InHand Networks IG902

Device Open Platform

QUICK START MANUAL

Version: V1.0 Date: 2019.03

InHand Networks
Global Leader in Industrial IoT
www.inhandnetworks.com

Overview

IG902-X device open platform is a development tool based on InHand IG902-X series edge computing gateway. It is a programmable Python open platform integrated with the Python development environment, commonly used Python modules, and system interactive APIs.

This document describes the system environment and development steps required for developing Python apps based on the IG902-X device open platform, instructing you to quickly develop application services using the Python.

Chapter 1 Setting Up the Environment

1. Preparations

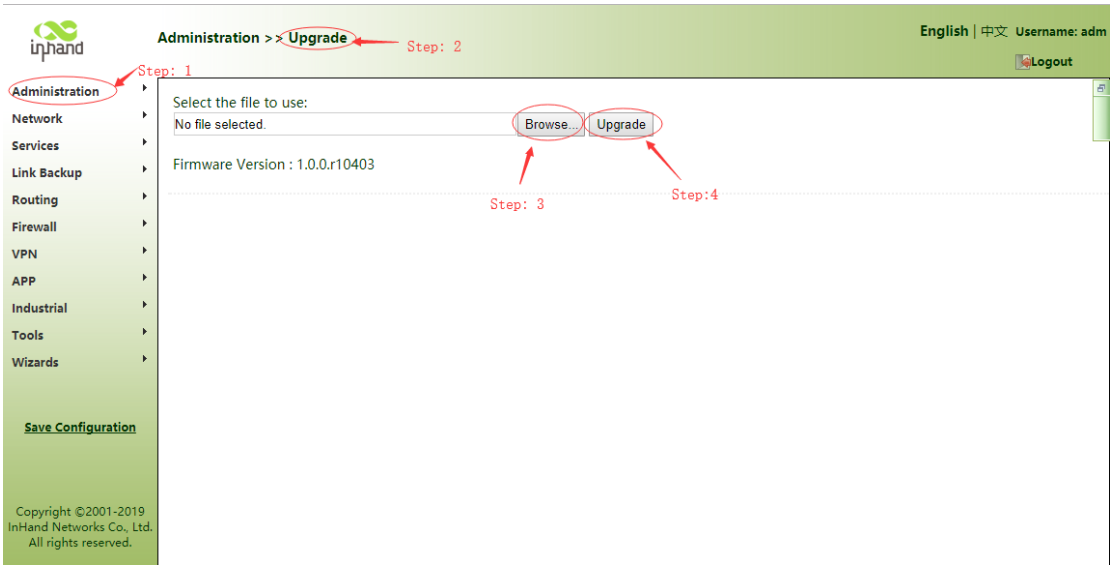
Contact InHand sales or technical support personnel to obtain the following:

- Gateway system firmware (version 10403 or later)
- Python SDK (version 1.0.5 or later)
- IDE (OpenDevice IDE)
- Python 2.7.14 installation package (Windows)

2. Upgrade the System Firmware

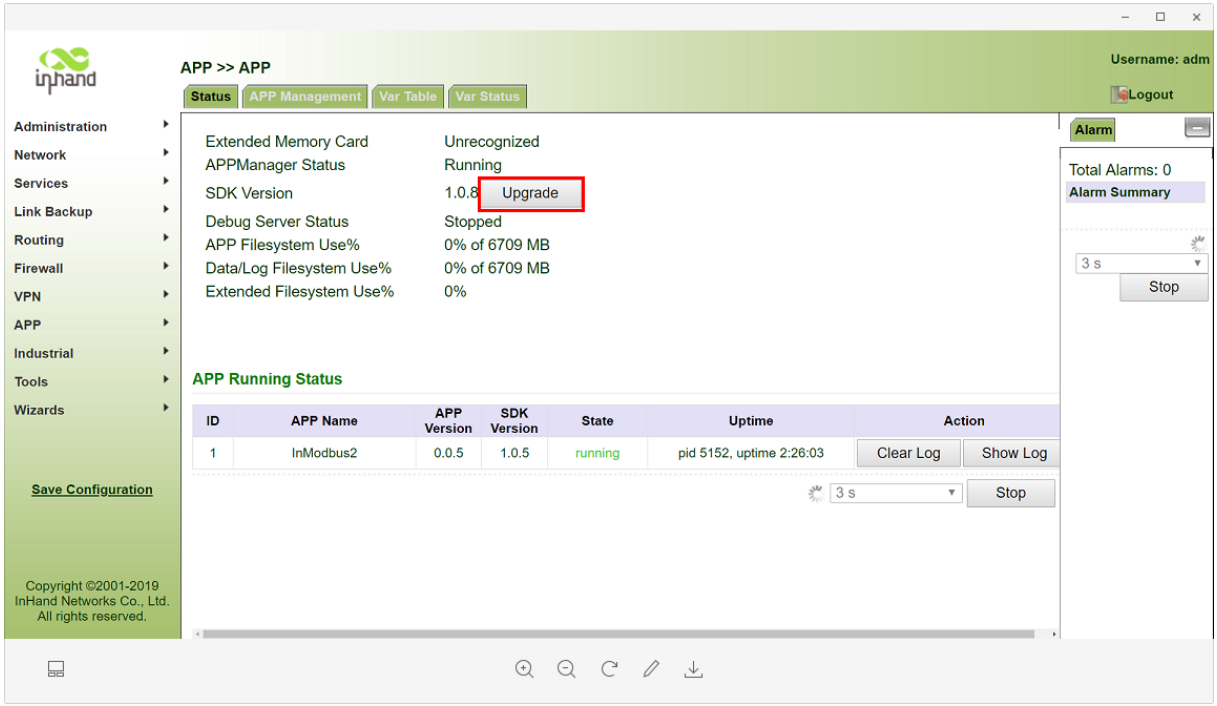
The upgrade procedure is as follows:

Log in to the web page of gateway (default IP address: 192.168.2.1 | user name: adm | password: 123456). Choose **Administration** > **Upgrade** to perform the upgrade. When the timeout period expires, restart the system to complete the upgrade.



3. Install and Upgrade PySDK

On the web page of gateway, choose **APP** > **APP** > **Status**, and click **Upgrade** to install or upgrade the PySDK. After the timeout period expires, the system completes the installation or upgrade, and makes the new version effective.



4. Install IDE Integrated Environment

The IDE integrated environment is developed based on the eclipse plug-in. It provides the functions such as Python 2.7.1X app development, compiling, packaging, and Python pip package management.

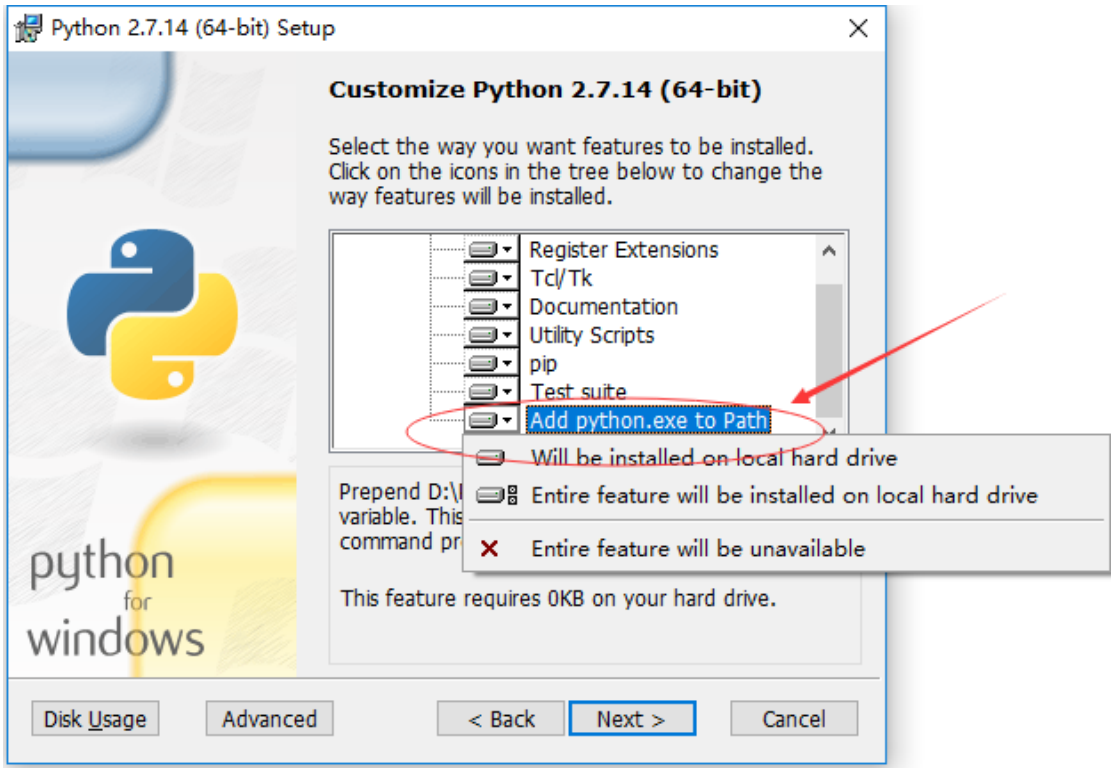
To install IDE, decompress OpenDevice.zip and run **OpenDevice-V***.exe** (* indicates the version number).

5. Install Python on Windows

The local Python 2.7.14 environment on Windows is used to connect IDE to local packages for fast compiling.

To install Python on Windows, download the installation package corresponding to your system from the Python official website https://www.python.org/downloads/release/python-2714/.

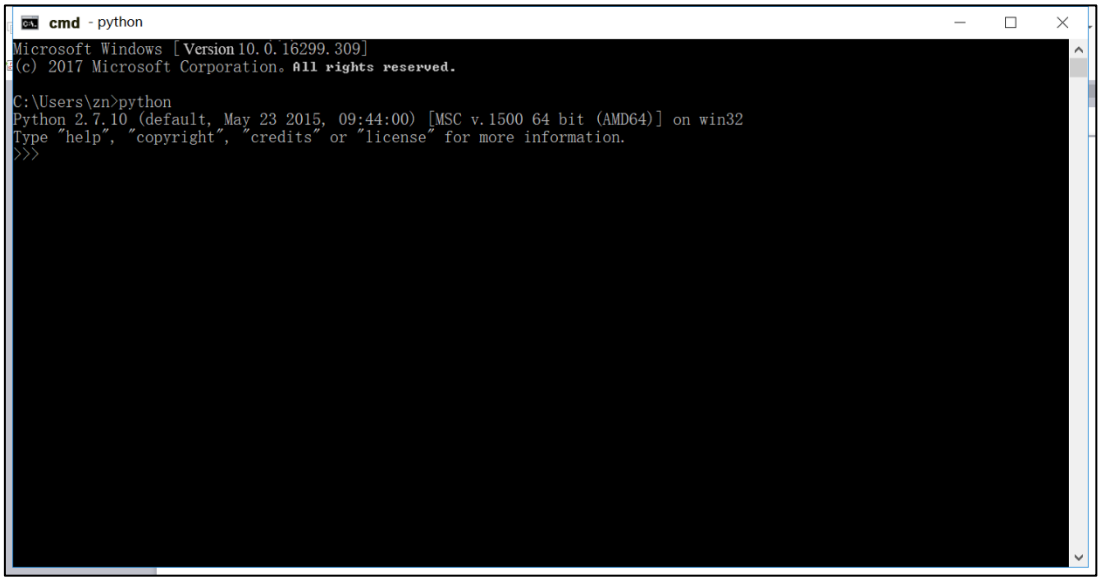
Keep clicking **Next** to retain default settings. When the following interface is displayed, select **Will be installed on local hard drive** to complete the Python installation.



Check the Python environment on PC:

1. Press **Win**+**R** to open the cmd window.

2. Enter **python** and press **Enter**. When the following interface is displayed, the Python environment settings are normal.



Chapter 2 Python App Development

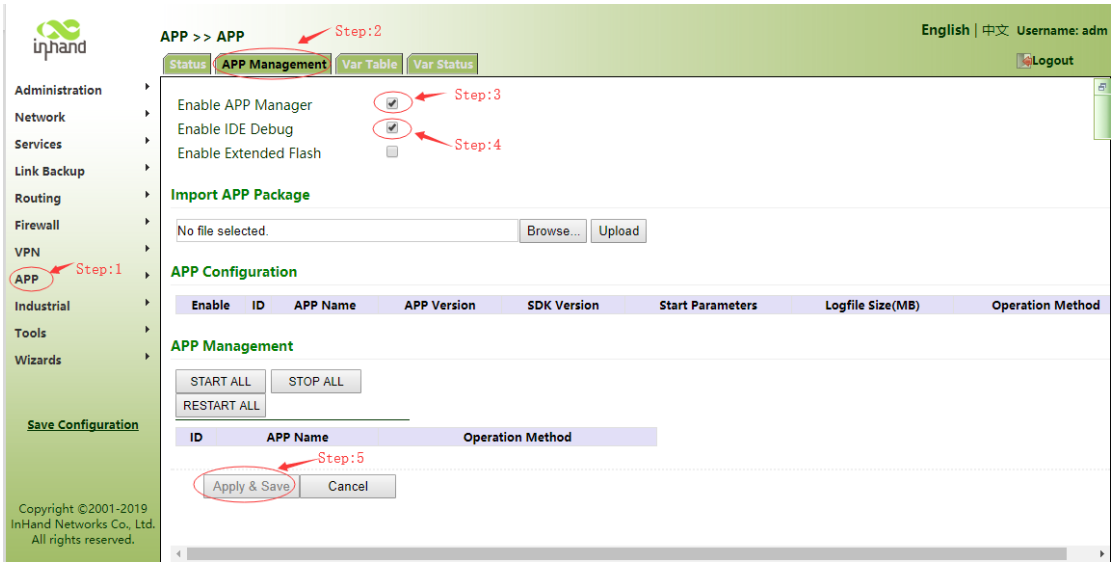
IDE provides the commonly used functions such as code editing, compiling, packaging, and package management. The IDE is available in the form of package. You only need to decompress the package to use IDE.

This chapter describes the steps and precautions of using IDE to develop Python apps.

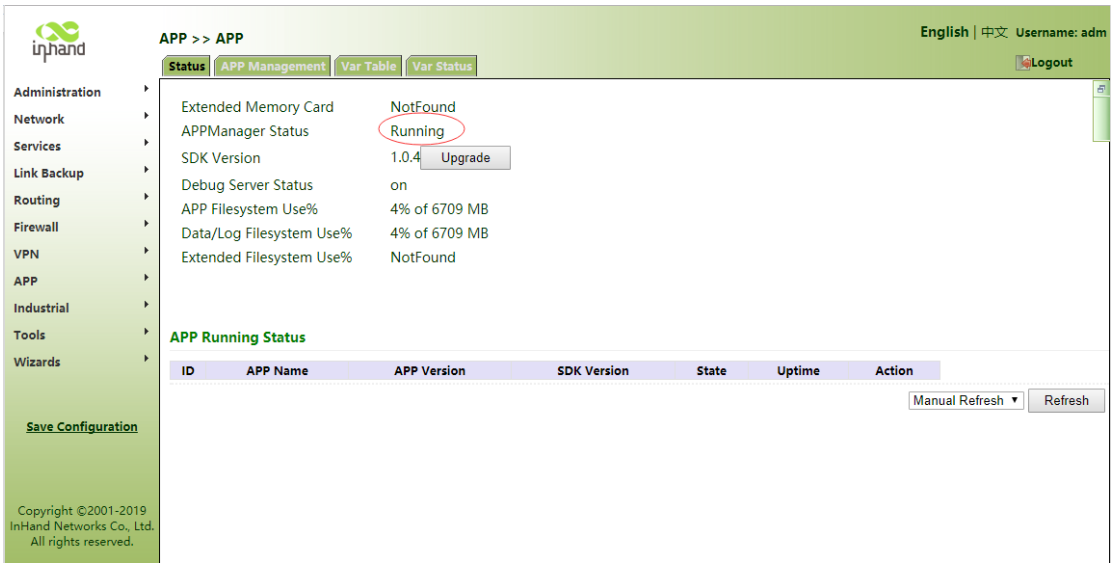
1. Enable Gateway App Management and IDE Debugging

To use all Python app management functions, enable the Python app manager first.

On the web page of gateway, choose **APP** >> **APP** and click **APP Management**. Select **Enable APP Manager** and **Enable IDE Debug**, and click **Apply & Save**.

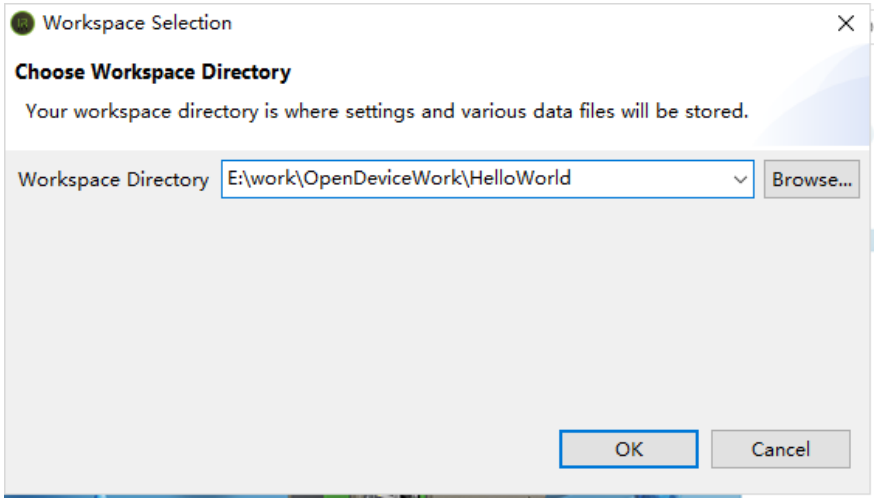


To view the app running status, choose **APP** > **APP** > **Status**. If the status is displayed as **Running**, the app is running normally.

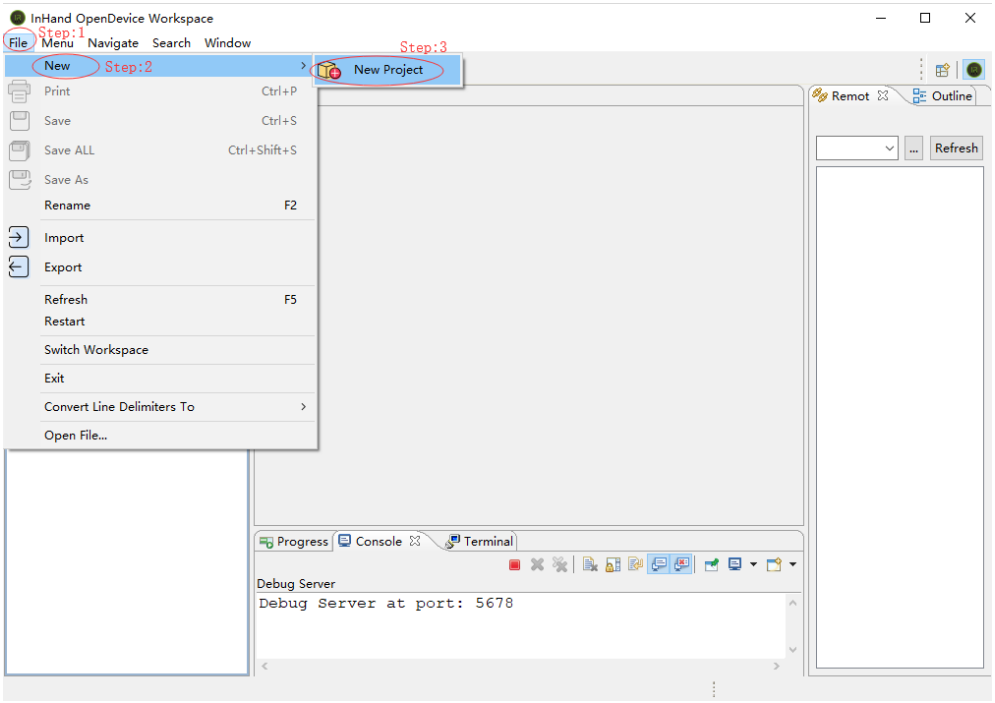


2. Develop App in IDE

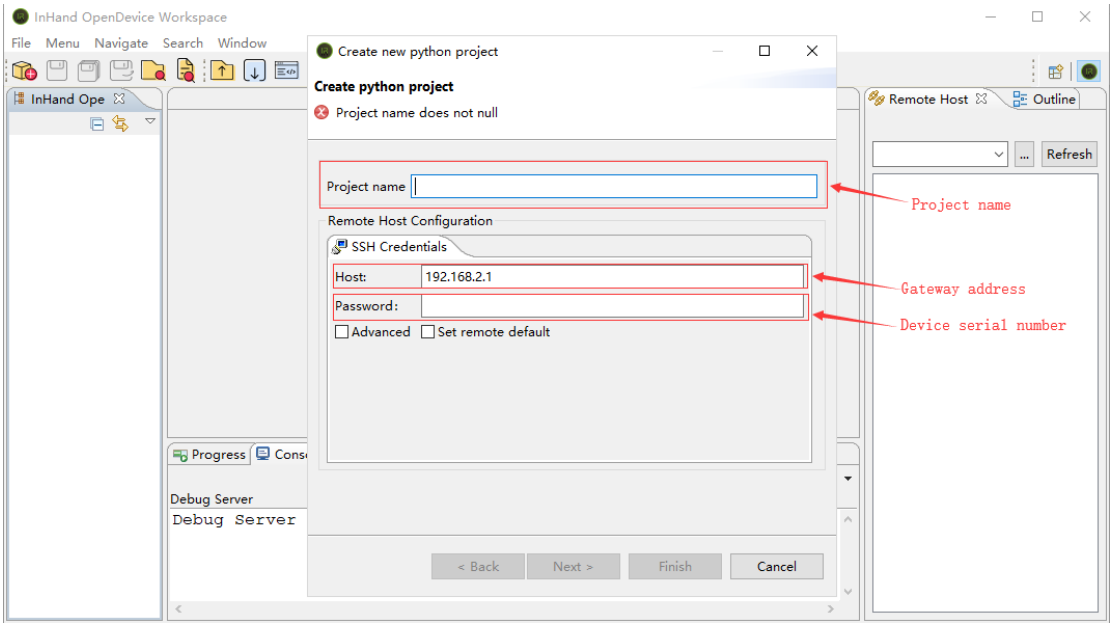
Step 1: Launch IDE and select a working directory. When you develop an app in IDE, each Python app project must have an independent working directory.

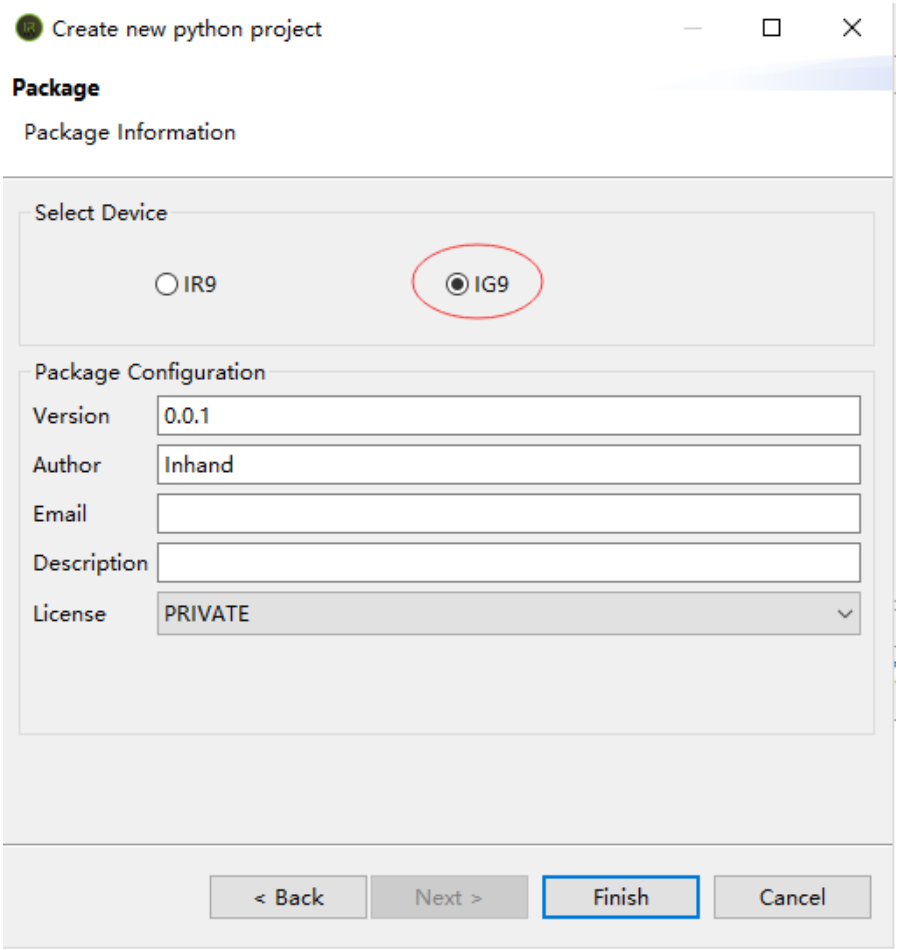


Step 2: Create a project. Choose **File** > **New** > **New Project**.

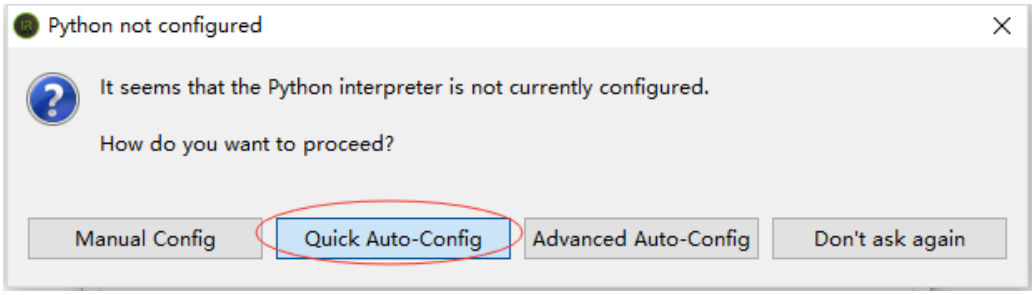


Project name indicates the name of the project, namely, the name of the compiled app. Set **Host** to the gateway IP address, set **Password** to the device sequence number (SN), and retain the default settings for other options.





When the new project is created, IDE prompts you to set the Python environment. The **Quick Auto-Config** function is selected by default.

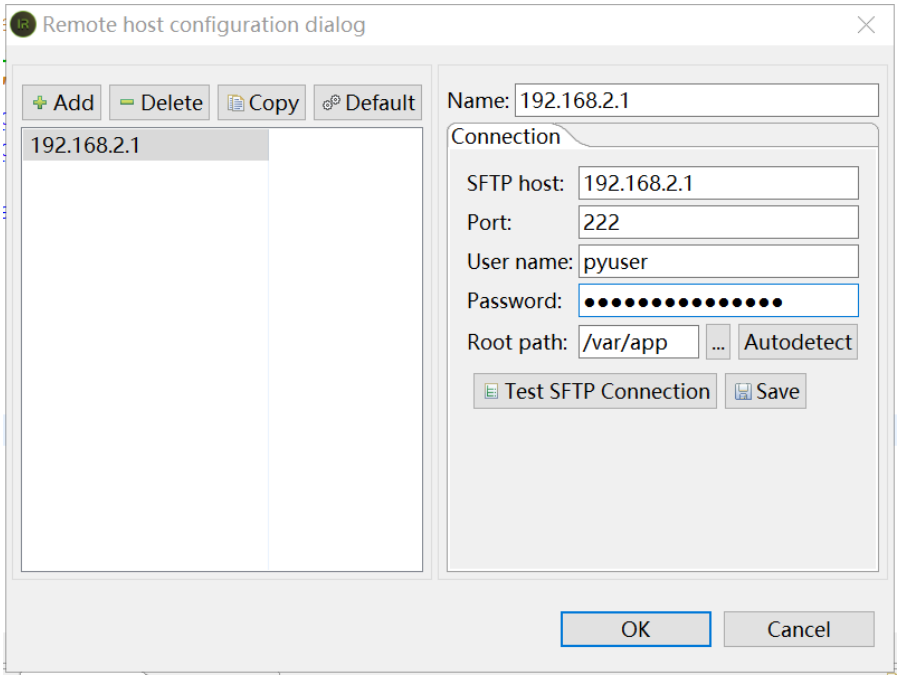


⚠ Note: If the IDE function is not enabled on the gateway web page or the SN is incorrectly set in project creation, IDE cannot be connected to the device. In this situation, you do not need to recreate the project. Instead, you only need to click ... in **Remote Host** on the right side of IDE to reconnect to the device. The settings are as follows:

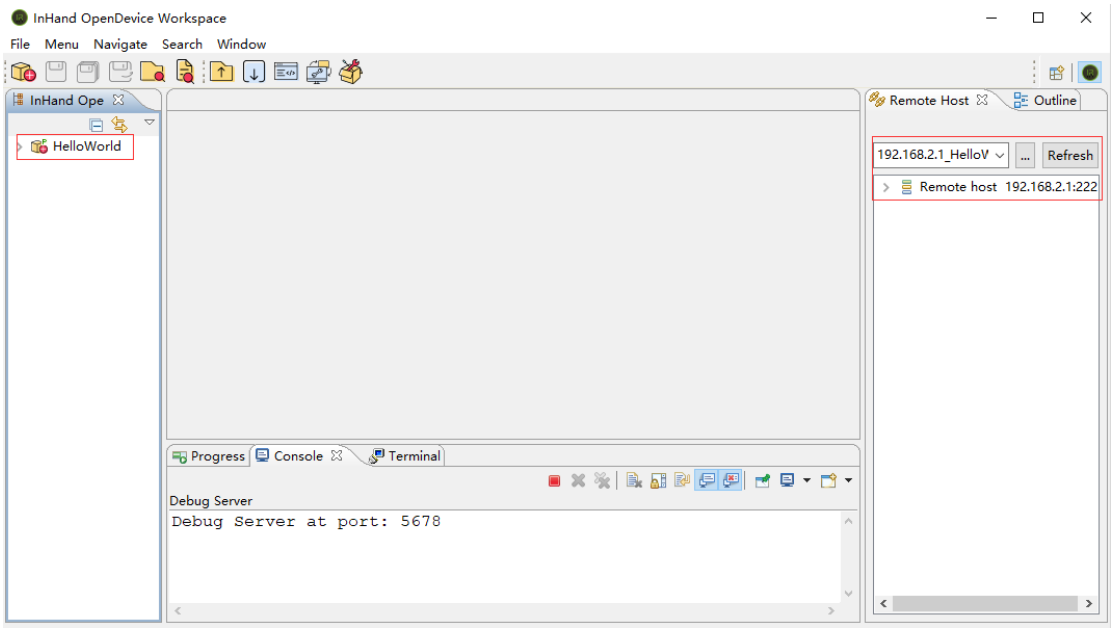
- Name: name of the connection (customized, cannot be empty)
- SFTP host: gateway IP address

- User name: pyuser (fixed user name)
- Password: xxx (device SN)

Click **Test SFTP Connection**. The connection status is displayed.



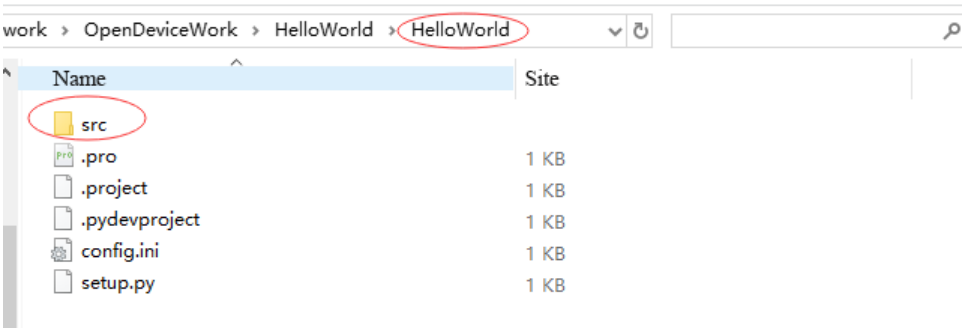
After the connection is set up, the successfully connected device is displayed in **Remote Host**.



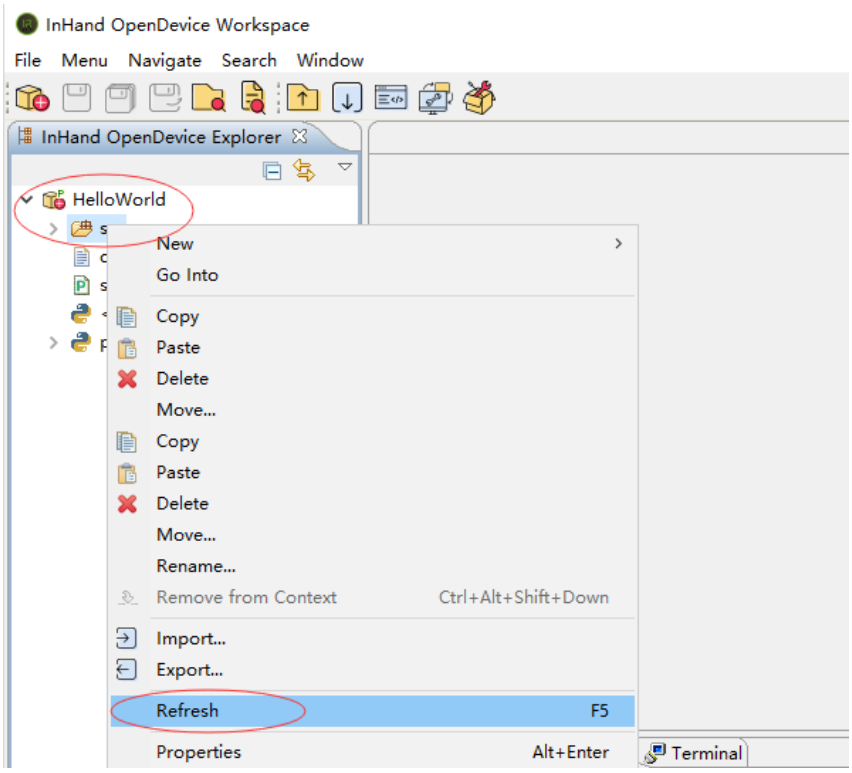
Step 3: Edit code. After the environment settings are completed, you can develop the Python app. This step describes the compiling process without mentioning the code editing. In this

example, an edited Python app is imported to the IDE project directory for packaging.

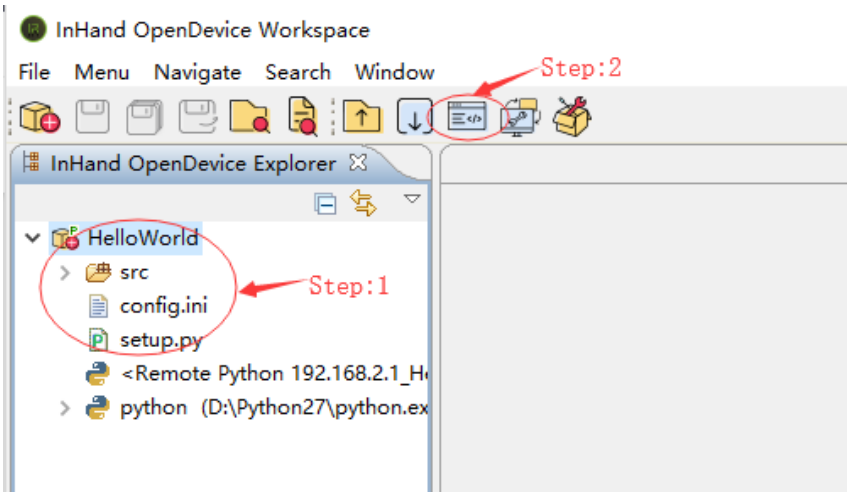
(1) Copy the files from the project directory to the IDE project directory **src**.



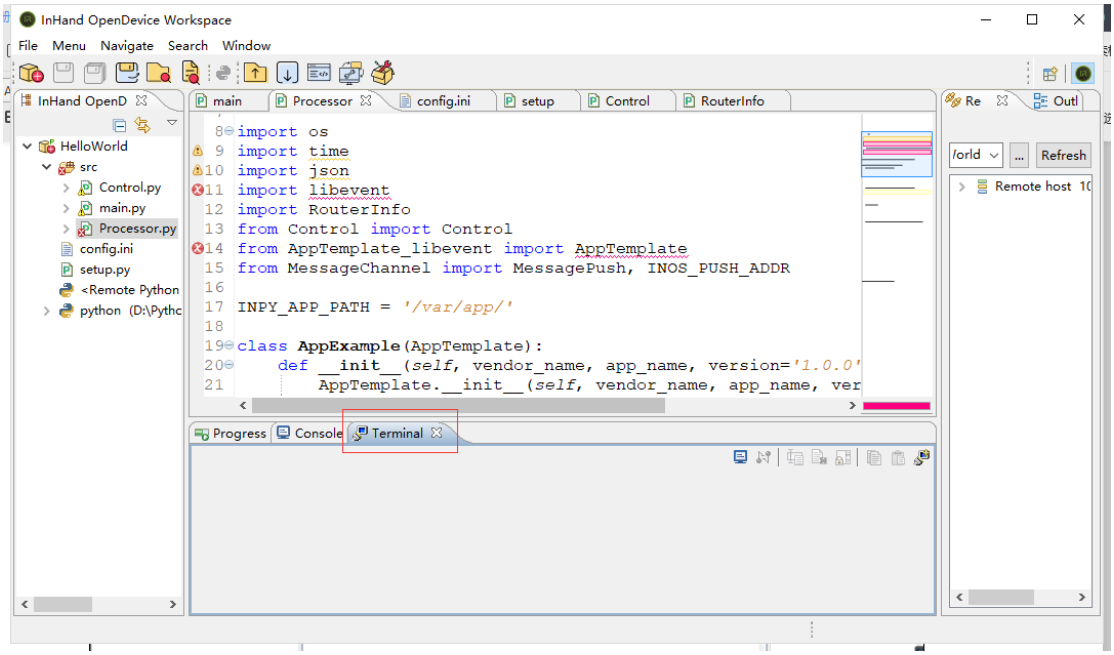
(2) Refresh the buffer.

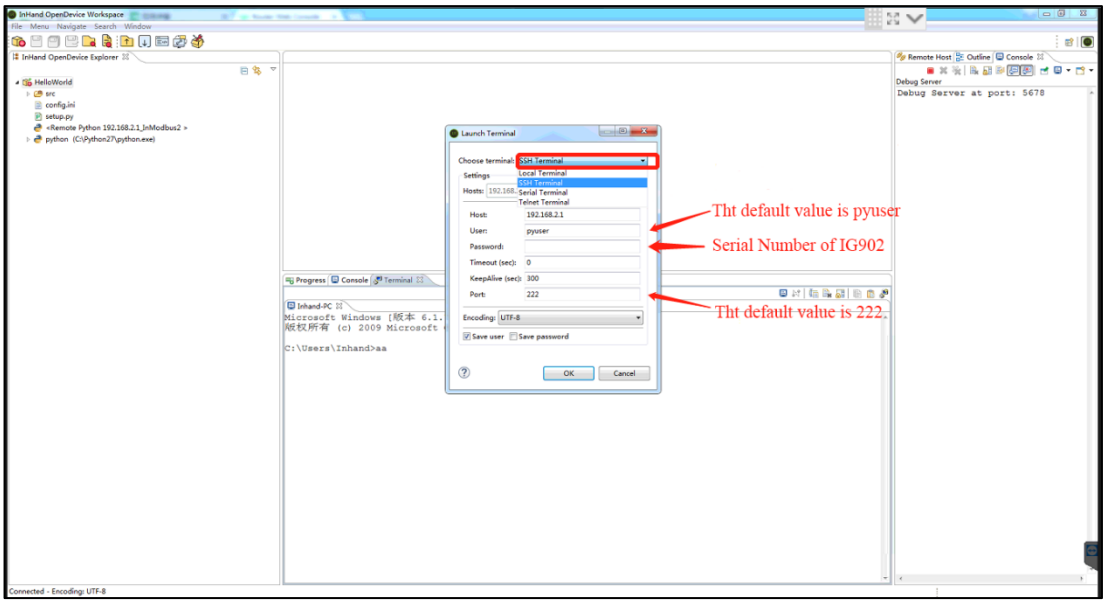


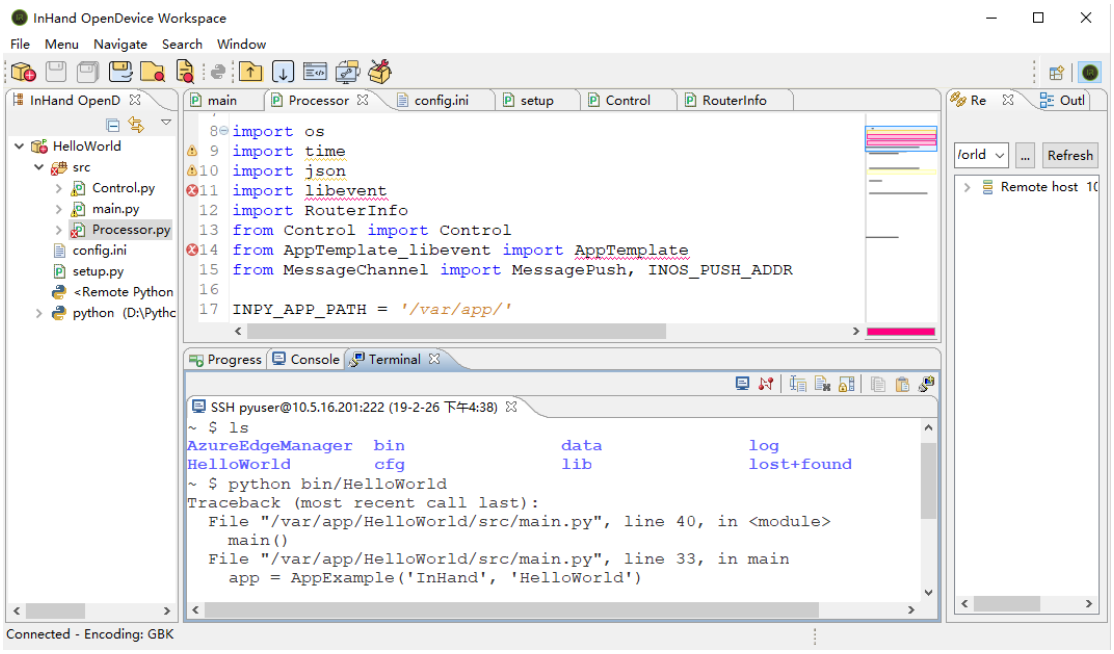
(3) Compile the project file. Select the project name, for example, **HelloWorld**, and click **Compile Pyc**.



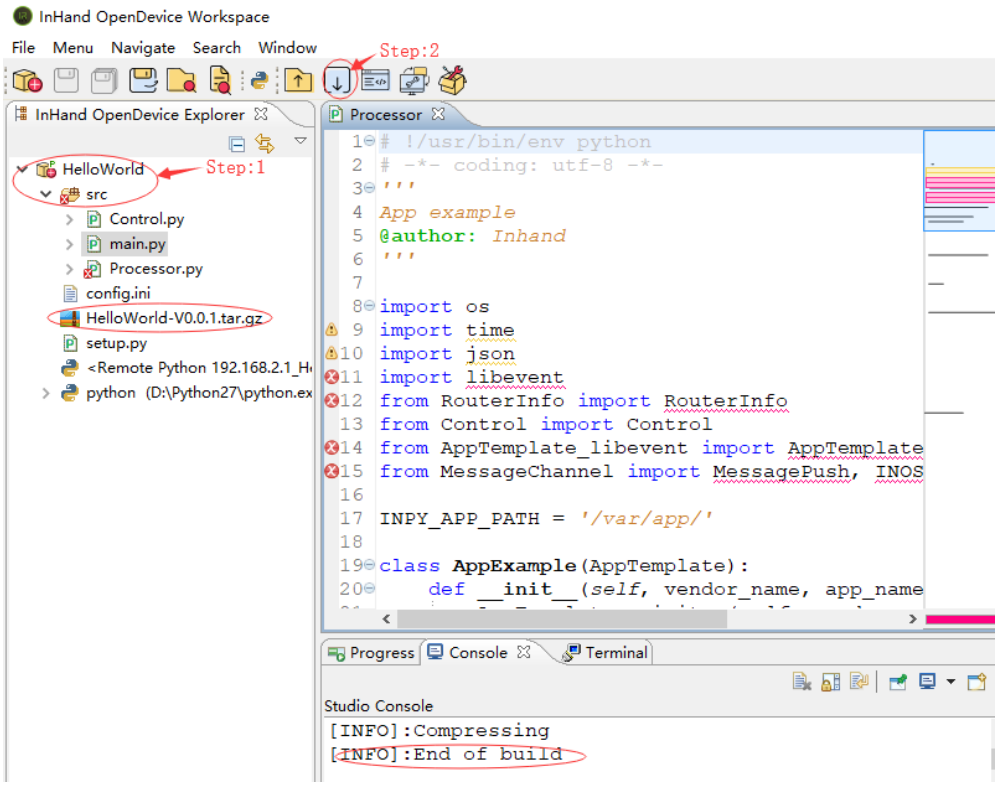
Step 4: Debug the program. After the program is completed, click the **Terminal** tab on the console, and click **Open a terminal** to open the **Launch terminal** interface. Set the parameters and run Processor to debug the program. The procedure is as follows:







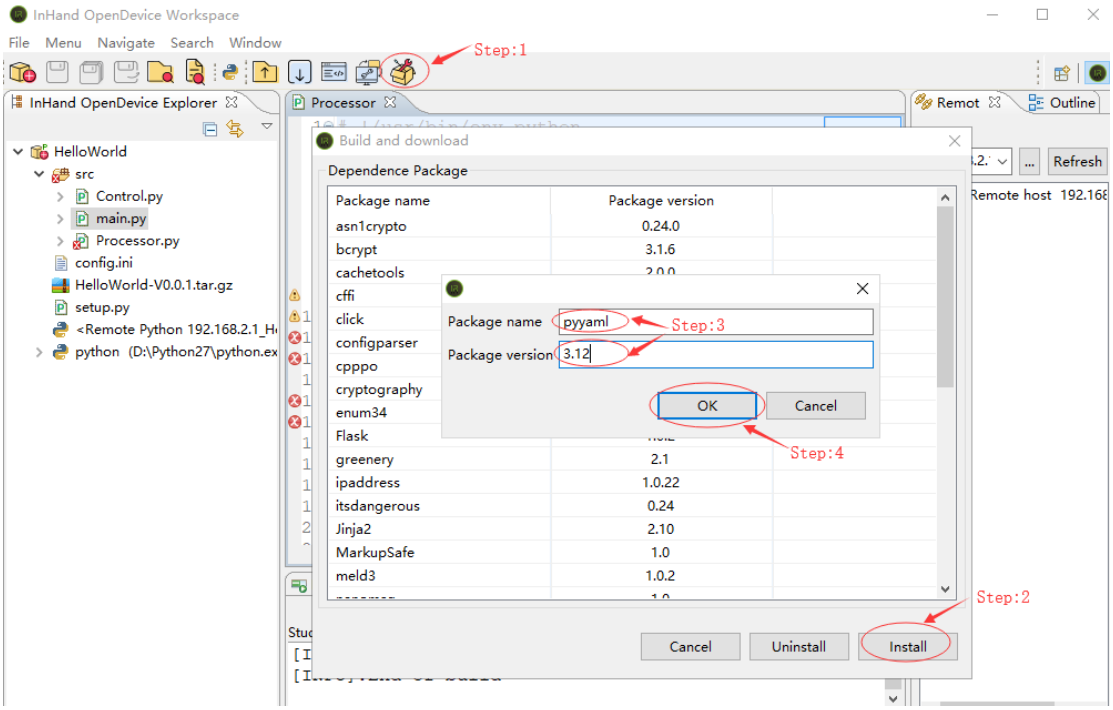
Step 5: Generate the program package. Click the project name Helloworld and click . An executable package file of the gateway is generated. That is, <project name +version number>.tar.gz is the generated app file package.



3. Install the Third-Party Dependence Package

The third-party dependence packages are often required during development. IDE provides the corresponding package management function for you to add and delete dependence packages.

For example, the procedure for installing the PyYAML package is as follows:



⚠ Note:

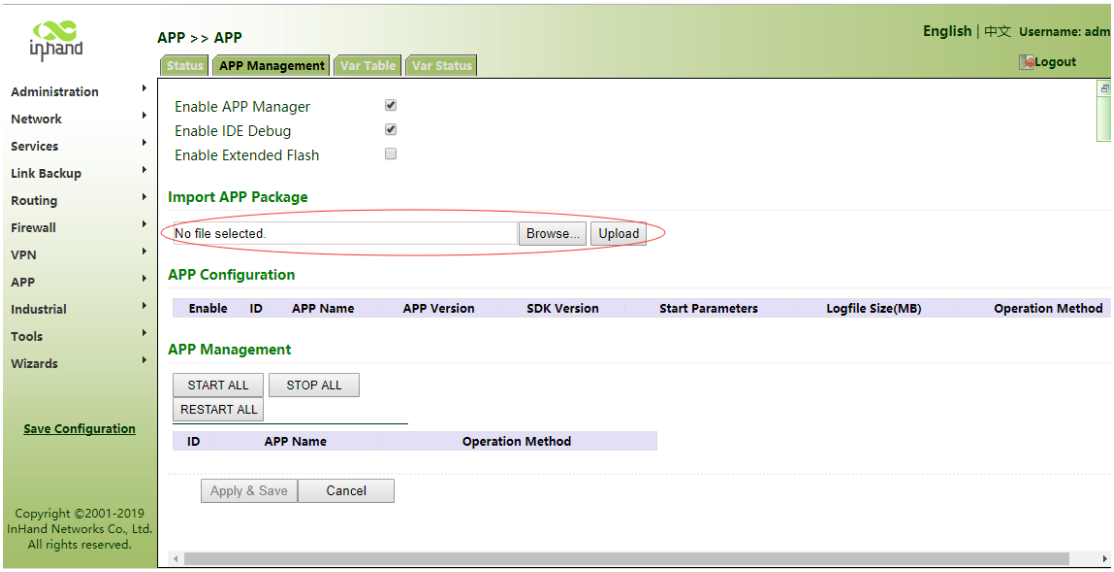
- If other language libraries (such as C/C++) are needed for installing the dependence package, contact the technical support personnel or wait until the compiling tool chain is open.

- If no pop-up window is displayed after you click **Package Manager** of IDE, do not click it repeatedly. Wait until the installation window is displayed.

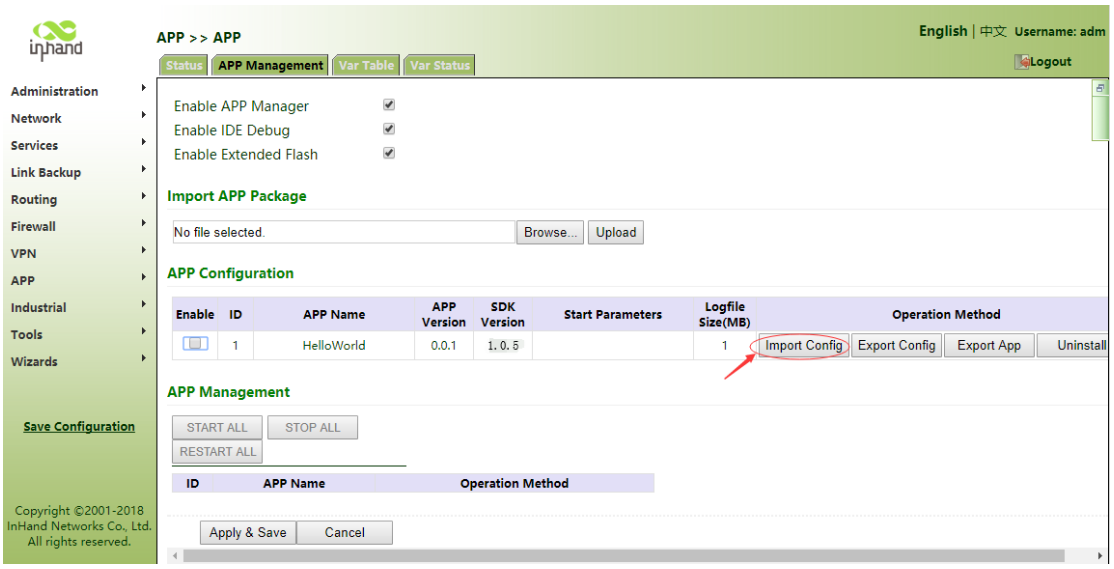
Chapter 3 Python APP Installation

1. Install Python App

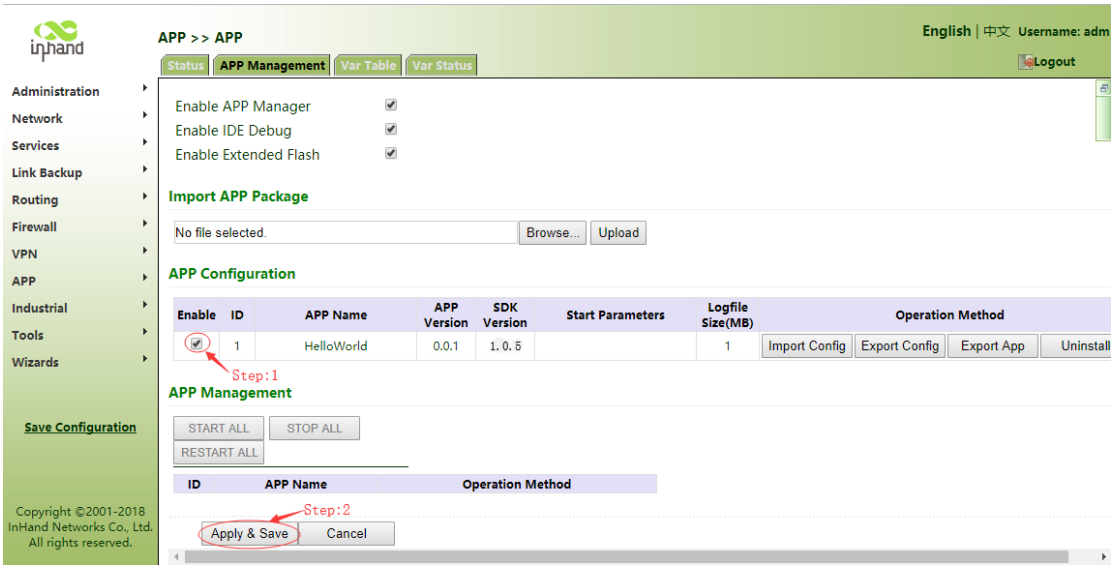
On the web page of the gateway, choose **APP** > **APP** > **APP Management**, click **Browse** to select the app package, and click **Upload** to upload the app to the device.



Import the customized app configuration file. If the configuration file does not need to be modified, the gateway uses the default configuration file in the app package by default.

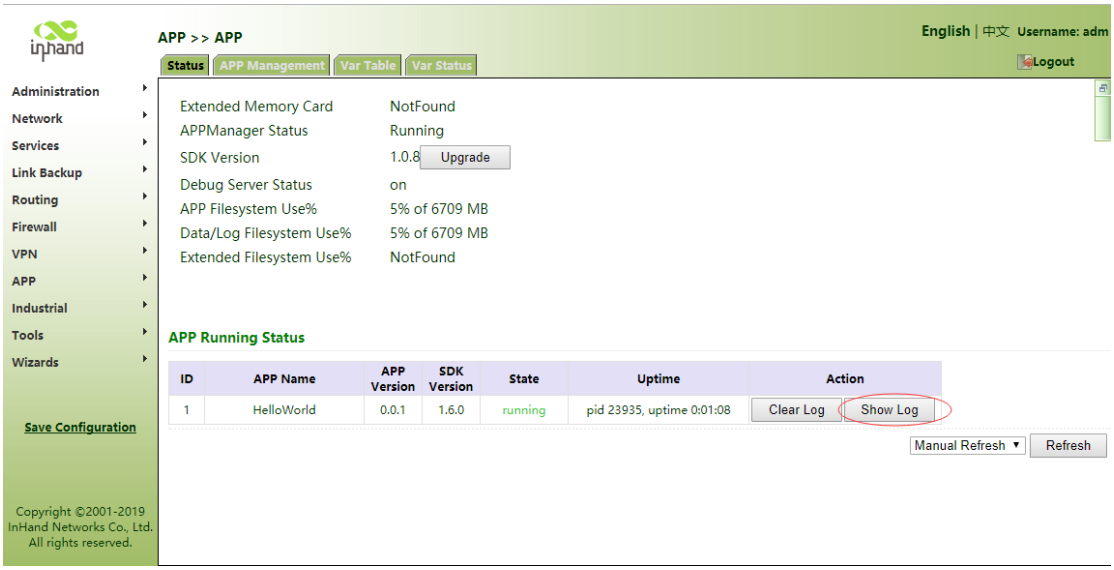


Run the app.



2. View the Python App Status

On the web page of gateway, choose **APP** >> **APP** >> **Status** to view details about the Python app. To view the running log of a specific app, click **Show Log**.



FAQs

1. IDE Abnormality

IDE reports the error: [EXCEPTION]:java.net.ConnectException: Connection refused: connect ... Why?

There are two possible reasons: (1) IDE debugging is not enabled on the gateway web page. (2) The session expires. Click **Refresh** on the **Remote Host** interface of IDE to set up the session again.

Contact Us

Add: 3900 Jermantown Rd., Suite 150, Fairfax, VA 22030 USA
E-mail: support@inhandneworks.com
T: +1 (703) 348-2988
URL: www.inhandnetworks.com

InHand Website