Ai-WB2 series linux build environment tutorial

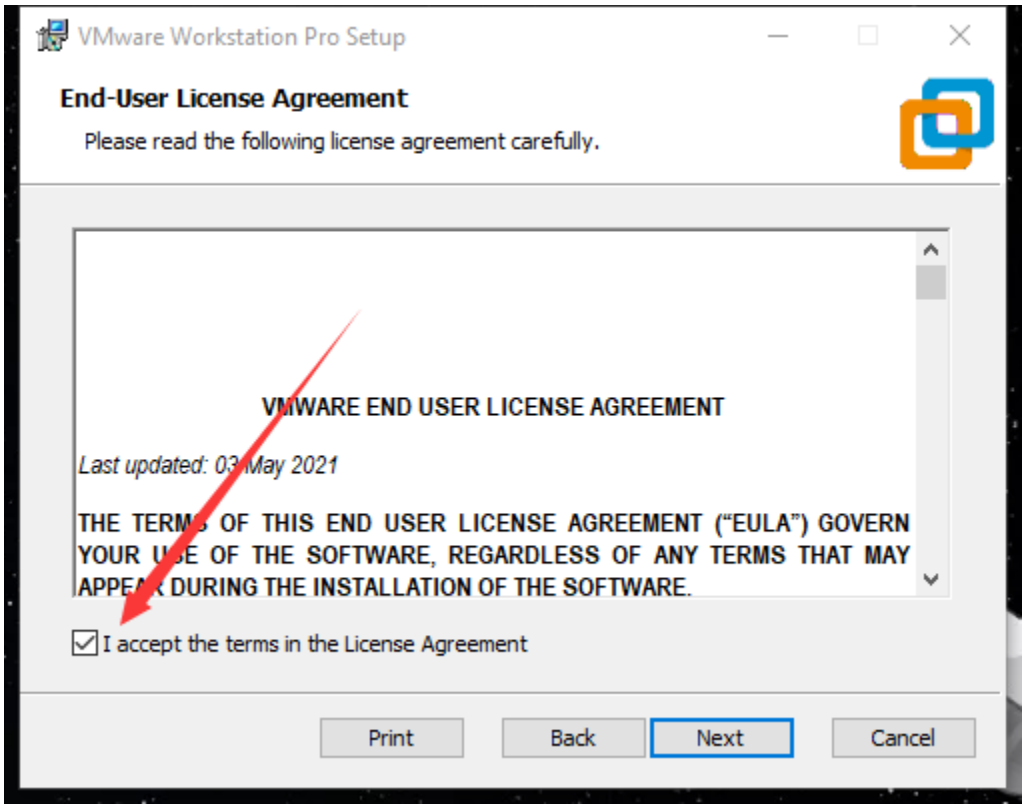
Content

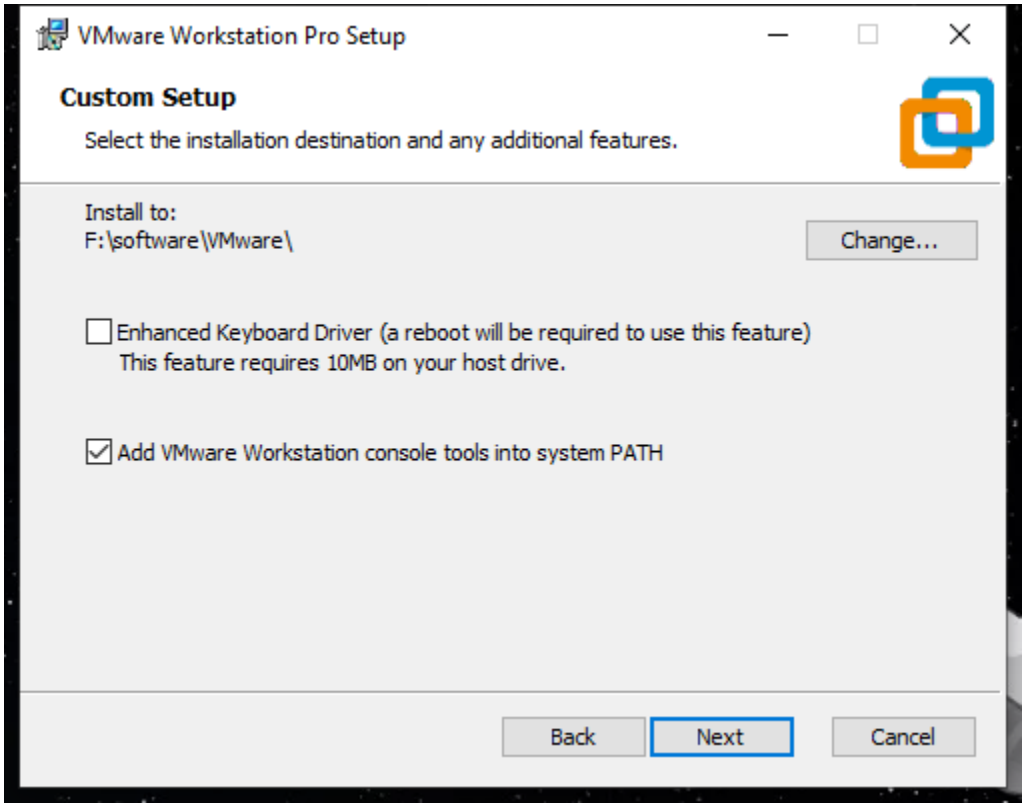
1. Ubuntu installation

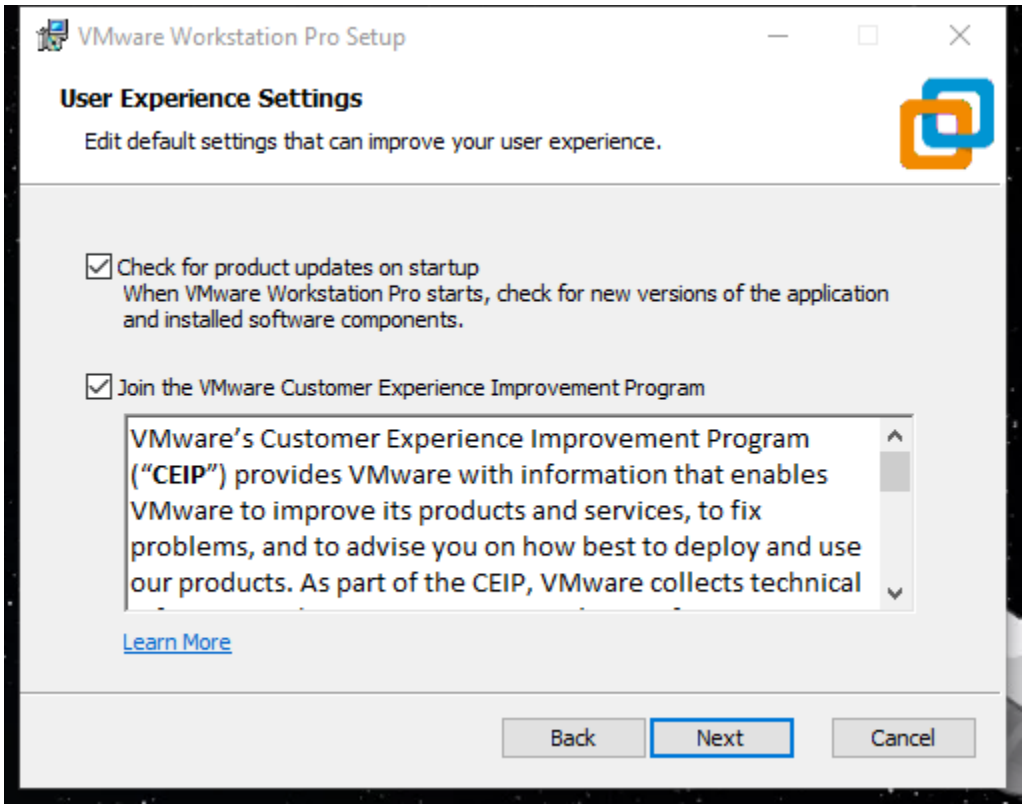
The compilation speed in Linux is much faster than that in windows. We recommend that you use linux for development first. Ubuntu is a common Linux operating system. This topic describes how to quickly build an Ubuntu virtual machine.

1.1 Install VMware Workstation Pro

Download action VMware Workstation Pro 16
Installation


VMware Workstation Pro Setup
End-User License Agreement
Please read the following license agreement carefully.
VMWARE END USER LICENSE AGREEMENT
Last updated: 03 May 2021
THE TERMS OF THIS END USER LICENSE AGREEMENT ("EULA") GOVERN YOUR USE OF THE SOFTWARE, REGARDLESS OF ANY TERMS THAT MAY APPEAR DURING THE INSTALLATION OF THE SOFTWARE.
I accept the terms in the License Agreement
Print
Back
Next
Cancel


VMware Workstation Pro Setup
Custom Setup
Select the installation destination and any additional features.
Install to:
F:\software\VMware\
Change...
Enhanced Keyboard Driver (a reboot will be required to use this feature)
This feature requires 10MB on your host drive.
Add VMware Workstation console tools into system PATH
Back
Next
Cancel



Click next until the installation is complete.

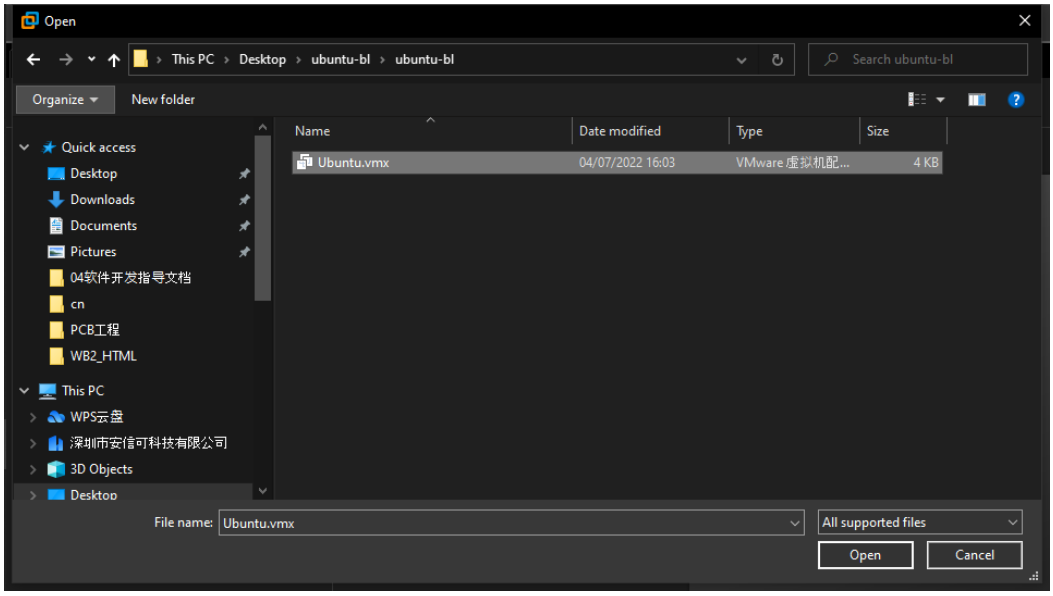
Activate:
open VMware Workstation Pro> Help> enter the license key.

1.2 Install Ubuntu

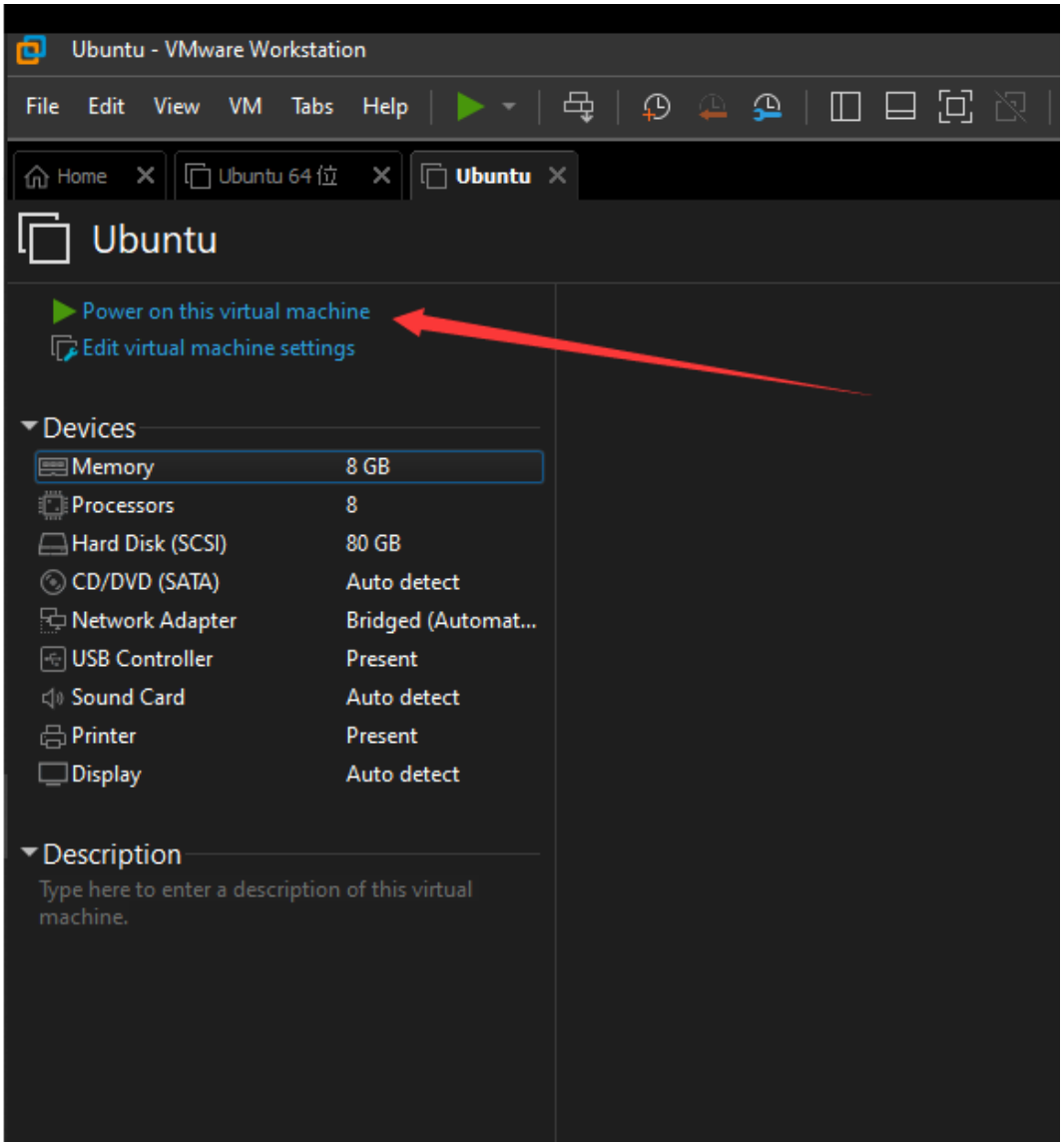
Directly download the installed Ubuntu20.04 link:
https://drive.google.com/drive/folders/1uP517y4QLhiECbYwFXdvGdFYzqp5AbbR?usp=sharing

decompress the download.

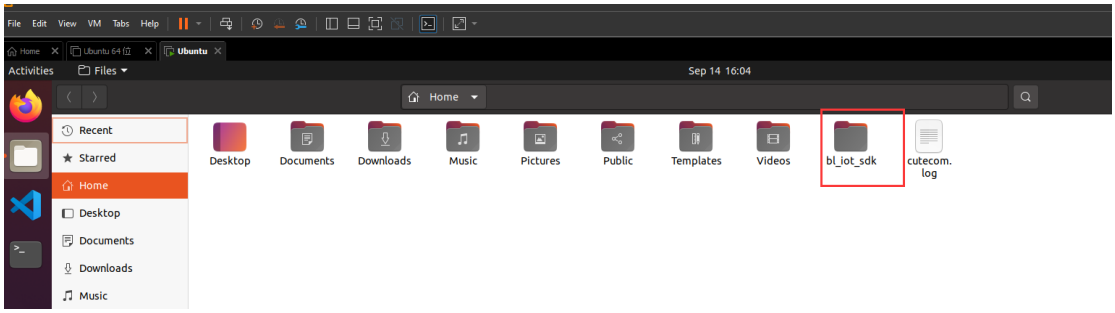
Open the VMware Workstation Pro -> File-> Open-> select the path of the downloaded and decompressed Ubuntu file



Boot directly, the user name and password are axk



the Ubuntu virtual machine has installed the development environment of Ai-WB2 series modules. The sdk is located in the home directory-> bl_iot_sdk.



The following brackets are for users who have installed Ubuntu before reading this blog post

{

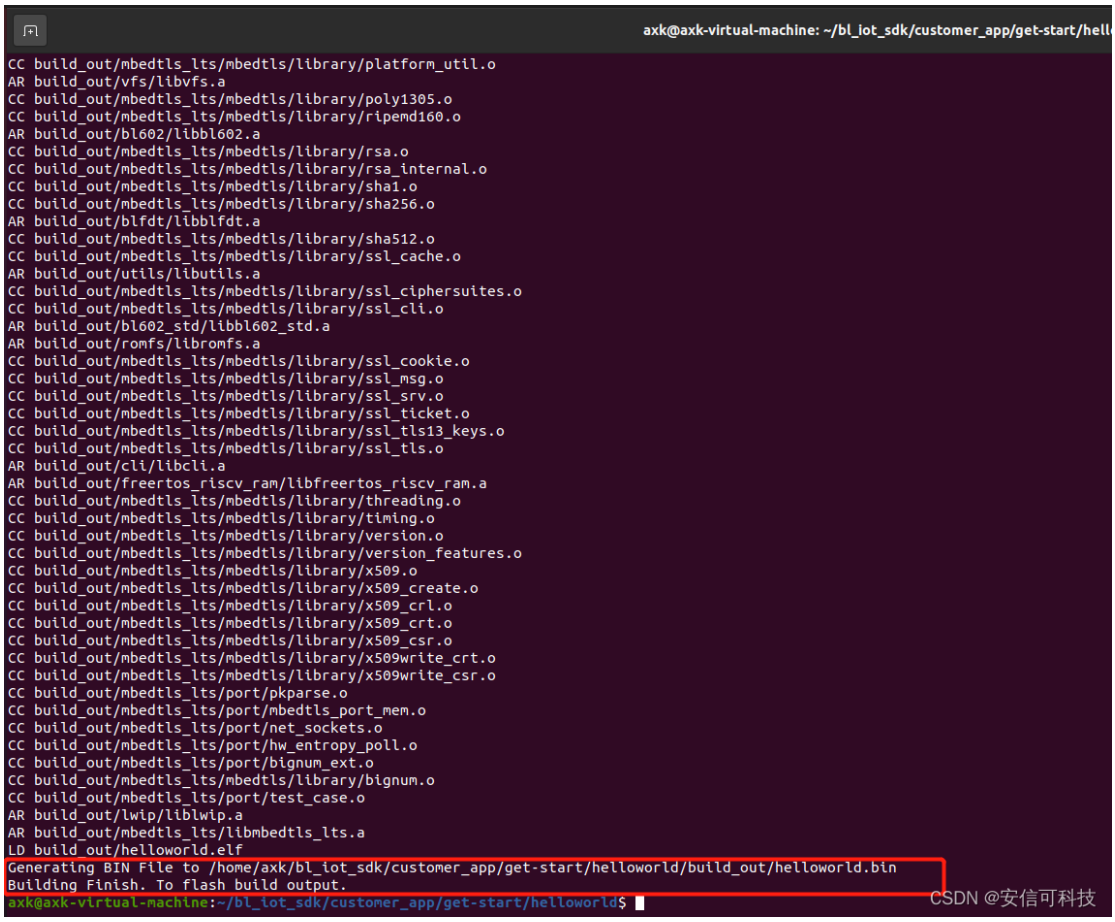
1. No need to reinstall Ubuntu
2. SDK: git clone https://github.com/bouffalolab/bl_iot_sdk.git
3. Open the terminal and install make. Run the sudo apt-get update and sudo apt-get install make commands.
4. install the serial port tool. Use cutecom here. Run the sudo apt-get install cutecom command.
5. Add a USER to dialout in Linux and run the sudo usermod -a -G dialout $USER command.

}

2. Compilation

Enter the Directory of the project to be compiled. For example, to enter the hello world project, run the command: **cd customer_app/get-start/helloworld** to compile the project. Run the command **./genromap**

The following log indicates that the data has been compiled.



The compiled firmware is in the build_out folder: helloworld.bin

3. Install firmware to the device

3.1 Method 1: Use command line to burn

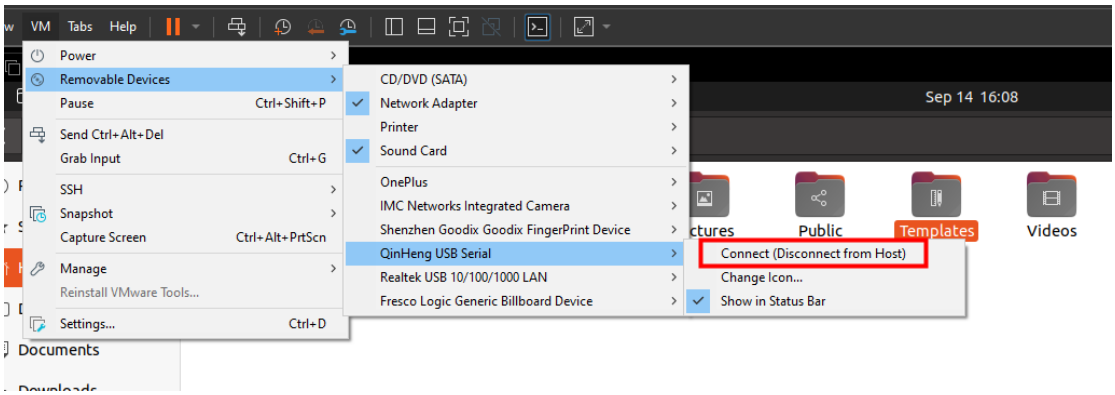
- Module wiring

Ai-WB2-XX	USB to TTL
TX	RX
RX	TX
IO8	DTR
EN	RTS
VDD	3.3V
GND	GND

Note that IO8 is on the dot on the back of the module.

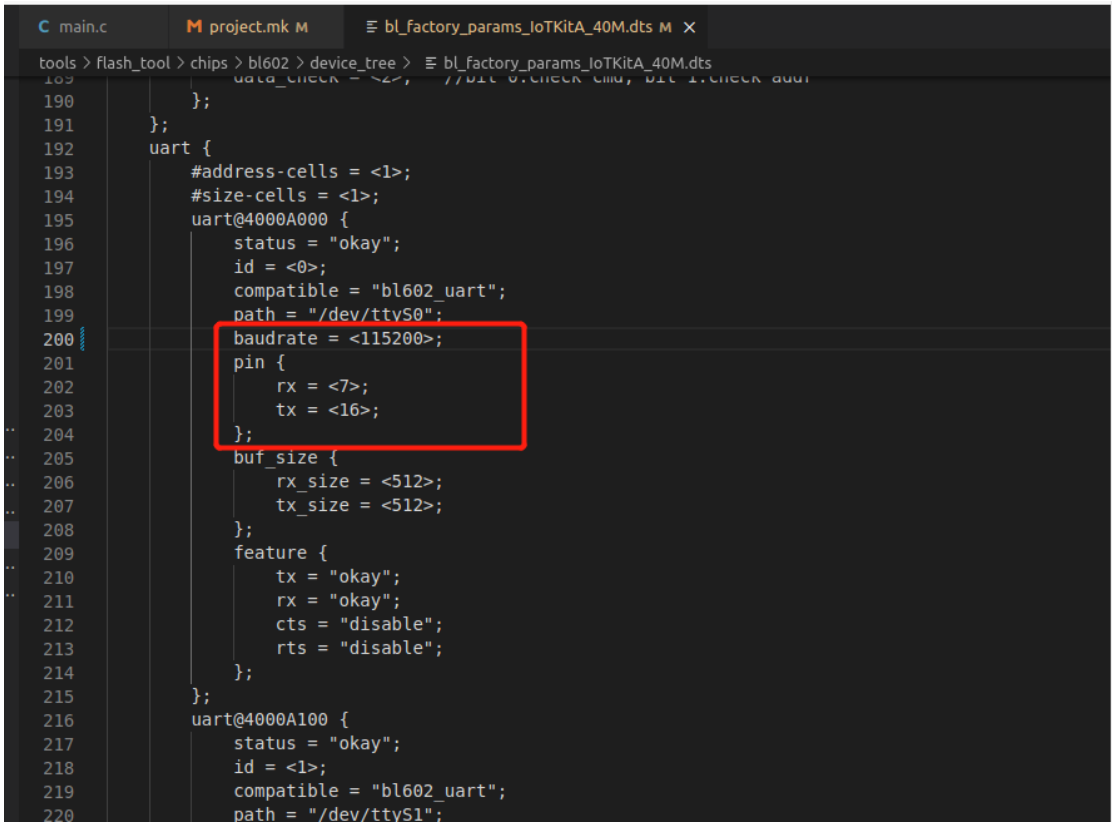
- The development board enters the programming mode:
  connect USB to the computer, press the BURN and EN buttons at the same time, release the EN button first and then release the BURN button

Connect a device to a virtual machine



set the baud rate of the serial port log:

open tools/flash_tool/chips/bl602/device_tree/... and set it in uart{ }. Here, set it to 115200



Run the following command:

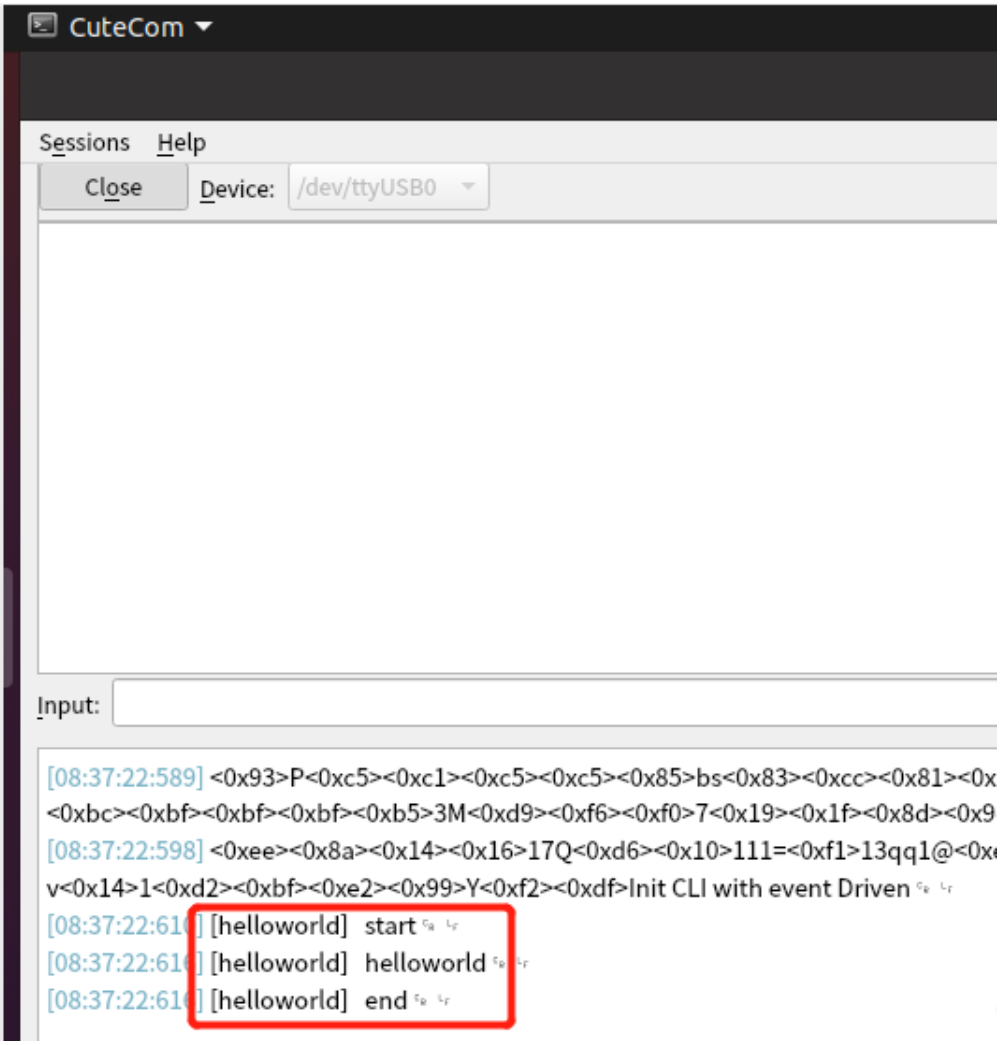
make flash_only

The following log indicates that the recording is successful.

```
[20:36:10.619] - Load 22528/46616 {"progress":48}
[20:36:10.636] - Load 24576/46616 {"progress":52}
[20:36:10.652] - Load 26624/46616 {"progress":57}
[20:36:10.668] - Load 28672/46616 {"progress":61}
[20:36:10.685] - Load 30720/46616 {"progress":65}
[20:36:10.702] - Load 32768/46616 {"progress":70}
[20:36:10.719] - Load 34816/46616 {"progress":74}
[20:36:10.735] - Load 36864/46616 {"progress":79}
[20:36:10.751] - Load 38912/46616 {"progress":83}
[20:36:10.769] - Load 40960/46616 {"progress":87}
[20:36:10.786] - Load 43008/46616 {"progress":92}
[20:36:10.804] - Load 45056/46616 {"progress":96}
[20:36:10.820] - Load 46616/46616 {"progress":100}
[20:36:10.820] - Load 46616/46616 {"progress":100}
[20:36:10.820] - Write check
[20:36:10.845] - Flash load time cost(ms): 427.244384765625
[20:36:10.845] - Finished
[20:36:10.846] - Sha caled by host: eddfb6c88ef11d97cdfd1bd777e6623650741955a77176
[20:36:10.846] - xip mode Verify
[20:36:10.874] - Read Sha256/86048
[20:36:10.874] - Flash xip readsha time cost(ms): 26.012451171875
[20:36:10.874] - Finished
[20:36:10.877] - Sha caled by dev: eddfb6c88ef11d97cdfd1bd777e6623650741955a77176b
[20:36:10.877] - Verify success
[20:36:10.877] - Dealing Index 4
[20:36:10.877] - ========= programming chips/bl602/device_tree/ro_params.dtb to 0x
[20:36:10.878] - ========= flash load =========
[20:36:10.879] - ========= flash erase =========
[20:36:10.879] - Erase flash  from 0x1f8000 to 0x1f965d
[20:36:10.881] - erase pending
[20:36:10.909] - erase pending
[20:36:10.958] - Erase time cost(ms): 79.621337890625
[20:36:10.960] - decompress flash load 1508
[20:36:10.973] - Load 1508/1508 {"progress":100}
[20:36:10.973] - Load 1508/1508 {"progress":100}
[20:36:10.973] - Write check
[20:36:10.993] - Flash load time cost(ms): 34.4931640625
[20:36:10.993] - Finished
[20:36:10.993] - Sha caled by host: 465321030a14218284175e8fc4c6ed55a6486e784c5843
[20:36:10.993] - xip mode Verify
[20:36:11.000] - Read Sha256/5726
[20:36:11.000] - Flash xip readsha time cost(ms): 4.412109375
[20:36:11.000] - Finished
[20:36:11.003] - Sha caled by dev: 465321030a14218284175e8fc4c6ed55a6486e784c58433
[20:36:11.003] - Verify success
[20:36:11.003] - Program Finished
[20:36:11.003] - All time cost(ms): 4841.936767578125
[20:36:11.105] - [All Success]
axk@axk-virtual-machine:~/bl_iot_sdk/customer_app/get-start/helloworld$
```

Open the serial port assistant after burning: sudo cutecom
open the serial port after setting the baud rate in cutecom

print helloworld after resetting the module



3.2 Method 2: Use visual software to install (windows)

Download software: https://docs.ai-thinker.com/_media/ai-wb2/docs/v1.7.4-release.zip

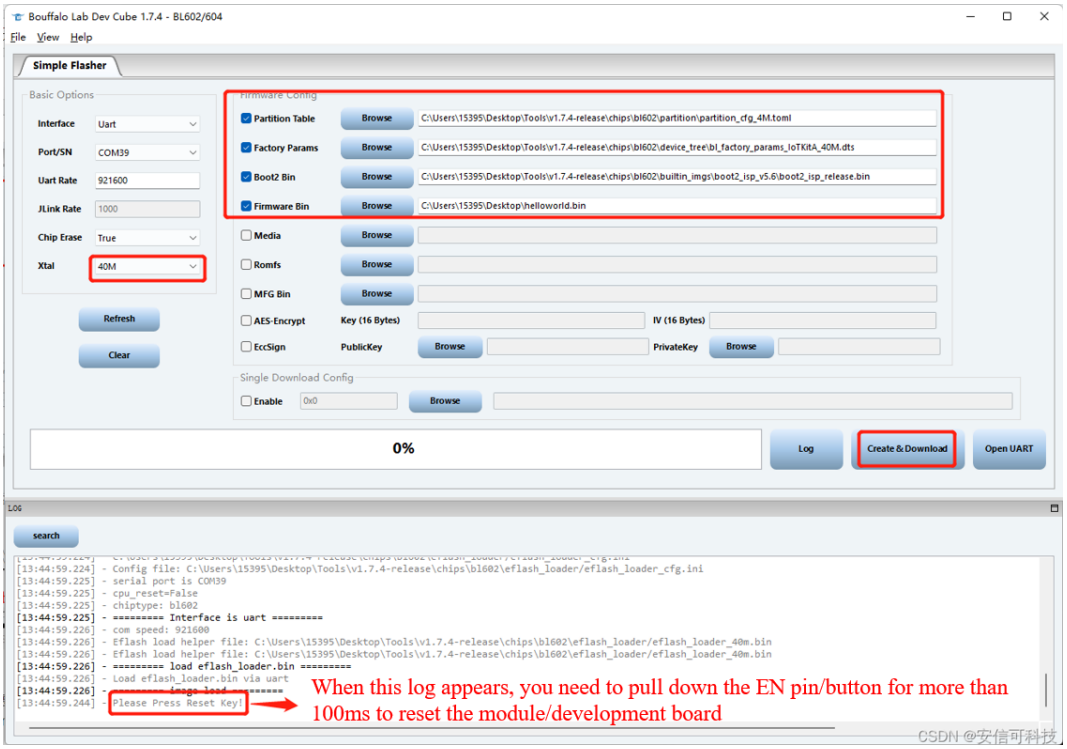
- Module wiring:

Ai-WB2-XX	USB to TTL
TX	RX
RX	TX
EN	RTS
VDD	3.3V
GND	GND

The serial port chips that have been verified and supported are FT232 and CH340.

Development Board wiring:
connect the usb directly to the computer.
The Parttion Table, Factory Params, and Boot2 bins to be burned are located in the following sections:
v1.7.4-release\chips\bl602\partition\partition_cfg_4M.toml
v1.7.4-release\chips\bl602\device_tree\bl_factory_params_IoTKitA_40M.dts
v1.7.4-release\chips\bl602\builtin_imgs\boot2_isp_v5.6\boot2_isp_release.bin

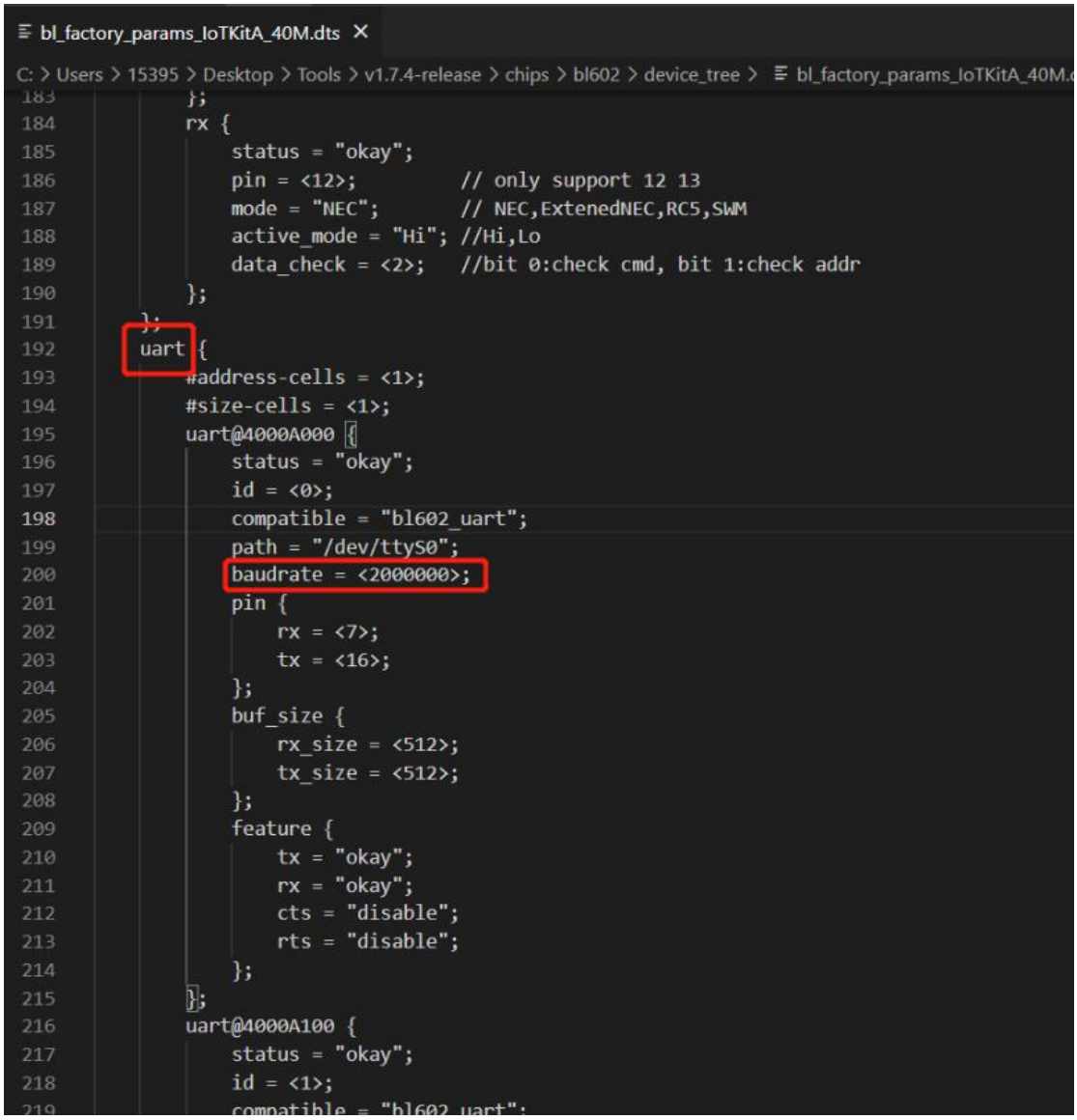
Compile the generated application-layer Firmware to Firmware Bin
Note: do not place the programming tools and firmware in the Chinese path.



The default baud rate of the log serial port is 2000000.

If you need to modify the baud rate of the log serial port, you can modify it in the file

v1.7.4-release\chips\bl602\device_tree\bl_factory_params_IoTKitA_40M.dts and download it again after modification.



3. How to modify the Makefile of a custom project

The routines in the sdk are relatively simple. It is OK to put all the source files in the same folder. However, when our project is relatively large, there will be many. c and. H files in the same folder, which will make it a little messy. At this time, it seems much more comfortable to put the code of different functional modules in different folders. How to implement it?

Take the simplest helloworld project as an example and modify it based on the official helloworld project.
Requirements:

1. Change the Project path from bl_iot_sdk/customer_app/get-start/helloworld to bl_iot_sdk/Ai-WB2-Demo/helloworld.

2. Add a folder component. Add hello.c and hello. H files to the component folder. Print helloworld into hello.c.

Implementation:

1. Configure the SDK path
   modify the sdk path of the Makeflie in helloworld
   BL60X_SDK_PATH_GUESS? = $(shell pwd)
   BL60X_SDK_PATH? = $(BL60X_SDK_PATH_GUESS)/... /... /...
   Change to
   BL60X_SDK_PATH_GUESS? = $(shell pwd)
   BL60X_SDK_PATH? = $(BL60X_SDK_PATH_GUESS)/... /...
2. Add components to Makeflie
   INCLUDE_COMPONENTS += component
   EXTRA_COMPONENT_DIRS += $(PROJECT_PATH)/component
3. Add a Makeflie file named **bouffilo. mk** to the component folder to compile hello.c. Add code to bouffilo. mk:

COMPONENT_OBJS := $(patsubst %.c,%.o, $(COMPONENT_SRCS))

For the complete project, see https://gitee.com/chencong_cc/Ai-WB2-Demo.git.

4. Development Data

Official bouffalolab SDK: https://github.com/bouffalolab/bl_iot_sdk the official routine is located in the customer_app folder.

Programming Guide: https://bouffalolab.github.io/bl_iot_sdk