

Run Ros1 on Ubuntu 20.04 of KV260

下載 KV260 版本的 Ubuntu 20.04

Download Link:

[Install Ubuntu on Xilinx | Ubuntu](#)

Ubuntu Desktop 20.04.3 LTS

The version of Ubuntu with up to 10 years of long term support, until April 2030.

Download 64-bit

Works on:

✓ Xilinx Kria KV260 Vision AI Starter Kit



設定 KV260 上的 Ubuntu 20.04 並安裝 Wi-Fi Driver

此篇內容參考至以下教學：

[Adding USB-WiFi for the Kria KV260 \(Ubuntu 20.04.3\) - Hackster.io](#)

詳細步驟：

1. 更新 Ubuntu 20.04 Package

```
sudo apt update
sudo apt upgrade
sudo reboot
```

2. 設定 Xilinx Development

```
sudo snap install xlnx-config --classic --channel=1.x
xlnx-config.sysinit
sudo reboot
```

3. 安裝 Wi-Fi Driver

```
sudo apt install -y dkms git build-essential
mkdir repos; cd repos
git clone https://github.com/morrownr/88x2bu-20210702
cd 88x2bu-20210702
sudo ARCH=arm64 ./install-driver.sh
```

4. 測試 Wi-Fi Driver

```
iwconfig
lo          no wireless extensions.

eth0        no wireless extensions.

sit0        no wireless extensions.

wlx1cbfceedb7e7b IEEE 802.11  ESSID:off/any
              Mode:Managed  Access Point: Not-Associated   Tx-Power=-100 dBm
              Retry short limit:7   RTS thr:off   Fragment thr:off
              Power Management:on
```

notice: wlx1cbfceedb7e7b 為偵測到的 Wi-Fi 裝置，每個人不一定一樣

接著打開 Wi-Fi 並搜尋目前能找到的熱點資訊

```
sudo apt install net-tools
sudo ifconfig wlx1cbfceedb7e7b up
sudo iwlist wlx1cbfceedb7e7b scan | grep ESSID
```

有搜尋到的話會像以下：

```
ubuntu@kria:~$ sudo iwlist wlx1cbfceedb7e7b scan | grep ESSID
ESSID:"BELL113"
ESSID:"ramduq"
ESSID:"BELL709"
ESSID:"BELL113"
ESSID:"ramduq"
ESSID:""
ESSID:"BELL709"
ESSID:"HP-Print-BB-Officejet Pro X476dw"
```

5. 連接 Wi-Fi

將 Wi-Fi 設定寫入到 config 內

```
wpa_passphrase your-ESSID your-passphrase | sudo tee /etc/wpa_supplicant.conf
eg. wpa_passphrase "BELL709" 12345678 | sudo tee /etc/wpa_supplicant.conf
```

連接

```
sudo wpa_supplicant -c /etc/wpa_supplicant.conf -i wlx1cbfceedb7e7b -B
```

連接成功之後給它一個 IP

```
sudo dhclient wlx1cbfceedb7e7b
```

測試網路是否有通

```
ping 8.8.8.8
```

成功的話會像下圖：

```
ubuntu@kria:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=116 time=8.19 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=116 time=7.45 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=116 time=8.51 ms
^C
--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 7.453/8.052/8.511/0.443 ms
```

在 KV260 Ubuntu 20.04 上安裝 ROS1

直接下載以下這包：

git clone https://piBOT_stm32:piBOT-zeus@code.aliyun.com/surewin1102/piBOT_ros.git

下載之後 cd 到 piBOT_ros

更改 piBOT_install_ros.sh 130 行後的内容為以下：

```
# read -s -n1 -p "install ros gui tools?(y/N)"
# if [ "$REPLY" = "y" -o "$REPLY" = "Y" ]; then
sudo apt-get -y --allow-unauthenticated install ros-${ROS_DISTRO}-rviz ros-${ROS_DISTRO}-rqt-reconfigure ros-${ROS_DISTRO}-rqt-tf-tree \
    ros-${ROS_DISTRO}-image-view

    if [ "$ROS_DISTRO" = "noetic" ]; then
        echo "please run ros_package/make_cartographer.sh to compile cartographer"
    else
        sudo apt-get -y --allow-unauthenticated install ros-${ROS_DISTRO}-cartographer-rviz
    fi
# fi
```

接著執行 sh piBOT_install_ros.sh 進行安裝

安裝結束後在 piBOT_ros 資料夾內進行 ./piBOT_init_env.sh 以進行 piBOT_ros 的環境設定
會要求輸入數字，依序為：

0 0 2 3 0

輸入結束後要記得 source 環境變數：

source ~/.bashrc

再來要 make piBOT_ros 形成動態連結檔

```
cd pibot_ros/ros_ws
catkin_make
```

再來等 make 結束後再 source ~/.bashrc 一次就可以了

以上就是 ROS1 在 KV260 Ubuntu 20.04 上的設定與安裝

在 KV260 Ubuntu 20.04 上安裝 cp210x Driver

基本上是按照 Xilinx 官方 Wiki 去做 custom Ubuntu function modified :

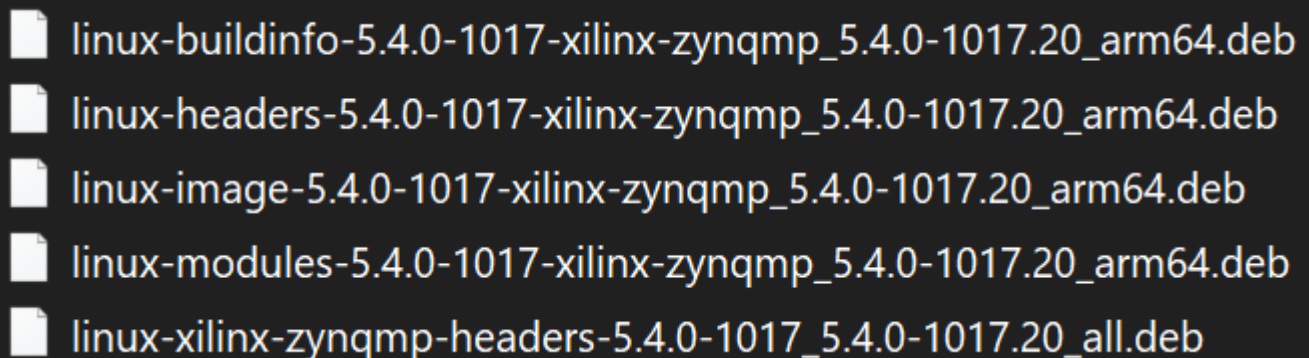
[Rebuilding the Certified Ubuntu for Xilinx Devices 20.04 LTS Kernel from Source - Xilinx Wiki - Confluence \(atlassian.net\)](https://www.xilinx.com/support/answers/69022.html)

需要注意的是你需要創建一個虛擬機（Ubuntu 20.04）或是一台 Ubuntu 20.04 的主機來去 build 你所要放到 KV260 Ubuntu 20.04 上的安裝檔案

當中執行到 fakeroot debian/rules editconfigs 這個步驟時，要去開啟 cp210x driver 的功能，主要是開啟以下兩者：

- Device Drivers -> USB support -> USB Serial Converter support
- Device Drivers -> USB support -> USB Serial Converter support -> USB CP210x family of UART Bridge Controllers

上述連結內都是在另外一台 Ubuntu 20.04 上作執行，到最後產生出來的五個檔案如下：



```
linux-buildinfo-5.4.0-1017-xilinx-zynqmp_5.4.0-1017.20_arm64.deb
linux-headers-5.4.0-1017-xilinx-zynqmp_5.4.0-1017.20_arm64.deb
linux-image-5.4.0-1017-xilinx-zynqmp_5.4.0-1017.20_arm64.deb
linux-modules-5.4.0-1017-xilinx-zynqmp_5.4.0-1017.20_arm64.deb
linux-xilinx-zynqmp-headers-5.4.0-1017_5.4.0-1017.20_all.deb
```

才需要將這些檔案丟到 KV260 Ubuntu 20.04 上做安裝，而在 KV260 Ubuntu 20.04 上做安裝的指令為：

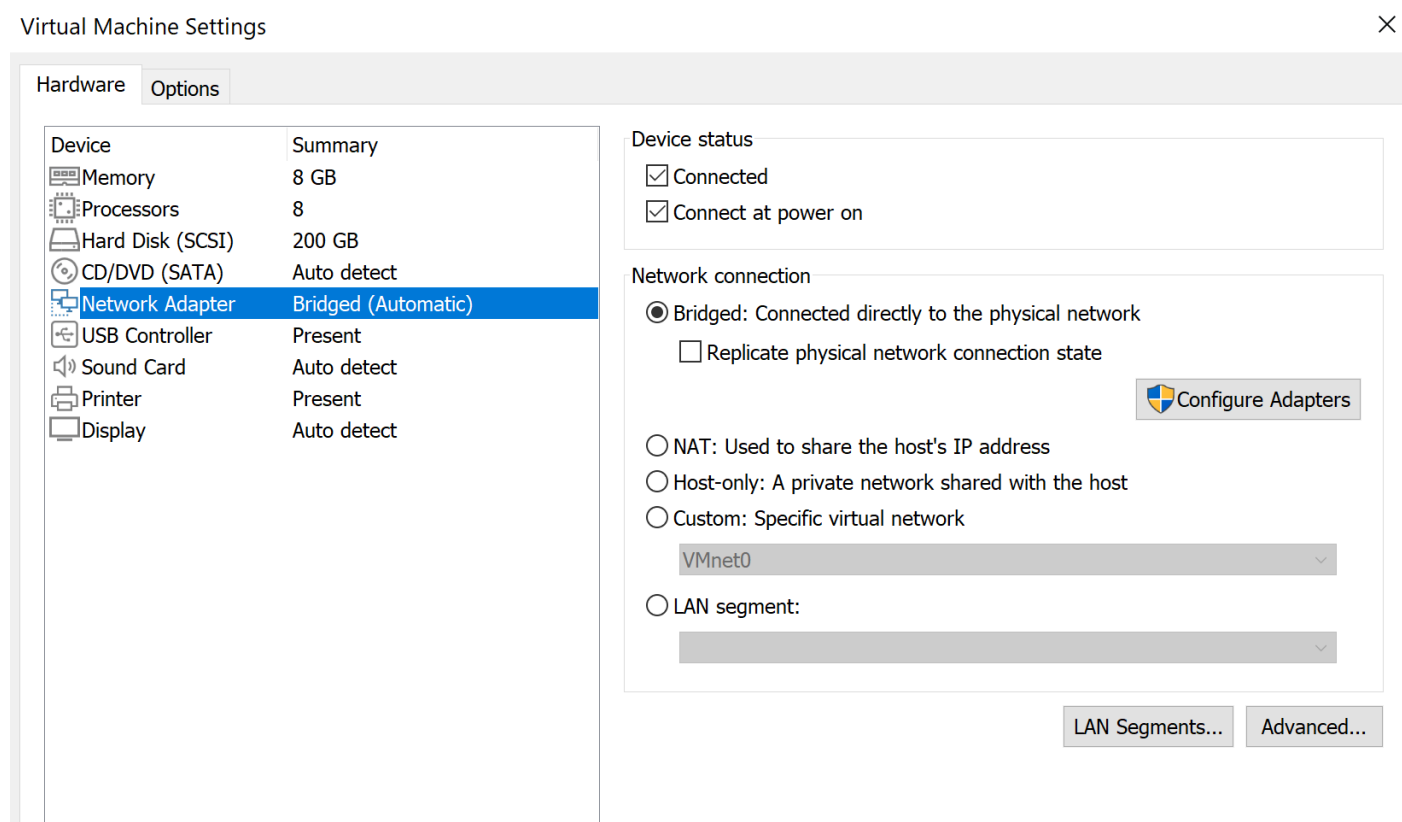
```
sudo dpkg -i *.deb
```

記得安裝完後 reboot，就可以認到 STM32F4 的 cp210x serial 了

虛擬機控制 KV260 與驅動馬達動作

此篇介紹如何透過 KV260 回傳圖像給 PC 虛擬機端，並在 PC 虛擬機端處理完圖像資訊後，傳送控制指令給 KV260 以進行馬達的前進、後退、轉彎等動作

PC 上虛擬機可以安裝 Ubuntu 18.04 或是 20.04，記得網路連接設定成 Bridged，如下：



才有辦法透過虛擬機去控制 KV260 上的馬達

再來虛擬機上也要安裝 pibot_ros 這包，詳細可參考 KV260 上 pibot_ros 的安裝，步驟一樣

安裝結束後記得執行 pibot_init_env，設定數字為：

0 2 3 0 1

最後一個數字代表透過此虛擬機來控制 KV260

並且 IP 要設定成 KV260 上的 IP

最後再 source ~/.bashrc 即可

這樣一來 KV260 和虛擬機方面就都準備好了，可以進行 Demo

Demo 流程為下：

KV260 端：

1. 開啟 ros master node 與驅動

pibot_bringup

2. 再開一個 command window 去執行 camera catch 的 example code

python camera_node.py

虛擬機端：

1. 執行影像處理與傳輸控制的 code

```
python linear_image.py
```

相關的 code 檔案放在 onedrive KR260 自走車的資料夾內部：

KR260 自走車			
 名稱 ▾	修改時間 ▾	修改者 ▾	日期 ▾
 linear_image.py	幾秒鐘前	Norris.Lin	
 camera_node.py	幾秒鐘前	Norris.Lin	
 Kria	7月6日	Hube.Chen	
 Vitis AI	7月6日	Hube.Chen	
 ROS AI光達鯊魚車線上教學 BY良棋科技-20...	8月21日	Mike.Chan	
 SmartSelect_20220711-150612_Samsung N...	7月11日	Norris.Lin	
 stm32f4_code-20220905T020235Z-001.zip	9月4日	Mike.Chan	
 STM32F407ZGT6(Mini)核心板--电路原理图.j...	9月4日	Mike.Chan	
 自走車方案套件.pptx	7月18日	Elliott.Wu	

notice: linear_image.py 內部的 code 功能都有，但還沒整理跟寫註解，所以蠻亂的，但目前可以簡單 Demo 用