Notes from MercuryOne.1 Build

PRE-REQUISITES:

- Manta E3EZ board installed
- 120R CANbus Jumper installed on mobo and is installed on EBB SB 2240
- Klipper service has been stopped (sudo service Klipper stop)

HARDWARE:

1. Assemble the machine according to the Zero G documentation
2. Assemble the Voron Stealthburner ClockWork2 Extruder with E3v6 Revo

SOFTWARE: Configure CANbus for Stealthburner EBB SB 2240/2209 with Manta e3ez

1. SSH into CB1

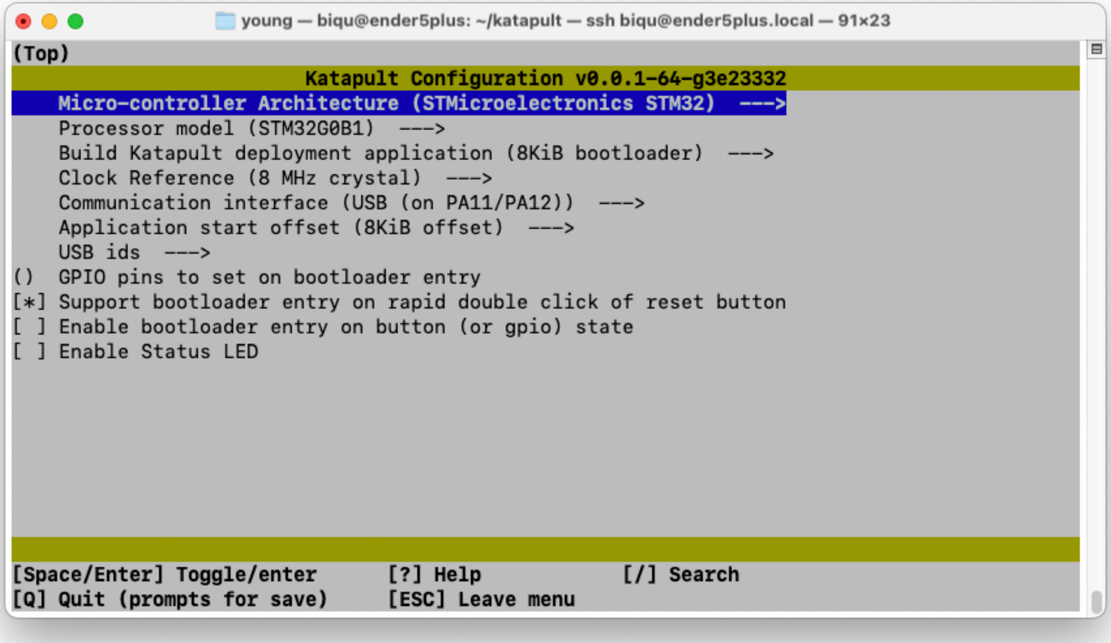
```
$ ssh biqu@ender5plus.local
```

2. Update the os and instal tools

```
biqu@ender5plus:~$ sudo apt update
biqu@ender5plus:~$ sudo apt upgrade
biqu@ender5plus:~$ sudo apt install python3 python3-pip python3-can
biqu@ender5plus:~$ pip3 install pyserial
biqu@ender5plus:~$ test -e ~/katapult && (cd ~/katapult && git pull) || (cd ~ && git clone https://github.com/Arksine/katapult) ; cd ~
```

3. Build the catapult firmware for the Manta e3ez mainboard

```
biqu@ender5plus:~$ cd ~/katapult
biqu@ender5plus:~$ make clean
biqu@ender5plus:~$ make menuconfig
```



young — biqu@ender5plus: ~/katapult — ssh biqu@ender5plus.local — 91×23

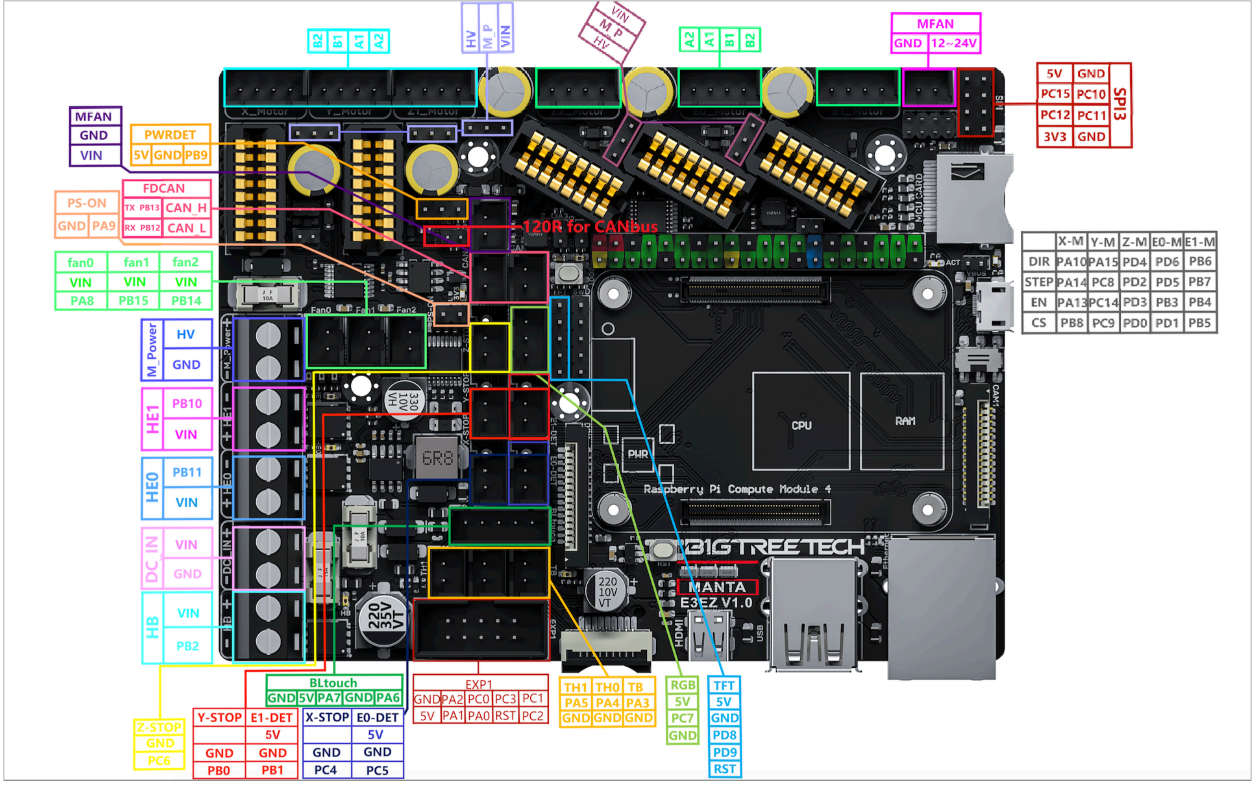
```
(Top)
                         Katapult Configuration v0.0.1-64-g3e23332
    Micro-controller Architecture (STMicroelectronics STM32)  --->
    Processor model (STM32G0B1)  --->
    Build Katapult deployment application (8KiB bootloader)  --->
    Clock Reference (8 MHz crystal)  --->
    Communication interface (USB (on PA11/PA12))  --->
    Application start offset (8KiB offset)  --->
    USB ids  --->
()  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
```

Hit “q” then “Y” to save.

4. Put the mainboard into DFU mode by pressing and holding down the “boot” button and pressing and releasing the “RST” button



	X-M	Y-M	Z-M	E0-M	E1-M
DIR	PA10	PA15	PD4	PD6	PB6
STEP	PA14	PC8	PD2	PD5	PB7
EN	PA13	PC14	PD3	PB3	PB4
CS	PB8	PC9	PD0	PD1	PB5

5. Check to see that the Pi can see the mainboard in DFU mode by running the following command:

```
biqu@ender5plus:~$ sudo dfu-util -l
du-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/

Found DFU: [0483:df11] ver=0200, devnum=8, cfg=1, intf=0, path="2-1.1", alt=2,
name="@Internal Flash   /0x08000000/256*02Kg", serial="20653172574B"
Found DFU: [0483:df11] ver=0200, devnum=8, cfg=1, intf=0, path="2-1.1", alt=1,
name="@Internal Flash   /0x08000000/256*02Kg", serial="20653172574B"
Found DFU: [0483:df11] ver=0200, devnum=8, cfg=1, intf=0, path="2-1.1", alt=0,
name="@Internal Flash   /0x08000000/256*02Kg", serial="20653172574B"
biqu@ender5plus:~$
```

6. Build the katapult firmware

```
biqu@ender5plus:~$ cd katapult/
biqu@ender5plus:~/katapult$ make
  Building out/autoconf.h
  Compiling out/src/sched.o
  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/stm32/usbfs.o
  Compiling out/src/stm32/chipid.o
  Compiling out/src/generic/usb_cdc.o
  Building out/compile_time_request.o
  Preprocessing out/src/generic/armcm_link.ld
  Linking out/katapult.elf
  Creating bin file out/katapult.bin
  Creating legacy binary out/canboot.bin
  Compiling out/src/deployer.o
  Compiling out/src/generic/armcm_boot.o
  Compiling out/src/generic/armcm_reset.o
  Building out/deployer_ctr.o
  Compiling out/katapult_payload.o
  Preprocessing out/src/generic/armcm_deployer.ld
  Linking out/deployer.elf
  Creating hex file out/deployer.bin
```

```
biqu@ender5plus:~/katapult$
```

7. Write the katapult firmware you just built for the manta mainboard to the board by using the dfu-util

```
biqu@ender5plus:~/katapult$ sudo dfu-util -R -a 0 -s 0x08000000:leave -D ~/katapult/out/katapult.bin -d 0483:df11
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 = dfuIDLE, status = 0
dfuIDLE, continuing
DFU mode device DFU version 011a
Device returned transfer size 1024
DfuSe interface name: "Internal Flash   "
Downloading to address = 0x08000000, size = 4216
Download	[=========================] 100%         4216 bytes
Download done.
File downloaded successfully
dfu-util: Error during download get_status
biqu@ender5plus:~/katapult$
```

8. Reboot the Pi

```
biqu@ender5plus:~/katapult$ sudo reboot now
```

9. Get mainboard out of DFU mode by pressing the RESET button twice, quickly.
10. Stop Klipper service by running the following command:

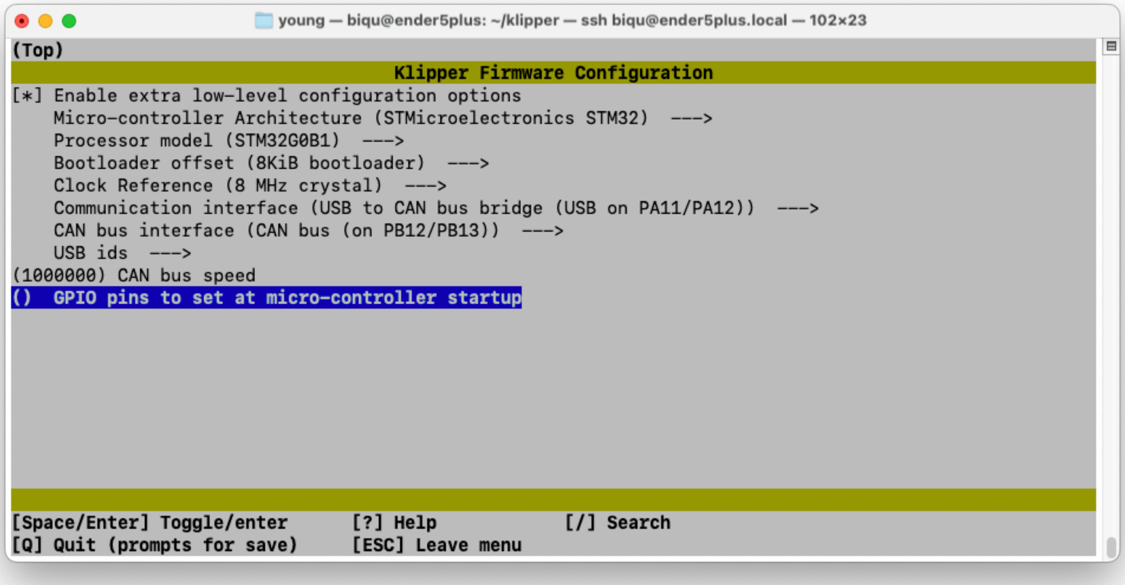
```
biqu@ender5plus:~$ sudo service klipper stop
```

11. Make sure the device attached shows up as a katapult device

```
biqu@ender5plus:~$ ls /dev/serial/by-id
usb-katapult_stm32g0b1xx_3D002C0002504B5735313920-if00
```

12. Make the Klipper firmware for the manta mainboard

```
biqu@ender5plus:~$ cd ~/klipper
biqu@ender5plus:~/klipper$ make clean
biqu@ender5plus:~/klipper$ make menuconfig
  Creating symbolic link out/board
Loaded configuration '/home/biqu/klipper/.config'
Configuration saved to '/home/biqu/klipper/.config'
```


young — biqu@ender5plus: ~/klipper — ssh biqu@ender5plus.local — 102×23

```
(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 PB12/PB13))  --->
    USB ids  --->
(1000000) 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
```

Hit “Q” then “Y” to save.

13. Make the firmware using make

```
biqu@ender5plus:~/klipper$ make
  Creating symbolic link out/board
  Building out/autoconf.h
  Compiling out/src/sched.o
  Compiling out/src/command.o
  Compiling out/src/basecmd.o
  Compiling out/src/debugcmds.o
  Compiling out/src/initial_pins.o
  Compiling out/src/gpiocmds.o
  Compiling out/src/stepper.o
  Compiling out/src/endstop.o
  Compiling out/src/trsync.o
  Compiling out/src/adccmds.o
  Compiling out/src/spicmds.o
  Compiling out/src/i2ccmds.o
  Compiling out/src/pwmcmds.o
  Compiling out/src/buttons.o
  Compiling out/src/tmcuart.o
  Compiling out/src/neopixel.o
  Compiling out/src/pulse_counter.o
  Compiling out/src/lcd_st7920.o
  Compiling out/src/lcd_hd44780.o
  Compiling out/src/spi_software.o
  Compiling out/src/i2c_software.o
  Compiling out/src/thermocouple.o
  Compiling out/src/sensor_adxl345.o
  Compiling out/src/sensor_angle.o
  Compiling out/src/sensor_mpu9250.o
  Compiling out/src/sensor_lis2dw.o
  Compiling out/src/sensor_bulk.o
  Compiling out/src/stm32/watchdog.o
  Compiling out/src/stm32/gpio.o
  Compiling out/src/stm32/clockline.o
  Compiling out/src/stm32/dfu_reboot.o
  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/gpioperiph.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: v0.12.0-114-ga77d0790
  Preprocessing out/src/generic/armcm_link.ld
  Linking out/klipper.elf
  Creating hex file out/klipper.bin
biqu@ender5plus:~/klipper$
```

14. Install the firwmware onto the Manta mainboard

```
biqu@ender5plus:~/klipper$ ls /dev/serial/by-id
usb-katapult_stm32g0b1xx_3D002C0002504B5735313920-if00
biqu@ender5plus:~/klipper$ python3 ~/katapult/scripts/flashtool.py -f ~/klipper/out/
klipper.bin -d /dev/serial/by-id/usb-katapult_stm32g0b1xx_3D002C0002504B5735313920-if00
Attempting to connect to bootloader
Katapult Connected
Protocol Version: 1.0.0
Block Size: 64 bytes
Application Start: 0x8002000
MCU type: stm32g0b1xx
Flashing '/home/biqu/klipper/out/klipper.bin'...

[##################################################]

Write complete: 15 pages
Verifying (block count = 470)...

[##################################################]

Verification Complete: SHA = EA619206A98977E59A400FA9F255273560926761
Flash Success
biqu@ender5plus:~/klipper$
```

15. Ensure Klipper is installed on the Manta mainboard

```
biqu@ender5plus:~/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 04d9:8030 Holtek Semiconductor, Inc. BTT-HDMI7
Bus 002 Device 007: ID 1d50:606f OpenMoko, Inc. Geschwister Schneider CAN adapter
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
```

16. Ensure the can bus is up

```
biqu@ender5plus:~/klipper$ ip -s -d link show can0
4: can0: <NOARP,UP,LOWER_UP,ECHO> mtu 16 qdisc pfifo_fast state UP mode DEFAULT
group default qlen 1024
    link/can  promiscuity 0 minmtu 0 maxmtu 0
    can state ERROR-ACTIVE restart-ms 0
          bitrate 1000000 sample-point 0.750
          tq 62 prop-seg 5 phase-seg1 6 phase-seg2 4 sjw 1
          gs_usb: tseg1 1..16 tseg2 1..8 sjw 1..4 brp 1..1024 brp-inc 1
          clock 48000000
          re-started bus-errors arbit-lost error-warn error-pass bus-off
          0          0          0          0          0          0         numtxqueues 1 numrxqueues 1
gso_max_size 65536 gso_max_segs 65535
    RX: bytes  packets  errors  dropped missed  mcast
    0          0        0       0       0       0
    TX: bytes  packets  errors  dropped carrier collsns
    0          0        0       0       0       0
```

17. Run the Klipper CANbus query to retrieve the canbus_UUID of the manta mainboard

```
biqu@ender5plus:~/klipper$ ~/klippy-env/bin/python ~/klipper/scripts/canbus_query.py can0

Found canbus_uuid=b57a742e4b29, Application: Klipper
Total 1 uuids found
biqu@ender5plus:~/klipper$
```

18. Start the klipper service back up

```
biqu@ender5plus:~/klipper$ sudo service klipper start
biqu@ender5plus:~/klipper$
```

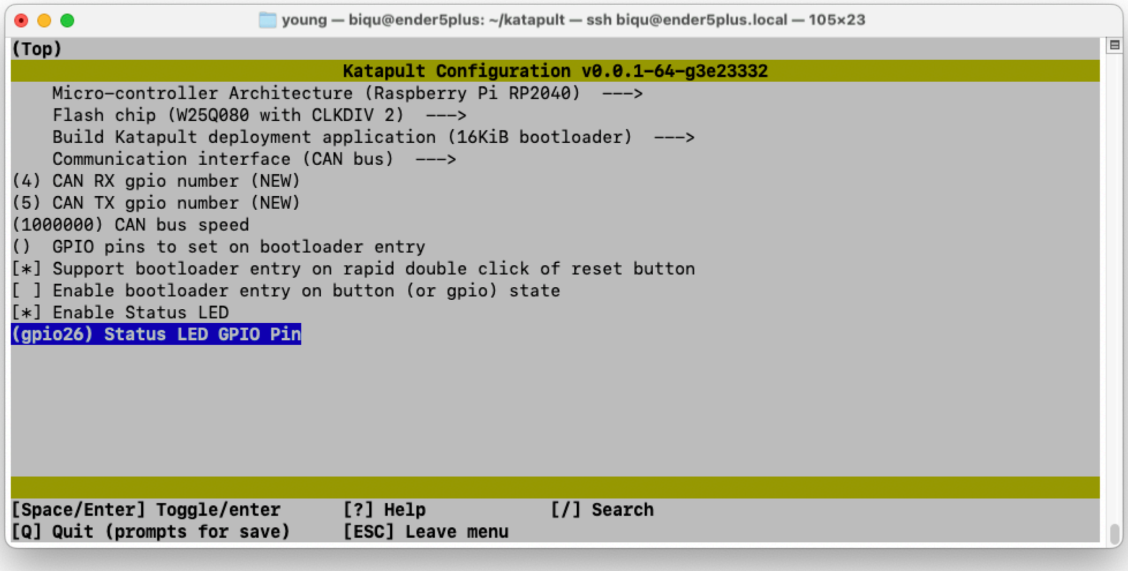
Install firmware on the EBB SB2240 RP2040

1. Create the katapult firmware for the RP2040
2. SSH into the raspberry PI

```
$ ssh biqu@ender5plus.local
```

3. Run the following commands to build the firmware using katapult

```
biqu@ender5plus:~$ cd katapult/
biqu@ender5plus:~/katapult$ make clean
biqu@ender5plus:~/katapult$ make menuconfig
```



```
young — biqu@ender5plus: ~/katapult — ssh biqu@ender5plus.local — 105×23
(Top)
                               Katapult Configuration v0.0.1-64-g3e23332
    Micro-controller Architecture (Raspberry Pi RP2040)  --->
    Flash chip (W25Q080 with CLKDIV 2)  --->
    Build Katapult deployment application (16KiB bootloader)  --->
    Communication interface (CAN bus)  --->
(4) CAN RX gpio number (NEW)
(5) CAN TX gpio number (NEW)
(1000000) 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
(gpio26) Status LED GPIO Pin



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

Hit "Q" then "Y"

4. Build the firmware using make

```
biqu@ender5plus:~/katapult$ make
  Creating symbolic link out/board
  Building out/autoconf.h
  Compiling out/src/sched.o
  Compiling out/src/bootentry.o
  Compiling out/src/command.o
  Compiling out/src/flashcmd.o
  Compiling out/src/initial_pins.o
  Compiling out/src/led.o
  Compiling out/src/rp2040/armcm_canboot.o
  Compiling out/src/rp2040/main.o
  Compiling out/src/rp2040/gpio.o
  Compiling out/src/rp2040/timer.o
  Compiling out/src/rp2040/flash.o
  Compiling out/src/../lib/rp2040/pico/flash/hw_flash.o
  Compiling out/src/generic/armcm_irq.o
  Compiling out/src/generic/crc16_ccitt.o
  Compiling out/src/rp2040/can.o
  Compiling out/src/rp2040/chipid.o
  Compiling out/src/../lib/can2040/can2040.o
  Compiling out/src/generic/canserial.o
  Compiling out/src/generic/canbus.o
  Compiling out/src/../lib/fast-hash/fasthash.o
  Building out/compile_time_request.o
  Building rp2040 stage2 out/stage2.o
  Preprocessing out/src/rp2040/rp2040_link.ld
  Linking out/katapult.elf
  Creating bin file out/katapult.bin
  Creating legacy binary out/canboot.bin
  Building out/lib/rp2040/elf2uf2/elf2uf2
  Creating uf2 file out/katapult.uf2
  Creating legacy uf2 file out/canboot.uf2
  Compiling out/src/deployer.o
  Compiling out/src/generic/armcm_boot.o
  Compiling out/src/generic/armcm_reset.o
  Building out/deployer_ctr.o
  Compiling out/katapult_payload.o
  Preprocessing out/src/generic/armcm_deployer.ld
  Linking out/deployer.elf
  Creating hex file out/deployer.bin
biqu@ender5plus:~/katapult$
```

5. Connect the PI to the RP2040 board and make sure you insert the jumper to power the board from the USB
6. Confirm the board is in BOOT mode

```
biqu@ender5plus:~$ 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 008: ID 2e8a:0003 Raspberry Pi RP2 Boot
Bus 002 Device 004: ID 04d9:8030 Holtek Semiconductor, Inc. BTT-HDMI7
Bus 002 Device 007: ID 1d50:606f OpenMoko, Inc. Geschwister Schneider CAN adapter
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
biqu@ender5plus:~$
```

7. Flash the katapult firmware to the RP2040

```
biqu@ender5plus:~$ cd katapult/
biqu@ender5plus:~/katapult$ make flash FLASH_DEVICE=2e8a:0003
  Building rp2040_flash
gcc -c -Wall -ggdb -I../rp2040/ `pkg-config libusb-1.0 --cflags` main.c -o main.o
gcc -c -Wall -ggdb -I../rp2040/ `pkg-config libusb-1.0 --cflags` picoboot_connection.c -o
picoboot_connection.o
gcc  main.o picoboot_connection.o `pkg-config libusb-1.0 --libs` -o rp2040_flash
  Flashing out/katapult.uf2
Loaded UF2 image with 31 pages
Found rp2040 device on USB bus 2 address 8
Flashing...
Resetting interface
Locking
Exiting XIP mode
Erasing
Flashing
Rebooting device
biqu@ender5plus:~/katapult$
```

8. Shutdown the PI and then shutdown the printer.

```
biqu@ender5plus:~/katapult$ sudo shutdown now
Connection to ender5plus.local closed by remote host.
```

Connection to ender5plus.local closed.

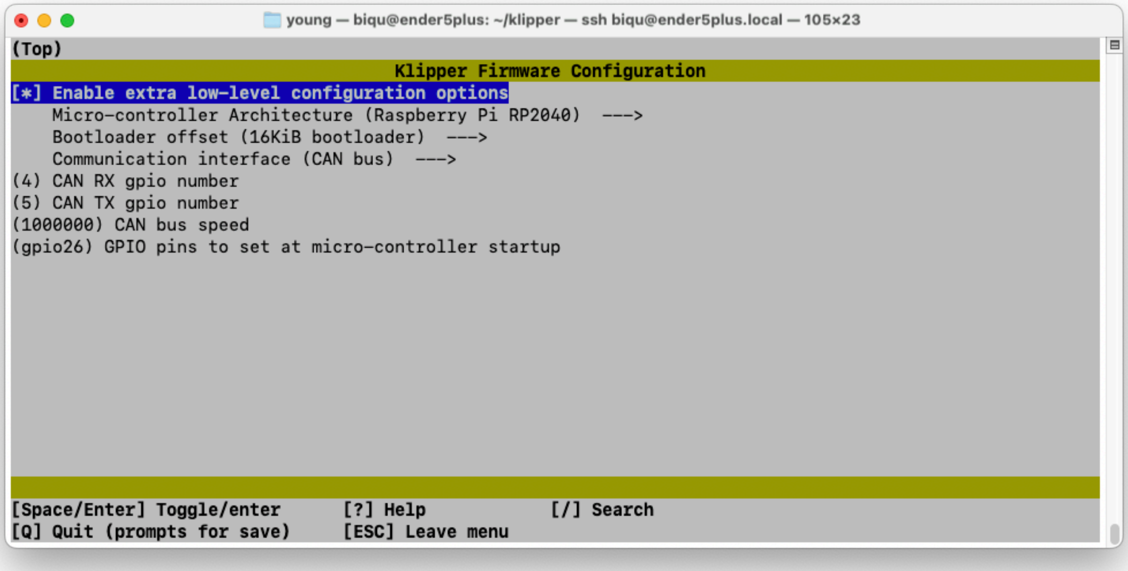
Connecting up the CAN cable

1. Go to the printer and remove the jumper for the USB power from the RP2040.
2. Install the CANbus cable into the RP2040 and wire the CANH and CANL to the ports on the manta respectively.
3. Power on the printer
4. SSH into the PI

```
biqu@ender5plus:~$ python3 ~/katapult/scripts/flashtool.py -i can0 -q
Resetting all bootloader node IDs...
Checking for Katapult nodes...
Detected UUID: b57a742e4b29, Application: Klipper
Detected UUID: 9bba08420308, Application: Katapult
Query Complete
biqu@ender5plus:~$
```

5. Make the klipper firmware for the RP2040

```
biqu@ender5plus:~$ cd klipper/
biqu@ender5plus:~/klipper$ make clean
biqu@ender5plus:~/klipper$ make menuconfig
  Creating symbolic link out/board
Loaded configuration '/home/biqu/klipper/.config'
Configuration saved to '/home/biqu/klipper/.config'
```



young — biqu@ender5plus: ~/klipper — ssh biqu@ender5plus.local — 105×23

```
(Top)
                                  Klipper Firmware Configuration
[*] Enable extra low-level configuration options
    Micro-controller Architecture (Raspberry Pi RP2040)  --->
    Bootloader offset (16KiB bootloader)  --->
    Communication interface (CAN bus)  --->
(4) CAN RX gpio number
(5) CAN TX gpio number
(1000000) CAN bus speed
(gpio26) GPIO pins to set at micro-controller startup

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

Hit “Q” and then “Y” to save

6. Stop the klipper service

```
biqu@ender5plus:~/klipper$ sudo service klipper stop
biqu@ender5plus:~/klipper$
```

7. Query the CANbus to get the UUID

```
biqu@ender5plus:~/klipper$ python3 ~/katapult/scripts/flashtool.py -i can0 -q
Resetting all bootloader node IDs...
```

```
Checking for Katapult nodes...
Detected UUID: b57a742e4b29, Application: Klipper
Detected UUID: 9bba08420308, Application: Katapult
Query Complete
biqu@ender5plus:~/klipper$
```

8. Flash the RP2040 with klipper

```
biqu@ender5plus:~/klipper$ python3 ~/katapult/scripts/flashtool.py -i can0 -u 9bba08420308
-f ~/klipper/out/klipper.bin
Sending bootloader jump command...
Resetting all bootloader node IDs...
Attempting to connect to bootloader
Katapult Connected
Protocol Version: 1.0.0
Block Size: 64 bytes
Application Start: 0x10004000
MCU type: rp2040
Verifying canbus connection
Flashing '/home/biqu/klipper/out/klipper.bin'...

[##################################################]

Write complete: 116 pages
Verifying (block count = 464)...

[##################################################]

Verification Complete: SHA = 53DFCDB852398DCA99B9BD1356B7427112518345
Flash Success
biqu@ender5plus:~/klipper$
```

9. Confirm flash has been completed

```
biqu@ender5plus:~/klipper$ python3 ~/katapult/scripts/flashtool.py -i can0 -q
Resetting all bootloader node IDs...
Checking for Katapult nodes...
Detected UUID: b57a742e4b29, Application: Klipper
Detected UUID: 9bba08420308, Application: Klipper
Query Complete
biqu@ender5plus:~/klipper$ ~/klippy-env/bin/python ~/klipper/scripts/canbus_query.py can0
Found canbus_uuid=b57a742e4b29, Application: Klipper
Found canbus_uuid=9bba08420308, Application: Klipper
Total 2 uuids found
biqu@ender5plus:~/klipper$ sudo service klipper start
```

10. Update the klipper configurations accordingly: printer.cfg, etc.