

# Abstract

1

---

This report for a bachelor project in electronic engineering at Aarhus School of Engineering, Aarhus University presents an explanation of the groups bachelor project: "Charcot-foot design and development of temperature measurement device". The purpose of the original project is to design and develop a system that can measure the temperature in different places on the foot of a patient suffering from charcot-foot. The modified version of the project entails designing a system with several sensors connected to a power line with overlaid communication. The focus is on developing the technology behind the power line communication bus.

A thorough documentation of the whole process, covering requirement specification, system architecture, design and implementation has been made. A series of internal tests are used as a baseline for improving the system.

The group has approached the project in an iterative manner in order to develop and improve the system. A SCRUM-like board has been used to structure the tasks and a time schedule was made to structure the different phases in the project.

A fully working proof of concept system has been made consisting of a central data unit and two sensor nodes. They communicate over the custom power line communication bus developed in this project. A chapter detailing the further development needed is included in this report.

The project has netted both group members a better understanding of hardware development as a whole.



# Resume 2

---

Denne rapport for et elektro bachelorproject på Ingeniørhøjskolen Aarhus, Aarhus Universitet præsenterer en gennemgang af gruppens bachelor project: "Charcot-fod design og udvikling af temperature-måle-device". Formålet med det originale projekt er at designe og udvikle et system der kan måle temperaturen på forskellige områder af fodden på en patient der lider af charcot-fod. Det modificerede projekt omhandler designet af et system med flere sensorer forbundet til en forsyningslinje hvorpå der også er kommunikation. Fokus er på at udvikle teknologien bag "power line communication bus", forsyningslinjekommunikationsbussen.

Der er blevet udfærdiget grundig dokumentation gennem hele forløbet der dækker over kravspecifikation, systemarkitektur samt design og implementering. En række af interne tests har lagt til grundlag for at forbedre system.

Gruppen har brugt en iterativ fremgangsmåde til at udvikle og forbedre systemet. En SCRUM-lignende tavle har været brugt til at planlægge og strukturere opgaver og en tidsplan har været brugt til at strukturere forskellige faser af projektet.

Et virkende "proof of concept" system bestående af en central data enhed (CDU) og to sensor noder er blevet lavet. Enhederne kommunikere ved hjælp af forsyningslinjekommunikationsbussen, som der er udviklet i dette projekt. Et kapitel der omhandler den videre udvikling af projektet er også forfattet i denne rapport.

Projektet har resulteret i at begge gruppemedlemmer har fået en bedre forståelse for udvikling af hardware.

# **Contents**

---

# Preliminary 3

---

This report is made by two electronics engineering students at the engineering college of Aarhus, Aarhus university. The report is the result of the bachelors project completed through spring of 2014. The report describes the general solution and execution of the project. Besides this report several documents and a physical product is developed.

It has been the intention of the group to learn and evolve as engineers. This includes using proper engineering development methods.

The group has had great emphasis on team work, project structure and planning.

The report is written with fellow engineers as target audience, therefore some assumptions may imply.

The group is very happy to have had the pleasure of having Preben Kidmose as Supervisor.



# Introduction 4

---

The goal of the project is to develop the technology to enable a product comprising of several sensors daisy chained on a power line with overlaid communication. This is an extension of the original project description. This report will describe the phases, methods and solutions made and developed throughout the project.

The project is build upon the requirements and descriptions made in the preliminary project. Some of which has been changed or modified. The preliminary project paper can be found on the enclosed CD [? ].

During the preliminary project a rough time schedule was composed, dividing the project into four phases.

- System Design
- Technology studies
- Development
- Documentation and finalizing

The overall purpose of the project is described in the course catalogue as these points:

- Use scientific research result and collect technical knowledge into a technical solution
- Develop new solutions
- Gather and evaluate new knowledge with regards to relevant engineering areas
- Complete routine engineering tasks.
- Communicate project results to professionals as well as customers
- Present project results orally using audio/visual communication tools
- Incorporate social, economic, environmental and health and safety consequences in a solution

## 4.1 Reading guide

The report should be seen as four sections, introduction, methods, results and conclusion. The introduction covers the specification of the project by project formulation, system description, requirement specification and project scope.

Methods explains the approach to the project and solutions, planning and scheduling, work distribution and project structure.

The results covers the result of the project namely what have been made and what have been learned. Most subjects here will be described briefly but further information can be

found in the more detailed descriptions in the documentation found on the CD. The conclusion rounds up the project and weighs of expectations and goals.

## 4.2 Glossary and abbreviations

This section contains all the abbreviations and terms used in this project. The full name will often be written the first time an abbreviation is encountered.

The abbreviations and terms are general for the entire project and might not be used in this documents.

Term/abbreviation	Definition
CDU	Abbreviation of Central data unit, a key component in the system.
SN	Abbreviation of Sensor node, a key component in the system.
LPF	Abbreviation of Low Pass Filter.
HPF	Abbreviation of High Pass Filter.
PLL	Abbreviation of Phase Lock Loop.
IC	Abbreviation of Integrated Circuit.
PCB	Abbreviation of Printed Circuit Board.
DLM	Abbreviation of Data Length Multiplier.
CRC	Abbreviation of Cyclic Redundancy Check.
ASIC	Abbreviation of Application specific integrated circuit.
PC / computer	General term for a computer / Personal Computer.
NRZ	Abbreviation for Non-return-to-zero .
PD	Abbreviation of Phase detector, a central part of a PLL.
VCO	Abbreviation of Voltage controlled oscillator, a key part of a PLL.
CDR	Abbreviation of Clock and data recovery.

# Project Formulation

5

---

*This chapter describes the project formulation forming the foundation for the project. The original project formulation was provided by DELTA and adopted by the group.*

## 5.1 Original project formulation

This is only a short excerpt of the original project formulation, for the entire original project description see the enclosed CD [? ].

### 5.1.1 Background

A small part of diabetic patients will develop Charcot-foot, a complication often introduced late in the illness. This complication immobilises the patient and reduces quality of life. It is believed that the charcot-foot is causing a rise in the temperature of the afflicted foot and that the heat signature is different from a healthy foot. Earlier projects have shown that temperature measurement can be used to evaluate on the condition of the foot in the rehabilitation phase.

### 5.1.2 Project description

The project should end in a prototype able to continually measure the temperature of the foot. The device should be worn by the patient throughout the daily life.

The project is expected to deliver a design of such a system, where focus should be on the temperature measurement device, although it is expected to describe the entire system.

Expected results:

- Prototype
- Requirement specification of prototype
- System description including interface descriptions

## 5.2 Modified project formulation

This project description is an extension of the original project description.

### 5.2.1 Background

The original project formulation was not fulfilling to the group with regards to new knowledge and technology. Therefore the project was modified to accommodate the groups needs. The change of the project has been made in collaboration with the supervisor.

The extension includes new communication protocol development and smart-sensor development.

Therefore this project will lead more towards a technology development rather than a product development.

### 5.2.2 Project description

The project should end in a prototype comprised of discrete component on a PCB with intentions of synthesizing it into an ASIC(for the sensor nodes). The project should make a proof of concept on a power line communication bus reducing wires needed for the system. The system will be comprised of a number of sensor nodes daisy chained to a central data unit. The central data unit will address the sensor nodes to extract the data and store it locally.

The modified system seeks to provide the technology for the desired system requested in original project description. The daisy chained sensors will limit the number of wires needed in the sock or shoe. The hypothesis that fewer wires cause less discomfort and makes it easier to build the end product is established.

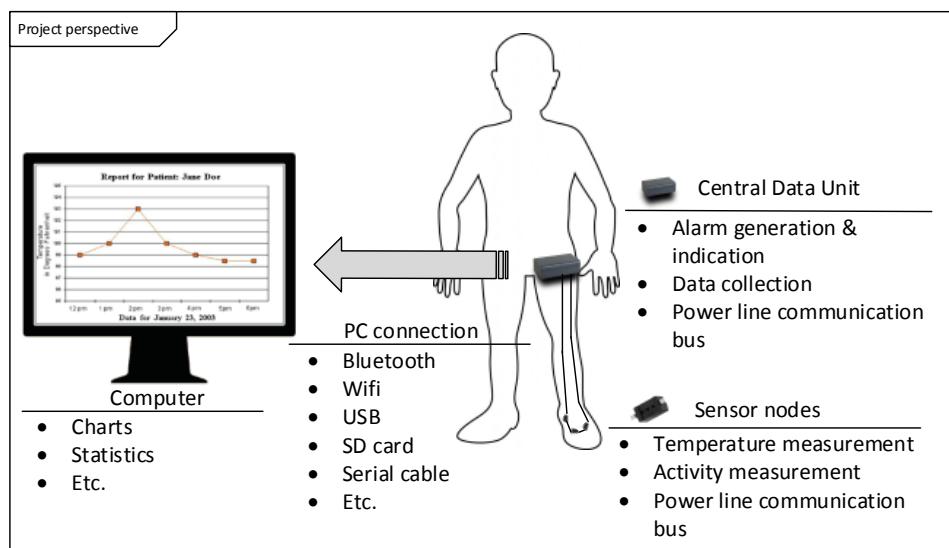
# System description 6

The description below describes the full system to give an overview and provide a project perspective.

The system is a monitoring system for patients likely to develop charcot foot or in the rehabilitation phase. The system should alert the patient if circumstances where injuries may occur by monitoring the foot temperature.

It comprises a main unit which is connected to a sensor daisy chain. The sensor daisy chain is mounted around the foot of the patient located in a sock or shoe. The main unit powers the entire chain of sensors as well as digitally extracting data from each sensor. The data is evaluated by the main unit to alert the patient if needed. The data is stored locally on the main unit for later extraction to a PC for statistics and analysis from professionals eg. doctors, nurses or specialists.

The sensor nodes are ASICs and will be covered with epoxy to accommodate the need for hygiene and practically make it possible to wash them. The size of the sensor nodes should be small to be of as little discomfort as possible. The sensor nodes should then be built into a sock or shoe.



**Figure 6.1.** System perspective



# Requirements specification

7

---

*The requirements specification is focused around the central data unit and the sensor node. It describes the functionality requirements along with other general requirements.*

## 7.1 General description

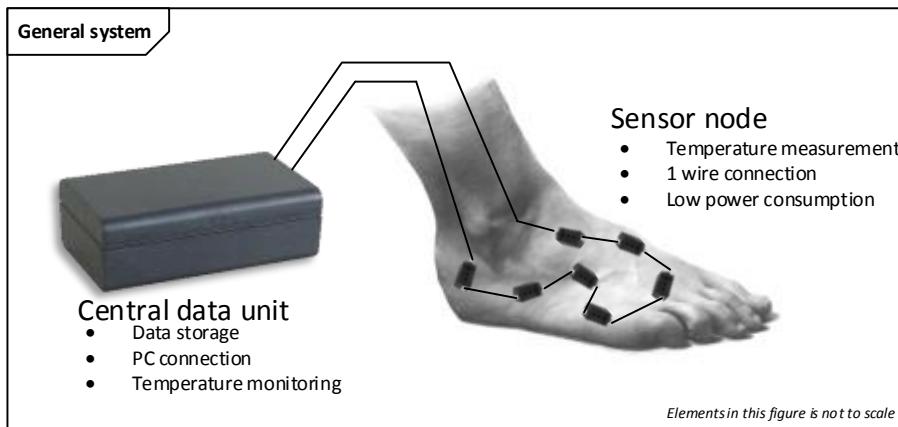
The system will be mounted on and around the foot of a person who is likely to develop charcot foot. The system comprises a Central data unit(CDU) and an arbitrary number of sensor nodes. The CDU serves as the main component and it has several main functionalities:

- Contact all sensor nodes within fixed time intervals
- Store data
- Evaluate data and warn patient if necessary
- Send data to PC for analysis

The CDU supplies a bus where all sensor nodes are interconnected in a daisy chain. The CDU continuously requests measurement data from each sensor node via the bus. The CDU will be powered by a rechargeable battery with enough power to run the system for atleast a day.

The data should be extractable from the CDU to a PC. The connection could be bluetooth, wireless, USB, UART or some other fourth option.

The system is shown on figure ?? below. The focus of the requirement specification will be upon the elements shown in this figure.



**Figure 7.1.** Project system

## 7.2 Functionality requirements

The system has two main functionalities:

1. Set system to normal operation
2. Extract data from CDU

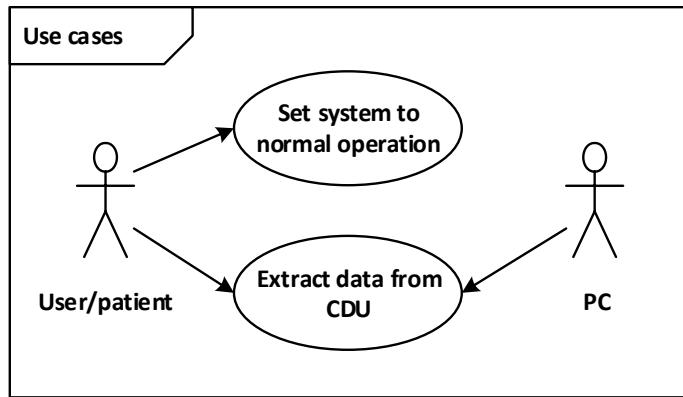
These use cases handles the expected normal functionality as seen from the end user. The users/actors of the system can be found in table ??.

Actor name	Type	Description
Patient	Primary	The patient sets the system to run and initiates data extraction from the CDU.
Computer	Secondary	The computer can extract data from the Central data unit. It is not intended as a control unit but mostly just for demonstration and debug purposes.

**Table 7.1.** Actors of the system

The use case set system to normal operation has the responsibility to power on the system and set it to normal operation mode. It is initiated by the patient actor.

The use case extract data from CDU has the responsibility to move the stored data from the CDU to a piece of computer software. It is initiated by the patient actor and has a computer as the secondary actor.



**Figure 7.2.** Use case diagram

For the use cases procedures see the requirements specification document which can be found on the CD [? ].

### 7.3 General requirements

The requirements stated in the specifications seeks to be non-constricting to avoid restricting design opportunities.

#### 7.3.1 Functionality requirements

- Cycle: 1 min

A cycle must include collecting data from every sensor connected to the bus. This can also be understood as the CDU must collect the overall temperature or other data every minute.

The data must be saved to memory which leads to the following requirement:

- Memory: 24 hours of data collection  
(~152kB) 256kB suggested.
- Real time clock: Yes
- Save entries must contain:
  - Sensor identifier.
  - Temperature.
  - Timestamp.
  - Type.
  - Errors.

A real time clock must be used in order to acquire the timestamp needed in the save entries. The type of the sensor should hard coded into the device.

### 7.3.2 Power consumption

In order to comply with the original idea of the system, a demand for low power consumption has been made. The power consumption requirements are based on rough estimates made by the group. The power consumption requirements are as follows:

- CDU Power consumption:  $<0.5\text{W}$
- Sensor node Power consumption:  $<0.05\text{W}$

The CDU power consumption requirement is excluding the sensor node power consumption.

### 7.3.3 Interface requirements

In the central data unit requirements part of the general requirement section in the requirements specification document it is stated that the system interfaces must be:

- Custom power line communication bus: Yes
- CDU to computer interface: Yes

The sensor nodes require the Custom power line communication bus as a system interface as well.

# Project scope 8

---

*This chapter describes the focus of the project development.*

The project will focus on a proof of concept build of a custom communications bus carrying power and communication. The solution should show the possibilities and enable for a smart sensor system.

The scope seeks to define the different components and features that the project will contain. The group will lay focus on the items described here:

- A central data unit (CDU)
  - Memory
  - PC interfacing
- Multiple sensor nodes
  - data acquisition
- Custom power line communication bus
  - Two way communication
  - One wire
  - Power
  - Multi-slave scheme
- PCB prototypes with discrete components

The CDU and the sensor nodes will require a communication bus to communicate. The custom power line communication bus will entail acquiring new knowledge and applying the knowledge to design hardware and software from scratch.



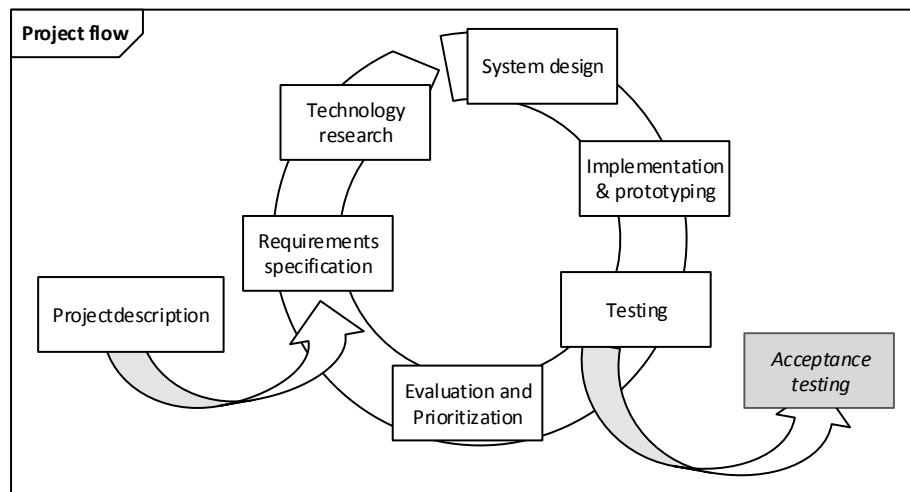
# Project execution

9

*This chapter contains the explanation of the groups working and development methods.*

## 9.1 Project approach

The group seeks to work iteratively to constantly improve on the architecture and design. The project flow is illustrated on the figure ?? below, which is also found in the preliminary project paper found on the enclosed CD [? ].



**Figure 9.1.** Project flow

This flow allows to redesign parts of the system which didn't work desirably. It also helps prioritize the parts that needs attention and downgrade phases of less importance or phases which has been fully formed.

An example of prioritization is the requirements specification phase which takes less and less time throughout the project because it simply isn't possible to keep making changes to it.

Another example is the prototyping part. Through multiple iterations there has been made a multitude of prototype print layouts. It is clear that the design and layout keeps improving.

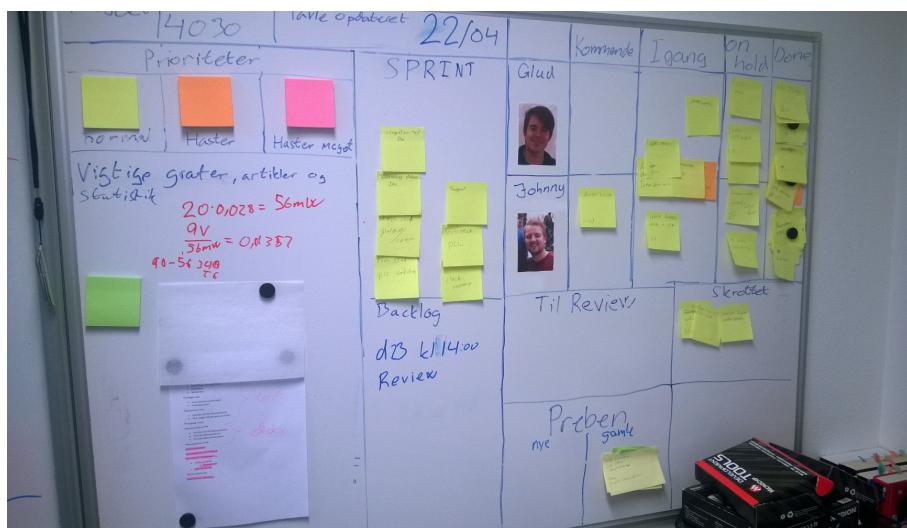
## 9.2 Work flow

The work flow of the project follows a scrum-like behaviour. The central part of this work flow is the project board. The board is used to structure work and progress. It is organized to give a good overview of the current tasks and upcoming tasks. A board meeting was held every work day with the following agenda:

- What did I accomplish yesterday?
  - What will I do today?
    - Todays "Must-Win" battles?
  - What obstacles are impeding my progress?
    - What will help me overcome this?

The agenda insures forward momentum and consistent structure in the meetings.

The tasks are equipped with an estimated time consumption to prevent time spend on tasks getting out of hand. If a task stretches over one or more days, the time estimate is updated at each meeting to reflect progress and the future time consumption. An example of how the project board is structured is found in figure ??.



*Figure 9.2.* Project board

Along with the daily meetings, supervisor meetings were held once every week. These meetings were built around the following agenda:

- Since last meeting
  - Present struggles/problems
  - Next weeks goals
  - Task related subjects
  - Closing remarks

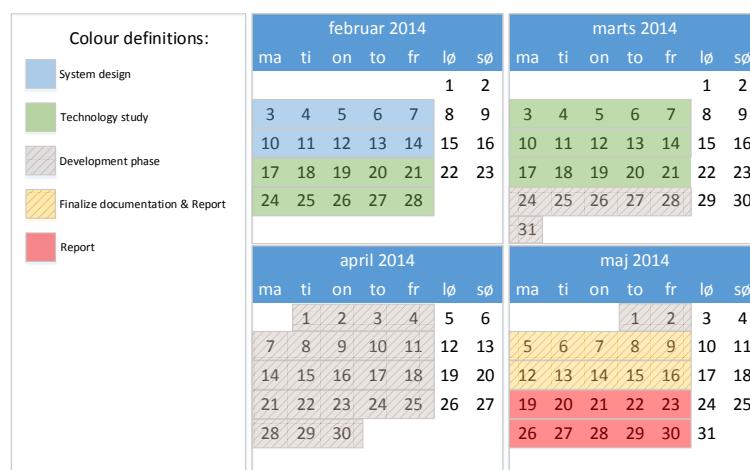
The supervisor meetings served to give both technical help as well as project guidance. These meetings enabled the supervisor to help the group move on if tasks seemed to be stuck or consuming too much time.

### 9.3 Project planning

The project is split into five phases. These phases serve as a linear progression from start to finish and plans which phases in the process to emphasize at what time in the project. The phases used are only describing the actual work in a very rough manner. The phases are:

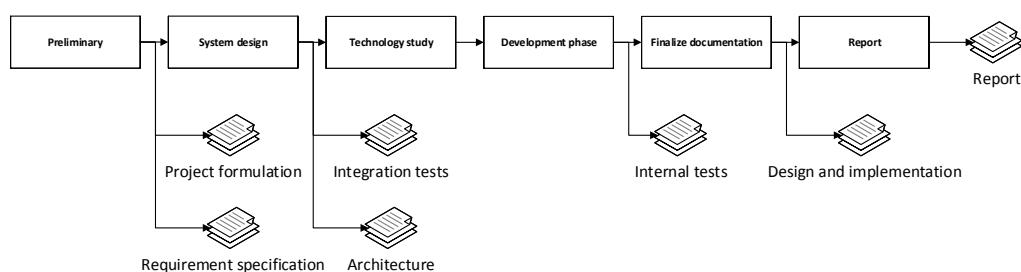
- System design
- Technology study
- Development
- Finalize documentation
- Report

The phases are used to make a rough time schedule throughout the project. In the time schedule each phase is given a specific duration. This duration reflects the extent of each phase and therefore how much time each phase required. The time schedule made in the preliminary project was modified a little during the project and the final time schedule is seen on figure ??.



**Figure 9.3.** Time Schedule

These phases gave base for the documentation needed for the project. Below is seen a document tree describing the relationship between documents and project phases.

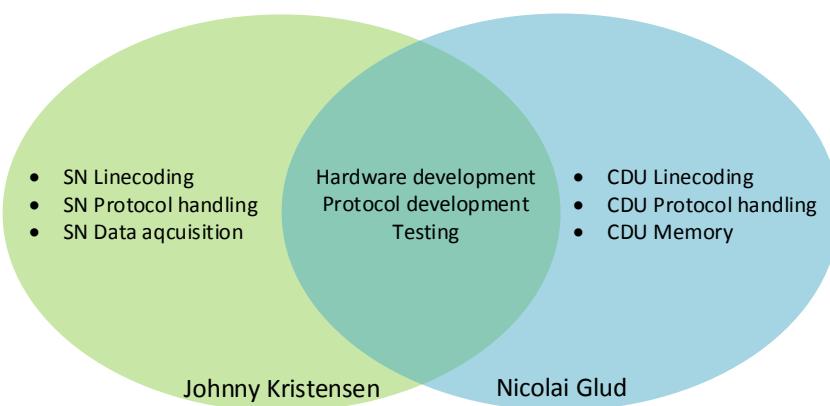


**Figure 9.4.** Document tree

Every artefact from this project has been subject to review atleast once. When a document was ready for review, it was through-read by each member of the group. A review meeting was held and a review document form was filled-out. The review document contained every information needed to correct the artefact under review. The review documents can be found on the enclosed CD [? ]. The meeting facilitator had the responsibility to import the changes to the artefact.

Together with the review scheme, revision control of all documents, hardware and software is held to maintain traceability throughout the project. Hardware print layouts were also reviewed and all of the revisions were kept in order to evaluate the progress made on the hardware.

Throughout the project, every module and piece of hardware has been made in collaboration. This means there is no clear cut division of the project that has a specific stakeholder. The only division is the CDU software and the sensor node logic. This means that one person alone was responsible for the part but not that the other person could not have fulfilled the same task. On figure ?? a venn diagram is shown to give a rough overview of the task/implementation distribution.



**Figure 9.5.** Workload/distribution venn diagram

# Technology studies

10

---

*The technology studies covers the studies leading up to the design and implementation. The technology studies help give an overview of how to realise the power line communication bus by providing good knowledge into the theory needed.*

During the technology studies the procedure were:

1. Discover literature
2. Create overview
3. Read and understand
4. Experiment

The studies below are partly based on previous knowledge from courses and experience and partly new knowledge and literature. The new literature can be found on the enclosed CD [? ].

## 10.1 Protocol and communications

### 10.1.1 Communications

The field of communication is very large, so to determine how to physically perform the communication in this system other methods were studied.

Communication systems use a lot of different ways of signaling data and it spans from frequency shift keying, phase shift keying, non-return to zero etc. Commonly used for communication is level shifting (NRZ [? ]), this is known from many common communication protocols such as RS232, I<sup>2</sup>C, SPI etc.

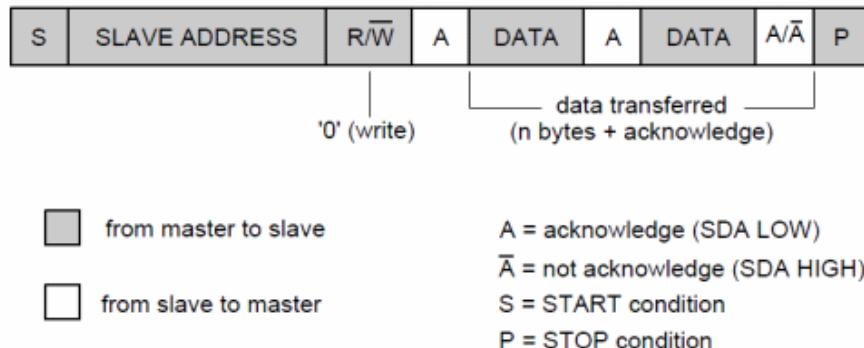
These physical protocol layers depend on synchronization either by including a clock or specifying a baudrate.

Another inspiration to the communication layer of the project is the HART industrial protocol [? ]. HART is a protocol overlaying the 4-20mA standard. The HART protocol transmit a logic 1 with a 2200Hz frequency and a logic 0 with a 1200Hz frequency. This enables filtering to separate the analog data and the digital data. Often times the transmitters are loop powered meaning they are powered by the 4-20mA loop also similar to this project.

### 10.1.2 Protocol

To determine how the protocol should be structured, parallels to known communication protocols were made. The bus with multiple slaves/nodes has a lot of similarities physically with the I<sup>2</sup>C protocol.

The I<sup>2</sup>C protocol is a two wire system with multiple slaves, which responds to requests from the master. An I<sup>2</sup>C transfer sequence is seen on the figure ???. The figure originates from the I<sup>2</sup>C protocol specification[? ].



**Figure 10.1.** I<sup>2</sup>C protocol transfer sequence

The transfer sequence is initiated by a start sequence from the master, this alerts the slaves and they are now listening on the bus. The next step for the master is to transmit the address of the slave. An addressing system must be utilized because several slaves are connected to the bus. Lastly the master tells whether it is a read or write command. The same principles are applied to the power line communication bus where an addressing system and start sequence is used.

Another known protocol is the modbus protocol[? ] which is a widely used industrial protocol. The modbus protocol has a start and addressing system, like I<sup>2</sup>C. However after the address a function code is transmitted. This code is used to tell the slave exactly what to do. The modbus frame format is shown on figure ??.

START	ADDRESS	FUNCTION	DATA	CRC CHECK	END
T1-T2-T3-T4	8 BITS	8 BITS	$n \times 8$ BITS	16 BITS	T1-T2-T3-T4

**Figure 10.2.** Modbus frame/header

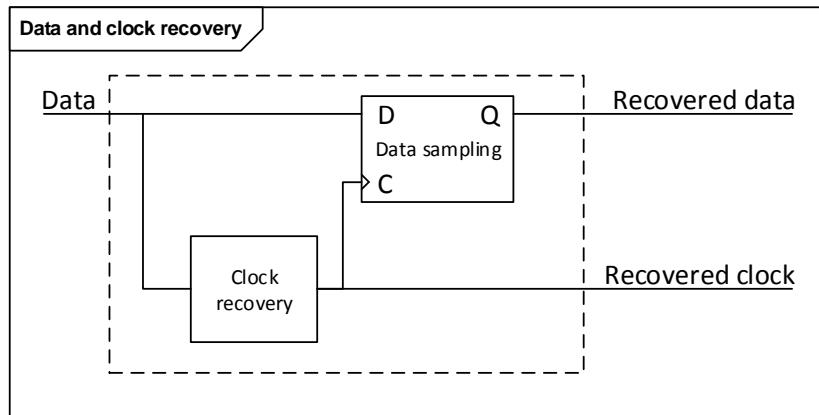
The function codes are standardized and covers a wide variety of uses, which covers all from simple request like, read discrete inputs, to diagnostic.

## 10.2 Clock and data recovery

Normally a data transmission consists of two elements, clock and data. That is not possible in this project and therefore the area of clock and data recovery was investigated.

Clock and data recovery is widely used in the field of data transmission. Especially in high bandwidth systems. Many of such systems use NRZ (Non-return to zero) line coding where in essence the clock is embedded in the data stream.

On figure ?? a conceptual block diagram of clock and data recovery is shown. From the recovered clock derived from the incoming data, the data is sampled. The clock recovery

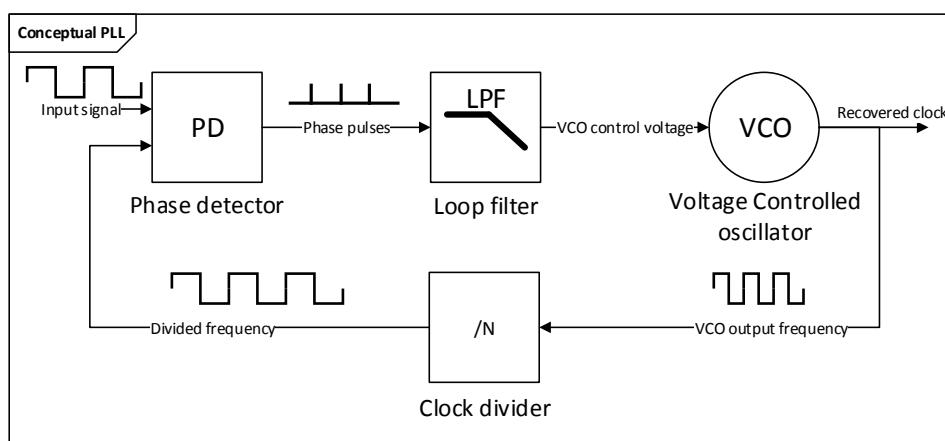


**Figure 10.3.** Conceptual clock and data recovery block diagram

can be achieved by a variation of different solutions but commonly used is the phase locked loop(PLL). The following subsection will describe the basics of the PLL loop used in this project.

### 10.2.1 The Phase-locked loop

The PLL comes in a large variety of designs but they all have one common conceptual diagram. A PLL consists of a phase-detector, a filter and a voltage controlled oscillator(VCO). Optionally it can contain a clock divider to speed up the VCO clock. The structure of the conceptual PLL is shown in figure ??.



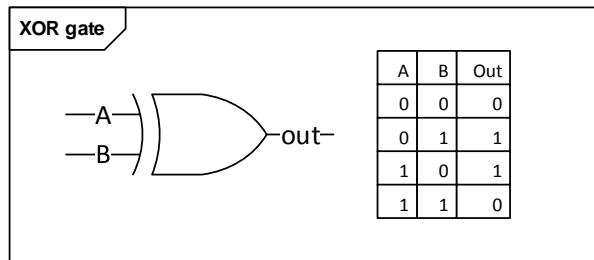
**Figure 10.4.** Conceptual PLL

### Phase detector and loop filter

The phase detector has two inputs. The input signal and the loop feedback signal. The phase detector compares the two input and produces a series of phase pulses depending on the phase difference of the two input signals. There is a variety of different phase detectors but two types are very common. They are named type 1 and type 2 phase detectors. Each type of detector has an impact on the loop.

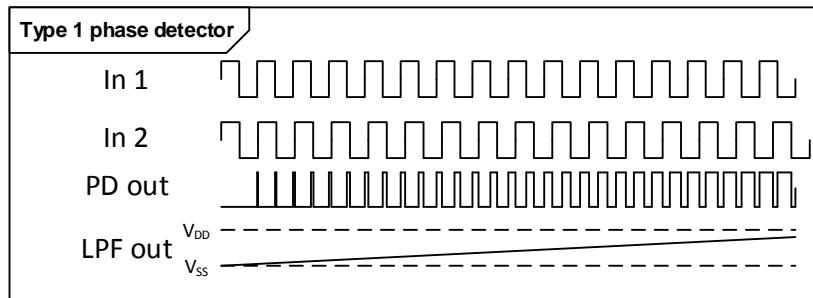
#### Type 1:

The type 1 phase detector is basically a XOR gate.



**Figure 10.5.** XOR gate

The XOR gate outputs a digital high when the two input signals are different from one another and a digital low when the two input signals are in the same state. Below on figure ?? is shown the waveforms produced by a type 1 phase detector.



**Figure 10.6.** Phase detector type 1 waveforms

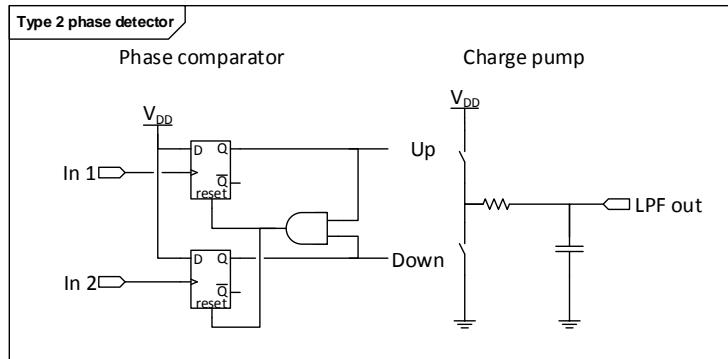
The two input signals are different in frequency. And on the PD out line it is shown how this changes with regards to the two inputs. The output of the low pass filter is used as an input to the VCO to either speed up the clock or slow it down.

When the PLL is locked onto the incoming frequency the LPF output will be  $\frac{V_{DD}}{2}$ . This effectively means that when the PLL locks onto the In 1 signal, the In 2 signal will be in exactly  $90^\circ$  deg phase.

#### Type 2:

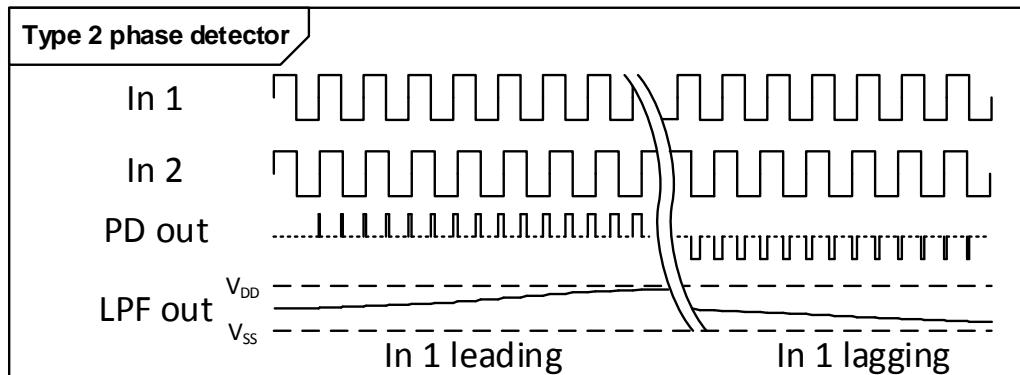
The type 2 phase detector comprises of a phase comparator and a charge pump. The

comparator is basically two D type flip-flops connected with an AND gate as seen on the figure ?? below. The output from the phase comparator controls the charge pump to either charge or discharge the filter.



**Figure 10.7.** Phase detector type 2 functional figure

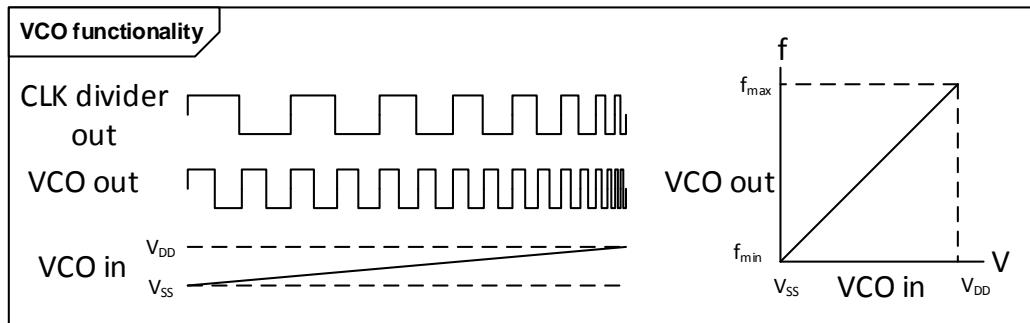
The advantage for the type 2 phase detector is the ability to detect a lead or a lag in frequency. Below on figure ??, the waveform is illustrated for both leading and lagging. Depending on the amount of lead or lag the pump will be on/off for longer or shorter periods of time.



**Figure 10.8.** Phase detector type 2 waveforms

### Voltage controlled oscillator and Clock divider

The voltage controlled oscillator serve to convert the voltage output of the LPF to an output frequency. The conceptual relationship between voltage input and frequency output is shown on figure ???. The frequency increases as the voltage input increases. For the PLL to operate properly it is important that the VCO frequency output related to the voltage input is as linear as possible. The slope of this relationship must be positive at any times to ensure stability.



**Figure 10.9.** Voltage controlled oscillator functionality

### 10.3 Outcome

Based upon the technology studies several choices were made.

The daisy chain of sensors will be constructed as a current loop to power the sensors, much like the scheme used in the HART protocol. The main unit can then control the current to modulate communication to the sensors. The sensors will be addressable and will all have a unique address. They will respond to the function code scheme and the necessary codes will be determined in the architecture.

The communication will be based upon a NRZ scheme. This allows for simplicity as well as a lesser need for computational power to for example FSK. The NRZ scheme will enable for clock recovery using general purpose PPLs and this clock can be used to drive the sensor nodes. An extension to the NRZ line coding is manchester coding [? ]. Although not covered above manchester coding ensures that the line is not low (or high) if a series of zeros (or ones) are transmitted. This helps to ensure a stable and synchronized data sampling clock from the PLL. The manchester coding will follow the E. G. Thomas definitions.

A minor part of the technology study was the study of the PIC24FJ  $\mu$ -controller. The PIC24FJ was new to the group and some new knowledge had to be obtained. This was made easy by the PIC24 Tutorial on the engscope website [? ].

# Project description

11

This chapter seeks to explain the design and implementation of the project along with the tests used for improving the system.

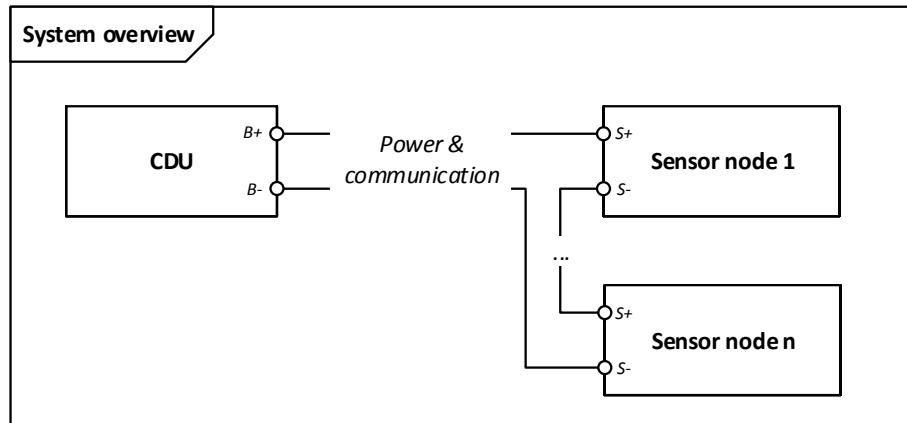
## 11.1 Architecture

The architecture of the system explains the conceptual design blocks and interfaces. The architecture incorporates the requirements into a system and describes the overall structure and functionality of the system.

### 11.1.1 General system description

The system consist of the CDU and a number of sensor nodes as seen on figure ???. The CDU is responsible for contacting the sensor nodes in order to collect and store the data that has been acquired by the sensor nodes. The sensor nodes sole responsibility is to acquire data.

The supply for the sensors nodes is taken from the custom power line communication bus. Communication signals between the CDU and sensor nodes are found on the bus as well.



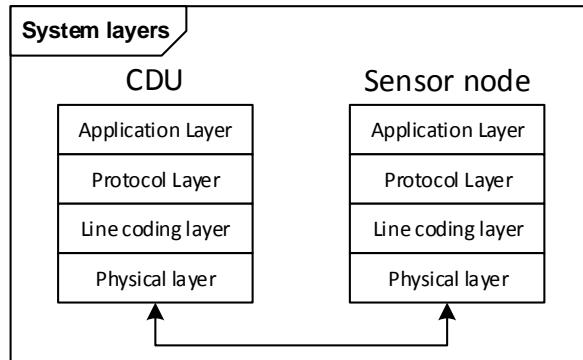
**Figure 11.1.** System overview

The general interface in the system is the bus. The sensor nodes are daisy chained. The B+ connected is connected to the S+ of the first sensor in the chain. The next sensor in the chain is then connected with S+ to the S- of the first sensor. The last sensor in the

chain is connected with S- to the B- on the CDU.

### 11.1.2 System layers

The CDU and the sensor nodes can be described as several layers with regards to communication as seen on figure ???. Each layer handles different abstraction layers of the functionality and communication.



*Figure 11.2.* System Layers

The model is inspired by the OSI model[?] and each layers responsibility is as follows:

- Application layer

Found in the application layer is the entire application. All functionality is handled here.

- Protocol layer

The message composed from the application layer is processed to comply with the protocol.

- Line coding layer

The line coding layer converts the protocol message to comply with the line code specified to the bus.

- Physical layer

The physical layer receives the line coded stream of data and applies it to the bus, to be received by either the CDU or the Sensor node.

These layers help simplify the structure of the communication and the responsibility of each layer is very clear.

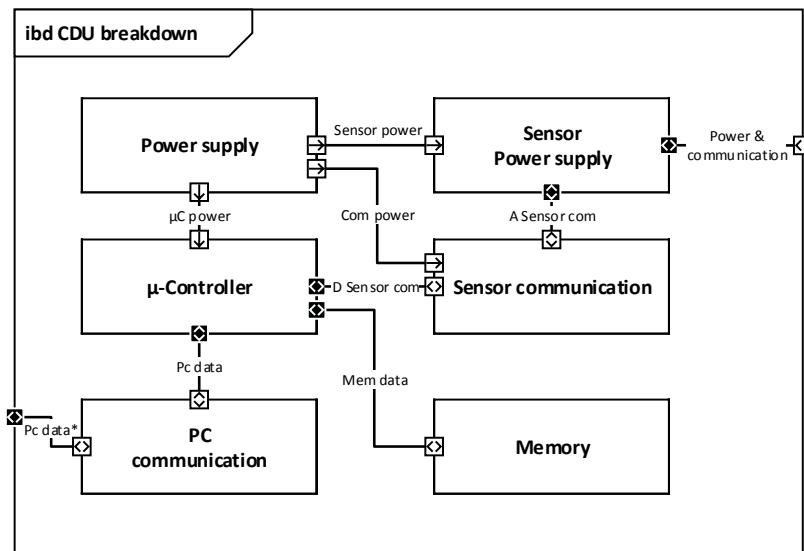
### 11.1.3 CDU conceptual architecture

Conceptual designs of the different parts of the system are made by taking the blocks from the system overview (figure ??) and creating internal block diagrams.

The internal block diagrams defines the internal structure of each block. These blocks should help make the functionality of each part possible and clearly define the boundaries

of the part.

The CDU is comprised of six conceptual blocks as seen on figure ??.



**Figure 11.3.** Internal Block Diagram of the CDU

Each block has a unique responsibility. This responsibility has to be fulfilled for the system to work.

- Power supply

The power supply delivers the correct voltage levels to all parts of the system. These voltage levels are not defined in the architecture but in the design and implementation.

- Sensor power supply

The sensor power supply delivers the power to the sensors. It receives a signal from the sensor communication which is modulated onto the bus. Together with the sensor communication block these handle the physical layer of the communication.

- $\mu$ -controller

The  $\mu$ -controller block handles the application layer, protocol layer and line coding layer of the system. It interfaces with the sensor communication block, memory and pc communication.

- Sensor communication

The sensor communication receives a digital stream of data from the  $\mu$ -controller and modulates it to the power supply.

- PC communication

The PC communication block is the physical block handling the level converting of a RS232 connection to a PC. The implementation of the RS232 protocol is handled by the  $\mu$ -controller.

- Memory

The memory block contains the physical memory and has a digital interface to the  $\mu$ -controller.

### 11.1.4 Sensor node conceptual architecture

A sensor nodes responsibility is to acquire data and respond to requests from the CDU. A sensor node can be divided into 4 conceptual blocks as seen on figure ??.

- Communication

The communication receives analog communication from the bus and converts it to digital logic levels. It also recovers the clock from the data stream

- Power supply

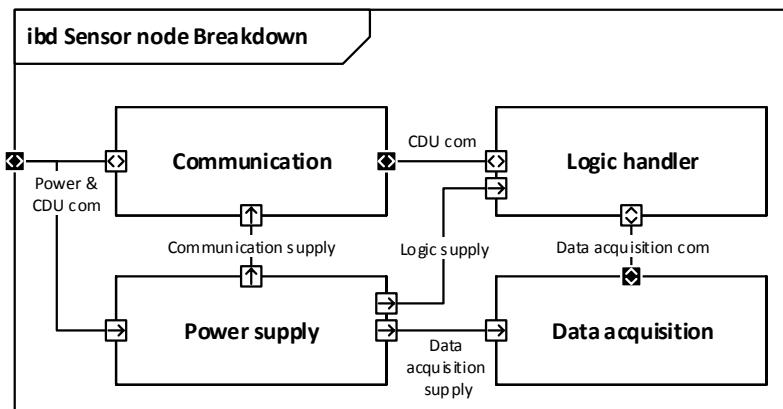
The power supply extracts power from the bus and supplies it to the internal circuitry.

- Logic handler

The logic handler is the main computational part of the sensor node. All line coding, protocol and application is handled here.

- Data acquisition

The data acquisition block serves to acquire physical data from the surroundings. The data acquisition block has a digital interface to the logic handler.



**Figure 11.4.** Internal Block Diagram of the sensor node

### 11.1.5 System communication

The protocol is defined in the following section. The bus, named custom power line communication bus, has a header as seen in table ???. The communication sequence is initiated by a start sequence. This sequence indicates the beginning of a transmission sequence. It is followed by an address and a function code. The address tells the sensor nodes which one is being contacted and the function code tells the sensor node how to react. The defined function codes can be found in table ???. An example could be the function code GetData. When a sensor node receives this function code it responds with the data measured in the sensor and errors if any occurred. The data length multiplier (DLM) defines how many bytes of data which is to be transmitted, and it can also be zero in which case no data is transmitted. This functionality is made to enable for variable message length so you don't waste time on sending dummy bytes etc. The data is then a multiple of the DLM in bytes. This could be the measured data from the sensor node

to the CDU or it could be calibration data to the sensor node etc. Lastly the CRC is a standard cyclic redundancy check. The CRC is used to determine if the data was correctly received. If this is not the case the message is discarded. The communication frame

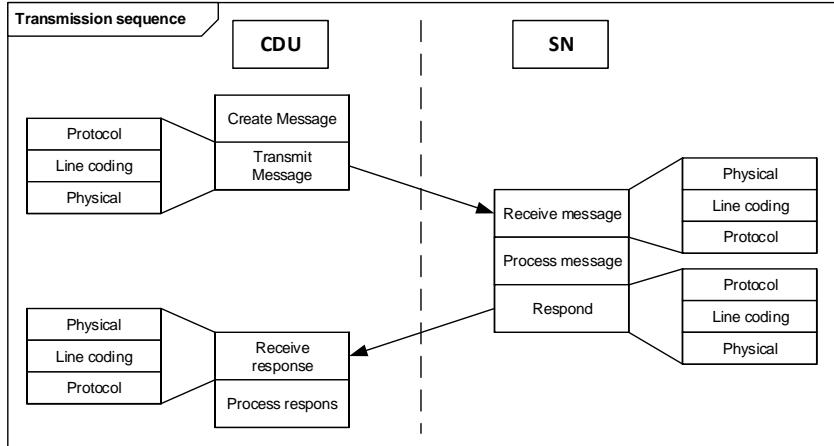
Function code	Name	Data
0001	GetInfo	Error codes and Type
0010	GetData	Error codes and 12 bit of data from sensor

**Table 11.1.** Function code table with names and data explanation

Start Sequence	Address	Function Code	DLM	Data	CRC
1 nibble	1 nibble	1 nibble	1 nibble	n bytes	1 byte

**Table 11.2.** Message format for writing and reading

is identical when transmitting or responding. Communication on the bus can be broken down to a series of steps. The CDU takes the initiative to write to a sensor node. It creates the message that will be sent and starts transmitting. The message is received on the sensor node and the sensor node starts processing the message. When it is finished processing it will respond according to the message. The response is received on the CDU where it will be processed. This transmission sequence can be seen on figure ???. The figure also shows the flow of information through the layers.



**Figure 11.5.** A single transmission

## 11.2 Design and implementation

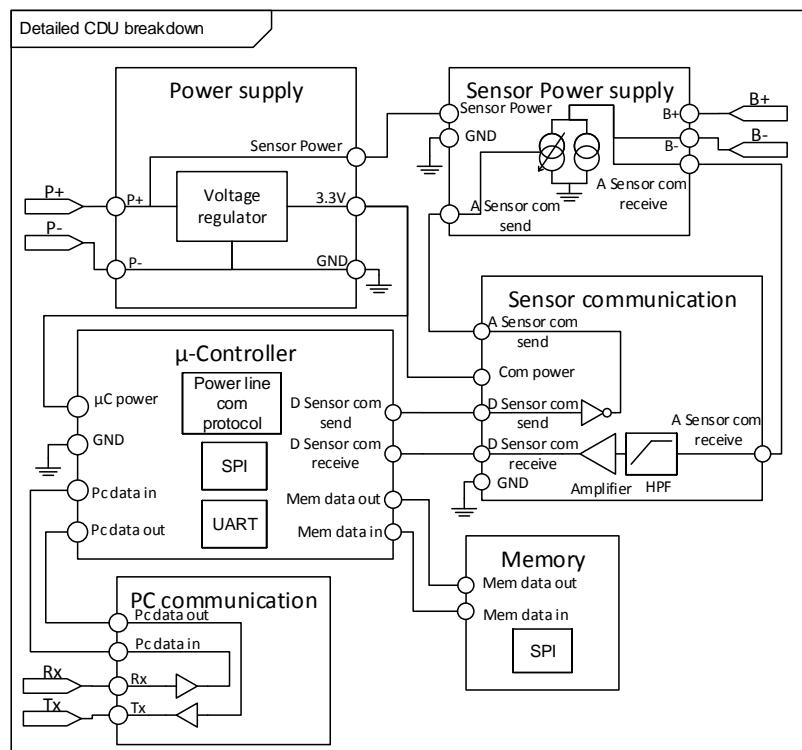
The design and implementation of the system is a naturally continuation of the technology study made in this project. The key part was integrating the acquired knowledge from the study and applying it to design and implement the prototype system.

### 11.2.1 Central data unit design

The blocks from the architecture document were incorporated in order to create a detailed breakdown of the different parts of the system. The CDU block breakdown can be found in figure ???. The figure seeks to explain the block in a conceptual manner. The blocks and components inside the different design blocks explains the design choices. The block also contains the different external interfaces.

For the CDU there are three external interfaces with two pins each. These pins belong to the power supply block, the sensor power supply block and the pc communication block. The power supply pins, P+ and P-, are used to power the whole system. The sensor power supply pins, B+ and B-, are a part of the communication bus. The sensors are daisy chained to these pins. The pc communication pins, Rx and Tx, are used for communicating with the PC using UART.

The brain of the CDU is the  $\mu$ -Controller block. As seen on the figure, it contains a power line communication protocol block, a spi block and an uart block. The power line communication protocol block handles the protocol and line coding layer with respect to figure ??.

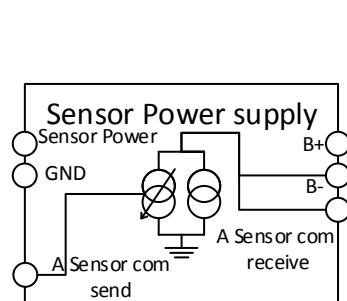


**Figure 11.6.** Detailed CDU breakdown

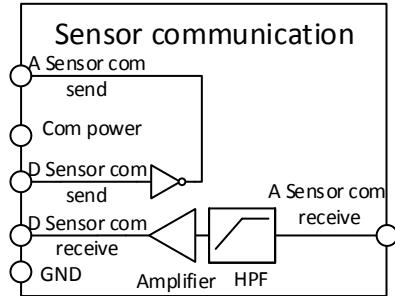
The design of the physical hardware layer was derived from the technology study of how a communication bus could be made. The central parts of the communication with regards to hardware is the conversion from digital signal to an analog signal and vice versa.

The conversion from digital levels to analog current levels is handled by two blocks in the CDU: The Sensor power supply block (figure ???) and the Sensor communication block

(figure ??).



**Figure 11.7.** Detailed CDU Sensor Power supply design.



**Figure 11.8.** Detailed CDU Sensor communication design.

As illustrated on the figure the  $\mu$ -controller block has a connection ending in a variable current generator coupled with a constant current generator. The constant current generator represents the current floor, meaning the lowest current to power the sensor nodes. The variable current generator is the communication current which is added on top of the current floor.

The receive functionality is designed as a filter and amplifier. The goal of which is to filter all DC and low frequency noise along with amplifying the received signal to the appropriate logic levels.

The physical layer of the PC communication is handled by the PC communication block. The block contains level converters which are used to achieve the communication levels needed by the UART connection to the PC.

The memory block contains a spi block. This coincides with the  $\mu$ -controller block also containing a spi block as spi is used for communicating with the block. The memory is divided into sections containing the data specified in the requirement specification. This can be seen in memory allocation subsection of the software section of the design and implementation document.

The CDU is powered by the power supply block that contains a voltage regulator. Some blocks on in figure ?? are implicitly supplied by the power supply.

### 11.2.2 Sensor node design

The sensor node breakdown can be found in figure ??.

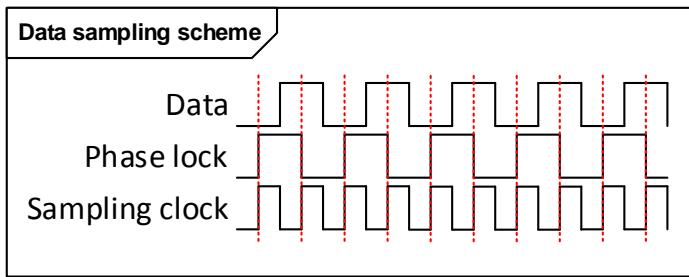
The power supply contains a voltage regulator. The voltage regulator converts the voltage drop from the sensor diode and the communication resistor to a steady 3.3V to supply the internal circuitry of the sensor node.

The communication resistor creates a small voltage drop change caused by the current modulated from the CDU. This voltage change is amplified to logic levels in the communication block and then sampled by the logic handler.

The communication block has two main functionalities, clock and data recovery (CDR) and respond functionality. The CDR serves to recover the clock from the incoming stream

of data so it can be sampled correctly. This is done firstly by amplifying the incoming data signal. This is done with an operational amplifier coupled as a comparator, where the reference voltage is an average of the communication signal.

The clock recovery is achieved by a PLL with the data signal as input. The output clock is double the frequency of the data clock. By doing this data should always be sampled on the rising edge of the clock. The sampling scheme is seen on figure ??.

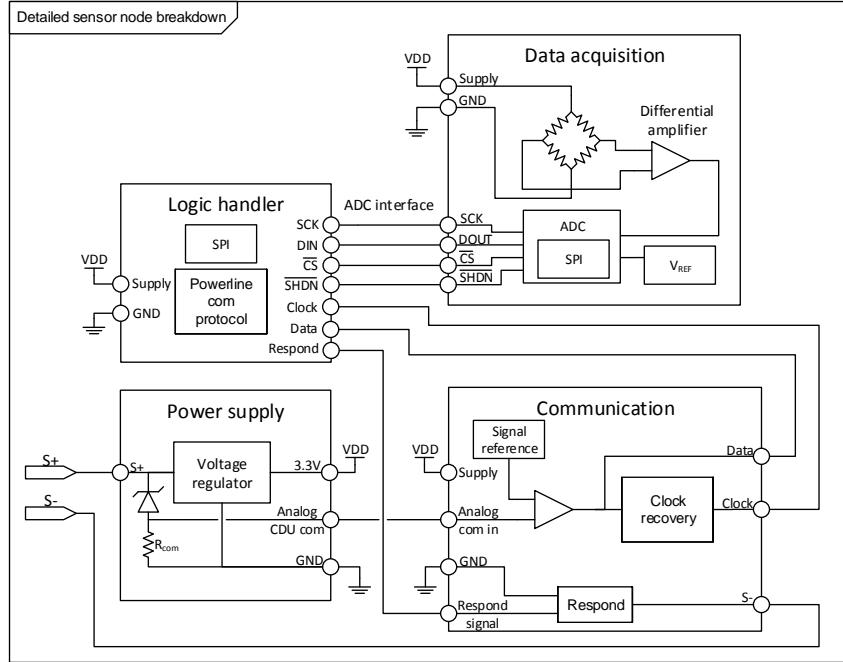


**Figure 11.9.** Data sampling scheme

When the PLL is in lock the phase of the loop clock will be  $90^\circ$ deg delayed with respect to the input signal.

The logic handler interfaces with the data acquisition block. This is done once every second to make sure new data is acquired before receiving a new request from the CDU. The logic handler also manages the power line communications protocol and line coding. It operates with the clock from the communication block.

The data acquisition block consists of two functionalities, a measurement wheatstone bridge modified to low current consumption and an ADC block. The logic handler interfaces with the ADC to acquire the measured data.



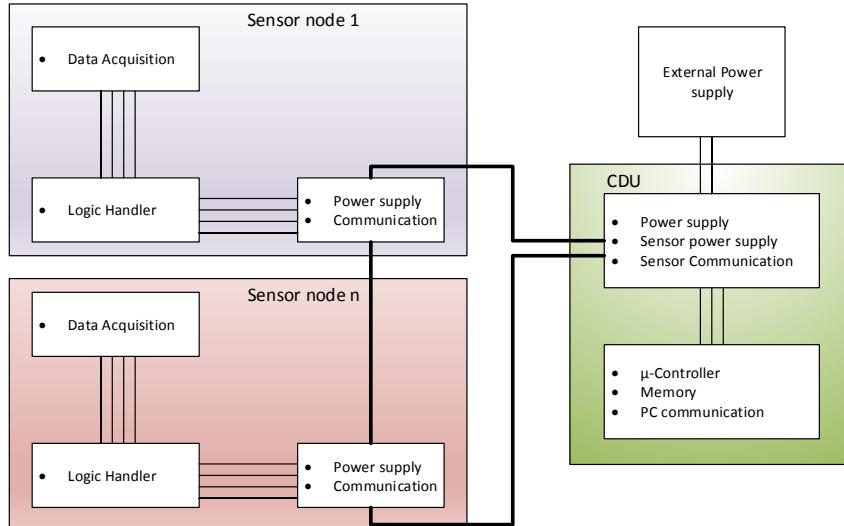
**Figure 11.10.** Detailed sensor node breakdown

Responding starts once the addressed sensor node has finished processing the message it has received. The digital level signal is converted to two different voltage levels by the communication block of the sensor node. The voltage levels on the bus is then directed from the sensor power supply of the CDU to the sensor communication block of CDU. The voltage levels will then be converted to digital logic levels and handled by the CDU.

### 11.3 Prototype

This section describes the end result of the prototype.

The development phase generated a lot of prototypes. The earliest versions of the prototypes were realised on breadboards, since this was a fast method of testing parts of the system. In these early iterations focus was more on finding the right design. As the design became more and more complete the magnitude of the implementation/realisation of the parts became larger and larger. The first prototype provided a basis for testing the system and improving on the design. With the improvements to the design, the hardware was routed and put on PCBs. The tightly coupled hardware parts were put on the same PCB. An overview of the prototype can be seen on figure ??, each white block represents a PCB (except for the external power supply).



**Figure 11.11.** Prototype system

The CDU (green box) is comprised of hardware made by the group which interfaces to a development board provided by Microchip Denmark. The board is an explorer 16 development kit. It contains a PIC24FJ family  $\mu$ -controller, eeprom memory, RS232 connection and in-circuit debugger. The CDU controls the power line communication bus. The levels on the bus are 20 mA when logic low and 24 mA when logic high. When the CDU communicates with the sensor nodes it uses the protocol and line coding layers to compose a message and the physical layer to transmit it. When not transmitting the CDU clocks with 10 kHz on the bus to ensure that the sensor nodes are locked in phase. The CDU is supplied with 15.5 V 29 mA by an external power supply.

The two sensor nodes (red and grey box) consists of two hardware PCBs made by the group. These represent the data acquisition block as well as the power line communication interface. The logic handler is a cyclone II fpga chip on a DE2 development board from Altera. These were available to lend from the school. The cyclone II chip is programmed to handle all application, protocol and line coding.

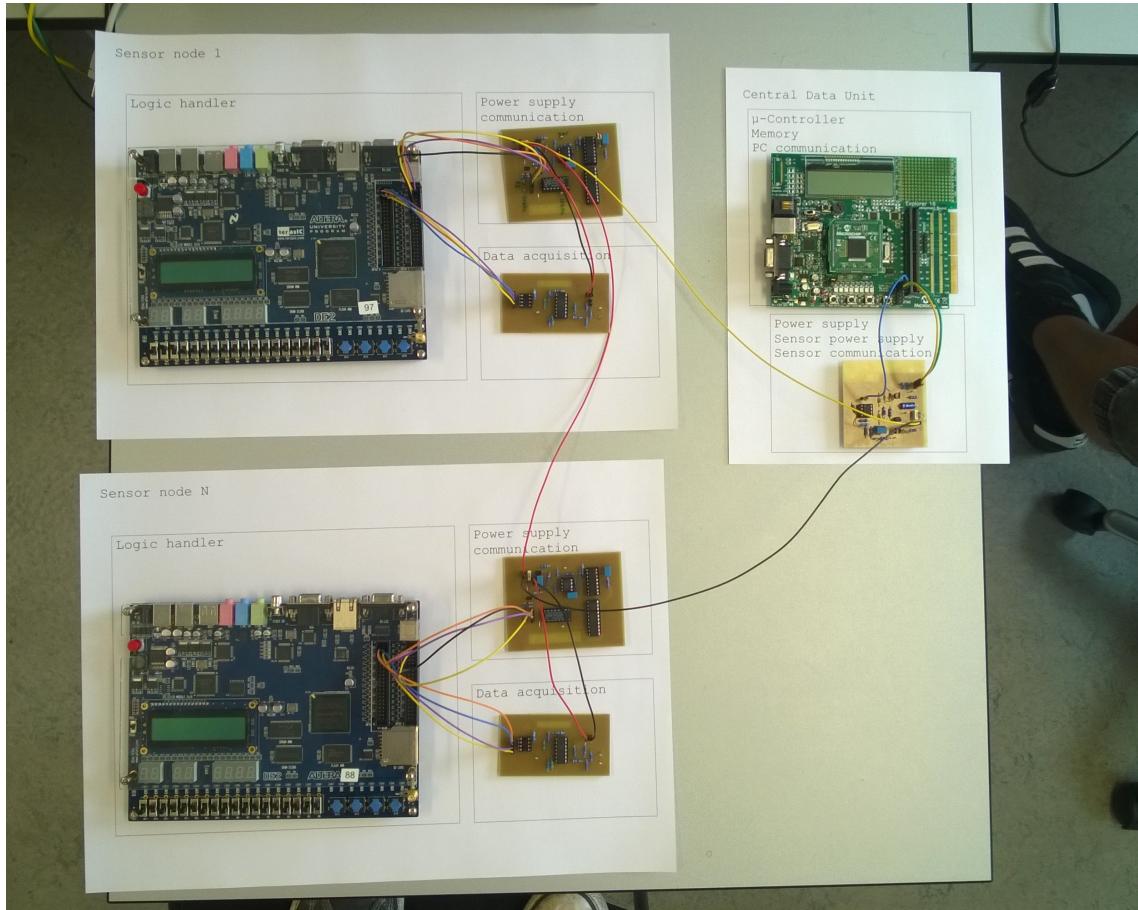
The current on the bus is read by a sensing resistor and amplified to 3.3 V. The messages are recovered using the 20 kHz output clock from the PLL to sample and convert the manchester encoded signal. The logic handler acts upon the message and responds to the CDU.

In addition to all the components found in the schematic, test points played a key part when debugging the circuits. These schematics have been routed and made into PCB's at the schools electronics laboratory.

For good practise all pin outs from both the explorer 16 and the DE2 board from each revision has been stored in a spreadsheet for an easy overview. The schematics for the CDU and the Sensor node can be found in the appendix. The CDU hardware schematic is seen on figure ???. This is the current iteration of the CDU and has the revision number 0.4.

The SN hardware schematic is seen on figure ???. The figure is displaying the current iteration of the two PCB's of the sensor node. The top schematic is the power supply, communication and clock recovery block. The bottom part is the data acquisition part of the sensor node comprising of a instrumental amplifier and an ADC with a reference voltage circuit.

The prototype (seen on picture ??) is connected as seen on figure ??.



**Figure 11.12.** Prototype

As seen on figure ?? the cdu boards are interconnected with 3 wires: digital send(Yellow), digital receive(Blue) and gnd(Green). Two sensor nodes are connected to the bus via the B+(Yellow) and B-(Black) wires.

An Altera DE2 board is connected to both sensor node prototype boards in the chain (Red wire). The connection wires are: Clock, Data, Respond and gnd. The data acquisition prototype board is powered by the sensor node prototype board with VCC(Red), gnd(Black). The data acquisition prototype board is connected to the DE2 board with the following wires: Chip select, SHDN, Data out and Clock. The DE2 board is powered by an external power supply.

## 11.4 Tests

*The tests section seeks to explain the methods, reasons and procedures for testing in this project.*

The internal tests in this project is made up of unit tests, integration tests and reliability tests. The unit tests seeks to test the internal functionality of each device while the integration tests are designed to test the communication in between devices.

The unit tests were made such that there were as few variable circumstances as possible. This means possible errors will most likely stem from the unit under test (uut). This principle of testing was also applied when integration testing.

The tests made in this project are by no means exhaustive. The tests made are suppose to provide a baseline for improving the system or otherwise prove a basic concept. As the outcome of this project is a technology and not an actual product there is no accept-test. The tests can be found in the internal test document on the enclosed CD [? ].

The content of a test can be explained on the basis of integration test case 1 which can be found in the Internal tests document in the appendix. Each test case contains the purpose of the tests, which tests equipment is used in the test, the procedure, the expected result and the actual result.

The purpose explains what is being tested. The test equipment used includes stubs of the units that are being tested. This could be a sensor stub needed for testing the CDU or a full sensor node tests stub used for testing communication. The procedure seeks to explain how to do the tests in a way such that a technician can perform the test without prior knowledge to the system. The expected result provides a range or a basis on which the actual result can be evaluated.

The full extent of tests are:

Integration test case	Goal:
1: Sensor getinfo request	To test the communication from CDU to Sensor node
2: Sensor getdata request	—  —
3: Sensor respond to getinfo	To test the communication from Sensor node to CDU
4: Sensor respond to getdata	—  —
5: Full transmission	To test the full communication scheme
6: Full transmission, unknown function code	To test the error responding scheme
7: PC sends getdata request to CDU	To test the communication with the PC
8: PC sends unknown request to CDU	To test the error in communication with the pc

**Table 11.3.** Integration test cases

Reliability test case	Goal:
1: Timed transmissions and errors	To test the reliability of the bus

**Table 11.4.** Reliability test cases

CDU unit test case	Goal:
1: 3.3 volt power supply	To test the power supply block
2: Communication to bus	To test the sensor communication block
3: Voltage reference in receiver circuit	—  —
4: Communication from bus	—  —
5: Test of function IntegerToBinary	Software test of function
6: Test of function PatMessage	—  —
7: Test of function ToManchester	—  —
8: Test of function InitSensorArray	—  —
9: Test of function CDUSend	—  —
10: Test of function CDUReceive	—  —
11: Test of function CDUReceive error handling	—  —

**Table 11.5.** CDU unit test cases

Sensor node unit test case	Goal:
1: 3.3 volt power supply	To test the power supply block
2: Data recovery circuit	To test the data recovery of the communication block
3: Phase lock circuit	To test the phase lock of the communication block
4: Respond circuit	To test the respond circuit of the communication block
5: CDU communication	To test line coding and protocol layers of the logic handler block
6: ADC interfacing	To test the interface of the ADC

**Table 11.6.** Sensor node unit test cases



# Project results 12

---

*This chapter seeks to present the results from this project.*

A system of a CDU and two sensor nodes has been made. The CDU can communicate with the sensors notes using the custom power line communication bus. The sensor nodes can respond to all the defined function codes and if the addressed sensor receives an unknown function code it will respond with an error.

The data from the sensor node is received by the CDU and stored in memory. This can be exported to a PC by requesting data from the CDU.

The sensor nodes can measure temperature.

## 12.1 CDU

The CDU prototype has memory, pc communication, sensor communication, sensor power supply and power supply fully implemented. The  $\mu$ -Controller used has the first iteration of the protocol and line-coding layer implemented. The  $\mu$ -controller, memory and pc communication are supplied by a power supply connected to the explorer 16 board. The prototype board is powered by an external power supply.

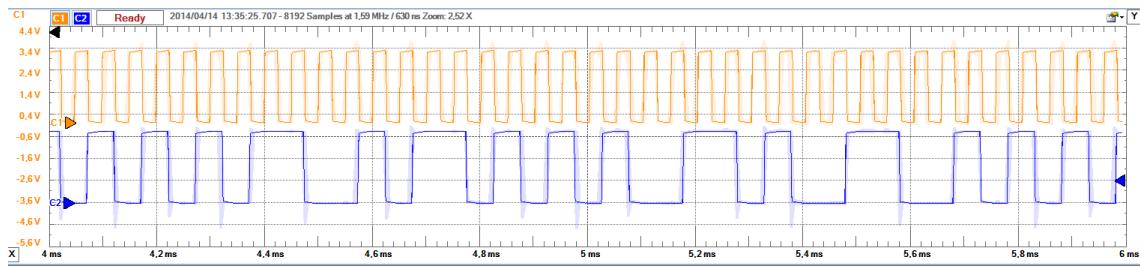
The software on the  $\mu$ -controller is written in C with the approximate size of 6kB (4% of total memory). The functionality currently implemented consists of communicating with the sensor nodes using the custom power line communication bus, saving data to- and loading data from memory and responding to requests from the PC.

## 12.2 Sensor node

The sensor node prototype has power supply, logic handler and communication fully implemented. The data acquisition has been implemented with a variable resistor as sensor. The logic handler is powered by a power supply connected to the DE2 board. The sensor node prototype is powered by the communication bus.

The "software" for the DE2 board is written in VHDL. It utilises less than 1% of the total gates in the FPGA chip embedded on the board.

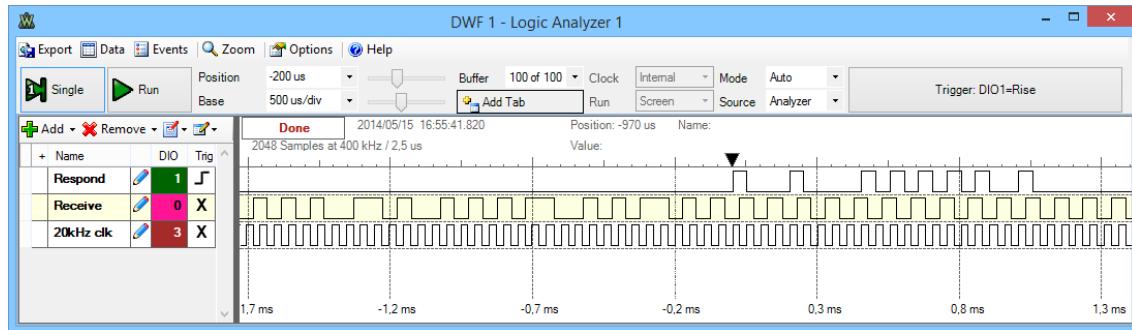
On figure ?? is shown a scope picture of the clock and data recovery. The orange line is the recovered data sampling clock and the blue is the recovered data. It also shows that the data sampling clock is in phase as desired.



**Figure 12.1.** Clock and data recovery scope picture

### 12.3 Custom power line communication bus

The custom power line communication bus has been fully implemented on the CDU and the sensor nodes. Two-way communication has been achieved with multiple slaves connected with one wire through the sensors. On figure ?? is shown a picture of a successful transmission sequence captured by a logic analyser. The transmission is captured on the sensor node and shows the incoming data, the recovered clock and the respond data. In the picture the CDU has requested GetInfo with the function code 1 on address 1. The sensor node responds with the address, the requested function code, no errors and the switch input states.



**Figure 12.2.** Logic analyser picture of a successful transmission sequence

### 12.4 Test results

The tests in this project provided a baseline for improving the system. The reliability test result showed that less than 1 in 7200 transmissions resulted in an error. The integration tests in the internal test document are all done on the first iteration of the protocol.

# **Further development**

**13**

---

*This chapter briefly suggests elements and thoughts of the further development needed to finalize the development of the product. The order of sections below is random and does not suggest any prioritization.*

## **13.1 Component selection**

The whole project was made with discrete components that were available from the schools electronics laboratory. This means that the prototype was developed with components that were not the most fitting or optimized for the project. To further improve the project some components will have to be changed for cheaper or more effective components or circuits. An example could be improving the operational amplifier handling the incoming data recovery. An amplifier with a better slew rate, a better signal-noise ratio or a wider gain-bandwidth product. This might improve stability in the data transmission.

## **13.2 Supplying the unit**

A full battery supply system will need to be designed. It must take the total power consumption over time in consideration in order to comply with the requirement specified in the original project formulation. The voltage to the bus must also be taken into consideration, see section ??.

## **13.3 IC synthesizing**

A big part of the further development is the design and synthesis of an ASIC. A feasibility study might be useful in order to ensure this is the best solution for the project and if it is even possible.

## **13.4 Application layer**

Another part is the application layer of the CDU. The functions and descriptions have been developed but the application will need to be written and optimised for this project to become an actual product. Further development on the CDU could be the ability to maintain the firmware.

### 13.5 Data acquisition

The measurement system will also have to be improved. Currently the measurement system isn't calibrated nor is the measurement range specified by any supporting data. Therefore this might have to be redesigned as it currently is a part of the proof of concept.

### 13.6 Design optimization

Currently the voltage supply level on the sensor nodes is 3.3V. The operating voltage must be reduced to make it practically possible to have a lot of sensors daisy chained. The group suggests a voltage maximum of about 2V for a sensor node. This enables for a lot more sensors than now or a lower voltage level to supply the bus. It also helps the battery power extension since the need for a high voltage can be inefficient.

Another design optimisation is the choice of  $\mu$ -Controller to be used in the CDU. The current software only utilises about 4% of the program memory and 7 (of a total of 85) of the general purpose pins. A smaller and cheaper  $\mu$ -Controller will most certainly fit the project.

# **Obtained experience**

**14**

---

## **14.1 Personal experience**

Working on this project has given the members of the group a good understanding of hardware components and how to utilise off the shelf components to create designs and solutions. The project has been far more in depth hardware development than any previous project, and has given us great experience with actual hardware development. This has also shown all the littlest things can intervene and disturb. Even though we thought the current loop was very immune to radiated noise, we found that it was the source for a lot of our problems.

Thereby hardware debugging has been a big part of the development phase and has helped to give a natural understanding of where to measure, redesign and rethink.

## **14.2 Technology studies experience**

Previous project have had little or next to no technology studies involved. This phase has helped give a better approach when searching for new technologies and utilizing them in a project. As it is also seen in the calendar schedule this phase takes a lot of time. This is part finding the technology needed and part understanding the technology and how to use it.

## **14.3 Development issues**

Throughout the project some hurdles and issues appeared and as expected these slow down the development. These issues were discussed at both the daily meeting and the weekly supervisor meeting depending on magnitude. Most issues were discussed on a conceptual level. If this did not help resolve the issue, help and knowledge was obtained from fellow students, teachers or researchers located at the school.

A problem with the simulation tool multisim was among these issues. While developing the power supply scheme throughout the system we didn't understand why the hardware design didn't work as expected. We found that the program couldn't handle the different voltage references supplied to various components.

## **14.4 Development approach**

The approach to development have changed from slowly dipping the feet into the water to rapid prototyping with almost new print layouts every week. This can be contributed to improved confidence in hardware design.



# Conclusion 15

---

The development methods used in this project led to a clear progression through the phases. Multiple iterations meant the group had to evaluate the current state of the solution and think of improvements to the design. The time schedule was modified during the project to better reflect the most important phase at that stage. There is no clear division of the project work as the group members have worked closely together.

In relation to the project scope the group has achieved a working central data collection unit referenced as the CDU. The CDU has a working PC interface. Multiple sensor nodes have been made with a functioning measurement system. The sensor nodes are powered by the bus. The two components of the system can successfully communicate using the custom power line communication bus. The group has achieved two way communication.

With regards to the requirements specification the group has achieved the ability to save data entries correctly in memory. The requirement for power consumption will be a part of the further development phase along with the optimisation of the design.

The whole system has been build as a prototype on PCBs with discrete components. In addition to a PCB, the CDU uses an Explorer 16 development board. The sensor node utilises an Altera DE2 board.

The technology developed in this project is a communication system built on top of a current loop. It serves as a proof of concept that it is possible to create a bus with a single wire daisy chaining multiple sensor nodes. The bus is stable and reliable.



# Appendix 16

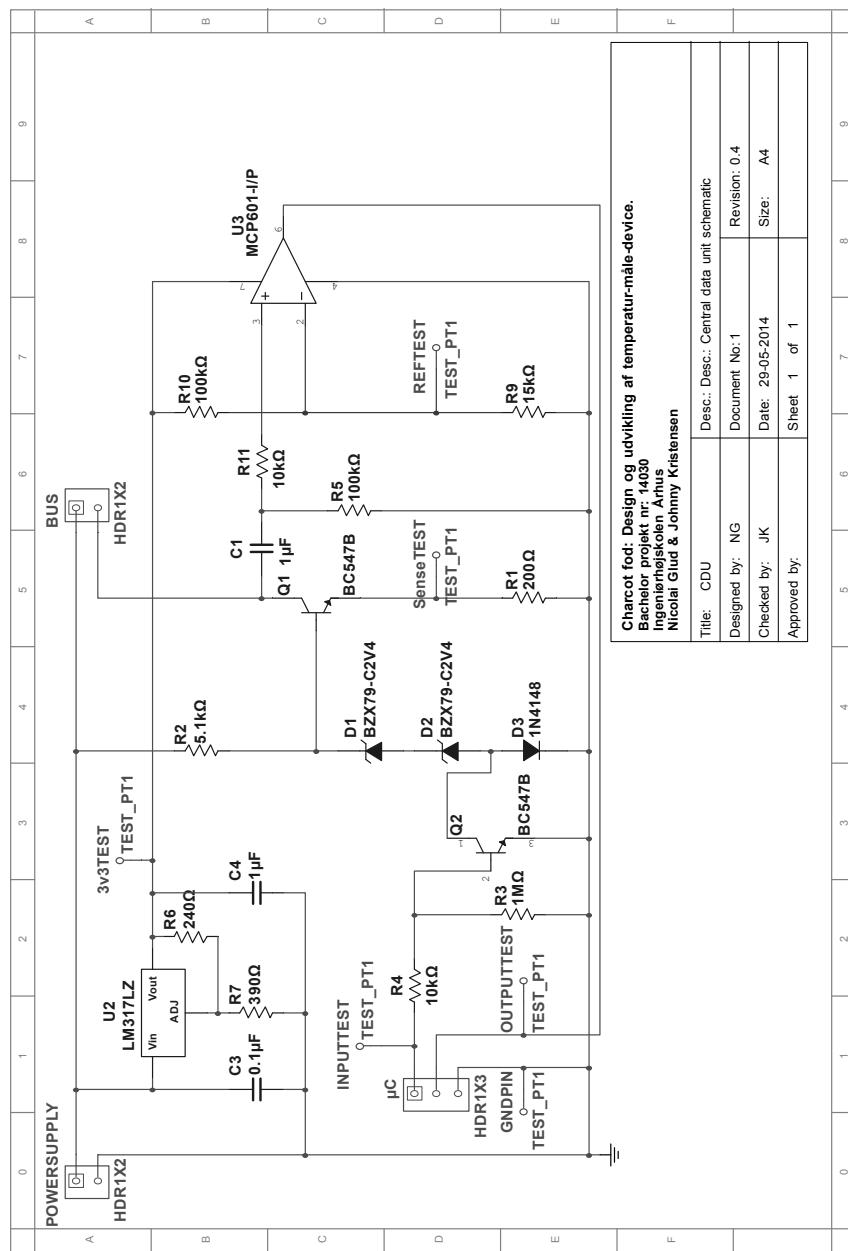


Figure 16.1. CDU hardware schematic

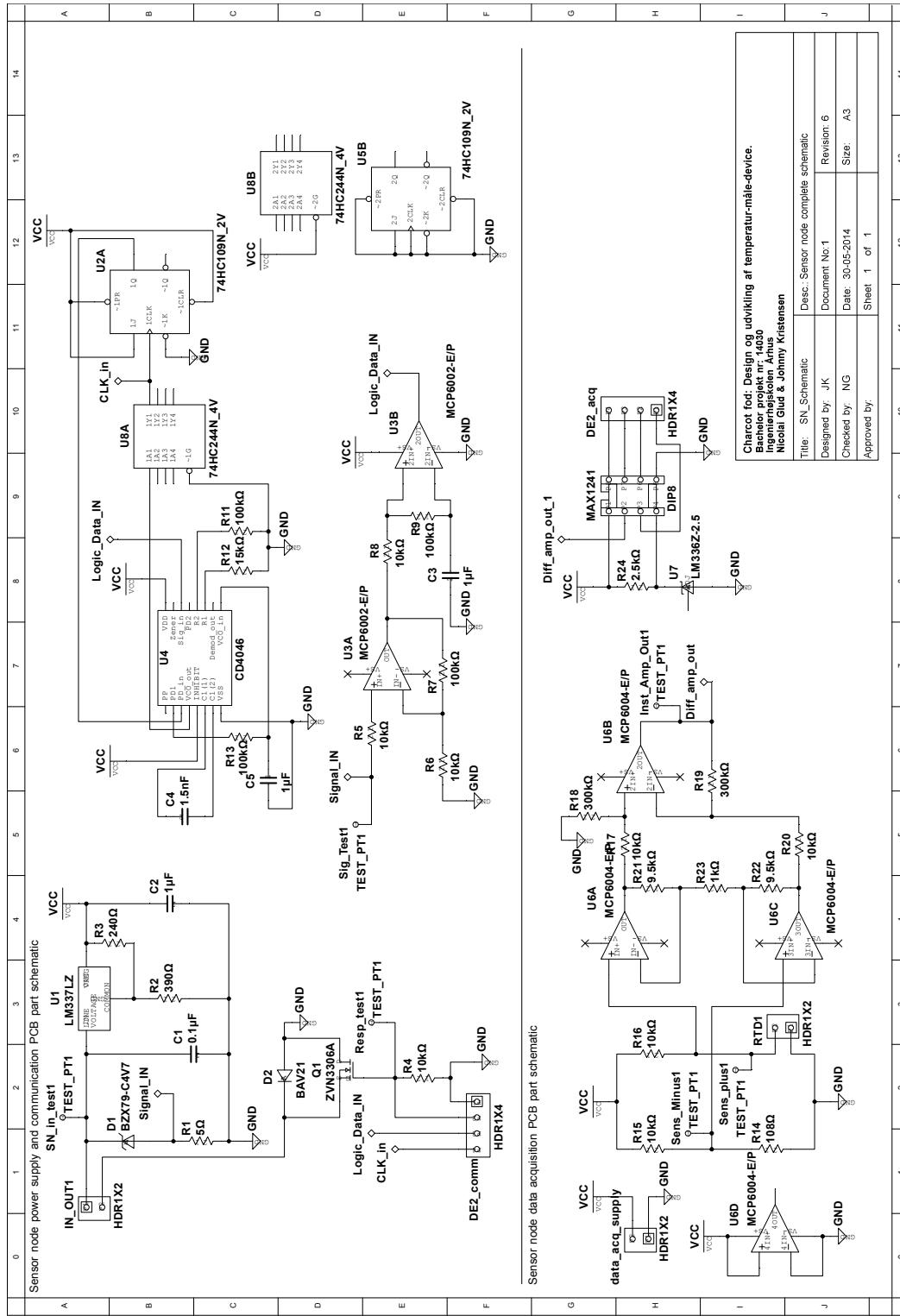


Figure 16.2. SN hardware schematic

# **Material on CD** 17

---

## **17.1 A - Report**

Report.pdf

## **17.2 B - Documentation**

Architecture.pdf

Design and implementation.pdf

Internal tests.pdf

opgavebeskrivelse.pdf

Preliminary project paper.pdf

Requirements specification.pdf

## **17.3 C - Litterature**

Clock and data recovery circuits.pdf

Clock-data recovery PLL using half-frequency Clock.pdf

Designing bang-bang PLLs for clock and data recovery in Serial Data Transmission Systems.pdf

Phase-locked loops with applications.pdf

PLL\_intro\_594a\_s05.pdf

slide regarding clockrecovery.pdf

Theoretical modeling and simulation of phase-locked loop for clock data recovery.pdf

## **17.4 D - Datasheets**

BC547.pdf

BZX79.pdf

LM317LZ.pdf

MCP601.pdf

PIC24FJ128GA010.pdf

74HC109.pdf

74hc244.pdf

cd4046b.pdf

LM336-2.5.pdf  
MAX1241.pdf  
MCP6002.pdf  
pt100.pdf  
ZVN3306A.pdf

## 17.5 E - Printlayouts

CDUPCB.pdf  
SN\_data\_acq.pdf  
SN\_ps\_com.pdf

## 17.6 F - Code

### **CDU:**

CDU.c.pdf  
CDU.h.pdf  
eeprom.c.pdf  
eeprom.h.pdf  
lcd.c.pdf  
lcd.h.pdf  
LineCoding.c.pdf  
LineCoding.h.pdf  
main.c.pdf  
MemCtl.c.pdf  
MemCtl.h.pdf  
spi.c.pdf  
spi.h.pdf  
uart.c.pdf  
uart.h.pdf

### **Sensor node:**

Data\_Acquisition\_interface.pdf  
Power\_line\_communication.pdf  
SN\_VHDL\_PROGRAM.qar

## 17.7 G - Tests

testcase7result.txt

## 17.8 H - Miscellaneous

### **Reviews:**

Architecture review document.pdf

design and implementation review document.pdf

Generic review document.pdf

REQ SPEC REVIEW.pdf