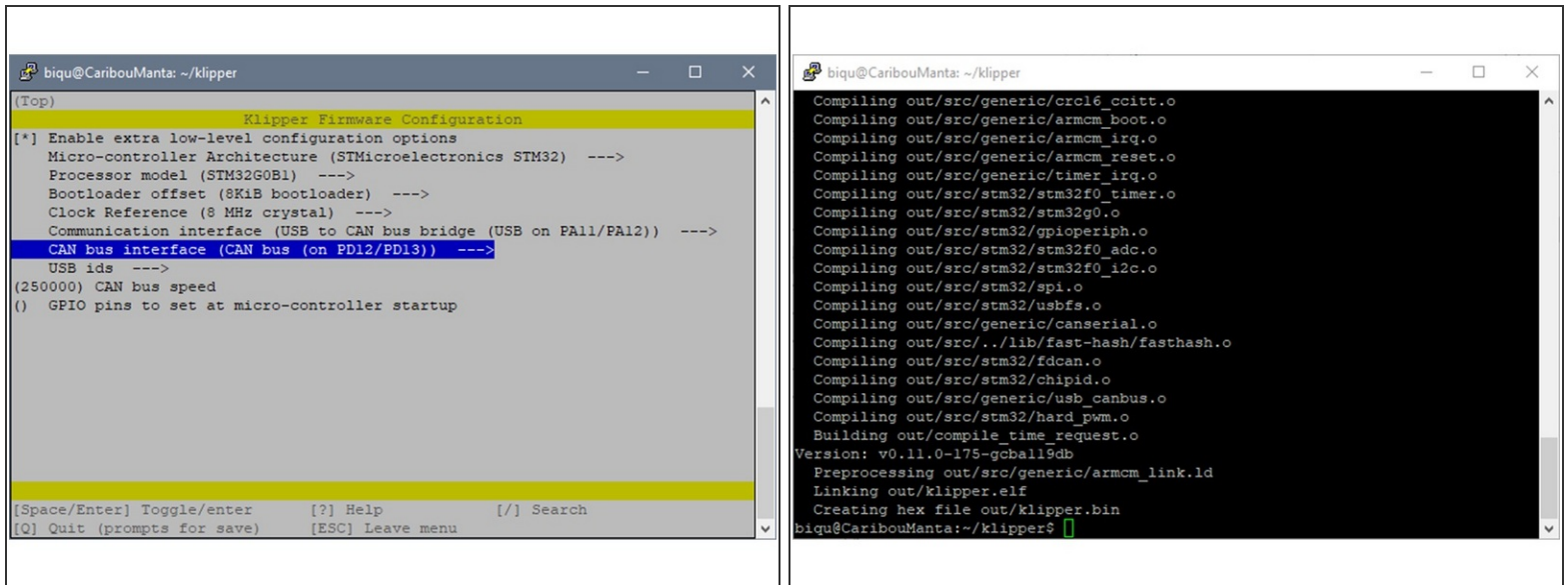
[Space/Enter] Toggle/enter    [?] Help    [/] Search
[Q] Quit (prompts for save)    [ESC] Leave menu

biqu@CaribouManta: ~/CanBoot
Compiling out/src/bootentry.o
Compiling out/src/command.o
Compiling out/src/flashcmd.o
Compiling out/src/initial_pins.o
Compiling out/src/generic/armcm_canboot.o
Compiling out/src/stm32/gpio.o
Compiling out/src/stm32/flash.o
Compiling out/src/stm32/clockline.o
Compiling out/src/stm32/dfu_reboot.o
Compiling out/src/generic/armcm_irq.o
Compiling out/src/generic/crc16_ccitt.o
Compiling out/src/stm32/stm32f0_timer.o
Compiling out/src/stm32/stm32g0.o
Compiling out/src/stm32/gpioperiph.o
Compiling out/src/generic/canserial.o
Compiling out/src/./lib/fast-hash/fasthash.o
Compiling out/src/stm32/fdcant.o
Compiling out/src/generic/canbus.o
Compiling out/src/stm32/chipid.o
Building out/compile_time_request.o
Preprocessing out/src/generic/armcm_link.ld
Linking out/canboot.elf
Creating bin file out/canboot.bin
biqu@CaribouManta:~/CanBoot$

```

- `cd ~ && git clone https://github.com/th33xitus/kiauh.git`
- `git clone https://github.com/Arksine/CanBoot`
- `cd Canbboot`
- `make clean`
- `make menuconfig`
- `make`

Step 2



The image consists of two side-by-side terminal window screenshots. The left window shows the 'Klipper Firmware Configuration' menu. The 'CAN bus interface (CAN bus (on PD12/PD13))' option is highlighted in blue. The right window shows the compilation process, listing various object files being compiled, such as 'crc16_ccitt.o', 'armcm_boot.o', 'armcm_irq.o', 'armcm_reset.o', 'timer_irq.o', 'stm32f0_timer.o', 'stm32g0.o', 'gpio_periph.o', 'stm32f0_adc.o', 'stm32f0_i2c.o', 'spi.o', 'usbfs.o', 'canserial.o', 'fasthash.o', 'fdcan.o', 'chipid.o', 'usb_canbus.o', and 'hard_pwm.o'. It also shows the building of 'compile_time_request.o', preprocessing of 'armcm_link.ld', linking of 'klipper.elf', and the creation of the final hex file 'klipper.bin'.

```
biqu@CaribouManta: ~/klipper
(Top)
Klipper Firmware Configuration
[*] Enable extra low-level configuration options
  Micro-controller Architecture (STMicroelectronics STM32) --->
  Processor model (STM32G0B1) --->
  Bootloader offset (8KiB bootloader) --->
  Clock Reference (8 MHz crystal) --->
  Communication interface (USB to CAN bus bridge (USB on PA11/PA12)) --->
  CAN bus interface (CAN bus (on PD12/PD13)) --->
  USB ids --->
  (250000) CAN bus speed
  ( ) GPIO pins to set at micro-controller startup

[Space/Enter] Toggle/enter    [?] Help    [/] Search
[Q] Quit (prompts for save)   [ESC] Leave menu

biqu@CaribouManta: ~/klipper
Compiling out/src/generic/crc16_ccitt.o
Compiling out/src/generic/armcm_boot.o
Compiling out/src/generic/armcm_irq.o
Compiling out/src/generic/armcm_reset.o
Compiling out/src/generic/timer_irq.o
Compiling out/src/stm32/stm32f0_timer.o
Compiling out/src/stm32/stm32g0.o
Compiling out/src/stm32/gpio_periph.o
Compiling out/src/stm32/stm32f0_adc.o
Compiling out/src/stm32/stm32f0_i2c.o
Compiling out/src/stm32/spi.o
Compiling out/src/stm32/usbfs.o
Compiling out/src/generic/canserial.o
Compiling out/src/./lib/fast-hash/fasthash.o
Compiling out/src/stm32/fdcan.o
Compiling out/src/stm32/chipid.o
Compiling out/src/generic/usb_canbus.o
Compiling out/src/stm32/hard_pwm.o
Building out/compile_time_request.o
Version: v0.11.0-175-gcball9db
Preprocessing out/src/generic/armcm_link.ld
Linking out/klipper.elf
Creating hex file out/klipper.bin
biqu@CaribouManta:~/klipper$
```

- `cd ../klipper`
- `make clean`
- `make menuconfig`
- `make`

Step 3

```

biqus@CaribouManta: ~/klipper
Compiling out/src/stm32/usbfs.o
Compiling out/src/generic/canserial.o
Compiling out/src/..lib/fast-hash/fasthash.o
Compiling out/src/stm32/fdcan.o
Compiling out/src/stm32/chipid.o
Compiling out/src/generic/nsh_cambus.o
Compiling out/src/stm32/hard_pwm.o
Building out/compile_time_request.o
Version: v0.11.0-175-gcbal19db
Preprocessing out/src/generic/asmm_link.ld
Linking out/klipper.elf
Creating hex file out/klipper.bin
biqus@CaribouManta:~/klipper$ lsusb
Bus 008 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 004 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 007 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 006 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 002 Device 004: ID 0483:df11 STMicroelectronics STM Device in DFU Mode
Bus 002 Device 002: ID 1a40:0101 Terminus Technology Inc. Hub
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 005 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
biqus@CaribouManta:~/klipper$

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 = dfuIDLE, status = 0
dfuIDLE, continuing
DFU mode device DFU version 011a
Device returned transfer size 1024
DfuSe interface name: "Internal Flash"
Performing mass erase, this can take a moment
Downloading to address = 0x08000000, size = 4280
Download [=====] 100% 4280 bytes
Download done.
File downloaded successfully
biqus@CaribouManta:~/klipper$

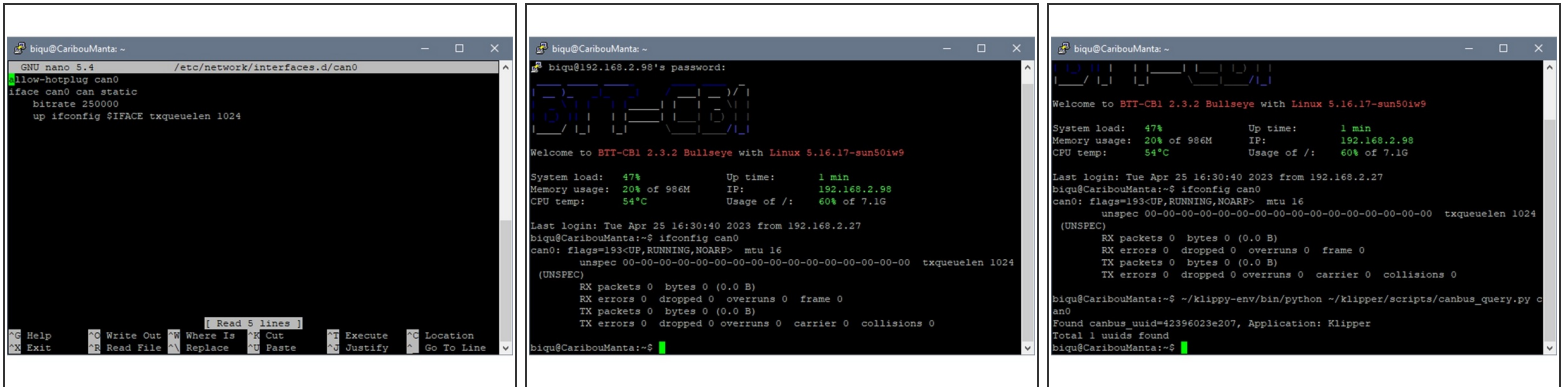
Copyright 2005-2009 Weston Schmidt, Harald Welte and OpenMoko Inc.
Copyright 2010-2016 Tormod Volden and Stefan Schmidt
This program is Free Software and has ABSOLUTELY NO WARRANTY
Please report bugs to http://sourceforge.net/p/dfu-util/tickets/

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 = dfuIDLE, status = 0
dfuIDLE, continuing
DFU mode device DFU version 011a
Device returned transfer size 1024
DfuSe interface name: "Internal Flash"
Downloading to address = 0x08002000, size = 28736
Download [=====] 100% 28736 bytes
Download done.
File downloaded successfully
dfu-util: Error during download get_status
biqus@CaribouManta:~/klipper$

```

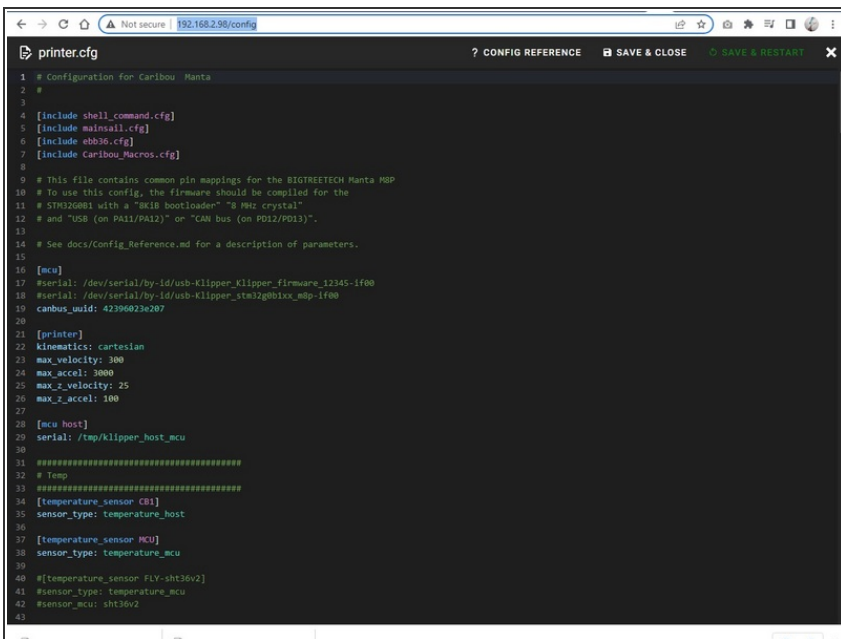
- keep boot pressed, press reset, release reset, release boot
- lsusb (board is in DFU mode)
- `sudo dfu-util -a 0 -D ~/CanBoot/out/canboot.bin --dfuse-address 0x08000000:force:mass-erase -d 0483:df11`
- `sudo dfu-util -a 0 -d 0483:df11 --dfuse-address 0x08002000:force:leave -D ~/klipper/out/klipper.bin`

Step 4



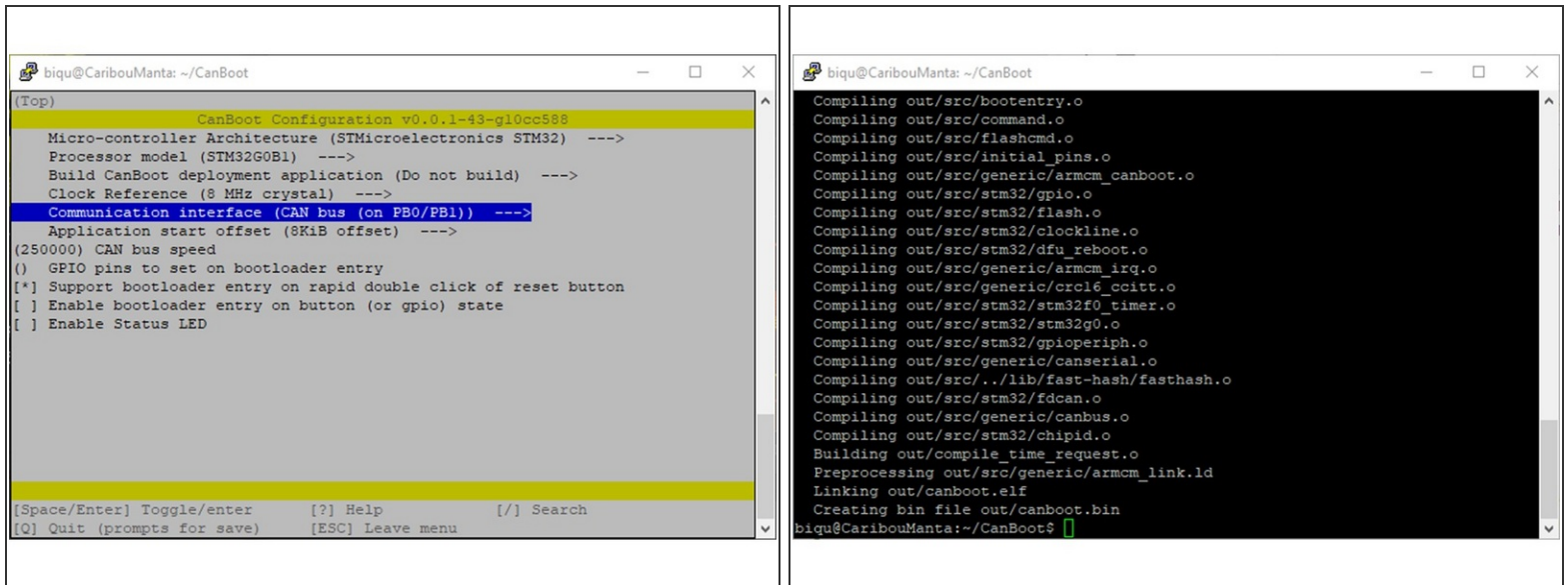
- `sudo nano /etc/network/interfaces.d/can0`
- `allow-hotplug can0` `iface can0 can static` `brtate 250000` `up ifconfig $IFACE txqueuelen 1024`
- `reboot`
- `ifconfig can0`
- `~/klippy-env/bin/python ~/klipper/scripts/canbus_query.py can0`

Step 5



- insert the canbus-id into printer.cfg

Step 6 — Preparing Canboot-Software for EBB36



The image shows two terminal windows side-by-side. The left window displays the CanBoot Configuration menu, and the right window shows the compilation process.

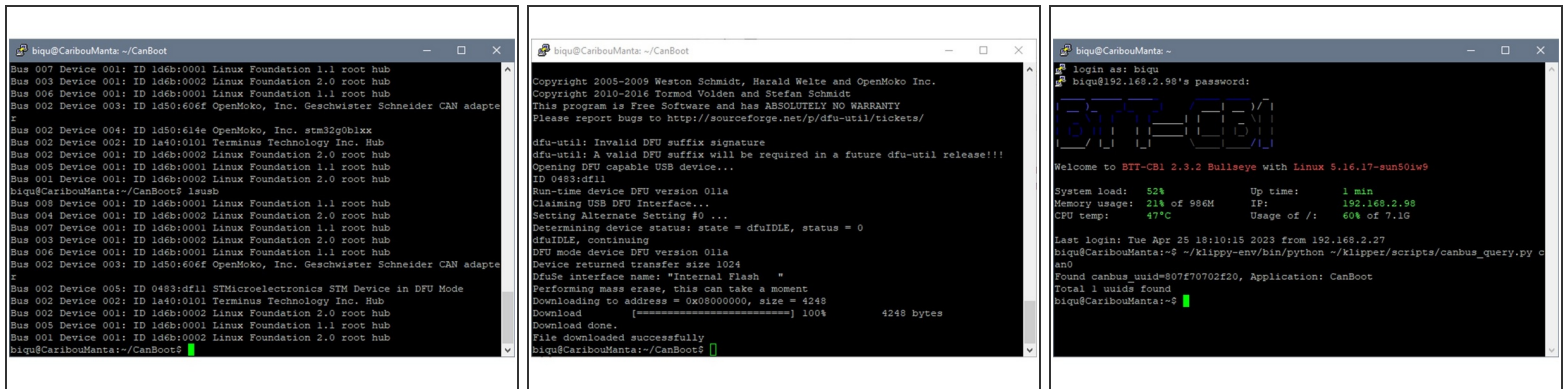
```
biqu@CaribouManta: ~/CanBoot
(Top)
CanBoot Configuration v0.0.1-43-g10cc588
Micro-controller Architecture (STMicroelectronics STM32) --->
Processor model (STM32G0B1) --->
Build CanBoot deployment application (Do not build) --->
Clock Reference (8 MHz crystal) --->
Communication interface (CAN bus (on PB0/PB1)) --->
Application start offset (8KiB offset) --->
(250000) CAN bus speed
() GPIO pins to set on bootloader entry
[*] Support bootloader entry on rapid double click of reset button
[ ] Enable bootloader entry on button (or gpio) state
[ ] Enable Status LED

[Space/Enter] Toggle/enter    [?] Help    [/] Search
[Q] Quit (prompts for save)    [ESC] Leave menu

biqu@CaribouManta: ~/CanBoot
Compiling out/src/bootentry.o
Compiling out/src/command.o
Compiling out/src/flashcmd.o
Compiling out/src/initial_pins.o
Compiling out/src/generic/armcm_canboot.o
Compiling out/src/stm32/gpio.o
Compiling out/src/stm32/flash.o
Compiling out/src/stm32/clockline.o
Compiling out/src/stm32/dfu_reboot.o
Compiling out/src/generic/armcm_irq.o
Compiling out/src/generic/crc16_ccitt.o
Compiling out/src/stm32/stm32f0_timer.o
Compiling out/src/stm32/stm32g0.o
Compiling out/src/stm32/gpioperiph.o
Compiling out/src/generic/canserial.o
Compiling out/src/./lib/fast-hash/fasthash.o
Compiling out/src/stm32/fdcan.o
Compiling out/src/generic/canbus.o
Compiling out/src/stm32/chipid.o
Building out/compile_time_request.o
Preprocessing out/src/generic/armcm_link.ld
Linking out/canboot.elf
Creating bin file out/canboot.bin
biqu@CaribouManta:~/CanBoot$
```

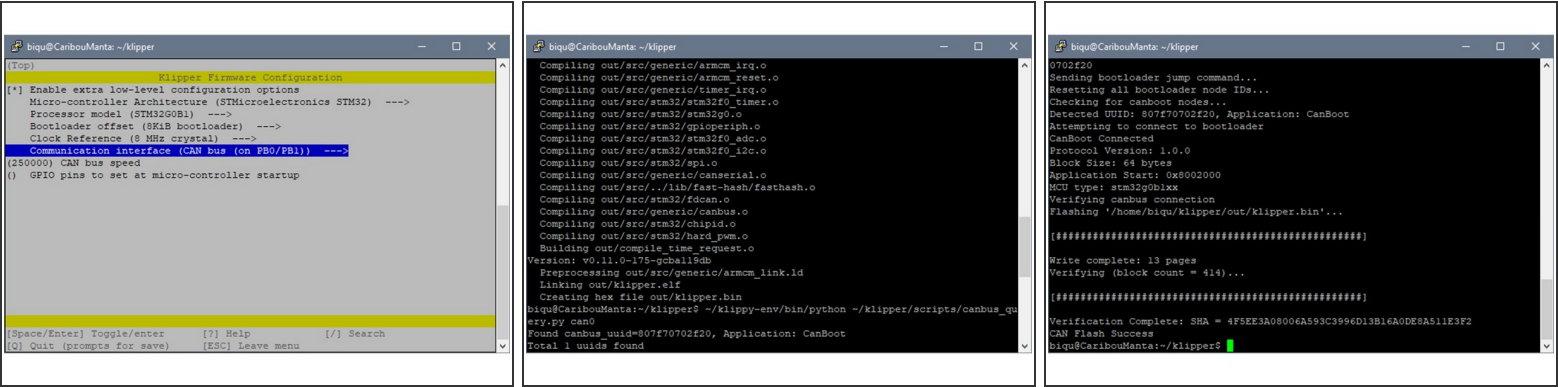
- `cd ~/CanBoot`
- `make clean`
- `make menuconfig`
- `make`

Step 7 — EBB36



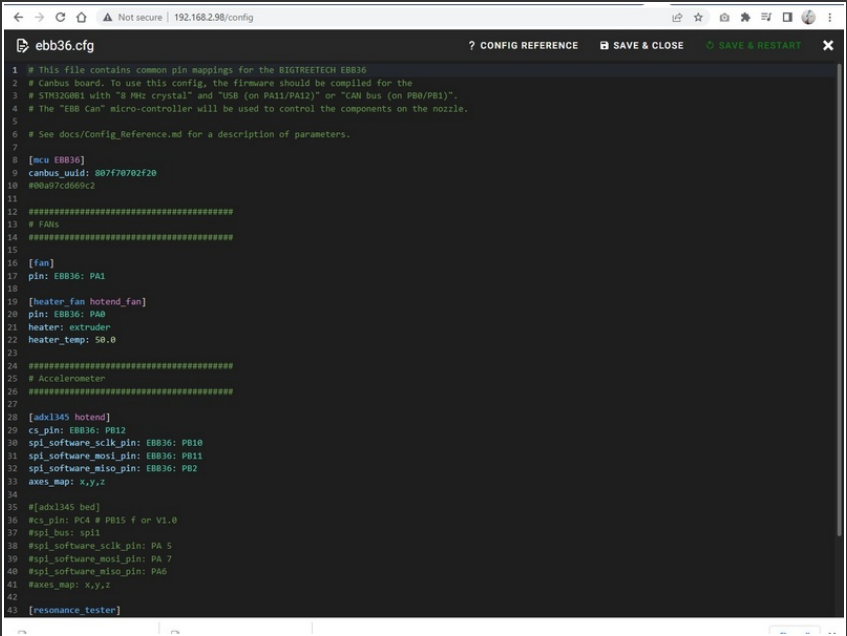
- keep boot pressed, press reset, release reset, release boot
- lsusb
- `sudo dfu-util -a 0 -D ~/CanBoot/out/canboot.bin --dfuse-address 0x08000000:force:mass-erase -d 0483:df11`
- disconnect USB
- connect EBB36 via Canbus connector and 24V
- connect Manta to 24V
- `~/klippy-env/bin/python ~/klipper/scripts/canbus_query.py can0`

Step 8



- `cd ../klipper`
- `make clean`
- `make menuconfig`
- `make`
- `python3 ~/klipper/lib/canboot/flash_can.py -u 807f70702f20`

Step 9



- Insert wisdom here.

Step 10 — Resources



- Manta part: https://www.youtube.com/watch?v=ekbxtDS_...
- EBB part: <https://www.youtube.com/watch?v=jgE3XMM9...>