

NAME

dppl – Digital Phase Locked Loop (DPLL) subsystem management

SYNOPSIS

dppl [*OPTIONS*] **device** { *COMMAND* | **help** }

dppl [*OPTIONS*] **pin** { *COMMAND* | **help** }

dppl [*OPTIONS*] **monitor**

OPTIONS := { **-V**[*ersion*] | **-j**[*son*] | **-p**[*retty*] }

DESCRIPTION

The **dppl** utility is used to configure and monitor Digital Phase Locked Loop (DPLL) devices and pins. DPLLs are used for clock synchronization in various hardware, particularly in telecommunications and networking equipment.

A DPLL device can lock to one or more input pins and provide synchronized output. Pins can be physical external signals (like GNSS 1PPS, SyncE), or internal oscillators.

OPTIONS

-V, -Version

Print the version of the **dppl** utility and exit.

-j, -json

Output results in JavaScript Object Notation (JSON).

-p, -pretty

When combined with **-j**, generates a pretty JSON output with indentation and newlines for better human readability.

DEVICE COMMANDS

dppl device show [*id ID*]

Display information about DPLL devices. If no arguments are specified, shows all devices in the system.

id ID Show only the device with the specified numeric identifier.

Output includes:

- Device ID
- Module name providing the device
- Clock ID (unique identifier)
- Operating mode (manual, automatic, holdover, freerun)
- Lock status (locked-ho-ack, locked, unlocked, holdover)
- Temperature (if supported)
- Type (PPS or EEC)

dpll device set id ID [mode { automatic | manual }] [phase-offset-monitor { enable | disable }] [phase-offset-avg-factor FACTOR]

Configure DPLL device parameters.

id ID Specifies which device to configure (required).

mode { *automatic* | *manual* }

Set the operating mode of the DPLL device. In **automatic** mode, the DPLL autonomously selects the best available input source based on priority and quality. In **manual** mode, the input source must be explicitly configured and the DPLL will not automatically switch sources.

phase-offset-monitor { *enable* | *disable* | *true* | *false* | *0* | *1* }

Enable or disable phase offset monitoring between the device and its pins. When enabled, the kernel continuously measures and reports phase differences.

phase-offset-avg-factor *FACTOR*

Set the averaging factor (1-255) applied to phase offset calculations. Higher values provide smoother but slower-responding measurements.

dpll device id-get [module-name NAME] [clock-id ID] [type TYPE]

Retrieve the device ID based on identifying attributes. Useful for scripting when you need to find a device's numeric ID. At least one attribute should be specified to identify the device.

module-name *NAME*

Kernel module name.

clock-id *ID*

64-bit clock identifier in decimal or hex (0x prefix).

type *TYPE*

Device type: **pps** or **eec**.

PIN COMMANDS

dpll pin show [id ID] [device ID]

Display information about DPLL pins. If no arguments are specified, shows all pins in the system.

id ID Show only the pin with the specified numeric identifier.

device ID

Show only pins associated with the specified device ID.

Output includes:

- Pin ID
- Module name
- Clock ID
- Board label (hardware label from device tree or ACPI)

- Pin type (mux, ext, synce-eth-port, int-oscillator, gnss)
- Frequency and supported frequency ranges
- Capabilities (state-can-change, priority-can-change, direction-can-change)
- Phase adjustment range, granularity, and current value
- Parent device relationships (direction, priority, state, phase offset)
- Parent pin relationships
- Reference sync information
- Esync frequency support (if applicable)

dpll pin set id ID [PARAMETER VALUE] ...

Configure DPLL pin parameters. Multiple parameters can be specified in a single command.

id ID Specifies which pin to configure (required).

frequency *FREQ*

Set pin frequency in Hz. The pin must support the specified frequency (check frequency-supported ranges in pin show output).

phase-adjust *ADJUSTMENT*

Set phase adjustment in picoseconds. This value fine-tunes the phase of the output signal. Negative values shift the phase backwards, positive values shift it forwards. The value must be within the phase-adjust-min and phase-adjust-max range.

esync-frequency *FREQUENCY*

Set enhanced SyncE (Synchronous Ethernet) frequency in Hz for capable pins.

parent-device *DEVICE_ID* [*direction DIR*] [*prio PRIO*] [*state STATE*]

Configure the relationship between this pin and a parent DPLL device.

direction { *input* | *output* }

Set the pin's direction relative to the parent device.

prio *PRIORITY*

Set priority (0-255) for this pin on the parent device.

state { *connected* | *disconnected* | *selectable* }

Set the pin's state on the parent device.

parent-pin *PIN_ID* [*state STATE*]

Configure the relationship to a parent pin.

reference-sync *PIN_ID* [*state STATE*]

Configure reference sync relationship with another pin.

dpll pin id-get [SELECTOR] ...

Retrieve a pin ID based on identifying attributes.

module-name *NAME*

Filter by kernel module name.

clock-id *ID*

Filter by 64-bit clock identifier.

board-label *LABEL*

Filter by board label (hardware identifier).

panel-label *LABEL*

Filter by panel label.

package-label *LABEL*

Filter by package label.

type *TYPE*

Filter by pin type: **mux**, **ext**, **synce-eth-port**, **int-oscillator**, **gnss**.

MONITOR COMMAND

dppl monitor

Monitor DPLL subsystem events in real-time. Displays notifications about:

- Device creation, deletion, and configuration changes
- Pin creation, deletion, and configuration changes
- Lock status changes
- Phase offset updates
- Frequency changes

Events are prefixed with their type: [DEVICE_CREATE], [DEVICE_CHANGE], [DEVICE_DELETE], [PIN_CREATE], [PIN_CHANGE], [PIN_DELETE].

Press Ctrl+C to stop monitoring.

EXAMPLES

Show all DPLL devices

dppl device show

Show specific device in JSON format

dppl -j device show id 0

Set device 0 to manual mode

dppl device set id 0 mode manual

Enable phase offset monitoring on device 0

dppl device set id 0 phase-offset-monitor enable

Show all pins
dppl pin show

Show pin with pretty JSON output
dppl -jp pin show id 5

Set pin frequency to 10 MHz
dppl pin set id 0 frequency 10000000

Configure pin relationship to parent device
dppl pin set id 1 parent-device 0 prio 10 direction input state connected

Adjust phase by -1000 picoseconds
dppl pin set id 2 phase-adjust -1000

Set multiple pin parameters at once
dppl pin set id 3 frequency 10000000 phase-adjust -1000

Monitor DPLL events
dppl monitor

Monitor events in JSON format
dppl -jp monitor

Get device ID by module name
dppl device id-get module-name ice

Get pin ID by board label
dppl pin id-get board-label GNSS-1PPS

PHASE ADJUSTMENT

Phase adjustment is specified in picoseconds (1e-12 seconds) and allows fine-tuning of signal phase. This is crucial for precise time synchronization applications like 5G networks and high-frequency trading.

Important considerations:

- Check phase-adjust-min and phase-adjust-max before setting values
- Some hardware requires values to be multiples of phase-adjust-gran
- Negative values shift phase backwards (earlier in time)
- Positive values shift phase forwards (later in time)
- The kernel validates all phase adjustment requests

CAPABILITIES

Pins may have various capabilities that determine which operations are allowed:

state-can-change

The pin's state (connected/disconnected/selectable) can be modified.

priority-can-change

The pin's priority can be modified. This may apply to top-level priority or priority within parent-device relationships.

direction-can-change

The pin's direction (input/output) can be modified.

Use **dppl pin show** to check which capabilities a pin supports before attempting configuration changes.

EXIT STATUS

0	Success
1	General failure
2	Invalid arguments or usage
255	Netlink communication error

NOTES

- The DPLL subsystem requires kernel support (CONFIG_DPLL=y or m)
- Hardware support varies by device and driver
- Some operations require specific hardware capabilities
- Phase offset values are measured by hardware and cannot be set directly
- Changes to device/pin configuration may affect system clock synchronization

SEE ALSO

ip(8), **devlink(8)**, **ethtool(8)**

Linux kernel documentation: *Documentation/driver-api/dpll.rst*

Netlink specification: *Documentation/netlink/specs/dpll.yaml*

AUTHOR

dppl was written by Arkadiusz Kubalewski, Vadim Fedorenko, and others.

This manual page was written by Petr Oros.

REPORTING BUGS

Report bugs to <netdev@vger.kernel.org>