Contents

[1. Exploring the principles of time 2](#_Toc192088350)

[1.1What is Clock? 3](#_Toc192088351)

[1.2 Development in Timekeeping 5](#_Toc192088352)

[2. Introduction to Time Scales 6](#_Toc192088353)

[2.1 Standardizing Time: Greenwich Mean Time (GMT) 7](#_Toc192088354)

[2.2 Coordinated Universal Time (UTC) 8](#_Toc192088355)

[3. Time and Frequency Dissemination 9](#_Toc192088356)

[3.1 Time Dissemination 9](#_Toc192088357)

[3.2 Time Dissemination over the Network 10](#_Toc192088358)

[3.3 Traceability in Timing 15](#_Toc192088359)

[4.Fetching and Displaying IST Time Using NTP Server on Arduino with Ethernet Shield 17](#_Toc192088360)

[5. Introduction to NTP and Chrony 19](#_Toc192088361)

[5.1 What is NTP? 19](#_Toc192088362)

[5.2 What is Chrony? 19](#_Toc192088363)

[5.3 Components of Chrony 20](#_Toc192088364)

[5.4 Difference between Chrony and NTP 21](#_Toc192088365)

[6. Raspberry Pi as NTP Chrony Server 23](#_Toc192088366)

[6.1 Procedure 23](#_Toc192088367)

[6.2. Configure Clients to Use the Raspberry Pi NTP Server: 25](#_Toc192088368)

[7.Integrating DS3231 Real-Time Clock (RTC) with Raspberry Pi and Chrony for Time Synchronization 31](#_Toc192088369)

[8.Raspberry Pi into a Stratum 2 NTP server using Chrony with hardware time stamping 34](#_Toc192088370)

[9.Integration of u-blox ZED-F9T with Raspberry Pi 4 for a Stratum 1 NTP Server Using PPS Input 37](#_Toc192088371)

# 1. Exploring the principles of time

Time - The Hidden Utility

Time is an integral concept in our daily lives and across various industries. Its significance ranges from helping us catch the fastest train, schedule international phone calls, and plan our day, to playing a crucial role in sectors such as transportation, energy, telecommunications, and finance.

**The Importance of Time in Various Industries**

**Transport**

Timing measurements are essential in satellite positioning systems, which form the backbone of modern air traffic control systems. Precise timing ensures that planes can be tracked and directed safely and efficiently.

**Energy**

In the energy sector, synchronised measurements of voltage and current are necessary for controlling the power grid. Synchronisation inherently requires precise timing to ensure the stability and efficiency of energy distribution.

**Telecommunication**

Telecommunication companies rely on timing measurements to synchronise their communication networks. This synchronisation is vital for maintaining the flow and integrity of data across vast networks.

**Finance**

In the finance sector, regulations mandate that trades be timestamped to maintain accurate records of when transactions occur. This ensures transparency and accountability in financial markets.

Given the pervasive reliance on timing in these sectors, time is often described as a ‘hidden utility,’ fundamental yet not always visible to the consumer.

**The Personal Utility of Time Measurements**

Timing measurements extend beyond industries to individual use. For instance, they enable the functionality of map applications on our phones and car satnav systems, providing accurate location information.

**Defining Time and the Second**

The philosophical question "What is time?" has no straightforward answer. However, focusing on the measurement of time, we start with defining the second, a unit we are all familiar with. The second, symbolized as 's', is the SI unit of time. It is defined by taking the fixed numerical value of the caesium frequency, ΔνCs, which is the unperturbed ground-state hyperfine transition frequency of the caesium 133 atom. This frequency is 9,192,631,770 Hz when expressed in the unit Hz, equal to s−1.

# 1.1What is Clock?

**Periodic Processes and Time Measurement**

A periodic process is one that repeats regularly, such as the beating of your heart, the swinging of a pendulum, or the tides. These processes help describe time because time is constructed from counts of complete cycles of a periodic system. For example, we measure time by the number of complete oscillations.

**Frequency and Time Periods**

The number of oscillations per unit time is called the frequency of the process. When the unit time is a second, the frequency is expressed in hertz (Hz). The period of a process is the time taken for one complete cycle. In the International System of Units (SI), the second is one of the seven base units, and the hertz is a derived unit: 1 Hz = 1 s−1.

To measure time accurately, the cycles of a periodic system should be equally spaced, meaning the frequency should be constant. In practice, achieving perfect regularity is impossible, so the goal is to make the frequencies as constant as possible for the intended purpose.

**Examples of Clocks**

Various methods exist to measure the passage of time, from hourglasses to atomic clocks. Here, we examine four types of clocks to understand the common elements that define them.

**Sundial**

Sundials use the Earth's rotation about its axis to indicate time. The shadow of the gnomon moves across the sundial face as the Earth rotates, providing an approximate local time reading.

**Pendulum Clock**

Pendulum clocks use the swinging of a weight to track time. This motion is linked to the gears and face of the clock, providing a time reading.

**Mechanical Clock**

Mechanical clocks contain a balance wheel that oscillates at a specific rate, driven by coiled springs. A series of gears move the hands around the clock face at appropriate rates.

**Quartz Clock**

Quartz clocks use periodic electrical signals from a quartz crystal. These signals are linked to gears (in analogue watches) or the electrical system (in digital watches) to display the time.

**The Three Elements of Clocks**

Despite their differences, all clocks share three elements:

1. **Oscillator**: A mechanism that performs a periodic process repeatedly.
2. **Counter**: A mechanism to count the number of oscillations.
3. **Frequency Reference**: Ensures the oscillations run at a known and desired rate.

**Sundial**

* **Oscillator**: Earth's rotation about its axis.
* **Counter**: Sundial face and shadow movement.
* **Frequency Reference**: The day-night cycle, which originally defined the SI second.

**Pendulum Clock**

* **Oscillator**: Pendulum swing.
* **Counter**: Clock gears and face.
* **Frequency Reference**: Pendulum length set to match a swing period of 1 SI second.

**Mechanical Clock**

* **Oscillator**: Balance wheel driven by a coiled spring.
* **Counter**: Clock gears and face.
* **Frequency Reference**: Resonance of the balance wheel aligned to a known rate traceable to the SI second.

**Quartz Clock**

* **Oscillator**: Electronic oscillations from a quartz crystal.
* **Counter**: Clock gears and face (analogue) or electronic counter and display (digital).
* **Frequency Reference**: Quartz crystal cut to a specific shape so its resonance is traceable to the SI second.

**Characteristics of Clocks**

Clocks are evaluated based on three characteristics: stability, accuracy, and resolution.

**Stability**

Stability describes how unchanging a clock's period of 'ticking' is. For example, a heartbeat would not make a good oscillator due to its variability. Metrologists quantify clock stability using measures like Allan deviation.

**Accuracy**

A clock is accurate if it tells the correct time. Accuracy can be compromised if the clock is set incorrectly initially or if it runs at the wrong rate. Adjusting a clock to the right time requires access to a more accurate clock.

**Resolution**

Resolution is the smallest change that can be measured by the clock. For instance, a stopwatch with a resolution of 0.01 seconds can measure time intervals in chunks of 0.01 seconds. Higher resolution clocks are necessary for applications requiring precise timing, such as high-frequency trading in finance or synchronising power grids.

# 1.2 Development in Timekeeping

Timekeeping has evolved from using astronomical methods to atomic clocks. Natural periodic phenomena, such as the Earth's revolution around the Sun and its rotation on its axis, have long been used to measure time.

**Units of Time Linked to Astronomical Processes**

* **Year**: The Earth takes approximately 365.25 days to orbit the Sun.
* **Day**: The Earth takes 24 hours to complete one rotation on its axis.
* **Month**: Related to the moon's revolution around the Earth.

Astronomers distinguish between a solar day (24 hours) and a sidereal day (23 hours 56 minutes), based on the Earth's rotation relative to the Sun and the stars, respectively.

**Refining Units of Time**

Time has been divided into increasingly smaller units, such as hours, minutes, seconds, and even nanoseconds, to suit various needs. The definition of the second has shifted from astronomical to atomic processes, reflecting advancements in clock technology.

**The Move to Atomic Time**

Atomic clocks use the interaction of atoms with electromagnetic radiation as a highly stable frequency reference. The SI second is now defined based on the caesium atom's interaction with electromagnetic radiation.

**Why Caesium?**

Caesium was chosen for several reasons:

* It has a stable isotope, ensuring uniform atomic behaviour.
* Measuring microwave frequencies (associated with caesium) was feasible with existing technology.
* Caesium is easily ionised, facilitating early atomic clock designs.

**Advantages of Atomic Time**

Atomic clocks offer unmatched stability and resolution. Their precision outperforms even the Earth's rotation, making atomic time a superior reference. This uniformity is beneficial for global and potentially interstellar timekeeping.

**Principles of Passive and Active Atomic Clocks**

Atomic clocks can be categorized as passive or active. Passive clocks measure the proportion of atoms in different states, while active clocks use the feedback loop of atomic interactions to stabilize the oscillator.

**The Big Picture of Atomic Clocks**

Regardless of type, atomic clocks share common principles: using electromagnetic radiation as the oscillator and atomic interactions as the frequency reference. This technology underpins the modern definition of the SI second, providing the most accurate timekeeping method available.

In conclusion, the concept of time, though seemingly abstract and philosophical, is deeply embedded in practical applications across various sectors. The evolution of timekeeping, from sundials to atomic clocks, reflects humanity's quest for precision and consistency in measuring this fundamental dimension. The integration of atomic time into our daily lives and industries underscores its importance as a hidden utility, essential yet often unnoticed. As technology advances, the precision and reliability of our timekeeping methods will continue to shape and enhance our world.

# 2. Introduction to Time Scales

We explored how clocks measure time intervals by counting cycles of periodic processes. This module will delve into the concept of time scales, examining how they allow us to tell 'the time' and the evolution of the current standard time scale, Coordinated Universal Time (UTC). Additionally, we'll explore how UTC relates to astronomical methods of timekeeping.

**Understanding Time Scales**

While all clocks measure time intervals, not all of them indicate 'the time'. For instance, a stopwatch can measure the 9 minutes it takes to bake cookies, but it doesn't tell you the time of day. Knowing 'the time' involves orienting events within a larger framework, which requires a time scale.

A time scale is a system that allows unambiguous ordering of events, helping us communicate about past occurrences and future expectations. Effective time scales need:

1. **Frequency Reference:** Ensures the system runs at a stable, desired rate.
2. **Time Reference:** An agreed-upon time that synchronizes all clocks.

In practice, no clock or frequency reference is perfect. Thus, a practical time scale uses a network of clocks and frequency references, continuously compared and adjusted to provide 'the time'.

**Solar Time and Its Limitations**

Historically, time scales were based on the Sun's apparent movement across the sky. Solar time relies on the position of the Sun, which varies due to the Earth's elliptical orbit and tilted axis, causing solar days to differ in length by about 50 seconds throughout the year.

**Apparent Solar Time**

Apparent solar time is based on the Sun’s position in the sky. Local apparent noon is when the Sun crosses the local meridian. However, the variability in the Earth's orbit and rotation means that apparent solar days aren't consistent in length.

**Mean Solar Time**

To address the variability, mean solar time was developed. It is based on an idealized Earth's orbit, resulting in consistent 24-hour days. Mean solar time, thus, served as a foundation for civil timekeeping until the advent of atomic time.

# 2.1 Standardizing Time: Greenwich Mean Time (GMT)

Local solar time varies with longitude, causing issues in coordination, especially with the advent of railways. This led to the standardization of time using Greenwich Mean Time (GMT), which became the international time standard in the late 19th century.

**Universal Time (UT) and the UT Second**

GMT was succeeded by Universal Time (UT) in 1928 to standardize the beginning of the day at midnight. UT is based on mean solar time at the Greenwich meridian, measured from stars rather than the Sun to account for the Sun’s disc width. However, the Earth's rotation is not perfectly stable, causing variations in the UT day length.

**Refining UT**

To address these variations, UT was refined into versions like UT0, UT1, and UT2. UT1 corrects for polar motion, providing a time consistent across the globe. UT2 further smooths out seasonal fluctuations.

**Transition to Atomic Time**

In the mid-20th century, atomic clocks, more stable than Earth’s rotation, were developed. This led to the creation of International Atomic Time (TAI). TAI is very stable, with its seconds based on atomic transitions. However, purely atomic timekeeping would eventually drift from the day-night cycle of UT1.

# 2.2 Coordinated Universal Time (UTC)

UTC, today's global time scale, combines atomic time (TAI) with adjustments to align with UT1. Leap seconds are occasionally added to UTC to keep it within 0.9 seconds of UT1, ensuring synchronization with the Earth’s rotation.

**Leap Seconds**

Leap seconds are introduced to account for the Earth's rotational variations. Since 1972, 27 leap seconds have been added to UTC. The necessity of leap seconds and their implementation remains a subject of debate.

**Generation of UTC**

UTC is generated through collaboration among national timing institutes, coordinated by the International Bureau of Weights and Measures (BIPM). These institutes maintain their versions of UTC, known as UTC(k), and contribute to the global UTC through a network of atomic clocks and primary frequency standards.

**National Time Scales**

National time scales, like UTC(NPL) in the UK, are physical realizations of UTC. These scales are maintained using atomic clocks and frequency standards, which are regularly compared and calibrated to ensure accuracy.

**Steering Clocks**

Steering clocks to maintain synchronization with UTC involves fine adjustments to the clock frequencies, balancing the need for accuracy and stability. This process is complex due to the inherent noise and drift in clock frequencies.

**Conclusion**

In summary, modern timekeeping relies on a blend of atomic and astronomical methods to maintain accurate and stable time scales. Coordinated Universal Time (UTC) exemplifies this integration, ensuring global synchronization while accommodating the Earth's rotational variations. Understanding the evolution and generation of time scales highlights the intricacies and precision involved in telling 'the time'.

# 3. Time and Frequency Dissemination

Time and frequency dissemination involves the precise distribution of time and frequency information from a reference source to users across various locations. This is essential for synchronization in fields like telecommunications, navigation, and financial transactions. Dissemination methods include satellite-based systems like GNSS, which provide one-way time transfer using highly accurate atomic clocks, and two-way satellite time and frequency transfer (TWSTFT) for enhanced accuracy. Networks using optical fibers also play a crucial role, offering direct and traceable time transfer with lower uncertainty. Calibration and traceability ensure the reliability and accuracy of these disseminated signals.

# 3.1 Time Dissemination

Accurate time dissemination is essential for a wide range of activities, including:

1. Communication Networks**:** Modern communication systems, such as the internet and mobile networks, rely on precise time synchronization to ensure data packets are transmitted and received in the correct order. This synchronization enables efficient data transfer and reduces delays.
2. Financial Transactions: Financial markets operate globally and rely on accurate time synchronization to ensure fair and transparent trading. Timestamps are crucial for tracking transactions and preventing fraud.
3. Transportation Systems: Air traffic control, rail networks, and maritime navigation require precise time synchronization to ensure safe and efficient operations.
4. Electric Power Distribution: Power grids rely on synchronized time to manage generation, distribution, and load balancing. Accurate timing prevents grid instability and blackouts.
5. Scientific Research: Many scientific experiments and measurements, such as particle accelerators and astronomy observatories, require precise time synchronization for data analysis and coordination.

**One-way Time Transfer**

The simplest method for disseminating time is one-way time transfer. In this approach, a timing signal is transmitted in a single direction from a source to a user. The user can then make corrections for the time delay using the provided information. While these corrections tend to be less accurate than those in two-way time transfer, one-way transfer is generally more cost-effective and sufficient for many applications.

**Two-way Time Transfer**

While one-way time transfer works well for many applications, two-way time transfer offers improved accuracy. In two-way time transfer, the process involves a source sending information to a user, who then sends a signal back to the source. The additional measurement data allows for more accurate calculation of the delay.

# 3.2 Time Dissemination over the Network

***Methods of Time Dissemination:***

*Several methods are employed to disseminate accurate time:*

**(a) Global Navigation Satellite Systems (GNSS)**

GNSS systems like GPS, GLONASS, and Galileo provide precise time signals that can be used for synchronization. These systems use atomic clocks aboard satellites to broadcast time information.  
Global Navigation Satellite Systems (GNSS) are particularly useful for one-way time transfer because their signals include encoded time information. Originally designed for location-finding and navigation, GNSS satellites contain highly stable atomic clocks, making them suitable for time transfer as well.  
However, GNSS is not the only method used for time transfer. Two-way satellite time and frequency transfer (TWSTFT) often employs geostationary satellites, providing an alternative approach.   
  
**GNSS Constellations:**

GNSS systems consist of several satellite constellations, the most notable being the United States’ GPS. Other active constellations include the EU’s Galileo, Russia’s GLONASS, and China’s BeiDou. GNSS systems are divided into three segments:

1. **Space Segment**: The space segment consists of the satellites orbiting Earth, each equipped with atomic clocks to broadcast precise time and positioning information to users.
2. **Control Segment**: The control segment comprises ground stations that monitor and manage the satellite constellation, ensuring accurate satellite orbits and clock settings.
3. **User Segment**: The user segment includes GNSS receivers used by individuals or systems to determine their exact position and time based on signals received from the satellites.

**(b) Network Time Protocol (NTP)**

NTP is a protocol used to synchronize the time of devices over computer networks. It corrects time discrepancies by querying reference time servers and adjusting the local clock accordingly.

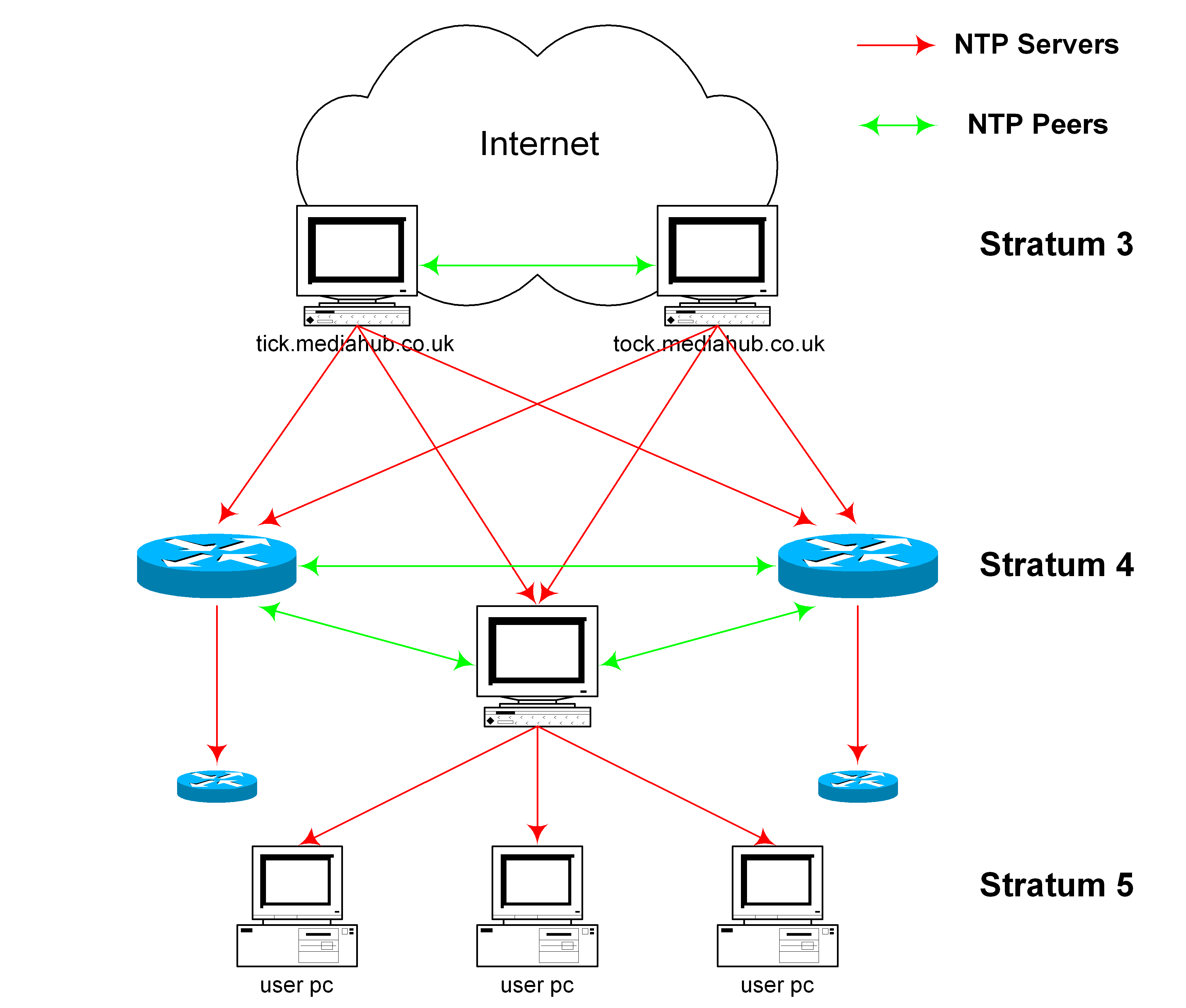
 

Figure 3.2.1 Network Time Protocol Figure 3.2.2 GPS Network Timeservers

The Network Time Protocol (NTP) packet is structured to synchronize the clocks of computers over a network. Here is a breakdown of its main fields:

1. LI (Leap Indicator): 2 bits
   * Indicates whether there is a leap second to be inserted or deleted in the last minute of the current day. Values:
     + 00: No warning
     + 01: Last minute of the day has 61 seconds
     + 10: Last minute of the day has 59 seconds
     + 11: Alarm condition (clock not synchronized)
2. VN (Version Number): 3 bits
   * Indicates the version number of the NTP. The current version is 4.
3. Mode: 3 bits
   * Indicates the mode of operation. Values:
     + 0: Reserved
     + 1: Symmetric active
     + 2: Symmetric passive
     + 3: Client
     + 4: Server
     + 5: Broadcast
     + 6: NTP control message
     + 7: Reserved for private use
4. Stratum: 8 bits
   * Indicates the stratum level of the local clock, with 0 being the highest level (primary reference) and increasing as the distance from the reference increases.
5. Poll Interval: 8 bits
   * Indicates the maximum interval between successive messages, expressed as a power of 2 in seconds.
6. Precision: 8 bits
   * Indicates the precision of the local clock, expressed as a power of 2 in seconds.
7. Root Delay: 32 bits
   * Total round-trip delay to the primary reference source, in seconds.
8. Root Dispersion: 32 bits
   * Total dispersion to the primary reference source, in seconds.
9. Reference Identifier: 32 bits
   * Identifier of the particular reference source.
10. Reference Timestamp: 64 bits
    * The time at which the local clock was last set or corrected, in NTP timestamp format.
11. Originate Timestamp: 64 bits
    * The time at which the request departed the client for the server, in NTP timestamp format.
12. Receive Timestamp: 64 bits
    * The time at which the request arrived at the server, in NTP timestamp format.
13. Transmit Timestamp: 64 bits
    * The time at which the reply departed the server for the client, in NTP timestamp format.
14. Key Identifier (optional): 32 bits
    * Used if NTP authentication is enabled, identifies the key used to compute the message authentication code (MAC).
15. Message Digest (optional): 128 bits
    * The message authentication code, used if NTP authentication is enabled.

This structure allows NTP to provide precise time synchronization and maintain accuracy across a network

**(c) Precision Time Protocol (PTP)**PTP is a protocol designed for high-precision time synchronization in industrial and telecommunications systems. It achieves sub-microsecond accuracy and is commonly used in applications like industrial automation and financial trading.

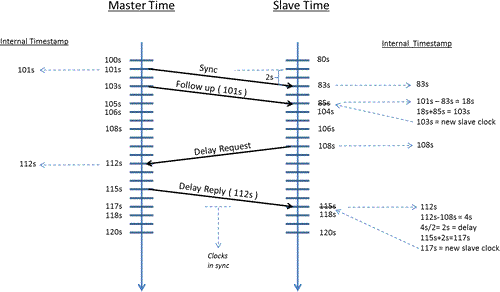
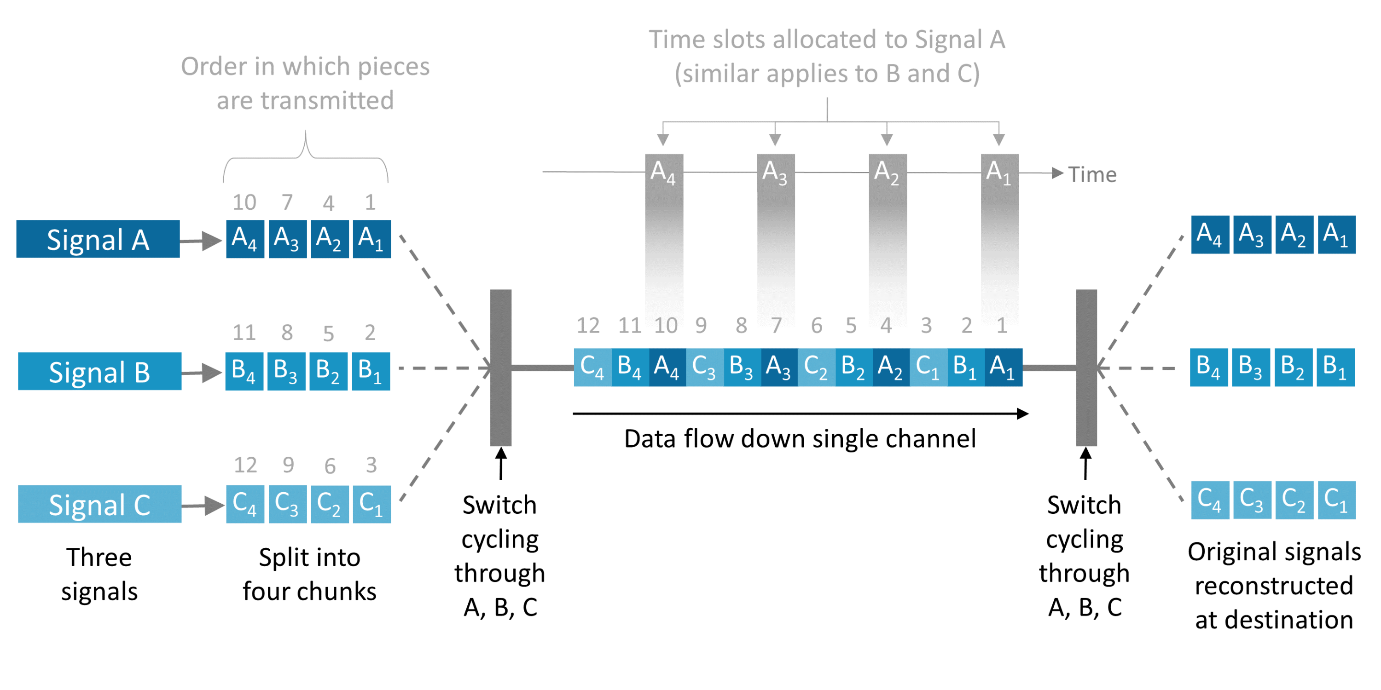
 

Figure 3.2.3 Precision Time Protocol Figure 3.2.4 PTP Delay and Offset

***3.2.1 Introduction to Network***  
  
**Packet Switching**

Packet switching is a method of breaking information into smaller chunks, or packets, allowing them to travel faster through a network. This is how information is transmitted over the internet.

**Multiplexing**

With billions of people using the internet simultaneously, many signals need to share the same cable or fiber. Multiplexing allows multiple signals to be transmitted over a single channel.  
  
  
  
 Figure 3.2.5 Multiplexing[1]  
 **Time Division Multiplexing (TDM):**

TDM divides the transmission time among different signals. For example, if you have signals A, B, and C, you can interleave their chunks by allocating specific time slots for each signal. If signal A is divided into four chunks (A1, A2, A3, A4) and similarly for signals B and C, the transmission stream might look like A1 B1 C1 A2 B2 C2 A3 B3 C3 A4 B4 C4. The receiving equipment must be synchronized with the splitting process to separate the signals back into A, B, and C.

**Frequency Division Multiplexing (FDM):**

Alternatively, FDM allocates different frequencies to different signals, allowing them to be transmitted simultaneously. In optical fibers, this is called wavelength division multiplexing (WDM). The receiving equipment must be capable of separating the frequencies (or wavelengths) back into the original signals.

# 3.3 Traceability in Timing

Traceability refers to a sequence of calibrations against increasingly accurate standards. A time and frequency measurement is traceable when it can be linked to a primary reference through an unbroken, documented chain of calibrations.

**The Importance of Traceability**

No measurement, including those of time and frequency, can ever be perfectly accurate. This inherent uncertainty is known as measurement uncertainty, which is a critical property of all measurement results. The goal is to quantify this uncertainty to achieve the necessary accuracy for specific applications.

**Calibration**

Calibration involves comparing an instrument or reference against a more accurate one and applying necessary corrections to ensure accuracy. In time and frequency, calibrations compare devices to UTC (Coordinated Universal Time) to measure time accuracy and quantify associated uncertainty. Regular calibration of timing equipment is essential for precision timing.

**Traceability to UTC(k)**

Traceability can be achieved through various methods of time and frequency dissemination, though some methods are more direct than others.

**Traceable Time via Satellite:**

GNSS signals are based on the atomic clocks onboard satellites and are aligned to UTC through corrections in the satellite navigation message. High-accuracy time transfer by GNSS requires calibrations to correct for processing delays.

**Traceable Time via Network:**

Networks can send time directly from national UTC(k) systems via optical fiber. Networks are easier to monitor and measure, offering a simpler, more direct traceability chain with less uncertainty.

**An Example of Traceable Time**

Some industries, like finance, require proof of traceability to meet regulations. The European Commission’s MiFID II regulations mandate that financial institutions synchronize time-stamping clocks to UTC, use timing within certain limits from UTC, and prove traceability to UTC. Precision timing services like NPLTime® help meet these regulations, as illustrated by the following traceability chain diagram.

By using services like NPLTime®, financial institutions can comply with regulations and access traceable time, ensuring that their time-stamping processes are accurate and reliable.

**Space Segment**: The space segment consists of the satellites orbiting Earth, each equipped with atomic clocks to broadcast precise time and positioning information to users.

**Control Segment**: The control segment comprises ground stations that monitor and manage the satellite constellation, ensuring accurate satellite orbits and clock settings.

**User Segment**: The user segment includes GNSS receivers used by individuals or systems to determine their exact position and time based on signals received from the satellites.

Applications and Challenges:

1. Security: Any vulnerabilities in time dissemination methods can be exploited by malicious actors.
2. Reliability: Dissemination methods must ensure continuous and reliable time synchronization. Network disruptions, signal interference, or technical failures can impact synchronization accuracy.
3. Diverse Environments**:** Different applications have varying requirements for time accuracy. While some applications can tolerate millisecond-level discrepancies, others require microsecond or even nanosecond precision.

Future Considerations:

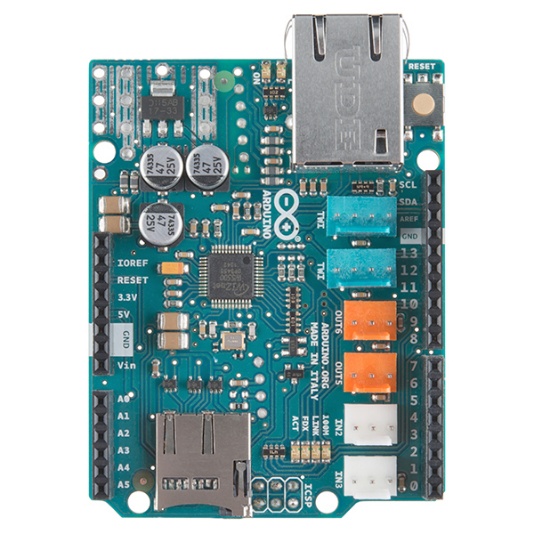
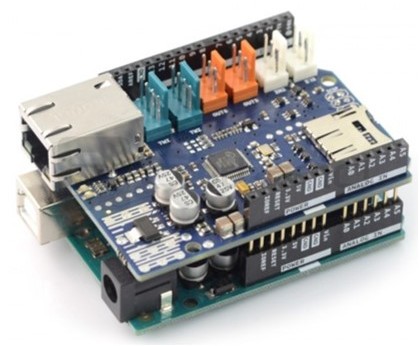
As technology evolves, time dissemination methods are likely to advance as well:

1. Quantum Timekeeping: The use of quantum technologies, such as atomic clocks based on quantum principles, could lead to even more accurate timekeeping and dissemination.
2. 5G and Beyond: Next-generation communication networks, like 5G and beyond, will require even tighter time synchronization for applications such as autonomous vehicles and industrial automation.
3. Resilience: Future time dissemination systems should be designed with increased resilience to cyberattacks and signal disruptions.

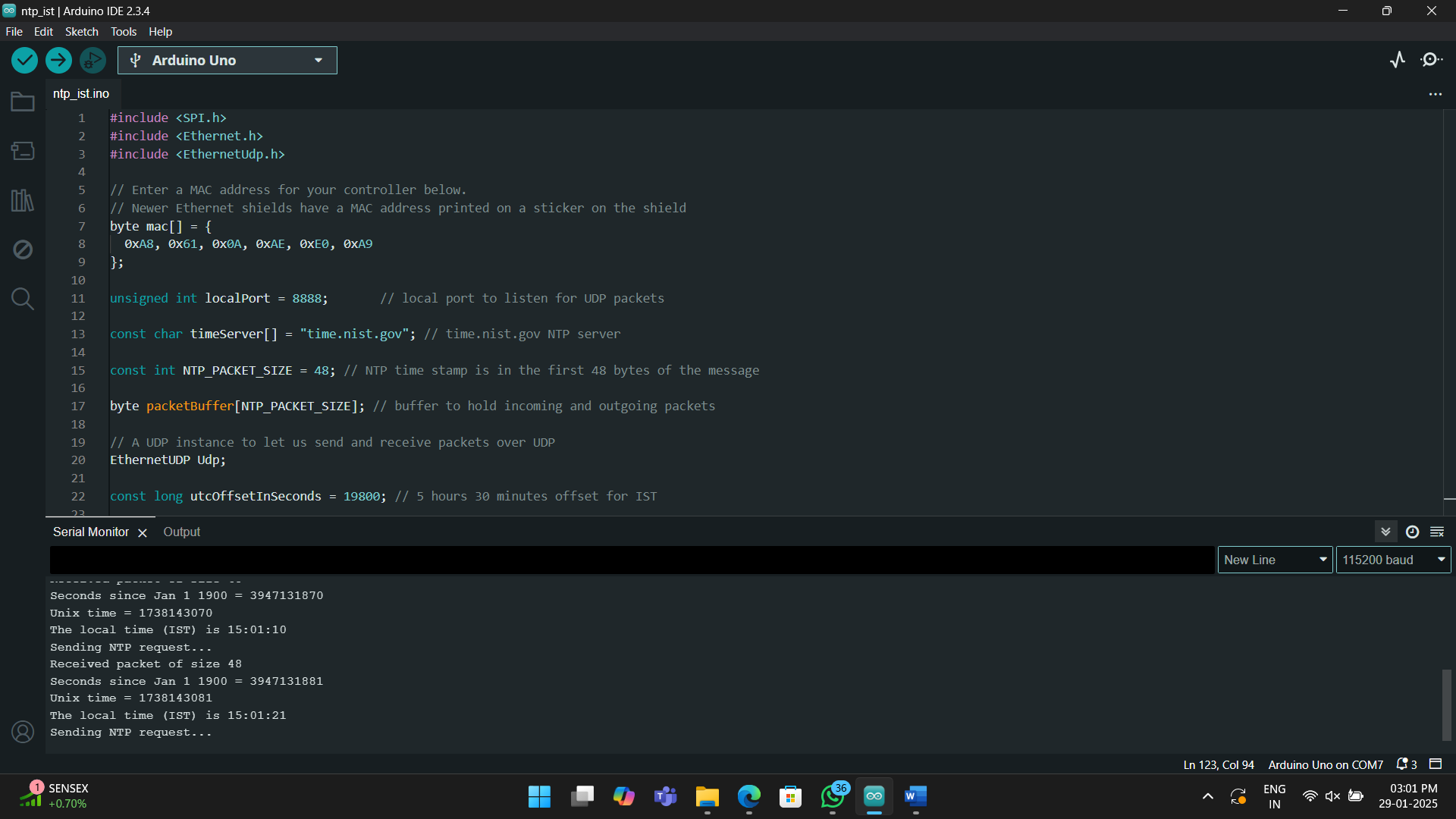
**Conclusion**

Time dissemination plays a pivotal role in modern society, enabling various critical applications to function accurately and efficiently. The methods used for time dissemination continue to evolve, driven by the increasing demands of technology and the need for ever-greater precision. As we move forward, ensuring accurate and reliable time synchronization will remain a fundamental challenge and opportunity for technological advancement.

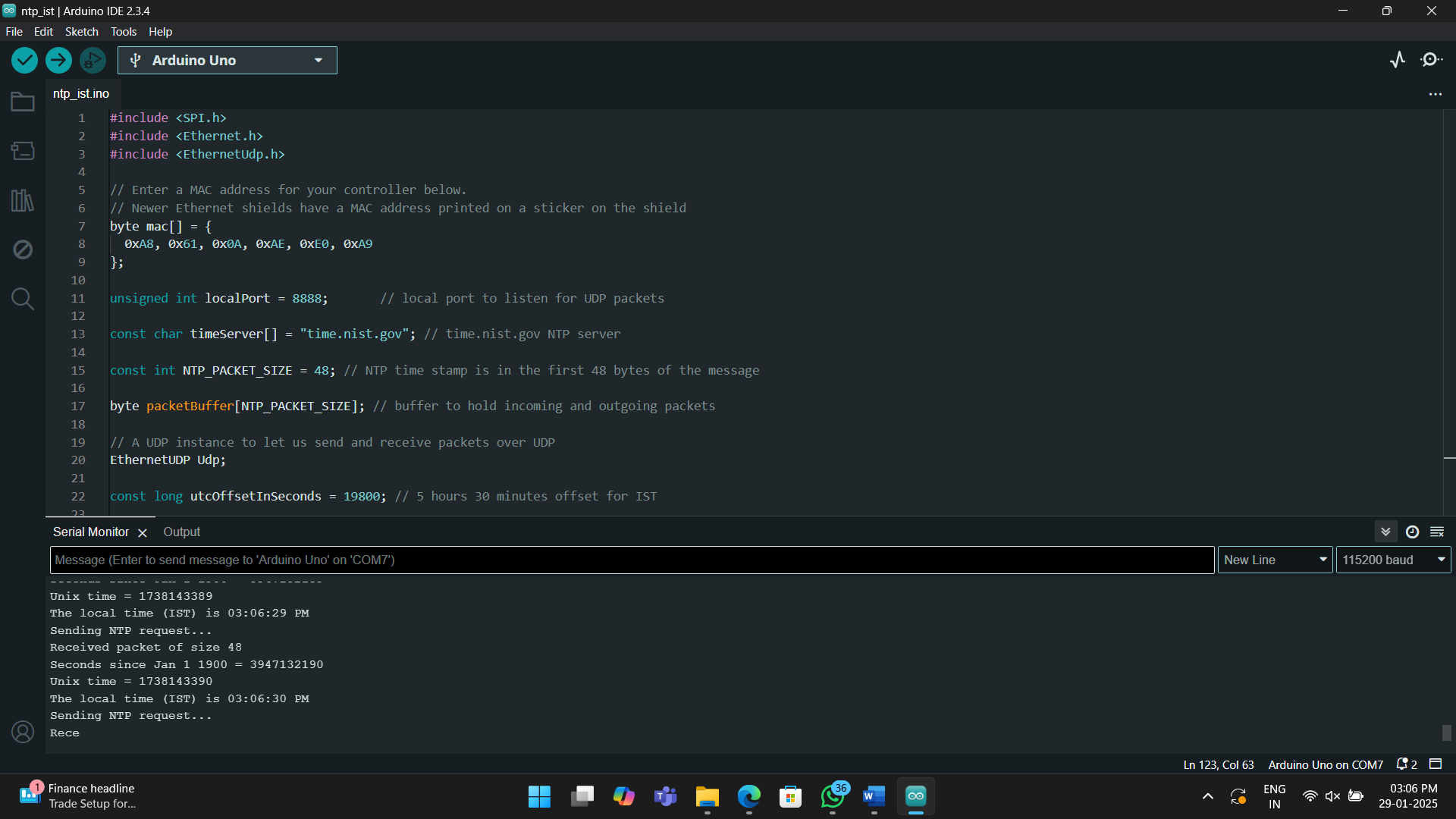
# Fetching and Displaying IST Time Using NTP Server on Arduino with Ethernet Shield

 Arduino Ethernet Shield Arduino with Arduino Ethernet Shield

The provided code is an Arduino sketch that fetches time from a Network Time Protocol (NTP) server and displays it on the serial monitor, specifically converting it to IST (India Standard Time).



The code is an Arduino sketch for obtaining the current time from a Network Time Protocol (NTP) server and displaying it on the Serial Monitor in a 12-hour format with AM/PM notation. (with 0 delay)



# 5. Introduction to NTP and Chrony

# 5.1 What is NTP?

NTP or Network Time Protocol is being well described under heading “Time dissemination over networks” earlier. So, let us jump on to understanding what exactly Chrony is.

# 5.2 What is Chrony?

Chrony was created by Richard Curnow. It is an open-source project designed to provide improved performance and features compared to traditional NTP implementations like ntpd. The project is actively maintained and has contributions from a community of developers to keep it up-to-date with modern time synchronization requirements.

Chrony is an implementation of the Network Time Protocol (NTP) designed for modern systems, providing a versatile solution for time synchronization on various network environments. Here are some key features and aspects of Chrony:

**1.** **Time Synchronization:** Chrony can synchronize the system clock with NTP servers, reference clocks (e.g., GPS receivers), or manual input.

**2. Versatility:** It is suitable for various environments, including desktops, laptops, embedded systems, and servers, providing accurate timekeeping whether the system has a continuous network connection or not.

**3. Fast Convergence:** Chrony is designed to quickly synchronize the system clock. It can handle large initial offsets and frequency errors, making it ideal for systems that start with incorrect time or are powered on and off frequently.

**4. Accuracy:** It can achieve high accuracy with sub-millisecond precision in typical network environments and better accuracy when using hardware timestamping or a reference clock.

**5. Dynamic Behavior**: Chrony can handle network disruptions gracefully, adjusting the system clock smoothly as conditions change.

**6. Low Resource Consumption:** It is lightweight and efficient, consuming minimal system resources, making it suitable for resource-constrained environments.

**7. Robustness:** It can correct for network latency and asymmetric delays, ensuring reliable time synchronization even in less-than-ideal network conditions.

**8. Flexibility:** It supports various configuration options and commands for customization, monitoring, and management.

# 5.3 Components of Chrony

- **Chronyd:** The daemon that runs in the background, performing time synchronization tasks.

**- Chronyc:** A command-line interface for managing and monitoring chronyd.

**Use Cases:**

- **Servers and Data Centers:** Ensuring accurate timekeeping for logging, security protocols, and scheduled tasks.

- **Embedded Systems:** Providing time synchronization for IoT devices and other embedded applications.

- **Desktops and Laptops:** Maintaining accurate time even with intermittent network connections.

# 5.4 Difference between Chrony and NTP

Chrony and NTP (Network Time Protocol daemon) are both software implementations used to synchronize the system clock with network time servers, but they have different features and performance characteristics. Here are the main differences between Chrony and NTP:

**Chrony:**

**1. Quick Synchronization:**

- Chrony can synchronize the system clock much faster than NTP, making it suitable for systems that are frequently rebooted or do not run continuously.

**2. Handling Network Conditions:**

- Chrony is better at handling unstable or intermittent network connections. It can maintain accurate time even when the network is not always available.

**3. Clock Discipline:**

- It uses a more sophisticated algorithm to discipline the system clock, which can result in better accuracy and stability.

**4. Dynamic Behavior:**

- Chrony adjusts quickly to changes in the system clock frequency, making it ideal for virtual machines and laptops that may undergo varying loads and temperature changes.

**5. Low Resource Consumption:**

- Chrony is lightweight and consumes fewer system resources compared to NTP, which is beneficial for embedded systems and devices with limited resources.

**6. Offline Operation:**

- Chrony can maintain accurate time even when not connected to any NTP servers, using an estimate of the drift rate of the system clock.

**7. Configuration:**

- Configuration in Chrony is often simpler and more flexible, with fewer lines of code needed to achieve similar functionality.

**NTP (ntpd):**

**1. Established and Widely Used:**

- NTP has been around longer and is more widely used, especially in traditional server environments.

**2. Stability and Compatibility:**

- It is considered stable and has broad compatibility with various systems and hardware.

**3. Precision and Accuracy:**

- NTP can achieve high precision and accuracy, suitable for environments where exact timekeeping is critical.

**4. Support for Reference Clocks:**

- NTP has extensive support for various hardware reference clocks, which can be critical in environments requiring highly precise time sources.

**5. Network Robustness:**

- While NTP also handles network disruptions, its algorithms are not as aggressive as Chrony's in recovering from them.

**6. Time Source Filtering:**

- NTP includes algorithms for filtering and selecting the best time sources from a pool of servers, which can improve accuracy in environments with multiple time servers.

**Summary:**

- Chrony is typically better suited for environments where quick synchronization, handling of intermittent connections, and low resource consumption are important. It's especially good for virtual machines, laptops, and embedded systems. NTP (ntpd) is often preferred in environments that require stable, long-term timekeeping with high precision and accuracy, such as traditional servers and systems with access to hardware reference clocks.

Both tools aim to achieve accurate time synchronization, but the choice between them depends on the specific needs and constraints of the environment in which they are used.

# 6. Raspberry Pi as NTP Chrony Server

# 6.1 Procedure

To turn a Raspberry Pi into a Stratum 2 NTP server using Chrony, follow these steps:

**Prerequisites:**

- A Raspberry Pi with a suitable OS installed (e.g., Raspberry Pi OS).

- Internet connectivity.

**1. Updating and Upgrading Raspberry Pi:**

sudo apt update

sudo apt upgrade -y

**2. Installation of Chrony:**

sudo apt install chrony -y

**3. Configure Chrony:**

Edit the Chrony configuration file:

sudo nano /etc/chrony/chrony.conf

Adding or modifying the following lines:

# pool 2.debian.pool.ntp.org iburst

server 0.pool.ntp.org iburst

server 1.pool.ntp.org iburst

server 2.pool.ntp.org iburst

server 3.pool.ntp.org iburst

# Use time sources from DHCP.

sourcedir /run/chrony-dhcp

#Internet Protocol address of NPL India Time server

# Allow NTP client access from local network.

allow 192.168.0.0/16

# Local network IP range

# Serve time even if not synchronized to any NTP server.

local stratum 3

# Specify the key file containing the NTP keys and the key ID.

keyfile /etc/chrony/chrony.keys

To save and close the file -> (Ctrl+X, Y, Enter).

**4. Enable and Start Chrony Service:**

sudo systemctl enable chrony

sudo systemctl start chrony

**5. Verify Chrony Configuration:**

Checking the status of the Chrony service to ensure it's running correctly:

sudo systemctl status chrony

Monitoring and Management and Sync-verification of Chrony:

To manage and monitor NTP server, usage chronyc, the command-line interface for Chrony.

Some useful commands include:

- Check sources:

chronyc sources

- Check tracking:

chronyc tracking

- Check server statistics:

chronyc sourcestats

# 6.2. Configure Clients to Use the Raspberry Pi NTP Server:

On the client machines, it is needed to configure them acccordingly so that they can use Raspberry Pi as their NTP server. This typically involves editing the NTP configuration file (/etc/ntp.conf or equivalent) and adding the IP address of your Raspberry Pi.

Example for a Linux client:

server <Raspberry\_Pi\_IP\_address> iburst

For windows client, we first need to,

1. Open Control Panel
2. Select Network and Internet
3. Choose clock and region
4. Go to date and time
5. Further select Internet time
6. Change its settings to add Raspberry Pi server’s IP
7. Click on Update Now
8. Select Ok

Following these steps will set up Raspberry Pi as a Stratum 2 NTP chrony server.

**6.2 Screenshots of the work done**

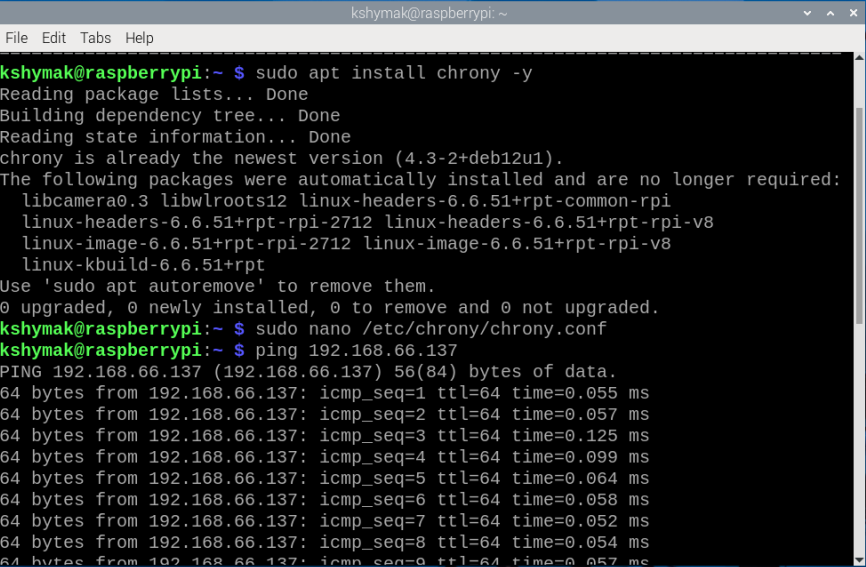
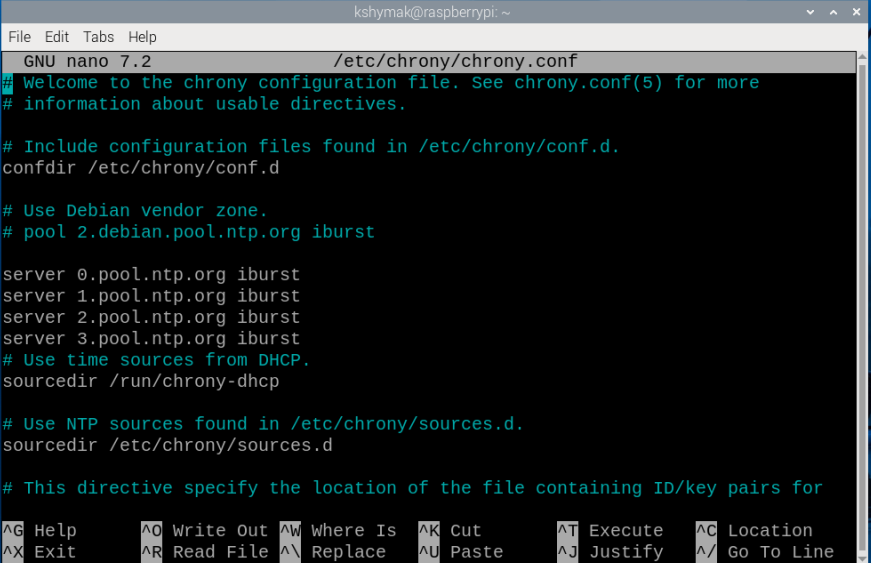
****

Figure 6.2.1 Configure Chrony

The main configuration file for Chrony is typically located at /etc/chrony/chrony.conf. This file contains various directives that control how Chrony operates, including which NTP servers to use, how often to poll these servers, and other behavior.

  
  
 Figure 6.2.2 Editing the Chrony Configuration File

Enabling a service in Linux ensures that it starts automatically when the system boots. This is managed by the system's in it system, typically systemd in modern Linux distributions.

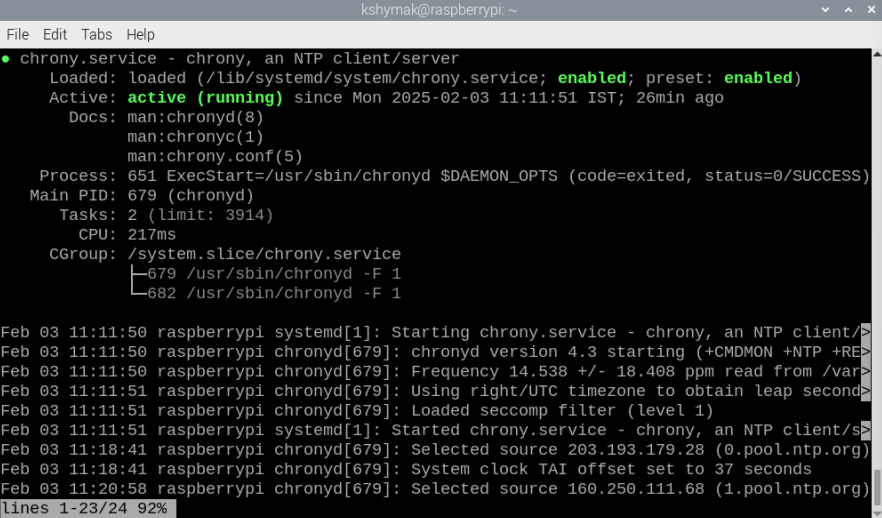


Figure 6.2.5 Starting of Chrony services

Checking the status of the Chrony service ensures time synchronization is functioning correctly, allowing administrators to verify the Chrony daemon is running, diagnose issues.

The status check is performed using the systemctl command in conjunction with the chronyc command-line interface.

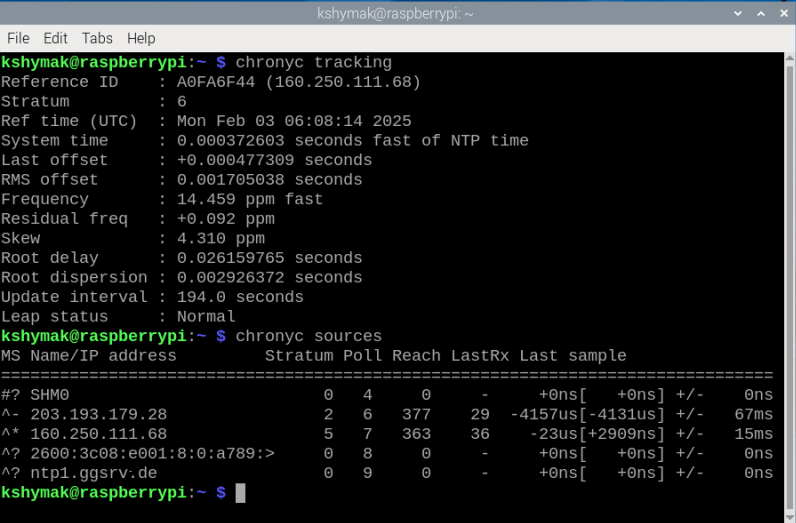


Figure 6.2.7 Monitoring and Management and sync-verification of Chrony

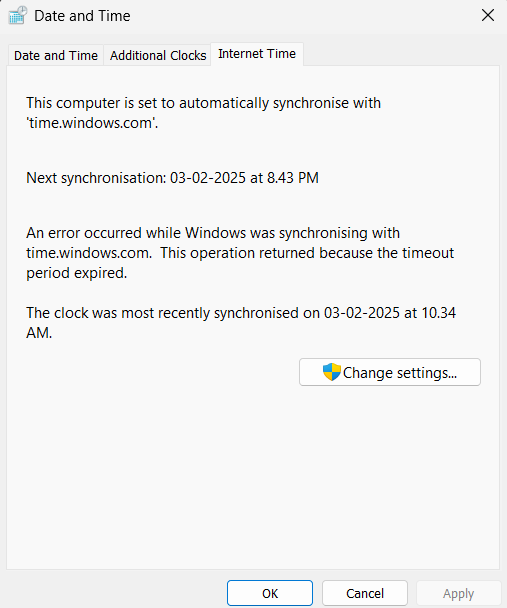


Figure 6.2.8 Configuring Internet Time of the client with Raspberry Pi NTP Server

Windows client to synchronize with a Raspberry Pi NTP server, open the Control Panel and navigate to Date and Time setting.

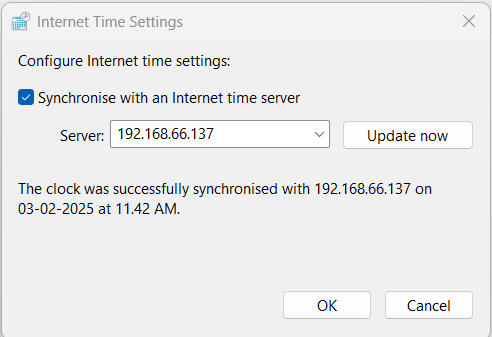


Figure 6.2.9 Adding the Raspberry Pi NTP Server Ip to the client machine

On each client machine, We need to configure the NTP client settings to use the Raspberry Pi's IP address as the time server. This ensures that the client's clock synchronizes with the Raspberry Pi, maintaining accurate time across all devices.

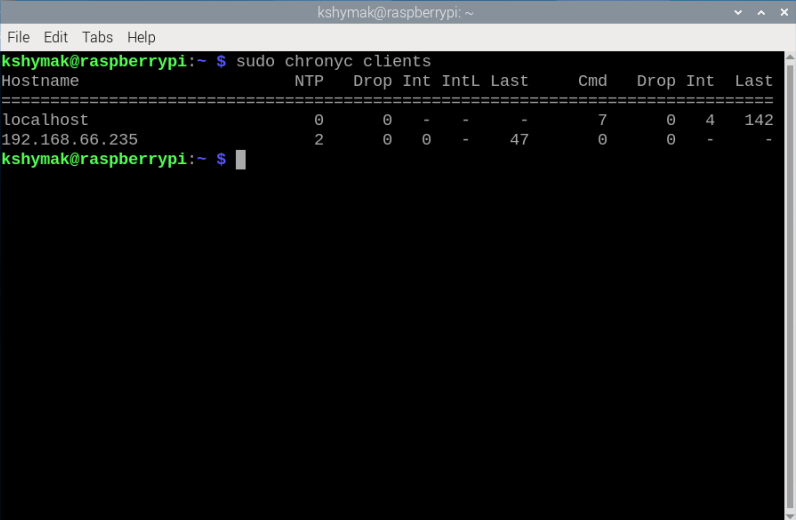


Figure 6.2.10 Monitoring of the clients

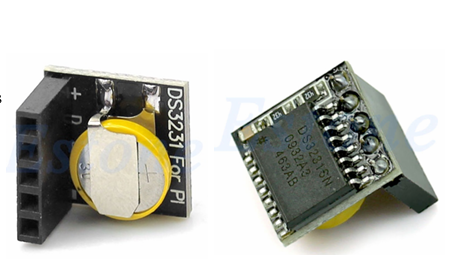
We check the number of clients connected to an NTP server using chronyc, start by opening a terminal on the server where chronyd is running. Execute the command chronyc clients to display a list of clients that have queried the server, including their IP addresses and connection status. By reviewing the output of this command, you can determine how many clients are currently connected to your Chrony NTP server and monitor their connection details.

***6. Conclusion***In conclusion, the study of time principles and advancements in timekeeping **underscores the deep** connection between our conceptual understanding of time and the technologies developed to measure and distribute it. From the early mechanical clocks that tracked celestial movements to the sophisticated timekeeping systems of today, each stage of development has enhanced our ability to measure time accurately. The creation of standardized time scales, such as Coordinated Universal Time (UTC), has been crucial in synchronizing global activities and ensuring consistency across various fields.

Modern time dissemination technologies, including the Network Time Protocol (NTP) and Chrony, have revolutionized the way time is distributed across networks. NTP synchronizes clocks on devices over a network, maintaining accuracy and consistency crucial for digital systems and communications. Chrony further refines this by offering enhanced precision, particularly in challenging environments.

The practical application of NTP on devices like the Raspberry Pi highlights how these advanced timekeeping principles can be implemented in accessible and versatile ways. Configuring a Raspberry Pi as an NTP server allows for reliable local time synchronization, demonstrating the practical utility of modern timekeeping technologies.

# Integrating DS3231 Real-Time Clock (RTC) with Raspberry Pi and Chrony for Time Synchronization



DS3231 Real-Time Clock (RTC)

1: Connect the DS3231 to the Raspberry Pi using I2C:

DS3231 VCC - Raspberry Pi 3.3V (Pin 1)

DS3231 GND - Raspberry Pi GND (Pin 9)

DS3231 SDA - Raspberry Pi SDA (Pin 3)

DS3231 SCL - Raspberry Pi SCL (Pin 5)

2: Enable I2C on the Raspberry Pi

1. Open a terminal window on the Raspberry Pi.
2. Run the following command to open the Raspberry Pi configuration menu:

sudo raspi-config

1. Navigate to Interfacing Options - I2C and select Enable.
2. Reboot the Raspberry Pi by running:

sudo reboot

3: Install I2C Tools and Verify the DS3231

1. sudo apt-get update
2. sudo apt-get install i2c-tools
3. sudo i2cdetect -y 1

4: Install and Configure the RTC Driver

1. sudo nano /boot/config.txt
2. Add the following line to the end of the file:

dtoverlay=i2c-rtc,ds3231

1. Save and exit the editor (press CTRL + X, then Y, and press Enter).
2. sudo reboot

5: Check and Set the RTC

1. Check the hardware clock:

sudo hwclock -r

1. If the clock is not set, use the following command to set the time on the DS3231 from the system time:

sudo hwclock -w

This command writes the current system time to the RTC.

6: Install and Configure Chrony for Time Synchronization

1. sudo apt-get install chrony
2. sudo nano /etc/chrony/chrony.conf
3. refclock SHM 0 offset 0.0 delay 0.2 stratum 3
4. Save and exit the editor (press CTRL + X, then Y, and press Enter).
5. sudo systemctl restart chrony

7: Enable the RTC on Boot

1. To ensure the DS3231 RTC is used for timekeeping even when the system is rebooted, you should link the hardware clock to the system clock. Add this line to your /etc/rc.local file before exit 0:
2. sudo hwclock -s

Save and exit the editor (press CTRL + X, then Y, and press Enter).

8: Verify Time Sync

1. Check the hardware clock (it should match the system time):

sudo hwclock -r

1. Monitor Chrony’s synchronization status:

chronyc tracking

Conclusion

By following these steps, you've successfully integrated the DS3231 Real-Time Clock with your Raspberry Pi and configured Chrony for accurate time synchronization. The DS3231 ensures that your Raspberry Pi retains the correct time even when powered off, and Chrony guarantees that your system clock is always synchronized with NTP servers. This setup provides an accurate and reliable timekeeping solution for your Raspberry Pi, especially for applications requiring precise time tracking.

# Raspberry Pi into a Stratum 2 NTP server using Chrony with hardware time stamping

**1: Update System**

Ensure your system is up to date:

sudo apt update && sudo apt upgrade -y

**2: Install Chrony**

Chrony is a lightweight and robust NTP implementation:

sudo apt install chrony -y

**3: Enable Hardware Time Stamping**

To benefit from hardware timestamping, check if your network interface supports it:

ethtool -T eth0

Look for:

* **tx-hardware and rx-hardware timestamps** enabled

If supported, enable it:

sudo ethtool -K eth0 tx-udp\_tnl-segmentation on

Replace **eth0** with the actual interface name.

**4: Configure Chrony**

Edit the Chrony configuration file:

sudo nano /etc/chrony/chrony.conf

Modify it as follows:

# Use reliable Stratum 1 NTP servers

server time.google.com iburst

server time.cloudflare.com iburst

server ntp.ubuntu.com iburst

server pool.ntp.org iburst

# Allow clients in your network (modify CIDR block if needed)

allow 192.168.1.0/24

# Enable hardware timestamping

hwtimestamp eth0

# Increase min/max polling intervals for better stability

minpoll 4

maxpoll 6

# Log statistics for analysis

logdir /var/log/chrony

log measurements statistics tracking

Save and exit.

**5: Restart Chrony and Enable at Boot**

sudo systemctl restart chronyd

sudo systemctl enable chronyd

**6: Verify Stratum Level**

Check Chrony status:

chronyc tracking

Look for:

* **Reference ID** (should be a known Stratum 1 NTP source)
* **Stratum: 2**

To check synchronized peers:

chronyc sources -v

# Integration of u-blox ZED-F9T with Raspberry Pi 4 for a Stratum 1 NTP Server Using PPS Input

**1. Hardware Setup**

**Required Components:**

* **Raspberry Pi 4** (running Raspberry Pi OS)
* **u-blox ZED-F9T** (USB interface + PPS output)
* **Jumper Wires** (for PPS connection)
* **Power Supply** (stable 5V for RPi & ZED-F9T)

**Connect PPS to GPIO:**

1. **Locate the PPS Output on the ZED-F9T**
   * If using the u-blox EVK-F9T board:
     + PPS is on **Pin 3** of the 10-pin connector.
     + GND is on **Pin 4**.
   * If using a custom board, check the schematic.
2. **Wire PPS to Raspberry Pi GPIO**
   * Connect **PPS (3.3V logic) to GPIO18 (Pin 12)**
   * Connect **GND to GND (e.g., Pin 14)**

**2: Enable Serial & PPS Support on Raspberry Pi**

1. Open the terminal and edit the boot configuration:

sudo nano /boot/config.txt

1. Add the following lines at the end of the file:

# Enable UART for GPS

enable\_uart=1

# Enable PPS on GPIO18

dtoverlay=pps-gpio,gpiopin=18

1. Save the file (CTRL+X, Y, Enter).
2. Reboot the Raspberry Pi:

sudo reboot

**3: Verify GPS & PPS Devices**

1. **Check if GPS is detected** (via USB, usually /dev/ttyACM0):

ls /dev/ttyACM\*

If you see /dev/ttyACM0, the GPS module is recognized.

1. **Check if PPS is detected**:

ls /dev/pps\*

If you see /dev/pps0, the PPS signal is working.

**4: Install Required SoftwareInstall GPSD & Chrony**

sudo apt update

sudo apt install -y gpsd gpsd-clients chrony pps-tools

**Stop gpsd if running:**

sudo systemctl stop gpsd.socket gpsd

sudo systemctl disable gpsd.socket

**Modify GPSD Default Config:**

sudo nano /etc/default/gpsd

Change to:

DEVICES="/dev/ttyACM0 /dev/pps0"

GPSD\_OPTIONS="-n"

START\_DAEMON="true"

USBAUTO="true"

GPSD\_SOCKET="/var/run/gpsd.sock"

Save and exit.

**Start gpsd:**

sudo systemctl restart gpsd

sudo systemctl enable gpsd

**Check GPS Data:**

gpsmon

or

cgps -s

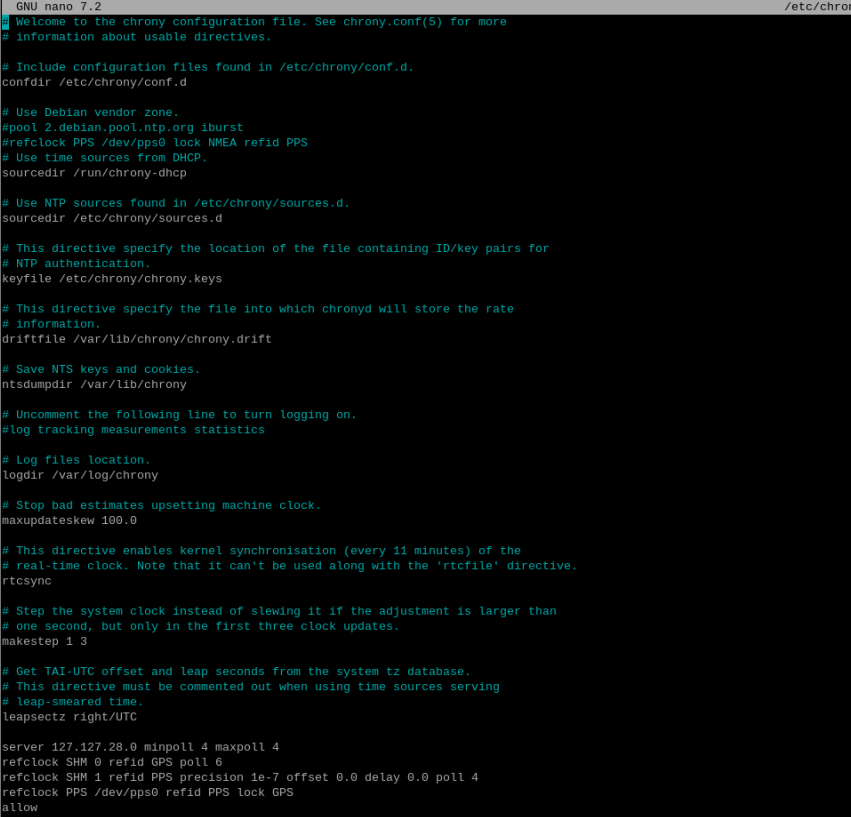
You should see GPS time and satellite data.

**5: Configure Chrony for NTP**

1. Open the Chrony configuration:

sudo nano /etc/chrony/chrony.conf

Add or modify these lines:



**6: Restart Chrony:**

sudo systemctl restart chronyd

sudo systemctl enable chronyd

**Verify PPS and GPS Sources in Chrony:**

chronyc sources -v

You should see:

* **GPS** marked as # (good signal)
* **PPS** marked as \* (chosen as time source)