

Crazyflie Push Demo Documentation

Objective

The objective of this procedure is to be able to successfully set up the crazyflie, mount the flow deck and multiranger deck, and flash the demo to the crazyflie.

Equipment Required

The equipment necessary for this project include

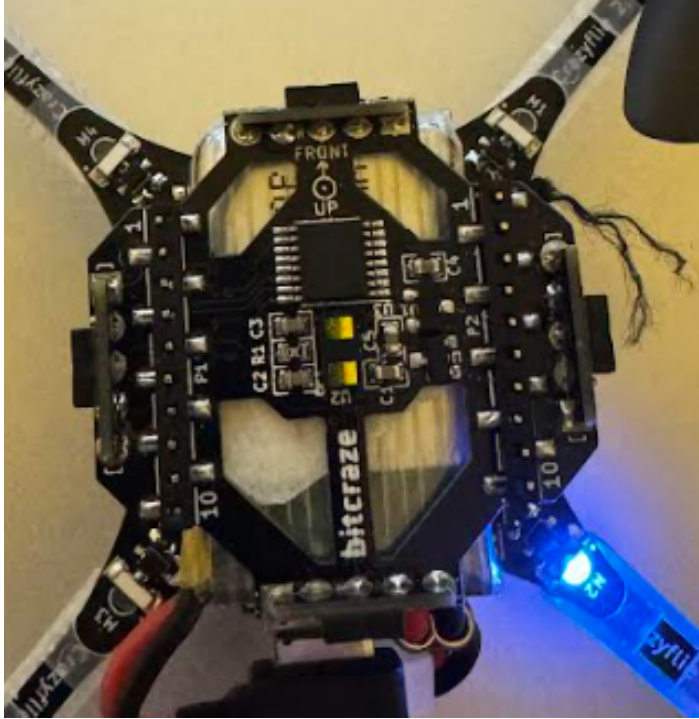
- Crazyflie 2.X
- Flow Deck V2
- Multiranger Deck
- Crazyradio PA

Notes

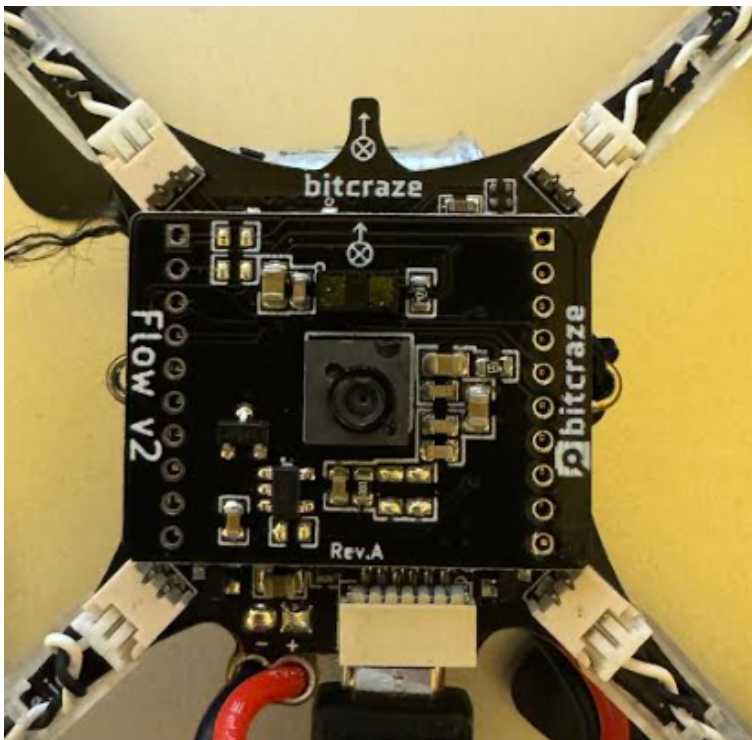
- The implementation of this project is conducted in a Windows 10/11 environment for the main computer, with a Ubuntu environment in VirtualBox
- If any problems occur, checking the console log in the Crazyflie Client will help determine any hardware issues with the decks
- It is recommended to update crazyflie firmware alongside the crazyradio firmware, but it is something that I personally did not do
- Multiranger deck replaces the battery holder
- Any extensive drifting that occurs during the demo can be a result of a faulty flow deck

Procedure

1. If the crazyflie being used is unassembled, follow the instructions at this link to assemble the crazyflie. The instructions can be followed all the way up to the power on step.
<https://www.bitcraze.io/documentation/tutorials/getting-started-with-crazyflie-2-x/>
2. The next step is to mount the flow deck V2 and multiranger deck. To do so, power the crazyflie on and determine the front and back. The back should be indicative of two blue LEDs. Using this information, align the pins of the multiranger deck onto the top of the crazyflie with the side facing up being the one that has the Front Up with a circle symbol. The symbol has an arrow pointing outward. Orient the multiranger deck so that the arrow pointing outward points towards the front of the crazyflie as seen below.



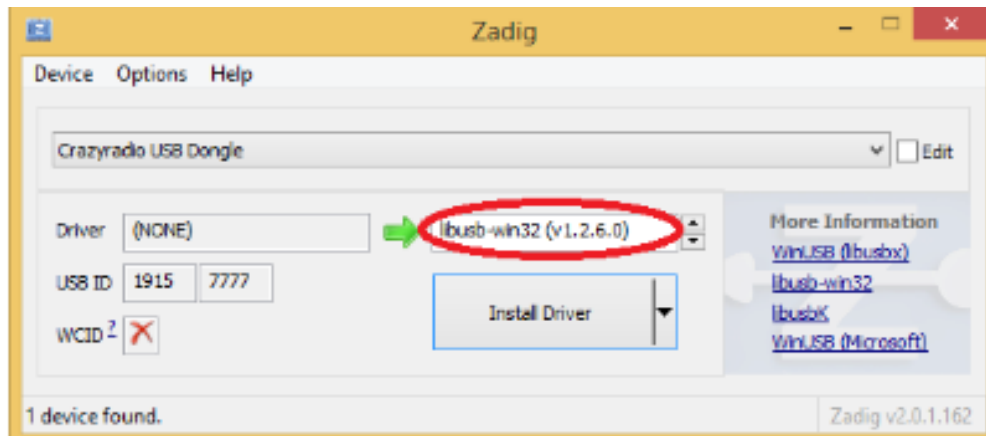
In mounting the flow deck, a similar method for mounting is implemented where the arrow facing outward should be pointing towards the front of the crazyflie. In this case, we want the side of the flow deck to face down has the circle symbol, but with an x through it as seen below.



For further assistance, please visit this link:

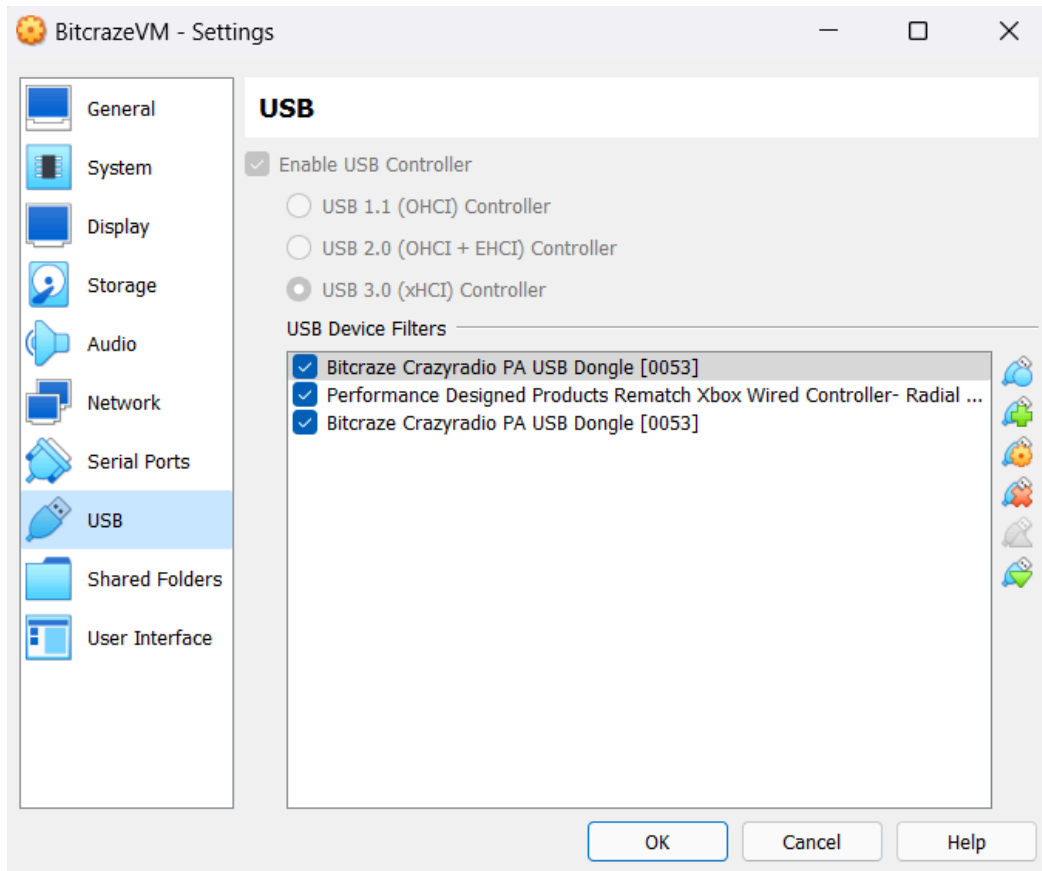
<https://www.bitcraze.io/documentation/tutorials/getting-started-with-expansion-decks/>

- Following the successful construction of the crazyflie, the next step is to download VirtualBox as this will be the platform which the Bitcraze VM will be ran on.
<https://www.oracle.com/virtualization/technologies/vm/downloads/virtualbox-downloads.html?msclkid=be59f68fc9f11ec801dcf4ea944dd1d>
- Next, download the latest version of the Bitcraze VM .ova file.
<https://github.com/bitcraze/bitcraze-vm/releases>
- Launch Virtual Box. Select File in the top left corner, Import Appliance. Browse your computer for the downloaded .ova file previously downloaded, click Next, then Finish.
- Before launching the actual VM in VirtualBox, let's set up the crazyradio drivers first. To do so, screw in the antenna for the crazyradio, plug into your machine, and download Zadig: <https://zadig.akeo.ie/>. Inserting the crazyradio dongle into your device for the first time should result in Windows conducting some setup process. Once complete, open Zadig, select the crazyradio device, select libusb, and click install.

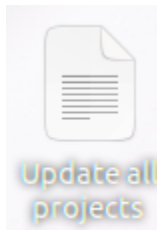


- With the crazyradio driver installed, next is to add the dongle to the device of USB filters so that the VM is able to detect and utilize dongle through your main machine. Open the main menu for VirtualBox and ensure the crazyradio is still inserted. Select the Bitcraze VM and select settings. Navigate to USB and add new USB filter. Select Bitcraze Crazyradio PA to add it to the list and OK to confirm. Your screen should look similar to the image below, however there only needs to be one instance of the device filter in your

case.

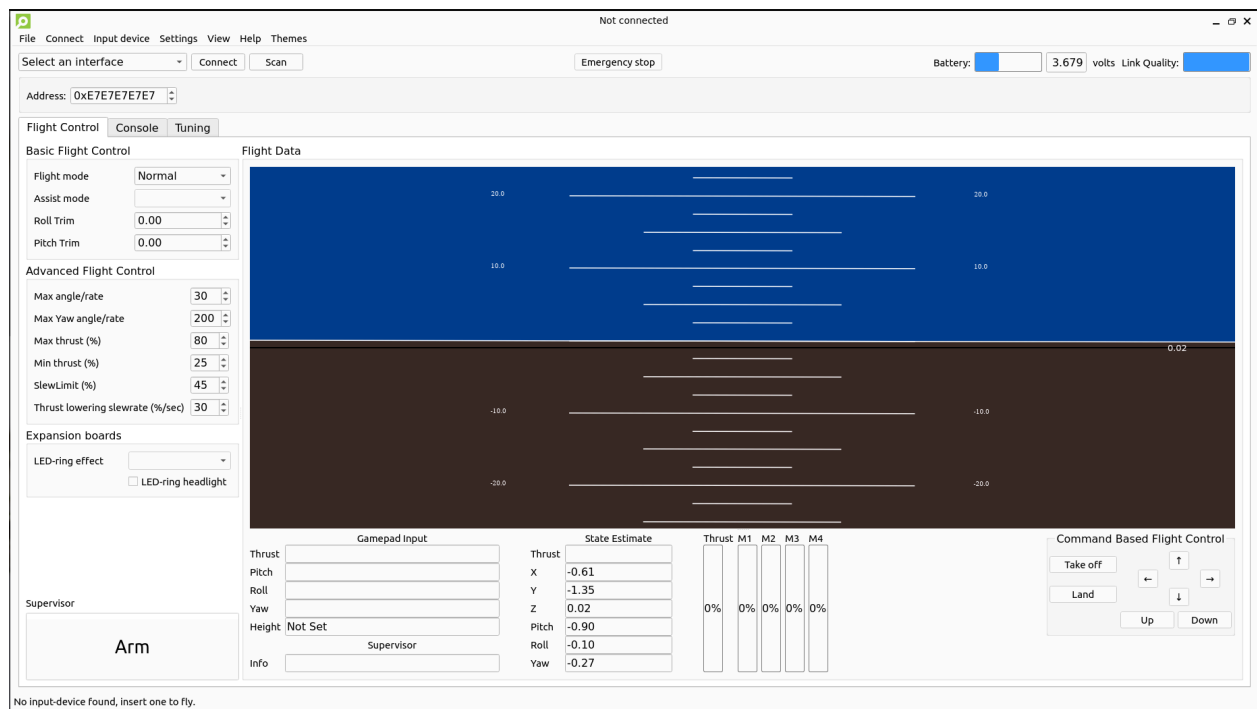


8. It's finally time to start the VM. Select the Bitcraze VM and press Start. First time startup should take a couple minutes. After finally entering the VM, the first thing that needs to be done is clicking the update all projects option on the right side.



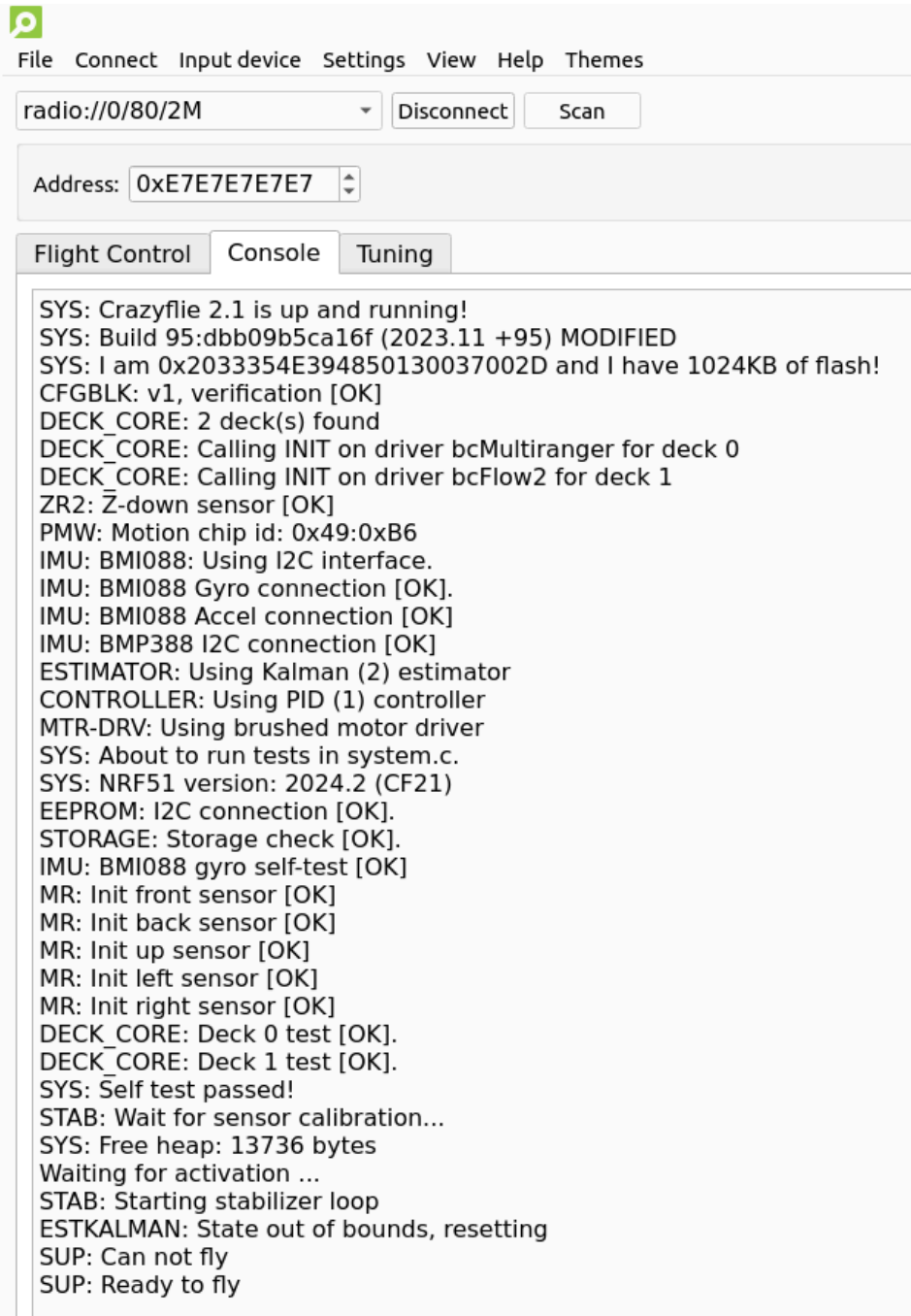
9. After all projects have been updated, it is important to note that the pre-installed Crazyflie client will not work initially. To fix this, open the Terminal and reinstall the Crazyflie client using the following commands:
pip uninstall cfclient
pip install cfclient

10. After successful reinstallation, open the crazyflie client and you should see the following



The console log tab is not automatically shown as a tab like in the image above. To view it, go to View, Tabs, then check the box labeled Console.

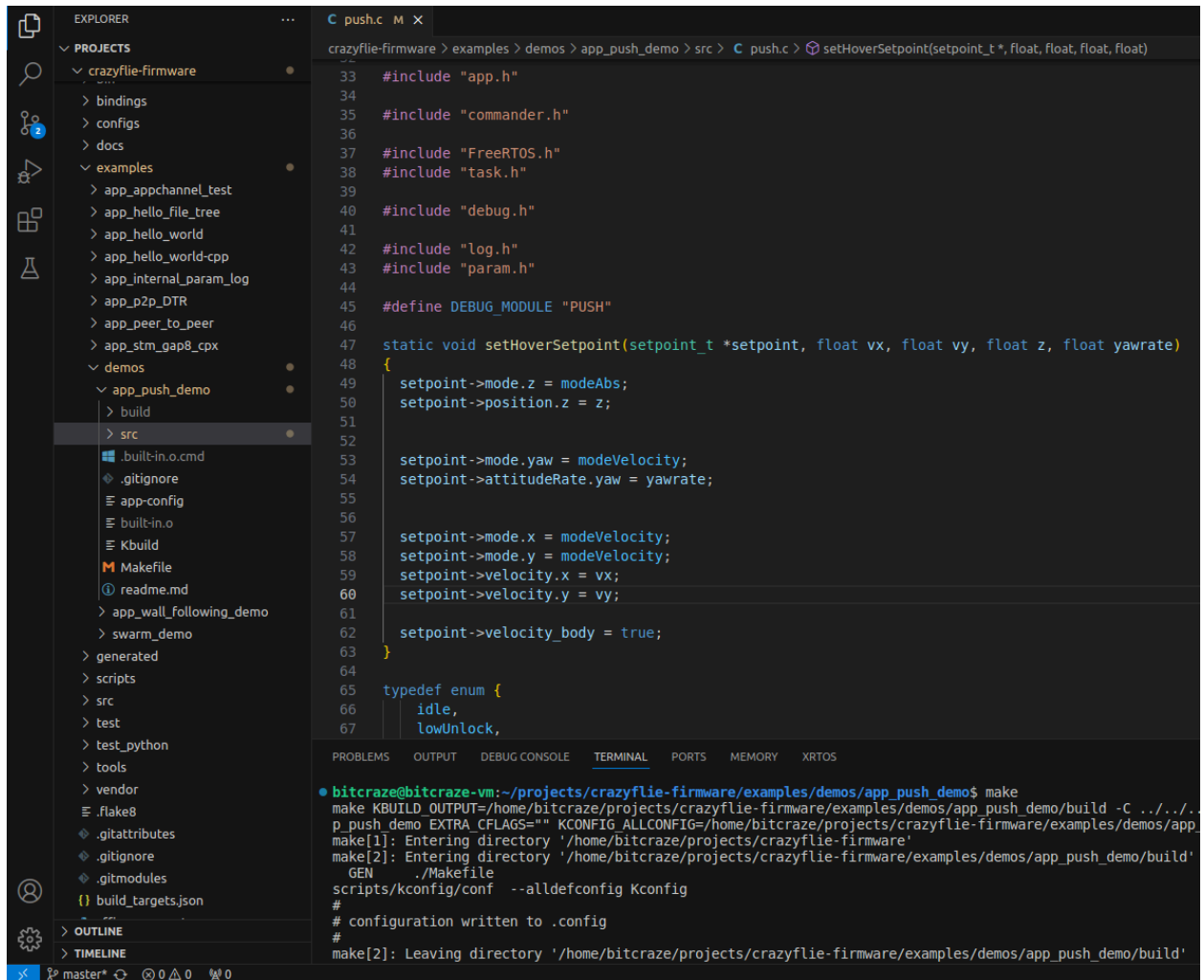
11. Now it's time to test out the crazyflie with the decks mounted. In the crazyflie client, power on the crazyflie and Scan for devices. Once the crazyflie is found, Connect. If scanning does not find any crazyflie, replug the crazyradio and completely restart the VM. After successfully connecting, open the Console tab. You are looking to find that the crazyflie boots properly, two decks are properly detected, and all self-tests pass. The image below is what your Console should look like after successfully powering on.



Should any self-test fail or any deck not be detected properly, attempt to remount the deck(s) and ensure they are mounted properly. Should this prove to not work, replace the deck with an issue with another if possible and repeat the steps above.

12. Upon verification of self-tests passing, close the crazyflie client and open Visual Studio Code which should be pre-installed on the VM. Open the projects file directory and navigate to `projects/crazyflie-firmware/examples/demos/app_push_demo`. Open the

integrated terminal in Visual Studio Code and type in the command make.



The screenshot shows the Visual Studio Code interface with the Explorer panel on the left displaying the project structure. The main editor shows the source code for `setHoverSetpoint` in `app_push_demo.c`. The integrated terminal at the bottom shows the output of the `make` command, which successfully builds the demo.

```
33 #include "app.h"
34
35 #include "commander.h"
36
37 #include "FreeRTOS.h"
38 #include "task.h"
39
40 #include "debug.h"
41
42 #include "log.h"
43 #include "param.h"
44
45 #define DEBUG_MODULE "PUSH"
46
47 static void setHoverSetpoint(setpoint_t *setpoint, float vx, float vy, float z, float yawrate)
48 {
49     setpoint->mode.z = modeAbs;
50     setpoint->position.z = z;
51
52
53     setpoint->mode.yaw = modeVelocity;
54     setpoint->attitudeRate.yaw = yawrate;
55
56
57     setpoint->mode.x = modeVelocity;
58     setpoint->mode.y = modeVelocity;
59     setpoint->velocity.x = vx;
60     setpoint->velocity.y = vy;
61
62     setpoint->velocity_body = true;
63 }
64
65 typedef enum {
66     idle,
67     lowUnlock,
```

```
bitcraze@bitcraze-vm:~/projects/crazyflie-firmware/examples/demos/app_push_demo$ make
make KBUILD_OUTPUT=/home/bitcraze/projects/crazyflie-firmware/examples/demos/app_push_demo/build -C ../../.
p_push_demo EXTRA_CFLAGS="" KCONFIG_ALLCONFIG=/home/bitcraze/projects/crazyflie-firmware/examples/demos/app
make[1]: Entering directory '/home/bitcraze/projects/crazyflie-firmware'
make[2]: Entering directory '/home/bitcraze/projects/crazyflie-firmware/examples/demos/app_push_demo/build'
GEN      ./Makefile
scripts/kconfig/conf --alldefconfig Kconfig
#
# configuration written to .config
#
make[2]: Leaving directory '/home/bitcraze/projects/crazyflie-firmware/examples/demos/app_push_demo/build'
```

13. Next, take the crazyflie and enter bootloader mode. To do that, power it off first. Then, hold the power button for three seconds where one blue LED lights up and as soon as it disappears, let go of the power button. Now you should be in bootloader mode where two blue LEDs are blinking rapidly. Go back to VS Code and in the integrated console, type `make load`
14. The binary should now be flashing to the crazyflie. Should an error occur, it is most likely with the crazyradio. To fix it, replug the crazyradio and restart the VM.
15. The demo should now be fully flashed to the crazyflie. To start the demo, place the crazyflie on the ground and power it on. Then, hold your hand really close above the crazyflie for a couple seconds and then move it away. The drone should start and hover in place. To operate the demo, attempt to slowly move your hand towards each side of the drone and observe it to fly away from your hand and other objects in its way. To end the demo, bring your hand to the top of the crazyflie and bring it closer to the top of the drone. The drone should respond by flying lower until it automatically stops hovering.