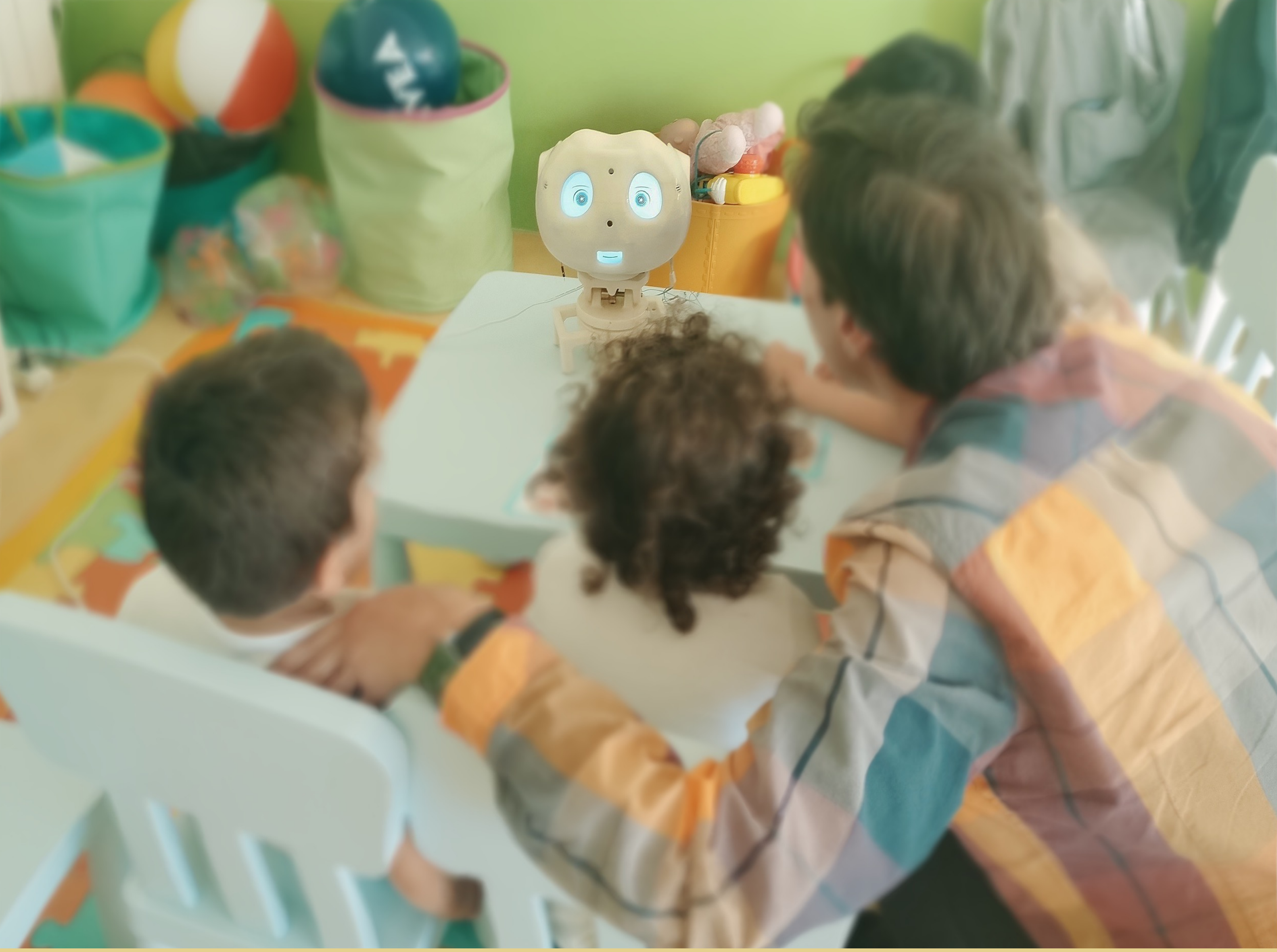


Ricardo Manuel Teixeira Pereira

ISR-RobotHead v2.0: New Hardware and Software Modules

Dissertation submitted for the degree of Master of Science in Electrical and Computer Engineering
September, 2018

· U C ·

Universidade de Coimbra

FCTUC

University of Coimbra
Faculty of Sciences and Technology
Department of Electrical and Computer Engineering

ISR-RobotHead v2.0: New Hardware and Software Modules

Ricardo Manuel Teixeira Pereira

Dissertation submitted to the Electrical and Computer Engineering Department of the Faculty of Science and Technology of the University of Coimbra in partial fulfilment of the requirements for the Degree of Master of Science.

Supervisor: Prof. Doctor Urbano José Carreira Nunes
Co-Supervisor: Prof. Doctor Luís Conde Bento

Jury:
Prof. Doctor Urbano José Carreira Nunes
Prof. Doctor Hélder de Jesus Araújo
Prof. Doctor Jorge Manuel Moreira de Campos Pereira Batista

Coimbra, September of 2018

Agradecimentos

Começo por agradecer ao meu orientador Professor Doutor Urbano Nunes e co-orientador Professor Doutor Luís Conde, por me terem dado a oportunidade de desenvolver esta dissertação bem como todas as orientações que levaram ao sucesso deste trabalho.

Um grande agradecimento a todos os colegas de laboratório que me acolheram e ajudaram no decorrer desta etapa final. Uma menção especial ao Luís Garrote por toda a sua disponibilidade, ajuda e conselhos dados no decorrer desta dissertação.

Agradeço a colaboração do Centro de Ação Social do Concelho de Ílhavo (CASCI), na pessoa da Professora Rosa Roldão, na realização de testes experimentais com as crianças da instituição. Um agradecimento ao Doutor Carlos Carona pela sua ajuda na avaliação das expressões faciais e pela disponibilidade que teve em me acompanhar no realização do teste experimental.

Um agradecimento ao ISR pelas excelentes condições e recursos que permitiram o cumprimento desta importante etapa. Este trabalho foi suportado pelo UID/EEA/00048/2013 com financiamento do FEDER, programas QREN e COMPETE.

A todos os meus colegas e amigos que ajudaram a ultrapassar todas as barreiras e dificuldades que foram surgindo no decorrer destes 5 anos, um muito obrigado. Um especial agradecimento ao Ivo Frazão e ao Miguel Antunes por todos os momentos, quer de trabalho quer de diversão que foram vividos nestes últimos anos. Um agradecimento carinhoso para a Mariana Roque pelo seu enorme apoio, motivação e todos os momentos partilhados que contribuíram para a conclusão deste percurso.

Por último, mas não menos importante, um grande e profundo agradecimento à minha família por todo o apoio e motivação prestada ao longo deste percurso académico, especialmente aos meus pais que me proporcionaram todas as condições para a realização deste percurso, mas também à minha irmã e cunhado Bruno Silva, por toda a disponibilidade e ajuda ao longo dos 5 anos.

A todos um muito obrigado,
Ricardo Pereira

Abstract

The technology has gained great relevance in human life and the robotic field, through the development of robots capable of interacting with human, has contributed much to this increase. A large part of these robots are considered social or social-interactive robots. They started by making non-verbal communication and currently have artificial intelligence modules that allow them to have innumerable interactions with humans. These robots have attracted the attention of psychologists because of their potential to assess human behavior as well as the contribution they can make in the area of rehabilitation.

The main goal of this Master's dissertation is to improve the interactions capabilities of the ISR-RobotHead prototype, a robotic head developed at the Institute of Systems and Robotics (ISR) with two LCDs to display the eyes and one to the mouth. The facial expressions have been improved and speech recognition modules have been added, thus allowing humans to interact with the robot through voice commands and in return displaying a facial expression. As the main controller of the expressions to be displayed by the robot, a micro-computer Raspberry Pi was used and to include the modules of speech recognition a Matrix Voice board attached to another Raspberry Pi was used. Through the new software architecture, both Raspberry Pis are able to communicate with each other. This architecture also allows an easy integration with others external systems developed later that may enhance the robot interaction capabilities.

In order to evaluate the non-verbal communication of this robot head prototype, experiments were carried out where children were asked to recognize the emotions presented. Regarding speech recognition, a comparison between all the software systems was made to determine which is the best and most appropriate for interpreting voice commands. With the results shown, it became clear that the emotional expressions were well recognized and the integration of the speech recognition systems becomes a good way to interact with the ISR-RobotHead, thus achieving the desired objectives.

Keywords: Human-Robot Interaction, LCD-based Robotic Head, Facial Expressions, Speech Recognition Systems, ISR-RobotHead

Resumo

A tecnologia tem vindo a ganhar grande importância na vida humana e a área da robótica, através do desenvolvimento de robôs capazes de interagir com o homem, contribui muito para esse aumento. Uma grande parte destes robôs são considerados robôs sociais ou sócio-interativos. Começaram por conseguir fazer uma comunicação não verbal e atualmente têm módulos de inteligência artificial que lhes permite ter inúmeras interações com o humano. Estes robôs têm despertado a atenção de psicólogos devido ao seu potencial para avaliar o comportamento humano bem como o contributo que podem dar na área da reabilitação.

Esta dissertação de Mestrado tem como objetivo principal melhorar as capacidades de interação com o homem do protótipo ISR-RobotHead, uma cabeça robótica desenvolvida no Instituto de Sistemas e Robótica (ISR) com dois LCDs para expressar os olhos e um para a boca. Foram melhoradas as expressões faciais e adicionados módulos de reconhecimento da fala, permitindo assim que o humano interaja com o robô através da voz e este responda com uma expressão facial. Como principal controlador das expressões a apresentar pelo robô foi utilizado um micro-computador Raspberry-Pi e, de modo a incluir os módulos do reconhecimento de fala foi utilizada a placa Matrix Voice acoplada a um outro Raspberry Pi. Através da nova arquitetura de software desenvolvida, ambos os Raspberry Pis são capazes de comunicar entre si. Esta arquitetura também permite uma fácil integração a qualquer outro sistema externo desenvolvido posteriormente que contribua para ajustar a melhor expressão a ser exibida.

De modo a validar a comunicação não verbal deste protótipo, foram efetuadas experiências com crianças para reconhecerem as emoções apresentadas. Relativamente ao reconhecimento da fala, foi efetuada uma comparação entre todos os serviços utilizados e determinado qual é o melhor e o mais adequado para interpretar comandos de emoções. Com os resultados apresentados, verifica-se que as expressões emocionais foram bem reconhecidas e a integração dos sistemas de reconhecimento da fala tornou-se uma boa maneira de interagir com o robô, alcançando assim os objetivos propostos.

Palavras Chave: Interação Homem-Robô, Cabeça Robótica baseada em LCDs, Expressões Faciais, Sistemas de Reconhecimento da Fala, ISR-RobotHead

"A dream becomes a goal when action is taken toward its achievement."
- Bo Bennett

List of acronyms

ASD	Autism Spectrum Disorder.
ASR	Automatic Speech Recognition.
CRI	Children-Robot Interaction.
CPU	Central Processing Unit.
DAC	Digital-to-Analogue Converter.
FPGA	Field Programmable Gate Array.
GPIO	General Purpose Input/Output.
HMI	Human-Machine Interface.
HRI	Human-Robot Interaction.
IDE	Integrated Development Environment.
IP	Internet Protocol.
LCD	Liquid Crystal Display.
LED	Light-Emitting Diode.
PWM	Pulse-Width Modulation.
RAM	Random Access Memory.
RGB	Red, Green, Blue.
RH	Robot Head.
RPi	Raspberry-Pi.

RPi3	Raspberry-Pi 3 Model B.
RPiCM3	Raspberry-Pi Compute Module 3.
SPI	Serial Peripheral Interface.
SSH	Secure Shell.
TCP	Transmission Control Protocol.
UART	Universal Asynchronous Receiver-Transmitter.
WER	Word Error Rate.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

For several years, the robots were considered best suitable to work on production lines and heavy duty jobs. Many of today's robots are able to interact with humans, hence they are increasingly applied in household chores and in human social life. The interactions that each robot can have depends on its purpose. The goals of social robots are to entertain people, play with and help them. People feel curious about what is like to interact with a robot and what are their capabilities. These become important factors to consider in the design and development of social robots [9].

1.1 Motivation and Context

We are in a digital and robotic era. Intelligence is today what distinguishes humans from robots. Despite this, there are already robots capable of making their own decisions which makes interaction with humans more interactive. This field of research, whose aim is to make robots intelligent, is undoubtedly a great challenge.

When humans communicate/dialogue with someone else, their faces tend to express, even if they don't realize it, their current emotions. Will robots be capable of the same? The answer began to emerge in the late 90's, when the robotic head called "Kismet" [8] appeared to express some facial expressions. After this, several robotic heads began to emerge capable of expressing facial emotions through a set of mechanical actuators to move their eyes and mouth. For the same purpose, there are robots that use LCDs (Liquid Crystal Displays) to display the eyes and mouth, such as, Baxter [49], Buddy robot [50] and ISR-RobotHead [31, 32].

Another way for the HRI (Human-Robot Interaction) is through the speech, where the robots are equipped with Speech Recognition Systems allowing their actions to be controlled by voice, such as Kismet [8] and Buddy [50].

This research aims to improve and implement forms of Human-Robot Interaction in the new version of the ISR-RobotHead. This robotic head was developed in ISR (Institute of Systems and Robotics) thanks to the previous work [31, 32] of several researchers. At the beginning of this work, the ISR-RobotHead was already able to express some facial emotions but it was necessary to improve them. The software architecture implemented does not allow a easy change of the emotional expression displayed and does not also allow the integration of the new modules, so in order to add newer technologies currently available the software architecture must be changed. To add new modules it is also necessary to expand and improve the hardware architecture. By improving these components, the new version of ISR-RobotHead should become more appealing to human eyes and to became a modular open platform for researchers.

1.2 Objectives

The main purpose of this work is to rebuild the software and hardware architecture previously developed and extend the features of ISR-RobotHead, by adding two different ways to interact with it. Since it is intended to interact with humans, more specifically, with children, it is of utmost importance to have a solid software architecture and to easily integrate new commands/modules from an external system. To that end, the following main goals were defined:

1. Change the software and hardware architecture in order to be easier to:
   (a) improve and add new facial expressions;
   (b) control the robot, or the facial expression with other system;
2. Improve the facial expressions;
3. Integrate a speech recognition system;
4. Develop a mobile application in order to control the expression to be shown and some parameters associated with expressions.

1.3 Implementation and Key Contributions

Figure 1.1 shows the new architecture implemented for the ISR-RobotHead along with integration of the new added systems, mobile application and the speech recognition system. The following main implementations and contributions are described in this dissertation:

ISR-RobotHead Facial Expressions Modules - Software and Hardware Overview (Chapter 4)

- Description of ISR-RobotHead's new software and hardware architectures to display facial expressions.
- Description of each software module and hardware component used to improve and display facial expressions.

ISR-RobotHead Sound System Modules - Software and Hardware Overview (Chapter 5)

- Description of hardware architecture and the new hardware components integrated in the ISR-RobotHead.
- Speech recognition system module.
- Sound output system enabling the robot to emit sounds.

ISR-RobotHead Modules Integration (Chapter 6)

- Integration of all developed modules, allowing the ISR-RobotHead to function as an integrated system.
- Development of a Mobile application to control which emotional expression to display and its characteristics.
- Description of the software architecture integrating all developed modules.

Results (Chapter 7)

In order to evaluate the developed work, were performed the following assessments:

- Recognition of the new emotional states exhibited by ISR-RobotHead.
- Evaluation of the speech recognition systems though Word Error Ratio calculating and specific words recognition (expression commands).

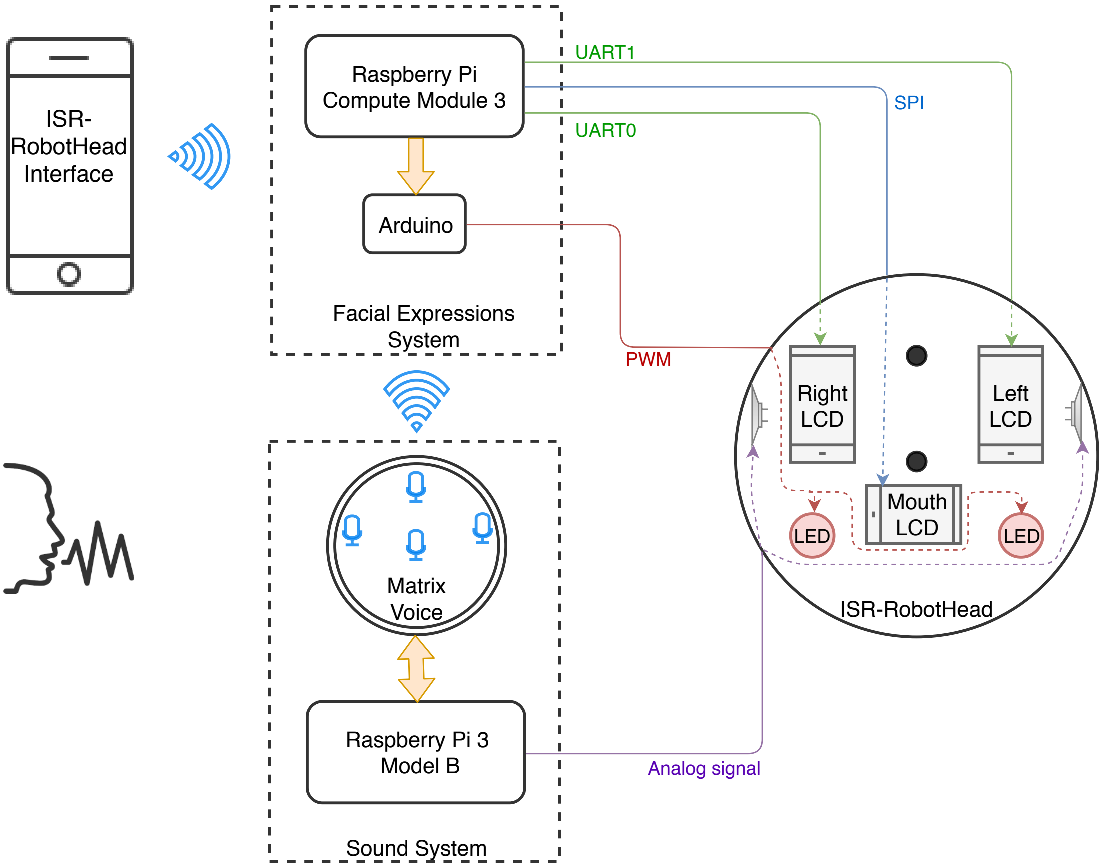


Fig. 1.1 ISR-RobotHead new hardware architecture.

Chapter 2

Interactive Robotic Heads: State of the Art

2.1 Social Robots

In our modern society, social robots will become very useful as educational tools, therapeutic caregivers and rehabilitation [13].

Interaction between robots and humans is strongly influenced by the communications capabilities of the robots. To that end, in order to capture Human's attention, the robots might use non-verbal forms of communication to develop an intuitive human-machine interaction [45]. The physical appearance of robots should be carefully developed. The idea of social robots is to communicate, cooperate and interact with humans but they may not be very successful if they have an anthropomorphized appearance. *M. Mori* [36] argues that robots that are very similar to human appearance fall into the *uncanny valley* (a common unsettling feeling people experience humanoid robots closely resemble humans in many respects but are not quite convincingly realistic). *M. Mori* also concluded that children prefer robots with toy's appearance rather than the human's appearance. This idea is reinforced by *Seyama & Nagayama* [44]. They did a study of how an artificial head should be built and the results were according to what Mori predicted, humans preferred a head that is not similar to a human head.

2.1.1 Robotic Heads able to Express Human-like Emotions

In order to socially interact with a human, the robot has to convince him that he has desires and intentions. This idea was supported by Cynthia Breazeal *et al.* [8] and, in order to test it, they have built a robotic head called 'Kismet' (see Fig. 2.1). Kismet head has one camera in each eyeball to recognize human face and fifteen degrees of freedom to move its eyes, eyebrows, ears, lips and mouth. The robot is capable to express nine emotional expressions such as: happiness, sadness, surprise, boredom, anger, calm, displeasure, fear and interest. When Kismet is alone it stays sad, but when it detects a human face it smiles, trying to capture the human attention.

After Kismet, several mechanical heads capable to express facial emotions started to appear. Some examples are EDDIE [45] with the dragon appearance, Flobi [33] with a anthropomorphic appearance and the EMYS [24] inspired by cartoons and series *Teenage Mutant Ninja Turtles* (see Fig. 2.1). These four robotic heads are equipped with speech recognition system implemented, which allows the reception through voice commands/requests, the emotions which the users wants the robot to express. Among these, Flobi is the only one that uses LEDs on the cheeks to express some color along with the expression of emotion.

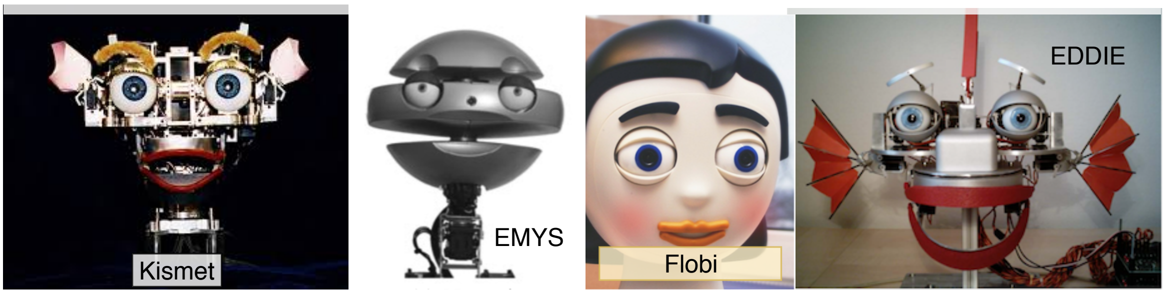
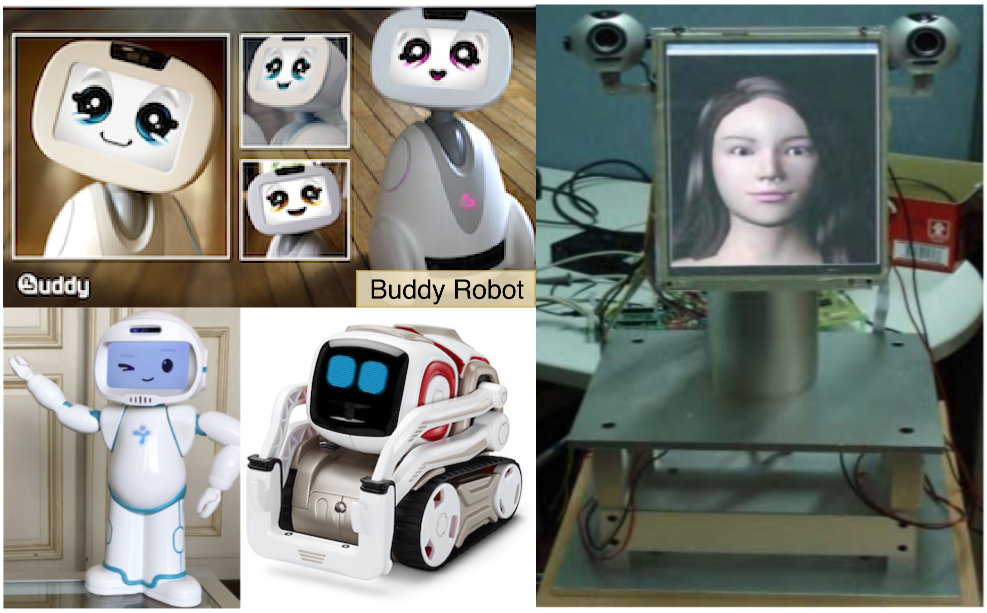


Fig. 2.1 Mechanical robotic heads capable to express facial emotions.

2.1.1.1 LCD-based Robotic Heads to Display Expressions

Another technical approach used in robots to convey emotions are LCD-based robotic heads. The idea is to use LCDs as a way to implement the eyes and even the mouth avoiding the use of mechanical mechanism with several degrees of freedom to be able to make all movements. The use of the LCDs allows a greater simplicity in the design of the head and enables greater freedom in the design of the eyes/mouth and their movements. An example of this type of robots is the Baxter robot [49] which, although not being a social robot, has a tablet-like screen, with cartoony eyes and eyebrows to display expressions along with intuitive messages to nearby users. Other examples of robotic heads based on LDCs are: QT robot [12], Cozmo [51], Buddy Robot [50] (see Fig. 2.2) and IROMEC [26]. Similar to Baxter, all have a tablet-like screen except Cozmo. With the exception of the Baxter, the main goal of all other robots is to interact with humans, especially children and help them in their development. The Buddy and QT robot are equipped with speech recognition systems that allow them to receive commands through the users' voice. Cozmo is a small commercial robot-toy developed by *Anki* with a small LCD to capture people attention in order to interact and play with them. Through the LCD, the robot is capable to express many expressions.

A robotic head with a different look was developed with the same goal, to interact with people through facial expressions. The name of the robot is PIL Head Robot [6] (see Fig. 2.2) and only has a huge LCD and two cameras. The LCD displays a virtual anthropomorphic head capable of expressing some facial expressions. To help the interaction, this robot is equipped with face recognition, speech recognition and gesture recognition systems.



QTrobot Cozmo PIL Head Robot

Fig. 2.2 LCD-based Human-Machine Interfaces (HMIs).

Unlike the other robots already mentioned, the ISR-RobotHead [32] uses three LCDs, one for display each eye and one to the mouth. The design of this head is located between the mechanical robots mentioned and the LCD-based HMIs, thus allowing a design closer to the anthropomorphic without mechanical actuators. Table 2.1 summarizes the features of the LCD-based HMIs.

	Baxter	QTrobot	Buddy Robot	Cozmo	ISR Robot Head v1.0
LCDs-used	One	One	One	One	Three
Degrees of Freedom	Only on arms	14 - Head and arms	-	-	-
Able to Play Games	No	Yes	Yes	Yes	No
Facial Emotion Expressions	Yes	Yes	Yes	Only eye expressions	Yes
Facial Recognition	Yes	Yes	Yes	Yes	No
Speech Recognition	Yes	Yes	Yes	Yes	No
Text-to-Speech System	-	Only emit pre-defined sounds	Yes	No	No

Table 2.1 Main features and interactive modes available on the LCD-based HMIs.

2.1.1.2 ISR-RobotHead

The ISR-RobotHead v1.0 prototype was developed aiming to be used in Children-Robot Interaction (CRI) studies. The prototype was equipped with a sound output system, two cameras for stereo vision and three LCDs to display facial expressions, two for the eyes and one for the mouth. To control all the components, a Raspberry Pi Compute Model 1 was used.

The first version of the ISR-RobotHead [31, 32] was centered on the development of human facial expressions and how to display them. Figure 2.3 shows an overview of the developed architecture. In order to display eye expressions, two uLCD-32PTU were used. These LCDs use the UART (Universal Asynchronous Receiver-Transmitter) communication protocol and were connected in parallel. In other words, both LCDs receive the same command every time. For mouth expressions, an LCD that uses the SPI (Serial Peripheral Interface) communication protocol was used, ITDB02-1.8SP. The images used to show the expressions are stored in different locations for each LCD: for the eyes, the images are stored in a micro-SD card for each LCD and for the mouth, the images are stored in binary file in Raspberry. Figure 2.4 shows the six expressions that the ISR-RobotHead v1.0 is able to do.

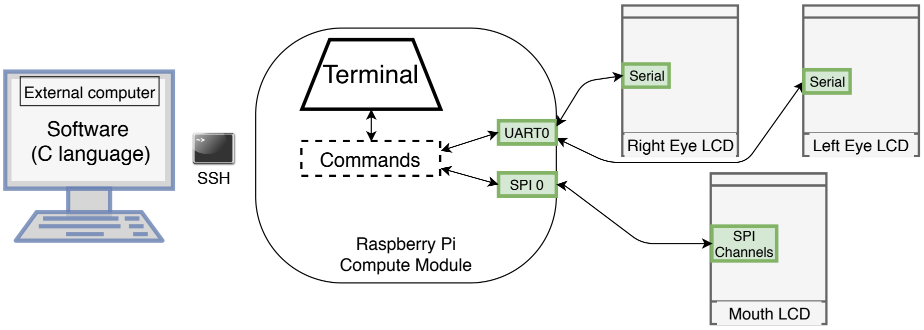
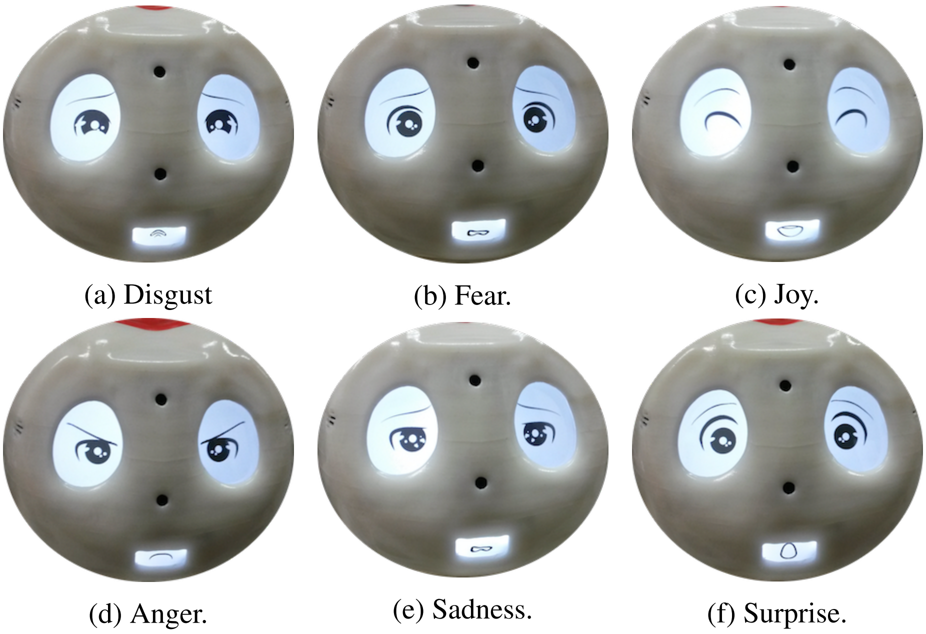


Fig. 2.3 Software/Hardware overview of the first version developed in the ISR-RobotHead.


(a) Disgust (b) Fear. (c) Joy.

(d) Anger. (e) Sadness. (f) Surprise.

Fig. 2.4 Pictures of the ISR-RobotHead v1.0 representing six emotional expressions.

In order to validate the emotional expressions displayed by the ISR-RobotHead v1.0 [31, 32], a preliminary test was performed with a sample of 9 children, aged between 6 and 10 years old, showing the results in Table 2.2. Some of children had some form of psychological disorder (n = 7), 3 had some degree of intellectual disability and 2 had a neurodevelopmental conditional. Children were presented to the robotic head displaying one of the six expressions and they were asked to identify the emotional state.

	Emotions					
	Fear	Sadness	Joy	Surprise	Anger	Disgust
Correct answers	6	9	9	6	9	1

Table 2.2 Correct detection of robotic emotional expressions [32].

Chapter 3

Background Knowledge

3.1 Robots - Psychology Field Point of View

When social robots, i.e. robots developed to interact with humans through verbal or non-verbal communication began to appear, the psychology field began to study the impact that robots had on humans. From a psychological point of view, robots are able to perform different tasks, such as entertainment, rehabilitation or medical assistant and even psychological therapy [13, 28]. Alexander Libin *et al* [28] concluded that the social robots have better interactions with humans if they have an anthropomorphized appearance and are able to imitate the facial expressions of humans, or with an animal appearance. David *et al* [13] conducted therapeutic sessions with the help of robots capable of expressing facial expressions with people with ASD (Autism Spectrum Disorder). It was verified that people responded better to treatment with the robot than without it. This happens because some people seem to have greater ease in recognizing the expressions in the robot than in a human. Hence this kind of robots, even at the psychological level are an added value to help the rehabilitation of humans.

3.2 Facial Expressions

Humans can make multiple facial expressions, but only six are considered as major: fear, sadness, joy, surprise, anger and disgust. But why? Why can not be others?

Paul Ekman [15] classified them as major, since they are the expressions that best characterize in a general way all the reactions and all emotional states that a human can have. Any other expression is likely to be associated with the major six ones. Each one of these emotions are generated through distinct circumstances and are characterized as follows [15, 43]:

- **Fear** → is the emotional response to an immediate threat. Fear can also be generated in anticipation of a perceived threat that puts integrity at risk.
- **Sadness** → is an emotional pain characterized by feelings of disappointment, hopelessness and loss.
- **Joy** → is defined as a pleasant emotional state that is characterized by feelings of contentment, gratification and satisfaction.
- **Surprise** → is the emotional response to something unexpected that just happened. This type of emotion can be positive, neutral and negative.
- **Anger** → is a powerful emotion characterized by feelings of hostility, agitation, frustration toward others. These kind of feelings are generated in the face of a threatening situation and prepare the body for a fight.
- **Disgust** → is a repulsive emotion that can be caused by taste, sight or smell. When people smell or taste foods that have gone bad, for example, disgust is a typical reaction. This can be a way the body can avoid some diseases.

3.2.1 Relation between the emotion states

After *James A.Russell* [39] conducted a study of assessing emotional behavior in humans, a circumplex model has been developed in order to categorize the emotion states and the relation between them (see Fig. 3.1). In this model, emotions are placed in a two-dimensional coordinate space with two axes to describe the emotions: arousal and valence/pleasure. Emotions are placed on the periphery of the circle around its center, which represents a neutral emotion. The distance between an emotion and the neutral center describes the intensity that emotion must have. The circumplex model also shows the emotional states intermediate that are made between two facial emotions.

Based on this model, *Steffen Witting* [53] has developed an application capable of expressing parametrized facial expressions, as well as transition animations between them. In addiction to choosing emotion, it is also possible to change some parameters, such as, arousal and pleasure, thus allowing to check the intensity of the emotion (see Figure 3.2).

The circumplex model has become a positive point in the development of facial movements for the robots whose purpose is to express facial emotions, ensuring smooth and

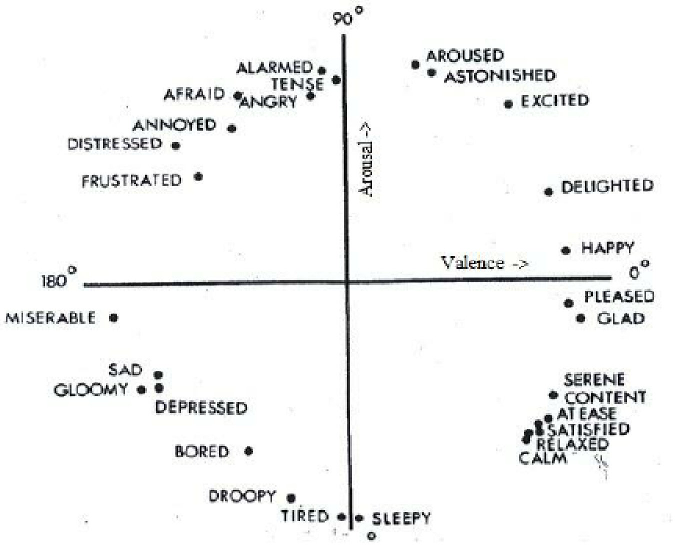
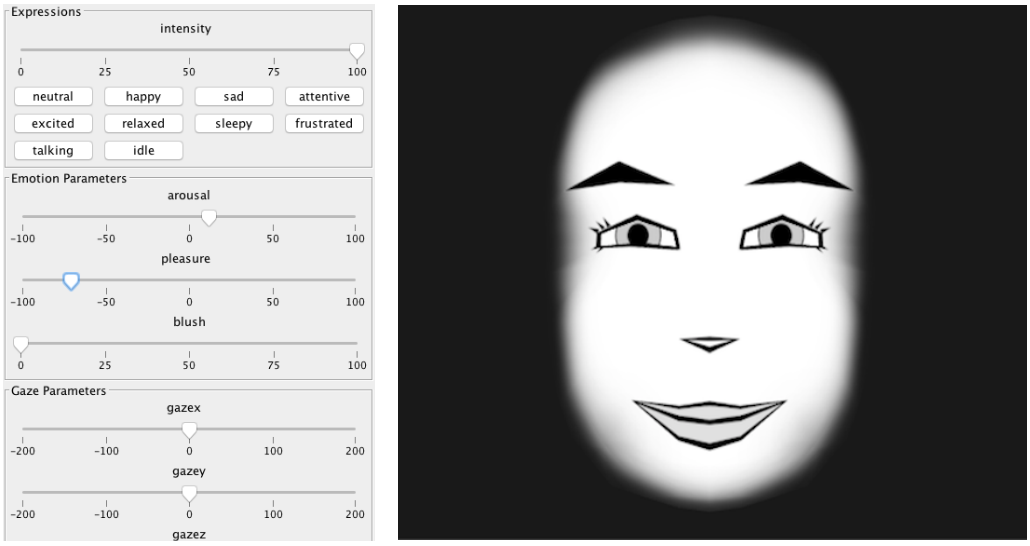


Fig. 3.1 Russell’s circumplex model for 28 effect words [39].



(a) Inputs.

(b) Result.

Fig. 3.2 Application to express facial expressions. Work developed by Steffen Wittig [53].

comfortable transitions between emotional states to the humans eyes. For that end, some of the social robots capable to display facial emotions expressions, such as EDDIE [45], use the circumplex model as a base to make movements of their eyes and mouth when changing their emotional state.

3.3 Mesh Warping

Mesh warping is a technique used to perform a smooth transition between images. Algorithm 1 shows how this technique works. In more detail, the algorithm needs a source image (I_S) and a destination image (I_D). Both images have meshes associated, the source mesh (M_S) that contains the coordinates of control points and the destination mesh (M_D) that contains the corresponding positions in the destination image. Together, M_S and M_D are used to define the spatial transformation that maps all points selected in I_S onto I_T. Each intermediate frame generated in the morph sequences uses a linearly interpolate mesh M, between M_S and M_D. The next step is to warp the source image to the I_1 using meshes M_S and M. Then, M_D is warped onto I_2 using meshes M_D and M. At last, a linearly interpolate image was made between I_1 and I_2, thus forming the intermediate frame.

Algorithm 1 Mesh Warping. Adapted from [54]

```
Inputs: Source image ($I_S$), Destination image($I_D$), Number of intermediate images ($N$)
Initialization:
$M_S \leftarrow I_S$-linked mesh                    ▷ Coordinates of control points
$M_D \leftarrow I_D$-linked mesh                    ▷ Corresponding coordinates

for $i \leftarrow 1$ to $N$ do
    $M \leftarrow$ linearly interpolate between $M_S$ and $M_D$
    $I_1 \leftarrow$ warped $I_S$ using $M_S$ and $M$
    $I_2 \leftarrow$ warped $I_D$ using $M_D$ and $M$
    $I_i \leftarrow$ linearly interpolate between $I_1$ and $I_2$
    Output: $I_i$
end for
```

3.4 Speech Recognition

Another way to interact with robots is by talking to them and nowadays, talking to robots or with any electronic device, is perfectly common and understanding the speech is a crucial step for natural Human-Robot Interaction (HRI).

3.4.1 Speech Recognition Overview

Speech recognition is also known as Automatic Speech Recognition (ASR), which means converting the human or computer speech to text. To achieve that goal, many typologies of speech recognition may be used [19, 40] such as:

- **Speaker dependent:** systems that have been adapted to a single speaker;
- **Speaker independent:** systems that have been adapted for any speaker;
- **Isolated word recognizers:** systems that only accept one word at a time but allow a naturally continuous speech;
- **Connected word:** systems that allow to speak slowly with a short pause between the words;
- **Continuous speech:** systems that allow speaking naturally.

The main components of any ASR system can be seen in Fig. 3.3. The first step consists in passing the sample speech or the capture in real time. Then, the collected sample goes through a feature extraction process, where the input signal is converted into a sequence of feature vectors [40]. In order to accomplish this phase, there are many techniques, such as: Linear Prediction Coefficients [37], Perceptual Linear Prediction [22] and the most used overall Mel-Frequency Cepstral Coefficients [14].

The next stage, *Decoding*, is responsible for finding the best match in the incoming feature vectors. As seen in Fig. 3.3, this stage makes use of an Acoustic Model and Language Model. The Acoustic Model is a statistically characterization of the speech that is being recognized. It is modeled using training sets of the speech that needs to be recognized. To build the acoustic model, many approaches have been applied, such as: Dynamic Bayesian Networks [29], Support Vector Machines [20], Artificial Neural Networks [35] and the most used overall Hidden Markov Model [40, 41, 5, 35]. The Language Model [7] models the word sequence probability distribution of a specific language. In other words it imposes on the recognition system the grammatical rules of the language.

The main metric used to evaluating the accuracy of an ASR system, no matter which approach/algorithm is used, is the Word Error Rate (WER): $WER = (I+D+S)/N$ where I is the number of words inserted, D words deleted, S words substituted and N the total number of words in the original phrase.

3.4.2 Available Speech Recognition Libraries

In order to perform a speech recognition, there are many commercial software systems available to perform it, such as: Microsoft Bing Speech [48], Google web Speech API

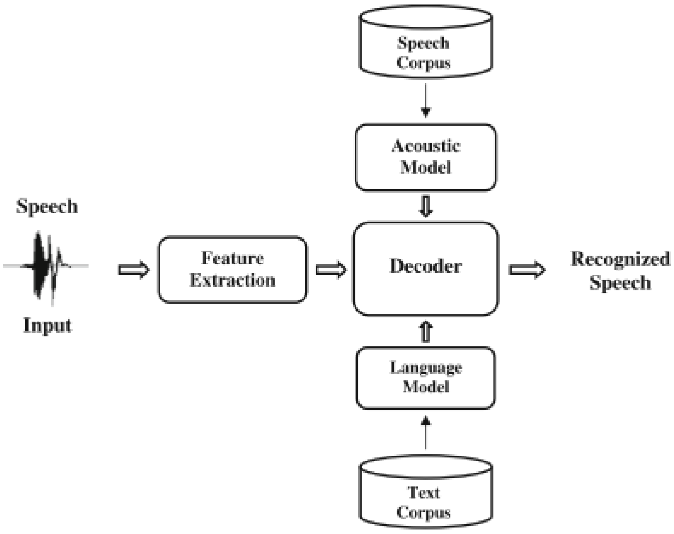


Fig. 3.3 Speech recognition system overview [42].

[47] and IBM Speech to Text [46]. Besides those software systems, there are others non-commercial speech recognition software systems, such as: JANUS [16], CMU Sphinx [27] and Kaldi [38] being the latter two open-source.

In order to use a speech recognition software system, an open-source library available in *GitHub* [55] *SpeechRecognition.py* developed by Anthony Zhang was evaluated. This library allows the use of five speech recognition software systems making all the required connections to ensure the user-friendliness. The user only needs to choose the service to be used and to make the sound capture. Table 3.1 shows the five systems supported by this library and some characteristics of these systems.

	Speech Recognition systems				
	Microsoft Bing Speech	Google Speech API	Wit.ai	IBM	CMU Sphinx
Supported languages	29 including English and Portuguese	119 including English and Portuguese	73 including English	11 including English	6+ including English
Online or Offline	Online	Online	Online	Online	Offline
Acoustic Model	HMM	HMM	-	HMM	HMM
WER [25]	18%	9%	-	-	37%

Table 3.1 Speech recognition systems supported in the open-source library.

However, this library does not allow a real-time recognition, since it is necessary to capture the sound first and then send it to the desired system.

Chapter 4

ISR-RobotHead Facial Expressions Modules - Software and Hardware

In this chapter, we introduce in detail the hardware and software modules, developed for the new version of ISR-RobotHead.

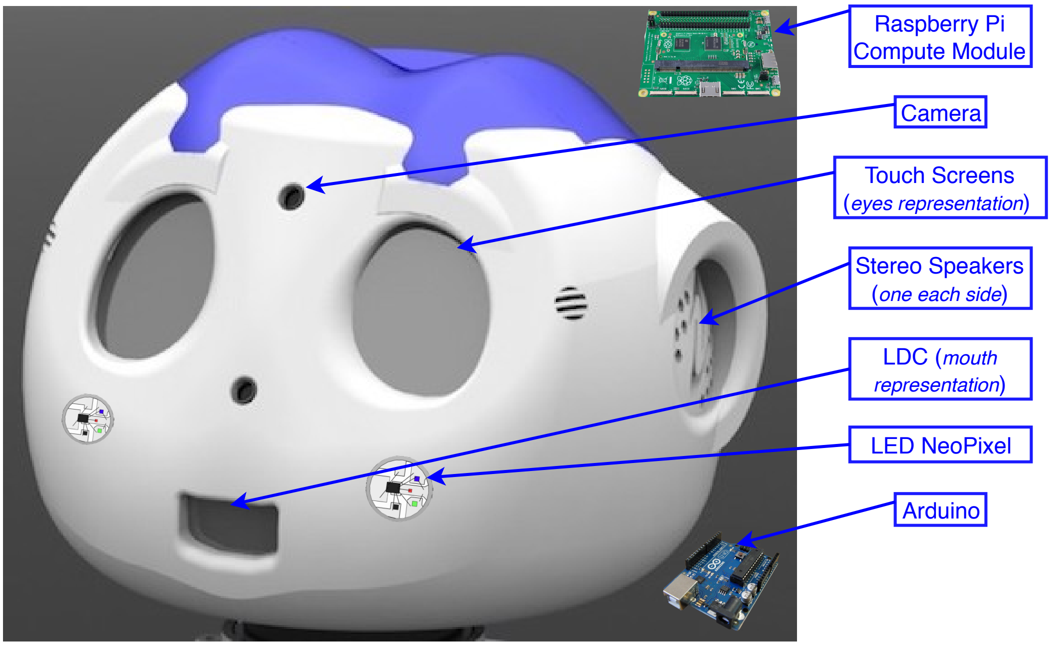

Raspberry Pi Compute Module
Camera
Touch Screens (*eyes representation*)
Stereo Speakers (*one each side*)
LDC (*mouth representation*)
LED NeoPixel
Arduino

Fig. 4.1 Layout drawing and hardware available in the ISR-RobotHead.

The robotic head has several peripherals to enhance the HRI (Human-Robot Interaction), which are the following: a sound output system, two cameras, addressable LEDs and three LCDs to display facial expressions, two for the eyes and one for the mouth. All these peripherals are controlled by a Raspberry Pi with the exception of the addressable LEDs, which are controlled by an Arduino.

One of the main goals of the underlying research work was to improve the ISR-RobotHead with better characteristics for applications of Children-Robot Interactions (CRI). In this way, libraries were developed in *C* programming language for the Raspberry-Pi Compute Module 3 (RPiCM3) in order to control the expressions displayed in both eyes and mouth LCDs.

4.1 Raspberry Pi Development Kit

4.1.1 Raspberry Pi Overview

The Raspberry Pi (RPi) is a small and affordable computer that operates with a Debian based Linux distribution operating system. This computer can interpret several programming languages such as *C* or *Python* and, in order to interconnect with external hardware, it has a several GPIOs (General Purpose Input/Output) available. The best way to program the RPi is through a secure shell (SSH) connection allowing to use an external computer.

4.1.2 Raspberry Pi 3 Model B VS Raspberry Pi Compute Module 3

In the previous ISR-RobotHead version, the Raspberry Pi Compute Module was used to control the majority of the sensing and control resources. A higher CPU (Central Processing Unit) speed, more RAM (Random Access Memory) and higher storage is recommended, since the Raspberry Pi Compute Module is limited to 4GBytes eMMC Flash. As an alternative, the new Raspberry Pi models are greatly improved by presenting the main specifications in Table 4.1.

By comparing the two new devices, Raspberry Pi 3 Model B and RPiCM3 with RPiCM, it becomes clear that the new models are far superior than the current one used in the RH v1.0. Comparing the new Raspberry Pi models the major difference is the number of GPIOs and the number of communication channels/protocols. Since we have several devices to control, such as three LCDs and two cameras, it is necessary to have the highest number of communication protocols and GPIOs available, so for these reasons, RPiCM3 becomes the best choice to integrate into the Robot Head (RH). Figure 4.2 shows the RPiCM3 development device board, where the blue boxes signals the differences between this device and the one previously used (RPiCM).

For this project the Raspbian Stretch operating system was used. It is a system based on Debian GNU/Linux optimized for the Raspberry Pi Hardware. The default operating

	RPiCM (previously used)	Raspberry Pi 3 Model B	RPiCM 3
CPU	700 MHz	1.2 GHz	1.2 GHz
RAM	512 Mbytes	1 Gbyte	1 Gbyte
USB Ports	1	4	1
HDD	4Gbytes eMMC Flash	Micro SD Card	4 Gbytes eMMC Flash or Micro SD Card
Cameras	2	1	2
GPIO	120	40	120
Ethernet Port	No	Yes	No
Communication Protocols	2·I^{2}C, 2·SPI, 2·UART	1·I^{2}C, 1·SPI, 1·UART	2·I^{2}C, 2·SPI, 2·UART

Table 4.1 Comparison between the previously RPiCM used, Raspberry Pi 3 Model B and RPiCM3.

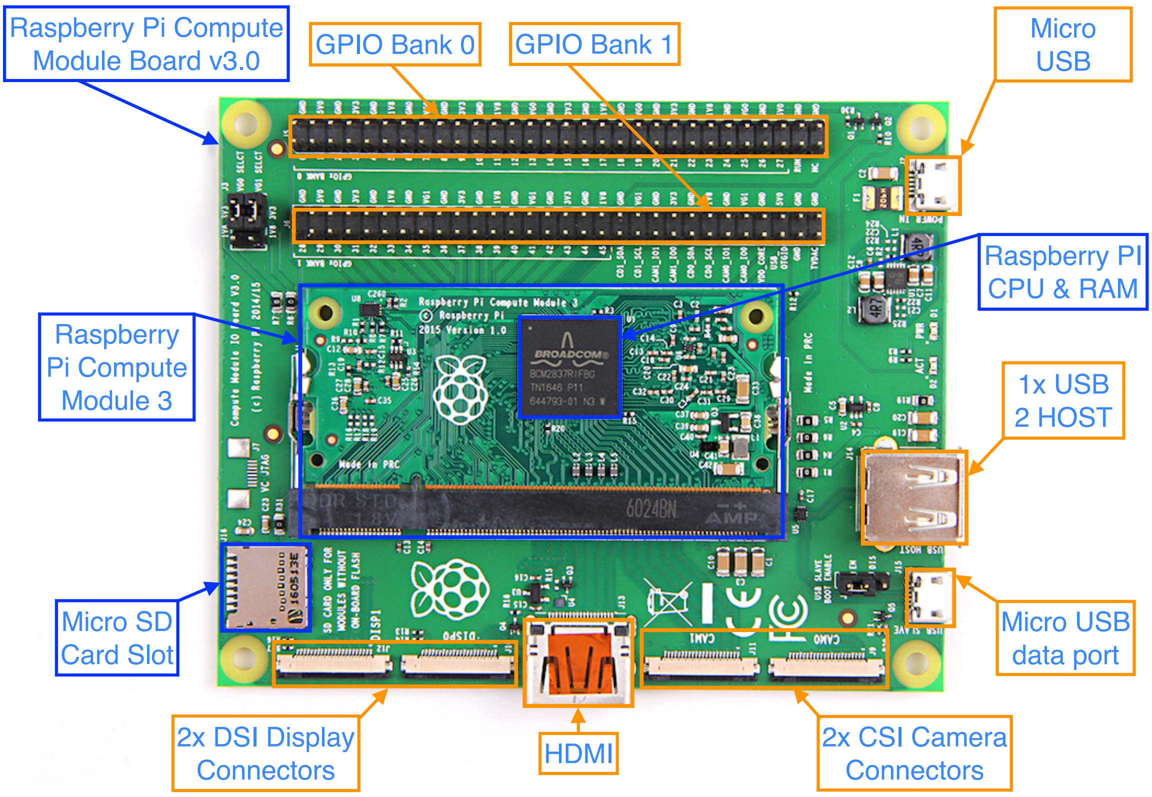


Fig. 4.2 Raspberry Pi Compute Module 3 mounted on the development kit board. The blue boxes signals the differences between RPiCM3 and the one previously used (RPiCM). (Image taken from [17]).

system only has software routines for UART0, SPI0 and I^2C0 peripherals. In order to use all communication channels available by this Raspberry Pi, it was necessary to change the configuration files, thus allowing to use the UART1, SPI1 and I^2C1. Similarly to the implementation of RH v1.1 in RH v2.0, the *Wiring Pi* library [21] was used for to programming Raspberry Pi peripherals.

4.2 Eyes Expressions Module

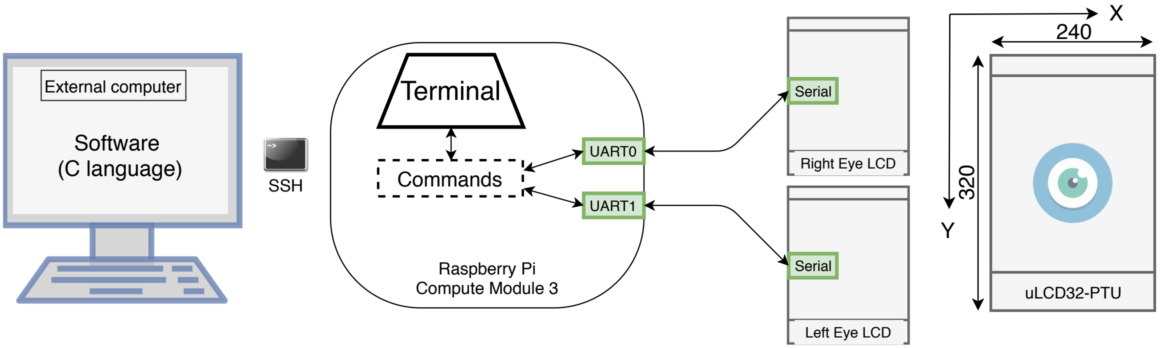
As described in Chapter 2, different communication protocols were used to control eyes and mouth LCDs. For the eyes the UART communication protocol was used, while for the mouth, the SPI communication protocol was used. With the objective to use only one communication protocol to operate the three LCDs, an evaluation was made between the LCD used in the previous version and a new LCD, 3.5inch RPi LCD. Table 4.2 shows the main features of both LCDs.

uLCD-32PTU	**3.5inch RPi LCD**
240×320 pixels resolution	320×480 pixels resolution
PICASSO processor	XPT2046 microcontroller
Serial communication protocol	SPI communication protocol
Touch screen	Touch screen
Micro-SD card slot	-

Table 4.2 Main features of the uLCD-32PTU and the 3.5inch RPi LCD.

The use of the 3.5inch RPi LCD would be advantageous since the SPI communication protocol sends the data at a higher rate than the serial communication (UART), and thus, the images that contain the eye expressions would be stored in the RPiCM3's memory. However, the Raspberry Pi only has two SPI channels available which make it a limitation to communicate with the three LCDs at the same time. With this, the best option was to maintain the LCD already used, uLCD-32PTU [1].

In the communication between the RPiCM3 and the LCDs, the two available UART communication channels were used, one for each LCD, thus allowing independent control of each LCD, unlike in the previous work where the LCDs were connected in parallel. Figure 4.3 shows a overview of the ISR-RobotHead's eyes software and hardware modules.



(a) System overview.

(b) Pixels dimensions.

Fig. 4.3 Overview of the ISR-RobotHead's eyes software/hardware modules.

4.2.1 RobotHead Eyes Library

For the Raspberry Pi Compute Module 3, a library in C was developed, **RHEyesLib.c**, to configure the UART0 and UART1 peripherals with all the GPIO configurations required and to send the required commands in order to display the expressions. These configurations allows to communicate with the two LCDs at the same time.

The eye's LCD can display computer graphics primitives as well as images and video from micro-SD card. The PICASSO processor has 135 commands and operates with a baud rate of 9600. Each data transmission, which can be a command or a parameter, sent to the LCD is composed by a 16 bit word.

4.2.1.1 Eyes Images

In the work previously developed, the *4D System Graphics Composer* program was used to load images from the external computer to the micro-SD card. This process had to be done manually. In this new version of the ISR-RobotHead two possible alternatives have been studied in order to display the eyes expressions: 1) Store a binary file of each image in the Raspberry Pi's memory and load to the LCD when required; 2) Drawing images directly on the LCD with the computer graphics primitives available in the PICASSO processor.

The first option was implemented/tested and it quickly became obvious that it was not viable, since it took more than ten minutes to send and show an image on the screen. After that, it became clear that there was a need to redraw all expressions directly on the screen, thus allowing to make improvements in all expressions.

In order to draw the eye expressions, some of the computer graphics primitives available in the PICASSO processor were used, thus avoiding the necessity of building a pixel matrix.

With this, each eye expression frame is formed through a set of commands capable of drawing geometric figures on the screen such as circles, ellipses, and polygons. Table 4.3 shows all the commands used in the development of the eye expressions.

Command	Description	Command	Description
0×FFCD	Clear Screen	0×0014	Draw Filled Polygon
0×FFC2	Draw Filled Circle	0×FFB1	Draw Filled Ellipse
0×FFC4	Draw Filled Rectangle	0×FFA4	Background Color

Table 4.3 Graphical primitives commands used.

Each eye expression developed follows the pattern presented in Fig. 4.4, which represents the sequence of commands used to form the neutral expression where the variables x and y correspond to the eye center coordinates and r to the radius. The sequence begins by displaying the eyebrow through two ellipse commands. Then, a kind of eyelid is drawn through two circle commands. At last, the eye is drawn with seven circle and a polygon commands.

```
DrawFilledEllipse(x-10,y-55,r+20,r-20,black, eye_port)
DrawFilledEllipse(x-10,y-53,r+22,r-22,white, eye_port)

DrawFilledCircle(x-5,y-2,r+12,black, eye_port)
DrawFilledCircle(x-1,y+5,r+12,white, eye_port)

DrawFilledCircle(x,y,r,eyeColor, eye_port)
DrawFilledCircle(x,y,r-8,black, eye_port)
DrawFilledCircle(x,y,r-12,eyeColor, eye_port)
DrawFilledCircle(x,y,r-15,black, eye_port)
DrawFilledCircle(x,y,r-33,white, eye_port)
DrawFilledCircle(x+10,y+8,1,white, eye_port)
DrawFilledCircle(x+13,y-6,2,white, eye_port)
DrawFilledPolygon(Xarray_points, Yarray_points, white, eye_port)
```

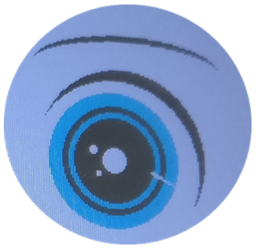


Fig. 4.4 Neutral eye expression. Sequence of commands used (left) and image result (right).

The sequence of commands for the sadness and surprise eye expressions are similar to the neutral expression, however, the remaining expressions need a different sequence. Figure 4.5 shows the sequence of commands used to form the anger expression. It starts by drawing the neutral expression and then three commands are added. The sequence of commands used for the remains eye's expressions are presented in Appendix A

In addition to developing eye expressions, transitions between the neutral state and all other expressions were developed. In almost all the transitions between animations four frames were used. Figure 4.6 shows an example of intermediate images used between the neutral and anger emotional states. In total, 102 frames were developed.

```
DrawFilledEllipse(x-10,y-55,r+20,r-20,black, eye_port)
DrawFilledEllipse(x-22,y-53,r+22,r-22,white, eye_port)

DrawFilledCircle(x-5,y-2,r+12,black, eye_port)
DrawFilledCircle(x-1,y+5,r+12,white, eye_port)

DrawFilledCircle(x,y,r,eyeColor, eye_port)
DrawFilledCircle(x,y,r-8,black, eye_port)
DrawFilledCircle(x,y,r-12,eyeColor, eye_port)
DrawFilledCircle(x,y,r-15,black, eye_port)
DrawFilledCircle(x,y,r-33,white, eye_port)
DrawFilledCircle(x+10,y+8,1,white, eye_port)
DrawFilledCircle(x+13,y-6,2,white, eye_port)
DrawFilledPolygon(Xarray_points, Yarray_points, white, eye_port)

DrawFilledRectangle(x-50,y-55,x+55,y-35,white, eye_port);
DrawFilledPolygon(Xarray_points, Yarray_points, black, eye_port);
DrawFilledPolygon(Xarray_points, Yarray_points, white, eye_port);
```

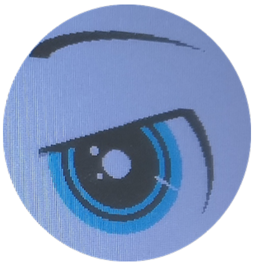
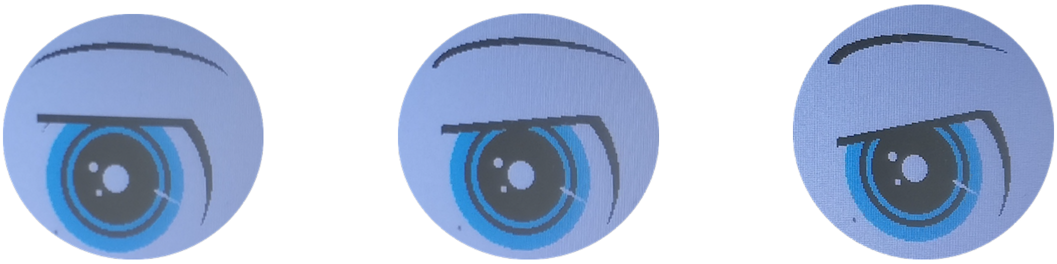


Fig. 4.5 Anger eye expression. Sequence of commands used (left) and image result (right).



(a) Intermediate image 1.

(b) Intermediate image 2.

(c) Intermediate image 3.

Fig. 4.6 Intermediate images of the eye between the neutral and anger emotional states.

4.2.1.2 Display Eye Expression

In order to display eye expressions, two possible alternatives were tested: 1) Send to the LCD all the graphical primitives required commands to build an image; 2) Save the image in the micro-SD card inserted in the LCD and read from it.

Once again, the first option was not viable. It is possible to show all expressions and animations between them but the commands are not send quickly enough to form an image without the formation being noticed. Unfortunately these LCDs do not have a *double buffering* option, so the best way to display the eyes expressions is through the micro-SD card.

Since the eye expressions were drawn with the graphical primitives commands, it is no longer required to remove the micro-SD from the LCD every time it is necessary to add or change an image. Once again, in order to save and display the images on the screen, the PICASSO processor has some commands available that allows an easy interconnection between the LCD and de micro-SD card. Table 4.4 shows all the commands used to display the eye expressions.

Command	**Description**	**Command**	**Description**
0×0005	File Exists	0×FF11	Display Image
0×000A	File Open	0×FF10	Screen Capture
0×FF18	File Close	0×0003	File Erase
0×FF03	File Mount	0×FF02	File Unmount

Table 4.4 File commands used.

The first step is to save the image (or a set of them) in the micro-SD. For that, a file is created with the name of the expression and all the images required are saved through the screen capture command available in the PICASSO processor. The size of any screen capture made is always 80×100 pixels. Once stored in the micro-SD card, it is only needed to read the image from the file and display it on the screen through the display image command. Algorithm 2 shows how to display an image (or a set of them) on the LCD.

4.2.2 Touch screen image

As already mentioned, the LCDs used to display the eye expressions are touch screen. So, in order to activate and use this feature the commands shown in Table 4.5 were used.

Command	Description	Command	Description
0xFF39	Touch Detect Region	0xFF38	Touch set
0xFF37	Touch Get		

Table 4.5 Touch screen commands used.

In order to react when the LCD is touched, a new image has been developed to display in these circumstances. Figure 4.7 shows the sequence of commands used and the image result.

```
DrawFIlledEllipse(x-10,y-45,r+20,r-20,black , eye_port)
DrawFilledEllipse(x-12,y-42,r+23,r-24,white , eye_port)
DrawFilledEllipse(x+32,y-55,r-10,r-30,white , eye_port)

DrawFIlledEllipse(x-2,y-82,r ,r-20,black , eye_port)
DrawFilledEllipse(x-4,y-85,r+2,r-20,white , eye_port)
DrawFilledEllipse(x-40,y-82,r+2,r-22,white , eye_port)

DrawFilledEllipse(x,y,r,3,black , eye_port)  145,175,50
```

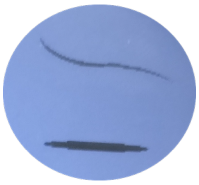


Fig. 4.7 Touched eye expression. Sequence of commands used (left) and image result (right).

Algorithm 2 Explanation of how the image is saved (or a set of them) in the micro-SD card and displayed it on the screen.

```
Inputs: LCD to communicate ($eye$), File descriptor communication ($eye_{port}$), X LCD position ($x$), Y LCD position ($y$)
Initialization:
ClearScreen                                          ▷ Clears everything on the screen
$n \leftarrow$ number of images to display
$Y_{img\_center} \leftarrow y + 20$                  ▷ Y LCD position for start the image display
if $eye == RightEye$ then
    $X_{img\_center} \leftarrow x + 20$              ▷ X LCD position for start the image display
else
    $X_{img\_center} \leftarrow 120 - x$
end if

if $FileExists(emotion, eye_{port})$ then            ▷ Verifies if the file with the images already exists
    $handle \leftarrow$ FileOpen($emotion, eye_{port}$)
    for $k = 1$ to $n$ do
        $DisplayImage(X_{img\_center}, Y_{img\_center}, eye_{port})$
    end for
else                                                 ▷ If a file does not exists, create one
    $handle \leftarrow$ FileOpen($emotion, eye_{port}$)
    for $k = 1$ to $n$ do
        Sequence of commands to form an emotional eye expression
        ScreenCapture($eye_{port}$)                  ▷ Save the image in the micro-SD card
    end for
end if
FileClose($handle, eye_{port}$)
```

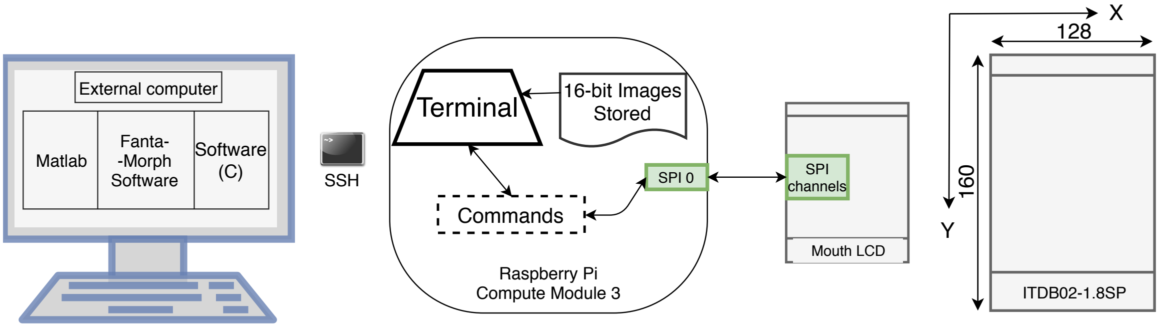
4.3 Mouth Expression Module

In order to display mouth expressions an ITDB02-1.8SP LCD [23] was used. Table 4.6 shows the main features of this LCD.

ITDB02-1.8SP	
ST7735 microcontroller	SPI communication protocol
128x160 pixels resolution	16 bit-color

Table 4.6 Main features of the ITDB02-1.8SP LCD.

To display an image in this LCD, 16 bits blocks must be send where each block defines the color of a pixel. This means that each expressions is represented by 128×160×16 bits. Each mouth image developed is stored in the RPiCM3 memory as binary files, unlike the eyes' images. Figure 4.8 shows an overview of the ISR-RobotHead's mouth software and hardware modules.



(a) System overview.

(b) Pixels dimensions.

Fig. 4.8 Software/Hardware overview of the mouth module.

4.3.1 Mouth Images

Unlike the eye expressions, the mouth expressions were developed in the image editor program *Paint S* available for the *macOS*. This program is an easy-to-use drawing tool, which give us the easy ability to draw any expression. Figure 4.9 shows two images created for the two mouth expressions, neutral and anger.

(a) Neutral mouth expression. (b) Anger mouth expression.

Fig. 4.9 Mouth expressions images.

4.3.1.1 Mouth intermediate images

Since the mouth expressions were built from scratch, it was also necessary to develop intermediate images to build the transitions between each two expressions. To achieve it, a program capable of generating a smooth transition between two images was used. The program is *FantaMorph* [2] developed by *Abrosoft*. Giving the initial and final images, this program is capable of generate a set of intermediate images. To achieve it, the *FantaMorph* program is based in the Mesh Warping algorithm described in Section 3.3.

The main features of this program can be seen in Fig. 4.10 and a sequence of intermediate images generated by this program can be seen in Fig. 4.11.

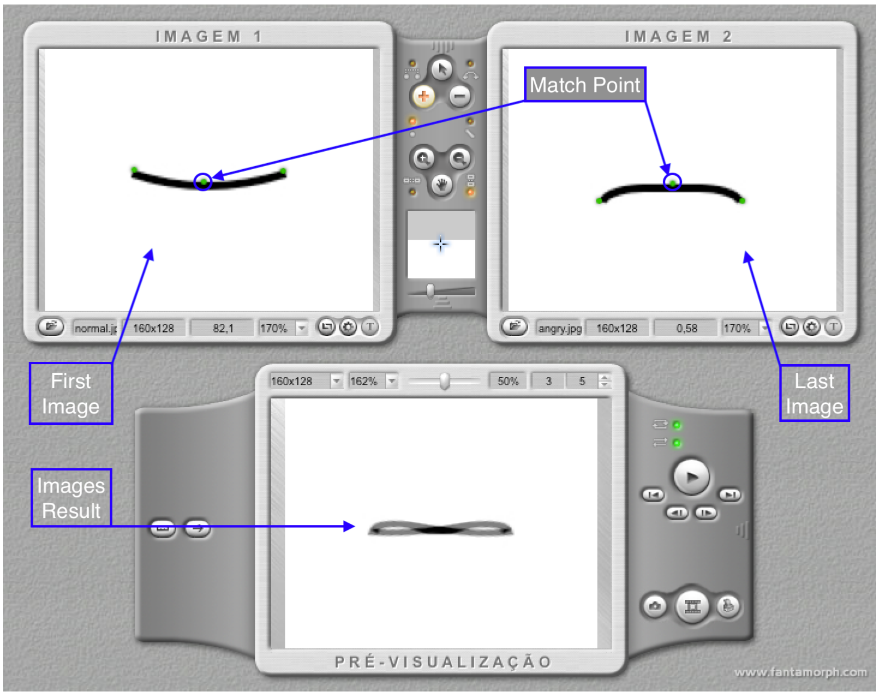
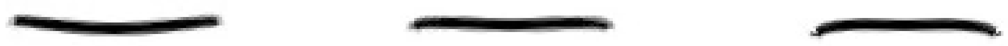


Fig. 4.10 Interface of the *FantaMorph* software program.



(a) Intermediate image 1.

(b) Intermediate image 2.

(c) Intermediate image 3.

Fig. 4.11 Intermediate images of the mouth between the neutral and anger emotional states.

4.3.1.2 Mouth image's pre-processing

The LCD used can only display images in 16 bit blocks. Each block has 5 bits for red, 6 for green and 5 for blue color. It turns out that the generated images were all 24 bit format, 8 bits for each RBG color, so it was necessary to convert all images to the 16 bit format. Quantization, involving image processing, is a lossy compression technique achieved by compressing a range of color levels to a single color level (as illustrated in Figure 4.12). In the previous work [31] a script was developed in *Matlab* capable of performing this quantization from 24 bits to 16 bit. In this project, the same script was used without making any changes.

The output of the script is a binary file, where the value that each pixel is written over 2-byte (16-bit) words.

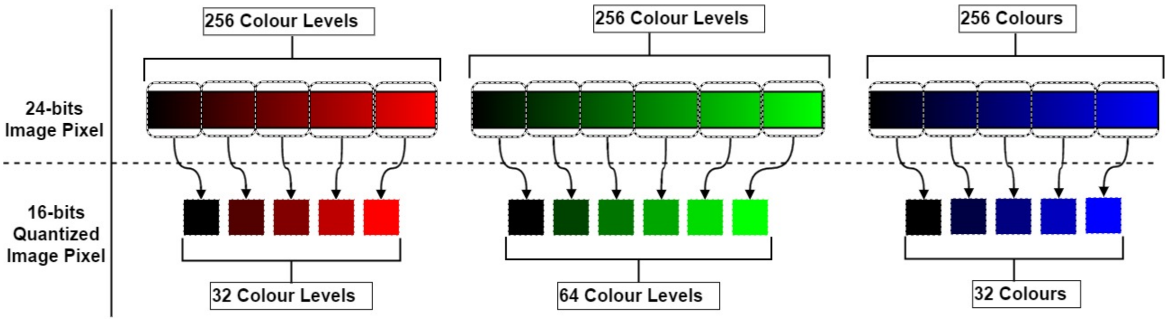


Fig. 4.12 Illustration of the quantization of 24 bit image to 16 bit. Image taken from the previous work [31].

4.3.2 RobotHead Mouth Library

For the Raspberry Pi Compute Module 3, a library in C was developed, *RHMouthLib.c*, in order to configure the SPI0 peripherals with all the GPIO configurations required, load and send images to the mouth's LCD.

4.3.2.1 Mouth LCD Set-up

Unlike the eye's LCDs, this one needs an initial setup. The microcontroller needs to be properly configured so that it can receive the images. This initial setup consists in sending a set of specific commands and parameters to the LCD. Algorithm 3 shows in more detail how the initial setup is made where the set of commands and parameters are represented in $configurationArray$.

As it is known, the SPI communication protocol has a digital channel known as RS that through its state (0 or 1) identifies whether the incoming data must be interpreted as a argument or a command.

4.3.2.2 Display Mouth Expression

In order to display mouth expressions and since the images are stored in the Raspberry Pi's memory, a *Display_image_mouth* function was developed in the library *RHMouthLib.c* in order to load the image file data onto the LCD. This function, shown in Algorithm 4, load the data of the binary file to the buffer and send the buffer data to the mouth's LCD.

Figure 4.13 shows the flowchart of the used procedure to display mouth expressions.

Algorithm 3 Send commands/arguments to perform initial LCD setup.

```
Initialization:
$N \leftarrow$ Sizeof($configurationArray$)
Configure the five communication peripherals.
$Fd_{SPI} \leftarrow SPI\_Setup()$

for $k \leftarrow 0$ $k$ to $N$ do
    if $k$ is a command then
        RS pin $\leftarrow$ $low$
    else
        RS pin $\leftarrow$ $high$
    end if
    write($Fd_{SPI}, configurarionArray[k]$)
end for
Output: $Fd_{SPI}$
```

Algorithm 4 *Display_image_mouth* function. Send image file data to the mouth's LCD.

```
Inputs: SPI file descriptor ($fd_{spi}$), Image file descriptor($fd_{img}$)
Initialization:
$buffer_{size} \leftarrow 1024$
$buffer[buffer_{size}]$                                  ▷ Buffer with 1024 bytes

for $i \leftarrow 0$ to endOfFile($fd_{img}$) do
    read($fd_{img}, buffer$)
    write($fd_{spi}, buffer$)
end for
```

As it can be seen, this module starts with the initial setup and then waits for an input signal. When the input signal arrives, a command is sent to the LCD indicating that the following data is to be displayed on the screen. After that, the *Display_image_mouth* function loads the data from the binary file and sends to the LCD.

4.4 Facial Color Expression

In order to improve the facial expressions, addressable LEDs were used to give color to some expressions. Two NeoPixel digital RGB LEDs were used, one on each side of the robot face, more precisely, on the cheek. As shown in the left of the Fig. 4.15 the NeoPixel is characterized by: 1) 3 LEDs (one for each RGB color); 2) each LED has 8 bits; 3) WS2812 Controller Chip; 4) require a PWM signal as input.

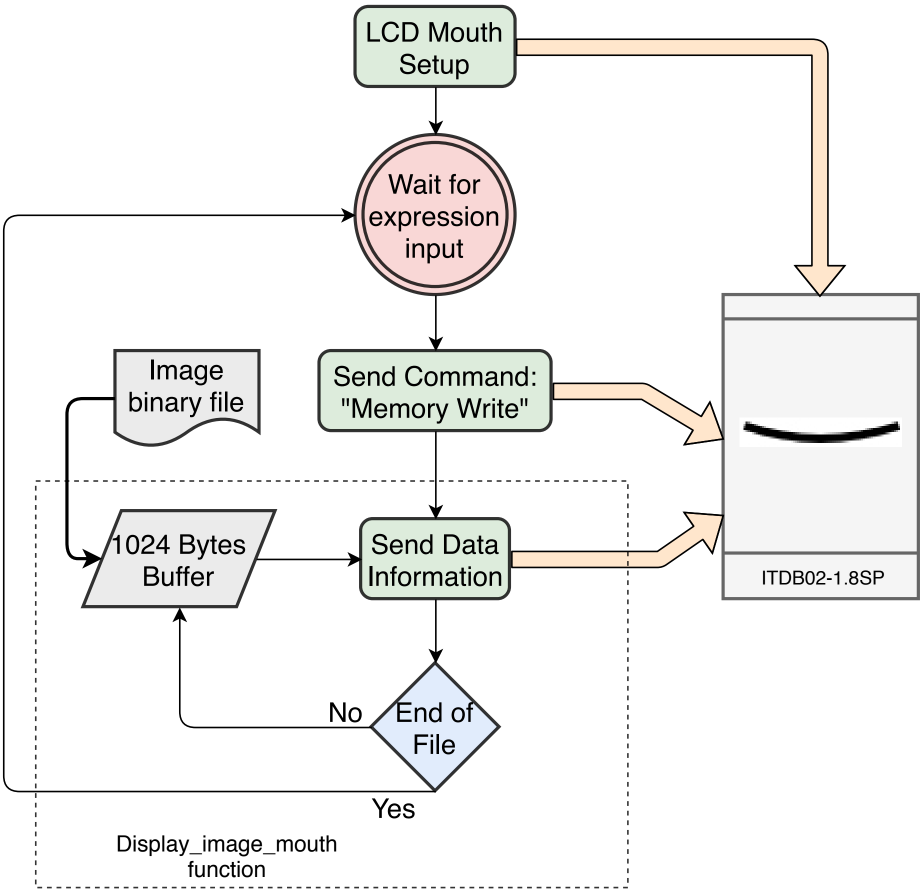


Fig. 4.13 Flowchart of the procedure to display mouth expressions.

To control the color of the NeoPixeis an *Arduino Duemilanove* microcontroller was used connected to the RPiCM3. Since we only used two different colors in the expressions (plus no color at all), we only needed three states to represent them, and thus two bits are enough to encode them between the RPiCM3 and the Arduino. With this purpose, two digital pins were connected between the RPiCM3 (pin 20 and 21) and the Arduino (pin 4 and 5) as shown in the Fig. 4.14.

Two emotional, disgust and happiness, expressions were improved with the color on the cheek. For that, the connection made between the Arduino and the RPiCM3 allows to interpret four possible states thats are characterized in the right of the Fig 4.15.

The PWM (Pulse-Width Modulation) input signal must be at a frequency of 800KHz and for that, the library *Adafruit_NeoPixel.cpp* provided by the manufacture [3] was used. This library makes all the required Arduino settings for the color display of the NeoPixel, including the output signal (PWM). For the happiness expression it was used a red color

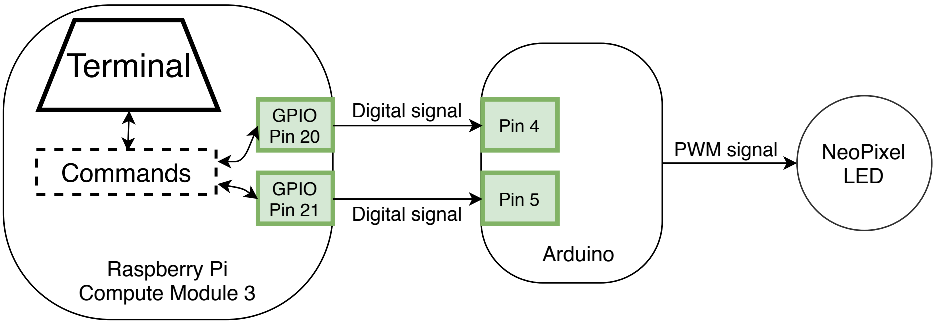
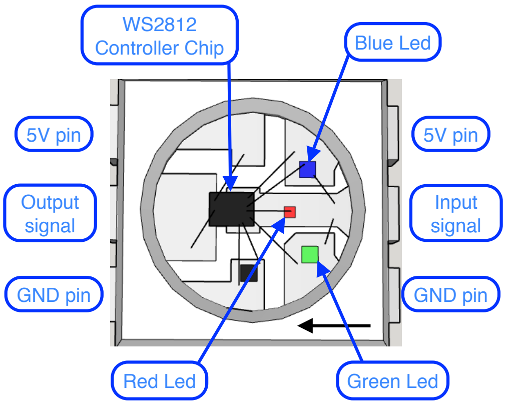


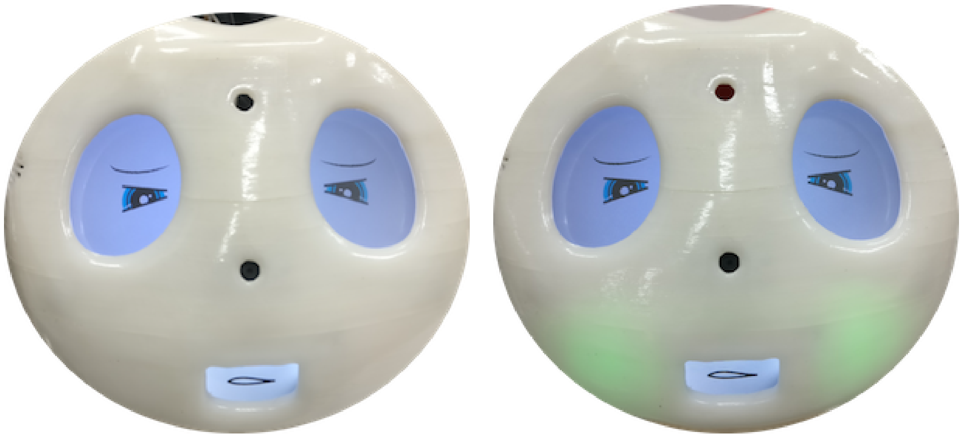
Fig. 4.14 All connections made of the displaying color module.



Arduino		
Pin 4	Pin 5	RGB
0	0	-
1	0	60.0.0
0	1	0.90.0
1	1	NA

Fig. 4.15 NeoPixel features (left) and all possible states for displaying color (right).

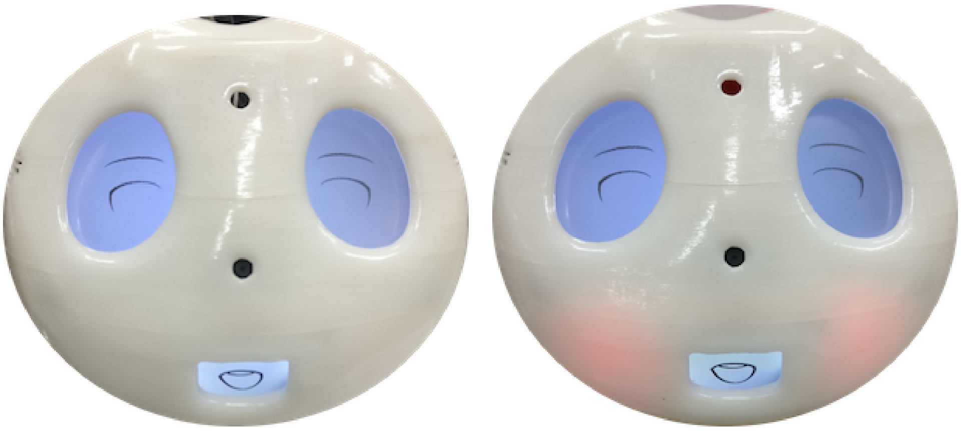
(RGB → 60.0.0) and for the disgust expression it was used a green color (RGB → 0.90.0). Both colors are presented in Fig. 4.16 and Fig 4.17.



(a) Without cheek color.

(b) With cheek color.

Fig. 4.16 Evaluation of the colors used for disgust emotional expression on the ISR-RobotHead's cheek.



(a) Without cheek color.

(b) With cheek color.

Fig. 4.17 Evaluation of the colors used for happiness emotional expression on the ISR-RobotHead's cheek.

Chapter 5

ISR-RobotHead Sound System Modules - Software and Hardware

This chapter introduces all the hardware and software modules used in the ISR-RobotHead to make it capable of reproducing sound and receive voice commands.

Besides displaying expressions, another main goal of this dissertation was to make the ISR-RobotHead capable of receiving voice commands and reproduce some predefined sounds. For this, a Matrix Voice kit [52] and a Raspberry Pi 3 Model B [18] (RPi3) were used to capture the sound. To reproduce sounds, the RH is equipped with two speakers. In order to orchestrate all the components, a library in python was developed, *RHSpeech.py*, for the RPi3.

5.1 Matrix Voice Hardware

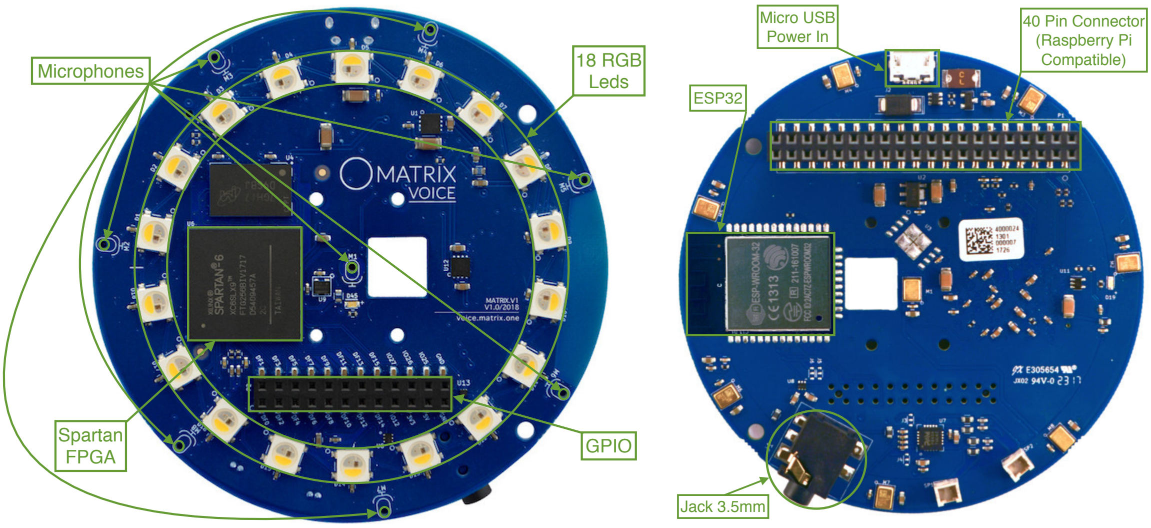
5.1.1 Matrix Voice Overview

The Matrix Voice (MV) is a small and affordable component that can be controlled by a Raspberry Pi or an ESP32 microcontroller. The focus of this component is to make sound captures and determine their direction, that is, to determine the orientation of where the sound comes from. To achieve this goal, the Matrix Voice board has the characteristics presented in Table 5.1. The Matrix Voice has 64MByte of RAM and 64Mbit of Flash Memory available. Figure 5.1 shows the Matrix Voice board. The manufacturer provides a few examples on how to use all the sensors available [34], such as: 1) turn LEDs on or off; 2) determine the sound orientation and represent the detected orientation by turning on some LEDs; 3) The use of microphones to record audio files. The incorporation of this component on the

ISR-RobotHead allows the capture of human voice and detection of which direction it is coming.

Matrix Voice board	
8 Microphones	16 RGB LEDs
24 Output GPIOs	40 Input GPIOs (Compatible with Raspberry Pi)
ESP32 microcontroller	Xilinx Spartan-6 FPGA

Table 5.1 Main features of the Matrix Voice board.



(a) Matrix Voice front.

(b) Matrix Voice back.

Fig. 5.1 Matrix Voice hardware used to acquire sound (Image taken from [10]).

5.2 Raspberry Pi 3 Model B Hardware

To integrate the Matrix Voice hardware into the ISR-RobotHead, a conventional Raspberry Pi (RPi) was necessary since the input peripherals of the Matrix Voice board are the same as the RPi3 output peripherals. The main characteristics of Raspberry Pi 3 are in Table 4.1 and the hardware kit can be seen in Fig. 5.2.

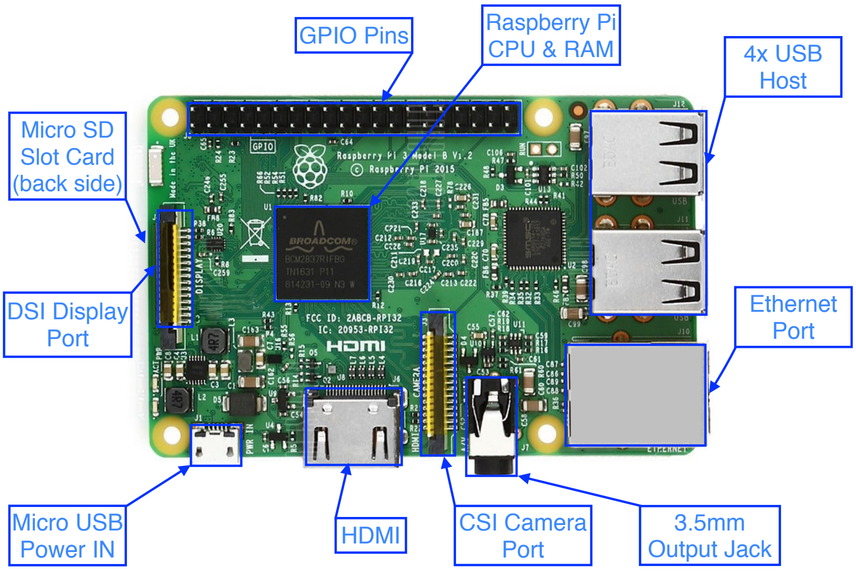


Fig. 5.2 Raspberry Pi 3 Model B Hardware.

5.3 Speech Recognition Module

The Matrix Voice hardware and a RPi3 were used to build a speech recognition module to capture and recognize human speech. Figure 5.3 shows an overview of the speech recognition module.

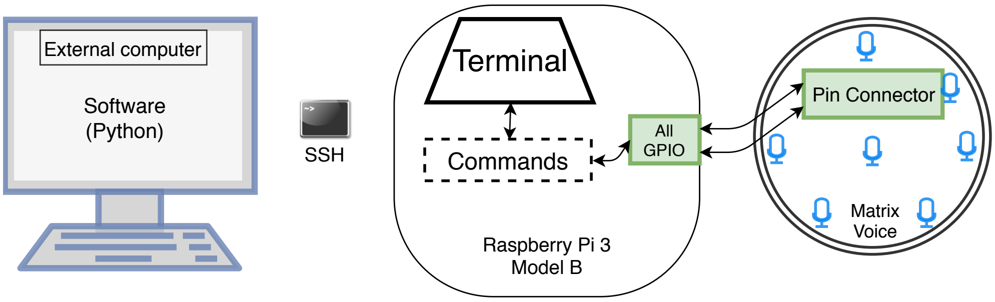


Fig. 5.3 Software/Hardware overview of the speech recognition module.

The sound waves are captured by the eight microphones and then the signal generated by each one is sent to the FPGA, which will combine all the data in order to obtain the highest resolution.

5.3.1 Speech Recognition Library

In order to do speech recognition over the sound capture, an open-source library described in Section 3.4.2, *SpeechRecognition.py*, was imported to the *RHSpeech.py*. Algorithm 5 shows how the library imported is integrated in this project.

Algorithm 5 Speech_Recognition function in the *RHSpeech.py* and initial setup.

```
Initialization:
$import\ speech\_recognition\ as\ sr$                          ▷ Import the SpeechRecognition.py
$r \leftarrow sr.Recognizer()$
$mic \leftarrow sr.Microphone(micprophone\_address)$              ▷ Choose the microphone to use.

function SPEECH_RECOGNITION($void$)
    $r.adjust\_for\_ambient\_noise(mic)$          ▷ Define automatically the signal threshold.
    $audio \leftarrow r.listen(mic)$                                    ▷ Start the sound capture.
    $StT \leftarrow r.recognize\_service()$    ▷ Request the software system to analyze the capture.
    Output: $StT$
end function
```

First of all, the Matrix Voice microphone must be chosen to capture the sound. The next step, before starting the capture, is to adjust the threshold of the ambient noise. This is an important aspect to take into account because the noise may interfere in the recognized result. The threshold value is automatically set through the ambient noise existing at the moment of the command is executed. At last, the capture is made and sent to the chosen service to do the speech recognition. The result of the request is a string representing the speech recognition.

5.4 Sound Output System

In order to increase the interaction of the RH, a sound output system was implemented. As mentioned in Chapter 4, the robotic head has two speakers for sound output.

The Raspberry Pi used to display and control the expressions, RPiCM3, does not possess a 3.5mm output jack or a Digital-to-Analogue Converter (DAC). For this reason, it was not possible to generate audio outputs without additional hardware. However, the RPi3 used for the Speech Recognition module possesses a 3.5mm output jack, for this reason this Raspberry Pi was used to generate audio output.

There was the need to use a sound amplifier, since the amplitude of the sound projected by the speakers was too low. Since we have only one sound output and two speakers, there was also the need to replicate the output signal. To solve both problems, an audio amplifier (Adafruit ST 2012 audio amp) [4] was used. The amplifier has a differential input allowing thus to have one or two input signals and has two equals output signals. The gain of the amplifier output signal can be chosen by the user between 6dB, 12dB, 18dB and 24db. Figure 5.4 shows how the sound output was made.

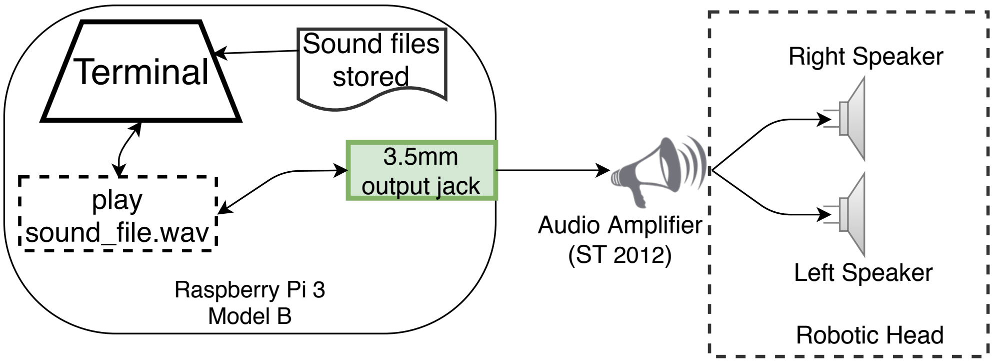


Fig. 5.4 Sound output system overview.

Chapter 6

ISR-RobotHead Modules Integration

This chapter describes the integration of all five modules that compose ISR-RobotHead: eye expression, mouth expression, facial color, speech recognition and sound output.

6.1 Expression Display Control

As shown in Fig. 6.1, in order to integrate all modules to display emotional expressions, a four-thread software architecture was implemented on the RPiCM3: three to communicate with LCDs (each one for each LCD) and one to control which emotional expression should be displayed.

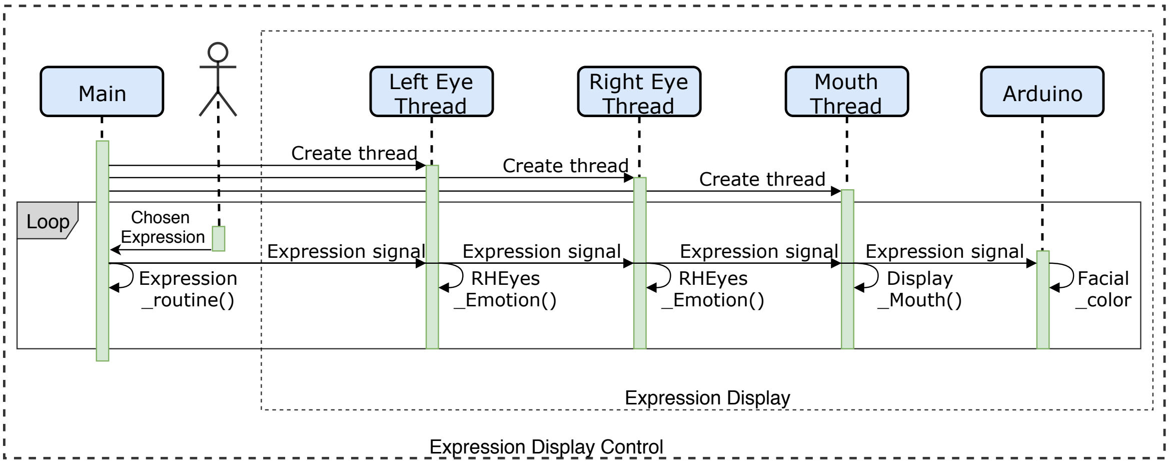


Fig. 6.1 The four-thread software architecture to display emotional expressions.

As illustrated in Fig. 6.1, when the user chooses an expression to be displayed, the main thread will send a signal to the other threads and for the Arduino at the same time, in order to establish a synchronization between all the modules involved described in Chapter 4 for displaying the emotional expression.

6.2 Sound System Interaction with Expression Display Control

The expression display control system is implemented in a RPiCM3 and the sound system is implemented in a RPi3, in other words, the two systems are implemented in two different devices. So, in order to integrate the two systems, a TCP/IP (Transmission Control Protocol/Internet Protocol) connection between the two devices was made, so that the expression display control system may emit sounds associated with expressions and to be able to received voice commands. For this connection, an additional thread was implemented in the RPiCM3. In the RPi3, a two-thread software architecture was implemented: one for establish the connection with RPiCM3 and emit sounds and the other thread to perform speech recognition module. Figure 6.2 shows the software architecture implemented in order to establish the connection between the two devices and perform the speech recognition module.

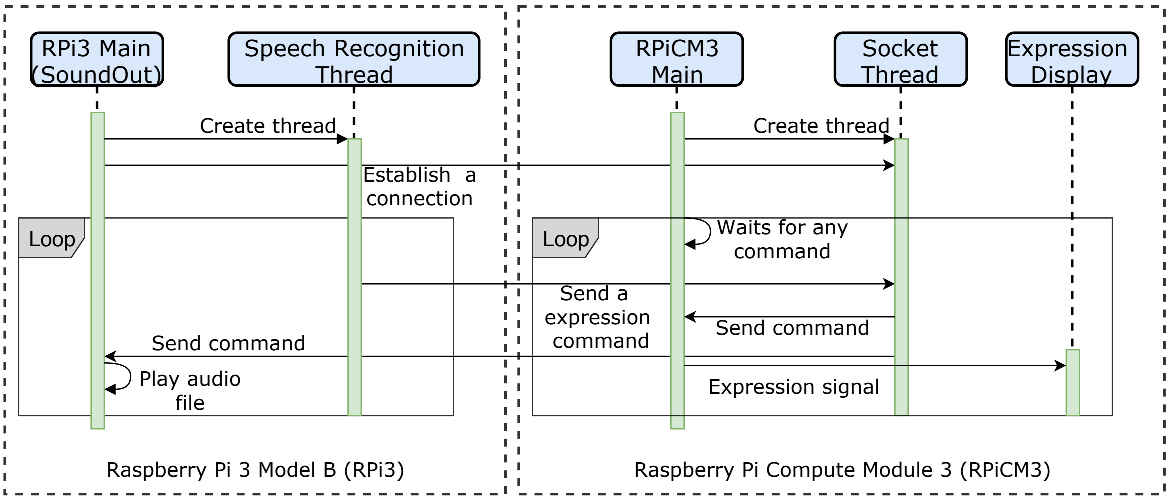


Fig. 6.2 Software architecture for the connection between the two devices.

As illustrated in Fig. 6.2, after the connection between the two devices is established, the expression display control system will wait for an expression command. When a voice command is recognized, a signal is sent to the other Raspberry Pi in order that emotional expression can be displayed and, if applicable, a sound can be emitted.

6.2.1 Voice Commands

In order to minimize the number of requests made to the speech recognition software systems, since some systems limit the number of requests, a trigger activated by the word "okay" has been implemented. The trigger is implemented using one of the following two non-commercial speech recognition software systems: CMU Sphinx or Wit.ai. When the word "okay" is recognized, the system will wait for a voice command. At this stage, any of the software systems can be used. If the word recognized corresponds to a voice commands predefined, a message is sent to the RPiCM3. Table 6.1 shows the main voice commands used for the trigger and expressions. Figure 6.3 shows the interconnection between voice commands and expression display control.

Voice Commands			
okay	surprise or surprised	anger or angry	disgust
fear or fearful	happy or happiness	sad or sadness	-

Table 6.1 Main voice commands used.

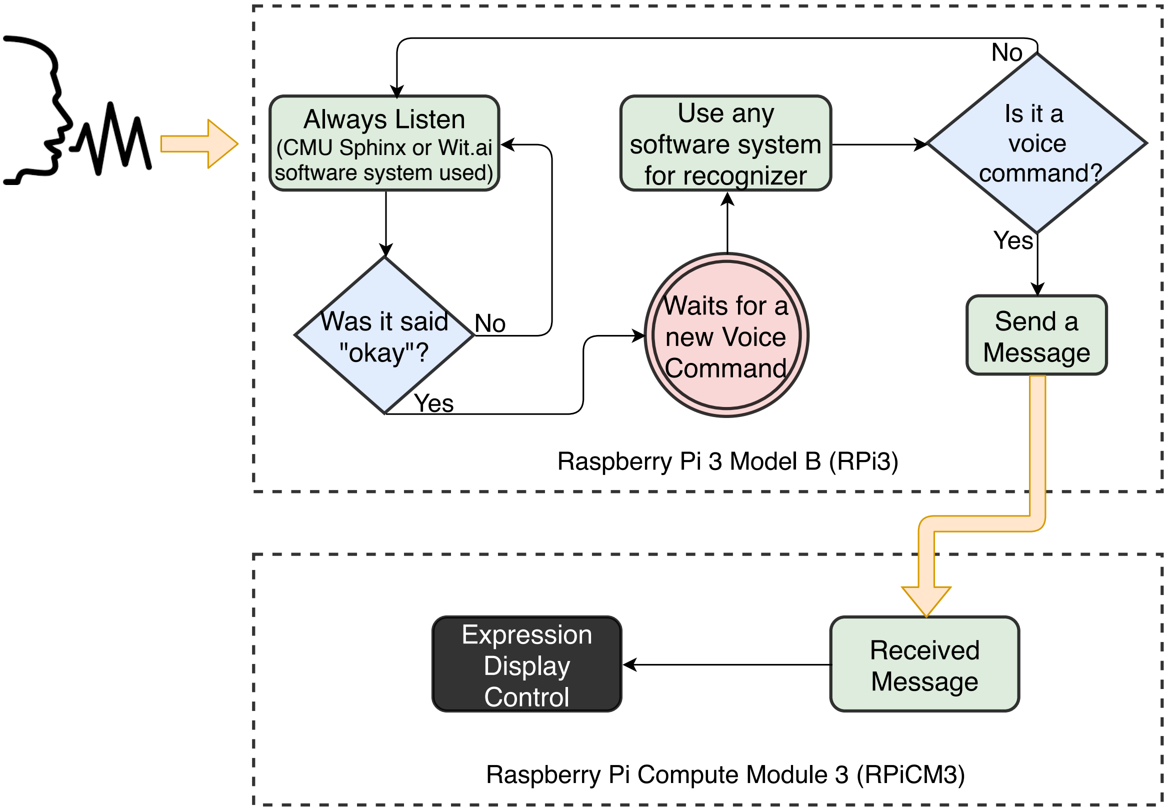

Always Listen
(CMU Sphinx or Wit.ai software system used)
Was it said "okay"?
No
Yes
Use any software system for recognizer
Waits for a new Voice Command
Is it a voice command?
No
Yes
Send a Message
Raspberry Pi 3 Model B (RPi3)
Expression Display Control
Received Message
Raspberry Pi Compute Module 3 (RPiCM3)

Fig. 6.3 Flowchart with the interconnection between voice commands and expression display control.

6.2.2 Sound Output

As described in the Section 5.4, the ISR-RobotHead is able to emit sounds through the audio files previously recorded and stored in RPi3's memory. To play the audio files, it is only necessary to call the *aplay* command available in the Raspberry Pi operating system.

6.3 ISR-RobotHead Interface

In order to be easier to configure some parameters of the ISR-RobotHead, a cross-platform interface (Android, Windows and Mac operating systems) has been developed with the following functionalities:

- to choose an emotional expression to be displayed;
- to control the transition time between two emotional expressions;
- to control the Cartesian coordinates of each eye;
- to control the gaze orientation of the eyes.

The cross-platform *QT Creator* [11] IDE (Integrated Development Environment) available to Windows, Linux and Mac operating systems was used to develop the interface. Figure 6.4 shows an overview of how the interface interacts with RPiCM3.

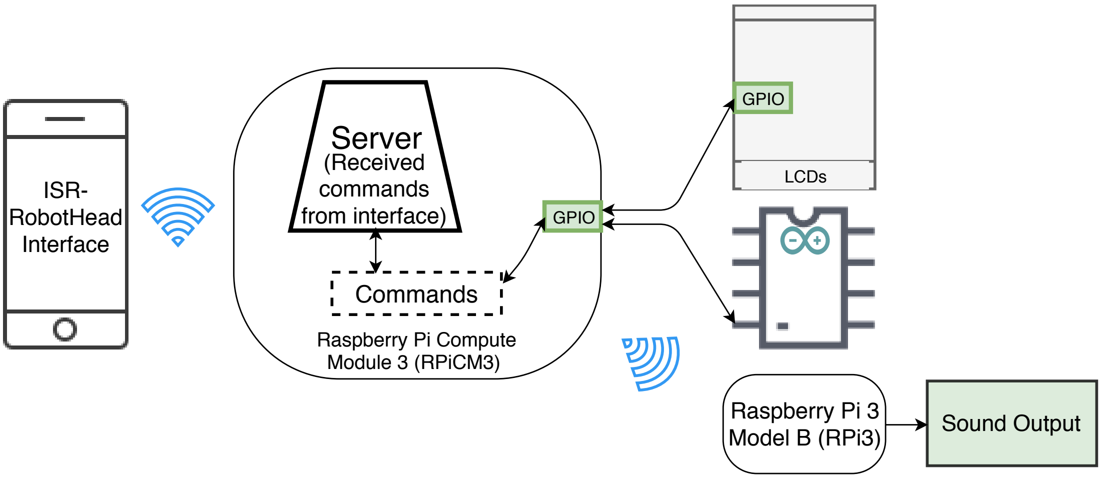

ISR-RobotHead Interface
Server (Received commands from interface)
Commands
GPIO
Raspberry Pi Compute Module 3 (RPiCM3)
GPIO
LCDs
Raspberry Pi 3 Model B (RPi3)
Sound Output

Fig. 6.4 Overview of interface interacts with RPiCM3 modules.

6.3.1 How it Works

In order for the interface to interact with the expression display control system, a TCP/IP connection with the RPiCM3 is used.

The user may interact with ISR-RobotHead through buttons, sliders and a joystick. Each time the user press a button, slider or the joystick, a message is sent to the server in RPiCM3 with the necessary information. Figure 6.5 shows an overview of the interaction of this interface with the expression display control and sound output.

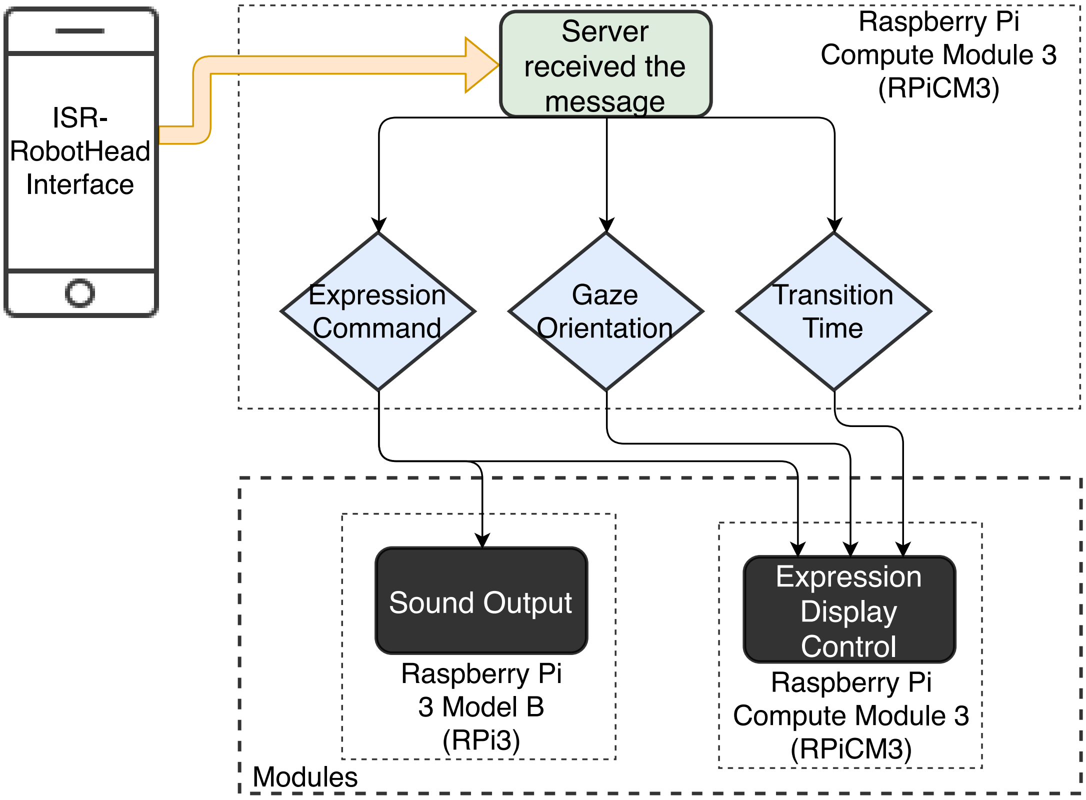
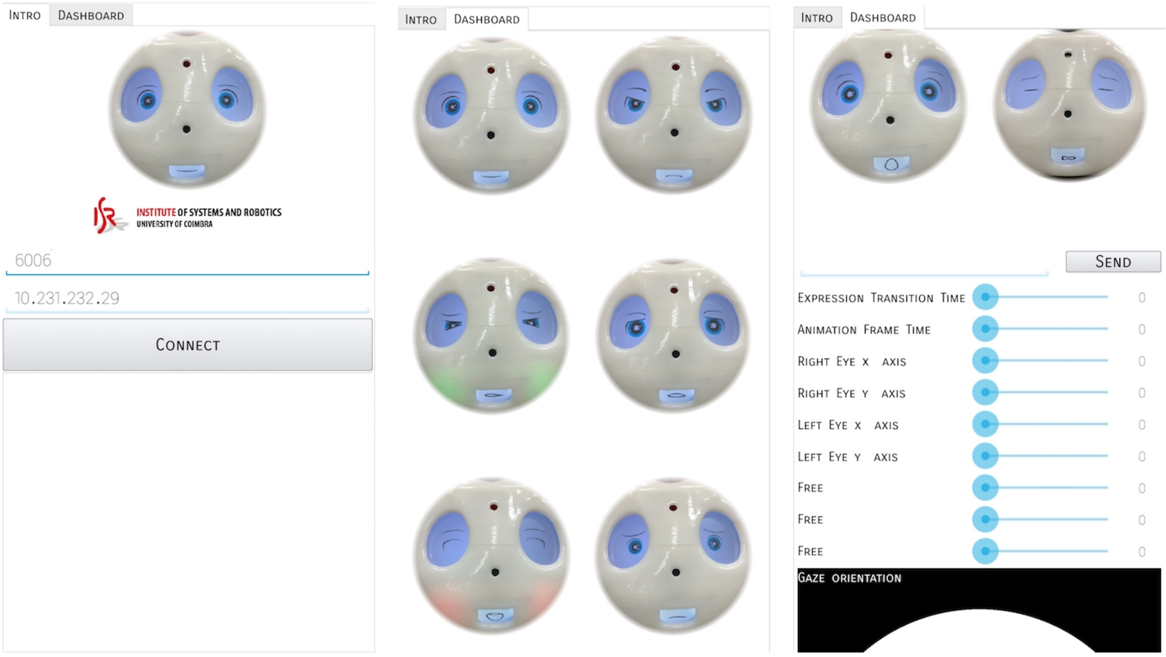

ISR-RobotHead Interface
Server received the message
Raspberry Pi Compute Module 3 (RPiCM3)
Expression Command
Gaze Orientation
Transition Time
Sound Output
Raspberry Pi 3 Model B (RPi3)
Expression Display Control
Raspberry Pi Compute Module 3 (RPiCM3)
Modules

Fig. 6.5 Overview of how modules work when a command is sent by the interface.

6.3.2 Interface Design

The interface is composed by two sections. The first one allows to define the connection to the RPiCM3. The connection is defined with the IP (Internet Protocol) of the RPiCM3 and the TCP port in which the server application is listening. After the connection is established, the second section, that allows to control the emotional expressions, is available. The control of the emotional expressions can be made using eight buttons, nine sliders and a joystick. Each one of the eight buttons represents an emotional expression. By pressing on a button, the corresponding expression is displayed by the robot head. The sliders allow to control parameters as the transition time of the emotional expressions and the Cartesian coordinates (x,y) of each eye. The joystick allows to control the orientation of the robot's gaze. Also, for test and debug purposes, a text box that allows to send directly commands to the server is also available. Figure 6.6 shows the interface developed.



6006
10.231.232.29

Expression Transition Time 0
Animation Frame Time 0
Right Eye x axis 0
Right Eye y axis 0
Left Eye x axis 0
Left Eye y axis 0
Free 0
Free 0
Free 0

(a) First window.

(b) Second window.

(c) Second window continuation

Fig. 6.6 ISR-RobotHead Interface.

Chapter 7

ISR-RobotHead's Experimental Results

This chapter is dedicated to present the main experimental results obtained from different tests. First, the new facial expressions and their main features are presented. In order to validate them, an evaluation with children was performed. Since external software systems to do the speech recognition were used, it was determined the best to be used. Finally, the verbal interaction with the robot and its response through facial expressions was evaluated.

7.1 Representation of Facial Expressions in ISR-RobotHead

The ISR-RobotHead is capable to display seven emotional states, the neutral and the major six described in Section 3.2: fear, sadness, happiness, surprise, anger and disgust. The result of the neutral or default expression can be observed in Fig.7.1. All other facial interactions always start from this expression.

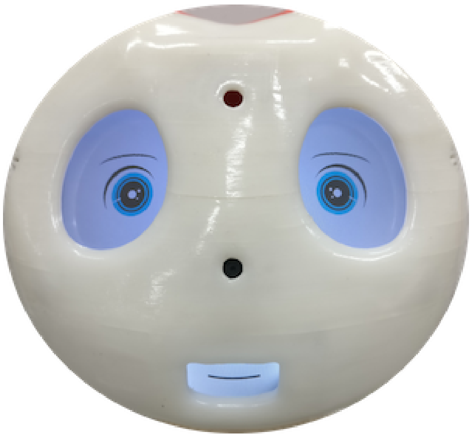
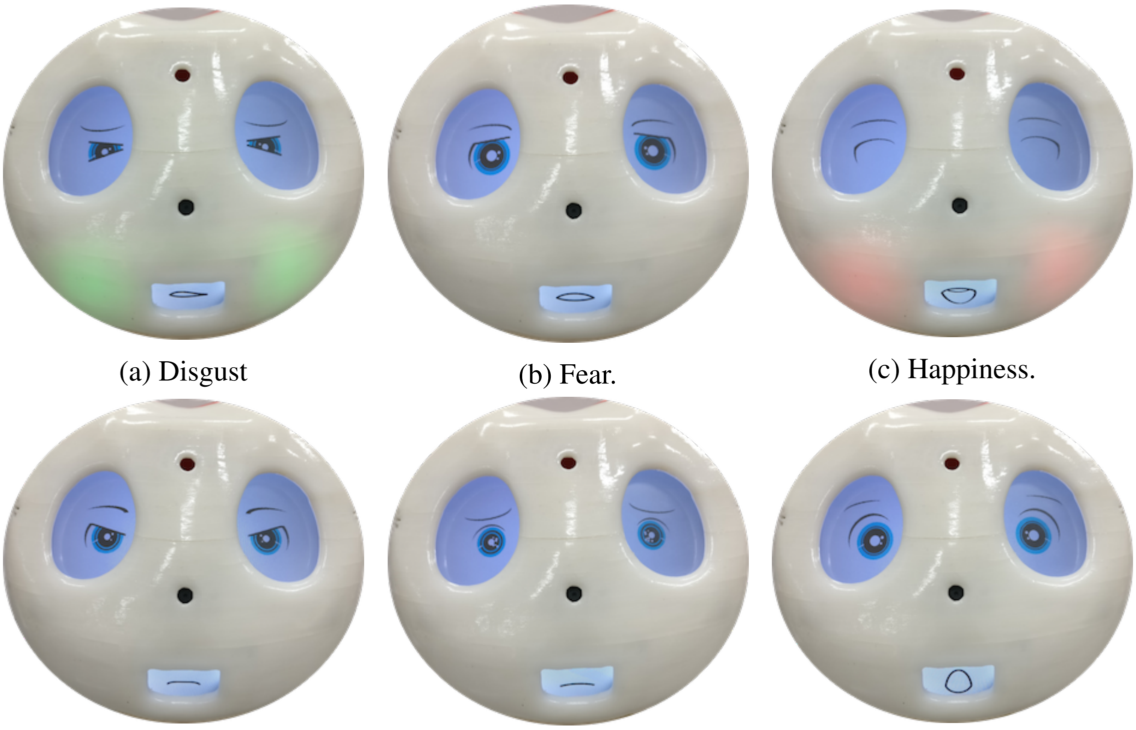


Fig. 7.1 ISR-RobotHead neutral expression.

In order to improve the facial expressions and make them more realist, animations for each expression and transition animations between two emotional states were developed. For that end, the best way to make a smooth and comfortable transition between two emotions was through Russel's circumplex described Section 3.2.1, but such implementation was not performed due the number of images necessary to contain all combinations. For that it was only developed transition animations between the neutral state to all the other expressions. As result, the major six emotional states expressed by ISR-RobotHead can be observed in Fig. 7.2.



(a) Disgust

(b) Fear.

(c) Happiness.

(d) Anger.

(e) Sadness.

(f) Surprise.

Fig. 7.2 Pictures of the ISR-RobotHead representing the new six facial expressions.

Figure 7.3 shows two examples of the images used, eyes and mouth, to create a transition animation between the neutral state to the surprise or sadness states. By default, these transitions take $\approx$ 300ms. When the neutral state is displayed, the robot is able to blink both eyes. In order to create this animation, ten images for each eye were developed and takes $\approx$ 500ms to display all images.

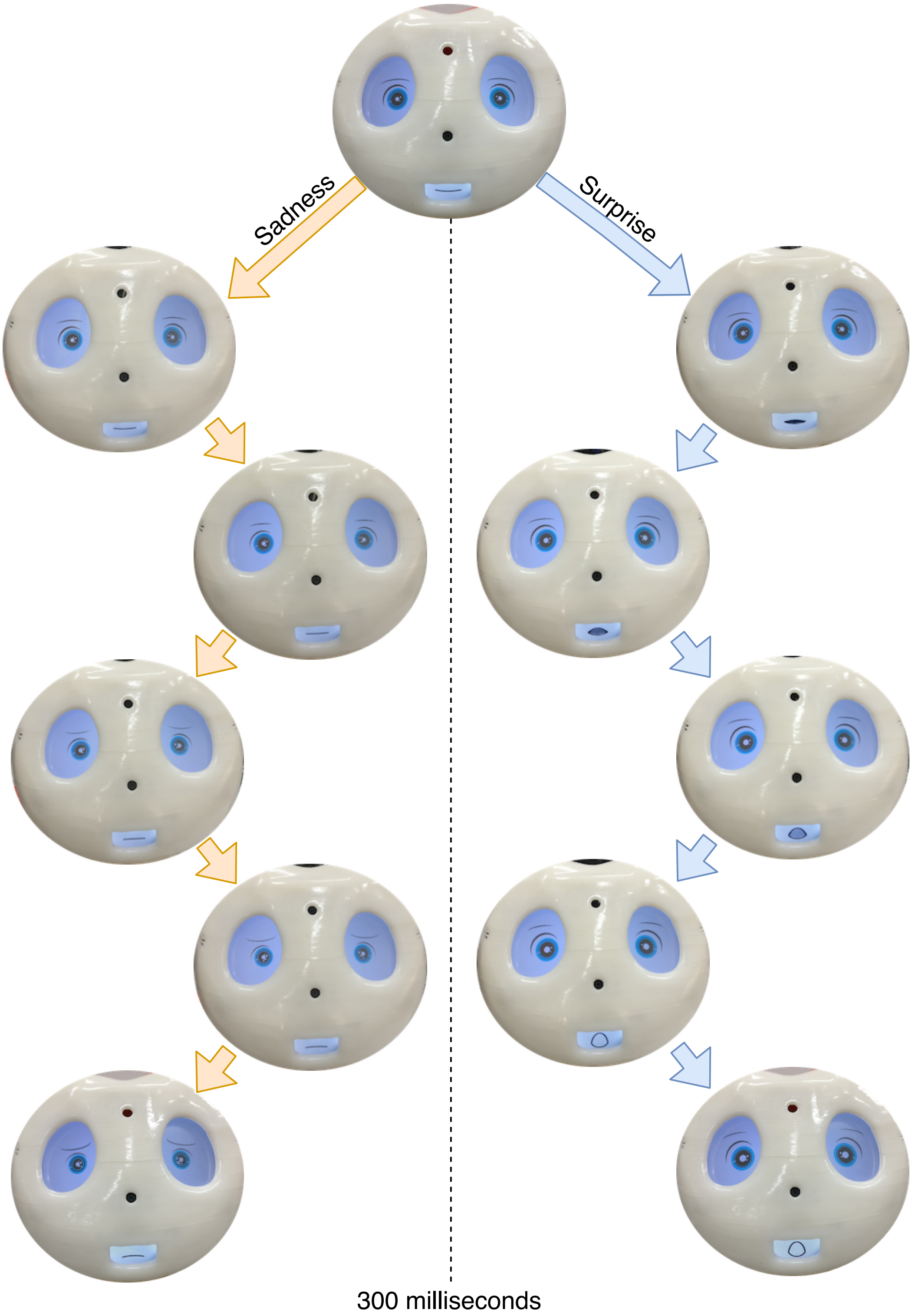
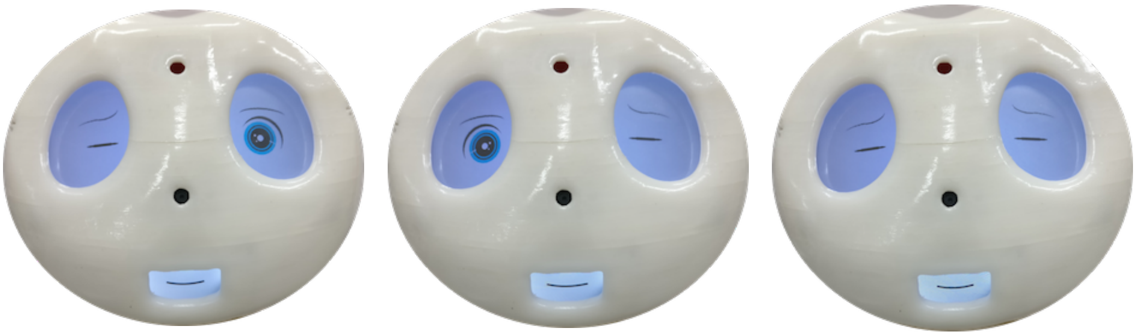


Fig. 7.3 ISR-RobotHead transition between animation frames. Normal to Sadness in the left. Normal to Surprise in the right.

Since the eye's LCDs used to display the eyes are touch-screen, this feature was used to increase the interaction with the robotic head. Each eye LCD can be touched individually or at the same time and in Fig. 7.4 are presented the responses of the robot head. When any eye LCD is touched, a sound is also emitted. After some seconds the robot returns to the neutral expression.



(a) Right eye touched. (b) Left eye touched. (c) Both eyes touched.

Fig. 7.4 Pictures of the ISR-RobotHead representing what happens when each eye LCD is touched.

7.1.1 Experimental Validation

In order to validate the new facial expressions displayed by ISR-RobotHead, a survey was performed at the preschool of CASCI ("Centro de Ação Social do Concelho de Íllhavo") with a sample of 24 children (N = 24), aged between 4 and 6 years old (experimental setup presented in Fig. 7.5). Since children were very young, implying a very restricted verbal ability, the best method for them to identify the six emotional states (disgust, fear, happiness, anger, sadness and surprise) shown by ISR-RobotHead was through a matching test between the emotion present on the robot with emotional states present in cards. Each card has one of the emotions represented in Fig. 7.6.

7.1.2 Results

As shown in Fig. 7.7, all participants correctly identified the emotional state of "Surprise" and most participants ($n = 22$) correctly detected the emotions of "Happiness" and "Sadness". However, the worst result of the matching test was obtained with the emotion of "Disgust", ($n = 16$) followed by the "Anger" ($n = 17$) and "Fear" ($n = 19$). An important aspect that should be mentioned is that a small group of children failed to match the emotion but said it verbally correct. These verbal answers were not taken into account in the results presented.

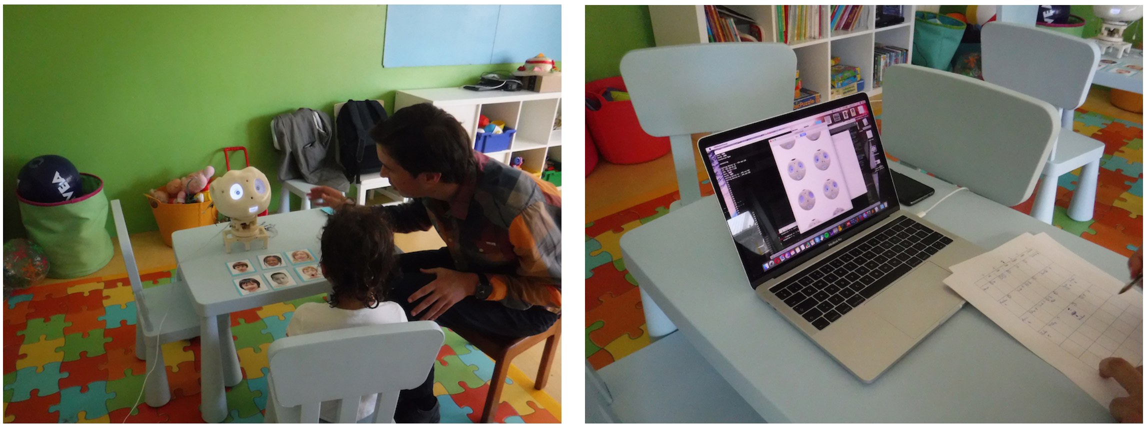
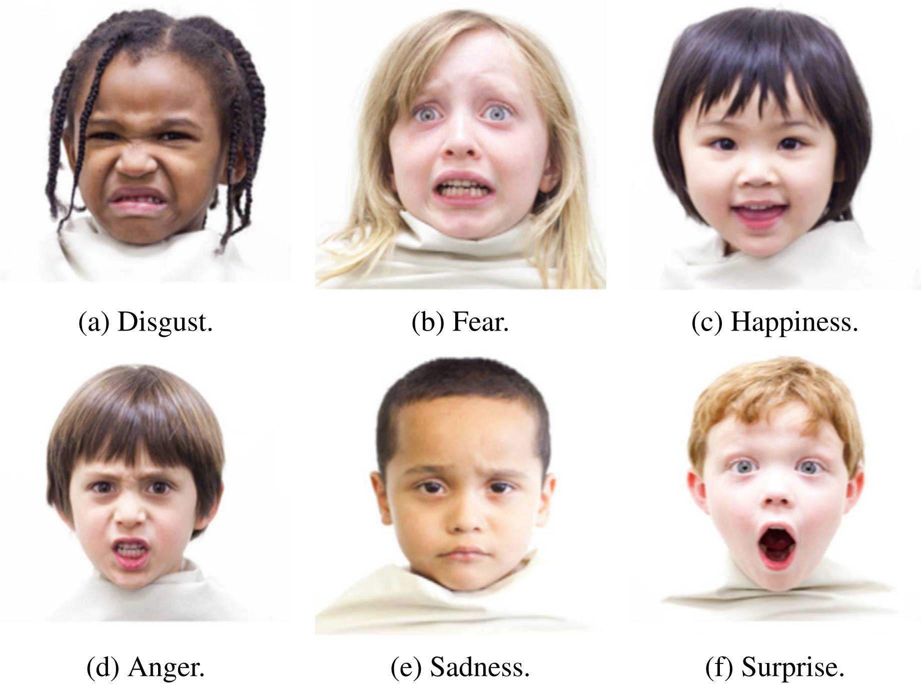


Fig. 7.5 Experimental setup in the CASCI's preschool.



(a) Disgust.

(b) Fear.

(c) Happiness.

(d) Anger.

(e) Sadness.

(f) Surprise.

Fig. 7.6 Pictures used to perform the matching test with children [30].

Comparing the results obtained with the results obtained in the previous work [32] (see Table 7.1), despite the great difference of the sample, it becomes clear that there was a great improvement in the expression "Disgust". Comparing with other robots that performed similar tests (see Table 7.1), Kismet [8] and EDDIE [45], it is possible to verify that the ISR-RobotHead's new facial emotions has a better recognition rate mean than Kismet and better recognition in all expressions relative to the EDDIE.

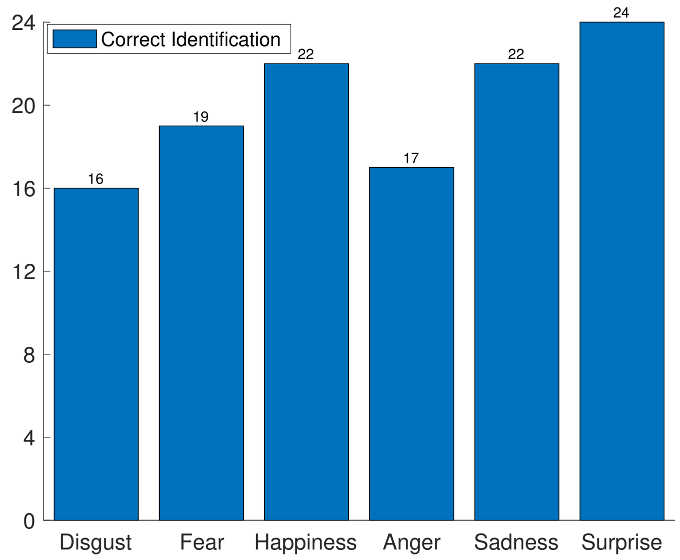


Fig. 7.7 Frequencies of correct detection of the robotic facial emotions displayed by children.

	Kismet	Eddie	ISR-RobotHead v1.0
Sample	-	24	9
Disgust	71%	58%	11%
Fear	47%	42%	67%
Happiness	82%	58%	100%
Anger	76%	54%	100%
Sadness	82%	58%	100%
Surprise	82%	75%	67%

Table 7.1 Identification rate of different robot faces in comparison.

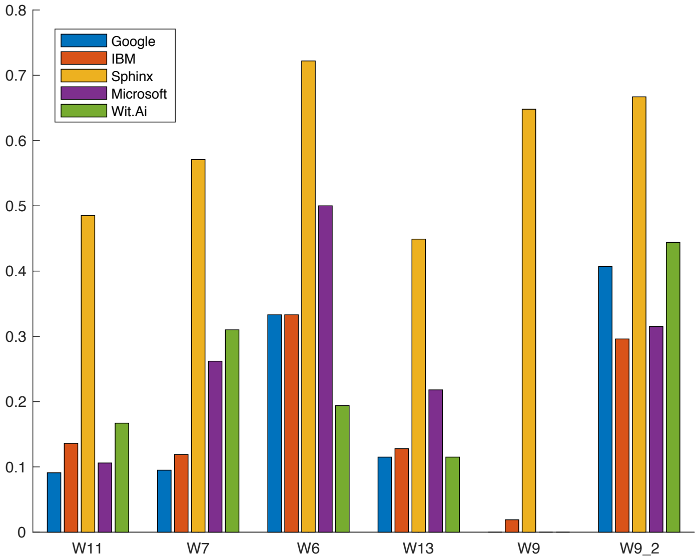
7.2 WER for the Speech Recognition Software Systems

To discover which, among the five speech recognition software systems, is the best, six phrases (see Table 7.2) spoken by six different persons were recorded in the same conditions i.e. the same background noise. The six phrases were chosen in order to contain a greater lexical and phonetic field. To record the audio files the Matrix Voice and the RPi3 were used. All the audio files were submitted to the five software systems, and each one of them returned a string with what was recognized. As shown in the right of the Fig. 7.8, the best software systems are the IBM and Google with $\approx 17\%$ WER but the Google software has the best response time with ≈ 2.2 seconds. The worst recognition and time of response is the Sphinx, but since this software system works offline in the RPi3, it was expected to take longer than other systems.

The recognition of a complex phrases or a dialogue is not the goal of this dissertation, but this test allows to understand the types of phonetics that should be avoided for the commands to be used for the expressions.

File Name	Phrases
W11	Please take this dirty table cloth to the cleaners for me.
W7	Jazz and swing fans like fast music.
W6	Who authorized the unlimited expense account?
W13	The fish began to leap frantically on the surface of the small lake.
W9	The quick brown fox jumps over the lazy dog.
W9_2	His hip struck the knee of the next player.

Table 7.2 Phrases used to submit in the speech recognition services [25].



	WER (average)	Time (average)
Google	0.174	2.261s
IBM	0.172	4.125s
Sphinx	0.590	13.938s
Microsoft	0.233	3.144s
Wit.ai	0.205	3.936s

Fig. 7.8 WER obtained. On the left WER is presented the average by phrase. On the right is the total average.

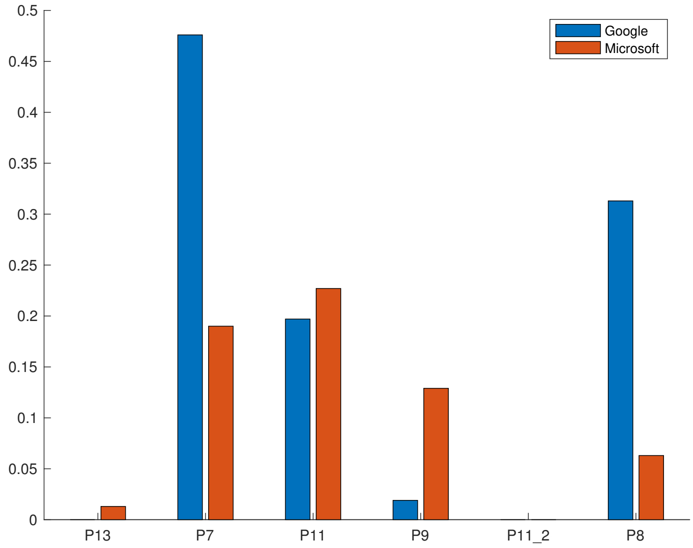
With these results it has become clear that the best software system to activate the trigger by the word "okay", mentioned in Section 6.2.1 is Wit.ai. To reinforce this choice Wit.ai and Sphinx systems were used to recognize forty times the word "okay" with different background noise, such as, a room without noise, with some noise and very noisy. The accuracy of the word recognized without noise was 90-100% in both services, but the increase of noise became a problem for the Sphinx. With a little of noise the accuracy of the Wit.ai got down to $\approx$ 80-90% while the Sphinx got to $\approx$ 40-50%. In the very noisy room the accuracy of the Wit.ai got down to $\approx$ 20-30% while the Sphinx was unable to recognize any word. With this, Wit.ai is the best choice with $\approx$ 70% on average of word recognition and the Sphinx $\approx$ 50%.

7.2.1 Portuguese Language

Of the software systems used to do speech recognition there are two that have been trained for the Portuguese language, Google and Microsoft. So, to check if this language could be used, six phrases were chosen (see Table 7.3) and spoken by six different persons and were submitted to the recognition systems. As shown in the right of the Fig. 7.9, Microsoft speech recognition was the best with $\approx 10\%$ WER. It should be noted that for the Portuguese recognition, both systems have a WER and response time lower than the recognition performed in English. This can happen because the most appropriate phrases have not been chosen, but once again, recognizing a dialogue is not a goal and this test allowed to understand the types of phonetics that should be avoided in Portuguese.

File Name	Phrases
P13	Por favor, leve esta toalha de mesa suja para a lavandaria por mim.
P7	Quem autorizou uma conta de despesas ilimitada?
P11	O peixe começou a saltar freneticamente na superfície do pequeno lago.
P9	A raposa castanha salta rapidamente sobre o cão preguiçoso.
P11_2	O rato roeu a rolha da garrafa do rei da Rússia.
P8	O Pedro pregou um prego na porta preta.

Table 7.3 Phrases in Portuguese used to submit in the speech recognition services.



	WER (average)	Time (average)
Google	0.167	1.506s
Microsoft	0.104	2.769s

Fig. 7.9 WER obtained for Portuguese language recognition. On the left WER is presented the average by phrase. On the right is the total average.

7.3 Expression Recognition through Voice Recognition

In order to test the expressions commands through voice, a sample of 7 adults, aged between 23 and 24 years old, were presented to the ISR-RobotHead and asked to identify the six emotional states. In each test 18 expressions were randomly displayed, using the random function available in *c* programming language, which would give a mean of 3 times for each emotional state and each person had to verbally say which emotional state was being represented. In each expression, a sound file was recorded with the verbal expression said by the person, e.g. sad or sadness, joy or happiness, disgust or disgusted, surprise or surprised, anger ou angry, fear ou fearful, among many others. Each sound file were submitted to the five systems in order to recognize the contents in the file. Table 7.4 shows the number of expression well recognized and Table 7.5 shows the number of well succeeded recognitions by each speech recognition software system. It should be emphasized that all participants correctly identified the emotions of "Happiness", "Anger" and "Surprise. These expression commands were also the most easily recognized by the speech recognition software systems used. The expression "sad" was the most difficult to recognize by the systems used, where the word was commonly misinterpreted with phonetically similar words such as "said".

Generally, the emotional states were well recognized and the Microsoft speech recognition is the best service to recognize the verbal expressions commands.

	Emotions					
	Disgust	Fear	Happiness	Anger	Sadness	Surprise
Correct answers	4	5	7	7	5	7

Table 7.4 Number of facial expression well recognized by adults.

	Emotions						
	Disgust	Fear	Happiness	Anger	Sadness	Surprise	Sum
Appeared	22	22	18	28	14	22	126
Google	8	5	16	22	4	21	76
IBM	0	7	13	26	1	21	68
Microsoft	11	9	17	27	4	16	84
Sphinx	1	3	2	11	0	14	31
Wit.Ai	6	6	13	22	3	21	71

Table 7.5 Number of correct verbal acknowledgments through speech recognition software systems.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

In this dissertation, the core of the work was mainly focused on improving the ISR-RobotHead's Human-Robot Interaction. A study of different types of robots capable to do a non-verbal communicate was conducted, leading to the decision to rebuild the software architecture in order to be possible to increase the interaction with humans.

With this work, in addition to having a new architecture implemented, the robotic head hardware has been improved with the replacement of an outdated Raspberry Pi by a newer one and with the addition of two LEDs, a Matrix Voice board and a conventional Raspberry Pi.

At the moment, the ISR-RobotHead is able to display six facial emotional expressions, respond when the eye's LCDs are touched and the facial expression can be controlled by voice commands and with an interface. The results from an experiment with children prove that this robot is capable to interact with them and the expressions were well interpreted. Although they were well recognized, it is always possible to improve the expressions and even increase their number.

In order to choose which expression to display, the implementation of the speech recognition software systems on the robot became a good way to interact with humans. In general, with the Word Error Rate (WER) study and the results obtained from facial expression commands, all speech recognition software systems can be used with the exception of Sphinx. With this new software architecture it is easy to increase the modes of interaction and to choose which expression to display, as long as the protocol used to communicate between the modules is respected.

8.2 Future Work

From all sensors available on the ISR-RobotHead, only the cameras were not used. So, in future implementations these cameras can be used for face and facial emotion recognition.

The robotic head needs a pan-tilt system which would allow it to move around and have the possibility to perform facial and sound tracking.

To add further insight, it would be important to do experiments with children that have autism spectrum disorder since they have more difficulty to recognize emotions, accompanied by studies in the field of psychology.

References

[1] 4DSystems. https://www.4dsystems.com.au/product/ulcd_32ptu/. [Online; accessed February 20, 2018].

[2] Abrosoft. Morphing Software Program. http://www.fantamorph.com. [Online; accessed April 14, 2018].

[3] Adafruit. NeoPixel Library. https://github.com/adafruit/adafruit_neopixel. [Online; accessed June 20, 2018].

[4] Adafruit. Stereo Audio Amplifier. https://learn.adafruit.com/adafruit-ts2012-2-8w-stereo-audio-amplifier/overview. [Online; accessed June 30, 2018].

[5] Rajesh Kumar Aggarwal and M. Dave. Acoustic modeling problem for Automatic Speech Recognition system: Advances and refinements (Part II). *International Journal of Speech Technology*, 14(4):309–320, 2011.

[6] Ho Seok Ahn, Pyo Jae Kim, Jeong Hwan Choi, Shamyl Bin Mansoor, Woo-Sung Kang, Seok Min Yoon, Jin Hee Na, Young Min Baek, Hyung Jin Chang, Dong Sung Song, Jin Young Choi, and Hyeong-Seok Ko. Emotional Head Robot with behavior decision model and Face Recognition. In *2007 International Conference on Control, Automation and Systems*, pages 2719–2724, Oct 2007.

[7] L.R. Bahl, P.F. Brown, P.V. De Souza, and R.L. Mercer. A tree-based statistical language model for natural language speech\nrecognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(7), 1989.

[8] C. Breazeal and B. Scassellati. How to build robots that make friends and influence people. In *Proceedings 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human and Environment Friendly Robots with High Intelligence and Emotional Quotients (Cat. No.99CH36289)*, volume 2, pages 858–863 vol.2, Oct 1999.

[9] Cynthia Breazeal. Socially Intelligent Robots. *Interactions*, 12(2):19–22, March 2005.

[10] Eric Brown. Matrix Voice RPi add-on with FPGA-driven mic array relaunches. http://linuxgizmos.com/matrix-voice-rpi-add-on-with-7-mic-array-relaunches/. [Online; accessed May 19, 2018].

[11] QT Company. Software Development IDE. https://www.qt.io/qt-features-libraries-apis-tools-and-ide/. [Online; accessed June 2, 2018].

[12] Andréia Cristina Peres Rodrigues da Costa and Georges Steffgen. Socially assistive robots for teaching emotional abilities to children with Autism Spectrum Disorder. 2017.

[13] D David, S. A. Matu, and O. David. Robot-Based Psychotherapy: Concepts Development, State of the Art, and New Directions. 7:192–210, 06 2014.

[14] Mohit Dua, R K Aggarwal, Virender Kadyan, and Shelza Dua. Punjabi Automatic Speech Recognition Using HTK. *International Journal of Computer Science Issues*, 9(4):359–364, 2012.

[15] Paul Ekman. An argument for basic emotions. *Cognition and Emotion*, 6(3-4):169–200, 1992.

[16] Michael Finke, Petra Geutner, Hermann Hild, Thomas Kemp, Klaus Ries, and Martin Westphal. The Karlsruhe-Verbmobil Speech Recognition Engine. *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, 1:83–86, 1997.

[17] Raspberry Pi Foundation. https://www.raspberrypi.org/products/compute-module-io-board-v3/. [Online; accessed February 18, 2018].

[18] Raspberry Pi fundation. Raspberry Pi webpage https://www.raspberrypi.org/products/raspberry-pi-3-model-b/. [Online; accessed June 2, 2018].

[19] Santosh K. Gaikwad, Bharti W. Gawali, and Pravin Yannawar. A Review on Speech Recognition Technique. *International Journal of Computer Applications*, 2010.

[20] A. Ganapathiraju, J. Hamaker, and J. Picone. Applications of Support Vector Machines to Speech Recognition. *IEEE Transactions on Signal Processing*, 52(8):2348–2355, 2004.

[21] Gordon Henderson. GPIO interface library for the Raspberry Pi. http://wiringpi.com. [Online; accessed February 27, 2018].

[22] Hynek Hermansky. Perceptual linear predictive (PLP) analysis of speech. *The Journal of the Acoustical Society of America*, 87(4):1738–1752, 1990.

[23] ITEADStudio. https://www.itead.cc/display/tft-lcm/itdb02-1-8sp.html. [Online; accessed February 22, 2018].

[24] Jan Kędzierski, Robert Muszyński, Carsten Zoll, Adam Oleksy, and Mirela Frontkiewicz. EMYS—Emotive Head of a Social Robot. *International Journal of Social Robotics*, 5(2):237–249, Apr 2013.

[25] Veton Këpuska. Comparing Speech Recognition Systems (Microsoft API, Google API And CMU Sphinx). *International Journal of Engineering Research and Applications*, 2017.

[26] T. Klein, G. J. Gelderblom, L. de Witte, and S. Vanstipelen. Evaluation of short term effects of the IROMEC robotic toy for children with developmental disabilities. In *2011 IEEE International Conference on Rehabilitation Robotics*, pages 1–5, June 2011.

[27] Paul Lamere, Philip Kwok, Evandro Gouvea, Bhiksha Raj, Rita Singh, William Walker, Manfred Warmuth, and Peter Wolf. The CMU SPHINX-4 speech recognition system. *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP 2003), Hong Kong*, 2003.

[28] A. V. Libin and E. V. Libin. Person-robot interactions from the robopsychologists' point of view: the robotic psychology and robotherapy approach. *Proceedings of the IEEE*, 92(11):1789–1803, Nov 2004.

[29] Karen Livescu, James R Glass, and Jeff A Bilmes. Hidden feature models for Speech Recognition using dynamic Bayesian networks. *Interspeech*, (1):2529–2532, 2003.

[30] Vanessa LoBue and Cat Thrasher. The Child Affective Facial Expression (CAFE) set: validity and reliability from untrained adults. *Frontiers in Psychology*, 5:1532, 2015.

[31] Ricardo Loureiro. *Development of a Robotic Head to Express Human Emotions*. Master degree Dissertation, Coimbra University, 2017.

[32] Ricardo Loureiro, Andre Lopes, Carlos Carona, Daniel Almeida, Fernanda Faria, Luís Garrote, Cristiano Premebida, and Urbano J. Nunes. ISR-RobotHead: Robotic head with LCD-based emotional expressiveness. *ENBENG 2017 - 5th Portuguese Meeting on Bioengineering, Proceedings*, pages 1–4, 2017.

[33] I. Lütkebohle, F. Hegel, S. Schulz, M. Hackel, B. Wrede, S. Wachsmuth, and G. Sagerer. The bielefeld anthropomorphic robot head Flobi. In *2010 IEEE International Conference on Robotics and Automation*, pages 3384–3391, May 2010.

[34] Matrix. Matrix Voice examples. https://matrix-io.github.io/matrix-documentation/matrix-hal/examples/. [Online; accessed June 5, 2018].

[35] N. Morgan and H. Bourlard. Continuous Speech Recognition. *IEEE Signal Processing Magazine*, 12(3):24–42, May 1995.

[36] M. Mori, K. F. MacDorman, and N. Kageki. The Uncanny Valley [From the Field]. *IEEE Robotics Automation Magazine*, 19(2):98–100, June 2012.

[37] Douglas O'Shaughnessy. Interacting with computers by voice: Automatic speech recognition and synthesis. *Proceedings of the IEEE*, 91(9):1272–1305, 2003.

[38] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely. The Kaldi Speech Recognition toolkit. *IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 1–4, 2011.

[39] James Russell. A Circumplex Model of Affect. 39:1161–1178, 12 1980.

[40] Preeti Saini and Parneet Kaur. Automatic Speech Recognition: A Review. *International Journal of Engineering Trends and Technology*, 2013.

[41] R.T. Salaja, R. Flynn, and M Russell. Alife-based classifier for Automatic Speech Recognition. *Applied Mechanics and Materials*, 679(June):189–193, 2014.

[42] Hassan Satori and Fatima Elhaoussi. Investigation Amazigh Speech Recognition using cmu tools. *International Journal of Speech Technology*, 2014.

[43] Karen Schmidt and Jeffrey Cohn. Human facial expressions as adaptations: Evolutionary questions in facial expression. 116:3 – 24, 01 2001.

[44] Jun'ichiro Seyama and Ruth S. Nagayama. The Uncanny Valley: Effect of Realism on the Impression of Artificial Human Faces. *Presence: Teleoperators and Virtual Environments*, 16(4):337–351, 2007.

[45] S. Sosnowski, A. Bittermann, K. Kuhnlenz, and M. Buss. Design and Evaluation of Emotion-Display EDDIE. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3113–3118, Oct 2006.

[46] IBM speech recognition. https://www.ibm.com/watson/services/speech-to-text/. [Online; accessed June 3, 2018].

[47] Google speech recognition API. https://cloud.google.com/speech-to-text/. [Online; accessed June 3, 2018].

[48] Microsoft Azure speech recognition API. https://azure.microsoft.com/en-us/services/cognitive-services/speech-to-text/. [Online; accessed June 3, 2018].

[49] Baxter Robot webpage. https://www.rethinkrobotics.com/baxter/. [Online; accessed February 15, 2018].

[50] Buddy Robot webpage. http://www.bluefrogrobotics.com/en/buddy/. [Online; accessed February 20, 2018].

[51] Cozmo Robot webpage. https://www.anki.com/en-us/cozmo. [Online; accessed March 18, 2018].

[52] Matrix Voice webpage. https://www.matrix.one/products/voice. [Online; accessed June 5, 2018].

[53] Steffen Wittig and Matthias Rätsch. Animation of Parameterized Facial Expressions for Collaborative Robots.

[54] G. Wolberg. Recent Advances in Image Morphing. In *Proceedings of CG International '96*, pages 64–71, June 1996.

[55] Anthony Zhang. Speech Recognition Library. https://github.com/uberi/speech_recognition. [Online; accessed May 28, 2018].

Appendix A

Sequence of Commands for the Expressions of the Eyes.

```
DrawFilledEllipse(x−20,y−80,r+10,r−20,black, eye_port)
DrawFilledEllipse(x−10,y−78,r+20,r−21,white, eye_port)

DrawFilledCircle(x−10,y−10,r+7,black, eye_port)
DrawFilledCircle(x−6,y−3,r+8,white, eye_port)

DrawFilledCircle(x,y,r+5,eyeColor, eye_port)
DrawFilledCircle(x,y,r−15,black, eye_port)
DrawFilledCircle(x,y,r−18,eyeColor, eye_port)
DrawFilledCircle(x,y,r−20,black, eye_port)
DrawFilledCircle(x,y,r−38,white, eye_port)
DrawFilledCircle(x+15,y+8,2,white, eye_port)
DrawFilledCircle(x+18,y−6,3,white, eye_port)
DrawFilledPolygon(Xarray_points, Yarray_points, white, eye_port)
```

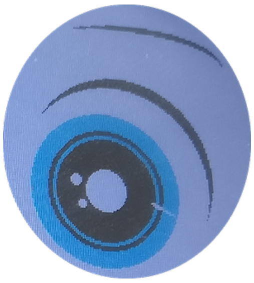


Fig. A.1 Surprise eye expression. Sequence of commands used (left) and image result (right).

```
DrawFilledEllipse(x−10,y−55,r+20,r−20,black, eye_port)
DrawFilledEllipse(x−10,y−53,r+22,r−18,white, eye_port)

DrawFilledCircle(x−5,y−5,r+12,black, eye_port)
DrawFilledCircle(x,y+2,r+12,white, eye_port)
DrawFilledEllipse(x−5,y−25,r+7,r−5,white, eye_port)
DrawFilledEllipse(x−7,y−9,r+5,r−20,black, eye_port)
DrawFilledEllipse(x−3,y−3,r+4,r−17,white, eye_port)
```

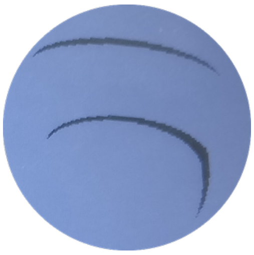


Fig. A.2 Happiness eye expression. Sequence of commands used (left) and image result (right).

```
DrawFilledEllipse(x−1,y−70,r+20,r−20,black, eye_port)
DrawFilledEllipse(x+3,y−73,r+24,r−19,white, eye_port)

DrawFilledCircle(x,y,r,eyeColor, eye_port)
DrawFilledCircle(x,y,r−8,black, eye_port)
DrawFilledCircle(x,y,r−12,eyeColor, eye_port)
DrawFilledCircle(x,y,r−15,black, eye_port)
DrawFilledCircle(x,y,r−33,white, eye_port)
DrawFilledCircle(x+10,y+8,1,white, eye_port)
DrawFilledCircle(x+13,y−6,2,white, eye_port)
DrawFilledPolygon(Xarray_points, Yarray_points, white, eye_port)

DrawFilledPolygon(Xarray_points, Yarray_points, black, eye_port)
DrawFilledPolygon(Xarray_points, Yarray_points, white, eye_port)
DrawFilledPolygon(Xarray_points, Yarray_points, black, eye_port)
DrawFilledPolygon(Xarray_points, Yarray_points, white, eye_port)
```

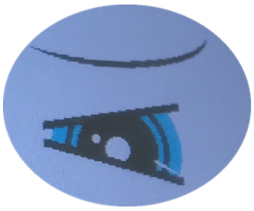


Fig. A.3 Disgust eye expression. Sequence of commands used (left) and image result (right).

```
DrawFilledEllipse(x-10,y-75,r+20,r-20,black, eye_port)
DrawFilledEllipse(x-14,y-79,r+27,r-22,white, eye_port)

DrawFilledCircle(x-5,y-2,r+12,black, eye_port)
DrawFilledCircle(x+1,y+5,r+13,white, eye_port)

DrawFilledCircle(x,y,r+5,eyeColor, eye_port)
DrawFilledCircle(x,y,r-5,black, eye_port)
DrawFilledCircle(x,y,r-8,eyeColor, eye_port)
DrawFilledCircle(x,y,r-10,black, eye_port)
DrawFilledCircle(x,y,r-28,white, eye_port)
DrawFilledCircle(x+15,y+8,2,white, eye_port)
DrawFilledCircle(x+18,y-6,3,white, eye_port)
DrawFilledPolygon(Xarray_points, Yarray_points, white, eye_port)

DrawFilledRectangle(x-50,y-55,200,140,white, eye_port)
DrawFilledPolygon(Xarray_points, Yarray_points, black, eye_port)
DrawFilledPolygon(Xarray_points, Yarray_points, white, eye_port)
```

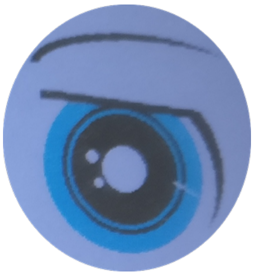


Fig. A.4 Fear eye expression. Sequence of commands used (left) and image result (right).

```
DrawFilledEllipse(x-10,y-75,r+20,r-20,black, eye_port)
DrawFilledEllipse(x-12,y-78,r+22,r-21,white, eye_port)

DrawFilledCircle(x-3,y+3,r+5,black, eye_port)
DrawFilledEllipse(x-2,y+7,r+7,r+5,white, eye_port)

DrawFilledCircle(x,y,r-5,eyeColor, eye_port)
DrawFilledCircle(x,y,r-12,black, eye_port)
DrawFilledCircle(x,y,r-17,eyeColor, eye_port)
DrawFilledCircle(x,y,r-20,black, eye_port)
DrawFilledCircle(x,y,r-33,white, eye_port)
DrawFilledCircle(x+10,y+8,1,white, eye_port)
DrawFilledCircle(x+13,y-8,2,white, eye_port)
DrawFilledCircle(x+5,y-10,1,white, eye_port)
DrawFilledCircle(x+5,y+8,2,white, eye_port)
DrawFilledCircle(x+13,y-2,2,white, eye_port)
DrawFilledCircle(x,y+14,2,white, eye_port)
DrawFilledCircle(x+15,y+12,2,white, eye_port)
DrawFilledPolygon(Xarray_points, Yarray_points, white, eye_port)
```

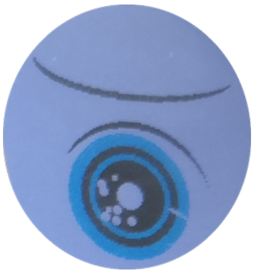


Fig. A.5 Sadness eye expression. Sequence of commands used (left) and image result (right).