[Raspberry Pi] Kết Nối Máy Tính Đến Raspberry Pi

 Vũ Văn Cơ 28 - 08 - 2020

♥️Chào các bạn!♥️

Khi sử dụng Raspberry thì việc kết nối từ laptop/PC đến chúng là điều rất quan trọng, bài viết của mình sẽ hướng dẫn các bạn kết nối trên 2 nền tảng hệ điều hành là Windows và Linux. Về cơ bản chúng ta có 3 cách kết nối phổ biến: giao tiếp qua SSH, sử dụng USB UART và chức năng Remote Desktop.

Bài viết của mình dựa trên nguyên tắc chia sẻ cộng đồng, đa phần nội dung là kiến thức mở mình tham khảo trên Internet mà mình tổng hợp được, rất mong các bạn theo dõi và đóng góp ý kiến hoàn thiện.

***1. Kết nối qua SSH***

***1.1 Kết nối SSH trong Windows với SSH Client PuTTY***

***1.2 Kết nối SSH trong Linux với Terminal***

***2. Kết nối qua USB UART***

***2.1 Với Windows OS***

***2.2 Với Linux OS***

***3. Truy nhập Raspberry Pi qua Remote Desktop***

***3.1 Với Windows OS***

***3.2 Với Linux OS***

*\* Khi cài Raspberry Pi (mình tạm gọi là Pi) ngay ban đầu, chắc chắn bạn đã Enable ssh trên Pi thì chúng ta mới có thể truy cập nó một cách dễ dàng được. Cách mình hay làm và cũng là cách nhanh gọn nhất là rút thẻ nhớ cài đặt, cắm vào PC và tạo một file tên "ssh" trên đường dẫn****/boot/****của Pi:*

*- Nếu cài Pi bằng máy Windows: tạo file****shh.txt****(click chuột phải, chọn New -> chọn Text Document).*

*- Nếu cài Pi bằng máy GNU/Linux: gõ lệnh****$ touch ssh***

*\* Trong quá trình cấu hình Raspberry lần đầu thì việc tìm IP mạng của chúng là cần thiết, các bạn có thể tìm IP Raspberry (sau khi cắm mạng LAN) bằng nhiều cách, tuy nhiên có 3 cách phổ biến mình hay dùng:*

*- Sử dụng host Gateway mạng bằng cách truy cập địa chỉ IP: 192.168.1.1 (trên trình duyệt) rồi đăng nhập vào host mạng và xem IP các thiết bị con, Raspberry thông thường sẽ có tên truy cập là Raspberry Pi Foundation.*

*- Sử dụng nmap (Trên Linux):*

*+ Tải nmap tại*[*đây*](https://nmap.org/download.html)*hoặc gõ:*

***$ sudo apt-get install nmap***

*+ Gõ command line trên Terminal để scan:*

***$ sudo nmap -sn 192.168.2.0/24***

*- Sử dụng Advanced IP Scanner (Windows)*

*+ Tải về tại*[*đây*](https://www.advanced-ip-scanner.com/vi/)

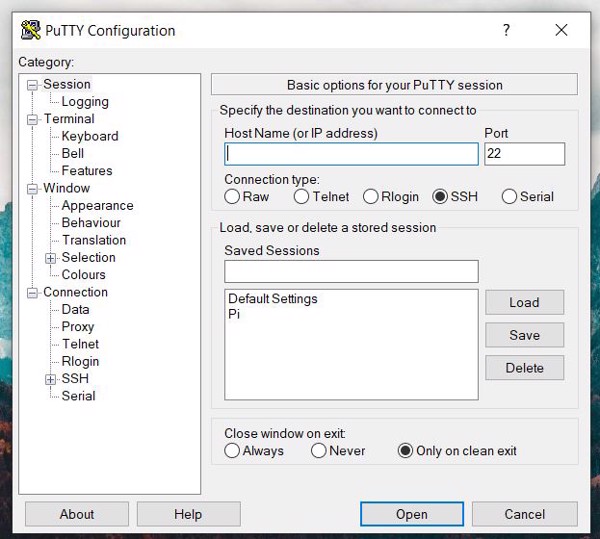
*+ Click Scan để quét*

**1. Kết nối qua Secure Shell (SSH)**

**1.1 *Kết nối SSH trong Windows với SSH Client PuTTY***

Các bạn tải SSH Client PuTTY tại [đây.](https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html)

- Mở PuTTY

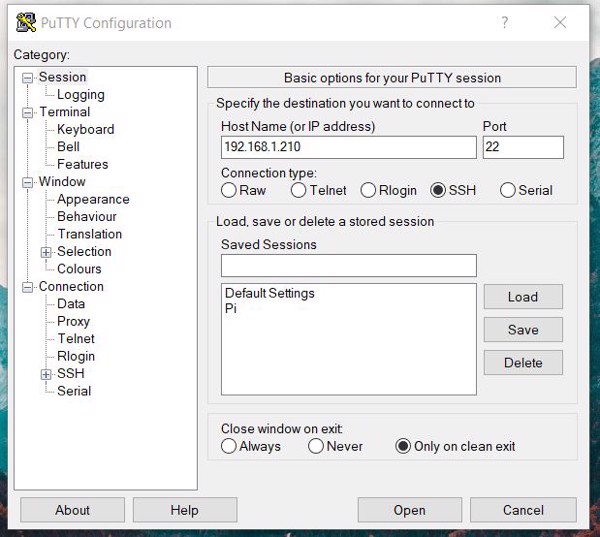


- Cài các thông số:

+ Chọn **Connection type**: giao thức **SSH**

+ Chọn **Port** logic : **22**

+ Nhập **Host name** (**IP address**) của Raspberry đã tìm được, ví dụ mình nhập 192.168.1.210



Nếu đã cài IP tĩnh cho Raspberry thì bạn có thể lưu lại thiết lập, còn không chúng ta sẽ nhấn Open để mở luôn kết nối.

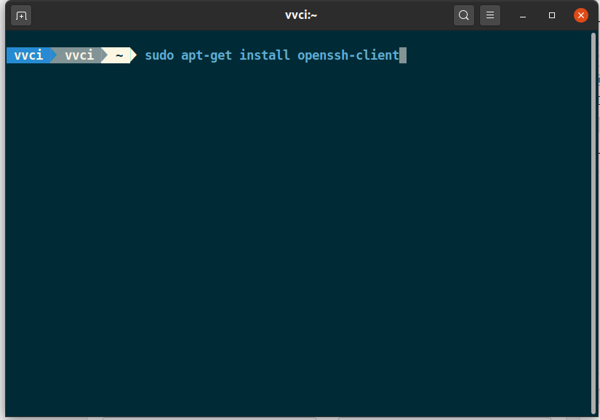
Nhập Username: ***"pi"***

Nhập Password: mặc định là ***"raspberry"***

**1.2. Kết nối SSH Client trong Linux với Terminal**

- Kiểm tra ssh đã được cài trên Linux hay chưa, mở Terminal và gõ lệnh:

**$ *sudo apt-get install openssh-client***

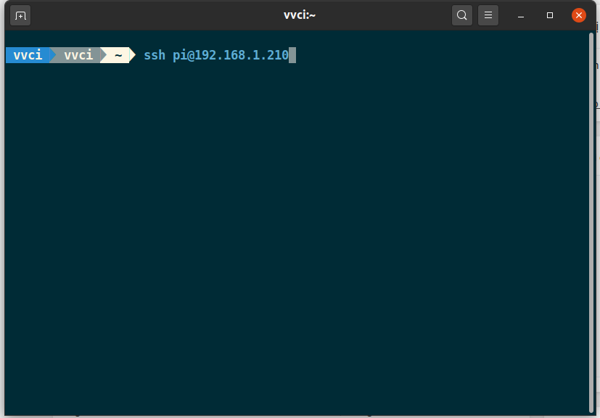


*- Kết nối, mở Terminal và gõ lệnh:*

***$ ssh pi@<IP Raspberry Pi>***

ví dụ IP Raspberry của mình là 192.168.1.210 thì mình sẽ gõ:

***$ ssh pi@192.168.1.210***



Nhập Username và Password để truy cập Raspberry.

**2. Kết nối qua USB UART (Chế độ Login Shell)**

**2.1 Trên Windows OS**

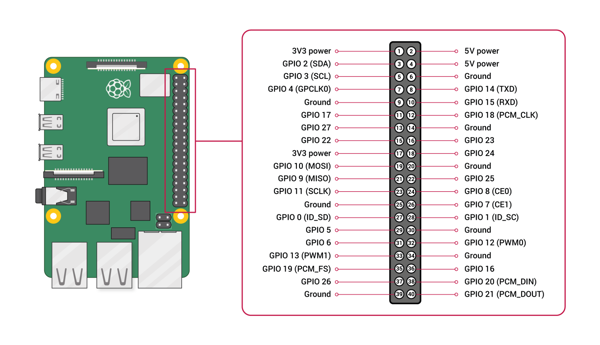
Các bạn chuẩn bị [**USB UART**](https://epcb.vn/products/usb-uart-ttl-cp2102) và phần mềm PuTTY đã tải

- Kết nối chân tín hiệu từ USB UART đến Raspberry như sau:

**GND - GND**

**RX - TX**

**TX - RX**



*\* Do đã cấp nguồn cho Pi và chuẩn UART giao tiếp chỉ cần 3 dây GND, RX, TX là đủ, mình khuyến cáo các bạn****không nên cắm chân nguồn 3.3V/5V DC từ USB UART đến Pi****để đảm bảo an toàn cho Pi.*

- Mở Device Manager trên Windows, check xem cổng COM đã nhận từ USB là bao nhiêu.

- Mở phần mềm PuTTy, cài các thông số sau:

+ Chọn **Connection type : Serial**

+ Chọn **Serial line : COMx**(phù hợp cổng COM đã tìm, ví dụ COM3)

+ Chọn **Speed : 115200 Baudrate**

**+**Nhấn **Open**

**2.2 Trên Linux OS**

- Tiến hành kết nối chân USB UART đến Raspberry như trên.

**-** Trên **Linux OS**các bạn có thể thao tác với Raspberry bằng công cụ Minicom, các bạn xem tại [đây](https://epcb.vn/blogs/news/console-linux-voi-cong-cu-minicom) mình đã giải thích rất chi tiết.

**3. Truy nhập Raspberry Pi qua Remote Desktop**

Trong 2 phương pháp kết nối từ PC đến Raspberry trình bày ở trên, tất cả chỉ có thể truy cập thông qua giao diện console (cửa sổ dòng lệnh). Vậy nếu bạn muốn truy cập giao diện desktop (đồ hoạ) thì sẽ làm như thế nào? Chúng ta sẽ sử dụng tool xrdp để làm điều đó.

Cài đặt **xrdp** trên Raspbian

+ Truy cập giao diện điều khiển dòng lệnh của Raspberry Pi (như một trong 2 bước kể trên)

+ Thực hiện dòng lệnh :

**$ *sudo apt-get install xrdp***

**3.1. Với Windows OS (Sử dụng Remote Desktop Connection)**

- Nhấn **Windows + S**, sau đó nhập Remote Desktop Connection và mở ứng dụng lên.

- Nhập địa chỉ IP của Raspberry đã tìm được, sau đó nhấn **Connect**

đăng nhập username và password của Raspberry, sau đó nhấn **OK**.

**3.2 Với Linux OS**

-  Cài đặt Remmina Remote Desktop Client:

Trong *Ubuntu Software Center thực hiện tìm “Remmina”*, nếu thấy phần mềm đã được cài đặt thì các bạn có thể sử dụng, nếu thấy phần mềm chưa được cài đặt thì các bạn có thể thực hiện cài đặt.

- Mở Remmina, tiến hành thiết lập các tham số:

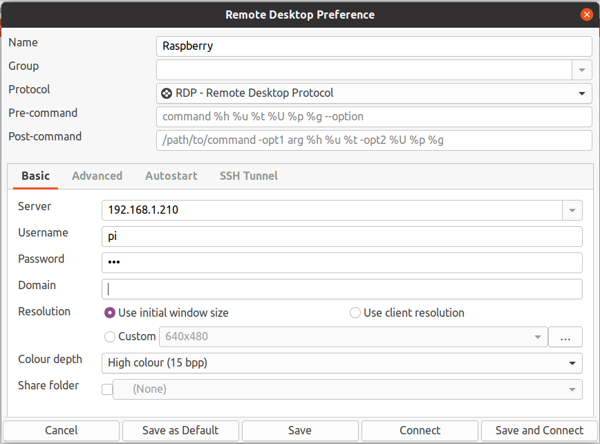
+ **Server : 192.168.1.210**(địa chỉ IP của Raspberry Pi)

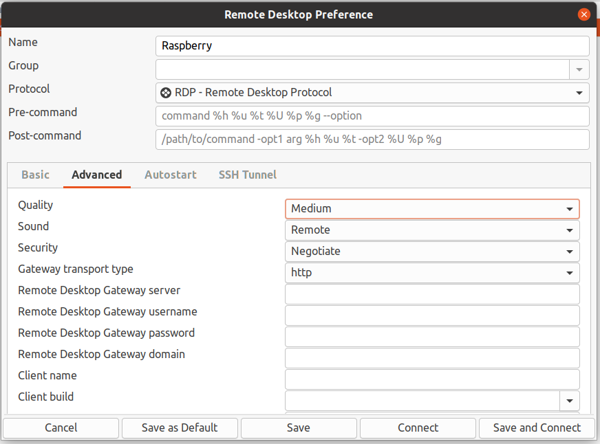
**+ User name : pi**

**+ Password :**(mặc định là "***raspberry***")

**+ Color depth : High color (15 bpp)**

**+ Quality : Medium**





Sau đó nhấn **Save and Connect**để lưu cài đặt và tiến hành kết nối, lần sau bạn chỉ cần mở Remmina lên và chọn Connect là được.

**\*Lưu ý:** Mỗi lần bạn thay đổi kết nối mạng thì moderm mạng Internet sẽ cấp lại một IP mới cho Raspberry nên chúng ta cần scan lại IP (đây gọi là cơ chế cấp IP động). Để thuận tiện cho việc kết nối đến Pi bạn nên cài IP tĩnh, trong bài viết khác mình sẽ hướng dẫn cho các bạn.

[Raspberry Pi] Hướng Dẫn Cài Đặt Và Lập Trình Raspberry Pi Pico

 CVV 12 - 05 - 2021

[Raspberry Pi Pico](https://www.raspberrypi.org/products/raspberry-pi-pico/) (Pi Pico) là board vi điều khiển hiệu năng cao, chi phí thấp được xây dựng dựa trên chip [RP2040](https://datasheets.raspberrypi.org/rp2040/rp2040-datasheet.pdf) – chip vi điều khiển được thiết kế bởi chính Raspberry Pi vừa được ra mắt vào năm 2020. Đây là bo mạch vi điều khiển đầu tiên của Raspberry Pi, được thiết kế đặc biệt cho các dự án điều khiển, lập trình nhúng. Do đó, Raspberry Pi Pico có bản chất khác hoàn toàn so với các máy tính nhúng đơn khác của hãng như Raspberry Pi 3, Raspberry Pi 4,... vì nó không hỗ trợ cài đặt hệ điều hành mà chỉ được dùng để lập trình thực hiện các tác vụ riêng lẻ. Pi Pico có thể được lập trình dễ dàng thông qua ngôn ngữ C/C++ hoặc MicroPython, đây cũng chính là 2 ngôn ngữ lập trình cơ bản hỗ trợ cho loại board vi điều khiển này.

**1. CHUẨN BỊ BOARD**

Pi Pico board hiện được bán phổ biến trên thị trường với giá cả dao động từ 4$ - 6$ với những tùy chọn khác nhau (có kèm chân jump, cáp nạp/kết nối).

Bước đầu cơ bản để chuẩn bị 1 board Pi Pico:

- Hàn chân jump (chân rào), vệ sinh board.

- Kiểm tra thông mạch, ngắn mạch.

- Cắm board Pi Pico vào break-board.



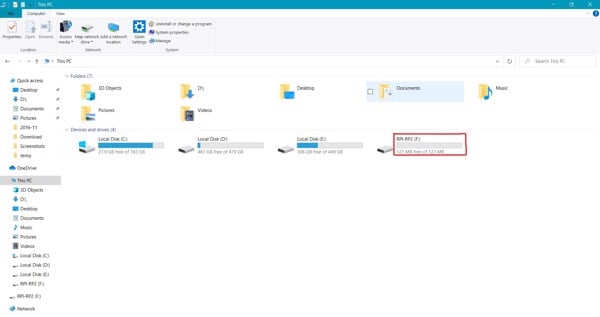
**2. CHUẨN BỊ CÔNG CỤ VÀ SDK**

**Nạp File Mirco Python**: Pi Pico hỗ trợ lập trình bằng MicroPython thông qua nạp file (*.uf2*). Các bạn tải [Micro Python](https://www.raspberrypi.org/documentation/rp2040/getting-started/#getting-started-with-micropython) để tải về file thực thi MicroPython.

**Nạp file thực thi**:

- Nhấn giữ nút BOOTSEL (nút nhấn màu trắng trên board) rồi cắm cáp kết nối vào board, sau đó cắm vào cổng USB của máy tính (PC, Raspbery Pi,...).

- Pi Pico board sẽ được kết nối máy tính như là ổ cứng lưu trữ và có tên là RPI-RP2, bạn click chuột kéo file *.uf2*mẫu vừa tảivào thư mục RPI-RP2 và reset lại board là đã hoàn thành nạp MicroPython cho board, lúc này là có thể lập trình thông qua các IDE hỗ trợ.



**Nạp file mẫu (blink LED)**:

- Tải về file Blink LED mẫu tại [đây](https://www.raspberrypi.org/documentation/rp2040/getting-started/#getting-started-with-c) (định dạng *.uf2*).

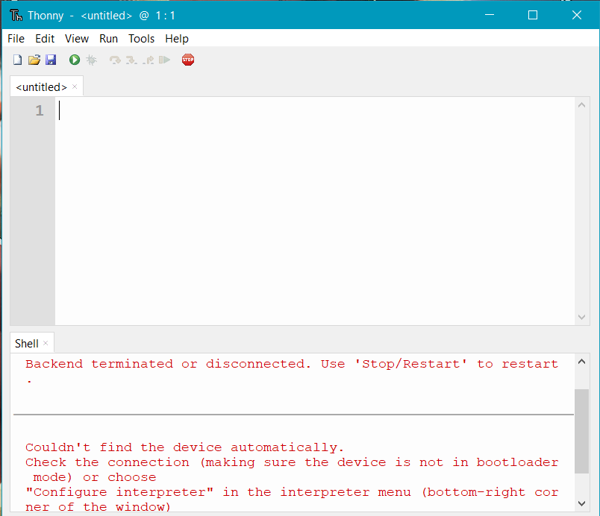
- Tiến hành nạp file *blink.uf2* như cách nạp file đã hướng dẫn ở trên.

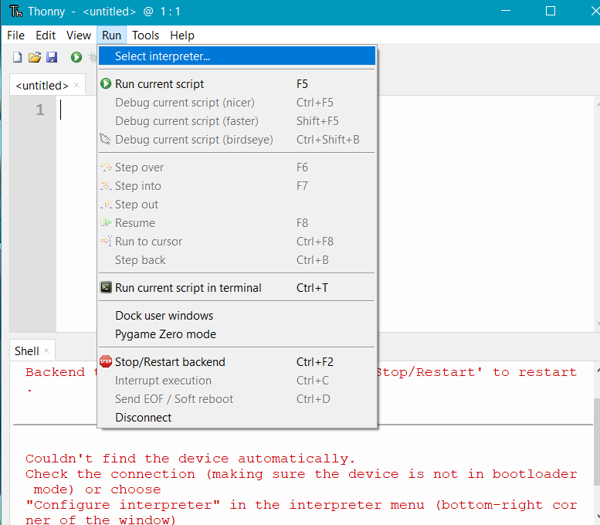
**Tải SDK**: Như đã nói, Pi Pico board hỗ trợ cả 2 ngôn ngữ Python và C/C++, 2 gói SDK này có sẵn các mã nguồn code mẫu và thư viện sẵn có, giúp bạn nhanh chóng triển khai các dự án.

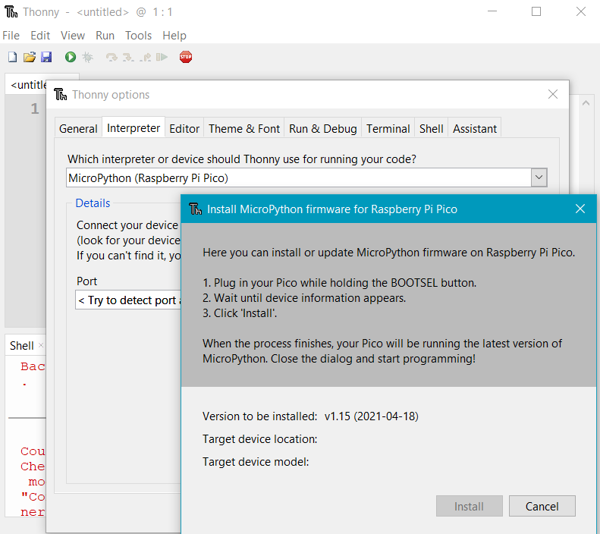
**3. CÀI ĐẶT VÀ LẬP TRÌNH PI PICO SỬ DỤNG NGÔN NGỮ PYTHON  
(SỬ DỤNG IDE THONNY)**

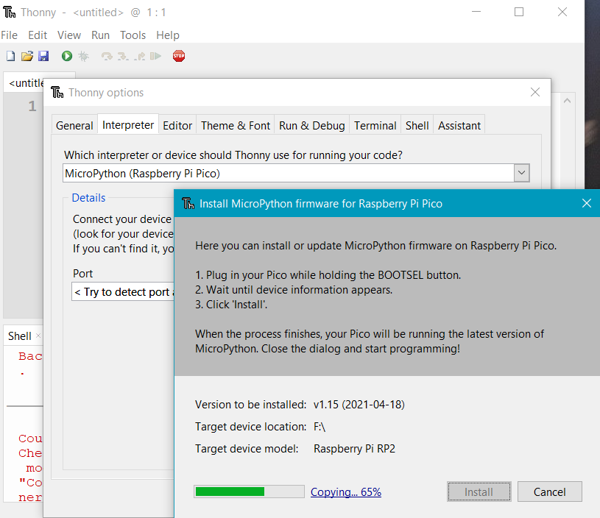
IDE khuyên dùng để lập trình cho Pi Pico bằng Python là Thonny, các bạn tải về Thonny IDE tại đây: <https://thonny.org/>. Thonny khả dụng trên cả 3 hệ điều hành Windows, Mac và Linux. Thonny hỗ trợ viết code MircoPython, tự động xuất file thực thi và nạp trực tiếp vào board Pi Pico.

- Sau khi cài đặt Thonny, bạn kết nối board đến máy tính và mở IDE lên.









- Sử dụng đoạn code mẫu blink LED sau và nhấn Run để build và nạp code trực tiếp vào board:

from machine import Pin

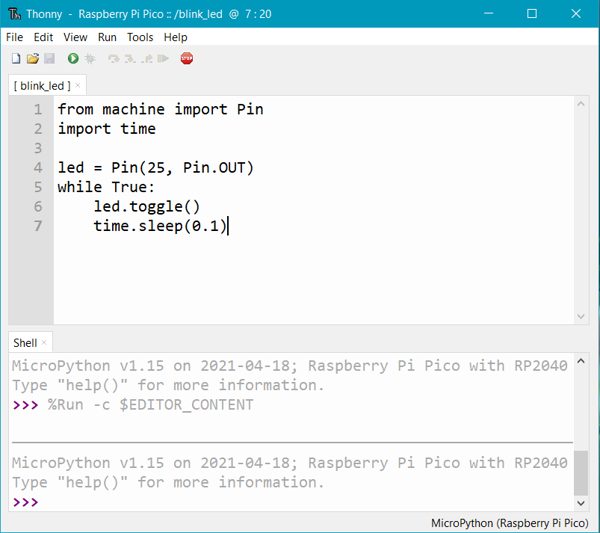
import time

led = Pin(25, Pin.OUT)

while True:

led.toggle()

time.sleep(0.1)



**Lưu ý**: Đối với Linux OS, sau khi cài đặt IDE bạn tiến hành cập nhật bằng lệnh:

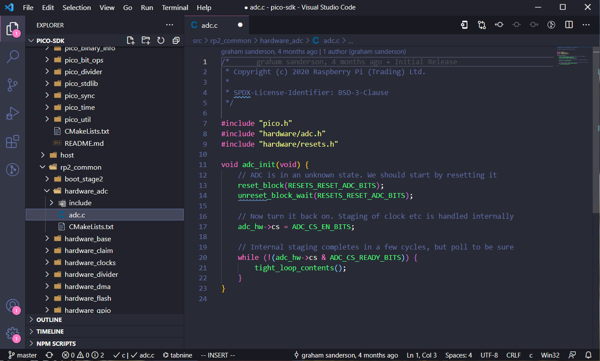
*$ sudo apt-get update && sudo apt-get upgrade -y*

để đảm bảo có phiên bản Thonny mới nhất và chọn được board Pi Pico trong thẻ Interpreter.

Các bạn có thể tải về tài liệu hướng dẫn sử dụng chi tiết về cách sử dụng bộ SDK Python của nhà sản xuất tại đây: [https://datasheets.raspberrypi.org/pico/raspberry-pi-pico-python-sdk.pdf](https://datasheets.raspberrypi.org/pico/raspberry-pi-pico-c-sdk.pdf)

**4. CÀI ĐẶT VÀ LẬP TRÌNH PI PICO SỬ DỤNG NGÔN NGỮ C/C++   
(SỬ DỤNG IDE VISUAL STUDIO CODE)**

Có thể sử dụng nhiều IDE để lập trình Pi Pico nhưng Visual Studio Code (VS Code) được nhà sản xuất khuyến khích sử dụng hơn cả. VS Code khá nhẹ, mượt và miễn phí, hỗ trợ đa nền tảng, các bạn có thể tải về tại đây: <https://code.visualstudio.com/>. Sau khi tải về VS Code và có được gói SDK, bạn import dự án và thực hiện viết chương trình bình thường giống như các chương trình C/C++ phổ biến khác.



Bản chất việc lập trình Pi Pico bằng C/C++ là chuyển đổi các file *.c* thành .*uf2* bằng VS Code hoặc bằng [**CMake**](https://viblo.asia/p/dao-dau-voi-cmake-thong-qua-vi-du-07LKXNbelV4). Ở đây, mình sẽ hướng dẫn các bạn chuyển đổi bằng CMake theo hướng dẫn của nhà sản xuất trên Linux/Raspbian OS.

**- Tạo thư mục pico tại home (hoặc workspace)**

*$ cd ~/*

*$ mkdir pico*

*$ cd pico*

**- Clone gói SDK và examples.**

*$ git clone -b master* [*https://github.com/raspberrypi/pico-sdk.git*](https://github.com/raspberrypi/pico-sdk.git)

*$ cd pico-sdk*

*$ git sub module update -init*

*$ cd ..*

*$ git clone -b master* [*https://github.com/raspberrypi/pico-examples.git*](https://github.com/raspberrypi/pico-examples.git)

**- Cài thêm các công cụ build GCC**

*$ sudo apt update*

*$ sudo apt install cmake gcc-arm-none-eabi libnewlib-arm-none-eabi build-essential*

**- Chuyển đổi file *.c* sang file *.uf2* (Chạy thử file blink.c)**

*$ cd pico-examples*

*$ mkdir build*

*$ cd build*

**cài đặt các đường dẫn:**

*$ export PICO\_SDK\_PATH=~/pico/pico-sdk*

**thực thi Cmake:**

*$ cmake ..*

*$ cd blink*

*$ make -j4*

Sau khi quá trình build thành công, file *.uf2* sẽ được tạo ra, bạn tiến hành kéo thả file vào thư mục RPI-RP2 như phần 2 đã nói.

**5. BỘ TÀI LIỆU THAM KHẢO PI PICO**

- Datasheet Raspberry Pi Pico: <https://datasheets.raspberrypi.org/pico/pico-datasheet.pdf>

- Hướng dẫn bắt đầu với Raspberry Pi Pico đầy đủ nhất: <https://datasheets.raspberrypi.org/pico/getting-started-with-pico.pdf>

- Hướng dẫn chi tiết sử dụng SDK và cách tải bộ Raspberry Pi Pico C/C++ SDK:  <https://datasheets.raspberrypi.org/pico/raspberry-pi-pico-c-sdk.pdf>

- Hướng dẫn chi tiết sử dụng SDK và cách tải bộ Raspberry Pi Pico Python SDK: <https://datasheets.raspberrypi.org/pico/raspberry-pi-pico-python-sdk.pdf>

- SDK Repo: <https://github.com/raspberrypi/pico-sdk>

**6. KẾT LẠI**

Như vậy, EPCB đã hướng dẫn các bạn một vài bước cài đặt và lập trình cơ bản với board Raspberry Pi Pico. Bài viết được rút gọn từ các chỉ dẫn của User Manual trên trang chủ của Raspberry và đúc kết từ kinh nghiệm cá nhân. Nếu có thắc mắc gì về bài viết cũng như trao đổi các thông tin kĩ thuật hữu ích, hãy liên hệ với EPCB qua [Fanpage: EPCB](https://www.facebook.com/epcb.vn) hoặc qua Email: *epcbtech@gmail.com*.

[OpenCV] Hướng Dẫn Tải Và Cài Môi Trường OpenCV

 Vũ Văn Cơ 19 - 09 - 2020

OpenCV (Open Source Computer Vision Library) là một thư viện mã nguồn mở dựa trên ngôn ngữ C++, Python và có thể bao gồm cả Java, Mathlab,... Thư viện này được sử dụng trong rất nhiều lĩnh vực với độ phổ biến và khả năng ứng dụng của chúng như: phân tích hình ảnh y tế, phân tích video, nhận biết đối tượng, ứng dụng máy học Machine Learning,...và rất nhiều ứng dụng khác nữa. OpenCV cũng có thể tận dụng sức mạnh đa lõi của bộ xử lí đồ hoạ GPU cho các ứng dụng vận hành real-time (thời gian thực).

Trang chủ cộng đồng OpenCV: https://opencv.org, bạn có thể tìm thấy tất cả phiên bản cập nhật của mã nguồn này tại đây cũng như các hướng dẫn về cách sử dụng mã nguồn.

Bài viết này hướng dẫn các bạn cài đặt OpenCV, build môi trường trên Ubuntu 16 trở lên.

**Bài viết có sử dụng tư liệu tham khảo trên trang: https://linuxize.com/**

**1. CÀI ĐẶT MÔI TRƯỜNG VÀ GÓI CÀI CẦN THIẾT CHO OPENCV**

Mở terminal, tiến hành cài các gói cài sau:

***$ sudo apt install build-essential cmake git pkg-config libgtk-3-dev \ libavcodec-dev libavformat-dev libswscale-dev libv4l-dev \ libxvidcore-dev libx264-dev libjpeg-dev libpng-dev libtiff-dev \ gfortran openexr libatlas-base-dev python3-dev python3-numpy \ libtbb2 libtbb-dev libdc1394-22-dev***

**2. TẢI VỀ REPO OPENCV**

- Tạo ra folder **opencv\_build** và đến đường dẫn đó.

***$ mkdir opecv\_build && cd ~/opencv\_build***

Đối với mình khi làm mình sẽ tạo ra đường dẫn như sau: **/home/covv/workspace/src/opencv\_build**

- Có hai cách lấy gói cài về máy tính: Clone trực tiếp từ github hoặc tải trực tiếp từ trang chủ OpenCV.

+ Clone trực tiếp:

***$ git clone https://github.com/opencv/opencv.git***

***$ git clone https://github.com/opencv/opencv\_contrib.git***

nếu bạn muốn cài các phiên bản cũ hơn của opencv và opencv\_contrib, bạn có thể dùng lệnh:

***$ git checkout <opencv-version>***

+ Tải trực tiếp từ trang chủ:

Truy cập [**https://opencv.org/releases**](https://opencv.org/releases/)và tải về các gói cài opencv.

Truy cập [**https://github.com/opencv/opencv\_contrib**](https://github.com/opencv/opencv_contrib) để tải về gói cài opencv\_contrib.

Mình hay dùng bản opencv 3.2 và opencv\_contrib 3.2, tuy hơi cũ nhưng bù lại vì độ ổn định và do quen thuộc với các chương trình của mình ^.^

**3. TẠO FOLDER BUILD**

- Copy hoặc di chuyển 2 gói cài opencv và opencv\_contrib vừa tải về vào thư mục opencv\_build vừa tạo ở bước 2.

- Thực hiện dòng lệnh:

***$ cd ~/opencv\_build/opencv***

***$ mkdir build && cd build***

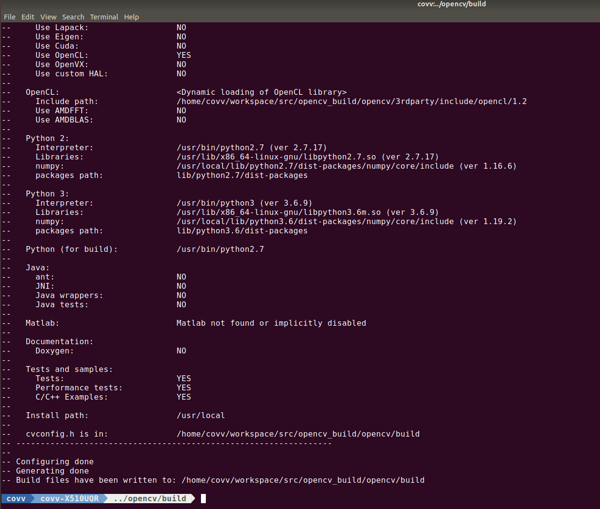
- Setup môi trường build OpenCV với CMake:

**$ cmake -D CMAKE\_BUILD\_TYPE=RELEASE \  
-D CMAKE\_INSTALL\_PREFIX=/usr/local \  
-D INSTALL\_C\_EXAMPLES=ON \  
-D INSTALL\_PYTHON\_EXAMPLES=ON \  
-D OPENCV\_GENERATE\_PKGCONFIG=ON \  
-D OPENCV\_EXTRA\_MODULES\_PATH=~/opencv\_build/opencv\_contrib/modules \  
-D BUILD\_EXAMPLES=ON ..**

Các bạn chú ý đến OPENCV\_EXTRA\_MODULES\_PATH phải là đường dẫn chính xác module của bạn.

Đối với máy tính của mình thì đường dẫn là: **/workspace/src/opencv\_build/opencv\_contrib/modules \**

Sau khi setup môi trường thành công, màn hình terminal thông thường sẽ hiển thị như sau:



**4. BIÊN DỊCH OPENCV**

Gõ dòng lệnh:

***$ make -j4*** (có thể là make -j8, -j16,..) tùy thuộc vào số thread CPU máy tính của bạn, số thread càng cao thì biên dịch càng nhanh.

Gõ: ***$ sudo nproc*** để biết được số thread.

Đoạn này các bạn có thể chuyển máy làm việc khác hoặc uống một tách cà phê vì sẽ mất thời gian đấy.

**5. CÀI ĐẶT OPENCV**

**$ sudo make install**

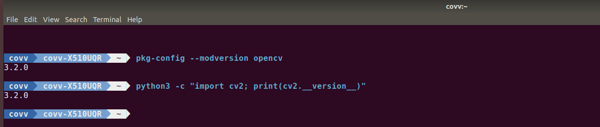
**6. KIỂM TRA VERSION**

***$ pkg-config --modversion opencv***

**3.2.0 -->**Đây là bản mình cài cho máy tính của mình

***$ python3 -c "import cv2; print(cv2.\_\_version\_\_)"***

**3.2.0 -->** Đây là bản mình cài cho máy tính của mình



Như vậy là các bạn đã hoàn tất quá trình cài OpenCV trên máy tính Ubuntu của mình. Từ đây, các bạn có thể lập trình các chương trình ứng dụng thư viện OpenCV với các ngôn ngữ C/C++, Python một cách dễ dàng.

Ở phần sau của chuỗi bài viết OpenCV, mình sẽ giới thiệu một vài sample code ứng dụng OpenCV, viết trên trình edit Qt Creator.

## Tự động chạy app khi khởi động Raspberry PI

 Nguyễn Nhựt Ninh 19 - 07 - 2021

# Bạn đã từng viết application trên Raspberry hay chưa ?

Nếu có thì chắc các bạn cũng gặp trường hợp là muốn application của mình được chạy mỗi khi Rasperry Pi khởi động đúng không ? Còn không thì hôm nay các bạn sẽ được biết cách viết application trên Raspberry PI nhé.

### Bước 1. Viết một ứng dụng C đơn giản nhất.

- Tạo một file tên là main.c chứa nội dung dưới đây.

#include <stdio.h>

#include <unistd.h>

*int main(void) {*

*while(1) {*

*printf("Hello Raspberry PI\n");*

*sleep(1);*

*}*

*}*

### **Bước 2. Biên dịch file main.c ra file thực thi.**

- Gõ command trên Raspberry PI.

*# gcc -g -o raspberry\_app main.c*

- Giờ bạn đã có file thực thi tên là **raspberry\_app**.

- Chạy thử nhé, bạn sẽ thấy dòng **“Hello Raspberry PI”** được print ra mỗi 1 giây.

 # ./raspberry\_app

- Copy file thực thi **“raspberry\_app” vào nơi chứa các file binary trên hệ thống.**

# sudo cp raspbery\_app usr/local/bin/

### Bước 3. Tạo service systemd tự động chạy app khi khởi động.

**Giới thiệu về SYSTEMD**

- Systemd là một công cụ với nhiều tính năng trên hệ thống Linux giúp quản lý các tiến trình trên Linux. Ở đây chúng ta sẽ sử dụng service systemd để quản lý app trên Raspberry PI.

- Tạo file tên **“raspberry.service”** chưa nội dung in nghiên dưới đây.

*[Unit]*  
*Description=gateway\_app\_manager pop service  
After=network.target  
StartLimitIntervalSec=0l  
[Service]  
Type=simple  
Restart=always  
RestartSec=2  
User=pi  
ExecStart=/usr/bin/sudo /usr/local/bin/raspberry\_app*  
  
*[Install]  
WantedBy=multi-user.target*

- Di chuyển file “r**aspberry.service**” vào thư mục chứ các service khác của hệ thông trên PI.

# sudo cp raspberry.service /etc/systemd/system/

**Giải thích nội dung cần thiết trong service “raspberry.service”**

**User=pi**

=> Khai báo tên user để chạy service này, ở đây mình dùng Raspberry PI nên user của mình là pi.

***ExecStart=/usr/bin/sudo /usr/local/bin/raspberry\_app***

=> Ở đây mình muốn chạy app với sudo nên thêm***“/usr/bin/sudo” vào service.***

=> “/usr/local/bin/raspberry\_app” là vị trí app “raspberry\_app” trên PI của mình. Bạn có thể thay thế bằng đường dẫn khác nếu bạn muốn.

### Bước 4. Enable service để chạy app lúc khởi động.

- Sau khi tạo mới hoặc chỉnh sữa một service nào đó chúng ta cần reload lại hệ thống systemd.

 # sudo systemctl daemon-reload

- Enable service vừa khởi tạo.

# sudo systemctl enable raspberry.service

- Sau khi enable, hãy reboot Raspberry PI để test xem service có hoạt động không nhé.

**- Chạy service.**

     # sudo systemctl start raspberry.service

- Kiểm tra trạng thái service, nếu thành công thì kết quả như hình sau.

    # sudo systemctl status raspberry.service



***- Tạm dừng service***

*# sudo systemctl stop raspberry.service*

=> Sau khi reboot service sẽ tự động chạy lại.

**- Không cho phép chạy service cho đến khi được enable lại.**

*# sudo systemctl disable raspberry.service*