
Laureline User Guide

Hardware 7.1, software 4.1

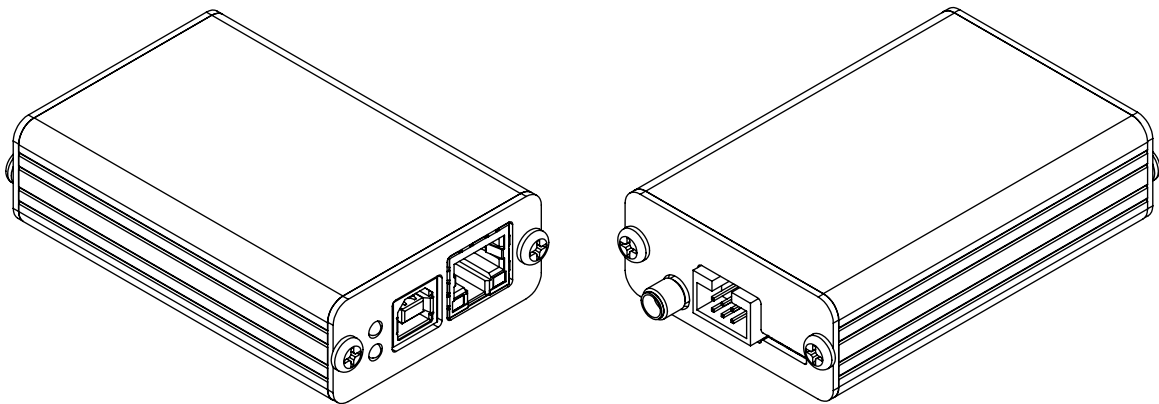
Partially Stapled Productions

Jul 15, 2018

CONTENTS

1	Introduction	1
1.1	Specifications	1
2	Quick Start	3
2.1	Steps	3
3	Configuration	5
3.1	How to connect	5
3.2	Command-line interface	6
3.3	Commands	6
3.4	Settings	7
3.5	Hardware Jumpers	11
3.6	Logging	11
4	Connectors and Indicators	13
4.1	Health	13
4.2	Pulse-per-second	13
4.3	Power and USB	13
4.4	Ethernet LAN	14
4.5	GPS Antenna	14
4.6	Data In/Out	14
4.7	MicroSD	14
4.8	Internal Connectors	14
5	Recommended Deployments	17
6	Theory of Operation	19
7	Firmware Updates	21
7.1	Firmware Changelog	21
8	Troubleshooting	23
8.1	Known Issues	23
9	Resources	25
9.1	Buy Laureline	25
9.2	Firmware download	25
9.3	Source code and Design files	25
9.4	Acknowledgments	26

INTRODUCTION



The Laureline GPS NTP Server is a small, high-performance, and low-power device that serves time to network clients from a built-in GPS receiver. Given only a GPS antenna, a power source, and a clear view of the sky it can track UTC to within 200 nanoseconds and make precise time available to your entire LAN, WAN, or to the Internet. Laureline interoperates seamlessly with any NTP or SNTP client and can sustain thousands of queries per second. Even under high throughput timekeeping operations are never disrupted or perturbed.

Laureline is [Open Source Hardware](http://www.oshwa.org/definition/)¹ and [Open Source Software](http://opensource.org/osd)². Hardware design files and user documentation are provided under the [Creative Commons Attribution 3.0 License](https://creativecommons.org/licenses/by/3.0/)³. Software source code is provided under the [MIT license](http://opensource.org/licenses/MIT)⁴.

1.1 Specifications

- Performance
 - 5000 NTP queries per second
 - 200 μ s round-trip latency
 - Locked to UTC to ± 200 ns < 75ns typical
- Power Supply

¹ <http://www.oshwa.org/definition/>

² <http://opensource.org/osd>

³ <https://creativecommons.org/licenses/by/3.0/>

⁴ <http://opensource.org/licenses/MIT>

- Voltage: 4.0-6.0VDC
 - Consumption: 150mA (0.75W) - not including antenna (typical < 1W)
 - Connector: USB Type B receptacle
- Network
 - 100Base-TX Ethernet
 - Full- or half-duplex
 - Auto-MDIX (auto-crossover)
 - Dedicated Ethernet MAC with minimal jitter and latency
- GPS
 - u-blox GPS receiver module
 - Sensitivity: -161 dBm (tracking)
 - PPS accuracy: 30ns RMS
 - Antenna: SMA receptacle; 5V DC power provided
- NTP and Timing
 - Supports normal queries from NTP and SNTP clients
 - Input capture resolution of 15ns
 - On-board TCXO
 - Software PLL keeps time in native NTP units
- Other features
 - Command-line configuration via USB
 - External GPS data and PPS input, 3.3V and 5V compatible
 - External GPS data and PPS output, 3.3V
 - Field upgrade of software via MicroSD card
 - Lead-free and RoHS-compatible parts and materials
 - Open source hardware and software

QUICK START

To get started with Laureline you will need:

- Laureline GPS NTP server
- GPS antenna with SMA connector and supporting 5V power
- USB A-to-B cable
- Either a PC with a USB port, or a USB charger
- Ethernet cable and a switch, hub, router to connect it to

2.1 Steps

For an illustration of where the following items are located see *Connectors and Indicators*.

1. Connect the GPS antenna to Laureline's antenna port. The connection should be firm but not more than finger-tight. Position the antenna so that it has a clear view of the sky.
2. Connect the LAN port to a switch, hub, router, or PC where you want to serve time. Connecting to a DHCP-enabled network will get things up and running more quickly as no manual configuration is required. If you need to configure Laureline with a static IP, see the *ip_addr* setting under Configuration.
3. Finally, connect the Power/USB port to a PC or USB charger with the USB cable.
4. The status indicators will illuminate steady red at first. The LEDs on the LAN port will also illuminate.
5. After about two minutes, the bottom indicator (PPS) will start flashing red once per second. After another two minutes it will start flashing green. If it does not start flashing after several minutes check the antenna connection and ensure that the antenna has a clear view of the sky.
6. The top indicator (Health) will change to orange when the PPS indicator starts flashing. At this point, the GPS receiver is slowly receiving almanac data from the GPS satellites. After a period of up to 30 minutes this process will complete and the top indicator will change to green.
7. When both indicators are green (steady green on top, flashing green on bottom), Laureline is operating normally and is ready to serve NTP responses.
8. If you are not using a static IP, you will need to discover the IP of the Laureline server. This can be retrieved from the logs of your DHCP server. If your DHCP server is a typical home router, the web interface usually has a list of devices on your network. You can also learn the IP using Laureline's command-line interface, see *the info command* under Configuration.
9. Once both indicators are green, perform a test query using a NTP client of your choice. For example, from a Linux command-line:

```
$ ntpdate -d 192.168.1.200
26 Mar 20:03:25 ntpdate[641]: ntpdate 4.2.6p5@1.2349-o Mon Dec 9 16:28:27 UTC
↪2013 (1)
Looking for host 192.168.1.200 and service ntp
host found : 192.168.1.200
transmit(192.168.1.200)
receive(192.168.1.200)
transmit(192.168.1.200)
receive(192.168.1.200)
transmit(192.168.1.200)
receive(192.168.1.200)
transmit(192.168.1.200)
receive(192.168.1.200)
server 192.168.1.200, port 123
stratum 1, precision -24, leap 00, trust 000
refid [GPS], delay 0.02580, dispersion 0.00002
transmitted 4, in filter 4
reference time: d6dde953.9dbf233a Wed, Mar 26 2014 20:03:31.616
originate timestamp: d6dde953.9dbf233a Wed, Mar 26 2014 20:03:31.616
transmit timestamp: d6dde953.9e66ea78 Wed, Mar 26 2014 20:03:31.618
filter delay: 0.02580 0.02585 0.02583 0.02583
               0.00000 0.00000 0.00000 0.00000
filter offset: -0.00262 -0.00265 -0.00267 -0.00267
               0.000000 0.000000 0.000000 0.000000
delay 0.02580, dispersion 0.00002
offset -0.002629

26 Mar 20:03:31 ntpdate[641]: adjust time server 192.168.1.200 offset -0.002629
↪sec
```

Where 192.168.1.200 is the IP of the Laureline server. A successful response will indicate stratum 1 and the offset to your local clock.

Finally, add Laureline's IP as a source for your NTP daemon(s). If you connected Laureline to a PC, you can move it to an independent power source now that configuration is complete, or you can leave it connected to a PC.

CONFIGURATION

Laureline can be configured via its USB port. When connected to a PC it will enumerate as a serial port. If required, drivers can be downloaded from [FTDI](http://www.ftdichip.com/Drivers/VCP.htm)⁵. You will need a terminal emulator to connect to the serial port. On Windows, PuTTY is recommended. On Linux and BSD “screen” works well.

3.1 How to connect

3.1.1 Parameters

Any serial terminal emulator will work as long as it can connect to the USB serial port. The general parameters are 115200 baud, 8 data bits, no parity, 1 stop bit. This is sometimes written as 115200 8n1. Make sure flow control (RTS/CTS) is disabled.

3.1.2 Windows - using PuTTY

Download [PuTTY](http://www.putty.org/)⁶, either the standalone executable `putty.exe` (no install required), or the full installer `putty-installer.exe`. After starting PuTTY, a window titled “PuTTY Configuration” will appear. In the right-side pane, click “Serial”. In the text box under “Serial line”, type the name of the COM port. You can find a list of COM ports in the Windows Device Manager, where Laureline will be named “USB Serial Port” with the COM port name following it in parentheses. In the “Speed” text box type 115200. Click “Open”. See below for instructions on how to use the command-line. When you are finished, close the window.

3.1.3 Linux/BSD - using screen

screen can be used to interact with a serial port with minimal fuss. To start a screen session with a serial port, or to open a new window from within an existing screen session, type:

```
screen /dev/ttyUSB0 115200
```

Where `/dev/ttyUSB0` is the device that appears when you connected Laureline. Consult `dmesg` or `/var/log/messages` or `journalctl` for logging that indicates which serial port was assigned. When you are finished, type `Ctrl-a`, then `k` (for “kill”), then `y` to confirm.

⁵ <http://www.ftdichip.com/Drivers/VCP.htm>

⁶ <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

3.2 Command-line interface

Normally, Laureline will print status and logging info to the serial port. When you connect you will see a periodic output indicating the health of the GPS and timing loop, as well as possibly raw readings from the PPS input. To configure the system or query its configuration, you must first enter command-line mode by pressing `Enter`. You should see a prompt, starting with a solitary `#` character. You can now enter commands, followed by the `Enter` key. When you are finished, type `exit` and press `Enter`, or press `Ctrl-d`, to return to the status display.

3.3 Commands

3.3.1 defaults

Resets the internal EEPROM to factory defaults and reboots immediately.

3.3.2 exit

Leaves command-line mode and returns to status mode.

3.3.3 help

Lists available command names with a short description.

3.3.4 info

Lists system information including hardware and software version, serial number, MAC address, LAN status, IP address, and system uptime.

3.3.5 save

Saves changes made by the `set` command to internal EEPROM and reboots immediately. Until saved, setting changes have no effect.

3.3.6 set

Displays and modifies settings. With no arguments, it displays the available settings with their current value. Otherwise it takes the form `set param = value`. Until saved with the `save` command, changes have no effect.

3.3.7 uptime

Displays the elapsed time since the system was powered on.

3.3.8 version

Displays hardware, software, and bootloader version information.

3.4 Settings

Settings are viewed and modified with the *set* command. Setting changes must be saved with the *save* command before they will have any effect. Saving causes the system to reboot, which will cause a 2 minute period where NTP service is not available while the PLL settles.

3.4.1 *gps_baud_rate*

Format: integer

Default: 0

Baud rate of the GPS serial port, when *gps_ext_in* or *gps_ext_out* is true. When *gps_ext_out* is true, this must be 57600 or greater.

3.4.2 *gps_ext_in*

Format: boolean (true or false)

Default: false

If true, instead of using the builtin GPS module Laureline will receive GPS data and pulse-per-second from the *Data In/Out port*. In this mode, the PPS connector is used as an input. Use *gps_baud_rate* to configure the baud rate. At present, supported protocols include NMEA, u-blox UBX, Trimble TSIP, and Motorola Oncore. However, only UBX is well-tested and many receivers require special configuration to output all of the data required. Use at your own risk. Not compatible with the *gps_ext_out* or *pps_out* settings.

3.4.3 *gps_ext_out*

Format: boolean (true or false)

Default: false

If true, Laureline will copy raw GPS data from the internal module to the external *Data In/Out port* at the baud rate configured by *gps_baud_rate*. Setting *gps_baud_rate* to less than 57600 baud will cause the output to become corrupted. May be used in combination with *pps_out*. Not compatible with *gps_ext_in*.

3.4.4 *gps_listen_port*

Format: integer

Default: 0

If set to a non-zero value, Laureline will listen for a TCP connection at this port. The client that connects can then receive raw data from the GPS module, and can transmit raw packets to the GPS as well. This feature is experimental and may cause instability or lock-ups. Even when working correctly it is a security risk if exposed to an untrusted network (i.e. the internet). Use at your own risk.

3.4.5 holdover_time

Format: integer

Default: 60

In case of loss of GPS reception or GPS receiver failure, Laureline will continue serving time normally for this many seconds after cessation of pulse-per-second (PPS) input. After this time has elapsed, all NTP responses will be marked with a Stratum value of 16, indicating loss of synchronization. During holdover the status LED will flash red, and after holdover expires the pulse LED will extinguish. It takes up to 5 seconds after loss of PPS to transition from regular operation to holdover mode.

3.4.6 holdover_test

Format: boolean (true or false)

Default: false

If set, then Laureline will measure and log the pulse-per-second signal but will not feed it into the main timing loop. This is useful for testing holdover performance. Set it after the PLL has locked and observe the phase drift over time. This setting will revert to false on powerup.

3.4.7 ip_addr

Format: IP address

Default: 0.0.0.0

If set to a non-zero value, Laureline will use this as its IP address. If set to zero (the default), Laureline will use DHCP to acquire an IP address automatically. If non-zero, *ip_netmask* must also be set and *ip_gateway* should usually be set.

3.4.8 ip_gateway

Format: IP address

Default: 0.0.0.0

If *ip_addr* is set, this should be set to the IP address of the network gateway router. This is not mandatory, but if not set then computers outside of the local network will not receive responses to NTP queries. If you are not sure what your network gateway is, use the `ipconfig` command on your PC. If using DHCP this must be set to zero.

3.4.9 ip_multicast

Format: IP address

Default: 0.0.0.0

If set to a non-zero value, the NTP server will listen on the specified multicast group for queries. Use this with the `manycastclient` option to `ntpd`. Note that `ntpd` requires authentication be working in order to receive manycast replies, see [ntp_key](#).

3.4.10 ip_netmask

Format: IP address

Default: 0.0.0.0

If `ip_addr` is set, this must be set to the associated network mask (subnet mask). The network mask is used to determine whether a given remote IP address is on the same LAN or not. If you are not sure what your network mask is, use the `ipconfig` command on your PC. If using DHCP this must be set to zero.

3.4.11 ip6_manycast

Format: IPv6 address (long form)

Default: 0:0:0:0:0:0:0:0

If set to a non-zero value, the NTP server will listen on the specified IPv6 multicast group for queries. Use this with the `manycastclient` option to `ntpd`. Note that `ntpd` requires authentication be working in order to receive manycast replies, see [ntp_key](#). Shortened IPv6 addresses – those with a “::” in them – are not parsed correctly, and must be expanded to the full 8 segments.

3.4.12 loopstats_interval

Format: integer

Default: 60

Sets the interval, in seconds, at which loop statistics will be logged to the console.

3.4.13 ntp_key

Format: 40 hexadecimal digits

Default: 0

This key is used to validate incoming client queries and to sign outgoing responses. One of [ntp_key_is_md5](#) or [ntp_key_is_sha1](#) must be set in order to select the key type. The key must be the raw, 40 hexadecimal digits. MD5 keys must be converted to hex. Key IDs do not need to be specified because the server will reply with the same ID that the client specified if query authentication succeeds. If the query is not authenticated then the response will also be unauthenticated.

3.4.14 ntp_key_is_md5

Format: boolean (true or false)

Default: false

If true, then *ntp_key* is a 40 digit hexadecimal key for use with the MD5 authentication scheme. Note that ntpd's keys file specifies MD5 keys as 20 plaintext bytes; this must be converted to 40 hexadecimal digits here.

3.4.15 ntp_key_is_sha1

Format: boolean (true or false)

Default: false

If true, then *ntp_key* is a 40 digit hexadecimal key for use with the SHA1 authentication scheme.

3.4.16 pps_out

Format: boolean (true or false)

Default: false

If true, the PPS pin on the *Data In/Out connector* will output a pulse-per-second signal. See the pinout description under Connectors for more electrical info. If false (the default), the PPS pin does not output a signal. Do not set this to true unless you are sure only compatible equipment is connected to the Data In/Out port. This setting is not compatible with the *gps_ext_in* setting.

3.4.17 syslog_ip

Format: IP address

Default: 0.0.0.0

If non-zero, Laureline will forward logging data in the *syslog*⁷ format to the specified IP address. Log lines will be sent in plain UDP format to port 514.

3.4.18 timescale_gps

Format: boolean (true or false)

Default: false

If true, then the GPS timescale will be used instead of UTC. The GPS timescale is offset from UTC by an integral number of seconds (+16 as of the time of this writing), but the offset changes instantaneously when a leap second occurs. Users who wish to have a timescale that does not have the leap second discontinuity may select this option at the cost of being out of sync with UTC.

Selecting this option also speeds up the time from cold boot until NTP is ready, because the server does not need to wait to receive the UTC offset data from the GPS almanac.

⁷ <http://tools.ietf.org/html/rfc5424>

3.5 Hardware Jumpers

If you wish to use Laureline with a 3.3V GPS antenna and are comfortable using a soldering iron, there is a hardware jumper on the PCB that can be changed to adjust the antenna voltage. Performing this modification as described will not void your warranty.

1. Open the enclosure by removing both screws on one end of the chassis. Remove the end panel and slide out the PCB.
2. Look at the bottom of the PCB near the antenna connector. Look for the “J2” designator with the label “ANT PWR”.
3. Using a scalpel or hobby knife, carefully slice the copper track between the center pad and the pad labeled “+5.0v”.
4. Inspect with a jeweler’s loupe or microscope to ensure there is no copper connecting the pads. Optionally, apply power to the board and use a multimeter to confirm that no voltage is present on the center pad.
5. Using a soldering iron, apply a blob of solder between the center pad and the pad labeled “+3.3v”.
6. Before reassembling the enclosure, apply power to the USB connector and use a multimeter to check the voltage present on the SMA antenna connector.
7. Reassemble the enclosure by checking that the board is the right way up and sliding it into the bottom-most channel, adding the end panel, and screwing it into place. Do not over-tighten.

3.6 Logging

When not in command-line mode, log messages are printed to the serial console. These messages show changes in status as well as periodic statistical reports about the performance of the NTP server. For example, shortly after startup and when networking is online, a line like this will appear:

```
2014-07-12T02:17:16.789433Z kernel NOTE GPS NTP Server version v3.0-0-ge9f578d started
```

Logging can also be forwarded to a remote server using the standard `syslog`⁸ UDP protocol, see `syslog_ip`.

In addition to state change messages, there is also a periodic “loopstats” report that indicates the state of the internal clock, similar to this:

```
2014-07-12T03:48:00.899535Z loopstats INFO off:-1ns freq:1433ppb jit:19ns fjit:559ppt
looptc:117s state:4 flags:PPS,ToD,PLL,QUANT
```

The meaning of these fields is as follows:

off Time offset between the internal clock and GPS time in nanoseconds (averaged)

freq Frequency offset between the oscillator and its nominal frequency in parts-per-billion

jit Root-mean-square (RMS) jitter of the time offset in nanoseconds

fjit RMS jitter of the frequency offset in parts-per-trillion

looptc Loop time constant of the internal clock in seconds

state Operating mode of the internal clock PLL. 1 and 2 are starting up. 3 is stable enough to count as “locked”. 4 and 5 are very stable and are the normal operating modes.

⁸ <http://tools.ietf.org/html/rfc5424>

flags A comma-separated list of status flags. Each flag is prefixed with an exclamation point “!” when it is in an error or non-asserted state. The ToD and PLL flags must all be asserted before the server will respond to NTP queries.

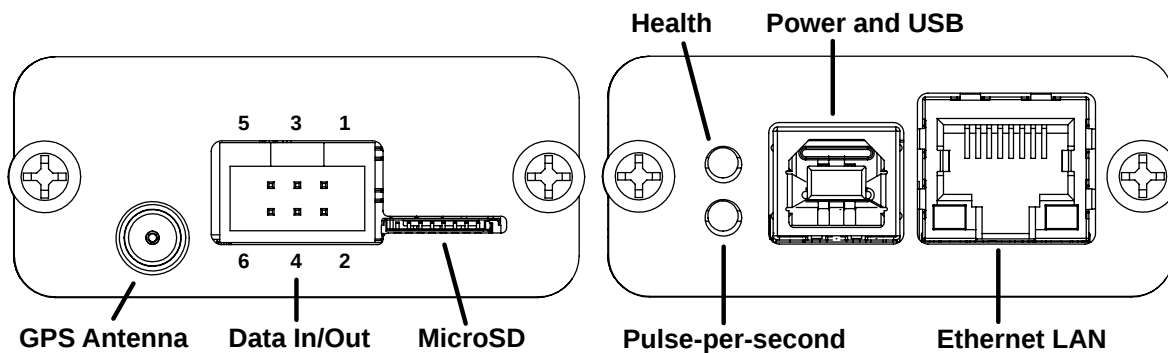
flags - PPS Indicates that the pulse-per-second signal is currently valid. It is cleared about 5 seconds after the signal ceases.

flags - ToD Indicates that valid time-of-day data was received from the GPS. This flag is only set when the UTC offset portion of the GPS almanac has been received, which can take up to 30 minutes after a cold start. Once it is set, it will remain set even if GPS signal is lost.

flags - PLL Indicates that the internal clock (phase-locked-loop) is settled and tracking UTC time closely.

flags - QUANT Indicates that quantization error correction was applied during the last second. This flag should be asserted when using the onboard GPS receiver. When using an external GPS, it may not be asserted depending on the receiver and how it is configured.

CONNECTORS AND INDICATORS



4.1 Health

The upper LED indicates the overall health of the GPS receiver and time-of-day data. Green indicates that correct time-of-day is available. Yellow/orange indicates that data is being received from the GPS but the Almanac is not fully downloaded yet so UTC time-of-day is not available. Yellow/orange will also be displayed if the tracking loop is not settled. Red indicates that time-of-day is not being received.

4.2 Pulse-per-second

The lower LED flashes in synchronization with the beginning of each second. Its color also indicates the status of the tracking loop. Green indicates that the tracking loop is locked and ready to serve time. Flashing red indicates that pulse-per-second is being received from the GPS receiver but the tracking loop is not yet settled. Steady red indicates that no pulse-per-second is being received.

4.3 Power and USB

USB type B receptacle. Used primarily for 5V DC power. If connected to a PC, it enumerates as a serial port which can be used to configure Laureline.

4.4 Ethernet LAN

Modular 8p8c (RJ45) receptacle for 100Base-TX Ethernet. Supports auto-negotiation (10 or 100 Mbit, half- or full-duplex) and Auto-MDIX (no crossover cable required). It is highly recommended to use auto-negotiation and not to “force” the switch port to a particular speed as this can cause compatibility problems and packet loss.

4.5 GPS Antenna

SMA (normal polarity) receptacle. Connect to a powered or unpowered GPS L1 antenna that supports 5V power. 3.3V power is also possible by changing a solder jumper.

4.6 Data In/Out

Multipurpose connector for input and output functions. Can be used to input data and PPS from an external GPS receiver, or to output data and PPS to other equipment. Also supplies a small amount of 3.3V and 5V power or can be used to input 5V power. Connect using an IDC “ribbon” style header. Do not input raw RS232 signals to this header. Data inputs must not exceed 0-5V.

Pinout:

1. Ground
2. Serial data out - 3.3V CMOS level
3. +3.3V power out
4. Serial data in - 3.3V-5V CMOS or TTL level
5. +5V power in/out
6. Pulse-per-second in/out - 3.3V-5V in, or 3.3V out

When the PPS pin is configured as an output, it outputs a buffered, unregistered 3.3V pulse from the GPS module. There is a small series resistor to protect the driver but it is not impedance matched nor can it withstand an indefinite short-circuit. Check compatibility with the equipment you plan to connect. The pulse has a width of 100 ms.

4.7 MicroSD

Optional MicroSD card. Must be formatted with a FAT12, FAT16, or FAT32 filesystem. Currently only used for firmware updates but may be used for logging, configuration, and/or diagnostics in the future.

4.8 Internal Connectors

4.8.1 P2, P3 - Debug

Serial Wire Debug connection for flashing and debugging the main microcontroller. P2 is a “ST-Link” header without the SWV pin. P3 is a Tag Connect pad for use with the TC2030-CTX-NL cable.

4.8.2 P5 - 5V IN

Alternate power input. Connect 5V DC here as an alternative to powering from USB. This is wired directly to the USB power so connecting both at once could damage your PC and/or your power supply. Mind the polarity, too.

4.8.3 P9 - Battery

Optional RTC backup battery. Use a 3V lithium coin cell or similar, not exceeding 3.3V. If present then the GPS receiver will be able to “warm start” within a few seconds after power is restored. The NTP server will not keep time nor answer queries while unpowered even if the backup battery is connected, and 1-2 minutes of settling time is still required after power-up before NTP is available.

RECOMMENDED DEPLOYMENTS

Laureline is a standalone NTP server that uses GPS as its only time source. It does not reference or fall back to other network time sources. This means that in case of a disruption of GPS reception, hardware failure, or global thermonuclear war, Laureline will not be able to answer NTP queries. If this would cause hardship for you, you probably do not want a single Laureline server to be the solitary NTP source for your clients.

For low-risk deployments such as a home network or a small business without stringent redundancy requirements, it is sufficient to simply keep the existing NTP server pool in your clients. As long as Laureline is healthy clients will invariably prefer it as a time source due to the ultra-low latency and jitter of a LAN time source. But in case of a problem, clients can fallback to the internet pool.

If you have dozens of clients you probably do not want them all polling internet time sources, especially when that data will be decimated during normal operation due to the strength of Laureline's data. Consider deploying `ntpd` on two or more general-purpose servers to act as a Stratum 2 intermediate server. Give these servers a normal pool of internet time servers, plus your Laureline server or servers. As before they will prefer Laureline as long as it is available. You can also configure them to peer with each other for additional confidence. Then configure your clients to the Stratum 2 servers. In case of a GPS outage, the Stratum 2 servers will fall back to internet time sources and clients will not be disrupted. This is a good configuration if you have important SNTP clients that do not support a full `ntpd` implementation and cannot query multiple servers. For full `ntpd` clients, you can continue to point them at the Laureline server(s) as well if you need the lowest possible jitter.

THEORY OF OPERATION

Laureline's main hardware elements are a STM32F1 main processor, a NEO-6M GPS receiver, a temperature-compensated crystal oscillator (TCXO), and a dedicated Ethernet transceiver (PHY). The main processor is clocked by an internal PLL which is fed from the TCXO, ensuring a stable time-base.

A dedicated hardware timer-counter in the main processor is used to capture pulse-per-second events from the GPS, which have a rising edge coinciding with the start of each UTC or GPS second. This rising edge is captured and timestamped against an internal "monotonic" timebase which runs at a steady 65-70MHz (depending on the TCXO used).

Concurrently, a virtual timebase ("vtimer") ticks at an adjustable rate relative to the fixed monotonic timebase. Each second, the elapsed number of monotonic ticks since the previous adjustment is divided by the adjusted rate output from the phase-locked loop in order to get the true time elapsed since the previous adjustment. This true time is added to an accumulator to track what the current UTC time is. The accumulated vtimer and monotonic values serve as a base for further UTC time calculations until the next adjustment 1 second later.

Each time a NTP query is performed, the number of monotonic ticks since the last vtimer adjustment is sampled and divided by the PLL output rate. This is added to the accumulator "base" value to get the final UTC time. The vtimer accumulator is in NTP's native timestamp format, so the calculated time is simply copied into the response packet directly and no further precision is lost due to conversion. The Ethernet stack is implemented using dedicated hardware MAC and PHY and is interrupt-driven and DMA-enabled so minimal latency is incurred during a query.

Finally, each second the captured PPS value is compared against the vtimer's calculated start-of-second. The difference between measured and expected value is fed into a software PLL which then produces an adjustment which feeds back into the vtimer to correct its rate of advancement.

FIRMWARE UPDATES

Laureline can be updated in the field by way of a MicroSD card.

To perform a firmware upgrade:

1. Download *ll.hex* from the [Laureline releases page](#)⁹.
2. Format a MicroSD card with a FAT12, FAT16, or FAT32 filesystem. exFAT is not supported.
3. Place the firmware in the root of the MicroSD card and rename it if needed to *ll.hex*
4. Safely eject the MicroSD card and insert it face-up into the slot on Laureline.
5. Cycle power to Laureline or use the *save* command to soft-reset the device.
6. Wait for the status LEDs to illuminate. If you are monitoring the command-line interface it will report progress as well.
7. You may now remove the MicroSD card.

7.1 Firmware Changelog

7.1.1 Version 4.2

- Fixed NTP replies being delayed by an ARP or NDP lookup. Replies are now always returned to the same MAC address from which they originated. (#8¹⁰).
- Faster, zero-copy Ethernet transmission.

7.1.2 Version 4.1

- Fixed DHCP leases not renewing as often as they should (#6¹¹).

7.1.3 Version 4.0

- Added IPv6 support. Only stateless autoconfiguration is supported; static IPs will be added later. The *ip6_manycast* option can be used to join an IPv6 multicast group.

⁹ <https://github.com/mtharp/laureline-firmware/releases>

¹⁰ <https://github.com/mtharp/laureline-firmware/issues/8>

¹¹ <https://github.com/mtharp/laureline-firmware/issues/6>

- Added detailed loopstats logging with additional jitter and time constant data. Reports are emitted periodically, by default every 60 seconds but adjustable using the `loopstats_interval` option. See [Logging](#) for details.
- Added a `timescale_gps` option to select the GPS timescale instead of UTC.
- The `gps_baud_rate` option can now be used in combination with `gps_ext_out` to output data at a faster rate than the internal 57600 baud. However, it may not be set to less than 57600 when used as an output.
- Fixed log messages being sent that had the “time-of-day OK” status flag set before the time-of-day was actually applied (#2¹²).
- Fixed a stability issue that could cause a spontaneous reset, especially under extremely high network load.
- Now based on the FreeRTOS kernel; see [Resources](#).

7.1.4 Version 3.0

- Added configurable holdover time. This allows NTP to keep running for a period of time after GPS signal is lost. See `holdover_time`.
- Added optional NTP auth using MD5 or SHA-1 digests. See `ntp_key`.
- Added NTP “manycast” server support. NTP clients can discover manycast servers on the LAN by sending a query to a multicast group and automatically establishing relationships with servers that respond. See `ip_manycast`. Note that ntpd clients require authentication in order to use this feature.

7.1.5 Version 2.4

- Tweaked the monotonic timer’s input capture algorithm. The new implementation performs identically but is easier to understand.

¹² <https://github.com/mtharp/laureline-firmware/issues/2>

TROUBLESHOOTING

If you are having difficulty using Laureline, the following may be of some use. Don't hesitate to contact us with questions.

Bottom status LED stays solid red The GPS is not tracking satellites and is thus not outputting a pulse-per-second signal. Check that the antenna is firmly connected, that it is compatible with the 5V power Laureline supplies, and that it has a clear view of the sky.

Bottom status LED flashes green but top LED is orange and NTP does not respond The GPS is not sending valid UTC time-of-day data. Before it can do so it must receive a complete almanac from the GPS satellites. This process can take up to 30 minutes.

NTP reports "stratum 16", "Server dropped: no data", or "no server suitable for synchronization found" The GPS receiver has lost satellite lock and is not outputting a pulse-per-second signal. Check that the antenna connection is secure and that it has a clear view of the sky.

Status LEDs and Ethernet LEDs do not illuminate or are dim Disconnect the GPS antenna and see if the problem persists. If it is resolved then your antenna is drawing too much power. If it is not due to a fault then you may need to use a DC block and external power source.

If you have recently flashed a firmware upgrade, the firmware may be corrupt. Try re-downloading the firmware.

8.1 Known Issues

8.1.1 Leap seconds

Laureline does not currently set the leap second bit in NTP replies. This is because the Laureline software itself does not support leap seconds. However, the GPS receiver does. When a leap second occurs Laureline will immediately step its own internal clock in response to the time-of-day data from the GPS and continue serving the correct time. But because the leap bit was not set in any NTP replies, clients that do not receive NTP from any other time source will not step in unison and will be "left behind". Manual intervention will be required e.g. by running `ntpdate` to step the clock, because `ntpd` will reject any time source that has a large offset.

If any other NTP servers are in use that set the leap bit, or if leap second data is manually loaded into the client, then the client will step in unison and continue operating correctly. SNTP clients such as routers and switches are typically not affected as they simply periodically set their local clock to whatever the server replies. Their clock will be off by one second until the next check, then will be correct.

RESOURCES

9.1 Buy Laureline

You can buy the Laureline GPS NTP server at [Tindie](https://www.tindie.com/products/gxti/laureline-gps-ntp-server/)¹³

9.2 Firmware download

Download the latest firmware here: <https://github.com/mtharp/laureline-firmware/releases>

See the *Firmware Updates* chapter for instructions on uploading new firmware.

9.3 Source code and Design files

Software, and the source for the documentation you are currently reading, is published on GitHub:

<https://github.com/mtharp/laureline-firmware>

Or it can be downloaded as a ZIP archive:

<https://github.com/mtharp/laureline-firmware/archive/master.zip>

Hardware design files can be downloaded as a ZIP bundle:

https://partiallystapled.com/pub/laureline/laureline_v7_1b_bundle.zip

If you are not sure where to start with building the software, we highly recommend the *GCC ARM Embedded toolchain*, which is available in binary form for Windows, Linux, and Mac OS X:

<https://launchpad.net/gcc-arm-embedded/+download>

This project uses the [SCons](http://scons.org/)¹⁴ build system. It is available in most Linux distributions; just type “scons” to get started.

¹³ <https://www.tindie.com/products/gxti/laureline-gps-ntp-server/>

¹⁴ <http://scons.org/>

9.4 Acknowledgments

Laureline includes and links against the following third-party software:

- [FreeRTOS¹⁵](#) real-time operating system ([Modified GPL license¹⁶](#))
- [lwIP¹⁷](#) - TCP/IP stack (BSD License)
- [ChaN's FatFs¹⁸](#) - FAT filesystem (permissively licensed)
- [NTPns¹⁹](#) by Poul-Henning Kamp - PLL math (Beer-ware License)
- [OpenSSL²⁰](#) - message digest functions (permissively licensed)

¹⁵ <http://www.freertos.org/>

¹⁶ <http://www.freertos.org/license.txt>

¹⁷ <https://savannah.nongnu.org/projects/lwip/>

¹⁸ http://elm-chan.org/fsw/ff/00index_e.html

¹⁹ <http://phk.freebsd.dk/phkrel/>

²⁰ <http://www.openssl.org/>