

Мануал рассчитывает на некоторые базовые знания, как перевести октопус в дфу. Как нажать Y если система того просит и т.д. Я привел два примера под октопус на 446 и на 723м проце. Если у вас 723й и вы уверены в своем кабеле, то скорость CAN шины можете попробовать в 1000000. Если же 446й, то везде где дальше увидите строчку 1000000 меняйте на 500000. PS: все на свой страх и риск, автор данного текста никакой ответственности не несет.

1)Начало, обновляем систему и до устанавливаем пакеты

```
sudo apt update
```

```
sudo apt upgrade
```

```
sudo apt install python3 python3-pip python3-can
```

```
sudo pip3 install pyserial
```

2) Ставим катапульт

```
git clone https://github.com/Arksine/katapult
cd katapult
make clean
```

3) Создаем прошивки

```
make menuconfig
```

446 проц

```
(Top)
CanBoot Configuration v0.0.1-8-g98844ea
Micro-controller Architecture (STMicroelectronics STM32) --->
Processor model (STM32F446) --->
Build CanBoot deployment application (Do not build) --->
Clock Reference (12 MHz crystal) --->
Communication interface (USB (on PA11/PA12)) --->
Application start offset (32KiB 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
```

Если 723й

```
(Top)
Katapult Configuration v0.0.1-61-gec4df2e
Micro-controller Architecture (STMicroelectronics STM32) --->
Processor model (STM32H723) --->
Build Katapult deployment application (Do not build) --->
Clock Reference (25 MHz crystal) --->
Communication interface (USB (on PA11/PA12)) --->
Application start offset (128KiB 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
```

make

И переносим прошивку в другую папку

```
mkdir ~/firmware
mv ~/katapult/out/katapult.bin ~/firmware/octopus_katapult.bin
```

4) Вводим октопус в режим DFU и вводим команду *lsusb*

```
biqu@BTT-CB1:~/katapult$ 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 005: 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
biqu@BTT-CB1:~/katapult$
```

Видим устройство в DFU, копируем его ID, в моем случае это было **0483:df11** и вставляем в следующую команду

Прошиваем октопус следующей командой, подставив свои ID

```
sudo dfu-util -a 0 -D ~/firmware/octopus_katapult.bin --dfuse-address
0x08000000:force:mass-erase:leave -d 0483:df11
```

Проверяем командой *lsusb* и видим плату на STM723, все ок.

```
biqu@BTT-CB1:~/katapult$ 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 1d50:6177 OpenMoko, Inc. stm32h723xx
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@BTT-CB1:~/katapult$
```

5) Подключаем EBB36 в режиме DFU к PI через кабель USB , предварительно установим перемычку для питания от USB (микрофит отключен), проверяем также командой *lsusb*

```

biqu@BTT-CB1:~/katapult$ 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 1d50:6177 OpenMoko, Inc. stm32h723xx
Bus 002 Device 010: 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
biqu@BTT-CB1:~/katapult$

```

Видим наш октопус (у меня это stm723hxx) и опять устройство в DFU.

Собираем прошивку для EBB36

```

cd katapult
make clean
make menuconfig

```

```

(Top)
Katapult Configuration v0.0.1-61-gec4df2e
Micro-controller Architecture (STMicroelectronics STM32) --->
Processor model (STM32G0B1) --->
Build Katapult deployment application (Do not build) --->
Clock Reference (8 MHz crystal) --->
Communication interface (CAN bus (on PB0/PB1)) --->
Application start offset (8KiB offset) --->
(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

```

make

переносим прошивку и прошиваем аналогично октопусу

```
mv ~/katapult/out/katapult.bin ~/firmware/ebb_katapult.bin
```

и прошиваем ebb, подставив также ID уже ее. Если они отличные от моих.

```
sudo dfu-util -a 0 -D ~/firmware/ebb_katapult.bin --dfuse-address
0x08000000:force:mass-erase:leave -d 0483:df11
```

6) Создаем Can интерфейс

```
cd /
sudo nano /etc/network/interfaces.d/can0
```

откроется текстовый редактор, вставляем туда следующие строки и сохраняемся:

```
allow-hotplug can0
iface can0 can static
    bitrate 1000000
    up ifconfig $IFACE txqueuelen 1024
```

```
biqu@BTT-CB1: /  
GNU nano 5.4 /etc/network/interfaces.d/can0  
allow-hotplug can0  
iface can0 can static  
    bitrate 1000000  
    up ifconfig $IFACE txqueuelen 1024
```

Сохраняем файл

7) Собираем прошивку для режима кан бридж октопуса

Если вам нужно, чтобы при включении октопус поднимал пин PS_on, прописываем его на данном этапе

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

446

```
pi@klipper: ~/klipper  
(Top)  
Klipper Firmware Configuration  
[*] Enable extra low-level configuration options  
    Micro-controller Architecture (STMicroelectronics STM32) --->  
    Processor model (STM32F446) --->  
    Bootloader offset (32KiB bootloader) --->  
    Clock Reference (12 MHz crystal) --->  
    Communication interface (USB to CAN bus bridge (USB on PA11/PA12)) --->  
    CAN bus interface (CAN bus (on PD0/PD1)) --->  
    USB ids --->  
(500000) CAN bus speed  
( ) GPIO pins to set at micro-controller startup (NEW)  
[Space/Enter] Toggle/enter      [?] Help      [/] Search  
[Q] Quit (prompts for save)     [ESC] Leave menu
```

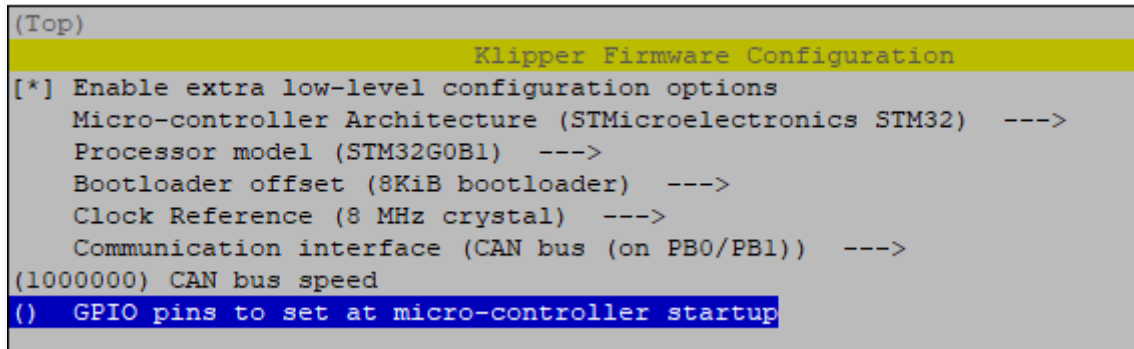
723

```
biqu@BTT-CB1: ~  
(Top)  
Klipper Firmware Configuration  
[*] Enable extra low-level configuration options  
    Micro-controller Architecture (STMicroelectronics STM32) --->  
    Processor model (STM32H723) --->  
    Bootloader offset (128KiB bootloader (SKR SE BX v2.0)) --->  
    Clock Reference (25 MHz crystal) --->  
    Communication interface (USB to CAN bus bridge (USB on PA11/PA12)) --->  
    CAN bus interface (CAN bus (on PD0/PD1)) --->  
    USB ids --->  
(1000000) CAN bus speed  
( ) GPIO pins to set at micro-controller startup
```

```
make
mv ~/klipper/out/klipper.bin ~/firmware/octopus_klipper.bin
```

8) Собираем прошивку для EBB

```
make clean
make menuconfig
```

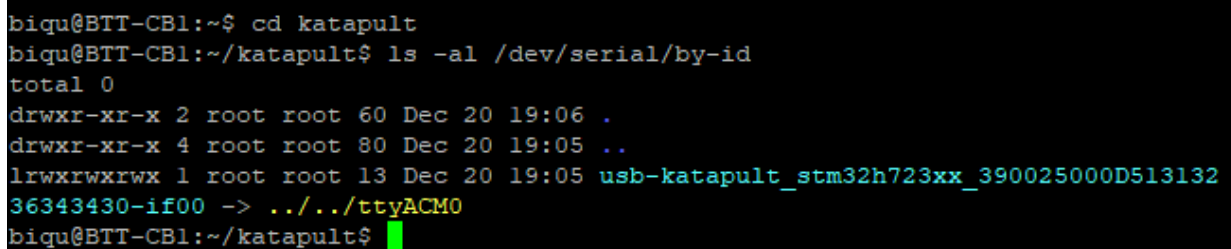


```
(Top)
Klipper Firmware Configuration
[*] Enable extra low-level configuration options
  Micro-controller Architecture (STMicroelectronics STM32) --->
  Processor model (STM32G0B1) --->
  Bootloader offset (8KiB bootloader) --->
  Clock Reference (8 MHz crystal) --->
  Communication interface (CAN bus (on PB0/PB1)) --->
  (1000000) CAN bus speed
  () GPIO pins to set at micro-controller startup
```

```
make
mv ~/klipper/out/klipper.bin ~/firmware/ebb_klipper.bin
```

8) Опрос устройств

```
ls -al /dev/serial/by-id
```



```
biqu@BTT-CB1:~$ cd katapult
biqu@BTT-CB1:~/katapult$ ls -al /dev/serial/by-id
total 0
drwxr-xr-x 2 root root 60 Dec 20 19:06 .
drwxr-xr-x 4 root root 80 Dec 20 19:05 ..
lrwxrwxrwx 1 root root 13 Dec 20 19:05 usb-katapult_stm32h723xx_390025000D513132
36343430-if00 -> ../../ttyACM0
biqu@BTT-CB1:~/katapult$
```

Видим строчку с октопусом.

```
cd ~/katapult/scripts
```

```
pip3 install pyserial
```

В следующую команду подставить свой Serial октопуса полученный на скрине выше.

```
python3 flash_can.py -f ~/firmware/octopus_klipper.bin -d /dev/serial/by-id/usb-
katapult_stm32h723xx_390025000D51313236343430-if00
```

```

biqu@BTT-CB1:~/katapult/scripts$ python3 flash_can.py -f ~/firmware/octopus_klipper.bin -d /dev/serial/
d/usb-katapult_stm32h723xx_390025000D51313236343430-if00
Attempting to connect to bootloader
Katapult Connected
Protocol Version: 1.0.0
Block Size: 64 bytes
Application Start: 0x8020000
MCU type: stm32h723xx
Flashing '/home/biqu/firmware/octopus_klipper.bin'...

[#####]

Write complete: 1 pages
Verifying (block count = 493)...

[#####]

Verification Complete: SHA = 77771CEC5F66DF8D2472D5ADC4762B2CDDA7AD90
Flash Success
biqu@BTT-CB1:~/katapult/scripts$ █

```

Октопус прошит, после чего выключаем пишку командой

```
sudo shutdown now
```

Отключаем питание. Подключаем EBB . не забываем перемычку 120ohm

9) Заново заходим в путти и выполняем команды (вводить сразу обе)

```
cd ~/katapult/scripts
python3 flash_can.py -i can0 -q
```

Если не работает, то команда

```
~/klippy-env/bin/python ~/klipper/scripts/canbus_query.py can0
```

Видим обе платы. Немного радуемся

```

biqu@BTT-CB1:~$ cd ~/katapult/scripts
python3 flash_can.py -i can0 -q
Resetting all bootloader node IDs...
Checking for Katapult nodes...
Detected UUID: ae4b3775ca3f, Application: Klipper
Detected UUID: 28c5348109be, Application: Katapult
Query Complete
biqu@BTT-CB1:~/katapult/scripts$ █

```

На данном этапе с пометкой Katapult . это EBB36. Берем ее uuid и вставляем в следующую команду и прошиваем ebb

```
python3 flash_can.py -i can0 -u 28c5348109be -f ~/firmware/ebb_klipper.bin
```

```

biqu@BTT-CB1:~/katapult/scripts$ ^C
biqu@BTT-CB1:~/katapult/scripts$ python3 flash_can.py -i can0 -u 28c5348109be -f
~/firmware/ebb_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: 0x8002000
MCU type: stm32g0blxx
Verifying canbus connection
Flashing '/home/biqu/firmware/ebb_klipper.bin'...

[#####]

Write complete: 14 pages
Verifying (block count = 448)...

[#####]

Verification Complete: SHA = 6A73D9182FF8A4377A1C71E581ABB92AA2DC4A67
Flash Success
biqu@BTT-CB1:~/katapult/scripts$ █

```

прошивка завершена

10)Вновь опрашиваем платы

python3 flash_can.py -i can0 -q

```

biqu@BTT-CB1:~/katapult/scripts$ python3 flash_can.py -i can0 -q
Resetting all bootloader node IDs...
Checking for Katapult nodes...
Detected UUID: ae4b3775ca3f, Application: Klipper
Detected UUID: 28c5348109be, Application: Klipper
█

```

копируем uuid в конфиги - и радуемся☺