

MODULAR FULL-BODY TRACKING SUITE

BACHELOR THESIS

submitted by
Simon Jarc

born on: 21.07.1995

Wirtsberg 5
85777 Kammerberg
Tel.: 08137 809638

MENSCH-MASCHINE-KOMMUNIKATION
Technische Universität München

Univ.-Prof. Dr.-Ing. habil. Gerhard Rigoll

Supervisor: Alona Kharchenko, M.Sc.
Start: 17.12.2018
Final Submission: 03.05.2019

Abstract

This thesis tries to develop a Tracking Suite based on SteamVR Tracking BaseStation 2.0. The functionality of BaseStation 2.0 is described. Linear Feedback Shift Registers are important and are described. To perform the detection successfully, the approach with Configurable Linear Feedback Shift Registers is introduced. Based on this approach, a solution algorithm for the detection of bitstreams based on an FPGA architecture is described. The difficulties in the course of the project are explained and other possibilites are considerd.

Thank You to all. Especially to all Roboyans.

Pleased to meet you, Roboy.

Thank You Mama, Papa and Vanesa for support in all imaginable situations.
Special Thanks to Felix and Johannes. You have motivated and pushed me until
the end.

Contents

| | | |
|----------|---|-----------|
| 1 | Introduction with Motivation | 5 |
| 2 | Theoretical Background | 7 |
| 2.1 | Linear Feedback Shift Register | 7 |
| 2.2 | Telecommunication Channel | 8 |
| 2.3 | ASIC from Triad Semiconductor TS4231 | 9 |
| 3 | System Design | 13 |
| 3.1 | GitHub Discussions | 13 |
| 3.2 | Photodiode | 14 |
| 3.3 | Arduino MKR Vidor 4000 | 14 |
| 4 | Implementation | 15 |
| 4.1 | Measurement of laser signals | 15 |
| 4.2 | Real-Time FPGA Detection | 16 |
| 4.3 | PCB Design | 17 |
| 5 | Results - split observation and interpretation | 23 |
| 6 | Discussion | 29 |
| 7 | Summary | 31 |
| | List of Figures | 33 |
| | Bibliography | 35 |

Chapter 1

Introduction with Motivation

The main idea behind this system is to record human motion. Roboy, like humans, consists of muscles and tendons. In this case the motors take on the task of the muscles. As the robot cannot be supplied data by unlimited sensors, the motors must be controlled specifically. This does not happen as naturally as humans. In order to improve this, we want to record human movements. For this a component of SteamVR Tracking is used, more exactly the newer SteamVR Tracking 2.0. In the following this is called Lighthouse 2. With this partial adaption, human movements should be recorded as precisely as possible. For the future these images are to be made usable for Roboy with the help of artificial intelligence.

The Lighthouse 2 is installed at a fixed position in the room and then the room is traversed by a laser in two axes. This laser is highly modulated so that a bistream is sent. This is now received with a module and a photodiode on it. By receiving the bitsream, the position in room can be determined in relation to Lighthouse 2. For this project the Lighthouse 2 must be reengineered, because of the lack of official documentation about the technical functionality.

For this purpose, chapter 2 will first create the basics to be able to understand a later discussion. Chapter 3 and 4 describe design and implementation in development. Chapter 5 and 6 summarize all events in the process of development. A special focus here is on problems arising during development. Chapter 7 forms the conclusion.

Chapter 2

Theoretical Background

This chapter gives a brief overview of the most important underlying technologies and products. Each section can be considered separately.

2.1 Linear Feedback Shift Register

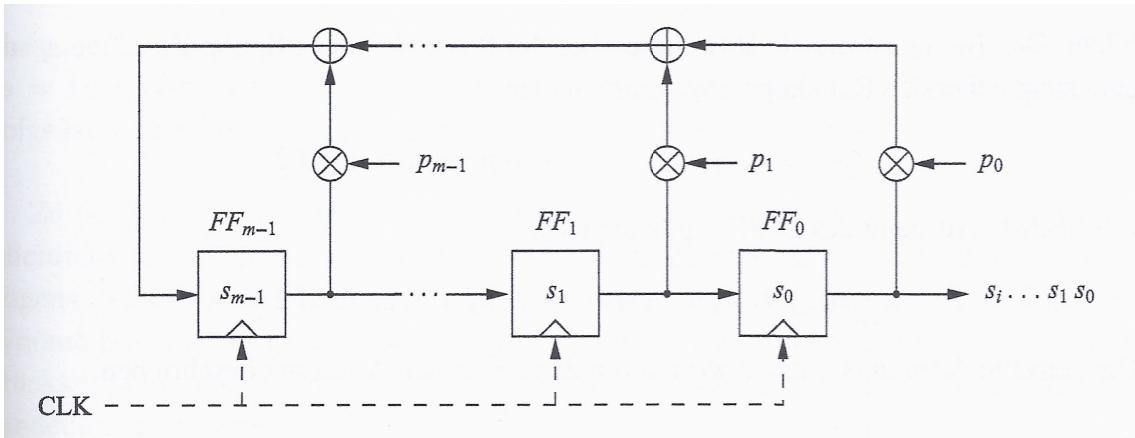


Figure 2.1: A Schematic of a Linear Feedback Shift Register [PP16]

Using a register is the easiest way to store one bit in digital form, often used in microcontrollers. However, the exact internal structure will not be considered here, since the simple mode of operation and rather the mathematical description is more important in this section. In order to store a number of bits, the same number of registers are used and connected according to the application.

We want to concentrate on a Linear Feedback Shift Register, which is structured in the following:

1. Take m registers and connect the input from step $i - 1$ to the output from step i .

2. Connect a clock to all registers so that all registers are synchronized with each other. Now with each clock cycle the data bit is moved one step further. With this wiring we buildt a shift register, which pushes out bits sequantiall at its output. This means that a bit at the input is only available after m steps at the ouput.
3. To get a Linear Feedback Shift Register we first clarify the concept of feedback. Feedback describes the process of feeding a signal to the output and back to a previous stage. In this context, linear feedback means that the signal is tapped and fed back only once.

For a Linear Feedback Shift Register, the output from stage j must be routed to input register s_m . If several steps are fed back, they are all exclusivley ored together. Figure 2.1 shows the basic structure of such a Linear Feedback Shift Register.

Next, it is useful to take a closer look at the mathematical description. For this we also use the nomenclature of figure 2.1, corresponding to [PP16].

The feedback is described by the coefficients $p_j \in \{0, 1\}$ and the register stages by $s_0, s_1 \dots s_{m-1}$. We are most interested in the output bits $s_0, s_1 \dots s_i$, which can generally be expressed as follows:

$$s_{m+i} \equiv \sum_{j=0}^{m-1} p_j \cdot s_{i+j} \bmod 2 \quad (2.1)$$

with $s_i, p_j \in \{0, 1\}$ and $i = 0, 1, 2, \dots, m$. The maximum length possible combinations depends on the degree m and can be described by equation 2.2.

$$2^m - 1 \quad (2.2)$$

With the feedback coefficients known to us, we can now express a known Linear Feedback Shift Register as a polynomial:

$$P(x) = x^m + p_{m-1}x^{m-1} + \dots + p_1x + p_0. \quad (2.3)$$

It should be noted that only primitive polynomials can generate the maximum length of sequence. Other polynomials that are not based on these do not reach the maximum length according to equation 2.2.

It is also useful to know how to calculate the feedback of unknown Linear Feedback Shift Registers. For this, $2m$ output bits are needed and with equation 2.1 they can be presented in the following:

$$\begin{aligned} i = 0, \quad s_m &\equiv p_{m-1}s_{m-1} + \dots + p_1s_1 + p_0s_0 \\ i = 1, \quad s_{m+1} &\equiv p_{m-1}s_m + \dots + p_1s_2 + p_0s_1 \\ &\vdots \\ &\vdots \\ i = m, \quad s_{2m-1} &\equiv p_{m-1}s_{2m-2} + \dots + p_1s_m + p_0s_{m-1} \end{aligned}$$

Since we know $2m$ output bits and have represented them as linear equations, we can easily calculate them, for example with the Gaussian Elimination method. [TSG16] [PP16] [MTT12]

2.2 Telecommunication Channel

The transmission of messages can be illustrated very well with figure 2.2.

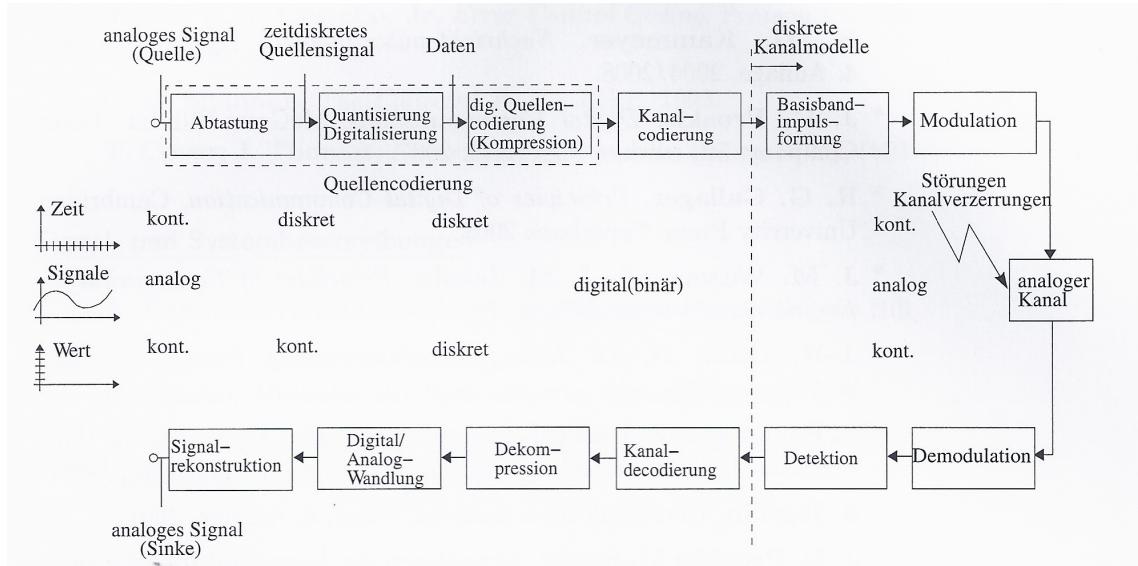


Figure 2.2: Communication Link [HKS20]

The most important point to mention here is the channel coding. This consists roughly of three areas:

- Cryptography: The message to be transmitted is encrypted.
- Source Coding: The message to be transmitted is compressed.
- Channel Coding: Increase tolerance against wrong transmitted message.

[Bos13]. These methods are applied in any digital communication. The transmitter carries out these tasks and transmits its message to the participant. The receiver there must process it again in a reverse order. The receiver knows the instructions for reading the message. [HKS20] [Bos13]

2.3 ASIC from Triad Semiconductor TS4231

The TS4231 from Triad Semiconductor is an Application Specific Integrated Circuit (ASIC) designed to detect laser signals and transmit them in digital form. For this

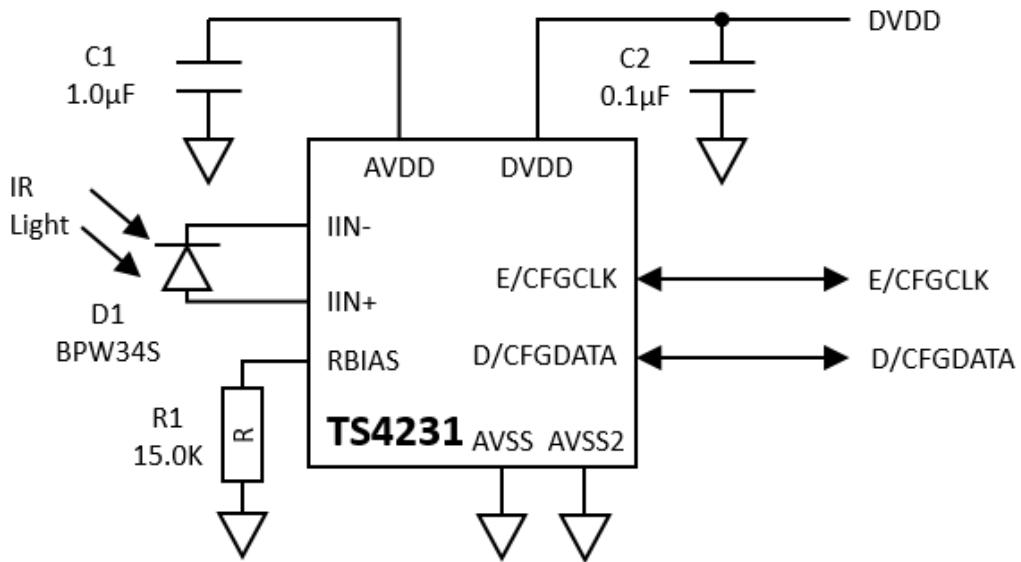


Figure 2.3: Simplified Circuit Schematic of TS4231 [Tri16b]

purpose a photodiode is required which is sensitive on the laser light and passed on in the form of currents. This signal is then converted using a transimpedance amplifier and processed, so that digital voltage signals can be further processed at the pins *E/CFGCLK* and *D/CFGDATA* of the ASIC [Tri16b].

In order to use this ASIC, it must be designed into a circuit board. Figure 2.3 shows the ASIC and how it can be equipped to components to the circuit. Particular attention must be paid to the layers of the board, due to the fact that mixed-signal circuits are very sensitive against electromagnetic fields, described in [Tri16a].

In addition, Triad Semiconductors provides a sample configuration of the ASIC for Arduino. [Tri16a] This can be used for initial tests and trials.

Chapter 3

System Design

This chapter describes how the system is designed. The main focus is on decoding the signals from the Lighthouse 2. For this purpose, the Lighhouse sends modulated laser signals, with which the current position can be determined during receiving. Section 3.1 forms the basis for the argumentation and is considered first. It is very important to mention that this discussion was not conducted under any scientific aspect. Unfortunately, there is no official documentation of the manufacturer about the construction of the bitstream and the exact functionality of the Lighthouse, which makes this discussion the only point of argumentation.

3.1 GitHub Discussions

In the first discussion, the laser signal is coming from the Lighthouse 2 was captured and visualized with the help of a measuring device. The device [Cyp28] with a photodiode was used. Unfortunately it does not show which diode it was. The next step was to display the received bits. It was recognized that this is an Linear Feedback Shift Register consisting of 20 steps. For the analysis of the feedback the Berlekamp-Massey algorithm was used. Per Lighthouse 2 two different feedback coefficients are used. Furthermore, two different speeds of bitstreams are transmitted. A fast stream with $0.0833\mu\text{sec}$ per bit for sequences of the Linear Feedback Shift Register and a slow bitstream for sequences of control. [.]

The second discussion tries to decode the slow bitstream. The "Device SN" (on the back of each device), "ootx.nonce" and a "CRC32" were analyzed. That are three blocks of bits that have been analzed. In between are unknown bits. [.b]

In a further discussion, an attempt is made to develope an algorithm for decoding the bitstream from these two results. The feedback is used to try to transfer the bitstream to the next bit. Parallel to that, a counter with 20 digits is incremented and if successful, the found bitstream is returned. [.c] It has been tried to apply the principle of a quantizer, where a tradeoff was made between power and performance by using a parity table. [HKS20]

It is very important to mention that reasonable and useful results were only achieved

after the deliberate destruction of the Lighthouse 2 case with a soldering iron. The bitstreams of the laser signal were measured with the help of an oscilloscope and then evaluated again as described before. [.e]

3.2 Photodiode

Triad Semiconductors proposes in [Tri16a] to use the *BPW34S* as photodiode. This is a PIN photodiode. In general, PIN diodes have very good high frequency characteristics. So they can be used as high frequency switches to receive a fast bistream like here. The *BPW34S* is very sensitive in the visible and infrared spectrum. [Vis17] [TSG16]

3.3 Arduino MKR Vidor 4000

Arduino offers with the MKR Vidor 4000 a very small and powerful SoC. It is equipped with an Intel Cyclone 10CL016 as a Field Programmable Gate Array and Microchip ATSAMD21 as a microcontroller. The board offers 22 digital I/O pins to connect the TS4231. The FPGA is equipped with 15.408 logic elements and is programmed with Intel Quartus Prime Lite Edition.

The Arduino is designed so that it can be used as a module with the sensors. The evaluation of the received bits should be implemented with the FPGA. The calculations of the current position can be done either by the microcontroller or externally. [.e] [Int] [Int17] [Mic18]

Chapter 4

Implementation

In order to use the Lighthouse 2 as a component in a closed system, it has to be analyzed. The first step is to receive, measure and evaluate the bitstream from the laser of the Lighthouse, see section 4.1. The goal is to design an new hardware with the help of an FPGA from the measurements and evaluation. In the final step, a Printed Circuit Board (PCB) design is created to meet the individual requirements.

4.1 Measurement of laser signals

To design a decoder for the bitstream based on streams from the Lighthouse 2, it must first be recorded and then evaluated. For the measurement of the bistreams, the procedure was almost the same as for physical measurement experiments in general. Figure 4.2 shows the test setup used.

A painter tape was used to determine a distance of 2.5meters between the Photodiode and the Lighthouse. The existing PCB module with the TS4231 and the suggested diode [Tri16a] was used as the sensor. In order to record now also different positions the sensor was not changed in the position, but the Lighthouse was turned on its axis. In addition the Lighthouse 2 stood once in front of the photodiode and in each case $\pm 45^\circ$. Figure 4.2 shows the Lighthouse in its orientation. So that the TS4231 can be used, it must be programmed with the demo software at the beginning. For this an Arduino MKR Vidor 4000 was used and programmed with a freely availabe sample configuration provided by Triad Semiconductor [Sei]. The signal was then picked up at the *D/CFGDATA* pin with an oscilloscope. An oscilloscope of the type SIGLENT SDS1204X-E was used. The trigger was set as follows:

- Type: Pattern
- Source: K1
- Logic1: 3.30V



Figure 4.1: A picture of the inside of the Lighthouse after the destruction of the front [.]b]

- Logic: AND
- Time: 2.20ns
- Minimum blocking time

The next step is the recording of measurement data. The Oscilloscope also offers the possibility to transfer data via a USB interface. With this interface and a USB stick screenshots can be taken as shown in Figure 4.3. It is also possible to record a dat-file and then read it into Matlab. Thus the furter evaluation of the measurement data takes place in Matlab.

The goal is to set up equation 2.3. The maximum length is known from section 3.1. So with the help of equation 2.1 a linear equation system of the 20th order is set up. This is now solved with the Gauss elimination method and as a result we get a polynomial.

Unfortunately, at no time was it possible to solve the linear system of equations with the existing methods. Matlab issues a warning with the following information: "Matrix is singular to working precision." This means that the matrix is not invertible and there is no unique solution for Gauss elimination. The result is written

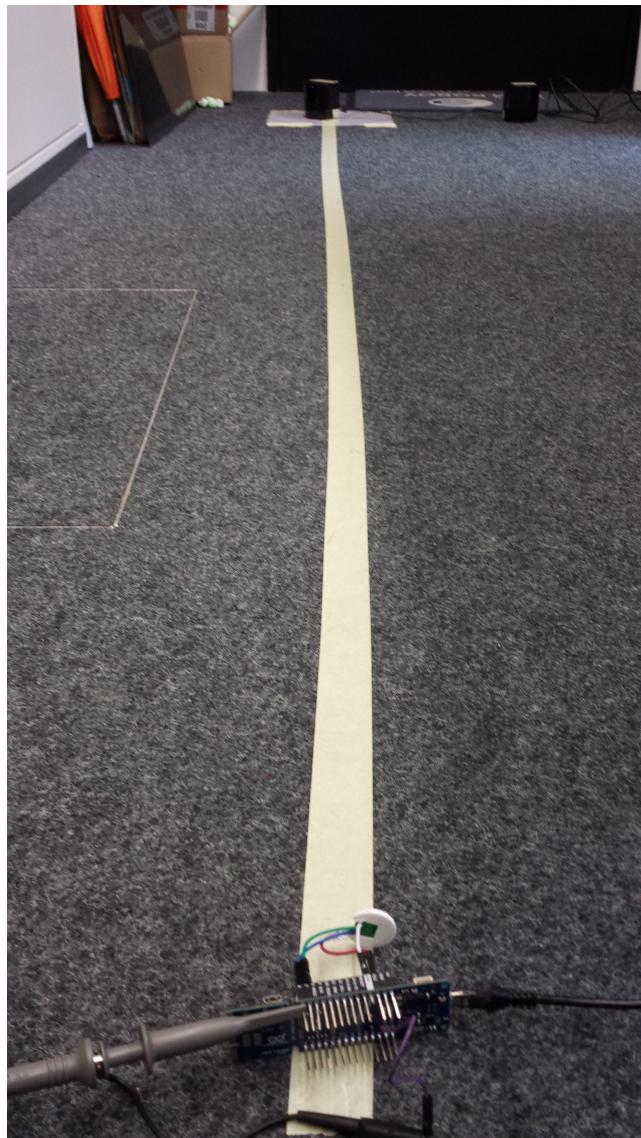


Figure 4.2: A picture of the test arrangement

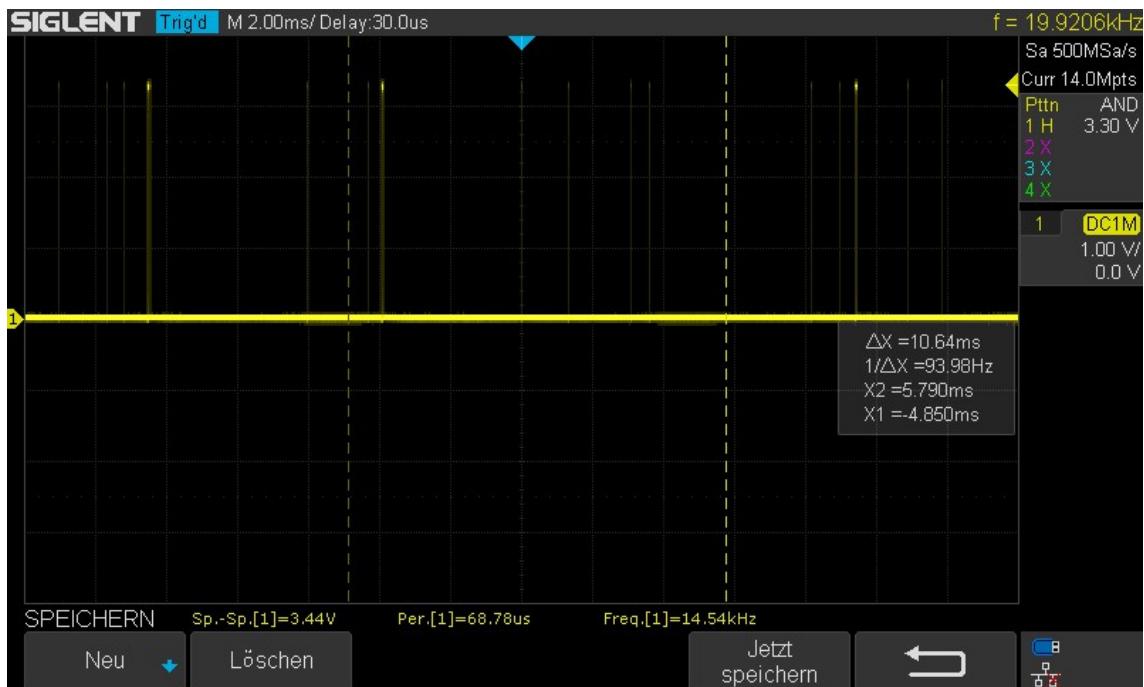


Figure 4.3: Screenshot on the oscilloscope

to the txt-file as "NaN", which means as much as "Not a Number." There were also no other measuring instruments available to check the counter. An experiment with the DSLogic Plus [.f] also failed because the TS4231 is sensitive to measuring devices. [Tri16a] In section 3.1 other possible brutal solutions are discussed in more detail.

4.2 Real-Time FPGA Detection

The main idea of using an FPGA is to be able to program an individual logic. The goal is to rebuild the hardware of the Lighthouse in reverse order to decode the signal.

With regards to the discussion in section 3.1, the main focus is on Linear Feedback Shift Registers. Since each Lighthouse has its own feedback coefficients, a Configurable Linear Feedback Shift Register must be programmed, as described in [MTT12]. For this, all possible coefficients and initial value must be known and stored. During the first signal acquisition, the coefficient pairs can now be determined and stored as known. These coefficients represent the ID of each Lighthouse. In the next incoming bitstream only the coefficinets have to be used to check which Linear Feedback Shift Register configuration is involved and thus the transmitted Lighthouse 2 can be accessed. For this purpose figure 4.4 describes a possible program sequence.

Since the bitstreams unfortunately come sequentially, they have to be buffered in a buffer. For this the first bit is taken and noted. With the next 19 bits, this bit sequence is XORed bit by bit with a possible coefficient. If the result matches the next bit from the buffer, the same is done with the next 19 bits. At the same time, a counter is running. The counter counts up by one if the next bit matches with the calculation. On the contrary, the counter is reset to zero. If successful, the detected sequence is transferred to the next module. The check can continue with 20 digits after the noted position from the buffer. In case of failure, the sequence must be started from the beginning. Important here: If a failure occurs somewhere in the bitstream, it must not be started with the next bit. The next bit follows from the position noted. Now this bit is taken and the check starts from the beginning until a success occurs.

In the next module the distance between the detected sequence and the start sequence of the Linear Feedback Shift Register is determined. The start sequence is taken and XORed bitwise with the coefficient. Here, a counter i is also used to count up until the detected sequence has been successfully reached. With the result from the counter and with the knowledge from the Linear Feedback Shift Register frequency, the position of the lighthouse can now be calculated as follows:

$$s = v \cdot \frac{i}{f} \quad (4.1)$$

where s stands for the distance in the room, v for the speed of the rotor, i for the previously calculated distance and f for the frequency of the LFSR. With regard to real-time application, time constraints should not be a problem. With reference to source [MTT12], a small sample calculation is performed here. According to equation 2.2, the maximum length of possible bit sequences is known:

$$2^{20} - 1 = 1.048.575.$$

The discussion in section 3.1 shows that a bit from Linear Feedback Shift Register needs $0.0833\mu sec$. Thus it takes a maximum

$$t = 0.0833\mu sec \cdot 1.048.575 = 87.34\mu sec$$

until the maximum length of the sequences is calculated. Since this is the maximum time and a sequence is detected earlier in most cases, this can be seen as an upper limit. Furthermore it can be assumed that with $87.34\mu sec = 0.00008734sec$ a simple Real-Time condition is present.

In summary, a Real-Time FPGA decoder for determining the position can be realized with all these findings. When designing an FPGA, it should be considered that this is always done in well thought-out and structured modules [Wak06].

4.3 PCB Design

The PCB Design is done according to the requirements. First, all necessary components have to be accommodated. The Arduino MKR Vidor 4000 is the largest

basic component and the rest adapts to the conditions. The next module to be accomodated is the TS4231 with the photodiode. There are two things which must be considered: First, the photodiode must have a free field of view in order to receive the laser signals successfully. Secondly, the wiring must be observed. Poor design of the layers and unfortunate connections can lead to electromagnetic interference resonances. In particular, the TS4231 must be observed and the design guidelines [Tri16a] followed.

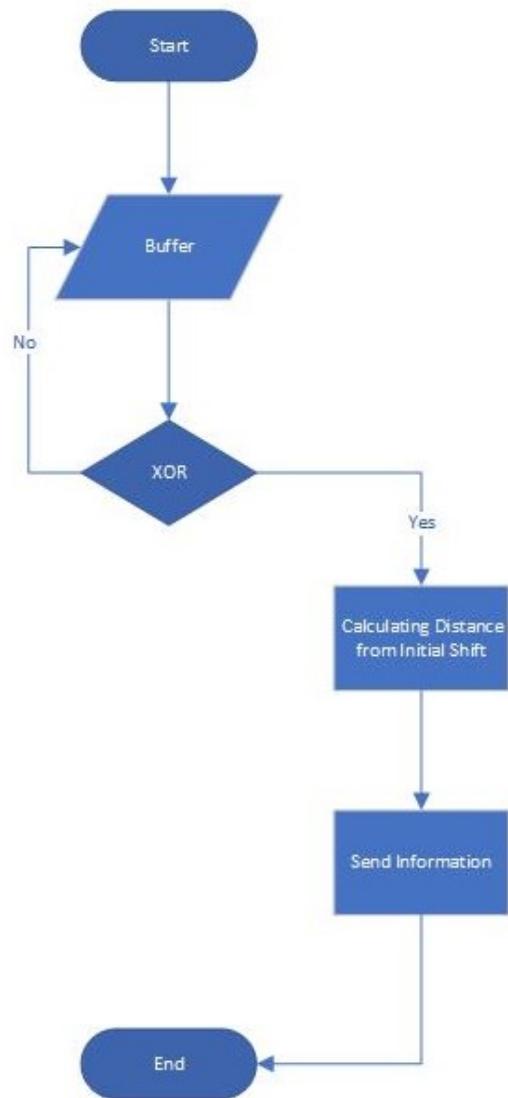


Figure 4.4: An algorithm for position determination

Chapter 5

Results - split observation and interpretation

As mentioned in section 4.1, the measurement data could not be successfully evaluated. There were two Lighthouses 2 available and each time three positions were measured. Thus a total of six different measurement series were recorded under the same critical condition. In addition, one Lighthouse 2 was measured under switched-on light conditions. These results were also not successful. For all measurement series the Gauss elimination was calculated without success (more detailed explanation in section 4.1).

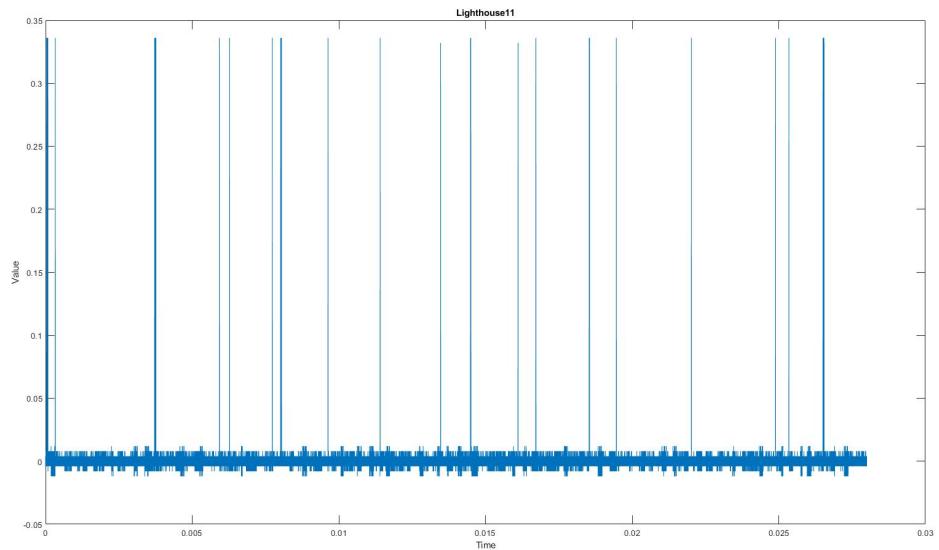


Figure 5.1: Lighthouse 1 in Position 1: a plot of the dat-file in Matlab

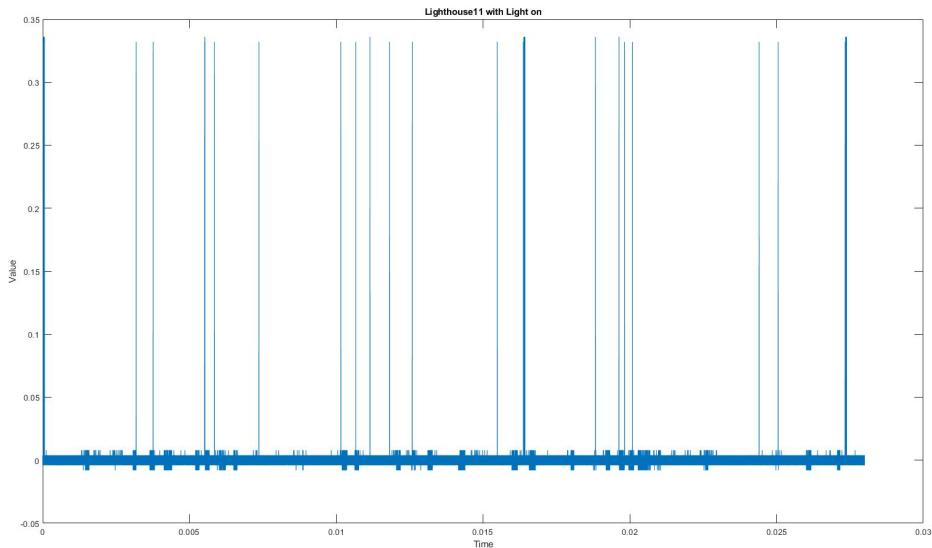


Figure 5.2: Lighthouse 1 in Position 1 with Light switched-on: a plot of the dat-file in Matlab

Figures 5.1 to 5.3 show measurement peaks belonging to the digital signal level of the TS4231, but a connection like in the discussion 3.1 could not be found. Since there is no official documentation from the manufacturer, this was the only source of information with which ideas could be collected and tested.

During the test recordings, the proposal to measure different positions also came up. The aim was not only to have one measurement series from one position, but several measurement series. Systematic errors were also avoided.

An attempt was also made to evaluate the materials available from the section 3.1. The figure 5.4 does not show any useful information either. Therefore it could not be checked which differences existed.

Another attempt to write down the housing and to measure the signals directly from the laser was not possible. Unfortunately it is only possible to write on one side of the case, but not to remove the PCB. In order to get to the board with all components, the Lighthouse 2 has to be destroyed as shown in this video [d]. There is also the question of further use of the device. Thus this possibility was not considered early on.

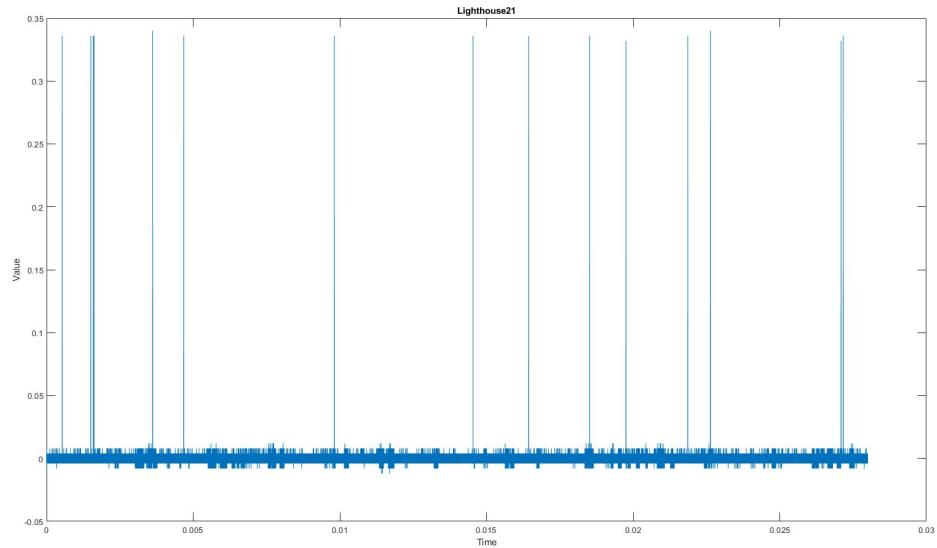


Figure 5.3: Lighthouse 2 in Position 1: a plot of the dat-file in Matlab

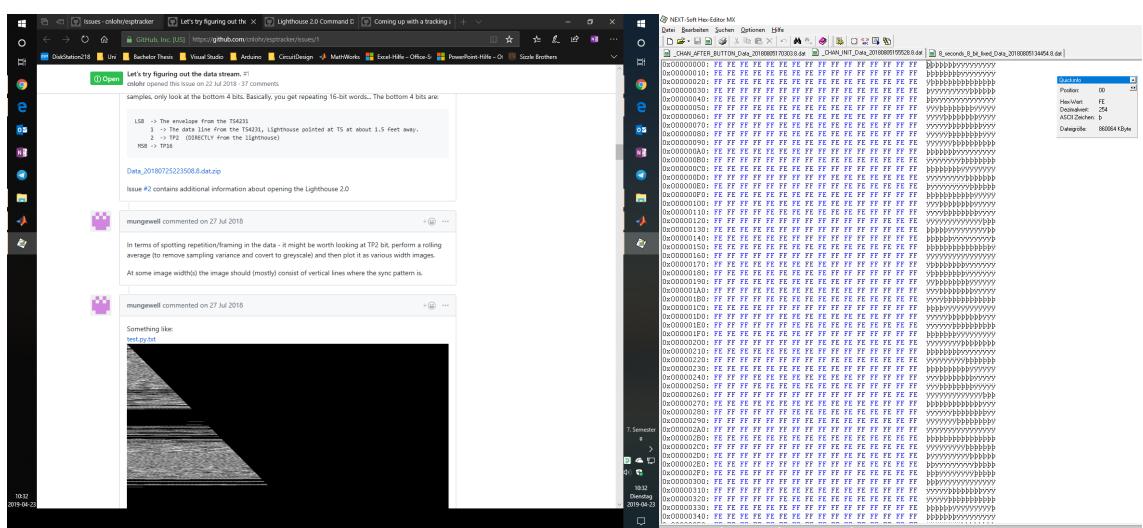


Figure 5.4: Evaluation of the provided dat-file from discussion 3.1

Chapter 6

Discussion

To prevent a completely new development for the Tracking Suite, it is possible to adopt the needed equipment on available products on the market.

In order not to have to take responsibility for entire developments, already used systems can be adopted on the market. The Lighthouse 2 was intended to provide a valuable basis for a tracking suite. It is a well thought-out concept that can be used successfully. With the development of a decoder the Lighthouse 2 would have been usable.

Unfortunately, the Lighthouse 2 is not an in-house development, but a partial use. This has the disadvantage that future software updates from the manufacturer can lead to major changes. So the functionality of the coder depends on the comming features. The manufacturer Valve invited to a press conference on 1.5.19 to talk about changes [.2019]. Due to such small changes, there is always a residual risk that the Tracking Suite will have to be adapted to future market developments.

A risk assessment must be considered after every success and especially after every failure. In the process of the developement it had to be decided whether the housing of the Lighthouse should be irreversibly damaged in order to decode the signal of the laser or not. It does not make sense to be able to decode the signal and then not be able to use the Lighthouse anymore. With the reverse engineering valuable time should be saved and thus a faster use of the system should be pursued.

In order to develop a decoder, it is necessary to analyse the corresponding remote station in detail. If the function of the opposite side is not known, no signal can be successfully decoded. This makes it so difficult to use a Lighthouse 2 in its own environment.

Chapter 7

Summary

The Lighthouse 2 is a very interesting device. Especially since the use of lasers brings great advantages in precision and is trend-setting. However, the laser signals could not be decoded. Decoding without determining the coded signal proves to be extremely difficult. Due to lack of documentation from the manufacturer and no other suitable possibilities for measuring the bitstream, this was not possible.

In addition, a Roboyan will probably develop a better system. This makes it obsolete to have two applications with the same functionality. In most cases it rarely makes sense to own and use two different systems at the same time.

List of Figures

| | | |
|-----|---|----|
| 2.1 | A Schematic of a Linear Feedback Shift Register [PP16] | 10 |
| 2.2 | Communication Link [HKS20] | 11 |
| 2.3 | Simplified Circuit Schematic of TS4231 [Tri16b] | 12 |
| 4.1 | A picture of the inside of the Lighthouse after the destruction of the front [.b] | 19 |
| 4.2 | A picture of the test arrangement | 20 |
| 4.3 | Screenshot on the oscilloscope | 21 |
| 4.4 | An algorithm for position determination | 22 |
| 5.1 | Lighthouse 1 in Position 1: a plot of the dat-file in Matlab | 24 |
| 5.2 | Lighthouse 1 in Position 1 with Light switched-on: a plot of the dat-file in Matlab | 25 |
| 5.3 | Lighthouse 2 in Position 1: a plot of the dat-file in Matlab | 26 |
| 5.4 | Evaluation of the provided dat-file from discussion 3.1 | 27 |

Bibliography

- [.] Coming up with a tracking algorithm. URL: <https://github.com/cnlohr/esptracker/issues/4>.
- [.2019] Base station 2.0 and controller 2.0, where are they?, 2019. URL: <https://community.viveport.com/t5/General-Vive-Discussion/Base-Station-2-0-and-Controller-2-0-Where-are-they/td-p/27446>.
- [.b] Let's try figuring out the data stream. URL: <https://github.com/cnlohr/esptracker/issues/1>.
- [Bos13] Martin Bossert. *Kanalcodierung*. De Gruyter, München, 3., überarbeitete aufl. edition, 2013. URL: http://www.degruyter.com/search?f_0=isbnissn&q_0=9783486755169&searchTitles=true, doi:10.1524/9783486755169.
- [.c] Lighthouse 2.0 command dump. URL: <https://github.com/cnlohr/esptracker/issues/3>.
- [Cyp28] Ez-usb fx3TM superspeed usb 3.0 peripheral controller, 2019-04-28. URL: <https://www.cypress.com/products/ez-usb-fx3-superspeed-usb-30-peripheral-controller>.
- [.d] Tearing down a vive lighthouse 2.0. - youtube. URL: <https://www.youtube.com/watch?v=JP5maGaumYo>.
- [.e] Arduino mkr vidor 4000. URL: <https://store.arduino.cc/mkr-vidor-4000>.
- [.f] Dslogic plus – dreamsourcelab. URL: <https://www.dreamsourcelab.com/product/dslogic-plus/>.
- [HKS20] J. Hagenauer, G. Kramer, G. Söder, and Kötter R. Skriptum zur vorlesung nachrichtentechnik 1. 2017-04-20.
- [Int] Intel cyclone 10 lp fpgas product table. URL: <https://www.intel.de/content/dam/www/programmable/us/en/pdfs/literature/pt/cyclone-10-lp-product-table.pdf>.

- [Int17] Intel cyclone 10 lp device overview, 2017. URL: <https://www.intel.de/content/dam/www/programmable/us/en/pdfs/literature/hb/cyclone-10/c10lp-51001.pdf>.
- [Mic18] Sam d21 family data sheet, 2018. URL: <http://ww1.microchip.com/downloads/en/DeviceDoc/SAMD21-Family-Datasheet-DS40001882D.pdf>.
- [MTT12] Shivshankar Mishra, Ram Racksha Tripathi, and Devendra Kr. Tripathi. Implementation of configurable linear feedback shift register in vhdl. In *2016 International Conference on Emerging Trends in Electrical Electronics & Sustainable Energy Systems (ICETEESES)*, pages 342–346. IEEE, 2016-03-11 - 2016-03-12. doi:10.1109/ICETEESES.2016.7581406.
- [PP16] Christof Paar and Jan Pelzl. *Kryptografie verständlich: Ein Lehrbuch für Studierende und Anwender*. eXamen.press. Springer Vieweg, Berlin and Heidelberg, 2016. URL: <http://dx.doi.org/10.1007/978-3-662-49297-0>, doi:10.1007/978-3-662-49297-0.
- [Sei] John Seibel. How to configure the ts4231 with arduino. URL: <http://help.triadsemi.com/ts4231-light-to-digital-ic-with-data-output/how-to-configure-the-ts4231-with-arduino>.
- [Tri16a] Ts4231 design guidelines, 2016. URL: <https://www.triadsemi.com/wp-content/uploads/2017/06/TS4231-Design-Guidelines.pdf>.
- [Tri16b] Ts4231 light to digital converter product datasheet, 2016. URL: <https://www.triadsemi.com/wp-content/uploads/2017/06/TS4231-Datasheet.pdf>.
- [TSG16] Ulrich Tietze, Christoph Schenk, and Eberhard Gamm. *Halbleiter-Schaltungstechnik*. Springer Vieweg, Berlin and Heidelberg, 15., überarbeitete und erweiterte auflage edition, 2016.
- [Vis17] Bpw34, bpw34s, 2017. URL: <https://www.vishay.com/docs/81521/bpw34.pdf>.
- [Wak06] John F. Wakerly. *Digital design: Principles and practices*. Pearson Prentice Hall, Upper Saddle River, NJ, 4. ed. edition, 2006.

License

This work is licensed under the Creative Commons Attribution 3.0 Germany License. To view a copy of this license, visit <http://creativecommons.org> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.