The Linux Kernel DPLL Subsystem

Hugo Ongetta-Dagnoni

Introduction to the API

- Provides a general interface to configure devices that use DPLLs (Digital Phase Locked Loops)
- Utilises an underlying Netlink protocol to receive & request changes to a DPLL's configuration information
- An alternative to the sysfs interface

History & Development

- First introduced in version 6.7 and continuously updated to provide more capabilities
  - Available at kernel.org & on RHEL 9.4, CentOS Stream 9, OCP 4.14
- Further patches are expected by end of year
- It's strongly suggested to use latest available kernel release

How does it work?

- Uses a general purpose YNL utility to encode & decode netlink messages
- Bases these messages on a DPLL YAML specification passed in as a parameter
- Can then pass in request to dump/edit info with --dump & --do flags

```
python3 ./tools/net/ynl/cli.py --spec Documentation/netlink/specs/dpll.yaml
```

Boilerplate for using API

--dump device-get

Retrieve information on all DPLL devices with netlink command device-get & --dump flag:

./tools/net/ynl/cli.py --spec Documentation/netlink/specs/dpll.yaml --dump device-get

```
[root@localhost linux-6.11.7]# ./tools/net/ynl/cli.py --spec Documentation/netlink/specs/dpll.yaml --dump device-get
[{'clock-id': 5799633565435105048,
  'id': 2,
  'lock-status': 'unlocked',
  'mode': 'automatic',
  'mode-supported': ['automatic'],
  'module-name': 'ice',
  'type': 'eec'},
 {'clock-id': 5799633565435105048,
  'id': 3,
  'lock-status': 'unlocked',
  'mode': 'automatic',
  'mode-supported': ['automatic'],
  'module-name': 'ice',
  'type': 'pps'}]
```

Example Command Output

--do device-get

Retrieve information on a specific DPLL device with netlink command device-get & --do flag:

./tools/net/ynl/cli.py --spec Documentation/netlink/specs/dpll.yaml --do device-get --json '{"id":<id>}'

where: <id> = id of DPLL of interest

```
[root@localhost linux-6.11.7]# ./tools/net/ynl/cli.py --spec Documentation/netlink/specs/dpll.yaml --do device-get --json
'{"id":2}'
{'clock-id': 5799633565435105048,
 'id': 2,
 'lock-status': 'unlocked',
 'mode': 'automatic',
 'mode-supported': ['automatic'],
 'module-name': 'ice',
 'type': 'eec'}
```

Example Command Output

--dump pin-get

Retrieve information on all DPLL pins with netlink command pin-get & --dump flag:

./tools/net/ynl/cli.py --spec Documentation/netlink/specs/dpll.yaml --dump pin-get

```
[root@localhost linux-6.11.7]# ./tools/net/ynl/cli.py --spec Documentation/netlink/specs/dpll.yaml --dump pin-get
[{'board-label': 'CVL-SDP22',
  'capabilities': {'state-can-change', 'priority-can-change'},
  'clock-id': 5799633565435105048,
  'frequency': 1,
  'frequency-supported': [{'frequency-max': 25000000, 'frequency-min': 1}],
  'id': 17,
  'module-name': 'ice',
  'parent-device': [{'direction': 'input',
                     'parent-id': 2,
                     'phase-offset': 0,
                     'prio': 8,
                     'state': 'selectable'},
                    {'direction': 'input',
                     'parent-id': 3,
                     'phase-offset': 0,
                     'prio': 8,
                     'state': 'selectable'}],
  'phase-adjust': 0,
  'phase-adjust-max': 16723,
  'phase-adjust-min': -16723,
  'type': 'int-oscillator'},
```

Snippet of Example Command Output

--do pin-get

Retrieve information on a specific DPLL pin with netlink command pin-get & --do flag:

./tools/net/ynl/cli.py --spec Documentation/netlink/specs/dpll.yaml --do pin-get --json '{"id":<id>}'

where: <id> = id of pin of interest

```
[root@localhost linux-6.11.7]# ./tools/net/ynl/cli.py --spec Documentation/netlink/specs/dpll.yaml --do pin-get --json '{"
id":33}'
{'capabilities': {'state-can-change'},
 'clock-id': 5799633565435105048,
 'id': 33,
 'module-name': 'ice',
 'parent-pin': [{'parent-id': 19, 'state': 'disconnected'},
                {'parent-id': 20, 'state': 'connected'}],
 'phase-adjust-max': 0,
 'phase-adjust-min': 0,
 'type': 'synce-eth-port'}
```

Example Command Output

--do pin-set

Modify a pin’s attributes with netlink command pin-set & --do flag:

./tools/net/ynl/cli.py --spec Documentation/netlink/specs/dpll.yaml --do pin-set --json ‘{“id”:<id>, “<attr>”: “<value>”}’

where: <id> = id of pin of interest

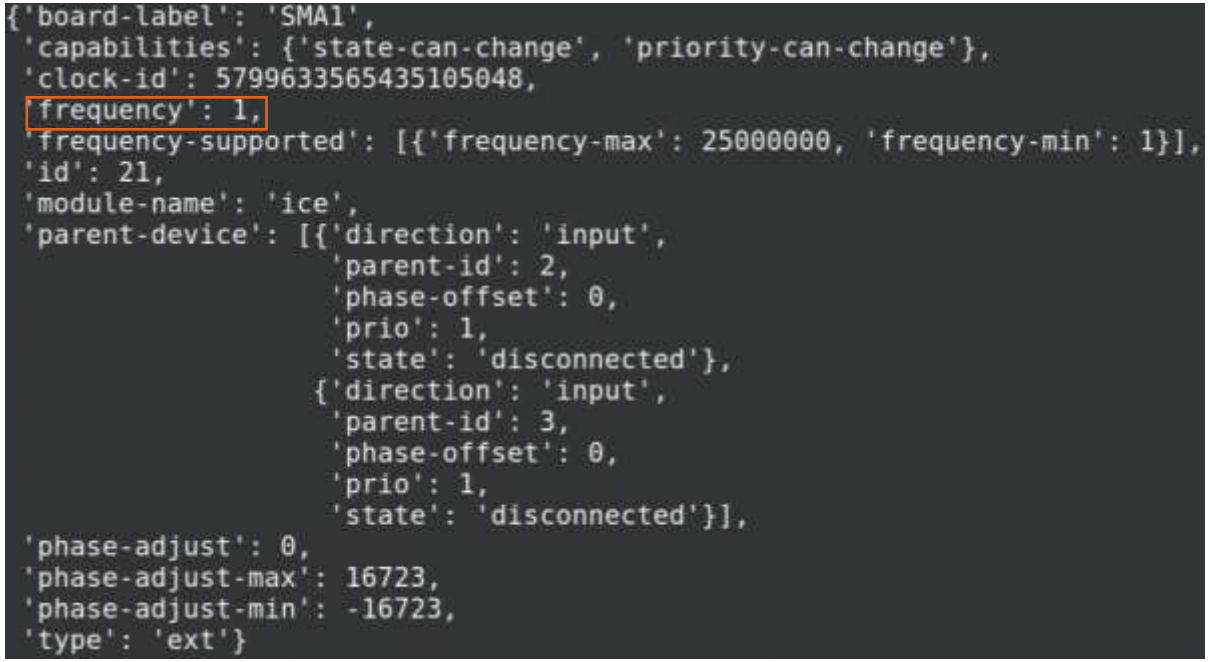
<attr> = attribute to be modified,

<value> = value to assign to attribute

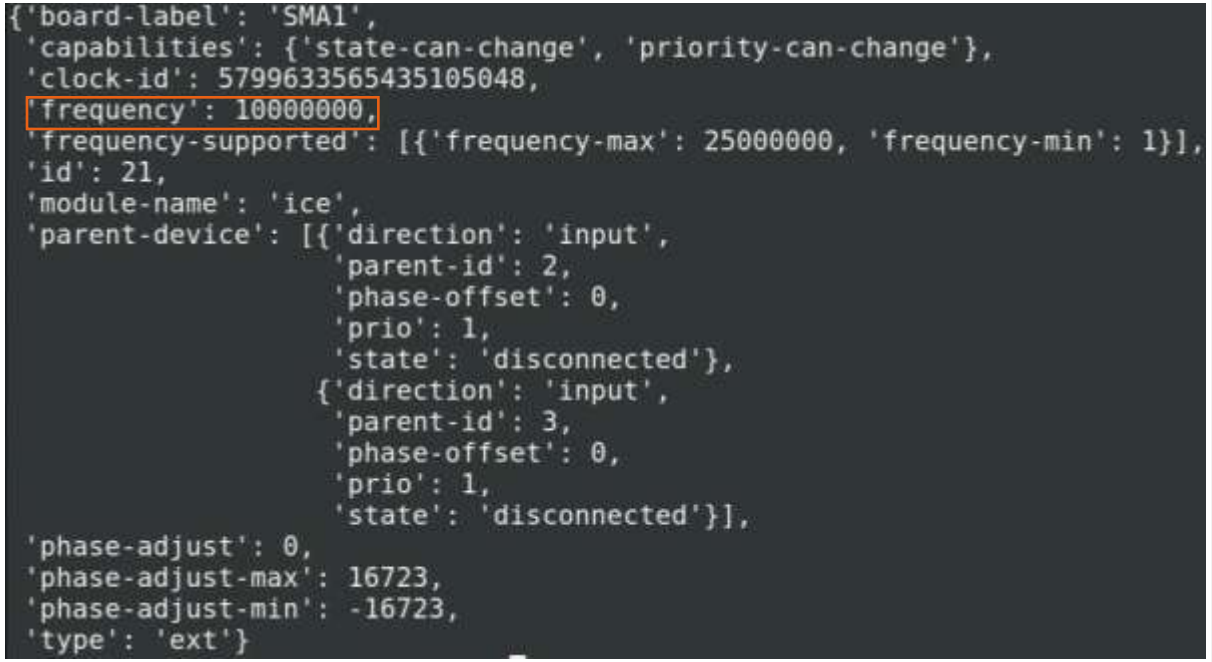
```
[root@localhost linux-6.11.7]# ./tools/net/ynl/cli.py --spec Documentation/netlink/specs/dpll.yaml --do pin-set --json '{"
id":21, "frequency":10000000}'
None
```

Example Command Output

--do pin-set



Before Command



After Command

--do pin-set on Parent DPLL

Modify a pin's attributes on their directly connected DPLL(s) with netlink command pin-set & --do flag:

./tools/net/ynl/cli.py --spec Documentation/netlink/specs/dpll.yaml --do pin-set --json '{"id":<id>, "parent-device": {"parent-id":<p_id>, "<attr>":"<value>"}}'

where: <id> = id of pin of interest,
<p_id> = id of parent DPLL,
<attr> = attribute to be modified,
<value> = value to assign to attribute

```
[root@localhost linux-6.11.7]# ./tools/net/ynl/cli.py --spec Documentation/netlink/specs/dpll.yaml --do pin-set --json '{"
id":20, "parent-device":{"parent-id":2, "prio":9}}'
None
```

Example Command Output

--do pin-set on Parent DPLL

```
{'board-label': 'C827_0-RCLKB',
 'capabilities': {'state-can-change', 'priority-can-change'},
 'clock-id': 5799633565435105048,
 'frequency': 1,
 'frequency-supported': [{'frequency-max': 25000000, 'frequency-min': 1}],
 'id': 20,
 'module-name': 'ice',
 'parent-device': [{'direction': 'input',
                    'parent-id': 2,
                    'phase-offset': 0,
                    'prio': 5,
                    'state': 'selectable'},
                   {'direction': 'input',
                    'parent-id': 3,
                    'phase-offset': 0,
                    'prio': 5,
                    'state': 'selectable'}],
 'phase-adjust': 0,
 'phase-adjust-max': 16723,
 'phase-adjust-min': -16723,
 'type': 'mux'}
```

Before Command

```
{'board-label': 'C827_0-RCLKB',
 'capabilities': {'state-can-change', 'priority-can-change'},
 'clock-id': 5799633565435105048,
 'frequency': 1,
 'frequency-supported': [{'frequency-max': 25000000, 'frequency-min': 1}],
 'id': 20,
 'module-name': 'ice',
 'parent-device': [{'direction': 'input',
                    'parent-id': 2,
                    'phase-offset': 0,
                    'prio': 9,
                    'state': 'selectable'},
                   {'direction': 'input',
                    'parent-id': 3,
                    'phase-offset': 0,
                    'prio': 5,
                    'state': 'selectable'}],
 'phase-adjust': 0,
 'phase-adjust-max': 16723,
 'phase-adjust-min': -16723,
 'type': 'mux'}
```

After Command

--do pin-set on Parent Pin

Modify a child pin's attributes on their parent pin with netlink command pin-set & --do flag:

./tools/net/ynl/cli.py --spec Documentation/netlink/specs/dpll.yaml --do pin-set --json '{"id":<id>, "parent-pin": {"parent-id":<p_id>, "<attr>":"<value>"}}'

where: <id> = id of pin of interest
<p_id> = id of parent pin
<attr> = attribute to be modified,
<value> = value to assign to attribute

```
[root@localhost linux-6.11.7]# ./tools/net/ynl/cli.py --spec Documentation/netlink/specs/dpll.yaml --do pin-set --json '{"
id":30, "parent-pin":{"parent-id":19, "state":"connected"}}'
None
```

Example Command Output

--do pin-set on Parent Pin

```
{'capabilities': {'state-can-change'},
 'clock-id': 5799633565435105048,
 'id': 30,
 'module-name': 'ice',
 'parent-pin': [{'parent-id': 19, 'state': 'disconnected'},
                {'parent-id': 20, 'state': 'disconnected'}],
 'phase-adjust-max': 0,
 'phase-adjust-min': 0,
 'type': 'synce-eth-port'},
```

Before Command

```
{'capabilities': {'state-can-change'},
 'clock-id': 5799633565435105048,
 'id': 30,
 'module-name': 'ice',
 'parent-pin': [{'parent-id': 19, 'state': 'connected'},
                {'parent-id': 20, 'state': 'disconnected'}],
 'phase-adjust-max': 0,
 'phase-adjust-min': 0,
 'type': 'synce-eth-port'},
```

After Command

--do pin-set on Parent Pin

```
{'board-label': 'C827_0-RCLKA',
 'capabilities': {'state-can-change', 'priority-can-change'},
 'clock-id': 5799633565435105048,
 'frequency': 1953125,
 'frequency-supported': [{'frequency-max': 25000000, 'frequency-min': 1}],
 'id': 19,
 'module-name': 'ice',
 'parent-device': [{'direction': 'input',
                    'parent-id': 2,
                    'phase-offset': 0,
                    'prio': 4,
                    'state': 'connected'},
                   {'direction': 'input',
                    'parent-id': 3,
                    'phase-offset': 0,
                    'prio': 4,
                    'state': 'connected'}],
 'phase-adjust': 0,
 'phase-adjust-max': 16723,
 'phase-adjust-min': -16723,
 'type': 'mux'}
```

Output after pin-get

--multi flag

Concatenate multiple --do operations with --multi flag:

./tools/net/ynl/cli.py --spec Documentation/netlink/specs/dpll.yaml --multi <do_operation> --multi <do_operation> ...

where: <do_operation> = operation that would be encapsulated with a --do flag.

E.g. pin-get '{"id":20}'

```
[root@localhost linux-6.11.7]# ./tools/net/ynl/cli.py --spec Documentation/netlink/specs/dpll.yaml --multi pin-get '{"id":
21}' --multi pin-get '{"id":24}'
```

Example Command

--multi flag

```
[{'board-label': 'SMA1',
  'capabilities': {'state-can-change', 'priority-can-change'},
  'clock-id': 5799633565435105048,
  'frequency': 1,
  'frequency-supported': [{'frequency-max': 25000000, 'frequency-min': 1}],
  'id': 21,
  'module-name': 'ice',
  'parent-device': [{'direction': 'input',
                     'parent-id': 2,
                     'phase-offset': 0,
                     'prio': 1,
                     'state': 'disconnected'},
                    {'direction': 'input',
                     'parent-id': 3,
                     'phase-offset': 0,
                     'prio': 1,
                     'state': 'disconnected'}],
  'phase-adjust': 0,
  'phase-adjust-max': 16723,
  'phase-adjust-min': -16723,
  'type': 'ext'},
 {'board-label': 'REF-SMA1',
  'capabilities': {'state-can-change'},
  'clock-id': 5799633565435105048,
  'frequency': 1,
  'frequency-supported': [{'frequency-max': 25000000, 'frequency-min': 1}],
  'id': 24,
  'module-name': 'ice',
  'parent-device': [{'direction': 'output',
                     'parent-id': 2,
                     'state': 'disconnected'},
                    {'direction': 'output',
                     'parent-id': 3,
                     'state': 'connected'}],
  'phase-adjust': 0,
  'phase-adjust-max': 480307,
  'phase-adjust-min': -480307,
  'type': 'ext'}]
```

Example Command Output

Live Demonstration

Thank you

Any questions?

With Special Thanks to
Arkadiusz Kubalewski, Zoli Fodor,
Przemek Korba & Stephen P Mroczek