

# Robot vacuum cleaner with personality traits

S.L. (Sverre) Boer

BSc Report

**Committee:**

Dr.ir. E.C. Dertien  
Dr.ir. D. Dresscher

July 2017

020RAM2017  
Robotics and Mechatronics  
EE-Math-CS  
University of Twente  
P.O. Box 217  
7500 AE Enschede  
The Netherlands

In this project the effect of personality traits in a robot vacuum cleaner is researched. The added personality traits should enhance the quality of cooperation between user and robot. Throughout applying multiple methods to ideate, a system specification is defined and a prototype is realized based upon those. The prototype is evaluated on functionality and user experience.

## Table of content

---

### I Context

1	Context	1
1.1	Introduction	2
1.2	Idea	2
1.3	Research questions	2
1.4	State of the art	3

### II Ideation

2	Ideation	9
2.1	Ideation process	10
2.1.1	Mind map	10
2.1.2	Action-emotion matrix	10
2.1.3	Sketches	10
2.2	Interviewing possible users	13
2.3	Ideation results	13

### III Specification

3	Specification	14
3.1	General specifications	15
3.2	Technical specifications	17

### IV Realization

4	Realization	18
4.1	Tchibo module	19
4.1.1	Existing hardware	19
4.1.2	Existing software	19
4.2	Hardware	21
4.2.1	Microcontrollers	21
4.2.2	Processing over Arduino	21
4.2.3	Wireless communication	21
4.2.3.1	XBee	22
4.2.3.2	NRF24L01	22
4.2.3.3	Parallax	23
4.2.4	Delay mapping	23
4.2.5	Distance sensor	24
4.2.6	LED Matrix	24
4.2.7	Additional features	24
4.3	Software	25
4.3.1	XCT-U	25
4.3.2	Pseudo code	26

	4.3.3	Code	27
	4.3.4	Libraries	29
4.4		Interactions	30
4.5		Exterior	30
<b>V</b>		<b>Evaluation</b>	
5		Evaluation	32
	5.1	Functional testing	33
	5.2	User evaluation	34
	5.2.1	Evaluation questions	35
	5.2.2	Evaluation method	35
	5.2.3	Evaluation results	36
<b>VI</b>		<b>Conclusion</b>	
6		Conclusion and future work	38
<b>VII</b>		<b>Appendices</b>	
7		Appendices	40
	A	Ideation	41
	B	Realization	46
	C	Evaluation	70
<b>VIII</b>		<b>Bibliography</b>	
8		Bibliography	75

## Table of figures

---

### I Context

- 1.1 Research questions 2
- 1.2 Mori's uncanny valley 3

### II Ideation

- 2.1 Mind map 11
- 2.2 Sketches 12

### III Specification

- 3.1 General specifications 15
- 3.2 Unique movement sets 16
- 3.3 Technical specifications 17

### IV Realization

- 4.1 Schematic existing framework 19
- 4.2 Interior Tchibo module 20
- 4.3 Bottom view Tchibo 20
- 4.4 Writing data to ATmega32 microcontroller 20
- 4.5 Communication cycle XBee and NRF24L01 chip 22
- 4.6 Delay mapping 23
- 4.7 Distance sensor code 24
- 4.8 Different one-personality-trait concepts 24
- 4.9 Example configuration XBee in XCT-U 25
- 4.10 Handwritten pseudo code 26
- 4.11 Summary of code functionality 26
- 4.12 Status update function 27
- 4.13 Safety function 27
- 4.14 Time-out function 27
- 4.15 Function to receive and read data 28
- 4.16 Reading ASCII strings input 28
- 4.17 Reading individual bits 29
- 4.18 3D print models 30
- 4.19 Laser cut models 31

### V Evaluation

- 5.1 Specification checklist table 33
- 5.2 Schematic of scenario user evaluation 34
- 5.3 Scenario setup: robot stuck between chair legs 34
- 5.4 Animal like body 35

# Part 1

## Context

# Context

---

## 1.1 Introduction

Throughout the past few decades, robots integrated into people's lives much better, because technological progression made high-end robots affordable for the average household. This increasing integration allows robots to play an increasingly bigger role in people's lives. As technology advances, so does the level of autonomy of robots. Higher levels of autonomy influence people's perception of such a robot, because they tend to start seeing it as an entity on its own rather than just a helpful device. This allows for relationships to grow from the user to the robot. Whether this happens consciously or unconsciously, it creates assumptions and expectations about the robot. Those, however, do not necessarily need to be positive or negative, but they for sure influence the band between user and robot. There has been done a lot of research already to modifying and personalizing robots, especially robot vacuum cleaners. Such a robot vacuum cleaner is taken as the base of this project. The environments in which they will operate mostly are people's houses.

## 1.2 Idea

The idea behind this project is to develop a robot vacuum cleaner with personality traits that improves the quality of cooperation between user and robot. Good cooperation is necessary in certain situations, like when a robot requires help from the user. Example situations are robots being stuck between chair legs or when they are laying upside down and get recover from it. A robot vacuum cleaner that has been used in a previous project will be used as the base for this project. It allows for easier manipulation of the robot because it has custom soft- and hardware in it. Adding extra features, such as sensors, actuators or a body and a control system to steer everything with, should form the base for the personality the robot vacuum cleaner will have.

## 1.3 Research questions

In order to see whether the done work this project has had a positive effect to the quality of cooperation between user and robot, research questions have be posed. The main- and sub research questions are formulated and displayed in table 1.1.

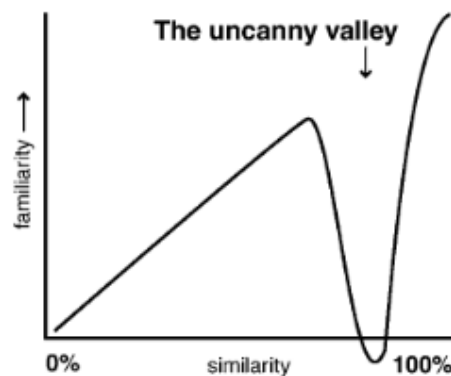
<b>Main</b>	<i>"How can the quality of co-operation between user and robot be improved by adding personality traits to it?"</i>
<b>Sub</b>	<i>"What personality traits are desired for better cooperation?"</i>
	<i>"How do people react to an intended interaction of the robot?"</i>
	<i>"How can people become more receptive towards robots?"</i>

Table 1.1 – Research questions

## 1.4 State of the art research

The work in this research is concerned with the area of social robotics. Social robotics is a field within robotics in general that specifies on autonomous robots that are able to interact and communicate with humans or other autonomous physical agents, while following certain social behavioral rules. In this field of expertise a lot of research is done to gain a deeper understanding about how to acquire and implement social skills and social intelligence into robots. Social robots are described by Fong, Nourbakhsh and Dautenhahn as “robots for which social interaction plays a key role” and exhibit ‘human social’ characteristics to them, amongst others; express and/or perceive emotions, communicate with higher-level dialogue, use natural cues (gaze, gestures, etc.), may learn/develop social competencies and exhibit distinctive personality and character [1].

In their research they explain that if a robot has to portray a living creature, it requires an appropriate amount of familiarity. It is, however, stated that the transition from non-realistic to realistic of a living thing is not linear. If the robot reaches a point where it is near perfect, the subtle imperfections seem to be more visible or disturbing. This causes the social robot to seem very unrealistic (figure 1.2). This effect is called the “uncanny valley” and was initially proposed by DiSalvo, Gemperle, Forlizzi and Kiesler [2].



1  
2 Figure 1.2 – Mori’s “uncanny valley” (DiSalvo, et al. [2]).

The term social robot was initially mentioned by Billard and Dautenhahn in 1997 in their research to “the usefulness of communication as a social skill for embodied robotic agents” [3]. Billard’s research contributes to the construction of autonomous social robots, which are able to interact with humans, by means of communication and imitation. In his study he tested in a teacher-student setup, in which the teacher had to perform movements and attach a name to the movement. The student robot had to imitate those movements and through this way, learn which movement belonged to which word. The outcome stresses the importance of robots possessing social skills in order to co-operate with humans more effectively: “Robots have to communicate with humans in order to get instructions or feedback, to learn or to express malfunctions. Communicative skills are necessary to express internal states, e.g. motivational or emotional states to other agents” [3].



Adams, Breazeal, Brooks, Scassellati and MIT Artificial Intelligence Laboratory discuss their projects aimed at developing robots that can behave like and interact with humans [4]. In their paper they explain the difference between a, what they consider to be, humanoid robot and social robot. Humanoid robots are robots that: “act autonomously without human control or supervision, in natural work environments and interact with people”. While social robots are robots that must be able to understand natural human cues and gestures. So they distinguish two different types of ‘social’ robots, where one is specifically aimed at understanding human behavior and the other must be able to function in an environment surrounding humans.

The presented studies all have their own interpretation of what the definition of a social robot is. However, for this research it is assumed that a social robot, a humanoid robot or robot expressing humanlike behavior is the same. In this research a social robot is a robot for which interactions with humans play a key role, the nature of those interactions is left undefined as long as they involve a human being.

### **Perception of social robots**

The reason for people perceiving robots in the way they do is due to how robots are depicted in the media. Harbers, Peeters and Neerincx state: “Most people’s conception of what a robot is appears to be largely based on the way robots are depicted in fiction” [5]. There are, however, notable differences between how robots occur in real life and in fiction. “In the field of robotics, robots are usually considered as computer-controlled machines that can perceive and manipulate their physical environment” [6]. While in fiction robots are depicted fundamentally different and therefore do not match the people’s perception of robots.

The first image about a robot is shaped by its look, there are many different sorts of robots and they can be categorized. Fong [1] describes five different categories. Morphology is a form and structure of a robot that helps to establish social expectations. This means that people can relate the appearance and behavior of the robot to something they already know. Examples are; a robot that resembles a dog will be seen as a dog and therefore will be treated, in general, with much more empathy than a robot that resembles kitchen appliances. Anthropomorphic robots are robots that possess a form or structure that seems like a human embodiment. Such robots often help humans will rationalizing the robot actions. It is often cited as a requirement for meaningful social interaction ([1], [7], [8] and [9]). Zoomorphic robots are robots that possess a form or structure that, most often, resembles animals or pets. “Avoiding the ‘uncanny valley’ may be easier with zoomorphic design because human-creature relationships are simpler than human-human relationships” [1]. Furthermore Fong describes caricatured and functional embodiments. Caricatured robots do not appear to be realistic, because their form and structure resemble that of a character, rather than an actual living creature. Last, the robots with a functional embodiment should reflect the task it has to perform. Other important aspects discussed by Fong that shape the perception of a robot, next to its physical embodiment are; body language, emotion, dialogue and

personality in social robots. Emotion can be expressed in multiple ways and is therefore hard to properly capture in a robot, ways of expressing emotion can be done through; body language, facial expression, speech, sounds and movement. It is however not limited to those option only. In [1] a couple of these methods are discussed. Speech is considered to be an effective way to express emotion, since it can be literately told directly to the user. Facial expression is a more indirect way of expressing emotion. It is just as with humans, an effective way of communicating emotion. Dialogue correlates to speech; however, there is a fundamental difference. A robot may be able to speech or say words, but it does not mean that it is able to have a dialogue with its user. Possessing the skills to have a dialogue with the user often shows a high level of intelligence.

Hendrik, Meerbek, Boess, Pauws and Sonneveld [9] did research on the desired personality in a robot vacuum cleaner. In this research they performed an interview before showing the participants a video prototype. Video prototyping is considered to them as “a suitable way of studying human-robot interaction and could lead to results that are comparable to those that could be obtained from live interactions with the robot”. They performed an experiment with ten different situations in which the robot vacuum cleaner, for example, has to recharge its battery, the vacuum cleaner has to be emptied or it is vacuuming dirty spots. Their results showed how the participants perceived the robot vacuum cleaner and as how likeable specific personality traits were experienced. It showed that people preferred a vacuum cleaner that is cooperative and has a structured way of working. It did not necessarily have to be calm or polite, although the opinions were divided about this.

Another study by Oestreicher and Eklundh researched user expectations on a domestic household robot showed that a big percentage of people are willing to have a robot support in their house [10]. This result complements other findings, in which was shown that that a bigger percentage of people (84.2%) are willing to have a robot support in their house [8]. Nonetheless, from the results obtained through interviews, they concluded that from the people who are receptive towards a domestic robot; expect help from a support robot for mainly: dish washing, window polishing, dusting, wet cleaning and washing clothes. Regarding tasks that people want to do themselves are, amongst others; walking the dog, being a butler at parties or taking care of children. So, this study by Oestreicher showed that people perceive a support robot an aid for a limited amount of actions, which humans usually consider as boring tasks to do.

So in general various studies show that people perceive robots, regardless of their form, mainly as an aid, which supports the user in doing day-to-day tasks that they rather not do themselves. The image that people have of robots comes mainly from the media and might be unrealistic, because in films they are usually depicted as futuristic devices, that do possess functionalities autonomous robots in real life do not yet have. However, people’s perception is not completely wrong, because they regard supportive robots as for what they are and not much more. Furthermore, these robots can come in all different forms and shapes, but regardless of that it is important for people that its form and behavior suits its task.

## **Expectations for social robots**

User expectations about social robots vary a lot, but the leading expectation is that those robots have to perform the task they are designed for. Ray, Mondada and Siegwart [8], Tapus, Ferland, Edgs and Goebel [6] and Oestreicher and Eklundh [10] support this statement in their researches. They all argue that the design of a robot should match its task description, to provide an example: if a robot is designed for a rough task, like drilling, it should have a solid and strong look. So it appears that the physical design of a robot strongly correlates with the perception of it, as is also mentioned by De Graaf et al., [11].

Commonly found domestic robots are butler, or so-called servant robots, which are found in many different types. Oestreicher, L. et al., did research to what disabled people seek in a domestic robot. Results of this research showed that they mainly want domestic robots to perform smaller tasks that they themselves cannot perform anymore, like lightings a cigarette, holding a drink or turning a page of a book [10].

In the research of Hendriks et al. [9] people got a vacuum cleaner robot to use in their house for a longer period of time and were asked what they did expect from it. It sounds rather straightforward, but they expected it to clean their houses as it was designed for.

Ray et al. [8] explain on the basis of their questionnaire results that people expect pragmatic and daily help from domestic robots, but they do not desire child or animal care. All the researches provide evidence that, how straightforward it might sound; people expect from a domestic robot to provide aid in the form the robot is designed for and expect it to properly execute its task.

## **Human experience of robot interaction**

The way people experience the interaction with a robot is different for every individual and the options are limitless due to the big variety of robots, however, in this literature research multiple research results are shown in order to give a glimpse of how people react to specific interactions.

General findings of different experiments show that the level of autonomy in domestic robots is still rather low. In [10] Oestreicher and Eklundh conduct an experiment in which they use different servant robots to perform several tasks, varying in complexity, within the house and measure how people react to that. The results show that users were not impressed by the level of autonomy of the robot and expected more intelligent actions than just pouring drinks or perform simple cooking tasks. A major lack within this robot intelligence is that these domestic robots are not able to take instructions during the execution of a task, they only take it prior to. Another interesting experiment of Oestreicher [12], in which he build a domestic robot with which the user can talk, gave a clear example that it is important to place such a robot in the right context. In the experiment his daughter and the robot had a one-on-one session in which they interacted with one another. The daughter spoke to the robot in Swedish and it understood all she said, but it responded in English. This caused confusion for the daughter. Oestreicher concluded

that users have expectations, which they consider being natural, but are for robots not obvious or natural.

According to Cuijperes, Bruna, Ham and Torta [13], humanlike robots are more likely to be trusted and therefore are better able to co-operate with their user. They, on the contrary, emphasize that movement and interaction are what improves the quality of co-operation instead of mainly design and looks. The conclusion they drew, which contradicts the observations in [12], was that the attitude towards robots is dependent on how the user experiences the interaction and is independent from how the robot anticipates on changes.

These experiments provide evidence that due to the lacking intelligence of robots, the co-operation between user and robot is far from desired. Fink, Bauwens, Kaplan and Dillenbourg stated that “At this moment, we believe that, as soon as robots and humans are sharing the same space, they need to adapt to each other to be a good match: people need to learn how to use a robot in an effective way, by building trust in it and by letting it to its intended task.” [14]. This explains that improvements have be done on both sides; the user must gain a better understanding of the capabilities of the robot and the robot has to be further developed to perform more intelligent or useful interactions.

### **Acceptance of social robots**

Creating acceptance amongst users towards social robots is mainly done by ensuring the robot is able to do its task or that users emphasize the robot and create a bond with it, regardless of how well it functions. Throughout a lot of studies different aspects are discussed that are important in raising acceptance towards domestic robots. Some results contradict each other, whilst some complement each other. For example, the results from [5] and [11] state that the receptiveness towards humanoid looking robots is high, whilst in [9] is stated that humanoid looking are not a good option and a domestic robot should look like a small machine. According to [9] and [7] robots do, however, not necessarily have to look humanoid in order to be anthropomorphized. Thus both humanoid looking robots and non-humanoid robots can both be considered as anthropomorphic. This shows that raising acceptance towards robots can be achieved in both forms. So, the look of a robot is an important aspect of creating acceptance, but more important is that the user itself anthropomorphizes the robot and this is mainly done through the interactions the robot performs.

Results that complement each other are that the user should feel comfortable and safe around the domestic robot. In addition to that it should meet the moral and ethical standards and must be safe to use [7] and [10].

Nonetheless, the most important aspect of creating acceptance, is shown in the results of many different studies ([5],[8],[9],[10],[11], [12] and [14]), is that the domestic robot should properly perform the task it is designed for. Consider buying a drill that does not drill holes, that is in any case the most undesirable aspect. So it is safe to conclude that the capabilities or functionality of a domestic robot should meet the user expectations.

## **Concluding words**

After the research done in the previous paragraphs an answer can be formulated to the question how the quality of co-operation between user and robot in a domestic (or apparently: a social-) environment can be improved by: by creating both more realistic and justified expectations about domestic robots, users will understand better how the robot will act, react and function and start seeing the robot for what it is. This will reduce if not erase the misconceptions about what the robot is intended to do and will ensure that the user's expectations match the task description of the domestic robot. Assuming the robot is capable of properly performing its task. It is proven that because the robot is operating in the house of the user, it is also important that the user feels safe around the robot and comfortable using it to improve the quality of co-operation.

Furthermore, this research does not provide a solid answer to how a robot should look to improve the co-operation. However, it can be concluded that when people anthropomorphize the robot, regardless of how it looks, they are more receptive for good a co-operation. So the quality of co-operation is seemingly independent of the looks of a robot. Because this research is rather limited in comparison to how big the field to cover is, further research to the relation between the look of a robot and its effect on the level of co-operation. Second, it is recommended to do further research on how to create more realistic and justified expectations about domestic robots. Because throughout this research it is concluded that this will improve the level of co-operation, but it does not clearly specify how. A third recommendation is to do research on how to create the feeling of safety and comfort around robots amongst users. The accentuation could hereby lie on the anthropomorphizing of a robot, because it seemed that this is strongly increasing the acceptance towards robots.

## Part 2

# Ideation

# Ideation

---

## 2.1 Ideation process

In order to fill the concept pool different methods are used to devise a sufficient amount of ideas and concepts, which will form the foundation of the rest of the project. As a starting point a mind map is made to explore different approaches of how personality can be expressed by a robot. Making the mind map was an iterative process; after writing things down and connecting certain emotions, actions or functionalities to each other it was revised to improve it. The reason why the mind map technique was preferred over simply writing down notes is that it provided better insight in the possible structure of the entire system, instead of just showing individual components. Further ideation about how to shape the actual personality of the robot vacuum cleaner is done by making a matrix in which individual emotions are linked to possible actions that should reflect that express it. A third method of filling the concept pool and finding useful information was to explore existing systems or applications that have similar functionalities. However before doing so, the goal and target group of this project is defined. This should make it easier to find relevant applications and extract useful information from it. The final method used was making sketches about possible design choices, personality traits and actions. After the concept pool has been filled ideas from the ideation phase were used to construct a survey, these results provided a better insight about which ideas were good to incorporate in the system and which should be left out.

### 2.1.1 Mind map

The mind map (figure 2.1) is constructed from concepts that revolve around the central theme: the robot vacuum cleaner. Each branch growing from the green, central block specifies into smaller concepts or aspects of the system. Because a personality can be expressed in multiple ways it has been approached in different ways, personality expressed throughout; lights, movements, sounds and the actual emotions. Each group has its own color in the mind map.

### 2.1.2 Action-emotion matrix

The matrix (appendix B) is a list in which three different groups of emotions are written down, each individual emotion is accompanied with an example of how the robot vacuum cleaner could express it. The three groups of emotions are positive (green), neutral (yellow) and negative (red). The purpose of this matrix is to ideate about how specific emotions could be visualized in the system.

### 2.1.3 Sketches

Paper sketches are made to visualize some ideas (figure 2.2) and explore new concepts. The sketch also contains visualization about the functionality, movement and structure of the Tchibo robot vacuum cleaner module

Figure 2.1– Mind map

# **Ideation phase: personality traits and actions**

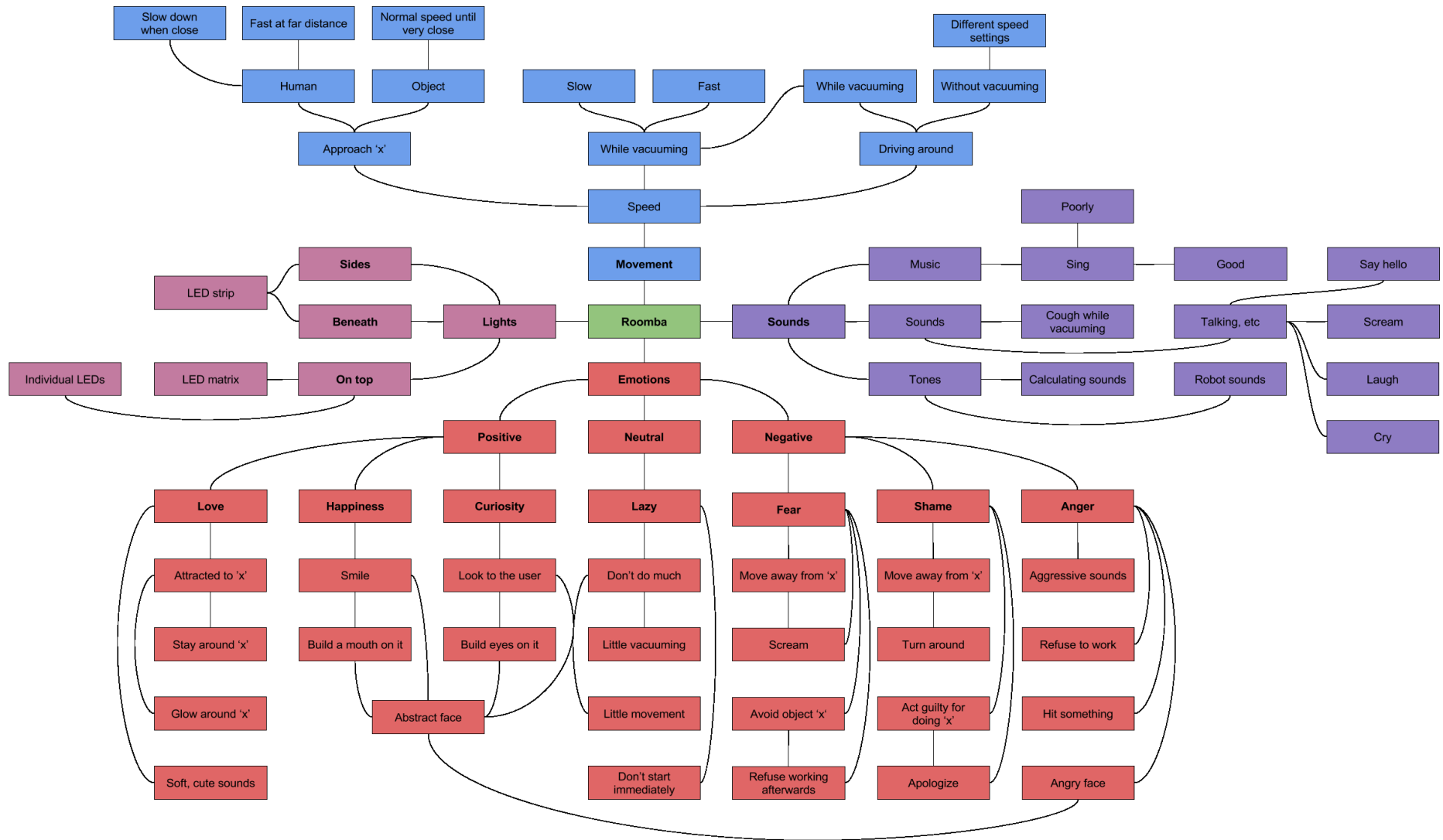
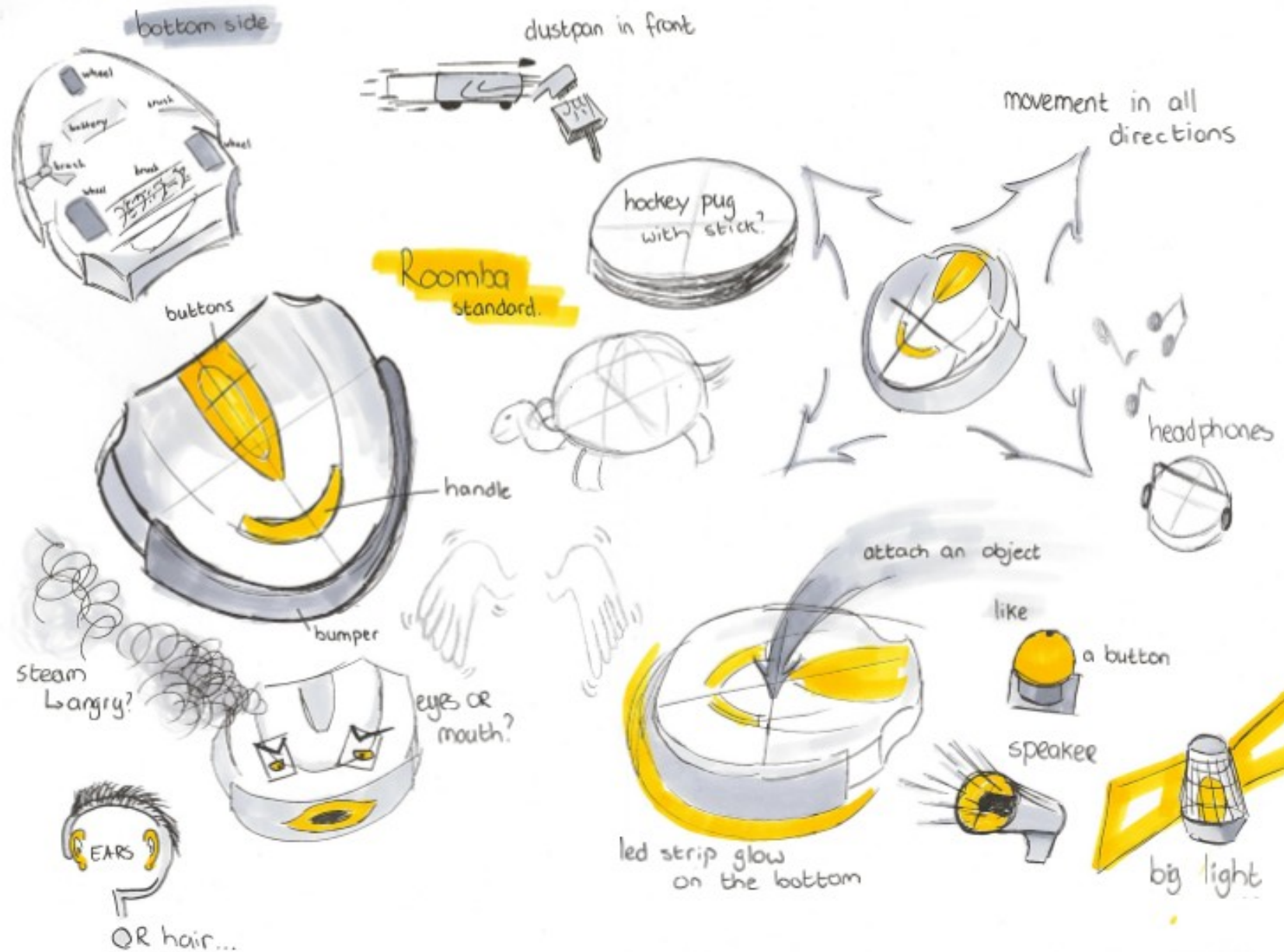




Figure 2.2 – Sketches



## 2.2 Interviewing possible users

After filling the concept pool with different concepts about how to shape the robot vacuum cleaner itself or its personality, some of these concepts were put to the test. A questionnaire (appendix B) with promising ideas and other important aspects was composed, in which users had to answer questions and give their opinion about statements in the following five categories: 'general information', 'appearance and form', 'behavior and movement', 'emotions and expressions' and 'user and control'. In the next paragraph the results are summarized and discussed.

## 2.3 Ideation results

Throughout multiple methods and approaches of the concept of a robot vacuum cleaner with personality traits a general idea has been established of how it should behave and look like. Respondents prefer a rather small vacuum cleaner, about the size of a middle size dog with preferable a robotic body. However, if the body resembles an animal or a pet, they would be fine with it too. It shouldn't drive too quickly and has to slow down when it is in a radius of approximately 1 meter from a human. Regarding personality traits it could express, they were enthusiastic about a robot that; is happy or could fall in love with objects, frustrated once in a while or yells at objects when it bumps into them, coughs when the floor is dirty, scared of some objects in the house or ashamed once in a while. Personality traits that were less likable to very undesirable are; rebellious or refusing to work, shy and sad. Furthermore the respondents want to have a feeling of control over the robot. Obviously it should be able to express its personality without having the user to interfere all the time, but the user still wants to have control over the robot when necessary. The ideation results are summarized in a point list and the ideation results from the interview can be found in the folder containing all the data and information of this thesis.

- Preferable a robotic body. An animal body is good too, but less desirable. Humanlike bodies were not preferred and the respondents felt neutral about the robot having a gender. Other suggestions for a body were; spherical shape, plants or flowers or comic super heroes.
- Respect personal space: reduce driving speed within a one-meter radius.
- The desired robot size is approximately the size of a middle-sized dog.
- Respondents would not mind if the robot was avoiding the user.
- Respondents would prefer the robot to be quite, ergo: it should not talk nor make too much noise.
- The robot must make a confirmation sound after a command has been given.
- The robot should be able to express its personality without intervention of the user; however the user must be able to intervene at any given moment.
- Personality traits:
  - Likeable: Happiness, frustration, falling in love (with certain objects or people), coughing (when the floor is dirty) and yelling (at certain objects or people).
  - Not likeable: rebellious behavior, refusing to work, scared and ashamed.
- Suggested personality traits: jealous, awkward, comedian, lonely, satisfied, desperate and motivated.

# Part 3

## **Specification**

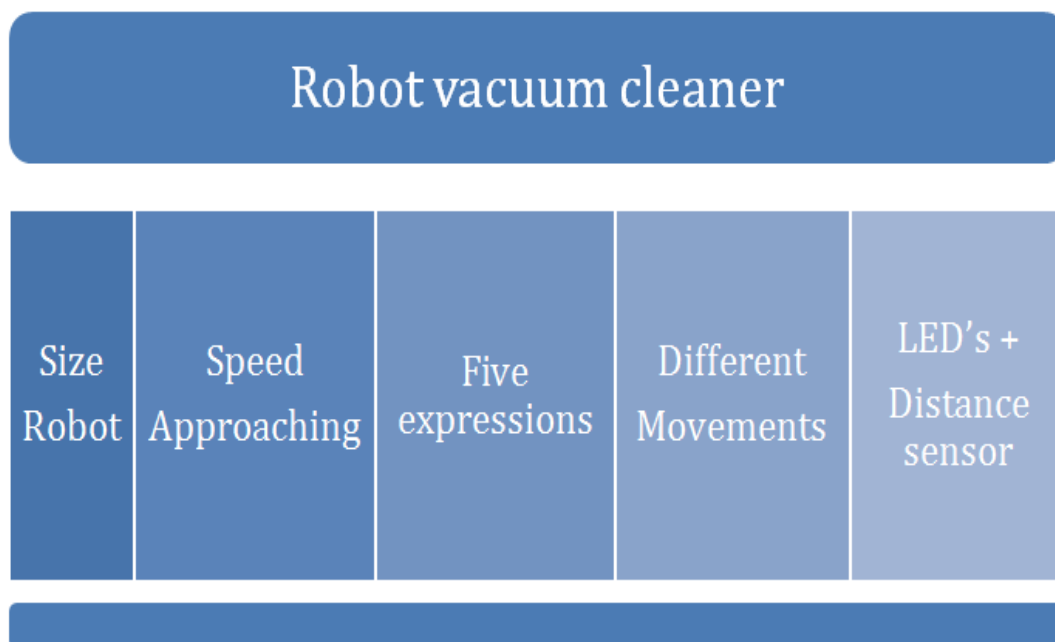
# Specification

---

With the ideations results of the applied methods, different approaches and help of respondents more specific design choices can be made. With all the possible concepts, specifications and designs it's now time to converge towards a more specific and fixed concept for the robot vacuum cleaner. This finite, fixed concept is later on taken to the realization phase in which it is put together and evaluated afterwards, from which conclusions can be drawn.

## 3.1 General specifications

Before the precise and more accurate specifications of the robot vacuum cleaner are defined, a more general overview is presented which shows in which fields the specifications are set. These general specifications are defined based upon the information gathered from the interview with possible users, in the ideation phase. They can be found in figure 3.1. It has been taken into account that the appearance of the robot is not yet specified in this table, because respondents stated that they would prefer a robotic body over any other. So therefore it is left open, for now.



*Figure 3.1 – General specification robot vacuum cleaner*

Size seemed to be an important aspect of the appearance of the robot; respondents stated that they would feel uncomfortable when the robot would be too big or too small. Furthermore size influences the thought of people about how efficient the robot vacuum cleaner could do its job. By translating all this information to a specific robot size, it resulted in the size of a middle sized dog. The Tchibo robot module available was already about this size; therefore no adjustments have to be made.

However, this also implies that additional features to the robot should not be too big as well. The movement speed of the robot influences the thought of people about how efficient the robot can do its job as well. It can also cause people to feel uncomfortable or unsafe when it is out of proportion, just like with the size of it. Another thing to take into account is that people want to have some sort of personal space (approximately one meter) and do not like it when the robot would move around or approach them too quickly. Given this information, the general movement speed of the robot is specified to about 80% of its maximum speed, which varies a little depending on its mood and it should slow down within the personal space of people.

There is a huge list of different emotions or expression that can be incorporated into the robot. Evers [15] has done a lot of research in this field. Social Robotics by Tapus, André, Ferlan, Eds and Goebe [6] is a research that focuses on robotic behavior in dynamic environments where humans are involved. It describes how robots should behave and interaction in order to be socially accepted. In this project the chosen amount of personality traits to incorporate is set at five, due to the scale of the project. It may be quite difficult to properly include rather specific expressions or emotions such as arrogance or flattered. Therefore the emotions that are picked for the robot vacuum cleaner are chosen based upon how easy they can be properly expressed with a limited set of methods, such as movement, sound and light. It is assumed that the easier and therefore better the emotion is expressed, the more vivid it is to the user which emotion is being expressed. To wrap this up, the five chosen emotions or expressions that are specified for the robot vacuum cleaner are: happiness, anger, scared, neutral and a coughing state. Each expression and emotion has to be accompanied by a unique set of movements in order to strengthen it. In figure 3.2 those five unique movement set are captured.

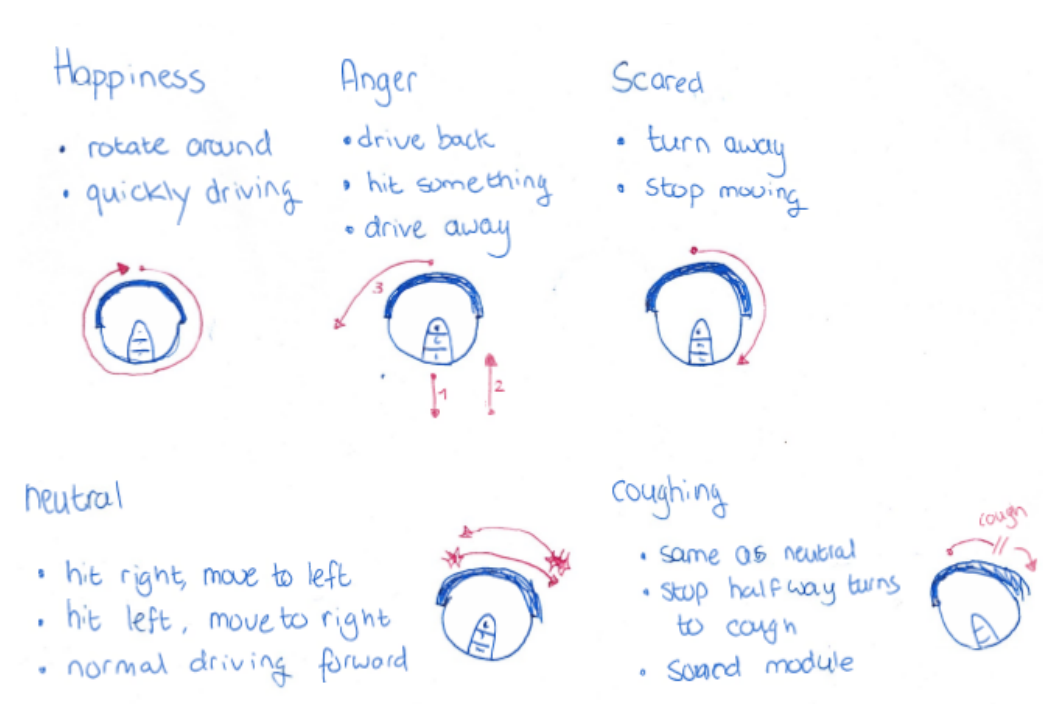


Figure 3.2 – Unique movement sets for each state

At last the way of expressing of these five states is done not only with the use of movements, but also throughout the use of facial expressions displayed on a LED matrix. To measure the distance towards objects and humans, a distance sensor has to be attached to the robot.

### 3.2 Technical specifications

The general specifications have specified which expressions and behavior will be included and how they will be incorporated into the robot, but these do not set explicit specifications for the technical side of the project. As there is also hard- and software involved, this has to be specified as well. The current modified Tchibo module runs with an ATmega32 microcontroller, which takes serial input from an AVR 15 connector. It is possible to send input to this microcontroller from Arduino and since that is the most accessible coding software, this will be used. Table 3.3 lists the technical specifications that are set for the robot vacuum cleaner.

What	Method	Specified tool
Writing software and programming the robot	Code	Arduino Nano (C)
Giving input to the ATmega32 microcontroller via AVR	AVR output	AVR 9 connection with 6 output pins
Measuring distance towards humans or objects to determine driving speed	Distance sensor	HC-SR04 Ultrasound sensor
Expressing emotions with facial expressions	LED Matrix	NeoPixel LED Matrix 16x16
Wireless connection to communicate with- and command the robot from the computer	Wireless communication	XBee 2.0, NRF24L01, parallax

*Table 3.3 – List of technical specifications*

## Part 4

# **Realization**

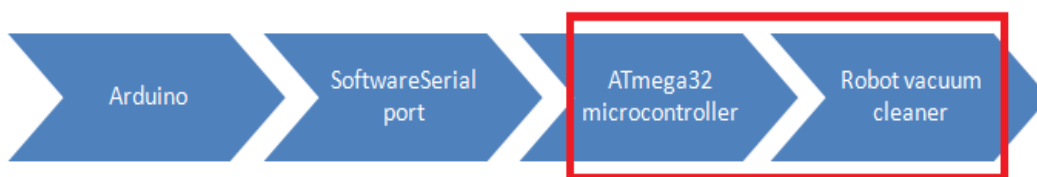
# Realization

---

As the system specifications regarding the appearance, behavior and design are set; it is time to realize the system by actually building it. The system specifications provide a good guideline to follow throughout the realization phase. However, while building it, it is possible that certain design choices do not seem viable or most optimal. Therefore it is possible that the robot design differs from the initial set of specifications. Realizing a project is after all an iterative process of reconsidering your design choices. The following chapter guides you through the entire process of realizing it in the same order as has been done during the project.

## 4.1 Tchibo module

The Tchibo robot vacuum cleaner module [16] is used as a base for this project. It is a modified vacuum cleaner with simple functionalities (which are also shown in figure 2.3). However, it has been used in former projects, such as the ‘vision controlled robot swarm’ by Stroeken [17]. Therefore the electronics and software are already modified. Figure 4.1 displays a superficial schematic about how to control the modified robot. The two blocks surrounded by the red square are the components that are in the robot and the robot itself. The other two components outside the red square is the recommended method of giving input to the system.



*Figure 4.1 – Schematic of how to work with the existing framework*

### 4.1.1 Existing Tchibo hardware

The modified robot vacuum cleaner is a Tchibo model from 1999. The electronics consist of an ATmega32 microcontroller running at 8 MHz, with two 75HC245 bus drivers for a 74HC373 output and require a 5 Volt power supply. Furthermore it has an AVR-9 connection to which external devices can be connected, such as an Arduino or an XBee (figure 4.2). The two motors and three brushes, as are shown in figure 4.3, are controlled by an L298 dual bridge driver and are powered by a MOSFET [18]. The electronic schematic of the system can be found in appendix B.

### 4.1.2 Existing Tchibo software

The software is programmed into the ATmega32 and is Arduino compatible. This means that data can be written to the modified Tchibo via a software port (called ‘Tchibo’ in the example down below) opened in Arduino. The main commands are listed and explained in figure 4.4. The Tchibo software is constructed so that it expects a carriage return (0x0D) and a new line (0x0A) byte after each command.



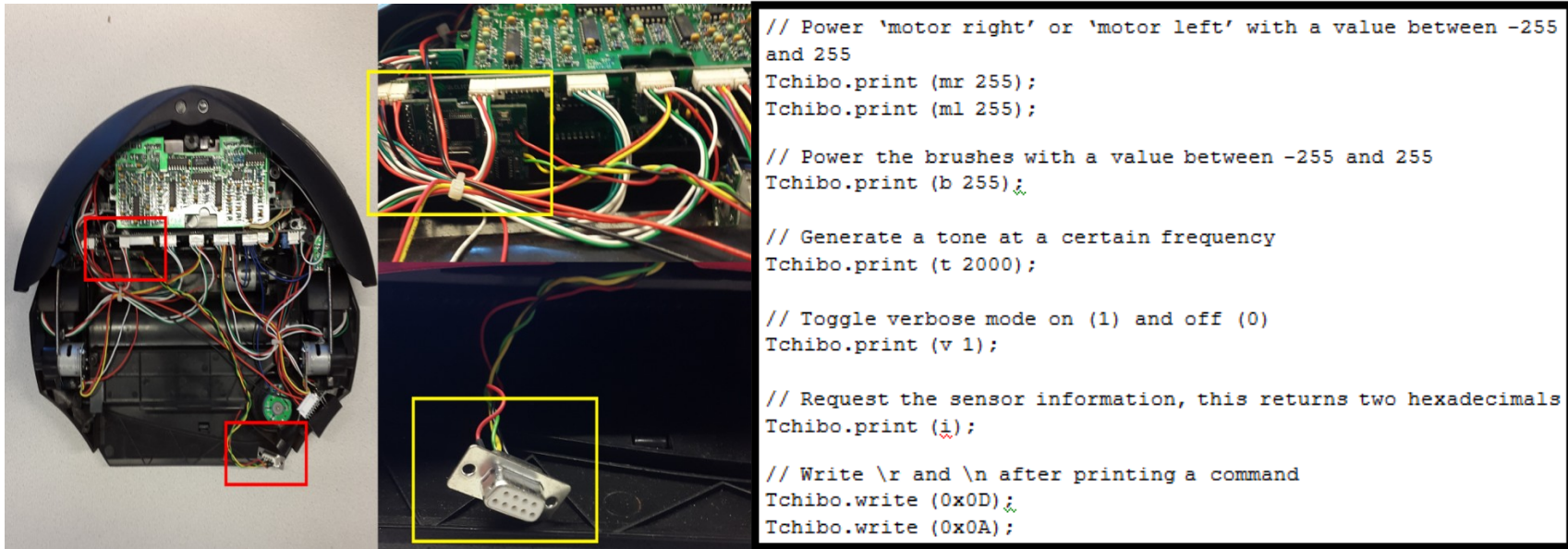
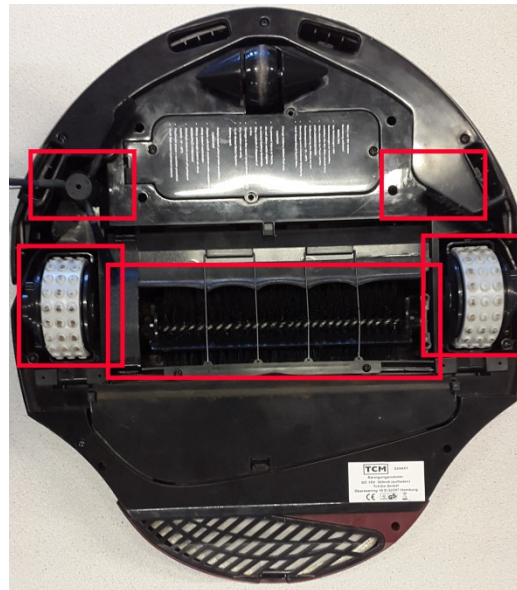


Figure 4.2 – Interior of the Tchibo

Figure 4.4 – Writing data to the ATmega32 microcontroller

Figure 4.3 – Bottom view highlighting the wheels and brushes



## 4.2 Hardware

After getting to understand the modified Tchibo module and how to use- and control it, additional pieces of hardware are added. In this paragraph all the components that have been tried or are being used in the final design are discussed, as well as why they are being used or discarded.

### 4.2.1 Microcontrollers

There are many different microcontrollers available and they all work in similar ways, but all of them differ slightly. These slight differences between microcontrollers are because they can be used with a different purpose. The Arduino Uno is the standard Arduino that is being used within Creative Technology and therefore has been used as a starting point to work with. However due to practical implementation issues in the robot it was more useful to go for a smaller Arduino, thus the Arduino Nano was used eventually. It was also considered to use a Galileo microcontroller instead, because this microcontroller provides a lot more functionality than an Arduino. It can be expanded with a Wi-Fi chip for example, which was an option to use for wireless communication. In the end this Galileo has not been used for wireless communication after all.

### 4.2.2 Processing over Arduino

During development and while setting up wireless communication between the robot and a computer, it has also been considered and tried to use Processing (Java) instead of Arduino. At the time it seemed like a solution to a problem with reading data from the robot in Arduino, however in the end it did not solve the problem.

### 4.2.3 Wireless communication

Three different communication applications have been tried; XBee, parallax and the NRF24L01 chip. However, in the end they were not incorporated in the robot after all. That problem that has been mentioned earlier was caused in the wireless communication protocol. Receiving and reading data in Arduino that was being received over any wireless communication system, had a delay of about one second. This caused too many problems with making the robot anticipate quickly enough to changes in its environment, because the software controlling the robot, only knows what happened a second after it happened. More factual information about this is provided in paragraph 4.3.

#### 4.2.3.1 XBee

The XBee series 2.0 have been used to communicate wirelessly between the computer and the robot. Two or more XBee's can communicate with each other after configuration with the XCT-U software and an XBee Explorer USB. In the configuration the channel ID, pan ID and the receiving- and sending addresses are set. Connect the XBee to the RX and TX pin of the software serial port on Arduino to use it. The specifications of the XBee 2.0 can be found in appendix B and in the extended folder with all the documentation of this thesis. Figure 4.5 shows the communication cycle in the system with two XBee's communicating. Furthermore there was a code written that makes use of the XBee's, consult this code in appendix B.



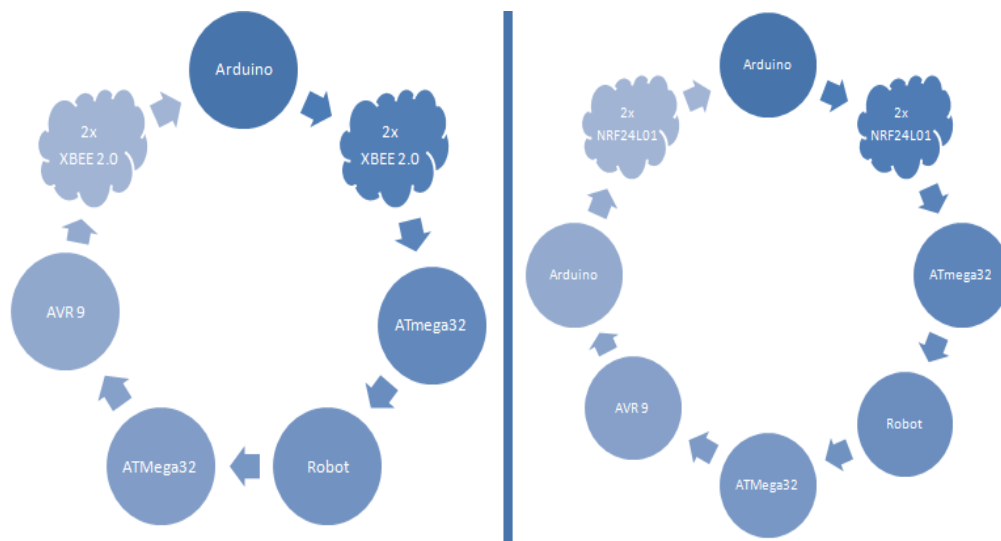
*Image: XBee S2 (source: [Sparkfun](http://Sparkfun.com))*

#### 4.2.3.2 NRF24L01

The NRF24L01 is a RF transceiver operating at a frequency of 2.4 GHz and is easily compatible with Arduino. It is a very tiny chip of a couple square centimeters and does not require any configuration, whereas the XBee does. A downside is that the NRF24L01 chip will also communicate with other chips, if they are nearby and two XBee will not. Figure 4.5 shows the communication cycle in the system with two NRF24L01 chips communicating. Furthermore there was a code written for this chip, consult this in appendix B.



*Image: NRF24L01 chip (Source: [dx](http://dx.com))*



*Figure 4.5 – Communication cycle for the XBee's (left) and NRF24L01 chips (right)*

#### 4.2.3.3 Parallax

The parallax is a RF transceiver operating at a frequency of 433 MHz and can be connected to Arduino, just like the other two options. It does not require much configuration like the XBee, but showed to be more complicated in usage. Therefore it was not a difficult decision to exclude it from the project and work with the other two wireless communication applications.

#### 4.2.4 Delay mapping

Instructing the robot vacuum cleaner via wireless communication from a computer worked perfectly. Sending information went quick and nothing was lost in the process. However, receiving information from the robot went slow. This caused a problem with reacting to environmental changes, such as driving into something and having to instruct the robot to stop driving or turn around. To find the exact problem all the delays between every communication component have been measured and put into a diagram. Initially this has been tested with the XBee's as communication tool. As can be seen in figure 4.6, showing the mapped delays, receiving data from the robot XBee on the Arduino XBee takes roughly 900 milliseconds. The figure displays the mapped delays for the XBee's, however, the same test has been done with the other transreceivers and the results were no different. It was unsure whether the precise problem of this huge delay was because either one of the following; Arduino reads data from software serial ports too slow or that the robot vacuum cleaner returns data too slow or that the wireless communication tools are returning data too slow. Anyhow, the conclusion of this test was that wireless communication was not viable to use for instructing the robot. Therefore it has been cut out completely and an internal Arduino directly connected to the robot will be used instead.

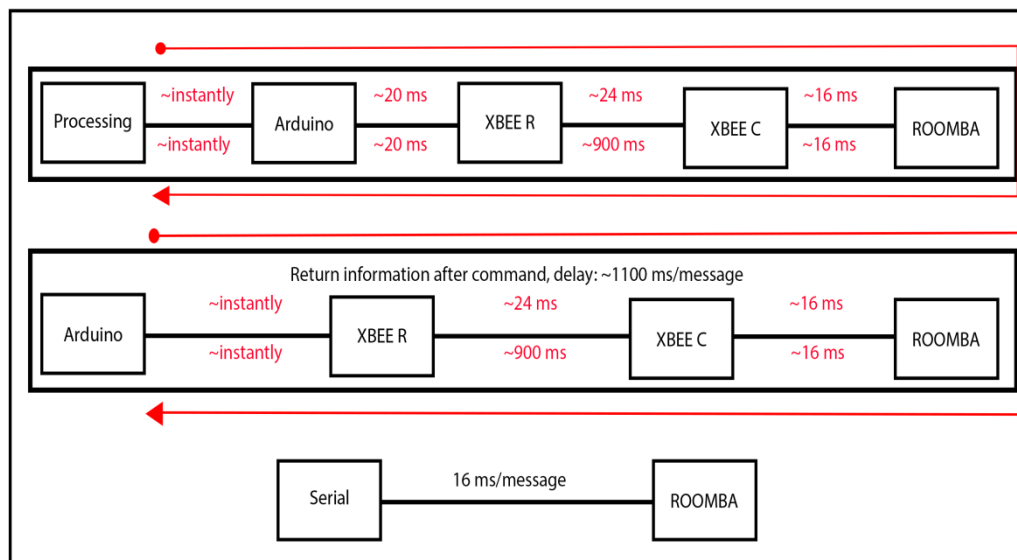


Figure 4.6 – Mapping delay between each communication component

#### 4.2.5 Distance sensor

A HC-SR04 ultrasound sensor (datasheet: [via Sparkfun](#)) is mounted on top of the robot vacuum cleaner which is used to measure the distance in front of the robot towards any object or human. The idea behind this is that within one meter of something it reduces its driving speed. It is a simpler concept of respecting the personal space of the user. The sensor makes use of four pins, of which two to power and ground it and two to measure the distance (the trigger- and echo pins). Figure 4.7 (left) shows the code that is used for this sensor; every distance above 1 meter is set to one meter.

#### 4.2.6 LED matrix

The NeoPixel 16x16 LED matrix (datasheet: [via Adafruit](#)) is mounted on top of the robot vacuum cleaner as well and it's used to display different faces each expressing one of the five emotions that are included. The matrix has three connections; a 5V power supply, a ground and the signal input. Figure 4.7 (right) shows the basic of the code for controlling the LED matrix.

```
void getDistance() {  
  // Send out a pulse, receive it and calculate distance  
  // by dividing the time between sending and receiving by the speed of sound  
  digitalWrite(trigPin, LOW);  
  delayMicroseconds(2);  
  digitalWrite(trigPin, HIGH);  
  delayMicroseconds(10);  
  
  duration = pulseIn(echoPin, HIGH);  
  distance = (duration / 2) / 29.1;  
  if (distance > 100) distance = 100;  
  if (distance < 0) distance = 0;  
  Serial.println(distance);  
}  
  
void power_a_led() {  
  // Set the brightness to value x; [0-255]  
  strip.setBrightness(x);  
  // Set LED n to the set RGB value  
  strip.setPixelColor(n, red, green, blue);  
  // Show all the set LEDs  
  strip.show();  
}
```

Figure 4.7 – Code for the distance sensor (left) and LED matrix (right)

#### 4.2.7 Additional hardware

In the ideation phase other design choices have been explored as well, in which only one specific personality trait is expressed but very obvious. It is a completely different concept, but still remains a robot vacuum cleaner with a robotic body. Even though it has not been developed any further, it is also a part of the realization phase.



Figure 4.8 – Three different one-personality-trait concepts

## 4.3 Software

All of the hardware in this project is supported by software behind it in order to make it work like it is supposed to work. The software is written in C in an Arduino environment. The XBee's had to be configured in the XCT-U software.

### 4.3.1 XCT-U

In order to configure XBee's to be able to communicate with each other they need to be on the same channel and have an identical pan ID; furthermore they need to know the address of destination of each other. Figure 4.9 shows an example of configuration settings for an XBee.

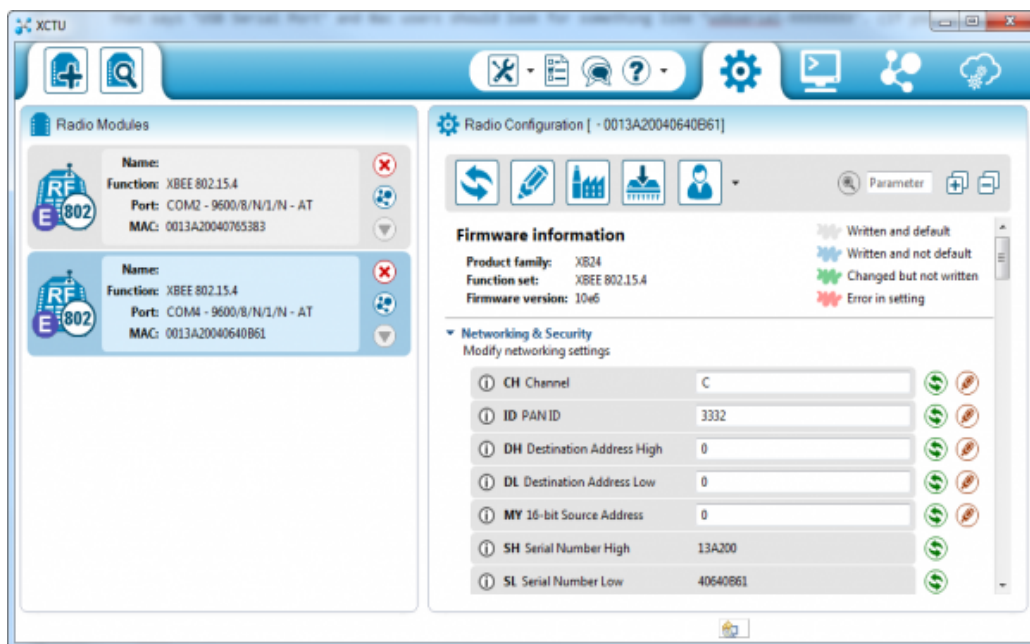


Figure 4.9 – Example configuration XBee (source: [sparkfun.com](http://sparkfun.com))



### 4.3.2 Pseudo code

Prior to writing an entire code, multiple pseudo codes have been written, which summarize how certain functions or groups of functions work and are structured. They provide a better understanding and insight in the code that has to be written. These pseudo codes have not been documented very well, because these have been written by hand on scraps of paper. An example of this can be found in figure 4.10. The different functionalities of the code can be summarized in four sentences. This is shown in figure 4.11.

The image shows two pieces of handwritten pseudo code on yellowed paper. The left piece is titled 'Transmit com 13 : ping-out, pin 9' and the right piece is titled 'Receive com 14 : ping back, pin 7'. Both pieces use C++-style syntax for radio communication and timing.

```
Transmit com 13 : ping-out, pin 9

radio.stopListening()
unsigned long time = millis()
bool ok = radio.write(&time, sizeof(time))

radio.startListening()
unsigned long start_wait_at = millis()
bool timeout = false

while (!radio.available && !timeout) {
  if (millis() - start_wait_at > 200) timeout = true
  if (timeout) // print it
  if (!timeout) {
    unsigned long got_time;
    radio.read(&got_time, sizeof(unsigned long))
    // print
  }
}
delay();

Receive com 14 : ping back, pin 7

if (radio.available) {
  unsigned long got_time;
  bool done = false
  while (!done) {
    done = radio.read(&got_time,
                     sizeof(unsigned long))
    // print
  }
  radio.stopListening()
  // some data
  radio.write(&data, sizeof())
  // print data
  radio.startListening()
}
```

Figure 4.10 – Handwritten pseudo code

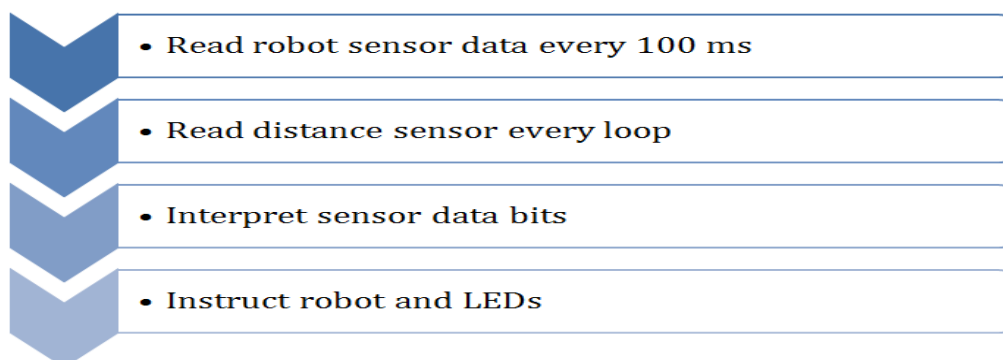


Figure 4.11 – Summary of the entire code in words

### 4.3.3 Code

The entire Arduino code consists of many different functions;, varying from interpreting translating hexadecimal numbers hidden in a string of ASCII numbers to turning on the LED's. In this paragraph the code is decomposed in its important functions and those are explained, the entire code can be found in appendix B.

The void setup() function of the code initializes; the Serial port, which was used for debugging the code, the software serial port for the robot vacuum cleaner and the LED matrix output pin. It starts some timers to check how long the code is running, ensures that the LED matrix is empty and makes sure that all the incoming data from the robot vacuum cleaner is disposed, because that is old and useless data.

The void loop() function is filled with four 'functions' that run other functions in it as well. The first function (figure 4.12) is one that requests a status update once every 100 ms when allowed. The second function (figure 4.13) is a build-in safety for the movement, so that the robot vacuum cleaner stops performing a set of movements to express an emotion after 2.5 seconds. The third one (figure 4.14) is a time-out function; it turns the robot to a neutral state if it does not receive any information from the robot. The fourth function (4.15) is the most interesting one, because in there the data from the robot is received, filter, read, interpreted and instructs the robot accordingly.

```
// ASK FOR A STATUS UPDATE EVERY 100 MILLISECONDS AND IF WE ARE ALLOWED TO ASK FOR ONE
if (timer - prevTimer > statusUpdateLoop && ackStatusUpdate == true) {
    // Read the distance
    getDistance();
    // Request status update and reset the timer
    prevTimer = timer;
    RequestStatusUpdate();
    ackStatusUpdate = false;
}
```

*Figure 4.12 - Status update function*

```
// IF WE MAKE A TIMEOUT AND HAVE NO STATUS UPDATE AFTER 3 SECONDS, REQUEST THEM AGAIN
if (timer - prevTimer > 2500 && ackStatusUpdate == false) {
    // Stop driving, and allow status updates to be requested again
    stopDriving();
    ackStatusUpdate = true;
}
```

*Figure 4.13 – Safety function*

```
// 5 SECONDS AFTER ANOTHER STATE IS PERFORMED, RETURN BACK TO THE NEUTRAL STATE
if (timer - prevTimer2 > 5000) {
    // Set it back to neutral state, instruct the robot and reset timer
    stateCycle = 1;
    controlStates(stateCycle, -1);
    prevTimer2 = timer;
}
```

*4.14 - Time-out function*



These functions are provided with comments that explain what is being done in there. In figure 4.15 the line 'DataHandling(incomingString)' is highlighted, because this is an important piece of code. The incoming string of ASCII numbers from the robot, after requesting a status update or even instructing the robot to do something, is filtered in such a way that only the sensor values are in the string. This string is passed onto the function called DataHandling(). In this function the data is being decomposed into useful sensor values. How this is done, is shown in figure 4.16.

```
// IF THERE IS DATA AVAILABLE FROM THE ROBOT VACUUM CLEANER, READ IT
if (Roomba.available()) {
    incomingChar = Roomba.read();

    // IF IT IS NOT THE LAST SYMBOL OF A STATUS UPDATE, READ THE CHARACTER
    if (incomingChar != '>') {
        incomingString.concat(incomingChar);
    }
    else {
        // FILTER OUT THE STATUS UPDATE, BY CHECKING FOR AN 'i' CHARACTER
        char firstChar = incomingString.charAt(0);

        if (firstChar == 'i') {
            if (stopped) stopDriving();
            if (!stopped) alwaysDriveForwardAtVariableSpeed();
            DataHandling(incomingString);

            // If we are done with reading and executing the data, accept status updates again
            ackStatusUpdate = true;
        }
        // Empty the string to put newly received data in it
        incomingString = "";
    }
}
```

Figure 4.15 - Function to receive & read data and call the DataHandling() function

Once the incoming string has been received and read, the DataHandling() function filters out the two hexadecimal values that represent up to sixteen different sensors or triggers inside the robot vacuum cleaner. In this project only eleven of the sensors are being used. The two hexadecimal numbers are translated to numerical values and then these integers are decomposed into bits. Each bit represented a certain sensor that is triggered or not. An example would be; bit number 5 of the second hexadecimal number means that the left bumper is pressed. Figure 4.17 shows the code that has been used for reading the bits.

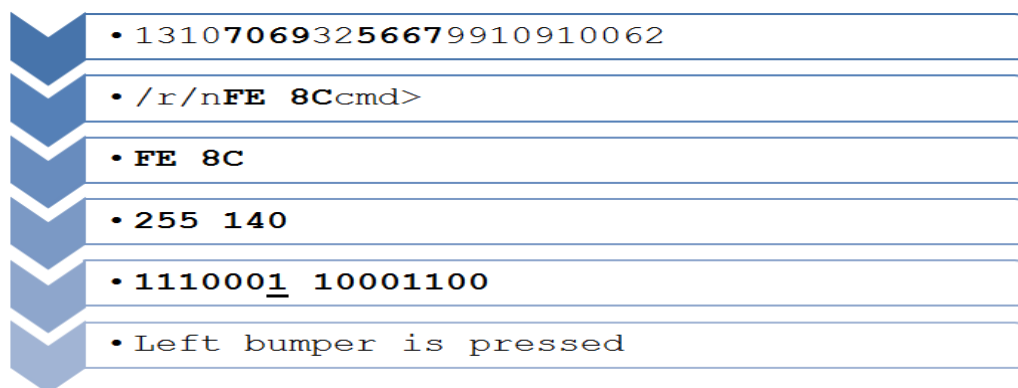


Figure 4.16 – Decomposing an ASCII string to usable data

```

// DO NOTHING WITH THE BYTES THAT ARE SEND IF THE ROOMBA IS TURNED OFF
if (byteone >= 0 && byteone <= 255 && bytetwo >= 0 && bytetwo <= 255) {
    if (byteone != 244 && bytetwo != 172) {

        // IF THE S-BUTTON IS PRESSED, STOP DRIVING
        if (byteone && byteone != -12288) {
            if (bitRead(byteone, BUTTON_S) == 0) {
                Serial.println("BUTTON S");
                stopped = false;
                stateCycle = 2;
                controlStates(stateCycle, -1);
                prevTimer2 = timer;
            }
        }

        // IF THE M-BUTTON IS PRESSED, START DRIVING, IF THE L-BUTTON IS PRESSED SWITCH STATES
        if (bytetwo && bytetwo != -12288) {
            if (bytetwo && bytetwo != -12288) {
                if (bitRead(bytetwo, BUMPER_L) == 1) {
                    Serial.println("BUMPER LEFT");
                    int bumper = 0;
                    stateCycle = random(1, 6);
                    controlStates(stateCycle, bumper);
                    ackStatusUpdate = true;
                    prevTimer2 = timer;
                }
            }
        }
    }
}

```

#### 4.17 – Reading individual bits

The function `controlStates()`, as can be seen in figure 4.17, does what its name implies; it controls the state of the robot. In this function instructions are send to the robot about what to do and how to behave.

#### 4.3.4 Libraries

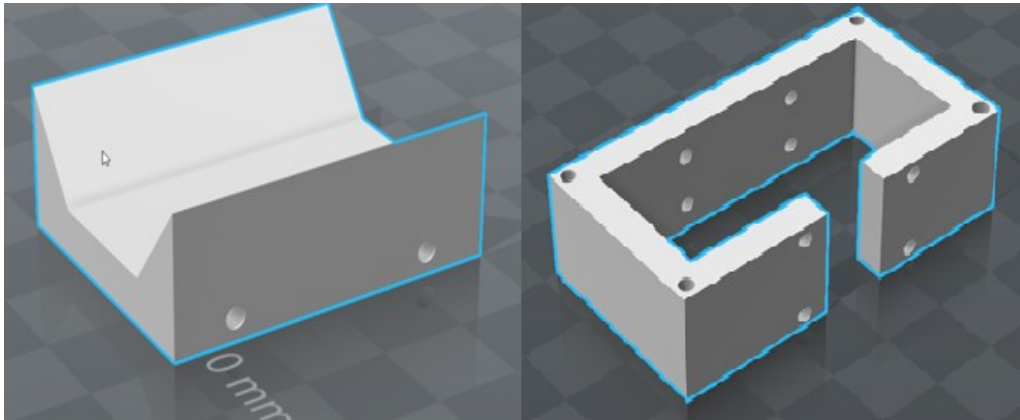
The code is for an extremely big part self written and libraries are barely used. However, to create a software serial port on an Arduino pin the `<SoftwareSerial.h>` library is used (source: Arduino). To control the NeoPixel LED matrix, the `<Adafruit_NeoPixel.h>` library is used (source: [adafruit.com](http://adafruit.com)).

#### 4.4 Interactions

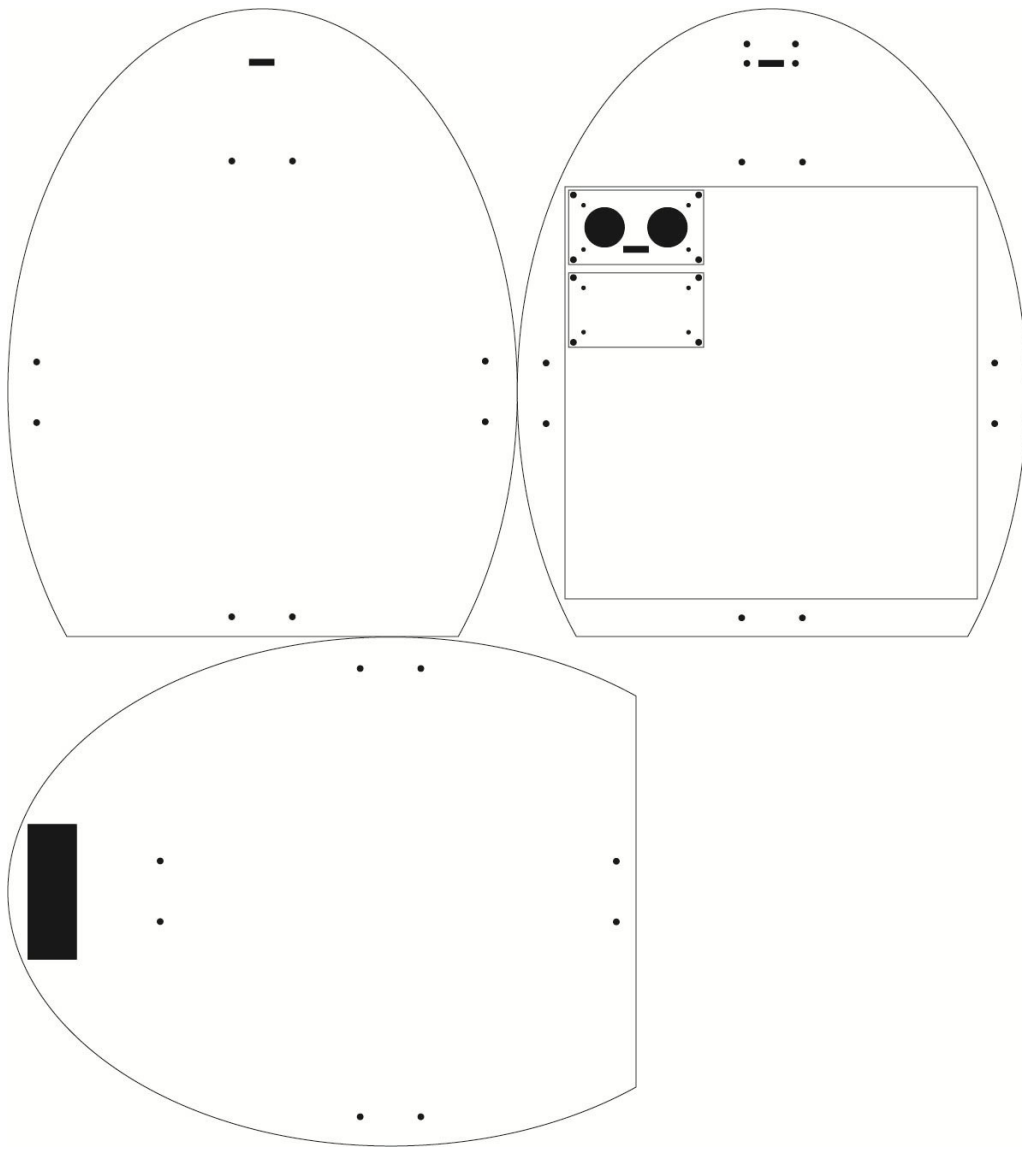
The interactions the robot vacuum cleaner will eventually make are already discussed into some extent through the report. However, the final interactions are the expression of five states; happiness, anger, neutral, scared and coughing. Each expression has a facial expression (consult appendix B for the sketches) that is displayed on the LED matrix and is accompanied by a unique set of movements (figure 3.2). Every time it enters a certain state, it makes a bleeping sound. There was too little time to include a sound module to the system; otherwise each expression would be accompanied by sound as well. To sketch a situation of how that would look: when the robot is vacuuming a very dirty floor and has to cough, it makes a coughing face, stutter in its movements and makes coughing sounds. Furthermore the robot has three buttons that can be pressed of which only two are programmed to do something. One button is to start driving and the other is to stop driving.

#### 4.5 Exterior

The exterior of the robot vacuum cleaner consists of a LED matrix and a distance sensor; those had to be mounted on the robot. In order to do so, a design has been constructed partly out of laser cut material and partly out of 3D printed models. In figures 4.18 and 4.19 these blueprints are displayed. Together they make up for a casing for the distance sensor and LED matrix with mounting pieces to drill it into the robot.



*Figure 4.18 – 3D models: montage piece (left), distance sensor holder (right)*



*Figure 4.19 – Laser cut models*

## Part 5

# **Evaluation**

# Evaluation

Now that the prototype in this project is finalized it is evaluated on functionality and user experience. The functional evaluation is based upon the defined specifications in chapter three and the user experience is evaluated with an interactive user test method in which they have to answer questions about the prototype and participate in a little scenario. Evaluating the user experience contributes to answer the research questions and the functional evaluation provides insight in how well the prototype has been realized.

## 5.1 Functional evaluation

The functionality of the prototype is evaluated again (it is also continuously done in the realization phase) by means of a test run in an exemplary environment, which is in a living room with an open kitchen. The robot vacuum cleaner is placed on the ground and started. After it has been running for a while it is checked whether all the initial specifications are included or met. In table 5.1 the specifications are listed and checked, there is also a third column with notes.

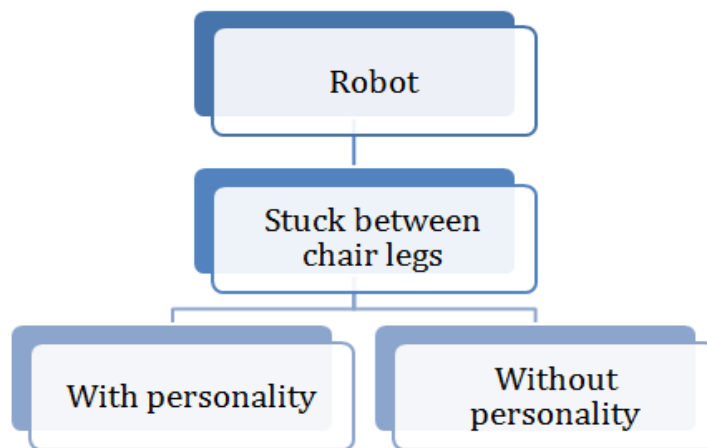
Specification	Is it met?	Note
Proper size	Yes	<i>Hadn't much to change, just not make the body too big.</i>
Five different facial expressions	Yes	<i>The facial expressions can be improved so that they are even more recognizable.</i>
Five different sets of movements	Yes	<i>The movements could be improved and made smoother, more different speeds could be included.</i>
Five different expressions	Yes	
Proper movement speed in general	Yes	
Slow down in personal space of user	Yes	
Use distance sensor	Yes	<i>Sometimes struggles with smaller objects, like chair legs</i>
Use LED matrix	Yes	<i>Drains battery very fast</i>
Sound module	No	<i>There was too little time to include this.</i>

Table 5.1 – Check list of met specifications

As can be seen in the table, all initial specifications were met except for including sound in the robot vacuum cleaner. This was due to too little time. It should however not be too much work to include it after all, whilst it makes a big difference in how the robot is perceived by the user. Sound contributes very well to expressing emotions, just like it does with people. Talking, screaming or laughing for example. Overall is the functional evaluation of the prototype very positive, because nearly all initial specifications are met.

## 5.2 User evaluation

In addition to a functional evaluation, the prototype is also subjected to a user evaluation in which the user experience of the entire prototype is caught. The user evaluation has been done in an interactive setup between the user and developer, in which the developer let the user use the prototype and was to answer questions about it (consult appendix C for the questions). The majority of the participants were between 18 years old and 25 years old and they have all done the test in the same living room. Furthermore the robot was also subjected to a specific staged scenario in which the robot vacuum cleaner required help of the user. Figures 5.2 and 5.3 explain and visualize this scenario. After this scenario the user has to pick a robot which they felt more eager to help. The results of this scenario contribute to answering the research questions, but bear in mind that this scenario is highly context dependent and the results might not be factual.



*Figure 5.2 – Schematic of scenario user evaluation*



*Figure 5.3 – Without personality traits (left), with personality traits (right)*

It is important to say that with the results of the interviews done in the ideation phase, respondents stated that an animal like body (figure 5.4) was preferred as second best appearance. Therefore, the animal like body, which looks like a turtle, has been put to the test as well in the user evaluation. It has no further functionality, despite the form and appearance. This body has been used to test which appearance of the robot is best in this context and situation.



*Figure 5.4 – Animal like body*

#### 5.2.1 Evaluation questions

The questions (appendix C) that were asked to the user concerned multiple relevant aspects of the project; the first category questions are about the design and appearance of the robot. In this category the animal like body is used as well. It is not used in the rest of the user evaluation, because the rest of it only concerns the robotic body. The second and third category of questions was about the movements and expressions of the robot, in whatever form they come. The last two categories of questions were about the overall experience of the user and ideas for future work, on which the user had to give their opinion. In the first four categories the users were also ought to give feedback and suggestions for improvements.

#### 5.2.2 Evaluation method

The user evaluation is based upon multiple qualitative methods; interviewing- and observing the participant while (s)he is using the prototype. The user test is completely focused on testing one user at a time in a 20-minutes session, because if so the user can fully explore the prototype themselves and also the developer only has to ask questions to one person at a time. It is considered to run tests on groups of people, where the robot would be demonstrated to them and they had to answer questions. If this were to be done in an interactive setup, where the people can just say what comes to mind. It is unsure if every individual can give their honest opinion about the prototype, because the group might form a common opinion. In the entire evaluation five users have participated in the evaluation sessions. The questionnaire that they had to fill in can be found in appendix C. As a last thing, which can't be



stressed enough, the evaluation is highly context dependent and the results of it only apply to this situation. If the evaluation were to be done on a bigger scale, with a more diverse test group and more different test environments, the results would be more credible for the prototype.

### 5.2.3 Evaluation results

The answers given to the questions that are posed in the evaluation had resemblances, but when asking for suggestions for improvements; different answers were given. The evaluation results are based upon five user tests with both female- and male students from various studies and varying between the age of 18 and 25. The most interesting results were those from the scenario in which the user had to say which robot he was more eager to help. It was designed to discover whether the added personality traits would increase the likelihood that people want to help the robot or in other words: improve the quality of cooperation between user and robot. The results from this scenario were very promising, because all of the participants preferred helping the robot with personality traits. The answers that were given to the questions about appearance and form mainly stated that round and spherical shapes would improve the design and make it more user-friendly. The robotic body was preferred over the animal body, as was also expected based upon the ideation interview results. Other results were that the movements contribute to better expressing the emotion or state, but do need improvement. Participants liked it when the robot respected their personal space and slowed down within it. The expressions, however, were sometimes misunderstood by the participants. The coughing state was often not understood. The reason for this is presumably that the sound module is not incorporated in the prototype, which is quite an essential part of this expression. The overall expression of the robot was positive, but on the same it might be too distractive according to some of the participants. In the end it does make sound, noises and irregular movements from time to time. Therefore a suggestion was to make less present. The evaluation results are point wise summarized as following:

- Five user test sessions with one user per session.
- Three male users and two female respondents.
- The age of users varied between 18 and 25.
- Scenario pointed out that users were more eager to help the robot with personality traits.
- Users liked the concept of respecting personal space; the robot slowing down within a one meter radius around the user.
- Movements accompanying expressions amplify the expression and make them more vivid. However, the movements seemed somewhat irregular and require improvement. Users often pointed out they liked the expression/movements for frustration or anger.
- The users liked the expressions, but the facial expressions: happiness, coughing and scared were not always clear to the user. Those require improvement. The other two states: frustration and neutral were always understood.

- The sound module has not been included and users pointed out that they would like to see it incorporated in the prototype, because they think it will increase the clarity of expressing an emotion.
- Design wise the users preferred round, spherical and smooth shapes and preferred the robotic body over an animal body. However, this is highly context dependent and only holds within the boundaries of this evaluation.

## Part 6

### **Conclusion**

## Conclusion and future work

---

Based on the experiments, the quality of cooperation between user and robot can be improved by adding personality traits to the robot by: stimulating a more realistic understanding of the actual functionalities and capabilities of the robot vacuum cleaner through the usage of personality traits. With these personality traits the robot is better able to express the state or situation it is currently and therefore more vivid to the user. Personality traits also contribute to forming some sort of bond with the robot; this means that the user will start seeing the robot as an individual rather than just a device. The personality traits that stimulate a better cooperation are the ones which trigger the user in positive way. This means that the personality trait should draw the attention of the user and make them interested in the robot. In general it seemed to be more the positive personality traits, like falling in love or happiness. However, personality traits that are considered to be negative like anger or unhealthiness seem to stimulate a better cooperation as well. This might be because these emotions are ought to be funny or because they are very known and relatable by the user. The interactions that the robot made were found to be good, but certainly not great. Facial expressions were not always recognized, the movements were not very smooth and the sound module is not included after all. Nonetheless, the users understood most of the time the concept and which interaction it was making. All of this information also answers the last question about increasing receptiveness towards robots, because that can be done by: increasing- and improving the interactions between humans and robot, that are perceived as useful, meaningful and fun.

The realized prototype met nearly all initial specifications and is working like it was planned in the first place. There are some minor improvements to meet all the initial specifications, such as including the sound module and make it move more smoothly. Other future work would be to do research to more relevant and good emotions to include, other features to add and more 'efficient-looking' forms. As has been concluded earlier on, the appearance influences how efficient it looks – regardless of how efficient it actually is, but this also means that an efficient appearance increases the quality of cooperation. Therefore this would definitely be a good field to research in the future. Furthermore better evaluations would be future research; with more statistical data of different user groups and tests in different environments with different robot bodies or appearances.

# Part 7

## **Appendices**

Ideation

## **Appendix A**

# Emotions and actions matrix

Character trait	Action
Accessible	On starting up, explain its functions
Active	Physically active
Adventurous	Drive around a lot or say things like 'let's go on an adventure'
Alert	React to sounds or movement and look towards it
Attracted	Be attracted by certain people or objects
Calm	Do not respond to much to environmental changes, just a little
Cheerful	Build an (abstract face) and smile a lot
Confident	Say self-motivational words
Cooperative	Ask for help or offer it
Disciplined	Perform the task, structured and neutral without further ado
Focused	Do not get distracted too often
Friendly	Greet people, apologize sometimes and wait for people when passing
Fun-loving	Ask if the user wants to hear a joke and laugh sometimes
Helpful	Offer help and drive towards the user
Humorous	Make a joke, rotate around, laugh
Intuitive	Make its functions and buttons intuitive
Mannered	Speak with two words
Observing	Add eyes that look around that appear to look around
Popular	Sound: audience clapping and cheering, after doing a task
Respectful	Speak with two words
Secure	Speak with two words
Sexy	Play seducing, sexy music
Simple	Have little extra functionality
Sober	Create a drunk version, than don't perform it
Stable	Work as it is supposed to be
Tolerant	Accept if it is being touched a lot or when it often bumps into things

Amusing	Play music sometimes, do not disturb too much
Cute	Shy, make a blushing face, laugh cute
Dominant	Claim objects, stay around it and defend it
Emotional	Cry, leak water, make sad sounds
Hurried	Fast movements, sharp rotations and quick sounds
Irregular	Make sudden and unexpected movements
Obedient	Wait for a command instead of doing things by himself
Religious	Say a prayer once in a while
Sarcastic	Make sarcastic comments
Sensual	Play sensual music and have red fading lights

Unaggressive	Approach things or people slowly and do not make sudden movements
Unreligious	Curse

Aimless	Drive around aimless, without vacuuming
Aggressive	Speed up towards something and abrupt movements
Angry	Fast movements, a lot of noise, cursing
Anxious	Move backwards a lot
Arrogant	Act very cocky, after getting a command talk like: "pft, is that all. Easy"
Asocial	Drive into things and drive way too fast, even near people or objects
Blunt	Drive into things
Brutal	Drive into things and keep driving
Calculating	Make calculating sounds
Childish	Laugh a lot, round movements, physically active
Clumsy	Bump into things a lot
Confused	Perform the wrong command before doing the right one
Cowardly	Make it scared of objects and flee from them
Crazy	Make unnecessary noises and drive irregular
Criminal	Make it steal something by pushing it away
Difficult	Do not obey commands
Disrespectful	Try to break something by driving into it
Disturbing	Just start brushing or making noise without having to
Hateful	Try to intimidate objects or people
Hesitant	Vacuum and move very slowly
Ignoring	Appear it to see something than obviously ignore it
Miserable	Feel sorry for himself all the time and complain about everything
Neglectful	Miss certain spots on purpose
Obsessive	Claim an object or spot
Opinionated	Dislike or like certain things or people
Perverse	Say dirty words
Possessive	Stay and move around a certain object
Predatory	Hunt moving objects or people
Provocative	Drive against object or people or talk provocatively
Slow	Be very slow
Stubborn	Do something else than is commanded
Suspicious	Drive suspicious around objects or doubt when commands are given
Unhealthy	Coughing
Unlovable	Speak words that he does not feel love
Unreliable	Do nothing when commanded, sometimes
Vulnerable	Scream auwch when it hits something, play breaking sounds



# Questionnaire

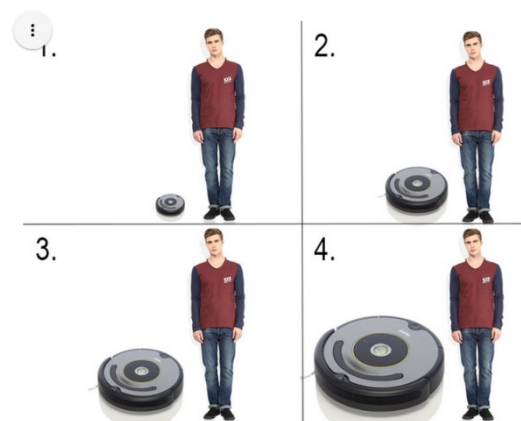
---

## General information

1. What is your gender?
2. What is your age?
3. Do you have children in your house or are you the child?
4. Do you have animals in your house? Caged or free?

## Appearance and form

1. Looking at the image on the right, which size looks most trustworthy?
2. Do you think the appearance of a robot vacuum cleaner influences your opinion about it?
3. Do you think the appearance of a robot vacuum cleaner influences your opinion about how well it could do its cleaning job?
4. Do you think the appearance of a robot vacuum cleaner influences the way in which you interact with the robot?
5. Do you think the appearance of a robot vacuum cleaner influences the quantity in which you want to interact with it?
6. Would you feel comfortable using a robot vacuum cleaner in your house?
7. Would it make you feel more comfortable if this robot vacuum cleaner had a more human like body?
8. Would it make you feel more comfortable if this robot vacuum cleaner had a male or female gender?
9. Would it make you feel more comfortable if this robot vacuum cleaner had a more animal like body?
10. Would it make you feel more comfortable if this vacuum cleaner kept its robot body?
11. Can you think of any other body type that would make you feel more comfortable using it?



## Behavior and movement (rate comfort level from 1 to 5)

1. Would it make you feel comfortable if a robot vacuum cleaner drives through your house rather quick?
2. Do you think a robot vacuum cleaner should approach you slower than it moves around in general?
3. Do you think the behavior of a robot vacuum cleaner would influence your desire to interact with it?

4. Would it make you feel comfortable if the robot vacuum cleaner would come very close to you?
5. Than what would feel like a comfortable distance between you and the robot vacuum cleaner while it is operating?
6. Would it make you feel comfortable if the robot vacuum cleaner tries to avoid you?
7. Would it make you comfortable if the robot vacuum cleaner could talk with you?
8. Do you think it should make a sound after you pressed a button or gave a command? As confirmation he understood it.
9. Shortly describe how you think you could cooperate with a robot vacuum cleaner, if possible.

**Emotions and expressions** (rate likability from 1 to 10)

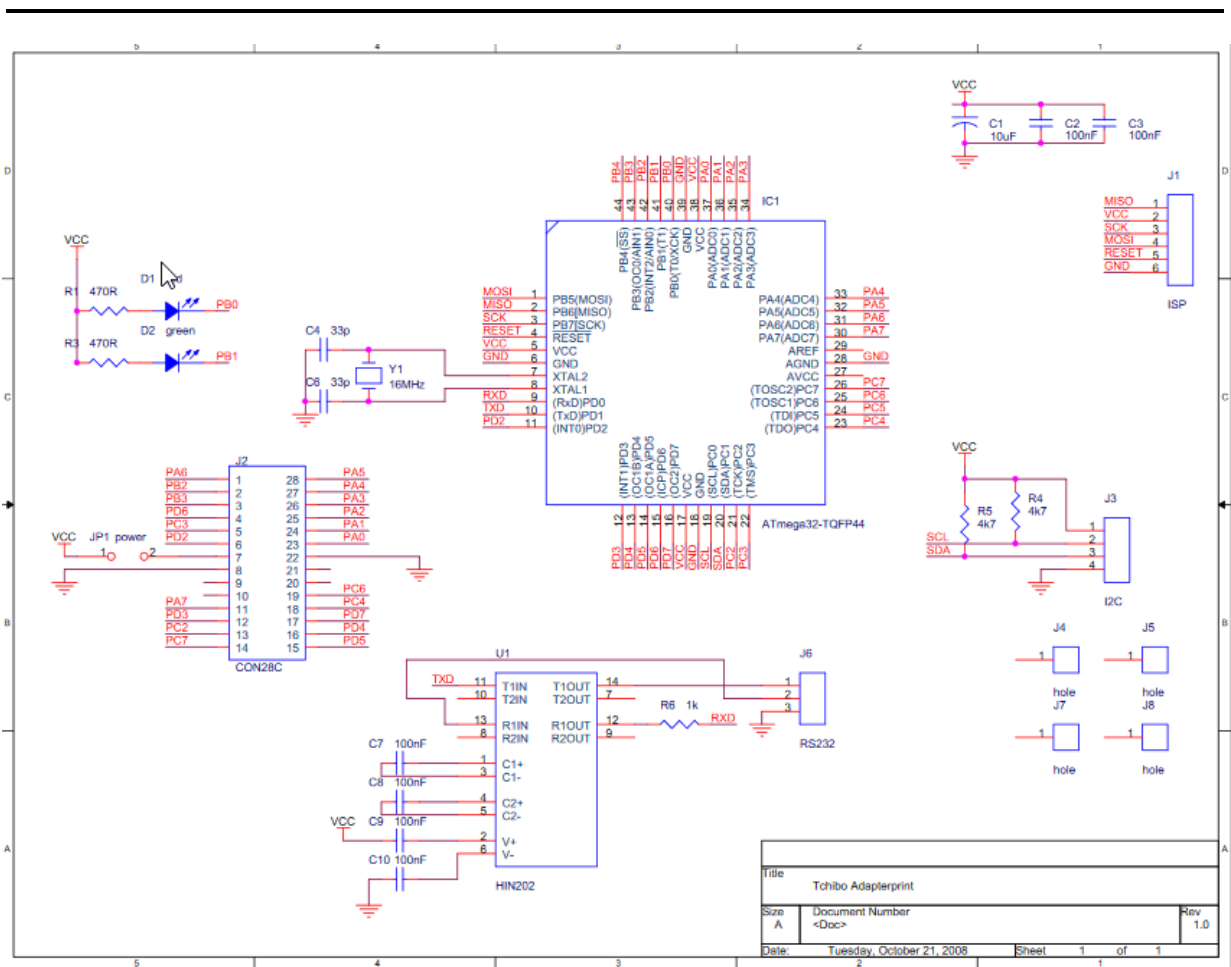
1. How likeable would you rate a robot vacuum cleaner that gets frustrated sometimes?
2. How likeable would you rate a robot vacuum cleaner that is happy all the time?
3. How likeable would you rate a robot vacuum cleaner that is scared of objects in your house?
4. How likeable would you rate a robot vacuum cleaner that is shy?
5. How likeable would you rate a robot vacuum cleaner that yells at things when it drives into something?
6. How likeable would you rate a robot vacuum cleaner that sometimes refuses to listen, like a rebel teenager?
7. How likeable would you rate a robot vacuum cleaner that would fall in love with something in your house?
8. How likeable would you rate a robot vacuum cleaner that would be ashamed sometimes?
9. How likeable would you rate a robot vacuum cleaner that would become sad sometimes?
10. How likeable would you rate a robot vacuum cleaner that coughs if your floor is filthy while vacuuming?

**User and control** (rate agreement from 1 to 5)

1. Do you think robots always do what people want them to do?
2. Do you sometimes do not understand why a robot is not functioning well?
3. If a robot is not functioning well, does it provide enough feedback about why it doesn't?
4. If a robot is not functioning well, do you know how to make it function well?
5. Do you think that if a robot has a personality you could understand better why it is not working?
6. Do you think that if you have more control over the robot, it is less able to express its own personality?
7. If you would own a robot with its own personality, into what extend would you like to have control over its personality and expressing of it?

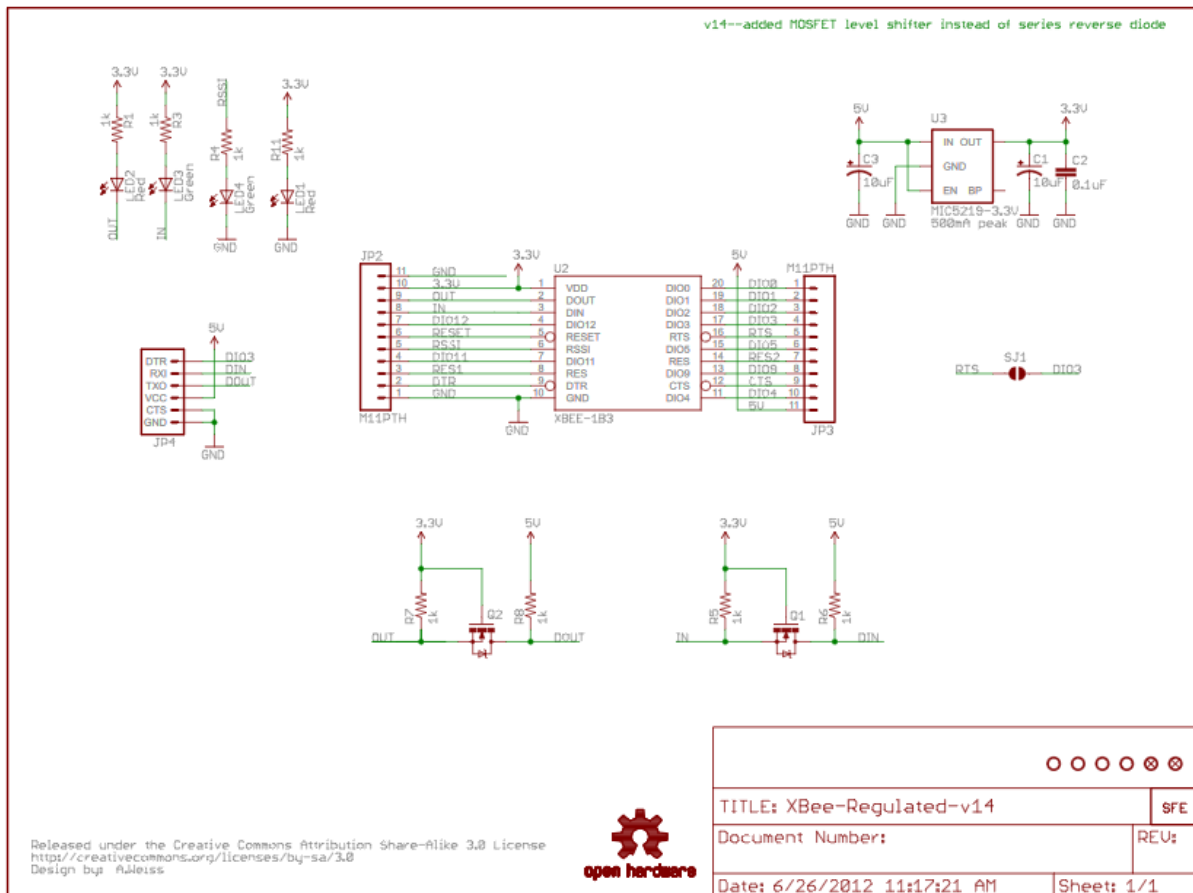
# Realization

## **Appendix B**



Source: [Edwindertien.nl/doku](http://Edwindertien.nl/doku)

# System specifications XBee



Source: [Sparkfun](http://www.sparkfun.com)

# User Test I – Recognizing movements and faces

---

Complete the date: ... / 06 / 2017

The following hyperlinks are directing you to YouTube videos please watch them and fill in the forms that are called 'linking movements open', then look at the five images and fill in what you think you see (there are 4 emotions and 1 action). At last you have to fill in the form 'linking movements closed'. In order to secure honest answers, please **answer and read everything in order from top to bottom**.

Movement Set 1

<https://www.youtube.com/watch?v=M8J5ZY0zRE0&feature=youtu.be>

Movement Set 2

<https://www.youtube.com/watch?v=1Crt1BFevRQ&feature=youtu.be>

Movement Set 3

[https://www.youtube.com/watch?v=e7ifm\\_xhuQs&feature=youtu.be](https://www.youtube.com/watch?v=e7ifm_xhuQs&feature=youtu.be)

Movement Set 4

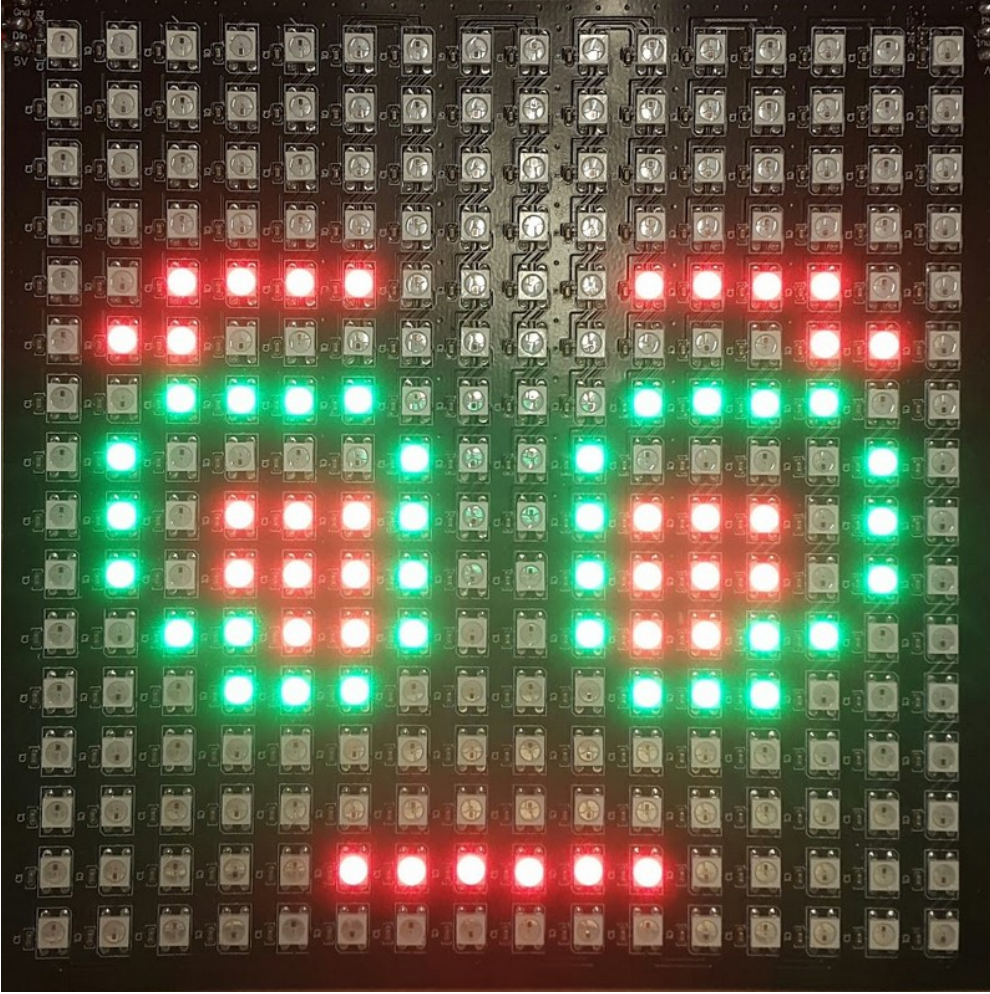
<https://www.youtube.com/watch?v=9vlzzOyhshc&feature=youtu.be>

Movement Set 5

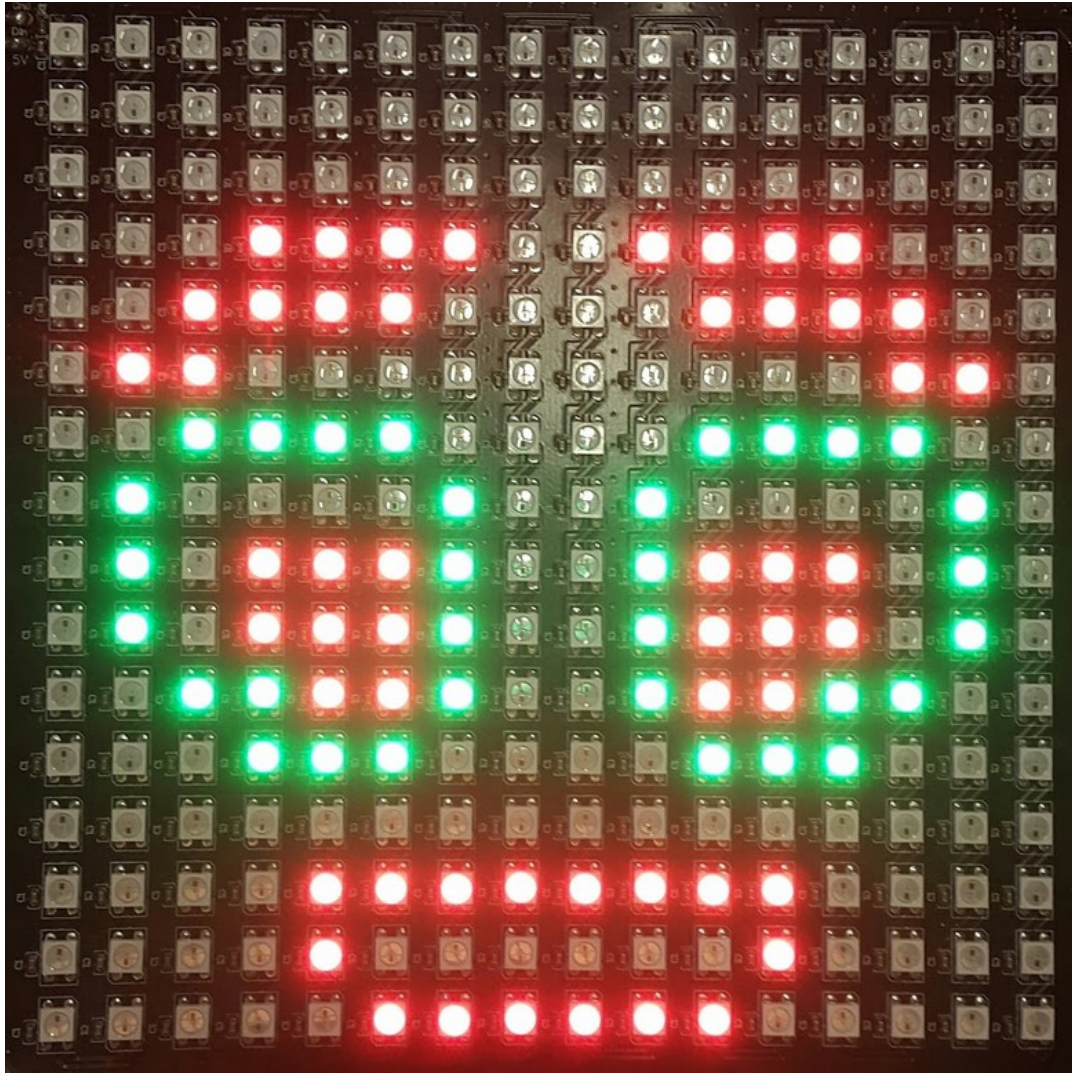
<https://www.youtube.com/watch?v=D2piT2mn8H8&feature=youtu.be>

Linking movements open		
Write down the emotion you think that accompanies this movement		
Movement set 1	I think this is...	[My answer]
Movement set 2	I think this is...	[My answer]
Movement set 3	I think this is...	[My answer]
Movement set 4	I think this is...	[My answer]
Movement set 5	I think this is...	[My answer]

Choose one		
Slowing down distance	The distance is...	Too far / just fine / too close
Approaching speed	The speed is...	Too fast / just fine / too slow

Recognizing facial expressions	
Write down for the following five images which emotion or action you see	
	
This is	[My answer]

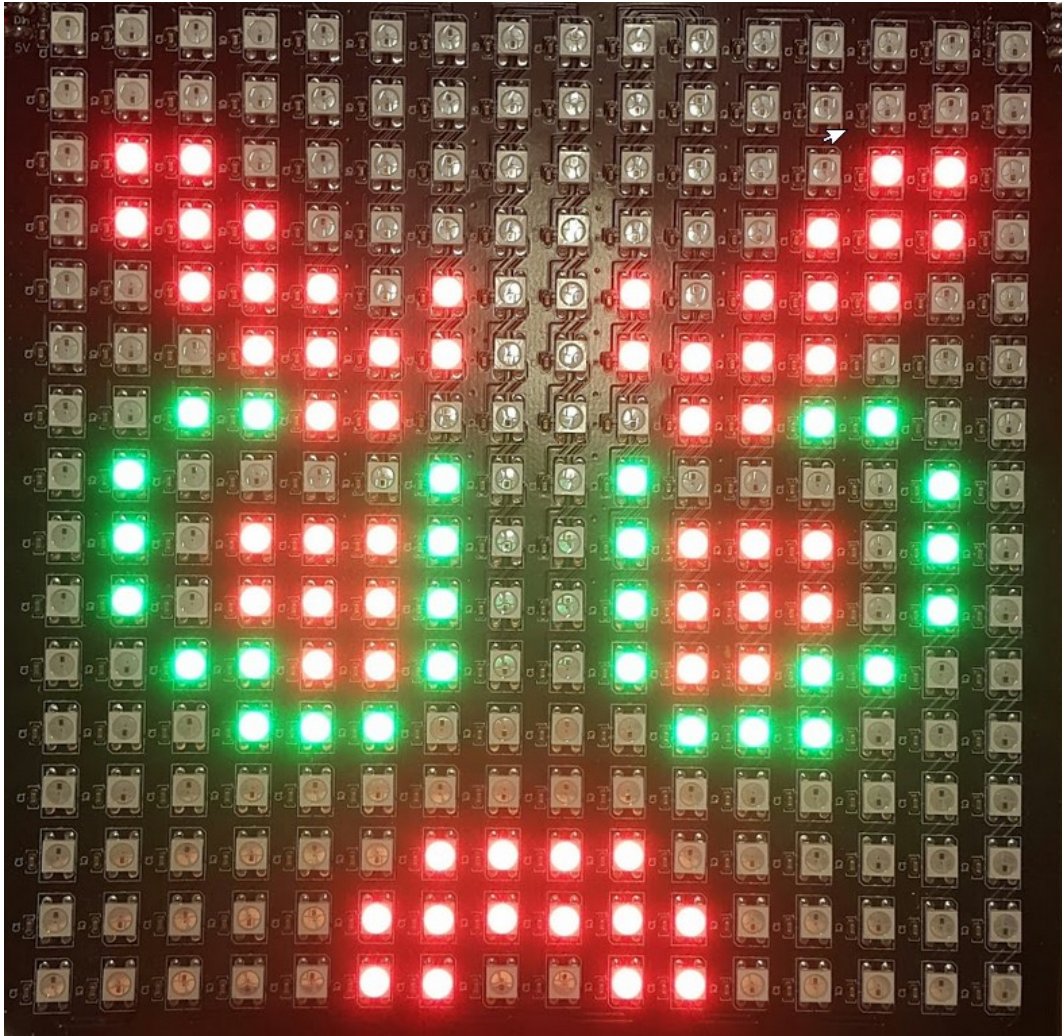




This is

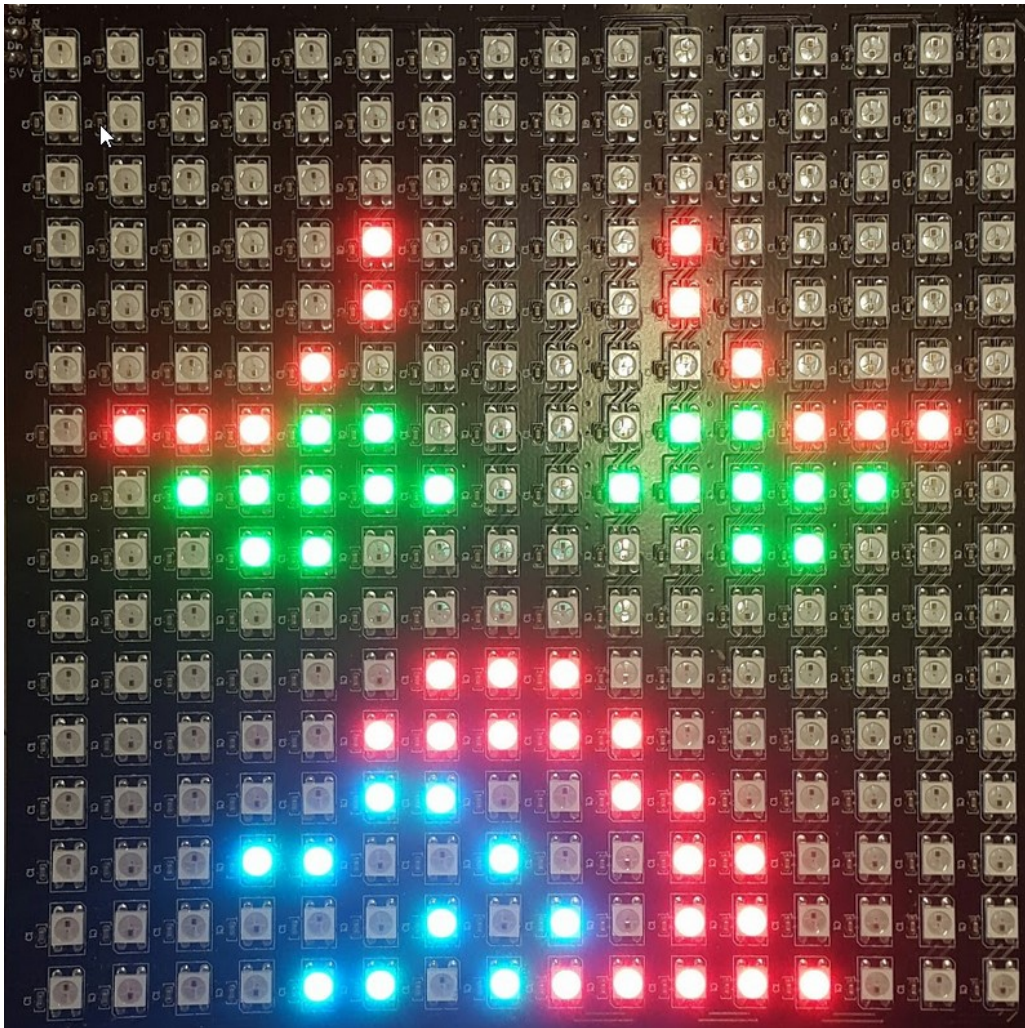
[My answer]





**This is**

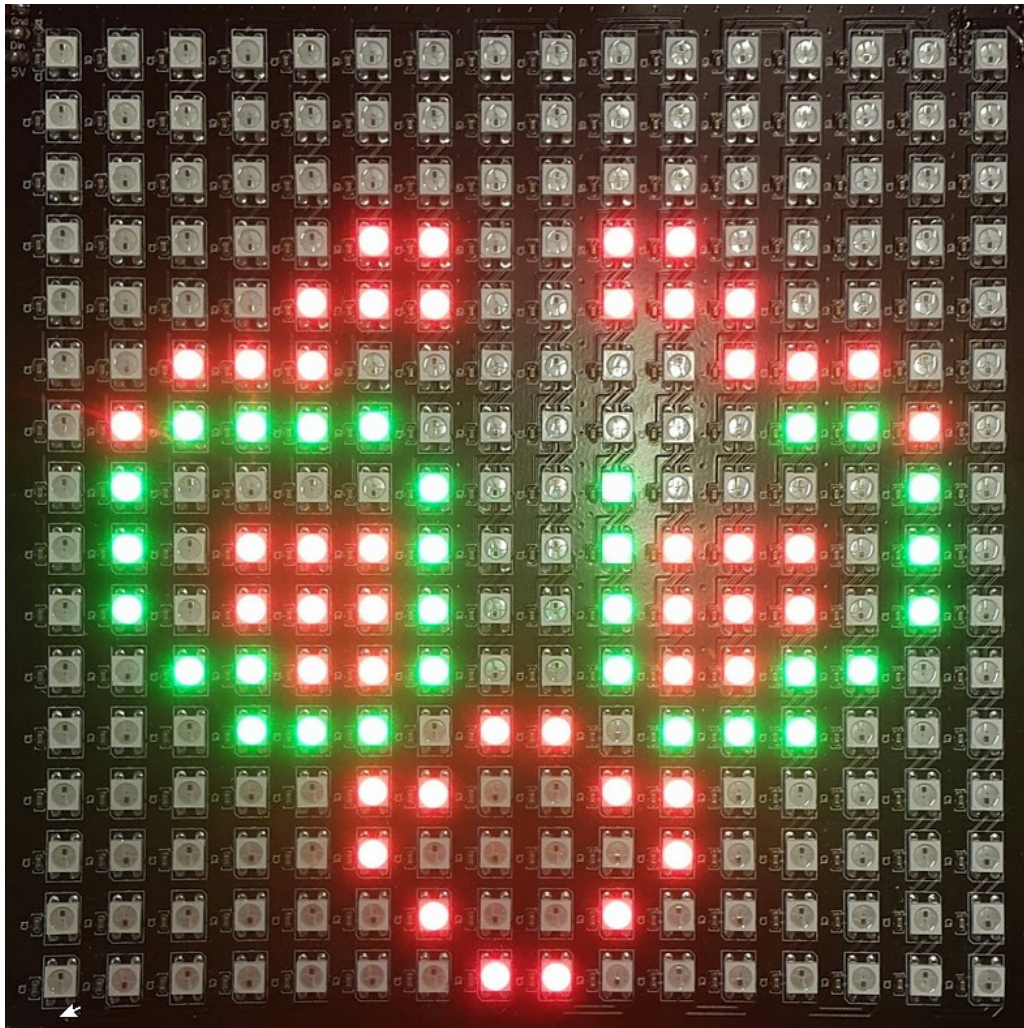
[My answer]



This is

[My answer]





This is

[My answer]

**PLEASE DO NOT READ THIS BEFORE YOU'VE ANSWERED THE PREVIOUS QUESTIONS**

Linking movements closed		
<p>Write down the emotion you think that accompanies this movement</p> <p>You can choose from the following five emotions: angry, coughing*, happy, neutral, scared</p>		
Movement set 1	I think this is...	[My answer]
Movement set 2	I think this is...	[My answer]
Movement set 3	I think this is...	[My answer]
Movement set 4	I think this is...	[My answer]
Movement set 5	I think this is...	[My answer]
Circle your answer		
Slowing down distance	The distance is...	Too far / just fine / too close
Approaching speed	The speed is...	Too fast / just fine / too slow

## XBee code

---

```
void setup() {
  Serial.begin(9600);
  xbee.begin(9600);                      // Start xbee at 9600 bps

  //delay(1000);
  //Serial.println("We are ready to operate, controls are ADSW to
  steer. O, P, B, C, V, T, I & H are other buttons.");
  //Serial.println("");
}

void loop() {
  // Keep asking for information updates
  if (!askedForStatus) {
    xbee.print("i");
    xbee.write(0x0D);
    xbee.write(0x0A);
    timeMillis = millis();
    askedForStatus = true;
    delay(50);
  }

  if (askedForStatus) {
    Serial.println("receiving...");
    // Read and print xbee data
    String statusMessage = xbee.readString();
    Serial.println(statusMessage);
    int statusMessageLength = statusMessage.length();
    Serial.println(statusMessageLength);

    // Check response time and print it
    timeMillisDelta = millis();
    unsigned long responseTime = timeMillisDelta - timeMillis;
    //Serial.print("Response time (ms): ");
    //Serial.println(responseTime);
    //Serial.println("---");

    /*
    // Print the received string and string length
    Serial.println(statusMessage);
    Serial.println("---");
    Serial.print("String length: ");
    Serial.println(statusMessage.length());
    Serial.println("---");
    */
  }
}
```

```

    // Convert to individual chars
    char firstbyte = byte(statusMessage.charAt(statusMessageLength -
11));
    char secondbyte = byte(statusMessage.charAt(statusMessageLength
- 10));
    char thirdbyte = byte(statusMessage.charAt(statusMessageLength -
8));
    char fourthbyte = byte(statusMessage.charAt(statusMessageLength
- 7));

    // Convert seperate the two HEX to a DEC
    int b1 = hexchartoint(firstbyte);
    int b2 = hexchartoint(secondbyte);
    int b3 = hexchartoint(thirdbyte);
    int b4 = hexchartoint(fourthbyte);

    byteone = combinetwohex(b1, b2);
    bytetwo = combinetwohex(b3, b4);

    if (byteone && byteone != -12288) {
        if (bitRead(byteone, WHEELS) == 1)    Serial.println("WHEELS");
        if (bitRead(byteone, FLOOR_3) == 1)    Serial.println("FLOOR
3");
        if (bitRead(byteone, FLOOR_2) == 1)    Serial.println("FLOOR
2");
        if (bitRead(byteone, FLOOR_1) == 1)    Serial.println("FLOOR
1");
        if (bitRead(byteone, FLOOR_4) == 1)    Serial.println("FLOOR
4");
        if (bitRead(byteone, BUTTON_S) == 0) Serial.println("BUTTON
S");
    }

    if (bytetwo && bytetwo != -12288) {
        if (bitRead(bytetwo, BUMPER_L) == 1) Serial.println("BUMPER
LEFT");
        if (bitRead(bytetwo, BUMPER_R) == 1) Serial.println("BUMPER
RIGHT");
        if (bitRead(bytetwo, BUTTON_M) == 0) Serial.println("BUTTON
M");
        if (bitRead(bytetwo, BUTTON_L) == 0) Serial.println("BUTTON
L");
        if (bitRead(bytetwo, WALL) == 1)      Serial.println("WALL");
    }

    // Add a new line and set the request for the status update to
false

```

```

    //Serial.println("");
    askedForStatus = false;
}

// Send out a request for a status message every 1100 ms
//statusMess();
//delay(1100);

// Method to control the vacuum cleaner with the serial port
if (Serial.available()) {
    char incomingChar = Serial.read();
    //Serial.print(incomingChar);
    switch (incomingChar) {
        case 'w':
            moveForward();
            break;
        case 's':
            moveBackwards();
            break;
        case 'a':
            turnLeft();
            break;
        case 'd':
            turnRight();
            break;
        case ' ':
            doNothing();
            break;
        case 'b':
            brush();
            break;
        case 'n':
            brushStop();
            break;
        case 't':
            initTone();
            break;
        case 'i':
            statusMess();
            break;
        case 'o':
            encoderLeft();
            break;
        case 'p':
            encoderRight();
            break;
        case 'h':

```

```

        displayText();
        break;
    case 'v':
        verbose_();
        break;
    case 'c':
        verbose_off();
        break;
    }
}

// WRITING DATA TO THE ROBOT VIA XBEE SOFTWARE SERIAL PORT
void statusMess() {
    Serial.println("STATUS MESSAGE");
    Serial.println("-----");

    xbee.print("i");
    xbee.write(0x0D);
    xbee.write(0x0A);
    timeMillis = millis();
    askedForStatus = true;
    //delay(1500);
}

int hexchartoint(char hex) {
    if (hex >= '0' && hex <= '9') {
        return hex - '0';
    }
    if (hex == 'A') return 10;
    if (hex == 'B') return 11;
    if (hex == 'C') return 12;
    if (hex == 'D') return 13;
    if (hex == 'E') return 14;
    if (hex == 'F') return 15;
}

int combinethwohex(int b1_, int b2_) {
    int combinedhex = (b1_ * 16) + b2_;
    return combinedhex;
}

```



## NRF24L01 Transmitter code

---

```
void setup(void) {
  Serial.begin(9600);
  radio.begin();
  radio.setDataRate( RF24_250KBPS );
  radio.enableAckPayload();

  radio.openWritingPipe(pipe);
  radio.openReadingPipe(1, pipe1);
  radio.startListening();

  delay(50);
  Serial.println("Ready\n");
}

void loop(void) {

  if (Serial.available() > 0) {
    char incomingChar = Serial.read();
    switch (incomingChar) {
      case 'w':
        theMessageToSend = "UP\n";
        messageSize = theMessageToSend.length();
        Serial.println("SERIAL UP");

        sendData();
        break;
      case 's':
        theMessageToSend = "DOWN\n";
        messageSize = theMessageToSend.length();
        Serial.println("SERIAL DOWN");
        sendData();
        break;
      case 'a':
        theMessageToSend = "LEFT\n";
        messageSize = theMessageToSend.length();
        Serial.println("SERIAL LEFT");

        sendData();
        break;
      case 'd':
        theMessageToSend = "RIGHT\n";
        messageSize = theMessageToSend.length();
        Serial.println("SERIAL RIGHT");
```

```

        sendData();
        break;
    }
    delay(100);
}

}

void sendData() {
    // SEND DATA
    for (int i = 0; i < messageSize; i++) {
        int charToSend[1];
        charToSend[0] = theMessageToSend.charAt(i);
        radio.write(charToSend, 1);

        if ( radio.isAckPayloadAvailable() ) {
            radio.read(ackMessg, ackMessgLen);
            Serial.print("Acknowledge received: ");
            Serial.println(ackMessg[0]);
        }
    }
    //send the 'terminate string' value...
    msg[0] = 2;
    radio.write(msg, 1);
}

```

## NRF24L01 Receiver code

---

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

  radio.begin();
  radio.setDataRate( RF24_250KBPS );
  radio.enableAckPayload();

  radio.openWritingPipe(pipe1);
  radio.openReadingPipe(1, pipe);
  radio.startListening();
  //radio.enableAckPayload();
  //radio.writeAckPayload(1, ackData, ackLen);

  delay(50);
  Serial.println("Ready\n");
}

void loop(void) {

  // RECEIVE DATA
  if (radio.available()) {
    bool done = false;
    done = radio.read(msg, 1);
    char theChar = msg[0];
    if (msg[0] != 2) {
      theMessage.concat(theChar);
      delay(1);
      radio.writeAckPayload(1, ackData, ackLen);
      Serial.println(ackData[0]);
      ackData[0] += valChange; // this just increments so you can
see that new data is being sent
    }
    else {
      Serial.print("\nRECEIVED: ");
      Serial.println(theMessage);
      Serial.println("-----\n");
      theMessage = "";
    }
  }
}
```

## Final code (most parts)

---

```
void setup() {
  Serial.begin(9600);
  Roomba.begin(9600);
  strip.begin();

  // PinModes for distance sensor
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  // Init timers
  timer = millis();
  prevTimer = millis();
  prevTimer2 = millis();

  // To empty incoming data at the start
  for (int i = 0; i < 100; i++) {
    if (Roomba.available()) {
      char incomingDataToEmpty = Roomba.read();
      Serial.println(incomingDataToEmpty);
    }
  }

  // RESET LEDS
  for (int i = 0; i < 256; i++) {
    strip.setPixelColor(i, 0, 0, 0);
  }

  strip.show();
  neutralState();

  Serial.println("Ready\n");
}

void loop() {
  timer = millis();

  // ASK FOR A STATUS UPDATE EVERY 100 MILLISECONDS AND IF WE ARE
  // ALLOWED TO ASK FOR ONE
  if (timer - prevTimer > statusUpdateLoop && ackStatusUpdate ==
true) {
    // Read the distance
    getDistance();
    // Request status update and reset the timer
    prevTimer = timer;
  }
}
```

```

    RequestStatusUpdate();
    ackStatusUpdate = false;
}

// IF WE MAKE A TIMEOUT AND HAVE NO STATUS UPDATE AFTER 3 SECONDS,
REQUEST THEM AGAIN
if (timer - prevTimer > 2500 && ackStatusUpdate == false) {
    // Stop driving, and allow status updates to be requested again
    stopDriving();
    ackStatusUpdate = true;
}

// 5 SECONDS AFTER ANOTHER STATE IS PERFORMED, RETURN BACK TO THE
NEUTRAL STATE
if (timer - prevTimer2 > 5000) {
    // Set it back to neutral state, instruct the robot and reset
timer
    stateCycle = 1;
    controlStates(stateCycle, -1);
    prevTimer2 = timer;
}

// IF THERE IS DATA AVAILABLE FROM THE ROBOT VACUUM CLEANER, READ
IT
if (Roomba.available()) {
    incomingChar = Roomba.read();

    // IF IT IS NOT THE LAST SYMBOL OF A STATUS UPDATE, READ THE
CHARACTER
    if (incomingChar != '>') {
        incomingString.concat(incomingChar);
    }
    else {
        // FILTER OUT THE STATUS UPDATE, BY CHECKING FOR AN 'i'
CHARACTER
        char firstChar = incomingString.charAt(0);

        if (firstChar == 'i') {
            if (stopped)    stopDriving();
            if (!stopped)  alwaysDriveForwardAtVariableSpeed();
            DataHandling(incomingString);

            // If we are done with reading and executing the data,
accept status updates again
            ackStatusUpdate = true;
        }
        // Empty the string to put newly received data in it
        incomingString = "";
    }
}

```

```

    }
}

void alwaysDriveForwardAtVariableSpeed() {
    int distSpeed = distance * 4;

    //Serial.print("Distance : ");
    //Serial.println(distance);

    String mr = "mr ";
    String ml = "ml ";

    // Keep it within bounds
    if (distSpeed > 255) distSpeed = 255;
    if (distance < 50)    distSpeed = 150;
    if (distance < 10)   distSpeed = 100;

    mr.concat(distSpeed);
    ml.concat(distSpeed);

    //Serial.println(mr);
    //Serial.println(ml);

    driveForwardVariable(mr, ml);
}

void getDistance() {
    // Send out a pulse, receive it and calculate distance
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);

    duration = pulseIn(echoPin, HIGH);
    distance = (duration / 2) / 29.1;
    if (distance > 100) distance = 100;
    if (distance < 0)    distance = 0;
    Serial.println(distance);
}

void DataHandling(String incomingString_) {

    StringLength = incomingString_.length();

    char firstbyte = byte(incomingString_.charAt(StringLength - 10));
    char secondbyte = byte(incomingString_.charAt(StringLength - 9));
    char thirdbyte = byte(incomingString_.charAt(StringLength - 7));

```

```

char fourthbyte = byte(incomingString_.charAt(StringLength - 6));

// Convert separte the two HEX to a DEC
int b1 = hexchartoint(firstbyte);
int b2 = hexchartoint(secondbyte);
int b3 = hexchartoint(thirdbyte);
int b4 = hexchartoint(fourthbyte);

byteone = combinetwohex(b1, b2);
bytetwo = combinetwohex(b3, b4);

// DO NOTHING WITH THE BYTES THAT ARE SEND IF THE ROOMBA IS TURNED
OFF
if (byteone >= 0 && byteone <= 255 && bytetwo >= 0 && bytetwo <=
255) {
    if (byteone != 244 && bytetwo != 172) {

        // IF THE S-BUTTON IS PRESSED, STOP DRIVING
        if (byteone && byteone != -12288) {
            if (bitRead(byteone, BUTTON_S) == 0) {
                Serial.println("BUTTON S");
                stopped = false;
                stateCycle = 2;
                controlStates(stateCycle, -1);
                prevTimer2 = timer;
            }
        }

        // IF THE M-BUTTON IS PRESSED, START DRIVING, IF THE L-BUTTON
        IS PRESSED SWITCH STATES
        if (bytetwo && bytetwo != -12288) {
            if (bytetwo && bytetwo != -12288) {
                if (bitRead(bytetwo, BUMPER_L) == 1) {
                    Serial.println("BUMPER LEFT");
                    int bumper = 0;
                    stateCycle = random(1, 6);
                    controlStates(stateCycle, bumper);
                    ackStatusUpdate = true;
                    prevTimer2 = timer;
                }

                else if (bitRead(bytetwo, BUMPER_R) == 1) {
                    Serial.println("BUMPER RIGHT");
                    int bumper = 1;
                    stateCycle = random(1, 6);
                    controlStates(stateCycle, bumper);
                    ackStatusUpdate = true;
                    prevTimer2 = timer;
                }
            }
        }
    }
}

```

```

    }
}
if (bitRead(bytetwo, BUTTON_M) == 0) {
    Serial.println("BUTTON M");
    stopped = true;
    stateCycle = 1;
    controlStates(stateCycle, -1);
}
if (bitRead(bytetwo, BUTTON_L) == 0) {
    Serial.println("BUTTON L");
    stopped = false;
    stateCycle = 2;
    controlStates(stateCycle, -1);
    prevTimer2 = timer;
}
}
}
}
}

```

```

////////// OTHER FUNCTIONS //////////
////////////////////////////////////

```

```

int hexchartoint(char hex) {
    if (hex >= '0' && hex <= '9') {
        return hex - '0';
    }
    if (hex == 'A') return 10;
    if (hex == 'B') return 11;
    if (hex == 'C') return 12;
    if (hex == 'D') return 13;
    if (hex == 'E') return 14;
    if (hex == 'F') return 15;
}

```

```

int combinetwohex(int b1_, int b2_) {
    int combinedhex = (b1_ * 16) + b2_;
    return combinedhex;
}

```

```

void RequestStatusUpdate() {
    Roomba.write("i");
    delay(minimalDelay);
    Roomba.write(0x0D);
    delay(minimalDelay);
    Roomba.write(0x0A);
}

```



```

void driveForwardVariable(String mr_, String ml_) {

    Roomba.print(mr_);
    delay(minimalDelay);
    Roomba.write(0x0D);
    delay(minimalDelay);
    Roomba.write(0x0A);
    delay(minimalDelay);

    Roomba.print(ml_);
    delay(minimalDelay);
    Roomba.write(0x0D);
    delay(minimalDelay);
    Roomba.write(0x0A);
    delay(minimalDelay);
}

void controlStates(int stateCycle_, int bumper_) {

    // NEUTRAL
    if (stateCycle_ == 1) {
        neutralState();
        initTone(stateCycle_);
        prevTimer2 = millis();

        // DO NOTHING (ONLY THE L-BUTTON IS PRESSED)
        if (bumper_ == -1) ;

        // LEFT BUMPER HIT
        if (bumper_ == 0) {
            turnRight(200, 200);
            delay(450);
        }
        // RIGHT BUMPER HIT
        if (bumper_ == 1) {
            turnLeft(200, 200);
            delay(450);
        }
        ackStatusUpdate = true;
    }

    // THE OTHER 4 CODES FOR THE OTHER STATES ARE THE SAME AS NEUTRAL,
    // Although it makes some different movements, therefore it is not
    // copied as well

}

```

## 69

# Evaluation

## **Appendix C**

# Interactive user test

---

Date: ... / ... / 2017

Participant number:

This user test session is to test the user experience with my modified robot vacuum cleaner. It has been modified so that it has personality traits, which it has to express. The overall goal of this project is to test whether personality traits on robots contribute to the quality of cooperation between the user and robot. The test will take about 10-15 minutes and will be done in an interactive setting between the participant and tester. It is an anonymous test; your answers will only be used to analyze the robot. If you agree to take part of this user test, please sign this form here down below.



Robot vacuum cleaner, unmodified

I hereby declare to take part in this user test and agree with the terms.

Signature: .....

Body design	Answer
What do you think of the robotic body?	
Would you recommend any design changes?	
What do you think of the animal body?	
Would you recommend any design changes?	
Which design do you prefer to use for daily use?	

Movement	Answer
Do you recognize that each expression is accompanied by different movements?	
Do you think they contribute to better express the expression?	
What improvements would you suggest for the movements?	

<b>What do you think about the speed it is driving at?</b>	
<b>Do you find it pleasant it slows down when it approaches you?</b>	

Expressions	Answer
<b>Which expressions do you see?</b>	
<b>Do you like the facial expressions?</b>	
<b>Do you think they contribute well to expressing personality?</b>	
<b>Which improvements would you suggest for the facial expressions?</b>	

Overall experience	Answer
<b>What is your overall experience with the robot vacuum cleaner?</b>	
<b>Would you feel comfortable using such a device for personal usage?</b>	

<b>Do you think you would interact with it more if it has such a personality?</b>	
<b>Do you think it would improve the quality of cooperation between you and the robot?</b>	

<b>Future work</b>	<b>Answer</b>
<b>Name some personality traits that would be nice to add</b>	
<b>How would you incorporate those?</b>	

## **Bibliography**



# Bibliography

---

- [1] T. Fong, I. Nourbakhsh, and K. Dautenhahn, "A survey of socially interactive robots," *Rob. Auton. Syst.*, vol. 42, no. 3–4, pp. 143–166, 2003.
- [2] C. F. DiSalvo, F. Gemperle, J. Forlizzi, and S. Kiesler, "All robots are not created equal: the design and perception of humanoid robot heads," *Conf. Des. Interact. Syst. Process. Pract. methods Tech.*, vol. pages, pp. 321–326, 2002.
- [3] A. G. Billard and K. Dautenhahn, "Grounding communication in situated, social robots," *Proc. TIMR*, vol. 97, no. April, pp. 97–99, 1997.
- [4] B. Adams, C. Breazeal, R. A. Brooks, B. Scassellati, and M. I. T. Artificial, "Humanoid Robots : A New Kind of Tool EXPLORE THEORIES OF HUMAN INTELLIGENCE," no. August, pp. 1–6, 1993.
- [5] M. Harbers, M. M. M. Peeters, and M. A. Neerincx, *Perceived autonomy of robots: Effects of appearance and context*, vol. 84. 2017.
- [6] A. Tapus, E. André, J. M. F. Ferland, M. A. Eds, and R. Goebel, "Social Robotics," vol. 1, pp. 401–410, 2015.
- [7] T. Nomura, T. Kanda, and T. Suzuki, "Experimental investigation into influence of negative attitudes toward robots on human-robot interaction," *AI Soc.*, vol. 20, no. 2, pp. 138–150, 2006.
- [8] C. Ray, F. Mondada, and R. Siegwart, "What do people expect from robots?," *2008 IEEE/RSJ Int. Conf. Intell. Robot. Syst. IROS*, pp. 3816–3821, 2008.
- [9] B. Hendriks, B. Meerbeek, S. Boess, S. Pauws, and M. Sonneveld, "Robot vacuum cleaner personality and behavior," *Int. J. Soc. Robot.*, vol. 3, no. 2, 2011.
- [10] L. Oestreicher and K. S. Eklundh, "User expectations on human-robot co-operation," *Proc. - IEEE Int. Work. Robot Hum. Interact. Commun.*, pp. 91–96, 2006.
- [11] M. M. A. De Graaf and S. Ben Allouch, "The evaluation of different roles for domestic social robots," in *Proceedings - IEEE International Workshop on Robot and Human Interactive Communication*, 2015, vol. 2015–Novem.
- [12] L. Oestreicher, "Cognitive, social, sociable or just socially acceptable robots?," in *Proceedings - IEEE International Workshop on Robot and Human Interactive Communication*, 2007.
- [13] R. H. Cuijpers, M. T. Bruna, J. R. C. Ham, and E. Torta, *Attitude towards robots depends on interaction but not on anticipatory behaviour*, vol. 7072 LNAI. 2011.
- [14] J. Fink, V. Bauwens, F. Kaplan, and P. Dillenbourg, "Living with a Vacuum Cleaning Robot: A 6-month Ethnographic Study," *Int. J. Soc. Robot.*, vol. 5, no. 3, 2013.
- [15] V. Evers, "HMI Researcher." [Online]. Available: <https://vanessaevers.wordpress.com/>.
- [16] L. Samsung Electronics Co. and T. S. Kim, "Robot cleaner," 1995. [Online]. Available:

<https://www.google.com/patents/US5440216>.

- [17] K. Stroeken, "Vision controlled robot swarm," 2012. [Online]. Available: <https://www.ram.ewi.utwente.nl/people/details/kees-stroeken.html>.
- [18] E. Dertien, "Tchibo module," 2008. [Online]. Available: [http://wiki.edwindertien.nl/doku.php?id=modules:tchibo&s\[\]=vacuum](http://wiki.edwindertien.nl/doku.php?id=modules:tchibo&s[]=vacuum).