

AN00166: Interfacing FlashRunner with Cypress PSoC3 Devices

FlashRunner is a Universal In-System Programmer, which uses the principles of In-Circuit Programming to program Cypress PSoC3 family microcontrollers. This Application Notes describes how to properly set up and use FlashRunner to program PSoC3 Flash devices

This Application Note assumes that you are familiar with both FlashRunner and the main features of the PSoC3 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 PSoC3 microcontrollers is performed through JTAG IEEE 1149.1-2001 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 capability, 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

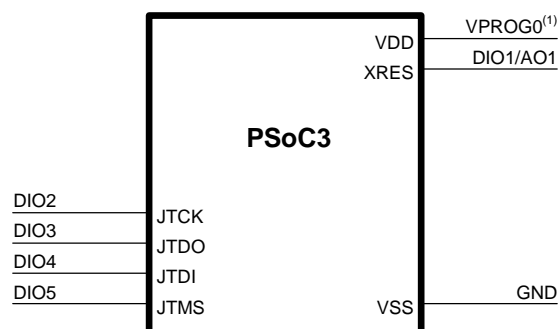
The microcontroller's lines needed to program a PSoC3 device are the following:

- **JTMS:** Test mode select.
- **JTCK:** Test Clock.
- **JTDI:** Test Data In.
- **JTDO:** Test Data Out
- **XRES:** Device reset input/output pin.
- **VDD:** Device power supply voltage.
- **VSS:** Device power supply ground.
- **CLOCK:** Clock Input/Output (optional)

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

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;
- Configuration Data FRB file;
- 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 XRES, JTCK, 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 μ s>

Parameters:

time μ s: 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 μ s>

Parameters:

time μ s: 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 CONFDATA

Command syntax:

TCSETPAR CONFDATA <filename.FRB>

Parameters:

filename.FRB: name of the file with the Configuration Data stored.

Description:

The name of the file has to specify the extension .FRB. This parameter is not necessary if you want to use the ECC setting with ECCEN bit programmed as '1'; in this way you will program only 256 bytes per row.

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.

Cypress PSoC3-specific target programming commands are the following:

- **TPCMD MASSERASE;**
- **TPCMD ECC;**
- **TPCMD BLANKCHECK;**
- **TPCMD PROGRAM;**
- **TPCMD VERIFY;**
- **TPCMD PROTECT;**
- **TPCMD READ;**
- **TPCMD RUN.**

TPCMD MASSERASE

Command syntax:

TPCMD MASSERASE F|E

Command options:

F|E: Specifies Flash (**F**), EEPROM (**E**) memory.

Description:

It erases Flash or EEPROM. '**F**' parameter erases all Flash data and ECC/configuration bytes, all Flash protection rows and all row latches in all Flash arrays on the device. '**E**' erases EEPROM memory.

TPCMD BLANKCHECK

Command syntax:

TPCMD BLANKCHECK F|E <tgt start addr> <len>

Command parameters and options:

- F|E:** Specifies Flash (**F**) or EEPROM (**E**) memory.
- tgt start address:** Device memory location from where the blankcheck operation will start.
- len:** Number of locations to be blankchecked.

Description:

It blankchecks Flash or EEPROM memory. Blankchecks **len** locations starting from the address specified by **tgt start address**.

TPCMD ECC

Command syntax:

TPCMD ECC <src ECC address>

Command options:

- src ECC address:** Specifies the address in FRB file where the byte with ECCEN bit is located.

Description:

It reads the byte of Device Configuration NVL with the ECCEN bit and it calculate if the Flash row are organized in 256 or 288 bytes per row.

This command is mandatory in each script execution.

It prints out the number of bytes per row foreseen by customer's source code.

Array Type	Size (Bytes)
Flash, with ECC Enabled	256
Flash, with ECC Disabled	288 (256 data bytes plus 32 configuration bytes)
EEPROM	16

TPCMD PROGRAM FLASH OR EEPROM

Command syntax:

TPCMD PROGRAM F|E <src offset> <tgt start addr> <len>

Command parameters and options:

- F|E:** Specifies Flash (**F**), EEPROM (**E**) 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 Flash or EEPROM memory starting from the **tgt start addr** address.

Flash memory is organized as 256 bytes or 288 bytes per row according to the ECCEN bit stored in the byte 3 of NVL: if ECCEN = '1' ECC is enabled (and so 256 bytes per row); if ECCEN = '0' ECC is disabled (and so 288 bytes per row). **len** specifies the number of locations to be programmed.

EEPROM memory is organized in 16 bytes row. **len** specifies the number of locations to be programmed.

The Flash memory is programmable in row (256 or 288 bytes per row) unit. The EEPROM is programmable in row (16 bytes per row) unit.

Register Address	7	6	5	4	3	2	1	0
0x00	PRT3RDM[1:0]		PRT2RDM[1:0]		PRT1RDM[1:0]		PRT0RDM[1:0]	
0x01	PRT12RDM[1:0]		PRT6RDM[1:0]		PRT5RDM[1:0]		PRT4RDM[1:0]	
0x02	XRESMEN						PRT15RDM[1:0]	
0x03	DIG_PHS_DLY[3:0]				ECCEN		DPS[1:0]	
							CFGSPED	

TPCMD PROGRAM NVL OR WONVL

Command syntax:

TPCMD PROGRAM L|W <src start address>

Command parameters and options:

L|W: Specifies Nonvolatile Latch (L), Write Once Nonvolatile Latch (W) memory.

src start address: Offset from the beginning of the source memory

Description:

The Nonvolatile Latch (NVL) are 4 bytes for device configuration; they are located starting from address h90000000 of the hex file.

The Write Once (WO) NVL are 4-bytes located in address h90100000 of the hex file. Note that programming WO NVL with the correct 32-bit key (0x50536F43) makes the device One Time Programmable (OTP). Any other key value does not have any impact on device security. Include this step after understanding its implications and only if it is required for the end application.

TPCMD VERIFY READ OUT

Command syntax:

```
TPCMD VERIFY F|E <src offset> <tgt start addr> <len>
```

Command parameters and options:

F E:	Specifies Flash (F) or EEPROM (E) 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 verifies through read out method **len** locations of Flash or EEPROM memory starting from the **tgt start addr** address.

TPCMD VERIFY CHECKSUM

Command syntax:

```
TPCMD VERIFY S <src offset> <data register addr> <len>
```

Command parameters and options:

- s:** Specifies the checksum (s) verify method.
- src offset:** Offset from the beginning of the source memory.
- data register addr:** Device memory location from where the verify operation will start.
- len:** Number of locations to be verified

Description:

It verifies through read out method **len** locations of memory starting from the **data register addr** address. **data register addr** is the address of the register with stored the checksum values calculated by the PSoC3 device.

It prints out the values of the 2 bytes with the checksum validated.

TPCMD PROTECT

Command syntax:

```
TPCMD PROTECT P|V <src start address>
```

Command options:

- src start address:** Offset from the beginning of the source memory

Description:

It programs (P) the Protection Flash data according to the number of bytes per row calculated from customer's source code.

It verifies (✓) the Protection Flash data.

Every Flash row has 2 protection bits in the protection area. The protection rows start in address h90400000 in the hex file.

Mode	Description	Read ^a	External Write ^b	Internal Write ^c
00	Unprotected	Yes	Yes	Yes
01	Read Protect	No	Yes	Yes
10	Disable External Write	No	No	Yes
11	Disable Internal Write	No	No	No

- a. Read – Applies to Test Controller and Read Commands.
b. External Write – Test Controller/third-party programmers.
c. Internal Write – Boot loading or writes due to firmware execution.

TPCMD READ

Command syntax:

TPCMD READ F|E <tgt start addr> <len>

Command parameters and options:

- F|E:** Specifies Flash (F) or EEPROM (E) memory.
- tgt start addr:** Device memory location from where the read operation will start.
- len:** Number of locations to be verified

Description:

It reads **len** locations of Flash or EEPROM memory starting from the **tgt start addr** address. The length has to be a multiple of a row size in Flash read operations, considering the ECC enabled or disabled (the row size is of 256 or 288 bytes respectively) or a multiple of a row size for EEPROM read operations (without considering the ECC that is related only to the Flash configuration).

TPCMD RUN

Command syntax:

TPCMD RUN

Command parameters:

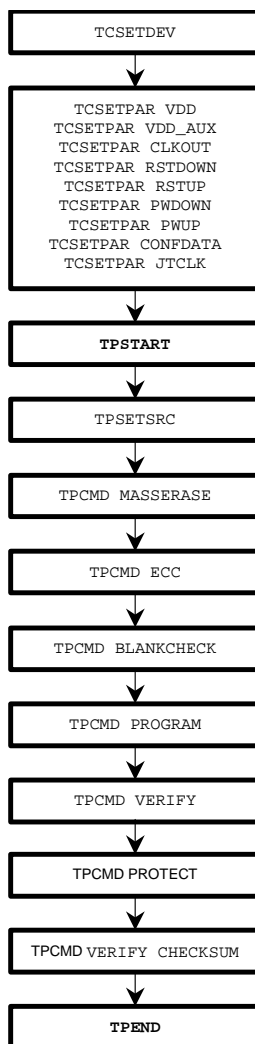
None.

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 a Cypress PSoC3 CY8C3666AXI-052 device.

```
;
; FLASHRUNNER SCRIPT EXAMPLE FOR CYPRESS CY8C3666AXI-052
;
; Use this example as a starting point for your specific programming needs
;
; -----
;
; Hardware connections
;
; DIO1 (XRES)
; DIO2 (JTCK)
; DIO3 (JTDO)
; DIO4 (JTDI)
; DIO5 (JTMS)
;

; Turns off logging
#LOG_OFF

; Halt on errors
#HALT_ON FAIL

; Sets device
TCSETDEV CYPRESS CY8C3666AXI-052 CY_B

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

; Target voltage, mV (change as needed)
TCSETPAR VDD 5500

; 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

; Selects the Configuration Data FRB file.
TCSETPAR CONFDATA CONFDATA.FRB

;-----
; CY_B ALGO Settings
;-----

; Set the frequency of the JTAG channel, Hz (change as needed)
TCSETPAR JTCLK 2000000

;-----
; Start Programming operation
;-----

; Starts programming block
TPSTART

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

; Mass erases Flash memory
TPCMD MASSERASE F

; The following commands check if the ECC is enabled or disabled. They are MANDATORY.
TPSETSRC FILE DCNVL.FRB
TPCMD ECC $3

; Blank checks Flash memory (change address and length as needed)
TPCMD BLANKCHECK F $0 $10000

; Programs Nonvolatile Latch
TPCMD PROGRAM L $0

```

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

; Programs Flash memory (change source, target address and length as needed)
TPCMD PROGRAM F $0 $0 $10000

; Verifies Flash memory (change source, target address and length as needed)
TPCMD VERIFY F $0 $0 $10000

;-----
; EEPROM commands
;-----

; Mass erases EEPROM memory
TPCMD MASSERASE E

; Blank checks Flash memory (change address and length as needed)
TPCMD BLANKCHECK E $8000 $800

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

; Programs EEPROM memory (change source, target address and length as needed)
TPCMD PROGRAM E $0 $8000 $800

; Verifies EEPROM memory (change source, target address and length as needed)
TPCMD VERIFY E $0 $8000 $800

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

; Programs Write Once Nonvolatile Latch
; TPCMD PROGRAM W $0

;-----
; PROTECTION commands
;-----

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

; Programs Flash protection
TPCMD PROTECT P $0

; Verifies Flash protection
```

```
; TPCMD PROTECT V $0  
  
;-----  
; CHECKSUM verify  
;-----  
  
; Image file to be programmed (must be placed in the \BINARIES directory)  
TPSETSRC FILE CHKSUM.FRB  
  
; Verifies Memory Checksum  
TPCMD VERIFY S $0 $4720 2  
  
; Ends programming block  
TPEND
```

The FlashRunner's system software setup will install script examples specific for each device of the PsoC3 family on your PC.

7. Cypress PSoC3 Specific Error

Cypress PSoC3-Specific Errors		
\$4D00	TCSETDEV	command: manufacturer not supported
\$4D01	TCSETDEV	command: algorithm not found on card
\$4D02	TCSETDEV	command: device not supported
\$4D03	TCSETDEV	command: internal hardware configuration error
\$4D04	TCSETDEV	command: internal error in algorithm closure
\$4D05	TCSETDEV	command: error due to algorithm name mismatch
\$4D06	TCSETPAR	command: missing parameter
\$4D07	TCSETPAR CLKOUT	command: missing parameter
\$4D08	TCSETPAR RSTDOWN	command: missing, invalid or out of range parameter
\$4D09	TCSETPAR RSTUP	command: missing, invalid or out of range parameter
\$4D0A	TCSETPAR JTCLK	command: missing parameter
\$4D0B	TCSETPAR JTCLK	command: invalid or out of range parameter
\$4D0C	TCSETPAR PWDOWN	command: missing, invalid or out of range parameter
\$4D0D	TCSETPAR PWUP	command: missing, invalid or out of range parameter
\$4D0E	TCSETPAR VDD	command: missing, invalid or out of range parameter
\$4D0F	TCSETPAR VDD_AUX	command: missing, invalid or out of range parameter
\$4D10	TCSETPAR CONFDATA	command: missing or invalid parameter
\$4D11	TPCMD	command: missing parameter
\$4D12	TPCMD	command: parameter not supported
\$4D13	TPCMD RUN	command: execution error
\$4D14	TPCMD MASSERASE	command: missing or invalid parameter
\$4D15	TPCMD MASSERASE	command: invalid parameter
\$4D16	TPCMD MASSERASE	command: Flash masserase error
\$4D17	TPCMD BLANKCHECK	command: missing or invalid parameter
\$4D18	TPCMD BLANKCHECK	command: invalid parameter
\$4D19	TPCMD BLANKCHECK	command: Flash blankcheck error
\$4D1A	TPCMD BLANKCHECK	command: EEPROM blankcheck error
\$4D1B	TPCMD PROGRAM	command: missing or invalid parameter
\$4D1C	TPCMD PROGRAM	command: invalid parameter
\$4D1D	TPCMD PROGRAM	command: source offset parameter out of range
\$4D1E	TPCMD PROGRAM	command: Flash program error
\$4D1F	TPCMD PROGRAM	command: Non Volatile Latch program error
\$4D20	TPCMD PROGRAM	command: Write Once Non Volatile Latch program error
\$4D21	TPCMD PROGRAM	command: EEPROM program error
\$4D22	TPCMD VERIFY	command: missing or invalid parameter
\$4D23	TPCMD VERIFY	command: target start address parameter or length parameter out of range
\$4D24	TPCMD VERIFY	command: source offset parameter out of range
\$4D25	TPCMD VERIFY	command: Flash verify error Algo layer
\$4D26	TPCMD VERIFY	command: Flash verify error API layer
\$4D27	TPCMD VERIFY	command: checksum error
\$4D28	TPCMD READ	command: missing parameter
\$4D29	TPCMD READ	command: invalid parameter
\$4D2A	TPCMD READ	command: read execution error
\$4D2B	TPCMD PROTECT	command: missing or invalid parameter

\$4D2C	TPCMD PROTECT command: protection execution error
\$4D2D	TPCMD PROTECT command: error in opening protection source file
\$4D2E	TPCMD ECC command: ECC check error
\$4D2F	TPCMD ECC command: error opening ECC source file
\$4D30	TPSTART command: execution error
\$4D31	TPSTART command: missing parameter
\$4D32	TPEND command: execution error
\$4D68	TPCMD read function: attempt to read a protected sector
\$4D69	TPCMD write function: attempt to write a protected sector

8. Programming Times

The following table shows programming times for selected Cypress PsoC3 CY8C3866AXI-040 devices.

Device	Mem. Size	Conditions	Operations	Time
Cypress CY8C3866AXI-040	64KB Flash 2KB EEPROM	JTCLK=8MHz 256 bytes per row	Erase + Blank Check + Program + Verify + Protect	19,37 s
Cypress CY8C3866AXI-040	64KB Flash 2KB EEPROM	JTCLK=8MHz 288 bytes per row	Erase + Blank Check + Program + Verify + Protect	24,51 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.

9. References

FlashRunner user's manual

Microcontroller-specific datasheets