**How to Setup Network with AIW-357**

**on Yocto**

**Prerequisite**

* ARK-1221 and 64G SSD storage
* Prebuild Yocto Image “ark-1221-yocto.img.bz2”
* Tarball file “aiw-357-mbim-set-ip.tar.bz2”
* Tarball file aiw357\_yocto-linux.tar.bz2 if needed to porting driver

**Flash image to SSD**

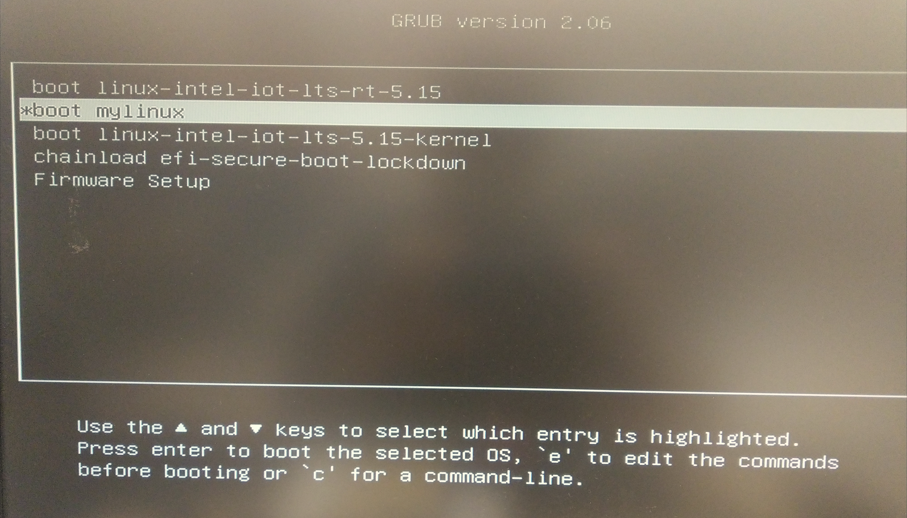
We flash the prebuild Yocto image to SSD by the below command on Ubuntu 20.04.

# bzcat ark-1221-yocto.img.bz2 | sudo dd of=/dev/sdX

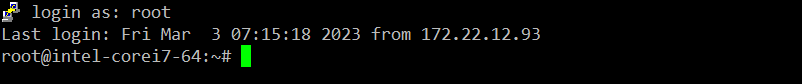
Note: /dev/sdX is depend on your system to find out the SSD storage

**Booting on ARK-1221**

After flash image complete, please select the item “bzImage-my” in grub menu as below screenshot and then the Yocto Linux can be booted by SSD with ARK-1220.



The user can login with default user “root” without password.



**Network Test**

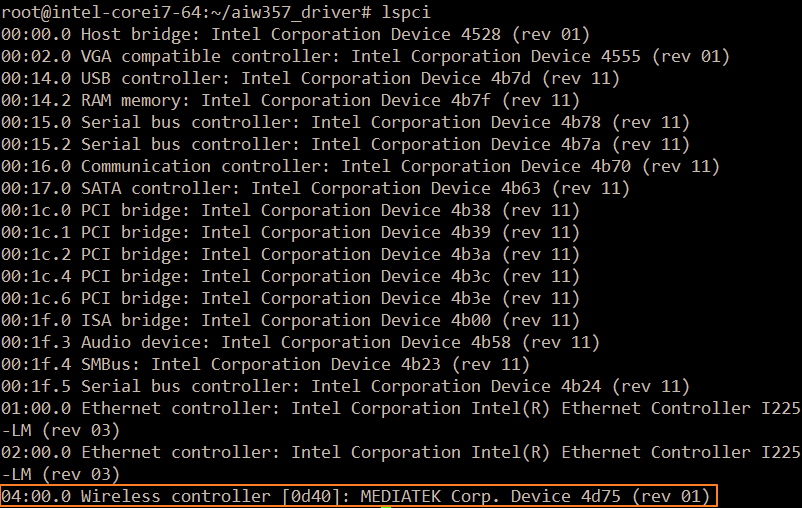
Please follow the below steps to test network on Yocto Linux.

1. Check some files that are build-in at home directory “/home/root on Yocto Linux.

* aiw-357-mbim-set-ip.tar.bz2 (application to bring up the network interface)
* aiw357\_driver/mtk\_pcie\_wwan\_m80.ko (AIW-357 driver)

1. Check AIW-357 module with PCIe

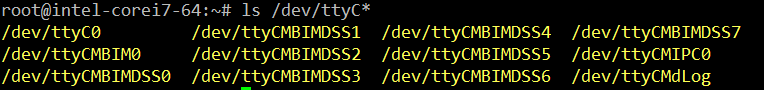
# lspci



1. Load driver

# insmod aiw357\_driver/mtk\_pcie\_wwan\_m80.ko

If the driver loaded successful, the user can find out device nodes as below.



1. Prepare some files for MBIM
   * Decompress the tarball aiw-357-mbim-set-ip.tar.bz2” and you will see the script “mbim-set-ip”

# tar jxvf “aiw-357-mbim-set-ip.tar.bz2”

* + Create the mbim configuration and copy it to /etc directory

# vi /etc/mbim-network.conf

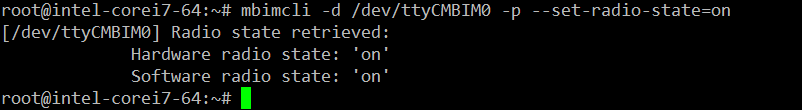
Please fill your APN in the below orange rectangle



1. Turn radio on

# mbimcli -d /dev/ttyCMBIM0 -p --set-radio-state=on

Note that please wait for 1-2 minutes to let module attaching to base station after running this command.



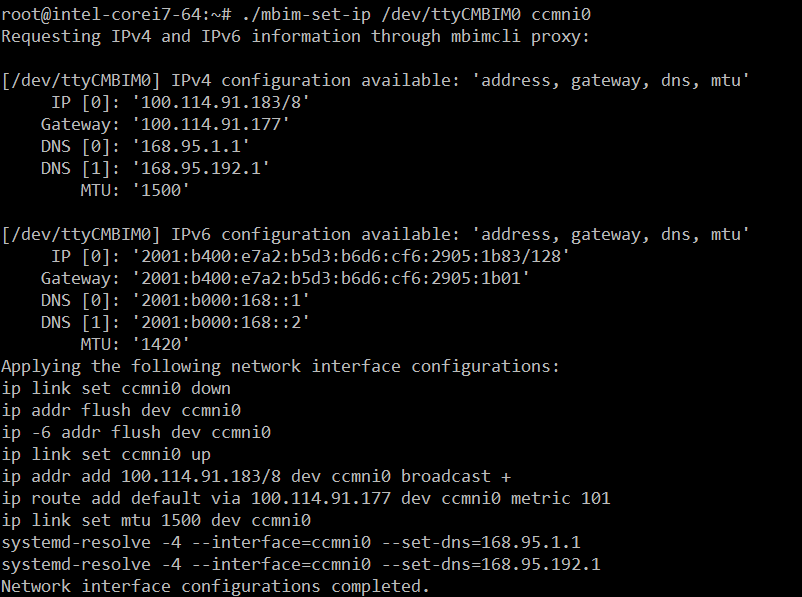
1. Start network connection

# mbim-network /dev/ttyCMBIM0 start



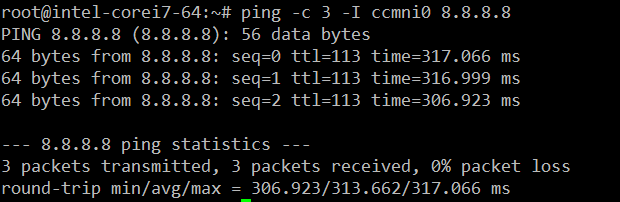
1. Set IP and related setting with connection

# ./mbim-set-ip /dev/ttyCMBIM0 ccmni0



1. Ping test

# ping -c 3 -I ccmni0 8.8.8.8

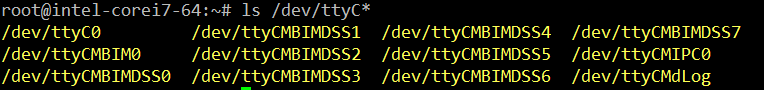


**AT command**

We can use AT command to communicate with AIW-357 module.

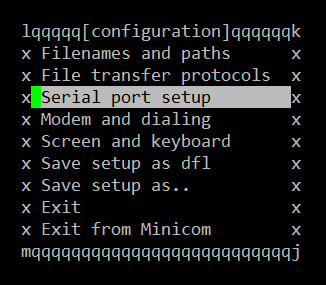
1. Check device node ttyC0 whether exist or not

# ls /dev/ttyC\*

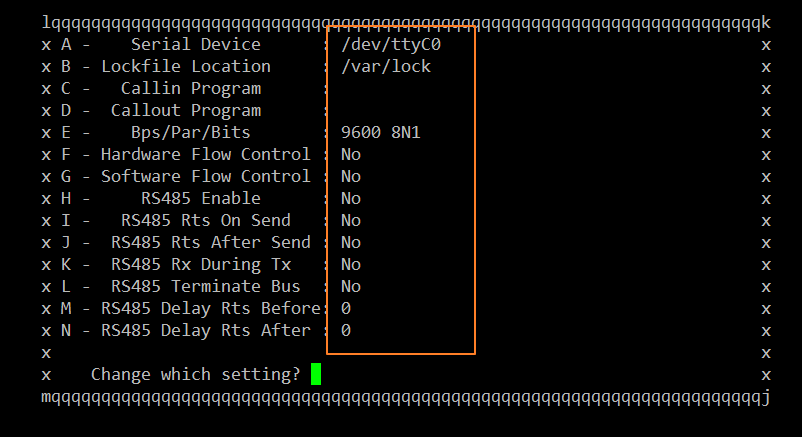


1. Get into AT command console by “Minicom”
   * Start to run minicom and select the “Serial port setup”

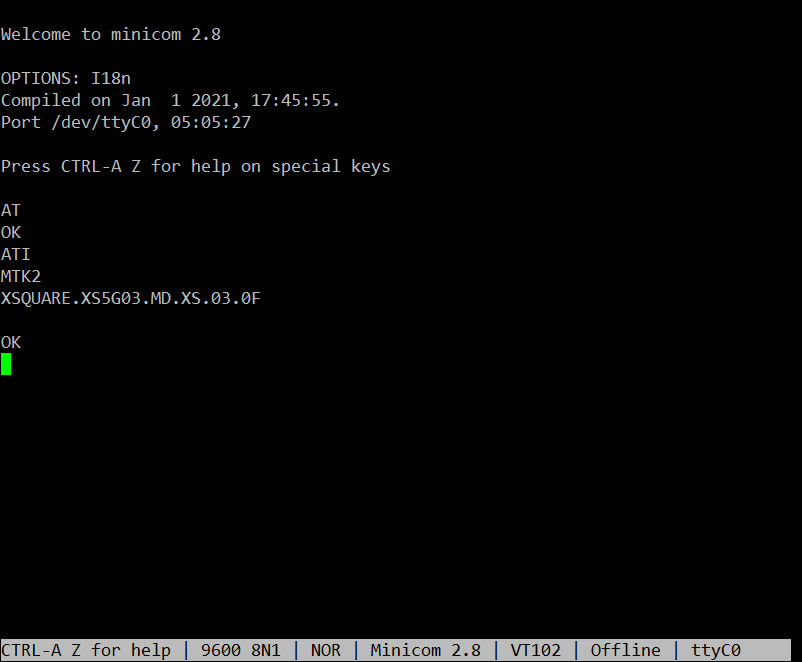
# minicom –s



* + Please set serial setting as below screenshot.



* + Test by AT command “AT” or “ATI”



**How to Porting Driver to Yocto with ARK-1221**

We setup development environment on Yocto Linux with ARK-1221. Please refer to the below steps for porting AIW-357 driver to Yocto Linux. The below table is about what we need on Yocto Linux after the tarball “aiw357\_yocto-linux.tar.bz2” is extracted

|  |  |
| --- | --- |
| **Directory** | **Description** |
| aiw357\_driver | AIW-357 driver source code |
| kernel | Yocto kernel source code |
| packages | Packages needed to install on Yocto |
| rootfs | The file of root file system is needed when building kernel |

1. Prepare some packages that are installed on Yocto Linux.
   * packages/bison-3.8.2-r0.corei7\_64.rpm
   * packages/gcc-plugins-11.3.0-r0.corei7\_64.rpm
   * packages/libmbim-1.26.2-r0.corei7\_64.rpm
   * packages/libmbim-bash-completion-1.26.2-r0.corei7\_64.rpm
   * packages/lrzsz-0.12.20-r6.corei7\_64.rpm
   * packages/minicom-2.8-r0.corei7\_64.rpm
2. Prepare the kernel source.

* kernel/linux-5.15.49.tar.bz2 (kernel source)
* kernel/boot\_yocto.tar.bz2 -> boot\_yocto/config-5.15.49-rt47-intel-ese-standard-lts (kernel configuration)

Please make sure the kernel setting with the below options if the user has the customized configuration：

* + Enable PCIE related options
    - CONFIG\_PCIEPORTBUS=y
    - CONFIG\_PCIEAER=y
    - CONFIG\_PCIEASPM=y
    - CONFIG\_PCIEASPM\_DEFAULT=y
    - CONFIG\_PCI\_MSI=y
  + Disable MTK T7XX
    - CONFIG\_MTK\_T7XX is not set
  + Locate the file system file if needed
    - CONFIG\_INITRAMFS\_SOURCE="/home/root/rootfs/mender-initramfs-intel-corei7-64.cpio"

1. Rebuild the kernel and modules

* Build the kernel

# make bzImage

* Build the modules and install it to root file-system

# make modules

# make modules\_install

* Copy the kernel image and related files to /boot directory.

# cp /kernel\_src/arch/x86/boot/bzImage /boot/bzImage-5.15.49-rt47-intel-ese-standard-lts

# cp /kernel\_src/System.map /boot/System.map-5.15.49-rt47-intel-ese-standard-lts

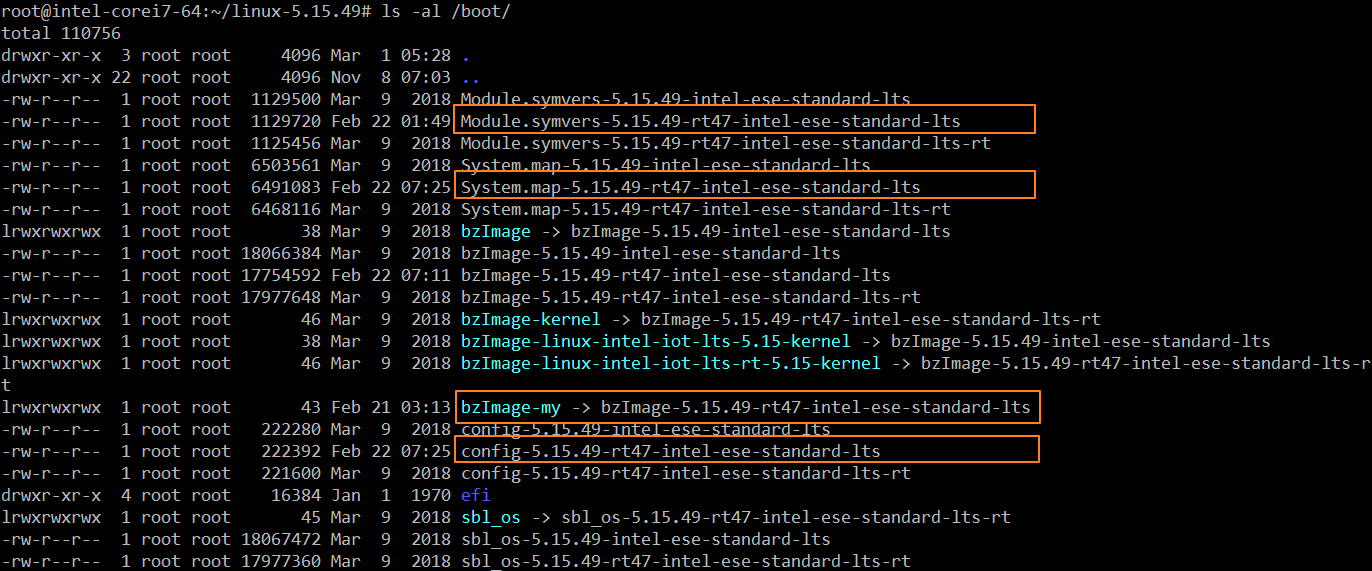
# cp /kernel\_src/Module.symvers /boot/Module.symvers-5.15.49-rt47-intel-ese-standard-lts

# cp /kernel\_src/.config /boot/config-5.15.49-rt47-intel-ese-standard-lts

* Create soft link to kernel image

# cd /boot

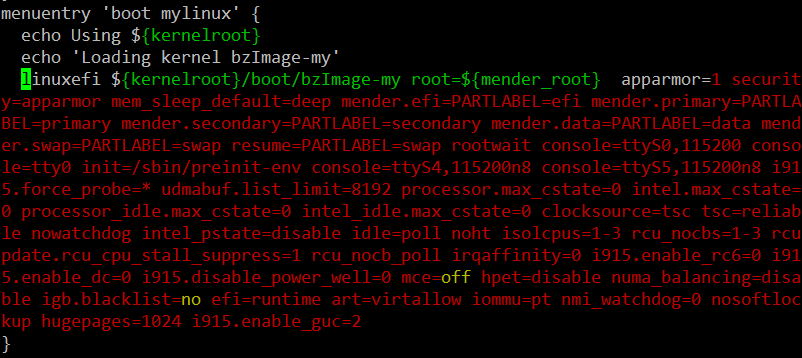
# ln –sf bzImage-5.15.49-rt47-intel-ese-standard-lts bzImage-my



1. Reboot by the rebuild kernel

* Modify boot-loader to boot from rebuild kernel.

In Yocto Linux, the boot configuration is at /boot/efi/EFI/BOOT/grub.cfg . Please refer to “kernel/boot\_yocto.tar.bz2 -> boot\_yocto/grub.cfg” for the below screenshot.



1. Build the AIW-357 driver

* Decompress the driver tarball “aiw357\_driver/xs5g03\_pcie\_driver\_Advantech.tar.gz”

# tar zxvf xs5g03\_pcie\_driver\_Advantech.tar.gz

* Build the driver

# cd XS5G03\_pcie\_driver\_Advantech

# make

If successful, you will find out the driver “mtk\_pcie\_wwan\_m80.ko”