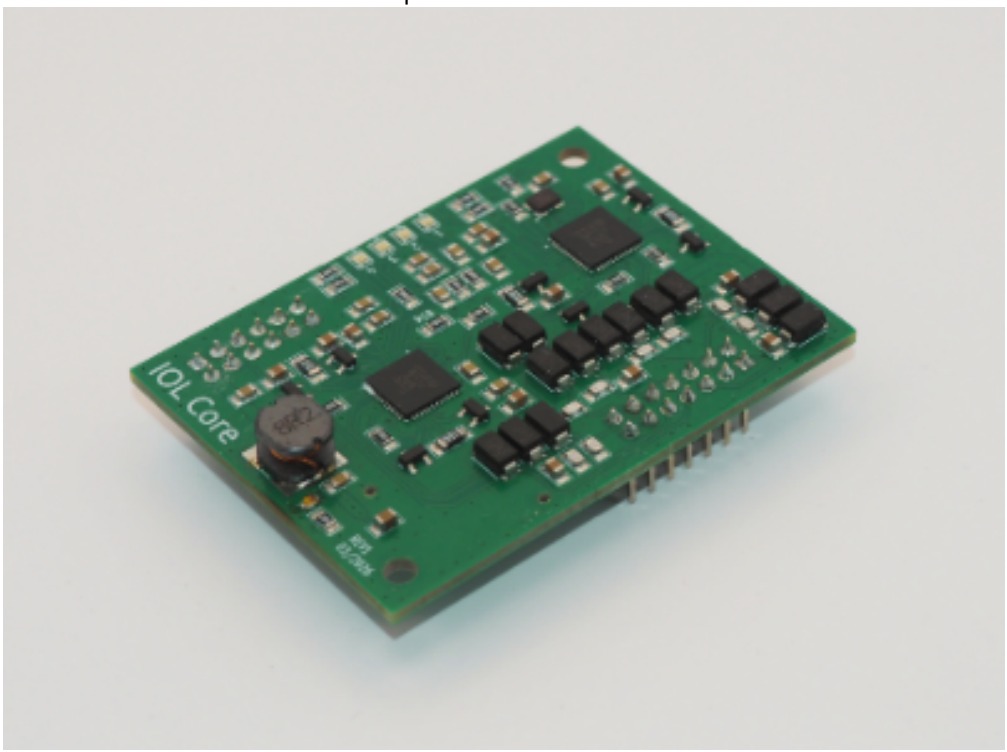
# 2.2 Arguments and Timing

## Hardware manual

www.pinetek-networks.com



# Table of Contents

2.2 Arguments and Timing ..... 5
*Command line arguments* ..... 5
*IOL Port modes* ..... 5
*Delay current limit / blanking time* ..... 5
*Timing considerations* ..... 6

# 2.2 Arguments and Timing

## Command line arguments

| | |
|---|---|
| -h, –help | shows help message and exits |
| -v, –version | prints version information and exits |
| -m0, –mode0 | Operating Mode Port 0 (X1), Possible values 0: IOL, 1: DI, 2: DO, 3: OFF |
| -m1, –mode1 | Operating Mode Port 1 (X2), Possible values 0: IOL, 1: DI, 2: DO, 3: OFF |
| -e, –extclock | Use clock for MAX14819 from ext source (not required for the IOL HAT) |
| -d, –delaycurrentlimit | Delay time in ms for current limit enabling after port enable, range [1..1500]. Please see below for further details |
| -b, –blankingtime | Blanking time setting 0=short…3=long for capacitive loads. Please see below for further details |
| -r, –realtime | Run realtime on given kernel. Requires root rights, see below for details |
| -i, –iolport | This specifies the IOL port (see jumper settings on the IOL HAT). Standard is port 1/2. See hardware manual for the SPI and GPIOs in use. Possible values 12 (default) and 34 |
| -t, –tcpport | Specify the TCP port for communication. If not specified here, TCP port is 12010 for IOL port 1/2 and 12011 if –iolport 34 is set. |

## IOL Port modes

The operation mode of the ports are defined with -m0 and -m1 (respectively –mode0 and –mode1) for the IOL HAT ports.

- ** 0 (Default) - SDCI/IO-Link operation**
  - Parameters (CMD_READ, CMD_WRITE) and Process Data (CMD_PD) as provided by the connected device
- ** 1 - DI: Input mode**
  - No parameters (CMD_READ, CMD_WRITE) available
  - Process data: 1 byte IN/ 0 byte OUT (CMD_PD)
    - CQ-Line against GND: Process data IN = 0x00
    - CQ-Line against +24V: Process data IN = 0x01
    - Power needs to be enabled (CMD_PWR) for operation
- ** 2 - DO: Output mode**
  - No parameters (CMD_READ, CMD_WRITE) available
  - Process data: 0 byte IN/1 byte OUT (CMD_PD)
    - Process data OUT = 0x01: CQ-Line driven to 24V
    - Process data OUT = 0x00: CQ-Line driven to GND
    - Power needs to be enabled (CMD_PWR) for operation
- **3 - OFF**: Port off/disabled

The port mode can only be set at program start.

## Delay current limit / blanking time

The -d (–delaycurrentlimit, default is 0) parameter allows to disable the current limit of 500mA after port enable after powering the port (CMD_PWR). This is needed for some devices that have a higher current demand at device start (note: often, the specifications for those devices are <500mA).



Use the **-d** parameter with caution, to much current on L+ may damage the IOL HAT circuit. Limit the current at the IOL HAT input ≤1A for testing and iterate to the required time with this limit. Damages to the IOL HAT hardware that are caused by the use of this are not covered by the IOL HAT product warranty. The usage is on your own risk.

The blanking time parameter -b (–blankingtime, default is 0) sets the current-limit blanking time for the L+ sensor supply. Longer blanking times allow for charging of larger capacitive loads.

| **Blanking time parameter** | 0 (default) | 1 | 2 | 3 |
|---|---|---|---|---|
| **Blanking time** | 5,5ms | 16,5ms | 55ms | 165ms |

## Timing considerations

IO-Link is a real-time protocol that requires fast response and cycle times to not run into timeouts. For running with sensors etc. that are queried 1-2 times per second, those timeouts are not crititcal. The iol-master-applcan run with standard user rights without further considerations.

If timeouts are critical (e.g., when HMI devices are connected that would change screen when timeouts occur), it is recommended to run the iol-master-appl in realtime mode. The solution is based on this description:

https://forums.raspberrypi.com/viewtopic.php?t=228727

As preparation, one core needs to be reserved. This is done by adding the this argument to /boot/firmware/cmdline.txt (for older versions of Raspberry OS, the file is /boot/cmdline.txt):

```
isolcpus=3
```

where 3 is the core you want to reserve (can be 0..3).



All arguments in the cmdline.txt need to be in ONE line, i.e. you need to add the isolcpus=3 at the end of the line. Otherwise the core reservation as described above will not work.

The core needs to be matching with the option that you are giving when starting the iol-master-appl, e.g., for core 3 (matching with the above example of the cmdline.txt) it would be

```
sudo iol-master-appl -r 3
```



Each core needs to be explicitly stated. To reserve e.g. core 2 and 3 (the last two cores), the statement would be isolcpus=2,3. In this case, two iol-master-appl can be executed: iol-master-appl -r 2 and iol-master-appl -r 3, both running in realtime mode.