02/10/2023

Driver v. 5.02

Moreno Ortolan

# Interfacing FlashRunner 2.0 with MARVELL MRV88

# MARVELL 88Q Introduction

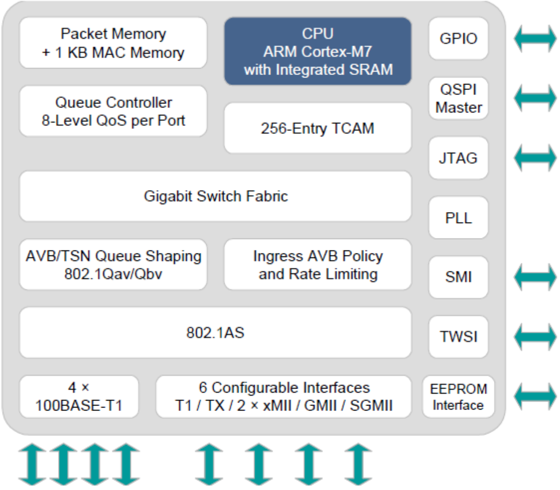
**88Q5050:**

The Marvell® 88Q5050/88Q5054 device is an AEC-Q100 qualified 8-port Ethernet switch, which is optimized for Automotive applications. Being equipped with configurable interfaces that support a combination of eight IEEE 100BASE-T1, 100BASE-TX, RGMII/RMII/MII, GMII, and SGMII ports, the switch is ideally suited for various application cases.

The switching core is designed to support all MAC ports operating at 1000 Mbps.

The 88Q5050/88Q5054 includes an ARM® CPU featured with a dedicated on-chip RAM, to support AVB protocols such as Precision Time Protocol (PTP). It provides support for TCAM with a Policy Control List (PCL) engine that supports 256 rules.

Both the low-power PHYs and the MACs integrated in the device comply fully with the applicable sections of the IEEE 802.3 standards. The IEEE 100BASE-T1 PHYs are all fully interoperable with the OPEN Alliance BroadR-Reach® (OABR) PHYs.



The 88Q5050/88Q5054 device's feature set is complemented by comprehensive local and remote management capabilities, which allow for easy access and configuration of the device.

**88Q5151, 885152, 88Q5192:**

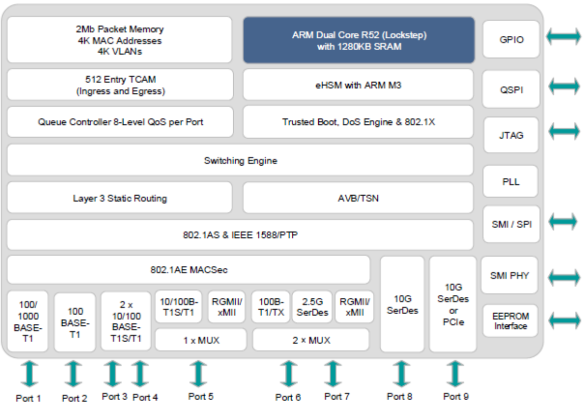
The Marvell® 88Q5152/88Q5151 is a single-chip integration of an AEC-Q100 qualified 9-Port Ethernet switch which is optimized for Automotive applications.

Being equipped with configurable interfaces that support a combination of integrated IEEE 1000BASE-T1, 100BASE-T1, 10BASE-T1S, 100BASE-TX, xMII, XFI, and PCIe ports, the switch is ideally suited for various application cases.

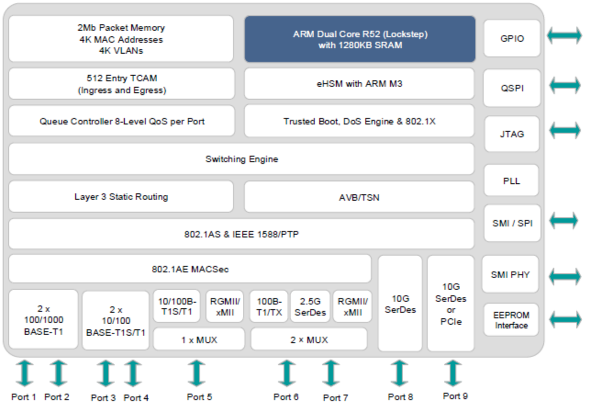
The switching core is designed to support all MAC ports operating at 30Gbps.
The 88Q5152/88Q5151 includes an ARM® Dual Core Lockstep R52 CPU with a dedicated on-chip RAM, to support AVB protocols such as Precision Time Protocol (PTP). It provides support for TCAM with a Policy Control List (PCL) engine that supports 512 rules. Both the low-power PHYs and the MACs integrated in the device comply fully with the applicable sections of the IEEE 802.3 standards.

The IEEE 100/1000BASE-T1 PHYs are all fully inter-operable with the OPEN Alliance BroadR-Reach® (OABR) PHYs.
The feature set of the 88Q5152/88Q5151 device is complemented by comprehensive local and remote management capabilities, which allow for easy access and configuration of the device.

**88Q5151:**



**88Q5152:**

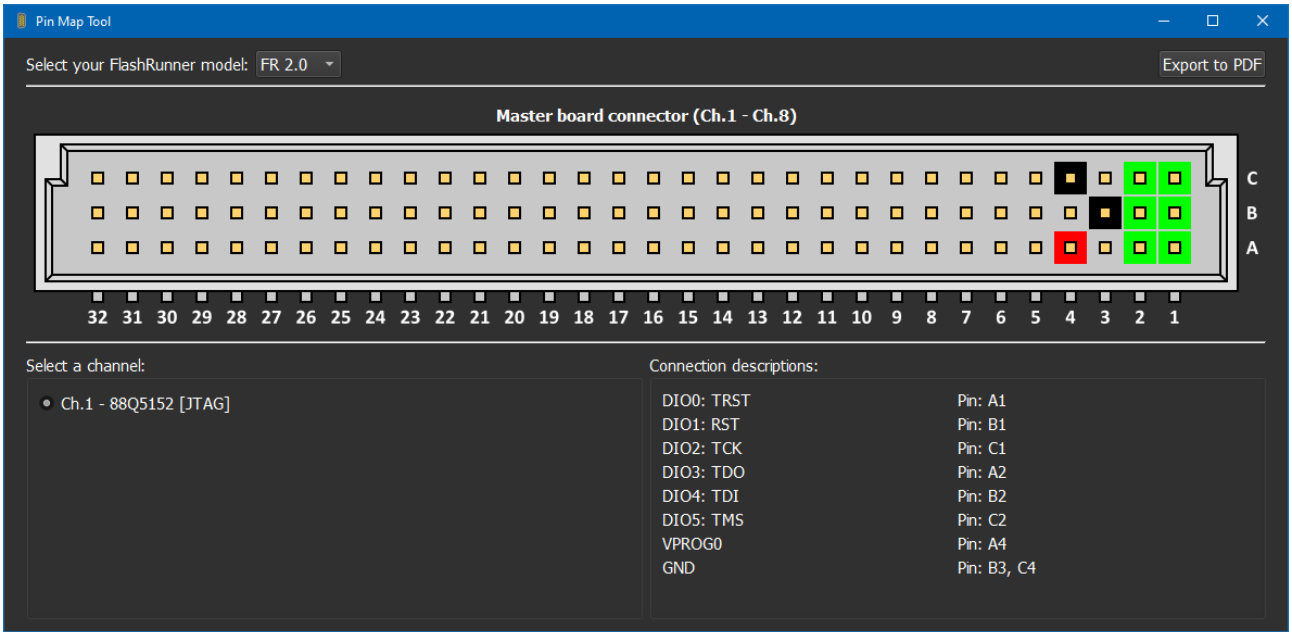


# MARVELL 88Q Protocol and PIN map

**88Q** devices support the JTAG protocol.

```
#TCSETPAR CMODE <JTAG>
```

## MARVELL 88Q PIN MAP



Select your FlashRunner model: FR 2.0

Select a channel:

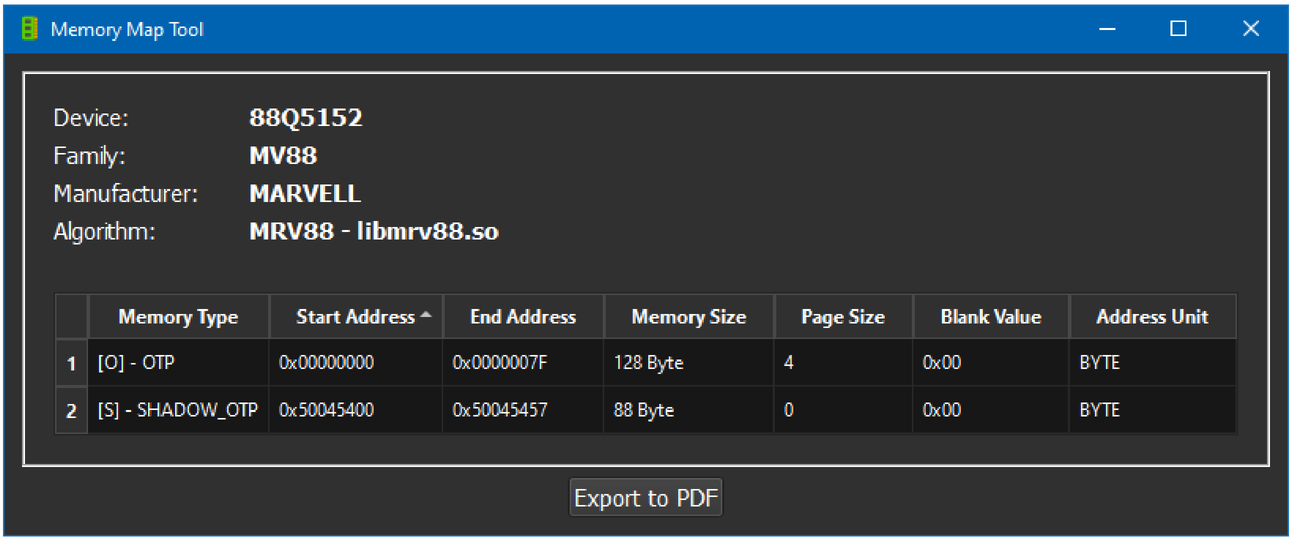
● Ch.1 - 88Q5152 [JTAG]

Connection descriptions:

| | |
|---|---|
| DIO0: TRST | Pin: A1 |
| DIO1: RST | Pin: B1 |
| DIO2: TCK | Pin: C1 |
| DIO3: TDO | Pin: A2 |
| DIO4: TDI | Pin: B2 |
| DIO5: TMS | Pin: C2 |
| VPROG0 | Pin: A4 |
| GND | Pin: B3, C4 |

## MARVELL 88Q Memory Map

| Memory Type | Start Address | End Address | Memory Size | Page Size | Blank Value | Address Unit |
|---|---|---|---|---|---|---|
| [O] - OTP | 0x00000000 | 0x0000007F | 128 Byte | 4 | 0x00 | BYTE |
| [S] - SHADOW_OTP | 0x50045400 | 0x50045457 | 88 Byte | 0 | 0x00 | BYTE |



Device: 88Q5152
Family: MV88
Manufacturer: MARVELL
Algorithm: MRV88 - libmrv88.so

| | Memory Type | Start Address ▲ | End Address | Memory Size | Page Size | Blank Value | Address Unit |
|---|---|---|---|---|---|---|---|
| 1 | [O] - OTP | 0x00000000 | 0x0000007F | 128 Byte | 4 | 0x00 | BYTE |
| 2 | [S] - SHADOW_OTP | 0x50045400 | 0x50045457 | 88 Byte | 0 | 0x00 | BYTE |

# MARVELL 88Q Driver Parameters

The standard parameters are used to configure some specific options inside 88Q driver.

## #TCSETPAR ENTRY_CLOCK

*Syntax:* `#TCSETPAR ENTRY_CLOCK <Frequency>`

<Frequency> Accepted parameters 4000000, 2000000, 1000000, 500000, 100000 Hz

*Description:* Set the JTAG/SWD frequency used in the Connect procedure before raising the PLL of the device, if the device PLL is available

*Note:* Default value 4.00 MHz

## #TCSETPAR SAMPLING_POINT

*Syntax:* `#TCSETPAR SAMPLING_POINT <Value>`

<Value> Accepted values are in the range 1-15

*Description:* Use this parameter to permanently set the sampling point of the FPGA
It is recommended to leave this parameter with the default value

*Note:* Default value 17

## #TCSETPAR RESET_HARDWARE [Obsolete]

*Syntax:* `#TCSETPAR RESET_HARDWARE <Value>`

<Value> Accepted parameters YES / NO

*Description:* 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

*Note:* This command is obsolete. It is available for retro compatibility only
Default value NO

## #TCSETPAR CONNECT_UNDER_RESET [Obsolete]

*Syntax:* `#TCSETPAR CONNECT_UNDER_RESET <Value>`

<Value> Accepted parameters YES / NO

*Description:* Perform a Hardware reset (DIO1) before the Connect procedure

*Note:* This command is obsolete. It is available for retro compatibility only
Default value NO

# MARVELL 88Q Driver Parameters for 88Q5050

There are eight specifics #TCSETPAR for **88Q5050** device:

```
#TCSETPAR KEY_DAT0 [255] - [223] - [191] - [159] - [127] - [95] - [63] - [31] - [0] - KEY_SFB_DAT0
#TCSETPAR KEY_ECC0 KEY_ECC0 - KEY_SFB_ECC0
#TCSETPAR KEY_DAT1 [255] - [223] - [191] - [159] - [127] - [95] - [63] - [31] - [0] - KEY_SFB_DAT1
#TCSETPAR KEY_ECC1 KEY_ECC1 - KEY_SFB_ECC1
#TCSETPAR KEY_DAT2 [255] - [223] - [191] - [159] - [127] - [95] - [63] - [31] - [0] - KEY_SFB_DAT2
#TCSETPAR KEY_ECC2 KEY_ECC2 - KEY_SFB_ECC2
#TCSETPAR KEY_DAT3 [255] - [223] - [191] - [159] - [127] - [95] - [63] - [31] - [0] - KEY_SFB_DAT3
#TCSETPAR KEY_ECC3 KEY_ECC3 - KEY_SFB_ECC3
```

These parameters are used to write the corresponding fuse.

If the bit KEY_SFB_DAT 0/1/2/3 or KEY_SFB_ECC 0/1/2/3 is set to 1, then it will be impossible to rewrite the corresponding eFUSE.

The internal procedure always programs these registers starting from the third up to the zero.
This is because, if we did the reverse procedure, there could be the possibility that by programming the KEY_DAT0 register the JTAG will be disabled forever.

In the driver's internal procedure, we must specify some choices made to handle various cases that could occur:

1. If user not set any #TCSETPAR but set #TPCMD PROGRAM E and/or #TPCMD VERIFY E R, there are no data to write into device, so the program/verify procedure is skipped and there's a warning into LOG like this:

```
Nothing to program. Skipping eFuses programming procedure.
Nothing to verify. Skipping eFuses verify procedure.
```

2. If a #TCSETPAR of an eFUSE is set and the user tries to program this eFUSE with the #TPCMD PROGRAM E procedure, but the corresponding KEY_SFB_DAT 0/1/2/3 or KEY_SFB_ECC 0/1/2/3 is already set to 1, it's impossible to write this eFUSE.

   To manage this situation, the programming procedure of this eFuses is skipped and the PROGRAM E procedure continue normally.

   There's a warning into LOG like this:

```
Warning: KEY_SFB_DAT 0/1/2/3 or KEY_SFB_ECC 0/1/2/3 is already set to 1
KEY DAT 0/1/2/3 or KEY ECC 0/1/2/3 eFUSE is not writable anymore.
Skipped programming procedure of this eFuse.
```

3. If the bits that disable the JTAG are set to 1 in KEY_DAT0, there is a special procedure for programming this eFUSE.

We need to do a special procedure because if we program the eFUSE directly with these bits, the JTAG is immediately disabled and it's impossible to continue communicating with the device.

So, in this case, in the program procedure there's this warning into LOG:

```
Skipped jtag disable bits. These bits are written only at the end of procedure if all is ok.
Is impossible to write these bits now. We need to maintain the jtag active for communication.
```

And if the KEY_SFB_DAT 0 bit is set to 1 there's another warning into LOG:

```
KEY_SFB_DAT0 bit is set to 1. We need to complete the eFuse procedure at the end of program.
KEY_SFB_DAT0 is set to 0. We will set it to 1 when we write the JTAG disable bits.
```

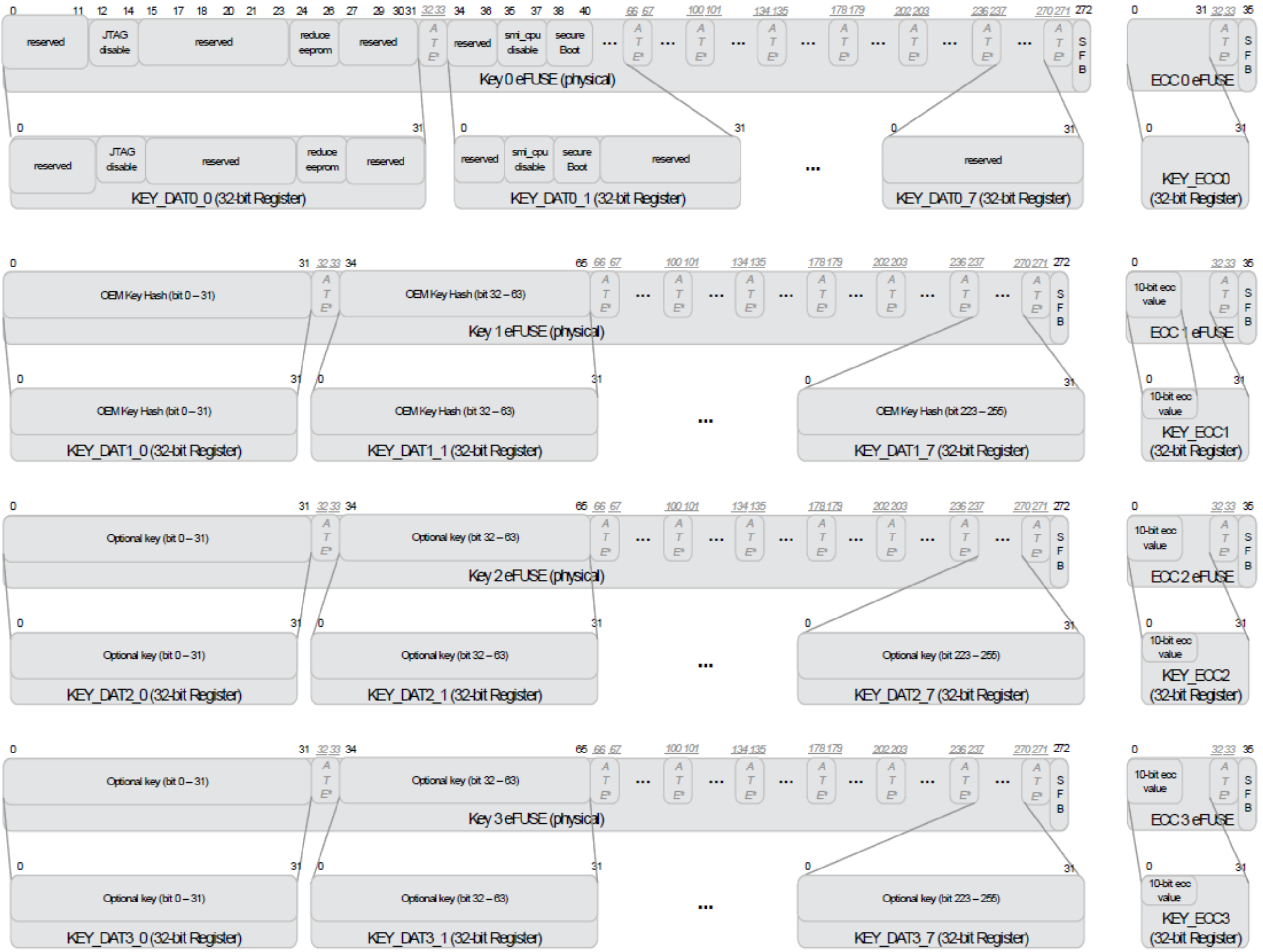
So, at this point, into **#TPCMD** PROGRAM_E procedure, we program the entire eFUSEs without the JTAG disable bits and with KEY_SFB_DAT0 set to 0.
After that, if all is ok, so **#TPCMD** PROGRAM_E response with "PASS" and the subsequent commands respond with a "PASS", in the **#TPCMD** DISCONNECT command we program the JTAG disable bits and KEY_SFB_DAT0 if it's set to 1.

If the Jtag disable bits are set into KEY DAT0 eFuse, the JTAG is disabled immediately, it's impossible to read/write a register inside 88Q5050 device.

Into LOG you can see this message:

```
Disable JTAG now.
```

At this point we need to close the communication because JTAG cannot longer be used.


reserved
JTAG disable
reduce eeprom
smi_cpu disable
secure Boot
ATE
SFB
Key 0 eFUSE (physical)
ECC 0 eFUSE
KEY_DAT0_0 (32-bit Register)
KEY_DAT0_1 (32-bit Register)
KEY_DAT0_7 (32-bit Register)
KEY_ECC0 (32-bit Register)
OEM Key Hash (bit 0 – 31)
OEM Key Hash (bit 32 – 63)
OEM Key Hash (bit 223 – 255)
Key 1 eFUSE (physical)
10-bit ecc value
ECC 1 eFUSE
KEY_DAT1_0 (32-bit Register)
KEY_DAT1_1 (32-bit Register)
KEY_DAT1_7 (32-bit Register)
KEY_ECC1 (32-bit Register)
Optional key (bit 0 – 31)
Optional key (bit 32 – 63)
Optional key (bit 223 – 255)
Key 2 eFUSE (physical)
ECC 2 eFUSE
KEY_DAT2_0 (32-bit Register)
KEY_DAT2_1 (32-bit Register)
KEY_DAT2_7 (32-bit Register)
KEY_ECC2 (32-bit Register)
Key 3 eFUSE (physical)
ECC 3 eFUSE
KEY_DAT3_0 (32-bit Register)
KEY_DAT3_1 (32-bit Register)
KEY_DAT3_7 (32-bit Register)
KEY_ECC3 (32-bit Register)

# MARVELL 88Q Driver Commands

Here you can find the complete list of all available commands for 88Q driver.

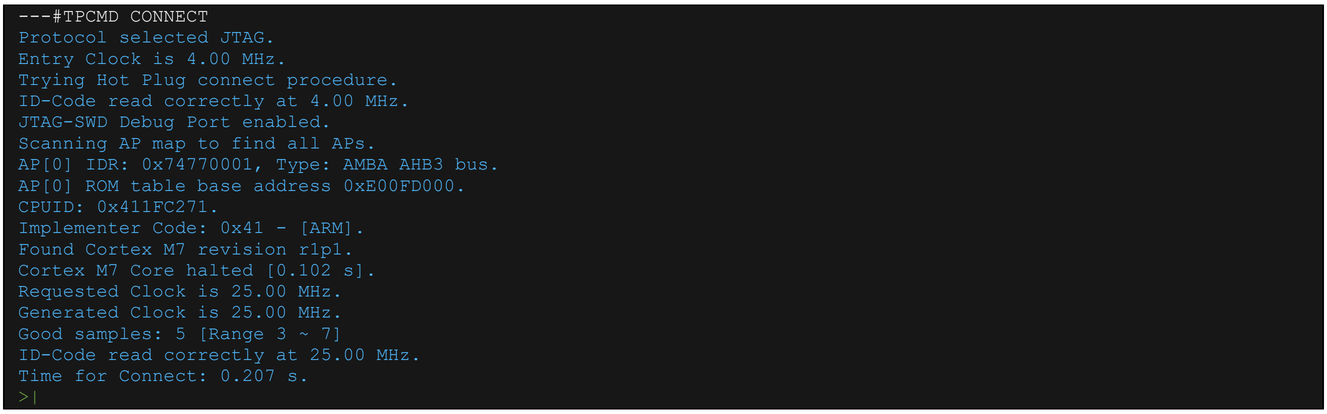
```
E → eFUSE
O → OTP area
S → Shadow OTP area
```

## #TPCMD CONNECT

**#TPCMD** CONNECT

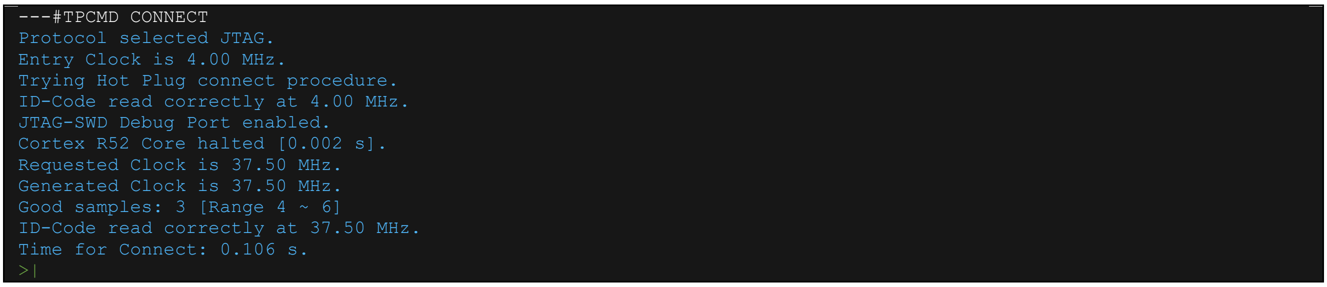
This function performs the entry and is the first command to be executed when starting the communication with the device.

**88Q5050:**



```
---#TPCMD CONNECT
Protocol selected JTAG.
Entry Clock is 4.00 MHz.
Trying Hot Plug connect procedure.
ID-Code read correctly at 4.00 MHz.
JTAG-SWD Debug Port enabled.
Scanning AP map to find all APs.
AP[0] IDR: 0x74770001, Type: AMBA AHB3 bus.
AP[0] ROM table base address 0xE00FD000.
CPUID: 0x411FC271.
Implementer Code: 0x41 - [ARM].
Found Cortex M7 revision r1p1.
Cortex M7 Core halted [0.102 s].
Requested Clock is 25.00 MHz.
Generated Clock is 25.00 MHz.
Good samples: 5 [Range 3 ~ 7]
ID-Code read correctly at 25.00 MHz.
Time for Connect: 0.207 s.
>|
```

**88Q5151:**



```
---#TPCMD CONNECT
Protocol selected JTAG.
Entry Clock is 4.00 MHz.
Trying Hot Plug connect procedure.
ID-Code read correctly at 4.00 MHz.
JTAG-SWD Debug Port enabled.
Cortex R52 Core halted [0.002 s].
Requested Clock is 37.50 MHz.
Generated Clock is 37.50 MHz.
Good samples: 3 [Range 4 ~ 6]
ID-Code read correctly at 37.50 MHz.
Time for Connect: 0.106 s.
>|
```

## #TPCMD BLANKCHECK

**#TPCMD** BLANKCHECK <E>

Blankcheck available only for eFUSE memory of 88Q5050 device.
Verify if all memory is erased.

## #TPCMD PROGRAM

**#TPCMD** PROGRAM <E>

Program available only for eFUSE memory of 88Q5050 device.
Programs all memory of the selected type based on the data in the FRB file.

**#TPCMD** PROGRAM <O>

Program available only for OTP memory of 88Q51x devices.
Programs all memory of the selected type based on the data in the FRB file.

#TPCMD PROGRAM <O> <start address> <size>
Program available only for OTP memory of 88Q51x devices.
Programs selected part of memory of the selected type based on the data in the FRB file.
Enter the Start Address and Size in hexadecimal format.

## #TPCMD VERIFY

#TPCMD VERIFY <E> <R>
R: Readout Mode.
Verify Readout available only for eFUSE memory of 88Q5050 device.
Verify all memory of the selected type based on the data in the FRB file.

#TPCMD VERIFY <O> <R>
R: Readout Mode.
Verify Readout available only for OTP memory of 88Q51x devices.
Verify all memory of the selected type based on the data in the FRB file.

#TPCMD VERIFY <O> <R> <start address> <size>
R: Readout Mode.
Verify Readout available only for OTP memory of 88Q51x devices.
Verify selected part of memory of the selected type based on the data in the FRB file.
Enter the Start Address and Size in hexadecimal format.

## #TPCMD READ

#TPCMD READ <E>
Read command available only for eFUSE memory of 88Q5050 device.
The result of the read command will be visible into the Terminal.

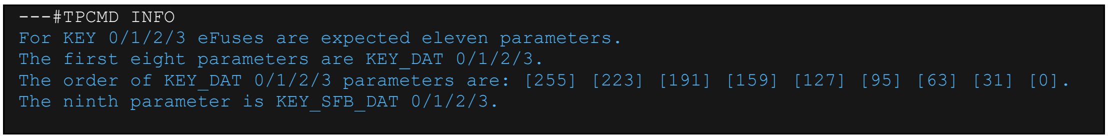
#TPCMD READ <O|S>
#TPCMD READ <O|S> <start address> <size>
Read command available for OTP and Shadow OTP memory of 88Q51x devices.
The result of the read command will be visible into the Terminal.

## #TPCMD DUMP

#TPCMD DUMP <O|S>
#TPCMD DUMP <O|S> <start address> <size>
Dump command available for OTP and Shadow OTP memory of 88Q51x devices.
The result of the dump command will be stored in the FlashRunner 2.0 internal memory.

## #TPCMD INFO

| | |
|---|---|
| *Syntax:* | #TPCMD INFO |
| *Prerequisites:* | This command can be executed without #TPCMD CONNECT |
| *Description:* | Print driver informations for 88Q5050 device |
| Note: | This command prints into Real Time Log |
| *Examples:* | Correct command execution: 😊 |



```
---#TPCMD INFO
For KEY 0/1/2/3 eFuses are expected eleven parameters.
The first eight parameters are KEY_DAT 0/1/2/3.
The order of KEY_DAT 0/1/2/3 parameters are: [255] [223] [191] [159] [127] [95] [63] [31] [0].
The ninth parameter is KEY_SFB_DAT 0/1/2/3.
```

```
For KEY_ECC 0/1/2/3 eFuses are expected two parameters.
The first parameter is KEY_ECC 0/1/2/3.
The second parameter is KEY_SFB_ECC 0/1/2/3.
Time for eFuse Info: 0.001 s
```

## #TPCMD READ_MEM8

*Syntax:* `#TPCMD READ_MEM8` <Address> <Byte Count>

<Address> Address in HEX format (i.e., 0x52002020)
<Byte Count> Byte count in decimal format (i.e., 8 -> eight bytes)

*Prerequisites:* none

*Description:* Read memory byte per byte from target 88Q device

Note: This command is available from driver version **4.06**
This command prints into Terminal and Real Time Log

*Examples:* Correct command execution: 😊

```
---#TPCMD READ_MEM8 0x52002020 8
Read[0x52002020]: 0xF0
Read[0x52002021]: 0xAA
Read[0x52002022]: 0x16
Read[0x52002023]: 0x14
Read[0x52002024]: 0x00
Read[0x52002025]: 0x00
Read[0x52002026]: 0x00
Read[0x52002027]: 0x00
Time for Read Mem: 0.002 s
```

## #TPCMD READ_MEM16

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

<Address> Address in HEX format (i.e., 0x52002020)
<16-bit Word Count> 16-bit Word count in decimal format (i.e., 4 -> four 16-bit words)

*Prerequisites:* none

*Description:* Read memory 16-bit word per 16-bit word from target 88Q device

*Note:* This command is available from driver version **4.06**
This command prints into Terminal and Real Time Log

*Examples:* Correct command execution: 😊

```
---#TPCMD READ_MEM16 0x52002020 4
Read[0x52002020]: 0xAAF0
Read[0x52002022]: 0x1416
Read[0x52002024]: 0x0000
Read[0x52002026]: 0x0000
Time for Read Mem: 0.002 s
```

## #TPCMD READ_MEM32

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

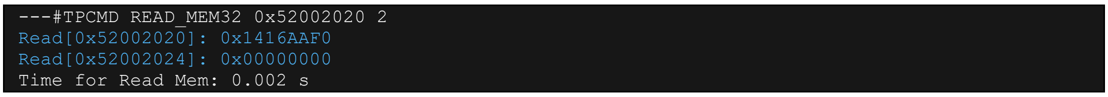
<Address> Address in HEX format (i.e., 0x52002020)
<32-bit Word Count> 32-bit Word count in decimal format (i.e., 2 -> two 32-bit words)

*Prerequisites:* none

*Description:* Read memory 32-bit word per 32-bit word from target 88Q device

*Note:* This command is available from driver version **4.06**
This command prints into Terminal and Real Time Log

*Examples:* Correct command execution: 😊



```
---#TPCMD READ_MEM32 0x52002020 2
Read[0x52002020]: 0x1416AAF0
Read[0x52002024]: 0x00000000
Time for Read Mem: 0.002 s
```

## #TPCMD DISCONNECT

**#TPCMD** DISCONNECT

Disconnect function. Power off and exit.

# MARVELL 88Q Driver Changelog

**Info about driver version 1.00 - 22/11/2019**
Supported 88Q5050 device.

**Info about driver version 1.01 - 04/01/2020**
Fixed verify readout procedure.

**Info about driver version 2.00 - 03/02/2020**
Supported FPGA static 9.

**Info about driver version 2.01 - 19/03/2020**
Fixed small printout bug of address value into Read command.

**Info about driver version 2.02 - 27/03/2020**
Now Connect command is able to detect if JTAG fuse are burned.

**Info about driver version 4.00 - 14/12/2020**
Added new JTAG Fpga to support FlashRunner HS. Added PLL, Sampling Point and DMA management.

**Info about driver version 4.01 - 06/04/2021**
Added new JTAG Fpga for FlashRunner HS GP2 and GP4.

**Info about driver version 4.02 - 19/07/2021**
Internal update for FPGA name and version tracking.

**Info about driver version 4.03 - 12/08/2021**
Internal upgrade of the algorithm, no change to the operations it performs.

**Info about driver version 4.04 - 25/08/2021**
Driver name changed from libmrv_88.so to libmrv88.so.
Internal upgrade of the algorithm, no change to the operations it performs.

**Info about driver version 4.05 - 04/11/2021**
Internal update, upgraded management for JTAG FPGA reset.

**Info about driver version 4.06 - 25/01/2022**
Added READ_MEM8, READ_MEM16, READ_MEM32 commands.

**Info about driver version 4.07 - 16/06/2022**
Print current FPGA version loaded into TPSTART command.
Upgraded internal code to align all drivers.

**Info about driver version 5.00 - 31/07/2022**
Added FPGA for new FlashRunner 2.0 models.

**Info about driver version 5.01 - 16/06/2023**
Supported 88Q51x series.

**Info about driver version 5.02 – 02/10/2023**
Internal driver update.