

# M68/M64 Module Data Sheet

---

This document describes the M68/M64, a family of Network Synchronization Modules for packet network timing applications based on the Silicon Labs AccuTime™ technology. Besides its local oscillator, these modules can use multiple reference time and frequency sources to generate precision time and frequency including PPS and ToD as well as IEEE Std 1588-2008. IEEE Std 1588-2008 defines the Precision Time Protocol (PTP) version 2 (PTPv2). In this manual, we will use PTP to refer to this standard. They implement both a Grandmaster and an Ordinary PTP clock. The M68 is a superset of the M64 and also implements both a transparent or boundary clock.

The M68/M64 features full connectivity to gigabit networks with wirespeed pass-through technology allowing for new applications.

This document describes how to design with the M68/M64 module and how to use it in an embedded application. The integration section includes the pinout of the module and other physical aspects like the options of power supply. It also describes how to connect the necessary key components which are a prerequisite for proper operation.

This document applies to software Release 3.0 of the M64/M68. Modules can be updated to Release 3.0 by following the directions in [23. Appendix 8: Firmware Upgrade](#).

## KEY FEATURES

- Low power microprocessor technology with hardware timestamping
- 8 Mbytes SDRAM memory, 2 Mbytes flash memory
- 10/100/1000 Ethernet PHY
- RGMII interface to Host, or to an external PHY
- Timing interface including PPS, TOD, synchronized frequency
- Comprehensive IEEE 1588-2008 implementation and SyncE support
- Communication servers for serial ports, Telnet and SSH
- Edge pads for surface mount to host boards
- RoHS compliant
- Built-in SNTP server
- Wirespeed pass-through eliminating the need for a switch in many applications

# Table of Contents

<b>1. General Description</b>	<b>8</b>
1.1 Key Feature	8
1.2 Host Interface	8
1.3 Communication Interfaces	8
1.4 JSON Machine-to-Machine Communications	8
1.5 Block Diagram	9
<b>2. Packaging</b>	<b>10</b>
2.1 Dimensions	10
2.2 Recommended Land Pattern	10
2.3 Soldering and Handling	11
2.4 Tape and Reel	11
2.4.1 Pin 1 Orientation	12
2.4.2 Tape and Reel Specification	13
<b>3. LED Description</b>	<b>14</b>
<b>4. Pin List</b>	<b>15</b>
4.1 Pin Descriptions	17
4.1.1 Analog Control of Oscillator	17
4.1.2 Write Protection	17
4.1.3 Serial Ports	17
4.1.4 PF3/LOCK Pin	17
4.2 Clocking	18
4.3 Power Supply	18
4.3.1 Digital Power Supply	18
4.4 Time I/O	19
4.4.1 Reference Time Input	19
4.4.2 Precision Time Output	19
<b>5. Specifications</b>	<b>20</b>
5.1 Metrics	20
5.2 Absolute Maximum Ratings	20
5.3 Recommended Operating Conditions	20
5.4 DC Electrical Characteristics	21
5.5 AC Electrical Characteristics	21
<b>6. PCB Design Considerations</b>	<b>22</b>
6.1 Land Pattern	22
6.2 Power	22
6.2.1 Required Voltage	22
6.2.2 Decoupling Capacitors	23

6.3 Clock Signal MCLKOUT1 . . . . .	.23
6.4 Ethernet Interface . . . . .	.23
6.4.1 Magnetics . . . . .	.23
6.5 RGMII Interface. . . . .	.24
<b>7. Application Software . . . . .</b>	<b>25</b>
7.1 System Access . . . . .	.25
7.2 System Files. . . . .	.25
7.3 Module Shell Environment . . . . .	.25
7.3.1 High-Level Initialization. . . . .	.25
7.3.2 System Commands . . . . .	.25
7.3.3 Communication Servers . . . . .	.25
7.4 Passthrough . . . . .	.26
7.5 VLAN . . . . .	.27
7.6 MDIO . . . . .	.27
7.7 Link Speed . . . . .	.28
7.8 PHY Reset . . . . .	.28
7.9 SyncE . . . . .	.28
7.10 SFP . . . . .	.29
<b>8. IEEE 1588/PTP Engine . . . . .</b>	<b>30</b>
8.1 General Features . . . . .	.30
8.2 PTP Software Implementation . . . . .	.30
8.2.1 Protocol Layer. . . . .	.30
8.2.2 OS Abstraction Layer . . . . .	.30
8.3 GNSS Driver. . . . .	.31
8.4 AccuTime Software . . . . .	.31
8.5 PTP API . . . . .	.31
8.5.1 Starting/Stopping the PTP Engine . . . . .	.32
8.5.2 Pulse Train . . . . .	.32
8.5.3 Human Machine Interface . . . . .	.33
<b>9. Network Subsystem. . . . .</b>	<b>38</b>
9.1 TCP/IP Stack Architecture . . . . .	.38
9.2 Application Layer Components . . . . .	.38
9.2.1 DHCP Client . . . . .	.38
9.2.2 DNS Client . . . . .	.38
9.3 BSD Sockets API . . . . .	.38
9.4 Network Initialization . . . . .	.39
<b>10. Using the AccuTime Software . . . . .</b>	<b>40</b>
10.1 PTP Engine Modes . . . . .	.40
10.1.1 Mode 0. . . . .	.40
10.1.2 Mode 1. . . . .	.40

10.1.3 Mode 2. . . . .	.40
10.1.4 Mode 3. . . . .	.40
10.1.5 Mode 4. . . . .	.41
10.1.6 Slave-Only Mode . . . . .	.41
10.2 GNSS Interface . . . . .	.41
10.3 PTP Clock States . . . . .	.41
10.3.1 FREE State . . . . .	.41
10.3.2 SYNTONIZING State . . . . .	.41
10.3.3 SYNCHRONIZING State. . . . .	.42
10.3.4 HOLDOVER State . . . . .	.42
10.4 PPS/TOD Output. . . . .	.43
10.5 Managing the Two Interfaces (M68 Only) . . . . .	.43
10.6 Unicast Operations . . . . .	.43
10.6.1 Unicast Master . . . . .	.44
10.6.2 Limiting the Total Message Rate . . . . .	.46
10.6.3 Unicast Slave . . . . .	.47
10.6.4 Unicast Both . . . . .	.49
10.6.5 Message Transmission Duration . . . . .	.50
10.6.6 Discovery Query Interval. . . . .	.50
10.6.7 Monitoring Unicast Operations . . . . .	.50
10.7 Gateway and Boundary Clock (M68 Only) . . . . .	.50
10.7.1 The Gateway Clock . . . . .	.51
10.7.2 Managing the Two Ports . . . . .	.52
10.7.3 Unicast. . . . .	.52
10.8 Performance Optimization. . . . .	.53
10.8.1 Direct Connection . . . . .	.54
10.8.2 Low Traffic Network . . . . .	.54
10.8.3 High Traffic Network . . . . .	.54
10.8.4 ITU-T G.8261, G.8265.1, G.8275.2 Settings . . . . .	.54
10.8.5 ITU-T G.8273.2, G.8275.1 settings . . . . .	.54
10.9 Power Profile . . . . .	.55
10.10 ITU-T G.8265 Profile . . . . .	.55
10.11 ITU-T G.8275.1 Profile . . . . .	.56
10.12 ITU-T G.8275.2 Profile . . . . .	.57
<b>11. Command Reference. . . . .</b>	<b>58</b>
11.1 Shell Commands . . . . .	.58
11.1.1 ptp2 start . . . . .	.58
11.1.2 ptp2 stop . . . . .	.58
11.1.3 ptp2 hmi . . . . .	.58
11.1.4 ptp2 config . . . . .	.59
11.1.5 ptp2. . . . .	.59
<b>12. JSON Usage . . . . .</b>	<b>60</b>
12.1 Enabling JSON . . . . .	.61

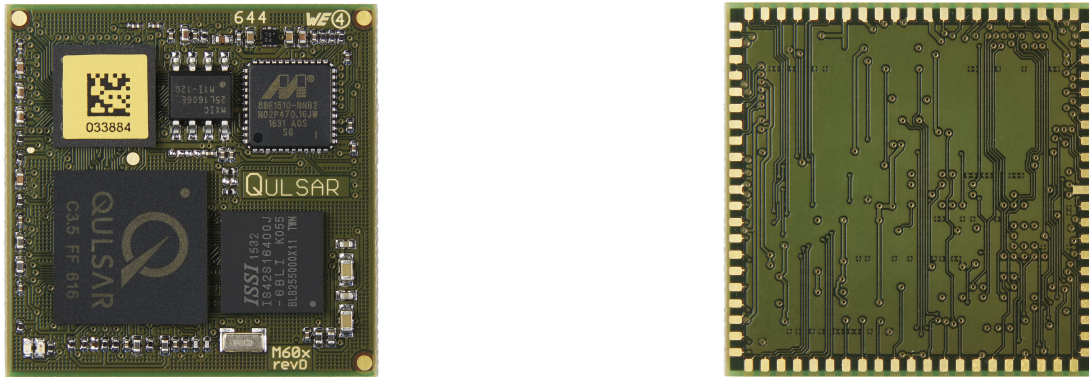
<b>13. Product Documentation . . . . .</b>	<b>62</b>
<b>14. Part Numbers . . . . .</b>	<b>63</b>
<b>15. Customer and Technical Support . . . . .</b>	<b>64</b>
<b>16. Appendix 1: System Files . . . . .</b>	<b>65</b>
<b>17. Appendix 2: System CLI Commands . . . . .</b>	<b>68</b>
<b>18. Appendix 3: PTP Command Reference . . . . .</b>	<b>108</b>
<b>19. Appendix 4: Troubleshooting LED Error Codes . . . . .</b>	<b>123</b>
<b>20. Appendix 5: Command Shell Limitations . . . . .</b>	<b>124</b>
<b>21. Appendix 6: Environment Variable . . . . .</b>	<b>125</b>
<b>22. Appendix 7: Clock Setup . . . . .</b>	<b>128</b>
<b>23. Appendix 8: Firmware Upgrade . . . . .</b>	<b>132</b>
<b>24. Appendix 9: JSON Interface Definition . . . . .</b>	<b>133</b>
24.1 Clock Commands . . . . .	134
24.1.1 "clock info" . . . . .	135
24.1.2 "clock type" . . . . .	135
24.1.3 "clock twostep" . . . . .	136
24.1.4 "clock slaveonly" . . . . .	136
24.1.5 "clock domain" . . . . .	136
24.1.6 "clock priority" . . . . .	136
24.1.7 "clock quality" . . . . .	137
24.1.8 "clock profile" . . . . .	137
24.2 Port Commands . . . . .	137
24.2.1 "port state" . . . . .	137
24.2.2 "port announce" . . . . .	138
24.2.3 "port receipt" . . . . .	138
24.2.4 "port sync" . . . . .	138
24.2.5 "port delay" . . . . .	138
24.2.6 "port pdelay" . . . . .	138
24.2.7 "port mechanism" . . . . .	139
24.2.8 "port protocol" . . . . .	139
24.2.9 "port compatibility" . . . . .	139
24.2.10 "port asymmetry" . . . . .	139
24.2.11 "port unicast" . . . . .	139
24.2.12 "port profile" . . . . .	140
24.3 Unicast Commands . . . . .	140
24.3.1 "unicast status" . . . . .	140
24.3.2 "unicast negotiation" . . . . .	140
24.3.3 "unicast query" . . . . .	141
24.3.4 "unicast duration" . . . . .	141
24.3.5 "unicast timeout" . . . . .	141
24.3.6 "unicast limit" . . . . .	141
24.3.7 "unicast filter" . . . . .	142

24.3.8 "unicast node" . . . . .	144
24.4 Power Profile Commands . . . . .	145
24.4.1 "power gmid". . . . .	145
24.4.2 "power inacc" . . . . .	145
24.5 G8265 Profile Commands. . . . .	145
24.6 G8275 Profile Commands. . . . .	145
24.6.1 "g8275 priority" . . . . .	146
24.6.2 "g8275 steps" . . . . .	146
24.6.3 "g8275 [port_number] priority" . . . . .	146
24.6.4 "g8275 [port_number] masteronly" . . . . .	146
24.7 Dataset Commands . . . . .	146
24.7.1 "dataset default" . . . . .	147
24.7.2 "dataset current" . . . . .	147
24.7.3 "dataset parent". . . . .	148
24.7.4 "dataset time" . . . . .	149
24.7.5 "dataset port [port_number]" . . . . .	150
24.7.6 "dataset foreign [port_number]" . . . . .	151
24.7.7 "dataset unicast [port_number]" . . . . .	153
24.7.8 "dataset power". . . . .	154
24.7.9 Dataset g8275 Commands . . . . .	154
24.8 Time Commands . . . . .	155
24.8.1 "time info". . . . .	156
24.8.2 "time sync" . . . . .	156
24.8.3 "time arb ptp utc ntp" . . . . .	157
24.8.4 "time offset" . . . . .	157
24.8.5 "time leap" . . . . .	158
24.8.6 "time leap save". . . . .	158
24.8.7 "time leap load" . . . . .	158
24.8.8 "time leap 59" . . . . .	158
24.8.9 "time leap 61" . . . . .	158
24.8.10 "time leap 59 61 YYYY-MM-DD" . . . . .	159
24.8.11 "time timescale" . . . . .	159
24.8.12 "time update" . . . . .	159
24.8.13 "time holdover" . . . . .	159
24.9 Frequency Commands . . . . .	160
24.9.1 "freq status" . . . . .	160
24.9.2 "freq option" . . . . .	160
24.9.3 "freq ref" . . . . .	160
24.9.4 "freq prefer" . . . . .	161
24.9.5 "freq input [input_number]" . . . . .	162
24.10 SyncE Commands . . . . .	163
24.10.1 "synce status" . . . . .	163
24.10.2 "synce esmc" . . . . .	163
24.10.3 "synce quality". . . . .	164
24.11 GPS (PPS/TOD-Input) Commands . . . . .	164
24.11.1 "gps status". . . . .	165

24.11.2	"gps ppsin"	165
24.11.3	"gps todin"	166
24.11.4	"gps delay"	166
24.11.5	"gps class"	166
24.11.6	"gps interface"	166
24.11.7	"gps format"	167
24.11.8	"gps void"	167
24.11.9	"gps message"	167
24.11.10	"gps cm"	168
24.11.11	"gps nmea"	169
24.12	Pulsetime (PPS/TOD-Output) Commands	169
24.12.1	"pulsetime status"	169
24.12.2	"pulsetime mode"	170
24.12.3	"pulsetime error"	170
24.12.4	"pulsetime auto"	170
24.12.5	"pulsetime pulse"	170
24.12.6	"pulsetime todout"	171
24.12.7	"pulsetime delay"	171
24.12.8	"pulsetime interface"	171
24.12.9	"pulsetime format"	171
24.12.10	"pulsetime void"	172
24.12.11	"pulsetime zone"	172
24.12.12	"pulsetime cm"	173
24.12.13	"pulsetime stop"	173
24.12.14	"pulsetime start"	174
24.13	Nettime (SNTP Service) Commands	174
24.13.1	"nettime status"	174
24.13.2	"nettime mode"	175
24.13.3	"nettime poll"	175
24.13.4	"nettime stop"	175
24.13.5	"nettime start"	175
24.14	Engine Commands	175
24.14.1	"engine json on off ver"	176
24.14.2	"engine init <key>"	176
24.14.3	"engine nvm"	176
24.14.4	"engine debug"	176
24.14.5	"engine verbose"	177
24.14.6	"engine monitor"	177
24.14.7	"engine port [port_number]"	177
24.14.8	"engine asymmetry [port_number]"	178
24.14.9	"engine freqout"	178
<b>25.</b>	<b>EULA</b>	<b>179</b>

## 1. General Description

This section shows the M68/M64 module in approximately two times the actual size. The M68 adds an oscillator to the module.



**Figure 1.1. Top Side and Bottom Side Views**

### 1.1 Key Feature

- Low power microprocessor technology with hardware timestamping
- 8 Mbytes SDRAM memory, 2 Mbytes flash memory
- 10/100/1000 Ethernet PHY with full PTP functionality
- RGMII interface to Host, or to an external PHY. For the M68 this includes full PTP functionality.
- Timing interface including PPS, TOD, syntonized frequency
- Comprehensive IEEE 1588-2008 implementation and SyncE support
- Communication servers for serial ports, Telnet and SSH
- Edge pads for surface mount to host boards
- RoHS and WEE compliant
- Built-in SNTP server
- Wirespeed pass-through eliminating the need for a switch in many applications

### 1.2 Host Interface

The RGMII Host Interface of the module is intended to be connected directly to a generic RGMII interface offered by the host, or an RGMII PHY. The designer can choose voltage levels for adaptation to his RGMII port of the host system.

For applications with host processors offering only an SGMII interface, a PHY IC such as the 88E1512 from Marvell that converts between SGMII and RGMII needs to be added.

### 1.3 Communication Interfaces

The M68/M64 offers three asynchronous serial ports besides the two Ethernet ports. Advanced users with the Development Kit will also have access to other serial and parallel interfaces including an analog section to connect to other peripherals outside the module.

### 1.4 JSON Machine-to-Machine Communications

New with Firmware release 3.0 is the addition of JavaScript Object Notation (JSON) structured configuration information. This greatly simplifies the development of software on host systems to interface with the M64/M68.



## 1.5 Block Diagram

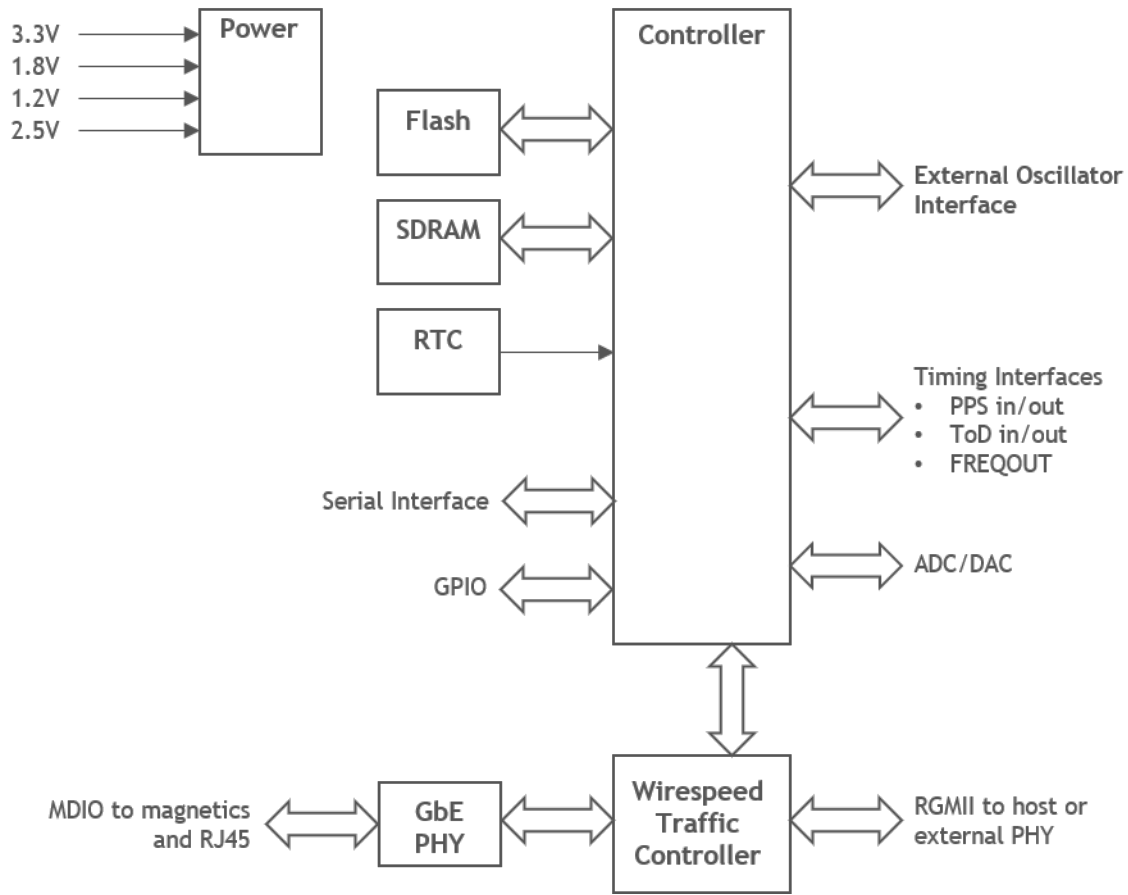


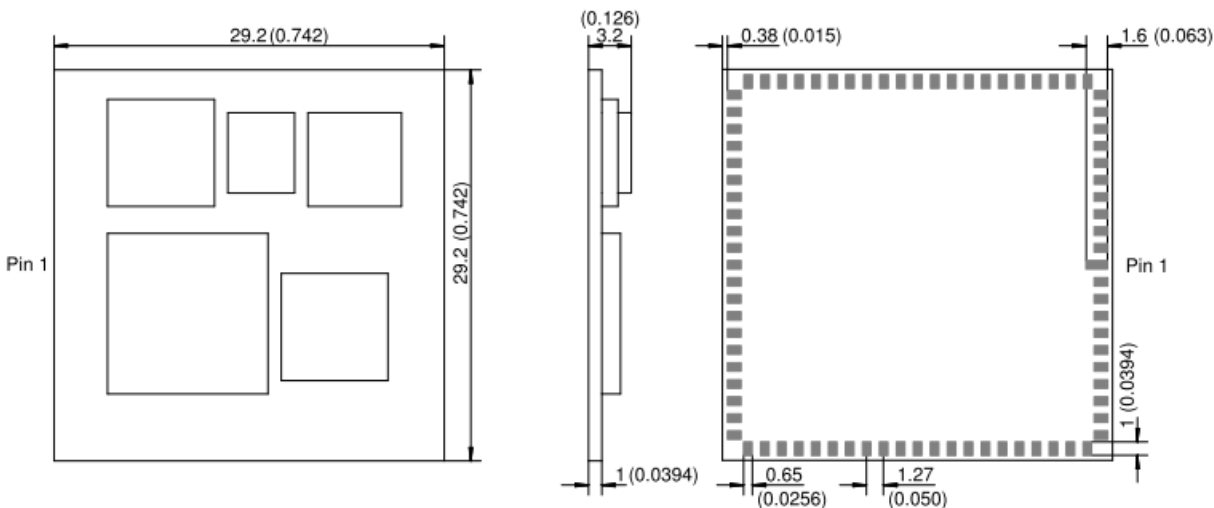
Figure 1.2. M68/M64 Block Diagram

## 2. Packaging

### 2.1 Dimensions

The module dimensions match the standard for a LGA-84 package, with exception for the height. The height varies over the module surface, with maximum and minimum values given below.

When soldered directly to the motherboards, the module's distance to that board depends on the amount of solder used but can usually be considered to be zero.

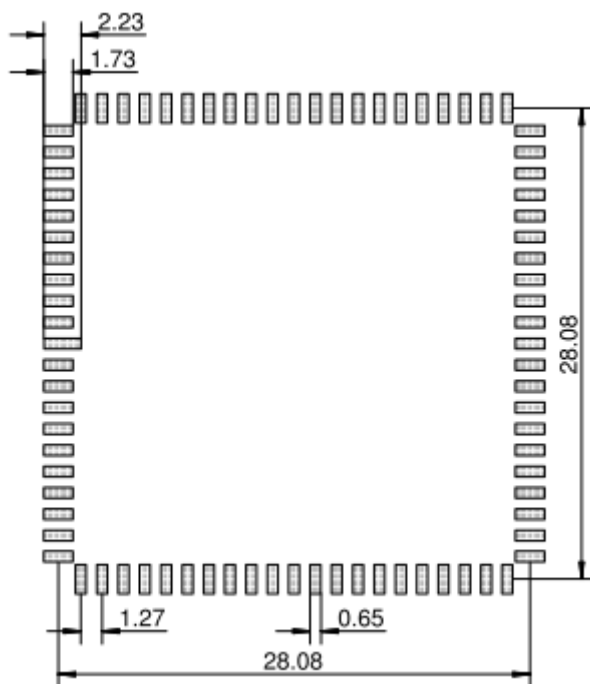


**Figure 2.1. Dimensions for M68/M64 (unit: mm)**

**Note:** The height may vary with component vendor specification.

### 2.2 Recommended Land Pattern

This is the recommended PCB land pattern for direct soldering of the M68/M64 module to a host board for all designs.



**Figure 2.2. Recommended Land Pattern (unit: mm)**

### 2.3 Soldering and Handling

Since the M68/M64 is a RoHS-compliant device, it is designed to tolerate lead-free soldering processes. The diagram below shows the recommended soldering profile for the device.

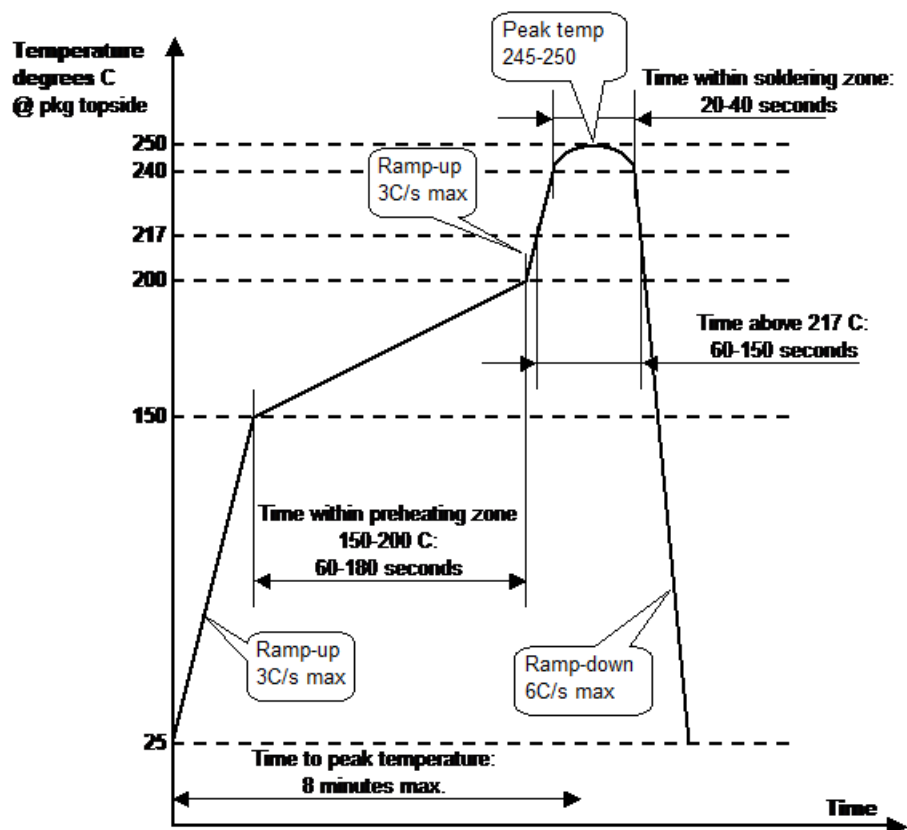


Figure 2.3. Reflow Soldering Profile

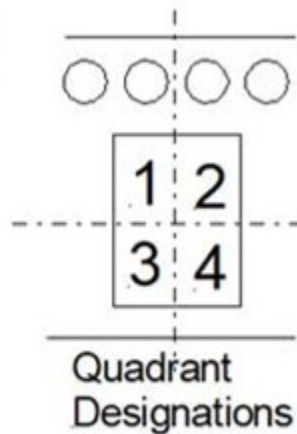
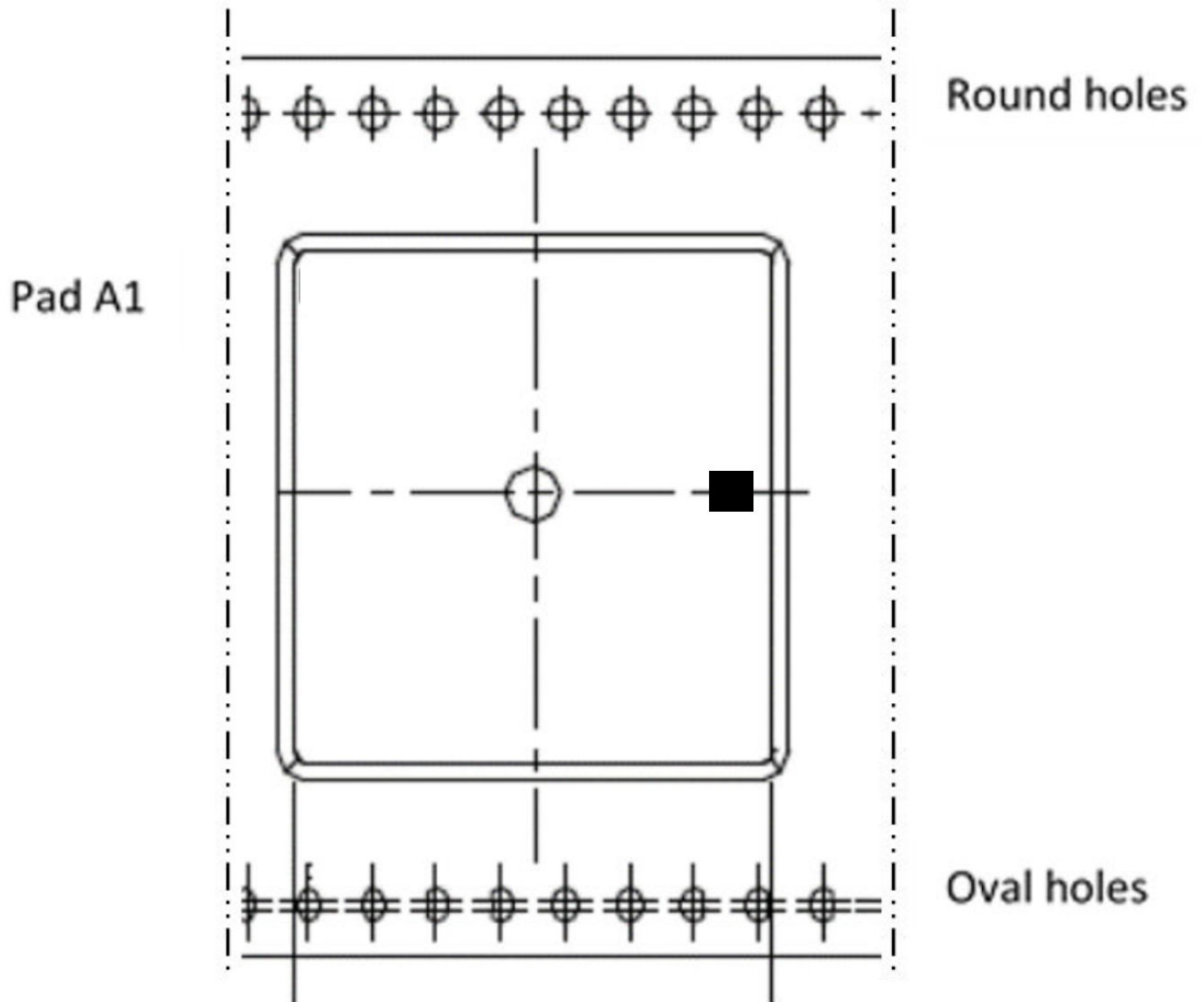
The M68/M64 is classified as MSL 3.

### 2.4 Tape and Reel

The M68/M64 is delivered on tape and reel and this section describes the orientation of the LGA components and the tape used for its packaging.

### 2.4.1 Pin 1 Orientation

Part orientation for tape and reel is illustrated below. The Pin 1 marker is between quadrants 3 and 4.



**2.4.2 Tape and Reel Specification**

Package	Tape width (mm)	Pitch (mm)	Reel size (mm)	Devices per reel
LGA84	44	36	330	250

### 3. LED Description

The module has two on-board LEDs. See [19. Appendix 4: Troubleshooting LED Error Codes](#) for more information. One LED is green which shows the status of the boot sequence. This LED can be controlled by the user using firmware. The other LED is red and will light up briefly at the start-up of the system. If the red LED keeps being lit, the system is not operating properly. This LED is not user controllable.

## 4. Pin List

The term “pin” used in the following table and throughout this document refers to one of the 84 pads on the bottom side of the module.

Pin 1 is located in the middle of the left side of the module when viewed from the top, see [Figure 2.1 Dimensions for M68/M64 \(unit: mm\)](#) on [page 10](#). The rest of the pins are enumerated counter-clockwise around the module from this pin.

The codes in the Type column below are: I for Input, O for Output, B for Bidirectional, P for Power, (PU) for Pull Up, and (PD) for Pull Down.

Most pins have 3.3 Volts signaling levels. In the Description column pins not belonging to the 3.3 Volts region, e.g., those pins powered by the External RGMII interface, are noted.

Pin Group	Pin #	Pin Name	Type	Description
RGMII	2	MIRQ0	I	Ext. GbE PHY Interrupt Powered by VCC_RGMII
	3	MCKOUT1	O	25 MHz Clock.  3.3 Volts output.  Not suitable for clocking external PHY.
Misc.	4	MRXOUT	O	RTC test / Active high power supply wake-up output
	5	MWAKE	I (PD)	Active high wake-up input
Clock	6	CPUCLK	I	External clock input (1.8 V level)
Misc.	7	MRSTOUT	O	Active low reset output
Time	8	TODIN	I	Time of day in
	9	TODOUT	O	Time of day out
GPIO	10	PF2	B	Port F GPIO
	11	PF1	B	
	12	PF5	B	
	13	PF4	B	
	14	PF7	B	
	15	PF0	B	
	16	PF3/LOCK	B	
	17	PF6	B	
Analog	19-21, 23	ACH0-ACH2, ACH3	I	Analog to Digital Converter multi-plexer inputs
	24, 25	AOUT0, AOUT1	O	Digital to Analog Converter outputs
	26	EXTREF	I	Optional external voltage reference input for ADC
GPIO/Async. Serial ports	27-30	PJ0_UTX1, PJ1_URX1, PJ2_URTS1, PJ3_UCTS1	B	Port J bits 0-3 / Serial port 1
	31-34	PE0_UTX2, PE1_URX2, PE2_URTS2, PE3_UCTS2	B	Port E bits 0-3 / Serial port 2

Pin Group	Pin #	Pin Name	Type	Description
GPIO / SPI / Generic Serial Interface	35-38	PE4_UTX3, PE5_URX3, PE6_URTS3, PE7_UCTS3	B	Port E bits 4-7 / Serial port 3
	39	PB7_PC0	B	See Pin descriptions
	40	PD5_GSIIN	B	Port D bit 5 / SPI MISO / GSI input
	41	PD7_GSICK	B	Port D bit 7 / SPI clock / GSI clock
	42	PD6_GSIIO	B	Port D bit 6 / SPI MOSI / GSI bidirectional
Ethernet	44,45	LED2,LED1	O	RJ45 LED's
	46,47 48,49 50,51 52,53	MDI_A+,MDI_A- MDI_B+,MDI_B- MDI_C+,MDI_C- MDI_D+,MDI_D-	B	Ethernet, direct connection to magnetics
Debug	54	MSDIN	I (PU)	Serial debug data in
	55	MSDOUT	O	Serial debug data out
	56	MRESET	I (PU)	Active low reset input
	57	MIRQOUT	O	Active low debug interrupt
	58	MIRQ1	I	Shared Interrupt
	59	MCKOUT0	O	Debug clock
	62	WP	I (PD)	Active low flash write protect
Time & Frequency	66	FREQOUT_2	O	Secondary Frequency out. Identical signal to FREQOUT
	67	FREQOUT	O	Frequency out
	68	PPSIN	I	Pulse per second in
	69	PPSOUT	O	Pulse per second out
RGMII All pins Powered by VCC_RGMII Supply	70	PB1_MDIO	B	Ext. GbE PHY Control (MDIO) Powered by VCC_RGMII
	71	PB2_MDC	O	Ext. GbE PHY Control (MDC) Powered by VCC_RGMII
	73,74	TXD0-TXD1	O	TX Data
	75	TX_CTL	O	RGMII control
	76	TX_CLK	O	TX Clock
	77,78	TXD2-TXD3	O	TX Data
	79-80	RXD0-RXD1	I	RX Data
	81	RX_CTL	I	RGMII control
	82	RX_CLK	I	RX Clock
	83,84	RXD2-RXD3	I	RX Data



Pin Group	Pin #	Pin Name	Type	Description
Supply	1,22, 43,64	GND	P	Ground
	18	VBAT	P	3V battery backup
	60	VCC33	P	3.3V power supply
	61	VCC25	P	2.5V power supply Can be connected to 3.3 V power supply at the cost of slightly higher power consumption.
	63	VCC18	P	1.8V power supply
	65	VCC12	P	1.2V power supply
	72	VCC_RGMII	P	RGMII power supply. Connect to same voltage as the external PHY.

## 4.1 Pin Descriptions

Here are pin descriptions for designing with the module.

### 4.1.1 Analog Control of Oscillator

- (M68) AOUT1: Analog output used for controlling the frequency of the crystal oscillator. If an external voltage controlled frequency source is used, this signal should be used to control it.
- (M64) AOUT1: Analog output used for controlling the frequency of the external voltage controlled frequency source.
- ACH0: Analog input used to measure the control voltage on external TCXO/OCXO.
- PB7: Connected together with PC0 on the module. Intended for use as SPI slave select, interconnection is for backwards compatibility. Only one of the GPIOs should be used to control this pin, the other should be tristated.

### 4.1.2 Write Protection

WP: This pin is connected to the write-protect pin of the on-board flash device. When pulled low or left open, the boot sector of the flash is write-protected. Typically, only pulled high to do initial flash programming during manufacturing.

### 4.1.3 Serial Ports

There are three asynchronous serial ports. The identical ports are named COM1, COM2 and COM3 and are located on pins 24 – 38 of the module.

The serial ports have hardware flow control using RTS/CTS, support several communication options with different combinations of parity, stop bits and character length, and are capable of baud rates from 300 bit/s up to 921,600 bits/s.

### 4.1.4 PF3/LOCK Pin

The PF3/LOCK pin can be configured to indicate that the PTP loop is locked. See the ptp2 lock command.

## 4.2 Clocking

By default the M68/M64 supports an external oscillator of 20 MHz. If another frequency is used, the PLL register (designated Configuration Block Register 4, CRB4) needs to be changed.

The CRB4 register has 3 fields, as shown in the following table:

CRB4 Bits	Description	Allowable Values
Bit 7	PLL_frang	Should be set to 1
Bits 6-2	pll_n - PLL frequency multiplication factor dividend.	0 – sets dividend to 32 1 – sets dividend to 1 2 – sets dividend to 2 : : 31 – sets dividend to 31
Bits 1-0	PLL_m – PLL frequency multiplication factor divisor.	0 – sets divisor to 5 1 – sets divisor to 1 2 – sets divisor to 2 3 – sets divisor to 3

The output frequency is calculated as the external oscillator frequency  $\times$  ( pll\_n / pll\_m). The resulting output frequency must be 150 MHz and the external oscillator frequency multiplied by 4 may not exceed 167 MHz.

The PLL register can be set using the “out crb4 0xNN” command in the system.ini file, and which will be run at powerup.

Example CRB4 setting for 20 MHz external oscillator, “out crb4 0xbe”, which sets pll\_n/pll\_m to 15/2, and the resulting output frequency of 150 MHz.

Example CRB4 setting for 10 MHz external oscillator, “out crb4 0xbd”, which sets pll\_n/pll\_m to 15/1, and the resulting output frequency of 150 MHz.

One important characteristic to remember when using an external oscillator other than 20 MHz: The communication with the M68/M64 might be difficult before the proper PLL settings are in place. For example, when using an external 10 MHz oscillator, the serial port communications would occur at 57600 baud, instead of 115,200.

NOTE: CPUCLK is a 1.8 V level pin. It must not be connected directly to a 3.3 V level oscillator output. A resistor or capacitive voltage divider is enough to ensure that the voltage doesn't exceed 1.8 V.

If the external oscillator has voltage control, it can be connected to the AOUT1 analog output. The control range of this pin is from 0.8 V to 2 V. Fixed-frequency external oscillators can also be used, but then the FREQOUT frequency will not be syntonized.

## 4.3 Power Supply

### 4.3.1 Digital Power Supply

The module has five different power pins. Required power sources are 3.3, 1.8 and 1.2 Volts.

The use of an optional 2.5 V source will slightly decrease the power consumption. Otherwise the VCC25 pin must be connected to VCC33.

The RGMII interface is powered by the user's system and the VCC\_RGMII pin shall be connected to the systems power source. In the case the RGMII interface is not intended to be used, the VCC\_RGMII pin still must be connected to a power source. VCC33 is the best choice.

## 4.4 Time I/O

### 4.4.1 Reference Time Input

It consists of signals PPSIN and TODIN, and is intended for connection to an external time source like a GNSS receiver or similar. If one or both signals are not used, they can be left unconnected.

PPSIN expects a pulse-per-second signal with LVTTTL levels. Rising edge should be on second boundary, the pulse width is not critical.

TODIN should receive time-of-day information, for example in NMEA 0183 format. The levels must be LVTTTL and polarity will be detected. The current FW supports 4800 or 9600 baud 8N1 on the TODIN pin.

### 4.4.2 Precision Time Output

The interface consists of signals PPSOUT, TODOUT, FREQOUT and SYNTFREQ.

PPSOUT outputs a pulse-per-second signal with LVTTTL levels. Pulse width, frequency, phase and polarity of this signal can be controlled by software. For more information, see [18. Appendix 3: PTP Command Reference](#).

TODOUT outputs time-of-day at LVTTTL levels, for example in NMEA 0183 format. Baudrate can be set to 4800 or 9600 baud 8N1. For more information, see [18. Appendix 3: PTP Command Reference](#).

(M64) FREQOUT outputs a syntonized frequency in the MHz range and at LVTTTL levels. The frequency is software selectable 5/10/20/25 MHz. The frequency will only be syntonized (kept in phase with PTP or GNSS time) if the external oscillator is voltage controlled by analog output AOUT1 from the module.

(M68) FREQOUT outputs a syntonized frequency in the MHz range and at LVTTTL levels. The frequency is software selectable 5/10/20/25 MHz. The frequency will only be syntonized (kept in phase with PTP or GNSS time) if the internal crystal oscillator is used or if the external oscillator is voltage controlled by analog output AOUT1 from the module.

The SYNTFREQ is the same as the FREQOUT.

## 5. Specifications

### 5.1 Metrics

Metric	Value
Dimensions	29.2 x 29.2 x 3.3 mm
Weight	3.0 g
Operating temperature	-40°C – +85°C, ambient
Storage temperature	-40 – +150 °C

### 5.2 Absolute Maximum Ratings

Parameter	Symbol	Min	Max
Supply voltage 3.3 V	V <sub>CC33</sub>	-0.3 V	3.6 V
Supply voltage 2.5 V	V <sub>CC25</sub>	-0.3 V	3.6 V
Supply voltage VCC_RGMII	V <sub>CC_RGMII</sub>	-0.3 V	3.6 V
Supply voltage 1.8 V	V <sub>CC18</sub>	-0.3 V	2.5 V
Supply voltage 1.2 V	V <sub>CC12</sub>	-0.3 V	1.32 V
RTC battery backup supply	V <sub>BAT</sub>	-0.3 V	4.0 V
I/O voltage (CPUCLK pin)	V <sub>IO18</sub>	-0.3 V	2.16 V
I/O voltage (all other pins)	V <sub>IO33</sub>	-0.3 V	4.0 V
ESD tolerance (Ethernet differential pairs, human body model)	V <sub>ESDE</sub>		6 kV
ESD tolerance (all other pins, human body model)	V <sub>ESD</sub>		2 kV

Permanent device damage may occur if the absolute maximum ratings are exceeded. These are stress ratings only, and functional operation should be restricted to within the conditions detailed in the next section.

Exposure to absolute maximum rating conditions for extended periods may affect the device's reliability.

### 5.3 Recommended Operating Conditions

Parameter	Symbol	Min	Typ	Max
Supply voltage 3.3 V	V <sub>CC33</sub>	3.15 V	3.3 V	3.45 V
Supply voltage 2.5 V	V <sub>CC25</sub>	2.38 V	2.5 V	2.62 V
Supply voltage VCC_RGMII	V <sub>CC_RGMII</sub>	3.15 V	3.3 V	3.45 V
		2.38 V	2.5 V	2.62 V
		1.71 V	1.8 V	1.89 V
		1.43 V	1.5 V	1.57 V
		1.14 V	1.2 V	1.26 V
Supply voltage 1.8 V	V <sub>CC18</sub>	1.71 V	1.8 V	1.89 V
Supply voltage 1.2 V	V <sub>CC12</sub>	1.14 V	1.2 V	1.26 V
RTC battery backup supply	V <sub>BAT</sub>	2.7 V		3.63 V
I/O voltage (CPUCLK pin)	V <sub>IO18</sub>	0 V		1.98 V

Parameter	Symbol	Min	Typ	Max
I/O voltage (RGMII pins)	V <sub>IO_RGMII</sub>	-0.3 V	V <sub>CC_RGMII</sub>	V <sub>CC_RGMII</sub> + 0.5V
I/O voltage (all other pins)	V <sub>IO33</sub>	0 V		3.63 V

#### 5.4 DC Electrical Characteristics

Parameter	Symbol	Min	Typ	Max
Power consumption				1060 mW
Supply current	I <sub>CCxx</sub>	(see table below)		
RTC backup current (V <sub>BAT</sub> = 3.0 V, V <sub>CC</sub> = 0 V)	I <sub>BAT</sub>		5.5 $\mu$ A	
Input low voltage (except CPUCLK)	V <sub>IL</sub>			0.8 V
Input high voltage (except CPUCLK)	V <sub>IH</sub>	2.0 V		
Input/tristate leakage current	I <sub>I</sub>		1 $\mu$ A	10 $\mu$ A
Output low voltage ( I <sub>OL</sub>   = max)	V <sub>OL</sub>			0.4 V
Output high voltage ( I <sub>OH</sub>   = max)	V <sub>OH</sub>	2.4 V		
Output drive current (GPIO pins)	I <sub>OL</sub>  ,  I <sub>OH</sub>	2/8 mA		

Supply currents per voltage for different choice of host RGMII voltages. These are preliminary values for a gigabit speed connected module:

1.8 V RGMII option	
V	mA
1.2	138
1.8	115
2.5	15
3.3	200

Note: The final power consumption will depend on how the user designs his PCB. The numbers given in the table are maximum values for a well-designed board with a 1.8 V feed of the external RGMII port. Increasing this to 3.3 V is optional, but this will increase the total power consumption.

Note: When planning to use the module in an existing design, the module contains a network side PHY. The existing PHY must be removed, also when calculating power consumption as the PHY consumes the majority of the total.

#### 5.5 AC Electrical Characteristics

Parameter	Symbol	Min	Typ	Max
(M68) On-board main oscillator frequency	f <sub>OSC</sub>		20 MHz	
On-board RTC oscillator frequency	f <sub>RTC</sub>		32768 Hz	
Core & SDRAM frequency (must be typical value for Ethernet operation)	f <sub>CORE</sub>	20 MHz	150 MHz	167 MHz
External oscillator frequency on CPUCLK	f <sub>EXT</sub>	5 MHz		41.7 MHz

## 6. PCB Design Considerations

This chapter should be read before starting a new design.

### 6.1 Land Pattern

See section [2.2 Recommended Land Pattern](#) and Accepted Land Pattern for further information.

### 6.2 Power

The module has five connections for five different voltages. Three of those voltages are mandatory, two power pins can be joined to the cost of slightly increased power dissipation. One voltage is used for an optional PHY. The module also comprises four connections to GND. An optional backup battery, for the RTC, can be connected to the module.

#### 6.2.1 Required Voltage

For Maximum and Operating Voltage Levels see sections [5.2 Absolute Maximum Ratings](#) and [5.3 Recommended Operating Conditions](#).

Maximum Supply Currents during operation are listed in [5.4 DC Electrical Characteristics](#). During start-up, the current consumption can momentarily be higher.

Pin Name	Voltage	Description
VCC33	3.3 V	This voltage is used for driving signals on the module and for the /O's of the module.
VCC25	2.5 V or 3.3 V	If connected to 3.3 V there will be slightly higher power dissipation.
VCC18	1.8 V	Used for the PHY located on the module.
VCC12	1.2 V	Power for the devices on the module.
VCC_RGMII	1.2 V to 3.3 V	If an optional external PHY is used this pin must be connected to the same voltage source as the PHY's RGMII interface. The pin must not be left unconnected. If no external PHY used this pin should be connected to VCC33.  This power pin is used only to drive the external RGMII interface.
VBAT	3 V	If backup is required, this pin shall be connected to a 3 V lithium coin cell battery. No current is drawn from the battery when power is available on pin VCC33.

### 6.2.2 Decoupling Capacitors

It is important that the selected decoupling capacitors are specified correctly, are placed close to the module and, to minimize inductance, are connected with as wide tracks as possible. The use of power planes is of course the ideal solution.

Decoupling capacitors on the module itself are placed very close to the on-module circuits. Those capacitors will provide for the immediate current requirement. Bulk capacitance must be placed external to the module. All 4.7 $\mu$ F capacitors mentioned below shall be placed as close as possible to the corresponding module connection. The larger capacitors, 100 $\mu$ F, can be placed further away from the module.

Pin Name	Decoupling Capacitors
VCC33	Connect one 4.7 $\mu$ F and one 100 $\mu$ F capacitor to this pin.
VCC25	Connect one 4.7 $\mu$ F and one 100 $\mu$ F capacitor to this pin.
VCC18	Connect one 4.7 $\mu$ F and one 100 $\mu$ F capacitor to this pin.
VCC12	Connect two 4.7 $\mu$ F capacitors to this pin.
VCC_RGMII	Connect one 4.7 $\mu$ F and one 100 $\mu$ F capacitor to this pin.

The decoupling capacitors must have the following specifications:

- Ceramic X7R or X5R, 6.3 V.
- 100  $\mu$ F: max size 1210, ESLmax 5nH, 10 mOhm < ESR <60 mOhm.
- 4.7  $\mu$ F: max size 0805, ESLmax 2nH, 10 mOhm < ESR <60 mOhm.

### 6.3 Clock Signal MCLKOUT1

Clock signals must be treated with great care, with signal integrity in mind. For this reason, MCLKOUT1 has a series termination resistor placed on the module, no further termination is required. The trace from the module must not have any stubs. If more than one load, those loads should preferably be placed close to each other. The trace shall preferably have 50-65 ohm line impedance.

This clock signal was intended to drive an external PHY. However, it has been shown that it is not suited for this purpose due to jitter. We suggest using an external 25 MHz oscillator to drive the external PHY instead.

### 6.4 Ethernet Interface

The four pairs of differential signals driving the magnetics are placed close to each other on one side of the module. The length of the traces for each pair shall be as equal as possible and so should the length of the four pairs.

Drivers for the LEDs, signaling connection status, are of Open Collector type. A resistor of nominally 470 ohms shall be placed between VCC33 and the anode side of the LEDs.

#### 6.4.1 Magnetics

The following is a list of magnetics that can be used with the module. The list is not complete, and all magnetics listed are not tested together with the module. All listed magnetics are single port RJ-45 connectors with integrated magnetics of 3 wire choke type.

Manufacturer	Part Number
Link-PP Electronics	LPJK2065AONL
	LPJK0036AINL
Pulse Electronics	JK0-0136NL
	JK0654219
Foxconn	JFM38010-01S1-4F
Belfuse	0813-1X1Y-43
	SI61021F

## 6.5 RGMII Interface

This interface is clocked at 125 MHz, when running GbE. Needless to say great care must be taken during design. Connections to the two channels, TX and RX, are adjacent on the module, having respectively clock in the center of the connections.

RGMII clocking is source synchronous, what is the RX clock is generated by the PHY and the TX clock by the module. All traces within a channel must have the same length. Delay of the clock, sampling in the middle of the data-eye, is taken care of by the external PHY and the RGMII interface on the module.



## 7. Application Software

The M64 and M68 are quite similar. In the CLI commands that follow, the M68 commands are given. The full details of a command are available via help commands in the CLI.

For the M64 the parameter, "port\_number" is always 1. For the M68, "port\_number" is either 1 or 2, the default is 1.

For the M64 the parameter, "iface" is always enet0. For the M68, it is either enet0 or enet1.

### 7.1 System Access

See the Quick Start Guide for setting up communication to the module in a P60 Evaluation Kit. To logon to the system the username is "root", with password "root" as the initial credentials.

You can change them later with the passwd command and add or delete users in the database, see the passwd.ini section.

On entering the system, you will see a prompt and an identifier for the system. The interface is very similar to UNIX.

### 7.2 System Files

The system directory *A:/system* contains firmware files and a number of system initialization and configuration files. These are listed in the Appendix 1 for reference. The files most applications should check and modify are the *system.ini* and *startup.ini* to setup the module in its environment and give the PTP engine the parameters for the synchronization scheme.

### 7.3 Module Shell Environment

The shell, called ISH, is responsible for the high-level system initialization, for execution of various servers and for providing a UNIX-like command interface.

#### 7.3.1 High-Level Initialization

The shell startup code completes initialization and configuration of the system performing the following tasks:

- Initializes the serial port interface
- Registers stdout and stderr functions
- Reads the Hardware Identification String from the flash memory
- Reads and processes a shell configuration file *ish.ini*.
- Reads and processes a system configuration file *system.ini*
- Initializes and starts a Serial server
- Configures a hostname
- Initializes and configures TCP/IP stack software
- Starts FTP and Telnet servers
- Sets the Time Zone and Daylight Saving
- Reads and executes startup file *startup.ini*

#### 7.3.2 System Commands

The shell includes Serial server, Telnet server and FTP server. The shell starts them automatically during startup, if specified in the system configuration file. Otherwise they can be started and stopped manually using *startserver* and *stopserver* commands.

#### 7.3.3 Communication Servers

The shell includes Serial server, Telnet server and FTP server. The shell starts them automatically during startup, if specified in the system configuration file. Otherwise they can be started and stopped manually using *startserver* and *stopserver* commands.

##### 7.3.3.1 Serial Server CLI

The Serial server provides the command-line user interface over a serial channel. The serial port number and port parameters can be specified through environment variables.

### 7.3.3.2 Telnet Server CLI

The Telnet server provides the command-line user interface over a TCP/IP communication channel. The TCP port number and the server priority can be specified through the environment variable.

### 7.3.3.3 SSH Server

The SSH server is not started by default but has to be started manually using the command:

```
startserver -d
```

Note: For the current implementation, initialization takes about 30 seconds and establishing the connection about another 60 seconds.

### 7.3.3.4 FTP Server

The FTP server provides a remote access to the local file system. It supports both get and put operations. The number of simultaneous connections and the server priority can be specified through environment variables.

### 7.3.3.5 JSON

The command interface (default is COM3) can be used in either CLI/HMI command mode or JSON mode. The JSON mode is turned on or off with the following commands:

```
ptp2 engine json on
```

```
ptp2 engine json off
```

See [12. JSON Usage](#) for more details.

## 7.4 Passthrough

The passthrough mode allows a system with two Ethernet interfaces to act as a "synchronization gateway" for non-PTP hosts.

On its primary Ethernet port such "gateway" is connected to a network which transports data and PTP synchronization. To its secondary Ethernet port, the non-PTP system is connected. The "gateway" participates in PTP activity acting as a PTP slave and synchronizing its local clock. The PTP traffic is filtered out while non-PTP data is passed through to the host along with PPS/TOD/FREQ-out synchronization signals.

The command to turn on and off pass-through is:

```
ifconfig frwd [on|off]
```

On the M68 the secondary (host-side), Ethernet is available for user or applications such as a general-purpose network interface.

On M64, the secondary (host-side), Ethernet interface is not available for user or applications as a general-purpose network interface, it is used only for pass-through and it is completely transparent from either host or network side.

## 7.5 VLAN

The M68/M64 supports VLAN tagging. If VLAN is used in combination with PTP, the VLAN needs to be set before the PTP engine is started.

To enable VLAN on the interface, use the command:

```
ifconfig [iface] vlan [parameters]
```

where [iface] is the interface name. If not given, 'enet0' will be used.

[parameters] can be

off - send and receive untagged frames only

none - send untagged frames, receive untagged and priority tagged frames

0 [prio\_code] - send priority-tagged frames, receive untagged and priority tagged frames

<vlan\_id> [prio\_code] - send and receive vlan-tagged frames only

For example:

```
ifconfig vlan 1588 0
```

To disable VLAN on the interface, use the command:

```
ifconfig enet1 vlan off
```

## 7.6 MDIO

The M68/M64 supports the possibility to control and monitor equipment such as a PHY connected over MDIO. To enable MDIO mode, use the command

```
ifconfig mdio
```

This will disable all other access to the MDIO registers to avoid conflicting writes to the page register. To then read from MDIO registers use the command

```
<reg>
```

And to write to a MDIO register in the exclusive MDIO mode use

```
<reg> <value>
```

To exit from the exclusive MDIO mode, type “q”.

For example, to enter MDIO mode and write 0x4885 to register 18 on page 3 (reg 22 is page register):

```
A:/root> ifconfig mdio
Enter exclusive MDIO mode on enet0
22 3
22: 0x0003
18
18: 0x4b85
18 0x4885
18: 0x4885
q
Quit exclusive MDIO mode on enet0
A:/root>
```

It is also possible to read or write directly to a MDIO register without entering the exclusive mode:

```
ifconfig mdio <reg> <value>
```

but beware any page register settings.

To simplify for the user, some common operations on the PHY has been defined as specific commands: to manage link speed and duplex, to reset the PHY and to manage SyncE master/slave behavior.

## 7.7 Link Speed

The PHY link speed and duplex can be controlled by the command

```
ifconfig [iface] link [parameters]
```

where [iface] is the interface name. If not given, 'enet0' will be used.

[parameters] can be one of the following modes: 1000F, 1000H, 100F, 100H, 10F, 10H which disables autonegotiation and force the interface into the desired speed and duplex (F – full, H – half) or auto [mode\_mask] which will enable autonegotiation and optionally set advertised modes. If mode\_mask is not given all supported modes will be advertised, i.e. 1000F|1000H|100F|100H|10F|10H

For example, to set 100Mb/s Full duplex:

```
ifconfig link 100F
```

or set interface 0 to autonegotiation with only 1000Mb/s and 100Mb/s full duplex supported.

```
Ifconfig enet0 link auto 1000F|100F
```

## 7.8 PHY Reset

It is possible to reset the PHY with the following command:

```
ifconfig [iface] reset [options]
```

Where [iface] is interface name. If not given, 'enet0' will be used.

[options] can be either sw, noint or not given at all.

If no option is given, a HW reset is performed, i.e. the reset pin of the PHY is pulled after which initialization is done.

If the option “noint” is given also a HW reset is performed, but no initialization of the PHY is performed. This can be used when PHY needs to be configured manually using MDIO.

The option “sw” sets the reset bit in the MDIO registers, performing a SW reset. This is of course only relevant if the PHY's MDIO interface lines are connected.

For example, to perform a HW reset of the PHY:

```
ifconfig reset
```

## 7.9 SyncE

For PHY devices supporting SyncE it is possible to define the direction of frequency on a link. This is only valid for 1000Base-T interfaces.

```
ifconfig [iface] synce [parameters]
```

Where [iface] is the interface name. If not given, 'enet0' will be used.

[parameters] can be

- master|slave which will disable autonegotiation and force the interface into either master or slave mode or
- auto master|slave which will enable autonegotiation with either master or slave preference

If no parameter is given, current resolved state will be reported.

For example, setting the module to be preferred SyncE master:

```
ifconfig synce auto master
```

## 7.10 SFP

There is support for managing an SFP connected to the specified network interface.

```
ifconfig [iface] sfp [parameters]
```

Where [iface] is the interface name. If not given, 'enet0' will be used.

[parameters] can be

raw, which prints SFP information in raw hex format

state, which shows state of LOS, enable, presence and fault

enable will enable the SFP transmitter

disable will disable the SFP transmitter

mdio <reg> will perform single read from MDIO register in SFP PHY

mdio <reg> <value> will perform single write to MDIO register in SFP PHY

If no parameter is given, prints formatted SFP information.

## 8. IEEE 1588/PTP Engine

The M64 and M68 are quite similar. In the CLI commands that follow, the M68 commands are given. The full details of a command are available via help commands in the CLI.

For the M64 the parameter, "port\_number" is always 1. For the M68, "port\_number" is either 1 or 2, the default is 1.

For the M64 the parameter, "iface" is always enet0. For the M68, it is either enet0 or enet1.

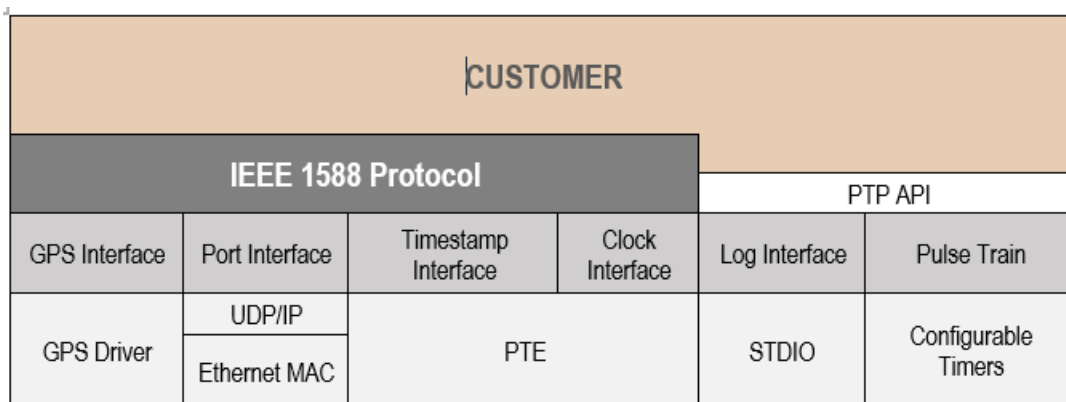
### 8.1 General Features

The IEEE 1588/PTP Engine (PTP Engine) has the following features:

- Implements an ordinary clock in accordance with IEEE Std 1588™-2008
- M68 implements boundary clock in accordance with IEEE Std 1588™-2008
- Fully flexible to comply with Default Telecom and Power Profiles
- Supports both GNSS and PTP as the primary and backup time source
- Built-in SNTP server
- Supports 5 different operational modes
- Provides the command-line interface (can be configured and monitored either locally via Terminal or over the network via Telnet or SSL)
- Supports PTP Management interface (can be configured and monitored over the network from a PTP Management station)
- Provides a rich set of APIs
- Uses high-precision hardware timestamp engine
- Can be partially customized by the customer

### 8.2 PTP Software Implementation

PTP software is divided into three layers; Protocol Layer, OS Abstraction Layer and the OS Layer.



**Figure 8.1. PTP Layers**

#### 8.2.1 Protocol Layer

The protocol layer contains the PTP engine which is implemented according to the IEEE 1588-2008 standard. The protocol engine encapsulates a number of functions necessary to run the PTP clock. Among those functions are best master clock algorithm, PTP message handling and maintaining of PTP clock data sets.

#### 8.2.2 OS Abstraction Layer

The OS abstraction layer implements a number of interfaces between the protocol engine and the underlying operating system.

##### 8.2.2.1 GNSS Interface

The GNSS interface contains functionality for GNSS synchronization. This interface is used when running the grandmaster clock.

### 8.2.2.2 PTP Interface

The interface PTP encapsulates the functions of the PTP environment and therefore only provides the functions that must be callable from the outside of the protocol engine. PTP is a wrapper of PTP environment – it represents the API of the PTP protocol.

### 8.2.2.3 Port Interface

The port interface is used to send and receive PTP messages. It makes use of the functions the OS provides (e.g. sockets) in order to communicate over the network.

### 8.2.2.4 Timestamp Interface

The time stamp interface is used to pass time stamps for Sync and Delay\_Req messages to the PTP protocol engine. It also implements the connection which is used to read the timestamps from the timestamp unit.

### 8.2.2.5 Clock Interface

The clock interface provides functions to read, write and configure the local clock. The clock interface uses the clock driver functions (set time, get time, drift compensation etc.) in order to access the clock.

### 8.2.2.6 Log Interface

Through the log interface the protocol engine can output log information. The log provides human readable information that shows current settings and current state of the protocol engine. This log information can be monitored either via a serial port or TCP/IP.

Log output is initiated and controlled through the Human Machine Interface.

## 8.3 GNSS Driver

The GNSS driver consists of a PPS Receiver unit and a GNSS Time Receiver.

When a PPS signal from the attached GNSS unit is received by the GNSS Receiver, the signal is timestamped by the PTP timestamp unit. The timestamp is then delivered to the GNSS Interface where it is put in an event queue for processing by the protocol stack.

The GNSS Time Receiver receives time of day information from the GNSS unit. This time information specifies the date and time of the PPS signals.

## 8.4 AccuTime Software

AccuTime software consists of configurable timers and MAC hardware with timestamp capabilities.

The MAC hardware reports timestamps for all incoming or outgoing Ethernet frames.

The PTP engine receives and sends PTP messages through the TCP/IP stack using a socket interface. It accesses the timestamp list to get timestamps for these messages.

The MAC hardware is augmented with input timestampers and output coincidence triggers to allow it to set timestamps at input events and cause output events at certain times.

## 8.5 PTP API

The PTP engine provides a rich set of C API calls that the application programmer can use to start and control the engine. This C API is only open to customers which have the Developer IDE software installed.

### 8.5.1 Starting/Stopping the PTP Engine

The PTP engine runs as a separate thread and there can only be one instance of the PTP engine running at one time. An application can start the PTP engine by a call to the function `PTP2_Start()`.

As the parameter to that function, a pointer to the configuration structure is passed. In this structure one can specify whether a pulse-train needs to be enabled and a mode for the PTP engine operations (see detail below).

The PTP engine can also be automatically started at boot time by the Shell, ISH. For more information about ISH see the ISH documentation in the Profile Help.

To stop the PTP engine, use the `PTP2_Stop()` function or the corresponding shell command.

### 8.5.2 Pulse Train

A pulse-train can be generated by the PTP engine. The pulse-train can be continuous or a specified number of pulses can be generated. The user can also specify the width and period of the pulse-train as well as the starting time for the pulse-train.



### 8.5.3 Human Machine Interface

The PTP engine has a command interface that the user can use to manually change settings and issue different printouts from the engine's current state. The Human Machine Interface (HMI) commands can be entered either manually (from a Terminal or Telnet console) or issued by the application with the `PTP2_hmi()` function.

To find exact syntax of each command simply type (or send from a program) "h" – stands for Help.

When using the HMI commands, you can type either the command word or use the letter(s) in the parentheses before the command. For example, to disable unicast negotiation, type `unicast n off` or type `u n off`. To see a list of the command set, first make sure you are in HMI mode by typing `ptp2 hmi` at the prompt. Next, type `help` and the commands display.

The following table shows the structure of the commands

Command	Example	Results
command	<code>unicast or u</code>	Prints a list of subcommands.
Command subcommand	<code>unicast n or u n</code>	Prints current value of associated parameter.
Command subcommand value	<code>unicast n off or u n off</code>	Sets associated parameter to a different value.

The next series of tables describe each command and its associated subcommands and values. The last table shows command examples.

Clock Command	Subcommand & Parameters	Description
(c)clock	(i)info	Print clock description.
	(u)user <string>	Change user description.
	(f)profile <default g8265 g8275.1 g8275.2 power [v2]>	Change clock profile.
	(m)mode <oc bc>	Change clock type mode (M68 only)
	(t)twostep <on off>	Change two-step mode.
	(q)quality <class> <accuracy> <variance>	Change clock quality.
	(p)priority 1 2 <value>	Change priority 1 or 2.
	(d)domain <number>	Change domain number.
	(s)slaveonly <on off>	Change slave-only mode.

Port Command	Subcommand & Parameters	Description
(p)port <port_number>	(s)state	Print port state.
	(m)mechanism <e2e p2p disable>	Change or disable delay mechanism.
	(p)protocol <udp eth udp6>	Change network protocol.
	(c)compatibility <on off auto [timeout]>	Change V1 HW compatibility mode.
	(a)asymmetry <value>	Change delay asymmetry.
	(si)sync <value>	Change log sync interval.
	(di)delay <value>	Change min log delay request interval.
	(pi)pdelay <value>	Change min log pdelay request interval.
	(ai)announce <value>	Change log announce interval.
	(rt)receipt <value>	Change the announce receipt timeout.
	(u)unicast <master slave both disable>	Enable/disable unicast operation for the master and slave clocks.
	(f)profile <default g8265 g8275.1 g8275.2 power [v2]>	Change PTP profile.
	(d)dscp <value>	Set DSCP in PTP messages.

Unicast Command	Subcommand & Parameters	Description
(u)unicast <port_number>	(s)status	Print port unicast information.
	(n)negotiation <on off>	Enable/disable unicast negotiation.
	node (m)master [add del] <address> [clock_id] <priority> <si x> <di x> [domain]	Change unicast master discovery table.
	node (s)slave [add del] <address> <ai x> <si x> <di x>	Change static unicast slave table.
	(q)query <interval>	Change unicast discovery query interval.
	(d)duration <value>	Change unicast transmission duration.
	(t)timeout <value>	Change cancel acknowledge receipt timeout.
	(l)limit <value>	Change total unicast messages rate limit.
	(f)filter (m)master [add del] <address *> [clock_id] [x   [priority] [si] [di x]]	Change acceptable unicast master table.
	(f)filter (s)slave [add del] <address *> [clock_id] [x   [ai] [si] [di x]]	Change acceptable unicast slave table.

Power Profile Command	Subcommand & Parameters	Description
(pw)power	(g)gmid <0..65535>	Change grandmaster ID.
	(i)inacc <grandmaster> <network>	Change grandmaster and network time inaccuracy.

G.8275 Profile Command	Subcommand & Parameters	Description
(g2)g8275	<port_number> (p)priority <1..255>	Change clock local priority.
	(s)steps <1..255>	Change max steps removed.
	<port_number> (m)masteronly <on off>	Change port master-only attribute

Dataset Command	Subcommand & Parameters	Description
(d)dataset	(d)default	Print default dataset.
	(c)current	Print current dataset.
	(m)parent master	Print parent dataset.
	(t)time	Print time properties dataset.
	(p)port	Print port dataset.
	(f)foreign	Print foreign master dataset.
	(u)unicast	Print unicast master or slave node dataset.
	(pw)power	Print power profile dataset.
	(g1)g8265	Print G.8265 profile dataset.
	(g2)g8275	Print G.8275 profile default dataset.
	(g2)g8275 <port_number>	Print G.8275 profile port dataset.

Time Command	Subcommand & Parameters	Description
(t)time	(i)info	Print time and synchronization information.
	(s)sync	Print synchronization status.
	arb <time>	Print or set time in +/- SSSSSSSSS.NNNNNNNNN format.
	ptp <time>	Print or set time in YYYY-MM-DD HH:MM:SS.NNNNNNNNN format.
	utc <time>	Print or set time in UTC YYYY-MM-DD HH:MM:SS.NNNNNNNNN format.
	ntp <server>	Print or set time from NTP server specified by name or IP address.
	(t)timescale <ptp arb>	Change timescale.
	(o)offset <value>	Change UTC offset.
	(l)leap <save load>	Print UTC leap table or save/load it into/from file.
	(l)leap 59 61 <0 1>	Change leap seconds flags.
	(u)update <on off>	Turn on/off automatic update of RTC time from PTP time.
	(h)holdover <value>	Change period of holdover state.

GPS Command	Subcommand & Parameters	Description
(g)gps	(s)status	Print GPS interface information.
	(n)nmea	Print NMEA position/satellites information.
	(p)ppsin <enable disable>	Enable/disable PPS input reception.
	(t)todin <enable disable>	Enable/disable TOD input reception.
	(c)class <active> <holdover> <free>	Change clock state to clock class mapping.
	(t)todin <required optional>	Change non-present or invalid TOD-in handling.
	(d)delay <value>	Change PPS-in cable delay compensation.
	(f)format <nmea ascii cm>	Change TOD-in message format.
	(v)void <ignore syntonize holdover>	Changes NMEA RMC void flag handling.
	(i)interface <default com[n]> <baudrate>	Change TOD-in interface.
	(m)message <string>	Send message to GPS receiver via TOD interface.
	cm <status> <class> <accuracy> <variance> <source> <tt> <ft>	Change CM status to clock properties mapping.

Pulsetime Command	Subcommand & Parameters	Description
(pt)pulsetime	(s)status	Print PPS-TOD status and configuration.
	(p)pulse <period> <width>	Change PPS-out pulse parameters.
	(d)delay <value>	Change TOD-out delay after PPS-out rising edge.
	(f)format <nmea [rmc zda both] ascii [utc local] cm custom>	Change TOD-out message format.
	(v)void <never free holdover>	Change NMEA RMC Active/Void flag behavior.
	(z)zone <none tz dst>	Change NMEA ZDA timezone/daylight field.
	cm <class> <status>	Change clock class to CM status mapping.
	(i)interface <default com[n]> <baudrate>	Change TOD-out interface.
	(t)todout <enable disable>	Enable/disable TOD output.
	(m)mode <1 2 3>	Change PPS-TOD output mode (sync, holdover, always)  1= sync 2= holdover 3= always
	(e)error <value>	Change error tolerance.
	(a)autorestart <on off>	Turn on/off automatic restart.
	start <time> <count>	Start/schedule PPS-TOD output.
	stop	Stop/cancel PPS-TOD output.

Nettime Command	Subcommand & Parameters	Description
(nt)nettime	(s)status	Print SNTP server status and configuration.
	(m)mode <1..4>	Change server mode (unicast, multicast, broadcast, and mixed).  1= unicast 2= multicast 3= broadcast 4= mixed
	(pi)poll <0..64>	Change poll/broadcast interval.
	start	Start SNTP server.
	stop	Stop SNTP server.

Engine Command	Subcommand & Parameters	Description
(e)engine	(i)init default nvm	Reinitialize clock. Currently not supported.
	(p)port <enable disable>	Change port operational status.
	(e)esmc <auto 0..15 off>	Enable ESMC message transmission.
	(e)esmc map <class> <ssm>	Change clock class to SSM code mapping.
	(n)nvm save reset	Change configuration in non-volatile storage. Currently not supported.
	(d)debug <on off>	Turn on/off debug output.
	(v)verbose <level>	Change verbose level for debug output.
	(m)monitor <on com[n] off>	Enable output for PTP monitor program.
	(f)freqout <5 10 20 25>	Change frequency output (5, 10, 20, 25 MHz).
	(a)asymmetry <value>	Change PHY asymmetry compensation.

### Examples

Function	Command
Set the sync rate to -6	p si -6
Set the delay rate to -6	p di -6
Start PPS	pt start
Print list of clock subcommands	c
Print value of priority1	c p 1
Set value of priority1 to 130.	c p 1 130
Print value of log sync interval.	p si
Set value of log sync interval to -2 (4 sync per sec).	p si -2
Print content of parent dataset.	d m
Print content of port dataset.	d p
Print list of debug output levels.	e v
Enable clock monitor program on COM3.	e m 3

## 9. Network Subsystem

The Network Subsystem, as a part of the System Software is responsible for the networking support and network communications. Its major part is TCP/IP protocol stack software, which provides the support for TCP/IP environment and TCP/IP communications over Ethernet and serial links.

The TCP/IP stack software was designed to be fully reentrant and multithreaded safe, as well as scalable and highly configurable. It is compliant with standards and has a rich application programming interface (API).

### 9.1 TCP/IP Stack Architecture

In general, functional architecture of the TCP/IP stack is the same as in any other TCP/IP implementations. It comprises four major layers: link layer, network layer, transport layer and application layer. These layers interact with each other via well-defined interfaces.

A Link layer provides an abstract view of a network interface device (or adapter) and isolates its details from upper layers. Link layer includes device drivers for real or virtual (pseudo) hardware and implementations of underlying physical network protocols. Currently, TCP/IP link layer software provides following components: driver for internal loopback adapter (pseudo-device), driver for the Ethernet Adapter and implementations of loopback and Ethernet II protocols.

A Network layer provides the basic packet transmission service as well as addressing and routing services. It uses the link layer interfaces to communicate with network devices. The TCP/IP network layer software includes an implementation of Internet Protocol (IP), as the major protocol for packet transmission, and an implementation of Address Resolution Protocol (ARP) as the protocol for the remote host's hardware address discovery. Note that architecturally ARP is implemented as the network layer module, while it's functionally belongs rather to the link layer than to the network layer.

A Transport layer provides for applications a number of various transport services to exchange data over network. It uses network layer interfaces to request an address, routing and control information, a packet transmission and so on. The most common transports are: a datagram transport, which provides mechanisms for unreliable datagram exchange, and a stream transport, which provides reliable and sequential data transfer. The TCP/IP software implements User Datagram Protocol (UDP) as datagram transport and Transmission Control Protocol (TCP) as stream transport. It also implements Internet Control Message Protocol (ICMP) and Internet Group Management Protocol (IGMP) as "raw" transport modules. Any application can send and receive data in ICMP and IGMP format. An additional transport module called "raw wildcard" transport allows applications to access packet transmission services of the IP module, i.e. to send and receive raw IP datagrams.

An Application layer provides unified access to the transport layer features, independently of underlying transport protocol's semantics and hides each protocol implementation details. The TCP/IP software includes reach protocol-independent application programming interface, compatible with BSD sockets. All network applications using this API also belong to the application layer.

### 9.2 Application Layer Components

The TCP/IP software also includes a number of application layer components. They implement application layer protocols which are parts of a TCP/IP suite, such as Dynamic Host Configuration Protocol and Domain Name System.

#### 9.2.1 DHCP Client

The DHCP Client plays an important role in the system initialization. To operate in the TCP/IP environment the system needs to know its IP address, network mask and, optionally, a default gateway. This information can be configured manually or distributed through the DHCP server. So, the DHCP client dynamically obtains these parameters from the DHCP server and automatically configures TCP/IP stack software thus eliminating the need for manual configuration. In the absence of a DHCP server on the network the system is connected to, the DHCP client uses a so-called Zero-Config protocol to select an IP address from a well-known address range and negotiate this selection with other nodes on the network.

#### 9.2.2 DNS Client

Any node on the TCP/IP network is identified by its unique IP address. But it is more convenient to use for identification a human-readable name instead of an IP address. The node name to the node address translation is performed by a DNS client. In order to perform this translation the DNS client interacts with one or more DNS servers, which store and maintain a distributed global name-to-address mapping database.

### 9.3 BSD Sockets API

An application can access communication facilities of the network subsystem through the well-known BSD Sockets application programming interface. This interface is based on the concept of a socket – a communication endpoint – which is quite similar to the file descriptor.

## 9.4 Network Initialization

Before communication facilities can be used, the network subsystem must be initialized and configured. Normally it is done during the initialization of the shell environment. The shell startup code reads configuration files and initializes the TCP/IP stack software. When the target system does not require the shell to be running it can initialize and configure the network subsystem by using a network initialization routine. It will do the following:

- If TCP/IP configuration parameters were not provided, it tries to read them from a permanent storage such as on-board flash memory.
- It initializes the link layer by calling each network interface driver's entry point.
- It initializes the network layer by invoking IP and ARP initialization routines.
- It initializes the transport layer by calling each transport module initialization routine.
- It configures the internal loopback interface.
- It either configures the Ethernet interface manually if configuration parameters were specified, or requests the DHCP client to configure it dynamically.
- Finally it sets the hostname and parameters for a name resolution client.

This routine has to be called after the OS has been initialized and started.

## 10. Using the AccuTime Software

The M64 and M68 are quite similar. In the CLI commands that follow, the M68 commands are given. The full details of a command are available via help commands in the CLI.

For the M64, the parameter "port\_number" is always 1. For the M68, "port\_number" is either 1 or 2, the default is 1.

For the M64, the parameter "iface" is always enet0. For the M68, it is either enet0 or enet1.

### 10.1 PTP Engine Modes

The PTP Engine is capable of operating in five different modes. The operational mode defines the functionality supported by the engine, PTP clock behavior and its properties in different operating conditions. The operational mode is specified upon startup and can only be changed by restarting the engine.

- Mode 0 – PTP-only mode;
- Mode 1 – GNSS-only mode;
- Mode 2 – GNSS-primary, PTP-secondary;
- Mode 3 – PTP-primary, GNSS-secondary;
- Mode 4 – PTP-only, GNSS-debug;

In addition to Mode 0-4 a Slave-only mode is supported. The clock can be switched to the slave-only mode and back at any time and from any of above operational modes.

#### 10.1.1 Mode 0

This is an ordinary PTP master-slave mode. The GNSS interface is disabled.

In this mode the clock normally acts as a PTP slave, but may also become a PTP master if no better clock exists on the network.

The clock class is initialized to DEFAULT (248) and is not changed by the engine while operating. So the manual control via HMI commands (see clock quality command) or API is entirely possible.

#### 10.1.2 Mode 1

This is the default mode. The clock is GNSS-clock and the GNSS is the only source of synchronization. It can never become a slave to another clock regardless of its clock class.

In this mode the clock class is automatically controlled by the engine. The clock is initialized with class DEFAULT (248) and as soon as the engine detects the presence of a stable GNSS signal it raises the class to PRC\_SYNC (6) or APP\_SYNC (13). If only the PPS-input signal is available then the class is selected depending on the configured timescale (see time scale command). If the TOD-input signal is available as well then the timescale is automatically switched to PTP and the clock class is PRC\_SYNC (6).

If later the GNSS-signal is lost the clock is switched to the holdover mode and lowers its class to PRC\_HOLDOVER (7) or APP\_HOLD-OVER (14). If after the holdover period the GNSS-signal is still not available the clock downgrades its class PRC\_DEGRADATION\_A (52) or APP\_DEGRADATION\_A (58) and stays as the PTP master in the free-running mode. If a better clock exists on the network the clock will switch to the PTP passive state.

#### 10.1.3 Mode 2

This mode is almost the same as Mode 1, but after the holdover interval the clock degrades its class to PRC\_DEGRADATION\_B (187) or APP\_DEGRADATION\_B (193), so it can potentially become a PTP slave if a better clock appears on the network.

It means that the clock has the GNSS-signal as its primary source of synchronization and the PTP as a backup source, i.e. when no GNSS-signal present.

#### 10.1.4 Mode 3

This mode is designed for unstable GNSS-reception environments, where the node having a better signal reception becomes a PTP master and all others become PTP slaves, even if they have their own GNSS-signal.

The clock is initialized with class DEFAULT (248) and the class is not changed by the engine while operating. Instead after detecting the stable GNSS-signal the engine increases the priority2 member of the Default Dataset (lowers its value) by some small margin, which might depend on the reception quality. That clock which has a higher priority2 (better GNSS signal reception) becomes the PTP master on the network and all others synchronize with it.



### 10.1.5 Mode 4

This mode is identical to Mode 0 in respect to the clock's behavior. But in this mode the GNSS interface is initialized and the PPS-input signal if present can be used to measure the accuracy of the clock being synchronized over the PTP network.

### 10.1.6 Slave-Only Mode

In this mode the clock can act as a PTP slave only. If there is no PTP master on the network the clock will stay in the listening state waiting for the master to appear.

When the clock is switched to this mode (see clock slave command) the engine changes the clock class to SLAVEONLY (255) and the clock accuracy to UNKNOWN (0xFE). Both values are then restored to the previous state when the clock switched back to its normal mode. If any manual or automatic changes to the clock quality were done while in the slave-only mode, those changes are remembered and come to effect as soon as the clock is back to normal mode.

## 10.2 GNSS Interface

The module supports the GNSS interface outputs PPS and TOD as input signals.

Both of these signals are enabled by default but either of them can be disabled at any time (this might be useful for test purposes).

The reception of the PPS input signal is only possible from a dedicated PPS\_IN pin, while the TOD input signal can be received either from a dedicated TOD\_IN pin (4800 or 9600 bauds) or from a COM-port (any available baud rate).

Three message formats are currently supported for TOD input: NMEA (ZDA and RMC), ASCII and China Mobile.

The current status and configuration of GNSS interface can be viewed and altered through the HMI interface (see GNSS set of commands).

## 10.3 PTP Clock States

The PTP clock at any instance of time can be in one of four following states:

- FREE
- SYNTONIZING
- SYNCHRONIZING
- HOLDOVER

In PTP modes 1-3 the clock class mapping for the FREE and HOLDOVER states can be changed using the GNSS class command.

### 10.3.1 FREE State

The PTP clock comes into this state upon initialization.

Its time is taken from the system RTC, the clock class is DEFAULT (248), clock accuracy is UNKNOWN (0xFE).

The timescale is PTP, the UTC offset is initially set to 36 sec, leap flags are FALSE and the time source is INTERNAL\_OSCILLATOR.

The clock time and all clock properties can be changed manually in this state.

### 10.3.2 SYNTONIZING State

This state is only possible when the PPS-input signal from the GNSS interface is available, but not the TOD-input signal and the clock becomes a PTP master.

When the PTP Engine is running in Mode 1 and 2 the clock class is automatically changed to either PRC\_SYNC (6) or APP\_SYNC (13) according to timescale parameter and the clock accuracy is set to WITHIN\_100\_NS. In Mode 3 the clock quality remains unchanged.

The frequency traceable flag is set to TRUE and the time traceable flag is set to FALSE. The time source remains unchanged.

Manual change of the timescale (see time scale command) will cause the corresponding change in the clock class. Changing the time of the clock (see time ptp/arb/ntp commands) will cause the time source to become HAND\_SET. The UTC offset (see time offset command) and leap flags (see time leap command) are also changeable.

As soon as TOD-input signal becomes available the clock switches to SYNCHRONIZING state.

### 10.3.3 SYNCHRONIZING State

The PTP clock comes into this state when it starts to synchronize its time and frequency with either a PTP or GNSS source.

If the synchronization source is the GNSS, then both frequency traceable and time traceable flags are set to TRUE. The timescale is changed to PTP, the clock class is changed to PRC\_SYNC (6) and the clock accuracy is set to WITHIN\_100\_NS. The time source is GNSS.

Manual changes of clock time and timescale are not allowed, while it is still possible to change the UTC offset and leap flags.

If the TOD-input signal becomes unavailable, while the PPS-input still presents the clock switches to SYNTONIZED state.

If the synchronization source is a PTP master, then the clock quality remains untouched. The timescale is set according to what is distributed by the PTP master. If the timescale distributed is PTP, then the UTC offset (if valid) and leap flags are also set to master's values. The time source is set to PTP.

Manual changes of clock time and timescale are not allowed. If the timescale distributed is PTP, changing the UTC offset (if valid) and leap flags are not allowed as well.

### 10.3.4 HOLDOVER State

The clock enters this state when the synchronization source is lost.

If the clock was synchronized with PTP master its quality remains unchanged. Otherwise the clock class is modified according the engine's operational mode (see above) and the clock accuracy is changed based on the time spent in the holdover state.

There is a static parameter which defines the clock stability. Currently it is fixed to 1 ns/s for a temperature-stable environment. During the holdover state an estimated error value is calculated and the clock accuracy is set according to that value.

The maximum time the clock stays in holdover state is defined by a holdover interval – a configurable parameter expressed in seconds (see time holdover command). By default it is set to 1000 seconds which gives about 1 microsecond error at the end of holdover interval. After holdover interval expires, the clock switches to the FREE state, and its accuracy is reset to UNKNOWN (0xFE).

Manual change of the clock time will terminate the holdover and bring the clock to the FREE state.

## 10.4 PPS/TOD Output

The module is able to provide an external synchronization signal in form of PPS and TOD output.

The PPS/TOD-output signal properties are defined by a number of configurable parameters:

- **PPS period** – defines the interval between pulses. Expressed in microseconds, minimum value is 500 us.
- **PPS width** – defines the length of high phase of the pulse. Expressed in microseconds, minimum value is 250 us, maximum value is (pulse period – 250 us).
- **TOD enable** – defines whether the TOD-output signal should be generated in addition to PPS-output.
- **TOD delay** – defines the delay after the rising edge of the pulse before starting to output TOD-message. Expressed in milliseconds.
- **TOD format** – defines the format of the TOD-message. Supported formats are NMEA (ZDA and RMC), ASCII and China Mobile.
- **TOD interface** – defines which interface to use to output TOD-message. A dedicated TOD\_OUT pin (only 4800 and 9600 bauds) and COM-ports (any available baud rate) are supported.
- **Mode** – defines the conditions in which PPS/TOD-output signal is generated.
- **Error** – defines error tolerance, expressed in nanoseconds.
- **Autorestart** – restart PPS/TOD-output automatically.

There are three modes of PPS/TOD output:

**SYNC** – in this mode the signal is generated only when the clock is either in SYNCHRONIZING or in SYNTONIZING state. As soon as the clock exits this state the signal is stopped. If autorestart is enabled then the signal is started again as soon as synchronization is reacquired.

**HOLDOVER** – prolongs the SYNC mode to the HOLDOVER state. I.e. the signal is generated in that state as well, but stopped as soon as holdover mode is over because of any reason (holdover interval is expired or time is manually changed).

**ALWAYS** – generates signal in any clock state.

In addition to mode parameter, the error tolerance affects PPS/TOS-output signal as well:

- If the mode is ALWAYS this parameter is simply ignored. For other two modes the error tolerance has the following effects:
- If this parameter is zero, then the signal is started/stopped as soon as the clock enters/leaves corresponding states.
- If this parameter is non-zero, and the clock is in the HOLDOVER state the signal is stopped as soon as either the estimated error becomes larger than the error tolerance or the clock exits the HOLDOVER state.

If this parameter is non-zero and the clock is in SYNCHRONIZING or SYNTONIZING state, the signal is started when the clock is synchronized with an error below the error tolerance and stopped when the error increases above the error tolerance.

The initialization defaults are:

1PPS/100 ms, TOD-enable, 0, NMEA, TOD\_OUT/4800, ALWAYS, 0, TRUE.

When the signal is scheduled using the HMI interface (see pulsetime start command), the start time (TAI) and the pulse count may optionally be specified. Without these parameters the start time is 0 (1970-01-01 00:00:00) and the number of pulses is unlimited.

## 10.5 Managing the Two Interfaces (M68 Only)

M68 can support PTP on both its Ethernet interfaces. To distinguish between the two ports in PTP commands related to the port or to unicast, use 1 for the eth0 interface and 2 for the eth1 interface. For example, \_ ptp2 port 2 state to check the state on port 2 (eth1). If no port identifier is set, port 1 is used as default.

## 10.6 Unicast Operations

The PTP Engine supports unicast operations on its port.

The PTP port operating in unicast mode can be in either of three states:

1. Unicast master
2. Unicast slave
3. Unicast both

By default, unicast operations are disabled and the PTP port operates in multicast mode. After unicast is enabled, no multicast communications are possible.

The PTP port can be switched to unicast operations and back at any time using the following HMI command:

```
(p)port <port_number> (u)unicast <master|slave|both|disable>
```

### 10.6.1 Unicast Master

A PTP port in unicast master state can support the following:

- Slave nodes which can dynamically request unicast message transmission services from the master using the unicast negotiation mechanism, as well as
- Slave nodes which do not support the unicast negotiation and simply rely on the reception of unicast messages from the master.

By default, no requests are accepted from and no messages are sent to any slave nodes.

To accept unicast negotiation requests from slave nodes the master needs to be configured as follows:

- Unicast negotiation must be enabled,
- Slave acceptance filter must be populated.

To provide message transmission services to slave nodes which do not support the unicast negotiation the master needs to be manually configured with the list of static slave nodes.

#### 10.6.1.1 Enabling Unicast Negotiation

The unicast negotiation state is controlled by the following HMI command:

```
(u)unicast <port_number> (n)negotiation <on|off>
```

When unicast negotiation is enabled the master accepts unicast transmission requests from negotiation-capable nodes. If a node is allowed by the acceptance filter and if enough resources are available the master grants message transmission services to that node.

If unicast negotiation is disabled no new requests are accepted, but all existing grants remain serviced until they are either expired or cancelled.

### 10.6.1.2 Maintaining the Slave Acceptance Filter

The slave acceptance filter is a mechanism to control which slave nodes may obtain unicast services from the master. If the filter table is empty no services will be granted to any node.

The following HMI command is used to manipulate the slave acceptance filter:

```
(u)unicast <port_number> (f)filter (s)slave [add|del]<address|*> [cid] [x | [ai][si][di|x]]
```

Where the first parameter is either a node's protocol address in corresponding transport protocol format, for example:

192.168.1.10	IPv4 address
fe80:0000:0000:0000:a8ed:4c2f:fc6:335c	IPv6 address
B9-7F-C2-11	Ethernet addresses
*	Stands for any address

The cid parameter is an optional clock identifier of the node.

The ai, si, di parameters are the minimum log message intervals (maximum message rates) for Announce, Sync and Delay Request/Response messages the master will grant to that node or 'x', which stands for denial of either any services or of only Delay Request/Response.

For example:

u f s	Prints the content of the slave acceptance filter.
u f s add * -2 -4 -3	Adds an entry to accept requests from any node and limits message rates to: 4 msg/s for Announce, 16 msg/s for Sync, 8 msg/s for Delay Request/Response.
u f s add 192.168.1.100 -1 -2 x	Adds an entry to accept requests from 192.168.1.100, limits the rates to: 2 msg/s for Announce, 4 msg/s for Sync, and denies requests for Delay Request/Response
u f s add * 00:01:02:0F:FE:03:04:05 x	Adds an entry to deny all requests from the node with clock identifier 00:01:02:0F:FE:03:04:05 transmitted from any address.
u f s 192.168.1.100 -1 -2 -2	Changes rate limits for the specified node.
u f s del 192.168.1.100	Removes the specified node from the filter table.

If, as the result of the filter table modification, a node becomes unacceptable or message rates of any active grants becomes beyond the newly configured limits, all affected grants will be cancelled.

### 10.6.1.3 Maintaining the Static Slaves List

To provide unicast services to slaves which do not support the unicast negotiation the master maintains a list of static slave nodes.

When a node is in this list the master can send Announce and Sync messages to that node and can reply to Delay Request messages received from that node.

The following HMI command is used to manipulate the static slave node list:

```
(u)unicast node <port_number> (s)slave [add|del] <address> <ai> <si> <di|x>
```

Where the address parameter is a node's protocol address in corresponding transport protocol format, for example: 192.168.1.10 – for IPv4 addresses, fe80:0000:0000:0000:a8ed:4c2f:fc6:335c – for IPv6 addresses, 00-0B-B9-7F-C2-11 – for Ethernet addresses,

**NOTE:** The node's protocol address format must match the transport protocol configured on the PTP port.

The ai and si parameters are the log message intervals for Announce and Sync messages the master sends to the node.

The di parameter is the Delay Request log message interval the slave node is configured with or 'x' which stands for not replying to Delay Requests from this node.

**Note:** The master cannot control how often Delay Request messages are transmitted by the slave, so the <di> value is used mainly for the master's resource estimation.

For example:

u node	Prints static slave nodes list
u node slave add 192.168.1.100 1 0 0	Adds a node to the static slave list with specified Announce, Sync and Delay Request/Response rates.
u node slave 192.168.1.100 1 0 x	Modifies the transmission rates for the node, so the master only transmits Announce and Sync messages, but does not reply to Delay Request messages.
u node slave del 192.168.1.100	Removes the node from the list.

### 10.6.2 Limiting the Total Message Rate

By default there is no limit on how much processing resources the master can use for servicing slave nodes. The master will try to accept all requests until its resources are exhausted and it becomes overloaded, which may cause performance degradation.

To avoid such situation the master can be configured with a limit on total number of unicast messages per second it is allowed to transmit.

For every new node requesting unicast services the master calculates the message rate required and adds it to the total. As soon as the total rate reaches the limit the master stops accepting requests from any new nodes.

To setup the limit the following HMI command should be used:

```
(u)unicast <port_number> limit <value>
```

Where the value is the maximum number of messages per second or 0 which stands for no limit.

### 10.6.3 Unicast Slave

A PTP port in unicast slave state can work with both:

- Master nodes which can provide unicast services upon request, using the unicast negotiation mechanism and
- Master nodes which transmit unicast messages to a preconfigured set of slaves.

To obtain unicast services from master nodes supporting the unicast negotiation the slave needs to be configured as follows:

- Unicast negotiation must be enabled.
- Master discovery table must be populated.

To accept unicast messages from master nodes transmitting to a set of slaves the master acceptance filter needs to be properly configured.

When the new node is added to the master discovery table the slave first requests an Announce message transmission from that node. The requested transmission interval for Announce messages is the same for all master nodes and is taken from a portDS.logAnnounceInterval configured for the PTP port. The requested duration is the same for all messages (see below). If there is no reply or the request is denied, the slave will continue its requests every query interval (see below). As soon as the Announce message transmission is granted the slave is ready to accept them.

When the slave receives Announce messages from a node other than those configured in the master discovery table, received Announces are compared against the master acceptance filter. If such node is allowed by the filter the slave will accept them.

All accepted Announce messages are processed by the slave and as the result the best master is selected. From the best master the slave requests Sync and optionally Delay Response messages according to the settings specified for that node. Or, if the best master was not from the master discovery table, the slave just starts accepting Syncs from and sending Delay Requests to this master.

#### 10.6.3.1 Enabling Unicast Negotiation

The unicast negotiation state is controlled by the following HMI command:

```
(u)unicast <port_number> (n)negotiation <on|off>
```

When unicast negotiation is enabled the slave is able to participate in unicast negotiation with negotiation-capable masters.

If unicast negotiation is disabled no new requests are transmitted, but all existing grants remain active until they are either expired or cancelled.

### 10.6.3.2 Maintaining the Master Discovery Table

The following HMI command is used to manipulate the master discovery table:

```
(u)unicast <port_number> node (m)master [add|del] <address> [cid] <priority> <si> <di|x> [domain]
```

Where the address parameter is a node's protocol address in corresponding transport protocol format.

For example:

192.168.1.10	for IPv4 addresses
fe80:0000:0000:0000:a8ed:4c2f:fc6:335	for IPv6 addresses
00-0B-B9-7F-C2-11	for Ethernet addresses

The node's protocol address format must match the transport protocol configured on the PTP port.

The priority parameter is the local priority of the master on a given port. The priority value should be between 0 and 255, 0 stands for highest priority. Local priority takes precedence over clock properties evaluated by dataset comparison algorithm when selecting best clock on a given port (Er\_best).

The si parameter is the log message interval for Sync messages the slave shall negotiate with the master.

The di parameter is the log message interval for Delay Request/Response messages the slave shall negotiate with the master or 'x' if Delay Request/Response messages shall not be used.

**Note:** Announce messages are always requested and the message interval is always taken from portDS.logAnnounceInterval configured for the PTP port.

For example:

u node	Prints the master discovery table.
u node master add 192.168.1.100 0 -1 -1	Adds a node to the table with local priority 0 and with intervals -1 (2 msg/s) for Sync and Delay Request/Response messages
u node master 192.168.1.100 0 0 x	Modifies the node's parameters, so the slave requests only Sync messages and does not use Delay Request/Response.
u node master del 192.168.1.100	Deletes the specified node from the master discovery table.



### 10.6.3.3 Maintaining the Master Acceptance Filter

The master acceptance filter is a mechanism for control which nodes the slave should accept unicast messages from. If the filter table is empty no messages will be accepted from any nodes other than those specified in the master discovery table.

The following HMI command is used to manipulate the master acceptance filter:

```
(u)unicast <port_number> (f)filter (m)master [add|del] <address|*> [cid] [x | [priority][si][di|x]]
```

Where the first parameter is either a node's protocol address in corresponding transport protocol format.

For example:

192.168.1.10	For IPv4 address
fe80:0000:0000:0000:a8ed:4c2f:fc6:335c	For IPv6 address
00-0B-B9-7F-C2-11	For Ethernet address
*	Stands for any address

The cid parameter is an optional clock identifier of the node.

The 'x' as the next parameter stands for not accepting any messages from the node.

The priority parameter is the local priority of the master on a given port. The priority value should be between 0 and 255, 0 stands for highest priority. Local priority takes precedence over clock properties evaluated by dataset comparison algorithm when selecting best clock on a given port (Er\_best).

The si parameter is the log message intervals for Sync messages the master is configured to send to the slave.

**Note:** The slave cannot control how often Sync messages are transmitted by the master, so the si value is used only for information.

The di parameter is the log message interval for Delay Request messages the slave shall use, or 'x', if Delay Request/Response messages will not be used.

For example:

u f m	Prints the content of the master filter table.
u f m add * 255 -2 -2	Adds an entry to accept requests from any master node. It indicates that the master is going to send 4 Sync messages per second, and instructs the slave to send 4 Delay Requests messages per second.
u f m add 192.168.1.100 0 -2 x	Adds an entry to accept requests from 192.168.1.100 and instructs the slave not to use Delay Requests.
u f m add * 00:01:02:0F:FE:03:04:05 x	Adds an entry to deny all messages from the node with specified clock identifier transmitted from any address.
u f m del 192.168.1.100	Removes the specified node from the filter table.

If as the result of the master acceptance filter table modification a node becomes unacceptable no more messages will be accepted from that node.

### 10.6.4 Unicast Both

A port is set up as "unicast both", means it can potentially switch between PTP slave and master states or between being grant port and request port, it is typically set up with one or more unicast masters in its discovery table and one or more acceptable unicast slave filter entries.

Assuming other relevant ports on the same network as this port are set up similarly, announce messages will be exchanged with them in both directions and whichever port is determined to be a PTP slave to another will request and be granted sync and delay response service from that port.

### 10.6.5 Message Transmission Duration

When the slave requests unicast transmission it specifies the duration, a time period in seconds for which the requested messages should be transmitted.

The requested duration is the same for all message types and for all masters. By default it is set to 300 seconds, but can be configured to any value between 60 and 3600 seconds.

The requested duration is changed with the following HMI command:

```
(u)unicast <port_number> (d)duration <value>
```

### 10.6.6 Discovery Query Interval

If there is no reply to the message transmission request issued by the slave or the request is denied, the slave will repeat the request every query interval.

By default it is set to 60 seconds, but can be configured to any value between 1 and 600 seconds.

The query interval is changed with the following HMI command:

```
(u)unicast <port_number> (q)query <interval>
```

### 10.6.7 Monitoring Unicast Operations

Besides unicast-related commands mentioned in previous chapters there is an HMI command to display the summary of unicast configuration parameters:

```
(u)unicast <port_number> (s)status
```

Another HMI command allows to see the list of all nodes with which unicast communications are established and the particular details of them:

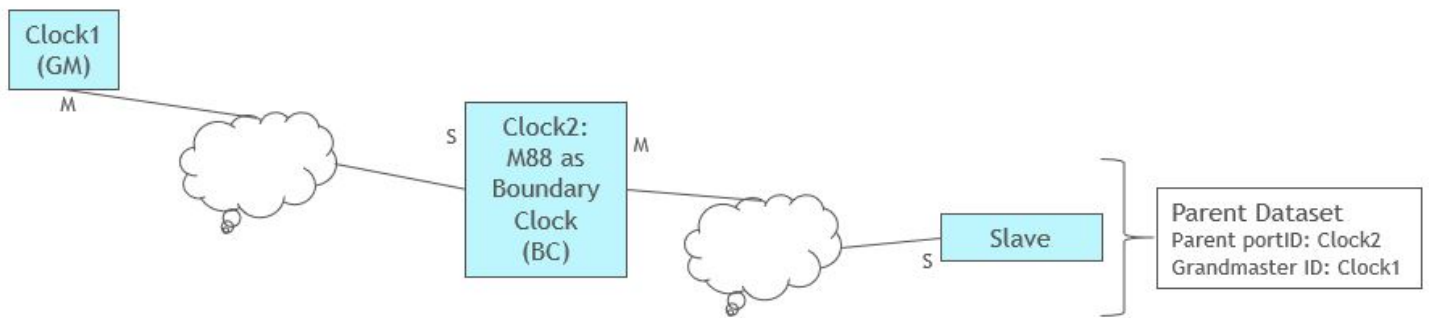
```
(d)dataset (u)unicast
```

## 10.7 Gateway and Boundary Clock (M68 Only)

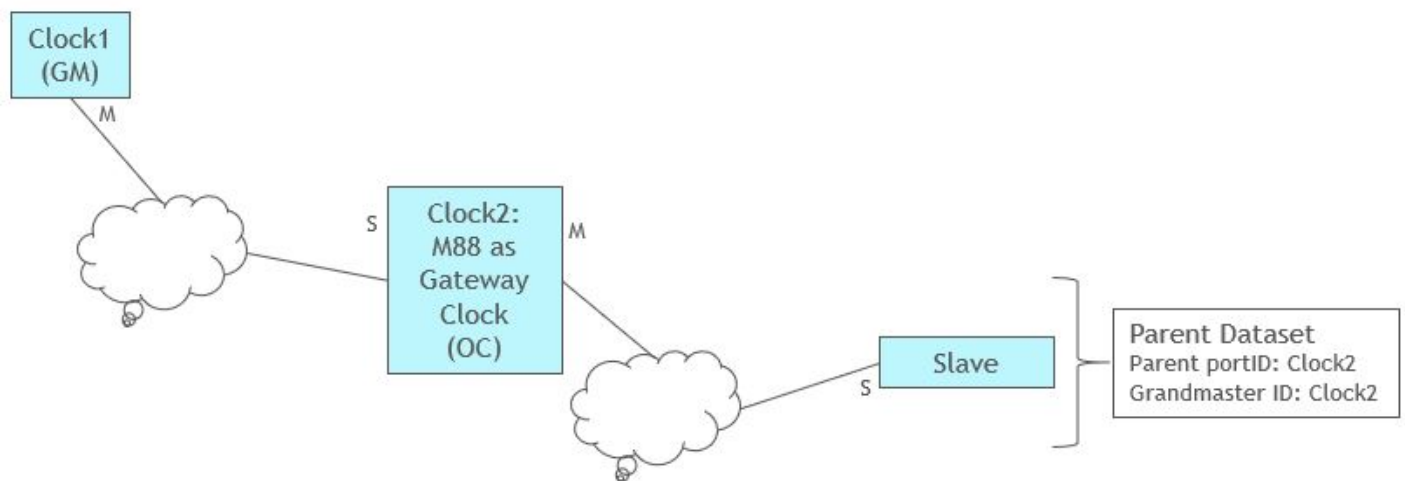
The M68 has the option of using both its Ethernet ports as PTP ports. This opens the possibility of using the M68 as a two port gateway or boundary clock.

### 10.7.1 The Gateway Clock

You can see the gateway clock as an ordinary clock with two ports. The figures below illustrate the difference between the gateway clock and the boundary clock.

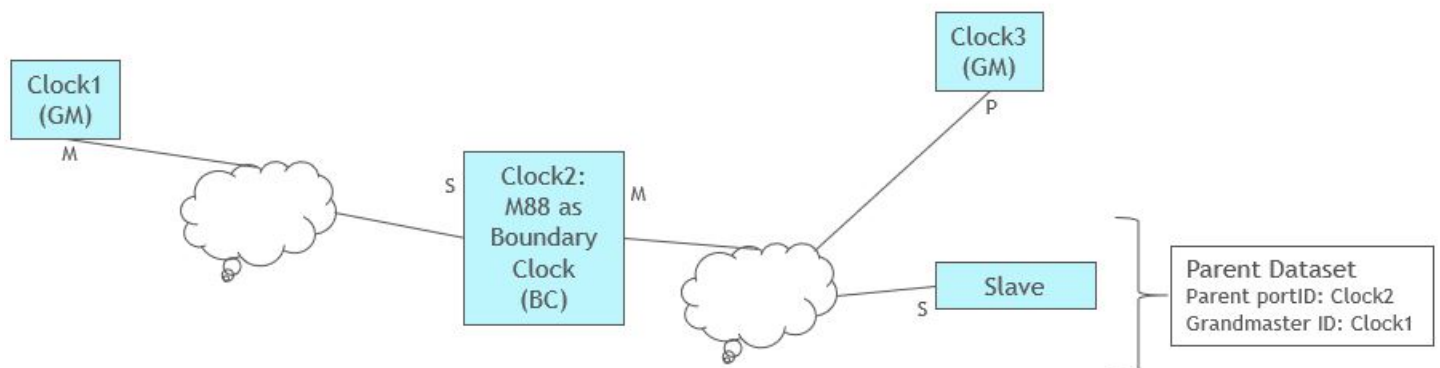


**Figure 10.1. M64/M68 as Boundary Clock – Slave Recognizes Clock1 as Grandmaster**

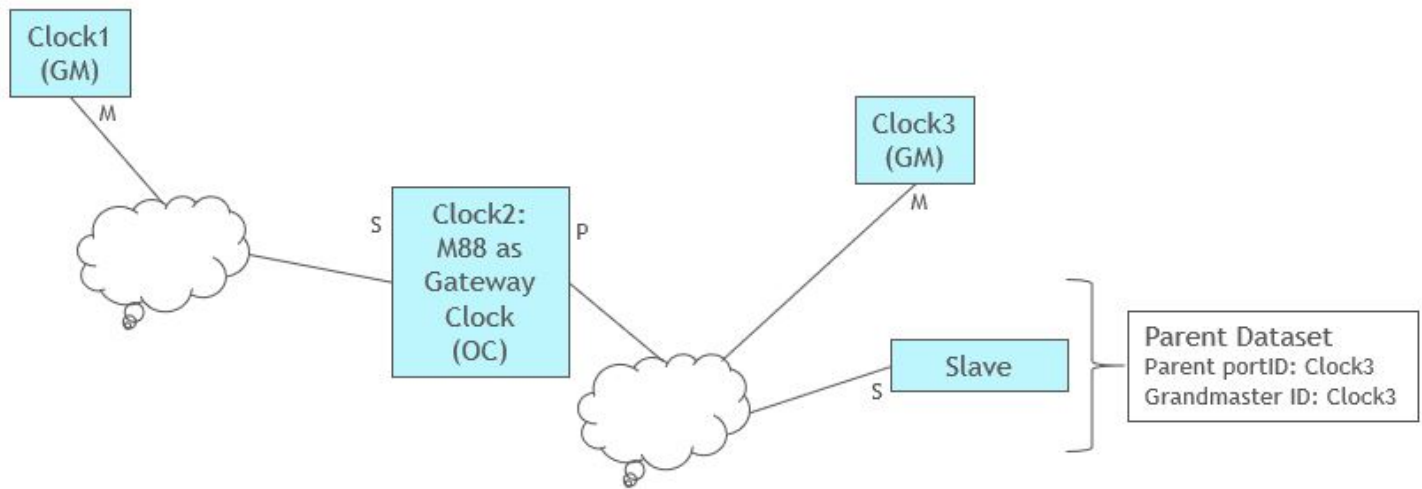


**Figure 10.2. M64/M68 as Gateway Clock – Slave Recognizes Clock2 as Grandmaster**

Figures below illustrate the selection of the best master clock is on the same side of the gateway clock



**Figure 10.3. M64/M68 as Boundary Clock: If Clock1 is Better than Clock3, Clock3 is Passive State**



**Figure 10.4. M64/M68 as Gateway Clock – If Clock3 is Better than Clock2, Clock2 is Passive State**

The type of clock is defined by the command `ptp2 clock mode oc | bc` where `oc` is the default mode. Both the gateway and boundary clock can of course be combined with the possibility of synchronizing the clock by PPS and ToD (using PTP engine mode 2 or 3).

### 10.7.2 Managing the Two Ports

To enable both ports for PTP, the following needs to be added/changed in the `system.ini` file:

```
TCPIP both
```

To enable either of the interfaces as PTP port, use the following:

```
TCPIP eth0
```

or

```
TCPIP eth1
```

The `eth1` does not support DHCP but needs a static IP address. In the `ipconfig` command the two ports are referred to as `enet0` and `enet1` as follows:

**Syntax** `ipconfig [IFACE] [OPTIONS]`

```
ipconfig all
```

The PTP ports are referred to as 1 and 2 in commands such as defining message rates, e.g. `ptp2 port 2 sync -2` which sets the synch rate on PTP port 2 (`enet1`) to 4 per second. If the PTP port number is omitted, port 1 is chosen by default.

The two PTP ports can be configured completely independent of each other but will of course operate on the same clock. Therefore, it's impossible to configure both ports as slaves as only one of them can synchronize the clock.

On the other hand, it's fully possible to configure one port as a unicast slave running on L2 (Ethernet) while the other port is multicast running P2P over UDP.

### 10.7.3 Unicast

In unicast operation the identification of the port is put after the unicast option. For example, `ptp2 unicast 2 negotiation on` to turn on unicast negotiation on port 2.

## 10.8 Performance Optimization

There are number of parameters related to the local oscillator properties and loop control algorithm that affect the clock synchronization behavior and performance. These parameters are static, i.e., cannot be altered while the PTP Engine is up and running, so they need to be configured before it starts up.

All these parameters are set to reasonable defaults, which are suitable for most types of networks. Hence, there is no need to change them right away, but tuning them for particular network environment may dramatically improve the synchronization performance.

- **VCO Control Range** – defines the drift difference between the midpoint and the endpoint for the control signal to the oscillator. It is expressed in ns/s. The default value suited for TCVCXO oscillator (mounted by default on the P6x) is 6000, which means that the controllable range of is +/- 6000 ns/s. If this parameter is set to zero, the oscillator frequency will not be controlled by the loop control algorithm but run free. In this case the time offset compensation is done through the software and no syntonized frequency output is available.
- **VCO Static Drift** – specifies the initial drift the oscillator should be configured with to get the desired frequency at startup. It is expressed in ns/s. The default value is zero, which corresponds to the middle point of the controllable range. After the system starts to synchronize the loop control algorithm automatically changes the oscillator frequency moving it to the correct point, but setting the initial drift closer to the correct point at startup will decrease the synchronization time.
- **Main Time Constant** – defines main time constant used by the control algorithm. It is expressed in seconds. A longer time constant makes the oscillator frequency change more slowly and less responsive to changes in an environment, such as the temperature or the network delays. For networks with low packet delay variation a short time constant (e.g., 30 sec) can be used. For networks with unknown or high traffic load it is better to use long time constant (e.g., 300 sec). The default value is 100.
- **Startup Time Constant** – defines the time constant used during the period of obtaining a preliminary synchronization (startup phase). It is expressed in seconds. If this field is set to 0 the main time constant will be used. This parameter can help to shorten the synchronization time when the long main time constant is required. But too short values may lead to unwanted oscillations which increase the total synchronization time and eliminate any positive effects of shorter time constant. The default value is 0.
- **Path Delay Time Constant** – defines the time constant used by control algorithm for path delay calculations. This parameter is expressed in seconds. The default value is 300. For networks with low packet delay variation the smaller value can be used if the fast reaction on topology changes is necessary.
- **Quality Threshold** – defines the maximum allowed variation on a statistical measurement before the slave goes into hold-over. The default value is 1500.
- **Square Threshold** – this parameter can be used to keep the changes on the syntonized frequency small, while allowing slightly larger time offset. It is expressed in ns. The threshold can be disabled (set to zero) if the offset is the most important property. If it is enabled, the threshold value should be larger than the observed packet delay variation (excluding outliers) but smaller than the expected maximum time offset. The default value 0 (disabled).

**Network type optimization** – this parameter adjusts the loop control algorithm behavior according to the underlying network characteristics. There are three types of networks currently supported: generic, engineered and networks with full on-path PTP support. “Generic” stands for a network with unknown or unspecified packet delay and delay variation or with delay distribution that may dramatically change its characteristics over the time.

“Engineered” is for a network where the packet delay distribution always has a distinct “floor”. That ‘floor’ may vary over the time but it should still be present.

“Full on-path” is the network with full timing support as described in G.8275.1 telecom profile. As all telecom boundary/slave clocks must have certain frequency response according to G.8273.2 this option must be combined with very short main time constant. See examples below.

**GNSS master main time constant** – defines main time constant used by the control algorithm when synchronizing to GNSS. The default value is 3.

**GNSS master startup time constant** – defines the time constant used during the period of obtaining a preliminary synchronization to GNSS. If set to 0 the main time constant will be used. The default value is 0.

**Minimize Frequency Change** – this works similar to the “Square Threshold” parameter but for a device synchronizing to PPS. It can only be turned on (1) or off (0). In the firmware releases prior to v2.9.13, setting this parameter to “on” effectively changes GNSS main and startup time constants to 100 and 30 respectively. Starting with v2.9.13 this parameter, when set to on, keeps time constants as configured, but selects lower values for control algorithm gains instead. To preserve the old behavior of this option on newer releases one must need to set GNSS main and startup time constants to 100 and 30 respectively, while keeping this option off.

For viewing values currently assigned to static parameters and for changing them the ISH command `ptp2 config` should be used.

To optimize the performance for special network conditions, such as a direct connection with no traffic load, the loop controller parameters can be changed.

### **10.8.1 Direct Connection**

A slave is connected to a master with a direct cable:

Set a short time constant, e.g., 30 s. The sync rate should be at least 1 Hz.

### **10.8.2 Low Traffic Network**

A slave is connected to a master through a couple of switches or a router, with no or low traffic load:

Set a medium time constant, e.g., 100 s (default). The sync rate should be at least 1 Hz.

### **10.8.3 High Traffic Network**

A slave is connected to a master through several switches or routers, with a traffic load of more than 50%:

Set a longer time constant, e.g., 300 s. The sync rate should be at least 16 Hz.

### **10.8.4 ITU-T G.8261, G.8265.1, G.8275.2 Settings**

ITU-T G.8261 specifies a number of test cases in a 10-switch loaded network.

Set a long time constant, 300 s or longer (use a 100 s startup time constant) and set the network mode to engineered. Use a sync rate of at least 16 Hz.

### **10.8.5 ITU-T G.8273.2, G.8275.1 settings**

Use a very short time constant, 1-3 s. Set the network mode to full on-path support.

## 10.9 Power Profile

Both versions of power profile, C37.238-2011 (version 1) and C37.238-2017 (version 2), are supported. To have the module working properly in a network adhering to these profiles, use the clock profile command:

```
(c)clock (f)profile <default|g8265|g8275.1|g8275.2|power [v2]>
```

e.g.,

```
c f power v2
```

sets the clock profile to C37.238-2017.

The following parameters will be changed on both ports when setting the clock profile to power or power v2:

- Network Protocol is set to 802.3
- Delay Mechanism is set to P2P
- Unicast communication is disabled
- Announce interval is set to 1, Sync interval to 0 and Peer Delay Request interval to 0

In addition, the clock domain is set to 0 for C37.238-2011 and 254 for C37.238-2017

There are several configuration parameters for the power profile. These are Grandmaster ID, Time Inaccuracy and Alternate Timescale information.

The following command configures the Grandmaster ID:

```
(pw)power (g)gmid <0..65535>.
```

The default value for Grandmaster ID is 0.

As in version 1 of Power profile the Grandmaster ID must not be 0, a port set to this profile will not transmit any the profile-specific TLVs until the Grandmaster ID is configured to non-zero value in its low order byte.

The Time Inaccuracy is configured with this command:

```
(pw)power (i)inacc <grandmaster> <network>
```

Both values are 0 by default.

In version 2 of Power profile these two values are summed up and transmitted as a single Total Time Inaccuracy field in the profile-specific TLV.

Alternate timescale information is automatically taken from the currently configured system's timezone (see 'date' command). To view the profile-specific values obtained by a slave from its master the following command can be used:

```
(d)dataset (pw)power
```

## 10.10 ITU-T G.8265 Profile

To have the module working properly in a network adhering to this profile, use the clock profile command:

```
(c)clock (f)profile <default|g8265|g8275.1|g8275.2|power [v2]>
```

e.g.,

```
c f g8265
```

sets the clock profile to G.8265.

The following parameters will be changed when setting the clock profile to G.8265:

- Network Protocol is set to UDP over IPv4
- Delay Mechanism is set to E2E
- Unicast Master
- Announce interval is set to 1, Sync interval to 0 and Delay Request interval to 0

For the M68 the change will be on both ports.

In addition, the clock domain is set to 4.

There are no profile specific parameters to be set for G.8265 profile.

## 10.11 ITU-T G.8275.1 Profile

To have the module working properly in a network adhering to this profile, use the clock profile command:

```
(c)clock (f)profile <default|g8265|g8275.1|g8275.2|power [v2]>
```

e.g.,

```
c f g8275.1
```

sets the clock profile to G.8275.1.

The following parameters will be changed when setting the clock profile to G.8275.1:

- Network Protocol is set to 802.3
- Delay Mechanism is set to E2E
- Unicast communication is disabled
- Announce interval is set to -3, Sync interval to -4 and Delay Request interval to -4

For the M68 the change will be on both ports.

In addition, the clock domain is set to 24.

There are several profile specific parameters to be set for G.8275.1 profile. These are Local clock priority, Max steps removed, local port priority and master only attribute.

The local clock priority is configured using the command

```
(g2)g8275 (p)priority <1..255>
```

The default is 128.

The Max steps removed is configured using the command

```
(g2)g8275 (s)steps <1..255>
```

The default is 255.

The local port priority is configured using the command

```
(g2)g8275 <port_number> (p)priority <1..255>
```

The default is 128.

The port master-only attribute is configured using the command

```
(g2)g8275 <port_number> (m)masteronly <on|off>
```

The default is off.



## 10.12 ITU-T G.8275.2 Profile

To have the module working properly in a network adhering to this profile, use the clock profile command:

```
(c)clock (f)profile <default|g8265|g8275.1|g8275.2|power [v2]>
```

e.g.,

```
c f g8275.2
```

sets the clock profile to G.8275.2. For the M68, it sets the profile for both ports.

The following parameters will be changed when setting the clock profile to G.8275.2:

- Network Protocol is set to UDP over IPv4
- Delay Mechanism is set to E2E
- Unicast Both
- Announce interval is set to 1, Sync interval to 0 and Delay Request interval to 0

In addition, the clock domain is set to 44.

There are several profile specific parameters to be set for G.8275.2 profile. These are Local clock priority, Max steps removed, local port priority and master only attribute.

The local clock priority is configured using the command

```
(g2)g8275 (p)priority <1..255>
```

The default is 128.

The Max steps removed is configured using the command

```
(g2)g8275 (s)steps <1..255>
```

The default is 255.

The local port priority is configured using the command

```
(g2)g8275 <port_number> (p)priority <1..255>
```

The default is 128.

The port master-only attribute is configured using the command

```
(g2)g8275 <port_number> (m)masteronly <on|off>
```

The default is off.

## 11. Command Reference

All commands can be issued by a user from a serial terminal or a telnet session. The output from any command is redirected to the output device it was issued from.

In contrast to commands the debug output can have only one output device. So, enabling it on one device (or in one session) will automatically disable it on another.

The commands can be sent from the command prompt of the shell.

A full list of commands is provided in [18. Appendix 3: PTP Command Reference](#).

The most commonly used shell commands are listed here.

**Note:** The choice of start parameter decides the basic functionality of the product, i.e. master, slave with a master capability or grandmaster.

### 11.1 Shell Commands

#### 11.1.1 ptp2 start

```
ptp2 start <mode> <option>
```

Starts PTP Engine in the following mode:

- 0 – GNSS disabled, only PTP (master-slave),
- 1 – GNSS reference only (master only). Default mode,
- 2 – GNSS primary, PTP secondary (master-slave),
- 3 – GNSS secondary, PTP primary (master-slave),
- 4 – GNSS debug.

Supported options are:

--silent: no unsolicited console output until HMI debug is turned on

**Note:** In this context, "GNSS" means GPS/GNSS.

#### 11.1.2 ptp2 stop

```
ptp2 stop
```

Stops PTP Engine.

#### 11.1.3 ptp2 hmi

```
ptp2 hmi
```

Enters the HMI command processing mode. To quit the HMI command processing mode type q.

### 11.1.4 ptp2 config

ptp2 config <options>

Configures oscillator and loop control related static parameters. Configuration options are:

- v [value] – VCO control range, in ns/s (default 6000),
- d [value] – VCO static drift, in ns/s (default 0),
- u [1|0]: voltage feedback, 1 - on (default 1)
- t [value] – slave main time constant, in sec (default 100),
- s [value] – slave startup time constant, in sec, 0 - off (default 0),
- m [value] – master main time constant, in sec (default 3),
- l [value] – master startup time constant, in sec, 0 - off (default 0),
- p [value] – path delay time constant, in seconds (default 300),
- q [value] – quality threshold, in ns, (default 1500),
- e [value] – square threshold, in ns, 0 - off (default 0).
- f [0|1] – minimize frequency change, 1 – on, 0 – off (default 0).
- n [g|e|f] – optimize for either generic ('g'), engineered ('e') or full on-path support ('f') network (default 'g').

### 11.1.5 ptp2

ptp2 <command>

Sends a single HMI command to the PTP Engine.

## 12. JSON Usage

As a machine-to-machine interface, JSON has several advantages over the current CLI implementation:

- JSON is designed to be easy to parse
- The interface information for each command is now documented (see Appendix 9: JSON Interface Definition: JSON Interface Definition)
- JSON is very human readable

Be cautioned that the command port (typically COM3) can be switched between JSON and CLI/HMI, but the switch is not automatic. The scope of the JSON commands is the ptp2 engine commands.

Example: Using the ptp2 clock info command as an example

- Before JSON

```
A:/root> ptp2 clock info
A:/root> *** Clock description ***
{
  Clock type: ORDINARY CLOCK
  Ports number: 2
  Physical layer protocol: IEEE 802.3
  Port 1:
    Physical address: FC:AF:6A:02:EA:28
    Network protocol: UDP/IPv4
    Network address: 0.0.0.0
    Profile ID: 00-1B-19-00-01-00
  Port 2:
    Physical address: FC:AF:6A:01:EA:28
    Network protocol: UDP/IPv4
    Network address: 192.168.2.99
    Profile ID: 00-1B-19-00-01-00
  Manufacturer ID: 90 FD 9F
  Product description: Silicon Laboratories;PTPv2 Engine;02-EA-28
  Revision data: ;3.0.0
  User description: ;
}
```

- After enabling JSON:

```
A:/root> ptp2 engine json on
A:/root> ptp2 clock info
A:/root> {
  "clockInformation":
  {
    "clockType": "ORDINARY CLOCK",
    "physicalLayerProtocol": "IEEE 802.3",
    "numberPorts": 2,
    "portTable":
    [
      {
        "portNumber": 1,
        "physicalAddress": "FC:AF:6A:02:EA:28",
        "protocolAddress":
        {
          "networkProtocol": "UDP/IPv4",
          "networkAddress": "0.0.0.0"
        },
        "profileIdentity": "00-1B-19-00-01-00"
      },
      {
        "portNumber": 2,
        "physicalAddress": "FC:AF:6A:01:EA:28",
        "protocolAddress":
        {
          "networkProtocol": "UDP/IPv4",
          "networkAddress": "192.168.2.99"
        },
        "profileIdentity": "00-1B-19-00-01-00"
      }
    ]
  },
  "manufacturerIdentity": "90 FD 9F",
  "productDescription": "Silicon Laboratories;PTPv2 Engine;02-EA-28",
  "revisionData": ";;3.0.0",
  "userDescription": ";"
}
```

## 12.1 Enabling JSON

- If the JSON interface is being used for M2M communications, the host software must be updated with the JSON commands and responses - see [24. Appendix 9: JSON Interface Definition](#).
- Once the ptp2 engine is running JSON can be enabled with

```
ptp2 engine json on
```

- Until the JSON capability is turned off, all responses from the ptp2 engine are in JSON format
- The JSON capability is turned off with

```
ptp2 engine json off
```

### 13. Product Documentation

The following documents can be downloaded from the Silicon Labs website (<https://www.silabs.com/>). Registration may be required. Any documents not available at the website can be requested from Silicon Labs by accessing the support section of the

<https://www.silabs.com/> website.

P60 User Guide. Note: The P60 is a prototyping board designed around the M68, which is ideal for evaluation, and low volume applications.

## 14. Part Numbers

Ordering Part Numbers:

- M64-Axxxxx-GM
- M68-Axxxxx-GM

Where

- A is the product revision
- xxxxx is either
  - 0xxxx where xxxx is the software revision, e.g. M68-A01912 where the 01912 stands for software release 1.9.12
  - 1xxxx where the xxxx is a custom part number
- GM is for -40C to 85C operating temperature range

An optional “R” suffix may be used for tape and reel ordering.

## 15. Customer and Technical Support

For the latest product information, see <https://www.silabs.com/>.



## 16. Appendix 1: System Files

File	Description
access.ini	Access rules for telnet and ftp server.
boot.ini**	Specifies what program shall be started by the boot loader.
group.ini	Contains the group definitions.
hostname.ini	Specifies system host name.
ish.ini	Common session configurations.
16. Appendix 1: System Files	User specific configurations.
passwd.ini	The user database.
ptp.ini	PTP protocol engine configurations. Currently not supported.
startup.ini	Contains startup commands.
system.ini	System configurations.
leapsec.ini	This file contains information about passed and upcoming leapseconds.
hostkey	Private RSA key for SSH server.
system.ver**	Firmware version file.
system.gpx**	The runtime environment.
M60_top.bit**	Binary file containing HW specific information.

\*\* The following files contain the system image and control the boot process and should not be modified in any way:

- system.gpx
- boot.ini
- system.ver
- any file that ends in .bit

Modifying any of these files may render the system unbootable. Contact Silicon Labs Applications for more information.

### access.ini

This file contains the access rules for the telnet and ftp server. Each line may either contain an allow entry or a deny entry. The order option defines which of the entries allow and deny that has the highest priority. If the order is allow, deny, the default is to allow connections from all hosts except those that matches a deny entry. If the order is deny, allow the default is to deny all connections, except those that matches an allow entry.

Example:

```
allow: 192.168.1.0/255.255.255.0
deny: 192.168.1.90/255.255.255.255
order: deny, allow
```

Which will allow all hosts from the 192.168.1.0 network to connect, except for 192.168.1.90 who will be declined access.

### boot.ini

The boot.ini file contains information of what executable file that will be started by the boot loader. The executable can be explicitly specified or a probe program can be started. The probe programs exit code will then be used by the boot loader to select which program to start.

The lines in the boot.ini file that specifies boot files starts with a one letter label followed by a colon character. The letters d and p are reserved for the default startup program and the probe program respectively.

In the following example the /system/system.gpx will be started by the boot loader.

```
d: /system/system.gpx
```

Using a Probe Program

A probe program can for example sense an external signal and decide whether the system should be booted with ordinary system program or some other program. In the following example, the system will be booted with `system.gpx` if the probe program (`probe.gpx`), returns the exit code 'd' and with `setup.gpx`, if the exit code is 'a'.

```
d: /system/system.gpx
p: /system/probe.gpx
a: /system/setup.gpx
```

### group.ini

This file contains the group definitions. Each line is an entry for a group. Every field is delimited with a ';':

Group name	The name of the group.
Password	This field is ignored.
Group ID	The numeric value associated with the group

The following groups will exist from the beginning:

- root
- guest
- user
- ppp

The root group is special since it enables administrator privileges.

### hostname.ini

The hostname of the system. You can either edit this file by hand or use the `hostname` command.

The default name is **localhost.localdomain**.

### ish.ini

This file contains common configuration data for all login sessions. This file will be read when a new session is initiated.

By default it contains some alias definitions and the `CLASSPATH` variable.

See also [16. Appendix 1: System Files](#).

When the `/system/ish.ini` has been parsed, the users home directory, `$HOME`, is searched for a `_ish.ini` file. Each user may store his/hers personal settings in this file.

### passwd.ini

This file contains the user database. It is in a standard UNIX format, where each line consists of an entry for a user.

Each line consists of the following fields separated with a semicolons:

Field	Description
User name	The user name, up to 16 characters.
Password	DES encrypted password.
User ID	A numeric value in the range [0-255]. 0 is reserved for the root account.
Group ID	A numeric value in the range [0-255]. 0 is reserved for the root group. See the <a href="#">group.ini</a> file.
Full user name	The user's full name.
Home directory	The user's home directory. This is where the shell will try to read the users <code>ish.ini</code> file.
Shell to start	This is only to be compatible with the standard UNIX format, and is currently not used.

To modify this file, use the `useradd` and `userdel` commands. To alter a password, use the `passwd` command.

In the beginning there exists two accounts, `root` and `guest`, which has the following passwords:

- root: root

- `guest: guest`  
`ptp.ini`

This file contains PTP protocol engine configuration settings. The protocol engine reads this file during the initialization phase when it is started.

NOTE: At the moment this file is only available in systems that supports IEEE 1588/PTP Version 1.

### **startup.ini**

This file may contain commands to start an application automatically. This file will be read and executed after `system.ini` has been read and the system has been initialized.

Commands executed in this file will have root permissions. The starting directory will be set to "A:/". If the serial debug terminal is enabled, the output of all commands in `startup.ini` will be redirected to this terminal.

Example:

```
mount c
date
```

### **system.ini**

This file contains the system configuration and will be read and executed at boot time. This file should contain the system wide configuration variables. See Environment Variables for more information.

### **leapsec.ini**

This file contains information about passed and upcoming leapseconds. This file is read by the PTP engine and should be used to schedule upcoming leapseconds. This file can be modified by the PTP commands related to removing and adding leapseconds. See [18. Appendix 3: PTP Command Reference](#).

### **hostkey**

This file contains the SSH server's private RSA key.

By default the firmware comes with a `hostkey` file containing 1536 bits RSA key generated at Silicon Labs for demonstration and testing purpose. If the system is going to be deployed to the secure live environment this file should be replaced with another one containing the customer's own secure private key.

If the `hostkey` file is missing the SSH server will try to generate a temporary 1024 bits key upon startup. This is a slow process and may take up to several minutes. The temporary key will be used by SSH server only during current session (i.e. until system restart or SSH server shutdown) and will not be saved into the `hostkey` file.

### **system.ver**

This file contains a string reflecting the current version of the firmware.

### **system.gpx**

The runtime environment. This executable which contains the ISH, telnet server, ftp server etc. will be started by the boot loader.

See also `boot.ini`.

## 17. Appendix 2: System CLI Commands

• <a href="#">about on page 68</a>	Print a short description about the system.
• <a href="#">alias on page 70</a>	Define or print aliases.
• <a href="#">cat on page 70</a>	Print contents of a file.
• <a href="#">cd on page 71</a>	Change directory.
• <a href="#">chgrp on page 71</a>	Change group owner of a file.
• <a href="#">chmod on page 72</a>	Change file permissions.
• <a href="#">chown on page 73</a>	Change owner of a file.
• <a href="#">clear on page 73</a>	Clear screen.
• <a href="#">cmp on page 73</a>	Compare two files.
• <a href="#">cp on page 74</a>	Copy file.
• <a href="#">date on page 75</a>	Get/set date and time.
• <a href="#">debug on page 76</a>	Enable/disable debug output.
• <a href="#">df on page 77</a>	Print information about device usage.
• <a href="#">echo on page 77</a>	Print a line to the terminal.
• <a href="#">exit on page 77</a>	End an interactive session.
• <a href="#">find on page 78</a>	Search for a file in the file system.
• <a href="#">ftp on page 79</a>	FTP client.
• <a href="#">hd on page 80</a>	Print contents of a file in hexadecimal.
• <a href="#">help on page 81</a>	Print help about command.
• <a href="#">history on page 81</a>	Show the ish command history.
• <a href="#">hostname on page 82</a>	Print/set hostname.
• <a href="#">id on page 82</a>	Print information about the current user.
• <a href="#">ifconfig on page 83</a>	Configure Ethernet interface
• <a href="#">in on page 85</a>	Read a byte from an I/O port.
• <a href="#">ipconfig on page 86</a>	Print/set the settings for the TCP/IP stack.
• <a href="#">kill on page 87</a>	Kill a thread.
• <a href="#">ls on page 88</a>	List directory contents.
• <a href="#">mkdir on page 88</a>	Create directory.
• <a href="#">mkrd on page 89</a>	Create a ramdisk.
• <a href="#">more on page 89</a>	Show contents of a file.
• <a href="#">mount on page 90</a>	Mount FAT drive.
• <a href="#">mv on page 90</a>	Move file.
• <a href="#">netstat on page 90</a>	Show information of current network connections.
• <a href="#">nslookup on page 91</a>	Resolve a hostname through DNS.
• <a href="#">od on page 91</a>	Print contents of a file in octal.
• <a href="#">out on page 92</a>	Write a byte to an I/O port.
• <a href="#">passwd on page 92</a>	Change password for user.

• <a href="#">pd on page 93</a>	Print current drive.
• <a href="#">ping on page 93</a>	Ping a node on the Internet.
• <a href="#">ps on page 94</a>	Show information about processes/threads.
• <a href="#">ptp2 on page 95</a>	PTP version 2 protocol engine.
• <a href="#">pwd on page 96</a>	Print current directory.
• <a href="#">reboot on page 97</a>	Reboot system.
• <a href="#">recv on page 97</a>	Receive a file from the PC through the serial cable.
• <a href="#">rm on page 98</a>	Remove file.
• <a href="#">rmdir on page 98</a>	Remove directory
• <a href="#">route on page 98</a>	Print routing table information.
• <a href="#">send on page 99</a>	Send a file to the PC over the serial cable.
• <a href="#">setenv on page 99</a>	Define an environment variable.
• <a href="#">sleep on page 100</a>	Make process sleep.
• <a href="#">source on page 100</a>	Execute the contents of a file.
• <a href="#">startserver on page 101</a>	Start firmware servers.
• <a href="#">stats on page 102</a>	Show information/statistics about the system.
• <a href="#">stopserver on page 102</a>	Stop firmware servers.
• <a href="#">su on page 103</a>	Become another user.
• <a href="#">touch on page 103</a>	Update time stamp of a file.
• <a href="#">umount on page 103</a>	Unmount a drive.
• <a href="#">unalias on page 104</a>	Remove an alias definition.
• <a href="#">update on page 104</a>	Update the firmware through TFTP.
• <a href="#">useradd on page 105</a>	Add a user to the user database.
• <a href="#">userdel on page 105</a>	Remove a user from the user database.
• <a href="#">wall on page 106</a>	Send message to all users.
• <a href="#">who on page 106</a>	Show a list on logged in users.
• <a href="#">write on page 107</a>	Send message to a specific user.

## about

**Syntax:** `about`

Print a short description about the system.

## Description

Print a short description about the system. Prints a copyright message and what third party software that are used.

**alias****Syntax:** `alias [ALIAS] [DEFINITION]`

Define or print aliases.

**Description:**

Define or print aliases. Without any arguments all current defined aliases will be printed.

If one argument is available, that argument will be treated as an alias, and the corresponding definition will be printed.

If two or more arguments are available a new alias will be defined (or replaced if the alias already exists). The first argument will be treated as the alias name, and all other arguments will be put in the definition.

Aliases are defined in `ish.ini`.**Examples:**

Print all defined aliases.

```
alias
```

Print the definition of the "ls" alias.

```
alias ls
```

Define alias "monkey" to find the file "monkey.txt".

```
aliasmonkey find / monkey.txt
```

**See also**

- [unalias on page 104](#)

**cat****Syntax**`cat FILE ...`

Print contents of a file.

**Options***FILE* The file to print.**Description**

Print contents of a file. Every argument will be treated as a filename and the files will be printed in order.

**Examples**

Print the contents of one file.

```
cat/system/system.ini
```

Print the contents of several files.

```
Cat/system/system.ini /system/ish.ini
```

**See also**

- [hd on page 80](#) | • [more on page 89](#) | • [od on page 91](#)

## **cd**

**Syntax** `cd [DIRECTORY]`

Change directory.

### **Options**

*DIRECTORY* The directory to go to.

### **Description**

Change directory. If the directory is omitted, the current directory will be set to the value of the HOME variable. The current directory is stored in the PWD variable.

### **Examples**

Go to home directory.

```
cd
```

Go to root directory

```
cd /
```

### **See also**

- [pwd on page 96](#).

## **chgrp**

**Syntax:** `chgrp GROUP FILE`

Change group owner of a file.

### **Description**

Change group owner of a file. Change group owner of a file. The first argument must either be a group ID or a group name. The second argument is the name of the file to alter.

### **Examples**

Change group of system.ini to group '1'

```
chgrp1 /system/system.ini
```

Change group of system.ini to group "ppp"

```
chgrpPPP /system/system.ini
```

### **See also**

- [chmod on page 72](#) | • [chown on page 73](#)

## **chmod**

**Syntax:** chmod MODE FILE

Change file permissions.

### **Options**

#### *MODE*

The file mode.

#### *FILE*

The path to the file to alter.

### **Description**

Change file permissions. MODE is an octal number in the range 000 to 777. The first digit maps to the user's rights, the second digit to the group's rights and the last digit to other's rights. Bit 0 is X, bit 1 is W, bit 2 R.

### **Examples**

Set write bit for all users

```
chmod 222 /system/system.ini
```

Set execution bit for owner

```
chmod 100 /system/system.ini
```

Set execution bit for group

```
chmod 010 /system/system.ini
```

Set execution bit for others

```
chmod 001 /system/system.ini
```

Clear all bits for all users

```
chmod 0 /system/system.ini
```

### **See also**

• [chgrp on page 71](#) | • [chown on page 73](#)



**chown****Syntax:** `chown OWNER FILE`

Change owner of a file.

**Description**

Change owner of a file. The first argument must either be a user id or a group name. The second argument is the name of the file to alter. If the owner is a name and ends with ":" the group of the file will be set to the users group.

**Examples**

Change owner of system.ini

```
chown 1 /system/system.ini
```

Change owner of system.ini

```
chown root /system/system.ini
```

Change owner and group of system.ini

```
chown root: /system/system.ini
```

**See also**

• [chgrp on page 71](#) | • [chmod on page 72](#)

**clear****Syntax:** `clear`

Clear screen.

**Description**

Clear screen. Clears the terminal screen.

**Examples**

Clear screen

```
clear
```

**cmp****Syntax:** `cmp [OPTIONS] FILE1 FILE2`

Compare two files.

**Options**`-l`

Print each mismatching byte and offset.

**Description**

Compare two files. By default cmp will stop comparing at the first mismatch and print the offset and values at the mismatching byte. If the -l option is specified all differences will be printed.

**Examples**

Compare ish.ini and system.ini

```
cmp /system/ish.ini /system/system.ini
```

Compare ish.ini and system.ini printing all bytes that differ

```
cmp -l /system/ish .ini /system/system.ini
```

## **cp**

**Syntax:** `cp [OPTIONS] SRC DEST`

Copy file.

### **Options**

`-R`

Recursive copy, SRC and DEST must be directories.

`-v`

Verbose.

### **Description**

Copy file. If `-v` is specified each file copied will be printed.

### **Examples**

Make backup copy of system.ini

```
cp /system/system.ini system.bak
```

### **See also**

- [mv on page 90](#)

**date****Syntax:** `date [OPTIONS]`

Get/set date and time.

**Options****-n Host:**

Acquire date and time through NTP.

**-p**

Print supported time zones.

**-s YYYYMMDDhhmmss**

Set date and time.

**-d MMODhhmm-MMODhhmm**

Set daylight saving rules.

**-z TIMEZONE**

Set time zone.

**+ FORMAT**

Set the print format for date and time.

**Description**

Get/set date and time. The command will print the current time after executing any options, the printed string may be formatted by the +FORMAT option.

*HOST* may be a dot notation ip number, or a hostname if the system is configured with a valid DNS server.

*YYYYMMDDhhmmss* must have all fields included, on the following format. YYYY - year (1900-9999), MM - month (01-12), DD - Day (01-31) hh - hour (0-23), mm - minute (00-59), ss - second (00-59). Every field must be exactly as long as specified by the letters.

*MMODhhmm-MMODhhmm* must have all fields included, or entirely omitted, on the following format. Specify start of daylight saving before the '-', end after. MM - month (01-12), O - occurrence (0=last, 1=1st, 2=2nd, ..), D - day (0=Sun, 1=Mon, 2=Tue, ..), hh - hour (0-23), mm - minute (00-59). Every field must be exactly as long as specified by the letters. If all fields are omitted (only the '-' left) then daylight saving is turned off.

*TIMEZONE* should be a three letter abbreviation, an offset in hours, or a string like CET+1CEST, specifying both names and offset.

The +*FORMAT* string will be used by the strftime() function to create the string printed. No options can follow the FORMAT string, since it can contain single spaces. See documentation on the ANSI C function strftime() for details.

**Examples**

Print all valid time zones

`date -p`

Set date using a NTP service

`date -n 192.168.1.9`

Set date to 26th april 1986 1:26

`date -s 19860426012600`

Set timezone to GMT

`date -z GMT`

Set timezone to CET (using both names and offset)

`date -z CET+1CEST`

Set the daylight saving start to last Sunday in March at 1:00, and DST end to last Sunday in October at 2:00

```
date -d 03000100-10000200
```

Just print the local time

```
date
```

Print the date as YYYYMMDD

```
date +%Y%m%d
```

## debug

**Syntax:** debug [OPTIONS]

Enable/disable debug output.

### Options

**-a**

Enable debugging for all modules.

**-d**

Enable debugging for DHCP.

**-h**

Print help text.

**-I**

Enable debugging for IP layer.

**-I LEVEL**

Set debug level.

**-n**

Enable debugging all network protocols.

**-p**

Enable debugging of PPP.

### Description

Enable/disable debug output. Without any arguments the debug levels for the modules will be printed.

### Examples

Enable debug information of PPP

```
debug -l 3 -p
```

Enable all possible debug information

```
debug -l 999 -a
```

Disable all debug information

```
debug -l 0 -a
```

### See also

• [netstat on page 90](#) | • [route on page 98](#) | • [debug on page 76](#)

## **df**

**Syntax:** `df`

Print information about device usage.

### **Description**

Print information about device usage. Information about all mounted devices will be printed.

### **Examples**

Print the mounted devices

```
df
```

## **echo**

**Syntax:** `echo STRING ...`

Print a line to the terminal.

### **Options**

*STRING* The string to print.

### **Description**

Print a line to the terminal. When parsing the arguments, variables will be substituted for their values.

### **Examples**

Print hello, world

```
echo hello, world
```

Print the value of the HOME variable

```
echo $HOME
```

Print the exit status of last command

```
echo $?
```

## **exit**

**Syntax:** `exit`

End an interactive session.

### **Description**

End an interactive session. If a su session was entered, that session will be exited, else the user will be logged off.

### **Examples**

Exit this session

```
exit
```

### **See Also**

- [su on page 103](#)

## **find**

**Syntax:** `find DIRECTORY FILE`

Search for a file in the file system.

### **Options**

#### *FILE*

The file to search for.

#### *DIRECTORY*

The directory to start in.

### **Description**

Search for a file in the file system. FILE may contain wild cards ('\*' matches several characters and '?' matches only one character). The search will start in DIRECTORY and will also go through all subdirectories.

### **Examples**

Find monkey.txt in the file system

```
find / monkey.txt
```

Find all .txt files

```
find / *.txt
```

Find all .class files starting with "my"

```
find / my*.class
```

## **ftp**

**Syntax:** ftp [HOST]

FTP client.

### **Options**

*HOST* The host to connect to.

### **Description**

FTP client. This is a simple FTP client similar to those found in U\*\*X or MS Windows systems. The following commands are available

Ascii

Put transfer in ASCII mode.

Bin

Put transfer in binary mode.

Bye

Quit ftp.

cd DIR

Change directory.

dir [DIR]

List directory.

Exit

Quit ftp.

get FILE

Download file.

Help

Show help text

list [DIR]

List directory.

ls [DIR]

List directory.

Open Host

Connect to HOST.

put FILE

Upload file.

Pwd

Print current directory.

Quit

Quit ftp.

user USER

Login as USER.

### **Examples**

Just start the FTP client

```
ftp
```

Start and try to connect to a server

```
ftp ftp.sunet.se
```

## hd

Syntax: `hd [OPTIONS] FILE ...`

Print contents of a file in hexadecimal.

### Options

`-o`

Print octal dump instead of hexadecimal.

`-c`

Print printable characters, and the rest as numbers.

### Description

Print contents of a file in hexadecimal. 16 bytes will be printed on every line. The offset into the file will be printed in the leftmost column.

### Examples

Look at the hex dump of system.gpx

```
hd /system/system.gpx
```

See what non-printable characters system.ini contains

```
hd -c /system/system.ini
```

### See Also

• [od on page 91](#) | • [cat on page 70](#) | [more](#)



## help

**Syntax:** help [OPTIONS] [CMD]

Print help about command.

### Options

-s

Print a short description on all functions.

### Description

Print help about command. Without any arguments a list of all available commands will be printed.

With the -s option a short description of each command will be printed together with the command name.

Any other argument on the command line will be treated as a command and the corresponding help text will be printed.

### Examples

List all commands

```
help
```

List all commands with a short description

```
help -s
```

Show help on the "ipconfig" command

```
help ipconfig
```

## history

**Syntax:** history [OPTIONS]

Show the ish command history.

### Options

-c

Clear history.

### Description

Show the ish command history. Lists all entries in the history, these can be accessed with the up-arrow.

### Examples

Show the history

```
history
```

Clear the history list

```
history -c
```

## hostname

**Syntax:** `hostname [NAME]`

Print/set hostname.

### Options

*NAME* The name to set.

### Description

Print/set hostname. Without any arguments the current hostname will be printed.

If a hostname is present on the command line, the current host name will be exchanged for this. If the hostname is fully qualified domain name both the hostname and the domain name will be exchanged. If only a hostname is present, the old domain name will be reused.

### Examples

Print the hostname

```
hostname
```

Set the FQDN

```
hostname monkey.say.monkey.do
```

Set the hostname keeping the domain name

```
hostname gorilla
```

## id

**Syntax:** `id`

Print information about the current user.

### Description

Print information about the current user. The user id, group id and the corresponding names will be printed.

### Examples

Print user information

```
id
```

**ifconfig****Syntax:** `ifconfig frwd [OPTIONS]`

Configure the Ethernet interface

**Options***On*

Enables pass-through operations between two Ethernet interfaces

*Off*

Disables pass-through operations.

**Description**

The pass-through mode allows non-PTP traffic to be transparently fed through systems with two Ethernet interfaces, for instance to act as a "synchronization gateway" for non-PTP hosts. Before the pass-through mode can be used on M68 the command must be entered.

On its primary Ethernet port such "gateway" is connected to a network which transports data and PTP synchronization. To its secondary Ethernet port the non-PTP system is connected. The "gateway" participates in PTP activity acting as PTP slave and synchronizing its local clock. The PTP traffic is filtered out while non-PTP data are passed through to the host along with synchronization signals in form of PPS/TOD/FREQ-out.

**Note:** On M68, the secondary (host-side) Ethernet interface is available for user or applications as a general-purpose network interface.

On M64, the secondary (host-side) Ethernet interface is not available for user or applications as a general-purpose network interface, it is used only for pass-through and it is completely transparent from either host or network side.

**Note:** Before the pass through mode can be used, this command must be entered.

**Syntax:** `ifconfig [iface] vlan [OPTIONS]`Configure VLAN support on the specified network interface. If no *iface* parameter specified, the command is applied to 'enet0'.**Options***Off*

Send and receive untagged frames only

None

Send untagged frames, receive untagged and priority tagged frames.

`0 [prio_code]`

Send priority-tagged frames, receive untagged and priority tagged frames.

`<vlan_id> [prio_code]`

Send and receive vlan-tagged frames only

**Description**

Configure VLAN settings

**Syntax:** `ifconfig [iface] link [OPTIONS]`Configure link properties on the specified network interface. If no *iface* parameter specified, the command is applied to 'enet0'.**Options***[mode]*

Disable autonegotiation and force into one of the following modes: 1000F, 1000H, 100F, 100H, 10F, 10H

*auto [mode\_mask]*

Enable autonegotiation and optionally set advertised modes. If `mode_mask` is not given all supported modes will be advertised, i.e., 1000F|1000H|100F|100H|10F|10H.

### Description

tbc

**Syntax:** `ifconfig [iface] synce [OPTIONS]`

Configure 1000BASE-T Synce frequency direction on specified network interface. If no `iface` parameter specified, the command is applied to 'enet0'.

### Options

*master|slave*

Disable autonegotiation and force into either master or slave mode.

*auto master|slave*

Enable autonegotiation with either master or slave preference.

If no option given, current resolved state will be reported.

### Description

tbc

**Syntax:** `ifconfig [iface] mdio [OPTIONS]`

Read/write directly from/to MDIO registers of specified network interface. If no *iface* parameter specified, the command is applied to 'enet0'.

### Options

*<reg>*

Perform single read from MDIO register.

*<reg> <value>*

Perform single write to MDIO register

If no option given, enter into exclusive MDIO read/write mode to perform an uninterrupted series of read/writes by just specifying register addresses and values. In this mode, no other MDIO access is allowed which can cause disruption in normal operations, such as link status interrupt handling. 'q' quits exclusive MDIO read/write mode.

### Description

Configures link settings on the specified interface.

**Syntax:** `ifconfig [iface] reset [OPTIONS]`

Reset PHY on the network interface. If no *iface* parameter specified, the command is applied to 'enet0'.

### Options

*sw*

Software reset only. Only relevant when PHY's MDIO interface lines are connected.

*noinit*

Do not perform PHY initialization. Can be used when PHY needs to be configured manually using MDIO.

If no option given, HW reset will be performed.

### Description

tbc

**Syntax:** `ifconfig [iface] sfp [OPTIONS]`

Manage SFP connected to the network interface. If no *iface* parameter specified, the command is applied to 'enet0'.

**Options***raw*

Prints SFP information in raw hex format

*state*

Shows state of LOS, enable, presence and fault

*enable*

Enable SFP transmitter

*disable*

Disable SFP transmitter

*mdio <reg>*

Perform single read from MDIO register in SFP PHY

*mdio <reg> <value>*

Perform single write to MDIO register in SFP PHY

If no option given, formatted SFP information will be printed.

**Description**

tbc

**See Also**

- [stats on page 102](#)

**in****Syntax:** in PORT

Read a byte from an I/O port.

**Description**

Read a byte from an I/O port.

The PORT is a two-character value constructed as [cd][a-p]. The first character specifies if it is a data port (d) or a control port (c).

The data port is the actual data read from the port, while the control port defines which bits that are output and which that are input.

The second character defines which port is read from. Valid port values are a – p.

Ports a - j correspond to external GPIO ports

Ports k - p are internal.

**Examples**

Read the control register of port A

in ca

**See Also**

- [out on page 92](#)

**ipconfig****Syntax:** `ipconfig [iface] [OPTIONS]`(M68 only) `ipconfig all`.Print/set the settings for the TCP/IP stack. If no *iface* parameter specified, the command is applied to 'enet0'.**Options****-a ADDRESS**

Specify the IP address.

**-c**

Commit changes to flash.

**-d**

Acquire an IP address through DHCP.

**-e MAC**

Set Ethernet MAC address.

**-g GATEWAY**

Specify the gateway address.

**-h MAILHOST**

Specify the mail host address.

**-m SUBNETMASK**

Specify the netmask.

**-p DNS**

Specify the primary DNS address.

**-q DNS**

Specify the secondary DNS address.

**-r**

Release DHCP address.

**-t TIMEOUT**

Specify the DNS timeout.

**-u**

Brings interface up without assigning IP address.

**Description**

Print/set the settings for the TCP/IP stack. Without any arguments the current settings are printed. Observe that the settings will only last for one session unless the -c flag is specified.

Several options can be given at the same time, but the order of the arguments is important. The order should be address, netmask, gateway, and then any others.

If the MAC parameter is omitted or invalid with the -e option, the MAC address for the selected interface will be returned to the factory programmed one. Still only volatily, unless the -c flag is also used.

**Examples**

Show settings for all interfaces

`ipconfig all`

#### Acquire a DHCP address

```
ipconfig -d
```

#### Release a DHCP address

```
ipconfig -r
```

#### Set static ip address

```
ipconfig -a 192.168.1.127 -m 255.255.255.0 -g 192.168.1.1
```

#### Set static ip address and store it to flash

```
ipconfig -a 10.0.0.7 -m 255.0.0.0 -c
```

#### Set MAC address and store it to flash

```
ipconfig -e fc:af:6a:12:34:56 -c
```

#### Revert MAC address for enet1 back to factory setting

```
ipconfig -e -c
```

#### Store current settings to flash

```
ipconfig -c
```

#### See Also

[21. Appendix 6: Environment Variable](#)

#### kill

Kill a thread.

#### Options

ID The identifier of the thread to terminate

#### Description

Kill a thread. Terminate a thread with the given ID.

**CAUTION!! Using this command may lead to unstable system behavior.**

#### Examples

Kill thread with ID 10

```
kill 10
```

#### See Also

• [ps on page 94](#)

## ls

**Syntax:** `ls [OPTIONS] [DIRECTORY]`

List directory contents.

### Options

**-a**

Also print files starting with '.'.

**-l**

Long output.

**-R**

Recursive output.

**-F**

Classify output.

### Description

List directory contents. Without any arguments, the current directory will be listed.

### Examples

List all files in the current directory

```
ls
```

List all files in the root directory

```
ls /
```

List all files in the file system

```
ls -R /
```

## mkdir

**Syntax:** `mkdir DIR ...`

Create directory.

### Description

Create directory. All items available on the command line will be created as directories.

### Examples

Create a directory

```
mkdir foobar
```

See Also



## **mkrd**

**Syntax:** `mkrd [OPTIONS] DRIVE`

Create a ramdisk.

### **Options**

**-f**

Free an already created ramdisk

**-s [SIZE]**

Create a ramdisk with a specified size.

**DRIVE**

Drive number (A, B, ...).

### **Examples**

Create a ramdisk and mount is as drive C

```
mkrd -s 100000 C
```

## **more**

**Syntax:** `more FILE`

Show contents of a file.

### **Options**

**FILE**

The file to view.

### **Description**

Show contents of a file. The file will be shown one page at a time. Keyboard controls:

Space

Scroll down one page.

Return

Scroll down one row.

Q

Quit.

1

Rewind to line 1.

Ctrl-C

Quit.

### **Examples**

View the contents of system.ini

```
more /system/system.ini
```

### **See Also**

• [cat on page 70](#) | • [hd on page 80](#) | • [od on page 91](#)

## **mount**

**Syntax:** `mount DRIVE`

Mount FAT drive.

### **Options**

*DRIVE*

The drive to mount (A, B, ...).

*PATH*

Where to mount the device.

*Description*

Mount FAT drive. Mount a FAT drive. For example an MMC or SD card.

### **Examples**

Mount an MMC drive as volume D

```
mount D
```

### **See Also**

- [umount on page 103](#)

## **mv**

**Syntax:** `mv OLDNAME NEWNAME`

Move file.

### **Description**

Move file. This also works for entire directories.

OLDNAME and NEWNAME must be on the same volume.

### **Examples**

Move the file "foo" to file "bar"

```
mv foo bar
```

### **See Also**

- [cp on page 74](#)

## **netstat**

**Syntax:** `netstat`

Show information of current network connections.

### **Description**

Show information of current network connections. It's not possible to alter any information through this command, it only shows the state.

### **Examples**

Print network connection status

```
netstat
```

### **See Also**

- [debug on page 76](#) | • [route on page 98](#)

## **nslookup**

**Syntax:** nslookup HOST ...

Resolve a hostname through DNS.

### **Options**

*HOST*

The host to resolve.

### **Description**

Resolve a hostname through DNS. Every item on the command line will be resolved, if the item is a dot notation IP address the hostname will be resolved. If the item is a hostname the IP address will be resolved.

### **Examples**

Resolve the IP address of the host www.mycompany.com

```
nslookup www.mycompany.com
```

Resolve the name of the host 129.42.19.99

```
nslookup 129.42.19.99
```

## **od**

**Syntax:** od [OPTIONS] FILE ...

Print contents of a file in octal.

### **Options**

**-c**

Print printable characters, and the rest as numbers

### **Description**

Print contents of a file in octal. 16 bytes will be printed on every line. The offset into the file will be printed in the leftmost column.

### **Examples**

Look at the octal dump of system.gpx

```
od /system/system.gpx
```

See what non-printable characters system.ini contains

```
od -c /system/system.ini
```

### **See Also**

• [cat on page 70](#) | • [hd on page 80](#) | • [more on page 89](#)

**out****Syntax:** `out PORT DATA`

Write a byte to an I/O port.

**Options***PORT*

The port to write to.

*DATA*

The value to write.

**Description**

Write a byte to an I/O port.

The PORT is a two character value constructed as [cd][a-p]. The first character specifies if it is a data port (d) or a control port (c).

The data port is the actual data written/read to/from the port, while the control port defines which bits that are output and which that are input.

The second character defines which port that is to be written to, valid values are a - p. Ports a - j correspond to external GPIO ports A - J, ports k - p are internal.

**Examples**

Make port C all outputs and output 0xaa on it

```
out cc 0xff;out dc 0xaa
```

Make port C all inputs

```
out cc 0
```

**See Also**

- [in on page 85](#)

**passwd****Syntax:** `passwd [USER]`

Change password for user.

**Options***USER*

The user to change the password for.

**Description**

Change password for user. A regular user can only change his/hers password, while root can change any users password.

**Examples**

Change my password

```
passwd
```

Change password for user "guest"

```
passwd guest
```

**See Also**

- [useradd on page 105](#)

## **pd**

**Syntax:** `pd`

Print current drive.

### **Description**

Print current drive. The current drive is stored in the PD variable.

### **Examples**

Print current directory

```
pd
```

Print current directory

```
echo $PD
```

### **See Also**

• [pd on page 93](#) | • [cd on page 71](#)

## **ping**

Ping a node on the Internet.

### **Options**

*HOST*

Either a host name or a dot notation ip address.

*COUNT*

The number of packets to send (default is 4).

### **Description**

Ping a node on the Internet. The round trip time will be printed if a response is received.

### **Examples**

Ping host

```
ping 192.168.1.1
```

Send many ICMP packets

```
ping 192.168.1.1 9999999
```

## **ps**

**Syntax:** `ps [OPTIONS]`

Show information about processes/threads.

### **Description**

Show information about processes/threads. The following list describes the information found in each column:

#### **1. Priority**

The priority of the thread

#### **2. ID**

The unique number which identifies the thread.

#### **3. UID**

The ID of the user that created the thread.

#### **4. Type – C**

C runtime thread.

#### **5. tty**

The tty of the thread (for threads created within interactive login session).

#### **6. State**

The current execution state of the thread.

#### **7. Name**

A thread name/description.

### **Examples**

Print all threads

```
ps -a
```

### **See Also**

- [kill on page 87](#)

**ptp2****Syntax:** `ptp2 [OPTION]`

PTP version 2 protocol engine.

**Options***start MODE*

Starts PTP V2 engine in the following mode:

- 0 – GNSS disabled, only PTP (master-slave);
- 1 – GNSS reference only (master only), DEFAULT
- 2 – GNSS primary, PTP secondary (master-slave)
- 3 – GNSS secondary, PTP primary (master-slave)

*Stop*

Stops PTP V2 engine.

*pulse TIME WIDTH PERIOD COUNT*

Starts or stops schedules a pulse train output with specified parameters. DEPRECATED. Use 'pulsetime' HMI command set instead.  
 Stops the running or cancels the *pulse* scheduled pulse-train. DEPRECATED. Use 'pulsetime' HMI command set instead

*config OPTIONS*

Configures static parameters of PTP V2 Engine. If static parameters need to be changed, it should be done before starting the engine.

Options supported are:

(none)

Prints current configuration;

*-v VALUE*

vco control range, in ns/s (default 0)

*-u VALUE*

Voltage feedback, 1 - on, 0 - off (default 0).

*-t VALUE*

slave main time constant, in sec (default 100)

*-s VALUE*

slave startup time constant, in sec, 0 - off (default 0)

*-m VALUE*

master main time constant, in sec (default 3)

*-l VALUE*

master startup time constant, in sec, 0 - off

*-p VALUE*

path delay time constant, in sec (default 300)

*-q VALUE*

quality threshold, in ns, 0 - off (default 1500);

*-e VALUE*

square threshold, in ns, 0 - off (default 0).

**-f VALUE**

Minimize frequency change, 1 - on, 0 - off (default 0).

**-n VALUE**

Optimize for network type, generic, engineered or full on-path support (default generic).

**Hmi**

Enters the command processing mode of the PTP V2 engine. To print a list of available commands type "h" or "help". To leave the command processing mode and return to the shell type "q" or "quit".

**COMMAND**

Sends a command to the PTP V2 engine without entering the command processing mode.

**Description**

PTP version 2 protocol engine. This command can be used to start/stop the PTP protocol engine, and to set up a pulse train output. Commands can also be sent to the protocol engine's HMI interface. Try "ptp2 help" to print a list of available HMI commands.

The parameters to the pulse train are start time, length of the high phase of the period, length of the period and number of pulses. The first three are specified in seconds, the fourth is an integral number. If the start time is in the past, the pulse train will start with the first pulse that is in the future. If the count is set to zero, an infinite number of pulses are generated.

**Note:** This command is only available in profiles that support IEEE 1588 version 2

**Examples**

Turn on verbose loop control printouts from the engine

```
ptp2 engine debug on
```

```
ptp2 engine verbose 13
```

or

```
ptp2 e d on
```

```
ptp2e v 13
```

Start a pulse train at 1970-01-01, 00:00:00, with period 10 ms and 30% duty cycle, and let it run forever.

```
ptp2pulse 0.0 0.003 0.01 0
```

**pwd**

**Syntax:** pwd

Print current directory.

**Description**

Print current directory. The current directory is stored in the PWD variable.

**Examples**

Print current directory

```
pwd
```

Print current directory

```
echo $PWD
```

**See Also**

• [pwd on page 96](#) | • [cd on page 71](#)



## **reboot**

**Syntax:** `reboot [OPTIONS]`

Reboot system.

### **Options**

`-h`

Hang system (don't restart) after power down.

`-f`

Force reboot (don't require user to acknowledge reboot).

`-q`

Quick shutdown. Only do the most necessary before power down.

`-r`

Reboot system after power down (default).

`-t TIME`

Wait TIME seconds before shutdown.

### **Description**

Reboot system. When `-h` is specified, the execution will stop and the red LED D300 will light up when the power down is complete.

The usage of this command is limited to the root user.

### **Examples**

Reboot system, requiring an acknowledge

```
reboot
```

Reboot system, without asking any questions

```
reboot -f
```

Bring system down without rebooting

```
reboot -f -h
```

## **recv**

**Syntax:** `recv`

Receive a file from the PC through the serial cable.

### **Description**

Receive a file from the PC through the serial cable. Since the protocol for transferring files between the target system and the PC is proprietary, this command requires a serial connection to be made with Developer or Commander utility. When the file has been transferred a MD5 checksum is calculated to ensure that the transfer was error free.

### **Examples**

Set shell in receive mode waiting for a file

```
recv
```

### **See Also**

- [send on page 99](#)

## **rm**

**Syntax:** `rm [OPTIONS] FILE ...`

Remove file.

### **Options**

`-r` Remove files recursively.

### **Description**

Remove file. The files on the command line will be removed. If the `-r` option is available and a directory is present on the command line the directory and its contents will be removed recursively.

### **Examples**

Remove foo.bar

```
rm foo.bar
```

Remove /root directory and contents recursively

```
rm -r /root
```

## **rmdir**

**Syntax:** `rmdir DIR ...`

Remove directory.

### **Description**

Remove directory. All directories available on the command line will be removed.

### **Examples**

Remove directory

```
rmdir /root
```

### **See Also**

• [mkdir on page 88](#) | • [rm on page 98](#)

## **route**

**Syntax:** `route`

Print routing table information.

### **Description**

Print routing table information. It's not possible to alter the routing information through this command. When debugging PPP connections this command may give additional information.

### **Examples**

Print routing information

```
route
```

### **See Also**

• [debug on page 76](#) | • [netstat on page 90](#)

**send****Syntax:** `send FILE ...`

Send a file to the PC over the serial cable.

**Description**

Send a file to the PC over the serial cable. Since the protocol for transferring files between the target system and the PC is proprietary, this command requires a serial connection to be made with Developer or Commander utility.

When the file has been transferred a MD5 checksum is calculated to ensure that the transfer was error free.

**Examples**

```
send ish.ini
```

```
send ish.ini
```

```
send system.ini and ish.ini
```

```
send system.ini ish.ini
```

**See Also**

- [recv on page 97](#)

**setenv****Syntax:** `setenv [NAME] [VALUE] ...`

Define an environment variable.

**Options**

*NAME* The variable name.

*VALUE* The variable value.

**Description**

Define an environment variable. Without any arguments all defined variables will be printed. If only one argument is available, the corresponding variable will be removed.

If two or more arguments are available, the first will be treated as the variable name and the rest will be the variable value.

**Examples**

Print all variables

```
setenv
```

Clear the value of the HOME variable

```
setenv HOME
```

Reconfigure your HOME

```
setenv HOME /system
```

## **sleep**

**Syntax:** `sleep SECONDS`

Make process sleep.

### **Description**

Make process sleep. Delay execution of the current process for the specified time. Useful primarily in scripts.

### **Examples**

Sleep for 10 seconds

```
sleep 10
```

## **source**

**Syntax:** `source FILE`

Execute the contents of a file.

### **Description**

Execute the contents of a file. The contents of FILE will be read and every line will be interpreted as an ish (shell) command and executed.

Another way to achieve this functionality is to set the execute bit of the script file (with `chmod`) and simply type its name. There is no PATH variable, so the absolute path to the script must be specified or standing in the same directory as the script will also work.

### **Examples**

Execute the contents of `/system/startup.ini`

```
source /system/startup.ini
```

Execute the contents of `/system/startup.ini`

```
cd /system;chmod 733 startup.ini;startup.ini
```

### **See Also**

- [chmod on page 72](#)

## **startserver**

**Syntax:** `startserver [OPTIONS]`

Start firmware servers.

### **Options**

`-s` Start the serial server.

`-t` Start the telnet server.

`-f` Start the FTP server.

`-d` Start the SSH server.

### **Description**

Start firmware servers. The corresponding server will be started (if not already running). The configuration will be read from the `system.ini` file.

### **Examples**

Start serial and telnet servers

```
startserver -s -t
```

### **See Also**

• [stats on page 102](#) | • [startserver on page 101](#) | `system.ini`

[21. Appendix 6: Environment Variable](#)

**stats****Syntax:** `stats [OPTIONS]`

Show information/statistics about the system.

**Options**

- `-b` Show build date.
- `-c` Show C heap information.
- `-d` Show disk usage information.
- `-h` Show hardware version information.
- `-f` Show firmware version information.
- `-n` Show software version information.
- `-r` Show when system was rebooted.
- `-s` Show server information.
- `-u` Show uptime information.
- `-v` Show version information (same as `-h -f -n`).
- `-w` Show disk wear information.

**Description**

Show information/statistics about the system. By default all information will be shown.

**Examples**

Show uptime and version information

```
stats -v -u
```

**See Also**

- [df on page 77](#)

**stopserver****Syntax:** `stopserver [OPTIONS]`

Stop firmware servers.

**Options**

- `-s` Stop the serial server.
- `-t` Stop the telnet server.
- `-f` Stop the FTP server.
- `-d` Stop the SSH server.

**Description**

Stop firmware servers. The process that receives incoming calls will be killed. Already running sessions will not be terminated, these sessions will end when the user logs off.

**Examples**

Stop all servers

```
stopserver -s -f -t
```

**See Also**

- [startserver on page 101](#) | • [stats on page 102](#)

## **su**

**Syntax:** `su [USER]`

Become another user.

### **Options**

*USER* The user to become.

### **Description**

Become another user. If no user is specified on the command line, root is assumed. If the current user is root, "su" won't prompt for the new user's password.

### **Examples**

Become root

```
su
```

Become "guest"

```
su guest
```

## **touch**

**Syntax:** `c`

Update time stamp of a file.

### **Options**

*FILE* The file to update.

### **Description**

Update time stamp of a file. The "last modified" time stamp of the file(s) will be set to the current time.

### **Examples**

Change time stamp of system.ini

```
touch /system/system.ini
```

## **umount**

**Syntax:** `umount DRIVE`

Unmount a drive.

### **Options**

*DRIVE* The drive to unmount.

### **Description**

Unmount a drive. The DEVICE must be a device node such as: /dev/mmca0.

### **Examples**

Unmount drive C

```
umount C
```

### **See Also**

- [mount on page 90](#)

**unalias****Syntax:** `unalias ALIAS ...`

Remove an alias definition.

**Options**

ALIAS The alias to remove.

**Description**

Remove an alias definition. All arguments on the command line will be treated as aliases and attempted to be removed.

**Examples**

Remove alias for "ls"

```
unalias ls
```

**See Also**

- [alias on page 70](#) |
- [setenv on page 99](#)

**update****Syntax:** `update [OPTIONS] [HOST] PATH`

Update the firmware through TFTP or FTP.

**Options****-t** *TIMEOUT*

Receive data timeout in milliseconds. Default is 10000 ms.

**-f** *USER:PASSWORD*

Username and password if HOST is an FTP server.

*HOST*

Can either be a hostname or a dot notation IP address.

*PATH*

The path on the TFTP or FTP server.

**Description**

Update the firmware through TFTP or FTP. This is an alternative way to update the firmware. The file to download must be a valid .snp or .snp2 file. Before starting the updating procedure, a "magic" number will be checked in the downloaded file, if the magic number is bad the update will halt.

While downloading the file there will be no output to the terminal (such as a progress bar). The downloading takes about 30 seconds (depending on network load).

If HOST is omitted, FILE is assumed to be on the local file system. In this case, the full path must be specified.

**Examples**

Fetch a myupdate1.snp file from a server

```
update 192.168.1.9 /myupdate1.snp
```

Update from local file myupdate2.snp

```
update A:/root/myupdate2.snp
```



**useradd****Syntax:** `useradd [OPTIONS] USER`

Add a user to the user database.

**Options**`-d DIR` The users home directory.`-u UID` The user ID (numeric).`-g GID` The group ID (numeric).`-s SHELL` The shell.`-c COMMENT` A comment.**Description**

Add a user to the user database. By default the user will be assigned the next ID in sequence and the GID for the user group.

If not specified, the home directory for the user will be left empty.

The SHELL field isn't used at all on the system, this is only available to be compatible with other systems. The COMMENT field is simply comment in the passwd.ini file.

**Examples**

Add user "ford" to the user database

```
useradd ford
```

**See Also**

- [userdel on page 105](#) |

**userdel****Syntax:** `userdel USER`

Remove a user from the user database.

**Description**

Remove a user from the user database. Removes the user from the passwd.ini file.

**Examples**

Remove user "ford" from the user database

```
userdel ford
```

**See Also**

- [useradd on page 105](#) |

## **wall**

**Syntax:** wall [OPTIONS]

Send message to all users.

### **Options**

-n Suppress the automatic header that normally is sent.

### **Description**

Send message to all users. This command works like the write command, but with the difference that the message is sent to all logged in users.

### **Examples**

Send message to all logged in users

```
wall
```

### **See Also**

- [write on page 107](#)

## **who**

**Syntax:** who

Show a list on logged in users.

### **Description**

Show a list on logged in users. The list will contain information about all logged in users.

The user name, the controlling tty and the time of login.

### **Examples**

Show who are logged on

```
who
```

Show who I am

```
who am i
```

## **write**

**Syntax:** write USER [TTY]

Send message to a specific user.

### **Options**

*USER* The user who should receive the message.

*TTY* If the user is logged in with several sessions, this option chooses which tty to send the message to.

### **Description**

Send message to a specific user. Each line will be sent to the receiving user when return is pressed. The message will end Send message to a specific user. Each line will be sent to the receiving user when return is pressed. The message will end when the sequence 'CR', '.', 'CR' is read from the keyboard.

### **Examples**

Send a message to user "guest"

```
write guest
```

Send a message to user "guest" who is logged in at several ttys, among them tty3

```
write guest tty3
```

### **See Also**

- [wall on page 106](#) | • [who on page 106](#)

## 18. Appendix 3: PTP Command Reference

All commands can be issued by a user from a serial terminal or a telnet session. The output from any command is redirected to the output device it was issued from.

In contrast to commands, the debug output can have only one output device. So enabling it on one device (or in one session) will automatically disable it on another.

### PTP Commands

#### ptp2 start

```
ptp2 start <mode> <option>
```

Starts PTP Engine in the following mode:

- 0 – GNSS disabled, only PTP (master-slave),
- 1 – GNSS reference only (master only). Default mode,
- 2 – GNSS primary, PTP secondary (master-slave),
- 3 – GNSS secondary, PTP primary (master-slave),
- 4 – GNSS debug.

Supported options are:

--silent: no unsolicited console output until HMI debug is turned on

**Note:** In this context "GNSS" means GPS/GNSS.

#### ptp2 stop

```
ptp2 stop
```

Stops PTP Engine.

#### ptp2 hmi

```
ptp2 hmi
```

Enters the HMI command processing mode.

To quit the HMI command processing mode type q.

**ptp2 config**

```
ptp2 config <options>
```

Configures oscillator and loop control related static parameters. Configuration options are:

- v [value] – VCO control range, in ns/s (default 6000),
- d [value] – VCO static drift, in ns/s (default 0),
- u [1|0]: voltage feedback, 1 - on (default 1)
- t [value] – slave main time constant, in sec (default 100),
- s [value] – slave startup time constant, in sec, 0 - off (default 0),
- m [value] – master main time constant, in sec (default 3),
- l [value] – master startup time constant, in sec, 0 - off (default 0),
- p [value] – path delay time constant, in seconds (default 300),
- q [value] – quality threshold, in ns, (default 1500),
- e [value] – square threshold, in ns, 0 - off (default 0).
- f [0|1] – minimize frequency change, 1 – on, 0 – off (default 0).
- n [g|e|f] – optimize for either generic ('g'), engineered ('e') or full on-path support ('f') network (default 'g').

**ptp2**

```
ptp2 <command>
```

Sends a single HMI command to the PTP Engine.

**HMI Commands**

The HMI command set is a part of the PTP Protocol implementation. They can be issued either directly from the HMI processing mode or through the ptp2 prefix from the ISH command prompt.

At the prompt, you can type either `hmi` or `help`:

```
A:/root>ptp2 hmi
```

or

```
A:/root> ptp2 help
```

Each command and subcommand has its short form or alias (given in parentheses in front of each command). An alias can be used in the same manner as full form of command.

For example:

```
A:/root>ptp2 hmi
```

or

```
A:/root> ptp2 h
```

**Clock Commands****clock**

Print the list of clock subcommands.

```
(c)clock
```

**clock info**

Prints clock description.

```
(c)clock (i)info
```

**clock user**

Change user description.

```
(c)clock (u)user <string>
```

With this command it is possible to change "user description" "product description", "revision data" and mfg OUI.

<string> is defined as

```
<user description> | prod <company> <product> <identifier> | rev <hw_ver> <fw_ver> | mfg <oui>
```

See the following table for the mapping of these fields to IEEE 1588 TLV fields. All of these can be viewed/confirmed with the ptp2 clock info command.

**Table 18.1. Mapping clock user <string> parameters to IEEE 1588 TLV**

Field	IEEE 1588 TLV	Length (bits)	Description
<User description>	userDescription	Max: 128	A user defined name
<Company> <Product> <Identifier>	productDescription	Max 64	The <company>, <product>, and <Identifier> are concatenated with a ";" separator to form the productDescription. If either <Product> or <Identifier> are omitted, the default system values will be used.
<hw_Ver> <fw_ver>	revisionData	Max 32	These two fields, plus the internal software version information form the PTP field < revisionData> with ";" separators.
<oui>	clockIdentity	64	This is the manufacturer's owned OUI. Please refer to IEEE 1588 sections 7.5.2.2.2 and 7.5.2.2.3.

If any of the text phrases has a space, the phrase must be enclosed in quotes.

**clock mode**

Change clock mode (for M68 only).

```
(c)clock (m)mode <oc | bc>
```

**clock profile**

Change clock profile.

```
(c)clock (f)profile <default|g8265|g8275.1|g8275.2|power [v2]>
```

Changing the clock profile will set the clock and all ports to work according to the specified profile. Use this command rather than the port profile command for proper operation according to the desired profile.

**clock twostep**

Change twostep mode.

```
(c)clock (t)twostep <on | off>
```

**clock quality**

Change clock quality.

```
(c)clock (q)quality <class> <accuracy> <variance>
```

**clock priority**

Change priority 1 or 2.

```
(c)clock (p)priority 1 | 2 <value>
```

**clock domain**

Change domain number.

```
(c)clock (d)domain <number>
```

**clock slaveonly**

Change slave-only mode.

```
(c)clock (s)slaveonly <on | off>
```

**Port Commands****port**

Print the list of port subcommands.

```
(p)port <port_number>
```

**port state**

Print port state.

```
(p)port <port_number> (s)state
```

**port mechanism**

Change or disable delay mechanism.

```
(p)port <port_number> (m)mechanism <e2e | p2p | disable>
```

**port protocol**

Change network protocol.

```
(p)port <port_number> (p)protocol <udp | eth | udp6>
```

**port compatibility**

Change V1 HW compatibility mode.

```
(p)port <port_number> (c)compatibility <on | off | auto <timeout>>
```

**port asymmetry**

Change delay asymmetry. Value is expressed in seconds.

```
(p)port <port_number> (a)asymmetry <value>
```

**port sync**

Change log sync interval.

```
(p)port <port_number> (si)sync <value>
```

**port delay**

Change min log delay request interval.

```
(p)port <port_number> (di)delay <value>
```

**port pdelay**

Change min log pdelay request interval.

```
(p)port <port_number> (pi)pdelay <value>
```

**port announce**

Change log announce interval.

```
(p)port <port_number> (ai)announce <value>
```

**port receipt**

Change announce receipt timeout.

```
(p)port <port_number> (rt)receipt <value>
```

**port unicast**

Enable unicast operation.

```
(p)port <port_number> (u)unicast <master | slave | disable>
```

**port profile**

Change PTP profile for the specified port. For proper operation, use the same profile on both ports, in which case it's more convenient to use the clock profile command.

```
(p)port <port_number> (f)profile <default|g8265|g8275.1|g8275.2|power [v2]>
```

**port dscp**

Differentiated Services Code Point in PTP messages.

```
(p)port <port_number> (d)dscp <value>
```

**Unicast Commands**

See [10.6 Unicast Operations](#) for more details.

**unicast**

Print the list of unicast subcommands.

```
(u)unicast <port_number>
```

**unicast status**

Print unicast port status information.

```
(u)unicast <port_number> (s)status
```



**unicast negotiation**

Change unicast negotiation state.

```
(u)unicast <port_number> (n)negotiation <on|off>
```

**unicast node**

Change master discovery table or static slave table.

```
(u)unicast <port_number> node (m)master [add|del] <address> [cid] <pri> <si> <di|x> [domain]
```

or

```
(u)unicast <port_number> node (s)slave [add|del] <address> <ai> <si> <di|x>
```

**unicast query**

Change unicast discovery query interval.

```
(u)unicast <port_number> (q)query <value>
```

**unicast duration**

Change unicast transmission duration.

```
(u)unicast <port_number> (d)duration <value>
```

**unicast limit**

Change total unicast messages rate limit.

```
(u)unicast <port_number> (l)limit <value>
```

**unicast filter master**

Change master acceptance filter.

```
(u)unicast <port_number> (f)filter (m)master [add|del] <address|*> [cid] [x|priority][si][di|x]]
```

**unicast filter slave**

Change slave acceptance filter.

```
(u)unicast <port_number> (f)filter (s)slave [add|del] <address|*> [cid] [x|[ai][si][di|x]]
```

**Power Profile Commands**

Before using these commands, set the clock to power profile using the clock profile command.

```
(c)clock <port_number> (f)profile power [v2]
```

**power**

Print the list of power profile subcommands.

```
(pw)power
```

**power gmid**

Change the grandmaster ID.

```
(pw)power (g)gmid <0..65535>
```

**power inacc**

Change grandmaster and network time inaccuracy (in ns).

```
(i)inacc <grandmaster> <network>
```

**G.8265 Profile Commands**

Before using these commands, set the clock to G.8265 profile using the clock profile command.

```
(c)clock (f)profile g8265
```

**g8265**

There are no profile-specific parameters to configure for G.8265.

```
(g1)g8265
```

**G.8275 Profile Commands**

Before using these commands, set the clock to G.8275.1 or G.8275.2 profile using the clock profile command.

```
(c)clock (f)profile g8275.1
```

or

```
(c)clock (f)profile g8275.2
```

**g8275**

Print the list of G.8275 profile subcommands.

```
(g2)g8275
```

**g8275 steps removed**

Change max steps removed.

```
(g2)g8275 (s)steps <1..255>
```

**g8275 priority**

Change clock local priority.

```
(g2)g8275 <port_number> (p)priority <1..255>
```

**g8275 port priority**

Change port local priority.

```
(g2)g8275 <port_number> (p)priority <1..255>
```

**g8275 port masteronly**

Turn master-only attribute on or off.

```
(g2)g8275 <port_number> (m)masteronly <on|off>
```

**Dataset Commands****dataset**

Print the list of dataset subcommands.

```
(d)dataset
```

**dataset default**

Print default dataset.

```
(d)dataset (d)default
```

**dataset current**

Print current dataset.

```
(d)dataset (c)current
```

**dataset parent|master**

Print parent dataset.

```
(d)dataset (m)parent | master
```

**dataset time**

Print time properties dataset.

```
(d)dataset (t)time
```

**dataset port**

Print port dataset.

```
(d)dataset (p)port <port_number>
```

**dataset foreign**

Print the list of foreign master datasets.

```
(d)dataset (f)foreign <port_number>
```

**dataset unicast**

Print the list of unicast master/slave node datasets.

```
(d)dataset (u)unicast <port_number>
```

**dataset power**

View the profile-specific values obtained by a slave from its master

```
(d)dataset (pw)power
```

**dataset g8265**

Print G.8265 profile dataset.

```
(d)dataset (g1)g8265
```

**dataset g8275**

Print G.8275 profile default dataset.

```
(d)dataset (g2)g8275
```

**dataset g8275 port**

Print G.8275.1 profile port dataset.

```
(d)dataset (g2)g8275 <port_number>
```

**Time Commands****time**

Print the list of time subcommands.

```
(t)time
```

**time info**

Print time information.

```
(t)time (i)info
```

**time ptp**

Change time of the PTP clock.

```
(t)time ptp <YYYY-MM-DD HH:MM:SS.NNNNNNNNN>
```

The time parameter should represent the PTP (TAI) time in date-time format. If the time parameter is missing the command prints current time in the same format.

When the clock is synchronizing with either a PTP master or with a GNSS time this command will have no effect.

**time arb**

Change time of the PTP clock.

```
(t)time arb <+/-SSSSSSSS.NNNNNNNNN>
```

The time parameter should represent the PTP (TAI) time in seconds-nanoseconds format. If the time parameter is missing the command prints current time in the same format.

When the clock is synchronizing with either a PTP master or with a GNSS time this command will have no effect.

**time ntp**

Change time of the PTP clock.

```
(t)time ntp <server>
```

This command obtains the current time and leap second indicators from an NTP server specified by its name or IP address. Then it converts the received time to the PTP (TAI) time using the UTC offset value configured locally and sets the time of the clock. If the parameter is missing the command prints current time converted to UTC.

When the clock is synchronizing with either a PTP master or with a GNSS time this command will have no effect.

**time scale**

Change timescale.

```
(t)time (t)timescale <ptp | arb>
```

**time offset**

Change UTC offset.

```
(t)time (o)offset <value>
```

**time leap**

Change leap second indications.

```
(t)time (l)leap 59|61 <0 | 1>
```

Print UTC leap table or save/load it into/from file.

```
(t)time (l)leap <save | load>
```

All scheduled leap seconds are stored in the leapsec.ini file in the /system directory on the target system.

**time update**

Keep system RTC in sync with clock time.

```
(t)time (u)update <on | off>
```

**time holdover**

Change holdover interval.

```
(t)time (h)holdover <interval>
```

**GNSS Interface Commands****gps**

**Note:** The command is "gps", but applies equally to GPS or GNSS receivers.

Print the list of GNSS Interface subcommands.

```
(g)gps
```

**gnss status**

Print the GNSS Interface status and configuration.

```
(g)gps (s)status
```

**gnss nmea**

Print NMEA position/satellites information.

```
(g)gps (n)nmea
```

**gnss ppsin**

Enable PPS-input signal reception.

```
(g)gps (p)ppsin <enable | disable>
```

**gnss todin**

Enable TOD-input signal reception.

```
(g)gps (t)todin <enable | disable>
```

**gnss todin**

Change non-present or invalid TOD-in handling.

```
(g)gps (t)todin <required | optional>
```

The presence of a valid TOD-input is required by default for the system to synchronize to the PPS/TOD-input and to become a GNSS-based master. To make the TOD-input optional, i.e. to allow the system to synthonize to the PPS-input use `gnss todin optional`. This affects the behavior of the NMEA RMC Void flag handling (see `gps void`). If configured to 'synthonize' while the syntonization is now allowed, the result will be the same as 'holdover', i.e., TOD-input is considered invalid and the system goes to holdover.

**gnss delay**

Change PPSIN cable delay compensation. Value is expressed in nanoseconds.

```
(g)gps (d)delay <value>
```

**gnss format**

Change TOD-input format.

```
(g)gps (f)format <nmea | ascii | cm>
```

**gnss void**

Changes NMEA RMC void flag handling.

```
(g)gps (v)void <ignore | syntonize | holdover>
```

**gnss interface**

Change TOD-input interface.

```
(g)gps (i)interface <default | com[n]> <baudrate>
```

**gps class**

Change clock class mapping.

```
(g)gps (c)class <active> <holdover> <free>
```

It is possible to set the clock class mapping for holdover and free-running states to 255. This will make the system to stop being the master even if no better clock available to prevent distributing its time/freq to slave-only clocks which may present on the network. This feature makes sense in mode 1 only.

**gps message**

Send message to GNSS/GPS receiver via TOD interface.

```
(g)gps (m)message <string>
```

<string> may be:

\* NMEA ASCII command, starting with a \$ and enclosed in single quotes (')

\* Binary string, encoded in hexadecimal (upper or lower case), starting with 0x, spaces maybe added. The message is converted to binary and sent as is, no adding of checksum, <CR>, <LF>, etc.

Examples:

`ptp2 g m 0x010203aabb` will send the binary bytes 0x01, 0x02, 0x03, 0xAA, 0xBB

`ptp2 g m "0x010203 AABB"` will send the binary bytes 0x01, 0x02, 0x03, 0xAA, 0xBB

`ptp2 g m "0x1 2 3 aa bb"` will send the binary bytes 0x01, 0x02, 0x03, 0xAA, 0xBB

**PPS/TOD-Output Interface Commands**

**pulsetime**

Print the list of PPS/TOD-output Interface subcommands.

```
(pt)pulsetime
```

**pulsetime status**

Print the PPS/TOD-output status and configuration.

```
(pt)pulsetime (s)status
```

**pulsetime pulse**

Change PPS-output pulse parameters. Period and width are expressed in microseconds ( $\mu$ s).

```
(pt)pulsetime (p)pulse <period> <width>
```

**pulsetime delay**

Change TOD-output delay.

```
(pt)pulsetime (d)delay <value>
```

**pulsetime format**

Change TOD-output format.

```
(pt)pulsetime (f)format <nmea | ascii | cm>
```

**pulsetime void**

Change NMEA RMC Active/Void flag behavior.

```
(pt)pulsetime (v)void <never | free | holdover>
```

**pulsetime zone**

Change NMEA ZDA timezone/daylight field.

```
(pt)pulsetime (z)zone <none | tz | dst>
```

**pulsetime interface**

Change TOD-output interface.

```
(pt)pulsetime (i)interface <default | com[n]> <baudrate>
```

**pulsetime todout**

Enable TOD-output.

```
(pt)pulsetime (t)todout <enable | disable>
```

**pulsetime mode**

Change PPS/TOD-output mode.

```
(pt)pulsetime (m)mode <1 | 2 | 3>
```

**pulsetime error**

Change PPS/TOD-output error tolerance.

```
(pt)pulsetime (e)error <value>
```

**pulsetime autorestart**

Change PPS/TOD-output error tolerance.

```
(pt)pulsetime (a)autorestart <on | off>
```

**pulsetime start**

Start PPS/TOD-output.

```
(pt)pulsetime start <YYYY-MM-DD HH:MM:SS.NNNNNNNNN> <count>
```

**pulsetime stop**

Stop PPS/TOD-output.

```
(pt)pulsetime stop
```

**SNTP Server Commands****nettime**

Print the list of SNTP server subcommands.

```
(nt)nettime
```

**nettime status**

Print the SNTP server status and configuration.

```
(nt)nettime (s)status
```

**nettime mode**

Change the SNTP server mode.

```
(nt)nettime (m)mode <1..4>
```

1 = unicast, 2 = manycast, 3 = broadcast, 4 = mixed

**nettime poll**

Change poll/broadcast interval.

```
(nt)nettime (p)poll <1..64>
```

**nettime start**

Start the SNTP server.

```
(nt)nettime start
```

**nettime stop**

Stop the SNTP server.

```
(nt)nettime stop
```

**Engine Commands****engine**

Print the list of engine subcommands.

```
(e)engine
```



**engine init**

Reinitialize clock. Currently not supported.

```
(e)engine (i)init default | nvm
```

**engine port**

Change port operational status.

```
(e)engine (p)port enable | disable
```

**engine esmc**

Enable ESMC message transmission.

```
(e)engine (e)esmc <auto | 0..15 | off>
```

**engine esmc map**

Change clock class to SSM code mapping.

```
(e)esmc map <class> <ssm>
```

To see the mapping, type ptp2 e e map

**engine nvm**

Change configuration in non-volatile storage. Currently not supported.

```
(e)engine (n)nvm save | reset
```

**engine debug**

Turns on/off debug output.

```
(e)engine (d)debug <on | off>
```

**engine verbose**

Change verbose level for debug output.

```
(e)engine (v)verbose <level>
```

**engine monitor**

Enable clock monitor program.

```
(e)engine (m)monitor <off | on <port>>
```

**engine freqout**

Selects frequency on the FREQOUT pin.

```
(e)engine (f)freqout <5 | 10 | 20 | 25>
```

**engine asymmetry**

Change PHY asymmetry compensation. Value is expressed in nanoseconds.

```
(e)engine (a)asymmetry <value>
```

## LOCK command

lock

Use the PF3/LOCK pin to indicate the lock state of the PTP servo.

(1) lock (p)pin <enable|disable>: Enable/disable lock indicator. The default is disable.

(1) lock (s)status: Lock indicator status

(1) set <auto|<error> <slope>>: Change set conditions. The default is auto.

(1) lock clear <auto|<error> <slope>>: Change clear conditions. The default is auto.

## 19. Appendix 4: Troubleshooting LED Error Codes

This section explains the behavior of the green LED (D102) on the module and how to troubleshoot a problem encountered. The LED signals an error condition at boot up, if the boot loader cannot load a valid executable. The error code is a number of green pulses, then a slightly longer delay, then the error code, and so on.

Number of Pulses	Description
2	The boot loader can't find a valid file system on the flash.  Solution: Try to reformat the flash.
3	Error initializing the flash or error reading a page from the flash.  Solution: Try to reformat the flash.
4	Error reading boot sectors.  Solution: Try to reboot.
5	Malloc error for some reason. malloc() returned a NULL pointer.  Solution: Try to reboot.
6	The /system/system.gpx file can't be read  Solution:  Try to reboot.  Reinstall firmware by running the Flash-it project in the Developer IDE
7	The /system/system.gpx file can't be found.  Solution:  Try to reboot.  Reinstall firmware by running the Flash-it project in the Developer IDE.
8	Error reading data from the serial port.  Solution: Try to reboot.
9	Can't find /system/boot.ini  Solution:  Try to reboot.  Reinstall firmware by running the Flash-it project in the Developer IDE.
10	Can't read /system/boot.ini  Solution:  Try to reboot.  Reinstall firmware by running the Flash-it project in the Developer IDE.
11	/system/boot.ini has a bad format  Solution: Reinstall firmware by running the Flash-it project in the Developer IDE.

## 20. Appendix 5: Command Shell Limitations

There are a number of limitations to the command shell:

1. There is no possibility to redirect input/output.
2. stdout and stderr are identical.
3. The checking of file permissions (read/write) is only done in some cases so it may be possible to overwrite a file by accident.
4. The command line length is limited to 200 characters, and can only contain a maximum of 29 arguments.
5. It is only possible to have 10 simultaneously interactive shells.
6. Command line flags can't be bundled.

## 21. Appendix 6: Environment Variable

The system contains a number of predefined environment variables. These variables can be set in configuration files that are read and parsed at boot time and when logging on to the system.

### Predefined Environment Variables

The following table lists the predefined environment variables. Some of the variables are configuration variables and effects the system. These variables can be set in system.ini. Others only effects the login session and they can be set in ish.ini and \_ish.ini.

Most of the variables can be changed through shell commands. If a variable in system.ini is changed by hand, the system must be rebooted to make the changes take effect.

All TCP/IP settings can be altered through the ipconfig command, which won't require a reboot

Name	File	Description
?	N/A	The return value from the latest command. A return value of zero is a successful return, while a value != zero indicates an error.
DAYLIGHT	system.ini	Defines how the daylight saving is used. See DAYLIGHT for a description.
DEBUG	system.ini	Sets the initial debug level. If the value is zero, or the variable doesn't exist at all, debugging is disabled (can be enabled by hand by using the debug command). The higher the number is, the more verbose the output will be.
FTP	system.ini	Defines if the FTP server should be enabled or disabled at boot time. Possible values: enable or disable. Default is enable. To manually start or stop the FTP server use the startserver and stopserver commands.
FTP_CMD_PORT	system.ini	Defines TCP port number for FTP control channel. Default value is 21.
FTP_DATA_PORT	system.ini	Defines TCP port number for FTP data channel. Default value is 20.
FTP_PRIO	system.ini	Defines the priority of the FTP server. Default value is 104. The priority of each FTP session will be the FTP server's priority + 1.
GID	N/A	The users group id according to group.ini. This variable will reflect the current user.
HOME	N/A	The user's home directory. Read from passwd.ini. This is what the current directory will be when executing cd without any arguments. This is also the starting directory for every login session. If the directory doesn't exist the starting directory will be set to /.
LOGIN_TIMEOUT	system.ini	The number of seconds before automatically logging out. Set to zero to disable. This variable defines the timeout for both Telnet and FTP sessions. Default is 900 seconds (15 minutes).
MAC_ADDRESS	system.ini	The Ethernet MAC address, such as fc:af:6a:12:34:56. If this environment variable is not set, the MAC address will be the factory programmed one.
MAC_ADDRESS_SEC	system.ini	The Ethernet MAC address for the secondary interface. If this environment variable is not set, the MAC address will be the factory programmed one.
PS1	ish.ini, \$HOME/.ish.ini	The prompt to use. Default is "> ". Other popular settings might be:  setenv PS1 "\\$PD:\\$PWD> "  setenv PS1 "\\$USER:\\$PWD> "

Name	File	Description
PWD	N/A	The current directory.
ROWS	ish.ini, \$HOME/.ish.ini	Specifies the number of rows in a terminal window.
SERIAL	system.ini	Defines the state of the serial server at boot time. Possible values: enable or disable. Default is enable. To manually start or stop the serial server use the and commands.
SERIAL_LOGIN	system.ini	Set to enable if logging in via the serial port should require a username and password. Possible values are enable or disable. Default is enable. It is recommended to keep this set to enable.
SERIAL_PORT	system.ini	The port that the serial server should use. Possible values: COM[1-4]. Default is COM1.
SERIAL_SPEED	system.ini	The speed of the serial port. Numeric value of the baud rate, such as 9600, 19200 etc. Default is 115200.
SSH	N/A	Unlike other servers the SSH server can only be started manually using command.
SSH_PRIO	system.ini	Defines the priority of the SSH server. Default value is 104. The priority of each SSH session will be the SSH server's priority + 1.
SSH_PORT	system.ini	Defines TCP port number for SSH server. Default value is 22.
TCPIP_ADDRESS	system.ini	<p>The IP address. May be one of the following values:</p> <p>Static ip, such as 192.168.1.222</p> <p>The string "dhcp" which indicates the use of DHCP.</p> <p>The special value 255.255.255.255 which also indicates the use of DHCP.</p> <p>The string "disable" disables all TCP/IP communication.</p> <p>The string "enable" enables TCP/IP communication, but doesn't initialize the Ethernet interface.</p> <p>By default an address is will be acquired through DHCP. If a DHCP request fails the system will fall back to a zero configuration scheme.</p>
TCPIP_PRIM_DNS	system.ini	The DNS in dot notation. Set to 0.0.0.0 to disable the use of DNS. If the IP address is acquired through DHCP and the DHCP server supplies a DNS server, the DNS will be set to the value fetched via DHCP.
TCPIP_SEC_DNS	system.ini	The DNS in dot notation. Set to 0.0.0.0 to disable the use of a secondary DNS. This value will not be set by the DHCP client.
TCPIP_SEC_TO	system.ini	The DNS timeout in seconds. Set to 0 to disable timeout. Default is 10 seconds.
TCPIP_GATEWAY	system.ini	The gateway in dot notation. Set to 0.0.0.0 to disable. For DHCP configured systems this will be supplied by the DHCP server.
TCPIP_MAILHOST	system.ini	The mail host in dot notation. Set to 0.0.0.0 to disable. If this value is to be used it must be set explicitly, it is not acquired via DHCP for DHCP configured systems.
TCPIP_NETMASK	system.ini	The netmask in dot notation. For DHCP configured systems this will be supplied by the DHCP server.

Name	File	Description
TELNET	system.ini	Defines the state of the TELNET server at boot time. Possible values: enable or disable. Default is enable. To manually start or stop the TELNET server use the startserver and stopserver commands.
TELNET_PORT	system.ini	Defines the TCP port number for the TELNET server. Default value is 23.
TELNET_PRIO	system.ini	Defines the priority of the TELNET server. Default value is 104. The priority of each TELNET session will be the TELNET server's priority + 1.
TERM	ish.ini, \$HOME/.ish.ini	Specifies terminal emulation. Could be set to vt100 or DUMB.
TIMEZONE	system.ini	The time zone. See DAYLIGHT for a description.
UID	N/A	The user's id according to passwd.ini. This variable will reflect the current user.
USER	N/A	The user's name, taken from the login.

## 22. Appendix 7: Clock Setup

This section describes how to set up the system time and date including time zones and daylight saving.

### Time Zones, Daylight Savings, and Setting the Clock

#### Introduction

To make the clock work correctly these things must be set up:

#### Time zone

#### Daylight saving

#### Set time (manually or via NTP)

The following sections describe how this is done.

#### Time Zones

The time zone is configured either by editing the system.ini file or by executing the date -z command (which will alter the system.ini file for you). The whole time zone concept is based on the environment variable TIMEZONE defined in the system.ini file. This variable can either have a valid three-letter abbreviation from the list below or a numeric value specifying the offset from GMT in hours. Note that half-hour offsets must be written as "3.5", not "3:30".

The TIMEZONE variable can also be entered as a string like "PST-8PDT". In that case, the number in the middle specifies the offset, the string before it the abbreviation for the standard time zone, and the string after it the abbreviation for the time zone during daylight saving time. Neither abbreviations nor offset need to be in the list, abbreviations can be longer than three letters.

#### Valid Abbreviations

The following table lists all available time zones on the system. It can also be printed with the date -p command. When there are more than one abbreviation for a time zone, the one listed first will be used by the date command to display the time.

Offset	Abbreviation	Description
-11	MIT	Samoa
-10	HST	Hawaii
-9	AST	Alaska
-8	PST	Pacific Standard Time
-7	MST,PNT	Mountain Standard Time
-6	CST	Central Standard Time
-5	EST,IET	Eastern Standard Time
-4	PRT	Atlantic Standard Time
-3.5	CNT	Newfoundland
-3	AGT,BET	Eastern South America
0	UTC,GMT,WET	Coordinated Universal Time
+1	ECT	European Central Time
+2	EET,CAT,ART	European Eastern Time
+3	EAT	Saudi Arabia
+3.5	MET	Iran
+4	NET	
+5	PLT	West Asia
+5.5	IST	India
+6	BST	Central Asia



Offset	Abbreviation	Description
+7	VST	Bangkok
+8	CTT	China
+9	JST	Japan
+9.5	ACT	Central Australia
+10	AET	Eastern Australia
+11	SST	Central Pacific
+12	NST	New Zealand

*Examples*

Any of the following lines in system.ini will set the time zone for central Europe.

```
setenv TIMEZONE ECT
```

```
setenv TIMEZONE 1
```

```
setenv TIMEZONE CET1CST
```

This can also be accomplished by executing one of the following.

```
date -z ECT
```

```
date -z 1
```

```
date -z CET1CST
```

*Daylight Saving*

Daylight saving is difficult to handle in a generic way, since it is handled differently between countries (and even within countries), continents, southern and northern hemisphere and in some countries it even differs from year to year.

The environment variable DAYLIGHT in the system.ini file contains rules for when the daylight saving starts and stops. It can be altered by editing the system.ini file or by executing the date -d command. The format of the DAYLIGHT variable might seem quite complex, but it allows specifying the correct daylight saving rules for almost all locations in the world.

The rule of the DAYLIGHT variable looks like: MMODHHmm-MMODHHmm. The section before the '-' is the daylight saving start condition and the section after the '-' is the end condition.

The fields are defined in the following table:

Field	Meaning	Possible values
MM	Month	01 - January 02 - February etc.
O	Occurrence	0 - Last occurrence in month 1 - First occurrence 2 - Second occurrence etc.
D	Day of week	0 - Sunday 1 - Monday etc.
HH	Hour	00-23
mm	Minute	00-59

The times should be specified in local time.

*Changing to and from Daylight Saving*

The transitions to and from daylight saving time are done automatically, when the system time reaches the transition times specified by the DAYLIGHT variable. If the time is set (by the date -s command), the daylight saving status is immediately recalculated. If the

daylight saving status cannot be determined (when the time is set within the hour that exists twice because daylight saving time ends), it is considered to be in effect.

The date -d command without additional parameters or options will have no effect, to make it backward compatible with earlier implementations which used this command to evaluate the daylight saving status.

### *Disabling daylight saving*

If daylight saving time is not used at all in the present location, it can be disabled by removing the DAYLIGHT variable or setting its value to an empty string. This can also be accomplished using the date -d "" command.

### *Examples*

The following line will set up the daylight saving for European standard, in the ECT time zone.

```
setenv DAYLIGHT 03000200-10000300
```

This can also be accomplished by executing the following command.

```
date -d 03000200-10000300
```

Which will be interpreted as: daylight saving starts at the last Sunday of March 02:00 local time, and ends the last Sunday of October 03:00 local summer time. Note that both transitions are at 1:00 UTC.

### *Setting the Clock*

When the two variables above has been set up correctly, it is possible to set the clock. This can be done in two ways:

Set the clock via the command line (date -s command)

Acquire the time via NTP (date -n command)

### *Setting Time via Command Line*

This is done by executing the date -s YYYYMMDDhhmmss command, where the YYYYMMDDhhmmss string has the following meaning:

Field	Meaning
YYYY	The year including century (such as 2003)
MM	The month with two digits (such as 03 for march)
DD	The day in the month with two digits
hh	The hour on 24 H format (two digits)
mm	The minute (two digits)
ss	The second (two digits)

### *Setting Time via NTP*

If you have access to a NTP server, the system can acquire the time via the network. Try:

```
date -n host.name.com
```

The host.name.com can either be a hostname or an IP number in dot notation.

The following is a list of some NTP servers from where the time can be acquired:

Norway - [fartein.ifi.uio.no](http://fartein.ifi.uio.no)

Scotland - [ntp0.strath.ac.uk](http://ntp0.strath.ac.uk)

Switzerland - [swisstime.ethz.ch](http://swisstime.ethz.ch)

USA - [ntp.ucsd.edu](http://ntp.ucsd.edu)

Japan - [clock.nc.fukuoka-u.ac.jp](http://clock.nc.fukuoka-u.ac.jp)

### *Information Resources*

The following links may give a hint on what it's all about:

About <https://www.timeanddate.com/time/dst/>

More information about [http://www.sunrisesunset.com/dst\\_info.html](http://www.sunrisesunset.com/dst_info.html)

In [https://en.wikipedia.org/wiki/Daylight\\_saving\\_time](https://en.wikipedia.org/wiki/Daylight_saving_time)

## 23. Appendix 8: Firmware Upgrade

Users can update and upgrade modules to the latest firmware version. A detailed description of how to upgrade the firmware on the target module is available on the website <https://www.silabs.com/>.

## 24. Appendix 9: JSON Interface Definition

The software definition for the JSON interface follows:

```
*****
JSON Interface Specification
*****

Version
-----
0.3

Definitions
-----

boolean: one_of[ true, false ]
u8: decimal number in range 0..255 (8-bit unsigned)
i8: decimal number in range -128..127 (8-bit signed)
u16: decimal number in range 0..65535 (16-bit unsigned)
i16: decimal number in range -32768..32767 (16-bit signed)
u32: decimal number in range 0..4294967295 (32-bit unsigned)
i32: decimal number in range -2147483648..2147483647 (32-bit signed)
x8: string in "0x%02x" format representing 8-bit unsigned integer (ex. "0x0a")
x16: string in "0x%04x" format representing 16-bit unsigned integer (ex. "0x000a")
x32: string in "0x%08x" format representing 32-bit signed integer (ex. "0x0000000a")
octet_string[N] string of N octets in "%02X" format separated by 'space' (ex. "FC AF 6A")

clock_id: octet_string[8] (ex. "FC AF 6A FF FE 02 E0 CE")
port_id:
{
    "clockIdentity": clock_id,
    "portNumber": u16
}
profile_id: string in "%02X-%02X-%02X-%02X-%02X-%02X" format representing PTP profile ID (ex.
"00-1B-19-00-01-00")

sec: number in range -18446744073709551615..18446744073709551615 (sign-extended 64-bit unsigned integer)
nsec: number in range 000000000..999999999
time: {"sec": sec, "nsec": nsec} (ex. {"sec": -0, "nsec": 1}, {"sec": 10, "nsec": 999999})

phys_addr: string in "%02X:%02X:%02X:%02X:%02X:%02X" format representing MAC address (ex. "FC:AF:6A:01:E0:CF")
ipv4_addr: string in "%u.%u.%u.%u" format representing IPv4 address (ex. "192.168.1.10")
ipv6_addr: string in "%x:%x:%x:%x:%x:%x:%x:%x" format representing IPv6 address (ex.
"fe80:0:0:0:feaf:6aff:fe01:e0ce")
8023_addr: string in "%02X-%02X-%02X-%02X-%02X-%02X" format representing IEEE 802.3 address (ex. "FC-AF-6A-01-
E0-CF")

protocol_address: one_of[
{
    "networkProtocol": "UDP/IPv4",
    "networkAddress": ipv4_addr
},
{
    "networkProtocol": "UDP/IPv6",
    "networkAddress": ipv6_addr
},
{
    "networkProtocol": "IEEE 802.3",
    "networkAddress": 8023_addr
}
]

clock_quality:
{
    "clockClass" : u8,
    "clockAccuracy" : x8,
    "offsetScaledLogVariance": x16
}

unicastNode:
{
```

```

    "nodeType": one_of[ "MASTER", "SLAVE" ]
    "isStatic": boolean,
    "clockIdentity": clock_id,
    "protocolAddress": prtocol_address,
    "localPriority": u8,
    "alternateDomain": boolean,
    "domainNumber": u8,
    "messageNegotiation": boolean,
    "announceGranted": boolean,
    "announceInterval": i8,
    "announceDuration": i32,
    "syncGranted": boolean,
    "syncInterval": i8,
    "syncDuration": i32,
    "delayRespGranted": boolean,
    "delayRespInterval": i8,
    "delayRespDuration": i32
}

quality_level: one_of[ "QL-PRC", "QL-SSU-A", "QL-SSU-B", "QL-SEC", "QL_DNU",
                      "QL-PRS", "QL-STU", "QL-ST2", "QL-TNC", "QL-ST3E", "QL-ST3", "QL-SMC", "QL-DUS" ]

```

Commands

-----

## 24.1 Clock Commands

Common Errors

```

{
    "Error": format_string("Command '%s' not found", command_name)
}

```

### 24.1.1 "clock info"

#### Format

```
{
  "clockInformation":
  {
    "clockType": one_of[ "ORDINARY CLOCK", "BOUNDARY CLOCK" ],
    "physicalLayerProtocol": "IEEE 802.3",
    "numberPorts": u16,
    "portTable": array_of[
      {
        "physicalAddress": phys_addr,
        "protocolAddress": protocol_addr,
        "profileIdentity": profile_id
      }
    ],
    "manufacturerIdentity": octet_string[3],
    "productDescription": "manufacturerName;modelName;instanceIdentifier",
    "revisionData": "hardwareRevision;firmwareRevision;softwareRevision",
    "userDescription": "userDeviceName;userDeviceLocation"
  }
}
```

#### Example

```
"clockInformation":
{
  "clockType": "BOUNDARY CLOCK",
  "physicalLayerProtocol": "IEEE 802.3",
  "numberPorts": 2,
  "portTable":
  [
    {
      "physicalAddress": "FC:AF:6A:02:E0:CD",
      "protocolAddress":
      {
        "networkProtocol": "UDP/IPv4",
        "networkAddress": "192.168.1.10"
      },
      "profileIdentity": "00-1B-19-00-01-00"
    },
    {
      "physicalAddress": "FC:AF:6A:01:E0:CE",
      "protocolAddress":
      {
        "networkProtocol": "UDP/IPv6",
        "networkAddress": "fe80:0:0:0:feaf:6aff:fe01:e0ce"
      },
      "profileIdentity": "00-1B-19-00-01-00"
    }
  ],
  "manufacturerIdentity": "FC AF 6A",
  "productDescription": "Qulsar Inc;C34-M5x/M6x/M8x;02-E0-CD",
  "revisionData": ";;1.0.18",
  "userDescription": ""
}
```

### 24.1.2 "clock type"

#### Format

```
{
  "clockType": one_of[ "ORDINARY CLOCK", "BOUNDARY CLOCK" ]
}
```

#### Errors

```
{
  "Error": "Clock type must be 'oc' or 'bc'"
}
```

### 24.1.3 "clock twostep"

```

Format
{
  "twoStep": boolean
}

Errors
{
  "Error": "Two-step mode must be 'on' or 'off'"
}

```

### 24.1.4 "clock slaveonly"

```

Format
{
  "slaveOnly": boolean
}

Errors
{
  "Error": "Slave-only mode must be 'on' or 'off'"
}

```

### 24.1.5 "clock domain"

```

Format
{
  "domainNumber": u8
}

Errors
{
  "Error": "Domain number must be 0..255"
}

```

### 24.1.6 "clock priority"

```

Format
{
  one_of[ "priority1": u8, "priority2": u8 ]
}

Errors
{
  "Error": "Parameter is missing"
}
{
  "Error": "Parameter must be '1' or '2'"
}
{
  "Error": "Priority must be 0..255"
}

```



### 24.1.7 "clock quality"

```

Format
{
    "clockQuality": clock_quality
}

Errors
{
    "Error": "Parameter is missing"
}
{
    "Error": "Clock class must be 0..254"
}
{
    "Error": "Clock accuracy must be 0x20..0x31 or 0xfe"
}
{
    "Error": "Clock variance must be 0x0000..0xffff"
}

```

### 24.1.8 "clock profile"

```

Format
{
    "clockProfile": one_of[ "UNDEFINED",
                           "DEFAULT",
                           "POWER V1",
                           "POWER V2",
                           "G8265",
                           "G8275.1",
                           "G8275.2" ]
}

Errors
{
    "Error": "Profile must be 'default', 'g8265', 'g8275.1', 'g8275.2', 'power'"
}
{
    "Error": "Parameter must be 'v1' or 'v2'"
}

```

## 24.2 Port Commands

```

Common Errors
{
    "Error": format_string("Command '%s' not found", command_name)
}
{
    "Error": format_string("Port %d not found", port_number)
}

```

### 24.2.1 "port state"

```

Format
{
    "portState": one_of[ "PTP2_STATE_NOT_DEFINED",
                        "PTP2_STATE_INITIALIZING",
                        "PTP2_STATE_FAULTY",
                        "PTP2_STATE_DISABLED",
                        "PTP2_STATE_LISTENING",
                        "PTP2_STATE_PRE_MASTER",
                        "PTP2_STATE_MASTER",
                        "PTP2_STATE_PASSIVE",
                        "PTP2_STATE_UNCALIBRATED",
                        "PTP2_STATE_SLAVE" ]
}

```

### 24.2.2 "port announce"

```

Format
{
  "logAnnounceInterval": i8
}

Errors
{
  "Error": "Log announce interval must be -4..5"
}

```

### 24.2.3 "port receipt"

```

Format
{
  "announceReceiptTimeout": u8
}

Errors
{
  "Error": "Announce receipt timeout must be 2..10"
}

```

### 24.2.4 "port sync"

```

Format
{
  "logSyncInterval": i8
}

Errors
{
  "Error": "Log sync interval must be -8..2"
}

```

### 24.2.5 "port delay"

```

Format
{
  "logMinDelayReqInterval": i8
}

Errors
{
  "Error": "Log min delay request interval must be -7..7"
}

```

### 24.2.6 "port pdelay"

```

Format
{
  "logMinPdelayReqInterval": i8
}

Errors
{
  "Error": "Log min pdelay request interval must be -7..7"
}

```

### 24.2.7 "port mechanism"

```

Format
{
    "delayMechanism": one_of[ "E2E", "P2P", "DISABLED" ]
}

Errors
{
    "Error": "Delay mechanism must be 'e2e', 'p2p', 'disable'"
}

```

### 24.2.8 "port protocol"

```

Format
{
    "networkProtocol": one_of[ "UDP/IPv4", "UDP/IPv6", "IEEE 802.3" ]
}

Errors
{
    "Error": "Network protocol must be 'udp', 'udp6', 'eth'"
}

```

### 24.2.9 "port compatibility"

```

Format
{
    "compatibilityMode": one_of[ "ON", "OFF", format_string("AUTO/%d sec", timeout) ]
}

Errors
{
    "Error": "Compatibility mode must be 'on', 'off', 'auto'"
}
{
    "Error": "Parameter is missing"
}
{
    "Error": "Compatibility timeout must be 0..600 s"
}

```

### 24.2.10 "port asymmetry"

```

Format
{
    "delayAsymmetry": time
}

```

### 24.2.11 "port unicast"

```

Format
{
    "unicastState": one_of[ "PTP2_UNICAST_STATE_DISABLED",
                           "PTP2_UNICAST_STATE_SLAVE",
                           "PTP2_UNICAST_STATE_MASTER",
                           "PTP2_UNICAST_STATE_MASTER_SLAVE" ]
}

Errors
{
    "Error": "Parameter must be: 'master', 'slave', 'both', 'disable'"
}

```

### 24.2.12 "port profile"

```

Format
{
    "portProfile": one_of[ "UNDEFINED",
                           "DEFAULT",
                           "POWER V1",
                           "POWER V2",
                           "G8265",
                           "G8275.1",
                           "G8275.2" ]
}

Errors
{
    "Error": "Profile must be 'default', 'g8265', 'g8275.1', 'g8275.2', 'power'"
}
{
    "Error": "Parameter must be 'v1' or 'v2'"
}

```

## 24.3 Unicast Commands

```

Common Errors
{
    "Error": format_string("Command '%s' not found", command_name)
}
{
    "Error": format_string("Port %d not found", port_number)
}

```

### 24.3.1 "unicast status"

```

Format
{
    "portUnicastInformation":
    {
        "unicastState": one_of[ "PTP2_UNICAST_STATE_DISABLED",
                                "PTP2_UNICAST_STATE_SLAVE",
                                "PTP2_UNICAST_STATE_MASTER",
                                "PTP2_UNICAST_STATE_MASTER_SLAVE" ],
        "negotiationEnabled": boolean,
        "cancelTimeout": u32,
        "transmissionDuration": u32,
        "queryInterval": u32,
        "messageRateLimit": u32,
        "currentMessageRate": u32
    }
}

```

### 24.3.2 "unicast negotiation"

```

Format
{
    "negotiationEnabled": boolean
}

Errors
{
    "Error": "Parameter must be 'on' or 'off'"
}

```

### 24.3.3 "unicast query"

```

Format
{
  "queryInterval": u32
}

Errors
{
  "Error": "Query interval must be 1..600 s"
}

```

### 24.3.4 "unicast duration"

```

Format
{
  "transmissionDuration": u32
}

Errors
{
  "Error": "Transmission duration must be 60..3600 s"
}

```

### 24.3.5 "unicast timeout"

```

Format
{
  "cancelTimeout": u32
}

Errors
{
  "Error": "Cancel timeout must be 1..10 s"
}

```

### 24.3.6 "unicast limit"

```

Format
{
  "messageRateLimit": u32
}

Errors
{
  "Error": "Message rate limit must be 0..1000 msg/s"
}

```

## 24.3.7 "unicast filter"

```

filterEntry: one_of[
  {
    "networkAddress": one_of[ "any", ipv4_addr, ipv6_addr, 8023_addr ],
    "clockIdentity": one_of[ "any", clock_id ],
    "action": "ALLOW",
    "announceLimit": one_of[ "deny", i8 ],
    "syncLimit": one_of[ "deny", i8 ],
    "delayReqLimit": one_of[ "deny", i8 ],
    "localPriority": u8
  },
  {
    "networkAddress": one_of[ "any", ipv4_addr, ipv6_addr, 8023_addr ],
    "clockIdentity": one_of[ "any", clock_id ],
    "action": "DENY"
  }
]

```

## Format

```

{
  "unicastSlaveFilter": array_of[ filterEntry ]
}
{
  "unicastMasterFilter": array_of[ filterEntry ]
}
{
  "filterEntry": filterEntry
}
{
  "Success": "Filter entry is deleted"
}

```

## Errors

```

{
  "Error": "Filter type is missing"
}
{
  "Error": "Address is missing"
}
{
  "Error": "Address format is incorrect"
}
{
  "Error": "Address format does not match network protocol"
}
{
  "Error": "Clock identity is too short"
}
{
  "Error": "Filter entry already exists"
}
{
  "Error": "Filter entry not found"
}
{
  "Error": "Priority must be 0..255"
}
{
  "Error": "Announce interval limit must be -4..4"
}
{
  "Error": "Sync interval limit must be -7..4"
}
{
  "Error": "Delay interval limit must be -7..4 or 'x'"
}
{
  "Error": "Unable to add filter entry"
}

```

## Example

```
{
  "unicastSlaveFilter":
  [
    {
      "networkAddress": "any",
      "clockIdentity": "any",
      "action": "ALLOW",
      "announceLimit": -4,
      "syncLimit": -7,
      "delayReqLimit": -7,
      "localPriority": 255
    }
    {
      "networkAddress": "192.168.1.120",
      "clockIdentity": "any",
      "action": "ALLOW",
      "announceLimit": 0,
      "syncLimit": -1,
      "delayReqLimit": "deny",
      "localPriority": 255
    }
    {
      "networkAddress": "any",
      "clockIdentity": "FC AF 6A FF FE 02 E0 CE",
      "action": "DENY"
    }
  ]
}
```

### 24.3.8 "unicast node"

#### Format

```
{
  "unicastNodeTable": array_of[ unicastNode("isStatic" == true) ]
}
{
  "unicastSlaveTable": array_of[ unicastNode("isStatic" == true && "nodeType" == "SLAVE") ]
}
{
  "unicastMasterTable": array_of[ unicastNode("isStatic" == true && "nodeType" == "SLAVE") ]
}
{
  "unicastNode": unicastNode
}
{
  "Success": "Unicast node is deleted"
}
```

#### Errors

```
{
  "Error": "Unicast operations are disabled"
}
{
  "Error": "Node type is missing"
}
{
  "Error": "Only slave nodes are allowed on unicast master port"
}
{
  "Error": "Only master nodes are allowed on unicast slave port"
}
{
  "Error": "Address is missing"
}
{
  "Error": "Address format does not match network protocol"
}
{
  "Error": "Clock identity is too short"
}
{
  "Error": "Unicast node already exists"
}
{
  "Error": "Unicast node not found"
}
{
  "Error": "Cannot modify or delete dynamic node"
}
{
  "Error": "Priority must be 0..255"
}
{
  "Error": "Announce interval must be -4..4 or 'x'"
}
{
  "Error": "Sync interval must be -7..4 or 'x'"
}
{
  "Error": "Delay Req interval must be -7..4 or 'x'"
}
{
  "Error": "Domain number must be 0..255"
}
{
  "Error": "Unable to add unicast node"
}
```

#### Example

```
{
  "unicastNode":
```



```
{
  "nodeType": "MASTER",
  "isStatic": true,
  "clockIdentity": "FC AF 6A FF FE 02 E0 CE",
  "protocolAddress":
  {
    "networkProtocol": "UDP/IPv4",
    "networkAddress": "192.168.1.1"
  }
  "localPriority": 128,
  "alternateDomain": true,
  "domainNumber": 4,
  "messageNegotiation": true,
  "announceGranted": true,
  "announceInterval": 0,
  "announceDuration": 600,
  "syncGranted": true,
  "syncInterval": 0,
  "syncDuration": 600,
  "delayRespGranted": true,
  "delayRespInterval": 0,
  "delayRespDuration": 600
}
```

## 24.4 Power Profile Commands

### Common Errors

```
{
  "Error": format_string("Command '%s' not found", command_name)
}
```

### 24.4.1 "power gmid"

#### Format

```
{
  "grandmasterID": u16
}
```

### 24.4.2 "power inacc"

#### Format

```
{
  "timeInaccuracy": { "grandmaster": u32, "network": u32 }
}
```

## 24.5 G8265 Profile Commands

### Common Errors

```
{
  "Error": format_string("Command '%s' not found", command_name)
}
```

## 24.6 G8275 Profile Commands

### Common errors

```
{
  "Error": format_string("Command '%s' not found", command_name)
}
```

### 24.6.1 "g8275 priority"

```

Format
{
    "clockLocalPriority": u8
}

Errors
{
    "Error": "Local priority must be 1..255"
}

```

### 24.6.2 "g8275 steps"

```

Format
{
    "maxStepsRemoved": u8
}

Errors
{
    "Error": "Max steps must be 1..255"
}

```

### 24.6.3 "g8275 [port\_number] priority"

```

Format
{
    "localPriority": u8
}

Errors
{
    "Error": format_string("Port %d not found", port_number)
}
{
    "Error": "Local priority must be 1..255"
}

```

### 24.6.4 "g8275 [port\_number] masteronly"

```

Format
{
    "masterOnly": boolean
}

Errors
{
    "Error": format_string("Port %d not found", port_number)
}
{
    "Error": "Master-only must be 'on' or 'off'"
}

```

## 24.7 Dataset Commands

```

Common Errors
{
    "Error": format_string("Command '%s' not found", command_name)
}

```

### 24.7.1 "dataset default"

#### Format

```
{
  "defaultDS":
  {
    "clockIdentity": clock_id,
    "numberPorts": u16,
    "twoStep": boolean
    "domainNumber": u8,
    "clockQuality": clock_quality
    "priority1": u8,
    "priority2": u8,
    "slaveOnly": boolean
  }
}
```

#### Example

```
{
  "defaultDS":
  {
    "clockIdentity": "FC AF 6A FF FE 02 E0 CE",
    "numberPorts": 2,
    "twoStep": false,
    "domainNumber": 0,
    "clockQuality":
    {
      "clockClass" : 248,
      "clockAccuracy" : "0xfe",
      "offsetScaledLogVariance": "0xffff"
    },
    "priority1": 128,
    "priority2": 128,
    "slaveOnly": false
  }
}
```

### 24.7.2 "dataset current"

#### Format

```
{
  "currentDS":
  {
    "stepsRemoved": u8,
    "offsetFromMaster": time,
    "meanPathDelay": time
  }
}
```

#### Example

```
{
  "currentDS":
  {
    "stepsRemoved": 255,
    "offsetFromMaster": {"sec": -0, "nsec": 1025},
    "meanPathDelay": {"sec": 0, "nsec": 624}
  }
}
```

**24.7.3 "dataset parent"****Format**

```
{
  "parentDS":
  {
    "parentPortIdentity": port_id
    "parentStats": boolean,
    "observedParentOffsetScaledLogVariance": x16,
    "observedParentClockPhaseChangeRate": x32,
    "grandmasterClockIdentity": clock_id,
    "grandmasterClockQuality": clock_quality
    "grandmasterPriority1": u8,
    "grandmasterPriority2": u8
  }
}
```

**Example**

```
{
  "parentDS":
  {
    "parentPortIdentity":
    {
      "clockIdentity": "FC AF 6A FF FE 02 E0 CE",
      "portNumber": 1
    },
    "parentStats": false,
    "observedParentOffsetScaledLogVariance": "0xffff",
    "observedParentClockPhaseChangeRate": "7fffffff",
    "grandmasterClockIdentity": "FC AF 6A FF FE 02 E0 CE",
    "grandmasterClockQuality":
    {
      "clockClass" : 248,
      "clockAccuracy" : "0xfe",
      "offsetScaledLogVariance": "0xffff"
    },
    "grandmasterPriority1": 128,
    "grandmasterPriority2": 128
  }
}
```

**24.7.4 "dataset time"****Format**

```
{
  "timePropertiesDS":
  {
    "ptpTimescale" : boolean,
    "currentUtcOffset": i16,
    "currentUtcOffsetValid": boolean,
    "leap59": boolean,
    "leap61": boolean,
    "timeTraceable": boolean,
    "frequencyTraceable": boolean,
    "timeSource": one_of[ "ATOMIC CLOCK",
                          "GPS",
                          "TERRESTRIAL RADIO",
                          "PTP",
                          "NTP",
                          "HAND SET",
                          "UNDEFINED",
                          "OTHER",
                          "INTERNAL OSCILLATOR" ]
  }
}
```

**Example**

```
{
  "timePropertiesDS":
  {
    "ptpTimescale" : true,
    "currentUtcOffset": 37,
    "currentUtcOffsetValid": true,
    "leap59": false,
    "leap61": false,
    "timeTraceable": false,
    "frequencyTraceable": false,
    "timeSource": "INTERNAL OSCILLATOR"
  }
}
```

**24.7.5 "dataset port [port\_number]"****Format**

```
{
  "portDS":
  {
    "versionNumber": 2,
    "portIdentity": port_id,
    "portState": one_of[ "PTP2_STATE_NOT_DEFINED",
                        "PTP2_STATE_INITIALIZING",
                        "PTP2_STATE_FAULTY",
                        "PTP2_STATE_DISABLED",
                        "PTP2_STATE_LISTENING",
                        "PTP2_STATE_PRE_MASTER",
                        "PTP2_STATE_MASTER",
                        "PTP2_STATE_PASSIVE",
                        "PTP2_STATE_UNCALIBRATED",
                        "PTP2_STATE_SLAVE" ],
    "logAnnounceInterval": i8,
    "announceReceiptTimeout": u8,
    "logSyncInterval": i8,
    "logMinDelayReqInterval": i8,
    "logMinPdelayReqInterval": i8,
    "delayMechanism": one_of[ "E2E", "P2P", "DISABLED" ],
    "peerMeanPathDelay": time
  }
}
```

**Errors**

```
{
  "Error": format_string("Port %d not found", port_number)
}
```

**Example**

```
{
  "portDS":
  {
    "versionNumber": 2,
    "portIdentity":
    {
      "clockIdentity": "FC AF 6A FF FE 02 E0 CE",
      "portNumber": 1
    },
    "portState": "PTP2_STATE_SLAVE",
    "logAnnounceInterval": 1,
    "announceReceiptTimeout": 3,
    "logSyncInterval": 0,
    "logMinDelayReqInterval": 0,
    "logMinPdelayReqInterval": 0,
    "delayMechanism": "E2E",
    "peerMeanPathDelay": {"sec": 0, "nsec": 624}
  }
}
```

**24.7.6 "dataset foreign [port\_number]"****Format**

```
{
  "foreignMasterTable":
  array_of[
    {
      "portIdentity": port_id,
      "announceMessages": u8,
      "recentAnnounce":
      {
        "sequenceId": u16,
        "grandmasterClockIdentity": clock_id,
        "grandmasterClockQuality": clock_quality,
        "grandmasterPriority1": u8,
        "grandmasterPriority2": u8,
        "stepsRemoved": u8
      }
      "unicastNodeAddress": one_of[ null, protocol_address ]
    }
  ]
}
```

**Errors**

```
{
  "Error": format_string("Port %d not found", port_number)
}
```

**Example**

```
{
  "foreignMasterTable":
  [
    {
      "portIdentity":
      {
        "clockIdentity": "FC AF 6A FF FE 02 E0 CE",
        "portNumber": 1
      },
      "announceMessages": 2,
      "recentAnnounce":
      {
        "sequenceId": 195,
        "grandmasterClockIdentity": "FC AF 6A FF FE 02 E0 CE",
        "grandmasterClockQuality":
        {
          "clockClass" : 248,
          "clockAccuracy" : "0xfe",
          "offsetScaledLogVariance": "0xffff"
        },
        "grandmasterPriority1": 128,
        "grandmasterPriority2": 128,
        "stepsRemoved": 255
      }
      "unicastNodeAddress": null
    },
    {
      "portIdentity":
      {
        "clockIdentity": "FC AF 6A FF FE 02 E0 CF",
        "portNumber": 2
      },
      "announceMessages": 1,
      "recentAnnounce":
      {
        "sequenceId": 20056,
        "grandmasterClockIdentity": "FC AF 6A FF FE 02 E0 CF",
        "grandmasterClockQuality":
        {
          "clockClass" : 248,
          "clockAccuracy" : "0xfe",
          "offsetScaledLogVariance": "0xffff"
        }
      }
    }
  ]
}
```

```
        "grandmasterPriority1": 128,  
        "grandmasterPriority2": 128,  
        "stepsRemoved": 255  
    }  
    "unicastNodeAddress":  
    {  
        "networkProtocol": "UDP/IPv4",  
        "networkAddress": "192.168.1.1"  
    }  
}  
]  
}
```



**24.7.7 "dataset unicast [port\_number]"****Format**

```
{
  "unicastNodeList": array_of[ unicastNode ]
}
```

**Errors**

```
{
  "Error": format_string("Port %d not found", port_number)
}
```

**Example**

```
{
  "unicastNodeList":
  [
    {
      "nodeType": "MASTER",
      "isStatic": true,
      "clockIdentity": "FC AF 6A FF FE 02 E0 CE",
      "protocolAddress":
      {
        "networkProtocol": "UDP/IPv4",
        "networkAddress": "192.168.1.1"
      }
      "localPriority": 128,
      "alternateDomain": true,
      "domainNumber": 4,
      "messageNegotiation": true,
      "announceGranted": true,
      "announceInterval": 0,
      "announceDuration": 600,
      "syncGranted": true,
      "syncInterval": 0,
      "syncDuration": 600,
      "delayRespGranted": true,
      "delayRespInterval": 0,
      "delayRespDuration": 600
    },
    {
      "nodeType": "SLAVE",
      "isStatic": false,
      "clockIdentity": "FC AF 6A FF FE 02 E0 CF",
      "protocolAddress":
      {
        "networkProtocol": "UDP/IPv4",
        "networkAddress": "192.168.1.2"
      }
      "localPriority": 255,
      "alternateDomain": false,
      "domainNumber": 0,
      "messageNegotiation": true,
      "announceGranted": true,
      "announceInterval": 0,
      "announceDuration": 900,
      "syncGranted": true,
      "syncInterval": -1,
      "syncDuration": 900,
      "delayRespGranted": true,
      "delayRespInterval": -1,
      "delayRespDuration": 900
    }
  ]
}
```

### 24.7.8 "dataset power"

#### Format

```
{
  "powerProfileDS":
  {
    "grandmasterID": u16,
    "timeInaccuracy": { "grandmaster": u32, "network": u32 },
    "alternateTimeOffsetIndicator":
    {
      "displayName": string,
      "currentOffset": i32,
      "nextJump": one_of[ null,
      {
        "jumpSeconds": i32,
        "timeOfNextJump": sec
        "displayTimeOfNextJump": "YYYY-MM-DD HH:MM:SS TAI"
      }
    ]
  }
}
```

#### Example

```
{
  "powerProfileDS":
  {
    "grandmasterID": 1,
    "timeInaccuracy": { "grandmaster": 50, "network": 1500 },
    "alternateTimeOffsetIndicator":
    {
      "displayName": "CET",
      "currentOffset": 3562,
      "nextJump":
      {
        "jumpSeconds": 3600,
        "timeOfNextJump": 1585447237
        "displayTimeOfNextJump": "2020-03-29 02:00:37 TAI"
      }
    }
  }
}
```

### 24.7.9 Dataset g8275 Commands

#### 24.7.9.1 "dataset g8275"

#### Format

```
{
  "g8275ProfileDS":
  {
    "clockLocalPriority": u8,
    "maxStepsRemoved": u8
  }
}
```

#### Example

```
{
  "g8275ProfileDS":
  {
    "clockLocalPriority": 128,
    "maxStepsRemoved": 255
  }
}
```

**24.7.9.2 "dataset g8275 [port\_number]"****Format**

```
{
  "g8275ProfilePortDS":
  {
    "portNumber": u16,
    "masterOnly": boolean,
    "localPriority": u8,
    "signalFail": boolean
  }
}
```

**Errors**

```
{
  "Error": format_string("Port %d not found", port_number)
}
```

**Example**

```
{
  "g8275ProfilePortDS":
  {
    "portNumber": 1,
    "masterOnly": true,
    "localPriority": 128,
    "signalFail": false
  }
}
```

**24.8 Time Commands****Common Errors**

```
{
  "Error": format_string("Command '%s' not found", command_name)
}
```

### 24.8.1 "time info"

#### Format

```
{
  "timeInformation":
  {
    "clockSyncState": one_of[ "FREE",
                              "SYNCHRONIZING",
                              "SYNCHRONIZING",
                              "HOLDOVER" ],
    "clockSynchronized": boolean,
    "timeSource": one_of[ "ATOMIC CLOCK",
                          "GPS",
                          "TERRESTRIAL RADIO",
                          "PTP",
                          "NTP",
                          "HAND SET",
                          "UNDEFINED",
                          "OTHER",
                          "INTERNAL OSCILLATOR" ],
    "timeScale": one_of[ "PTP", "ARB" ],
    "currentTime": time,
    "currentUtcOffset": i16,
    "leap59": boolean,
    "leap61": boolean,
    "displayTimeARB": "[...]SSSSSSSSSS.NS",
    "displayTimePTP": "YYYY-MM-DD HH:MM:SS.NS",
    "displayTimeUTC": "YYYY-MM-DD HH:MM:SS",
    "displayTimeLocal": "YYYY-MM-DD HH:MM:SS TZ",
    "autoUpdateRTC": boolean,
    "holdoverInterval": u32
  }
}
```

#### Example

```
{
  "timeInformation":
  {
    "clockSyncState": "SYNCHRONIZING",
    "clockSynchronized": true,
    "timeSource": "GPS",
    "timeScale": "PTP",
    "currentTime": { "sec": 1581329920, "nsec": 204466328 },
    "currentUtcOffset": 37,
    "leap59": false,
    "leap61": false,
    "displayTimeARB": "1581329920.204466328",
    "displayTimePTP": "2020-02-10 10:18:40.204466328",
    "displayTimeUTC": "2020-02-10 10:18:03",
    "displayTimeLocal": "2020-02-10 10:18:03 GMT",
    "autoUpdateRTC": true,
    "holdoverInterval": 1000
  }
}
```

### 24.8.2 "time sync"

#### Format

```
{
  "syncStatus":
  {
    "clockSynchronized": boolean,
    "clockSyncState": one_of[ "FREE",
                              "SYNCHRONIZING",
                              "SYNCHRONIZING",
                              "HOLDOVER" ]
  }
}
```

### 24.8.3 "time arb|ptp|utc|ntp"

```

Format
{
    one_of[
        "displayTimeARB": "[-]SSSSSSSSSS.NS",
        "displayTimePTP": "YYYY-MM-DD HH:MM:SS.NS",
        "displayTimeUTC": "YYYY-MM-DD HH:MM:SS"
    ]
}

Errors
{
    "Error": format_string("Time can not be set. Clock is synchronizing with %s time source", one_of[ "GPS",
"PTP" ])
}
{
    "Error": "Nanoseconds must be 0..999999999"
}
{
    "Error": "Date is missing"
}
{
    "Error": "Time is missing"
}
{
    "Error": "Year is not correct"
}
{
    "Error": "Month is not correct"
}
{
    "Error": "Day is not correct"
}
{
    "Error": "Hours must be 0..23"
}
{
    "Error": "Minutes must be 0..59"
}
{
    "Error": "Seconds must be 0..59"
}
{
    "Error": "Unable to resolve NTP server name"
}
{
    "Error": "Unable to obtain time from NTP server"
}
{
    "Error": "NTP server is not synchronized"
}

```

### 24.8.4 "time offset"

```

Format
{
    "currentUtcOffset": i16
}

Errors
{
    "Error": "Offset must be -32768..32767 s"
}
{
    "Error": "Unable to set UTC offset"
}

```

### 24.8.5 "time leap"

```

Format
{
  "leapSecondTable":
  array_of[
    format_string("%04d-%02d-%02d %d %+d", year, month, day, offset, leap)
  ]
}

Errors
{
  "Error": "Parameter must be 'load', 'save', '59' or '61'"
}

Example
{
  "leapSecondTable":
  [
    "2008-12-31 33 +1",
    "2012-06-30 34 +1",
    "2015-06-30 35 +1",
    "2016-12-31 36 +1",
    "default 37 0"
  ]
}

```

### 24.8.6 "time leap save"

```

Format
{
  "Success": "UTC leap seconds table is saved"
}

Errors
{
  "Error": "Unable to save UTC leap seconds table"
}

```

### 24.8.7 "time leap load"

```

Format
{
  "Success": "UTC leap seconds table is loaded"
}

Errors
{
  "Error": "Unable to load UTC leap seconds table"
}

```

### 24.8.8 "time leap 59"

```

Format
{
  "leap59": boolean
}

```

### 24.8.9 "time leap 61"

```

Format
{
  "leap61": boolean
}

```

**24.8.10 "time leap 59|61 YYYY-MM-DD"**

```

Format
{
    "Success": format_string("Leap second is scheduled: %04d-%02d-%02d %d %d", year, month, day, offset, leap)
}

Errors
{
    "Error": "Year is not correct"
}
{
    "Error": "Month is not correct"
}
{
    "Error": "Day is not correct"
}
{
    "Error": "UTC leap seconds table is corrupted"
}
{
    "Error": "Date must be later than of any already defined leap seconds"
}
{
    "Error": "Unable to schedule leap second"
}

```

**24.8.11 "time timescale"**

```

Format
{
    "timeScale": one_of[ "PTP", "ARB" ]
}

Errors
{
    "Error": format_string("Timescale can not be set. Clock is synchronizing with %s time source",
one_of[ "GPS", "PTP" ])
}
{
    "Error": "Timescale must be 'ptp' or 'arb'"
}

```

**24.8.12 "time update"**

```

Format
{
    "autoUpdaterRTC": boolean
}

Errors
{
    "Error": "Parameter must be 'on' or 'off'"
}

```

**24.8.13 "time holdover"**

```

Format
{
    "holdoverInterval": u32
}

Errors
{
    "Error": "Interval must not be negative"
}

```

## 24.9 Frequency Commands

### Common Errors

```
{
  "Error": format_string("Command '%s' not found", command_name)
}
```

### 24.9.1 "freq status"

#### Format

```
{
  "frequencyInformation":
  {
    "frequencyOption": one_of[ "OPT1", "OPT2" ],
    "referenceQL": quality_level,
    "autoselectionPreference": one_of[ "INPUT", "REF" ],
    "activeSource": one_of[ "REF", format_string("IN_%d", input_number) ],
    "activeQL": quality_level
  }
}
```

#### Example

```
{
  "frequencyInformation":
  {
    "frequencyOption": "OPT1",
    "referenceQL": "QL-SSU-B",
    "autoSelectPreference": "REF",
    "activeSource": "IN_1"
    "activeQL": "QL-PRC"
  }
}
```

### 24.9.2 "freq option"

#### Format

```
{
  "frequencyOption": one_of[ "OPT1", "OPT2" ]
}
```

#### Errors

```
{
  "Error": "Parameter must be 'opt1' or 'opt2'"
}
```

### 24.9.3 "freq ref"

#### Format

```
{
  "referenceQL": quality_level
}
```

#### Errors

```
{
  "Error": one_of[ "Quality level must be 'prc', 'ssu-a', 'ssu-b', 'sec', 'dnu'",
                  "Quality level must be 'prs', 'stu', 'st2', 'tnc', 'st3e', 'st3', 'smc', 'dus' " ]
}
```



#### 24.9.4 "freq prefer"

##### Format

```
{  
  "autoSelectPreference": one_of[ "INPUT", "REF" ]  
}
```

##### Errors

```
{  
  "Error": "Parameter must be 'input' or 'ref'"  
}
```

## 24.9.5 "freq input [input\_number]"

```

frequencyInput:
{
    "inputNumber": u8,
    "inputType": one_of[ "EXTERNAL", format_string("SYNCE_%d", port_number) ],
    "autoSelect": boolean,
    "inputEnabled": boolean,
    "signalFail": boolean,
    "currentQL": quality_level,
    "activeSource": boolean
}

Format
{
    "frequencyInputTable": array_of[ frequencyInput ]
}
{
    "frequencyInput": frequencyInput
}

Errors
{
    "Error": "Input must be 1..3"
}
{
    "Error": "Input type must be 'synce' or 'ext'"
}
{
    "Error": "Port number is missing"
}
{
    "Error": format_string("Port %d not found", port_number)
}
{
    "Error": "Quality level is missing"
}
{
    "Error": one_of[ "Quality level must be 'prc', 'ssu-a', 'ssu-b', 'sec', 'dnu'",
                    "Quality level must be 'prs', 'stu', 'st2', 'tnc', 'st3e', 'st3', 'smc', 'dus' " ]
}

Example
{
    "frequencyInputTable":
    [
        {
            "inputNumber": 1,
            "inputType": "SYNCE_1",
            "autoSelect": true,
            "inputEnabled": true,
            "signalFail": false,
            "currentQL": "QL-PRC",
            "activeSource": true
        },
        {
            "inputNumber": 2,
            "inputType": "SYNCE_2",
            "autoSelect": true,
            "inputEnabled": true,
            "signalFail": false,
            "currentQL": "QL-DNU",
            "activeSource": false
        },
        {
            "inputNumber": 3,
            "inputType": "EXTERNAL",
            "autoSelect": false,
            "inputEnabled": false,
            "signalFail": true,
            "currentQL": "QL-DNU",
            "activeSource": false
        }
    ]
}

```

```

    }
  ]
}

```

## 24.10 SyncE Commands

### Common Errors

```

{
  "Error": format_string("Command '%s' not found", command_name)
}
{
  "Error": format_string("Port %d not found", port_number)
}

```

### 24.10.1 "sync status"

#### Format

```

{
  "syncPortInformation":
  {
    "portNumber": ul6,
    "esmcEnabled": boolean,
    "inputQL": one_of[ "AUTO", quality_level ],
    "outputQL": one_of[ "AUTO", quality_level ],
    "currentLinkMode": one_of[ "NONE", "MASTER", "SLAVE", "MASTER-SLAVE" ],
    "effectiveInputQL": quality_level,
    "effectiveOutputQL": quality_level,
    "activeSource": boolean
  }
}

```

#### Example

```

{
  "syncPortInformation":
  {
    "portNumber": 1,
    "esmcEnabled": true,
    "inputQL": "AUTO",
    "outputQL": "AUTO",
    "currentLinkMode": "SLAVE",
    "effectiveInputQL": "QL-PRC",
    "effectiveOutputQL": "QL-DNU",
    "activeSource": true
  }
}

```

### 24.10.2 "sync esmc"

#### Format

```

{
  "esmcEnabled": boolean
}

```

#### Errors

```

{
  "Error": "Parameter must be 'enable' or 'disable'"
}

```

### 24.10.3 "sync quality"

```

Format
{
    one_of[
        "inputQL": one_of[ "AUTO", ql ],
        "outputQL": one_of[ "AUTO", ql ]
    ]
}

Errors
{
    "Error": "Parameter is missing"
}
{
    "Error": "Parameter must be 'in' or 'out'"
}
{
    "Error": one_of[ "Quality level must be 'auto', 'prc', 'ssu-a', 'ssu-b', 'sec', 'dnu'",
                    "Quality level must be 'auto', 'prs', 'stu', 'st2', 'tnc', 'st3e', 'st3', 'smc', 'dus'" ]
}

```

### 24.11 GPS (PPS/TOD-Input) Commands

```

Common Errors
{
    "Error": format_string("Command '%s' not found", command_name)
}
{
    "Error": "GPS interface is disabled"
}

```

### 24.11.1 "gps status"

#### Format

```
{
  "gpsInterfaceInformation":
  {
    "operationMode": one_of[ "GPS ONLY", "GPS PRIMARY", "GPS SECONDARY", "DEBUG" ],
    "ppsSignalStable": boolean,
    "todSignalStable": boolean,
    "ppsInputEnabled": one_of[ boolean, { "0": boolean, "1": boolean } ],
    "ppsDelay": u32,
    "todInputEnabled": boolean,
    "todInputRequired": boolean,
    "todInputInterface": { "name": one_of[ "DEFAULT", format_string("COM%d", port) ], "baudrate": u32 },
    "todInputFormat": one_of[ "NMEA", "ASCII", "CM" ],
    "mmeaRmcVoidFlag": one_of[ "IGNORE", "SYNCHRONIZE", "HOLD OVER" ],
    "clockClassMapping":
    {
      "active": u8,
      "holdover": u8,
      "free": u8
    }
  }
}
```

#### Example

```
{
  "gpsInterfaceInformation":
  {
    "operationMode": "GPS ONLY",
    "ppsSignalStable": true,
    "todSignalStable": true,
    "ppsInputEnabled": { "0": true, "1": false },
    "ppsDelay": 60,
    "todInputEnabled": true,
    "todInputRequired": true,
    "todInputInterface": { "name": "DEFAULT", "baudrate": 9600 },
    "todInputFormat": "NMEA",
    "mmeaRmcVoidFlag": "HOLD OVER",
    "clockClassMapping": {
      "active": 6,
      "holdover": 7,
      "free": 52
    }
  }
}
```

### 24.11.2 "gps ppsin"

#### Format

```
{
  "ppsInputEnabled": one_of[ boolean, { "0": boolean, "1": boolean } ]
}
```

#### Errors

```
{
  "Error": "Parameter must be 'enable' or 'disable'"
}
```

### 24.11.3 "gps todin"

```

Format
{
    one_of[
        "todInputEnabled": boolean,
        "todInputRequired": boolean
    ]
}

Errors
{
    "Error": "Parameter must be 'enable', 'disable', 'required', 'optional'"
}

```

### 24.11.4 "gps delay"

```

Format
{
    "ppsDelay": u32
}

Errors
{
    "Error": "Delay must be 0..999999999 ns"
}

```

### 24.11.5 "gps class"

```

Format
{
    "clockClassMapping":
    {
        "active": u8,
        "holdover": u8,
        "free": u8
    }
}

Errors
{
    "Error": "Clock class must be 0..255"
}

```

### 24.11.6 "gps interface"

```

Format
{
    "todInputInterface": { "name": one_of[ "DEFAULT", format_sting("COM%d", port) ], "baudrate": u32 }
}

Errors
{
    "Error": "Interface must be 'default' or 'com[n]'"
}
{
    "Error": "COM port must be 1..3"
}
{
    "Error": format_string("Unable to select interface: %s %d", interface_name, baudrate)
}

```

**24.11.7 "gps format"**

```

Format
{
  "todInputFormat": one_of[ "NMEA", "ASCII", "CM" ]
}

Errors
{
  "Error": "Format must be 'nmea', 'ascii', 'cm'"
}

```

**24.11.8 "gps void"**

```

Format
{
  "mmeaRmcVoidFlag": one_of[ "IGNORE", "SYNTONIZE", "HOLDOVER" ],
}

Errors
{
  "Error": "Parameter must be 'ignore', 'syntonize', 'holdover'"
}

```

**24.11.9 "gps message"**

```

Format
{
  "Success": "Message is sent"
}

Errors
{
  "Error": "Not supported on default interface"
}

```

## 24.11.10 "gps cm"

```

cmStatusToClockClassMapping:
{
    "cmStatus": x8,
    "clockClass": u8,
    "clockAccuracy": x8,
    "offsetScaledLogVariance": x16,
    "timeSource": x8,
    "timeTraceable": boolean,
    "frequencyTraceable": boolean
}

Format
{
    "cmStatusToClockClassMappingTable": array_of[ cmStatusToClockClassMapping ]
}
{
    "cmStatusToClockClassMapping": cmStatusToClockClassMapping
}

Errors
{
    "Error": "Status must be 0x00..0x0f"
}
{
    "Error": "Clock class must be 0..255"
}

Example
{
    "cmStatusToClockClassMappingTable":
    [
        {
            "cmStatus": "0x00",
            "clockClass": 6,
            "clockAccuracy": "0x21",
            "offsetScaledLogVariance": "0x4e5d",
            "timeSource": "0x20",
            "timeTraceable": true,
            "frequencyTraceable": true
        },
        {
            "cmStatus": "0x01",
            "clockClass": 140,
            "clockAccuracy": "0xfe",
            "offsetScaledLogVariance": "0xffff",
            "timeSource": "0xa0",
            "timeTraceable": false,
            "frequencyTraceable": true
        },
        {
            "cmStatus": "0x03",
            "clockClass": 160,
            "clockAccuracy": "0xfe",
            "offsetScaledLogVariance": "0xffff",
            "timeSource": "0xa0",
            "timeTraceable": false,
            "frequencyTraceable": false
        },
        {
            "cmStatus": "0x05",
            "clockClass": 150,
            "clockAccuracy": "0xfe",
            "offsetScaledLogVariance": "0xffff",
            "timeSource": "0xa0",
            "timeTraceable": false,
            "frequencyTraceable": false
        }
    ]
}

```



### 24.11.11 "gps nmea"

```
Format
{
}

Errors
{
  "Error": "Only available in NMEA mode"
}
```

## 24.12 Pulsetime (PPS/TOD-Output) Commands

```
Common Errors
{
  "Error": format_string("Command '%s' not found", command_name)
}
```

### 24.12.1 "pulsetime status"

```
Format
{
  "pulsetimeOutputInformation":
  {
    "outputStatus": one_of[ "RUNNING", "PENDING", "STOPPED" ],
    "startTime": time,
    "displayStartTime": "YYYY-MM-DD HH:MM:SS.NS",
    "pulseCountLimit": u32,
    "pulseCount": u32,
    "outputMode": one_of[ "SYNC", "HOLDOVER", "ALWAYS" ],
    "errorTolerance": u32,
    "autoRestart": boolean,
    "pulseProperties": { "period": u32, "width": u32 },
    "timeOutputEnabled": boolean,
    "timeOutputDelay": u32,
    "timeOutputInterface": { "name": one_of[ "DEFAULT", format_sting("COM%d", port) ], "baudrate": u32 },
    "timeOutputFormat": one_of[ { "format": "NMEA", "options": one_of[ "RMC", "ZDA", "RMC+ZDA" ] },
                                { "format": "ASCII", "options": one_of[ "LOCAL", "UTC" ] },
                                "CM",
                                "CUSTOM" ],
    "nmeaRmcVoidFlag": one_of[ "NEVER", "FREE", "HOLDOVER" ],
    "nmeaZdaTzInfo": one_of[ "NONE", "TZ", "TZ+DST" ]
  }
}
```

```
Example
{
  "pulsetimeOutputInformation":
  {
    "outputStatus": "RUNNING",
    "startTime": { "sec": 1581329920, "nsec": 0 },
    "displayStartTime": "2020-02-10 10:18:40",
    "pulseCountLimit": 0,
    "pulseCount": 621,
    "outputMode": "ALWAYS",
    "errorTolerance": 0,
    "autoRestart": true,
    "pulseProperties": { "period": 1000000, "width": 1000 },
    "timeOutputEnabled": true,
    "timeOutputDelay": 0,
    "timeOutputInterface": { "name": "DEFAULT", "baudrate": 9600 },
    "timeOutputFormat": { "format": "NMEA", "options": "RMC+ZDA" },
    "nmeaRmcVoidFlag": "FREE",
    "nmeaZdaTzInfo": "NONE"
  }
}
```

**24.12.2 "pulsetime mode"**

```

Format
{
  "outputMode": one_of[ "SYNC", "HOLDOVER", "ALWAYS" ]
}

Errors
{
  "Error": "Parameter must be 'sync', 'holdover', 'always'"
}

```

**24.12.3 "pulsetime error"**

```

Format
{
  "errorTolerance": u32
}

```

**24.12.4 "pulsetime auto"**

```

Format
{
  "autoRestart": boolean
}

Errors
{
  "Error": "Parameter must be 'on' or 'off'"
}

```

**24.12.5 "pulsetime pulse"**

```

Format
{
  "pulseProperties": { "period": u32, "width": u32 }
}

Errors
{
  "Error": "Period must be at least 500 us"
}
{
  "Error": "Parameter is missing"
}
{
  "Error": "Width must be at least 250 us"
}
{
  "Error": "Width must be at least 250 us shorter than period"
}

```

### 24.12.6 "pulsetime todout"

```

Format
{
    "timeOutputEnabled": boolean
}

Errors
{
    "Error": "Parameter must be 'enable' or 'disable'"
}
{
    "Error": "TOD output cannot be enabled. Pulse period is not an integral number of seconds"
}

```

### 24.12.7 "pulsetime delay"

```

Format
{
    "timeOutputDelay": u32
}

Errors
{
    "Error": "Delay must not be negative"
}

```

### 24.12.8 "pulsetime interface"

```

Format
{
    "timeOutputInterface": { "name": one_of[ "DEFAULT", format_sting("COM%d", port) ], "baudrate": u32 }
}

Errors
{
    "Error": "Interface must be 'default' or 'com[n]'"
}
{
    "Error": "COM port must be 1..3"
}
{
    "Error": format_string("Unable to select interface: %s %d\n", name, baudrate)
}

```

### 24.12.9 "pulsetime format"

```

Format
{
    "timeOutputFormat": one_of[ { "format": "NMEA", "options": one_of[ "RMC", "ZDA", "RMC, ZDA" ] },
                                { "format": "ASCII", "options": one_of[ "LOCAL", "UTC" ] },
                                "CM",
                                "CUSTOM" ]
}

Errors
{
    "Error": "Format must be 'nmea', 'ascii', 'cm', 'custom'"
}
{
    "Error": "Parameter must be 'rmc', 'zda', 'both'"
}
{
    "Error": "Parameter must be 'utc' or 'local'"
}

```

#### 24.12.10 "pulsetime void"

```
Format
{
  "nmeaRmcVoidFlag": one_of[ "NEVER", "FREE", "HOLDOVER" ]
}

Errors
{
  "Error": "Parameter must be 'never', 'free', 'holdover'"
}
```

#### 24.12.11 "pulsetime zone"

```
Format
{
  "nmeaZdaTzInfo": one_of[ "NONE", "TZ", "TZ+DST" ]
}

Errors
{
  "Error": "Parameter must be 'none', 'tz', 'dst'"
}
```

**24.12.12 "pulsetime cm"**

```

clockClassToCmStatusMapping:
{
    "clockClass": u8,
    "cmStatus": x8
}

Format
{
    "clockClassToCmStatusMappingTable": array_of[ clockClassToCmStatusMapping ]
}
{
    "clockClassToCmStatusMapping": clockClassToCmStatusMapping
}

Errors
{
    "Error": "Clock class must be 0..255"
}
{
    "Error": "Status code must be 0x00..0xf"
}

Example
{
    "clockClassToCmStatusMappingTable":
    [
        {
            "clockClass": 6,
            "cmStatus": "0x00"
        },
        {
            "clockClass": 140,
            "cmStatus": "0x01"
        },
        {
            "clockClass": 150,
            "cmStatus": "0x05"
        },
        {
            "clockClass": 160,
            "cmStatus": "0x03"
        },
        {
            "clockClass": 165,
            "cmStatus": "0x04"
        },
        {
            "clockClass": 248,
            "cmStatus": "0x07"
        }
    ]
}

```

**24.12.13 "pulsetime stop"**

```

Format
{
    "Success": one_of[ "PPS-TOD output is stopped",
                      "PPS-TOD output is cancelled" ]
}

```

#### 24.12.14 "pulsetime start"

```

Format
{
    "Success": one_of[ "PPS-TOD output is started",
                       "PPS-TOD output is scheduled" ]
}

Errors
{
    "Error": "Nanoseconds must be 0..999999999"
}
{
    "Error": "Time is missing"
}
{
    "Error": "Year is not correct"
}
{
    "Error": "Month is not correct"
}
{
    "Error": "Day is not correct"
}
{
    "Error": "Hours must be 0..23"
}
{
    "Error": "Minutes must be 0..59"
}
{
    "Error": "Seconds must be 0..59"
}

```

#### 24.13 Nettime (SNTP Service) Commands

```

Common Errors
{
    "Error": format_string("Command '%s' not found", command_name)
}

```

##### 24.13.1 "nettime status"

```

Format
{
    "sntpServerInformation":
    {
        "serverRunning": boolean,
        "onPortNumber": one_of[ "all", format_string("%d", port_number) ],
        "serverMode": one_of[ "UNICAST", "MANYCAST", "BROADCAST", "MIXED" ],
        "logPollInterval": u8
    }
}

Example
{
    "sntpServerInformation":
    {
        "serverRunning": true,
        "onPortNumber": "all",
        "serverMode": "UNICAST",
        "logPollInterval": 4
    }
}

```

### 24.13.2 "nettime mode"

```

Format
{
    "serverMode": one_of[ "UNICAST", "MANYCAST", "BROADCAST", "MIXED" ]
}

Errors
{
    "Error": "Server mode must be 'unicast' 'manycast', 'broadcast', 'mixed'"
}

```

### 24.13.3 "nettime poll"

```

Format
{
    "logPollInterval": u8
}

Errors
{
    "Error": "Log Poll/Broadcast interval must be 0..16"
}

```

### 24.13.4 "nettime stop"

```

Format
{
    "Success": "SNTP server is stopped"
}

```

### 24.13.5 "nettime start"

```

Format
{
    "Success": "SNTP server is started"
}

Errors
{
    "Error": "SNTP server is already running"
}
{
    "Error": format_string("Port %d not found", port)
}
{
    "Error": "Unable to start SNTP server"
}

```

## 24.14 Engine Commands

```

Common Errors
{
    "Error": format_string("Command '%s' not found", command_name)
}

```

**24.14.1 "engine json on|off|ver"**

```

Format
{
    one_of[
        "jsonMode": boolean,
        "jsonVersion": "version_string"
    ]
}

Errors
{
    "Error": "Parameter must be 'on', 'off', 'ver'"
}

```

**24.14.2 "engine init <key>"**

```

Format
{
    "Success": format_string("Clock initialisation requested with key: %s", key)
}

Errors
{
    "Error": "Parameter is missing"
}
{
    "Error": "Parameter must be 'default' or 'nvm'"
}

```

**24.14.3 "engine nvm"**

```

Format
{
    "Success": one_of["Current configuration is saved in NVM",
        "NVM is reset to default configuration" ]
}

Errors
{
    "Error": "Parameter is missing"
}
{
    "Error": "Parameter must be 'save' or 'reset'"
}

```

**24.14.4 "engine debug"**

```

Format
{
    "debugEnabled": boolean
}

Errors
{
    "Error": "Parameter must be 'on' or 'off'"
}

```



#### 24.14.5 "engine verbose"

```

Format
{
    "verboseOn": array_of[ u8 ]
}

Errors
{
    "Error": "Debug output needs to be turned on"
}
{
    "Error": "Level must be 0..23"
}

Example
{
    "verboseOn": [ 8, 15 ]
}

```

#### 24.14.6 "engine monitor"

```

Format
{
    "clockMonitor": one_of[ format_string("COM%d", port), "OFF" ]
}

Errors
{
    "Error": "Parameter is missing"
}
{
    "Error": "Parameter must be 'on' or 'off'"
}
{
    "Error": "COM port must be 1..3"
}
{
    "Error": format_string("Unable to open COM%d", port)
}

```

#### 24.14.7 "engine port [port\_number]"

```

Format
{
    "portEnabled": boolean
}

Errors
{
    "Error": format_string("Port %d not found", port_number)
}
{
    "Error": "Parameter must be 'enable' or 'disable'"
}

```

#### 24.14.8 "engine asymmetry [port\_number]"

```
Format
{
  "portAsymmetry": i16
}

Errors
{
  "Error": format_string("Port %d not found", port_number)
}
{
  "Error": "Asymmetry compensation must be -10000..10000 ns"
}
```

#### 24.14.9 "engine freqout"

```
Format
{
  "freqOut": u8
}

Errors
{
  "Error": "Frequency must be 5, 10, 20, 25 MHz"
}
{
  "Error": "Unable to change frequency output"
}
```

## 25. EULA

### Firmware License for Silicon Labs IEEE 1588-Capable Modules

Your use of this product is conditioned upon your acceptance of the terms of this Firmware License (“Agreement”) between you (either an individual or a single entity for whom you have authority to act) and Silicon Labs. If you do not agree to the terms of this License, you are not authorized to use this product.

#### 1. Definitions

For the purpose of this Agreement:

- **“Product”** means a Silicon Labs IEEE 1588 module or semiconductor purchased by you from Silicon Labs or one of its authorized distributors;
- **“Silicon Labs”** means Silicon Laboratories Inc., a Delaware corporation located at 400 W. Cesar Chavez, Austin, TX 78701 if you are physically located within the United States. If you are physically located outside of the United States, Silicon Labs means Silicon Laboratories International Pte Ltd., a Singapore corporation located at No. 18 Tai Seng Street, #05-01, 18 Tai Seng, Singapore 539775;
- **“Software”** means the firmware programs and associated files in object code form only that are provided by Silicon Labs to you with the Product pursuant to this Agreement;

#### 2. Grant of License

Subject to your full compliance with all of the terms of this Agreement, Silicon Labs grants to you a limited, non-exclusive, non-transferable (except as expressly provided under Section 5 of this Agreement) license to use the Software in conjunction with a Product (the “License”).

#### 3. Reservation of Rights and Ownership

Silicon Labs or its licensors retain all right, title and interest in the Software and all copies thereof, including all copyright, and other intellectual property rights. Silicon Labs reserves all rights not expressly granted to you in this Agreement. The Software is licensed to you and not sold. Portions of the Software are subject to the following open source licenses, the text of which may be found at the respectively listed URL:

Open Source Code License	URL
BSD 2-clause License	<a href="https://opensource.org/licenses/BSD-2-Clause">https://opensource.org/licenses/BSD-2-Clause</a>
BSD 3-clause License	<a href="https://opensource.org/licenses/BSD-3-Clause">https://opensource.org/licenses/BSD-3-Clause</a>
Crypt-DES License (Apache2.0)	<a href="https://www.apache.org/licenses/LICENSE-2.0.txt">https://www.apache.org/licenses/LICENSE-2.0.txt</a>
SSleay (Eric Young) License	<a href="http://publib.boulder.ibm.com/tividd/td/TWS/SC32-1267-00/en_US/HTML/eqqm1mst68.htm">http://publib.boulder.ibm.com/tividd/td/TWS/SC32-1267-00/en_US/HTML/eqqm1mst68.htm</a>
MIT License	<a href="https://opensource.org/licenses/MIT">https://opensource.org/licenses/MIT</a>
RSA Message-Digest License	<a href="https://www.copyleftlicense.com/licenses/rsa-message-digest-license-version-10/view.php">https://www.copyleftlicense.com/licenses/rsa-message-digest-license-version-10/view.php</a>
zlib License	<a href="http://www.nf.mpg.de/vinci3/doc/license-zlib.html">http://www.nf.mpg.de/vinci3/doc/license-zlib.html</a>

#### 4. Limitations on End User Rights

This paragraph applies to all proprietary portions of the Software. You agree that only Silicon Labs has the right to enhance, or otherwise modify, the Software. You have no rights to any source code for the Software. You agree not to, or cause or permit, directly or indirectly, the modification, disassembly, recompilation, or reverse engineering of the Software or any part thereof, or otherwise gain or attempt to gain access to the source code of the Software. You agree not to create derivative works or cause or permit others, directly or indirectly, to create derivative works based upon the Software. The Software is licensed as a single product. Its component parts may not be separated. Except as provided under Section 5 of this Agreement, under no circumstances shall you sell, license, sub-license, publish, display, distribute, rent, lease, assign, or otherwise transfer to a third party the Software or any copy of it, in whole or in part.

#### 5. Software Transfer

If you transfer the Silicon Labs Product on which the Software is used, you may transfer the License to the end user of the Product provided that the end user agrees to be bound by the provisions of this Agreement. Any other transfer is void and automatically terminates this License.

## 6. Updates

Silicon Labs has no obligation to provide updates, bug fixes or error corrections (collectively, “Updates”) to the Software. If Silicon Labs provides an Update, such Update may change or delete the nature or features of the Software, including features that you rely on. You consent to the Updates provided by Silicon Labs. Any Updates shall be deemed to be and shall constitute part of the Software and the terms of this Agreement shall apply to the any and all such Updates.

## 7. Disclaimer of Warranty

YOU EXPRESSLY ACKNOWLEDGE AND AGREE THAT THE USE OF THE SOFTWARE IS AT YOUR OWN RISK. THE SOFTWARE IS PROVIDED “AS IS” AND WITHOUT WARRANTY OF ANY KIND. SILICON LABS EXPRESSLY DISCLAIMS AND MAKES NO WARRANTIES, CONDITIONS, REPRESENTATIONS OR TERMS, EXPRESS OR IMPLIED, WHETHER BY STATUTE, COMMON LAW, CUSTOM OR USAGE OR OTHERWISE AS TO THE SOFTWARE OR ANY COMPONENT THEREOF, INCLUDING BUT NOT LIMITED TO NON-INFRINGEMENT OF THIRD PARTY RIGHTS, INTEGRATION, QUIET ENJOYMENT, MERCHANTABILITY, SATISFACTORY QUALITY OR FITNESS FOR ANY PARTICULAR PURPOSE. SILICON LABS DOES NOT WARRANT OR MAKE ANY REPRESENTATIONS THAT THE FUNCTIONS PROVIDED BY THE SOFTWARE WILL MEET YOUR REQUIREMENTS, OR THAT THE OPERATION OF THE SOFTWARE WILL BE FREE FROM ERRORS OR DEFECTS, EFFECTIVE OR CORRECT. FURTHERMORE, SILICON LABS DOES NOT WARRANT OR MAKE ANY REPRESENTATIONS REGARDING THE USE OR THE RESULTS OF THE USE OF THE SOFTWARE IN TERMS OF ITS FREEDOM FROM ERROR OF DEFECT, EFFECTIVENESS, CORRECTNESS, ACCURACY, RELIABILITY, PERFORMANCE OR OTHERWISE. NO ORAL OR WRITTEN INFORMATION OR ADVICE GIVEN BY SILICON LABS OR ITS REPRESENTATIVES SHALL CREATE A WARRANTY. SILICON LABS DOES NOT WARRANT THAT ANY DEVICE, SOFTWARE OR DATA WILL NOT BE DAMAGED BY THE SOFTWARE.

## 8. Limitation of Liability

SILICON LABS AND ITS RESPECTIVE DIRECTORS, OFFICERS, EMPLOYEES WILL NOT BE LIABLE TO YOU OR ANYONE ELSE FOR ANY INCIDENTAL, SPECIAL, INDIRECT, EXEMPLARY, OR CONSEQUENTIAL DAMAGES FOR BREACH OF ANY WARRANTY, BREACH OF CONTRACT, BREACH OF STATUTE, NEGLIGENCE, STRICT LIABILITY OR ANY OTHER LEGAL THEORY RELATED TO THIS SOFTWARE. SILICON LABS AND ITS RESPECTIVE DIRECTORS, OFFICERS, EMPLOYEES WILL NOT BE LIABLE TO YOU OR ANYONE ELSE FOR ANY LOSS OF PROFITS, LOSS OF REVENUE, LOSS OF DATA, LOSS OF USE OF THE SOFTWARE OR ANY ASSOCIATED HARDWARE OR DEVICE, DOWN TIME AND USER'S DOWNTIME, COST OF PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, OR ANY CLAIM OR DEMAND AGAINST YOU OR THEM BY ANY OTHER PARTY, EVEN IF ANY OF SILICON LABS OR ITS RESPECTIVE DIRECTORS, OFFICERS, AND EMPLOYEES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT SHALL SILICON LABS' AGGREGATE LIABILITY UNDER OR IN CONNECTION WITH THIS AGREEMENT BE GREATER THAN FIFTY UNITED STATES DOLLARS (USD\$50). YOU ACKNOWLEDGE THAT THESE LIMITATIONS ARE REASONABLE, AND FURTHER AGREE THAT THESE LIMITATIONS SHALL APPLY NOTWITHSTANDING ANY FAILURE OF ESSENTIAL PURPOSE OF ANY LIMITED REMEDY.

## 9. Termination

Your rights under this Agreement, including the License granted hereby, are effective until terminated. Your rights under this Agreement and the License will terminate automatically without notice from Silicon Labs if you fail to comply or breach any of the provisions of this Agreement. Upon termination of your rights under this Agreement and the License you must cease to use the Software and the Product immediately. The rights of Silicon Labs under this Agreement shall survive the termination of your rights under this Agreement.

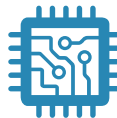


**Connecting Everyone  
and Everything,  
All the Time**



**Portfolio**

[www.skyworksinc.com](http://www.skyworksinc.com)



**Quality**

[www.skyworksinc.com/quality](http://www.skyworksinc.com/quality)



**Support & Resources**

[www.skyworksinc.com/support](http://www.skyworksinc.com/support)

**Copyright © 2021 Skyworks Solutions, Inc. All Rights Reserved.**

Information in this document is provided in connection with Skyworks Solutions, Inc. ("Skyworks") products or services. These materials, including the information contained herein, are provided by Skyworks as a service to its customers and may be used for informational purposes only by the customer. Skyworks assumes no responsibility for errors or omissions in these materials or the information contained herein. Skyworks may change its documentation, products, services, specifications or product descriptions at any time, without notice. Skyworks makes no commitment to update the materials or information and shall have no responsibility whatsoever for conflicts, incompatibilities, or other difficulties arising from any future changes.

No license, whether express, implied, by estoppel or otherwise, is granted to any intellectual property rights by this document. Skyworks assumes no liability for any materials, products or information provided hereunder, including the sale, distribution, reproduction or use of Skyworks products, information or materials, except as may be provided in Skyworks' Terms and Conditions of Sale.

THE MATERIALS, PRODUCTS AND INFORMATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE, INCLUDING FITNESS FOR A PARTICULAR PURPOSE OR USE, MERCHANTABILITY, PERFORMANCE, QUALITY OR NON-INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHT; ALL SUCH WARRANTIES ARE HEREBY EXPRESSLY DISCLAIMED. SKYWORKS DOES NOT WARRANT THE ACCURACY OR COMPLETENESS OF THE INFORMATION, TEXT, GRAPHICS OR OTHER ITEMS CONTAINED WITHIN THESE MATERIALS. SKYWORKS SHALL NOT BE LIABLE FOR ANY DAMAGES, INCLUDING BUT NOT LIMITED TO ANY SPECIAL, INDIRECT, INCIDENTAL, STATUTORY, OR CONSEQUENTIAL DAMAGES, INCLUDING WITHOUT LIMITATION, LOST REVENUES OR LOST PROFITS THAT MAY RESULT FROM THE USE OF THE MATERIALS OR INFORMATION, WHETHER OR NOT THE RECIPIENT OF MATERIALS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Skyworks products are not intended for use in medical, lifesaving or life-sustaining applications, or other equipment in which the failure of the Skyworks products could lead to personal injury, death, physical or environmental damage. Skyworks customers using or selling Skyworks products for use in such applications do so at their own risk and agree to fully indemnify Skyworks for any damages resulting from such improper use or sale.

Customers are responsible for their products and applications using Skyworks products, which may deviate from published specifications as a result of design defects, errors, or operation of products outside of published parameters or design specifications. Customers should include design and operating safeguards to minimize these and other risks. Skyworks assumes no liability for applications assistance, customer product design, or damage to any equipment resulting from the use of Skyworks products outside of Skyworks' published specifications or parameters.

Skyworks, the Skyworks symbol, Sky5®, SkyOne®, SkyBlue™, Skyworks Green™, Clockbuilder®, DSPLL®, ISOModem®, ProSLIC®, and SiPHY® are trademarks or registered trademarks of Skyworks Solutions, Inc. or its subsidiaries in the United States and other countries. Third-party brands and names are for identification purposes only and are the property of their respective owners. Additional information, including relevant terms and conditions, posted at [www.skyworksinc.com](http://www.skyworksinc.com), are incorporated by reference.

Skyworks Solutions, Inc. | Nasdaq: SWKS | [sales@skyworksinc.com](mailto:sales@skyworksinc.com) | [www.skyworksinc.com](http://www.skyworksinc.com)

USA: 781-376-3000 | Asia: 886-2-2735 0399 | Europe: 33 (0)1 43548540 |    

# Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

Skyworks:

[M64](#) [M68](#) [M68R](#) [M64R](#)