

# N58 OpenCPU

## SDK Developer Guide

Issue 1.0 Date 2020-05-14



**Copyright © Neoway Technology Co., Ltd 2020. All rights reserved.**

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Neoway Technology Co., Ltd.

**neoway** is the trademark of Neoway Technology Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

**Notice**

This document provides guide for users to use N58 OpenCPU.

This document is intended for system engineers (SEs), development engineers, and test engineers.

THIS GUIDE PROVIDES INSTRUCTIONS FOR CUSTOMERS TO DESIGN THEIR APPLICATIONS. PLEASE FOLLOW THE RULES AND PARAMETERS IN THIS GUIDE TO DESIGN AND COMMISSION. NEOWAY WILL NOT TAKE ANY RESPONSIBILITY OF BODILY HURT OR ASSET LOSS CAUSED BY IMPROPER OPERATIONS.

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE DUE TO PRODUCT VERSION UPDATE OR OTHER REASONS.

EVERY EFFORT HAS BEEN MADE IN PREPARATION OF THIS DOCUMENT TO ENSURE ACCURACY OF THE CONTENTS, BUT ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS DOCUMENT DO NOT CONSTITUTE A WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

Neoway provides customers complete technical support. If you have any question, please contact your account manager or email to the following email addresses:

Sales@neoway.com

Support@neoway.com

**Website:** <http://www.neoway.com>

# Contents

1 Overview.....	5
2 Installing USB Driver .....	6
3 Setting Up SDK Environments .....	7
3.1 Compilation Environment.....	7
3.2 Development Environment.....	7
3.3 Debugging Environment.....	7
4 Developing Application .....	8
4.1 APIs .....	8
4.2 Adding Log Printing Interface .....	9
4.3 Customizing Configuration File .....	9
4.4 Compiling Program .....	10
4.5 Burning APPIMG .....	11
4.6 Debugging APPIMG .....	14

# About This Document

## Scope

This document is applicable to the N58 OpenCPU series.




## Audience

This document is intended for [system engineers \(SEs\)](#), [development engineers](#), and [test engineers](#).

## Change History

Issue	Date	Change	Changed By
1.0	2020-03	Initial draft	Wu Guoqing

## Conventions

Symbol	Indication
	This warning symbol means danger. You are in a situation that could cause fatal device damage or even bodily damage.
	Means reader be careful. In this situation, you might perform an action that could result in module or product damages.
	Means note or tips for readers to use the module

# 1 Overview

The N58 OpenCPU module runs a FreeRTOS. This document mainly describes the process of developing an application based on N58 OpenCPU SDK, including drivers installation, SDK environment establishment, source code development, APPImg burning, APPImg debugging, etc.

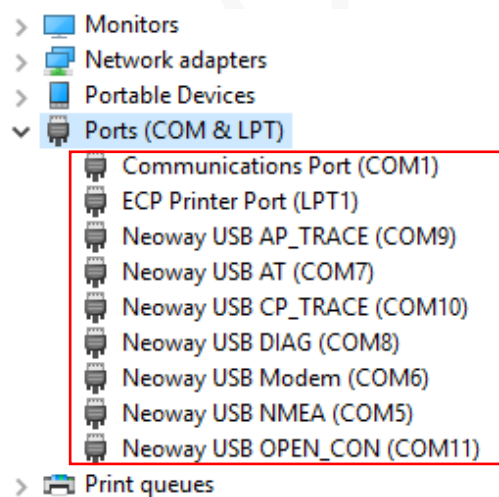
Before the development, you need to have the following knowledge:

- FreeRTOS
- Standard C language
- Inter-task communication and synchronization mechanisms in FreeRTOS
- Network protocols including TCP, FTP, HTTP, etc.
- Mechanisms of controlling the I/O interfaces such as UART, SPI and I2C and some peripherals of the N58 Open CPU module.

## 2 Installing USB Driver

After the USB drivers are installed in Windows OS, Device Manager displays 7 COM ports. The table below lists these ports and their functions.

No.	Port name	Description
1	Neoway USB AP_TRACE	Used to output logs of the applications. The log can be captured through <b>Coolwatch</b> .
2	Neoway USB AT	Used for AT communications
3	Neoway USB CP_TRACE	Used to output logs of the kernel. The logs can be captured through <b>ArmTrace</b> .
4	Neoway USB Diag	Diagnostic port
5	Neoway USB Modem	Modem port, used for PPP dialing
6	Neoway USB Modem	Used to output GPS data (used by the N58 AT module)
7	Neoway USB OPEN_CON	Used to output logs by N58 OpenCPU.



For how to install N58 drivers, see *Neoway\_N58\_Driver\_Installation\_Guide*.



For the details of capturing the AP and CP logs, see *Neoway\_N58\_Log\_Capturing\_Guide*.

## 3 Setting Up SDK Environments

### 3.1 Compilation Environment

- System Requirements
  - Windows 7 SP1, x64 or Windows 10, x64
  - Visual C++ Redistributable for Visual Studio 2015 x86 or later versions
  - Python 3 or later versions
- Cross Compiler
  - Stored in the **/tool/** directory of the SDK package.



For the environment installation packages of Windows OS, contact Neoway FAE.

The name of the directory that stores the SDK package should contain letters, digits, or underscores only. Otherwise, errors will occur in SDK compilation.

---

### 3.2 Development Environment

- System requirements
  - Windows 7/8/10
- Programming editor
  - Source Insight (other options are also available)

### 3.3 Debugging Environment

- System requirements
  - Windows 7/8/10
- Debugging tool
  - PuTTY or other common debugging tools
  - The logs are output from the **Neoway USB OPEN\_CON**.

## 4 Developing Application

To facilitate the development of application images in N58 OpenCPU, Neoway provides various SDK APIs. For details, see *Neoway\_N58 OpenCPU\_API\_Note*.

### 4.1 APIs

Category	Function
IO interfaces	To enable and invoke IO interfaces, including <ul style="list-style-type: none"> <li>• UART</li> <li>• I2C</li> <li>• SPI</li> <li>• GPIO</li> <li>• ADC</li> </ul>
Peripheral	To enable and invoke peripheral interfaces, including <ul style="list-style-type: none"> <li>• Audio</li> <li>• LED</li> <li>• Keypad</li> </ul>
Service	To enable and invoke network service interfaces, including <ul style="list-style-type: none"> <li>• Data</li> <li>• Voice</li> <li>• SIM</li> <li>• SMS</li> <li>• Location</li> <li>• Wi-Fi Scan (only STA is supported, used to assist in indoor positioning)</li> <li>• Network (network registration)</li> <li>• FOTA (delta upgrade)</li> <li>• AT (to send and receive AT commands)</li> <li>• Socket</li> </ul>
Protocol (POSIX)*	<ul style="list-style-type: none"> <li>• TCP</li> <li>• FTP</li> <li>• HTTP</li> <li>• UDP</li> </ul> For details, learn from the internet or refer to the demo programs provided by Neoway.
Other interfaces*	<ul style="list-style-type: none"> <li>• FreeRTOS queue</li> <li>• Timer</li> </ul>



\* indicates in development.

## 4.2 Adding Log Printing Interface

To facilitate the module commissioning, Neoway provides the API that is used to print logs. The log can be output to the USB port by calling the **nwy\_ext\_echo()** function.

### Header File

```
#include "nwy_usb_serial.h"
```

### Function

```
void nwy_ext_echo(char* fmt, ...)
{
    static char    echo_str[NWY_EXT_SIO_RX_MAX];
    va_list      a;
    va_start(a, fmt);
    vsprintf(echo_str, fmt, a);
    va_end(a);
    nwy_usb_serial_send(echo_str, strlen((char *)echo_str));
}
```

### Log Capturing Methods

Output to	Description
USB port (recommended)	Use a serial port tool such as Putty to view log messages Call <b>nwy_ext_echo()</b> that is provided by Neoway.
UART port	Use <b>Coolwatcher</b> to view log messages of an application. Call the OSI_LOGI API that is provided by UNISOC

## 4.3 Customizing Configuration File

CMakeLists.txt is the input to the CMake build system for building software packages and it contains a set of directives and instructions describing the source files and target files (executable, library, or both) of the project. To compile the program, edit the file as follows:

### Adding File Name of Source Code into CMake Configuration File

Assume that the name of the source code file to be compiled is **Neoway\_test.c** (saved in the SDK home directory), make some modifications as shown below in the **CMakeLists.txt** file.

```
if(CONFIG_APPIMG_LOAD_FLASH)
    set(target Neoway_test_flash)
    add_appimg(${target} ${flash_ldscript} Neoway_test.c)
...
if(CONFIG_APPIMG_LOAD_FILE)
    set(target Neoway_test_file)
    add_appimg(${target} ${file_ldscript} Neoway_test.c)
...
```

If **Neoway\_test.c** needs to call a function of other source files such as **Neoway\_other.c**. Make the following modifications.

- Add the related header file in **\components\include**.
- Modify **CMakeLists.txt**

```
if(CONFIG_APPIMG_LOAD_FLASH)
    set(target Neoway_test_flash)
    add_appimg(${target} ${flash_ldscript} Neoway_test.c Neoway_other.c)
...
if(CONFIG_APPIMG_LOAD_FILE)
    set(target Neoway_test_file)
    add_appimg(${target} ${file_ldscript} Neoway_test.c Neoway_other.c)
...
```












### Modifying the Directory of APPIMG

The generated image file **hello\_flash.pac** is stored in **out\lappimage\_release\hex** by default. You can modify the content shown below in the **CMakeLists.txt** file to change the directory.

```
set PROJECT_OUT=%PROJECT_ROOT%\out\%BUILD_TARGET%_%BUILD_RELEASE_TYPE%
```

## 4.4 Compiling Program

The home directory of N58 Open SDK contains the **nwy\_opencpu.bat** compilation script. If the source code is saved in the home directory, double-click the compilation script directly to generate the image file in **out\lappimage\_release\hex**.

 cmake	2020/3/9 ...	文件夹	
 components	2020/3/9 ...	文件夹	
 ldscripts	2020/3/9 ...	文件夹	
 out	2020/3/2...	文件夹	
 prebuilts	2020/3/9 ...	文件夹	
 tools	2020/3/9 ...	文件夹	
 .clang-format	2020/2/2...	CLANG-FORMAT...	1 KB
 .gitignore	2020/2/1...	GITIGNORE 文件	1 KB
 CMakeLists.txt	2020/3/2...	文本文档	7 KB
 Neoway_test.c	2020/3/4 ...	C Source file	82 KB
 nwy_opencpu.bat	2020/3/9 ...	Windows 批处理...	1 KB

## 4.5 Burning APPIMG

After generating an APPIMG, you can burn it into the N58 module through the **RESEARCHDOWNLOAD** tool.



Before burning APPIMG, ensure that the N58 module is embedded with the underlying firmware of N58 OpenCPU. Otherwise, the burning fails.

- Underlying firmware: the file system that is used to run programs, about 6M.
- APPImg: Generated in the SDK compilation, 100K

For the underlying firmware version of N58 OpenCPU, contact Neoway FAE.

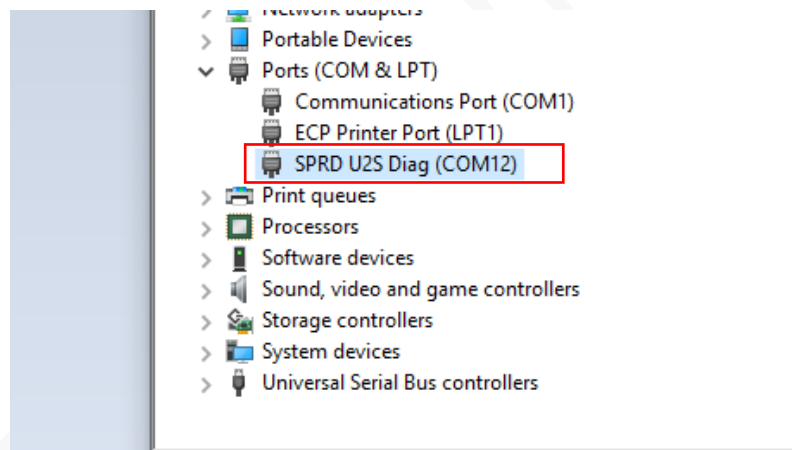
**Step 1:** Set the module to download mode through the following methods:

- Hardware

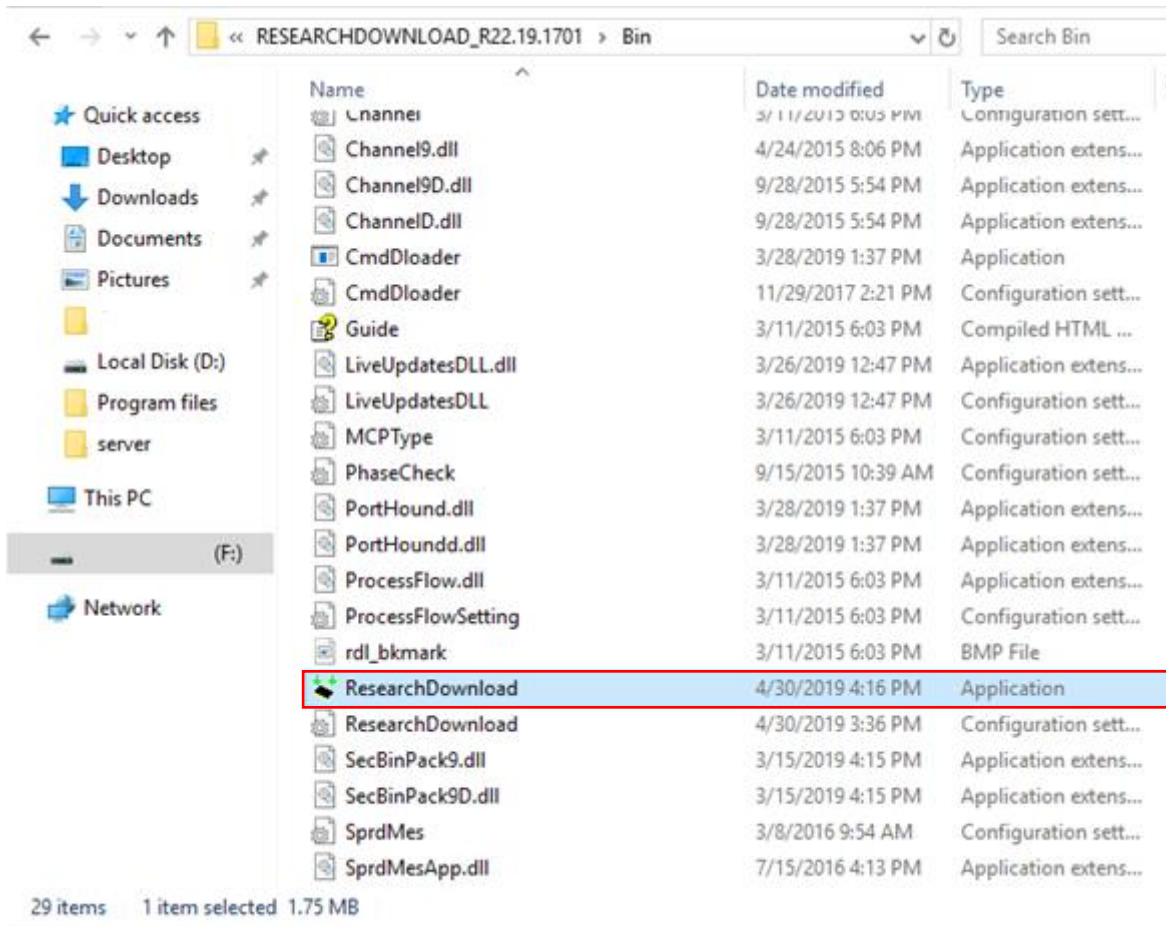
Short connect PIN\_48 to the PIN\_45. Then power the module and start it.

- AT command

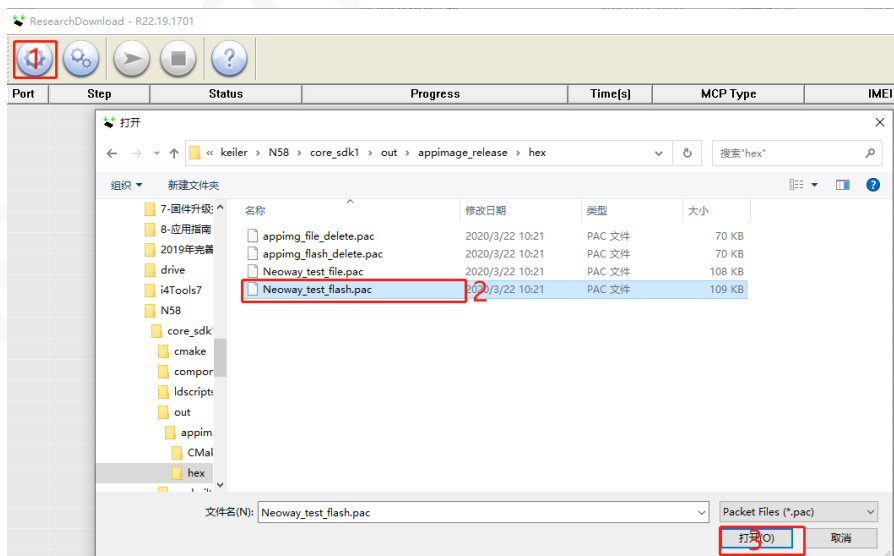
Send **AT^FORCEDNLD** or **AT+CFUN=9,0** to the module through the serial port tool.



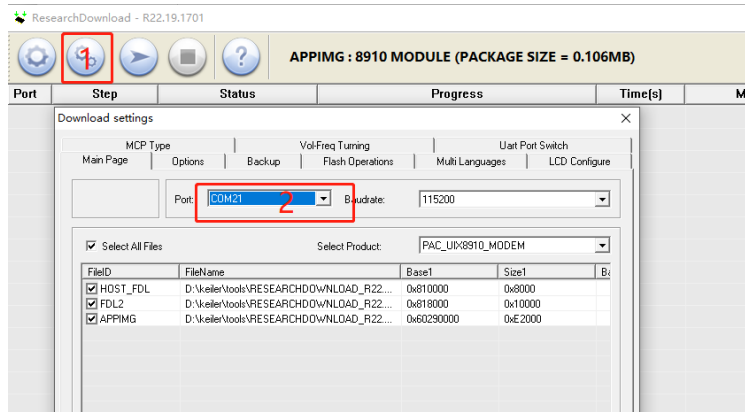
**Step 2:** Start the **RESEARCHDOWNLOAD** tool as an administrator.



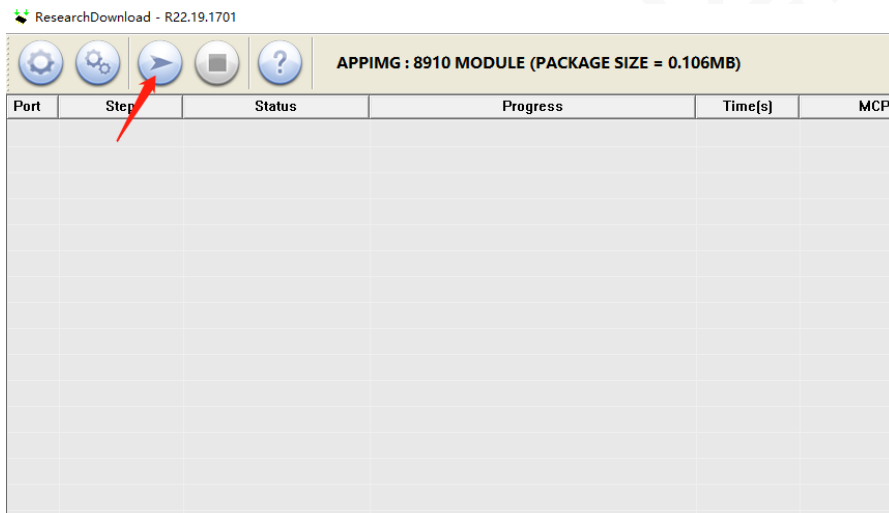
**Step 3:** Click the **Load packet** button and then select the **.pac** file. For example, **Neoway\_test\_flash.pac**.



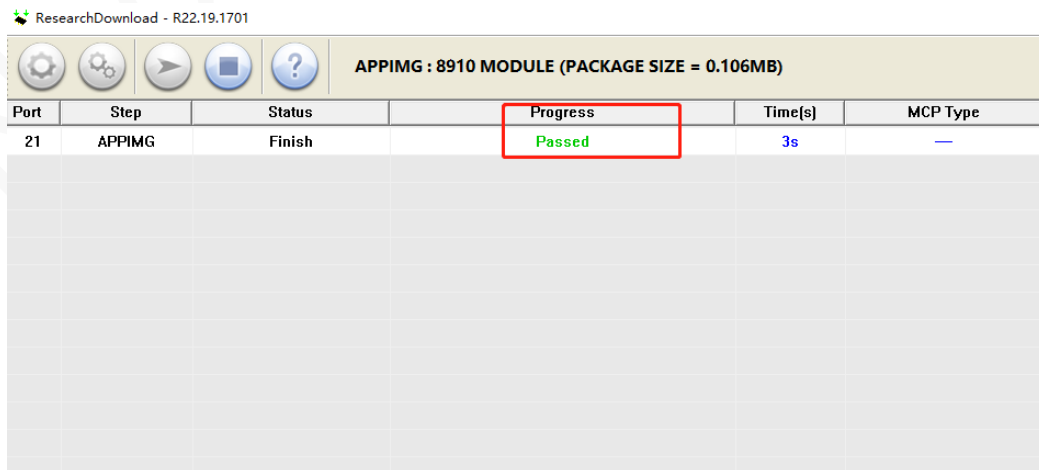
**Step 4:** Click the **Setting** button to select a download port.



**Step 5:** Click the **Start downloading** button to flash the APPImg.



Wait for a while, the tool interface displays **Passed** in green, indicating that the APPIMG is flashed into the module successfully. Cut off power supply of the module and power on it again, and the APPIMG runs automatically when the module starts.



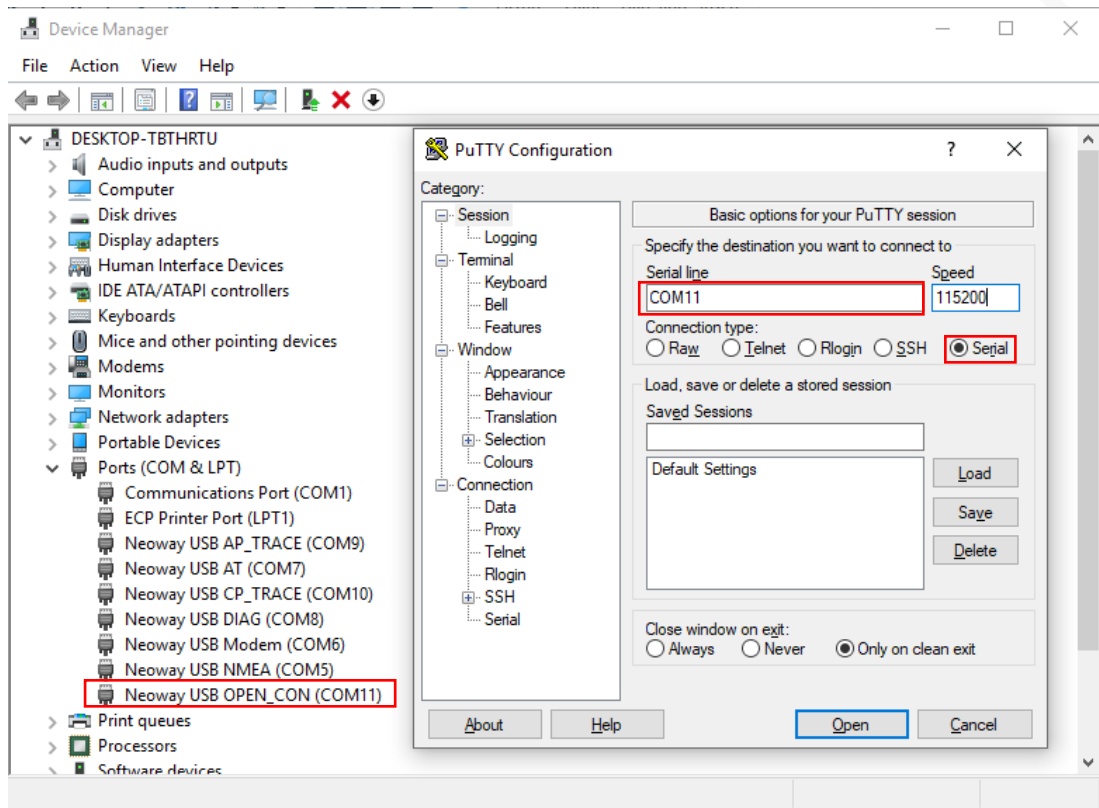
## 4.6 Debugging APPIMG

To print logs to the Neoway USB OPEN\_CON USB port, perform the following steps:

**Step 1:** Start PuTTY.

**Step 2:** Check **Serial** to set its connection type.

**Step 3:** In the **PuTTY Configuration** dialog box, enter the USB port number and then click **Open**.



**Step 4:** Debug the APPIMG.

The **Neoway USB OPEN\_CON** port outputs debugging logs.

