Puzzlebot NVIDIA Jetson® Edition.

Introduction

{Learn, Create, Innovate};

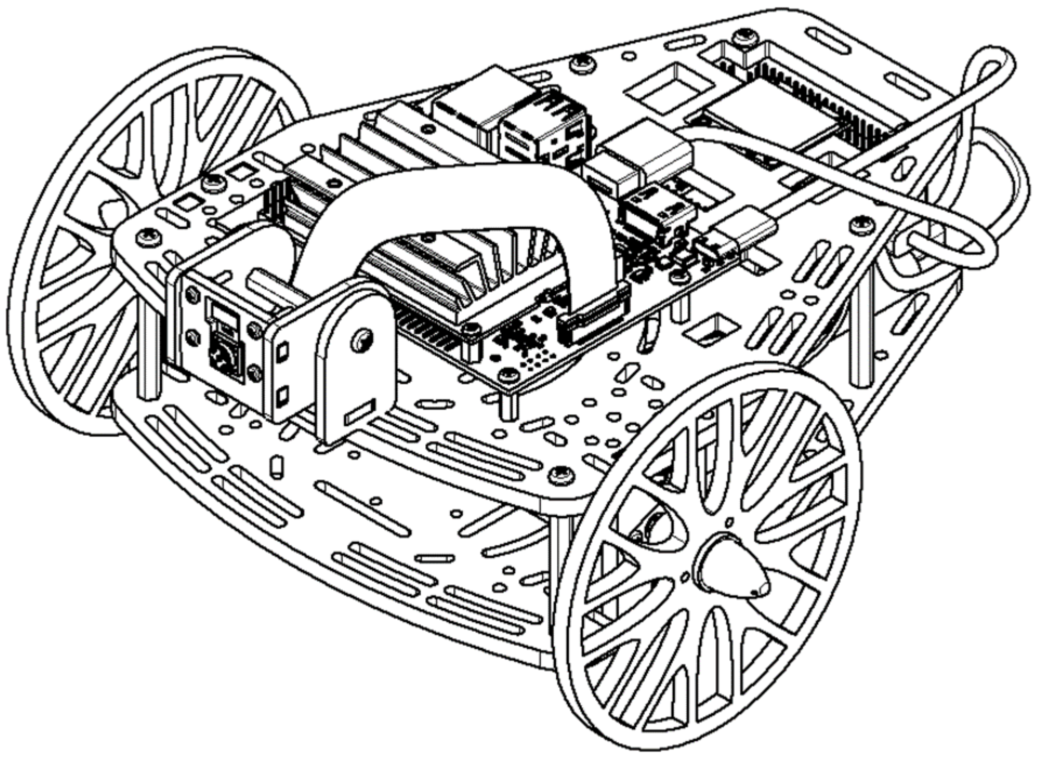
Puzzlebot: NVIDIA Jetson Edition

Introduction

- The Puzzlebot NVIDIA JETSON® Edition is an extension of the Puzzlebot Hacker Edition encompassing an NVIDIA Jetson® CPU and a Raspberry Pi® Camera.
- Combining the power of the Hacker Board and the NVIDIA JETSON Nano® allows users to implement research-level, real-time algorithms such as AI & Computer Vision, SLAM and autonomous driving algorithms using ROS.
- The Puzzlebot NVIDIA JETSON® Edition works by communicating the Hacker Board (Plug and play) with the NVIDIA Jetson Nano®.

Puzzlebot: NVIDIA Jetson ® Edition

- The following slides will guide the user through the basic usage of the Puzzlebot Jetson ® Edition.
- The initial configuration will take place in two steps
  - The Hackeboard configuration: In this configuration, the user will learn how to
  - The NVIDIA Jetson® configuration.



System Overview

NVIDIA Jetson Nano

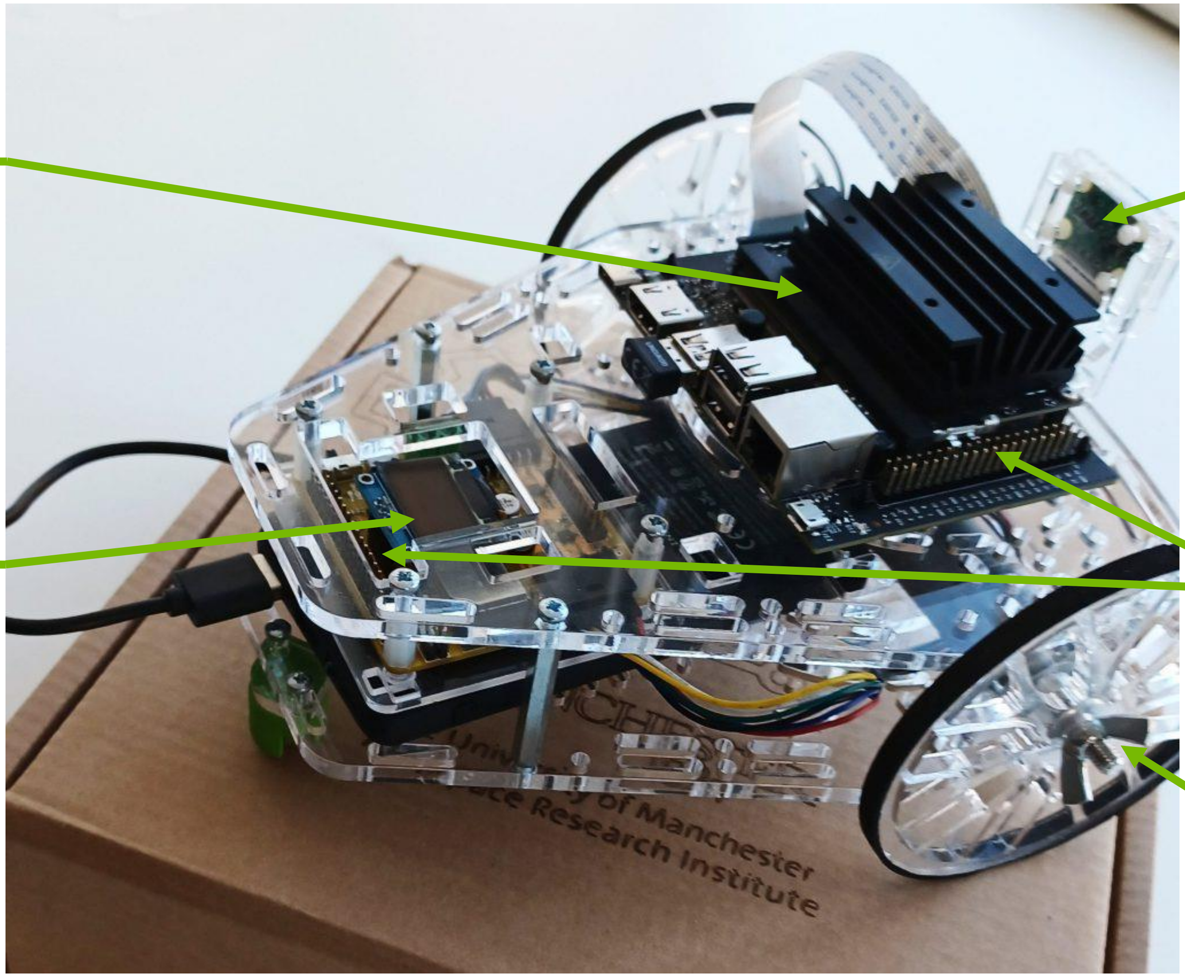
For AI and computer vision

- Higher processing power
- Time-sharing operating system
- Good for more complex, slower tasks
- Specifically designed by NVIDIA for AI applications

Hackerboard

For low-level control algorithms

- Low processing power
- Real-time operating system
- Good for simple, fast, time-sensitive tasks



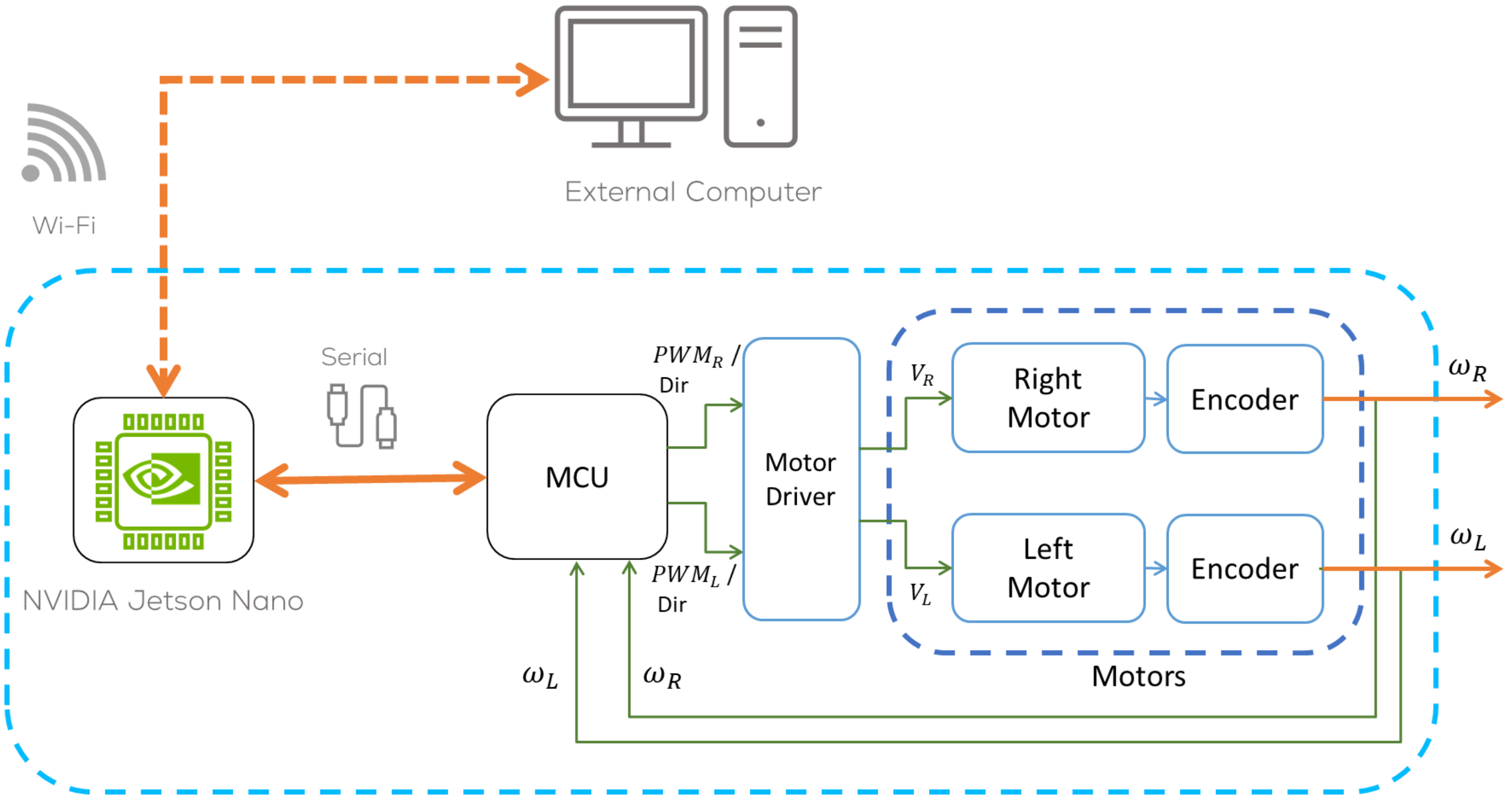
Raspberry Pi Camera

GPIO Arrays

Expansion is possible via the Jetson or the Hacker Board.

Puzzlebot Hacker Edition

Puzzlebot Diagram



Puzzlebot Diagram*

Hackerboard

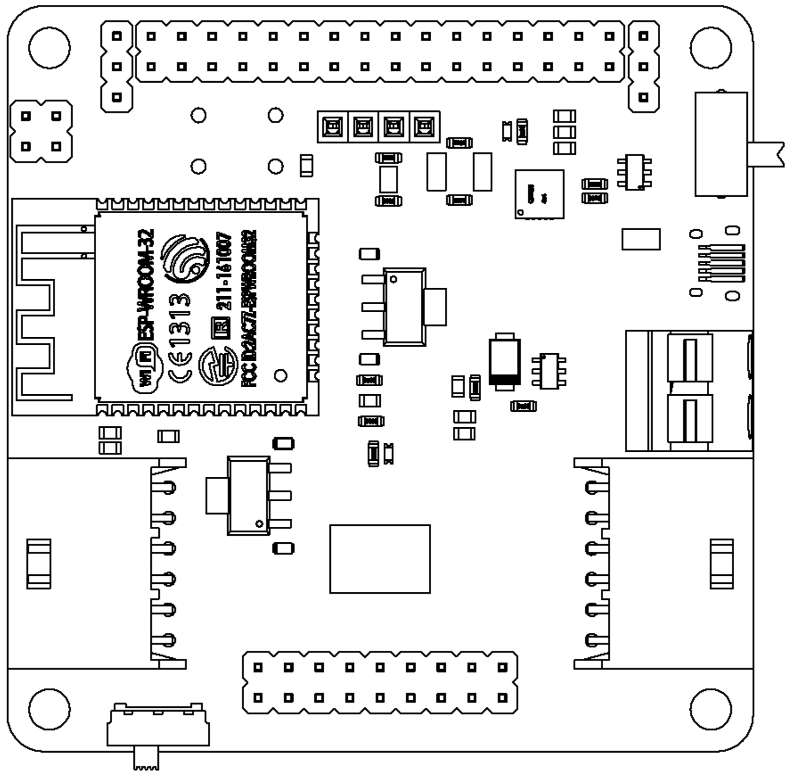
Introduction

{Learn, Create, Innovate};

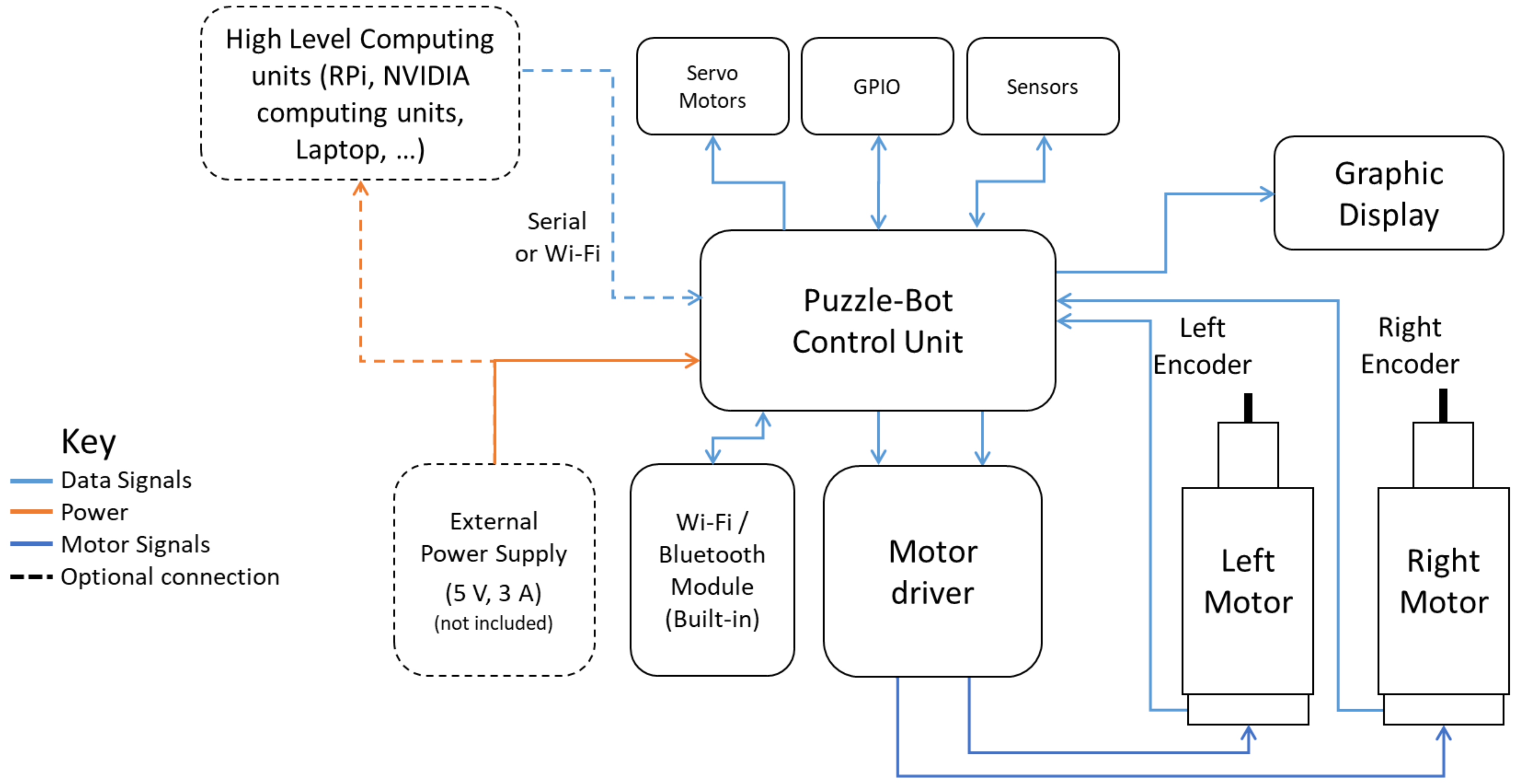
The Hacker Board

General characteristics

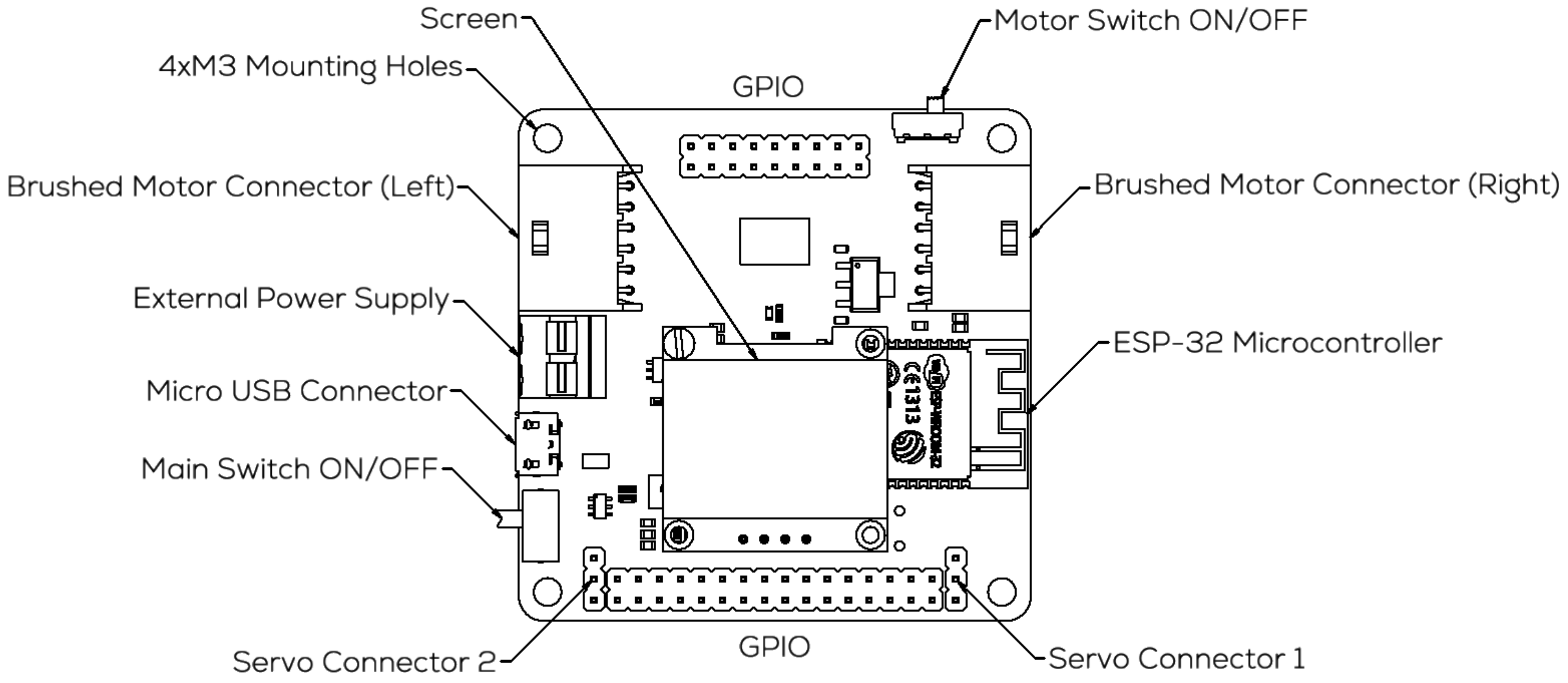
- ESP32-based Microcontroller
  - Xtensa dual-core 32-bit LX6 microprocessor
  - 520 KB of SRAM
  - Wi-Fi & Bluetooth
- DC-DC Converter
- Motor Driver
- 0.96" I2C LCD Display



System Architecture



The Hacker Board



For additional details, please refer to the pinout diagram.

Hackerboard

Control Modes

The Hackerboard has two different control modes depending on the user's requirements.

- The two programming configurations:
  - Standalone Configuration
  - External-Control Configuration

Control Mode: Standalone Configuration (Information purpose only)

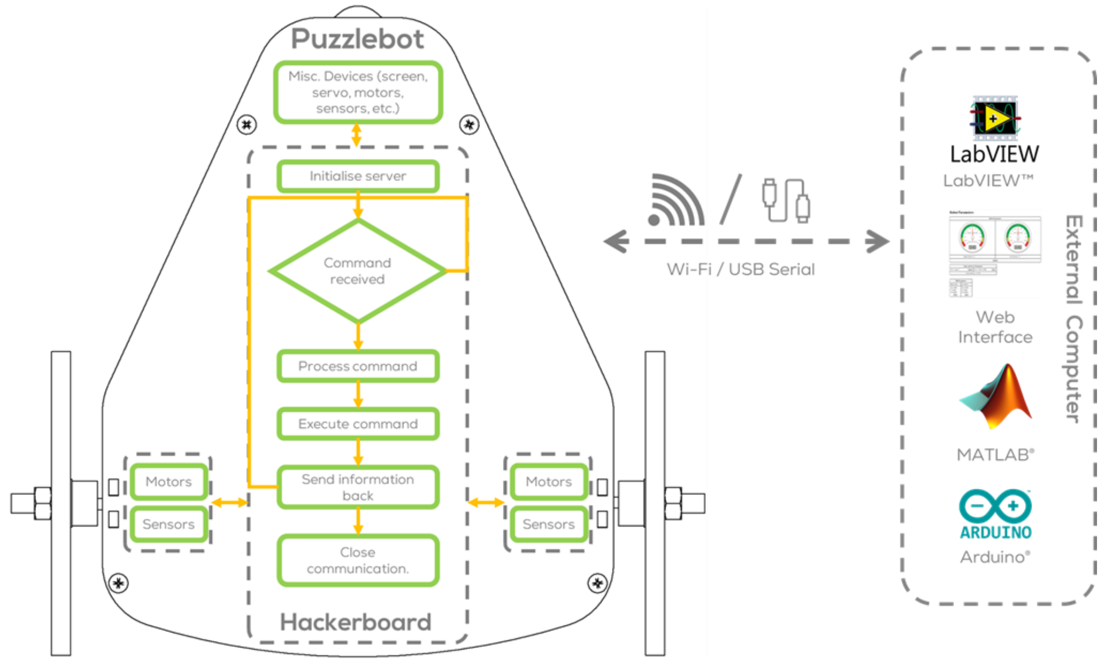
- The user directly programs the Hacker Board, using the Arduino IDE.
- Libraries for control and communication with computing units, sensors, and actuators are provided by MCR2.
- 3rd Party peripherals can be attached.
- This configuration will not be used for this Puzzlebot version.
- For additional examples and a more in-depth understanding, please consult the "Puzzlebot Hacker Edition" manual.

Hackerboard

Control Mode: Standalone Configuration

The robot is controlled from an external computer via Wi-Fi or Serial Communication.

- The internal firmware and libraries for communicating with the robot's sensors and actuators are provided by MCR2.
- Basic web interface for configuring and testing provided.
- MCR2 provides MATLAB , ROS and LabVIEW libraries for communicating with the robot.
- This configuration will be used for this Puzzlebot version.
- For additional examples and a more in-depth understanding, on how to use MATLAB or LabVIEW communications and simulators, contact us.



Hackerboard

Flashing the Hackerboard

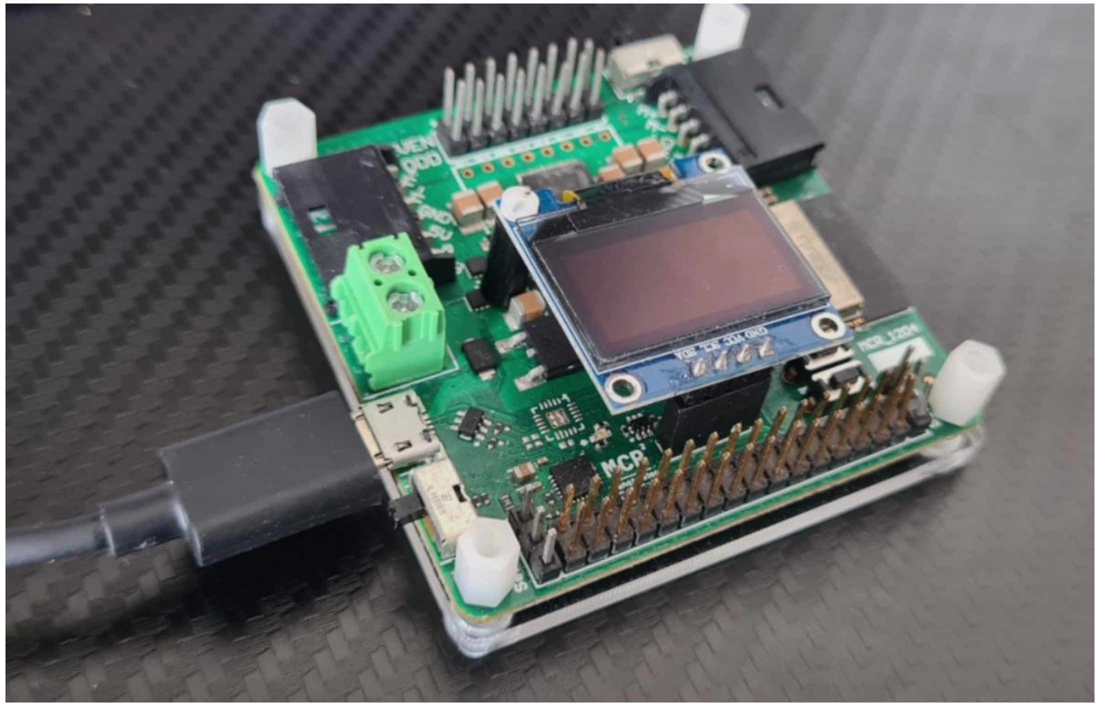
{Learn, Create, Innovate};

MCR² Manchester Robotics

Hackerboard: Flashing the Binaries

Flashing Hackerboard

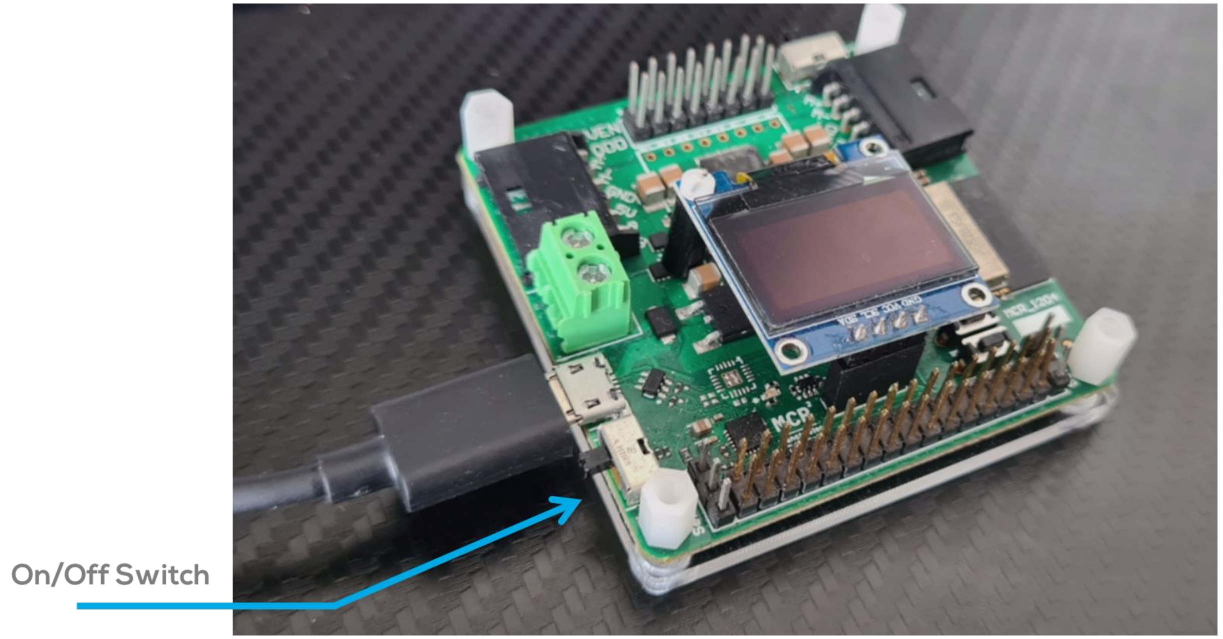
- MCR2 provides the firmware binaries for the External Computing Unit Control Mode.
- This section will guide the user on how to flash such binaries.
- All the robots come preprogrammed with such binaries unless the user modifies the program (On Board Configuration).
- The original binaries can be flashed anytime by following the steps in this section.



Hackerboard: Flashing the Binaries.

Steps

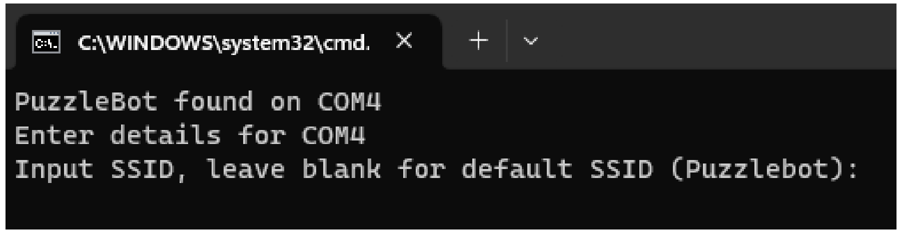
1. Attach the micro-USB cable to the Puzzle-Bot Control Module. *Note the image is just for explanation purposes, detachment of the Puzzle-Bot Control module from the robot is not necessary.*
2. Make sure the Hackerboard power switch is turned on.
3. Connect the USB to any free USB port in the computer
4. Download the firmware from the MCR2 GitHub and extract it.
5. Select the OS of the computer that will perform the flash.
6. Run the file *"FirmwareFlash"*
   - For Linux and MacOS users, it may be necessary to set execution permissions in properties.



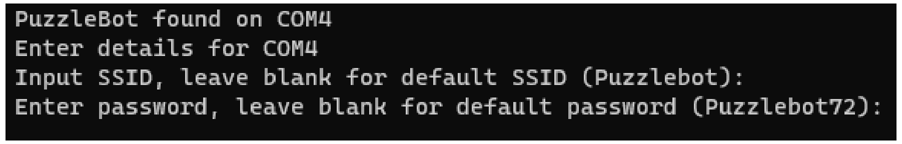
Hackerboard: Flashing the Binaries.

Steps

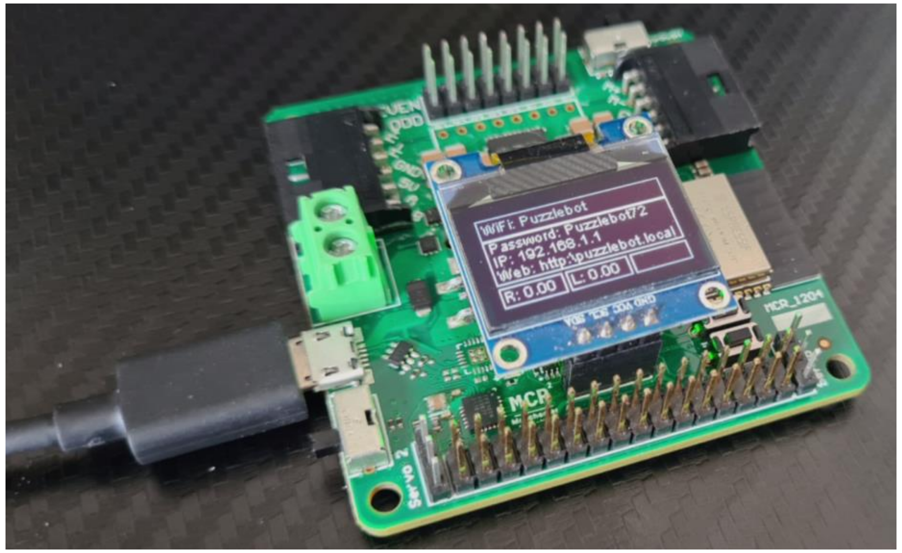
7. Select the SSID identification for the Hackerboard and press Enter (recommended to use a different one if several Puzzlebots are being used simultaneously).
8. Select the password for the Hackerboard's Wi-Fi Network and press Enter (recommended to use a different one if several Puzzlebots are being used simultaneously).
9. Wait until the program finish flashing.
10. The screen must turn on.



```
PuzzleBot found on COM4
Enter details for COM4
Input SSID, leave blank for default SSID (Puzzlebot):
```



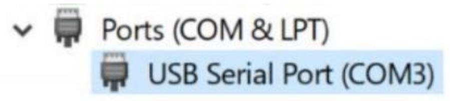
```
PuzzleBot found on COM4
Enter details for COM4
Input SSID, leave blank for default SSID (Puzzlebot):
Enter password, leave blank for default password (Puzzlebot72):
```



Troubleshoot

Troubleshoot (Drivers)

- Drivers are usually installed automatically by Windows and Ubuntu even for the Virtual Machines.
- How do I know if the drivers are properly installed (Windows)?
  - Plug the Puzzle-Bot into the USB port.
  - Go to Start › Device Manager
  - The Serial port should appear as shown in the following figure (The COM port may vary).

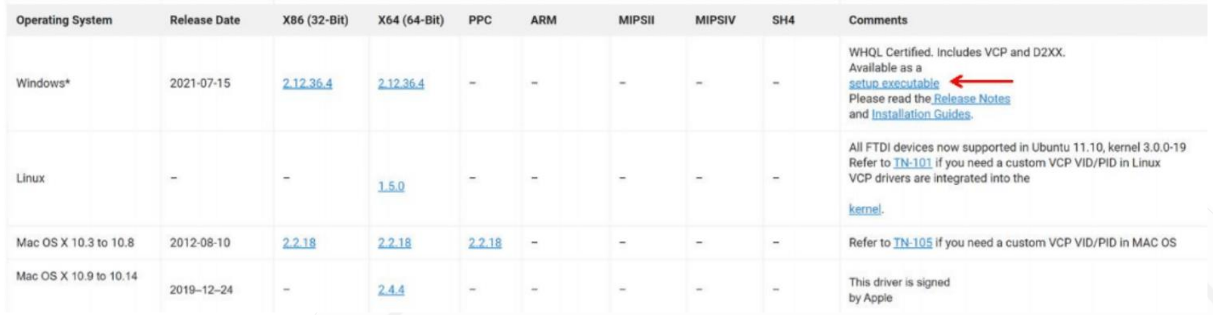


- If the computer cannot find the drivers, download the drivers from the following link

  https://ftdichip.com/drivers/vcp-drivers/

- Verify that the USB cable is a data cable and not only a power cable!

- Scroll down and download the executable setup as shown in the following figure



Operating System	Release Date	X86 (32-Bit)	X64 (64-Bit)	PPC	ARM	MIPSII	MIPSIV	SH4	Comments
Windows*	2021-07-15	2.12.36.4	2.12.36.4	-	-	-	-	-	WHQL Certified. Includes VCP and D2XX. Available as a setup executable Please read the Release Notes and Installation Guides.
Linux	-	-	1.5.0	-	-	-	-	-	All FTDI devices now supported in Ubuntu 11.10, kernel 3.0.0-19 Refer to TN-101 if you need a custom VCP VID/PID in Linux VCP drivers are integrated into the kernel.
Mac OS X 10.3 to 10.8	2012-08-10	2.2.18	2.2.18	2.2.18	-	-	-	-	Refer to TN-105 if you need a custom VCP VID/PID in MAC OS
Mac OS X 10.9 to 10.14	2019–12–24	-	2.4.4	-	-	-	-	-	This driver is signed by Apple

Before Installing the drivers!!

- Unplug the Puzzlebot from the computer.
- Unzip the drivers and run the setup (some computers must be restarted after the installation).
- Plug the Puzzlebot back into the computer.

Troubleshoot

Troubleshoot (Drivers)

- Some Hackerboard have a different USB-UART chip the CP210x.
- Drivers are usually installed automatically by Windows and Ubuntu even for the Virtual Machines.
- Verify if they are installed by following the steps in the previous slide.
- Verify that the USB cable is a data cable and not only a power cable!.
- If the computer cannot find the drivers, download the drivers from the following link

https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers?tab=downloads

Before Installing the drivers!!

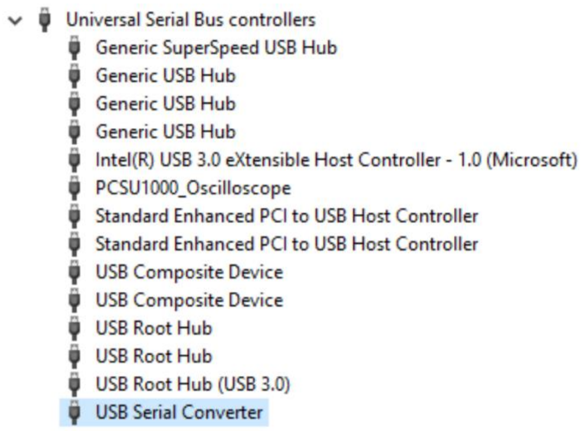
- Unplug the Puzzle-Bot from the computer.
- Unzip the drivers and run the setup (some computers are required to be restarted after the installation).
- Plug the Puzzle-Bot back into the computer.

A troubleshoot guide can be found here.

Troubleshoot

Troubleshoot (Drivers)

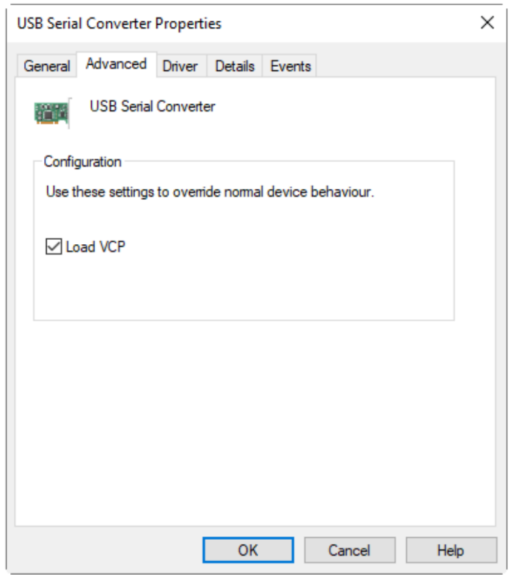
- My computer still not recognize the drivers even after the installation
- Plug the Puzzle-Bot into the USB port.
- Go to Start › Device Manager.
- Look for the USB Serial Converter as shown in the following picture.



Universal Serial Bus controllers
- Generic SuperSpeed USB Hub
- Generic USB Hub
- Generic USB Hub
- Generic USB Hub
- Intel(R) USB 3.0 eXtensible Host Controller - 1.0 (Microsoft)
- PCSU1000_Oscilloscope
- Standard Enhanced PCI to USB Host Controller
- Standard Enhanced PCI to USB Host Controller
- USB Composite Device
- USB Composite Device
- USB Root Hub
- USB Root Hub
- USB Root Hub (USB 3.0)
- USB Serial Converter

FIGURE: USB SERIAL CONVERTER

- Right Click to Properties › Advanced Tab.
- Make sure the Load VCP box is checked.
- Reconnect the Puzzle-Bot to the computer.



USB Serial Converter Properties

USB Serial Converter

Configuration

Use these settings to override normal device behaviour.

☑ Load VCP

FIGURE: VCP PORT

Hackerboard

Connections and configurations

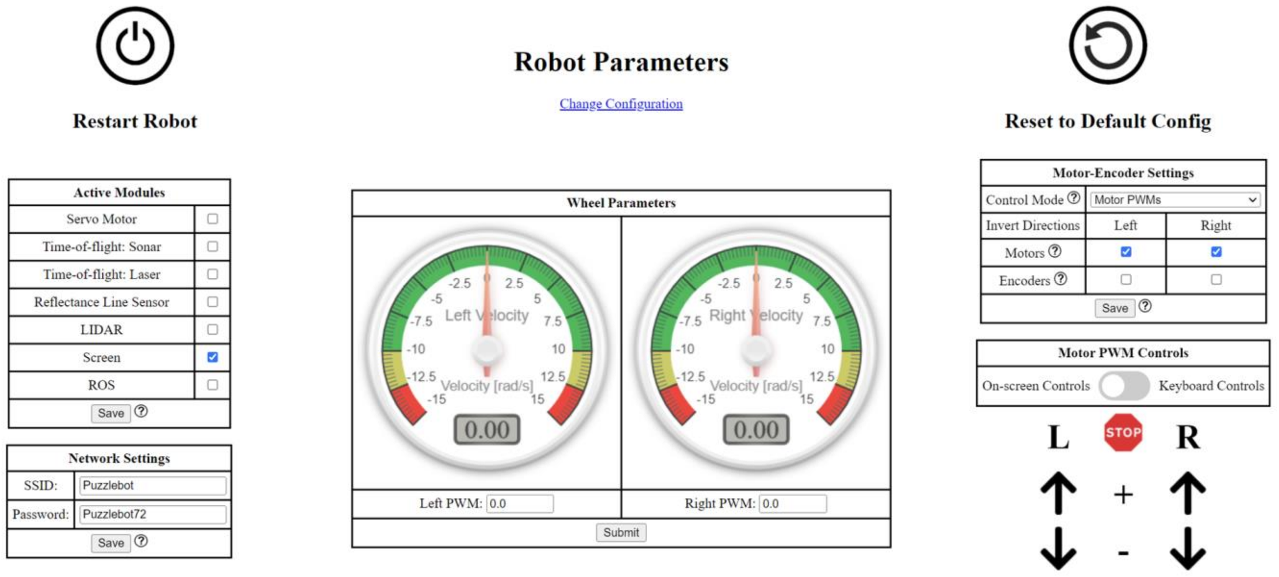
{Learn, Create, Innovate};

Connecting to the Hackerboard

Puzzlebot Web Interface

The Puzzlebot has a web interface allowing it to configure and test the different sensors and actuators equipped in the robot.

The website offers a visual interface that enables the user to configure and test various internal settings of the robot, including motor controllers, sensor activation, actuator activation (if applicable), and communication with external computing units.



Restart Robot

Robot Parameters

Change Configuration

Reset to Default Config

Active Modules	
Servo Motor	☐
Time-of-flight: Sonar	☐
Time-of-flight: Laser	☐
Reflectance Line Sensor	☐
LIDAR	☐
Screen	☑
ROS	☐
Save ⓘ	

Network Settings	
SSID:	Puzzlebot
Password:	Puzzlebot72
Save ⓘ	

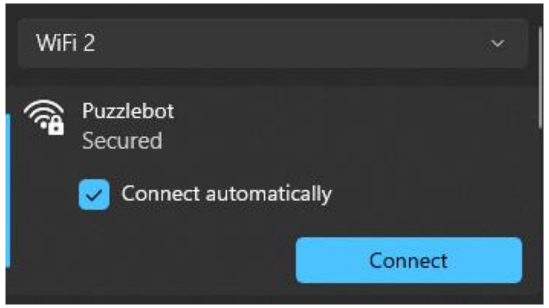
Wheel Parameters	
Left Velocity 0.00	Right Velocity 0.00
Left PWM: 0.0	Right PWM: 0.0
Submit	

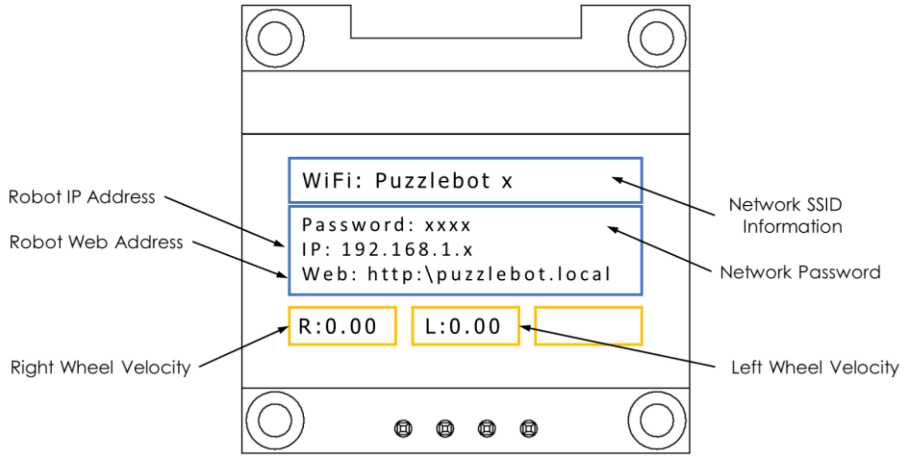
Motor-Encoder Settings		
Control Mode ⓘ	Motor PWMs	
Invert Directions	Left	Right
Motors ⓘ	☑	☑
Encoders ⓘ	☐	☐
Save ⓘ		

Motor PWM Controls
On-screen Controls / Keyboard Controls

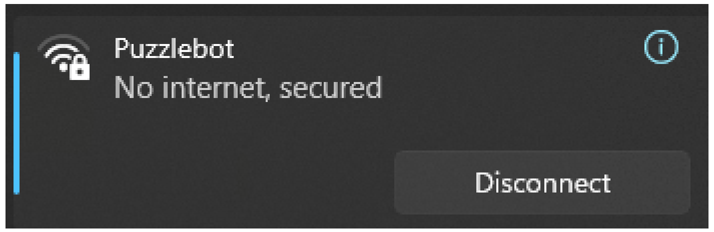
Connecting to the Hackerboard

1. Go to the Wi-Fi Network connections.
2. Connect to the Wi-Fi network created by the Puzzlebot by choosing the Puzzle-Bot Wi-Fi network of the robot you want (Puzzlebot x), then select Connect.
3. Type the network password that you selected when flashing (shown on the LCD Display), and then select Next.
4. Once Connected the network should say "No internet, secured" (Windows only).


WiFi 2
Puzzlebot
Secured
Connect automatically
Connect



```
WiFi: Puzzlebot x
Password: xxxx
IP: 192.168.1.x
Web: http:\puzzlebot.local
R:0.00  L:0.00
```


Puzzlebot
No internet, secured
Disconnect

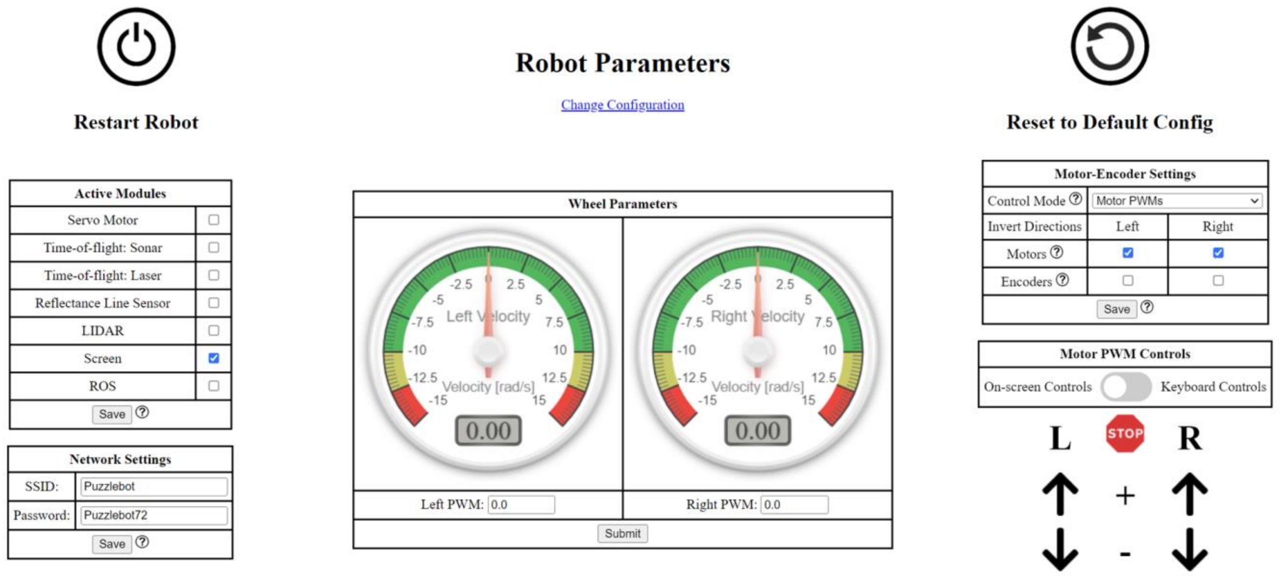
Connecting to the Hackerboard

Connecting to the Puzzlebot Web Interface

1. Open a web browser
2. Type the following IP address on the search bar

```
http:\\192.168.1.x
```

3. The Puzzlebot Robot Parameters interface should load.



Restart Robot

Robot Parameters

Change Configuration

Reset to Default Config

Active Modules	
Servo Motor	☐
Time-of-flight: Sonar	☐
Time-of-flight: Laser	☐
Reflectance Line Sensor	☐
LIDAR	☐
Screen	☑
ROS	☐
Save ⓘ	

Network Settings	
SSID:	Puzzlebot
Password:	Puzzlebot72
Save ⓘ	

Wheel Parameters	
Left Velocity 0.00	Right Velocity 0.00
Left PWM: 0.0	Right PWM: 0.0
Submit	

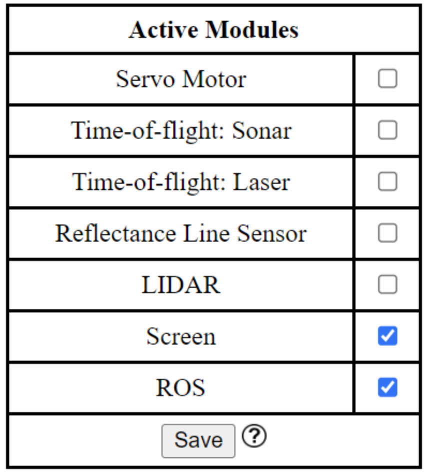
Motor-Encoder Settings		
Control Mode ⓘ	Motor PWMs	
Invert Directions	Left	Right
Motors ⓘ	☑	☑
Encoders ⓘ	☐	☐
Save ⓘ		

Motor PWM Controls
On-screen Controls / Keyboard Controls

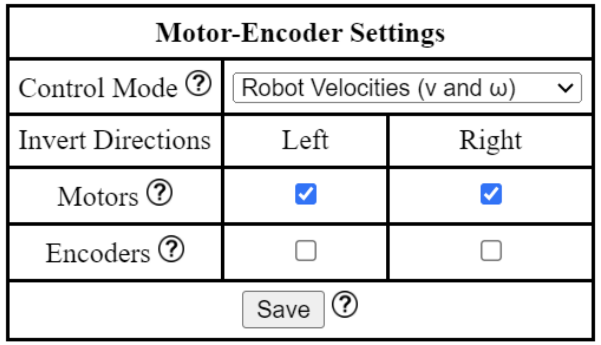
Connecting to the Hackerboard

Configuring ROS communication

5. Activate ROS; select the option in Sensor/actuators and comms. Modules.
6. Press the Save button.
7. The ROS topics to be published by the Puzzlebot depend on the "Control Mode" Selected.
   - See table in next slide.
8. Having selected the Control Mode and activated the ROS Module, press Save and disconnect from the robot.



Active Modules	
Servo Motor	☐
Time-of-flight: Sonar	☐
Time-of-flight: Laser	☐
Reflectance Line Sensor	☐
LIDAR	☐
Screen	✓
ROS	✓
Save ?	



Motor-Encoder Settings		
Control Mode ?	Robot Velocities (v and ω)	
Invert Directions	Left	Right
Motors ?	✓	✓
Encoders ?	☐	☐
Save ?		

Connecting to the Hackerboard

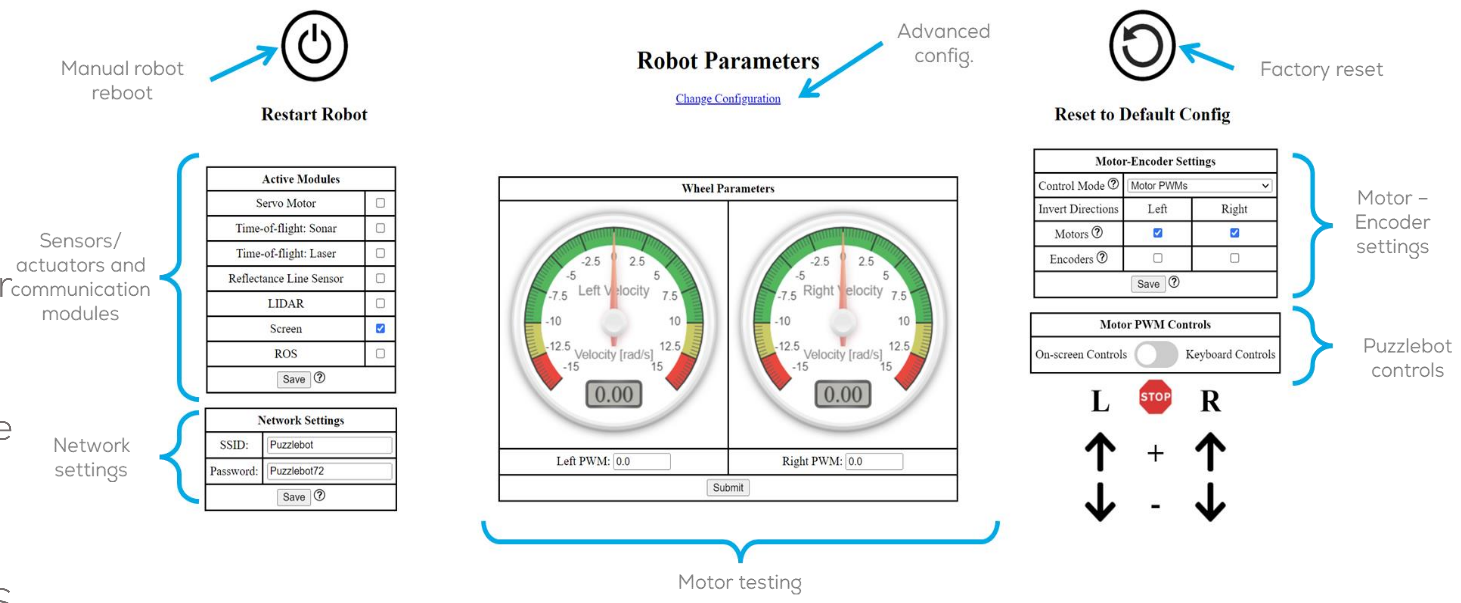
Control Mode	Description	Topic	Type	Information
Motor PWMs	Wheel PWM voltage signal	/cmd_pwmL, /cmd_pwmR	std_msgs/Float32	data: PWM duty cycle for each motor [-1,1]
Wheel velocities (ωL and ωR)	Wheel angular velocities setpoint (PID control)	/cmd_wL, /cmd_wR	std_msgs/Float32	data: Control Set Point for wheel velocities
Robot Velocities (v and ω)	Linear and Angular Velocities (PID control)	/cmd_vel	geometry_msgs/Twist	linear x: Linear speed of the robot y: Not used z: Not Used angular: x: Not Used y: Not Used z: Angular Speed of the robot

Configuring the Hackerboard

**This slide is only for informative purposes and is not required for basic Jetson connection.

- **Manual robot reboot:** Manually reboots the Hackerboard.
- **Factory Reset:** Reset all the configurations to factory configurations.
- **Network settings:** Configuration of the SSID and password of the Puzzlebot. Useful when multiple Puzzlebots are being used.
- **Puzzlebot controls:** Simple Puzzlebot controls for the user to move the robot forward, backwards or turning.
- **Sensor/actuators and comms. modules:** Activation of the sensors/actuators if included. ROS communication must be active to be used with the NVIDIA Jetson Nano. The sensors and actuators in this section can connect to ROS.

Moving the cursor over the question marks (?) displays more information about each configuration parameter.



Restart Robot

Robot Parameters

Change Configuration

Reset to Default Config

Active Modules	
Servo Motor	☐
Time-of-flight: Sonar	☐
Time-of-flight: Laser	☐
Reflectance Line Sensor	☐
LIDAR	☐
Screen	✓
ROS	☐
Save ?	

Network Settings	
SSID:	Puzzlebot
Password:	Puzzlebot72
Save ?	

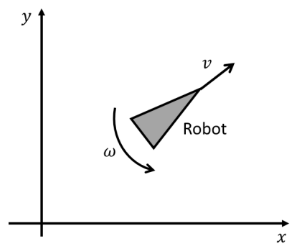
Wheel Parameters	
0.00	0.00
Left PWM: 0.0	Right PWM: 0.0
Submit	

Motor-Encoder Settings		
Control Mode ?	Motor PWMs	
Invert Directions	Left	Right
Motors ?	✓	✓
Encoders ?	☐	☐
Save ?		

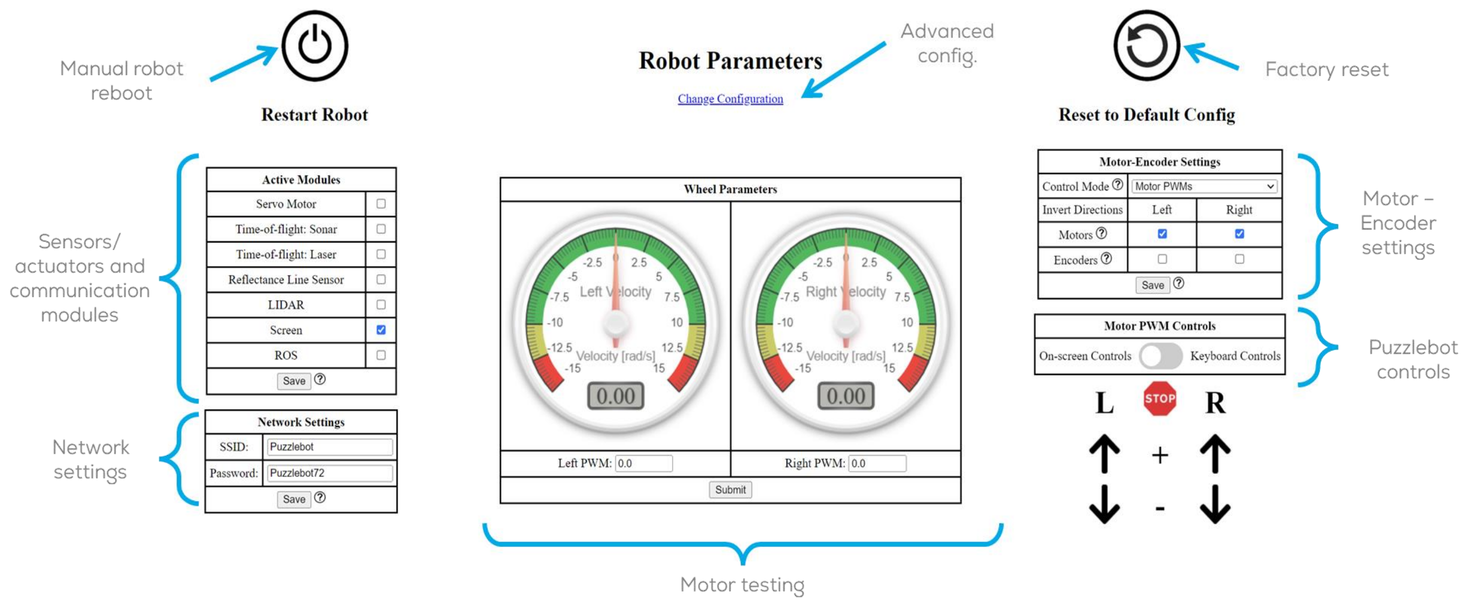
Motor PWM Controls
On-screen Controls ○ Keyboard Controls

Configuring the Hackerboard

- **Advanced configuration:** Manually configure all aspects of the Puzzlebot. **Warning! adjusting the parameters of the Puzzlebot could result in significant malfunctions.**
- **Motor-Encoder Settings:** Motor and Encoder configurations.
  - Motor PWM: Values in the range of [-1, 1] (No control)
  - Wheel velocities (ωL and ωR): Wheel velocities (Inner PID control used)
  - Robot Velocities (v and ω): Linear and angular speed of the robot.



- **Motor testing:** Manually send values to the motors for testing. The values sent depend on the Control Mode.



Restart Robot

Robot Parameters

Change Configuration

Reset to Default Config

Active Modules	
Servo Motor	☐
Time-of-flight: Sonar	☐
Time-of-flight: Laser	☐
Reflectance Line Sensor	☐
LIDAR	☐
Screen	☑
ROS	☐

Save

Network Settings	
SSID:	Puzzlebot
Password:	Puzzlebot72

Save

Wheel Parameters

Left PWM: 0.0 Right PWM: 0.0

Submit

Motor-Encoder Settings		
Control Mode	Motor PWMs	
Invert Directions	Left	Right
Motors	☑	☑
Encoders	☐	☐

Save

Motor PWM Controls

On-screen Controls Keyboard Controls

NVIDIA Jetson Nano®

Connections and configurations

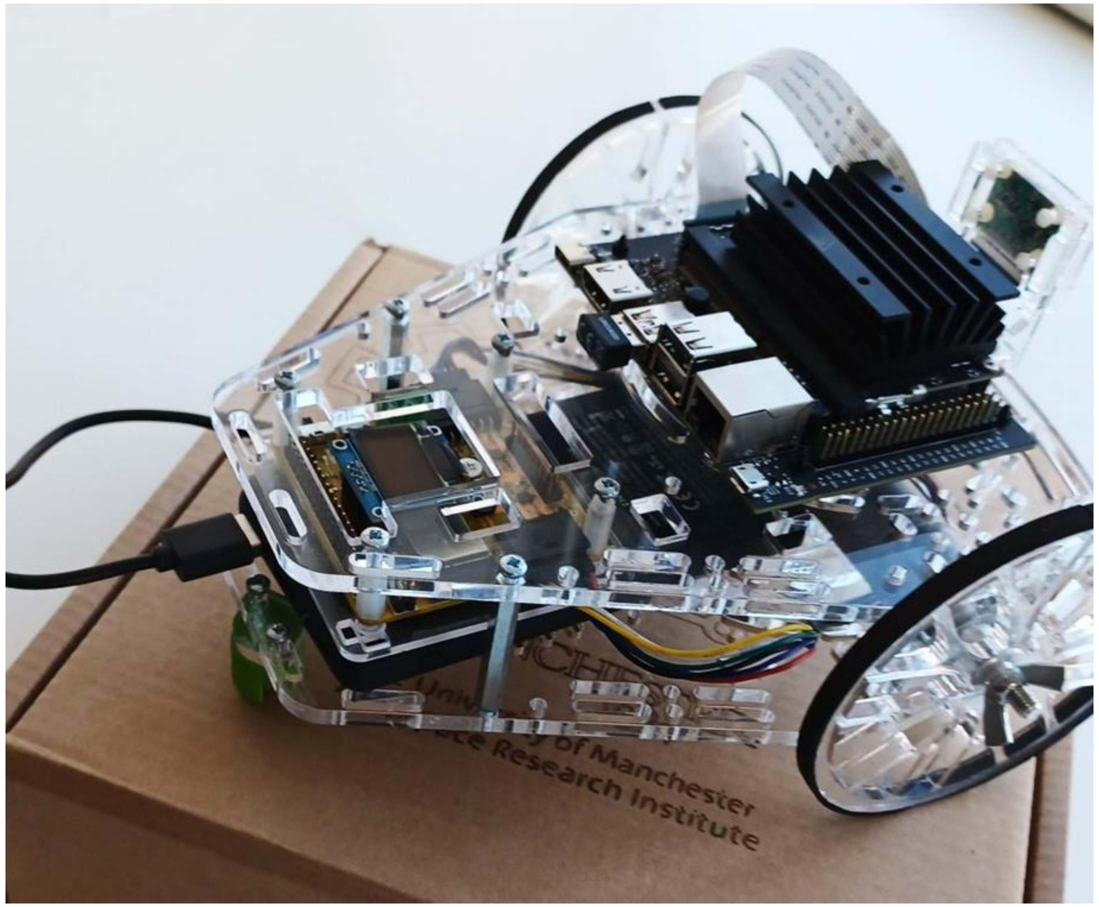
{Learn, Create, Innovate};

MCR²
Manchester Robotics

NVIDIA Jetson Nano®, 2 GB

Overview

- MCR2 provides a modified version of the Jetson Image, with some preloaded packages, that connects with the Hackerboard (plug and play).
- This section will guide the user on how to flash such image.
- The original image can be flashed anytime by following the steps in this section.
- This guide will help you start using the Jetson board on the Puzzlebot.



NVIDIA Jetson Nano®, 2 GB

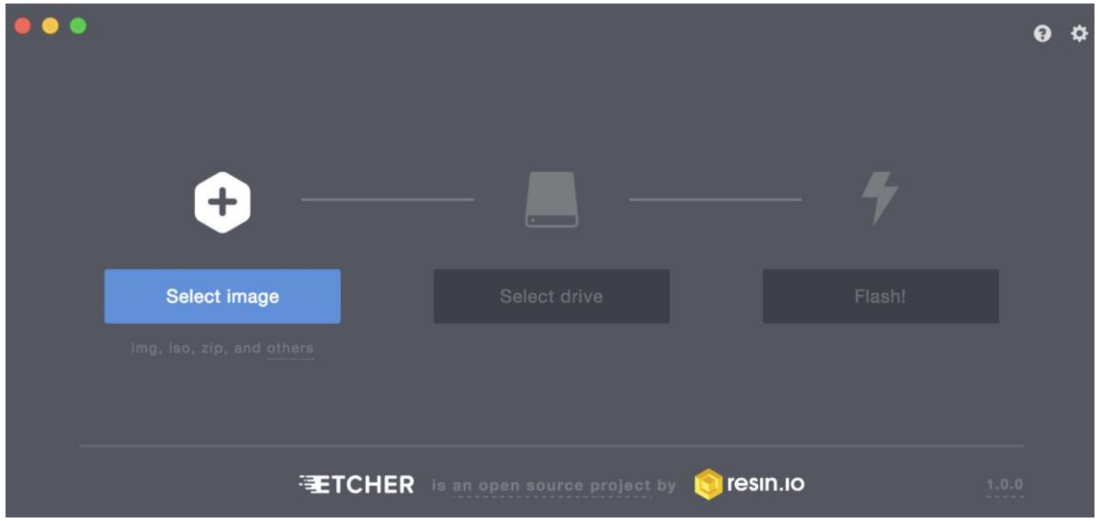
General characteristics

- 128-core NVIDIA Maxwell GPU
- 1.43 GHz Quad-core ARM A57 CPU
- 2 GB of 64-bit LPDDR4 Memory
- SD card for storage
- Ethernet & Wi-Fi
- CSI-2 Connector for Camera
- Runs a modified version of Ubuntu 18.04, with the following preinstalled software:
  - ROS
  - TensorFlow
  - OpenCV
  - Nvidia Camera nodes

MCR2 Image Installation

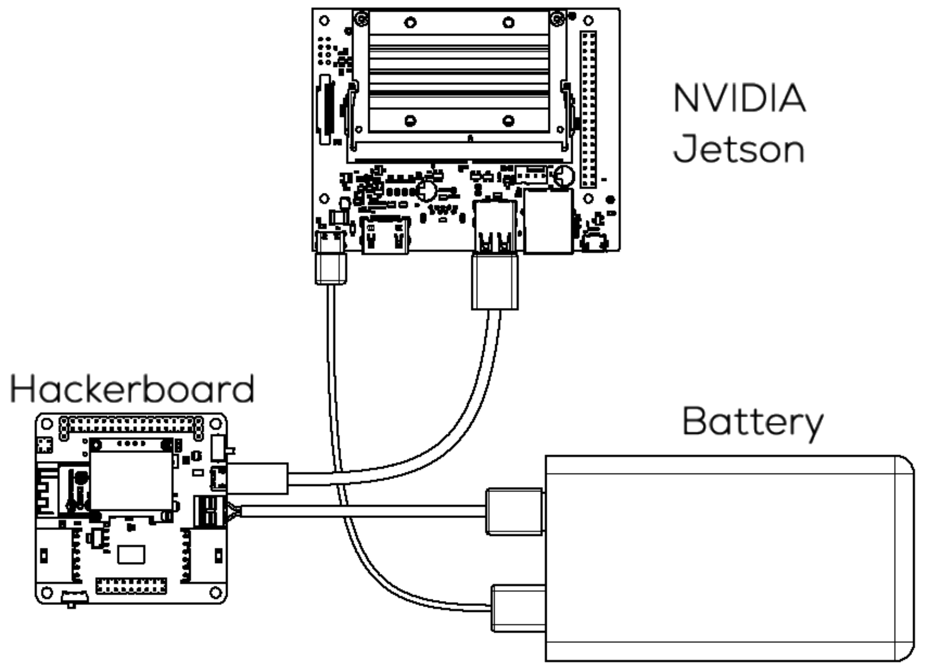
General characteristics

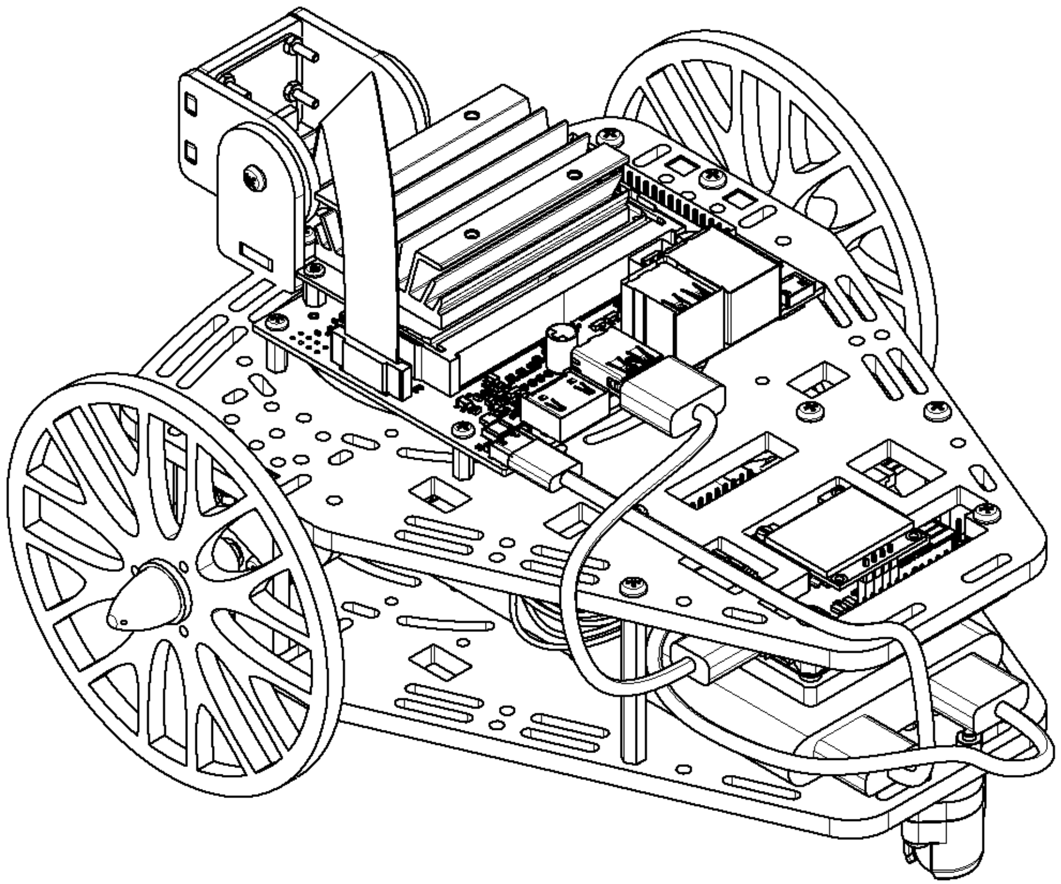
- The OS for the Jetson is stored on an SD card
- An image must be flashed to the SD card; download it from here
- This is flashed from an image using the Balena Etcher tool
- To flash the SD card:
    1. Insert the SD into your PC
    2. Launch the etcher software
    3. Select the image downloaded 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.


Select image
img, iso, zip, and others
Select drive
Flash!
ETCHER is an open source project by resin.io
1.0.0

Electrical Connections

Electrical Connection Diagram





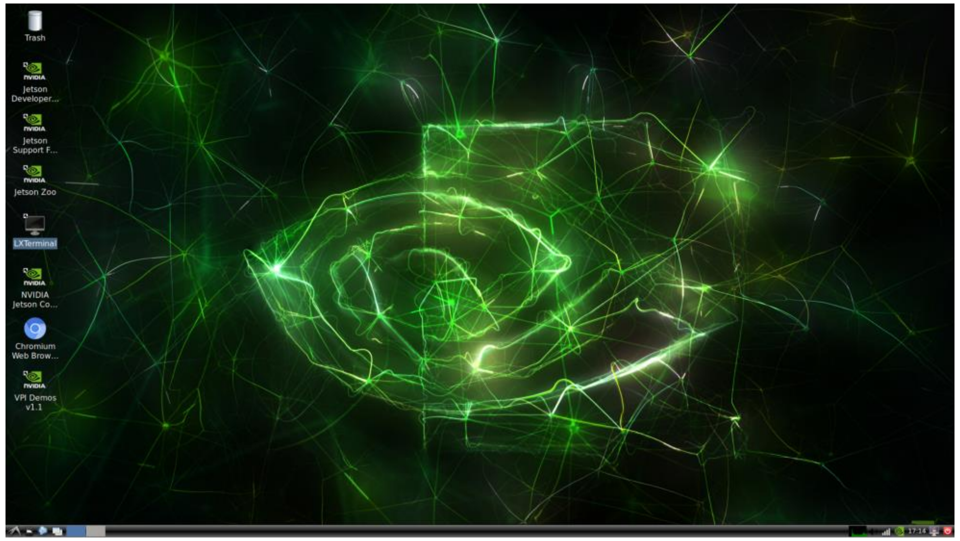
*The battery must be able to provide 5V, 3A per port independently.

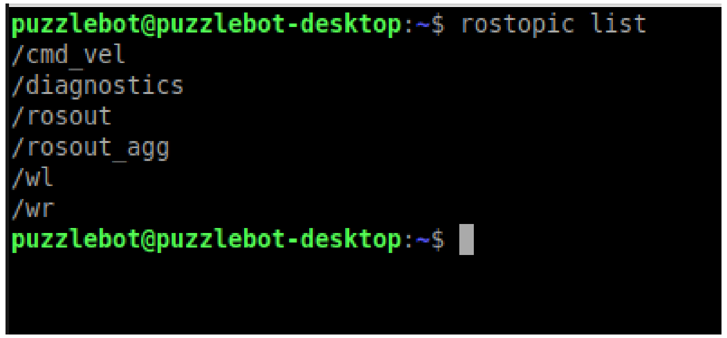
NVIDIA Jetson Nano®, 2 GB

Initial Setup

1. Connect the Hackerboard to the Jetson.
2. Connect the Jetson Nano to a screen
   - The Hacker Board communication starts each time the Jetson is booted up.
3. Connect the Jetson to the battery
4. Open a terminal *"LXTerminal"*
5. Type "*rostopic list*".
   - You should see a list of topics as shown, although this will depend on which Control Mode the Hackerboard uses.
6. Publish into the topics to make the robot move
   - `rostopic pub /wl std_msgs/Float32 data:0.8`

- If the communication fails, the protocol can be restarted with the command:
  - `sudo systemctl restart puzzlebot.service`
- *Make sure the Hackerboard is turned On, and the Motor Switch is turned On.



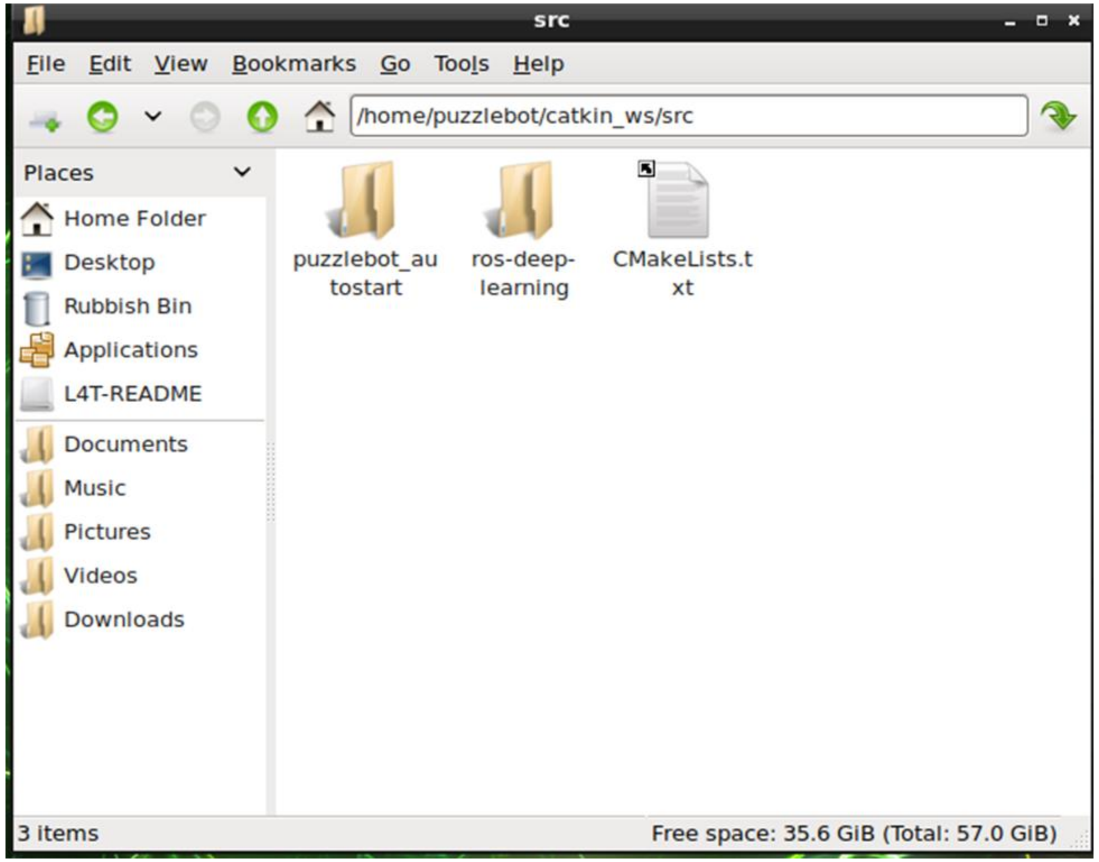


```
puzzlebot@puzzlebot-desktop:~$ rostopic list
/cmd_vel
/diagnostics
/rosout
/rosout_agg
/wl
/wr
puzzlebot@puzzlebot-desktop:~$ 
```

MCR2 Jetson Image

General Information

- The Hackerboard starts a *roscore* each time the Jetson is booted up, allowing the communication with the Hackerboard to start.
- There is a pre-setup catkin workspace on the Jetson.
- The *roscore* starts the launch file inside the "*puzzlebot_autostart*" folder.
    - The user can modify this launch file to launch their own nodes every time the Puzzlebot starts automatically.
- The "*ros-deep-learning*" is an NVIDIA library for ROS, necessary for the Puzzlebot and camera communication, and should not be changed in any way



MCR2 Jetson Image

General Information

The *roscore* autostart is done via a service launched at boot.

- To check the status of the service, open a terminal and type

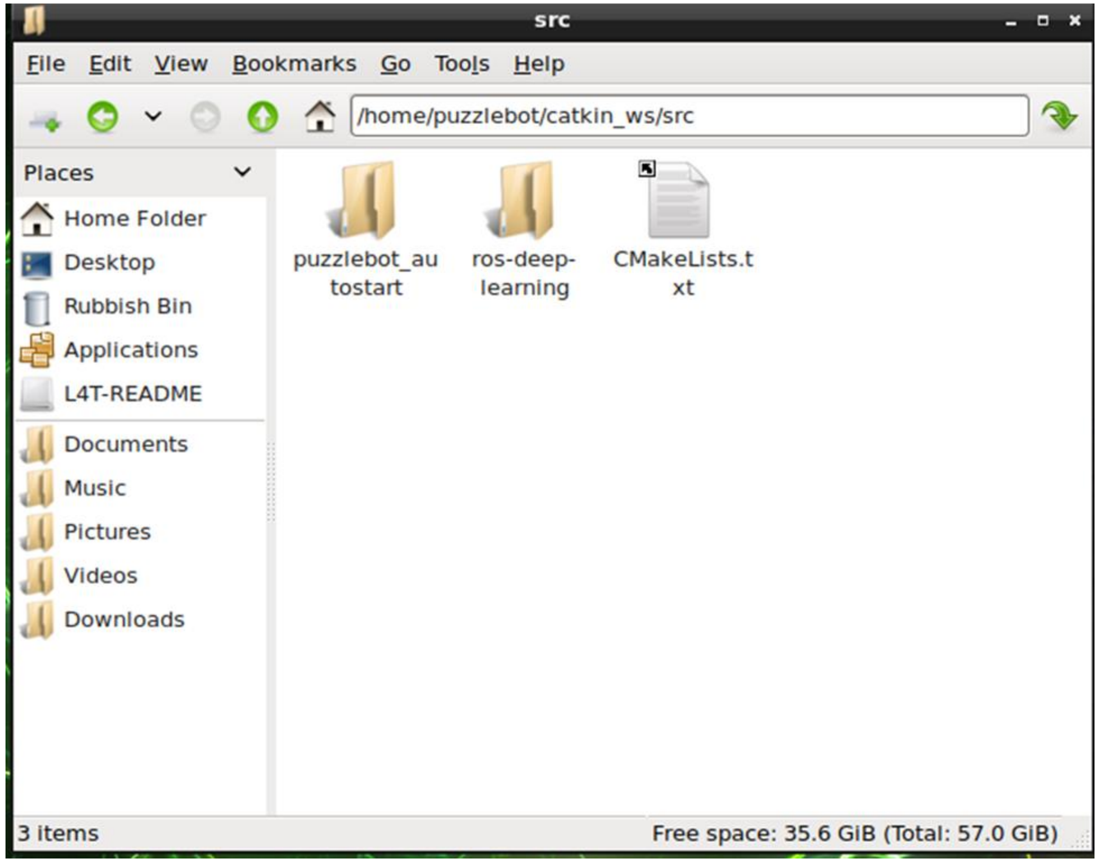
```
sudo systemctl status puzzlebot.service
```

- To restart the service, in case the communication fails, open a terminal and type

```
sudo systemctl restart puzzlebot.service
```

- To stop the service if the user wants to use the Jetson without ROS, to connect to the Internet open a terminal and type

```
sudo systemctl stop puzzlebot.service
```



Remote Access

1. It can be useful to control the Jetson from a remote PC, as the robot cannot be in motion and hooked up to a monitor
   - Another way to control the robot is via ROS network, which will be explained in the next slides.
2. To do this, SSH is used. It uses Wi-Fi to give your computer access to the Jetson.
3. The Jetson generates its own Wi-Fi network (Access point) for communication with an external device:
   - **Default Network Name:** PuzzlebotJetson
   - **Default Network Password:** Puzzlebot72

- **To prevent conflicts when many Jetsons are used, change the Network Name** to something unique (if applicable)
  - The Wi-Fi details can be changed on the Jetson by selecting:

    Networks->Edit Connections->Hotspot

Remote Access

4. On the external PC, open a cmd window and type:
   - `ssh puzzlebot@10.42.0.1`
   - Password: `Puzzlebot72`
5. If prompted, type `yes` and then press enter.
6. This command window is now equivalent to one running on the Jetson.
7. Any command can be run from this window, equivalent to running one on the Jetson.
8. Once any control code is written on the Jetson, it can be tested and debugged remotely via SSH, enabling the PuzzleBot to move around

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

Raspberry Pi Camera

- NVIDIA provides a package for interfacing with a CSI camera.
- To access the camera and test the connection type on a terminal

```
nvgstcapture-1.0
```

- To interface the camera with ROS, NVIDIA provides a ROS package that comes pre-installed on the Puzzlebot image
- For this package, 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
```

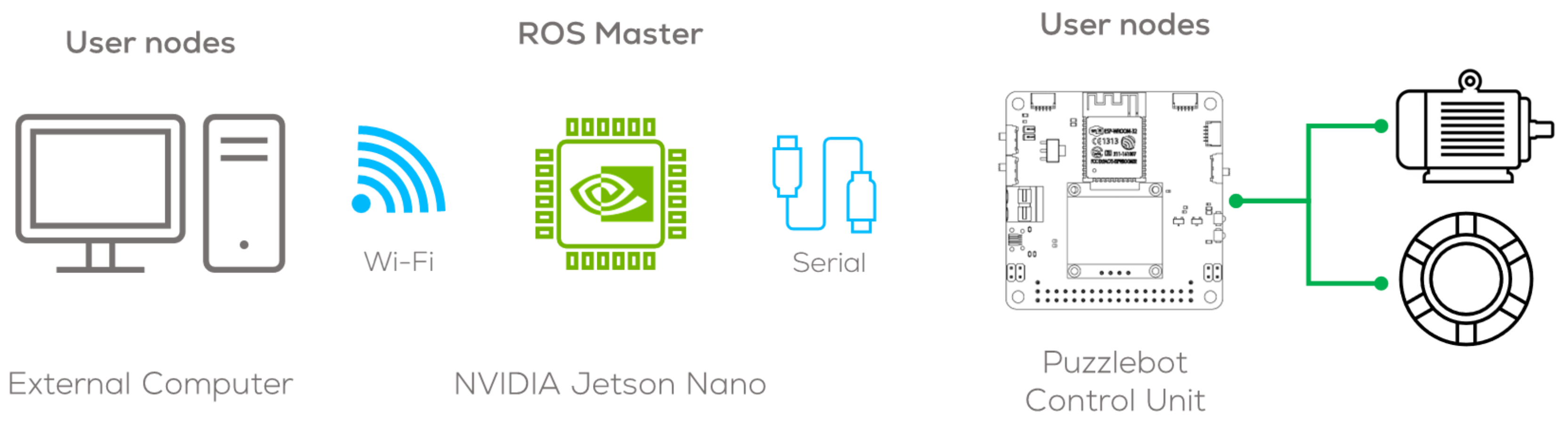
- The camera view should be displayed on the screen.
- To only publish the image to a topic without any viewer, run the following command

```
roslaunch ros_deep_learning video_source.ros1.launch
```

Multi-device Communication with ROS (ROS network)

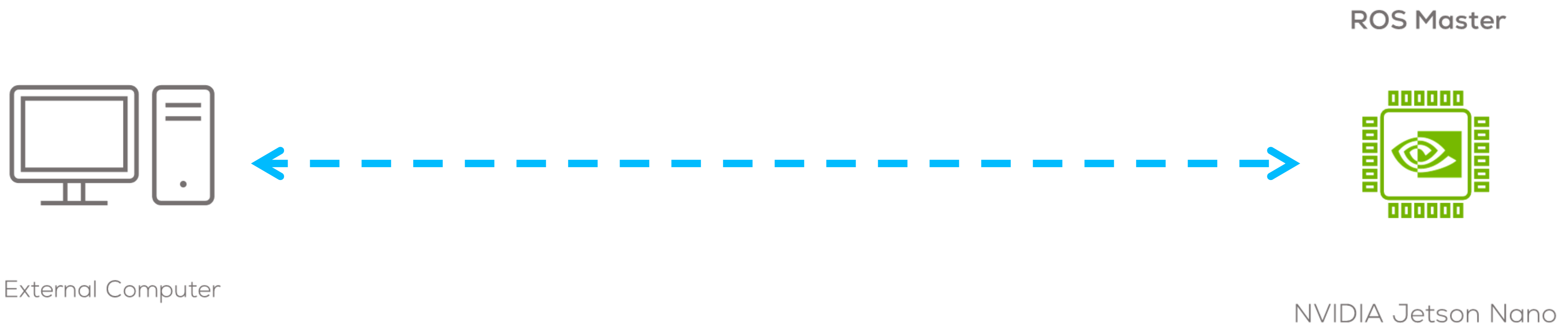
Introduction

- ROS facilitates communication between various devices on a shared network (No SSH).
- The Jetson Nano acts as an Access Point (AP) to provide this network.
- In the context of the Puzzlebot, the ROS master is typically run on the Jetson Nano. Then, by connecting to the Jetson's AP, other devices can communicate with the ROS master.



Multi-device Communication with ROS (ROS Network)

- 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



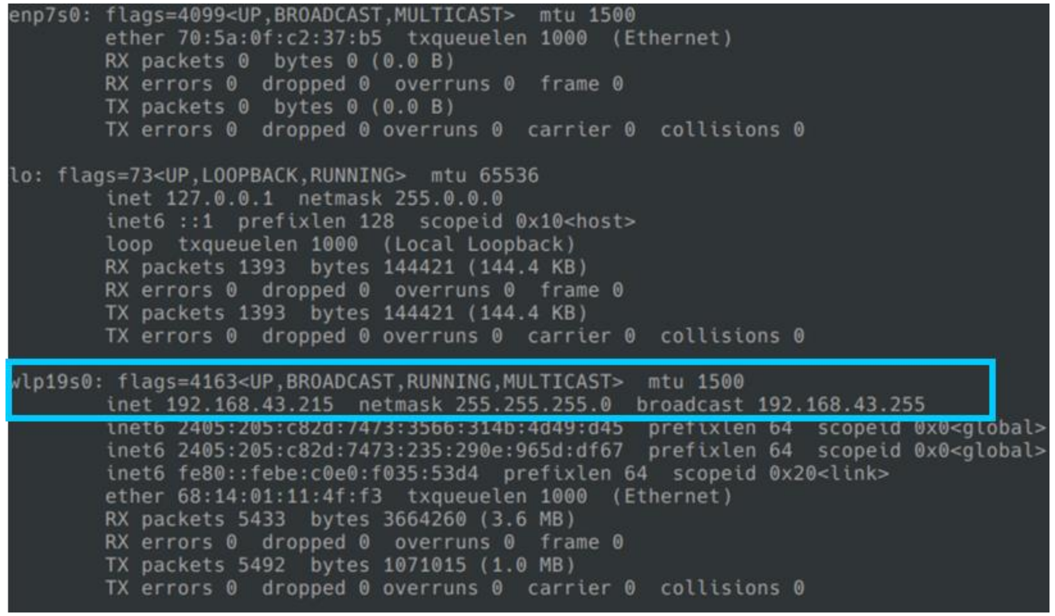
- **ROS IP – local IP with reference to Jetson Network**
- **ROS Master URI – Points to master on the Jetson**

- **ROS IP : 10.42.0.1 – local IP with reference to Jetson Network**
- **ROS Master URI: https://10.42.0.1:11311 – Points to local master**

Multi-device Communication with ROS (ROS Network)

1. Connect to the AP on the Jetson Nano
2. Set the variables on the external computer as follows
   - *export ROS_MASTER_URI=http://10.42.0.1:11311* (Jetson Nano IP address)
   - *export ROS_IP= <your_local_ip>*
   - To command *ifconfig* can be used to verify your local IP

- This connection is **not** a remote access 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.



```
enp7s0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 70:5a:0f:c2:37:b5  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 1393  bytes 144421 (144.4 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1393  bytes 144421 (144.4 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlp19s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.43.215  netmask 255.255.255.0  broadcast 192.168.43.255
        inet6 2405:205:c82d:7473:3566:314b:4d49:d45  prefixlen 64  scopeid 0x0<global>
        inet6 2405:205:c82d:7473:235:290e:965d:df67  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::febe:c0e0:f035:53d4  prefixlen 64  scopeid 0x20<link>
        ether 68:14:01:11:4f:f3  txqueuelen 1000  (Ethernet)
        RX packets 5433  bytes 3664260 (3.6 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 5492  bytes 1071015 (1.0 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

- More information and other connections (multiple robots) can be found here.

Troubleshooting

- The Hacker Board communication will only work correctly if the Jetson is connected to its own Wi-Fi network (AP)
  - Go to Network -> Connect to Hidden Wi-Fi network and select “Hotspot” from the dropdown.
  - The PuzzlebotJetson Network should now be visible in the networks; connect to it, to activate the hotspot.
  - If no hotspot is created, create one by setting a fixed IP address (default ip address= 10.42.0.1) and a password.
  - If the network is configured incorrectly, you will likely see the message “unable to communicate with master”
- If the ROS master is available, but the topics `/wr` and `/wl` are not available, check the connection to the Hackerboard, and restart it. Then, restart the communication with

```
sudo systemctl restart puzzlebot.service
```

- Sometimes, the Jetson Wi-Fi Network takes a couple of minutes to appear on other devices after a reboot. If it does not, make sure no other connections are saved. It is only guaranteed to connect to PuzzlebotJetson on boot if there are no other connections saved
- Always consider checking the hackerboard if these don’t work: is ROS turned on?

Troubleshooting

- When compiling other scripts or programs using catkin, the following error appear "Vision Messages not found; install them by connecting the Puzzlebot to the internet and type the following into the terminal

```
sudo systemctl stop puzzlebot.service
sudo apt install ros-<ROSDistro>-vision-msgs
```

- Do not forget to reactivate the hotspot once this finished
- Do not forget to restart the Puzzlebot service by typing

```
sudo systemctl restart puzzlebot.service
```

Thank you

{Learn, Create, Innovate};

MCR2 Manchester Robotics

T&C

Terms and conditions

{Learn, Create, Innovate};

Terms and conditions