LumiNet 2

An Organic Interactive Illumination and Sensor Network for Fashion

Diploma Thesis at the
Media Computing Group
Prof. Dr. Jan Borchers
Computer Science Department
RWTH Aachen University

by
Mariana Bocoi

Thesis advisor:
Prof. Dr. Jan Borchers

Second examiner:
Prof. Dr. -Ing. Stefan Kowalewski

Registration date: Nov 4th, 2011
Submission date: May 18th, 2012

I hereby declare that I have created this work completely on my own and used no other sources or tools than the ones listed, and that I have marked any citations accordingly.

Hiermit versichere ich, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Aachen, May 2012
Mariana Bocoi

Contents

List of Figures

List of Tables

Abstract

Wearable computing is a growing field of study but there are aesthetic and ergonomic properties to be considered in order to make wearable computing attractive to a wider audience. Fashion designers can best understand and address the requirements for creating clothing that is comfortable and at the same time aesthetically pleasing.

Fashion designers often use technology in their creations, especially displays embedded in clothes. Thus the usability of current systems for fashion designers is a relevant issue. The main question that this thesis is answering is what characteristics the hardware and software should have in order to be easily employed by fashion designers.

During the interactions with the designers, we noticed that they had problems writing the code needed for the design. Using an iPhone app makes it possible to create visual input that will be converted by the system into code and sent to the nodes. The higher level abstraction of the hardware introduces new possibilities of further development and studies.

To address the hardware needs, we propose a small and cost efficient solution for distributed systems on clothes. The system is created as a smart pixel, to be compatible with fabrics, and easy to understand and use. With that, we hope to observe more complex projects emerging using the multiple interconnected entities.

Überblick

Wearable Computing ist ein wachsender Forschungsbereich, aber es gibt ästhetische und ergonomische Eigenschaften, die es zu berücksichtigen gilt, wenn man Wearable Computing für ein breiteres Publikum attraktiv machen möchte. Fashion Designer sind diejenigen, die es verstehen und die es vermitteln können, wie man Kleidung derart gestaltet, dass sie zugleich komfortabel und geschmackvoll ist.

Modedesigner verwenden oft Technologien in den Kreationen, insbesondere in Kleidung eingebettete Anzeigeelemente, daher ist die Nutzbarkeit der bestehenden Systeme ein relevantes Thema. Diese Thesis will deswegen die Frage beantworten, welche Besonderheiten die Soft und Hardware aufweisen müssen, um von Fashion Designern leicht benutzt werden zu können.

In der Zusammenarbeit mit den Designern haben wir festgestellt, dass sie Probleme haben, den für das Konzept nötige Quelltext zu schreiben. Eine iPhone Applikation macht es möglich, eine visuelle Vorgabe so zu übersetzen, dass sie an das System übertragen werden kann. Die architekturelle Abstraktion der Hardware führt neue Möglichkeiten der Forschung und Entwicklung ein.

Was die Bedürfnisse der Hardware angeht, schlagen wir eine kleine und kostengünstige Lösung für verteilte Systeme auf Kleidung vor. Das System ist als intelligenter Pixel konzipiert, damit es zugleich einfach in Kleidung integriert werden kann und einfach zu benutzen ist. Hiermit hoffen wir in Zukunft weitere komplexere Projekte aufkommen zu sehen, die die Idee der vernetzten Einheiten aufgreifen.

Acknowledgements

" Do you realize if it weren't for Edison we'd be watching TV by candlelight?"

—Al Boliska

I would like to thank my thesis advisors Prof. Dr. Jan Borchers and Prof. Dr.-Ing. Stefan Kowalewski for their guidance and assistance. Special thanks to René Bohne for his invaluable guidance and competent advice throughout the project.

I greatly appreciate the help and insightful comments I received from Clio Kakoulli, Alex Popa, Manuel Dejonge, David Peters, Swapnil Shah and Hendrik Kolvenbach during the development of the project. I would also like to thank those who participated in the user studies for their feedback and recommendations. Also many thanks to Sophie Stoner and Larisa Katz for giving me an insight into fashion.

Many thanks to my sister, who always believed in me. I would also like to thank my parents for everything.

Conventions

Throughout this thesis we use the following conventions.

Text conventions

Definitions of technical terms or short excursus are set off in coloured boxes.

Excursus:
Excursus are detailed discussions of a particular point in a book, usually in an appendix, or digressions in a written text.

Definition:
Excursus

Source code and implementation symbols are written in typewriter-style text.

```
matrix.scrollStringLeft("Hello World");
```

The whole thesis is written in American English.

The plural "we" will be used throughout this thesis instead of the singular "I", even when referring to work that was primarily or solely done by the author.

The generic "she" will be used where the person's gender is undefined.

Chapter 1

Introduction

"Ubiquitous computing names the third wave in computing, just now beginning. First were mainframes, each shared by lots of people. Now we are in the personal computing era, person and machine staring uneasily at each other across the desktop. Next comes ubiquitous computing, or the age of calm technology, when technology recedes into the background of our lives."

—Mark Weiser

The way we interact with technology has drastically changed over the years. The definition of a computer has changed from a room filled with hardware and cables to hand held smart devices in a matter of a few decades. There is a trend of merging technology with our day to day objects, making them smart and ubiquitous.

Organic User Interfaces (OUIs) are a part of ubiquitous computing. According to Holman and Vertegaal, an OUI is a flexible computer interface where the input blends in with the output. Because the main characteristics of OUIs are flexibility and user satisfaction, they inspire users to be creative rather than merely productive. [HV08]

1.1 Wearable Computing

Wearable computing is also part of the ubiquitous computing universe. Wearable computing is a very general term implying any technology that can be worn on the body. We will refer to wearable computing as in the following definition.

Definition:
Wearable Computing

WEARABLE COMPUTING:
Wearable computing is the study or practice of inventing, designing, building, or using miniature body-borne computational and sensory devices. Wearable computers may be worn under, over, or in clothing, or may also be themselves clothes (i.e. "Smart Clothing" [Man96]). [Man12]

In "Design for Wearability" Gemperle et al. describe wearability and the considerations and principles necessary for the design of wearable products. Wearability is defined as the interaction between the human body and the wearable object [GKS+98]. We will use the term integrability in clothes, that refers to the ease with which hardware can be attached and concealed in garments.

Some wearable technologies are also OUIs and blend into the fabric of day to day life. At the same time they display artistic patterns or text meant to inform and inspire. Over the past years there are constant appearances of wearable displays in fashion, we will show some of them in the chapter 2.3—"Wearable LED Displays".

1.1.1 Open Source Hardware (OSH) and Do It Yourself (DIY)

There are two different directions in wearable computing, professional systems and open systems. The first consists of closed systems that are not open to the public. The second follows the Open Source Hardware(OSH) philosophy where the hardware is transparent and openly available to the public. OSH is like publishing a cookbook, the hardware is not given away for free, only the recipe on how to make it out of other existing materials [Wei08].

While the knowledge of how things are created is available at our fingertips, the Do It Yourself (DIY) trend is growing in parallel with the OSH. DIY refers to making, changing or customizing something independently rather than trying to find the most fitting version in the market[LB10]. Frameworks and environments designed for novices were created in response. These made something like creating a shirt that communicates with your inbox accessible to the general public. Arduino [Ard05] is one of the more popular frameworks because it is straight forward, simple and has a great community that drives it.

1.1.2 DIY Wearables

Leah Buechley realized that current boards lack basic wearability and started exploring new flat designs and published several papers on fabric printed circuit boards (PCBs) [Bue06]. Her research gave way to a new DIY hardware board based on the Arduino platform called the Arduino LilyPad [BECC08] . Besides LilyPad other more popular frameworks are Sparkle [Spa05] and Fabrickit [DP10]. The possibility of users enhancing their everyday garments increased considerably with the introduction of DIY boards designed for clothes.

1.1.3 Fashion and Wearable Technology

The social aspects of wearable computing are addressed in the paper "A Short Note on the Social Consequences of Wearables" [KP01]. It debates introducing wearable technology to the "common man" and tackles several key issues for acceptability. The appearance and ergonomics of the clothing are features that play an important role in wether wearables would be widely used or not.

The aesthetics is where the fashion designers come into the picture. Engineers are good at creating hardware and software that is functional and efficient but often do not pay enough attention to the ergonomics or the visual aspect of the end product. Engineers and fashion designers can collaborate to use their expertise in their respective fields and try to create a harmonious solution keeping in mind the possibilities and limitations.

We will exemplify in chapter 2—"Related Work" some projects where fashion designers successfully collaborated with technical people and found a good balance between aesthetics and function.

1.2 Distributed Computing

We frequently turn to the complex natural systems for our inspiration where we find many interconnected systems and networks that distribute the work. However, looking at the current available options in DIY wearable technology, we notice that there are not many distributed systems.

Definition:
Distributed System

DISTRIBUTED SYSTEM:
A distributed system is a collection of independent computers that appear to the users of the system as a single computer. [Tan99]

LumiNet 1 [BBH09] distinguishes itself from the other frameworks as one of the few distributed systems for clothes. It has the advantage that it can form complex networks and also the high speed with which many boards can be reprogramed at the same time.

1.3 Objective

We propose a distributed system that is designed for beginners and created to be integrated in clothes. We want to give more power to the user in order to distribute and scale their designs while keeping it accessible also for smaller projects.

LumiNet 1 further developed the idea of 'smart pixel' introduced by BlinkM[KK06] by creating a distributed system able to communicate and create complex patterns and, to some extent sense the surrounding environment. The downside is that not many understood the complexity of the system. LumiNet 2 proposes to increase the visibility of the concepts introduced by LumiNet 1[BBH09] through moving them on a higher layer, closer to the user. As proof of concept we will present an application that controls an LED display built on top of the LumiNet 2 base framework.

1.4 Thesis Overview

The thesis is organized as follows. Chapter 2—"Related Work" provides an overview of different physical frameworks and their advantages, our view on some wearable computing projects in more depth, ending with a table comparing the frameworks.

Chapter 3—"System Design" lists the project requirements followed by a description of the system. Then each system component is described in detail.

In chapter 4—"Evaluation", the final prototype is evaluated based on the previously specified requirements. We also present studies that show the evolution of LumiNet.

The thesis concludes with a summary in chapter 5—"Summary and future work", followed by a quick look onto future work.

Chapter 2

Related Work

"Study the past, if you would divine the future."

—Confucius

This chapter starts with an overview of the different existent physical frameworks that have been used for wearables.We also present different designers that created clothes with LED displays also mobile devices that use interactive displays on clothes as a medium of expression. At the end of each section we compare the presented setups to each other. Finally the integration of the electronics in the clothes is presented along with an architecture that links Arduinos with Android phones.

2.1 Physical Computing Frameworks

This section presents frameworks that either played an important role in the development of wearable frameworks. Some could be added to clothes but were designed for a different purpose.

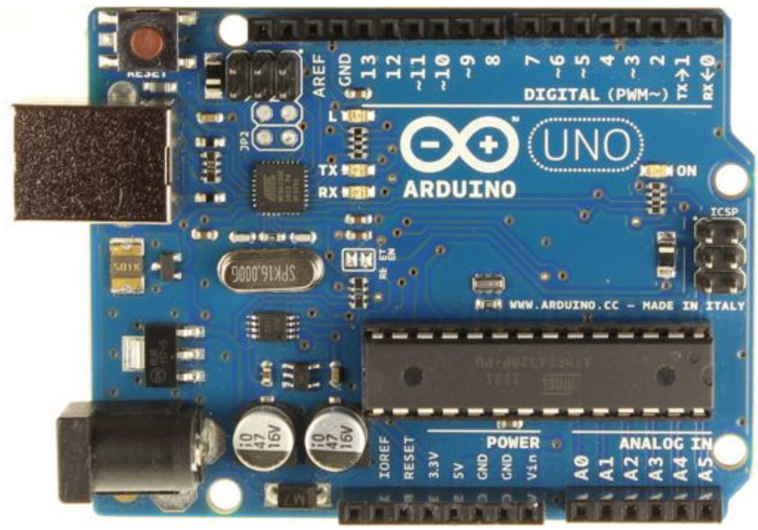


Figure 2.1: Arduino Uno.

2.1.1 Arduino

Arduino is a physical computing platform that emerged from the need to have a cheap OSH platform to study electronics. It was developed by educators and students at the Ivrea Interaction Design Institute in 2005 [Ard05]. The initial board evolved into a complete environment, it is hardware, a graphical user interface for programing and also a community.

The hardware (see figure 2.1[1]) is straight forward, it pulls out the pins of the micro-controller and has an USB and power connectors built in. For processing power it uses an Atmel AtMega 328 that operates at 16MHz.

To program the board, the users do not need to learn much to get immediate results. A multi-platform integrated development environment (IDE) is provided (see figure 2.2) to program a sketches into it. Sketches are programs written in a combination of the Processing and C languages. The Arduino community provides extensive documentation for the language and also for the hardware.

[1] picture credits to http://www.arduino.cc/

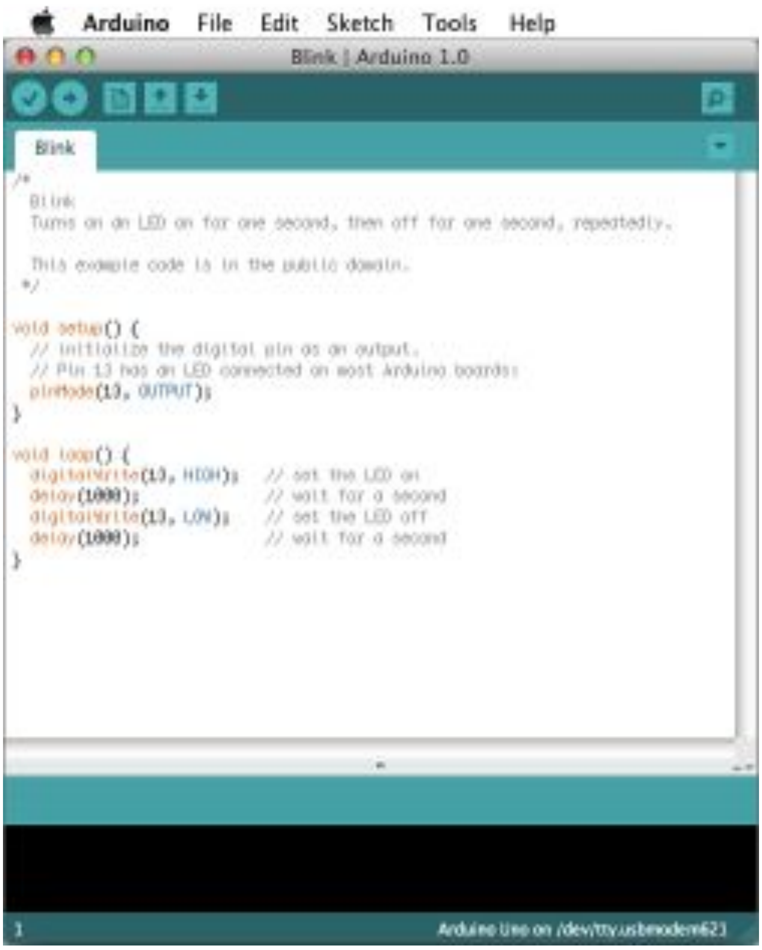


Figure 2.2: Arduino IDE.

Because the system it is so easy to understand many people have adopted the framework. Thus, the community grew, contributing with constant testing and feedback, innovative ideas of how to use the board and new add-ons called shields that add new features. The shield types are varied, some add WiFi capabilities, others can control motors, there are over 288 commercial shields [2] and many more shields that are created by users. Arduino has arguably become the standard open source platform for physical prototyping [HVSS12].

[2] Taken from http://shieldlist.org/, the data was valid at the time the thesis was written.

2.1.2 BlinkM

BlinkM [KK06] was designed as a smart LED designed for interaction designers, industrial designers and artists. It was designed to be inexpensive, robust and to offer just enough capabilities to be easy to work with out of the box, while still remaining open-ended [Kun08].

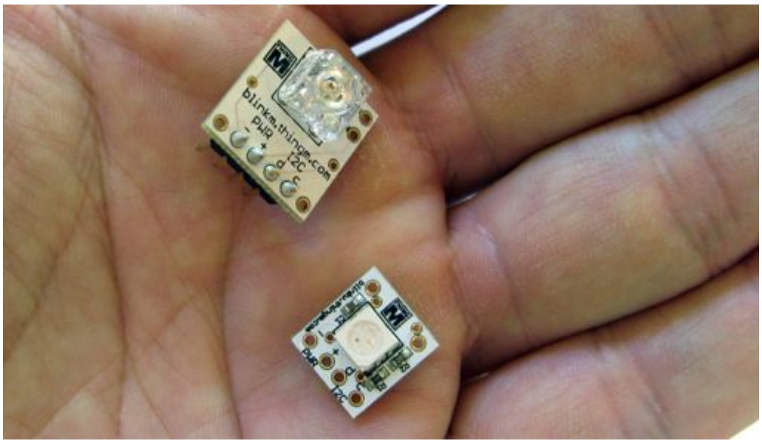


Figure 2.3: BlinkM and BlinkM MinM.

On the nodes the hardware is minimalistic, it has an AtTiny 45S micro-controller, 3 resistors, an LED, and a capacitor. These are the bare minimum components that should be combined in order to drive an LED. The board comes in several sizes, BlinkM MinM being the smallest one (see figure 2.3[3]).

Unlike Arduino, BlinkM is programmed with a visual IDE (see figure 2.4). With it frames can be composed by composing a sequence of colors that is loaded on the board. It has a very clean layout with a clear purpose. It also supports receiving commands through infrared by attaching a separate board. For advanced users that want more than what the IDE offers, BlinkM can also be

[3]picture credits to http://thingm.com/

Figure 2.4: BlinkM graphical interface.

reprogrammed with an Arduino.

2.1.3 Seeeduino film

Seeeduino Film [Fil10] is perhaps the first Arduino compatible board on flexible printed circuit (FPC). The goal of the board is to be extremely small, slim, flexible while not loosing the versatility needed for developing. It is split in three parts, each communicating with the next part through a 20pin bus (see figure 2.5[4]). The bus is also compatible with the Seeeduino Frame[5] boards that give additional functionality, like sensors. The boards can be cut to size at every joint according to the users needs.

Although it can be integrated into clothes well due to the flexibility, the flexible material is more fragile than the normal boards. After we sew on the film twice, some pads were cut by the thread. To solve this issue the Seeeduino team created a sewing extension[6] from normal PCB material that is compatible with the 20 pin bus.

[4]picture credits to www.seeedstudio.com/wiki/Frame_Series
[5]http://www.seeedstudio.com/wiki/Frame_Series
[6]http://www.seeedstudio.com/wiki/Seeeduino_Frame:_Sewing

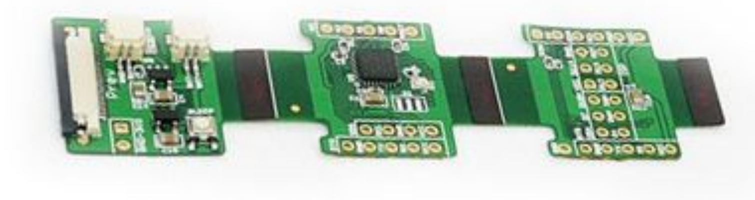


Figure 2.5: Seeeduino film.

2.1.4 Digital Addressable RGB LED strips

Besides self-reliant boards, there are other systems that need to have another board attached in order for it to function. One of these is the Digital Addressable RGB LED strip offered by Ladyada at Adafruit Industries[7] . It is made of FPC like the Seeeduino Film and contains 32 individually controllable LEDs per meter (see figure 2.6[8]).

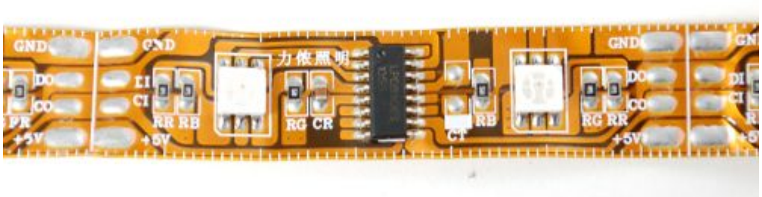


Figure 2.6: Digital Addressable RGB LED strips.

The LED strip can be controlled by any micro-controller but the library and tutorials[9] are offered only for the Arduino platform.

[7] http://adafruit.com/
[8] picture credits to http://www.ladyada.net/products/digitalrgbledstrip/index.html
[9] http://www.ladyada.net/products/digitalrgbledstrip/index.html

2.2 DIY Wearable Physical Computing Frameworks

Even though the technology has been available for a long time, it was only used by hobbyists. People that wanted to integrate electronics in clothes needed to learn various things before creating a functional result. Now there exist many varied boards that are created for clothes. This section presents widely used DIY wearable physical computing frameworks.

2.2.1 Arduino LilyPad

After Arduino was introduced, it was not long until people started using the system with clothes. For example a jersey that was used for team sports awareness [PVM07]. The users had the Arduino setup strapped to their chest. Leah Buechley noticed the need for a different type of hardware that is easier to integrate into clothes

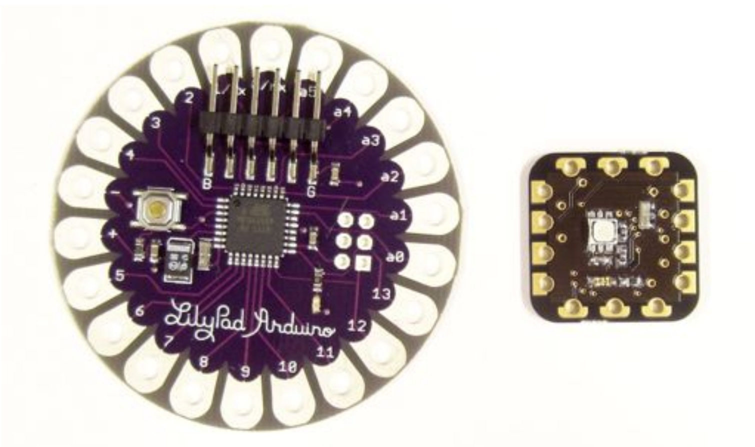


Figure 2.7: LilyPad board next to a LumiNet2 node.

The LilyPad board has been commercially available since 2007, after being an academic research project for over a year [BH10]. Although the initial prototypes were made out of fabric [BECC08], the commercial version has been made out of hard circuit boards.

The design of the board is considerably different form the classical Arduino. The circular shape allows for bigger pads (i.e. sew taps) and also more spacing in between the stitches, ensuring secure sewing. It has a low profile form, allowing the board to lay flat on the fabric and be easy to conceal. It also has an appealing design, resembling a flower (see figure 2.7), sometimes being used on the top side of the clothes.

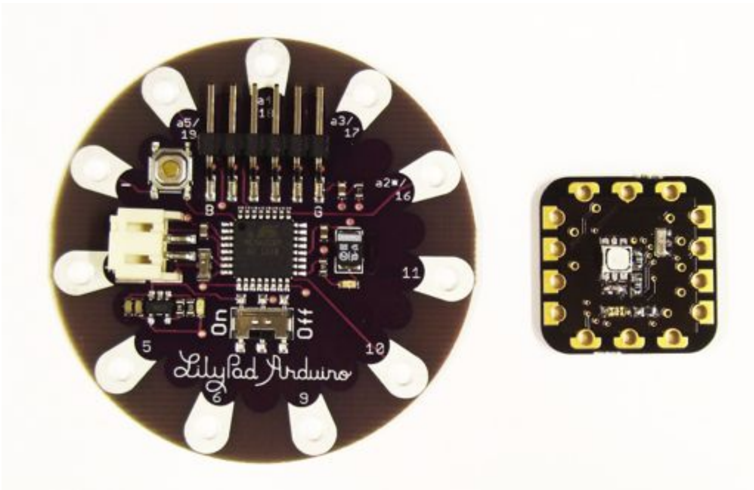


Figure 2.8: LilyPad Simple board next to a LumiNet2 node.

Two years after the original LilyPad, a new simplified version was released (see figure 2.8). It has fewer pins, a built in power supply socket, and an on/off switch. Also we noticed the In-System Programming (ISP) pinouts disappeared from the layout, but they are still accessible through small via. A wide range of accessories were developed along with the board with a consistent design.

In conclusion, we decided to make LumiNet 2 hardware and software compatible with the Arduino for the following reasons:

- A wide range of users already know how to use a LilyPad board and an even more have worked with an Arduino.

- The language and IDE are fairly easy to learn compared to other existing systems.
- Arduino has a diverse and goring community.
- While being compatible with the Arduino, the board can also use the LilyPad accessories.

2.2.2 Sparkle

Other frameworks emerged after LilyPad, one of them is Aniomagic Sparkle[Spa05]. The Sparkle kit (see figure 2.9[10]) is composed of a main board and

Figure 2.9: Sparkle kit.

The system, formerly known as Schemer or Button-Schemer [EE08] has a completely different approach to programming. The need to bypass cables and programing interfaces was addressed by programing through light. They use light signals generated by a screen animation to transmit code to the board. A visual interface to configure the blinking patterns that are sent to the board is also provided, thus eliminating any need to write code.

A downside is that the kit is limited only to a small set of operations and also, the communication with the screen is not always reliable. Other problems that we experienced while using the kit were that the boards have small sewing pads, making it difficult to pass more times through the

[10]picture credits to http://www.sparkfun.com/

holes.

2.2.3 Fabrickit

Fabrickit [DP10] is a very low tech system. It has no micro controller board only a battery board, light boards, connectors and a conductive ribbon (see figure 2.10[11]). The connectors and the ribbon are noteworthy. It is the first commercial product that offers the possibility to solder connections but at the same time keep the flexibility of fabrics.

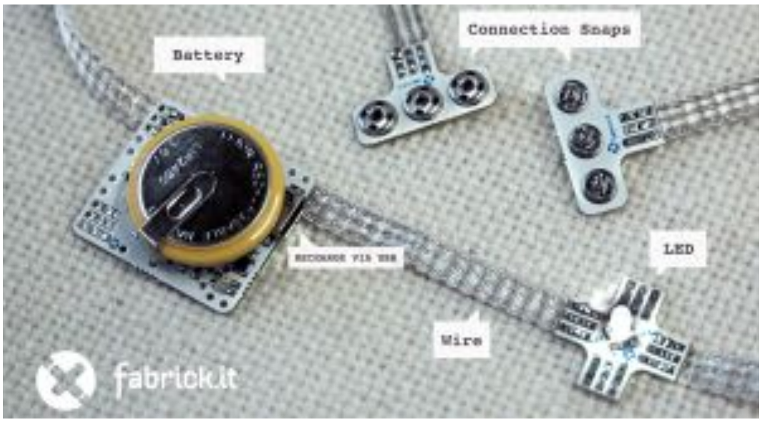


Figure 2.10: Fabrickit set.

The ribbon is narrow, made out of synthetic fibers and three conductive traces. It matches the connectors that have three pads for the ribbon and then extend to three snap connectors (see figure 2.10). But while using the kit we found that the conductive traces were to o close together, and also the number of traces were insufficient for serial communication.

2.2.4 LumiNet 1

LumiNet 1 [BBH09] is a distributed physical computing framework, developed for physical organic interfaces, simulation of tree algorithms and swarm behavior and

[11]picture credits to http://www.fabrick.it/

for wearable computing projects. The hardware design was created by Prof. Dr. Jan Borchers and is based on an AtTiny84 micro-controller and a bright Osram RGB LED. The communication port has two signal pins and two power pins (VCC and GND), supporting any protocol that uses this type of connection. Each node can communicate with up to four direct neighbors (see figure 2.11).

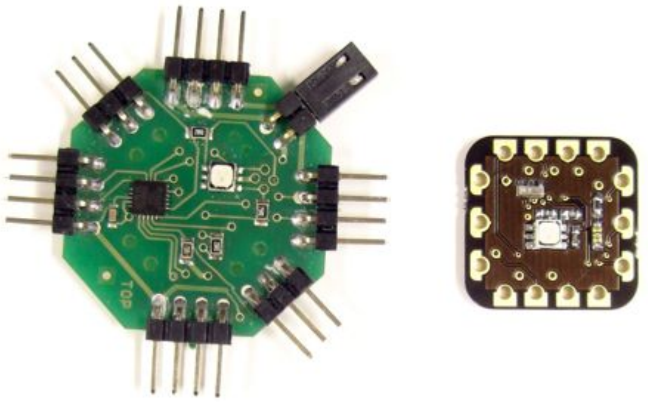


Figure 2.11: LumiNet 1 node next to a LumiNet 2 node.

It addresses the problem of reprogramming many nodes that are already in a network in an innovative way. By using a process called programing by infection, it spreads the new code to the entire network from only one node called a vector node. Another notable feature of the LumiNet 1 system is the use of bio-inspired algorithms. Because it doesn't have a central controller, it is reconfigurable and supports decentralized stimuli from sensors at any node.

The system was not fully compatible with the Arduino, and a new core has been written to support the AtTiny84 microcontroller. With the latest software version though, there is a patch that can be applied to support the AtTiny microcontroller family.

2.2.5 Hardware Comparison

The following table compares the previously presented frameworks according to key aspects needed for a wearable network. Most frameworks can be extended to support some properties but we show what is available by default.

Table columns description:

- **Visual programing.** By visual programing we mean that the user will not need to write code to reprogram the board.
- **Distributed.** The ones that have a network of boards that each have their own micro-controller.
- **Analog Sensors.** The system already has or could have analog sensors. If the value of the cell is "limited" then, there are only a limited number of sensors that work with the system.
- **Easy to sew.** We consider a framework to be easy to sew if it has adequate pads for connecting it to other components through sewing (e.g. LilyPad sewing taps).
- **Dimension (cm).** It refers to the size of the main board in kits or, in the case of Fabrickit, the LED Brick.
- **Price (€).** The prices are estimations that were valid when this thesis was written.

[1]The price is for a complete kit.

	Visual programing	Distributed	Analog Sensors	Easy to Sew	Dimension (cm)	Price (€)
Arduino	x	x	√	x	6.86 x 5.33	25,80
BlinkM	√	x	x	x	1.25 x 1.11	12
Seeeduino film	x	x	√	x	7.75 x 2	31,95[1]
Addressable RGB LED strips	x	√	x	x	100 x 1.65	32,041
LilyPad	x	x	√	√	5 ⌀	21
LilyPad Simple	√	√	√	√	5 ⌀	18,15
Sparkle	√	√	limited	√	1.5 x 1.8	11.28
Fabrickit	x	x	x	x	2.2 x 2.2	28[1]
Luminet 1	x	√	√	x	3.1 x 3.1	6
Luminet 2	√	√	√	√	2 x 2	5

Table 2.1: Computing frameworks comparison.

2.3 Wearable LED Displays

LumiNet 2 serves as a wearable display that is intended for use in fashion. In this section we shortly present other solutions for wearable displays comparing them in the end.

2.3.1 Ingo Maurer and Janet Hansen - Light Messages

One of the first wearable displays was a collaboration between illumination artist Ingo Maurer and Janet Hansen, president and chief fashion tngineer of Enlighted Designs[12] in 2002. They created an LED hat first then they followed up with several jackets and shirts (see figure 2.12[13]). In the end they made matching wedding clothes for a bride and groom that scrolled love messages.

The design is very impressive for that time. The hat hav-

[12] http://enlighted.com/index.html
[13] picture credits to http://enlighted.com

Figure 2.12: Light Messages by Ingo Maurer and Janet Hanse.

ing 400 lights while the jacket over 1000 LEDs on flexible panels. The hat and jacket were presented in several international exhibitions, in Milan and Frankfurt in April 2002[14] , and New York in May 2002[15] .

2.3.2 Leah Buechley - Tank top

In 2005 Leah Buechley created a tank top[16] with a 140 LED display. The display is capable of playing the game of life [Wai74] and displaying text. An IR receiver can also be used in combination with a PDA to set the initial state of the cells in the game. Due to the components being sewn in with conductive thread, the display looks very natural, and when the LEDs are off, the electronics are not noticeable (see figure 2.13[17]).

The method used to control the display is called, LED multiplexing. In case of pixel oriented displays it is created

[14] http://enlighted.com/pages/events2002milan.shtml
[15] http://enlighted.com/pages/events2002ny.shtml
[16] http://web.media.mit.edu/~leah/grad_work/projects/LED_clothing/tank.html
[17] picture credits to http://craftzine.com/magazine/

Figure 2.13: LED Tank top by Leah Buechley.

by having LEDs at the intersection of the column and row lines positive end of the LED will always connect to one of the two and the negative to the other. That means that only one row or column can be displayed at a time but due to persistence of vision, the viewer believes the entire display is always active. Persistence of vision occurrs when a bright image stays as a visual impression a fraction of a second after it has disappeared.

2.3.3 Philips Lumalive - Dressing Light

Philips Research developed a special fabric with RGB LEDs woven in the structure, called Lumalive (see figure 2.14[18]). The LEDs are woven into a flexible fabric and covered with a light diffusing padding. it has a good enough resolution for images, text, logos, and even animations, having the primary use in advertisement [CKV11]. Lumalive is one of the first displays fully integrated in fabric.

In 2006 the german fashion designer Anke Loh used the Lumalive fabric in her first fashion project in Chicago,

[18]picture credits to http://www.research.philips.com

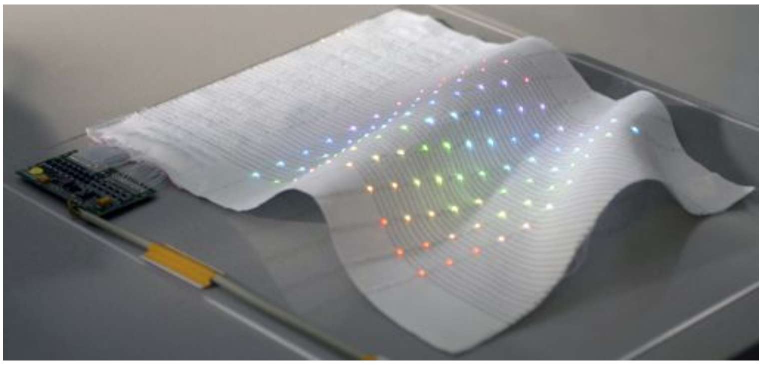


Figure 2.14: Lumalive fabric from Philips.

'Dressing Light'[19] . The dresses attracted media coverage both in the USA and the rest of the world (see figure 2.15[20]).

Figure 2.15: 'Dressing Light' dress by Anke Loh using lumalive fabric.

2.3.4 Hussein Cahalayan - Video dress

Hussein Cahalayan collaborated with Moritz Waldemeyer in 2007 to create two video dresses. The first shows a time lapse of a rose opening and closing while the second sharks moving through water (see figure 2.16[21]).

The dresses were created in just 4 weeks, while using

[19] http://ankeloh.net/2010/07/dressing-light/
[20] picture credits to http://ankeloh.net
[21] picture credits to http://www.talk2myshirt.com

Figure 2.16: Video dress in by Hussein Cahalayan in colabortion with Moritz Waldemeyer.

only standard manufacturing techniques and off the shelf components. The Video dresses have each 15000 LEDs and were part of the Autumn/Winter 2007/8, "Airborne".

2.3.5 Moritz Waldemeyer - Lighted jackets

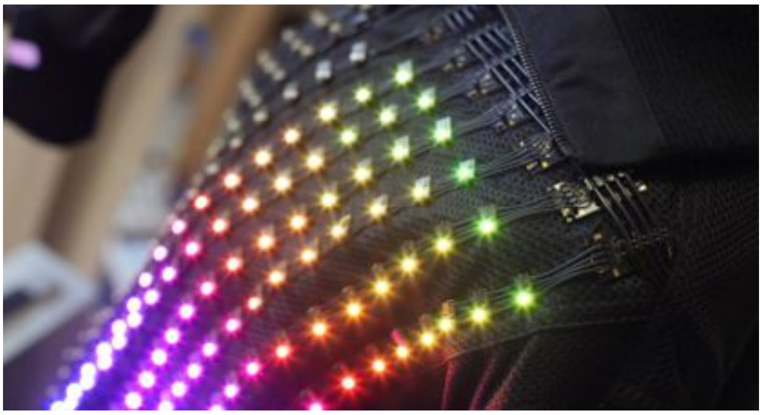


Figure 2.17: One of the latest lighted jacket design by Moritz Waldemeyer.

After the collaboration with Hussein Chalayan, Moritz Waldemeyer continued making LED art, perfecting his system for video enabled clothes. One of his latest creations are five video jackets[22] for the members of the band Take That. The jackets had a custom made full color LED pixel that was especially designed for wearable applications (see

[22] http://www.waldemeyer.com/take-that-video-jackets

figure 2.17[23]). Each jacket had 400 LEDs and is controlled by a miniature video player.

2.3.6 Jan Borchers and René Bohne - LumiNet 1 jacket

René Bohne created a jacket with 80 LumiNet 1 nodes under the supervision of Prof. Dr. Jan Borchers (see figure 2.18[24]). The jacket has a grid of self aware nodes that react to the messages sent by the rest of the network.

Figure 2.18: LumiNet 1 jacket by Jan Borchers and René Bohne.

Unlike most displays, LumiNet can have sensors attached at any node to induce events into the system. The jacket was programmed with several bio-inspired algorithms. One of them was "Light Ring" where a ripple of light was passed from one side to the other of the jacket, and even from one jacket to another through infrared lights and sensors.

[23]picture credits to http://www.waldemeyer.com
[24]picture credits to luminet.cc

2.3.7 Cute circuit - Galaxy dress

The Galaxy dress designed by Cute Circuit[25] is the largest wearable display in the world. Made out of a layer of silk embroidered with 24000 full color LEDs that measure only two square millimeters (see figure 2.19[26]). The power needed to power the display is provided by several iPod batteries that last in average 30 minutes. The entire dress has four layers of chiffon to difuse the LED light and is also embroidered with more than 4000 swarovski crystals, looking great even when turned off [Sey10].

Figure 2.19: Galaxy dress by Cute circuit.

2.3.8 David Forbes - The Video Coat

On his website[27] , Forbes describes in detail where the idea of his video coat. A video coat is a coat that is capable of displaying full video output. It is completely covered in flexible display strips. Each strip has six Allegro A6282 chips and many groups of 3 LEDs each (red, green and blue) that are controlled by the chips. He used FPC that was partially strengthened with kapton on the back side to prevent the

[25] http://www.cutecircuit.com/collections/special-projects/galaxy-dress/
[26] picture credits to http://www.fashioningtech.com
[27] http://www.cathodecorner.com/videocoat/index.html

Figure 2.20: The Video Coat by David Forbes.

copper traces breaking at the solder joints while bending. The coat has in total 19200 LEDs and is capable of displaying a video of 160x120 pixels (see figure 2.20[28]).

The video coat is one of the most successful wearable display for video rendering. The downside is that despite the flexibility and low weight of the display strips, the electronic boards that control it are bulky and heavy. Furthermore, to put everything together, Forbes relied on large amounts of hot glue and double sided sticky tape.

2.3.9 Barbara Layne - Wearable Absence

Wearable absence uses bio-sensing, wireless technologies, soft cabling systems with a small display to activate memories of a person that is not present (absent) from a particular location(see figure 2.21[29]). The sleeve has woven in LEDs to create an 8 by 8 display used to scroll words. The fascinating aspect is that the output is triggered by the bio-sensors .and thus the users have no control over when the memories will come.

[28] picture credits to www.cathodecorner.com
[29] picture credits to http://finearts.concordia.ca

Figure 2.21: Wearable Absence by Barbara Laynes.

2.4 Wearable LED Displays Comparison

Based on the related projects presented, we identified some current trends in wearable LED displays.

Table columns description:

RGB Indicates if the display is full color or monochrome.

Soft connections It is considered a soft connection, one that is made completely out of fabrics or threeds.

Scalable Not limited to a maximum number of LEDs.

Connected Can communicate through WiFi, Bluetooth, Xbee or any other wireless technology.

Sensors The system contains sensors to allow interaction with the display.

Open We mean ether released as Open Source Hardware or that the complete information on how it is made is available online.

Device reprogrammable The behavior of the display can be modified with the help of a mobile device.

Text enabled The display can show text, most commonly through scrolling.

LED multiplex It shows if the display uses multiplexing or not.

Led nr. The number of LED's existent in the system.

A dash ("-") in a column indicates that the relevant information is unavailable. Due to the fact that some of the projects are commercial products, sometimes we did not have enough information to fill all the cells in the respective row.

	RGB	Soft connections	Scalable	Connected	Sensors	Open	Device reprogrammable	Text enabled	LED multiplex	Led nr.	Year
2.3.1—“Ingo Maurer and Janet Hansen - Light Messages”	x	x	x	x	x	x	x	√	-	>1000	2002
2.3.2—“Leah Buechley - Tank top”	x	√	x	√	x	√	√	√	√	140	2005
2.3.3—“Philips Lumalive - Dressing Light”	√	√	x	√	x	x	√	√	√	196	2006
2.3.4—“Hussein Cahalayan - Video dress”	√	x	x	-	x	x	-	x	x	15.000	2007
2.3.5—“Moritz Waldemeyer - Lighted jackets”	√	x	-	x	x	x	√	x	√	400	2011
2.3.6—“Jan Borchers and René Bohne - LumiNet 1 jacket”	√	x	√	x	√	√	x	x	x	80	2009
2.3.7—“Cute circuit - Galaxy dress”	√	√	-	x	-	x	x	x	x	24000	2012
2.3.8—“David Forbes - The Video Coat”	√	x	√	x	x	√	√	√	x	19200	2011
2.3.9—“Barbara Layne - Wearable Absence”	x	√	x	√	√	x	x	√	√	64	2009

Table 2.2: Wearable LED displays comparison.

2.5 Smart Phones as Medium

Amarino [KB10] was created to answer the developers' need to have an easier way to connect Arduino boards to their mobile applications. The system is an important step towards blurring the barriers between mobile phones and wearable computing. However, even though the interface is created for novice users, code still needs to be written for more complex applications.

2.6 Integration

Hannah Perner-Wilson offered through her thesis "A kit of no parts"[PW11] a systematic and complete guideline on how to integrate electronic circuits into almost any material. What we will refer to is mainly her experiments with fabrics. She offers on the website HOW TO GET WHAT YOU WANT[30] in collaboration with Mika Satomi a compilation of many tutorials and information on how to combine many materials to get wearable sensors, actuators or just to connect parts to each other (see figure 2.22[31]).

Figure 2.22: Fabric elements found on the HOW TO GET WHAT YOU WANT website.

[30] http://www.kobakant.at/DIY/

[31] picture credits to http://www.kobakant.at/DIY

Chapter 3

System Design

> *"Art is the desire of a man to express himself, to record the reactions of his personality to the world he lives in."*
>
> *—Amy Lowell*

In this chapter we present the structure and design of our system. We follow with a short description of the initial studies finally bringing together the requirements that resulted from both our interaction with the designers during the workshop and also the overview tables (see tables 2.1—"Computing frameworks comparison." and 2.2—"Wearable LED displays comparison.") . The description of the development process illustrates our experiences with the hardware prototypes leading to the final result. In the end we will elaborate on the firmware and an iPhone app we created for the system.

3.1 Research Questions

This work aims to investigate how distributed systems can be improved for Fashion projects.

- Based on LumiNet 1, how can the hardware be improved to be used by fashion designers?

- How to better interact with an illumination and sensor network using a mobile device?

3.2 Initial Studies

The initial studies at the beginning of the project were made in parallel with the trials to create and improve the hardware.

3.2.1 Research Workshop

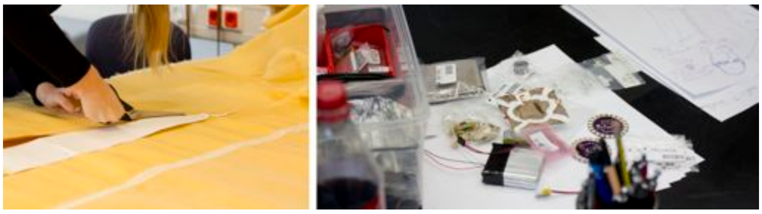


Figure 3.1: Smart Fashion Aachen workshop pictures.

We organized a wearables workshop called Smart Fashion Aachen[1] to have a better understanding of how fashion designers perceive creating an application with smart components. A variety of users, ranging from electrical engineers to fashion designers participated in the one day workshop.

The schedule of the workshop was:

- Brainstorming session with the theme: "We are in the future, anything you can imagine is possible! What would you create with technology?".
- An introduction to the concept of wearable computing, projects and the latest hardware.
- Hands-on activity.

[1]http://hci.rwth-aachen.de/smartfashionaachen18112011

- Short presentation of each project.

A more detailed account of the workshop results can be found on the workshop page.

We will summarize what we deduced from the interaction and the feedback.

- The designers had many creative ideas during brainstorming and concentrated more on the visual aspect, compared to the electrical engineers who were more interested in the clothes function.
- After we explained the way components should be connected, most designers did not have any difficulty in attaching the electronics.
- They were open to learning new technologies, but one day was not enough to grasp the concepts of programming Arduino boards. One was the exception, since she had previous experience with programing micro-controllers.
- A lot more time was attributed to thinking and creating the garments than we expected, resulting in very aesthetically pleasing results.
- For one concept the LilyPad board was not adequate and when offered to use the prototype the designer was convinced it was a better option.
- Some were interested in staying in contact and learning more about ways to integrate electronics into clothing.

3.2.2 Experiments

To better understand the difficulties being faced by designers while putting together an LED display, we decided to create one by ourselves. Following the instructions from Leah Buechley for the LED Tank top we created a skirt

Figure 3.2: Smart Fashion Aachen workshop brainstorming session (left) and the hands-on activity (right).

with an 8 by 8 display using an Arduino LilyPad instead of the AtMega16 (see figure 3.3).

Figure 3.3: The skirt with LED matrix controlled by a Lily-Pad.

For the skirt we developed an Arduino library that scrolls text on the display with a 5 by 8 font. The font includes all the printable characters found in the American Standard Code for Information Interchange (ASCII) character set. It can also scroll text sideways in both directions.

We concluded that even though sewing is more familiar to the users, we need to provide a faster way of connecting the nodes to the material while keeping the connections soft (made of fabrics). The diagram for the skirt and a pictures of the final setup can be found in Appendix A—"LED Matrix Skirt".

3.3 Requirements

Based on the new knowledge from the initial experiments and the related work presented in chapter 2—"Related Work" we came to the following main requirements.

M1 **No programing needed.** We will provide an easy graphical interface for designers on a familiar system like the iPhone.

M2 **Improve the integrability.** To improve the way the nodes integrate with fabrics by making them smaller and providing various ways to use soft connections like conductive thread, conductive fabric and conductive ribbons. The aim is to enable the users to easily create a circuit with the nodes even without soldering.

Nine more requirements should be met by the implementation that are not related to the end user but were considered important for further developments.

R1 **Extensibility.** Provide a set of simple functions to control the system so that it can be accessed through wireless communication. It makes it easier for developers to build new interfaces in the application layer without the need to change the framework on the nodes.

R2 **Simpler node connections.** The connection possibilities of LumiNet 1 left a lot of room for erroneous connections that may even lead to hardware damage. We want to provide a simpler connection where hardware damage can be avoided.

R3 **Fast integration.** Even though sewing provides a flexible connection between the hardware modules and also between the hardware modules and the clothes, it is very time consuming, rendering it impractical for a large amount of nodes. We want to offer a faster and more efficient way to interconnect the nodes to each other and to clothes.

R4 **Sensor support.** Attaching more sensors than the integrated light sensor to the nodes should be possible.

R5 **Full Arduino compatibility.** The reasons are stated in the sections 2.1.1—"Arduino" and 2.2.1—"Arduino LilyPad" of the related work chapter.

R6 **Fast and reliable communication.** LumiNet 1 does not have a crystal and relies on a bio-inspired protocol called Bynase. Since we changed the hardware we need to evaluate which communication protocol is better for the new configuration.

R7 **Platform independent.** We want the hardware abstraction layer to be accessible from any platform, even though we will provide an implementation for the application layer on the iPhone.

R8 **Open source hardware and software.** We will publish the eagle files and source code for the LumiNet 2 nodes. We want the framework to be used and further adapted by the community.

R9 **Fast firmware loading.** For LumiNet 1, if there is any change in the bootloader, the update procedure is time consuming. LumiNet 2 should offer a faster way to change the bootloader on the nodes.

3.4 Hardware Design

LumiNet 2 is designed to be part of a network which has two main types of hardware components, the WiFi node and LumiNet 2 nodes.

3.4.1 Network topology

The physical topology for LumiNet 2 node network is a linear daisy chain. It means that adjacent nodes linked one to the next in a line (see figure 3.4). LumiNet 1 nodes can be connected to up to four other nodes, allowing for more complex structures. Due to the unknown structure of the network, deadlocks can occur frequently. A deadlock occurs when processes holding some resources request access to resources held by other processes in the same set. The simplest illustration of a deadlock consists of two processes, each holding a different resource in exclusive mode and each requesting an access to resources held by other processes [Sin89]. Handling deadlocks is a complex problem that still does not have a fast reliable solution in distributed systems hence we decided to simplify the connection [OV11].

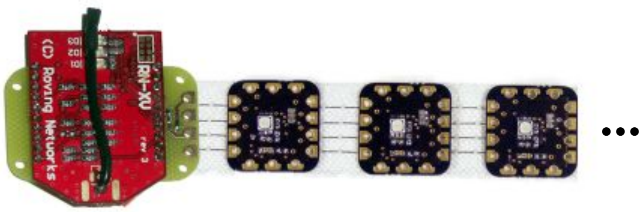


Figure 3.4: LumiNet 2 network example.

3.4.2 WiFi node

The WiFi node consists of the RN-XV WiFi module from Roving Networks[2] soldered to a milled adapter. The adapter couples the LumiNet 2 connection pads to the serial port of the WiFi module, permitting it to be attached to the network like all the other nodes (see figure 3.4). It also has four holes on each corner for sewing to the fabric.

[2]http://www.rovingnetworks.com/products/RN_XV

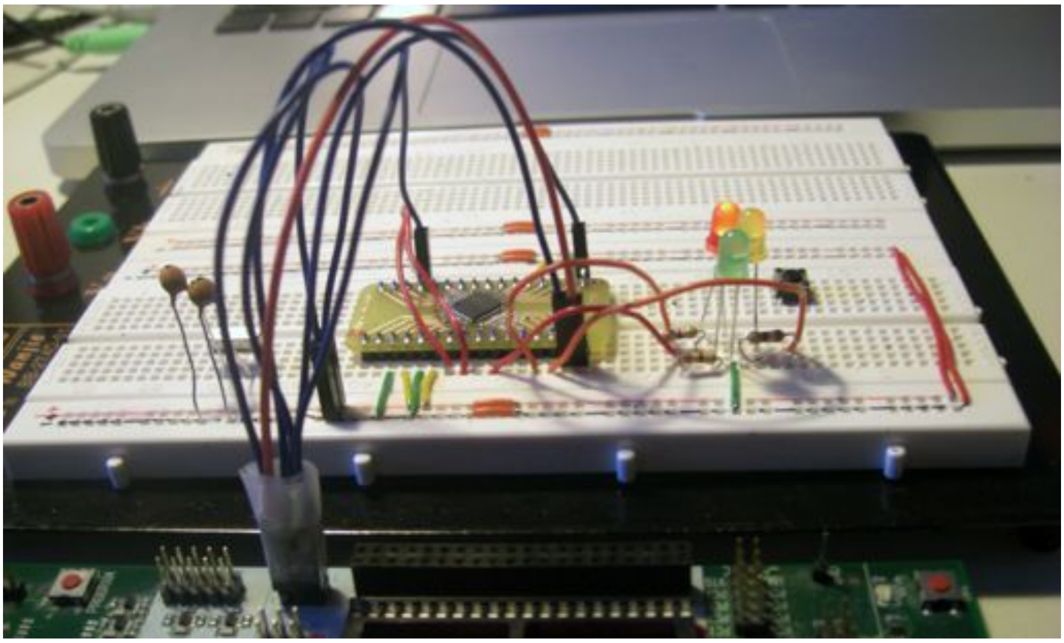


Figure 3.5: LumiNet 2 design on a breadboard.

3.4.3 LumiNet 2 Node Development

The hardware board was completely redesigned in order to meet the integrability, communication protocol and Arduino compatibility requirements.

Initial component change:

- **AtMega 168 micro-controller.** The AtMega 168 TQFP package is used instead of the AtTiny84 QFN package because the AtTiny controllers are not natively supported by the Arduino environment. The preferred package also includes hardware support for serial communication. The pins of the AtMega 168 micro-controller are also compatible with the AtMega 328.
- **8 MHz Quartz crystal.** A crystal was added to the node to improve the stability and the communication speed of the network.

The first step was to test the new micro-controller with the quartz crystal and decide on the basic components that need to be added on the new board. It was initially accomplished by emulating a bare-bones version of an Arduino board on a breadboard. A breadboard is a basic construction that permits prototyping a circuit without the need to solder the parts(see figure 3.5). We used the actual micro-controller package that we intended to use in the final version but the other parts were bigger through hole components.

Figure 3.6: LPKF Protomat S62 PCB Milling machine.

After grasping the basics of the hardware we started creating real prototypes on PCB boards. The LPKF Protomat S62 PCB Milling machine (see figure 3.6) from the FabLab Aachen[3] was used for that purpose. It has relatively good performance and different designs were tested to try and improve further through making progressive prototypes.

The following sections show five prototypes which contributed to some of the significant design decisions. The changes between the versions and the conclusions we

[3]http://fablab.rwth-aachen.de

reached after use are shorty detailed.

LumiNet 2 prototype design v1.0

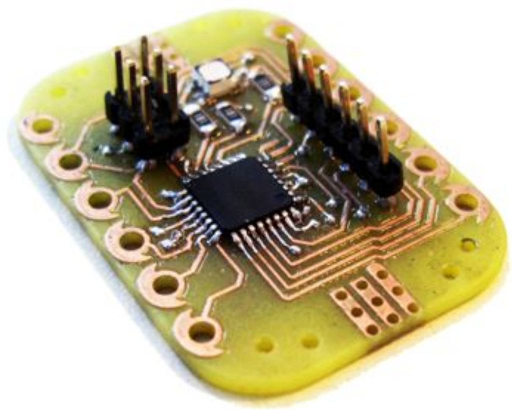


Figure 3.7: LumiNet 2 prototype design v1.0

Design:

- All the components on one side of the board.
- Standard programing pinouts for ISP and Serial.
- Fabrickit pads for the board to board connections.
- LilyPad petal-small sew taps for additional sensors.

The Fabrickit pads were used because we found it important to have a fast and easy way to connect one board to another and then to the clothes to satisfy requirement R3.

The LilyPad connections are called petals in the hardware library and come in various sizes: long, large, medium, small and tiny. We decided to use them because they are familiar to some users and they are made specifically for sewing with conductive thread.

The board was bigger than the LumiNet 1 nodes. The milled boards are an exercise in milling and component assembly as well as to have a hands on experience with electronics.

LumiNet 2 prototype design v1.1

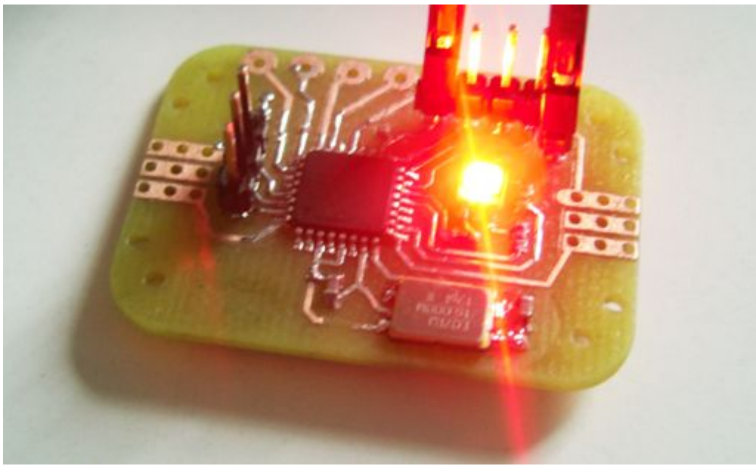


Figure 3.8: LumiNet 2 prototype design v1.1

Design changes:

- Custom programming pinouts, duplicate pins were removed (Reset, GND and VCC).
- Only one row of pads for extra sensors.
- Changed the sensor sew taps to the LilyPad petal-tiny.

The result was still not smaller than LumiNet 1. With the new choice in micro-controller it was difficult to keep all the components on only one side of the board and still decrease the size.

LumiNet 2 prototype design v1.2

Design changes:

- The board has components on both sides.

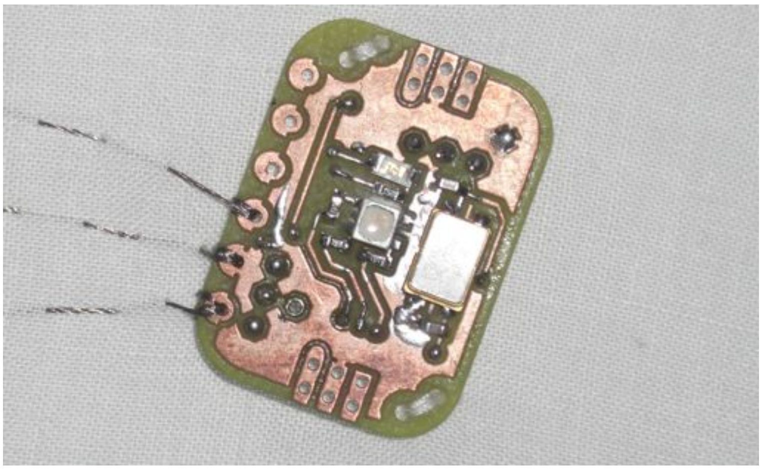


Figure 3.9: LumiNet 2 prototype design v1.2

- The programming pin position is optimized.
- Only two sets of fixing holes for sewing.
- An LED is added that is used as a light sensor.

The sewing holes of the taps were not big enough and made sewing difficult and also the board was still too big. Sewing the board in a circuit was found to be enough to secure it and the extra sewing holes were not required. Keeping the standard programming pinout took too much space. The Fabrickit connectors were too long and the shape was not optimal for routing the traces.

LumiNet 2 prototype design v1.3

Design changes:

- A phototransistor was added instead of the LED.
- The standard pin pads were replaced by pogo pin connectors.
- Custom pads for the connections.
- Custom pads for the sensors.

The phototransistor was chosen to increase the light sensing speed. To use the LED as a light sensor, the micro-

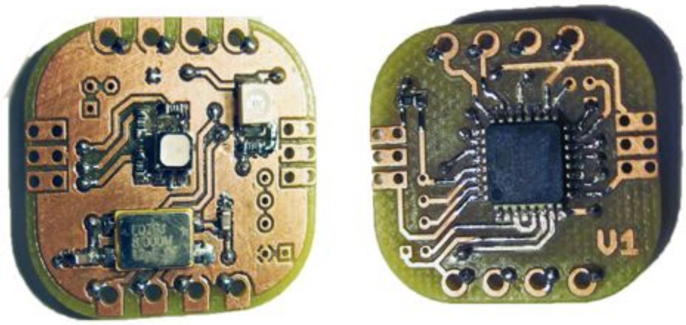


Figure 3.10: LumiNet 2 prototype design v1.3

controller had to run a sequence of code that introduced delays into the system while only one simple command is needed for phototransistors.

The evolution of the LilyPad Simple board from the LilyPad board was an inspiration to remove the ISP port and in our case also the serial port replacing them with small ports for pogo pins.

Pogo pins are spring loaded pins that are used in testing for creating temporary connections with a board. They are placed in a test board that has matching holes to the pads that they need to connect to and contains a code that automatically runs and gives feedback on the performance. When the board needs to be tested it is just pushed against the pogo pins for the duration of the test procedure. They were preferred because they considerably decrease the board reprograming time.

This version of the prototype was used during the Smart Fashion Aachen workshop embedded into a hand guard (see figure 3.11). Due to the space and size constraint of the design the LilyPad board was not suitable. The available prototype of the node was presented as a possible

Figure 3.11: Hand guard design with LumiNet 2 prototype

solution and it perfectly matched the requirements. An external light sensor was selected instead of the one on the board to make the design more aesthetically pleasing. All pins, including the communication pins, were used to control the components. Due to the fact that the Fabrickit connectors do not afford sewing, we had to solder wires on the pads.

LumiNet 2 prototype design v1.4

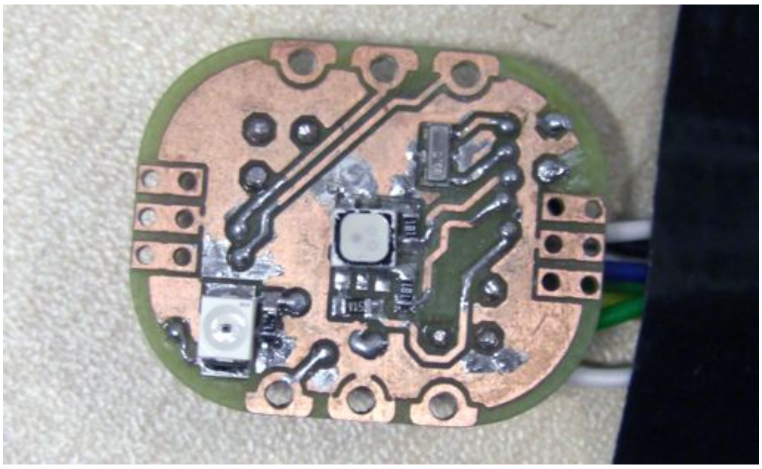


Figure 3.12: LumiNet 2 prototype design v1.4

Design changes:

- The bigger quartz crystal was replaced by a tiny resonator.

- The footprint of the pogo pin connections changed.
- The the number of connectors for sensors was decreased to 4.
- The custom pads for the sensors were changed.

At this point we were looking into alternatives for the Fabrickit ribbon and a smaller package phototransistor. Previous experience dictated that we place phototransistor in line with the LED. The positioning of two components which could be partially or completely uncovered is aesthetically very important. The number of additional pins was decreased to increase the distance between the pads and the size of the pad. Due to these changes we can better differentiate the network and the sensor connections. The remaining pads are connected to a pin that supports hardware pulse-width modulation(PWM) and 3 analog pins.

3.4.4 Final Design, LumiNet 2 V2.0

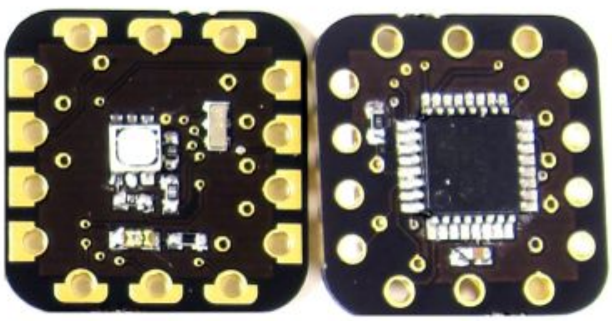


Figure 3.13: LumiNet 2 nodes v2.0.

Connection

A conductive ribbon from Amohr[4] was chosen for the node connections. We sourced a 10m custom made ribbon for the

[4]http://www.amohr.com

LumiNet 2 v2.0 board from Amohr that was modified from one of their existing products by reducing the number of traces.

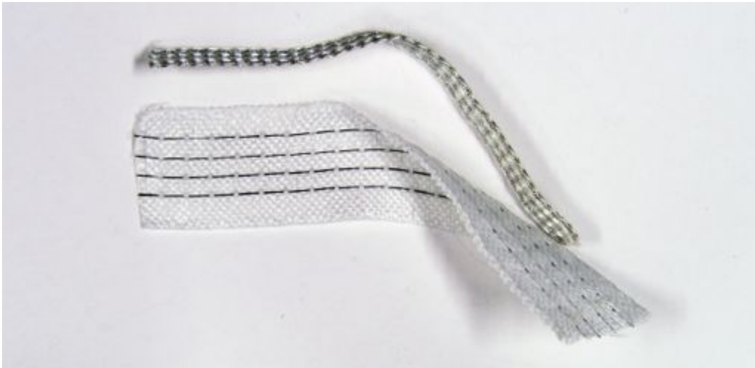


Figure 3.14: FabrickKit ribbon next to the Amohr ribbon

The chosen conductive ribbon has the following properties that match our needs.

- **Low resistance.** The ribbon has around 0.05 Ohm/m resistance.
- **Matching spacing.** The spacing between the conductive traces of the ribbon matches the connection pads of the LumiNet 2 v2.0 node.
- **Solderable traces.** The traces are solderable, giving the possibility of a fast connection between boards without loosing the flexibility of the materials. Compared to the Fabrickit ribbon, that is also solderable, it has bigger spacing between traces making it easier to solder.
- **Iron on.** That means it can be fused to fabrics by ironing it with a clothes iron. That means that once all the nodes are soldered, they can be attached by ironing the ribbon in between the nodes on the material.
- **Thin**. Once the ribbon is ironed on a fabric it blends in really well unlike the Fabrickit ribbon that is at least two times thicker.

- **Width compatibility with LumiNet 2.** Connecting one node to another with it transforms the nodes in a continuous strip as it can be seen in figure 3.15.

The description of the Amohr conductive ribbon and details on the model number can be found in the Appendix C—"Amohr conductive ribbon".

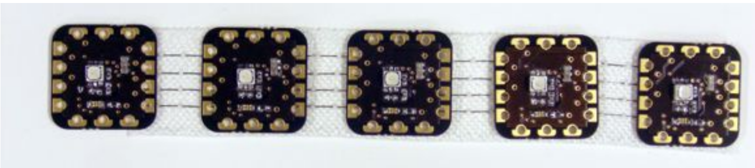


Figure 3.15: LumiNet 2 node chain with Amohr conductive band.

Final Board Design

Design changes:

- The side connections were changed.
- Rerouted the board.
- A smaller phototransistor was used.
- The footprint of the pogo pin connections was changed again.
- A capacitor between the power and ground pin was added.
- Placed the light sensor on a straight line under the LED.

Before sending the board to manufacturing the board was rerouted to fit the specifications of the manufacturer and also to remove workarounds that were introduced to be able to mill the board with the previous machine. The final version of the nodes was created with 4 pads to match the Amohr conductive ribbon.

We considered using FPCs for the board but we found several disadvantages.

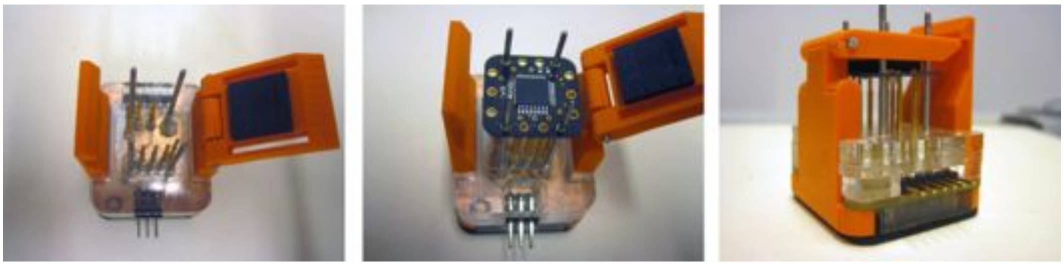


Figure 3.16: LumiNet 2 pogo pin adapter.

- The failed sewing experiments with Seeeduino film (see section 2.1.3—"Seeeduino film").
- David Forbes, the creator of the video coat, who also used FPCs (see section 2.3.8—"David Forbes - The Video Coat") warns that the copper traces are very sensitive to bending around solder joints and might break.
- The joints of the components would be under a lot of stress with every movement of the fabric because our design is double sided.

The schematic, board design and bill of materials for the LumiNet 2 v2.0 node can be found in the Appendix B—"Luminet 2 v2.0 Hardware Design".

3.4.5 Programing Interface

For quickly loading the bootloader and firmware on the board we created a pogo pin adapter.

The adapter was created with the mill and the case for securing the board on the pogo pin adapter was created partially in the laser cutter[5] and partially printed. The clear acrylic slices that can be seen in figure3.16 were initially used as guides for soldering the pogo pins in place. They also serve to protect and conceal the circuit. The orange part was printed with a professional 3D printer[6] to provide good contact during programing.

[5]http://hci.rwth-aachen.de/lasercutter
[6]http://hci.rwth-aachen.de/3dprinter

The schematic, board design and bill of materials of the pogo pin adapter can be found in the Appendix D—"Pogo Pin Adapter".

3.5 System Architecture

Figure 3.17: LumiNet 2 system overview.

The final LumiNet 2 system evolved into a multi layered architecture. There are three main layers, the physical layer, the hardware abstraction layer, and the application layer (see figure 3.17). Each layer of the architecture and the communication between them will be explained in the following sections.

3.6 Physical Layer

The physical layer is represented by the unmodified Arduino core. It provides access to the lower level hardware functions of the micro-controller. An advantage of using this model is that each node remains fully compatible with the Arduino environment.

3.7 Hardware Abstraction Layer

The hardware abstraction layer consists of the LumiNet 2 firmware. It handles the setup and runtime for the nodes and is very similar to a virtual machine.

The following conventions are introduced in order to better explain the functionality of this layer.

- The **top side** of the board is the side with the LED.
- The **right side** of the board is where the arrow on the top side of the board is pointing (see figure 3.18)
- The **left side** is the opposite of the arrow.
- The **first node** is the one linked to the WiFi node or without any neighbors on the left side.
- The **last node** is the one without any neighbors on the right side.

Figure 3.18: LumiNet 2 V2.0 board design, top side.

3.7.1 Connectivity

The nodes connect to a special WiFi node that acts like a ad-hoc network . It allows devices to connect and communicate to the network. There were no changes done to the network module. It is used only to convert the wireless signals from the connected devices to a serial stream.

Node to Node Communication

The network follows a master-slave model for the communication. The same firmware is running on all nodes regardless of its function. At boot time, the first node becomes the master and also each node receives a unique id corresponding to the position in the network. We chose this model to keep the nodes synchronized. When the nodes are running, the master receives commands through WiFi and fulfills them by distributing messages to the other nodes. Slave nodes only reply when asked by the master (see figure 3.19). The master serves only for coordination and communication and, all nodes are still working independently.

Firmware Structure

There are two main phases in the firmware, the setup and the runtime. The setup is in charge of the topology scan and initializations while the runtime plays the learned behavior of the node.

Setup

During the setup phase there are three main goals.

- Find out which are the first and the last nodes.
- Determine the id of each node

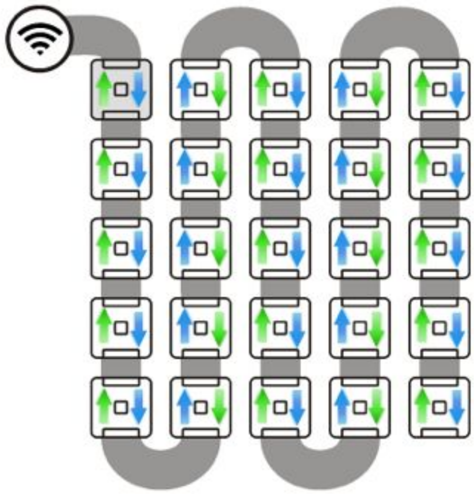


Figure 3.19: LumiNet 2 network diagram, blue arrows represent the commands from the master and the green ones the reply.

- Determine the base delay it takes for a message to reach from the first to the last node.

At this point, all nodes have the same behavior. To determine if a node has neighbors it loops through the following steps until there is a timeout or it finds both neighbors.

1. Check if there are messages coming from the left. If there are then it has a neighbor to the left and it is not the first node.
2. Send a test message to the connection on the right.
3. Check if there are messages coming from the right. If there are then it has a neighbor to the right and it is not the last node.
4. Check if both neighbors were found. If yes, exit the loop, if not go to 1.

Then the first node sets it's id to 0 and sends the character 'i' followed by the id. Each other node, is listening for the command. When they get the id of the node on their left they set their id as the received id plus one and send their new id to the right. The last node will send back an acknowledgment followed by its id. When the nodes receives the acknowledgment, they store the number of nodes as the last id plus one.

The base delay is important in order for all the nodes to start blinking at the same time. To determine the base delay the first node starts a timer and sends a message to the last node. The last node sense an acknowledgement when it receives the message. The other nodes just listen first to the left and then to the right, forwarding the message. When the acknowledgement comes to the first node, it sets its internal delay to half of the roundtrip time and sends the character 'd' followed by the delayPerNode (delay divided by total nr. of nodes). The other nodes calculate their delay with the formula, delayPerNode*(nrOfNodes-id-1). In that way, the last node will start immediately.

All nodes exit setup mode and turn green after the delay is set, except for the first node that turns blue signaling the end of the setup phase.

Runtime

The runtime phase is split into three phases, listening to commands, running the display code and forwarding or receiving answers.

Video playback practices were the inspiration for our framework due to nodes being part of a display, and playing animations. Thus when we mention frame, we refer to one still image from the animation. Also at node level, a frame contains only the RGB value needed to be displayed by it. For optimization we also decided to store the frame id, the node only memorizing the frame id and the RGB value for when the color of the corresponding pixel changes.

Listening to Commands

In this phase there is a slight difference between the master node and the slave node. The master node listens to the WiFi module to check for commands. When it receives recording signals it gets set in record mode and forwards the values for each node while storing the ones for itself. It can also receive query signals, then it asks the slave nodes for input and then forwards the answer attaching its own input.

There are a number of possible commands that can come from the master, each triggered by receiving a special character. The table 3.1—“Node commands listing. ” serves as a reference for the command triggers. Trailing data represents the additional information the node needs to listen to before triggering the respective method.

The possible commands are the following

- storeFrame(frameId, RGB value), adds the value with the id to the internal list of frames of the node.
- fireFrame(frameId), it forwards the command to the right and if the node does not have the frameId in the memory it does nothing. If the value is found it sets a target time called tick as the current time plus the delay needed for the synchronization. It also sets the next frame value into a variable called nextValue according to the received frameId.
- returnSensorXValue() where X is the received number, the node forwards the request to the right and responds to the left with the sensor value.

Trigger	Trailing Data	Trigered Method
’s’	6 bytes (12 bits node address, 12 bits frame number and 3 bytes RGB value)	storeFrame(frameId, RGB value)
’f’	2 bytes (12 bits for the frame number)	fireFrame(frameId)
’1’ to ’5’	no trailing bytes	returnSensorXValue() where X is the received number

Table 3.1: Node commands listing.

Running Display Code

If the node is not in record mode then it is playing the last recorded frames according to the fireFrame(x) (where x is the frame id) Each node has a set of variables that permit them to learn a blink pattern and replay it. One is a list of frames where each element contains the frame id and the RGB value the node should display in that particular frame. Then there are three variables that are linked to each other, the currentValue, nextValue and tick.

The code is fairly simple, it changes the currentValue RGB value to nextValue when the current time is the same as tick then it displays the color.

```
if( (tick - milis()) <= 0){
currentValue = nextValue;
tick=millis()+delay;
}
display();
```

Forwarding or Receiving Answers

In this phase, slave nodes check if they are receiving any signals. The master node can receive answers for the sensor queries and it forwards them to the application layer.

3.8 Application layer

The application layer has two distinct components, the LumiNet 2 core and the user interface. The modularity allows the creation of varied apps that use the core classes.

3.8.1 LumiNet 2 Core

The core classes have the task of transforming the received data into commands for the nodes and map the matrix coordinates to node coordinates. It supports two inputs, two dimensional arrays with RGB colors or text. For the text there are special parameters needed like the size and the position on the display. If the parameters are

Figure 3.20: LumiNet 2 app v 1.0.

not set, the system will use the default values. They are also the ones that mange the communication with the node.

3.8.2 User Interface

The feedback from the designers at the research workshop suggested we should create a user interface that allows graphical input. It should alow the user to send created patterns or text to the display. Also, it should set the base proprieties for the core classes.

A simple tabbed app for the iPhone that has three tabs, one for drawing the other for writing and the last for changing the settings (see figure 3.20).

Drawing is done in the first tab, that offers basic functionality and a grid that represent the nodes in the LumiNet 2 display. After discussions with users we also added the fill tool, that in this case clears the complete display and the eyedropper tool. To change the color, we used a color picker (see figure 3.20). The text tab is very basic, offering a text box and a color picker. The settings for the layout and connection can be changed in the third tab. We used graphical representation for the network topology settings, and simplified the other options.

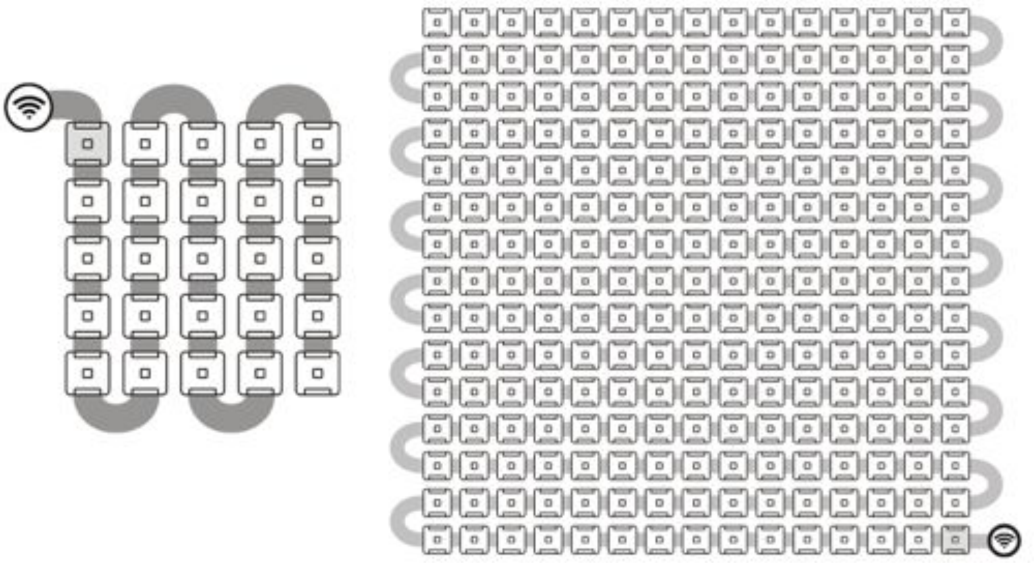


Figure 3.21: LumiNet 2 hardware setup diagrams.

In order to determine the usability of the iPhone app we did an initial user study. We wanted to know that the system is intuitive and easy to use. No usage instructions for the app itself were given to the users since our goal is to verify that the system was useable for untrained users. In order to test the system independently from the hardware we did explain the basics of the hardware. If the user could not accomplish a task, help was provided so she could finish it.

The user was asked to accomplish several different tasks in order to evaluate the system. The tasks were chosen to cover the planned functionality of the system.

User study tasks:

- **Task 1:** Open the "LumiApp" application and verify if it is connected to the LED display.
- **Task 2:** Draw a pattern with at least three different colors and send it to the LED display.
- **Task 3:** Write a text in green color and send it to the

LED display.

- **Task 4:**The user is shown a new system (see figure 3.21). Change the layout settings to match the new LED display. Change the layout settings to match the new system.
- **Task 5:** Draw a pattern with at least three colors and send it to the LED display.
- **Task 6:** Write a text in another color than green and send it to the LED display.

The user tests showed that the users had little to no troubles completing all the assigned tasks. Due to the low learning curve all users completed task 5 and 6 with no difficulty.

After the user test, the participants were asked to fill out the System Usability Scale (SUS)[[Bro96]] questionnaire (see Appendix E—"SUS - A quick and dirty usability scale").

Definition:
System Usability Scale (SUS)

SYSTEM USABILITY SCALE (SUS) :
The System Usability Scale (SUS) is a simple, ten-item scale giving a global view of subjective assessments of usability.

The results of the scale are from 0 to 100. A higher score means better usability. The final score of the interface was 91.5. More details about the SUS usability scale and the results for each user can be found in Appendix E—"SUS - A quick and dirty usability scale".

Suggestions And Feedback From The Users Users suggested additional features for the "Draw" tab of the app. Since it is difficult to pick the exact same color in the current color picker, an eyedropper tool would be useful to pick a previously used color. Another user proposed to offer different color picker interfaces, one with preset colors or with the possibility to insert hexadecimal or RGB values. Users also wanted undo functionality, for example, tapping a pixel twice turns it back to the previous color. The bucket tool was misinterpreted, and users expected it to fill only pixels with the same color rather than the entire screen. In the "Write" tab not many users did not notice the Clear text button.

There were many users that were not used to iPhones and had troubles understanding when the task 4 was completed. Most of them expected a button to apply the settings. They were also confused with the keyboard used for the row and column number text fields. They could not find the button to dismiss the keyboard.

On another note, they were fascinated about the idea of having clothes with displays and being able to change the image or text through their phone. They also found the comparison with graphics design software appropriate.

Final User Interface After the user test we made the following changes o the interface (see figure 4.1).

- A paintbrush tool was added to the tools.
- The inconsistencies between the writing tab and the drawing tab were removed.
- A button to apply the layout was added to the Settings tab
- The keyboard for editing the column and row number was changed to the classical number keyboard.
- The application gives more feedback through alerts.

Figure 3.22: LumiNet 2 app v 1.0.

Chapter 4

Evaluation

4.1 User Studies

To verify the main requirements, we used two different studies. One evaluates the usability of the iPhone app and the other shows the improvement in integrability of the hardware from LumiNet 1 to LumiNet 2.

4.1.1 User Interface Study

In order to determine the usability of the iPhone app we conducted a second formal user study. We wanted to know that the system is intuitive and easy to use. We are using as similar user test as the initial study. The tasks given to the users have been changed to verify if the offered tools are recognized and used.

User study tasks:

- **Task 1:** Open the "LumiApp" application and verify if it is connected to the LED display.
- **Task 2:** Draw a green line, blue line and red line then send it to the LED display.

- **Task 3:** Use the same green that was used before and draw a dot on the red line and send it to the LED display.
- **Task 4:** Use the same blue as before to paint over the complete display and send it to the LED display.
- **Task 5:** Write "Hello World!" in green color and send it to the LED display.
- **Task 6:** The user is shown a new system (see figure 3.21). Change the layout settings to match the new LED display.
- **Task 7:** Draw a pattern with at least three colors and send it to the LED display.
- **Task 8:** Write a text in another color than green and send it to the LED display.

The user tests showed that the users had little to no troubles completing all the assigned tasks. After the user test, the participants were asked to fill out the SUS questionnaire.

The final score of the SUS questionnaires was 94.75, slightly better than of the previous version. More details about the SUS usability scale and the results for each user can be found in Appendix E—"SUS - A quick and dirty usability scale".

Suggestions And Feedback From The Users

Users that did not notice some of the functions from the first tasks, explored and used them in the second task. There were still some minor issues, the users did not recognize the roll button function.

Proposed features:

- scrolling text on the custom pattern
- loading videos
- saving and opening layouts

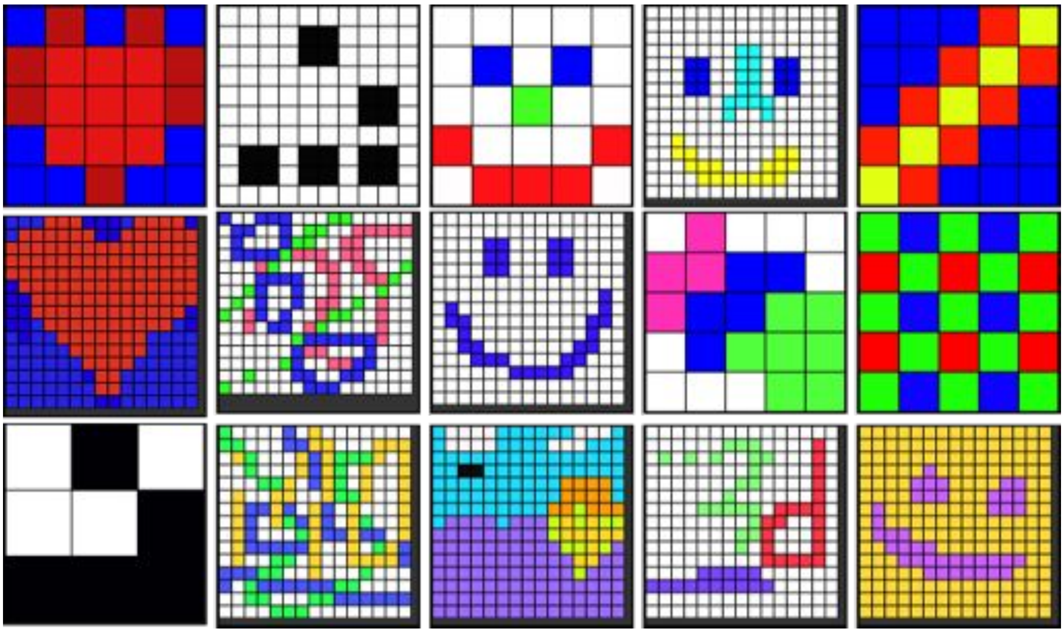


Figure 4.1: LumiNet 2 drawings.

- creating animations frame by frame
- having the colors synchronized so that you can choose the same color from the drawing for the text
- adding borders around the canvas and also around the color button to distinguish them even when black is chosen or drawn.

In figure 4.1 we present some of the designs made by users in the Draw tab.

In conclusion the users found the interface simple, friendly and easy to use. They thought the drawing metaphor for choosing the pattern was great and easy to understand.

4.1.2 Integrability Improvement Study

To test the new hardware design against the LumiNet 1 hardware we used a comparative study. The users were given a set of tasks to complete and in the end fill a questionnaire related to their experience with sewing, soldering, connecting the LumiNet 1 and 2 nodes, as well as some demographic questions. For the test we had some fashion designers and other users that have various backgrounds and ages.

Phase I tasks:

- **Task 1:** Sew two LumiNet 2 nodes to each other with conductive thread and check for short circuits with the multimeter.
- **Task 2:** Sew two LumiNet 1 nodes to each other with conductive thread and check for short circuits with the multimeter.

Phase II tasks:

- **Task 1:** Solder conductive ribbon to one side of a LumiNet 2 board and then to the side of a second LumiNet 1 board. Check for short circuits with the multimeter.
- **Task 2:** Solder conductive ribbon to one side of a LumiNet 1 board and then to the side of a second LumiNet 1 board. Check for short circuits with the multimeter.
- **Task 3:** Arrange on a piece of paper four LumiNet 1 nodes and draw how they connect to each other. Note - connect only 4 pads to 4 pads.
- **Task 4:** Arrange on a piece of paper four LumiNet 2 nodes and draw how they connect to each other.Note - connect only 4 pads to 4 pads.

Phase I

During the first phase the users had to sew the nodes as shown in figure 4.2.The fashion designers managed to

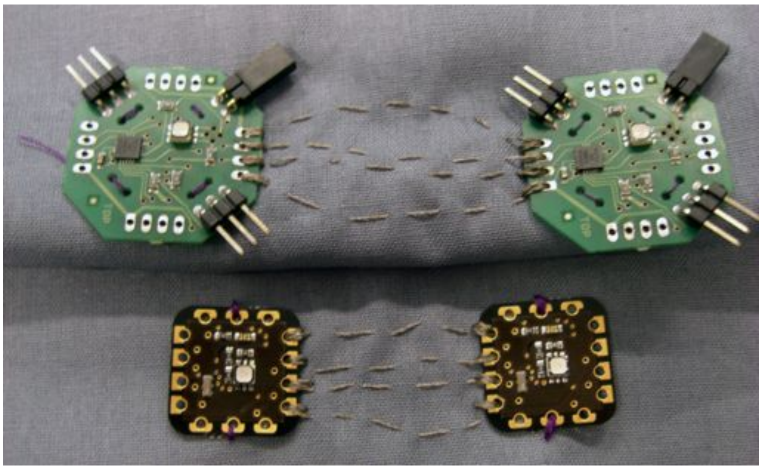


Figure 4.2: LumiNet nodes connected through sewing.

sew the LumiNet 2 node without any connections but not LumiNet 1, they said the holes were too small and close together to be able to connect it nicely. Some users with no experience in sewing took up to two hours to finish the first task, not managing to connect LumiNet 1 and sometimes even LumiNet 2 without short circuits. For sewing LumiNet 1 the users also had problems with the jumper and the programming pins sticking out, the thread was continuously caught in them.

Phase II

The second phase consisted of soldering the nodes together with the Amohr ribbon as shown in figure 4.3. Phase II took less time, even for novices. They thought connecting the ribbon was easier then sewing. They had problems while soldering LumiNet 1 due to the spacing not matching the pads but there were no short circuits.

Some of the final comments were:

- "LumiNet 2 is friendly for the usage in textile has no pointing out scratching pins. I like working with it."

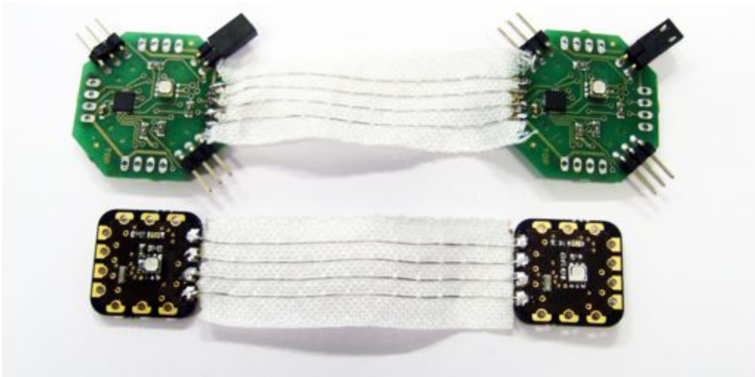


Figure 4.3: LumiNet nodes connected through soldering.

- "I like tho color and appearance of the LumiNet 2 nodes better"
- "For deploying in fashion wear the chip size should be even smaller, but it would be difficult to sew. if the sewing part can be done by a machine, then it should be fine."
- "The size of the conductive ribbon was matching nicely the connection points of the LumiNet2."

In conclusion LumiNet 2 was the preferred platform both for sewing and soldering. More details about the questionnaire data can be found in Appendix F—"Hardware Comparison".

4.2 Secondary Requirements

While analyzing the secondary requirements we can show that they have been met implicitly.

R1 Extensibility. The modularity and clear separation of the system allows each part to be updated and changed independently. The design gives great flexibility to empower developers to build new interfaces on top of the existing structure.

R2 Simpler node connections. By decreasing the number of possible connections, we simplified the topology making it easier to grasp.

R3 Fast integration. the Amohr conductive ribbon was found ideal for fast integration. It provides both easy connection to the nodes through soldering and attachment through ironing it on.

R4 Sensor support. Attaching more sensors is possible both in hardware and software. The board allows up to 5 pins for sensors.

R5 Full Arduino compatibility. Because we are using the original Arduino core bootloader, the notes can be programmed also using Arduino.

R6 Fast and reliable communication. Serial communication is implemented in the micro-controller and supported by hardware interrupts making the communication more reliable. Because Bynase is more reliable if the data is repeated, serial is probably faster at the same reliability level.

R7 Platform independent. Except for the drawing interface, due to the platform independent languages used, LumiNet 2 is hardware independent.

R8 Open source hardware and software. We will publish the eagle files and source code for the LumiNet 2 nodes on the luminet.cc[1] website.

R9 Fast firmware loading. The pogo pin adapter allows for very fast loading of both the bootloader and the firmware.

[1] luminet.cc

4.3 Hardware Integrability

For fashion, the aspect of the clothes is very important and the designers should have versatile options of how to integrate the electronics. LumiNet 2 has already been integrated in a few projects that we will shortly present in the next paragraphs.

4.3.1 Ginko Leafs Necklace

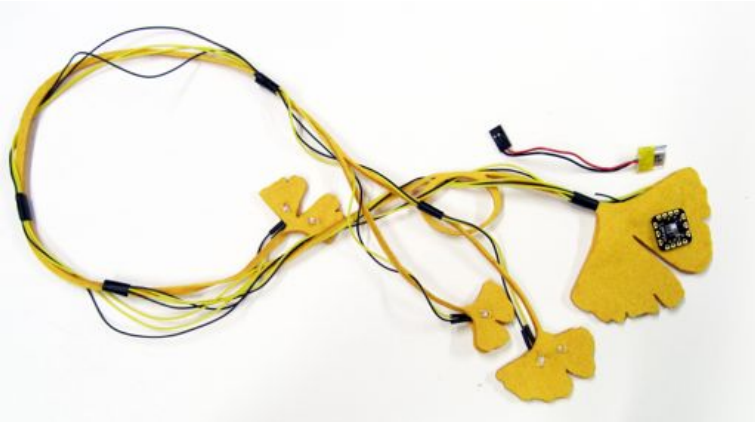


Figure 4.4: Necklace concept that uses LumiNet2 in plain sight.

Larisa Katz[2] liked the aesthetic of the board and created a necklace which has the board in plain sight. It gives the necklace a modern look. Each leaf has one or two LEDs and the leafs light up in a random order with a soft fading.

4.3.2 Light Jacket

The light jacket is a collaboration with Institut für Textiltechnik Aachen (ITA)[3] , we provided the electronic parts and they embedded them into the jacket. It has one

[2]http://www.larisakatz.com/
[3]http://www.ita.rwth-aachen.de/

Figure 4.5: Luminet2 node controlling an 8 by 8 matrix of LED made with addressable flexible LED strips.

LumiNet 2 node that controls two meters of RGB LED strip. The string that scrolls on the jacket according to a hard coded value.

4.3.3 Exeriments

We experimented with methods to conceal the electronics in clothes and also on creating set programming interfaces.

4.3.4 Soft Programmer

The soft programer is laser cut to shape and assembled with interfacing material. It then gets the snaps punched in for easy connection. The interface is completely concealed under the fabric and only the snaps are visible.

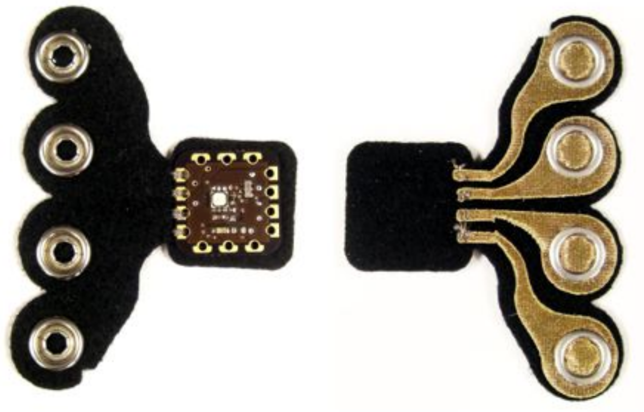


Figure 4.6: LumiNet2 soft programming interface.

Figure 4.7: Luminet2 prototype design and accessories v1.2 with FIMO flower embellishments.

FIMO Flowers

FIMO[4] is a modeling clay that hardens when placed in the oven for 30 minutes at 100 degrees celsius. We used FIMO to hide the LumiNet 2 hardware leaving only the LED visible. We also created accessories for it, two LEDs and a light sensor (see figure 4.7).

[4] http://fimo.com/

Fabric Flowers

LumiNet 2 also fits well under fabric embellishments like a fabric flower. We created a flower to decorate the node (see figure 4.8). The reason it fits so well is that the node has a very flat profile. Fashion designers can create more complex patterns or shapes and insert the nodes.

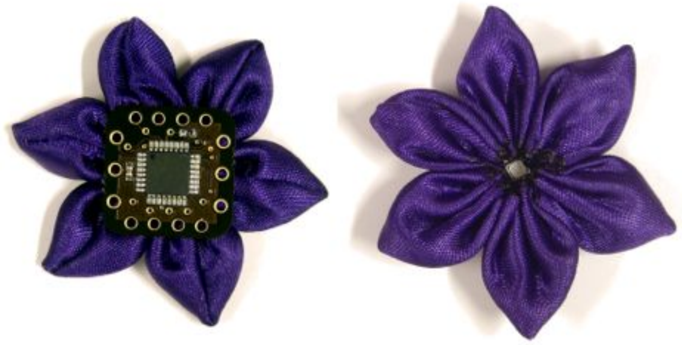


Figure 4.8: Luminet2 covered with a flower made of fabric.

4.4 Conclusion

LumiNet 2 was proven versatile in many ways. Due to the small footprint combined with accessible sew taps and low weight, it is ideal for use with fabrics. In addition, the firmware gives it great flexibility to be used though mobile devices.Based on the results from the user tests we consider that the resulting system adequately answered the research questions that were asked at the beginning in section 3.1—"Research Questions".

Chapter 5

Summary and future work

"The Web as I envisaged it, we have not seen it yet. The future is still so much bigger than the past."

—Tim Berners-Lee

This work produced an accessible system for distributed computing on clothes. The LumiNet 2 nodes are thin, small and easy to add to clothing. In addition, it comes with a simple and clean interface for visual reprograming.

5.1 Summary and contributions

Wearable computing is a young field of study full of possibilities. This work presented the landscape of wearable DIY frameworks, and showed the options available to users. It also gave an overview of the positive aspects as well as possible improvements. We then presented a comprehensive comparison of the frameworks, including the LumiNet 2 nodes. There are many frameworks available but not many distributed systems. Distributed systems are useful when there are many similar tasks that need to be accomplished in parallel independently from one another. Considering all the qualities of the presented frameworks, we then decided on what would be ideal to have in our hardware design. Then we introduce a short history of displays being used in fashion where we state the existing interest in displays. Thus motivating our efforts in empowering the designers by making the programming more natural for them.

Research questions were then identified. They focused on how can the board be integrated into fashion while remaining robust and how the resulting hardware can be used without writing code.

Hardware prototypes have been created and tested. After reaching the desired requirements we then ordered professional printed boards. In parallel we defined the requirements for the firmware and software. We then made a first prototype of the app that should program the network. The app is kept very simple, allowing the users to ether draw in a pattern to be displayed or to type a text that will scroll on the display. The settings are also kept minimal. After an initial user study we improved the usability of the system. Another user study showed that the users considered the app self explanatory and easy to use. They were also very interested on having a display that can be controlled through a mobile phone.

5.2 Future work

This thesis introduces a different way of interacting with distributed systems but, the hardware and the software can be further developed.

Different Nodes

Different nodes can be created to be used with the framework. The current node is a generic node that can be used for different functions. One idea would be a simplified node with only the LED and no other sensor and an AtTiny.

It would then have a really small footprint and decrease the spacing needed between the pixels. Another thought could be a node with proximity detection, or touch detection. Many other customizations can be done to make the system even more versatile.

Hardware Tests

The system could be tested with only a low number of nodes. It is interesting how it would react with many nodes. It should be determined if more optimizations need to be done for increasing the speed or for better synchronization of the nodes.

User Tests

More detailed user tests should be performed to determine what the future direction of the system could be and determine more applications for it.

User Interface

Due to the high importance placed on the hardware and firmware, the user interface is a proof of concept and can be developed further. In the evaluation chapter we had many feature requests from the users that could greatly improve the functionality.

Additional Applications

The firmware supports a plethora of sensors, which are not used in the current application. Different applications can be built on top of the firmware to take full advantage of the possible interactions.

Appendix A

LED Matrix Skirt

A skirt inspired by Leah Buechley's tank top.

The led sequins were made with 0603 package green SMD LED and crimp beads (quetschperlen) for the connections. Silver was attached to the + (Vcc) and golden for - (GND)(see figure A.1). To help bind them together a two component epoxy transparent glue was used.

Figure A.1: Self made LED sequin closeup.

Like Leah, we used Multiplexing for the 8 by 8 LED display. The the rows are connected to the + of the

LED and columns to the -. The traces are separated by the material, the rows have the conductive thread on the bottom and the columns on the top side (see figure A.2).

Figure A.2: Skirt display top side.

An Arduino LilyPad controls the LED matrix (see figure A.3) with a library created for the Arduino environment to handle scrolling strings.

Figure A.3: Skirt display back side detail.

The library contains a complete representation of a 8x5 pixels font for all the visible characters(see figure A.4).

!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~

Figure A.4: Skirt display font listing.

In the end, we had a good result (see figure 3.3).

Appendix B

Luminet 2 v2.0 Hardware Design

For the board we used the following parts:

Nr. crt.	Part name	Reichelt[1] part no.	Farnell[2] part no.
1	AtMega 168PA	ATMEGA 168-20 TQ	
2	RGB Led	LRTB G6TG	
3	2 x Resistor 56 ohm 0603	SMD-0603 56	
4	Resistor 180 ohm 0603	SMD-0603 180	
5	Resonator SMD 8Mhz	CSTCE 8,00	1615352RL
6	Capacitor 10 μF	X5R-G0603 10/6	
7	Phototransistor		1814879RL
8	Resistor 56k ohm 0603	SMD-0603 56K	
9	Resistor 10k ohm 0603	SMD-0603 10K	

Table B.1: Bill of materials.

[1] http://www.reichelt.de/
[2] http://de.farnell.com/

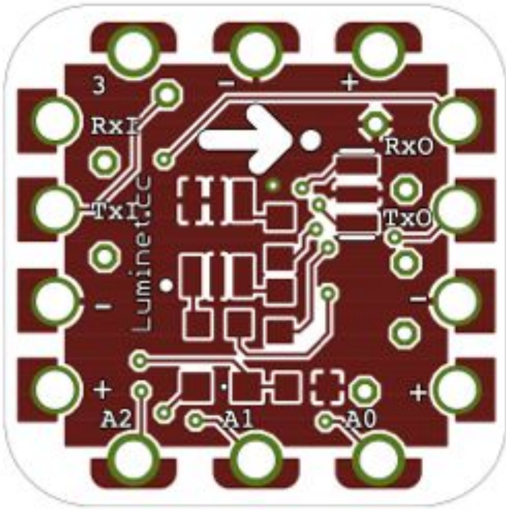


Figure B.1: LumiNet 2 v2.0 hardware board top layer.

Figure B.2: LumiNet 2 v2.0 hardware board bottom layer.

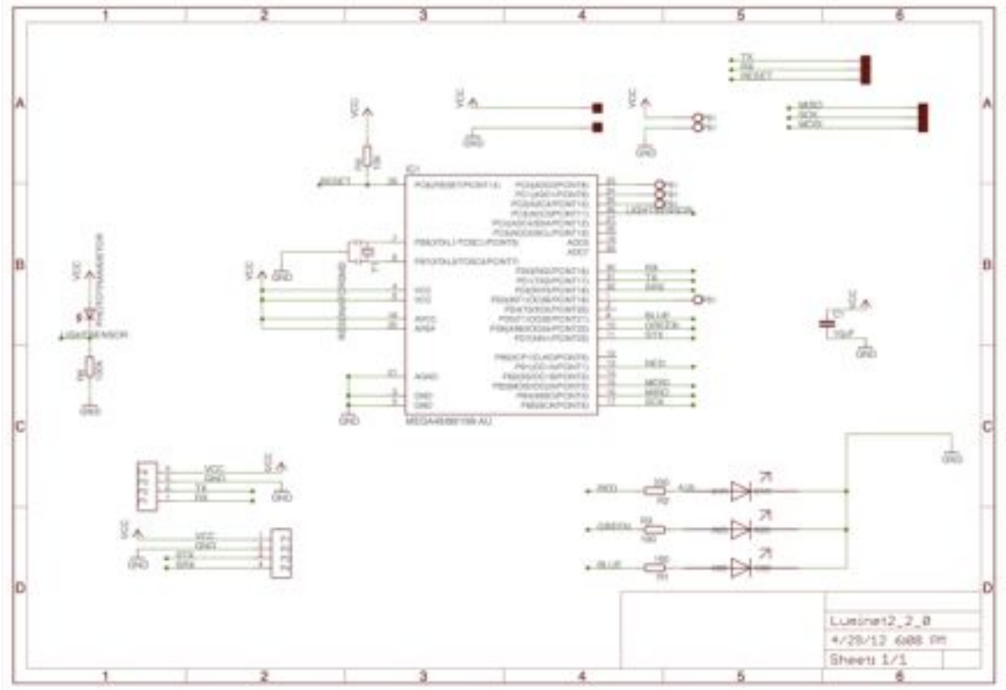


Figure B.3: LumiNet 2 v2.0 hardware board schematic.

Appendix C

Amohr conductive ribbon

This chapter shows details on the Amohr[1] conductive ribbon recommended for LumiNet 2 v2.0.

Ihre Bestellung vom:	26.03.12	Ihre Email
Einteilung Nr.:	21200389	

Pos	Artikel	Breite	Typ
10	**45385**	**21 mm**	**00090**
	Zolltarif-Nr.:		5806 32 10
	Kontaktierband mit 4x20er CU verzinnt		

Figure C.1: Clipping from the receipt that we received with the Amohr conductive ribbon.

On the next page we are attaching the data sheet for the conductive ribbon that was modified from 11 traces to 4 traces.

[1] http://www.amohr.com

AMOHR Technische Textilien GmbH

art. 3586 – 21mm

Conductive tape	**Leitfähiges Band**
This standard article is a woven tape made from Polyester yarn and includes 11 tin-plated copper strands. The tape contains hotmelt material on the back side and can be ironed onto almost any surface. The electrical resistance per meter is app. 0,05 OHM. The tape can be adapted in width and no. or material of strands on request. The price of the sample is € 0,80/m for minimum orders of 2.000m. Smaller quantities will be charged with a flat rate.	Dieser Standardartikel wird aus Polyestergarn und 11 verzinnten Kupferleitern gewebt. Durch Schmelzklebermaterial kann das Band auf fast jede Oberfläche aufgebügelt werden. Der elektrische Widerstand beträgt nur ca. 0,05 OHM/m. Das Band kann in Breite, Leiterzahl und –beschaffenheit kundenspezifisch angepasst werden. Der Preis des Musters beträgt €0,80/m für eine Mindestmenge von 2.000m. Kleinere Mengen werden pauschal berechnet.

Christoph Mohr 09.03.2009

Appendix D

Pogo Pin Adapter

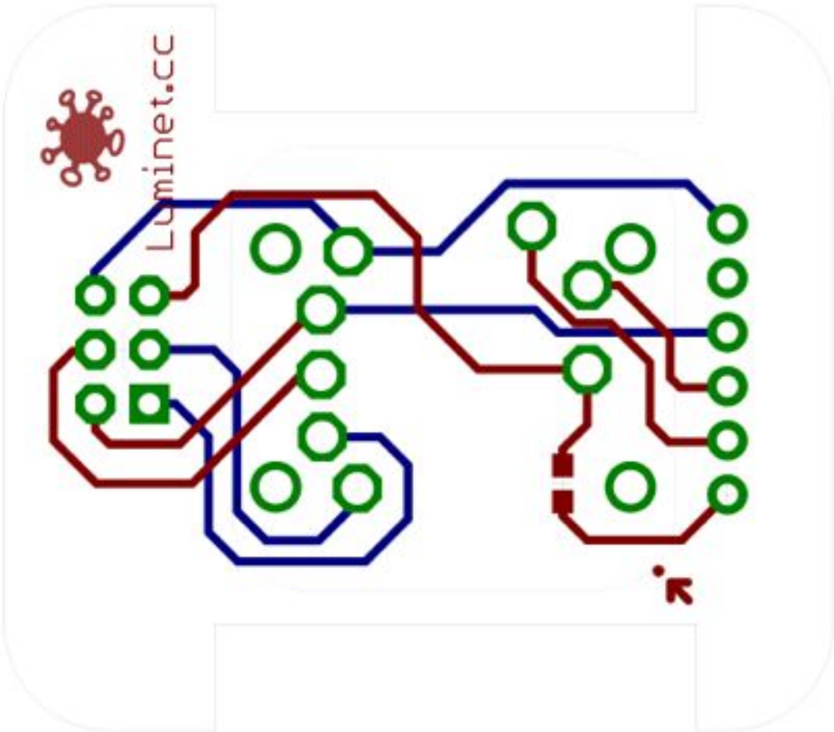


Figure D.1: Pogo pin adapter design.

Appendix E

SUS - A quick and dirty usability scale

E.1 SUS

This section explains the usage of SUS and the scoring using excerpts from the publication "SUS - A quick and dirty usability scale."[Bro96].

E.1.1 Using SUS

"The SU scale is generally used after the respondent has had an opportunity to use the system being evaluated, but before any debriefing or discussion takes place. Respondents should be asked to record their immediate response to each item, rather than thinking about items for a long time.

All items should be checked. If a respondent feels that they cannot respond to a particular item, they should mark the centre point of the scale."

System Usability Scale

	Strongly disagree 1	2	3	4	Strongly agree 5
1. I think that I would like to use this system frequently					
2. I found the system unnecessarily complex					
3. I thought the system was easy to use					
4. I think that I would need the support of a technical person to be able to use this system					
5. I found the various functions in this system were well integrated					
6. I thought there was too much inconsistency in this system					
7. I would imagine that most people would learn to use this system very quickly					
8. I found the system very cumbersome to use					
9. I felt very confident using the system					
10. I needed to learn a lot of things before I could get going with this system					

Figure E.1: SUS questionnaire.

E.1.2 Scoring SUS

"SUS yields a single number representing a composite measure of the overall usability of the system being studied. Note that scores for individual items are not meaningful on their own.

To calculate the SUS score, first sum the score contributions from each item. Each item's score contribution will range from 0 to 4. For items 1,3,5,7,and 9 the score contribution is the scale position minus 1. For items 2,4,6,8 and 10, the contribution is 5 minus the scale position. Multiply the sum of the scores by 2.5 to obtain the overall value of SU.

SUS scores have a range of 0 to 100."

E.2 Results

From our user tests we got the results shown in tables E.2—"The results for the second user test calculated as SUS scores." and E.2—"The results for the second user test calculated as SUS scores.".

E.2.1 First test

The first test was done mainly with computer science students and a few novice users. The calculated average SUS score for the first user test is 91.5.

User ID	1	2	3	4	5	6	7	8	9	10
Question 1	4	2	3	4	3	4	4	3	4	4
Question 2	4	4	3	4	3	4	4	3	3	4
Question 3	3	4	4	4	4	4	4	2	3	4
Question 4	3	4	4	4	4	4	4	3	4	4
Question 5	4	4	3	4	3	4	4	3	2	4
Question 6	4	4	4	4	4	4	4	4	4	4
Question 7	4	4	3	4	4	4	4	1	3	4
Question 8	4	4	4	4	4	4	4	3	4	4
Question 9	3	4	4	4	3	3	3	2	3	4
Question 10	4	4	4	4	4	4	4	4	3	4
Score	92.5	95	90	100	90	97.5	97.5	70	82.5	100

Table E.1: The results for the first user test calculated as SUS scores.

E.2.2 Second test

The calculated average SUS score for the second user test is 94.75.

User ID	1	2	3	4	5	6	7	8	9	10
Question 1	4	4	3	4	4	3	2	3	4	4
Question 2	4	4	4	4	4	4	3	4	4	4
Question 3	4	4	4	4	4	4	4	4	4	4
Question 4	4	4	4	4	4	3	4	4	4	4
Question 5	4	3	4	4	3	3	3	3	4	4
Question 6	4	4	4	4	4	4	2	4	4	4
Question 7	4	3	4	4	4	4	3	4	4	4
Question 8	4	4	4	4	4	4	4	4	4	3
Question 9	4	4	4	4	4	3	4	4	4	4
Question 10	3	4	3	4	4	4	4	3	4	4
Score	97.5	95	95	100	97.5	90	82.5	92.5	100	97.5

Table E.2: The results for the second user test calculated as SUS scores.

For the second user test we asked the users to fill in a questionnaire related to the user's experience with smart phones and graphic design software as well as some demographic questions.

The ages were in average 27 and between 23 and 37.

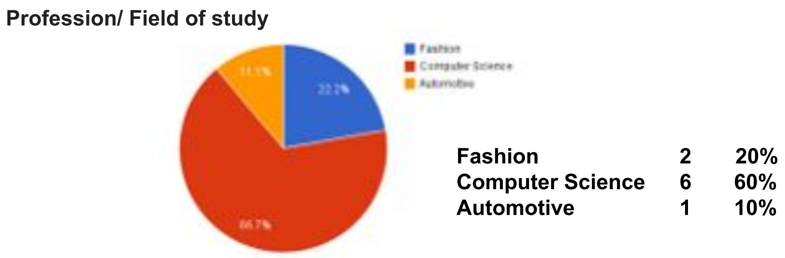


Figure E.2

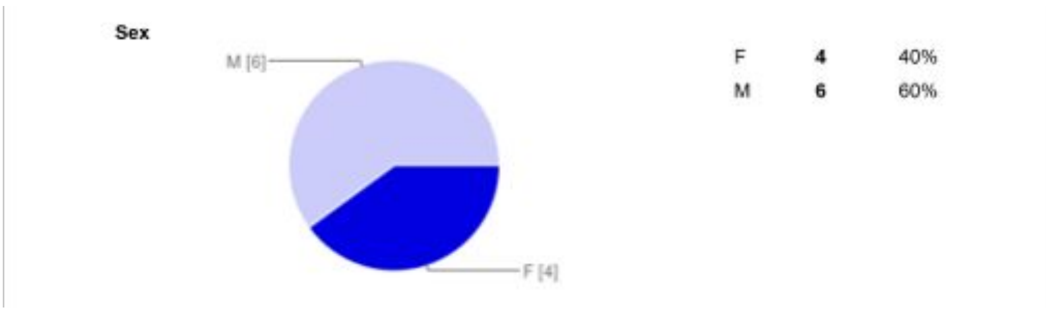


Figure E.3

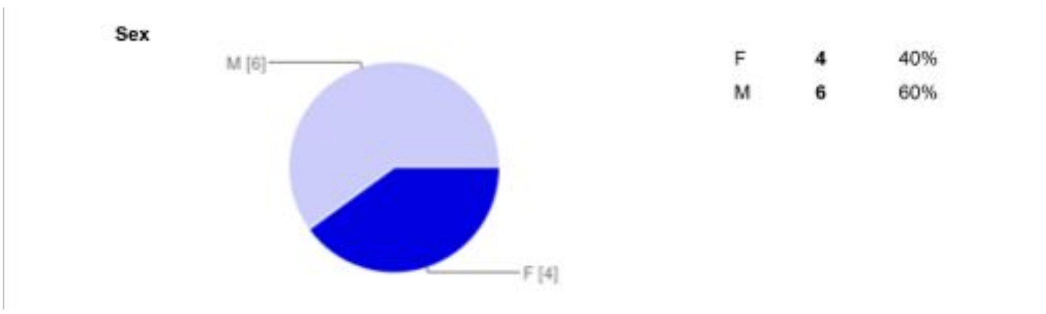


Figure E.4

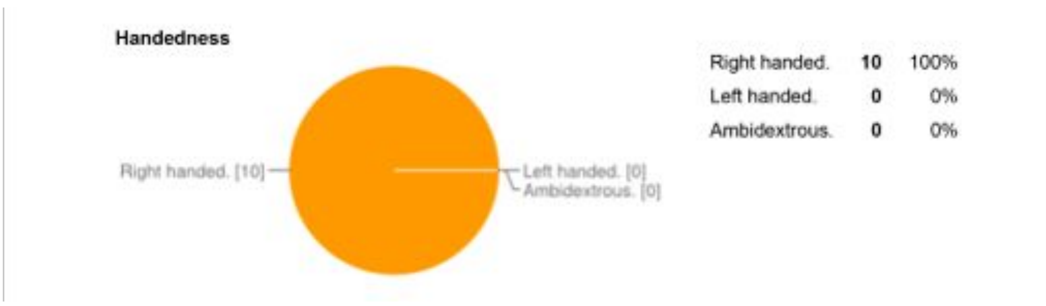


Figure E.5

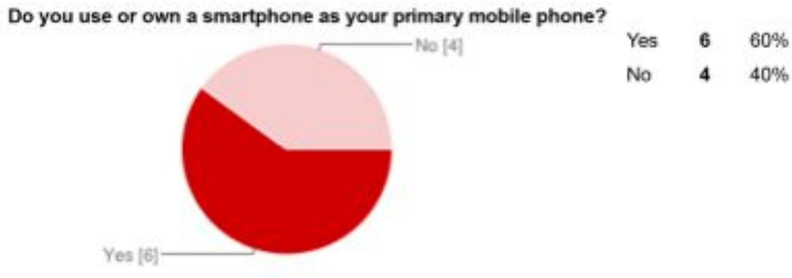


Figure E.6

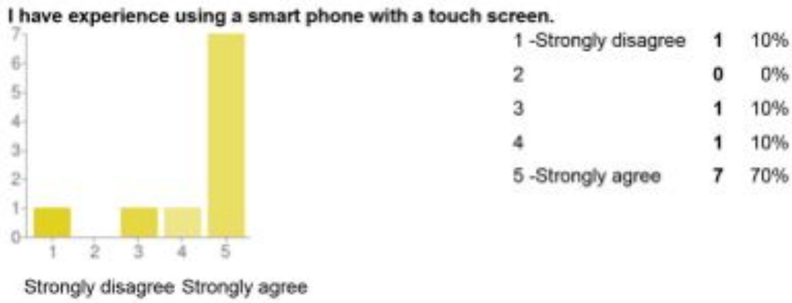


Figure E.7

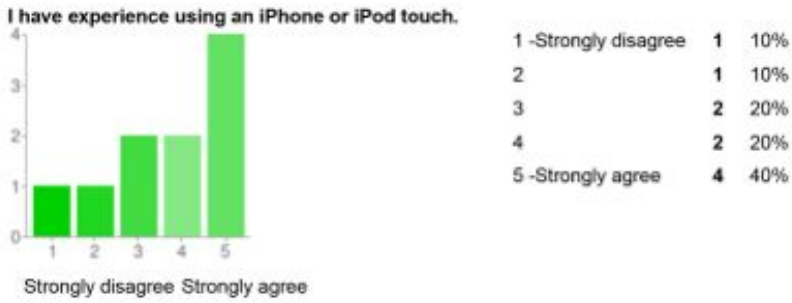
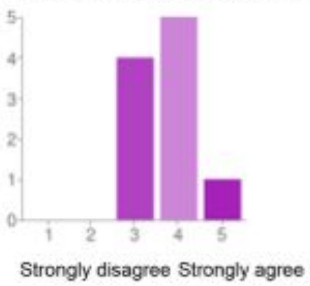


Figure E.8

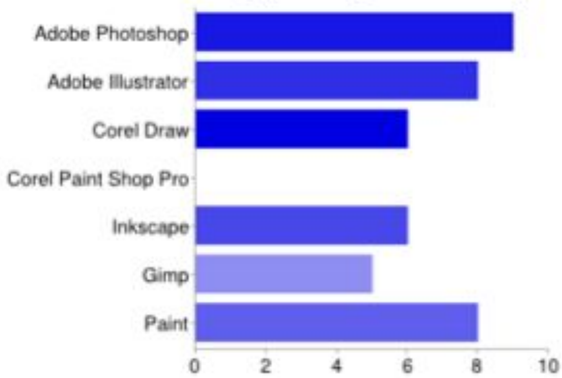
I feel confident using graphic design software.



1 -Strongly disagree	**0**	0%
2	**0**	0%
3	**4**	40%
4	**5**	50%
5 -Strongly agree	**1**	10%

Figure E.9

Select from the list all the graphic design software that you have used previously.



Adobe Photoshop	**9**	90%
Adobe Illustrator	**8**	80%
Corel Draw	**6**	60%
Corel Paint Shop Pro	**0**	0%
Inkscape	**6**	60%
Gimp	**5**	50%
Paint	**8**	80%

People may select more than one checkbox, so percentages may add up to more than 100%.

Figure E.10

Appendix F

Hardware Comparison

The ages were in average 27 and between 23 and 37.

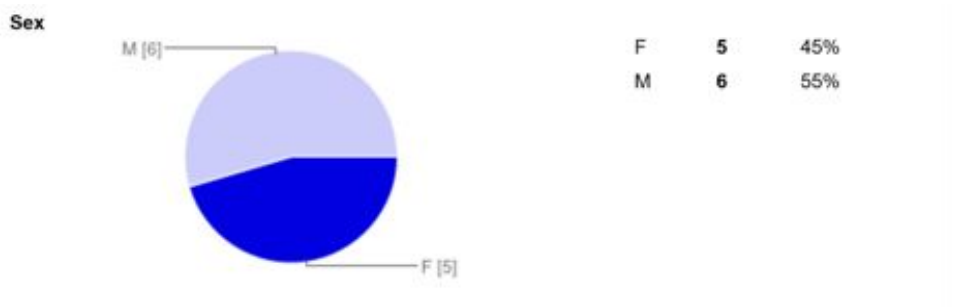


Figure F.1

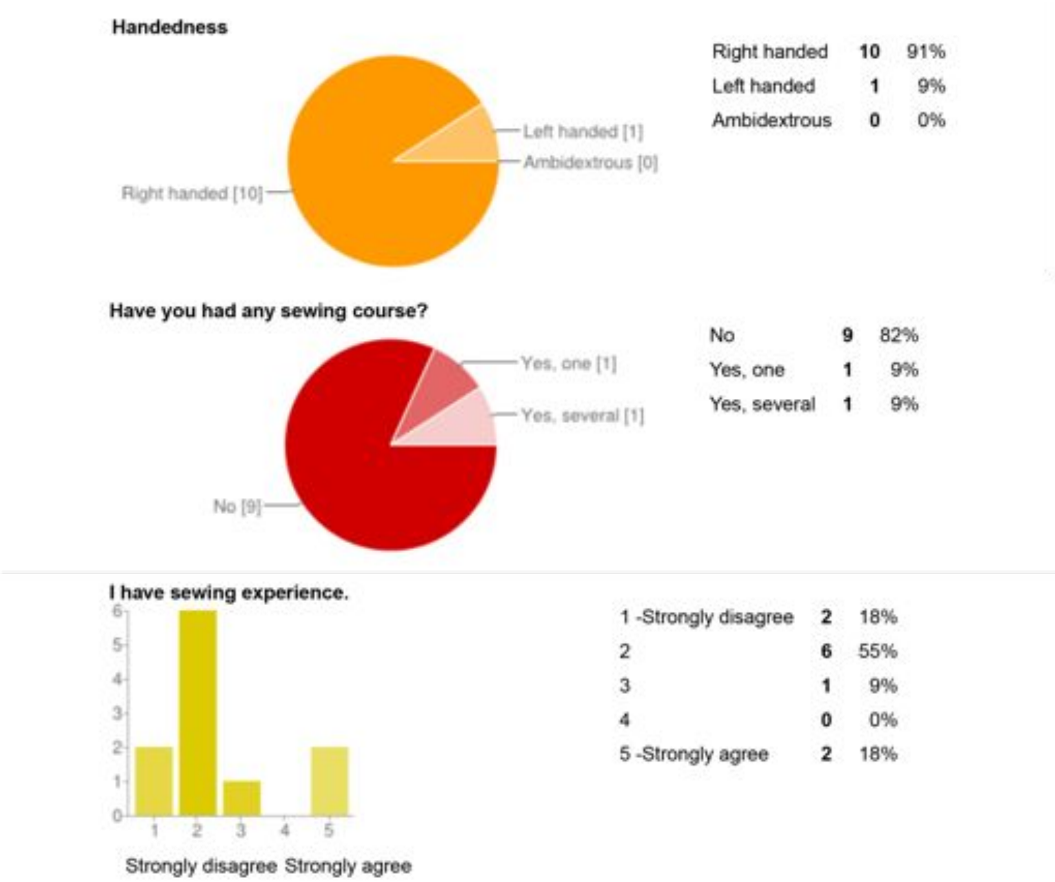
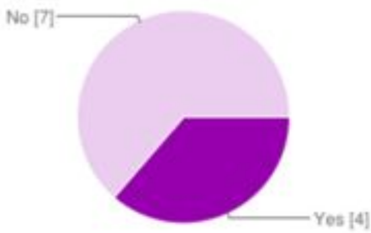


Figure F.2

I sew often.

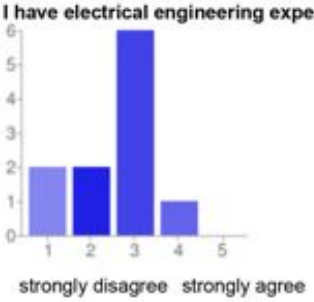
1 -Strongly disagree	**7**	64%
2	**2**	18%
3	**1**	9%
4	**0**	0%
5 -Strongly agree	**1**	9%

Have you had any soldering course?



Yes	**4**	36%
No	**7**	64%

I have electrical engineering experience.



1 -strongly disagree	**2**	18%
2	**2**	18%
3	**6**	55%
4	**1**	9%
5 -strongly agree	**0**	0%

Figure F.3

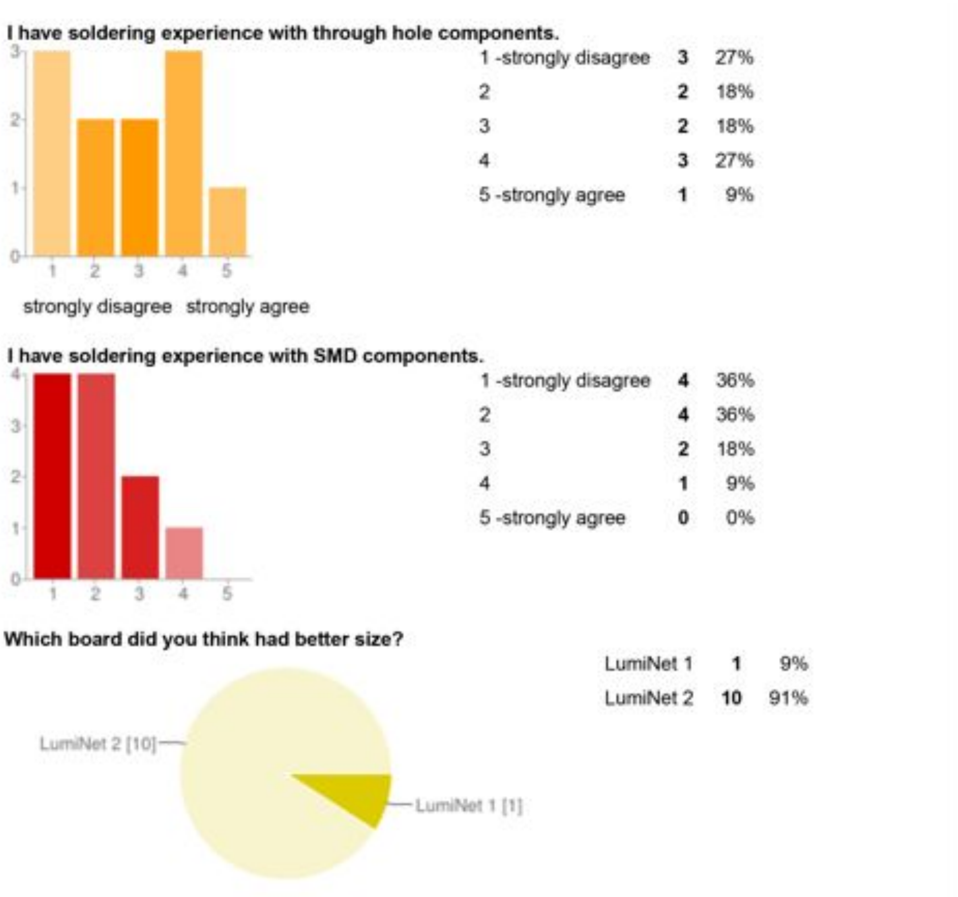

I have soldering experience with through hole components.
1 -strongly disagree 3 27%
2 2 18%
3 2 18%
4 3 27%
5 -strongly agree 1 9%
strongly disagree strongly agree
I have soldering experience with SMD components.
1 -strongly disagree 4 36%
2 4 36%
3 2 18%
4 1 9%
5 -strongly agree 0 0%
Which board did you think had better size?
LumiNet 1 1 9%
LumiNet 2 10 91%
LumiNet 2 [10]
LumiNet 1 [1]

Figure F.4

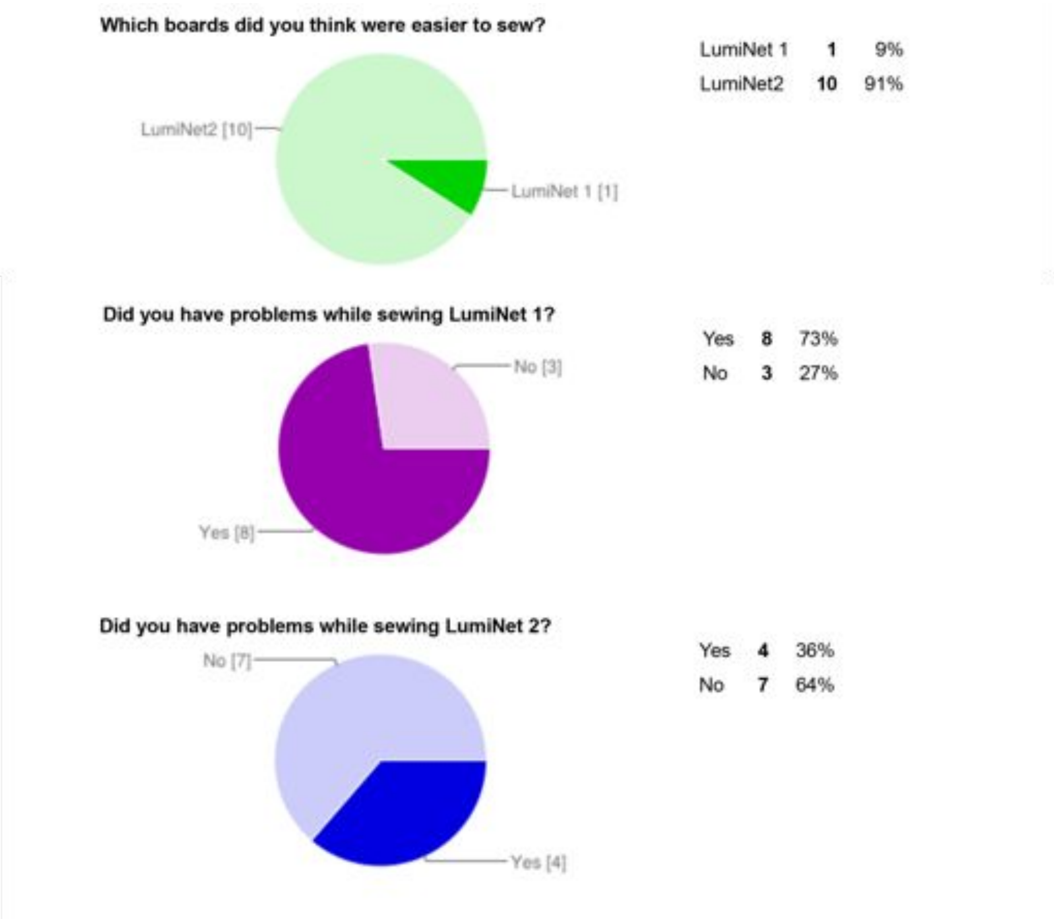

Which boards did you think were easier to sew?
LumiNet2 [10]
LumiNet 1 [1]
LumiNet 1 1 9%
LumiNet2 10 91%
Did you have problems while sewing LumiNet 1?
No [3]
Yes [8]
Yes 8 73%
No 3 27%
Did you have problems while sewing LumiNet 2?
No [7]
Yes [4]
Yes 4 36%
No 7 64%

Figure F.5

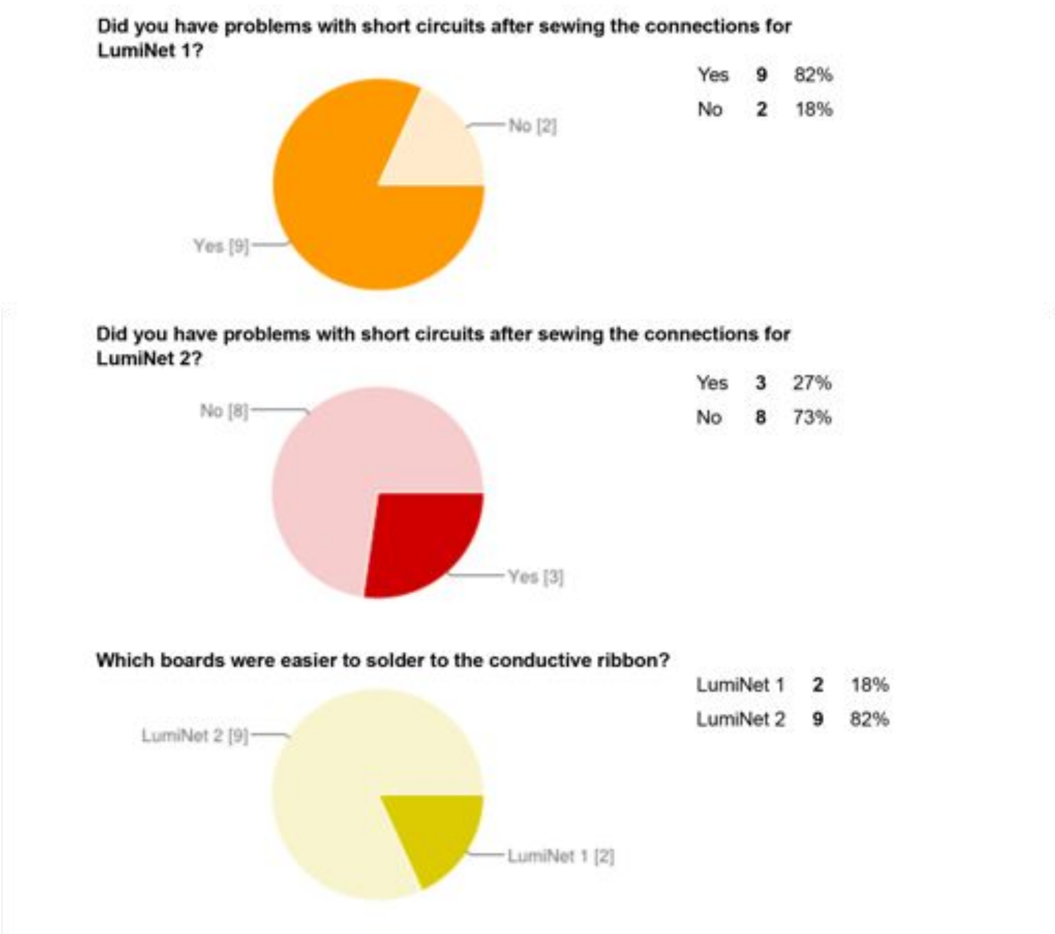

Did you have problems with short circuits after sewing the connections for LumiNet 1?
Yes 9 82%
No 2 18%
No [2]
Yes [9]
Did you have problems with short circuits after sewing the connections for LumiNet 2?
Yes 3 27%
No 8 73%
No [8]
Yes [3]
Which boards were easier to solder to the conductive ribbon?
LumiNet 1 2 18%
LumiNet 2 9 82%
LumiNet 2 [9]
LumiNet 1 [2]

Figure F.6

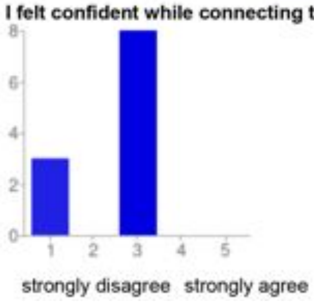
Did you have problems with short circuits after soldering the connections for LumiNet 1?

Yes	**0**	0%
No	**11**	100%

Did you have problems with short circuits after soldering the connections for LumiNet 2?

Yes	**0**	0%
No	**11**	100%

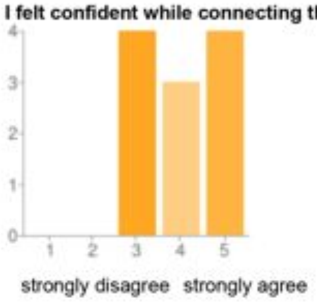
I felt confident while connecting the LumiNet 1 nodes.



1 -strongly disagree	**3**	27%
2	**0**	0%
3	**8**	73%
4	**0**	0%
5 -strongly agree	**0**	0%

Figure F.7

I felt confident while connecting the LumiNet 2 nodes.



1 -strongly disagree	**0**	0%
2	**0**	0%
3	**4**	36%
4	**3**	27%
5 -strongly agree	**4**	36%

Figure F.8

Bibliography

[Ard05] Arduino, 2005.

[BBH09] Jan Borchers, René Bohne, and Gero Herkenrath. LumiNet : An Organic Interactive Illumination Network. Master's thesis, RWTH Aachen University, Germany, 2009.

[BECC08] Leah Buechley, Mike Eisenberg, Jaime Catchen, and Ali Crockett. The lilypad arduino: using computational textiles to investigate engagement, aesthetics, and diversity in computer science education. In *Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, CHI '08, pages 423–432, New York, NY, USA, 2008. ACM.

[BH10] Leah Buechley and Benjamin Mako Hill. Lilypad in the wild: how hardware's long tail is supporting new engineering and design communities. In *Proceedings of the 8th ACM Conference on Designing Interactive Systems*, DIS '10, pages 199–207, New York, NY, USA, 2010. ACM.

[Bro96] J. Brooke. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189:194, 1996.

[Bue06] Leah Buechley. A construction kit for electronic textiles. In *Wearable Computers, 2006 10th IEEE International Symposium on*, pages 83–90. IEEE, 2006.

[CKV11] Sylvia H. Cheng, Kibum Kim, and Roel Vertegaal. Tagurit: a proximity-based game of tag using lumalive e-textile displays. In *Proceedings of the 2011 annual conference extended abstracts on*

Human factors in computing systems, CHI EA '11, pages 1147–1152, New York, NY, USA, 2011. ACM.

[DP10] Zach Eveland Despina Papadopoulos. Fabrickit, 2010.

[EE08] N. Elumeze and M. Eisenberg. Buttonschemer: ambient program reader. In *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*, pages 323–326. ACM, 2008.

[Fil10] Seeeduino Film, 2010.

[GKS+98] F. Gemperle, C. Kasabach, J. Stivoric, M. Bauer, and R. Martin. Design for wearability. In *Wearable Computers, 1998. Digest of Papers. Second International Symposium on*, pages 116–122. IEEE, 1998.

[HV08] David Holman and Roel Vertegaal. Organic user interfaces: designing computers in any way, shape, or form. *Commun. ACM*, 51(6):48–55, June 2008.

[HVSS12] S. Hodges, N. Villar, J. Scott, and A. Schmidt. A new era for ubicomp development. *Pervasive Computing, IEEE*, 11(1):5–9, 2012.

[KB10] B. Kaufmann and L. Buechley. Amarino: a toolkit for the rapid prototyping of mobile ubiquitous computing. In *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services*, pages 291–298. ACM, 2010.

[KK06] Mike Kuniavsky and Tod E. Kurt, 2006.

[KP01] K. KARVONEN and J. PARKKINEN. A short note on the social consequences of wearables. *University of Helsinki*, 2001.

[Kun08] M. Kuniavsky. Feature user experience design for ubiquitous computing. *interactions*, 15(6):20–22, 2008.

[LB10] Emily Lovell and Leah Buechley. An e-sewing tutorial for diy learning. In *Proceedings of the 9th International Conference on Interaction Design and Children*, IDC '10, pages 230–233, New York, NY, USA, 2010. ACM.

[Man96] S. Mann. Smart clothing: The shift to wearable computing. *Communications of the ACM*, 39(8):23–24, 1996.

[Man12] Steve Mann. *Wearable Computing*. The Interaction-Design.org Foundation, Aarhus, Denmark, 2012.

[OV11] M.T. Ozsu and P. Valduriez. *Principles of distributed database systems*. Springer-Verlag New York Inc, 2011.

[PVM07] Mitchell Page and Andrew Vande Moere. Evaluating a wearable display jersey for augmenting team sports awareness. In Anthony LaMarca, Marc Langheinrich, and Khai Truong, editors, *Pervasive Computing*, volume 4480 of *Lecture Notes in Computer Science*, pages 91–108. Springer Berlin / Heidelberg, 2007.

[PW11] Hannah Perner-Wilson. A Kit-of-No-Parts. Master's thesis, Massachusetts Institute of Technology, 2011.

[Sey10] S. Seymour. *Functional Aesthetics: Visions in Fashionable Technology*. Springer, 2010.

[Sin89] M. Singhal. Deadlock detection in distributed systems. *Computer*, 22(11):37 –48, nov. 1989.

[Spa05] Sparkle, 2005.

[Tan99] A. Tanenbaum. What is a distributed system? *Notes*, 1999.

[Wai74] R.T. Wainwright. Life is universal! In *Proceedings of the 7th conference on Winter simulation-Volume 2*, pages 449–459. Winter Simulation Conference, 1974.

[Wei08] Aaron Weiss. Open source hardware: freedom you can hold? *netWorker*, 12(3):26–33, September 2008.

Index

Typeset May 18, 2012