



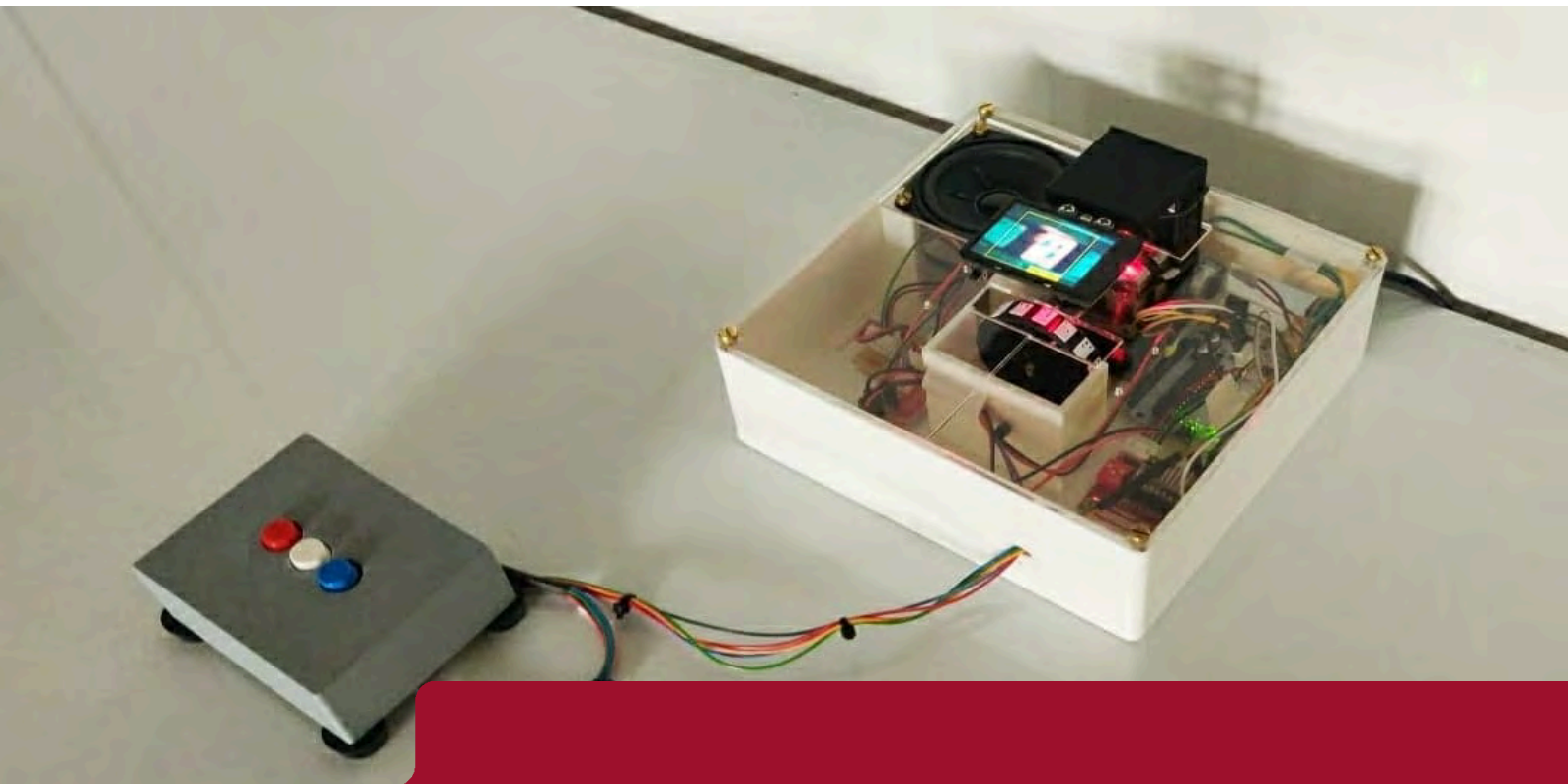
DEGREE PROJECT IN TECHNOLOGY,
FIRST CYCLE, 15 CREDITS
STOCKHOLM, SWEDEN 2021

Brailled

A Braille translation aid

JOSEFIN DANDANELL

AGNES HENRIKSSON



**KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF INDUSTRIAL ENGINEERING AND MANAGEMENT**



Brilled

A Braille translation aid

JOSEFIN DANDANELL
AGNES HENRIKSSON

Bachelor's Thesis at ITM
Supervisor: Nihad Subasic
Examiner: Nihad Subasic

TRITA-ITM-EX 2021:37

Abstract

The purpose of the project was to develop a product that would serve as a support for the learning of Braille.

Learning Braille is a time consuming process. People who have recently suffered a visual impairment have a lower sensitivity to touch than those who have been visually impaired for a longer period. According to a study [1], the learning process can lead to depression. At the same time, the technology for support systems for the visually impaired is underdeveloped.

The questions examined in the project were partly about the environment's influence on the accuracy of the instrument's translation of Braille. Partly, if it was possible to create an instrument that can translate Braille in standard format and at the same time be a suitable aid for a user with a visual impairment.

The instrument consisted of four main parts; reading surface, reading head, audio output and keyboard. The purpose of the reading surface was to place the Braille to be translated under the reading head using a stepper motor. The reading head would then interpret the letter, which would then be called out as an audio file using the audio output. All operations of the instrument would be controlled by a keyboard with three pushbuttons.

The resulting product was promising. The analysis showed that it was fully possible to create a functioning Braille translator. However the product requires some further development in order to be used as an effective aid.

Keywords

Mechatronics, Braille translator, Arduino

Referat

Punktskriftsöversättare

Syftet med projektet var att ta fram en produkt som skulle fungera som ett stöd för inläring av punktskrift.

Personer som nyligen drabbats av en synskada har sämre känslighet för beröring än de som varit synskadade under en längre period. Enligt en studie [1] kan tidskrävande inlärningsprocessen leda till depression, samtidigt som tekniken för stödsystem åt synskadade är underutvecklad.

De frågeställningar som undersöktes i projektet handlade dels om omgivningens påverkan av exaktheten av instrumentets översättning. Dels om det var möjligt att skapa ett instrument som kan översätta punktskrift i standardformat och samtidigt vara ett passande hjälpmedel för en användare med en synskada.

Instrumentet bestod av fyra huvudsakliga delar; avläsningsyta, avläsningshuvud, ljudutgång samt tangentbord. Avläsningsytans syfte var att placera punktskriften som skulle översättas under anläsningshuvudet med hjälp av en stegmotor. Avläsningshuvudet skulle därefter tolka bokstaven som sedan skulle utropas som en ljudfil med hjälp av ljudutgången. Instrumentets samtliga operationer skulle styras av ett tangentbord med tre tryckknappar.

Resultatet av den framtagna produkten var lovande. Undersökningarna påvisade att det var fullt möjligt att skapa en fungerande översättare. Dock krävs det en vidareutveckling för att produkten skall kunna användas som ett effektivt hjälpmedel.

Nyckelord

Mekatronik, Punktskriftsöversättare, Arduino

Acknowledgements

First and foremost we would like to thank our supervisor and examiner Nihad Subasic for organizing seminars, valuable lectures as well as providing guidelines and knowledge throughout the project. A special thanks to Staffan Qvarnström as well as to our assistants Amir Avdic and Malin Lundvall for continuous guidance, support and providing electrical components. Lastly we would like to thank our classmates for invaluable discussions, feedback and encouragement on our project.

Josefin Dandanell Agnes Henriksson
Stockholm, May, 2021

Contents

1	Introduction	1
1.1	Background	1
1.2	Purpose	1
1.3	Scope	2
1.4	Method	2
2	Theory	3
2.1	Braille	3
2.1.1	The standard of Braille	3
2.1.2	Learning of Braille	5
2.2	Components	5
2.2.1	Microcontroller	5
2.2.2	Sensors	6
2.2.3	Lighting	7
2.2.4	Speaker	8
2.2.5	Stepper motor	9
2.2.6	Stepper motor driver and its extension board	9
3	Demonstrator	11
3.1	Circuit Analysis	11
3.1.1	First concept	11
3.1.2	Second concept	12
3.2	Design	12
3.2.1	First design concept	12
3.2.2	Second design concept	13
3.2.3	Development of selected concept	13
3.3	Components	14
3.3.1	Stepper motor	14
3.3.2	Speaker	16
3.3.3	AI Camera (Huskylens)	17
3.4	Hardware	18
3.4.1	Electronics Box	18
3.4.2	Braille wheel	18

3.4.3	Keyboard	19
3.4.4	Camera holder	19
3.5	Software	19
3.5.1	Proof of concept	19
3.5.2	Second version of code structure	19
3.5.3	Third version of code structure	20
3.6	Assembling of the components	20
3.7	Analysis	21
4	Results	23
4.1	Analysis of ambient light	23
4.2	Design and user-friendliness	24
5	Discussion and conclusions	25
5.1	Discussion	25
5.1.1	Analysis of ambient light	25
5.1.2	Design and user-friendliness	26
5.2	Conclusion	27
6	Recommendations and Future work	29
	Bibliography	31
	Appendices	35
A	Flow Chart	35
B	CAD Models	37
C	Acumen	41
D	Arduino Code	45

List of Figures

2.1	Enumerating and format of a Braille cell. [2]	4
2.2	The Braille alphabet. [2]	4
2.3	The size of a Braille cell. [3]	5
2.4	Microcontroller Playknowlogy Uno. [4]	6
2.5	Force Sensitive Sensor (FSR). [5]	7
2.6	Basic construction of a loudspeaker. [6]	8
3.1	Simulated circuit. [7]	11
3.2	Design concept 1. [8]	13
3.3	Design concept 2. [8]	13
3.4	New design of concept 2. [8]	14
3.5	Wire colours of the six wire stepper motor. [9]	15
3.6	Wiring diagram. [10]	16
3.7	Speaker circuit using Talkie.h. [11]	17
3.8	Identifying an object by using object classification. [12]	18
3.9	The resulting prototype. [12]	20
B.1	The CAD model of the electronics box. [13]	37
B.2	The CAD model of the camera holder. [13]	38
B.3	The CAD model of the wheel. [13]	39
C.1	The prototype created i Acumen. [14]	41

List of Tables

3.1	Table of measured resistance. [10]	15
4.1	Table displaying the results from the analysis of ambient light with white dots.	23
4.2	Table displaying the results from the analysis of ambient light with black dots.	24

List of Abbreviations and Nomenclature

List of Abbreviations

<i>3D</i>	Three Dimensional
<i>ADC/DAC</i>	Analog-To-Digital Converter/ Digital-To-Analog Converter
<i>AI</i>	Artificial Intelligence
<i>CAD</i>	Computer Aided Design
<i>Dpi</i>	Dots per inch
<i>ID</i>	Identification
<i>IDE</i>	Integrated Development Environment
<i>LED</i>	Light Emitting Diode
<i>PWM</i>	Pulse Width Modulation

Nomenclature

Ω	Ohm
<i>cm</i>	Centimeter
<i>kB</i>	Kilo Byte
<i>lux</i>	Unit of illuminance
<i>mm</i>	Millimeter
<i>V</i>	Volt

Chapter 1

Introduction

In this chapter, the background, purpose, and delimitations of the project will be presented. The method used during the project will be designated as well.

1.1 Background

Braille is a tactile writing system that enables reading for people with a visual impairment. Each letter is made up of one cell, in which a raised dot can be placed. Depending on the level of encoding used, the placement of the dots represents different letters, numbers or even words. The size of the cell forms an area small enough to fit under one fingertip. The reader can therefore only read one letter or word at a time depending on what type of Braille the user is reading. [15]

People born with visual impairment have been taught to trust other senses like their sense of touch, smell and hearing. This causes them to have developed a greater sensitivity to touch which makes it easy to read Braille. However, for those who recently lost their eyesight, the process of creating a greater sensitivity to touch as well as learning a new alphabet is a lengthy process.[16]

The aim of this project was to develop an aid for the learning process of Braille. This would be accomplished by making an instrument with the ability to recognize and translate Braille into voice messages.

1.2 Purpose

The purpose of the project was to investigate how different types of sensors could be implemented to an Arduino-based system which translates Braille into voice messages. In addition to this, the following questions were also to be answered:

- To which extent is it possible to create an instrument that can translate standard sized Braille and at the same time be a suitable aid for a user with visual impairment?

- How does the surrounding environment affect the accuracy of the translation of Braille?

1.3 Scope

The study focused on developing the technical requirements for a prototype to read individual letters in the Braille system and respond with a voice message. Since the target audience was people who suffered from a visual impairment, the product needed to be sufficiently easy to manoeuvre.

The product were to be built using Arduino and would thus be based on its associated standard components. Therefore, the size of some parts of the prototype were limited to the specified components. The project's resources were delimited to free use of the mechatronics-laboratory's inventory of components, as well as a budget of 1000 SEK for other components that needed to be bought.

1.4 Method

In order to answer the research questions presented in section 1.2, a demonstrator was created. Digital models of the circuit structure were developed via the modelling program Fritzing [7]. The *computer aided design* (CAD) program, Solid Edge [13] was used to create *three-dimensional* (3D) models which were then printed with a 3D-printer. An Arduino Uno microcontroller [17] was used to control and program the desired solution for the electrical components that were used in the system. These included force sensors, an AI-camera, a *light emitting diode* (LED), a speaker and a stepper motor as well as a stepper motor driver and its extension board. An analysis was developed to examine the impact of the surrounding onto accuracy of the instrument.

Chapter 2

Theory

This chapter will go through the underlying theory of the project. Here, the Braille system will be explained as well as the components that will be used in the system.

2.1 Braille

This chapter is divided into two sections. The first section will explain how Braille works, while the second section breaks down the learning process of the system.

2.1.1 The standard of Braille

Braille is a writing system created for people with a visual impairment. The system can be seen as a script that is a standard for many different languages and is used by thousands of people. The system consists of raised dots where each letter is no more than the size of a fingertip. [15]

For each letter, there are six possible slots for a raised dot, which are placed within a surface called Braille cell. The slots are numbered from one to six. They are placed in a $3 \cdot 2$ matrix as seen in Figure 2.1. Depending on the combinations and numbers of the raised dots, 64 different characters are possible to write. Each character has its own combination of the number and placement of the raised dots. The Braille alphabet can be seen in Figure 2.2. Each cell can represent a letter, a character, a number and even an entire word. [15]

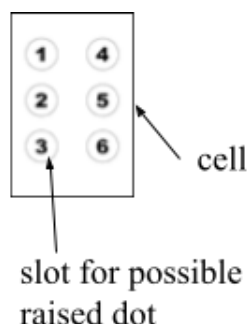


Figure 2.1. Enumerating and format of a Braille cell. [2]

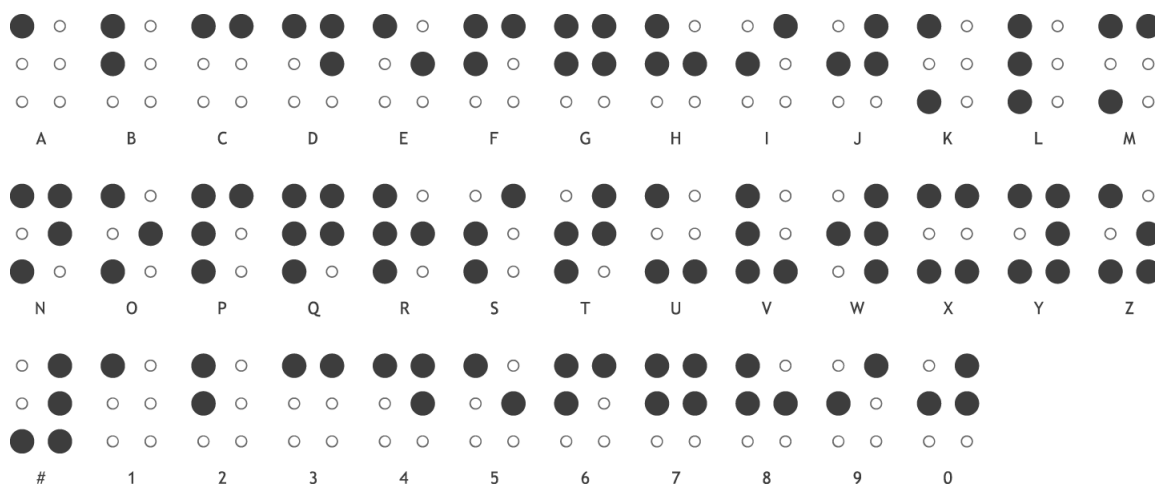


Figure 2.2. The Braille alphabet. [2]

There are two different ways to communicate with Braille; uncontracted and contracted Braille. Uncontracted Braille is the system where text is written letter by letter. This type of writing is mostly used for children’s books and for people who have just started learning Braille. Contracted Braille on the other hand, is more popular as it is faster to read. Here, each cell as well as combinations of cells stands for different words. This can contract a sentence consisting of ten letters to only four. [15]

The size of each cell varies from country to country. However, the majority uses the size which is seen in Figure 2.3. For this standard, the cell must be a size of 3,5 · 6 mm. The distance between each cell must be 6 mm horizontally and 10 mm vertically. Inside the cell, the diameter of each raised dot should be about 1 mm,

2.2. COMPONENTS

the height 0,25 mm and the distance between two dots should be 2,5 mm. [3]

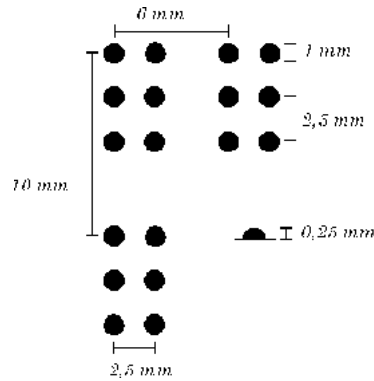


Figure 2.3. The size of a Braille cell. [3]

2.1.2 Learning of Braille

Unfortunately, learning Braille requires a lot of effort. It is even more difficult for those who have lost their sight during the course of their lives. Reportedly, this has led to depression for some. Furthermore is the technology for support systems underdeveloped [1].

Furthermore, people who have had a visual impairment for a long time, have a greater sensitivity to touch in the fingertips. This is because they have improved their sense of touch by having to trust other senses than sight. However, people who lost their sight recently, not only needs to learn a new alphabet, but also improve their sense of touch. [16]

2.2 Components

In this section the main components of the circuit intended to be built are reviewed and explained.

2.2.1 Microcontroller

To be able to interpret signals from components and then from the interpretations control the operations in a circuit, a microcontroller is required. It can be described as a small programmable calculating machine [18]. In order for components to communicate with the microcontroller, inputs receiving their signals are needed. Similarly it needs outputs to send signals to other components.

There is a natural language barrier between a microcontroller and the remaining components. The former operates with the binary language and the latter communicates analogously, in form of voltage. Consequently the microcontroller

needs to translate the analog input into binary code to read it, called *analog-to-digital converter*, ADC. In the same way the microcontroller needs to translate its binary output into an analog signal in order to send it to a component, so-called *digital-to-analog converter*, DAC. [19]

Arduino is an open-source electronics platform based on easy-to-use hardware and software according to the Arduino website [17]. In this course every project group has been assigned the Playknowlogy Uno which is a microcontroller fully compatible with the Arduino Uno, see Figure 2.4. Arduino Uno is a microcontroller board based on the datasheet ATmega328P developed by Arduino.cc. It has six analog and 14 digital inputs, of which six of them have *pulse width modulation*, PWM, support. It has 32 kB flash memory and is programmed and driven with a USB-B cord. [20]



Figure 2.4. Microcontroller Playknowlogy Uno. [4]

2.2.2 Sensors

Normally, Braille is read by touching the text with a fingertip. The finger senses the configuration of the raised dots and the brain associates the pattern with a certain memorized letter. However, the circuit intended to be built could not use a fingertip as sensor out of obvious reasons. Hence it required a sensor of some kind which replaced the function of the finger. There seemed to be two types of sensors suitable for this project, and in the text below they will be explained in more detail.

Force sensors

The raised dots constitute a compressing force when pressed against a surface. Hence a force sensor of some type would make a good fit in recognising the pressure from the raised dots. There are several kinds of force sensors, whereas load cells, strain gauges and force sensitive resistors are a few of them, see Figure 2.5. They are constructed differently but they all send an analog signal in form of a voltage when deformed to the Arduino. [5]

The first concept was based on using a force sensor for each slot where a raised dot can be placed. Therefore six individual force sensors was needed since a Braille

2.2. COMPONENTS

cell is made out of six slots. It was intended to use the smallest sensors available in the storage of the KTH Mechatronics lab.

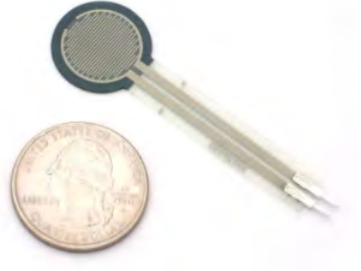


Figure 2.5. Force Sensitive Sensor (FSR). [5]

Vision sensors

The second concept was based on the usage of a vision sensor. This type of sensor can be an AI-camera with the ability to recognise objects, colors, faces and also tracking lines in real time. The main idea in this concept was to use an AI-camera with the ability to recognise and translate Braille letters. There were several AI-cameras suitable for this assignment such as a Huskylens [12] as well as Pixy2 [21].

The camera chosen was the Huskylens. It is an AI-camera made by the company DFRobot [12]. AI is an abbreviation standing for artificial intelligence, and it could be described as "the purpose of artificial intelligence is to artificially mimic the brain's ability to draw conclusions, plan, solve problems, acquire new knowledge, and understand natural language" [22]. This AI-camera has several built in functions, the function used in this project was named object classification and it is one of the new functions in version V0.5.1. The Huskylens, has a two inch display with the resolution 320 · 240, which gives 160 dpi in one dimension and 120 dpi in the other. Generally when speaking of high resolution there should be at least 300 dpi [23]. Therefore the resolution of the Huskylens is quite low.

Object classification allows the user to teach the AI-camera several new objects. For every new object, an ID-number is created. When the object is identified, it can signal the microcontroller via serial communication. [24]

2.2.3 Lighting

Lighting intensity indicates how bright or dull it is in a room. By changing the level and location of the light, the experience of shadows, contrast, colours and atmosphere changes. [25]

To facilitate the recognition of the letters, an LED was added to the circuit. By illuminating the Braille at an angle such that distinct shadows were cast by the

dots, stronger contrasts were created. This would in turn make it easier for the camera to interpret the letters.

By using a red LED, the light from the LED collides with ambient daylight and the lighting from the ceiling, which is often of a more yellow or bluish colour. It further clarifies the contours between the colour of the shadows and the red light from the LED. Through a test where the ambient light is varied, the optimal ambient illuminance can be identified. This was done through a lux measurement using the Swedish Work Environment Authority's lighting app [26].

2.2.4 Speaker

A speaker is a component that is able to transform electrical signals into sound waves. This is done by using a thin membrane, a coil and a permanent magnet, see Figure 2.6. The coil is wrapped around the permanent magnet which is placed in a hollow cylindrical shape. The cylinder is connected to a thin membrane on one of its short sides. When current flows through the coil, a magnetic field is induced according to the law of induction [27]. The induced magnetic field interacts with the permanent magnet, by repulsion and attraction depending on the charge. In this way the cylinder will vibrate back and forth and compresses the air into sound waves. [28]

The speaker was connected with the Arduino using a BC550 transistor.

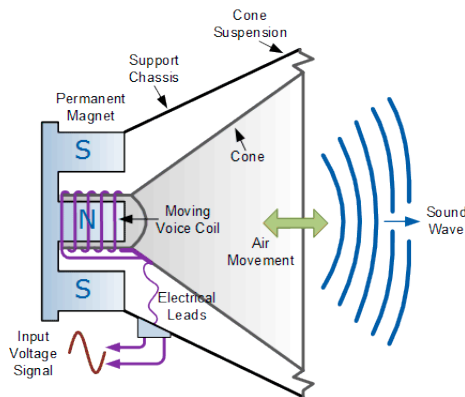


Figure 2.6. Basic construction of a loudspeaker. [6]

In this project, the speaker had to be able to receive a signal from the micro controller, translate it, and then read the letter out loud. Arduino has different libraries which provide extra functionalities for different uses. Talkie.h is a software implementation of the Texas Instruments speech synthesis architecture and contains over 1000 different words that can be used for Arduino [11]. This library will be used to communicate between the Arduino and the speaker.

2.2. COMPONENTS

2.2.5 Stepper motor

A stepper motor were to be used to rotate the wheel with the letters one step at a time. It was of great importance that the stepper motor rotates the wheel with the same angle in each step. Otherwise the letters would gradually move away from the camera's field of view.

This type of motor was chosen due to the fact that it is used in products such as 3D printers, where precise movement and control is desired [29].

The stepper motor works in such a way that a series of electromagnetic coils are charged up positively and negatively at a certain type of frequency. The coils are connected to a series of magnets that rotate at different rates depending on the frequency. The frequency can be specified for a movement forwards, backwards, how fast it should go and when it should stop. Since stepper motors require more power than the Arduino can supply, it must be driven by an external power supply. [30]

There are three types of stepper motors, unipolar, bipolar and universal ones. Depending on how a universal stepper motor is connected it becomes either unipolar or bipolar.

The type used in this project was a universal stepper motor. It has two coils and three wires per coil. Of the three wires within the same coil, one was connected at the beginning of the coil and another to the end of the coil. The third wire was connected to the middle of the coil and it did not need to be connected to the stepper motor driver.

2.2.6 Stepper motor driver and its extension board

A stepper motor driver is a component used to control a stepper motor [31]. The external supply should be connected to the stepper motor driver. The extension board is a component used to facilitate the process of connecting a stepper motor driver into a circuit.

Chapter 3

Demonstrator

3.1 Circuit Analysis

Two possible concepts of the circuit were created. The first one was created to read Braille with force sensors and the other with a visual sensor, an AI-camera. Initially both concepts focused only on reading the dots of one letter. Hence the stepper motor was not yet included in the circuit.

3.1.1 First concept

The first concept used six force sensors to read the raised dots of one letter. They were connected to an analog input to the Arduino as well as a resistor with a resistance of $1\text{ k}\Omega$. When the raised dots were pushed against the sensors, they would read the force input and send the value to the Arduino. The Arduino would then forward the result to the speaker which in turn played the letter. The speaker was connected with a resistance of $100\ \Omega$ and a digital output from the Arduino. This concept was simulated in Tinkercad, since all components existed in the simulator, see Figure 3.1.

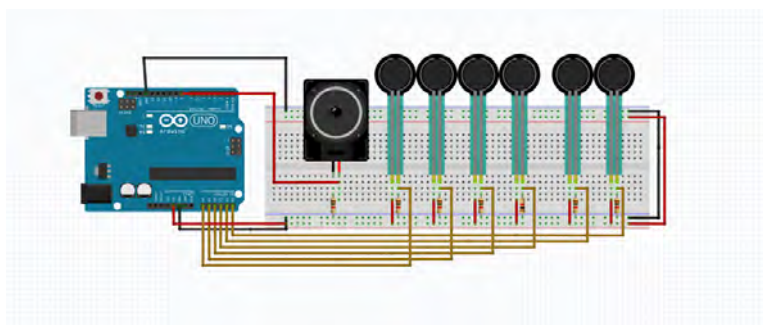


Figure 3.1. Simulated circuit. [7]

A program similar to the one in Tinkercad was then created to test the circuit in reality. The circuit consisted of an Arduino, a breadboard, six force sensors, a LED and a 3D-printed Braille letter. Since the speaker had not arrived yet, it was represented by a turned on LED instead. When the 3D-printed letter was pressed against the sensors the LED was turned on.

3.1.2 Second concept

When the first concept of the demonstrator had been developed, a vision sensor concept concept was further investigated. It was supposed to be using an AI-camera. Since there was no simulation program that had components such as a AI-camera in their circuit simulator, this concept could not be simulated. Instead the components had to be ordered and tested physically.

3.2 Design

Once the second concept had been selected, the design of the prototype was iterated. The iteration allowed for two different main solutions, where either the reading head or the reading surface was intended to be moved. The reading surface was chosen as the area of movement.

The idea that the reading surface were to be moved was that it would allow the user to easily read the letter that the reading device were to translate, without accidentally recognizing the wrong letter. In that way, uncontracted as well as contracted Braille could be taught by the user, who later could use the knowledge to read books.

Also if the reading head were to be moved, longer cables and a bigger area to move around was required. This was considered a risk for the user to get stuck in as well as it would reduce the interaction between user and book, which was not desirable. The prototype was intended to function as an aid that can teach Braille to the user without having to ask for help from another person.

3.2.1 First design concept

The first generated design concept was supposed to have the letters attached to a rubber driver belt. The belt would then be placed between the stepper motor and a rotating shaft. The AI-camera would identify the letter placed underneath it, making the speaker play the letter, see Figure 3.2. The design concept allowed for many words to be placed on the band while not taking up too much space. The disadvantage, however, was that it was not considered to provide a stable surface for the user to read the letters.

3.2. DESIGN

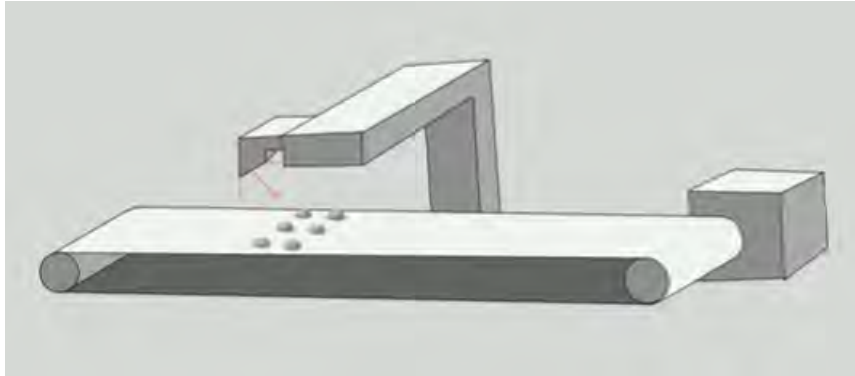


Figure 3.2. Design concept 1. [8]

3.2.2 Second design concept

The second design concept was intended to have the letters written on a plastic cylinder that would rotate around the stepper motor shaft according to Figure 3.3. The advantage of this design concept was that the surface on which the text was placed was much more stable. It was also considered easier to exchange the cylinder since the user only needed to use one hand. Finally, the cylinder, except the top part where the letter to be read was placed, could be hidden inside of a box. This would have made the instrument more robust.

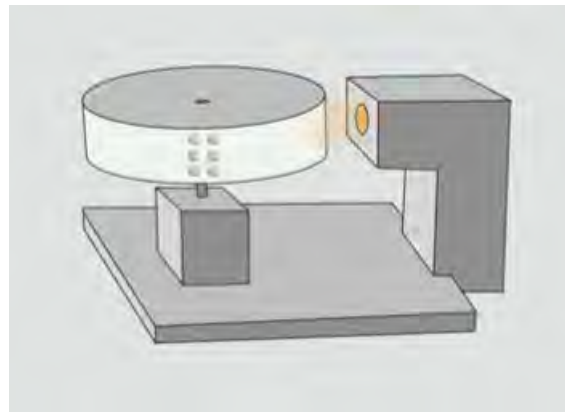


Figure 3.3. Design concept 2. [8]

3.2.3 Development of selected concept

The final concept chosen was the second design concept. The concept was further iterated and the final design is visible in Figure 3.4.

The new design allowed an easier surface for the user to read the correct letters from. All parts that the user did not need to access were placed in a box. The AI-camera was placed at such distance that a finger easily could get between the reading head and the letter while the AI-camera was able to clearly read the differences between the letters.

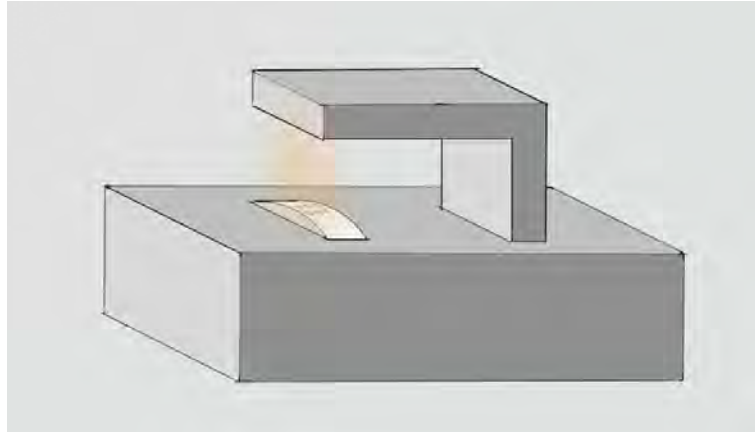


Figure 3.4. New design of concept 2. [8]

3.3 Components

This chapter goes through the process of connecting all components into a singular circuit. The initial step of the process was to test every component with the Arduino and a breadboard. This was made in order to understand how they work and how they should be connected.

The following step was to combine a couple of components with each other and test the different subsystems. The final step was then to combine all of the subsystems into one coherent system.

3.3.1 Stepper motor

The stepper motor used in this project was a two phased stepper motor. Since it only was supposed to carry the load of a lightweight plastic wheel, an assumption was made to not consider neither the load's nor the motors own inertia in when programming the motors movement.

Initial focus was set into configure a circuit consisting of a stepper motor, a stepper motor driver and an Arduino.

The stepper motor had six wires as mentioned in Chapter 2. Out of them, only four needed to be connected to the driver. In order to identify the stepper motor's two redundant wires, the resistance was measured between all wires. The resistance

3.3. COMPONENTS

between the middle wire and an outer wire from the same coil was supposed to be less than the resistance between the two outer wires from the same coil.

The resistance between two wires from different coils were to be zero since the circuit was open.

The resistance between every possible pair of wires was measured and put into Table 3.1. The result of the connections between the wires can be seen in Figure 3.5. The white and black wires were identified as middle wires. The four remaining wires were connected according to the wiring diagram in Figure 3.6.

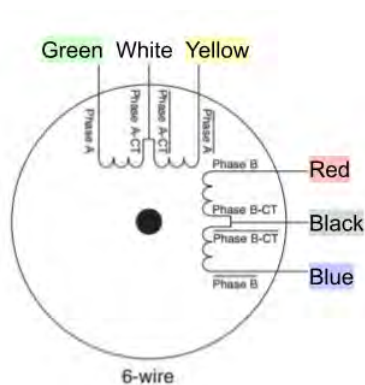


Figure 3.5. Wire colours of the six wire stepper motor. [9]

Table 3.1. Table of measured resistance. [10]

	green	yellow	blue	red	black	white
white	50	50	0	0	0	
black	0	0	50	50		
red	0	0	100			
blue	0	0				
yellow	100					
green						

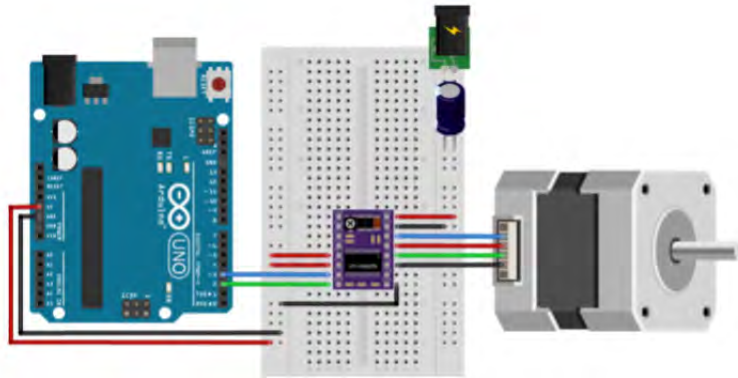


Figure 3.6. Wiring diagram. [10]

Figure 3.6 also shows how the stepper motor driver was connected. The driver was connected to three different pins of the Arduino and to the external power supply. One of the pins was in charge of the steps, the second controls the direction of rotation. The third controls if the stepper motor is disabled from the current or not. However the third wire is not shown in Figure 3.6.

When all components of the stepper motor circuit were connected, it was tested using an example code from the the site "Makerguides" [32] which only took the movement of the stepper motor to account.

3.3.2 Speaker

The speaker used in the prototype had a resistance of 4Ω [33]. It was connected to the Arduino using a BC550 transistor and the speech library Talkie.h [11].

The circuit was connected according to Figure 3.7. The collector pin was connected to the positive terminal of the speaker. The base pin was connected to a digital pin on the Arduino and the emitter pin was connected to ground. The negative terminal of the speaker was connected to 5 V.

3.3. COMPONENTS

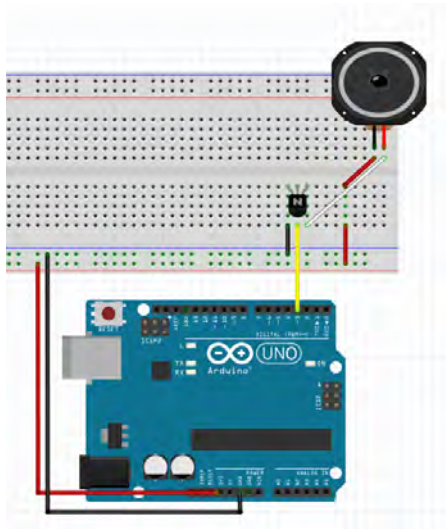


Figure 3.7. Speaker circuit using Talkie.h. [11]

The speaker was then tested with an example code from the library talkie.h [11].

3.3.3 AI Camera (HuskyLens)

The first step of testing the AI-camera was to download the required libraries to the Arduino IDE program. The following step was to understand how the AI-camera's different functions were reached and how an object was taught to the AI-camera.

The following step surrounded how to connect it properly to the Arduino and then test it together with the speaker. To make a test code, an example code given on the website "hackster.io" [34] was modified. The aim of the test was to make the AI-camera recognize a letter as a specific ID and then via the Arduino get the speaker to say the letter connected with that specific ID.

An example of how the AI-camera recognizes one of the letters using object classifications can be seen in Figure 3.8



Figure 3.8. Identifying an object by using object classification. [12]

3.4 Hardware

CAD, 3D printing and laser cutting was used when making the prototype. This chapter intends to explain each one of the components were made.

3.4.1 Electronics Box

A box was made where the majority of the components would be placed. In order to facilitate mounting of the components, the box was split into two parts: a bottom part and a lid.

The dimensions of the box were decided by putting each component's outer shape into a sketch in Solid Edge. Then either recesses or elevations of the components shape were made in the box's base plate to facilitate mounting. The box was 3D-printed.

The lid was divided into two parts. Otherwise the AI-camera had to be dismounted before the lid could be removed. The lid was laser cut in Plexiglas, since an opaque lid displayed rather than hid the circuit and components.

3.4.2 Braille wheel

When making a CAD model of the wheel it was desirable for it to be big enough to fit all letters in the alphabet. Since the English Braille alphabet consists of 26 letters and every Braille letter is six millimeter wide, the circumference of the wheel needed to be $6 \cdot 26$ mm. The height of a Braille letter was six millimeter, therefore the edge of the wheel was made one centimeter tall. See CAD model in Appendix B.

However, when testing the wheel, the AI-camera had difficulty to differentiate between the letters from each other. This was due the proximity of the dots.

3.5. SOFTWARE

Hence, a second iteration of the wheel was made. This time it was made angular with 16 edges, and therefore placing the dots of the letters further from each other. This design also made it more explicit for the user which letter to sense at which time.

3.4.3 Keyboard

The Keyboard was divided into a base plate and a lid. Hence two CAD models were made in Solid Edge and 3D printed. See Appendix B for CAD models.

3.4.4 Camera holder

A CAD model was made of a camera holder, See Appendix B. It was necessary to lock the AI-camera in position at the optimal height from the wheel. Also, it was necessary to leave enough space for the user to place their finger between the AI-camera and the rotating wheel.

3.5 Software

In this section the process of developing the software will be discussed. The code can be found in Appendix D.

3.5.1 Proof of concept

When the components were tested separately, different example codes were used, see earlier references in each component's sub chapter.

The next step was to combine the components into smaller circuits. At this point the example codes required editing and adjusting in order to become compatible with the project.

When testing out the whole circuit, the program was made to function for only two letters. However, it had to be further developed in order to work with all letters and the steering buttons. Here, the case control structure available in Arduino was used [35]. It consists of a certain amount of numbered cases and switches between the cases with a switch variable.

3.5.2 Second version of code structure

Initially the case structure was made with each letter having a separate case. However, the code could be simplified.

Instead, all letters were put into one single case. Hence the code consisted of only three cases: a starting case, a letter case and a case for the motor. At this point the program worked properly, but no regard had been taken to the usage of push-buttons. Therefore it called for further development.

3.5.3 Third version of code structure

The idea of the third version was to have three buttons control the instrument. A ON/OFF-button, a PLAY-button, and a STEP-button. The PLAY-button's task was to make the speaker play the letter and the STEP-buttons task was to make the motor move the wheel one step forward.

The case structure was divided into four cases which can be seen in the flow chart in Appendix A. The flow chart is made with the program draw.io [36]. As long as the Arduino was supplied with power, the instrument was placed in case 0, the OFF-mode. When the ON/OFF-button was pressed, the program switched into case 1 and made the speaker play the word "ON".

Case 1 had three possible paths. If the ON/OFF-button was pushed, the instrument returned to case 0. If the PLAY-button was pressed, the instrument switched to case 2. If the STEP-button was pressed, the instrument switched to case 3.

Case 2 made the speaker play a letter out loud. Case 3 made the stepper motor take one single step, changing the letter on display for the AI-camera. The instrument returned to case 1 from both case 2 and 3.

3.6 Assembling of the components

All electrical components and hardware were then mounted and connected to the box. The result is visible in Figure 3.9. A simulation of the instrument was also created using Acumen and the resulting model can be seen in Appendix C [14].

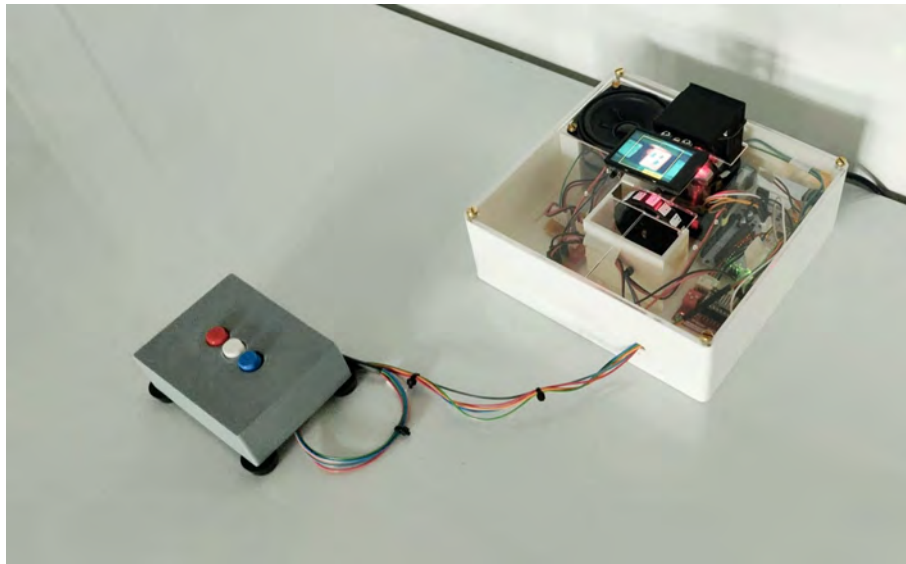


Figure 3.9. The resulting prototype. [12]

3.7. ANALYSIS

3.7 Analysis

When the prototype was finished, a test was made that would examine the type of environment which would make the prototype function smoothly. Since the Huskylens is an AI-camera, the test was reduced to only examine the light that would affect the system. By using a lux meter, the prototype and its function were examined in different levels of illuminance. Thereafter, the illuminance was measured with the aid of the lux meter and the number of errors per amount of letters tested was recorded.

Chapter 4

Results

In this chapter the results of the project are presented.

4.1 Analysis of ambient light

The results of the analysis of the ambient light can be seen in Table 4.1 and 4.2.

For each combination visible in the table, the margin of error is presented as a percentage. Every mean value is based on ten different test iterations.

Table 4.1. Table displaying the results from the analysis of ambient light with white dots.

LUX	DOTS	LED	MARGIN OF ERROR
0	white	on	20%
0	white	off	100%
90	white	on	17%
90	white	off	17%
800	white	on	12%
800	white	off	13%

Table 4.2. Table displaying the results from the analysis of ambient light with black dots.

LUX	DOTS	LED	MARGIN OF ERROR
0	black	on	50%
0	black	off	100%
0	black	camera	25%
90	black	on	12%
90	black	off	6%
90	black	camera	10%
800	black	on	0%
800	black	off	2%

4.2 Design and user-friendliness

The final design concept allowed the user to have full access to the prototype by themselves. This was met, among other things, by placing all electrical and moving parts in a box that the user did not need to have access to during any part of the user process. Since the AI-camera was able to read Braille in standard size, the user was also able to go directly from learning Braille on the demonstrator to reading books and other texts in everyday life.

The design allowed an easy way for the user to find the reading area as well as getting access to it regardless of whether the user was right- or left-handed. The keyboard was designed so that the user could easily learn the mechanism and keep track of all the buttons with one hand while the other hand could read the Braille. Besides playing the letters, Audio files were also used to communicate which button was pressed and what it did by saying "OFF" and "ON".

The prototype was also available for use when learning numbers, characters and short words in uncontracted Braille.

Chapter 5

Discussion and conclusions

5.1 Discussion

5.1.1 Analysis of ambient light

The results indicated that the AI-camera was able to detect the different Braille letters. The prototype was able to recognize each and every one of the letters although it had a generally quite low accuracy when recognizing the IDs.

One of the possible underlying causes might be the quite low resolution of the AI-camera, see "Sensors" Section 2.2.2, about the Huskylens. This might have had an effect on the learning process, especially when the picture consisted of smaller details.

Moreover, the results showed that the AI-camera's ability to recognize a letter mainly depended on two different factors.

The first factor was partly noticed in the beginning of testing and is quite obvious. This factor related to the differences between the taught ID and the scanned ID, due to slight changes in the surroundings. For example changes in position of the letter as well as differences in light.

The margin of error was initially much higher than expected and possible causes were examined. It turned out that the positions of the test performers had a slight effect on the ambient lighting and therefore on the learning process of the IDs. Whenever the test performers changed their positions between the learning process and the actual testing, the AI-camera's margin of error increased. However, when the test performers positions were held fixed, the margin of error decreased. This was due to the slight change of shadows affecting the system.

The margin of error also depended on the placement of the letters under the AI-camera. In the test, the user had to center the starting point by themselves. Therefore, the starting point might have been displaced slightly from the center of frame already from start. Since the AI-camera was quite sensitive, even a small displacement would be able to affect the AI-camera's ability to recognize the ID.

The second factor affecting the AI-camera's ability to recognize IDs was the intensity of the contrast between the raised dots and empty slots. When dots and

background were in same color, the contrast was relatively low.

The purpose of lighting up the Braille cells with a red LED was to increase the contrast. With an ambient lighting of 90 lux, the margin of error decreased slightly when the red LED turned on. With an ambient lighting of 800 lux on the other hand, the LED did not seem to affect the margin of error.

The raised dots were then colored black to increase the contrast even more. This time, the margin of error was significantly lower, almost regardless of the ambient lighting. Overall the best combination according to the results was to have an ambient lighting of 800 lux, black raised dots, and the red LED turned on. However the same combination but with the red LED turned off, gave almost as high accuracy.

5.1.2 Design and user-friendliness

The user can handle the instrument on their own. Therefore the learning process can be tailored after users individual needs. By allowing the user to learn Braille independently, the individual's self esteem and confidence will increase, which in turn increases the user's well being.

Beyond increasing the individual's own well being, the instrument will also give quick access to the learning of Braille as a writing system. This is due to the fact that it is teaching the user Braille of standard size from the beginning.

On the other hand, the users sensibility might not be as developed at the initial state of learning. Therefore, it may be more efficient to initially learn bigger sized letters and then make them gradually smaller. Also, the letters currently used are made in paper, making them fragile. They will be worn out over time if exposed to frequent touch.

Furthermore, the wheel is also sensitive to heavy pressure. This can cause a displacement of letters and in turn affect the accuracy of the instrument. Should displacement occur, the wheel must be reset manually. For a person with visual impairment, this can be a difficult task.

The idea of the learning process is that the wheel can be replaced with another wheel containing new letters. The AI-camera should be able to recognize these without any adjustments. However, in this iteration of the prototype, the entire lid needs to be dismounted to access the wheel. This design was made as a compromise between having easy access and being able to hide wires and other components from the user.

If the initial concept with the pressure sensors was compared to the final concept with AI-camera, it is most likely that the former would receive a more reliable result than the latter. On the other hand, it would be a less user-friendly process to use the pressure sensors as these require pressure to be applied to them. For example this constitutes a risk of fingers being squeezed. Additionally, the Braille cells would most likely have to be bigger than standard size if pressure sensors were to be used.

5.2 Conclusion

Several conclusions can thereby be drawn.

It is fully possible to create an instrument that translates standard sized Braille. The prototype is also a suitable aid for users with visual impairment. For instance the easy maneuvering and the possibility to feel standard sized Braille. The usage of a wheel, the placement of the letter to be read and the concealment of the other components and wires are facilitating aspects.

Conclusions related to the second question, the results were for the most part influenced by the human factor. Therefore it is difficult to determine if minor trends were true or not.

However more explicit trends can be substantiated. A higher contrast between the raised dots and the empty slots increased the accuracy of the AI-camera. Furthermore, a brighter ambient lighting also had a positive effect on the accuracy.

The AI-camera used, Huskylens, was affected by external factors. Therefore it can be concluded that the Huskylens was not robust enough to be used in this project, if the product were to be sold in reality.

Chapter 6

Recommendations and Future work

As shown in the results, there are some areas of the instrument in need of further development.

The biggest problem with the prototype was the AI-camera and its features. As the Object Classification function was relatively new to the Huskylens AI-camera, there was very little information about the function and it was not as well developed as the remaining functions on the AI-camera. Because of this, it had difficulty reading the Braille.

Since external factors affected the instrument's accuracy, it entailed the risk of either confusing or teaching the user improperly. In practise, this limitation made the instrument neither reliable nor salable. The accuracy of the AI-camera should be independent of the surrounding factors but this calls for further development. Since the AI-camera had difficulties reading one letter at a time, it was not possible to develop the prototype to read and translate words or sentences.

Another field of improvement related to the AI-camera is its ability to learn and recognize objects closely positioned to each other. Early on, a cut out sentence of Braille from a paper was placed on the circumference of the wheel. The AI-camera struggled to recognize the individual letters, because it had several letters in its frame.

This is not desirable in reality, since the user should be able to feel the correct placements of the Braille cells. Also, if the correct placement is used, more letters can fit onto the wheel. This gives the user a longer efficient learning time, since the wheel does not need to be changed as often.

Additionally, closer placement of the letters relative each other facilitates the transition from only being able to recognize letters to recognize whole words.

Furthermore if the instrument were to be sold as a product, it would need to have a power supply compatible with a regular wall socket at 230 V. At this point the motor driver is connect to a 12V wall socket available in the mechatronics laboratory.

Lastly an area where improvement can be made is the wheel and its movements. For instance, the wheel only rotates in one direction at this moment. It should be

CHAPTER 6. RECOMMENDATIONS AND FUTURE WORK

able to take a step backwards also, since the user might want to go back and repeat a letter.

Additionally a better mechanism would be implemented regarding the wheel's starting position. At this point, the user moves the wheel into the correct starting position. This could be improved by making the wheel self position itself into the correct starting position every time the instrument is "turned on". Also, there is a slight displacement of the step over time, making the letters gradually move out of the AI-camera's field of vision. This could be avoided by making the instrument self-regulate and move back to the right position.

Bibliography

- [1] Tanaka M, Miyata K, Nishizawa T, Chonan S. Development of a tactile sensor system for reading Braille: Fundamental characteristics of the prototype sensor system. *Smart Materials and Structures*. 2005 04;14:483–484. Available from: <http://dx.doi.org/10.1088/0964-1726/14/4/004>.
- [2] PharmaBraille. The Braille Alphabet; 2021. Accessed: 2021-02-10. Available from: <https://www.pharmabraille.com/pharmaceutical-braille/the-braille-alphabet/>.
- [3] Punktskriftsnämnden. Punktskrift; 2021. Accessed: 2021-02-10. Available from: <https://www.mtm.se/punktskriftsnamnden/punktskrift/>.
- [4] Company K. Playknowlogy Uno Rev. 3 Arduinokompatibelt utvecklingskort; 2021. Accessed: 2021-02-14. Available from: <https://www.kjell.com/se/varumarken/playknowlogy/el-verktyg/arduino/utvecklingskort/playknowlogy-uno-rev.-3-arduino-kompatibelt-utvecklingskort-p88860>.
- [5] Nosonowitz DAL. Force Sensitive Resistor (FSR); 2012. Accessed: 2021-02-14. Available from: <https://learn.adafruit.com/force-sensitive-resistor-fsr/overview>.
- [6] Segovia E. 12. Output devices; 2019. Accessed: 2021-02-08. Available from: <https://fabacademy.org/2019/labs/ied/students/eduardo-segovia/assignments/week12/>.
- [7] GmbH F. Fritzing; 2021. Accessed: 2021-04-04. Available from: <https://fritzing.org/>.
- [8] Procreate. Procreate; 2021. Accessed: 2021-02-20. Available from: <https://procreate.art/>.
- [9] instruments N. Difference Between 4-Wire, 6-Wire and 8-Wire Stepper Motors; 2021. Accessed: 2021-05-08. Available from: <https://knowledge.ni.com/KnowledgeArticleDetails?id=kA00Z000000PAkPSAW&l=en-US>.

BIBLIOGRAPHY

- [10] Makerguides. How to control a stepper motor with DRV8825 driver and Arduino; 2021. Accessed: 2021-02-14. Available from: <https://www.makerguides.com/drv8825-stepper-motor-driver-arduino-tutorial/>.
- [11] Peter Knight AJ. Talkie; 2021. Accessed: 2021-03-20. Available from: <https://www.arduino-libraries.info/libraries/talkie>.
- [12] DFRobot. SEN0336; 2021. Accessed: 2021-03-17. Available from: https://wiki.dfrobot.com/HUSKYLENS_V1.0_SKU_SEN0305_SEN0336#target_5.
- [13] Software SP. Solid Edge; 2021. Accessed: 2021-03-08. Available from: <https://www.plm.automation.siemens.com/en/products/solid-edge/>.
- [14] Acumen. Acumen; 2021. Accessed: 2021-05-08. Available from: <http://www.acumen-language.org/>.
- [15] for the Blind AF. What is Braille?; 2021. Accessed: 2021-02-02. Available from: <http://www.afb.org/info/living-with-vision-loss/braille/what-is-braille/123>.
- [16] Lachiver G, Vachon J, Seufert WD. An Optoelectronic Device to Read and Spell Braille-Braillect. in IEEE Transactions on Biomedical Engineering. 1984;BME-31(8):pp. 560–563. Available from: <http://dx.doi.org/10.1109/TBME.1984.325427>.
- [17] Arduino cc. What is Arduino?; 2018. Accessed: 2021-02-14. Available from: <https://www.arduino.cc/en/Guide/Introduction>.
- [18] Subasic N. Lecture F3. MF133X, KTH. 2021:4.
- [19] Subasic N. Lecture F7, ”Mikroprocessorn i inbyggda system”. MF1016, KTH. 2020:9–11.
- [20] Arduino cc. Arduino Uno Rev3; 2021. Accessed: 2021-02-14. Available from: <https://store.arduino.cc/arduino-uno-rev3>.
- [21] Pixycam. Introducing Pixy2; 2021. Accessed: 2021-04-08. Available from: <https://pixycam.com/>.
- [22] Balkenius C, Skeppstedt J, Gärdenfors P. artificiell intelligens. 2010. Available from: <http://www.ne.se/uppslagsverk/encyklopedi/l/unhbox/voidb@x\bgroup\let\unhbox\voidb@x\setbox\@tempboxa\hbox{a\global\mathchardef\accent@spacefactor\spacefactor}\let\beginngroup\def{\endgroup\relax\let\ignorespaces\relax\accent23a\egroup\spacefactor\accent@spacefactorng/artificiell-intelligens>.
- [23] Christians D. What is hi-res; 2021. Accessed: 2021-05-05. Available from: <https://www.techsmith.com/blog/what-is-hi-res/>.

BIBLIOGRAPHY

- [24] Payá L, Garcia OR. Visual Sensors. *Sensors*. 2020;pp.xi–xii. Available from: <https://doi.org/10.3390/books978-3-03928-339-2>.
- [25] Belysningsbranschen. Visuella Förhållanden. *Ljus och rum*. 2013;3rd ed.:18–27. Available from: https://ljuskultur.se/wp-content/uploads/2016/04/ljus-och-rum_visuella-frhllanden.pdf.
- [26] Arbetsmiljöverket. Mät ljus med din mobil; 2019. Accessed: 2021-05-08. Available from: <https://www.av.se/inomhusmiljo/ljus-och-belysning/mat-ljus-med-din-mobil/>.
- [27] Johansson HB. Elektroteknik. Institutionen för Maskinkonstruktion, KTH. 2013;2:28(Chapter 1).
- [28] M Luz III R. Speakers; 1998. Accessed: 2021-02-14. Available from: <https://web.mit.edu/2.972/www/reports/speaker/speaker.html>.
- [29] Ardestam F, Soltaniah S. Dot Master: Braille printer [Internet] [Dissertation]. 2018;pp. 7.(TRITA–ITM–EX). Available from: <http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-232998>.
- [30] Arduino. Stepper Speed Control; 2018. Accessed: 2021-02-14. Available from: <https://www.arduino.cc/en/Tutorial/LibraryExamples/StepperSpeedControl#unipolar-stepper-circuit-and-schematic>.
- [31] Toma J, Ghebreamlak S. Num2Braille: A braille calculator [Internet] [Dissertation]. 2019;pp.8. Available from: <http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-264437>.
- [32] makerguides. How to control a stepper motor with DRV8825 driver and Arduino; 2020. Accessed: 2021-04-08. Available from: <https://www.makerguides.com/drv8825-stepper-motor-driver-arduino-tutorial/?fbclid=IwAR0MjIHG276Jx78suTwTxexIY-9dHS9UW2NuXGsV3kxFHQ6ae66ZT1D6qk>.
- [33] Farnell. Speaker; 2021. Accessed: 2021-02-17. Available from: <https://se.farnell.com/visaton/2015/loudspeaker-fullrange-7-cm-4-ohm/dp/2357160?st=speaker>.
- [34] hackster io. New Function Update: Object Classification; 2020. Accessed: 2021-04-08. Available from: https://www.hackster.io/Lover/new-function-update-object-classification-e272a9?fbclid=IwAR0qU-qx8wtTyJK1wsZPjxmhirTb9qZ4qc_mr1eG_HmdkgXX04npgJg9JFE.
- [35] Arduino. Switch...case; 2019. Accessed: 2021-04-08. Available from: <https://www.arduino.cc/reference/tr/language/structure/control-structure/switchcase/>.

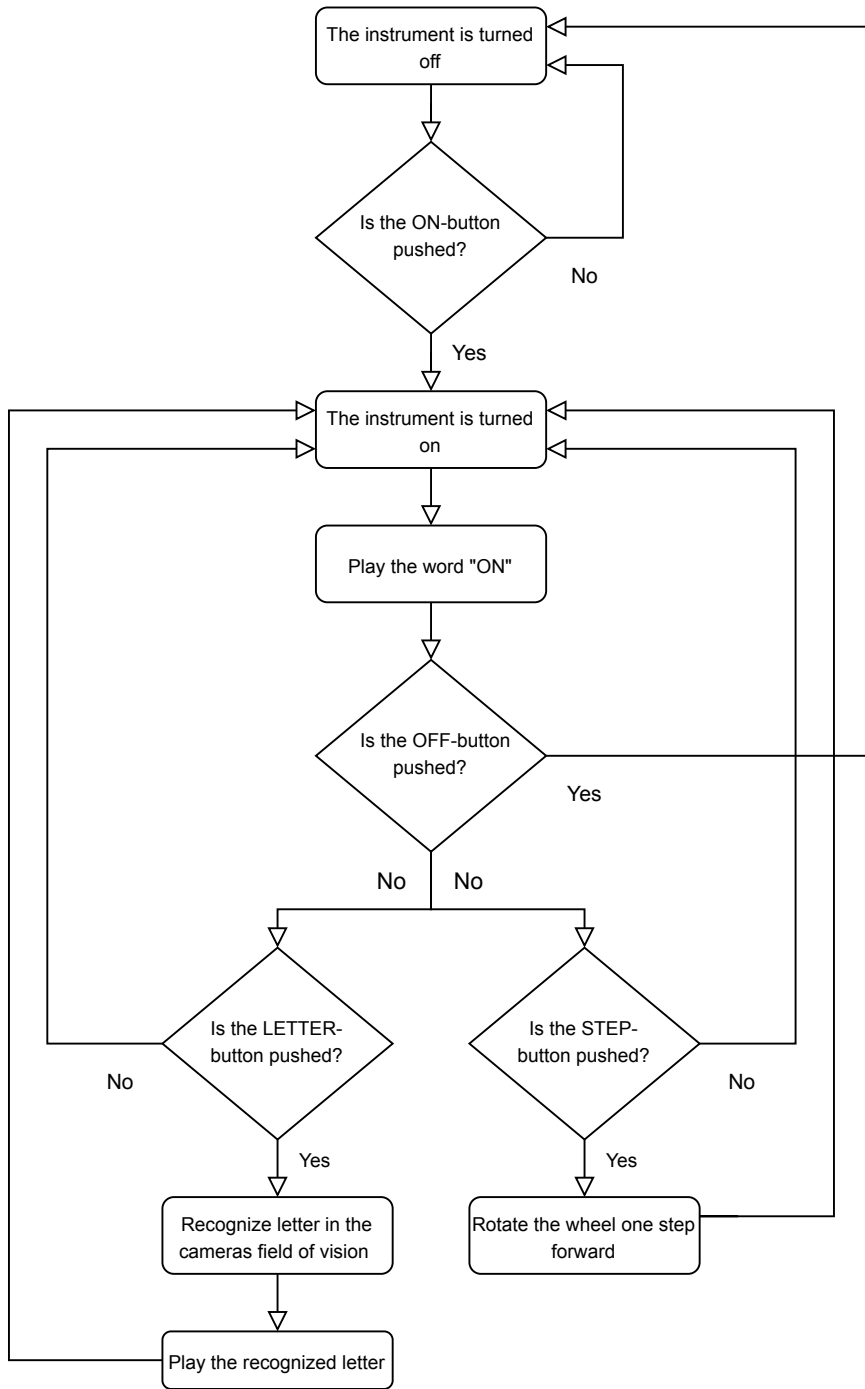
BIBLIOGRAPHY

- [36] io D. Draw.io; 2021. Accessed: 2021-05-09. Available from: <https://app.diagrams.net/>.

Appendix A

Flow Chart

Flow Chart made in draw.io [36]



Appendix B

CAD Models

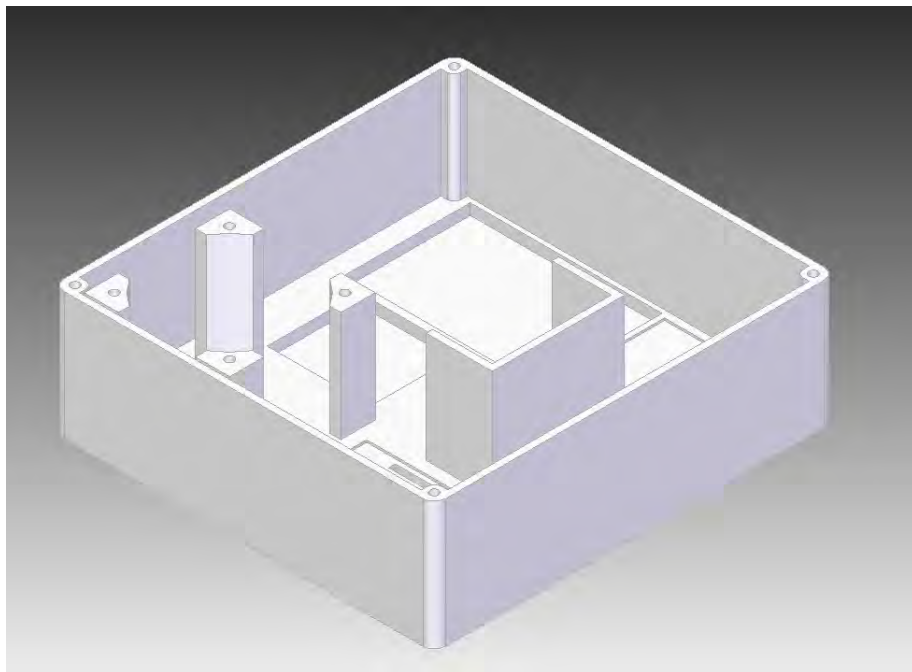


Figure B.1. The CAD model of the electronics box. [13]

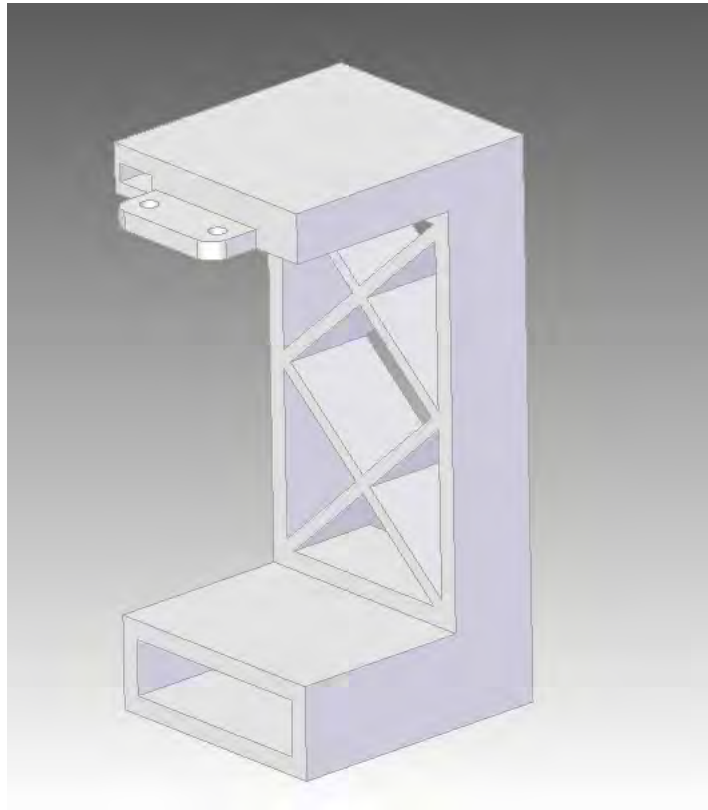


Figure B.2. The CAD model of the camera holder. [13]

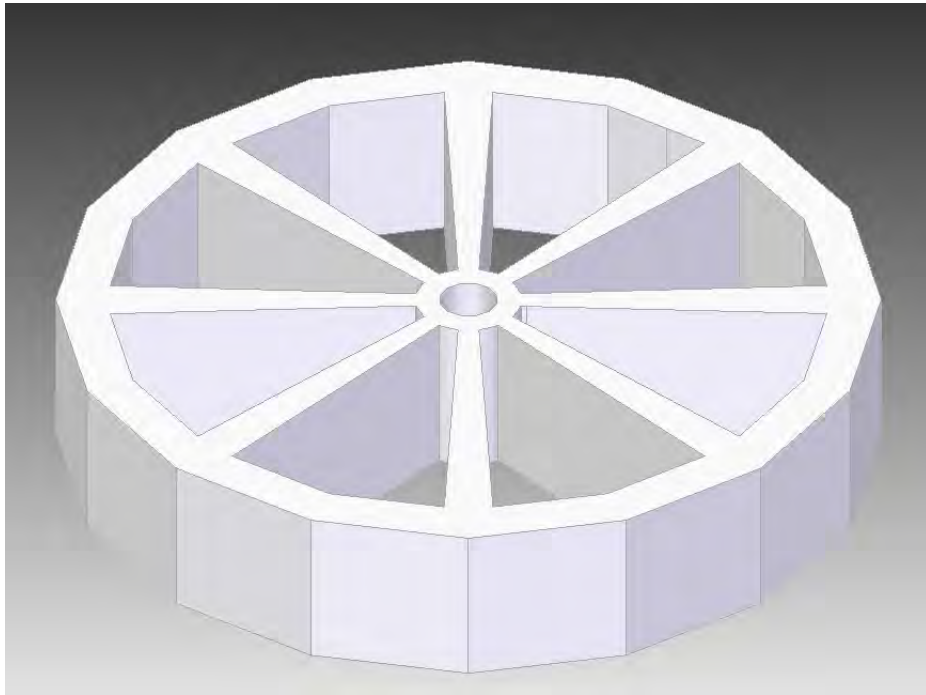


Figure B.3. The CAD model of the wheel. [13]

Appendix C

Acumen

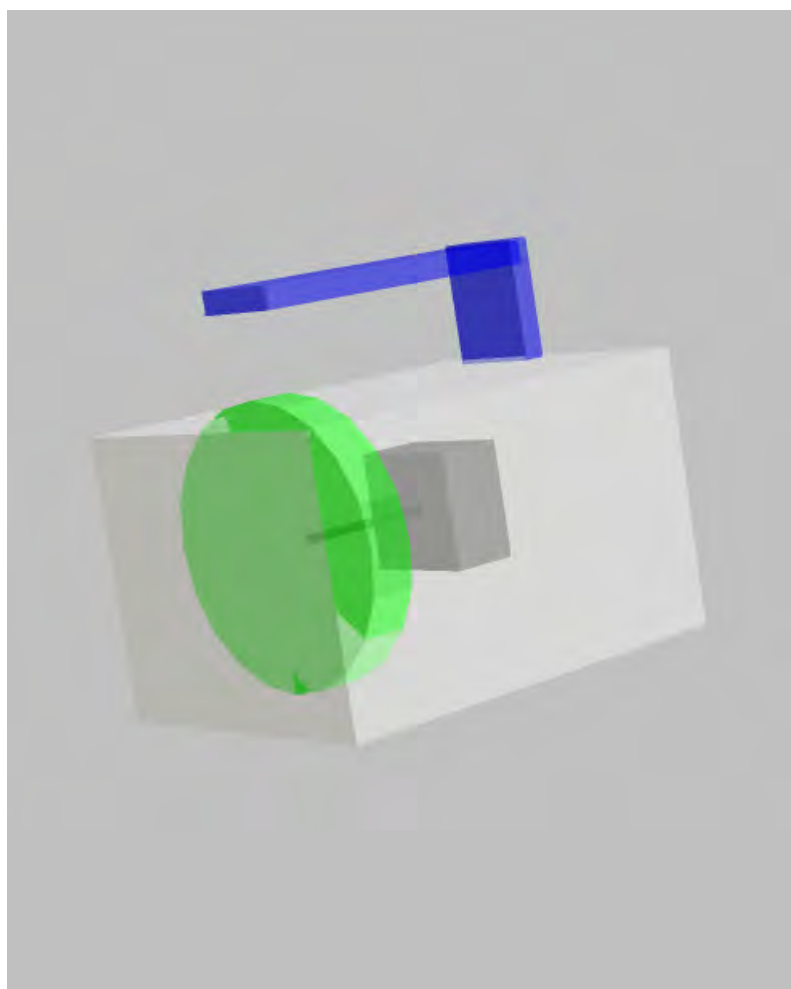


Figure C.1. The prototype created i Acumen. [14]

```

/*
* University:      Royal Institute of Technology, KTH
* Course:         MF133X, Degree Project in Mechatronics
* TRITA no:      TRITA-ITM-EX 2021:37
* Authors:       Josefin Dandanell and Agnes Henriksson
* Program:       CDEPR 3
* Project name:  Braille Translator
* Finalized:     2021-05-08
* Info:          This program is made in order to simulate the prototype of our project

*/

```

```

model Main(simulator) =

```

```

// Defining initial values that the code will follow from start

```

initially

```

angle = 0,           // Initial angle
angle' = pi/5,      // Initiates the angular velocity
rate = pi/5,        // The angle increases with a rate of 1/10 Hz
_3D = (),           // Initiates 3D-object
_3DView = ()        // Initiates camera's starting field of vision

```

always

```

// Setting the value of the angular velocity equal to the value of the variable "rate".
angle' = rate,

```

```

//Defining requirements for the 3D-objects

```

```

_3D = (

```

```

//The rotating wheel, where the braille letters are to be placed

```

Cylinder

```

center=(0,4,0)      //Setting the object's center of placement
size=(1, 3)         //Setting the size
color=green         //Setting the colour
rotation=(0, angle, 0) //Setting the rotation: in y-direction with angle
transparency=1      //Setting transparency

```

```

//The stepper motor

```

Box

```

center=(0, 0, 0)      //Setting the object's center of placement

```

```
size=(2.5,1.5,2.5) //Setting the size
color=black //Setting the colour
rotation=(0,0,0) //Setting the rotation: no rotation
transparency=1 //Setting transparency
```

```
//The shaft of the stepper motor
```

Cylinder

```
center=(0,2,0) //Setting the object's center of placement
size=(3, 0.1) //Setting the size
color=black //Setting the colour
rotation=(0,0,0) //Setting the rotation: no rotation
transparency=1 //Setting transparency
```

```
//Reading-head: Camera and Camera holder
```

Box

```
center=(5, 1, 0) //Setting the object's center of placement
size=(0.5,7,2) //Setting the size
color=blue //Setting the colour
rotation=(0,0,0) //Setting the rotation: no rotation
transparency=1 //Setting transparency
```

```
//Smaller part of reading-head
```

Box

```
center=(4, -2.5, 0) //Setting the object's center of placement
size=(2.6,0.5,2) //Setting the size
color=blue //Setting the colour
rotation=(0,0,0) //Setting the rotation: no rotation
transparency=1 //Setting transparency
```

```
//The electronic's box
```

Box

```
center=(-0.4, 1, 0) //Setting the object's center of placement
size=(6,10,7) //Setting the size
color=white //Setting the colour
rotation=(0,0,0) //Setting the rotation: no rotation
transparency=1 //Setting transparency
),
```

```
//Setting the camera's starting field of vision
```


`_3DView=((1.2, 0, -52), (0, 0, 0))`

Appendix D

Arduino Code

```

/*
 * University:           Royal Institute of Technology, KTH
 * Course:              MF133X, Degree Project in Mechatronics
 * TRITA no:           TRITA-ITM-EX 2021:37
 * Authors:            Josefin Dandanell and Agnes Henriksson
 * Program:            CDEPR 3
 * Project name:       Braille Translator
 * Finalized:         2021-05-08
 * Info:              This code is used for a braille translator.
The translator uses
 *
 *                   a Huskylens camera, a stepper motor, a
keyboard and a speaker.
 *
 *                   The camera uses the function object
classification. Each letter
 *
 *                   of the Braille script is connected to a
specific ID on the camera.
 *
 *                   The camera then sends in found ID to the
arduino which thereafter
 *
 *                   sends out corresponding audio to the
speaker. The stepper motor
 *
 *                   then moves a wheel with the letter on it
from one letter to the next.
 *
 *                   The keyboard has three buttons, on/off,
translating letter, and
 *
 *                   move the stepper motor to the next step.
 */

//Including libraries
#include <SoftwareSerial.h> //Library used for the Huskylens
#include "HUSKYLENS.h" //Library used for the Huskylens
#include <Arduino.h>
#include "Talkie.h" //library used for the speaker
#include "Vocab_US_Large.h" //Library used for the speaker
Talkie voice;

//Defining correct pin to corresponding connection on the stepper
motor.
#define dirPin 2 //Directory pin connects to pin
2
#define stepPin 6 //Stepping pin connects to pin 6
#define enablePin 5 //Enable pin connects to pin 5

//Defining function for the Huskylens
HUSKYLENS huskylens;
void printResult(HUSKYLENSResult result);

int state = 0; //Beginning on case 0
int steps = 49; //Declaring how many steps the stepper
motor will turn (quarter step)
bool condition = false;

//Creating current button states
bool currentButtonState1; // On/Off button - variable
bool currentButtonState2; // Sound button - variable
bool currentButtonState3; // Step button - variable

//Creating last puttonstate
static int lastButtonState1 = LOW; // On/Off button - variable
static int lastButtonState2 = LOW; // Sound button - variable
static int lastButtonState3 = LOW; // Step button - variable

```

```

//Defining correct pin to the corresponding button.
const int ButtonPin1 = 4;           //On/Off button connects to pin 4
const int ButtonPin2 = 7;           //Sound button connects to pin 7
const int ButtonPin3 = 8;           //Step button connects to pin 8

void setup() {

  //Declaring input and output pins
  pinMode(ButtonPin1, INPUT);
  pinMode(ButtonPin2, INPUT);
  pinMode(ButtonPin3, INPUT);
  pinMode(stepPin, OUTPUT);
  pinMode(dirPin, OUTPUT);
  pinMode(enablePin, OUTPUT);

  //Starts the serial communication
  Serial.begin(115200);

  //Starting up the Huskylens camera and recognizing it's settings
  Wire.begin();
  while (!huskylens.begin(Wire)) {
    Serial.println(F("Begin failed!"));
    Serial.println(F("1.Please recheck the \"Protocol Type\" in HUSKYLENS
(General Settings>>Protocol Type>>I2C)"));
    Serial.println(F("2.Please recheck the connection."));
    delay(100);
  }

  //Turning the enable pin to high
  digitalWrite(enablePin, HIGH);

}

void loop() {

  //Reading the corrent button states
  currentButtonState1 = digitalRead(ButtonPin1);
  currentButtonState2 = digitalRead(ButtonPin2);
  currentButtonState3 = digitalRead(ButtonPin3);

  //Saving the current buttonstate to last button state.
  if(currentButtonState1 != lastButtonState1) { //If button one
(on/off) if pressed:
    lastButtonState1 = currentButtonState1; //Save the value
to last button state
  }
  else if(currentButtonState2 != lastButtonState2) { //If button two
(sound) if pressed:
    lastButtonState2 = currentButtonState2; //Save the value
to last button state
  }
  else if(currentButtonState3 != lastButtonState3) { //If button three
(step) if pressed:
    lastButtonState3 = currentButtonState3; //Save the value
to last button state
  }

  //Creating states that refers to the buttons on the keyboard
  switch (state) {

```

```

//Creating a case for the on/off button
case 0:
if (condition == false) {           //If the button is not pressed:
    voice.say(sp2_OFF);              //Voice message "off"
    delay(500);
    condition = true;                //Turning condition to true
}
if(lastButtonState1 == 1) {         //If the button has been pressed to
turn the device on:
    voice.say(sp2_ON);               //Voice message "on"
    delay(500);
    state = 1;                       //Move to case 1
}
break;

//Creating a case for when each button is pressed
case 1:
    if(lastButtonState1 == 1){       //If the on/off button has been
pressed to turn the device off:
        delay(500);
        condition = false;          //Turning condition to false
        state = 0;                  //Move to case 0
    }
    else if (lastButtonState2 ==1){ //If sound button has been pressed:
        state = 2;}                 //Turn to case 2
    else if (lastButtonState3 ==1){ //If step button has been pressed:
        state = 3;}                 //Turn to case 3
break;

//Creating a case that translates the Braille with the Huskylens and
Talkie.h library
case 2:

//Recognizing the Huskylens settings
if (!huskylens.request()) Serial.println(F("Fail to request data from
HUSKYLENS, recheck the connection!"));
else if(!huskylens.isLearned()) Serial.println(F("Nothing learned,
press learn button on HUSKYLENS to learn one!"));
else if(!huskylens.available()) Serial.println(F("No block or arrow
appears on the screen!"));
else{
    Serial.println(F("#####"));

    while (huskylens.available()){ //While the Huskylens
is available
        HUSKYLENSResult result = huskylens.read(); //Read the values
from the Huskylens
        printResult(result); //Turn to void
printresult
    }
}
    delay(500);
    state = 1; //Turn to state 1
break;

//Creating a case that moves the stepper motor one step from one letter
to the next one.
case 3:
    digitalWrite(enablePin, LOW);
    delay(500);
    digitalWrite(dirPin, HIGH);

```

```

        for (int i = 0; i < steps; i++) { //Making the stepper
motor turn 49 steps on quarter step mode //Turning the stepper
        digitalWrite(stepPin, HIGH); //Turning the stepper
motor on to move the motor //Turn off the
        delayMicroseconds(2000); //Turn off the
        digitalWrite(stepPin, LOW); //Turn off the
stepper motor //Turning the
    } //Turning the
    digitalWrite(enablePin, HIGH); //Turning the
enablePin on high //Goes back to case 1
    state = 1; //Goes back to case 1
    break;
}
}

```

//Creating a function that matches each letter ID from the Huskylens to the corresponding letter in the Talkie.h library

```

void printResult(HUSKYLENSResult result){
    if (result.command == COMMAND_RETURN_BLOCK){ //If the camera
is detecting an ID-block: //If the camera
        if(result.ID == 1){voice.say(sp2_START);} //If the camera
is detecting ID1: voice message "start" //If the camera
        else if(result.ID == 2){voice.say(sp2_A);} //If the camera
is detecting ID2: voice message "A" //If the camera
        else if(result.ID == 3){voice.say(sp2_B);} //If the camera
is detecting ID3: voice message "B" //If the camera
        else if(result.ID == 4){voice.say(sp2_C);} //If the camera
is detecting ID4: voice message "C" //If the camera
        else if(result.ID == 5){voice.say(sp2_D);} //If the camera
is detecting ID5: voice message "D" //If the camera
        else if(result.ID == 6){voice.say(sp2_E);} //If the camera
is detecting ID6: voice message "E" //If the camera
        else if(result.ID == 7){voice.say(sp2_F);} //If the camera
is detecting ID7: voice message "F" //If the camera
        else if(result.ID == 8){voice.say(sp2_G);} //If the camera
is detecting ID8: voice message "G" //If the camera
        else if(result.ID == 9){voice.say(sp2_H);} //If the camera
is detecting ID9: voice message "H" //If the camera
        else if(result.ID == 10){voice.say(sp2_I);} //If the camera
is detecting ID10: voice message "I" //If the camera
        else if(result.ID == 11){voice.say(sp2_J);} //If the camera
is detecting ID11: voice message "J" //If the camera
        else if(result.ID == 12){voice.say(sp2_K);} //If the camera
is detecting ID12: voice message "K" //If the camera
        else if(result.ID == 13){voice.say(sp2_L);} //If the camera
is detecting ID13: voice message "L" //If the camera
        else if(result.ID == 14){voice.say(sp2_M);} //If the camera
is detecting ID14: voice message "M" //If the camera
        else if(result.ID == 15){voice.say(sp2_N);} //If the camera
is detecting ID15: voice message "N" //If the camera
        else if(result.ID == 16){voice.say(sp2_O);} //If the camera
is detecting ID16: voice message "O" //If the camera
        else if(result.ID == 17){voice.say(sp2_P);} //If the camera
is detecting ID17: voice message "P" //If the camera
        else if(result.ID == 18){voice.say(sp2_Q);} //If the camera
is detecting ID18: voice message "Q" //If the camera
        else if(result.ID == 19){voice.say(sp2_R);} //If the camera
is detecting ID19: voice message "R" //If the camera
        else if(result.ID == 20){voice.say(sp2_S);} //If the camera
is detecting ID20: voice message "S"

```

```

        else if(result.ID == 21){voice.say(sp2_T);}           //If the camera
is detecting ID21: voice message "T"
        else if(result.ID == 22){voice.say(sp2_U);}           //If the camera
is detecting ID22: voice message "U"
        else if(result.ID == 23){voice.say(sp2_V);}           //If the camera
is detecting ID23: voice message "V"
        else if(result.ID == 24){voice.say(sp2_W);}           //If the camera
is detecting ID24: voice message "W"
        else if(result.ID == 25){voice.say(sp2_X);}           //If the camera
is detecting ID25: voice message "X"
        else if(result.ID == 26){voice.say(sp2_Y);}           //If the camera
is detecting ID26: voice message "Y"
        else if(result.ID == 27){voice.say(sp2_Z);}           //If the camera
is detecting ID27: voice message "Z"
        else{Serial.print("Unidentified letter");}           //If the camera
don't recognize the ID-block: print text
        else{Serial.println("Object unknown");}             //If the camera
don't recognize any ID-block: print text
    }

```


TRITA TRITA-ITM-EX 2021:37