Delft University of Technology
Master's Thesis in Embedded Systems

Leveraging VLC for energy disaggregation in Smart Buildings

Johnny Verhoeff

embedded
software

TU Delft Delft University of Technology

Leveraging VLC for energy disaggregation in Smart Buildings

Master's Thesis in Embedded Systems

Embedded Software Section
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology
Mekelweg 4, 2628 CD Delft, The Netherlands

Johnny Verhoeff
j.s.c.j.verhoeff@student.tudelft.nl

19th October 2016

Author
Johnny Verhoeff (j.s.c.j.verhoeff@student.tudelft.nl)
Title
Leveraging VLC for energy disaggregation in Smart Buildings
MSc presentation
24th October 2016

Graduation Committee
Chair: Prof. dr. K.G. Langendoen, Faculty EEMCS, TU Delft
Committee member: dr. ir. L.M. Ramirez Elizondo, Faculty DCE&S, TU Delft
Committee member: dr. ir. M.A. Zúñiga Zamalloa, Faculty EEMCS, TU Delft

Abstract

Energy consumption is an ever pressing issue. Therefore there is a need to know what that energy is being used for. Smart-meters can already disaggregate home appliances by looking at the distinct energy signatures of appliances. But other appliances such as lighting has not been successful yet in disaggregating on a per lamp basis, due to the similar energy signatures lighting has. In this work IDs are assigned to lights which are transmitted via VLC. The modulation of the ID will also propagate through the current that the lights draw. To this purpose coding sequences were investigated that would allow for disaggregation of these lights by looking at the aggregated energy use, so that individual lights can be identified as being on or off. With piggybacking on VLC, each light will get a unique ID, that they can transmit via VLC and the same ID will also be transmitted through the power they draw due to the modulation. Hardware was designed, specifically to let that ID map into the energy signature. And experiments were performed with commercial LEDs. With a setup of multiple LEDs, each LED could be successfully identified as being on or off, in both DC and AC environments. Furthermore, simulations have been performed in order to show the scalability of the proposed method and multiple items are highlighted that could be improved.

Preface

This Master thesis is the final part of the Master of Science in the Embedded Systems program I followed at Delft University of Technology. Prior to this thesis, I had little knowledge of VLC or energy disaggregation. Using VLC in combination with energy disaggregation has not been explored yet. The first steps towards disaggregating individual lights are taken in this thesis. There are experimental results achieved as well as theoretical results.

First, I would like to thank my loving family, who have supported me throughout this nine-month thesis project. I would also like to thank my advisor Marco Zúñiga Zamalloa and Akshay Narashiman for their guidance to help me finish my thesis. Finally I want to acknowledge Koen Langendoen and Laura Ramirez Elizondo for being members of my graduation committee.

Johnny Verhoeff

Delft, The Netherlands
19th October 2016

Contents

Preface v

1 Introduction 1
- 1.1 Problem Definition 3
- 1.2 Thesis Contributions 3
- 1.3 Thesis Organization 3

2 Related Work & Proposed Method 5
- 2.1 Related Work 5
- 2.2 Hypothetical Methods 6
  - 2.2.1 Power-line Communication 6
  - 2.2.2 Power over Ethernet 7
- 2.3 Proposed Method 7

3 Design Requirements 9
- 3.1 Encoding 9
- 3.2 LED Modulator 11
- 3.3 Smart Meter 11

4 Code Division Multiple Access 13
- 4.1 Performance Metrics of a CDMA Sequence 13
- 4.2 Code Sequences 15
  - 4.2.1 Orthogonal Sequences 15
  - 4.2.2 Pseudorandom Noise Sequences 17
  - 4.2.3 Gold Sequences 20
  - 4.2.4 Comparison of Sequences 23
- 4.3 Interference Solution 24
  - 4.3.1 Continuous Method 24
  - 4.3.2 Probabilistic Method 26
- 4.4 Mapping Problem 28

5 Hardware Design 33
- 5.1 DC Environment 33
  - 5.1.1 Modulator 34

5.1.2 Current Sampler 35
5.1.3 Testbed 36
5.2 AC Environment 38
5.2.1 Modulator 39
5.2.2 Current Sampler 50
5.2.3 Testbed 52

6 Evaluation **55**
6.1 Hardware 56
6.1.1 DC 57
6.1.2 AC 60
6.2 Simulation 63

7 Conclusions and Future Work **69**
7.1 Conclusions 69
7.2 Future Work 69
7.2.1 Power Limiting Capacitor 70
7.2.2 Other Appliances 70
7.2.3 Dimming Lights 70
7.2.4 Transmitting Data 70

A DC Testbed **77**

B Modified 230 V AC LED Schematic **79**

C Custom LED Modulator Schematic **81**

D Custom Current Sampler Schematic **83**

Chapter 1

Introduction

In a world where the vast majority of consumed energy is provided by unsustainable fossil fuels [31], the amount of energy we use must be reduced. Reducing energy consumption requires knowing which devices are actually consuming the energy. When consumers are given feedback about their energy consumption, energy savings can be made (up to 15 % according to [20]). If a consumer knows which devices are actually using energy, he or she can decide if those devices really need to be used at that particular time. If a device is on but it is not being used, the energy used to power it, is essentially wasted. For example, lights on in a room which is not occupied.

Current energy meters give an aggregate power consumption, which cannot tell a consumer which specific device is responsible for the observed energy consumption or waste thereof. This is where energy disaggregation comes in. Energy disaggregation aims to break down the aggregate power consumption of, for example a household, to tell the consumer which devices consumes power at which times.

Energy disaggregation has been applied successfully to identify the operation of appliances, such as refrigerators and HVAC [31] [40]. Identifying the energy consumption of other devices such as lighting has not been so successful [40] [18]. Each appliance in a household draws power in a certain way, this can be thought of as a signature of that appliance. As can be seen from Figure 1.1, appliances such as the washer dryer and the dishwasher can be distinguished from the aggregated power draw. These appliances can be recognized by their signature: the amount of time they draw power, how

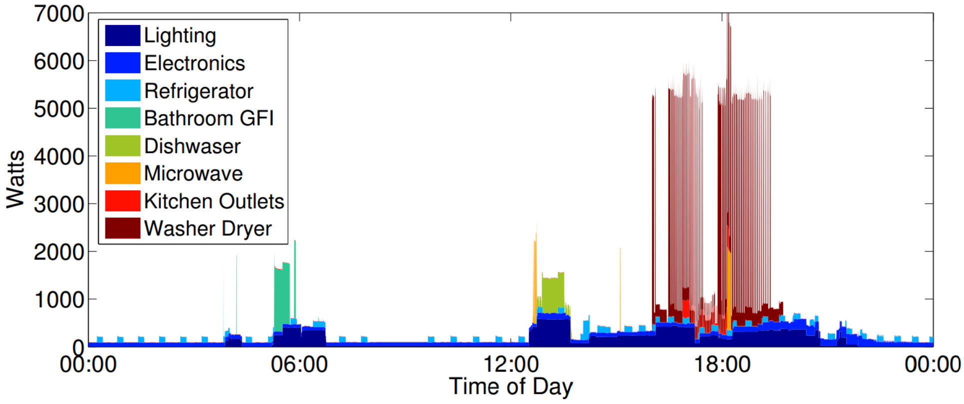


Figure 1.1: An example of energy consumption of a household over the course of a day [31].

large the power draw is and if it has a recurring pattern. For example, the refrigerator can be easily recognized due its periodic pattern, captured by the light blue peaks in Figure 1.1. But the lighting can not be disaggregated on a per lamp basis. The reason for this, is that most lighting fixtures do not have a unique signature, instead many lights have the same signature. Disaggregating the energy consumed by lighting is important, because it is the third largest energy consumer in the average home [18]. In buildings, lighting consumes around 30 % of the total energy consumption [26]. In the EU as well as in the USA, approximately 11 % of all the energy produced is used only for lighting [19] [35].

If every light in a household would have a unique power draw that is distinguishable from all other lights, the consumer can get better insight about how lighting is being used in his household. With this feedback, a consumer can see which individual lights are being used in the middle of the day for example or in an unoccupied room. The consumer can then turn the lights off. Thereby saving energy and monetary costs.

Without adding extra infrastructure, giving each light a unique signature is an almost intractable problem. But the advent of smart-lights can help with this. Smart-lights can use VLC (Visible Light Communication). VLC is a short-range optical means of wireless communication using the visible light spectrum. It modulates data by turing the light on and off at high frequencies so that no flickering effects can be seen by the human eye. When a light is transmitting data via VLC, the current signature of the light changes. The current signature of a light can then be decoded with a smart-meter. And in this way, real-time information of lights in the home building is acquired.

1.1 Problem Definition

Energy disaggregation can estimate the energy consumption of individual devices based on the distinct power draw that these devices have. What it cannot yet do, is the disaggregation of appliances that have very similar power draw, such as lighting. With the advent of VLC, we pose the following question: Can a smart-meter identify lighting as being on or off on a per lamp basis ?

1.2 Thesis Contributions

The aim of this thesis is to propose a framework consisting of theoretical methods, hardware and software, so that lights with similar power draw can be distinguished with a single smart-meter. This thesis takes advantage of advances in two areas: CDMA codes and VLC.

The specific contributions of this thesis are:

- Coding methods are analyzed to allow each LED to have a unique power draw. These codes make it possible to identify if an LED is on or off, even when multiple LEDs are modulating and thereby interfering with each others unique current signature.
- Hardware is introduced to allow the LEDs to be modulated by a microcontroller. This will allow the LEDs to propagate their unique signatures via either AC or DC powerlines. We also develop a smart-meter that is sensitive enough to sample the high frequency and low energy consumption of the signatures of the LEDs.
- An evaluation of the proposed hardware and software is carried out, with a testbed which uses standard LED light fixtures. For larger scale evaluations, software simulation is used.

1.3 Thesis Organization

The remainder of this thesis is organized as follows. First the related work and the new proposed method is discussed in Chapter 2. Then the design requirements are outlined in Chapter 3. In Chapter 4 the theory about code sequences is explained. Next, in Chapter 5, the existing hardware and the design of the new hardware is discussed. In Chapter 6 the hardware and software is evaluated for small scale scenarios and simulations are shown for large scale scenarios. Finally the work is concluded in Chapter 7 and future work is identified.

Chapter 2

Related Work & Proposed Method

This chapter first describes the state-of-the-art in energy disaggregation research and their results with the disaggregation of individual lights. Next, other methods are discussed, which could hypothetically identify if a light is on or off. And finally the proposed method is introduced.

2.1 Related Work

Energy disaggregation can categorize devices such as a refrigerator or washer dryer, but it cannot disaggregate two devices which have the same signature. For example: The same lights in different rooms could not be disaggregated in the work performed by the authors of [23]. In other words, if there is more than one light in a house, and a subset of those lights are on, it cannot be identified which subset of lights are actually on, only that there are some lights on.

The authors of [38] have shown that their method is able to disaggregate up to three separate lights. The key difference here is that the lights have different power ratings. Because the lights were different and had different power ratings, these lights could be disaggregated from each other. Nothing is mentioned about how the disaggregation results would change, if there were multiple devices used with the same power ratings, for example three identical lights.

In [25] the authors tried to disaggregate appliances which were powered with a switching mode power supply (SMPS). These power supplies continuously generate high frequency electromagnetic interference (EMI). Appliances can be distinguished based on the differences between the switching frequencies characteristics of each SMPS. The method of disaggregating appliances which are powered with a SMPS requires training data of the characteristics of each SMPS. Many lights also use some kind of SMPS, due to the high efficiency of these types of power supplies. It is claimed that two appliances of the same brand and model can be distinguished by looking at the EMI from the SMPS. However it was also observed that the system could not distinguish similar light fixtures when they were installed spatially near each other. The EMI from the power supplies did not have sufficient differences to correctly distinguish between these similar lights.

In [22] the authors present a method which correlates the light switching events obtained from a Building Control Network (BCN) with the power measurements on the mains distribution level. A BCN can manage the lights and HVAC in a building. It can turn the lights on or off depending on the time of day, for example. So this work already knows which lights are on and off. This work uses the combined sources of the BCN and the power measurements to estimate the electricity consumption for each light individually, with an error below 11 %. They use the BCN to retrieve which lights are being switched on or off and then they look at how the power consumption of the building changes. In this way they can estimate what the power consumption of each individual light is, without having explicit training or manual inputs. The system adapts itself to the characteristics of any building within 8 to 14 days.

2.2 Hypothetical Methods

In this section other methods will be discussed that can potentially identify which lights are on and off, by other means than looking at the energy signature.

2.2.1 Power-line Communication

Power-line communication or PLC is a communication method that uses the existing AC wiring to simultaneously carry both data and AC power [32]. The obvious benefit is that the existing wiring infrastructure can be used. Data is transmitted with a frequency that is much higher than the

50 or 60 Hz AC power frequency in order to ensure that the power wave does not interfere with the data signal. Each light should be equipped with a modulator that can transmit the information about a light over the powerline. Then a receiver, can decode the information that is transmitted by the modulators. Multiple lights can then transmit their status via for example orthogonal frequency-division multiplexing (OFDM) [27].

2.2.2 Power over Ethernet

Power over Ethernet or PoE is a standard which passes DC power along with data on an Ethernet cable [36]. With this solution each light becomes a node in a network with an Ethernet cable being connected to the light. The PoE technique provides the power to use the light and the Ethernet standards allow a micro-controller to transmit status information about that light to a central server, for example. The number of lights that can transmit their status is limited by the network protocols used. A drawback is that an Ethernet cable must be used for each light instead of the existing AC wiring. Another drawback is that PoE provides low voltage DC power and so the power dissipated in the cable itself will also be greater than with the existing high voltage AC wiring.

2.3 Proposed Method

If a building uses PoE, the monitoring of lights is trivial. But many legacy buildings do not have this infrastructure. The proposed method aims to still work in legacy buildings, so it must use the legacy AC wiring that is already present. Since the method is aimed at legacy buildings, we cannot rely on a Building Control Network (BCN). A single smart-meter is used to measure the energy usage. The purpose of the smart-meter is to identify which lights are on and which are off. It does this by looking at the the energy signature of each light. But as discussed many lights have the same signature, so it is not possible to accurately identify which lights are on using only a smart-meter.

So besides a smart-meter, smart-lights will also be used. These smart-lights can take advantage of Visible Light Communication (VLC). VLC is a short-range optical means of wireless communication using the visible light spectrum. VLC is made possible by the advances in LED technology. VLC can turn the lights on or off at high frequencies to avoid seeing flickering effects. The interest in VLC has grown since the widespread deployment

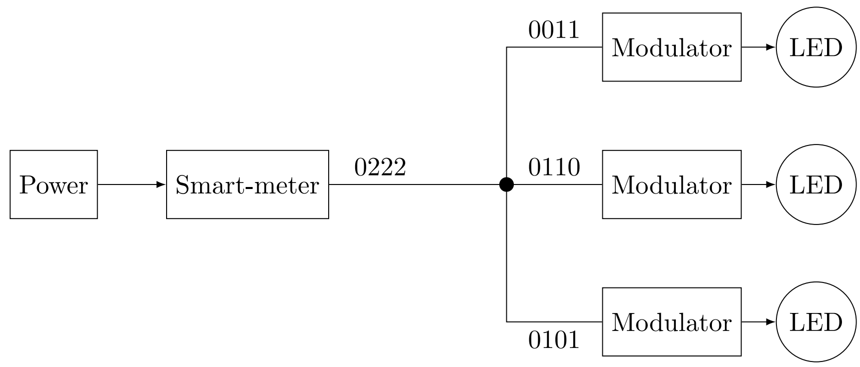


Figure 2.1: Block diagram how each component interacts with other components.

of LED lighting fixtures for energy efficiency over the normal incandescent light bulb [37].

Visible light communication opens up the opportunity to monitor the energy consumption of lights. Applications such as indoor localization are becoming widely popular with VLC and these applications require lights to broadcast a unique ID. Given that VLC modulates LEDs by turning them on and off, the ID that is used for indoor localization can also be used for detecting if the light is on or off by looking at the current signature.

The information that a light is transmitting via VLC, will also propagate to the current that the light draws. And with the smart-meter, we are looking at the aggregated current draw of all attached devices, including these lights. The information or ID that the light will be transmitting, needs to be constructed in such a way that the sum of all these currents, which flows through the smart-meter, can still tell us which lights are actually on or off. How these IDs can be constructed needs be to be investigated. With this proposed method the lights can still be used as beacons for applications such as indoor localization because the IDs will still be unique. The current signature of each light will tell us which lights are on and it can be done over the existing wiring infrastructure.

In Figure 2.1 an overview of the method can be seen. The smart-meter is shown along with multiple smart-lights. Each smart-light is a standalone device which has the necessary embedded hardware to modulate the light. The IDs that will be used can also be seen next to the smart-lights. The sum of these IDs will then flow as current through the smart-meter, which will sample this aggregated current and in turn be able to identify which of the LEDs is on and which is off.

Chapter 3

Design Requirements

In this section we introduce the main requirements for our system. The overall architecture has three main components: the codes, the modulator and the meter. These components were shown in Figure 2.1. In this figure multiple lights can be seen. We want to monitor which lights are on and off in a timely manner, by only looking at the energy signature of each LED. To do this, three things must be investigated: How to construct the IDs, what hardware to use for the modulators and how to sample and process the current. First, the requirements for the IDs will be discussed. Next, the modulator is discussed. This piece of hardware will be responsible for the transmission of the ID via VLC. This data can be used for applications such as indoor localization. This hardware is also responsible for translating the ID into a current signature. Finally, the smart-meter will be discussed. The meter must sample the current that is drawn by all the LEDs. And process that data into status indicators for each LED.

3.1 Encoding

To be able to distinguish multiple lights from each other, which are all connected in parallel, each light must have a unique identification sequence of some sort. Furthermore, that identification sequence must somehow be detected and interpreted by a smart-meter.

The most common way to use VLC with LEDs, is to use on-off keying (OOK). OOK works by turning the LED on or off. If a data bit '1' needs to be transmitted, the LED is turned on. If a data bit '0' needs to be transmitted, the LED is turned off. This is how the information is transmitted. Since the LED is turning on and off according to the ID in an OOK fashion, this ID will also propagate in the current that this LED draws. This unique current signature flows through the smart-meter. If only a single light is used with an ID, the meter can search for only that ID. If it finds that ID, the light is on, else the light is off.

A problem rises when more than one light source is sending its identification sequence. Since the lights are connected in parallel, the current that flows through the smart meter will be the sum of all the currents that are drawn by all the light sources. This means that the light sources, which are effectively transmitters, interfere with each other. Because of that interference the unique identification sequences which are assigned to each light source, need to be carefully selected.

To build the necessary codes we borrow from the field of telecommunications. In that field, similar issues occurred: For example multiple cellphones transmitting to the same base station, at the same time, at the same frequency. The solution was to use code sequences that do not interfere with each other. The specific codes are called Orthogonal codes and Pseudo random noise codes. For our scenario, the codes need to satisfy the following requirements:

- The codes should be detectable with good accuracy:
  - A code sequence must be able to be detected, even when multiple codes are aggregated.
  - The codes must have as little interference as possible with each other, so that a large number of LEDs can transmit at the same time and still achieve accurate results when the smart-meter tries to detect which LEDs are transmitting.
- The codes need to work in a synchronous and asynchronous manner. The synchronous case is represented by scenarios where multiple in a single room can be all switched on together. But when there are multiple lights in multiple rooms they need not be turned on or off at the same time, this is the asynchronous case.
- The system must be scalable:
  - The codes should not be too long, because the system must identify each LED as being on or off in a timely manner. The longer the code, the longer it takes to transmit this code.

- The number of codes that can be used must scale well. In other words: The number of codes that can be constructed, should be proportional to the length of the codes.

The exact properties, benefits and drawbacks of the codes and how they can be used in both DC and AC environments will be discussed in Chapter 4.

3.2 LED Modulator

A piece of hardware is needed to modulate the commercial LED. This hardware needs to translate the unique identification sequence that is assigned to each light source and modulate the LED. Contrary to standard VLC, which is only concerned with modulating the light intensity to transmit data, the same data must also be transmitted via the current draw. The modulator must not only turn the light on or off, but it must also make sure that the current that is drawn can be decoded later on by the smart-meter. The hardware should also allow for fast modulation to avoid seeing flickering effects.

For the design of this hardware, or when using pre-designed hardware, the way the identification sequence translates to the current draw needs to be taken in mind. Since OOK is used, the modulator should ideally draw zero current when a ‘0’ data bit is transmitted and draw a certain amount of constant current when a ‘1’ data bit is to be transmitted. This current draw translation should be the case irrespective of using DC or AC. In a DC environment the modulator hardware will get a constant voltage, so there are no extra challenges. But in an AC environment the modulator will get an alternating voltage, which is not constant. This will introduce multiple challenges, which will be discussed in section 5.2.

3.3 Smart Meter

The smart meter needs to be able to detect relatively small current changes. More concretely, it needs to be able to detect the current change when even a single light is modulating.

When selecting a method of measuring the current these points need to be taken into consideration:

- The speed at which the current can be sampled needs to be high enough to be able to correctly sample the current as the LEDs are modulating.
- The noise introduced in the samples must be low enough to detect the correct modulation of even one LED with consistency.
- The sensitivity of the meter must balance a tradeoff between the ability to detect an LED and the likelihood of getting saturated.

Chapter 4

Code Division Multiple Access

Each LED requires a unique ID. When the LEDs are modulating to transmit their IDs, the current of multiple LEDs are aggregated. The aggregated current will flow through the smart-meter, in other words, the only information the smart-meter has, is the aggregated current. And from these aggregated IDs, the smart-meter should be able to identify which IDs are present. If it can do that, the smart-meter is able to tell which LEDs are on and which or off.

This chapter will explain how these IDs can be constructed by using CDMA codes. To see which codes are the best, first the performance metrics are explained. Then several codes will be discussed and the performance metrics for each code will be highlighted. Finally the codes will be compared to each other.

4.1 Performance Metrics of a CDMA Sequence

To be able to objectively determine which code sequence is the best for certain environments, metrics are needed to compare the performance of a sequence.

Such metrics are:

- Auto- and cross-correlation: This will determine how accurate we can detect the ID of an LED in the aggregated signal, and also the amount of interference there will be from the other IDs.

- Code length: If the code is too long, the system will not be able to detect an LED in a timely manner.

- Number of codes: The number of unique codes that can be used in the same system is important for the scalability.

- Asynchronous behavior: It should also be considered if the codes can be used in a synchronous only or in an a-synchronous environment.

Correlation is a measure for determining how much sequence X is similar to sequence Y and can be found in Equation 4.1. With L being the length of the code and τ the time-shift. When sequence X and Y are the same sequence, we speak of the auto-correlation. When they are two different sequences, we speak of the cross-correlation.

$$R(\tau)_{xy} = \sum_{i=0}^{L-1} x(i) \times y(i+\tau) \text{ with } \tau = 0, 1, 2, \ldots, L \tag{4.1}$$

The properties of an ideal set of codes should be, that the auto-correlation for each code should have a clear peak to identify that this code is present in the signal. This peak should only occur when there is no time-shift, so at $\tau = 0$. The value of this peak should then be L, meaning that each chip of the code is equal to the value in the received signal. When there is a time-shift, $\tau \neq 0$, the auto-correlation should be as close to zero as possible. If the signal is the sampled current and the code is the ID of an LED, then we can say the LED is on when the auto-correlation peak is seen. The ideal cross-correlation properties should be 0 for every time-shift τ, so that no code interferes with any other code, hereby causing no MAI (Multiple Access Interference).

The length of a code is also of importance, because each chip of the code has to be transmitted. Assuming a constant modulating frequency, the time it takes to transmit a code sequence is proportional to the length of that code sequence. The length of the code will also determine, to some extent, the number of codes that can be used together in the system. These codes are said to be in the same set. The number of codes in the same set determines the scalability of the system.

4.2 Code Sequences

In this section different code sequences will be explained. Their performance will be measured by the metrics detailed in section 4.1. Finally a comparison between the sequences is made.

4.2.1 Orthogonal Sequences

Orthogonal sequences, also known as Walsh-Hadamard sequences, are sequences which are created using a Hadamard matrix. Hadamard matrices are square $n \times n$ matrices which are recursively generated. Starting with a 1×1 matrix: $H_1 = \begin{bmatrix} 1 \end{bmatrix}$, then $H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$. See Equation 4.2 for a general recursive formula to generate other ranks of Hadamard matrices [21].

$$H_{2n} = \begin{bmatrix} H_n & H_n \\ H_n & -H_n \end{bmatrix} \tag{4.2}$$

The matrix can also be filled with binary values: 0 and 1. In that case the general recursive formula is stated in Equation 4.3.

$$H_{2n} = \begin{bmatrix} H_n & H_n \\ H_n & \overline{H_n} \end{bmatrix} \tag{4.3}$$

The Hadamard matrix has the property that every row in the matrix, apart from the first row, is orthogonal to every other row, meaning that the cross-correlation is zero. And apart from the first row, all other rows have the exact same number of $+1$s and -1s, meaning that the codes are balanced.

Hadamard matrices exist for every power of 2, so the code length is also a power of 2. For $\tau = 0$, the cross-correlation is 0, but when $\tau \neq 0$ not all the rows have a zero cross-correlation with all other rows. All rows of the matrix have the property that the auto-correlation at $\tau = 0$ is equal to L. But when $\tau \neq 0$, undesirable behavior occurs as can be seen in Figure 4.1. The autocorrelation function has several high peaks where only one is desired, namely at $\tau = 0$. This means that if an LED starts modulating its code, but the smart-meter does not know when in time the start of this code is, the smart-meter will get false-positives. If there was a way that the

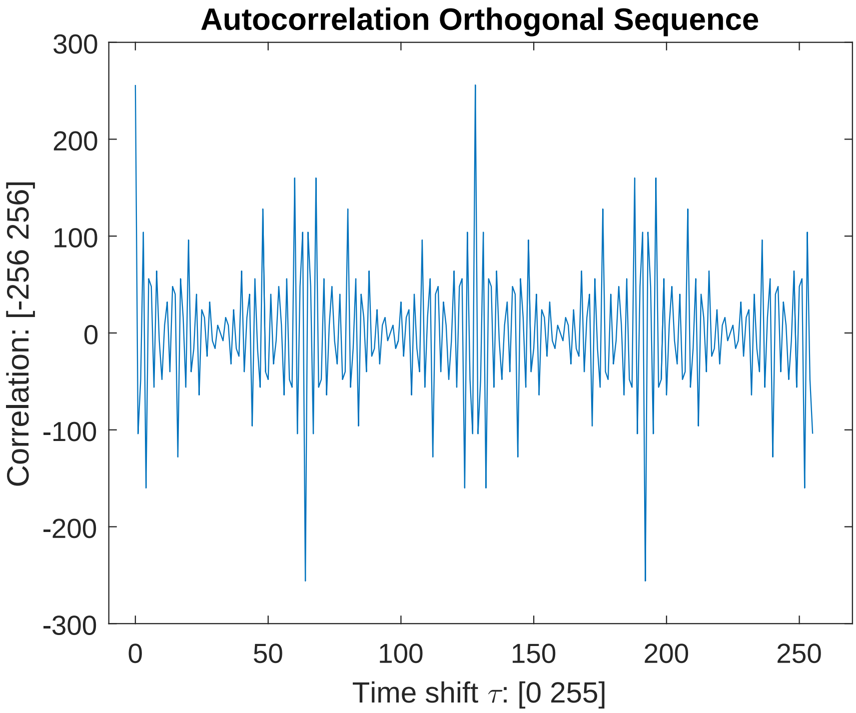


Figure 4.1: Autocorrelation of orthogonal sequence with row index 120 of length 256.

smart-meters and all the LEDs could be synchronized with each other, the orthogonal codes will work.

To conclude: the entire set of orthogonal codes of length L, has $L-1$ codes in the set which does make it a scalable set. But the auto- and cross-correlation only have the desired properties when the codes are sent synchronously.

Cyclically Orthogonal Walsh Hadamard Codes

To overcome the problem when sending orthogonal codes in an asynchronous manner, a subset of the orthogonal codes have been identified that are still orthogonal to each other, no matter how these codes are time-shifted with respect to each other.

In [42] the authors proved that an Hadamard matrix of size 2^P could be divided into $P+1$ subsets of rows, where one row could be selected giving $P+1$ orthogonal rows for each time-shift τ. In other words, this subset of rows has a cross-correlation of zero, for every time-shift τ. These codes are called Cyclically Orthogonal Walsh Hadamard Codes (COWHC). With code length L there are $\log_2 L$ codes in the set, which makes these codes not scalable [42]. Also the auto-correlation does not have a clear peak to identify the code, because these codes are an unmodified subset of the original orthogonal sequences. For these sequences the auto-correlation has already been shown in Figure 4.1.

These codes suffer from the same auto-correlation problems as the normal orthogonal codes as described in subsection 4.2.1 and they have the additional drawback that they are less scalable.

4.2.2 Pseudorandom Noise Sequences

The main drawback with the orthogonal codes, was that these codes cannot be used in an asynchronous manner. To overcome this drawback, PN sequences were investigate which can be used in an asynchronous manner.

PN sequences are sequences which look like they are randomly generated but they are easily generated in software or hardware. They are generated with linear shift registers of length n. The sequences have the following noise-like properties [33]:

- Balance property: Any PN sequence of length $L = 2^n - 1$ contains exactly 2^{n-1} ones and exactly $2^{n-1} - 1$ zeros.
- Runs property: A run is a subset of the sequence where all the consecutive numbers are the same. In any PN sequence, 1/2 of the runs have length 1, 1/4 have length 2, 1/8 have length 3 and so on.
- Auto-correlation property: The auto-correlation function of a PN sequence will take on two values as can be seen in Equation 4.4 and Figure 4.2.

$$R(\tau) = \begin{cases} L & \text{if } \tau = 0 \\ -1 & \text{if } \tau \neq 0 \end{cases} \tag{4.4}$$

PN sequences are generated using a linear feedback shift register (LFSR) [41]. Figure 4.3 shows an n length LFSR with XOR gates attached to each element of the register and a feedback loop into the last element. The

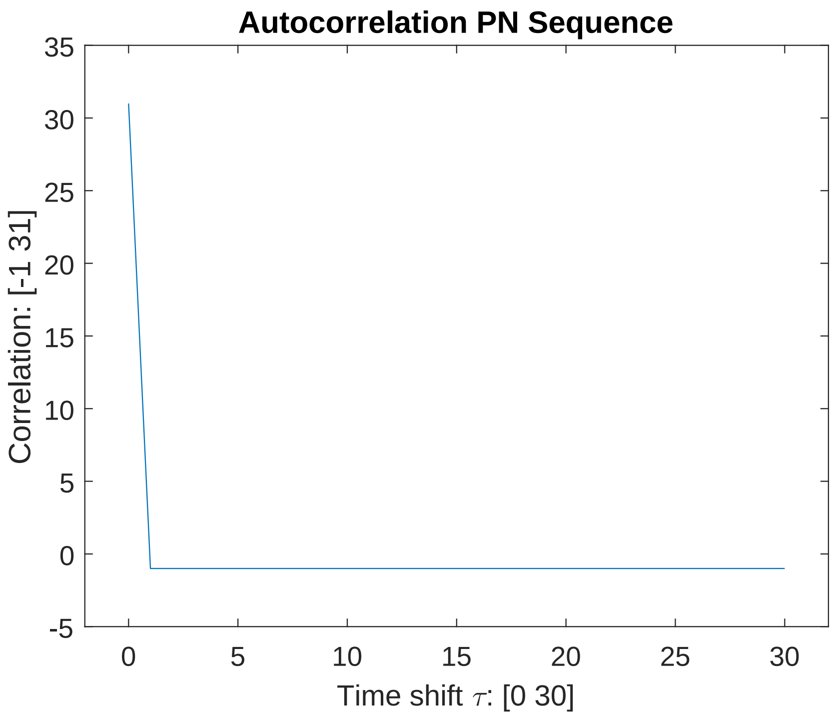


Figure 4.2: Autocorrelation of PN sequence of length 31.

LFSR is defined entirely by the feedback function, also called a characteristic polynomial. It determines the length and the type of sequence generated. The polynomial looks like Equation 4.5. For example: The polynomial $q(x) = x^5 + x^2 + 1$ tells us that the LFSR has length 5 and that the outputs of elements with number zero and two are XORed and fed back into the last element.

$$p(x) = x^n + C_{n-1}x^{n-1} + C_{n-2}x^{n-2} + \ldots + C_2x^2 + C_1x + C_0 \tag{4.5}$$

For PN sequences, there exists no formula for the cross-correlation of two different PN sequences. Exhaustive analysis is required to find out which sequences or entire sets have the cross-correlation characteristics that are good enough for the user's application. In Table 4.1 the calculated peak cross-correlations per PN sequences from the same set can be found. For each sequence the cross-correlation is calculated with every other sequence for every time-shift τ. Then, the maximum of those cross-correlations is taken and used for the peak cross-correlation. For the PN codes the auto-correlation does not have to be calculated, this is already shown in Equation 4.4.

The size of the code set is limited. For a LFSR with n registers, the

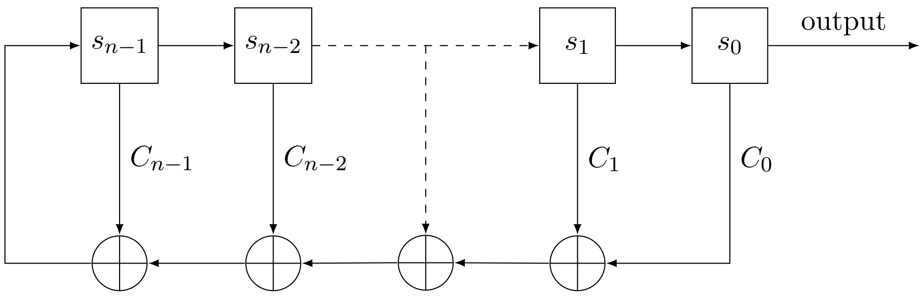


Figure 4.3: Linear feedback shifter register of length n, with XOR gates, to produce a PN sequence.

Code Length (L)	Number of Codes (C)	Peak cross-correlation
7	2	5
15	2	9
31	6	11
63	6	23
127	18	41
255	16	95
511	48	113
1023	60	383
2047	176	287
4095	144	144

Table 4.1: Table showing the number of PN sequences of the same length along with the peak cross-correlation [30].

maximum number of possible codes C is given by Equation 4.6 [34], where P_i are the prime factors of $2^n - 1$ and α_i is the power of ith prime factor.

$$C = \frac{1}{n} \prod \{P_i^{(\alpha_i - 1)} \times (P_i - 1)\} \tag{4.6}$$

For example when using a LFSR of size $n = 6$, $2^n - 1 = 63$, which can be factored into $3^2 \times 7$. Giving $P_1 = 3$, $P_2 = 7$, $\alpha_1 = 2$ and $\alpha_2 = 1$. Thus, the maximum number of codes is: $C = \frac{1}{6} \times \{3^{2-1} \times (3-1)\} \times \{7^{1-1} \times (7-1)\} = 6$. Another example: Say there are going to be 144 LEDs, so 144 codes are needed. This means a code length of 4095 chips. For PN sequences of other lengths, see Table 4.1 for the number of codes.

To conclude: PN sequences have less number of sequences in the same set, compared to orthogonal sequences, so they are less scalable. But the PN sequences can be used in an asynchronous manner, with good auto-correlation. But the cross-correlation properties causes interference, which limits the amount of transmitters that can transmit concurrently.

4.2.3 Gold Sequences

The PN sequences gives us a set of codes which is not scalable and suffers from interference issues, but it can work in an asynchronous environment. To solve these issues, Gold codes were investigated.

Gold sequences are a type of PN sequence. They are created by using two LFS registers as shown in Figure 4.4. In this figure there are two vectors s and t and two polynomials C and D. For a sequence to be constructed in this way that is a Gold sequence, only preferred pairs of polynomials can be used [29]. Only preferred pairs of polynomials will yield a set of codes that will have a three valued cross-correlation function. A method for selecting these preferred pairs of polynomials can be found in [24]. The polynomials are the same as explained for the PN sequences. For example, $p(x) = x^3 + x^2 + 1$ and $q(x) = x^3 + x + 1$ are a preferred pair and can be used with two LFS registers of length 3 in a construction as shown in Figure 4.4. When we have the polynomials $p(x)$ and $q(x)$, which are a preferred pair and produce the PN sequences d_1 and d_2 respectively, the resulting set of Gold codes is defined as can be seen in Equation 4.7, where T^k represent the cyclic shift of k bits.

$$Gold(d_1, d_2) = \{d_1, d_2, d_1 \oplus d_2, d_1 \oplus Td_2, d_1 \oplus T^2 d_2, \ldots, d_1 \oplus T^{L-1} d_2\} \tag{4.7}$$

The auto-correlation properties of Gold sequences are not as good as that of the PN sequences, as can be seen from Figure 4.5 compared to Figure 4.2. The auto-correlation of all the Gold sequences are plotted in Figure 4.6 to illustrate that the auto-correlation of all sequences look like each other. Apart from the original two PN sequences the auto-correlation values can take four values. The auto-correlation of a PN sequence can be one of two values, see Equation 4.4. See Equation 4.8 and Equation 4.9 for the auto-correlation properties of Gold sequences, where n is the length of the LFSR.

$$R(\tau) = \begin{cases} L & \text{if } \tau = 0 \\ \{-t(n),\ -1,\ t(n) - 2\} & \text{if } \tau \neq 0 \end{cases} \tag{4.8}$$

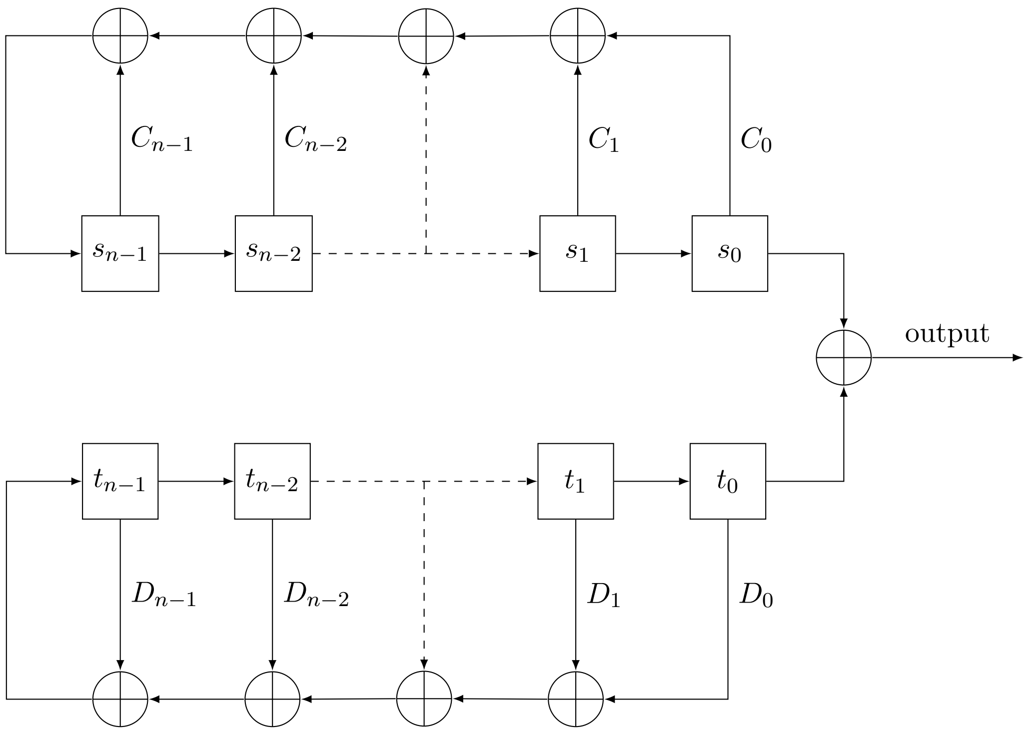


Figure 4.4: Two linear feedback shifter registers of length n, with XOR gates to produce a set of Gold sequences.

$$t(n) = \begin{cases} 1 + 2^{\frac{n+1}{2}} & \text{for odd } n \\ 1 + 2^{\frac{n+2}{2}} & \text{for even } n \end{cases} \tag{4.9}$$

See Equation 4.10 and Equation 4.9 for the cross-correlation properties of Gold sequences [33].

$$R_{xy}(\tau) = \{-t(n), -1, t(n) - 2\} \tag{4.10}$$

From these equations it is clear to see that the absolute maximum cross-correlation is bounded by $t(n)$. See Table 4.2 for the cross-correlation for different lengths of Gold sequences.

As seen from Table 4.2, the number of codes scales linearly with the code length, so the gold codes are scalable. With a LFSR of length n, the sequence length is $L = 2^n - 1$ and the number of sequences is $C = 2^n + 1$. Compared to the peak cross-correlation of the PN sequences, Table 4.1, the peak cross-

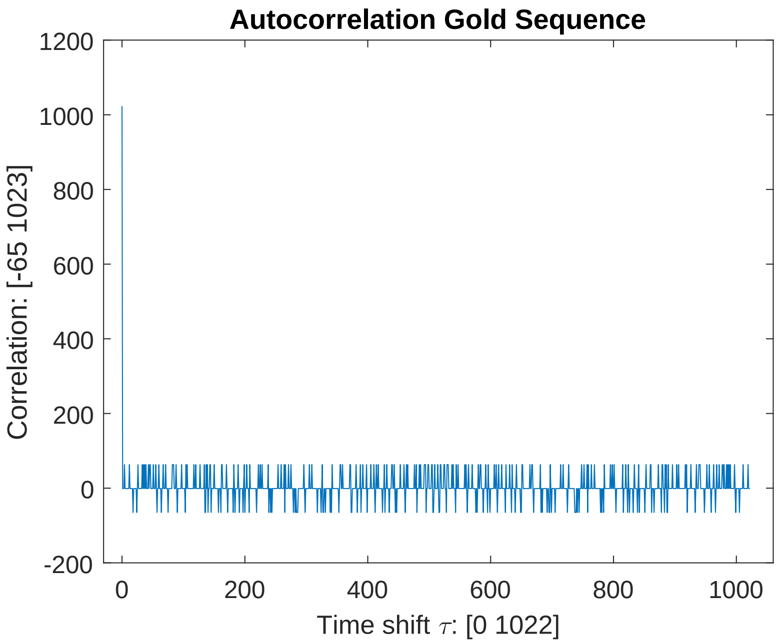


Figure 4.5: Autocorrelation of one Gold sequence of length 1023.

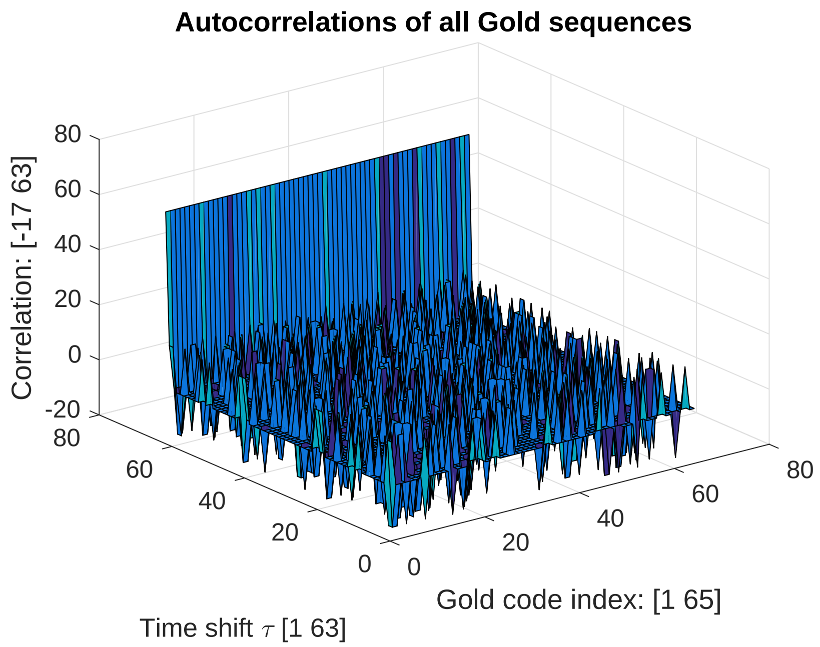


Figure 4.6: Autocorrelation of all Gold sequences in the same set of length 63.

Code Length (L)	Number of Codes (C)	Peak cross-correlation ($\|t(n)\|$)
7	9	5
31	33	9
63	65	17
127	129	17
511	513	33
1023	1025	65

Table 4.2: Table showing the number of Gold sequences of the same length along with the peak cross-correlation.

correlation of the Gold sequences are lower for the same code length, see Table 4.2.

Not all Gold sequences have the balance property that the PN sequences do have. Roughly half of the sequences in the same set are balanced, sometimes this can be as high as three quarters [28].

To conclude: The cross-correlations of the Gold sequences are better than those of the PN sequences. Also the scalability is better and the sequences can still be used for asynchronous transmission.

4.2.4 Comparison of Sequences

In Table 4.3 a comparison is made among the CDMA sequences discussed. The sequences are compared for the metrics discussed in section 4.1 in both synchronous as well as asynchronous environments. From this table it is clear that the Gold sequences are the best sequences suited for the environment of disaggregating individual lights, due to their asynchronous nature, low cross-correlation, auto-correlation with only one peak and the codes scale linearly.

	Orthogonal	PN	Gold
Synchronous Transmission	✓	✓	✓
Asynchronous Transmission	✗	✓	✓
Math. bounded cross-corr. (synchronous)	✓	✗	✓
Math. bounded cross-corr. (asynchronous)	✗	✗	✓
Scalability ($C \propto L$)	✓	✗	✓

Table 4.3: Table showing a comparison of the discussed CDMA sequences.

4.3 Interference Solution

In the previous sections, interference was discussed when using a CDMA approach. When there are too many transmitters on the same channel, at the same time, the interference can be so great that it destroys a code sequence of one or more transmitters. This is not a problem for orthogonal sequences, since they have no cross correlation, when transmitted synchronously (See subsection 4.2.1). But this is an issue for Gold sequences. As stated in subsection 4.2.3, the cross-correlation of Gold sequences is bounded by $t(n)$, see Equation 4.9. This means that the maximum number of transmitters can be calculated such that not too much interference will occur. With this information two methods can be used to overcome the interference problem.

4.3.1 Continuous Method

One method is to only use the maximum number of transmitters such that no interference will occur. A threshold T must be set for which we will accept or reject a correlation as being a valid result. When a particular code sequence is present in the received signal, the correlation will be equal to the length of the code, L. To prevent false-negatives, meaning there is a code sequence present in the received signal but it is lost due to too much interference from other code sequences, the correlation result needs to be higher than the threshold T. Lets say we have a signal with m codes and we correlate with a code that is also present in this signal. The correlation for the code that is present will be L and the correlation for all the other codes will be bounded by $t(n)$. Assuming the worst case scenario and each correlation with each other code sequence will be the negative of the absolute maximum cross-correlation $t(n)$, the correlation is: $R = L - m \times t(n)$, where m is the number of transmitters. The correlation needs to be higher than the threshold T to prevent false-negatives, so we get the following equation Equation 4.11.

$$R = L - m \times t(n) > T \tag{4.11}$$

Now to also prevent false-positives, meaning for the code sequence that we are correlating with is not present but due to interference the correlation result suggests that it is present, we get Equation 4.12.

$$R = m \times t(n) < T \tag{4.12}$$

Code length (L)	Peak cross-correlation	$\lfloor m \rfloor$
31	9	1
63	17	1
127	17	3
511	33	7
1023	65	7

Table 4.4: Table containing the maximum number of simultaneous transmitters $\lfloor m \rfloor$, such that no destructive interference takes place.

If we equalize Equation 4.11 and Equation 4.12, we can calculate what m and T are. The results can be seen in Equation 4.13 and Equation 4.14, respectively.

$$m = \frac{L}{2 \times t(n)} \tag{4.13}$$

$$T = \frac{L}{2} \tag{4.14}$$

Using the equations above, a table can be compiled showing the maximum number of simultaneous transmitters for which it is guaranteed that there will be no false-positives and/or false-negatives when decoding the incoming signal. The table can be seen in Table 4.4, since m is a real number but the number of transmitters can only be an integer, the 'floor' function is used on m.

Because of this value m, not all transmitters can transmit continuously. But if we had a system with a total amount of m transmitters, we could reverse calculate what code length to use such that all transmitters can transmit continuously. In Equation 4.13, m is written as a function of n. We need n as a function of m, so that L can be expressed as a function of m. The length of the code sequence L as a function of the number of simultaneous transmitters m can be found in Equation 4.15. For simplification reasons, the cross-correlation function $t(n)$ was taken for n odd. The methods used to reverse calculate the equation is beyond the scope of this thesis.

$$L = \left(\sqrt{2 \times m^2 + 2 \times m + 1} + \sqrt{2} \times m\right)^2 - 1 \tag{4.15}$$

What we can conclude from Equation 4.15, is that L is a polynomial

Number of transmitters (m)	Code Length (L)	Time (t)
7	511	0.0511 s
15	2047	0.2047 s
31	8191	0.8191 s
63	32767	3.2767 s
127	131071	13.107 s
255	524287	52.429 s
511	2097151	209.72 s (3.5 min)
1023	8388607	838.86 s (14.0 min)
2047	33554431	3355.4 s (55.9 min)
4095	134217727	13421.7 s (3.7 hour)
8191	536870911	53687.1 s (14.9 hour)
16383	2147483647	214748.4 s (2.5 day)
32767	8589934591	858993.5 s (1.4 week)

Table 4.5: Table containing time it takes to receive each transmission from each transmitter, as a function of the number of transmitters, with a constant modulating frequency $f = 10$ kHz.

function of m. The time t it takes to complete one transmission can be found in Equation 4.16, where L is length of the code, a function of m, the number of simultaneous transmitters, and f is the constant modulating frequency in Hz. This will guaranty good decoding properties, meaning no false-positives and/or false-negatives and all transmitters can transmit the entire time, simultaneous.

$$t = \frac{L}{f} \tag{4.16}$$

Table 4.5 states the time needed to receive a transmission from a transmitter as a function of the total number of transmitters in the systems. All the other transmitters in the system will also be transmitting at the same time.

4.3.2 Probabilistic Method

Another solution to overcome the interference problem, is to use a probabilistic approach. The benefit of this method is that it can utilize all code sequences in the set, instead of a limited amount of them like in the solution above. The drawback is that the time it takes to identify if all the LEDs are on or off becomes a function of the precision that the user requires.

Each transmitter is given k slots, in one and only one of these slots will the transmitter transmit its code. The probability that is associated with this is $p = \frac{1}{k}$. Since the transmitters have a probability p for which it will transmit and with probability $1 - p$ it will not transmit, they follow a Bernoulli distribution. Since there is more than one transmitter which follows a Bernoulli distribution, the number of transmitters which will be transmitting at any point in time, is a Binomial distribution. Now that every transmitter has a probability p, p must be calculated such that no interference will take place and the number of slots k needs to be determined. When those values are calculated the time in order to successfully identify each LED can be calculated.

The probability p must be chosen such that at every point in time the number of transmitters transmitting their sequence will not exceed m, in order to not have interference issues. The cumulative distribution function for a Binomial distribution can be seen in Equation 4.17, where X is the random variable for the number of transmitters at every point in time, m is the maximum number of transmitter to avoid interference issues and N is the total number of transmitters used. The probability that $X \leq m$ needs to be as high as possible to avoid the interference issue, but as the CDF goes asymptotically to 1, we cannot choose the probability 1. Instead a value of $1 - \epsilon$ is chosen, where ϵ is the error, so $1 - \epsilon$ is the precision or accuracy. Equation 4.17 is then equalized to $1 - \epsilon$ to find a probability p, since every other variable is known. With the found probability p, the probability that the number of transmitters at every point in time, $\mathrm{PR}(X \leq m)$, will not exceed m is equal to $1 - \epsilon$.

$$\text{CDF: } \mathrm{PR}(X \leq m) = \sum_{i=0}^{m} \binom{N}{i} \times p^i \times (1-p)^{N-i} \tag{4.17}$$

Now that the probability p can be calculated, k can be calculated. We know that $p = \frac{1}{k}$ so $k = \frac{1}{p}$. After k attempts the transmitter will have transmitted once. This holds for one transmitter, but since each transmitter works in parallel and each random variable is independent and identically distributed, this holds for the entire system with N transmitters. Now the time it takes for all transmitters to have transmitted can be calculated. The result can be seen in Equation 4.18, where L is the length of the sequence, f is the constant modulating frequency, k is the number of slots and p is the probability that a transmitter will transmit. L is a function of N, the number of transmitters in the system. And p is a function of the error ϵ and N.

Number of transmitters (N)	Code Length (L)	Time (t), $\epsilon = 0.001$	Time (t), $\epsilon = 0.1$
9	7	6.31 s	0.06 s
33	31	2.22 s	0.19 s
65	63	8.91 s	0.77 s
129	127	3.79 s	0.93 s
513	511	13.24 s	5.62 s
1025	1023	53.10 s	22.50 s
2049	2047	65.36 s (1.09 min)	37.64 s
8193	8191	387.28 s (6.45 min)	268.38 s (4.47 min)
16385	16383	1549.95 s (25.83 min)	1074.30 s (17.90 min)
32769	32767	2551.95 s (42.53 min)	1989.50 s (33.16 min)

Table 4.6: Table containing time it takes to let each transmitter transmit, as a function of the number of transmitters and error ϵ, with a constant modulating frequency $f = 10$ kHz.

$$t = \frac{L}{f} \times k = \frac{L}{f} \times \frac{1}{p} \tag{4.18}$$

This probabilistic method allows the use of all the sequences in the same set. The drawback is that the time for which all transmitters have transmitted, is dependent on the accuracy $1 - \epsilon$. In Table 4.6 the time it takes to let each transmitter transmit, is listed as a function of the number of transmitters in the total system and the error ϵ. The time in the table is for a constant modulation frequency chosen to be 10 kHz, for other frequencies the time scales linearly, see Equation 4.18.

When ϵ is quite small, $\epsilon = 0.001$, the probability of interference is low, but the times grow larger. For $\epsilon = 0.1$, the times are smaller, but the probability of interference will be higher. A simulation will be performed in order to asses what values of ϵ will give acceptable results. The results of the simulation can be found in section 6.2.

4.4 Mapping Problem

The coding methods as discussed in subsection 4.2.1 and subsection 4.2.2 are used in the field of telecommunication. Since these signals are analog radio waves, the symbols are $+1$ and -1 and they are balanced around 0. The LFS registers used with the generation of PN and gold sequences, only output zeros and ones. For the use with radio-signals a code sequence is mapped to a form with +1 and −1, where the original 0 is mapped to +1 and the original 1 is mapped to −1 [3].

First the equations are shown, how to calculate the correlation for a particular code when the code sequences from all the transmitters only have +1 and −1 symbols. In Equation 4.19 it is shown how to calculate the correlation R, where we use Equation 4.1 and where $s(t)$ is the received signal which is the composed signal of m distinct codes, which all use the +1 and −1 symbols and where $c_1^r(t)$ is the code sequence, with symbols +1 and −1, for which we want to calculate the correlation.

$$
\begin{aligned}
R_{sc_1^r} &= \sum_{t=0}^{L-1} s(t) \times c_1^r(t) \\
s(t) &= \sum_{i=1}^{m} c_i^r(t) \\
R_{sc_1^r} &= \sum_{t=0}^{L-1} \left\{ c_1^r(t) \times \sum_{i=1}^{m} c_i^r(t) \right\}
\end{aligned}
\tag{4.19}
$$

The correlation calculation holds for any sequence with +1 and −1 symbols. For example: Assume $c_1^r = \{-1, 1, -1, -1, -1\}$ and $c_2^r = \{1, -1, -1, 1, -1\}$. Note that code c_1^r is not balanced, the sum is equal to −3, code c_2^r is balanced because the sum is equal to −1. Also note that these sequences are completely chosen at random, the codes are not orthogonal, PN or Gold sequences. These codes are the IDs of transmitters 1 and 2, respectively. Also assume that these codes are transmitted at the same time to create $s = c_1^r + c_2^r = \{0, 0, -2, 0, -2\}$. The calculated correlation using Equation 4.19 will now be: $R_{sc_1^r} = 4$ and $R_{sc_2^r} = 4$. These are the correlation results when codes are used with +1 and −1 symbols.

The result of Equation 4.19 is the sum of all the correlations. If the code that is being correlated with, is present in the signal $s(t)$ then one of the outcomes of the correlation will be the peak auto-correlation L then we can write the total correlation as follows: $R_{sc_1} = L + \sum_{t=0}^{L-1} \left\{ c_1(t) \times \sum_{i=2}^{m} c_i(t) \right\}$, where the result is the auto-correlation L plus all the other correlations. If the orthogonal sequences were used all the other correlations are zero. If the Gold sequences were used, each of the other correlations could be one of the values as stated in Equation 4.10. This is where the multiple access interference shows.

The above correlation calculation only holds, when using a coding sequence which has $+1$ and -1 symbols. However, for the states of the LEDs we can only choose between an on or off state with OOK, so a 1 or a 0. This means a different correlation calculation is required. First a formula for the mapping from $+1$ to zero and -1 to one is needed. The formula can be found in Equation 4.20, where r denotes the $+1$ or -1 symbols and the outcome b will be our binary value, 0 or 1. This mapping formula is based on the mapping used by PN sequences to work with wireless communication A PN sequence outputs zeros and ones and these are mapped to $+1$ and -1 respectively [3].

$$b = \frac{1 - r}{2} \tag{4.20}$$

The codes that were used in the example above, can now be mapped to 0 and 1 symbols. This results in the following codes: $c_1^b = \{1, 0, 1, 1, 1\}$ and $c_2^b = \{0, 1, 1, 0, 1\}$. The b notation indicates that these sequences consists of 0 and 1 symbols.

The previous equation (Equation 4.19) can now be altered to incorporate the fact that the LEDs work with a one and zero state. In Equation 4.21 it is shown how the correlation formula changes when the codes that are used for the LEDs now have 0 and 1 values instead of the $+1$ and -1 values. In Equation 4.21 the new correlation $\hat{R}$ is calculated, where $c_i^r(t)$ is the code sequence i and consists of symbols -1 and $+1$, and $c_j^b(t)$ is the code sequence j and consists of symbols 1 and 0 and where $s(t)$ is the received signal which is the composed signal of m distinct codes. To clarify: The codes that the LEDs use have values 0 and 1 due to the OOK, the code that is being correlated with at the receiver or current-sampler side still uses the $+1$ and -1 symbols.

The altered correlation calculation, denoted by $\hat{R}$ shows different results than the previous result R. Where $R_{sc_1} = \sum_{t=0}^{L-1} \left\{ c_1^r(t) \times \sum_{i=1}^{m} c_i^r(t) \right\}$ and $\hat{R}_{sc_1^r} = \frac{m}{2} \times \sum_{t=0}^{L-1} c_1^r(t) - \frac{1}{2} \times \sum_{t=0}^{L-1} \left\{ c_1^r(t) \times \sum_{i=1}^{m} c_i^r(t) \right\}$. We can write it such that R will become a function of $\hat{R}$ as can be seen in Equation 4.22. With this equation we get the same output domain as with the normal correlation formula used for the codes that have $+1$ and -1 symbols.

$$
\begin{aligned}
\hat{R}_{sc_1^r} &= \sum_{t=0}^{L-1} s(t) \times c_1^r(t) \\
s(t) &= \sum_{i=1}^{m} c_i^b(t) = \sum_{i=1}^{m} \frac{1 - c_i^r(t)}{2} \\
\hat{R}_{sc_1^r} &= \sum_{t=0}^{L-1} \left\{ c_1^r(t) \times \sum_{i=1}^{m} \frac{1 - c_i^r(t)}{2} \right\} \\
\hat{R}_{sc_1^r} &= \tfrac{m}{2} \times \sum_{t=0}^{L-1} c_1^r(t) - \frac{1}{2} \times \sum_{t=0}^{L-1} \left\{ c_1^r(t) \times \sum_{i=1}^{m} c_i^r(t) \right\}
\end{aligned}
\tag{4.21}
$$

$$
R_{sc_1^r} = m \times \sum_{t=0}^{L-1} c_1^r(t) - 2 \times \hat{R}_{sc_1^r} \tag{4.22}
$$

When the transmitters or LEDs in this case, transmit the codes c_1^b and c_2^b, the following signal s would be created: $s = c_1^b + c_2^b = \{1, 1, 2, 1, 2\}$. Equation 4.21 can now be used to calculate the new correlation $\hat{R}$: $\hat{R}_{sc_1^r} = -5$ and $\hat{R}_{sc_2^r} = -3$. These results are not the same as with the previous correlation results, where the results were 4 and 4, respectively. This is because R and $\hat{R}$ cannot be compared, instead the results obtained from $\hat{R}$ need to be mapped to R. The values obtained using $\hat{R}$ can now be mapped to R by using Equation 4.22. For this equation we also need the sum of the code sequences, which were already calculated in the beginning of the example to show that one of the code sequences is not balanced. The sums are: $\sum c_1^r = -3$ and $\sum c_2^r = -1$. To obtain R: $R_{sc_1^r} = m \times \sum c_1^r - 2 \times \hat{R}_{sc_1^r} = 2 \times -3 - 2 \times -5 = 4$ and $R_{sc_2^r} = m \times \sum c_2^r - 2 \times \hat{R}_{sc_2^r} = 2 \times -1 - 2 \times -3 = 4$. And these correlation results are the same as the correlation results as with the codes which had -1 and $+1$ symbols. Hence, we show that any sequence can be mapped from -1 and $+1$ symbols to 0 and 1 symbols and still get correct correlation results.

In Equation 4.22 the sum of the sequence used to correlate with, is needed. The sum of the sequence depends on the sequence used, when using orthogonal codes the sum will be zero, see subsection 4.2.1. If the sum is equal to zero, the factor m will not matter. However when the sequence used is a PN sequence or a balanced Gold sequence, the sum will be -1. The sum will be -1 due to the fact of the balance property of the PN sequences, as explained in subsection 4.2.2. The sequence will have exactly 2^{n-1} ones, where the ones will be mapped to -1 and the sequence will have exactly $2^{n-1} - 1$ zeros, which will be mapped to 1. These mappings can be found when inversing Equation 4.20, this yields: $r = 1 - 2 \times b$. When we calculate the sum of the sequence we get: $2^{n-1} \times -1 + (2^{n-1} - 1) \times 1 = -1$. Because the sum is equal to -1: $R = -m - 2 \times \hat{R}$, and then m, the number of signals, is important. When using an unbalanced Gold sequence, the sum of that sequence is not equal to -1. All the chips of the unbalanced Gold sequence can be added, in order to find the sum of this particular sequence and use the outcome to calculate the correct correlation value as we have done for the example with code c_1^r.

Chapter 5

Hardware Design

The codes that will be assigned to each LED have now been introduced. Now, hardware is needed to put the theory of those codes into practice. Since hardware will never behave exactly like the software simulations, this chapter will discuss the existing hardware that is out there. Then, the shortcomings of the hardware are highlighted and custom hardware is introduced. First the DC case is considered and then the AC case.

5.1 DC Environment

In a DC environment, the code framework works well because the DC power signal is an easy signal to understand and work with. For example, in Figure 5.1 the voltage output of a DC power supply can be seen. A load is switched on and off, and the current that flows over time can also be seen in the figure. This switching causes the current to be a square wave with sharp edges. The current is zero when the load is off and the current is a constant value for when the load is on.

This is the ideal way the modulator for an LED must work. In the next sections, the hardware for the modulator will be explained as well as how the current will be measured to extract the encoded information and finally a testbed will be presented.

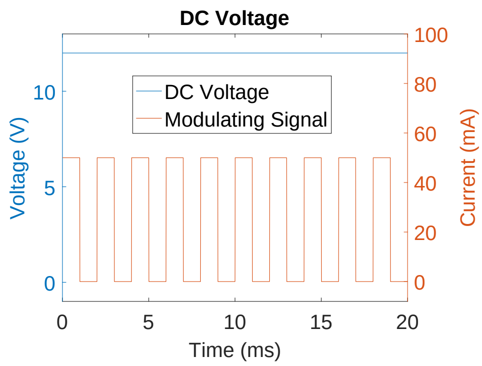


Figure 5.1: Voltage of a DC power supply, along with the current of a switching load.

5.1.1 Modulator

The task of the modulator is to switch the LED on or off based on the ID that is assigned to the light fixture. It is the responsibility of this piece of hardware to translate the ID into the unique current signature.

The way an LED (Light Emitting Diode) works, is that current has to flow through it in order for it to emit light. In other words, an LED is controlled via current, it is not controlled by applying a voltage to it. When a certain amount of current flows through the LED, a certain amount of voltage will drop over the LED.

The easiest way to make an LED emit light, is to put a current limiting resistor in series with a voltage source and an LED. A schematic can be seen in Figure 5.2. But there exists no ideal DC power supply. Depending on the load, the provided voltage of the power supply may fluctuate. Due to the fluctuation of the provided voltage, the LED can start to change in brightness. The current that flows through the LED is a function of the voltage over the resistor and the value of the resistor: $I = \frac{U}{R}$. So if U, the provided voltage, is fluctuating, the current that flows through the LED, I, will also fluctuate. This causes the brightness of the LED to fluctuate.

A better way to power an LED, is by using a current source. Where an ideal voltage source will always deliver a certain amount of voltage independent of the load, a current source will always deliver a certain amount of current independent of the load. If a constant amount of current flows through the LED, the brightness will not fluctuate. In Figure 5.3 a schematic can be seen, which shows an example of a current source powering an LED. This current source can be toggled on and off via a 0 V or 3.3/5 V

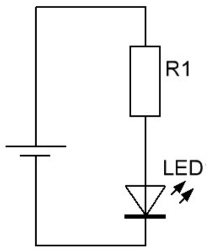


Figure 5.2: Simplest way to power an LED.

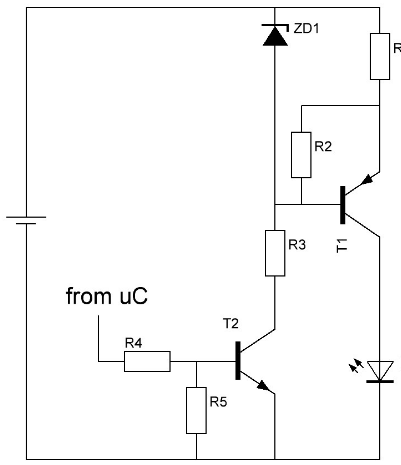


Figure 5.3: Current source powering an LED.

signal, coming from a micro-controller, uC in the schematic.

By using a current source, two benefits can be identified:

- Since the current that flows through the LED is constant, the LED will not flicker.
- The current source will make sure that the current that is drawn, is either zero or some constant value depending on the bit that is being encoded. This will yield a signal with two distinct values. When multiple of these signals are aggregated and measured by a smart-meter, the measured signal will also have distinct values. This signal will make it easy to decode the information that was encoded by the modulators.

5.1.2 Current Sampler

Now that the hardware is created to translate the IDs of the LEDs into a unique current signature, we also need a way to measure the current. The measured current can then be processed by a micro-controller. And in turn it can be identified which LEDs are on and which are off.

In the interest of time the most simple manner was chosen to measure the current for the DC hardware. Other options are available for measuring current and they will be discussed in the AC part. The most simple way to measure current is by using a series resistor. The resistance does not variate and therefore no noise is introduced in the sampled signal. The resistor is placed in series between the DC power supply and the LEDs. The voltage drop over the resistor is linearly proportional to the current that flows through the resistor, according to Ohm's Law $U = R \times I$. If the value of the resistor is chosen such that the maximum voltage will never exceed the rated voltage for a micro-controller, it can be directly measured by the micro-controller's ADC in question.

5.1.3 Testbed

To be able to test all aspects of the system a testbed was created. This testbed will allow the testing of the correlation calculations as discussed in section 4.4, the interference solutions (subsection 4.3.1), and the performance of the modulator and current-sampler.

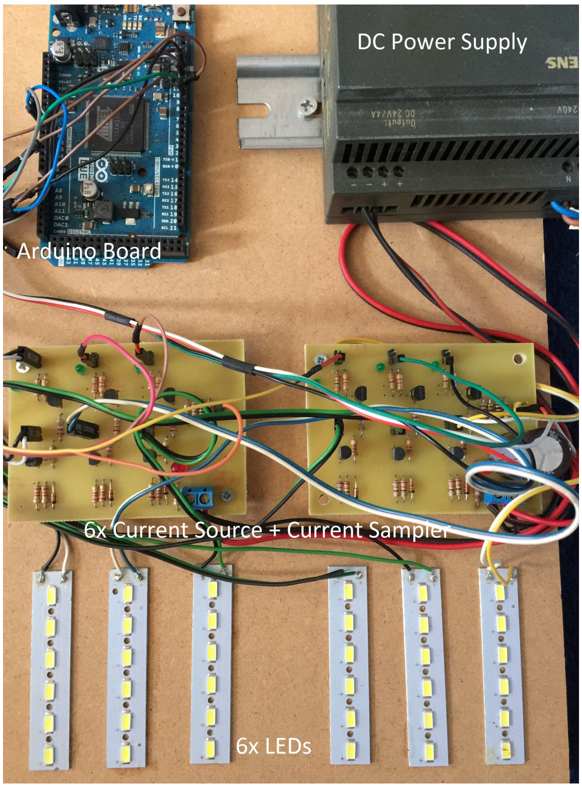


Figure 5.4: Picture of the DC testbed, showing the six LED strips, current sources, current sampler and an Arduino board.

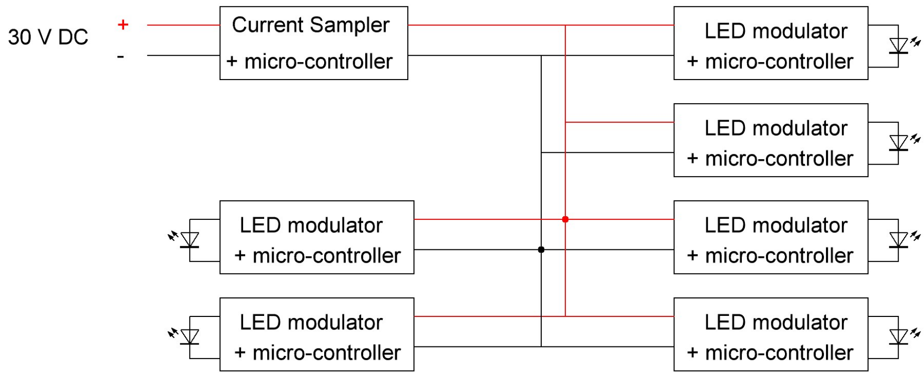


Figure 5.5: Architectural overview of the DC testbed. Six LED modulators are connected in parallel with each other and in series with the current sampler.

The testbed works with a DC power supply and has six individual controllable current sources. Each current source powers one LED fixture. The testbed itself can be seen in Figure 5.4 and an architectural overview of how everything is connected to each other can be seen in Figure 5.5.

The aim of the testbed was to use commercial LEDs. The LED fixtures used in this testbed all came from the same commercial LED, which can be found in [1]. A picture of the LED can be seen in Figure 5.9. The strips are taken out of this commercial LED and used individually for this testbed.

The current is measured by a series resistor and fed to the ADC of a micro-controller. The entire schematic of the DC testbed can be found in Appendix A.

The current sources and therefore the LEDs, can be toggled on and off by a micro-controller. By toggling the current sources on and off, the current that flows through the series resistor will change accordingly. A change of voltage over the series resistor can then be measured by the ADC.

The measured current from the DC testbed can be seen in Figure 5.6. In this experiment, all six LEDs are modulating with the unique ID that was assigned to each LED. The voltage drop over the resistor is measured with the ADC. The raw ADC value can be seen on the left y-axis and the calculated aggregated current that is drawn can be seen on the right y-axis. From this figure seven distinguishable states can be seen. When there are no LEDs on, all LEDs are encoding a ‘0’ data bit, the current is zero. When one of the six LEDs is transmitting a ‘1’ data bit, the current jumps to

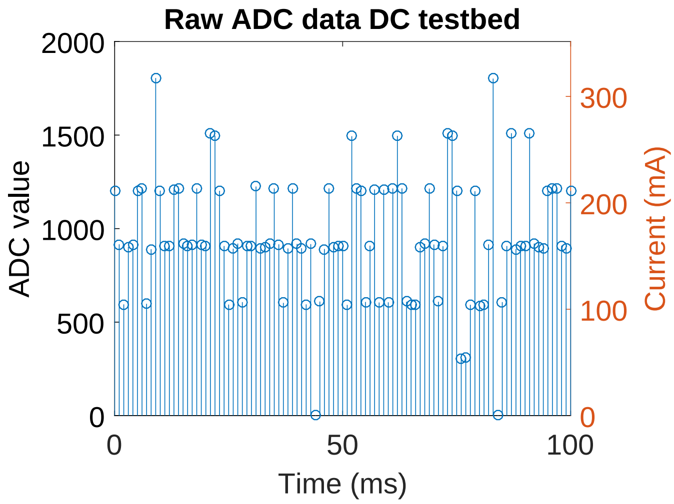


Figure 5.6: Raw current data from the DC testbed with the six LEDs modulating with their unique IDs.

roughly 50 mA. When two of the six LEDs are encoding a '1', the current is roughly 100 mA and so on.

An evaluation of this testbed is done in subsection 6.1.1 where a longer signal will be shown along with correlation data to identify if a certain LED is on, while the other LEDs are modulating and thereby causing interference.

5.2 AC Environment

The translation from an ID, that is assigned to a light fixture, to a unique current signature is more difficult in an AC environment compared to the DC environment. This is due to the supplied voltage. In a DC environment, the voltage is a constant value, but in an AC environment the supplied voltage is a sinusoidal wave. This means that the voltage will be both positive and negative. In the Netherlands the AC has an RMS voltage of 230 V and a frequency of 50 Hz.

In the next subsections, we will first describe existing hardware and their limitations. Then new hardware is proposed to solve the limitations of the existing hardware. Finally, the AC testbed is showcased.

5.2.1 Modulator

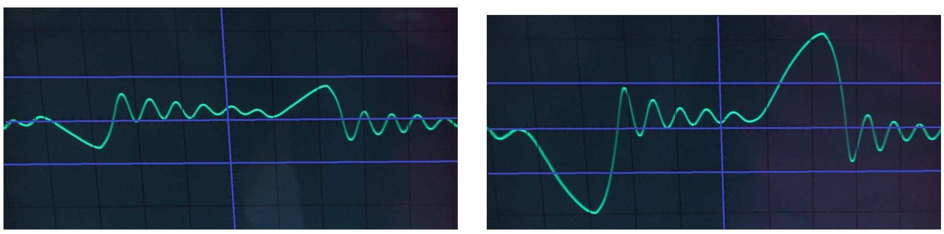
The job of the modulator in an AC environment, is the same as in a DC environment. Just like in the DC environment, when a ‘0’ has to be encoded the current should be zero and when a ‘1’ has to be encoded the current should be some constant value. The transition between a ‘0’ and ‘1’ should be fast, so that a square wave is created, just like in the DC case.

If the translation is done in this manner, the mapping between the code sequence symbols and the current is clear and the aggregated current drawn by multiple lights will also be a square wave, which will allow for decoding the information. Next, existing hardware is investigated to see if they can modulate the current draw as ideal square waves.

SMPS and LED

In Figure 5.8 a commercial LED lighting fixture can be seen with a switching mode power supply (SMPS). The power supply transforms the AC to a DC signal which is used to power the LED.

For this setup we investigate what the current signature looks like when the LED is on, representing a ‘1’, and when the LED is off, representing a ‘0’. A 10 ohm resistor is placed in series with the SMPS on the AC side and the voltage drop over this resistor is measured. So a voltage is measured, but the current that flows can be calculated using Ohm’s law: $I = \frac{U}{R}$. Since we are investigating how the current signature behaves, it is not needed to have exact numbers on the current draw.



(a) Encoding a ‘0’, LED off. (b) Encoding a ‘1’, LED on.

Figure 5.7: Voltage measured over a 10 ohm series resistor at the primary side, to determine the current flow in two situations of a SMPS, when encoding a ‘0’ and a ‘1’. X-axis: 2 ms/div, Y-axis: 500 mV/div.

Figure 5.8: Picture of a commercial LED light fixture with a DC SMPS.

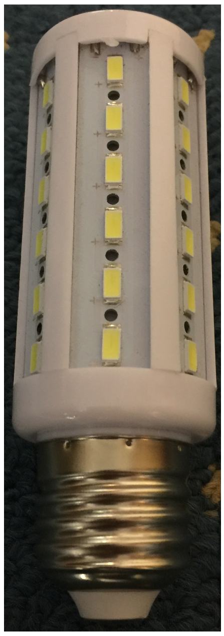


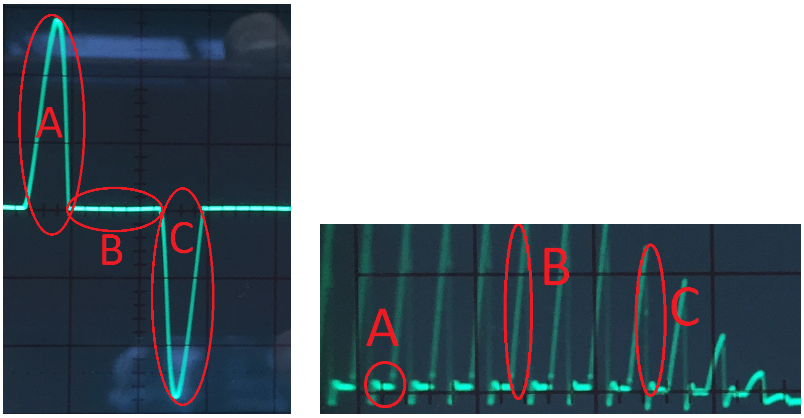
Figure 5.9: Picture of a commercial 230 V AC LED light fixture.

In Figures 5.7(a) and 5.7(b) the current signature of the SMPS can be seen, when encoding a '0' (LED off) and when encoding a '1' (LED on), respectively. From these figures, it is clear to see that when encoding a zero, the current is not zero and when encoding a '1', the current is not a constant value. This makes it difficult to determine what the ID of the transmitting LED is. This becomes even more difficult when multiple of these current signatures are aggregated.

LEDs powered directly by 230 V AC

There also exists LED lighting fixtures which work directly with AC, without having an external SMPS. An example of such an LED can be seen in Figure 5.9 [1].

For this LED it is investigated what the current signature looks like when the LED is constantly on, and when the LED is encoding information. To measure the current, a 22 ohm resistor is placed in series with the light and the voltage drop over this resistor is measured. Again, we are only looking at how the current signature behaves, at this point we are not interested in how



(a) LED is constantly on. X-axis: 5 ms/div, Y-axis: 1 V/div.

(b) LED is being modulated at 4 kHz. X-axis: 1 ms/div, Y-axis: 5 V/div.

Figure 5.10: Voltage measured over a 22 ohm series resistor at the primary side, to determine the current signature of a commercial 230 V AC LED.

much current is exactly being drawn. In order to modulate the LED so that information is encoded in the current, changes had to be made to the internal hardware of the light fixture. In Appendix B a schematic can be found, which shows the original circuit and what was modified in order to modulate the LED. The modifications made, include two transistors to switch the LEDs off and stop the capacitor from charging. With these transistors the entire current flow can be stopped thereby drawing zero current when we want to modulate a '0' symbol. When a '1' symbol needs to be modulated, the transistors are turned on, and the LED behaves in a normal manner. The transistors are controlled via a micro-controller through an optocoupler to protect the micro-controller in the development stages.

When the LED is off, there is no current draw. In Figure 5.10(a) the current signature can be seen when the LED is constantly on. This figure shows 20 ms of the current signal, this is exactly one period of the supplied AC voltage: $t = \frac{1}{f} = \frac{1}{50} = 0.020$ s $= 20$ ms. Every 20 ms this current draw repeats itself. In the figure, three regions can be seen: A, B and C. Each region will now be explained:

- In region A, the supplied AC voltage is positive, and so the current that flows is positive. As can be seen from the schematic in Appendix B, all the LEDs are connected in series. This means that a certain voltage is required in order for the LEDs to draw current. The required voltage is available at the start of region A and is no longer available at the end of region A and the current flow stops. The width of the peak is

4 ms. This is because in those 4 ms, there is enough voltage available to power all the LEDs in series. The height of the peak indicates how much current is drawn.

- The start of region B starts when region A stops. The required voltage is no longer available and so the current flow stops and the LED no longer emits light. During this region there is no current flow. This is a good property: If the LED is off there is no current flow, exactly what we require.
- Region C starts when region B stops. At this point in time, there is enough voltage available for the LEDs to start drawing current again. But the voltage that is now available, is not positive as it was in region A, but is negative. That is why this peak goes the opposite direction of the peak seen in region A. The current that flows is in that case negative, because the voltage is also negative. In other aspects this peak has the same properties as those of the peak in region A, it has the same width and height.

Only in region A and C, the LED can be used to encode information. Because in region B, the is no current draw and therefore we cannot modulate information. The peaks in region A and C are both 4 ms wide. This means that there is 8 ms of time available for modulation in the 20 ms period, so $\frac{8}{20} = 40$ % of the time is available for encoding the ID with this LED. To compare with a DC environment: In a DC scenario, the power supply always supplies a constant voltage, so the LED can always emit light and therefore can always draw current, so 100 % of the time is available for modulation. If the ID would have a certain length, it would take time t to modulate this ID in a DC environment and time $2.5 \times t$ with this AC LED if the ID could not be transmitted inside the 4 ms window.

Now that the current signature has been investigated when the LED is on, the LED will be now be turned on and off with a frequency of 4 kHz to investigate how the current signature behaves. In Figure 5.10(b) the current signature can be seen when the LED is being modulated with a frequency of 4 kHz. The entire figure takes place inside region A of Figure 5.10(a). If this figure would have been showed for region C of Figure 5.10(a), the amplitudes would have been negative. Again, there are three regions highlighted:

- In region A, the data being encoded is a '0' and the current draw is also zero. As discussed this is a desired property.
- Region B shows the transition from the LED being off to the LED being on. The current does not go straight up, but instead a slope can

be seen. This is not a desired property. A square wave is desired, this LED is producing a triangular wave.

- In region C, the transition from the LED being on to the LED being off can be seen. This time the current does go immediately to zero, which is a good property. Around region C another issue can be seen. The amplitude of the current becomes lower and lower until region A of Figure 5.10(a) ends. This is also not a desirable property, the current should always be some constant value and not decreasing over time.

From this investigation it can be concluded that this 230 V AC LED is not suitable to map the ID into the current signature which is desired. And also the time that would be available for modulation is limited, because there are many LEDs in series which require a high voltage which is only available for a short amount of time.

Custom AC Modulator

Given the shortcomings of the existing hardware, we decided to build custom hardware that would behave exactly how it was needed in order to successfully encode information in such a way that the currents could be decoded by a smart-meter. This means that the current signature should behave very similar to the DC case: For a '0' data bit encoding, the current should be zero and for a '1' data bit encoding, the current should be some constant value. The transitions between the symbols must be such, that a square wave is created, just like in the DC case.

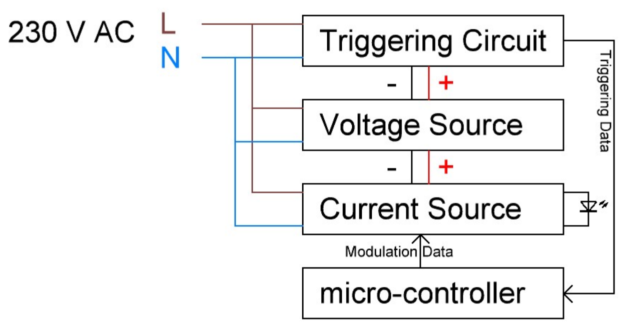


Figure 5.11: Architectural overview of the AC LED modulator.

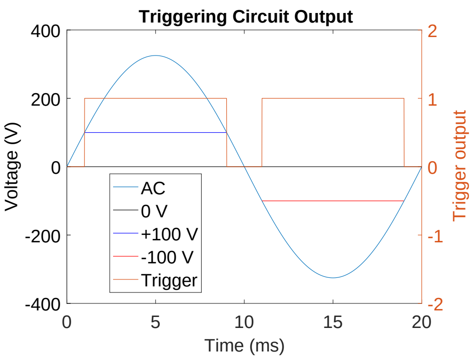


Figure 5.12: Output form the triggering circuit alongside the incoming AC voltage.

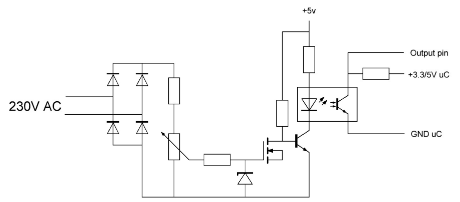


Figure 5.13: Triggering circuit to determine when the voltage is sufficiently high enough to start encoding the ID.

The custom modulator consists of three separate parts: A trigger circuit, a current source and a voltage source. In Figure 5.11, an overview of how these parts are connected to each other can be seen. The commercial LED that will be used for this solution was already shown Figure 5.8, but the SMPS is not used, instead this custom hardware will be powering the LED. The entire modulator's schematic can be found in Appendix C. In the following subsections each part of the hardware of the custom LED modulator will be explained.

Triggering As discussed before, LEDs need a certain amount of voltage before they draw current. That is why Figure 5.10(a) has two separate regions where it draws current, instead of drawing current continuously. Given the fact that the ID of the LED cannot be transmitted the entire time, due to the voltage issue, we must somehow give the micro-controller a signal for when the modulation can start and stop. If the micro-controller does not have this information, parts of the ID would be encoded when there is no current draw and thus information would be lost, since the encoding of the ID is done with the current draw itself.

To let the micro-controller know when to start and stop modulating, a triggering circuit is designed. This circuit tells the micro-controller when more than a preset voltage is made available by the AC power. It also tells the micro-controller when there is less than the preset voltage available. In Figure 5.12 the AC voltage can be seen, along with the preset voltage and the corresponding logical output of the triggering circuit.

In Figure 5.12, the preset voltage is set at 100 V. When more than 100

V is available the LED will emit light and start drawing current. At that point, around 1 ms in the figure, the triggering circuit will output a logical '1' to the micro-controller, meaning that the encoding of the ID may start. When less than 100 V is available, around 9 ms in the figure, the logical output of the circuit becomes '0', indicating to the micro-controller that it is time to stop encoding the ID, else information is lost since the LED is not turning on anymore and therefore no longer draws current. The same happens for the negative part of the sine wave, except the preset is now −100 V.

From Figure 5.12 it can be deduced that two times 8 ms is available to encode the ID in a period of 20 ms. So $\frac{2\times 8}{20} = 80$ % of the time is available for modulation, which is two times more than the 230 V AC LED (Figure 5.10(a)) provided. To transmit the ID, this solution would be two times faster, but it would still take 25 % more time than in a DC environment.

In Figure 5.13, the circuit can be seen which signals the micro-controller when the voltage is more or less than the preset voltage. The voltage preset can be set by using a potentiometer. The output of the circuit is electrically isolated from the micro-controller by the means of an optocoupler. This is done to protect the micro-controller from the high voltage AC in the development stages.

Current Source To solve the issue of the non-constant current draw of the other LED solutions, a current source will be used to power the LED. The principle is the same as in the DC environment, where also a current source is used. In Figure 5.14 the schematic for the current source implementation can be seen. An optocoupler is used to electrically isolate and protect the micro-controller from the current source, during the development stages.

To investigate exactly how the current source implementation performs, a 2.8 Ohm resistor is placed in series with the AC and the current source implementation. A micro-controller will toggle the current source on and off via the optocoupler with a frequency of 500 Hz. This relatively low frequency is chosen to show the distinct on and off states of the current. If a higher frequency was chosen, the on and off state are harder to see on the figure. The voltage drop over the series resistor is measured to determine the current draw. The result of the current draw can be seen in Figure 5.15.

In Figure 5.15 six areas are highlighted. The regions highlighted with a '1' suffix occur when the supplied AC voltage is positive and the regions with a '2' suffix occur when the AC voltage is negative. What happens in

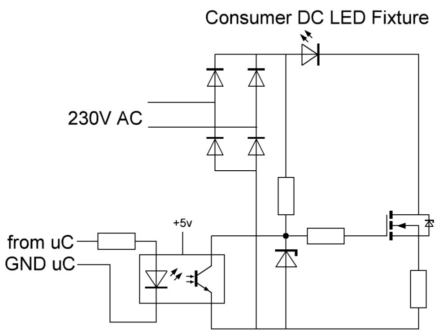


Figure 5.14: Current source to power the commercial LED fixture, can be toggled on and off with a microprocessor.

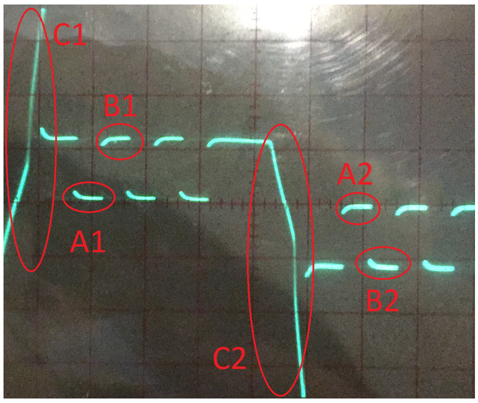


Figure 5.15: Current that is drawn by the current source. Measured over an 2.8 Ohm resistor. Settings: 200 mV/div, 2 ms/div.

these regions will now be explained:

- In region A1 and A2, the LED is turned off because the micro-controller is encoding a '0' and we can see that the current draw is also zero. The current draw is zero, independent if the applied AC voltage is positive, in A1, or negative, in A2.
- In region B1 and B2, the LED is on, the micro-controller is encoding a '1'. In both regions the current draw remains constant over time, until the LED is turned off. But in B1, the current is a positive constant and in B2 the current is a negative constant. This is due to the applied AC voltage.
- What happens in regions C1 and C2 is caused by a voltage source, which powers parts of the triggering circuit and the current source. What exactly is happening will be explained in the next section, but for now we can assume that it will not affect the encoding of the ID.

The combination of the trigger circuit and this implementation of a current source allows this solution to detect when to start and stop modulating the LED and it is able to define a specific current draw for the LED. This makes mapping the bits '0' and '1' to current levels zero and some constant, easy. Which in turn makes detecting the IDs of the LEDs in the smart-meter easy.

However, this solution also has a drawback. The current source will make sure a constant amount of current will flow through the LEDs. The current that flows through the LEDs will cause a voltage difference over the LEDs. This is the voltage the LEDs need in order for current to flow, as discussed in the previous sections. In the triggering circuit section, it was explained that the LEDs used, require 100 V before the current will flow, see Figure 5.12. This voltage is all the voltage that is needed for these LEDs. If less voltage is provided, the LEDs will not emit light. But more voltage than necessary will result in power dissipation in some component and will turn into heat. Since the applied AC voltage is rated at 230 V RMS, the excessive voltage has to go somewhere. This voltage will be dissipated over the current source. This means that some energy is being wasted, because there is more voltage provided by the AC than is needed to power the LEDs.

The other LED solutions discussed above (SMPS LED and 230 V AC LED), solve this problem is two separate manners. The SMPS LED, uses a power supply to transform the supplied AC voltage to a certain DC voltage which is exactly what the LEDs need. The SMPS has a high efficiency so no power is being wasted, but at the same time this power supply disturbs the encoding of the ID in such a way that the ID is not recognizable anymore. The 230 V AC LED uses many LEDs in series and so it requires a very high voltage before the LEDs turn on. So only a limited amount of time is used to turn the LEDs on and therefore only a small amount of time is available for modulation of the ID. And in the process of encoding the ID, the current is not constant and it does not produce a square wave which makes recognizing the ID difficult.

The current source implementation, as discussed in this section, solves the encoding problems which the other solutions had. But this solution introduces an efficiency problem. The priority of this thesis was to create a solution which could recognize which LEDs are on and off by only looking at the aggregated energy consumption. The efficiency of this solution has a lower priority. However, attempts have been made to try and solve the efficiency problem. The first attempt involves the use of a transformer and the second attempt uses a capacitor.

The first attempt uses a transformer to transform the 230 V AC to a lower AC voltage which will solve the efficiency problem. The outputted sine wave of the transformer can be seen in Figure 5.16. The LEDs still need the same amount of voltage: 100V. This 100 V threshold can also be seen in Figure 5.16. The time that is now available for modulation is limited by the use of the transformer, and is 6 ms, which means that only $\frac{6}{20} = 30$ % of the time is available for modulation. Another drawback of using a transformer, is that the distinct current signature that is made using the ID, is distorted by the transformer, as no ideal transformer exists. For this

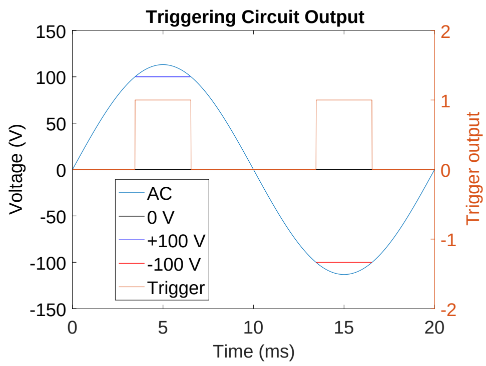


Figure 5.16: Output triggering circuit alongside the output of the AC voltage after a transformer.

reason this solution was abandoned.

The second attempt showed promising results, however due to time constraints it was not possible to fully investigate this solution. The solution is made up out of two parts: A capacitor and a zero-crossing optotriac. The capacitor is placed in series with incoming AC to the rectifying bridge. And the optotriac is parallel to this capacitor. The schematic can be seen in Figure 5.17.

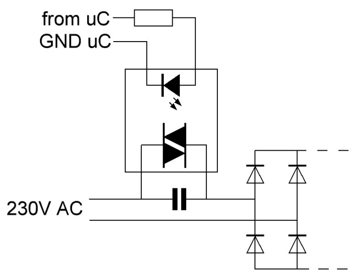


Figure 5.17: Schematic to show how the capacitor and optotriac are connected.

The series capacitor will limit the power that can be dissipated by the current source. The capacitor is able to do this by phase shifting the current as seen from the voltage. The voltage drop over the capacitor can be calculated using Ohm's law $U = R \times I$ and the resistance of a capacitor which is called the reactance, X_c measured in ohms. The reactance of a capacitor is dependent on the frequency of the voltage f and the capacitance C: $X_c = \frac{1}{\omega \times C} = \frac{1}{2 \times \pi \times f \times C}$. The frequency of the voltage is 50 Hz, because the AC power is rated at 50 Hz. Since a current source is used to power the LED, the current will be a constant value when the LED is on. The voltage drop can be calculated by assuming a current of 0.1 A and a standard capacitance value of 3.3 μF. The voltage drop will then be: $U = R \times I = X_c \times I = \frac{1}{2\times\pi\times f\times C} \times I = \frac{1}{2\times\pi\times 50\times 3.3\times 10^{-6}} \times 0.1 = 96.5$ V. This means that the voltage applied to the modulator circuit will now be lowered by approximately 100 V. Even though there is a voltage drop over the capacitor, the capacitor itself will not dissipate any power. This is because the capacitor has no resistance, the real part of the impedance is zero, but it only has reactance which is the imaginary part of the impedance. Due to the voltage drop the efficiency problem can be solved. However by limiting the voltage that the current source can use, there is much less time available to modulate, similar to the solution with the transformer as described above, see Figure 5.16. This is were the zero-crossing triac comes in. If the probabilistic approach is used, the LED is not constantly transmitting its code, instead most of the time it is not modulating. Whenever it is decided to start modulating by the probabilistic approach, the micro-controller activates the triac and then it will short-circuit the capacitor. When this happens, effectively the capacitor is no longer connected and it no longer limits the voltage. And so the time available for modulation is restored to the full 80 %. When the transmission of the ID of the LED is finished, the triac is switched off and the capacitor limits the power once again. During the transmission of the ID, the current source is dissipating power, but this is only during the transmission of the ID, which is the amount of time to encode the ID. Whenever the LED is not transmitting its ID, the capacitor is limiting the voltage and thus the efficiency issue is solved. But due to time constraints it was not possible to fully investigate this solution with the testbed.

Non-disturbing Voltage Source The two parts of the custom modulator which solve the translation problem of mapping zeros and ones of the ID into the current draw, have now been explained in the prior sections: The triggering circuit and the current source implementation. However, as can be seen from the schematics of these parts (Figure 5.12 and Figure 5.14), a 5 Volt voltage source is required to power parts of these circuits. In this section it will be explained where this voltage comes from and how it is able to provide this voltage while not distorting the distinct current signature of the ID.

In the triggering circuit section, it is explained that the LEDs need 100 V before they draw current. So every time the voltage is above 100 V the encoding of the ID may begin and whenever the voltage drops below 100 V the encoding stops. So whenever the voltage is below 100 V, it does not matter what current is drawn, because no encoding of the ID will take place

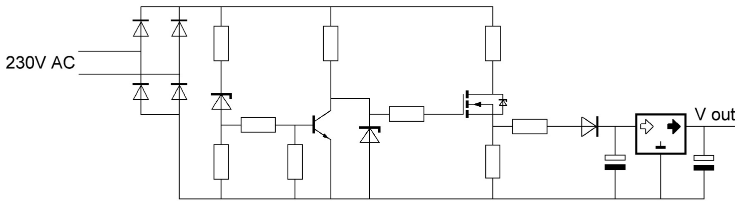


Figure 5.18: Non-disturbing voltage source to power other parts of the circuit.

there. Below 100 V is an ideal place to let a circuit create a stable 5 V output to power the triggering circuit and the current source, because it will not distort the encoding of the ID.

As mentioned before the time that is available for modulation, is 16 ms per period of 20 ms. During 16 ms, the voltage is above 100 V and $20 - 16 = 4$ ms is the time that the voltage is below 100 V. So in 4 ms per period of 20 ms, it is allowed to draw any current necessary to store the required energy to power the trigger and current source circuits. And in the remaining 16 ms per period of 20 ms, this stored energy will be consumed by the triggering circuit and the current source. During this 16 ms no current will be drawn to create the stable 5 V output.

To store the energy that will be used later on, a capacitor is used. The output of this capacitor goes to a voltage stabilizer IC that will ensure a stable 5 V output. A schematic of the solution can be found in Figure 5.18. When the capacitor is empty, the capacitor is charged by the AC only when the voltage is below 100 V. The charging of the capacitor causes a current spike as could already be seen in Figure 5.15. In this figure, regions C1 and C2 clearly show this current spike. But since no encoding is done in this region, it will not interfere with the encoding. This solution provides a continuous stable 5 V voltage source without distorting the encoding of the ID, since the encoding and the voltage source work in different time sections.

5.2.2 Current Sampler

Now that there exists a solution to encode the ID of an LED into the current draw, as explained in subsection 5.2.1, it is time to explore the possibilities to measure the current. The measured current can then be processed by a micro-controller, and then the status of the LEDs can be determined.

To measure the AC current, first we investigated existing solutions. One such a product is a 'Hall Effect-Based Linear Current Sensor' [15]. Based on the Hall effect, this sensor outputs a voltage which is proportional to the current that goes through the sensor. This sensor has the issue that the noise is larger than the voltage signal outputted using by LEDs. According to [15], the highest sensitivity is 185 mV / A and the noise is rated at 21 mV. The commercial LED fixtures which were provided, are rated at 15 Watts. With the 230 V AC, this works out to a current of $I = \frac{P}{U} = \frac{15}{230} = 0.065$ A. With that current the output voltage will be $185 \times 0.065 = 12$ mV. The noise of the output is 21 mV, which is almost double the output voltage when one LED is on. This sensor would not be able to reliably detect one or two LEDs.

Another solution had to be pursued to overcome the noise problem of the Hall effect sensor. This solution has two parts: The current sampler itself and a triggering circuit.

The triggering circuit is needed to know when the modulators start and stop encoding the ID. This will help the smart-meter, by telling it when to start and stop looking for the IDs of the LEDs. The triggering circuit is the same circuit that was used for the modulator, which was explained in subsection 5.2.1. The part which samples the current will be explained next, but first an overview is given of how the different parts of the smart-meter are connected in Figure 5.19. The meter is placed in series with an incoming AC power-line and the LEDs will be connected with the AC output. The current-sampler forwards information about the current draw to a micro-controller for processing. And the triggering circuit detects when the voltage is high enough to demodulate the signal sent by the LEDs.

For the current-sampler itself a resistor will be used, since this approach ensures that no noise will be introduced to the sampled current signal. As could be seen from Figure 5.14, the current that flows is both positive and negative, due to the nature of the AC that is used. This means that the voltage that is measured over the resistor will also be positive and negative. An ADC (Analog-to-Digital Converter) cannot measure a negative voltage. So a constant offset voltage is summed with the voltage over the resistor to ensure that the ADC will always measure a positive voltage. This offset voltage comes from a reference voltage IC [16], which outputs a very stable reference voltage often used in scenarios where an ADC is used to measure analog signals.

At this point the ADC is measuring the voltage over a resistor which is in series with the ground (N) or phase (L) of the 230 V AC. For safety reasons the ADC must now be electrically isolated from the micro-controller,

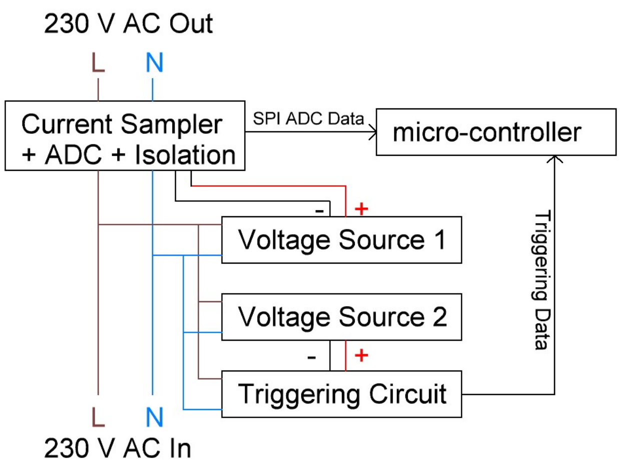


Figure 5.19: Architectural overview of the AC current sampler.

such that no harm can come from even touching the micro-controller. To this end, an SPI isolating chip was used [17] to electrically isolate the SPI signals between the ADC and the micro-controller.

This solution to sample current with an external ADC can sample the current with a high frequency (> 10 kHz) with no noise introduced in the signal. The full schematic of the current sampler can be found in Appendix D.

5.2.3 Testbed

An overview of the AC testbed can be found in Figure 5.21 and a picture of the actual testbed itself can be seen in Figure 5.20. In the testbed three commercial LEDs are used which are powered by the custom modulators as explained in subsection 5.2.1. Arduino boards are controlling the modulators. The modulators are connected in parallel. And the custom current-sampler is connected in series to measure the current and decode the information that the modulators encode into the current draw. The custom current-sampler has its own Arduino board to process the current signal. All Arduino board are stand-alone, and are not connected to each other in any way.

In Figure 5.22 the raw data that is collected by the smart-meter can be seen. The current signal is shown here alongside the trigger output. In the raw data the charging peaks of the capacitor can be seen at timestamps: 5,

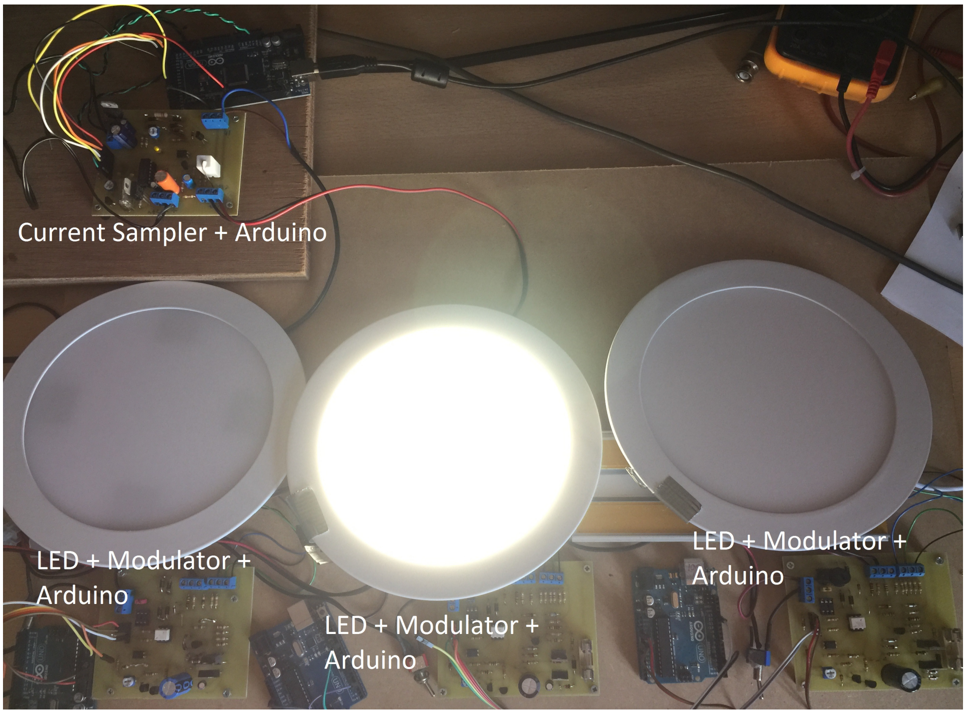


Figure 5.20: Picture of the AC testbed, consisting of three modulators each connected to three commercial LEDs. The modulators are controlled via three separate Arduino boards. Finally there is the current sampler with its own Arduino board.

15 and 25 ms. Note that these peaks indeed do not interfere with the modulated data. The modulated data and the charging peaks can be filtered out by the micro-controller with the help of the triggering circuits logical output. As discussed in subsection 5.2.1, the time that is available for modulation is 8 ms in a 10 ms period. Depending on the length of the ID L and the modulation frequency f, it can happen that this window of 8 ms is too small to encode the entire ID. In this case, the remaining bits of the ID which could not fit inside the first 8 ms window, will be transmitted in the next window. An evaluation with this testbed will be done in subsection 6.1.2.

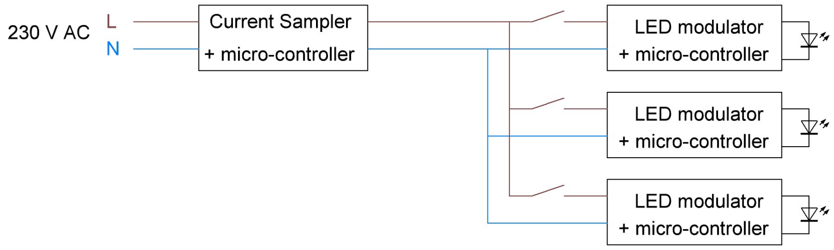


Figure 5.21: Architectural overview of the AC testbed. Three LED modulators, with switches to turn the LED on and off, are connected in parallel with each other and in series with the current sampler.

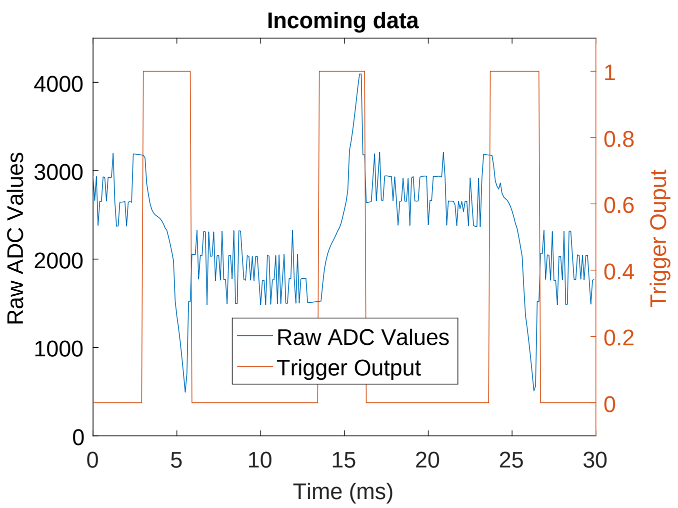


Figure 5.22: Incoming data to the AC current sampler. The raw ADC values are plotted as well as the triggering circuit output.

Chapter 6

Evaluation

Now that the IDs that will be assigned to each LED have been investigated in Chapter 4 and the hardware testbeds to encode the IDs and to measure the aggregated energy consumption have been explained in Chapter 5, it is time to evaluate the entire system. In this chapter, we evaluate the accuracy and timing of the system to detect if the lights are on and by looking at the aggregated energy consumption. Two small scale systems will be evaluated. The first system is with the DC testbed which has six lights. And the second system is the AC testbed which has three lights. These are only small scale systems, so to test the setup with a greater number of lights, software simulations are used to evaluate the accuracy and timing of a system with more than 100 lights.

In every setup, each light will be assigned an ID. This ID is shared with the smart-meter. The energy data that the smart-meter samples, is correlated with each ID and the outcome of that calculation will determine if the corresponding light is on or off. As was explained in section 4.3 the outcome of the correlation calculation is compared to a threshold. The outcome can either be less than or equal to the threshold or it can be greater than the threshold. For each of these cases, the outcome of the calculation can be correct or incorrect. For example, there can be too much interference so that one of the lights gets mistakenly identified as being on, while in reality it is off. So in total four categories must be considered, where the correlation outcome is denoted by R and the threshold is denoted by T:

- True Positive: $R > T$, indicating the light is on and the light is actually on.

- False Positive: $R > T$, indicating the light is on but the light is actually off.
- True Negative: $R \leq T$, indicating the light is off and the light is actually off.
- False Negative: $R \leq T$, indicating the light is off but the light is actually on.

When there is a system for which we can classify the results as false/true-positives and false/true-negatives, we can evaluate the system's performance by investigating the precision and recall. The definitions of precision and recall can be found in Equation 6.1 and Equation 6.2 respectively, where tp stands for the number of true-positives, fp stand for the number of false-positives and fn stand for the number of false-negatives.

$$\text{Precision} = \frac{tp}{tp + fp} \tag{6.1}$$

$$\text{Recall} = \frac{tp}{tp + fn} \tag{6.2}$$

This method provides two metrics to consider. Ideally we would like only one metric and draw conclusions based on that. We can take the weighted average of the two metrics. This is called the F-measure [39] and is the harmonic mean of the precision and recall. The definition of the F-measure can be seen in Equation 6.3. When the F-measure is equal to 1, the system has perfect accuracy. The lower the F-measure is, the more false-negatives and/or false-positives have occurred.

$$F = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{6.3}$$

The next sections will explain the evaluation of the hardware testbeds and the software simulation.

6.1 Hardware

The following two sections will describe the evaluation of the designed hardware and software, first for the DC testbed and then the AC testbed. All

LEDs will be assigned an ID that has the same length. As discussed in Chapter 4, the correlation calculations only hold when CDMA codes are used from the same set. The ID is programmed in the micro-controller. Based on the length of this ID, a certain amount of LEDs are supported, which is explained in section 4.3. But when the number of lights in the building increases, the new total number of lights may exceed the number of IDs. When this happens all the IDs must be changed to another ID of a different length that does support the new amount of lights. This may be done via a photo-diode as a receiver for each light, to receive the new ID. In this way each light can be used again, even when the total number of lights increases.

In the following evaluation sections, all attached LEDs will be modulating with a different ID per LED. This represents the worst case: Every LED is modulating and thereby causing interference. Since every LED will be modulating we can evaluate if the state of the LEDs can indeed be identified as being on. If this is not the case, this will be classified as a false-negative. But for completeness we must also evaluate what happens when the system tries to identify an ID which corresponds to an LED that is off. For this reason, an extra ID will be used to represent an LED in an off state. The raw ADC signal will be showed as well as the steps that are required in order to process the raw data such that the correlation calculations can be performed. Next, the correlation graph will be showed for an ID that is being used to modulate one of the LEDs as well as the correlation graph for the extra ID which represents an LED in an off state. Finally the F-measure will be showed for these correlation graphs.

6.1.1 DC

The DC testbed has six LEDs, as explained in subsection 5.1.3. Therefore six IDs will be used for the LEDs with a seventh code used to represent an LED in an off state. From Table 4.4, we can see that for $m = 6$, where m is the number of simultaneous transmitters such that no destructive interference takes place, requires a code length of 511 or higher. With the DC testbed the experiments have been performed with a constant modulation frequency of 1 kHz.

In Figure 6.1 the raw ADC data from the DC testbed can be seen. The left y-axis represents the raw ADC data and the right y-axis is converted to current. Six LEDs are continuously and simultaneously modulating with different starting times. From the figure, seven horizontal lines can be seen. Each line represents how many LEDs are on: When all LEDs are encoding a

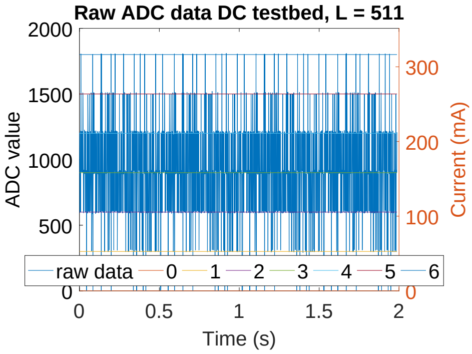


Figure 6.1: Raw ADC data from the DC testbed. With seven distinguishable entries, following the on-state of the combinations of LEDs. With a sequence length of 511.

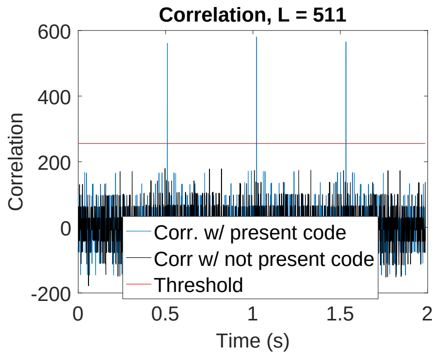


Figure 6.2: Correlations results from the IDs to identify which LEDs are on and off, with the decision threshold. With a sequence length of 511.

'0', the current will also be zero, when only one of the six LEDs is encoding a '1' the current is roughly 50 mA, when exactly two LEDs are encoding a '1', the current is roughly 100 mA, and so on. In contrast to the AC case which will be explained in the next section, in the DC case there is only positive current, so this raw data is already in a form for which we can start calculating the correlation for the ID of an LED.

In Figure 6.2, two signals and a line are plotted. The straight line represents the threshold, which is based on the length of the codes used, see section 4.3. The other two signals represent the outcome of the correlation with the ID of two LEDs. One of those IDs is from an LED which is on and the other ID is from the LED which represents an LED in an off state. The correlation results from the LED which is in an off state, stay below the threshold for all points in time. This is a good result, since being below the threshold line means that the corresponding LED is off, which in this case is actually true.

The other correlation result shown in Figure 6.2 comes from the ID which corresponds to an LED which is on and modulating. The results of this correlation show three noticeable peaks which are above the threshold. These peaks indicate that the ID is present in the sampled current signal by the smart-meter. There are three peaks, because the LED is continuously transmitting its ID and the length of the ID in combination with the modulation frequency can be transmitted three times in the two second window. All the peaks mean the same thing: The ID is present in the sampled current signal. Since this ID represents an LED which is on, these peaks are correct

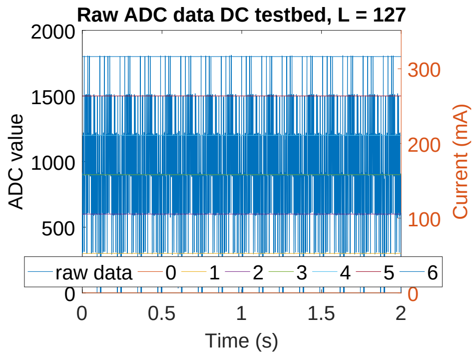


Figure 6.3: Raw ADC data from the DC testbed. With seven distinguishable entries, following the on-state of the combinations of LEDs. With a sequence length of 127 on the DC testbed.

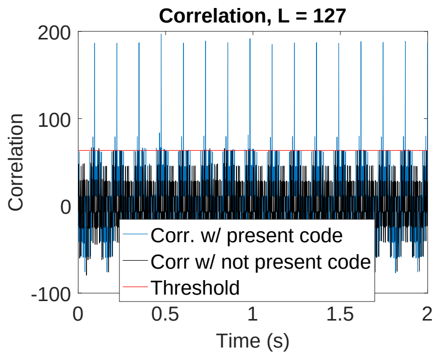


Figure 6.4: Correlations results from the IDs to identify which LEDs are on and off, with the decision threshold. With a sequence length of 127.

results as they indicate that the LED is indeed on, which is the case.

The evaluation as discussed above for the DC testbed has been run more than ten times. And for each time the duration of the run was approximately five minutes. Each run showed the same results as were shown in Figure 6.2. As discussed, all correlation results shown in Figure 6.2 are correct. This means that there are only true-positives and true-negatives. So the precision, recall and the F-measure are all equal to 1 according to Equation 6.1, Equation 6.2 and Equation 6.3, respectively.

To show how the accuracy of the system would change when there are false positives and/or false negatives another code length is chosen which does not support six LEDs modulating at the same time. A length of 127 is chosen, since it may only have three concurrent modulating LEDs (Table 4.4). Again the six LEDs are continuously and simultaneously modulating with different starting times. For which the raw ADC data can be seen in Figure 6.3 on the left y-axis and on the right y-axis the current. And the correlation results can be found in Figure 6.4. In the correlation figure, large peaks can be seen that cross the threshold line, these are the peaks that represent the ID of an LED that is on. But also other results can be seen that cross the threshold line. These are the false-positives, they occur because this code length can not support more than three simultaneous LEDs and the interference is too much. The F-measure in time for these correlation results can be found in Figure 6.5. And the precision, recall and F-measure can be found in Figure 6.6. Initially the F-measure is high ($\approx$ 0.7), because not many false-positives have occured yet. But over time more and more

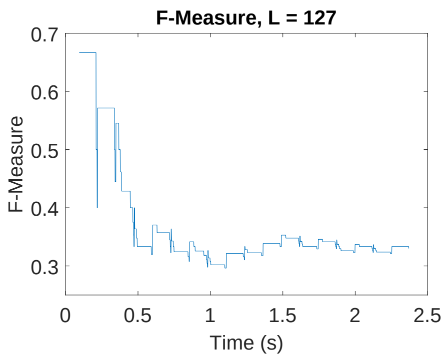


Figure 6.5: F-Measure of DC testbed correlation (Figure 6.4), with sequence length of 127.

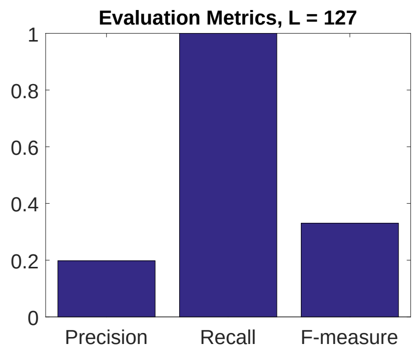


Figure 6.6: Evaluation metrics of DC testbed correlation (Figure 6.4), with sequence length of 127.

false-positives occure and the F-measure drops lower to approximately 0.3.

6.1.2 AC

The AC testbed has three commercial LEDs, as explained in subsection 5.2.3. Therefore three IDs will be for the LEDs with a fourth ID being used to represent an LED in an off state. From Table 4.4, we can see that for $m = 3$, the number of simultaneous transmitters such that no destructive interference takes place, requires a code length of 127 or higher. With the AC testbed, the experiments were performed with a constant modulation frequency of 10 kHz.

In Figure 6.7 the incoming signals to the current sampler can be seen. The raw data from the ADC can be seen as well as the output of the triggering circuit. From 2.5 ms until 6 ms in the figure, the charging peaks of the capacitor can be seen as explained in subsection 5.2.1. In the same time window, the output of the triggering circuit can also be seen. The triggering output is used to filter out the charging peaks and only the aggregated IDs are left. The aggregated IDs can be seen in the time window from 6 ms until 14 ms. This is the window which holds the encoded data and that data is retrieved by using the triggering circuit to detect when to start and stop looking for the encoded data.

As is the case in an AC environment, the current that will flow is both positive and negative. As explained in subsection 5.2.2 to overcome this bipolarity problem the voltage measured over the series resistor is added with a constant reference voltage in order for the ADC to only measure

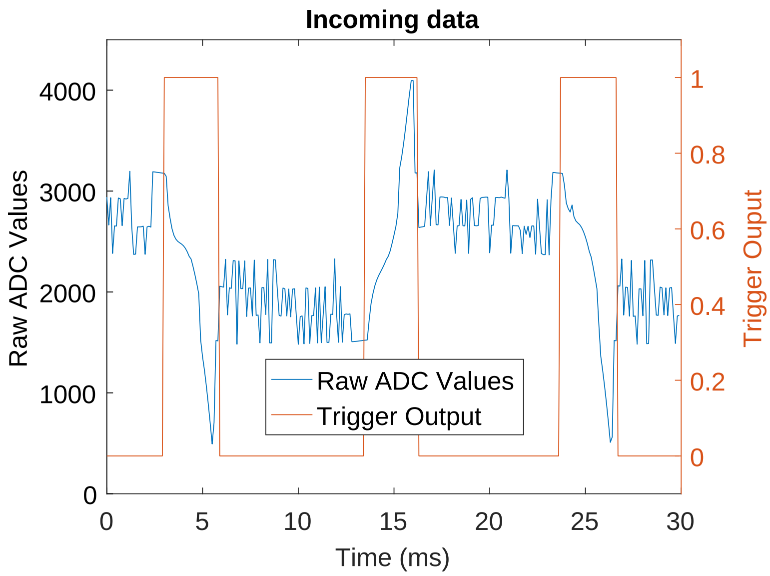


Figure 6.7: Incoming data to the AC current sampler. The raw ADC values are plotted as well as the triggering circuit output. Gold sequence of length 127.

positive voltages. The encoded data that is shown in Figure 6.7 is now filtered out with help of the triggering circuit. The next step is to subtract the reference voltage from each sample of the encoded data. And then the absolute value of the result is considered. The result of the processing of the raw ADC data can be seen in Figure 6.8. In this figure the ADC values are on the left y-axis and the current on the right y-axis.

In Figure 6.8, four distinguishable current levels can be seen. When all the LEDs are encoding a '0', the current is (almost) zero, when one LED is encoding a '1' the current is approximately 100 mA and so on. With this processed signal the correlation calculations can be done to identify which LEDs are on and which are off.

Now that the raw ADC signal is processed, the correlation can be calculated. In Figure 6.9, two signals and one line are plotted. The straight line represents the threshold, which is based on the length of the codes used, see section 4.3. The other two signals represent the outcome of the correlation with the ID of two LEDs. One of those IDs is from an LED which is on and the other ID is from the LED which represents an LED in an off state. The correlation results from the LED which is in an off state, stay below the threshold for all points in time. This is a good result, since being below the threshold line means that the corresponding LED is off, which in this

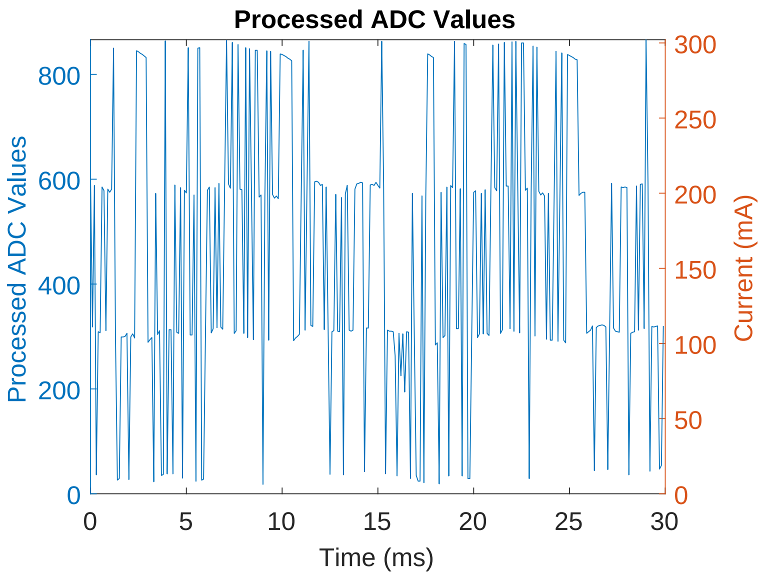


Figure 6.8: Fully processed ADC values, where four distinguishable levels can be seen. With the on-state from zero to all three LEDs, on the AC testbed.

case is actually true.

The other correlation result, shown in Figure 6.9 comes from the ID which corresponds to an LED which is on and modulating. The results of this correlation show two noticeable peaks above the threshold. These peaks indicate that the ID is present in the sampled current signal by the smart-meter. There are two peaks, because the LED is continuously transmitting its ID and the length of the ID in combination with the modulation frequency can be transmitted two times in the 30 ms window. All the peaks mean the same thing: The ID is present in the sampled current signal. Since this ID represents an LED which is on, these peaks are correct results.

The evaluation as discussed above for the AC testbed has been run in a similar fashion compared to the DC testbed evaluation. The evaluation was run for more than ten times. And for each run the duration was approximately five minutes. Each run showed the same results as presented in Figure 6.9. All the results we obtained, were correct. This means that there are only true-positives and true-negatives. So the precision, recall and the F-measure are all equal to 1.

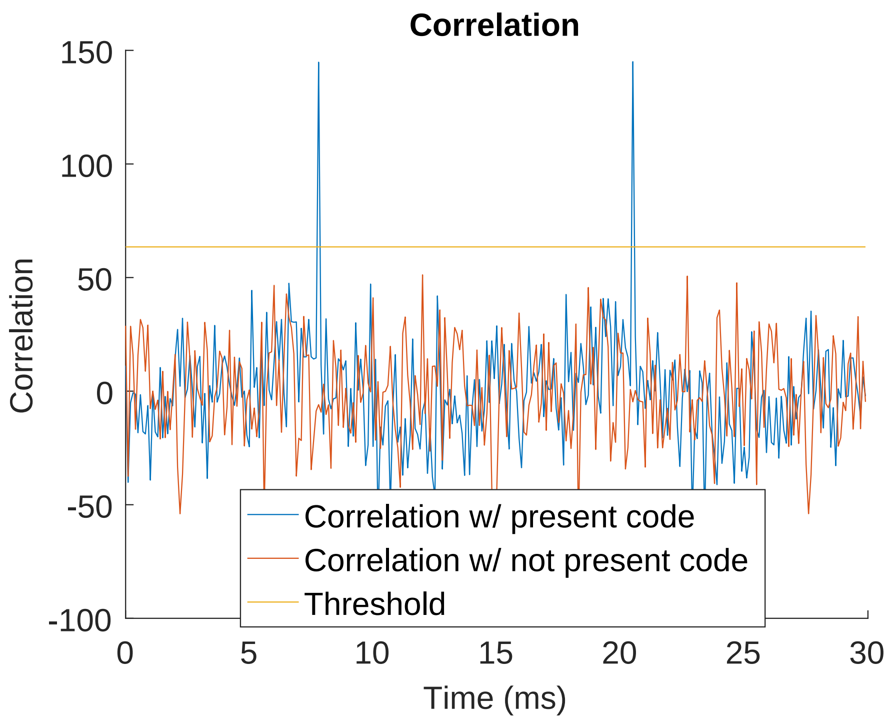


Figure 6.9: Correlations results from the IDs to identify which LEDs are on and off, with the decision threshold. With a sequence length of 127 on the AC testbed.

6.2 Simulation

The evaluation with actual hardware has now been done. With that evaluation the continuous transmission method was used for each LED as detailed in subsection 4.3.1. The software simulation that will be done in this chapter, will use the probabilistic method as explained in subsection 4.3.2. Both of the interference solutions will then have been used to evaluate the system.

Before the actual simulation is performed, the parameters of the probabilistic approach such as ϵ need to be set. $1 - \epsilon$ is the probability that for every point in time the number of LEDs that are modulating will not exceed m, as discussed in subsection 4.3.2. This will determine if interference can occur and therefore this will determine the accuracy of the system.

For the rest of this simulation a Gold sequence set with a LFSR length of $n = 7$ is used. So all the IDs that are assigned to the LEDs have length $L = 2^n - 1 = 127$ and there will be $N = 2^n + 1 = 129$ unique LEDs.

This simulation does the following:

- For the error ϵ the probability p and the time to complete the k slots are calculated (See subsection 4.3.2).
- At each time slot, each LED will use the probability p to determine if it will transmit its ID.
- If a certain LED transmits its ID, the ID is added to a signal vector which represents the current draw in the hardware testbeds.
- When the entire signal has been constructed, the decoding process begins. All 129 IDs are used to decode the incoming signal for every time slot.
- During the decoding, the results are analyzed and together with the information that is known when a particular LED transmitted at what time, the results are classified into four categories: true-positive, true-negative, false-positive and false-negative. This is done for each time slot.
- With those four categories, the precision, recall and the F-measure are calculated.

Let's set $\epsilon = 0.1$, meaning that the probability that there will be less than or equal to m LEDs modulating for every point in time will be 0.9, where m is the maximum number of simultaneous transmitters such that no destructive interference takes place. Now that the simulation steps have been explained and all variables have been set, the simulation itself is performed.

The results of the simulation can be seen in Figure 6.10. In this figure the number of concurrent transmitters over time are represented by the blue dots and the left y-axis. The maximum number of simultaneous transmitters such that no destructive interference takes place, m, is represented with the green dashed line. When the number of concurrent transmitters is below or equal to the m-line, it is guaranteed that no interference will occur, as proven in section 4.3. But when the number of concurrent transmitters is above the m-line, it may not be possible to successfully decode the signal This depends on what the cross-correlation with the other IDs is, if they cancel each other out or if they enhance each other.

In Figure 6.10 the F-measure is represented by the red line. At each point in time the correlation calculations are performed for all the 129 IDs. All the true-positives, true-negatives, false-positives and false-negatives are used to

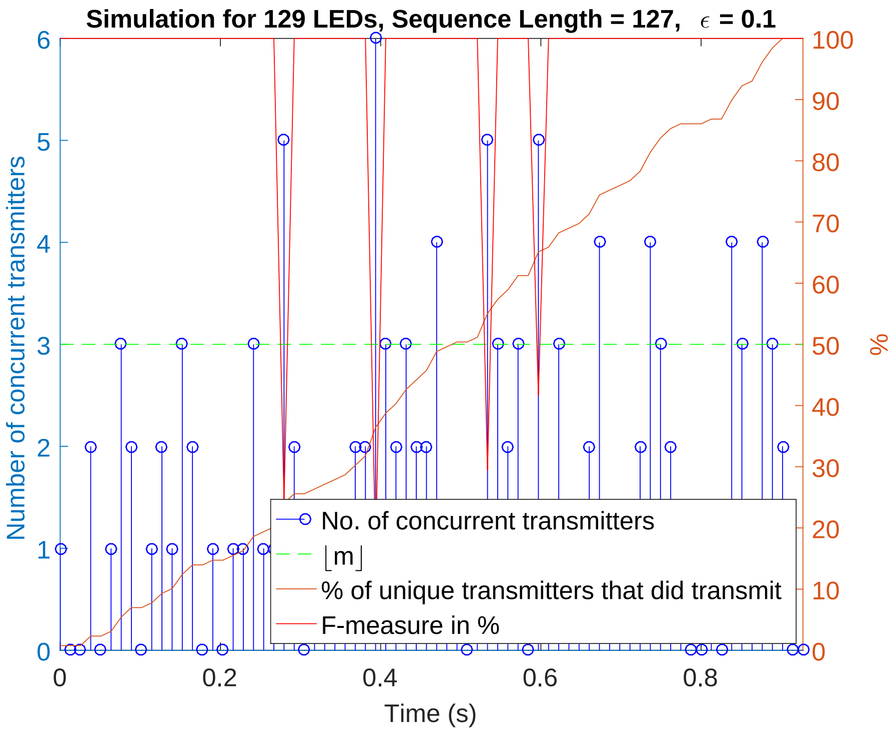


Figure 6.10: Results of the simulation with $\epsilon = 0.1$. The figure shows the number of concurrent transmitters at each point in time along with m and the percentage of transmitters that have transmitted their ID. Also the F-measure is shown.

calculate the F-measure according to Equation 6.3. In Figure 6.10, it can be seen that whenever the number of concurrent transmitters is below the m-line, the F-measure is equal to 1. For example, from time 0 s until 0.25 s, the number of concurrent LEDs transmitting their IDs never exceed the m-line and as a result the F-measure remains at 1 during this time. This means that everything in the decoding process went well, there were no false-positives or false-negatives.

When the number of concurrent transmitters is above the m-line, two things can happen: either there is enough interference to cause false-positives and/or false-negatives, or there is not enough interference and everything will go well. An example of the latter case: From 0.6 s until the end, the number of concurrent LEDs which are transmitting their IDs goes above the m-line four times but the F-measure stays 1 during this window. This is an example when the interference created by these IDs is not high enough to cause problems when decoding. Examples for when the interference is too high to successfully decode the data can be seen at the following times in the Figure 6.10: 0.27 s, 0.4 s, 0.55 s and 0.6 s. At each of these timestamps the number of concurrent LEDs which are transmitting their IDs is so high that they cause so much interference that there are problems with the decoding. And as a result the F-measure has significant drops at each of those timestamps. For example at time 0.4 s there were 31 false-positives, 0 false-negatives and 5 true-positives. At this time there were 6 LEDs transmitting, meaning that one of the LEDs did not get identified as being on and 31 LEDs were classified as being on while they were actually off.

In Figure 6.10 the percentage of transmitters that have successfully transmitted their IDs to the smart-meter can also be seen (orange line). This line is approximately a linear line due to the fact that each LED has a limited amount of slots to transmit and each slot has the same probability of being chosen so the number of transmitters that are transmitting is approximately uniformly divided.

The percentage of time where the number of concurrent LEDs is above m is roughly equal to 12 %. This number will get closer and closer to ϵ the longer the simulation takes.

This simulation has been performed more than ten times. Each run was slightly different due to the probabilistic approach. But every time similar results could be seen concerning the accuracy and the total time taken. The total time that the simulation takes is roughly 1 s. The time corresponds with the theoretical time calculated in subsection 4.3.2 as can be seen in Table 4.6. So this was a fast time but multiple errors occurred as can be seen from the drops in the f-measure.

The same simulation is done again, but now with $\epsilon = 0.001$. The plots can be seen in Figure 6.11 which show the same kind of information as in the simulation figure shown before. This simulation with the new value for ϵ has also been performed more than ten times. The distribution of the number of concurrent LEDs was slightly different each time but the F-measure was always the same. At no time the number of concurrent LEDs goes above the m-line, so there will be no interference as can also be seen by looking at the F-measure which is a constant 1. So $\epsilon = 0.001$ is an accurate setup to use all the IDs that are provided by the LFSR to construct the Gold sequences. But the simulation takes roughly 4 s. The time corresponds with the theoretical time calculated in subsection 4.3.2 as can be seen in Table 4.6. This is still a relative small amount of time, but still 4 times larger than the previous simulation. This simulation represents a slow but accurate result.

These simulations show that for the probabilistic method there is a clear trade off between time and accuracy. For a given set of IDs, a low value for

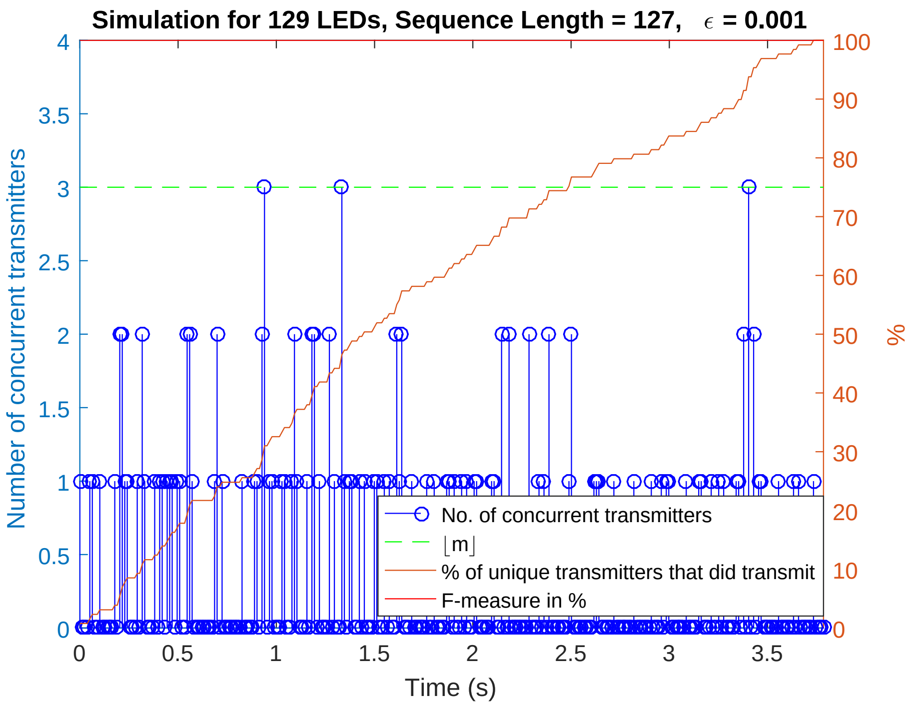


Figure 6.11: Results of the simulation with $\epsilon = 0.001$. Figure shows the number of concurrent transmitters at each point in time along with m and the percentage of transmitters that have transmitted. Also the F-measure is shown.

ϵ gives high accuracy but also a longer time. A high value for ϵ gives lower accuracy but the time becomes shorter.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

The aim of this thesis is to find out if lights could be identified as being on or off, by measuring the aggregated energy draw of those lights, with the help of VLC and a single smart-meter. CDMA codes have been investigated and compared to see which one is the best suited for this scenario. Two solutions have been discussed to overcome the interference problem that comes with these types of codes. One solution allows the LEDs to transmit at all times without causing too much interference, while the other solution uses probabilities to determine which LEDs will transmit. To validate the applicability of these codes, two testbeds were developed, one for DC and one for AC. When the correct codes are used, each individual light in each testbed could be successfully identified as being on or off in a timely manner. For larger systems a simulation was performed which shows that there can be made a trade-off between time and accuracy. But the simulation showed that if a high level of accuracy is required, the lights could still be identified in a timely manner.

7.2 Future Work

This work is only the first step to disaggregate which lights are on and off. There remains more work to be done, below there are ideas for future work.

7.2.1 Power Limiting Capacitor

The solution of using a capacitor in order to limit the power dissipated in the current source, introduced in subsection 5.2.1 can be further investigated. In particular what the drawbacks or benefits are of the current that is phase-shifted. And if the triggering circuit will still function properly or what the modifications are that need to be made in order for this scheme to work properly.

7.2.2 Other Appliances

As the testbeds represent cases where only these lights were connected, an important question rises: What will happen when other appliances are connected. These could for example be an incandescent light bulb or a refrigerator. If the smart-meter samples an aggregated signal from many appliances, one can apply filters to obtain the signals associated with the IDs of the LEDs, because the LEDs are modulating at a certain constant frequency.

7.2.3 Dimming Lights

LEDs can be often too bright for a persons liking, so the lights are dimmed. Dimming of an LED can be done in two ways: PWM, by lowering the duty cycle and so less power is dissipated by the LED or by limiting the current that flows through the LED. Since the LEDs are modulating via an OOK scheme, PWM cannot be used, so instead current limiting must be done in order to dim the lights. But the CDMA codes are designed to work when all LEDs have amplitude or in this case the same current. By dimming the LED and changing the current the codes may not work anymore. A solution for this can be that each LED will have multiple dimming levels and that for each dimming level other frequencies are used to modulate at. A filter can then be used to filter LEDs according to their modulating frequency and then use the proposed method.

7.2.4 Transmitting Data

With the current state of this system, the LEDs can be identified as being on or off by detecting if their unique code is present or not. This can be seen as transmitting data, namely one bit, if the LED is on or off. If the LED needs to send other data about the status of the light two approaches can be thought of:

- The unmodified code assigned to the light will be transmitted for the data-bit '0' and the negation of the code will be transmitted for the data-bit '1'. This gives a problem with the definition of the cross-correlation, which is defined only for the unmodified codes. When the negation of the codes is also used, the cross-correlation between the LEDs that are transmitting is no longer bounded by the mathematical formula and all the calculation on how many LEDs can transmit at the same time can no longer be used with these codes. It can be investigated if other codes do have a cross-correlation definition where the negation of the codes is also taken into consideration.

- Assign two unique codes from the same set to each light. The lights will send the first code for the data-bit '0' and the second code for the data-bit '1'. Since the cross-correlation is defined for the codes from the same set this solution should not yield any problems. But further investigation may be required to see if this is a viable solution.

Bibliography

[1] Commercial 230 V AC LED. https://nl.aliexpress.com/item/SMD-5730-E27-E14-LED-Lamp-5730SMD-LED-Lights-Corn-Led-Bulb-24-36-48-56/32629651035.html?spm=2114.48010608.4.143.cOIHcN. [Online; accessed 4-September-2016].

[2] New Jersey Semi-Conductors MTH6N60. http://www.njsemi.com/datasheets/MTM6N60.pdf. [Online; accessed 4-September-2016].

[3] Pseudo-Noise Sequences. http://paginas.fe.up.pt/~hmiranda/cm/Pseudo_Noise_Sequences.pdf. [Online; accessed 4-September-2016].

[4] Siemens BUZ80. http://www.bucek.name/pdf/buz80.pdf, 1996. [Online; accessed 4-September-2016].

[5] Vishay 2N7000. http://www.vishay.com/docs/70226/70226.pdf, 2001. [Online; accessed 4-September-2016].

[6] Fairchild 2N3904. https://www.fairchildsemi.com/datasheets/2N/2N3904.pdf, 2002. [Online; accessed 4-September-2016].

[7] Fairchild BC556. https://www.fairchildsemi.com/datasheets/BC/BC556.pdf, 2002. [Online; accessed 4-September-2016].

[8] Microchip MCP3204/3208. http://ww1.microchip.com/downloads/en/DeviceDoc/21298c.pdf, 2002. [Online; accessed 4-September-2016].

[9] Fairchild H11L1M. https://www.fairchildsemi.com/products/optoelectronics/high-performance-optocouplers/high-speed-logic-gate/H11L1M.html, 2005. [Online; accessed 4-September-2016].

[10] Vishay SFH617A. http://www.vishay.com/docs/83740/sfh617a.pdf, 2005. [Online; accessed 4-September-2016].

[11] Fairchild LM78XX. https://www.fairchildsemi.com/datasheets/LM/LM7805.pdf, 2006. [Online; accessed 4-September-2016].

[12] Fairchild LM78XX. https://www.fairchildsemi.com/datasheets/LM/LM7824.pdf, 2006. [Online; accessed 4-September-2016].

[13] Vishay 4N35. http://www.vishay.com/docs/81181/4n35.pdf, 2010. [Online; accessed 4-September-2016].

[14] On Semiconductor MJE13009G. http://www.onsemi.com/pub_link/Collateral/MJE13009-D.PDF, 2011. [Online; accessed 4-September-2016].

[15] Allegro MicroSystems ACS712. http://www.allegromicro.com/en/Products/Current-Sensor-ICs/Zero-To-Fifty-Amp-Integrated-Conductor-Sensor-ICs/ACS712.aspx, 2013. [Online; accessed 4-September-2016].

[16] Texas Instruments LM336z. http://www.ti.com/lit/ds/symlink/lm136-5.0.pdf, 2013. [Online; accessed 4-September-2016].

[17] Texas Instruments ISO7241M. `http://www.ti.com/lit/ds/symlink/iso7241m.pdf`, 2016. [Online; accessed 4-September-2016].

[18] Nipun Batra, Amarjeet Singh, and Kamin Whitehouse. Neighbourhood nilm: A big-data approach to household energy disaggregation. *arXiv preprint arXiv:1511.02900*, 2015.

[19] Paolo Bertoldi and Bogdan Atanasiu. Electricity consumption and efficiency trends in european union–status report 2009. *JRC-IE, Ispra, Italy*, 2009.

[20] Sarah Darby et al. The effectiveness of feedback on energy consumption. 2006.

[21] E. H. Dinan and B. Jabbari. Spreading codes for direct sequence cdma and wideband cdma cellular networks. *IEEE Communications Magazine*, 36(9):48–54, Sep 1998.

[22] Frank Englert, Patrick Lieser, and Marius Rettberg-Paplow. Lightmon: Apportioning the effect of light switching events on the electricity consumption of buildings. In *Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks*, EWSN '16, pages 77–88, USA, 2016. Junction Publishing.

[23] Jon Froehlich, Eric Larson, Sidhant Gupta, Gabe Cohn, Matthew Reynolds, and Shwetak Patel. Disaggregated end-use energy sensing for the smart grid. *IEEE Pervasive Computing*, 10(1):28–39, 2011.

[24] R. Gold. Maximal recursive sequences with 3-valued recursive cross-correlation functions (corresp.). *IEEE Transactions on Information Theory*, 14(1):154–156, Jan 1968.

[25] Sidhant Gupta, Matthew S. Reynolds, and Shwetak N. Patel. Electrisense: Single-point sensing using emi for electrical event detection and classification in the home. In *Proceedings of the 12th ACM International Conference on Ubiquitous Computing*, UbiComp '10, pages 139–148, New York, NY, USA, 2010. ACM.

[26] L Halonen, Elno Tetri, and P Bhusal. Guidebook on energy efficient electric lighting for buildings. *Aalto University, School of Science and Technology, Department of Electronics, Lighting Unit, Espoo2010*, 2010.

[27] Martin Hoch. Comparison of plc g3 and prime. In *IEEE Int. Symposium on Power Line Communications and Its Applications (ISPLC 2011)*, pages 165–169, 2011.

[28] J.K. Holmes. *Spread Spectrum Systems for GNSS and Wireless Communications*. Number v. 45 in GNSS technology and applications series. Artech House, 2007.

[29] Deepak Kedia. Comparative analysis of peak correlation characteristics of non-orthogonal spreading codes for wireless systems. *arXiv preprint arXiv:1206.2138*, 2012.

[30] Kimmo Kettunen. Code selection for cdma systems. *Department of Information Studies, University of Tampere, Finland*, 1997.

[31] J Zico Kolter and Matthew J Johnson. Redd: A public data set for energy disaggregation research. Citeseer, 2011.

[32] T. Komine and M. Nakagawa. Integrated system of white led visible-light communication and power-line communication. *IEEE Transactions on Consumer Electronics*, 49(1):71–79, Feb 2003.

[33] Abhijit Mitra. On pseudo-random and orthogonal binary spreading sequences. *Int. J. Information Techn*, 4(2):137–144, 2008.

[34] RN Mutagi. Pseudo noise sequences for engineers. *Electronics & Communication Engineering Journal*, 8(2):79–87, 1996.

[35] Annual Energy Outlook et al. Energy information administration. *Department of Energy*, 2010.

[36] Martin Patoka. Power over ethernet eases design implementations. *Power Electronics*, 2003.

[37] Sridhar Rajagopal, Richard D Roberts, and Sang-Kyu Lim. Ieee 802.15. 7 visible light communication: modulation schemes and dimming support. *IEEE Communications Magazine*, 50(3):72–82, 2012.

[38] Huijuan Shao, Manish Marwah, and Naren Ramakrishnan. A temporal motif mining approach to unsupervised energy disaggregation: Applications to residential and commercial buildings. In *AAAI*, 2013.

[39] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437, 2009.

[40] Stephan Spiegel and Sahin Albayrak. Energy disaggregation meets heating control. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, pages 559–566. ACM, 2014.

[41] L.-T. Wang and E. J. McClusky. Linear feedback shift register design using cyclic codes. *IEEE Trans. Comput.*, 37(10):1302–1306, Oct. 1988.

[42] Yongxiang Xia, Pu Li, Xiuming Shan, and Yong Ren. Cyclically orthogonal subsets of walsh functions. In *Communication Systems, 2002. ICCS 2002. The 8th International Conference on*, volume 1, pages 107–111 vol.1, Nov 2002.

Appendix A

DC Testbed

In this chapter, the schematic of the DC testbed can be found in Figure A.1. The relevant datasheets for parts used, can be found in: [6] and [7].

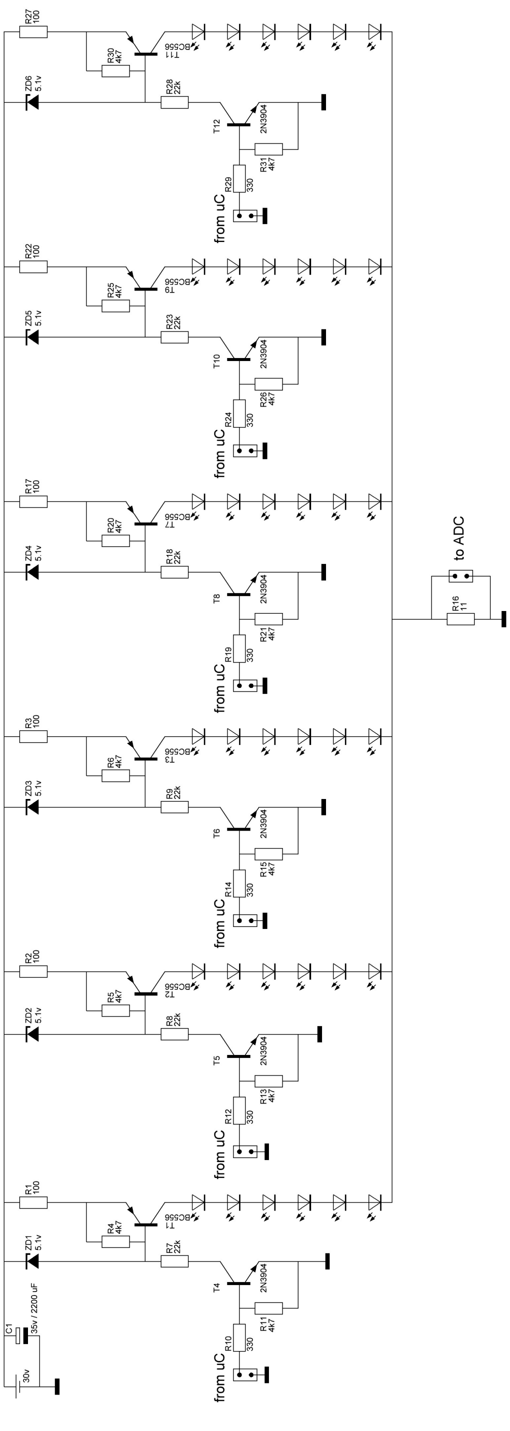


Figure A.1: Schematic of the DC testbed, to modulate six individual LEDs and measure the combined current.

Appendix B

Modified 230 V AC LED Schematic

In this chapter, the schematic of the original and modified circuit of a commercial 230 V AC LED can be found in Figure B.1. The relevant datasheets for parts used, can be found in: [13], [2] and [14].

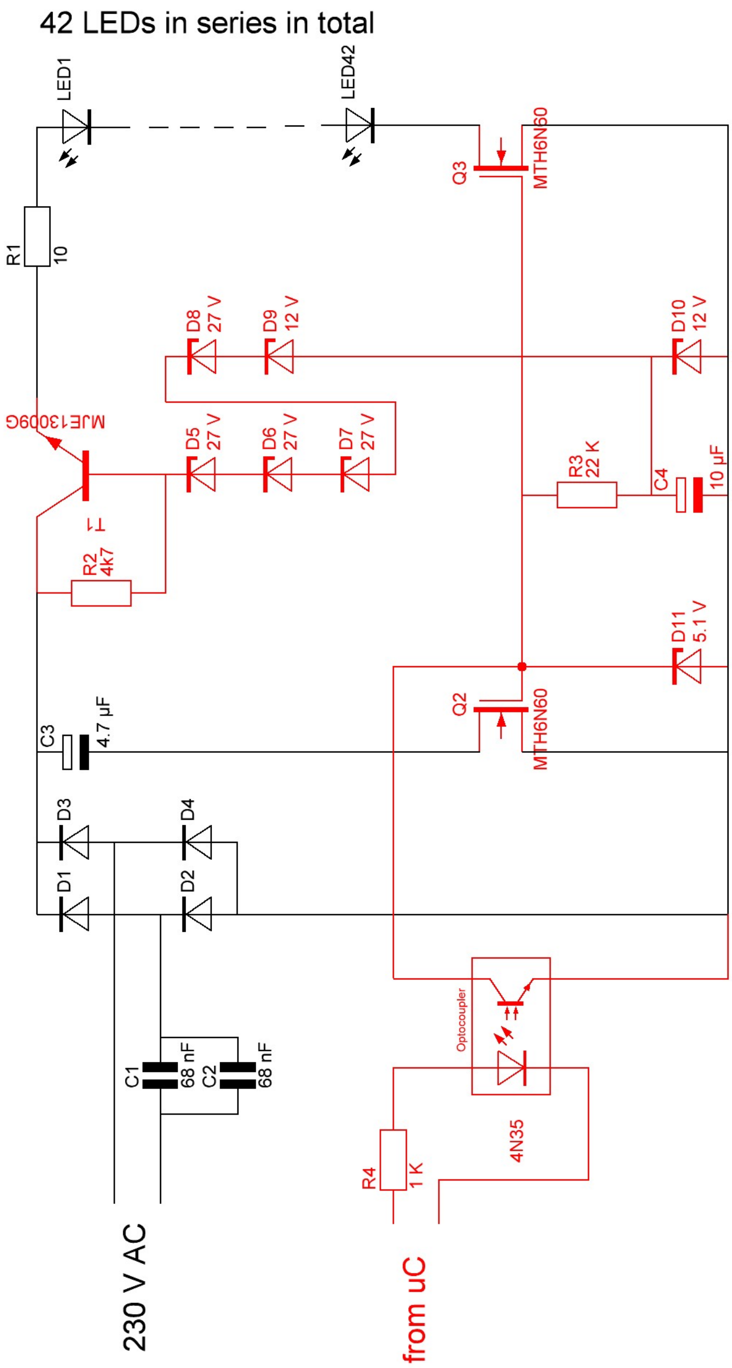


Figure B.1: Schematic of the modified commercial 230 V AC LED. Everything in black is from the original schematic. Everything in red is added in order to modulate the LED safely with a micro-controller.

Appendix C

Custom LED Modulator Schematic

In this chapter, the schematic of the LED modulator can be found in Figure C.1. The relevant datasheets for parts used, can be found in: [11], [5], [10], [9], [4] and [6].

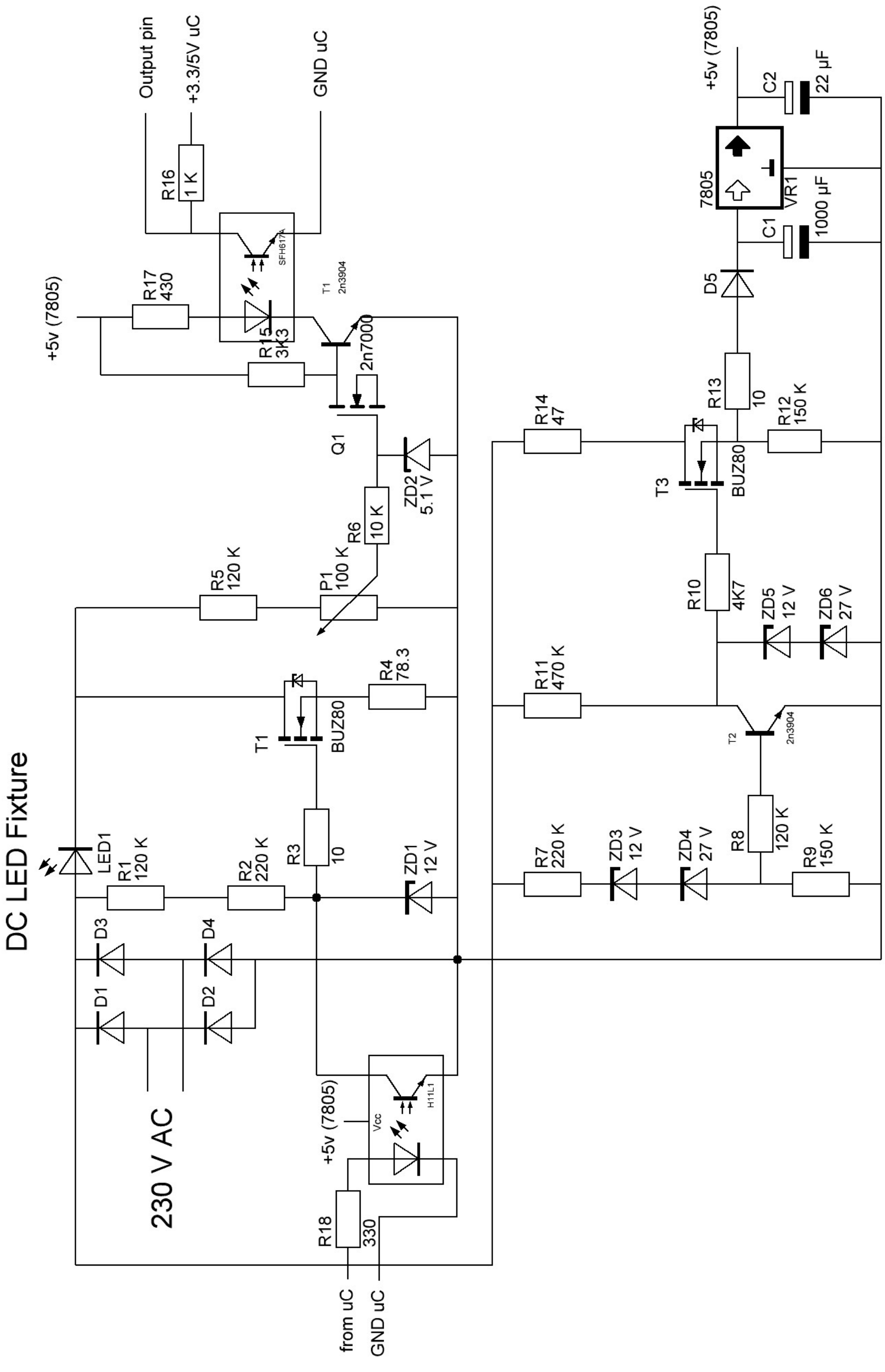


Figure C.1: Schematic of the LED modulator consisting of a current source with optocoupler, triggering circuit with optocoupler and non-disturbing voltage source.

Appendix D

Custom Current Sampler Schematic

In this chapter, the schematic of the current-sampler used to measure the current of the 230 V AC can be found in Figure D.1. The relevant datasheets for parts used, can be found in: [12], [11], [16], [8], [17], [5] and [10].

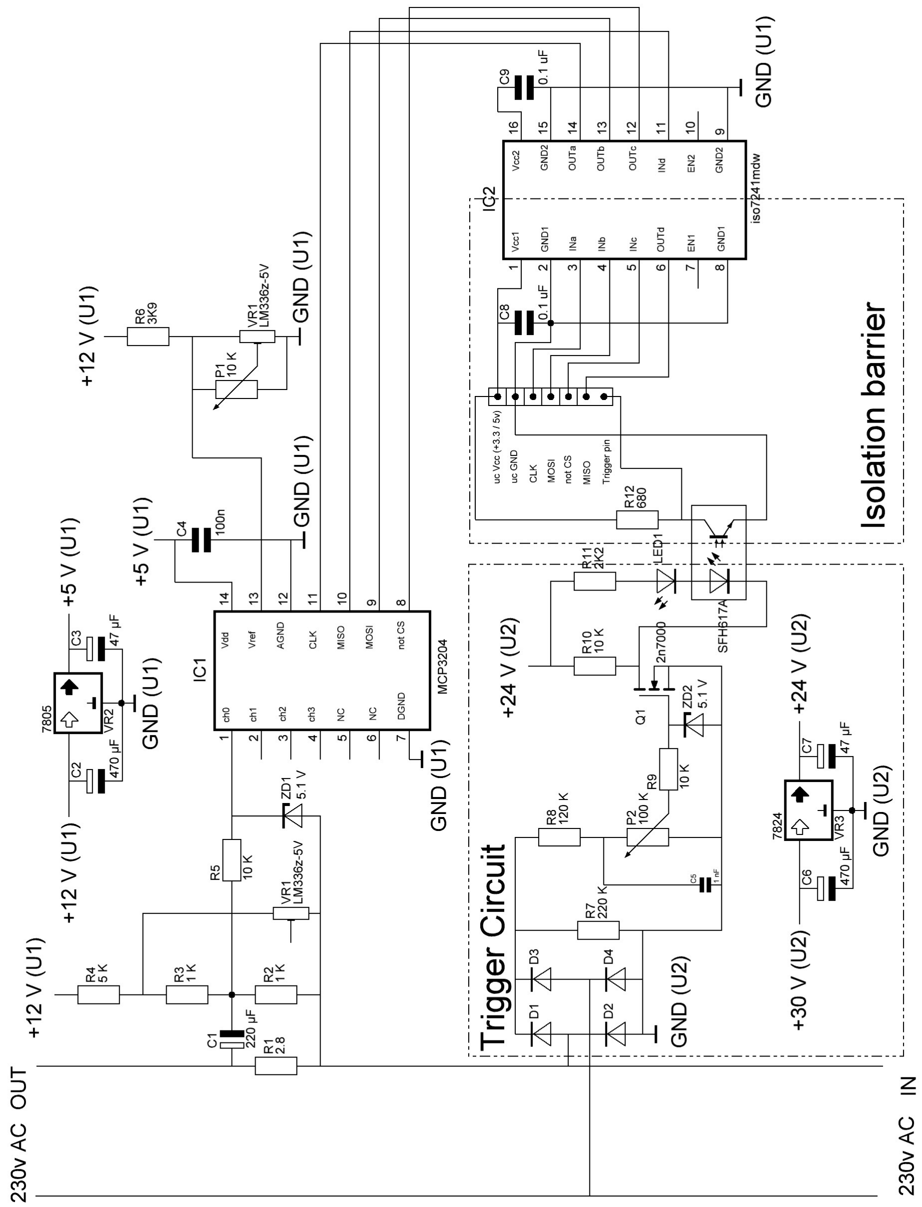


Figure D.1: Schematic of the current sampler, with an external ADC which is isolated from the micro-controller and with a triggering circuit.