Supporting Learning Physical Computing Through Design Activities

Melissa Kaivo
Media Technology, Malmö University
Master Thesis, 15 credits, Advanced Level
Supervisor: Daniel Spikol
Examiner: Fredrik Rutz
22nd of May 2019
ABSTRACT

Students and teachers encounter new challenges as Nordic countries, and many other countries decided to implement computational thinking and programming into the compulsory education curriculum. Likewise, universities have modified programmes to respond to the skills required in the future’s digital world. Computational thinking is nowadays a fundamental skill for problem-solving, and to successfully implement it to education, new approaches and methods need to be developed. This paper explored the use of a physical computing platform called Arduino as a means of introducing computational thinking to university students. This study aimed to investigate the challenges that students new to Arduino have when learning physical computing and explore ways to support learning activities. The prototype for this study was a visual support material that eases the challenges and shifts the focus from the process to design. The results were derived from empirical research done in Arduino workshops held in four different universities context to train participants computational thinking and programming skills. Results have implications for the benefits of design activities as a method for teaching computational thinking to university students. Findings show that design activities can provide an enjoyable, meaningful, and more feasible approach.
ACKNOWLEDGEMENT

Foremost, I would like to express my special thanks of gratitude to my supervisor Daniel Spikol, for his enthusiastic encouragement and providing invaluable guidance throughout this thesis. I would also like to thank him for all the great discussions and a good sense of humour.

Besides my supervisor, I would like to thank the following people for their assistance with the collection of my data: Benjamin Brink Allsopp and Morten Misfeldt from Aalborg University, Milica Vujovic and Davinia Hernández-Leo from Universitat Pompeu Fabra, and Jörn Christiansson from IT University of Copenhagen.

Thank you.
# Table of Contents

1 Introduction..........................................................................................................................1
   1.1 Research Questions ............................................................................................................4
   1.2 Personal Motivation ...........................................................................................................5
   1.3 Limitations .........................................................................................................................5
   1.4 Overview of the Thesis .......................................................................................................6

2 Theoretical Framework.................................................................................................................7
   2.1 Arduino .................................................................................................................................7
   2.2 Computational Thinking ......................................................................................................8
   2.3 Learning Programming .......................................................................................................9
   2.4 Physical Computing ............................................................................................................12
   2.5 Learning Through Design ..................................................................................................15
   2.6 New Methods for Learning Computational Thinking Skills ...........................................17

3 Research Methodology...........................................................................................................19
   3.1 Iterative Design Process .....................................................................................................20
   3.2 Prototype ..........................................................................................................................23
   3.3 Observation .......................................................................................................................26
   3.4 Semi-Structured Interviews ..............................................................................................27
   3.5 Questionnaires ..................................................................................................................28
   3.6 Analysis ..............................................................................................................................30
   3.7 Reliability and Validity ......................................................................................................31
   3.8 Ethical Considerations .......................................................................................................32

4 Design Phase I: Idea...............................................................................................................33
   4.1 Sketching ............................................................................................................................33
   4.2 Observation .........................................................................................................................36
   4.3 Teacher Assistant Interviews ............................................................................................37
   4.4 Findings From Phase I ........................................................................................................38

5 Design Phase II: Content........................................................................................................45
   5.1 Prototype ............................................................................................................................45
   5.2 Testing .................................................................................................................................47
   5.3 Expert Interviews ................................................................................................................49
   5.4 Findings From Phase II ......................................................................................................49

6 Design Phase III: Usage .........................................................................................................58
1 INTRODUCTION

We live in an era of a technological revolution, where interactions with technology are exponentially increasing, and the need for understanding the digital world is of utmost importance (Heintz & Mannila, 2018). The world is packed with digital artefacts, objects constructed around the essence of information technologies (Löwgren & Stolterman, 2004). These objects we use daily are creating different forms of networking and communicating with each other, changing radically the way we live today. More than 20 billion connected devices exist in the world; however, less than 1 per cent of people have the skills to understand and influence them (British Council, 2019). To be able to understand these digital artefacts and explore the world around them, people need to be introduced with computational thinking and programming. Furthermore, schools and universities are seeking ways to introduce these subjects to a diverse population of students and teachers to prepare them for the changing future (Trust, Maloy & Edwards, 2018). One of the focal challenges is to develop methods for teaching computational thinking as a means for structured problem-solving to students from various majors, and with different backgrounds. As well as to train teachers, who need to teach these skills to students without themselves having the same kind of approach in their studies.

In future, computational thinkers are problem-solvers whose skills are needed in all fields (Denning, 2017). Problem-solving using computational thinking includes algorithmic thinking, decomposition, automation, and abstraction and taking this kind of thinking into action is programming. Nevertheless, many countries who struggle with the high rate of unemployment have plenty of unfilled positions related to technology and scientific knowledge (García-Peña, Reimann & Maday, 2018). At the same time, this technological transformation is making technologies more accessible to everyone. Internet is full of open-source projects, affordable components and equipment, and shared knowledge. For example, people can nowadays afford to buy their 3D-printers to design and create the objects they need, local makerspaces offer a place for people to create digital artefacts,
and schools implement computer science to multiple subjects. These all provide the possibility for new approaches and methods for introducing the principles of computing and teaching computational thinking as a means of developing problem-solving skills. Everyone should have the same opportunity to get familiar with programming and computational thinking. For that reason, Myers (1990) argued that there is a need for programming tasks that are more accessible to all users as well as more recent studies (Milne and Rowe, 2002; Bocconi et al., 2016; Heintz & Mannila, 2018) have stated that developing new approaches to teaching computational thinking and programming for everyone is crucial.

Today, Nordic countries’ compulsory education curriculum has a new approach to introduce computational thinking and programming to children. Computational thinking and programming can be integrated into the curriculum in three different ways: 1) cross-curriculum strategy, 2) accommodation in a subject that is already being taught, 3) the establishment of a new, purposely designed subject (Bocconi, Chioccariello and Earp, 2018). To teach computational thinking to children in primary schools, already a various number of tools and platforms exist in the market. Primary schools use different tools, such as block coding with the online platform called Scratch, building and programming robots with LEGO Mindstorms, and using electronics to program physical objects with Arduino. These tools make it fun and tangible for the children to develop the skills required in the 21st century (Jenkins et al., 2009).

However, present-day teachers and students at universities were not provided with the opportunity to develop computational thinking skills when they were in primary school. “The introduction of CT and Programming in the curriculum calls for major in-service teacher training initiatives to up-scale competence. “(Bocconi, Chioccariello and Earp, 2018, p. 5). For that reason, universities and programming platform providers have started to arrange different kinds of workshops, programs, and online courses to introduce the basics of computational thinking and programming. In
addition, one of the physical computing platforms Arduino decided to put educators at the centre of their concept and created Arduino Education (Cuartielles, 2018). They noticed that the needs of educators were not addressed when introducing the new curriculum. Currently, Arduino provides workshops where participants learn the basics of electronics and programming, physical computing through the process of constructing their projects.

The learning theory of constructionism perceives a strong connection between design and learning: “It asserts that activities involving making, building, or programming - in short, designing - provide a rich context for learning.” (Kafai and Resnick, 1996, p. 4). Learning through design is a constructionist approach seeing learners as builders of their knowledge (Kafai & Resnick, 1996). Resnick and Ocko (1991) state that students explore and get a more in-depth understanding of mathematical and scientific concepts when they have the freedom to create projects that are meaningful to them. Moreover, Kafai and Resnick (1996) address that it is more feasible for learners to develop new skills when being actively engaged in making an artefact. Artefacts can be anything from making sculptures out of clay to creating complex computer games. With physical computing, these artefacts are digital prototypes that communicate with the analogue world.

Learning and combining two new things at the same time, software and hardware can be overwhelming. Software is a program with instructions to operate a computer, execute different tasks, and manipulate physical devices, hardware. Students encounter frustration, and their focus is more on learning how to use the platform for a specific purpose rather than understanding what they could do with it. Often students mainly copy the example codes and wire the components with the example wiring diagrams without understanding the purpose behind everything. Likewise, several studies discuss the difficulties that students have when trying to learn to program (Milne & Rowe, 2002; Booth & Stumpf, 2013; Sentance & Csizmadia, 2016). For that reason, when it is time for the learners
SUPPORTING LEARNING PHYSICAL COMPUTING

to use these skills to build and program their project, they have a hard time to be creative and design innovative concepts.

The purpose of this study is to explore the use of Arduino as a means of introducing electronics and programming to university level students. First, aiming to investigate the challenges that learners have in the area of programming and wiring physical prototypes. Second, aiming to design and create learning material to ease these challenges, and third to use this learning material to support students when designing and creating their projects.

1.1 RESEARCH QUESTIONS

In today’s world, computational thinking and programming are acknowledged to be as fundamental skills as numeracy and literacy are (Bocconi, Chioccariello and Earp, 2018). To teach these skills, schools and universities in Nordic countries are commonly using different physical computing platforms. Physical computing platforms allow teachers to introduce the subject to students by building tangible and fun projects. However, these platforms require the learner to work simultaneously with both software and hardware. Moreover, driving the learners’ focus more towards understanding the process of wiring and programming of particular tasks, rather than gaining skills on computational and understanding digital artefacts, these different inputs and outputs everywhere around us. Therefore, the following research questions were generated:

**RQ1:** What are the challenges that university level students encounter when learning physical computing?

**RQ2:** How can we support learning computational thinking through design activities?
SUPPORTING LEARNING PHYSICAL COMPUTING

The aims of this thesis are:

1. To investigate the challenges that students encounter with software, hardware, and when combining these two, and explore what the most necessary concepts to understand when designing projects are.
2. To design and create learning material to support these challenges.
3. To explore how this learning material can be used through design activity.

1.2 PERSONAL MOTIVATION

My educational background is in computer science, and I have created educational content for a broadcasting company for several years. The idea for this thesis started from the problem I noticed with my friends who were studying to become teachers. The new approach to introducing computational thinking and programming in primary school curriculum required them to learn how to teach programming. At the same time, they thought that universities did not offer enough support for them to develop the required competence. Courses were a few day introductions to different tools like Scratch\(^1\) and Bee-Bot\(^2\), and the focus was more on what is the tool rather than understanding how to use it to develop children’s computational thinking and programming skills, or how to explore different technologies present in our everyday lives.

1.3 LIMITATIONS

Arduino is commonly used to teach physical computing, and this study will only focus on how to support learning physical computing and computational thinking with Arduino. However, it is not the only platform for physical computing. More of why Arduino was chosen for this study is explained in Chapter 2. As I am not an expert in electronics, I had first to learn how to build and wire Arduino

---

\(^1\) Scratch, https://scratch.mit.edu/
\(^2\) Bee-Bot, https://www.bee-bot.us/
SUPPORTING LEARNING PHYSICAL COMPUTING

projects correctly. Nevertheless, with this experience, I got the insights of what concepts are challenging to understand and see the situation from the student’s perspective.

To measure the students learning outcomes and the ability to apply these skills in a more extended period was out of the scope of this study. This study focused on the experience of learning computational thinking with physical computing and the participants' ability to be creative and explorative with the help of the prototype. Exploring computational thinking as a means to learn about structured problem-solving and designing. Striving away from the overwhelming feeling of creating projects without the knowledge of how to do it. For that reason, the participants’ motivation towards learning to program was evaluated as during this study most of these workshops were mandatory for their programme and motivation can influence on how students experienced the workshop and learning situation.

1.4 OVERVIEW OF THE THESIS

Next, the following chapters of the thesis are presented briefly. In chapter two, a theoretical framework of the study is introduced. The theoretical framework explains key elements: computational thinking, learning programming, physical computing, and learning through design. In the third chapter, the research methodology is described. The design process includes three iteration phases and chapters four, five, and six describe the procedure and findings gathered from each phase. Finally, chapter seven and eight have the discussion and the conclusions of the results of this study and suggestions for future work.
SUPPORTING LEARNING PHYSICAL COMPUTING

2 THEORETICAL FRAMEWORK

In this chapter, previous research is discussed to gain a more in-depth understanding of the relevant areas and investigate the challenges of learning computational thinking and programming. First, introducing Arduino, a physical computing platform that is studied in this paper. Second, going into what is meant with computational thinking. Third, discussion on how people learn traditional programming languages and what are the challenges in that. Fourth, explaining more what physical computation is and why it is used to teach programming. Fifth, learning through design as a teaching method is discussed. Finally, conclusions are drawn, and the next steps for the study are presented.

2.1 ARDUINO

Arduino\(^3\) is an open-source physical computing platform. Arduino was developed in 2005 to teach Interaction Design, a design discipline that sees prototyping as the core process (Banzi, 2009). It is relatively easy to use without a technical background as well as building prototypes does not require significant investment. Because Arduino is an open-source platform, it has a big community who share their projects, designs, and solutions. Arduino consists of two major parts: the Arduino IDE (software), and the Arduino board (hardware).

However, Arduino is not the only physical platform; there are platforms, such as Raspberry Pi\(^4\), LEGO Mindstorms\(^5\), and micro:bit\(^6\). For this study, Arduino was chosen because of its qualities, and it is commonly used in Nordic countries to teach physical computing. For example, Arduino is a lot more affordable than LEGO Mindstorms. Compared with Raspberry Pi, Arduino is more flexible because it can be used almost with sensors of any kind, and its IDE is easier to use. Arduino IDE’s software extension ability makes it a great deal better than micro:bit. It already contains several

---

\(^3\) Arduino, https://www.arduino.cc/
\(^6\) micro:bit, https://microbit.org/
library files created by the community. Even though Arduino was the one chosen for this study, all these physical computing platforms are great tools for learning computational thinking and programming. The purpose of using these tools at schools to teach computational thinking is to provide students with a better understanding of the digitalised world. Allowing them to develop skills in problem-solving, enabling them to be more proactive and less reactive.

2.2 COMPUTATIONAL THINKING

Wing (2008, p. 2) describes computational thinking: “taking an approach to solving problems, designing systems and understanding human behaviour that draws on concepts fundamental to computing.” Computational thinking concepts and processes include logical thinking, problem-solving, decomposition, and abstraction (Brennan & Resnick, 2012). In fact, Wing (2008) argues that computational thinking will eventually influence all the fields, not only science, and for that reason, it will touch everyone directly or indirectly. That means that everyone should know what computational thinking is and how it works. Similarly, Flórez et al. (2017) define computational thinking and programming:

We define CT as a way of reasoning that compiles several high-level skills and practices that are at the heart of computing, but applicable to many areas far beyond computer science. We define programming as the process through which a person is able to provide a set of instructions that will communicate, as specifically and accurately as possible, a procedure, method, practice, or task to a machine. (p. 836)

Furthermore, Wing states that it is more important for everyone to understand the principles of computing rather than trying to become the computer programmers themselves.

Nowadays, children are associated with the term “digital natives” because of their fluency with digital technologies. Children are exposed to digital technologies from the day they are born and from
a reasonably young age, they are comfortable with chatting online, playing video games, and using multiple applications simultaneously. However, children are merely consuming the content provided by these technologies rather than creating the content themselves, such as designing new tools or applications, programming games, or building interactive toys. Resnick et al. (2009, p. 62) describe this: “It is as if they can “read” but not “write”.” According to Resnick et al. (2009), digital fluency does not merely mean that children know how to chat, browse, and interact but to have the knowledge how to design, create, and invent with new media. In addition, Iversen et al. (2016) argue that new digital tools, such as 3D printers, laser cutters, and construction kits should matter of fact expand the forms of learning in classrooms enabling children to learn through the processes of constructing and thinking rather than disabling their thinking by letting them merely carry out ready planned projects.

2.3 Learning Programming

Papert (1980) argues that more profound knowledge of programming gives the learner a more significant educational benefit in more areas than solely in computer and technology-related fields. Nevertheless, Sentance and Csizmadia (2016) observed that in schools, programming is perceived to be the most challenging aspect of computer science. Even with the right tools and approaches, programming is not an easy subject to learn. Beginners need to understand abstract concepts to recognise what the problem is, how they should approach it and create a solution. Sentance and Csizmadia (2016) conducted a survey with teachers teaching computing to find out the challenges that students have when learning to program. Their findings revealed that students have problems with understanding the abstract and mathematical concepts, such as variables, objects, and Boolean Algebra rules. As an example, variables can be explained with simple metaphors, but students need to understand the concept to be able to use different variables in multiple ways in their program. In the study of Sentance and Csizmadia (2016) many teachers answered that students have problems with connecting the theoretical concepts to the practical applications and thinking computationally,
breaking problems into smaller pieces was also difficult for students. When starting programming, students become familiar with the concept of troubleshooting. With troubleshooting, it is also essential to understand how to break down the program into small tasks that can be tested one at a time to detect what is not working as desired.

Milne and Rowe (2002) conducted research considering students studying object-oriented programming. According to them, even students who had studied programming for a while had problems to understand the nonvisual part of the code, for example, what happens in the memory of their program. A more concrete approach is required when teaching programming. In fact, students need visualisations to endorse their understanding of the complex concepts. Based on their findings, Milne and Rowe (2004) created a program visualisation tool for explaining the difficult parts in a more effortless way. The study shows that program visualisations can include both static and dynamic content. Even a static image that visualises what happens in the background of the program makes it easier to grasp for students. This information consolidates the idea to design a visual learning material for students to use when they are learning to program and understand computational thinking.

2.3.1 VISUAL PROGRAMMING

To make it more accessible for non-technical students to start programming, already a lot of visual programming tools exist, such as Scratch, mBlock, and GraspIO. Visual programming languages use a visual representation in addition to more conventional text-based representations of program source code (Booth & Stumpf, 2013). Users can create programs by using graphic elements, for example, by dragging blocks into the desired order to create a program’s behaviour. This kind of block-coding is used in Scratch, where the user creates programs by snapping different coloured and shaped graphical blocks together to form piles. These piles represent procedures and blocks are

7 mBlock, http://www.mblock.cc/
8 GraspIO, https://www.grasp.io/
different data types, functions, and so forth. The colour and shape of the block determine what it is meant for and with what other blocks it can be connected. Users have a library from where they can drag blocks to the coding area and with the help of the shapes of the blocks, it is easy for the users to pick the right ones. The shape guides users and gives hints about what is expected and possible.

However, the differences between visual programming and traditional text-based programming are considerably extensive. Visual programming languages simplify procedures and guide the user through the creation. This kind of help is not available with traditional programming languages, and for example, users have to write their own functions, and they have more data types from where they have to choose. Sentance and Csizmadia (2016, p. 482) affirmed as well that: “Bridging the gap from graphical programming to text-based programming is a challenge.” Visual programming languages have only global variables, no procedural abstraction and no place for comments (Myers, 1989). The lack of comments makes it difficult to understand the program afterwards and debug it. Booth and Stumpf (2013, p. 27) argue that the problem with block-coding languages is the bottom-up approach to program construction: “This can lead to a trial and error approach of programming with extreme decomposition and, in turn, this can make the code highly concurrent and difficult to debug.”.

With Arduino user works in an environment that is similar to the traditional programming languages. Matter of fact, in Arduino the programming language is traditional text-based C-language, but it has converted to a simple form for the users to write it. Making it easier for students to then move to other traditional programming languages. With Arduino hardware works as the visualiser of the code.
2.4 Physical computing

Physical computing means creating interactive systems using software and hardware to sense and respond to an external stimulus. Physical computing takes a hands-on approach to understand computational thinking, which can mean spending a lot of time building circuits, writing programs, and figuring out how to make different parts talk to each other and give the desired output. (Introduction to Physical Computing, 2019).

Computing is like human thinking: “Physical computing is about creating a conversation between the physical world and the virtual world of the computer” (O’Sullivan & Igoe, 2004, p. xix). Physical computing platforms give an opportunity for end users to be more than the consumers of technology. They have the option to be the creators. In fact, they are building an artefact as a means to learn computational thinking results in better learning outcomes. According to Huang, Yang, and Cheng (2013) students who used LEGO robots when learning programming had better test results in standardised programming compared with the students who learned to program using standard methods like flowcharts. When making an artefact with physical computing, it is already an entire computing system.

Physical computing brings computational thinking and programming closer to the students’ everyday lives. They can build systems that are present in an environment familiar to them and which they find compelling to understand more. For example, building a small weather station teaches how data is being collected automatically continuously around them and how they can code and decode information that is being collected with the help of different sensors (Przybylla & Romeike, 2014). While creating their projects, students need to think about what the problem is that they want to solve, what are the events required and in what order everything should happen. In addition, Pryzybylla and Romeike (2014, p. 358) state that “key aspects of computational thinking include identifying, analysing, and implementing possible solutions with the goal of achieving the most efficient and
effective combination of steps and resources.” Many times, programming and computational thinking are taught to kids by problem-based learning, where students work together to create solutions of their kind. Problem-based learning is a learner-centred approach, where students are responsible for their learning working collaboratively with the aim to solve ill-structured and open-ended problems (Savery, 2006). Likewise, Arduino believes that doing together helps people to learn easier. Their Immersive Educational Environment is project-based learning, focusing on student interaction and collaboration (Arduino - Education, 2018).

However, Vihavainen, Paksula and Luukkainen (2011) noticed that a common problem when teaching text-based programming languages is that the focus is too much on learning specific syntax or semantics rather than understanding the process. When students understand the process of coding they are able to construct more meaningful programs (Vihavainen, Paksula and Luukkainen, 2011). In universities, programming is often taught by having first traditional lectures and then working home alone with the assignments. According to Black (2006), cognitive apprenticeship learning model can offer a better way to introduce programming to students. Cognitive apprenticeship is based on the model of apprenticeship education, where a person learns skills by working in close contact with a person who is an expert in that field. Black (2006) divides the cognitive apprenticeship model into two key aspects: coaching and scaffolding. With coaching, the student can follow the example of the teacher, and learn the process step by step. The idea of scaffolding is that students are provided only with the most crucial instructions and feedback making it possible for them to continue their work. However, it is important that the expert does not reveal everything, allowing the students to find the answers on their own. Furthermore, Vihavainen, Paksula and Luukkainen (2011) created an extension to the cognitive apprenticeship called extreme apprenticeship. Extreme apprenticeship highlights the importance of continuous feedback. It implies that students can master the programming skills only by practising and doing the tasks, not by watching. It emphasises the fact that feedback to both directions between student and teacher is essential, eventually reaching the point
where the student becomes the expert. Continuous feedback motivates and encourages students to explore more.

Getting encouraged and commented by the teachers is not the only option to learn from the feedback. Vihavainen, Paksula and Luukkainen (2011) state that the critical factor in their method is learning by doing a reality. Physical objects can give excellent feedback to the students. When using physical objects with computing, it is easier for students to see if the code is working or not. They get immediate feedback from the object itself. For example, if the light in Arduino’s electric circuit is not turning on and everything is connected with the right way, something in the code is wrong. Pryzybylla and Romeike (2015) affirm that when the system fails to meet the expectations, for example, because of inappropriate sensors or delayed responsiveness, students can immediately notice this. The feedback from objects will stimulate students to create robust, expanded, complete, and correct programming, and to voluntarily spend lots of time debugging programs (Lawhead et al., 2003).

As mentioned above, many researchers discuss the benefits of using physical computing and how it is more feasible than traditional and visual programming languages. However, physical computing is a complex activity, learning both software and hardware at the same time can be overwhelming and frustrating. Physical computing requires an understanding of electricity, for example, how current flows and what is a resistor. It also requires the students to understand the abstract concepts, the same ones than with traditional programming languages. With physical computing hardware, they have a tangible object that helps to see the concepts in action. Moreover, students have to learn how to figure if the problem is in the hardware or the software. To identify these challenges, investigation and field research is needed.

Nevertheless, O’Sullivan & Igoe (2004) state that when planning a project maker should forget their understanding of computers and target their focus more to the world around them and how that could be supported by computers. How people act, what are the needs of people or the environment,
SUPPORTING LEARNING PHYSICAL COMPUTING

and what are the problems that they should solve. In addition, Przybylla & Romeike (2014) assert that physical computing encourages learners to use their imagination and creativity, focusing more on ideas, not on technical limitation. Students have the possibility to develop projects that are meaningful to themselves.

2.5 LEARNING THROUGH DESIGN

At Arduino workshops taught by David Cuartielles (Co-Founder of Arduino), one of the key elements is that students are given the assignment to design their projects. If the workshop lasts for five days, time given to their project is usually around two days. The idea is that students learn programming and electronics by designing their artefacts. Papert (1980) asserts that the best learning outcomes are achieved when learners are engaged in designing, creating, and inventing with materials. According to Buechley, Eisenberg and Elumeze (2007), programming and electronics are fundamental design disciplines, and therefore they should be taught through design activities rather than limiting education to textbook readings and “cookbook” laboratory exercises. Even if the outcome does not turn out to be as hoped, students can revise their ideas and create a new version. As Resnick (2006, p. 4) describes it “It is an iterative cycle: new ideas, new creations, new ideas, new creations.” Moreover, the project gives students something to reflect upon. Iversen et al. (2016) argue that design-based activities with tangible digital artefacts provide learners with competences that reach beyond STEM (Science, Technology, Engineering, and Mathematics) skills. They continue that students encounter ill-defined problems, and practices of trial-and-error, allowing them “to learn from failures, and to make their own choices based on their experiences in collaborative design processes in which the teacher acts as coach and facilitator.” (Iversen et al., 2016, p. 2). STEM is an educational approach where learning is placed in context, and students gain skills in solving real-world problems through designing creative solutions and using modern technologies (Watson & Watson, 2013). Likewise, Meyrick (2011) argues that STEM approach connects the school world and curriculum to the real
world, “[...] providing authentic purposes for learning and solving problems”. Furthermore, STEM is nowadays known better as STEAM where the “A” is the addition of art and design. “Inclusion of artistic thinking in the education of scientists and engineers improves their ability to create relevant products and services.” (Watson & Watson, 2013). These processes of designing digital artefacts have more similarities with design theories and constructionist approaches than traditional curriculum-based education.

Learning through design draws from constructionist theories arguing that learning is the most effective when pupils are engaged in creating a tangible artefact for a personally meaningful purpose, and throughout that, they build their knowledge and develop their abilities (Papert, 1980). Kafai and Resnick (1996) explain constructionism being both: a theory of learning, and a strategy for education. They argue that knowledge should not be something that is merely transmitted from teachers to students and activities that involve design provide a rich context for learning. “When learners are asked to design something for the use of others, their learning becomes instrumental to a larger intellectual and social goal.” (Kafai, 1996, p. 72). Kafai and Resnick (1996) affirm that both design and learning theorists agree that “construction of meaning” is a core process and that the final artefact itself is secondary. Kafai (1996) adds that learning through design is represented in the outcome but moreover in the process of doing it. Learning through design is strongly present at the Arduino workshops. When the workshop is part of a university program, the outcomes of the workshop are not evaluated; the process is the one that matters. Students have to explain how they approached their problem and designed their concept.

Resnick (1994) explains that constructionism focuses on new ways for learners to construct and on contrast, instructionism focuses on new ways for teachers to instruct. Furthermore, he states that constructionism is more likely to be the path to improvements in education. However, this requires educators to create tools that engage students in construction: “educators need to design
things that allow students to design things” (Resnick, 1994, p. 24). Kafai (1996) examines learning through design in a context where learners designed their video games. After the research project, one of the main questions raised to her mind was: “What kind of support for the process we can offer students?” (Kafai, 1996, p. 93). Additionally, this thesis aims to provide students with the most necessary information when learning physical computing with Arduino. Enabling students to design meaningful projects and construct their knowledge of problem-solving with computational thinking and programming when learning skills required to complete the project.

2.6 NEW METHODS FOR LEARNING COMPUTATIONAL THINKING SKILLS

Opportunities exist to use physical computing as a means to support developing skills in computational thinking and problem-solving. However, learning physical computing is hard, and there is a need for new tools to be developed. For that reason, schools and universities are looking for new methods for introducing computational thinking and programming. Based on those findings, the idea to create a supportive learning material, a set of cards was generated.

As Nordic countries, as well as many other countries, have decided to implement computational thinking and programming into the primary school curriculum, teachers need to be trained with these skills first. Bocconi et al. (2016) explored computational thinking as a 21st-century skill and state that the development of suitable assessment approaches and teachers’ training are vital points for successful development of computational thinking education. However, as resources and time are minimal, Tyrén et al. (2018) assert that new methods need to be developed to help teachers to learn computational thinking efficiently. Similarly, Heintz and Mannila (2018) state that there is a significant need for professional development and training initiatives for the teachers.

Learning through design argues that students construct their knowledge by designing and creating meaningful projects, and as previous research state, physical computing can be a more
natural way for learners to grasp on programming and computational thinking. Building tangible artefacts visualise abstract programming concepts. Arduino provides an excellent platform for students new to physical computing to develop computational thinking skills when designing their projects and develop programming skills when creating these projects. However, learning software and hardware at the same time raises new problems and time spend at the introduction workshops is very limited. Learning physical computing requires comprehensible support and based on the previous research, visualisations are needed, and there is a room for new kind of supportive materials. To be able to create this learning material, it is essential to investigate what are the challenges that students have when learning physical computing and what are the crucial concepts that need to be explained. The support material can offer a new way of approaching Arduino projects. The material gives learners the information that is necessary to know when starting to design their project. As this material is developed, it needs to be tested with users in the appropriate settings.
3 RESEARCH METHODOLOGY

The research methodology for this study draws from research through design and action research. When taking the research through design approach, a prototype is created to develop a product as well as gain new knowledge during the process (Zimmerman, Stolterman & Forlizzi, 2010). As one of the aims for this study was to create learning material with the most significant content, a prototype of the material had to be designed and tested with the target group. The learning material created for this study was developed to get insights into the challenges as well as a solution to ease these challenges. The prototype made it possible to interact with the students in a natural way and observe their progress and challenges during the workshop. As well as it was used as a tool to overcome or avoid those challenges.

When designing learning material, action research enables the researcher to investigate participants’ reactions to these materials, and at the same work together with the actual users to develop engaging context-specific materials. (Edwards & Burns, 2015). One of the reasons for using action research in this study was to include the stakeholders in the process. Stakeholders were included to develop a product that would be helpful and feasible for the actual users. Working with stakeholders throughout the development of the prototype gave useful insight. Stakeholders in this study were students, Arduino teachers as experts, and teacher assistants. They were all part of shaping the prototype to its final version. However, stakeholders did not make any decisions considering the prototype in this study. Stringer (2014) description of action research:

Action research uses continuing cycles of investigation designed to reveal effective solutions to issues and problems experienced in specific situations and localized settings, providing the means by which people in schools, businesses and organizations may increase the effectiveness and efficiency of their work. (p. 1)
This study focuses on Arduino workshops provided by universities to teach computational thinking to non-technical students. To take away the overwhelming aspect of learning software and hardware at the same time and make it more meaningful and rewarding for the students who will not be programmers but who should understand the principles of computing.

Constructing the research methodology from research through design, a prototype for this study was designed and developed. As action research uses a cyclical approach to investigate problems and develop solutions, an iterative design process was used to structure the data gathering and design of the prototype.

3.1 Iterative Design Process

The design methodology for this study follows a cyclic process of prototyping, testing, analysing, and refining called the iterative design process (Figure 1). Buxton and Sniderman (1980) argue that it is unlikely that the first solution designed would be the “right one” and the best possible outcome. Likewise, Zimmerman (2003) states that it is impossible to predict the experience of a user ultimately. For that reason, it is essential to have an iterative approach where each version of the solution is designed, tested and analysed.

![Design Process Diagram]

*Figure 1. Iterative design process. Adapted from Design Research: Methods and Perspectives (p. 177), E. Zimmerman, 2003, MIT Press.*

In the iterative design process, the project develops through an ongoing dialogue between the designers, the design and the testing audience basing the decisions on the experience of the prototype.
SUPPORTING LEARNING PHYSICAL COMPUTING

in progress (Zimmerman, 2003). Similarly, Buxton and Sniderman (1980, chap. 2) describe each phase of the iteration "[…] as being a prototype whose purpose is to test a critical mass of the overall problem". According to Buxton & Sniderman (1980), each phase should answer questions:

- What is to be prototyped, and how?
- What is observed, and how?
- How are results evaluated, and subsequently applied? (chap. 2)

The iterative approach to design is presented in Figure 2. This study was divided into three design phases that all include prototyping, testing and analysing: 1) to investigate the challenges and test the initial idea, 2) to find the most important and helpful content for the supportive learning material, and 3) to explore the use of this material.

Figure 2. Design process and phases. Adapted from Design Research Methods and Perspectives (p. 177), E. Zimmerman, 2003, MIT Press.

As the aim of this study was to investigate the challenges that students encounter and to design a prototype to ease those challenges, different methods were used to gain knowledge. The first design phase consisted of sketching, observation on Arduino workshop at Malmö University, and interviewing teacher assistants. The second design phase started with designing a first version of the prototype, a set of sixteen cards. These cards and especially the content in them, were tested in a one-week Arduino course with university-level students and evaluated with the help of a survey. In
addition, during the second design phase experts were interviewed. The third design phase included a second version of the prototype. Several short (70- and 45-minutes) Arduino workshops were conducted with university-level students, and surveys were answered to test and evaluate the prototype. Table 1 introduces all the methods chosen for this study, and the following sections describe the methods and their purpose in more details.

Table 1. The table includes all the methods chosen for this study.
SUPPORTING LEARNING PHYSICAL COMPUTING

3.2 Prototype

Arduino consists of software and hardware. For a person to be able to create their projects, they have to understand and use both. To learn these people can participate in Arduino workshops. These workshops can be provided by different facets, such as Arduino itself, universities as part of their programs, private companies through online courses, and so on. In this paper, the workshops provided by universities as part of the program curriculum are investigated. The prototype developed for this study was a set of Arduino cards to be used when introduced to physical computing at Arduino workshop. Cards contain a set of basic programming and electronic concepts.

Prototypes are meant to express, test and explain ideas. They are not aiming to be or work as finalised products. Simple prototypes save time allowing the researcher to focus on testing the critical elements and iterate based on the findings (The field guide to human-centered design, 2015). For that reason, the design or the appearance of the cards were not emphasised in this study. Creating beautiful cards would have taken too much time from the testing and iterating process. Creating a prototype starts with identifying the key elements of the idea and deciding what needs to be tested (The field guide to human-centered design, 2015). The Arduino cards aimed to explain the key elements that students struggled to understand with text and visualisations. A user journey was used to plan the prototype, and sketches were used as a quick prototype to test the initial idea. After that, two different versions of the cards were created.

Cards were chosen to be the form for the prototype for multiple reasons. Paper prototypes are a quick form to sketch, create, modify, and multiply (Snyder, 2003). Cards make it possible for students to collaborate while building their Arduino projects. That because cards are on the table for the whole project group to see at the same time and not on everyone’s own computer screen. Cards allow groups to work on their own pace and not to follow the pace that teachers or other students set. With the cards, students will not have to switch from programming editor to slides and shift their...
focus. Particularly when students are required to learn already two new things: Arduino IDE and Arduino board, explained in more details in the following sections. For that reason, it was intended that this prototype should not be a complex tool that they need to learn before using it.

3.2.1 ARDUINO IDE

The Arduino IDE stands for Integrated Development Environment, is the software that users run on their computer. Users have the possibility to use the online version of the IDE, the web editor in a browser (Figure 3) or download the development environment to a computer. Users write code, sketches, in a simple language modelled after the Processing language (Banzi, 2009). The programming cycle on Arduino as its simplest:

1. Plugging the Arduino board into a computer’s USB port.

2. Writing a sketch.

3. Uploading the sketch to the board.

4. Board executing the sketch. (Banzi, 2009).

![Example](https://example.com/example.png)

*Figure 3. Arduino IDE: Programming platform.*
SUPPORTING LEARNING PHYSICAL COMPUTING

3.2.2 Arduino Board

The Arduino Uno board is a small microcontroller board with a USB port, allowing the board to be connected to a computer to power and upload code to the board. The microcontroller is a tiny computer which can make other objects interactive. Microcontrollers might be hard to notice even though they can be found in many things people use every day, as an example from TVs, refrigerators, and timers. Microcontrollers use sensors to listen to the physical world and then act accordingly to what they were programmed. A button in a remote controller is a sensor, and when pushing the button program switches the channel. Arduino has many different versions of the microcontroller board, in Figure 4 is one of them, Arduino Uno board. Figure 4 is an example of how components can be wired to the board. Here Uno board is wired to a breadboard that has LED with a resistor, a button and LDR with a resistor.

Figure 4. Wiring diagram. Controlling LED with button or LDR. Arduino UNO board and components wired to a breadboard.
3.2.3 Arduino Workshops

At the workshop, participants are introduced with the Arduino board, different components to be used with the board and basic programming concepts. Usually, at the workshop, people work in pairs or in small groups, and after being introduced to a new component, students try wiring and programming with their Arduino kit. One teacher is in charge of the workshop, although as a help, she/he has teacher assistants who guide students while working with their kit. Workshops investigated for this study are for students without a technical background and meant for introduction to computational thinking and programming through physical computing.

3.3 Observation

Observation is an efficient method of gaining preliminary knowledge about a current state or situation (Walliman, 2017). During all three design phases, observations were conducted to gain more knowledge and test the prototype. By observing at the workshop, information was gathered about students’ behaviour, tasks that are given to them, key elements they struggle to understand and helpfulness of different kinds of support provided by the prototype. Observation gives an opportunity for the researcher to see if people act differently from what they say or answer surveys and at the same time, it is a more natural way for people to demonstrate their understanding in practice (Gray, 2004; Walliman, 2017). The aim of gathering data through observation rather than asking questions is to get an outside view of the phenomena (Walliman, 2017).

Observation can be divided into overt and covert observation. When people are aware that they are being observed it is called overt, and when people are unaware of this it is called covert (Gray, 2004). With overt observation people might change their behaviour because they think more how they would like to behave in front of other people instead of how they usually act. Furthermore, covert observation can reveal behaviour that people would not show if they knew they were being observed. Nevertheless, covert observation can be seen unethical and for that reason, all the participants were
informed before the workshop that there is a researcher present observing their behaviour for a thesis study.

Observation can be also divided into participant and non-participant observation. In non-participant observation the researcher observes the situation without taking part and in participant observation the researcher is actively part of the situation and working alongside the people. With participant observation, the researcher understands the situation better by becoming immersed in the research setting, a member of the group being researched, and sharing the same experiences with participants (Gray, 2004). Both participant and non-participant observation aims to understand what people are doing. However, participant observation is merely qualitative and focuses on why people are acting in certain ways whereas non-participant observation is more structured and focuses on how often people do certain things (Gray, 2004).

In the first and third design phase, students were observed without the researcher getting involved in the situation. Additionally, in the second design phase, participant observation was conducted by the researcher working as a teacher assistant at a workshop. Working together with the students for five days and not only observing them made the students open for the discussion, providing a lot of valuable information about their thoughts, attitudes, and experiences.

3.4 Semi-structured interviews

Semi-structured interviews consist of standardized and open type questions often used in qualitative research (Gray, 2004). Interviews are useful for collecting qualitative data, and they offer flexibility for the interviewee to answer freely or ask questions if having trouble to understand the question (Walliman, 2017). Likewise, semi-structured interviews give the opportunity for the researcher to ask the interviewees to clarify and expand their answers (Gray, 2004).
In the first design phase, semi-structured interviews were conducted individually with teacher assistants. The purpose of the interviews was to get the insights of concepts that students’ have difficulties of understanding or implementing into their own projects from a person who works closely with the students. Conducting semi-structured interviews with teacher assistants provided more information about the challenges that students encounter with soft- and hardware, especially when they start to build their own projects. Semi-structured interviews enabled the possibility for an open conversation and space for teacher assistants to talk about subjects they found interesting and relevant.

In the second design phase, experts in the field of teaching physical computing were interviewed individually. These semi-structured interviews focused on talking about teaching Arduino, introducing the physical computing through a design approach as well as getting their feedback of the prototype.

3.5 Surveys

Using a survey to gather data is a flexible and structured way to collect a high number of answers without having to spend time talking to everyone (Gray, 2004). In addition, the researcher cannot influence the answers, and people can answer truthfully and anonymously without feeling, for example, embarrassed (Walliman, 2017). As the workshop attended in the second design phase had 50 students using a survey was considered as a useful tool for gathering data. It is also a simple tool for all of the participants to answer and get their opinions and feedback recorded.

Descriptive surveys are meant for measuring what occurred (Gray, 2004). Pre-surveys conducted at the workshops gave insights into what was the students' experience in programming and electronics, and how students perceived the usefulness of knowing programming. Students answered one survey at the beginning of the workshop, and after the workshop students received a Creativity
Support Index survey with few questions considering feedback for the cards and their experience. All the surveys were anonymous for the students to feel no pressure about their answers.

3.5.1 Creativity Support Index (CSI)

The Creativity Support Index (CSI) is a survey designed to evaluate the effectiveness of a tool in supporting creative work: “A creativity support tool is any tool that can be used by people in the open-ended creation of new artifacts.” (Cherry & Latulipe, 2014). In this study, the CSI survey is used to evaluate the prototype. Carroll and Latulipe (2009) state that creativity as well as how well tool can support creativity are challenging to measure and often researchers use observation and interviews to evaluate the tool. However, qualitative methods alone can make it difficult to compare the results between tools and activities. The CSI helps the researchers to understand better what aspects of creativity support need improvements and based on that they can generate new design requirements (Cherry & Latulipe, 2014). The CSI can be used for various purposes, as an example to compare 1) different tools with the same task, 2) the same tool with different tasks, 3) different groups with the same task, and 4) the same group with different tasks. Additionally, it can be used to evaluate the tool without any comparison to other groups or tools. All three test sessions conducted in this study included the same kind of open-ended design task, where students generated the problem and the solution themselves. However, the prototype was modified after the first test session changing the way the tool supports students’ creativity in problem-solving and allowing them to approach the task from a different angle. With the CSI survey, these two versions of the prototype and how well they could support students when designing and creating projects were evaluated. In the third design phase, two different groups were compared as well, students from ITU and UPF had the same version of the tool and the same task.

The CSI consists of six factors: 1) collaboration, 2) enjoyment, 3) exploration, 4) expressiveness, 5) immersion, and 6) results worth effort. See the CSI survey in Appendix E.
first page, participants give a factor score on a Likert scale to each of the factors, rating how well the prototype matched with the particular factor and statement. This value can be between 1 and 20, and higher numbers indicate that cards supported that factor well. Next, on the other side of the page, participants rate which of the factors are the most important for them while doing this kind of activity. The factor count is the amount that participants chose that factor to be important for them when doing this kind of activity. This value can be between 0 and 5. The higher the number, the more participants valued this factor. Finally, these two ratings, factor score, and factor count are then calculated to create the weighted factor score and to form the final CSI score for the cards. See the equation for scoring the CSI in Appendix F.

In addition, students had the option to answer an open format question at the bottom of the survey. That gave them the freedom of expression and allowed them to answer in their own words to give valuable feedback for the cards.

3.6 ANALYSIS

The evaluation and analysis of the implementation and gathered data were carried out at the end of each design phase. After each design phase, data collected was broken down, organised and examined. Next, the data from interviews, observation and surveys were compared and clustered into themes. These themes included patterns; actions and challenges that occurred repeatedly when observing students as well as things that interviewees mentioned and were noticed during the observation. Based on these themes, insights were extracted, and new design requirements were developed for the next phases. When analysing the data repetition of themes provides valuable information for the researcher to consider those topics as essential things to include in the study. Looking for themes and patterns is called pattern coding (Walliman, 2017). From this pattern coding it was possible to formulate generalisations and relationships.
Supporting Learning Physical Computing

All the interviews conducted for this study were audio recorded, and during the conversations, raw field notes were written. Walliman (2017) states that audio recording is used to retain the uninterpreted record of the conversation; however, to be able to analyse the data, the record needs to be transcribed. Because transcription can take a lot of time, it is crucial to have well-thought questions and responders. For this study, all the five interviews were transcribed soon after the meeting keeping everything fresh in mind. Likewise, the information should be structured soon after the data is being collected and transcribed. That way, it is easier for the researcher to identify gaps, develop new ideas, challenge their own assumptions, and avoid bias (Walliman, 2017). With the semi-structured interviews, it is important to ask open-ended questions and look for themes. Similarly, at the second and third phase workshops were video recorded, photographed, notes were written during the observation, and full report was always written on the same day after each session. While field testing the prototype, oral and written feedback and comments from the students were collected for later evaluation.

3.7 Reliability and Validity

Reliability means the reproducibility of the findings (Bryman, 2016). Something that is tested today should give the same results tomorrow when tested the same way. For that reason, all the prototypes, interviews, observation, and surveys were documented using written notes as well as video and audio recording.

Validity determines if the study tests the topics it was intended to (Bryman, 2016). The validity was taken into consideration in all parts of the study. For questionnaires, it is essential to avoid things, such as irrelevant questions, confusing structure or not covering the research area, since these all can threaten its validity (Gray, 2004). To make sure questions were easy to understand and not too long, questionnaires were tested with people similar to the target group.
3.8 Ethical Considerations

For rules and guidelines for research, the Swedish Research Council (Codex, 2019) was consulted. All the participants, students, teacher assistants, teachers, and experts in this study were adults. Before taking part in this study, all the participants received written consent and information form. The participants were also informed verbally about the purpose of the study and the type of data being collected. The written consent was taken to confirm that everyone agreed to be part of the study. No personal information was collected from the participants, and all the data collected was kept strictly confidential and anonymous.

Most of the workshops observed during this study were part of the students’ university programme. For that reason, all the participants were given the same tools and opportunity to develop their skills in Arduino with the help of the prototype. Nevertheless, the prototype did not replace any of the content generally taught by the teacher in these workshops.
4 DESIGN PHASE I: IDEA

The idea to create learning material that could be used when introducing computational thinking through Arduino raised from the problem that learning Arduino is hard. Combining software and hardware is overwhelming, and time at the workshop is limited. This chapter includes sketching the learning material, observation and testing the sketches at a workshop, and interviews with teacher assistants. The aim of the first design phase was to answer to the first research questions: to investigate the challenges and find out what kind of knowledge is required for students to be able to build their own projects.

4.1 SKETCHING

Research through design approach uses design practices as methods of gaining new knowledge. With this approach, each sketch can be seen as a “quick hunch” or proposition for a possible solution to the problem (Zimmerman & Forlizzi, 2008). Moreover, the sketches of the learning material were created by following the book *Getting Started with Arduino* written by Massimo Banzi (2009), the

![Figure 5. Sketches of cards for a task “blinking LED”](image)
SUPPORTING LEARNING PHYSICAL COMPUTING

co-founder of Arduino. The workshops follow the same kind of structure when introducing different
components and concepts. The first task was “blinking LED” (Figure 5).

The blinking LED assignment is the first thing students get to experience at the workshop. To
be able to create a blinking LED, already a lot of different components and concepts are involved.
For a person to make an LED blink, they need to use Arduino board, breadboard, two wires, LED,
resistor and program all this with Arduino IDE. In Table 2, a user journey for creating this blinking
LED is described. In the user journey, actions are the steps user take during this task and paints points
are the challenges that the user might encounter. In the prototype row, the support that learning
material could give for the user is described.

Table 2. The user journey when a student is conducting the task of blinking LED.

<table>
<thead>
<tr>
<th>ARDUINO</th>
<th>Hardware</th>
<th>Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTIONS</td>
<td>Connecting an LED to a breadboard</td>
<td>Connecting wires to an LED</td>
</tr>
<tr>
<td>PAINT POINTS</td>
<td>Current flow and terminal strips</td>
<td>The polarity of the LED</td>
</tr>
<tr>
<td>PROTOTYPE</td>
<td>Visualisation of the breadboard and the current flow</td>
<td>How to recognise negative and positive sides</td>
</tr>
</tbody>
</table>

When sketching, topics were divided into categories. Categories in this phase were hardware
platform, programming, inputs, outputs, and troubleshooting. The platform category included the
yellow colour-coded cards that would be used when starting to learn Arduino: board, web editor,
breadboard, and wiring diagram. Blue programming cards explained the structure of the code, if-then
SUPPORTING LEARNING PHYSICAL COMPUTING

cancept and arrays. Orange input and green output cards included the standard components used when starting with Arduino. Red troubleshooting cards included common mistakes that beginners make. The idea of the categorising was to divide concepts and components into smaller pieces that are easier to handle. Some of the sketches are presented in Figure 6, and more sketches can be found in Appendix A. Sketches of three cards in Figure 6 could be used when wiring a speaker, piezo and programming an array to play a melody with the piezo. The front side of the yellow wiring diagram card included visualisations of how to read wiring diagrams and how to wire multiple components. On the front side of the piezo, green output card was an example where piezo should be connected in the Arduino board. The back side of this card had an example of how to program the piezo. Arrays are used with piezo to create melodies that consist of a set of notes. However, arrays are complex programming concept and required more explanation. For that reason, arrays had an own card, blue programming card. On the front side, arrays are explained with the help of storing a melody to play it in piezo. The back side of the card included examples of how to work with arrays in the code.

Figure 6. Sketches of three cards: frontside of the Wiring diagram, and the Piezo cards and both sides of the Arrays card.

Sketches of the cards were full of information. All the information that could be helpful was included in the cards because then later it could be narrowed down to the most helpful content. First different kind of content needed to be tested to be able to draw conclusions of what are the most
necessary things to share with students and what kind of tips or visualisation are the most helpful. To have clear in mind what concepts should be observed, preliminary requirements were developed (Table 3). These requirements were developed during sketching based on what concepts were found necessary to understand. Challenges were divided into two categories: hardware and software.

Table 3. Preliminary requirements for observation.

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Wiring</td>
<td>• Web editor</td>
</tr>
<tr>
<td>• Electricity/Current flow</td>
<td>• Declaring objects</td>
</tr>
<tr>
<td>• Digital pins</td>
<td>• Syntax</td>
</tr>
<tr>
<td>• Outputs and Inputs</td>
<td>• If-statements</td>
</tr>
</tbody>
</table>

4.2 Observation

An on-site observation was conducted at Malmö University, where 42 bachelor students from interaction design and product design had a four-day workshop to learn Arduino (see Table 1 in Chapter 3). The purpose of this observation was to get an overview of students working in a multiple day workshop. On the first day, students were introduced with Arduino IDE, Arduino micro board, breadboard, LED, button and examples how to program these. After being introduced with all those elements and trying to create the blinking LED, students received an assignment to create a game that is controlled with three buttons. Students worked in groups of three to first trying to design the concept and then to build and program it. For this assignment, they had four hours to work with.

When starting to create the game, one of the groups at the workshop were given some of the sketches. They received four cards named: 1) Button, 2) Breadboard, 3) Wiring diagram, and 4) If-then. The cards and their purpose were introduced to the group, and each member of the group got their own set of cards. Participants were encouraged to think out loud to get more information about
what kind of help they were looking from the cards. This felt natural to the participants because they were working in a group and talking to each other.

4.3 Teacher Assistant Interviews

After the observation on the workshop, three teacher assistants were interviewed. In the findings, these assistants are referred to Teacher Assistant A, B, and C. Three semi-structured interviews were conducted individually in the workshop environment. All three teacher assistants were current students at Malmö University and had experienced being participants in the Arduino workshop as part of their studies. Matter of fact, their first contact with Arduino was through one of these workshops. Before that, they had some prior knowledge about circuits and electronics but not about programming. For that reason, they found that it is easier for them to build than program the desired outcome. After being introduced to Arduino, they have used Arduino to create school and free-time projects. These assistants were chosen because 1) they had previous experience on working as teacher assistants in different workshops, 2) they were helping at the same workshop at Malmö University as the observing of students was conducted, and 3) they were willing to be part of this study.

The interviews started with questions considering the interviewees’ background in physical computing and working as a teacher assistant. Followed with a question about concepts they struggled to understand when starting to learn Arduino and what are the concepts students at the workshop have difficulties in understanding. Then asking about their opinion on the sketches and topics covered in them. A list of the questions can be found in Appendix B. All of the teacher assistants mentioned the challenges of the same kind with wiring the components, understanding the circuits, complex programming concepts and combining the software and hardware.
4.4 FINDINGS FROM PHASE I

During the observation, pictures and written notes were taken. Immediately after the session, a report was written. This report included challenges that students encountered, questions they asked during the workshop and requirements for the learning material. As well as evaluating what kind of visualisations and written explanations were helpful when working with the cards. When analysing the gathered data patterns and themes were searched and compared with the transcription of teacher assistant interviews. Both interviewing teacher assistants and observing students at the workshop revealed the challenges of the same kind. Findings are divided into four sections: challenges with hardware, challenges with software, working with the sketches, and requirements for the prototype.

4.4.1 CHALLENGES WITH HARDWARE

For many of the students, electronics were not a familiar concept. Most of the people are introduced with electronics in primary school, and they use products that are based on the same general concept of electricity. Nevertheless, it was difficult to understand, for example, what a closet circuit is and caused a lot of frustrations. However, this understanding is crucial to be capable of wiring multiple components and creating projects. All three teacher assistants mentioned that electricity in the breadboard and closed circuits are one of the most common things that students struggle to understand.

As an example, Teacher Assistant C answered:

You experienced it every day, for example, when putting batteries to your remote control. But people always seem to struggle with that there has to be this closed circuit for it actually to work. The most problem I see when I help people is that the jumper wires are not in the right pin or not in the ground or power and they don’t really understand that. (personal communication, January 29, 2019)
In addition, Teacher Assistant B had the same kind of experience trying to explain how a breadboard is constructed and how the current flows:

I tried to explain the concept that there are two different sides in the breadboard, and that was not easy for them to understand. But once you do, it makes everything so much more easier. (personal communication, January 29, 2019)

This same was shown in action when students started to work with the first assignment blinking LED. With many of the Arduino assignments, students start by wiring. With this assignment, students needed to wire a LED and a resistor to the breadboard. LED itself has a polarity meaning it has to be wired in a certain way to power and ground. Internet is full of wiring examples for students to see how to wire everything correctly. However, following these wiring examples without understanding why components are connected in specific ways makes it difficult to modify and program them. Especially when starting to create projects, wiring examples are not useful if people cannot apply that information to their own needs. Teachers Assistant A has found a lot of the visualisations unhelpful because they do not explain these concepts:

They usually just show you that when you plug the red one here and black one here and then you have power here and ground here. (personal communication, January 29, 2019)

4.4.2 CHALLENGES WITH SOFTWARE

When all the desired components were wired to the Arduino board, and breadboard students moved on to the next step, programming. The program for making an LED blink is not complicated; however, it requires students to understand how to use these components connected to the board in their code. They have to see the connection between physical electronics they use and the code. For example, if students followed simply some wiring example without understanding why a certain thing is connected to a particular place, it would be difficult at this point to combine the electronics and the
code. In this case, the positive side of the LED should have been connected to a digital pin in the Arduino board. This was a common thing that all the teacher assistants had noticed too. Teacher Assistant C answered:

Lots of the students struggle to understand how I change this code to make it mine to make it fit. Very small things, like when I connect to digital pin 9 I need to change it to 9 in my code and getting that connection might be difficult. Understanding where I should change and what and how. (personal communication, January 29, 2019)

Students were introduced to conditionals with the help of a button. Conditional concept if-then-else was used when pressing a button, one thing should happen, and when releasing the button, something else happens. Parallel to if-then-else condition for-loop was introduced. However, understanding how to use if-conditions took time for students, and once they managed to program something that worked in the desired way, they did not feel necessary to learn another concept. Even teacher assistants were not eager to use other than if-else in their projects. Teacher Assistant C explained this:

I normally use a lot of if-conditions. I know that it is not the prettiest or the best thing, but that is what I know. That is how I understand the code. I haven’t really learned arrays or for loops, because stuff worked when I had if statements. So, I just went with it instead of actually making it better, because it doesn’t really interest me. (personal communication, January 29, 2019)

Teacher Assistant C explained how he finally moved from using only if-else conditions to using loops when he had a project with 200 lights. Students need this same kind of motivation to learn different components and concepts. Motivation drives them to try harder to achieve the desired outcome. For example, when the observed group started to create their multiplayer game, they needed to use three buttons instead of only one. At the workshop, it was only explained how to use one button, and now that knowledge should be applied to their own project. This motivated the group with the help of the
learning material to understand jumper wires, current flow in the breadboard, and naming variables.

The overall atmosphere at the workshop starts from being excited to learn to program and to be able to build robots and shifting gradually to feeling frustrated and not understanding why nothing works even though everything looks the same as in the pictures. Teacher Assistant A commented this:

A lot of people feel like they can’t do it. Even though it is not that hard. (personal communication, January 29, 2019)

4.4.3 Working with the sketches

Testing the sketches at the workshop gave information about what kind of visualisations and tips are helpful. One of the groups was given four sketches of the cards: 1) breadboard, 2) wiring diagram, 3) button, and 4) if-then. The group with the sketches found the breadboard and wiring diagram sketches to be the most helpful for them to understand how everything should be connected for them to work. In the breadboard card visualisations, such as arrows, highlights and colours were found helpful. Teacher Assistant C commented on the breadboard sketch:

That is really good. I feel like that is something I try to explain for them a lot because I saw that some of them had put the led on this strip (points the sketch) and I had to explain to them that the reason it doesn’t work is because at the moment the electricity doesn’t go into the led. (personal communication, January 29, 2019)

With the help of the button, the wiring diagram, and the breadboard sketches, the group managed to wire multiple buttons successfully. For the button card, as well as for the LED, it is essential to explain if a component has polarity or if there are other things that matter when wiring the legs. For example, with a button, it can be hard to know how everything is connected by merely looking at the button because its legs are partly inside of the component when programming buttons group used the if-then sketch to understand the concept and found the explanation to be useful. Even though the group
understood the logic and how they should use it phrasing the statement right was difficult, and the sketch did not help with that.

Because the time is limited at the workshops, everything cannot be explained thoroughly. When programming, small things are important; however, in the beginning, it can be hard to remember them. For that kind of problems, this learning material can provide valuable information and used as “cheat sheets” for the students. Teacher Assistant A commented on the sketch of a structure card:

That is really good like say the normal bracket, and that is a function, and the curly brackets are for something else. That is very fundamental. Like if you have the equal, why would you write “equal equal” (==) and not just one “equal” (=). When I learned those things or got introduced to them, that is when I really started to make progress.  
(personal communication, January 29, 2019)

While observing students building their projects question was raised in mind: how to teach computational through design? Because at the moment a lot of the focus was on learning the Arduino platform. Students were given the assignment to choose a real computer game and make it work as a multiplayer game with cardboard controllers as they were buttons. If using computational thinking skills, the process should start with designing the concept. Breaking the game into small pieces and understanding how the game logic and controllers work. After that, the next step would be to think what physical components are needed, how to wire and program them, and after everything works put effort to what the final product will look like. However, most of the students started with crafting and building the final products and putting a lot of effort into that.
4.4.4 Requirements for the learning material

At the workshop, concepts are introduced parallel to each other. This because it is easier to create meaningful exercises when multiple concepts are combined. To be able to create a blinking LED, students are required to understand how Arduino board and a breadboard work, what is the web editor, and how to wire and program LED. This means already four to five cards to be used at the same time. As an example, the workshop uses different physical examples to explain programming concepts: controlling an LED with a button to explain if-then-else, playing a melody with a piezo to explain arrays, detecting light with LDR to explain storing values, and controlling servos with a potentiometer to map values. This approach should be considered in the learning material to motivate better where and how to use different inputs and outputs, as well as to avoid having too many cards. A list of challenges that could be eased with the learning material can be found in Table 4. In Appendix C, cards to be developed for the second phase are listed.

Table 4. Challenges to ease with the prototype.

<table>
<thead>
<tr>
<th>Challenges</th>
<th>Support required</th>
<th>Learning material</th>
</tr>
</thead>
</table>
| How to understand which way to connect the component | Visual explanation and tips to explain what polarity is, and what leg should be connected to where. | -LED’s positive and negative side  
-Button’s legs  
-Servo motor’s wires colours |
| Where the current flows in the breadboard        | Basic understanding of electricity is required, visualisation to explain the connections.  | -Highlighting the current in a breadboard |
| What is the difference between analogue and digital pin or output and input | How the different parts in Arduino board work.  | -Uno board with callouts |
| Where to use different components               | Examples of where users can find different inputs and outputs in their everyday lives. | -Potentiometer as a volume knob  
-LED in traffic lights |
### SUPPORTING LEARNING PHYSICAL COMPUTING

<table>
<thead>
<tr>
<th>Structure of the code, syntax</th>
<th>Explain what the different elements in the code are.</th>
<th>-Brackets, comments, operators, void setup(), and void loop()</th>
</tr>
</thead>
<tbody>
<tr>
<td>How to read values in the code</td>
<td>Mapping, if statements, reading the value and using it to control outputs.</td>
<td>-Example code and visualising map() with a potentiometer</td>
</tr>
<tr>
<td>How to combine wiring and code</td>
<td>Using the same pins in the board and the program. Introducing Serial monitor for troubleshooting.</td>
<td>-Explaining new concepts with the help of practical examples, such as if with buttons, arrays with piezo, loop with servo motor</td>
</tr>
</tbody>
</table>
5 Design Phase II: Content

This chapter includes the first version of the prototype, observation and testing at a one-week workshop, and expert interviews. The purpose of the prototype was to support students, especially when starting to design and create their own projects. The aim of the second design phase was to test the content in the learning material and explore what kind of visual or textual support is the most helpful to ease the challenges.

5.1 Prototype

The second design phase started with designing the first version of the prototype. One of the aims was to take the overwhelming aspect away and include small tips that explain important things but do not provide too much or unnecessary information. This content was created based on the findings from the first phase. The first version of the prototype included 16 cards, and the full set can be found in Appendix D. For example, all the teacher assistants agreed that the current flow is the most challenging thing for students to understand as well as most crucial when wiring multiple components. That is why it was decided that a breadboard card should be created with a visualisation intending to highlight the current flow. Figure 7 is a front side and a back side of a card for a breadboard. On the front side of the card visual help and callouts provide useful information for the user and on the back side, these elements are explained in more depth.
Figure 7. Card for a breadboard. Explains with visuals and text how breadboard is constructed.

Figure 8 presents the card for a button. The front side of the card indicates what components (button, two wires, ground and digital pin) are required, how electricity works with a button, and how to wire it with Arduino board. The back side of the card includes programming example and hints that the user should take an IF card if more information is needed.

Figure 8. Card for a button. Wiring and programming examples.
5.2 Testing

On-site participant observation was conducted at the Aalborg University of Copenhagen with 50 master students participating in a one-week workshop to learn Arduino. The workshop was a mandatory part of their course to learn computational thinking and programming. Participant observation included working as a teacher assistant at the workshop. This gave the opportunity to work and talk in close contact with the students in a natural situation. During these five days, students working with the cards and assignments were documented by video recording, photographing, observing, asking questions, and writing notes. The purpose of this observation was to see students working with the cards during a more extended period, for the whole week and gather data about helpful content in the cards and how to work with them. As well as experience the challenges students

Figure 9. Students creating projects at the workshop.
encountered from a teacher assistant perspective. All the participants answered pre-survey and gave feedback for the cards through answering the CSI survey. Students worked in groups of three, and each group was given one Arduino kit and one set of cards (Figure 9).

The first day of the workshop focused on introducing Arduino in general and getting to know the platform. Students tried the assignment “blinking LED” where they needed to wire an LED and a button to the breadboard and program them to work. On the second day, more components were introduced, such as servo motors, potentiometer, and light sensor. Students worked on their own pace, trying to wire and program different inputs and outputs. The third day last components were introduced and after that the final assignment, to create an environmental-themed Arduino project was given to the students. Last two days of the workshop were reserved for the project designing and building. At the end of the workshop, each group presented its concept for the rest of the class. Presentations were outstanding for other groups to see what solutions were generated from the same components and how other groups approach the same problem from a different angle. As well as these presentations gave valuable information on the process and concepts students understood or struggled to implement. Presentations were video recorded, and projects’ source code were collected to evaluate the outcomes, how innovative, complicated, and creative they were.

5.2.1 Surveys

Pre-survey was used to gather information about the background of the students, and how they perceived the easiness and usefulness of knowing programming and electronics. They rated on a Likert scale from “Strongly disagree” to “Strongly agree” their current skills on programming and electronics. In addition, post-survey had questions considering how students experienced working and learning with Arduino.

After finishing the projects, students received a Creativity Support Index (CSI) survey to fill out. With CSI, students evaluated how it was to work with the cards. At the bottom of the survey,
SUPPORTING LEARNING PHYSICAL COMPUTING

students had the option to give feedback for the cards by answering an open format question. The CSI survey was used to evaluate the cards and to have quantitative data to compare with the observation. The CSI survey’s questions can be found in Appendix E.

5.3 EXPERT INTERVIEWS

Two experts in the field of teaching physical computing were interviewed individually. In the findings, these experts are referred to as Expert A and B. Both have long experience in teaching physical computing for university-level students. The purpose of these semi-structured interviews was to understand more how physical computing is taught at universities and what kind of approach they used to introduce Arduino. Interviews started with questions about the background of the expert, and then going into more detailed what kind of courses they were teaching and to whom. Furthermore, questions about the goals and approaches at their courses were asked. At the end of the interview, experts gave their feedback for the cards. A list of all the questions can be found in Appendix G.

5.4 FINDINGS FROM PHASE II

When analysing the data gathered during the second design phase, findings were divided into themes: challenges with hardware, challenges with software, working with the prototype, and requirements for the prototype. Notes from observation and answers from surveys provided information about the challenges and were used to evaluate the cards. In addition, expert interviews explained some of the findings from observation and gave useful information about how to shift focus towards developing required skills through design activities.
5.4.1 Challenges with Hardware

Before the workshop, many of the students were not familiar with using physical computing platforms. Seven of the students had worked with physical computing platforms, micro:bit or LEGO Mindstorms and only one of the students had used Arduino before. Presented in Figure 10, when answering the question “I feel that I know about physical computing” 59 per cent of the students chose options “Strongly disagree” or “Disagree”. As well as 68 per cent answered, “Strongly disagree” or “Disagree” to the statement “I know what a resistor or a breadboard does”.

![Figure 10. Students experience in electronics (n = 50).](image)

During the observation, challenges of the same kind were encountered as previously mentioned by teacher assistant and revealed by the observing in the first design phase. The electric current and closed circuits were not familiar concepts, and students got caught up in things that were not actually important. For example, when the example wiring diagram used a yellow wire to connect the LED to a digital pin, students tried to find this yellow wire. Even though the colour of the wire used to make the connection does not matter. However, when wiring a servo motor, it is crucial to
understand that the colours of the wires coming from the motor had a purpose and had to be connected the right way.

In addition, it was difficult for the students to know if all the components were wired correctly. Because to test these connections, the code needs to be uploaded to the board, which means that students were required to work with the hardware and the software simultaneously to see that both parts worked appropriately. Matter of fact, Expert B has noticed that one of the critical issues is that there are too many new things involved in this process:

They struggled with this overwhelming feeling, learning both electronics and programming at the same time. (personal communication, February 18, 2019)

Furthermore, Expert A discussed the aspect that it is vital to limit the information given to the students simultaneously:

You become a director of an orchestra when you try to teach these things. You have to maintain focus, and that goes for all of them. (personal communication, March 6, 2019)

When introduced to new inputs and outputs, students desired to know more about what this particular component is suitable for. It would have helped them more to have the knowledge of where it could be found in the real world and why to use it. The information in the cards was too much at once. The front side of the card already explained three different things: what the component is, what the components are required to wire it correctly, and a wiring diagram to give an example of how the components should be connected. All the necessary information could be found in the cards; however, it was difficult for the students to see what information was important in each phase only with a glance. For example, the LED card had an explanation of the polarity, yet students struggled to wire the LED’s legs the right way.
5.4.2 Challenges with software

Based on the pre-survey students participating in the workshop were not highly experienced with programming either. Twelve students had programmed with a visual programming tool called Scratch, and five of these students had some experience in HTML and JavaScript. Figure 11 presents that 77 per cent of the students answered, “Strongly disagree” or “Disagree” to the question “I can explain Boolean logic concepts, for example, if-else statements”. As well as 72 per cent answered, “Strongly disagree” or “Disagree” to the statement “I can explain what different types of variables are for”.

Both concepts, if-else and variables were included in the cards because the first observation and the teacher assistant interviews brought up this problem of understanding abstract concepts. Expert B had the same kind of experience:

I taught programming in a university, so I know that same problem too just describing what a variable is to someone who has never programmed is a quite abstract thing. It is not easy at all. (personal communication, February 18, 2019)
SUPPORTING LEARNING PHYSICAL COMPUTING

Students had cards that explained programming concepts, such as the structure of the code, variables, if-else, arrays and loops. However, when starting to write the program for the project, students encountered problems for knowing what card they should use to search for help. When working with a servo motor, students could have created loops, and for that, they had a card that explained the loops. Nevertheless, they did not have the understanding that the program can make something loop; for example, a servo to rotate as long as a particular condition is true. This could be emphasised when introducing the component. For example, servo motors can be used to spin the tires of a car and this spinning can be done by adding a loop to the code.

5.4.3 WORKING WITH THE CARDS

Each group worked with one set of cards to collaboratively create an Arduino project. Some of the groups were using the cards more throughout the project and some of them less only in situations when they were told to look for something particular in the card. Students explained this in the open-ended questions in the Creativity Support Index (CSI) survey.

The average CSI score for the cards generated by the students in the workshops was 67.70 (SD = 12.32). In Table 5, the average factor scores, the average factor counts, and the average weighted factor score for each factor are presented.

Table 5. CSI results for the prototype. The average CSI score was 67.70 (SD = 12.32, n = 50).

<table>
<thead>
<tr>
<th>Factors</th>
<th>Avg. Factor Score (SD)</th>
<th>Avg. Factor Counts (SD)</th>
<th>Avg. Weighted Factor Score (SD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collaboration</td>
<td>15.35 (3.68)</td>
<td>2.67 (1.43)</td>
<td>41.47 (24.83)</td>
</tr>
<tr>
<td>Results worth effort</td>
<td>13.39 (3.32)</td>
<td>2.27 (1.73)</td>
<td>41.71 (25.29)</td>
</tr>
<tr>
<td>Exploration</td>
<td>13.22 (3.76)</td>
<td>2.27 (1.35)</td>
<td>28.98 (18.46)</td>
</tr>
<tr>
<td>Immersion</td>
<td>10.53 (3.80)</td>
<td>2.16 (1.49)</td>
<td>36.16 (23.02)</td>
</tr>
<tr>
<td>Expressiveness</td>
<td>12.98 (2.89)</td>
<td>2.71 (1.71)</td>
<td>22.04 (21.56)</td>
</tr>
<tr>
<td>Enjoyment</td>
<td>14.53 (3.53)</td>
<td>2.88 (1.51)</td>
<td>28.33 (23.69)</td>
</tr>
</tbody>
</table>
Based on the factor scores cards were supporting collaboration, students giving them an average score of 15.35. It was easy for students to work together while using the cards. Furthermore, based on the average factor counts, students rated enjoyment, collaboration, and creativity to be the most important for them when learning new. Students did not get immersed with the cards; the average factor score was only 10.53. However, immersion was as well the least important for the participants. Even though factor score for enjoyment was the highest and participants enjoyed working with the cards, many times they forgot to use them. Students could have been encouraged more to use the cards first and then ask from a teacher assistant if they could not find help. One way to bring the cards more present at the workshop would be to include them as part of the teaching saying for example: "now use the LED card" when they started to wire the led for the first time. Feedback from a student who crossed mostly to the middle section of the factor scores:

I feel like I would have enjoyed using them if I had remembered to use them. We could have been introduced more to the cards, so we knew that we could turn to them for more help. And not always asked from the teacher first. (personal communication, February 15, 2019)

Students would have hoped more examples of how and when they should use the cards at the beginning of the course. Cards should have a more significant meaning than being a learning material, a specific way to use them. Feedback from a student who gave good overall grading for the cards:

I needed a purpose to use the cards. It was not until the own project that I realised why I had the cards. (personal communication, February 15, 2019)

Because one set of cards included 16 cards, students had problems to understand what information they should search in which card. As well as the workshop had slides, meaning that some of the things that could be found in the cards were also in the slides. For that reason, some students were using the slides on their computer to get information. Using slides was a more traditional way to
receive learning materials from the teacher and felt natural for the students. However, Expert A argues that these cards could be a better solution than sharing the slides:

This (cards) is a good approach, give them something physical. Because usually, they only have one screen. Some of them like to have the slides but switching between them, and web editor shifts focus. (personal communication, March 6, 2019)

5.4.4 Computational thinking for everyone

When starting to create projects in groups of three students found it hard to know where to begin with. Many of the groups limited their thinking to inputs and outputs that they managed to wire and program successfully during the first days. Expert A had noticed this in his courses as well.

For them, it is a weird situation. They are here (in university) to gain knowledge and not to do stuff without any knowledge. (personal communication, March 6, 2019)

Based on the pre-survey (Figure 12), most of the students perceived that knowing how to program will become an essential skill in their future work, and they were eager to learn these skills. However, when struggling with the overwhelming feeling of not understanding anything, students questioned why they need to know these things even though they are not aiming to be programmers.

![Figure 12. Perceived usefulness of knowing programming.](image-url)
SUPPORTING LEARNING PHYSICAL COMPUTING

Moreover, the focus was on the process of wiring and programming specific Arduino components. Even though students are not going to need these specific Arduino skills later in their education or work. For them, the main goal should be to learn computational thinking and understand how to apply computing principles in their future work whatever it will be. Furthermore, Expert A explained his approach to teaching Arduino:

It is not what you can do with it as a professional because nobody uses it (Arduino) as professional as less you are an engineer. So, the core of the first course for me was developing a critical mindset towards your own design process and then coupled with the understanding of technology. Then it becomes interesting to teach and interesting for them to learn. (personal communication, March 6, 2019)

Students struggled to design innovative projects. Especially tricky was to know where to start with, how to decide what they want to build and how to break the idea into smaller tasks, that are easier to handle and conduct. Many of the groups created projects with similar functionalities, and the source codes were mostly copied from the example codes. However, reading existing code can be a great way to understand programming and based on the answers from post-survey students perceived that learning with Arduino was helping them to understand programming (Figure 13). Based on the observation, students were enjoying and having fun while creating the projects. Physical components made it easier for them to understand the abstract programming concepts. As an example, when changing the condition in if-statement students could immediately observe how the behaviour of the physical output component changed.
After seeing the challenges students had and talking with experts, it was clear that the focus should be shifted from the process of wiring and programming to solving problems with the help of design thinking. Through designing meaningful projects, participants would become motivated to learn more of these skills required to create the projects. Matter of fact, Expert A mentioned that:

"Usually the ones with the least technological understanding or pre-knowledge of programming were the ones who made the best projects. (personal communication, March 6, 2019)"

The theoretical framework already stated the benefits of learning through design; however, the first version of the prototype did not manage to support students in being creative when designing their own projects. The prototype was helpful when wiring and programming; however, more support was needed when designing the concept. The problem was that these students did not have a design background or design thinking skills. To help the students to be creative and design meaningful projects, the prototype should be modified to shift the focus from the process to design.
6 DESIGN PHASE III: USAGE

This chapter includes the second version of the prototype, observation, and testing. The aim of the third and last design phase was to test the revised learning material and its usage through design activity. This time the emphasis was more on developing skills in problem-solving through computational thinking and being innovative with different technologies.

6.1 PROTOTYPE

Previous phases have revealed that important aspect when developing computational thinking skills is first to understand what different outputs and inputs exist and how to use them. The first and second design phases aimed to explore what are the crucial concepts that students need to understand to be able to be creative and innovative when designing their projects. However, to understand how inputs and outputs work, it is not necessary at all to know how to wire or program them. If students want to be creative when designing projects, they need a possibility to conceptualise and an opportunity to explore various components. The prototype was modified to create a solution that provides support in conceptualising and exploring, see the sketches in Appendix H. The first modification was to create cards that would only have the picture of the component, short explanation of what it can be used and examples from the real world where to use them.

The second modification was to develop a way for the students to design their flowcharts before starting the wiring and programming. The idea was to create a visual program or pseudocode with the cards that could be then transferred to an actual working project with Arduino components. For this, if (?) and then (!) conceptualising cards were created (Figure 14). If the input sensor is triggered, then the output component should react. Cards would be used to generate a detailed description of what the program should do. However, these cards should be filled with natural language rather than focusing on how to phrase them the right way for the program. The only
important thing for the students to know when filling out these cards is that the question should be answered true or false, the same way as with an if statement in the program. Colour coding was designed for the cards to help the students to understand the structure of the cards when designing their project. Input cards have a yellow side that should relate to the yellow side of the “?” card and output cards have a green side that should be connected with the “!” card.

Third modification for the cards was to make them foldable, with four pages to have information. This way, the information provided for students could be regulated into smaller pieces. The first page of the input or output card included only the information about what it is and where it could be used. With this information and help of the new conceptualising cards, students could design their projects without focusing on how to wire or program the components. Then when the idea for the projects has been created with the help of the cards, students can move on to wiring and programming. For that, they can open the folded card and find explanations and wiring and programming examples (Figure 15). Previously IDE and board had their own cards now these cards were divided into two foldable cards. To make it easier for the students to understand what kind of

**Figure 14.** Second version of the cards. Temperature sensor, if-then, and servo motor cards.
information they should search in each card. As well as to reduce the number of cards on the set. The full set of cards can be found in Appendix 1.

![Example how to wire it](image)

**Figure 15.** The foldable LED card from the inside.

### 6.2 Testing at UPF

To test this idea of supporting students when designing their own physical computing projects six (6) sets of 70-minute workshops were conducted at Universitat Pompeu Fabra. Together 36 participants were divided into six workshops with each workshop having six students working in two groups of three. To develop students’ computational thinking skills, they were given a design activity to accomplish in 70 minutes. This task was to create an interactive toy or a friend for a specific target group to help them with some kind of a problem (Figure 16). Students had the freedom to decide what problem they would like to solve with this toy. In this workshop, each group of three had one set of cards and one Arduino Kit to use. During the 70-minutes, students worked in their groups without the help of a teacher. Before starting the workshop, students answered a survey considering their prior knowledge about physical computing.
At the workshop, Jigsaw teaching technique (Slavin, 1989) was used as a method of learning Arduino. With Jigsaw the learning session is divided into three sections, in this test session: 1) 10 minutes in the first group, 2) 15 minutes in expert groups with students from another group, and 3) 45 minutes working again with the first group. The first 10 minutes students worked in their groups of three to read the assignment and deciding which group member would be in charge of the hardware, software, and design. For the next 15 minutes, students were divided into the expert groups. Each expert group had two people, and their task was to focus on only one of the aspects of creating a project with Arduino: hardware, software, or design. In these groups, students started to work with the second version of the prototype. After that, students went back to the first groups, and the remaining 45 minutes were reserved for the group to design and build their interactive toys.

Figure 16. One of the groups building a mobile for kids that lights up and starts to spin when the room is dark.
For the interactive toy, students were given three input cards: button, light sensor, and temperature sensor and three output cards: light, speaker, and motor. Each of the inputs and outputs had their own card to help the students. Additionally, students had two cards introducing the platform elements, such as Arduino Uno board, breadboard, and Arduino IDE. When designing projects, students were asked to think of the target group and the purpose of this toy. After the workshop, students presented their concept and answered the CSI survey to evaluate the cards. When analysing the observation and CSI results, the concepts created by students were evaluated. If students managed to create innovative projects using multiple inputs and outputs. As well as how far the groups managed to assemble these projects with the help of the cards.

6.3 Testing at ITU

Similar kind of test session was conducted at the IT University of Copenhagen with 16 participants. Students were divided into five groups, and they had 45 minutes to design and build interactive toys. Jigsaw technique was once again used to give the possibility for students to first focus only on one of the elements of Arduino. This time the first part of the test session was merely used to introduce the assignment and divide students into expert groups. After working for 15 minutes in the expert groups, students went back to the first groups, and for the remaining 30 minutes, they had time to design a meaningful toy. Because the time was limited, the focus was more on creating an innovative concept and prototyping only some parts of the functionality. At this point, students were not supposed to think about how the actual product would look like. The participants in this workshop had previous experience working with Arduino, and this time, the focus was observing if this new kind of tool was helpful for them when creating their concepts. Before the workshop, students answered pre-survey, including questions about their experience in physical computing, and after the test session, students answered the CSI survey and gave written feedback for the cards by answering open format questions.
6.4 FINDINGS FROM PHASE III

During the testing, students worked in their groups without the help of any teacher. While working with Arduino and the cards, students were observed, and field notes were written. After testing the prototype notes from observation, outcomes from workshops, and survey answers were analysed, and themes were searched. Both workshops revealed findings of the same kind, how these design activities could be used as a practical approach to introduce physical computing. Findings are divided into two sections: designing concepts and evaluating the cards.

6.4.1 DESIGNING CONCEPTS

The problem that was noticed already in the first design phase was now starting to solve. With the conceptualising cards, students were able to design meaningful and well-thought projects. The shift from putting too much focus on the process and giving space for developing computational thinking skills was succeeded with the conceptualising cards. Students started to create their concepts from thinking about a problem they would like to solve with their project.

In Figure 17, one of the groups at the UPF designed a concept of a wristband that detects the body temperature of a kid.

*Figure 17. Concept of thermometer wristband for kids designed by a group at UPF.*
SUPPORTING LEARNING PHYSICAL COMPUTING

The wristband would measure the temperature, and when it is more than 37 degrees, it will play an alarm sound and light up to tell parents that their kid might be ill. This group managed to wire all the necessary inputs and outputs correctly and program them. Other concepts designed at the UPF workshops were memory games for older adults, automated food distributor for the pets at home, over bed mobile that will light up when it is dark so that child does not have to be afraid of the dark, and many more. When creating these concepts, students were eager to learn the required skills and used the cards for wiring and programming examples. Matter of fact, when the time was up, a lot of the groups would have wanted to continue with the project to finish it.

In Figure 18, one of the groups at ITU designed a toy for a dog when it is alone at home. This toy would sense if the ball is on the sensor and then when true it would throw the ball and play a fetch recording. When the dog would return the ball, this toy would complement the dog.

Figure 18. Concept of toy for a dog who is home alone designed by a group at ITU.
SUPPORTING LEARNING PHYSICAL COMPUTING

Student groups in the third design phase had considerably less time to design and create a project than what the groups had in the second design phase. Nevertheless, in both workshops with the help of the conceptualising cards, students managed to create meaningful and complex projects. When approaching problems from the real world and trying to design solutions, students create meaningful projects, and that can motivate them to learn physical computing skills required to finish the project. Lot less frustration was present at the workshop compared to the previous observations. As well as students were not limiting their thinking to inputs and outputs that were the easiest to wire and program.

6.4.2 Evaluating the cards

After the second design phase, the Likert scale of the pre-survey was modified. In the third design phase, on the pre-survey students rated their programming skills and knowledge about Arduino on a scale from 1 to 10. Based on the survey, most of the students in UPF had some experience in programming with Python. However, 75 per cent of the students rated their knowledge about Arduino to be on the level of 1 or 2. In contrast, students in ITU had some experience with Arduino, and 37 per cent rated their knowledge about Arduino to be higher than 5. Nevertheless, 44 per cent of students in ITU rated their programming skills to be lower than 4. Even though students had some experience in programming and working with Arduino, the set of cards were providing a new approach to learning physical computing and problem-solving skills.

As students worked on their own without the help of a teacher or teacher assistants, they were more depended on the content of the cards. For the cards, an average CSI score of 82.4 (SD 14.19) was generated by the participants in the UPF workshop. In Table 6, the average factor counts, the average factor scores and the average weighted factor scores from the UPF workshop are presented. Students at UPF chose enjoyment remarkable more times than the other factors. Average factor count being 4.19 for the enjoyment. In addition, cards received a good average factor score for the
enjoyment 17.58, meaning that students enjoyed creating their projects with the help of the cards.

Students at UPF generated an average factor score of 17.61 for collaboration, indicating that it was easy to work with other students while using these cards.

**Table 6.** CSI results for the second version of the prototype generated by the students in UPF. The average CSI score was 82.4 (SD = 14.19, n = 36).

<table>
<thead>
<tr>
<th>Factors</th>
<th>Avg. Factor Counts (SD)</th>
<th>Avg. Factor Score (SD)</th>
<th>Avg. Weighted Factor Score (SD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collaboration</td>
<td>2.44 (1.30)</td>
<td>17.61 (1.86)</td>
<td>42.5 (22.24)</td>
</tr>
<tr>
<td>Results worth effort</td>
<td>2.64 (1.44)</td>
<td>16.67 (4.01)</td>
<td>42.5 (24.14)</td>
</tr>
<tr>
<td>Exploration</td>
<td>2.08 (1.36)</td>
<td>16.11 (3.11)</td>
<td>33.61 (24.58)</td>
</tr>
<tr>
<td>Immersion</td>
<td>1.22 (1.73)</td>
<td>15.08 (4.37)</td>
<td>17.72 (25.04)</td>
</tr>
<tr>
<td>Expressiveness</td>
<td>2.42 (1.20)</td>
<td>15.17 (3.72)</td>
<td>37.03 (21.71)</td>
</tr>
<tr>
<td>Enjoyment</td>
<td>4.19 (0.98)</td>
<td>17.58 (3.33)</td>
<td>73.75 (23.36)</td>
</tr>
</tbody>
</table>

Furthermore, participants in the ITU test session generated an average CSI score of 63.5 (SD 14.92) for the cards. Overall average CSI score from the ITU test session was noticeably lower than the average score from UPF. However, this might be affected by the fact that students at ITU had less time to work with the cards. In the open format questions, students wrote that they would have wanted to have more time to get to know the cards accurately. One of the students explained:

The cards helped, but there was not enough time to actually explore the cards.

(personal communication, April 29, 2019)

Even though cards can be commonly known concept for users, this set of cards has still a unique purpose and way of using it. The fact that students needed first to learn how to use these cards means that there should be time reserved for getting to know how to use the cards. In Table 7, the average factor counts, the average factor scores and the average weighted factor scores from ITU test session are presented.
Table 7. CSI results for the second version of the prototype generated by the students in UPF. The average CSI score was 63.5 (SD = 14.92, n = 16).

<table>
<thead>
<tr>
<th>Factors</th>
<th>Avg. Factor Counts (SD)</th>
<th>Avg. Factor Score (SD)</th>
<th>Avg. Weighted Factor Score (SD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collaboration</td>
<td>2.06 (1.61)</td>
<td>14.06 (3.42)</td>
<td>27.13 (21.59)</td>
</tr>
<tr>
<td>Results worth effort</td>
<td>2.69 (1.49)</td>
<td>12.00 (3.65)</td>
<td>31.94 (19.99)</td>
</tr>
<tr>
<td>Exploration</td>
<td>2.75 (1.61)</td>
<td>13.13 (3.67)</td>
<td>34.44 (22.33)</td>
</tr>
<tr>
<td>Immersion</td>
<td>0.94 (1.18)</td>
<td>8.31 (4.80)</td>
<td>7.5 (9.98)</td>
</tr>
<tr>
<td>Expressiveness</td>
<td>3.25 (1.34)</td>
<td>13.25 (3.57)</td>
<td>43.06 (22.51)</td>
</tr>
<tr>
<td>Enjoyment</td>
<td>3.31 (1.25)</td>
<td>13.56 (3.88)</td>
<td>46.38 (24.62)</td>
</tr>
</tbody>
</table>

Based on all the three test sessions and CSI surveys (AAU, UPF, and ITU) students valued enjoyment the most and immersion least. The factor expressiveness, with the average score of 3.25 was more critical for the students in ITU than in other universities. Furthermore, almost all of the students find the conceptualising cards helpful and proper way to visualise the code. The conceptualising cards made it easier and faster to find a variety of solutions to a problem. When asked how it was to first design the project with the conceptualising cards (“?” and “!”) one of the students from ITU wrote that these cards would be a useful tool especially for beginners when starting to learn to program:

It was a fun approach which would definitely help me program if I didn’t have any knowledge. (personal communication, April 29, 2019)

Furthermore, another student answered:

It helped visualize the idea and understand the different sensors. (personal communication, April 29, 2019)

Multiple answers stated that input and output cards were helpful when understanding and discussing different sensors. As well as getting useful information about how to work with the component.
SUPPORTING LEARNING PHYSICAL COMPUTING

However, few of the students at the ITU noticed a problem when collaboratively working with the cards. One of the students wrote:

They were helpful, but sometimes multiple people needed the same card for different tasks. (personal communication, April 29, 2019)

If one group member is trying to wire the component with the help of the card, and another member would like to write the program for that particular component, it can be hard to share the card. The collaboration received the highest average score from all of the six factors in ITU. That can be explained because the cards were helpful when designing the concept and discussing the components together with other group members.
7 DISCUSSION

In the first chapter, the aims of this study were divided into three parts. First, to investigate the challenges that university level students encounter when learning computational thinking using physical computing platform Arduino. Second, to design learning material to ease these challenges. Third, use the prototype through design activities to support developing computational thinking skills. Additionally, to make it accessible for everyone to learn computational thinking and programming skills. By approaching these research questions using research through design, a prototype was created to understand the challenges more and to develop a solution to ease these challenges. By applying an iterative design process, this prototype was developed through three design phases. In this chapter, research questions are discussed using findings from three design phases in relation to the theoretical framework.

7.1 CHALLENGES WITH PHYSICAL COMPUTING

As stated in the theoretical framework, computational thinking will be part of everyone’s life eventually, and for that reason, it is essential that everyone is provided with the opportunity to develop these skills. However, not everyone is going to be programmers. When developing these skills with Arduino, students are overwhelmed, and mastering two things, software and hardware at the same time is complicated. Regarding the first research question: “What are the challenges that university level students encounter when learning physical computing?” findings can be divided into two aspects: first the challenges that students encounter when working with concrete step by step assignments, such as the blinking LED, and second the challenges that students encounter when given the assignment to design and create their projects, such as the interactive toy.

Challenges with the step by step assignments were observed during the first and second phase. Because the students were introduced to different components with simple assignments, they lacked
the understanding of what can be done in the real world with different components. The blinking LED assignment will guide the learner on how to create a lamp. However, LEDs can be found in multiple places and can be used for multiple purposes. When wiring components, students struggled to understand closed circuits and current flow. For that reason, it was hard to wire multiple components to the breadboard, as well as to connect the components the right way, or understanding the connection between pins used in the Arduino board and when declaring objects in the code. When programming students encountered challenges with the structure of the code, where everything should be placed and how they should be named, a lot of the step by step challenges can be created with the help of example code without understanding the functions.

Significant challenges were encountered when students were required to apply the knowledge gained from the step by step assignments to their projects. Because previously students used wiring and coding examples without more profound understanding, they were not able to modify these examples to work with their project. These lead students to limit their thinking to components that they managed to wire and program correctly with the previous assignments. Students were struggling to explore what can be done with different components, and lots of the groups were using the outputs and inputs for the same purposes.

7.2 THE PROTOTYPE

The second research question: “How can we support learning computational thinking through design activities?” was explored more in the second and third design phases. The aims of these phases were to create the learning material and use it through design activities. From the beginning, the idea was that these cards would make it easier for the students to be creative when starting to design their projects. Having the set of cheat sheet cards would lighten the overwhelming aspect of physical computing. Based on the observation in the second design phase, the cards were helping with wiring
SUPPORTING LEARNING PHYSICAL COMPUTING

and programming. However, they were not helping the students to design and explore different outputs and inputs.

Challenges with step by step assignments can be eased with visualisations, such as highlighting the current flow in a breadboard or giving tips, such as how to recognise which wire colour in a servo motor goes to what pin in Arduino board. Even more complex programming concepts, such as loops can be explained with the help of visualising the logic with flowcharts and using examples familiar from the real world. For these kinds of challenges to understand the processes of physical computing, already the first version of the prototype was a helpful and useful tool. However, for the challenges that students encountered when working with their projects, the first version of the prototype was not enough. Students needed help on how to construct their project, how to think logically, and where to start. Moreover, this is where the design activity, tested in the third design phase with the second version of the prototype, was found helpful. With the help of the second version of the prototype, only in 70-minutes students were able to design as sophisticated and meaningful projects as students in the one-week workshop. The students in the second design phase designed projects from the perspective of using those components what they already knew how to program and wire. In comparison, the students in the third design phase used those same components without any prior knowledge of how to use them. However, with the help of the front sides of the component cards, and the conceptualising cards, they were able to design a project using these different inputs and outputs. These results relate back to the theoretical framework and to the idea of scaffolding, which is a way to support students in their learning. Vihavainen, Paksula and Luukkainen (2011) describe that scaffolding’s key aspect is that students receive only the most necessary hints allowing them to discover the answers on their own: “[...] learning is most efficient when a student is given just enough information that is enough to boost the student’s ability to finish the task”.
The ground-breaking realisation happened after the second design phase. Support that was needed to explain the abstract concepts had to be something else than visualisations of each part of the process, such as tips how to wire an LED correctly. Moreover, students required help with conceptualising and exploring different technologies. Creating a design of the project with the input, output, and conceptualising cards were working as a bridge between building the understanding of technology and developing the skills to work with the technology. Students could take a problem they have witnessed in the real world and with the help of the cards break this problem into smaller, easier to handle tasks. Breaking down to what inputs are needed to detect the world and what outputs should be triggered. When they had chosen all the inputs and outputs they could start to think logically, if this is detected then this should act accordingly. Students constructed their knowledge of how computational thinking, technologies, and different components work as well as they were immediately able to apply and test their new knowledge when building and programming designed projects.

7.3 LEARNING THROUGH DESIGN ACTIVITIES

As stated in the theoretical framework when designing and creating meaningful projects, students are constructing new knowledge while developing skills required to complete the project. In this study, supporting learning physical computing through design activity was explored. Design thinking, and computational thinking are both tools for problem-solving. Bowler (2014, p. 60) describes design thinking as “[…] an open-ended, nonlinear, and often messy way to generate innovation and creative solutions”. She argues that having the opportunity to think like a designer will encourage students in their creativity. As the design task in this study was open-ended, it gave the opportunity for students to solve a problem they found important and relevant.

When students have the freedom to be creative and generate solutions to problems they find meaningful, they understand more how technologies can be modified and influenced in everyone’s
SUPPORTING LEARNING PHYSICAL COMPUTING

benefit. Students in the third design phase used a larger scale of different inputs and outputs in their projects, they also used these components in more various ways and for various purposes compared to the students in the second design phase. Kafai (1996) asserts that learning through designing these meaningful projects, students can apply their knowledge immediately and experience how these skills can be useful now or in future. Similarly, the educational approach STEAM implies that it is important to prepare students to solve real-world and ill-defined problems and gain skills for jobs that do not even exist yet (Watson & Watson, 2013). These results are essential because developing computational thinking and programming skills should be a right for everyone, not a privilege for few.

7.4 COMPUTATIONAL THINKING FOR EVERYONE

The critical aspect of implementing computational thinking and programming to primary schools is to give the same opportunity to everyone to develop these skills. This will help students to understand what the technologies and digital artefacts we use every day are as well as have the option to go for careers that new technologies have created. Both boys and girls should be encouraged from an early age by introducing computational thinking already in primary school.

In the study conducted by Sentance and Csizmadia (2016, p. 483) teachers were concerned that students lack a realistic view of what computing is all about, and for that reason, girls will not be as interested to learn these skills: “It is a very practical, hands-on subject and we need to be reflecting this in our teaching style”. In addition, Expert A stated that design activities could make it more relevant and exciting for both genders. Not only using robots as inspiration for the assignments, for example, but wearable technologies and fashion have as compelling cases to be introduced to the students. As well as Teacher Assistant C had noticed that boys were taking control over the computer and programming in the workshops:
It is quite intense in the workshop and really fast, especially for people who haven’t done coding. I saw that some of the groups where guys were sitting in front of the program and girls were sitting next to them doing nothing. (personal communication, January 29, 2019)

For that reason, it is essential to introduce computing with activities of various kinds. Buechley, Eisenberg and Elumeze (2007) stated that the vast number of students applying to study in the fields of engineering and computing disciplines are males. Furthermore, Buechley, Eisenberg and Elumeze (2007, p. 1) state that “[...] any effort towards creating new alternative activities that might attract female students and thus address this demographic imbalance should be welcomed.”. Introducing the subject through design activity, students generate the problems and solutions by themselves, giving more space for creating meaningful projects and exploration of technologies.
8 CONCLUSION

This study aimed to answer to the questions what the challenges are when learning physical computing and how we can ease these challenges with design activities in the context of teaching computational thinking as a means to learn problem-solving and programming. A methodological approach, based on research through design and action research, was used to answer the questions. Resulting in the prototype of learning material, a set of cards that was developed, evaluated and modified through the iterative design process including three design phases. This set of cards allowed students to approach computational thinking and programming through a design activity. This study has demonstrated that design activities can provide a more accessible approach to introduce physical computing to students from various majors. Furthermore, this study implies that when students creatively explore and design meaningful projects, they will gain new knowledge and more in-depth understanding of computational thinking while learning the required programming and wiring skills to create their projects. As well as this study has shown that with the set of learning cards, students were able to explore the technology and its usage, create more complex, meaningful, and innovative projects, and develop skills required to build and program these intended projects. Moreover, learning physical computing through design activities allows the learner to develop computational thinking and design thinking skills to solve problems in everyday lives. Finally, this study emphasises that it is essential to develop different approaches to teach computational thinking and programming to a diverse population of people. To give the opportunity for everyone to understand the digital world around us and if wanted, develop the skills to influence it.

8.1 FUTURE WORK

This study focused on testing the prototype and design activities with university-level students. Next step would be to train teachers with these kinds of skills. Beyond the scope of this thesis, a pilot test with teachers using the second version of the prototype was conducted. Even though teachers have
SUPPORTING LEARNING PHYSICAL COMPUTING

university-level education as students researched in this study, they had more problems when starting to assemble the designed projects. More time was required to understand especially the structure and syntax of the program. Teachers’ time outside of the school world to gain these skills is minimal, and for that reason, they were not part of this study.

This thesis can be considered as a pre-study for developing and creating more design activities for introducing physical computing platforms for a variety of learners. Based on the findings, different kinds of design activities could be developed as well as the current prototype could be modified to work with other physical computing platforms. The idea of the conceptualising cards is the same with all physical computing platforms; they have different kinds of inputs and outputs to be explored. As these cards were not developed to be an actual product for the market, the design of the cards was not emphasised in this study. The design of the cards should be revised to make the cards more appealing for the users.
9 REFERENCES

https://www.arduino.cc/en/Main/Education


SUPPORTING LEARNING PHYSICAL COMPUTING


SUPPORTING LEARNING PHYSICAL COMPUTING


79
SUPPORTING LEARNING PHYSICAL COMPUTING


SUPPORTING LEARNING PHYSICAL COMPUTING


SUPPORTING LEARNING PHYSICAL COMPUTING


10 APPENDIX

A. Sketches
B. Teacher assistant interview questions

- When was the first time you were introduced to Arduino?
- Did you attend yourself to one of these workshops?
- How often do you work as a teacher assistant at the workshops?
- How often do you use Arduino and for what purposes?
SUPPORTING LEARNING PHYSICAL COMPUTING

- Do you remember what was the most difficult thing for you to understand or still is?
- If you think of the students, is there something the many of them struggle with?
- What are the biggest challenges when students start to develop their own projects?
- Is there something you want to add?
- Comments about the sketches

C. Card requirements for the second phase

<table>
<thead>
<tr>
<th>Concept</th>
<th>Category</th>
<th>Key elements to include</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uno board</td>
<td>Platform</td>
<td>-Pins</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-Ground</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-How to see is the board working</td>
</tr>
<tr>
<td>Wed editor</td>
<td>Platform</td>
<td>-EduIntro and using libraries</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-How to make a new sketch</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-Serial monitor</td>
</tr>
<tr>
<td>Breadboard</td>
<td>Platform</td>
<td>-Difference between power rails and terminal strips</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-Visualisation for current flow</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-Gap</td>
</tr>
<tr>
<td>Wiring diagram</td>
<td>Platform</td>
<td>-Provide different ways to wire components</td>
</tr>
<tr>
<td>LED</td>
<td>Output</td>
<td>-What are all the things you need to wire one light</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-Positive and negative leg -&gt; wiring</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-What is an output</td>
</tr>
<tr>
<td>Button</td>
<td>Input</td>
<td>-Visualisation of the legs -&gt; wiring</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-What is an input</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-How to read the input in code and use it to output</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-Connection to If then</td>
</tr>
<tr>
<td>Piezo</td>
<td>Output</td>
<td>-Arrays</td>
</tr>
<tr>
<td>Component</td>
<td>Category</td>
<td>Relevant Details</td>
</tr>
<tr>
<td>-------------</td>
<td>----------</td>
<td>------------------</td>
</tr>
</tbody>
</table>
| Servo motor | Output   | - Three different coloured wires -> wiring example  
|             |          | - Where to use   |
| Potentiometer | Input | - Mapping the value |
| LDR         | Input   | - Reading the value |
| Structure   | Programming | - Comments  
|             |          | - Syntax  
|             |          | - Semicolon and brackets |
| Variables   | Programming | - Different types of variables  
|             |          | - Naming |
| If then     | Programming | - Use real life and real code examples  
|             |          | - How to form a condition -> logical operators |
| Arrays      | Programming | - Declare  
|             |          | - Access  
|             |          | - Put and get a value  
|             |          | - When to use it |
| Loop        | Programming | - How it works  
|             |          | - When to use it |
D. The first version of the prototype
E. Creativity Support Index (CSI) survey

Using Arduino Learning Cards

**Collaboration:** It was easy work together with others while using the cards

Disagree | | | | | | | | | | Agree

**Enjoyment:** I enjoyed using the cards and would use them again

Disagree | | | | | | | | | | Agree

**Exploration:** It was easy for me to explore many different options, ideas, designs or outcomes, using these cards

Disagree | | | | | | | | | | Agree

**Expressiveness:** With the help of the cards, I was able to be very creative while creating my own Arduino project

Disagree | | | | | | | | | | Agree

**Immersion:** I became so absorbed in working with the cards that they “disappeared” and became a natural part of my workflow rather than an extra tool

Disagree | | | | | | | | | | Agree

**Results worth the effort:** Using the cards helped me achieve what I wanted to create with Arduino

Disagree | | | | | | | | | | Agree
Learning physical computing

For each pair below, please select which factor is most important to you when learning physical computing with Arduino:

- Enjoyment ___ OR ___ Expressiveness
- Collaboration ___ OR ___ Results worth the effort
- Exploration ___ OR ___ Immersion
- Expressiveness ___ OR ___ Collaboration
- Results worth the effort ___ OR ___ Exploration
- Immersion ___ OR ___ Expressiveness
- Enjoyment ___ OR ___ Collaboration
- Expressiveness ___ OR ___ Results worth the effort
- Immersion ___ OR ___ Enjoyment
- Collaboration ___ OR ___ Exploration
- Results worth the effort ___ OR ___ Immersion
- Exploration ___ OR ___ Expressiveness
- Enjoyment ___ OR ___ Exploration
- Collaboration ___ OR ___ Immersion
- Results worth the effort ___ OR ___ Enjoyment

Free comments about the Arduino cards:

F. The equation for scoring the CSI

\[
\text{CSI} = \frac{(\text{Collaboration Score} \times \text{Collaboration Count}) + (\text{Enjoyment Score} \times \text{Enjoyment Count}) + (\text{Exploration Score} \times \text{Exploration Count}) + (\text{Expressiveness Score} \times \text{Expressiveness Count}) + (\text{Immersion Score} \times \text{Immersion Count}) + (\text{Results worth effort Score} \times \text{Results worth effort Count})}{3}
\]
G. Expert interview questions

- How long have you been teaching physical computing?
- What platforms did you use to teach physical computing?
- What programs have you been teaching?
- What was the typical structure for a course with Arduino?
- How was the different components and concepts introduced?
- Was there something that many of the students struggled with? specific components, concepts.
- What kind of assignments or inspiration did you use?
- How did you measure the learning?
- Is there something you want to add?
- Feedback for the cards

H. Sketch of the second version of the prototype

I. The second version of the prototype
SUPPORTING LEARNING PHYSICAL COMPUTING

IF? ... THEN!

```cpp
#include <Arduino.h>

LightSensor ldr(4);

void setup() {
  Serial.begin(9600);
}

void loop() {
  int ldrVal = analogRead(ldr);
  if(ldrVal > 100) {
    Serial.print("It is dark");
    delay(1000);
  }
}
```

Condition inside if's parentheses can return only true or false.

When the answer is true code after if, inside { } is executed. When false that code gets skipped.

If ldr connected to the board senses light more than 50 this text is printed to Serial monitor.

RESISTORS

A resistor is an electronic component that limits the amount of current in a circuit. But only some of the components in Arduino require them. Resistance is measured in Ohms (Ω). It can be hard to calculate which resistor should be used. For that, resistor calculators are useful tools for finding the right one.

1. USB
2. Reset
3. Digital pins
4. Analog pins
5. Power pins

ARDUINO BOARD

1. Power and upload sketches to board
2. Starts sketch from the beginning
3. 13 digital pins, used both as inputs and outputs
4. 6 analog pins, work only as inputs
5. Power and ground

HOW TO USE MAP() FUNCTION

Map() function re-maps a number from one range to another.

```cpp
map(value, fromLow, fromHigh, toLow, toHigh);
```

value: the number to map
fromLow: the lower bound of the value's current range
fromHigh: the upper bound of the value's current range
toLow: the lower bound of the value's target range
toHigh: the upper bound of the value's target range

From potentiometer's range to servo's range:

```cpp
map(knobVal, 0, 1023, 0, 180);
```

WEB EDITOR

1. Verify, test if the sketch (code) has errors
2. Upload, send the sketch to the board
3. Sketchbook, create a new sketch and store them
4. Libraries, find Edalintro example sketches
5. Serial monitor, read serial data from sketch and board

POTENTIOMETER

Simple knob to control outputs

EXAMPLES WHERE TO USE THEM

- Light dimmer
- Control servo motors
1. ```/** */ and // are for comments, they are used to explain the code but won’t have an affect when uploading the code```  
2. `#include<board.h>`  
3. `void setup() {`  
4. `void loop() {`  
5. `// turn led on after 1 second`  
6. `delay(time);`  
7. `pwm.on();}`

1. **Power rails**
   - Power rails are vertical rows on both sides of a breadboard. One row is connected all the way from top to bottom. Wire one row to 5V and the other to GND, this will give an easy access to power and grounding on breadboard.

2. **Breadboard**
   - Breadboard has a gap in the middle. This gap is perfect size for components with two parallel rows of pins (button).

3. **Terminal strips**
   - Breadboard is full of terminal strips. One strip contains five holes in a horizontal row. When one of them is wired all the holes in that row are electrically connected. One component leg to one terminal strip (except resistors).

4. **Jumpers**
   - Use the same power sources on both sides of breadboard by connecting power rails with jumpers.

**EXAMPLE HOW TO WIRE IT**

Value sent from knob is between 0 - 1023:

When the knob is turned all the way in one direction 0 volts is going to the pin (0 in the code). When in other direction, 5 volts is going to the pin (1023 in the code).

**EXAMPLE HOW TO PROGRAM IT**

Find this example from Arduino web editor

LILIBRARY - EDUNIO - TVOPL - SERVOKNOB

Potentiometer needs an analog pin

Declaring two variables for storing values from potentiometer and from map() function

Potentiometer can return values between 0 - 1023 and servo can take values between 0 - 180. That is why the value read from the knob needs to be re-mapped to values that the servo can interpret.
SUPPORTING LEARNING PHYSICAL COMPUTING

LDR
LIGHT DEPENDENT RESISTOR
Light sensor to detect if it is dark or light

EXAMPLES WHERE TO USE THEM
• Automatic night light
• Camera shutter control
• Distance measurement

FUNCTIONS TO USE WITH BUTTONS
TO MAKE BUTTON WORK LIKE A LAMP SWITCH:

```java
if (button.readSwitch() == LOW) {  // True when button is ON
    led.on();
}
if (button.readSwitch() == HIGH) {  // True when button is OFF
    led.off();
}
```

TO MAKE BUTTON WORK LIKE A DOORBELL:

```java
if (button.pressed()) {  // True the first moment when button is pressed
}
if (button.held()) {  // True when held down
    piezo.play(music);
}
if (button.released()) {  // True when released
}
```

EXAMPLES WHERE TO USE THEM
• Lamp switch
• Remote controller
• Start an engine

DHT11
DIGITAL HUMIDITY/TEMPERATURE SENSOR
Measure humidity and temperature levels

EXAMPLES WHERE TO USE THEM
• Automatic humidifier
• Weather station
• Controlling block heater for a car
**SUPPORTING LEARNING PHYSICAL COMPUTING**

**EXAMPLE HOW TO WIPE IT**

LDR is connected to an analog pin because it doesn’t change only from dark to light (LOW, HIGH), it can send values also between those two. A resistor enables voltage in the Arduino input to change when LDR senses different amount of light.

**CONTROLLING LED WITH LDR**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>GND</td>
<td>ANALOG PIN</td>
<td>GND</td>
<td>ANY DIGITAL PIN</td>
</tr>
</tbody>
</table>

**EXAMPLE HOW TO PROGRAM IT**

Find this example from Arduino web editor

libraries - EDUINTRO - BY TOPIC - LIGHTSENSOR

```c
#include <Wire.h>
#include <Adafruit_LightSensor.h>
LightSensor myLDR;
int myLDRValue = 0;

void setup() {
  myLDR.begin();
}

void loop() {
  myLDRValue = myLDR.read();
  Serial.print("My LDR brightness ");
  Serial.println(myLDRValue);
  Serial.println("Setting and reading the amount of light LDR sends");
  Serial.println("Controlling LED’s brightness with values LDR sends");
  delay(1000);
}
```

*Variables are used to label and store data, different types of variables:
  int = whole numbers, float = decimal numbers, String = text, char = characters, boolean = true or false

**EXAMPLE HOW TO WIPE IT**

Pushbutton has four legs, however two of the legs are one piece of metal (A&B, C&D). Meaning that only one leg from each group should be wired (example B and C).

**EXAMPLE HOW TO PROGRAM IT**

Find this example from Arduino web editor

libraries - EDUINTRO - COURSEWARE - BUTTON

```c
#include <Edumanio.h>
int button_state;

void setup() {
  pinMode(BUTTON_PIN, INPUT);
}

void loop() {
  button_state = digitalRead(BUTTON_PIN);
  if (button_state == LOW) {
    Serial.println("Button pressed...");
    led1.write(1);
  } else {
    Serial.println("Button released...");
    led1.write(0);
  }
  delay(1000);
}
```

* Logical comparison:
  == Equal, != Not equal, < Less than, > More than

**EXAMPLE HOW TO WIPE IT**

Every second DHT11 reads a new value:

<table>
<thead>
<tr>
<th>POWER</th>
<th>GROUND</th>
</tr>
</thead>
</table>

**EXAMPLE HOW TO PROGRAM IT**

Find this example from Arduino web editor

libraries - EDUINTRO - BY TOPIC - DHT11

```c
#include <DHT.h>
DHT dht(11, DHT11);

void setup() {
  dht.begin();
}

void loop() {
  dht.print temperature();
  dht.print humidity();
  delay(1000);
}
```

*Serial monitor (Open Monitor in Arduino web editor):*
To activate the monitor and view data sent from the code or sensor

Print text = Serial.print("Between quote marks write normal language:");
Print variable value = Serial.print(nameOfTheVariable no quote marks);
FUNCTIONS TO USE WITH SERVO

With blue servo motor, write() function will set the angle of the shaft, between 0 and 180.

```c
servo.write(0);
servo.write(20);
servo.write(180);
```

These same functions work differently with the black servo. It is a continuous rotation servo and write() function will set the speed of the shaft. 0 is full speed to one direction and 180 is full speed to the other. 90 degrees will stop it.

FUNCTIONS TO USE WITH LED

These functions will turn led on and off

```c
led.on();
delay(1000);
led.off();
```

Set the brightness for led between 0-255

```c
led.brightness(50);
```

*delay(1000); = waiting for 1000 milliseconds (1 sec) before the code continues to execute the next row.

HOW TO ACCESS VALUES IN ARRAY

Declare an array

```c
int numbers[] = {1, 2, 3};
```

Access an array, 0 is the first value in the array

```c
if(numbers[0] == 1)
```

Put a new value to an array

```c
numbers[2] = 4;
```

Get a value from an array

```c
int x = numbers[1];
```

SERVO MOTOR

When you want to move something you need a motor

EXAMPLES WHERE TO USE THEM

- Robotic arm
- Open and close a door
- Barometer

Blue motor turns 0-180 degrees and black 360

LED

Small light that can be turned on and off

EXAMPLES WHERE TO USE THEM

- Traffic lights
- Blinking eyes
- Led panel with scrolling text

PIEZO

Speakers that can play various tones

EXAMPLES WHERE TO USE THEM

- Sound effects
- Making own music
- Confirmation of user input
SUPPORTING LEARNING PHYSICAL COMPUTING

EXAMPLE HOW TO WIRE IT
Colors matter when connecting a motor:

GND | Brown/Black wire
5V | Red wire
PIN | Orange/Yellow/White wire

EXAMPLE HOW TO PROGRAM IT

Find this example from Arduino web editor
LIBRARIES - EDUCINTRO - COURSEWARE - SERVO

```cpp
#include <Servo.h>

void setup() {
  myservo.attach(0);
}

void loop() {
  for (pos = 0; pos <= 180; pos = pos + 1) {
    delay(15);
  }
}
```

*Servo library needs to be included when using servos
*Change here the same pin number where motor is connected in Arduino board
*This loop will turn the servo as long as position is smaller or equal than 180. At every round -1 is added to pos and shaft is turned amount of the pos.

*For-loop:
for(initialization; condition; increment)
{ this will be executed in every iteration }

EXAMPLE HOW TO WIRE IT
3 tips how to connect the led right way:

1. Long leg Short leg
2. Round side Flat side
3. Small peg Big peg

Stick leg to one of the connection holes, then connect wire to the same terminal strip and the other end of the wire to Arduino board. (More about the boards in the red card.)

EXAMPLE HOW TO PROGRAM IT

Find this example from Arduino web editor
LIBRARIES - EDUCINTRO - COURSEWARE - BLINK

```cpp
#include <Edunino.h>

void setup() {
  Serial.begin(9600);
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
  digitalWrite(LED_BUILTIN, HIGH);
  delay(1000);
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000);
}
```

*Declaring a variable - Type: Name - Value - ;
int howManyTimes = 0;
Led redLight[101];

*Remember to write the name of the led here exactly the same way as when declaring it

EXAMPLE HOW TO WIRE IT
Piezo has two metal legs, connect one of them to a digital pin and the other to ground

EXAMPLE HOW TO PROGRAM IT

Find this example from Arduino web editor
LIBRARIES - EDUCINTRO - COURSEWARE - MELODY

```cpp
#include <Edunino.h>

void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
  digitalWrite(LED_BUILTIN, LOW);
}

void loop() {
  myPiezo.playMelody(0);
  delay(1000);
  myPiezo.selectMelody(0);
  myPiezo.playMelody();
}
```

*An array is a container that holds values stored in it

*Changing notes in this array to play the desired note
This one will play the whole melody.
To access only the first note write melody[0]
SUPPORTING LEARNING PHYSICAL COMPUTING

? Is it more than +20 degrees inside the room?
!

Turn on the fan

SERVO MOTOR
When you want to move something you need a motor

EXAMPLES WHERE TO USE THEM
• Robotic arm
• Open and close a door
• Barometer

DHT11
DIGITAL HUMIDITY/Temperature SENSOR

Measure humidity and temperature levels

EXAMPLES WHERE TO USE THEM
• Automatic humidifier
• Weather station
• Controlling block heater for a car