PublicNTP

Open Stratum One NTP Server

Dell R240 + Meinberg GPS180PEX v1.0

2019-03-18

Introduction

PublicNTP, Inc., is a 501(c)(3) non-profit public charity incorporated in Virginia, USA, with the mission of deploying network time sources to underserved portions of the world. We are openly publishing a step-by-step guide detailing how our company built a stratum one Network Time Protocol server -- using Global Positioning Satellite (GPS) signals as its upstream time reference -- utilizing Commercial Off The Shelf (COTS) equipment.

PublicNTP hopes that others will find value in this information when building or deploying their own network time sources.

Platform Overview

The system consists of a a 1U Dell PowerEdge R240 paired with a Meinberg GPS180PEX PCI Express GPS receiver.

Total hardware costs came to ~2,700.00 US Dollars ("USD").

Platform Highlights

- Dell PowerEdge R240
  - Can be ordered globally, with option for onsite hardware support
  - Supports mounting in both 2- and 4-post cabinets
- Meinberg GPS receiver
  - GPS signals are available almost anyplace on the planet
  - Paired active antenna reliably synchronizes with GPS signals
  - Accuracy: one-pulse-per-second output within +/- 100ns (**!!!**) of UTC
  - Manufacturer-provided and -supported Linux drivers make it extremely easy to integrate with Linux NTP server software

Overall: the platform's completely open architecture avoids vendor lock-in with an attractive price point, it is easy to install, and is extremely reliable when deployed.

Hardware Ordering

Server (Dell PowerEdge R240)

Ordered configuration

- *Trusted Platform Module*: No
- *Chassis*: 3.5" chassis with up to two cabled hard drives
- *Processor*: Intel® Pentium G5500 3.8GHz, 4M cache, 2C/4T, no turbo (54W)
- *Memory*
    - *Type*: 8GB 2666MT/s DDR4 EC2 UDIMM
    - *Quantity*: 1
- *RAID*: C20, No RAID
- *RAID/Internal Storage Controllers*: No controller
- *Hard drive*
    - *Type*: 1TB 7.2K RPM SATA Entry 3.5in Cabled Hard Drive
    - *Quantity*: 1
- *Additional Network Cards*: On-Board Broadcom 5720 Dual Port 1Gb LOM
- *Additional PCIe cards*: None
- *Embedded Systems Management*: iDRAC9 Basic
- *Internal SD Module*: None
- *Internal Optical Drive*: None
- *Rack Rails*: 1U/2U 2/4-Post Static Rails
- *Bezel*: No Bezel
- *BIOS and Advanced System Configuration Settings:* Performance BIOS Setting
- *Power Cords*:
    - NEMA 5-15P to C13 Wall Plug, 125 Volt, 15 AMP, 10 Feet (3m), Power Cord, North America
    - C13 to C14, PDU Style, 12 Amp, 2 feet (0.6 m), North America
- *Power Supply*: Single, Cabled Power Supply, 250W
- System Documentation: No Systems Documentation, No OpenManage DVD Kit

- *Operating System*: No Operating System
- *OS Media Kits*: No media required
- *Licenses*: None
- *Microsoft SQL Server*: None
- *Advanced System Configurations*: None
- *iDRAC System Management Options*: None
- *IDSDM and VFlash Card Reader*: None
- *Boot Optimized Storage Cards*: None
- *Group Manager*: iDRAC Group Manager, Disabled
- *Password*: iDRAC, Factory Generated Password
- *PCIe Riser*: PCIe Riser with Fan with up to 1LP,x8 PCIe + 1FH/HL, x16 PCIe Slots
- *Optics & Cables for Network Cards*: None
- *Warranty*: 1 year, Basic Hardware Repair, 5x10 HW-Only, 5x10 Next Business Day Onsite
- *Deployment Services*: No Installation
- *Keep your hard drive*: None
- *iDRAC Service Module*: None

Cost as specified: 696.99 USD

Dell official R240 Technical Guide

GPS Receiver (Meinberg GPS180PEX)

PublicNTP orders our Meinberg equipment through JTime, as JTime is the exclusive distributor for Meinberg in the United States.

Receiver, Coaxial Cable, Antenna

Manufacturer: Meinberg
Model: GPS180PEX
Description:
MEINBERG Model GPS180PEX GPS Synchronized Slot Card for PCI Express Bus, includes GPSANT and 20 meters RG58 antenna coax cable.

Qty: 1
Cost (each): 1,735.00 USD

Subtotal: 1,735.00 USD

Surge Suppressor

Manufacturer: Meinberg
Model: MBG S-PRO
Description:
Surge Suppressor. Includes 5m RG58 connecting cable & mounting plate

Quantity: 1
Cost (each): 175.00 USD

Subtotal: 175.00 USD

Shipping: 60.00 USD flat fee

Total: 1,970.00 USD

Serial Port Cable

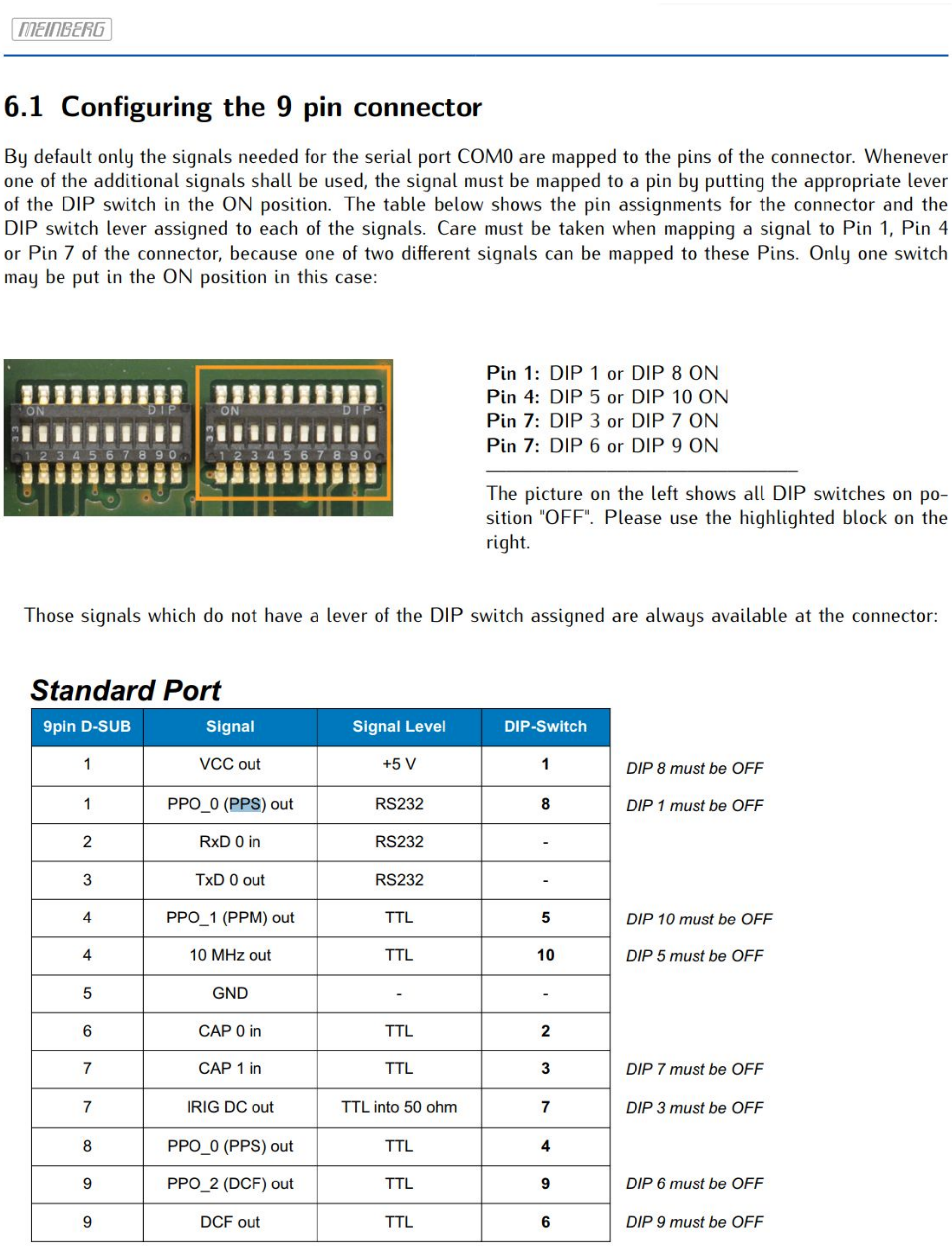
Order a 0.3 meter/1 foot **DB9 male** to **DB9 female** serial cable.

Example: C2G 25211.

Hardware Setup

Pulse Per Second Output on GPS Receiver Serial Port

Configure the highlighted DIP switches as follows:



MEINBERG

6.1 Configuring the 9 pin connector

By default only the signals needed for the serial port COM0 are mapped to the pins of the connector. Whenever one of the additional signals shall be used, the signal must be mapped to a pin by putting the appropriate lever of the DIP switch in the ON position. The table below shows the pin assignments for the connector and the DIP switch lever assigned to each of the signals. Care must be taken when mapping a signal to Pin 1, Pin 4 or Pin 7 of the connector, because one of two different signals can be mapped to these Pins. Only one switch may be put in the ON position in this case:

Pin 1: DIP 1 or DIP 8 ON
Pin 4: DIP 5 or DIP 10 ON
Pin 7: DIP 3 or DIP 7 ON
Pin 7: DIP 6 or DIP 9 ON

The picture on the left shows all DIP switches on position "OFF". Please use the highlighted block on the right.

Those signals which do not have a lever of the DIP switch assigned are always available at the connector:

Standard Port

9pin D-SUB	Signal	Signal Level	DIP-Switch	
1	VCC out	+5 V	**1**	*DIP 8 must be OFF*
1	PPO_0 (PPS) out	RS232	**8**	*DIP 1 must be OFF*
2	RxD 0 in	RS232	-	
3	TxD 0 out	RS232	-	
4	PPO_1 (PPM) out	TTL	**5**	*DIP 10 must be OFF*
4	10 MHz out	TTL	**10**	*DIP 5 must be OFF*
5	GND	-	-	
6	CAP 0 in	TTL	**2**	
7	CAP 1 in	TTL	**3**	*DIP 7 must be OFF*
7	IRIG DC out	TTL into 50 ohm	**7**	*DIP 3 must be OFF*
8	PPO_0 (PPS) out	TTL	**4**	
9	PPO_2 (DCF) out	TTL	**9**	*DIP 6 must be OFF*
9	DCF out	TTL	**6**	*DIP 9 must be OFF*

- **DIP 1**: OFF
- **DIP 8**: ON

Serial Port Cable

Connect the DB9 *male* serial port on the rear of the PowerEdge R240 to the DB9 *female* serial port on the rear of the Meinberg GPS180PEX.

Base Software

The “base” software is that which is required just to confirm the receiver can acquire the GPS satellites.

Install Ubuntu 18.04 LTS

Latest Ubuntu Long Term Support (“LTS”) release as of this writing.

Ubuntu 18.04 LTS gets full updates through October 2020, then maintenance updates from October 2020 through April 2023.

Uncomment source repos in /etc/apt/sources.list

Your new `/etc/apt/sources.list` should begin with:

```
# See http://help.ubuntu.com/community/UpgradeNotes for how to upgrade to
# newer versions of the distribution.
deb http://us.archive.ubuntu.com/ubuntu/ bionic main restricted
deb-src http://us.archive.ubuntu.com/ubuntu/ bionic main restricted

## Major bug fix updates produced after the final release of the
## distribution.
deb http://us.archive.ubuntu.com/ubuntu/ bionic-updates main restricted
deb-src http://us.archive.ubuntu.com/ubuntu/ bionic-updates main
restricted
```

After saving the file,

```
sudo apt-get update
```

Compile Meinberg Linux Tools

Download latest “MBGTOOLS for Linux” (4.2.2 as of this writing) from Meinberg download site.

```
sudo apt-get -y install linux-headers-generic
sudo apt-get -y build-dep linux-headers-generic

make clean
make
sudo make install
sudo modprobe mbgclock
```

Confirm mbgsvcd is running

Make sure mbgsvcd is running

```
ps auxww | grep mbgsvcd
```

Should return “/usr/local/sbin/mbgsvcd -f -q”

Wait for GPS receiver to synchronize

It can take upwards of 20-40 minutes for the Meinberg to fully synchronize with the GPS satellite system for the first time.

To monitor the synchronization status, run:

```
mbgstatus
```

Once locked, the program’s output will be similar to the following:

```
mbgstatus v4.2.1 copyright Meinberg 2001-2018

GPS180PEX 012345678901 (FW 2.04, ASIC 8.06) at port 0xD000, irq 19
Normal Operation, 6 GPS sats tracked, 9 expected to be visible
Date/time:  Su, 2018-12-30  20:42:51.81 UTC
Status info: Input signal available
Status info: Time is synchronized
Status info: Receiver position has been verified
```

High-Quality Time

After running the instructions in this section, your server will be able to deliver time within 20-30 microseconds (or 0.000020 seconds) of UTC.

That’s very accurate time, and more that sufficient for many purposes. Feel free to stop after running the instructions in this section.

If you would like to explore with getting time within 100 *nanoseconds* (or 0.000000100 -- two full orders of magnitude more accurate), continue on with the “Extremely High-Quality Time” section that follows the current section.

Install/configure ntpd

```
sudo apt-get -y install ntp
sudo systemctl stop ntp.service
sudo vi /etc/ntp.conf
```

Update ntp.conf with the following contents:

```
# /etc/ntp.conf, configuration for ntpd; see ntp.conf(5) for help

driftfile /var/lib/ntp/ntp.drift

# Leap seconds definition provided by tzdata
leapfile /usr/share/zoneinfo/leap-seconds.list

# Sync with the meinberg
#    "127.127" is a driver; don't actually go out on the net
#    "28" is the shared memory driver for NTP:
#         http://doc.ntp.org/4.2.8/drivers/driver28.html
#    "0" says use the first index shared mem area (zero-indexed)
#    “prefer” keyword is required for proper operation when using
#         PPS driver
server 127.127.28.0 minpoll 4 maxpoll 4 iburst prefer
fudge  127.127.28.0 refid GPS

# Add other backup sources to use when GPS signal is list,
#    low latency NTP servers over the internet are best
# Examples:
#server stratum2-01.sfo01.publicntp.org
```

```
#server stratum2-01.fra01.publicntp.org
#server stratum2-01.bom01.publicntp.org
#server stratum2-01.syd01.publicntp.org

# By default, exchange time with everybody, but don't allow
configuration.
restrict -4 default notrap nomodify nopeer noquery limited
restrict -6 default notrap nomodify nopeer noquery limited

# Local users may interrogate the ntp server more closely.
restrict 127.0.0.1
restrict ::1
```

Restart ntpd

```
sudo systemctl start ntpd.service
```

Start running `ntpq -p`. At the start you'll see the daemon is not synchronized with any sources:

```
$ ntpq -p
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
 SHM(0)          .GPS.            0 l    6   16   377   0.000   -0.051   0.009
 stratum2-0.sfo0 .GPS.            1 u   36   64   377  36.549    2.118   0.548
 stratum2-0.fra0 .GPS.            1 u   26   64   377  37.459    2.222   2.450
 stratum2-0.bom0 .CDMA.           1 u   22   64   377  26.428    2.367   0.328
 stratum2-0.syd0 .PPS.            1 u   27   64   377  37.234    3.754   0.220
```

After a few minutes, you'll see the NTPd has synchronized with the local GPS hardware (the "SHM" line with the "*"), and will have other fallback candidates marked with a "+" sign that it will use if the local GPS receiver loses synchronization with the GPS satellites.

```
$ ntpq -p
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
*SHM(0)          .GPS.            0 l    6   16   377   0.000   -0.051   0.009
+stratum2-0.sfo0 .GPS.            2 u   36   64   377  36.549    2.118   0.548
-stratum2-0.fra0 .GPS.            2 u   26   64   377  37.459    2.222   2.450
-stratum2-0.bom0 .CDMA.           2 u   22   64   377  26.428    2.367   0.328
+stratum2-0.syd0 .PPS.            2 u   27   64   377  37.234    3.754   0.220
```

At this point your server is providing impressively accurate time to the world. Congratulations!

Extremely High-Quality Time (OPTIONAL)

Running the instructions in this section will improve the accuracy of your time server.

After these instructions, your server will be able to deliver time within 100 *nanoseconds* (or 0.000000100 seconds) of UTC--the most accurate time the GPS receiver is able to provide.

Load Kernel PPS Line Discipline Module Every Boot

Edit `/etc/modules`.

Update the file with the line in bold at the bottom:

```
# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.
pps_ldisc
```

Attach Pulse-Per-Second Line Discipline

Edit `/etc/udev/rules/09-pps.rules`.

Add the following content to the file:

```
KERNEL=="ttyS0", RUN+="/bin/setserial /dev/ttyS0 low_latency"
KERNEL=="ttyS0", RUN+="/usr/sbin/ldattach pps /dev/ttyS0"
KERNEL=="pps0", OWNER="root", GROUP="root", MODE="0600"
```

Reboot the server

```
sudo reboot
```

Confirm System Changes Are In Place After Reboot

```
sudo setserial -a /dev/ttyS0
```

Should report the following:

```
/dev/ttyS0, Line 0, UART: 16550A, Port: 0x03f8, IRQ: 4
        Baud_base: 115200, close_delay: 50, divisor: 0
        closing_wait: 3000
        Flags: spd_normal skip_test low_latency
```

```
$ ls /dev/pps0
```

Should return the following:

```
crw------- 1 root root 243, 0 Mar 18 20:00 /dev/pps0
```

```
sudo apt-get -y install pps-tools
sudo ppstest /dev/pps0
```

Assuming that the serial wiring is correct, should return the following:

```
$ sudo ppstest /dev/pps0
trying PPS source "/dev/pps0"
found PPS source "/dev/pps0"
ok, found 1 source(s), now start fetching data...
source 0 - assert 1552953883.200722267, sequence: 227 - clear
1552955359.002705642, sequence: 1703
source 0 - assert 1552955359.202702433, sequence: 228 - clear
1552955359.002705642, sequence: 1703
source 0 - assert 1552955359.202702433, sequence: 228 - clear
1552955360.002680722, sequence: 1704
```

Update ntpd configuration to use PPS driver

```
sudo systemctl stop ntp.service
sudo vi /etc/ntp.conf
```

Update ntp.conf with the following contents (new section in **BOLD**):

```
# /etc/ntp.conf, configuration for ntpd; see ntp.conf(5) for help

driftfile /var/lib/ntp/ntp.drift

# Leap seconds definition provided by tzdata
leapfile /usr/share/zoneinfo/leap-seconds.list

# Sync with the meinberg
#    "127.127" is a driver; don't actually go out on the net
#    "28" is the shared memory driver for NTP:
#         http://doc.ntp.org/4.2.8/drivers/driver28.html
#    "0" says use the first index shared mem area (zero-indexed)
#    “prefer” keyword is required for proper operation when using
#         PPS driver
server 127.127.28.0 minpoll 4 maxpoll 4 iburst prefer
fudge  127.127.28.0 refid GPS

# Read one-pulse-per-second (1PPS) signal from serial port
#      22.0 means “read from /dev/pps0”
# NOTE: “flag2 1” tells the driver to read from falling
#           edge of 1PPS signal
#       “flag3 1” tells the driver to use kernel PPS discipline
server 127.127.22.0 minpoll 4 maxpoll 4
fudge  127.127.22.0 flag2 1 flag3 1 refid GPS

# Add other backup sources to use when GPS signal is list,
#    low latency NTP servers over the internet are best
# Examples:
#server stratum2-01.sfo01.publicntp.org
#server stratum2-01.fra01.publicntp.org
#server stratum2-01.bom01.publicntp.org
#server stratum2-01.syd01.publicntp.org

# By default, exchange time with everybody, but don't allow
configuration.
restrict -4 default notrap nomodify nopeer noquery limited
restrict -6 default notrap nomodify nopeer noquery limited

# Local users may interrogate the ntp server more closely.
restrict 127.0.0.1
restrict ::1
```

Restart ntpd

```
sudo systemctl start ntpd.service
```

Start running `ntpq -p`. At the start you'll see the daemon is not synchronized with any sources:

```
$ ntpq -p
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
 SHM(0)          .GPS.            0 l    6   16  377    0.000   -0.051   0.009
 PPS(0)          .GPS.            0 l    6   16  377    0.000   -0.051   0.009
-stratum2-0.sfo0 .GPS.            2 u   36   64  377   36.549    2.118   0.548
-stratum2-0.bom0 .GPS.            2 u   26   64  377   37.459    2.222   2.450
-stratum2-0.sin0 .CDMA.           2 u   22   64  377   26.428    2.367   0.328
-stratum2-0.syd0 .PPS.            2 u   27   64  377   37.234    3.754   0.220
-stratum2-0.fra0 216.86.146.46    2 u   62   64  377   23.461    0.291   1.888
```

After several minutes, you'll see the server has locked to a combination of the shared memory ("SHM") driver and the serial port ("PPS") driver:

```
$ ntpq -p
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
*SHM(0)          .GPS.            0 l    6   16  377    0.000   -0.051   0.009
oPPS(0)          .GPS.            0 l    6   16  377    0.000   -0.001   0.001
-stratum2-0.sfo0 .GPS.            2 u   36   64  377   36.549    2.118   0.548
-stratum2-0.bom0 .GPS.            2 u   26   64  377   37.459    2.222   2.450
-stratum2-0.sin0 .CDMA.           2 u   22   64  377   26.428    2.367   0.328
-stratum2-0.syd0 .PPS.            2 u   27   64  377   37.234    3.754   0.220
-stratum2-0.fra0 216.86.146.46    2 u   62   64  377   23.461    0.291   1.888
```

At this point your server is providing the most accurate time to the world that the GPS receiver can provide!