PHY62XX

BurnMode and DebugMode User Guide

Version 1.0

Author: PHY+SW

Security: Public

Date: 2021.3

PhyPlus

Revision History

Revision	Author	Participant	Date	Description
v1.0	PHY+SW		03/13/2020	First Edition

Table of Contents

1 Introduction 1
1.1 Introduction to mode 1
1.2 Burning mode 2
1.3 Run debug mode 2

Figures and Tables

Figure 1: PHY6200_32_V1.4 Development Board 1
Figure 2: Burning Mode 2
Figure 3: Select JLINK 3
Figure 4: Identifying the JTAG Tool Chain 3
Figure 5: Don't Download Code to FLASH 4
Figure 6: Debug Code 4
Figure 7: Configuring the Sleep Options 5

1 Introduction

PHY62XX has two modes: Flash burning mode and Flash running debugging mode, hereinafter referred to as burning mode and running debugging mode.

Take the development board PHY6200_32_V1.4_20180314 as an example:

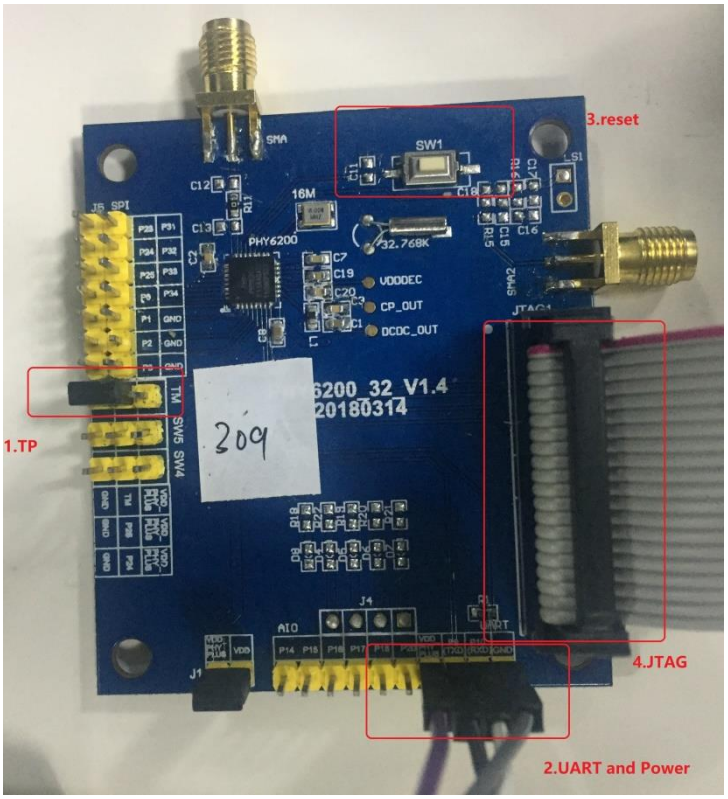

Figure 1: PHY6200_32_V1.4 Development Board

Interactive interfaces commonly used during development and debugging:

1. TP: TP_Pin8, used to set the working mode of the system, burning mode or running debug mode.
2. UART and Power: Serial port debugging and system power supply, UART can be used to burn codes and print logs to assist debugging.
3. Reset: system reset key.
4. JTAG: Source code can be debugged step by step in debug mode.

1.1 Introduction to mode

TP is used to configure the mode of the development board:

- TP_Pin8 is high level: burning mode, you can use PhyPlusKit.exe to erase and burn Flash through the serial port.
- TP_Pin8 is low level: run debugging mode, you can run the code at full speed, and you can debug the code step by step.

1.2 Burning mode

The Flash burning process is as follows:

- Pull TP_Pin8 high to select the burning mode.
- Select the corresponding serial port on PhyPlusKit and set the correct baud rate.
- Erase Flash, if erasing fails, you can reset the system.
- Load the burned hex or img file and click Write.

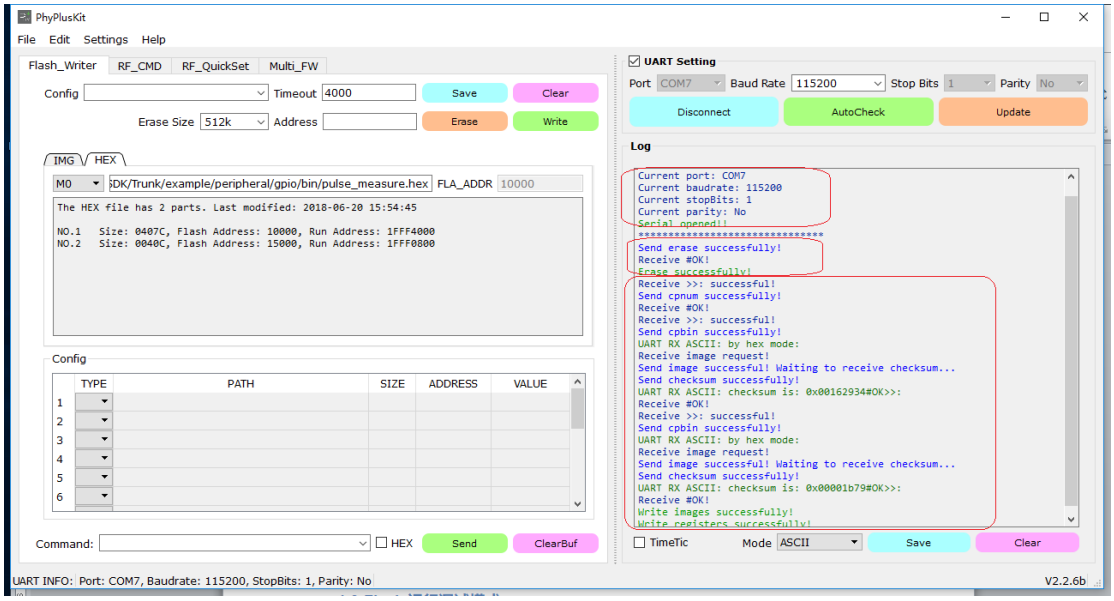

Figure 2: Burning Mode

1.3 Run debug mode

Pull TP_Pin8 low, reset or power on again, and set the system to enter the running debugging mode.

After reset or re-power on, if there is no operation on JTAG, the system will run at full speed.

After reset or re-power on, the PC can take over the system through JTAG, and the system can be debugged in a single step.

When debugging, there are some requirements for the source code:

- If the code being debugged will not enter sleep, you can debug it directly.
- If the debugged code will go to sleep, JTAG will be disconnected after going to sleep, so it cannot be debugged. One of the recommended methods is to erase the contents of Flash to be empty in burning mode. At this time, the system can be debugged before going to sleep.
- CFG_SLEEP_MODE can be set to PWR_MODE_SLEEP or PWR_MODE_NO_SLEEP in the project configuration item to allow the system to enter sleep and not allow the system to enter sleep.

Take Jlink as an example. You can refer to the following steps when debugging:

- Connect Jlink to the development board.

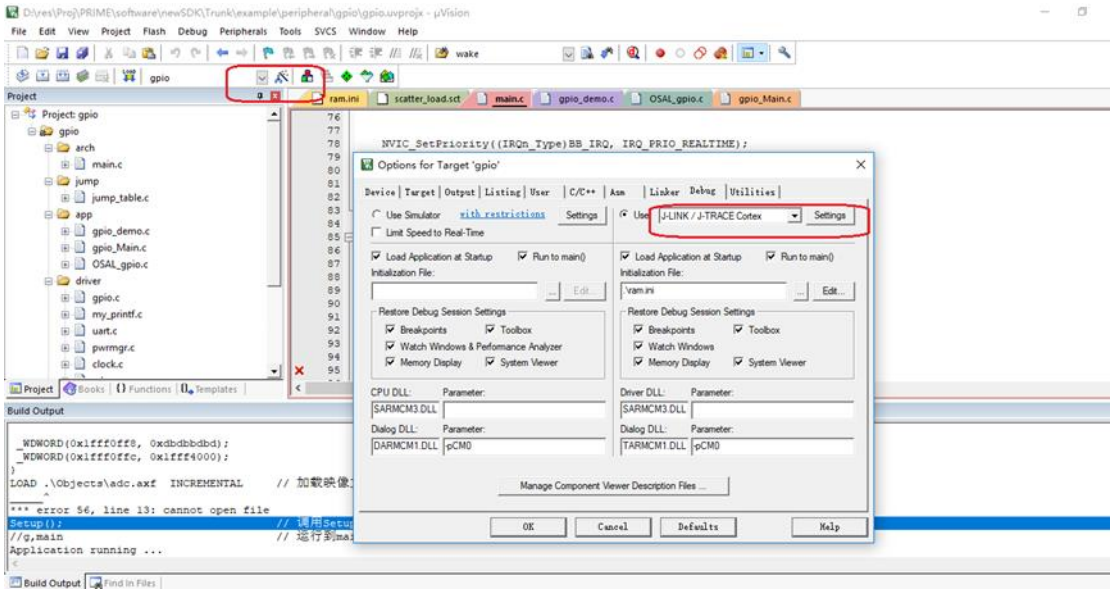

Figure 3: Select JLINK

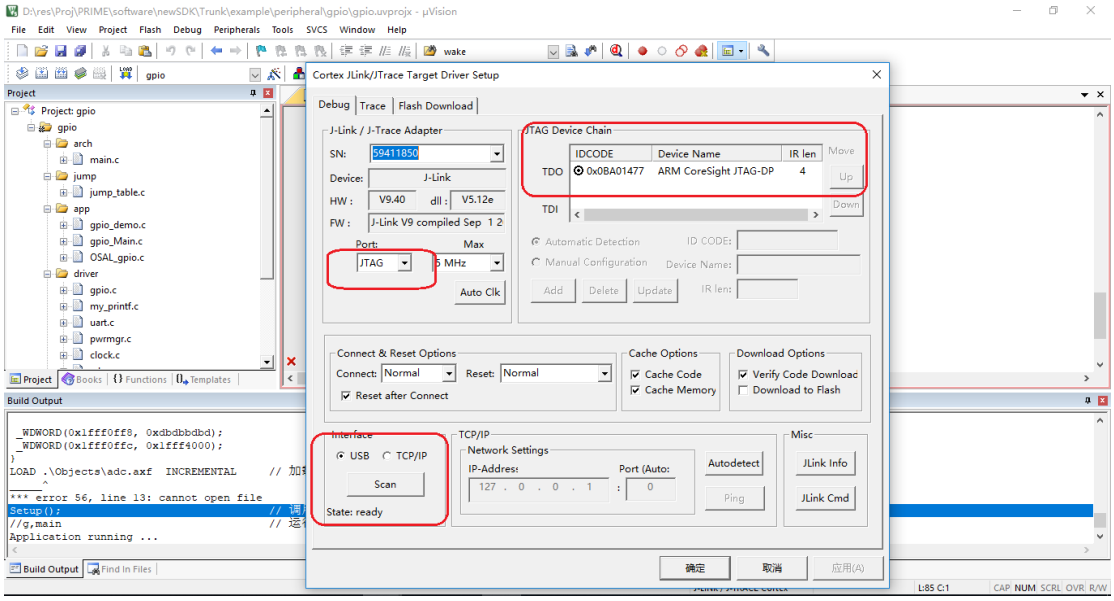

Figure 4: Identifying the JTAG Tool Chain

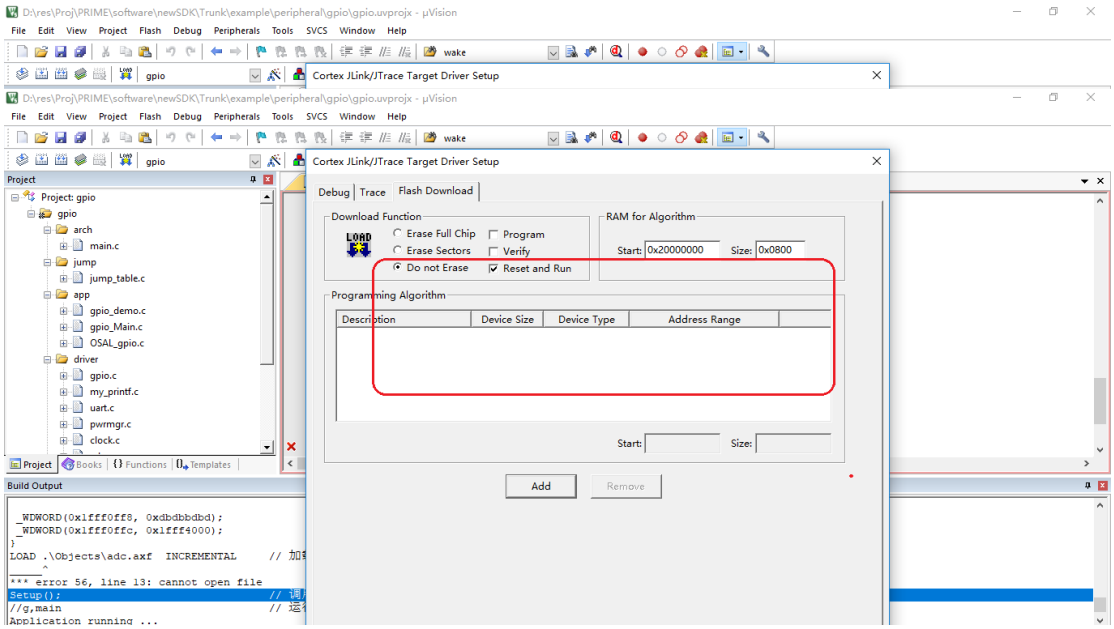

Figure 5: Don't Download Code to FLASH

- Start debugging the code.

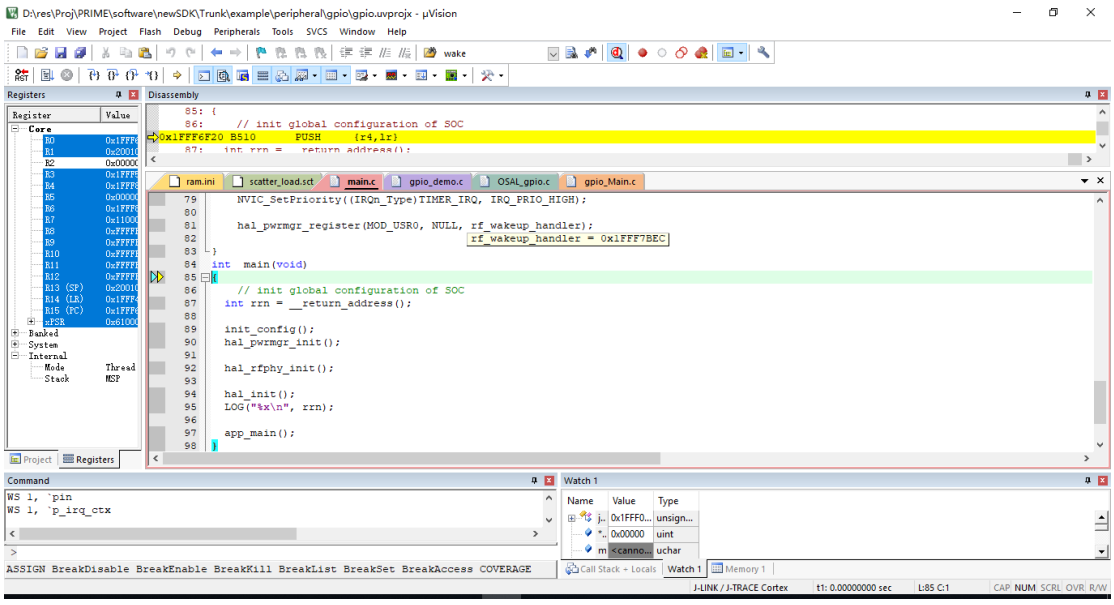

Figure 6: Debug Code

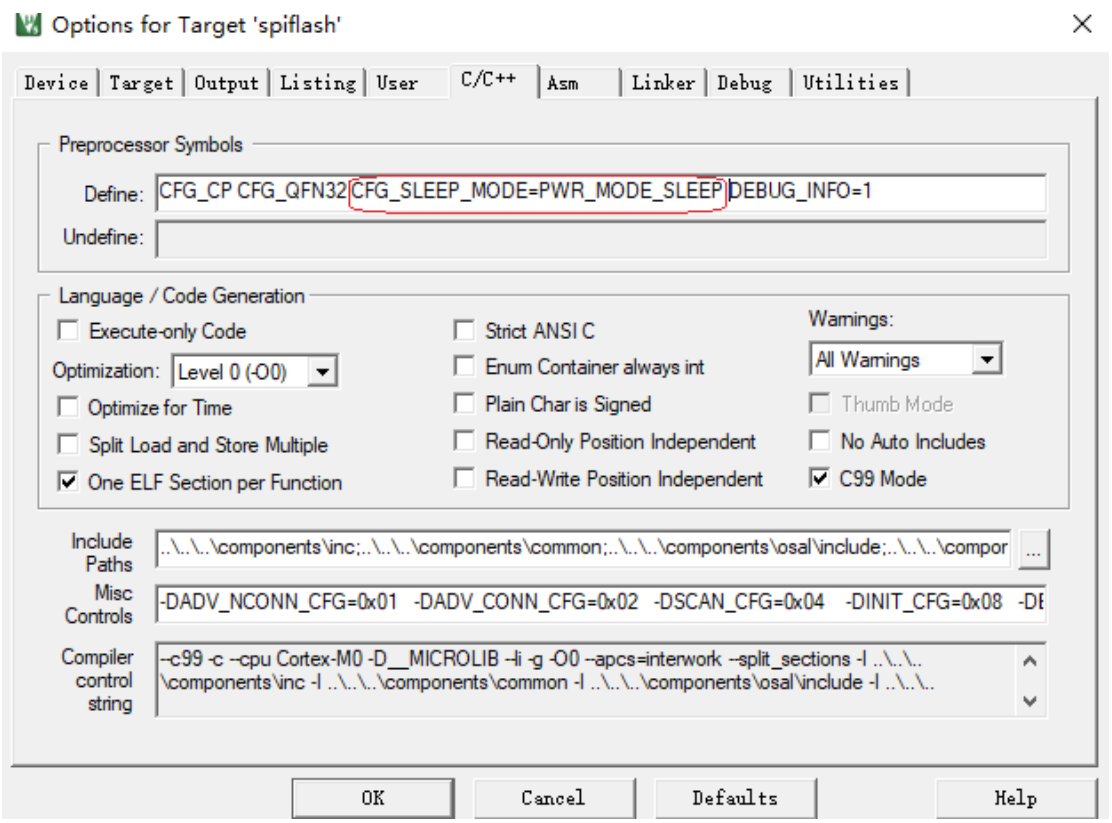

Figure 7: Configuring the Sleep Options