Delft University of Technology


Picture by Hani Al-Ers

Smart Teddy: Elderly monitoring and support system using ambient intelligence

Human Interaction and Integration

BSc Thesis

Shea Haggerty and Laura Croes

Smart Teddy: Elderly monitoring and support system using ambient intelligence

Human Interaction and Integration

by

Shea Haggerty and Laura Croes

Students:	Shea Haggerty	4713249
	Laura Croes	4534611
Supervisors:	Dr. Zaid Al-Ars	Associate Professor the Delft University of Technology
	Dr. Hani Al-Ers	Researcher at the Hague University of Applied Sciences
Thesis Committee:	Dr. Prof. Catholijn M. Jonker	Full professor at the Delft University of Technology
	Dr. Matthias Möller	Associate Professor at the Delft University of Technology
	Dr. Zaid Al-Ars	Associate Professor the Delft University of Technology
	Dr. Hani Al-Ers	Researcher at the Hague University of Applied Sciences
Institution:	Delft University of Technology Faculty of Electrical Engineering, Mathematics and Computer Science, Delft	
Defence:	09:00-10:30, 29th of June, 2021.	

Preface

In September 2018, the Smart Teddy project was founded by a group of researchers within the Hague University of Applied Sciences[1] in the Netherlands. The Smart Teddy project is a multidisciplinary project aiming to create an interactive system, using a teddy bear as a focus point, which collects the data needed in order to enable seniors with dementia to live independently for a longer period of time. A visual of the functionality of the Smart Teddy is pictured in Figure 1. Over the last three years, three prototypes of the Smart Teddy have been developed. The Smart Teddy project was introduced as a final project for students following the BSc program Electrical Engineering at the Delft University of Technology[2]. Starting in April 2021, a team of six students attempted to further develop the Smart Teddy over the course of 11 weeks.

This thesis contains the Human Interaction & Integration sub-domain of the Smart Teddy thesis project, where Human Interaction refers to the aspects of the Teddy that encourage interaction with the user, and Integration refers to the combination of all sub-domains into one fully functioning prototype. In this thesis, the design choices, implementation methods and verification are discussed. The contribution to the prototype regarding Human Interaction & Integration are the addition of a movement system using pneumatics, the implementation of a flexible touch sensor, the ability for the Teddy to produce audio, to communicate wirelessly with the Base Station, and for all components in the Teddy to communicate with the main controller.

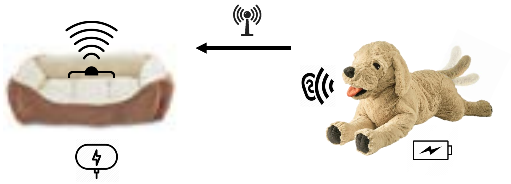


Figure 1: Visual of the Smart Teddy.

The final prototype has been implemented using the Raspberry Pi Pico microcontroller, which was mounted on a custom PCB. All controls are provided by the Pico, and uses I2C, SPI, UART, and analog and digital inputs to communicate with the sensors and actuators. These sensors and actuators were implemented using off-the-shelf breakout boards and drivers, to allow for fast design- and test iterations. The movement of the Teddy has been implemented using air pumps and molded silicon rubber, and the tail wagging is implemented using the same principle used for soft robotic grippers. The final prototype is fully functional and meets 16 of the 20 requirements - the requirement concerning speech recognition and the noise produced by the pumps have not been met.

We would like to acknowledge and thank our supervisors, Zaid and Hani, for their ongoing support and feedback we received during the project. We would also like to thank Ezra and Annemarie, who were so kind as to meet with us and share their extensive knowledge on people suffering from dementia - your input gave us extremely valuable insights and your enthusiasm on the topic was inspiring. We owe a big thank you to Madelene, for not only giving birth to one of the authors of this thesis, but also for her large contribution to the design and development of the physical structure needed for the Teddy, which required a lot of creativity and experience with a sowing machine. And last but not least: Lyana, Taha, Alan and Tim, you make up for a great team, and we are glad we got to develop and build the Smart Teddy together. We feel proud to have contributed to a product which will make a positive social, financial and physical impact on seniors suffering from dementia and their loved ones.

Shea Haggerty and Laura Croes
Delft, June 2021

---

[1] To be hereafter mentioned as the Haagse Hogeschool
[2] To be hereafter mentioned as TU Delft

Contents

1

Introduction

1.1. Situational Assessment

Year by year, the life expectancy of people in the Netherlands increases, leading to an increase in the strain on the healthcare system [39]. Additionally, each year almost ten million new dementia patient are diagnosed world wide, making the total number of people that will have been diagnosed with dementia to be 82 million by 2030 [48]. The past two years, the lack of resources in the healthcare system has become even more troublesome due to the rise of COVID-19. According to research conducted by Numbers & Brodaty, people suffering from dementia are more likely to contract severe Covid-19 and, in addition, are experiencing an extreme level of isolation due to the lockdown restrictions [33] - and sadly, the risk of Alzheimer disease more than doubles for seniors when they experience extensive loneliness [53].

Dementia is an incurable, progressive syndrome which causes deterioration in cognitive function beyond what might be expected from normal ageing. It affects memory, thinking, orientation, comprehension, calculation, learning capacity, language, and judgement, while contrarily, the consciousness of the person suffering from dementia is not affected [48] - making it an especially awful syndrome for both the patient and their loved ones.

The decrease in the cognitive function of a person suffering from dementia is accelerated when they are socially isolated [28]. Social isolation is a common occurrence among seniors, who make up the majority of the people diagnosed with dementia - and sadly, approximately seven percent of people over the age of seventy suffer from a form of dementia [2].

As the dementia of the patient progresses, it becomes considerably more and more dangerous for them to live independently; the risk arises of developing a disturbed sleeping and eating schedule, as well as the occurrences of dangerous situations (e.g. leaving the gas stove on, falling, wandering). In order for seniors suffering from dementia to live independently for a longer period of time, it is imperative they receive regular check-ups from loved ones or carers. However, this can be a strenuous task for loved ones, both mentally and physically [17], and involving a carer into the routine of the senior suffering from dementia can pose as a large financial burden. However, the seniors being able to live independently for a longer period of time will relieve strain from the healthcare systems [17], as care homes often do not have access to sufficient budget and staffing to cope with the number of patients in need of being admitted into a care home. The latter is known to induce emotional, physical and mental strain on care home staff [20].

1.2. The Smart Teddy

In September 2018, a group of researchers from the Haagse Hogeschool in the Netherlands started the Smart Teddy project, in order to tackle this enormous problem which increases in severity as each year passes. The Smart Teddy project aims to develop a teddy especially for seniors suffering from the early stages of dementia, who wish to prolong the time for which they can live independently. This can be achieved by monitoring and recognising daily activities and routines of the senior, as well as recognising previously mentioned dangerous situations. Using the data collected by the Teddy, the quality of life can be determined, which can thereafter be sent to the caregiver or close relative of the senior. As a result of this, the caregiver or relative receives regular updates and reassurance regarding the well being of the senior, and will be alerted when dangerous situations occur.

Besides monitoring the senior, the Smart Teddy provides companionship in order to battle social isolation, which in return helps to slow down the progression of dementia [53]. The Smart Teddy is meant for inviting the user for human interaction, providing the user with a sensation of constant company.

> *"The goal of the Smart Teddy project is to create an interactive system, using a teddy as the focus point, which collects the data needed for determining the quality of life of seniors suffering from dementia by monitoring pre-defined indicators, to prolong the time for which they can live independently."*

Since the start of the Smart Teddy project in 2018, three versions of the Smart Teddy have been developed and tested (some of which virtual machines). The most recent prototype consists of a teddy bear and a base station, which is meant for charging the Teddy[1] and performing calculations on the collected data.

The Smart Teddy thesis project is tackled and further developed by six electrical engineering students. The six students are divided into duos, creating three sub-groups within the project. Each sub-group will address one sub-domain of the Smart Teddy system, and write a bachelor thesis accordingly. A more detailed and visual overview of the sub-domains of the Smart Teddy bachelor thesis project is depicted in Figure 1.1. The division of the sub-domains is the following:

- **Human Interaction & Integration:** this is the sub-domain being addressed in this thesis. It concerns the aspect of stimulating human interaction with the Teddy by implementing key features such as movement and sound, and covers the overall integration of all sub-domains into one complete system. Aspects belonging to the Integration aspect are enabling communication between the sensors and actuators in the Teddy, communication between the Teddy and the base station, control of the actuators and the integration of all components into a Smart Teddy prototype.
- **Power:** this sub-domain will implement the (wireless) charging of the Teddy, and analyse the power flows within the system as to supply each component or device with the necessary power.
- **Sensors & Data Acquisition:** this sub-domain will perform research on how to best make use of different types of sensors within the Smart Teddy to acquire data about the daily activities of the senior suffering from dementia.

This thesis concerns the Human Interaction & Integration sub-domain of the Smart Teddy project. The process of the design, implementation of the design choices, and the final prototype are documented in this thesis.

---

[1] In this thesis, the teddy bear of the Smart Teddy system shall be referred to as the Teddy, whereas the base station shall be simply referred to as the base station.

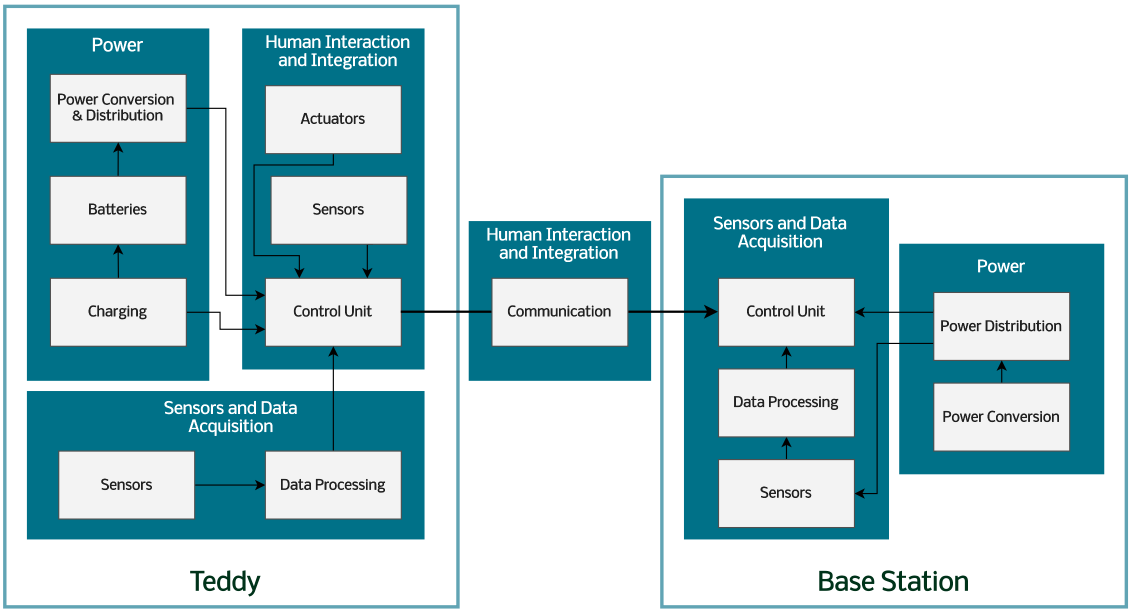


Figure 1.1: General system overview

1.3. Problem Definition

This section contains the problem definition that is used to define the problem definition of the sub-domain 'Human Interaction & Integration'. Section 1.1 contains an extensive situational assessment to provide for context. To create an overall view on the problem, the stakeholders surrounding the solution the Smart Teddy provides will be analysed, as well as the scoping and bounding of the problem, which will be addressed in Sections 1.3.1 and 1.3.2, respectively.

The overall problem definition of the Smart Teddy project is as follows: as of now, there is no working prototype of the Smart Teddy which is able to actively interact with the user using audio, movement or temperature. On top of that, the Smart Teddy is not able to sense touch of the user or record audio/perform speech recognition. Lastly, the sensors in the Teddy are not functioning due to the way they are integrated in the final design of the prototype.

As this thesis addresses just one of three sub-domains of the Smart Teddy project, a more specific problem definition must be defined. The problem definition within the sub-domain Human Interaction & Integration is the following: as of now, there is no Smart Teddy prototype capable of human interaction (i.e. the implementation of movement, audio etc.). Additionally, the sensors inside the Teddy used to acquire information about the daily activities of the senior do not function - although sensors used for data acquisition are not within the scope of this sub-domain, the malfunctioning of them due to poor integration in the Teddy directly addresses the integration part of the sub-domain.

> The goal of the Human Interaction part is to enable the Teddy to physically, visually or vocally react in a pre-determined way to pre-determined inputs from the user. The goal of the Integration part is to combine all sub-domains into one working prototype, which complies with the technical, aesthetic and practical requirements provided by the client.

The latter addresses the integration of the battery system of the Power sub-domain into the Teddy, the integration of the sensors of the Sensors & Data Acquisition into the Teddy, the communication of the different components with the microcontroller in the Teddy, the communication between the Base station and the Teddy and the control module within the Teddy.

1.3.1. Scoping

The Smart Teddy project combines many different technical elements and asks for viewpoints from various fields of expertise. For the technical design of the Smart Teddy, next to extensive knowledge on electrical engineering (especially hardware related), knowledge on computer engineering, industrial design engineering and mechanical engineering is needed. On top of that, as the Smart Teddy is designed for seniors suffering from dementia, it is crucial to consult with professionals who are specialised in dementia, or who have experience in working with/caring for people with dementia. Consulting with the latter is imperative as to verify whether the design requirements are in line with the wants and needs of our target group. In order to obtain a complete view on the problem, and in particular all stakeholders who play a role in the problem, a stakeholder analysis is conducted. Table C.1 in Appendix C lists all stakeholders, their role in (solving) the problem, and their influence on the proposed solution, which is in our case the Smart Teddy. Lastly, the level of influence they impose on the proposed solution is stated.

1.3.2. Bounding

As the product is meant for seniors suffering from dementia, we must be aware of the fact of the physical limits that this target group brings along. As seniors tend to suffer from reduced muscle mass and a reduced range of motion, the weight and size of the Teddy should be within a specified range in order to ensure manageability of the Teddy. Additionally, we must be aware of the cultural limitations regarding the target group: in many Asian, African and Latinx cultures, it is custom for seniors to move in with their children once their health starts declining. This custom eliminates the need for a Smart Teddy system in the home - hence, most likely, there will be a strong cultural limitation on the size and demographic of the target group. However, the Smart Teddy could still be of added value if used in hospitals or care homes in the aforementioned regions.

Finally, as the Smart Teddy is primarily designed for private use, we must keep the financial limitations of the product into account when designing the Smart Teddy. To make the Smart Teddy an accessible solution to the problem stated in Section 1.3, the Smart Teddy should be affordable for seniors/loved ones/care takers to purchase it.

1.4. State-of-the-Art Analysis

In this state-of-the-art analysis, well-known and recently developed products similar to the Smart Teddy will be listed and analysed, particularly focusing on their ability to stimulate human interaction and, if known, how well these technologies have been integrated into the product to maximise the ease, practicality and effectiveness of use. Table 1.1 shows a list of the aforementioned existing products - as of now, several products providing seniors suffering from dementia with companionship to support their mental -and indirectly the physical - health exist [40]. The most widely known product is the PARO seal; the first cognitive therapy-approved robot [45] developed by Takanori Shikbata at Japan’s National Institute of Advanced Industrial Science and Technology [21].

While PARO is the benchmark for robots that provide companionship for seniors suffering from dementia, many other robots are either available on the market, or have been developed for research. These robots can be categorized into the following types;

1. The ’assistance’-type: these robots are designed to support seniors who live independently by assisting them e.g. in doing household tasks or reminding them to take their medicine [40].
2. The ’monitoring’-type: robots developed to monitor the health and safety of seniors [40].
3. The ’companionship’-type: a social companion robot. The user provides input to the robot to which it reacts, allowing for robot-human interaction [40].
4. A hybrid of the above types.

The Smart Teddy is a combination of the monitoring and companionship type. As this thesis addresses the sub-domain Human Interaction & Integration, the existing robots in these categories, which are able to stimulate human interaction, are analysed. Table 1.1 lists these robots, together with their type, key features which are designed to enable human interaction, availability (cost, available on the market etc.) and, if known, technologies used for implementation of the key features into the product. Table

1.1 shows that even though there are many applications of robots providing either companionship or either monitor their user, not many applications provide both. This is unique feature that the Smart Teddy will have.

For monitoring purposes, vastly different approaches as opposed to human/animal-like robots have been taken. As seniors spend most of their time at home, the implementation of smart homes is proven to be a useful approach in order to monitor the senior [34]. However, a companionship type robot provides a different type of support as the user creates a social bond with the robot, which can help solve the problem of loneliness among seniors.

All robots stated in Table 1.1 which are both affordable (comparative to the other robots listed) and can perform movement, make use of servos. Although this is a low-cost and non-complex option, the use of servos creates weak links in the design which are prone to breakage when dropped or handled roughly. The rise of soft robotics is continuously becoming more dominant in the world of robotics [24]: soft robots make use of flexible and easily deformable materials (such as silicon or rubber) and induce movement via e.g. pnuematics[2] or electroactive polymers. Figure 1.2 shows an example of a soft robot. Whereas servos induce rigid and mechanical motion, soft robotics enable soft and natural biological-robotics [24], which, if implemented in the Teddy, could add greatly to its life-likeness. On top of that, a driving factor for the implementation of soft robotics is its safety - soft robots are expected to play a larger role in human-robot interaction in e.g. factories, public spaces, households, and healthcare. As the soft-robots mimic biological tissue, it is much safer and more effective than the rigid robots that are available on the market now [24].

Figure 1.2: A small silicon-rubber based soft robot. By inflation of the robot, the flexible material expands and causes deformation, leading to movement. *Source: Ryan Truby/Harvard SEAS.*

Lastly, the brain of the Teddy, the micro controller is also a relevant topic in research and in a wide range of industries [29]. This is a consequence of the constant need to make devices smart and adding them to the Internet of Things (IoT). This is a rising trend in health care as well, as it allows for better care and closer monitoring of the patient, by being able to record data from different devices and having them communicate with each other [43]. These devices can be as specific as a wearable insulator monitor, or as general as an apple watch. More and more home made options are evolving from the growing market for home projects. This gives the average person access to microcontrollers that are simple to program, like the Arduino. More complex microcontrollers, like the Raspberry Pi or the ESP-controllers are more commonly used by experts in the field. This has created somewhat of a shift for product development as engineers are no longer required to design all components from scratch and has allowed for rapid prototyping. This same approach will be followed during this thesis in order to have a working prototype in the given ten weeks.

1.5. Synopsis

This thesis contains the literary research, design process and development of the Human Interaction & Integration sub-domain of the Smart Teddy project. Chapter 2 addresses the list of requirements presented by either the client or healthcare professionals which were consulted during the project. Chapters 3 and 4 contain a detailed and thorough overview of the design choices made regarding human interaction and integration of the Smart Teddy. The implementation of the design choices made in the aforementioned chapters can be found in Chapter 5, along with the verification whether or not the implementation complies with the requirements. In Chapter 6, the discussion and conclusion can be read.

---

[2] Pnuematics is a branch of engineering which makes use of pressurised gas or air [51].

Table 1.1: Table containing the different types of products on the market/being developed in order to stimulate human interaction with seniors suffering from dementia [40].

Product	Type	Features	Availability	Technologies used
PARO	Companionship	Produces seal noises, has life-like temperature, moves its head and flippers, can maintain balance. Stimulates high level of interaction with user [17].	Commonly used in care homes world wide, but financially unattractive for private use at €5752,66.	PARO contains tactile, light, audition, temperature and pressure sensors and can perform speech recognition. It is electromagnetically shielded, so it can be safely used by users who have pace-makers [45].
Music Teddy-bear	Companionship	Reacts to user touching its paw by playing music or audio fragments.	Available on the market for €69,99.	Contains a touch sensor and a microphone. Via a USB connection the user can upload MP3 files to the integrated device.
Cat Robot	Companionship	Is able to move its head and limbs, can produce cat-like sounds, and can adjust position (roll over, lie down).	Available on the market for €129.	Contains touch sensors on its head, back and stomach. Servos enable the movement of the Cat.
AIBO	Companionship	Able to move (walk, sit, lie down), stimulate emotion by facial expressions and produce barking sounds.	Available on the Asian and American market for €2900.	Contains a 64-bit Quad-Core processor, pressure-based capacitive touch sensors, front camera, 4 microphones, a speaker and 2 OLED displays for its eyes [3].
NeCoRo	Companionship	Can move its limbs and head as a reaction to touch.	Available on the market for €699.	Contains 15 servos to enable movement, a microphone, and 5 touch sensors throughout its body [30].
Huggable	Companionship	Can move its limbs when sensed that it is touched.	Not available on the market.	Contains many pressure sensitive sensors. Voice coil actuators enable silent movement. Contains a Pentium M embedded PC [**Hug**].
Icat	Companionship	Can make facial expressions, indicate its current activity using colored lights, play audio and move its head.	Not available on the market.	Contains 13 servomotors to enable movement and generate facial expressions. The feet contain touch sensors and a microphones in order to perform speech recognition. The ears contain LEDs [32].
Pearl	Monitoring	A human-like robot which can express emotion through facial expressions.	Not available on the market.	Contains a differential drive system, two Pentium PCs, wireless Ethernet, sonar sensors, microphones for speech recognition, displays which react to touch and speakers [38].
Wakamaru	Monitoring	Can recognise and produce speech, stimulate emotion by facial expressions, move and ’walk’.	Available on the market for approximately €15.000,-.	Wakamaru contains touch sensors, odometers, a LED display, a microphone and a speaker [27].

2

Program of Requirements

This chapter contains the program of requirements for the Human Interaction & Integration sub-domain of the Smart Teddy project. Firstly, context and assumptions are provided in order to understand the origin of (specific) requirements. Next, the verification methods used to test if the requirements are met are listed, after which the actual requirements are stated.

2.1. Assumptions and Context

To narrow down the scope of the project and provide for more context, assumptions have been made on the target group and the setting in which the Smart Teddy is to be used. The main assumption is that the target group are seniors suffering from the early stages of dementia, and that they still live independently. The Smart Teddy will be developed with this in mind, as this setting is vastly different from a person living in a care home. Due to the fact that the users live independently, we assume they will not be receiving help on a daily basis from carers, meaning the Smart Teddy must be able to operate and be useful to the senior without frequent assistance. However, we are aware that a number of stakeholders mentioned in Table C.1 will exercise a great amount of influence on the choice whether or not to purchase and use the Smart Teddy (e.g. carers, loved ones, the senior them self), which may bring along the need to alter some requirements. Lastly, considering the user will show early stage symptoms of dementia, we must recognise that they can be forgetful and show behavioural changes [48]. Keeping the above in mind, and using insights and knowledge provided by Ezra and Annemarie, a program of requirements is set up.

The program of requirements is split up into nonfunctional and functional requirements, which can be seen in Section 2.3 and 2.4, respectively. Nonfunctional requirements describe the qualities, characteristics or attribute the Smart Teddy must or should have [23], while the functional requirements illustrate what the Smart Teddy must or should be able to do, concerning the Human Interaction & Integration sub-domain.

2.2. Verification Methods

In order to test if the requirements are met once the Smart Teddy prototype is finished, we must define verification methods. Conforming with the systems engineering approach, there are four methods used for determining whether or not a requirement has been met [26]. These methods are:

Analysis	A mathematical approach or simulation is used to check if the requirement is verified.
Test	The requirement is tested during a predefined test, which replicates the operating conditions, and is executed following a procedure.
Inspection	Verification of the requirement is done by design and is defined in the documentation.
Demonstration	During operation and demonstration the requirement is verified.

2.3. Nonfunctional Requirements

The list of nonfunctional requirements can be seen in Table 2.1. Along with the requirement itself, the reason, or ’rationale’, behind each requirement is stated. Lastly, the verification method which will be used to examine if the deliverable meets the requirement is stated as defined in Section 2.2.

Table 2.1: Table containing nonfunctional design requirements. Nonfunctional requirements are labeled NF.XX.

ID	Nonfunctional Requirement	Rationale	Verification Method
NF.01	The Teddy must not appear to be a machine.	The Teddy should give the user the illusion that it is a Teddy, as opposed to a robot.	Demonstration
NF.02	The Teddy must not contain parts that could be a choking hazard.	Some people who suffer from dementia might put foreign objects in their mouth.	Inspection
NF.03	The Teddy must weigh less than 2.0 kg.	The user must be able to easily pick up the Teddy.	Analysis
NF.04	The Teddy must be designed in such a way that overheating of the components inside of the Teddy shall not occur.	Overheating can cause damage to the components and could cause dangerous situations, and must be avoided at all times.	Test
NF.05	Moving parts must never become potentially dangerous for the user.	When, for instance, the Teddy moves and the user is blocking this movement, a limiter should be placed to avoid harming the user.	Test
NF.06	The Teddy must be safely machine washable.	As the Teddy will be used extensively, the Teddy must be easy to clean.	Demonstration
NF.07	The Teddy must be robust (i.e. must be able to fall without breaking).	As the Teddy will most likely be carried around the house, and dropped occasionally, it must be robust to withstand extensive usage.	Demonstration
NF.08	The measurements of torso of the Teddy should range between 30-40cm (length) x 15-20cm (width).	The Teddy should be a comfortable size to place on the users lap, but large enough to stroke the Teddy to stimulate interaction.	Inspection
NF.09	The user should not be able to feel hard objects inside the Teddy.	In order to stimulate interaction, the Teddy should feel like an actual cuddly toy.	Demonstration
NF.10	The movements of the Teddy should preferably be silent (not more than 20 dB).	If the movements make too much (mechanical) noise, the Teddy will come across as a robot in stead of a cuddly toy.	Demonstration
NF.11	The manufacturing costs of the prototype Teddy should preferably not exceed €400.	In order to keep the Smart Teddy financially accessible to users, the manufacturing costs should be kept as low as possible.	Analysis

2.4. Functional Requirements

The list of functional requirements can be seen in Table 2.2. Along with the requirement itself, the reason, or 'rationale', behind each requirement is stated. Lastly, the verification method which will be used to eventually examine if the deliverable meets the requirement is listed.

Table 2.2: Table containing functional design requirements. Functional requirements are labeled F.XX

ID	Functional Requirement	Rationale	Verification Method
F.01	The Teddy shall be able to sense when stroked/touched.	This enables the Teddy for human interaction, as the Teddy can react to touch.	Test
F.02	The Teddy shall recognise 2-3 words using speech recognition.	In order for the Teddy to react to speech, the Teddy should be able to recognise specific words which trigger a response.	Test
F.03	The Teddy must contain an indicator of the state of the battery.	It must be clear to the user when the Teddy needs charging, to avoid an empty battery.	Inspection
F.04	The Teddy shall be a stand alone system which doesn't require control from external sources.	To avoids latency and avoid problems when there is bad communication between the Teddy and the base station, all control should be done locally inside the Teddy.	Inspection
F.05	There shall be an indication that the Teddy is charging that is in agreement with NF.01	This will help the user in knowing when the Teddy is placed correctly without having the Teddy look like a gadget.	Test
F.06	The components inside the Teddy shall be able to communicate with the main controller.	All sensors and actuators will communicate to the main controller. The main controller then decide on how to handle.	Inspection
F.07	The components inside the Teddy shall be able to communicate wirelessly with the Base Station.	In order to analyse the data acquired from the sensors, it must be sent to the Base Station.	Test
F.08	The response time in between touching the Teddy and the reaction of the Teddy should be no more than 0.5 seconds.	To give the user the feeling the Teddy is responding to their voice/touch, the reaction time should be as short as possible.	Test
F.09	The Teddy should be able to react to human touch and speech (certain words) using movement and sound.	This reaction of the Teddy gives the user the sense that he/she is interacting with the Teddy.	Inspection

3

Design: Human Interaction

This chapter contains the design steps and the final design choices of the Human Interaction part of the Human Interaction & Integration sub-domain. The Human Interaction aspect of the Smart Teddy addresses all features which stimulate or create human interaction with the Teddy - think of the Teddy reacting to speech in the form of movement, or barking when the Teddy senses it is spoken to.

The design process is initiated by constructing a design option tree, showing all possible implementations and solutions on how to stimulate human interaction with the Teddy. Thereafter, the individual elements of the design option tree are elaborated on. Finally, the final design choices for the Human Interaction part are summarised.

3.1. Design Option Tree

Figure 3.1 shows a visual overview of the design option tree made for the Human Interaction aspect of the Smart Teddy. As is depicted, five main triggers for verbal and non-verbal human interaction are identified, based on the human senses: movement, audio, scent, visuals and temperature. In order to stimulate frequent interaction with the user, the Teddy must include features in its design which fall into the aforementioned categories. In the design option tree, all possible implementations of different ways on how to incorporate these triggers into the Teddy have been identified. Naturally, not all are realistic or comply with the requirements. That is why in Figure 3.1, some triggers and implementations are coloured grey and others green. Grey means the implementation is not suitable for the Smart Teddy, and green means the implementation is a suitable option for the Smart Teddy and will be incorporated into the prototype of the Smart Teddy.

Sections 3.2, 3.3, 3.4 and 3.5 elaborate on the reasoning for the design choices as indicated in the design option tree, and the implementation of the triggers in the Smart Teddy.

It must be noted that the Teddy used for prototyping is a cuddly toy dog. This means that, when evaluating possible movements and sounds the Teddy could make, only dog-like characteristics are considered to maintain a high level of life-likeness of the Teddy.

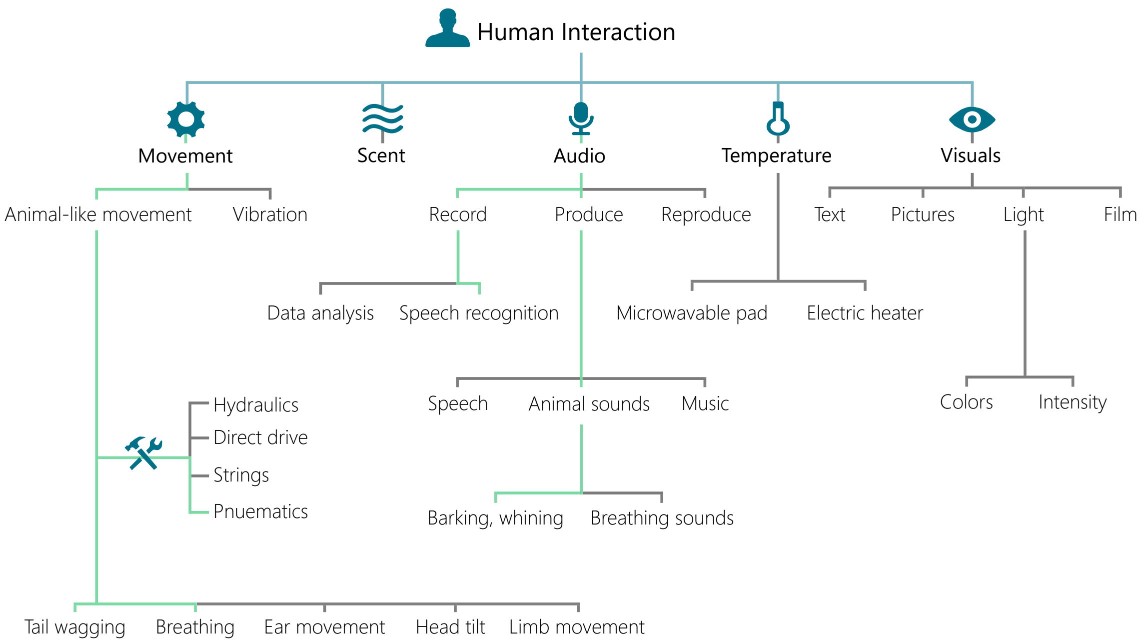


Figure 3.1: A visual overview of the design option tree for the Human Interaction aspect of the Smart Teddy.

3.2. Movement

This section addresses trade-offs regarding the movement aspect of the Smart Teddy. Movement is an imperative aspect to the Smart Teddy, as when a robot physically interacts with a human, it stimulates affection [31]. As is depicted in Figure 3.1, there are a total of five possible movements in the Teddy: wagging of its tail, movement of the ears, breathing, a head tilt, and limb movement. Table 3.1 shows these movements, along with their scores on the relevant criteria: the level of which the movement stimulates physical and verbal interaction, the robustness, its weight and its expected power consumption. Although the implementation method is not yet specified, the weight, power consumption and robustness can be comparatively analysed - for instance, moving limbs of the Teddy as opposed to stimulating breathing will most likely be less robust, as limbs are thinner and more fragile than the torso.

As can be seen in Table 3.1, breathing obtains a high score. The simulation of breathing improves the life-likeness and user perception of robots [54], to the extent where the user interprets the robot as a living being [56]. Simulating a breathing motion in a robot is a widely used technique to increase physical interaction with the robot, and is proven to slow down the users own breathing rhythm, resulting in a more calm state of mind [25]. Some examples of robots/products that use the stimulation of breathing to enable users to calm down, or even meditate, is the sleep-robot developed by Somnox [46] or the ZZZoo pillow developed by researchers from the Kanagawa Institute of Technology in Japan [55]. Considering the above, the effect of incorporating breathing in the Smart Teddy could potentially help the user, as dementia patients can suffer from anxious or even fearful episodes [9].

Both movement of the ears and the head tilt score very high on their ability to stimulate verbal interaction. This is because when spoken to, most animals will respond by moving their ears up or to the side [41]. Although both movements score high on verbal interaction, they do not necessarily stimulate physical interaction, which is an important criterion for the Smart Teddy. Next to the movement and position of the head and the ears of an animal, the movement of the tail says a lot about the mental state of the animal. As the tail is a very lightweight and relatively small part of the Teddy, it will most likely be a low-cost, lightweight and low-power option. Ideally, all four high scoring movements would be implemented in the prototype. However, due to budgetary and time constraints, a selection must be made. As breathing scores the highest in the trade-off, and tail wagging is expected to be a low-cost and light-weight motion, these features will be implemented in the prototype of the Teddy.

Table 3.1: Comparison of the possible movements to be implemented in the Smart Teddy using scoring on a scale of 1 to 5. Note that a high score is positive - e.g, a score of 5 for 'cost' means it is a highly low-cost option.

Movement	Physical interaction	Verbal interaction	Robustness	Weight	Power consumption	Total
Tail wagging	1	4	3	5	4	17
Ear movement	1	5	2	5	4	17
Breathing	5	2	4	4	3	18
Head tilt	3	5	4	3	2	17
Limb movement	2	3	2	1	1	9

3.3. Audio

Audio can be recorded, produced, and reciprocated. All three are forms of interaction between the Smart Teddy and the user. However, reciprocating audio contradicts Requirement NF.01, which states that the Teddy must not appear machine-like. Therefore, reciprocating sound is eliminated as a design option. The two remaining forms of interaction through audio, recording and producing of sound, are discussed in detail in the following sections.

3.3.1. Recording Audio

The recording of audio can be used for multiple purposes: to save the recorded data for analysis, or to recognize words or the pitch of the human speech in order to form a response, stimulating human interaction. As saving the recorded data for analysis does not stimulate interaction, recognising words or the pitch of human speech remains. Table 3.2 contains a trade-off which compares recognising the pitch of the user in order to perform speech recognition. By recognizing words, the Teddy can respond to what the user is specifically saying, which stimulates interaction. On the other hand, it is less complex to implement solely a pitch recognition.

A study was done by the University of Birmingham where robots were used to correct the behavior of seniors [7] - the robot was programmed to disregard the user's request when their pitch was classified as angry. While this is an interesting use of the interactive function of the robot, and additionally being low in complexity to implement, it brings along a complex ethical discussion outside of the scope of the Smart Teddy bachelor project. The latter consideration leads to the implementation of speech recognition scoring the highest in Table 3.2, and will therefore be implemented in the Teddy.

Table 3.2: Comparison of audio recording methods that stimulate interaction in the Teddy using scoring on a scale of 1 to 5. Note that a high score is positive - e.g, a score of 5 for 'complexity' means it is a less complex option.

Use of recorded audio	Complexity	Function Enhancing	Encourages Interaction	Total
Speech Recognition	4	4	5	13
Pitch Recognition	2	5	5	12

3.3.2. Producing Audio

The benefit of including the production of audio is that the Teddy will be perceived as more interactive, and therefore more fulfilling as a social companion - and the more useful a robot is to the user, the more likely they are to continue using them [40]. The decision on including the production of audio is not only justified using the latter reasoning, but also complies with Requirement F.09. There are three kinds of audio the Teddy could produce: speech, music and animal-like sounds. A trade-off as to which sound is the most suitable is given in Table 3.3.

The production of music, speech and animal-like sounds all stimulate interaction, especially when used as a reaction to user input. Keeping requirement NF.01 in mind (regarding the life-likeness of the Teddy), including speech or music would be contradictory. Using whining and/or barking, however, allows for a natural indication of both the emotional status of the Teddy and the current state of charge of the Teddy's battery. As for the complexity: including music or speech requires more memory, next to the fragments consisting of complex frequency compositions. On top of that, barking and whining are relatively short and simple sounds compared to speech and music. The latter drastically reduces the complexity of implementing whining and barking, combined with the fact that a bark can be used in multiple situations where the possibilities of implementing speech might be more restricted. The potential benefit of adding speech could, for example, be that games could be played with the user and/or reminders can be given (e.g. for the user to take their medicine), which is proven to be beneficial for people with dementia [40]. To stay in compliance with the requirements, the choice was made to solely include whining and barking in the Smart Teddy.

Table 3.3: Comparison of audio usage in the Smart Teddy using scoring on a scale of 1 to 5. Note that a high score is positive - e.g, a score of 5 for 'complexity' means it is a less complex option.

Audio usage	Comforting	Complexity	Function Enhancing	Encourages Interaction	Total
Whining/ Barking	3	5	5	4	17
Speech	4	1	5	4	14
Music	2	2	3	3	10

3.4. Visuals

According to requirement NF.01, the life-likeness of the Teddy should be respected, and the Teddy must not appear machine-like. Adding visual indications on the Teddy will violate this requirement, and will therefore not be implemented. In order to still have a form of indication on, e.g., the state of charge (SoC) of the battery inside the Teddy, a more 'dog-like' form of indication is used, making use of sound and movement - its implementation is discussed in Section 3.3 and Section 3.2, respectively. The Base station, however, could still have a visual indication. As the Base station has a straightforward integration and is done by the Power sub-domain, it is not considered part of the scope of this thesis.

3.5. Temperature

The temperature of a robot significantly influences the user perception of the robot [35]. As the Smart Teddy must evoke the feeling within the senior of being in the company of a life-like companion, the implementation of temperature regulation could influence their perception of the Teddy in a positive way. Table 3.4 shows the comparison of two possible methods for regulating (heating) the temperature of the Teddy: including a microwavable pad inside the Teddy or making use of an electrically heated film.

As can be seen in Table 3.4, the microwavable pad is the least suitable heating method for implementation in the Teddy, mainly due to its safety and weight. A microwavable pad might be left in the microwave for too long, resulting in potentially damaging components inside the Teddy, and in the worst case, hurting the user. On top of that, microwavable pads are often very heavy (filled with water, rice, dry beads) which is in defiance with Requirement NF.03. Using an electrical heating film is much more suitable. When combined with a temperature sensor, the risk of overheating can be diminished, and the temperature can be more precisely regulated than when using a microwavable pad, making it a safer option. Additionally, heat films and temperature sensors are lightweight components. However, a heat film does consume a considerable amount of power [1], which could result in reducing the battery life significantly. Considering the power consumption of electrically heating the Teddy, and budgetary and time constraints of the project, the design choice is made to leave out heating the Teddy in this prototype of the Smart Teddy.

Table 3.4: Comparison of the implementation methods for heating the Teddy using scoring on a scale of 1 to 5.

Temperature implementation	Safety	Complexity	Power consumption	Weight	Total
Microwavable heat pad	1	2	5	1	9
Electrically heated	4	3	3	4	14

---

[1] On average 0.1-0.2 Watt/cm^2.

4

Design: Integration

This chapter contains the design steps and the final design choices of the Integration part of the Human Interaction & Integration sub-domain. The Integration part is split up into 'Communication' and 'Control', as the design choices and reasoning differ for both these aspects. A visual overview of these subsystems are shown in Figure 4.1. All components inside the grey box labeled 'integration' are part of the integration subsystem. For clarity, the specific aspects addressed within integration are:

- The design choice regarding the microcontroller in the Teddy;
- The design choice regarding the type of wired communication in the Teddy and the Base Station;
- The design choice for the type of wireless communication used between Teddy and Base Station.

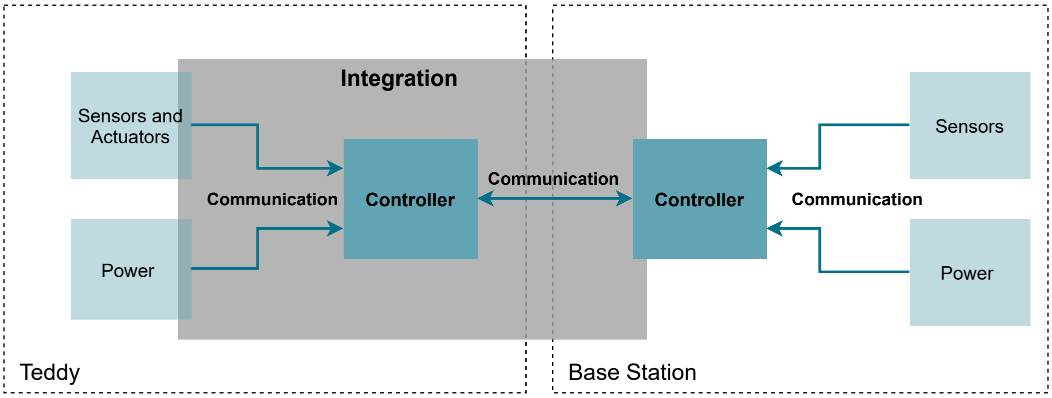


Figure 4.1: Overview of integration in the Smart Teddy with the integration subsystem contained inside grey box. Note that controller inside the Base Station is not part of the integration design of this thesis as the main focus is on the Teddy.

4.1. Communication

In order to integrate all three sub-domains into a fully functioning system, a method of communication must be established between the different elements of all sub-domains. In Appendix D, different communication protocols[1] are compared, as to find the best fit for the Smart Teddy. As is depicted in Figure 4.1, we wish to establish both an intra-connected wired and inter-connected wireless communication [4]; the latter in order for the controller of the Teddy to communicate with the controller of the Base Station. Wired communication for intra-device communication is preferred, as wired protocols are more reliable, secure, easier in use [6] and can transfer data at higher rates than wireless protocols [19]. The options for both wired and wireless communication are explored, in Sections 4.1.1 and 4.1.2, respectively.

---

[1] Communication protocols determine the manner in which devices can mutually interconnect [12].

4.1.1. Communication within the Teddy and Base Station: Wired

The different elements in the Teddy and the Base Station require a bi-directional communication mechanism in order to exchange the data retrieved from the sensors in the Teddy and the Base Station to the controller, and additionally get the user actions from the controller to be applied to the actuators [19].

Initially, a full analysis was performed in order to decide on the optimal communication protocol for the Smart Teddy. The full analysis can be read in Appendix D - the conclusion being that I2C was the most suitable protocol for implementation in the Smart Teddy. However, in practice, we noticed that there is little to no room to choose a communication protocol, as we make use of pre-fabricated modules that are compatible with limited (1 or 2) communication protocols. Figure 5.4 shows the different modules in the Smart Teddy, together with their communication protocol. Although I2C is our preferred communication protocol, the wired communication methods used in the Smart Teddy are I2C, UART and SPI.

4.1.2. Communication Between Teddy and Base Station: Wireless

A wireless communication protocol is needed for transmitting data collected by the sensors inside the Teddy to the Base Station, where the data is collected and potentially sent to carers/family members of the senior. The five wireless communication protocols which we examine and compare are WiFi, Bluetooth (Classic and Low Energy, or LE), ZigBee and LoRa [2] - in order to perform a comparative analysis, different criteria of these protocols are analysed, as is portrayed in Table 4.2. The criteria are:

- **Reliability:** as the data sent from Teddy to Base Station contains vital information about the health and wellbeing of the senior, it is imperative the wireless communication module contains error reduction techniques to ensure correct transmission of the data.
- **Range:** as the Teddy is portable, it is likely that the situation will occur that the Teddy is located in another room than the Base Station, meaning the range of the protocol is important to consider.
- **Power consumption:** we strive to keep our power consumption as low as possible to reduce the battery size and life, in order to comply with Requirement NF.03 and NF.09.
- **Widely supported:** considering Requirement NF.03 and NF.09, the amount of hardware needed inside the Teddy must be minimised. A number of microprocessors have built-in modules for the most common wireless communication protocols, hence reducing the hardware inside the Teddy.
- **Complexity:** the ideal wireless communication protocol would be as straightforward to implement as possible, due to time constraints of the project.

Note that we do not consider the data transfer rate and bandwidth as important criteria, as there is no need for fast transmission of the data between the Teddy and the Base Station, or the need for a high amount of data to be sent per second. The transmission has no action-reaction aspect to it, and thus has no influence on whether Requirement F.08 is met. Table 4.1 shows that of the four wireless protocols, WiFi and Bluetooth Classic are the most widely supported for our implementation purpose. When comparing WiFi and Bluetooth (Classic and LE), we notice that WiFi consumes considerably more power. ZigBee and LoRa consume the least amount of power of the listed wireless protocols.

All communication protocols have both encryption and authentication mechanisms: Bluetooth Classic and LE, ZigBee and LoRa use a 16-bit Cyclic Redundancy Check (CRC) [3] while WiFi uses a 32-bit CRC [18] [8]. In conclusion, for our implementation purpose, all protocols feature sufficient error reduction techniques.

When we inspect the range of the four protocols, we see that ZigBee and LoRa have the an extremely long range, in contrast with Bluetooth Classic and LE: these have a nominal range of 10 meters, which means that if the senior takes the Teddy to another room, it is likely that the Teddy can no longer transmit data to the Base Station. Considering the above, and the information given in Table 4.1, LoRa scores the highest and thus the most suitable choice for the Smart Teddy.

---

[2] Long Range

[3] CRC is an error-detecting code used to detect accidental changes to data [50].

Table 4.1: Different wireless communication protocols and their properties [19][10].

Protocol	Error reduction	Range	Power consumption	Price
WiFi	32-bit CRC	10-100 m	100-350 mA	€5-€8
Bluetooth Classic	16-bit CRC	10 m	1-35 mA	€6-€12
Bluetooth LE	16-bit CRC	10 m	1-15 mA	€3-€5
ZigBee	16-bit CRC	10-1000 m	1-10 mA	€10-€22
LoRa	16-bit CRC	2000 m	1-10 mA	€8-€15

Table 4.2: Comparison of the different wireless communication protocols rated on a scale of 1 to 5 [19][10].

Protocol	Reliability	Power consumption	Widely supported	Range	Complexity	Cost	Total
WiFi	5	1	5	3	2	4	20
Bluetooth Classic	5	3	5	2	3	3	21
Bluetooth LE	5	5	4	2	3	5	24
ZigBee	5	5	4	4	3	1	23
LoRa	5	5	3	5	5	2	25

4.2. Control

Figure 4.1 provides an overview indicating the context of the controllers in the Smart Teddy. The sub-domain Human Interaction & Integration covers the controller in the Teddy, while the sub-domain Sensors & Data Acquisition examine the controller in the Base Station.

Before exploring the different kind of controllers, we should critically analyse whether a controller in the Teddy is required at all. It is possible to simply collect and send the data from the Teddy to the Base Station, or to control the Teddy by implementing a fast controller in the Base Station combined with real-time wireless communication between the Teddy and the Base Station. However, this increases power consumption, leading to needing a larger battery inside the Teddy. Additionally, according to requirement F.04, the Teddy shall be a stand alone system, meaning it must perform its own control. This is due to the Teddy having to be able to respond to its user even when the connection is lost to the base station, as well as having a low latency to provide a life-like interaction. This is reflected in requirement F.08, which states that there should be a latency of no more than 0.5s between user input and reaction of the Teddy. As this thesis focuses on the integration and human interaction aspects of the Smart Teddy, more time is invested in these aspects rather than real-time wireless communication. Therefore a controller will be placed inside the Teddy. The requirements, as defined in Chapter 2, are used to determine what criteria are relevant in selecting a controller for inside the Teddy. These requirements are summarized in Table 4.3.

Table 4.3: A brief explanation regarding the trade-off criteria and relevant requirements.

Criterion	Relevant requirements
Weight	The Teddy must weigh less than 2 kg, meaning the controller must be light-weight and low-power in order to minimise the size of the battery.
Cost	Requirement NF.11 states that costs should be minimised for financial accessibility.
Computational power	The Teddy must be able to perform speech recognition.
Interface flexibility	The Teddy shall be able to communicate with all components inside the Teddy, in compliance with Requirement F.06.

Before an objective score can be given for each criterion, research is done into relevant controller properties. These properties are summarised in Table 4.4, where '# GPIO' refers to the number of general purpose in- and output pins. The final trade-off is given in Table 4.5. When comparing the options for the trade-off, criteria other than the aforementioned are used. These are not as easily quantifiable, but very relevant. Firstly, the criterion reliability: some boards are less conventional, or are still prototypes, and therefore score lower on this criterion. Furthermore, the available online support is also crucial. Due to the time limit of the project, the availability of online tutorials and thorough documentation are crucial, as prototyping must be done within three weeks. Finally, the complexity is relevant, and is mostly determined by the programming language. For example, when implementing speech recognition, it is less complex to use Python as there are many relevant libraries readily available.

The highest scoring controller is the Raspberry Pi Pico, and will be used for the Teddy. It is important to note that the scoring for the trade-off in Table 4.5 is relevant to this project with its time- and budget limitations, but other controllers could be more suitable in different conditions.

Table 4.4: Different controllers and their properties.

Controller	Source	Price	Weight (g)	Size (mmxmm)	Programming language
Arduino Zero	[5]	€40.00	12	68x53	Arduino IDE, C, C++
Raspberry Pi Pico	[37]	€10.00	$<$ 10	51x21	C, C++, MicroPython
Raspberry Pi 4B	[36]	€60.00	45	85.60x56	C, C++, Python, and more
Beagle Bone Black	[11]	€65.00	40	86.4x53.3	C, C++, Python, and more
NodeMCU ESP8266	[1]	€8.00	$<$ 10	58x31	C, C++n Python, and more
ESP32-S2-DevKitM-1	[16]	€7.00	100	54x25.4	C, C++
MSP-EXP430G2ET	[22]	€17.00	195	TBD	Code Composer Studio

Controller	Comm. Protocol	# GPIO	Power (W)	Flash Memory	SRAM Memory
Arduino Zero	UART, I2C, SPI	20	0.2	256KB	32KB
Raspberry Pi Pico	UART, I2C, SPI	28	0.4	2MB	264KB
Raspberry Pi 4B	UART, I2C, SPI, MPI	40	3	32GB+ [4]	8GB
Beagle Bone Black	UART, I2C, SPI, MPI	69	1	4GB	512MB
NodeMCU ESP8266	UART, I2C, SPI	16	0.35	4MB	64KB
ESP32-S2-DevKitM-1	I2C, I2S, SPI, UART	46	$<$0.001	4MB	320KB
MSP-EXP430G2ET	UART, I2C, SPI	24	N/A	16 KB	512 bytes

Table 4.5: Comparison of different controllers rated on a scale from 1 to 5.

Controller	Cost	Weight	Reliability	Power	Comp. power	Flexibility	Support	Complexity	Total
Arduino zero	4	4	4	4	2	4	5	5	34
Raspberry Pi Pico	5	5	4	4	4	4	5	4	35
Raspberry Pi 4B	3	3	4	3	5	5	5	5	33
Beagle Bone Black	3	4	3	3	5	5	4		32
NodeMCU ESP8266	5	5	4	4	3	3	4	3	30
ESP32-S2-DevKitM-1	5	3	3	5	3	3	4	3	29
MSP-EXP430G2ET	5	2	4	N/A	4	4	4	2	25

5

Implementation and Verification

This chapter contains the implementation method and verification for the design choices made in Chapters 3 and 4. For each subsystem (movement, audio, wireless communication, the controller, and hard- and software integration) the implementation method is stated, after which is verified whether or not the concerned requirement is met, using the verification methods stated in Chapter 2.

5.1. Movement

This section contains the implementation and verification of the movement sub-system. The subsystem contains the movement of the Teddy, paired with the touch sensor to trigger the movement. The implementation and verification can be read in Sections 5.1.1 and 5.1.2, respectively. Please note that the software implementation of the movement-subsystem is elaborated on in Section 5.5.1.

5.1.1. Implementation

There are four potentially suitable implementations for inducing movement in the Teddy: using hydraulics, strings, direct drive (servo motors) and pneumatics. Table 5.1 shows a comparative analysis of the four implementation methods, scoring each method according to five criteria: it preferably being low in cost, weight and power consumption and it being robust (as to comply with Requirement NF.11, NF.03 and NF.07) and finally should be low in complexity. The latter enables for easier reparation when the movement system is broken, making the Smart Teddy more sustainable by increasing its longevity.

As Table 5.1 shows, pneumatics and strings score the highest. Using strings is lightweight and cheap: only one servo motor would be necessary to tighten and loosen the strings. However, strings are more likely to break over a longer period of extensive use of the Teddy. Pneumatics lends itself particularly good for the breathing motion of the Teddy, as opposed to the use of strings. The use of pnuematics requires two air pumps to function as the motor of the system, in stead of the two or three servo motors needed in order to achieve a breathing motion and tail wagging using the direct drive method. As we want the Teddy to stimulate breathing, the most natural looking and feeling solution would be to use air to inflate a balloon-like component. Hydraulics would not be able to reproduce the natural feeling of 'lung' inflation of the Teddy, and would weigh down the Teddy significantly. Lastly, a leak in the hydraulic system could cause all the electrical elements in the Teddy to be destroyed. Considering the above, the pneumatic implementation method is most suitable for usage in the Teddy.

Table 5.1: Comparison of the methods to implement movement in the Smart Teddy using scoring on a scale of 1 to 5.

Implementation method	Cost	Weight	Power consumption	Robustness	Complexity	Total
Hydraulics	2	1	2	2	1	8
Strings	5	5	4	2	2	18
Direct drive	3	4	3	2	2	14
Pneumatics	4	3	3	5	3	18

Breathing

Figure 5.1 depicts the regulation of the air flow in and out of the lungs using a 5V air pump and two three-way 5V valves. The lungs are constructed from Ecoflex-00-30, a light-weight, strong and very flexible silicon rubber. Upon close inspection of Figure 5.1, one could conclude that only one valve is needed to in- and deflate the lungs. However, upon testing it became apparent that deflation of the lungs without a vacuum pump takes a long time, hence the decision to deflate the lungs using the air pump and an added valve. The control of the air pumps and valves is achieved using two motor drivers [1]. The valves are controller with digital signals, and the two pumps are controlled by two PWM signals, with a frequency of 2000 Hz as to minimise the mechanical noise of the pumps. A 100% duty cycle is used to maximise the air pressure the pumps can deliver. The code for the control of the airflow can be seen in Appendix G.

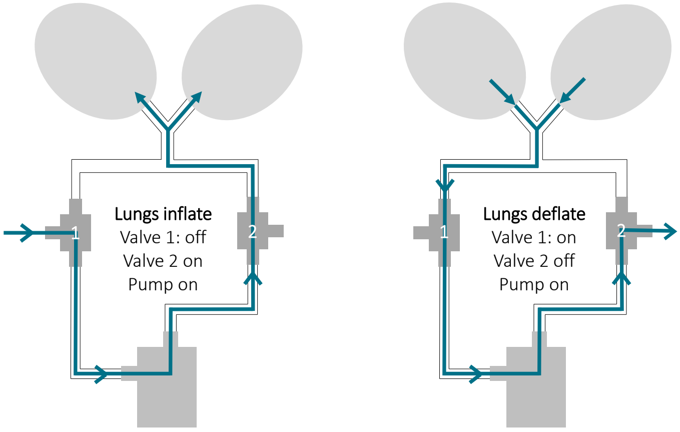


Figure 5.1: Schematic of the connection of the lungs with the two valves and the pump.

Tail Wagging

Using a second 5V air pump, an extra 5V valve and a soft robotic arm, the wagging of the tail can be achieved. The soft robotic arm is also made from Ecoflex-00-30, and consists of two parts: a non-flexible piece (made non-flexible by integration of a piece of paper) and an expanding flexible piece. By increasing the air pressure inside the arm, the flexible part expands, inducing a natural curling movement.

Touch Sensor

As stated in Section 3.2, holding a breathing robot reduces stress and anxiety. Hence, the Teddy should react to touch, and a pressure sensitive touch sensor is needed. According to the requirements, the touch sensor has to minimise its required area, its cost, weight and maximise robustness. Additionally, the touch sensor has to be flexible, as it is placed close to the 'skin' of the Teddy to optimally sense touch. Considering these criteria, Velostat in combination with the Adafruit Momentary Touch Sensor [2] are the most fitting option. Velostat is a highly flexible, lightweight, cheap and robust conductive sheet, which reduces its capacitance when pressurized. The Adafruit board, which contains the AT42QT1010 touch sensor IC, is highly suitable in combination with the Velostat and the Raspberry Pi Pico, as it supports I^2C and contains a soldering hole for a custom touch pad.

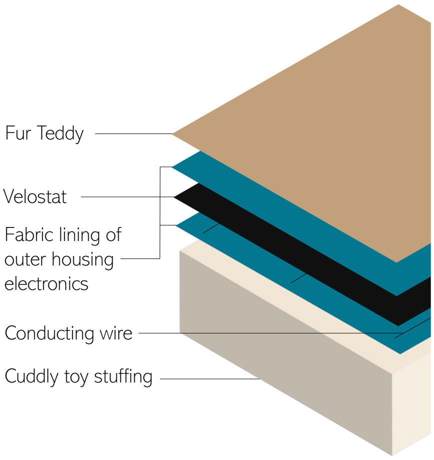


Figure 5.2: Schematic of the integration of the Velostat conductive sheet in the Teddy.

Figure 5.2 shows that the Velostat sheet is placed in between two linings of the outer housing. Using conductive thread, a connection is made between the Velostat and the wire[3] which connects the sheet to the breakout board. The equivalent circuit of the touch sensor can be seen in Figure E.2: the conductive Velostat sheet forms one plate of the capacitor. When the sheet is touched, the circuit treats the person applying the load as a virtual ground, forming the second plate of the capacitor. Depending on how much pressure is applied, the capacitance varies and the sensor is able to register touch based on this deviance.

---

[1] The Adafruit Stepper Motor Driver break-out board, containing the AT42QT1010 motor driver chip.

[2] The Momentary sensor lends it name to the fact that its output pin is only high when it detects a capacitive load (i.e. touch).

[3] As conductive thread is not isolated, we must connect it to a wire as to avoid short circuits.

5.1.2. Verification

bIn order to verify whether or not the movement sub-system meets the relevant requirements, various tests were done. Table B.2 shows the exact requirements together with a short statement on how, and if, they are verified. Furthermore, in Appendix F, the exact procedures can be found as to how the tests were carried out and their corresponding test results.

Movement

The functionality of the lungs and tails was tested by running a test script which triggered breathing and tail wagging. During testing, it became apparent that for a noticeable difference in the size of the lungs, the inhalation should last for at least 8 seconds and that the lungs should be deflated by using a vacuum pump. The lungs and tail have proven to function correctly.

Touch Sensor

In order to verify Requirement F.08 and F.09, the time between a registered touch and the actuation of movement has to be tested, and whether or not the touch sensor is able to actually register touch. For the latter, we tested the accuracy of the touch sensor by performing a load test, shown in Table F.2, using a sample size of n=25 for different loads. The response time in between triggering an interrupt caused by a registered touch and the actuation of movement was measured by placing a time stamp in the code: we made use of the tick function in MicroPython, which allows you to measure the time in between a pre-defined start- and stop command. The start signal was placed in the touch sensor interrupt handler, and the stop-signal was placed after calling the actuator function. A sample size of

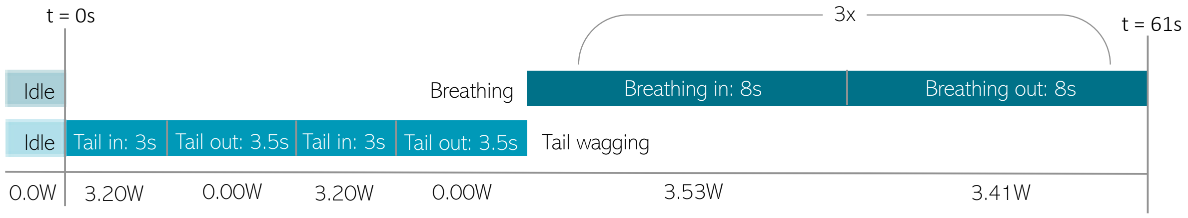


Figure 5.3: Overview of the movement cycle triggered by the touch sensor. Note that the figure is not scaled, the exact time intervals are mentioned in the figure.

Table 5.2: Measured power consumption of the movement sub-system based on a situational analysis.

Situation	Power consumption [W]	Max. current [A]
Idle (valves and pumps off, touch sensor not triggered)	0.00	0.000
Breathing in	3.53	0.706
Breathing out	3.41	0.681
Tail inflates	3.20	0.640
Tail deflates	0.00	0.000

5.2. Audio

This section contains the implementation and verification of the audio sub-system. The audio recording design choices, discussed in Section 3.3.1, are not implemented in the design: because MicroPython is a fairly new programming language, no libraries implementing speech recognition were available. Recommendations on how to possibly include speech recognition in the future are given in Chapter 6.

5.2.1. Implementation

Producing audio can be done is several ways: by generating sound using the Raspberry Pi Pico, an I2S module or by using a Bluetooth speaker. Table 5.3 contains a trade-off of all the possible options. Initially, the I2S module seemed the most promising, however, as MicroPython is fairly new, the are no available libraries yet. This means a library would have to be written, making it a high-complex option. If an I2S library was readily available, its score would have been equivalent to that of the play back board.

A Bluetooth speaker is heavy and power hungry compared to the other options, as it constantly requires wireless communication over Bluetooth. However, it would provide for the best sound quality. Another, more simple approach, is to use the PWM of the Pico. This solution would not require any external hardware other than a number of filtering components. Yet, its quality is low compared to the other options as it is limited by the filter design and the accuracy of the Pico. The playback board is an off the shelf solution: the board has local memory, on which WAV or MP3 files can be uploaded. Additionally, the board has a built-in amplifier to which a speaker driver can be directly connected. This solution keeps the design simple, easy to modify, and consumes virtually no memory on the Pico. Hence, it is the preferred implementation for the prototype.

Table 5.3: Comparison of the methods to implement audio production in the Smart Teddy using scoring on a scale of 1 to 5. Note that a high score is positive - e.g, a score of 5 for 'cost' means it is a highly low-cost option.

Implementation method	Cost	Weight	Power consumption	Quality	Complexity	Total
I2S module	5	4	5	4	2	20
Bluetooth speaker	3	2	2	5	4	16
PWM from Pico	5	5	5	2	4	21
Playback board	4	4	5	4	5	23

5.2.2. Verification

The complete demonstration procedure to verify whether the Teddy can produce sound, can be seen in Appendix F.6. However, the sound quality was low and needed fine tuning. Sadly, during integration of the playback board on the PCB, the playback board broke, making integration into the final design impossible. According to requirement F.03, the Teddy must be able to indicate its SoC - this should verified by depleting the battery to that the Teddy starts whining. Note that according to requirement F.09, the Teddy must react to human touch and speech. Hence, F.09 is only partially verified.

A more detailed test regarding the sound intensity can be found in Appendix F.6, measuring the sound intensity at different ranges with the speaker being placed both in- and outside the Teddy. The accuracy of the measurements is not high, as the ambient sound varied noticeably during the tests. At a distance of 0.5m, the difference in sound intensity with regards to the ambient sound was measured to be 5 ± 2 dB. This is 2 dB above the sound intensity of the pumps, meaning the user can hear the response of the Teddy. Therefore, it satisfies Requirement F.09. Furthermore, the added sound of the pumps does not surpass that off the speaker at any distance, making the audio produced the dominant sound. However, it does not meet Requirement NF.10 as the sound intensity of the pumps reaches 50 ± 2 dB.

5.3. Communication

In Section 4.1.2, the design choice was made to implement LoRa in order for the Teddy to send data to the Base Station. The next step regarding implementation is choosing which type of transceiver to use. Table 5.4 contains certain features of five types of LoRa transceivers: its operating voltage, operating frequency, the communication protocol, its sensitivity and its price. The frequency is an important aspect of the module, as there exist strict regulations in Europe[4] as to which frequencies are allowed for amateur use. The operating voltage of the transceiver and its communication protocol are used to determine the compatibility of the transceiver within the Teddy and in combination with the Pico. Table 5.5 shows the comparative analysis on the different LoRa transceivers. As can be seen, the LoRa RFM95SX1276 transceiver scores the highest, due to its low cost and good compatibility with the Pico.

---

[4] 868 MHz is approved for license-free use in Europe.

Table 5.4: Characteristics of the 5 LoRa transceivers.

Transceiver	Operating voltage	Frequency	Comm. Protocol	Sensitivity	Price
LoRa-e5 mini	3.7-5V	868, 915 MHz	SMA, USB-C	-136 dBm	€17,21
Adafruit RFM95W LoRa Transceiver	3.3-5.5V	433, 868, 915 MHz	SPI	-148 dBm	€17,82
SX1276 LoRa breakout board	3.3V	868 MHz	SPI	-	€13,49
LoRa RFM95 SX1276 transceiver	1.8-3.7V	868 MHz	SPI	-148 dBm	€7,49
LAMBDA62 LoRa Transceiver	1.8-3.7V	868, 915 MHz	SPI	-148 dBm	€9,60

Table 5.5: Comparison of the different LoRa transceivers rated on a scale from 1 to 5.

Transceiver	Compatibility	Sensitivity	Price	Total
LoRa-e5 mini	1	4	2	7
Adafruit RFM95W LoRa Transceiver	5	5	1	11
SX1276 LoRa breakout board	5	5	3	13
LoRa RFM95 SX1276 transceiver	5	5	5	15
LAMBDA62 LoRa Transceiver	5	5	4	14

5.3.1. Implementation

As the implementation of the wireless communication is in the form of peer-to-peer communication, there is no need to access the LoRaWAN[5]. Because of this, there is no restriction to the amount of data packets that are sent per day (the LoRaWAN network has a communication policy which restricts the transmitting time to 30 seconds per 24 hours). As the Base Station contains a Raspberry Pi, we need to establish a LoRa communication between the Pico and the Raspberry Pi. In order to configure the Pico with the LoRa transmitter, and the Raspberry Pi as the receiver, a Micropython library and Python library are used, respectively. The code can be found in Appendix G. To ensure a solid wireless connection, 868MHz antennas were purchased to connect to the LoRa module [6].

Collecting the data acquired in the Teddy is the responsibility of the Sensors & Data Acquisition subdomain. Once a deviation in the sensor readings is registered, this deviation is stored in an array with size 7. Every two hours, this array is to be sent wirelessly to the Base Station. The code for sending the array using LoRa to the Base Station can be seen in Appendix G.

5.3.2. Verification

In order to verify the reliability, efficacy and strength of the wireless communication, a number of tests were conducted - the exact tests and the test procedure can be found in Appendix F.4. A summary of the obtained results can be seen in Table 5.6. Up until a 100m range, all messages are received with a high accuracy (i.e. high % of successful transmissions). However, at the 500m range, more than 25% of the 25 sent messages were not received, and 39% of those messages were distorted. At a 1000m range, no messages were received. It must be noted that the tests were performed in an environment with a high density of flats and large buildings. It is very likely that in a more rural area, the maximum range is higher. In addition to the results shown in Table 5.6, the tests were repeated while the receiving LoRa module was on the third floor of a flat (as the assumption can be made that a lot of seniors live in flats). The results of these tests can be seen in the Appendix in F.

The signal strength shows an expected decline - the further away the receiver is from the transmitter, the lower the signal strength. At a range of more than 50m, the signal strength is less than -100 dB, which is considered a weak signal, and generally a 'usable' signal has a signal strength of above -85 dB, which in this case would be when the range is smaller than 50m. In order to ensure the correct

---

[5] Long Range Wide Area Network

[6] Initially wire antennas were implemented, however, they were not able to successfully transmit/receive data packets.

message is received, the same message will be sent three times: the odds of not (correctly) receiving a message is then down to 0.03% when assuming a 100m range (calculated using the cumulative binomial probability). The specific data for this test can be found in Table F.7.

Table 5.6: Test results for the implementation of LoRa communication between the Raspberry Pi Pico and the Raspberry Pi. Note that all tests were done with n=25 samples. More data can be found in the Appendix in Table F.7

Range	% received transmissions	% successful transmissions	Median signal strength (dB)	Standard deviation of transmission time (μs)
1m	100%	88%	-31.13	473
5m	100%	88%	-63.13	473
10m	100%	84%	-75.40	440
50m	100%	76%	-107.93	717
100m	100%	70%	-112.2	2832
500m	72%	61%	-114.87	7623
1000m	0%	0%	-	-

The power consumption of both the receiving and the transmitting LoRa module were inspected. Both the receiving and the transmitting module consume 0.023 W while in sleep mode. The transmitting module consumes 0.135 W while transmitting, while the receiving module does not show any change in power consumption when it is receiving messages.

5.4. Hardware System Integration

A crucial aspect of delivering a working prototype is the integration of all sub-systems: a high-level integration overview can be seen in Figure 5.4. The integration and its implementation must ensure the systems robustness, there being traceable and clear connections, and provide for housing and mounting in such a way that it fits the practical and aesthetic requirements of the Teddy. Section 5.4.1 contains a detailed description of the integration of all the different components shown in Figure 5.4, and Section 5.4.2 contains the verification of the hardware integration based off the relevant requirements.

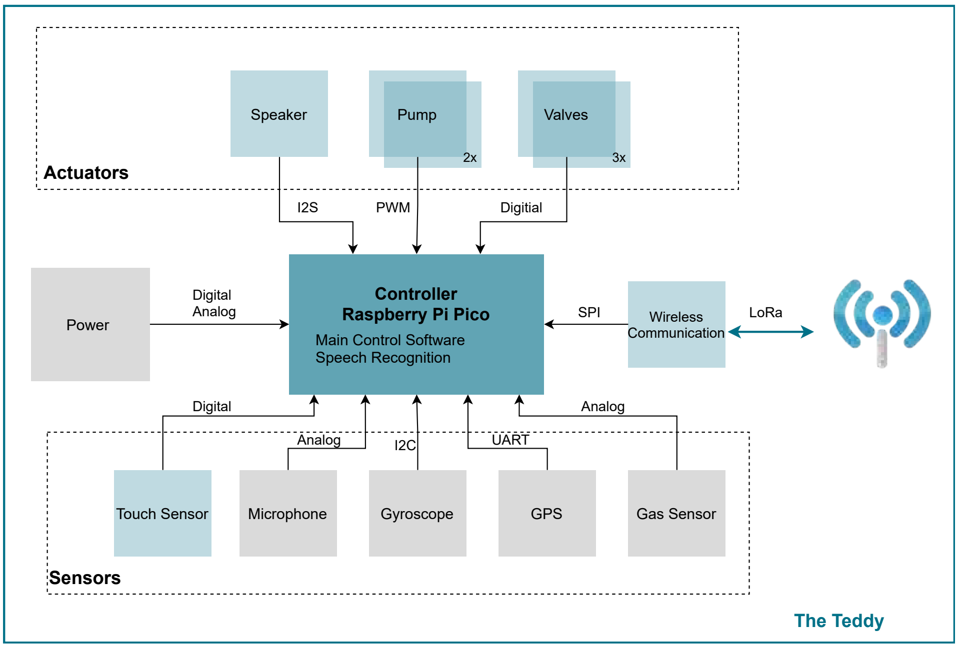


Figure 5.4: Overview of integration in the Smart Teddy, showing the different sensors, actuators and communication protocols inside the Teddy.

5.4.1. Implementation

If the timeline were to be longer, a printed circuit board (PCB) could be made with all the necessary converters and amplifiers directly placed on the board. However, as this is not the case, most conversion boards that allow for the implementation of the sensors and actuators are bought off the shelf. This greatly reduces complexity as the designs have already been tested, and many tutorials are available. For integration purposes though, it is much less efficient as the design now includes many small PCBs which all have to be connected to the Pico, the sensors and actuators. There are several options as to how this could be integrated. These options are displayed in Table 5.7.

Table 5.7: Trade-off on integration with the integration methods ranked from 1 to 5, 5 being the best option and highest score.

Integration Method	Robustness	Packaging	Flexibility	Cable management	Price	Total
Custom PCB	5	5	1	5	3	19
Prototype Board	4	4	4	2	4	18
Wired Connections	2	1	5	1	4	13

Table 5.7 shows that a custom PCB and a prototyping board are equally attractive options. Both create robust connections, and allow for a small packaging size. The custom PCB, however, scores much higher on cable management as all wires will be integrated into the board. The prototyping board allows for much for flexibility, but will be less clean when it comes to cable management. The final option, using wired connections, will likely lead to unmanageable cabling, but does provide added flexibility.

The benefits of the custom PCB outweigh those of the prototyping board and the wired connections. A PCB will therefore be used, which allows for direct mounting of the smaller of the shelf boards. The only sensors and actuators that will have a wired connection are those which require specific placement according to their functionality. The schematic of the PCB can be found in Figure E.1 in the Appendix, and an image of the board can be seen in Figure 5.5. An interesting feature is that both the Pico and the LoRa board have castellated holes, allowing for them to be surface mounted onto the PCB.

Power Consumption and Distribution

Two 5V power lines are fed to the Human Interaction & Integration system from the power system. One feeds the two air pumps and their motor driver, and the other feeds the valves, their motor driver, the Pico, and all the other sensors and actuators which require 5V. The Pico has an on-board converter to 3.3V, which is extended over the board as the 3.3V bus, which feeds the sensor and actuators that require 3.3V.

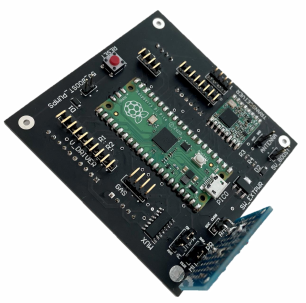


Figure 5.5: Isometric view of the PCB with the Pico soldered in the center and the transceiver on the right.

5.4.2. Verification

Each subsystem's power consumption has been tested with their results in their respective subsection in this chapter. In Table 5.8 the results of a full system power test was performed with all the sensors and actuators connected to the board. The maximum current, as well as the current consumption on idle were measured.

Table 5.8: Results from power consumption measurement during hardware integration tests.

Net	Active systems at max.	Max. current [A]	Active systems during idle	Idle current [A]
5 V pumps	One pump	0.640	Not active	0
5 V bus	One valve, speaker driver, drivers and breakout boards	2.047	Drivers and breakout boards	0.04
3.3 V bus	Raspberry Pi Pico, drivers and breakout boards	0.023	Raspberry Pi Pico	0.019

5.5. Software System Integration

This section contains the method of implementation of the control in the Teddy, in terms of software. Sections 5.5.1 and 5.5.2 contain the implementation and verification of the software, respectively.

5.5.1. Implementation

The main control of the Teddy contains, essentially, an infinite loop enabling the sensors to continuously read out sensor data. All functionalities within the Human Interaction & Integration are triggered by either the sensor data needing to be sent to the Base Station, or by input from the user or the power domain. As all Human Interaction & Integration functions are triggered by either human input or a flag in the code, the interrupt function can be used to enable a quick response. An interrupt can interrupt the main code at any given time, in response to a rising/falling clock edge of a selected pin input. When the rising/falling clock edge is detected, the interrupt handler is executed. For the software integration, the only interrupt implemented is for the touch sensor. The interrupt ensures the code for the movement sub-system will be executed almost immediately. The main code can be found in Appendix G.

Using the two cores of the Pico is the only way to execute functions in parallel, as the Pico does not contain an operating system that manages CPU cycles, and therefore is only able to run one thread per core [37]. Hence, multi-threading is needed to ensure reliability of the functionality: when breathing is triggered by human touch, the breathing and tail wagging must continue for one minute. As the situation could occur that the user is cuddling with the Teddy for a long time, it would mean that crucial sensor data readings could be missed. Hence, it was chosen to implement multi-threading for the execution of the breathing and tail-wagging motions of the Teddy. One could now question why we don't run, for instance, half of the all the control on the second core. However, as the functionality of the Smart Teddy relies heavily on (unpredictable) user input, it is extremely complex to divide the code on to two cores. On top of that, when a command is given to execute a function on the second core, but the second core is already in use, it causes an error.

5.5.2. Verification

The testing done on the software can be divided into two types; functional and non-functional testing. Functional testing is essentially requirement testing, whereas non-functional testing are additional performance tests to obtain a better understanding on how further improvements can be done.

Functional Testing

All individual sub-systems have been functionally tested - the corresponding results can be found in their respective verification subsections and in Appendix F. However, these tests have to be repeated after full integration of all sub-domains, to assure the functionality is not affected by the integration with the other domains. Table 5.9 contains the results of the full integration tests.

Table 5.9: Results from functional testing of the software after integration.

Sub-system	Pass/fail	Elaboration on results
Touch	Pass	The interrupt, that is triggered by a positive clock edge of the touch sensor, works.
Movement	Pass	In the interrupt handler of the touch triggered interrupt, the command to execute the movement on the second core is stated. This has proven to work as expected.
Audio production	Pass	As we wish to produce audio for just 2 seconds, the code for triggering an audio output is contained within the infinite loop in the main code, and functions as expected.
Communication	Pass	The code for transmitting of LoRa messages is evoked from within the infinite loop in the main code, and functions as expected.

Non- Functional Testing

The only non-functional performance test that is done, is determining the average execution time per function. This will allow for the next design iteration to schedule functions, or make use of the second core more efficiently. Additionally, the results were used to determine whether the interrupt-functions are short enough as to not disturb the monitoring functionality of the Teddy. For the exact test procedures, refer to Appendix F. The results show that the maximum execution time of the interrupt handler is 3ms, and is therefore negligible as it will not interfere with the monitoring functionality.

6

Discussion & Conclusion

The goal of the Smart Teddy project is, in short, to develop an interactive monitoring system, consisting of a Teddy and a Base Station. In this thesis, the contribution to the goal stated above was to create the sub-systems needed in order to stimulate human interaction with the Teddy, and to enable integration of all sub-domains into one functioning prototype. The latter goal is achieved, by designing and developing movement, communication, audio, hardware- and software-integration sub-systems. Out of all 20 requirements, 16 were met - for a detailed overview of which requirements were met, please refer to Appendix B. For each sub-system, the conclusion, discussion and recommendations for future work are given.

Movement

To enable movement, an approach using pnuematics was taken, making use of inflatable lungs and a tail. It was found that Ecoflex 00-30 was the perfect material for the construction of the lungs and tail. Using two 5V DC air pumps and three three-way valves, in- and deflation could be stimulated, resulting in a natural breathing and tail wagging motion of the Teddy. As Ecoflex is very flexible, the usage of it ensured we met Requirement NF.05. Although the visual and physical look of this movement sub-system is very life-like (meeting Requirement F.09, and NF.01), the air pumps produce a noticeable level of mechanical noise. For a later prototype, the option to include more silent air pumps (like piezo-electric air pumps) could be investigated, or the air pumps could be surrounded by some sort of sound-proofing material in order to meet Requirement NF.10.

It was chosen to trigger the movement of the Teddy with a pressure sensitive touch sensor. To achieve this, a break-out board containing a capacitive touch sensor IC, Velostat and conductive thread were used. The Velostat in combination with the conductive thread proved to be a very flexible for integration in a cuddly toy, and once placed inside the Teddy, were hardly noticeable: as a result, we were able to meet Requirement F.01, F.08, F.09, and NF.01. The only problem noticed during testing is that the wire connecting the break-out board to the Velostat is also pressure sensitive, causing unwanted triggering of the movement. It could be an option to investigate the optimal placement of the wire inside the Teddy, as to reduce unwanted triggers of the touch sensor.

Audio

Initially, the goal was to include both speech recognition and the production of audio in the Teddy. However, due to our choice of microcontroller, speech recognition was extremely complex to include. The Pico uses MicroPython as programming languages, and it was found later on in the project that there are no speech recognition libraries available in MicroPython. The latter caused us to decide not to implement speech recognition in this iteration of the Smart Teddy, meaning we were not able to meet Requirement F.02. In future research, the usage of other microcontrollers could be investigated (which are compatible with e.g. Python), which would make the implementation of speech recognition more straightforward.

We would have preferred to implement the production of audio using I2S, a communication protocol especially meant for audio-related purposes. However, the Pico does not have I2S integrated onto the board, making it time-consuming and complex to integrate (write own libraries, in MicroPyton and machine language). In the end, a playback module was chosen, on which MP3 and WAV files could be downloaded as it has on board flash memory and an amplifier. This proved to work together with the chosen speaker and the Pico (meeting Requirement F.03 and F.09), however, the quality of the sound was very low. In future work, the usage of I2S could be explored using other microcontrollers, or tests could be performed with better sound modules.

Communication

In our design phase, we made the deliberate choice to use I2C as communication protocol between the Pico and the other components. However, in practice, it became clear that a lot of sensors/modules support one or two communication methods, leaving little to no room for choice of communication protocol. In the end, we had to make use of UART, SPI and I2C to connect all components to the Pico. As this worked, we successfully met Requirement F.04 and F.06.

We chose LoRa as a wireless communication method, due to its long range, low price and low power consumption. In testing it became apparent that the nominal range is about 500 meters; for longer ranges, the accuracy decreased very fast and the messages that were received contained highly distorted messages. By successfully integrating LoRa as a wireless communication method between the Teddy and the Base Station, we met Requirement F.07. The only problem with LoRa is its security - as we do not make use of the LoRaWAN, we miss out on the added security this protocol encompasses. However, in future work, the libraries of the LoRa transceivers could be expanded, as to include e.g. handshakes, more specific client adresses and/or encryption to increase the security of the transmitted messages.

Software Integration

Interrupts were included in the main code, to enable almost immediate execution of the movement control code on the second core of the Pico whenever the touch sensor detected a touch (meeting Requirement F.08). In order to ensure parallel execution of reading the sensor data and the movement of the Teddy, multi-threading was used. Although the code functions well, the second core is not used at full capacity, as only the movement of the Teddy is executed there. For future iterations, usage of another microcontroller could be investigated, where parallel executions are an option: this way, the capacity of the cores can be used more optimally, making the functionality more efficient and possibly quicker.

Hardware Integration

For a robust and compact integration method of the hardware in the Teddy, a customised PCB was designed. Almost all the sensors, the touch sensor, antenna, the Pico and the LoRa module were mounted onto the PCB. Due to the time limit of ten weeks, we decided to purchase off the shelf modules to allow for rapid prototyping. This greatly reduced the number of wires and general ability to debug the design as this was rather time consuming when integrating on a bread board. For future prototypes, we would recommend creating an entire PCB with all the functionalities incorporated. This way, the design can be made lighter and more compact, allowing for integrated into an even smaller Tddy. Another interesting idea to keep the soft texture of the Teddy, would be to make use of flexible PCBs. This way it could form to the shape of Teddy, and be soft to the touch.

For high-level hardware integration, a cotton pouch was constructed in which all electrical components were placed. The pouch contained an inner pocket, in which the three rubber balls containing all components were placed. The outer pouch contained cuddly toy stuffing, as to meet Requirement NF.01 and NF.09. As the bottom of the Teddy contains an opening which can be easily opened/closed with Velcro, the entire pouch with all electronics can be removed from the Teddy, making it machine-washable and hence we were able to meet Requirement NF.06. The only component not within the pouch is the coil needed for wireless charging - although not in the scope of our sub-domain, we would recommend more research to be done for ways to incorporate wireless charging without the need for an exposed coil on the outside of the Teddy, for both safety and aesthetic reasons.

References

[1] Components 101. *NodeMCU ESP8266 Pinout Features and Datasheet*. URL: `https://components101.com/development-boards/nodemcu-esp8266-pinout-features-and-datasheet` (visited on 05/11/2021).

[2] Health in Aging. *Dementia*. URL: `https://www.healthinaging.org/a-z-topic/dementia`.

[3] AIBO. *Important Specifications AIBO*. URL: `http://aibo.sony.jp/feature/spec.html` (visited on 05/13/2021).

[4] Anson He. *Basic Electronics: Wired Communication Protocols in Embedded Design*. 2019. URL: `https://www.seeedstudio.com/blog/2019/07/03/basic-electronics-wired-communication-protocols-in-embedded-design/` (visited on 04/29/2021).

[5] Arduino. *Arduino Zero*. URL: `https://store.arduino.cc/arduino-zero` (visited on 05/11/2021).

[6] P. Musilek B. Lashkari Y. Chen. "Comparison of communication protocols for low cost Internet of Things devices". In: *Energy Management for Smart Homes - State of the Art*. 2017, pp. 1–6. DOI: `10.23919/SEEDA-CECNSM.2017.8088226`.

[7] Michael Beetz, Benjamin Johnston, and Mary-Anne Williams (Eds.) "Social Robotics. 6th International Conference, ICSR 2014, Sydney, NSW, Australia, October 27-29, 2014. Proceedings". In: 8755. Springer International Publishing, 2014. ISBN: 9783319119731. DOI: `10.1007/978-3-319-11973-1`.

[8] D. Boyle and T. Newe. "Security Protocols for Use with Wireless Sensor Networks: A Survey of Security Architectures". In: *2007 Third International Conference on Wireless and Mobile Communications (ICWMC'07)*. 2007, pp. 54–54. DOI: `10.1109/ICWMC.2007.80`.

[9] C. Bowler C. Ballard A. Boyle and J. Lindesay. "Anxiety Disorders in Dementia Sufferers". In: *International Journal of Geriatric Psychiatry* 11 (1996), pp. 987–990.

[10] E. A. Cheikh C. Saad B. Mostafa and H. Abderrahmane. "Comparative Performance Analysis of Wireless Communication Protocols for Intelligent Sensors and Their Applications". In: *International Journal of Advanced Computer Science and Applications* 5.4 (2014).

[11] Gerald Coley. *Beagle Bone Black System Reference Manual*. URL: `https://github.com/beagleboard/beaglebone-black/wiki/System-Reference-Manual` (visited on 05/11/2021).

[12] Y. Chen D. Mocrii and P. Musilek. "Internet of Things". In: *IoT□based smart homes: A review of system architecture, software,communications, privacy and security*. Vol. 1. 2018, pp. 81–98. DOI: `10.1016/j.iot.2018.08.009`.

[13] Mihai Daraban and Dan Pitica. "Statistical information of crosstalk on parallel bus". In: *2011 IEEE 17th International Symposium for Design and Technology in Electronic Packaging (SIITME)*. 2011, pp. 211–216. DOI: `10.1109/SIITME.2011.6102720`.

[14] *Difference between serial and parallel communication*. URL: `https://circuitglobe.com/difference-between-serial-and-parallel-communication.html` (visited on 04/29/2021).

[15] Control Engineering. *Robustness and reliability in CAN-based networks*. URL: `https://www.controleng.com/articles/robustness-and-reliability-in-can-based-networks` (visited on 05/16/2021).

[16] Espressif. *ESP32-S2 DevKit M-1 Programming Guide*. URL: `https://docs.espressif.com/projects/esp-idf/en/latest/esp32s2/hw-reference/esp32s2/user-guide-devkitm-1-v1.html` (visited on 05/11/2021).

[17] Interview with Ezra van Zadelhoff and Annemieke. by Laura Croes, Shea Haggerty, Lyana Usa, Taha Küçükçelebi, Tim van der Spijk and Alan Hamo, April 23, 2021.

[18] E. Ferro and F. Potorti. "Bluetooth and Wi-Fi wireless protocols: a survey and a comparison". In: *IEEE Wireless Communications* 12.1 (2005), pp. 12–26. DOI: `10.1109/MWC.2005.1404569`.

[19] André Glória, Francisco Cercas, and Nuno Souto. “Comparison of communication protocols for low cost Internet of Things devices”. In: *2017 South Eastern European Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*. 2017, pp. 1–6. DOI: 10.23919/SEEDA-CECNSM.2017.8088226.

[20] H. Hasson and J. E. Arnetz. “Nursing staff competence, work strain, stress and satisfaction in elderly care: a comparison of home-based care and nursing homes”. In: *J Clin Nurs* 17.4 (2008), pp. 468–481.

[21] IEEE. *Robots IEEE - Paro*. URL: https://robots.ieee.org/robots/paro/ (visited on 05/06/2021).

[22] Texas Instruments. *MSP430G2553 LaunchPad™ Development Kit (MSP□EXP430G2ET)*. URL: https://www.ti.com/lit/ug/slau772a/slau772a.pdf?ts=1620649044496&ref_url=https%253A%252F%252Fwww.google.com%252F (visited on 05/11/2021).

[23] Ioan E. Lager, Koen Bertels, Victor Scholten, Egbert Bol, Cristina Richie and Seyedmahdi Izadkhast. *Bachelor Graduation Project*. Delft, the Netherlands: TU Delft, 2020.

[24] H. Hauser J.M. Rossiter. “Soft Robotics - The Next Industrial Revolution?” In: *IEEE Robotics and Automation Magazine* 23.3 (2016), pp. 17–20. DOI: 10.1109/MRA.2016.2588018..

[25] W. Janig. *Autonomic Nervous System*. 8th ed. Berlin, Heidelberg: Springer, 1989. DOI: https://doi.org/10.1007/978-3-642-73831-9_16.

[26] Ali K. Kamrani and Maryam Azimi. “Systems Engineering Tools and Methods”. In: Boca Raton: CRC Press, 2010, pp. 85–86. ISBN: 9781439809266. DOI: 10.1201/b10452. URL: https://doi-org.tudelft.idm.oclc.org/10.1201/b10452.

[27] Keiichi Kemmotsu, Yoshihiro Koketsu, and Masato Iehara. “Human behavior recognition using unconscious cameras and a visible robot in a network robot system”. In: *Robotics and Autonomous Systems* 56.10 (2008), pp. 857–864. DOI: https://doi.org/10.1016/j.robot.2008.06.004.

[28] Elvira et al. Lara. “Are loneliness and social isolation associated with cognitive decline?” In: *International journal of geriatric psychiatry* 34.11 (2019), pp. 1613–1622.

[29] I. Lee and K. Lee. “The Internet of Things (IoT): Applications, investments, and challenges for enterprises”. In: *Business Horizons* 58.4 (2015), pp. 431–440. DOI: https://doi.org/10.1016/j.bushor.2015.03.008.

[30] Megadroid. *Necoro*. URL: https://www.megadroid.com/Robots/necoro.htm (visited on 05/13/2021).

[31] T. Mitsui et al. “Psychophysiological Effects by Interaction with Mental Commit Robot”. In: *International Conference on Intelligent Robots and Systems* (Oct. 2001).

[32] ERCIM News. *iCat - A Friendly Robot that Helps Children and Grown-Ups*. URL: https://ercim-news.ercim.eu/en67/special-theme-embedded-intelligence/icat-a-friendly-robot-that-helps-children-and-grown-ups (visited on 05/13/2021).

[33] K. Numbers and H. Brodaty. “The effects of the COVID-19 pandemic on people with dementia”. In: *Nature Reviews Neurology* 17.2 (2021), pp. 69–70.

[34] A. Olsson et al. “Sensor technology more than a support”. In: *Scandinavian Journal of Occupational Therapy* 25.2 (2018), pp. 79–87.

[35] E. Park and J. Lee. “I am a warm robot: the effects of temperature in physical human-robot interaction”. In: *Robotica* 32 (2013), pp. 133–142. DOI: doi:10.1017/S026357471300074X.

[36] Raspberry Pi. *Raspberry Pi 4 Computer Model B*. URL: https://datasheets.raspberrypi.org/rpi4/raspberry-pi-4-product-brief.pdf (visited on 05/11/2021).

[37] Raspberry Pi. *Raspberry Pi Pico*. URL: https://datasheets.raspberrypi.org/pico/pico-product-brief.pdf (visited on 05/11/2021).

[38] Martha Pollack et al. “Pearl: A Mobile Robotic Assistant for the Elderly”. In: (June 2002).

[39] RIVM. *RIVM verkent toekomstige ontwikkelingen in zorgvraag, bredere determinanten van gezondheid en technologie*. 2017. URL: https://www.rivm.nl/nieuws/rivm-verkent-toekomstige-ontwikkelingen-in-zorgvraag-bredere-determinanten-van-gezondheid-en (visited on 04/28/2021).

[40] H. Robinson, B. Macdonald, and E. Broadbent. “The Role of Healthcare Robots for Older People at Home: A Review”. In: *International Journal of Social Robotics* 6.4 (2014), pp. 575–591.

[41] A. Saito and K. Shinozuka. “Vocal recognition of owners by domestic cats (Felis catus)”. In: *Anim Cogn* 16 (2013), pp. 685–690. DOI: `https://doi.org/10.1007/s10071-013-0620-4`.

[42] Seedstudio. *RS232 vs RS485 – What are their differences?* URL: `https://www.seeedstudio.com/blog/2019/12/06/what-is-rs485-and-its-difference-between-rs232/` (visited on 05/06/2021).

[43] S. Selvaraj and S. Sundaravaradhan. “Challenges and opportunities in IoT healthcare systems: a systematic review”. In: *SN Applied Sciences* 2.1 (2020), p. 139. DOI: `https://doi.org/10.1007/s42452-019-1925-y`.

[44] *Serial Protocols Compared*. URL: `https://www.embedded.com/serial-protocols-compared/` (visited on 04/29/2021).

[45] Takanori Shibata and Joseph Coughlin. “Trends of Robot Therapy with Neurological Therapeutic Seal Robot, PARO”. In: *Journal of Robotics and Mechatronics* 26 (Aug. 2014), pp. 418–425. DOI: `10.20965/jrm.2014.p0418`.

[46] Somnox. *The Science of Sleep*. URL: `https://somnox.com/science-research/`.

[47] *UART vs I2C vs SPI – Communication Protocols and Uses*. URL: `https://www.seeedstudio.com/blog/2019/09/25/uart-vs-i2c-vs-spi-communication-protocols-and-uses/` (visited on 04/29/2021).

[48] WHO. *Dementia*. URL: `https://www.who.int/news-room/fact-sheets/detail/dementia` (visited on 05/30/2021).

[49] Wikipedia. *Bus contention*. URL: `https://en.wikipedia.org/wiki/Bus_contention` (visited on 05/06/2021).

[50] Wikipedia. *Cyclic Redundancy Check*. URL: `https://en.wikipedia.org/wiki/Cyclic_redundancy_check` (visited on 05/16/2021).

[51] Wikipedia. *Pneumatics*. URL: `https://en.wikipedia.org/wiki/Pneumatics`.

[52] Elliot Williams. *What could go wrong: SPI*. URL: `https://hackaday.com/2016/07/01/what-could-go-wrong-spi/` (visited on 05/06/2021).

[53] R. S. Wilson et al. “Loneliness and Risk of Alzheimer Disease”. In: *Archives of General Psychiatry* 64.2 (2007), p. 234.

[54] B. Mutlu Y. Terzioglu and E. Sahin. “Designing Social Cues for Collaborative Robots: The Role of Gaze and Breathing in Human-Robot Collaboration”. In: *Human Robot Interaction* (Mar. 2020), pp. 15–16.

[55] Shunsuke Yanaka, Takayuki Kosaka, and Motofumi Hattori. “ZZZoo Pillows: Sense of Sleeping alongside Somebody”. In: *SIGGRAPH Asia 2013 Emerging Technologies*. SA ’13. Hong Kong, Hong Kong: Association for Computing Machinery, 2013. ISBN: 9781450326322. DOI: `10.1145/2542284.2542301`. URL: `https://doi.org/10.1145/2542284.2542301`.

[56] Naoto Yoshida and Tomoko Yonezawa. “Investigating Breathing Expression of a Stuffed-Toy Robot Based on Body-Emotion Model”. In: *Proceedings of the Fourth International Conference on Human Agent Interaction*. HAI ’16. Biopolis, Singapore: Association for Computing Machinery, 2016, pp. 139–144. ISBN: 9781450345088. DOI: `10.1145/2974804.2974817`. URL: `https://doi.org/10.1145/2974804.2974817`.

A

The Prototype

The final prototype was still being worked on as this thesis had to been submitted. The functionality of the prototype is as documented in this thesis, but further improvements on quality and reliability will done before the final demonstration. Table A.1 contains the main specifications of the prototype and Table A.2 contains links to videos demonstrating the features of the Smart Teddy. Lastly, pictures of the project are added below to give a better idea of what the prototype looks like.

Note that a breakdown of the weights of the components of the Human Interaction & Integration components can be found in Tables A.3 and A.4. For determining the total weight the values for the aforementioned tables were added to the total weights of the Sensors & Data Acquisition and Power sub-systems.

Table A.1: Main specifications of the prototype

Total weight:	1.503 kg.
Total maximum power consumption:	7.11 W
Size:	60 cm head to tail, 40 cm body, 20 cm width body, 20 cm height body
Battery life:	12 hours
Working data acquisition:	GPS, microphone, gas sensor, gyroscope

Table A.2: Videos of demonstrations of the Smart Teddy

Breathing and tail wagging demonstration	`https://youtu.be/irsqKesD8Cs`
LoRa testing	`https://youtu.be/CoSEh-Zduxc`
Movement demonstration on PCB	`https://www.youtube.com/watch?v=z2myGH-bBQg`

Figure A.1: Assembly overview of the Teddy. Looking from top to bottom you can see the teddy, the pouch, the mounting/ integration balls, and the PCB, the batteries, and the pumps.

Figure A.2: The cuddly toy dog used for the Smart Teddy prototype.

Figure A.3: The inner pouch containing the integration balls inside the outer pouch.

Figure A.4: The outer pouch, containing stuffing on the inside. The outer pockets hold the lungs in place.

Figure A.5: The inner stuffing of the Teddy is sown in place, as to avoid stuffing spilling out of the Teddy when the pouch is removed.

Figure A.6: The inner pouch containing the rubber integration balls.

Figure A.7: The outer pouch contains a pocket for the Velostat sheet, with velcro to keep the sheet in place.

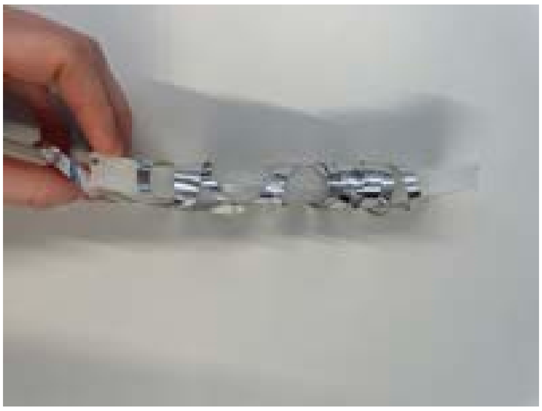


Figure A.8: The tail when deflated.

Figure A.9: The tail when inflated.

A.1. Materials and Components

Table A.3 shows the electrical components used inside the Teddy, along with the quantity of the components, the total weight and the total price.

Table A.3: All components used in the Human Interaction & Integration sub-domain.

Component	Quantity	Total weight [grams]	Total price
AT42QT1010 Touch Sensor Breakout Board	1	4	€5,95
6V air valve	3	24	€7,50
5V DC air pump	2	120	€13,90
Motor driver	2	19	€17,90
Raspberry Pi Pico	1	8	€4,00
Conductive thread	1	5	€9,95
Velostat pressure sensitive conductive sheet	1	3	€4,37
Connector headers	6	6	€5,00
868MHz antenna	1	22	€5,94
SMA to UFL mount connector	1	1	€0,21
Coaxial cable	1	4	€4,85
Customised PCB	1	20	€12,40
Playback board	1	3	€4.95
Total		**239**	**€90,52**

Table A.4 shows the non-technical components used inside the Teddy, along with the quantity of the components, the total weight and the total price. Note that some prices have been roughly estimated, e.g. for the Ecoflex silicon rubber components: the total costs for the Ecoflex were around €40,-, however, to make the tail and lungs about a fourth of the Ecoflex was needed.

Table A.4: All non-technical materials used in the Human Interaction & Integration sub-domain.

Component	Quantity	Total weight [grams]	Total price
Outer housing	1	150	€8,00
Inner housing	1	20	€4,00
Dog teddy bear	1	610	€12,95
Balls for integration components	3	63	€18,00
Ecoflex 00-30 Rubber Silicon: Lungs	2	104	€4,00
Ecoflex 00-30 Rubber Silicon: Tail	1	60	€4,00
Tube connectors	3	3	€9,00
Zip ties	8	16	€2,00
Silicon tubing	5	30	€2,20
Heat shrinks and cabling	2	2	€2,00
Total		**1058**	**€65,15**

B

Requirement Verification Overview

B.1. Non Functional Requirements Verification

Table B.1: Table containing nonfunctional design requirements with verification methods and results.

ID	Nonfunctional Requirement	Verification Method	Result
NF.01	The Teddy must not appear to be a machine.	Demo	By using Ecoflex and air, the breathing and wagging of the tail look very life-like. Additionally, by using a soft pouch which contains all components, no deformations/hard objects can be seen or felt. Hence, the requirement is met.
NF.02	The Teddy must not contain parts that could be a choking hazard.	Inspection	The Teddy does not contain glass eyeballs or any loose components on the outside. Velcro in stead of a zipper is used on the outside of the Teddy for this reason. Thus, the requirement is met.
NF.03	The Teddy must weigh less than 2.0 kg.	Analysis	The Teddy was weighed containing all its components and parts, and the total weight was 1.503 kg. Hence, the requirement is met.
NF.04	The Teddy must be designed in such a way that overheating of the components inside of the Teddy shall not occur.	Analysis	The only components which produce heat are the buck-/boost converters and the battery inside the Teddy. As we make use of airflow in the Teddy, we positioned the valves in such a way that they blow air on these components. In testing it became apparent that the components do not overheat. The requirement is met.
NF.05	Moving parts must never become potentially dangerous for the user.	Test	As the tail and are the only moving parts, the requirement is met. This is due to the fact that the tail and lungs are soft and will not harm the user.
NF.06	The Teddy must be safely machine washable.	Demo	As all components (except the tail and wireless charging coil) are included in the pouch, which is removeable, the Teddy is fully machine washable. Hence, the requirement is met.

NF.07	The Teddy must be robust (i.e. must be able to fall without breaking).	Demo	All components are enclosed in the pouch, which contains a considerable amount of padding. The tail is virtually indestructible, as it is made from Ecoflex. When the Teddy is dropped from a hip/chest height, nothing breaks. Hence, the requirement is met.
NF.08	The measurements of torso of the Teddy should range between 30-40cm (length) x 15-20cm (width).	Inspection	The measurements of the torso of the Teddy are 34x20cm, hence the requirement is met.
NF.09	The user should not be able to feel hard objects inside the Teddy.	Demo	By making use of Velostat as a touch sensor indicator, Ecoflex for the construction of the tail and lungs, and by creating a custom pouch containing padding around the electronic components, the Teddy feels soft to the touch. Hence, the requirement is met
NF.10	The movements of the Teddy should preferably be silent (not more than 20 dB).	Demo	The movements of the Teddy produce a sound level of 48.8dB at a 1 m distance. Hence, the requirement is not met.
NF.11	The manufacturing costs of the prototype Teddy should preferably not exceed €400.	Analysis	By inspecting our own components and their price, and adding the prices of the components from the other sub-group, a prototype price of €300,- was determined. Hence, the requirement is met.

B.2. Functional Requirements Verification

Table B.2: Table containing functional design requirements with verification methods and results.

ID	Functional Requirement	Verification Method	Results
F.01	The Teddy shall be able to sense when stroked/touched.	Test	In Appendix Section F.2 can be read that the touch sensor was tested and proved to work. Hence, the requirement is met.
F.02	The Teddy shall recognise 2-3 words using speech recognition.	Test	As there are no existing MicroPython libraries for speech recognition, the choice was made to not integrate speech recognition. Hence, the requirement is not met.
F.03	The Teddy must contain an indicator of the state of the battery.	Test	When the SoC of the battery becomes low, the Teddy will start whining. The test can of the whining can be found in Appendix F. The requirement is met.
F.04	The Teddy shall be a stand alone system which doesn't require control from external sources.	Inspection	As we use a Raspberry Pi Pico to perform the control within the Teddy, no external control is needed. Hence, the requirement is met.

F.05	There shall be an indication that the Teddy is charging that is in agreement with NF.01	Test	As this indication is not possible to implement in the Teddy (the Teddy will 'switch off' when charging), the indicator is outside of the scope of this sub-domain as it must be implemented in the Base Station. There is no indicator on the Base Station to indicate whether or not the Teddy is charging. Hence, the requirement is not met.
F.06	The components inside the Teddy shall be able to communicate with the main controller.	Test	As can be read in Appendix Section F.4, LoRa has proven to be an effective way for the Teddy to communicate wirelessly with the Base Station (within a range of 500 meters). Thus, the requirement is met.
F.07	The components inside the base station shall be able to communicate with the main controller.	Inspection	Using SPI, UART and I2C, we made sure all components inside the Teddy can communicate with the Raspberry Pi Pico. Hence, the requirement is met.
F.08	The response time in between touching the Teddy and the reaction of the Teddy should be no more than 0.5 seconds.	Test	As can be read in Appendix Section F.2, the response time between touching the touch sensor and actuator excitation is almost immidiate, thus less than 0.5 seconds. Hence, the requirement is met.
F.09	The Teddy should be able to react to human touch and speech (certain words) using movement and sound.	Inspection	We successfully developed an air-induced movement and a touch sensor. However, we were not able to implement speech recognition. Hence, the requirement is partially met.

C

Stakeholder Analysis

Table C.1: Table containing the stakeholders surrounding the defined problem and their influence on the solution the Smart Teddy provides.

Stakeholder	**Role in the problem**	**Influence on the solution (Smart Teddy)**	**Level of influence on solution**
Dementia patients	The increasing number of newly diagnosed dementia patients poses a strain on the health care system and on the mental and physical well being of not only their selves, but also their loved ones.	The main priority of the solution is to provide for the needs and wants of the patient suffering from dementia being met, as the goal is for the solution to perfectly blend in their daily routines and offer them support and comfort.	High
Loved ones of the dementia patient	Loved ones of the dementia patient often play a large role, as they want the best for their parent/relative/loved one and often play the role of carer of the dementia patient [17].	They will most likely be the purchasing party of the proposed solution, and decide even before the patient if the solution is good enough for their parent/relative/friend.	High
Health insurance companies	Health insurance companies play a large role in the decision whether or not a dementia patient moves to a care home, purely from a financial perspective.	If health care companies see the added value of the proposed solution, they might introduce a financial compensation for purchasing parties, making the solution a more widely financially accessible option.	High
Carers of the dementia patient	Carers often play a role in deciding whether or not the dementia patient is still capable of living independently [17]. On top of that, they are the party in the health care system who are in direct contact with the dementia patients.	Most carers will have a lot of experience in working with dementia patients, and might fulfill an advisory role towards the purchasing party of the solution.	Moderate

Doctors specialising in dementia	Doctors/specialists form the party responsible for finding a cure/slowing down dementia, hence, they play a large role in (trying to solve) the problem. They know the most about the disease.	Some purchasing parties might consult a doctor/specialist when deciding if the proposed solution is of added value to the dementia patient. Furthermore, the health care companies will most likely consult specialists in order to decide on adding the proposed solution to insurance plans.	Moderate
Academic instances	Academic instances play a role in trying to solve the problem, as they form a large percentage of research centres where research is done towards curing/slowing down dementia and improving the quality of life of dementia patients.	The funding from academic instances for projects like the proposed solution can make the difference in whether or not the solution is further developed.	Moderate
Designers developing (similar) solutions	This party plays a role in reducing the strain on the health care system by developing products to offer dementia patients either companionship, or to monitor their quality of life.	These researching/developing parties might pose a threat for the retail perspective of the proposed solution, as there might be another party who markets approximately the same product before the launch of the proposed solution.	Moderate
Care homes	Care homes play a large role in the problem, as they are often under-funded and under-staffed, leading to the mental and physical strain of both dementia patients and care home staff.	The care home might become a purchasing party of the proposed solution, to reduce the strain on staff and provide their residents with constant companionship.	Low

D

Wired Communication

In order to choose the most fitting digital communication protocol for the Smart Teddy, a list of basic criteria must be set up. As there exists a finite amount of communication protocols, the communication protocols will be compared to one and other, resulting in some criteria being comparative in stead of quantitative. The criteria are:

- To allow multiple devices to communicate back and/or forth with the controller;
- To minimise the hardware needed for the implementation of the communication protocol, in order to comply with Requirements NF.02 and NF.09;
- To minimise the risk of bus contention [1] or cross-talk [2] ;
- To maximise the speed of communication, in order to comply with Requirement F.08;
- To minimise the complexity of implementing the protocol: not only due to time constraints, but also to enable easy and quick implementation of changes and updates in later prototypes of the Smart Teddy.
- To be compatible with the components and devices used in the sub-domains 'Power Operations & Distribution' and 'Data Acquisition & Integration' of the Smart Teddy system.

An important aspect to address before investigating communication protocols is deciding whether to make use of a parallel or a serial interface. Serial communication makes use of a single communication link to send data, while parallel communication makes use of multiple parallel links [14]. Although parallel communication is quicker, serial communication is generally cheaper, requires less hardware (thus is more lightweight), avoids the risk of cross-talk and greatly simplifies upgrading the Smart Teddy circuitry in future prototypes [14]; these features of serial communication are in line with Requirements NF.03, NF.09 and NF.11, thus using serial communication is the preferred option.

The serial communication methods which are investigated can be seen, together with their properties, in Table D.1. We examine I^2C, SPI, CAN, 1-Wire and Microwire. Note that we also need an UART[3] connection, as the Sensors & Data Acquisition sub-domain make use of GPS, which is (due to their design choice regarding the GPS device) only configurable with UART. As UART is definitely needed for the integration of the sub-domains, we consider compatibility with UART as a criterion for the micro-controller. Using the information given in Table D.1, a comparison is made between the communication protocols, which can be seen in Table D.2.

Table D.2 shows that I^2C scores the highest. I^2C requires just two wires (serial data line SDA, and serial clock line CLK), is straightforward in its implementation and is widely supported - almost every micro-controller or sensor which was analysed supports I^2C. SPI and CAN are close runner-ups. However,

---

[1] Bus contention is an undesirable state where more than one device on a bus attempts to place values on the bus at the same time [49].

[2] Cross-talk is a signal integrity issue that appears due to parasitic capacitive/inductive coupling between adjacent wires [13].

[3] UART is not a communication protocol, but a physical circuit in a micro-controller [42]. UART stands for Universal Asynchronous Receiver and Transmitter.

the difference in required hardware between SPI and I^2C is large, doubling the amount of wires needed. Although SPI is a lot faster than I^2C, minimising the amount of hardware in the Teddy contributes to Requirement NF.03 and NF.09, which is of higher priority than Requirement F.08 (which regards the response time of the Teddy).

The CAN protocol is an extremely robust protocol due to its built-in failure detection mechanisms [15]. However, it is considerably slower than I^2C and is less widely supported for our implementation purposes. Hence, I^2C is preferred over CAN.

Table D.1: Different serial communication protocols and their properties [44].

Protocol	Type	(A)synchronous	Pin count	Duplex	Peripheral devices allowed	Max speed (Kbits/sec)
I^2C	Multi-source	Synchronous	2	Half	128 [47]	400
SPI	Multi-source	Synchronous	4	Full	Limit defined by bus capacitance/bit rate	>1000
Microwire	Source/sink	Synchronous	4	Full	Idem	>625
1-Wire	Source/sink	Asynchronous	1	Half	Idem	16
CAN	Multi-source	Synchronous	2	Full	Idem	40-125

Table D.2: Comparison of the selected serial communication protocols using scoring on a scale of 1 to 5 [19][42][52][47].

Protocol	Required area	Robustness	Speed	Complexity	Widely supported	Total
I^2C	4	4	3	5	5	21
SPI	2	4	4	3	5	18
Microwire	2	4	3	3	4	16
1-Wire	5	4	1	4	2	16
CAN	4	5	2	4	4	19

E

Schematics

Figure E.1 shows the schematic of the custom made PCB. As can be seen in the Figure, the Raspberry Pi Pico and the LoRa module are mounted directly on the PCB. Using 90 degree header connectors, the sensors, antenna and speaker module are attached to the PCB.

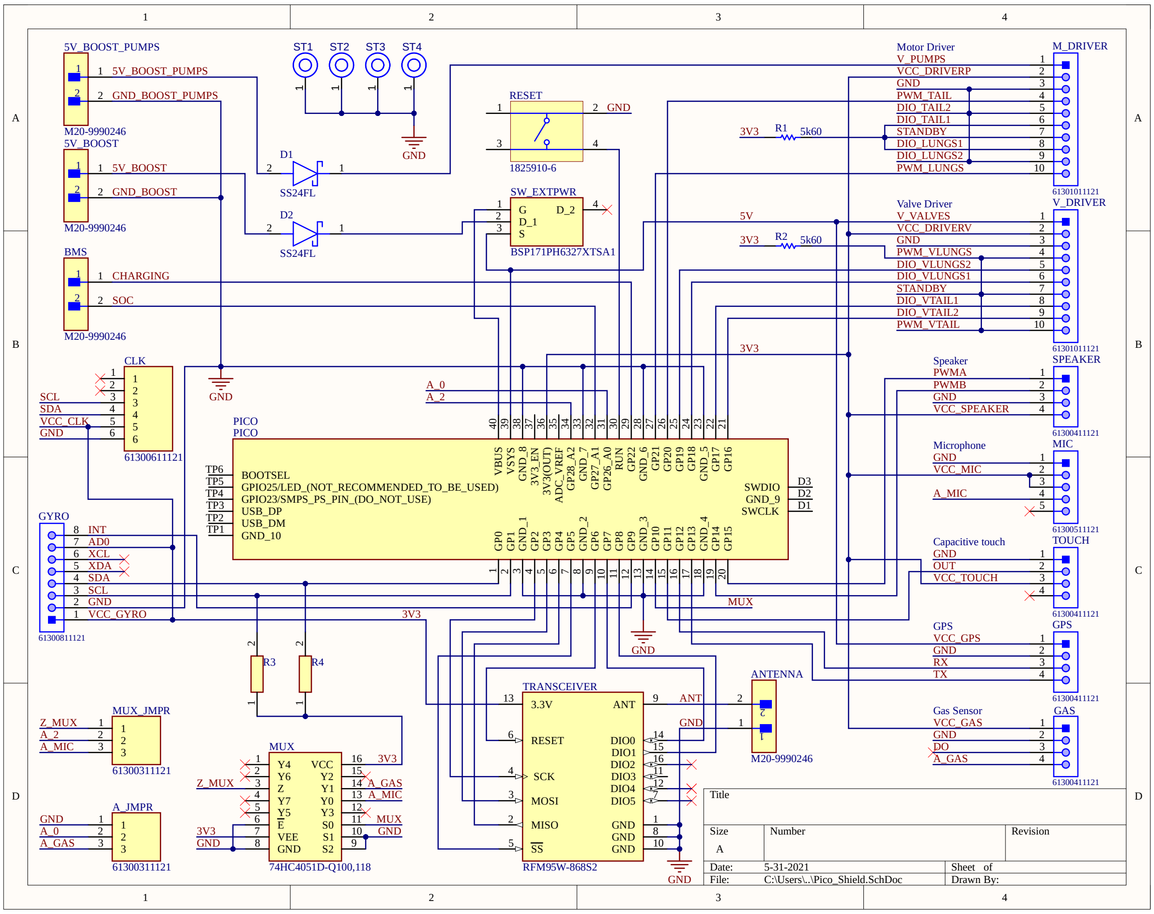
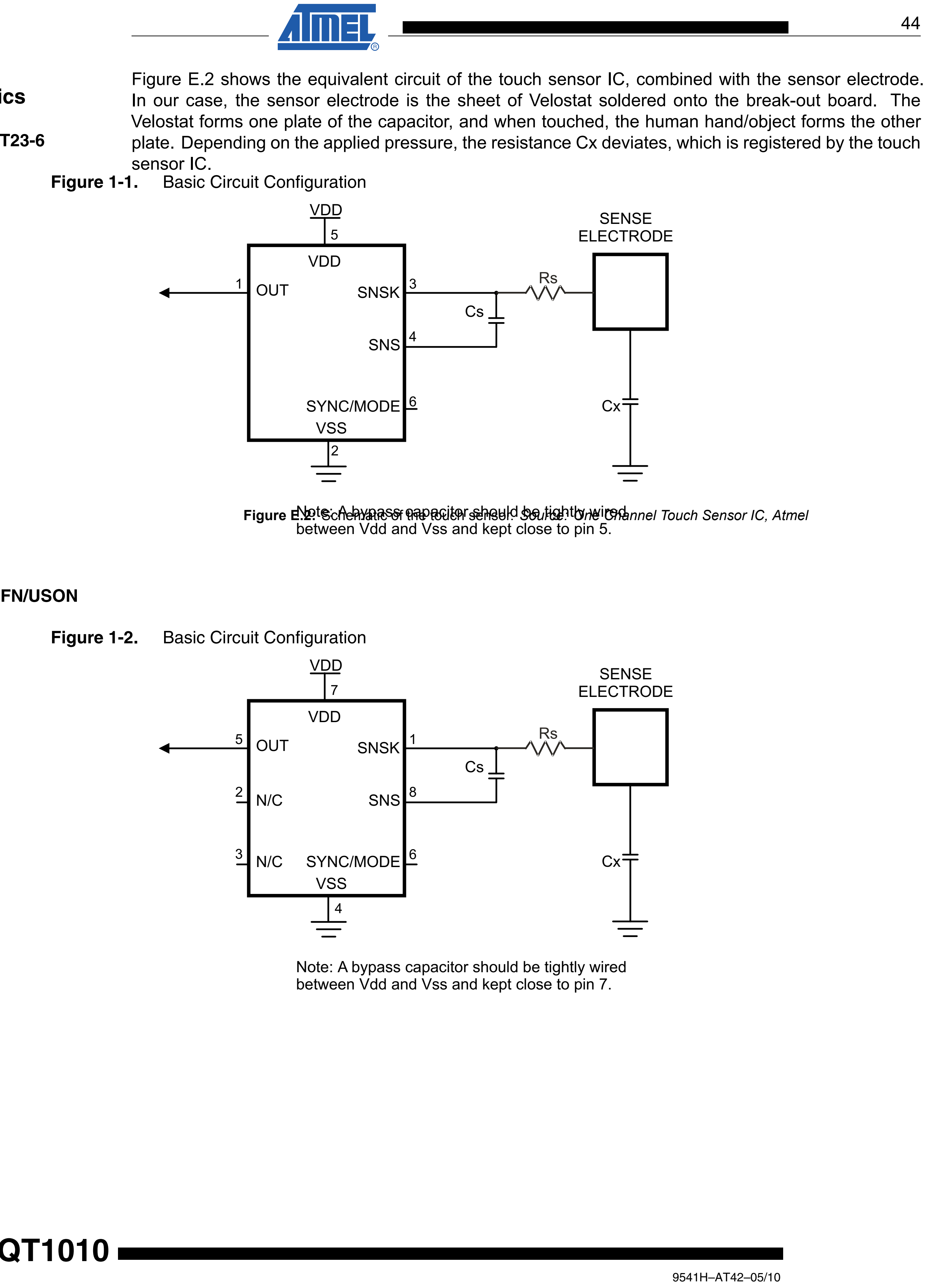


Figure E.1: Schematic of integration PCB

Figure E.2 shows the equivalent circuit of the touch sensor IC, combined with the sensor electrode. In our case, the sensor electrode is the sheet of Velostat soldered onto the break-out board. The Velostat forms one plate of the capacitor, and when touched, the human hand/object forms the other plate. Depending on the applied pressure, the resistance Cx deviates, which is registered by the touch sensor IC.



ics

T23-6

Figure 1-1. Basic Circuit Configuration

Note: A bypass capacitor should be tightly wired between Vdd and Vss and kept close to pin 5.

FN/USON

Figure 1-2. Basic Circuit Configuration

Note: A bypass capacitor should be tightly wired between Vdd and Vss and kept close to pin 7.

QT1010

9541H–AT42–05/10

Figure E.2: Schematic of the touch sensor. *Source: One Channel Touch Sensor IC, Atmel*

F

Test Procedures

F.1. Movement Performance Test

Test specialist:	Shea Haggerty
Date of completion:	07/06/2021
Test objective:	Test the functionality of the movement, and use test results to fine-tune the variables in the airflow control code to optimise breathing/tail wagging efficacy.
Results:	Both breathing and tailwagging can be achieved with the tested setup. In order for the best breathing motion, the pumps should inflate for 8 seconds and deflate for 8 seconds as well. The tail should inflate for 3.5 seconds and deflate for 3 seconds.

Table F.1: Summary movement performance test.

F.1.1. The Test Setup and Procedure

Figure F.1 shows the test setup used to perform the movement tests. One motor driver is used for driving the two 5V air pumps, and one motor driver is used for the three 5V valves. The two motor drivers are connected to 3.3V from the Pico for the on-board logic, and the VM pin on the boards are connected to a 5V power supply, which is used to power the pumps and valves. The two pumps are driven with a PWM signal (2000 Hz, 100% duty cycle) to reduce the mechanical noise they produce when turned on. The valves are controlled with a digital output pin from the Pico, and are connected to the MOTOR A/B pins on the motor driver and the common ground. The code for the control of the pumps and valves, together with a self-written library, can be found in Appendix G.

F.1.2. Test Results

The test results for the movement tests are not numerical - the main objective is to ensure the functionality of the pumps and valves in combination with the motor drivers. When the code for the control of the pumps and valves was run, the lungs and tail inflated and deflated successfully. Initially, we wanted to use just one valve for the breathing system. However, upon testing it became apparent that deflation by means of difference in air pressure was too slow. Hence, the decision was made to include an extra valve, which would enable one air pump to both inflate and deflate the lungs. For optimal inflation and deflation, the pumps needed 8 seconds for both inflating and deflating, and the tail needs 3.5 and 3 seconds, respectively.

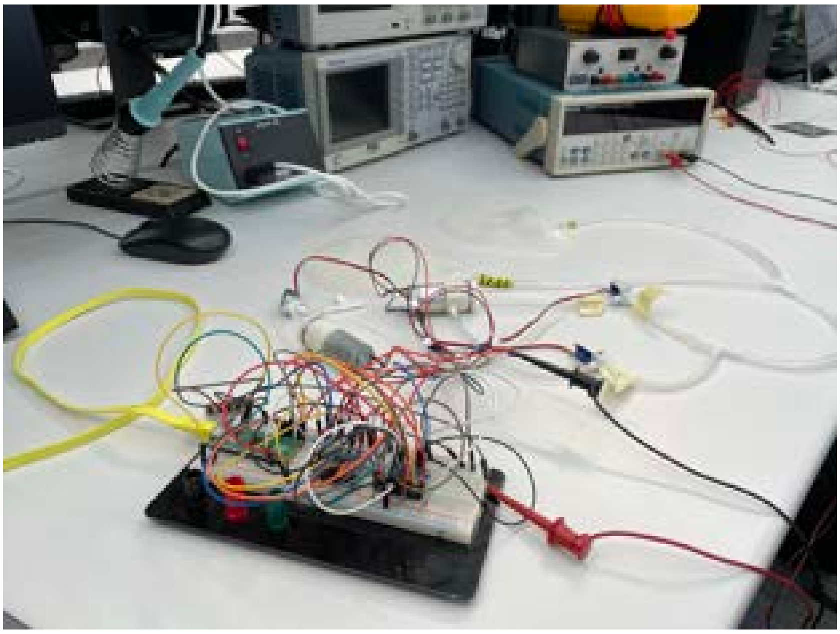


Figure F.1: Movement test setup.

F.1.3. Conclusion

Both the lungs and the tail are able to in- and deflate, performing the movement they are meant to do. Because of this, Requirement NF.05 and F.09 are met. Figure 5.3 shows a simplified overview of the movement cycle, which in total lasts for 66 seconds.

F.2. Touch Sensor Performance Test

Table F.2: Summary capacitive touch sensor performance test.

Test specialist:	Shea Haggerty
Date of completion:	09/06/2021
Test objective:	Determine the statistical accuracy of the capacitive touch sensor over a varying applied load.
Results:	The touch sensor has an average accuracy of 76%.

F.2.1. The Test Setup and Procedure

The touch sensor consists of a break-out board containing a capacitive touch sensor IC. The break-out board has 4 pins: Vdd, OUT, GND and LED. In order to test its functionality with the Pico, the Vdd was connected to the 3V output of the Pico, the GND was connected to the ground of the Pico and the OUT was connected to a general in-output pin. The test setup can be seen in Figure F.2. The break-out board also contains a soldering hole, to apply a custom touch pad. As we make use of the Velostat sheet as an indicator, we soldered a wire to the break-out board, and created a loop at the end of the wire: as can be seen in Figure F.3, this loop is necessary to make the connection between the conductive thread and the wire. A simple code was written to print ”The touch sensor has been touched!” whenever the OUT pin was high. As the touch sensor is designed for human touch, specifically stroking of the Teddy, the objective is to test for what applied pressure (load in grams) the touch sensor registers touch. Hence, we test the touch sensor accuracy with a sample size of n=25 ’touches’, using a load range of 5-2000 grams.

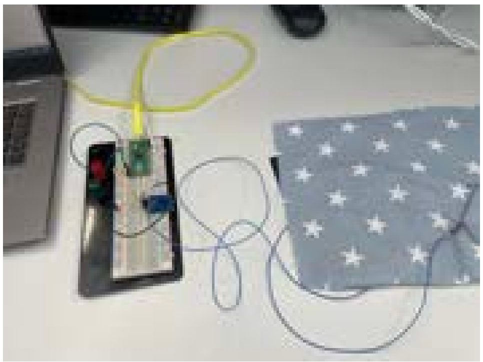


Figure F.2: Test setup of the touch sensor test.

Figure F.3: Close up of the way the wire and conductive thread are integrated to make contact with the Velostat.

F.2.2. Test Results

Table F.3 shows the test results of the test procedure described above. As can be seen, the accuracy of the touch sensor is 76%.

Table F.3: Results of the accuracy test of the touch sensor with variable loads.

Load	Response out of n=25 samples	Accuracy
5 grams	17	68%
50 grams	12	48%
250 grams	13	52%
500 grams	22	88%
1000 grams	25	100%
2000 grams	25	100%
Average accuracy		**76%**

F.2.3. Conclusion

In conclusion, the touch sensor works well when combined with the custom touch pad, which in our case is wire, conductive thread, fabric and Velostat. As the average person will stroke the Teddy with an applied load of about 250-500 grams, the average accuracy is 70%, which is high enough for the touch sensor to meet Requirement F.00.

F.3. Full Movement Sub-System Performance Test

Table F.4: Summary full movement sub-system performance test.

Test specialist:	Shea Haggerty and Laura Croes
Date of completion:	09/06/2021
Test objective:	Test the functionality of the movement sub-system, which encompasses the air pumps, valves, tail and lungs, touch sensor, and the usage of interrupts in the code.
Results:	Once the touch sensor is touched, the reaction time between registered touch and actuator excitation is less than 0.5 seconds (meeting Requirement F.08). The interrupt works.

F.3.1. The Test Setup and Procedure

Figure F.4 shows the test setup to test the functionality of the full movement sub-system. The code, which contains the interrupt concerning the touch sensor, can be found in Appendix G. The on-board logic of the motor drivers and the touch sensor are connected to the 3.3V output of the Pico, and the VM pins of the motor drivers are powered by a 5V power supply. All are connected to a common ground. The main code only contains the definition of a general in/output pin of the Pico as an interrupt pin (triggered by a rising clock edge), and the interrupt handler. The interrupt handler contains the command to execute the movement control code on the second core of the Pico.

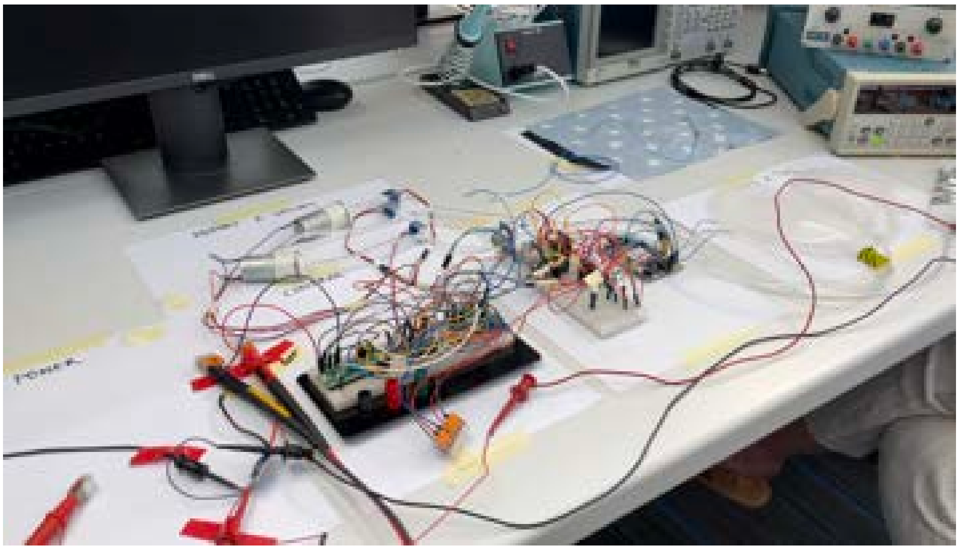


Figure F.4: Full movement sub-system test setup.

F.3.2. Test Results

The interrupt and the multi-threaded execution work as expected. When the touch sensor is touched while the movement is in action, a written error appears. However, this does not affect the functionality in any way. The response time in between a registered touch and the actuator excitation is less than 0.5 seconds.

F.3.3. Conclusion

As the interrupt, multi-threaded execution and the combination of the motor drivers and the touch sensor work, the conclusion is that the full movement sub-system functions correctly. With that, Requirement NF.08, NF.09, F.04 and F.05 are met.

F.4. Wireless Communication Performance Test

Table F.5: Summary wireless communication performance test.

Test specialist:	Shea Haggerty and Laura Croes
Date of completion:	10/06/2021
Test objective:	Determine the statistical accuracy of messages over a varying distance.
Results:	The LoRa communication works effectively between a range of 0 to 500m. Above that, the received messages are highly distorted and the reliability decreases by a lot. We were able to transmit and receive messages when the receiver was outside, and when the receiver was inside (ground floor and third floor) while the transmitter was outside. When sending a message three times, the odds of correctly receiving a message is 99.7% when assuming a 100 m distance from a third floor apartment to the outside. These odds are calculated using the cumulative binomial probability.

F.4.1. The Test Setup and Procedure

For the test two Raspberry Pi Picos were used. Both Picos were communicating with the RFM95 Lora Module which then drove a rubber 868MHz whip antenna. A drawing of the setup is visible in Figure F.5 and a picture of the setup at $1m$ is visible in Figure F.6, The code for the pico's was being run in the laptop they were connected to and the data was printed in the shell of Thonny, the IDE which was used. Table Table F.6 contains all the important test parameters.

Table F.6: Important parameters wireless communication performance test.

Transmitted message:	'Message:X. I am Shea and I am 21 years old, I was born at 14:00 PM on the 7th of August.'
Number of transmitted messages:	25
Send interval:	$3s$
Power value:	20
Test distances:	$1m$, $5m$, $10m$, $50m$, $100m$, $500m$, $750m$, $1000m$
Test scenarios:	Inside steel constructed building to outside, open air, and third floor apartment to outside.

The test procedure was as followed:

1. Measure distance between whip antennas with measuring tape for distances up until $10m$ and with 'footh path' app for larger distances.
2. Run program on Pico using the computer.
3. Verify that messages are arriving and the measured parameters are printed in the shell.
4. Wait for all 25 messages to have been sent.
5. Copy all the contents from the shell and place in excell file for later analysis.

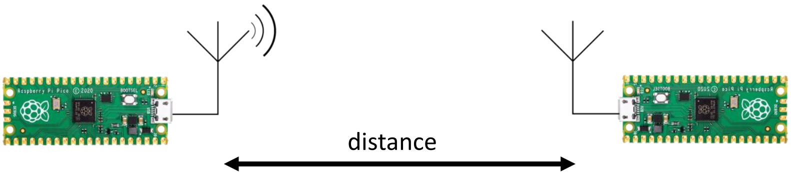


Figure F.5: Drawing of measurement set up for wireless communication performance testing.

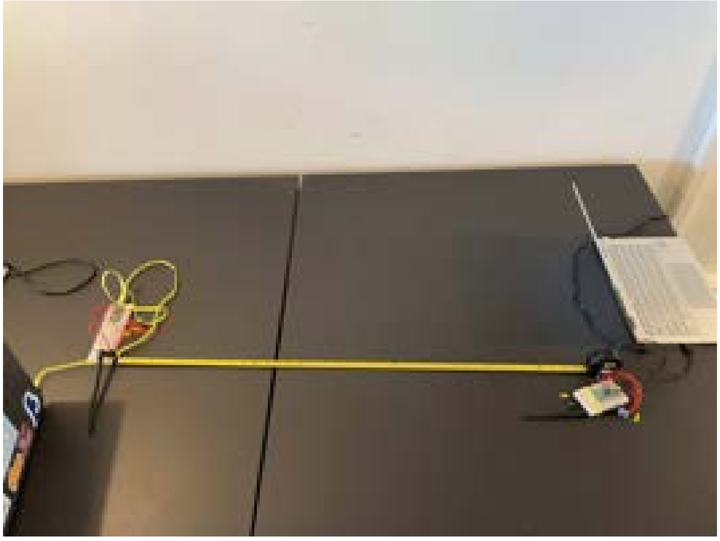


Figure F.6: Picture of $1m$ distance wireless communication performance testing.

F.4.2. Test Results

Table F.7: Results of distance wireless communication performance test with variable distance.

Scenario	Distance (m)	Received out of n=25 samples	Correctly received out of n=25 samples	Median signal strength (dB)	Standard deviation of transmission time (μs)
Inside steel constructed building to outdoor	1	100%	88%	-31.13	473
	5	100%	88%	-63.14	473
	10	100%	84%	-75.40	440
	50	100%	76%	-107.93	717
	100	100%	56%	-112.20	2832
Open air	100	100%	84%	-110.07	565
	500	72%	44%	-114.87	7623
	750	0%	0%	-	-
	1000	0%	0%	-	-
From third floor apartment to outside with the window closed	100	100%	88%	-83.4	1600
	250	84%	68%	-114.33	2134
	500	0%	0%	-	-
From third floor apartment to outside with the window closed	50	100%	80%	-100.47	724
	100	100%	84%	-102.07	724

F.4.3. Conclusion

The LoRa communication works effectively between a range of 0 to 500 meters. Above that, the received messages are highly distorted and the reliability decreases by a lot. We were able to transmit and receive messages when the receiver was outside, as well as inside (ground floor and third floor) while the transmitter was outside. For our use case, a range of $100m$ is more than sufficient, as we expect the Teddy to be in the seniors' house most of the time[1]. By sending the same message three times, the odds of correctly receiving a message is down to 99.7% when assuming a 100 m distance from a third floor apartment to the outside. These odds are calculated using the cumulative binomial probability.

F.5. Function Execution Time Performance Test

Table F.8: Summary function execution time performance test.

Test specialist:	Laura Croes and Shea Haggerty
Date of completion:	18/06/2021
Test objective:	Determine execution time of function used in the code which controls the Teddy.
Results:	The code for breathing and tailwagging takes $66s$, sending messages via LoRa $12s$, the touch sensor interrupt $3ms$, and the SoC code $2ms$.

F.5.1. The Test Setup and Procedure

For this test, all the functions which are used to control the Human Interaction & Integration sub-system are individually tested to determine their run time. The is done by printing time stamps, using the utime MicroPython library, before and after execution of the functions. The functions are run 5 times in a row, after which the average time is noted. The results are given in Table F.9.

F.5.2. Test Results

Table F.9: Results of execution time performance test with each function having been run 25 times.

Function	**Average time n=5 samples**
Touch interrupt handler	$3ms$
Breathing & tail wagging	$66002ms$
Determining SoC	$2ms$
LoRa transmission	$11999ms$

F.5.3. Conclusion

The longest function takes $66002ms$, which is about $66s$. However, this function will be executed on the second core, as to not interrupt data readings or other functionalities of the Teddy. The interrupt, which will disturb the sensor readings as it executes on the first core, takes $3ms$, and analog data will not be measured for this time. As the touch sensor triggered interrupt takes $3ms$, we consider this interrupt as negligible as there are no time-critical functions which monitor the safety of the user. The longest function which executes on the first core is sending messages using LoRa, which takes almost $12s$. The function sends three consecutive messages, to enlarge the chance of the receiver receiving

---

[1] When the senior is outside of this range the GPS will be used, sending messages over text. More details on this can be found on the report of the Sensor and Data acquisition subsystem.

a correct message.

This data can be used in the future to better schedule function between both cores.

F.6. Sound Performance Test

Table F.10: Summary sound performance tests.

Test specialist:	Laura Croes and Shea Haggerty
Date of completion:	15/06/2021
Test objective:	Measure the sound intensity of system parts which produce noise at a varying distances, and measure their power consumption if relevant.
Results:	The accuracy of the measurements is not very high as there were frequent variation of the ambient sound during the tests. The peak value was taken during measurements with an estimated accuracy of ± 2 dB due the noise. A noticeable difference in sound intensity is therefore observed up to 1 m distance. As the sound is meant to be heard when the user is interacting with the Teddy at 1 m distance is sufficient for the audio intensity to be considered sufficient. Furthermore, the added sound of the pumps does not surpass that off the speaker at any distance, making the audio produced the dominant sound.

F.6.1. The Test Setup and Procedure

In practice, the only element in the prototype which is meant to produce sound is the playback board (containing an amplifier) in combination with the speaker driver. During the test, the audio module was enabled at maximum load, and the sound intensity was recorded for the same audio segment at 0.5 m, 1 m and 5 m. On top of that, the prototype contains two air pumps which produce a non-negligible amount of noise. As this noise is unwanted, we want to measure their sound intensity as well, to verify if the noise of the pumps does not overshadow the produced audio.The sound intensity of the pumps was measured for the same ranges mentioned above. The exact test procedure was as follows:

1. The audio system in question was placed on a table, and connected to a 5V power supply.
2. As there is always background noise present, the ambient sound intensity was measured at all distances for which the sound intensity of the relevant audio system was measured as well.
3. The system was switched on and the sound intensity at the pre-defined ranges were measured using decibel meter app on an Iphone.
4. Measurements were also done with the noise producing components placed inside the Teddy, as this is the place they will be in the final prototype.
5. The above steps were repeated for all specified ranges.

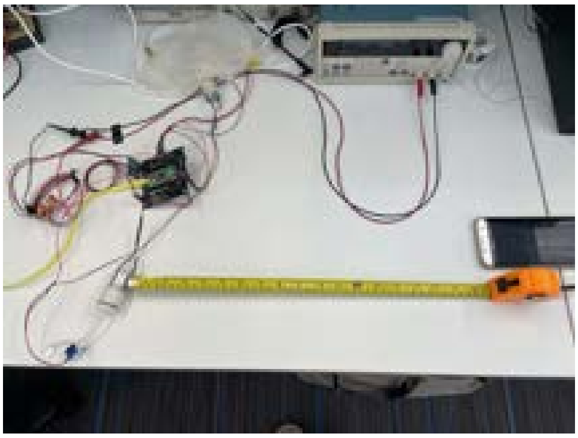


Figure F.8: The test setup for measuring the sound intensity of the pumps, with the pumps being on the table.

Figure F.9: The test setup for measuring the sound intensity of the pumps, with the pumps being inside the two pouches.

[H]

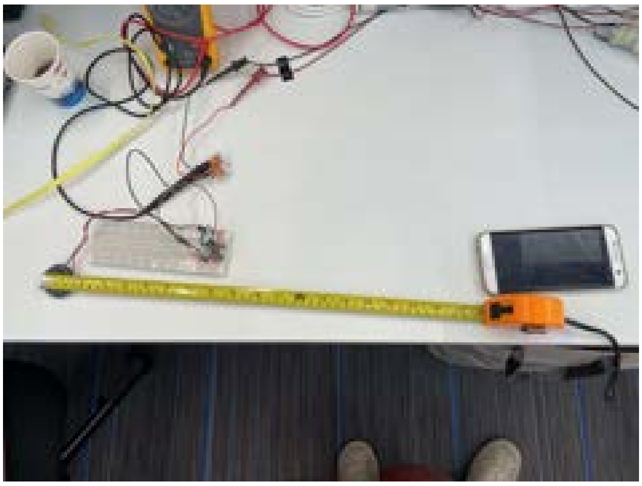


Figure F.7: The test setup for testing the playback board with the speaker driver at 0.5m.

F.6.2. Test Results

Table F.11 and F.12 show the results of the measurements described in the previous section.

Table F.11: Results of sound intensity measurements of the audio at various distances.

Audio	Outside Teddy		Inside Teddy	
	Ambient sound intensity [dB]	**Sound intensity outside Teddy [dB]**	**Ambient sound intensity [dB]**	**Sound intensity inside Teddy [dB]**
$0m$	47	90	48	70
$0.5m$	51	60	48	53
$1m$	48	55	47	51
$5m$	49	52	48	70

Table F.12: Results of sound intensity measurements of the pumps at various distances, with an accuracy of approximately ± 2 dB.

Pumps	**Outside Teddy**		**Inside Teddy**	
	Ambient sound intensity [dB]	**Sound intensity outside Teddy [dB]**	**Ambient sound intensity [dB]**	**Sound intensity inside Teddy [dB]**
$0m$	50	74	52	69
$0.5m$	50	55	49	53
$1m$	50	53	48	49
$5m$	51	52	48	48

F.6.3. Conclusion

The accuracy of the measurements is not very high as there were frequent variation of the ambient sound during the tests. This lead to the sound intensity measured to vary during the measurement. The peak value was taken and noted in Tables F.11. An estimated accuracy of ± 2 dB is assumed due this noise. A noticeable difference in sound intensity is therefore observed up to 1 m distance. After this the sound made by the Teddy blends in with the ambient noise.

As for the pumps, these contribute significantly to the total noise as well when placed outside of the Teddy. However, the pumps will be placed inside the Teddy inside the inner pouch. This reduces the sound intensity down to where the difference between the pumps and the ambient sound is not measurable - considering the accuracy of the measurements.

As the sound is meant to be heard when the user is interacting with the Teddy a 1 m distance is sufficient for the audio intensity to be considered sufficient. Furthermore, the added sound of the pumps does not surpass that off the speaker at any distance, making the audio produced the dominant sound.

G

Code

G.1. Movement

Airflow control

```python
# This file contains two functions: one for breathing and one for tail wagging.
# All movement is enabled using pneumatics (soft robotics in our case).
# An air pump, valves and tubing are used to inflate the lungs of the Teddy and wag its tail.
# When the touch sensor detects pressure, the Teddy will start breathing and wagging its tail slowly.

from machine import Pin, PWM
from Motor_Driver import Pumps, Valves
import utime
import _thread

# Assigning the pumps and valves to certain pins on the Raspberry Pi Pico
BIN2_valvetail = 18
BIN1_valvetail = 19
AIN1_valvelung1 = 17
AIN2_valvelung2 = 16
PWMA_pumps = 21
PWMB_pumps = 20

pump = Pumps(PWMA_pumps,PWMB_pumps)
valve = Valves(BIN1_valvetail,AIN1_valvelung1,AIN2_valvelung2)


def breathing_tailwagging():
    frequency = 2000    # Telling the Raspberry Pi Pico how often per second to switch power on and off of the pumps.

    valve.valve_tail_on()    # Tail curls up for 3.5 seconds
    pump.pump_tail_on()      # Breathe in
    utime.sleep(3.5)

    valve.valve_tail_off()   # Tail moves back to normal position
    pump.pump_tail_off()
    utime.sleep(2)

    valve.valve_tail_on()    # Tail curls up for 3.5 seconds
    pump.pump_tail_on()
    utime.sleep(3.5)

    valve.valve_tail_off()   # Tail moves back to normal position
    pump.pump_tail_off()
    utime.sleep(2)

    valve.valves_lungs_in() # Breathe out
    pump.pump_lungs_on()
    utime.sleep(9)
```

```
    valve.valves_lungs_out()
    utime.sleep(9)

    valve.valves_lungs_in() # Breathe out
    utime.sleep(9)

    valve.valves_lungs_out()
    utime.sleep(9)

    valve.valves_lungs_in() # Breathe out
    utime.sleep(9)

    valve.valves_lungs_out()
    utime.sleep(10)

    touch = False
    valve.all_valves_off()
    pump.all_pumps_off()
```

Motor Driver Library

```
from machine import Pin,PWM
from time import sleep

class Valves():
    def __init__(self, BIN1_valvetail,AIN1_valvelung1,AIN2_valvelung2):
        self.bin1 = Pin(BIN1_valvetail, mode=Pin.OUT, pull=None)
        self.ain2 = Pin(AIN2_valvelung2, mode=Pin.OUT, pull=None)
        self.ain1 = Pin(AIN1_valvelung1, mode=Pin.OUT, pull=None)


    def valves_lungs_in(self):    # Inputs for the valves of the lungs when breathing in (inflating)
            self.ain1.value(1)
            self.ain2.value(0)


    def valves_lungs_out(self):   # Inputs for the valves of the lungs when breathing out (deflating)
            self.ain1.value(0)
            self.ain2.value(1)


    def valve_tail_on(self):
            self.bin1.value(1)


    def valve_tail_off(self):
            self.bin1.value(0)


    def all_valves_off(self):
        self.ain1.value(0)
        self.ain2.value(0)
        self.bin1.value(0)


class Pumps():
    #def __init__(self, BIN2_pump2,BIN1_pump2,AIN1_pump1,AIN2_pump1):
    def __init__(self,PWMA_pumps,PWMB_pumps):
        #self.bin2 = Pin(BIN2_pump2, mode=Pin.OUT, pull=None) # Pin 26 GP 20
        #self.bin1 = Pin(BIN1_pump2, mode=Pin.OUT, pull=None) # Pin 25 GP 19
        #self.ain2 = Pin(AIN2_pump1, mode=Pin.OUT, pull=None) # Pin 24 GP 18
        #self.ain1 = Pin(AIN1_pump1, mode=Pin.OUT, pull=None) # Pin 22 PG 17
        self.apwm = PWM(Pin(PWMA_pumps))    # Pin 21, gp16 The 50 is the frequency. Can adjust in main?
        self.bpwm = PWM(Pin(PWMB_pumps))    # Pin 27, GP 21
```

```
    def pump_lungs_on(self):
       self.apwm.duty_u16(65530)


    def pump_tail_on(self):
        self.bpwm.duty_u16(65530)

    def pump_lungs_off(self):
        self.apwm.duty_u16(0)

    def pump_tail_off(self):
        self.bpwm.duty_u16(0)

    def all_pumps_off(self):
        self.apwm.duty_u16(0)
        self.bpwm.duty_u16(0)
```

Test Code: Full Movement Sub-System Integration

This code contains the touch sensor interrupt, the interrupt handler, and the code for reading the SoC. As the main code on the Pico will consists mostly of code by the Sensors & Data Acquisition sub-domain, this code will be given in their report. The functions in this code segment are added to their main code.

```
from machine import Pin
import Motor_Driver
import _thread
import utime
from Motor_Driver import Pumps, Valves

touch_sensor = Pin(11, Pin.IN, Pin.PULL_UP)

analog_value = machine.ADC(27)
voltage_conversion =  0.0000757203 #3.3 / 2^16 /(100/150))
calibration = 1.028;

def SoC():
    reading = analog_value.read_u16()
    soc = float(reading) * voltage_conversion #* calibration #+ offset
    #print("ADC: ",soc)
    #print(voltage_conversion)
    if soc <= 3:
        battery_low = 1
    else:
        battery_low = 0


def interrupt_touch(self):
    touch = True
    _thread.start_new_thread(breathing_tailwagging, ())


touch_sensor.irq(trigger=Pin.IRQ_RISING, handler=interrupt_touch)
```

G.2. Wireless Communication

LoRa: TX Code

```
from time import sleep
from ulora import LoRa, ModemConfig, SPIConfig

# Lora Parameters
RFM95_RST = 27
RFM95_SPIBUS = SPIConfig.rp3_0
RFM95_CS = 1
RFM95_INT = 15
RF95_FREQ = 868.0
RF95_POW = 20
CLIENT_ADDRESS = 1
SERVER_ADDRESS = 2

# initialise radio
```

```
lora = LoRa(RFM95_SPIBUS, RFM95_INT, CLIENT_ADDRESS, RFM95_CS,
            reset_pin=RFM95_RST, freq=RF95_FREQ, tx_power=RF95_POW, acks=True)


def sending_lora():
    i = 1
    while i <= 3:
        lora.send_to_wait("Message:" + str(i) + ". This is a full load test, and now we are testing LoRa.", SERVER_ADDRESS)
        i = i + 1
```

LoRa: RX Code

Note that this is code in Python, as this code is implemented on the Raspberry Pi in the Base Station.

```
from time import sleep
from ulora import LoRa, ModemConfig, SPIConfig
from machine import Pin
import utime

# This is our callback function that runs when a message is received
def on_recv(payload):
#     print("From:", payload.header_from)
#     print("Received:", payload.message)
#     print("RSSI: {}; SNR: {}".format(payload.rssi, payload.snr))
    print(str(payload.header_from) + "\t" + str(payload.message) + "\t" + str(payload.rssi) + "\t" + str(payload.snr) + "\t" + str(utime.ticks_ms()))
# Lora Parameters
RFM95_RST = 27
RFM95_SPIBUS = SPIConfig.rp2_0
RFM95_CS = 5
RFM95_INT = 15
RF95_FREQ = 868.0
RF95_POW = 20
CLIENT_ADDRESS = 1
SERVER_ADDRESS = 2

led = Pin(25, Pin.OUT)
led.on()

# initialise radio
lora = LoRa(RFM95_SPIBUS, RFM95_INT, SERVER_ADDRESS, RFM95_CS,
             reset_pin=RFM95_RST, freq=RF95_FREQ, tx_power=RF95_POW, acks=True)

#lora = LoRa(RFM95_SPIBUS, RFM95_INT, SERVER_ADDRESS, RFM95_CS,
#             freq=RF95_FREQ, tx_power=RF95_POW, acks=True)


# set callback
lora.on_recv = on_recv

# set to listen continuously
lora.set_mode_rx()


# loop and wait for data
while True:
    sleep(0.1)
```