Msc. Thesis

Applied Visible Light Communication for Robotics

Martino Secchi
msec@itu.dk

Supervisor:
Kasper Støy

June 2017

Todo list

much more introduction . . . 1
mention cheap and simple components, getting it to work is simple, getting it to work fast and far is very hard/ expensive . . . 1
mention fiber vs open space? . . . 1
talk about situated communication . . . 1
mention other forms of optical communication, like finer and infrared . . . 1
expand, and also add some applied robotics arguments, and communication . . . 2
do a graph of logical control and physical. . . . 4
in the experimental setup, it's considered almost exclusively low power DC LED . . . 4
add wiring? diagram for small setup and most importantly setup with ssr and big bulb . . . 6
introduce, motivation, also explain setup, it's almost always with small LED unless otherwise stated 6
stats for warmup . . . 6
table with times to warmup from x to y, eg 0-0, from 0 to 10, 0-20 ,0-30, ..., 10-0 10-10; then discuss meaningful milestones . . . 6
produce statistics for different ambient light interferences . . . 6
stats for big bulb . . . 6
stats for different distances . . . 6
stats for different angles . . . 7
talk about AC vs DC, discussion about power vs brightness vs distance vs speed . . . 7
remove outlier in the picture of ac? . . . 7
small graph serial -¿ control -¿ physical . . . 8
find a place for this . . . 8
information about clock rates of the boards . . . 8
rate of serial connection arduino to computer? . . . 8
how much decrease in 1 ms? . . . 8
prove that there is more reliability with ambient interference . . . 9
add more setups, 3 and 4 ms too . . . 9
solid state relay limitations . . . 9
redo this figure with a more recent transmission . . . 9
stats about reception rate, std deviation. . . . 12

Contents

1 Introduction **1**

2 Visible Light Communication **1**
2.1 Related work and technology description 1
2.2 Standard and specifications 2

3 System Overview **3**
3.1 Experimental setup 4
3.1.1 Transmitter 4
3.1.2 Receiver 5
3.1.3 List of components 5
3.1.4 Wiring 6

4 Physical layer **6**
4.1 Warmup time 6
4.2 Ambient light interference 6
4.3 Distance 6
4.4 Angle 7
4.5 Power source 7

5 Control layer **7**
5.1 Serial connection to a terminal 8
5.2 Reception rates 8
5.3 Transmission rates 8
5.4 Experiments and Results 9

6 Logical layer **9**
6.1 Reading the signal 10
6.2 Protocol 12
6.2.1 Additional fields 14
6.3 Results 14

7 Evaluation and Results **14**
7.1 Experimental setup 15
7.2 Experiments 15
7.3 Performance 15
7.4 Discussion 15

8 Discussion **15**

9 Conclusion **15**

Abstract

[This is my abstract, what I did and my results, in about 100 words]

1 Introduction

[This is some introduction. Here I introduce what visible light is, and the vision of how it could be used in general and in the specific scenarios for robot communication, environment to robot and robot to robot and why it would make sense to suggest it.]

Visible Light Communication (VLC) is a type of data communication achieved with the transmission of signals in the spectrum of visible light. This form of communication, compared to more classic ones, could potentially be beneficial in the context of mobile robotics among others, for the reason that light is a form of **situated** communication, where the message is not separated from the physical environment in which it has meaning.

This allows to transmit a message that includes additional information embedded in the medium of transmission rather than its content. Particularly relevant for mobile robotics is the information relative to localisation and direction, that can just be seen as the source of the light instead of it being encoded in the message. In this way, the content of the message and the properties of the transmission combined together produce the information to be received.

Such a technology could be used in it simplest form in the context of mobile robotics to provide an aid for environment navigation, or a form of remote control for agents' movement or for assistance in the execution of specific tasks. These are a few examples of mono directional, environment-to-robot communication systems. The technology could also be used in a slightly more complex scenario of robot-to-robot 1- or even 2-way communication between multiple agents to enable cooperation. The design of a prototype system will help analyse the properties of a generic VLC system, its performances and results in the given applicative scenario. The design process will be focused on simplicity and inexpensiveness where possible, by using mostly off-the-shelf and easily obtainable components.

...

In recent years optical communication has seen a great spread mostly thanks to optical fibre. As opposed to optical fibre communication, VLC is meant to be propagated wirelessly over an open space. By modulating a light source, like could be a common LED bulb, through a controller it is possible to encode messages from digital form into light signals. The simplest example of this encoding is a binary representation of data through light, where the presence of light represents a binary 1 and its absence a 0. This form of communication is a simplified variation of the technology known as Li-Fi, a term that stands for Light-Fidelity.

much more introduction

mention cheap and simple components, getting it to work is simple, getting it to work fast and far is very hard/ expensive

mention fiber vs open space?

talk about situated communication

mention other forms of optical communication, like finer and infrared

2 Visible Light Communication

[chapter about visible light communication in general (from thesis prep). Related work, plus current standards and specifications]

In this section there will be a summary of the findings that came through research on the subject. In particular, the research has been focused on Li-Fi systems with specific aim towards any information that would help in the implementation of one.

2.1 Related work and technology description

The term Li-Fi was first introduced by a german physicist, Harald Haas, co-founder of PureLiFi, in occasion of a TED Global talk on Visible Light Communication in 2011. [8] Since then, the term has been reported in many articles and has gained popularity as a common synonymous for Visible Light Communication. After the name was adopted by the Optical Wireless Communication (OWC) community with the launch of the LiFi Consortium in 2011, an industry group that promotes OWC technologies, it can be sometimes extended to describe general wireless data access points that use light, visible or otherwise. This includes also the infra-red and ultraviolet band.

The main reason why this technology is quickly gaining popularity in the research is because it potentially allows to unlock a vast amount of electromagnetic spectrum in the visible light region, unused for transmission.[9] This has been seen as a promising reaction to the saturation of the Radio Frequency (RF) spectrum, a very likely outcome of the huge success of wireless technology, also predicted by the US Federal Communication Commission[5].

A second reason is that transmission through light can achieve surprising high speeds. Starting from 2010, research had been able to improve the transmission rate further and further. In his TED talk in 2011 Haas demonstrated real time video streaming from a white LED at data rates up to 130 Mbps[8], while another group achieved over 513 Mbps[10]. In the following years, there have been continuous reports of improved data rates in transmission. A single white LED has been proven to transmit from about 1 Gbps[1], up to 3.5 Gbps[6], while 3.4 Gbps have been demonstrated with a single RGB LED[7].
This rates can be further improved by the use of arrays of light sources and more complex systems. The Mexican software company Sisoft together with scientists from the Autonomous Technological Institute of Mexico in 2014 reached the surprising data rate of 10 Gbps, setting the record.
Other appealing aspects to this technology are the low cost of implementation for the use of off-the-shelf LED bulbs and its security, for the reason that communication can be eavesdropped only in direct line of sight within short distance.

There are a few groups that are leading the research in the field of Visible Light Communication. One of these is the Li-Fi R&D Centre at the University of Edinburgh, of which prof. Harald Haas is the director. Another very important research group operating in the same area is Disney Research, often in collaboration with ETH Zurich. Another group worth mentioning is the Li-Fi Consortium, an international organisation formed by companies in optical wireless communication technology and research institutes. A good part of the research is also carried out by multiple other research groups worldwide, especially in Asia and India in particular.

expand, and also add some applied robotics arguments, and communication

2.2 Standard and specifications

Visible light communication is regulated by a standard similar to the one of wireless networks, in the same IEEE 802 family.[4] The IEEE 802.15.7 standard defines a draft of the physical layer (PHY) and the media access control (MAC) layer for VLC. According to Gordon Povey, former CEO at PureLiFi[11], the MAC layer as of April 2011 supports three multiple access topologies: peer-to-peer, star configuration and broadcast mode.
The Physical layer is divided into three types, that use different modulation schemes. The three modulation schemes are:

- On-Off Keying
- Variable pulse position modulation (VPPM)
- Colour shift keying (CSK)

On-Off Keying
On-Off Keying is the simplest modulation scheme. In this scheme, a digital 1 is represented by the light state being on, and 0 otherwise. In the 802.15.7 standard, Manchester coding is used to ensure the period of each pulse is the same. This type of encoding, instead of having a digital 0 represented by a low signal and a digital 1 by a high signal, encodes each data bit as either low then high (1), or high then low (0), in equal time.

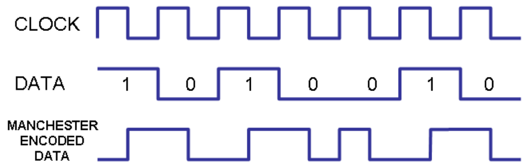

Figure 1: OOK modulation using Manchester coding.

Variable pulse position modulation (VPPM)
VPPM is similar to pulse position modulation (PPM), in which the data is encoded using the position of the pulse within a set time period. In this modulation scheme, light dimming is allowed as long as the period containing the pulse is long enough to allow different positions to be identified. As in the Manchester coding a positive pulse at the beginning of the period followed by a negative pulse at the end can represent a digital 0, and a 1 is represented by a negative pulse at the beginning followed by a positive one at the end.

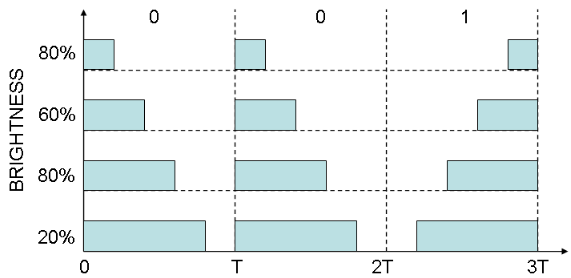

Figure 2: Variable pulse position modulation to support light dimming.

Colour shift keying (CSK)
This scheme allows the light intensity to be constant by encoding the information in the colour of the light. For implementing this kind of transmission the system must use RGB type LEDs.

3 System Overview

[In the next chapters I describe the overall characteristics and limitations that a VLC system should present according to my findings, identify the variables involved and the set of parameters necessary for it to work This chapter should provide a useful starting point for whoever plans to implement one in practice]
What does one need to make his own VLC system?

Every visible light communication system shares the same underlying structure. Communication needs to be established between two ends, one end that acts as a transmitter and the other that acts as a receiver. There are at least three levels of abstraction needed to transform a digital information into a signal that can be sent and received, and these are:

- logical layer
- control layer
- physical layer

Each of these levels present different characteristics that combined form the overall performance of the system.

In VLC, the transmitter end of the communication operates on a light emitter to vary a property of the light depending on the modulation scheme used (see 2.2), like brightness or colour. The receiver on the other end needs to be able to detect such variations in a measurable manner. This level can be described as the **physical layer** of the communication. At this level, the system's performance can be influenced by physical properties of the hardware, like maximum brightness of the light emitter, speed in switching ON/OFF state and warmup time, but also by other factors, like distance between the two ends, the medium of transmission, the angle of incidence of the light between transmitter and receiver and so on.

This variations need to be controlled and organised to carry specific information. A **control layer** is necessary in order to make the link between logical information and variation of physical property. This layer is of critical importance for the performance of any system, since it influences the frequency of the physical variations which results in the rates for transmission and reception. Factors that are to be considered in this layer are directly linked to the hardware components, namely any characteristic that influences the overall speed of transmission/reception, like clock rates of the micro controllers, speed of the transistors used and so on.

The outer most level of abstraction is the **logical layer**, where information is handled and manipulated at a software level. For a transmitter, this layer produces the instructions to pass on to the control layer in order to generate a signal given a specific information. This process can be seen as the logical encoding of the information, ready to be transferred and become physical encoding. At the opposite end of the transmission, the logical layer of a receiver is given data about physical variations measured by the sensor(s) used in the system, and needs to reconstruct and interpret such data back to the original information. Contrary to the previous layers, the performance of the logical layer doesn't rely much on the hardware and the physical characteristics of the components, but rather on the software techniques and algorithms that implement it. A well structured logical layer can even add more reliability to an otherwise uncertain medium of communication, as will be seen in later chapters.

do a graph of logical control and physical.

3.1 Experimental setup

In order to verify feasibility, investigate characteristics and test performance of general VLC systems, a prototype system has been developed. The system is composed of two main modules: a transmitter module, and a receiver. The transmitter module uses On Off Keying with Manchester Coding (see 2.2) to convey signals with light. In order to allow testing, this module takes some arbitrary input from a user, encapsulates the information and encodes it to produce variations in light intensity, through the control of a light emitter. The receiver side measures the light variations through the use of a photoresistor, or light sensor, reconstructing and interpreting the sensor data back into the original messages. Each module includes different components, listed in section 3.1.3. Fig. 3 shows an overview of the prototype architecture.

in the experimental setup, it's considered almost exclusively low power DC LED

3.1.1 Transmitter

Transmission starts from a terminal, where a client can input messages as strings. These are then encapsulated into Protocol Data Units (PDUs) and sent to a micro controller through a serial connection. In this case, the micro controller that was used for transmission is an Arduino board. The board encodes the received bytes into Manchester Code, and then controls the light signal by switching on and off the emitter accordingly. A light emitting diode is the furthest end of the transmitter module. For this part, multiple setups have been tried. The fastest emitter that has been tested is single low power LED connected to the board and powered directly by it, which allows very fast switching. A second setup has also been tested

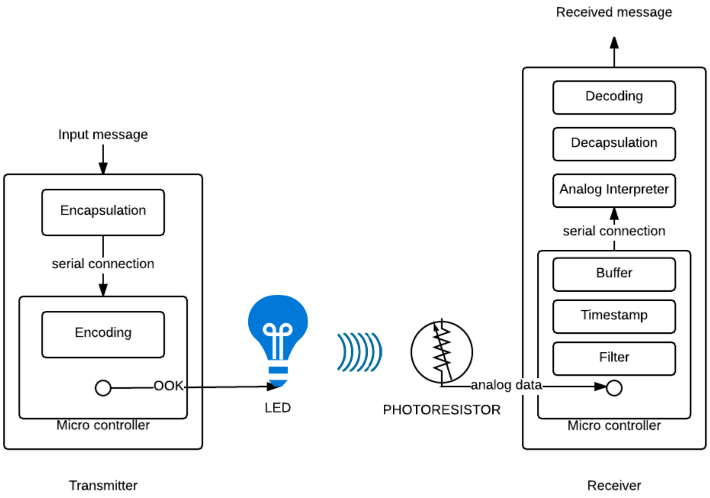

Figure 3: Prototype system.

with a commercial LED bulb powered by an external power source (mains electricity), and controlled by the board through a switcher.

3.1.2 Receiver

The signal is received by a photodiode read as analog input by a micro controller. On the board, values are software-filtered to reduce noise, and sent to the receiving terminal with a timestamp. The board and the terminal are connected through a serial connection. Some controllers could read directly analog input and perform the remaining processes to interpret the signal, with enough computational power.
For this application, the final terminal is a Raspberry Pi, since it has a good power/size ratio for the designed purpose and allows a simple prototyping process. The micro controller in between the sensor and the terminal is necessary to read and forward analog data, which is not possible directly from the Raspberry Pi. On the final terminal, the variations of sensory data are interpreted as sequences of digital 0s and 1s, finally decapsulated and decoded back into a message.

3.1.3 List of components

Each module of the prototype system is composed of different components, listed in the following.
Transmitter:

1. personal computer as main terminal for user input and information processing
2. Genuino Uno micro controller board, based on ATmega328P [2]
3. Light emitting diodes:
   (a) low power LED, 5V
   (b) commercial LED bulb, 230 V AC, 3 W, 240 lumen
4. Solid State Relay for Arduino, 5V activation, 240V load

Receiver:

1. Photoconductive Cell VT900
2. Genuino Yun micro controller board, based on ATmega32u4 [3]

3. Terminals:
    (a) personal computer for test of communication
    (b) Raspberry Pi 3 Model B to be applied on a mobile robot

3.1.4 Wiring

add wiring? diagram for small setup and most importantly setup with ssr and big bulb

4 Physical layer

This layer is all about physical properties of the hardware and the signal itself.

introduce, motivation, also explain setup, it's almost always with small LED unless otherwise stated

4.1 Warmup time

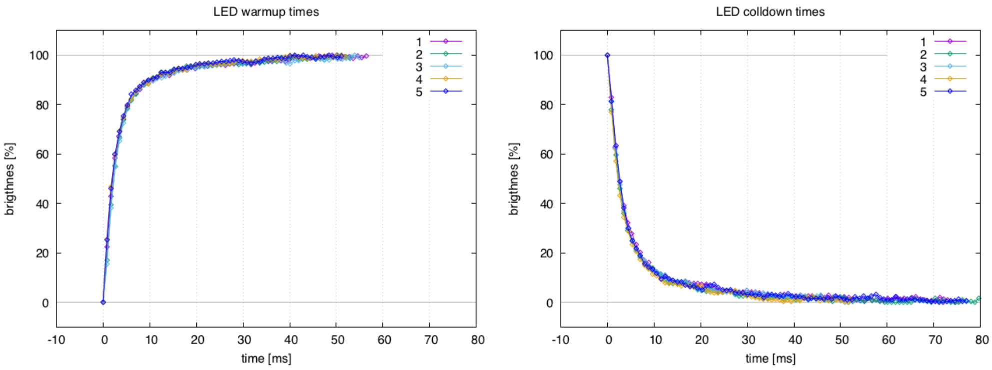

Figure 4: LED warmup and cooldown times.

stats for warmup

table with times to warmup from x to y, eg 0-0, from 0 to 10, 0-20 ,0-30, ..., 10-0 10-10; then discuss meaningful milestones

The speed of light transmission depends in primis on the speed at which the light itself can be turned on and off. Figure 4 shows the warmup times for the low power LED over multiple instances, meaning the time that it takes for turning the light completely on from completely off, and vice versa. These measurements are bound to the reception rate of the system, which will be discussed in section 5.2. Table 1 shows the times for the LED to switch between specific brightness levels. Each row represents the time to reach the level on each column, for example the first row represents the time to reach any brightness level starting from 0% brightness. The table works both ways, meaning it shows the time for the warmup as well as the time for the cool down of the LED. The last row shows how long it takes to reach any level from a completely ON state, meaning 100% of brightness.

4.2 Ambient light interference

produce statistics for different ambient light interferences

stats for big bulb

4.3 Distance

test for different distances, talk about brightness-distance (dark, no interference)

stats for different distances

	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
0%	-	0.3	0.66	1.08	1.52	2.12	2.82	3.78	5.38	10.4	50.66
10%	55.88	-	0.36	0.78	1.22	1.82	2.52	3.48	5.08	10.1	50.36
20%	61.86	5.98	-	0.42	0.86	1.46	2.16	3.12	4.72	9.74	50.0
30%	63.76	7.88	1.9	-	0.44	1.04	1.74	2.7	4.3	9.32	49.58
40%	64.9	9.02	3.04	1.14	-	0.6	1.3	2.26	3.86	8.88	49.14
50%	65.76	9.88	3.9	2.0	0.86	-	0.7	1.66	3.26	8.28	48.54
60%	66.4	10.52	4.54	2.64	1.5	0.64	-	0.96	2.56	7.58	47.84
70%	66.98	11.1	5.12	3.22	2.08	1.22	0.58	-	1.6	6.62	46.88
80%	67.48	11.6	5.62	3.72	2.58	1.72	1.08	0.5	-	5.02	45.28
90%	67.92	12.04	6.06	4.16	3.02	2.16	1.52	0.94	0.44	-	40.26
100%	68.28	12.4	6.42	4.52	3.38	2.52	1.88	1.3	0.8	0.36	-

Table 1: Warmup times of the LED, for specific levels of brightness.

4.4 Angle

test the light with different angles to the sensor (dark)

stats for different angles

4.5 Power source

talk about AC vs DC, discussion about power vs brightness vs distance vs speed

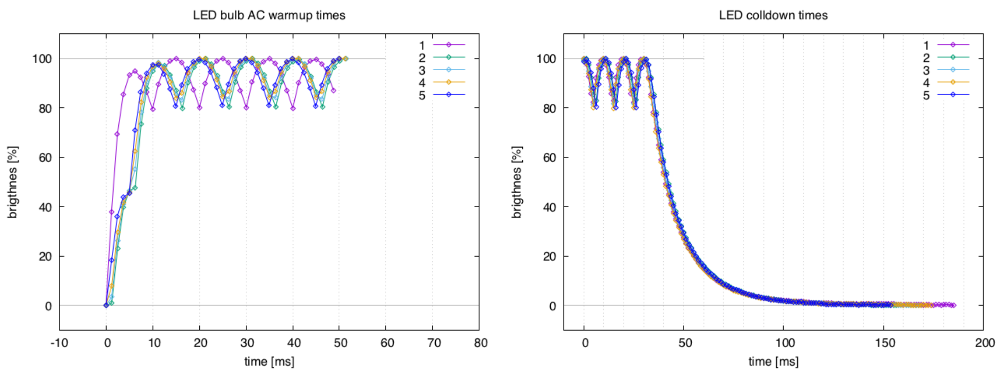

Figure 5: AC bulb warmup time, and phases of alternating current.

remove outlier in the picture of ac?

5 Control layer

[This layer serves as a link between the logical and physical layer.
In practice it takes logical input and tries to control hardware to produce physical variations in response.
Switching speed allowed by the control layer: boards but not only, relays and transistors too.
Speed of reception also depending on the board.
Frequency of encoding as a result.
Also speed of communication from a controller to a terminal like in the prototype (serial)?]

This layer is responsible of connecting the physical and the logical layers of the system, in a structured way. For a transmitter this translates to the process of receiving a logical message, and control the physical components of the system accordingly. For the receiver the control layer is responsible of reading the values from the sensory layer and forward them to the logical layer, perform filtering and other minor operations. In the prototyped system, this layer is physically represented by micro controller boards connected with a terminal through a serial connection on one side, and with the components of the physical layer on the other. This layer is crucial for establishing performance of the system in terms of **rates**, as in how many values can be read from a sensor per time unit for reception, and how fast can a control signal be sent to control the light emitter for transmission.

small graph serial -¿ control -¿ physical

5.1 Serial connection to a terminal

find a place for this

One aspect to consider in implementing this passage is that the serial buffer has a limited size of 64 bytes in most Arduino boards, so it's important to keep the serial communication as light as possible to avoid overflow. In most cases, it's possible to encode a char into one single 8-bits byte, but not all the programming languages implement this automatically. The system described in this paper was implemented using Python 2.7, which uses dynamic types and therefore doesn't have a specific type for *chars*. In Python, strings have an overhead of 37 bytes, plus one byte for every character in the string, this would result in a very heavy serial transmission for just a few characters. Characters need to be converted in single bytes before sending them.

information about clock rates of the boards

5.2 Reception rates

Reception happens as fast as the micro controller allows, meaning that the control process doesn't force a specific speed on the analog input coming from the sensor, but rather the rate of reception is bound to the clock rate of the processor in the control layer and the speed of the physical sensor. The average reception rate is of about one value every 0.833 ms, about 1200 Hz (or bits per second). This value naturally becomes the upper bound for the transmission rate, since if signals were sent faster, they wouldn't be received in time. The control layer also performs the two operations of appending a timestamp to each value once received before sending it to the next process, and filtering each value as an average of its direct predecessor and itself. With the use of timestamps it's fairly simple to measure the reception rate before the serial connection. This rate is not constant, but has a standard deviation of about 0,0184 ms.

rate of serial connection arduino to computer?

5.3 Transmission rates

[maximum rates possible for the control layer, and rates adjusted to physical layer limitations and reception limitations]
Transmission takes as input a message that needs to be converted into Manchester code (see 2.2). Just like pure binary, Manchester encoding only has two symbols, 1s and 0s. A symbol, in telecommunication, represents the smallest amount of data that can be sent in form of an analog signal. The symbol rate (symbols per time unit) is measured in baud. Each pair of symbols in Manchester represent a single binary bit. Given the limitations for the reception side, it's in practice very difficult to measure transmission rates above 1000 Hz in the prototype system. From the measurements of warmup times of the LED presented in section 4, it's possible to see that in 1 ms the LED can produce an increase of at most 15% of brightness in the best case. The control layer is directly responsible for the transmission rate, therefore it needs to guarantee a rate that allows a reliable reception. These previous limitations would suggest that a maximum acceptable rate is of 1000 Hz, with a potential difference between an average 1 and an average 0 of at most 15% of brightness, and a reception rate that is slightly higher. Would this rate produce acceptable readings on the receiver side?

how much decrease in 1 ms?

Experimentally it was found that sending signals at 1ms intervals produces a degree of uncertainty of about 30%, while a 2 ms interval performs much better at 0%. More on the experiments and results in section 5.4. According to these results, each single symbol is forced to last for 2 ms before the transmitter board can send the next. The transmission rate would therefore be at best of 500 bauds in theory, in the case where the activity of the micro controller pays no role. This value was experimentally measured to be slightly smaller, at around 450 bauds.

5.4 Experiments and Results

[Experimental setup, experiments, performance and the the results I achieved in the prototype. also talk about limitations for the solid state relay and light bulb]
To find the perfect transmission rate experimentally, the prototype has been set up to transmit a sequence of alternating 1s and 0s of known length multiple times. For this experiment, the distance and angle from the sensor to the light source have been minimised to obtain the best reception possible. The experiment aims to measure the number of local maximums and local minimums in the transmission, that would clearly define the number of 0 and 1 signals received. Also, the brightness levels have been measured for such local peaks.
The fist transmission rate to be tested is a 1000 bps rate, with an interval of 1 ms between subsequent signals. Over 30% of the signals haven't been received.
A second rate of about 500 bps has also been tested to compare with the first one. This time the interval between two subsequent signals is of 2 ms. Table 2 shows the results of various measurements with the two rates, while fig. 6 shows overlapped samples of the transmissions with the two different setups.
As can be seen in the figure, during transmission the brightness achieved from the LED used for testing is not at 100%, but transmission is still clearly recognised at a level of about 60%. In the figure, the starting and ending points of the transmission are at the minimum and maximum brightness level of the LED, to the far left and far right respectively.
The transmission is always the same over all the instances and in both setups. Reception with the first setup is almost double as fast as the one in the second setup, which is consistent with the rate difference. Another difference that is clearly noticeable is that the difference between 1s and 0s also doubles, making the second setup more reliable in case of ambient light interference. These results suggest that the lower the rate, the more reliable the communication. However, the rate of 2 ms per signal has been chosen to be final transmission rate in the prototype system with the low power LED, being the fastest reliable rate. With a system that uses a different light emitter, or designed to be used at longer distances, different experiments would be advised.

prove that there is more reliability with ambient interference

add more setups, 3 and 4 ms too

rate interval	missing 1s	missing 0s	average 1 brightness	average 0 brightness
1 ms	34.52%	30.62%	66.75% +- 6.10	55.78% +- 6.72
2 ms	0.00%	0.00%	70.22% +- 7.06	50.18% +- 5.79
3 ms	0.00%	0.00%	76.72% +- 6.18	43.03% +- 5.20
4 ms	0.00%	0.00%	82.77% +- 4.00	35.54% +- 3.60

Table 2: Rates results compared.

solid state relay limitations

6 Logical layer

This layer represents the pure software part.
Logical encoding may try to overcome some of the previous limitations.
Standard for VLC is encoding in Manchester Code, also some kind of protocol could attempt to do bit recovery.

redo this figure with a more recent transmission

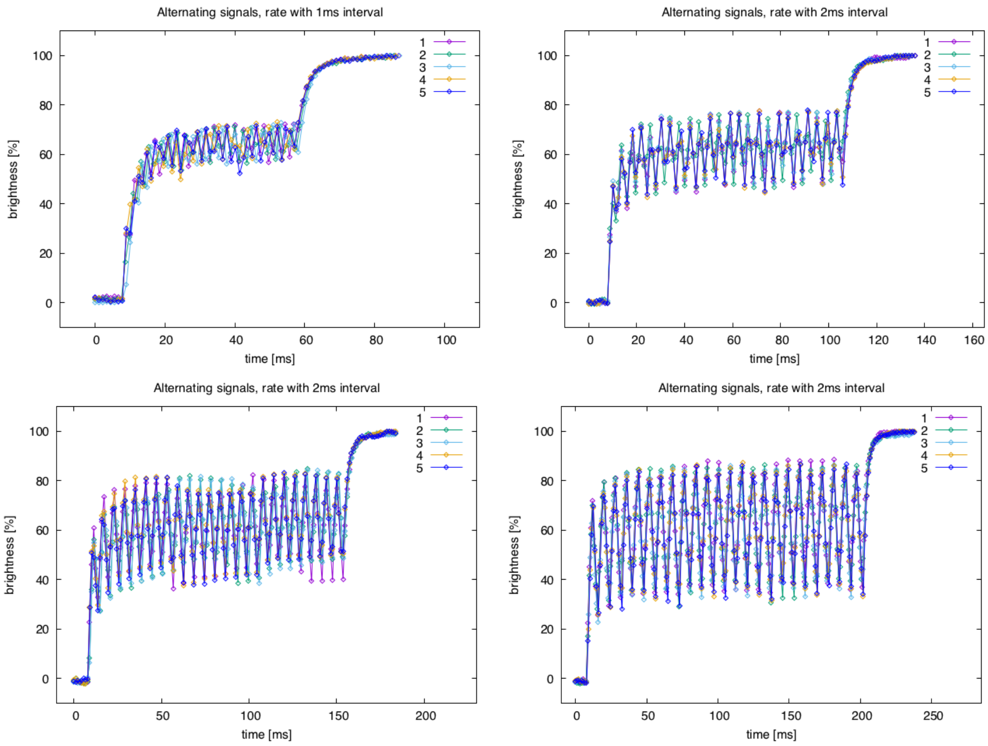

Figure 6: Test of rate with 1 ms, 2 ms, 3 ms and 4 ms intervals (left to right, top to bottom).

6.1 Reading the signal

[shape, noise, machine learning difficulties, final resolution]
A key part of this project is to convert analog sensor data representing light variations into binary code. Fig. 8 shows a sample of transmission. The green line represents the original values from the light sensor, while the purple line is a reconstruction of what the digital transmission would look like. Manchester encoding allows transmission to only have two kinds of peaks, either representing single bits, like a 1 or a 0, or two bits of the same kind. Having a longer sequence of identical symbols would be impossible with this encoding.

Multiple attempts have been made to design an algorithm that would interpret this sequences of values into binary data.
The first version of it made use of Machine Learning techniques to train a model with training data, "teaching" a classifier what kinds of data and results one would expect in this application. The data was taken in different circumstances of ambient light interference, with different messages sent and at different times. Multiple classifiers were tested, with a maximum success rate of about 75%. The classifiers included NaiveBayes, RandomForest, AdaBoost and ArtificialNeuralNetworks, all with comparable results.
There were major difficulties with this approach. For starters, there were problems identifying the labels to train these models, namely the classes or categories where the data would belong. Initially, the four categories were "1", "0", "11", and "00". The basic idea is that given a sequence of values it should fall in one of these four classes. The problem was though that single bits and double bits have different sizes in amount of values that form them. This is also called the number of features.
Most standard modules for machine learning require the same number of features for all the labels.

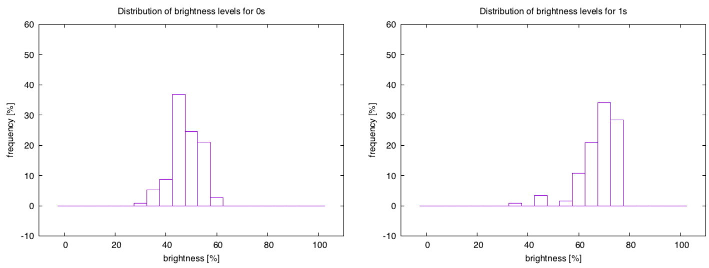

Figure 7: Distributions of brightness levels during transmission (2 ms interval), brightness of binary 0s to the left, 1s to the right.

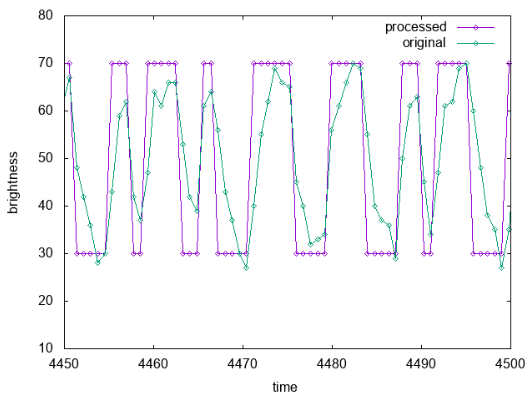

Figure 8: Example of the results of interpreting digital information from analog input.

In order to get around this rule, the labels became "10", "01", "11", and "00". Eventually, also these labels were produced by different numbers of features. Other combinations were tried, and eventually all leading to the same result. One way to get around this problem was to have a fixed number of features, but big enough to include all the possibilities. Samples smaller than this maximum amount (virtually all of the samples) would be filled up with blank values until they reached the intended size.
Another way would be to train a classifier for each specific number of features that may occur. In this case, the amount of classifiers needed would be around 4, for number of features ranging between 7 and 10. The downside of this approach is that each classifier becomes very specific to a restricted set of the training data, potentially loosing track completely of certain labels.
The final setup for machine learning prediction used the labels "101", "100", "011", "110", "001", "11", "00", "10", "01", scaling of the features and larger sample size to be filled up with blanks. This produced overall the best results for the machine learning approach, but never above 80% success rate.
A second downside to this approach would be that when receiving values from the sensor in real time, one would need to keep a moving window of the size of the number of features and try to obtain a prediction from the classifier each time. This might likely slow down the reading process.

These poor results ultimately led to a change of approach for reading the data. Just looking at fig. **??** it's pretty straightforward to see that the high peaks represent 1s and the low peaks represent 0s. By setting some rules, a custom classifier could predict a result without the need to be trained with sample data. In particular, the classifier analyses sequences of values, trying to find sequences that are either monotonically increasing or monotonically decreasing. Also a noise factor has to be taken into account, in this way small enough variations are not considered. As mentioned before, there are only two kinds of peaks in the sequences, either single or double peaks. Therefore, it's of critical importance to be able to distinguish the two cases. One criteria that was originally deployed for this was to count the number of values that progress in the same direction, either up or down. This was later changed to the more precise criteria of duration of such sequences, and this is the reason why for each value a timestamp is included representing the time of reception for that specific value.

stats about reception rate, std deviation.

With this in mind, the algorithm has been developed to very simply count the duration of either monotonically increasing or decreasing sequences of values, and produce a prediction when the direction changes, based on such duration. This technique performs particularly well in this application, producing up to 100% success rate in certain cases. Runtime wise, the algorithm takes constant time for each value that that is received, and may or may not produce a result. A simplified version of the algorithm can be found in fig. 9.

6.2 Protocol

As can be seen the rates of transmission and reception in this prototype system are not very high, if compared to average wireless transmission speeds, which are measured in Mbps. Transmission in visible light is also exposed to a high degree of uncertainty, depending on parameters like distance, maximum brightness, interference, noise, and so on. In this prototype, communication is only established in one direction, which additionally increases the treat of getting wrong results. In order to compensate all this aspects, an effort to strengthen the success rate has been made by structuring the communication into a very basic protocol. Each PDU is included between two single bytes: STX, that indicates the start of a message, and a byte ETX which indicates its end. These two are bytes 0x02 and 0x03 respectively. To avoid that the reader assumes an ETX byte wrongfully while reading the message, a length byte LEN is also included in the header of the PDU, to specify the length of the message in bytes. Additionally, each PDU is restrained to have a maximum size, and longer messages need to be split into more PDUs.

```
def feed(self, time, value):
        pred = None

        # staying
        if abs(value - self.prev) <= self.epsilon:
                pass

        #going down
        elif value <= self.prev:
                if self.direction: # up
                        pred = self._predict(time)

        # going up
        elif value > self.prev:
                if not self.direction: # down
                        pred = self._predict(time)

        self.prev = value
        return pred

def _predict(self, time):
        m = self.direction
        delta = time - self.seqstart
        pred = '-'

        if delta >= self._doubletime:
                pred = '11' if m else '00'
        else:
                pred = '1' if m else '0'

        self.direction = not self.direction
        self.seqstart = time
        return pred
```

Figure 9: Simplified algorithm for signal interpretation (Python 2.7).

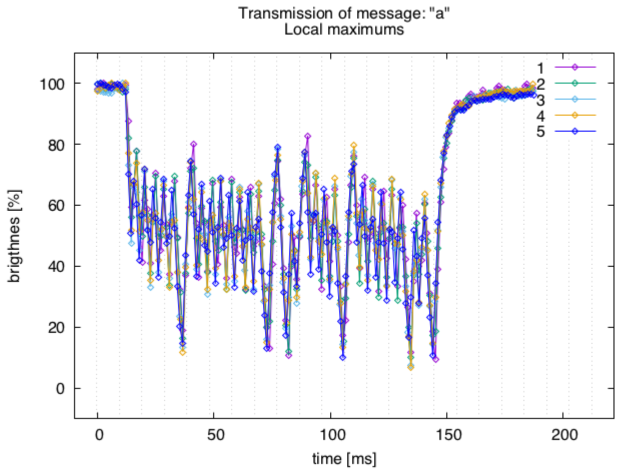

Figure 10: Example of transmission of message ”a”, the transmission include STX, ETX, and LEN bytes.

6.2.1 Additional fields

Increasing the functionalities or complexity of the system, additional parameters in the PDU might result useful if not necessary. Imagining a scenario with multiple receivers of messages broadcasted through light in a unilateral way, a receiver field could be added to the PDU. Also, it would be wise to perform encryption on the PDUs, in a way that only the intended receivers would be able to read the message directed to them. Establishing 2-way communication would also require more steps. 2-way communication would be a very good way to ensure correctness of the transmission. For this, at least 2 more parameters seem necessary: a sequence number to identify the packet, and a checksum to check correctness of the received message. A header for visible light communication doesn't require many parameters since it's likely to be the very last end of a communication process. Since it's mainly short distanced, it doesn't need any routing parameters as it would in big networks.

6.3 Results

[Experimental setup, experiments, performance and the the results I achieved in the prototype.]
how many errors per total bits sent?
correctness rate for different distances
different bulbs might also change
compare with other forms of communication (here or somewhere else?)
Also compare the use of protocol vs not protocol

7 Evaluation and Results

[How did it go as an application to robotics? does it make sense? performance? tests? comparisons to other communication forms?]

7.1 Experimental setup

7.2 Experiments

7.3 Performance

7.4 Discussion

8 Discussion

[This is some discussion. Here I can link my results to the vision that was presented in the introduction, are the results good enough for some of those applications? Are those scenarios plausible according to my findings? what could be done with my system, more than what has been done? How could one take my work and use it for other purposes, or expand it? how to make it faster?]

9 Conclusion

[This is the conclusion. Here I sum up all the crucial points of my work, from the vision, to my results, to some of the major points in the discussion.]

References

[1] R. Corsini P. Choudhury E. Ciaramella A.M. Khalid, G. Cossu. 1-gb/s transmission over a phosphorescent white led by using rate-adaptive discrete multitone modulation. *IEEE Photonics Journal 4, 14651473*, 2012.

[2] Arduino. Arduino uno overview. `https://www.arduino.cc/en/main/arduinoBoardUno`.

[3] Arduino. Arduino yun overview. `https://www.arduino.cc/en/Main/ArduinoBoardYun`.

[4] IEEE Standards Association. Ieee 802.15 wpan task group 7 (tg7) visible light communication. `http://www.ieee802.org/15/pub/TG7.html`.

[5] US Federal Communication Commission. `https://www.fcc.gov/general/spectrum-crunch`.

[6] S. Rajbhandari J.J.D. McKendry S. Videv E. Gu M. Haji S. Watson A. Kelly G. Faulkner M.D. Dawson H. Haas D. OBrien D. Tsonev, H. Chun. A 3-gb/s single-led ofdm-based wireless vlc link using a gallium nitride led. *Photonics Technology Letter 99(99)*, 2014.

[7] P. Choudhury R. Corsini E. Ciaramella G. Cossu, A.M. Khalid. 3.4 gbit/s visible optical wireless transmission based on rgb led. *Optics Express 20, B501B506*, 2012.

[8] Harald Haas. Ted talk online by harald haas on wireless data from every light bulb. `http://www.ted.com/talks/harald_haas_wireless_data_from_every_light_bulb`.

[9] Harald Haas. High-speed wireless networking using visible light. 2013.

[10] S. Nerreter K. D. Langer J. W. Walewski J. Vucic, C. Kottke. 513 mbit/s visible light communications link based on dmt modulation of a white led. *J. Lightwave Technol. 28 (24), pp. 35123518*, 2010.

[11] Gordon Povey. `http://visiblelightcomm.com/an-ieee-standard-for-visible-light-communications/`.