

NAME

decutil – Diligent Eclipse Utility

SYNOPSIS

decutil [--help] [--version] *command* [options]

DESCRIPTION

decutil provides a command line interface for discovering information about the features and configuration of an Eclipse platform board. Commands are also provided for discovering the Zmods (SYZYGY pods) attached to the SmartVIO port(s) of the board and for manipulating various Platform MCU settings.

COMMANDS**getinfo**

Get general configuration and information about the supported features of the Platform MCU (PMCU). This command communicates with the PMCU over the I2C bus to retrieve general information about the capabilities of the PMCU and the board configuration. This information includes the PMCU firmware revision, SmartVIO port count, power supply group counts (5V0, 3V3, VADJ), the number of temperature probes supported by the board, and the number of fans supported by the board. If the board supports one or more temperature probe, then the capabilities of each supported probe and the most recent temperature measurement of that probe are displayed via the console. If the board supports one or more fan, then the capabilities of each supported fan are displayed and if a fan supports RPM measurement then the most recent RPM measurement is also displayed.

getinfo5v0

Get information about the on board 5V0 power supplies that are associated with the on board SmartVIO ports. This function communicates with the Platform MCU (PMCU) over I2C to determine the number of on board 5V0 power supplies, to retrieve the amount of current that each supply is capable of providing, and to retrieve the sum of current requested by all SmartVIO ports that are associated with each supply.

The "-chanid" option can be used to specify the channel identifier (0-7, a-f, or A-F) of a specific power supply in order to limit the information that is displayed to a specific channel. If no channel identifier is provided, then this command will retrieve and display information for every channel supported by the board.

getinfo3v3

Get information about the on board 3V3 power supplies that are associated with the on board SmartVIO ports. This function communicates with the Platform MCU (PMCU) over I2C to determine the number of on board 3V3 power supplies, to retrieve the amount of current that each supply is capable of providing, and to retrieve the sum of current requested by all SmartVIO ports that are associated with each supply.

The "-chanid" option can be used to specify the channel identifier (0-7, a-f, or A-F) of a specific power supply in order to limit the information that is displayed to a specific channel. If no channel identifier is provided, then this command will retrieve and display information for every channel supported by the board.

getinfovio

Get information about the on board VIO (VADJ) power supplies that are associated with the on board SmartVIO ports. This function communicates with the Platform MCU (PMCU) over I2C to determine the number of on board VIO power supplies, to retrieve the amount of current that each supply is capable of providing, to retrieve the sum of current requested by all SmartVIO ports that are associated with each supply, and to retrieve

all status and configuration information associated with each supply. The "-chanid" option can be used to specify the channel identifier (0-7, a-f, or A-F) of a specific power supply in order to limit the information that is displayed to a specific channel. If no channel identifier is provided, then this command will retrieve and display information for every channel supported by the board.

getinfopower Get information about the on board power supplies (5V0, 3V3, VIO) that are associated with the on board SmartVIO ports. This command communicates with the Platform MCU (PMCU) over I2C to determine the number of on board 5V0, 3V3, and VIO power supplies that are associated with the on board VIO ports and to retrieve various information about each of these supplies. Executing this command is equivalent to sequentially executing "getinfo5v0", "getinfo3v3", and "getinfovio".

enum Enumerate SmartVIO ports. This command communicates with the Platform MCU over the I2C bus to determine how many SmartVIO ports the board contains and to retrieve the configuration and status for each of those ports. If a SmartVIO port has a SYZYGY pod installed then the I2C bus is used to retrieve the Standard SYZYGY firmware registers, as well as the SYZYGY DNA (including all string fields), and output that information to the console. Additional information may be displayed when a Digilent Zmod is attached to the port.

setplatcfg Modify one or more field of the Platform MCU (PMCU) Platform Configuration Register. This function uses the I2C bus to retrieve the contents of the PMCU's Platform Configuration Register, modifies the specified field(s) of the register, and then writes the new settings to the register. Settings that may be modified include enforcing the 5V0 current limit, enforcing the 3V3 current limit, enforcing the VIO current limit, and performing CRC checks of SYZYGY headers. Please note that the Platform Configuration is stored in the PMCU's EEPROM and is only read during firmware initialization. Therefore, any changes made to the Platform Configuration Register will not take effect until the next time the PMCU is reset. The PMCU may be reset by issuing the "resetpmcu" command.

The "-enforce5v0" option can be used to enable or disable enforcement of 5V0 current limits. If enforcement is enabled and the sum of the current requested by all associated SmartVIO pods exceeds the total current that the onboard supply can provide then the VIO supply associated with the SmartVIO port will not be enabled.

The "-enforce3v3" option can be used to enable or disable enforcement of 3V3 current limits. If enforcement is enabled and the sum of the current requested by all associated SmartVIO pods exceeds the total current that the supply can provide then the VIO supply associated with the SmartVIO port will not be enabled.

The "-enforcevio" option can be used to enable or disable enforcement of VIO current limits. If enforcement is enabled and the sum of the current requested by all associated SmartVIO pods exceeds the total current that the supply can provide then the VIO supply associated with the SmartVIO port will not be enabled.

The "-checkcrc" option can be used to enable or disable SYZYGY header DNA checks. If CRC checks are enabled and the CRC computed does not match, then the VIO supply associated with the SmartVIO port will not be enabled.

setviocfg Modify one or more field of the Platform MCU (PMCU) VADJ_n_OVERRIDE register. The VADJ_n_OVERRIDE register can be used to override the state of a specific VIO supply. This includes enabling or disabling the supply, as well as setting the output

voltage. When a `VADJ_n_OVERRIDE` register is written the PMCU will check to make sure that the specified settings do not conflict with the requirements of any SmartVIO port associated with the specified supply. If there aren't any conflicts, then the specified settings will be applied immediately. However, if there is a conflict then the changes to the `VADJ_n_OVERRIDE` register, and the associated power supply, will be restricted to meet the requirements of all associated SmartVIO ports.

The `-chanid` option must be used to specify the channel identifier (0-7, a-f, or A-F) of the VIO supply.

The `-override` option can be used to enable ('y') or disable ('n') overriding the VIO supply configuration. If override is enabled, then the associated VIO supply will be configured based on the enable and voltage fields of the `VADJ_n_OVERRIDE` register.

The `-enable` option can be used to enable ('y') or disable ('n') the associated VIO supply. This setting has no impact when the override field of the `VADJ_n_OVERRIDE` register is cleared.

The `-voltage` option can be used to specify the voltage (in millivolts) of the associated VIO supply. This setting has no impact when the override field of the `VADJ_n_OVERRIDE` register is cleared.

setfancfg

Modify one or more field of the Platform MCU (PMCU) `FAN_n_CONFIGURATION` register. The `FAN_n_CONFIGURATION` register is used to specify the settings of the associated fan. This may include the enable state of the fan, the fan's speed, and the associated temperature probe. Please note that not all fan ports support enable/disable, fixed speed control, or automatic speed control (temperature based). Changes to a `FAN_n_CONFIGURATION` register will be restricted to be within the supported capabilities of the port and take effect immediately after the register is written. Additionally, the FAN configuration is written to EEPROM and will be restored each time the PMCU is reset or power cycled.

The `-fanid` option must be used to specify the identifier (1-4) of the fan configuration to be modified.

The `-enable` option can be used to enable ('y') or disable ('n') the associated fan.

The `-speed` option can be used to specify the speed ("minimum", "medium", "maximum", or "auto") of the associated fan. Please note that not all fans support this functionality and some ports that do support this functionality may not support automatic fan speed control.

The `-probe` option can be used to specify the temperature probe ("none", "p1", "p2", "p3", "p4") associated with a fan if that fan supports automatic speed control.

resetpmcu

This command uses the I2C bus to write a positive value to the software reset register of the Platform MCU (PMCU), which causes the processor to perform a software reset.

writedna

Write the specified binary file to the DNA flash section of the peripheral MCU (pMCU) attached to the specified SYZYGY port. The entire binary file is written to the DNA section of the pMCU flash memory starting at address 0x8000 (as specified in the SYZYGY specification). If the pMCU firmware utilizes a magic number to prevent accidental flash

writes, then that magic number should be specified using the "-magic" option so that it gets sent to the pMCU before any attempts are made to write flash memory. If a magic number was specified, then "0x00" will be sent to the magic address after the flash write operation has completed.

The "-port" option must be used to specify the physical SYZYGY port (a-z,A-Z,or 0-25) that the pMCU is attached.

The "-dnafile" option must be used to specify the pathname to a binary file containing data to write to the pMCU's flash memory.

The "-magic" option may be used to specify a hexadecimal magic number for enabling flash writes on devices that support write protection.

OPTIONS

- chanid** Specify the channel identifier to get information about a specific power supply when executing one of the getinfo commands or to specify the VIO channel that's being configured by the "setviocfg" command. The channel identifier may be specified using the numbers 0 through 7, the characterse 'a' through 'h', or the characters 'A' through 'H'. The number of VIO supplies, or channels, may vary from product to product. Use the "getinfo" command to determine how many 5V0, 3V3, and VIO channels the device supports.
- fanid** Specify the fan identifier used when setting a fan configuration with the "setfanid". The fan identifier may be specified using the numbers 1 through 4. The number of fans support may vary from product to product. Use the "getinfo" command to determine how many fans the device supports.
- port** Specify the port identifier for the physical port containing the pod whose DNA you wish to write when executing the "writedna" command. The port identifier command may be speified using numbers 0 through 25, the characterse 'a' through 'z', or the characters 'A' through 'Z'. The number of physical ports supported may vary from product to product. Use the "getinfo" or "enum" commands to determine how many SmartVIO ports the device supports.
- enable** Specify whether a function is enabled ('y') or disabled ('n').
- override** Specify whether the configuration of a VADJ supply is overridden ('y') or not ('n') by an alternative configuration using the "setviocfg" command.
- voltage** Specify the VIO voltage (in millivolts) to set in the VADJ_n_OVERRIDE register when executing the "setviocfg" command.
- enforce5v0** Specify if the 5V0 current limit is enforced ('y') or ignored ('n'). This setting is written to the Platform MCU's Platform Configuration Register by issuing the "setplatcfg" command.
- enforce3v3** Specify if the 3V3 current limit is enforced ('y') or ignored ('n'). This setting is written to the Platform MCU's Platform Configuration Register by issuing the "setplatcfg" command.

- enforcevio** Specify if the VIO current limit is enforced ('y') or ignored ('n'). This setting is written to the Platform MCU's Platform Configuration Register by issuing the "setplatcfg" command.
- checkerc** Specify if a CRC check is performed ('y') or skipped ('n') for each SYZYGY header. This option must be issued alongside the "setplatcfg" command to specify if the Platform MCU performs CRC checks when it enumerates SmartVIO ports. This option may also be used to tell the "enum" command to skip CRC checks when enumerating SmartVIO ports.
- speed** Specify the fan speed setting used when the "setfancfg" command is issued. The fan speed may be specified as "minimum", "medium", "maximum", or "auto". Not all fan headers support setting a fixed speed or automatic speed control. Use the "getinfo" command to determine how many fan headers the device contains and if a fixed speed or automatic speed setting is supported for each header.
- probe** Specify the temperature source, or probe, utilized when performing automatic speed control. Temperature probes may be specified as "none", "p1", "p2", "p3", or "p4". Use the "getinfo" command to determine how many temperature probe's the device supports.
- dnafile** Specify the pathname of the DNA binary file, or blob, to be written to the DNA flash section of the peripheral MCU. Flash is written two bytes at a time and therefore the binary file should contain an even number of bytes. If the file does not contain an even number of bytes then the image will be padded with 0xFF.
- magic** Specify an 8-bit hexadecimal magic number used to disable flash write protection when issuing the "writedna" command.
- verify** Perform readback verification of data.
- ,--help** Display typical application usage, a list of supported commands, and a list of supported options.
- v,--version** Display information about the program version.
- verbose** Display more detailed error messages.

EXAMPLES

\$ decutil getinfo

Display information about the Eclipse platform configuration.

\$ decutil enum

Display information about the on board SmartVIO ports and any attached SYZYGY devices.

\$ decutil getinfovio

Display information about the configuration of all on board VIO supplies.

\$ decutil getinfovio -chanid a

Display information about the configuration of VIO supply A (VADJA).

\$ decutil getinfopower

Display information about the configuration of all on board 5V0, 3V3, and VIO supplies that are associated with one or more SmartVIO ports.

\$ decutil setplatcfg -enforcevio n

Disable enforcement of VIO current limits. This will allow a VIO supply to be enabled even if the sum of the current requested by all SmartVIO ports associated with the supply exceeds the limit set for that supply. Please note that you must issue the "resetpmcu" command or power cycle your board before this setting will have any impact.

\$ decutil setplatcfg -checkcrc n

Skip CRC checks when retrieving SYZYGY headers from SmartVIO ports during enumeration. Please note that you must issue the "resetpmcu" command or power cycle your board before this setting will have any impact.

\$ decutil setviocfg -chanid a -enable y -override y -voltage 3300

Set the voltage of VIO Supply Group A (VADJA) to 3.3 volts. If all pods attached to the ports associated with this supply group are compatible with 3.3 volts, then the voltage will immediately be set to 3.3 volts. If one or more pod attached to a port associated with this supply group is incompatible with 3.3 volts then the voltage will be set to the minimum or maximum safe voltage for all pods.

\$ decutil setviocfg -chanid a -enable n -override y

Disable VIO Supply Group A (VADJA). This will result in the VIO supply associated with Group A immediately powering down.

\$ decutil setviocfg -chanid a -override n

Stop overriding VIO Supply Group A's (VADJA) configuration. This will immediately result in VIO group A's configuration being return to the state that was determined when the Platform MCU (PMCU) came out of reset. If the supply was enabled and set to 1.8 volts then the supply will be enabled with an output voltage of 1.8 volts. If the supply was disabled coming out of PMCU reset, then it will be disabled.

\$ decutil setfancfg -fanid 1 -speed minimum

Configure Fan 1 to operate at the minimum allowed RPM. This setting will be applied immediately, and the fan will operate a minimum speed, even after a power on reset.

\$ decutil setfancfg -fanid 1 -speed auto

Configure Fan 1 to utilize automatic speed control. This setting will be applied immediately, and fan RPM will be controlled based on the temperature measured by the selected temperature probe.

\$ decutil resetpmcu

Reset the Platform MCU (PMCU). The PMCU will immediately perform a software reset. This will result in the Platform Configuration and Fan Configuration being read from the EEPROM section of the flash, all SmartVIO ports being re-enumerated, and any associated Power Management Units being reconfigured.

AUTHOR

Digilent, Inc. <support@digilentinc.com>