Micro-ROS

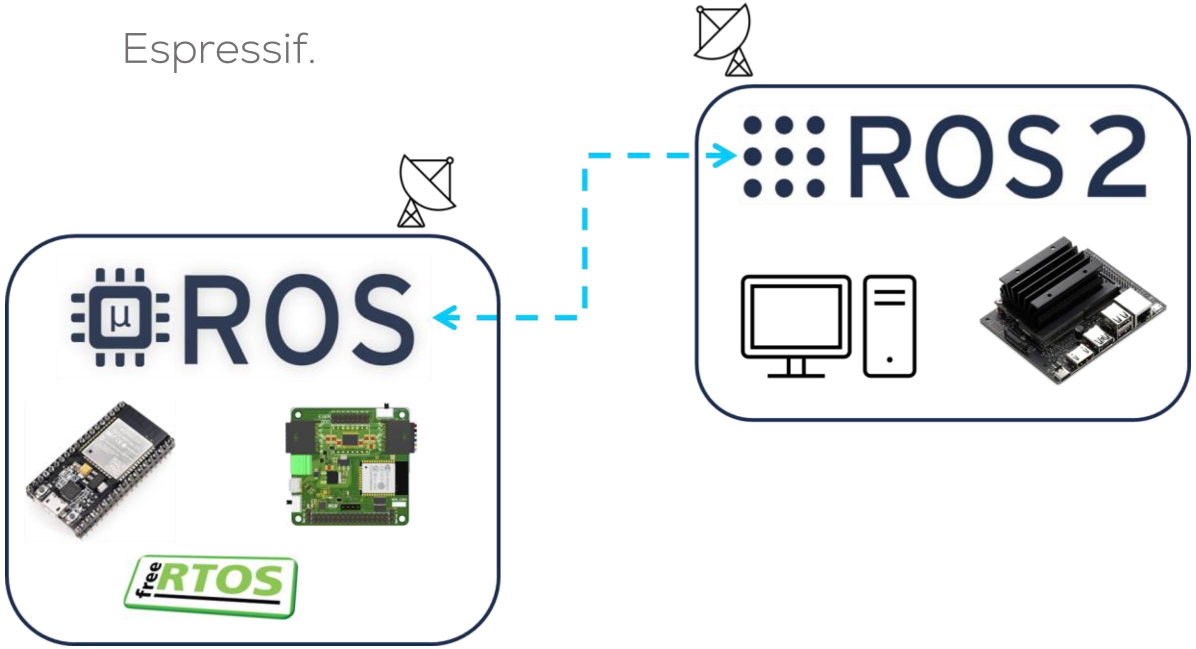
Installation

{Learn, Create, Innovate};

micro-ros

- Micro-ros is a tool that bridges the gap between resource-constrained microcontrollers and larger processors in robotic applications that are based on the Robot Operating System.
- The installation of micro-ros is divided in two parts.
  1. Installing the “agent” on an Ubuntu-based system with ROS 2, which is the firmware in charge of communicating with the microcontroller from the computer.
  2. Installing the microcontroller libraries required to compile the microcontroller code.

- This tutorial will show how to install both libraires for Ubuntu 22.04 with ROS2 Humble.
- micro-ros_arduino libraries for the Arduino IDE.
- The microcontroller to be used is the ESP32 by Espressif.



Ubuntu

micro-ros agent installation

{Learn, Create, Innovate};

Manchester Robotics

micro-ros agent installation

Create a ROS 2 workspace and build this package for ROS2 Humble

```
$ source /opt/ros/humble/setup.bash
$ mkdir uros_ws && cd uros_ws
$ git clone -b humble https://github.com/micro-ROS/micro_ros_setup.git src/micro_ros_setup

$ sudo apt update -y
$ sudo apt upgrade -y
$ sudo apt full-upgrade -y
$ sudo apt autoremove -y
$ sudo apt autoclean -y
$ sudo apt purge
```

Install and update dependencies

```
$ sudo rosdep init
$ rosdep update && rosdep install --from-paths src --ignore-src -y
```

Build package

```
$ sudo apt install python3-pip -y
$ colcon build
$ source install/local_setup.sh

$ ros2 run micro_ros_setup create_agent_ws.sh
$ ros2 run micro_ros_setup build_agent.sh
$ source install/local_setup.sh

$ echo "source ~/uros_ws/install/local_setup.bash" >> ~/.bashrc
```

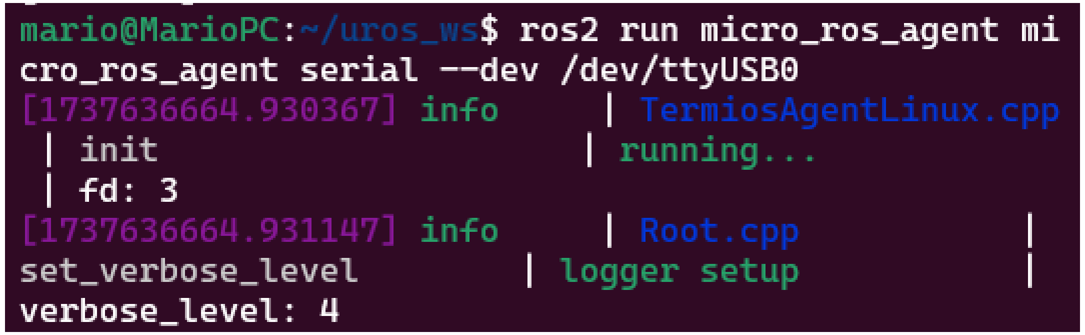
Port Permissions

```
$ dmesg | grep tty
$ sudo chmod a+rw /dev/tty*
$ sudo usermod -a -G dialout $USER
```

micro-ros agent installation

- To test the installation, type the following.

```
$ ros2 run micro_ros_agent micro_ros_agent serial --dev /dev/ttyUSB
```



```
mario@MarioPC:~/uros_ws$ ros2 run micro_ros_agent mi
cro_ros_agent serial --dev /dev/ttyUSB0
[1737636664.930367] info     | TermiosAgentLinux.cpp
 | init                    | running...
 | fd: 3
[1737636664.931147] info     | Root.cpp              |
set_verbose_level        | logger setup            |
verbose_level: 4
```

- For now, nothing is published since no microcontroller is connected to the computer.
- In the next section, the instructions for installing micro-ros libraries for Arduino will be shown.

Ubuntu

Arduino IDE Configuration

{Learn, Create, Innovate};

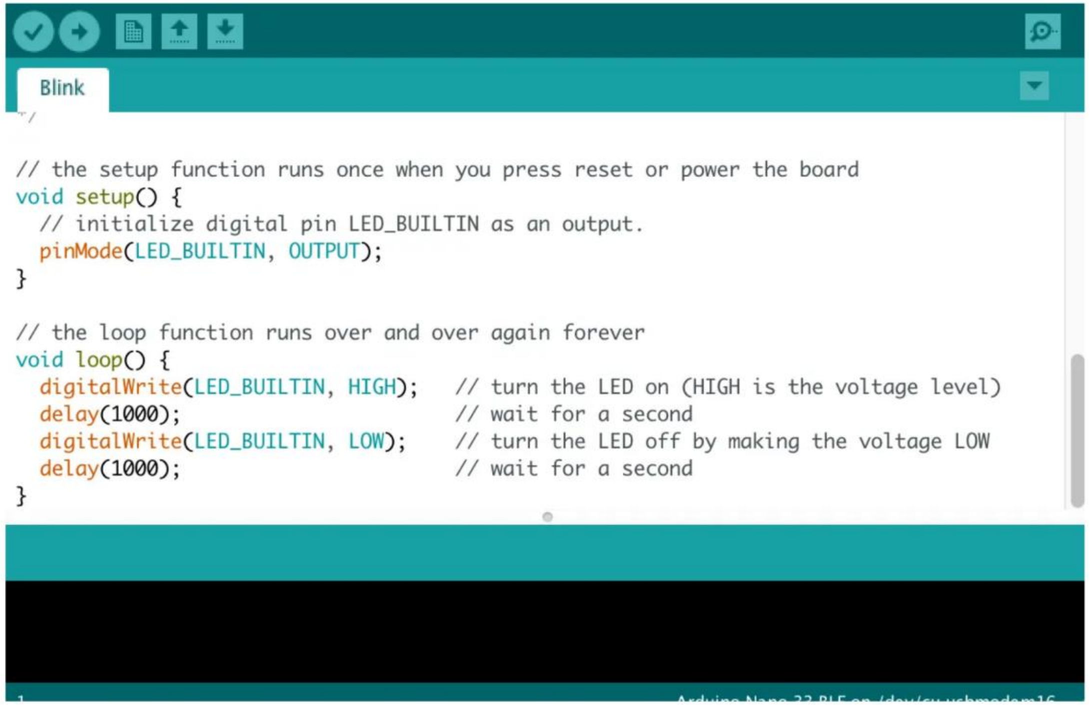
MCR2

Manchester **Robotics**

Arduino IDE

Configuring the Arduino IDE

- The Arduino and Arduino IDE are great tools for quickly and easily programming hardware.
- The micro_ros_arduino package, allows the usage of ROS directly with the Arduino IDE.
- micro_ros_arduino provides an ROS communication protocol that works with your Arduino's UART.
- micro_ros_arduino, allows Arduino to be a ROS node which can directly publish and subscribe to ROS messages, publish TF transforms, and get the ROS system time.
- Arduino IDE can also be used to program the Hackerboard and other microcontrollers like the ESP32.



```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
}
```

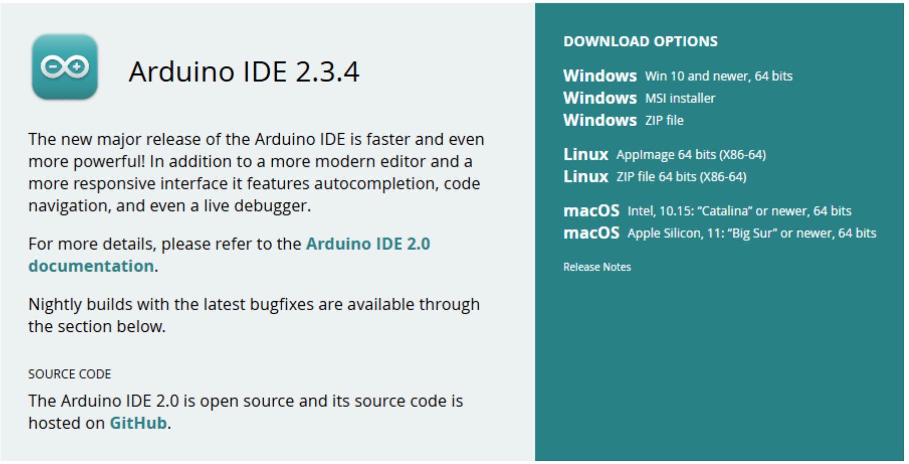
Arduino IDE

Arduino IDE

Installation

- Download the Arduino IDE from the website.
- Install the Arduino IDE application into a folder on the desktop (Windows), or home folder (Ubuntu).
- Follow the installation instructions for Windows and Linux here.
- Once installed, launch the application if you want to select your sketchbook location (File››Preferences››Sketchbook location).
    - Sketchbook is a standard place to store your programs, or sketches.
- Close the IDE when done.

Note: The Arduino IDE can be installed in the Virtual Machine, following the same steps.



Arduino IDE 2.3.4

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

For more details, please refer to the **Arduino IDE 2.0 documentation**.

Nightly builds with the latest bugfixes are available through the section below.

SOURCE CODE

The Arduino IDE 2.0 is open source and its source code is hosted on **GitHub**.

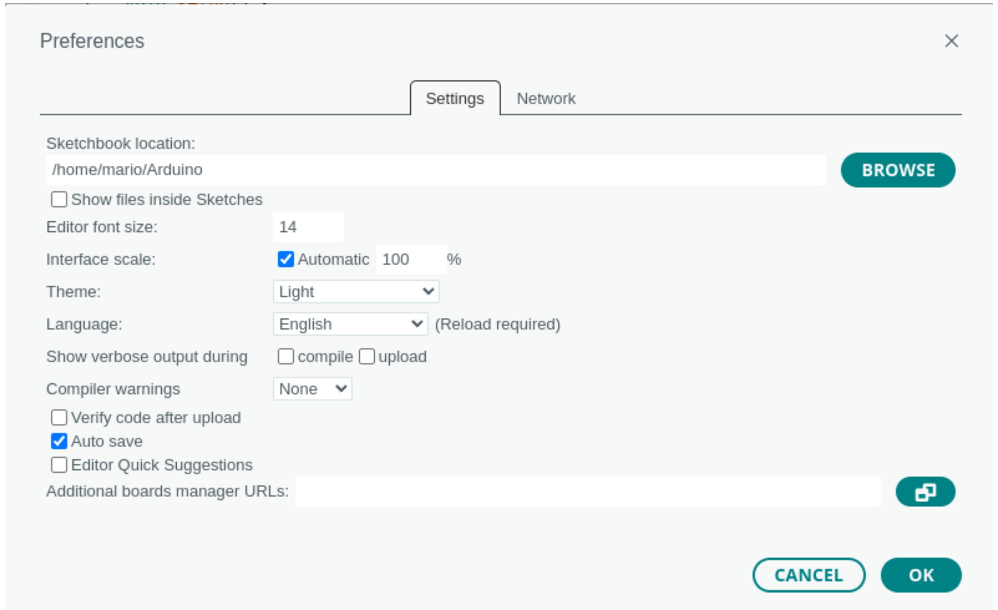
DOWNLOAD OPTIONS

Windows Win 10 and newer, 64 bits
Windows MSI installer
Windows ZIP file

Linux AppImage 64 bits (X86-64)
Linux ZIP file 64 bits (X86-64)

macOS Intel, 10.15: “Catalina” or newer, 64 bits
macOS Apple Silicon, 11: “Big Sur” or newer, 64 bits

Release Notes



Preferences

Settings | Network

Sketchbook location:
/home/mario/Arduino
BROWSE

□ Show files inside Sketches

Editor font size: 14

Interface scale: ☑ Automatic 100 %

Theme: Light

Language: English (Reload required)

Show verbose output during □ compile □ upload

Compiler warnings None

□ Verify code after upload

☑ Auto save

□ Editor Quick Suggestions

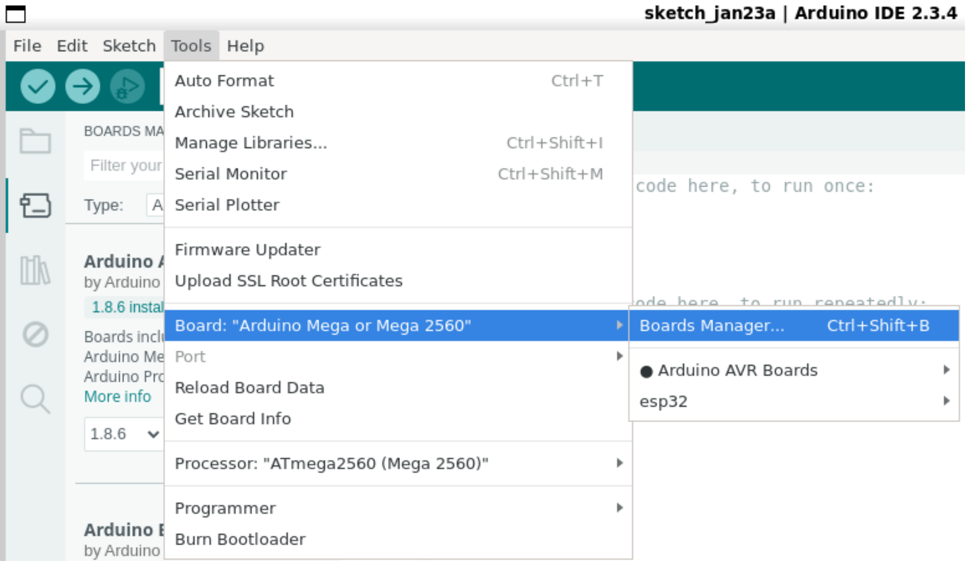
Additional boards manager URLs:

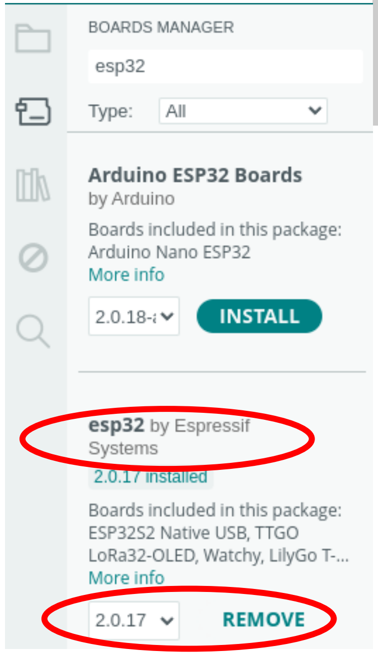
CANCEL OK

Arduino IDE-ESP32 Setup

Arduino IDE – ESP32 and Hackerboard setup

- To install the ESP32 libraries simply
- Open the Arduin IDE
- Go to Tools››Board››Boards Manager
- In the search bar of the left panel type “esp32”
- Select the “esp32 by Espressif”
- Select the **version 2.0.17 NOT THE NEWEST!! IS NOT FULLY COMPATIBLE WITH MICRO-ROS!**
- DO NOT UPGRADE THE LIBRARY!
- Press “Install”
- Test the installation using the example (at the end) [here]() or [here]().





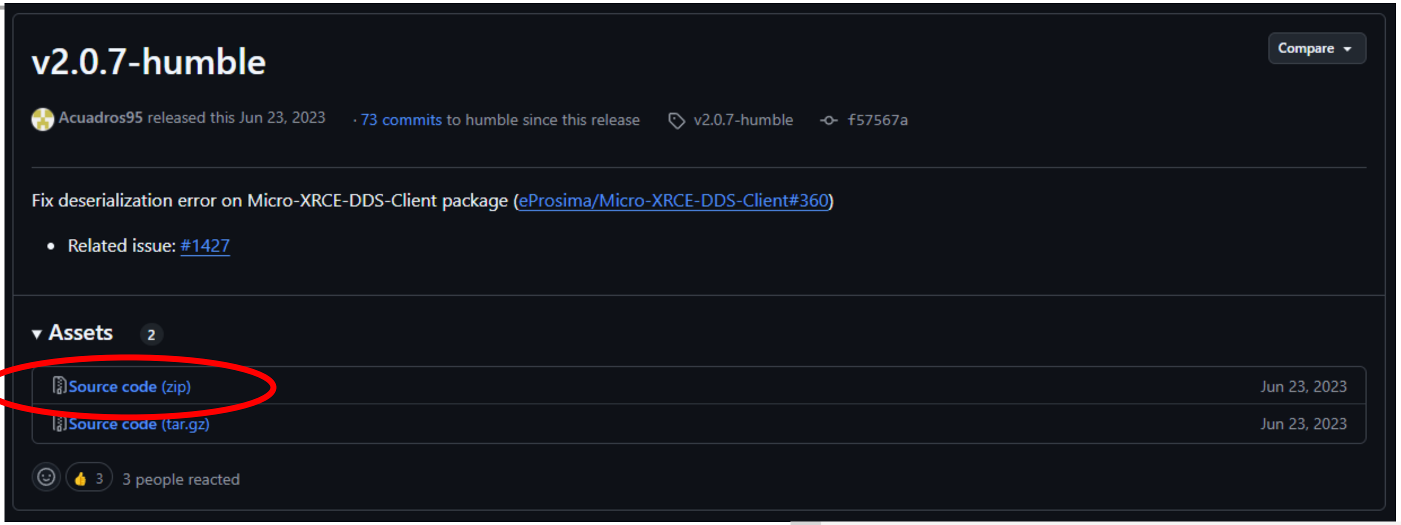
Arduino ESP32 Boards by Arduino
Boards included in this package: Arduino Nano ESP32
More info
2.0.18-a INSTALL

esp32 by Espressif Systems
2.0.17 installed
Boards included in this package: ESP32S2 Native USB, TTGO LoRa32-OLED, Watchy, LilyGo T-...
More info
2.0.17 REMOVE

micro-ros-arduino Setup

micro-ros setup (windows/ubuntu)

1. Download the “micro_ros_arduino” package here
2. Click on **the v2.07-humble version** (stable)
3. Download “Source code.zip”
4. Open Arduino IDE
5. Go to Sketch>>Include Library>>Add .ZIP Library
6. Select the “micro_ros_arduino-2.0.7-humble.zip” library you downloaded in the first step.
7. Restart the Arduino IDE
8. If properly installed, you should be able to see the examples File>>Examples>micro_ros_arduino



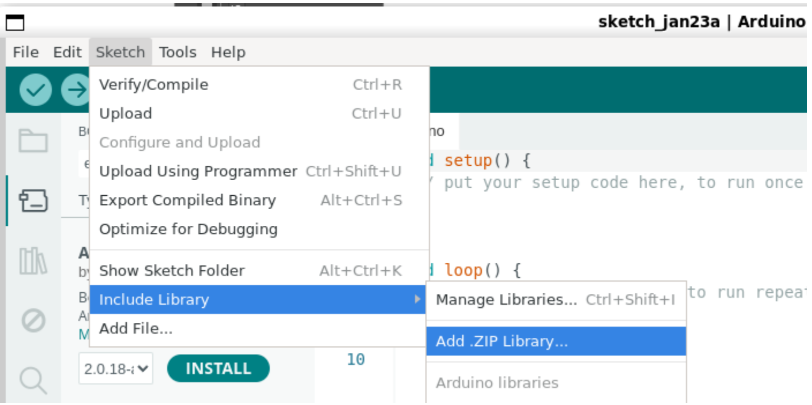
v2.0.7-humble

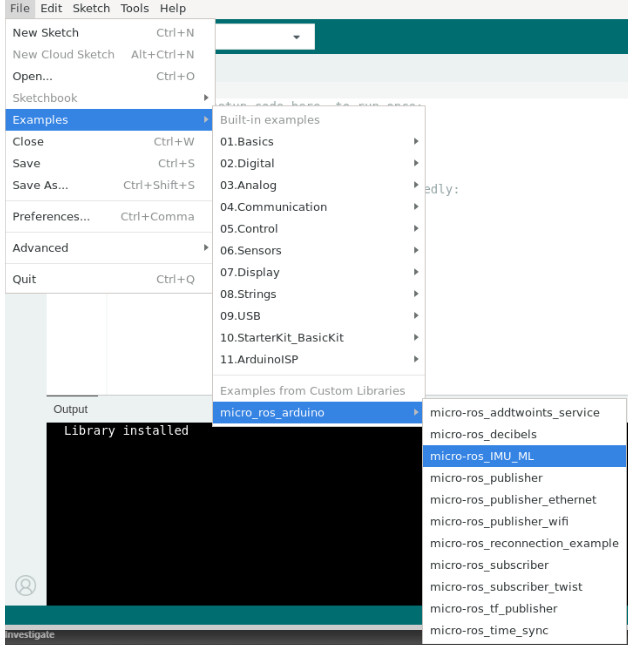
Acuadros95 released this Jun 23, 2023 · 73 commits to humble since this release v2.0.7-humble f57567a

Fix deserialization error on Micro-XRCE-DDS-Client package (eProsima/Micro-XRCE-DDS-Client#360)

- Related issue: #1427

Compare
Assets 2
Source code (zip) Jun 23, 2023
Source code (tar.gz) Jun 23, 2023
3 people reacted

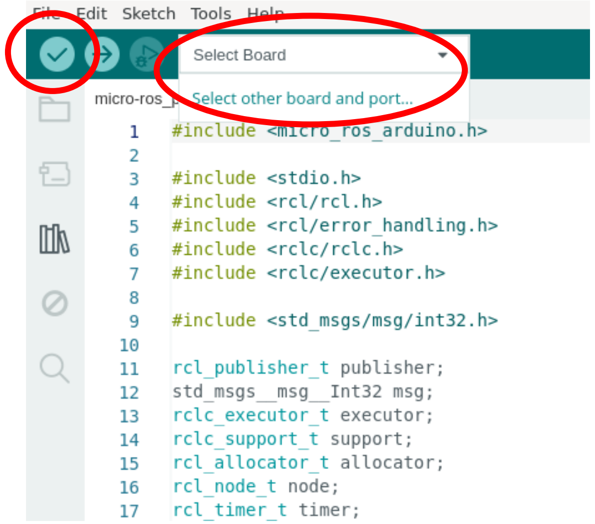




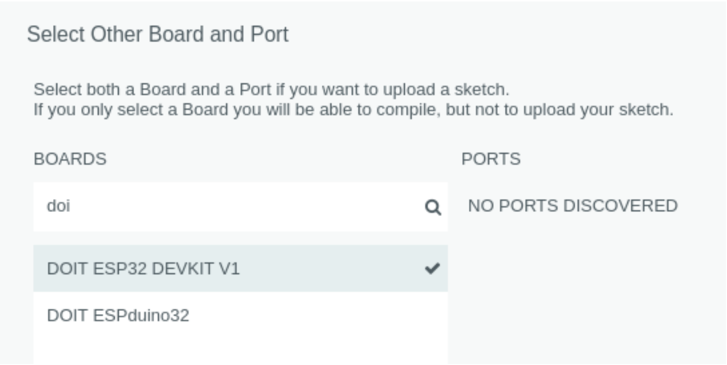
micro-ros-arduino Setup

micro-ros verification

1. Open the example micro-ros_publisher
2. Compile the example by selecting on the "Select Board" bar "DOIT ESP32 DEVKIT V1"
3. Click in the check mark at in the top left bar to compile.
4. The program should compile correctly.
5. If not, go to the Troubleshoot section of this presentation.

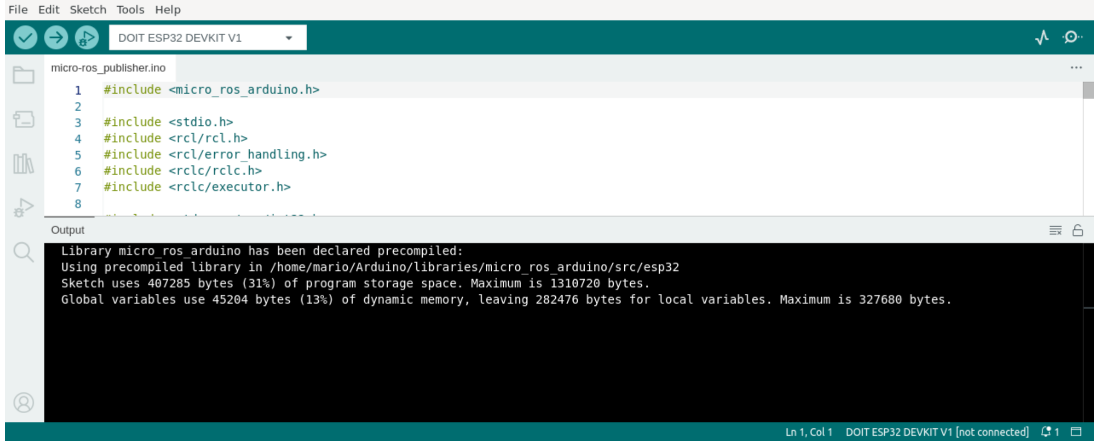


```
#include <micro_ros_arduino.h>

#include <stdio.h>
#include <rcl/rcl.h>
#include <rcl/error_handling.h>
#include <rclc/rclc.h>
#include <rclc/executor.h>

#include <std_msgs/msg/int32.h>

rcl_publisher_t publisher;
std_msgs__msg__Int32 msg;
rclc_executor_t executor;
rclc_support_t support;
rcl_allocator_t allocator;
rcl_node_t node;
rcl_timer_t timer;
```



Select Other Board and Port

Select both a Board and a Port if you want to upload a sketch.
If you only select a Board you will be able to compile, but not to upload your sketch.


File Edit Sketch Tools Help
DOIT ESP32 DEVKIT V1
micro-ros_publisher.ino
Output
Ln 1, Col 1 DOIT ESP32 DEVKIT V1 [not connected] 1

```
#include <micro_ros_arduino.h>

#include <stdio.h>
#include <rcl/rcl.h>
#include <rcl/error_handling.h>
#include <rclc/rclc.h>
#include <rclc/executor.h>
```

```
Library micro_ros_arduino has been declared precompiled:
Using precompiled library in /home/mario/Arduino/libraries/micro_ros_arduino/src/esp32
Sketch uses 407285 bytes (31%) of program storage space. Maximum is 1310720 bytes.
Global variables use 45204 bytes (13%) of dynamic memory, leaving 282476 bytes for local variables. Maximum is 327680 bytes.
```

Port Permissions (Ubuntu)

- To use Arduino or the ESP32 in Ubuntu, the user must give permissions to the system for accessing ports.
- Make sure the port permissions are granted for the user.
  - In a new terminal type `cd ~/dev` to visualise the port designated by Ubuntu to the MCU. This port are usually called `/ttyACM0` or `/ttyUSB0`.
  - Having obtained the name of the port type the correspondent command to enable the permissions (replace the asterisk with the port number).

```
sudo chmod 666 /dev/ttyACM*
sudo chmod 666 /dev/ttyUSB*
```

- To make the change permanently, follow the steps here.

WSL (Windows Subsystem for Linux)

- To install the Arduino IDE on WSL follow the next steps.

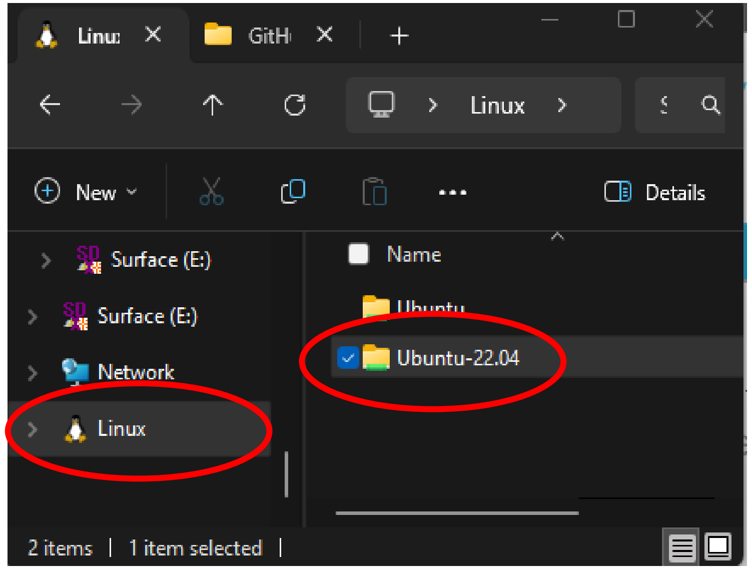
1. Install the Nautilus (File manager)

```
$ sudo apt install nautilus -y (Ubuntu Window)
$ sudo apt-get update && sudo apt-get upgrade
$ nautilus
(After nautilus window pop-up, close it) (Ubuntu Window)
```

2. Download Arduino IDE Linux AppImage under Windows

3. Copy the file using the Windows File Explorer (open a folder) then select the file and copy it or move it to Linux

```
Ubuntu-22.04/home/$USER$/
```



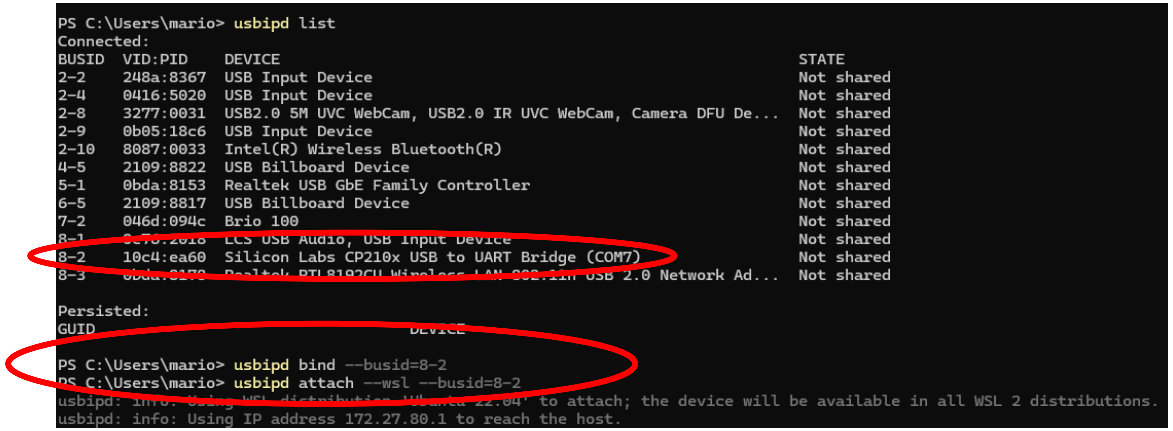
- Give executable permissions and update the libraries to the file in Ubuntu

```
$ chmod u+x arduino-ide_2.0.3_Linux_64bit.AppImage
$ sudo apt-get update && sudo apt-get upgrade
$ sudo apt-get install -y libgbm-dev
$ nautilus
$ Double Click AppImage File & Wait for Arduino to
  Download Packages
```

WSL (Windows Subsystem for Linux)

- To access the USB Port from WSL Install the “usbipd-win” package in windows.
- More information here and here.

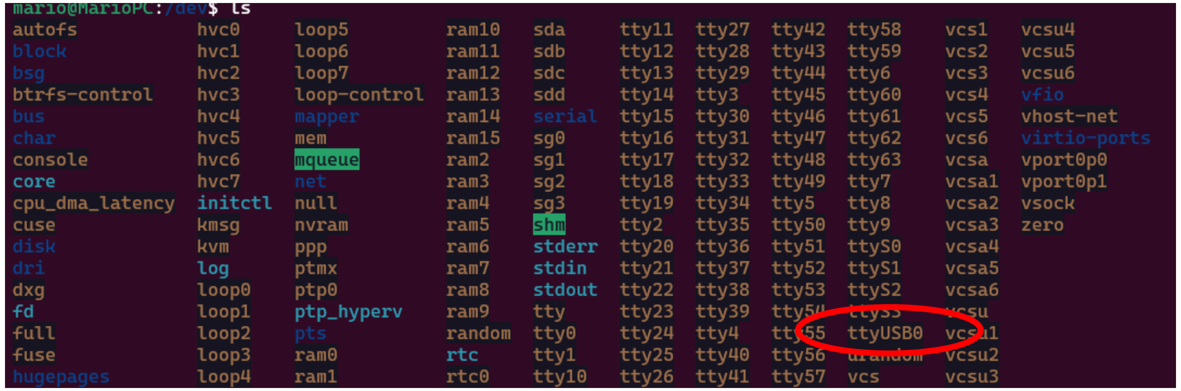
1.
```
winget install usbipd
```
2. Attach Arduino/ESP32 Board with PC through USB
3. Make Sure Com Port Appears in Device Manager (Windows)
4. Install COM port Drivers If required (Windows)
5. Restart Powershell and run as admin
```
usbipd list
usbipd bind --busid=BUSID
usbipd attach --wsl --busid= BUSID
```
6. To finish the session
```
usbipd detach --busid= BUSID
```



```
PS C:\Users\mario> usbipd list
Connected:
BUSID  VID:PID    DEVICE                                                        STATE
2-2    248a:8367  USB Input Device                                              Not shared
2-4    0416:5020  USB Input Device                                              Not shared
2-8    3277:0031  USB2.0 5M UVC WebCam, USB2.0 IR UVC WebCam, Camera DFU De...  Not shared
2-9    0b05:18c6  USB Input Device                                              Not shared
2-10   8087:0033  Intel(R) Wireless Bluetooth(R)                                Not shared
4-5    2109:8822  USB Billboard Device                                          Not shared
5-1    0bda:8153  Realtek USB GbE Family Controller                             Not shared
6-5    2109:8817  USB Billboard Device                                          Not shared
7-2    046d:094c  Brio 100                                                      Not shared
8-1    0c76:2018  LCS USB Audio, USB Input Device                               Not shared
8-2    10c4:ea60  Silicon Labs CP210x USB to UART Bridge (COM7)                 Not shared
8-3    0bda:8178  Realtek RTL8192CU Wireless LAN 802.11n USB 2.0 Network Ad...  Not shared

Persisted:
GUID                                  DEVICE

PS C:\Users\mario> usbipd bind --busid=8-2
PS C:\Users\mario> usbipd attach --wsl --busid=8-2
usbipd: info: Using WSL distribution 'Ubuntu 22.04' to attach; the device will be available in all WSL 2 distributions.
usbipd: info: Using IP address 172.27.80.1 to reach the host.
```

- In Linux, check that the port is displayed and provide port permissions as shown previously. Now, you can work as normal in WLS. Do not forget to detach at the end of the session.

```
$ cd /dev
$ ls
```



```
mario@MarioPC:/dev$ ls
autofs           hvc0      loop5         ram10   sda     tty11  tty27  tty42  tty58    vcs1   vcsu4
block            hvc1      loop6         ram11   sdb     tty12  tty28  tty43  tty59    vcs2   vcsu5
bsg              hvc2      loop7         ram12   sdc     tty13  tty29  tty44  tty6     vcs3   vcsu6
btrfs-control    hvc3      loop-control  ram13   sdd     tty14  tty3   tty45  tty60    vcs4   vfio
bus              hvc4      mapper        ram14   serial  tty15  tty30  tty46  tty61    vcs5   vhost-net
char             hvc5      mem           ram15   sg0     tty16  tty31  tty47  tty62    vcs6   virtio-ports
console          hvc6      mqueue        ram2    sg1     tty17  tty32  tty48  tty63    vcsa   vport0p0
core             hvc7      net           ram3    sg2     tty18  tty33  tty49  tty7     vcsa1  vport0p1
cpu_dma_latency  initctl   null          ram4    sg3     tty19  tty34  tty5   tty8     vcsa2  vsock
cuse             kmsg      nvram         ram5    shm     tty2   tty35  tty50  tty9     vcsa3  zero
disk             kvm       ppp           ram6    stderr  tty20  tty36  tty51  ttyS0    vcsa4
dri              log       ptmx          ram7    stdin   tty21  tty37  tty52  ttyS1    vcsa5
dxg              loop0     ptp0          ram8    stdout  tty22  tty38  tty53  ttyS2    vcsa6
fd               loop1     ptp_hyperv    ram9    tty     tty23  tty39  tty54  ttyS3    vcsu
full             loop2     pts           random  tty0    tty24  tty4   tty55  ttyUSB0  vcsu1
fuse             loop3     ram0          rtc     tty1    tty25  tty40  tty56  urandom  vcsu2
hugepages        loop4     ram1          rtc0    tty10   tty26  tty41  tty57  vcs      vcsu3
```

Testing

Testing installation

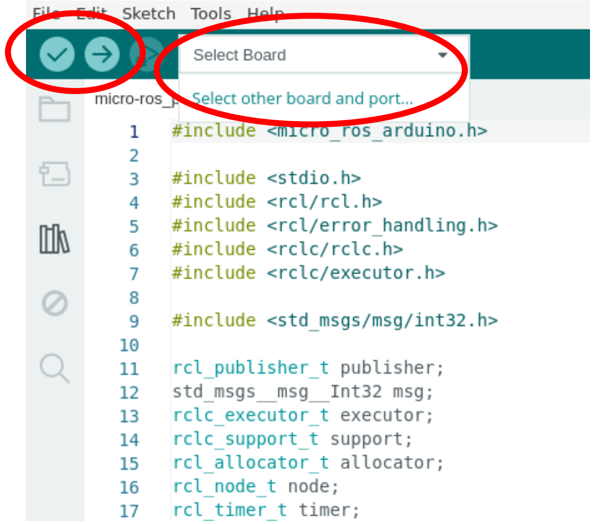
{Learn, Create, Innovate};

MCR2
Manchester Robotics

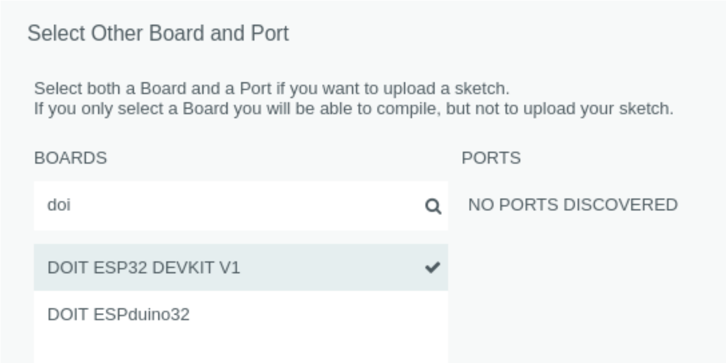
Testing installation

Test

1. Open the example micro-ros_publisher on Arduino IDE
2. Compile and upload the example by selecting on the “Select Board” bar “DOIT ESP32 DEVKIT V1”
3. Click on the arrow in the top left bar to compile.
    1. (WSL) If using WSL check how to access serial ports in the previous section.
    2. (Ubuntu) Verify that the ports have permissions in Ubuntu.
    3. (Windows) Verify the correct ports are selected in Windows and that the drivers are installed (troubleshoot section)
4. The program should compile and upload correctly.



```
#include <micro_ros_arduino.h>

#include <stdio.h>
#include <rcl/rcl.h>
#include <rcl/error_handling.h>
#include <rclc/rclc.h>
#include <rclc/executor.h>

#include <std_msgs/msg/int32.h>

rcl_publisher_t publisher;
std_msgs__msg__Int32 msg;
rclc_executor_t executor;
rclc_support_t support;
rcl_allocator_t allocator;
rcl_node_t node;
rcl_timer_t timer;
```

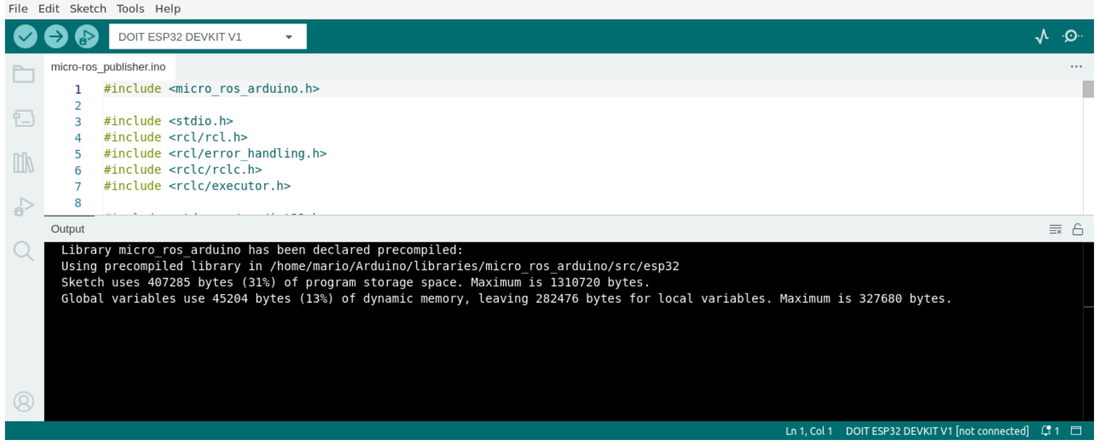


Select Other Board and Port

Select both a Board and a Port if you want to upload a sketch.
If you only select a Board you will be able to compile, but not to upload your sketch.

BOARDS: doi
- DOIT ESP32 DEVKIT V1 ✓
- DOIT ESPduino32

PORTS: NO PORTS DISCOVERED



```
#include <micro_ros_arduino.h>

#include <stdio.h>
#include <rcl/rcl.h>
#include <rcl/error_handling.h>
#include <rclc/rclc.h>
#include <rclc/executor.h>
```

```
Library micro_ros_arduino has been declared precompiled:
Using precompiled library in /home/mario/Arduino/libraries/micro_ros_arduino/src/esp32
Sketch uses 407285 bytes (31%) of program storage space. Maximum is 1310720 bytes.
Global variables use 45204 bytes (13%) of dynamic memory, leaving 282476 bytes for local variables. Maximum is 327680 bytes.
```

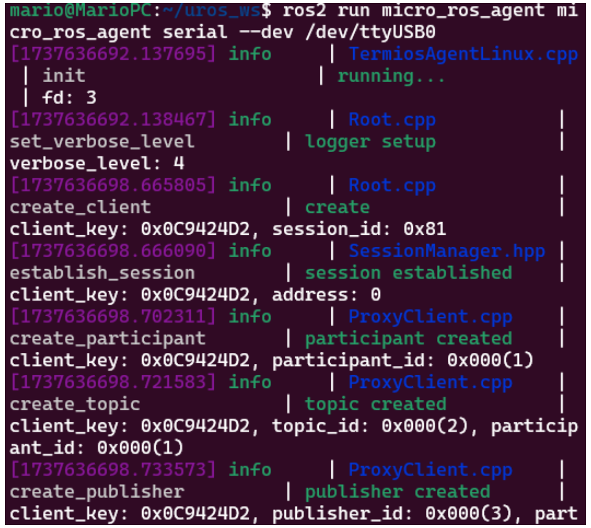
Testing installation

Test

1. Open another terminal and type the following.

```
$ ros2 run micro_ros_agent micro_ros_agent serial --dev /dev/ttyUSB0
```

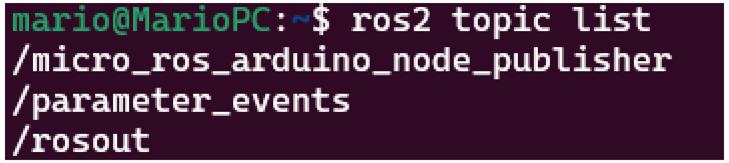
2. Reset the ESP32 (pressing the reset button) to reconnect to the computer.



```
mario@MarioPC:~/uros_ws$ ros2 run micro_ros_agent mi
cro_ros_agent serial --dev /dev/ttyUSB0
[1737636692.137695] info     | TermiosAgentLinux.cpp
 | init                     | running...
 | fd: 3
[1737636692.138467] info     | Root.cpp             |
set_verbose_level        | logger setup         |
verbose_level: 4
[1737636698.665805] info     | Root.cpp             |
create_client            | create               |
client_key: 0x0C9424D2, session_id: 0x81
[1737636698.666090] info     | SessionManager.hpp   |
establish_session        | session established  |
client_key: 0x0C9424D2, address: 0
[1737636698.702311] info     | ProxyClient.cpp      |
create_participant       | participant created  |
client_key: 0x0C9424D2, participant_id: 0x000(1)
[1737636698.721583] info     | ProxyClient.cpp      |
create_topic             | topic created        |
client_key: 0x0C9424D2, topic_id: 0x000(2), particip
ant_id: 0x000(1)
[1737636698.733573] info     | ProxyClient.cpp      |
create_publisher         | publisher created    |
client_key: 0x0C9424D2, publisher_id: 0x000(3), part
```

1. Open another terminal and type the following.

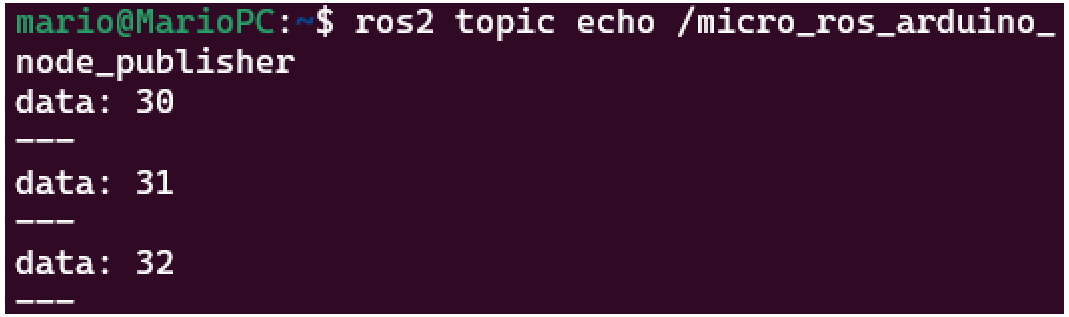
```
$ ros2 topic list
```



```
mario@MarioPC:~$ ros2 topic list
/micro_ros_arduino_node_publisher
/parameter_events
/rosout
```

2. Echo the topic "/micro_ros_arduino_node_publisher"

```
$ ros2 topic echo /micro_ros_arduino_node_publisher
```



```
mario@MarioPC:~$ ros2 topic echo /micro_ros_arduino_
node_publisher
data: 30
---
data: 31
---
data: 32
---
```

Troubleshoot

Common problems with Arduino IDE

{Learn, Create, Innovate};

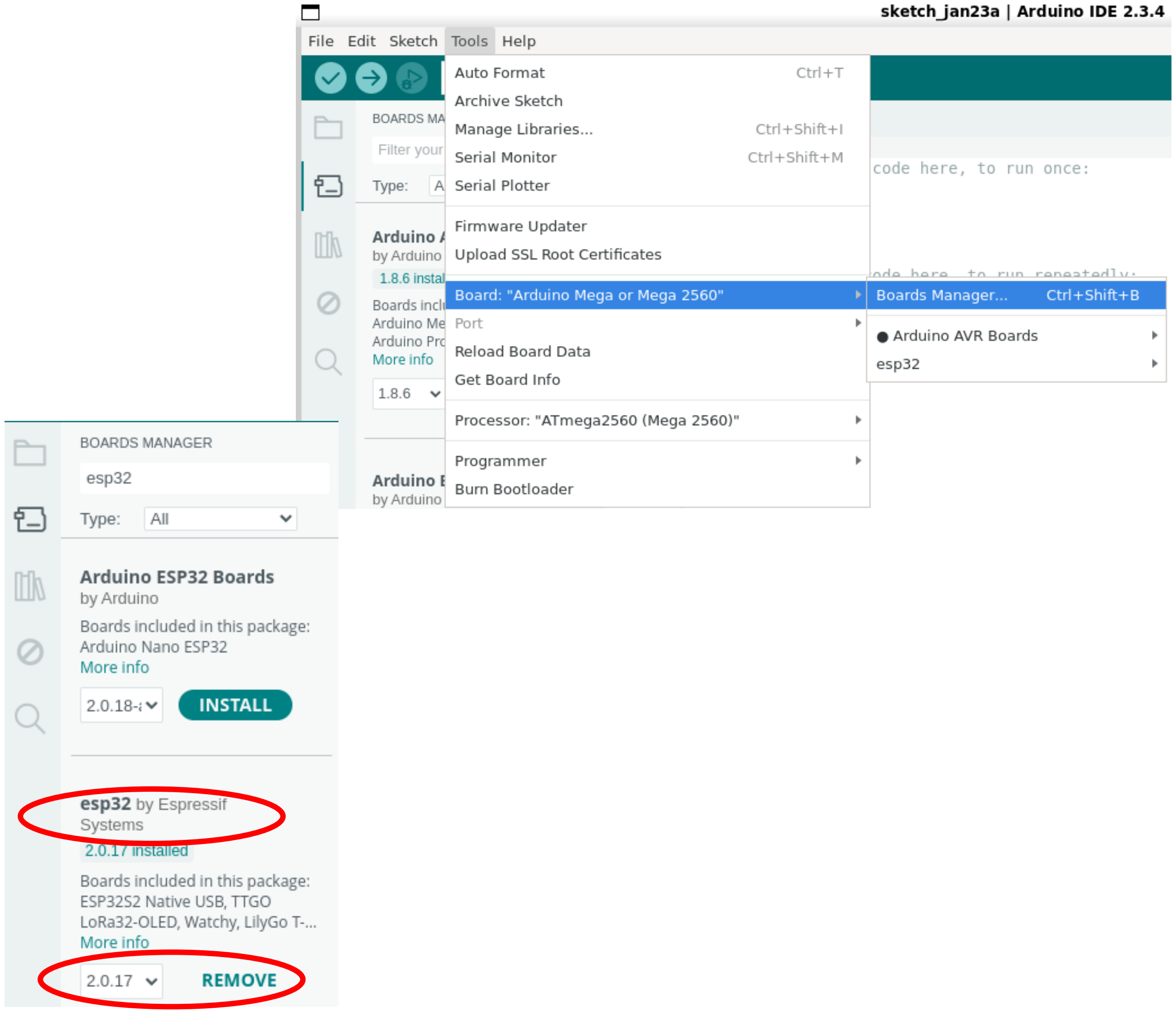
MCR²

Manchester Robotics

Arduino IDE-ESP32 Setup

Troubleshoot

- If the “esp32 by Espressif” is not available follow one of the next tutorials.
- Simple tutorial (not official)
    - Simple tutorial Arduino IDE 2.0 (not official)
    - Espressif Tutorial (official)
    - Official Github
- Connect the Hackerboard/ESP32 to the computer.
- Test the installation using the example (at the end) here or here.



sketch_jan23a | Arduino IDE 2.3.4
File
Edit
Sketch
Tools
Help
Auto Format Ctrl+T
Archive Sketch
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter
Firmware Updater
Upload SSL Root Certificates
Board: "Arduino Mega or Mega 2560"
Port
Reload Board Data
Get Board Info
Processor: "ATmega2560 (Mega 2560)"
Programmer
Burn Bootloader
Boards Manager... Ctrl+Shift+B
● Arduino AVR Boards
esp32
code here, to run once:
BOARDS MANAGER
esp32
Type: All
Arduino ESP32 Boards
by Arduino
Boards included in this package:
Arduino Nano ESP32
More info
2.0.18-
INSTALL
esp32 by Espressif Systems
2.0.17 installed
Boards included in this package:
ESP32S2 Native USB, TTGO LoRa32-OLED, Watchy, LilyGo T-...
More info
2.0.17
REMOVE

USB Ports in VM

- When connecting a USB to a VM several steps must be performed for the virtual machine to be able to recognise the USB Port from the host computer. More information can be found here.
- Make sure the correct drivers for the device are installed in the host computer. More information can be found here.
- Give permissions to the VM to access the USB ports of the host machine. More information can be found here and here.

A video tutorial on how to connect USB devices to the VMWare Player can be found here.

Troubleshoot

Troubleshooting

- When compiling for the ESP32 the following error appears

“Missing Python: “python”: executable file not found in $PATH”

To avoid this error, you can install the python-is-python3 package to create the symbolic links.

```
sudo apt install python-is-python3
```

- When compiling the following error appears

“ImportError: No module named serial”

To avoid this error, install the pyserial library

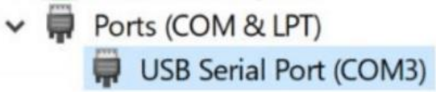
```
sudo apt install python3-pip
pip3 install pyserial
```

Additional Troubleshoot can be found here, here and here.

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).



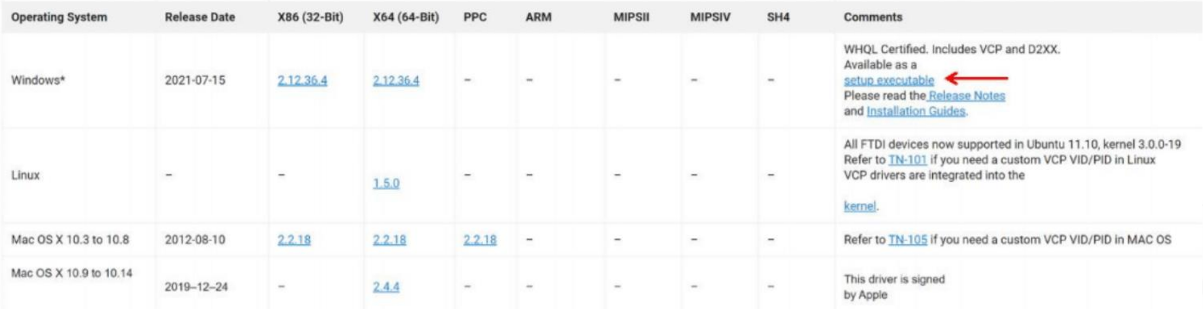
Ports (COM & LPT)
USB Serial Port (COM3)

- 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 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.

Troubleshoot

Troubleshoot (Drivers)

- Some Hackerboards 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!!

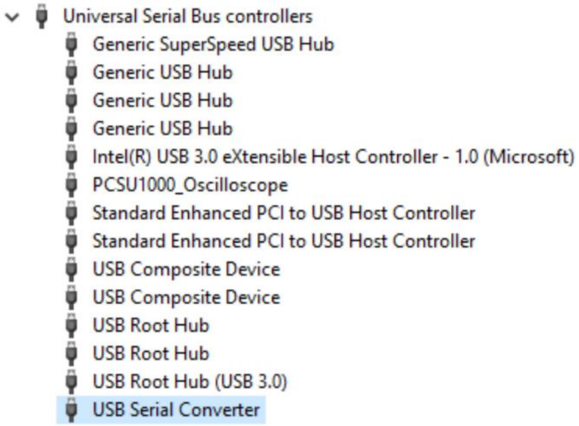
- Unplug the Puzzle-Bot from the computer.
- Unzip the drivers and run the setup (some computers are required to be restarted after the installation).
- Plug the Puzzle-Bot back into the computer.

A troubleshoot guide can be found here.

Troubleshoot

Troubleshoot (Drivers)

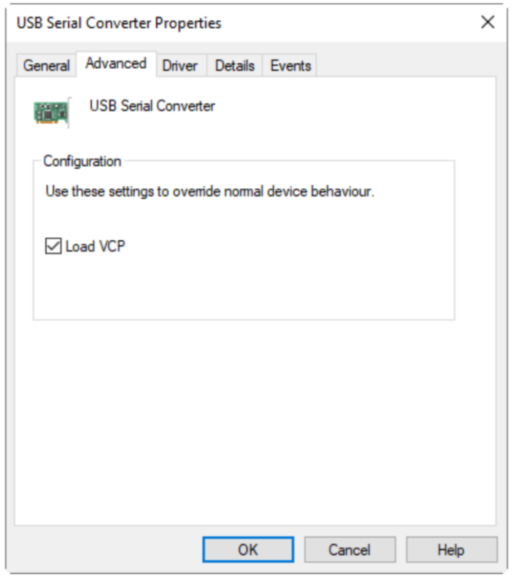
- My computer still not recognize the drivers even after the installation
- Plug the Puzzle-Bot into the USB port.
- Go to Start › Device Manager.
- Look for the USB Serial Converter as shown in the following picture.



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

Thank you

{Learn, Create, Innovate};

MCR2

Manchester **Robotics**

T&C

Terms and conditions

{Learn, Create, Innovate};

Manchester Robotics

Terms and conditions

- *THE PIECES, IMAGES, VIDEOS, DOCUMENTATION, ETC. SHOWN HERE ARE FOR INFORMATIVE PURPOSES ONLY. THE DESIGN IS PROPRIETARY AND CONFIDENTIAL TO MANCHESTER ROBOTICS LTD. (MCR2). THE INFORMATION, CODE, SIMULATORS, DRAWINGS, VIDEOS PRESENTATIONS ETC. CONTAINED IN THIS PRESENTATION IS THE SOLE PROPERTY OF MANCHESTER ROBOTICS LTD. ANY REPRODUCTION, RESELL, REDISTRIBUTION OR USAGE IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF MANCHESTER ROBOTICS LTD. IS STRICTLY PROHIBITED.*
- *THIS PRESENTATION MAY CONTAIN LINKS TO OTHER WEBSITES OR CONTENT BELONGING TO OR ORIGINATING FROM THIRD PARTIES OR LINKS TO WEBSITES AND FEATURES IN BANNERS OR OTHER ADVERTISING. SUCH EXTERNAL LINKS ARE NOT INVESTIGATED, MONITORED, OR CHECKED FOR ACCURACY, ADEQUACY, VALIDITY, RELIABILITY, AVAILABILITY OR COMPLETENESS BY US.*
- *WE DO NOT WARRANT, ENDORSE, GUARANTEE, OR ASSUME RESPONSIBILITY FOR THE ACCURACY OR RELIABILITY OF ANY INFORMATION OFFERED BY THIRD-PARTY WEBSITES LINKED THROUGH THE SITE OR ANY WEBSITE OR FEATURE LINKED IN ANY BANNER OR OTHER ADVERTISING.*