[Chrony](https://chrony-project.org/) — это реализация NTP, но она позволяет использовать преимущества аппаратной поддержки, разработанной для PTP, двумя способами:

* он может считывать данные с входа PPS, подключенного к контакту аппаратных часов PTP (например, к контакту SYNC\_OUT на CM4)
* он может использовать аппаратную временную метку

Эта страница написана с учётом того, что в качестве ОС используется [Fedora](https://github.com/jclark/pc-ptp-ntp-guide/blob/main/fedora.md).

**Входной сигнал PPS**

Chrony может работать очень хорошо, используя только сигнал PPS от GPS: импульс точно указывает, когда начинается секунда; Chrony может определить, какая это секунда, по сетевым источникам. Убедитесь, что вы настроили Chrony так же, как и при получении времени по сети по умолчанию, обычно с помощью директивы pool

Чтобы использовать PPS в качестве источника времени, добавьте эту строку в /etc/chrony.conf:

refclock PHC /dev/ptp0:extpps:pin=0 poll 0 width 0.1 delay 6e-8 precision 2e-8 refid PPS

Здесь:

* /dev/ptp0 это устройство для PHC (аппаратных часов PTP); как его определить, см. в разделе [Проверка PPS](https://github.com/jclark/pc-ptp-ntp-guide/blob/main/service-linux.md#verify-pps)
* опция extpps позволяет включить режим работы, при котором chrony сам считывает временные метки импульсов на выводе PHC; без этой опции chrony ожидает, что PHC будет синхронизирован с правильным временем другим процессом, и просто считывает время с PHC
* pin=0 указано, что нужно использовать вывод 0; это физический вывод, к которому подключен сигнал PPS
* poll 0 рекомендуется использовать образцы сразу, а не хранить их (поскольку дрожание должно быть очень незначительным)
* width 0.1 это ширина импульса в секундах; как её определить, см. в разделе [«Ширина импульса»](https://github.com/jclark/pc-ptp-ntp-guide/blob/main/service-linux.md#pulse-width)
* delay 6e-8 указана задержка в 60 нс; это подходит для GPS с точностью 30 нс
* precision 2e-8 указана точность 20 нс; это приблизительная ожидаемая стабильность показаний PHC
* refid PPS называет этот refclock PPS

Чтобы использовать chrony в качестве сервера в вашей сети, вам также понадобится что-то вроде этого:

allow 192.168.1.0/24

В Fedora вам также потребуется добавить правила брандмауэра:

sudo firewall-cmd --add-service ntp

sudo firewall-cmd --add-service ntp --permanent

Затем перезапустите chrony:

sudo systemctl restart chronyd

Убедитесь, что все началось нормально:

sudo systemctl status chronyd

А теперь беги

chronyc sources

Должна появиться строка, начинающаяся с #\* PPS. Это означает, что она успешно синхронизировалась с эталонными часами PHC.

**Использование последовательного подключения от GPS**

Подключение последовательного выхода GPS позволяет chrony работать даже при отсутствии подключения к какому-либо другому NTP-серверу. Если у вас нет такой необходимости, я не рекомендую этого делать: последовательное подключение к выходу GPS, как правило, менее точное и более хлопотное, чем использование интернета с NTP.

Установить gpsd:

sudo dnf install gpsd

Измените строки OPTIONS и USBAUTO в /etc/sysyconfig/gpsd:

OPTIONS="-n /dev/ttyUSB0"

USBAUTO=""

Теперь мы настроим файл модуля systemd:

sudo systemctl edit gpsd.service

Затем введите

[Unit]

PartOf=chronyd.service

и сохраните файл, а затем закройте редактор. Это устанавливает переопределение в /etc/systemd/system/gpsd.service.d/override.conf, которое гарантирует, что при перезапуске chronyd также будет перезапущен gpsd.

Теперь включите и запустите службу:

sudo systemctl enable --now gpsd.service

А теперь беги

gpsmon

чтобы проверить, видит ли gpsd GPS. Для выхода нажмите Ctrl-C.

Теперь добавьте эту строку в /etc/chrony.conf:

refclock SOCK /run/chrony.clk.ttyUSB0.sock offset 0 delay 0.1 poll 0 noselect refid UART

Смысл этой строки заключается в следующем:

* тип SOCK считывает данные о дате и времени из сокета
* /run/chrony.clk.ttyUSB0.sock это имя файла сокета, который chrony создаст для получения информации от gpsd; когда gpsd начинает считывать данные с /dev/ttyX, он проверяет наличие сокета /run/chrony.clk.ttyX.sock; если этот сокет существует, он будет отправлять сообщения с текущей датой и временем, полученные от GPS через этот сокет
* offset 0 Это означает, что chrony должен предполагать, что между моментом начала отсчёта секунды и моментом получения сообщения о дате и времени для этой секунды через сокет проходит 0 секунд. Это, безусловно, неправильное значение. Мы определим правильное значение ниже.
* delay 0.1 это означает, что разница между минимальным и максимальным смещением составляет 0,1 секунды
* poll 0 это означает, что образцы используются сразу, а не хранятся и не фильтруются (нам не нужно хранить их, так как у нас есть noselect)
* noselect Этот параметр означает, что данный опорный сигнал не будет использоваться в качестве источника времени сам по себе: он слишком неточен для этого. (Вместо этого он будет использоваться для дополнения сигнала в импульсах в секунду.)
* refid UART назвал этот refclock UART

Сейчас

sudo systemctl restart chronyd

Оставьте на минуту или около того, а затем сделайте

chronyc sources

Вы должны увидеть строки, похожие на эти:

#\* PPS 0 0 377 1 -6ns[ -19ns] +/- 867ns

#? UART 0 3 377 9 +216ms[ +216ms] +/- 1434us

а также несколько строк для серверов NTP. 216ms указывает на смещение, которое нам нужно указать в строке refclock SOCK: 216 мс — это 0,216 с, поэтому нам нужно указать 0,216 в качестве значения смещения.

Теперь мы готовы внести последние изменения в две линии опорных тактовых импульсов. Сначала в строке reflock SOCK мы меняем offset 0 на, например, offset 0.216. Затем мы можем добавить опцию lock UART в строку refclock PHC. Это означает, что импульс должен быть объединён с датой и временем из опорного тактового импульса UART, чтобы обеспечить полную и точную выборку времени.

Линии эталонного времени теперь выглядят следующим образом:

refclock PHC /dev/ptp0:extpps:pin=0 poll 0 width 0.1 delay 6e-8 precision 2e-8 refid PPS lock UART

refclock SOCK /run/chrony.clk.ttyUSB0.sock delay 0.1 offset 0.216 poll 0 noselect refid UART

TODO: лучше ли убрать lock UART из строки PHC и убрать noselect из строки UART?

Теперь мы можем снова запустить Chrony:

sudo systemctl restart chronyd

Функция /run/chrony.clk.ttyX.sock была добавлена в gpsd 3.25. До этого она поддерживала только /run/chrony.ttyX.sock, который работает только с данными PPS. Если вы используете версию gpsd, которая не поддерживает /run/chrony.clk.ttyX.sock, вам следует использовать опорный тактовый сигнал SHM 0.

refclock SHM 0 poll 0 delay 0.1 offset 0.216 noselect refid UART

**Аппаратная временная метка**

Чтобы chrony использовал аппаратную метку времени, мы просто добавляем строку

hwtimestamp enp1s0

Но это не сработает с клиентами, чьи сетевые карты имеют возможность устанавливать временные метки только для пакетов PTP. Чтобы поддерживать таких клиентов, нам также нужно использовать [NTP-over-PTP](https://datatracker.ietf.org/doc/draft-ietf-ntp-over-ptp/) для установки аппаратных временных меток. Чтобы включить эту функцию, мы добавляем строку

ptpport 319

Нам также потребуется добавить правила брандмауэра для разрешения входящего трафика на порту PTP:

sudo firewall-cmd --add-service ptp

sudo firewall-cmd --add-service ptp --permanent

На стороне клиента вам понадобится что-то вроде этого:

server 192.168.1.2 minpoll 0 maxpoll 0 xleave port 319

hwtimestamp enp1s0 rxfilter ptp

ptpport 319

Вы также можете добавить extfield F323 в строку server, чтобы включить поддержку chrony для [передачи частоты в NTP](https://mlichvar.fedorapeople.org/ntp-freq-transfer/).

Chrony поддерживает только NTP-over-PTP между сервером и клиентом, использующими одинаковые версии chrony.