

Implementing a ZigBee-Based Smart Lighting System

Arsham Asgharian
Philip Hatziantoniou



LUND
UNIVERSITY

Department of Automatic Control

MSc Thesis
ISRN LUTFD2/TFRT--5955--SE
ISSN 0280-5316

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden

© 2014 by Arsham Asgharian & Philip Hatziantoniou. All rights reserved.
Printed in Sweden by Media-Tryck
Lund 2014

Abstract

All industrial facilities and large warehouses require some sort of lighting solution. They have ongoing activities with workers and trucks moving around at various speed, and to avoid accidents these lighting solutions are often very bright and energy consuming. The most common solution is fluorescent light. These lights are energy consuming and not dimmable. Turning these off by night or when the facilities are empty makes their lifespan short, so they are usually always on. This ends up being very expensive and bad for the environment. By using LED as the main lighting solution it is possible to decrease the energy consumption to save money and help the environment. To decrease the figures even further there are smart LED lighting solutions. These solutions will detect ongoing activities and brightness levels in the facilities and automatically adapt the lighting needed in real time. There are many different solutions to smart lighting. Usually there are motion sensors and/or light sensors. These sensors will recognize any movement in the facility and increase the effect of lamps if more light is needed.

The purpose of this master thesis is to add another sensor to give even more accuracy in light to facilities. Facilities mainly use trucks or other moving vehicles. This extra sensor is provided by attaching a ZigBee module to the lamps. The modules communicate with the moving vehicles and estimate their position to supply light wherever is needed. This provides an extra dimension to the smart light solution. Furthermore, using ZigBee one can detect a moving object close to the facility but not yet in the facility. This means that if a vehicle is moving towards a space that is not lit up, the LED will detect it and light up the space before the vehicle has entered, so the vehicle has appropriate amount of light when it arrives.

Acknowledgements

First of all we would like to thank Nader Salessi for all the help and for giving us the chance to work on our degree project at LED-Era. It was a great experience for both of us to do our master thesis abroad and we learned a lot. In addition, we would like to thank João Alcantara for taking his time to help us, especially with the coding.

Thanks to Karl-Erik Årzén for being our examiner and for helping us time after time when help was needed. Last but not least, we would like to thank all of the family members in the US who took care of us and helped us while we were there.

Contents

1. Introduction.....	9
1.1 Background.....	9
1.2 Goal.....	10
1.3 LED Era, Inc.	11
1.4 Individual Contributions	11
1.5 Outline	11
2. Background on Energy and Lighting	12
2.1 Energy and Lighting Consumption	12
2.2 Light-Emitting Diodes and Smart Lighting	12
2.3 LED versus Fluorescence Lighting	13
2.4 Related work	14
3. System Overview	17
3.1 The Lamp.....	17
3.2 Sensors	20
4. Radio Protocols.....	23
4.1 ZigBee Radio	23
5. Implementation	25
5.1 ZigBee Implementation	25
6. Simulation.....	30
6.1 TrueTime	30
6.2 TrueTime Model	35

7. Result	42
7.1 Lamp	42
7.2 Simulation	43
8. Summary and Conclusion	49
8.1 Lamp	49
8.2 Simulation	50
8.3 Problems	51
8.4 Conclusions	52
9. Bibliography	53
Appendix 1	56
Appendix 2	57
Appendix 3	58
Appendix 4	59
Appendix 5	60

1. Introduction

This Master Thesis investigates smart LED lighting systems for indoor and outdoor use with ZigBee being the main controller of the system. The project is a cooperation between LED Era, Inc., Irvine California, the Department of Automatic Control, Lund University, and the Department of Industrial Electrical Engineering and Automation, Lund University. The major part of the work was performed at LED Era, Irvine, California.

1.1 Background

We are living in a fast growing world, with a population rate higher than ever and demands on resources such as food and consumer products are sky high. To be able to meet these demands new factories and warehouses are built every year. As the industrial sector is growing so is the energy consumption. As a matter of fact the energy consumption of the industrial sector is growing with a rate of 1.4% every year (U.S. Energy Information Administration, 2014). This rate will probably change to the worse with the projected population and economic growth in developing countries.

Today, compared to the consumer sector the industrial sector consumes half of the world's total delivered energy (U.S. Energy Information Administration, 2014). The energy consumed by factories and warehouses ranges across different purposes, one of these purposes is lighting. Around 20 percent of the energy consumption in warehouses and factories is due to electricity consumption in lighting systems.

With a growing population other demands are high as well. In 2010 the number of cars exceeded one billion. That is almost one car for every nine person in the world and with increasing cars on the road there is an increase in numbers

of parking lots as well. In the US alone there are more than 100 million commercial parking spaces. Every parking space use some kind of lighting solutions, and most of them use fluorescent tubes. (International Parking Institute, 2014)

Those are two examples of places where lighting is needed at so massive scale that it impacts our environment in a negative way.

“I predict that very shortly the old-fashioned incandescent lamp, having a filament heated to brightness by the passage of electric current through it, will entirely disappear” – Nikola Tesla (Tesla, 1930). Though this prediction of Nikola Tesla almost came true, we have a long way to go before we find the perfect solution of lighting.

In today’s environment strive towards more energy efficient solutions is important. In large scale facilities such as schools, warehouses and parking garages etc., the energy consumption of lighting has a negative impact on the environment. One of the most common light sources in these types of facilities is fluorescent tubes. These are normally more efficient than the regular light bulbs but still are not the most optimal solution. They contain mercury, which is a hazardous element and is dangerous for health and to the environment. The European 20/20/20 directives imply a reduction of emissions from greenhouse gas by at least 20 percent, a 20 percent increase in energy efficiency and to have at least 20 percent of electricity from renewable energy sources. (Energimyndigheten, 2011)

With the directive in mind it is a good time to start making the lighting as efficient as possible. In Sweden, the consumption of lighting amounts to 14 TWh per year. There the industry and the public sector account for the majority of the consumption – about 10TWh. (Belysningsbranschen, 2013)

1.2 Goal

It is common that a warehouse with an operation of 24 hours a day, 7 days a week, does not have a smart lighting system. This means that the lights are turned on all the time and drains a lot of energy that could have been saved by implementing a smart system. An issue that can occur with light fixtures only controlled by a motion detector is for example when forklifts are used. A vehicle moving at various speeds in a facility with a lot of obstacles and narrow aisles can be hard to detect in time. This can result in accidents and be a safety hazard. The forklift will

need proper lighting prior to entering an area for optimal maneuvering. One solution is to try to detect the forklift before it approaches and light up the area in advance for a proper lightning condition. (Belysningsbranschen, 2012)

To do so another sensor has to be implemented. A ZigBee module at every light source and a ZigBee tag attached to every vehicle was a solution that met the criteria. The goal of this master thesis was to implement this system to allow all LEDs to communicate with each other and with the vehicles.

1.3 LED Era, Inc.

This thesis has been performed at LED-Era, Inc., which was founded by Nader Salessi in Irvine, California. It is a company in the lighting sector with a specialization in LED technology, also known as Solid-State Lighting (SSL). They make energy efficient and environmentally friendly lighting for both interior and exterior usage, with a range of different products which are suitable for commercial and industrial applications.

1.4 Individual Contributions

The major part of the work was done together. Arsham has had more focus on the lamp part and he has taken care of the automatic control part. Philip has had more focus on the simulation part and the energy aspects of the project.

1.5 Outline

Chapter 2 gives a brief introduction to energy consumption and smart lighting solutions. There are also comparisons between different lighting solutions. In Chapter 3 a system overview of the different components used in the project is given. Chapter 4 describes the theory of radio protocols in the ZigBee and how mesh networking works. The chapter also describes the fundamentals of CSMA/CA. Chapter 5 is a detailed walkthrough of how the implementation of the lamp and ZigBee was done. The simulation in TrueTime and Matlab is described in Chapter 6. Chapter 7 shows the results obtained during the project. Chapter 8 is a summary of the project and conclusions that were made during and after the project.

2. Background on Energy and Lighting

This chapter gives a brief introduction to the background on energy and lighting. First, there is a brief overview of how a smart LED system works and how it differs from today's most common lighting solutions. Then, an overview of other related work from companies and universities who have tackled the same issues, is given.

2.1 Energy and Lighting Consumption

The energy consumption for lighting in the US for the commercial sector which for example includes parking and highway lighting constitutes about 274 TWh. This is around 21 percent of the commercial sector's total electricity usage in 2012. For manufacturing facilities 52 TWh were consumed in 2010 (U.S. Energy Information Administration, 2014).

In Sweden the lighting consumption is around 14 TWh and the public and industry sectors account for about 10 TWh (Belysningsbranschen, 2013).

2.2 Light-Emitting Diodes and Smart Lighting

The main principle is to only light up and control lighting where it is needed. There is no reason having the lights fully turned on if no one is there and it is vital to have the correct light where it is needed. The two main categories for smart lighting are occupancy and daylight sensing.

Occupancy sensing

Within this category there are two methods, the first one is presence control/damping which is the most basic one. One or more motion detectors are used to turn on the lights if there has been any movement within the room and turn the lights off after a certain time if no movement has been registered. If damping is used, the lights will not be turned off completely, instead they will be set to a low value and when presence is detected the light will be turned up again.

The other method is absence control/damping, which also uses motion detectors. The difference between presence- and absence control/damping is that the lights have to be turned on manually and it turns off on a preset time if no movement has been detected during that time. The damping works basically the same way, except the lights will turn down to a low value instead of being turned off.

When using occupancy sensing it is important to place the detectors in an area where it gets 360 degrees coverage for it to work as well as possible. Any obstacles that can block the detector can reduce its ability. (Belysningsbranschen, 2012).

Daylight sensing

With daylight sensing there is a preset value of brightness that will remain constant during the whole time the lamp is turned on. The brightness level will remain the same regardless of light source. If the outside light is enough the light fixtures will be turned off and if the outside light is not enough the light fixture will compensate the light until the preset level is reached.

Some problems using daylight sensing can occur when the weather is unstable. Changes in weather will cause more frequent changes in the lights. If this happens the lights will take damage which could lead to a reduced lifespan and it will also disturb the people in that area. A way to prevent this is to have fewer intervals, for example 0-, 30-, 60-, 90 percent intervals to reduce the risk of frequent changes in the brightness level. (Belysningsbranschen, 2012).

2.3 LED versus Fluorescence Lighting

Most large scale facilities use fluorescent tubes as their main light source. There are, however, many reasons why switching to LEDs can be a more viable option. Most of the fluorescent tubes are not dimmable, that means they can only be in two different states, either on or off. Every time the light comes on it loses a little of its lifespan and that fact causes them to be on most of the time. The

consequence of leaving the lights turned on all the time is that they consume more energy and burns out faster. Whenever a tube has to be changed, the facility often needs to close that section or aisle. It usually requires two or more persons to have it changed since the lighting often is attached high up in the ceiling. This could lead to problems with daily operations and cause setbacks in production.

The fluorescents contain mercury, which is hazardous for health and the environment if it is not recycled properly. LEDs on the other hand do not contain mercury and with a lifespan of roughly seven or eight times greater compared to the fluorescent tubes, reduces the need to change the lighting. They are also dimmable which contributes with another dimension of energy saving options. Some other advantages are that they do not emit UV-, IR-radiation or heat. LEDs have their problems as well, for instance they are temperature sensitive, i.e., they work well in cold environments but can quickly drop in illumination when the temperature gets higher. They are also highly dependent on the right operation- and ambient temperatures (Laike & Küller, 2012).

There are no international LED standards, which mean that the quality of the LEDs can vary. In (Schultz, 2012), it was shown how LEDs fulfil the requirement on electromagnetic compatibility (EMC). The investigation was done by 18 different electrical safety authorities in Europe; the result was that only 29 out of 168 LEDs fulfilled the requirements. A common reason why they did not meet the requirements needed, were that many of the LEDSs lacked CE-labeling or type designation.

Even though the transition to LEDs might have a high initial cost, the low energy consumption and the fact that it has a longer lifespan than other options contributes to a more profitable solution in the long run (Ljuskultur & Belysningsbranschen, 2011).

2.4 Related work

Research on ZigBee controlled lighting is increasing. Even some appliances have started to show up on the market.

In (Hung, Bai, & Tsai, 2011) it is described how only a few products combine the dimming circuits with the lighting control systems remote circuit. They describe their model using a ZigBee wireless module that sends the control and senses the signal between the units. An advantage is that each ZigBee sensor

has a unique ID for sending information between the control systems. It can also increase the number of sensor nodes to 255 under certain protocols.

As a remote control the recommendation is the use of a smart phone or a personal digital assistant (PDA) just because it can connect to the Internet. This makes it easier for the lighting controller to organize the control group. For the LEDs they divided them into two groups, one is controlled using PWM and the other uses digitizing halftone brightness control.

The result was achieved by implementing the system with four different lighting types: fluorescent, incandescent, LED-PWM and LED halftone. By comparing different aspects such as dimming methods, total power consumption with and without sensor and control, the result for home lighting was that LED halftone with the control system was the best option.

Another company that has started designing and selling smart lighting solutions with ZigBee networks is Marvell Technology Group Ltd. The company has developed an end-to-end lighting system for both commercial and residential lighting applications. A rough schedule how their system works can be seen in Figure 2.1. The system includes two-way communication and has support for cloud-based applications or via a smart phone app to be able to remotely regulate the lighting (Marvell Technology Group Ltd, 2014).

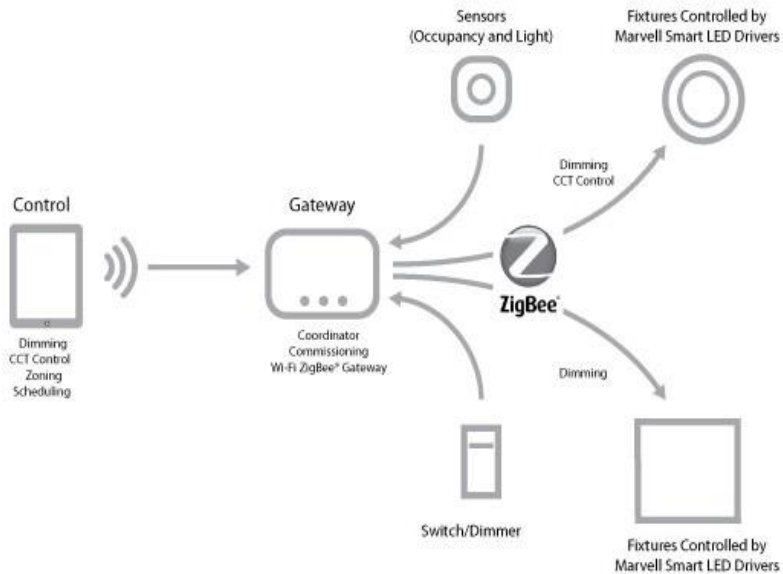


Figure 2.1. A brief overview of the Marvell system (Marvell Technology Group Ltd, 2014)

Also Philips has a product on the market which is called Hue (Philips, 2013). It works similar to the Marvell product, by having a control system called “the bridge” which has support for ZigBee and Wi-Fi and can connect up to 50 light bulbs per bridge. The lamps can be remotely controlled by an app on a smart phone.

3. System Overview

This chapter gives an overview of the components used to create a smart LED system. It gives the reader a brief explanation of how the components work and how they are used in the system.

3.1 The Lamp

The system is structured into a few components that together build up a smart controlled LED-lighting. The LED structure has an aluminum housing, as seen in Figure 3.1, that contains LEDs of groups of two by seven, see Figure 3.2. Every group of LEDs is controlled separately which gives the possibility of a more adjustable brightness level. Every lamp contains at least four groups of LEDs. The microprocessor/microcontroller (MCU), see Figure 3.5, is the heart of the system, It controls all the LEDs and is located in the power supply in Figure 3.4. It is a 32-bit Cortex-M3 ARM processor from Texas Instruments and has the model number LM3S9B92. The MCU provide 256 KB of on-chip Flash memory. In other words it has 256 KB blocks, each divided in a 1 KB block. They can all be individually erased by changing each blocks entire contents to 1s. To protect the code in the memory these blocks are further paired into a set of 2 KB blocks. These sets can be individually set to different protection levels by marking them as read-only or execute-only. By marking some blocks as read-only, one does not have to worry about jeopardizing important parts of the memory. The MCU operates at a frequency of 80 MHz (Texas Instruments, 2011)

Furthermore, there is a ZigBee module, a light sensor, and a motion detector, all communicating with the microprocessor. This combination of different controls is a way to keep the lighting as precise as possible and to conserve energy.



Figure 3.1. The LED-lamp structure with aluminum housing.

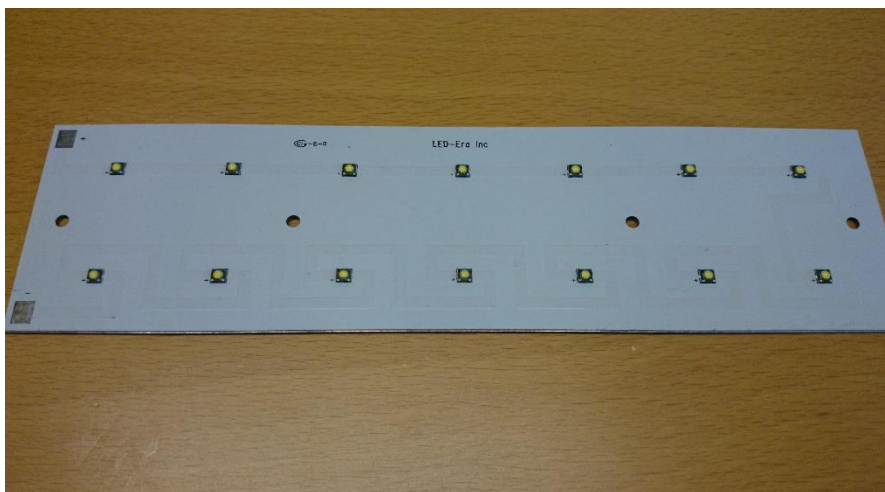


Figure 3.2. Group of LEDs



Figure 3.3. Power supply containing the MCU

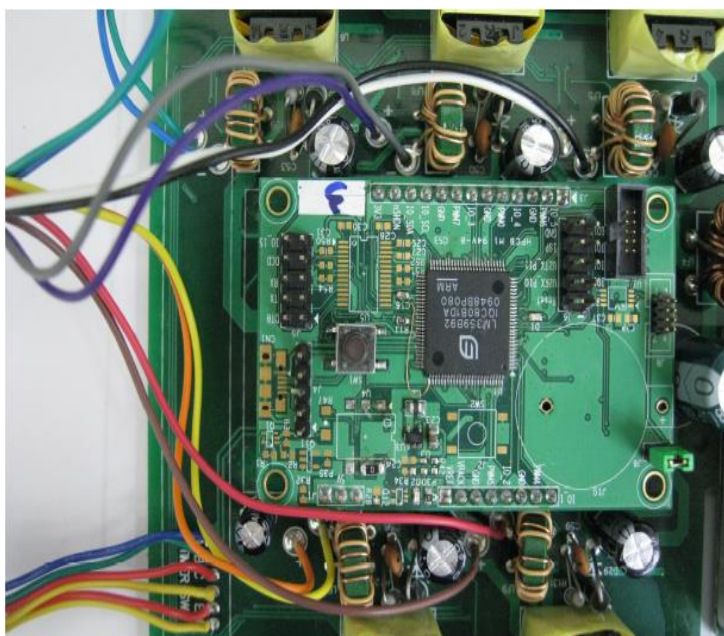


Figure 3.4. The MCU is located inside the power supply. The schematics can be seen in Appendix 4.

3.2 Sensors

Light Sensor

The light sensor used is a simple LED, as seen in Figure 3.5, which is in a reverse-bias mode, also called a photodiode. This means that, contrary to the LED, the photodiode converts exposed light to either a voltage or current. When a photon strikes an atom in the diode the atom releases a negative charged electron and a positive charged hole. This free electron can either combine with the hole or remain free and be pulled away by an external field. For the latter case the current through the diode will change and photocurrent is produced.

The objective of the photodiode is to sense the light in that area; that is, if the area is bright enough through external light sources, the lights can be reduced to a lower setting to save energy.

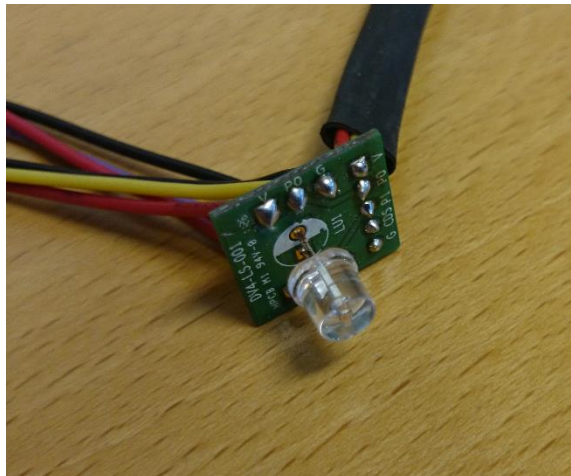


Figure 3.5. Light sensor LED

Motion Detector

The motion detector seen in Figure 3.6 detects moving objects and sends an interrupt to the microprocessor so the LEDs can increase in light if motion is present. The motion detector is a passive infrared sensor (PIR sensor). Its purpose is to register heat emitted by objects. The infrared radiation (heat) is detected in the center of the sensor by a crystalline material. Since all objects above absolute zero emits heat energy the PIR sensor has to detect changes in radiation when an object enter its field of view. This will lead to a change in voltage generated which is measured by an amplifier. The change in voltage will then be registered in the MCU. (Mathas, 2012)



Figure 3.6. Motion detector

ZigBee

To make the LED communicate and detect objects the ZigBee module is used. ZigBee is an open global standard for wireless communication and is based on the IEEE 802.15 standard. It uses unlicensed bands such as the 2.4 GHz, 868 MHz and 900-928 MHz bands (ZigBee Alliance, 2014). Some of the advantages of using ZigBee instead of other wireless network standards such as Bluetooth or Wi-Fi are the low cost, low latency and low energy consumption. Another advantage

is that the ZigBee has support for multiple network topologies like ad-hoc, peer to peer, star and mesh (Synapse, 2010). The ability to support mesh networking is a significant feature for communication between the LEDs and objects. For a fully connected mesh network, each node is interconnected with every other node. If an error were to occur in a link, only that link is affected and the mesh topology will reconfigure the routing path so a new network structure can be used (Forouzan, 2007). This is further explained in Chapter 4.

The module that was used in this project is called ZICM2410P2. It has extended range using external power amplifier. The manufacturers of the module CEL (California Eastern Laboratories) also manufactures the ZigBee tags, see Figure 5.1.

4. Radio Protocols

Radio protocols such as the ZigBee protocol have limited performance due to the limited range of the wireless receiver and the transmitter. Other factors such as environments with obstructions will further limit the reach of the wireless signal. The industrial, scientific and medical (ISM) radio bands such as 2.4 GHz are widely used by equipment with the ability to send radio signals. This can result in collision with signals with the same frequencies.

4.1 ZigBee Radio

The IEEE 802.15.4 standard is the basis for the ZigBee. The standard mainly focuses on very low complexity, low speed communication and low cost by consuming very low power. The range of communication is around ten meters, and it has a bit rate of 250 kbits/s, 40 kbits/s or 20 kbits/s. It operates on the unlicensed 2.4 GHz frequency band. It uses CSMA/CA as a method for avoiding collisions and also has support for secure communications.

In carrier sense multiple access (CSMA), when a packet is going to be sent via a medium, the station uses a carrier sensing scheme to sense the medium before trying to use it. In other words, CSMA is based on the principle “listen before talk”. What the station is listening to is signals from other stations. If there is no collision the station will only hear one signal which is its own signal. In the case of a collision the station will receive two signals; its own signal and the signal transmitted from a second station. In a wireless network, since a lot of the energy is lost in transmission, a collision will add around 5 to 10 percent additional energy to the detected signal. This additional percentage in energy is too small to be useful for wireless networking. Carrier sense multiple access with collision avoidance (CSMA/CA) is used to avoid collision in a wireless network. The principle is do delay any transmission, even if the channel is idle. If an idle

channel is found, the station waits for a period of time before transmitting. The time it waits is called the Interframe space (IFS). This will minimize possibility of collision with a signal from a distant station that have not yet been detected. The IFS time provides time for the distant signal to be detected. If the channel is still idle the station will send. In CSMA/CA stations can be given higher priorities by assigning them short IFS.

The importance of collision avoidance in a network is huge in the case of a warehouse where there are multiple stations, the lamps (ZigBee module) and the vehicles (ZigBee tag) (Forouzan, 2007). ZigBee is the most widely installed enhancement to the IEEE 802.15.4 standard. It has some additional communication features by using layer 3 and layer 4 in the OSI model (ZigBee Alliance, 2014). One of these features is support for mesh topology. With the mesh topology any node can communicate with any other node. This can be done by being in the range of a specific node or via multiple nodes. The benefit is that the network can be spread out over large areas. In the example of a warehouse, a lamp can communicate via a truck/tag through other lamps if the truck is out of range. Furthermore, if a node becomes disabled the signal will take a different path. Self-healing algorithms is a way for the mesh network to reconfigure if any paths or nodes are blocked or disabled, or with other word to check if it is a fully connected network. (Modiano, 2009)

5. Implementation

This chapter describes the methods used in this project to accomplish the task. To create and optimize a smart lamp the hardware explained in the previous chapter where used. Software was provided from manufacturers and LED Era to implement the hardware and make it possible to code and test the lamp. During the development and implementation process some obstacles occurred that will be discussed in this chapter.

5.1 ZigBee Implementation

The ZigBee Module (see Figure 5.1) and the ZigBee tags (see Figure 5.2) were purchased by LED-Era from manufacturers in Taiwan. The module was connected to a circuit board together with other sensors, such as the light sensor and the motion sensors. The board was then connected to the MCU inside the power supply with wires, see Figure 1 in Appendix 1 for the pin locations.

A software called Portal was used in order to make the module communicate with the tags. The Portal software came with the ZigBee devices from the manufacturer, and the language used to code the ZigBee was Python. By connecting the ZigBee tag via USB to a computer the module, now connected to the LED, could be remotely operated and initialized with the help of Portal.

To implement the module a new set of codes was required for the MCU. In order to program the MCU a computer was connected via JTAG to the MCU. The programming was done with a software called μ Vision and the language used to code the MCU was C.

The module has to measure the power of the radio signal from the tags attached to a vehicle to determine the distance between the lamp and the vehicle. Therefore, different power modes were implemented in the MCU. The power modes decide how much light the lamp needs to emit. The power modes also take

into account the brightness level already present in the area with help of the light sensor. The power modes can be changed in order to satisfy conditions and brightness level desired by the consumers.

The ZigBee tags only task is to frequently broadcast a signal that reaches ten meters. If a ZigBee module is in reach of the signal it will calculate the power level of the signal received by the antennas. An increase in the power level (signal strength) indicates that the vehicle is approaching the lamp. By using this technique, the lamp can estimate the vehicle's position. The ZigBee module can send out signal as well as read signals. Since ZigBee has support for mesh topology, see Chapter 4, other lamps outside the vehicle's signal reach can also estimate the vehicle's position by communicating with nearby lamps. If a lamp does not receive any signal from other lamps or vehicles then the lamp will automatically activate the motion sensor. The flowchart in Figure 5.3 shows this in more details.

The lamp that we received was already programmed to work with motion- and light sensors, so only the implementation of the ZigBee module was needed. The structure of the code in the MCU can be divided into two parts, depending on the different states of the lamp, i.e. the state when no ZigBee tag was detected and the state when the ZigBee tag was detected. In the former case the motion detector would be the only sensor that detects the movement. Since, when we received the lamp it already worked in this state, we mostly needed to focus on the latter case, i.e. the case that the ZigBee module receives the signal from a vehicle and the motion detector is not required. In both cases the light sensor is on and provides information about the lighting of the area. If more light is needed nearby lamps will emit more to fill the need.

The module came with basic software, but the code was hard to understand and lacked sufficient comments and descriptions. We encountered some problems in the initial stage of the work but after understanding the code it could be implemented on the MCU. After implementing the ZigBee module and the board containing the other sensors on the MCU the ZigBee tags had to be programmed so the module and the tags could communicate. Moreover, the Portal software was not quite user friendly and it was difficult to establish a connection between the tag and the module via Portal; Portal could not find the tag. However, after a reinstallation of the firmware on the tag the connection was successful. When the connection was open, it was possible to program the module and the tag, but the system did not react properly. The lamp was acting like there was no ZigBee

module connected. The voltage was measured on one of the pins in order to see if a ZigBee module was connected to the lamp or not. A model of the ZigBee module can be seen in Figure 1 in Appendix 2. No voltage was discovered on the pin hence the MCU could not detect the module. After some troubleshooting and tweaking in the code the pin was successfully activated. The ZigBee and the lamp now worked well and the different power modes could easily be differentiated. One remaining problem is a loss in signal between the tag and the module. The module loses the tag occasionally and the lamp goes back to only using the motion sensor and the light sensor before connecting to the tag again and working properly. This worked differently between the different tags so a conclusion is that there was a fault in the tags.



Figure 5.1. The board connecting all the sensors. The ZigBee module can be seen to the left in the picture.

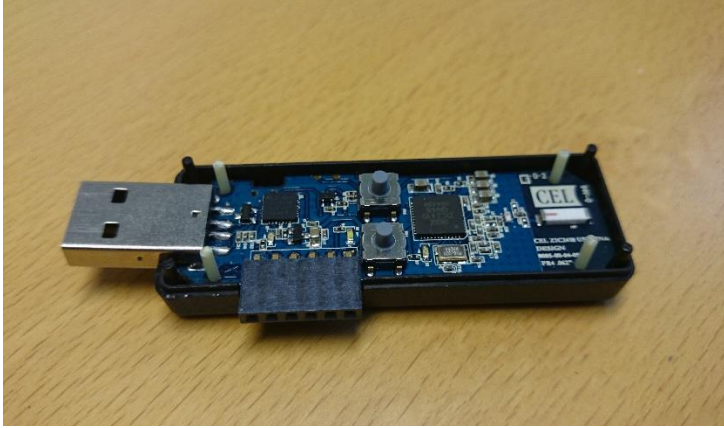


Figure 5.2. The ZigBee tag is connected to vehicles and communicates with the ZigBee modules in the lamps

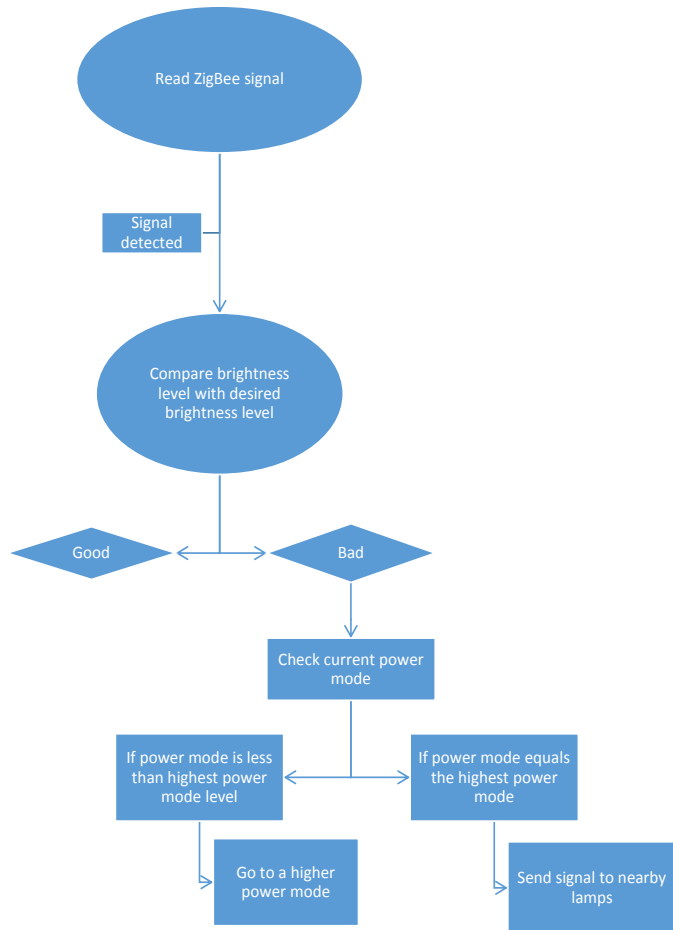


Figure 5.3. The flowchart shows the lamp's different states. The first state is a loop that constantly is waiting for a signal from a ZigBee tag and calculating the signal strength when it occurs. When a signal is detected and the signal strength is measured the system compares the brightness level of the area with the desired brightness level when a vehicle is present. If the brightness level is smaller than the desired brightness level the lamp dims up to a higher power mode. This will occur until the power mode is set to the highest value or the brightness level is satisfied. If the power mode is set to the highest value and the brightness level still does not satisfy the desired brightness level nearby lamps will go to higher power modes to provide the area with more light.

6. Simulation

In this chapter the simulation part of the project is introduced. The background of the program, TrueTime, that was used and some of its different functions will be described. Further, the process of creating the model used in the simulation will be explained.

6.1 TrueTime

TrueTime is an add-on for Matlab/Simulink developed by the Department of Automatic Control, Lund University (Cervin, Henriksson, Lincoln, Årzén, & Eker, 2003). It is a simulator for real-time control systems which supports co-simulations of controller task execution in real-time kernels, network transmissions, and continuous plant dynamics.

TrueTime introduces a new set of blocks, as can be seen in Figure 6.1. One of them is the computer block which is event driven; it is also used for executing user-defined tasks and has interrupt handlers. The other block is the network block which also is event driven and is used for sending and receiving messages over the network, depending on the chosen network model. An example of an event is “a message arrived on the network” which corresponds to an interrupt (Cervin, Henriksson, Lincoln, Årzén, & Eker, 2003).

Since the user determines the level of simulation this allows for either constant, random or data-dependent models of the transmission times of messages and execution time of tasks.

The interface uses Simulink blocks and the coding can either be done using Matlab-code or C++. In this report only the TrueTime kernel- and wireless network blocks were used and the coding was done using Matlab-code (Cervin, Henriksson, Lincoln, Årzén, & Eker, 2003).

6.1.1 Compilation

For running the M-file API there is no need for a compiler since the TrueTime archive contains pre-compiled files.

A compiler will be needed if the code were to be written in C++ code. This is configured in Matlab using the command

```
>> mex -setup
```

The recommended C++ compilers for TrueTime are:

- Visual Studio C++ for Windows
- gcc, g++ for Linux

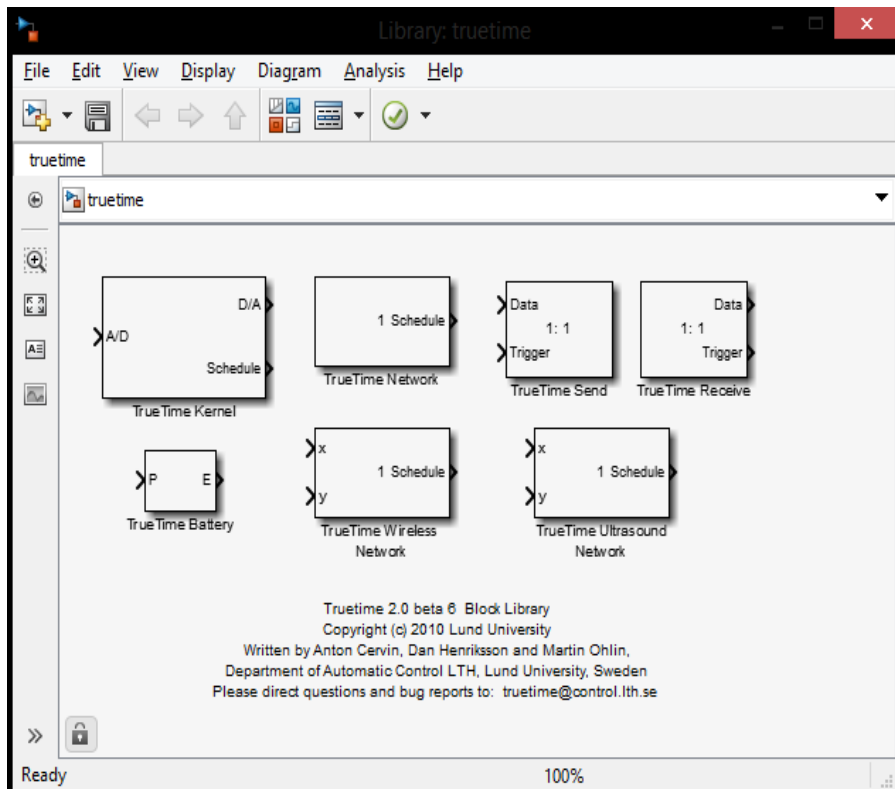


Figure 6.1. The different blocks that TrueTime has.

6.1.2 TrueTime Kernel

The TrueTime kernel block is a Simulink s-function that has been configured to work as a computer with a flexible but simple real-time kernel. It contains A/D and D/A converters as well as a network interface and external interrupt channels. The data structures of the kernel are much like the ones found inside a real-time kernel such as a time queue and a ready queue, monitors and timers. It also contains interrupt handlers and records for tasks and these are defined by user-written code functions either in C++ or Matlab-code, (Cervin, Henriksson, Lincoln, Årzén, & Eker, 2003).

6.1.3 TrueTime Wireless Network

The TrueTime wireless network block is also event driven. This means that it will execute when a message containing information about the sending and receiving node, usually just arbitrary data, enters or leaves the network. The wireless network supports IEEE 802.11b/g (WLAN) and IEEE 802.15.4 (ZigBee). The wireless network block is configured through the block mask, see Figure 6.2. All of the parameter default values (Cervin, Henriksson, & Ohlin, TrueTime 2.0 beta 5 - Reference Manual, 2010, ss. 21-22) were used, except for the network number, which was set to 1 and the number of nodes, which was set to 5.

The ZigBee protocol that TrueTime uses has a quite low bandwidth and also low power consumption. It is designed with sensors and simple control networks, which are based on the CSMA/CA (see Chapter 4.1) and 802.11b/g, although the protocols are not the same and the ZigBee is much simpler.

As can be read in (Cervin, Henriksson, & Ohlin, TrueTime 2.0 beta 5 - Reference Manual, 2010, s. 23), the packet transmission model in ZigBee is similar to WLAN although the MAC procedure differs and is modeled as:

1. Initialize:
NB=0.
BE=macMinBE.
2. Delay for a random number of backoff periods in the interval $[0, 2^{BE} - 1]$.

3. Is the medium idle?
if yes: send.
else: goto 4.
4. Update the backoff counters:
NB=NB+1.
BE=min(BE+1, aMaxBE).
5. Is NB>macMaxCSMABackoffs?
if yes: drop the packet.
else: goto 2.

An explanation of the variable names (Cervin, Henriksson, & Ohlin, TrueTime 2.0 beta 5 - Reference Manual, 2010, ss. 23-24), which are taken from the standard to make comparisons easier, follows:

NB: Number of backoffs.

BE: Backoff exponent.

macMinBE: The minimum value of the backoff exponent in the CSMA/CA algorithm. The default value is 3.

aMaxBE: The maximum value of the backoff exponent in the CSMA/CA algorithm. The default value is 5.

macMaxCMABackoffs: The maximum numbers of backoffs the CSMA/CA algorithm will attempt before declaring a channel access failure. The default value is 4.

Function Block Parameters: TrueTime Wireless Network1

Wireless Network (mask) (link)

Parameters

Network type: 802.15.4 (ZigBee)

Network Number:
1

Number of nodes:
5

Data rate (bits/s):
800000

Minimum frame size (bits):
272

Transmit power (dbm):
20

Receiver signal threshold (dbm):
-48

Pathloss function: default

Pathloss exponent ($1/\text{distance}^x$):
3.5

ACK timeout (s):
0.00004

Retry limit:
5

Error coding threshold:
0.03

Loss probability (0-1):
0

Initial seed:
0

☒ Show Schedule output port

☐ Show Power consumption output port

OK Cancel Help Apply

Figure 6.2. The dialog of the TrueTime wireless network block with ZigBee as the chosen network type.

6.2 TrueTime Model

To add another dimension to the thesis a simulation of a warehouse was made. This was made using TrueTime which has support for ZigBee-networks. The simulation was made based on the example of a forklift in a warehouse from Chapter 2. Initially seven blocks were used representing the different components for the simulation, which can be seen in Figure 6.3.

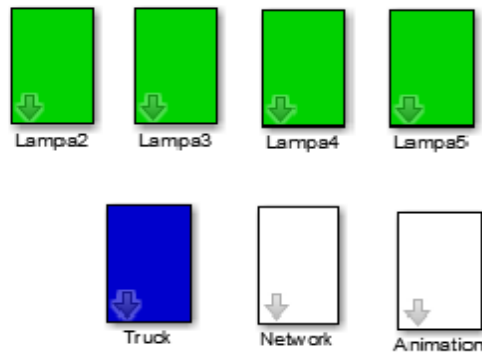


Figure 6.3. Overview of the blocks which were used in the simulation.

The network block. Figure 6.4, contains a TrueTime wireless network block that has been set to a ZigBee network. The inputs are retrieved from the different kernels representing the forklift and the lamps. The network's task is to keep the lamps informed of the forklift's position; this is done by measuring the strength of the signal that is being broadcasted through the network. The code for the network can be found in Table 6.1.

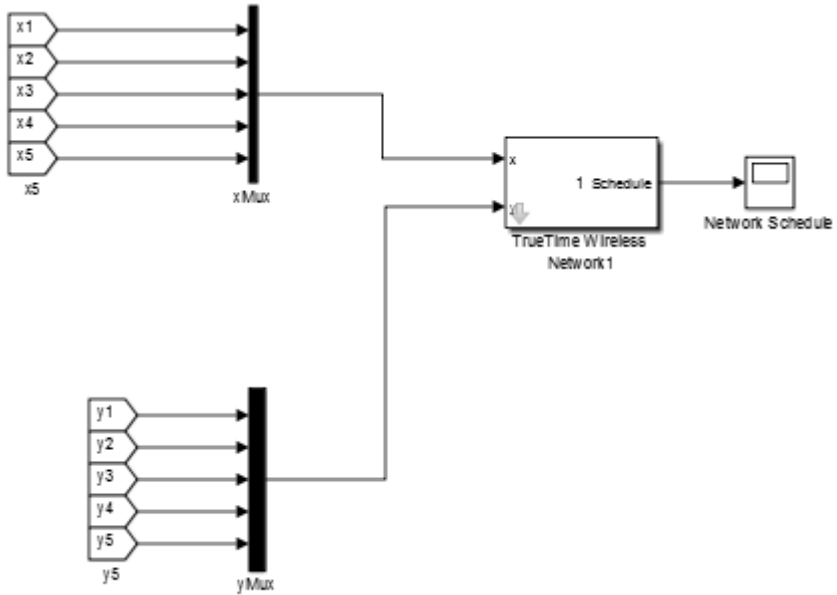


Figure 6.4. The network model contains the fetched x- and y inputs and a TrueTime wireless network.

The block called “truck”, Figure 6.5, is used to control the forklift properties. It contains a kernel with an integrator connected to it to make it move from left to right (starting from zero) on the x-axis and a constant value, set to zero, on the y-axis. While moving on the x-axis the forklift will broadcast an arbitrary data message throughout the network every second. See Table 6.2 for the Matlab code for the forklift.

```

function node_lamp_init(arg)

% Initialize TrueTime kernel
ttInitKernel('prioDM')
% deadline-monotonic scheduling

% Create and attach network interrupt handler

ttCreateHandler('network_handler',1, 'nwhandler_code')
ttAttachNetworkHandler('network_handler')
ttNoSchedule('network_handler');

function [exectime, data] = nwhandler_code(seg, data)
global signalPower
persistent u;

switch seg
case 1
    [u, signalPower] = ttGetMsg(1);
    ttAnalogOut(1, signalPower)
    exectime = 0.0005;

case 2
    exectime = -1; % finished
end

```

Table 6.1. Matlab code for the lamps and network handler.

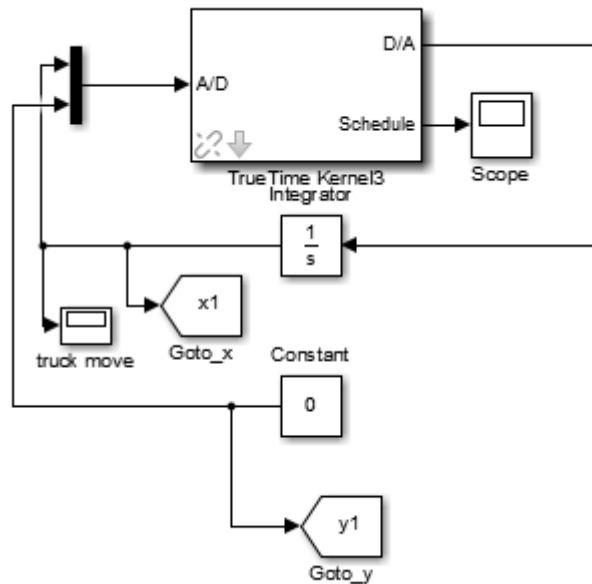


Figure 6.5. The truck block contains a TrueTime kernel, an integrator that changes the x position and a constant for the y position.

The lamps coordinates are constant values that have been preset to keep them on the same level on the y-axis and some distance apart from each other on the x-axis. The lamps will listen to the message that is being broadcasted from the forklift. Depending on the strength of the signal, the lamps will enter different states that are also known as power modes that will represent the dimming modes in the lamps. In Figure 6.6 the lamp model is shown and Table 6.1 shows the code for the lamps.

```

function node_truck_init(arg)

ttInitKernel('prioFP');
%initializing kernel, fixed priority

data.x1 = 0;
starttime = 0.0;           % control task start time
period = 0.5;              % control task period
sendperiod = 1;

ttCreatePeriodicTask('move_task',starttime,period,
'move_code', data);

ttCreatePeriodicTask('send_task',starttime,sendperiod,
'send_code',data);

```

```

function [exectime, data] = send_code(segment, data)

switch segment
case 1
    ttSendMsg(0,1,80); % send msg (80 bits) to node 1
    exectime = 0.01;
case 2
    exectime = -1; % finished
end

```

```

function [exectime, data] = move_code(segment, data)

switch segment
case 1
    data.x1 = 6;
    ttAnalogOut(1,data.x1);
    exectime = 0.01;
case 2
    exectime = -1; % finished
end

```

Table 6.2. Matlab code for the forklift. With the initialization-, send- and move code.

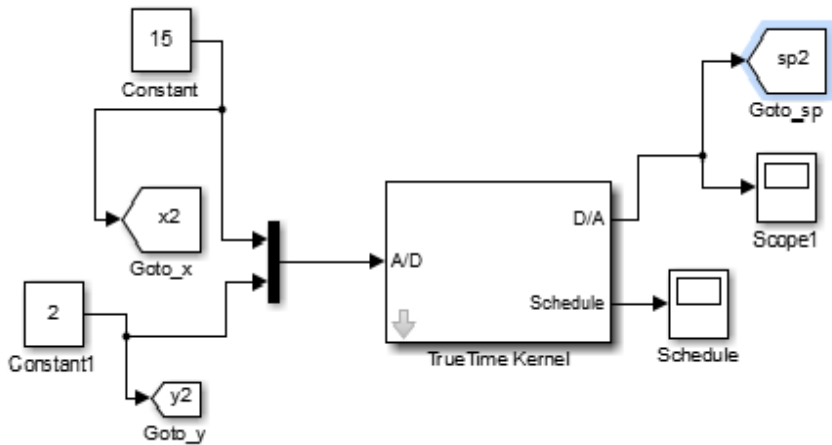


Figure 6.6. Model of how the lamps were made. It contains a TrueTime kernel, two constants that set the x- and y positions for the lamp, which are different for each lamp.

For the simulation an animation, Figure 6.7, was made to give a visual interpretation of how this would work. The x- and y-coordinates are retrieved from the different blocks and are sent into the animation function. The “sp” input fetches the varying signal strength for every individual lamp. The lamps are represented as colors which will indicate the power mode it is currently on. The colors follow:

- Red color represents the lamp being off.
- Yellow color represents the lamp on dim mode, which means it is either increasing or decreasing in power whether the forklift is moving towards or away from it.
- Green color represents the lamp at full power.

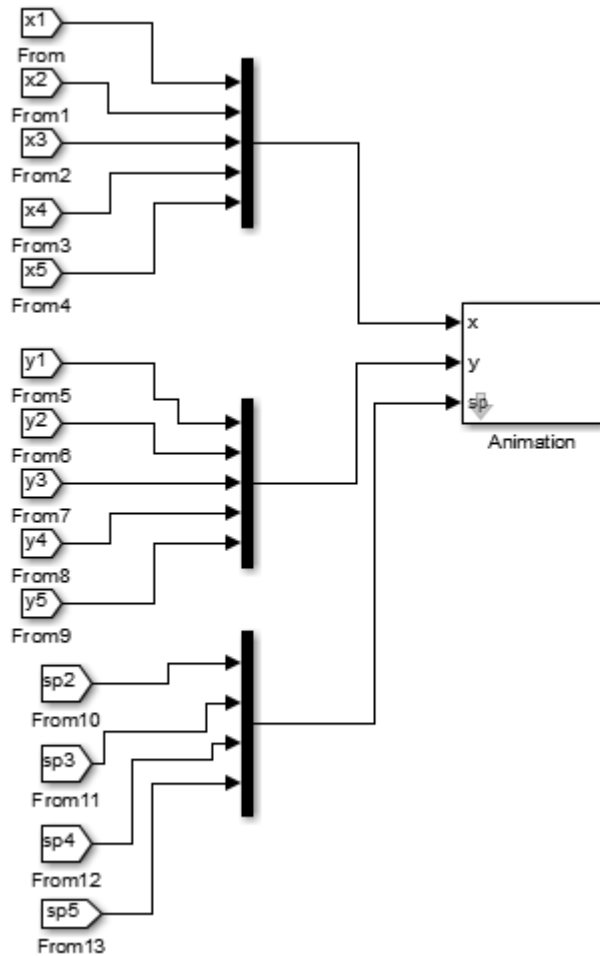


Figure 6.7. The animation model that retrieves the three different inputs and puts it into the animation block. The animation code can be found in Appendix 5.

7. Result

In this chapter, the results of the lamp and the simulation are presented. The first section is a walkthrough of how the lamp worked after programming it and the implementation of the ZigBee described in Chapter 5. This is followed by results from the simulation described in Chapter 6.

7.1 Lamp

After the initialization of the lamp and the ZigBee the system performed quite well. A problem occurs when the ZigBee tag gets a little too far away from the lamp. The lamp loses connection with the tag and start using the other sensors (motion and light sensors) instead. The problem is that the ZigBee signal should be able to reach longer than it does at the moment. This could be a factor of many things, e.g. too many signals using the frequency of 2.4 GHz such as Bluetooth, Wi-Fi etc. that disturbs the signal between the tag and the module. It can also be that there are too many insulating walls between the module and the tag at the place where the ZigBee was tested. Nevertheless, this solution proved to work well at desired environments with large spaces such as warehouses and industrial facilities.

A fluorescent light tube usually consumes around 60 Wh (Durda, 2007). Since every lamp contains at least four tubes the consumption of the lamp is 240Wh. In a warehouse with a 24/7 operation the fluorescent light will consume 5760 Watts. The LED used in this project consumes 88 Watts in average (LP5/LS5 Specifications). With the smart lighting system the LED will be emitting light for an average of 12 hours a day. This leads up to an average of

1056 Watts per day. One fluorescent lamp will in this case consume 5.45 times more energy than a smart LED lighting system.

7.2 Simulation

The first part of the simulation went smoothly, with some minor problems but with some trial and error good results was achieved. The graphs in Figure 7.1, 7.2, 7.3 and 7.4, show the different lamps signal power during the simulation. The signal power output has negative values and never reaches zero but it still follows the pattern with increasing signal power while getting closer to the specific lamp and decreasing when moving away from the lamp.

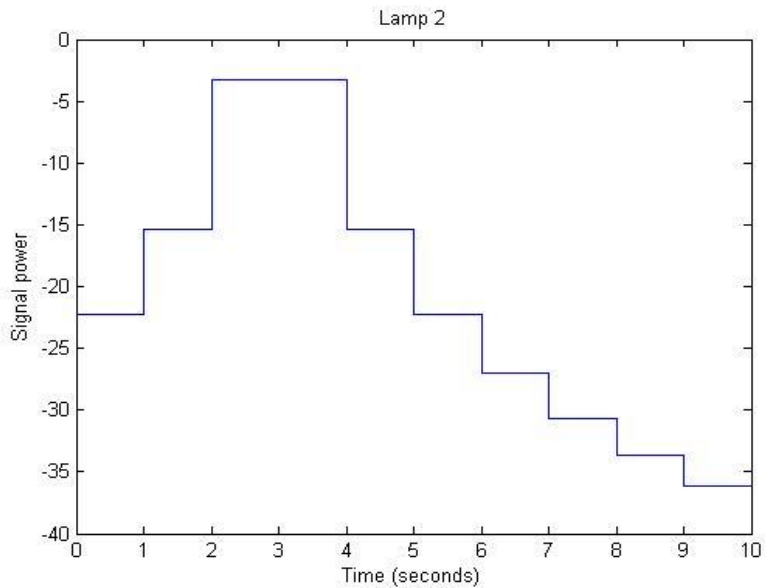


Figure 7.1. The signal power from the first lamp

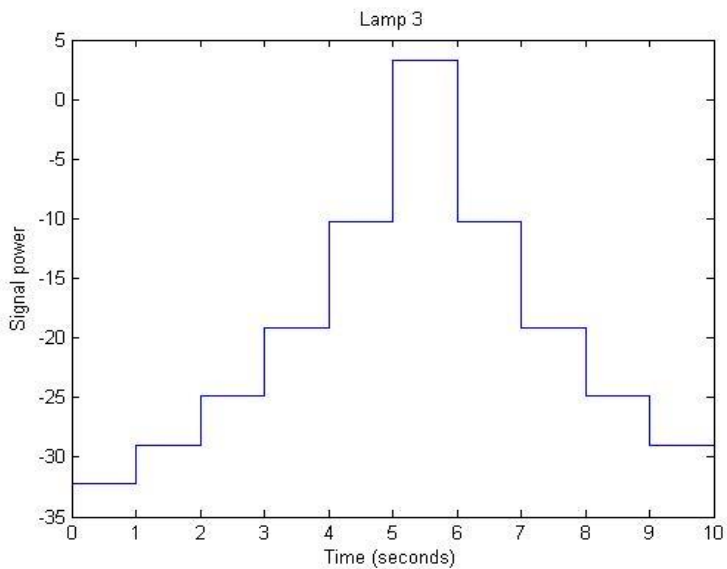


Figure 7.2. The signal power from the second lamp

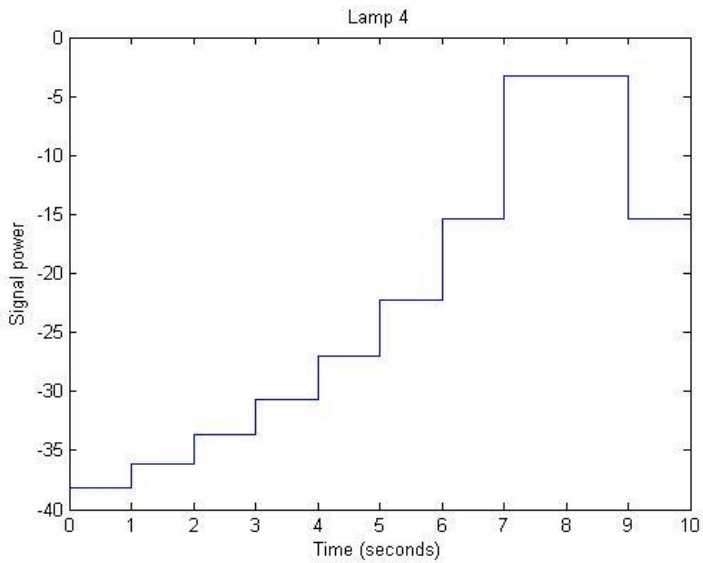


Figure 7.3. The signal power from the third lamp.

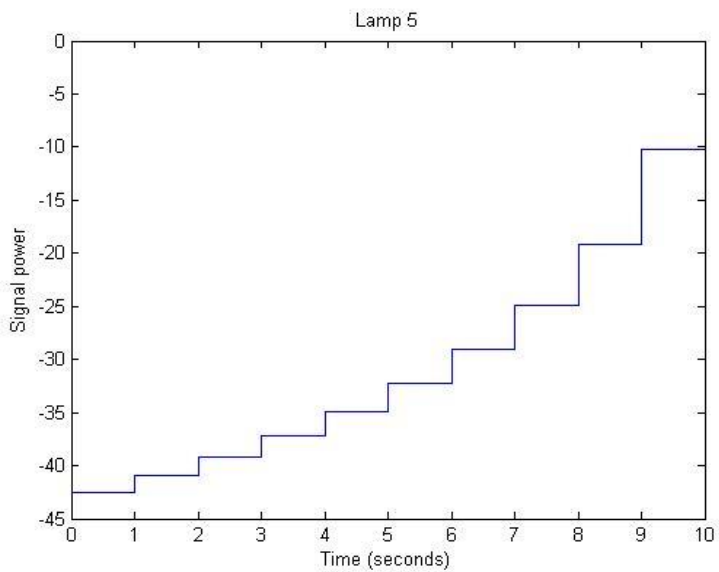


Figure 7.4. The signal power from the fourth lamp.

The network works as intended as well, as can be seen in Figure 7.5. The communication in the ZigBee network or more specifically the communication between the forklift, that sends a message on the network every second, and the lamps that are listening and waiting to execute every time a message is retrieved. The y-axis positions indicate which lamp that is listening and the x-axis is the time.

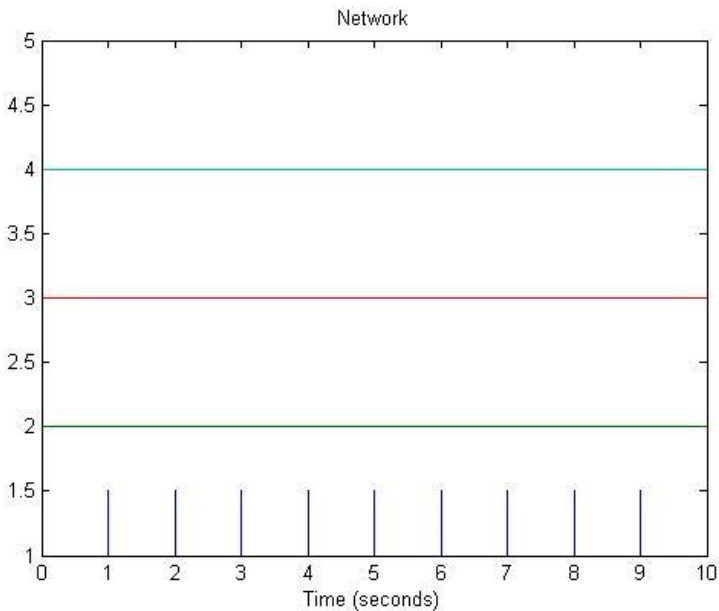


Figure 7.5. The network where the forklift is broadcasting a signal every second and the lamps are listening.

The second part of the simulation, the animation, which connected all the components, gave good results. Because the signal power had negative values, the limits of the different power modes were set to:

- For off, red color, when the signal power is less than minus twenty.
- At the dimming level, the color yellow, the signal power is greater or equal to minus twenty but less than minus twelve

- When the signal is rising to full power, green color, the signal power is greater or equal to minus twelve.

This led to a good working animation which could give a brief and simplified overview of the system. Figure 7.6, 7.7 and 7.8, shows the animation part of the simulation. The blue object is the forklift moving, from left to right, on the x-axis and the four other objects are the lamps. The lamps change their dimming state depending on the position of the forklift.

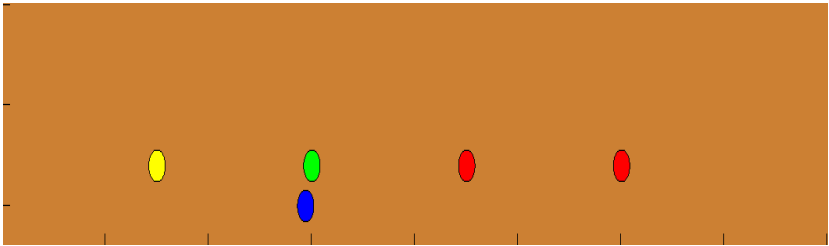


Figure 7.6. A picture from the animation where the blue circle represents the moving forklift from left to right. The yellow circle is a dimmed lamp, the green circle is a lamp at full power and the red circles represents lamps that are off.

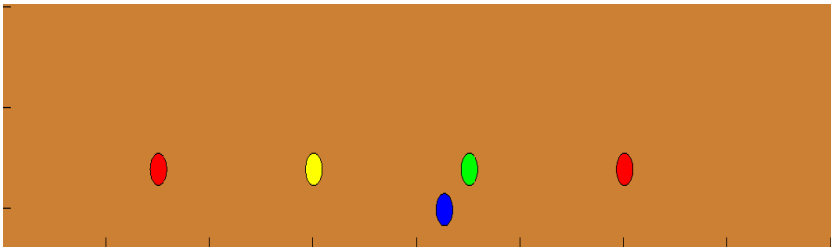


Figure 7.7. An updated picture from the animation where the blue circle represents the moving forklift from left to right. The yellow circle is a dimmed lamp, the green circle is a lamp at full power and the red circles represent lamps that are off.

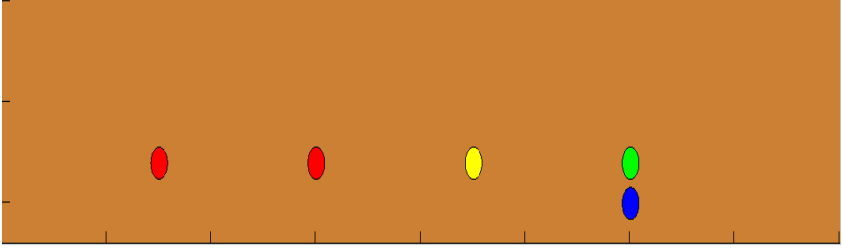


Figure 7.8. An updated picture from the animation where the blue circle represents the moving forklift from left to right. The yellow circle is a dimmed lamp, the green circle is a lamp at full power and the red circles represent lamps that are off.

8. Summary and Conclusion

This section is a summary of the project. The first part is about the lamp and the conclusions made during the project and future improvements and implementations that could be made to get more out of the system. The second part is a summary of the simulation and conclusions done during the process. Last there is a summary of the problems and obstacles that occurred.

8.1 Lamp

Using ZigBee as a main sensor to detect moving vehicles was proven to be very successful. The ZigBee protocol's advantages see Chapter 3 and 4, were the main reason why that protocol was used during the project are presented. During the project we found that ZigBee also has its limitations. Every vehicle must have a ZigBee tag to be detected by the module on the lamp. This limits the systems usage. In warehouses or in garages where there are no vehicles or the vehicles has no tags there must be other solutions. In a warehouse without vehicles the people that work there must carry a tag, or in garages where different vehicles enter, the system has to take advantage of sensors other than the ZigBee sensor for a smart lighting solution. This systems main focus was therefore only warehouses and places where the same vehicles are present all the time. Another disadvantage is the high initial cost of a ZigBee module. Compared to for example Wi-Fi and Bluetooth the ZigBee is cheaper, but to set up an entire network with more than a 100 modules the initial price would become very high. In the long run the costumers would benefit from the system since it is energy saving. There are still some improvements to be made to the system before it is fully ready for the market.

There is a main controller for the system that was not implemented that controls every lamp from a wireless controller. This controller can manually dim all the lights to fit desired conditions by the consumers. This could be a work for

the future, to also try to control the lights manually. During the project no time could be found to learn how this works and how to implement it. Some minor improvements such as smoother dim transition are also required. Since there was only one lamp to work with it was hard to really try out the result in real life. Simulations were made and many tests on the lamp were made but unless there is a place to try several lamps it is hard to really see how well the system works.

The expectations of the project were high. Even though we faced obstacles and problems during the time of the project, we still managed to cope with our task and reach the goal. During the process we talked to our supervisor at LED Era about the potential of creating a smartphone app. The purpose of the app was to manually set up an entire network of lamps at a warehouse. Imagine setting up all the lamps on their desired location (ceilings or walls) and then wirelessly setting up the network using an app with graphic user interface for convenience. This way it would be very easy for customers to set up a network without the need of professional help.

For future improvements to the system the ZigBee could be used to pinpoint the location of any forklift carrying a tag. By using several lamps and calculating the signal strength from a tag to these lamps one can make an accurate estimation of the position of the forklift. This data can be used to adjust the brightness level of the lamps even better.

8.2 Simulation

The purpose of the simulation was to make a simple yet clearer picture of how the lamps could work in reality. The choice of just having four lamps and three different power modes was made because of the animation part. The animation part was harder than expected and to get good results within the time frame we had to limit the model. We tried to implement more lamps but the results did not come up any good. The course was too small which means that the distance between the lamps was too small and by just moving on the x-axis and with the scaling of the figure it became too tight. We also tried to implement more power modes than just three, but what happened was that the change between the power modes was too jumpy and it lacked a natural flow, meaning it did not look good. Instead we chose to let the yellow color represent a varying dimming level, meaning the higher signal power the higher brightness and vice versa to give it a

more natural transition. Unfortunately neither this could be demonstrated in the animation. By adding more lamps the same issues also apply, it basically did not look any good in the animation.

There was no communication between the lamps implemented in the simulation. Figure 6.8 is a good example when illustrating the difference between the simulation and the real lamp. In the picture one can see that the second lamp is at full power (green) and the third lamp is off (red). This differs with what would happen in reality when the communication between the lamps is implemented. When the second lamp receives a continuously increasing signal power and switches to full power, it would communicate to the third lamp and letting it know that the forklift is moving towards it. Then the third lamp would switch into dim mode (yellow) in advance even though it has not received the signal power required for it to enter the dim mode.

What could have been made for the simulation was a more sophisticated model. Meaning, if this simulation could have been implemented in another program, with support for a more powerful graphic modelling then a more warehouse realistic model could have been made. One could for example make obstacles represent shelves, add more lamps and other objects normally found in a warehouse and to make the forklift to move on both the x- and y-axis. But due to lack of time and knowledge of a program that has support for all this, this will be considered work for the future.

8.3 Problems

Initially there were some problems with the hardware. Some of the components did not work and quite some time went to figuring out what was wrong. At first the lamp did not respond to the tags and by some troubleshooting it was found that the ZigBee module was broken, fortunately another board was available so after connecting all the wires to the new board a connection could be found.

When trying to change the code on the tags another problem occurred, the new code did not work. The code could not be transferred to the tag even though the program said that the new code had been implemented. To be able to put new code on the tags a hard reset (restoring it to the factory default mode) of them and putting the firmware back on it was needed and it took some time before we got that information.

The code that was provided almost had no comments at all and neither of us had much experience with C programming so it took a while before understanding it.

8.4 Conclusions

Essentially, the projects goal was achieved; the system works. There are still some improvements needed before it is ready for the market. We are convinced that this system has the potential to save energy if used in its intended environment, as compared to what many companies are using as their main lighting source today. However, it is hard to measure exactly how much energy a whole system would save, since only one lamp has been tested.

9. Bibliography

- Belysningsbranschen. (2012, December). *Allt ljus på smart belysning*. Retrieved from <http://belysningsbranschen.se/files/2012/12/Allt-ljus-p%C3%A5-Smart-belysning.pdf>
- Belysningsbranschen. (2012, September). *Så här fungerar smart belysning*. Retrieved from <http://smartbelysning.nu/2012/09/sa-har-fungerar-smart-belysning/>
- Belysningsbranschen. (2013, September). *En ljusare framtid*. Retrieved from http://belysningsbranschen.se/files/2013/09/En-ljusare-framtid_2013_final_low1.pdf
- Cervin, A., Henriksson, D., & Ohlin, M. (2010, June). *TrueTime 2.0 beta 5 - Reference Manual*. Retrieved from <http://www3.control.lth.se/truetime/report-2.0-beta5.pdf>
- Cervin, A., Henriksson, D., Lincoln, B., Årzén, K.-E., & Eker, J. (2003, June). *How Does Control Timing Affect Performance? Analysis and Simulation Using Jitterbug and TrueTime*. Retrieved from <http://www.control.lth.se/documents/2003/cer+03.pdf>
- Durda, F. (2007). *The Fluorescent Lighting System*. Retrieved from <http://nemesiss.lonestar.org/reference/electricity/fluorescent/efficiency.html>
- Energimyndigheten. (2011). *Energipolitik EU*. Retrieved from <https://www.energimyndigheten.se/Internationellt/Internationellt-energisamarbete--/Energifragan-inom-EU/Aktuellt-inom-EU/>
- Forouzan, B. A. (2007). *Data communications and networking* (fourth edition ed.).
- Hung, C.-H., Bai, Y.-W., & Tsai, R.-Y. (2011). *Digital Control for Home Lighting Systems with ZigBee Communication*. Retrieved from <http://cs.ee.fju.edu.tw/paper/201101.pdf>

- International Parking Institute. (2014). *IPI*.
- Laike, T., & Küller, M. (2012). *Kunskapskällor för ljusforskning och belysning*. Retrieved from http://mpe.arkitektur.lth.se/fileadmin/miljopsykologi/foton/hemsdieversion_med omslag.pdf
- Ljuskultur & Belysningsbranschen. (2011, May). *Värt att veta om belysning med LED*. Retrieved from How Stuff Works: http://www.energimyndigheten.se/Global/F%C3%B6retag/Belysning/Vart_att_veta_om_belysning_med_LED_2011.pdf
- Marvell Technology Group Ltd. (2014). *Ltd, Marvell Technology Group*. Retrieved from <http://www.marvell.com/led-lighting/>
- Marvell Technology Group Ltd. (2014). *Marvell Technology Group Ltd*. Retrieved from <http://www.marvell.com/led-lighting/control-platforms/commercial-platforms/>
- Mathas, C. (2012, June 28). *Sensing Motion with Passive Infrared (PIR) Sensors*. Retrieved from <http://www.digikey.com/en/articles/techzone/2012/jun/sensing-motion-with-passive-infrared-pir-sensors>
- Modiano, E. (2009). *Packet multiple access: The Aloha Protocol*. Massachusetts Institute of Technology. Retrieved from http://ocw.mit.edu/courses/aeronautics-and-astronautics/16-36-communication-systems-engineering-spring-2009/lecture-notes/MIT16_36s09_lec21_22.pdf
- Philips. (2013). *Meet Hue*. Retrieved from www.meethue.com
- Schultz, C. v. (2012, February 15). *Fyra av fem led-lampor underkända*. Retrieved from NyTeknik: http://www.nyteknik.se/nyheter/energi_miljo/energi/article3408512.ece
- Synapse. (2010). *SNAP Hardware: Technical Manual*.
- Tesla, N. (1930). *Science and Invention* (Volume XVII ed.).
- Texas Instruments. (2011). *LM3S9B92*.
- U.S. Energy Information Administration. (2014). *U.S. Energy Information Administration*. Retrieved from <http://www.eia.gov/tools/faqs/faq.cfm?id=99&t=3>
- ZigBee Alliance. (2014). www.zigbee.org.
- ZigBee Alliance. (2014). *ZigBee Alliance*. Retrieved from <http://www.zigbee.org/Specifications/ZigBee/Overview.aspx>

Appendix 1

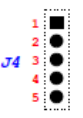


MCU Board interface:			25-wires Pin-Number/Color	
 J4	pin-1	USB_DEC	No. 1	Black (P570)
	pin-2	USB_ID	No. 2	Brown (P571)
	pin-3	USB_DP	No. 3	Red (P572)
	pin-4	USB_DM	No. 4	Orange (P573)
	pin-5	VUSB: 5V	No. 5	Yellow (P574)
 J5	pin-1	U1_RI GND	No. 6	Green (P575)
	pin-2	U1_DTR	No. 7	Blue (P576)
	pin-3	U1_DSR CDS	No. 8	Violet (P577)
	pin-4	U1_TX	No. 9	Gray (P578)
	pin-5	U1_RTS PIR1	No. 10	White (P579)
	pin-6	U1_RX	No. 11	Pink (P600)
	pin-7	U1_CTS PIR0	No. 12	Light Green (P601)
	pin-8	U1_DCD	No. 13	Black/White (P570/P579)
	pin-9	U1_GPIO_16 3V3_LS	No. 14	Brown/White (P571/P579)
	pin-10	U1_GPIO_15	No. 15	Red/White (P572/P579)
 J6	pin-1	3V3_ZB	No. 16	Orange/White (P573/P579)
	pin-2	ZB_RESET	No. 17	Green/White (P575/P579)
	pin-3	ZB_GPIO_11	No. 18	Blue/White (P576/P579)
	pin-4	ZB_U2RX_P10	No. 19	Violet/White (P577/P579)
	pin-5	ZB_GPIO_12	No. 20	Red/Black (P572/P570)
	pin-6	ZB_U2TX_P11	No. 21	Orange/Black (P573/P570)
	pin-7	ZB_GPIO_13	No. 22	Yellow/Black (P574/P570)
	pin-8	ZB_ISP	No. 23	Green/Black (P575/P570)
	pin-9	ZB_GPIO_14	No. 24	Gray/Black (P578/P570)
	pin-10	GND	No. 25	Pink/Black (P600/P570)

Figure 1. The pin directories used when connecting the MCU to the board containing the sensors.

Appendix 2

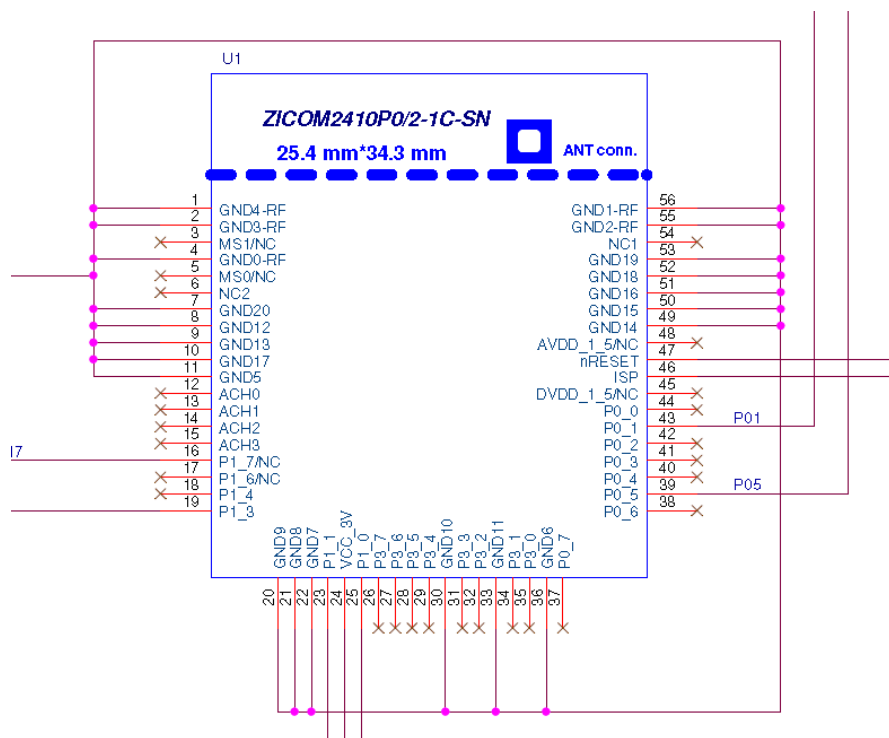


Figure 1. A view of the pins on the ZigBee module.

Appendix 3

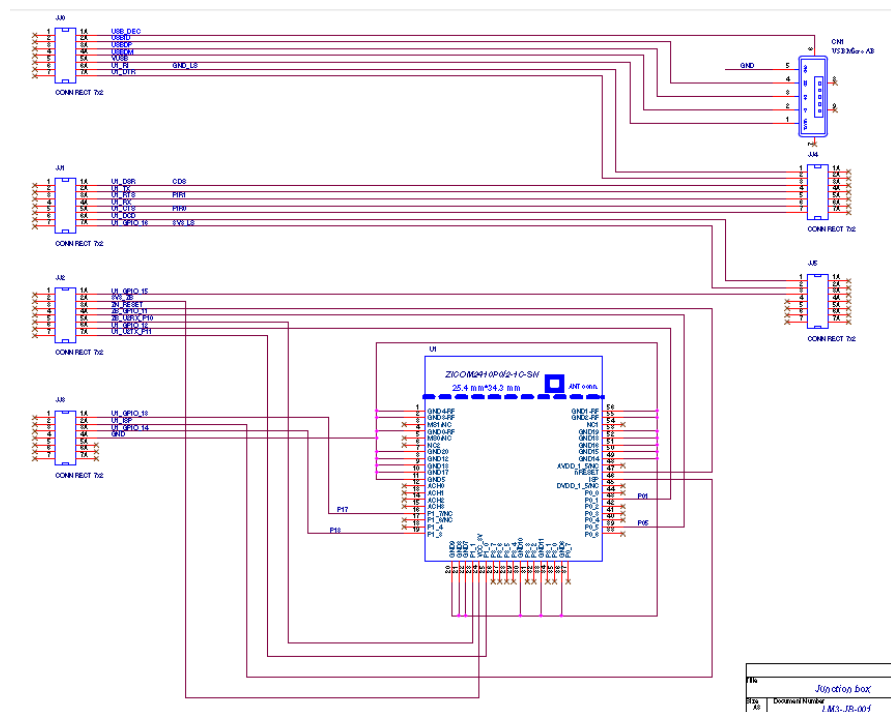
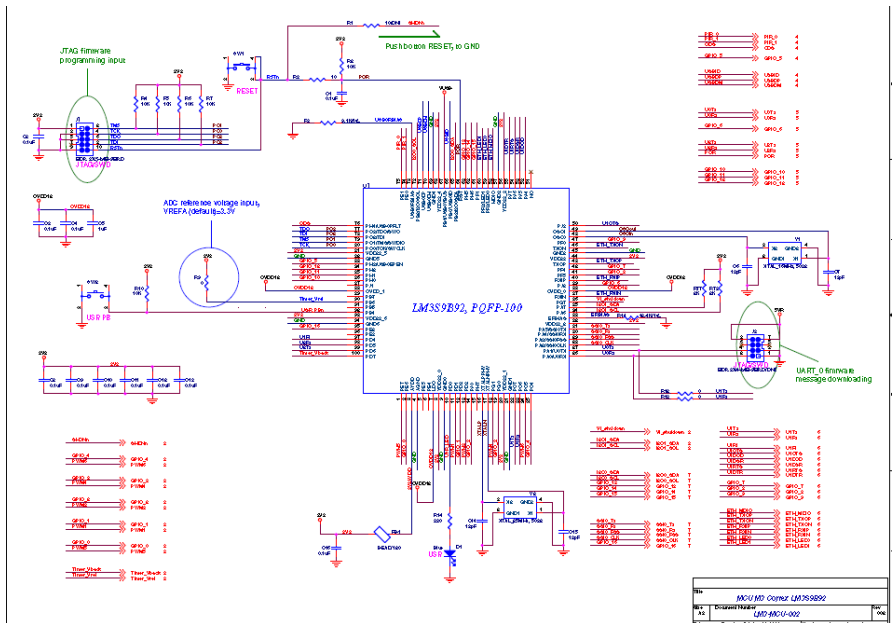


Figure 1. A view of how the pins on the ZigBee module are connected.

Appendix 4



Appendix 4. A view of the MCU pins and how they are connected

Appendix 5

The Matlab code for the animation.

```
function
[sys,x0,str,ts]=lampanimation(t,x,u,flag,params,xInit,y
Init)

xInit = 2:2:10;
yInit = [2 2 2 2];

switch flag,
    case 0,

        [sys,x0,str,ts]
mdlInitializeSizes(params,xInit,yInit);
    case 2,
        sys = mdlUpdate(t,x,u);
    case 3,
        sys = mdlOutputs(t,x,u);
    case 9,
        sys = mdlTerminate(t,x,u);
    otherwise
        error(['Unhandled flag = ',num2str(flag)]);
end

function
[sys,x0,str,ts]=mdlInitializeSizes(params,xInit,yInit)

global lamphdl ID truckhdl

% Read parameters
```

```

numLamps = params(1);
ts = params(2);
xmax = params(3);
ymax = params(4);

```

```

% Initialize the figure for use with this simulation

figure(1),clf;
title('Warehouse');
set(gcf,'Renderer','OpenGL');
set(gcf,'Position',[0 20 550 800])    %[700 500 700
1050])
grid off
set(gca,'Xlim',[0                                xmax],
'Ylim',[-2 ymax],'nextplot','add');
%set(gca,'XTickLabel',[]);
%set(gca,'YTickLabel',[]);
%set(gca,'XTick',[-xmax+0.5 0 xmax-0.5]);
%set(gca,'YTick',[-ymax+3 0 ymax-3]);
set(gca,'Color',[0.8 0.5 0.2]);

lampradius = 0.8;

grad = 0:10:360;

X = lampradius*cos(grad*pi/180);
Y = lampradius*sin(grad*pi/180);

truckX = X;
truckY = Y;
truckhdl = patch(1*truckX,1*truckY,'b');

sizes = simsizes;
sizes.NumContStates = 0;
sizes.NumDiscStates = 0;

```

```

sizes.NumOutputs      = 0;
sizes.NumInputs       = 14; % x and y
sizes.DirFeedthrough  = 0;
sizes.NumSampleTimes  = 1;

```

```

sys = simsizes(sizes);

x0  = [];
str = [];
ts  = [ts 0]; % Hybrid block

%=====
%=====
% mdlOutputs
%=====
%=====

function [sys]=mdlOutputs(t,x,u)

sys = [];

%=====
%=====
% mdlUpdate
%=====
%=====

function [sys]=mdlUpdate(t,x,u)

global lamphdl ID truckhdl signalPower col
signalPower;
%numLamps = length(lamphdl);
numLamps = 5;

lampradius = 0.8;
grad = 0:10:360;
X = lampradius*cos(grad*pi/180);
Y = lampradius*sin(grad*pi/180);

```

```

%=====
=====
%update graphics
%=====
=====

truckX=X'+u(1);
truckY=Y'+u(6);

set(truckhdl,'XData',truckX, 'YData', truckY);
drawnow;
xInit = [u(2) u(3) u(4) u(5)];
yInit = [u(7) u(8) u(9) u(10)];

for k=1:4
    lampX= X'+xInit(k);
    lampY= Y'+yInit(k);
    if u(10+k)<-20
        col = 'r';
    else if u(10+k) > -20 & u(10+k)< -12
        col = 'y';
    else
        col = 'g';
    end
end

    lamphdl(k) = patch(lampX,lampY,col);
    %set(lamphdl(k),'XData',lampX, 'YData', lampY);

end

drawnow;

sys=[];

%=====
=====

```

```
% mdlTerminate
% Perform any end of simulation tasks.
%=====
=====

function sys=mdlTerminate(t,x,u

sys = [];
```

Lund University Department of Automatic Control Box 118 SE-221 00 Lund Sweden		<i>Document name</i> MASTER'S THESIS	
		<i>Date of issue</i> September 2014	
		<i>Document Number</i> ISRN LUTFD2/TFRT--5955--SE	
<i>Author(s)</i> Arsham Asgharian Philip Hatziantoniou		<i>Supervisor</i> Nader Salessi, LED-Era, USA Karl-Erik Årzén, Dept. of Automatic Control, Lund University, Sweden (examiner)	
		<i>Sponsoring organization</i>	
<i>Title and subtitle</i> Implementing a ZigBee-Based Smart Lighting System			
<i>Abstract</i> <p>All industrial facilities and large warehouses require some sort of lighting solution. They have ongoing activities with workers and trucks moving around at various speed, and to avoid accidents these lighting solutions are often very bright and energy consuming. The most common solution is fluorescent light. These lights are energy consuming and not dimmable. Turning these off by night or when the facilities are empty makes their lifespan short, so they are usually always on. This ends up being very expensive and bad for the environment. By using LED as the main lighting solution it is possible to decrease the energy consumption to save money and help the environment. To decrease the figures even further there are smart LED lighting solutions. These solutions will detect ongoing activities and brightness levels in the facilities and automatically adapt the lighting needed in real time. There are many different solutions to smart lighting. Usually there are motion sensors and/or light sensors. These sensors will recognize any movement in the facility and increase the effect of lamps if more light is needed.</p> <p>The purpose of this master thesis is to add another sensor to give even more accuracy in light to facilities. Facilities mainly use trucks or other moving vehicles. This extra sensor is provided by attaching a ZigBee module to the lamps. The modules communicate with the moving vehicles and estimate their position to supply light wherever is needed. This provides an extra dimension to the smart light solution. Furthermore, using ZigBee one can detect a moving object close to the facility but not yet in the facility. This means that if a vehicle is moving towards a space that is not lit up, the LED will detect it and light up the space before the vehicle has entered, so the vehicle has appropriate amount of light when it arrives.</p>			
<i>Keywords</i>			
<i>Classification system and/or index terms (if any)</i>			
<i>Supplementary bibliographical information</i>			
<i>ISSN and key title</i> 0280-5316		<i>ISBN</i>	
<i>Language</i> English	<i>Number of pages</i> 1-64	<i>Recipient's notes</i>	
<i>Security classification</i>			