RflySim-RT

None

kechenxu@buaa.edu.cn

None

Table of contents

1. RflySim-RT

RflySim-RT RflySim
RflySim RT

RflySim-RT SocFPGA ARM A53 FPGA

1.1 1.1

- ■ ARM A53 , .
- ■ Letter Shell , . xc_shell.
- ■ FATFS SD littlefs
- ■ wifi RflySim3D ,
- ■ wifi Shell
- ■ 2000Hz
- ■ (cJSON) . SD , PC , , ,

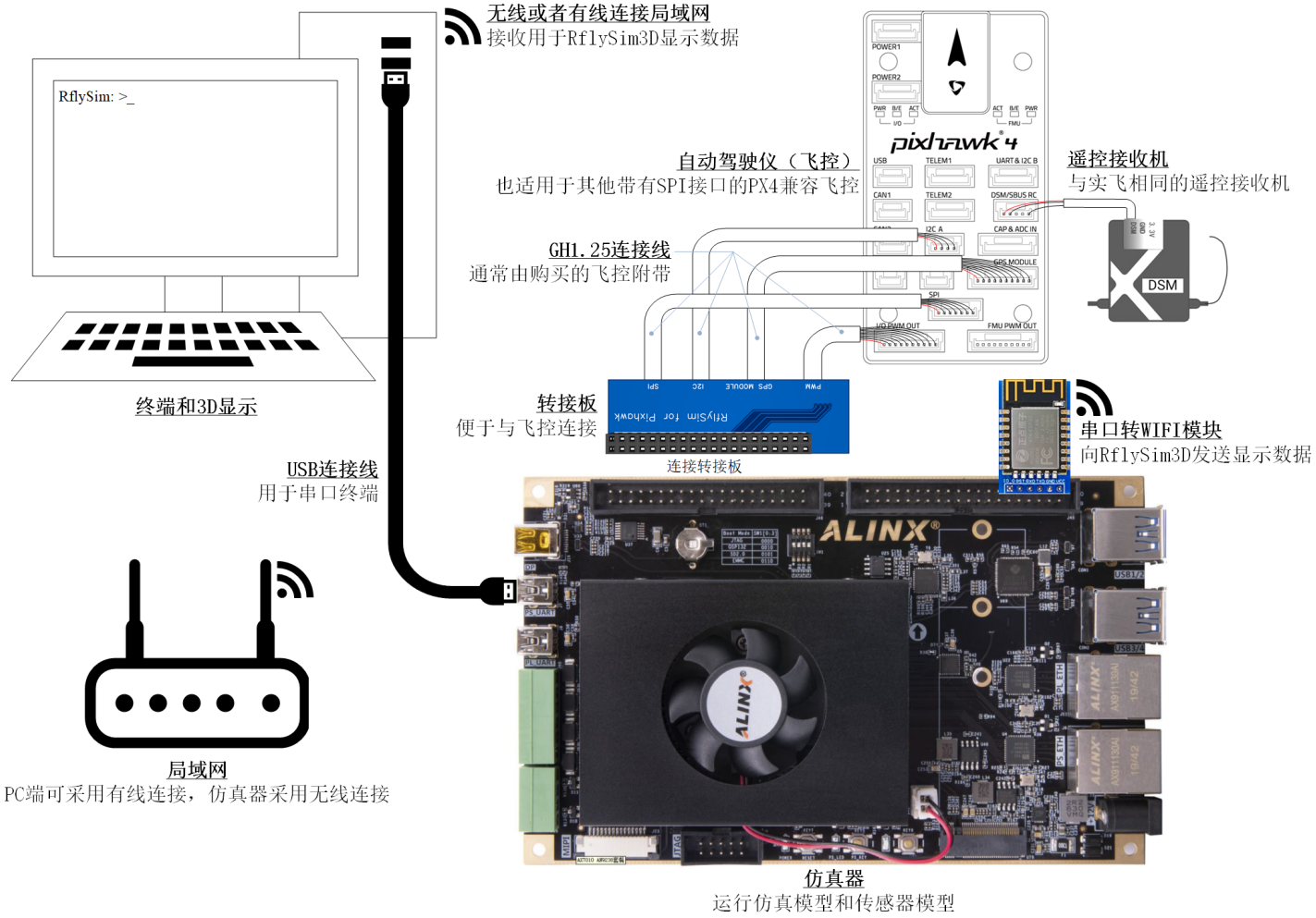
1.2 1.2

- □ PX4 ulog , ulog API .
- □ MAVLINK , QGC ,
- □ PC
- □ , PC (shell PC), , , , .
- □ ls cd rm cat

2.

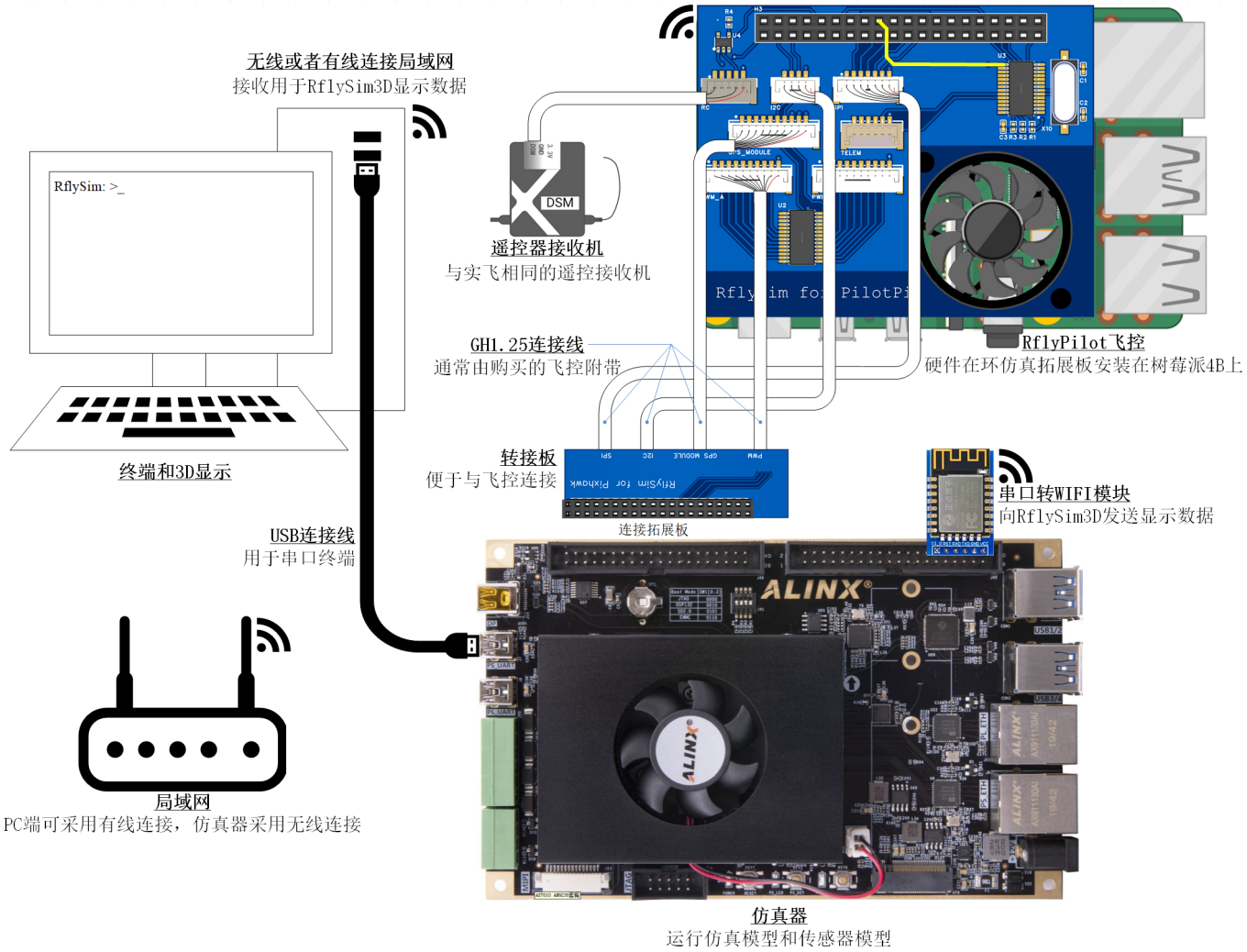
2.1

2.1.1 1.


无线或者有线连接局域网
接收用于RflySim3D显示数据
RflySim: >_
终端和3D显示
自动驾驶仪（飞控）
也适用于其他带有SPI接口的PX4兼容飞控
pixhawk 4
遥控接收机
与实飞相同的遥控接收机
DSM
GH1.25连接线
通常由购买的飞控附带
转接板
便于与飞控连接
连接转接板
串口转WIFI模块
向RflySim3D发送显示数据
USB连接线
用于串口终端
局域网
PC端可采用有线连接，仿真器采用无线连接
ALINX
仿真器
运行仿真模型和传感器模型

Tip

RflyPilot RflyPilot



Tip

1.1

- ALINX MPSoC AXU2CG-E XCZU2CG-SFVC784-1-E
- Pixhawk 4
- GH1.25
- wifi

Note

Pixhawk 4 SPI PX4 SPI

Pixhawk 4	STM32F765	216MHz
Pixhawk 5X	STM32F765	216MHz
Pixhawk 6X	STM32H753	480MHz
Holybro Durandal	STM32H743	480MHz
CUAV V5+	STM32F765	216MHz
CUAV X7+	STM32H743	480MHz
CUAV X7+ Pro	STM32H743	480MHz
CUAV Pixhawk V6X	STM32H753	480MHz

Kore Carrier board Hex Cube- SPI

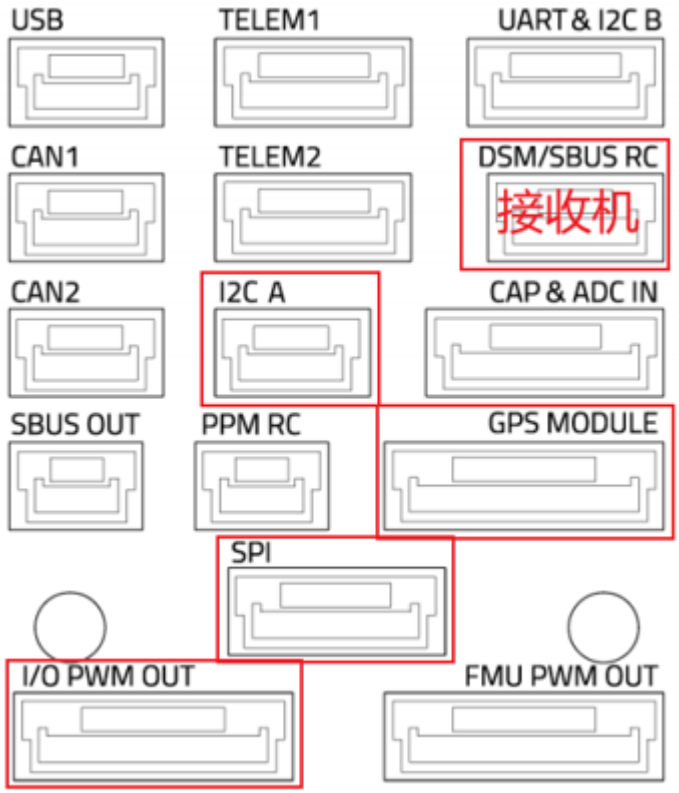
1.2

GH1.25 , .

Pixhawk 4

Note

PIN 1, VCC


USB
TELEM1
UART & I2C B
CAN1
TELEM2
DSM/SBUS RC
接收机
CAN2
I2C A
CAP & ADC IN
SBUS OUT
PPM RC
GPS MODULE
SPI
I/O PWM OUT
FMU PWM OUT


I/O PWM OUT
SPI
GPS MODULE
PPM RC
SBUS OUT
CAP & ADC IN
I2C A
CAN2

I2C A

Pin	Signal	Volt
1(red)	VCC	+5V
2(black)	SCL4	+3.3V
3(black)	SDA4	+3.3V
4(black)	GND	+3.3V

Pixhawk GPS

Pin	Signal	Volt
1(red)	VCC	+5V
2(black)	TX4(out)	+3.3V
3(black)	RX4(in)	+3.3V
4(black)	SCL1	+3.3V
5(black)	SDA1	+3.3V

Pixhawk SPI

Pin	Signal	Volt
1(red)	VCC	+5V
2(black)	SCK	+3.3V
3(black)	MISO	+3.3V
4(black)	MOSI	+3.3V
5(black)	CS1	+3.3V
6(black)	CS2	+3.3V
7(black)	GND	GND

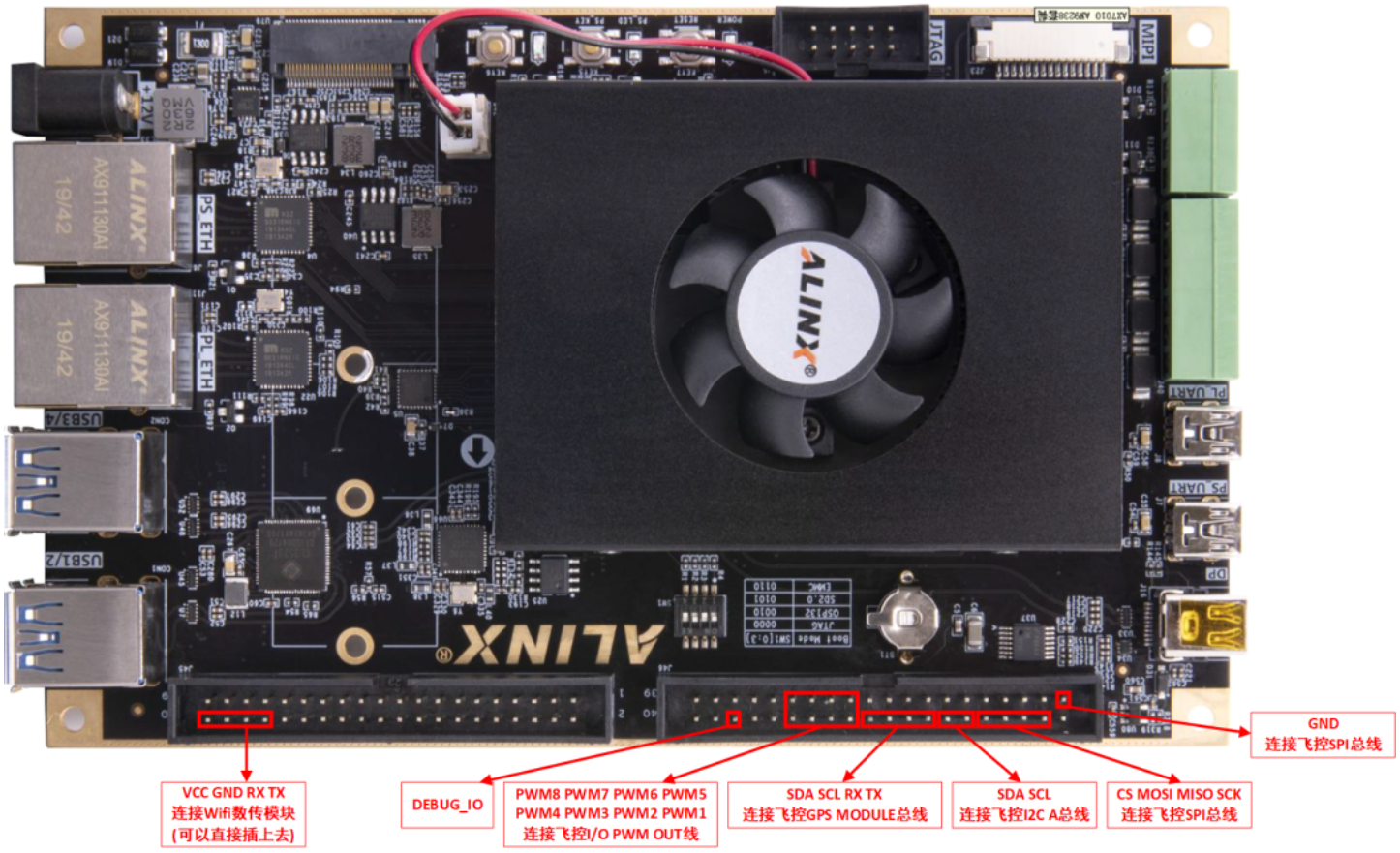
Pixhawk I/O PWM OUT FMU PWM OUT

Pin	Signal	Volt
1(red)	VDD_SERVO	
2(black)	IO_CH1\FMU_CH1	+3.3V
3(black)	IO_CH2\FMU_CH2	+3.3V
4(black)	IO_CH3\FMU_CH3	+3.3V
5(black)	IO_CH4\FMU_CH4	+3.3V
6(black)	IO_CH5\FMU_CH5	+3.3V
7(black)	IO_CH6\FMU_CH6	+3.3V
8(black)	IO_CH7\FMU_CH7	+3.3V
9(black)	IO_CH8\FMU_CH8	+3.3V
10(black)	GND	GND

Note

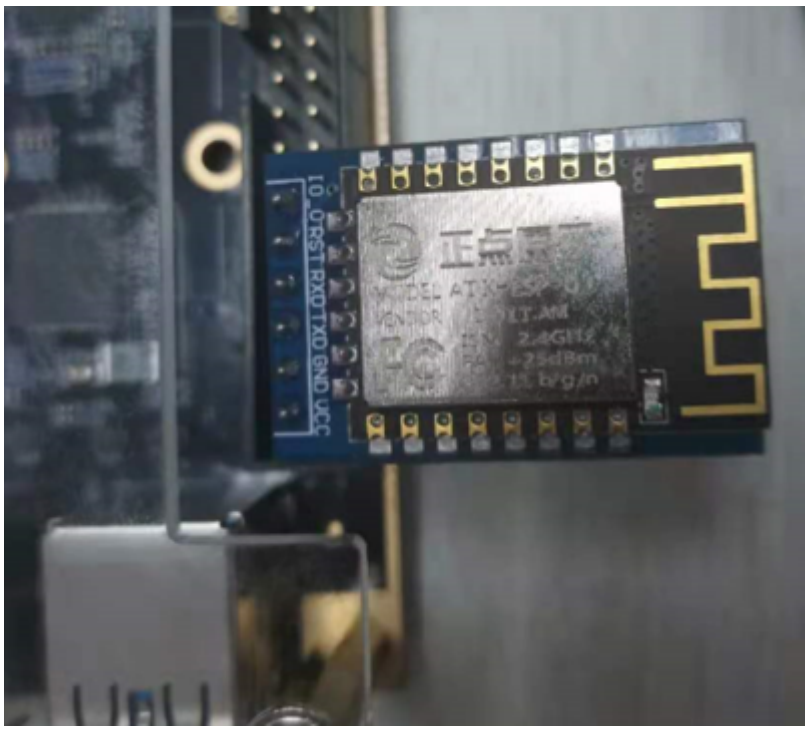
Pixhawk4-Pinouts.pdf

IO ,



FPGA IO引脚定义，与飞控对应接口匹配关系

IO	wifi



1.3

1. PS USB

> **Note**
>
> **MobaXterm**\
> Putty\
> SecureCRT\
> WindTerm

2.

> **Note**
>
> DC ,

2.1.2 2.

RflySim-RT FPGA Pixhawk 4 SPI I2C
SPI I2C
PX4 SPI IMU

> **Note**
>
> PX4 v1.11 v1.12 PX4 1.12

RflySim RflySim-RT Pixhawk 4 **X:**
\PX4PSP\Firmware\boards\px4\fmu-v5\init\rc.board_sensors
\Firmware\boards `X` RflySim C

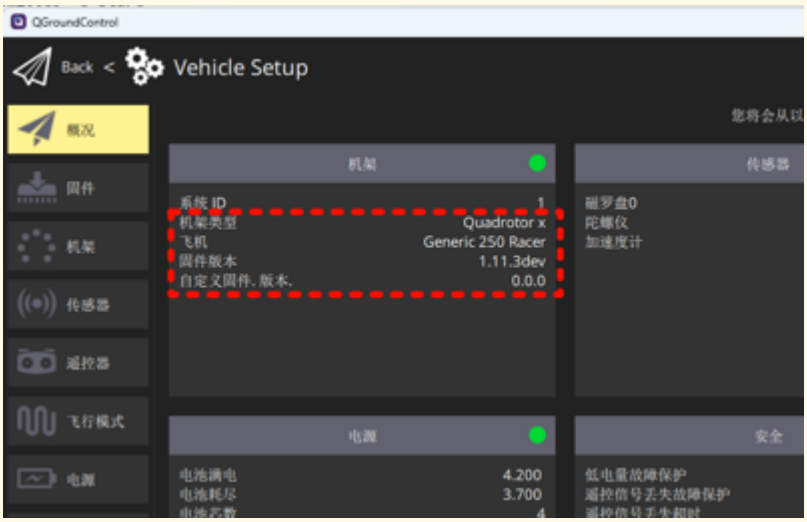
> **Note**
>
> `Firmware\boards\px4\fmu-v5\init\` rc.board_sensors vscode
> Windows

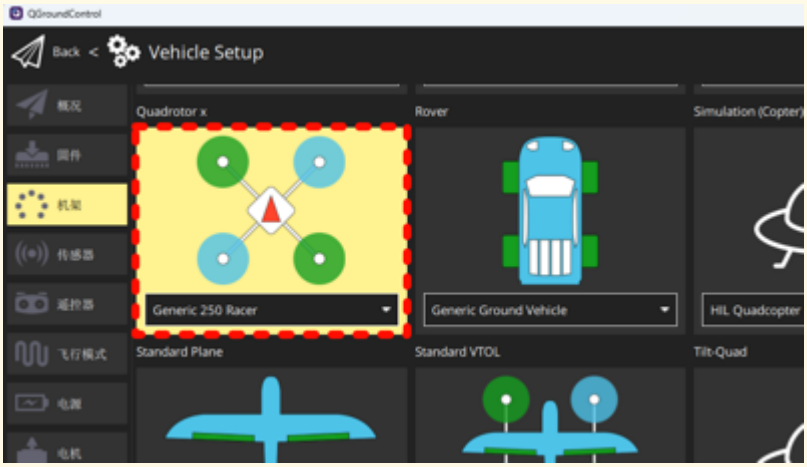
rc.board_sensors

```
#!/bin/sh
#
# PX4 FMUv5 specific board sensors init
#------------------------------------------------------------------------------

adc start
if ! icm20689 -S start
then
# Internal SPI bus ICM-20602
#icm20602 -s -R 2 start

# Internal SPI bus ICM-20689
icm20689 -s -R 2 start

# Internal SPI bus BMI055 accel/gyro
#bmi055 -A -R 2 -s start
#bmi055 -G -R 2 -s start

# internal compass
#ist8310 -I start
fi

if ! ms5611 -X start
then
# Baro on internal SPI
ms5611 -s start
fi
#icm20689 -S start
#ms5611 -X start
```

Warning

- PX4 1.11.3





2.1.3 3.

```
SER_GPS1_BAUD        115200 8N1          Baudrate for the GPS 1 Serial Port
```

```
GPS_UBX_DYNMODEL     airborne with <1g ac u-blox GPS dynamic platform model
```

2.1.4 4.

SD :

1. SD SD
   https://bhpan.buaa.edu.cn:443/link/1FF3201F500F3D00A9C483235A36B653 2023-09-01 23:59

Warning

- SD FAT32
- BOOT.bin

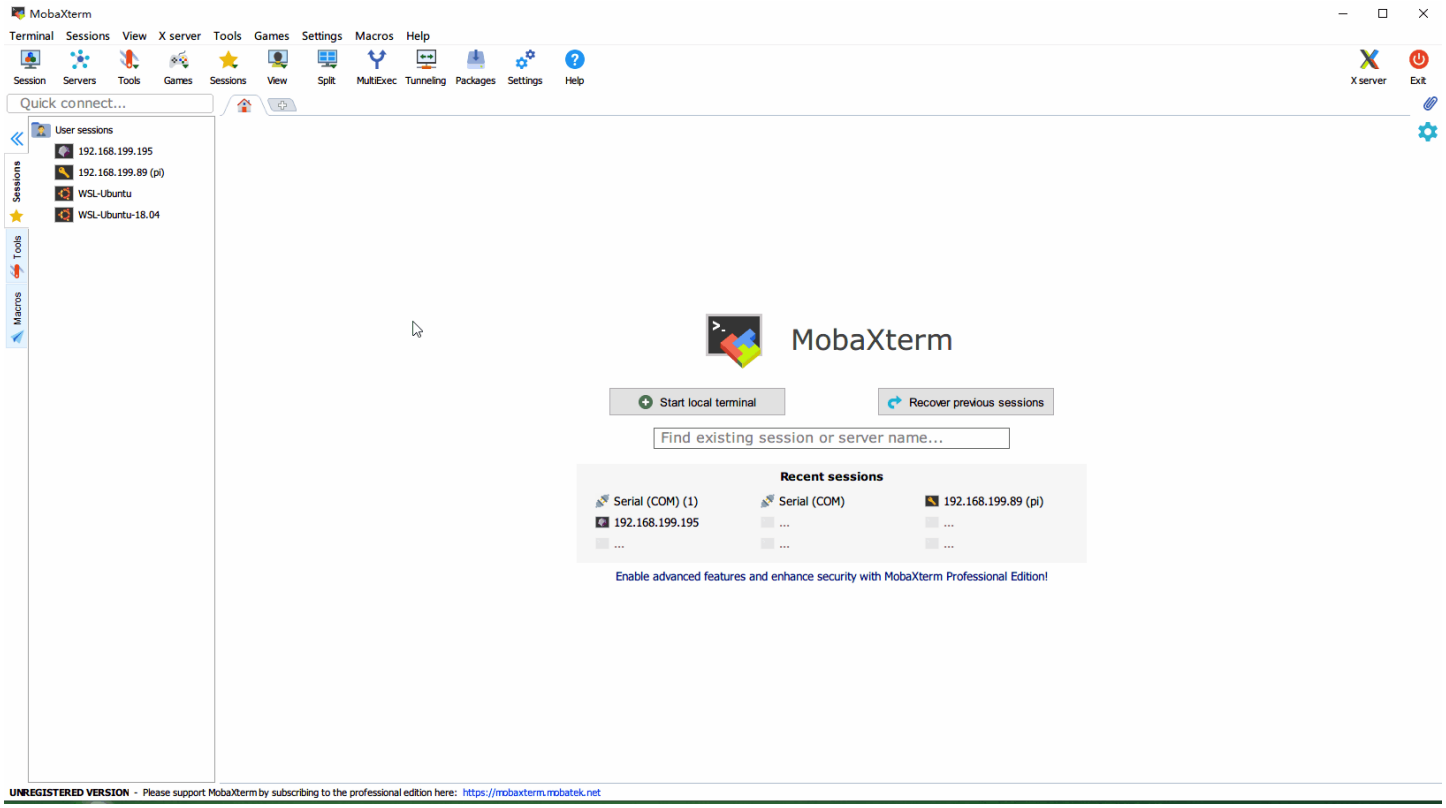
2. SD 0101

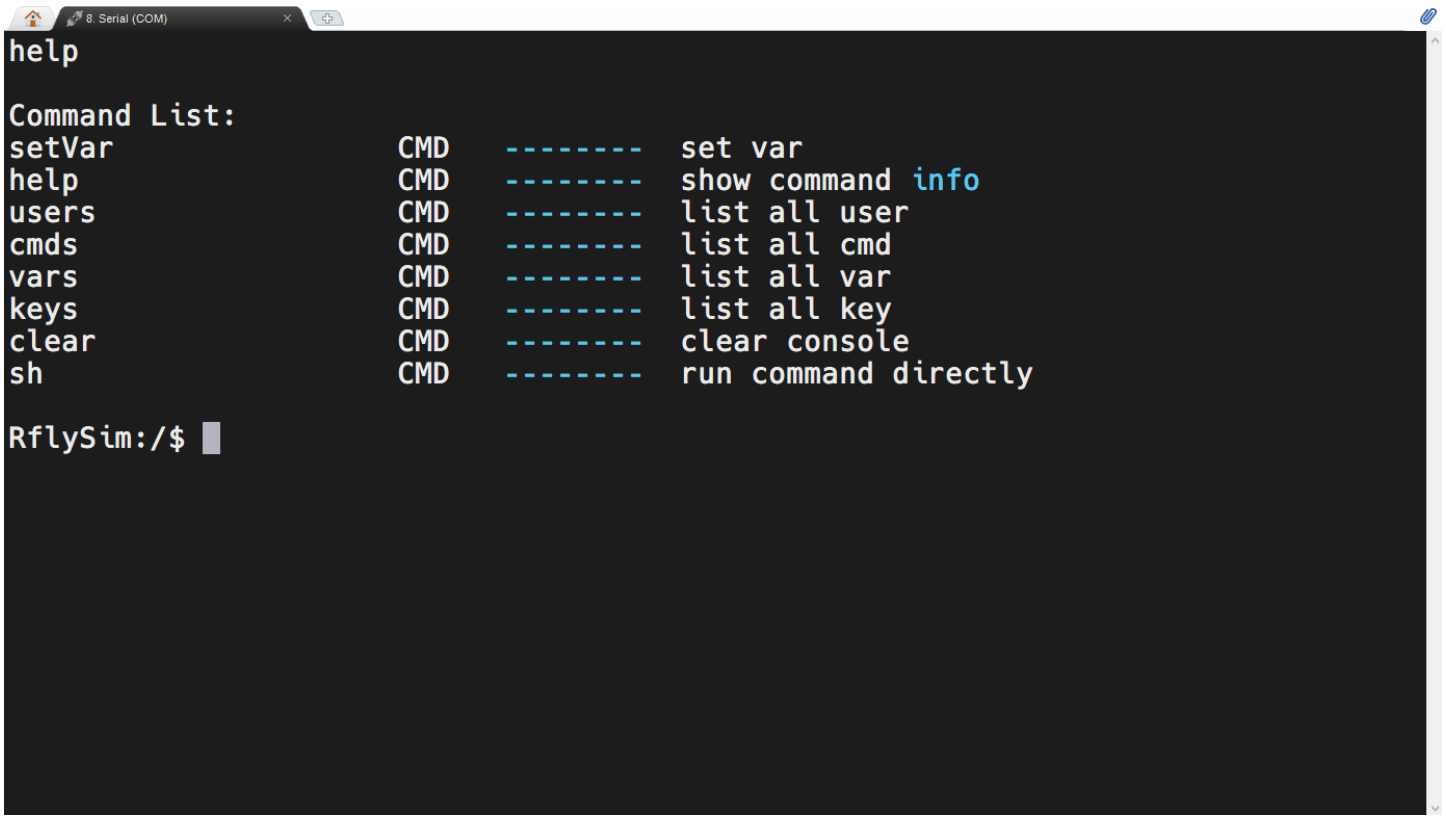
3. SD

USB **CP210X**

端口 (COM 和 LPT)
Silicon Labs CP210x USB to UART Bridge (COM16)

MobaXterm 115200 **help**

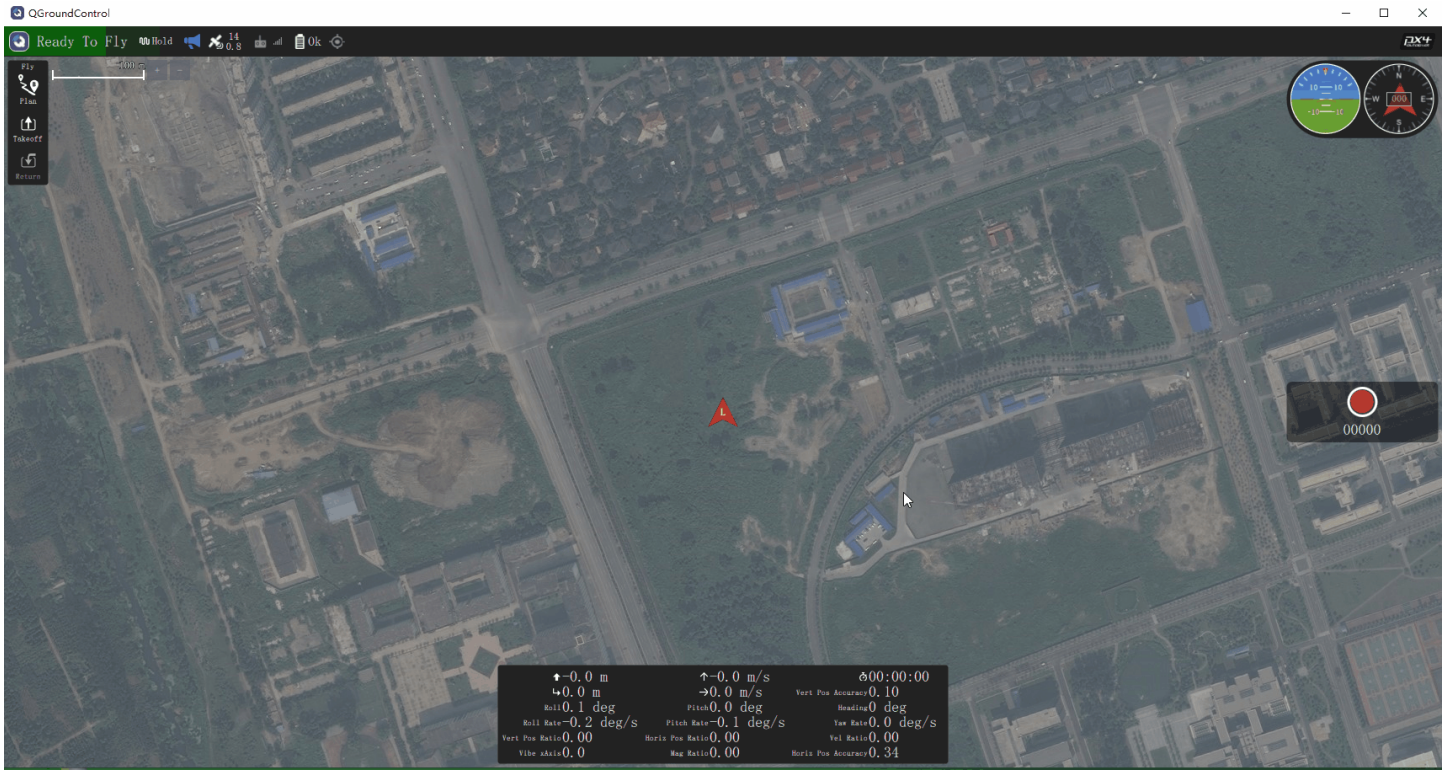




wifi RflySim3D wifi

2.1.5 5.

QGC . `sensors status`



FPGA

```
NuttShell (NSH)
nsh> sensors status
INFO  [sensors] selected gyro: 3932202 (0)
INFO  [data_validator] validator: best: 0, prev best: 0, failsafe: NO (0 events)
INFO  [data_validator] sensor #0, prio: 100, state: OK
INFO  [data_validator]     val:  -0.0103, lp:  -0.0001 mean dev:  -0.0000 RMS:   0.0060 conf:   1.0000
INFO  [data_validator]     val:   0.0043, lp:  -0.0002 mean dev:   0.0001 RMS:   0.0061 conf:   1.0000
INFO  [data_validator]     val:   0.0048, lp:   0.0001 mean dev:  -0.0000 RMS:   0.0060 conf:   1.0000

INFO  [sensors] selected accel: 3932202 (0)
INFO  [data_validator] validator: best: 0, prev best: 0, failsafe: NO (0 events)
INFO  [data_validator] sensor #0, prio: 100, state: OK
INFO  [data_validator]     val:   0.0600, lp:   0.0050 mean dev:   0.0005 RMS:   0.1177 conf:   1.0000
INFO  [data_validator]     val:   0.1085, lp:   0.0060 mean dev:   0.0009 RMS:   0.1187 conf:   1.0000
INFO  [data_validator]     val:  -9.8621, lp:  -9.7857 mean dev:   0.0004 RMS:   0.1202 conf:   1.0000

INFO  [sensors] selected mag: 396809 (0)
INFO  [data_validator] validator: best: 0, prev best: 0, failsafe: NO (0 events)
INFO  [data_validator] sensor #0, prio: 125, state: OK
INFO  [data_validator]     val:   0.2790, lp:   0.2767 mean dev:  -0.0000 RMS:   0.0016 conf:   1.0000
INFO  [data_validator]     val:  -0.0360, lp:  -0.0350 mean dev:  -0.0000 RMS:   0.0016 conf:   1.0000
INFO  [data_validator]     val:   0.4736, lp:   0.4723 mean dev:  -0.0000 RMS:   0.0016 conf:   1.0000

INFO  [vehicle_air_data] selected barometer: 4027937 (0)
INFO  [data_validator] validator: best: 0, prev best: 0, failsafe: NO (0 events)
INFO  [data_validator] sensor #0, prio: 75, state: OK
INFO  [data_validator]     val: 101325.0000, lp: 101324.5313 mean dev:  -0.0874 RMS:   0.6128 conf:   1.0000
INFO  [data_validator]     val:  24.9900, lp:  24.9900 mean dev:   0.0000 RMS:   0.0000 conf:   1.0000
INFO  [data_validator]     val:   0.0000, lp:   0.0000 mean dev:   0.0000 RMS:   0.0000 conf:   1.0000

INFO  [sensors] Airspeed status:
INFO  [data_validator]     no data

INFO  [vehicle_acceleration] selected sensor: 3932202 (0), rate: 396.9 Hz
INFO  [vehicle_acceleration] estimated bias: [0.0000 0.0000 0.0000]
INFO  [sensor_calibration] ACC 3932202 EN: 1, offset: [-0.0048 0.0039 -0.0002] scale: [1.0002 0.9999 0.9990]

INFO  [vehicle_angular_velocity] selected sensor: 3932202 (0), rate: 396.9 Hz
INFO  [vehicle_angular_velocity] estimated bias: [0.0000 0.0000 0.0000]
INFO  [sensor_calibration] GYRO 3932202 EN: 1, offset: [-0.0004 0.0001 -0.0006]


INFO  [vehicle_imu] IMU ID: 3932202, accel interval: 2512.7 us, gyro interval: 2512.7 us
vehicle_imu: accel data gap: 1 events
vehicle_imu: gyro data gap: 1 events
vehicle_imu: accel update interval: 24206 events, 2518.92us avg, min 2509us max 3495us 69.303us rms
vehicle_imu: gyro update interval: 24208 events, 2518.92us avg, min 2509us max 3495us 69.300us rms
INFO  [sensor_calibration] ACC 3932202 EN: 1, offset: [-0.0048 0.0039 -0.0002] scale: [1.0002 0.9999 0.9990]
INFO  [sensor_calibration] GYRO 3932202 EN: 1, offset: [-0.0004 0.0001 -0.0006]
nsh>
```

- Px4 1.11.3
-
-
-

3.

3.1 WIFI

WIFI WIFI

WIFI 921600 UDP IP 20010

ATK-ESP8266

USB WIFI

Note

1 2 2

3.1.1 1

COM-SAT UDP WIFI

PC USB WIFI AT

Note

WIFI 115200

1. WIFI SAT

```
AT+CWMODE=1
```

2.

```
AT+RST
```

3. WIFI `<wifi-name>` `<password>`

```
AT+CWJAP="<wifi-name>","<password>"
```

4.

```
AT+CIPMUX=0
```

5. `192.168.XXX.XXX` IP `255.255.255.255`

```
AT+CIPSTART="UDP","192.168.XXX.XXX",14550
```

3.1.2 2

WIFI
RflySim3D 20010

Tip

- QGC 14550
- RflySim3D **20010**
- WIFI **921600**

1.

```
+++
```

2.

```
AT+CIPSTATUS
```

3. IP

```
AT+CIFSR
```

4. WIFI

```
AT+CWJAP="<wifi-name>","<password>"
```

5.

```
AT+SAVETRANSLINK=<mode>,<remote IP>,<remote port>,<type>,<TCP keep alive>,<UDP local port>
```

<mode> 0 1
<remote IP> IP "192.168.XXX.XXX"
<remote port>
<type> "TCP" "UDP"
<TCP keep alive>
<UDP local port> UDP

```
AT+SAVETRANSLINK=1,"192.168.XXX.XXX",20010,"UDP",20011
```

6.

```
AT+UART=921600,8,1,0,0
```

7.

```
AT+SAVETRANSLINK=0
```

3.1.3 3

ATK-ESP8266 WIFI　　_V1.3.pdf

3.2 PX4 1.12

1.12 1.11

1. IST8310
2. IST8310 FPGA

1.12

1.12 1.11 1.12 rc.board_sensors Pixhawk 4

```
#!/bin/sh
#
# PX4 FMUv5 specific board sensors init
#------------------------------------------------------------------------------

board_adc start

if ! icm20689 -S start
then
# Internal SPI bus ICM-20602
icm20602 -s -R 2 -q start

# Internal SPI bus ICM-20689
icm20689 -s -R 2 start

# Internal SPI bus BMI055 accel/gyro
bmi055 -A -R 2 -s start
bmi055 -G -R 2 -s start
fi

if ! ms5611 -X start
then
# Baro on internal SPI
ms5611 -s start
fi

if ! ist8310 -X -R 10 start
then
# internal compass
ist8310 -I -R 10 start

# External compass on GPS1/I2C1 (the 3rd external bus): standard Holybro Pixhawk 4 or CUAV V5 GPS/compass puck (with light
ist8310 -X -b 1 -R 10 start
fi
```

3.3

RflySim-RT param

3.3.1

```
param
```

param

param

param list
param listfile
param save

param save <filename>

param load
param load <filename>
param set <index> <value>
param set <index> <value_index> <value>

param set <filename>

param

3.3.2

- **param list**

```
RflySim:/$ param list
FrameConfigPath is: H250.json
SensorConfigPath is: SensorConfig.json
EnvConfigPath is: EnvConfig.json
[0]Mass is:  0.752000
[1]C_md is:  0.000100  0.000100  0.000060
[2]J is:
    0.005600  0.000000  0.000000
    0.000000  0.005600  0.000000
    0.000000  0.000000  0.010400
[3]motorCr is:  0.000000
[4]motorFitType is:  2.000000
[5]motorJm is:  8.849300
[6]motorMinThr is:  0.148000
[7]motorRateCurveCoeffi is:  -2143.000000  5113.000000  -458.400000
[8]motorTc is:  0.032000
[9]motorWb is:  0.000000
[10]rotorCt is:  1.345000
[11]NoiseVarAcc0 is:  0.000100  0.000100  0.001000
[12]NoiseVarGyro0 is:  0.000050  0.000050  0.000050
[13]NoiseVarMag0 is:  0.000002  0.000002  0.000002
[14]PositionAcc0 is:  0.000000  0.000000  0.000000
[15]DisplayUAVType is:  3.000000
[16]CopterID is:  6031.000000
[17]RotorDirection is:  1.000000  1.000000  -1.000000  -1.000000  0.000000  0.000000  0.000000  0.000000
[18]EfficiencyMatrix is:
    0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000
    0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000
    -1.000000  -1.000000  -1.000000  -1.000000  -1.000000  -1.000000  -1.000000  -1.000000
    -0.088400  0.088400  0.088400  -0.088400  -0.000000  -0.000000  -0.000000  -0.000000
    0.088400  -0.088400  0.088400  -0.088400  0.000000  0.000000  0.000000  0.000000
    0.016600  0.016600  -0.016600  -0.016600  0.000000  0.000000  0.000000  0.000000
[19]BoardRotation is:  0.000000  0.000000  45.000000
[20]IST8310_ConvertRatio is:  1320.000000
[21]Using_OneShot is:  1.000000
```

```
FrameConfigPath is: H250.json
```

`H250.json`

- **param listfile**

SD

```
RflySim:/$ param listfile
: System Volume Information
: Config.txt
: MainConfig.json
: FrameConfig.json
: BOOT.BIN
: H250.json
: F450.json
: F550.json
: octo.json
: backup
: modulocto.json
no more file
```

MainConfig.json SD

- **param save**

  param list

```
RflySim:/$ param save
File <H250.json> Save Successfully.
```

- **param save <filename>**

  - <filename> `.json`

```
RflySim:/$ param save H250.json
File <H250.json> Save Successfully.
File <H250.json> is updated to FrameConfigPath Successfully.
```

- **param load**

```
RflySim:/$ param load
File <H250.json> Load Successfully.
```

- **param load <filename>**

  - <filename> `.json`

```
RflySim:/$ param load H250.json
File <H250.json> Load Successfully.
File <H250.json> is updated to FrameConfigPath Successfully.
```

- **param set <index> <value>**

  param save param save <filename>

  - <index> param list
  - <value>

- **param set <index> <value_index> <value>**

  param set <index> <value>

  - <index> param list
  - <value_index> value_index 3x3 value_index 0-8

    0 3 6
    1 4 7
    2 5 8

    0
  - <value>

```
param set 2 3 1
```

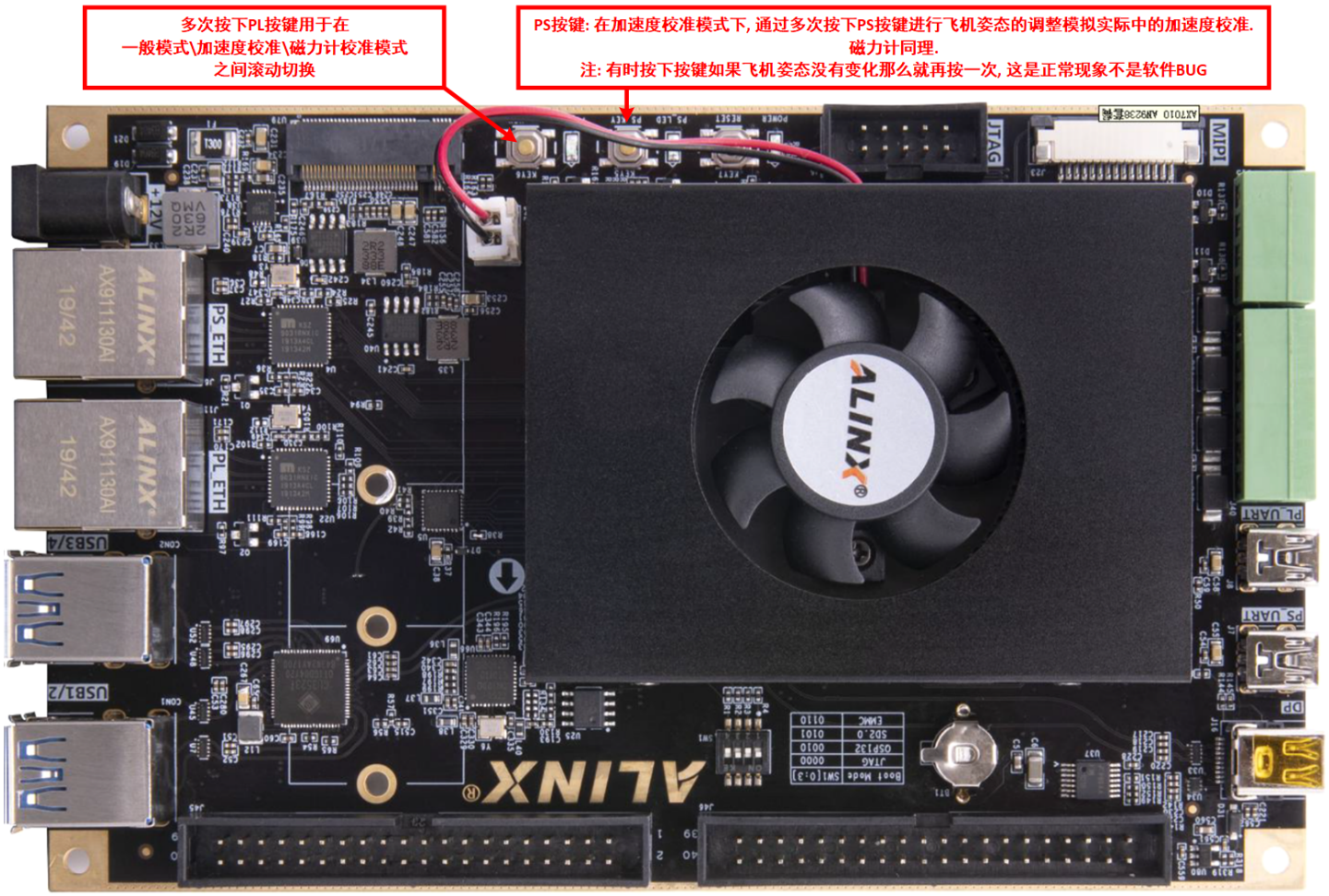
2 3 1

- **param set <filename>**
  - <filename> `.json`

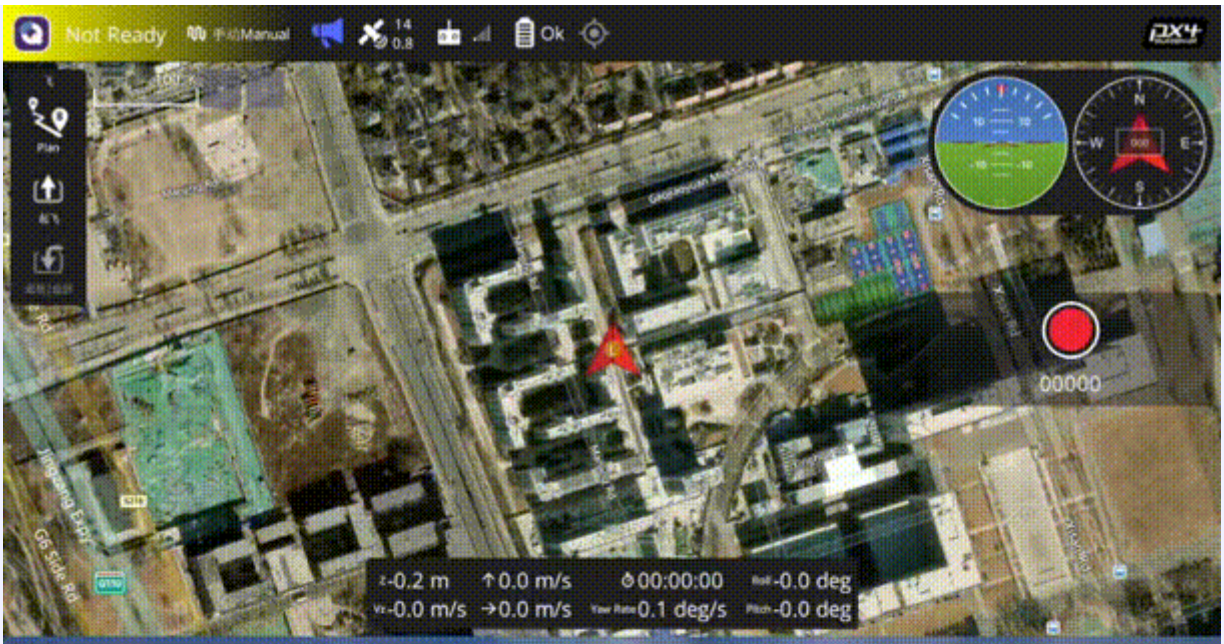
3.4

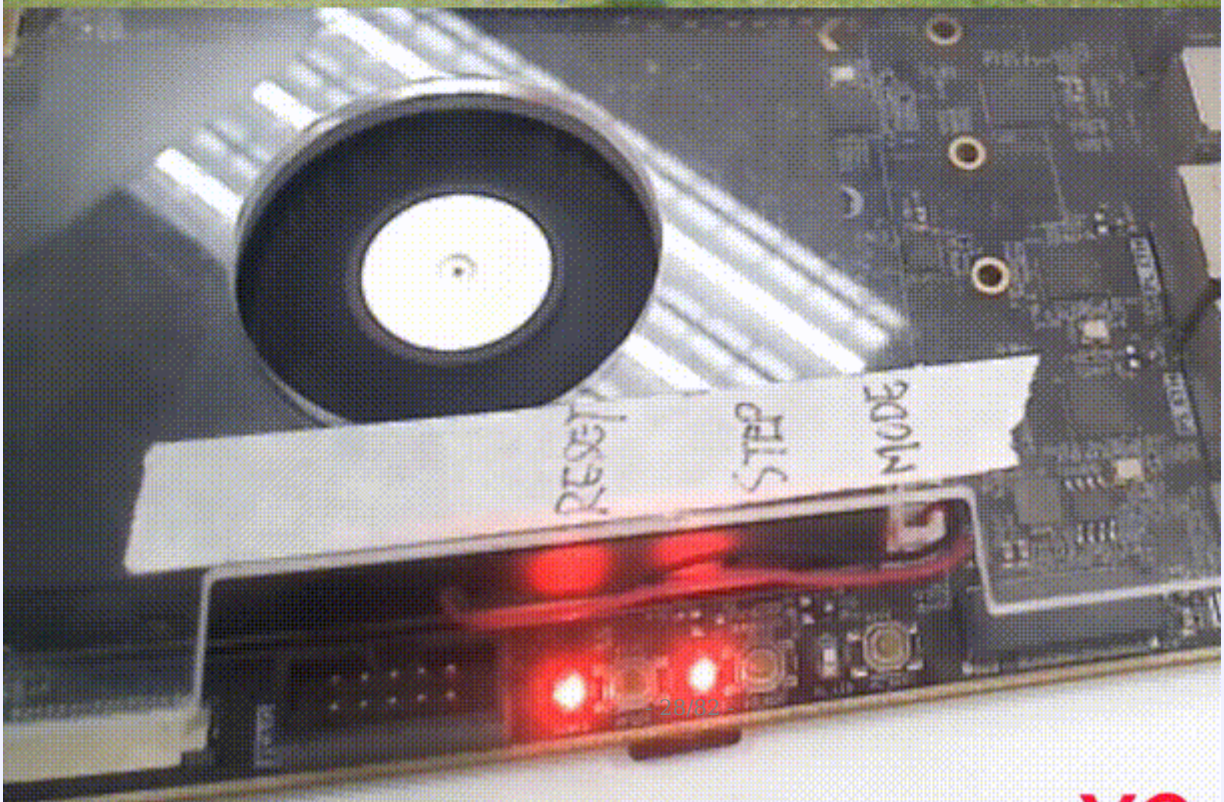
PX4

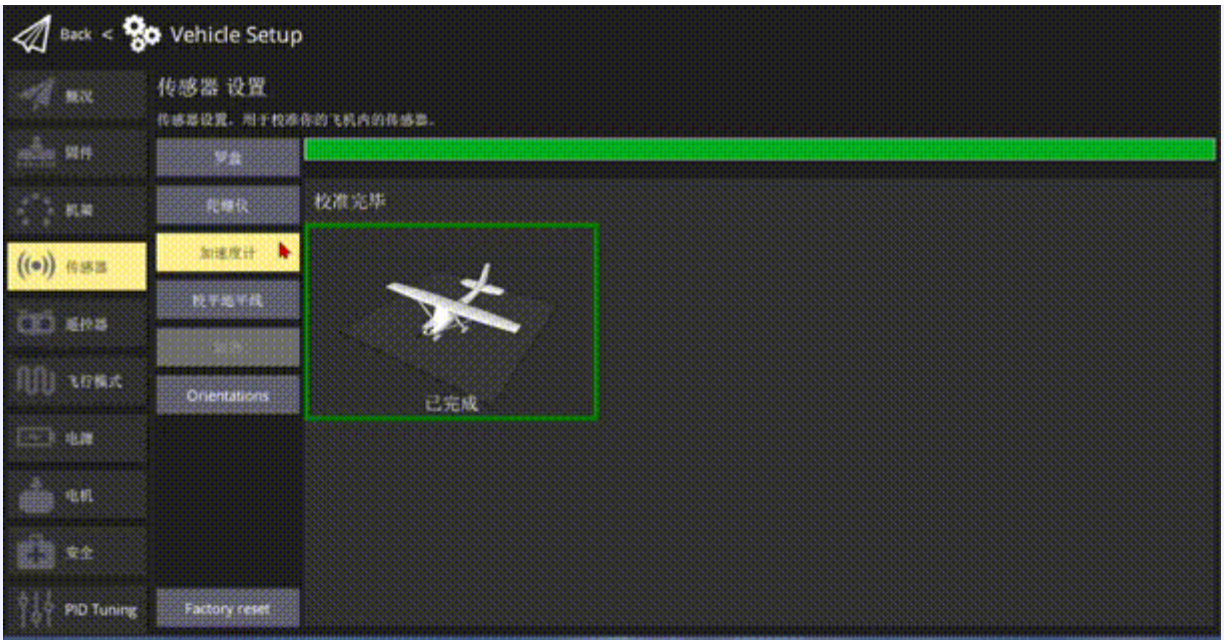
校准按键说明示意图

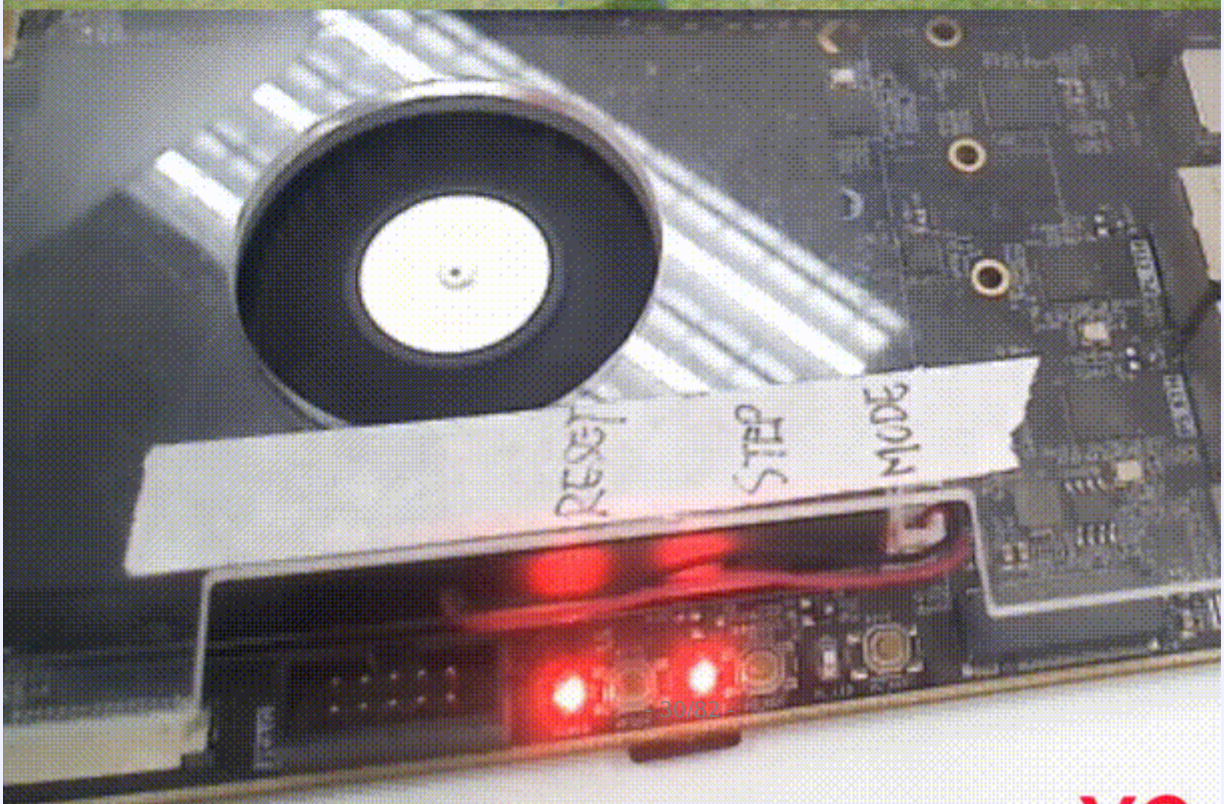


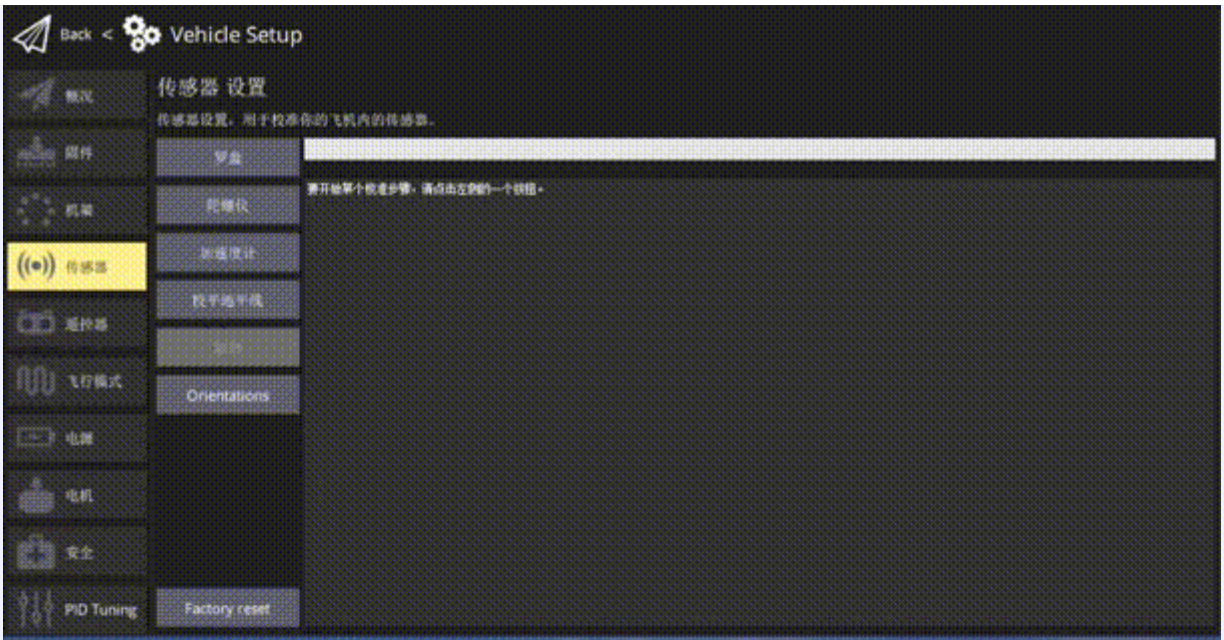
- QGC RflySim3D
- QGC


Not Ready
Manual
Ok
Plan
00000
z -0.2 m
↑0.0 m/s
00:00:00
Roll -0.0 deg
Vx -0.0 m/s
→0.0 m/s
Yaw Rate 0.1 deg/s
Pitch -0.0 deg

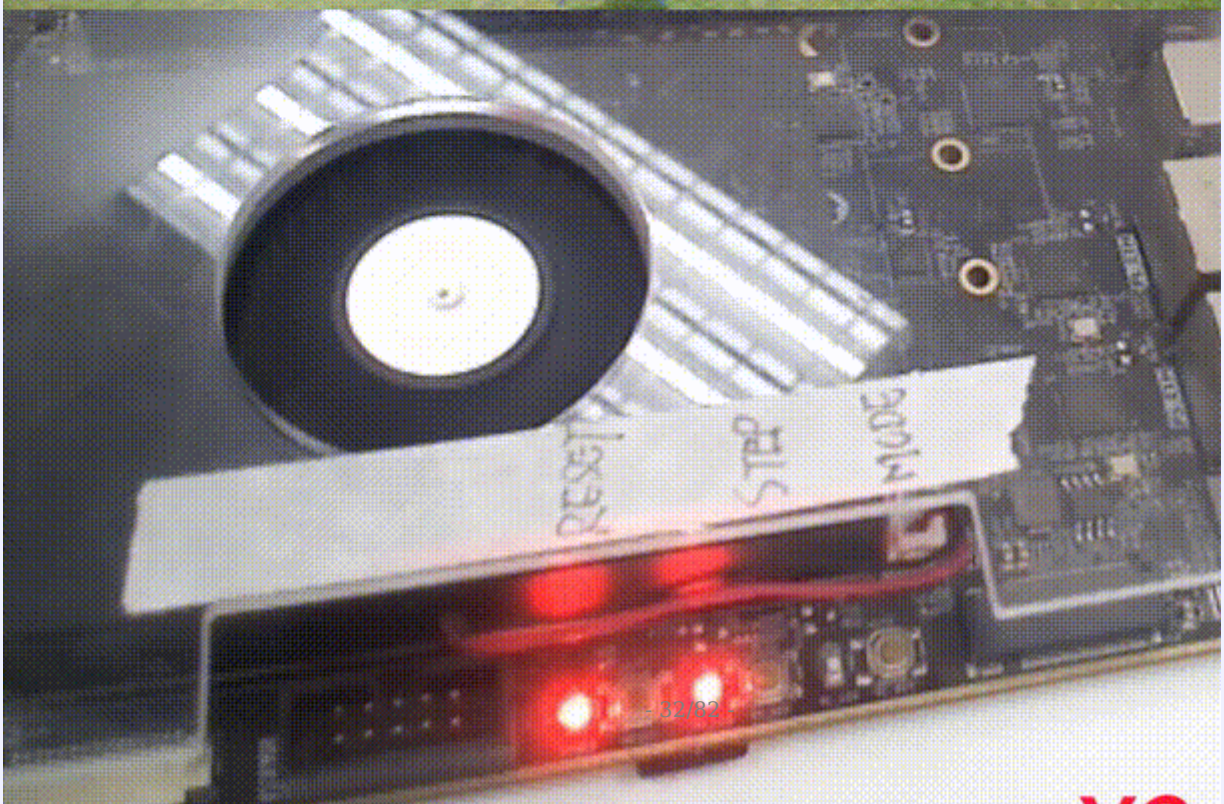

RESET
SAFE
MODE


Back
Vehicle Setup
传感器 设置
传感器设置，用于校准你的飞机内的传感器。
罗盘
陀螺仪
加速度计
校平地平线
Orientations
校准完毕
已完成
机架
传感器
遥控器
飞行模式
电源
电机
安全
PID Tuning
Factory reset


RESET
SIG
MODE


Back
Vehicle Setup
传感器 设置
传感器设置，用于校准你的飞机内的传感器。
罗盘
陀螺仪
加速度计
校平地平线
Orientations
电调
电机
安全
PID Tuning
Factory reset


Vehicle solo of ID 6021 with style 3
Timestamp (s): 41.8745
Position xyz: 0.000, 0.000, 0.000
Euler angles: 0.000, -0.000, 0.000
Velocity xyz: 0.000, 0.000, 0.000
Angular rate: 0.000, 0.000, 0.000
Acceleration xyz: 0.000, 0.000, 0.000
GPS Pos: 40.15403020, 116.25936830, -0.00000000
Actuator output 1 - 8: 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00


RESET
STEP
MODE

4.

4.1

4.1.1 1.

Windows 10/11

MATLAB 2021a Vivado
HDL Coder Support Package for Xilinx Zynq Platform
HDL Coder Support Package for Xilinx FPGA Boards
Embedded Coder Support Package for Xilinx Zynq Platform
Embedded Coder Support Package for ARM Cortex-A Processors
Xilinx Unified 2020.1 (Vitis Vivado)
Visual Studio 2017 (VS2017)

1.1 MATLAB 2021a

MATLAB 2021a Vivado ____

Warning

FPGA IP CORE

1. __ `attachment_2656440_2022-01-07.zip`
2. MATLAB
3. `R2021a` `IPEmitterVivado.p` `<MATLAB>\toolbox\hdlcoder\hdlcommon\+hdlturnkey\+ip\IPEmitterVivado.p` `<MATLAB>` MATLAB 2021a

1.2

1. `C:\Users\<username>\Downloads\MathWorks\SupportPackages\R2021a` <username>
2. MATLAB `<MATLAB>\bin\win64` `SupportSoftwareInstaller.exe`
3.

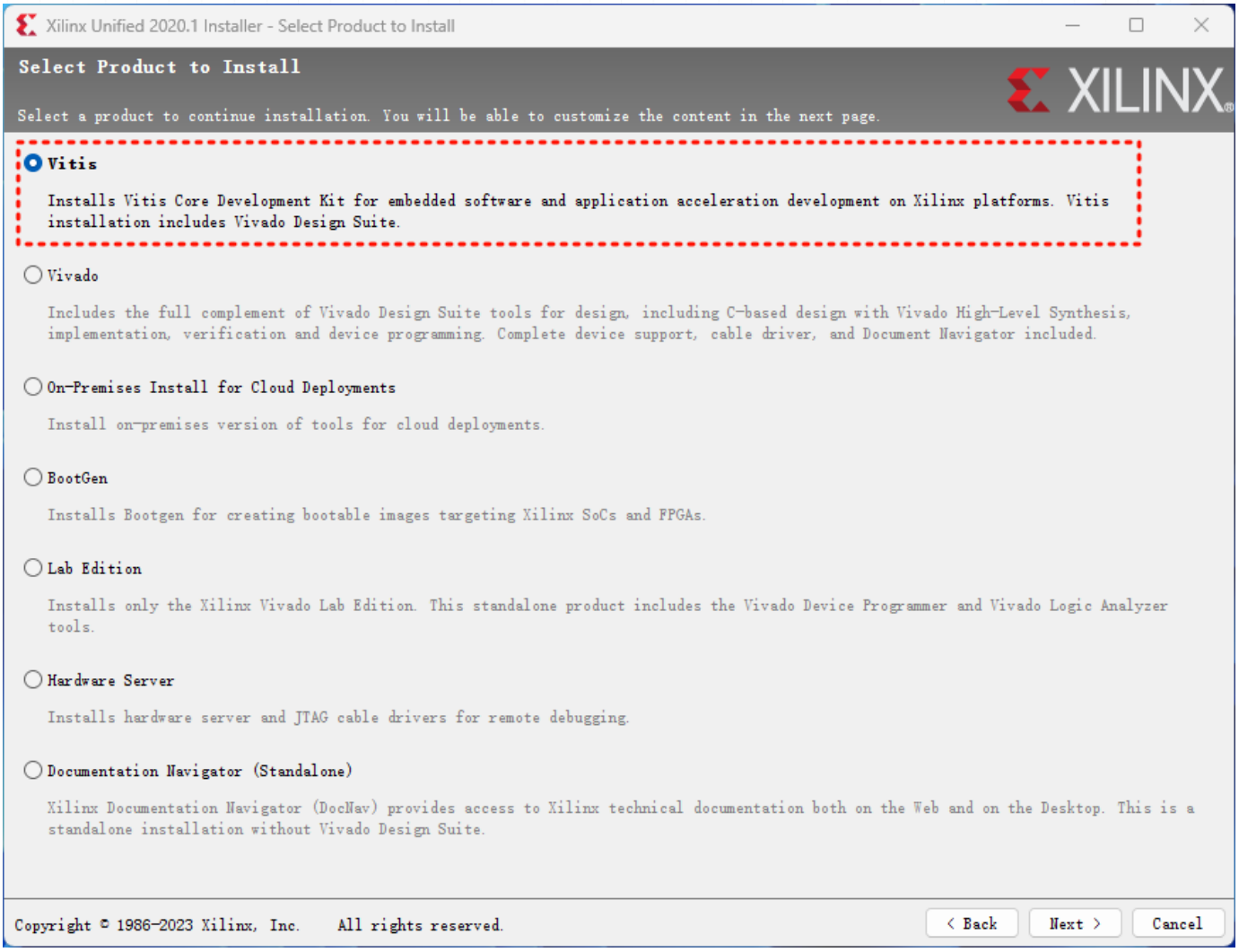
Note

- Embedded Coder Support Package for Xilinx Zynq Platform
- Embedded Coder Support Package for ARM Cortex-A Processors Linaro Toolchain v4.8

1.3 Xilinx Unified 2020.1

https://china.xilinx.com/member/forms/download/xef.html?filename=Xilinx_Unified_2020.1_0602_1208.tar.gz

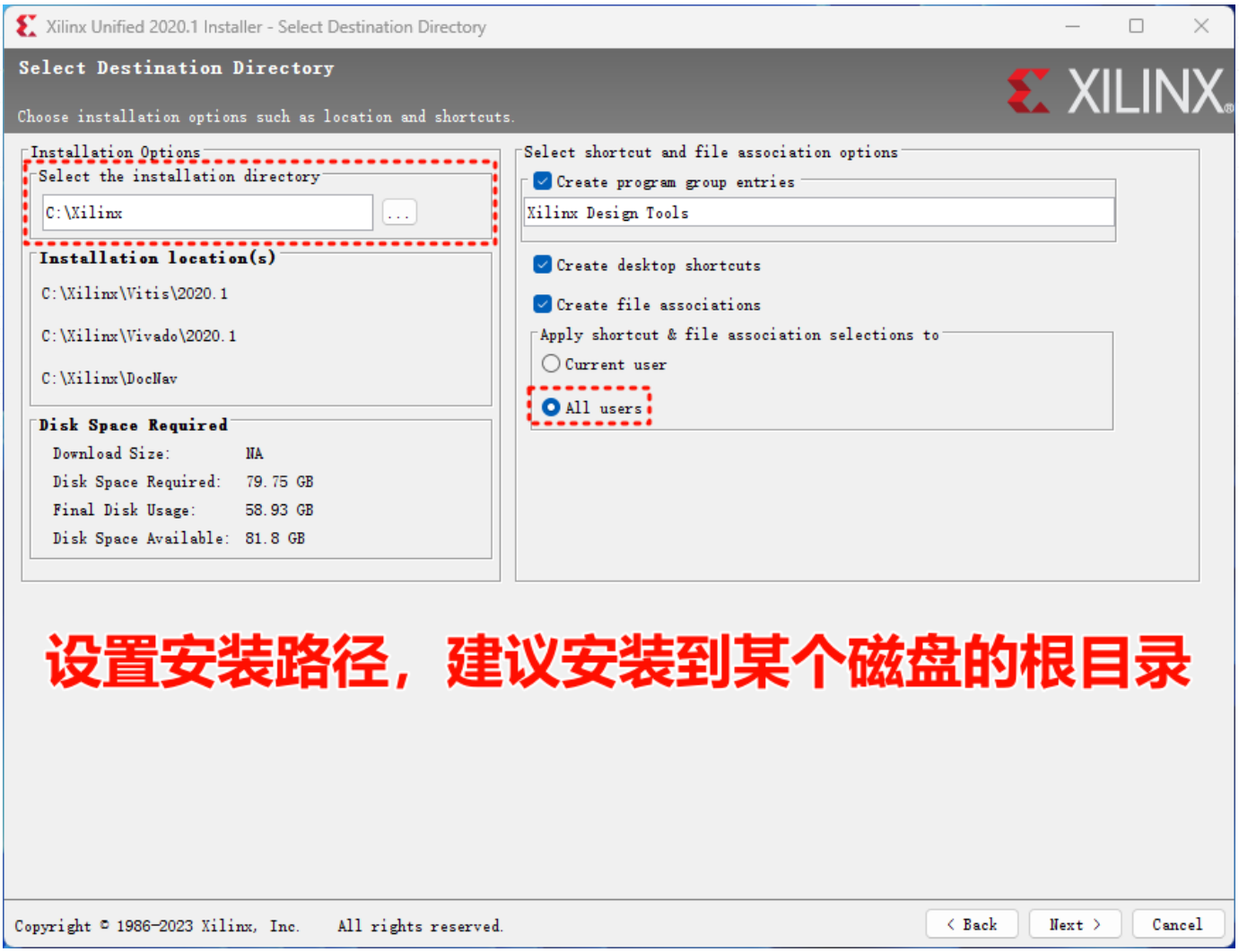
- **1. Vitis**



- **2.**


Xilinx Unified 2020.1 Installer - Vitis Unified Software Platform
Vitis Unified Software Platform
Customize your installation by (de)selecting items in the tree below. Moving cursor over selections below provide additional information.
XILINX
The Vitis unified software platform enables the development of embedded software and accelerated applications on heterogeneous Xilinx platforms including FPGAs, SoCs, and Versal ACAPs. It provides a unified programming model for accelerating Edge, Cloud, and Hybrid computing applications. This installation is a superset that includes the Vivado Design Suite as well. Users can optionally add Model Composer to this installation.
Design Tools
Vitis Unified Software Platform
DocNav
Devices
Install devices for Alveo and Xilinx edge acceleration platforms
Devices for Custom Platforms
SoCs
7 Series
UltraScale
UltraScale+
Engineering Sample Devices for Custom Platforms
Installation Options
Install Cable Drivers (You MUST disconnect all Xilinx Platform Cable USB II cables before proceeding)
Install WinPCap for Ethernet Hardware Co-simulation
Launch configuration manager to associate System Generator for DSP with MATLAB
保持默认即可
Download Size: NA
Disk Space Required: 79.75 GB
Reset to Defaults
Copyright © 1986-2023 Xilinx, Inc. All rights reserved.
< Back
Next >
Cancel

- **3.**



1.4 Visual Studio 2017

VS MALAB 2021a 2021 VS

vs_Community.exe

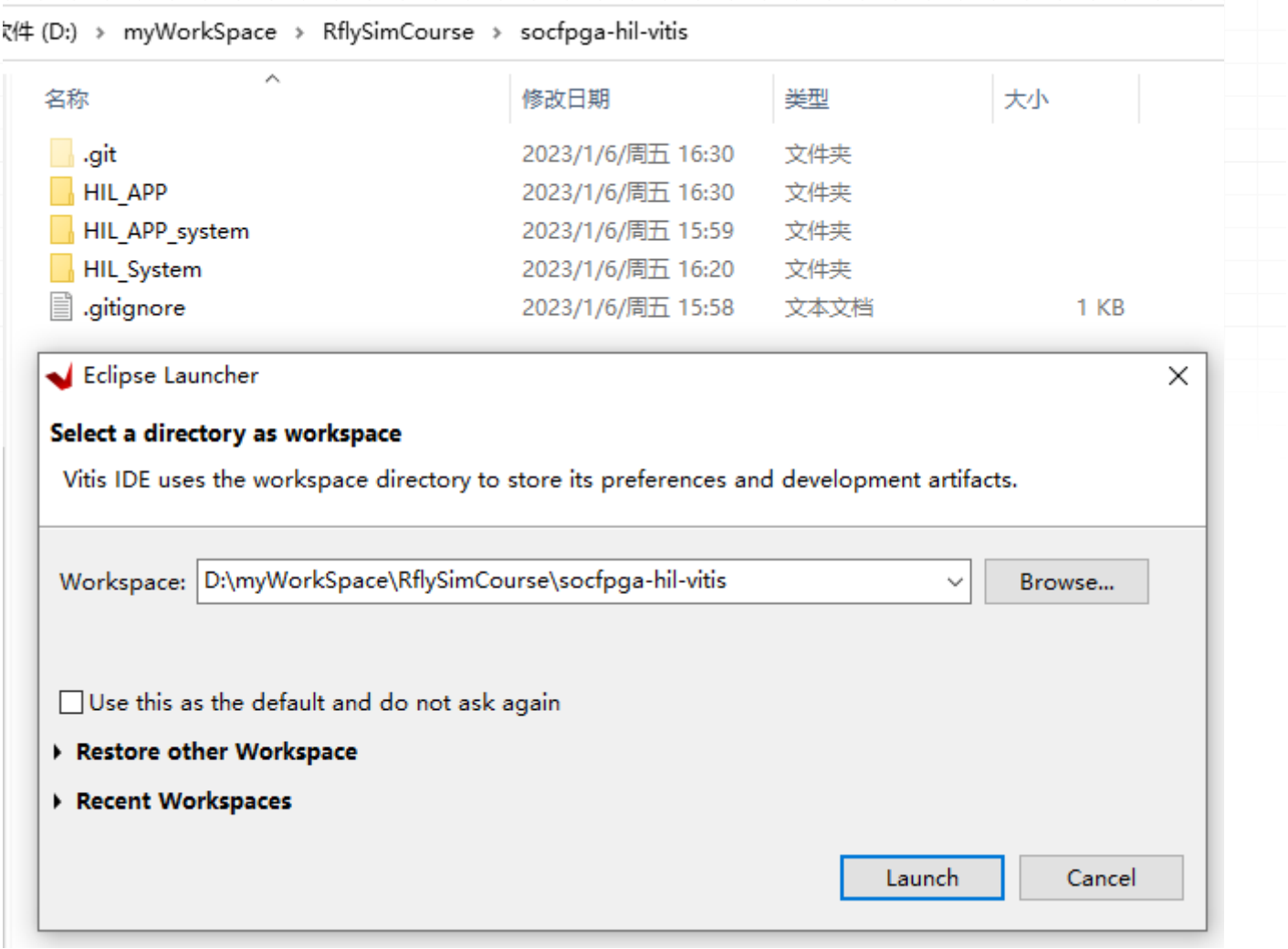
4.1.2 2. Vitis

git git

- **1.** **Vitis**

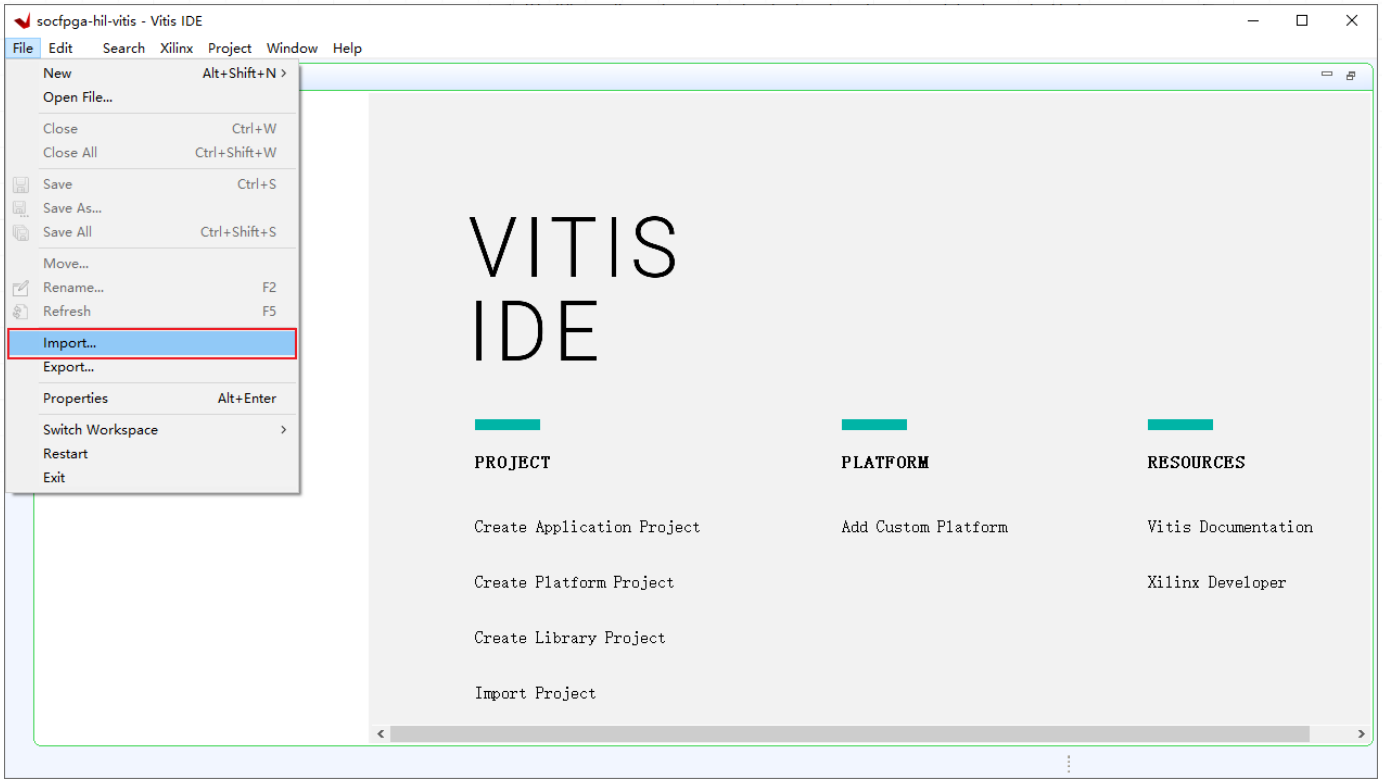
```
git clone https://gitee.com/RflyBUAA/rfly-sim-rt-vitis.git
```

Vitis

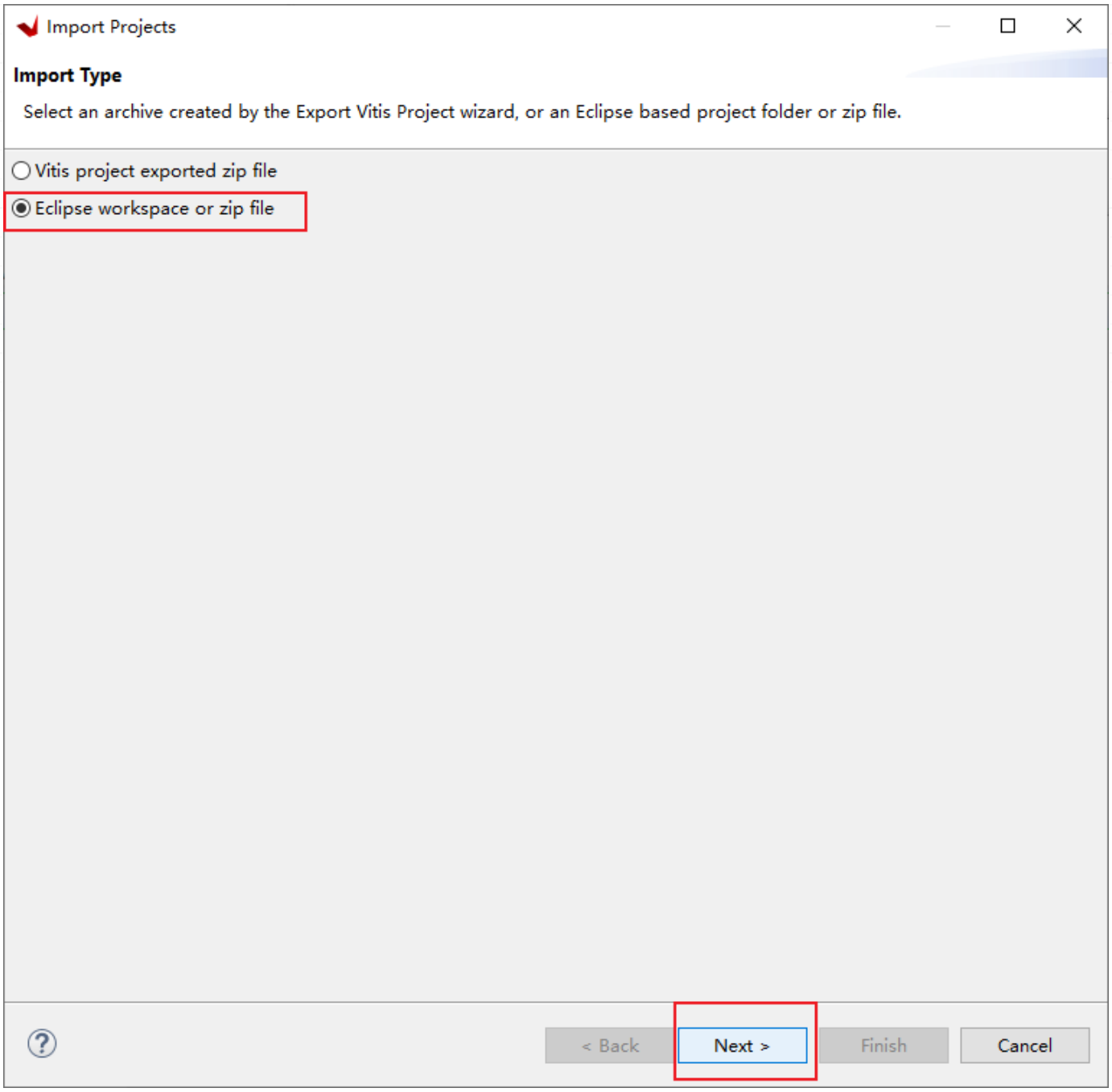

件 (D:) › myWorkSpace › RflySimCourse › socfpga-hil-vitis
名称
修改日期
类型
大小
.git
2023/1/6/周五 16:30
文件夹
HIL_APP
2023/1/6/周五 16:30
文件夹
HIL_APP_system
2023/1/6/周五 15:59
文件夹
HIL_System
2023/1/6/周五 16:20
文件夹
.gitignore
2023/1/6/周五 15:58
文本文档
1 KB
Eclipse Launcher
Select a directory as workspace
Vitis IDE uses the workspace directory to store its preferences and development artifacts.
Workspace:
D:\myWorkSpace\RflySimCourse\socfpga-hil-vitis
Browse...
Use this as the default and do not ask again
Restore other Workspace
Recent Workspaces
Launch
Cancel

- **2.**

Import...

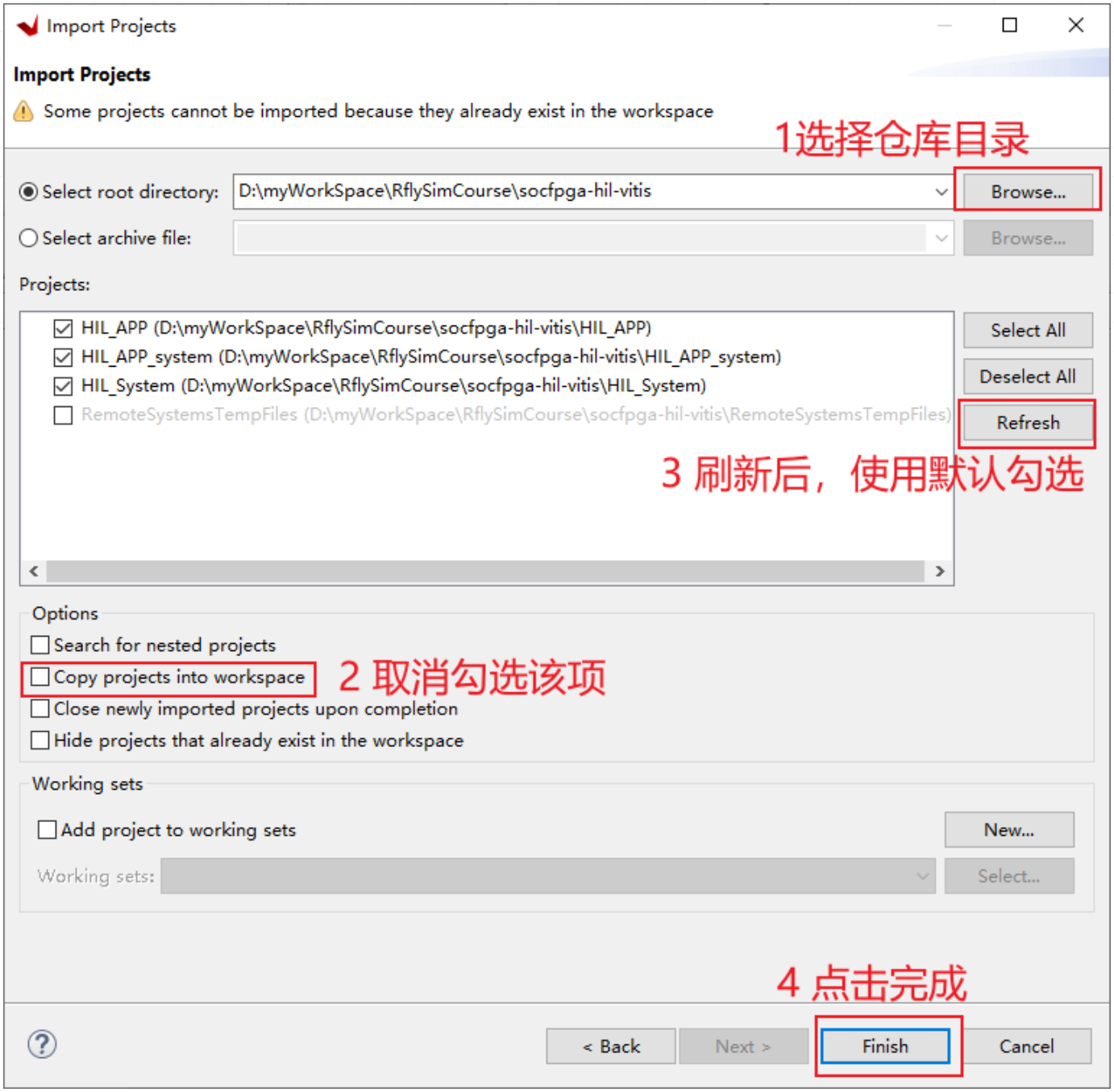


- **3. Eclipse**

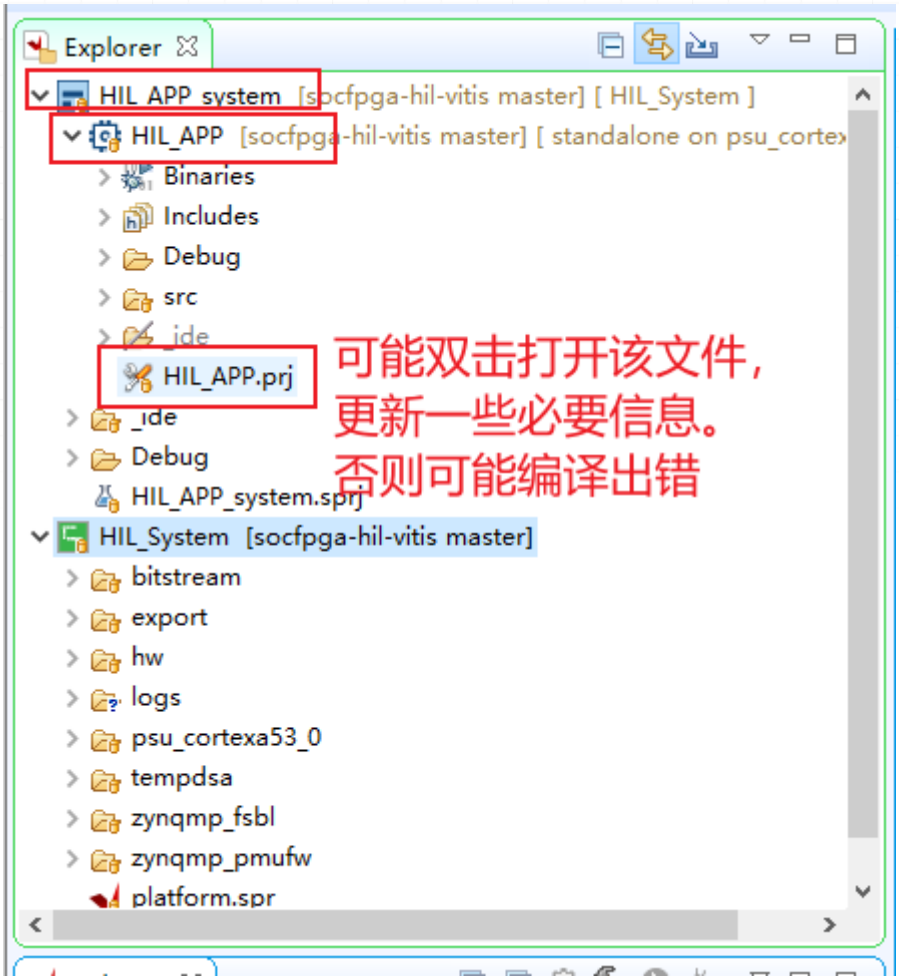

Import Projects
Import Type
Select an archive created by the Export Vitis Project wizard, or an Eclipse based project folder or zip file.
Vitis project exported zip file
Eclipse workspace or zip file
< Back
Next >
Finish
Cancel

- **4.**

clone

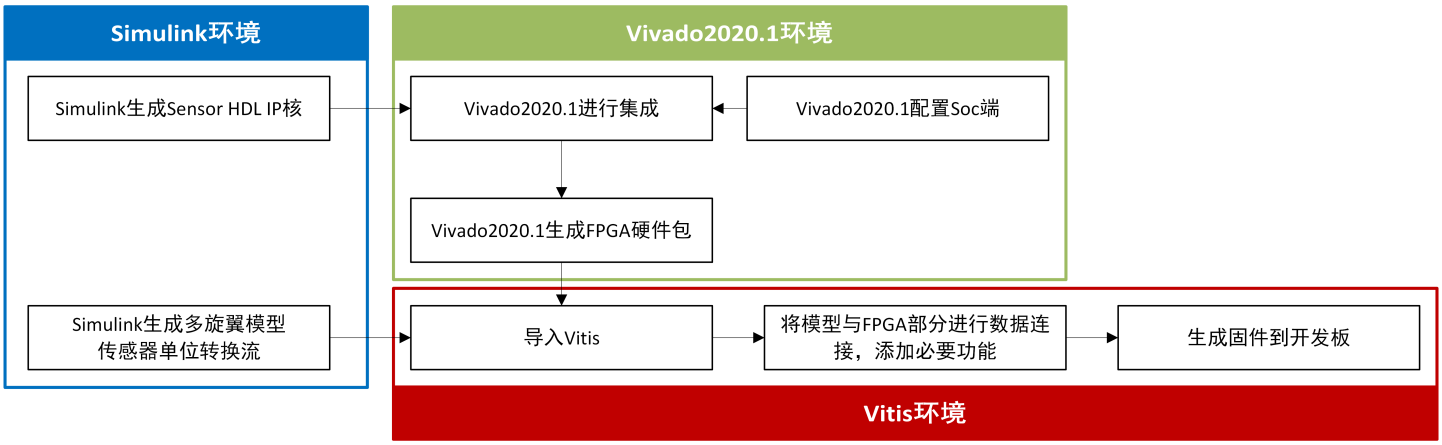

Import Projects
Import Projects
Some projects cannot be imported because they already exist in the workspace
1选择仓库目录
Select root directory:
D:\myWorkSpace\RflySimCourse\socfpga-hil-vitis
Browse...
Select archive file:
Browse...
Projects:
HIL_APP (D:\myWorkSpace\RflySimCourse\socfpga-hil-vitis\HIL_APP)
HIL_APP_system (D:\myWorkSpace\RflySimCourse\socfpga-hil-vitis\HIL_APP_system)
HIL_System (D:\myWorkSpace\RflySimCourse\socfpga-hil-vitis\HIL_System)
RemoteSystemsTempFiles (D:\myWorkSpace\RflySimCourse\socfpga-hil-vitis\RemoteSystemsTempFiles)
Select All
Deselect All
Refresh
3 刷新后，使用默认勾选
Options
Search for nested projects
Copy projects into workspace
2 取消勾选该项
Close newly imported projects upon completion
Hide projects that already exist in the workspace
Working sets
Add project to working sets
New...
Working sets:
Select...
4 点击完成
< Back
Next >
Finish
Cancel

- **5.**



Vitis

4.1.3 3.



Vitis Simulink Vivado

```
git clone https://gitee.com/RflyBUAA/socfpga-hil-platformv2.git
```

MATLAB/Simulink Vivado

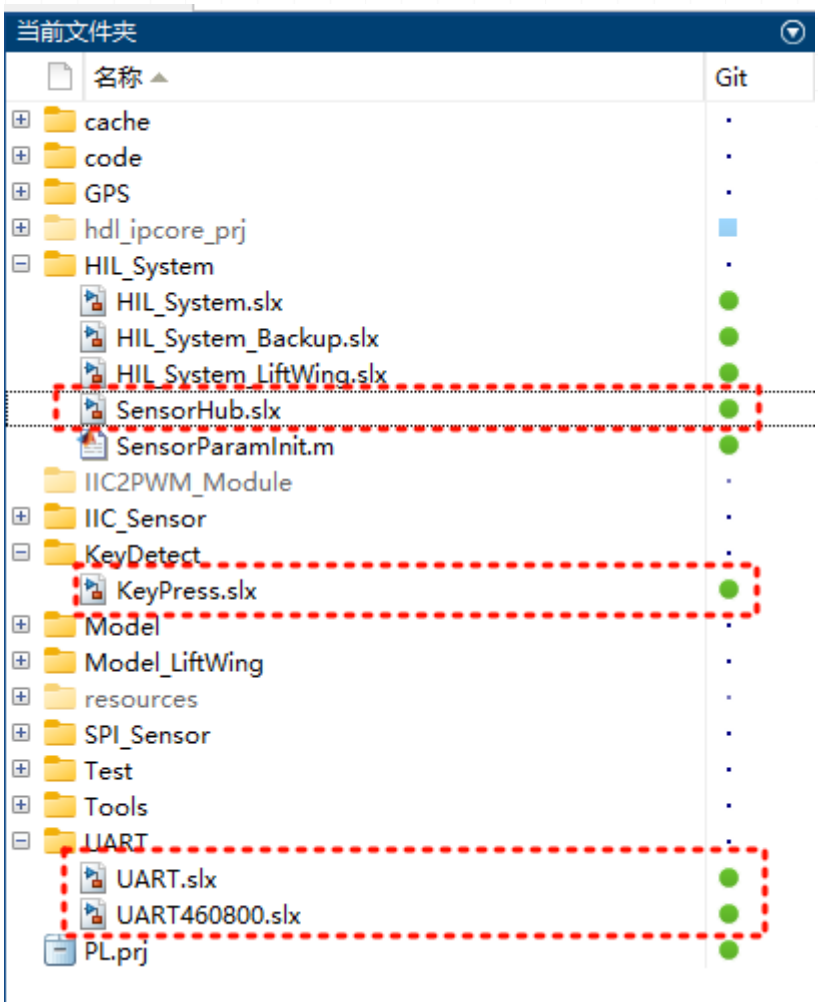
IP Vivado
PL MATLAB/Simulink
Packages

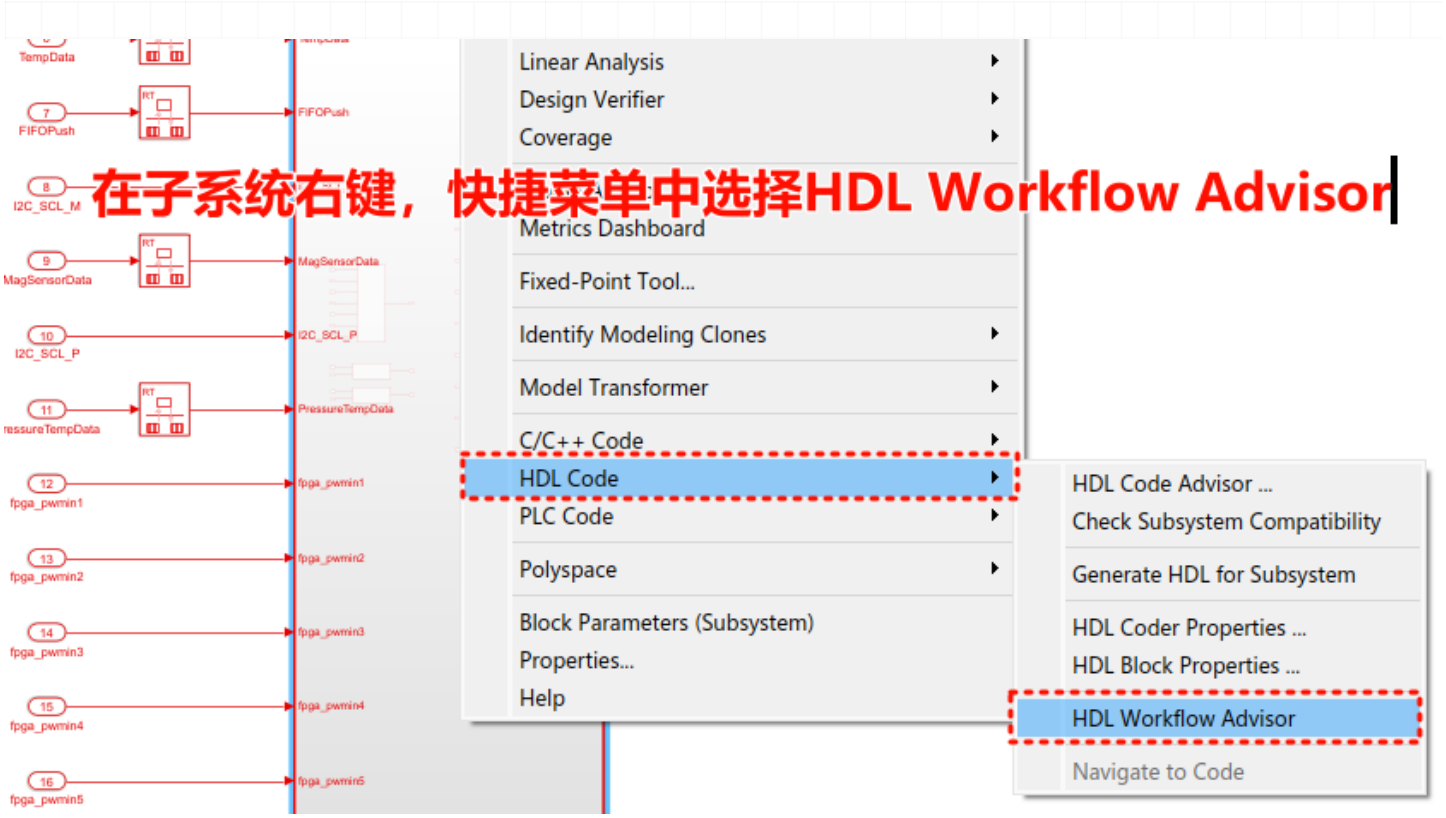
4.2 FPGA

MATLAB Vivado MATLAB/Simulink FPGA IP Core vivado vivado xsa

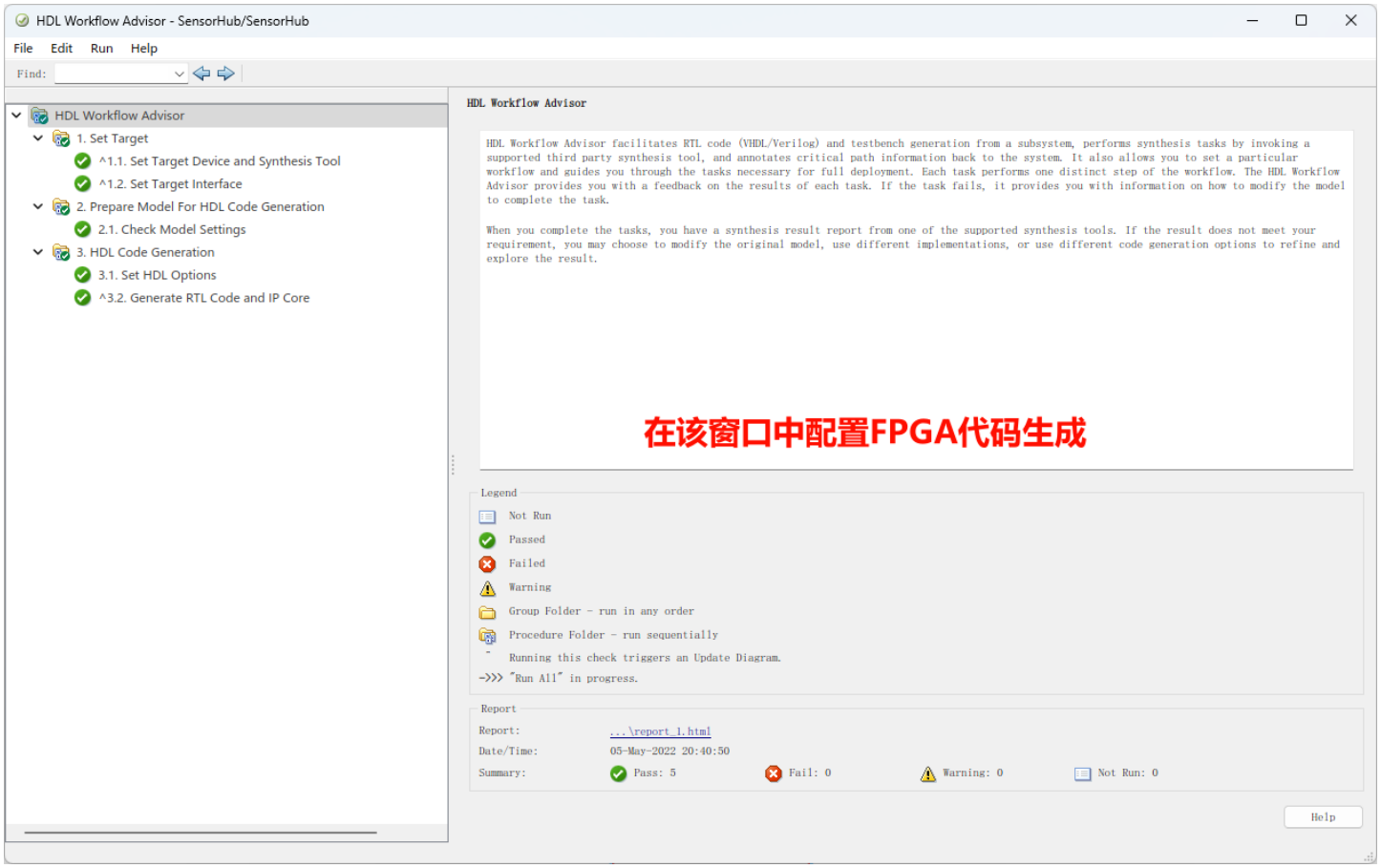
4.2.1 1. MATLAB FPGA

FPGA IP Core SensorHub.slx

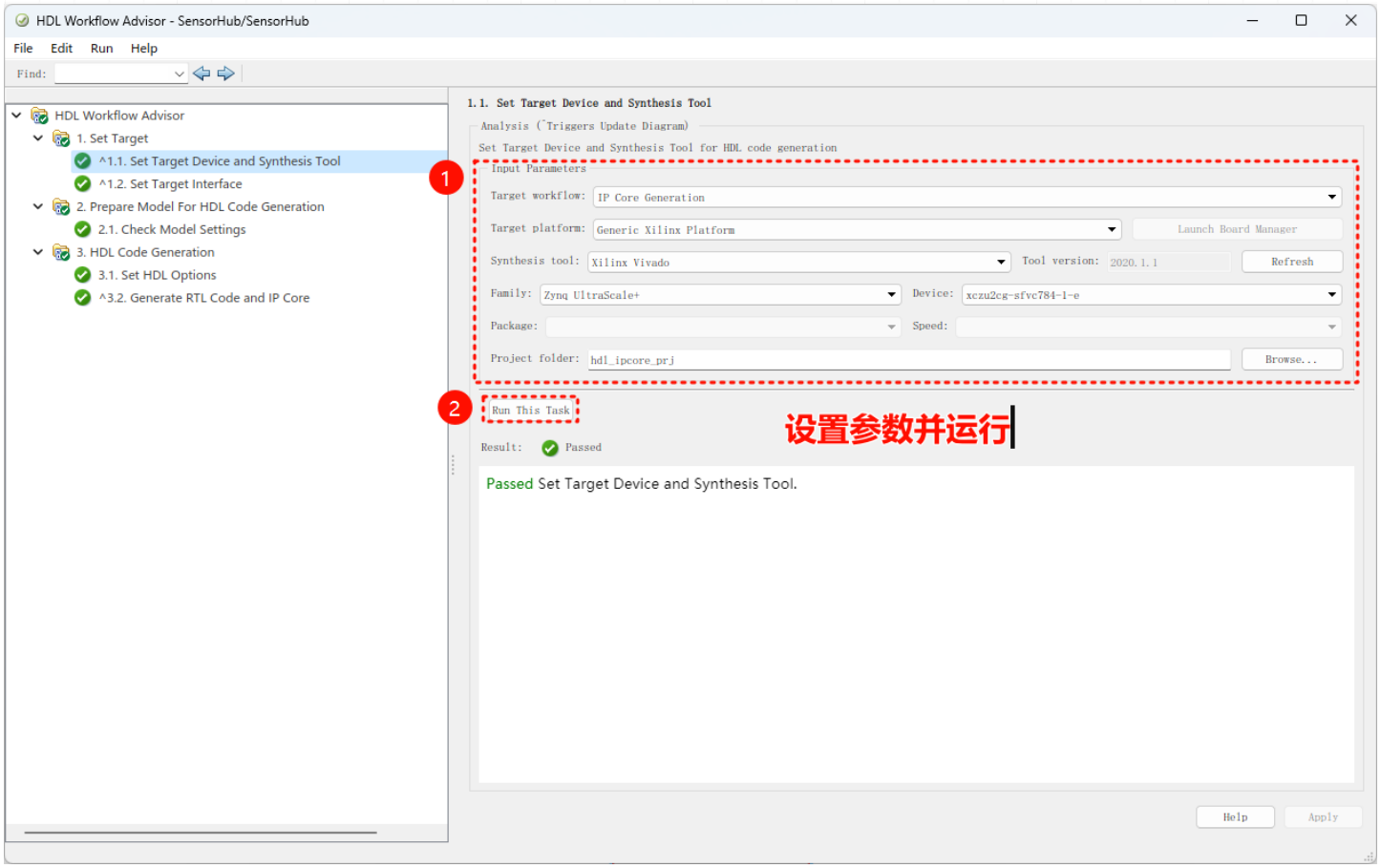


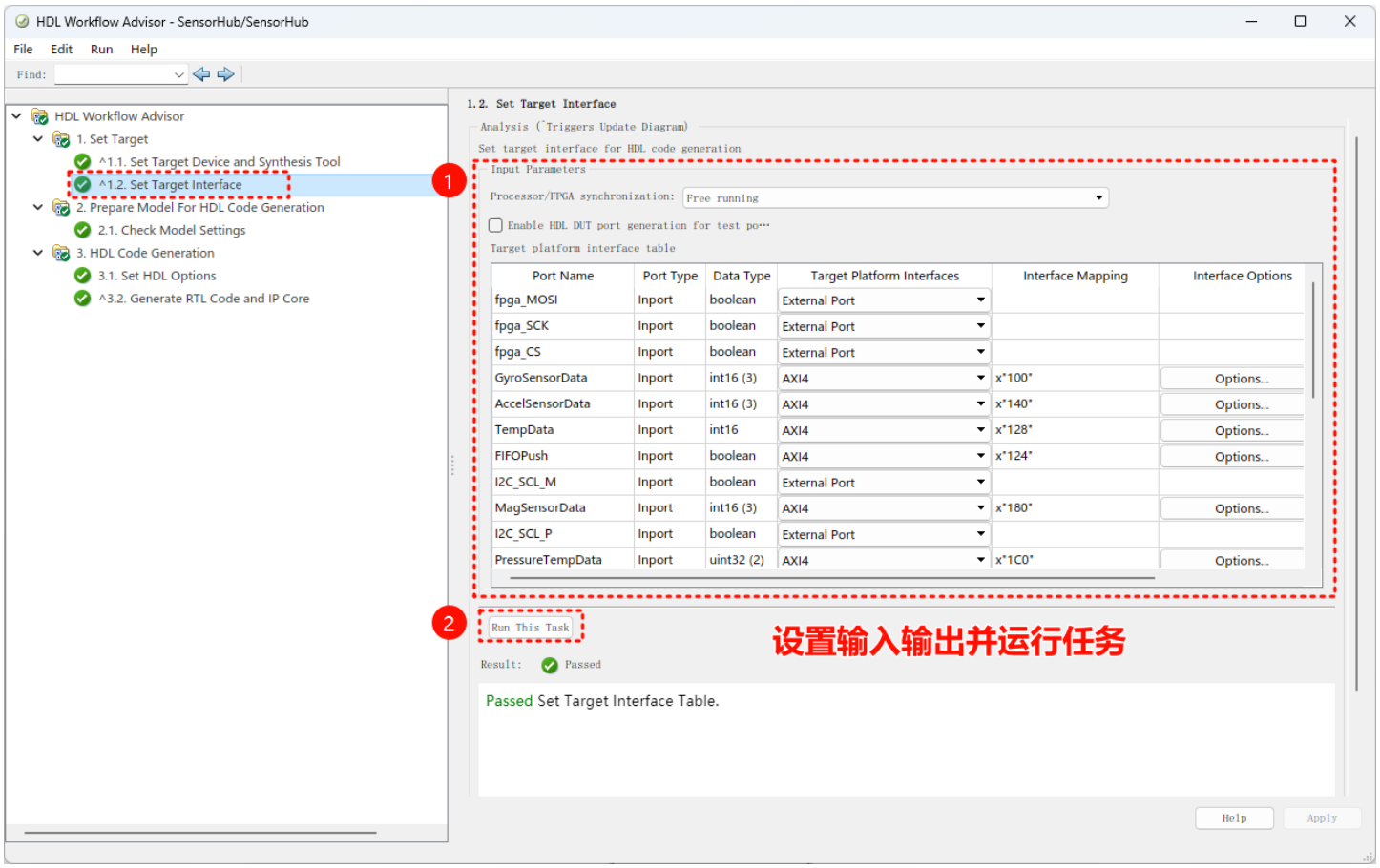

在子系统右键，快捷菜单中选择HDL Workflow Advisor
Linear Analysis
Design Verifier
Coverage
Metrics Dashboard
Fixed-Point Tool...
Identify Modeling Clones
Model Transformer
C/C++ Code
HDL Code
PLC Code
Polyspace
Block Parameters (Subsystem)
Properties...
Help
HDL Code Advisor ...
Check Subsystem Compatibility
Generate HDL for Subsystem
HDL Coder Properties ...
HDL Block Properties ...
HDL Workflow Advisor
Navigate to Code

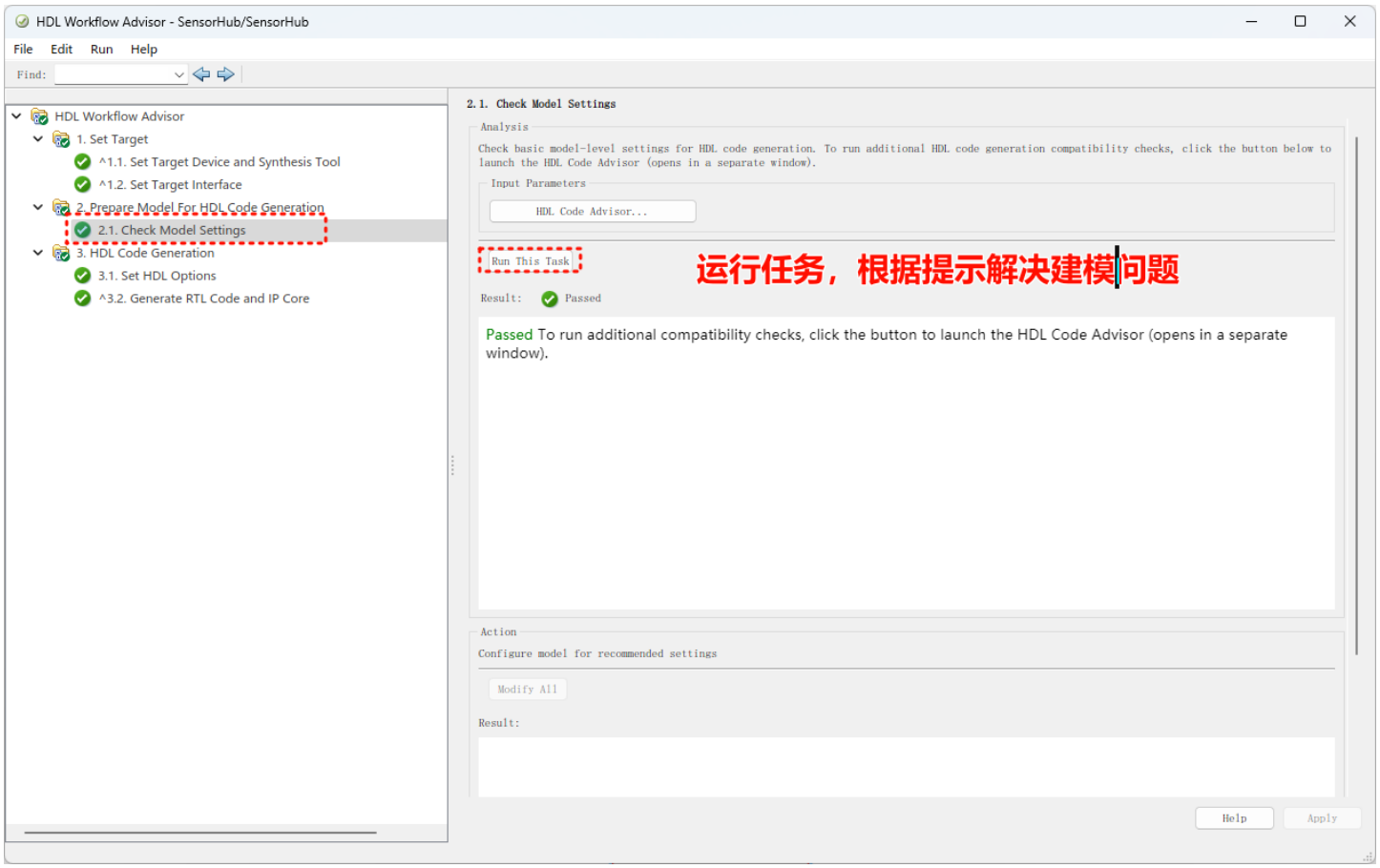
Yes No

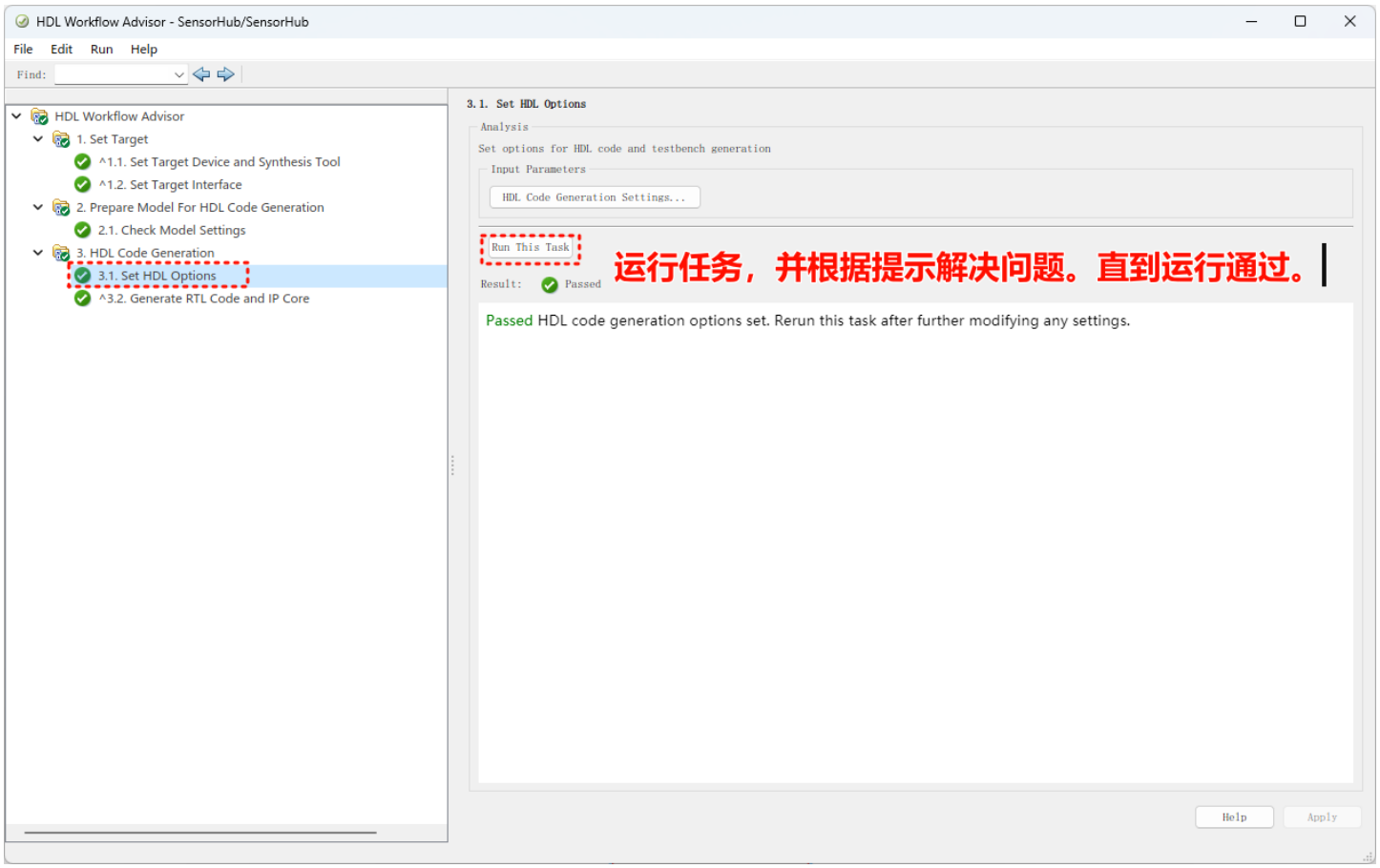

HDL Workflow Advisor - SensorHub/SensorHub
File Edit Run Help
Find:
HDL Workflow Advisor
1. Set Target
^1.1. Set Target Device and Synthesis Tool
^1.2. Set Target Interface
2. Prepare Model For HDL Code Generation
2.1. Check Model Settings
3. HDL Code Generation
3.1. Set HDL Options
^3.2. Generate RTL Code and IP Core
HDL Workflow Advisor
HDL Workflow Advisor facilitates RTL code (VHDL/Verilog) and testbench generation from a subsystem, performs synthesis tasks by invoking a supported third party synthesis tool, and annotates critical path information back to the system. It also allows you to set a particular workflow and guides you through the tasks necessary for full deployment. Each task performs one distinct step of the workflow. The HDL Workflow Advisor provides you with a feedback on the results of each task. If the task fails, it provides you with information on how to modify the model to complete the task.
When you complete the tasks, you have a synthesis result report from one of the supported synthesis tools. If the result does not meet your requirement, you may choose to modify the original model, use different implementations, or use different code generation options to refine and explore the result.
在该窗口中配置FPGA代码生成
Legend
Not Run
Passed
Failed
Warning
Group Folder - run in any order
Procedure Folder - run sequentially
Running this check triggers an Update Diagram.
->>> "Run All" in progress.
Report
Report: ...\report_1.html
Date/Time: 05-May-2022 20:40:50
Summary: Pass: 5 Fail: 0 Warning: 0 Not Run: 0
Help

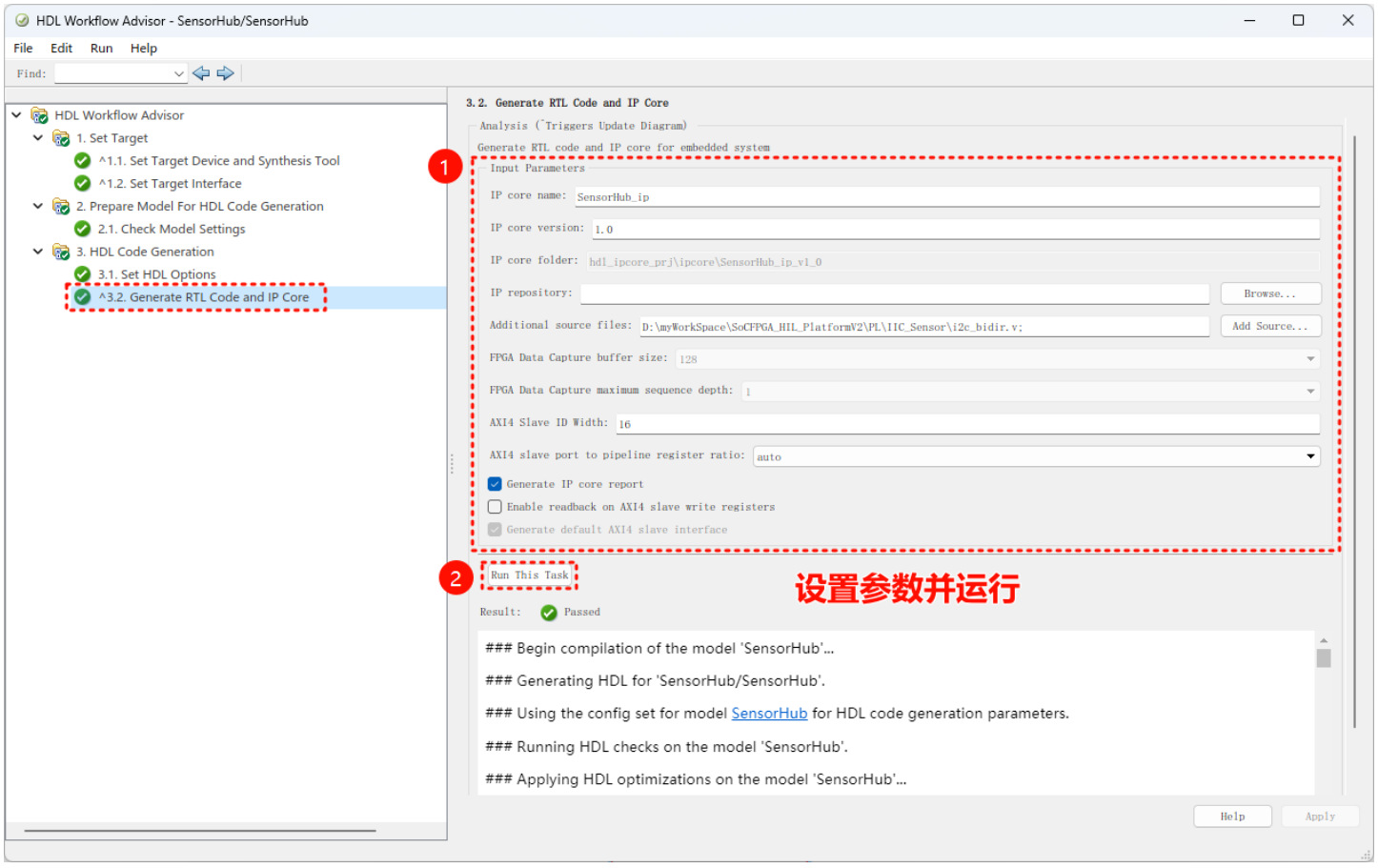
Run This Task Passed



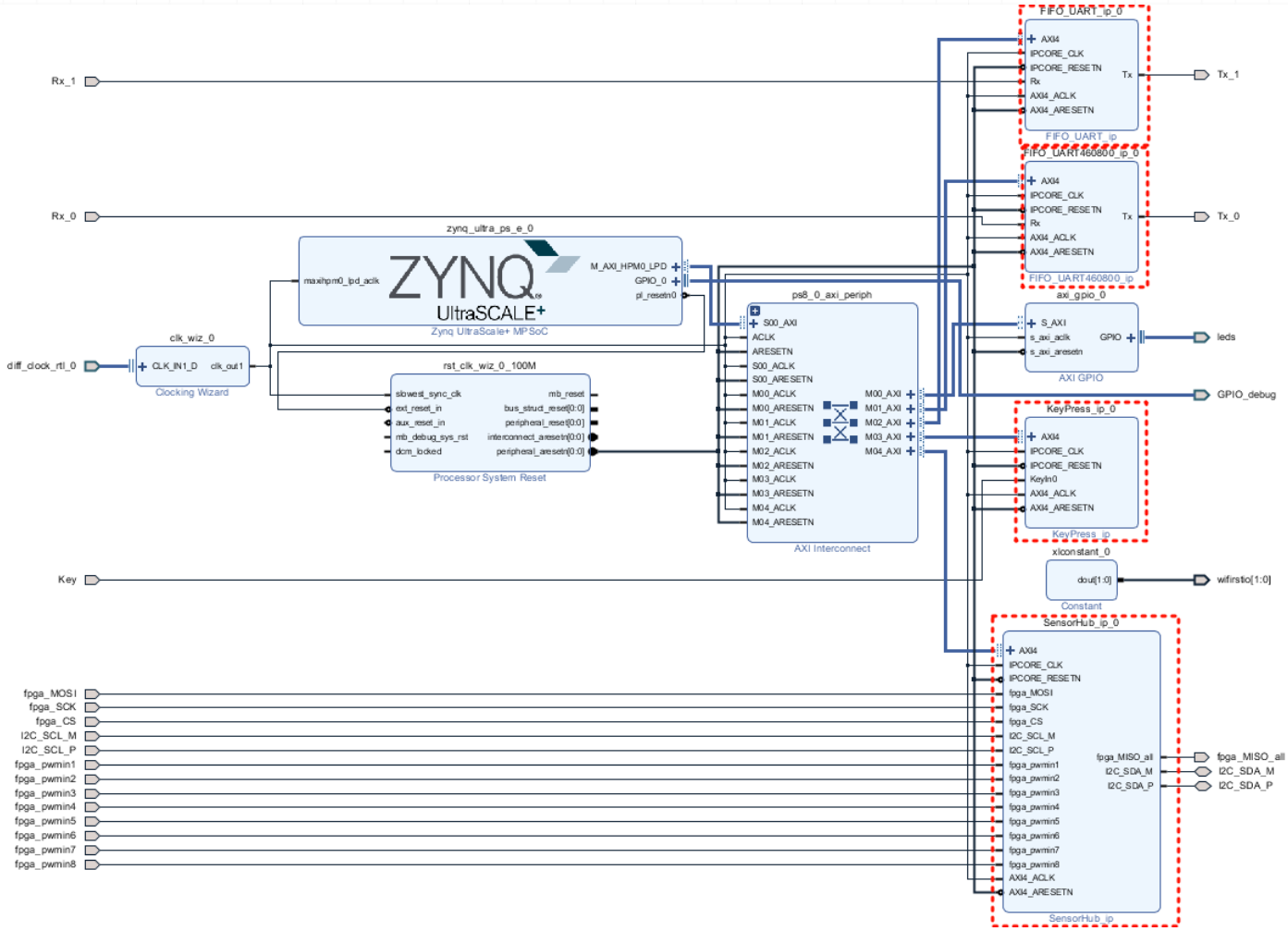

HDL Workflow Advisor - SensorHub/SensorHub
File Edit Run Help
Find:
HDL Workflow Advisor
1. Set Target
^1.1. Set Target Device and Synthesis Tool
^1.2. Set Target Interface
2. Prepare Model For HDL Code Generation
2.1. Check Model Settings
3. HDL Code Generation
3.1. Set HDL Options
^3.2. Generate RTL Code and IP Core
1.2. Set Target Interface
Analysis (Triggers Update Diagram)
Set target interface for HDL code generation
Input Parameters
Processor/FPGA synchronization: Free running
Enable HDL DUT port generation for test po…
Target platform interface table
Port Name | Port Type | Data Type | Target Platform Interfaces | Interface Mapping | Interface Options
fpga_MOSI | Inport | boolean | External Port
fpga_SCK | Inport | boolean | External Port
fpga_CS | Inport | boolean | External Port
GyroSensorData | Inport | int16 (3) | AXI4 | x"100" | Options...
AccelSensorData | Inport | int16 (3) | AXI4 | x"140" | Options...
TempData | Inport | int16 | AXI4 | x"128" | Options...
FIFOPush | Inport | boolean | AXI4 | x"124" | Options...
I2C_SCL_M | Inport | boolean | External Port
MagSensorData | Inport | int16 (3) | AXI4 | x"180" | Options...
I2C_SCL_P | Inport | boolean | External Port
PressureTempData | Inport | uint32 (2) | AXI4 | x"1C0" | Options...
1
2
Run This Task
设置输入输出并运行任务
Result: Passed
Passed Set Target Interface Table.
Help
Apply


HDL Workflow Advisor - SensorHub/SensorHub
File Edit Run Help
Find:
HDL Workflow Advisor
1. Set Target
^1.1. Set Target Device and Synthesis Tool
^1.2. Set Target Interface
2. Prepare Model For HDL Code Generation
2.1. Check Model Settings
3. HDL Code Generation
3.1. Set HDL Options
^3.2. Generate RTL Code and IP Core
2.1. Check Model Settings
Analysis
Check basic model-level settings for HDL code generation. To run additional HDL code generation compatibility checks, click the button below to launch the HDL Code Advisor (opens in a separate window).
Input Parameters
HDL Code Advisor...
Run This Task
运行任务，根据提示解决建模问题
Result: Passed
Passed To run additional compatibility checks, click the button to launch the HDL Code Advisor (opens in a separate window).
Action
Configure model for recommended settings
Modify All
Result:
Help
Apply


HDL Workflow Advisor - SensorHub/SensorHub
File Edit Run Help
Find:
HDL Workflow Advisor
1. Set Target
^1.1. Set Target Device and Synthesis Tool
^1.2. Set Target Interface
2. Prepare Model For HDL Code Generation
2.1. Check Model Settings
3. HDL Code Generation
3.1. Set HDL Options
^3.2. Generate RTL Code and IP Core
3.1. Set HDL Options
Analysis
Set options for HDL code and testbench generation
Input Parameters
HDL Code Generation Settings...
Run This Task
运行任务，并根据提示解决问题。直到运行通过。
Result: Passed
Passed HDL code generation options set. Rerun this task after further modifying any settings.
Help
Apply

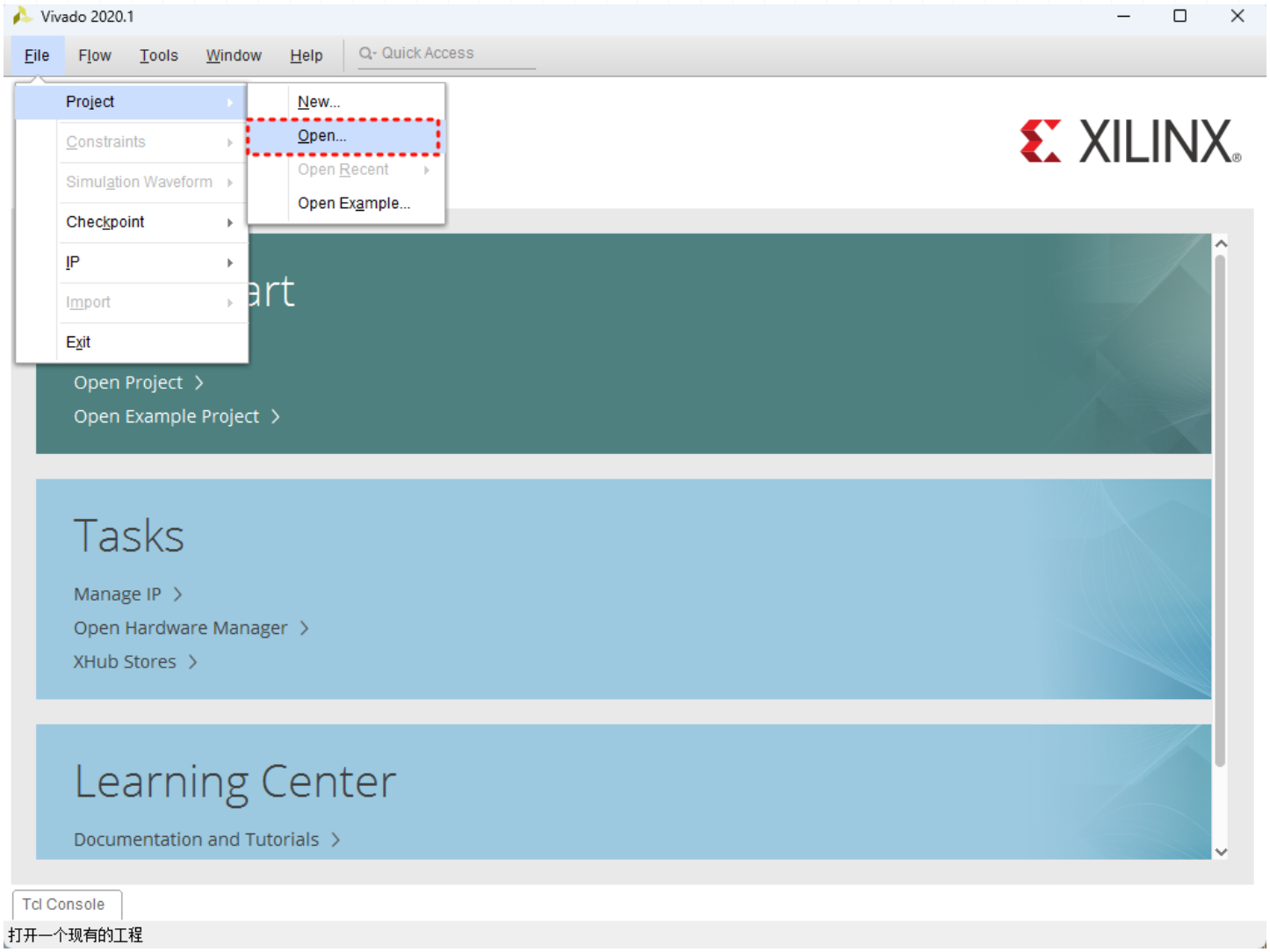

HDL Workflow Advisor - SensorHub/SensorHub
File Edit Run Help
Find:
HDL Workflow Advisor
1. Set Target
^1.1. Set Target Device and Synthesis Tool
^1.2. Set Target Interface
2. Prepare Model For HDL Code Generation
2.1. Check Model Settings
3. HDL Code Generation
3.1. Set HDL Options
^3.2. Generate RTL Code and IP Core
3.2. Generate RTL Code and IP Core
Analysis (Triggers Update Diagram)
Generate RTL code and IP core for embedded system
1
Input Parameters
IP core name: SensorHub_ip
IP core version: 1.0
IP core folder: hdl_ipcore_prj\ipcore\SensorHub_ip_v1_0
IP repository:
Browse...
Additional source files: D:\myWorkSpace\SoCFPGA_HIL_PlatformV2\PL\IIC_Sensor\i2c_bidir.v;
Add Source...
FPGA Data Capture buffer size: 128
FPGA Data Capture maximum sequence depth: 1
AXI4 Slave ID Width: 16
AXI4 slave port to pipeline register ratio: auto
Generate IP core report
Enable readback on AXI4 slave write registers
Generate default AXI4 slave interface
2
Run This Task
设置参数并运行
Result: Passed
Begin compilation of the model 'SensorHub'...
Generating HDL for 'SensorHub/SensorHub'.
Using the config set for model SensorHub for HDL code generation parameters.
Running HDL checks on the model 'SensorHub'.
Applying HDL optimizations on the model 'SensorHub'...
Help
Apply

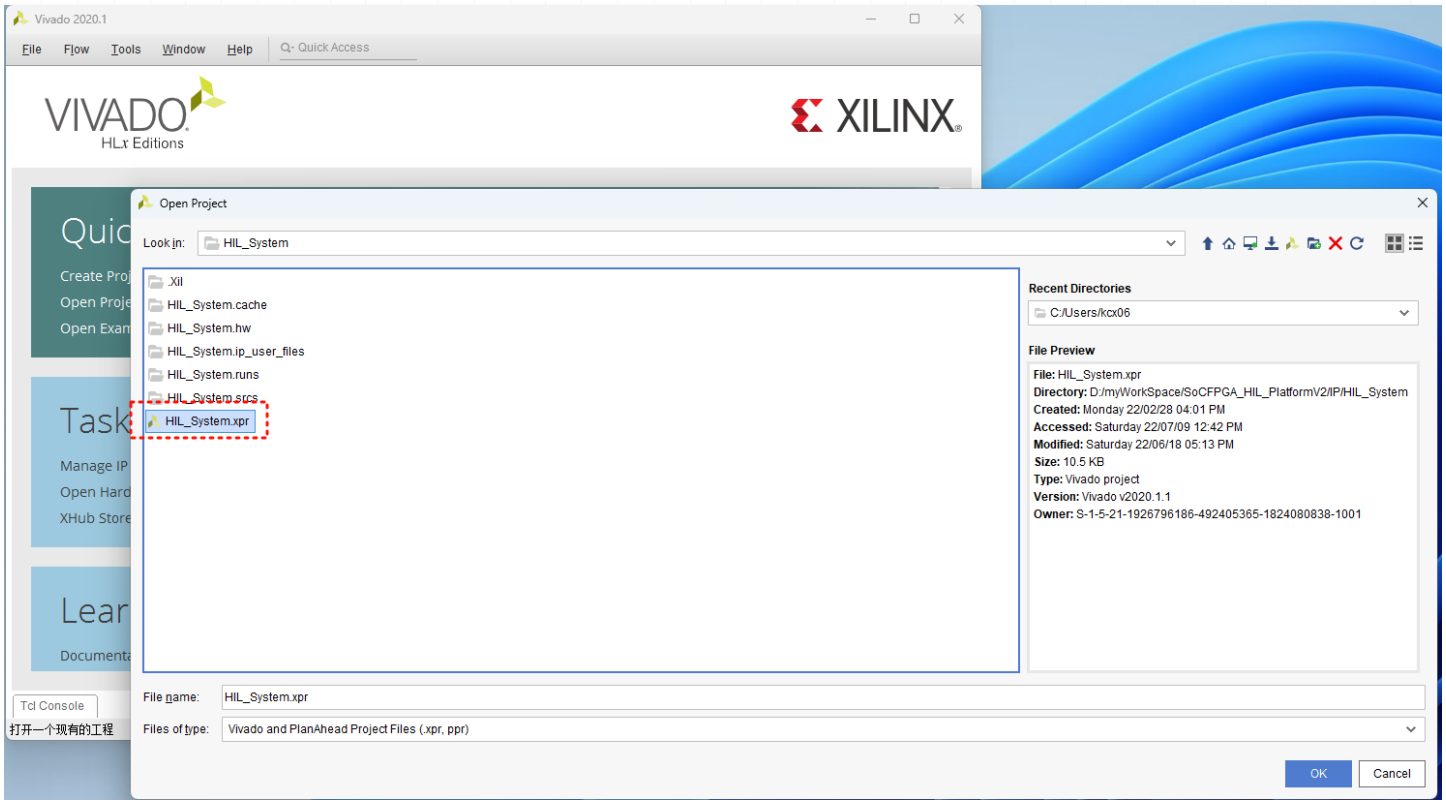
slx IP Core FPGA



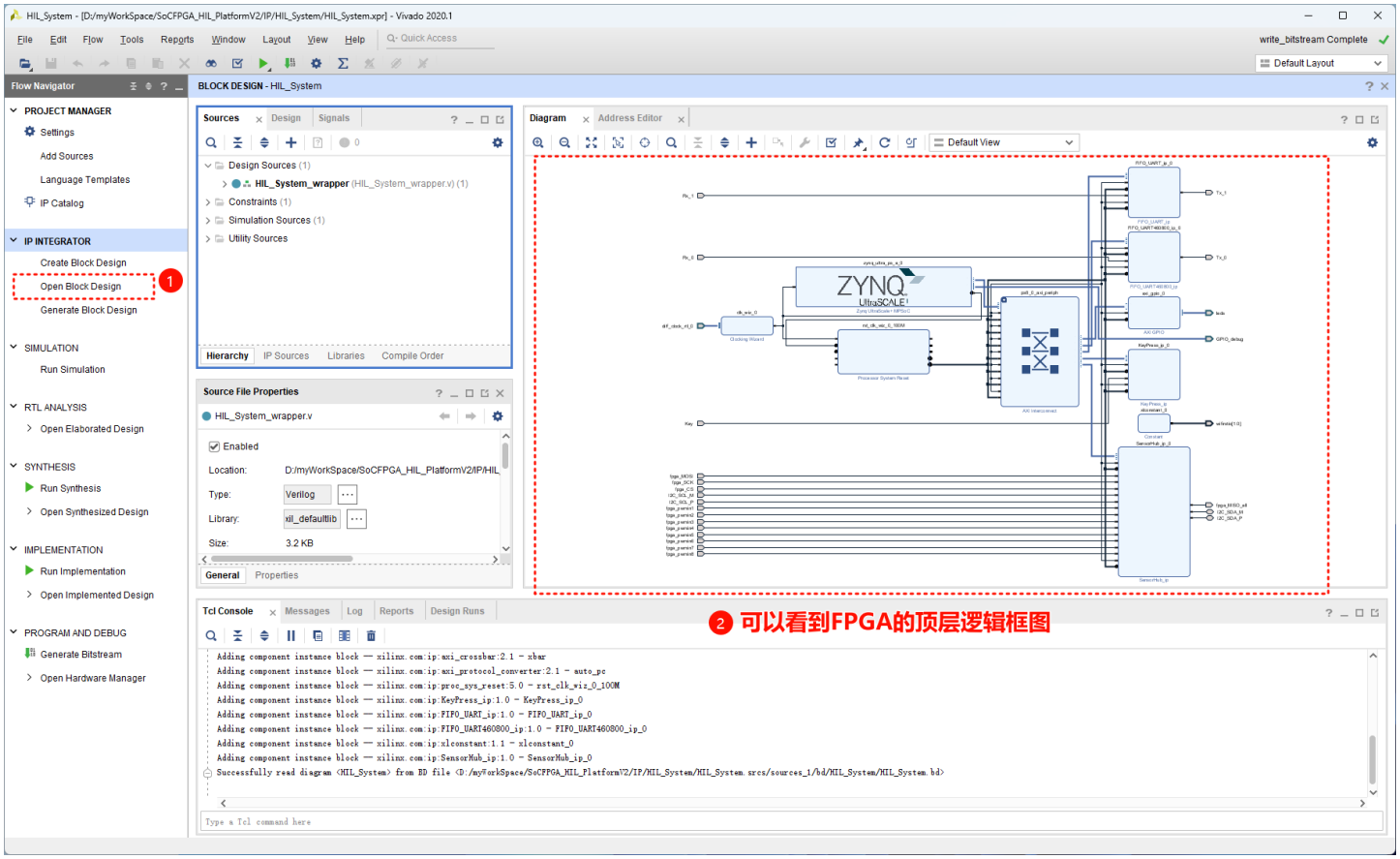
4.2.2 2. Vivado .xsa

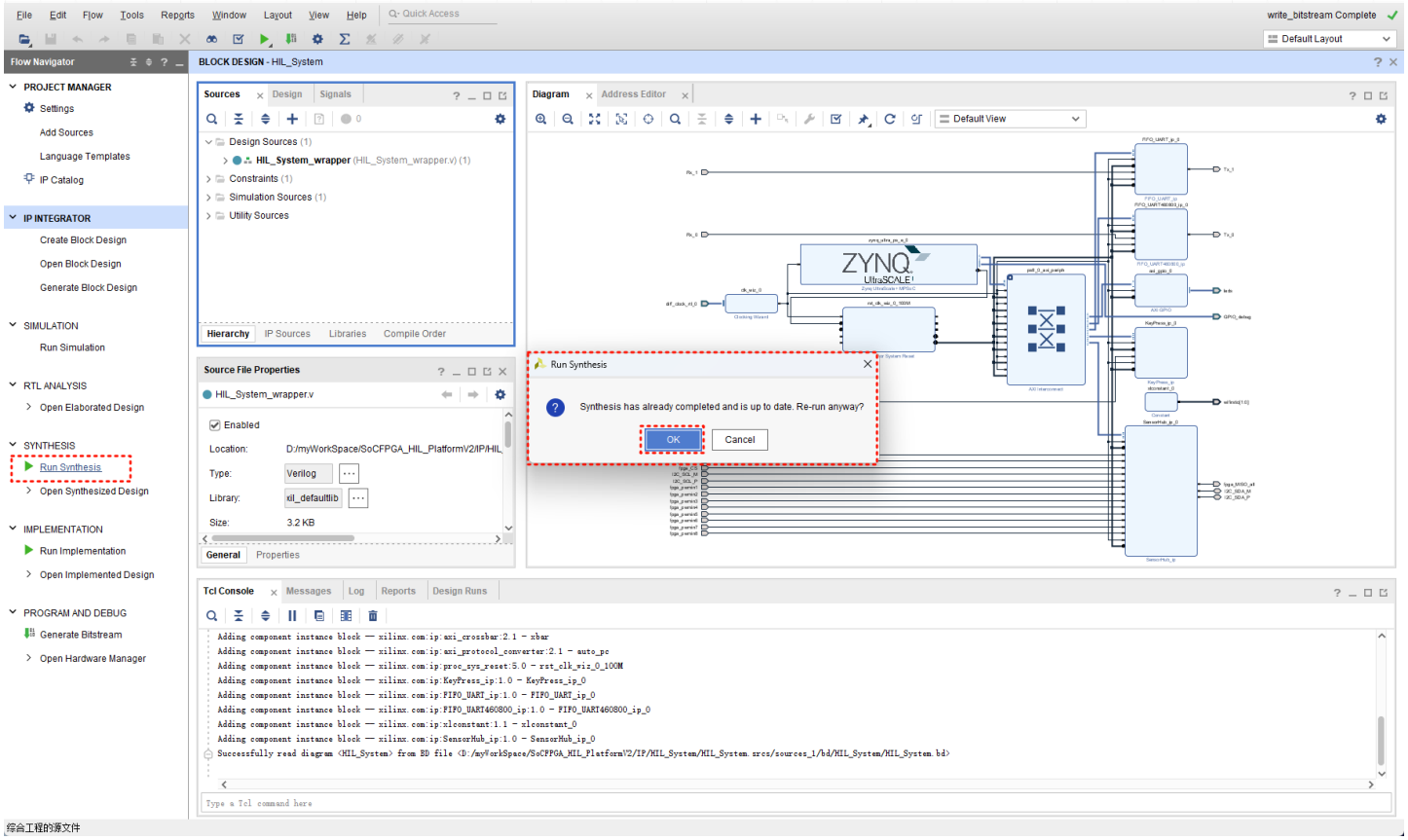
vivado FPGA

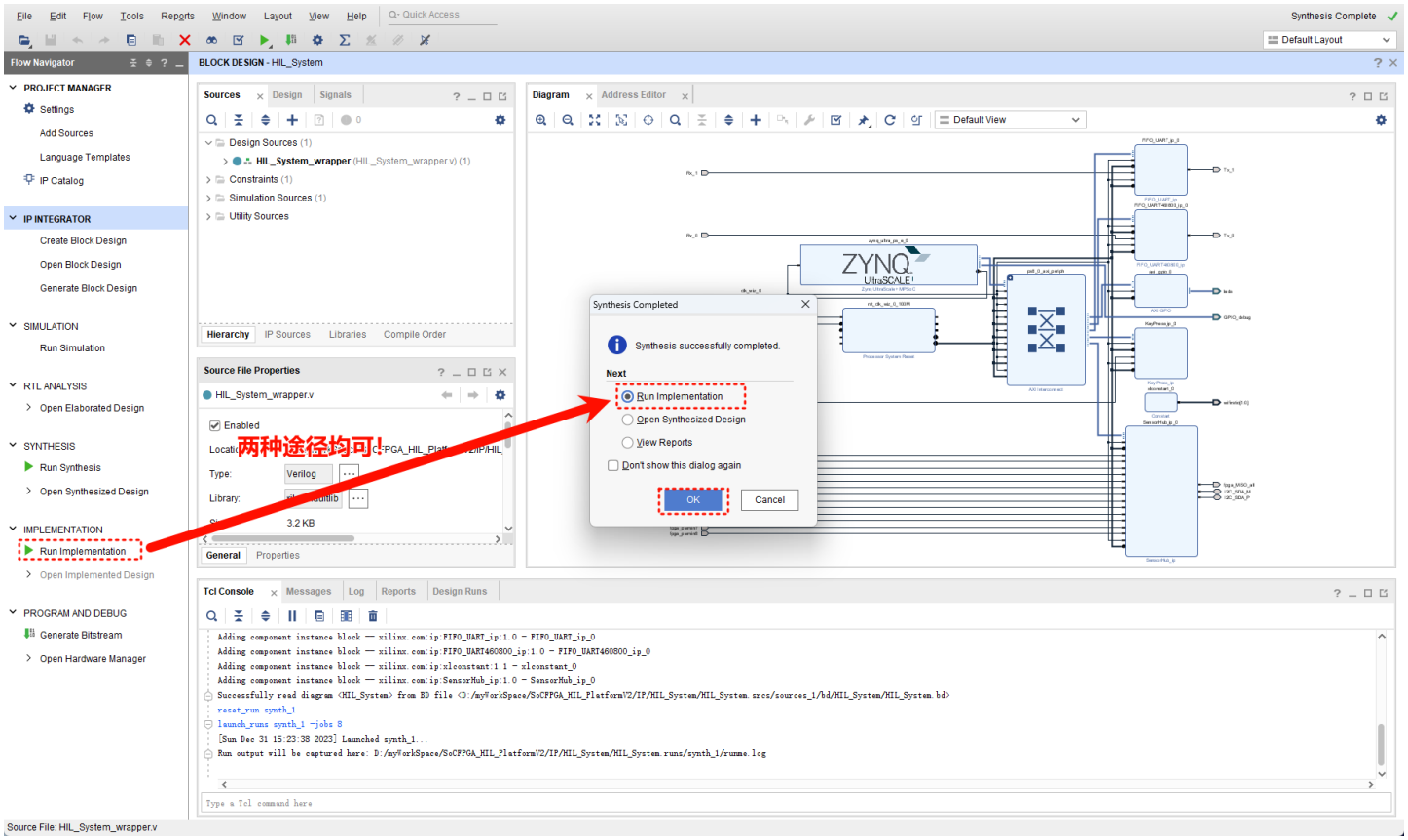


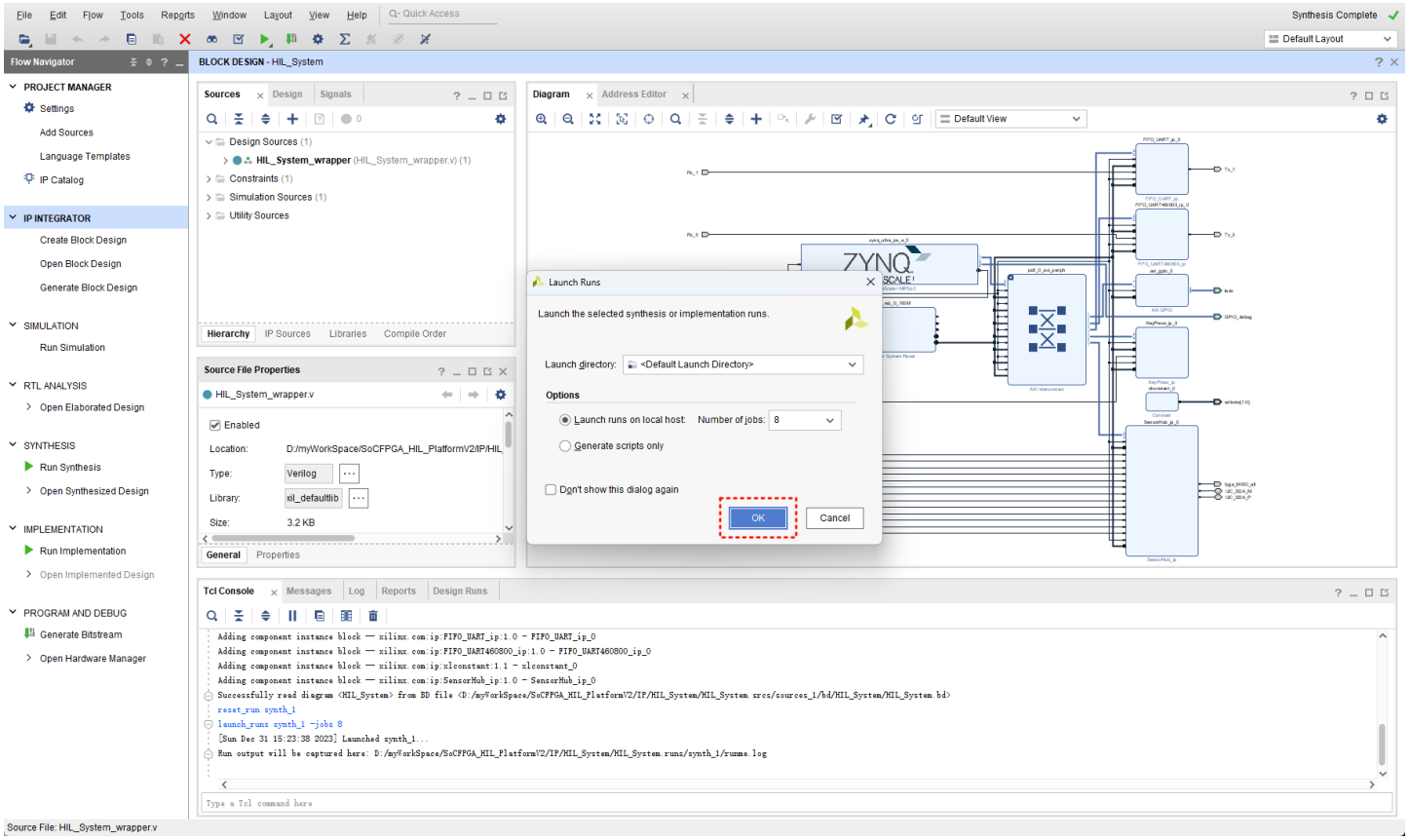

Vivado 2020.1
File Flow Tools Window Help Quick Access
VIVADO HLx Editions
XILINX
Open Project
Look in: HIL_System
.Xil
HIL_System.cache
HIL_System.hw
HIL_System.ip_user_files
HIL_System.runs
HIL_System.srcs
HIL_System.xpr
Recent Directories
C:/Users/kcx06
File Preview
File: HIL_System.xpr
Directory: D:/myWorkSpace/SoCFPGA_HIL_PlatformV2/IP/HIL_System
Created: Monday 22/02/28 04:01 PM
Accessed: Saturday 22/07/09 12:42 PM
Modified: Saturday 22/06/18 05:13 PM
Size: 10.5 KB
Type: Vivado project
Version: Vivado v2020.1.1
Owner: S-1-5-21-1926796186-492405365-1824080838-1001
File name: HIL_System.xpr
Files of type: Vivado and PlanAhead Project Files (.xpr, ppr)
OK
Cancel
Tcl Console
打开一个现有的工程

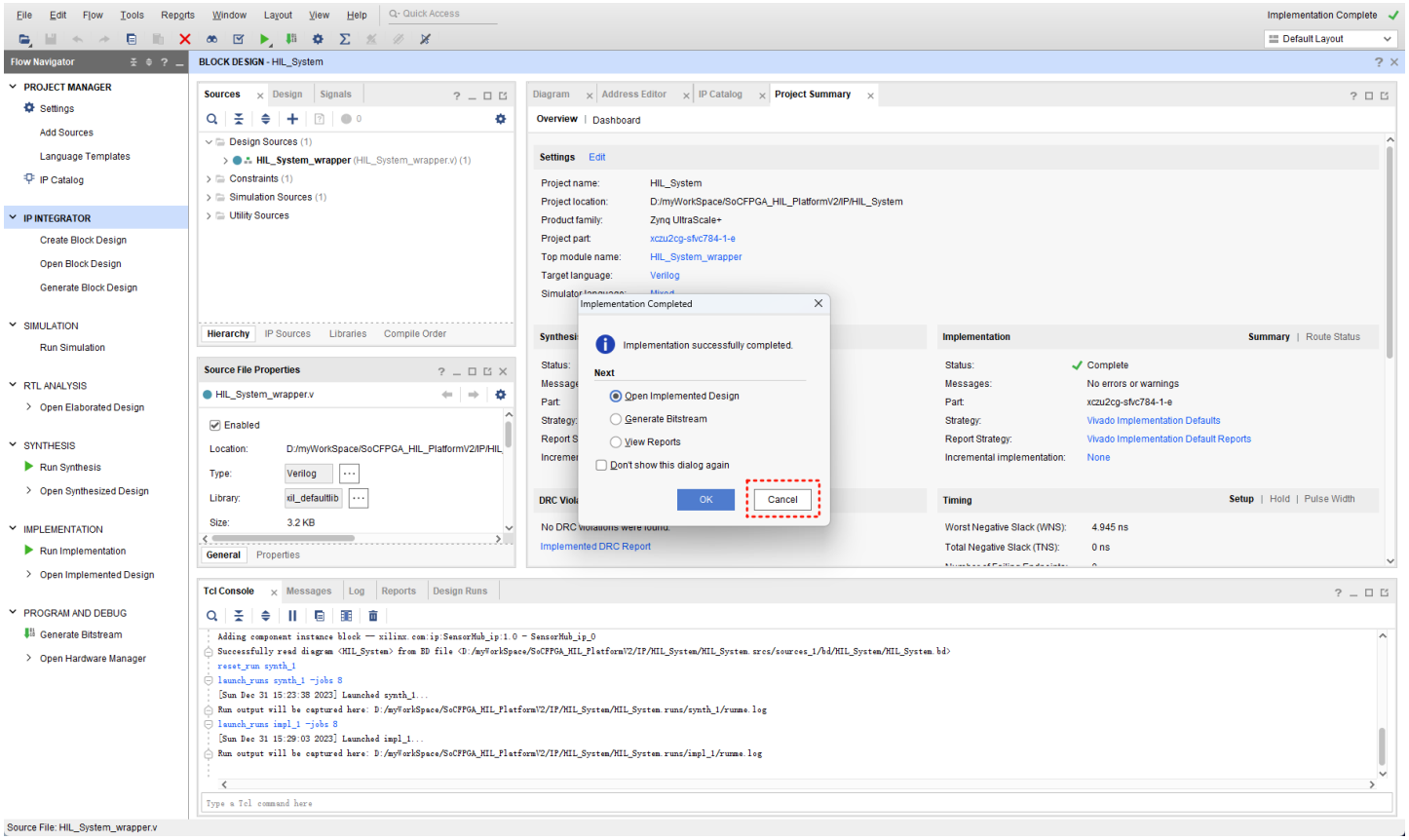
FPGA PL


HIL_System - [D:/myWorkSpace/SoCFPGA_HIL_PlatformV2/IP/HIL_System/HIL_System.xpr] - Vivado 2020.1
write_bitstream Complete
Default Layout
Flow Navigator
PROJECT MANAGER
Settings
Add Sources
Language Templates
IP Catalog
IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Open Elaborated Design
SYNTHESIS
Run Synthesis
Open Synthesized Design
IMPLEMENTATION
Run Implementation
Open Implemented Design
PROGRAM AND DEBUG
Generate Bitstream
Open Hardware Manager
1
BLOCK DESIGN - HIL_System
Sources
Design Sources (1)
HIL_System_wrapper (HIL_System_wrapper.v) (1)
Constraints (1)
Simulation Sources (1)
Utility Sources
Source File Properties
HIL_System_wrapper.v
Enabled
Location: D:/myWorkSpace/SoCFPGA_HIL_PlatformV2/IP/HIL
Type: Verilog
Library: xil_defaultlib
Size: 3.2 KB
Diagram
Address Editor
Default View
ZYNQ UltraSCALE+
Tcl Console
Messages
Log
Reports
Design Runs
Adding component instance block -- xilinx.com:ip:axi_crossbar:2.1 - xbar
Adding component instance block -- xilinx.com:ip:axi_protocol_converter:2.1 - auto_pc
Adding component instance block -- xilinx.com:ip:proc_sys_reset:5.0 - rst_clk_wiz_0_100M
Adding component instance block -- xilinx.com:ip:KeyPress_ip:1.0 - KeyPress_ip_0
Adding component instance block -- xilinx.com:ip:FIFO_UART_ip:1.0 - FIFO_UART_ip_0
Adding component instance block -- xilinx.com:ip:FIFO_UART460800_ip:1.0 - FIFO_UART460800_ip_0
Adding component instance block -- xilinx.com:ip:xlconstant:1.1 - xlconstant_0
Adding component instance block -- xilinx.com:ip:SensorHub_ip:1.0 - SensorHub_ip_0
Successfully read diagram <HIL_System> from BD file <D:/myWorkSpace/SoCFPGA_HIL_PlatformV2/IP/HIL_System/HIL_System.srcs/sources_1/bd/HIL_System/HIL_System.bd>
Type a Tcl command here
2 可以看到FPGA的顶层逻辑框图

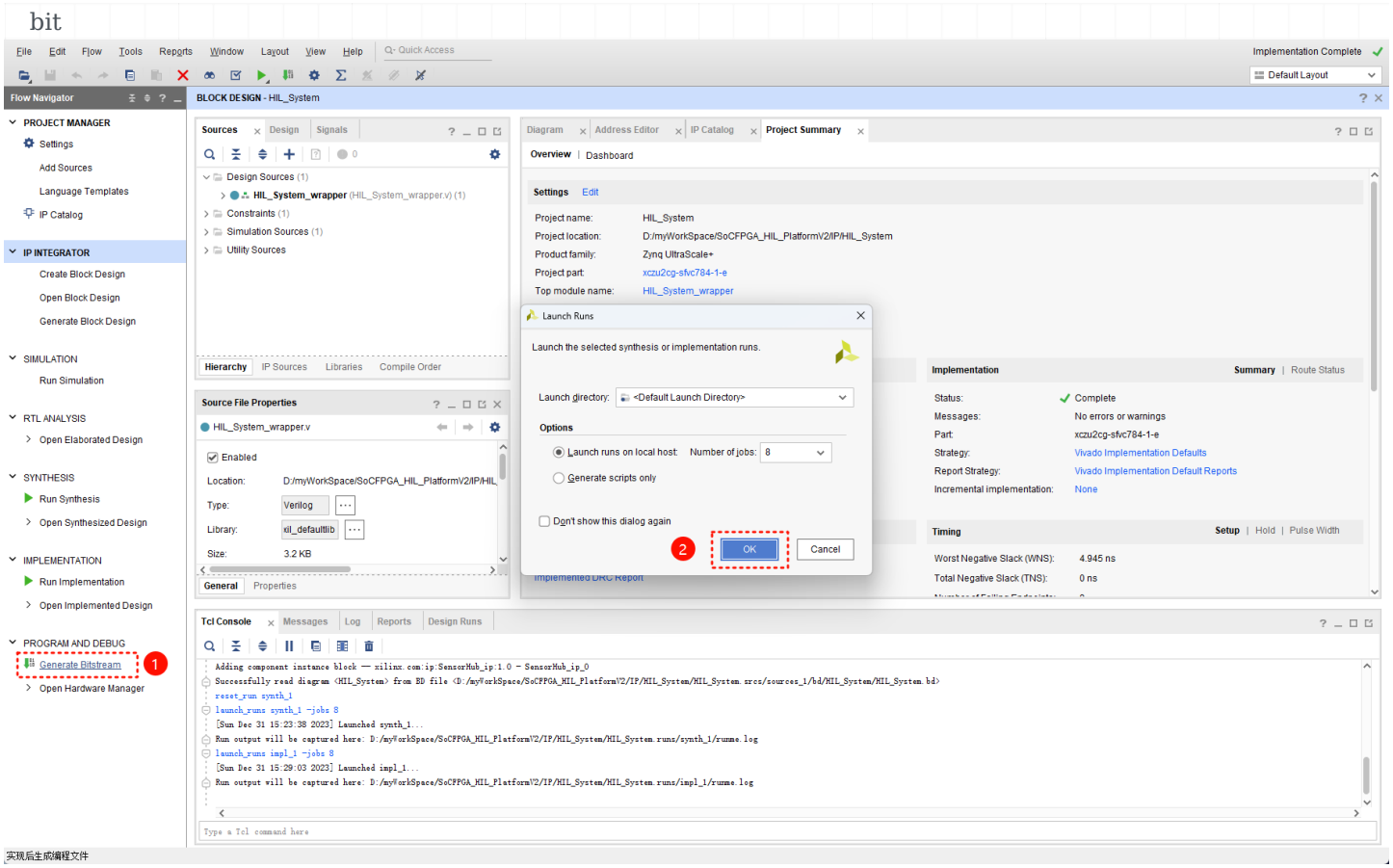

File Edit Flow Tools Reports Window Layout View Help
write_bitstream Complete
Default Layout
Flow Navigator
BLOCK DESIGN - HIL_System
PROJECT MANAGER
Settings
Add Sources
Language Templates
IP Catalog
IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Open Elaborated Design
SYNTHESIS
Run Synthesis
Open Synthesized Design
IMPLEMENTATION
Run Implementation
Open Implemented Design
PROGRAM AND DEBUG
Generate Bitstream
Open Hardware Manager
Sources
Design
Signals
Design Sources (1)
HIL_System_wrapper (HIL_System_wrapper.v) (1)
Constraints (1)
Simulation Sources (1)
Utility Sources
Hierarchy
IP Sources
Libraries
Compile Order
Source File Properties
HIL_System_wrapper.v
Enabled
Location:
Type:
Verilog
Library:
xil_defaultlib
Size:
3.2 KB
General
Properties
Diagram
Address Editor
Default View
Run Synthesis
Synthesis has already completed and is up to date. Re-run anyway?
OK
Cancel
Tcl Console
Messages
Log
Reports
Design Runs
Adding component instance block -- xilinx.com:ip:axi_crossbar:2.1 - xbar
Adding component instance block -- xilinx.com:ip:axi_protocol_converter:2.1 - auto_pc
Adding component instance block -- xilinx.com:ip:proc_sys_reset:5.0 - rst_clk_wiz_0_100M
Adding component instance block -- xilinx.com:ip:KeyPress_ip:1.0 - KeyPress_ip_0
Adding component instance block -- xilinx.com:ip:FIFO_UART_ip:1.0 - FIFO_UART_ip_0
Adding component instance block -- xilinx.com:ip:FIFO_UART460800_ip:1.0 - FIFO_UART460800_ip_0
Adding component instance block -- xilinx.com:ip:xlconstant:1.1 - xlconstant_0
Adding component instance block -- xilinx.com:ip:SensorHub_ip:1.0 - SensorHub_ip_0
Successfully read diagram <HIL_System> from BD file <D:/myWorkSpace/SoCFPGA_HIL_PlatformV2/IP/HIL_System/HIL_System.srcs/sources_1/bd/HIL_System/HIL_System.bd>
Type a Tcl command here
综合工程的源文件

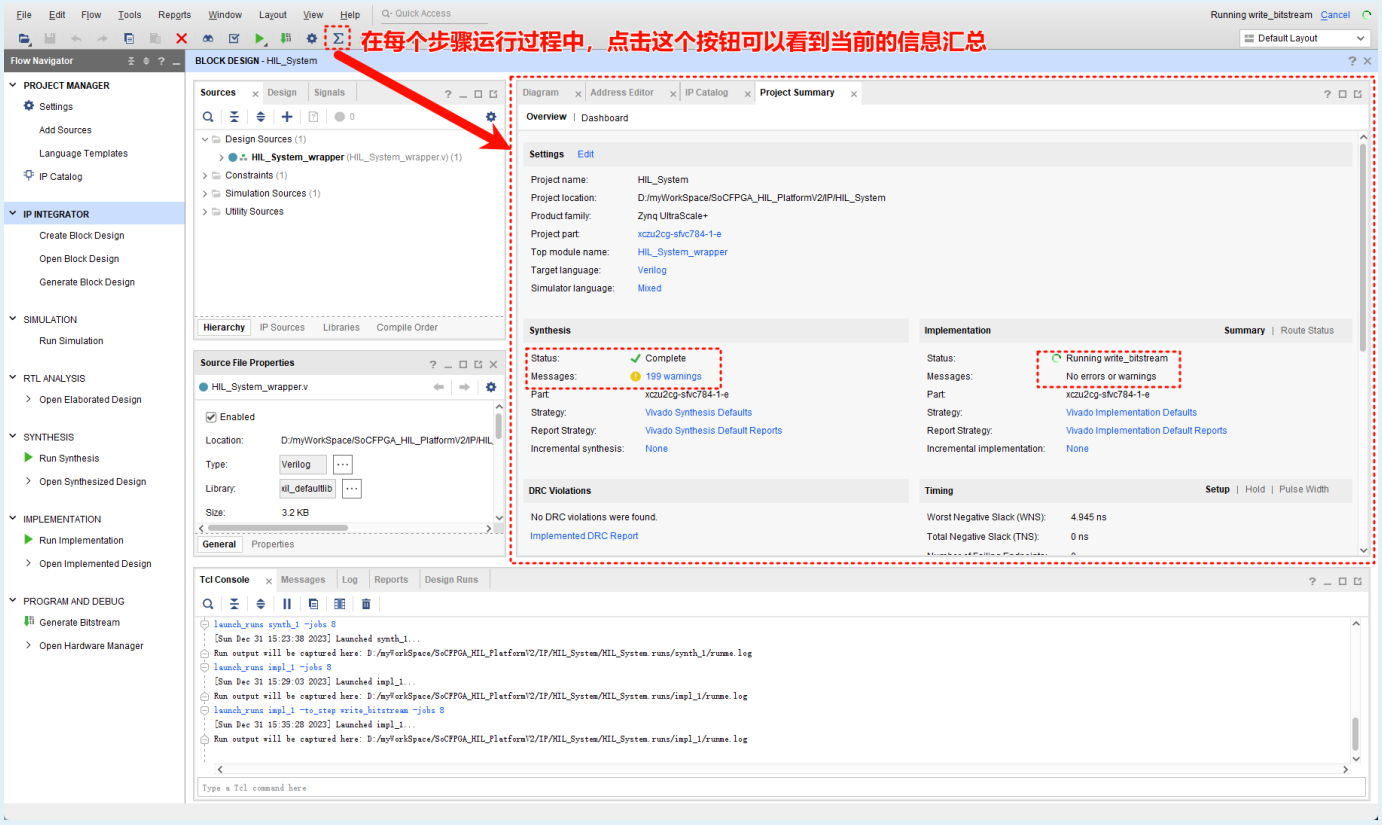

File Edit Flow Tools Reports Window Layout View Help
Synthesis Complete
Default Layout
Flow Navigator
BLOCK DESIGN - HIL_System
PROJECT MANAGER
Settings
Add Sources
Language Templates
IP Catalog
IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Open Elaborated Design
SYNTHESIS
Run Synthesis
Open Synthesized Design
IMPLEMENTATION
Run Implementation
Open Implemented Design
PROGRAM AND DEBUG
Generate Bitstream
Open Hardware Manager
两种途径均可!
Synthesis Completed
Synthesis successfully completed.
Next
Run Implementation
Open Synthesized Design
View Reports
Don't show this dialog again
OK
Cancel
Tcl Console
Messages
Log
Reports
Design Runs
Adding component instance block -- xilinx.com:ip:FIFO_UART_ip:1.0 - FIFO_UART_ip_0
Adding component instance block -- xilinx.com:ip:FIFO_UART460800_ip:1.0 - FIFO_UART460800_ip_0
Adding component instance block -- xilinx.com:ip:xlconstant:1.1 - xlconstant_0
Adding component instance block -- xilinx.com:ip:SensorHub_ip:1.0 - SensorHub_ip_0
Successfully read diagram <HIL_System> from BD file <D:/myWorkSpace/SoCFPGA_HIL_PlatformV2/IP/HIL_System/HIL_System.srcs/sources_1/bd/HIL_System/HIL_System.bd>
reset_run synth_1
launch_runs synth_1 -jobs 8
[Sun Dec 31 15:23:38 2023] Launched synth_1...
Run output will be captured here: D:/myWorkSpace/SoCFPGA_HIL_PlatformV2/IP/HIL_System/HIL_System.runs/synth_1/runme.log
Type a Tcl command here
Source File: HIL_System_wrapper.v


Launch Runs
Launch the selected synthesis or implementation runs.
Launch directory: <Default Launch Directory>
Options
Launch runs on local host: Number of jobs: 8
Generate scripts only
Don't show this dialog again
OK
Cancel

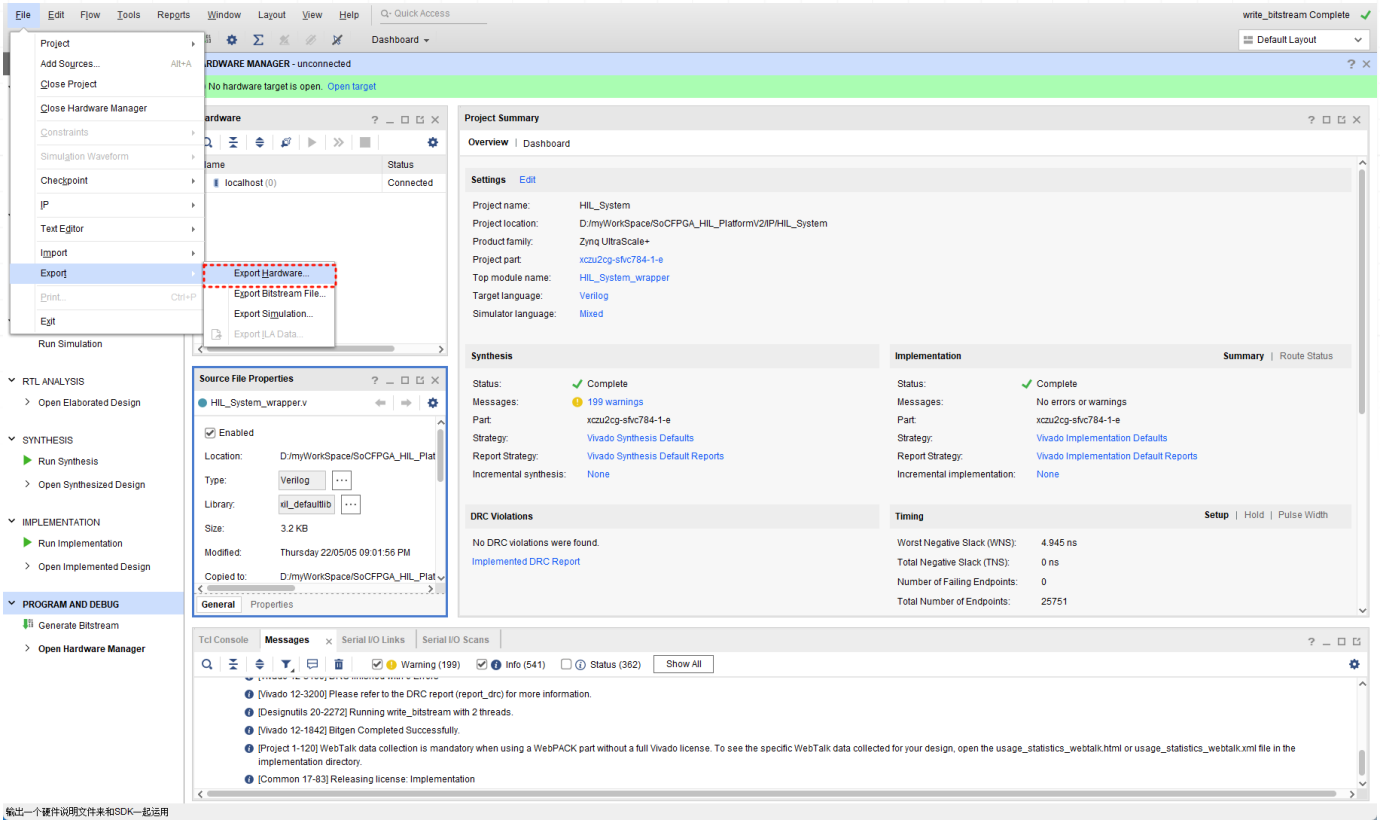

Implementation Completed
Implementation successfully completed.
Next
Open Implemented Design
Generate Bitstream
View Reports
Don't show this dialog again
OK
Cancel

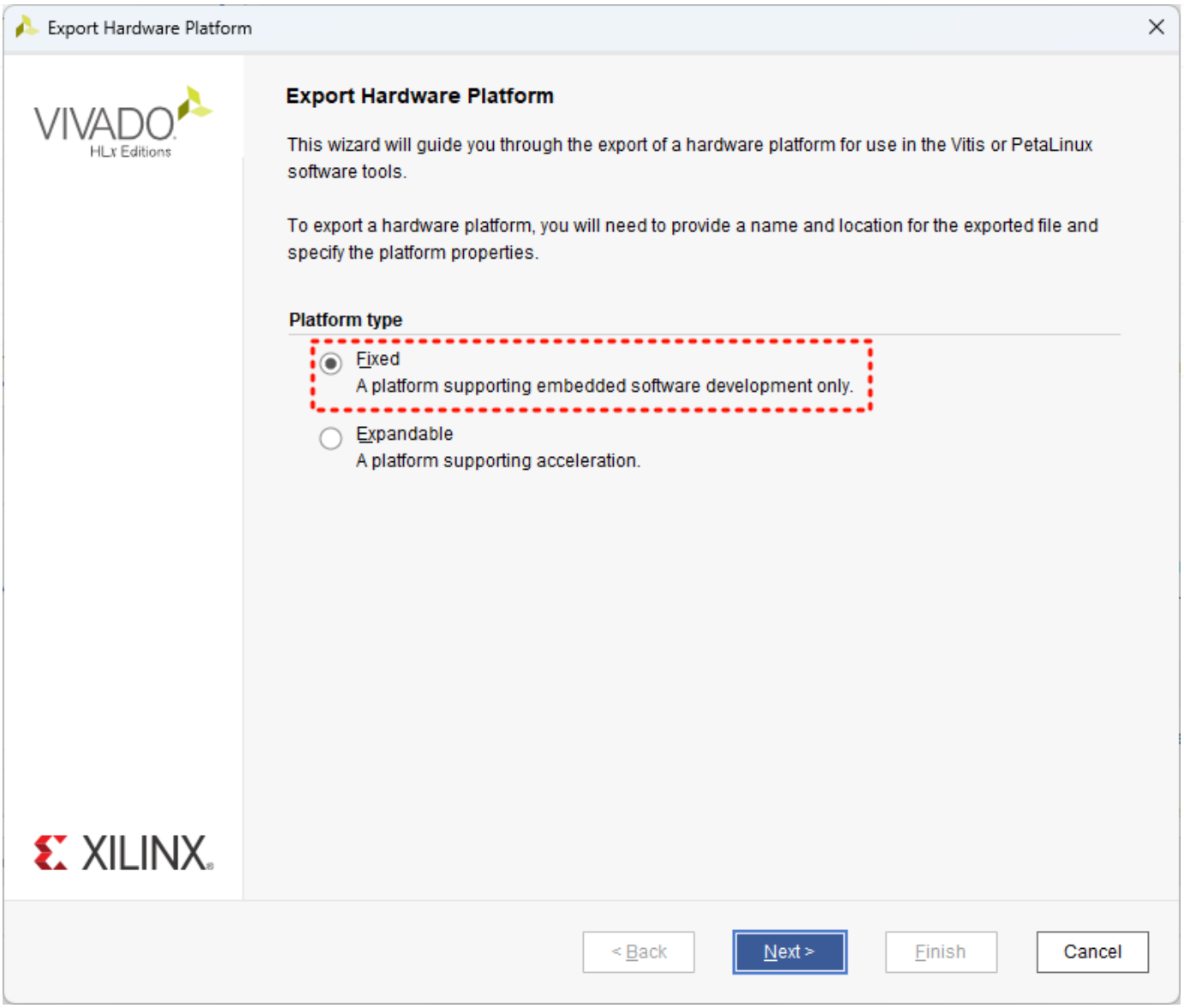
bit

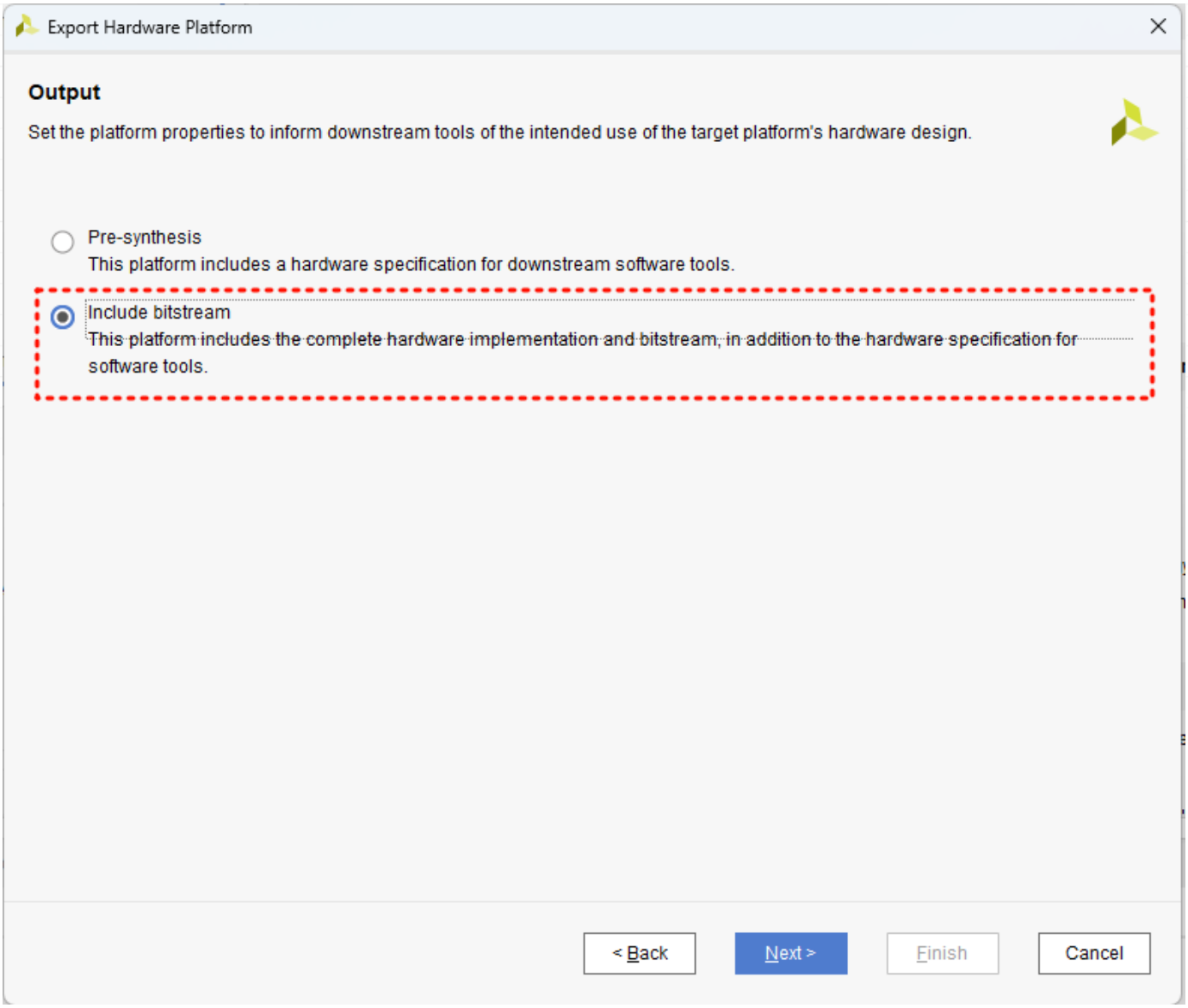

File Edit Flow Tools Reports Window Layout View Help
Quick Access
Implementation Complete
Default Layout
Flow Navigator
BLOCK DESIGN - HIL_System
PROJECT MANAGER
Settings
Add Sources
Language Templates
IP Catalog
IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Open Elaborated Design
SYNTHESIS
Run Synthesis
Open Synthesized Design
IMPLEMENTATION
Run Implementation
Open Implemented Design
PROGRAM AND DEBUG
Generate Bitstream
1
Open Hardware Manager
Sources
Design
Signals
Design Sources (1)
HIL_System_wrapper (HIL_System_wrapper.v) (1)
Constraints (1)
Simulation Sources (1)
Utility Sources
Hierarchy
IP Sources
Libraries
Compile Order
Source File Properties
HIL_System_wrapper.v
Enabled
Location: D:/myWorkSpace/SoCFPGA_HIL_PlatformV2/IP/HIL
Type: Verilog
Library: xil_defaultlib
Size: 3.2 KB
General
Properties
Diagram
Address Editor
IP Catalog
Project Summary
Overview | Dashboard
Settings Edit
Project name: HIL_System
Project location: D:/myWorkSpace/SoCFPGA_HIL_PlatformV2/IP/HIL_System
Product family: Zynq UltraScale+
Project part: xczu2cg-sfvc784-1-e
Top module name: HIL_System_wrapper
Launch Runs
Launch the selected synthesis or implementation runs.
Launch directory: <Default Launch Directory>
Options
Launch runs on local host: Number of jobs: 8
Generate scripts only
Don't show this dialog again
2
OK
Cancel
Implementation
Summary | Route Status
Status: Complete
Messages: No errors or warnings
Part: xczu2cg-sfvc784-1-e
Strategy: Vivado Implementation Defaults
Report Strategy: Vivado Implementation Default Reports
Incremental implementation: None
Timing
Setup | Hold | Pulse Width
Worst Negative Slack (WNS): 4.945 ns
Total Negative Slack (TNS): 0 ns
Tcl Console
Messages
Log
Reports
Design Runs
Adding component instance block -- xilinx.com:ip:SensorHub_ip:1.0 - SensorHub_ip_0
Successfully read diagram <HIL_System> from BD file <D:/myWorkSpace/SoCFPGA_HIL_PlatformV2/IP/HIL_System/HIL_System.srcs/sources_1/bd/HIL_System/HIL_System.bd>
reset_run synth_1
launch_runs synth_1 -jobs 8
[Sun Dec 31 15:23:38 2023] Launched synth_1...
Run output will be captured here: D:/myWorkSpace/SoCFPGA_HIL_PlatformV2/IP/HIL_System/HIL_System.runs/synth_1/runme.log
launch_runs impl_1 -jobs 8
[Sun Dec 31 15:29:03 2023] Launched impl_1...
Run output will be captured here: D:/myWorkSpace/SoCFPGA_HIL_PlatformV2/IP/HIL_System/HIL_System.runs/impl_1/runme.log
Type a Tcl command here
实现后生成编程文件

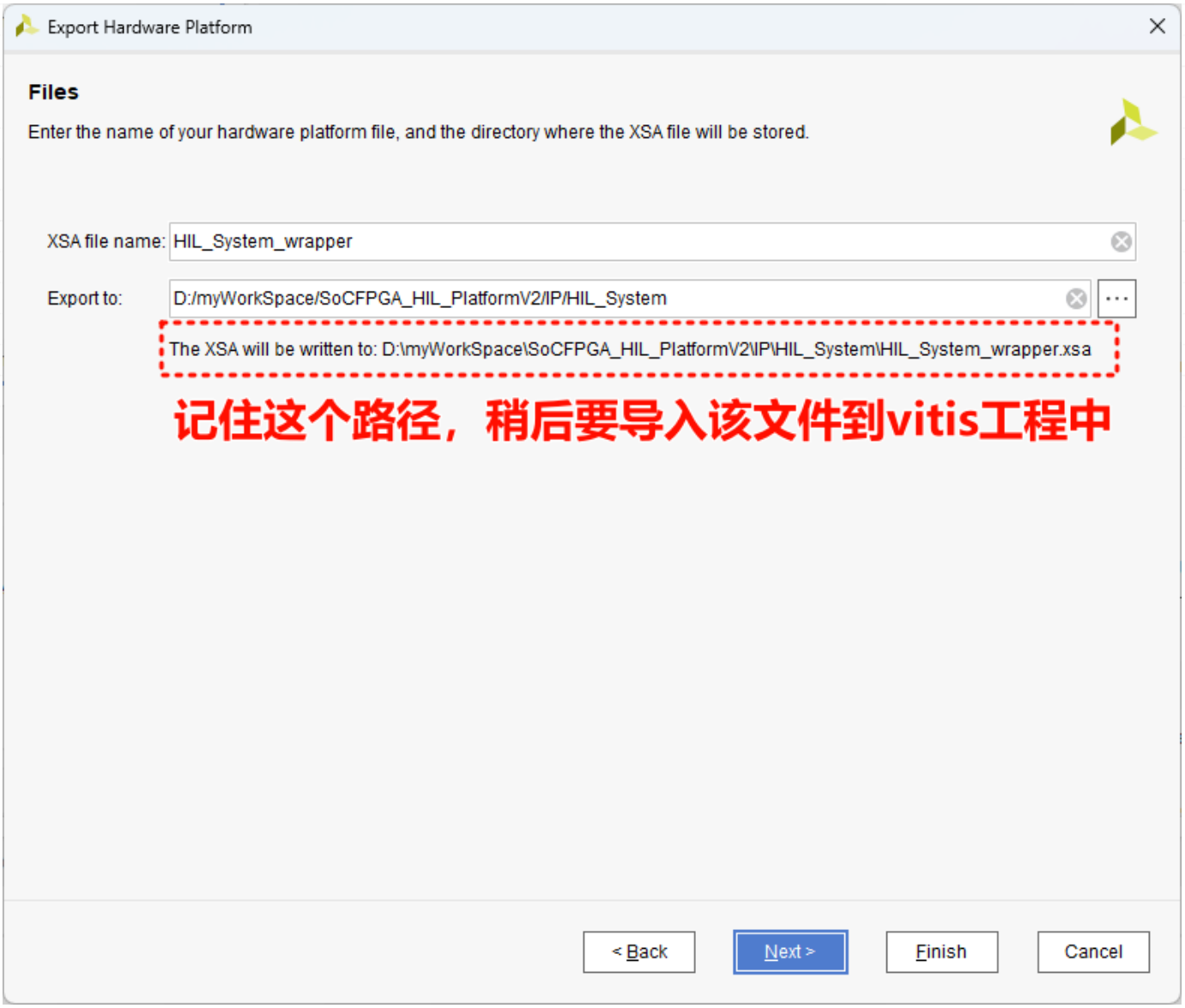

在每个步骤运行过程中，点击这个按钮可以看到当前的信息汇总
File Edit Flow Tools Reports Window Layout View Help
Running write_bitstream Cancel
Default Layout
Flow Navigator
BLOCK DESIGN - HIL_System
PROJECT MANAGER
Settings
Add Sources
Language Templates
IP Catalog
IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Open Elaborated Design
SYNTHESIS
Run Synthesis
Open Synthesized Design
IMPLEMENTATION
Run Implementation
Open Implemented Design
PROGRAM AND DEBUG
Generate Bitstream
Open Hardware Manager
Sources Design Signals
Design Sources (1)
HIL_System_wrapper (HIL_System_wrapper.v) (1)
Constraints (1)
Simulation Sources (1)
Utility Sources
Hierarchy IP Sources Libraries Compile Order
Source File Properties
HIL_System_wrapper.v
Enabled
Location: D:/myWorkSpace/SoCFPGA_HIL_PlatformV2/IP/HIL
Type: Verilog
Library: xil_defaultlib
Size: 3.2 KB
General Properties
Diagram Address Editor IP Catalog Project Summary
Overview | Dashboard
Settings Edit
Project name: HIL_System
Project location: D:/myWorkSpace/SoCFPGA_HIL_PlatformV2/IP/HIL_System
Product family: Zynq UltraScale+
Project part: xczu2cg-sfvc784-1-e
Top module name: HIL_System_wrapper
Target language: Verilog
Simulator language: Mixed
Synthesis
Status: Complete
Messages: 199 warnings
Part: xczu2cg-sfvc784-1-e
Strategy: Vivado Synthesis Defaults
Report Strategy: Vivado Synthesis Default Reports
Incremental synthesis: None
Implementation
Summary | Route Status
Status: Running write_bitstream
Messages: No errors or warnings
Part: xczu2cg-sfvc784-1-e
Strategy: Vivado Implementation Defaults
Report Strategy: Vivado Implementation Default Reports
Incremental implementation: None
DRC Violations
No DRC violations were found.
Implemented DRC Report
Timing
Setup | Hold | Pulse Width
Worst Negative Slack (WNS): 4.945 ns
Total Negative Slack (TNS): 0 ns
Tcl Console Messages Log Reports Design Runs
launch_runs synth_1 -jobs 8
[Sun Dec 31 15:23:38 2023] Launched synth_1...
Run output will be captured here: D:/myWorkSpace/SoCFPGA_HIL_PlatformV2/IP/HIL_System/HIL_System.runs/synth_1/runme.log
launch_runs impl_1 -jobs 8
[Sun Dec 31 15:29:03 2023] Launched impl_1...
Run output will be captured here: D:/myWorkSpace/SoCFPGA_HIL_PlatformV2/IP/HIL_System/HIL_System.runs/impl_1/runme.log
launch_runs impl_1 -to_step write_bitstream -jobs 8
[Sun Dec 31 15:35:28 2023] Launched impl_1...
Run output will be captured here: D:/myWorkSpace/SoCFPGA_HIL_PlatformV2/IP/HIL_System/HIL_System.runs/impl_1/runme.log
Type a Tcl command here

.xsa


File Edit Flow Tools Reports Window Layout View Help
Quick Access
write_bitstream Complete
Dashboard
Default Layout
Project
Add Sources... Alt+A
Close Project
Close Hardware Manager
Constraints
Simulation Waveform
Checkpoint
IP
Text Editor
Import
Export
Print... Ctrl+P
Exit
Export Hardware...
Export Bitstream File...
Export Simulation...
Export ILA Data...
No hardware target is open. Open target
Project Summary
Overview | Dashboard
Settings Edit
Project name: HIL_System
Project location: D:/myWorkSpace/SoCFPGA_HIL_PlatformV2/IP/HIL_System
Product family: Zynq UltraScale+
Project part: xczu2cg-sfvc784-1-e
Top module name: HIL_System_wrapper
Target language: Verilog
Simulator language: Mixed
Synthesis
Status: Complete
Messages: 199 warnings
Part: xczu2cg-sfvc784-1-e
Strategy: Vivado Synthesis Defaults
Report Strategy: Vivado Synthesis Default Reports
Incremental synthesis: None
Implementation
Summary | Route Status
Status: Complete
Messages: No errors or warnings
Part: xczu2cg-sfvc784-1-e
Strategy: Vivado Implementation Defaults
Report Strategy: Vivado Implementation Default Reports
Incremental implementation: None
DRC Violations
No DRC violations were found.
Implemented DRC Report
Timing
Setup | Hold | Pulse Width
Worst Negative Slack (WNS): 4.945 ns
Total Negative Slack (TNS): 0 ns
Number of Failing Endpoints: 0
Total Number of Endpoints: 25751
Source File Properties
HIL_System_wrapper.v
Enabled
Location: D:/myWorkSpace/SoCFPGA_HIL_Plat
Type: Verilog
Library: xil_defaultlib
Size: 3.2 KB
Modified: Thursday 22/05/05 09:01:56 PM
Copied to: D:/myWorkSpace/SoCFPGA_HIL_Plat
General Properties
Tcl Console Messages Serial I/O Links Serial I/O Scans
Warning (199) Info (541) Status (362) Show All
[Vivado 12-3200] Please refer to the DRC report (report_drc) for more information.
[Designutils 20-2272] Running write_bitstream with 2 threads.
[Vivado 12-1842] Bitgen Completed Successfully.
[Project 1-120] WebTalk data collection is mandatory when using a WebPACK part without a full Vivado license. To see the specific WebTalk data collected for your design, open the usage_statistics_webtalk.html or usage_statistics_webtalk.xml file in the implementation directory.
[Common 17-83] Releasing license: Implementation
Run Simulation
RTL ANALYSIS
Open Elaborated Design
SYNTHESIS
Run Synthesis
Open Synthesized Design
IMPLEMENTATION
Run Implementation
Open Implemented Design
PROGRAM AND DEBUG
Generate Bitstream
Open Hardware Manager
输出一个硬件说明文件夹和SDK一起运用


Export Hardware Platform
VIVADO
HLx Editions
Export Hardware Platform
This wizard will guide you through the export of a hardware platform for use in the Vitis or PetaLinux software tools.
To export a hardware platform, you will need to provide a name and location for the exported file and specify the platform properties.
Platform type
Fixed
A platform supporting embedded software development only.
Expandable
A platform supporting acceleration.
XILINX
< Back
Next >
Finish
Cancel


Export Hardware Platform
Output
Set the platform properties to inform downstream tools of the intended use of the target platform's hardware design.
Pre-synthesis
This platform includes a hardware specification for downstream software tools.
Include bitstream
This platform includes the complete hardware implementation and bitstream, in addition to the hardware specification for software tools.
< Back
Next >
Finish
Cancel


Export Hardware Platform
Files
Enter the name of your hardware platform file, and the directory where the XSA file will be stored.
XSA file name: HIL_System_wrapper
Export to: D:/myWorkSpace/SoCFPGA_HIL_PlatformV2/IP/HIL_System
The XSA will be written to: D:\myWorkSpace\SoCFPGA_HIL_PlatformV2\IP\HIL_System\HIL_System_wrapper.xsa
记住这个路径，稍后要导入该文件到vitis工程中
< Back
Next >
Finish
Cancel

4.3

4.3.1 1.

MATLAB 2021a PL `PL.prj`

`.\Tools\CopyModelCode2VitisPrj.m` vitis

```
vitisPrjPath = 'D:\myWorkSpace\RflySimCourse\socfpga-hil-vitis\';
unzip('code\Model.zip','Tools\Model')
delete .\Tools\Model\code\Model_ert_rtw\ert_main.c  .\Tools\Model\code\Model_ert_rtw\buildInfo.mat  .\Tools\Model\code\Mod
copyfile('.\Tools\Model',[vitisPrjPath,'HIL_APP\src\TaskINT\Model'])
rmdir Tools\Model s
```

`vitisPrjPath` Vitis

4.3.2 2.

Xilinx

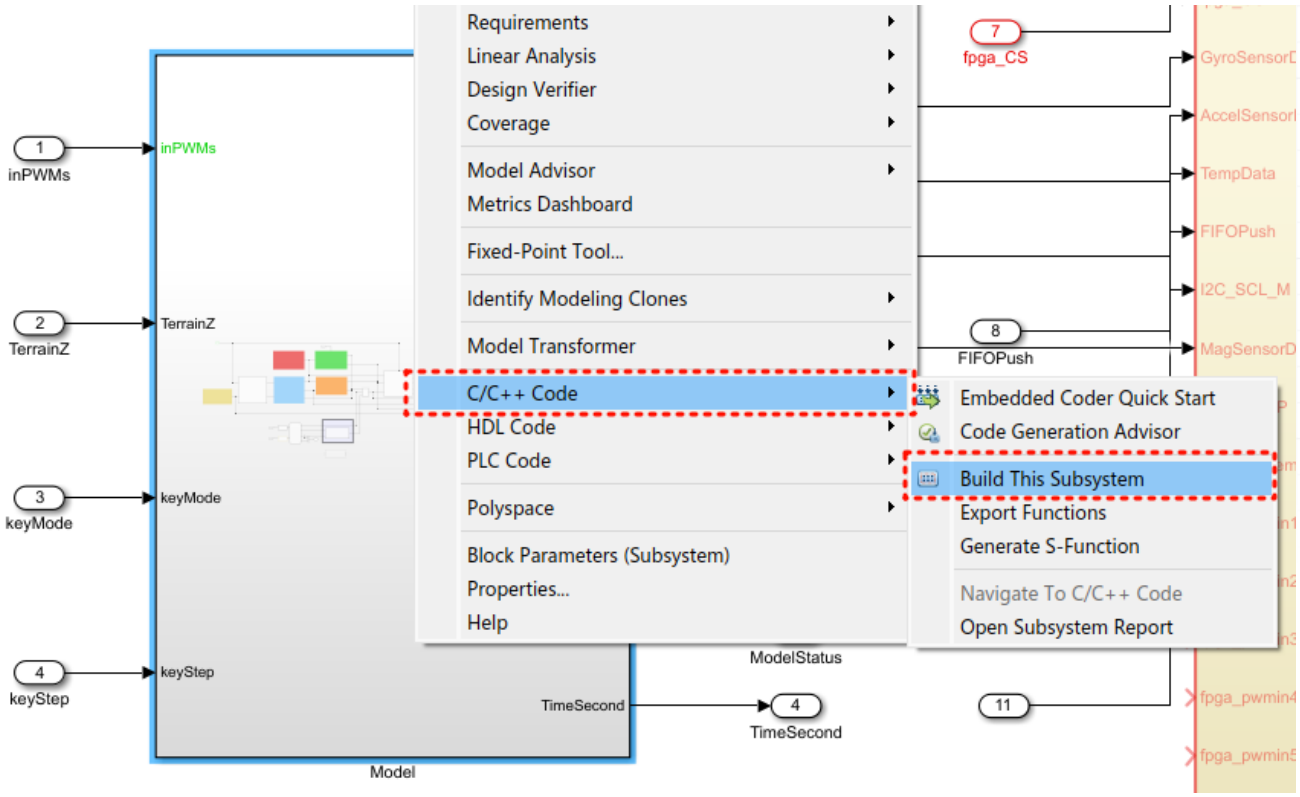
```
hdlsetuptoolpath('ToolName','Xilinx Vivado','ToolPath','D:\Xilinx\Vivado\2020.1\bin\vivado.bat');
```

- PL\Model\H250DegradedParamInit4S.m
- PL\HIL_System\SensorParamInit.m

- PL\HIL_System\HIL_System.slx

  model

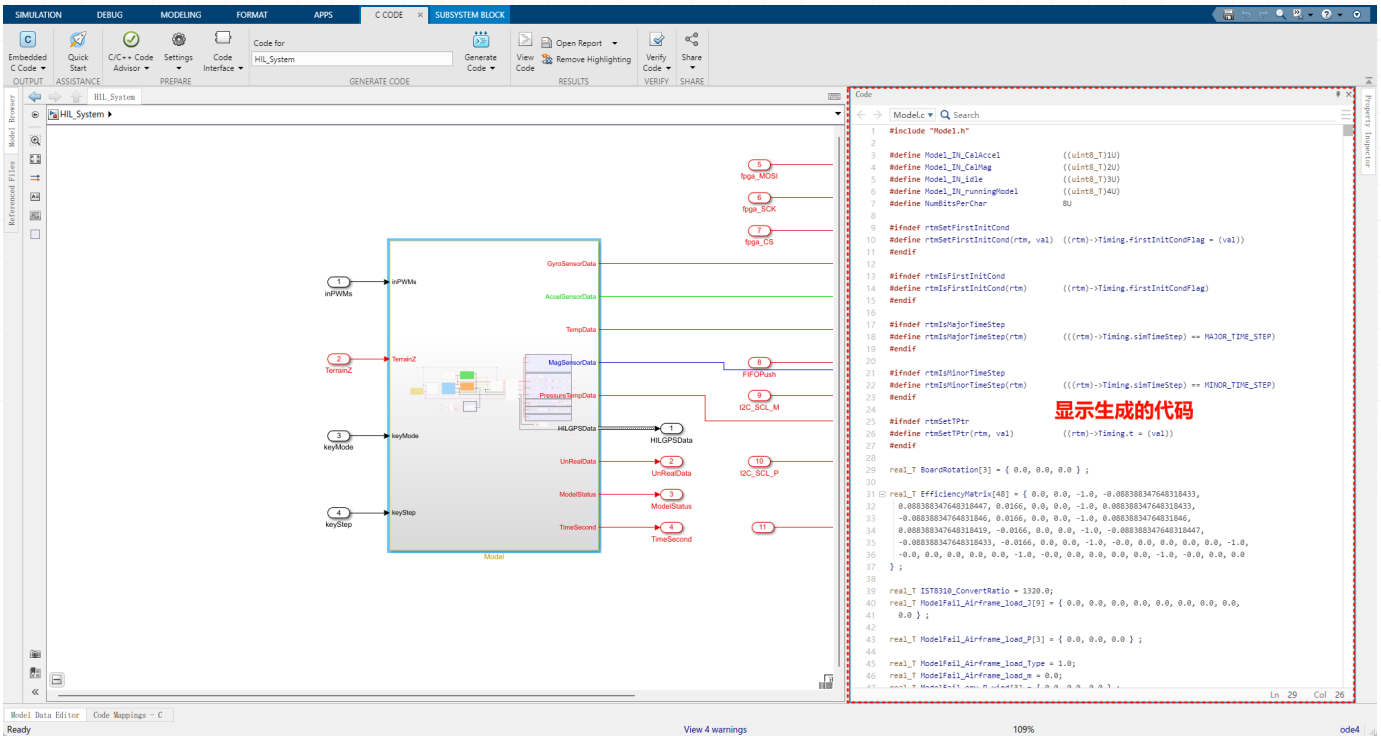
-


Requirements
Linear Analysis
Design Verifier
Coverage
Model Advisor
Metrics Dashboard
Fixed-Point Tool...
Identify Modeling Clones
Model Transformer
C/C++ Code
HDL Code
PLC Code
Polyspace
Block Parameters (Subsystem)
Properties...
Help
Embedded Coder Quick Start
Code Generation Advisor
Build This Subsystem
Export Functions
Generate S-Function
Navigate To C/C++ Code
Open Subsystem Report
1
inPWMs
2
TerrainZ
3
keyMode
4
keyStep
inPWMs
TerrainZ
keyMode
keyStep
TimeSecond
Model
ModelStatus
4
TimeSecond
7
fpga_CS
8
FIFOPush
11
GyroSensorD
AccelSensorI
TempData
FIFOPush
I2C_SCL_M
MagSensorD
fpga_pwmin4
fpga_pwmin5

Inlined


Build code for Subsystem:Model
Pick tunable parameters
Variable Name
Class
Storage Class
BoardRotation double ExportedGlobal
EfficiencyMatrix double ExportedGlobal
IST8310_ConvertRatio double ExportedGlobal
ModelFail_Airframe_load_J double ExportedGlobal
ModelFail_Airframe_load_P double ExportedGlobal
ModelFail_Airframe_load_Type double ExportedGlobal
ModelFail_Airframe_load_m double ExportedGlobal
ModelFail_env_P_wind double ExportedGlobal
ModelFail_motor_isEnable double ExportedGlobal
ModelFail_motor_kCt double ExportedGlobal
ModelFail_motor_kTc double ExportedGlobal
ModelFail_motor_kw double ExportedGlobal
ModelInit_AngEuler double ExportedGlobal
ModelInit_PosE double ExportedGlobal
ModelInit_RateB double ExportedGlobal
ModelInit_VelB double ExportedGlobal
除了模型采样时间以外
其他参数可设置为ExportedGlobal
Blocks using selected variable
Block
Parent
在脚本中定义的参数
Build
Cancel
Help
Status
Select tunable parameters and click Build
设置完成后点击build即可

-

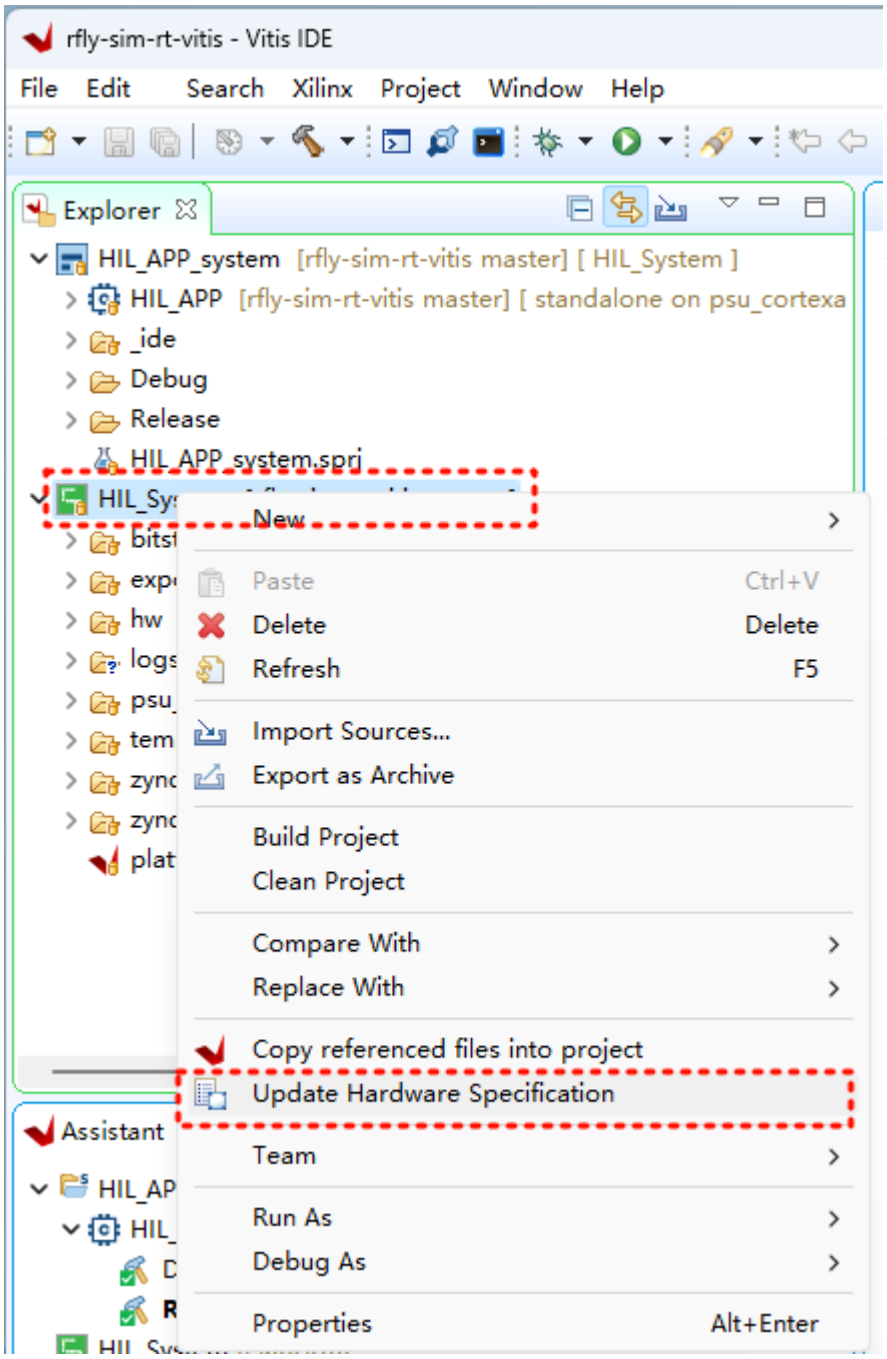


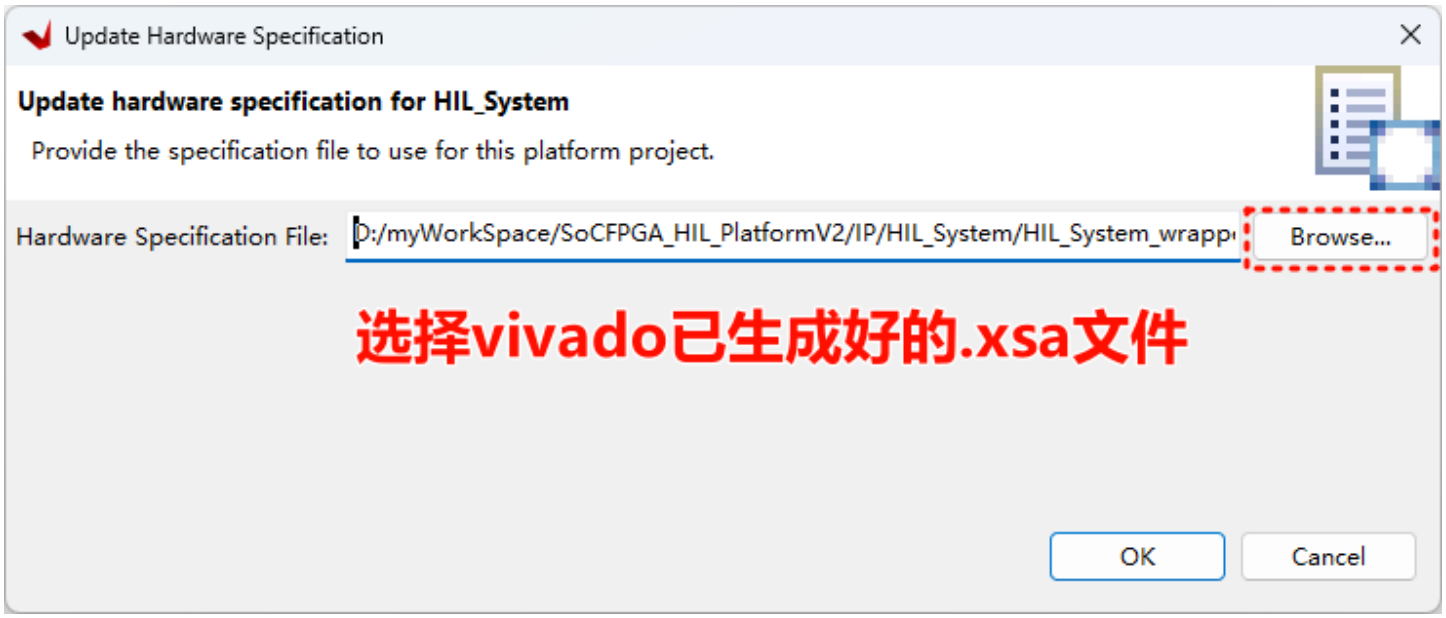
- PL\Tools\CopyModelCode2VitisPrj.m

  Vitis HIL_System.slx

4.3.3 3. .xsa

FPGA FPGA xsa FPGA xsa vitis


rfly-sim-rt-vitis - Vitis IDE
File Edit Search Xilinx Project Window Help
Explorer
HIL_APP_system [rfly-sim-rt-vitis master] [HIL_System]
HIL_APP [rfly-sim-rt-vitis master] [standalone on psu_cortexa
_ide
Debug
Release
HIL_APP_system.sprj
New
Paste Ctrl+V
Delete Delete
Refresh F5
Import Sources...
Export as Archive
Build Project
Clean Project
Compare With
Replace With
Copy referenced files into project
Update Hardware Specification
Team
Run As
Debug As
Properties Alt+Enter
Assistant

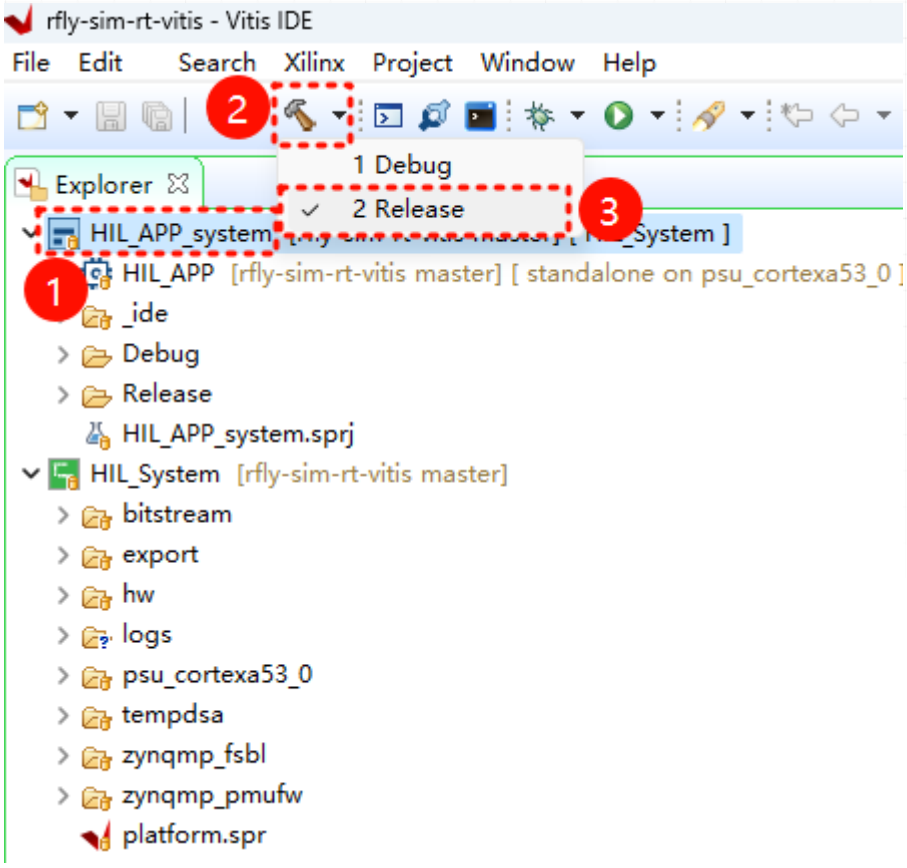


4.3.4 3.

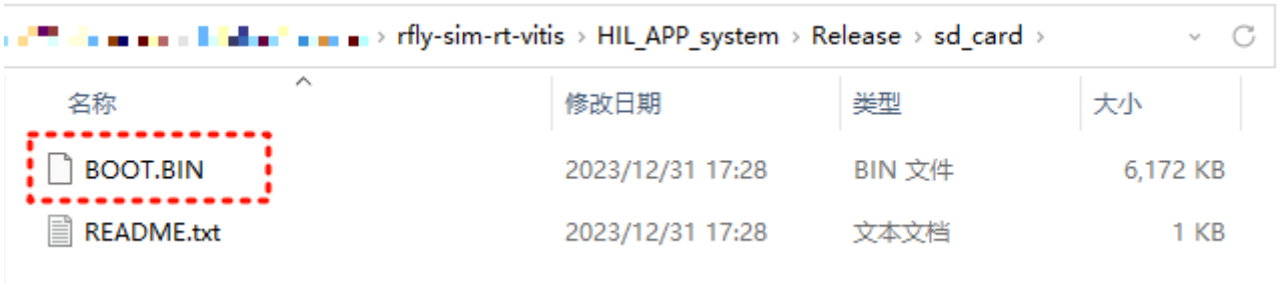
Vitis 2020.1 Vitis

Note

HIL_APP_system Release



BOOT.BIN



BOOT.BIN SD

4.4

vitis

`rfly-sim-rt-vitis\HIL_APP\src\Config\config.c` ADD_PARAM()

c

```
const ModelParam modelParamList[] =
{
//
        ADD_PARAM(Mass,&ModelParam_Airframe_m,1),
        ADD_PARAM(C_md,ModelParam_envC_md,3),
        ADD_PARAM(J,ModelParam_Airframe_J,9),
        ADD_PARAM(motorCr,&ModelParam_motorCr,1),
        ADD_PARAM(motorFitType,&ModelParam_motorFitType,1),
        ADD_PARAM(motorJm,&ModelParam_motorJm,1),
        ADD_PARAM(motorMinThr,&ModelParam_motorMinThr,1),
        ADD_PARAM(motorRateCurveCoeffi,ModelParam_motorRateCurveCoeffi,3),
        ADD_PARAM(motorTc,&ModelParam_motorTc,1),
        ADD_PARAM(motorWb,&ModelParam_motorWb,1),
        ADD_PARAM(rotorCt,&ModelParam_rotorCt,1),
        ADD_PARAM(NoiseVarAcc0, ModelParam_NoiseVarAcc0,3),
        ADD_PARAM(NoiseVarGyro0, ModelParam_NoiseVarGyro0,3),
        ADD_PARAM(NoiseVarMag0, ModelParam_NoiseVarMag0,3),
        ADD_PARAM(PositionAcc0, ModelParam_PositionAcc0,3),
        ADD_PARAM(DisplayUAVType,&RflySimDisplayUAVType,1),
        ADD_PARAM(CopterID,&RflySimCopterID,1),
        ADD_PARAM(RotorDirection, RotorDirectionVector, 8),
        ADD_PARAM(EfficiencyMatrix, EfficiencyMatrix, 48),
//
        ADD_PARAM(BoardRotation, BoardRotation,3),
        ADD_PARAM(IST8310_ConvertRatio, &IST8310_ConvertRatio,1),
        ADD_PARAM(Using_OneShot, &Using_OneShot,1)
};
```

ADD_PARAM()

```
#define ADD_PARAM(_name, _addr, _len)\
        {\
                .name = #_name,\
                .addr = (double *)_addr, \
                .len = _len \
        }
```

ADD_PARAM()

```
ADD_PARAM(Using_OneShot, &Using_OneShot,1)
```

c\c++

```
ADD_PARAM(BoardRotation, BoardRotation,3)
```

___ json json json

5.

5.1

MkDocs WSL

1. mkdocs.org

1.1 `pip install mkdocs-pdf-export-plugin` `mkdocs.yml`

```
plugins:
    - pdf-export
```

1.2 `pip install pymdown-extensions`

1.3 `pip install markdown-callouts` https://github.com/sondregronas/mkdocs-callouts

1.4 `pip install mdx-gh-links`

1.5 `pip install mkdocs-click`

1.6 `pip install mkdocs-autorefs`

1.7 `pip install mkdocstrings` `pip install 'mkdocstrings[crystal,python]'`

1.8 `pip install mkdocs-gitbook`

1.9 `pip install mkdocs-with-pdf`

```
plugins:
    - with-pdf
```

1.10 `pip install markdown-checklist`

```
markdown_extensions:
    - markdown_checklist.extension
```

1.11 `pip install mkdocs-video`

```
plugins:
    - mkdocs-video
```

1.12 `pip install mkdocs-git-revision-date-localized-plugin`

```
plugins:
    - git-revision-date-localized
```

1.13 `pip install markdown-captions` `pip install mkdocs-video`

```
markdown_extensions:
    - markdown_captions
```

1.14 `pip install mkdocs-resize-images`

```
plugins:
    - resize-images
```

1.15 `pip install mkdocs-git-latest-changes-plugin`

```
plugins:
    - git-latest-changes
```

1.16 `pip install mkdocs-latest-release-plugin`

```
plugins:
    - git-latest-release
```

2. Git

2.1 WSL Git

```
git config --global user.name "Your Name"
git config --global user.email "email@example.com"
```

2.2 ssh key

```
ssh-keygen -t ed25519 -C "your_email@example.com"
```

```
ssh-keygen -t rsa -b 4096 -C "your_email@example.com"
```

```
Your public key has been saved in /home/kcx064/.ssh/id_ed25519.pub
```

cat .pub

```
cat /home/kcx064/.ssh/id_ed25519.pub
```

github SSH keys

2.3 Git lfs

```
sudo apt-get install git-lfs
```

3.

3.1 `git clone https://github.com/RflyBUAA/RflySimRTDoc.git`

3.2 `git checkout master` (`gh-pages`, `gh-pages`)

3.3 markdown

3.4 `mkdocs build` / `site`

3.5 `mkdocs serve`

3.6 `mkdocs gh-deploy` site

3.7 `mkdocs gh-deploy` push `git push origin gh-pages`

3.8 master

6.

6.1 MkDocs Plugins

A Guide to installing, using and creating MkDocs Plugins

6.1.1 Installing Plugins

Before a plugin can be used, it must be installed on the system. If you are using a plugin which comes with MkDocs, then it was installed when you installed MkDocs. However, to install third party plugins, you need to determine the appropriate package name and install it using `pip`:

```
pip install mkdocs-foo-plugin
```

> **Warning**
>
> Installing an MkDocs plugin means installing a Python package and executing any code that the author has put in there. So, exercise the usual caution; there's no attempt at sandboxing.

Once a plugin has been successfully installed, it is ready to use. It just needs to be enabled in the configuration file. The Catalog repository has a large ranked list of plugins that you can install and use.

6.1.2 Using Plugins

The `plugins` configuration option should contain a list of plugins to use when building the site. Each "plugin" must be a string name assigned to the plugin (see the documentation for a given plugin to determine its "name"). A plugin listed here must already be installed.

```
plugins:
  - search
```

Some plugins may provide configuration options of their own. If you would like to set any configuration options, then you can nest a key/value mapping (`option_name: option value`) of any options that a given plugin supports. Note that a colon (`:`) must follow the plugin name and then on a new line the option name and value must be indented and separated by a colon. If you would like to define multiple options for a single plugin, each option must be defined on a separate line.

```
plugins:
  - search:
      lang: en
      foo: bar
```

For information regarding the configuration options available for a given plugin, see that plugin's documentation.

For a list of default plugins and how to override them, see the configuration documentation.

6.1.3 Developing Plugins

Like MkDocs, plugins must be written in Python. It is generally expected that each plugin would be distributed as a separate Python module, although it is possible to define multiple plugins in the same module. At a minimum, a MkDocs Plugin must consist of a BasePlugin subclass and an entry point which points to it.

BasePlugin

A subclass of `mkdocs.plugins.BasePlugin` should define the behavior of the plugin. The class generally consists of actions to perform on specific events in the build process as well as a configuration scheme for the plugin.

All `BasePlugin` subclasses contain the following attributes:

config_scheme

A tuple of configuration validation instances. Each item must consist of a two item tuple in which the first item is the string name of the configuration option and the second item is an instance of `mkdocs.config.config_options.BaseConfigOption` or any of its subclasses.

For example, the following `config_scheme` defines three configuration options: `foo`, which accepts a string; `bar`, which accepts an integer; and `baz`, which accepts a boolean value.

```
class MyPlugin(mkdocs.plugins.BasePlugin):
    config_scheme = (
        ('foo', mkdocs.config.config_options.Type(str, default='a default value')),
        ('bar', mkdocs.config.config_options.Type(int, default=0)),
        ('baz', mkdocs.config.config_options.Type(bool, default=True))
    )
```

New in version 1.4

Subclassing `Config` to specify the config schema

To get type safety benefits, if you're targeting only MkDocs 1.4+, define the config schema as a class instead:

```
class MyPluginConfig(mkdocs.config.base.Config):
    foo = mkdocs.config.config_options.Type(str, default='a default value')
    bar = mkdocs.config.config_options.Type(int, default=0)
    baz = mkdocs.config.config_options.Type(bool, default=True)

class MyPlugin(mkdocs.plugins.BasePlugin[MyPluginConfig]):
    ...
```

Examples of config definitions

- **Example**

```
from mkdocs.config import base, config_options as c

class _ValidationOptions(base.Config):
    enabled = c.Type(bool, default=True)
    verbose = c.Type(bool, default=False)
    skip_checks = c.ListOfItems(c.Choice(('foo', 'bar', 'baz')), default=[])

class MyPluginConfig(base.Config):
    definition_file = c.File(exists=True)  # required
    checksum_file = c.Optional(c.File(exists=True))  # can be None but must exist if specified
    validation = c.SubConfig(_ValidationOptions)
```

From the user's point of view `SubConfig` is similar to `Type(dict)`, it's just that it also retains full ability for validation: you define all valid keys and what each value should adhere to.

And `ListOfItems` is similar to `Type(list)`, but again, we define the constraint that each value must adhere to.

This accepts a config as follows:

```
my_plugin:
  definition_file: configs/test.ini  # relative to mkdocs.yml
  validation:
    enabled: !ENV [CI, false]
    verbose: true
    skip_checks:
      - foo
      - baz
```

- **Example**

```
import numbers
from mkdocs.config import base, config_options as c

class _Rectangle(base.Config):
    width = c.Type(numbers.Real)  # required
    height = c.Type(numbers.Real)  # required

class MyPluginConfig(base.Config):
    add_rectangles = c.ListOfItems(c.SubConfig(_Rectangle))  # required
```

In this example we define a list of complex items, and that's achieved by passing a concrete `SubConfig` to `ListOfItems`.

This accepts a config as follows:

```
my_plugin:
  add_rectangles:
    - width: 5
      height: 7
    - width: 12
      height: 2
```

When the user's configuration is loaded, the above scheme will be used to validate the configuration and fill in any defaults for settings not provided by the user. The validation classes may be any of the classes provided in `mkdocs.config.config_options` or a third party subclass defined in the plugin.

Any settings provided by the user which fail validation or are not defined in the `config_scheme` will raise a `mkdocs.config.base.ValidationError`.

config

A dictionary of configuration options for the plugin, which is populated by the `load_config` method after configuration validation has completed. Use this attribute to access options provided by the user.

```
def on_pre_build(self, config, **kwargs):
    if self.config['baz']:
        # implement "baz" functionality here...
```

New in version 1.4

Safe attribute-based access

To get type safety benefits, if you're targeting only MkDocs 1.4+, access options as attributes instead:

```
def on_pre_build(self, config, **kwargs):
    if self.config.baz:
        print(self.config.bar ** 2)  # OK, `int ** 2` is valid.
```

All `BasePlugin` subclasses contain the following method(s):

load_config(options)

Loads configuration from a dictionary of options. Returns a tuple of `(errors, warnings)`. This method is called by MkDocs during configuration validation and should not need to be called by the plugin.

on_<event_name>()

Optional methods which define the behavior for specific events. The plugin should define its behavior within these methods. Replace `<event_name>` with the actual name of the event. For example, the `pre_build` event would be defined in the `on_pre_build` method.

Most events accept one positional argument and various keyword arguments. It is generally expected that the positional argument would be modified (or replaced) by the plugin and returned. If nothing is returned (the method returns `None`), then the original, unmodified object is used. The keyword arguments are simply provided to give context and/or supply data which may be used to determine how the positional argument should be modified. It is good practice to accept keyword arguments as `**kwargs`. In the event that additional keywords are provided to an event in a future version of MkDocs, there will be no need to alter your plugin.

For example, the following event would add an additional static_template to the theme config:

```
class MyPlugin(BasePlugin):
    def on_config(self, config, **kwargs):
        config['theme'].static_templates.add('my_template.html')
        return config
```

New in version 1.4

To get type safety benefits, if you're targeting only MkDocs 1.4+, access config options as attributes instead:

```
def on_config(self, config: MkDocsConfig):
    config.theme.static_templates.add('my_template.html')
    return config
```

Events

There are three kinds of events: Global Events, Page Events and Template Events.

- **See a diagram with relations between all the plugin events**
    - The events themselves are shown in yellow, with their parameters.
    - Arrows show the flow of arguments and outputs of each event. Sometimes they're omitted.
    - The events are chronologically ordered from top to bottom.
    - Dotted lines appear at splits from global events to per-page events.
    - Click the events' titles to jump to their description.

One-time Events

One-time events run once per `mkdocs` invocation. The only case where these tangibly differ from global events is for `mkdocs serve`: global events, unlike these, will run multiple times -- once per build.

on_startup

::: mkdocs.plugins.BasePlugin.on_startup
options:
show_root_heading: false
show_root_toc_entry: false

on_shutdown

::: mkdocs.plugins.BasePlugin.on_shutdown
options:
show_root_heading: false
show_root_toc_entry: false

on_serve

::: mkdocs.plugins.BasePlugin.on_serve
options:
show_root_heading: false
show_root_toc_entry: false

Global Events

Global events are called once per build at either the beginning or end of the build process. Any changes made in these events will have a global effect on the entire site.

on_config

::: mkdocs.plugins.BasePlugin.on_config
options:
show_root_heading: false
show_root_toc_entry: false

on_pre_build

::: mkdocs.plugins.BasePlugin.on_pre_build
options:
show_root_heading: false
show_root_toc_entry: false

on_files

::: mkdocs.plugins.BasePlugin.on_files
options:
show_root_heading: false
show_root_toc_entry: false

on_nav

::: mkdocs.plugins.BasePlugin.on_nav
options:
show_root_heading: false
show_root_toc_entry: false

on_env

::: mkdocs.plugins.BasePlugin.on_env
options:
show_root_heading: false
show_root_toc_entry: false

on_post_build

::: mkdocs.plugins.BasePlugin.on_post_build
options:
show_root_heading: false
show_root_toc_entry: false

on_build_error

::: mkdocs.plugins.BasePlugin.on_build_error
options:
show_root_heading: false
show_root_toc_entry: false

Template Events

Template events are called once for each non-page template. Each template event will be called for each template defined in the extra_templates config setting as well as any static_templates defined in the theme. All template events are called after the env event and before any page events.

on_pre_template

::: mkdocs.plugins.BasePlugin.on_pre_template
options:
show_root_heading: false
show_root_toc_entry: false

on_template_context

::: mkdocs.plugins.BasePlugin.on_template_context
options:
show_root_heading: false
show_root_toc_entry: false

on_post_template

::: mkdocs.plugins.BasePlugin.on_post_template
options:
show_root_heading: false
show_root_toc_entry: false

Page Events

Page events are called once for each Markdown page included in the site. All page events are called after the post_template event and before the post_build event.

on_pre_page

::: mkdocs.plugins.BasePlugin.on_pre_page
options:
show_root_heading: false
show_root_toc_entry: false

on_page_read_source

::: mkdocs.plugins.BasePlugin.on_page_read_source
options:
show_root_heading: false
show_root_toc_entry: false

on_page_markdown

::: mkdocs.plugins.BasePlugin.on_page_markdown
options:
show_root_heading: false
show_root_toc_entry: false

on_page_content

::: mkdocs.plugins.BasePlugin.on_page_content
options:
show_root_heading: false
show_root_toc_entry: false

on_page_context

::: mkdocs.plugins.BasePlugin.on_page_context
options:
show_root_heading: false
show_root_toc_entry: false

on_post_page

::: mkdocs.plugins.BasePlugin.on_post_page
options:
show_root_heading: false
show_root_toc_entry: false

Event Priorities

For each event type, corresponding methods of plugins are called in the order that the plugins appear in the `plugins` config.

Since MkDocs 1.4, plugins can choose to set a priority value for their events. Events with higher priority are called first. Events without a chosen priority get a default of 0. Events that have the same priority are ordered as they appear in the config.

::: mkdocs.plugins.event_priority

There may also arise a need to register a handler for the same event at multiple different priorities.

`CombinedEvent` makes this possible since MkDocs 1.6.

::: mkdocs.plugins.CombinedEvent

Handling Errors

MkDocs defines four error types:

::: mkdocs.exceptions.MkDocsException
::: mkdocs.exceptions.ConfigurationError
::: mkdocs.exceptions.BuildError
::: mkdocs.exceptions.PluginError

Unexpected and uncaught exceptions will interrupt the build process and produce typical Python tracebacks, which are useful for debugging your code. However, users generally find tracebacks overwhelming and often miss the helpful error message. Therefore, MkDocs will catch any of the errors listed above, retrieve the error message, and exit immediately with only the helpful message displayed to the user.

Therefore, you might want to catch any exceptions within your plugin and raise a `PluginError`, passing in your own custom-crafted message, so that the build process is aborted with a helpful message.

The on_build_error event will be triggered for any exception.

For example:

```
from mkdocs.exceptions import PluginError
from mkdocs.plugins import BasePlugin


class MyPlugin(BasePlugin):
    def on_post_page(self, output, page, config, **kwargs):
        try:
            # some code that could throw a KeyError
            ...
        except KeyError as error:
            raise PluginError(f"Failed to find the item by key: '{error}'")

    def on_build_error(self, error, **kwargs):
        # some code to clean things up
        ...
```

Logging in plugins

To ensure that your plugins' log messages adhere with MkDocs' formatting and `--verbose` / `--debug` flags, please write the logs to a logger under the `mkdocs.plugins.` namespace.

Example

```
import logging

log = logging.getLogger(f"mkdocs.plugins.{__name__}")

log.warning("File '%s' not found. Breaks the build if --strict is passed", my_file_name)
log.info("Shown normally")
log.debug("Shown only with `--verbose`")

if log.getEffectiveLevel() <= logging.DEBUG:
    log.debug("Very expensive calculation only for debugging: %s", get_my_diagnostics())
```

`log.error()` is another logging level that is differentiated by its look, but in all other ways it functions the same as `warning`, so it's strange to use it. If your plugin encounters an actual error, it is best to just interrupt the build by raising [`mkdocs.exceptions.PluginError`][mkdocs.exceptions.PluginError] (which will also log an ERROR message).

New

New in MkDocs 1.5

MkDocs now provides a `get_plugin_logger()` convenience function that returns a logger like the above that is also prefixed with the plugin's name.

::: mkdocs.plugins.get_plugin_logger

Entry Point

Plugins need to be packaged as Python libraries (distributed on PyPI separate from MkDocs) and each must register as a Plugin via a setuptools `entry_points`. Add the following to your `setup.py` script:

```
entry_points={
    'mkdocs.plugins': [
        'pluginname = path.to.some_plugin:SomePluginClass',
    ]
}
```

The `pluginname` would be the name used by users (in the config file) and `path.to.some_plugin:SomePluginClass` would be the importable plugin itself (`from path.to.some_plugin import SomePluginClass`) where `SomePluginClass` is a

subclass of BasePlugin which defines the plugin behavior. Naturally, multiple Plugin classes could exist in the same module. Simply define each as a separate entry point.

```
entry_points={
    'mkdocs.plugins': [
        'featureA = path.to.my_plugins:PluginA',
        'featureB = path.to.my_plugins:PluginB'
    ]
}
```

Note that registering a plugin does not activate it. The user still needs to tell MkDocs to use it via the config.

Publishing a Plugin

You should publish a package on PyPI, then add it to the Catalog for discoverability. Plugins are strongly recommended to have a unique plugin name (entry point name) according to the catalog.