软件环境：

Windows 端：

- CCS 10.2.0
- simplelink_cc13x0_sdk_4_20_00_05

Linux 端：

- Ubuntu 18.04 x64
- TI-15-4-STACK-GATEWAY-LINUX-SDK_3.30.01.02

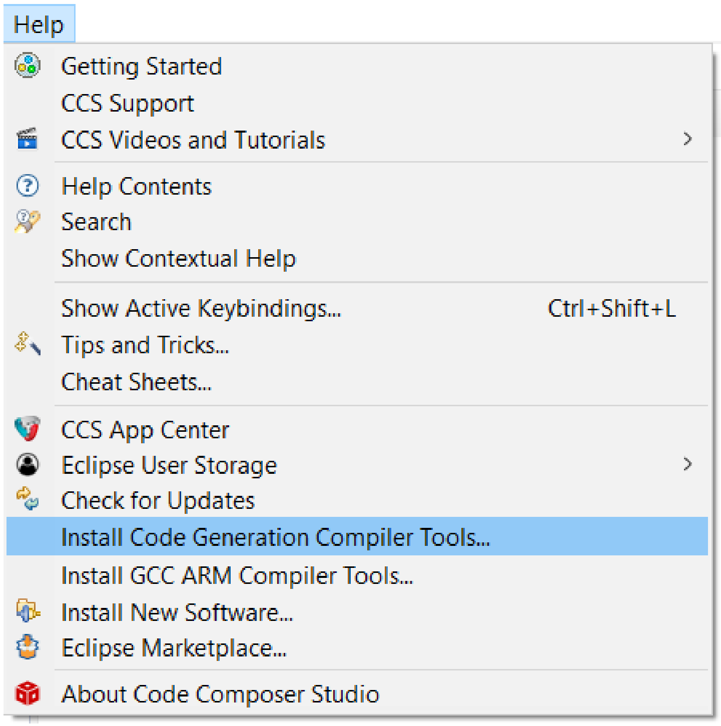
1. 在 CCS 中安装 ARM Compiler Tools 18.12.8

CC1310 SDK 中推荐的软件环境为：

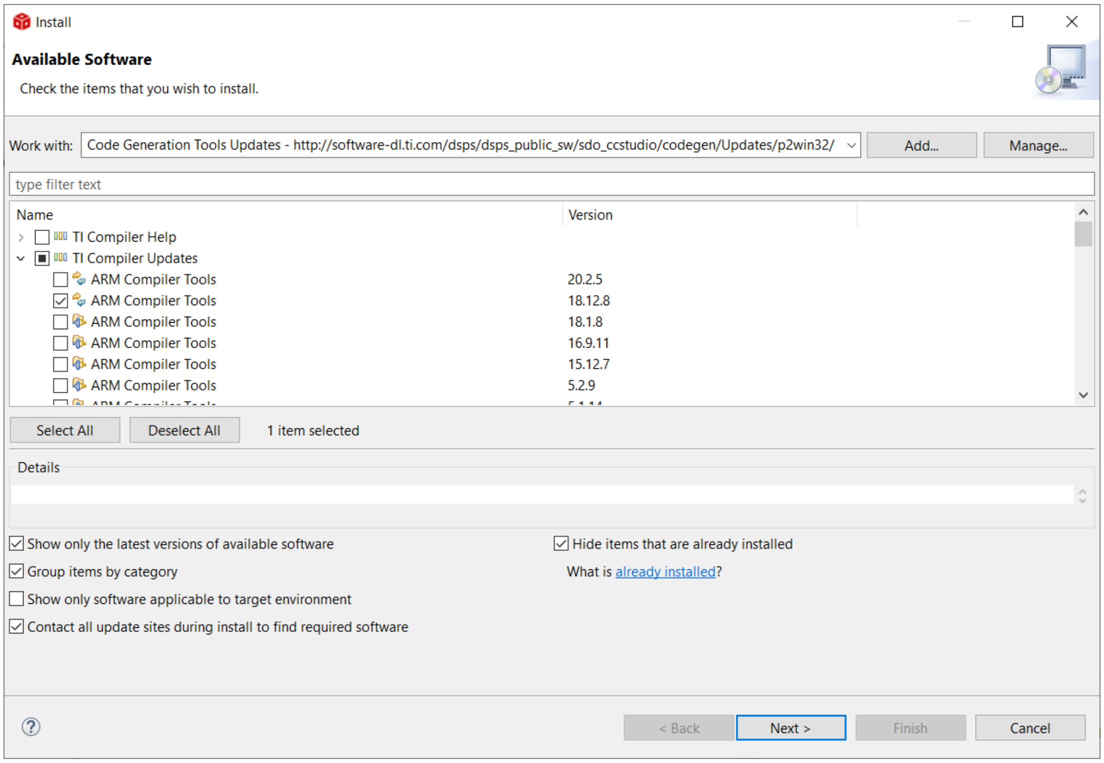
- TI Code Composer Studio: 9.1.0
- IAR Embedded Workbench for Arm: 8.32.2
- C Compiler for Arm: 18.12.2.LTS
- XDCTools: 3.51.03.28
- TI-RTOS for CC13XX: 4.20.05.01

其中 XDCTools 和 TI-RTOS 会随 SDK 一起安装，但 CCS10.0 以上版本中默认 ARM 编译器是 20.0.2 版本，所以建议手动安装 18.12.x 版本的 ARM 编译器，步骤如下：

1) 点击 help->Install Code Generation Compiler Tools



2) 勾选 TI Compiler Updates -> ARM Compiler Tools 18.12.8，然后一路 next 完成安装，安装完成后按照提示重启 CCS



Install

Available Software

Check the items that you wish to install.

Work with: Code Generation Tools Updates - http://software-dl.ti.com/dsps/dsps_public_sw/sdo_ccstudio/codegen/Updates/p2win32/

type filter text

Name	Version
□ TI Compiler Help	
■ TI Compiler Updates	
□ ARM Compiler Tools	20.2.5
☑ ARM Compiler Tools	18.12.8
□ ARM Compiler Tools	18.1.8
□ ARM Compiler Tools	16.9.11
□ ARM Compiler Tools	15.12.7
□ ARM Compiler Tools	5.2.9

1 item selected

Details

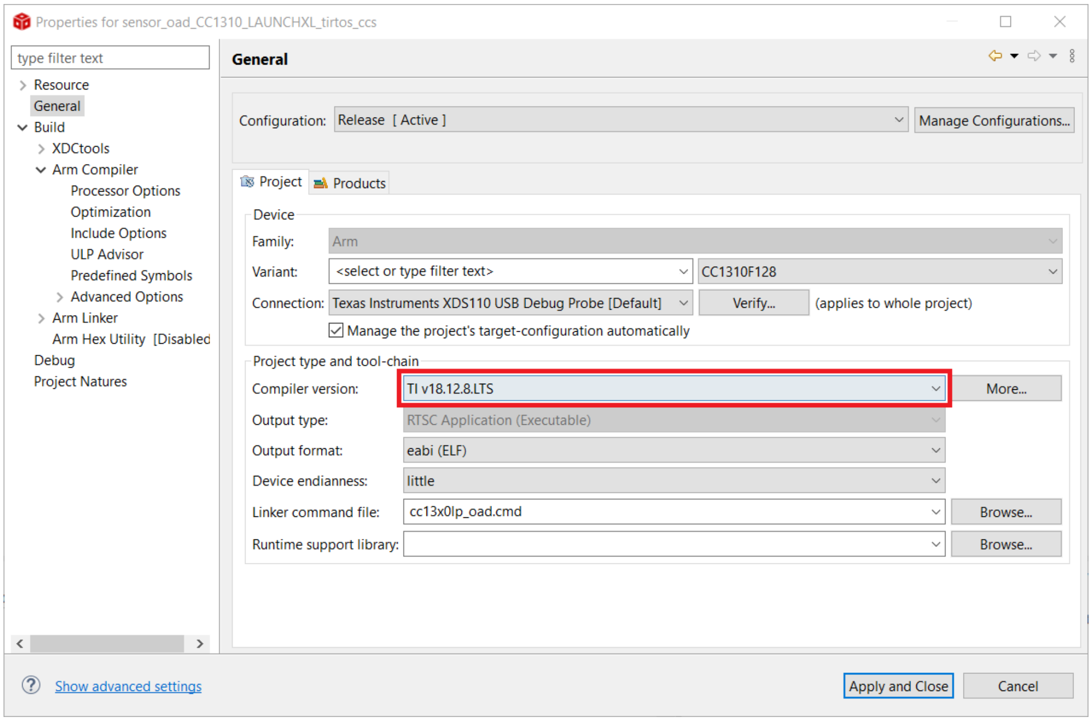
- ☑ Show only the latest versions of available software
- ☑ Group items by category
- □ Show only software applicable to target environment
- ☑ Contact all update sites during install to find required software
- ☑ Hide items that are already installed

What is already installed?

2. 制作 Turbo OAD 所需镜像

1) 在 Windows 端 CCS 里导入 sensor_oad 工程；

2) 在 sensor_oad 工程属性中将编译器版本修改为 18.12.8



Properties for sensor_oad_CC1310_LAUNCHXL_tirtos_ccs

General

Configuration: Release [Active]

Project | Products

Device

- Family: Arm
- Variant: <select or type filter text> CC1310F128
- Connection: Texas Instruments XDS110 USB Debug Probe [Default] (applies to whole project)
- ☑ Manage the project's target-configuration automatically

Project type and tool-chain

- Compiler version: TI v18.12.8.LTS
- Output type: RTSC Application (Executable)
- Output format: eabi (ELF)
- Device endianness: little
- Linker command file: cc13x0lp_oad.cmd
- Runtime support library:

3) 在工程中的 Application -> Degines -> sensor_oad_offchip.opts 中，将 DxFEATURE_TOAD 改为 DFEATURE_TOAD

4) 编译 sensor_oad 工程，得到 sensor_oad_CC1310_LAUNCHXL_tirtos_ccs.hex 文件（此 hex 文件后面烧写 CC1310 要用，要备份一下）

5) 运行 oad_image_tool.py，得到 sensor_oad_cc13x0lp_app.bin 文件

```
python  ../../../../../../tools/common/oad/oad_image_tool.py -v [0xXXYY] -i app
sensor_oad_CC1310_LAUNCHXL_tirtos_ccs.hex -ob sensor_oad_cc13x0lp_app.bin -m 0x14F0 -
r :0x1E000
```

6) 修改 oad_client.c 中的版本号 FW_VERSION，rebuild 工程，并重复步骤 5）得到更新后的 bin 文件，此处分别将两个 bin 文件命名为 sensor_oad_cc13x0lp_app_v2.bin 和 sensor_oad_cc13x0lp_app_v3.bin

7) 运行 toad_image_tool.py，得到 sensor_oad_cc13x0lp_app.dim 文件

```
python <SDK_DIR>/tools/ti154stack/turbo_oad/toad_image_tool.py -oimg
sensor_oad_cc13x0lp_app_v2.bin -nimg sensor_oad_cc13x0lp_app_v3.bin -o
sensor_oad_cc13x0lp_app.dim
```

8) 将此文件拷贝到 Linux 端备用

3. 配置 Linux Collector

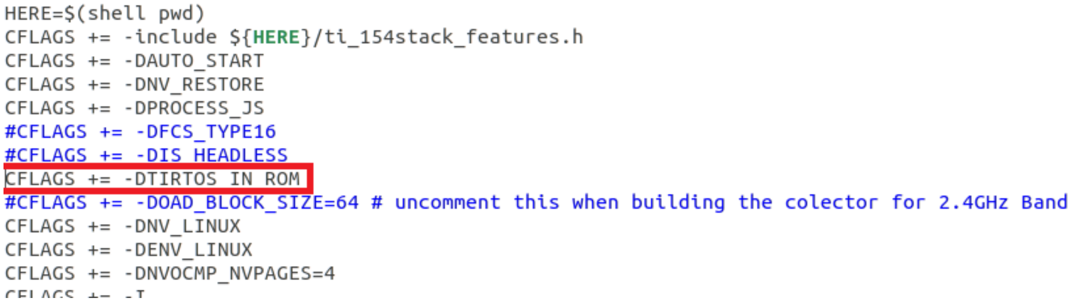
1) 在 Windows 端将 simplelink_cc13x0_sdk_4_20_00_05\examples\rtos\CC1310_LAUNCHXL\ti154stack\hexfiles\coprocessor_cc1310lp_tirtos_ccs.hex 烧写到 CC1310 LaunchPad

2) 在 Linux 端安装 ti154stack_linux_x64_3_30_01_02，并按照其中 User’s Guide 的要求安装相关依赖

```
$sudo apt-get update
$sudo apt-get install build-essential
$sudo apt-get install Node.js
$sudo adduser $USER dialout
```

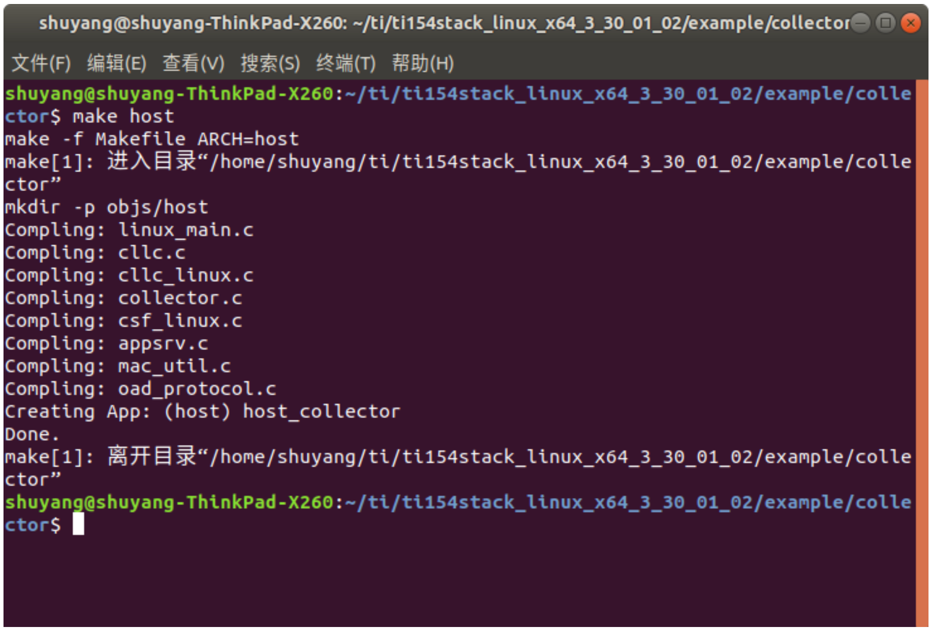
3) 运行 ti154stack_linux_x64_3_30_01_02 文件夹下的 build_all.sh 脚本

4) 修改 example -> collector ->make 文件，去掉 CFLAGS += -DTIRTOS_IN_ROM 前面的注释符



```
HERE=$(shell pwd)
CFLAGS += -include ${HERE}/ti_154stack_features.h
CFLAGS += -DAUTO_START
CFLAGS += -DNV_RESTORE
CFLAGS += -DPROCESS_JS
#CFLAGS += -DFCS_TYPE16
#CFLAGS += -DIS_HEADLESS
CFLAGS += -DTIRTOS_IN_ROM
#CFLAGS += -DOAD_BLOCK_SIZE=64 # uncomment this when building the colector for 2.4GHz Band
CFLAGS += -DNV_LINUX
CFLAGS += -DENV_LINUX
CFLAGS += -DNVOCMP_NVPAGES=4
CFLAGS += -I
```

5) 在 example -> collector 文件夹中启动终端，执行 make host


shuyang@shuyang-ThinkPad-X260: ~/ti/ti154stack_linux_x64_3_30_01_02/example/collector
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

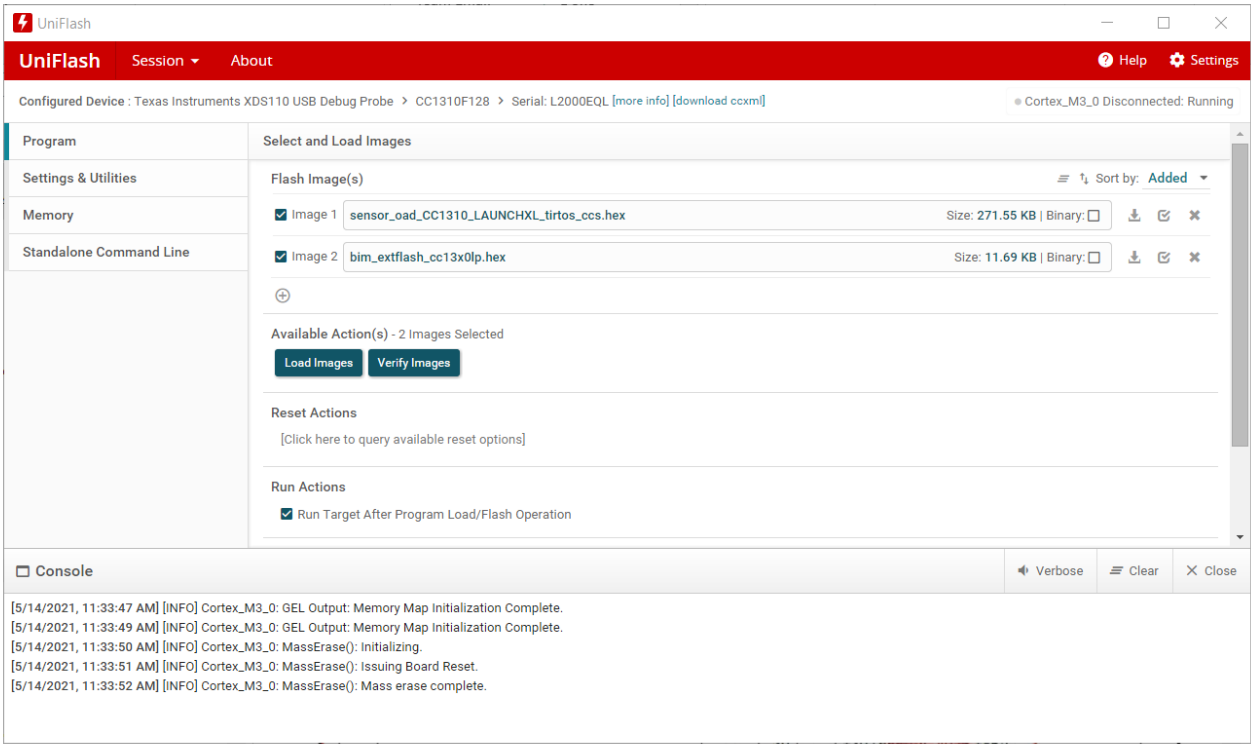
```
shuyang@shuyang-ThinkPad-X260:~/ti/ti154stack_linux_x64_3_30_01_02/example/colle
ctor$ make host
make -f Makefile ARCH=host
make[1]: 进入目录“/home/shuyang/ti/ti154stack_linux_x64_3_30_01_02/example/colle
ctor”
mkdir -p objs/host
Compling: linux_main.c
Compling: cllc.c
Compling: cllc_linux.c
Compling: collector.c
Compling: csf_linux.c
Compling: appsrv.c
Compling: mac_util.c
Compling: oad_protocol.c
Creating App: (host) host_collector
Done.
make[1]: 离开目录“/home/shuyang/ti/ti154stack_linux_x64_3_30_01_02/example/colle
ctor”
shuyang@shuyang-ThinkPad-X260:~/ti/ti154stack_linux_x64_3_30_01_02/example/colle
ctor$ 
```

6) 将步骤 1）中的 LaunchPad 连接至 Linux PC，运行 host_collector

```
$./host_collector collector.cfg
```

4. 准备 sensor 端

1) 在 Windows 端将 bim_offchip 工程导入 CCS
2) 在工程属性中将编译器版本修改为 18.12.8
3) 编译工程得到 bim_extflash_cc13x0lp.hex 文件
4) 将 CC1310 LaunchPad 连接 Windows PC，打开 Uniflash，将 sensor_oad_CC1310_LAUNCHXL_tirtos_ccs.hex 和 bim_extflash_cc13x0lp.hex 同时烧写进 LaunchPad。如果此 LaunchPad 之前加入过网络，建议先烧写 simplelink_cc13x0_sdk_4_20_00_05\examples\rtos\CC1310_LAUNCHXL\easylink\hexfiles\offChipOad\ erase_extflash_cc13x0lp.hex 清除外部 flash，再烧写 sensor 程序



Configured Device : Texas Instruments XDS110 USB Debug Probe > CC1310F128 > Serial: L2000EQL [more info] [download ccxml]

Cortex_M3_0 Disconnected: Running

☑ Image 1 sensor_oad_CC1310_LAUNCHXL_tirtos_ccs.hex Size: 271.55 KB | Binary: □

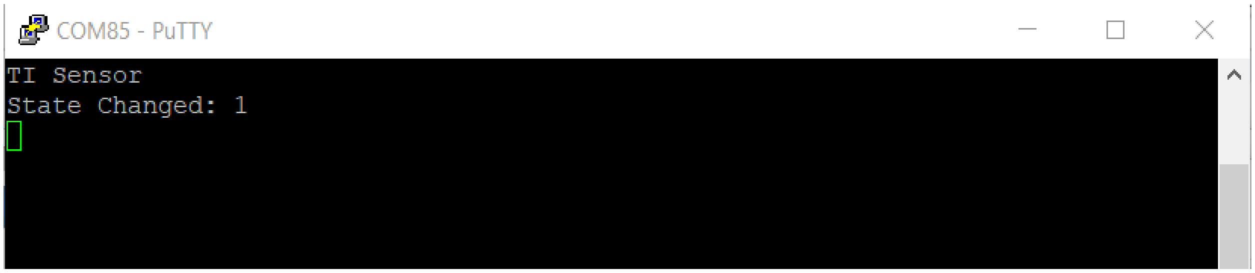
☑ Image 2 bim_extflash_cc13x0lp.hex Size: 11.69 KB | Binary: □

Available Action(s) - 2 Images Selected

☑ Run Target After Program Load/Flash Operation

```
[5/14/2021, 11:33:47 AM] [INFO] Cortex_M3_0: GEL Output: Memory Map Initialization Complete.
[5/14/2021, 11:33:49 AM] [INFO] Cortex_M3_0: GEL Output: Memory Map Initialization Complete.
[5/14/2021, 11:33:50 AM] [INFO] Cortex_M3_0: MassErase(): Initializing.
[5/14/2021, 11:33:51 AM] [INFO] Cortex_M3_0: MassErase(): Issuing Board Reset.
[5/14/2021, 11:33:52 AM] [INFO] Cortex_M3_0: MassErase(): Mass erase complete.
```

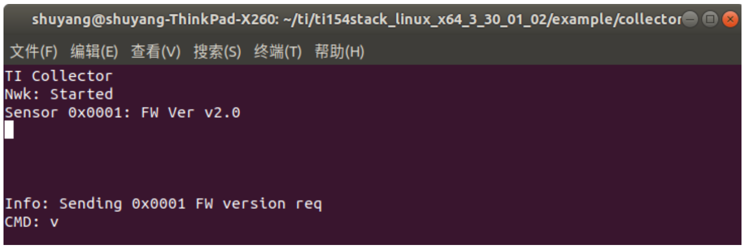
5) 按 LaunchPad 上的 reset 按键复位，sensor 程序开始运行，打开 putty（或其他串口终端），可以看到如下界面：



```
TI Sensor
State Changed: 1
```

5. 进行 Turbo OAD

1) 接着第 3 部分，在 Linux 端输入命令 o 打开网络，sensor 会自动加入网络并上传温度和 RSSI 数据
2) 输入 s1 命令选择当前操作的 sensor，之后可用 t 命令开关 sensor 上的 LED 灯来验证是否成功
3) 输入 v 命令，查看当前 sensor 中 firmware 版本



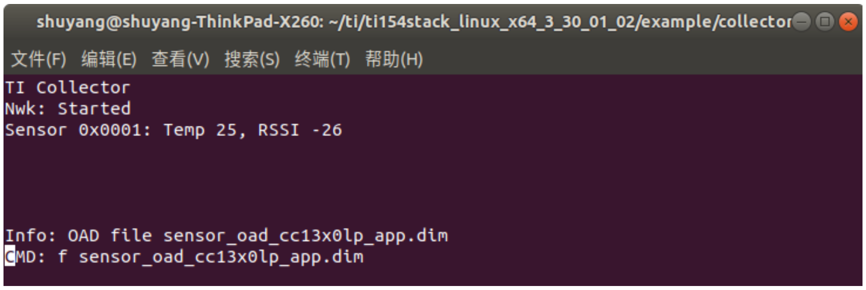
shuyang@shuyang-ThinkPad-X260: ~/ti/ti154stack_linux_x64_3_30_01_02/example/collector
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

```
TI Collector
Nwk: Started
Sensor 0x0001: FW Ver v2.0



Info: Sending 0x0001 FW version req
CMD: v
```

4) 将第 2 部分制作完成的 sensor_oad_cc13x0lp_app.dim 文件拷贝到 Linux 端 ti154stack_linux_x64_3_30_01_02/example/collector 文件夹下，然后在 host_collector 中使用命令 f 指定该镜像为将要传送的 turbo OAD 镜像

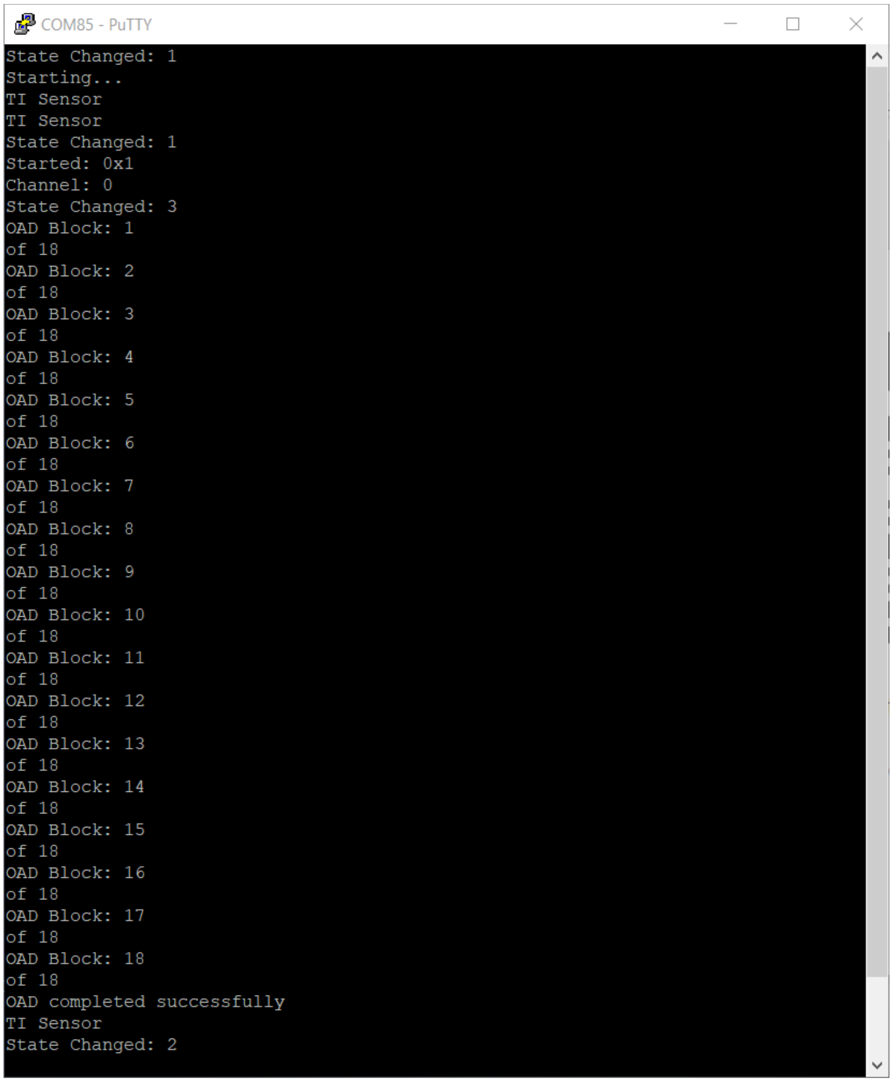

shuyang@shuyang-ThinkPad-X260: ~/ti/ti154stack_linux_x64_3_30_01_02/example/collector
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

```
TI Collector
Nwk: Started
Sensor 0x0001: Temp 25, RSSI -26



Info: OAD file sensor_oad_cc13x0lp_app.dim
CMD: f sensor_oad_cc13x0lp_app.dim
```

5) 输入命令 u 开始执行 OAD，在 sensor 端的串口终端里会显示传送过程，成功后会显示 OAD completed successfully



```
State Changed: 1
Starting...
TI Sensor
TI Sensor
State Changed: 1
Started: 0x1
Channel: 0
State Changed: 3
OAD Block: 1
of 18
OAD Block: 2
of 18
OAD Block: 3
of 18
OAD Block: 4
of 18
OAD Block: 5
of 18
OAD Block: 6
of 18
OAD Block: 7
of 18
OAD Block: 8
of 18
OAD Block: 9
of 18
OAD Block: 10
of 18
OAD Block: 11
of 18
OAD Block: 12
of 18
OAD Block: 13
of 18
OAD Block: 14
of 18
OAD Block: 15
of 18
OAD Block: 16
of 18
OAD Block: 17
of 18
OAD Block: 18
of 18
OAD completed successfully
TI Sensor
State Changed: 2
```