SMH Technologies®

Interfacing FlashRunner 2.0 with GD32 devices

Protocols

SWD for all GD32 devices.
We currently support GD32 programming through SWD (Serial Wire Debug) protocol.

```
#TCSETPAR CMODE [SWD]
```

- DIO 1 - RST
- DIO 2 - SWCLK a clock driven by FlashRunner.
- DIO 5 - SWDIO a bidirectional data line.

PIN Map

SWD:

- DIO 1 - RST
- DIO 2 - SWCLK
- DIO 5 - SWDIO

Families

The GD32 family of 32-bit microcontrollers based on the Arm® Cortex®-M processor is divided into four groups and each of these groups are divided into various GD32 series:

GD32 Entry Level MCUs

- **GD32E2 Series**
  - GD32E230
  - GD32E232
- **GD32F1 Series**
  - GD32F130
  - GD32F150
- **GD32F3 Series**
  - GD32F310
  - GD32F330
  - GD32F350

GD32 Mainstream MCUs

- **GD32C1 Series**
  - GD32C103
- **GD32E1 Series**
  - GD32E103
- **GD32E5 Series**
  - GD32E501
- **GD32F1 Series**

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com

info@smh-tech.com

- GD32F101
- GD32F103
- GD32F105
- GD32F107

- **GD32F3 Series**
  - GD32F303
  - GD32F305
  - GD32F307
- **GD32L2 Series**
  - GD32L233
- **GD32VF Series**
  - GD32VF103

GD32 High Performance MCUs

- **GD32E5 Series**
  - GD32E503
  - GD32E505
  - GD32E507
  - GD32E508
- **GD32F2 Series**
  - GD32F205
  - GD32F207
- **GD32W5 Series**
  - GD32W515
- **GD32F4 Series**
  - GD32F403
  - GD32F405
  - GD32F407
  - GD32F425
  - GD32F427
  - GD32F450
  - GD32F470

GD32 Specific MCUs

- **GD32EP Series**
  - GD32EPRT
- **GD32FF Series**
  - GD32FFPR

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com

info@smh-tech.com

SMH Technologies®

Supported Families

- **GD32E2 Series**
  - GD32E230
  - GD32E232
- **GD32F1 Series**
  - GD32F130
  - GD32F150
- **GD32F3 Series**
  - GD32F310
  - GD32F330
  - GD32F350

GD32 Mainstream MCUs

- **GD32C1 Series**
  - GD32C103
- **GD32E1 Series**
  - GD32E103
- **GD32E5 Series**
  - GD32E501
- **GD32F1 Series**
  - GD32F101
  - GD32F103
  - GD32F105
  - GD32F107
- **GD32F3 Series**
  - GD32F303
  - GD32F305
  - GD32F307
- **GD32L2 Series**
  - GD32L233
- **GD32VF Series**
  - GD32VF103 [Available under request]

GD32 High Performance MCUs

- **GD32E5 Series**
  - GD32E503
  - GD32E505
  - GD32E507
  - GD32E508
- **GD32F2 Series**

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com

info@smh-tech.com

- GD32F205
- GD32F207

- **GD32W5 Series**
  - GD32W515
- **GD32F4 Series**
  - GD32F403 [Available under request]
  - GD32F405
  - GD32F407
  - GD32F425
  - GD32F427
  - GD32F450
  - GD32F470

GD32 Specific MCUs

- **GD32EP Series**
  - GD32EPRT
- **GD32FF Series**
  - GD32FFPR [Available under request]

Available Operations

- **GD32 Series**

MEMORY	MASSERASE	ERASE PAGE	BLANKCHECK	PROGRAM	VERIFY R	VERIFY S	READ	DUMP
Flash [F]	✓	✓	✓	✓	✓	✓	✓	✓
System Memory [S]							✓	✓
Option Bytes [O]				✓	✓		✓	✓

Other commands available:

- `#TPCMD UNPROTECT`
- `#TPCMD GET_PROTECTION`
- `#TPCMD SET_PROTECTION [Value]`
- `#TPCMD CHECK_PROTECTION [Value]`
- `#TPCMD RESTORE_OPTION_BYTES`
- `#TPCMD OVERVIEW_OPTION_BYTES`
- `#TPCMD GET_DEVICE_INFORMATIONS`
- `#TPCMD ERASE [Start Address] [Size]`
- `#TPCMD WRITE_OPTION_BYTE [Address] [Value]`
- `#TPCMD COMPARE_OPTION_BYTE [Address] [Value]`

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com

info@smh-tech.com

Option Bytes

Option Bytes Organization

The Option Bytes are configured by the end user depending on the application requirements.
Depending on the GD32 device being examined, typically Option Bytes can have this form:
Option Bytes Organization case 1:
Option Byte is written on the first eighth bits of the register and from bit 16 to bit 23.
The complementary bits are written in the remaining part of the register.

Write Option Bytes - Start

What you need to know before you can write Option Bytes:
First of all, the Option Bytes are configuration bits of the device, so pay attention to which configuration you want to set and consequently which bits you will have to set to 1 or 0 in the various Option Byte registers.
An incorrect configuration could lead to device behaviors that are not what you would expect.
In addition to this, as previously mentioned, it is advisable to know how the Option Bytes are made in memory and how they should be written.
About the Security Protection Level, you need to know that you can move from level 0 to level 1 by changing the value of the SPC byte to any value, except the value 0xCC.
By programming the 0xCC value in the SPC byte, it is possible to go to level 2 either directly from level 0 or from level 1.
Once in level 2, it is no more possible to modify the Read protection level.
When the SPC is reprogrammed to the value 0xA5 to move from Level 1 to Level 0, typically a Masserase of the Flash main memory is performed.
If the SPC protection level is set to 1, in some GD32 it is not possible to read the memory area where the Option Bytes are present.

What does the GD32 driver do when writing Option Bytes

The GD32 driver does not consider the complementary part of the Option Bytes as input.
For example, if we consider Option Bytes Organization case 1 and have this Option Byte as input:
0x 00 FF 5A A5
The GD32 driver will read the Option Bytes in this way:
0x 00 FF 00 A5
So, GD32 driver doesn't consider the complementary part of the Option Bytes.
This occurs for three important reasons:

- Some GD32 self-calculate the complementary part of the Option Byte when writing the same, therefore it does not make sense to read the complementary part.
- If the GD32 device does not self-calculate the complementary part, the GD32 driver will do it, therefore it is not useful again to read the complementary part.
- Based on this reading method, the user will only have to modify the "active" part of the Option Bytes without having to pay attention to calculating the complementary part.

The GD32 driver does not know the configuration you are setting on the device, and it cannot do any checks to avoid critical situations.
For example, if you set the SPC Option Bytes to 0xCC, you set level 2 of Security Protection Level into device.

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com
info@smh-tech.com

The GD32 driver will write this Option Byte and subsequently it will no longer be possible to return to a lower protection level.
So please pay attention to what you are configuring via Option Bytes.
The GD32 driver checks all Option Bytes and also the complementary part of Option Bytes during a verify command.

Write, Read and Check Security Protection Level

We have implemented three commands to read, write and check the SPC value into all GD32 devices.
These three commands are:

```
#TPCMD GET_PROTECTION
#TPCMD SET_PROTECTION <Hex Value [0x00-0xFF]>
#TPCMD CHECK_PROTECTION <Integer Value [0-2]>
```

Pay attention to the fact that if the protection level is set at level 1 and you bring it to level 0, a device masserase is automatically performed.
Another feature to know is that all GD32 have the SPC value of level 0 equal to 0xA5.
You will not need to know how the Option Byte of the device is made or how it is programmed, the GD32 driver does all this for you.
This is the safest method to set the SPC if you don't know the device you are programming.
This is the scheme that allows you to understand how you can go from one level of Security Protection Level to another:
To avoid setting the level 2 protection by mistake, if the command #TPCMD SET_PROTECTION receives the value 0xCC as input, it will respond with this error:
So, as you are also reported on the LOG, the correct syntax to enable the Security Protection Level at level 2 is the following:

```
#TPCMD SET_PROTECTION 0xCC Y
```

Remember that after the execution of this command, you set the SPC level to 2, and it will be no longer possible to communicate via SWD protocol.

Parameters List

#TCSETPAR ENTRY_CLOCK

```
#TCSETPAR ENTRY_CLOCK
```

Default value is 4 MHz.
Frequency used into Connect procedure before raising the PLL of the device, if device PLL is available.

#TCSETPAR PLL_ENABLED

```
#TCSETPAR PLL_ENABLED
```

Default value YES.
Accepted parameters YES / NO.
Enable the PLL of the device at the highest possible frequency.

#TCSETPAR RESET_HARDWARE

```
#TCSETPAR RESET_HARDWARE
```


HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com
info@smh-tech.com

Default value NO.
Accepted parameters YES / NO.
Use Hardware reset (DIO1) into Connect procedure during Halt Cortex Core.
Please leave this parameter to NO except when it is strictly necessary.
Usually the Software Reset is enough to proceed with the reset of the device and to continue with the programming procedure.

#TCSETPAR SAMPLING_POINT

#TCSETPAR SAMPLING_POINT

Default value 17.
Accepted values are in the range 1-15.
Use this parameter to permanently set the sampling point of the FPGA.
It is recommended to leave this parameter with the default value.

Commands List

#TPCMD CONNECT

#TPCMD CONNECT

Connect function. Power on and entry.
Example of Connect Procedure wih LOG LEVEL 1:

```
|---#TPCMD CONNECT
|Protocol selected SWD.
|Entry Clock is 4.00 MHz.
|ID-Code read correctly at 4.00 MHz.
|Scanning AP map to find all APs.
|AP[0] IDR: 0x04770031, Type: AMBA AHB3 bus.
|Debug Port enabled.
|AP[0] ROM table base address 0xE00FF000.
|CPUID: 0x410CC601.
|Implementer Code: 0x41-[ARM].
|Found Cortex M0+ revision r0p1.
|Cortex M0+ Core halted [0.001 s].
|The device SPC's level is 0.
|PLL Enabled.
|Requested Clock is 37.50 MHz.
|Generated Clock is 37.50 MHz.
|Good samples: 6 [Range 4-9].
|IDCODE: 0x0BC11477.
|Designer: 0x23B, Part Number: 0xBC11, Version: 0x0.
|Time for Connect: 0.106 s.
```

#TPCMD MASSERASE

#TPCMD MASSERASE <Memory Type>

Masserase command isn't available for Option Bytes Area and OTP Area.
With this command, a Masserase of the device memory will be performed, whether this memory is made up of a single bank or two banks.
If the Security Protection Level (SPC) is set at level 1, this Masserase command will proceed by changing the protection level to level 0.
By this operation, an internal Masserase of the device will be automatically performed.

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com
info@smh-tech.com

Masserase depending on the SPC value:
If SPC level is 1, erase all memory of selected type with option bytes procedure.
If SPC level is 0, erase all memory of selected type with standard procedure if available.
Available for all GD32 Devices.

#TPCMD UNPROTECT

`#TPCMD UNPROTECT`

Remove the Flash write protected sectors and perform a Masserase F of all Flash memory.
If there are no write protected sectors in the flash memory, then a standard Masserase F is performed.
Available for all GD32 Devices.

#TPCMD ERASE

`#TPCMD ERASE <Memory Type>`

Erase command isn't available for Option Bytes Area and OTP Area.
Erase all memory with Page/Sector erase.
With this command, a Page/Sector Erase of the device FLASH memory will be performed, whether this memory is made up of a single bank or two banks.
Typically running the Page Erase of the entire Flash memory takes much longer than running the Masserase command.
If the Security Protection Level (SPC) isn't set at level 0, an error will be returned.
Available for all GD32 Devices.

`#TPCMD ERASE <Memory Type> <Start Address> <Size>`

Erase selected part of memory with sector/page erase.
Erase command isn't available for Option Bytes Area and OTP Area.
The driver will automatically convert the value of Start Address and Size to the corresponding number of pages.
If this command is performed on a device with two FLASH memory banks but configured on a single bank, the driver automatically manages the different size of the pages.
If the Read Protection Level (SPC) is not set to level 0, an error will be returned.
Available for all GD32 Devices.

#TPCMD BLANKCHECK

`#TPCMD BLANKCHECK <Memory Type>`

Verify if all memory of selected type is erased.
Blankcheck command isn't available for Option Bytes Area.
If the Read Protection Level (SPC) is not set to level 0, an error will be returned.
Available for all GD32 Devices.

`#TPCMD BLANKCHECK <Memory Type> <Start Address> <Size>`

Verify if selected part of memory of selected type is erased.
Blankcheck command isn't available for Option Bytes Area.
If the Read Protection Level (SPC) is not set to level 0, an error will be returned.
Available for all GD32 Devices.

#TPCMD PROGRAM

`#TPCMD PROGRAM <Memory Type>`

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com
info@smh-tech.com

Program all memory of selected type.
Available for all GD32 Devices.

```
#TPCMD PROGRAM <Memory Type> <Start Address> <Size>
```

Program selected part of memory of selected type.
Available for all GD32 Devices.

#TPCMD VERIFY

```
#TPCMD VERIFY <R/S> <Memory Type>
```

Verify all memory of selected type.
R: Readout - S: Checksum 32Bit.
Available for all GD32 Devices.

```
#TPCMD VERIFY <R/S> <Memory Type> <Start Address> <Size>
```

Verify selected part of memory of selected type.
R: Readout - S: Checksum 32Bit.
Available for all GD32 Devices.

#TPCMD READ

```
#TPCMD READ <Memory Type>
```

Read all memory of selected type.
Available for all GD32 Devices.

```
#TPCMD READ <Memory Type> <Start Address> <Size>
```

Read selected part of memory of selected type.
Available for all GD32 Devices.

#TPCMD DUMP

```
#TPCMD DUMP <Memory Type>
```

Dump all memory of selected type.
Available for all GD32 Devices.

```
#TPCMD DUMP <Memory Type> <Start Address> <Size>
```

Dump selected part of memory of selected type.
Available for all GD32 Devices.

#TPCMD RESTORE_OPTION_BYTES

```
#TPCMD RESTORE_OPTION_BYTES
```

This command is used to restore Option Bytes to a default value, when they are in a not optimal condition, due for example to an incorrect input file.
With this command, as you can guess from the name itself, it is possible to return the Option Bytes to the Default state (when this operation is possible).
The Default value of the Option Bytes is often the one present when the device leaves the factory.
More precisely, in the various Reference Manuals you can find the various default values for each GD32 family and subfamily.
Remember to use this command when you are not be able to restore manually the option bytes.
Available for all GD32 Devices.

#TPCMD OVERVIEW_OPTION_BYTES

```
#TPCMD OVERVIEW_OPTION_BYTES
```


HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com
info@smh-tech.com

Reads all option bytes present in the device and represents these values with various tables in the Real Time Log.
Available for all GD32 Devices.

#TPCMD GET_PROTECTION

`#TPCMD GET_PROTECTION`

Print into Real Time Log the current SPC protection present into target device.
Available for all GD32 Devices.

#TPCMD SET_PROTECTION

`#TPCMD SET_PROTECTION <Protection>`

Set inserted SPC protection level into target device.
Protection value in HEX format [0x00 - 0xFF].
The SPC level 0 is 0xA5 for all devices.
The SPC level 2 is 0xCC.
Other values set SPC to level 1.
When Level 1 is active, programming the protection option byte (SPC) to Level 0 causes the Flash memory to be mass-erased.
Available for all GD32 Devices.

#TPCMD CHECK_PROTECTION

`#TPCMD CHECK_PROTECTION <Protection>`

Check if current SPC protection level into device is the same as the one entered as a command parameter.
If it is different, the command returns an error.
The parameter values of this command can be 0, 1, 2.
Available for all GD32 Devices.

#TPCMD WRITE_OPTION_BYTE

`#TPCMD WRITE_OPTION_BYTE <Address> <Value>`

Writes the Option Byte to the selected address (Address) with the entered value (Value).
The reserved bits of the Option Bytes are not changed.
Be careful not to set the Security Protection Level to 0xCC (level 2) by mistake.
Available for all GD32 Devices.

#TPCMD COMPARE_OPTION_BYTE

`#TPCMD COMPARE_OPTION_BYTE <Address> <Value>`

Compare the Option Byte at the selected address (Address) with the entered value (Value).
The reserved bits of the Option Bytes are not considered.
Available for all GD32 Devices.

#TPCMD GET_DEVICE_INFORMATIONS

`#TPCMD GET_DEVICE_INFORMATIONS`

Reads all information present in the device and represents these values with a table in the Real Time Log.
Available for all GD32 Devices.

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com
info@smh-tech.com

#TPCMD GET_PROTECTED_SECTORS

`#TPCMD GET_PROTECTED_SECTORS`

Get all sectors Write-Erase Protected.
Available for all GD32 Devices.
Log Example:
[4KB] Protection enabled for Flash pages 0 ~ 3.
[4KB] Protection enabled for Flash pages 8 ~ 11.
[4KB] Protection enabled for Flash pages 16 ~ 19.
[4KB] Protection enabled for Flash pages 24 ~ 27.
[4KB] Protection enabled for Flash pages 32 ~ 35.
[4KB] Protection enabled for Flash pages 40 ~ 43.
[4KB] Protection enabled for Flash pages 48 ~ 51.
[4KB] Protection enabled for Flash pages 56 ~ 59.

#TPCMD READ_MEM8

`#TPCMD READ_MEM8 <Address> <Byte Count>`

It reads [Byte Count] bytes from the memory at the selected [Address] address and prints the data read in the Terminal and in the Real Time Log.
Available for all GD32 Devices.

#TPCMD READ_MEM16

`#TPCMD READ_MEM16 <Address> <16-bit Word Count>`

It reads [16-bit Word Count] 16-bit words from the memory at the selected [Address] address and prints the data read in the Terminal and in the Real Time Log.
Available for all GD32 Devices.

#TPCMD READ_MEM32

`#TPCMD READ_MEM32 <Address> <32-bit Word Count>`

It reads [32-bit Word Count] 32-bit words from the memory at the selected [Address] address and prints the data read in the Terminal and in the Real Time Log.
Available for all GD32 Devices.

#TPCMD RUN

`#TPCMD RUN`

Available for all GD32 Devices.

#TPCMD DISCONNECT

`#TPCMD DISCONNECT`

Disconnect function. Power off and exit.
Available for all GD32 Devices.

HQ and Registered Office
Via Giovanni Agnelli 1
33083 Villotta di Chions (PN) Italy
Società Unipersonale

Capitale sociale €102.040
P.I. 01697470936
C.F. 01697470936
REA PN-97255

D-U-N-S® 51-724-9350
T + 39 0434 421 111
F + 39 0434 639 021

→smh-tech.com
info@smh-tech.com