Offloading cognitive load for expressive behaviour: small scale HMMM with help of smart sensors

B.T. (Bastian) van Manen

BSc Report

Committee:
dr.ir. J.F. Broenink
dr.ir. E. Dertien
dr.ir. A.Q.L. Keemink

July 2019

029RAM2019
Robotics and Mechatronics
EE-Math-CS
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Summary

With current technology advancing in a wide variety of fields, more and more robots are found in the modern world. For industrial purposes, military or in the form of toys, they become irreplaceable. However, one relatively new field of application is social robotics. These robots are designed with human-robot interaction in mind and thus human-like and expressive behaviours are crucial for their performance. Instead of being actors, mixing autonomous en deliberate behaviour in one smooth action, many are still passive agents who respond only to direct inputs. HMMM is a software architecture that responds to this problem efficiently by combining behaviour requests, currently applied to robots such as the EyePi. However, these robots also bring their limitations, they remain very expensive, complex and not suited for rapid low-cost prototyping. They can also not be disassembled and reassembled at will by any bachelor level student making then unpractical for robotic or sensor courses.

The goal of this bachelor's thesis is to assemble a collection of off the shelf building blocks in speech recognition, computer vision and visual display, together with a set of programming scripts for rapid and low cost prototyping of minimal social robots. Effectively creating a toolkit to enable bachelor level students of wide backgrounds to create a simple social robot with smart sensors and a simple processing unit.

The toolkit takes into account emotion recognition and multi- modal mixing for a fluent behaviour. The project seeks a simple, low-cost alternative for creating social robots as the previously mentioned EyePi.

Contents

1 Introduction

1.1 Context

As technology continues to advance in the field of robotics, robots become increasingly more present in the daily life and in the industrial sector. In the form of toys, drones or robotic arms they play an important role in today's society. Nonetheless, one area remains fairly new and a lot of research remains to be done, this area is social robotics. The concept of social robotics refers to robots designed with Human-Robot interaction (HRI) in mind. They adhere to social cues and rules while interacting and they are often also able to show emotions.

In the field of social interaction and autonomous conversations, virtual assistants such as the google home or Siri already exist but they have their limitations. For instance, Voice assistants or chatbots only communicate via text or speech reducing their communication capabilities. Physical robots on the other hand, have the ability to convey emotions and imitate human-like behaviour. Applications for such a robot could be a waiter greeting the customers and taking their orders or a receptionist at an office.

So far most Human-robot interactions (HRI) as well as the robots' movements lack this natural fluency humans posses. Life-like behaviour, expressive behaviour and overall environment responsiveness are essential for creating a robot that is effective in HRI and at the same time increase the willingness of the human to interact with the robot (Hoffman (2007)), (Hoffman and Breazeal (2009)).

Currently most robots still hold a passive role, obeying pre-programmed instructions, where a human is always controlling their actions much like a puppet and its puppeteer. To deviate from this model, it is necessary that the robot becomes an actor, acquiring autonomous behaviour such as breathing, blinking, avoiding dead time, and to be able to have a conversation semi-autonomously. The robot then only needs a director to direct the conversation, asking for certain behaviours. Nonetheless, the robot must be able to overwrite the director's request in order to be surprised by sudden movements.

1.2 Related work

Bob van de Vijver achieved this in his master's thesis (van de Vijver (2016)) by implementing the Heterogeneous Multilevel Multimodal Mixing (HMMM) developed during the international eNTERFACE'16 workshop (Davison et al. (2016)), in a small three degrees of freedom robot called EyePi. HMMM will be further discussed in chapter 2.1, for now it is important to know that HMMM mixes different behaviors to create a fluent human-like response. The EyePi is transformed from what was a passive agent responding only to direct controls, to an active agent deliberately executing actions.

Other research also found that even non-anthropomorphic robots are capable of conveying emotions and intent to the user (Mendez (2018)). In his report, Reynaldo explored the persuasive potential of non-humanoid robots by developing a 5 DOF robot arm shaped like a desk light. Through the use of different colors and body movements the robot was successful in overcoming the lack of verbal cues and facial expressions. The HMMM system was again used to blend multiple behaviours such as general body movements and breathing motions into one smooth action.

Finally, in the work of Suhaib Aslam (Aslam (2018)), a process for adults with autism to co-create their own social robot with the help of a human facilitator was created. Called the SoCo process, it uses a set of 5 different types of cards corresponding to the 5 different choices that the user will have to make during the prototyping phase. The cards give information about the

various tasks it should accomplish while the robot is made by assembling different building blocks to enable it to listen, see or talk. Suhaib validated the process through a case study he realized with three participants of the target group.

Although, agents such as the EyePi integrated with HMMM are social actors and can be used as platform for future projects. They remain very complex and unsuited for inexperienced users in programming and robotics. Moreover, the EyePi costs around €900 without counting the many hours spend on building and programming, which makes it a rather expensive device. Consequently, it is not a robot that can be handed out to students for disassembling, a hands-on course or Suhaib's project. If it would be possible to find a collection of building blocks that already contain many of the complex algorithms required to create a social robot, one can drastically reduce the complexity of the design and facilitate the entire process. Thus effectively offloading the computational load from one central computer to multiple smart sensors tied together by a simple micro-controller such as the Arduino Mega. All while at the same time inheriting HMMM traits such as autonomous behaviour, emotion mapping and salience detection as basis built into the building blocks.

1.3 Research goal and report outline

In this bachelor's thesis the main objective is to achieve this by creating a toolkit which consists of an assemblage of off-the-shelf components functioning as building blocks, along with a set of example scripts. Designed for students and teachers, the collection of building blocks should allow for rapid and simple prototyping of simple social robots while minimising the cost. In this report, the word toolkit is defined as a collection of tools required to achieve a specific goal. Here, the tools allude to the off-the-shelf building blocks as well as all the software and hardware developed during the course of this bachelor's thesis. The goal is creating a minimal social robot.

All building blocks considered in this report covering the fields of speech recognition, computer vision and visual display, will be compared allowing for easy selection in the robot's design phase. The toolkit will be put to test by designing and constructing a prototype of a simple social robot. According to the design criteria of the prototype, the most fitting building block of each field will further be tested on an individual level first, to provide a stepping stone for the prototype as well as evaluate the building blocks. The final product should be able to show basic autonomous behaviour of blinking, gazing and breathing. Moreover it should be capable of mixing this with external inputs from the sensors, much like the high level architecture of HMMM does. It's performance will consequently be compared to the EyePi implemented with HMMM. It is important to understand that the goal of this bachelor's project is to create a set of tools that can be utilized and combined to build minimal social robots. The prototype designed in this report is used to test this.

The report starts with the analysis of the existing software architecture of HMMM which is broken down into minimal viable functional components in chapter 2. Next, multiple existing social robots are explored and analysed after which the requirements for the prototype, the toolkit and subsequently the building blocks are defined. In chapter 3, state of the art building blocks in speech recognition, computer vision and visual display are compared from which three are chosen for further testing. Concluding chapter 3 with the design of a minimal social robot. Chapter 4 shows the results of the prototype evaluation, toolkit validation and building block assessment based on the requirements set in Chapter 2. In this chapter, the minimal social robot will also be compared to the existing EyePi. Chapter 5 concludes the report and some recommendations are given for future work.

2 Analysis

2.1 Breaking down existing software architecture

In the first stage of the analysis it is necessary to acquire knowledge about the essential components that make out the architecture of HMMM. As this is the software used in the EyePi to control it's behaviour, knowing the minimal viable functional components it is composed of, offers better insight in the role each type of building block will be playing in the final prototype.

In the paper of Davison et al. (2016) describing HMMM, fluent dialogues and behaviour is obtained through the architecture used during the international eNTERFACE'16 workshop consisting of four different components: perception module, dialogue manager, behaviour realiser and agent control. An overview of the system architecture can be found in figure 2.1. It shows the relations between the four components as well as the modules or applications that fill in these roles.

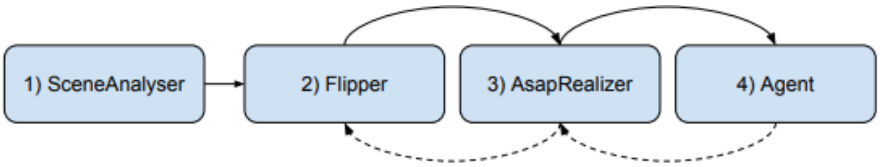

Figure 2.1: An overview of the eNTERFACE system architecture, highlighting four distinct components: (1) the signal acquisition module SceneAnalyzer; (2) the dialogue manager Flipper; (3) the behaviour realiser AsapRealizer; (4) the agent (for example, the Zeno or EyePi robots, or a virtual agent created in Unity). (Davison et al. (2016))

Although this report will not go in great lengths in how HMMM exactly works, it is clear that the building blocks together with the micro-controller should satisfy each role in figure 2.1 in order to obtain comparable results from the prototype and the EyePi. Now looking at each separate component, the different tasks that must be performed by the collection of building blocks can be laid out.

Starting with the perception or signal acquisition module. Here executed by the SceneAnalyzer with a Kinect sensor, it detects if a person is speaking, head, spine and hand movements as well as proxemics and gestures. This part of the system architecture is crucial for the proper functioning of the agent as it provides it with important information about the outside world.

The Second block takes care of the dialogue management. Realized by Flipper, it supplies fluency to the dialogue between agent and user. Context and user actions are furthermore taken into account. Information is then communicated with the next part in the architecture handling the movement of the agent.

As Flipper manages the conversation, the fluency comes from AsapRealizer. Developed by the Human-Media Interaction group at the University of Twente (van Welbergen and Kopp (2014)), it uses a behaviour mark-up language (BML). A BML is a tool that describes the autonomous actor behavior in virtual or real agents. Along with fluent interaction, AsapRealizer is responsible for the human-like movements of the agent and resolves the synchronisation issues. AsapRealizer administers the motors responsible of the actual movement of the agent, while receiving and providing feedback to the agent and the dialogue manager.

Finally, relating the above back to the project, a smart camera with integrated computer vision algorithms such as face tracking and saliency detection could replace the SceneAnalyzer. Flipper and AsapRealizer however are more difficult to integrate in the prototype due to the their shear complexity and variety of tasks. To overcome this, the dialogue management and the be-

haviour realizer will both be tackled directly in the Arduino IDE. A speech recognition module together with a synthesizer module are required for the dialogue manager and servos for the behaviour realizer.

This architecture can currently be found for instance in the EyePi. Figure 2.2 shows how HMMM is implemented in the robot and how the EyePi looks like.

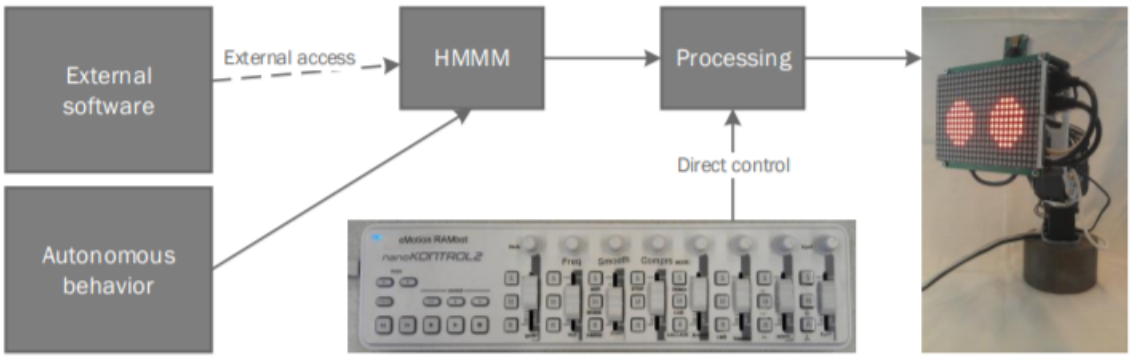

Figure 2.2: A schematic overview of the EyePi system and the EyePi (van de Vijver (2016))

One can see that the director's request is mixed with the autonomous behaviour.

Previous versions were already equipped with autonomous characteristics such as breathing and blinking but were only controllable using a MIDI panel, see figure 2.2.

Among updating the hardware, Bob also showed comparisons between a robot with and without HMMM enabled. This is done for the specific case of which the robot is interacting with a person and something unexpected happens in the robot's field of view. In his experiment, he noticed that only the robot with HMMM reacted to the sudden movement next to the person.

Additionally, Bob describes the emotion mapping inside the EyePi. Based on arousal and valence levels, the robot's emotion is illustrated in figure 2.3.

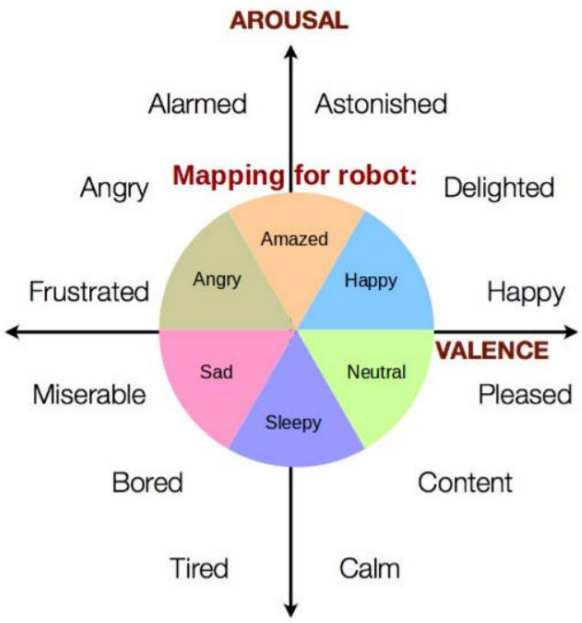

Figure 2.3: Arousal and valence to robot emotion mapping (van de Vijver (2016))

Arousal determines the degree of awakening while valence the degree of positivity.

2.2 Existing social robots and available building sets

Simple social robots

Until now the prototype that will later in the report be designed to test the toolkit, remained largely undefined. As the goal of the prototype is principally to prove that a relatively simple social robot can be successfully designed using low-level off the shelf building blocks, simple social robots are more interesting in the scope of this project.

Starting with Kismet, a robot that is considered to be the first social robot. Kismet perceives the world using a camera for motion, color and saliency detection. The robot is not actually capable of understanding what the user says, instead it recognizes tonality and volume to interpret the user's emotions (Breazeal and Scassellati (1999)). Figure 2.4 illustrates the robot, one can see that it disposes of multiple motors located at key-points to portray facial expressions.

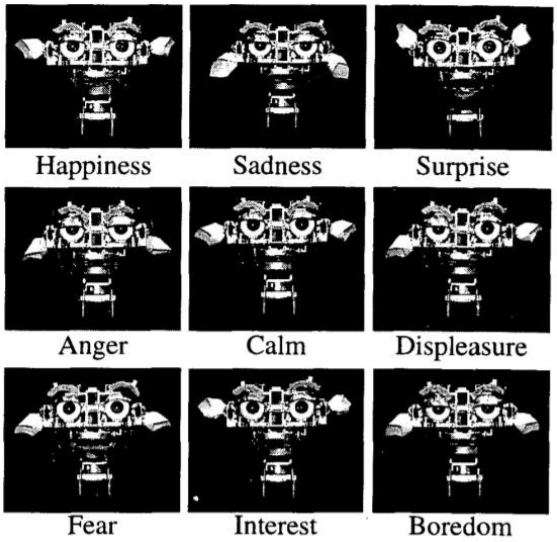

Figure 2.4: Kismet, a robot capable of conveying intentionally through facial expressions and behavior (Breazeal and Scassellati (1999))

Without speech recognition, Kismet interacts with the human by means of turn taking. Once it detects the user has finished talking, it reacts appropriately depending on the measured volume and tonality. Kismet was designed by MIT in the late 1990's, it truly represents the basis of social robotics.

Another good example of low-level social robotics is Ono, a small plushy like robot designed to help children with autism (Vandevelde et al. (2014)). Developed in the year 2014, Ono is more recent than Kismet but includes many of the same features. Eyebrows, eyes and mouth are equipped with motors to create an array of facial expressions, creating a total of 13 DOF for the entire face, figure 2.5.

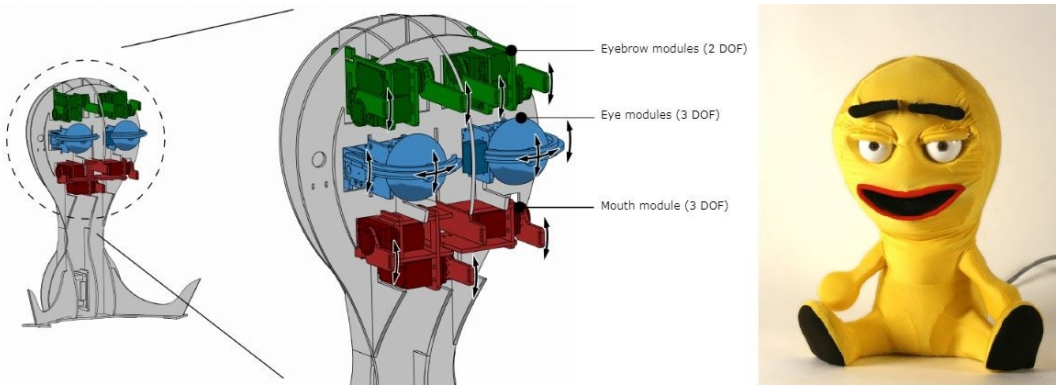

Figure 2.5: CAD model showing construction and DOFs of Ono. Servo actuators are shown in a darker color and the current prototype of social robot Ono (right) (Vandevelde et al. (2014)).

At this stage of Ono's development, it did not yet carry sensors and could only be controlled through a joystick interface. It is interesting to see that both Kismet and Ono opt for physical elements instead of a display as seen previously with the EyePi. It can be fairly safe to assume this is to add realism to their prototype.

Similar to Ono in the fashion of DIY oriented projects, the cuddlebits are a very popular platform for social robotic experimentation specifically for haptic display. Cuddlebits are small furry animals that as opposed to Kismet and Ono, only have 1 DOF (Cang et al. (2015)). They are accommodated with a fabric containing pressure sensors able to recognize different gestures such as rubbing, tickling and patting. Cuddlebits have motors attach to their rib cage that can simulate breathing movement.

All three robots incorporate different design choices which make them unique. Nonetheless, they remain related by their design approach. Kismet focused on the facial expression capabilities while Ono and the Cuddlebits concentrated on the robot's appearance. This is a point brought up further by Hoffman (Hoffman and Ju (2014)), who distinguishes between two common design paths. The first being the pragmatic approach and the second the visual. Pragmatic refers here to requirements set for the robot's spatial activity, while visual denotes a design with appearance in mind. In his paper, Hoffman presents a different design method where the quality of the expressive movement is most important. He argues that expressive movement is a powerful tool for interaction regardless of the robot's appearance. He applies the method on a robot head capable of playing the marimba instrument. Design is done with movement in mind, using 3D animation tools.

At last, Rory the cute robot plant falls in the same category as many of the above discussed robots despite its simple and unfinished look. Seen in figure 2.6, Rory is an Arduino based project equipped with facial recognition, interactive talking, mouth expressions and multi-sensor monitoring.

Figure 2.6: Rory the robot plant (Azouz (2019))

Available as a DIY project for the common hobbyist, it only requires minimal knowledge about programming and electronics to make (Azouz (2019)). The plant is made from cheap low-level components and is open sourced. It actively monitors the moisture level in the small plant situated in front and notifies accordingly the user.

Concluding this short exploration of simple social robots, the prototype that will subsequently be designed will resemble a combination between Kismet and Rory the robot plant. First, Rory defines the physical appearance of the prototype in the sense that it will be made of similar inexpensive low-level components. Second, the skills that Kismet had are taken as goal for the prototype. Not seeking to recreating Kismet, it gives a good idea of what the prototype designed as part of this bachelor's thesis should be able to do.
The reason for these design choices, including the pragmatic design approach that will be taken during the prototyping, can be explained by referring back to the initial goal of the prototype. As the core function of the robot is to test and validate the toolkit, it is sought to create a minimal social robot inheriting HMMM features. Kismet was chosen as example for the prototype as it is one of the first concept of social robotics. Rory in turn provides a good example of the level at which the prototype will be designed at. The specific requirements can be found in section 2.3. Note also that during the remainder of the report, the prototype will sometimes be referred to as Kismet due to the reason previously mentioned.

Building sets

Before defining the requirements for Kismet, the toolkit, its users and the building blocks, first a look is taken at which ready to buy building kits already exist on the market. Defined under building kits are DIY packages including hardware and software completed with building instructions. The advantages of building kits is that they already are small toolkits on themselves, which if used correctly could make the prototyping process cheaper, easier and faster. It is important to first create an overview of the available building kits as it may help accurately set the requirements subsequently.

One of the most prominent and well known building kits available today are Google's AIY series. The Google AIY vision and voice kit both include a fully assembled Raspberry Pi zero with a camera module or voice bonnet and speaker. Without any soldering required, it enables the user to use artificial intelligence pre-loaded on a SD card to create a smart google assistant or an image recognition device. They can be bought on Adafruit for €87.78 and €52.65 respectively.

In the same spirit, the ReSpeaker box coming at a lower price of €36.95 (on Kiwi electronics) also gives the opportunity to create a google assistant. Although cheaper, it does not come with a Raspberry Pi which consequently must be bought separately. These kits are all equipped with a small protective housing designed for testing only, after which the internal building blocks can be rearranged and assembled in a bigger project. Advantages of using them include direct implementation capability of full google voice assistant and object recognition technology. Limitations occur when access to internet is a problem. An active wifi connection is crucial, without it the Raspberry Pi cannot access the artificial intelligence algorithms made by google. Besides, a credit card number is needed when buying a Google AIY kit, making it less attractable for educational purposes and thus this toolkit.

Also commonly used in learning environments are Lego mindstorms. Existing in the form of multiple ready to build packages as well as individual components, Lego mindstorms could be a potential candidate for this toolkit. On top of the Lego mindstorms, several other robot building kits are available on the market today. Provided on stores like Sparkfun and Antratek, they consist usually of small line following robots and alternatives. Nevertheless, very useful and suitable for the average DIY'er, their pricing remain over the 100 euros. Which for a low-level toolkit is quite expensive considering many sensors would still need to be added.

2.3 Requirements

Specific requirements can now be set for each part of the project. The requirements are divided into two categories: qualitative and quantitative. Both must be validated in section 4.

2.3.1 Kismet

Kismet must be able to perform basic social robotic functions in order to be considered a social robot. As such it is chosen that Kismet should be able to understand and display at least six different emotions, based on the aforementioned emotion mapping in section 2.1. Emotions expressed by the robot should reflect those of the user. This not only gives a reason for understanding the user's emotions but also echos empathy towards the opposing person. Empathy benefits the human-robot interaction as humans tend to find robots showing empathy friendlier than if presented with a neutral face (Iolanda Leite (2013)). Additionally, it should be able to recognize if one or multiple users are present and no internet connection should be required for operation. Running offline not only simplifies the system but also enables it to be used anywhere without the need to establish a connection first. Lastly, Kismet must be entirely constructed from smart sensors and low-level micro-controller, with minimal soldering required, enabling it to be taken apart and reconstructed into a different configuration.

Until now, the requirements describe most of Kismet's functionalities as designed in the late 1990's. This is however not sufficient to compare it to the EyePi at a later stage. For this reason it is added that Kismet must be able to follow at least one face during conversations and show gazing, blinking and breathing motions mixed into Kismet's behaviour. Finally, going a little further, it is required that Kismet can interact verbally and incorporates a program for fluent conversation, contrary to turn taking used in the first Kismet. To ensure fluency during the interaction, the robot should be able to respond within one to three seconds. A budget limit of €300 is also established for the total cost of the prototype. Physical appearance is not of importance as it does not influence the goals set in the beginning of this report.

	Requirements
Qualitative	- Reflect user emotion - Gazing - Blinking - Breathing motion - Verbal communication - Fluent conversation - Behaviours mixed together - Offline - Made from smart sensors and micro-controller - Minimum amount of soldering required (simple design)
Quantitative	- Recognize >= 6 emotions - Display >= 6 emotions - Detect >= 2 faces at a time - Follow >= 1 face - 1s <= audio response time <= 3s - Total cost < €300

Table 2.1: Qualitative and quantitative requirements for the toolkit's prototype (Kismet)

Requirements must be met in real-time when applicable.

2.3.2 Toolkit and users

Apart of the more straight forward requirements, the toolkit and it's users should comply to certain specific prerequisites. First, basic knowledge in Arduino and python programming is required from the user. This to be able to work with some of the building blocks present in the toolkit. These include the most common coding environment in the world of low-level technology, the Arduino IDE and a regularly used programming language for artificial intelligence, Python. On top of that, the toolkit must be practical for students of broad university level studies. Examples include for instance Industrial Engineering and Creative technology at the University of Twente. Ultimately, it is required that it can also be engaged as a tool during a course about robotics or sensors.

Requirements	
Toolkit and toolkit users	- Easy to use - Clear overview of available building blocks - Enables rapid and cheap prototyping of minimal social robots - Only requires basic knowledge in Arduino and/ or python - Usable by students of broad studies - Usable as learning tool in robotic or sensor courses - Cover the fields of speech recognition, computer vision and visual display

Table 2.2: Requirements for the toolkit and its users

A distinction must be made here regarding the possible use of the toolkit in the project described by Suhaib earlier. Although, the format of this toolkit conforms to the needs of Suhaib and the toolkit could definitely be used in his context in the future. This bachelor's thesis does not take the toolkit far enough to be used effectively in Suhaib's case. Little modifications would need to be done to expand the use of this toolkit to a broader demographic, these are further explained in the recommendations, section 5.

2.3.3 Building blocks

Only with inter-compatible building blocks will the toolkit as a whole be successful. It is thus required that all building blocks can be used in combination with each other and that they can communicate with one or multiple micro-controllers. This is especially important as the micro-controller acts as the glue between each building block and manages the entire communication. Furthermore, each block should be fast and easy to use through the examples files included in the library. A time goal of 15 minutes is defined here, accounting for no previous knowledge about the building block. Lastly, a limit of 100 euros is set per component which would lead to a maximum of 300 euros for all three areas.

Requirements	
Qualitative	- Documentation available - Library with example files - Compatible with other building blocks - Compatible with Arduino like micro-controller - Supported by manufacturer until newest version is released
Quantitative	- cost <100€ - Setup time <15 mins

Table 2.3: Qualitative and quantitative requirements for the building blocks

Above the formerly stated requirements it would also be a nice addition if the building blocks had an active community. Although not a hard requirement, this would definitely be a decisive factor between two components in the same field of application. A Common development area between all blocks is also greatly appreciated and will be taken into account during the building block selection process.

3 Design

3.1 Building block comparison and selection

Grounded on the requirements for the building blocks, research can now be done in the state of the art sensors relating to speech recognition, computer vision and displays. The three tables in figures 3.1, 3.2 and 3.3 represent the collection of building blocks available in the toolkit for each corresponding field. Notice that the Google AIY vision and voice kit are still added to the toolkit. Despite not meeting the requirements set for the building blocks, they are still a very powerful asset and thus will be accepted. Now regarding the Lego mindstorms, they are considered to expensive to be part of the toolkit. One of these kits would already surpass the budget set for Kismet. The ReSpeaker here refers to the series of ReSpeaker cores, also available as an individual module.

The toolkit is made to make it easier to choose the correct building block based on the intended project. Even though many of these modules will become outdated at some point in the future as development in these areas continuous, the toolkit still provides a good first stepping stone for further in-depth research.

For the voice recognition modules (figure 3.1), all offline functionalities are limited to recognizing sentences /commands that must be defined beforehand. Some modules allow the user to define those, others are built-in and cannot be changed. Only modules that have an active wifi connection provides the option for a voice assistant such as the Google speech API.

	Grove	ReSpeaker	Audeme Movi	EasyVR Shield 3.0	Google AIY Voice kit
Price (€)	22.50	61.45 – 88.16	66.77	43.95	52.75
Max number of recognizable voice commands	22 (SI)	Not specified (SI)	200 (SI)	32 (SD) & 26 (SI)	Unlimited (SI)
Keyword detection/ Callsign	No/ Yes	Yes/ Yes	Yes/ Yes	Yes/ Yes	Yes/ Yes
Speaker connection	JST four-pins	3.5mm audio jack and JST 2-pins	3.5mm audio jack	3.5mm audio jack	No direct audio output
Microphone connection	No	Yes	Yes	No direct audio input	No direct audio input
Audio synthesiser	Yes	Yes	Yes	No	Yes
Can play sounds	Yes	Yes	Yes	Yes	Yes
Built-in speaker	No	Yes (1W output)	No	No	Yes
Built-in microphone	No	Yes (6 mic-array)	Yes	No	Yes
Expandable with shields	Yes	Yes	Yes	Yes	Yes
Programmable operating system	No	Yes	No	No	Yes
Coprocessor	No	ATMega32U4	No	No	No
Programming tool or language	Arduino IDE	Python and C/C++	Arduino IDE	Arduino IDE	Not specified
Online/ Offline speech recognition	No/ Yes	Yes/ Yes	No/ Yes	No/ Yes	Yes/ No
Languages	GB	Not defined (see Appendix)	US, GB, DE, ES, FR, IT	US, IT, JP, DE, ES, FR	Not defined (Google home)
Supply voltage (V)	3.6	5	7 to 16 (500mA)	3.3 to 5	USB powered

Figure 3.1: Speech recognition building block comparison (SI = Speaker Independant, SD = Speaker Dependant)

Not mentioned in figure 3.1 are two slightly different modules, the Geeetech and Elechouse V3 voice recognition modules. Both require the user to record voice commands instead of the usual written sentences for the other boards. Although they are not very suited for this project

due to a number of different reasons, they may be useful in different circumstances. For instance, as sound must be recorded, any noise or language can be used as command.

Some camera modules allow the consumer to modify the on-board computer vision algorithms to create custom applications. The number of modes hints to the amount of these on-board programs present on the unit. Figure 3.2 shows the camera module comparison.

	Pixy series	OpenMV series	Sony Spresense	Grove Serial camera	Google AIY vision kit
Price (€)	60.44	74.96	42.50	37.49	87.86
Number of modes/ programs	4	16	Not specified	Not specified	2 and more
Create custom application	No	Yes	Yes	Yes	Yes
Programming environment	Arduino IDE	OpenMV IDE (MicroPython)	Arduino IDE	Arduino IDE	Not specified
Interchangeable lens	No	Yes	No	Yes	Yes
Image filter option	No	Yes	Yes	No	Yes
Communication protocol	SPI/ I2C/ UART	SPI/ I2C/ UART	SPI/ I2C/ UART	UART (9600~115200)	SPI/ I2C/ UART
Highest resolution	1296x976	640x480	2608x1960	640x480	3280x2464
Online/ Offline	No/ Yes	No/ Yes	No/ Yes	No/ Yes	Yes/ Yes

Figure 3.2: Computer vision building block comparison

Essentially there are 2 categories for the speech recognition modules and the cameras. The first category is for rather simple projects that require relatively simple features. The second one is for the projects that are more extensive, requiring a chat bot (cognitive responses) or extensive computational power.

The last table in figure 3.3 presents a collection of display units. As a wide variety of screens exist in terms of sizes alone, also the size is specified.

	LCD (16x2)	Adafruit Animat (2x1.54")	Dot Matrix (4*8x8)	OLED (64x48 pix)	NeoPixel (16x 5050) (RGB ring)	Touchscreen (2.4")	Epaper (2.13") (monochrome)
Price (€)	8.12	44.47	6.32	14.20	8.86	24.14	15.38
Number of custom characters	8	Not specified	Unlimited	Not specified	n/a	Unlimited	Not specified
Arduino compatible	Yes	No	Yes	Yes	Yes	Yes	Yes
Shield form	Yes	Yes	Yes	Yes	No	No	No
Flexible	No	No	No	No	No	No	Yes
Touch screen	No	No	No	No	No	Yes	No
Number of pins needed	11	11	5	7-8	3	4	13
Supply voltage (V)	3.3 to 5	3.3 to 5	5	3.3	4-7	5	3.3

Figure 3.3: Visual display building block comparison

It is important to know that the prices listed in all previous tables are for indicative purposes only as they may vary with time and seller, dollars to euros are converted with 0.89 ratio. Because not every component included in this toolkit was available at one retailer, the list of visited websites can be found in Appendix A.1. Care has been taken that each seller conformed to the University's quality standards.

Additionally, one could argue that a computer with the right software would be capable of performing the same tasks performed by the 3 types of building blocks as good or even better. This

might be true but the same can be said for the Raspberry Pi, which boils down to essentially the same problem. Both could indeed achieve the same tasks better on their own without the need of three or more smart sensors. However the benefit of using this table with the low-level building blocks lays in the possibility of disassembling, assembling, combining and replacing all building blocks by others. It also makes the prototype repeatable by others due to the many pre-programmed algorithms and wide variety of libraries. Being able to choose the right building block depending on the need of the intended project can save money and prevent unnecessary over complication.

Continuing the analysis of the toolkit's building blocks. In Appendix A.2 a more in-depth research is done in the strengths and weaknesses for modules relevant to the project. If using this toolkit to find the appropriate building blocks for a minimal social robot, it is advised to first consult the aforementioned Appendix page.

Finally, the correct building blocks can now be selected for the Kismet prototype.

(1) For speech recognition the Audeme Movi module is chosen. Considering Kismet should be kept simple and should consist of only a micro-controller with smart sensors, the Google AIY and the ReSpeaker are discarded. The Grove module on the other hand is too minimal since it does not provide the ability for custom commands. The EasyVR 3.0 could very well have worked, however compared to the Audeme it was slightly more complicated to acquire a higher number of voice commands along with less features such as speech synthesizer or voice options.
(2) For computer vision the OpenMV M7 camera is chosen. At this time this is the newest version released by OpenMV but the H7 version should be available soon. The main issue with the popular Pixy2 camera it that it is limited to color detection, line tracking and pan/tilt programs. Usually social robots require features like face recognition, smile and salience detection or gesture recognition. The Google vision is eliminated for the same reasons as the Google voice kit. Sony's camera, the Sony Spresense together with the Grove Serial camera do not provide enough support regarding computer vision algorithms needed for Kismet. Therefore, the OpenMV M7 is the best option for this prototype. It not only allows for the same features as the Pixy2 does, integrated with a Pixy emulator, it also provides many of the sought after computer vision algorithms. These will further be discussed in subsection 3.2.3.
(3) For the displays serving as platform to show the robot's emotion, the 8x8 DOT matrix is selected. Simply due to the smaller size, the stack ability of the modules, low price and overall creative possibilities.

3.2 Component testing

This section covers the individual building block testing previously selected for the Kismet application. While an 8X8 DOT matrix display was picked, here a 16x2 LCD will be tested instead. The matrix display was not available yet and hence could not be tested. This should however not be a problem considering that the program that will be written as test purpose already exists for the DOT matrix within the research chair.

All scripts except for the OpenMV M7 camera which is written in the corresponding MicroPython IDE, are written using the Arduino IDE. The eye gazing on the LCD screen is tested using Tinkercad circuit designer (Autodesk (2010)).

3.2.1 Audeme MOVI

After studying of the basic library examples, subsequently working the ladder up to more complex example scripts, 3 different programs have been attempted. The first one consists of the default Eliza file available within the library. The second script is a derivation from the first one and tries to seek a way of learning Eliza to understand partial unknown sentences. For read-

ability these sections will be kept as short as possible. To get a full grasp on the workings of each part, a look should be taken in the Arduino scripts themselves.

Before explaining how each script works it is useful to know what exactly Eliza is. Natural language processing computer programs have existed for many years now. One of the first being Eliza, created between 1964 and 1966 at MIT by Joseph Weizenbaum (Ireland (2012)). The program simulates a conversation using pattern matching and substitution methods (Norvig (1992)). For instance, Eliza looks for certain keywords which it then uses again in her questions. This way the user is deceived into thinking the computer understood what was said. Moreover, when the machine did not find a pattern, it answered with a general response.

Looking at the Eliza example script that came with the Arduino MOVI library, it is clear that it is not possible to recreate the same Eliza developed in the 1960's. Problem arises simply due to the fact that speech recognition modules like MOVI must learn a set of sentences/ voice commands before being able to recognize them. As such techniques employed in the real Eliza cannot be applied as she is not able to understand everything the user says. This does not have to be a problem if everything would be transformed from speech-to-text, regardless if it was understood or not. Sadly, this is also not the case, MOVI only correctly transforms speech to text if it recognized the sentence. Bear in mind that this applies to every speech recognition module that is offline. Eliza's current architecture as presented in the MOVI library can be seen in figure 3.4.

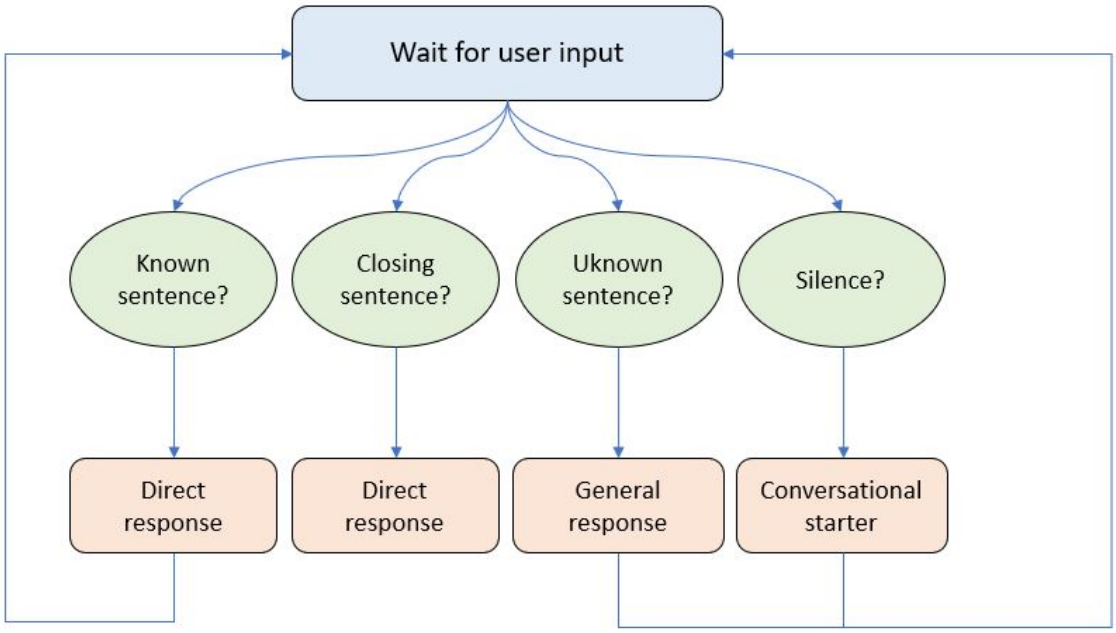

Figure 3.4: Eliza example script architecture

For each sentence learned by Eliza, a corresponding direct response is attached. If she did not understand what the user said (this would be the case if the person said a sentence that Eliza did not learn), she responds with a general answer and waits for a new input. Conversational starters work the same way but only trigger if a certain time threshold has been passed without voice input. The biggest issue with this program is that it is scripted, orchestrated by the developer. If it was possible to make the program less predictable and give more freedom to the user, only then it would become interesting for HRI use. This is exactly what is attempted in the next phase. Achieved by learning Eliza chunks of sentences and subsequently using these to construct phrases. The new architecture is shown in figure 3.5.

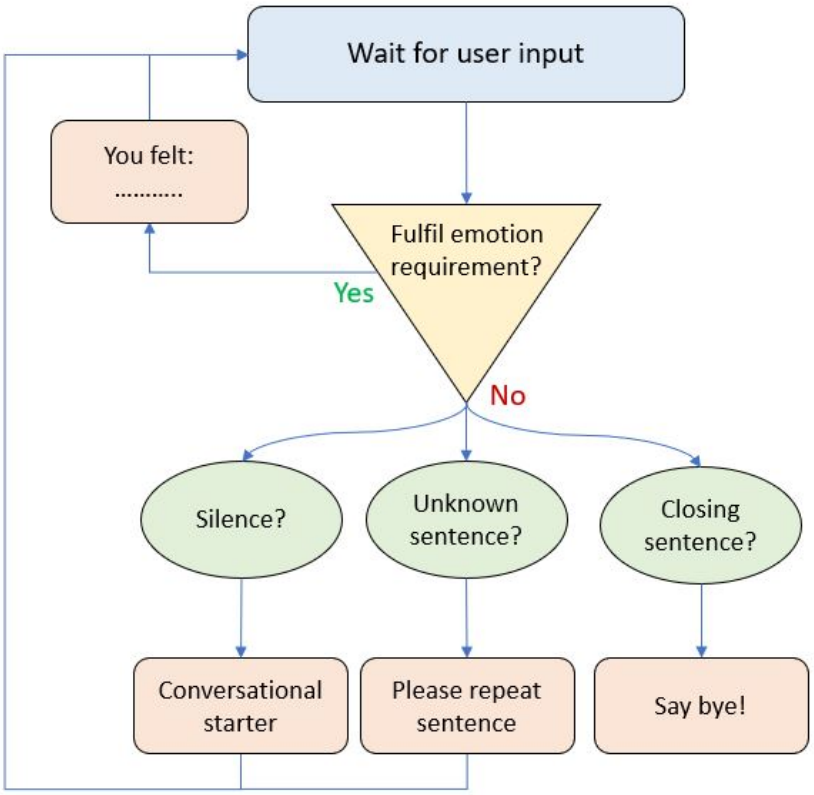

Figure 3.5: New Eliza program architecture

At the start of the program MOVI learns sentence segments comparable to "I feel" or "I am" and a set of 36 different emotions and feelings. The sentence segments correspond to the most likely way the user will start his or her phrase when expressing emotions verbally. Next, instead of using the internal sentence matching which was used previously, the raw input is used. As long as the sentence spoken by the subject corresponds to a combination of words learned by MOVI, it will be capable of understanding it allowing for further processing. Next, requirements are set to filter out false-positives. Especially to distinguish between the case where the user correctly expressed an emotion and when MOVI just misunderstood the spoken sentence. For the program to accept an emotion, referring to the "Yes" path in figure 3.5, the recognized sentence must fulfil one of the following requirements:

- Emotion + sentence length of 1 word
- Emotion + "FEEL"
- Emotion + "AM"

The trigger criteria above are not very elaborate but seem to already filter out most of the false-positive. Eliza also shows to be a great way to test Audeme's speech recognition software. During the many testing hours, it was discovered that specific words are more often misheard or swapped. These mostly included those that have similar pronunciations but also some where MOVI just seems to have more difficulty recognizing. In the same manner, the recognition software could perceive other words with minimum effort. As a result the call sign was also changed from Eliza to Arduino. Eliza had more trouble with the word "Eliza" while "Arduino" was no problem. Interestingly, MOVI also spelled out certain words. The reason is thought to be due to the internal dictionary in MOVI's SD card. Yet, it is still unknown how broad the dic-

tionary exactly is. Of course there is still a lot to be improved in this test program (which is also done for Kismet) but it is sufficient for the testing phase .

3.2.2 LCD 16x2

In social robotics, displays are mostly used to convey emotions by means of facial expressions. The 16x2 LCD used in this report is not big enough however to illustrate both the eyes and the mouth therefore, only the eyes are shown on the LCD. Furthermore, since the script is merely an example script designed as basis for future projects and as learning purpose, only the gazing of the eyes is implemented. Also as a result of the maximum of 8 custom characters available with LCDs. The simple program switches between three different eye images to create the iris animation from central position to the left or right side of the eye, see figure 3.6.

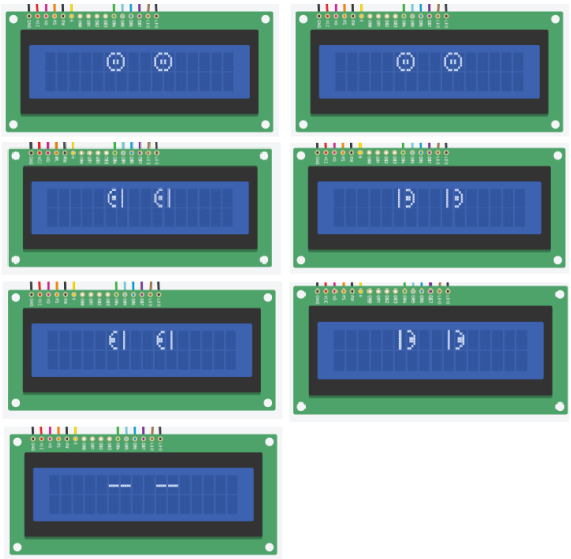

Figure 3.6: Eye animation frames on a 16x2 virtual LCD

Moreover, the speed of the animation and the target position of the iris can be determined by the programmer. The target position refers here to the final destination of the iris which can be middle, left/ right or full left/ full right as well as an option to loop. These can again be seen in figure 3.6. The loop options here means the iris will not stop at any position but instead keep looping through the set of animations. At last, one can define the direction of movement of the iris, either left or right. Lastly, to complete the gazing animation, the direction, speed and target are set to random and blinking is added at an arbitrary interval. The target position of the iris is randomized with twice the chance to fall on the center position if not already there. This is done so that the eyes are not always moving around.

The described program sets a basis for eye animation using a 16x2 LCD. The eyes are placed on the display in a way that the script can also be used for smaller boards like the 8x2 LCD. Note that all eight available custom characters have already been used here, therefore if wished to proceed by elaborating the script for emotion display, one would first need to simply the animation from three steps to two steps. Alternatively, a collection of ASCII characters could be used instead of custom characters or different custom characters must be created that can further be reused in different eye images.

3.2.3 OpenMV M7

The OpenMV M7 camera provides a wide range of examples programs, hence the more difficult aspects are already tackled. Nevertheless, multiple crucial elements must still be tested. Com-

bining face detection with servo motors to create a face tracking pan/ tilt device is the initial goal. The second goal is to send the servo control values to the Arduino to test the camera's communication skills.

Before starting the pan/ tilt program, the face detection script is tested. OpenMV has two different programs capable of finding and tracking faces. The first one is the face detection using Haar cascades, an approach that is common in machine learning (Wilson and Fernandez (2006)). Next, an image is taken and processed to find the most probable location of the face. A box is then drawn at these coordinates delimiting the face (see figure 3.7). The size of the box can also be used to determine the proximity of the individual. Conversely, the second program also makes use of the Haar cascade in the same fashion but rather then drawing the face outline it looks for keypoints within. These keypoints are defined at the start of the program, where the camera will take a picture of the first face it finds. With the specific keypoints now saved for that face, it will compare keypoints from subsequent faces with those saved in the beginning. If enough keypoints match, the face is recognized. The first algorithm works for any face while the second specifically recognizes one face. Both work well in practice. Figure 3.7 shows face detection with the OpenMV M7.

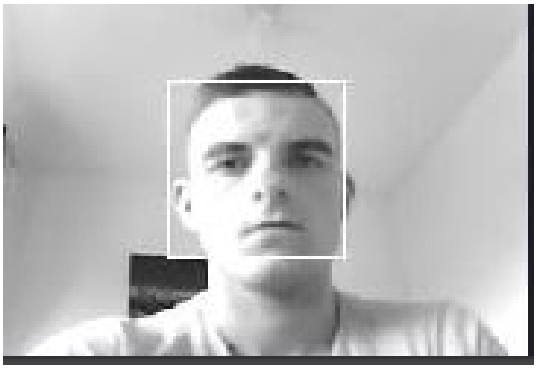

Figure 3.7: Face detection with the OpenMV M7, screenshot taken for the OpenMV IDE

Pan/ tilt is achieved similarly for the two. Once the face is found, the location is used after which the error is calculated according to a set point, usually the middle of the image. The error is then summed or subtracted from the current servo value and multiplied by a weight factor. These weight factors must be tuned appropriately. Tuning does not require a difficult process and instead can be done by means of trial and error. The weight factors affect the smoothness of the servo movements.

At last, supporting SPI, I2C and UART, all three are tested to see which one works best. Communication is performed with an Arduino Uno. At first, SPI and I2C perform well when sending simple messages. However, whilst combined with face detection or recognition, problem occurs with the camera's frame rate. The frame rate drop indicates that the main loop is running slower. Although a solution was never found, it became clear later that both are also less supported by the camera. Confirmed by OpenMV's documentation, who advises the use of serial transmission for communication with other devices, apparently the camera can have difficulties synchronizing the connection correctly. Succeeding the UART test, it is indeed found to be the case and servo values were successfully send to the Arduino in real-time. Multiple limitations were noticed though during operation. After launching the face detection program for the first time, one notices directly that the angle of view is very small. Also a digital zoom is available and sometimes applied by default. Reducing this zoom already provides an increased angle of view. Furthermore, care must be taken that the lens is correctly focused.

3.3 Kismet

Proceeding from the past chapters, a minimal social robot can now be designed. The next section will first give an overview of the features included in Kismet as a result of the design process. Afterwards, the hardware and software design is explained and justified.

3.3.1 Overview of Kismet's capabilities

Kismet is divided into three separate operation modes. For each mode Kismet is able to perform different tasks as will be explained next. A good overview of Kismet's total architecture is seen later in subsection 3.3.3, figure 3.10. The underlying dialogue software is called Eliza and learns to recognize different sentences according to the current operating mode. Together with Eliza, a MIDI panel is also used to quickly switch operation modes.
The default mode when the power is first turned on is the so called therapist mode. In here, Kismet presents the following features.

Therapist mode features:

- Face detection (1 or more faces)
- Pan/ Tilt face following
- Eliza: Therapist mode
- Volume and sensitivity control
- Gazing
- Blinking
- Breathing motion
- Next mode option

The Eliza therapist mode allows the user to express its feeling or emotions. If successfully understood by the speech recognition software, Eliza expresses empathy by repeating how the user feels and adopt the corresponding emotional behaviour. Split into 6 distinct emotions (happy, surprised, confused, angry, sad, suspicious, sleepy and neutral), she understands up to 34 different interpretations of feelings and emotions. The complete list of words can be found in Appendix A.3. Each category of emotion affects the breathing amplitude and frequency as well the eyes of Kismet. The next mode option simply allows the user to go to the second operation mode described next, either through the use of the MIDI panel or by asking Eliza. Note that Eliza will always ask for confirmation first.

When in the second mode, Eliza adopts a more passive behaviour formed on the servo recorder program written by Dertien (2019). The program gives the user full control over the pan-tilt system and Kismet's emotions. Sequences of movements can also be recorded for each servo and subsequently played back. The mode can be considered a sandbox mode giving the user freedom of creativity.

Sandbox

- Full pan-tilt control
- Kismet emotion control
- Movement recording
- Eliza dialogue manager

- Volume and sensitivity control
- Gazing
- Blinking
- Breathing motion
- Next mode option

The third and last mode serves as a place-holder showing off the possibility to add even more modes. For now it only allows the user to go back to the second mode. Here of course, Eliza together with the autonomous behaviour of gazing, blinking and breathing still function on the background.

3.3.2 Hardware

Kismet's hardware mainly consists of an assemblage of smart sensors, displays and an Arduino Mega ADK, with the complete list in found in the bill of materials Appendix A.4. Here the spatial configuration is shown together the connections between the building blocks.

First, Audeme's software serial communication was changed to hardware serial, freeing digital pins 10 and 11 of the Arduino UNO board whilst providing slightly faster speed and flexibility. Due to the extra serial port, the Arduino Uno does not suffice any more. As a consequence, the micro-controller must be upgraded to an Arduino Mega ADK. This board provides three additional serial ports satisfying the OpenMV M7 and the Audeme MOVI speech recognition module.

Second, earlier was mentioned that the OpenMV M7 had a small angle of view. Based on this reason, it is chosen to opt for a pan-tilt configuration were the camera is mounted on top of the moving head. The camera is thus always facing the user and therefore the face following is capable of operating on the entire range of operation allowed by the servos. Instead one could also have opted to place the camera in a fixed position while only moving Kismet's head. This however gives major drawbacks and very little advantages with the current M7 model used. The OpenMV M7 does not provide wide view lenses which means the user would need to be in the small angle of view for the program to work. Second, the pan-tilt system would then only perform little movements because as soon as the user exits the field of view of the camera, no servo instructions are send any more. Recommendations are given at the end of the report regarding this topic.

Finally, a casing is made to house all the components. The casing should be easy and fast to put together while being durable enough to last the testing phase of the robot. Just like the cardboard boxes of the Google AIY kits, the casing is not meant for permanent use but only serves as a temporarily housing during the initial testing phase. Consequently, a Solidwork design together with wood laser cutting techniques are chosen to create the casing. Figure 3.8 shows the current prototype setup.

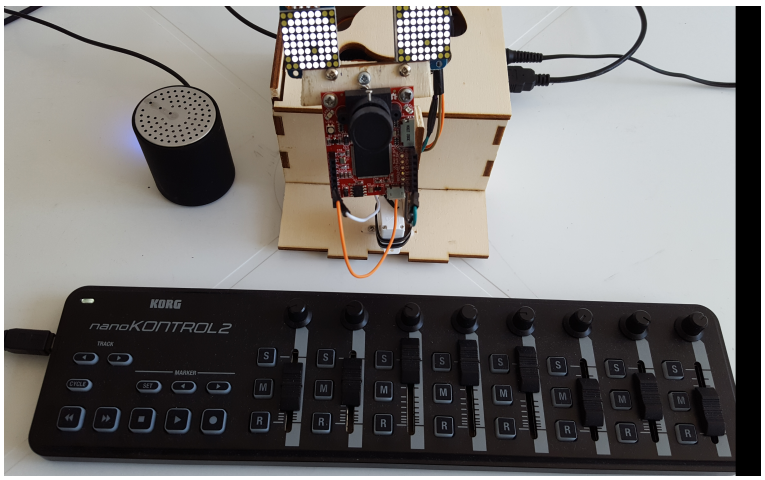

Figure 3.8: Kismet picture

Before proceeding to the internal connections between the components, one should know that when more than 12 servo motors are being used at the same time, PWM may be disabled on certain pins depending on the chosen micro-controller.

Building blocks are selected and tested but without the proper relations linking them, Kismet cannot function properly. Hence, the flow diagram (figure 3.9) shows the different physical connections between components.

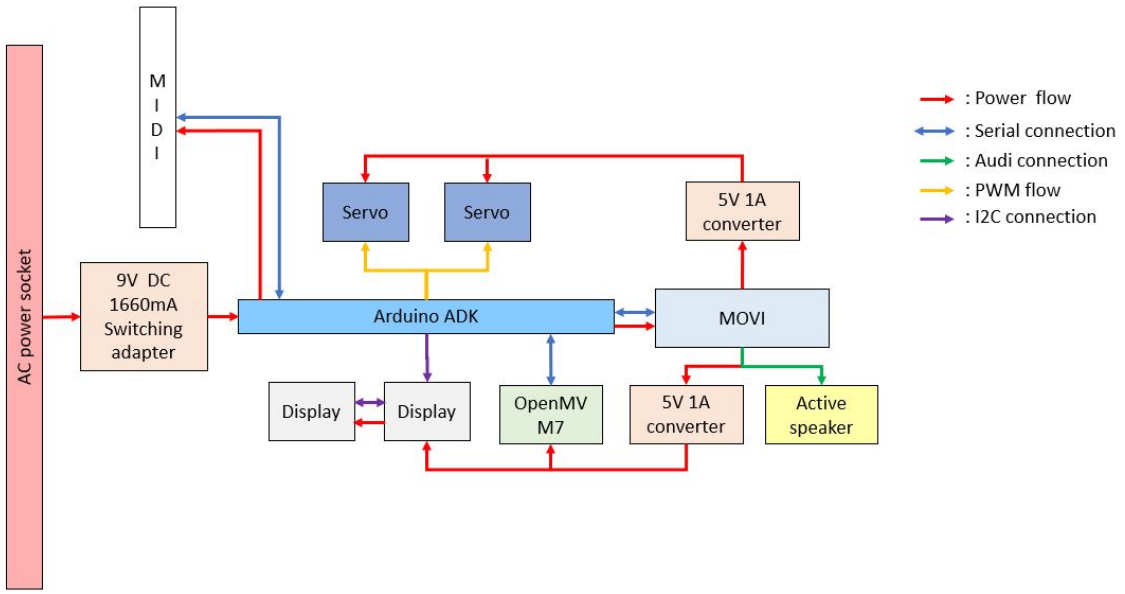

Figure 3.9: Flow diagram showing building block physical connections

The speech recognition module MOVI cannot operate on the power delivered by the USB attached to the Arduino, hence it requires a separate power supply between 7 to 16 volts. A 9V DC switching adapter suffices to provide enough power for the whole circuit. The MOVI shield does not have a power port but sits on top of the Arduino board. Both share a Vin pin and thus power can further be distributed from the MOVI shield. Nonetheless, two 5V 1A converters are needed to transform power to safe levels for the sensors and servo motors. Remark that the servo motors are arranged in a separate power line. Quick changes in the servo positions can lead to high current peaks which can damage the board. A high value capacitor (1000 μF) is

therefore attached in parallel after the converter. Lastly, it is important that common ground is used for all modules otherwise serial data may wrongly be read. This was experienced with the OpenMV M7. Besides an active speaker is connected so as prescribed by MOVI.

3.3.3 Software

The software part of Kismet is explained in steps. First, the general architecture is studied to get a good overview of how the software is designed. Followed next by details in each dominant section in the structure.

Architecture

With the knowledge of the previous sections, it becomes progressively more apparent how Kismet exactly functions. In figure 3.10, one can recognize the three different operation modes again. The diagram illustrates which blocks are active in which operation mode. Additionally, the arrows represent the type of data that is shared between each block. Information stream refers to a wide array of data material like images, voice input or button pressing while control stream relates to internal data exchange resulting in a physical change. Included are for instance servo PWM values, salience and valence values and volume levels. Take into account that the arrows do not represent a direct physical link but rather express the notion of who controls who. For the physical connections refer back to figure 3.9.

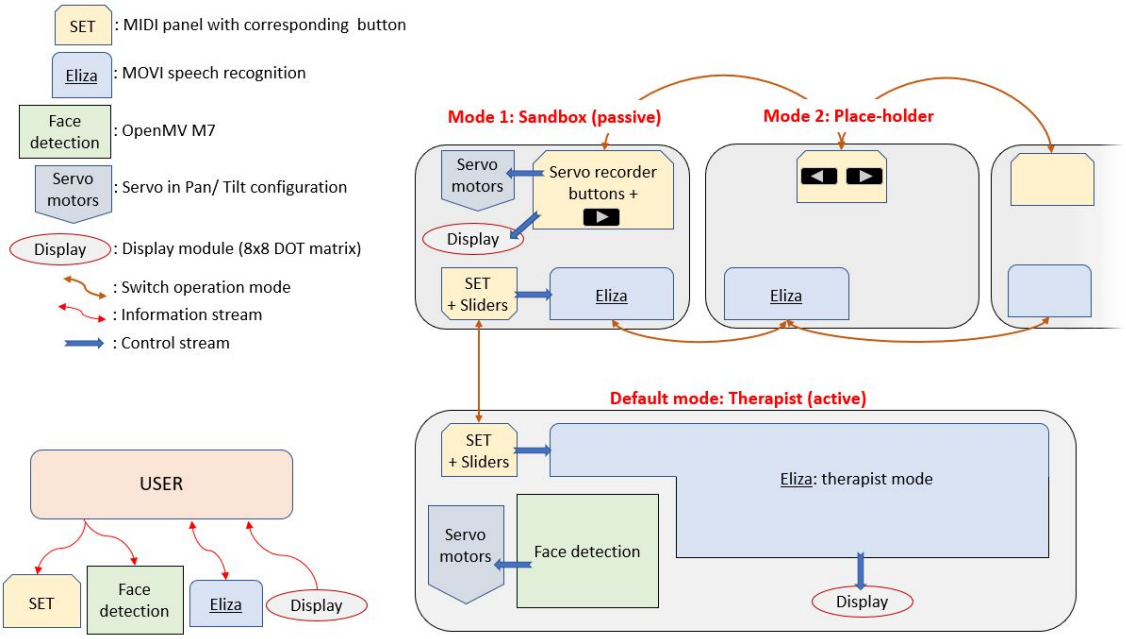

Figure 3.10: Representation of Kismet's software architecture

Here the third box which is half present on the figure repeatedly hints to the opportunity to expand the current number of modes.

Autonomous behaviour is introduced in all parts of the software. Blinking and gazing is dealt entirely in the eye animation subsection 3.3.3. Breathing on the other hand is handled in the Arduino IDE for all cases except when the face following is active where it is instead done in the OpenMV IDE. Face following dictates the servo movement and thus the breathing motion must be mixed in for a smooth behaviour, later covered in subsection 3.3.3. For the more trivial case where the breathing movement is the only motion performed by the servos, a simple sine wave with varying frequency and amplitude is applied to obtain the desired result.

Eliza

Described next is the basic operation process of Eliza. The description mostly suggest to Eliza in "Therapist mode" as it is by far the most elaborate part of Eliza. Eliza in mode 1 and on serve principally as control tool to switch between the different modes. The reason is that for this project Eliza did not need a more complicated role in these sections.

Based on the second Eliza program developed in subsection 3.2.1, an improved version is made in order to be integrated in Kismet. The new Eliza retained the same basic principle of constructing phrases from sentence segments. However, several parts have been improved. The software architecture remains identical but the filter criteria for the "Yes" path in figure 3.5 are slightly expanded. Moreover, a negation option is added, allowing the user to add the word "not" in the emotional expressive sentence. On top of the pre-mentioned requirements the following are thus also included:

- Emotion + "not"
- Emotion + "think"

Here a different word like "do not" could be added or substituted instead of "not" of course. Furthermore, a custom sentence matching algorithm is adopted founded on the Levenshtein distance algorithm (Babar (2018)). Originally employed in an Audeme example file, it is here used among more to check for silences and unknown sentences. Using several distinct sets of sentences, MOVI can switch which sentences are utilized within the sentence matching algorithm depending on the expected user answer. In the case the user would like to leave the "Therapist mode". Once activated, Eliza will ask for confirmation. Here the answers "Yes", "No" or "Maybe" are expected. If a different answer is given, Eliza will ask to repeat. Note that the algorithm will thus only check the input sentence against the three previously mentioned words. In addition, a matching thresholds needs to be specified meaning that it is also possible to use the algorithm to correct misinterpreted words. By forcing a match between the control sentences and the input, "once" could be corrected to "one" for instance. The "maybe" answer leads to Eliza choosing for the user through semi-random number generation.

Concluding this part, depending on the emotion expressed, Eliza attributes new valence and arousal values resulting in a change in Kismet's emotion.

Face detection

Compared to subsection 3.2.3, minimal changes have been performed regarding the pan-tilt functionality. Useful data is however sent between the camera and the Arduino Mega. The OpenMV sends information regarding the number of faces detected and the servo values vital for face following. Inside the Arduino these can simply be read and sections of code can be executed depending on the number of faces recognized.

Vice versa, the Arduino sends the desired frequency and amplitude for the sine breathing motion depending on the present emotion. Contrarily, to the Arduino board, serial data cannot be directly read in OpenMV. Therefore, inspired by OpenMV's Pixy-emulator example script, a state machine is developed to accurately and reliably read the incoming data stream. The state machine ensures data is parsed in the correct order of transmission.

HMMM: breathing mixing

Until now no behavioural mixing was necessary. Yet when combining face following with autonomous behaviour, conflict occurs. The main problem manifests itself in how the program works. If one would to add an offset in the form of a sine wave on top of the tilt PWM values, the face following algorithm would try and counter-act the former as an artificial error is introduced. Instead, the target location of the user's face on the camera image is moved to resemble

a sine function. In other words, a small up and down offset is introduced to the set-point position for the center of the square box in figure 3.7.

Emotion mapping

Emotion mapping is performed comparable to the one in the EyePi (section 2.1). However, 2 additional emotions are incorporated since eye animations were available. The valence and arousal graph implemented now in Kismet is shown by figure 3.11

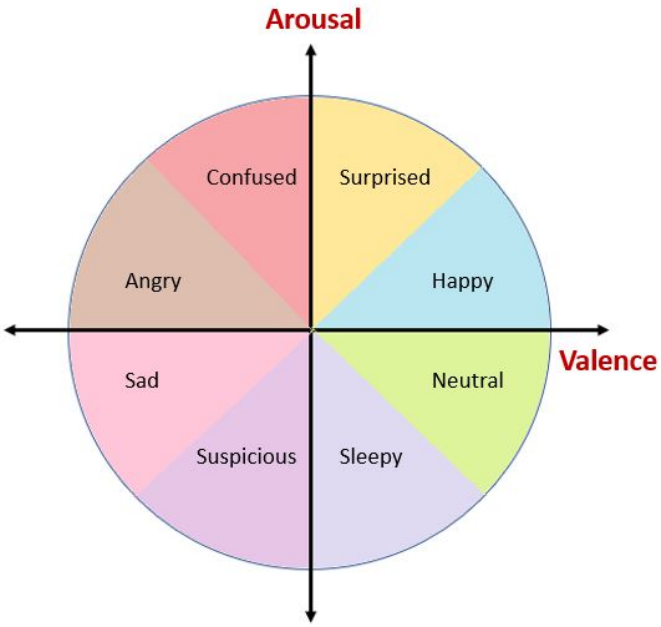

Figure 3.11: Valence and arousal emotion mapping implemented in Kismet

Lastly, the reader should know that in Kismet's current form, the sleepy eyes are represented by two love eyes functioning as place-holder. This may seem peculiar but there was simply not enough time in the project to create animations for the sleepy emotion. Also considering the goal of the report, priorities had to be set.

Eyes

Next, eye animations for the various emotions come from previous work of Kasper de Kruif (de Kruif (2018)). In his bachelor's thesis animations were created for love, thinking eyes and all emotions shown above except for the sleepy one. Blinking was also already present. Nevertheless, to enable gazing the iris is removed on each animation frame. Partly following an Arduino example file for the 8x8 DOT matrix library, the iris is reinstated later by turning off 4 LEDs. The position of this little square can be shifted and randomized, causing a gaze effect with variable speed.

Servo recorder

Finally, the last part of the prototype's software is the servo recording. Minor changes have been applied to the initial program found in Dertien (2019). In essence, the execution remains identical, only the play back for individual recordings is modified to stop at the exact amount of steps heretofore recorded.

4 Results

This chapter deals with the results obtained after testing Kismet, the toolkit through the creation of Kismet and the composing building blocks on an individual level. First, Kismet's performance is evaluated and compared to the requirements. Subsequently, the toolkit is validated and critically discussed. Finally, the three building blocks originating each from a different area of expertise are also compared to the initial requirements set in section 2.3.

4.1 Kismet evaluation

Firstly, Kismet is indeed only constituted of smart sensors and a micro-controller. It can also successfully reflect user emotions, blink and gaze, seen in figure 4.1.

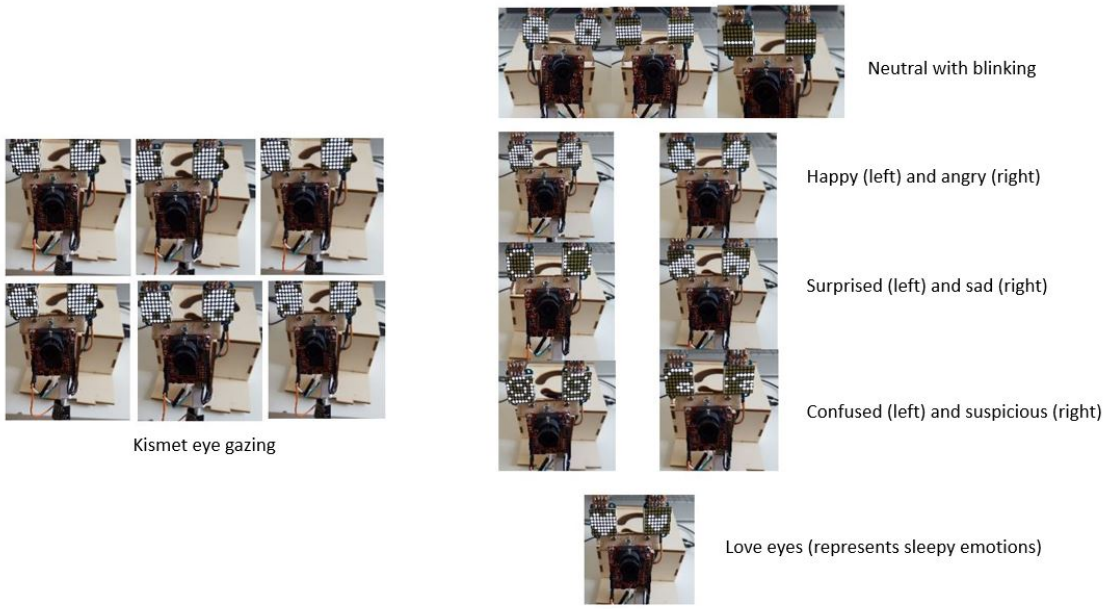

Figure 4.1: Kismet gazing (left) and Kismet emotions and blinking (right)

Only soldering for the DC-DC converters is needed and very little for the two displays. It can be said that the minimal soldering requirement is thus likewise validated. Verbal communication is also satisfied with the Audeme Speech recognition together with the offline criteria because no wifi connection is required during Kismet's operation. Breathing simulation works but clear differences in amplitude and frequency are not very noticeable. Although sufficient for this project, the motion is not very smooth either. On figure 4.2, the left picture is the lowest and the right the highest. Comparing them, it can be noticed that the left shoulder is visible on the left but not on the right. The fact that breathing functions on the OpenMV during face following simultaneously validates the behaviour mixing.

Figure 4.2: Series of openmv pictures for breathing

Testing for the breathing motion was done in a room during the day next to a window providing enough natural light and with a threshold value of 0.75 and scale factor of 1.25. The user was about 40 cm away form the camera, sitting in the exact same position for both.

Nevertheless, multi modal mixing was not added for the breathing motion and the servo recording. Primary reason for this lack is the finite time available for the project and secondly the fact that the servo recording is plainly less interesting for the Kismet's purpose. Although a nice addition, focus was more laid on the active side and less on the already developed passive.

Regarding fluent conversation, the prototype still waits for user input in most cases resulting in what is called turn taking. However, when input takes to long, Kismet is able to notice this and act upon it. Effectively animating the conversation, therefore the fluent conversation is satisfied but still considered minimum.

Secondly, all requested quantitative requirements are also met. Kismet can recognise and display 8 different emotions, detect 2 or more faces while following one with a pan-tilt setup, both shown in figure 4.3.

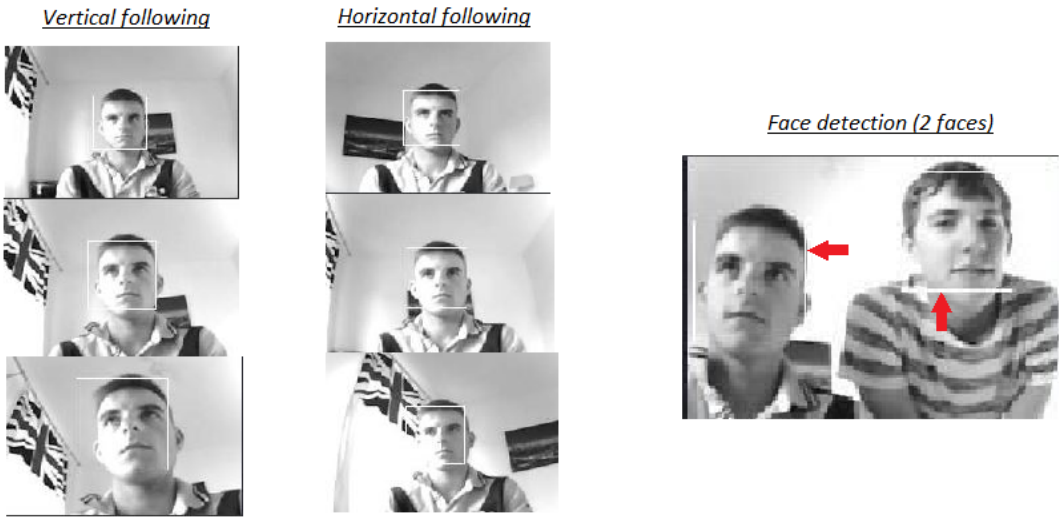

Figure 4.3: Face following (left) and 2 faces detection (right)

The same test conditions as for figure 4.2 have been used. The result shows that the camera recognizes both faces and marks there location with a rectangular box. Red arrows have been added in post-processing to highlight these boxes. Same conditions were used for the face following. In figure 4.3, it can be seen that the closer the user gets to the camera, the more the camera tilts up. Same for the horizontal movement.

Audio response is determined by measuring the time between when the user finished speaking and Audeme answers. The measurement setup was a small room, no background noise, using the on-board microphone while the Audeme sits in the wooden casing. Repeated 6 times, the experiment resulted in an average response time of 1.69 seconds. Thus also complying within the required 1 to 3 second range for fluency. Ultimately, the total hardware cost settles at €247.87, with the detailed budgeting in Appendix A.4, satisfying the last requirement.

Coming in last, a brief comparison is made of the capability results between the low-level approach of Kismet and the High-level approach of the EyePi. The most noticeable difference lays in the difference in physical appearance. Kismet is still in the prototyping phase and does not have a sturdy durable look. Furthermore, camera placement is also non-identical. According to Bob (van de Vijver (2016)), the reason for choosing a fixed camera approach for the EyePi was to simplify image processing and acquire a wide field of view. With Kismet, wide-angle lenses were not available leading to the current design choice. Drawbacks are that more cables do run to the head on this configuration, which makes it less desirable. Kismet also does not provide HMMM for the servo recorder and the breathing motion yet as opposed to the EyePi. Dead time is completely removed for the high-level robot while for Kismet it is still present in the case of horizontal movement. This leads in to another distinction, salience detection. Although never a goal for this initial stage of development, it does add substantial improvements to the robots lifelike behaviour. At last, kismet does cost less, around 250 euros compared to 900 euros, is less complicated to create and manipulate along with being able to be assembled and dissembled at will. Kismet also brings voice interaction and emotion recognition, features that the EyePi does not present yet.

4.2 Toolkit validation

The toolkit gives a clear overview of the available building blocks and enables rapid and cheap prototyping of social robots demonstrated in this report by Kismet. Moreover, the found building blocks do cover the fields of speech recognition, computer vision and visual display. Additionally, they provide plenty of examples and help through libraries and thus only require minimal knowledge about Arduino IDE and OpenMV IDE.

For the remaining requirements however, these could not be tested thoroughly. The three requested together to be usable by students of broad studies, as learning tool during a course in robotics or sensors and easy to use. To test these a proper testing environment should be constructed or a case study could be build. Once the toolkit experienced slight modifications to make it suitable for user with even less experience and knowledge about Arduino and programming in general, only then could it be tested in environments as described by Suhaib (Aslam (2018)). If these requirements are essential for the project, one should first design a test setup to accurately assess them.

4.3 Building block evaluation

Despite not all building blocks compared in section 3.1 being able to be tested, it was found during selection that all provided sufficient documentation and libraries, which can easily be verified. Also each building block remains supported and available by the supplier as much as this can be checked at least. Kismet proved that the three inspected modules were intercompatible and Arduino compatible as an Arduino Mega ADK was the only micro-controller used.

In concerns of the defined quantitative criteria, all cost were indeed limited to less than 100 euros proven by the building block comparison tables in figures 3.1, 3.2 and 3.3. At last, a setup time less than 15 minutes was prescribed. The former is reviewed and validated for the three chosen building blocks and the 8x8 DOT matrices. Testing can only be done once per person as it needs the user to be unfamiliar with the building block. For this reason, each block could

only be tested one time. The experiment includes downloading the relative library, setting up the connections for the module and loading a first example script. For all four, timing results laid below the 10 minute mark. But again for a more accurate assessment, one should approve this with independent users on a larger scale as described earlier and repeat the experiment multiple times.

5 Conclusion and recommendations

5.1 Conclusion

The goal of this bachelor's thesis was to create a toolkit to prototype simple social robots using smart sensors and a simple micro-controller. The report promised a collection of building blocks in the fields of speech recognition, computer vision and visual display. Furthermore, the design and creation of a simple social robot using the selected building blocks was set to be achieved. Additional requirements of autonomous behaviour and behavioural mixing were also prescribed on the robot.

In conclusion it is possible to create a prototype of a minimal social robot using low-level building blocks and a micro-controller. The robot in the end correctly fulfilled the requirements defined in section 2.3 and successfully tested the toolkit. Regarding the toolkit, looking at the created prototype, one can conclude that the toolkit contains for now enough tools to create a simple social robot. Guaranteeing speech recognition, face detection and following as well as emotion display with HMMM for the breathing motion.

The robot is however limited by the capabilities of its building blocks, as such offline speech recognition limits the user's free speech. It can be drawn that the social robot remains very minimal in its capabilities and is still in its early phase of development. The design approach shows that it results in lower performance quality and capabilities limited to what the smart sensors can do, compared to the EyePi. Because the pan-tilt system showed to be the weakest point in the structure, the prototype is not yet ready for larger scale testing until this is resolved. For the toolkit, ease of use, student usability and learning tool usability still need to be tested from which it can be concluded that the toolkit is partly satisfied but needs to be modified or additionally tested for use in specific conditions. Behavioural mixing was also only applied with the camera and not with the servo recording.

5.2 Critical reflection

First, it is important to state that qualitative evaluation is performed by only one person in this report, which also happens to be the creator. Even though care has been taken to adopt a critical attitude during testing and evaluation, a better assessment should be performed for more accurate results. This improved evaluation should at least involve multiple independent individuals fulfilling the user requirements. Likewise, involvement in a robotic or sensor course could be a great way to test Kismet's performance and overall user perception together with the toolkit's performance and ease of use.

In Kismet's evaluation, the principal weakness regarding the building block configuration is the pan-tilt design. Servo motors are now directly attached the wood casing resulting in unreliability. A better approach would have been to buy off the shelf durable plastic pan-tilt setups for the selected motors. the camera attached to the pan-tilt system combined with the use of small servo motors results in less fluent movement.

5.3 Recommendations

Along the report multiple opportunities for possible improvements and future work have been mentioned.

First, dead time could further be removed by implementing autonomous behaviour for the horizontal servo motor. Currently, when Kismet does not detect a face, the head remains fixed in the same horizontal position. Instead, the robot could look around and adopt a bored behaviour. This leads to the second improvement, salience detection. Although with the OpenMV M7 no salience detection program was found, rapid sudden movement could be identified us-

ing frame differencing. Frame differencing compares two images and looks at the differences to spot movement. Problems could occur here with the ongoing camera configuration, therefore a good option could be to, like the EyePi, move the camera into a fixed position. Although not efficient with the OpenMV M7 used in Kismet, the new version OpenMV H7 out now provides interchangeable lenses, namely also wide-angle lenses. Care must be taken that the despite the lens, face detection still works as expected.

Next, to improve the hardware, off the shelf pan-tilt system could be bought. Pan-tilt brackets would drastically improve Kismet's appearance together with it's reliability and sturdiness. One can find those for less than 10 euros on websites such as Antratek.

Furthermore, a safeguard against sudden servo movements can be added. For instance, when the power is turned on, Kismet will reinitialise the servo angles to the starting positions. If Kismet was previously turned off while the servo angles were in the opposite direction, the next time Kismet will be turned on again, the servo motors will violently return to the starting position. Of course a safe turn off option can be added but this would not inhibit the user of removing the power suddenly. Thus, instead Arduino should save the last servo positions in long term memory or on a SD card, such that on the next switch the starting position can slowly be changed from the last known location to the actual starting position.

In addition, the Audeme speech recognition module provides even more features than exploited in this report. Different speech synthesizer can be used, like a kid's voice, creepy voice, whispering or slow and fast talking. These can be implemented in the program to enhance the already present emotion expressions or in different modes.

Finally, an SD card module could be used to store the set of sentences required for Eliza to be learned once. The module could be permanently attached to read the sentences when needed for teaching Eliza, saving precious Arduino memory. Alternatively, if during the process no new sentences have to be learned, which is not the case for Kismet, the SD card module can be removed subsequently after learning has been performed. An active speaker can also be replaced by a passive speaker with an audio amplifier.

5.4 Future work

Toolkit could be used with Suhaib project. For this slight modification would need to be done. Even with the presence of a human facilitator, it would be nice if the building blocks were implemented with software enabling them to help the user. Accordingly, taking over partly the facilitator's role by creating a simple interface for the building blocks which would eliminate the need for programming skills. Here easy to use is important thus thorough testing is required.

Examples for further Eliza expansion could be a Kismet mode where the user can play a game either instructional or not. Eyes can be used to display text or other relative information and different voices for role playing. The possibilities are unlimited. Also without even mentioning the set of languages supported by Audeme. Ideas of potential games thought of were tetris using vocal commands or hand gestures, a snake game with similar features or a program to learn children a language through fun little games guided by Eliza. Stickers and color blobs can be used as learning tool.

Lastly, as mentioned earlier, it can be employed as a learning tool for sensor or robotic courses, disassembled by the students and reassembled in another small robot. Application could be smart cars for instance.

A Appendix

A.1 List of retailers

While comparing building blocks in section 3.1, several retail stores were visited. These are summed up in the following list:

- Antratek
- Kiwi electronics
- Sparkfun
- Seeed Studios Electronics
- Adafruit
- Conrad
- Vanallesenmeer
- Farnell
- Audeme
- OpenMV
- Pixycam

A.2 Building block advantages and limitations

There are 2 main categories of speech recognition modules and smart cameras. The first category are for simple projects that require relative simple features. The second category are for the project that need more stuff, such as a chat bot (cognitive responses) or extensive computational power.

Grove speech recognition:

Best for: Home automation projects and simple speech recognition projects. Uses pretty limited.

Advantages:

- Cheap
- Easy to use and fast to set up
- Lots of information with Arduino use
- Compatible with passive speakers

Limitations:

- Fixed list of 22 commands (see here)
- Fixed callsign "Hicell"
- Speaker will only repeat the command (internal dictionary only contains the 22 built-in commands)

- Connected speaker must be lower than 1W to avoid damage to the chip
- Can have more difficulties understanding the callsign.

Respeaker series:

Best for: Projects that require more advanced voice recognition

Respeaker core V1.0:

Advantages:

- Does not require an Arduino micro-controller as it has it's own ATMega32U4 chip, programmable with Arudino IDE
- Online and offline speech recognition (custom commands for smart voice assistant)
- Extensive documentation
- 6 mic-array offers good sound capture
- Possibility of cognitive API integration due to wifi connection

Limitations:

- Does not support Android
- ATMega32U4 can only be used to control 12 onboard RGB LEDs and the 8 touch sensors.
- Setup requires internet access
- Less of an active community

Respeaker core V2.0:

(on top of the advantages and limitations of the Respeaker core v1.0)

Advantages:

- Contains more advanced features such as beam forming and noise suppression
- Many more different possible uses and applications
- Supports android devices with SDK
- Many extra shields available
- Supports many languages (Alexa languages)
- Probably the best speech recognition board out, provides everything, very customizable, could be utilized for more professional uses

Limitations:

- No ATMega32U4 and touch sensors
- More complex than offline boards and than the Google AIY voice kit

The ReSpeaker can be a smart assistant comparable to the Google AIY voice kit and commercial voice assistants but fully customizable and without many of the disadvantages that come with the google AIY voice kit. ReSpeaker would be a replacement for the Arduino and/ or Raspeberry Pi as a stand-alone kit. Nonetheless, setup is not as easy as any micro-controller shield and requires some knowledge about coding. It is also slower than a Raspberry Pi.

Audeme MOVI:

Best for: Intermediate to big offline projects

Advantages:

- 200 sentences have been tested by the manufacturer, however users reported it can understand up to 1000 different voice commands
- New firmware provides push-to-talk functionality
- Callsign can be reomoved and changed
- Can recognize long sentences
- Can learn new sentences while running
- Can change the type and the speed of the voice

Limitations:

- Needs an external power supply
- Requires an active speaker

Lcd screen:

Best for: Projects that do not require many custom characters

Advantages:

- Cheap
- Exists in many different sizes and color models
- Simple screen

Limitations:

- Only 8 custom characters

Epaper:

Advantages:

- Flexible, could be mounted on a bracelet
- Small

Limitations:

- Needs a fixed background (cannot be bend constantly)

Spresense:

Advantages:

- Main board has GPS
- Main board is a multi-core micro-controller
- Main board is compatible with Arduino

Limitations:

- Less commonly used than Arduino boards
- Requires its own Sony main board as micro-controller

A.3 List of emotions and feelings understood by Eliza

Eliza understands 34 different feelings and emotions. For the neutral emotion, no words are defined, instead it is used as a default emotion for Kismet. The words are:

Happy

- alright
- good
- fine
- happy
- joyful
- grateful

Surprised

- surprised
- stunned
- startled
- astounded

Confused

- stressed out
- stress
- worried
- confused
- nervous
- tension

Angry

- angry
- annoyed
- mad
- outraged
- irritated
- furious

Suspicious

- suspicious
- disoriented
- chaotic

Sad

- sad
- bad
- heartbroken
- unhappy
- sorry
- pessimistic

Sleepy

- sleepy
- tired
- drained

A.4 Bill of materials (Kismet)

The bill of materials presented by table A.1 describes the material cost for the Kismet prototype. Cables and small electronic components are not taken into account. Furthermore, all prices in dollars are converted in euros with a factor of 0.89. The estimated cost is based on the actual transaction price or if not available an estimated number according to pricing from websites listed in Appendix section A.1. Note that the Arduino Mega ADK has retired and thus the selling price of the new Arduino Mega 2560 Rev3 is taken as estimation. An external active speaker is used for testing purposes and it does not participate in the total cost of this prototype. It thus comes on top of the bill of materials presented in table **??**.

Product	**Description**	**Quantity**	**Estimated cost**	**Total**
Micro-controller	Arduino Mega ADK	1	€34.70	€34.70
Speech recognition	Audeme MOVI shield	1	€66.67	€66.67
Camera	OpenMV M7	1	€69.88	€69.88
Display	8x8 DOT matrix MAX7219	2	€5.95	€11.90
Servo motor	H-king - HK 15168	2	€2.41	€4.82
5V 1A DC-DC converter	SR10S05 - XP Power	2	€7.45	€14.90
MIDI panel	Korg Nanokontrol 2	1	€45	€45
Casing	Laser cut wooden housing	1	€0	€0
Total		11	€247.87	

Table A.1: Qualitative and quantitative requirements for the building blocks

By far the most expensive components of Kismet are the speech recognition module and the OpenMV M7 camera. This is expected as they also provide the most complex features.

A.5 Demonstration manual

Powering on and off Kismet can be turned on and off by plugging and unplugging the 9V power cable into the Arduino Mega.

IMPORTANT, do not power the whole system by USB only! Always make sure that the 9V power cable is unplugged last and the USB to the computer first if necessary. Otherwise MOVI can remain in an undefined state.

Resetting the Arduino resets the program. If MOVI is not acting normally, then also the MOVI shield reset button can be pushed. Note that MOVI should never be connected to a passive speaker, always an active one!

Eliza therapist mode Volume control is done using the last slider on the MIDI panel and the MOVI threshold can be changed with the knob above. Further, the SET button and the two adjacent arrows are used to control the current mode.

Sandbox and others In the sandbox mode, the same MIDI controls are used except that the two sliders next to the volume control are used to change Kismet's emotion. While the first two control the two servos. Recording is performed exactly as described in Dertien (2019).

Bibliography

Aslam, S. (2018), Co-designing a Collaborative Sobot Co-creation Toolkit (Co3), Bachelor's Thesis, Univesity of Twente, The Netherlands.

Autodesk (2010), Tinkercad circuit designer.
https://www.tinkercad.com/things/3caTBKyJgSk-bodacious-albar-amur/editel?tenant=circuits

Azouz, A. (2019), Rory the robot plant.
https://create.arduino.cc/projecthub/AhmedAzouz/rory-the-robot-plant-f7e74b

Babar, N. (2018), The Levenshtein Distance Algorithm, *Big Data Zone*.
https://dzone.com/articles/the-levenshtein-algorithm-1

Breazeal, C. and B. Scassellati (1999), How to build robots that make friends and influence people, in *Proceedings of the 1999 IEEERSJ International Conference on Intelligent Robots and Systems*, MIT Artificial Intelligence Lab 545 Technology Square Cambridge, MA 02139, pp. 858–863, doi:10.1109/IROS.1999.812787.
https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=812787

Cang, L., P. Bucci and K. E. MacLean (2015), CuddleBits: Friendly, Low-cost Furballs that Respond to Touch, in *ICMI '15 Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, Seattle, WA, USA, pp. 365–366, ISBN 978-1-4503-3912-4, doi:10.1145/2818346.2823293.
https://dl.acm.org/citation.cfm?doid=2818346.2823293

Davison, D., B. Gorer, J. Kolkmeier, J. Linssen, B. Schadenberg, B. van de Vijver, N. Campbell, E. Dertien and D. Reidsma (2016), Things that Make Robots Go HMMM: Heterogeneous Multilevel Multimodal Mixing to Realise Fluent, Multiparty, Human-Robot Interaction, in *Proceedings of eNTERFACE'16*, Eds. K. P. Truong and D. Reidsma, CTIT Workshop Proceedings WP 17-02, University of Twente, Netherlands (July 2017), The 12th Summer Workshop on Multimodal Interfaces, pp. 6–20, ISSN 0929-0672.
https://ris.utwente.nl/ws/portalfiles/portal/21753635/2016_truong_proceedings_enterface16.pdf

Dertien, E. (2019), Animatronics Workshop - Servo Recorder.
http://wiki.edwindertien.nl/doku.php?id=modules:servorecorder

Hoffman and Ju (2014), Designing Robots with Movement in Mind, **vol. 3**, no.1, pp. 82–122, doi:10.5898/JHRI.3.1.Hoffman.
http://guyhoffman.com/publications/HoffmanJuJHRI14s.pdf

Hoffman, G. (2007), *Ensemble: Fluency and Embodimen for Robots Acting with Humans*, Ph.D. thesis, School of Architecture and Planning, in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Media Arts and Sciences at the MASSACHUSETTS INSTITUTE OF TECHNOLOGY, 77 Massachusetts Ave, Cambridge, MA 02139, USA.
http://alumni.media.mit.edu/~guy/publications/HoffmanPhDThesis.pdf

Hoffman, G. and C. Breazeal (2009), Effects of anticipatory perceptual simulation on practiced human-robot tasks, Springer Science+Business Media, doi:10.1007/s10514-009-9166-3.
http://hrc2.io/assets/pdfs/papers/HoffmanBreazealAuRo10.pdf

Iolanda Leite, André Pereira, S. M. C. M. R. P. A. P. (2013), The influence of empathy in humanâĂŞrobot relations, *International Journal of Human-Computer Studies*, **vol. 71, issue 3**, pp. 250–260.
https://www.sciencedirect.com/science/article/abs/pii/

S1071581912001681

Ireland, C. (2012), Alan Turing at 100.
https://news.harvard.edu/gazette/story/2012/09/alan-turing-at-100/

de Kruif, K. (2018), RAM wiki.
https://www.ram.ewi.utwente.nl/e13/

Mendez, R. C. (2018), *A robotic social actor for persuasive Human-Robot Interactions*, Master's thesis, University of Twente, 7500 AE Enschede The Netherlands.

Norvig, P. (1992), *Eliza: Dialog With a Machine*, chapter 5, pp. 151–174, doi:10.1016/C2009-0-27663-X.
https://www.sciencedirect.com/book/9780080571157/paradigms-of-artificial-intelligence-programming

Vandevelde, C., J. Saldien, M. C. Ciocci and B. Vanderborght (2014), Ono, a DIY open source platform for social robotics, embedded and embodied interaction.
https://www.researchgate.net/publication/285339202_Ono_a_DIY_open_source_platform_for_social_robotics

van de Vijver, B. (2016), *A Human Robot Interaction Toolkit with Heterogeneous Multilevel Multimodal Mixing*, Master's thesis, University of Twente, 7500 AE Enschede The Netherlands.
https://pdfs.semanticscholar.org/a216/15081944fe138742401b6c5f4843c8144efa.pdf?_ga=2.152895353.1738165896.1553112674-1334187617.1553112674

van Welbergen, R. Y. and S. Kopp (2014), AsapRealizer 2.0: The Next Steps in Fluent Behavior Realization for ECAs in: Intelligent Virtual Agents, *14th International Conference, LNCS*, **vol. 8637, Springer**, pp. 449–462.
http://www.herwinvanwelbergen.nl/publications/asaprealizer2.pdf

Wilson, P. I. and J. Fernandez (2006), Facial feature detection using Haar classifiers, **vol. 21**, no.4, pp. 127–133.
https://dl.acm.org/citation.cfm?id=1127416