

# IoT デバイス間の時刻同期に関する研究

奥田 誠(情報・生産技術部 システム技術グループ)

## 1. はじめに

近年、IoT や Industrie 4.0 を背景に、センサなどのデバイスは増加傾向にあり、振動や音、電力などのセンサデータを解析することで、装置の故障診断や異常検出などの付加価値が期待されている。IoT デバイスに接続されたセンサから得られる時系列データに対して、正確な解析を行うためには、それぞれのデータの時間軸を一致させることが重要である。時間軸を一致させるには、異なる IoT デバイスで時刻同期を行えばよく、各 IoT デバイスに時刻同期を行う方法はすでにいくつか存在している。

一つは、IoT デバイスそれぞれに、GPS アンテナを接続して、GPS 時刻を取得・設定する方法である。これは、実際に工場内に設置することを考えた場合に現実的ではない手法である。GPS の電波を受信するために、窓際に設置する必要があるが、装置が窓近くに設置されているとは限らず、また大量の IoT デバイスを用いる場合には GPS アンテナも同数用意する必要がある。

もう一つは、NTP<sup>1)</sup> (Network Time Protocol)を用いる手法である。これは、LAN でネットワーク接続されているデバイスに対して、LAN 経由での時刻同期を可能とし、設置は簡単であるが、同期精度が数 ms ~ 数百 ms であるため、高時間分解能のデータが必要となる振動や音の解析には不十分である可能性がある。また、正確な時刻を設定したい場合には、インターネットに接続する必要がある。

本研究では、工場への設置を想定し、GPS アンテナを1個のみ用いたシステム構成で、NTP よりも高精度な時刻同期技術の PTP<sup>2)</sup> (Precision Time Protocol, IEEE 1588)で、複数のデバイスに対して時刻同期を行うシステムを試作し、その時刻同期の精度を実測した結果について報告する。

## 2. 実験と結果

### 2. 1 実験構成

図1に示すシステム構成で実験を行った。同期する時刻の基になるマスタークロック(他のデバイスへ配信する時刻)には、Raspberry Pi 3B (master) (以下、マスターという)のシステムクロックを用いる。マスタークロックには、GPS (Global Positioning System) HAT(Hardware Attached on Top)を介して接続した GPS アンテナから取得した正確な時刻を設定する。時刻同期されるデバイス(以下、スレーブという)の実装種別により、2台のデバイスを用いた。TS とは、日付と時刻を表すデータであり、あるデータに対して追加することで時刻情報の付与を可能とし、一般的には実装方法の種類によりハードウェア TS とソフトウェア TS に区

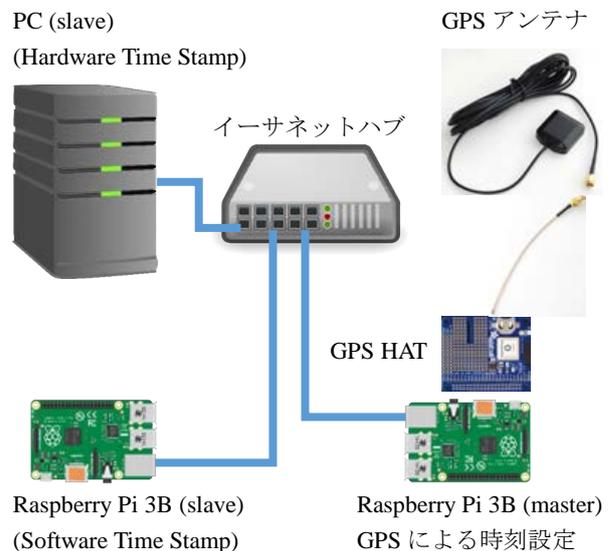


図1 実験システム構成図

分される。ハードウェア TS の PC (slave)およびソフトウェア TS の Raspberry Pi 3B (slave)を、スレーブとして用いる。マスターとスレーブは、イーサネットハブを介して、LAN ケーブルでスター接続している。また、時刻精度向上のため、GPS HAT 上の GPS 受信機から出力される 1PPS (Pulse Per Second)信号を用いている。1PPS 信号とは、正確に1s 間隔で、約 100ms のパルス幅で、出力される信号のことである。時刻同期には、PTP を用いる。これらの機能を実装するのに使用したソフトウェアパッケージ一覧を表1に示す。

### 2. 2 実験方法および結果

時刻同期の精度は、基準となるデバイスの時刻から、あるデバイスの時刻に対する時刻差(Offset)を取得して、その最小値と最大値の差(range)を評価した。表2に示した a), b), c) それぞれのデバイス間に対して、縦軸を時刻差、横軸を時間(24h 分)としたグラフを図2に示す。表2 a)のデバ

表1 使用したソフトウェアパッケージ一覧

技術略称	用途	パッケージ名
GPS	正確な時刻を取得、マスタークロックへの時刻設定	gpsd <sup>4)</sup> , gpsd-client <sup>5)</sup> , python-gps, ntpd
1PPS	時刻精度向上	pps-tools <sup>6)</sup>
PTP	各デバイスへの時刻同期	Linuxptp <sup>7)8)</sup> (ptp4l <sup>9)</sup> , phc2sys <sup>10)</sup>

イス間では、NTP のステータス情報である loopstats<sup>11)</sup>ファイルの内容から、約 16s 間隔で 24h 分の時刻差を取得した。表 2 b), c) のデバイス間では、Unix 系 OS のログファイルである syslog ファイルから、約 1s 間隔で 24h 分の時刻差を取得した。図 2 a) については、単純に GPS 時刻とマスタークロック間の時刻差となり、約  $12\mu\text{s}$  である。他の IoT デバイスも同様に、GPS 時刻との時刻差が約  $12\mu\text{s}$  と想定されるため、各 IoT デバイスにそれぞれ GPS アンテナを接続する IoT システムにおける時刻同期精度は、これらを加算した値の約  $24\mu\text{s}$  であると考えられる。図 2 b) は、マスターとソフトウェア TS のスレーブ間の時刻同期精度を表しており、約  $325\mu\text{s}$  である。図 2 c) は、マスターとハードウェア TS のスレーブ間の時刻同期精度を表しており、約  $100\mu\text{s}$  である。したがって、ソフトウェア TS とハードウェア TS の違いにより、時刻同期精度が約 3 倍異なる。数

ms ~ 数百 ms の時刻同期精度の NTP と比較すると、PTP ではソフトウェア TS で約 3 倍、ハードウェア TS では約 10 倍の精度が得られることが分かった。

### 3. 考察および今後の展望

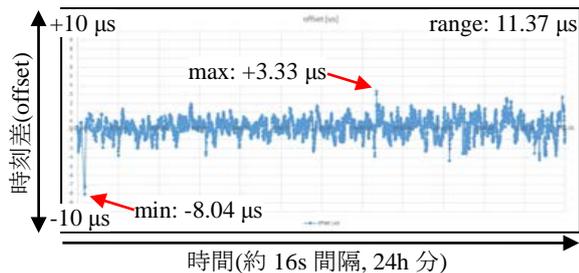
ソフトウェア TS およびハードウェア TS で同期精度が異なる要因について考察した。ソフトウェア TS では、大半のデータは  $\pm 50\mu\text{s}$  以内にあるが、一部のデータにおいて、時刻差の絶対値が大きくなっているため、CPU が他の処理を優先している場合があるためと推測できる。一方、ハードウェア TS では、他の処理に負荷を取られるようなことがないため、24h 分の全データにおいてほとんどバラつきがなく、 $\pm 50\mu\text{s}$  以内に収まっている。

IoT デバイス間に必要な時刻同期精度は、データ活用の目的や対象装置の環境によって異なる。例えば、人の不快度指数を求めるための湿度・温度などのデータでは、1 分間隔程度での取得で十分であり、時刻同期も NTP で十分である。しかし、高速に動作する工作機械などに対して設備の故障診断や異常検出、加工品の良否診断を目的とした場合、振動や音、電力などのデータを高時間分解能で取得する必要があることが想定される。この場合、異なる IoT デバイスでデータの時間軸を合わせるために、高精度での時刻同期が必要となる。

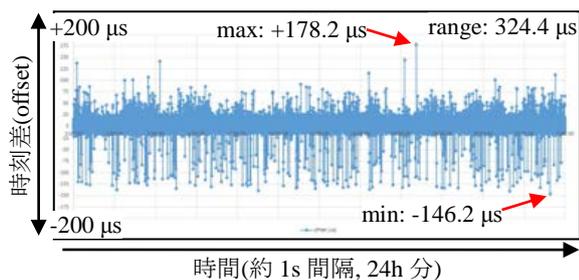
今後、IoT デバイス間の時刻同期の精度の違いにより、設備の故障診断や加工品の良否診断にどの程度影響を与えるか調査するため、工作機械に対して振動や音、電力などのデータを取得する IoT システムを開発し、実データの解析を試行する予定である。

表 2 時刻差測定条件

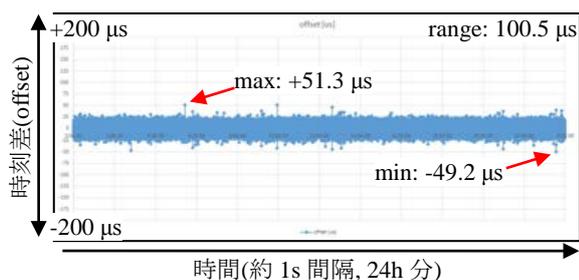
	基準デバイス	時刻差測定デバイス
a)	GPS アンテナ	Raspberry Pi 3B (master)
b)	Raspberry Pi 3B (master)	Raspberry Pi 3B (slave) (Software Time Stamp)
c)	Raspberry Pi 3B (master)	PC (slave) (Hardware Time Stamp)



a) GPS アンテナと Raspberry Pi 3B (master)



b) Raspberry Pi 3B (master) と Raspberry Pi 3B (slave)



c) Raspberry Pi 3B (master) と PC (slave)

図 2 時刻同期精度の実験結果

#### 【参考文献】

- 1) NTP web site, <http://www.ntp.org/>
- 2) IEEE-1588 Standard Ver.2 Tutorial Agilent Technologies
- 3) Timestamp wiki, <https://en.wikipedia.org/wiki/Timestamp>
- 4) gpsd web site, <https://gpsd.gitlab.io/gpsd/index.html>
- 5) GPSD Client HOWTO web site, <https://gpsd.gitlab.io/gpsd/client-howto.html>
- 6) Ubuntu 18.04 LTS pps-tools web site, <http://manpages.ubuntu.com/manpages/bionic/man8/pps-tools.8.html>
- 7) linuxptp web site, <http://linuxptp.sourceforge.net/>
- 8) raspberrypi-ptp web site, <https://github.com/twteamware/raspberrypi-ptp>
- 9) ptp4l を使用した PTP の設定 web site, [https://access.redhat.com/documentation/ja-jp/red\\_hat\\_enterprise\\_linux/7/html/system\\_administrators\\_guide/ch-configuring\\_ptp\\_using\\_ptp4l](https://access.redhat.com/documentation/ja-jp/red_hat_enterprise_linux/7/html/system_administrators_guide/ch-configuring_ptp_using_ptp4l)
- 10) クロックの同期 (phc2sys) web site, [https://access.redhat.com/documentation/ja-jp/red\\_hat\\_enterprise\\_linux/7/html/system\\_administrators\\_guide/sec-synchronizing\\_the\\_clocks](https://access.redhat.com/documentation/ja-jp/red_hat_enterprise_linux/7/html/system_administrators_guide/sec-synchronizing_the_clocks)
- 11) NTP documentation web site, <http://doc.ntp.org/current-stable/monopt.html>