

## STM32CubeIDE ST-LINK 服务器

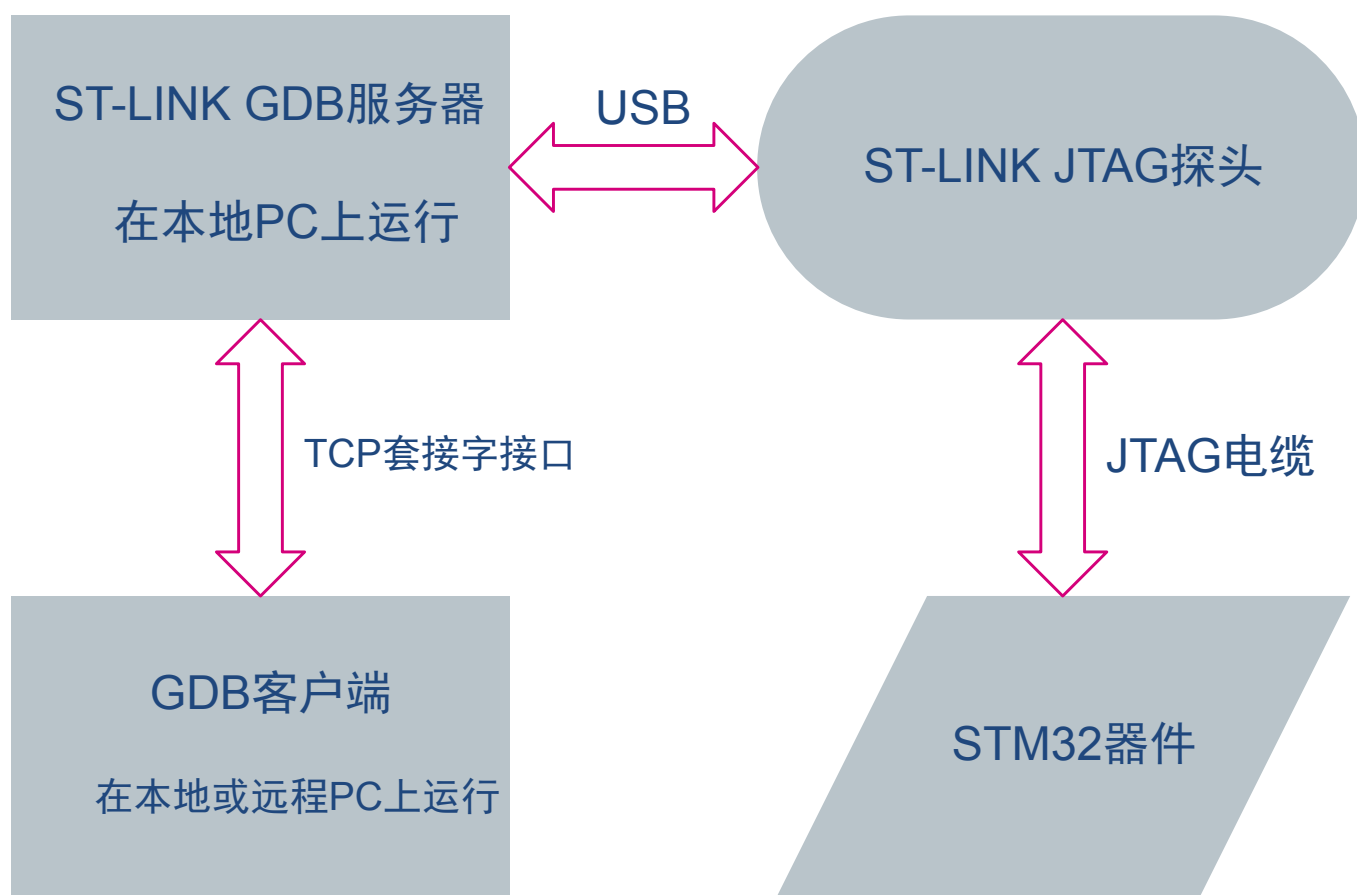
### 引言

STM32CubeIDE ST-LINK GDB server 也被称为 GDB server，是通过 ST-LINK JTAG 探头在与 Arm® Cortex®-M 目标设备连接的 PC 上运行的命令行应用程序。

ST-LINK GDB server 启动时会通过 ST-LINK JTAG 连接至 STM32 Arm® Cortex®-M 目标设备。与目标设备侧建立通信之后，将等待客户端连接至 TCP 监听套接字。客户端连接至 TCP 监听套接字之后，ST-LINK GDB server 将处理客户端发送的远程串行通信协议（RSP）消息，并对目标设备侧执行适当操作，随后向客户端回复 RSP。

图 1 展示了使用 ST-LINK GDB server 和 STMicroelectronic ST-LINK 探头调试 Arm® Cortex®-M 目标设备的标准调试会话。

图 1. 调试设置概览



该图展示了如何利用 TCP 套接字接口将 GDB 客户端连接至 ST-LINK GDB server, 以便对 ST-LINK JTAG 上所连接的 Arm® Cortex®-M 目标进行调试。

## 1 GDB server 使用

STM32CubeIDE-Link GDB 服务器是命令行应用程序，该应用程序可通过以下方式启动

- 输入一组命令行选项
- 指示 GDB server 从配置文件中加载选项

若未指定任何选项，GDB server 将以预配置默认选项启动。启动选项及对应的默认值列于第 1.1 节 GDB server 启动选项中。

STM32CubeIDE-Link GDB 服务器利用 STM32CubeProgrammer (STM32CubeProg) 对需要调试的设备进行 flash 下载。当 gdb 发出 load 命令时，GDB server 将自动使用 STM32CubeProgrammer 软件。

提示 当需要进行高级设备控制（例如 Flash 擦除或者设置选项字节）时，可以使用 STM32CubeProgrammer (STM32CubeProg)。它包含在 STM32CubeIDE 中并且可以从 STMicroelectronics 网站 [www.st.com](http://www.st.com) 单独下载。

STM32CubeIDE 支持基于 Arm® Cortex® 处理器的 STM32 32 位产品。

提示 Arm 是 Arm Limited（或其子公司）在美国和/或其他地区的注册商标。

arm

### 1.1 GDB server 启动选项

GDB 服务器可以利用以下命令启动：

ST-LINK\_gdbserver.exe [options]

大多数选项有长格式和短格式两种格式。以下将介绍 GDB server 支持的选项组，其中默认模式和配置以粗体表示：

- -h, --help  
使 ST-LINK\_gdbserver 显示使用信息。
- -c <config file>, --config-file <config file>  
指示 GDB server 从配置文件中读取选项。附录 B 中提供了有关语法分析的配置文件示例。
- -f <log file>, --log-file <log file>  
指定日志文件名称。默认名称为 debug\_log.txt。
- -l <log level>, --log-level <log level>  
指定记录级别。记录级别分为 0 至 31 级。将记录级别指定为 0 时将关闭记录功能，将记录级别指定为 31 时将启用完全记录。默认记录级别为 31。记录级别可以根据以下列表进行设置：
  - = 0: 禁用记录。
  - ≥ 1: 启用错误消息记录。
  - ≥ 2: 增加警告消息。
  - ≥ 4: 增加通信特定消息。
  - ≥ 8: 增加所有信息消息
  - ≥ 16: 增加所有 HW 特定消息。

因此若未指定选项 -f 和 -l，服务器将以默认的 31 级记录级别（完全记录）启动，默认日志文件为 debug\_log.txt。

- -p <port number>, --port-number <port number>  
指定服务器监听客户端连接对应的 TCP 端口号。服务器监听的默认端口号为 61234。
- -v, --verbose  
在命令行中指定 -v 将启用详细模式；此外，还将打印服务器调试日志标准输出。默认情况下，详细模式处于禁用状态。
- -r <refresh delay>, --refresh-delay <refresh delay>  
指定调试日志文件中硬件状态更新的间隔时间（单位秒）。注意，状态更新间隔时间很短会导致调试日志文件扩大。默认状态刷新延迟时间为 5 秒。
- -s, --verify  
启用 Flash 下载验证。

- `-e, --persistent`  
指定选项 `-e` 后将在持续模式下启动服务器。默认情况下，持续模式将处于禁用状态，这意味着服务器将在非持续模式下启动。
- `-d, --swd`  
启用串行线调试（SWD）模式指定使用 SWD 而非使用 JTAG。  
[对 SWV 使用此选项。]
- `-z <port number>, --swo-port <port number>`  
指定服务器输出原始 SWO 数据对应的 TCP 端口号  
[对 SWV 使用此选项。]
- `-a <cpu clock>, --cpu-clock <cpu clock>`  
指定 CPU 的时钟频率，单位 Hz。  
[对 SWV 使用此选项。]
- `-b <SWO CLOCKDIV>, --swo-clock-div <SWO CLOCKDIV>`  
指定 SWO 时钟分频器。利用 `-a` 选项可以使其作用于给定的 CPU 时钟速度。  
[对 SWV 使用此选项。]  
  
同时使用 `-z`、`-a` 和 `-b` 选项可指定 SWO 速度。例如当核心以 168 Mhz 频率运行，目标 SWO 频率为 1 Mhz 时，将按照以下方式配置上述选项： `-z 61235 -a 168000000 -b 168`。当核心以 72 Mhz 频率运行，SWO 频率为 125 kHz 时，将使用 `-z 61235 -a 72000000 -b 576`。
- `-k, --initialize-reset`  
在复位状态下将设备初始化。
- `-q, --debuggers`  
列出已连接的调试器，ST-LINK 序列号。
- `-i <ST-LINK S/N>, --serial-number <ST-LINK S/N>`  
指定服务器应连接的 ST-LINK 序列号。
- `-m <apID>, --apid <apID>`  
定义需要调试的 apID。  
[对多内核使用此选项。]
- `-g, --attach`  
连接正在运行的目标。
- `-t, --shared`  
允许使用一个 ST-LINK 探头将两个或以上程序连接至同一台设备。  
[对 ST-LINK server 使用此选项。]
- `--erase-all`  
擦除所有存储内容。
- `--memory-map <device id>`  
显示给定设备标识符对应的存储器映射例如 0x410。
- `--ext-memory-loaders`  
提供可供使用的外部存储器加载程序列表。
- `-el <file_path>, --extload <file_path>`  
选择自定义外部存储器加载程序。
- `-cp <path>, --stm32cubeprogrammer-path <path>`  
STM32CubeProgrammer (STM32CubeProg) 安装路径。
- `--temp-path <path>`  
启动调试会话的临时文件存储在此前提供的路径下。
- `--licenses`  
提供已使用的工具和许可列表。
- `--, --ignore-rest`  
忽略剩余的、后跟此标志的标签参数。
- `--版本`  
显示版本信息并退出。

## 1.2 GDB server 工作模式

此服务器可以在持续模式或非持续模式下运行 - 具体由配置选项决定。在持续模式下，服务器将在客户端断开连接后继续运行并等待新连接，而在非持续模式下客户端关闭连接后服务器将直接退出。但如果服务器与目标进行通信时出现错误，无论服务器处于哪一种模式，服务器都将关闭所有连接并关机。

## 1.3 启动 GDB server

GDB server 可以在命令窗口中以下列方式启动：

1. `cd`  
`C:\ST\STM32CubeIDE_1.0.0.19w12patch\STM32CubeIDE\plugins\com.st.stm32cube.ide.mcu.externaltools.stlink-gdb-server.win32_1.0.0.201903011553\tools\bin\ST-LINK_gdbserver`
2. 利用 `-cp` 和 `STM32_Programmer_CLI.exe` 路径启动 GDB server  
例如：`ST-LINK_gdbserver.exe -d -v -cp`  
`"C:\ST\STM32CubeIDE_1.0.0.19w12patch\STM32CubeIDE\plugins\com.st.stm32cube.ide.mcu.externaltools.cubeprogrammer.win32_1.0.0.201903011553\tools\bin"`
3. 随后 GDB server 将通过 ST-LINK JTAG 与 STM32 设备相连，并等待 GDB 调试会话命令。

## 2 利用 GDB 调试

本章节将介绍通过 GDB 客户端调试程序所需的最小 GDB 命令集。在 GDB 命令行提示位置键入 `help` 可以获取完整命令列表。或者也可以查看 [STM32CubeIDE](#) 信息中心提供的 GDB 用户手册。

测试 ST-LINK GDB server 采用的是 v7.0 和 8.1 版 GDB 客户端。

### 2.1 启动 GDB

利用以下命令启动命令行 GDB 客户端

```
>arm-none-eabi-gdb <program.elf>
```

其中<program.elf>是需要调试的程序。

### 2.2 连接至服务器

此服务器可以在持续模式或非持续模式下启动：

- 若服务器在非持续模式下启动，必须使用以下命令将客户端连接至服务器  

```
>target remote <ip-address>:<port-number>
```

 在非持续模式下使用此命令将客户端连接至服务器时，必须关闭客户端和服务端，否则将无法重新启动调试会话。
- 若服务器在持续模式下启动，必须使用命令将客户端连接至服务器  

```
>target extended-remote <ip-address>:<port-number>
```

 在持续模式下利用此命令将客户端连接至服务器时，用户只需操作设备重新启动调试会话即可，无需关闭服务端或客户端。在此操作模式下，客户端和服务端可反复启动调试会话。

### 2.3 将程序加载至目标

利用 GDB 命令将可执行程序加载至目标

```
>load <program.elf>
```

此命令可将程序加载至目标。若程序的代码段位于 Flash ROM，在写入新代码之前请擦除 Flash 中对应页面。

**提示** *Flash 编程（代码下载至闪存内）通过 GDB 服务器实现并且对用户透明。*

### 2.4 设置断点

利用 GDBbreak 命令放置断点。

```
> break [LOCATION]
```

其中，LOCATION 为行号、功能名称或“\*”加地址。当指定行时，GDB 将在该行对应的代码起始处中断。当指定功能时，GDB 将在该功能对应的代码起始处中断。当指定地址时，GDB 将在确切的地址处中断。

利用 break 命令可将软件断点置于指定位置。若代码段位于 RAM 内，此命令有效。但若代码段位于 Flash ROM 内，则使用 hbreak 命令而非 break 命令。Hbreak 命令参数与 break 命令参数一致。

**提示** *最多可同时放置的硬件断点数取决于调试后的 STM32 设备。注意，软件断点并无此类限制。*

### 2.5 设置监控点

通过命令可以跟踪变量的读取、写入和访问操作

```
>rwatch <variable-name>, 读取变量时将停止程序,
```

```
>watch <variable-name>, 写入变量时将停止程序,
```

```
>awatch <variable-name>, 访问变量时将停止程序,
```

**提示** *最多可同时放置的监控点数由硬件配置决定。*

## 2.6 运行程序

加载程序之后，调试器将开始执行命令

```
>continue
```

提示

目前没有能够将命令行参数指定给参数的机制。

程序运行时可按下 Ctrl-C 使程序停止。用户可利用 next 命令跳过声明。此外，用户也可以利用 step 命令启动某功能。

## 2.7 退出调试会话

用户通过发出 GDB 命令 quit 可退出调试会话。

## 2.8 在目标 STM32 主板上调试

ST-LINK GDB server 支持 STM32 Arm® Cortex®-M 内核设备调试。支持 RAM 和 Flash 调试。Arm® Cortex®-M 硬件断点功能支持 Flash 断点。应用代码经过编译后可在 RAM 或 Flash 中运行。由于 GDB 在内部区分 RAM 和 Flash 模式，因此无需特殊命令。

要在 Arm® Cortex®-M 微控制器上调试 RAM 或 Flash 内的代码，必须将栈指针（SP、R13）和程序计数器（PC、R15）正确初始化：

- 若待调试的代码位于 Flash 内，将从 0x0 开始读取 SP，从 0x4 开始读取 PC。
- 若待调试代码位于 RAM 内，矢量起点位于 RAM 下载区域的起始位置。对于大多数设备，该值为 0x20000000。这意味着栈指针（SP）的初始值从 0x20000000 开始读取，PC 初始值从 0x20000004 开始读取。

下载代码后使用命令

```
>monitor reset
```

正确设置 SP 和 PC。STM32CubeIDE 将代码下载至目标设备后将自动发出此命令。

## 2.9 监控命令

支持 GDB 监控命令，该命令可实现特定目标功能，特别是执行 Flash 控制器专用命令。

以下命令是 ST-LINK GDB server 支持的监控命令：

- >monitor help  
打印有关可支持监控命令的信息。
- >monitor reset  
重置目标设备。
- >monitor flash mass\_erase  
用于擦除 Flash 存储器中的所有页面。注意，使用此命令会导致存储在 Flash 内的数据全部丢失。
- >monitor ReadAP <register index>  
读取 CoreSight AP 寄存器。

暂不支持以下命令，如使用以下命令将返回正常状态：

- >monitor ReadDP <register index>  
读取 CoreSight DP 寄存器。
- >monitor WriteAP <register index> <data32>  
写入 CoreSight AP 寄存器。
- >monitor WriteDP <register index> <data32>  
写入 CoreSight DP 寄存器。



### 3 故障排除

当服务器挂起或无法启动时，请按照以下步骤进行操作：

1. 关闭 GDB server 以及与此服务器相连的所有客户端。
2. 将目标主板断电。
3. 断开 ST-LINK JTAGUSB 电缆连接。
4. 重新连接 ST-LINK JTAGUSB 电缆。
5. 将目标主板通电。
6. 检查调试配置设置，例如 SWD/JTAG 接口
7. 启动 GDB server，这里使用不同的端口号为宜。

最佳做法是尝试调试另一个项目，以消除因项目配置或启动代码引发的特定项目问题。

我们还建议使用另一条 USB 电缆对 ST-LINK 或 STM32 硬件进行调试，以排除因硬件故障引发调试问题。

此外，还建议尝试使用 STMicroelectronics 中的其他工具连接至目标设备，例如 STM32CubeProgrammer (STM32CubeProg)。若 STM32CubeProgrammer 可连接至目标设备，请尝试：

1. 利用 STM32CubeProgrammer 擦除 Flash。
2. 利用 STM32CubeProgrammer 编写新程序。
3. 若编程成功，请再次利用 STM32CubeProgrammer 擦除 Flash。
4. 断开 STM32CubeProgrammer 与目标设备之间的连接。
5. 尝试利用 ST-LINK\_gdbserver 调试程序。

## 附录 A 返回/错误代码信息

表 1. ST-LINK GDB 服务器返回/错误代码

代码	定义	说明
0	TARGET_SUCCESS	正常
1	TARGET_CONNECT_ERR	当前使用的端口，某些器件已接入 STM32CubeProgrammer。
2	TARGET_DLL_ERR	未连接 ST-LINK。
3	TARGET_USB_COMM_ERR	USB 通信错误。
4	TARGET_NO_DEVICE	ST-LINK 已连接，但未连接主板
5	TARGET_UNKNOWN_MCU_TARGET	例如，当只有 SWD 引脚连接至 HW 时，不使用 -d 选项。
6	TARGET_FIRMWARE_OLD	原有 ST-LINK_firmware。
7	TARGET_RESET_ERR	重置错误。
8	TARGET_HELD_UNDER_RESET	已按下重置按钮。
9	TARGET_NOT_HALTED	目标设备无法停止。
10	TARGET_CMD_ERR	命令错误。
11	TARGET_APP_RESET_ERR	应用程序重置错误。
12	TARGET_VERSION_ERR	版本错误。
13	TARGET_GET_STATUS_ERR	获取状态错误。
14	TARGET_FORCE_HALT_ERR	强制停止错误。
16	TARGET_STLINK_SELECT_REQ	连接若干个主板。需要制定 -i <ST-LINK S/N>.
17	TARGET_STLINK_SERIAL_NOT_FOUND	当使用 -i 选项时，无法找到 ST-LINK 序列号。
18	TARGET_DEVICE_UNKNOWN_VENDOR	无法验证供应商信息。
255	TARGET_UNKNOWN_ERR	未知错误。

## 附录 B 配置文件格式

此附录介绍了利用配置文件启动 **ST-LINK GDB server** 所使用的配置文件格式。以下规则简要介绍了配置文件格式：

- 以一个#开头的行视为评论。
- 各个配置选项（参见第 1.1 节 **GDB server 启动选项**）指定为单行。

以下是配置文件示例：

```
#####
#### Sample Configuration File
#####

#####
# -e : Enables persistent mode
#####
-e

#####
# -f <Log-File> : Name of log file.
#####
-f debug_log_v0.txt

#####
# -l <Log-Level> : Logging level between 0 & 31
#####
-l 31

#####
# -p <Port-Number> : TCP-Listen Port-Number.
#####
-p 61234

#####
# -v : Enables verbose mode
#####
-v

#####
# -r <refresh-delay-sec> : Maximum Delay in status refresh
#####
-r 1
```

## 版本历史

表 2. 文档版本历史

日期	版本	变更
2019 年 4 月 18 日	1	初始版本。

## 目录

<b>1</b>	<b>GDB server 使用</b>	<b>4</b>
1.1	GDB server 启动选项	4
1.2	GDB server 工作模式	6
1.3	启动 GDB server	6
<b>2</b>	<b>利用 GDB 调试</b>	<b>7</b>
2.1	启动 GDB	7
2.2	连接至服务器	7
2.3	将程序加载至目标	7
2.4	设置断点	7
2.5	设置监控点	7
2.6	运行程序	8
2.7	退出调试会话	8
2.8	在目标 STM32 主板上调试	8
2.9	监控命令	8
<b>3</b>	<b>故障排除</b>	<b>9</b>
<b>附录 A</b>	<b>返回/错误代码信息</b>	<b>10</b>
<b>附录 B</b>	<b>配置文件格式</b>	<b>11</b>
	Revision history	12
	目录	13
	表一览	14
	图一览	15

## 表一览

表 1.	ST-LINK GDB 服务器返回/错误代码 .....	10
表 2.	文档版本历史 .....	12

## 图一览

图 1.	调试设置概览 .....	2
------	--------------	---

**IMPORTANT NOTICE – PLEASE READ CAREFULLY**

STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST's terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of Purchasers' products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, please refer to [www.st.com/trademarks](http://www.st.com/trademarks). All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2019 STMicroelectronics – All rights reserved