Not a Demo Replicating:

"Realtime Edge Based Visual Inertial Odometry (REBVO) for MAV Teleoperation in Indoor Environments
Juan Jose Tarrio · Sol Pedre"

Joshua McCready

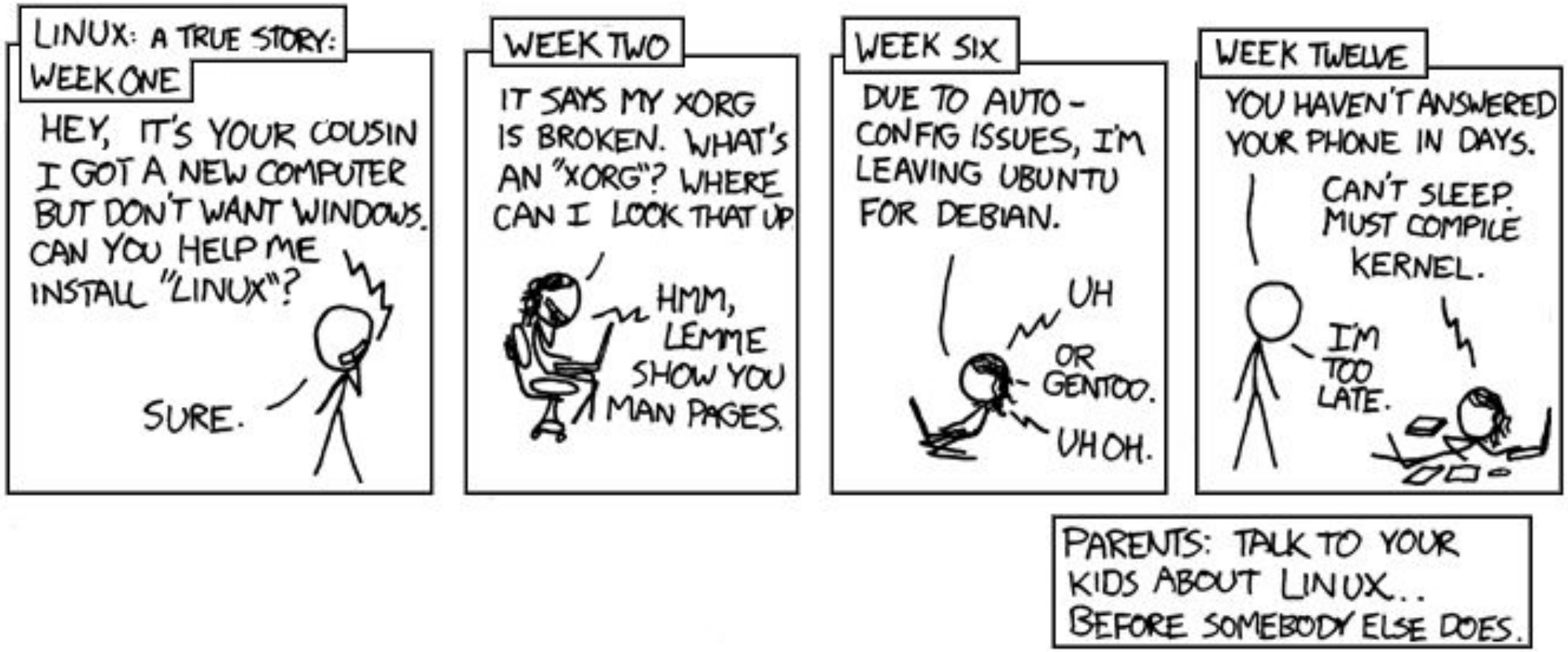
December 4th, 2019

Stated goals in project proposal:

Use monocular visual odometry that focused on addressing the constraints of real time operation on mobile robotics with limited computational resources:

1. Replicate runtime environment for REBVO on the Picar and required dependencies so the source code can compile on the Picar and run the benchmark EuRoC dataset. **– Skid right to a halt here**
2. Use the "Custom Camera" REBVO class to interface with images gathered by the usbcam ros package.
3. Configure the system with gathered camera parameters from previous Camera Calibration and other required parameters.
4. Try to use the built in ROS support for the application to transmit depth maps that could be used by a PID controller to show proof of concept.

XKCD: Cautionary



https://xkcd.com/456/

Into the rabbit hole:

Realtime Edge Based Visual Inertial Odometry for MAV Teleoperation in Indoor Environments code base can be found here:

https://github.com/JuanTarrio/rebvo

Includes an application to simulate a dataset with no hardware or a video and imu data feed. Importantly it also has a ros implementation that I hoped to use.

Top Left: Original image + Edgemap compression data rate
Top Right: Surface fitting Bottom Left: Full Edgemap Bottom Right: Compressed edgemap @ 4FPS

Too far into the rabbit hole

After a long hunt to install all package dependencies, the project built but…

I screwed something up quite badly and broke the rqt_xxx commands necessary for ros debugging. I think it was reinstalled Qt (?) or something else I thought I needed.

After many attempts to remedy the issues, I am re-imaging on a new SD card and starting over…

- **System requirements**

In ubuntu and most linux dist this libraries can be downloaded directly from the repos, except for TooN.

-- C++11

-- Linux, X11, v4l2

-- OpenGL development libraries (GL,GLU,glut)

-- TooN 2.2 mathematical library (http://www.edwardrosten.com/cvd/toon.html - ZIP provided in the repo)

-- Lapack (for advanced TooN functions)

-- LibAV (Video Codecs)

-- LibGD (Image managment)

-- Optionally NE10 for ARM Neon optimizations

Also need an IMU

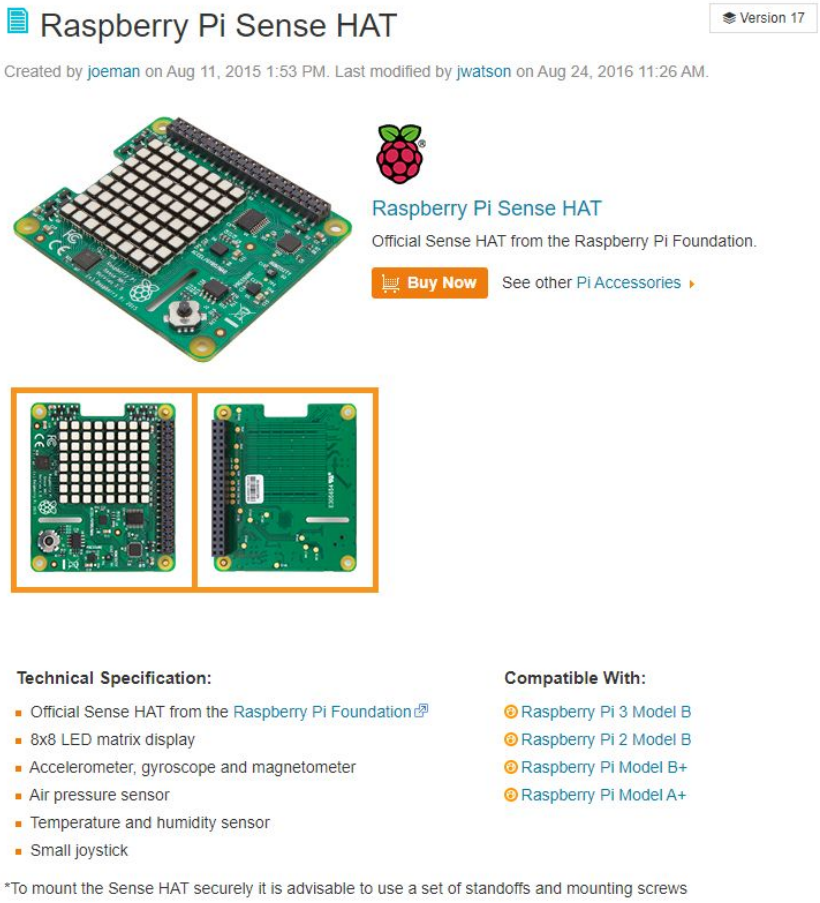
It was easy to find a IMU that will work easily with raspberry pi that doesn't interfere with Sunfounder Hat.

Dedicated python library:

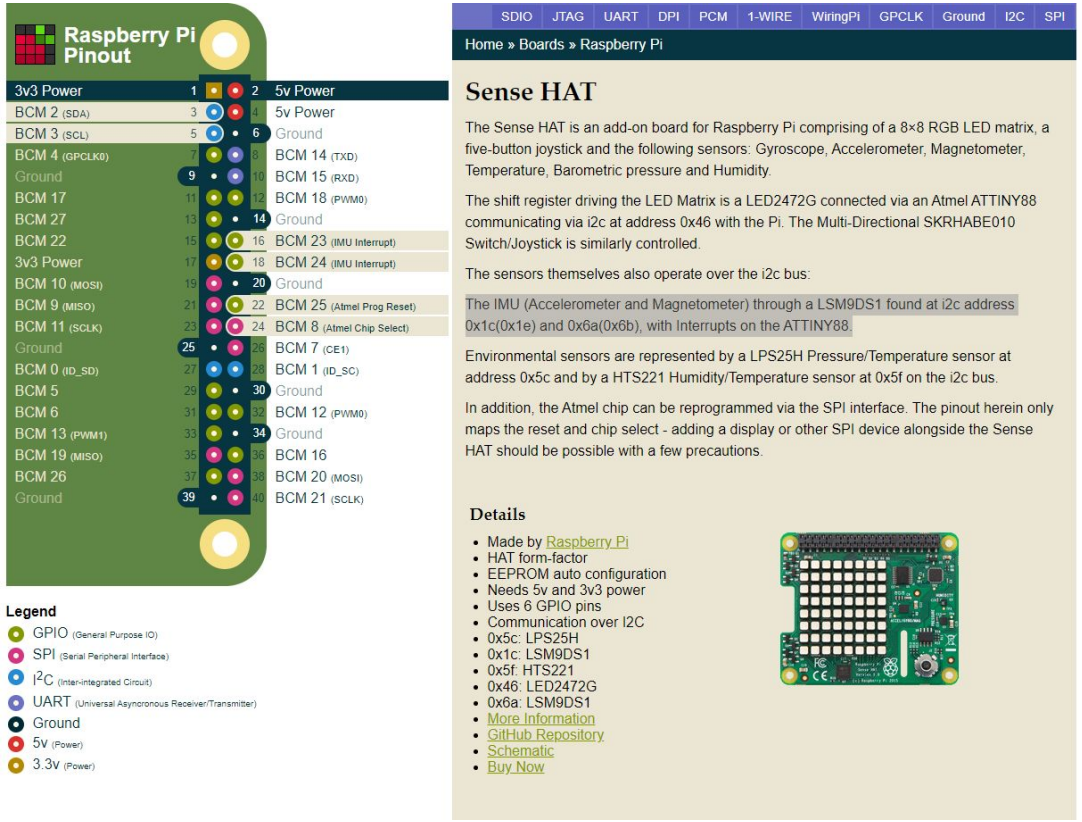
https://pypi.org/project/sense-hat/

Starting point ROS Node:

https://github.com/mirkodcomparetti/ros_sensehat



Here’s an IMU – will it work…? Probably



Picar hat: Robot HATS uses:

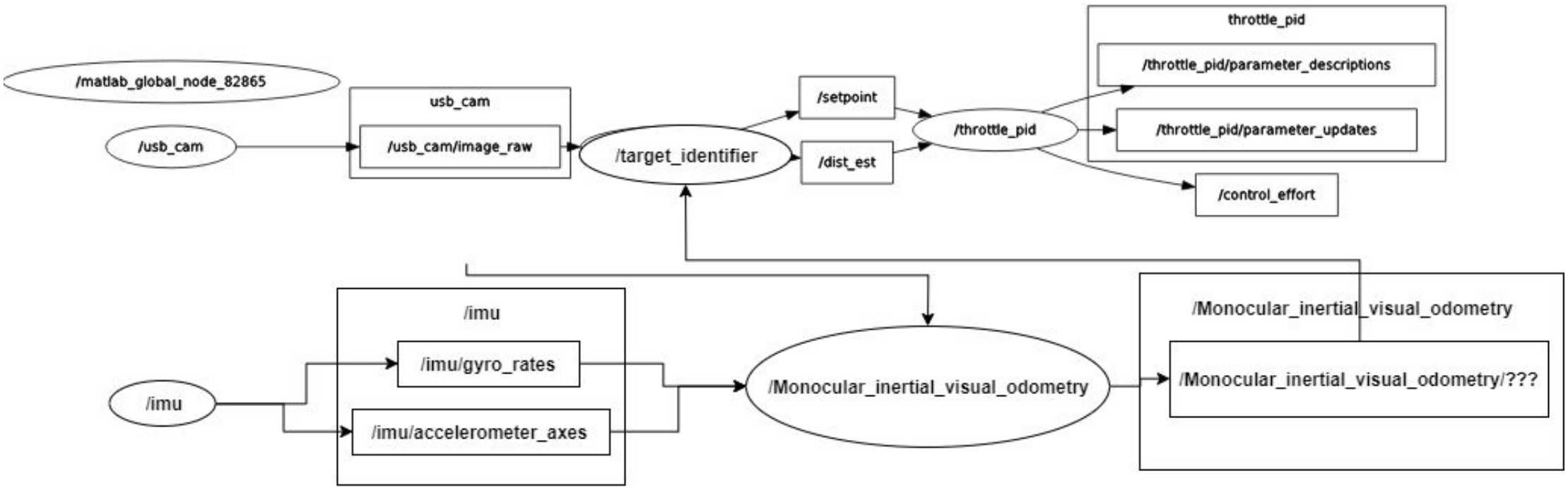
I2C: PCF8591 is used as the ADC chip, with I2C communication, and the address 0x48

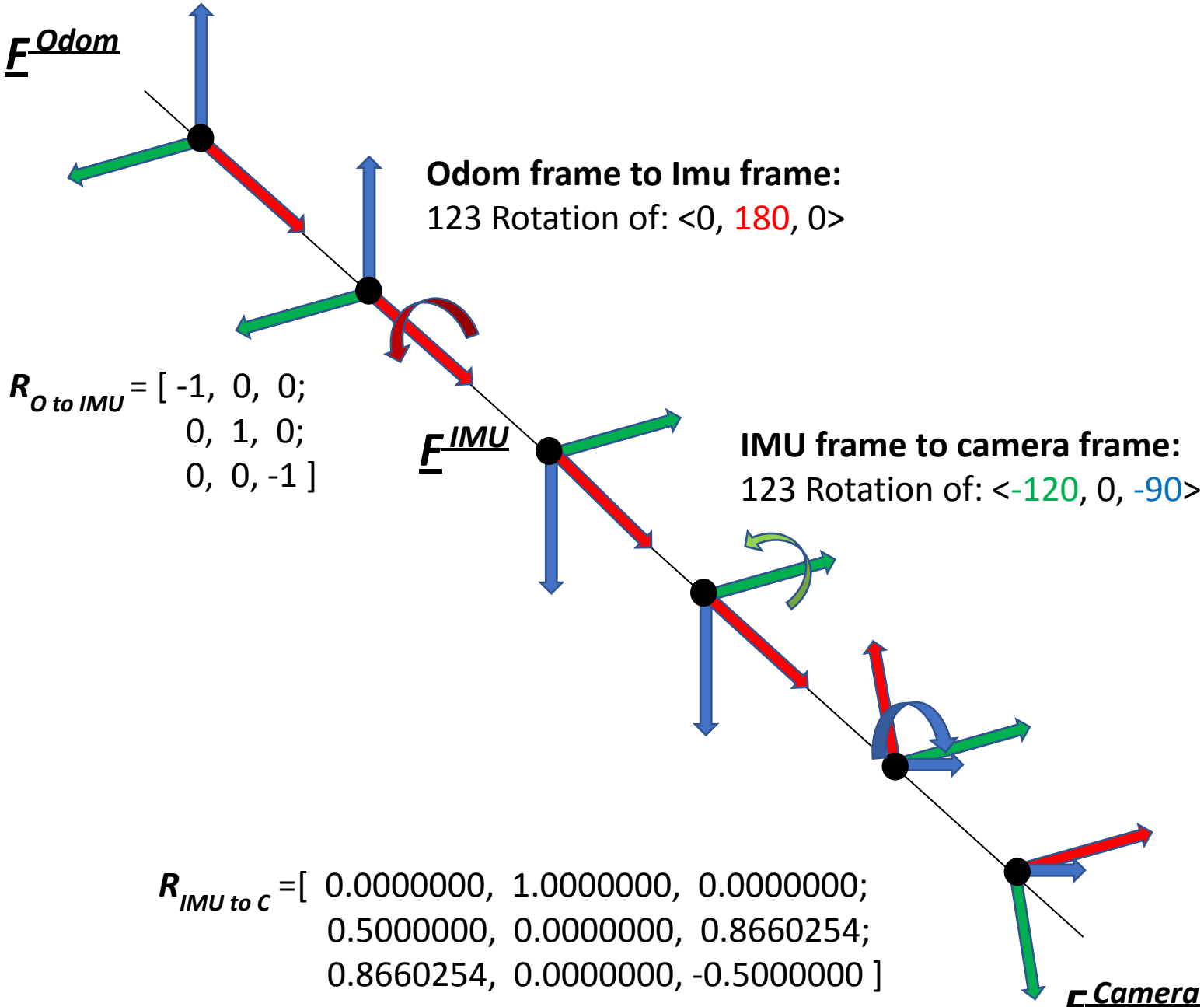
TB6612 motor control ports: includes 3.3V for the TB6612 chip, 5V for motors, and direction control of motors MA and MB; working with SunFounder TB6612 Motor driver

If I interpret the silkscreening right then the only other concern is B27 and B17 which I believe are not used by the Pi Sense Hat.

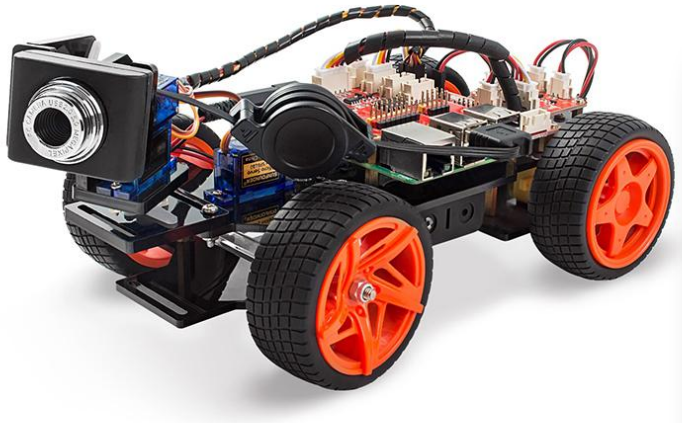
Unable to find a pinout so I will “give it a try…”

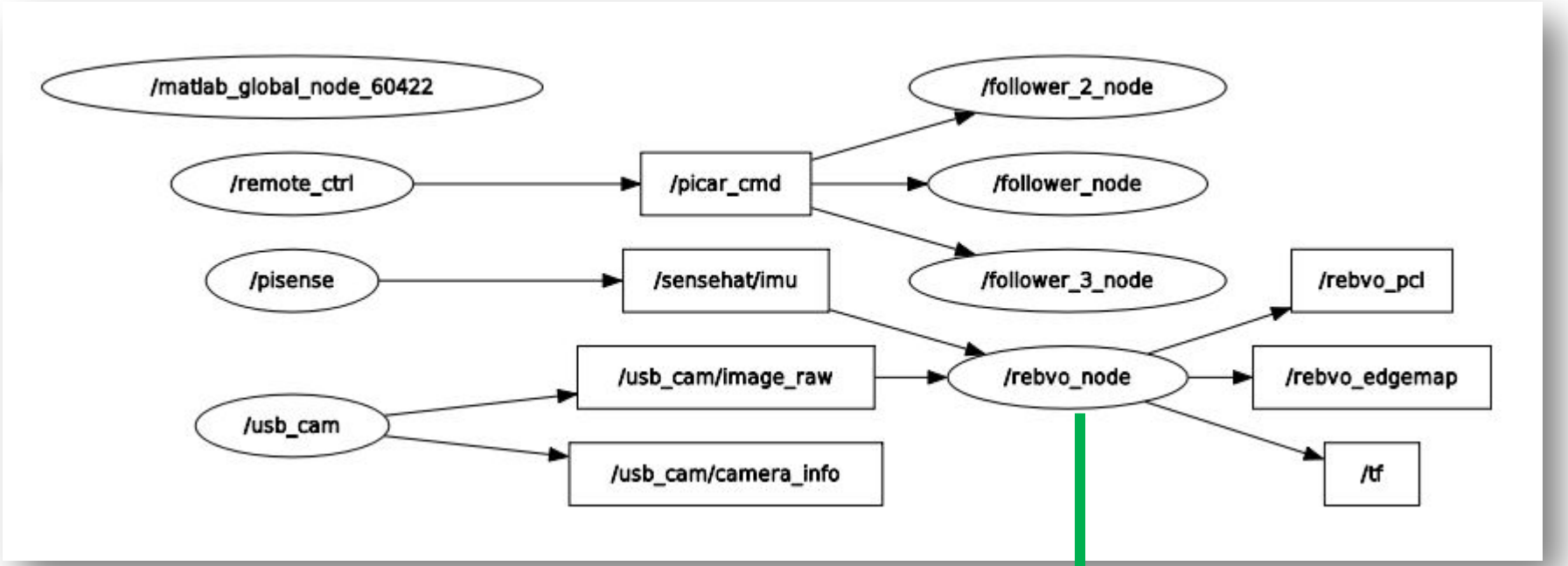
Promised Land: Rqt_graph mock up




F Odom
Odom frame to Imu frame:
123 Rotation of: <0, 180, 0>
R O to IMU = [-1, 0, 0;
0, 1, 0;
0, 0, -1]
F IMU
IMU frame to camera frame:
123 Rotation of: <-120, 0, -90>
R IMU to C =[0.0000000, 1.0000000, 0.0000000;
0.5000000, 0.0000000, 0.8660254;
0.8660254, 0.0000000, -0.5000000]
F Camera

Raspberrypi: 192.168.1.84:2708
Picar robot and REBVO





UDP

Visualizer

raspberrypi_viewer: 192.168.1.71:2708
Remote visualization of REBVO output