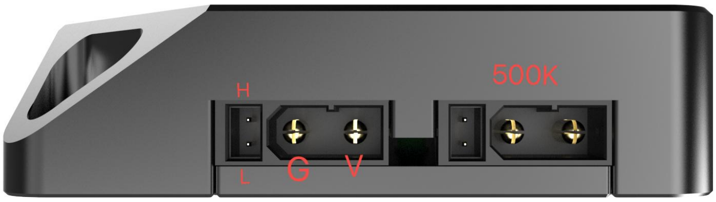
IMU User Guide

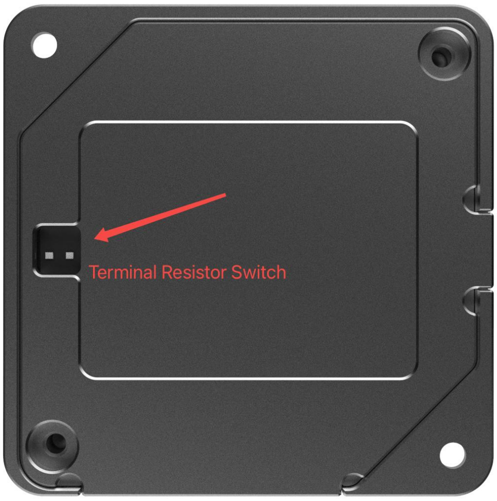
Get Started

We provide linux driver using SocketCAN. It is the recommended way to work with IMU.

Before you start, you need to connect the device to your PC! Here's the can bus wire and baudrate definition of the IMU HEX CAN



The IMU have a switch to control terminal resistors inside them. We use split resistor so there will be two switches. Either turn them all on or all off.



Use our driver

> Only linux is supported

Connect the can to your computer, via USB-CAN convertor or ETH-CAN convertor. Like CANable or something.

No matter what you use, make sure that you can read the can bus using SocketCAN.

Get the ROS Driver

1. Clone the repo from https://github.com/hexfellow/hex_imu into a workspace
2. Run `catkin_make` or `colcon build` according to your ROS version
3. Make sure IMU is connected, CAN is set to correct baudrate as HEX CAN and the CAN bus is up.
4. Run the roslaunch file `launch/ros1/canopen_imu.launch` or `launch/ros2/canopen_imu.launch.py` if you are on ROS2
5. Read from topics specified in `README.md`

Write your own driver to directly communicate using CAN bus

> Besides the CANOpenID used, our device also reserves EXT-ID 0x0000000A for future use.

Our device implements the CANOpen protocol. By default, the NodeID is 0x10.

If you don't know CANOpen, here's a nice intro https://www.csselectronics.com/pages/canopen-tutorial-simple-intro

CANOpen EDS file

See attached link