

Hamster Technical Specification

LLC Rev 4 / Ubuntu 16.04

Raspberry Pi 3 Model B / ROS Kinetic



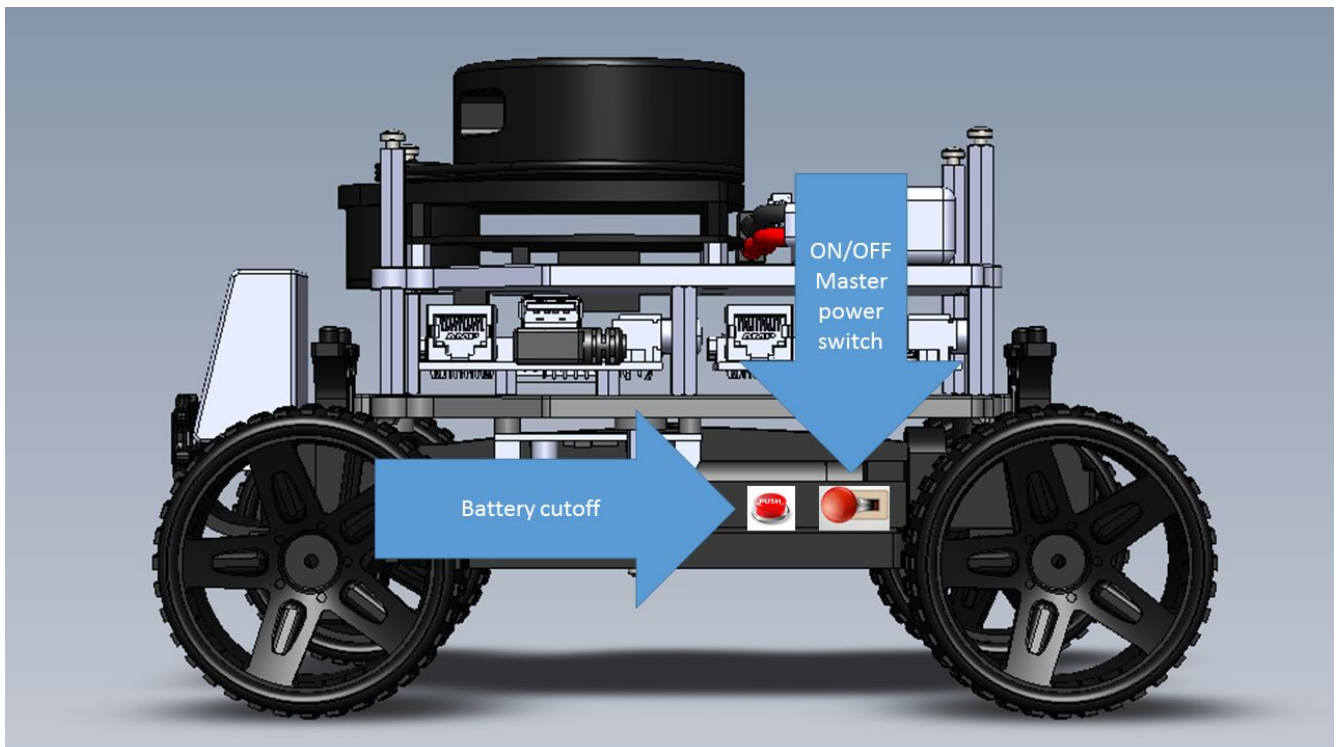
Introduction

Hamster robot is a robotic research platform focused on autonomy.

Hamster Overview

The hamster platform arrives with various components pre-installed:

- A 8m range 360 2D Lidar supported by the ROS community here <http://wiki.ros.org/rplidar>.
- An HD capable camera supported by ROS https://github.com/fpasteau/raspicam_node
- 1 Raspberry PI board with Linux (Debian/Raspbian) and ROS Kinetic
- A Low Level controller (LLC) with dedicated hardware for power distribution and monitoring



Hamster main components

Lidar

- Range 8m 360deg
- Angular resolution 1°
- Scan Frequency 5Hz
- Power 5VDC
- USB connection to computing module



Figure 1 Lidar

Camera (raspicam)

- five megapixel fixed-focus camera
- Supports 1080p30, 720p60 and VGA90 video modes
- Supports stills capture.
- It attaches via a 15cm ribbon cable to CSI port on
- Can be accessed through the MMAL and V4L APIs, and there are numerous third-party libraries built for it
 - Including a Python library

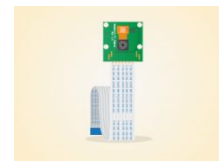


Figure 2 Camera module

CPU: RaspberryPi 3 Model B (ARM Linux)

- 1 x ARM CPU 1.2 Ghz quad core
- 1 x 1G RAM
- 1 x Videocore 4 GPU
 - .1. The GPU is capable of BluRay quality playback
 - .2. Using H.264 at 40Mbits/s.
 - .3. Fast 3D core
 - .4. Supplied OpenGL ES2.0 and OpenVG libraries
- 16Gb SD
- 40 dedicated GPIO pins
 - .1. UART
 - .2. i2c bus
 - .3. SPI bus with two chip selects
 - .4. i2s audio,
 - .5. 3v3, 5v, and ground.
 - .6. DIO
 - .7. PWM
 - .8. ADC
- 4 USB
- Current stabilization to 5v for up to 7A
- Wifi



CPU (Arduino)

- Arduino pro mini 5v
- Low Level control
- Battery state monitor



Motors

- 9.7:1 Metal Gearmotor 25Dx48L mm HP
- EMAX Servo motor 2kg Turque ES08MA II With Metal Gear



Pololu



Battery

- Lipo
- 2200MAh
- 1H continuous operation
- 3H at standby
- Extendable



Charger

- Provided with the System

ROS Architecture Overview

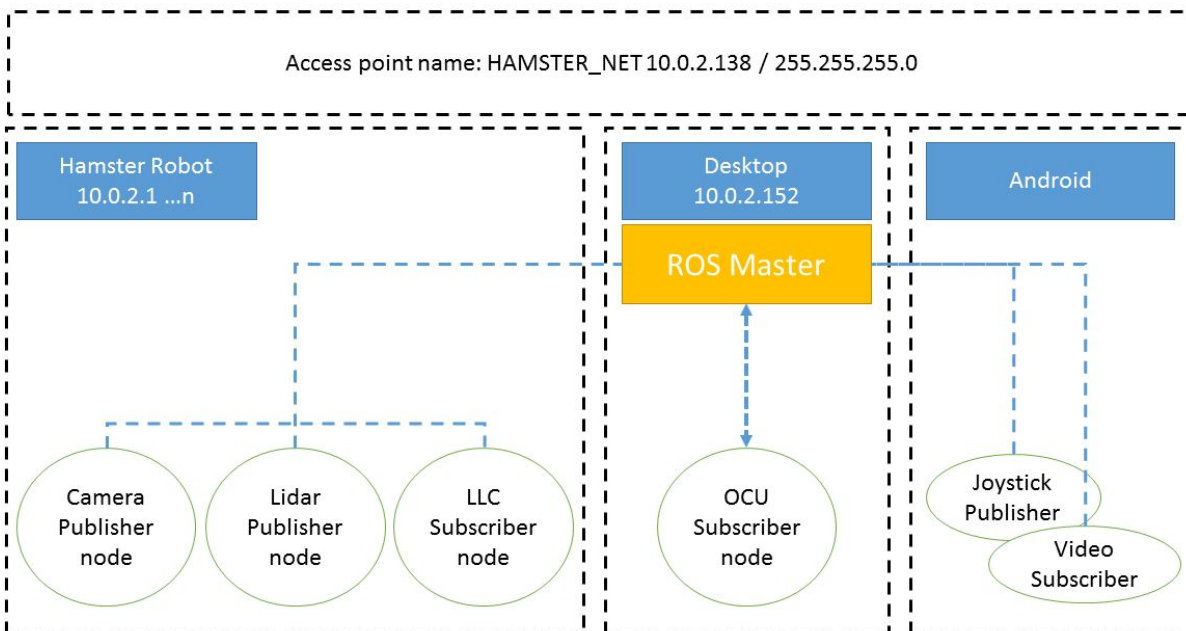
Hamster robot arrives with Linux (ARM) and ROS pre-installed.
Thus, the best method of communication with the robot is through ROS.

What is ROS?

ROS (Robot Operating System) is a BSD-licensed system for robotic components development. A ROS system is comprised of a number of independent nodes, each of which communicates with the other nodes using a publish/subscribe messaging model.

Initial setup (accessing ROS Master)

The ROS master is very similar to a DNS server in ROS. It keeps the IP/Port of the various nodes (components) in the system and enables the initial connection setup between them.



Once the connection is made between the publisher and subscriber, the communication turns peer to peer and is direct over TCP (sensor data does not passed through the ROS master).
