SMH Technologies

UNIVERSAL PRODUCTION
IN–SYSTEM PROGRAMMING

03/07/2017 Enrico Segatto

DC10909

AN00151: Interfacing FlashRunner with Infineon TriCore Devices

FlashRunner is a Universal In-System Programmer, which uses the principles of In-Circuit Programming to program Infineon TriCore family microcontrollers. This Application Note describes how to properly set up and use FlashRunner to program TriCore Flash devices

This Application Note assumes that you are familiar with both FlashRunner and the main features of the TriCore family. Full documentation about these topics is available in the FlashRunner user's manual and in device-specific datasheets.

1. Introduction

In-system programming of TriCore microcontrollers is performed through JTAG IEEE 1149.1 standard protocol.

In order to use FlashRunner to perform in-system programming, you need to implement the appropriate in-circuit programming hardware interface on your application board.

Thanks to its in-system programming capabilities, FlashRunner allows you to program or update the content of the Flash memory when the chip is already plugged on the application board.

2. Hardware Configuration

The microcontroller's lines needed to program a TriCore device are the following:

- **JTMS:** Test mode select.
- **JTCLK:** Test Clock.
- **JTDI:** Test Data In.
- **JTDO:** Test Data Out
- **RST:** Device reset input/output pin.
- **VDD:** Device power supply voltage.
- **VSS:** Device power supply ground.
- **JTRST:** JTAG Module Reset/Enable Input (optional)
- **BRKIN:** OCDS Break Input (Alternate Output) (optional)
- **CLOCK:** Clock Input/Output (optional)

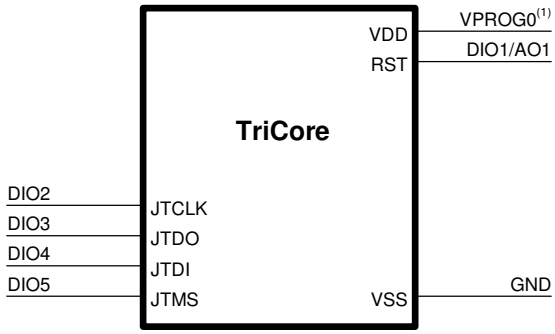
The optional lines indicated above can be connected to the FlashRunner's "ISP" lines in this way:

JTRST connected to **DIO0/AO0**

BRKIN connected to **DIO6**

CLOCK connected to **CLOCKOUT**

The lines mentioned above must be connected to the FlashRunner's "ISP" connector according to the following diagram:



(1) Connect this line if you want FlashRunner to automatically power the target device

3. Specific TCSETPAR Programming Commands

Overview

`TCSETPAR` commands set device-specific and programming algorithm-specific parameters. These commands must be sent after the **`TCSETDEV`** command and before a **`TPSTART`** / **`TPEND`** command block.

All of the following parameters must be correctly specified through the relative **`TCSETPAR`** commands (although the order with which these parameters are set is not important):

- VDD voltage;
- VDD_AUX voltage;
- Power Up time;
- Power Down time;
- Reset Up time;
- Reset Down time;
- Internal CPU frequency;
- Clock oscillator frequency driven by FlashRunner;
- Communication frequency.

TPSETPAR VDD

Command syntax:

`TCSETPAR VDD <voltage mV>`

Parameters:

`voltage mV`: Target device supply voltage, expressed in millivolts.

Description:

This command is used to properly generate the voltage level of the RST, JTCLK, JTDO, JTDI and JTMS signals. Additionally, the specified voltage is routed to the VPROG0 line of the FlashRunner's "ISP" connector, which can be used as a supply voltage for the target board.

TPSETPAR VDD_AUX

Command syntax:

`TCSETPAR VDD_AUX <voltage mV>`

Parameters:

`voltage mV`: Auxiliary supply voltage, expressed in millivolts, in the range 3000-14500mV.

Description:

This command is used to generate an optional, auxiliary voltage level for user purposes. The specified voltage is routed to the VPROG1 line of the FlashRunner "ISP" connector.

A value of 0 drives the VPROG1 line to GND. If the **`TCSETPAR VDD_AUX`** is not sent, the VPROG1 line is driven to HiZ.

TPSETPAR PWUP

Command syntax:

`TCSETPAR PWUP <time ms>`

Parameters:

`time ms`: Power rising time, expressed in milliseconds.

Description:

This command is necessary because, to enter the programming mode, FlashRunner must properly drive the V_{DD} line during the power-on reset.

The V_{DD} rising time (PWUP) is expressed in milliseconds and depends on the features of your target board. Make sure to choose a value large enough to ensure that the V_{DD} signal reaches the high logic level within the specified time. Note that, if the V_{DD} line has a high load, a longer time is required for the V_{DD} signal to reach the high logic level. If PWUP is not long enough, FlashRunner could not be able to enter the programming mode.

TPSETPAR PWDOWN

Command syntax:

`TCSETPAR PWDOWN <time ms>`

Parameters:

`time ms`: Power falling time, expressed in milliseconds.

Description:

The V_{DD} falling time (PWDOWN) is expressed in milliseconds and depends on the features of your target board. Make sure to choose a value large enough to ensure that the V_{DD} signal reaches the low logic level within the specified time. Note that, if the V_{DD} line has a high load, a longer time is required for the V_{DD} signal to reach the low logic level.

TPSETPAR RSTUP

Command syntax:

`TCSETPAR RSTUP <time µs>`

Parameters:

`time µs`: Reset rising time, expressed in microseconds.

Description:

The Reset rising time (RSTUP) is expressed in microseconds and depends on the features of your target board. Make sure to choose a value large enough to ensure that the Reset signal reaches the high logic level within the specified time. Note that, if the Reset line has a high load, a longer time is required for the Reset signal to reach the high logic level. If RSTUP isn't long enough, FlashRunner could not be able to enter the JTAG programming mode.

TPSETPAR RSTDOWN

Command syntax:

`TCSETPAR RSTDOWN <time µs>`

Parameters:

`time µs`: Reset falling time, expressed in microseconds.

Description:

The Reset falling time (RSTDOWN) is expressed in microseconds and depends on the features of your target board. Make sure to choose a value large enough to ensure that the Reset signal reaches the low logic level within the specified time. Note that, if the Reset line has a high load, a longer time is required for the Reset signal to reach the low logic level.

TCSETPAR JTCLK

Command syntax:

`TCSETPAR JTCLK <frequency Hz>`

Parameters:

`frequency Hz`: communication frequency, expressed in Hertz.

Description:

This command is used to set up the communication frequency between FlashRunner and target microcontroller. Must be lower than the FCPU.

TCSETPAR FCPU

Command syntax:

`TCSETPAR FCPU <frequency Hz>`

Parameters:

`Frequency Hz:` Internal CPU frequency, expressed in Hz.

Description:

This frequency depends by the value of the external clock source, by the package and by the settings of the internal PLL.

TCSETPAR CLKOUT

Command syntax:

`TCSETPAR CLKOUT 25000000 | 12500000 | 6250000 | 0`

Command options:

Frequency of a clock signal to be generated at the CLOCKOUT pin of the FlashRunner “ISP” connector, expressed in Hertz.

Description:

Generates an auxiliary clock signal at the CLOCKOUT pin of the FlashRunner “ISP” connector. This signal can be used to enter the target device’s JTAG mode when the target device does not have an external clock. Furthermore, this signal can be used to speed up programming (when you want to use a clock faster than that provided by your target board).

Make sure that the clock frequency you select isn’t greater than the maximum allowed frequency for your device. If the target device has an internal frequency divider, the actual device’s frequency will be a fraction of the CLKOUT frequency.

If you specify **`0`** as the CLKOUT frequency, no clock signal is generated.

.

4. Specific TPCMD Programming Commands

Overview

TPCMD commands perform a programming operation (i.e. mass erase, program, verify, etc.) These command must be sent within a **TPSTART** / **TPEND** command block.

Infineon TriCore-specific target programming commands are the following:

- **TPCMD MASSERASE**;
- **TPCMD BLANKCHECK**;
- **TPCMD PROGRAM**;
- **TPCMD VERIFY**;
- **TPCMD PROGRAMVERIFY;**
- **TPCMD RUN**.

TPCMD MASSERASE

Command syntax:

```
TPCMD MASSERASE P|D
```

Command options:

P|D:	Specifies PFlash (**P**), DFlash (**D**) memory.

Description:

It erases Pflash or Dflash. **'P'** parameter mass erases Flash memory. **'D'** mass erases DFlash memory.

TPCMD BLANKCHECK

Command syntax:

```
TPCMD BLANKCHECK P|D <tgt start addr> <len>
```

Command parameters and options:

P|D:	Specifies PFlash (**P**) or DFlash (**D**) memory.
tgt start address:	Device memory location from where the blankcheck operation will start.
len:	Number of locations to be blankchecked.

Description:

It blankchecks PFlash or Data memory. Blankchecks **len** locations starting from the address specified by **tgt start address**. The start address is not related to PDI memory map but is the logical address specified in each single device datasheet under “Memory” and “Program Memory Unit (PMU)” chapters.

TPCMD PROGRAM

Command syntax:

```
TPCMD PROGRAM P|D <src offset> <tgt start addr> <len>
```

Command parameters and options:

P|D:	Specifies PFlash (**P**), DFlash (**D**) memory.
src offset:	Offset from the beginning of the source memory.
tgt start addr:	Device memory location from where the program operation will start.
len:	Number of locations to be programmed.

Description:

It programs **`len`** locations of PFlash or DFlash memory starting from the **`tgt start addr`** address.

PFlash memory is organized as 32-bit locations: **`len`** specifies the number of locations to be programmed. **`tgt start addr`** is not related to PDI memory map but is the logical address specified in each single device datasheet under “Memory” and “Program Memory Unit (PMU)” chapters.

DFlash memory is organized as 32-bit locations: **len** specifies the number of locations to be programmed**. `tgt start addr`** is not related to PDI memory map but is the logical address specified in each single device datasheet under “Memory” and “Program Memory Unit (PMU)” chapters.

TPCMD VERIFY

Command syntax:

`TPCMD VERIFY P|D R|S <src offset> <tgt start addr> <len>`

Command parameters and options:

`P|D`:	Specifies PFlash (**`P`**) or DFlash (D) memory.
`R|S`	Specifies Readout (**`R`**) or CRC (**`S`**).
`src offset`:	Offset from the beginning of the source memory.
`tgt start addr`:	Device memory location from where the verify operation will start.
`len`:	Number of locations to be verified

Description:

It verifies **`len`** locations of Flash or DFlash memory starting from the **`tgt start addr`** address. The CRC parameter print back the CRC of the entire flash memory calculated from target microcontroller.

PFlash memory is organized as 32-bit locations: **len** specifies the number of locations to be verified.

DFlash memory is organized as 32-bit locations: **len** specifies the number of locations to be verified.

TPCMD PROGRAMVERIFY

Command syntax:

```
TPCMD PROGRAMVERIFY P|D <src offset> <tgt start addr> <len>
```

Command parameters and options:

P|D:	Specifies PFlash (**P**) or DFlash (D) memory.
src offset:	Offset from the beginning of the source memory.
tgt start addr:	Device memory location from where the verify operation will start.
len:	Number of locations to be verified

Description:

It programs and verifies **len** locations of PFlash or DFlash memory starting from the **tgt start addr** address, with a unique interaction.

PFlash memory is organized as 32-bit locations: **len** specifies the number of locations to be programmed and verified. **tgt start addr** is not related to PDI memory map but is the logical address specified in each single device datasheet under “Memory” and “Program Memory Unit (PMU)” chapters.

DFlash memory is organized as 32-bit locations: **len** specifies the number of locations to be programmed and verified**. tgt start addr** is not related to PDI memory map but is the logical address specified in each single device datasheet under “Memory” and “Program Memory Unit (PMU)” chapters

Note for PROGRAMVERIFY

ProgramVerify command is faster than the separated Program and Verify but less secure.
It's available only for few devices.

TPCMD RUN

Command syntax:

```
TPCMD RUN
```

Command parameters:

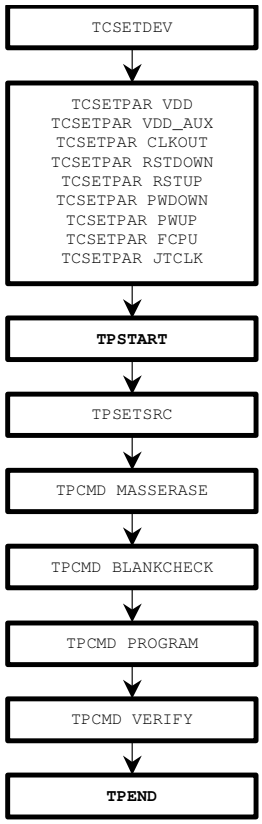
None.

Description:

It runs the target application.

5. Typical Programming Flow

The following flow chart illustrates typical steps to help you write your own script file.



6. Script Example

The example below shows a typical programming flow for an Infineon TriCore TC1793 device.

```
;
; FLASHRUNNER SCRIPT EXAMPLE FOR INFINEON TC1793
;
; Use this example as a starting point for your specific programming needs
;
; ----------
;
; Hardware connections
;
; DIO0/AO0 (JTRST - optional)
; DIO1/AO1 (RST)
; DIO2     (JTCLK)
; DIO3     (JTDO)
; DIO4     (JTDI)
; DIO5     (JTMS)
; DIO6     (BRKIN - optional)
; CLKOUT   (CLOCK - optional)
;

; Turns off logging
#LOG_ON 2
; Halt on errors
#HALT_ON FAIL

; Sets device
TCSETDEV INFINEON TC1793 INF_C

;-------------------------
; FLASHRUNNER I/O Settings
;-------------------------

; Target voltage, mV
TCSETPAR VDD 3300

; VPROG1 voltage, mV (from 3000 to 14500, 0 to disable)(change as needed)
TCSETPAR VDD_AUX 0

; Clock oscillator frequency driven by FlashRunner, Hz
; Possible frequencies are: 25000000, 12500000, 6250000, 0 (DISABLED)
```

```
TCSETPAR CLKOUT 0

; RESET down time (from 0 us to 65535 us)
TCSETPAR RSTDOWN 1000
; RESET up time (from 0 us to 65535 us)
TCSETPAR RSTUP 1000

; Power down time (from 0 ms to 65535 ms)
TCSETPAR PWDOWN 10
; Power up time (from 0 ms to 65535 ms)
TCSETPAR PWUP 10

;-----------------------------------
; TRICORE AUDO MAX ALGO Settings
;-----------------------------------

; Internal CPU frequency, Hz (change as needed)
; This frequency depends by the value of the external clock source,
; by the package and by the settings of the internal PLL.
; Default Internal CPU frequency is 100000000 Hz with PLL in Freerunning Mode.
; For this device the maximum FCPU is 270000000 Hz
TCSETPAR FCPU 100000000

; Set the JTCLK frequency (change as needed)
; NOTE: JTCLK frequency must be lower than FCPU
TCSETPAR JTCLK 12500000

;-----------------------------
; Start Programming operations
;-----------------------------

; Starts programming block
TPSTART

; Image file to be programmed (must be placed in the \BINARIES directory)
TPSETSRC FILE TEST.FRB

;------------------------
; PROGRAM FLASH commands
;------------------------

; Mass erases PFlash memory
TPCMD MASSERASE P

; Blank checks PFlash memory (change address and length as needed)
```

```
TPCMD BLANKCHECK P $A0000000 $200000

; Programs PFlash memory (change source and target addresses and length as needed)
TPCMD PROGRAM P $0 $A0000000 $200000

; Verifies PFlash memory, R = read-out method, S = checksum method (change source and target addresses and length as needed)
TPCMD VERIFY P R $0 $A0000000 $200000

; Blank checks PFlash memory (change address and length as needed)
TPCMD BLANKCHECK P $A0800000 $200000

; Programs PFlash memory (change source and target addresses and length as needed)
TPCMD PROGRAM P $200000 $A0800000 $200000

; Verifies PFlash memory, R = read-out method, S = checksum method (change source and target addresses and length as needed)
TPCMD VERIFY P R $200000 $A0800000 $200000

;--------------------
; DATA FLASH commands
;--------------------

; Mass erases DFlash memory
TPCMD MASSERASE D

; Blank checks DFlash memory (change address and length as needed)
TPCMD BLANKCHECK D $AF000000 $18000

; Programs DFlash memory (change source and target addresses and length as needed)
TPCMD PROGRAM D $400000 $AF000000 $18000

; Verifies DFlash memory, R = read-out method, S = checksum method (change source and target addresses and length as needed)
TPCMD VERIFY D R $400000 $AF000000 $18000

; Blank checks DFlash memory (change address and length as needed)
TPCMD BLANKCHECK D $AF080000 $18000
; Programs DFlash memory (change source and target addresses and length as needed)
TPCMD PROGRAM D $418000 $AF080000 $18000

; Verifies DFlash memory, R = read-out method, S = checksum method (change source and target addresses and length as needed)
TPCMD VERIFY D R $418000 $AF080000 $18000
;--------------------------
; End Programming operation
;--------------------------
; Ends programming block
TPEND
```

The FlashRunner's system software setup will install script examples specific for each device of the TriCore family on your PC.

7. Programming Times

The following table shows programming times for selected Infineon TriCore devices.

Device	Mem. Size	Conditions	Operations	Time
TriCore TC1793	4MB Pflash 192KB DFlash	JTCLK=12.5MHz Checksum Verify	Erase + Blank Check + Program + Verify	275,18 s
TriCore TC1793	4MB Pflash 192KB DFlash	JTCLK=12.5MHz Read Out Verify	Erase + Blank Check + Program + Verify	408,48 s
TriCore TC264D-40F200W	2.5MB PFlash 96KB DFlash	JTCLK=12.5MHz Read Out Verify	Erase + Blank Check + Program + Verify	199,41 s
TriCore TC297TP-128F300S	8MB Pflash 96KB DFlash	JTCLK=12.5MHz Read Out Verify	Erase + Blank Check + Program + Verify	584,25 s
TriCore TC277TP-64F200S	4MB Pflash 448KB Dflash	JTCLK=12.5MHz Read Out Verify	Erase + Blank Check + Program + Verify	342,51 s
TriCore TC277TP-64F200S	4MB Pflash 448KB Dflash	JTCLK=12.5MHz Read Out Verify	Erase + Blank Check + ProgramVerify	187,94 s

Programming times depend on Programming Algorithm version, target board connections, communication mode, target microcontroller mask, and other conditions. Programming times for your actual system may therefore be different than the ones listed here. SMH Technologies reserves the right to modify Programming Algorithms at any time.

8. References

FlashRunner user's manual

Microcontroller-specific datasheets