

---

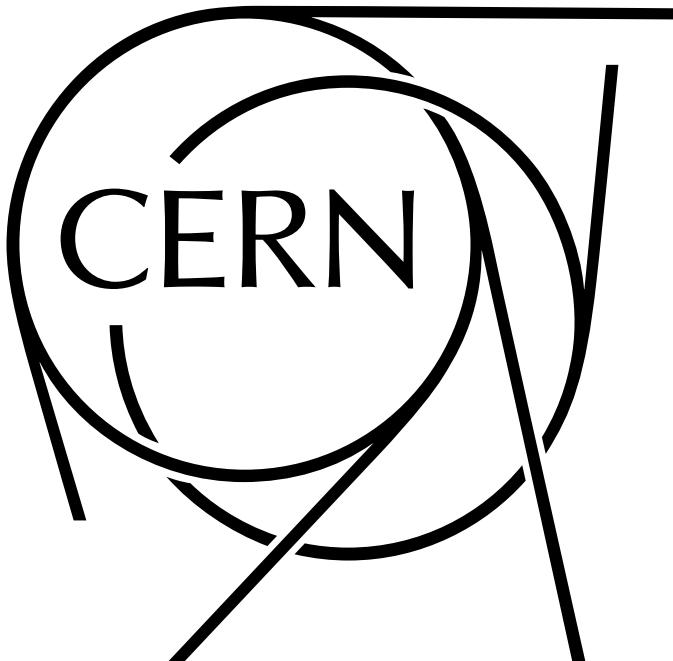
# QPS Data Acquisition

---

## The Synchronisation Sub-System

- Interim Report -

---



Project Report  
Group 802

Aalborg University  
Department of Electronic Systems  
Fredrik Bajers Vej 7B  
DK-9220 Aalborg

Copyright © Aalborg University 2020

This report was written in LaTeX using Overleaf licensed to all students at Aalborg University.



# AALBORG UNIVERSITY

## STUDENT REPORT

### Department of Electronic Systems

Fredrik Bajers Vej 7

DK-9220 Aalborg Ø

<http://es.aau.dk>

**Title:**

QPS Data Acquisition  
The Synchronisation Sub-System

**Theme:**

Internship

**Project Period:**

Autumn Semester 2020

**Project Group:**

Group 802

**Participant(s):**

Magnus Bøgh Borregaard Christensen

**Supervisor(s):**

Petar Popovski  
Anders Ellersgaard Kalør  
Tomasz Podzorny (CERN)

**Page Numbers:** 50

**Appendix Page Numbers:** 0

**Date of Completion:**

January 31, 2021

**Abstract:**

The MPE-EP section at CERN is developing a new data acquisition system. A key requirement of this system is the need for synchronising end nodes to CERN's centralised timing system. This report documents the progression of this aspect of the new data acquisition system in the timer interval of September 2020 to December 2020. In this phase, a main priority was to learn and build knowledge regarding the precision time protocol as well as other relevant synchronization and associated CERN specific technologies. Additionally, the report describes the creation of relevant support systems that will facilitate future development. Overall, progression is satisfactory and I am confident that we are in an appropriate position to initiate the next steps of development and conclude timely in June 2021.

The content of this report is freely available, but publication (with reference) requires an agreement with the author.

# Preface

I am currently participating in a one-year internship position at the European Organisation for Nuclear Research (CERN) in Geneva, Switzerland. Specifically, I am working in the Electronics for Protection (EP) section within the Machine Protection and Electrical Integrity (MPE) Group. The MPE-EP section is responsible for the development of the electronics and data acquisition systems used in the protection systems of the superconducting magnets used in CERN's particle accelerators [1].

This report documents the background, progress, and results of the work I have undertaken for the duration of my 9<sup>th</sup> Semester (from September to December 2020) as part of my internship in the MPE-EP section.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Background</b>	<b>3</b>
2.1	The Large Hadron Collider . . . . .	3
2.2	Quench Protection System . . . . .	5
2.3	QPS Data Acquisition System . . . . .	7
<b>3</b>	<b>Goals, Requirements, and Constraints</b>	<b>9</b>
3.1	CERN Timing Network . . . . .	11
3.2	Performance Requirements . . . . .	11
3.3	Existing Equipment and Constraints . . . . .	11
3.4	Network Topology . . . . .	12
<b>4</b>	<b>Precision Time Protocol</b>	<b>13</b>
4.1	Clock Model . . . . .	13
4.2	Packet Based Synchronization . . . . .	15
4.3	Synchronization Methods . . . . .	19
4.4	Minimizing Asymmetry and Time-Stamping Delay . . . . .	20
4.5	Clock Types and Clock Hierarchy . . . . .	22

## Contents

<b>5 Timing Network Topology and Hardware</b>	<b>25</b>
5.1 Topology Overview . . . . .	25
5.2 Front End Computer Hardware . . . . .	26
5.3 Switch . . . . .	28
<b>6 Data Acquisition Board Development</b>	<b>30</b>
6.1 Hardware Design . . . . .	30
6.2 Firmware . . . . .	31
<b>7 System Integration</b>	<b>36</b>
7.1 FEC and CTRIE Synchronisation . . . . .	36
7.2 PTP Application . . . . .	38
7.3 Synchronisation Measurement . . . . .	38
<b>8 Preliminary System Results</b>	<b>39</b>
8.1 Setup . . . . .	39
8.2 Synchronisation Accuracy between CTRIE and I210 NIC . . . . .	41
8.3 Synchronisation Accuracy between CTRIE and Data Acquisition Boards .	41
<b>9 Summary and Conclusion</b>	<b>45</b>
<b>10 Personal Reflection</b>	<b>46</b>
<b>Bibliography</b>	<b>48</b>

# Chapter 1

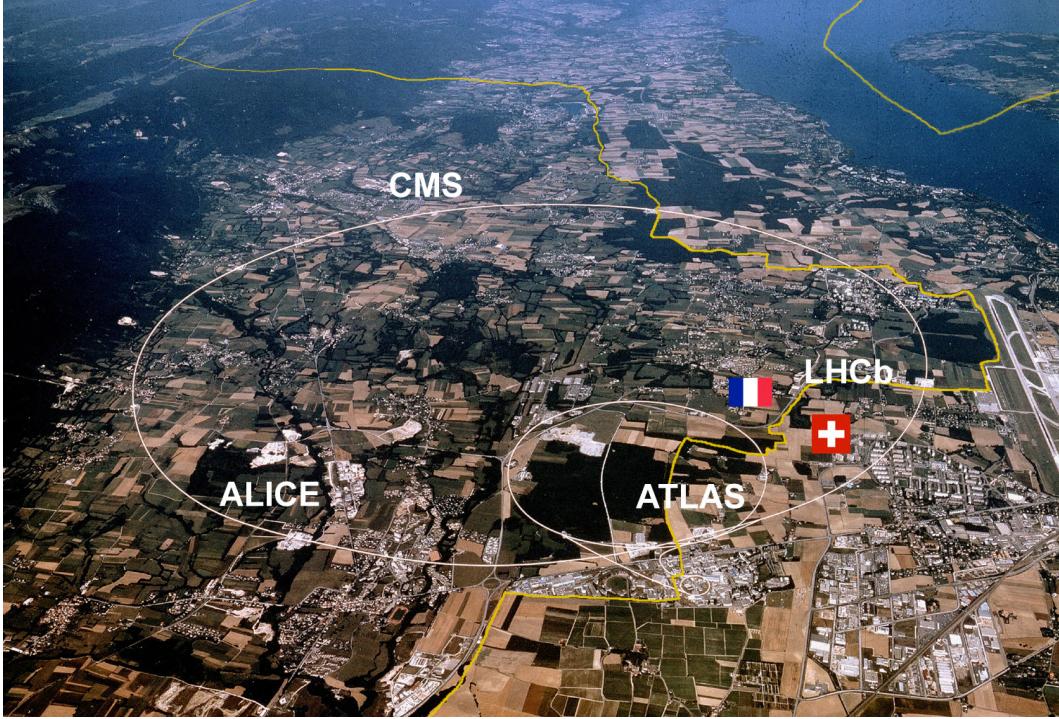
## Introduction

The European Organization for Nuclear Research, known as CERN (from *Conseil européen pour la recherche nucléaire*), is a research organisation known for operating one of the world's largest particle physics laboratory. The laboratories was established in 1954 with the goal of rebuilding the European scientific community in the aftermath of the second world war, uniting European scientists, and to allow countries to share the costs of costly nuclear physics facilities [2].

CERN has a long history of building particle colliders [3], and in 2008 this culminated with the completion of the Large Hadron Collider (LHC), the world's largest particle accelerator [4]. This modern engineering marvel measures 27 km in circumference and is located deep underground on the Franco Swiss border [4] as shown on figure 1.1. The unprecedented size of the LHC combined with modern advancements in electronics, magnets, and superconductors allows for the acceleration of particles to record high velocities, and therefore in extensions allows for higher energy particle collisions. These collisions produce subatomic debrief which helps provide clues about the nature of our universe.

At CERN I am taking part in the development of a new data acquisition system for the LHC's magnet protection system. This new data acquisition system shall comprise of end nodes located at each of the LHC magnets, as well as centralised servers that collects data from each of the nodes. My role in this undertaking focuses on the implementation of a synchronisation scheme for synchronising the end nodes with the centralised time used at CERN.

As of my arrival at CERN, the new data acquisition system had already been under development for a year, and multiple other student interns and staff members at CERN have contributed to it. These contributions include various decisions, developments, and creations, that both aid and constrain the scope of my tasks. Key of which include the decision to use the precision time protocol to perform synchronization, and the



**Figure 1.1:** Aerial Image of the Franco Swiss border region where CERN and the LHC is located [5]. The larger circle illustrates the location of the LHC, while the smaller circle is the represents the 7 km radius Super Proton Synchrotron (SPS) accelerator. The four written markings illustrate the location of the four main LHC experiments where particles are collided and analysed.

initial design and creation of a development board acting as the end nodes of the data acquisition system.

This report documents the first phase of the project. In this phase the goal was to implement rudimentary synchronization on the provided hardware, and setup a proper development environment to facilitate rapid and continuous testing, evaluation, and development of the system. Following this phase, the complete system is developed and implemented. This is to be presented in the final report.

The report is structured as follow. Additional background regarding CERN, the laboratory accelerators, and the broader context of my work is described in chapter 2. Based on the presented background chapter 3 proceeds to list the requirements and goals of the desired solution. Chapter 4 details the theory behind time synchronization. Following; chapter 5 describes the hardware and infrastructure provided by CERN that the solution must use and comply with. Additional software and hardware development are described in chapter 6. The integration of sub-systems is presented in chapter 7. Chapter 8 analyses the performance of the solution. Penultimately, chapter 9 briefly summarizes and concludes on the progression of the project. Finally, chapter 10 reflects on my time interning at CERN.

# Chapter 2

## Background

To provide context to the assigned project, this chapter describes the fundamental workings of particle accelerators. Furthermore, the current magnetic protection data acquisition system is detailed, and its limitations are highlighted to illuminate why a new solution is desired.

### 2.1 The Large Hadron Collider

The LHC is capable of accelerating protons to 99.999991% the speed of light [6]. For the uninitiated reader it might be hard to imagine how the various elements of the accelerator comes together to perform this feat. Therefore, the next section will provide a brief introduction to the workings of particle accelerators, and especially the LHC. This section intents to provide background and context to the system that will be detailed in this report.

#### 2.1.1 Workings of an Accelerator

In the LHC particles are accelerated using RF cavities [7]. When charged particles, such as protons, enter the oscillating electromagnetic field created by the cavity they are attracted to the field and thus accelerated [7].

Fundamentally, two distinct kinds of particle accelerators exist, linear and circular accelerators. A linear accelerator consists of one linear beam-path where particles accelerates along a straight tube and finally collides with a target. In linear accelerators the length of the accelerator imposes a hard limit on the velocity the particles can achieve [3].

## 2.1. The Large Hadron Collider

In contrast, the beam-path of a circular accelerator is shaped as a ring and the accelerated particles will therefore move along circular path. This allows the particles to remain within the accelerator indefinitely, thus allowing for more time (as opposed to linear accelerators) to accelerate the particles to higher velocities and in extension higher energy levels [3].

While a circular design allows for the acceleration of particles for longer, it imposes other constraints on the maximum particle velocity. When particles are accelerated by the RF cavities, they tend to remain in motion in a straight line, following from newtons first law of motion [8]. Thus, external force is required to steer the particles along the circular beam path. In circular accelerators this is achieved by using powerful magnets that bend the particle stream along the desired path [8]. In this scenario the maximum achievable velocity is limited by the magnets ability to maintain the curvature of the trajectory and counter the centripetal force of the moving particles <sup>1</sup> [8]. As the centripetal force increases with the particle velocity and decreases as the radius of the beam-path increases, the maximum velocity can be increased by either increasing the accelerator radius or using more powerful magnets [9]. In the LHC, the main electromagnets are super conducting and use 11 kA to produce magnetic fields up to 8.3 T [8].

### 2.1.2 CERN Accelerator Complex

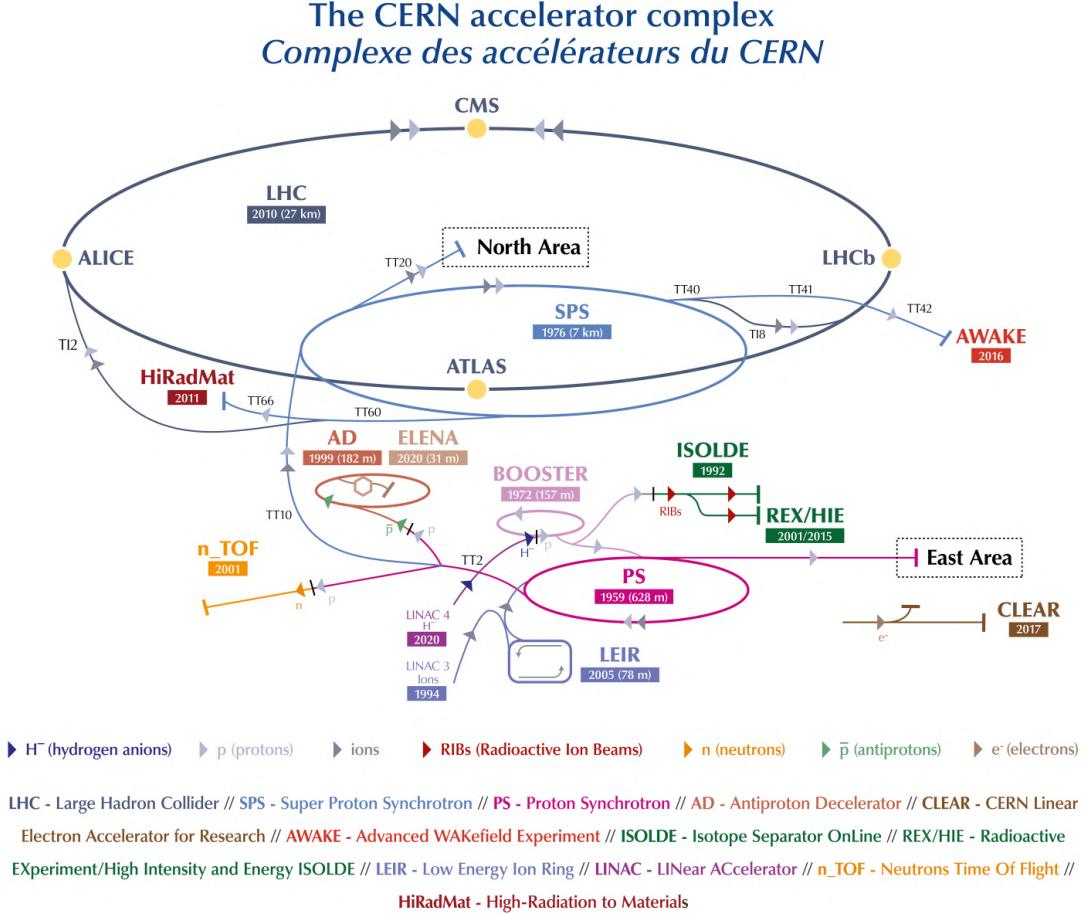
At CERN, particles do not begin their journey in the LHC, but are accelerated through a sequence of accelerators [10] as shown in figure 2.1. First, hydrogen gas ( $H^-$  ions) is injected into a linear accelerator, LINAC 4, and accelerated to 52% the speed of light [11]. Subsequently, the proton beam is injected into a series of circular accelerators [10]. First it enters the Proton Synchrotron Booster where the  $H^-$  ions are passed through an electric field to strip away the electrons and produce pure protons. These protons are then transferred to the Proton Synchrotron, and finally the Super Proton Synchrotron [10]. Each of the accelerators accelerates the particles to higher energy levels, before the particles are injected into the LHC [10]. In the LHC two beams of protons accelerate in opposite directions, to allow for double the energy release when colliding opposing moving particles [10]. The two beams of protons are continuously accelerated for 20 minutes, before they reach their maximum speed as permitted by the beam steering magnets [10].

Once the particles have reached their maximum velocity, the LHC starts colliding particles from the two opposing beams at various locations in the accelerator ring [12]. Particle detectors are placed around these collision points to measure the particles that appear from the collisions[12].

---

<sup>1</sup>The physics of this is significantly more complicated than the classically mechanics explanation provided here, as the high velocities necessitates considerations of relativistic effects

## 2.2. Quench Protection System



**Figure 2.1:** Illustration of the accelerator complex at CERN [13]. Note that in addition to the accelerators mentioned explicitly in the text, the illustration also shows the other acceleration and detection facilities at CERN.

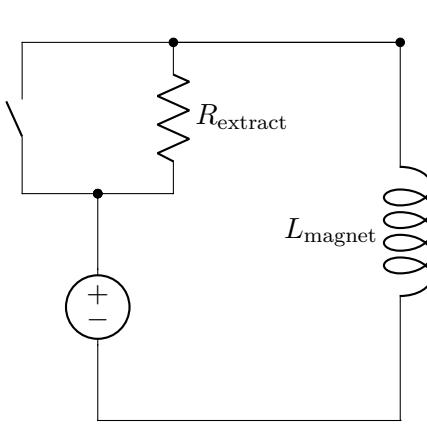
## 2.2 Quench Protection System

As mentioned previously, the LHC uses powerful superconducting magnets to steer and control the particle beam. To achieve superconductivity, these magnets are cooled to 1.9 K using liquid helium [14]. If any of a superconducting magnet's three key parameters limits (temperature, magnetic field, and current density) are exceeded, the part of the magnet experiencing the change will transition from a superconducting to a normal conducting state [15]. This phenomenon is known as a quench. Quenches usually occur in the LHC because of stray beam particles colliding with the superconducting magnets, thus dissipating heat within them [15].

## 2.2. Quench Protection System

When a quench occurs, the now normally conducting magnet no longer has zero ohmic resistance. This transition from a zero-resistance state transfers the energy stored in the magnet's magnetic field to the magnet's coil windings. In the worst case this can damage or even destroy the magnet [15].

To mitigate the risk of quenches causing irreversible damages, various passive and active safety mechanisms have been installed throughout the LHC as part of the Quench Protections Systems (QPS). These systems aim to detect the presence of a quench event, and then subsequently ensure that the magnetic energy is not dumped into the quench region [16]. One example of such a system involves the use of an energy extraction resistor, as shown in figure 2.2 and figure 2.3. In the illustrated system a quench must first be detected electronically using a quench detection system (QDS). This is usually done by continuously monitoring the voltage drop over the magnet [16]. Once a quench has been detected by the QDS, the normally closed-circuit breaker is opened, and the stored energy is dumped into the large resistor as opposed to the magnet winding [16].



**Figure 2.2:** Example of a QPS using an energy extraction resistor. When a quench is detected, the normally closed switch is opened [16].

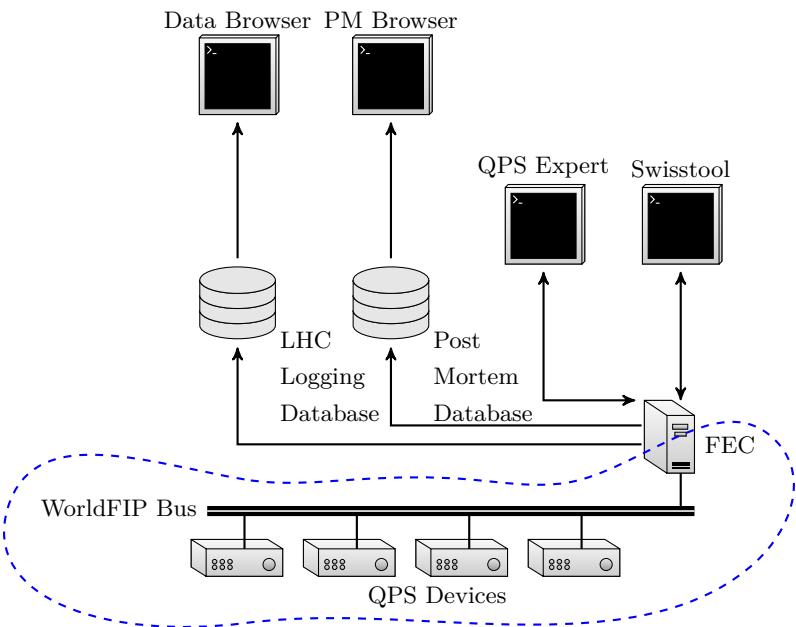


**Figure 2.3:** Picture of three stacked Large Energy Extraction Resistors used at CERN [17].

This exemplifies just one of several safety systems that protects against quenches. Unfortunately, despite these extensive measures' accidents may still occur. An accident occurred on the 19<sup>th</sup> of September 2008, where a magnetic quench event occurred [18]. During this event some of the safety systems failed to deploy properly, resulting in a helium leak that damaged multiple magnets [18], and caused \$30 million worth of damage in direct cost [19]. Following this incident, the QPS has been continuously updated, and while magnetic quenches continues to be a normal part of operating the LHC, they are now routinely dealt with without costly consequences.

### 2.3. QPS Data Acquisition System

To ease the maintenance and performance evaluation of the QPS it is desirable to monitor the systems current state and log its behaviour during quenching events. Currently this is done using a data acquisition setup like the one illustrated in figure 2.4.



**Figure 2.4:** Illustration of current data acquisition network configuration [20]. The QPS devices communicate with a Front-End Computer (FEC) through a WorldFIP fieldbus. This FEC then in turn relays data going to/from the QPS devices to various databases or control terminals. The blue encirclement highlights the systems that are addressed in this report, i.e. the communication between the FEC and the QPS devices.

As shown, the devices in the QPS communicate with a Front-End Computer (FEC) through a WorldFIP field bus protocol running over a single pair of twisted cables. The data collected by the FEC is logged in databases for future analysis.

The use of a field bus to facilitates communication between the FEC and the QPS devices has its own distinct set of advantages. First, the traffic over the protocol is deterministic, thus guaranteeing bounded latency [21]. Additionally, the protocol allows for multi node connections over a single pair of twisted cables, thereby significantly reducing the cabling needed in the tunnels <sup>2</sup> [21]. Most importantly though, the implementation has been verified as being radiation tolerant to the environment found in the LHC tunnels [21].

---

<sup>2</sup>despite this a considerable amount of cabling is still required, in total over 450 km of WorldFIP cables are installed at CERN [22]

### 2.3. QPS Data Acquisition System

However, the current implementation is not without key limitations. Firstly, the bandwidth of each device is limited to 1 kbit/s. Furthermore, field bus solutions are becoming increasingly rare in industry, thus making it difficult to source the hardware required to maintain the current infrastructure. In fact, the original hardware is no longer supported by the original manufacturer, and CERN has been forced to develop their own WorldFIP hardware to continue operation and upkeep of the current infrastructure [23]. Finally, the deployment flexibility is limited, as each WorldFIP cable must be connected to a dedicated FEC and it is not possible to route WorldFIP data through a network of routers and switches [21].

The MPE-EP wishes to modernize the data acquisition system to alleviate the previously listed disadvantages of WorldFIP. During previous intern assignments feasibility studies were undertaken [20], and it was concluded in the EP section, that an Ethernet solution should be pursued. The maturity and standardization of Ethernet ensures that the technology will be supported for years to come, and that no specialised hardware is required in the infrastructure. Ethernet also allows for the use of standard networking equipment, such as switches, to support more flexible networking solutions. Additionally, the bandwidth of modern Ethernet is also much higher than that of WorldFIP, with Fast Ethernet allowing for speeds of up to 100 Mbit/s. Finally, while Ethernet is not usually thought of as partaking in a daisy chaining topology (but rather in a star topology network), modern standards, such as 10BASE-T1S [24] allows for daisy chained/bus topology Ethernet over a single pair of twisted cables, while facilitating bandwidths up to 10 Mbit/s.

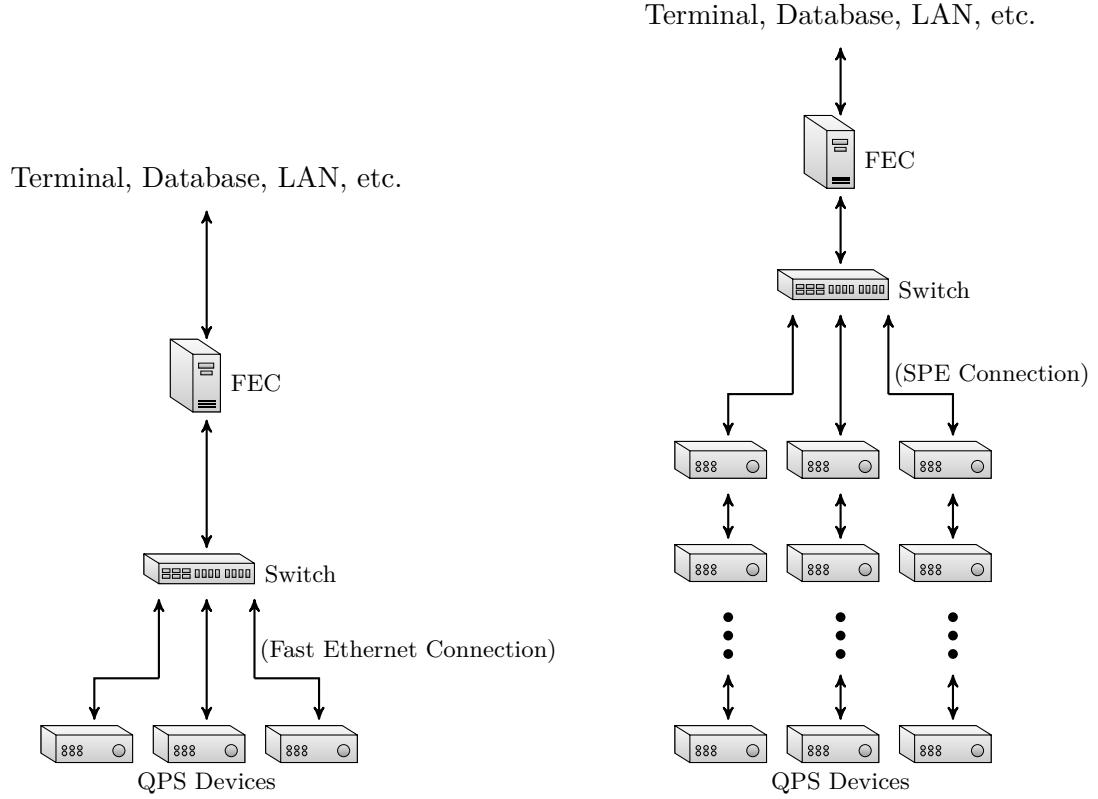
# Chapter 3

## Goals, Requirements, and Constraints

The goal of the data acquisition modernization campaign is to completely replace the WorldFIP system with Ethernet. As a minimum this replacement should allow each node to communicate with a bandwidth of 1 Mbit/s (1000 times more than what is currently supported). Additionally, like with the current WorldFIP system, a new system must support daisy chained nodes over single twisted pairs, so as not to incur expensive infrastructure upgrades in the form of additional cabling. A diagram of such a setup is shown in figure 3.1b. However, to get to this stage, several challenges must be resolved. Most importantly, such a system must be proven to function in the radiation environment found in the LHC tunnels. Additional development into daisy chained/bus topology Ethernet is also required, as it does not appear that any existing standard fulfils our needs. For example, the previously mentioned 10BASE-T1S appears to have been designed for the automotive industry and only guarantees a to function with cable length of up to 15 m. This is insufficient for our needs.

Before these problems are addressed, I have been tasked with assisting in the development of prototype hardware and software for a data acquisition system using Fast Ethernet in a standard star topology network, as shown in figure 3.1a.

While such a system does not fulfil the requirements for deployment in all areas of the LHC tunnels, it can be used in certain radiation protected areas of the tunnel, and for testing of new magnets. These two scenarios will not expose the equipment to elevated levels of radiation, and they are physically located close to FEC's thereby ensuring the added cabling from the star topology will not be a major cost concern. Once deployed, this system will provide valuable insight into the use Ethernet and should form a steppingstone to which the final system should be designed upon. As of time of writing, the goal is to deploy this first stage of the modernization campaign in June 2021.



(a) Network configuration currently under development.

(b) Final configuration conceptualization. In this configuration, multiple QPS devices are connected through Single Pair Ethernet (SPE).

**Figure 3.1:** Illustration of the data acquisition network configuration [21] envisioned for the modernization campaign.

This interim report covers the phase of the project stretching from September 2020 to December 2020. During this time, the main priorities were familiarization with the chosen technologies, and to setup a development environment where synchronization performance can be measured under a variety of networking scenarios. This setup should incorporate the existing timing infrastructure and hardware used at CERN and should mimic the configuration shown in figure 3.1a. To evaluate the validity of the setup, a rudimentary synchronization scheme should also be implemented. More specific requirements and constraints are elaborated in the following sections.

### 3.1 CERN Timing Network

CERN puts great emphasis on synchronizing all their time sensitive equipment to the same central timing system. Thus far, synchronization in the new data acquisition system has been conceptualized as being between a FEC and QPS devices directly. However, such a configuration would only synchronize the QPS devices to their host FEC, and it cannot be guaranteed that their synchronization agrees with other equipment at CERN. To overcome this the FEC would also need to be synchronized with the centralised timing network at CERN. This synchronization must be done using a CERN designed a PCI express add in timing card that is inserted into a FEC and allows for collecting central timing information.

## 3.2 Performance Requirements

When dealing with the data logging of events, it is critical that the electronics systems agree upon the time. If this is not done, one risks receiving a warped perception of the timing of events, or at worst that certain events are logged out of order. To minimize the risk of this occurring the MPE-EP section wants the new data acquisition system to guarantee a maximum time offset of  $10\text{ }\mu\text{s}$  and ideally less than  $1\text{ }\mu\text{s}$  between the FEC's timing card's clock and the QPS devices clocks. Note that this is a requirement for the final deployment, and it is not expected nor required that the implementation in this phase of the project will be able to achieve this level of synchronization accuracy.

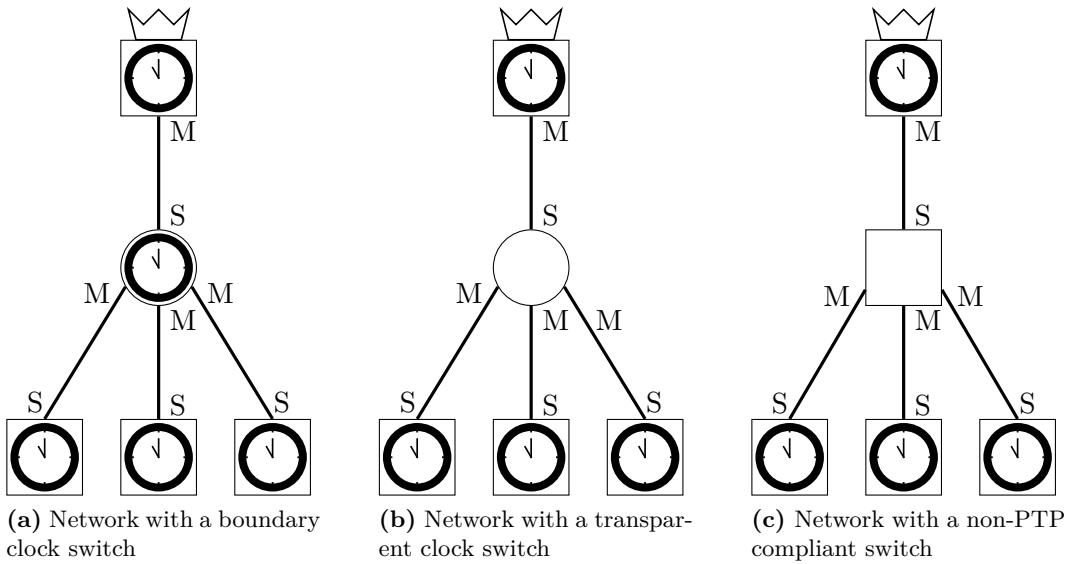
## 3.3 Existing Equipment and Constraints

As mentioned in chapter 1, various developments and design choices have already been established prior to my arrival at the MPE-EP section. Principal of which is the decision to use Precision Time Protocol (PTP) version 2, over UDP to perform synchronisation. To go along with this choice, a first iteration of a development board has also been designed and produces in limited quantities [25]. This development board acts as the data acquisition card placed in each of the QPS devices. The board contains most of the technologies required for the development of the new data acquisition system, such as support for Fast Ethernet, and quad SPI lanes to allow for communication between board's MCU and an associated QPS crate's FPGA. To facilitate synchronization, the development boards include the necessary hardware required for PTP synchronization.

Additionally, it also required that the setup uses existing CERN infrastructure and equipment such as the use of standard CERN FEC's.

## 3.4 Network Topology

Since the MPE-EP section is building a flat network, as shown in figure 3.1a, only a limited number of PTP configurations are possible. Specifically, a FEC is designated as a Grand Master (GM) ordinary clock and as each of the QPS devices are end nodes. To connect the FEC to multiple QPS devices a switch is needed. In a PTP network, a switch can either function as a boundary clock, transparent clock, or as non-PTP conforming device. These terms are described in detail in chapter 4. Each of these three networking setups are shown in figure 3.2.



**Figure 3.2:** The three PTP network configurations that are to be investigated. "M" next to a port indicates that it operates as a Master device, while "S" means Slave device behaviour.

While switches with transparent clock capabilities are broadly accepted to provide the best synchronization performance, this feature usually comes at a significantly higher unit cost. On the other hand, non-PTP switches can often be acquired for significantly less. Despite these switches degrading the synchronization accuracy, some research suggest that the synchronization penalties caused by the inclusion of high performance non-PTP switches may be manageable in many settings [26]. To learn which switch types can provide acceptable performance in the QPS data acquisition system, the MPE-EP section would like to investigate and compare each of the three configurations. This comparison is relevant from a cost perspective, as significant cost saving can be achieved if it becomes apparent that synchronization performance of non-PTP switches can remain competitive with PTP compliant counterparts.

## Chapter 4

# Precision Time Protocol

To achieve the required synchronisation performance, the MPE-EP section has decided to use the Precision Time Protocol V2, specified in IEEE 1588-2008. It was decided to use this protocol as it was designed for sub microsecond synchronization, and hardware support is mature. Other more accurate protocols exist, such as PTP V3 (IEEE 1588-2019) also known as White Rabbit. This synchronization technology was developed at CERN and allows for sub nano second synchronization over Synchronous Ethernet. However, such accuracy is not required for the QPS data acquisition system, and the hardware support for this protocol is not as mature as for PTP V2.

As a proper understanding of PTP V2 and IEEE 1588-2008 is crucial for the development of the synchronization system, this chapter describes the standard. Note that IEEE 1588-2008 is an extensive standard that specifies several irrelevant features for the desired network configuration, thus only the relevant subset of the standard is described.

### 4.1 Clock Model

To understand the difficulties in synchronizing clocks, a good start point is to understand digital clocks and how they differ from their idealization.

The most basic digital clock is made up of two key components. An oscillator operating at a fixed frequency, and a counter that is incremented following every oscillation. For example, a 100 MHz has an oscillation period of 10 ns. Thus, the counter must be incremented by 10 ns for every oscillation.

In more complex systems a clock does not simply derive a frequency from an oscillator, but from e.g. a frequency synthesizer such as a PLL. This allows one to synthesize a custom frequency by scaling the frequency of a source oscillator by some factor. The

#### 4.1. Clock Model

advantage of such a system is that it allows one to change how fast time is tracked, thus providing additional flexibility.

Ideally a digital clock tracks time exactly.

$$g(t) = t \quad (4.1)$$

However, digital clocks deviate from this model in several key areas.

Firstly, oscillators rarely oscillate with the exact frequency that they are designed for. One reason for this is manufacturing imperfection in the oscillator. For example, in a quartz crystal resonator, variations in the crystal cuts may cause the operating frequency of the oscillator to differ from its nominal frequency [27]. Furthermore, over time an oscillator's internal structure may also change, thus causing its frequency to drift further away from its nominal frequency. Finally, external factors such as operating temperature or air pressure may also impact an oscillator's frequency [27].

Another way digital clocks differ from an idealized time source, is that digital clocks only start tracking time when they are initially turned on. Therefore there is a time offset between their logged time, and the exact time. Thus, this offset will have to be corrected. However due to delay in digital systems, even following the best correction, a small offset will persist. Additionally, this offset will usually only increase over time due to the frequency variations described previously.

A conceptual model of a free running clock with the above parameters can be described as:

$$f(t) = \int_0^t \frac{X(s)}{F_{\text{Nominal}}} ds + \text{offset}_{\text{init}} \quad (4.2)$$

where

$f(t)$  is the time measured by clock.

$t$  is time (in seconds) since the clock was turned on.

$F_{\text{Nominal}}$  is the nominal frequency of the clock's oscillator.

$X(t)$  is some random process symbolizing the frequency drift of the oscillator due to aging, temperature variation, pressure variations, etc.

$\text{offset}_{\text{init}}$  is the initial offset of the clock when it is turned on.

To obtain a more accurate model of digital clocks, various other aspects, such as jitter, clock skew, and the fact that digital clocks operate in discrete intervals rather than

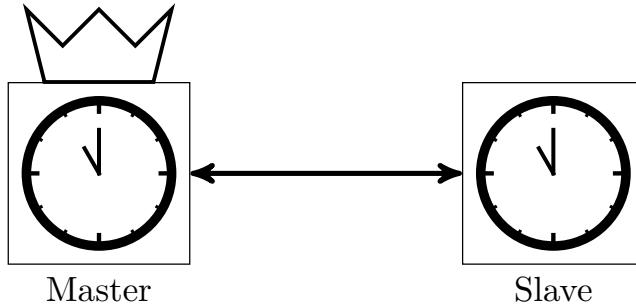
## 4.2. Packet Based Synchronization

continuously may also be accounted for. However, to understand the problems and potential solutions faced when performing synchronization between two clocks, the above model is adequate.

The stochastic nature of a clock's frequency means that in a cluster of multiple clocks they will all drift out of synchronisation over time. To circumvent this and synchronize the clocks, they must be connected in a network and each be configured to periodically share information regarding their current state with each other. This information can then be used by other clocks to infer the extend of the lost synchronisation with respect to each other. Each individual clock can then use knowledge of this time offset to alter their own state to synchronize themselves to the network.

## 4.2 Packet Based Synchronization

Consider the scenario illustrated in figure 4.1, where a slave and a master are sharing a connection, and one wishes to synchronize the clock of the slave to track that of the master.

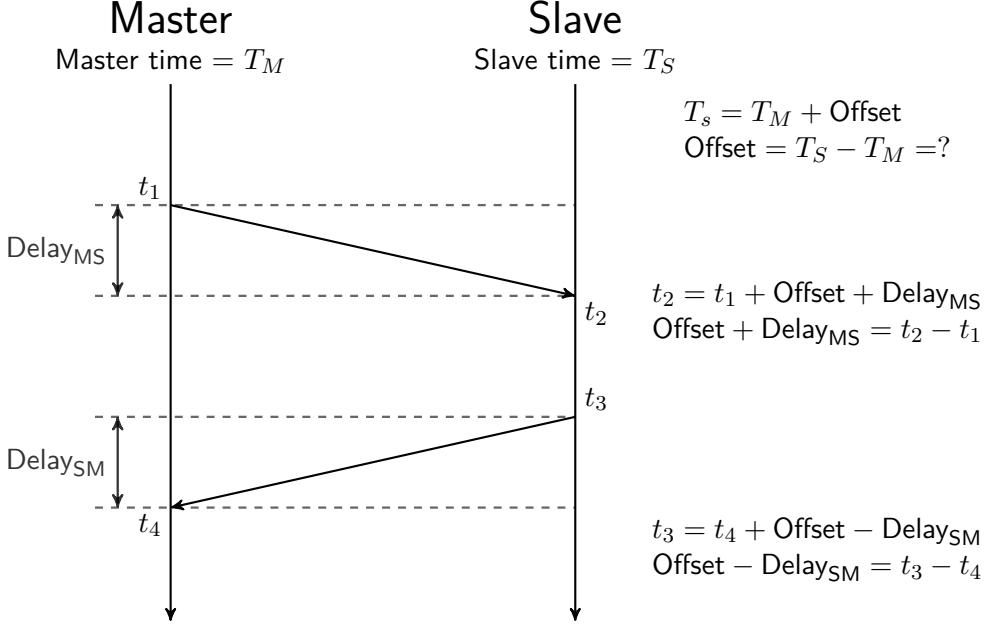


**Figure 4.1:** Setup of a simple Master/Slave communication system. The device with the crown is the master device, and the arrow between the devices symbolizes a communication channel.

The simplest method one could conceive to perform this synchronization, would be to use a one-way time transfer scheme where the master device periodically shares its time with the slave device. However, such a scheme does not consider the effects of propagation delay, thus limiting the achievable synchronization accuracy.

To compensate for the propagation delay, PTP uses a two-way time transfer scheme where an additional set of transfers are performed to make it possible to calculate the propagation delay. Figure 4.2 shows how two-way time transfer allows a slave device to calculate its local time offset with respect to the master device.

## 4.2. Packet Based Synchronization



**Figure 4.2:** Visualisation of how two-way time transfers can be used to calculate the time offset between a master and slave device [28].

As indicated by the figure, two pairs of timestamps are required to calculate the offset between the master time and slave time. The first pair,  $t_1$  and  $t_2$ , allows the slave to calculate the offset between master and slave, plus the propagation delay between master and slave:

$$\text{Offset} + \text{Delay}_{\text{MS}} = t_2 - t_1 \quad (4.3)$$

Similarly, the second pair,  $t_3$  and  $t_4$ , provides enough information to find the offset between master and slave, **minus** the propagation delay between slave and slave:

$$\text{Offset} - \text{Delay}_{\text{SM}} = t_3 - t_4 \quad (4.4)$$

If the propagation delay of the connection between the two devices is symmetrical, then  $\text{Delay}_{\text{MS}} = \text{Delay}_{\text{SM}}$ . When this is the case, the information from the two sets of timestamps can be combined, to find the value of the offset:

$$2 \cdot \text{Offset} + \text{Delay}_{\text{MS}} - \text{Delay}_{\text{SM}} = (t_2 - t_1) + (t_3 - t_4) \quad (4.5)$$

$$2 \cdot \text{Offset} = (t_2 - t_1) + (t_3 - t_4) \quad (4.6)$$

$$\text{Offset} = \frac{(t_2 - t_1) + (t_3 - t_4)}{2} \quad (4.7)$$

## 4.2. Packet Based Synchronization

If the delay is asymmetrical, this method will not produce accurate results without further knowledge of the asymmetry. Thus, to achieve an accurate estimate of the offset, this aspect of the system is of critical importance.

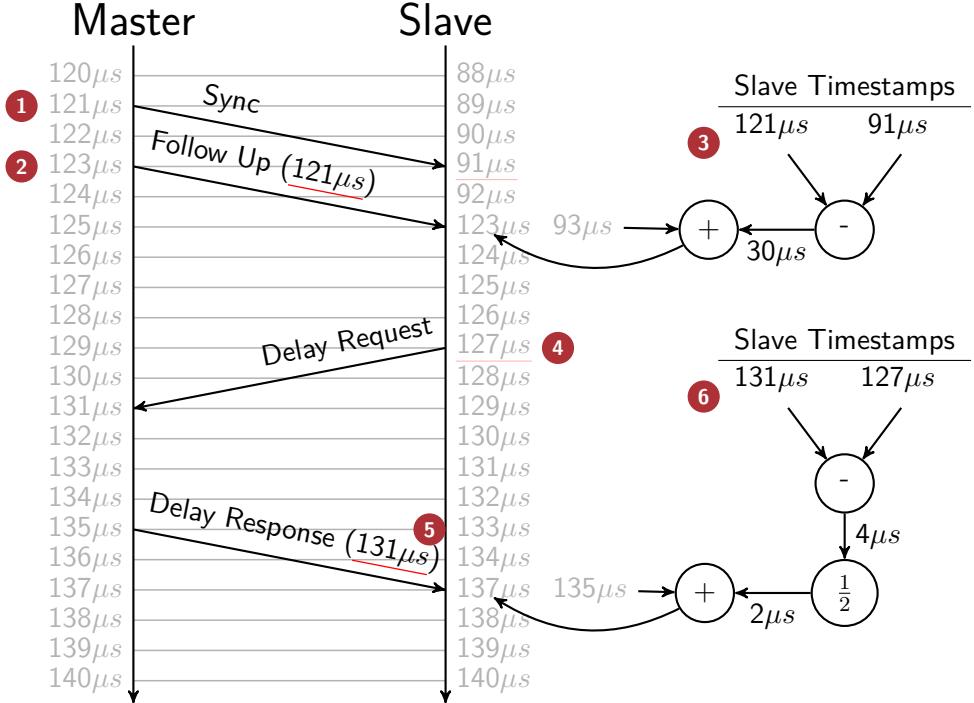
Once the time offset is found, it can be used to synchronize the clocks. PTP does not specify how this synchronization, is performed and it therefore left as a design challenge.

To distribute the four timestamps to the slave device, PTP defines four different packets.

1. Sync. This packet is sent from the master to the slave. Upon transmission the packet is timestamped by the master to produce  $t_1$ . This timestamp is included in the packet. Similarly, upon reception the slave creates timestamp  $t_2$ .
2. Follow Up. In certain cases, it is not feasible to include  $t_1$  in the Sync packet. In these situations, a Follow Up packet containing  $t_1$  is send immediately following a Sync packet.
3. Delay Request. Send from the slave to the master. It is timestamped by both the slave and master to produce  $t_3$  and  $t_4$  respectively.
4. Delay Response. Is used to send  $t_4$  from the master to the slave.

To show how this can be done in practice, the timing diagram in figure 4.3 and associated descriptions provide an example of PTP synchronization where the calculated offset is used to directly alter the slave clock to match that of the master.

## 4.2. Packet Based Synchronization



**Figure 4.3:** PTP timing Example [29] [30]. The red underlined times are the timestamps which are saved on the slave and required for performing a synchronization.

- ➊ The master sends a `Sync` packet to the slave. The master logs when this packet was send. Similarly, the slave logs when it received the packet.
- ➋ The master sends a `Follow Up` packet which includes the timestamp that the `Sync` message was send.
- ➌ When the slave receives the `Follow Up` packet with the master's `Sync` timestamp, it has enough information to calculate the offset and propagation delay between its clock and the master's. This information is used to complete the first step of the synchronization.
- ➍ The slave sends a `Delay Request` packet to the master. The slave logs when the packet was sent, and similarly the master logs when it received the packet.
- ➎ The master sends a `Delay Response` packet back to the slave. This packet contains the timestamp of the reception of the `Delay Request` packet.
- ➏ The slave can now calculate and compensate for the propagation delay. This completes the synchronization.

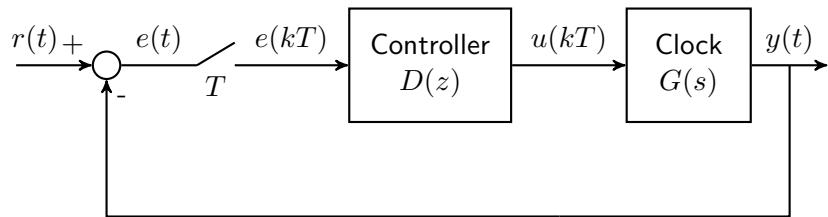
### 4.3. Synchronization Methods

The PTP synchronization described above must be performed periodically. Otherwise, slight imperfections in the master's and slave's clock will over time cause the pair to lose synchronisation again. To mitigate the magnitude of this drift, one option is to use high accuracy clocks with smaller frequency deviation. Another option is to increase the frequency that a PTP synchronization takes place.

In non-time sensitive computing systems, it is quite common to use cheaper oscillators with a max frequency deviation of  $\pm 50\text{ppm}$  (parts per million) or  $0.005\%$ . In other words, for every million cycle, such an oscillator may be off by up to  $\pm 50$  cycles. When tracking time this equates to a worst-case time drift of  $50\text{ }\mu\text{s}$  per elapsed second. Furthermore, if one is to synchronize two clocks, one must sum each of the clocks frequency deviation to calculate the impact on the synchronization accuracy. For example, if both the master and the slave use  $\pm 50\text{ppm}$  clock then in relation to each other their frequency deviation might be as high as  $\pm 100\text{ppm}$ . In this scenario the time drift will be up to  $100\text{ }\mu\text{s}$ , and thus more than 100 PTP exchanges must be performed per second if sub microsecond precision is to be guaranteed. If on the other hand higher quality  $\pm 1\text{ppm}$  clocks are used, then only two or more exchanges are required per second. While such clocks are more expensive to source, it allows for a larger share of the devices bandwidth to be allocated to sharing data rather than for exchanging synchronization packets.

## 4.3 Synchronization Methods

In the previous example in figure 4.3, synchronization is performed by using the calculated offset to set the current time in the slave device. However, immediately following synchronization the clocks will lose synchronisation due to clock imperfection as mentioned in section 4.1. To overcome this, most PTP implementation perform synchronization by using a feedback control system to continuously change the frequency of the slave clock [31, p. 146]. The purpose of such a controller is to speed up or slow down the slave clock until the offset between itself and the master clock is zero. Once synchronization is achieved, the controller continues to alter the slave's clock frequency to track that of the master's, thus counteracting the naturally occurring frequency deviations that inevitable occur between the two clocks. A conceptual example of a feedback clock control system is visualized using a block diagram in figure 4.4.



**Figure 4.4:** Example of a control loop used to synchronize a slave clock.

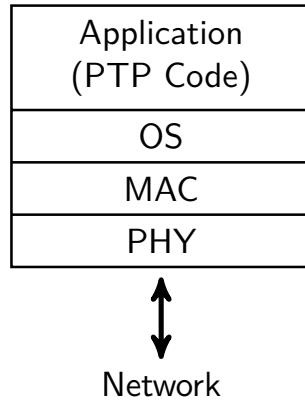
#### 4.4. Minimizing Asymmetry and Time-Stamping Delay

Here  $r(t)$  and  $y(t)$  represents the master and slave time respectively.  $e(t)$  is the difference, or offset, between these. This offset is periodically sampled with a sample period of  $T$  to produce a discretized offset signal,  $e(kT)$ . When using PTP, the interval at which Sync and Delay Response packets are exchanged, corresponds to the sampling period  $T$ , and the resulting offset is  $e(kT)$ . The controller uses  $e(kT)$  to calculate a new clock frequency,  $u(kT)$  which in turn is used as input to the plant. In this setup the plant is the slave clock and it uses the input  $u(kT)$  to determine the frequency its frequency synthesizer should be set to, thereby directly altering how fast the clock tracks time.

While many types of controllers can be used, the most common choice is a PI controller [31, p. 146]. Additionally, one is not limited to classical feedback control, as different schemes have been developed. These alternative methods are commonly based on stochastic modelling of the jitter and clock variations found in PTP devices [32] [33]. This interim report does not cover these synchronisation schemes in greater detail, as the focus is on setting up the PTP development environment. Instead it is expected that the topic will be of paramount importance in the next phases of the project where implementing and evaluating PTP becomes the central priority.

## 4.4 Minimizing Asymmetry and Time-Stamping Delay

One key concern with the presented model is the assumption that the propagation delay is symmetrical and the instantaneous generation of timestamps. This is infeasible to achieve software solution, due to the many software and hardware abstraction layers separating a PTP application from the networking and timing hardware of a device. Figure 4.5 illustrates an example of some of these layers.



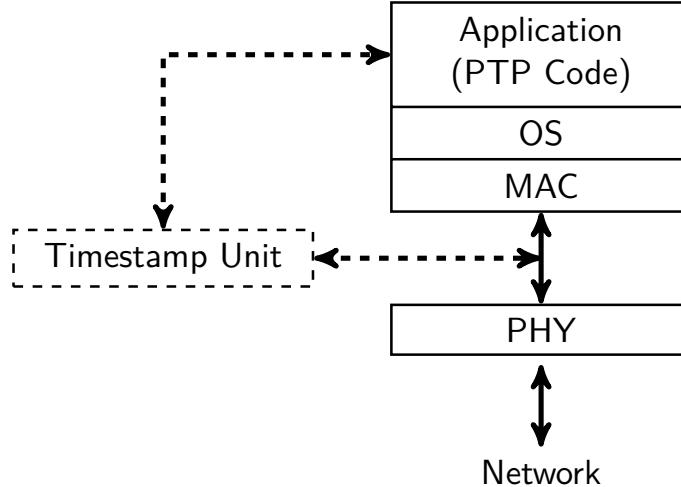
**Figure 4.5:** Visualization of the hardware/software layers that PTP packets will have to traverse to be received/transmitted by a PTP application.

#### 4.4. Minimizing Asymmetry and Time-Stamping Delay

Using the figure, consider the case where the application receives a Sync or Delay Request packet and is therefore required to accurately timestamp the message reception. In this case the message is initially received at a physical layer circuit, called a PHY, where the analogue signal of the transmission medium is interpreted and converted to a digital format for the Medium Access Control (MAC) layer. Once the MAC has validated the packet it is send through the operating system (if the device has one) and on to the network stack, e.g. a TCP/IP stack. Based on an identifier (such as port number 320 and 319 when using PTP over UDP) the packet can be identified as a PTP packet and is send to a PTP application for further processing. Here the packet is decoded and upon identification as a Sync or Delay Request packet the application must finally generate a timestamp.

The time required for the PTP packet to pass through the various hardware and software layers can be significant. Furthermore, as few of these processes are deterministic the delay between packet reception and timestamp generation is not constant and can vary depending on factors such as network and CPU load. As delay occurs in both the slave and the master device it becomes difficult to accurately synchronize their clocks.

To get as close as possible to the idealized assumptions the delay between packet transmission/reception and timestamp generation must be minimized. To achieve this, IEEE1588 specifies the use of hardware timestamping as early in the transmission/reception layers as possible. As exemplified by figure 4.6, this logic is often implemented somewhere between the MAC and PHY.



**Figure 4.6:** Visualization of hardware timestamping used in PTP [30].

The time-stamping unit (TSU) keeps its own track of time and continuously monitors network traffic going to and coming from the PHY. When a PTP packet is identified, the TSU logs the current time in a register. The PTP app can access the logged timestamps

#### 4.5. Clock Types and Clock Hierarchy

from the TSU registers and use these directly instead of generating its own timestamps. This hardware-based solution eliminates a significant amount of the delay and unpredictability associated with a pure software solution. Some TSU implementations go a step further and allow for direct manipulation of outgoing PTP packets. This is used when a master device sends a Sync packet. Here the TSU can modify the packet and set the exact transmission time just before it is sent across the medium by the PHY. If a TSU does not have such modification capabilities, the transmission timestamp is instead logged and must be send by the PTP application in a subsequent Follow Up packet.

## 4.5 Clock Types and Clock Hierarchy

Previously, only networks with a single master and slave were considered. Real networks may have a significantly more complicated topology than this. In larger networks, one node, called the grandmaster (GM) clock, is the ultimate time authority and all other nodes in the network must be synchronized to its local time. To accommodate differing needs, IEEE1588 defines a set of different nodes, called clocks, which may be present in a PTP network. These clocks have one or more ports allowing it to synchronize with other clocks in the network. Each port on the device is either acting as PTP master or slave. The clocks defined by IEEE1588 are described in table 4.1.

## 4.5. Clock Types and Clock Hierarchy

**Table 4.1:** PTP clock descriptions and symbols.

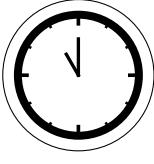
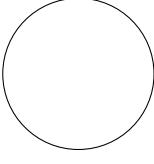
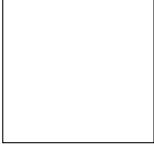
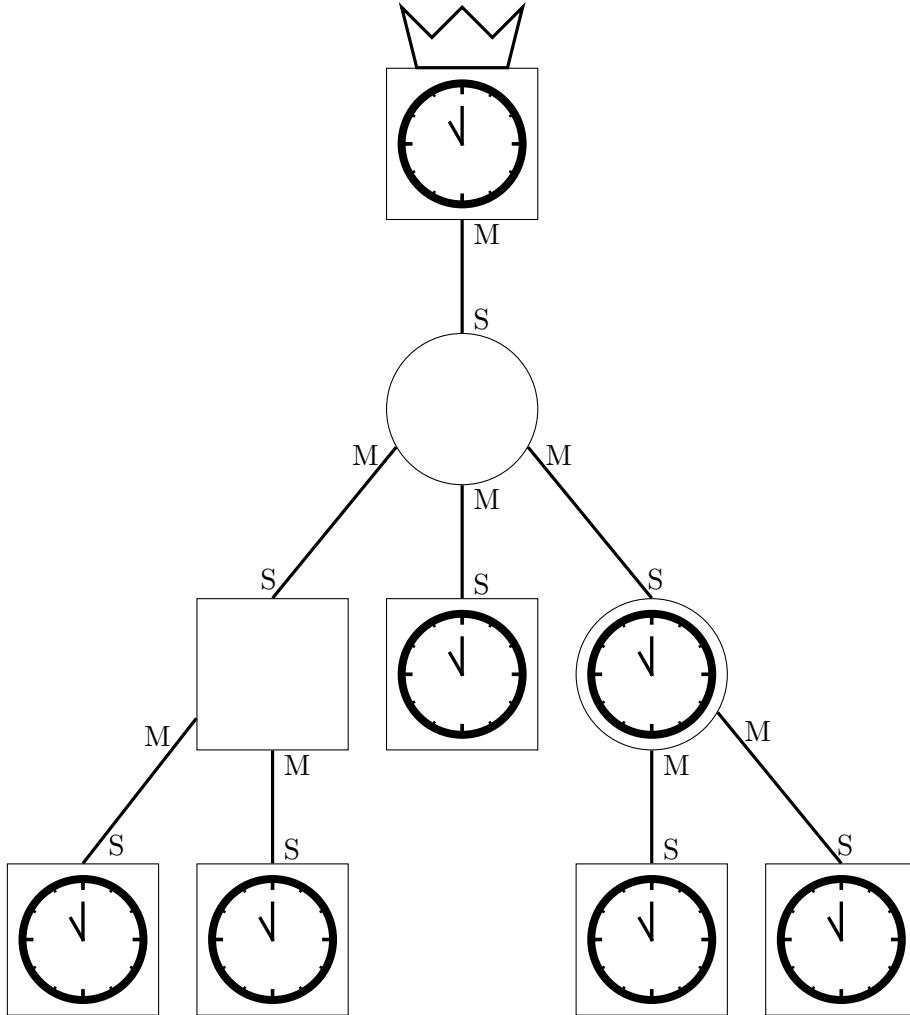
Symbol	Description
	<b>Ordinary Clock:</b> These nodes support running one copy of the PTP protocol and can directly connect with one other device through one port. Ordinary clocks are either slave or grandmaster devices. They usually represent the end node application devices in a network [30].
	<b>Boundary Clock:</b> These clocks can run multiple copies of the PTP protocol over many ports thus allowing it to synchronize multiple devices. They are most often synchronized to another master device and are used to expand the network by distribute the timing to multiple other slave devices. Boundary clocks often take the form of special PTP switches or other networking equipment that is used to connect multiple devices together [30].
	<b>Transparent Clock:</b> Like boundary clocks, transparent clocks are used to distribute the time to multiple slave devices. However, transparent clocks are not synchronized to a master device but instead forwards a master device's PTP packets to the transparent clock's slaves. As routing packets through additional equipment will add additional, and random, delay to the transmission of packets, synchronization may be impaired. To overcome this, transparent clocks uses sophisticated hardware to measure the duration a packet resides in the hardware before it is relayed. This time is written in the packet, thus allowing slave devices to use this information to compensate for the added delay [30].
	<b>Non-PTP Clock:</b> While discouraged by IEEE1588, PTP networks may contain devices which are non-PTP compliant such as standard networking switches. Such devices will not help synchronize other clocks in the network and may in fact add unaccounted packet delay that can degrade the synchronization accuracy.
	<b>Grandmaster Clock:</b> The Grandmaster clock is the ultimate authority of time in the network, for which all other devices derive their own local time. In subsequent diagrams the GM is denoted by a clock wearing a crown as seen on the figure on the left.

Figure 4.7 illustrates an example of how these clocks can be connected in PTP network.

#### 4.5. Clock Types and Clock Hierarchy



**Figure 4.7:** Example of PTP network using a GM ordinary clock, a transparent clock, an ordinary clock, a non-PTP Clock, and five ordinary clocks as end nodes.

This concludes the description of the aspects of PTP and IEEE1588 that are necessary to allow network synchronization in the future QPS data acquisition system. As mentioned in the introduction, only a small part of IEEE1588 has been described in full. In a feature complete PTP implementation, each of the nodes in a network would support many other features, such as the use of a *best master clock* algorithm to automatically select a master clock with the best clock quality [30].

# Chapter 5

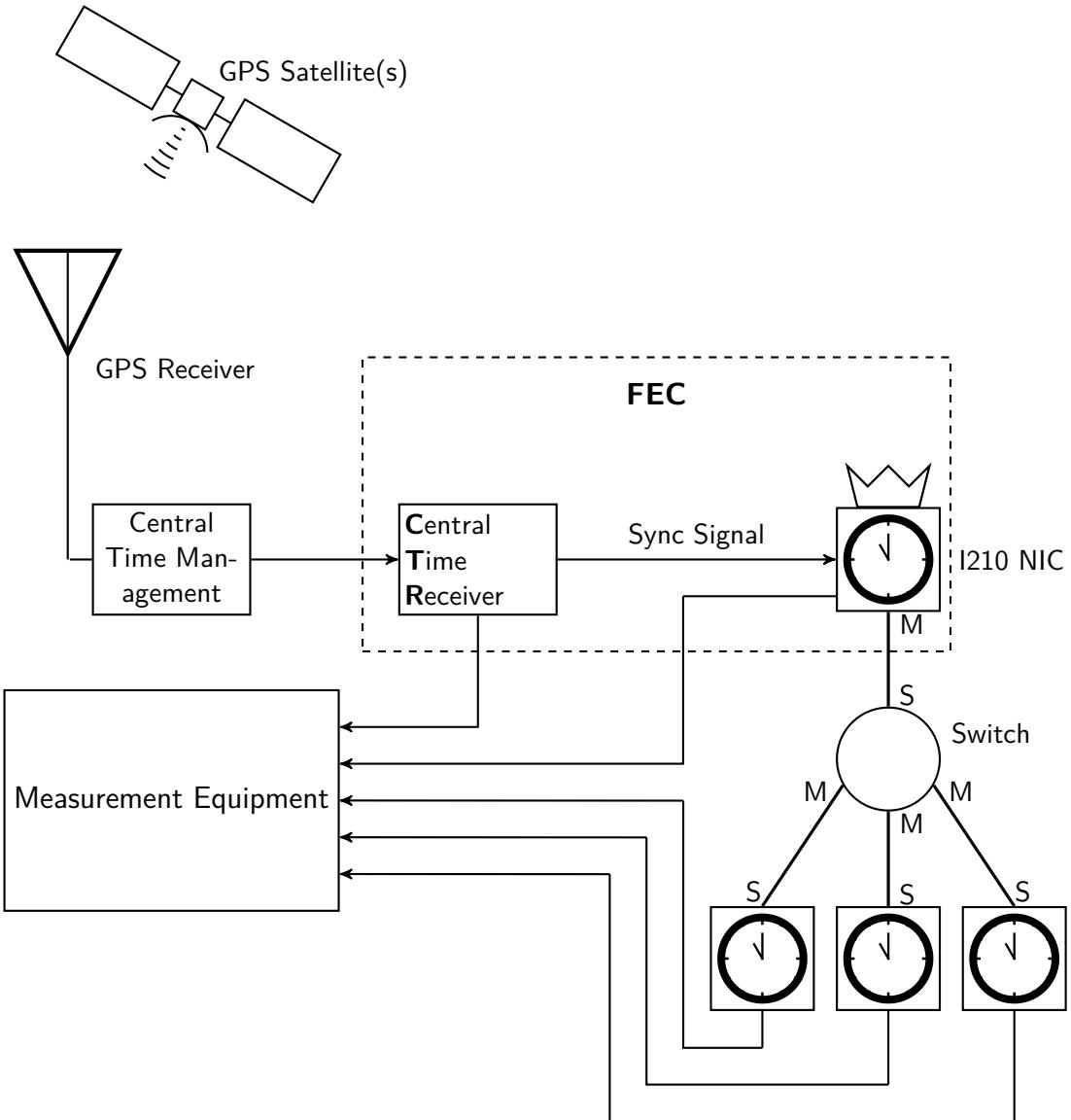
## Timing Network Topology and Hardware

To facilitate development my section has provided me with a range of pre-existing tools, devices, and software frameworks either created within my section or at one of the many other development teams at CERN. This chapter details the hardware and infrastructure that is used to setup the synchronisation development environment.

### 5.1 Topology Overview

A diagram of the desired system topology is shown in figure 5.1. The diagram illustrates how time propagates through the CERN timing network to finally reach the QPS devices. At CERN, all timing critical equipment derives their time from a single GPS receiver that receives accurate timing events from GPS satellites. This receiver distributes the time to central timing systems associated with each accelerator. Finally, the central timing systems distribute the time to the accelerator equipment through central time receivers (CTR) [34]. These central time receivers are made as expansion cards that are inserted into the accelerators various FECs. For the FEC in use in this project, the CTR is PCIe device, with designation CTRIE. The FEC is connected to the data acquisition system PTP network through its networking interface. This interface contains the TSU required to timestamp PTP packets. To synchronize the PTP network to the centralised CERN time, the TSU is synchronised with the time received by the CTRIE add in card. Finally, to allow for measuring the synchronisation performance, all distinct nodes part of the synchronisation network must periodically output signals that will allow one to calculate their temporal offset with respect to each other. This is also shown in figure 5.1.

## 5.2. Front End Computer Hardware



**Figure 5.1:** The timing distribution and measurement system to be created.

## 5.2 Front End Computer Hardware

The FEC at disposal is a Siemens SIMATIC IPC647E rack mounted industrial PC, as seen in figure 5.2. This model is CERN's new industrial PC of choice, and it is in the process of replacing older models in more than 800 installations. [35].

## 5.2. Front End Computer Hardware



**Figure 5.2:** Picture of the Siemens SIMATIC IPC647E rack mounted industrial PC [36], that is used as a FEC.

The PC contains an Intel I5-8500 and is configured with 8 GB of RAM. It is a diskless machine that network boots a CERN variant of CentOS 7 [18].

The FEC contains a CTRIE card, shown in figure 5.3, that can directly interact with the FEC through its PCIe connection, or with other external equipment through its three coaxial outputs.



(a) Top view of the CTRIE card [37].



(b) Side view of the CTRIE card's IO [38]. The IO contains a two-port differential RS-485 input driven by the central timing network, external oscillator input, three coaxial output channels, and status LED's.

**Figure 5.3:** Images of the CTRIE card.

### 5.3. Switch

The operations of the CTRIE can be programmed to suite its specific purpose. For example, the central timing system keeps track of key events in the acceleration cycle, such as particle beam extraction, or initiation of beam acceleration. These events are distributed to the CTR, and the cards can be programmed to perform actions based on a certain event occurring. Actions include triggering an interrupt in the FEC to start a software routine, or alternatively it can directly generate a pulse on its front IO to activate external equipment. In the specified use case, it is only desirable to know the exact current time, and specific acceleration events are not of interest.

Additionally, the FEC is equipped with a Network Ethernet Controller (NIC), the Intel I210, that has an inbuilt TSU which is used to timestamp PTP events. Furthermore, the I210 chip can be configured to accurately timestamp rising/falling edge input signals on one of its four software defined pins. Like with PTP, these timestamps can be used to aid synchronization, and broadly eliminates the delay problem caused by a pure software solution. Unfortunately, none of the programmable pins on the chip were exposed on the FEC's motherboard. To solve this, an external PCIe network expansion card, shown in figure 5.4, with exposed pins was acquired. To ensure driver compatibility with the available Linux distribution, the chosen network card uses an I210 NIC that is identical to the one present on the FEC's motherboard. The details of how these pins are used for synchronising the FEC's TSU to the CTRIE is described in chapter 7.



**Figure 5.4:** I210 NIC installed in the FEC [39]. Notice the exposed jumper pins. These are connected to ethernet controllers programmable IO.

## 5.3 Switch

In addition to the GM clock, a switch is also needed to distribute the timing to the multiple QPS devices. As mentioned in chapter 4, testing and verifying the behaviour

### 5.3. Switch

of both a transparent and boundary clock, as well as a non-PTP compliant switch is desirable. To simplify future testing, it is decided to acquire a switch that can operate as all three types of clocks in a PTP network. The chosen switch is a Juniper QFX5110-48S and is shown in figure 5.5.



**Figure 5.5:** Image of the Juniper QFX5110-48S[40].

This is a switch designed for data centres, and as such it supports many features that are unnecessary for our needs. While cheaper switches with PTP capabilities exist, this is chosen as it will be easier to integrate in the operational environment since most of CERN's modern network infrastructure is built with Juniper equipment.

# Chapter 6

## Data Acquisition Board Development

The final element in the PTP network to consider is the data acquisition system in the QPS devices acting as ordinary clocks. As this does not exist as an off the shelf solution, this aspect of the system constitutes the most significance part in terms of the labour and design work required.

### 6.1 Hardware Design

One of the central aspects of new data acquisition system, is the need for a new data acquisition PCB that supports Ethernet and has hardware support for PTP. Recall that upon my arrival at CERN, an initial PCB had already been developed by a previous student intern [25]. However, it became clear to me and my immediate supervisor that key changes to the PCB were required for improved PTP performance. The most important of these changes, were the replacement of the low grade  $\pm 50$  ppm clock with a superior temperature-compensated crystal oscillator (TCXO). This change was made based on recommendations from literature to not use low grade oscillators for the clocks in PTP networks[41]. It is hoped that this substitution will make the desired synchronisation accuracy of less than 1  $\mu$ s more attainable.

The redesign and update of the PCB to match our revised requirements, included many changes that were not paramount to the synchronization aspects I would be working on, but would be beneficial for later developments of the data acquisition system. In addition, to the data acquisition board, I also designed a backplane for a 19-inch rack. This backplane allows for up to 14 data acquisition boards to be powered simultaneously while organizing them in an easily accessible manner.

## 6.2. Firmware

An image of the data acquisition PCB and the backplane is shown in figure 6.2 and figure 6.1 respectively. Certain regions/components in these images are labelled, and a corresponding description of the highlighted features are provided in table 6.2 and table 6.1.

## 6.2 Firmware

For the development and testing of PTP synchronization it also required to write firmware for the designed acquisition hardware. The firmware written for this purpose is only to be used for PTP development and does not include the numerous other functionalities that the data acquisition platform must support when it is eventually finished and deployed to the operational environment in the LHC. While currently undecided, it is expected that another section, the Machine Protection Software section (MPE-MS), will aid in the consolidation of the feature set to create the final firmware that can fulfil CERN's stringent operational and radiation requirements. Due to this later focus on redundancy and polish, the purpose of the current PTP firmware is only to investigate synchronization performance and to learn how to implement PTP using the manufacturer provided networking library, and the MCU's timestamping hardware. As a result, the firmware development has focused on flexibility and simplicity. For example, the PTP firmware is built on top of the real time operating system FreeRTOS. CERN prefers to avoid the use of RTOS's for operational deploy, as they believe that it is hard for an RTOS implementation to compete with a close to the metal implementation in terms of deterministic behaviour, and the level of control over the code and hardware. However, for the case of testing PTP the choice is justified since the use of an RTOS allows for the concurrent execution of multiple processes, thus simplifying development.

As these data acquisition devices are to exclusively operate as slave nodes in a PTP network, only slave functionality has been implemented in the firmware. In the current implementation, synchronization is performed by using the calculated offset between the master and slave device to update the slave clock so that it matches with that of the master. This mimic the scheme used in the example shown in figure 4.3. The current synchronization method acts as a placeholder until superior and more advanced synchronization schemes are investigated and implemented.

In addition to the PTP functions, for the purpose of future testing and development it is necessary to measure the accuracy of the synchronization. This functionality has also been accounted for in the design, however further description of this is postponed until the following chapter.

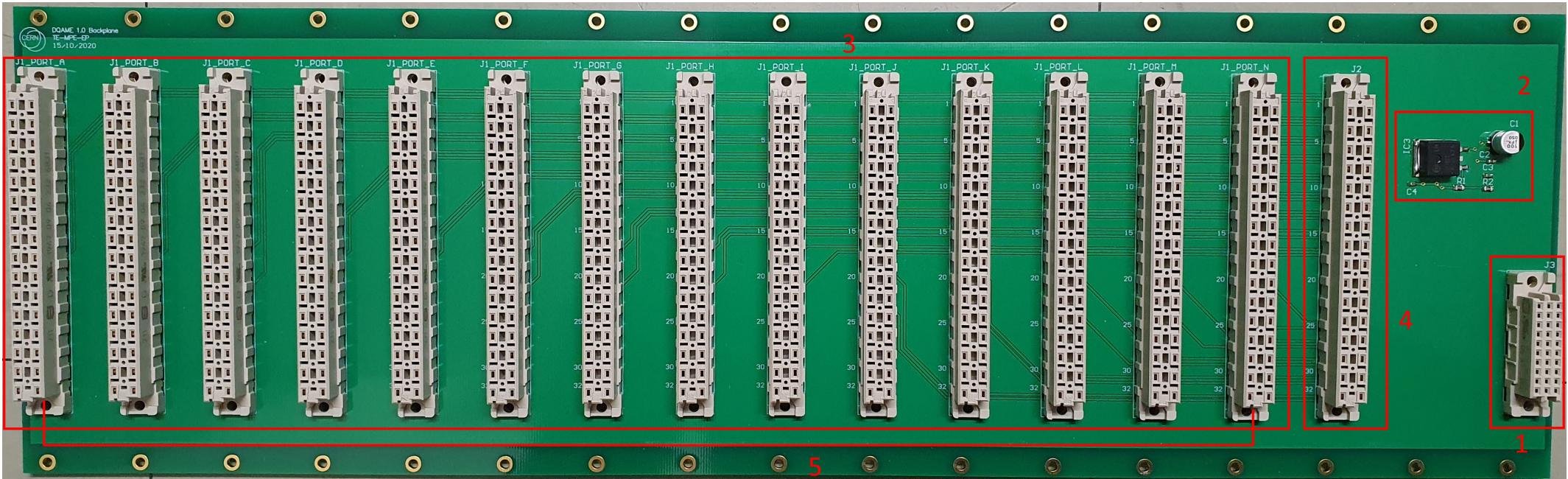
Finally, since the firmware is rapidly evolving and up to 14 data acquisition cards (the number supported by the backplane) are required for the performance evaluation of the PTP network, it will be quite cumbersome and time consuming to have to manually

## 6.2. Firmware

update the firmware of every single device every time a new firmware version is ready for testing. To overcome this, I implemented a bootloader provided by the MCU's manufacturer. This allows for remotely updating the firmware using UDP. In practice, when the data acquisition is running the PTP firmware, it monitors incoming packets. When it receives a specially marked "go into bootloader" packet, the MCU resets and boots in the bootloader. When in this mode, MCU can receive the firmware binaries over UDP and automatically reflashes the application.

**Table 6.1:** Description of key backplane features shown in figure 6.1.

#	Description
1	Power supply connector. Provides the data acquisition boards with 5 V of power.
2	Voltage regulator used to regulate the 5 V input down to 2.5 V. This is equivalent to the QPS FPGA's logic level, and it is used to provide a voltage reference to the data acquisition board's QSPI logic level converters.
3	Slots for connection of up to 14 data acquisition boards to the backplane.
4	Debugging port. Each of the 14 data acquisition board slots is connected to this port through two PCB traces connected to their GPIO pins. The port eases the testing phase as it allows measurement equipment to monitor all the exposed GPIO pins through just one connector.
5	The SPI lanes of the outer two most data acquisition board slots are connected to facilitate testing of the Quad SPI system.



**Figure 6.1:** Top down image of the backplane PCB . A description of the labelled components/sections can be found in table 6.1.

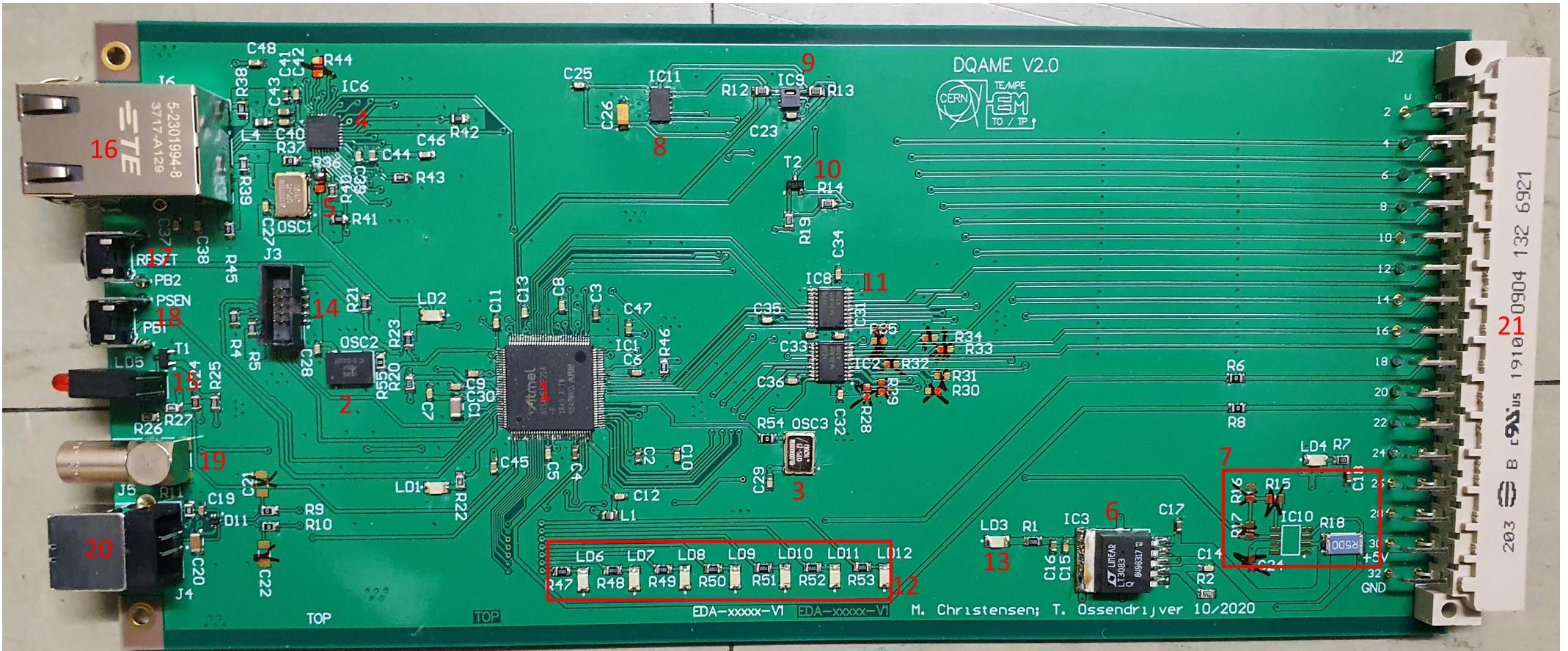


Figure 6.2: Top down image of the data acquisition PCB. A description of the labelled components/sections can be found in table 6.2.

## 6.2. Firmware

**Table 6.2:** Description of key data acquisition PCB features shown in figure 6.2.

#	Description
1	Microchip ATSAME54P20A 32 bit-MCU. This MCU was chosen due to its support for ECC RAM (critical for safe operation in radiation environments), Quad SPI, PTP hardware timestamping, and relatively low price [25].
2	High precision 40 MHz temperature compensated crystal oscillator (TCXO). It has a max frequency deviation of about 1PPM. In this application, it is critical to have a high precision oscillator to maintain accurate synchronization. Chosen as a compromise between the maximum frequency deviation that can be expected through one year of use, price, and availability.
3	Secondary standard 40 MHz XO. Allows for two independent timing systems within the MCU. This can be used to increase redundancy in case of faults caused by radiation.
4	Ethernet PHY chip.
5	Standard 25 MHz used as the clock for the Ethernet PHY.
6	Voltage regulator, regulating a 5 V input down to 3.3 V.
7	Current sensing circuit build using a difference amplifier and a current sensing resistor. To be used to monitor the board's current consumption, and aid in fault detection. Note: not soldered on shown board, as the circuit does not currently behave as expected.
8	Accelerometer to be used to detect if and when the crate with the data acquisition board is moved/touched/impacted.
9	Humidity Sensor to measure the humidity of the board's operating environment.
10	Transistor to turn off power to accelerometer and humidity sensor. In case the sensors break or interfere with the MCU, this transistor allows one to completely disable the sensors.
11	Logic Level shifter that converts from ATSAME54P20A logic level (3.3 V) to the QPS FPGA's communication logic level (2.5 V).
12	Cluster of LED's connected to GPIO pins on the MCU. To be used for debugging.
13	Power indicator LED.
14	Programming Port.
15	Green and Red LED's connected to GPIO pins on the MCU. Visible on front IO.
16	RJ45 Ethernet socket.
17	Reset button.
18	GPIO push button.
19	LEMO female connector (coaxial connector used at CERN). Connected to MCU GPIO port.
20	USB B female connector. Primarily used for serial communication.
21	32 pin DIN 41612 Type D. Supplies power to the board, contains pins connected to the MCU SPI interface (for communicating with a QPS FPGA board), and is connected to two GPIO pins for general purpose debugging.

# Chapter 7

# System Integration

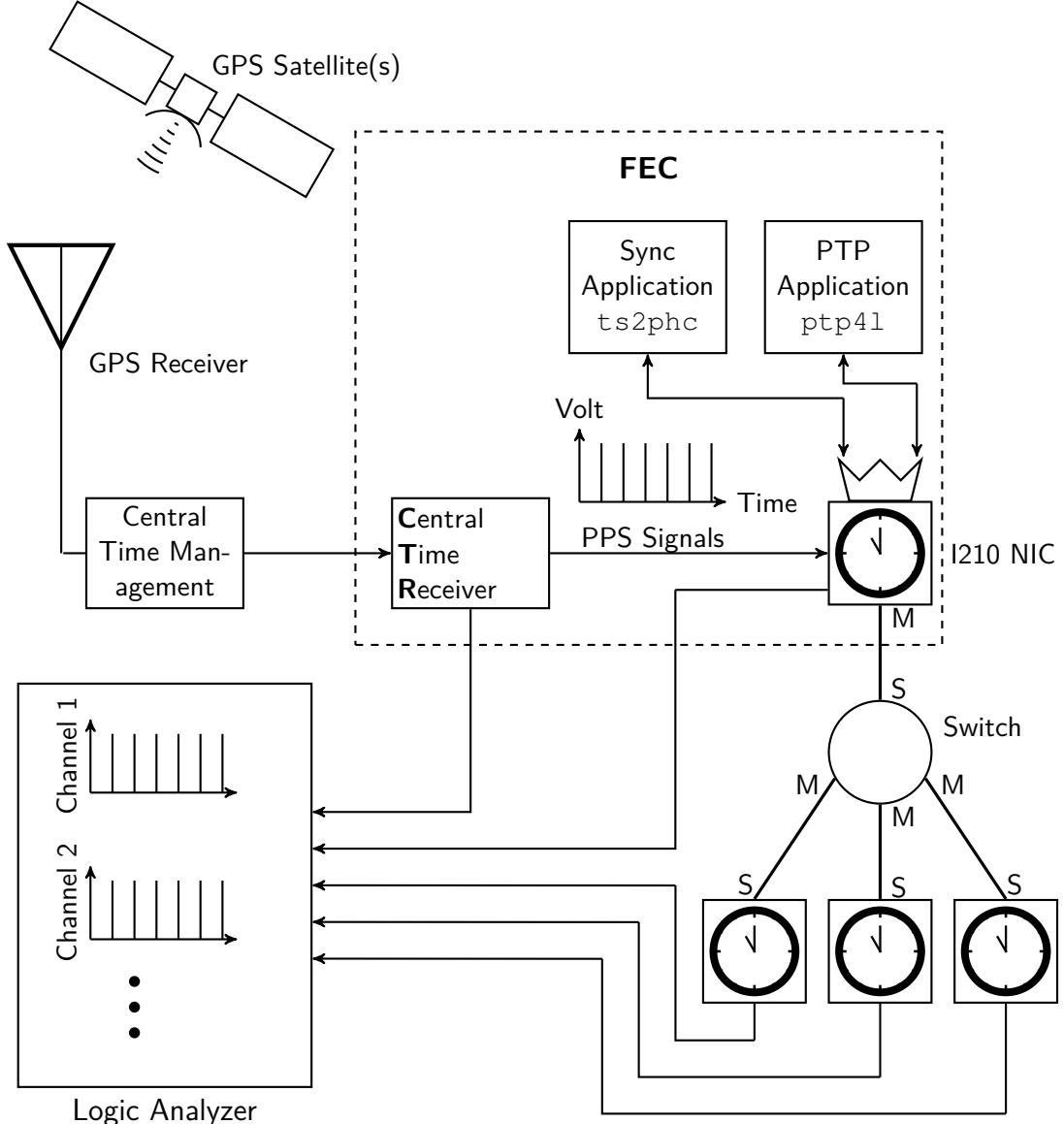
Having listed the system hardware and detailed the custom hardware/software creations, this chapter clarifies how these sub-systems are integrated and combined to form the final system. The interaction between the sub-systems is illustrated in figure 7.1. The details of these interactions are elaborated further upon throughout the chapter.

## 7.1 FEC and CTRIE Synchronisation

When performing synchronisation, the total time offset between the CTRIE clock and the acquisition board's clocks depends on both the offset between the CTRIE and I210 NIC in the FEC, and the subsequent offset between the I210 and the acquisition cards. Thus, to achieve the synchronisation performance requirement between the CTRIE and the acquisition cards it is of critical importance that the CTRIE and the I210 are synchronised such that their clock's offset is less than the requirements. To perform synchronization between the CTRIE and FEC's I210 NIC, a software approach was initially considered where the FEC would periodically read the time from the CTRIE and then write it to the clock on the Ethernet controller where the TSU resides. However, as discussed in chapter 4, software-based synchronization solutions introduce significant delay impairing the synchronization accuracy. Thus, like with PTP, a hardware assisted solution is desired to minimize delay. This inspired the use of the Ethernet controller chip's programmable IO to aid in the synchronisation.

The CTRIE can be configured to periodically emit pulses aligned at the beginning of every second, known as a Pulses Per Second (PPS). For example, with 2PPS output pulses are send every 500 ms with every second pulse being emitted at the second increment. In the following it is assumed that a 1PPS output is used.

## 7.1. FEC and CTRIE Synchronisation



**Figure 7.1:** More detailed illustration of figure 5.1. This figure specifies the software and signals used for performing and measuring synchronisation.

The CTRIE 1PPS output is connected to one of the Ethernet controller's software defined pins, where each pulse is timestamped by the controller's clock. Software in the FEC retrieves this timestamp and processes it. Since it is known that the pulse was sent on the exact second, the timestamps' deviation from the nearest second is set as the offset between the CTRIE and the Ethernet controller's clock <sup>1</sup>. The offset of these pulses

---

<sup>1</sup>This only works if the clocks are already within  $\pm 0.5\text{s}$  of each other. It is initially the case as the approximate current time is derived from the NTP protocol running on the FEC's OS.

## 7.2. PTP Application

is used to estimate the frequency deviation between the slave and the master clock. This frequency offset is subsequently used to synchronize the Ethernet controller's clock using a control loop to slightly adjust its own clock frequency. This is equivalent to the PTP synchronization scheme covered in section 4.3. In practice, to perform the synchronization described above, the `ts2phc` program distributed by the *Linux PTP Project* is used [42]. It was chosen to use this program as it has been in development for many years and appears to enjoy wide adoption and support from industry.

## 7.2 PTP Application

Once the Ethernet controller is synchronized to the CTRIE, and by extension the timing network, the FEC can start running a PTP application to synchronize the QPS devices through the network configuration previously shown in figure 3.2. The program `ptp4l` is chosen as the PTP application like with the CTRIE and Ethernet controller synchronization application, `ptp4l` also part of the *Linux PTP Project* [42]. `ptp4l` is a linux implementation of PTP that supports hardware time-stamping and allows for flexible configuration. In our use case, the program is configured to act as grand master that periodically (subject to the configuration) broadcasts PTP sync packets to the network. When the FEC receives `Delay Request` packets from other nodes in the network, the program automatically responds with corresponding `Delay Response` packets.

## 7.3 Synchronisation Measurement

In addition to the PTP functions, for the purpose of testing and development it is necessary to measure the accuracy of the synchronization. This is measured by programming the data acquisition devices to output PPS signals, and then using an oscilloscope or logic analyser to compare their time shift/offset with PPS signals created by the CTRIE timing card. To obtain reliable results, it is critical that software overhead is minimized so that the PPS signals are electrically generated as close to their desired generation time as possible. The chosen MCU's PTP time stamping hardware allows for precise time tracking and can be configured to generate an event when the clock reaches a specified time stamp. This event is routed to the MCU's event system which allows it to directly create pulse by toggling a GPIO pin. This is done in hardware and requires no CPU and software intervention. To generate periodic PPS signals, the event also triggers an interrupt service routine in the MCU where the timestamping hardware is configured to generate a new event at the timestamp of the next PPS signal.

Similarly, to evaluate the accuracy of the synchronization between the CTRIE and the FEC's TSU, the I210 NIC is also configured to output PPS signals on one of its programmable IO pins.

# Chapter 8

## Preliminary System Results

Having completed the development and testing environment, this chapter aims to document the synchronisation functionality and performance of the setup through a rudimentary test.

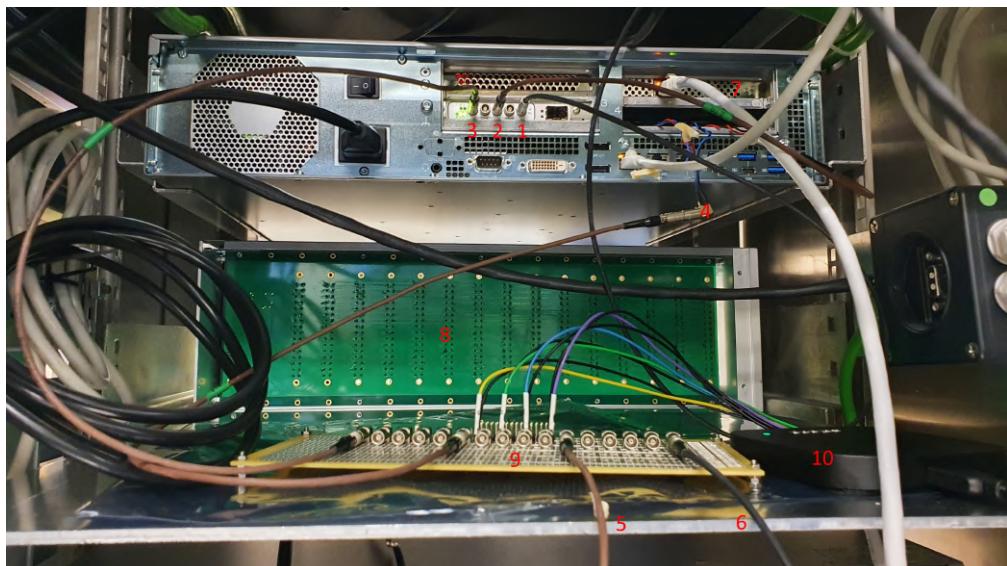
### 8.1 Setup

To determine the performance of the system, two data acquisition cards were connected to the FEC through a switch, as seen in figure 8.1. Note that due to sourcing issues, the SFP transceiver required to connect the FEC and acquisition board to the Juniper QFX 5110 switch were unable to arrive in time for the test. As a substitute the test was performed using a commercial HP 1420 switch (thus acting as a Non PTP device in the network). PTP synchronisation is performed at one hertz, meaning that the `ptp4l` application is configured to send sync packets every second. To measure the offsets between the different equipment, each component was configured to output 10PPS signals, i.e. 10 signals per seconds aligned to the beginning of each second. These signals were measured by a Saleae Logic Pro 16 logic analyser. Data was captured for a duration of 653 seconds.

## 8.1. Setup



**Figure 8.1:** Image of the front of the test setup. The FEC is placed in the top rack, the backplane and associated acquisition boards are in the middle position, and the QFX5110 switch is in the bottom position. The temporary substitute switch is placed on top of the QFX 5110.

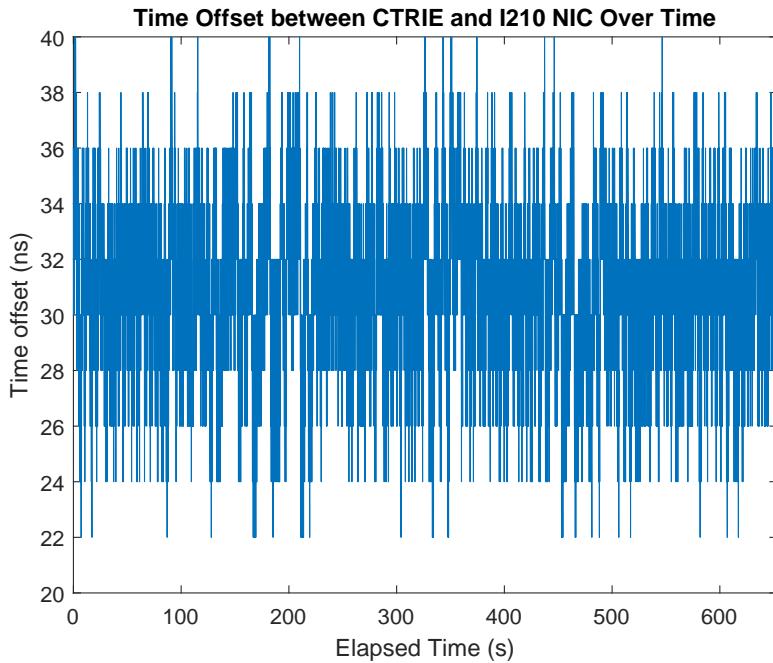


**Figure 8.2:** Image of the back of the test setup. The annotations have the following meaning: #1 cable with central timing going into the CTRIE. #2 cable transmitting the 1PPS signal used to synchronise the I210 to the CTRIE. #3 cable transmitting the 10PPS signal from the CTRIE to the logic analyser. #4 cable transmitting the 10PPS signal from the I210 to the logic analyser. #5, #6 cables transmitting the 10PPS signal from the acquisition boards to the logic analyser. #7 I210 NIC. #8 Rear side of the backplane. #9 Board used to convert coaxial LEMO connectors to the logic analyser's connectors. #10 Saleae Logic Pro 16 logic analyser.

## 8.2 Synchronisation Accuracy between CTRIE and I210 NIC

### 8.2 Synchronisation Accuracy between CTRIE and I210 NIC

The synchronisation performance between the CTRIE and the I210 NIC in the FEC is illustrated in figure 8.3 as a graph that plots the time-offset of the measured PPS signals over time. As seen in the graph, the I210 pulses are between 22 ns and 40 ns ahead of the CTRIE pulses throughout the duration of the test. This is an excellent result as it leaves the PTP synchronisation between the I210 and data acquisition boards with an offset margin of 960 ns, still well within the 1  $\mu$ s requirements. It indicates that the ts2phc program works well as a synchronisation tool.



**Figure 8.3:** Graph of the offset between the clocks of the I210 NIC and the CTRIE.

### 8.3 Synchronisation Accuracy between CTRIE and Data Acquisition Boards

The synchronisation performance between the CTRIE and data acquisition boards achieved during the test is shown in figure 8.4 and figure 8.5. At a first glance, these results are less satisfying than the ones between the CTRIE and the FEC. The offset between acquisition card #1 varies wildly for the first 200 seconds, but then settles at around a  $-2 \mu$ s offset, with exception of the occasional anomaly. The second card performed worse. It

### 8.3. Synchronisation Accuracy between CTRIE and Data Acquisition Boards

settled at around a  $-12\text{ }\mu\text{s}$  offset, while also experiencing anomalies where the offset would drastically change. At one point towards the end of the measurements, the offset momentarily jumped to over  $100\text{ }\mu\text{s}$ .

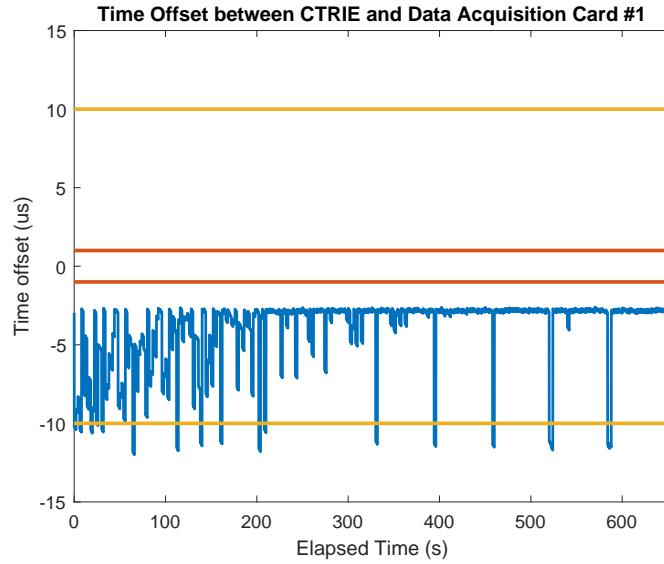
To obtain a better understanding of the PTP synchronisation behaviour, the offset of both acquisition boards was plotted on the same graph, and a representative sample of the measurement duration has been highlighted in figure 8.6.

It is evident from this plot that the synchronisation behaves as a saw-tooth like pattern, where the offset is discretely decreased every second, while slowly increasing in between each second. This is in line with the expectations, as in the current PTP implementation the time in the data acquisition board's clock is corrected according to the offset calculated from the received PTP packets.

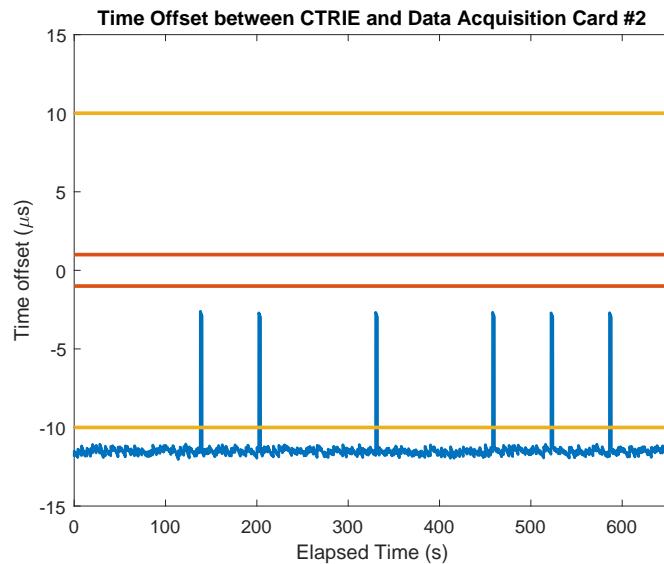
However, outside of the general shape of the plots' waveform, there are several discrepancies within the acquired data. Firstly, it is unexpected that the offset does not jump closer to 0 whenever PTP packets are received every second. This indicates that the boards are unable to calculate the time offset with great precision. Secondly, it is curious why the two cards perform so differently and exhibit such different synchronisation performance. Thirdly, both cards experienced anomalies where the offset suddenly changes radically for one or two PTP packet receptions. In some cases, these anomalies are correlated and occur at both cards at the same time. Currently it is unknown where in the system or why these errors occur. This will be a topic of future testing and development.

It was never, however, expected the required synchronisation performance would be achieved at this stage in the project. Thus, the results are interpreted as positive since they clearly show that the constructed system has merit to enable future developments.

### 8.3. Synchronisation Accuracy between CTRIE and Data Acquisition Boards

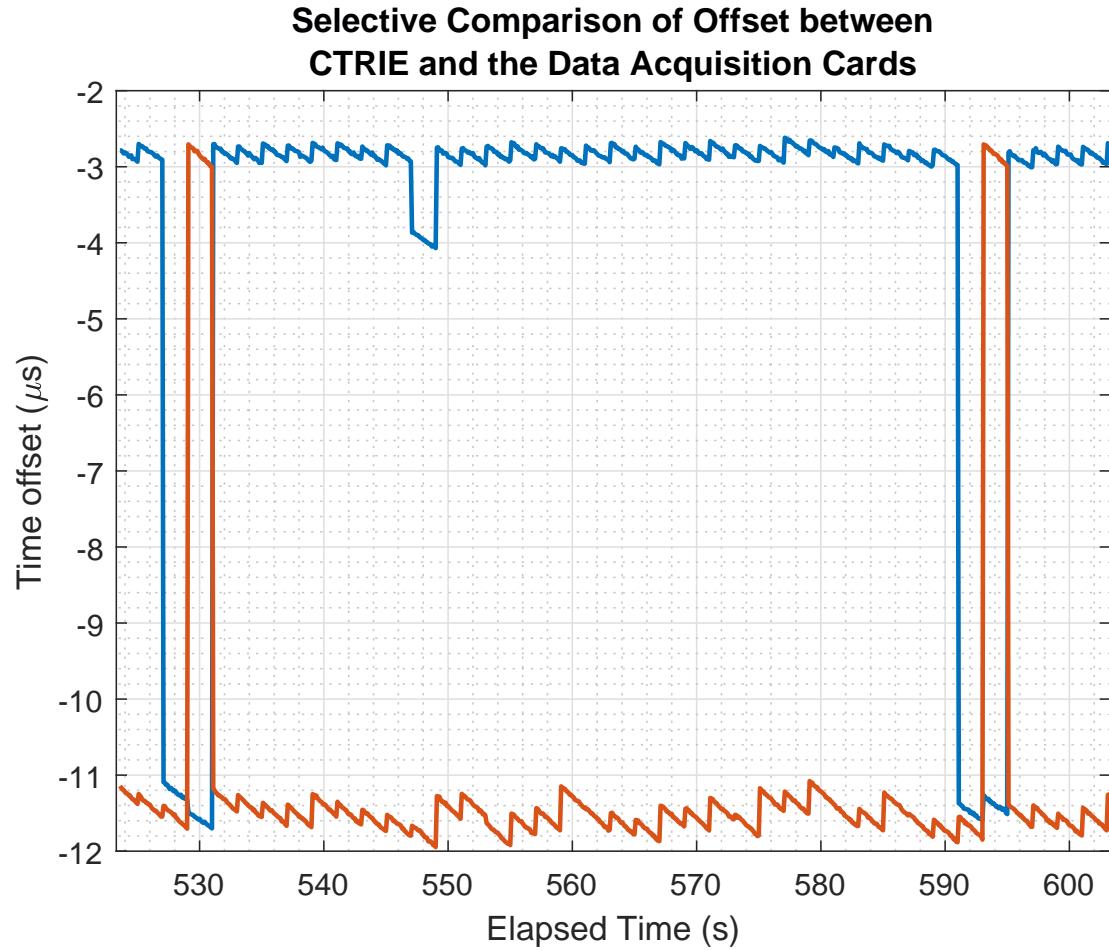


**Figure 8.4:** Graph of the offset between the clocks of data acquisition card #1 and the CTRIE. The area enclosed by the lines in ■ represent the maximum offset of  $\pm 10\text{ }\mu\text{s}$  that must be guaranteed in the final deployment. Similarly, the area enclosed by the lines in ■ represent the desired offset of  $\pm 1\text{ }\mu\text{s}$ .



**Figure 8.5:** Graph of the offset between the clocks of data acquisition card #2 and the CTRIE. The area enclosed by the lines in ■ represent the maximum offset of  $\pm 10\text{ }\mu\text{s}$  that must be guaranteed in the final deployment. Similarly, the area enclosed by the lines in ■ represent the desired offset of  $\pm 1\text{ }\mu\text{s}$ .

### 8.3. Synchronisation Accuracy between CTRIE and Data Acquisition Boards



**Figure 8.6:** Graph of the offsets between the data acquisition cards the CTRIE. The data points between 523 and 603 seconds are shown. The plot in ■ is acquisition card # 1 and the plot in □ is card # 2.

## Chapter 9

# Summary and Conclusion

This interim report documents the progress on the synchronisation aspect of the new data acquisition system for the LHC’s quench protection system. The documentation period spans from September 2020 to December 2020. At the end of this period, I was expected to have acquainted myself with the context and technologies associated with the project. This is achieved. Furthermore, to ease future development the construction of proper testing and development environment has also been a major priority. This report documents all these aspects and demonstrates that the constructed system(s) can support future development and testing needs in a variety of ways. This forms a solid foundation for the next stage of the project, and I am confident that the goal of initiating field testing in June 2021 is attainable.

In the immediate future the next steps are to improve the PTP implementation in the data acquisition boards by working on new synchronization techniques.

# Chapter 10

## Personal Reflection

Working at CERN has been and is an amazing opportunity. It has been a fun, educational, and at time an overwhelming experience. Before interning at CERN, I have only experienced practical engineering work in my university projects and as part of a part-time job at a start-up in Aalborg. Due to the sheer size and complexity of the work undertaken at CERN, there were many notable differences between these past experiences and my experience working at CERN. Centrally of which is that instead of charged with the responsibility of an entire project, at CERN my contributions make up a minuscule part of a much larger system of which I only have a superficial understanding of. On one hand, I suspect that this at times can be a bit dull, as projects tend to be extremely focused and narrow in scope. However, on the other hand, I also feel a great satisfaction in contributing to an endeavour that aims to push the understanding of our universe to the limit.

On a technical level, the size of the organisation meant that a lot of my work has been dependent on other CERN technologies. As a result, it took me quite a while to be fully integrated in the work process. Another consequence of this, is that most of my solutions had to be designed to be compatible with the existing infrastructure and equipment at CERN. This was very educational, as it constrained my work in a way that I have not been used. The vastness of CERN also has its benefits, as experts and support personnel of all kinds were at my disposal to help simplify development. For example, when I was doing PCB design, I would send my component datasheets to another department which would create the component schematic symbols and footprints for me.

From a purely academic point of view, the stay has been less satisfying than I would have preferred. So far, my project has primarily focused on practical engineering work where development and system integration has been the primary tasks. I had little time to produce unique technical solutions, or to immerse myself in deep theoretical topics. I hope that there will be more opportunities for this during the remainder of the stay.

Socially, the stay has been limited by various restriction put in place due to the current COVID-19 pandemic. In spite of this I have made many great friends during my stay and have had several opportunities to explore the natural beauty of the local area.



**Figure 10.1:** Photo of me in the LHC tunnel.

# Bibliography

- [1] CERN, “Electronics for Protection.” [Online]. Available: <https://mpe.web.cern.ch/electronics-protection>
- [2] ——, “Our Mission.” [Online]. Available: <https://home.cern/about/who-we-are/our-mission>
- [3] ——, “Accelerators.” [Online]. Available: <https://home.cern/science/accelerators>
- [4] ——, “The Large Hadron Collider.” [Online]. Available: <https://home.cern/science/accelerators/large-hadron-collider>
- [5] Unknown, “Untitled Image of LHC taken from the air. Retrieved from "CMS Winter Results at Fermilab Wine & Cheese Seminar 2017" by Jim Hirschauer.”
- [6] S. Charley, “Inside the Large Hadron Collider | symmetry magazine,” 2018. [Online]. Available: <https://www.symmetrymagazine.org/article/inside-the-large-hadron-collider>
- [7] CERN, “Accelerating: Radiofrequency cavities.” [Online]. Available: <https://home.cern/science/engineering/accelerating-radiofrequency-cavities>
- [8] ——, “Pulling together: Superconducting electromagnets.” [Online]. Available: <https://home.cern/science/engineering/pulling-together-superconducting-electromagnets>
- [9] M. Ferrario and B. J. Holzer, “Introduction to Particle Accelerators and their Limitations,” in CAS-CERN Accelerator School: Plasma Wake Acceleration, vol. 001, no. November 2014. CERN, 2014, pp. 29–50.
- [10] CERN, “The accelerator complex.” [Online]. Available: <https://home.cern/science/accelerators/accelerator-complex>
- [11] ——, “Linear accelerator 4.” [Online]. Available: <https://home.cern/science/accelerators/linear-accelerator-4>
- [12] ——, “How a detector works.” [Online]. Available: <https://home.cern/science/experiments/how-detector-works>

## Bibliography

- [13] E. Mobs, “The CERN accelerator complex - 2019. Complexe des accélérateurs du CERN - 2019,” 2019. [Online]. Available: <https://cds.cern.ch/record/2684277>
- [14] CERN, “Cryogenics: Low temperatures, high performance.” [Online]. Available: <https://home.cern/science/engineering/cryogenics-low-temperatures-high-performance>
- [15] R. Schmidt, R. Assmann, E. Carlier, B. Dehning, R. Denz, B. Goddard, E. B. Holzer, V. Kain, B. Puccio, B. Todd, J. Uythoven, J. Wenninger, and M. Zerlauth, “Protection of the CERN large hadron collider,” *New Journal of Physics*, vol. 8, 2006.
- [16] M. Wilson, “Superconducting Magnets for Accelerators. Lecture 2,” 2008.
- [17] CERN, “Large Energy Extraction Resistors.” [Online]. Available: <https://wikis.cern.ch/display/MPEEP/Large+Energy+Extraction+Resistors>
- [18] “CERN CentOS 7 - Linux @ CERN.” [Online]. Available: <https://linux.web.cern.ch/centos7/>
- [19] “"Big Bang" collider repairs to cost up to \$29 million | Reuters.” [Online]. Available: <https://www.reuters.com/article/us-cern-repair-idUSTRE4B42F420081205>
- [20] P. Praczyk, “Machine Protection & Electrical Integrity INTERNAL NOTE Data acquisition solutions for QPS,” no. June, 2020.
- [21] T. Podzorny, “Ethernet data acquisition,” CERN, Tech. Rep., 2020.
- [22] “Worldfip · Wiki · Projects / CernFIP · Open Hardware Repository.” [Online]. Available: <https://ohwr.org/projects/cern-fip/wiki/WorldFIP>
- [23] “Home · Wiki · Projects / CernFIP · Open Hardware Repository.” [Online]. Available: <https://ohwr.org/project/cern-fip/wikis/home>
- [24] IEEE Standards Association, P802.3cg, 2019. [Online]. Available: [https://standards.ieee.org/standard/802\\_3cg-2019.html](https://standards.ieee.org/standard/802_3cg-2019.html)
- [25] T. Ossendrijver, “Design of the QPS crate controller Data acquisition board,” no. January, pp. 1–51, 2020.
- [26] R. Zarick, M. Hagen, and R. Bartoš, “Transparent clocks vs. enterprise ethernet switches,” IEEE International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication, ISPCS, pp. 62–68, 2011.
- [27] Hewlett-Packard, “Fundamentals Quartz Oscillators,” Hewlett-Packard, Tech. Rep., 1997.
- [28] J.-L. Ferrant, M. Gilson, S. Jobert, M. Mayer, L. Montini, M. Ouellette, S. Rodrigues, and S. Ruffini, Synchronous Ethernet and IEEE 1588 in Telecoms. John Wiley & Sons, 2013.

## Bibliography

- [29] “How a PTP slave syncs with a PTP master - YouTube.” [Online]. Available: [https://www.youtube.com/watch?v=Forh3XfD\\_Ec](https://www.youtube.com/watch?v=Forh3XfD_Ec)
- [30] T. Committee, IEEE Std 1588-2008, 2008, vol. 2008, no. July.
- [31] J. C. Eidson, Measurement, Control, and Communication Using IEEE 1588, 2006.
- [32] G. Giorgi and C. Narduzzi, “Performance analysis of Kalman-filter-based clock synchronization in IEEE 1588 networks,” IEEE Transactions on Instrumentation and Measurement, vol. 60, no. 8, pp. 2902–2909, 2011.
- [33] D. R. Jeske, “On maximum-likelihood estimation of clock offset,” IEEE Transactions on Communications, vol. 53, no. 1, pp. 53–54, 2005.
- [34] “Central Timing - Timing & Sequencing - Controls Wikis.” [Online]. Available: <https://wikis.cern.ch/display/TIMING/Central+Timing>
- [35] R. Voirin, “Introduction to Front-End computers.”
- [36] Siemens, “Image of SIMATIC IPC647E.” [Online]. Available: <https://new.siemens.com/global/en/products/automation/pc-based/simatic-rack-ipc.html#SIMATICIPC647E>
- [37] CERN, “Top Down image of CTRIE.” [Online]. Available: <https://wikis.cern.ch/display/HT/CTRIE++PCI+express+GMT+Timing+Receiver>
- [38] ———, “Front image of CTRIE.” [Online]. Available: <https://wikis.cern.ch/display/HT/CTRIE++PCI+express+GMT+Timing+Receiver>
- [39] B & H Foto & Electronics Corp, “Image of HP Intel Ethernet I210-T1 GbE Network Interface Card.” [Online]. Available: [https://www.bhphotovideo.com/c/product/1024193-REG/hp\\_e0x95aa\\_intel\\_etherne\\_i210\\_t1\\_gbe.html](https://www.bhphotovideo.com/c/product/1024193-REG/hp_e0x95aa_intel_etherne_i210_t1_gbe.html)
- [40] Juniper, “Image of Juniper qfx5110-48s.” [Online]. Available: <https://www.juniper.net/us/en/products-services/switching/qfx-series/qfx5100/>
- [41] Symmetricom, “IEEE 1588 Precise Time Protocol: The New Standard in Time Synchronization,” Tech. Rep., 2005. [Online]. Available: [http://www.symmetricom.com/media/files/secure/white-papers/wp\\_IEEE\\_1588.pdf](http://www.symmetricom.com/media/files/secure/white-papers/wp_IEEE_1588.pdf)
- [42] R. Cochran, “The Linux PTP Project.” [Online]. Available: <http://linuxptp.sourceforge.net/>