

NEARLAB – NES

TECHNICAL REPORTS



Title:	ESP32 Integration in ROS
Report Number:	200303
Author:	Stefano Dalla Gasperina
Author's Email:	stefano.dallagasperina@polimi.it
Date:	2020.03
Brief Description	<p>This guide guides through the integration of microcontrollers based on ESP32 in the ROS network via WiFi connection.</p> <p>The ESP32 is programmed in Arduino IDE in Ubuntu</p>
Revision:	1.0

I. INTRODUCTION

This guide provides an ease-of-use way to connect a wireless sensor to a ROS network, without implementing an ad-hoc communication protocol. In particular, this document walks through the procedure of installing a ROS node on a microcontroller based on **ESP32 WROOM** or **ESP32**, programmed with the **Arduino IDE** in **Ubuntu** environment.

Most of the report was written with the information found on the internet and on ROS documentation site. You can find the links to some of these online pages in the references section.

II. REQUIREMENTS

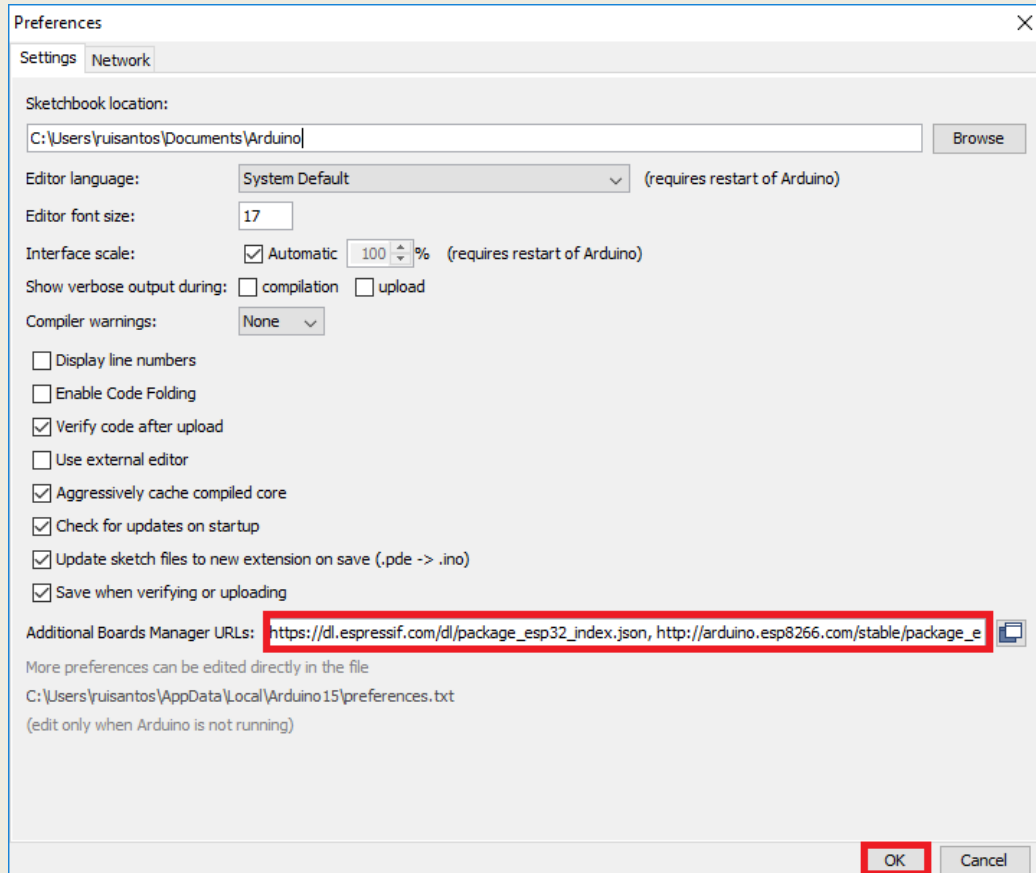
Before starting, please verify you have the following requirements:

1. 1x [ESP32 DevKitC](#) (ESP32 WROOM) or 1x [WiFi Kit 32](#) (ESP32) or NodeMCU (ESP8266-12E)
2. Wi-Fi Network
3. Install Linux [Ubuntu OS](#) (tested on Xenial 16.04 and Bionic 18.04)
4. Install [ROS](#) (tested on Melodic)
5. Install [Arduino IDE](#)

III. ESP32 ADD-ON INSTALLATION IN ARDUINO IDE

To install the ESP32 board in your Arduino IDE, follow these next instructions:

1. In your Arduino IDE, go to **File > Preferences**
2. Enter https://dl.espressif.com/dl/package_esp32_index.json into the “**Additional Board Manager URLs**” field as shown in the figure below. Then, click the “OK” button:



Note: if you already have the ESP8266 boards URL, you can separate the URLs with a comma as follows:

3. Open the **Boards Manager**. Go to **Tools > Board > Boards Manager...**
4. Search for **ESP32** and press **Install** button for the “ESP32 by Espressif Systems”:
5. That's it. It should be installed after a few seconds.

https://dl.espressif.com/dl/package_esp32_index.json,
http://arduino.esp8266.com/stable/package_esp8266com_index.json

IV. ALTERNATIVE ESP32 ADD-ON INSTALLATION (NOT RECOMMENDED)

If you have Arduino installed to ~/Arduino, modify the installation as follows:

```
cd ~/Arduino
mkdir -p hardware/espessif
cd hardware/espessif
git clone https://github.com/espessif/arduino-esp32.git esp32
cd esp32
git submodule update --init --recursive
cd tools $python3 get.py
```

V. ESP32 ADD-ON TEST

Open **Arduino IDE**

Tools > Board > Select your board

File > Examples > ESP32 Arduino > WiFi > WiFi Scan

Compile and Upload

Open CTRL+MAIUSC+M

If you are using Heltec WiFi Kit 32:

Sketch > Include Library > Manage Libraries

Install Heltec ESP32 Library

File > Examples > Heltec ESP32 Dev-Boards > FactoryTest (according to your board)

Compile and Upload

Open CTRL+MAIUSC+M

VI. ROSSERIAL INSTALLATION

You can install roserial for Arduino by running:

```
sudo apt-get install ros-kinetic-rosserial-arduino
sudo apt-get install ros-kinetic-rosserial
cd ~/Arduino/libraries
rm -rf ros_lib
roslaunch roserial_arduino make_libraries.py ~/Arduino/libraries
```

After restarting your IDE, you should see ros_lib listed under examples

VII. INSTALL REPOSITORY

Clone repository in Arduino IDE workspace

```
cd ~Arduino/sketchbook
git clone https://github.com/stex2005/ESP32-ROS-Driver.git
```

Replace all file from repository to ~/Arduino/libraries/ros_lib

```
cd ~/Arduino/sketchbook/ESP32-ROS-Driver
cp *.h *.cpp ~/Arduino/libraries/ros_lib
```

VIII. RUN THE CODE

1. Open ESP32HelloWorld with Arduino IDE
2. Open Terminal (CTRL+T) and type

```
ifconfig
```

Take note of the IP address of your computer and modify the sketch with the proper IP address:

```
// Set the roserial socket server IP address (ifconfig on roscore terminal)
IPAddress server(192,168,1,3);
```

Replace 192.168.1.3 with your IP address (N.B. comma separated).

3. Add "WiFicredentials.h" with your WiFi information

```
cd ~/Arduino/sketchbook/ESP32-ROS-Driver/ESP32HelloWorld
gedit WiFicredentials.h
```

Modify the file as follows and save

```
#define UID "YourSSID"
#define PASS "YourPassword"
```

4. Compile and Upload

5. Open some new Terminals and start **roscore** and **roserial server socket**:

Option 1:

```
roslaunch roserial_server socket.launch
```

Option 2:

```
roscore  
roslaunch roserial_server socket.launch
```

6. Check if the ESP32 is publishing at the correct frequency (example is 20Hz)

```
rostopic list  
rostopic echo chatter  
rostopic hz chatter
```

