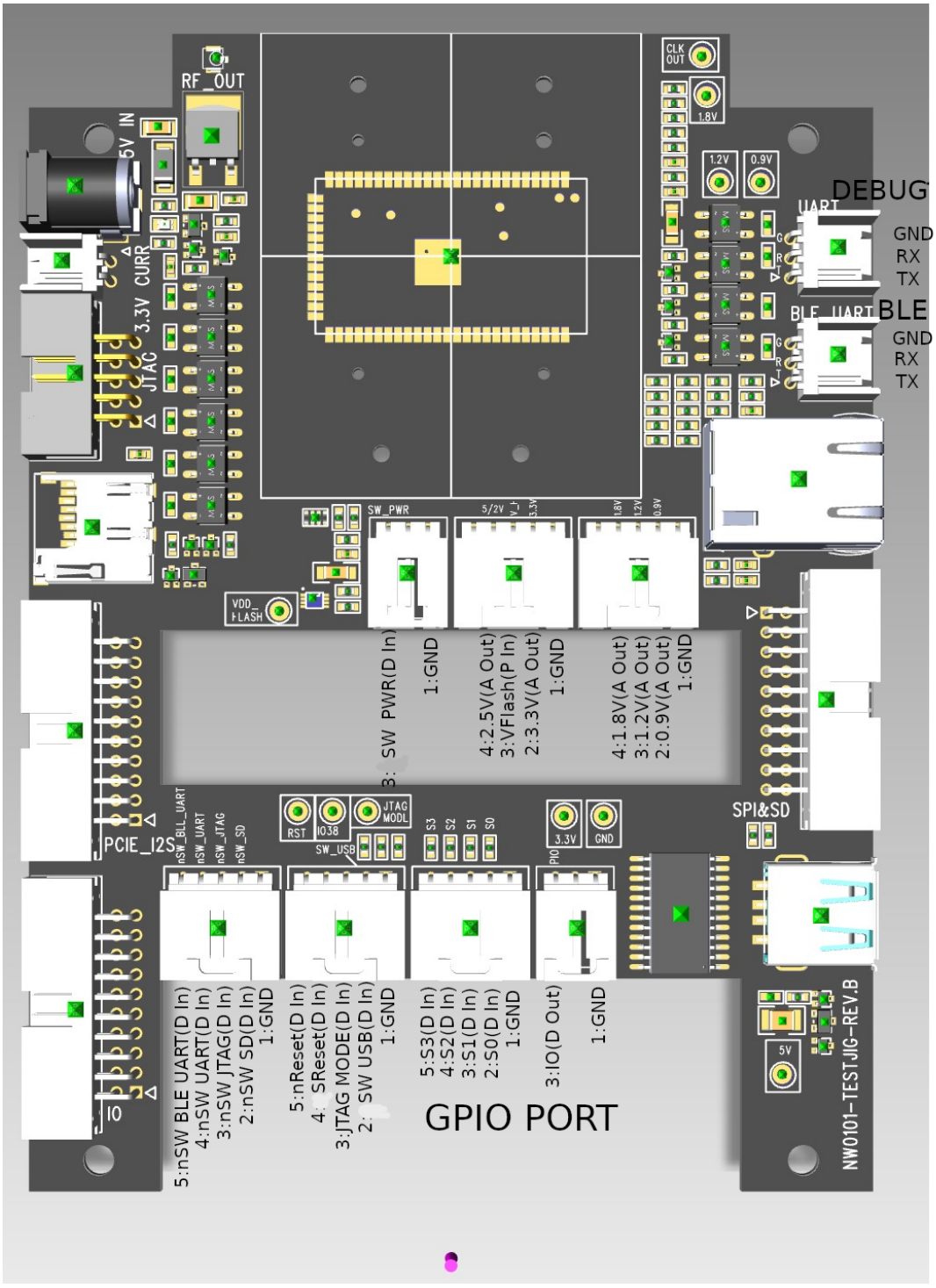
- # Hardware & Pinout

Pinout



Pin description

Switch of solid state relay:

Pin Name	Function	Type	Value	Default
SW PWR	Switch of 5V power	Digital IO Input	0: off, 1: on	1
nSW SD	Switch of SD bus	Digital IO Input	0: on, 1: off	0
nSW JTAG	Switch of JTAG	Digital IO Input	0: on, 1: off	0
nSW UART	Switch of debug UART	Digital IO Input	0: on, 1: off	0
nSW BLE UART	Switch of BLE UART	Digital IO Input	0: on, 1: off	0
SW USB	Switch of 5V USB power	Digital IO Input	0: off, 1: on	1

Strap Pins:

Pin Name	Function	Type	Value	Default
nReset	Hardware reset	Digital IO Input	0: reset	1
SReset(IO38)	Software reset	Digital IO Input	1: reset	floating
JTAG Mode	JTAG Mode/Normal Mode	Digital IO Input	0: JTAG mode, 1: Normal Mode	1

Power:

Pin Name	Function	Type	Value	Default
2.5V	5V Power divider	Analog Output	2.5V	2.5V
3.3V	3.3V Power	Analog Output	3.3V	3.3V
1.8V	1.8V Power	Analog Output	1.8V	1.8V
1.2V	1.2V Power	Analog Output	1.2V	1.2V
0.9V	0.9V Power	Analog Output	0.9V	0.9V
GND	Ground	Analog Output	0V	0V
VFlash	In System Programming Power	**Power Iutput**	In System Programming: 3.3V	0V
			Normal mode: 0V	

GPIO:

Pin Name	Function	Type	Value	Default
S0	Port selection bit0	Digital Input	N/A	N/A
S1	Port selection bit0	Digital Input	N/A	N/A
S2	Port selection bit0	Digital Input	N/A	N/A
S3	Port selection bit0	Digital Input	N/A	N/A
IO	GPIO Output	Digital Output	N/A	N/A

GPIO Pin Table

S3-S0	Pin Number	Pin Name
0000	GPIO0	I2S_SDI
0001	GPIO1	I2S_SDO
0010	GPIO2	I2S_WS
0011	GPIO3	I2S_CLK
0100	GPIO6	SPI_CS1
0101	GPIO11	IO11
0110	GPIO14	IO14
0111	GPIO15	IO15
1000	GPIO16	IO16
1001	GPIO17	IO17
1010	GPIO18	PWM0
1011	GPIO19	PWM1
1100	N/A	RXD1
1101	N/A	N/A
1110	N/A	N/A
1111	N/A	N/A

- # Testing steps

Power Up

nReset = high or floating,

nSReset = low or floating,

JTAG Mode = high or floating,

SW PWR = high or floating,
nSW SD = low or floating,
nSW JTAG = low or floating,
nSW UART = low or floating,
nSW BLE UART = low or floating,
SW USB = high or floating,

VFlash = 0V

Measure 2.5V, 3.3V,1.2V

Program the default image(Only needed when the SPI flash is not programmed before soldering)

nReset = low,

VFlash = 3.3V

Program the entire image file: **O2S_entire_img.bin**

Reboot

nReset = floating,

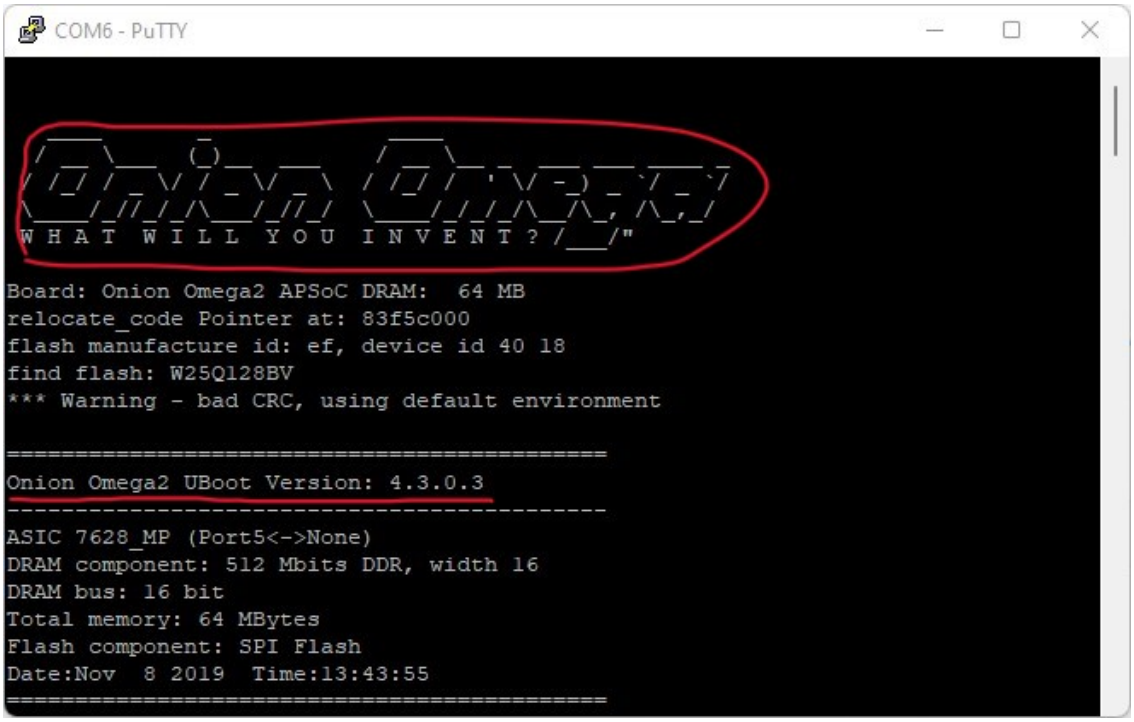
VFlash = 0V

SW PWR = low -> high -> low

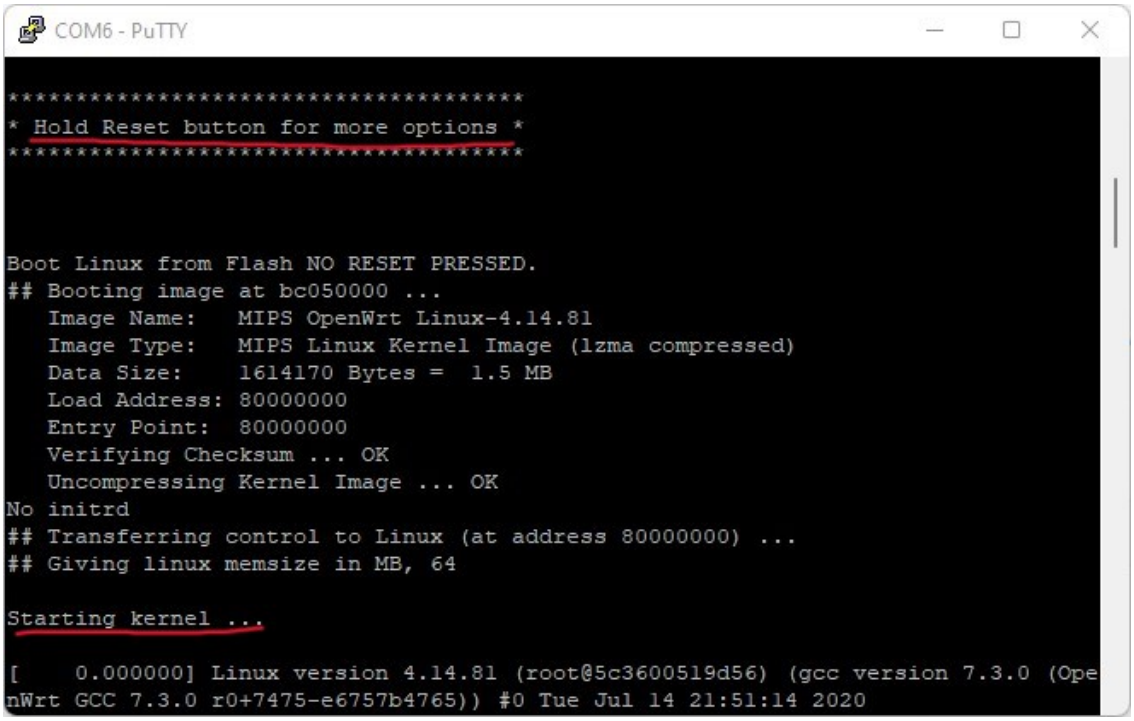
Measure 1.8V, 0.9V

Check on UART(Debug) if OpenWRT boots up, see below:

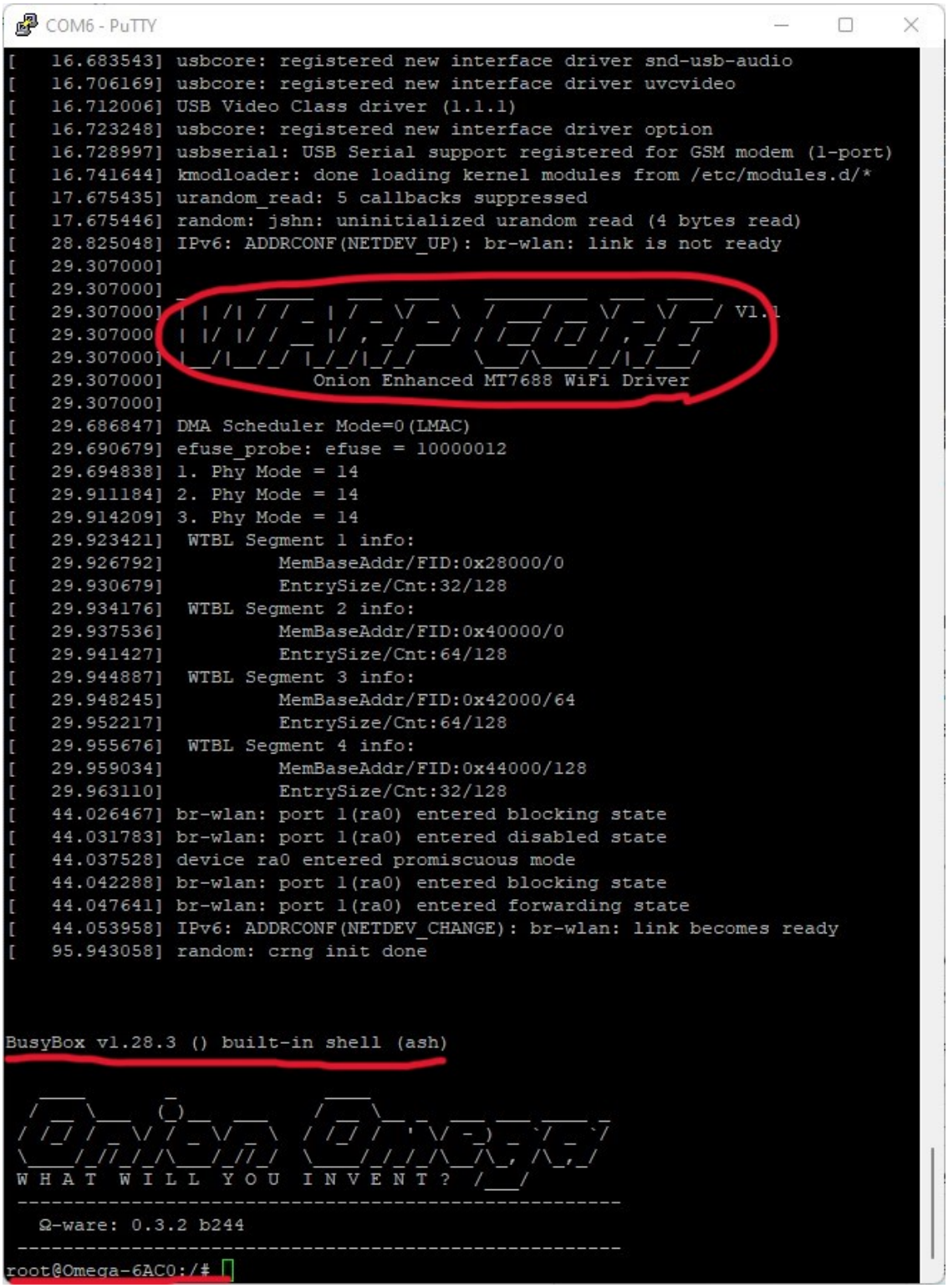
The logo shows the Uboot booted



It shows the kernel booted

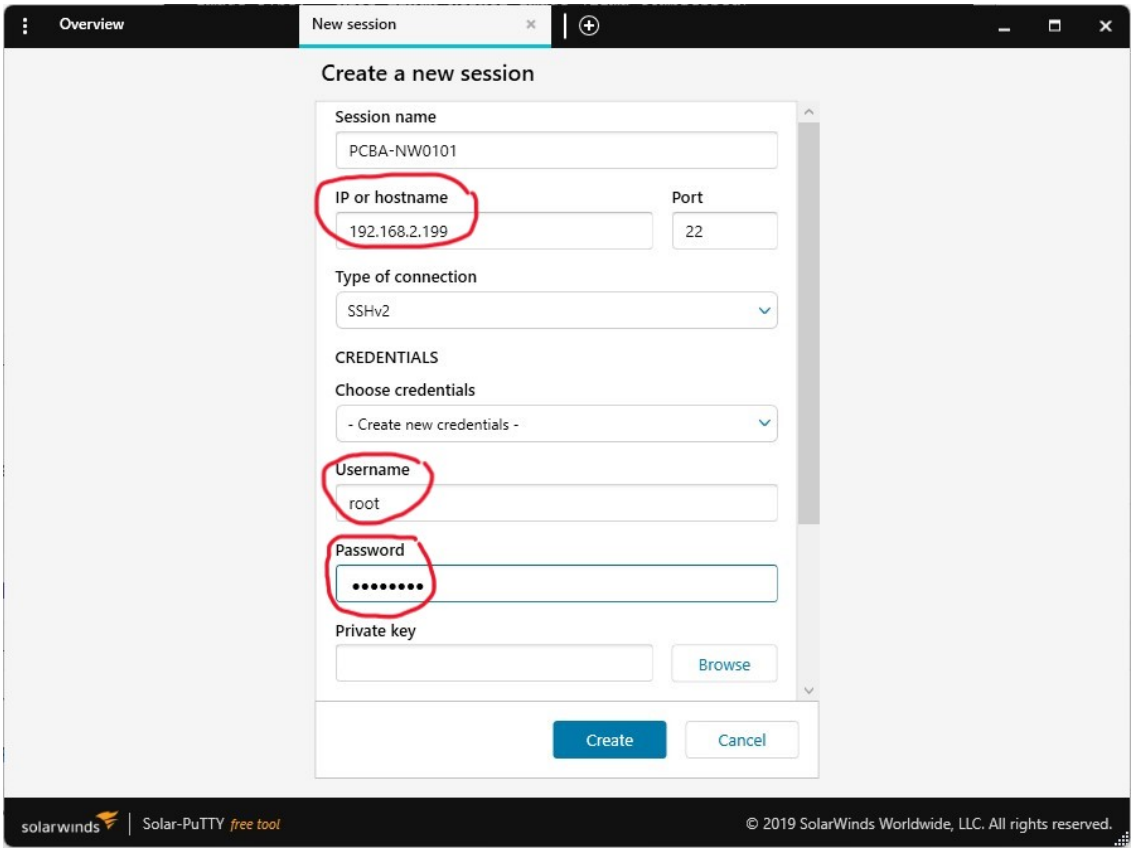


It shows the bootup finished, then press 'enter' to show the bash:

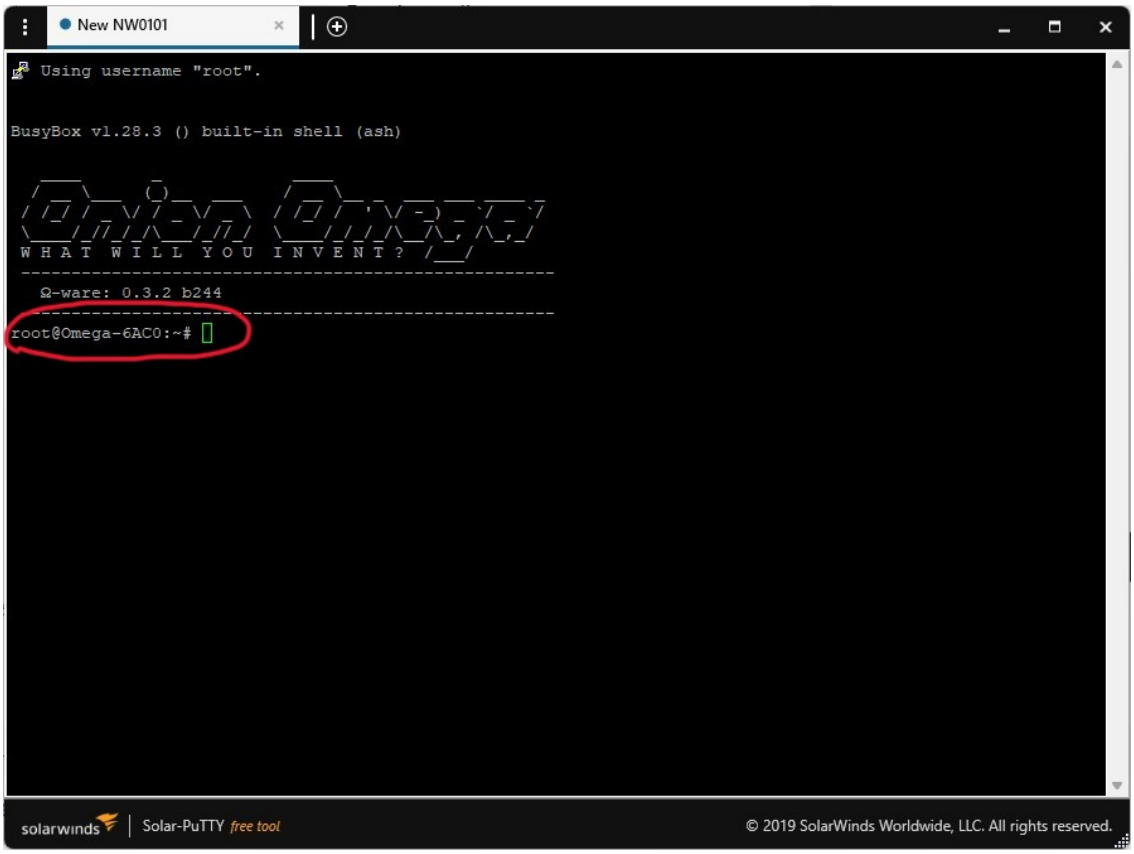


Open Solar Putty SSH client on Windows

Creat a new SSH session(**ip: 192.168.2.199, username: root, password: onioneer**)



open the session that just created:



or use the command below to SSH

```
ssh root@192.168.2.199
```

or SSH 192.168.2.199 in any code

Assign MAC address in OpenWRT

```
# RYSE's MAC address range is from D0-16-F0-20-00-00 to D0-16-F0-2F-FF-FF
# MAC address of WIFI is D0-16-F0-2X-XX-XX
# MAC address of ethernet is D0-16-F0-2X-XX-XX+0x01
# MAC address of apcli0 is D0-16-F0-2X-XX-XX+0x02
dd if=/dev/mtd2 of=/tmp/art-bk.bin
echo -e -n '\xD0\x16\xF0\x20\x00\x00' | dd of=/tmp/art-bk.bin seek=4 bs=1
count=6 conv=notrunc
echo -e -n '\xD0\x16\xF0\x20\x00\x01' | dd of=/tmp/art-bk.bin seek=40 bs=1
count=6 conv=notrunc
echo -e -n '\xD0\x16\xF0\x20\x00\x02' | dd of=/tmp/art-bk.bin seek=46 bs=1
count=6 conv=notrunc
mtd write /tmp/art-bk.bin /dev/mtd2
```

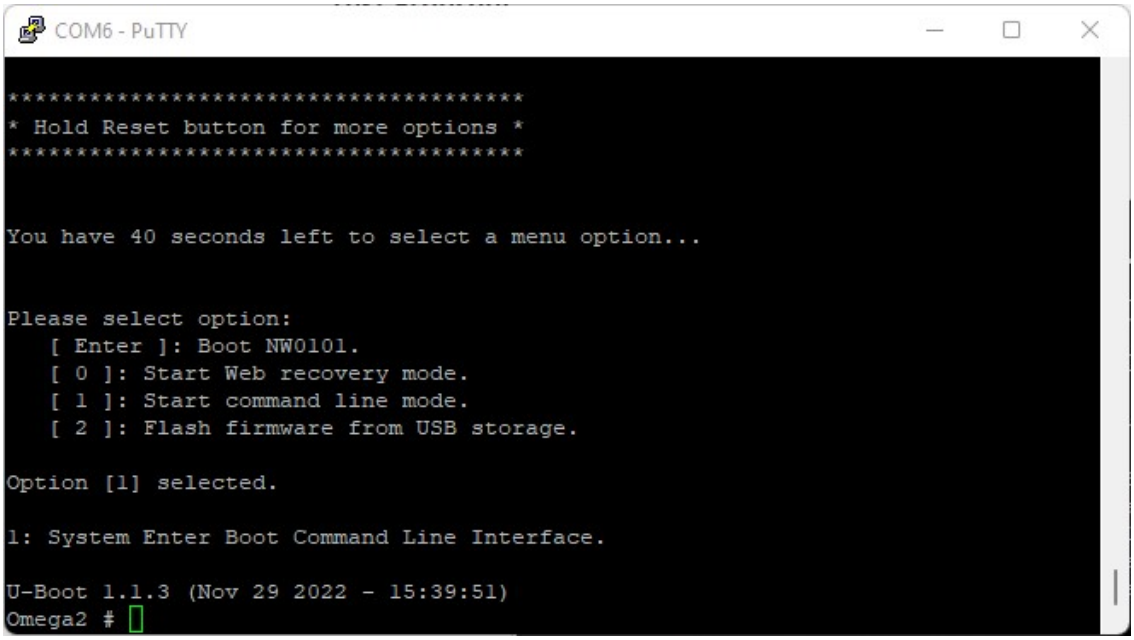
Assign MAC address in Uboot

nSReset = high,

nReset = high or floating,

SW PWR = low and then high

select '1' from the manual



erase the page at 0x40000 in SPI flash

```
spi erase 0x40000 1
```

program data from 0x40000 to 0x40130

```
# replace the MAC address at 0x40004, 0x40028 0x40028
# D016F0200000 is the MAC address D0-16-F0-20-00-00 in the following command
spi write 0x40000 28760002D016F0200000000000000000
# D016F0200001 is the MAC address D0-16-F0-20-00-01 in the following command
# D016F0200002 is the MAC address D0-16-F0-20-00-02 in the following command
spi write 0x40020
00000000200000000D016F0200001D016F020000222340020FFFF000100000000
spi write 0x40040
00002200000000003000000000000000820000944B0C0CA1282818240CA2081808000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000C6C6C4C4C4C0C0C4C4C4C4C0C00000
spi write 0x400b0
00000000000000000000000000000000000000000000000001A222A313501353940464D7F7F
7F00000000000000000111D111D1C351C351E351E351719171902000000C50000880a000000
00000000
spi write 0x40120
00000000000000000000000000007700111D111D157F157F177F177F103B103B
```

Verify the data

```
spi read 0x40000 200
```

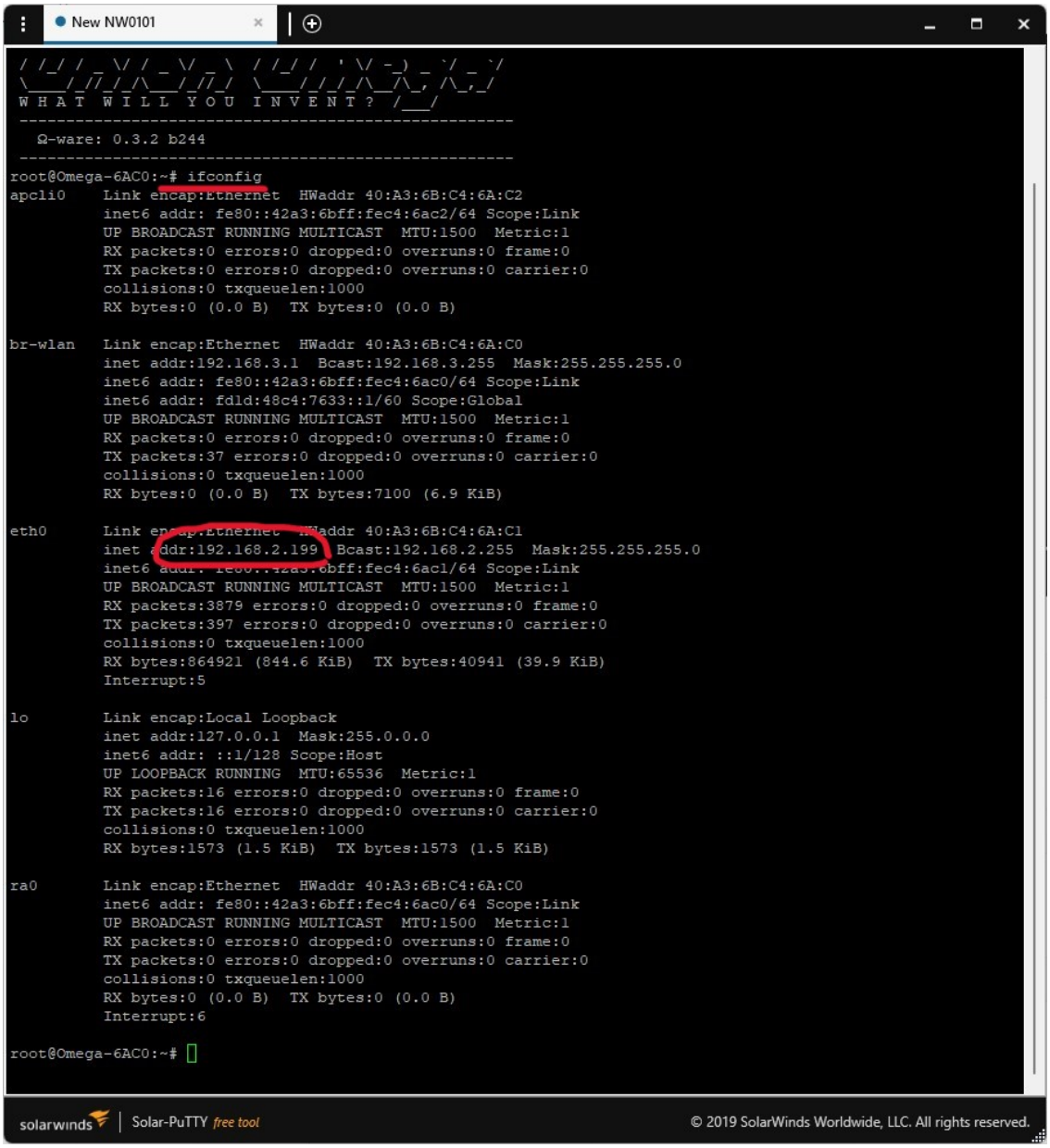
the data should be:

00040000	28	76	00	02	40	A3	6B	C4	6A	C0	00	00	00	00	00	00
00040010	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
00040020	00	00	00	00	20	00	00	00	40	A3	6B	C4	6A	C1	40	A3
00040030	6B	C4	6A	C2	22	34	00	20	FF	FF	00	01	00	00	00	00
00040040	00	00	22	00	00	00	00	00	30	00	00	00	00	00	00	00
00040050	82	00	00	94	40	B0	C0	CA	12	82	81	82	40	CA	20	81
00040060	80	80	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00040070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00040080	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00040090	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000400A0	C6	C6	C4	C4	C4	C0	C0	C4	C4	C4	C4	C4	C0	C0	00	00
000400B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000400C0	00	00	00	00	00	00	00	00	00	1A	22	2A	31	35	01	35
000400D0	39	40	46	4D	7F	7F	7F	00	00	00	00	00	00	00	00	00
000400E0	11	1D	11	1D	1C	35	1C	35	1E	35	1E	35	17	19	17	19
000400F0	02	00	00	00	C5	00	00	88	0A	00	00	00	00	00	00	00
00040100	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
00040110	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
00040120	00	00	00	00	00	00	00	00	00	00	00	00	00	00	77	00
00040130	11	1D	11	1D	15	7F	15	7F	17	7F	17	7F	10	3B	10	3B

Test Ethernet

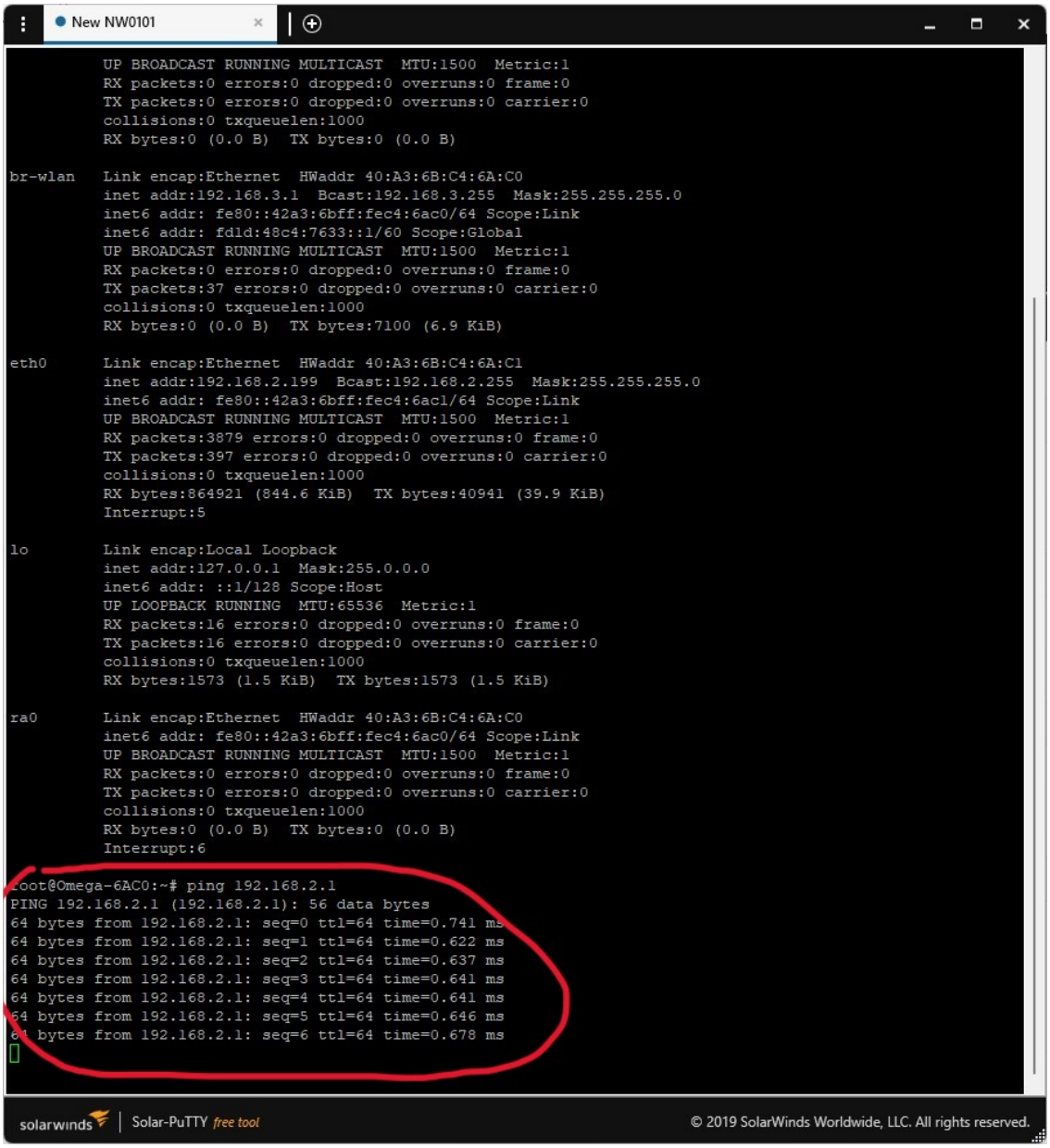
```
ifconfg
```

Check if the ipaddress of eth0, as below:




```
ping 192.168.2.1
```

Check if 192.168.2.1 are reachable, as below

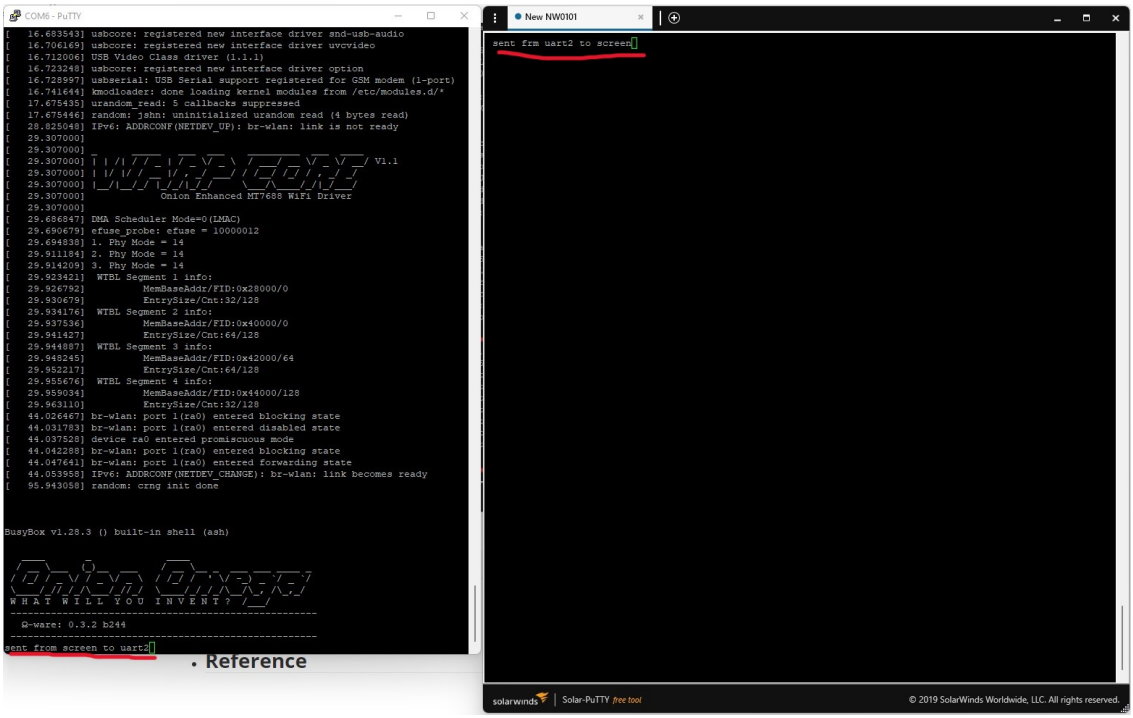


Test BLE UART

Open another UART terminal for BLE UART

```
screen /dev/ttyS2 115200
```

Type strings on both SolarPutty and BLE Uart consol:



on SolarPutty, type 'Ctl'+'a'+'k' and 'y' to quit screen

Test GPIO

Configure the mux of GPIO

```
omega2-ctrl gpiomux set uart1 gpio  # Configure it as GPIO
omega2-ctrl gpiomux set uart1 uart1 # Configure it as UART1
```

uart1 can be any of these group name blow:

```
Group i2c - [i2c] gpio
Group uart0 - [uart] gpio
Group uart1 - [uart] gpio
Group uart2 - [uart] gpio pwm
Group pwm0 - [pwm] gpio
Group pwm1 - [pwm] gpio
Group refclk - refclk [gpio]
Group spi_s - spi_s [gpio]
Group spi_cs1 - [spi_cs1] gpio refclk
Group i2s - i2s [gpio] pcm
Group ephy - [ephy] gpio
Group wled - wled [gpio]
```

Set S3~S0 = xxxx,

```
gpioctl dirout-high 14
gpioctl dirout-low 14   # Take IO14 as example
```

Read IO pin

WIFI test

need to discuss

IIC and MFI Chip test

need to discuss

USB test

need to discuss

SD bus test

need to discuss

- # Reference