

```
#####  
#                               UNO-220 Packages                               #  
#                               Installation Guide                               #  
#####
```

0. Requirements:

- UNO-220
- Raspberry Pi 4
- Debian Buster Lite Image 2019-09-30 Released for Raspberry Pi
- Ubutu 16.04 on x86_64 with RS-232 interface
(USB to RS-232 cable is ok too)

1. Download the raspbian lite image.

```
$ curl -o 2019-09-26-raspbian-buster-lite.zip  
http://downloads.raspberrypi.org/raspbian_lite/images/raspbian_lite-2019-09-  
30/2019-09-26-raspbian-buster-lite.zip
```

```
$ unzip 2019-09-26-raspbian-buster-lite.zip
```

2. Insert SD Card and if the disk is located '/dev/sdX'. 'X' can be 'e', 'f', and etc.

** The following example disk device uses 'sde'. **

3. Clone image data into SD card.

```
$ dd if=2019-09-26-raspbian-buster-lite.img of=/dev/sde bs=4M  
status=progress conv=fsync
```

4. Check partitions.

```
$ fdisk -l /dev/sde
```

...

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sde1		8192	532479	524288	256M	c	W95 FAT32 (LBA)
/dev/sde2		532480	4390911	3858432	1.9G	83	Linux

5. Enable the ssh server by modifying image, and enable i2c interface.

```
# Enable ssh
$ mkdir -p sdcard.root
$ mount /dev/sde2 sdcard.root
$ ln -sf /lib/systemd/system/ssh.service
sdcard.root/etc/systemd/system/multi-user.target.wants/ssh.service
$ umount sdcard.root

# Enable i2c
$ mkdir -p sdcard.boot
$ mount /dev/sde1 sdcard.boot
# Uncomment the parameter as below for i2c in 'config.txt'.
$ vim sdcard.boot/config.txt
...
dtparam=i2c_arm=on
..
$ umount sdcard.boot
```

6. Inset SD card and power on Raspberry Pi 4, and check your DHCP environment for Pi's IP.

7. Copy packages and connect to Pi's openssh server. (Default login: pi/raspberry)

```
$ ls *.deb
```

```
uno220rtc_0.1-1_armhf.deb      # rtc package
uno220gpio_0.1-1_armhf.deb    # gpio package
```

```
$ scp *.deb pi@${PI_IP}:~
$ ssh pi@${PI_IP}
```

```
# This is optional. If you want to use i2c-tools to debug i2c devices.
pi@raspberrypi:~ $ sudo apt-get update && sudo apt-get install -y i2c-tools
```

```
# Install rtc and gpio packages.
pi@raspberrypi:~ $ cd ~
pi@raspberrypi:~ $ sudo dpkg --install uno220rtc_0.1-1_armhf.deb
```

```
pi@raspberrypi:~ $ sudo dpkg --install uno220gpio_0.1-1_armhf.deb
```

8. RTC

```
# Get RTC time
```

```
pi@raspberrypi:~ $ sudo hwclock -r  
2020-01-13 06:34:43.545566+00:00
```

```
# Set RTC by system time
```

```
pi@raspberrypi:~ $ sudo hwclock -w
```

9. GPIO

```
# Show usage
```

```
pi@raspberrypi:~ $ sudo uno220gpio -h
```

```
Usage:
```

```
uno220gpio --export=[all|0~7]           # Export GPIO  
uno220gpio --unexport=[all|0~7]        # Unexport GPIO  
uno220gpio --pin=[0~7] --direction=[in|out] # Set GPIO Direction  
uno220gpio --pin=[0~7]                  # GPIO Read
```

```
Operation
```

```
uno220gpio --pin=[0~7] --value=[0|1]    # GPIO Write Operation  
uno220gpio --status
```

```
# Get all GPIO Status
```

```
pi@raspberrypi:~ $ sudo uno220gpio
```

```
pin      | 0  1  2  3  4  5  6  7  
-----  
export   | 0  0  0  0  0  0  0  0  
direction | X  X  X  X  X  X  X  X  
value    | X  X  X  X  X  X  X  X
```

```
# 1. Export all
```

```
pi@raspberrypi:~ $ sudo uno220gpio --export=all
```

```
pi@raspberrypi:~ $ sudo uno220gpio
```

```
pin      | 0  1  2  3  4  5  6  7  
-----  
export   | 1  1  1  1  1  1  1  1
```

```

direction |  |  |  |  |  |  |  |  |
value     |  | 1  1  1  1  1  1  1  1  1

```

2. Set direction (ex: pin=0, direction=out)

```
pi@raspberrypi:~ $ sudo uno220gpio --pin=0 --direction=out
```

```
pi@raspberrypi:~ $ sudo uno220gpio
```

```

pin       |  | 0  1  2  3  4  5  6  7
-----
export    |  | 1  1  1  1  1  1  1  1
direction |  | 0  |  |  |  |  |  |
value     |  | 0  0  1  1  1  1  1  1

```

3. Set value (ex: pin=0, direction=out, value=1)

```
pi@raspberrypi:~ $ sudo uno220gpio --pin=0 --value=1
```

```
pi@raspberrypi:~ $ sudo uno220gpio
```

```

pin       |  | 0  1  2  3  4  5  6  7
-----
export    |  | 1  1  1  1  1  1  1  1
direction |  | 0  |  |  |  |  |  |
value     |  | 1  1  1  1  1  1  1  1

```

10. Install serial package and enable serial port.

```
$ ls *.deb
```

```
uno220uart_0.1-1_armhf.deb      # serial package
```

```
$ scp *.deb pi@${PI_IP}:~
```

```
$ ssh pi@${PI_IP}
```

```
pi@raspberrypi:~ $ cd ~
```

```
pi@raspberrypi:~ $ sudo dpkg --install uno220uart_0.1-1_armhf.deb
```

```
pi@raspberrypi:~ $ sudo uno220uart --enable
```

Then, reboot Pi again.

!! If you want to fall back serial port debugging, please follow the below command. !!

```
pi@raspberrypi:~ $ sudo uno220uart --kernel=1 --systemd=1 --enable
```

11. Serial port test - PC (Ubuntu 16.04 x86-64) vs Pi

Connect PC's RS-232 TxD/RxD/GND pins connect to IO Board corresponding pins.

PC side command:

```
$ sudo ./host_send /dev/ttyUSB0 'hello!!'
```

Pi side command:

```
pi@raspberrypi:~ $ sudo uno220uartrecv
```