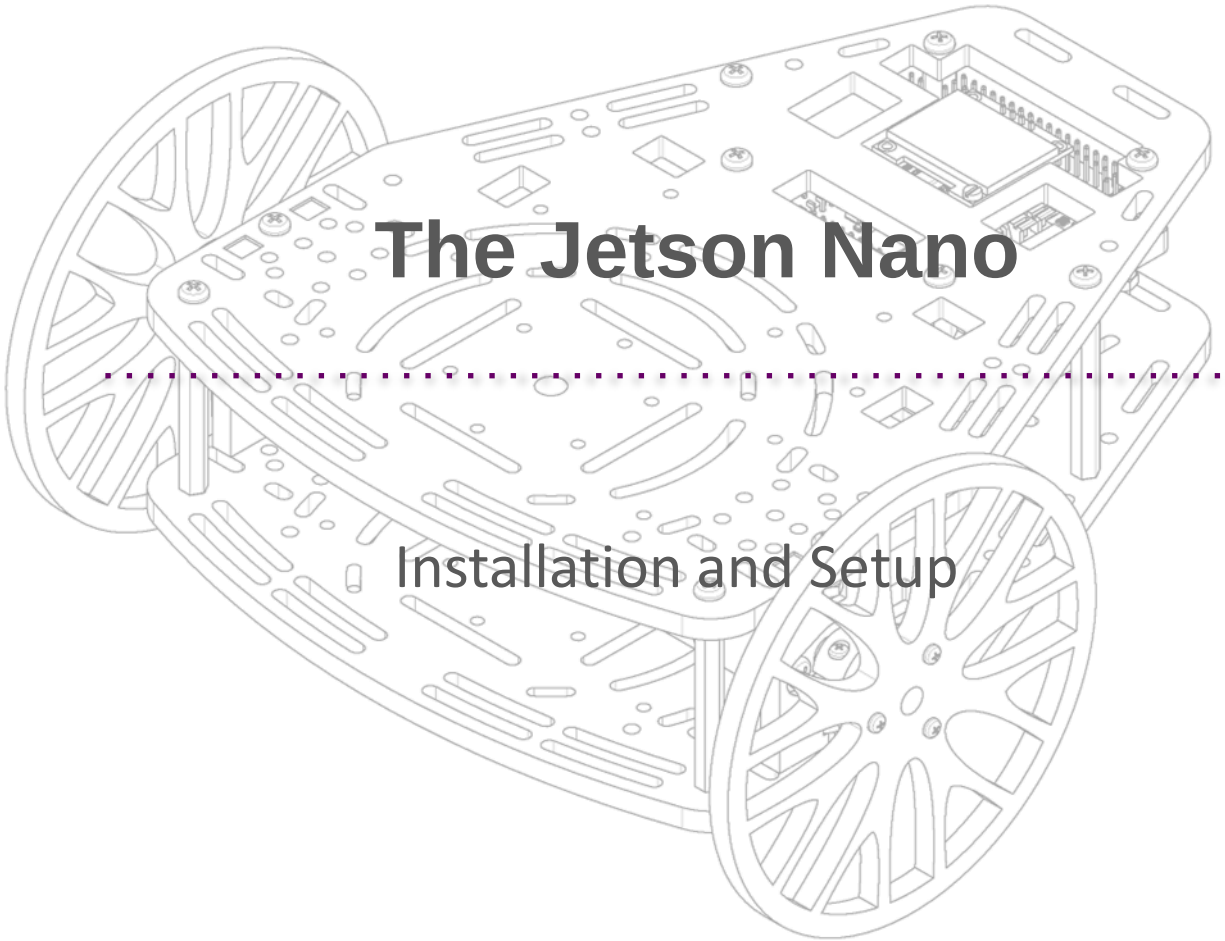
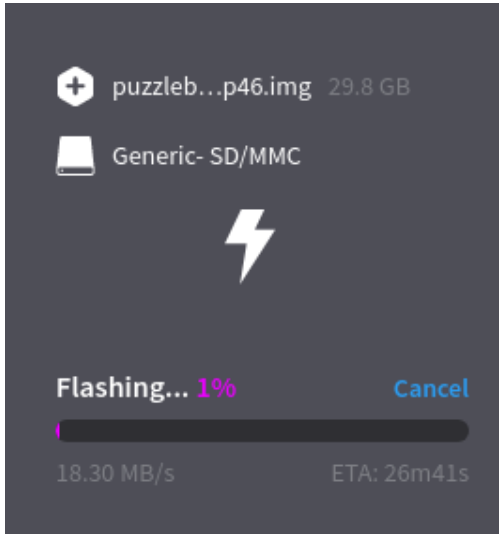


# The Jetson Nano

## Installation and Setup

# Installation

- The OS for the Jetson is stored on an SD card
- An image must be flashed to the SD card, download it from here:
  - https://www.dropbox.com/s/xafcp9n64sq0raz/puzzlebot-003-nano-2gb-jp46.zip?dl=1
- This is flashed from an image using this software:
  - https://www.balena.io/etcher/
- To flash the SD card:
  1. Insert the SD into your PC
  2. Launch the etcher software
  3. Select the image downloaded from the link above in the “flash from file” section
  4. Select the SD card in the “select target” section
  5. Click “flash”, and wait.



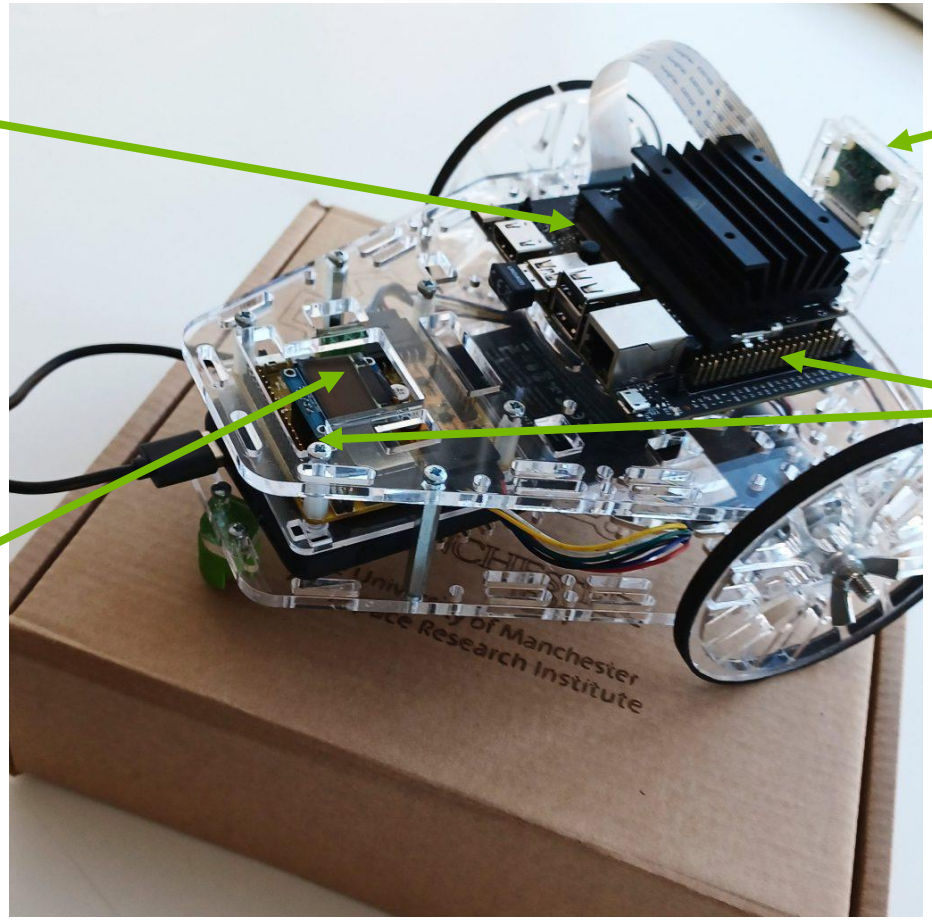
# Puzzlebot: Jetson Edition

NVIDIA Jetson Nano

For AI and computer vision

Hacker Board

For low-level control algorithms




Raspberry Pi Camera

GPIO Arrays

Expansion possible via the Jetson or the Hacker Board

# The Jetson Nano 2GB

- 128-core NVIDA Maxwell GPU
- 1.43 GHz Quad-core ARM A57 CPU
- 2 GB of 64-bit LPDDR4 Memory
- SD card for storage
- Ethernet & WiFi
- CSI-2 Connector for Camera
- Runs a modified version of Ubuntu 18.04

# The Jetson Nano 2GB

- Communicates with the Hacker Board serially via ROS
- Runs NVIDIA's own version of Linux, similar to Ubuntu
- The OS is flashed onto the SD card by a PC
- Three options for setup
  - Use the provided image in place of the NVIDIA image (recommended)
  - Run a setup bash file
  - Manual installation

# Booting the Jetson

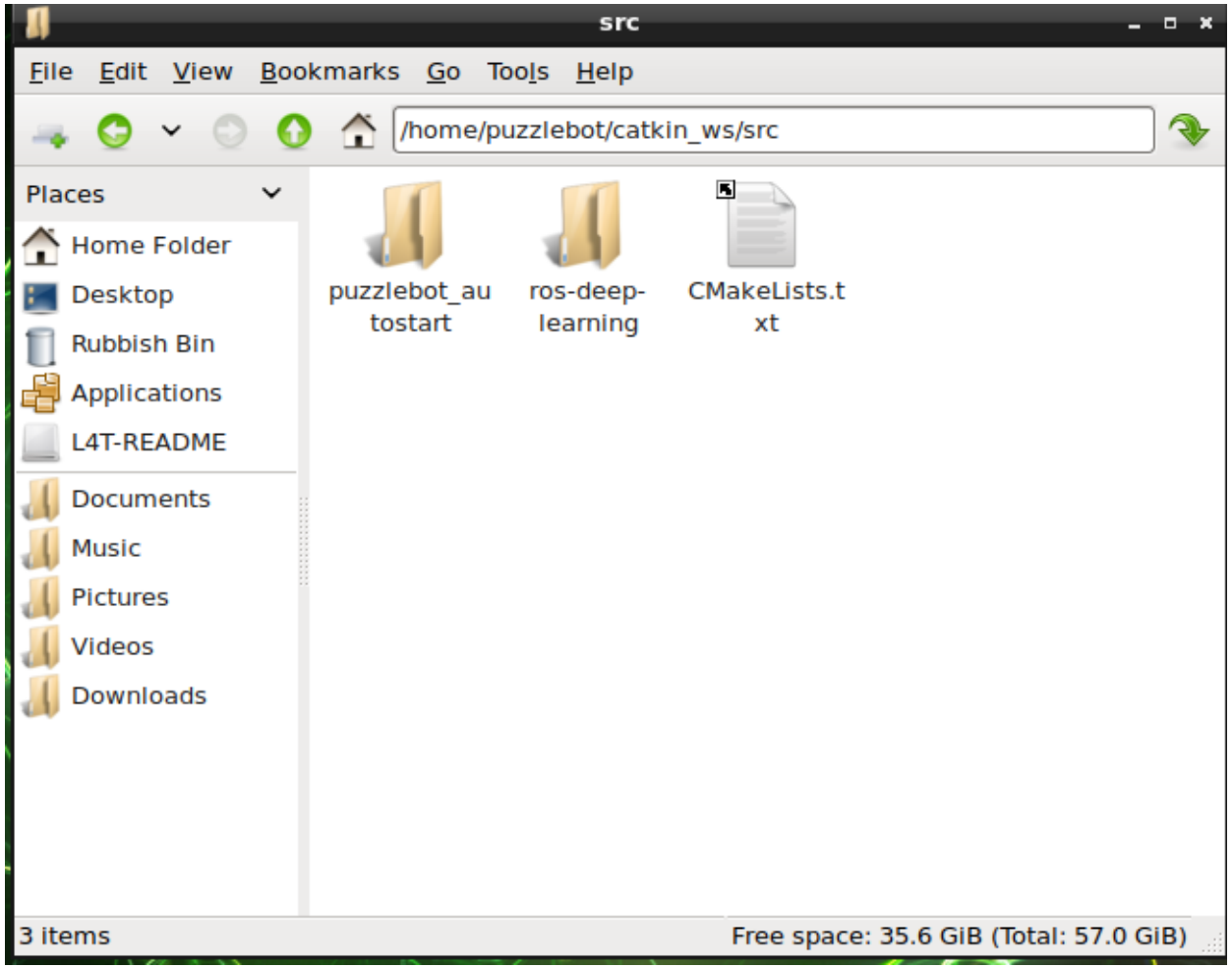
- Connect the Jetson to a screen, mouse, and keyboard.
- Ensure the Hacker Board is connected and booted
- Connect the USB-C port of the Jetson to a battery pack
  - Other USB power sources work but the Jetson draws up to 3A of current and may crash if the power supply cannot provide this
- Once the Jetson has booted, you should see the Jetson Desktop


Trash
Jetson Developer...
Jetson Support F...
Jetson Zoo
LXTerminal
NVIDIA Jetson Co...
Chromium Web Brow...
VPI Demos v1.1
17:14

# Preinstalled software

- ROS
- TensorFlow
- OpenCV
- nvidia Camera nodes
- Hackerboard Communication Routines
  - The Hacker Board communication starts each time the Jetson is booted up
  - To test the communication, use `rostopic list`. You should see list of topics as shown, although this will depend which control mode the Hacker Board is using.
  - If the communication fails, the protocol can be restarted with the command: `sudo systemctl restart puzzlebot.service`

# Preinstalled software




- There is a pre-setup catkin workspace on the Jetson
- These two packages are necessary for the PuzzleBot and camera communication, and should not be changed in any way

# Testing

- Test the ROS communication with `rostopic echo`
  - Echo the topics /wr and /wl, and rotate the wheels
  - The speed of the wheels should be displayed
- Publish to the command topics, the wheels should turn
  - If control mode 1 is used, publish to `/cmd_vel`
  - If control mode 2 is used, publish to `/cmd_wR` and `/cmd_wL`
  - If control mode 3 is used, publish to `/cmd_pwmR` and `/cmd_pwmL`
  - The control mode is changed on the Hacker Board webpage

# Rasperry Pi Camera

- NVIDIA provides a package for interfacing with a CSI camera
  - This Package is pre-installed on the PuzzleBot image
- Several launch files are available. Only 2 are of interest to us:
  - ros_deep_learning video_viewer.ros1.launch
  - ros_deep_learning video_source.ros1.launch
- On your Jetson, run the command:
  - `roslaunch ros_deep_learning video_viewer.ros1.launch`
- The camera view should be displayed on the screen.

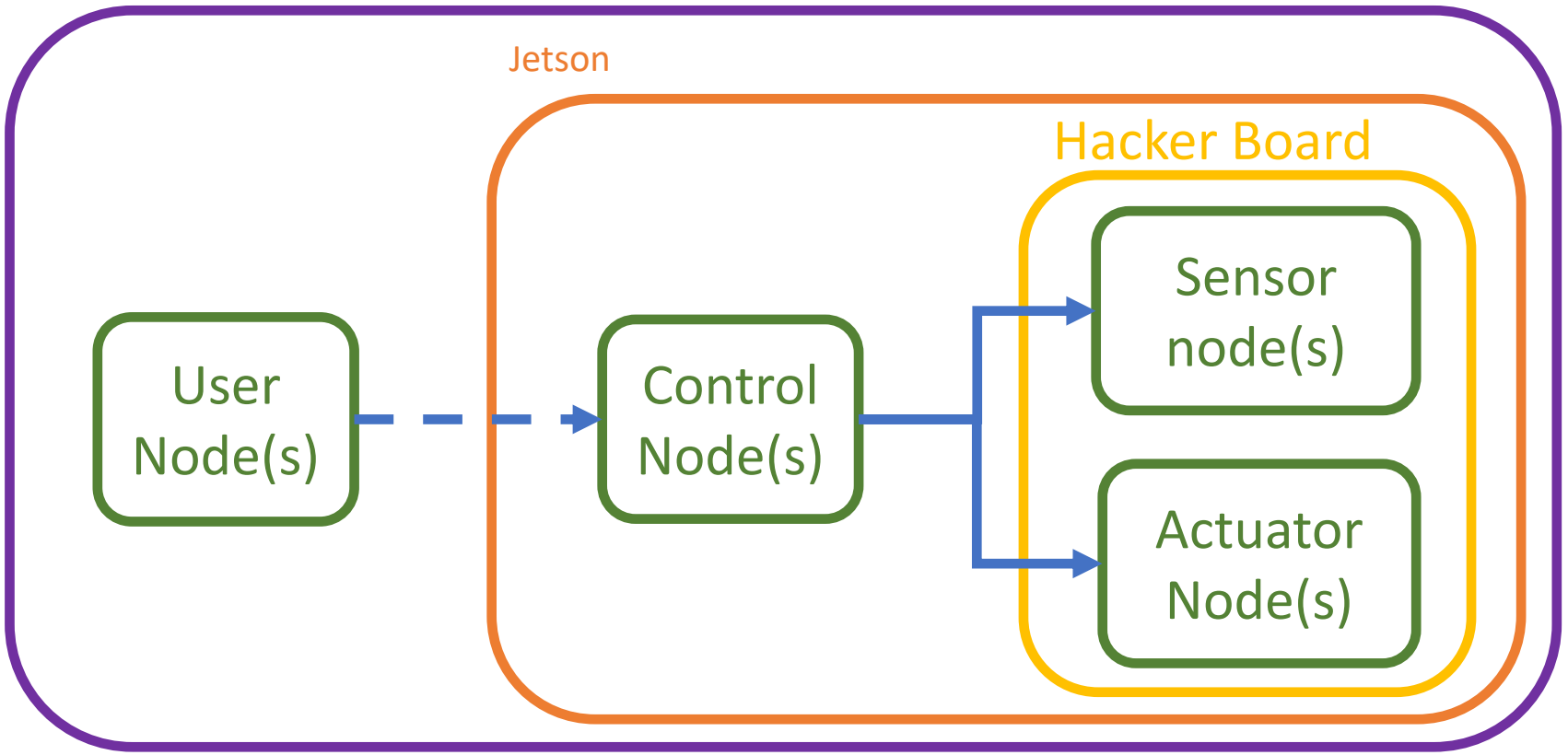
# Remote access

- It can be useful to control the Jetson from a remote PC, as the robot cannot be in motion and also hooked up to a monitor
- To do this, SSH is used. It uses WiFi to give your computer access to the Jetson.
- The Jetson generates its own WiFi network for communication with an external device:
  - Default Network Name: PuzzlebotJetson
  - Default Network Password: Puzzlebot72
- **Before** attempting remote access, **change the Network Name** to something unique
  - The WiFi details can be changed on the Jetson by selecting:
    Networks->Edit Connections->Hotspot
- As with the Hacker Board, there will be conflict issues if many Jetsons are in the same room and share the same network name

# Activity: Remote access

- On the external PC, access the Jetson with ssh:
  - `ssh puzzlebot@10.42.0.1`
  - Password: `Puzzlebot72`
- If prompted, type `yes` and then press enter.
- This command window is now equivalent to one running on the Jetson.
- Any command can be run from this window, and it is equivalent to running one on the Jetson.
  - Test this by echoing the wheel velocities again.
- Once any control code is written on the Jetson, it can be tested and debugged remotely via SSH, enabling the PuzzleBot to move around

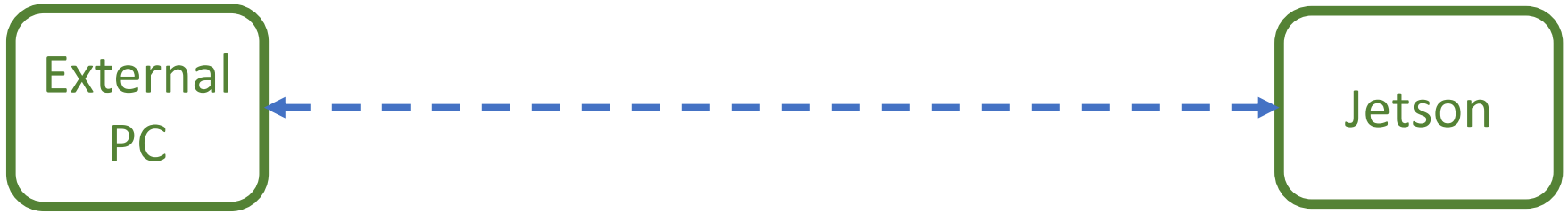
# The Jetson Nano with ROS




ROS Implementation

# Multi-device Communication

- Each device has its own ROS_IP and ROS_ MASTER_URI variables
- The ROS_IP is always the local IP of the device
- The ROS_MASTER_URI informs the devices where in the network the ROS master is
- By default, both IP and URI are local to each device




- ROS IP – local IP with reference to Jetson Network
- ROS Master URI – Points to master on the Jetson

- ROS IP – local IP with reference to Jetson Network
- ROS Master URI – Points to local master

# Activity Teleoperation

- Setup a PuzzleBot Jetson
- Install the ROS teleop twist keyboard package on an external PC
  - sudo apt install ros-melodic-teleop-twist-keyboard
- Connect the external laptop/PC
- Remotely connect to the PuzzleBot from the external device
  - Use `ifconfig` to get local IP. It will be of the form 10.42.0.XXX
  - `export ROS_MASTER_URI=http://10.42.0.1:11311`
  - `export ROS_IP=<your_local_ip>`
- Once connected, use `rostopic list` to check if the connection has been successful
  - The topics `/cmd_vel`, `/wr`, and `/wl` should be displayed, along with a few others

# Activity: Teleoperation

- Once your ROS_IP and ROS_MASTER_URI are set, your remote device has access to all the ROS topics, services, etc that the Jetson has.
- It is **not** a remote into the Jetson like SSH, we cannot start nodes or run other commands
- However, it can be more useful, as we cannot easily display visualisations via SSH.
- Use the external device to remotely operate the puzzlebot
  - `rosrun teleop_twist_keyboard teleop_twist_keyboard.py`
  - Follow the instructions displayed in the command window
- Use SSH to start a source camera node
  - `roslaunch ros_deep_learning video_source.ros1.launch`
- Use rqt to view the image from the camera on your machine
  - `rosrun rqt_image_view rqt_image_view`, and select the image_raw topic