Valtteri Mäntymäki

Time synchronization of multiple biosignals using software interface

Metropolia University of Applied Sciences
Bachelor of Engineering
Software engineering
Bachelor’s Thesis
25 October 2022

Abstract

Author:	Valtteri Mäntymäki
Title:	Time synchronization of multiple biosignals using software interface
Number of Pages:	27 pages
Date:	2 December 2022
Degree:	Bachelor of Engineering
Degree Programme:	Information and Communications Technology
Professional Major:	Software engineering
Supervisors:	Senior Lecturer, Sakari Lukkarinen Senior Scientist, PhD, Kati Pettersson

The purpose of this thesis was to time synchronize different biological measuring devices using a software interface. In this case the software interface is written in C# and the devices that are synchronized consist of electrocardiogram (ECG), galvanic skin response (GSR) and eye tracking. Measurements from these devices are collected with Triers Social Stress Test (TSST) that is used to induce stress reaction in test subjects. Measurements from TSST should be accurate enough that the onset and offset of the reactions match the stimuli created by the TSST.

Work for this thesis was divided into three sections, research, testing and implementation. At the research stage the plan was to gather a sufficient understanding on overall time synchronization and related topics. After reading the general concepts and some additional documentation about the devices the testing phase could be started. The testing phase comprised of testing the actual devices used and checking that they work and produce data that is expected. Also, a quick recap period on C# programming basics was required for the building of the interface. The last stage was meant to be the implementation phase where application interface would have been built to make sure that the delay is calculated and processed into the timestamp data of the device either at the time of the recording or at post processing.

Unfortunately, the code base for the devices C# library was full of undocumented code and due to time constraints, the writing of the interface could not be made.

An alternative solution was written with Python programming language to show case a solution for simple time delay calculation. This solution show cases a delay caused by the devices start and calculates the delay in post processing step into the time stamp data.

Keywords: time synchronization, biosignals, application programming interface

Tiivistelmä

Tekijä:	Valtteri Mäntymäki
Otsikko:	Usean biologisen mittauslaitteen aikasynkronisointi ohjelmisto rajapinnan avulla
Sivumäärä:	27 sivua
Aika:	2.12.2022
Tutkinto:	Insinööri (AMK)
Tutkinto-ohjelma:	Tieto- ja viestintätekniikka
Ammatillinen pääaine:	Ohjelmistotuotanto
Ohjaajat:	Lehtori, Sakari Lukkarinen Erikoistutkija, FT, Kati Pettersson

---

Tämän opinnäytetyön tarkoituksena on aikasynkronoida erilaisia biologisia mittalaitteita ohjelmistorajapinnan avulla. Tässä tapauksessa ohjelmistorajapinta on kirjoitettu C#:lla. Synkronoidut laitteet koostuvat elektrokardiogrammista (EKG), galvaanisesta ihovasteesta (GSR) ja silmien seurannasta. Näiden laitteiden mittaukset kerätään Triersin sosiaalisella stressitestillä (TSST), jota käytetään stressireaktion aiheuttamiseen koehenkilöillä. TSST:n mittausten tulee olla riittävän tarkkoja, jotta reaktioiden alkamisen ja loppumisen ajat vastaavat TSST:n luomia ärsykkeitä.

Opinnäytetyön työmäärä on jaettu kolmeen osaan: tutkimus, testaus ja toteutus. Tutkimusvaiheen tarkoitus oli saavuttaa yleinen ymmärrys aikasynkronisaatioon liittyvistä aiheista ja yleisistä konsepteista. Kun jonkinlainen ymmärrys oli saavutettu lukemalla, voitiin siirtyä testausvaiheeseen. Testausvaiheessa käytiin läpi laitteiden toimivuus sekä niiden tuottaman datan oikeellisuus. Lisäksi testausvaiheessa tarvittiin jakso, jossa käytiin läpi C# ohjelmoinnin nopea kertaus ja yleiset konseptit. Lopuksi olisi tullut toteutusvaihe, jossa olisi luotu ohjelmistorajapinta, jonka avulla viive olisi laskettu ja lisätty mitattuihin aikaleimoihin joko mittauksen aikana tai mittauksen jälkeen.

Valitettavasti laitteiden C # -kirjaston koodipohja oli täynnä dokumentoimatonta koodia mikä aiheutti sen, että koodin kirjoittamiseen olisi kulunut erittäin paljon aikaa. Tämä ei olisi ollut mahdollista projektin antamassa aikataulussa. Python-ohjelmointikielellä luotiin vaihtoehtoinen esimerkkiratkaisu, joka esittää yksinkertaisen ratkaisun aikaviiveen laskentaan. Tämä ratkaisu näyttää laitteiden käynnistymisen aiheuttaman viiveen ja laskee jälkikäsittelyvaiheen viiveen aikaleimatietoihin.

Avainsanat: aikasynkronisaatio, biologinen signaali, ohjelmistorajapinta

Contents

List of Abbreviations

ADC: Analog-to-digital. Conversion where analog signals are converted into digital signals.

BCG: Ballistocardiogram. Measure of ballistic forces generated by hear beat.

DAC: Digital-to-analog. Conversion where digital signal is converted into analog signal.

ECG: Electrocardiogram. A record of hearts electrical activity.

EOG: Electrooculography. A method to record eye movements using electrodes placed around the eye.

GPS: Global positioning system. A satellite system used for locations services and time synchronization.

GSR: Galvanic skin response. Electricity measured change in skins ability to conduct electricity.

HR: Heart rate. Typically measured in beats per minute.

HRI: Human robot interaction. A research field that focuses on humans and robots' coordination and co-operation.

HRV: Heart rate variability.

IoT: Internet of things. A shorthand used to describe all the devices connected to the Internet. Ranging from lightbulbs to servers.

IS: International second. Scientifically standardized measurement of a second.

LSL: Labstreaminglayer. Software library used for time synchronization.

MATB-2: Multi-Attribute Task Battery 2. Test developed by NASA to test human performance under workload.

NTP: Network time protocol. Time synchronization algorithm used to synchronize network devices clocks.

PCCR: Pupil center corneal reflection. A reflection that humans eye gives when a near-infrared light is flashed at it.

PPG: Photoplethysmography. Optical technique to measure changes in humans blood volume.

QRS: QRS-complex. A shorthand term used to describe a part of electrocardiogram.

SCL: Skin conduction level. Skin’s ability to conduct electricity over long period of time.

SCR: Skin conduction response. Skin’s ability to conduct electricity over short period of time.

TAI: Temps atomique international. High precision atomic time standard.

TSST: Trier social stress test. Test to induce a stress reaction on test subjects.

UDP: Communication protocol used to send messages.

UT: Universal Time. A time standard based on earth’s rotation.

UTC: Coordinated universal time. Time measurement unit that is used to globally to keep record of time

VR:	Virtual reality.

1 Introduction

The Technical Research Centre of Finland (VTT) is conducting a research where they study acute stress reaction. The purpose is to induce acute stress reaction using VR and measure the reaction using GSR (Galvanic Skin Response), ECG (Electrocardiography) and Eye-tracking technology.

The study will comprise of a baseline period to measure person baseline signals, after that two or three controlled stress reaction tests using VR and one game/exercise simulating real world to induce stress reaction.

The problem this paper tries to solve is how to get multiple biological measuring devices working synchronously without a large time delay using a software interface. The focus of this project will be to synchronize time between the different biological measuring devices in a single program interface so the signals would be more easily managed and maintained.

Overall, time synchronization is a broad topic with a lot of research on dependent solutions that relies solely on the implementation of the conducted experiment. Therefore, it is difficult to find literature concerning time synchronization in software form.

The thesis is structured as follows: In the second chapter the biosensing devices are presented, the third chapter describes the Trier Social Stress Test (TSST) and Trier Social Stress Test- Virtual Reality (TSST-VR) experiments that induces acute stress reactions for a user. Chapter four contains overview of time synchronization and different solutions for different research setups. In the chapter five the summary of the thesis project is provided, while in the chapters six and seven conclusions and discussion of the thesis work are presented.

2 Devices

In this chapter the measurement devices and the signals they produce, and the actual devices used in this thesis and their properties are discussed. There are three devices in total that measure heart rate, skin electrical conduction and eye-movement.

2.1 Electrocardiogram

Electrocardiogram (ECG) is a biosignal used to measure the electrical activity of the heart. The QRS complex is a part of hearts electrical activity. Heart beats consist of five parts P wave that comes before QRS complex and the QRS complex that is divided into Q wave, R wave, S wave and finally T wave. After this the beat start over and repeats. [1, p. 7-8]

P wave is the initial starting point of the heartbeat. It represents the depolarization of the atrial. This means that the top part of the heart is contracting, taking in blood, and relaxing. It increases the electrical potential in the measurement. This is the first small rise that can be seen in figure 1. [1, p. 7]

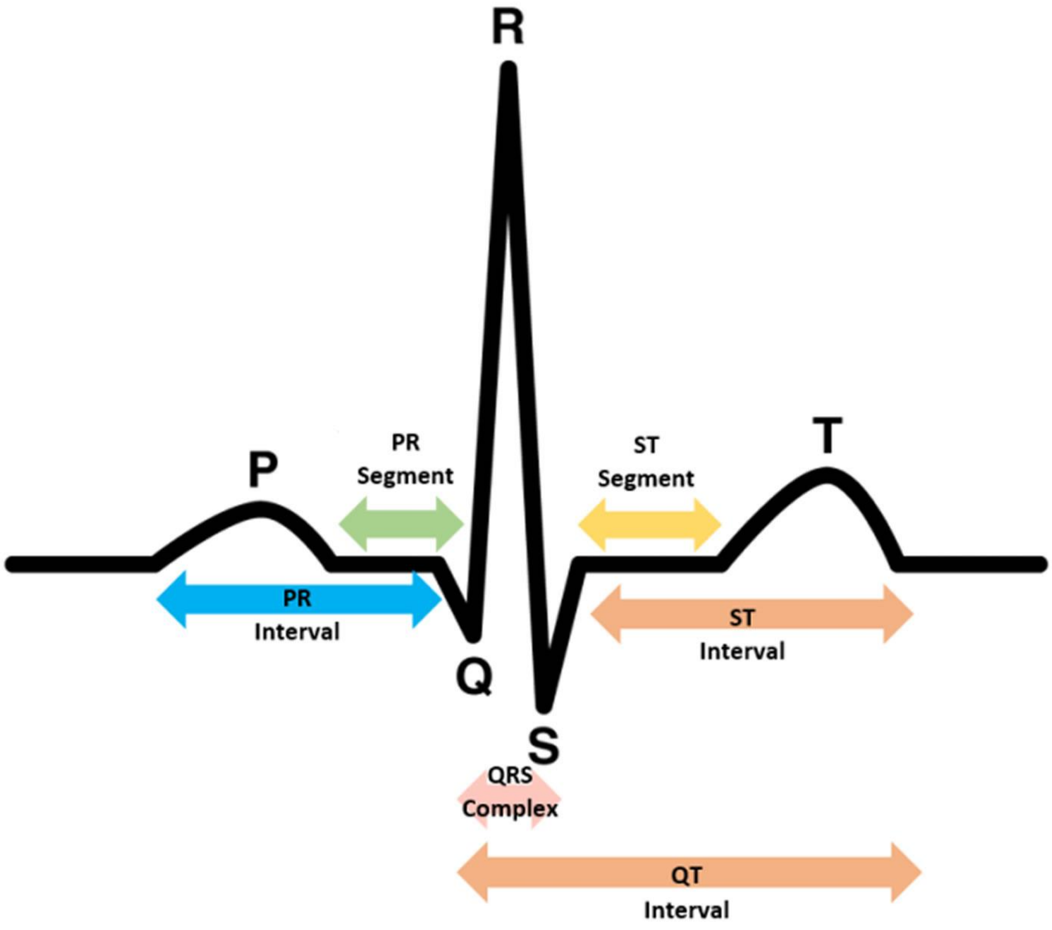


Figure 1. Graphical illustration of normal components of electrocardiogram. [1, s.7]

Q wave is small little dip illustrated in figure 1. Electrical signal dips because of the depolarization in the interventricular septum that is the wall separating the two ventricles. Now that the heart is full of blood, we get to R wave. [1, p. 7]

R wave is repolarization of heart. Because the ventricles are the largest part of the heart electrical signal increase is large. In R wave the first heart sound is heard as ventricles and atria valves close.

After the R peak the ECQ signal get down to the S wave. As the signal decreases and depolarizes the heart is pushing blood out as new blood is coming in. This causes the signal fall below the base line.

Finally, T wave is the part where the hearts electrical activity resets as it starts a new cycle. At the end of contraction, when the release begins aorta and pulmonary valves close creating the second thud.

There are different types of ECG devices mostly they consist of ECG-unit, electrodes, and cables. Components of a typical ECG signal setup that records frontal heart activity contains four electrodes. There are also tree different types of leads bipolar, augmented, or unipolar. They all take a different perspective on hearts electric activity. Placement of the electrodes is well standardised. It is likely that the device has four electrodes that are placed on chest or at each limbs end. A typical parameter is measured from ECG is Heart Rate (HR). It is merely heart beats over a certain time interval, usually beats per minute (bpm). Another important parameter that can be measured from ECG is IBI (Inter-Beat Interval). IBI is millisecond (ms) measure of each individual heartbeat. From IBI you can also get HRV (Heart Rate Variability). HRV is variation calculations of IBI from beat to beat. [2]

In this thesis Shimmer3 ECG unit was used to measure ECG and heart rate. The sensor is shown in Figure 2. It is easy to use and its portable due to its low power consumption and small size. Because of this, the test subject can move more freely and is able to perform more varying tasks.[3]

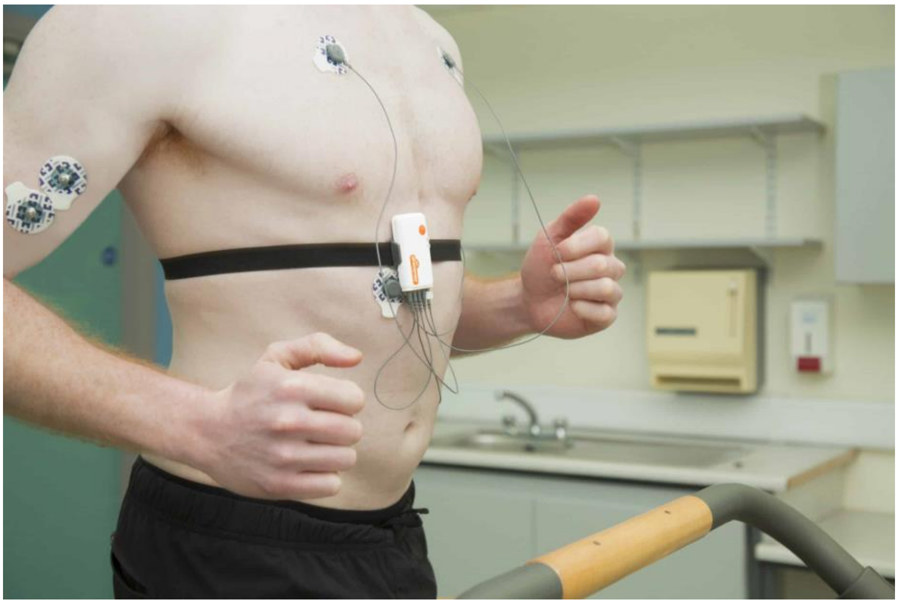


Figure 2. Shimmer ECG unit in chest measurement setup. [3]

2.2 Galvanic Skin Response

Galvanic Skin Response (GSR) measures the skin conduction in volts that changes depending on skin’s sweat glands. When sweat glands activate due to stimulus which leads to skins conduction increase, the electrical charge can pass true more freely. This electrical difference in potential charge is usually measured in microvolts (uV). [5, p. 6]

Theory behind sweating is that humans are emotion sweaters and the amount changes due to sympathetic nervous system. Sympathetic nervous system oversees “fight or flight” reaction. This is something we cannot control and therefor is an ideal research signal for human behavior. It is used in psychological research, clinical research, psychotherapy, and consumer neuroscience. [5, p. 7]

Most typical use case to measure GSR is related to human’s stress behavior. There is an interesting article on GSR where they reviewed studies on biofeedback. Biofeedback means that the biological signal measured from the test person is given back to the person in audio or visual way for the person to control this signal. If the person is subjected to this type of training, it allows the person to control the signal and therefor decrease the possible harmful effects the signal might have. In the article it was tested on epileptic patients that showed clear signs of reduced seizure frequency. [4, p. 4]

For measuring GSR only two electrodes are needed. Usually, they have some form of coating that has a good electric conductor, like Ag/AgCI (silver/silver-chloride). The electrodes are attached to skin. Some usual contact places are middle and index finger or both sides of the hand or the foot where the electrodes are placed into inner side of the foot. The finger placement can be seen in figure 3. [5, p. 21]

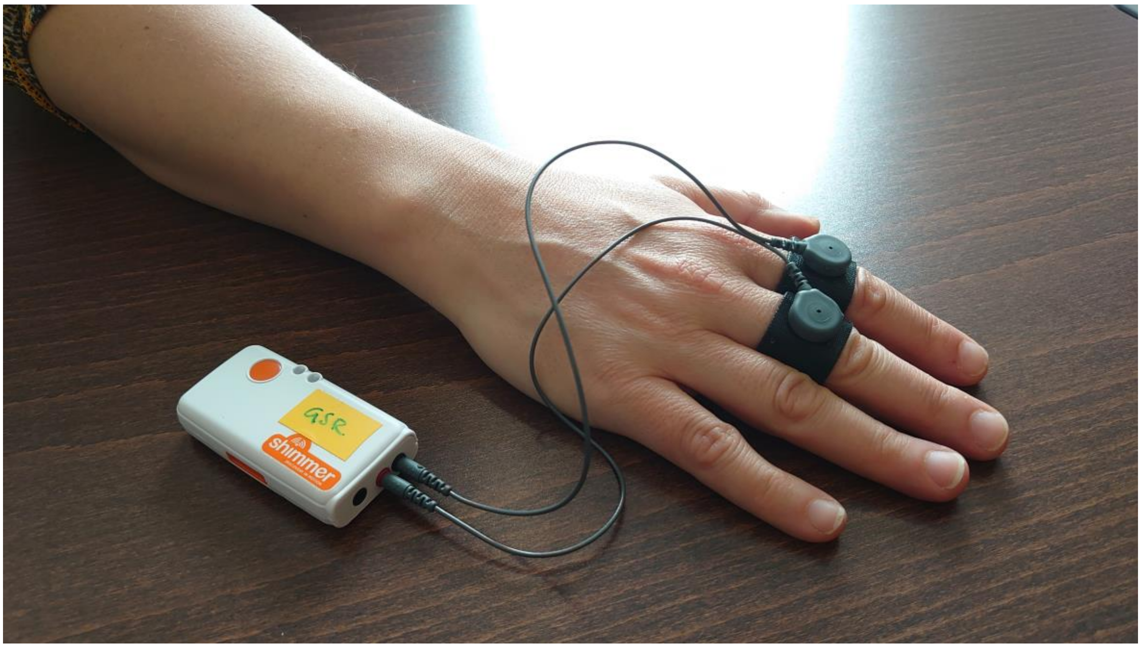


Figure 3. GSR placement with Shimmer3 GSR+ unit.

GSR produces two main signals. Skin conductance level (SCL) is slow varying in time interval between 10 seconds to one minute. SCL is dependent on the individual’s hydration, skin dryness, and autonomic regulation, which is not ideal on research point of view, because it introduces variation that is not controllable by the test subject. The other signal is SCR (Skin Conductance Response). SCR is much faster in response to stimulus and can be seen on top of SCL as high peaks. These peaks occur between 1 – 5 seconds after stimulus. SCR peaks are also known as SCR bursts. [5, p. 21]

One common type of data usually measured from the GSR is event-related SCRs. It is tied to an event purposefully planed for the experiment. When this certain stimulus happens there is four metrics that can be taken from SCR. [5, p. 25] Figure 4 demonstrates these metrics and how the usual SCR goes over time.

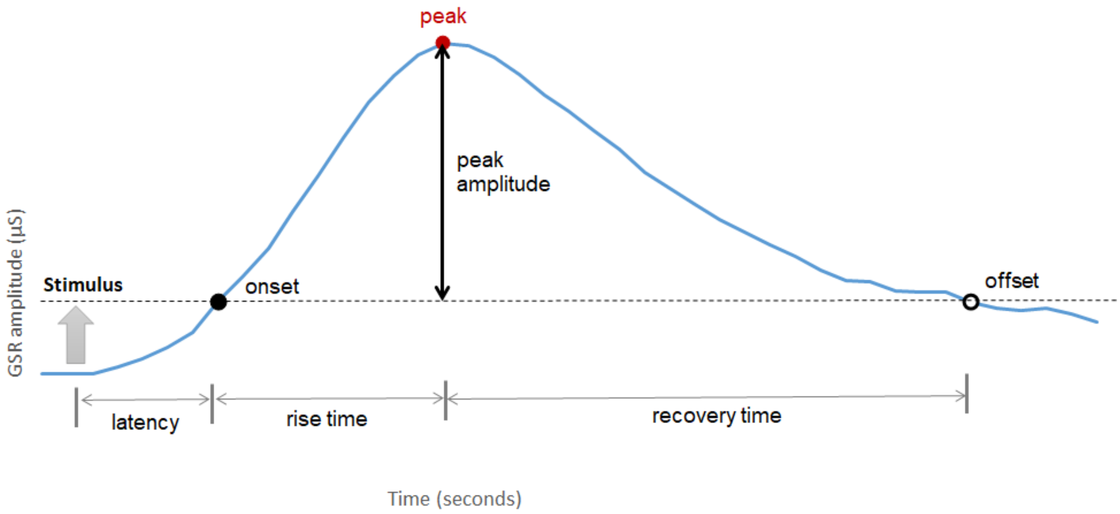


Figure 4: Graphical presentation of GSR metrics. [5, p. 6]

Four different stages in the Figure 4:

- *Latency* starts from stimulus onset to onset of SCR burst
- *Peak amplitude* is the difference between onset and peak.
- *Rise time* is the duration from onset to peak
- *Recovery time* that is measured from peak to offset of the stimulus

In this thesis Shimmer3 GSR+ Unit is used for GSR measurements. It is from the same manufacturer as the ECG unit. It comes whit the central charging unit that also lets to specify the measuring details for the device.

2.3 Eye-tracking

Mechanism for eye tracking can be measured using pupil Centre Corneal Reflection (PCCR). A near-infrared light is aimed at the centre of the pupil and the reflection that comes back from the eye is tracked by a camera. This type of eye tracking is called video-oculography. [6, p. 5; 7]

There are couple of different types of eye trackers. One is screen based where the eye tracker is placed under the screen that is to be tracked. Then there are wearable glasses, either ones that look more like regular glasses or ones that come with a VR- head set like shown in figure 5. [6, p. 6]

Figure 5. A Vario 3 VR-headset with eye-tracking system.

Use cases for eye tracking technology vary. Usually, they are coupled with other measuring devices to get better understanding of the research topic. One example of this is the cognitive load detection from wearable sensors by Tervonen & Pettersson et.al. They measured HR, heart rate variability, galvanic skin response and skin temperature to train a model that detects persons cognitive load in an ultra-short time window (30s or less). The recording device used for eyes movement in this study was electrooculography (EOG). [8]

Most used eye tracking metrics and parameters are fixation and gaze points. Eye tracker captures data by certain intervals and each data capture is on point. If enough points are concentrated in certain area, we can call it fixation point. Each fixation point means the eye stayed in that location long enough. Usually this is between 100 to 300 milliseconds. [6, p. 13]

Other interesting metrics include heat map where the eyes gaze points are turned into colors where different colors represent the amount the area is looked at.

Areas of interest (AOI) are also a parameter that is collected. AOI is defined that is of some special interest for the test. After that the number of gazes is calculated that tells how many times the test subject looked at the AOI.

3 Trier Social Stress Test

Trier Social Stress Test (TSST) is a standardized laboratory tool that has shown reliably induce acute stress reaction in humans. This is because the test stimulates high socio-evaluative threat and uncontrollability of the situation. [8, p.1]

TSST is a speech task that simulates an interview scenario where tree actors are listening to test person’s speech but are refrained on giving the person any feedback. During the speech the person is also required to do a mentally taxing arithmetic calculation that adds the amount of stress in the test person. [10, p. 859]

3.1 Trier Social Stress Test – Virtual Reality

Trier Social Stress Test in Virtual Reality (TSST-VR) is the same as basic TSST, but it is implemented in virtual reality where the participant is wearing a VR headset and doing the TSST test in a controlled virtual environment.

Test on viability of TSST-VR is showing that it is possible to induce stress reaction on test subjects via VR. Liu & Zhang tested sex differences in the TSST-VR, and it showcased the validity of the test, and induced stress using the VR setting. [10, p. 859]

One of the upsides of TSST in a VR setting is that the experimenters that sit at the table and listen to the participant are computer generated so they can be set to behave similarly every time the experiment is conducted. This ensures that the experiment can be replicated, and the experimenter’s human errors and emotional effect can be subtracted from the study. [10, p. 860]

4 Time synchronization

There is a lot of variations in time synchronization solutions that give different results depending on the use case. There is no one size fits for all solution available. Because of this the chapter on time synchronization are divided in a smaller subcategory.

First, overall perspective of time keeping is introduced and some problems that come with it, then a deeper analysis how computers keep track of time is given. Common algorithms used to synchronize clocks are explained, and lastly a use case examples that have been used in different types of solutions for these problems are introduced. The use cases have been selected based on this thesis work in mind.

4.1 Time in general

Humans have developed time measurement methods over the course of our history. First thing that comes to mind is the obvious cycles of day and night and seasons of the year that people started to calculate. Early time keeping was mostly done by means of astronomy. At same time when clocks started to also evolve astronomers noticed that the time of day varied over the seasons, so they needed more accurate ways of calculating the passage of time.

The first group of time scales were universal times (UT0, UT1, and UT2). These were time scale systems that would increase in accurate as the numbers went higher. UT0 is measured from a perfect clock that is corrected at exactly at noon when the sun is at Greenwich meridian. As the earth doesn’t follow the sun in linear manner and there is some wobbly rotation in it, so the UT1 calculates the corrections for that motion. UT2 corrects the remaining random variations that cause an error on the order of 10^{-9} which is about 60 ms per year. [11, p. 27]

Modern way of time keeping is the coordinated universal time (UTC) which was taken in the use in 1986. UTC is based on TAI (temps atomique international), that has made the standard for the international second (IS). The IS is defined as follows:

> “The IS is based on the transition between the two hyperfine levels of the ground state caesium atom, and is defined as the duration of 9,192,631,770 periods of radiation from this transition”

This procedure of calculation is done in 200 different national laboratories over the world and the average over this clock is calculated to form the TAI time scale. [11, p. 27]

4.2 Overview of time synchronization

Time goes forward and different processes happen along this single line of time. There might be multiple processes along the timeline but how can we ensure that the *event a* happened before *event b* when they are in different processes? This is where concepts of time synchronizing become important.

Time measurement devices can be divided into two clock categories, logical clock, and physical clocks. Logical clocks are concerned about the sequence of the events in which they happen. Logical clocks can be ordered or unordered. Unordered logical clock just time stamps the event as they happen in each process, and it doesn’t care if the events in different processes are in order. Where in the other hand, ordered logical clock tries to ensure that the events in different processes are in sequential order, this is called total ordering. Physical clocks as the name suggests are clocks that exist in the real world. Most common one being the clock on your wrist and the ones in electronic devices that need to keep track of time. [12, p. 2]

Systems that use clocks and time can also be separated in to two categories. Local system where the passage of time is measured only locally and global systems where the passage of time is needed to set to the current global time scale. Local systems measure time with a certain device and gives the time between event a and event b locally. Global systems are tied to a global passage of time such as all of transportation and communication. [11 p. 29]

Keeping accurate time on a single clock is a straightforward process although there are some underlying causes that affect the clocks accuracy. Challenges start to arise when there are two clocks, and the aim is to measure the same time as closely as possible. The measurement system having multiple clocks are called distributed systems.

4.3 Problems in distributed systems

Keeping accurate time in a distributed system is still an open problem that has hampered many researchers and scientist. Reason for this is the fact that world is a chaotic place where random variations in different parts of the time measurement process affect the results. One most important of these variations is *clock drift*. [12, p. 6]

Layman explanation for clock drift would be as follows. Let’s say that you have a wristwatch and you have set it to keep time from your local bell tower clock. Now you go on about your day and couple of weeks later you come back to the bell tower to check the time. You notice that your clock is dragging 2 minutes behind. It could have also been that the time would be 2 minutes ahead of the bell towers clock. This time difference is known as clock drift. [13, p. 1]

Modern day example of clock drift is a clock on a computer. If we have two computers that start to measure time at exactly same moment, over a period they will start to show different times. This is because on the atomic level the quartz crystals that are used to measure time in computer clocks oscillate in different frequencies and over time they will drift apart. The difference between these two clocks is called *skew.* [12, p. 6]

For accurate timekeeping methods the quartz crystal is usually cut in a shape of a disk from a single piece of crystal quartz at a specific angle. This ensures the quality of the quartz piece. Reason why quartz is commonly used for oscillators is because of its temperature error properties. When kept at a wrist the temperature doesn’t fluctuate that much and humans body temperature is optimal for quartz crystals. This temperature property can be seen in figure 6. [11. p. 22]

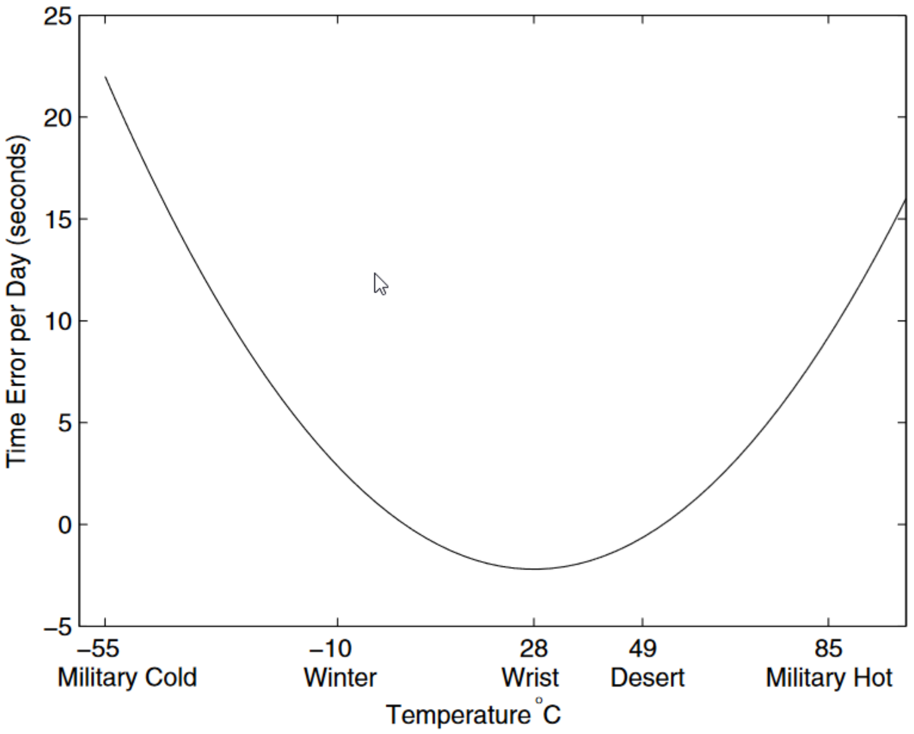


Figure 6. Wristwatches accuracy as a function of temperature. [11, p. 22.]

Other important effect on clock synchronization accuracy is *jitter.* Jitter is defined as timing variations at the edges of a signal. Depending on the device it is prone to environmental variation that affect the quality of the signal. This environment variations include thermal noise, power supply variations, loading conditions, device noise, and inference coupled from nearby circuits. [14, p. 1]

4.4 Algorithms

The three algorithms introduced here, are in the categories of simple, most used, and popular but controversial. They all are meant for different use cases. Cristian's algorithm is the simplest one that is used for crude applications where high precision is not paramount. Second, a bit more modern and widely used network time protocol (NTP) that is used in most Internet of Things (IoT) solution and lastly global positioning system (GPS) is discussed about.

4.4.1 Cristian’s algorithm

Cristian’s algorithm is a single server algorithm that is designed to work as a master clock for its other nodes that check time from it. The master clock will make timed requests to a time server to keep its own clock accurate. [12, p. 8.]

Cristian’s algorithm does not account for network or processing delays. In addition, the errors in this algorithm are cumulative. Meaning a delay at one node of ±3 ms and in a another ±5 then its ±(3+5) ms amounts to ±8 ms at the master clock. It means the useful applications of this algorithm are limited. Calculation for a new time in this algorithm is extremely simple. It is the following:

$$T_{new} = T_{server} + \frac{T_1 - T_0}{2}$$

Where T_{server} is server time, T_1 is response received and T_0 is request send.

Even with the hinderances of this algorithm it’s modified version is used for the postprocessing of time stamps in this thesis.

4.4.2 Network time protocol, NTP

One of the main applications for time synchronization is IoT. Networking and networking protocols need to keep accurate time to work properly so there are multiple different solutions. The most used solution is network time protocol (NTP). [15, p. 399]

Non-ideal communication channel and clock generators is a major challenge affecting the time synchronization over the Internet. The non-ideal affects include quantization, frequency changes, random variations with every tick (jitter), random frequency changes (wander), aging affect (only affects long periods of time), network delay and variable processing time. Therefore, it is important to have reliable time synchronization protocol for network devices [1, p. 399.]

NTP was developed by David L. Milss in 1991. Since his first article where the number of hosts was under 1 million but in 2019 it was over 1012 million host according to Statista. Because of the increasing number of users, the NTP algorithm is now at its 4th version as it has been revised for modern networking needs. [16; 17, p. 1]

NTP is now the most used synchronization protocol on Internet. NTP is also purely software-based protocol which makes it possible to implement on any device. Its estimated accuracy is approximately 1 ms. This is an overall good accuracy for commercial purposes but if more accurate timing is needed it gets trumped by other algorithms. [15, p. 404]

NTP works in a hierarchical model where it gives stratum numbers from 0 to 15 depending on how close you are to the 0 stratum, where 0 being the most accurate and then going down from there. We can see the diagram in figure 7

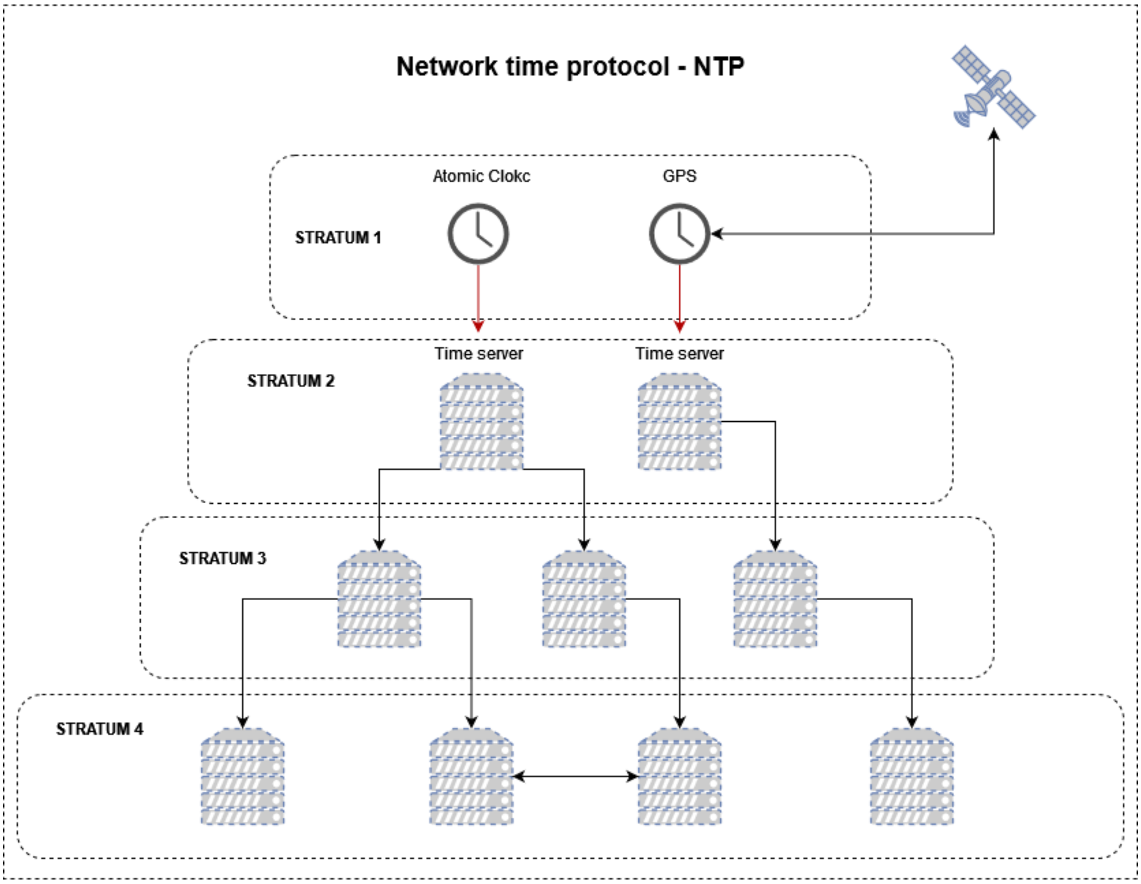


Figure 7. NTP diagram of stratum levels.

At stratum 0 we have a reference clock that is an actual high precision timekeeping device for example an atomic clock, GPS, or a radio clock. The reference clock is connected to a primary time server that is at the stratum 1. This primary time server is a reference point to other devices on the network. The stratum number decreases on every connection between devices in that network. Anything over 15 is not considered accurate anymore and should not be used. NTP servers are usually at User Datagram protocol (UDP) port 123. [18]

4.4.3 Global positioning system, GPS

Global positioning system (GPS) that is most known for its use in mapping services, but it can be also used to synchronize clocks. There are 24 satellites that orbit around the earth. The system is operated by the United States Department of Defence, and it was originally developed for military purposes. Today its most used system for keeping UTC up to date. [11, p. 32]

Every GPS satellite contains an atomic clock inside it. The signal is sent to a ground receiver and correction calculations are made. The correction calculations allow the timing to be accurate to about few nanoseconds. GPS system is highly reliable. [11, p. 32]

Only downside is that it is owned by the United States of Department of Defence. Reason for the downside is the fact that US could make the GPS time algorithm slower and not tell the users how to correct the slowness. Implications of this are massive because GPS is so widely used time protocol. They could for example stop an entire country’s public transport infrastructure if it’s timed by GPS. Concerns over GPS have caused other nations to start their own satellite projects. Good example of this is the EU’s Galileo satellite system that has been live since 2016. [11, p. 32]

4.5 Use cases

The following use cases were selected having the aim of this thesis in mind. The challenge of using multiple biosensors having different local clocks, the synchronization of the signal becomes a challenge.

4.5.1 Human Robot Interaction

Robert L. Wilson et.al studied human’s mental workload using multiple biosensors in a VR environment where the participant was tasked to ride a simulated lunar rover on the surface of the moon and completed different tasks. They used NASA readymade Multi-Attribute Task Battery II (MATB-2) that is designed to benchmark operators’ performance and workload by engaging the participant to keep track of different gauges and meters. [19, p. 1; 20]

In this study they measured six different biological signals. Electrocardiogram, electrodermal activity, photoplethysmography, respiration, skin temperature and pupillometry. These signals where recorded with three devices. Empatica’s E4 wristband that measures PPG, GSR and skin temperature, Zephyr BH a heartbeat monitor and Vive Pro Eye for VR headset that contains the pupillometry.

Main interest of this study was at the discussion portion of the paper where they described difficulties by using two different operating systems (Windows and Linux). Reason for this solution was that all the devices didn’t support both of operating systems. In addition, it was mentioned that they used Object Oriented Programming methods to achieve easier modifications in the future.

4.5.2 Hardware solution

Another solution for time synchronization stems from the hardware level of the measuring devices. Where one would build a custom measuring setup with hardware part. One such system was built by Siddharth from University of

California San Diego. He built a custom multi-sensory setup that contained EEG, photoplethysmography (PPG), eye-gaze headset, body motion capture and GSR. The main purpose was to close the gap between the real world and the laboratory environment to create a system that one could wear while moving around in a real world. [21, p. 1137]

All the measuring devices were connected to a single processing unit that keeps track of time synchronization using the Lab streaming layer (LSL). LSL is a synchronization library for different programming languages. It is used for timestamping every piece of recorded data which either can be provided by the user via their own clock system or from a clock source where the synchronization is handled by LSL. [21, p. 1140; 22]

This solution produced satisfactory results but the downside for this type of implementation is the need to be truly knowledgeable at electrical engineering to be able to implement it.

4.5.3 Algorithmic Solution

An algorithmic solution is also an option where one would develop a synchronization algorithm from ground up to fit researchers needs. Isaac S. Chang in his paper developed a novel method to synchronize multiple biosensors. It presents an algorithm that can align signals recorded by different sampling frequencies to a millisecond precision. [23, p. 1]

The measuring setup consisted of 5 different measuring devices at two different locations inside a room. Purpose for this kind of divide was that the measuring devices are scattered so that the measurements could be recorded without any additional effort from the researchers than the initial start of the recording. Devices where ballistocardiogram (BCG) and accelerometers at the bed, and two devices recording signals of load cells and capacitive-coupled (CC) electrodes installed on chair that also measured BCG, weight and ECG. [23, p. 1]

Central function of the algorithm is digital-to-analog (DAC) converter that is attached to each measurement device’s analog-to-digital (ADC) converter. DAC generates a waveform between 0 and 2 volts. Each device is given its unique 0,5 V range where its signals reside. This cycled waveform can be seen in figure 8.

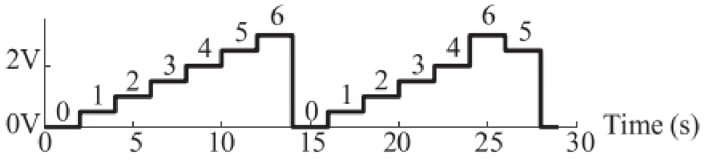


Figure 8. Two cycles of synchronization waveform. [23, p. 2]

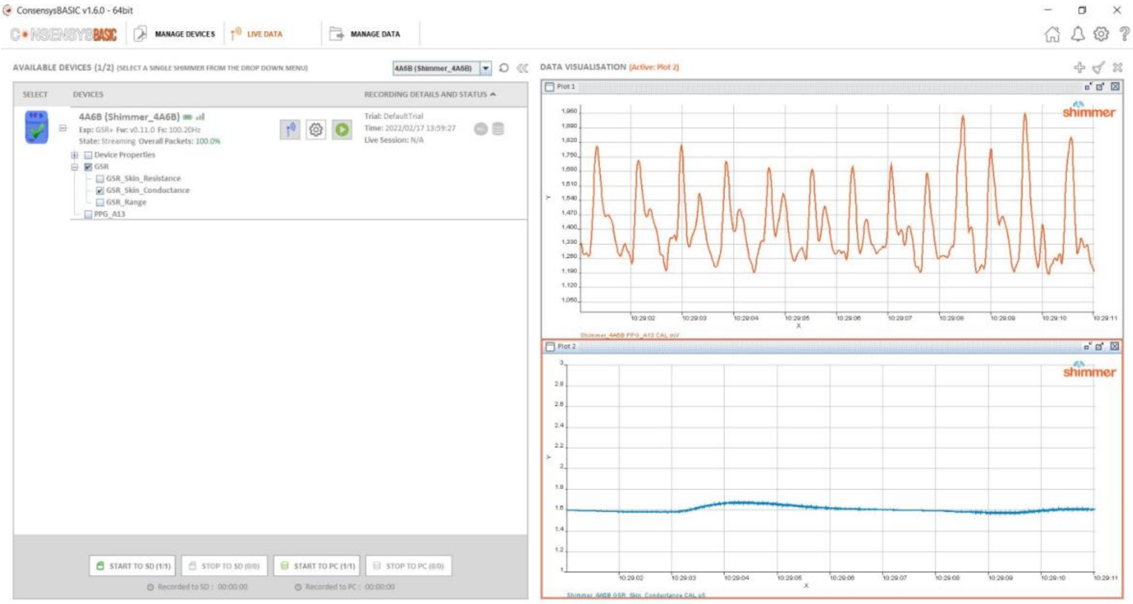
The cycle begin, they are clearly identifiable and can now be ordered in post processing. [23, p. 2]

This type of solution for time synchronization is complicated and requires a lot of work but the results are promising. Downside of this method is the fact that it is developed in 2019 and it is not yet peer reviewed.

5 Development process

At the beginning of the thesis work I read the devices documentation and “how to guide” for the initial setup. Shimmers ECG and GSR device came with a charging station that also function as a dock that can be used to operate the devices. Dock can be seen in Figure 9. I installed drivers and tested that the dock connects to computer. Then ECG and GSR were connected via Bluetooth and the Consensys program was ran that is made for the dock. In Consensys the measuring settings for the ECG and GSR are modified, and the data is captured and recorded. After the setup I started testing that the devices worked by making couple of test measurements. They worked fine and produces the data that was expected. These measurement results can be seen in a figure 10.

Figure 9. Shimmer3 docking station with four measuring units.



4A6B (Shimmer_4A6B)

4A6B (Shimmer_4A6B)
Exp: GSR+ Fw: v0.11.0 Fs: 100.20Hz
State: Streaming Overall Packets: 100.0%

- ☐ Device Properties
- ☑ GSR
  - ☐ GSR_Skin_Resistance
  - ☑ GSR_Skin_Conductance
  - ☐ GSR_Range
- ☐ PPG_A13

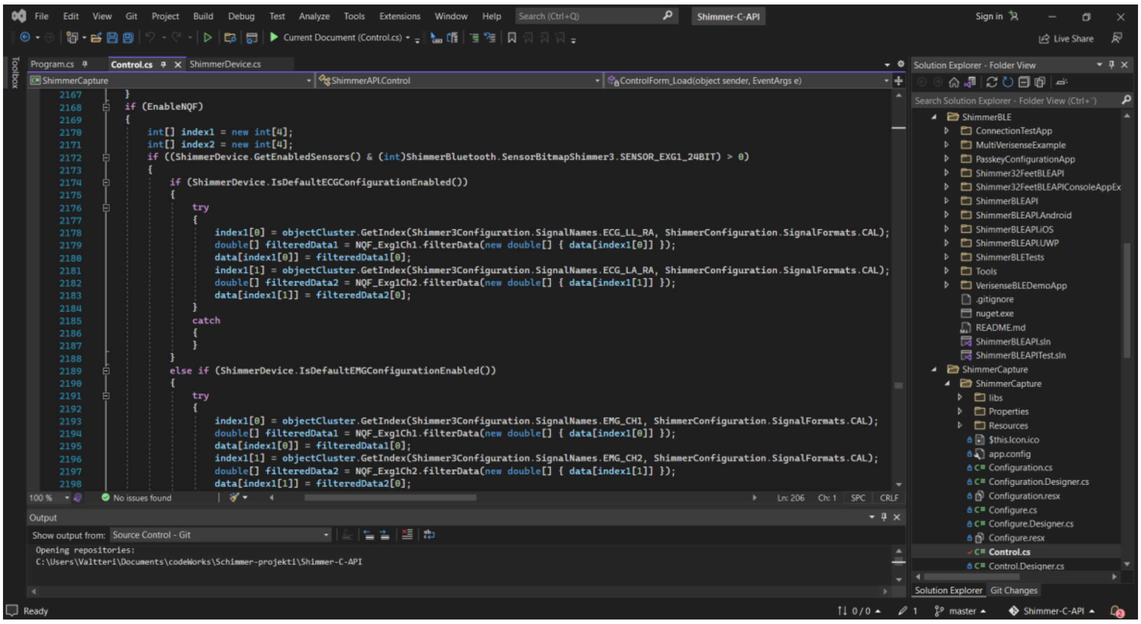
Trial: DefaultTrial
Time: 2022/02/17 13:59:27
Live Session: N/A

Recorded to SD : 00:00:00
Recorded to PC : 00:00:00

Figure 10. Consensys data streaming application. Plot1 is showing GSR skin resistance data and Plot2 is shows skins conduction.

When I was more familiar with the devices and how they operate I downloaded Shimmers C# API from GitHub and started to test it. I got the example code working and measuring with the devices. The coding process usually starts at this point by implementing and testing couple of small parts from the sample code, but here is where the challenges started.

Shimmer code base that was meant to be for developers was not documented or commented at all. This means that the code base with thousands of lines of code doesn't have any instructions on how to use it. An example of source code without documentation can be seen in figure 11.



```
}
if (EnableNQF)
{
    int[] index1 = new int[4];
    int[] index2 = new int[4];
    if ((ShimmerDevice.GetEnabledSensors() & (int)ShimmerBluetooth.SensorBitmapShimmer3.SENSOR_EXG1_24BIT) > 0)
    {
        if (ShimmerDevice.IsDefaultECGConfigurationEnabled())
        {
            try
            {
                index1[0] = objectCluster.GetIndex(Shimmer3Configuration.SignalNames.ECG_LL_RA, ShimmerConfiguration.SignalFormats.CAL);
                double[] filteredData1 = NQF_Exg1Ch1.filterData(new double[] { data[index1[0]] });
                data[index1[0]] = filteredData1[0];
                index1[1] = objectCluster.GetIndex(Shimmer3Configuration.SignalNames.ECG_LA_RA, ShimmerConfiguration.SignalFormats.CAL);
                double[] filteredData2 = NQF_Exg1Ch2.filterData(new double[] { data[index1[1]] });
                data[index1[1]] = filteredData2[0];
            }
            catch
            {
            }
        }
        else if (ShimmerDevice.IsDefaultEMGConfigurationEnabled())
        {
            try
            {
                index1[0] = objectCluster.GetIndex(Shimmer3Configuration.SignalNames.EMG_CH1, ShimmerConfiguration.SignalFormats.CAL);
                double[] filteredData1 = NQF_Exg1Ch1.filterData(new double[] { data[index1[0]] });
                data[index1[0]] = filteredData1[0];
                index1[1] = objectCluster.GetIndex(Shimmer3Configuration.SignalNames.EMG_CH2, ShimmerConfiguration.SignalFormats.CAL);
                double[] filteredData2 = NQF_Exg1Ch2.filterData(new double[] { data[index1[1]] });
                data[index1[1]] = filteredData2[0];
```

```
Opening repositories:
C:\Users\Valtteri\Documents\codeWorks\Schimmer-projekti\Shimmer-C-API
```

Figure 11. Example of undocumented code.

This kind of work is equivalence of having a car engine and all its part but no manual. It is possible to start building piece by piece, but it takes a lot of time and resources which unfortunately at this point of the project I did not have.

After realization of the missing documentation, I contacted shimmer and asked if they had any code base or additional documentation. They did not and they didn't have any additional documentation. As a summary they have a code base that other developers are supposed to use but there is no guide for how to use it.

Because of this unfortunate incident a more theoretical chapter was added due to time management. In the next chapter is discussed about how the code solution could have been done and what could have been done differently.

6 Alternative solution and thoughts

Final chapter is divided into two parts. I am going through the possible solution in an easily understood programming language python to give the reader a clearer explanation for the code. In the final thoughts part, I am discussing my overall reflections of the thesis work.

6.1 Possible solution

Because the device provider’s C# library was not adequately documented, I am doing a demonstration of how I would have solved the time synchronization using python. This is a modified version of the Cristian’s algorithm that is simple to reproduce.

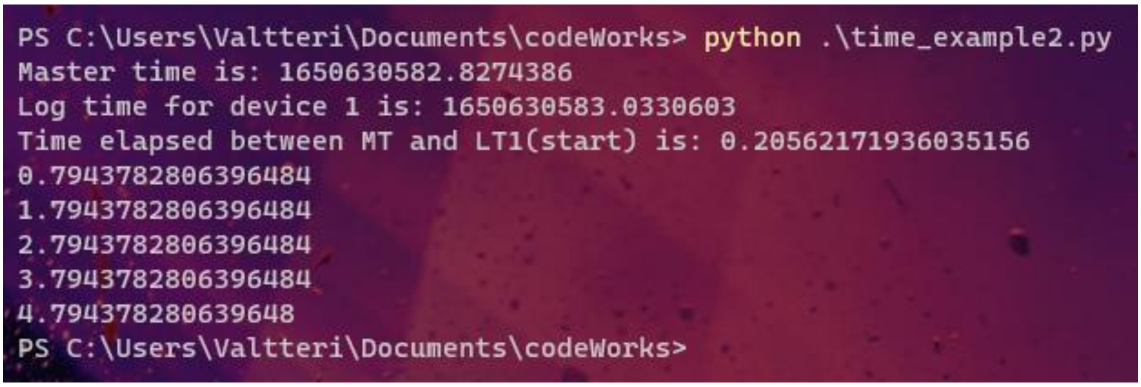
When evaluating the algorithm using python time library, it is important to note that most Unix systems clock “ticks” only 50 or 100 times a second. Consequently, this kind of application of time measuring is only feasible if we do not need to be extremely accurate. Otherwise, we would set up a NTP server in LAN (Local Area Network) which would be accurate to < 1ms. [16]

In listing 1 we can see the process written in python. Simulated time data is used to represent some form of data modification. Time is calculated using epoch time in seconds using the pythons Time library. Important part comes in the post-processing where all the data is recalculated to represent the real start time. The output can be seen in figure 12.

This solution doesn’t consider the delay that accumulates during the recording of the data, or the delay caused by the jitters of the quartz clock on the computer.

```
import time

# Start the master clock when the program starts
# Master time (MT)
main_time = time.time()
print("Main time is: {}".format(main_time))

# Artificial delay added
time.sleep(.2)

# Log time 1
LT1 = time.time()
print("Log time for device 1 is: {}".format(LT1))

# Save the time difference between (MT - LT1) = D1_delay
LT1_MT_difference = LT1 - main_time
print("Time elapsed between MT and LT1(start) is:
{}".format(LT1_MT_difference))

# Stop recording on devices
imaginary_time_data = [1.0, 2.0, 3.0, 4.0, 5.0]

# Post-processing
# Process data's time stamps by subtracting delays
# from each datapoin (datapoints timestamp - D1_delay)
i = 0
while i < len(imaginary_time_data):
    new_time = imaginary_time_data[i] - LT1_MT_difference
    imaginary_time_data[i] = new_time
    i += 1

for x in imaginary_time_data:
    print(x)
```

Listing 1. Python code example of post processing timestamps.



```
PS C:\Users\Valtteri\Documents\codeWorks> python .\time_example2.py
Master time is: 1650630582.8274386
Log time for device 1 is: 1650630583.0330603
Time elapsed between MT and LT1(start) is: 0.20562171936035156
0.7943782806396484
1.7943782806396484
2.7943782806396484
3.7943782806396484
4.794378280639648
PS C:\Users\Valtteri\Documents\codeWorks>
```

Figure 12. Python codes post processing output on a command line.

As it can be seen in the Figure 12 the elapsed time between master time and the devices start time was 0.2 seconds. All the devices would have been using the same programming interface and same centralized clock so the only calculation that would have been needed is the elapsed time between the devices.

6.2 Final thoughts

The most crucial part that did not go as planned was the undocumented C# library that I was supposed to use. To use the library, it would have required more time and effort than was available in the time frame of this work. Due to that I had to rethink my thesis process all over again and it raised concerns if I had to abandon my work altogether. Fortunately, this was not the case. A very valuable lesson learnt from this experience was to check the code base first. What type of state is it in? When was it last updated? And so on.

Calibration and installation of the devices was successful. I learned a lot from measuring devices and things to consider when using them. For example, where and how the placement of the measuring device effects on the measured data and what types of devices are on the market and what could be done with them.

I also improved as a writer. This is after all a full fletched report that might be read by someone someday. In addition, now that I worked in a project that has a scientific use case, I learned a lot of new things that need to be taken into consideration.

But the main take away of this work was the deeper dive inside the world of time keeping and what types of things it brings with it if you aim to measure time accurately. Time synchronization is a major challenge of many data acquisitions projects, so I am glad that I chose this topic.

References

1 Imotions. 2020. Electrocardiography (ECG) The Complete Pocket Guide. [Internet][Accessed 10 February 2022]. Available from: https://imotions.com/guides/eda-gsr/

2 Alila Medical Media. 2014. Cardiac Conduction System and Understanding ECG, Animation. [Internet, video] [Accessed 10 February 2022]. Available from: https://www.youtube.com/watch?v=RYZ4daFwMa8

3 Shimmer. 2018. ECG User Guide Revision 1.12. [Internet] [Accessed 05 March 2022] Available from: https://shimmersensing.com/product/shimmer3-ecg-unit-2/

4 Nagai Y, Jones CI and Sen A (2019) Galvanic Skin Response (GSR)/Electrodermal/Skin Conductance Biofeedback on Epilepsy: A Systematic Review and Meta-Analysis. Front. Neurol. 10:377. doi: 10.3389/fneur.2019.00377

5 Imotions. 2017. GSR Pocket Guide. [Internet] [Accessed 22 March 2022]. Available from: https://imotions.com/guides/eda-gsr/

6 Imotions. 2017. Eye Tracking The Complete Pocket Guide. [Internet] [Accessed 05 March 2022]. Available from: https://imotions.com/guides/eye-tracking/

7 Video-oculography. Wikipedia. 2022. [Internet] [Accessed 10 September 2022]. Available from: https://en.wikipedia.org/wiki/Video-oculography

8 Tervonen, J.; Pettersson, K.; Mäntyjärvi, J. Ultra-Short Window Length and Feature Importance Analysis for Cognitive Load Detection from Wearable Sensors. Electronics 2021, 10, 613. https:// doi.org/10.3390/electronics10050613

9 Vatheuer, Carolyn & Vehlen, Antonia & Dawans, Bernadette & Domes, Gregor. (2021). Gaze behavior is associated with the cortisol response to acute psychosocial stress in the virtual TSST. Journal of Neural Transmission. 128. 10.1007/s00702-021-02344-w.

10 Liu Q, Zhang W. Sex Differences in Stress Reactivity to the Trier Social Stress Test in Virtual Reality. Psychol Res Behav Manag. 2020 Oct 29;13:859-869. doi: 10.2147/PRBM.S268039. PMID: 33154681; PMCID: PMC7605969.

11 John C. Eidson. 2006. Measurement, Control and Communication Using IEEE 1588. London: Springer-Verlag.

12 Paul Kryzanowski. 2000 – 2009. Lectures on distributed systems Clock Synchronization. Rutgers University – CS 417: Distributed Systems.

13 Leslie Lamport. 1978. Time, Clocks, and the Ordering of Events in a Distributed System. Massachusetts Computer Associates, Inc: Advanced Research Projects Agency of the Department of Defense and Rome Air Development Center.

14 Clock Jitter Definitions and Measurement Methods. 2019. 5451 Patrick Henry Drive, Santa Clara, CA 95054, USA. SiTime Corporation.

15 Puttnies, Henning, Peter Danielis, Ali Rehan Sharif, and Dirk Timmermann. 2020. "Estimators for Time Synchronization—Survey, Analysis, and Outlook" *IoT* 1, no. 2: 398-435. https://doi.org/10.3390/iot1020023

16 How Accurate Is Network Time Protocol (NTP)? 2020. [Internet] [Accessed 13 August 2022]. Available from: https://timetoolsltd.com/ntp/ntp-timing-accuracy/

17 David L. Mills. 1991. Internet Time Synchronization: The Network Time Protocol. Department of Electrical Engineering, University of Delaware, Newark, DE 19716: Network Protocols of the IEEE Communi-cations Society.

18 CertaBros. 2020. NTP Explained | Network Time Protocol | Cisco CCNA 200-301. [Internet, video] [Accessed 5 August 2022]. Available from: https://www.youtube.com/watch?v=oCtkwEjhyD4&t=199s

19 Wilson, Robert & Browne, Daniel & Wagstaff, Jonathan & McGuire, Steve. (2021). A Virtual Reality Simulation Pipeline for Online Mental Workload Modeling.

20 Eric Chancey. 2016. MATB-II. [Internet] [Accessed 6 June 2022]. Available from: https://matb.larc.nasa.gov/

21 Siddharth, Patel, A. N., Jung, T. P., & Sejnowski, T. J. (2019). A Wearable Multi-Modal Bio-Sensing System Towards Real-World Applications. *IEEE transactions on bio-medical engineering*, *66*(4), 1137–1147. https://doi.org/10.1109/TBME.2018.2868759

22 LabStreamingLayer’s Documentation. 2019. [Internet] [Accessed 17 June 2022]. Available from: https://labstreaminglayer.readthedocs.io/info/time_synchronization.html

23 I. S. Chang et al., "Novel Method for Synchronization of Multiple Biosensors," in IEEE Sensors Letters, vol. 4, no. 1, pp. 1-4, Jan. 2020, Art no. 5500104, doi: 10.1109/LSENS.2019.2962365