

Cognitive Interaction with Robots Project

Emotive Robot Companion: Ada

Antoine Caytan
Augustin Lambert
David Dueñas



January 31, 2021

Professors : Cecilio Angulo Bahón, Anaía Garrell Zulueta

Contents

1	Description	3
1.1	Objective	3
1.2	State of the art	3
1.3	Ada’s Skills	4
1.4	Target Profile	4
2	System Requirements	5
2.1	Hardware Requirements	5
2.2	Software Requirements	6
2.2.1	Cognition Requirements	7
2.2.2	Behavior Requirements	10
2.2.3	Communication between the Nano and Otto’s chip	10
2.3	Functional & Non Functional Requirements	11
3	Evaluation	12
3.1	Research Question	12
3.2	Evaluation Procedure	12
3.2.1	Experimental Setting	12
3.2.2	Metrics	13
3.2.3	Methods	13
3.2.4	Biases	13
3.3	Results	14
3.4	Discussion & Interpretation	15
4	Conclusion	15
	References	17
	Annexes	17

Abstract

According to Matt Simon[1], “a companion robot is A robot created for the purposes of creating real or apparent companionship for human beings”. This is exactly the purpose of this project, named *Ada*. Most existing robots propose to help children or the elderly, either to learn things or to fight loneliness. Ours is different. In this first phase the objective is purely entertainment, and its target is an intermediate age group, between 15 and 40. The robot we developed is capable of responding by movement to simple words from the users, using two separate aspects: First, a speech recognition solution, using *Edge Impulse*[2] on an Arduino Nano 33 BLE Sense. The second is to use an Nano ATmega328 chip to control a physical robot, known under the name of *Otto*, able to react accordingly to what was understood by the audible event.

More generally, this paper gathers all that is needed to understand and manipulate the prototype. The words to be spoken to trigger reactions, but also the sources of inspiration and the components of the robot. The technical aspects of its functioning are some of the motives in this document, but not only ones; the creation of the robot is an important point, but so is the evaluation, to see if it works as expected. Therefore, the last pages to be read here are therefore devoted to this, with the aim of having a study as complete as possible concerning the cognitive and interactive capacities of this robot.

1 Description

In this section, the purpose of Ada design is given. Then the idea is detailed, starting from the base, to the final prototype. The different skills of Ada are reviewed. Finally, the target profile is explained.

1.1 Objective

First of all, it is important to understand why Ada was first imagined. The project's objective is to build a robot that combines interaction with humans and cognition. In that perspective, a companion robot is ideal. Indeed, it is composed of a speech recognition part - a very cognitive part where the objective is to detect and understand human language, and a movement reaction part - perfect fit for interaction.

The final prototype's objective is only entertainment, and has no kind of educational or medical purpose. The robot was mainly built to be let on a corner of the desk, and be used when the user is bored from her or his current activity, nothing more.

1.2 State of the art

This concept of a companion robot is not new, and some major corporations have already produced some outstanding prototypes, only couple are listed below.

- **The Buddy Companion :**

Blue Frog Robotics claims to offer a affordable, intelligent and emotional robot, whose goal is to "assist, entertain, educate, and make everybody smile" [3]. It can walk around the house like a pet, and is equipped with a screen which is its major strength. Indeed, it allows to show feelings through the display of a cartoon face. The screen also allows it to play games, make video calls, handle an agenda, etc. Finally, it is able to detect intrusion and fire detection when the owners are not at home. This robot is much more evolved than the one we expected to build during this semester.



Figure 1: Buddy Companion Robot.

- **Vector the robot :** Digital Dream Labs released in 2019 a robot to keep company. It is able to move autonomously and avoid obstacles. More over, it can take some low quality pictures thanks to a little camera. At first, the voice interactions were very limited, but they have greatly improved when it was linked to Amazon *Alexa*[4]. It is not very big, and easily fit on a desk. This robot is a close inspiration to what we wanted to achieve for this project.



Figure 2: Vector the Robot

1.3 Ada's Skills

As for Ada's skills, they can be split into two parts : speech recognition and reaction movement. That is, it firstly understands what has been said to secondly react accordingly. The reaction is made of eye movements, sounds, and leg movements. 5 reactions, each linked to an emotion are implemented. Indeed, Ada can express love, happiness, sadness and anger. Besides that, he can also dance! As a side note, the sound is not described in depth since it is difficult to express in a written report; however, it follows the premise that with low-to-high pitch and tone modulation, it can be interpreted as an uplifting noise as opposed to a sorrow noise otherwise. We encourage the reader to experience the robot to learn more about it.

1.4 Target Profile

It is important to have the target audience clearly in mind. This helps to steer the project in the right direction. The profile drawn up as being ideal for liking Ada has the following characteristics:

- **Geographical aspects** : Western European
- **Demographic aspects** : Between 15 and 40 years old, Male or Female
- **Psychographic aspects** : Finds social robots interesting.
- **Behavioral aspects** : Frequent use of technology.

This user profile is very similar to the one of our university peers and those around them. The geographical aspects of the target profile are such that it includes people with a similar mind set and behaviour to ours. More specifically, people who have grown up with technology and to whom it does not represent an obstacle or a danger. This ensures a certain appeal for robotics. These users are only a small part of a wide range of potential users of our prototype. Secondary stakeholders are not considered for this prototype due to time constraints, but are nevertheless worth mentioning:

- **The toy industry:** Toy robots.
- **Home appliances:** Integrating emotional interaction into an environment.
- **Education:** Teaching programming.
- **Health:** Stress relief.

2 System Requirements

The design requires a certain number of skills. Due to the nature of Ada, it is necessary to first design a physical robot that is not only cute but also can move, make and perceive sounds in order to interact. Then, of course, it has to be made smart. Its code is composed of 2 major parts. First of all the part concerning the management of his body. In other words, interaction. Secondly, give Ada cognitive abilities. In particular, voice recognition which requires software based on machine learning models. In order to perform all those skill, the robot obviously need some specific hardware, described in the next lines.

2.1 Hardware Requirements

The hardware part is essentially made of basic electronics. The modular and open source Otto robot[5] can be used as a basis. Indeed, following online instructions, it is quite straightforward to build the robot from scratch. It has legs with 2 degrees of freedom which allows it to make all kinds of movements to move or even dance. Then, the robot is equipped with a panel of LEDs that are used to display different types of eyes, which will be very useful to express different emotions. Finally, the robot is equipped with a piezo that can emit sounds.

Until here, Ada is only equipped with the chip specific to Otto able manage only physical capabilities, and a shield to link the motors and other connectors. However, this is not enough. The voice recognition is based on more advanced machine learning models, which require more resources. Therefore, it is necessary to add an Arduino Nano on which the voice recognition calculations can take place. Of course, it is necessary to connect the 2 chips so that they can communicate between them.

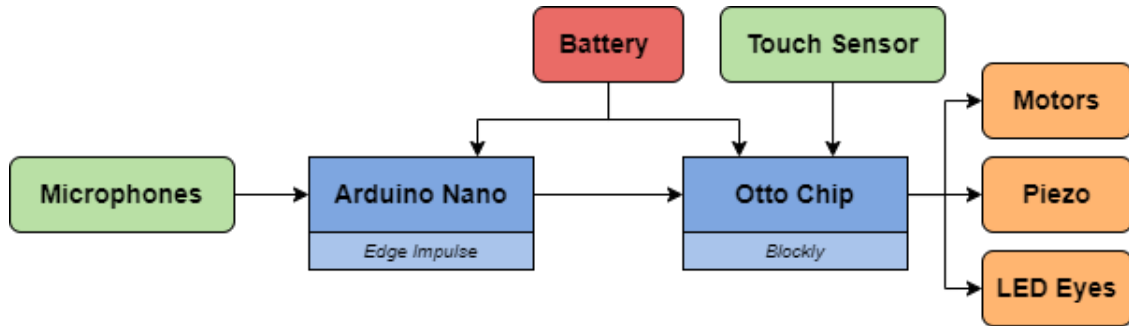


Figure 3: On board links diagram of Ada

One can think of the structure this way: the Nano is the brain that perceives auditory information from the user, processes it and sends the information to the nervous system - Otto's chip, so that it can take care of the motion, sound or visual interaction. See the list of components in Fig. 1.

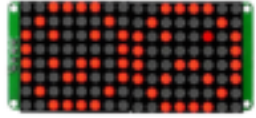



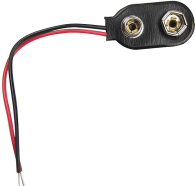

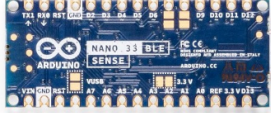
Hardware	Description	Graphic
Matrix led 16×8	Allows Ada express emotions through the eyes	
Servo MG90S	Allows Ada to move its legs	
Touch sensor TTP223	Allows Ada be activated by touching	
Piezo Speaker	Allow Ada to communicate emotions through sounds	
Battery adapter	Powers Ada with 9V battery	
Otto's Nano ATmega328	Physical and actuator's MCU	
Arduino Nano 33 BLUE Sense	Speech recognition brain of Ada	

Table 1: Hardware components

2.2 Software Requirements

As two different chips are used, it makes sense to have two different software, both running on a specific hardware. Edge Impulse[2] helps the speech recognition, while Otto libraries[6] support physical reactions.

2.2.1 Cognition Requirements

Once the microphone of the Nano has recorded sound, it must be processed to know what should be sent to the Otto chip. In that perspective, Edge Impulse is a precious ally. Edge Impulse is an online software developer tool allowing to deal with data acquisition and handling aspects and simplifying the machine learning pipeline for embedded systems (TinyML), among which speech recognition stands out. The principal advantage is that it can be exported onto an Arduino Nano 33 BLE Sense, assuming the proper libraries are included to the C# script in the Open source Arduino IDE. This microcontroller is a very popular and low-power board that enables the deployment of Machine Learning models due to its 32-bit ARM processor running at 64 MHz and 1MB RAM (32 times greater than Arduino Uno), that features an on-board digital microphone for audio tasks [7].

MFCC Feature Extractor

Speech recognition, just like most machine learning fields, requires a large database of examples in order to train a classification algorithm to determine which word is spoken. To do this, a bunch of samples were collected by ourselves from people in our surroundings in order to reach a model of minimal generalized performance. As summarized by Fig. 4, 3710 audio-clips were collected from 15 people, equating to more than 500 samples for each word. Their consent form can be found in Annexes - Database Participant Information Sheet 4. Likewise, the recordings were then submitted to Edge Impulse servers to be splitted and tagged, and finally train and embed the classifier into the Nano 33 chip.

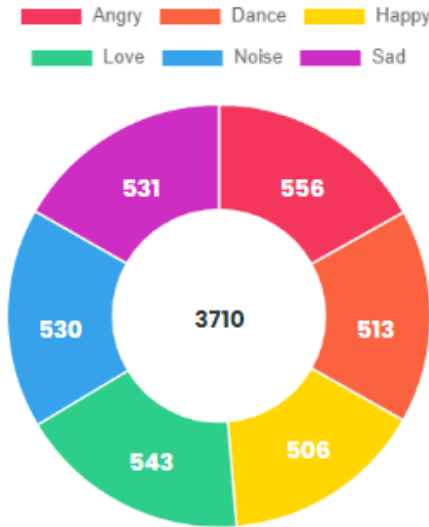


Figure 4: Proportion of data collected for each class

More specifically, Ada’s cognitive skill-set was determined to be using Machine Learning and Neural Networks (NN), to build a system that could recognize audible events, particularly the voice of a bystander, through a task known as *audio classification*. In this task, the system is trained to recognize a keyword from a continuous digital signal processing in the presence of background noise and other distractors such as chatter. Creating a

keyword spotting solution involves gathering data, isolating signals in a time frame, and performing a feature extraction process on them in order to train a neural network that can classify whether the keyword has been said. More importantly, this process requires not only data augmentation for robustness against over-fitting, but also the optimization of the NN's weights in order to be embedded on an inference-capable micro-controller.

MFCC Feature Extractor

Since the low-frequency excitation and formant filtering of the vocal tract are located in different regions of the cepstral domain, the influence of the vocal cords and the vocal tract in a signal can be separated. Mel-Frequency Cepstral Coefficients (MFCC) is the most common non-linear technique for signal processing when it comes to human voice [8]. The Mel spectrum is computed by passing a Fourier transformed signal in the time domain through a Mel-filter bank, resulting in a compressed representation of the filterbanks. Each logarithmic filterbank is passed through a Discrete Cosine Transform to return back to the time domain and extract cepstral coefficients [9]. Next, a visual representation of the 13x50 cepstral coefficients calculation can be seen in Fig. 5.

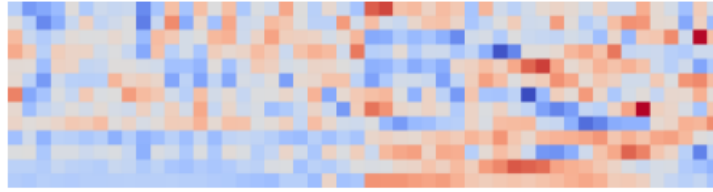


Figure 5: Cepstral coefficients' representation for word Angry (source: Edge Impulse).

Finally, a number of features are shifted and withheld in order to retain the necessary information to emulate the features required to recognize human speech, while the rest are discarded since they do not represent useful patterns for speech recognition. The more relevant parameters used for this project based on the literature [9] are listed below:

- **Number of coefficients:** 13, (12 cepstral features plus energy) coefficients are the usual case in a MFCC when using Neural Networks in a keyword spotting application.
- **Frame stride:** 0.02, is the parameter to determine how many splits will be made in 1 second audio sample; hence 50 showed to be enough for an accurate feature extraction to each sample. This parameter, coupled with the *number of coefficients*, sets the network's input layer ($50 \times 13 = 650$).
- **Low frequency:** 1000 Hz as humans will not perceive lower frequencies than 1k Hz.
- **High frequency:** 20000 Hz as this is the maximum audible range for a human.

In addition, only after the features are created, they can be displayed in 3D space using dimension reduction techniques, as shown in Fig. 6. Although the MFCC allowed for a

distinction of the six classes, certain samples between *angry* and *happy* are very close. This is to be expected since both words are quite similar in tone and vocal terms, therefore difficulties among these words may occur more often in practice.

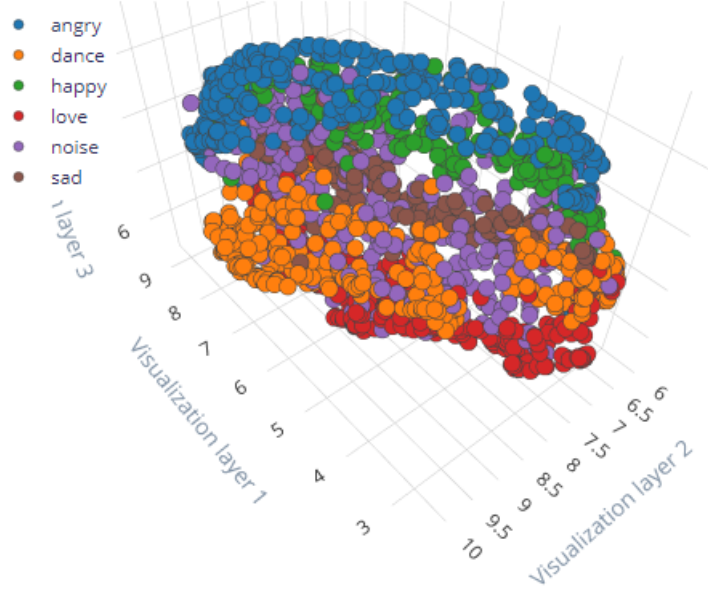


Figure 6: MFCC features visualization in 3D. (source: Edge Impulse)

Neural Network Architecture

Neural Networks are mathematical and computational models that use algorithms for learning how to recognize patterns in data and to generalize on untrained data. The NN used in this case was designed to take the MFCC output as input to attempt to classify the various keywords that were selected. Note that 1D convolutional neural networks are mostly used for one dimension and spatial properties as in the case of audio signal or time-series, which was suitable in this case. After some test, and bearing in mind that needs to be a simple and lightweight Neural Network to be embedded into a tiny micro-controller, the final architecture found and selected is showcased in Table 2.

NN Model results

The expected outcome was determined using an F1 Score and the Accuracy, with a validation data comprising about 20% of the recorded samples, a final loss of 0.14 with an accuracy of 86.52 percent was achieved after 200 training epochs. A learning rate of 0.005 with the Adam optimizer was configured after found in the literature that this stochastic gradient descent method is not only computationally efficient, but has little memory requirements [10]; the results of the confusion matrix can be found in Fig. 7

Neural Network Architecture
Input Layer (650 MFCC features)
Reshape layer (13 columns)
1D conv / pool layer (8 neurons, 3 kernel size, 1 layer)
Dropout (rate 0.25)
1D conv / pool layer (16 neurons, 3 kernel size, 1 layer)
Dropout (rate 0.25)
Flatten layer
Output layer (6 Classes)

Table 2: Neural Network architecture.

	ANGRY	DANCE	HAPPY	LOVE	NOISE	SAD	UNCERTAIN
ANGRY	92.6%	0%	0.7%	0%	0%	0%	6.7%
DANCE	0%	83.8%	0%	9.5%	1.0%	0%	5.7%
HAPPY	1.1%	0%	72.6%	0%	11.2%	1.7%	13.4%
LOVE	1.0%	0%	0%	92.1%	1.0%	0%	5.9%
NOISE	0%	0%	0%	0%	96.3%	0%	3.7%
SAD	0%	0%	0%	1.8%	0.9%	89.4%	8.0%
F1 SCORE	0.95	0.91	0.84	0.90	0.89	0.93	

Figure 7: Model testing results (source: Edge Impulse).

2.2.2 Behavior Requirements

The embedded code in the Otto chip is in C#. It is responsible for making Ada react to signals received by the Arduino Nano. Indeed, Ada is provided with certain emotions. Each one is represented by a sequence of movements, eye displays and noises. These have been implemented using libraries provided by the Otto Blockly program [11].

2.2.3 Communication between the Nano and Otto's chip

So, the principle is that the Arduino Nano sense Ada's environment, in particular the sound. When a word is recognised, the Nano sends a signal to the Otto chip so that it can launch the sequence of interactions that are supposed to represent the corresponding emotion. For example, if a user says "I love you Ada!". The robot will recognise the word "love" in the continuous audio signal. To this, it should react by displaying hearts instead of eyes and doing a little movement with a sound.

The communication between the two chips was done in the following way: The Universal Asynchronous Receiver/Transmitter (UART) performs serial-to-parallel conversion on data received from a peripheral device and parallel-to-serial conversion on data received from the

CPU [12]. In Arduino boards, UART can be utilized for serial communication between two devices. Otto’s MCU is a typical Arduino Nano based on ATmega328, which is normally used for autonomous systems that require a simple micro-controller, so one wire is for transmitting data (Tx pin 0) and the other is for receiving data (Rx pin 1) [13]. Therefore, the serial connection is created between the Otto Nano and the Nano 33 BLE Sense via a cross connection of Tx-Rx and Rx-Tx, and when an audible event is detected, the Nano 33 BLE Sense merely needs to do an on-device inference and immediately communicate the classified keyword through the serial connection to the ATmega328 board, which will trigger the actuators attached to it, the only prerequisite is that they both share the same ground to equalize the voltage. Note that some attempts were made to simplify the circuit interaction and avoid a serial connection, but these were futile for compatibility issues between libraries and therefore both components were required separately, see Annex 4 for the circuit scheme.

2.3 Functional & Non Functional Requirements

In accordance with the proposed work, a definition of the functional and non-functional needs is provided in Tables 3 and 4.

Functional requirements		
No.	Description	Priority
1	Should be able to move four servos in coordination	High
2	Should be able to turn on the leds’ matrix	High
3	Should be able to classify audible events (keywords)	High
4	Should be triggered by a keyword	High
5	Should be able to classify background noise	Medium
6	Should be able to produce sounds with the piezo	Medium
7	Should be portable and wireless	Low
8	Should be triggered by a touch sensor	Low

Table 3: Functional requirements of Ada.

Non-functional requirements		
No.	Description	Priority
1	Should be able to express (Love, happy, sad, angry, dance)	High
2	Should ensemble two servos for each of the two legs/feets of the robot	High
3	Should move each leg of the robot to express (No.1)	High
4	Should be able show eye in the matrix leds to express (No.1)	High
5	Should be able to produce tonal sounds that resemble (No.1)	Medium
6	Should be able to change from emotions by touching the robot’s head	Low

Table 4: Non-functional requirements of Ada.

3 Evaluation

The goal of this project is not only to design and build a robot, but also to evaluate its cognitive and interactive abilities. To do so, the literature’s approach is to create an experimental setup, and apply statistical methods to determine whether the null hypothesis is accepted or rejected. This is the purpose of the last section of this paper.

3.1 Research Question

The first thing to do is to determine what is the research question. That is, find what aspect will be evaluated, and express it in a non-ambiguous way. In the case of Ada the robot, the objective is to know if speech recognition improves the user’s experience significantly. The research question is then : “Does the speech recognition strengthen the interaction between the human and the robot?”. From that question, one can extract a null hypothesis as: “There would be no difference in the interaction between the human and the robot with or without speech recognition.”

3.2 Evaluation Procedure

It is now time to build a whole procedure to follow, in order to end up with an answer to the research question. This gathers three main points. Firstly, the precise description of the experimental setting, in order to be as reliable as possible (try to reduce the external factor to the maximum). Secondly, ask what metrics are used, whether they should be qualitative and/or quantitative. Finally, a choice has to be made concerning the evaluation method itself. In other words, for instance consider whether Chi square or Anova is more appropriate for the situation.

3.2.1 Experimental Setting

In order to test the research question, the following set up has been submitted to the study participants.

Before anything else, the samples were separated into two parts: a control group and a test group. Of course, the participants were not aware of it. The difference between them is that the control group has a “dumb” version of the robot, that makes random actions, no matter what the participant says. The second group faces the complete robot, able to react accordingly to what is told. Both groups however receive the same instructions, available in Annexes - Study Questionnaire. More precisely, the test takes place as follows. The participant is in a room without further distraction than the examiner (who remains quiet), and the robot. The participant is asked to pronounce a series of 5 sentences matching what Ada is supposed to understand, and to evaluate the reaction, on a scale to 1 to 5. After that, he is thanked and it’s the turn of the next contributor.

In this context, the independent variables are the dumb/smart version of the robot (according the participant’s group), as well as the form to fill and the sentences to say to the

robot. The dependent variables, on the other hand, are the participants answers to the form.

3.2.2 Metrics

After the experience is over, 5 numbers per participant are gathered. Those numbers are the tester's appreciation concerning each one of the 5 reactions he or her saw. Since the reactions have the same weight in the survey, an easy way to end up with only one number per participant is to take the mean.

The metric is now clear. It is quantitative, and represent the person's feeling about the robot's reaction. The fact that it is quantitative makes it easier to deal with, for example to find p-value to accept or reject the null hypothesis.

3.2.3 Methods

With the help of the metric, how to answer the research question? The usual approach is to compute a p-value of 95% certainty, and reject the null hypothesis if it's below 0.05. The method to compute this p-value still remains though. For this task, T-test is well suited. Indeed, the task only has 2 dependent groups (control and test group), and the values can be argued to be normally distributed.

3.2.4 Biases

When conducting any study, it is very important to think of the bias that could exist; especially in case of a study implying human participants, such as this one. Indeed, even though the robot itself should not have much biases, the participants are likely to have some. Most of the time, it is difficult or even impossible to get rid of them - it is in human nature, but it is essential to be aware of them, and take them into account when drawing conclusions. Below are 4 of those biases. Among them, some are participant biases, others concern the experimental setting, or the study itself.

- **Sampling Bias** It concerns the way the participants are gathered. One should be careful to contact a sample of people that are representative of the population that is tried to be studied. In this case, the population is already quite precise, which makes things easier at sampling.
- **Response Bias** This bias is linked with the first one. Even if the group of participant look representative of the whole population, beware that they still might share traits of personality. Especially if the volunteers are friends or relative to the searchers, they will tend to have similar ideas that could influence the outcome of the study.
- **Social Desirability Bias** This bias refers to the participants answers. More precisely, they will - mostly implicitly - try to give responses that will please the interviewer, even though their opinion is slightly different. In other words, people tend to be more politically correct than honest.

- **Confirmation Bias** Confirmation bias also refers to the participants answers, but in another way. In studies where the hoped outcome is clear, people have a tendency to convince themselves that it goes in that direction. In the case of this study, it is pretty clear that the interviewers hope that speech recognition indeed improves the interaction, and volunteers might give higher scores to the interaction than they would in another context. To fight this bias, it is important to give as few information as possible.

3.3 Results

The survey's results have been treated with the Python script from Annex 4, whose output can be found on Figure 8.

```
=====
Variance of control group: 0.247
Variance of test group:    0.148
P-value for two tailed test is 0.085494
Null hypothesis cannot be rejected
=====
```

Figure 8: P-value results from python script

The plot from Figure 9 displays all survey numbers in a visual way, using a box plot. On this figure, the purple lines correspond to the mean score, while the blue boxes represent the area containing 50% of all scores, for both control and test groups.

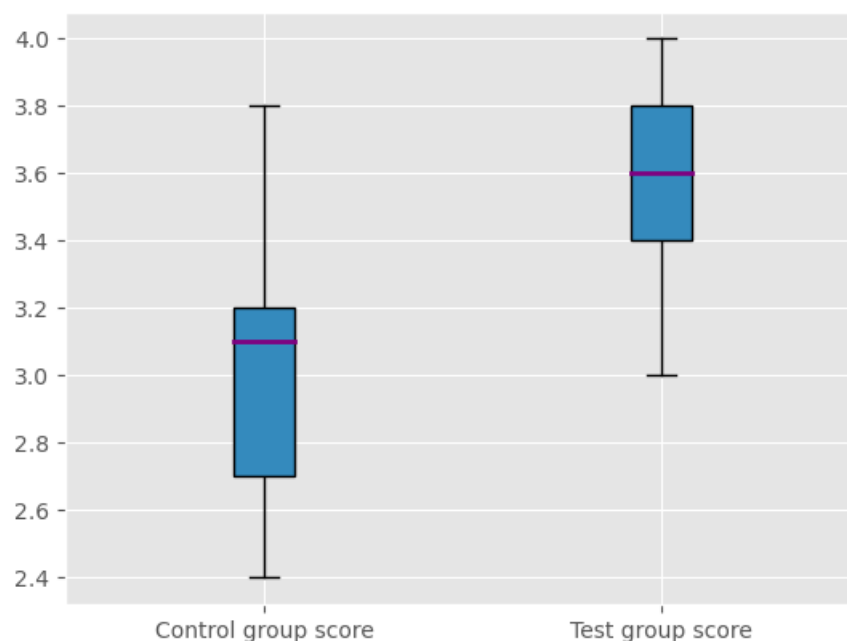


Figure 9: Box plot of the survey's results

3.4 Discussion & Interpretation

Before any interpretation of the p-value, it is smart to analyse the data from the survey with the help of Figure 9. First thing to see, the control group grades the robot with worse marks than the test group, since the blue box is located under the other on the plot, and its mean is lower. This is encouraging: it says that speech recognition improves the interaction with human! This result is good, but it might be a stroke of luck. P-value will tell more about this, but let's go on analysing the graph. The second point is the variance of the grades, that can be seen as the spread of the results. It can already be seen intuitively on figure 9 that the first group's variance is higher, and the python script confirms it (see Figure 8). This also makes sense, since the control group saw random reactions, from a uniform distribution (largest variance possible). Knowing this, one can even ask why it is not higher. Two arguments can explain it. The first one is that the scores from each participant are not likely to be in the extremes (1 or 5), since they come already from a mean. Secondly, the confirmation bias tend to make people give good grades, thus lowering the variance (the spread is smaller as the results are closer to highest possible grade).

Is this interpretation enough ? As previously said, no. It still remains to check whether or not luck was part of the conclusions. To do that, the best is to look at the T-test p-value from Figure 8. A p-value of 0.085 means that there is a 8.5% chance that the above conclusion (saying that interaction is indeed improved by speech recognition) was a stroke of good luck, and are in fact not true and null hypothesis cannot be rejected.

In HRI, people usually consider that the null hypothesis should be kept as long as the p-value is higher than 5%. In a formal sense, so should be ours. With a bit of criticism however, it can be argued that 91.5% of chance to be accurate is high enough.

4 Conclusion

This document recapitulates an attempt to approach how the project was developed, from the first ideas to the final implementation. In general, the plan was very well laid out from the beginning, which allowed us to move forward efficiently without changing ideas at meeting. Indeed, the only thing that differs between the first and this final version is the use of two chips instead of one. Even if this has caused some delay in the planning, the fact that they could be connected made it possible to stick to the plan, and not give up any functionality because of it.

Although the plan was clear and respected, some expectations had to be lowered, while staying in the same spirit. Indeed, the amount of training sample for the speech recognition being quite low as compared to other performing softwares, we choose to implement fewer abilities but with higher success rate. The power of speech recognition associated to Otto is then demonstrated, and can be improved with a large database. The future amelioration for Ada depends mostly on the training database's size (to add features and improve the current ones). Yet, some features of initial Otto robot were not used

in this project, such as the ability to create any song. A possibility would then be to ask Ada to play any song, so it can find it, reproduce it and perhaps dance! Apart from using other Otto functionalities, one must not forget that all the robot is powered with Arduino hardware. The ideas of improvement using all available technologies are vast, for instance, using WiFi or Bluetooth connectivity. Likewise, improving the amount of testing subjects must be another thing to improve in future versions so a proper sample size can be use for relevant insights.

To conclude, this course has given us a lot during this semester. First of all, the fact of working in a team on a consistent project is a great challenge to take. The fact that this team is composed of international members is even more interesting. Dealing with speech recognition is probably the biggest of this course. In particular, none of us had ever worked on it before. The testing part with foreign participants was also a personal achievement for all of us, and allowed us to go deeper in the project's evaluation and interaction than we wouldn't have done otherwise. Thus, we are proud to present our final prototype of Ada the Robot which, from our survey's volunteers point of view, was very well accepted. But don't take our word for it, have a look at the demonstration on YouTube[14].

References

- [1] Matt Simon. “Companion Robots Are Here. Just Don’t Fall in Love With Them”. In: *WIRED* (2021).
- [2] Edge Impulse. *Advanced ML for every solution*. URL: <https://www.edgeimpulse.com/>. (accessed: 07.01.2022).
- [3] Blue Frog Robotics. *Buddy the first emotional companion robot*. URL: <https://buddytherobot.com/en/buddy-the-emotional-robot/>. (accessed: 17.10.2021).
- [4] Numerama. *Vector, le petit robot qui ne sert vraiment à rien, est de retour*. URL: <https://www.numerama.com/tech/598386-vector-le-petit-robot-qui-ne-sert-vraiment-a-rien-est-de-retour.html>. (accessed: 16.10.2021).
- [5] OttoDIY. *build, code & design your own robot*. URL: ottodiy.com. (accessed: 09.01.2022).
- [6] OttoDIY. *OttoDIYLib - GitHub*. URL: github.com/OttoDIY/OttoDIYLib. (accessed: 09.01.2022).
- [7] Arduino. *Arduino Nano 33 BLE Sense*. URL: <https://store-usa.arduino.cc/products/arduino-nano-33-ble-sense>. (accessed: 07.01.2022).
- [8] M. Likitha et al. “Speech based human emotion recognition using MFCC”. In: *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)* (2017), pp. 2257–2260.
- [9] Lindasalwa Muda, Mumtaj Begam, and Irraivan Elamvazuthi. “Voice Recognition Algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques”. In: *ArXiv abs/1003.4083* (2010).
- [10] Diederik P. Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: *CoRR abs/1412.6980* (2015).
- [11] OttoDIY. *Otto Blocky*. URL: <https://ottodiy.github.io/blockly/www/>. (accessed: 08.01.2022).
- [12] Yuan Liu et al. “Design and Verification of UART Circuit of SoC Based on AMBA Bus”. In: *2020 7th International Conference on Information Science and Control Engineering (ICISCE)* (2020), pp. 2370–2374.
- [13] Atmel. *ATmega328P Data sheet*. URL: <http://www.datasheet.es/PDF/1057332/ATmega328P-pdf.html>. (accessed: 9.01.2022).
- [14] Antoine C. *Ada, the companion robot*. URL: https://www.youtube.com/watch?v=UjGXaB7CqaM&feature=youtu.be&ab_channel=AntoineC.

Annexes

Participant Information Sheet

Title of the study: Does the speech recognition strengthen the interaction between the human and the robot?

Title of the project: Ada, the companion robot

Promotor: Universitat Politecnica de Catalunya

Principal investigator: Augustin Lambert

Researchers: Antoine Caytan, David Gaviria

Center of affiliation: Universitat Politecnica de Catalunya

Place: Live It Ramblas - Working Room - Carrer de la Unio 7, Barcelona

1) Introduction and Procedures

Dear Participant,

You are invited to participate in the research project described below.

a. Project Description

The aim of the project is to create a companion robot that serves as a distraction for a young adult interested in technology.

b. Participation Implication

Your involvement in the project is to interact with the robot in order to analyse your behaviour towards it. Your participation in this study is totally voluntary. You can decide to participate or not in this project. Similarly, you can leave the study at any time by revoking the informed consent without affecting it in any way. You have the possibility to choose the destination of your data in case of withdrawing from the study, including its destruction.

c. Description of the Activity

The test will take about **5 minutes**. It consists of **5 operations**. In each operation, you will be given a **sentence**. You will have to express yourself to the robot by **reading** this sentence. Then, evaluate the interaction you had with it on a scale of 1 to 5.

- 1 : **very bad** or almost no interaction.
- 5 : **good** interaction.

2) Nature of the Participation

a. Benefits

The participant in this study will be able to know the research that takes place in our centre about social robotics. In addition, you can interact with the Ada robot and resolve any doubts you may have about the technology involved.

b. Risks

The security risks are minimal. In this case: the robot has no sharp edges and no part of the electronic system can create any sensation. In any case, the box is solid and no physical interaction will be required from the participant. The robot has a unique on/off button that is directly connected to the battery that can play the role of emergency stop buttons. In addition, the robot is programmed to operate in docile mode and at a relatively low speed, which guarantees the safety of the participants.

c. Disseminations of the results

The only personal information that will be collected from each participant is their age, gender and country of origin. The name of the participant will not be asked so that the study is guaranteed to be anonymous. No audio or visual recording will be made. The results will not be published but simply used by the project team.

3) Additional Information

You have the right to clarify all the doubts that are presented to you at any time, being able to request more detailed information about the investigation. For this you can contact the principal investigator or the responsible researcher who will be in the same room. If you consider that all doubts have been clarified and you are convinced to participate in this study, you can then sign the informed consent form here under.

4) Informed Consent Form

I **declare** that I have read the Participants Information Sheet and that a copy has been given to me, that I have had sufficient time and been given the opportunity to ask questions, and that I have received sufficient information from the researcher, who has adequately informed me of the conditions of my participation in this investigation. I have been assured of the confidential treatment of my data.

I further **declare** that I understand that my participation is voluntary, so that I can withdraw from the research freely, at any time during the experiment and for any reason, and that:

☐ **I GIVE,**

☐ **I DO NOT GIVE,**

my consent to participate in the research that has been proposed to me.

Date and **Signature** of the participant:

Study Questionnaire

Title of the study: Does the speech recognition strengthen the interaction between the human and the robot?

Title of the project: Ada, the companion robot

Promotor: Universitat Politecnica de Catalunya

Principal investigator: Augustin Lambert

Researchers: Antoine Caytan, David Gaviria

Center of affiliation: Universitat Politecnica de Catalunya

Place: Live It Ramblas - Working Room - Carrer de la Unio 7, Barcelona

1) Participant Information

Dear Participant,
You are invited fill in your following information:

Age:

Gender: ☐ **Male**, ☐ **Female**

Country:

2) Instructions

The test will take about **5 minutes**. It consists of **5 operations**. In each operation, you will be given a **sentence**. You will have to express yourself to the robot by **reading** this sentence. Note that Ada does not always react correctly. Feel free to **repeat** the sentence several times until Ada understands. Once Ada has reacted, write down the number of times you had to repeat yourself and rate your appreciation of the reaction on a scale of 1 to 5.

- 1 : **very bad** or almost no interaction
- 5 : **good**

3) Evaluation

a. **Say :** *"I love you Ada"*

How many times have you had to repeat :

Rate your appreciation of the reaction : 1 - 2 - 3 - 4 - 5

b. **Say :** *"Are you happy Ada ?"*

How many times have you had to repeat :

Rate your appreciation of the reaction : 1 - 2 - 3 - 4 - 5

c. **Say :** *"I'm sad"*

How many times have you had to repeat :

Rate your appreciation of the reaction : 1 - 2 - 3 - 4 - 5

d. **Say :** *"Let's dance !"*

How many times have you had to repeat :

Rate your appreciation of the reaction : 1 - 2 - 3 - 4 - 5

e. **Say :** *"I'm really angry"*

How many times have you had to repeat :

Rate your appreciation of the reaction : 1 - 2 - 3 - 4 - 5

Database Participant Information Sheet

Title of the study: Does the speech recognition strengthen the interaction between the human and the robot?

Title of the project: Ada, the companion robot

Promotor: Universitat Politecnica de Catalunya

Principal investigator: Augustin Lambert

Researchers: Antoine Caytan, David Gaviria

Center of affiliation: Universitat Politecnica de Catalunya

Place: Live It Ramblas - Working Room - Carrer de la Unio 7, Barcelona

1) Introduction and Procedures

Dear Participant,

You are invited to participate in the creation of an audio database.

a. Project Description

The aim of the project is to create a companion robot that serves as a distraction for a young adult interested in technology. It must be able to recognise certain key words. This is achieved by using machine learning models that require a database of examples to train this artificial intelligence.

b. Participation Implication

Your participation in the project consists of recording yourself saying the different key words out loud. Your participation in this study is completely voluntary. You can decide whether or not to participate in this project. You can also withdraw your participation at any time by revoking your informed consent without affecting the study in any way. You have the option of choosing what happens to your data if you withdraw from the study, including its destruction.

c. Description of the Activity

What you are asked to do will take just over 5 minutes. It consists of 5 operations. In each operation, you are given a word. You will have to say this word out loud and repeat it for one minute. Be careful to leave a short silence of about half a second between each repetition of the word and not to go too fast. Try to change your intonation a little sometimes.

2) Nature of the Participation

a. Benefits

Participation in this audio database will allow researchers to train the model that Ada will be equipped with to be able to recognise speech.

b. Disseminations of the results

The only personal information that will be collected from each participant is their first and last name to prove their contentment. The recordings will then be anonymised. The audio files will never be published publicly but will be used by the project team to train the model.

3) Additional Information

You have the right to clarify all the doubts that are presented to you at any time, being able to request more detailed information about the investigation. For this you can contact the principal investigator or the responsible researcher who will be in the same room. If you consider that all doubts have been clarified and you are convinced to participate in this study, you can then sign the informed consent form here under.

4) Informed Consent Form

I **declare** that I have read the Participants Information Sheet and that a copy has been given to me, that I have had sufficient time and been given the opportunity to ask questions, and that I have received sufficient information from the researcher, who has adequately informed me of the conditions of my participation in this database. I have been assured of the confidential treatment of my data.

I further **declare** that I understand that my participation is voluntary, so that I can withdraw from the operation freely, at any time and for any reason, and that:

☐ **I GIVE,**

☐ **I DO NOT GIVE,**

my consent to participate in the creation of that audio database that has been proposed to me.

Date and **Signature** of the participant:

Evaluation Python Script

```
1  import numpy as np
2  from scipy import stats
3  import statistics
4  import matplotlib.pyplot as plt
5  plt.style.use('ggplot')
6
7  # Scores obtained for the interaction appreciation by the control group
8  # The penalties due to late reaction have already been applied.
9  test = np.array([[4.0, 4, 2, 5, 4],
10                  [2, 4, 3, 4, 5],
11                  [2, 2, 1, 5, 5],
12                  [4, 4, 2, 5, 5],
13                  [4, 4, 3, 4, 2]])
14
15  # Scores obtained for the interaction appreciation by the test group
16  # The penalties due to late reaction have already been applied.
17  control = np.array([[1.0, 3, 2, 5, 1],
18                     [4, 3, 1, 5, 3],
19                     [5, 3, 5, 2, 4],
20                     [3, 4, 1, 3, 2],
21                     [2, 1, 4, 3, 5],
22                     [5, 4, 2, 3, 2]])
23
24  # get the mean score of each participant
25  control_mean = np.nanmean(control, axis = 1)
26  test_mean = np.nanmean(test, axis = 1)
27  print(control_mean)
28  print(test_mean)
29  # display both results
30  fig, ax = plt.subplots()
31  ax.boxplot((control_mean, test_mean), vert=True, showmeans=False, meanline=True,
32             labels=('Control group score', 'Test group score'), patch_artist=True,
33             medianprops={'linewidth': 2, 'color': 'purple'},
34             meanprops={'linewidth': 2, 'color': 'red'})
35  plt.show()
36
37  # compute the p-value with T-test
38  _, p_value = stats.ttest_ind(control_mean, test_mean)
39  alpha = 0.05
40
41  print("=====")
42  print("Variance of control group:", round(statistics.variance(control_mean), 3))
43  print("Variance of test group: ", round(statistics.variance(test_mean), 3))
44  print(' P-value for two tailed test is %f'%p_value)
45  if p_value <= alpha:
46      print(' Null hypothesis can be rejected')
47  else :
48      print(' Null hypothesis cannot be rejected')
49  print("=====")
```

Arduino IDE

Hardware and actuator's code



```
Otto_Arduino $
#include <Otto.h>
Otto Otto;
#include <Wire.h>
#include "Adafruit_LEDBackpack.h"
Adafruit_8x16matrix ematrix = Adafruit_8x16matrix();

int emotion = -1;

#define LeftLeg 2 // left leg pin, servo[0]
#define RightLeg 3 // right leg pin, servo[1]
#define LeftFoot 4 // left foot pin, servo[2]
#define RightFoot 5 // right foot pin, servo[3]
#define Buzzer 13 //buzzer pin

static const uint8_t PROGMEM
eyes_bmp[] = {  B00000000, B00111100, B01000010, B01001010, B01000010, B01
happy_bmp[] = {  B00000000, B00111100, B00000010, B00000010, B00000010, B0
sad_bmp[] = {   B00000000, B00010000, B00010000, B00010000, B00010000, B00
angry_bmp[] = {  B00000000, B00011110, B00111100, B01111000, B01110000, B0
angry2_bmp[] = { B00000000, B00000010, B00000100, B00001000, B00010000, B
love_bmp[] = {   B00000000, B00001100, B00011110, B00111100, B00111100, B00

void setup() {
  Serial.begin(9600);
  Otto.init(LeftLeg, RightLeg, LeftFoot, RightFoot, true, Buzzer);
  Otto.home();

  ematrix.begin(0x70); // pass in the address

  pinMode(A0, INPUT);
  ematrix.setBrightness(15); //the brightness of the LEDs use values from 0 to
  ematrix.clear();
  ematrix.drawBitmap(0, 0, + eyes_bmp , 8, 16, 1);
  ematrix.writeDisplay();
  delay(1 * 1000);
}
```

```

void loop() {
  if (digitalRead(A0)) {
    emotion = emotion + 1;
  }
  char e = Serial.read();
  if (emotion == 0 || e == '0') { // Angry
    if (emotion == 0)
      emotion = emotion + 1;
    long r = random(0, 100);
    ematrix.clear();
    if ( r > 50) {
      ematrix.drawBitmap(0, 0, + angry_bmp, 8, 16, 1);
      ematrix.writeDisplay();
    }
    else if ( r <= 50) {
      ematrix.drawBitmap(0, 0, + angry2_bmp, 8, 16, 1);
      ematrix.writeDisplay();
    }
    Otto.playGesture (OttoAngry);
    ematrix.clear();
    ematrix.drawBitmap(0, 0, + eyes_bmp, 8, 16, 1);
    ematrix.writeDisplay();
  }
  else if (emotion == 2 || e == '1') { // DANCE
    if (emotion == 2)
      emotion = emotion + 1;
    long r = random(0, 100);
    ematrix.clear();
    ematrix.drawBitmap(0, 0, + happy_bmp, 8, 16, 1);

    Otto.moonwalker(1, 1000, 25, 1);
    Otto.moonwalker(1, 1000, 25, -1);
    Otto.crusaito(1, 1000, 25, 1);
    Otto.crusaito(1, 1000, 25, -1);
    Otto.flapping(1, 1000, 25, 1);
    Otto.flapping(1, 1000, 25, -1);
    ematrix.clear();
    ematrix.draw28Bitmap(0, 0, + eyes_bmp, 8, 16, 1);
    ematrix.writeDisplay();
  }
  if ( emotion == 9) {
    emotion = -1;
  }
}

```

```

else if (emotion == 4 || e == '2') { //HAPPY
    if (emotion == 4)
        emotion = emotion + 1;
    long r = random(0, 100);
    ematrix.clear();
    ematrix.drawBitmap(0, 0, + happy_bmp, 8, 16, 1);
    ematrix.writeDisplay();
    if ( r > 50) {
        Otto.playGesture (OttoHappy);
    }
    else if ( r <= 50) {
        Otto.playGesture (OttoSuperHappy);
    }
    ematrix.clear();
    ematrix.drawBitmap(0, 0, + eyes_bmp, 8, 16, 1);
    ematrix.writeDisplay();
}
if (emotion == 6 || e == '3') { //LOVE
    if (emotion == 6)
        emotion = emotion + 1;
    long r = random(0, 100);
    ematrix.clear();
    ematrix.drawBitmap(0, 0, + love_bmp, 8, 16, 1);
    ematrix.writeDisplay();
    Otto.playGesture (OttoLove);
    ematrix.clear();
    ematrix.drawBitmap(0, 0, + eyes_bmp, 8, 16, 1);
    ematrix.writeDisplay();
}
if (emotion == 8 || e == '5') { //SAD
    if (emotion == 8)
        emotion = emotion + 1;
    long r = random(0, 100);
    ematrix.clear();
    if ( r > 50) {
        ematrix.drawBitmap(0, 0, + sad_bmp, 8, 16, 1);
        ematrix.writeDisplay();
    }
    else if ( r <= 50) {
        ematrix.drawBitmap(0, 0, + fail_bmp, 8, 16, 1);
        ematrix.writeDisplay();
        Otto.playGesture (OttoSad);
        ematrix.clear();
        ematrix.drawBitmap(0, 0, + eyes_bmp, 8, 16, 1);
    }
}

```

Keyword spotting main code



```
serial_robust_nano_ble33_sense_microphone_continuous

/* Includes ----- */
#include <PDM.h>
#include <blobquiet-project-1_inferencing.h>

/** Audio buffers, pointers and selectors */
typedef struct {
    signed short *buffers[2];
    unsigned char buf_select;
    unsigned char buf_ready;
    unsigned int buf_count;
    unsigned int n_samples;
} inference_t;

static inference_t inference;
static bool record_ready = false;
static signed short *sampleBuffer;
static bool debug_nn = false; // Set this to true to see e.g. features generated
static int print_results = -(EI_CLASSIFIER_SLICES_PER_MODEL_WINDOW);

/**
 * @brief      Arduino setup function
 */
void setup()
{
    // put your setup code here, to run once:
    Serial.begin(115200);
    Serial1.begin(9600);
    Serial.println("Edge Impulse Inferencing Demo");

    // summary of inferencing settings (from model_metadata.h)
    ei_printf("Inferencing settings:\n");
    ei_printf("\tInterval: %.2f ms.\n", (float)EI_CLASSIFIER_INTERVAL_MS);
    ei_printf("\tFrame size: %d\n", EI_CLASSIFIER_DSP_INPUT_FRAME_SIZE);
    ei_printf("\tSample length: %d ms.\n", EI_CLASSIFIER_RAW_SAMPLE_COUNT / 16);
    ei_printf("\tNo. of classes: %d\n", sizeof(ei_classifier_inferencing_categories)
        sizeof(ei_classifier_inferencing_categories[0]));

    run_classifier_init();
    if (microphone_inference_start(EI_CLASSIFIER_SLICE_SIZE) == false) {
        ei_printf("ERR: Failed to setup audio sampling\r\n");
        return;
    }
}
```

```

void loop()
{
    bool m = microphone_inference_record();
    if (!m) {
        ei_printf("ERR: Failed to record audio...\n");
        return;
    }

    signal_t signal;
    signal.total_length = EI_CLASSIFIER_SLICE_SIZE;
    signal.get_data = &microphone_audio_signal_get_data;
    ei_impulse_result_t result = {0};

    EI_IMPULSE_ERROR r = run_classifier_continuous(&signal, &result, debug_nn);
    if (r != EI_IMPULSE_OK) {
        ei_printf("ERR: Failed to run classifier (%d)\n", r);
        return;
    }

    if (++print_results >= (EI_CLASSIFIER_SLICES_PER_MODEL_WINDOW)) {
        // print the predictions
        ei_printf("Predictions ");
        ei_printf("(DSP: %d ms., Classification: %d ms., Anomaly: %d ms.)",
                   result.timing.dsp, result.timing.classification, result.timing.anomaly);
        ei_printf(": \n");
        for (size_t ix = 0; ix < EI_CLASSIFIER_LABEL_COUNT; ix++) {
            ei_printf("    %s: %.5f\n", result.classification[ix].label,
                      result.classification[ix].value);
        }
        if (result.classification[0].value > 0.70) {
            Serial1.write('0');
        }
        else if (result.classification[1].value > 0.70) {
            Serial1.write('1');
        }
        else if (result.classification[2].value > 0.70) {
            Serial1.write('2');
        }
        else if (result.classification[3].value > 0.70) {
            Serial1.write('3');
        }
        else if (result.classification[5].value > 0.70) {
            Serial1.write('5');
        }
    }
    #if EI_CLASSIFIER_HAS_ANOMALY == 1
        ei_printf("    anomaly score: %.3f\n", result.anomaly);
    #endif
}

```


Neural network architecture code

```
1 import tensorflow as tf
2 from tensorflow.keras.models import Sequential
3 from tensorflow.keras.layers import Dense, InputLayer, Dropout
4     , Conv1D, Conv2D, Flatten, Reshape, MaxPooling1D,
5     MaxPooling2D, BatchNormalization, TimeDistributed
6 from tensorflow.keras.optimizers import Adam
7
8 # model architecture
9 model = Sequential()
10 # Data augmentation, which can be configured in visual mode
11 model.add(tf.keras.layers.GaussianNoise(stddev=0.45))
12 model.add(Reshape((int(input_length / 13), 13), input_shape
13     =(input_length, )))
14 model.add(Conv1D(8, kernel_size=3, activation='relu', padding
15     ='same'))
16 model.add(MaxPooling1D(pool_size=2, strides=2, padding='same'
17     ))
18 model.add(Dropout(0.25))
19 model.add(Conv1D(16, kernel_size=3, activation='relu', padding
20     ='same'))
21 model.add(MaxPooling1D(pool_size=2, strides=2, padding='same'
22     ))
23 model.add(Dropout(0.25))
24 model.add(Flatten())
25 model.add(Dense(classes, activation='softmax', name='y_pred'))
26
27 # this controls the learning rate
28 opt = Adam(lr=0.005, beta_1=0.9, beta_2=0.999)
29 # Data augmentation for spectrograms, which can be configured
30 # in visual mode.
31 # To learn what these arguments mean, see the SpecAugment
32 # paper:
33 # https://arxiv.org/abs/1904.08779
34 sa = SpecAugment(spectrogram_shape=[int(input_length / 13),
35     13], nF_num_freq_masks=3, F_freq_mask_max_consecutive=4,
36     nT_num_time_masks=3, T_time_mask_max_consecutive=2,
37     enable_time_warp=True, W_time_warp_max_distance=6,
38     mask_with_mean=False)
39 train_dataset = train_dataset.map(sa.mapper(), tf.data
40     .experimental.AUTOTUNE)
41
42 # this controls the batch size, or you can manipulate the tf
43 # .data.Dataset objects yourself
44 BATCH_SIZE = 32
45 train_dataset = train_dataset.batch(BATCH_SIZE, drop_remainder
46     =False)
47 validation_dataset = validation_dataset.batch(BATCH_SIZE,
48     drop_remainder=False)
49 callbacks.append(BatchLoggerCallback(BATCH_SIZE,
50     train_sample_count))
51
52 # train the neural network
53 model.compile(loss='categorical_crossentropy', optimizer=opt,
54     metrics=['accuracy'])
55 model.fit(train_dataset, epochs=500, validation_data
56     =validation_dataset, verbose=2, callbacks=callbacks)
```

Circuit scheme on breadboard

