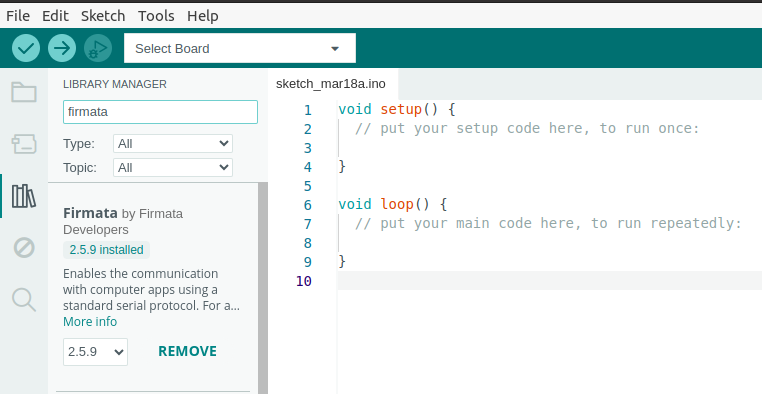
### **Uploading the Standard Firmata Sketch to Arduino Uno**

**Introduction**

The Standard Firmata sketch allows you to control your Arduino Uno from software on your computer, such as through Python scripts using the pyfirmata library. This guide will walk you through the process of uploading the Standard Firmata sketch to your Arduino Uno using the Arduino IDE.

**Step 1: Install Arduino IDE**

1. **Download** the Arduino IDE from the official Arduino website from the following link
   * https://www.arduino.cc/en/software
2. **Install** the Arduino IDE 2.0
   * Linux AppImage 64 bits (X86-64)
3. In the **Downloads** Folder, follow the instructions from the following link for **Linux**
   * https://docs.arduino.cc/software/ide-v2/tutorials/getting-started/ide-v2-downloading-and-installing/
4. On the left hand side, select LIBRARY MANAGER and install Firmata library



**Step 2: Connect Your Arduino Uno**

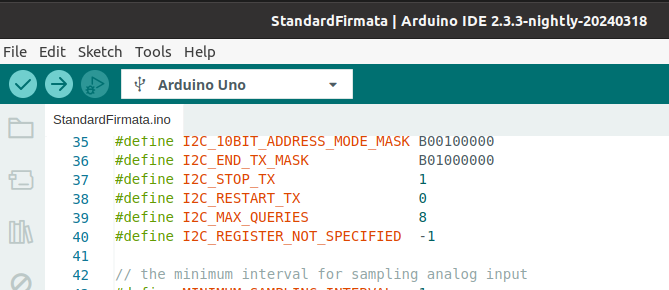
1. **Connect** the Arduino Uno to your computer using a USB cable.
2. **Power** should come on the Arduino board, indicating a successful connection.

**Step 3: Open the Standard Firmata Sketch**

1. **Launch** the Arduino IDE.
2. **Navigate** to the menu bar and select File > Examples > Firmata > StandardFirmata.  
   This action opens the Standard Firmata sketch in a new window

**Step 4: Select Your Arduino Board and Port**

1. **Go** to the drop down menu
2. **Select** Arduino Uno
3. **Note** the port that your Arduino Uno is connected to, it should be labeled with /dev/tty... in Linux



**Step 6: Upload the Sketch**

1. **Click** the upload button (right arrow icon) located in the toolbar of the Arduino IDE.
2. **Wait** for the upload process to complete. The IDE displays a message, "Done uploading," indicating the sketch has been successfully uploaded.

**Step 7: Confirm Operation**

1. Once uploaded, your Arduino Uno is ready to communicate with your computer using the Firmata protocol.
2. You can now control your Arduino Uno programmatically from your computer using various programming languages compatible with the Firmata protocol. In our Lab today, we’ll be using Python.

**Troubleshooting Tips**

* If the IDE cannot find your Arduino, recheck the cable connection and the port selection
* Make sure you've selected the correct serial port for your board

### **Arduino Error**

avrdude: ser\_open(): can't open device "/dev/ttyACM0": Permission denied Failed uploading: uploading error: exit status 1

### **1. Add Your User to the dialout Group**

Linux restricts access to serial ports to members of the dialout group. Adding your user account to this group should resolve the permission issue. You can do this by running the following command in the terminal:

sudo usermod -a -G dialout $USER

After running this command, you will need to log out and then log back in for the group changes to take effect. This change is permanent and will allow your user to access serial devices without needing to use sudo or change permissions each time.

### **2. Change Permissions Temporarily**

If you prefer not to modify user groups, you can temporarily change the permissions of the serial port with the following command:

sudo chmod a+rw /dev/ttyUSB0

This command makes the serial port readable and writable by all users. Note that this change is temporary and will be reset when the system is restarted. If you disconnect and reconnect your Arduino, you might have to run this command again.

### **Verifying Connection**

After applying either of the above solutions, try uploading your sketch or running your Python script again. If you're still encountering issues, ensure that your Arduino is correctly connected and that you're specifying the correct serial port. You can list all connected serial devices with ls /dev/ttyACM\* /dev/ttyUSB\* to verify the device name.

### Visual Studio Code

在 Ubuntu 20.04 上安裝 Visual Studio Code（VSCode）是一個簡單直接的過程。VSCode 是一個免費的開源編輯器，支援偵錯、語法高亮、智慧型代碼完成、代碼片段、代碼重構以及嵌入式 Git。以下是安裝步驟：

### **1. 更新軟體包索引**

首先，更新您的軟體包索引以確保所有系統包都是最新的。打開終端並執行以下命令：

sudo apt update

### **2. 安裝依賴**

在安裝 VSCode 之前，確保您已安裝了 wget 軟體包，這將用於下載 VSCode 包。如果您還沒有 wget，可以通過運行以下命令安裝：

sudo apt install wget -y

### **3. 下載 VSCode**

使用 wget 執行以下命令在終端中下載最新的 VSCode 包：

wget -qO- https://packages.microsoft.com/keys/microsoft.asc | gpg --dearmor > packages.microsoft.gpg

sudo install -o root -g root -m 644 packages.microsoft.gpg /usr/share/keyrings/

sudo sh -c 'echo "deb [signed-by=/usr/share/keyrings/packages.microsoft.gpg] https://packages.microsoft.com/repos/vscode stable main" > /etc/apt/sources.list.d/vscode.list'

這些命令為您的系統添加了 Microsoft GPG 密鑰和 VSCode 倉庫。

### **4. 安裝 Visual Studio Code**

現在，再次更新軟體包索引並通過運行以下命令安裝 Visual Studio Code：

sudo apt update

sudo apt install code -y

### **5. 啟動 Visual Studio Code**

安裝完成後，可以通過運行以下命令啟動 Visual Studio Code：

code

或者，您可以在系統的應用程序菜單中找到 VSCode 並從那裡啟動它。

### **6. 可選：安裝擴展**

VSCode 支援廣泛的擴展，可以增強您的開發體驗。要安裝擴展，請點擊側邊欄中的擴展圖標或按 Ctrl+Shift+X，然後搜索您想安裝的擴展。

就是這樣！您已成功在 Ubuntu 20.04 上安裝了 Visual Studio Code。享受您的編碼之旅！

### **Sample GUI Code**

1. In your **Terminal** run the following to install pyfirmata Library

sudo apt install python3-pip



pip3 install pyfirmata

1. In **Visual Studio**, create a python script called example\_GUI.py and copy the code snippet below into the new .py file
   * **Note** the connection port need to be set accordingly, comment/uncomment according to board

import tkinter as tk

from pyfirmata import Arduino, util

import time

class ArduinoGUI:

def \_\_init\_\_(self, master):

self.master = master

master.title("Arduino LED Control")

# Connect to Arduino

# Adjust your Arduino connection port accordingly

self.board = Arduino('/dev/ttyUSB0')

# self.board = Arduino('/dev/ttyUSB1')

# self.board = Arduino('/dev/ttyACM0')

# self.board = Arduino('/dev/ttyACM1')

self.led\_pin = self.board.get\_pin('d:13:o') # d for digital, 13 for pin, o for output

# Create a GUI button for toggling the LED

self.led\_state = False

self.toggle\_button = tk.Button(master, text="Turn LED on", command=self.toggle\_led)

self.toggle\_button.pack()

def toggle\_led(self):

self.led\_state = not self.led\_state

self.led\_pin.write(self.led\_state)

if self.led\_state:

self.toggle\_button.config(text="Turn LED off")

else:

self.toggle\_button.config(text="Turn LED on")

def close(self):

self.board.exit()

self.master.destroy()

# Create the GUI window

root = tk.Tk()

app = ArduinoGUI(root)

# Ensure the application cleans up properly upon exit

root.protocol("WM\_DELETE\_WINDOW", app.close)

# Run the application

root.mainloop()

**download tkinter**

sudo apt-get install python3-tk