

## Set up MBot firmware

In this session, we are going to work on setup of the Control Board.

### 1. Compile the firmware files

- Now we'll clone our MBot repo. Open a new Terminal in the VSCode remote session:

```
cd ~
git clone https://github.com/SuryaPratapSingh37/Robot-Navigation-SLAM.git
```

- Compile the firmware code to get `.uf2` binary files

- Install lcm related stuff

```
cd ~/Robot-Navigation-SLAM/mbot_lcm_base
./scripts/install.sh
```

- Run the firmware setup script

```
cd ~/Robot-Navigation-SLAM/mbot_firmware
./setup.sh
```

- Build firmware

```
cd ~/Robot-Navigation-SLAM/mbot_firmware
mkdir build
cd build
cmake ..
make
```

- Now you will have 2 relevant `.uf2` files under `/build`

- The calibration script, `mbot_firmware/build/tests/mbot_calibrate_classic.uf2`
- The MBot firmware, `mbot_firmware/build/src/mbot.uf2`

### 2. Calibrate the MBot and flash the firmware

In this step, we are going to flash the calibration script onto the Pico to calibrate it before we flash the firmware.

- Place the MBot on the floor in a spot with at least 2 feet of clear space all around the robot, preferably on the same type of surface that you plan to use the robots on.
- Run the following command, the Pico will reboot automatically, and will then run its calibration routine right away. Allow the Pico to finish its calibration routine without interference.

#### WARNING

Hold off on running this command until the robot is placed on the floor.

```
cd ~/Robot-Navigation-SLAM/mbot_firmware
# upload the calibration scripts
sudo ./upload.sh flash build/tests/mbot_calibrate_classic.uf2
```

Note that during the calibration routine, the robot should turn in a counterclockwise circle first, then turn clockwise. If it is not executing in this order, you might have wrong motor polarity. Modify it in the `mbot_firmware/tests/mbot_calibrate_classic.c` to be either 1 or -1.

```
#define MOT_LEFT_POL 1
#define MOT_RIGHT_POL 1
```

- The calibration script will have saved parameters onto the Pico's memory. We can now flash the firmware that will run on the Pico during operation.

```
cd ~/Robot-Navigation-SLAM/mbot_firmware
sudo ./upload.sh flash build/src/mbot.uf2
```

### 3. Using Minicom to verify

Here we introduce you to the tool Minicom. It is a program designed for serial communication that connects devices to a Linux PC via serial ports. We will use Minicom to read the Pico printouts from the Jetson module.

- After flashing the firmware to the Pico, run the following command to start Minicom:

```
minicom -D /dev/mbot_tty -b 115200
```

`-D` indicates the serial port device, and `-b` sets the communication speed or baud rate.

If the Minicom command doesn't work, run this:

```
ls /dev | grep mbot
```

```
mbot@mbot-0018-shaw:/dev$ ls /dev | grep mbot
mbot_lcm
mbot_tty
```

If you do not see the 2 outputs above, unplug the USB which connects Jetson and Pico, then plug it back in.

To exit Minicom, press `CTRL-A` to get to command mode, then press `x` to quit.

**Successful Firmware Flashing:** After flashing the firmware successfully, Minicom will display your encoder counts, IMU values, and more. Manually turning the wheel will update the encoder counts in the Minicom terminal.

**Unsuccessful Firmware Flashing:** If the firmware doesn't flash correctly, repeat the calibration and firmware flashing steps. Open a second terminal window with Minicom to monitor its outputs for troubleshooting.

To make running Minicom easier, consider creating a permanent alias by editing your `.bashrc` file. Add the line `alias start-minicom='minicom -D /dev/mbot_tty -b 115200'` to the end of your `.bashrc` file, then run the command `source ~/.bashrc` in a terminal. This will let you run Minicom using the command `start-minicom`.

### 4. Manually enter bootloader mode

If the firmware was successfully flashed, skip this step and proceed to the next.

If your firmware flashing was not successful because the `./upload.sh` script from step two doesn't work for your MBot, you can manually enter the bootloader mode (namely let Pico ready to get the firmware) by following instructions here: