Fusion Digital Power Designer Command Line Overview

Texas Instruments

Updated: August 30, 2022

Overview

Scriptable" way to:

- Download configuration (files exported from Fusion GUI): *FusionConfigWriter*
- Download script file (.csv) (script file exported from Fusion GUI): *FusionScriptRunner*
- Export device's configuration to a file: *FusionDeviceExporter*
- Read single PMBus command: *FusionParamReader*
- Write a single PMBus command: *FusionParamWriter*
- Download firmware (UCD devices): *FusionFirmwareDownload*

Download & Install

- Command Line tools are installed typically in the "C:\Program Files\Fusion Digital Power Designer\bin" or equivalent. By default, this folder is added to your PATH when you install Fusion Digital Power Designer. This allows you to type the command name from any folder on your PC. If this not the case, either manually add the path in your PC's window PATH, or re-run the installer and select the checkbox shown below
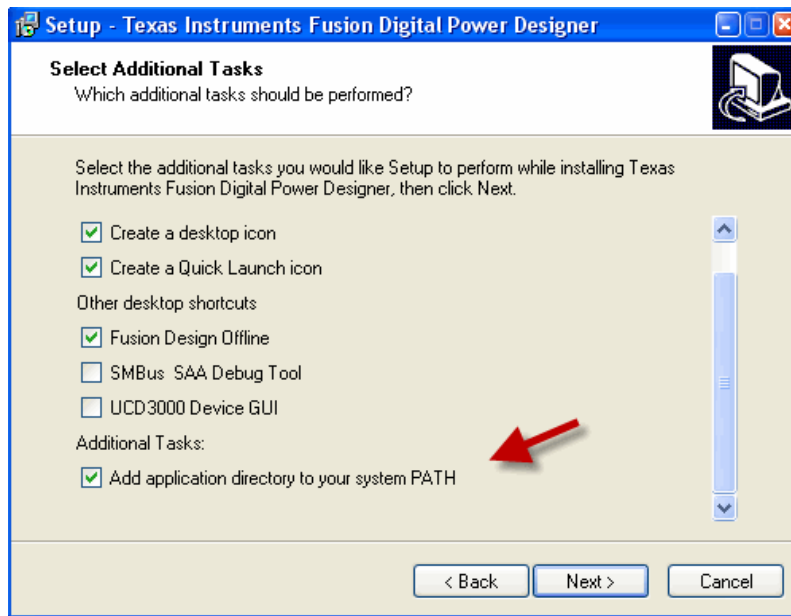
Common Flags

Flag Types

- Flag: no input requires followed the flag. If the flag present, indicate a true value.
- Input: User must enter a value followed the flag, either a string, integer, or decimal number.

--help

Description: display all the flags

Type: Flag.

FusionConfigWriter –help

FusionParamWriter –help

FusionParamReader –help

--address:

Description: Specify the address of the device. NOTE that users must enter either --*address* flag, or --*scan* flag in the commmand line/

Type: Input as integer. Value = from 1d to 127d

FusionParamReader --address 126 0x8B

FusionParamWriter --address 92 --cmd-id CLEAR_FAULTS

FusionConfigWriter --address 92 --project --infile 174_0001_430_UCD9246_92_000032.xml --erase-data-flash-before-importing 0xFF --force-write --project-file-warnings-ok --reset-when-done always --reset-wait-time 1000

--scan:

Description: Scan for the target device. By default this will just look for the first device that responds to a DEVICE_ID and/or IC_DEVICE_ID, and/or DEVICE_CODE read.. If –scan-mode flag is omited, the program use default scan-mode which is combo. See --scan-mode for more detail. NOTE that users must enter either --*address* flag, or --*scan* flag in the commmand line.

Type: Flag

- Use default scan-mode which is combo

FusionParamWriter --scan --cmd-id 0x9A   --value-encoded 0x4D46525F4D4F44454C

- Use device-id as scan-mode

FusionParamWriter --scan --scan-mode device_id --cmd-id VOUT_COMMAND --rail 3   --value-encoded 0x34CD

--scan-mode:

Description: Specify what device discovery method to use.

- device_id is for devices that implement the DEVICE_ID command (read block 0xFD). UCD92xx, UCD91xx, UCD90xxx, and UCD3xxx implement this.
- device_code is for devices that implement the DEVICE_CODE command (read word 0xFE). TPS404xx devices use this scheme.
- Ic_device_id: for command using PMBus Standard command IC_DEVICE_ID (Read block 0xAD).
- combo to look at DEVICE_ID and and then IC_DEVICE_ID, and then DEVICE_CODE..

Type: Input options is below:

Values = device_id | device_code | ic_device_id | combo
Default = combo

--quiet

Description: Normally, progress/status information about the selected task is displayed on standard output. To disable this and only print error messages (to standard error), pass this flag. Note that, regardless of the setting of this option, Fusion tools will set a 0 exit code on success and 1 on failure.

Input: Flag (true | false)
Default = false

FusionConfigWriter

Flags Options:

See Common Flags for more detail: --help, --address, --scan, --scan-mode, --device-exact-version

--infile

Description: Specify input file for the write. The –infile flag should followed by one of the flag to specify what type of file it is: --project, --dflash, or –system-file.

Type: Input as string. File name stored configuration to import to device.

--system-file

Description: file is system file (.tifsp) => Enable system file write mode. Device ID and PMBus address must match exactly for the importing.

Input: Flag (true | false)
Default = false

--dflash

Description: file is data flash file (.hex) => Enable EEPROM data flash write mode.

Input: Flag (true | false)
Default = false

--project

Description: file is project file (.xml) => enable project file write mode.

Type: Flag (true | false)
Default = false

--force-write

Description: In project file mode, write all parameters, even if the value currently on the device does not differ from that in the project file. Applied to project file (.xml) only. This flag is ignore for other file types.

Input: Flag (true | false)
Default = false

--project-file-warning-ok

Description:Normally, if there are any warnings about a project file you are trying to import -- such as it contains commands that are no longer supported on the target device or is missing some parameters that the target device supports -- FusionConfigWriter will exit after printing the warnings. Pass this option to display the warnings but continue on with the configuration write anyway.

Input: Flag (true | false)
Default = false

--all-or-none

Description: Used for system file mode only. If set to true, all devices in the system must be present to continue with the operation. Applied to system file (.tifsp) only. This flag is ignore for other file type.

Input: Flag (true | false)
Default = false

--skip-stopping-power-conversion-on-rails

Description: Normally, writes to commands such as PHASE_INFO, GPIO_SEQ_CONFIG, and FREQUENCY_SWITCH (UCD92XX), require stopping power conversion on all rails. Skip this step, to save some downloading time, only if either device is blank device, or rails already stop (or are not required to stop) power conversion before downloading configuration; otherwise, downloading configuration might fail.

Input: Flag (true | false)
Default = false

--import-using-nvm-data

Description: Option to import NVM configuration stored in project file. If project file does not contain NVM configuration, will return error.
NOTE: available in GUI 7.7.3 or later

Input: Flag (true | false)
Default = false

--save-config-as-project

Description:Save current device configuration as project outfile (xml). If importing data flash/NVM configuration, make sure to perform device's reset so that NVM configuration is synchronized with PMBus configuration. If importing PMBus configuration, make sure to perform STORE_DEFAULT/STORE_USER_ALL to synchronize NVM config with PMBus config.
NOTE: available in GUI 7.7.3 or later

Input: project file (.xml) to be exported to. Users enter absolute path + file name

--don't-perform-device-reset

Description: Normally, writes to commands such as PHASE_INFO and GPIO_SEQ_CONFIG (UCD92XX) and GPI_CONFIG, GPO_CONFIG, SEQ_CONFIG (UCD90XX) are done first, then STORE_DEFAULT_ALL is done, then

the device is reset, and then other writes continue. If true, the step to reset the device will be skipped. Apply only to downloading project file.

Input: true | false
Default = false

--reset-when-done

Description: In project file mode, controls whether the device is reset before exiting. Available modes are never, always, and updated. "never" is the default, which does not perform any additional device resets after programming has completed. \"always\" will always reset the device before exiting, even if no parameters were written. If a parameter was written, the reset is performed after STORE_DEFAULT_ALL is executed. "updated" will only perform a reset if the device's configuration was updated. The reset is performed after STORE_DEFAULT_ALL is executed.

Input: never | always | updated
Default = never

--reset-wait-time

Description: "The maximum wait time for device to come back to live after reset command is issued to reset the device. Minimum is 500 ms. Default is 5000 ms."

Input: integer value from 500 to number specified by users
Default = 5000

--store-default-wait-time-after-executing

Description: The wait time after Store_To_Flash command is issued to store configuration to non-volatile memory. Default is 2000 ms. Minimum value is 1000 ms. Apply only to downloading project file

Input: 1000 to number specified by users
Default = 2000

--store-default-wait-time-before-executing

Description: The wait time before sending Store_To_Flash command to store configuration to non-volatile memory. Default is 0 ms, except UCD90XX devices which require at least 1000 ms. If the wait time is less than the minimum wait time required by the part, it will be ignored.

Input: zero to number specified by users
Default = 0

--erase-data-flash-before-importing

Description: Erase the data flash and fill data flash with the specified pattern (either 0xFF or 0xAA pattern) before importing the configuration

Input: 0xFF | 0xAA
Default = 0xFF

Examples:

Example 2: Import project file at specific address

Scan for device at address 92d by sending PMBus command Adh, FDh, and FCh to address 92. If device is found, then erase data flash, fill data flash with pattern 0xFF. Check if device firmware version is exact "5.7.0.11359" to continue with importing. Import project file using PMBus configuration, force write all PMBus commands, ignore all warnings. When complete, reset device, wait for 1000 ms before doing something else

FusionConfigWriter --address 92 --project --infile 174_0001_430_UCD9246_92_000032.xml --erase-data-flash-before-importing 0xFF --force-write --project-file-warnings-ok --reset-when-done always --reset-wait-time 1000 --device-exact-version "5.7.0.11359"

Example 3: Import project file to first device found (address 1d to 127d)

Scan for device from address 1 to 127d, by sending PMBus command Adh, FDh, and FCh to address 1 to 127. If device is found, then erase data flash, fill data flash with pattern 0xFF. Import project file using Data Flash configuration, force write all PMBus commands, ignore all warnings. When complete, reset device, wait for 5000 ms before saving the update configuration to "C:\Temp\my_output_project.xml"

FusionConfigWriter --scan --project --infile 174_0001_430_UCD9246_92_000032.xml --erase-data-flash-before-importing 0xFF --project-file-warnings-ok --reset-when-done always --reset-wait-time 5000 –import-using-nvm-data –save-config-as-project "C:\Temp\my_output_project.xml"

Example 4: save project file (xml) from first found device using DEVICE_ID as scan mode

Scan for device from address 1 to 127d, by sending PMBus command FDh to address 1 to 127. If device is found, then save device configuration to "C:\Temp\my_output_project.xml"

FusionConfigWriter --scan --scan-mode device_id –save-config-as-project "C:\Temp\my_output_project.xml"l

Example 5: save project file (xml) for specific address

If there are multiple devices on the system, but want to save config from a specific device, then set the --address flag instead of using –scan flag

FusionConfigWriter –address 92 -scan-mode ic_device_id –save-config-as-project "C:\Temp\my_output_project.xml"l

Example 6: Import data flash (.hex) to first found device using IC_DEVICE_ID as scan mode

Scan for device from address 1 to 127d, by sending PMBus command Adh to address 1 to 127. If device is found, then erase data flash, fill data flash with pattern 0xAA. Import data flash file in Intel Hex format. When complete, reset device, wait for 5000 ms before saving the update configuration to "C:\Temp\my_output_project.xml"

FusionConfigWriter –scan -scan-mode ic_device_id --dflash --infile my_data_flash_config_file.hex --erase-data-flash-before-importing 0xAA --reset-when-done always --reset-wait-time 5000 –save-config-as-project "C:\Temp\my_output_project.xml"l

FusionScriptRunner

Flags Options

--address

Description: For PMBus Programmer scripts -- which do not embed a device address in each line -- set the address to target. For other scripts, override addresses present in the script in favor of the specified address. You SHOULD NOT use this option for scripts that interact with the UCD ROM, since the ROM has a fixed address of 11d or 0x42. FYI, firmware upgrade scripts interact with the ROM.

Type: Input as integer. Value is from 1d to 127d

--infile

Description: Specify input file for the write.

Type: Input as string. Script file name to import to device.

--ignore-read-mismatch-errors

Description: If script file contains step to verify each write-command with read-back, or verify configuration after importing by reading back the entire configuration and compare read-back value with write-value, enter this flag will ignore the mismatch and continue with operation.

Type: Flag (true | false)
Default = false

--ignore-read-errors

Description: If a ReadByte, ReadWord, or BlockRead is NACKed or there is any other error, script execution will normally stop. If this flag is specified, script execution will continue and the error will be noted in output.

Type: Flag (true | false)
Default = false

--ignore-write-errors

Description: If a SendByte, WriteByte, WriteWord, or BlockWrite is NACKed or there is any other error, script execution will normally stop. If this flag is specified, script execution will continue and the error will be noted in output.

Type: Flag (true | false)
Default = false

--extra-delay

Description: Extra delay, in millseconds, to add after any device read or write is performed. This could be used to slow down a script to monitor for system changes based on the script actions.

Type: Input as integer.

Examples

Example 2: Import script to device, using device's address in script (--address is omited)
FusionScriptRunner *--infile 174_0001_430_UCD9246_92_000032.csv*

Example 2: import script to specified device
Overwrite address stored in script file with the specified address. Let say address stored in script file is 101d, the below wil send all commands to device address 92d

FusionScriptRunner *--address 92 --infile 174_0001_430_UCD9246_92_000032.csv*

Example 3:import to device at address 92, ignore all reads, writes error
FusionScriptRunner *--address 92 --infile 174_0001_430_UCD9246_92_000032.csv --ignore-read-errors --ignore-write-errors*

Example 4: ignore read-back mismatch when verify the write, or verify the configuration
FusionScriptRunner *--infile 174_0001_430_UCD9246_92_000032.csv   --ignore-read-mismatch-errors*

FusionDeviceExporter

Flags Options

--outfile
Description: output file.

Type: Input as string. File name to be exported to.

--project
Description: export configuration as PMBus command to project file (.xml).

Type: Flag (true | false)
Default = false

--dflash
Description: export configuration stored in data flash to a file

Input: Flag (true | false)
Default = false

--eeprom-format
Description: The format of the EEPROM data flash image. Specify one of srec or intel.
Input: Input. Value = srec | intel. Default value is 'intel'.

--text
Description: export configuration to text file (.txt).

Input: Flag (true | false)
Default = false

--pmbus-script

Description: export configuration to script file (.csv). Script file contains sequence of necessary steps (SMBus or I2C writes, pause, power cycle device, etc.) to write the full device configuration via PMBus (command-by-command).

Input: Flag (true | false)
Default = false

.

--dflash-script

Description: export configuration to script file (.csv). Script file contains sequence of necessary steps to write the full device configuration via a write of the data flash.

Input: Flag (true | false)
Default = false

.

--script-format

Description: For --dflash-script and --pmbus-script exports, defines whether the configuration will be written in terms of SMBus or I2C. Specify one of smbus or i2c.

Input: Input. Value = smbus | i2c.

--quiet

Description: Normally, progress/status information about the selected task is displayed on standard output. To disable this and only print error messages (to standard error), pass this flag. Note that, regardless of the setting of this option, FusionDeviceExporter will set a 0 exit code on success and 1 on failure.

Input: Flag (true | false)
Default = false

Examples

Example 1: export to project file (.xml)
FusionDeviceExporter *--address 92 --project --outfile 'C:\Temp\TPS544C22 @ PMBus Address 27d Project.xml'*

Example 2: export to data flash file
FusionDeviceExporter *--scan 92 --dflash --outfile 'C:\Temp\ucd9090-dflash.srec' --eeprom-format srec*

FusionDeviceExporter *–scan –scan-mode device-id --dflash 'C:\Temp\ucd9090-dflash.hex' --eeprom-format intel*

Example 2: export to script file (.csv)
FusionDeviceExporter *--address 92 --pmbus-script --outfile 'C:\Temp\ucd9090-pmbus-i2c-script.csv' --script-format i2c*

FusionDeviceExporter *--address 92 --pmbus-script --outfile 'C:\Temp\ucd9090-pmbus-script.csv' --script-format smbus*

FusionDeviceExporter *--address 92 --dflash-script --outfile 'C:\Temp\ucd9090-dflash-i2c-script.csv' --script-format i2c*

FusionDeviceExporter *--address 92 --dflash-script --outfile 'C:\Temp\ucd9090-dflash-script.csv' --script-format smbus --quiet*

FusionParamReader:

Flags Options

See Common Flags for more detail: --help, --address, --scan, --scan-mode, --device-exact-version

--rail

Description: Specify the rail number for the command. Numbering starts at 1.

- If you specify --rail for a command that is not "PAGEd", the program will exit will an error.
- If you do not specify --rail and a command is PAGEd
  - If device has more than one rail configured, the program will exit will an error.
  - If device has only one rail configured, the program will write the value for rail #1.

Type: Input as integer. Value starts at 1.

--no-header

Description: Normally a column header is printed before the actual device readings. --no-header will supress this.
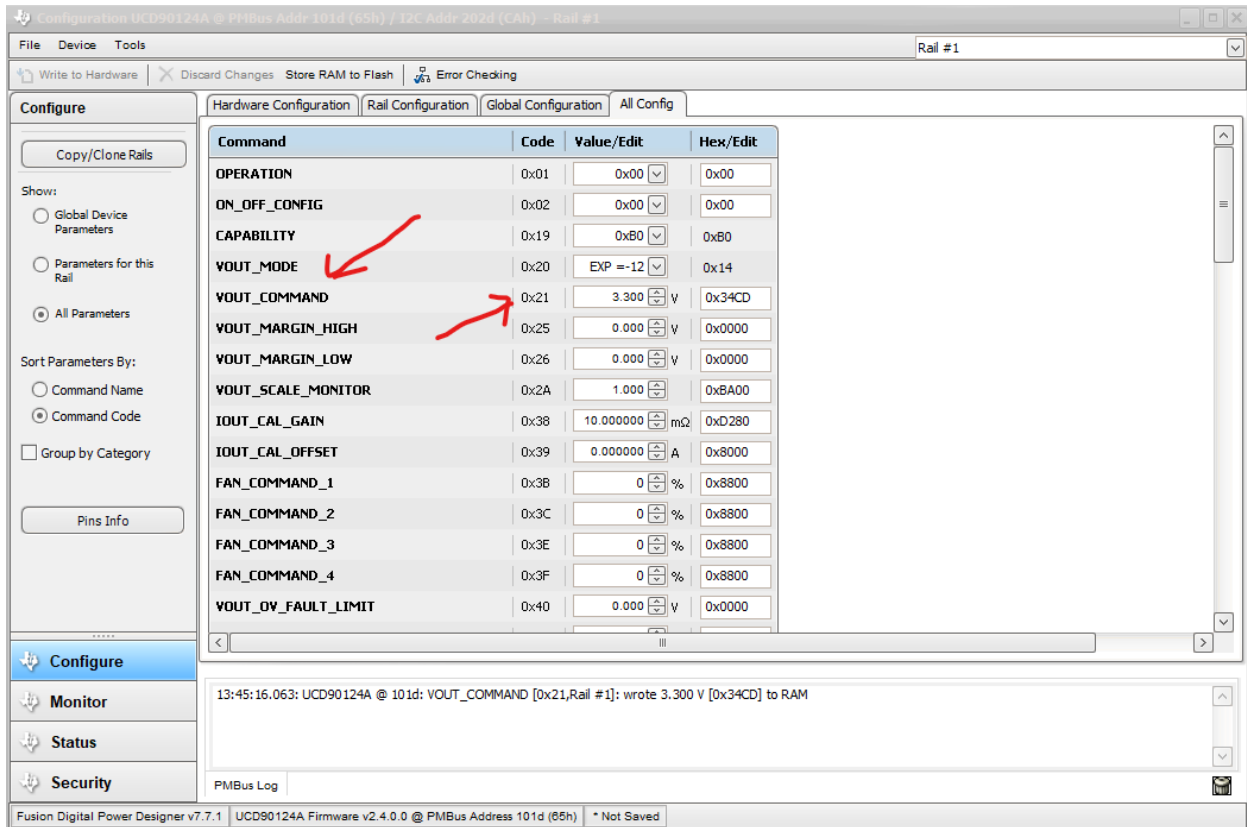
Type: Flag (true | false)
Default = false


```
C:\Anne_TI\Support\Ciena\April-1-2014>FusionParamReader --address 126 0x8B
CmdIDWithPhase   PageHex SAAStatus        Decoded DecodedNumeric  EncodedHex
READ_VOUT        0x00    ACK      0.000 V 0        0x0000
READ_VOUT        0x01    ACK      0.000 V 0        0x0000
READ_VOUT        0x02    ACK      0.000 V 0        0x0000
READ_VOUT        0x03    ACK      0.056 V 0.056    0x00E6

C:\Anne_TI\Support\Ciena\April-1-2014>
```

--cmd-id

Description: Specify the parameter/command to read.

Type: Input. Value is either the command code in hexadecimal format, or command name which can be found on Fusion GUI advanced configuration screen (below screenshot), for example, VOUT_COMMAND, IOUT_CAL_GAIN_1, etc.

Examples

Example 1: redirect result to a file.

Read non-paged command 0xFD at the first device found, using scan-mode = device-id;, redirect the result to outfile.txt

FusionParamReader *--scan --scan-mode device_id 0xFD >> outfile.txtl*

Example 2: read at specific device's address

Read command 0x8B (READ_VOUT) at device address 126d. Since 0x8B is paged-command and –rail flag is omited => return the all rails' values to screen, with header.

FusionParamReader *--address 126 0x8B*

Example 3: No header vs with header

With header:

FusionParamReader *--address 126 0xFD >> outfile.txt*

Without header:

FusionParamReader *--address 126 0xFD --no-header* >> *outfile.txt*

FusionParamWriter:

See Common Flags for more detail: --help, --address, --scan, --scan-mode, --device-exact-version

Flags Option

--rail

Description: Specify the rail number for the command. Numbering starts at 1.

- If you specify --rail for a command that is not "PAGEd", the program will exit will an error.
- If you do not specify --rail and a command is PAGEd
    - If device has more than one rail configured, the program will exit will an error.
    - If device has only one rail configured, the program will write the value for rail #1

Type: Input as integer. Value starts at 1.

--cmd-id

Description: Specify the parameter/command to write.

Type: Input. Value is either the command code in hexadecimal format, or command name which can be found on Fusion GUI advanced configuration screen (below screenshot), for example, VOUT_COMMAND, or IOUT_CAL_GAIN_1, etc.,
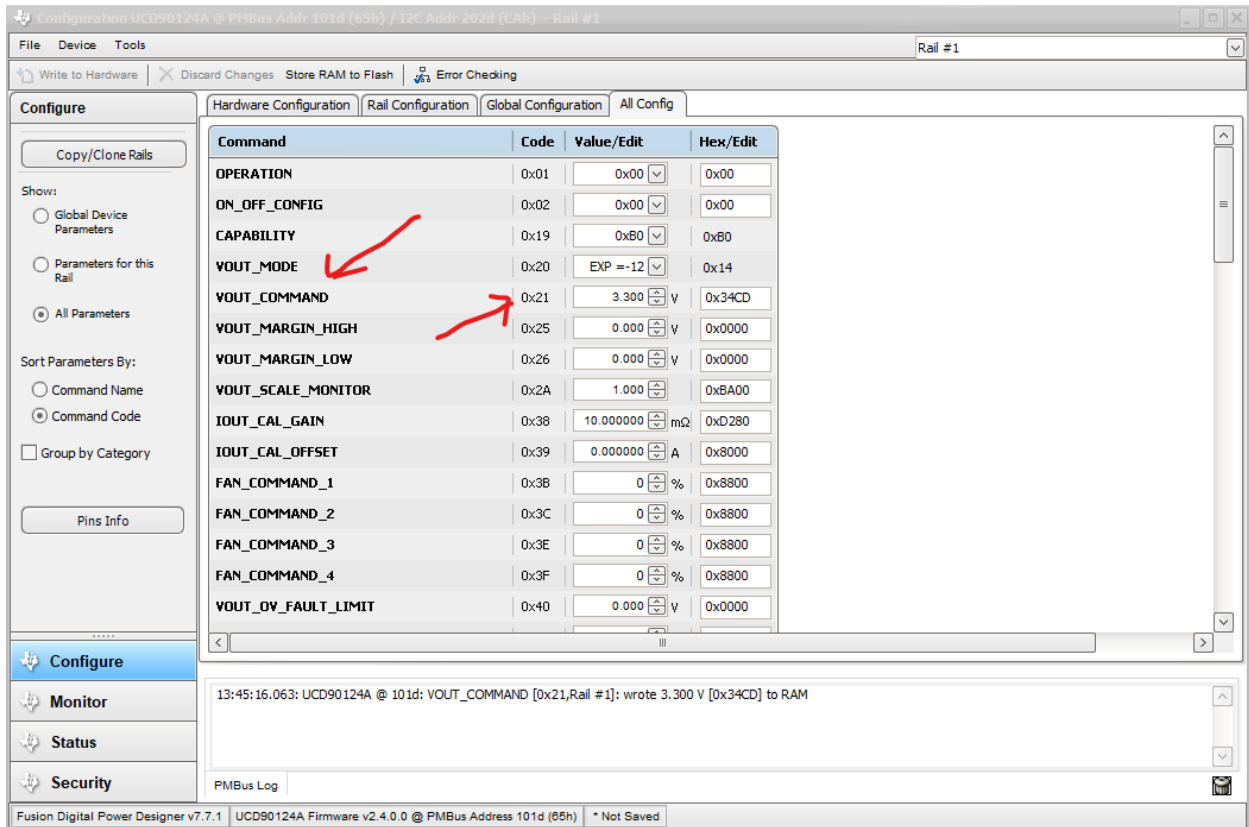
--value-encoded

Description: Specify the value to write to the command in device-level terms: the hexadecimal value to write.

Type: Input required. Value is in hexadecimal format. The hex can either be in the format of 0xAABB, AABB, or Aabbh. Case does not matter.You can find the encoded value using Fusion Tools in the advanced configuration screen (aka All_Configuration). For SendByte comman, neither value-encoded, nor value-decoded needed.
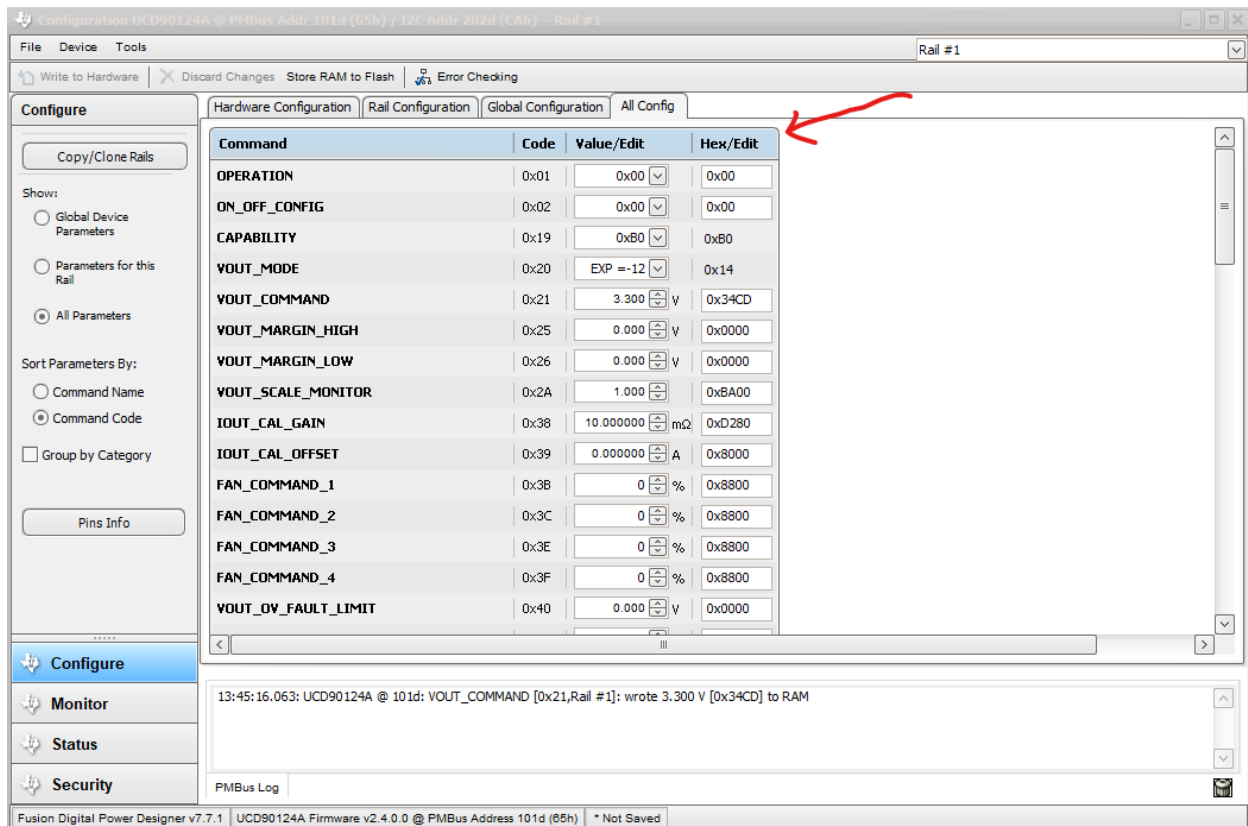
--value-decoded

<u>Description</u>: Specify the value to write to the command in high-level, decoded terms.

<u>Type</u>: Input required. Depends on which command to write, the decoded-value can be an integer, a decimal number, or a string You can find the decoded value using Fusion Tools in the advanced configuration screen (aka All_Configuration). This mode can only be used with commands that represent floating point numbers (VOUT_COMMAND, IOUT_CAL_GAIN, FAN_COMMAND_1), integers (CLF_EVENTS, EADC_TRIM), or strings (MFR_DATE, MFR_REVISION, USER_DATA_00)."
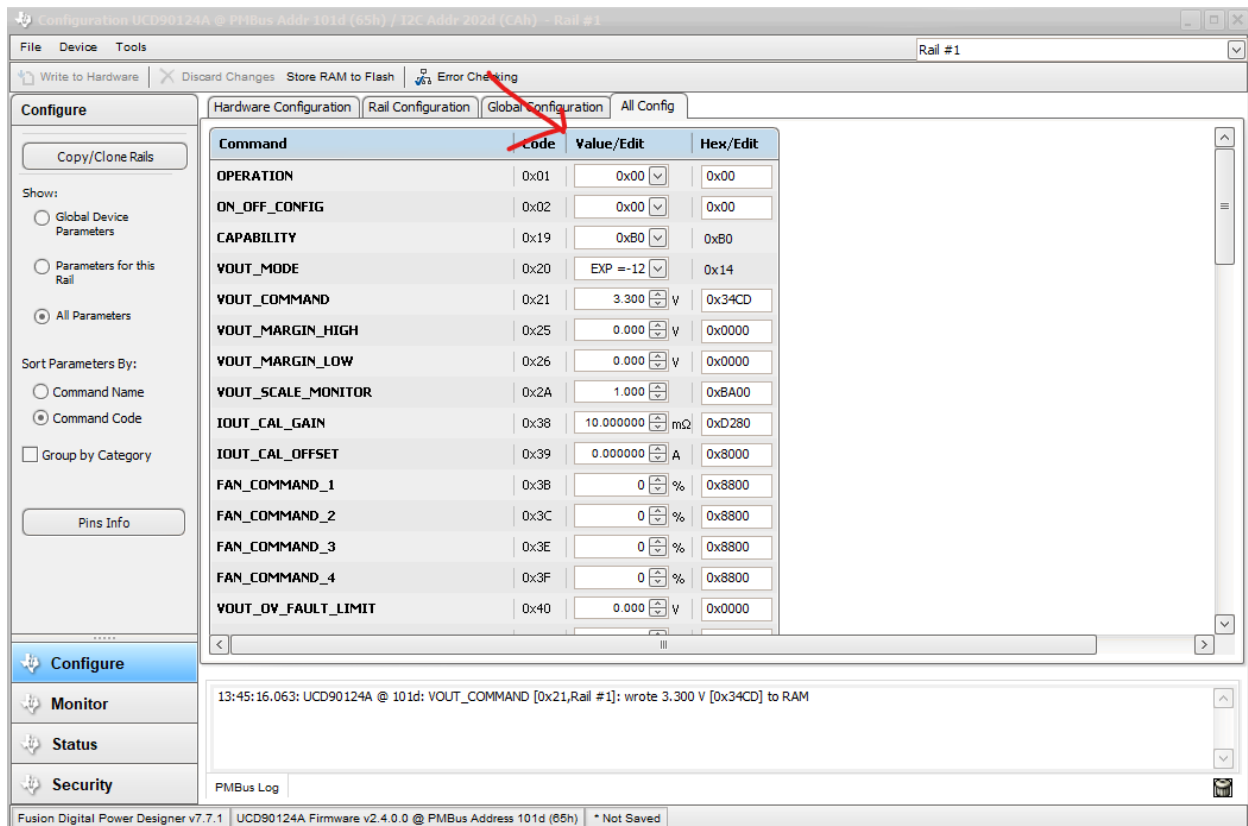
--quiet

Description: Normally, progress/status information about the selected task is displayed on standard output. To disable this and only print error messages (to standard error), pass this flag. Note that, regardless of the setting of this option, FusionParamWriter will set a 0 exit code on success and 1 on failure.

Examples

Example 1: SendByte command with no data

Write to device at address 92d to clear fauts CLEAR_FAULTS (0x03), clear loged faults LOGGED_FAULTS (0xEA), and clear logged peak LOGGED_PEAKS (0xE9)

FusionParamWriter *--address 92 --cmd-id 0x03*

FusionParamWriter *--address 92 --cmd-id CLEAR_FAULTS*

FusionParamWriter *--address 92 --cmd-id 0xE9*

FusionParamWriter *--address 92 --cmd-id-0xEA*

Example 2: Write paged-command

Write paged-command VOUT_COMMAND (code = 0x21) to the first found device using device-id scan-mode (0xFD command)

FusionParamWriter *--scan --scan-mode device_id --cmd-id VOUT_COMMAND --rail 3   --value-encoded 0x34CD*

FusionParamWriter --scan --scan-mode device_id --cmd-id VOUT_COMMAND --rail 3 --value-decoded 3.3

FusionParamWriter --scan --scan-mode device_id --cmd-id 0x21 --rail 3 --value-decoded 3.3

Example 3: Write non-paged command
Write non-paged command MFR_MODEL to first found device using combo scan-mode (0xFD, then 0xAD, then 0xFC). The -scan-mode flag is omited => use default scan-mode which is combo.

FusionParamWriter --scan --cmd-id MFR_MODEL --value-decoded "MFR_MODEL"

FusionParamWriter --scan --cmd-id MFR_MODEL --value-encoded 0x4D46525F4D4F44454C

FusionParamWriter --scan --cmd-id 0x9A --value-encoded 0x4D46525F4D4F44454C

FusionFirmwareDownload

Download firmware image file to device in ROM mode. If the image file includes the data flash, users will have option to download also the data flash portion to device in ROM mode.

NOTE:

- In GUI earlier than 7.8.1:
    - Users have to enter both –pflash and –dflash flags
    - Users have to enter –pflash-checksum
    - UCD90320x device can only accept firmware file with extension ".bin"
- In GUI 7.8.1 or later
    - If –pflash is missing, it is set to "skip"
    - If –dflash is missing, it is set to "skip"
    - If –pflash-checksum is missing, it is set to:
        - If –pflash is set to "skip" => set to "none"
        - If –pflash is set to "erase" => set to "none"
        - If –pflash is set to "download" => set to "calc"
    - UCD90320x device can accept firmware file with extension ".bin", ".hex", ".srec", and ".x0"

Flags Options

--state
Description: Specify what state the device should be in when the tool is launched. If program or auto, you can specify the program mode address to scan via --address. Or, if you do not specify --address, all valid PMBus addresses will be scanned. The default mode is auto.

Type: Input. Values = rom | program | auto

--infile
Description: The firmware image file. This can be in x0, Intel Hex, S-Record, or bin format.

Type: Input as string.

--pflash

Description: What to do with the program flash section of flash/file. Note that if you specify download but the input file does not contain pflash data, the net result will be erase, as the program flash will always be erased first.

NOTE: For UCD90320x: "erase" is not a valid option. Set to either "download" or "skip".

NOTE: For GUI 7.8.1 or later, this flag can be omited.

Type: Input. Values = download | erase | skip

--dflash

Description: What to do with the data flash section of flash/file. Note that if you specify download but the input file does not contain data flash data, the net result will be erase, as the data flash will always be erased first.

NOTE: For UCD90320x device, the only option is valid is "skip". If –dflash is set to either "download" or "erase", the flag is ignore.

Type: Input. Values = download | erase | skip

--pflash-checksum

Description: Program flash checksum mode. In calc mode, the checksum will be calculated based on the source flash image. In none mode, a 0xFFFFFF checksum will be used. In source mode, whatever checksum was present in the original flash input file will be passed through. If calc is set and there is bootflash present then the pflash checksum will exclude the boot.

NOTE: For UCD90320x, the –pflash-checksum flag is ignore.

Type: Input. Values = calc | none | source

--no-trim-ok

Description: Normally the firmware download is aborted if the target device has not been trimmed at the factory. The phase "device did not pass factory trim check" will be printed to the console and the program will exit with a status of 1. You can chose to make this non-fatal and continue with programming by passing this option. A warning will be printed to the console in this case.

NOTE: For UCD90320x: this option is ignored.

Type: Flag (true | false)
Default = false

--execute-program

Description: If specified, the tool will execute program flash when all tasks have been completed. The default is not to execute the program.

NOTE: For UCD90320x: this option is ignore. After firmware download, device is already in Program Mode without the need of sending device to its program.

Type: Flag (true | false)
Default = false

Description: If specified, the tool will scan for the device after instructing the ROM to execute the program (--execute-program). If the device was previously in program mode, the same address will be assumed and the tool will execute with an error condition if a device is not present on that address. If the device was in ROM mode when the tool was launched, a scan of all valid addresses will be performed. If one device is not found, the tool will exit with an error condition. The default is not to rescan.

Type: Flag (true | false)
Default = false

--rescan-delay

Description: When --rescan is specified, the tool will pause to give the firmware time to startup. This lets you specify how long this delay is, in milliseconds. The default is 750 milliseconds.

Type: Input as integer.

Examples

Example 1:
Download UCD90124A firmware to device. GUI will auto-detect device either in ROM or Program mode. Program flash checksum is calculated based on the source. Since –execute-program flag is missing, device stays at ROM mode after finish. Return failure if device has not been trimmed at factory.

FusionFirmwareDownload *--address 101 --infile* C:\Projects\UCD90124A_2_4_0_111121.x0 *--pflash* download *--dflash* skip *–pflash-checksum* calc

Example 2
Download UCD90124A firmware to device. GUI will auto-detect device either in ROM or Program mode. Program flash is skipped, only download the data flash. After complete downloading firmware, device will execute its program. If device has not been trimmed at factory, continue with downloading, display warning at the end.

FusionFirmwareDownload *--address 101 --infile* C:\Projects\ UCD90124A-flash.hex *--pflash* skip *--dflash* download *–no-trim-ok –execute-program*

Example 3:
Download UCD90320U firmware to device. GUI will auto-detect device either in ROM or Program mode. Program flash checksum is calculated based on the source. After complete downloading firmware, device will be in the Program Mode.

FusionFirmwareDownload *--address 19 --infile* C:\Projects\UCD90320U.bin *--pflash* download

```
Scan for adapter ...Found adapter!
Scan for device in ROM mode ...no ROM found
Scan for device in Program mode ...
Found UCD90320 Firmware v v3.0.0.3029 @ PMBus Address 19d, sending it to ROM mod
e ...
Parsing firmware file ...
Firmware is in .bin format
Starting download ...
...............................................................................
...............................................................................
...............................................................................
...............................................................................
...............................................................................
...............................................................................
```

Example 4

Download UCD90320U firmware to device. GUI will detect device either in Program mode only. Program flash checksum is calculated based on the source. After complete downloading firmware, device will be in Program mode. Rescan for device at address 19d after finish downloading firmware

FusionFirmwareDownload *--address 19* *--state program* *--infile* C:\Projects\UCD90320U.hex *--pflash* download *--rescan*