

Arduino IDE on Ubuntu LTS 22.04



9/16/2023

Sangwon Lee

Contents

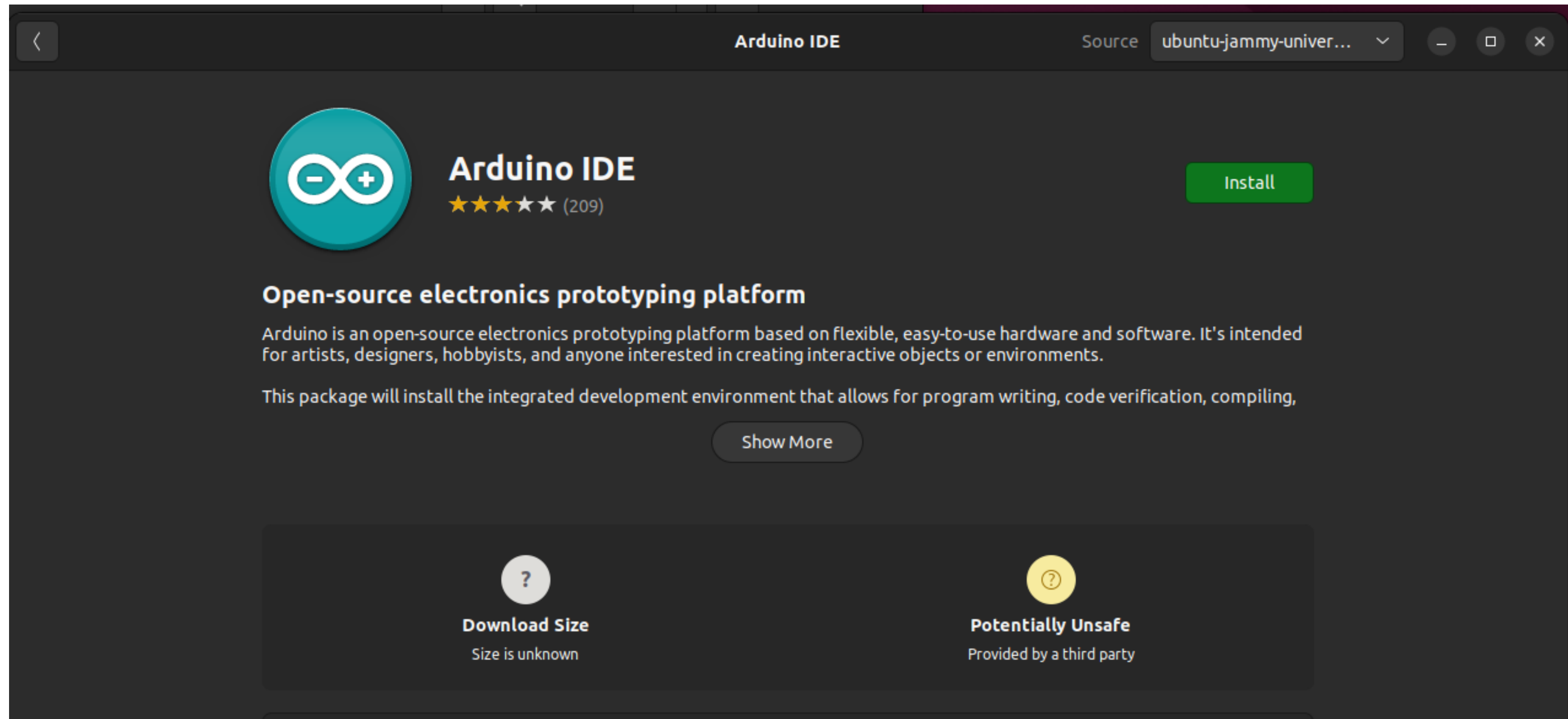
- ❑ Installation of Arduino IDE with script typing at Console
- ❑ Arduino IDE installation by snap-store
- ❑ Board Manager and Include Library
- ❑ Publisher Example “Hello ros2arduino 01”

Installation of Arduino IDE

- ❑ Installation of Arduino IDE with script typing at Terminal

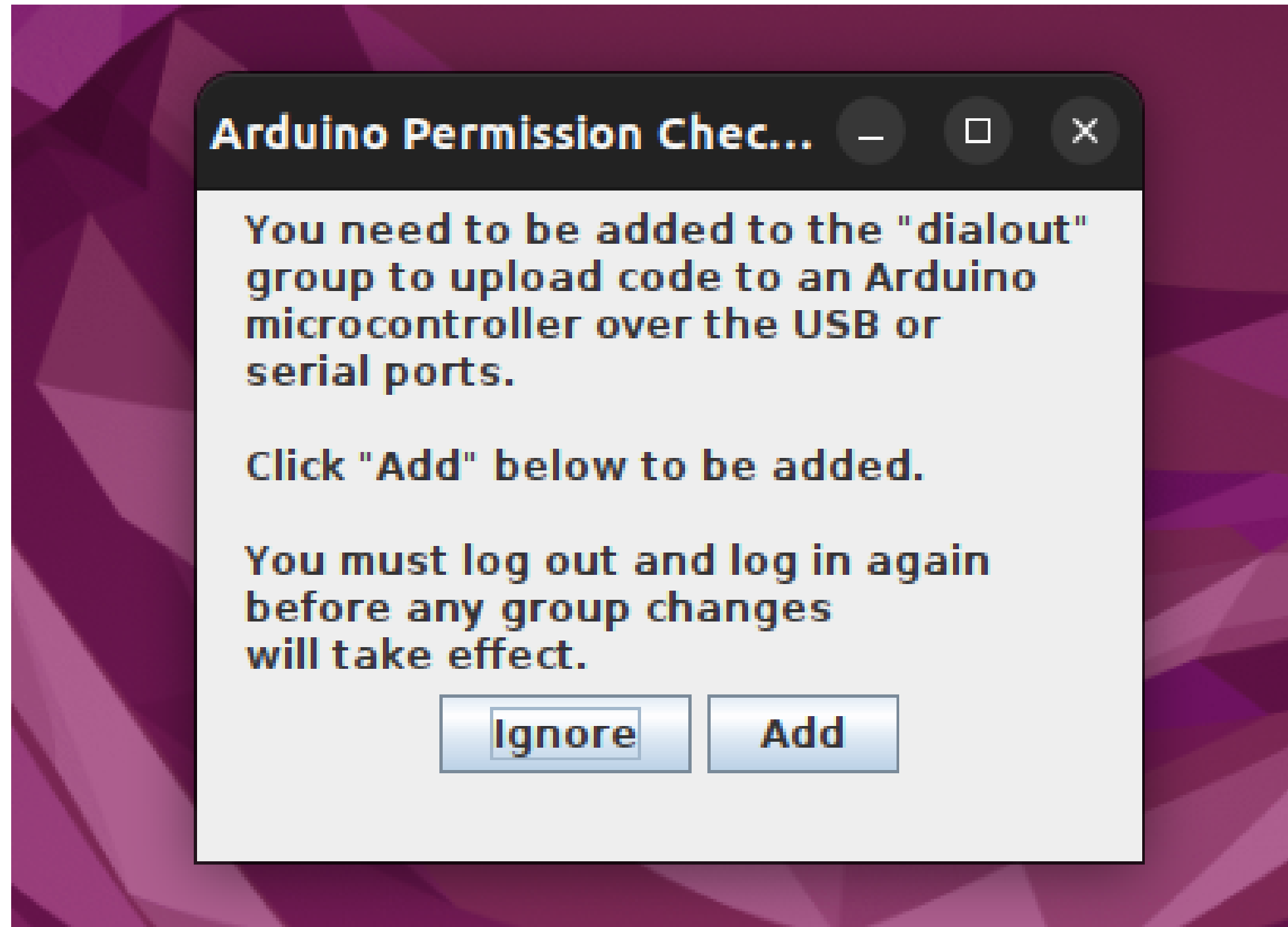
Installation of Arduino IDE

❑ Arduino IDE installation by snap-store 1



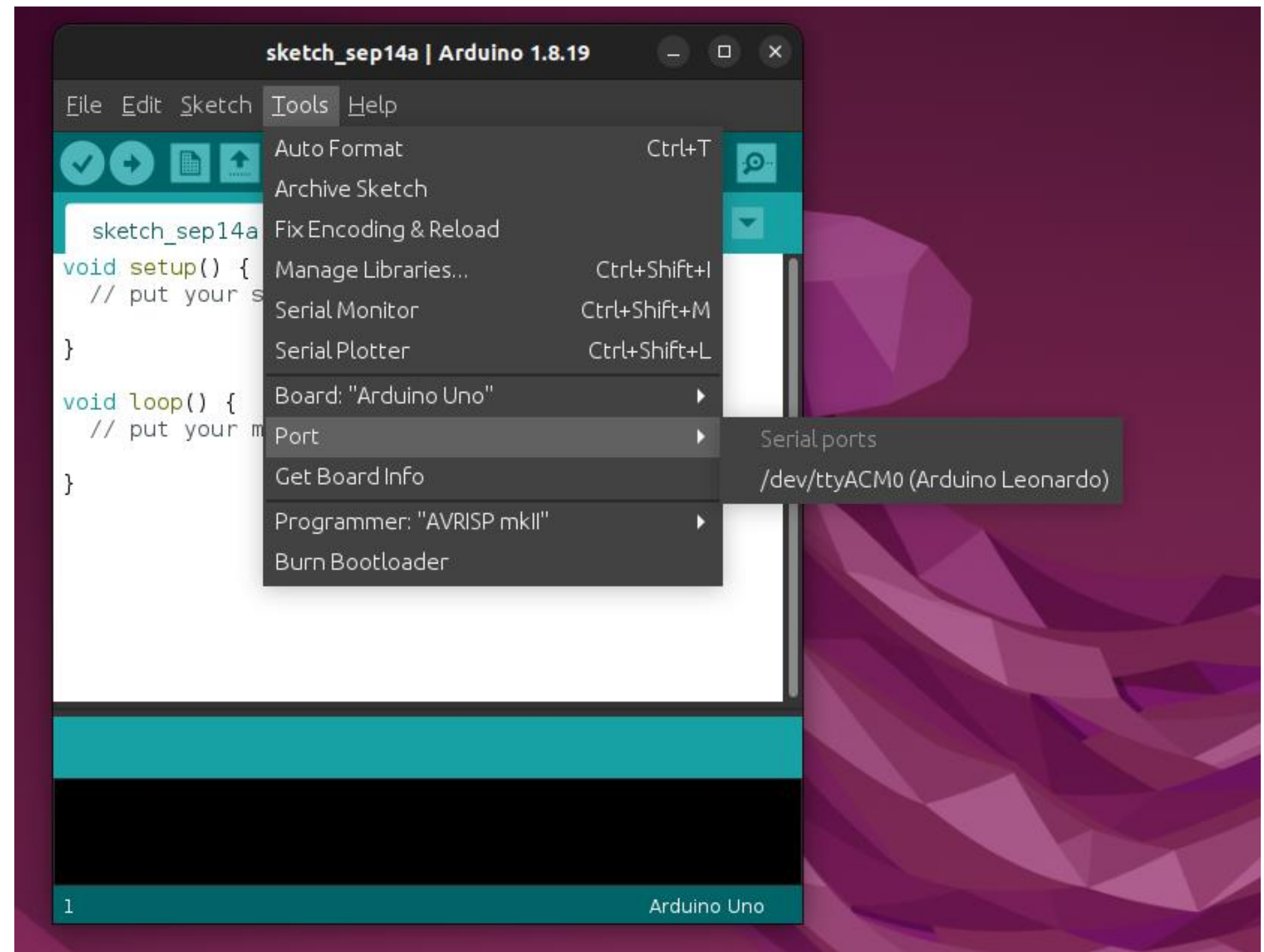
Installation of Arduino IDE

- ❑ Arduino IDE installation by snap-store 2



Installation of Arduino IDE

❑ Arduino IDE installation by snap-store 3



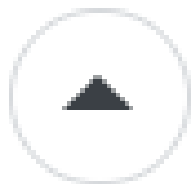
Installation of Arduino IDE

- ❑ Arduino IDE installation by snap-store 4
- ❑ CH340 USB-to-Serial Port Driver Installation or using a braille display

4 Answers

Sorted by:

Highest score (default)

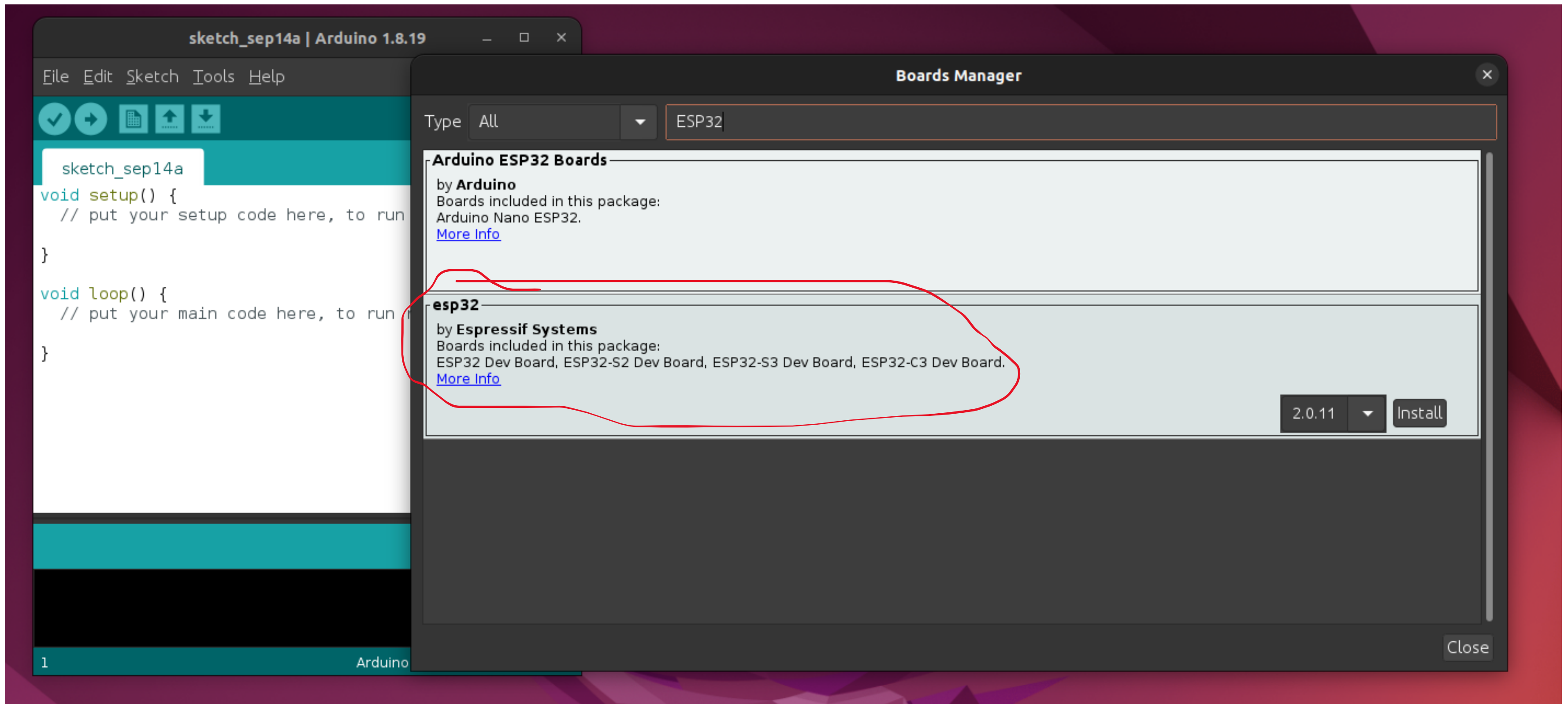


Unless you are using a braille display this should do the trick:

50

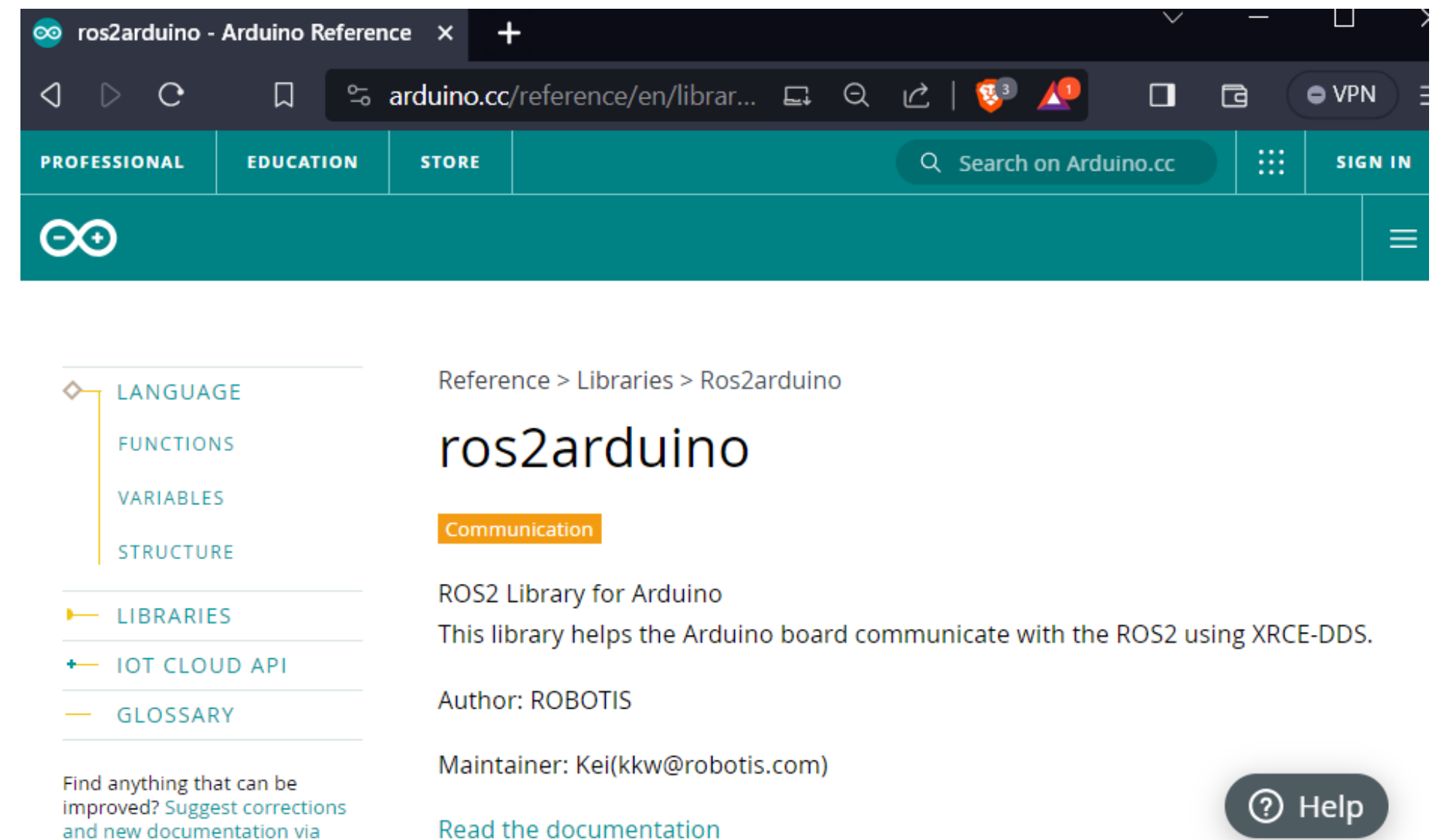
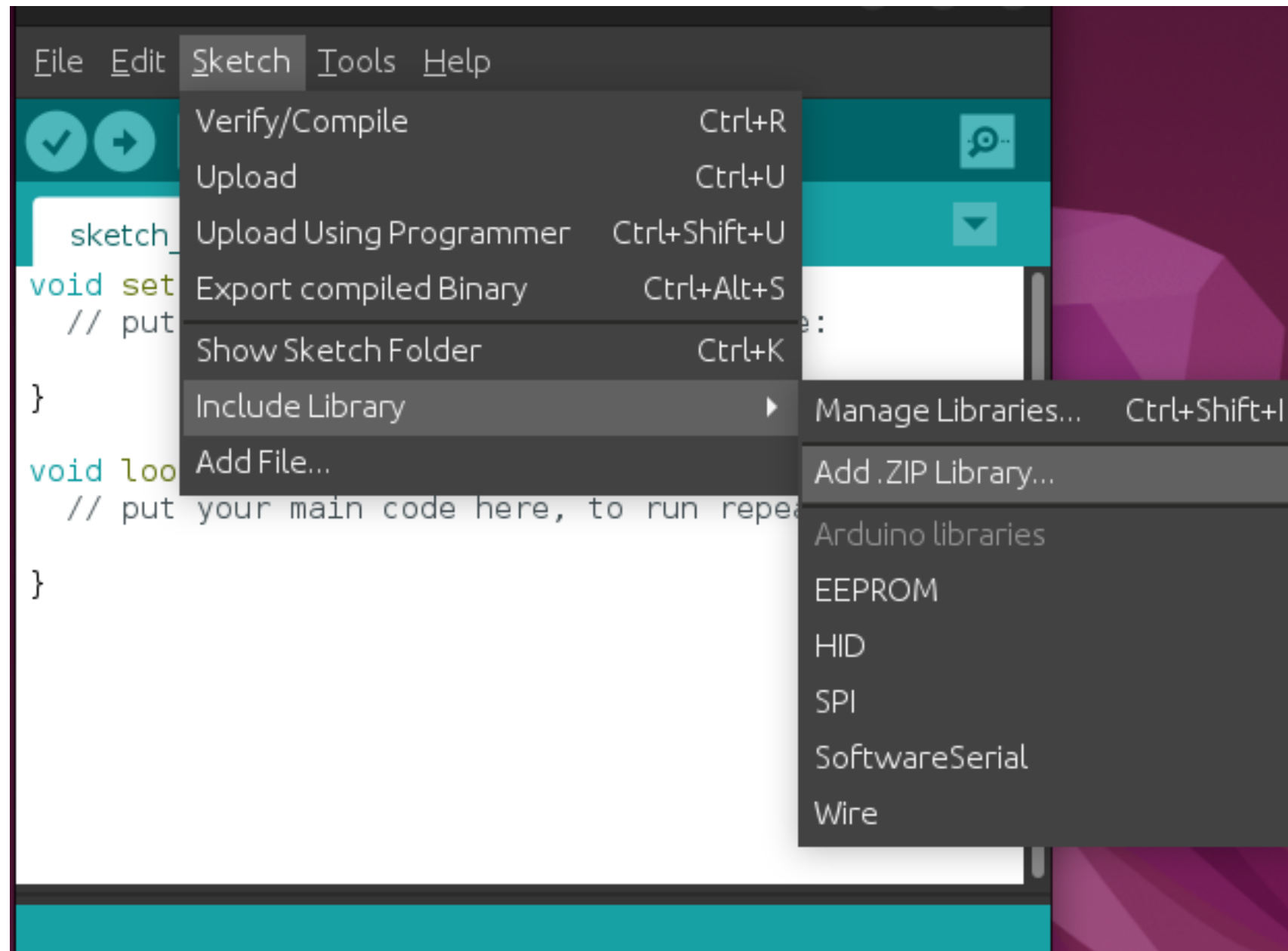
```
sudo apt remove brltty
```


Board Manager and Include Library

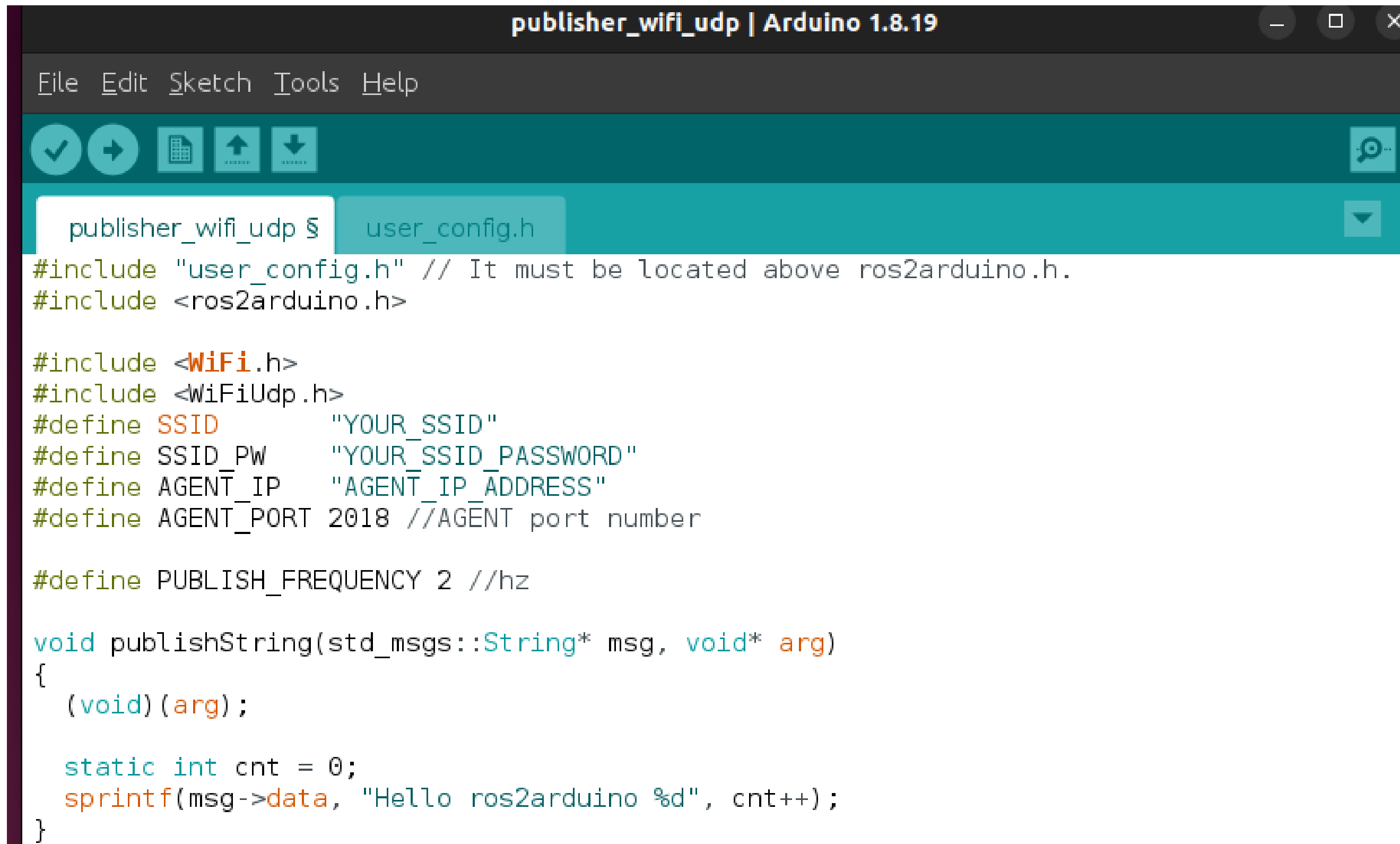


Board Manager and Include Library

- ❑ Download ros2arduino library from Arduino Official Website
- ❑ <https://www.arduino.cc/reference/en/libraries/ros2arduino/>
- ❑ Install ros2arduino library with “Add .Zip Library feature”



Pubulisher wifi udp example

A screenshot of the Arduino IDE interface. The title bar reads "publisher_wifi_udp | Arduino 1.8.19". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for checking, running, uploading, and downloading. The tab bar shows two tabs: "publisher_wifi_udp" (active) and "user_config.h". The main editor area contains the following C++ code:

```
#include "user_config.h" // It must be located above ros2arduino.h.
#include <ros2arduino.h>

#include <WiFi.h>
#include <WiFiUdp.h>
#define SSID "YOUR_SSID"
#define SSID_PW "YOUR_SSID_PASSWORD"
#define AGENT_IP "AGENT_IP_ADDRESS"
#define AGENT_PORT 2018 //AGENT port number

#define PUBLISH_FREQUENCY 2 //hz

void publishString(std_msgs::String* msg, void* arg)
{
    (void)(arg);

    static int cnt = 0;
    sprintf(msg->data, "Hello ros2arduino %d", cnt++);
}
```

Pubulisher wifi udp example

```
class StringPub : public ros2::Node
{
public:
    StringPub()
    : Node("ros2arduino_pub_node")
    {
        ros2::Publisher<std_msgs::String>* publisher_ =
            this->createPublisher<std_msgs::String>("arduino_chatter");
        this->createWallFreq(PUBLISH_FREQUENCY,
            (ros2::CallbackFunc)publishString, nullptr, publisher_);
    }
};

WiFiUDP udp;

void setup()
{
    WiFi.begin(SSID, SSID_PW);
    while(WiFi.status() != WL_CONNECTED);

    ros2::init(&udp, AGENT_IP, AGENT_PORT);
}
```

Pubulisher wifi udp example

```
class StringPub : public ros2::Node
{
public:
    StringPub()
    : Node("ros2arduino_pub_node")
    {
        ros2::Publisher<std_msgs::String>* publisher_ =
            this->createPublisher<std_msgs::String>("arduino_chatter");
        this->createWallFreq(PUBLISH_FREQUENCY,
            (ros2::CallbackFunc)publishString, nullptr, publisher_);
    }
};

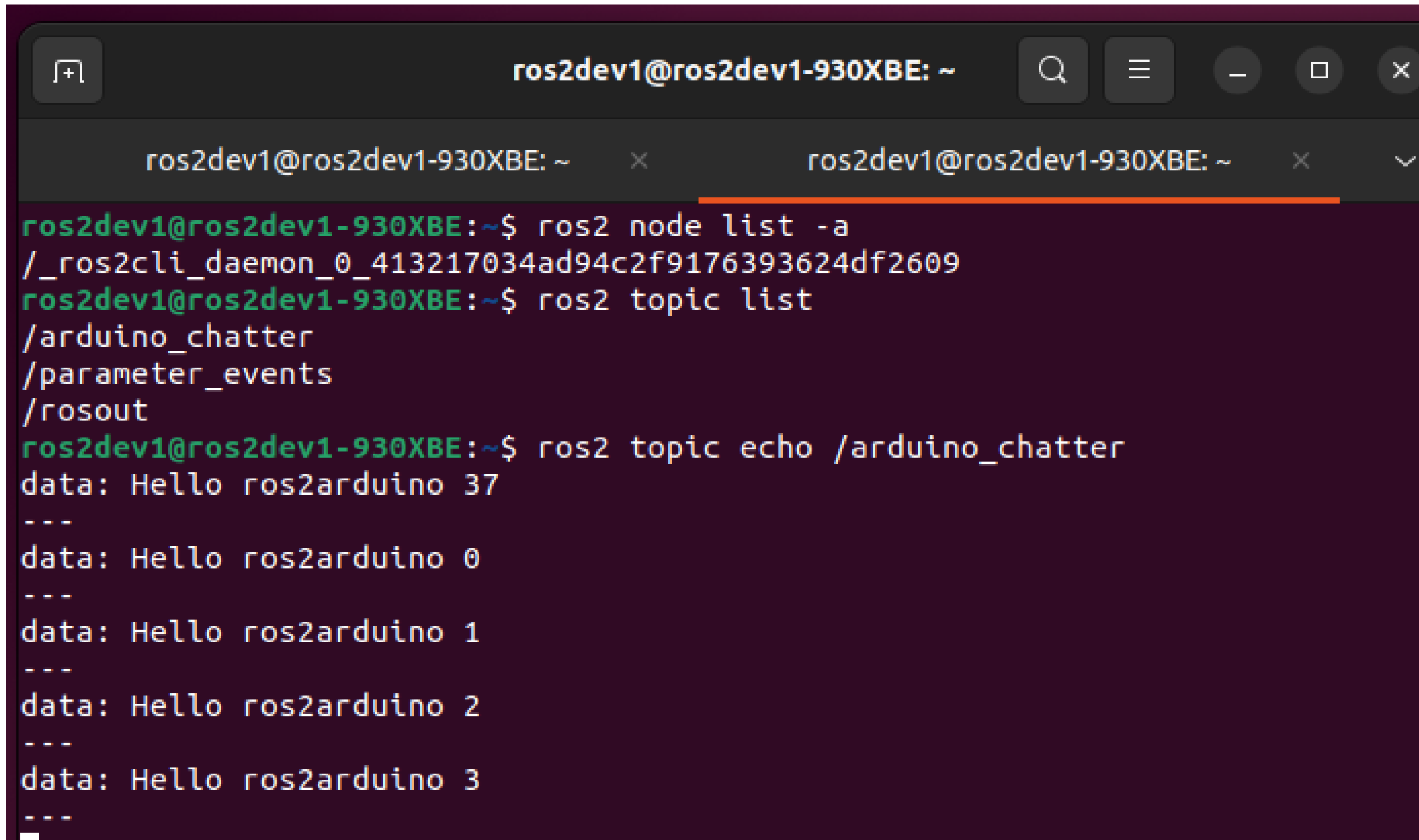
WiFiUDP udp;

void setup()
{
    WiFi.begin(SSID, SSID_PW);
    while(WiFi.status() != WL_CONNECTED);

    ros2::init(&udp, AGENT_IP, AGENT_PORT);
}
```

Pubulisher wifi udp example

- ❑ Micro-XRCE-DDS-Agent and ROS2 (dashing) are required.

A terminal window with a dark purple background. The title bar shows 'ros2dev1@ros2dev1-930XBE: ~' and standard window controls. The terminal shows the following commands and output:

```
ros2dev1@ros2dev1-930XBE:~$ ros2 node list -a
/_ros2cli_daemon_0_413217034ad94c2f9176393624df2609
ros2dev1@ros2dev1-930XBE:~$ ros2 topic list
/arduino_chatter
/parameter_events
/rosout
ros2dev1@ros2dev1-930XBE:~$ ros2 topic echo /arduino_chatter
data: Hello ros2arduino 37
---
data: Hello ros2arduino 0
---
data: Hello ros2arduino 1
---
data: Hello ros2arduino 2
---
data: Hello ros2arduino 3
---
```