جامعة أبو بكر بلقايد - تلمسان

Université Abou Bakr Belkaïd de Tlemcen

Faculté de Technologie

Département de Génie Biomédical

MEMOIRE DE PROJET DE FIN D'ETUDES

Pour l'obtention du Diplôme de

MASTER en GENIE BIOMEDICAL

Spécialité : Instrumentation Biomédicale

Présenté par : MEZOUAR Bochra Nour ElHouda et BOUREGA Mohammed.

Smart infant cry detection system with raspberry pi

Soutenu lundi 28 septembre 2020 devant le Jury

Mr	**KHOLKHAL** Mourad	MCB	Université de Tlemcen	Président
Mr	**DEBBAL** Sidi Mohammed	Pr	Université de Tlemcen	Examinateur
Melle	**HABIBES** Naima	MAA	Université de Tlemcen	Encadreur

Année universitaire 2019-2020

Acknowledgements

Writing these words and wondering how can we give these five years of studying the justice to revive it, we never thought this last year would be such a challenge for us, in just a few months of second semester, we managed to achieve what started with an idea, what a journey full of ups and downs with all sorts of events, where can we get the device components? Where can we work? With this pandemic, how can we acquire a new knowledge correctly in such a very narrow time?…., surprisingly and proud to say this was the most productive year, and that a lot of New technologies are absorbed.

We have to admit that me and my co-partner could ‘not start this without the encouragement of our beloved supervisor “HABIBES NAIMA”, she was here for every enquiry and question from the start to the end, without any hesitation for our demands and helping us even with her own components.

Without forgetting the jury professor “KHOLKHAL” and professor “DEBBAL” for spending every minute to read our thesis and giving us his constructive comments for better results, every Remark is important to us and valuable, for our progress.

We cannot finish this thanking note without mentioning the best engineers staff of biomedical department for being at our service at all-time even during lunch and in late evening when we had to stay for extra time, also the administration staff from the head of the department to his assistants and security agent ”Omar” he made sure that our class is always reserved with keeping silence, thanking also the best teachers of biomedical department for their passion to give the information and share the maximum.

Special acknowledgment for mister “Shivam Sharma” for sharing his work papers and database without any hesitation and being around for answering our questions, thank you.

Every step was with the guidance of ALLAH, we are eternally grateful for being in our side, all praises to Allah.

Dedicates

I want to dedicate this work for:

My beloved parents Miloud and Naima for being all the time at my side, hopping I will always make you proud and May Allah keep you beside me.

To my dearest brothers Sidahmed,Yacine ,Abdennour ,for putting up with my tempers during this period.

For my deaf cousins Meriem and Khouira with her kids they were inspiration of this project, I hope this solution will provide them even a quarter of comfort.

For my cousin Halima Berouain she was always here for motivation, and my darling Nesrine Belhadji.

For my supervisor "Habibes Naima" for her valuable time and support.

To my project partner Mohammed for crossing this road together and for consisted motivation.

To my best friend forever Fatima Zohra Bechlaghem for barring my crazy moment of pressure.

For mister "Dib Amazigh", he believed in me from first time I pitched my idea, and helped me to share it in the German biomedical event.

For mister "Mostafa El Habib Daho" for always supporting students with new opportunities, that I shared the project poster idea in it.

Thanks for every treasured family members of MEZOUAR & ARBAOUI

, Friends, and to INB promotion 2020.

With love & peace Bouchra Mezouar

Dedicates

I want to dedicate this work to:

My beloved parents Lazhari and Ismahane for being all the time at my side, hopping I will always make you proud and May Allah keep you beside me.

To my brothers Lokmane,Oussama ,Abdelhakim and our little princess Meriem for supporting me and putting up with my tempers during this period.

For my cousin Abderahman Boubekeur and Imad, they were always beside me and motivation me.

To my project partner Bouchra for all the support, the motivation and for her work ethic and the inspiration.

To my friend Ahmed Zazia (Cabrage),Berete Moriba, Anis, Shaib, Khiredine (Khayro), Walid, Adberazak, Rachid, Nadir, Abdelkarim, Deyaa, Hamza, Morad (ABMK) for the great and crazy moments of university and of course I must mention Aziez Mostapha (Sami), Bilal, Seddyk, Aimen,Mohammed for the good old days and moments.

For all the deaf people especially deaf parents, I hope this solution will provide them some comfort.

Friends, and to INB promotion 2020 (Boys and girls).

Bourega Mohamed (L'Mouh)

Abstract

The infant communicates with the environment by crying. Babies produce this cry after being stimulated by pain, discomfort, emotional attention needs or environmental factors. It also contains valuable information about the infant's condition.

This project concerns the development and the realization of a device based on the Raspberry Pi to detect baby cries and predict its cause.

This study describes a database compiled to analyze the relationship between infant crying and its reasons, using spectrograms that provide topographic truth and fundamental frequency information with harmonics.

Signal processing methods such as linear predictive analysis, CWT, DWT and MFCC were used to analyze the infant cries and extract the energy characteristics and the coefficient parameters. We attempts to classify baby cries into two categories such as pain and discomfort using several methods, we mention Neural Networks (NN), Convolutional Neural Networks (CNN) and Support Vector Machine (SVM), and we showcase the application into Node red dashboard for production part.

Such a device can be used by parents (especially deaf parents) or a person who takes care of babies. This type of technology and device exists, but it is not available to a large group of people so we tried to make it available for all the different slices of community.

Keywords:

Baby cry, Deaf, Sound detector, Raspberry Pi, Arduino, Spectrogram, FFT, CNN, SVM, Machine learning, Artificial intelligence, Watson studio, Node red, signal processing, python, sounds, microphone.

Résumé

Le nourrisson communique avec son environnement en pleurant. Les bébés produisent ces pleurs après avoir été stimulés par la douleur, l'inconfort, les besoins d'attention émotionnelle ou des facteurs environnementaux. Il contient également de précieuses informations sur l'état du nourrisson.

Ce projet concerne le développement et la réalisation d'un dispositif basé sur le Raspberry Pi pour détecter les pleurs des bébés et prédire leur cause.

Cette étude décrit une base de données compilée pour analyser la relation entre les pleurs du nourrisson et leurs causes, en utilisant des spectrogrammes qui fournissent une vérité topographique et des informations sur les fréquences fondamentales avec des harmoniques.

Des méthodes de traitement du signal telles que l'analyse prédictive linéaire, la transformée en Ondelette Continu CWT, la transformée en Ondelette Discrète DWT et le MFCC ont été utilisées pour analyser les pleurs du nourrisson et en extraire les caractéristiques énergétiques et les coefficients caractérisant les pleurs de bébés.

Nous essayons de classer les pleurs de bébé en deux catégories telles que la douleur et l'inconfort en utilisant plusieurs méthodes, nous mentionnons les réseaux neuronaux (NN), les réseaux neuronaux convolutifs (CNN) et la machine à vecteurs de support (SVM), nous présentons l'application dans le tableau de bord Node Red pour la partie production.

Un tel dispositif peut être utilisé par les parents (en particulier les parents sourds) ou par une personne qui s'occupe de bébés. Ce type de technologie et de dispositif existe, mais il n'est pas disponible pour une grande population, nous avons donc essayé de le rendre disponible pour différentes groupes de la communauté.

الملخص

بتواصل الرضيع مع بيئته بالبكاء. ينتج البكاء بعد أن يحفزه الألم أو عدم الراحة أو احتياجات الاهتمام العاطفي أو العوامل البيئية. يعكس هذا البكاء أيضًا معلومات قيمة عن حالة الرضيع .الهدف من هذا المشروع هو تصنيع جهاز يعتمد على الحاسوب المصغر الرازبري باي لالتقاط اشارة البكاء ومعرفة السبب .

تتضمن هذه الدراسة تطبيق قاعدة البيانات المجمعة لتحليل العلاقة بين بكاء الرضيع وأسبابه باستخدام مخططات طيفية توفر الحقيقة الطبوغرافية ومعلومات عن الترددات الأساسية مع التوافقيات .تم استخدام طرق معالجة الإشارات مثل التحليل التنبئي الخطي (LP) و CWT و DWT و MFCC لتحليل بكاء الرضع لاستخراج خصائص الطاقة و المعاملات .

قمنا بتصنيف قاعدة البيانات إلى فئات مثل الألم وعدم الراحة واستكشافها بعدة طرق تصنيف مثل الشبكات العصبية (NN) والشبكات العصبية التلافيفية (CNN) والشعاع الحامل للآلة (SVM)، لإنتاج تطبيق في شكل واجهة ويب بواسطة برنامج Node Red للمستخدم .يمكن استخدام مثل هذا الجهاز الذكي من قبل الوالدين (خاصة الآباء الصم) أو من قبل الأشخاص الذين يعتنون بالأطفال (مربية الاطفال، إلخ). هذا النوع من التكنولوجيا موجود ، لكنه غير متاح لعدد كبير من السكان ، لذلك حاولنا إتاحته لمجموعات مختلفة في المجتمع .

Table of content

Figures list

Tables List

Abbreviations list

ADC………………Analogic digital converter

AF .... *audio frequency*

ANN ..... *artificial neural network*

BFCC ..... *Bark frequency cepstral coefficients*

BFS .... *Best Feature Selection*

CFS ... *Correlation-based Feature Selector*

CNN .... *convolutional neural network*

DBL ...... *Dunstan Baby Language*

ECG .... *Electrocardiogram*

EEG .... *Electroencephalogram*

EOG .... *Electro oculogramme*

F.N.S.A *Fédération Nationale des Sourds d'Algérie*

FET .... *: Field-Effect Transistor*

GSM .... *The Global System for Mobile Communications*

HPF:.................... *high pass filter*

IC……………*integrated circuit*

ICSD… .... *Infant Cry Sounds Database*

IoT-BBMS ..... *Internet of Things-Based Baby Monitoring System*

JFET…… *The junction gate field-effect transistor*

KNN .... *k-nearest neighbors*

LDC. *large-diaphragm condenser*

LFCC *Linear Frequency Cepstral Coefficient*

LPC .... *linear predictive coding*

LPCC... *linear predictive cepstral coefficients*

LPF .... *:low pass filter*

LVQ *learning vector quantization*

MBBS ...... *Bachelor of Medicine, Bachelor of Surgery*

MFCC ..... *Mel frequency cepstral coefficients*

Op-amp.... *: operational amplifier*

PCA…………Principal component analyze

PTFE ...... *Polytetrafluoroethylene*

RAM ..... *Random-access memory*

RC ....…………… *Resistor-Capacitor*

RF ....……………….. *radio frequency*

RLC .... *: Resistor-Inductor-Capacitor*

ROM .... *Read-only memory*

S.I .... *systeme international*

SDC *small-diaphragm condenser*

SIDS .... *Sudden Infant Death Syndrome*

SNN .... *Simple Neural Network*

STFT ...... *The Short Time Fourier Transform*

STZC .... *Short time zero crossing*

SVD…………..Singular Value Decomposition

SVM…………Support vectors machines

VQ .... *Vector Quantization*

WEKA . *Waikato Environment for Knowledge Analysis*

Wi-Fi.................. *Wirless Fidelite*

Equations list

General introduction

Human intelligence and his unlimited creativity always faced many challenges during existence to concur the different problems from ages ago until our days searching for solutions.

When it comes to exploring human structure and different physiological signals, sound and hearing mechanism come to take a part. The diversity of hearing degrees in people let humans in general; scientists and engineers work on developing life conditions to have equal chances for living an equal live, especially when it comes to providing better life for deaf community.

Infant cry signal is a biomedical acoustic signal that is usually high-pitched[1]. Infant cry is a mode of communication, for interacting and drawing attention. The infants cry due to physiological, emotional or some ailment reasons[2].

According to world health organization it proves that 466 million people this is over 5% of the world's population suffer from hearing difficulties, and according to our studies, we found that those difficulties have a big influence on their social and economic level[3].

Our project goal is consist to realize a devise or a prototype based on the Raspberry Pi to detect baby cries and predict the cause of it.

This document has four main chapters and in the end, there is a general conclusion and documents support.

The theoretical part is divided in two chapters which we introduced the state of art of other projects and works in the same field with a brief explanation and we give a medical vision, several definitions of the sound and the physics behind it, the hearing anatomy and physiology.

Then, the practical part consists of two chapters, which come to explain our methodology gradually with addressing the implementations and the results obtained to reach our goal. Moreover, the other chapter demonstrates the final design implementation using NODE-RED technology by showing some features results.

Chapter I

State of the art

1. Introduction

Historic researches show that so many scientists are interested in the study of neonate and their emotional state, trying to analyze their actions and cries during growing up mostly from first day born to 24 months, which is the most sensitive phase that parents make sure to have healthy base for their babies.

It must be hard task for parent with hearing difficulties not to be able to tell when the baby is crying and not knowing the reason in often cases it is a quit a challenge, so many researches related to automatic recognition of emotional states[4].

In this first chapter, we will discuss the difference related work done. Therefore, we will begging with the methods of how to detect the sound, which proposing by the four previous works and what are the parts used to capture sound mainly the system realization for collecting data. Then proceed to the work that address audio signal processing and knowing the different classification methods used, followed by integration of deep learning.

Finally proposing our contributions to this research and the different challenges faced, with showing our roadmap for establishing this project.

2. Related work

This part is divided into three parts the first part will handle the different detection systems, the second part about signal processing and feature extraction methods; the third part is the implantation of Deep Learning in speech recognitions.

2.1. Building smart detection systems:

Current time requires adaptable solution there for baby telemonitoring for parents that always occupied with several responsibilities and obliged to be away from their kids. Generally, babies cry because they are hungry, tired, unwell, or need their diaper changed.

Sudden Infant Death Syndrome (SIDS) is also known as crib death, because many babies who die of SIDS, are found in their cribs. It occurs to infants younger than 12 months old.

These mentioned works developed smart solutions are the following:

a. IoT-BBMS (Internet of Things-Based Baby Monitoring System for Smart Cradle)

IoT-BBMS working on a baby monitoring system consisting of a video camera and microphone without limitations of coverage[5].

It can sent data and immediately notify the parents about urgent situations, thereby shortening the time needed to handle such scenarios[5].

So facing this problem with their solutions by adopting the following methodology and contributions:

- A smart baby cradle prototype is designed and fabricated with auto-swinging support, web camera and musical toy.
- A new Algorithm is proposed and implemented in NodeMCU controller to perform the required monitoring and control tasks.
- Utilizing the NodeMCU as the microcontroller and Adafruit MQTT as IoT server to retrieve data from sensors and send commands to actuators.

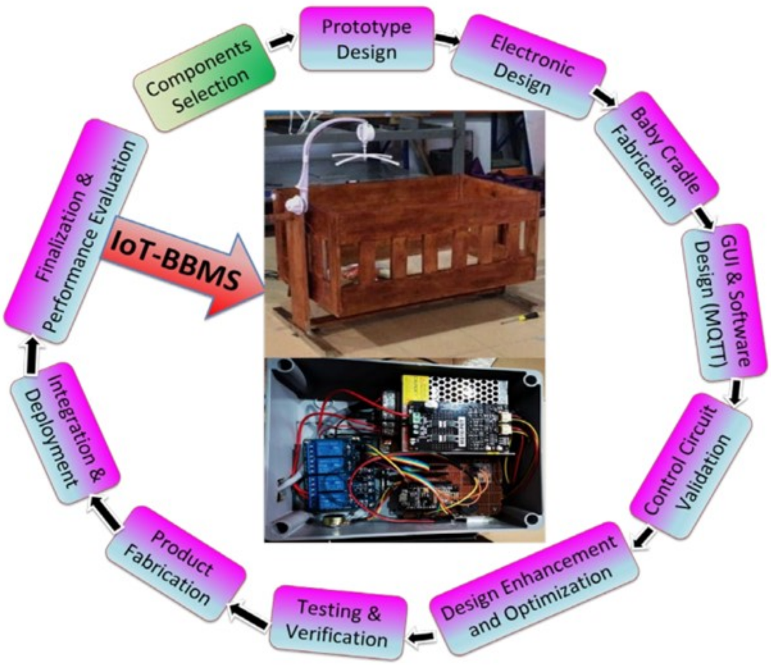


Figure 1: Flowchart of research activities[5].

A temperature and humidity sensor, DHT22, is used in the proposed system to measure the room temperature. This system consisted of a four-channel relay module, which is controlled by the microcontroller.

When the condenser MIC detected sound from the baby, a signal is sent to NodeMCU. NodeMCU switches the relay that is connected to a DC motor, which in turn is connected to the cradle for swinging purposes.

The temperature and humidity sensor measure the room temperature, records the readings in the NodeMCU, and uploads the readings to the MQTT server at the same time.

If the room's temperature exceeded a certain temperature, then the NodeMCU switches on the relay connected to the mini fan to lower the temperature and cool the baby, as well as prevent overheating.

An external Wi-Fi camera is used in this system to allow the parent to monitor the real-time baby condition however, this system is smart but is it costly for parents and Real-time monitoring can be very hard with different IP address network.

b. **Design and Development of a Smart Baby Monitoring System based on Raspberry Pi and Pi Camera**

The designed system can spot the motion and crying condition of the baby.

They used condenser MIC to spot the crying condition and PIR motion sensor to spot the baby movement with the help of Pi camera. The camera is turned on only when the condenser MIC detects a sound and sends a signal to Raspberry Pi.[6]

However, the output of this system is only available on monitor display; thus, the parents can only view the data on a limited number of devices within a fixed area.[6]

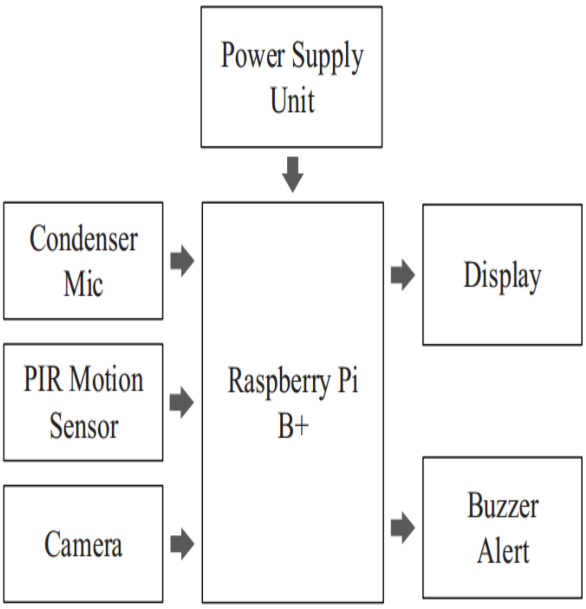


Figure 3: Block Diagram of the proposed System[6]

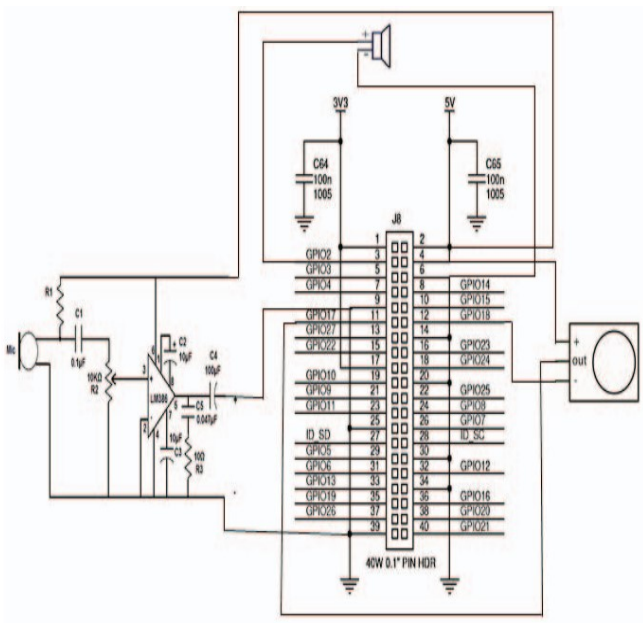


Figure 2: Circuit diagram of baby monitoring system[6]

c. **Smart Cradle for Baby Using FN-M16P Module**

This work cites the Infant Monitoring System with Real-Time Alerts to Parents using GSM.

Activity Monitoring includes Infant’s cry detection, which makes the cradle swing with pre-recorded voice play and alert to parents for their intimidation. [7]

Cry detector system, which detects the baby cry by an M213 sound sensor based on ECM and sends the information to the microcontroller (Renesas micro-controller has been used in this project. ROM: 512 KB, RAM 32 KB, Data flash memory: 8 KB), after a time interval of three counts and it swings the cradle until the baby stops crying. The swinging action is due to DC motor. [7]

The user can control the speed of the cradle as per his need and if the cry last for 5 counts then the mother voice is played by FN-M16P module will have recorded voice input of the mother and plays it whenever the baby cries for a long time. [7]

The ultrasonic and accelerometer sensors are used for detecting respiratory and non-respiratory movements of the baby.

- **Respiratory movement:** An ultrasonic sensor is used to detect the breathing of infant. When baby suffer from Apnea, information will be sent to parent by micro-controller via GSM.
- **Non-respiratory movement:** It is a secondary process, which plays a major role, if the respiratory part does not work properly. Accelerometer is connected to the wrist or ankle of the infant that will help to detect the infant movement with respect to the x-axis and y-axis. When there will be no movement it will be detected, and alert will be sent.

The model would decrease the existing complex methods of monitoring where typically involves (Electroencephalogram) EEG, Electro-oculogramme (EOG), two or three lead chest Electrocardiogram (ECG) were used to monitor are intrusive procedures and not well tolerated by infants and elderly. [7]

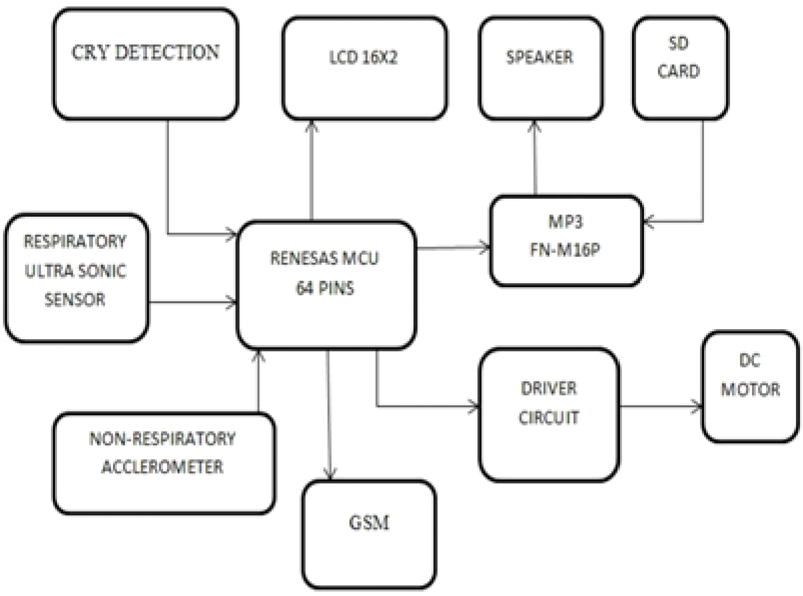


Figure 4: Smart cradle Block Diagram [7]

d. **Smart Baby Cradle**

The system proposed is designed to help parents and nurses in infants care with low cost then the other solutions. Cradle starts swinging automatically when baby cry and swings till the baby stops crying[8].

Then sounds an alarm when mattress gets wet, sounds an alarm if baby cries for more than a stipulated time indicating that baby needs attention.

The sensors, including the three-pivot accelerometer and infrared sensor, are tried and connected under swinging sufficiency control and processed by ARDUINO UNO the swaying is automated by **Artificial Metabolic Algorithm** which is a method for calculating Labor power to control the gadget and work effectively. [8]

The flag acknowledgment circuit likewise decides whether the zero-crossing rate of the beat motion over the whole time interim is more noteworthy than or equivalent to a second low limit however less than or equivalent to a third high limit. [8]

On the off chance, which these two conditions are satisfied. At that point, the flag acknowledgment circuit yields a flag demonstrating that an infant's cry was distinguished.

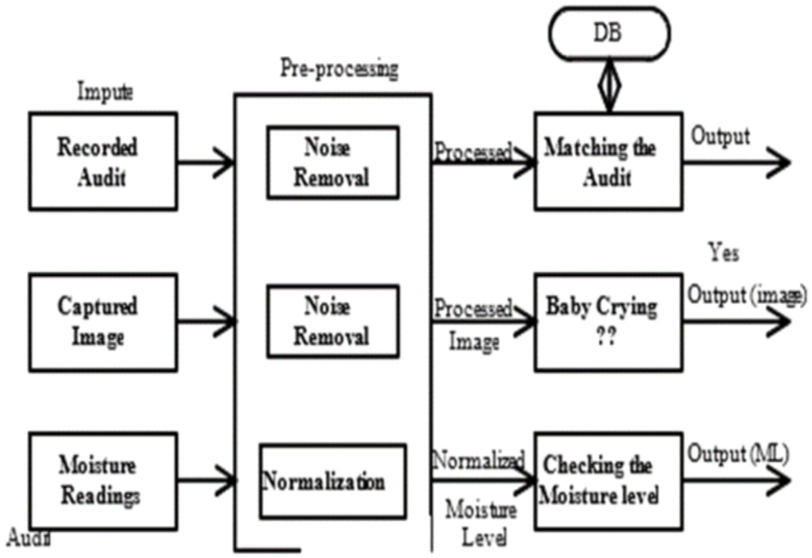


Figure 5: Block diagram[8]

2.2. Signal processing and classification:

Infant cry signal is a biomedical acoustic signal that is usually high-pitched. Infant cry is his only means of communication, production of which involves significant changes in its characteristics.

Possibly rapid changes in its short-time segments carry information about the cause of cry, which a mother can perceive.[1]

a. Discriminating Features of Infant Cry Acoustic Signal for Automated Detection of Cause of Crying

In this study, we describe an 'Infant Cry Sounds Database' (ICSD), collected especially for the study of likely cause of an infant's cry. [1]

The database consists of infant cry sounds due to six causes: pain, discomfort, emotional need, ailment, environmental factors and hunger/thirst. The ground truth cause of cry is established with the help of two medical experts and parents of the infants.

The Short Time Fourier Transform (STFT) as preliminary analysis is carried out using the sound production features from Spectrograms give the base reference to the instantaneous fundamental frequency and frame energy derived from the cry acoustic signal, using auto correlation and linear prediction (LP) analysis to extract the coefficients that are computed using the least squares method, that minimizes the LP residual or total prediction error. [1]

In both methods, the signal was divided into frames of 10ms, with a shift of 6ms. The F_0 contour was smoothed further, using a binary mask derived using the signal energy for each frame and let them arrive to the ground truth.

In this figures, [1] we show sample of their results and that they used both methods to have more efficient results for pain features:

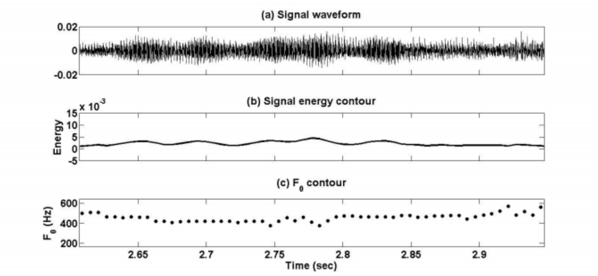


Figure 6: Illustration of (a) signal waveform, (b) signal energy contour and(c) F_0 contour of Pain Cry of infant #1 (using auto-correlation) [1]

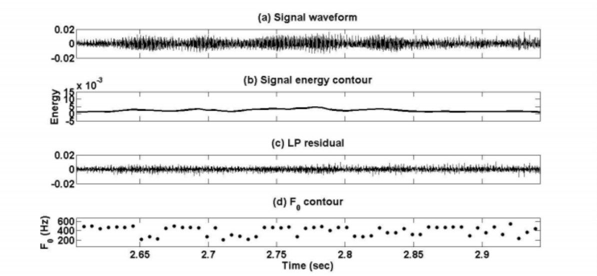


Figure 7: Illustration of (a) signal waveform, (b)signal energy contour, (c) LP residual and (d) F0 contour of Pain Cry of infant #1 (using LP analysis) [1]

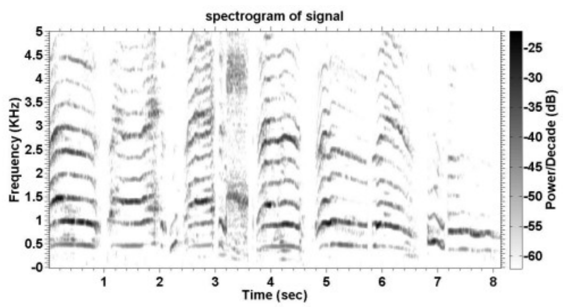


Figure 9: Spectrogram of Pain Cry of Infant #1. [1]

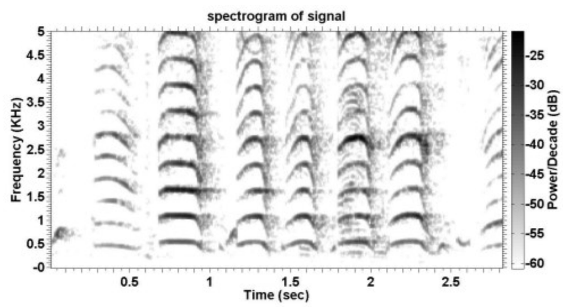


Figure 9: Spectrogram of Pain Cry of Infant #2. [1]

Note: They are planning to use different signal processing methods in the future, and we are planning to use their database in our work, which will be explained afterwards.

b. Infant Cry Signal Detection, Pattern Extraction and Recognition

The speech signal identification technique to recognize infant cry signals. Advanced signal processing methods are used to analyze the infant cry by using audio features in the time and frequency domains are Short-Time Energy and Short time zero crossing (STZC) for classifying each cry to a specific need[9].

The features extracted from audio feature space include linear predictive coding (LPC), linear predictive cepstral coefficients (LPCC), Bark frequency cepstral coefficients (BFCC) and Mel frequency cepstral coefficients (MFCC)[9]. The primary classification technique used were nearest neighbor approach, neural networks method. The cry recognition of specific infants yielded promising results.

Their database collected at china hospital containing 30 recording includes 8-draw attention cry wave files, 6-diaper change needed cry wave files, and 16 hungry cry wave files. Recording sampling frequency is 44.1 kHz each[9].

This are some features classification results shown in the table below:

Table 1: Infant cry recognition correct rate by using different work.

Features	LPC	LPCC	MFCC	BFCC
Nearest neighborhood	63.84	47.95	63.89	65.22
ANN LVQ	54.55	51.88	60.45	76.47

Highest recognition correct rate for infant's cry application is achieved at 76.47% by using BFCC-10 feature[9].

The simulation results show that classification rate is around 70%. It is believed that there are patterns that distinguish the meaning of infant cries.

c. Automated Baby Cry Classification on a Hospital-acquired Baby Cry Database

This work brings us closer to real-time automatic baby cry recognition, as not only baby crying recognition accuracy is increased, but also both computational complexity and computation time are vastly decreased[10].

Altering the complex parameters of the machine learning algorithms further than the baseline values could be the basis of future research, but will likely require variations for each cry type using[10].

For feature extraction they used The Munich open Speech and Music Interpretation by Large Space Extraction (open SMILE) tool enable extraction of large audio feature spaces in real-time, allowing for various configurations for distinct purposes, such as speech emotion recognition, sincerity recognition, etc.[11] [12].

They applied machine-learning algorithms using WEKA[13], an open-source Java application developed by the University of Waikato in New Zealand.

Best Feature Selection or BFS is a technique that aims to reduce the dimensionality of the feature vector by excluding all the features that are irrelevant to the classification task while selecting the minimum subset of features embodying the relevant properties of the dataset.

- The first approach (Corr) computes the Pearson correlation coefficient between each feature.
- The second approach (InfoGain) computes the entropy for each feature for the output variable.

Lastly, using the Correlation-based Feature Selector (CFS) and the Best-First Search Method introduced by Hall[14]. The CFS algorithm is able to rank feature subsets according to the degree of correlation.

Baby Language (DBL) that has some meaning like “I am hungry”, “I am sleepy”, and others. The baby language is grouped into five meanings that use as universal language of babies[14].

Dunstan Baby Language (DBL) is through three main stages, the first is pre-processing to normalize all sound data, the second is feature extraction, and the last one is classification process. The popular feature extraction methods for audio processing are Mel Frequency Cepstral Coefficient (MFCC) and Linear Frequency Cepstral Coefficient (LFCC), they use sound frequency based[14].

For the classification process, there are so many methods that have been used, but in this study, KNN classification, Vector Quantization (VQ) and Simple Neural Network (SNN) are analyzed to inform the appropriate condition when using this method combination from feature extraction to the classification[14].

2.3. Baby Cry Detection in Domestic Environment using Deep Learning

Reading this work, we saw that they propose two machine-learning algorithms for automatic detection of baby cry in audio recordings.

The first algorithm is a low-complexity logistic regression classifier, used as a reference[15].

To train this classifier, we extract features such as Mel-frequency Cepstrum coefficients, pitch and formants from the recordings.

The second algorithm uses a dedicated convolutional neural network (CNN), operating on log Mel-filter bank representation of the recordings[15].

Performance evaluation of the algorithms is carried out using an annotated database containing recordings of babies (0-6 months old) in domestic environments[15].

They plan to train a CNN classifier to detect various types of domestic sounds in addition to cry signals.

Table 2: A summary of the false-positive rates for a given detection rate between the two classifiers.

Classifier / Detection rate	80%	85%	90%	95%
Logistic Regression	0.9%	2,1%	4,3%	9,0%
CNN	0,7%	1,6%	4,2%	12,0%

Note: The classification results are very satisfying that ensure that CNN is the best for such a method

3. Work contribution

The contribution of this project (A system used for cry interpretation for the deaf parents) is to combine the system of the cry detection with a program or software to achieve the interpretation of the cry to help the parent en general and the deaf parent in particular.

We are using the Raspberry Pi with microphone via digital converter (Sound card) to detect and read the input data (Baby cries) and testing several models as CNN using spectrogram images, NN using multi layers perceptron, SVM using continuous wavelet transform for the signal processing.

Then integrate machine-learning model to have the cry prediction using NODE-RED interface and dashboard to display results.

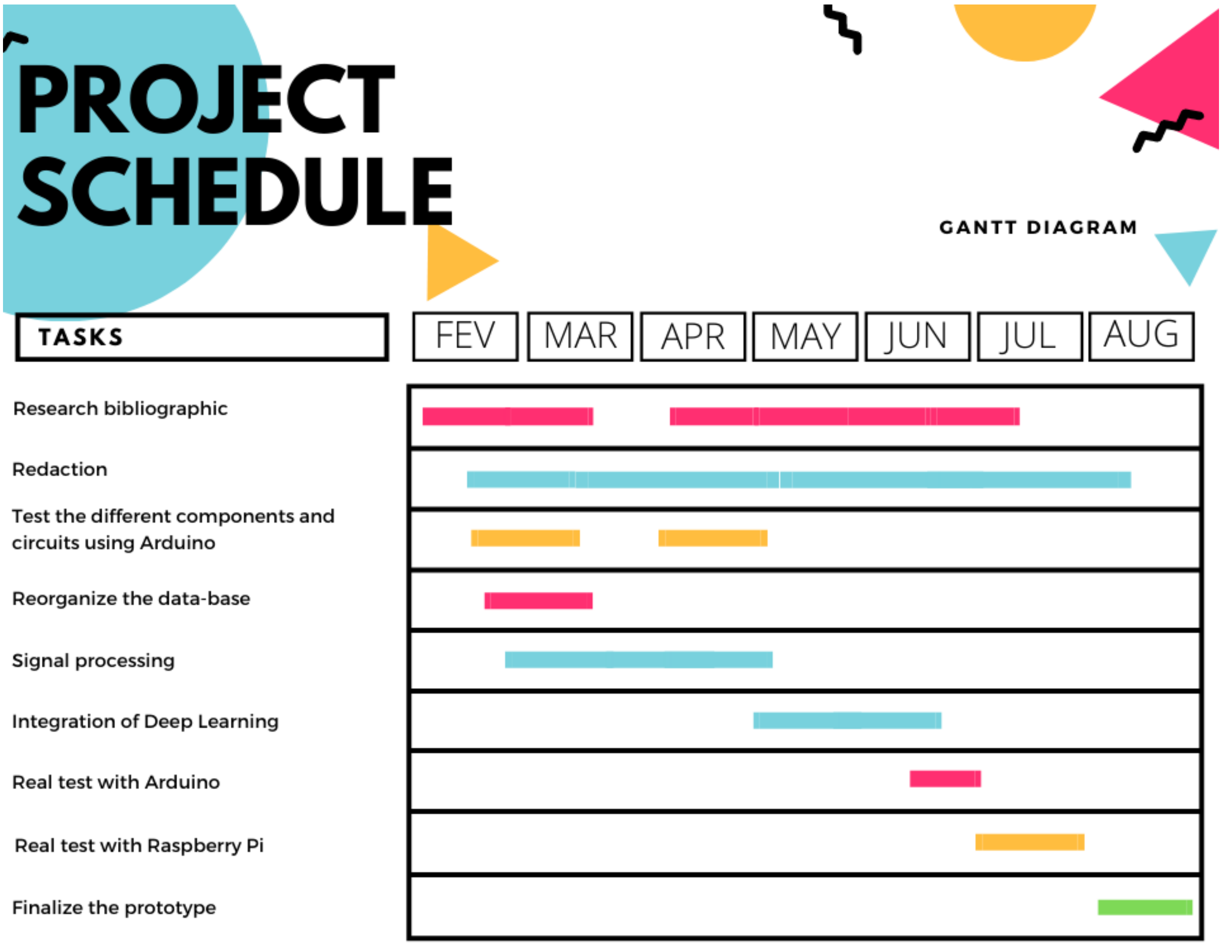
To achieve this project, we have to pass by:

- Testing phase include the use of Arduino Uno connected to electronic circuit to tests the acquisitions of the sound.
- Analyzing the data for preparation of machine learning system.
- The signal processing, in this part, we are testing several methods using PYTHON as our framework and in the classification part, we compare between the CNN, NN and SVM methods to choose the suitable method for our application.
- The deep learning model implanted in the detection device to detect the cry with mentioning the type of the cry to notify the parents or the babysitter.

In addition, our goal is to realize a finished product from all the angles (Performance and design).

The devise can be controlled via a Wi-Fi or Bluetooth if the parents allow it, if else via GSM or even hotspot.

4. Gantt diagram



5. Roadmap

This work is composed of four chapters ending with technical support paper, summarizing the work as following:

In the first chapter, we introduced the state of art of the other projects and works in the same field with a brief explanation about the principles and techniques used to realize these sorts of projects.

The second chapter is about several definitions of the sound and the physics behind it, the hearing anatomy and physiology. In addition, we will address the impact of deafness on the child, parent and the society, without forgetting the communities that help the harmed ones.

In the third chapter, we will explain our methodology gradually, passing from the testing phase to the realization of the final product using deferent techniques with addressing the implementations and the results obtained to reach our goal.

In the fourth chapter working on final design implementation using NODE-RED technology by showing some features results. The transmission of the information or the result to the parents or the babysitter will be using one of the different techniques (Wi-Fi, Bluetooth or GSM).

Finally, in the end, there is a general conclusion showing results and documents support.

Chapter II

Definition and physics of sound

1. Introduction

Sound is the heart of speech communication. A sound wave is the end product of the speech production mechanism and the primary source used by the listener to recover the speaker's message.

Because of the central role of sound in verbal communication (Speech), it is important to have a good understanding of how sound is produced, changed and measured[16].

The purpose of this chapter will be:

- To review some of the basic principles of the physics of sound, especially the ideas that play a particularly important role in speech and hearing. The mechanism of speech generation is an assembly line that works by producing some relatively simple sounds.
- To see, a critical step at the receiving end occurs, when the ear breaks down this complex sound into individual frequency elements.
- To know more about hearing loss and the communities that help people who suffer from deafness.
- To show our survey for this dilemma.

Before getting into these ideas, it is first necessary to cover the basic principles of vibration and sound propagation.

2. Sound definition

In physics, sound is a vibration that propagates as an acoustic wave, through a transmission medium such as a gas, liquid or solid[17].

In human physiology and psychology, sound is an audible mechanical wave propagating through matter, or the perception of such waves by the brain. [18]

Only acoustic waves that have frequencies lying between about 20 Hz and 20 kHz elicit an auditory percept in humans. Sound waves above 20 kHz are known as ultrasound and are not audible to humans. Sound waves below 20 Hz are known as infrasound as shown in figure 10. Different animal species have varying hearing ranges[17].

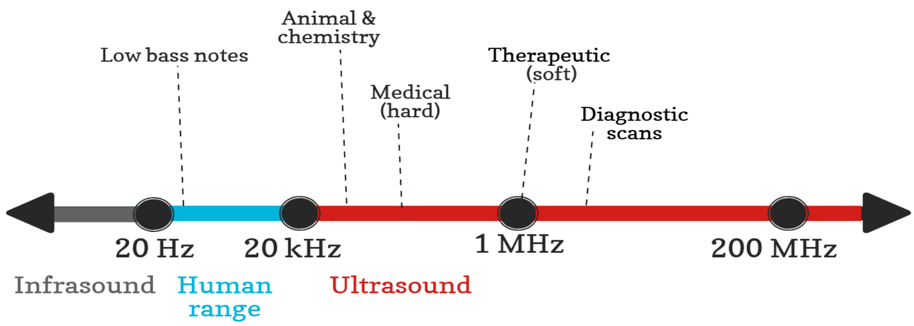


Figure 10: Diagram of sound waves. [17]

3. Sound and Vibration

A sound wave is an air pressure disturbance that results from vibration. The vibrations can come from a guitar string, an air column in an organ pipe, a speaker's diaphragm, a vocal cords, or almost anything that vibrates in the frequency range that the listener can hear (for humans, it's about 20 to 20,000 vibrations per second). [16]

Two conditions are required for the production of sound waves, one is vibrational interference and the other is an elastic medium, the most familiar of which is air. We first introduce the properties of vibrating objects and then look at what happens when vibrational motion occurs in an elastic medium such as air.[16]

4. Physics of sound

4.1. Propagation of sound

Sound is a series of pressure waves that propagate through a compressible medium such as air or water. (Sound can also be transmitted through solids) [17] .

During propagation, waves are reflected, refracted or attenuated by the medium. All media have three properties that affect the behavior of sound transmission[17] :

- The relationship between density and pressure. This relationship, influenced by temperature, determines the speed at which sound travels through the medium.
- The motion of the medium itself, such as **wind**, etc. Unrelated to the movement of sound through the medium, if the medium is in motion, the sound propagates further through the medium.
- The Viscosity of the medium. This determines the speed at which the sound is attenuated. For many media, such as air or water, the attenuation due to viscosity is negligible.

When sound moves in a medium that does not have constant physical properties, it may be refracted (dispersed or focused).

Mechanical vibrations that can be interpreted as sound can pass through all material forms: gas, liquid, solid and plasma. The substance that supports sound is called a medium. Sound cannot travel through a vacuum[17].

4.2. Longitudinal and transverse waves

There are two basic types of waves, transverse and longitudinal. Differentiated by the way of the wave is propagation.

In transverse waves (figure 11) is generated by electromagnetic sources such as light or radio, in which the electric and magnetic fields that make up the wave oscillate in a direction perpendicular to its propagation.

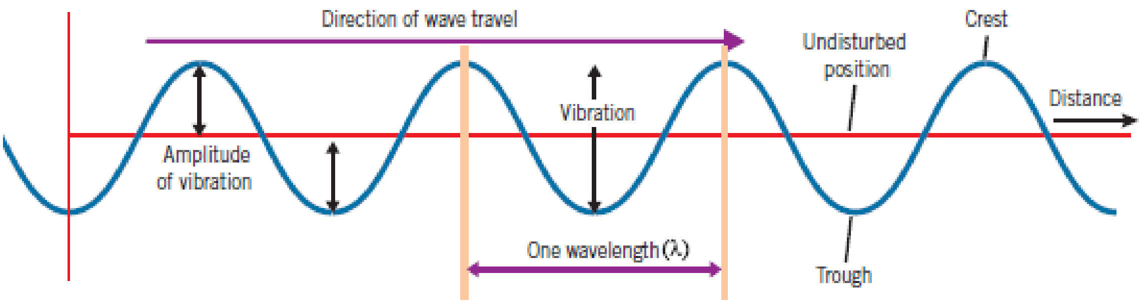


Figure 11: A representation of a transverse wave [19]

A longitudinal wave (figure12) can be created by squeezing several turns of the spring to form a compression, which is then released so that the compression travels along the length of the spring.

Sound propagates through air or other media in the form of longitudinal waves, where mechanical vibrations that form the wave occur along the direction of propagation of the wave [19].

Sound waves are regions of high and low pressure moving at a certain speed. It consists of a periodic (i.e., oscillatory or vibrational) pressure change that occurs around an equilibrium pressure at a particular time and place.

Equilibrium pressure and pure sound waves (i.e., waves of a single frequency) pass through the resulting sinusoidal wave changes [19].

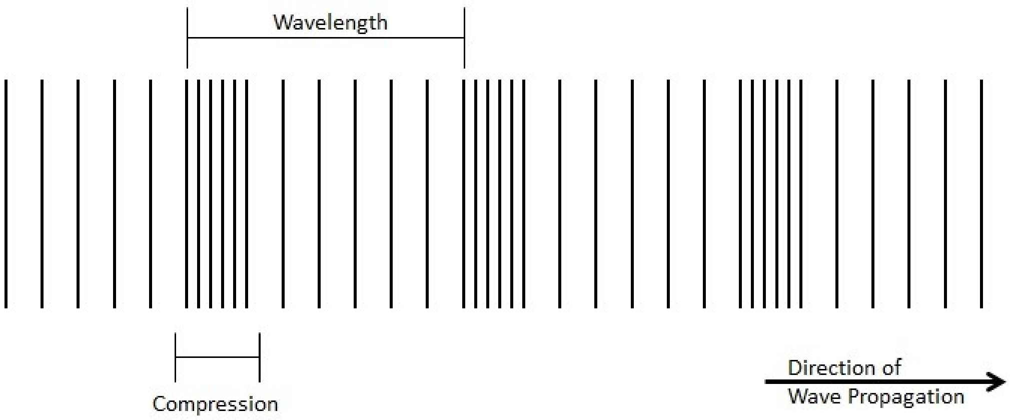


Figure 12 A representation of a longitudinal wave[19]

4.3. Sound wave properties and characteristics

Sound wave can be described by five characteristics: Wavelength, Amplitude, Time-Period, Frequency and Velocity or Speed[20].

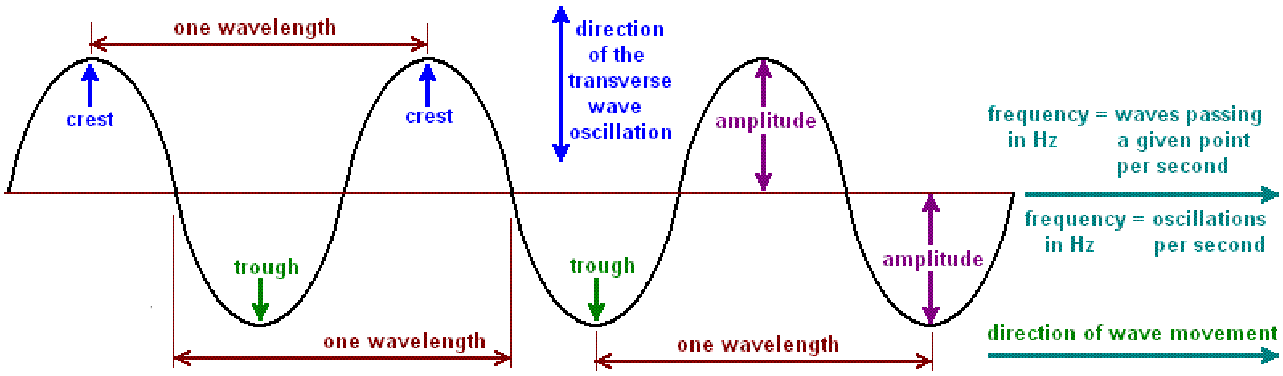


Figure 13: Wave characteristics[20]

4.3.1. **Wavelength**

The minimum distance in which a sound wave repeats itself is called its wavelength. That is it is the length of one complete wave. It is denoted by a Greek letter λ (lambda).

We know that in a sound wave, the combined length of a compression and an adjacent rarefaction is called a wavelength. Also, the distance between the centers of two consecutive compressions or two consecutive rarefactions is equal to its wavelength[20].

Note: The distance between the centers of a compression and an adjacent rarefaction is equal to half of its wavelength i.e. $\lambda/2$. The **S.I** unit for measuring wavelength is meter (m).

4.3.2. **Amplitude**

When a wave passes through a medium, the particles of the medium are displaced temporarily from their original undisturbed positions. The maximum displacement of the particles of the medium from their original undisturbed positions, when a wave passes through the medium is called amplitude of the wave. [20]

In fact the amplitude is used to describe the size of the wave. The **S.I** unit of measurement of amplitude is meter (m) though sometimes it is also measured in centimeters.[20]

4.3.3. **Time-Period**

The time required to produce one complete wave or cycle or cycle is called time-period of the wave. Now, one complete wave is produced by one full vibration of the vibrating body. Therefore, we can say that the time taken to complete one vibration is known as time-period. It is denoted by letter T. The unit of measurement of time-period is second (s) [20].

4.3.4. **Frequency**

The number of complete waves or cycles produced in one second is called frequency of the wave. Since one complete wave is produced by one full vibration of the vibrating body, so we can say that, the number of vibrations per second is called frequency[20].

For example: if 10 complete waves or vibrations are produced in one second then the frequency of the waves will be 10 hertz or 10 cycles per second.

The **S.I** unit of frequency is hertz or Hz. The frequency of a wave is denoted by the letter f.

The frequency of a wave is the same as the frequency of the vibrating body, which produces the wave[20].

4.3.5. Velocity of Wave (Speed of Wave)

The distance travelled by a wave in one second is called velocity of the wave or speed of the wave. It is represented by the letter v. The **S.I** unit for measuring the velocity is meters per second (m/s or ms-1) [20].

d. The relation between time-period and frequency of a wave

The time required to produce one complete wave is called time-period of the wave. Suppose the time-period of a wave is T seconds[20].

In T seconds number of waves produced = 1 ; so, in 1 second, number of waves produced will be = 1/T.

However, the number of waves produced in 1 second is called frequency.

Therefore, f = 1/Time-period; f = 1/T, Where f = frequency of the wave, T = time-period of the wave.

e. The relationship between Velocity, Frequency and Wavelength of a Wave

Velocity = Distance travelled/ Time taken

Let $v = \lambda / T$, Where T = time taken by one wave.

$v = f * \lambda$: This formula is known as wave equation.

Where v is velocity of the wave, f is frequency and **λ** is the wavelength.

Velocity of a wave = Frequency * Wavelength.

This applies to all the waves like transverse waves like water waves, longitudinal waves like sound waves and the electromagnetic waves like light waves and radio waves[20].

4.3.6. Other Terminology

a. Pitch

Pitch is the perception of the frequency of sound waves—i.e. The number of wavelengths passing through a fixed point in a unit time [19].

b. Loudness

Loudness refers to the intensity of the sound, i.e. the pressure that the sound waves exert on the eardrum. The greater the amplitude or intensity of a sound wave, the greater the pressure or intensity of the sound and therefore the louder the sound. The intensity of sound is measured and reported in decibels (dB)[19].

Note: A decibel is a unit that represents the relative size of a sound in logarithms.

5. Human voice

In the vowels produced by the human voice, the stressed harmonic group, the so-called "consonant", plays a crucial role.

The vowel comes from the resonance in the vocal cords. The average length of the human vocal cords is about 17.5 cm, with the lower end at the vocal folds and the upper end at the lips. Like a reed or like a lip at the mouthpiece of a wind instrument, the vocal fold plays an acoustically closed role, so the vocal column is a closed tubular resonator with resonance frequencies of about 500 Hz, 1500 Hz, 2500 Hz, 3500 Hz, etc. [19]

The vibration frequency of the vocal folds, determined by the folds' tension, determines the frequency of the vocal sound. When a sound is produced, all harmonics are present in the spectrum, but those near the resonant frequencies of the vocal column are increased in amplitude. These emphasized frequency regions are the vocal formants. By changing the shape of the throat, mouth, and lips, the frequencies of the formants are varied, creating the different vowel sounds.[19]

6. Baby cries

Newborns are communicating by crying. In this way, newborns express their physical and emotional states and express their needs. Hunger, tiredness, pain, discomfort, aches, colic, discharge, bloating, need for attention, etc. are the main reasons for a baby to cry. [21]

Studies have shown that there are different types of cries depending on the needs of the newborn, such as hunger, pain, discomfort, etc. A neonatologist or pediatrician can distinguish between different types of cries and can find patterns in each type of cry [21].

The cry of a newborn is a special condition of human speech, and like human speech, the cry is a short-term stationary signal. Since newborns do not have full control of the vocal tract, cry signals are considered to be more stationary than speech signals[22].

Compared to the production of adult speech, it can be demonstrated that crying is again a complex neurophysiological behavior.

It is the result of an intense expulsion of air pressure in the lungs, causing oscillations of the vocal cords and leading to the formation of sound waves[21].

7. Hearing

Hearing is the process by which the ear translates sound vibrations in the external environment into nerve impulses and transmits them to the brain, where they interpreted as sound.

Sound is when vibrating objects, such as strummed strings on a guitar, produce pulses of pressure from vibrating air molecules, known as sound waves.

The human ear can distinguish different subjective aspects of sound, such as loudness, pitch, etc., by detecting and analyzing different physical properties of sound waves.

The ear is most sensitive and perceptible to frequencies between 1000 and 4000 Hz, but at least for normal young ears, the entire audible range of sound is between about 20 and 20,000 Hz[23].Sound waves of still higher frequency are referred to as ultrasonic, although other mammals can hear them.

In order for a sound to be transmitted to the central nervous system, the energy of the sound undergoes three transformations.

First, the vibrations in the air are transformed into vibrations in the tympanic membrane of the eardrum and middle ear and in the cochlea. These vibrations in turn become vibrations of the fluid in the cochlea.

Finally, the vibration of the fluid builds up a motion wave along the basal membrane that stimulates the hair cells of the Corti organ. These cells convert the vibrations of sound into nerve impulses in the nerve fibers of the cochlea, which are then transmitted to the brainstem, where they are heavily processed and then relayed from the brainstem to the primary auditory area of the cerebral cortex, which is the ultimate center of hearing in the brain[23].

Only when the nerve impulses reach this area does the listener become aware of the sound.

7.1. Anatomy of auditory system

The anatomy of our auditory system (figure 14) is extremely complex, but can be roughly divided into two parts, one called the “peripheral” auditory system and the other called the “central” auditory system[24].

The peripheral auditory system consists of three parts, the outer ear, middle ear and inner ear[24]:

- The outer ear consists of the auricle (also known as the pinna), the ear canal and the eardrum.
- The middle ear is a small, air-filled space that contains three small bones called the malleus and incus, but collectively called the ossicles. The large ear bone (the malleus) is connected to the tympanic membrane (the eardrum), the outer ear to the tympanic membrane.
- The inner ear has hearing and balance organs. The auditory part of the inner ear is called the cochlea because of its unique curly shape. There are thousands of sensory cells (called hair cells) in the cochlea that is connected to the central hearing system by the hearing or auditory nerve. The cochlea is filled with special fluids that are important to the hearing process.

The central auditory system consists of the auditory nerve and an extremely complex pathway that runs through the brain stem and down to the auditory cortex of the brain[24].

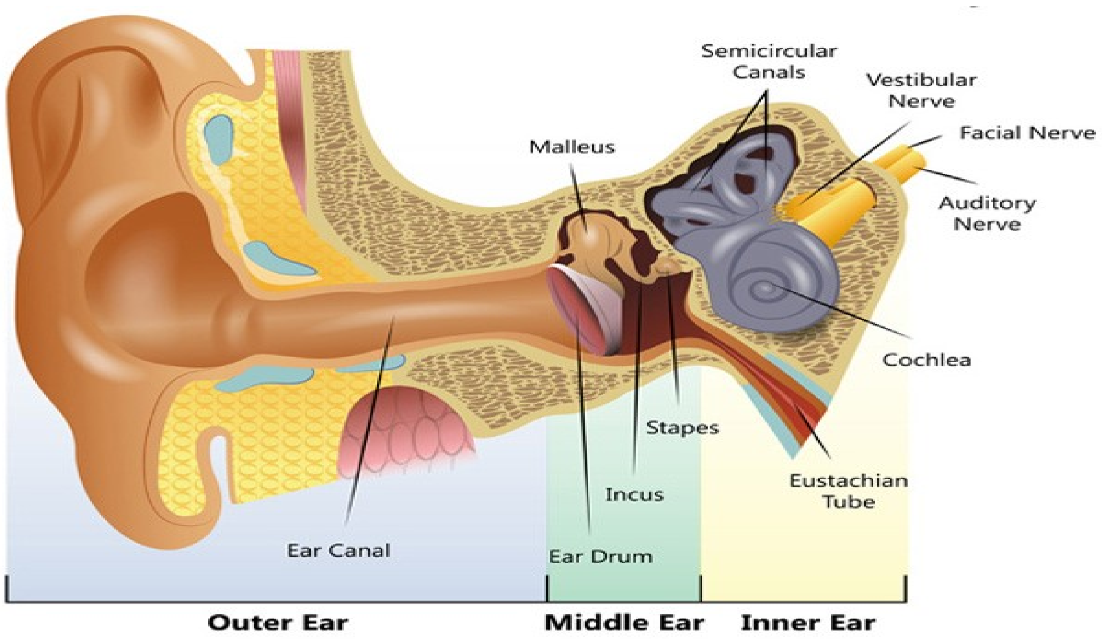


Figure 14: Diagram of the main parts of the peripheral hearing system. [24].

7.2. Physiology of hearing

The physiology of hearing, like its anatomy, is indeed very complex and for best understanding, we must take a look at the roles played by the various parts of the auditory system mentioned above[24].

Sound waves, which are actually vibrations in the air around us, are collected through the needles on either side of our head and funneled into the ear canal. These sound waves cause the tympanic membrane to vibrate.

The eardrum is so sensitive to sound vibrations in the ear canal that it can detect even the faintest sounds and reproduce even the most complex sound vibration patterns[24].

The tympanic membrane vibrations move the chain of small bone in the middle ear (the ossicles – malleus, incus and stapes), sending sound vibrations into the cochlea in the inner ear.

This is because of the stapes, the last of the three bones in this chain, are located in a periosteal-covered window in the bone wall that separates the middle ear from the cochlea in the inner ear[24].

When the stapes vibrate, it causes the fluid in the cochlea to move in a wave-like manner, stimulating microscopic hair cells.

It is worth noting that the hair cells in the cochlea respond to different sounds depending on the pitch or frequency of the sound. High-pitched sounds stimulate hair cells in the lower cochlea and low-pitched sounds stimulate hair cells in the upper cochlea[24].

What happens next is even more remarkable because when each hair cell detects the pitch or frequency of the sound to which it is responding, it generates nerve impulses that move instantaneously along the auditory nerve[24].

These nerve impulses travel along complex pathways in the brainstem before reaching the brain's auditory nucleus, the auditory cortex. Here, the nerve impulse flow is transformed into meaningful sound.

All of this happens in almost a fraction of a second after the sound waves first enter our ear canals. It is quite true to say that we end up listening with our brains[24].

8. Deafness and hearing loss

There are two main types of deafness (partial or complete inability to hear), conductive deafness and nerve deafness. Conductive deafness refers to the disturbance of sound vibrations as they travel from the outer ear to the nerve cells of the inner ear[25].

The obstruction may be that the earwax is blocking the auditory canal in the outer ear, or the stapes fixation so that the stapes (one of the small bones in the middle ear) cannot transmit sound vibrations to the inner ear.

In nerve deafness, certain defects in the sensory cells of the inner ear (e.g., damage caused by excessive noise) or in the vestibular cochlear nerve prevent the transmission of sound impulses from the inner ear to the auditory center of the brain. Deafness at birth is usually neurotypical and cannot be improved by medical means[25].

There are 466 million people in the world with disabling hearing loss. This is over 5% of the world’s population; 34 million of these people are children, 60% of childhood hearing loss is preventable through public health actions[3].Nearly one out of every three people over 65 years are affected by disabling hearing loss left untreated.

Hearing loss can lead to people being excluded from the most basic communication, thereby contributing to feelings of loneliness, frustration and social isolation. Hearing loss in the elderly is linked with early cognitive decline and dementia. Age-related hearing loss can be managed effectively through a variety of means, including hearing aids[3].

It is estimated that 1.1 billion people (aged between 12-35 years) are at risk of developing hearing loss due to noise exposure.

Unless action is taken, by 2030 there will be nearly 630 million people with disabling hearing loss. By 2050, the number could rise to over 900 million[3].

Note: Disabling hearing loss refers to hearing loss greater than 40dB in the better hearing ear in adults and a hearing loss greater than 30dB in the better hearing ear in children.

9. Impact of hearing loss

9.1. Functional impact

One of the main impacts of hearing loss is on the individual's ability to communicate with others. Spoken language development is often delayed in children with unaddressed hearing loss[3].

Unaddressed hearing loss and ear diseases such as otitis media can have a significantly adverse effect on the academic performance of children. They often have increased rates of grade failure and greater need for education assistance. Access to suitable accommodations is important for optimal learning experiences but are not always available[3].

9.2. Social and emotional impact

Exclusion from communication can have a significant impact on daily life, causing loneliness, isolation and frustration, especially among older people with hearing loss.

Parents with hearing impairments face unique challenges, such as finding a daycare program or nanny (babysitter) who can communicate with them and their children. Other challenges come from parents' inability to hear [26].

For example, a day care provider who was caring for a hearing child of a deaf parent noticed a tendency for the child to scream or yell. In her letter, she asked whether it was common for hearing children of deaf parents to shout loudly on a regular basis.

Another problem for deaf parents of hearing children is that the children may take advantage of the fact that their parents cannot hear them. This issue came up in the blog post "Deaf Parents with Unruly Hearing Children". In that post, a teacher commented that her students who had deaf parents were misbehaving and taking advantage of their parents' deafness[26].

9.3. Economic impact

WHO estimates that unaddressed hearing loss costs the world $750 billion annually, including costs to the health sector (excluding the cost of hearing equipment), educational support costs, lost productivity and social costs[3].

In developing countries, children with hearing loss and deafness rarely receive schooling. The unemployment rate for adults with hearing loss is also much higher. Among the employed population, a higher proportion of people with hearing loss have a lower level of employment compared to the general workforce [3].

Increased access to education and vocational rehabilitation services and raising awareness of the needs of employers, especially those with hearing loss, will reduce the unemployment rate for people with hearing loss[3].

10. Deaf communities

10.1. The World Federation of the Deaf

The World Federation of the Deaf is one of the oldest international organizations of persons with disabilities in the world , recognizing the barriers to full accessibility, equal human rights and participation in decisions affecting deaf people around the world [27].

The World Federation of the Deaf was founded in Rome, Italy, on 23 September 1951. At the First World Congress, only representatives of deaf associations from 25 countries attended. Today, our ordinary members represent 135 countries on five continents and it has been working with the United Nations and its agencies since the late 1950s [27].

The WFD has played an important role in advocating and developing the United Nations Convention on the Rights of Persons with Disabilities (CRPD) [27]. It remains actively involved in the CRPD implementation, monitoring, and promoting its ratification by Member States.

- Vision: Human rights for deaf people, including recognition of sign language in all aspects of life.
- Mission: The WFD works for the realization of deaf people's human rights in partnership with the United Nations and its agencies, national organizations of deaf people, and relevant stakeholders.
- Values: Respect, Equality, Diversity, Impartiality and Professionalism.

10.2. National Federation of the Deaf of Algeria (F.N.S.A.)

The National Federation of the Deaf of Algeria, in French “Fédération Nationale des Sourds d'Algérie”, is an Association created in 1979 and was approved on 14 March 1981 by the Ministry of the Interior.

It is represented in all the Wilayas through the Unions of Deaf-Mutes whose approvals are provided by the Walis[28].

National Federation of the Deaf in Algeria's official or approximate number of Deaf people: 240,000(WFD. 2008. Global Survey Report)[29].

The F.N.S.A. works to develop solidarity actions to achieve the different goals:

- **Employment:** A hearing-impaired person who is employed can consider himself or herself to be integrated into society.
- **Information:** Heavy media should take over the communication to the deaf community.
- **Training:** Develop sign language for all deaf and hearing people.
- **Special education:**
    - Organization of seminars in collaboration with specialized institutions.
    - To develop associations of parents of hearing impaired children.
    - Encourage the creation of audio phonology and audiometry centers.
    - Opening of kindergartens.
    - Encouraging the teaching of sign language.

- Promote national and international exchanges.
- Developing cultural and sports activities for the benefit of deaf children.
- Opening holiday and leisure centers.
- Reserve exhibition spaces for the works and artwork of deaf people.
- Helping to build up a documentary fund for deaf people.

Here some of the statistics from the ADAS Tlemcen about deaf people in the region of Tlemcen, details in the following table:

Table 3: Statistics of deaf people in Tlemcen region

Age	Percentage of deafness	Number		Total
		Male	Female	
Under 18 years	Less than 100%	145	123	2081
	100%	185	196	
Above 18 years	Less than 100%	423	336	
	100%	361	312	

11. Dilemma survey

This survey is lunched on 22 January 2020 via Google forum asking some questions to know the severity of the problem and having some feedback (29 answers) the following (figures15)show some statistics.

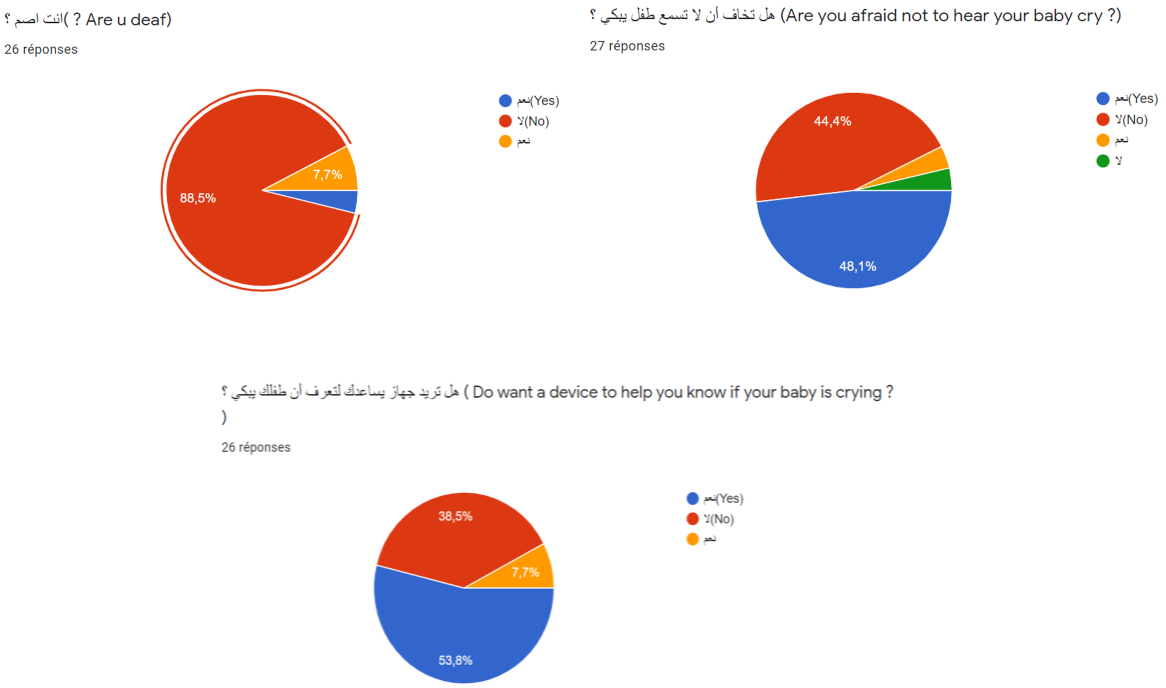


Figure 15: Survey results

Based on the data showed, we can assume that our solution is needed not just for the deaf community but also from the rest of communites.

12. Conclusion

In this chapter, we introduced some of the definition to the most important highlights that give importance to the sound and hearing issues especially for the people with hearing difficulties and what they face in the world with the most important actions taken in the organizations to reduce the outcomes for better conditions.

In the next chapter, we are presenting the steps for solving this social problem, proposing our technical solutions and methods to achieving the best answer to confront and tackle this problem.

Chapter III

Methodology

1. Introduction

Nowadays, baby cry detection is implemented on a smart home technology that makes it easy to monitor your baby's cries. There are many studies and researches on the detection and classification of infant cries for many purposes.

The right interpretation of the crying baby is notable for the medical objective so caregiver (parents, babysitters, etc.…) knows how to treat baby well.

In this chapter, we will focus on our methodology to realize this project, passing by the realization of the detection system (microphone, raspberry pi, etc.…), data collection, signal processing, classification methods and the implantation of the deep learning to make the right prediction of the cry, adding some definitions for the used technologies.

In this diagram, we present the work of the chapter: A [‘detection system’] B [‘signal processing’] C [‘classification phase’].

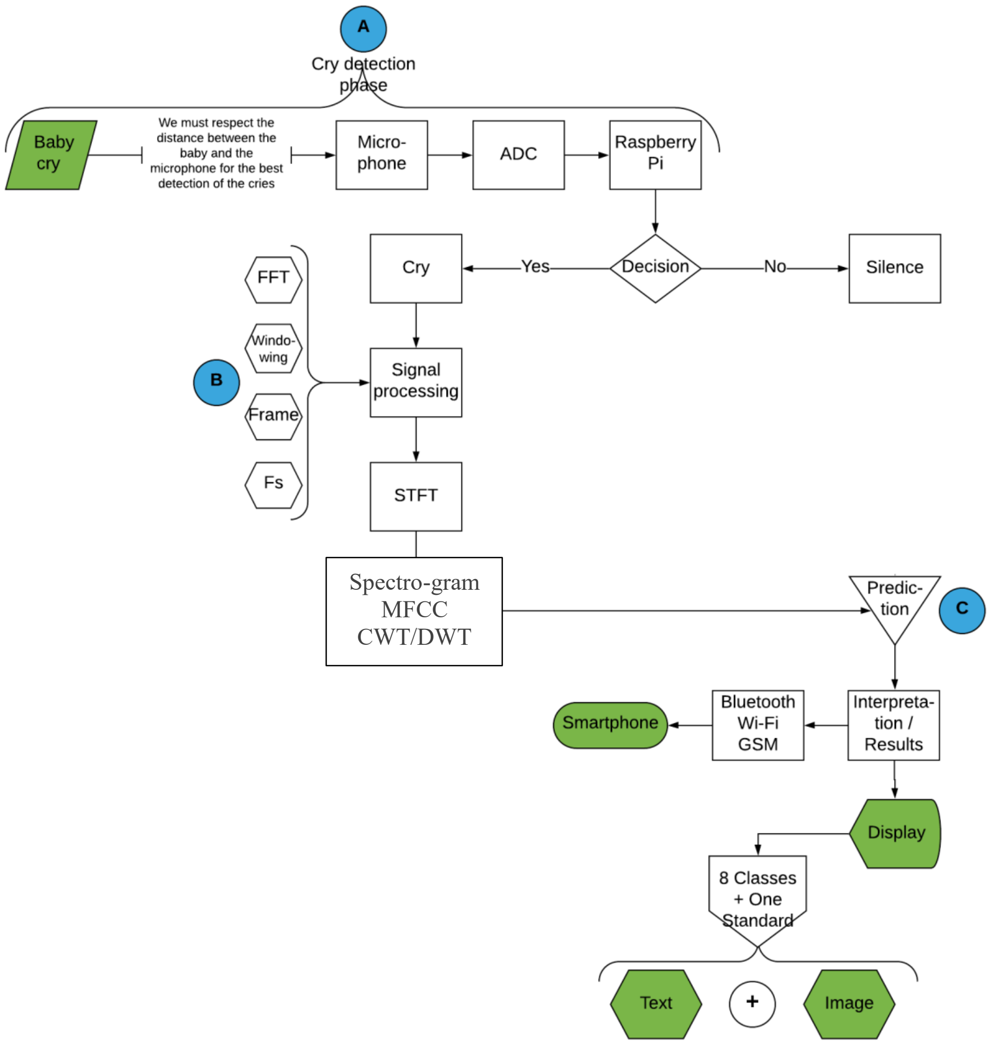


Figure 16: Diagram represents the realization of the project

2. Work dependencies

2.1. Framework dependencies

Table 4: Framework used

Specifics	HP Elite Book	Lenovo thinkpad L460
Systèmes d'exploitation	Windows 10 Professionnel 64 bits.	Windows 10 professionnel 64 bits
Processeurs	APU AMD PRO A10-8700B with graphic card Radeon™ R6 (1,8 GHz, till 3,2 GHz, 2 Mo of cached memory, 4 hearts).	Intel core i5-6800U Vpro with CPU:2,40 GHz; Intel graphic card 520;
Stockage	256 Go SSD M.2 SATA MLC SE + 500 Go HDD.	256 Go SSD
RAM	8 GB (7.42 usable).	8 GB (7.87 usable).

2.2. Hardware dependencies

Table 5: Hardware used

	Communication support	Equipment	Components
Used hardware	Raspberry pi 3 model B Arduino UNO Sound module KY-083 Compatible SD card reader for Arduino Audio jack module for Arduino ADS1115 (ADC module) MAX9814 (Microphone module) Microphone Sound card	Oscilloscope Base frequency generator BFG Multi-meter DC generator Soldering iron/Desoldering pump	Breadboard wire jumpers (M_M),(F_M) microphone electret Resistors / Capacitors TL081 LM741 LED (red, yellow, green)

2.3. Software dependencies

Table 6: Software used

Software	Definition
Matlab 2015	MATLAB is a multi-paradigm numerical computing environment and proprietary programming language developed by MathWorks.
Python 3.74	Python is an interpreted, high-level, general-purpose programming language.
Anaconda environment 3	Conda environments are language agnostic. That is, they support languages other than Python and serve to help manage dependencies and isolate projects.
Proteus 7.7	The Proteus Design Suite is a proprietary software tool suite used primarily for electronic design automation.
Multisim	NI Multisim is an electronic schematic capture and simulation program which is part of a suite of circuit design programs, along with NI Ultiboard.
Arduino IDLE 1.8.10 / 1.8.11.0	The Arduino Integrated Development Environment is a cross-platform application that is written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards.
Fritzing.0.9.3b.64.pc	Fritzing is an open-source initiative to develop amateur or hobby CAD software for the design of electronics hardware, to support designers and artists ready to move from experimenting with a prototype.
Spyder 3.3.6	Spyder is an open source cross-platform integrated development environment for scientific programming in the Python language.
Shell	Shell is piloting a new cloud-based, deep learning solution built on Microsoft Azure. The solution uses closed-circuit camera footage and Internet ...
Jupyter notebook	Project Jupyter is a nonprofit organization created to "develop open-source software, open-standards, and services for interactive computing across dozens of programming languages.
Visual code VS	Visual Studio Code is a free source-code editor made by Microsoft for Windows, Linux and macOS.
Node-red	programming tool for wiring together hardware devices, APIs and online services in new and interesting ways

3. Testing phase

In this phase, we decided to test the microphone (sensibility, reliability, etc.) using Arduino Uno and the purpose of using Arduino is to visualize the sound signal. Its facility and the availability of the open source help us achieve our goal (visualization of the sound signal).

First we passed by simulation phase to test if we have results using proteus.as shown in the Figure (17) the output gave very positive feedback to let us forward to the next step.

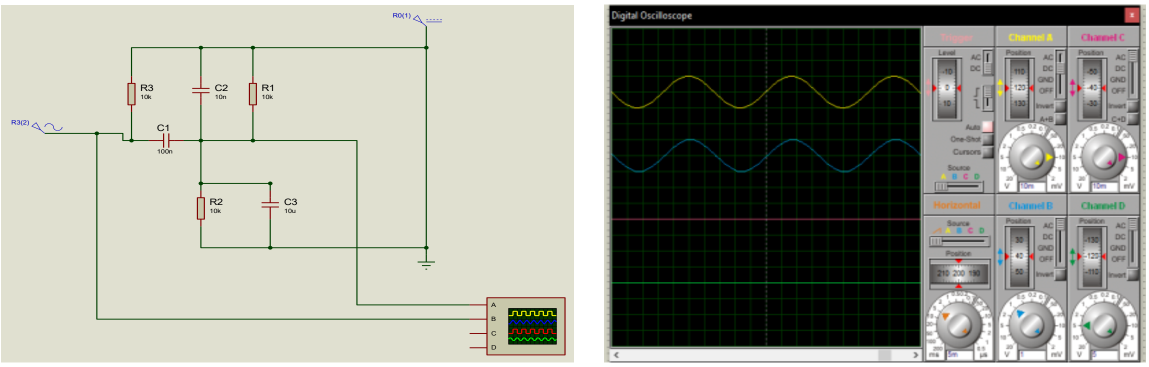


Figure 17: microphone-testing circuit, plot of results (yellow=input) (bleu=output)

This next step started by gathering the components for designing ”sound detector”, based on previous works and based on the electret microphone. After checking it data sheet (page ref data sheet), we used the sound sensor available in our laboratory which is microphone electret capacitor standard that have a medium sensitivity to detect the sound only from short distance and lot of noise detection, but it have a good frequency range in the audible phase that interested us.

Our realization of the circuit requires:

- Electret microphone.
- Polarization stage.
- Filtering stage.
- Arduino for the transmission of the sound.
- Screen computer used for visualization.

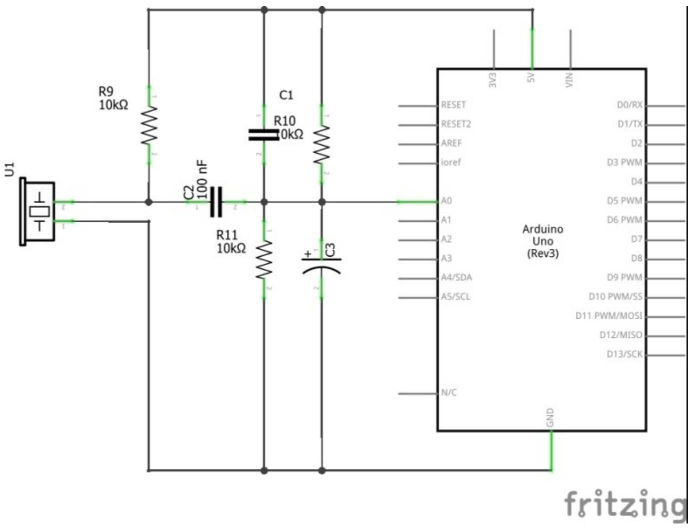


Figure 18: Scheme electric for the circuit

After gathering all the components, we started the realization of the circuit (Figure 19) based on the satisfied results of the simulation.

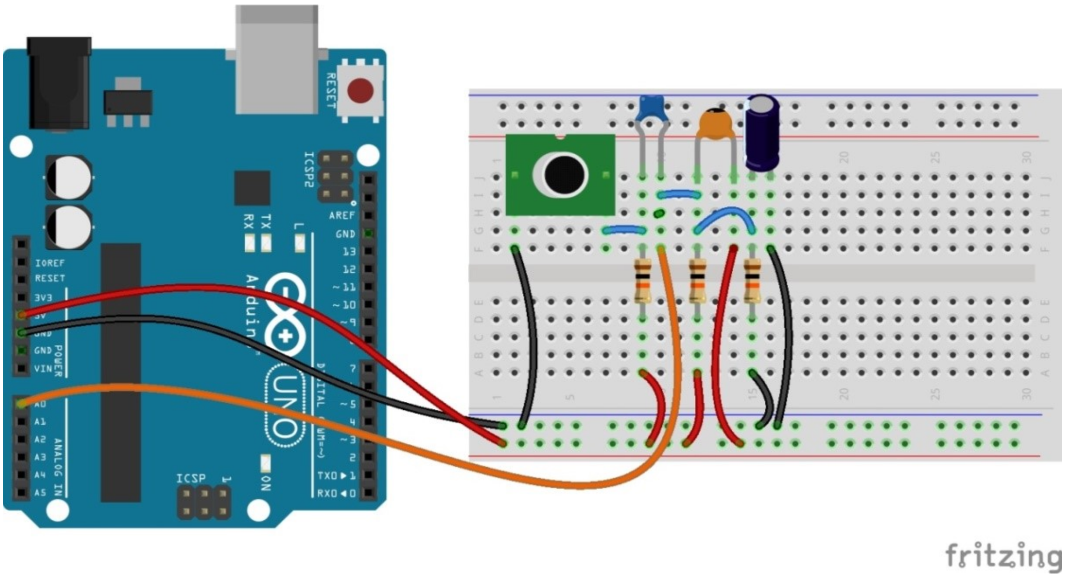


Figure 19: circuit realization with microphone and Arduino

The Arduino was the source of the voltage, the current for the alimentation of the circuit and for the visualization of the output signal via the computer, (the Arduino give us the possibility of working and testing the circuit outside the hours of studying and working in the laboratory).

By using the script below (Figure 20), we visualize the output signal through the computer. In addition, here some explications of the script.

- The `setup ()` function is called when a sketch starts. Use it to initialize variables, pin modes, start using libraries, etc…. The `setup ()` function will only run once, after each power up or reset of the Arduino board.
- The `loop ()` function does precisely what its name suggests, and loops consecutively, allowing the program to change and respond. Use it to actively control the Arduino board.
- The delay used is based on the frequency of the signal detected by the microphone (around 15000 Hz).
- The last line in the script is for the scale of the graph of the signal detected.

```
void setup() {
Serial.begin(9600);
pinMode(A0,INPUT);
}

void loop() {
  int val;
  val=analogRead(A0);
 // Serial.print("sound=");
  Serial.println(val);
 delay(0.000066);

 float voltage = val * (5.0 / 1024.0);
 //Serial.println(voltage);
}
```

Figure 20: Arduino script used

Note: The Arduino that we used have the characteristics that are not quite compatible with our needs but we tried at least to have some results by testing different circuits even the plotting of signal by using Arduino.

3.1. Microphone

We used the two condenser microphones with two different characteristics and proprieties for sound detection. It is first necessary to cover the basic definitions and the types of condenser microphone.

The most basic definition of a condenser microphone is an active microphone transducer (it requires power to function) with a capacitor-based capsule that employs electrostatic principles to convert sound into audio.

Even though there are countless examples of condenser microphones, they do share one core working principle[30]. With this principle come a few key components that condenser microphones share:

- Parallel-plate capacitor-based capsule
- A diaphragm (or more) that acts as one plate of the capacitor
- A back plate (or more) that acts as the other plate of the capacitor
- An impedance converter
- Circuitry to allow electrical power to properly charge and/or power the active components.

Condenser microphones are often chosen for their wide frequency responses; high-sensitivities; accurate transient responses; and overall sound quality. Of course, some condensers outperform others and with the wide variety of condenser microphones on the market. [30]

As a major differentiator between condenser microphones is the size of their diaphragms. [30]

Therefore, we look at the differences between SDCs and LDCs in the following table:

Table 7: The differences between SDCs and LDCs microphones

Type of microphone	Small-Diaphragm Condenser Microphones	Large-Diaphragm Condenser Microphones
Diaphragm Size	1/2" (12.7 mm) or less	1" (25.4 mm) or more
Transient Response	More accurate	Less accurate
Frequency Response	Flatter and more extended	More colored especially in the high-end
Address Type	Top or side	Typically side
Polar Patterns	Very consistent	Less consistent
Sensitivity	High	High
Self-Noise	More	Less
Price	Cheap to very expensive	Inexpensive to very expensive

To know more about microphones types, we cite the following:

3.1.1. Miniature-diaphragm condensers

Miniature-diaphragm condenser is separately from SDCs and LDCs. These microphones are often used in conjunction with wireless systems. They connect to wireless transmitters, which are generally used not only to send the signal wirelessly but also to provide the JFET impedance converter with proper DC biasing voltage in order for the microphone to work. [30]

3.1.2. Electret condensers

Electret condenser microphones have electret material built into their capsules, which maintains a quasi-permanent electric charge across the plate. These microphones are considered pre-polarized and so not require an external power source to provide a polarization voltage for the capsule. [30]

The electret material is typically Polytetrafluoroethylene (PTFE) plastic in film form or in solute form. [30]

3.1.3. Externally-polarized “true” condensers

Externally polarized condensers, as the name suggests, require an external voltage to properly polarize their capsules. Manufacturers used the term “true” to differentiate their externally polarized condensers from the lesser electret condenser microphones. [30]

3.1.4. Tube condensers

Tube condensers utilize vacuum tube electronics as their impedance converters; they are often preferred for their character. Therefore, although tube electronics are not as precise as transistor-based electronics, tube microphones are still sought after because they sound magnificent. [30]

Note: Tube condensers all have externally polarized capsules.

3.1.5. FET condensers

FET condenser microphones (otherwise known as solid-state condensers) have transistor-based impedance converters. Because transistor technology is so popular in these microphones, we have a wide range of condenser microphones that utilize FET ICs. [30]

Note: FET condensers can be pre-polarized or externally polarized and can have small or large diaphragms.

3.1.6. RF condensers

Thus far, in the article, we have been discussing AF (audio frequency) condenser microphones. These microphones use a high-impedance capacitor-based capsule to store a fixed charge and vary the capacitance of the capsule to produce a voltage. These microphones require an impedance converter if the capsule signal is to be used at all. [30]

However, there is no winning against high humidity with an AF condenser. This causes noisy and reduced output. The high biasing voltage also attracts dust particles to the diaphragm, reducing its efficiency and linearity. [30]

There is another type of condenser capsule; they are practically immune to humidity due to the low-impedance of the circuit (RF condensers).

This system was developed by Sennheiser for use in their MKH shotgun microphones and is known as the RF (radio frequency) condenser microphone. [30]

RF condensers utilize a low-impedance capsule as a tuning capacitor for an RF oscillator. This oscillator employs the capacitor/capsule in a low-impedance circuit where a high-frequency signal is passed through the capacitor at all times. [30]

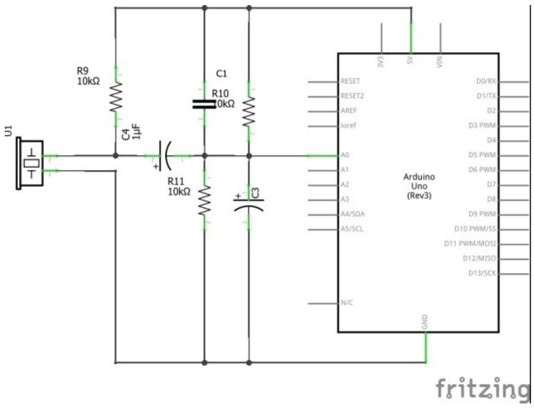
An RF demodulator (rather than an impedance converter) is then put in-line to restore the output to an audio signal. [30]

3.2. Polarization

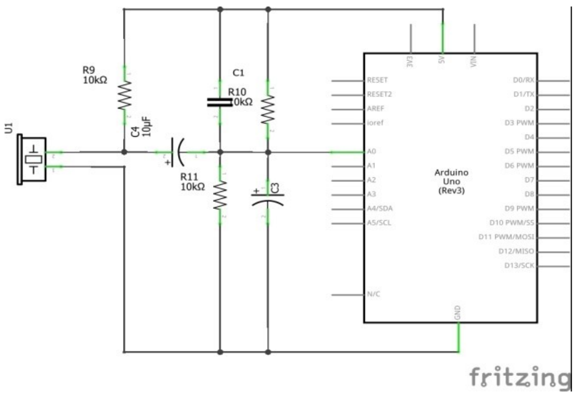
Polarization is the method in which the condenser capsule of the microphone is polarized (charged). The polarizing voltage that causes a fixed charge across the plates is generally supplied by external means (typically via phantom power or an external power supply with FET and tube condensers, respectively)[30].

For the polarization stage, we tested about five different circuits to find the most compatible with our application and the one that can gives us the best results for the microphone polarization, we mention them by order.

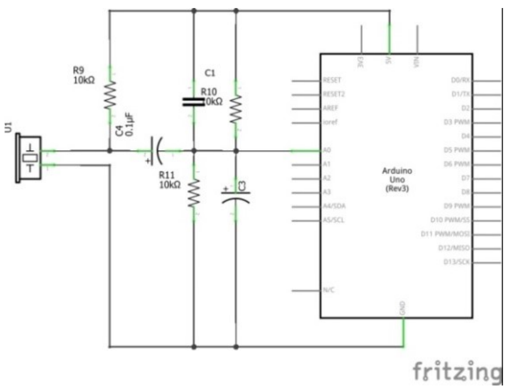
- First polarization circuit:
  - R= 10 kΩ; C=1 µF



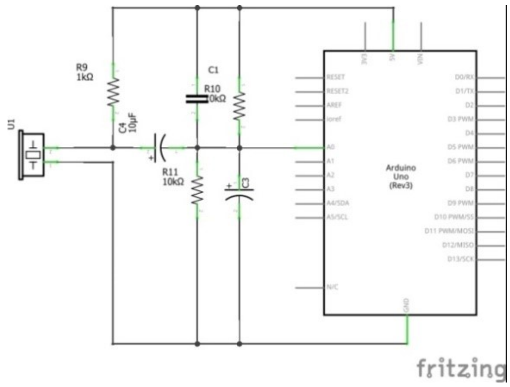
- Second polarization circuit:
  - R= 10 kΩ; C=10 µF



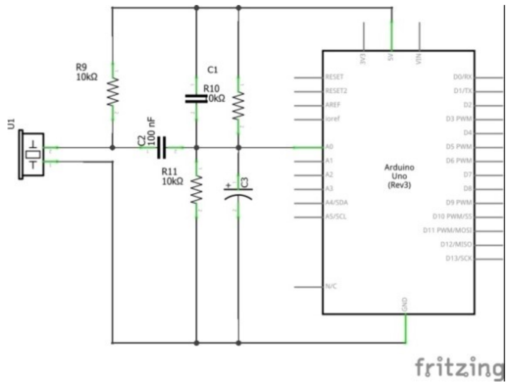
- Third polarization circuit:
  - R= 10 kΩ; C=0.1 µF



- Fourth polarization circuit:
  - R= 10 kΩ; C=1 µF



- Fifth polarization circuit:
  - R= 10 kΩ; C=100 nF



By the end of this process, we adopted the fifth solution because of the optimal wave signal found with it (see figure 21).

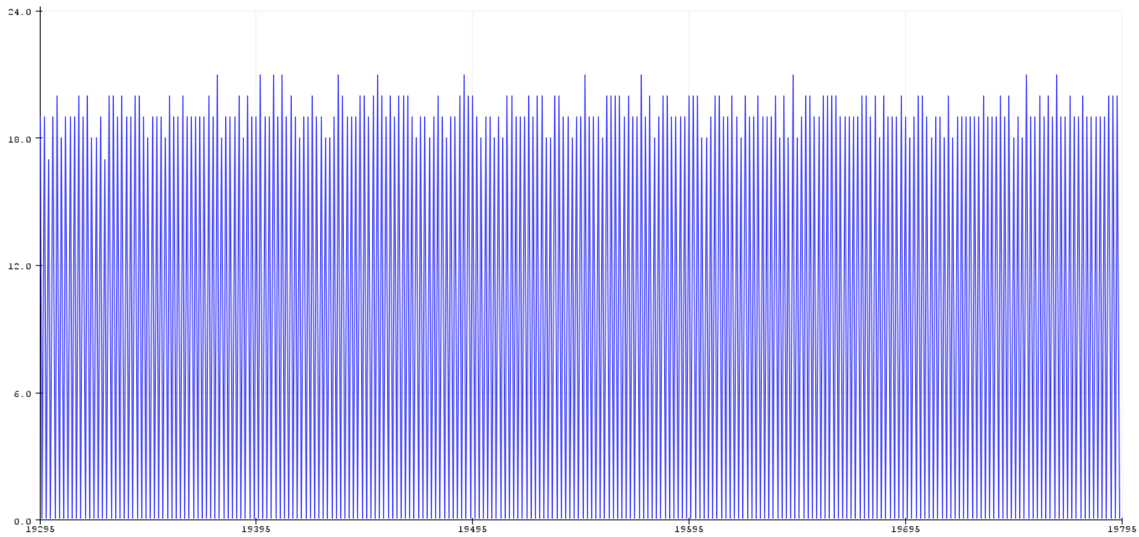


Figure 21: The testing plot using the mounted circuit

3.3. Filtration

Electronic filters are circuits that perform signal processing functions; furthermore, an electronic filter is device acting on a circuit to limit unwanted frequency, electronic noise, common mode voltages, peak voltage, current magnitude or other undesirable current characteristics. They can be passive or active, analog or digital[31].

As shown in the (figure17) of the simulation we used two filter ,HPF(high pass filter) ,LPF (low pass filter) , and in another test we used BPF(band pass filter) all in the passive format the results were negative, then we used the active filters using TL081 amplifier and LM741 amplifier searching for the best wave sound signal visualization.

3.3.1. High Pass Filter

HPF is a filter, which passes only those signals whose frequencies are higher than cutoff frequencies thereby attenuating signals of lower frequencies. The value of cutoff frequency depends on the design of the filter. The basic High Pass Filter is built by a series connection of capacitor and resistor. While the input signal is applied to the capacitor, the output is drawn across the resistor.

$$\boldsymbol{F(\omega) = \frac{1}{1 + jRC\omega}}$$

Equation 1: Transfer function for HPF

a. HPF passive

For all Passive RC High Pass filter as the circuit is built using only passive elements. There is no need of applying external power for working of the filter. Here capacitor is the reactive element and output is drawn across the resistor.

- The used HPF characteristics :

The cutoff frequency Fc depends on the values of “R” and “C” in the circuit used, here we have τ=RC, the Fc is the proportional invers of the time; Fc=1/RC

To determine the value capacitors we fixed the resistor value (R=10kΩ) and we know the frequency cutoff (Fc=30 Hz):

$$\mathbf{Fc=1 / 2\pi{*}R{*}C \Rightarrow C=1 / 2\pi{*}R{*}Fc}$$

$$\mathbf{C=1uF}$$

So the HPF cutoff the frequencies below then 30 Hz using R=10kΩ and C=1uF.

- High pass filter Bod plot

In high pass filter, all frequencies lying below the cutoff frequency ‘fc’ are attenuated. At this cut off frequency point we get -3dB gain and at this point reactance of the capacitor and resistor values will be same .i.e. R = Xc. Gain is calculated as

Gain (dB) = 20 log (Vout/Vin)

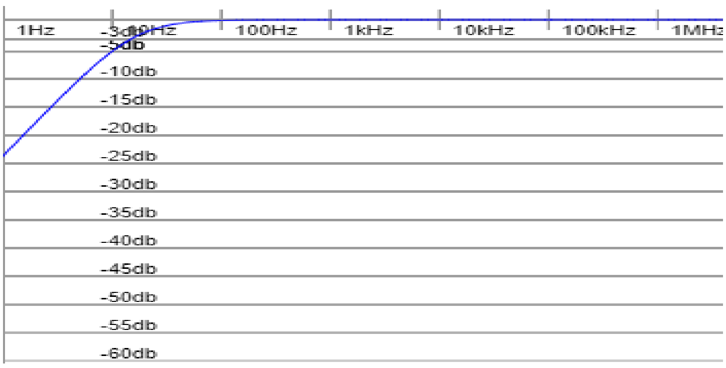


Figure 22: 1st order high pass filter Bode plot

b. HPF active

In this high pass filter along with passive filter elements, we add **Op-amp** to the circuit. Instead of getting an infinite output response, here the output response is limited by open loop **characteristics of the Op-amp**.

Using this active element, we can control the cutoff frequency and output response range of the filter. Hence, this filter acts as a **band-pass filter** with a cut off frequency, which is defined by the bandwidth and gain characteristics of Op-amp.

The **gain of the filter** using non-inverting Op-amp is given by:

AV = Vout/Vin = (Af (f/Fc)) / √(1+ (f/Fc)^2)

Where Af is the gain of the filter = **1+ (R2/R1)**

3.3.2. **Low Pass Filter:**

A Low Pass Filter is a circuit that can be designed to modify, reshape or reject all unwanted high frequencies of an electrical signal and accept or pass only those signals wanted by the circuit designer[32].

$$\boldsymbol{F}(\boldsymbol{\omega}) = \frac{\mathbf{jC\omega}}{\mathbf{R} + \boldsymbol{jC\omega}}$$

Equation 2: Transfer function for LPF

a. **Passive LPF**

Passive filters are generally constructed using simple RC (Resistor-Capacitor) networks, while higher frequency filters (above 100 kHz) are usually made from RLC (Resistor-Inductor-Capacitor) components.

The low pass filter only allows low frequency signals from 0Hz to its cut-off frequency, fc point to pass while blocking those any higher.

- The used LPF characteristics

To determine the value capacitors we fixed the resistor value (R=10kΩ) and we know the frequency cutoff (Fc=15 kHz):

Fc=1/2πRC ➔ C=1/2πRFc

C=1nF

So the LPF cutoff the frequencies below then 15 kHz using R=10kΩ and C=1nF.

- Low pass filter Bod plot

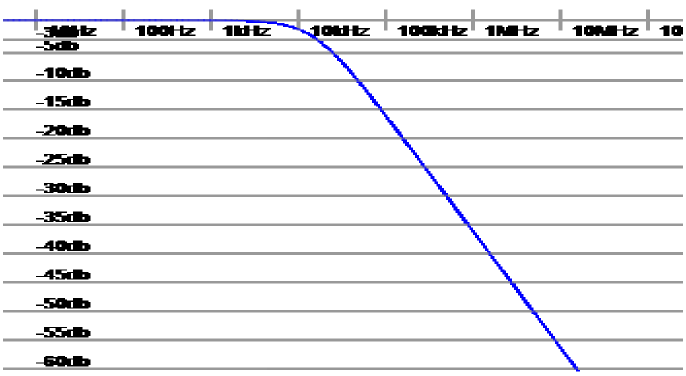


Figure 23: 1st order low pass filter

b. **Active LP**

By combining a basic RC Low Pass Filter circuit with an operational amplifier, we can create an Active Low Pass Filter circuit complete with amplification.

Contain active components such as operational amplifiers, transistors or FET's within their circuit design. They draw their power from an external power source and use it to boost or amplify the output signal.

3.4. Circuit realization using the module KY-083:

The circuit realization in this part using sound sensor module KY-083 (See figure 24); we had more optimal results for the signal wave (figure 25).

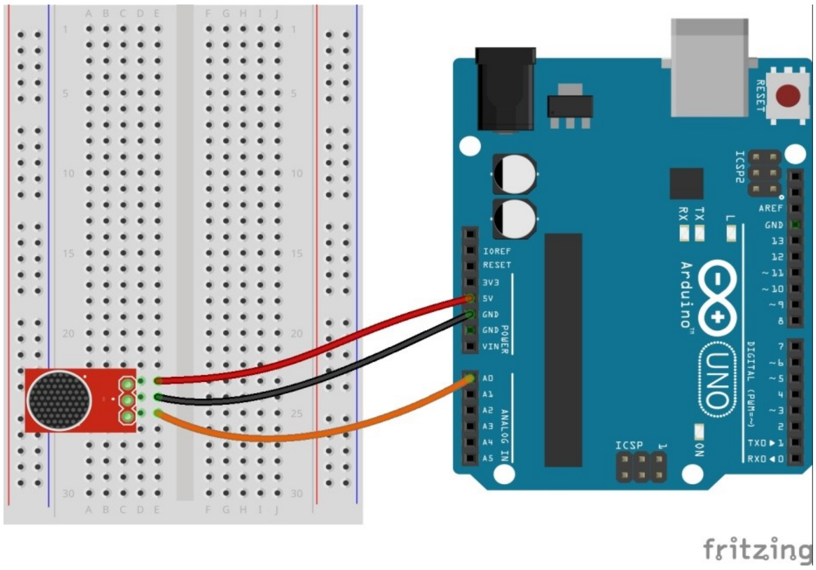


Figure 24: Wiring for KY-083 module with Arduino

Note: the Arduino and computer are the equipment used for the signal visualization. And we use the same script code used before.

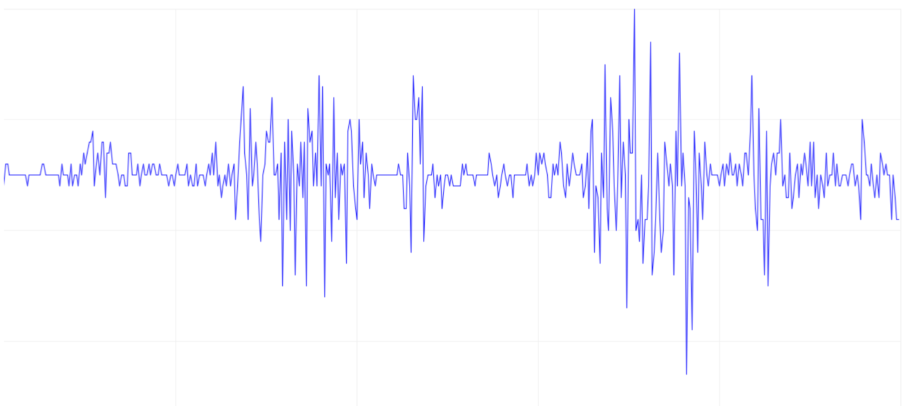


Figure 25: Finale plot results using microphone module KY-083

In conclusion, we found that the Arduino Uno is not going to be good for our application so we upgraded to the Raspberry Pi especially for the sampling frequency, bits number and for the speed of the calculation.

4. The ICSD data used

4.1. Data collection

The data was collected from Pranaam hospital, Madinaguda, Hyderabad, under the supervision of Dr. Manish Gour (MBBS, DCH) and Dr Nizam (MBBS). The age group of infants was restricted between 3 months and 2 years. [2]

The cry signals of infants we recollected during their gular checkup visits, the vaccination trips or any emotional need of attention. People present in the room were requested to maintain silence, to record the cry sample. [2]

In addition, the parents advised not to comfort the baby for brief duration, to ensure the uninterrupted data collection. Along with data, the personal details noted include infant name, parent name, parent profession, sex and age of the infant and predictive causes of the cry. For the recording purpose, they used Roland R-09 Wave/MP3 recorder and it placed at 10-20 cm from the infant's mouth. [2]

To avoid unwanted noise or interference (Crosstalk), they take some precautions. Sampling rate of 48 KHz, with 24 bit coding rate, was used for recording in stereo mode. There were no interruptions from the social environment during the data recording. The only unwanted noise that could overlap the cry sound maybe from fan and air-conditioner. The ambient temperature during the recordings was 38◦C, which was regulated by the air conditioner at 25◦C. [2]

4.2. Organization of the IIIT-SICSD

The terminologies used in naming of the database files, given in Table 8, are described below

Table 8: Template for naming files in IIIT-S ICSD

SPKR01 M S1a CRY07	
(a) Symbols	(b) Interpretation
SPKR#	The infant number (Ex: 01)
M/F	Sex of the infant (Ex: Male)
S#a	Session number and session subpart (Ex: 1a)
CRY#	Number of cries in the session being considered (Ex: 07)

- Session: The acoustic signal, right from the time an infant starts crying (including all inhalations and exhalations), until the infant becomes quiet, is a session. [2]
- Session subpart: Each session consists of subparts, characterized by the contiguous set of signals, separated by some noise. [2]
- Cry: Each session subpart comprises of a number of cries, separated by some noise. [2]

A two-stage process was followed for data collection in the study. The first stage involved raw data collection at the hospital, and the second stage included pre-processing. The unwanted noise, in the raw data was removed using 'WaveSurfer' tool, to render it cross-talk free. The cries we categorized as per the ground reality, i.e., the actual cause as per the doctor or parent. The main causes of cry that we came across are described in Table 9, in columns (a) and (b).[2]

Table 9: Causes of Infant cry in IIIT-S ICSD

(a) Cry Causes	(b) Description
1. Pain	Cry due to pain (caused by vaccination, physical hurt or internal pain)
2.discomfort	Cry due to irritation caused by the external environment(e.g., the doctor opening baby's mouth to pour-in drops, or the vaccination)
3.emotional need	Cry when the baby wants to go back to parents arms
4.aliment	Cries due to any ailments like cold, cough, fever
5.enviromental factors	Cry due to fear of the surroundings or change in environmental conditions
6.hunger/thirst	Cry when the baby is hungry or thirsty

The cry as combination of two or more causes is retained in a separate category. There was also a special case, when an infant cried by listening to another infant cry.

Not many samples could be obtained for this cry due to Domino effect, but this is retained as another special category, for future study. The database consists of total 76 cry sound files, which can be categorized in six classes. The database summary is given in Table 10.

Table 10: Summary of contents in IIIT-S ICSD

Attributes	Values
Total number of files	**76**
Total number of speakers	**33**
Total number of session	**76**
Total number of cries in one sessions	**693**
Average number of sessions per speaker	**2.3**
Average number of cries per speaker	**9.1**
Total duration of all sessions	**670.1 s**
Average duration of each session	**8.817 s**

4.3. Reorganization of the data base

For the purpose of using this data with deep learning we had to reorder it according to our need, by splitting it into 8 sub files with categories as shown in the figure 26.

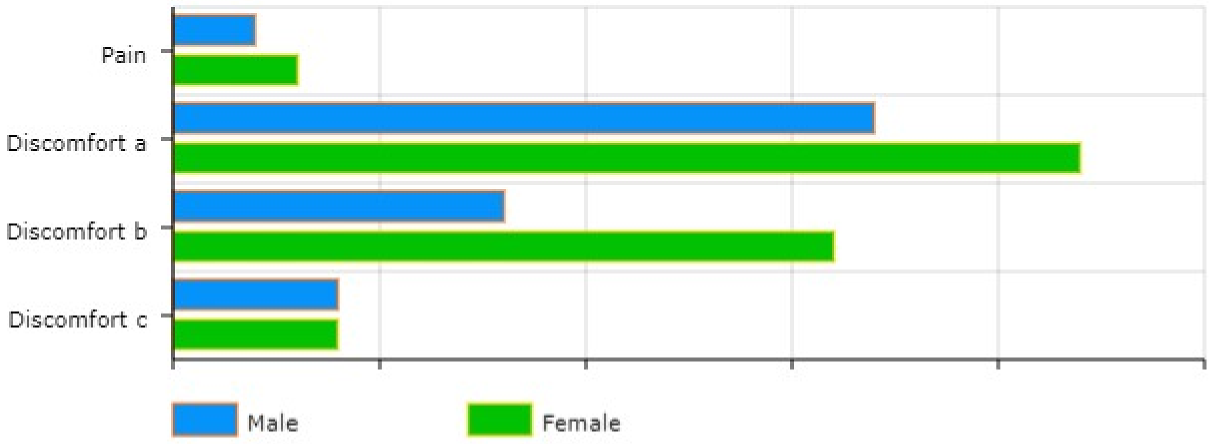


Figure 26: Data reorganization

5. Signal processing phase

Signal processing is a technic that manipulate the signals from real-world events, in our case sound converted into digital data and analyzed using algorithms, to decorticate the data for better understanding.[33]

5.1. Signal processing Libraries in python

During the whole process using python, the Libraries used in this first part:

5.1.1. Numpy:

NumPy is the fundamental package for scientific computing with Python[33]. It contains among other things:

a. **A powerful N-dimensional array object**
b. **Sophisticated (broadcasting) functions**
c. **Tools for integrating C/C++ and Fortran code**
d. **Useful linear algebra, Fourier transform, and random number capabilities**

5.1.2. Matplotlib:

Matplotlib is a Python 2D plotting library as well as rudimentary 3-D plotting. Which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms[33].

5.1.3. **Spicy**

Collection of numerical algorithms and domain-specific toolboxes including signal processing, optimization, statistics, and much more. Tools for data management and computation[33].

5.1.4. **Pwt**

Wavelet transforms software for Python. It combines a simple high level interface with low level C and Python performance. To analysis de-noising and compression of signals and images. This section describes functions used to perform single- and multilevel Discrete Wavelet Transforms.

5.1.5. **Librosa**

LIBROSA is a python package for music and audio analysis. It provides the building blocks necessary to create music information retrieval systems[34]. It’s the most important tool used for the whole work, All required libraries must be installed and updated to insure proper results.

5.2. Initial steps for SP:

The process of signal processing is resumed in the following diagram (figure 27) followed with further explications:

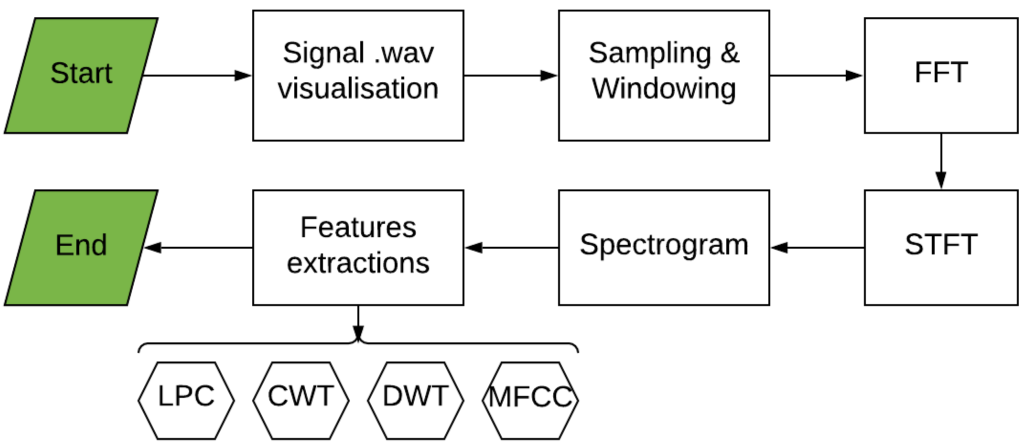


Figure 27: Signal-processing diagrams

The Detaille of the processing using one sample of the data (spkr15_F_S1a_cry06.wav):

Obtaining the plot of the waveform sound for the cry in the following figure (figure 28):

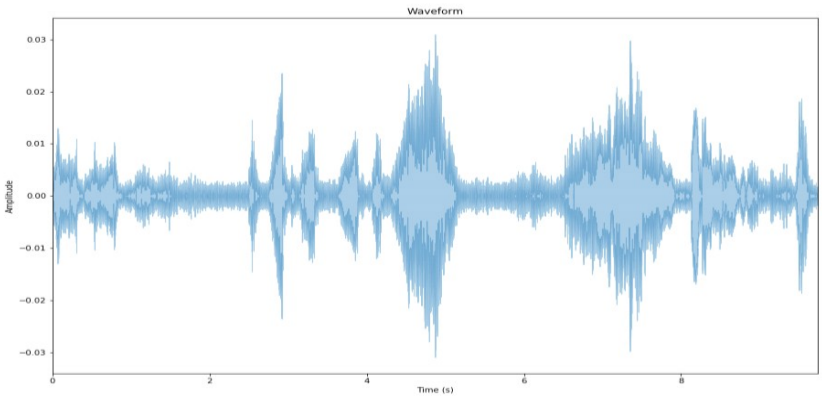


Figure 28 Sound Cry waveform of spkr15_F_S1a_cry06

5.2.1. Applying the fast Fourier transform:

The Fourier Transform (Equation 3) decomposes a function of time (signal) into constituent frequencies. In the same way a musical chord can be expressed by the volumes and frequencies of its constituent notes, a Fourier Transform of a function displays the amplitude (amount) of each frequency present in the underlying function (signal).

$$\boldsymbol{f_j = \sum_{k=0}^{n-1} x_k e^{-\frac{2\pi i}{n} jk} \qquad j = 0, \ldots\ldots, n-1}$$

Equation 3: The Fourier Transform

The fft results showed in the (Figure 29).

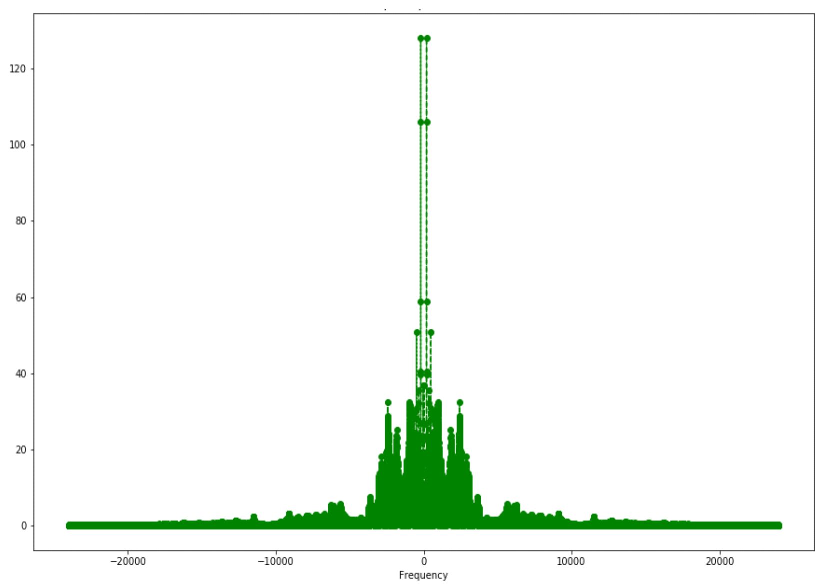


Figure 29: fast Fourier transform for spkr15_F_S1a_cry06

5.2.2. Power spectrum

Calculate the abs values on complex numbers to get magnitude using the next function, the **power spectrum** of a time series is a way to describe the distribution of power into discrete frequency components composing that signal. The statistical average of a signal measured by its frequency content.

The Figure 30 shows the plot of the sample:

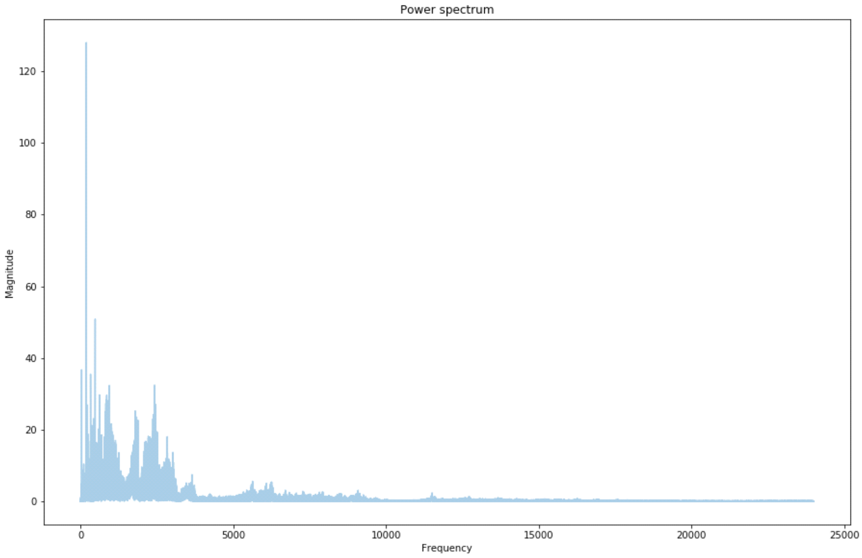


Figure 30: power spectrum of spkr15_F_S1a_cry06

5.2.3. Short-time Fourier transform

The Short-time Fourier transforms (Equation 4), which is implemented in the Librosa library and involves splitting an audio signal into frames and then taking the Fourier Transform of each frame. In audio processing generally, the Fourier is an elegant and useful way to decompose an audio signal into its constituent frequencies.(figure 31)

$$X(\tau,\omega) = \sum_{n=-\infty}^{n=\infty} x[n]w[n-m]e^{-i\omega n} \qquad 0 \le n \le N-1$$

Equation 4: Short-time Fourier transforms

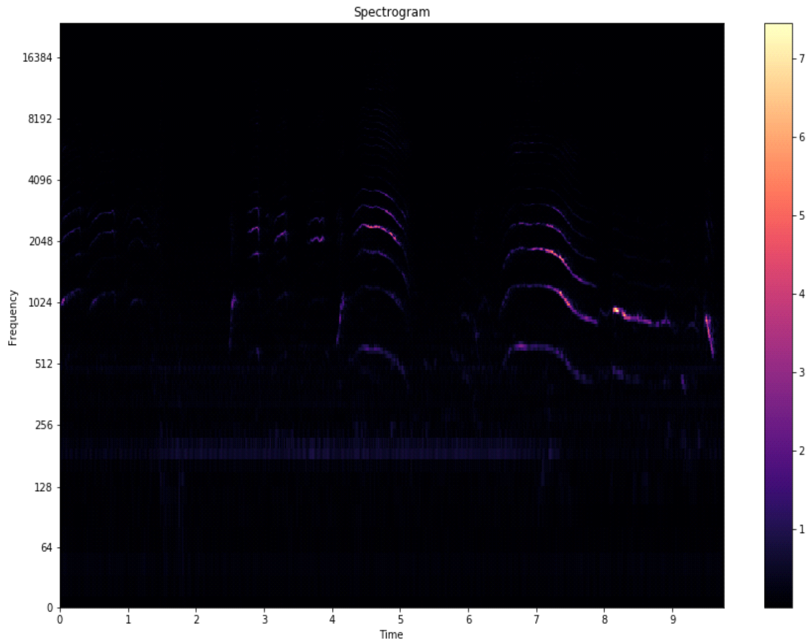


Figure 31: STFT for spkr15_F_S1a_cry06

5.2.4. Spectrogram

A spectrogram is a visual representation of the spectrum of frequencies of a signal as it varies with time. A nice way to think about spectrograms is as a stacked view of periodograms across some time-interval digital signal. Apply logarithm to cast amplitude to Decibels to visualize the information more clearly.

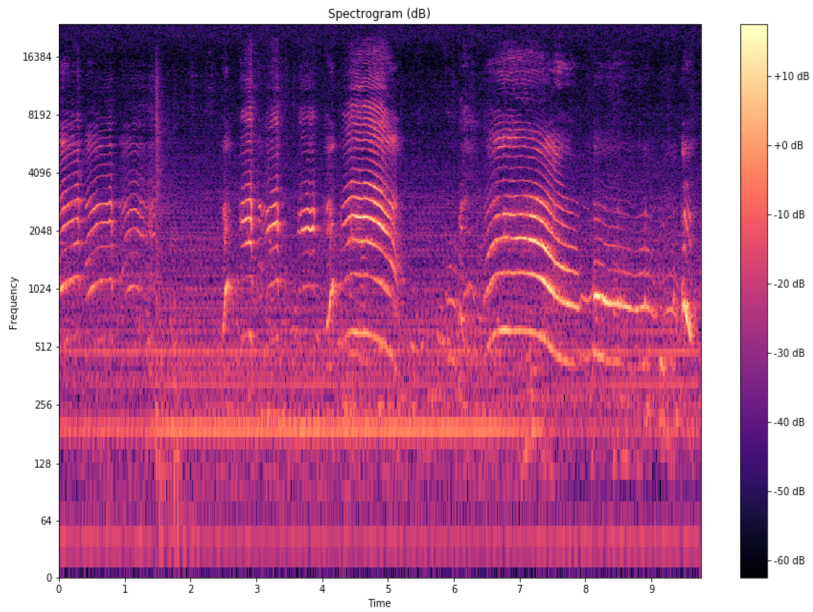


Figure 32: spectrogram for spkr15_F_S1a_cry06

Having the spectrograms (figure 32) means we finished the first part of signal processing that are explored in the next steps for features extractions used with the deep learning.

5.2.5. Feature Extraction Methods

Audio's features are extracted by dividing input signal from the frame with a length of hop_ length then each feature value is calculated; in this part, we give comparison using four method for feature extraction discrete wavelet transform (DWT) and continuous wavelet transform (CWT) and Mel Frequency Cepstral Coefficient (MFCC) and Linear Predictive (LP) to decide the taken methods.

a. Linear prediction (LP)

Linear Predictive Coding (LPC) of speech began. The linear-prediction voice model (figure 33) is classified as a parametric, spectral, source-filter model, in which the short-time spectrum is decomposed into a flat excitation spectrum multiplied by a smooth spectral envelope capturing primarily vocal formants (resonances).

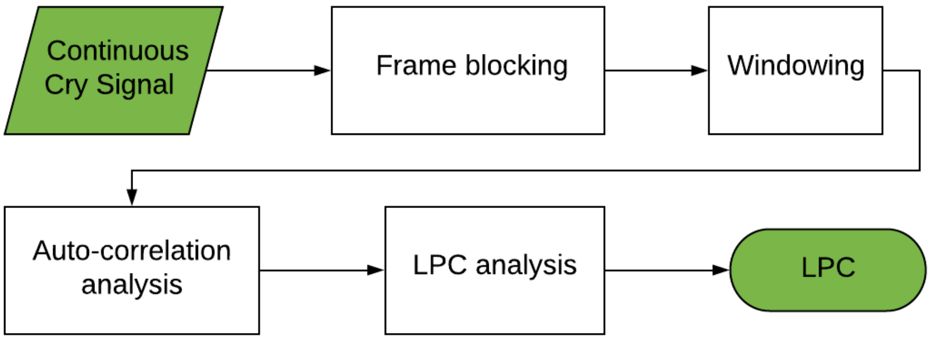


Figure 33: Block diagram of LPC processor.

Linear prediction method is applied to obtain the filter coefficients equivalent to the vocal tract by reducing the mean square error in between the input speech and estimated speech. Linear prediction analysis of speech signal forecasts any given speech sample at a specific period as a linear weighted aggregation of preceding samples. The linear predictive model of speech creation is given as[35]:

$$\hat{S}(n) = \sum_{k=1}^{p} a_k s(n-k)$$

Equation 5: Linear predictive model

Where $\widehat{S}$ is the predicted sample, s is the speech sample, p is the predictor coefficients. The prediction error is given as:

$$e(n) = s(n) - \hat{s}(n)$$

Equation 6: Prediction error

Subsequently, each frame of the windowed signal is auto correlated, while the highest autocorrelation value is the order of the linear prediction analysis. This is followed by the LPC analysis, where each frame of the autocorrelations is converted into LPC parameters set which consists of the LPC coefficients[35]. A summary of the procedure for obtaining the LPC is seen in (Figure 33). LPC can be derived by:

$$a_m = log\left[\frac{1-k_m}{1+k_m}\right]$$

Equation 7: LPC coefficients

Where a_m is the linear prediction coefficient, k_m is the reflection coefficient.

(The figure 34) shows the results and we can notice that we cannot extract lot of the information's we need.

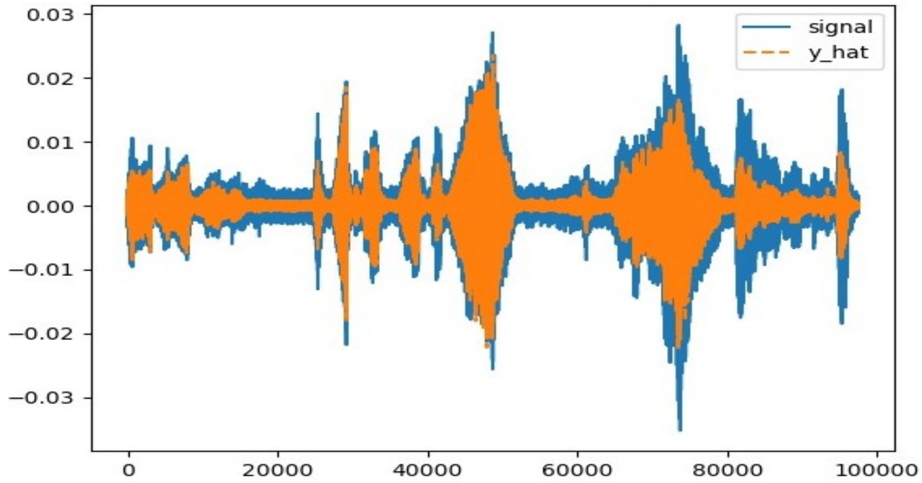


Figure 34: LP model forward prediction for" spkr15_F_S1a_cry06.wav"

b. Continuous Wavelet Transform (CWT)

In mathematics, the continuous wavelet transform (CWT) is a formal tool that provides an over complete representation of a signal by letting the translation and scale parameter of the wavelets vary continuously. The continuous wavelet transform of a function $x(t)$ at a scale (a>0) $a \in R^{+*}$ [36] ;and translational value $b \in R$ is expressed by the following integral:

$$x_\omega(a,b)=\frac{1}{|a|^{1/2}}\int_{-\infty}^{\infty} x(t)\overline{\varphi}\left(\frac{t-b}{a}\right)dt$$

Equation 8: The continuous wavelets transform (CWT)

The choice of the scale is based on previous work made[36] and the proper choice of scales depends on the chosen wavelet.

Small scale $a \Rightarrow$ Compressed wavelet $\Rightarrow$ rapidly changing details $\Rightarrow$ High frequency ω.

Long scale $a \Rightarrow$ Stretched wavelet $\Rightarrow$ slowly changing, coarse features $\Rightarrow$ Low frequency ω.

And the rest of parameter is in default mode “false” mean are not activated and not used in the python function yelled.

- Output [ ]:

Table 11: CWT parameters

	Family name	Short name	Symmetry	DWT	CWT	Complex CWT
Continuous Wavelet mexh	Mexican hat wavelet	Mexh	Symmetric	False	True	False
Continuous Wavelet morl	Morlet wavelet	morl	symmetric	False	True	False

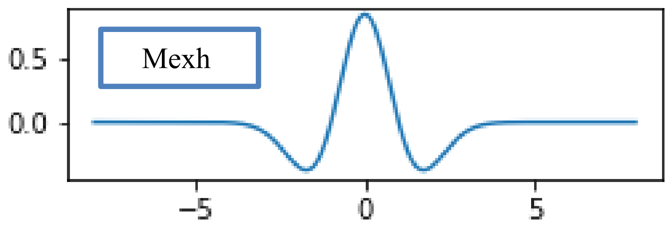


Figure 35: Mexican hat wavelet representation

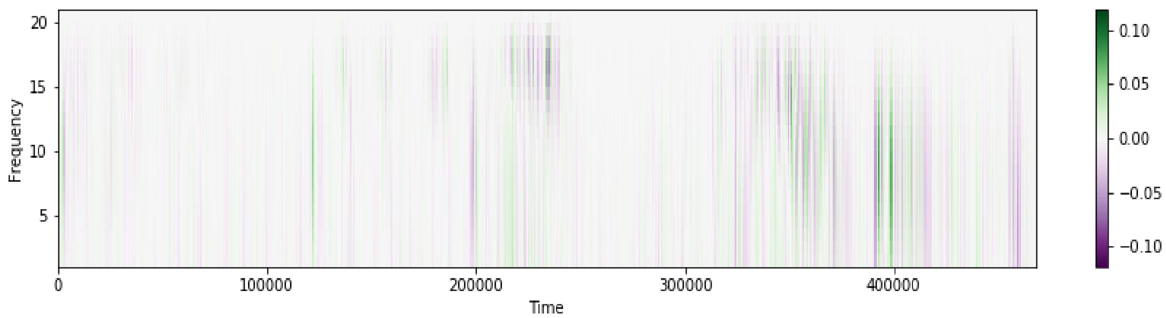


Figure 36: Continuous wavelet transforms spkr15_F_S1a_cry06 with mexh

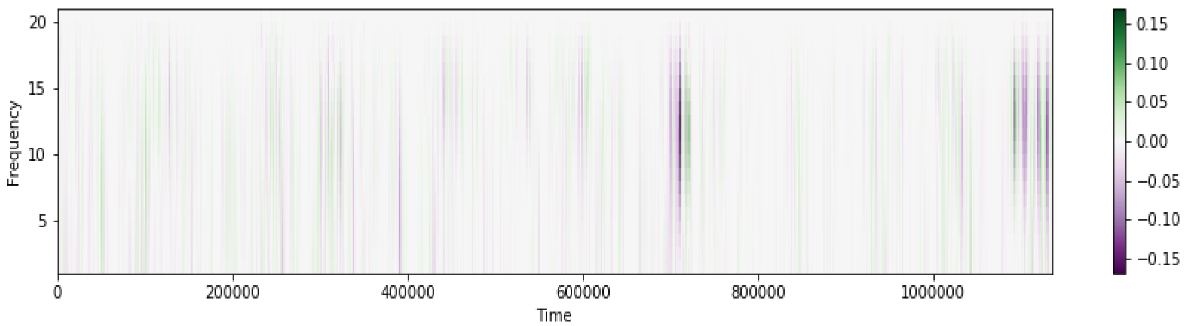


Figure 37: Continuous wavelet transform for "spkr09_M_S1a_cry14.wav"

We repeat same method using « morlet » wavelet to see the difference between it and the Mexican hat wavelet:

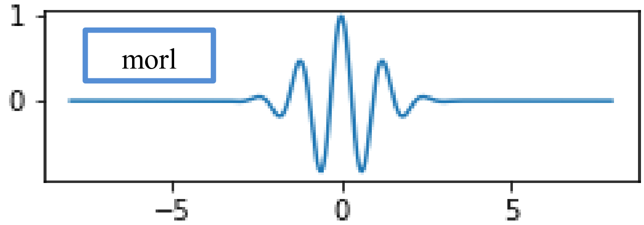


Figure 38: morl wavelet representation

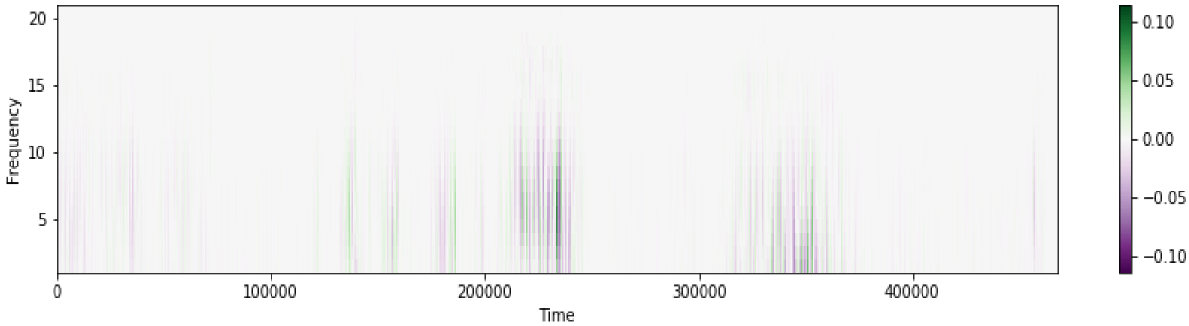


Figure 39: Continuous wavelets transform for" spkr15_F_S1a_cry06 " with morl

As the figures (35,36,37) above show, we are not having much information's to extract the parameters, which eliminate this method from the list even though we used the Mexican hat wavelet that gives the best results then the Morlet one (figure 38, 39).

c. Discrete Wavelet Transform (DWT)

To analysis, de-noising and compression of signals and images we are using Daubechies wavelet considered as the most compatible families with sound signals.

Daubechies wavelets are localized in the temporal domain, and approximately localized in the frequency domain [29].

$$\boldsymbol{y}[\boldsymbol{n}] = (\boldsymbol{x} * \boldsymbol{g})[\boldsymbol{n}] = \sum_{\boldsymbol{k}=-\infty}^{\infty} \boldsymbol{x}[\boldsymbol{k}]\boldsymbol{g}[\boldsymbol{n}-\boldsymbol{k}]$$

Equation 9: Discrete Wavelet Transform (DWT)

Table 12: DWT parameters

Wavelet db2	Family name	Short name	Filters length	Orthogonal	Biorthogonal	Symmetry	DWT	CWT	Maximum decompose level
	Daubechies	dB	4	True	True	asymmetric	True	False	17

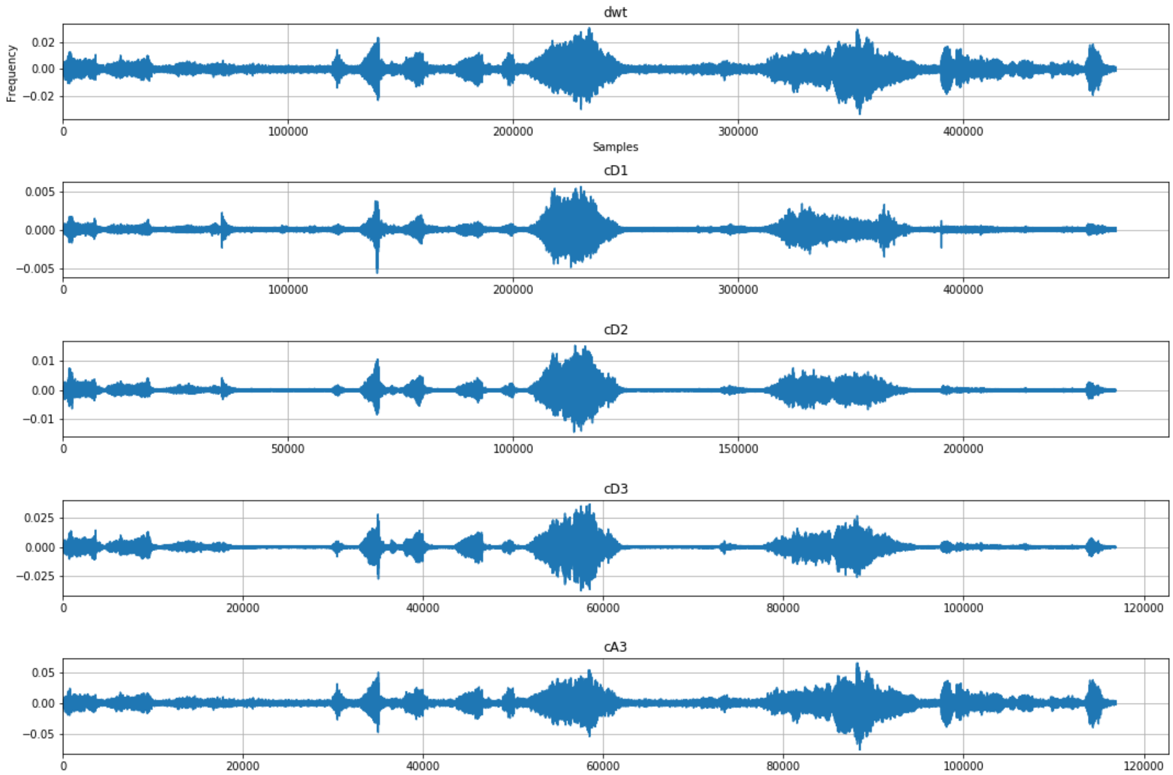


Figure 40: Discontinued wavelet transform for spkr15_F_S1a_cry06

Same results shown in (figure 40), we cannot extract much information using the DWT, even though we can notice differential between levels ,which means this method is not suitable for our application.

d. Mel Frequency Cepstral Coefficient (MFCC):

It a popular method used for feature extraction in audio it convert the sound into signal vector, because it is considered similar to the concept of human hearing. This method provides a representation of the short-term power spectrum of the signal. [37]

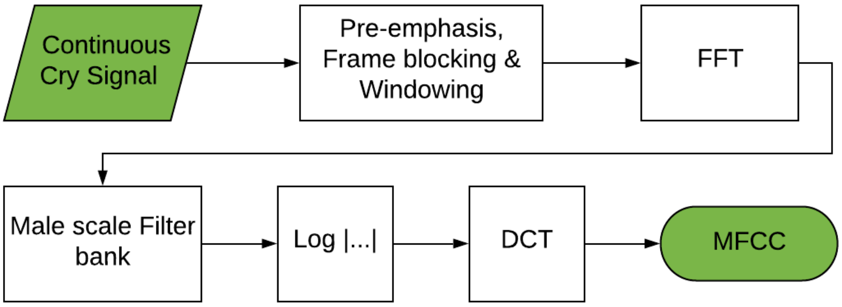


Figure 41: Block diagram of MFCC processor.

MFCC concept (figure 41) was similar to human hearing which has a critical bandwidth of the human ear at the frequency below 1000Hz. MFCC process start from dividing the sound signal into the form of a frame with a duration of 10-40 milliseconds time frame, this is the frame blocking. [37]

Then frame blocking is windowed by hamming to eliminate aliasing effects that occur due to the framing. The windowing process where w is a windowing function and N is a number of samples in one frame, the Equation is the formula for the windowing process:

$$W_n = 0.54 - 0.46 \cos \frac{2\pi n}{N-1'}$$

Equation 10: Hamming window

The results of this windowing process then followed by the calculation of the Fast Fourier Transformation (FFT), which converts the signal from the time domain into the frequency domain. Filter-bank applied to the signal with the frequency domain so that signal it turns into Mel frequency by the equation[37]:

$$mel(f) = 2595\ \log_{10}(1 + \frac{f}{700})$$

Equation 11: MFCC

MFCC using bank filter Mel-scale, which is a band-pass filter triangle logarithmic. It makes a higher frequency filter results the greater bandwidth. [37]

The final stage in the MFCC process is Discrete Cosine Transformation (DCT) after the results of the previous process is converted back into the time domain, so that the signal can be presented well. These results form a row of an acoustic vector, which named Mel Frequency Cepstral Coefficient.[37]

Based on previous literature, the human ear has non-linear characteristics in tone perception. The relationship between Frequency and Mel-scale described for a case about frequency above 500Hz where the increasing value of the interval is proportional to the increasing in the same pitch, for example, four octaves on a scale Hz over two octaves on the Mel scale value[38].

The relationship between the use of Mel scale and scale Hz in a nonlinear mapping function is useful for analyzing seismic signals where there are few differences between the speech signal and seismic signal. The filter is used for the frequency range 0-22050 Hz in speech recognition but in the case study took samples on the band seismic signal is below 500Hz. Mapping function which gets under 1000Hz relatively linear so that the use of MFCC work is not good enough at frequencies below 1000 Hz[39].

We present a samples from two different cries to see how different is for every cry the plot in figures (42,43)

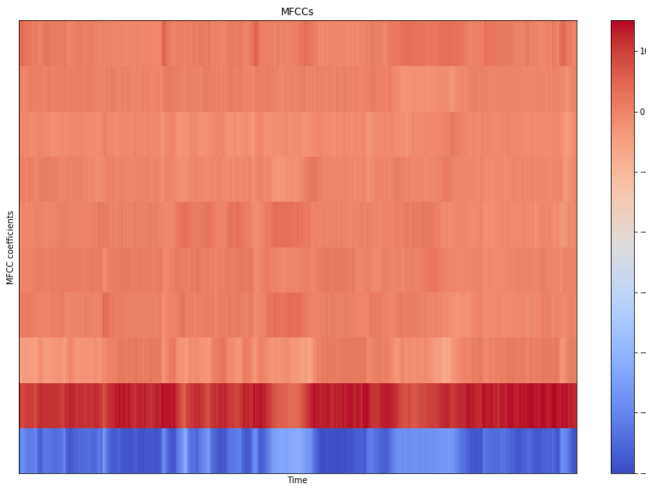


Figure 42: MFCC for spkr15_F_S1a_cry06

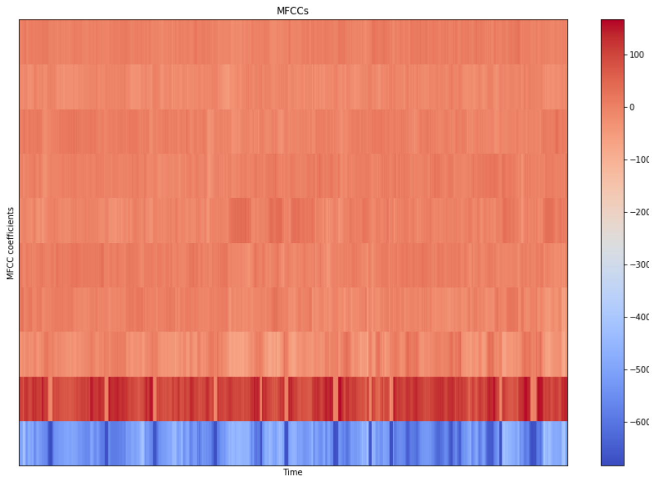


Figure 43: MFCC for spkr09_M_S1a_cry14

MFCC has some advantages in extracting feature that used for analysis of baby crying classification[40], such as:

- It can identify the character of the sound so that it can determine the pattern of sound.
- The output vector has a small data size but does not remove the noise characteristics in the extraction.
- MFCC works similar to the way of a human listener works in giving their perceptions.

6. Classification phase

As we assume that every baby cry has a specific reasons and different pattern features after tacking the method of MFCC to extract the feature in comparison with the spectrograms we classify using Neural Networks (NN), Convolutional Neural Networks (CNN) and Support Vector Machine (SVM).

6.1. Deep learning with artificial neural network "ANN"

Artificial neural networks (ANNs) are non-algorithmic, non-numeric and parallel information processing systems. ANNs are similar to a biological nervous system and consist of layers of parallel units called neurons[41].

A large number of weighted links through which signals links these neurons together or information can pass. A neuron receives input through its afferent connections, combines inputs, generally performs nonlinear operations, and outputs results[41].

a. Input or Visible Layers

The bottom layer that takes input from your dataset is called the visible layer, because it is the exposed part of the network. Often a neural network is drawn with a visible layer with one neuron per input value or column in your dataset. These are not neurons as described above, but simply pass the input value through to the next layer[42].

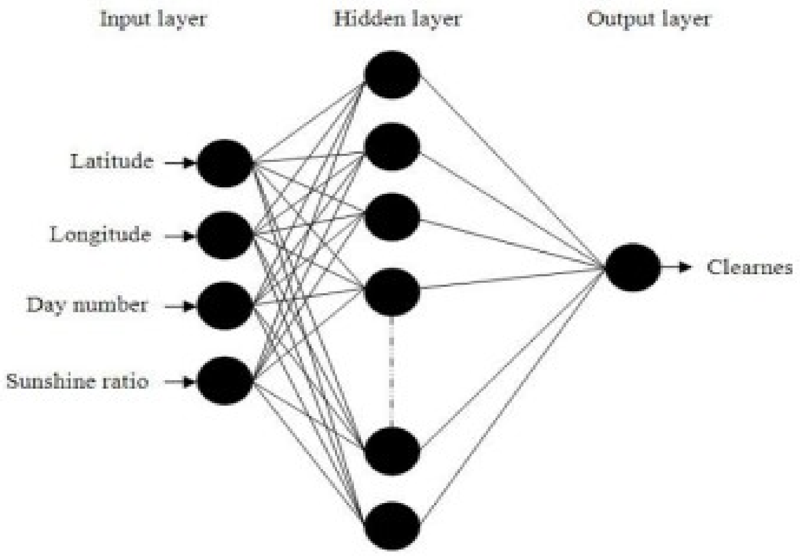


Figure 44: Representation of the layers of NN

b. Hidden Layers

Layers after the input layer are called hidden layers because those are not directly exposed to the input. The simplest network structure is to have a single neuron in the hidden layer that directly outputs the value.

c. Output Layer

The final hidden layer is called the output layer and it is responsible for outputting a value or vector of values that correspond to the format required for the problem. The choice of activation function in the output layer is strongly constrained by the type of problem that we are modeling[42].

6.1.1. Multi-Layer Perceptron

This is an area of research on how to use simple models of the biological brain to solve difficult computational tasks, such as the predictive modeling tasks we see in machine learning. The goal is not to build realistic models of the brain, but to develop robust algorithms and data structures that we can use to model difficult problems[42].

The power of neural networks comes from their ability to learn the representation in your training data and how best to relate to it output variables, you want to predict. In this sense, neural network learning is a mapping. Mathematically, they can learn any mapping function and have been shown to be a general approximation algorithm[42].

The predictive capability of neural networks comes from the hierarchical or multi-layered structure of the networks. The data structure can pick out (learn to represent) features at different scales or resolutions and combine them into higher-order features. For example from lines, to collections of lines, to collections of shapes[42].

a. **Neurons**

The building blocks of a neural network are artificial neurons. These simple computing units have weighted input signals and use activation functions to generate output signals[43].

b. **Neuron Weights**

Each neuron has a bias, which can be thought of as an input that always has 1.00, which must also be weighted. One for each input and bias[44].

Weights are often initialized to small random values, such as values from 0 to 0.3, and while more complex initialization schemes can be used, larger weights represent increased complexity and fragility. It is best to keep the weights in the network small and regularization techniques can be used. The weighted input is summed and passed through an activation function, sometimes called a transfer function[42].

Note: A neuron may have two inputs in which case it requires three weights.

c. **Activation**

The activation function is the simple mapping of the weighted inputs to the neuron's output when the weighted inputs are summed[42].

Historically, simple step activation functions have been used, where if the sum of the inputs is above a threshold value, say 0.5, then the neuron outputs a value of 1.0, otherwise it outputs 0.0[42].

Traditionally, non-linear activation functions are used. This allows networks to combine inputs in more complex ways, and in turn provides a richer capability in the functions they can model[42].

Nonlinear functions, such as logarithmic functions, also known as Sigmundian functions, have an S-shaped distribution of values between output 0 and 1, and hyperbolic tangent functions, also known as “tanh”, have the same distribution of output in the range -1 to +1.

Recently, rectifier activation functions have been shown to provide better results.

Note: It is called the activation function because it determines the threshold at which neurons are activated and the strength of the output signal.

6.1.2. **Multiclass classification with MFCC features extraction:**

In this phase, we have study for classification using MFCC features extraction multiclass method. We will start with the dataset preprocessing (libraries used…etc.)

a. Libraries used

- Import JSON: This library is for saving the extracted data from the wav files into parameters of MFCC.
- Import Os: This one is used for finding the path and directory.
- Import math: To use mathematical equations for calculating vectors equations.
- Import Librosa: To load "wav" files and calculate features.
- Import Os.path: To read all the files in the path.

Using MFCC method for extracting parameters into JSON format file. Data preparation for classification using developed functions with specific arguments, we use the number of segments to shop of the data tracks into a specific number so we can increase our data set to the model.

b. Input

```
    """
Extracts MFCCs from cry dataset and saves them into a JSON file along width genre labels.
def save_mfcc(dataset_path, json_path, num_mfcc=13, n_fft=2048, hop_length=512, num_segments=2):
    :param dataset_path (str): Path to dataset.
    :param json_path (str): Path to json file used to save MFCCs.
    :param num_mfcc (int): Number of coefficients to extract.
    :param n_fft (int): Interval we consider to apply FFT. Measured in # of samples.
    :param hop_length (int): Sliding window for FFT. Measured in # of samples.
    :param: num_segments (int): Number of segments we want to divide sample tracks into.
    :return:
        """
```

c. Extract parameters in same path

- Input []:

```
  data = {
        "mapping": [], it a array that contain the name of classes used in the path files
        "mfcc": [], it a vector containes the mfcc value the input data
          "labels": [], contains the output data that contains a label for every mfcc existed in the mapping
  }
signal, sample_rate = librosa.load(file_path, sr=SAMPLE_RATE,res_type='kaiser_fast')
used to load the MFCC from the WAV files ,
then we will analyze a slice of the signal from the start of sample to the end.
mfcc = librosa.feature.mfcc(signal[start:finish], sample_rate, n_mfcc=num_mfcc, n_fft=n_fft, hop_length=hop_length)
```

- output []:

After loading the extracted data and having then in sub file as the following picture:

Figure 45: JSON DATA FILE

We can say we have our data preprocessed and ready for trying save on the dictionary at the JSON file.

6.1.3. MFCC steps with Multiclass classification

We start by defining our data from the JSON file and we direct our working flow to its path using python modules mentioned previously.

Then we have to converts the obtained list into Numpy array for both targets and inputs, we reach the part for splitting into 30% for testing and 70% for training.

Next step is building the network architecture using tensorflow frame with keras, using a simple multilayer perception including an input layer and three hidden layers with output layer fully connected and dense.

For the input layer will be using flatten layer it takes dimensional array of the MFCC and flatten it into vectors for the hidden layer will be using dense with 512 neurons and activation function of RELU [1]because it gives us a faster convergence and reduce the vanishing gradient of multiplication from the output toward the input.

For the second layer 256 neurons and for the third layer is 64 neurons, the output layer we are using dense layer with 8 neurons according to the classes number we have and activation function this time is softmax that normalize the output into one neurons with the highest prediction .

[1] RELU: **Rectified linear units** find applications in computer vision and speech recognition using deep neural nets.

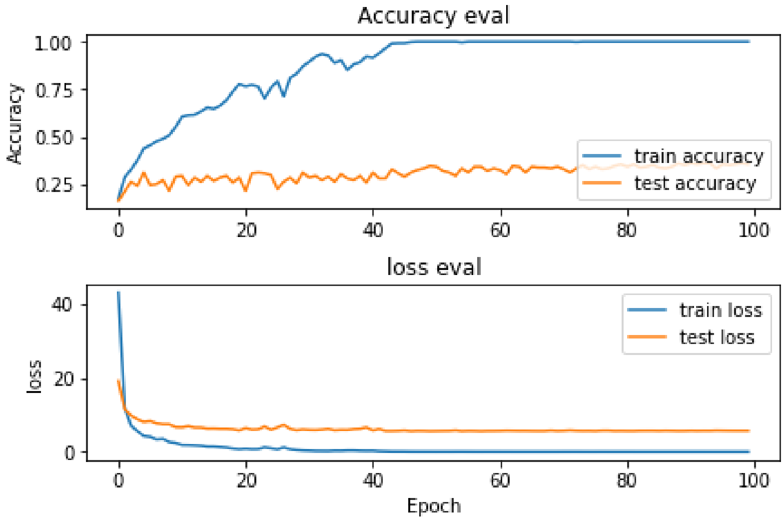


Figure 46: Plot of train and test accuracy, train and test loss

For the network compiling, we are using ADAM optimizer[2] and cross entropy[3] for calculating the errors. The network training with epochs=100 and batch_size=32, this choice is not very memory intensive consuming because it a mini-batch[4].

In the (figure 46), we see that results are presenting an over fitting and we notice a very big difference between the training that reaches 100% and the test accuracy is very low 42% at the accuracy_eval print and similar case for the loss_eval print.

To solve the over fitting issue by applying two dropout methods and regularization from tensorflow:

Dropout[5] method is one of the effective methods to solve over fitting without changing data or the model, and it work by dropping neurons. While training means it use only a part of the network stochastically, it can be done by any hidden layer (second or third) at once. This is how we increase the network robustness, using a hyper parameter named by “The dropout probability” in the universal rule it between 0.1-0.5 in our dropout we are relying on 0.3 it gives us the best results as shown in the (figure 47).

---

[2] Adam optimizer: is an **optimization** algorithm that can be used to update network weights iterative based in training data.

[3] Cross entropy: Cross-entropy loss, or log loss, measures the performance of a classification model whose output is a probability value between zero and one.

[4] The typically **mini-batch sizes** are 64, 128, 256 or 512. In the end, make sure the **mini-batch** fits in the CPU/GPU.

[5] Dropout: is a technique where randomly selected neurons are ignored during training. They are “dropped-out” randomly. A Simple Way to Prevent Neural Networks from Over-fitting,

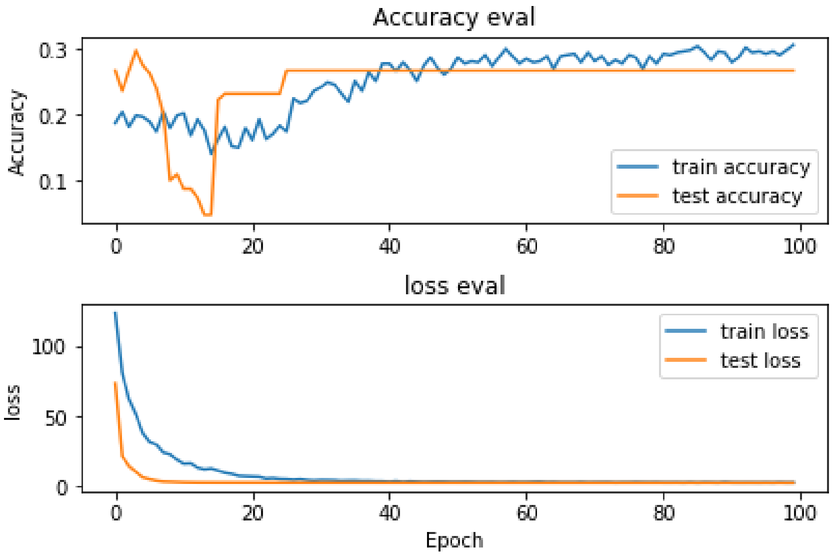


Figure 47: Plot without over fitting train and test accuracy, train and test loss

Stimulatingly we are solving the problem also by regularization technic and it very effective by adding a penalty to error function, that only punish the higher weights.

In regularization[6], we have two types L1 and L2 mostly used in deep learning, L2 commonly used to minimalize the squared value of the weights and less robust to out layers.

In opposite it can learn a quite complex patent. The choice it depends on the application need, in our case we are using:

$$E(p, y) = \frac{1}{2}(p - y)^2 + \lambda \sum w_i^2$$

$$w_i: weights;$$

$$\lambda: \textit{hyper parameter optimizer };$$

$$(p - y)^2: error$$

Equation 12: Regularization equation

[6] Regularization: is a **technique**, which makes slight modifications to the learning algorithm such that the model generalizes better. This in turn improves the model's performance on the unseen data as well.

a. **Model summary ()**

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
flatten (Flatten)            (None, 247)               0
_________________________________________________________________
dense (Dense)                (None, 512)               126976
_________________________________________________________________
dropout (Dropout)            (None, 512)               0
_________________________________________________________________
dense_1 (Dense)              (None, 256)               131328
_________________________________________________________________
dropout_1 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_2 (Dense)              (None, 64)                16448
_________________________________________________________________
dropout_2 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_3 (Dense)              (None, 8)                 520
=================================================================
Total params: 275,272
Trainable params: 275,272
Non-trainable params: 0
_________________________________________________________________
```

Figure 48: MLP Model summary

b. **Results:**

Test accuracy: 0.28947368

Target: 1, Predicted label: [1]

c. **Model plot and definitions:**

Defining the model:

- keras.Sequential model (Flatten, Dense,Dropout) :
    - Input numpy vectors shape for different MFCCs
    - Add input_faltten layer with 247 shape output till next dense reached all neurons layers.
    - Involve second layer of Dense having 512 neurons with activation function "relu" for faster convergence operations, until dense layer not reached.
    - Adding Keras regularize type "L2" with 0,001 learning rate, with drop rate of 0.3 to avoid overfitting
    - Add fully connected layer with Dense classes Perform Relu activation through 256 neurons with same dropout and regularize.
    - Adding same characteristics for third Dense layer with 64 neurons.
    - Final Dens layer reaching all neurons of 8 classes with "softmax" activation.

- Define the accuracy metric and compile the model with ADAM optimizer.

- Viewing model_configuration :
    - SequentialModel(get_config, get_weights)
    - View generated model configuration.
    - View model input and output shape.
    - Assign the weights to individual layers.
    - Model is ready to train once weights are updated.

- Training :
    - Perform model fitting using "32"batch size and epochs "100".
    - Validation of data also tested till training completed.

- Visualizing losses and accuracy
    - Visualize model loss and validation loss with obtained accuracy until epochs reaches 100.

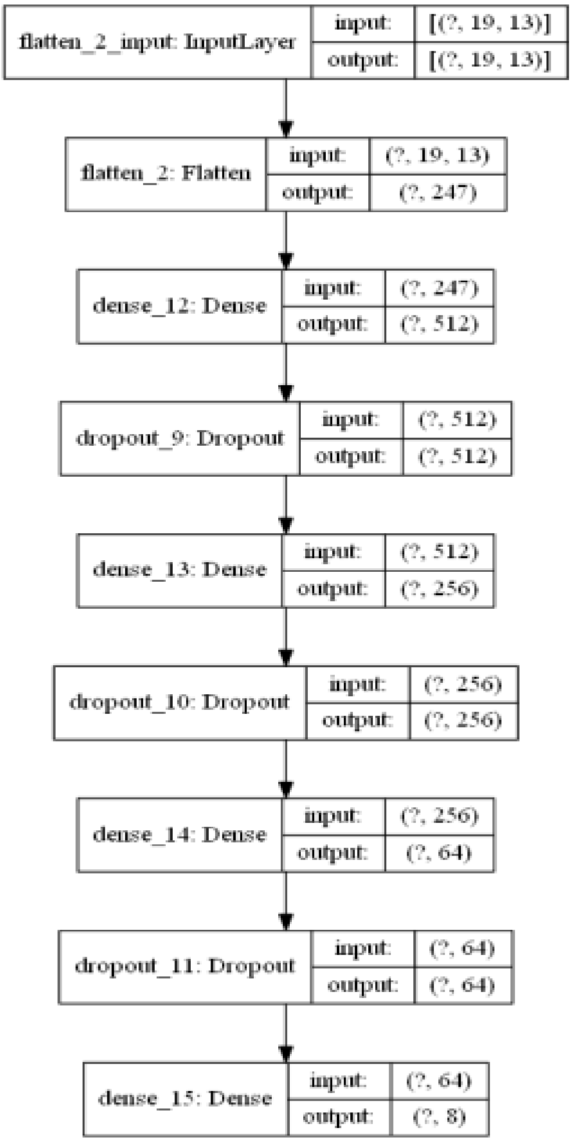


Figure 49: ANN model

6.2. Deep learning with Convolutional Neural Networks

The Convolutional Neural Networks (CNN) taking as entry a spectrogram and the parameters of the MFCC considered as an image on which are different types of structure.

The audio clips have a sample rate of 48000 Hz and duration of about ~3 secs to 15 secs. This means there are about 48000*DURATION numbers per second representing the audio data.

We take a fast Fourier transform (FFT) of a 2048 sample window, slide it by 512 samples and repeat the process of the DURATION (sec) clip. The resulting representation can be shown as a 2D image and is called a Short-Time Fourier Transform (STFT).

Since humans perceive sound on a logarithmic scale, we will convert the STFT to the **Mel-scale** MFCC and to spectrogram. **The Librosa** library lets us load an audio file and convert it to a Mel spectrogram and spectrograms[45].

We precede these steps on the following diagram:

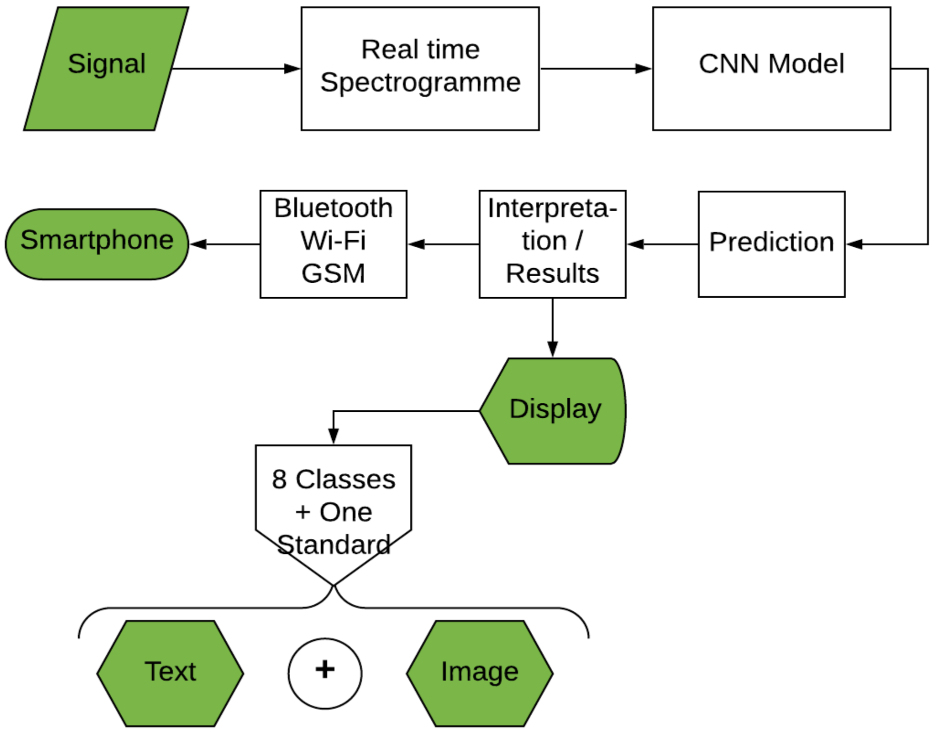


Figure 50: Block diagram of classification phase using CNN model.

6.2.1. Data pre-processing

a. Description

Data preprocessing is an important step in the data mining process. The phrase "garbage in, garbage out" is particularly applicable to data mining and machine learning projects. Data-gathering methods are often loosely controlled, resulting in out-of-range values, impossible data combinations, missing values.

b. Split to train_test files

We start by loading the data file that contain 8 classes sub files(Figure 52) using the “librosa.load” “plt.specgram” to perform the previous step to save the spectrograms(Figure 52) files and splitting it into sub files (80% for train,20% for test) (Figure 53) using “split_folders.ratio” with the same path preserved.

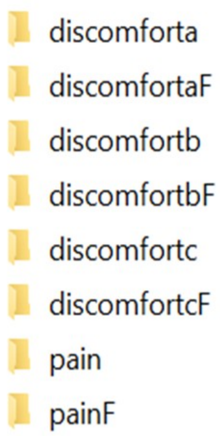


Figure 52: sub files of the eight classes in both train and test contain spectrogram

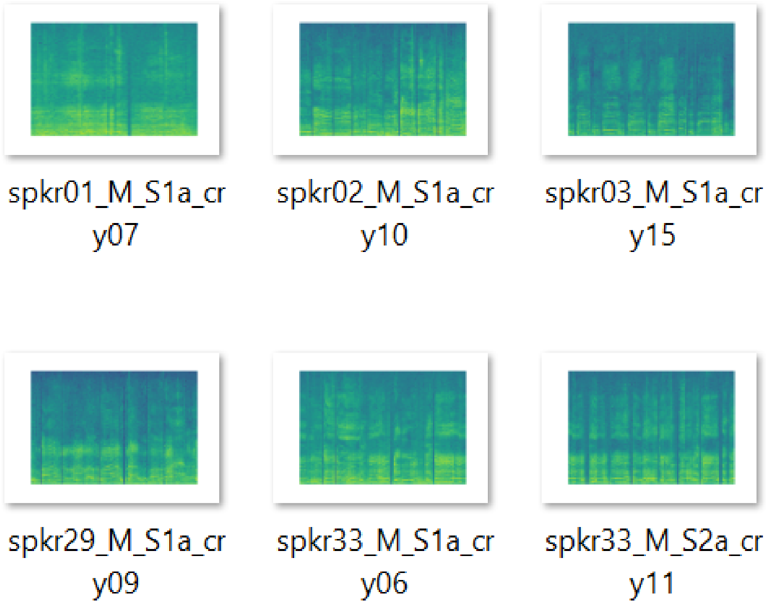


Figure 52: Spectrogram samples taken from sub file "train" class discomfort a

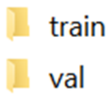


Figure 53: Sub files of the train and test of the dataset contain spectrogram

c. Generating new data

The step for generate new data for purpose augmentation to use it in the learning and testing using the **"ImageDataGenerator ":** Generate batches of tensor image data with real-time data augmentation. The data will be looped over (in batches).

Appling the next parameters such as:

Rescale all pixel values from 0-255, so after this step all our pixel values are in range (0, 1) to apply some random transformations, and apply a zoom. After "train/test_datagen.flow_from_directory()" used with parameters :

- target_size: resizing images to the new size;
- batch_size: number of the image yield per batch;
- class_mode='categorical' ;because we have several classes(more than two) ;
- shuffle = False; we don't want to reorder the data.

The following (figure 54) shows how the generating of the data work:

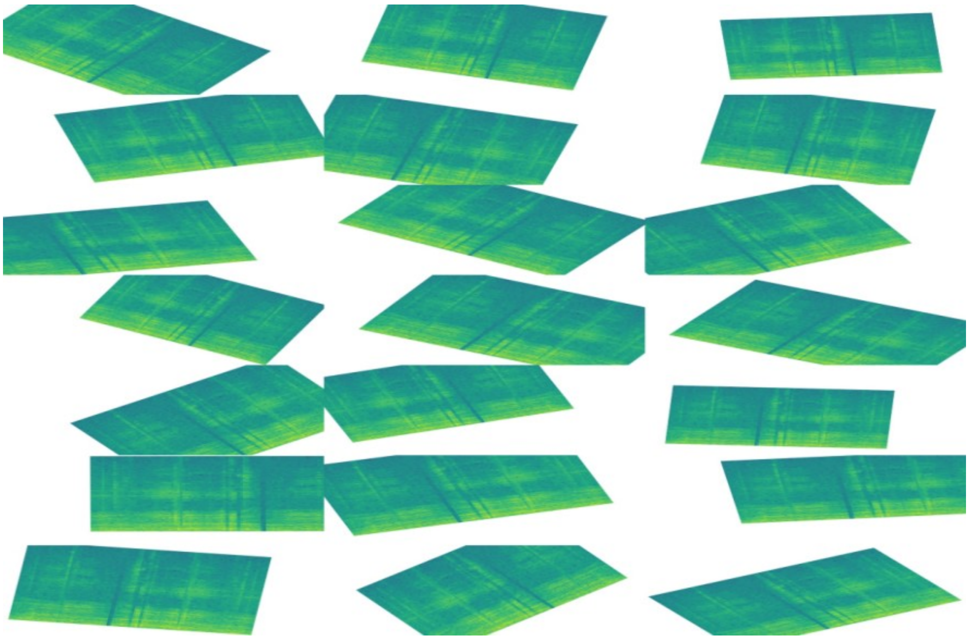


Figure 54: Generated images for one spectrogram

After having the generated data augmentation heading to constructing the conventional neural network using three layers as the optimal based on previous work “lavner2016ML”.

d. **Building the conventional neural network “CNN” model**

Convolutional neural networks (CNN) have wide applications in the fields of computer vision, natural language processing and many others, especially where huge amounts of data have to be processed and classified.

Like ordinary neural networks, they consist of several layers connected by neurons that have learnable weights. Each CNN layer is composed of several filters, applied to outputs provided by the previous layer using the convolution operation.[15]

CNNs learn the filters during the training process, which can be thought of a way to generate important features out of the data. Thus, in contrast to traditional classification algorithms, the lack of dependence on prior knowledge is a major advantage of CNNs. [15]

Therefore, we used the following model for our CNN:

We visualize the model architecture using the ‘model.summary()’ function in Keras using tensorflow background. This is an important step before we get to the model building part. We need to make sure the input and output shapes match our problem statement, hence we visualize the model summary (Figure 55).

```
Model: "sequential_7"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_19 (Conv2D)           (None, 31, 31, 32)        896
_________________________________________________________________
average_pooling2d_19 (Averag (None, 15, 15, 32)        0
_________________________________________________________________
activation_31 (Activation)   (None, 15, 15, 32)        0
_________________________________________________________________
conv2d_20 (Conv2D)           (None, 15, 15, 64)        18496
_________________________________________________________________
average_pooling2d_20 (Averag (None, 7, 7, 64)          0
_________________________________________________________________
activation_32 (Activation)   (None, 7, 7, 64)          0
_________________________________________________________________
conv2d_21 (Conv2D)           (None, 7, 7, 64)          36928
_________________________________________________________________
average_pooling2d_21 (Averag (None, 3, 3, 64)          0
_________________________________________________________________
activation_33 (Activation)   (None, 3, 3, 64)          0
_________________________________________________________________
flatten_7 (Flatten)          (None, 576)               0
_________________________________________________________________
dropout_13 (Dropout)         (None, 576)               0
_________________________________________________________________
dense_13 (Dense)             (None, 64)                36928
_________________________________________________________________
activation_34 (Activation)   (None, 64)                0
_________________________________________________________________
dropout_14 (Dropout)         (None, 64)                0
_________________________________________________________________
dense_14 (Dense)             (None, 8)                 520
_________________________________________________________________
activation_35 (Activation)   (None, 8)                 0
=================================================================
Total params: 93,768
Trainable params: 93,768
Non-trainable params: 0
```

Figure 55: Model summary for CNN

Example for Accessing Individual Layers [‘ conv2d_21’] to expose the parameters for relation layers :

- Output [ ]:

```
print(layers_info['conv2d_21'])
{'name': 'conv2d_21', 'trainable': True, 'dtype': 'float32', 'filters': 64,
 'kernel_size': (3, 3), 'strides': (1, 1), 'padding': 'same',
 'data_format': 'channels_last', 'dilation_rate': (1, 1),
 'activation': 'linear', 'use_bias': True, 'kernel_initializer':
 {'class_name': 'VarianceScaling', 'config': {'scale': 1.0, 'mode': 'fan_avg',
                                              'distribution': 'uniform', 'seed': None}},
 'bias_initializer': {'class_name': 'Zeros', 'config': {}},
 'kernel_regularizer': None, 'bias_regularizer': None, 'activity_regularizer': None,
 'kernel_constraint': None, 'bias_constraint': None}
```

Visualizing building blocks for CNN (Figure56):

- Defining the model:
    - SequentialModel (Convolution2D, Pooling, Dense, Activation)
        - Input image shape for different convolutions
        - Add 2D convolutions with 32 filters and size 3*3 until filter reaches 64
        - Perform sigmoid and relu activations till dense reached 64 layers.
        - Involve mask of size 2*2 through max pooling operations until dense layer not reached
        - Hyper-tune model with drop rate of 0.5 to avoid overfitting
        - Add fully connected layer with Dense classes
        - Perform Softmax activation through SGD optimizer to avoid model decay
        - Define the accuracy metric and compile the model
- Viewing model_configuration :
    - SequentialModel (get_config, get_weights)
        - View generated model configuration.
        - View model input and output shape.
        - Assign the weights to individual layers.
        - Model is ready to train once weights get updated.

- Training
    - Perform model fitting using batch size and epochs.
    - Validation of data also tested till training completed.

- Visualizing losses and accuracy
    - Visualize model loss and validation loss with obtained accuracy until epochs reaches 100.

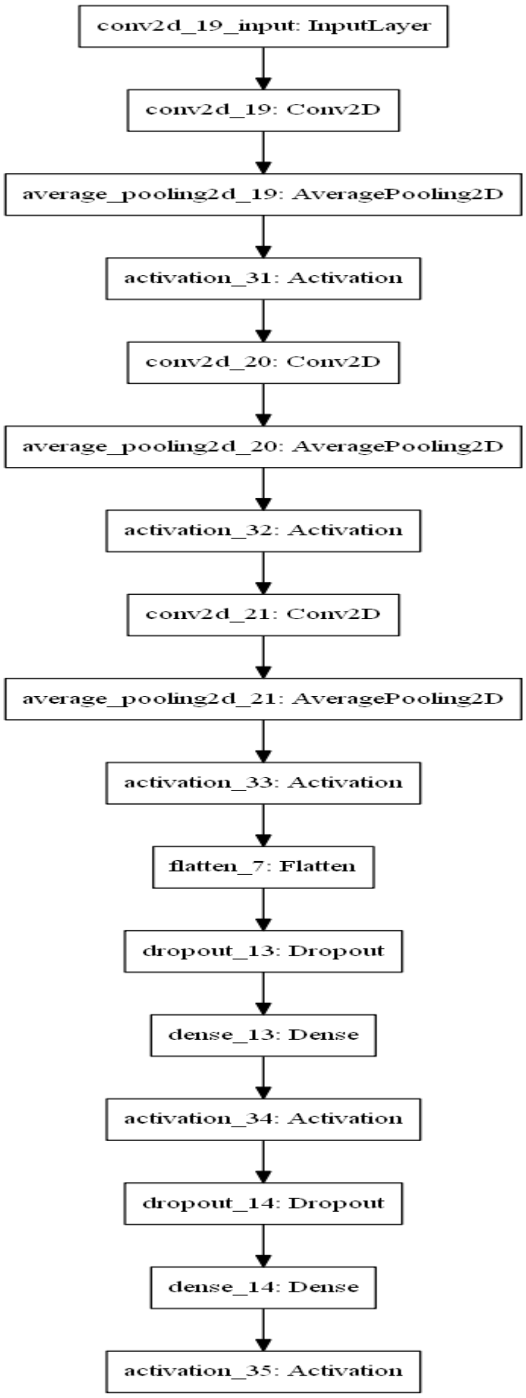


Figure 56: CNN architecture

We have 64 training examples, and our batch size is 32, then it will take two iterations to complete one epoch.

e. Stochastic gradient descent data optimization

Stochastic gradient descent is an iterative method for optimizing an objective function with suitable smoothness properties (e.g. differentiable or sub differentiable). It can be regarded as a stochastic approximation of gradient descent optimization, since it replaces the actual gradient (calculated from the entire data set) by an estimate thereof (calculated from a randomly selected subset of the data) [46].

Especially in big data applications this reduces the computational burden, achieving faster iterations in trade for a slightly lower convergence rate.

- Input [ ]:

```
    """ SGD learning training """
epochs 200
batch_size = 8
learning_rate = 0.01
decay_rate = learning_rate / epochs momentum = 0.9
sgd = SGD(1rulearning_rate, momentum=momentum, decay=decay_rate,
nesterovaFalse)
model.compile(optimizer="sgd", loss="categorical_crossentropy",
metrics=['accuracy'1])
```

After the phase of learning we fit the model with 100 epochs ,we can say that the model is trained and first it will predict the outputs using training inputs then have the evaluation processes that give us the result shown in (figure 57):

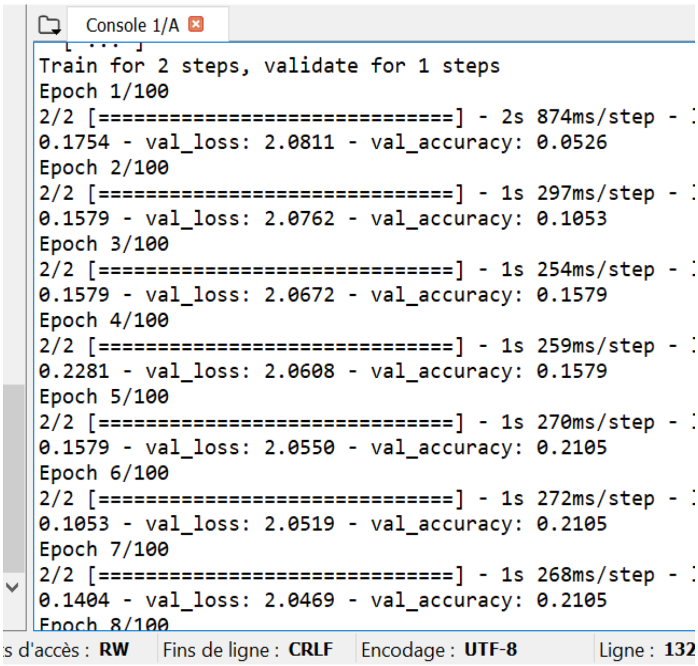
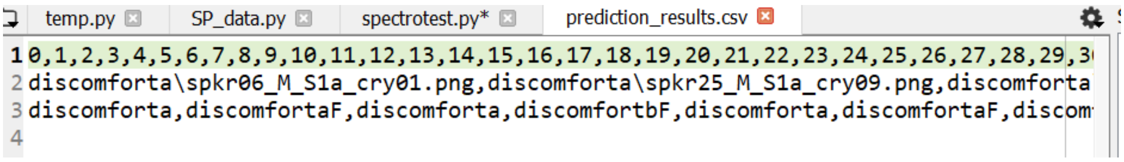


Figure 57: Results of accuracy and loss

We call the **predict_generator** to make some prediction on the data set, we save the results of prediction on file.csv as shown in (figure 58).



temp.py SP_data.py spectrotest.py* prediction_results.csv

```
1 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,3
2 discomforta\spkr06_M_S1a_cry01.png,discomforta\spkr25_M_S1a_cry09.png,discomforta
3 discomforta,discomfortaF,discomforta,discomfortbF,discomforta,discomfortaF,discom
4
```

Figure 58: prediction results

Now we have the model we go to the next step of implementing our model in the raspberry so we can detect the cry and predict then send the notification to the parents. And continue with the last classification method.

6.3. Support vector machine with CWT on IBM Watson

SVM or Support vectors machines is supervised machine learning algorithm requires the labeling of data input for the wavelets sub files, and using the CWT(continues wavelets transform) coefficients to train our machine ,and we did not use the overall data just two files for test (discomfort and pain)[47].

6.3.1. SVM technique Principle

This technique is a two-class classification method that attempts to separate the positive examples of the negative examples in the set of examples. The method then looks for the hyper plane that separates the positive from the negative examples, by ensuring that the margin between the closest positive and negative is maximum.

This ensures that the principle is generalized, as new examples may not be too similar to those used to find the hyper plane but to be located on one side or the other one from the border. The interest of this method is the selection of support vectors, which represent the discriminant vectors by which the hyper plane is determined. The examples used when searching for the hyper plane are no longer useful and only these examples can be used. Support vectors are used to classify a new case, which may be considered as an advantage for this method.

The advantages of support vector machines are[48]:

- Effective in high dimensional spaces.
- Still effective in cases where number of dimensions is greater than the number of samples.
- Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
- Versatile: different Kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

The disadvantages of support vector machines include[48]:

- If the number of features is much greater than the number of samples, avoid over-fitting in choosing Kernel functions and regularization term is crucial.

SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation .

This process is done via IBM WATSON STUDIO environment using virtual machine [PYTHON XS 3.6, 2 vCPU 8G RAM][49].

The privilege of this system is the set on the IBM cloud so we can generate as much data as we want (more than 5GB of free space to use), and it allows you to add the file data to the project and creating a notebook to have easy visualizing of the work step by step each time. You are free to add assets to work with, which make the pastime very fast[49].

6.3.2. Required packages and libraries

We start by setting up the environment, using forge Conda to install required packages and libraries to use for the process and using scikit-learn for classification.

a. Soundfile

SoundFile is an audio library based on libsndfile, CFFI, and NumPy. To install this package with conda run the following instruction: **conda install soundfile.**

b. Sys

This module provides access to some variables used or maintained by the interpreter and to functions that interact strongly with the interpreter. It is always available.

c. Os

The OS module is to interact with your operating system. The primary use I find for it is to create folders, remove folders, move folders, and sometimes change the working directory. You can also access the names of files within a file path by doing listdir(). We do not cover that in this video, but that's an option.

The os module is a part of the standard library, or stdlib, within Python 3. This means that it comes with your Python installation, but you still must import it.

d. Types

This module defines names for some object types that are used by the standard Python interpreter, but not for the types defined by various extension modules.

In addition, it does not include some of the types that arise during processing such as the listiterator type. It is safe to use from types import * — the module does not export any names besides the ones listed here. New names exported by future versions of this module will all end in Type.

Typical use is for functions that do different things depending on their argument types.

e. **ibm_boto3**

The ibm_boto3 library provides complete access to the IBM® Cloud Object Storage API. Endpoints, an API key, and the instance ID must be specified during creation of a service resource or low-level client

f. **Pandas**

In computer programming, pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.

g. **BytesIO**

A BytesIO object isn't associated with any real file on the disk. It's just a chunk of memory that behaves like a file does.

After it returns, you can get any data the library wrote to the file from the BytesIO using the getvalue() method.

h. **ZipFile**

The ZIP file format is a common archive and compression standard. This module provides tools to create, read, write, append, and list a ZIP file. Any advanced use of this module will require an understanding of the format.

This module does not currently handle multi-disk ZIP files. It can handle ZIP files that use the ZIP64 extensions (that is ZIP files that are more than 4 GByte in size).

It supports decryption of encrypted files in ZIP archives, but it currently cannot create an encrypted file. Decryption is extremely slow as it is implemented in native Python rather than C.

i. **Svm**

Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression and outliers detection.

The support vector machines in scikit-learn support both dense (**numpy.ndarray** and convertible to that by **numpy.asarray**) and sparse (any **scipy.sparse**) sample vectors as input.

However, to use an SVM to make predictions for sparse data, it must have been fit on such data. For optimal performance, use C-ordered **numpy.ndarray** (dense) or **scipy.sparse.csr_matrix** (sparse) with **dtype=float64**.

j. Accuracy_score

Accuracy classification score. In multi label classification, this function computes subset accuracy.

6.3.3. Split train/test

Split arrays or matrices into random train and test subset. Split train/test is a quick utility that wraps input validation and **next(ShuffleSplit().split(X, y))** and application to input data into a single call for splitting (and optionally subsampling) data in a one-liner. It has multiple parameters[50]:

a. test_sizefloat or int, default=None

If float, should be between 0.0 and 1.0 and represent the proportion of the dataset to include in the test split. If int, represents the absolute number of test samples. If None, the value is set to the complement of the train size. If train_size is also None, it will be set to 0.25.[50]

b. train_sizefloat or int, default=None

If float, should be between 0.0 and 1.0 and represent the proportion of the dataset to include in the train split. If int, represents the absolute number of train samples. If None, the value is automatically set to the complement of the test size.[50]

6.3.4. Principal Component Analyze

Principal component analyze (PCA) is unsupervised ,nonparametric statically technique that let you take from several parameters components only the parameters that you are using in our case continuous wavelet transform coefficients, by reducing dimensionality for the complexity of computational algorithms in machine learning to make the perform faster[51].

We will explain the calculation steps:

- First: We start by centering dataset in 3D plot (X; Y; Z) into zero point of the coordinate system so we can draw perpendicular line to calculate the covariance of the extracted matrix, from the eigenvectors that are principal component to find the greatest covariance.
- Second: The eigenvectors provide the proper variance.
- Third: Select the new data dimension and we project them.

In PCA, linear dimensionality reduction using Singular Value Decomposition of the data to project it to a lower dimensional space. The input data is centered but not scaled for each feature before applying the SVD.[51]

Moreover, the principal parameter is the **n_components**, there are more parameters but all of them are stay in default:

a. n_componentsint, float, None or str

Number of components to keep. if **n_components** is not set all components are kept:

n_components == min(n_samples, n_features)

If **n_components == 'mle' and svd_solver == 'full', Minka's MLE** is used to guess the dimension. Use of **n_components == 'mle' will interpret svd_solver == 'auto' as svd_solver == 'full'**.

If **0 < n_components < 1 and svd_solver == 'full'**, select the number of components such that the amount of variance that needs to be explained is greater than the percentage specified by **n_components**[51].

If **svd_solver == 'arpack'**, the number of components must be strictly less than the minimum of **n_features** and **n_samples**.

Hence, the None case results in: **n_components == min(n_samples, n_features) – 1**

After setting up the environment and install all the packages and libraries by the following instruction:

```
"setting up the enviroment with proper packages and backgrounds"
!conda config --add channels conda-forge
!conda install -y libsndfile
!conda install pywt
!conda install soundfile
!conda install librosa

"import path and systemm libraries"
import os
import sys
import types
import ibm_boto3

"import vector and math and plot libraries"
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

"import libraries for reading and encrypting file in this case (file.zip)"
from io import BytesIO
from zipfile import ZipFile
from botocore.client import Config

"calssification phase train and test with SVM"
from sklearn.model_selection import train_test_split
from sklearn import svm
from sklearn.metrics import accuracy_score

"loading and resamling audio (fil.wav)"
import soundfile as sf
import librosa

"for extracting CWT,DWT coefccients"
import pywt
```

Then, we import our audio data to train and classify.

- Input[]:

```
ZipFile.namelist(zip_file)
```

- Output[]:

```
['dataset/',
 'dataset/pain/',
 'dataset/pain/spkr09_M_S1d_cry12.wav',
 'dataset/pain/spkr28_M_S1d_cry12.wav',
 'dataset/discomfortb/',
 'dataset/discomfortb/spkr05_M_S1b_cry07.wav',
 'dataset/discomfortb/spkr06_M_S1b_cry01.wav',
 'dataset/discomfortb/spkr06_M_S2b_cry14.wav',
 'dataset/discomfortb/spkr09_M_S1b_cry09.wav',
 'dataset/discomfortb/spkr14_M_S1b_cry04.wav',
 'dataset/discomfortb/spkr25_M_S1b_cry07.wav',
 'dataset/discomfortb/spkr28_M_S1b_cry09.wav',
 'dataset/discomfortb/spkr29_M_S1b_cry07.wav']
```

Reading the file information and resampling all data into one sapling rate for next process (for this part Librosa v 0.6.3 can run)

- Input[]:

```
audio_data = []
labels = []
sampling_rate = []
file_names = []
for index in range(len(audio_data)):
    if (sampling_rate[index] == 48000):
        audio_data[index] = librosa.resample(audio_data[index], 48000, 44100)
        sampling_rate[index] = 44100
```

Note: if you already have processed data no need for this step

Now, we will transform data .wav into mono channel, and prepare the data for plotting.

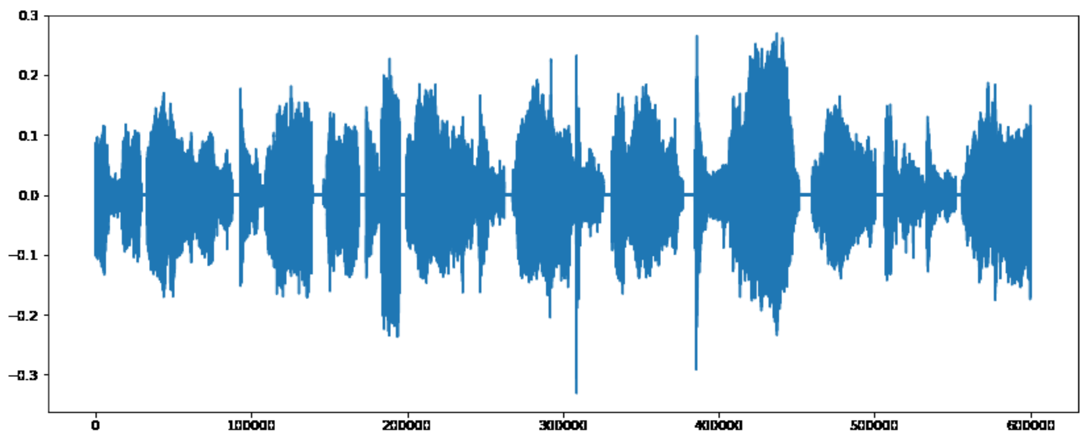
- Input[]:

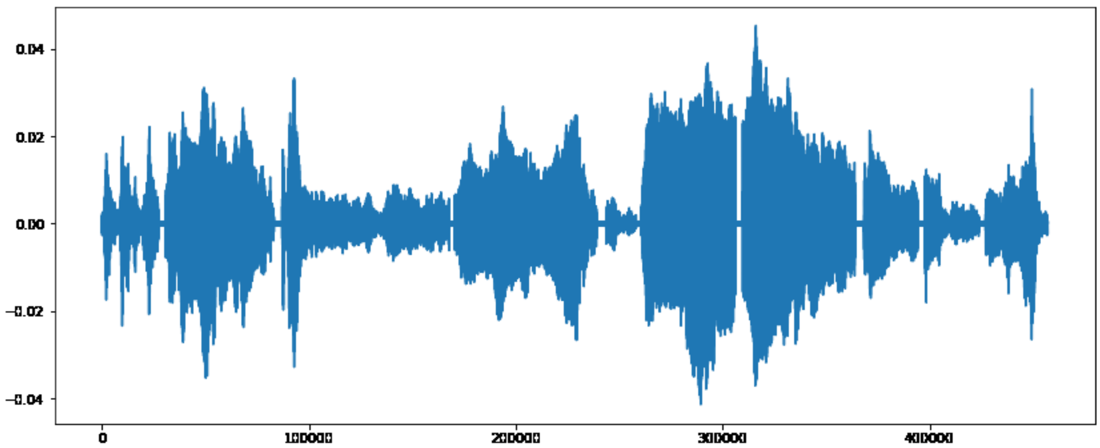
```
def to_mono(data):
    if data.ndim > 1:
        data = np.mean(data, axis=1)
    return data

for index in range(len(audio_data)):
    audio_data[index] = to_mono(audio_data[index])

"plot sample 1 from subfile 1"
fig = plt.figure(figsize=(14,6))
plt.plot(audio_data[1])
"plot sapmle 4 from subfile 2"
fig = plt.figure(figsize=(14,6))
plt.plot(audio_data[4])
```

- Output[]:





Then scaling data, to use only the needed information for extracting CWT coefficients by using the following instructions:

- Input[]:

```
scales = np.arange(1, 101)
coeff1, freqs1 = pywt.cwt(audio_data[1][:25000], scales, 'morl')
coeff2, freqs2 = pywt.cwt(audio_data[4][:25000], scales, 'morl')

#plot CWT energy
plt.figure(1, figsize=(20,10))
plt.subplot(121)
plt.imshow(coeff1, cmap='coolwarm', aspect='auto')
plt.subplot(122)
plt.imshow(coeff2, cmap='coolwarm', aspect='auto')
plt.show()
```

- Output[]:

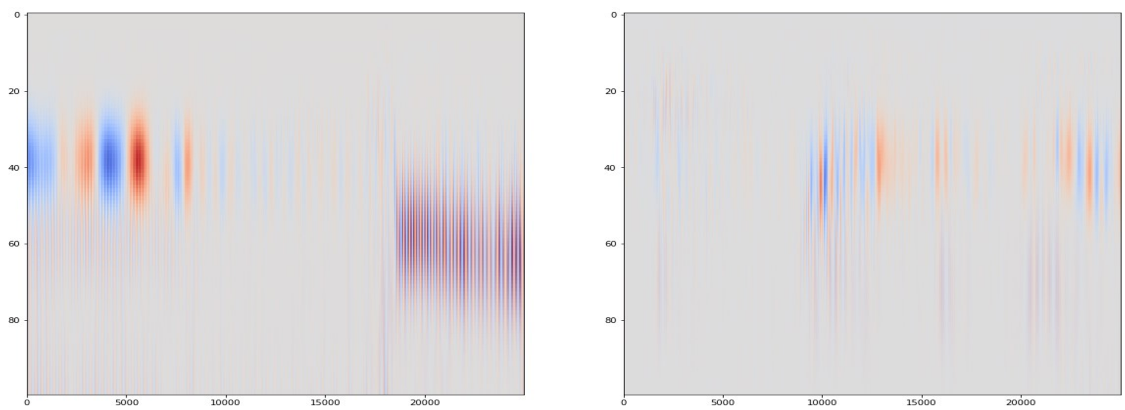


Figure 59: pain1 and pain2

Moreover, we will use the 3D plot to visualize the power from all angles.

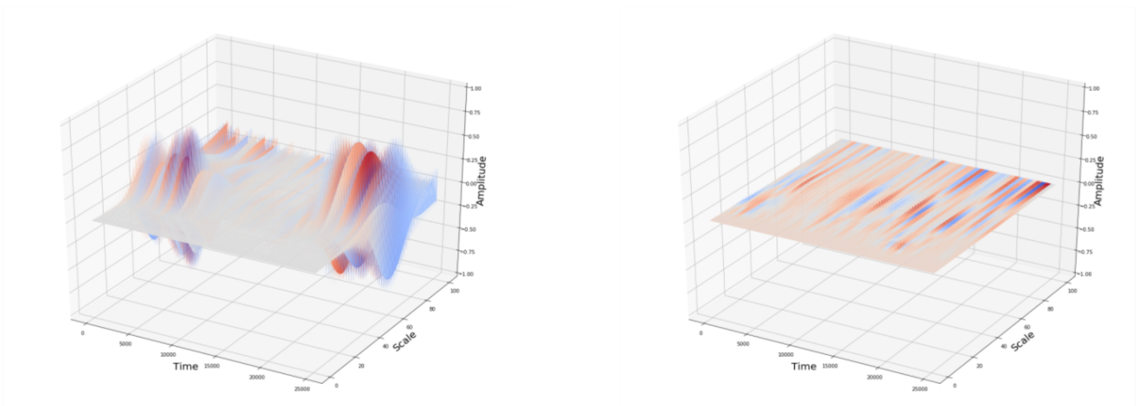


Figure 60: Pain1

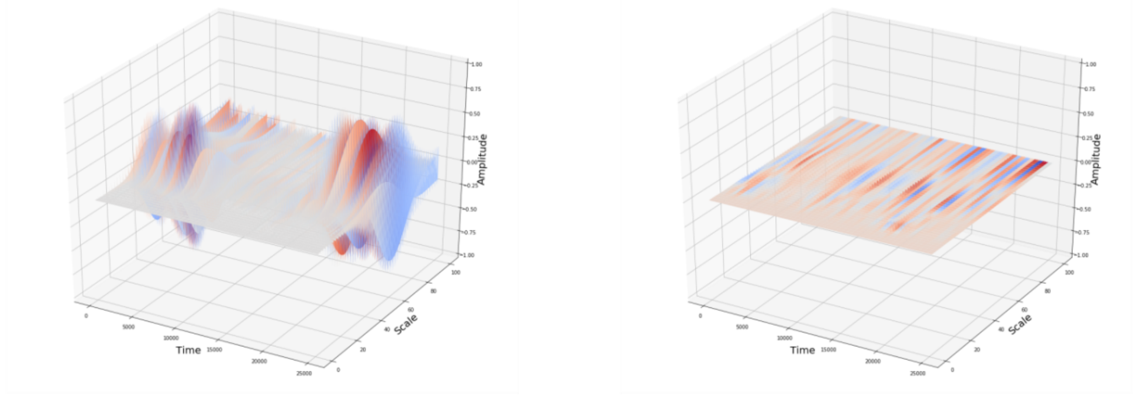


Figure 61: Pain2

- Input[]

```
"principal coponent analyse reducing vector dimensions"
from sklearn.decomposition import PCA

pca = PCA(n_components=1)
features = np.empty((0,100))
for ind in range(len(audio_data)):
    print('.', end='')
    coeff, freqs = pywt.cwt(audio_data[ind][:25000], scales, 'morl')
    features = np.vstack([features, pca.fit_transform(coeff).flatten()])

"split data to (20% test,80% train)"
X_train, X_test, y_train, y_test = train_test_split(
  features, labels, test_size=0.20, random_state=1234)

"calssification with SVM"
clf = svm.SVC()
clf.fit(X_train, y_train)

"predictiong and learning accuracy"
y_pred = clf.predict(X_test)
print("Accuracy : %.2f%%" % (accuracy_score(y_test, y_pred) * 100))
```

- Output[]:

```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
  kernel='rbf', max_iter=-1, probability=False, random_state=None,
  shrinking=True, tol=0.001, verbose=False)

Accuracy : 100.00%
```

7. Comparison & Conclusion

In this table, we show a comparison between the used methods. It is true that SVM give us the best results but because of the uploading speed, we cannot use it to fully train the model with the big data we have. The rest two methods have a low accuracy, which is just due to low amount of data. In the next chapter we will decide the final model to implement in the Raspberry Pi.

Table 13: Comparison between trained models

	Technics architecture	Performance (accuracy %)
MFCC-MLP	NN(Neural network): Multi-layer perceptron	28.9%
Spectrograms images	CNN: conventional neural network	21.05%
SVM (CWT)	Support vector machine & PCA (Principal component analyze)	100%

In this third chapter, it was imperative to use different technics to know the best one for our application by trying several methods from signal processing (STFT), features extractions methods (Spectrogram, MFCC, CWT, DWT, LPC) to optimal deep learning model (CNN, NN) or SVM to apply it on baby cry database.

In addition, we try multiple circuits for creating proper hardware device to use it for sound detection, passing by the test phase via Arduino and testing different sound modules.

This experience shows us how difficult and sensitive to work with audio signal processing and applying deep learning techniques on audio signals.

Chapter IV

Design and implementation

1. Introduction

In this chapter, we will connect the detection system (Based on microphone and sound card) with the Raspberry Pi, with the implementation of the software (Script) in the Raspberry Pi model B.

The detection of the sound is based on our algorithms that functions using parameters such as amplitude and frequency range extracted specially from the baby cry database to have the primary information's to lunch the system.

In addition, we will make sure that the framework systems functioning very well, specially that we are using NODE-RED to perfection wiring and to have an online deploying for real time processing.

Before getting into this final phase, we have a testing phase on the Arduino before anything else so we can make sure that we have all the right components along with their libraries installed on Raspberry Pi.

2. Conception of the detection system

2.1. Test phase with Arduino

In this step, we will try to test the functionality of the components before using them on the Raspberry Pi to make sure that everything is perfectly ready for the integration.

Therefore, what we are doing is testing the ADC (ADS1115) with Arduino then testing the Arduino with the sound detector (MAX9814), we will explain it in more details in the following steps.

2.1.1. Required components (Arduino)

- Arduino Uno
- Ads1115
- Sound sensor (MAX9814)
- Wires (jumpers)
- Arduino IDE

2.1.2. System wiring for Arduino

a. Arduino with ADS1115/1011

The wiring has six connections

- 5V pin of the Arduino with the pin VDD
- GNG pin of the Arduino with GND
- Pin A5 (SCL) (Arduino) with pin 3 SCL
- Pin A4(SDA) (Arduino) with pin 4 SDA
- Same GND (Arduino) pin with pin 5 ADDR
- 3.3V Pin 3 (Arduino) with Pin A0

The following figures show the wiring with Arduino.

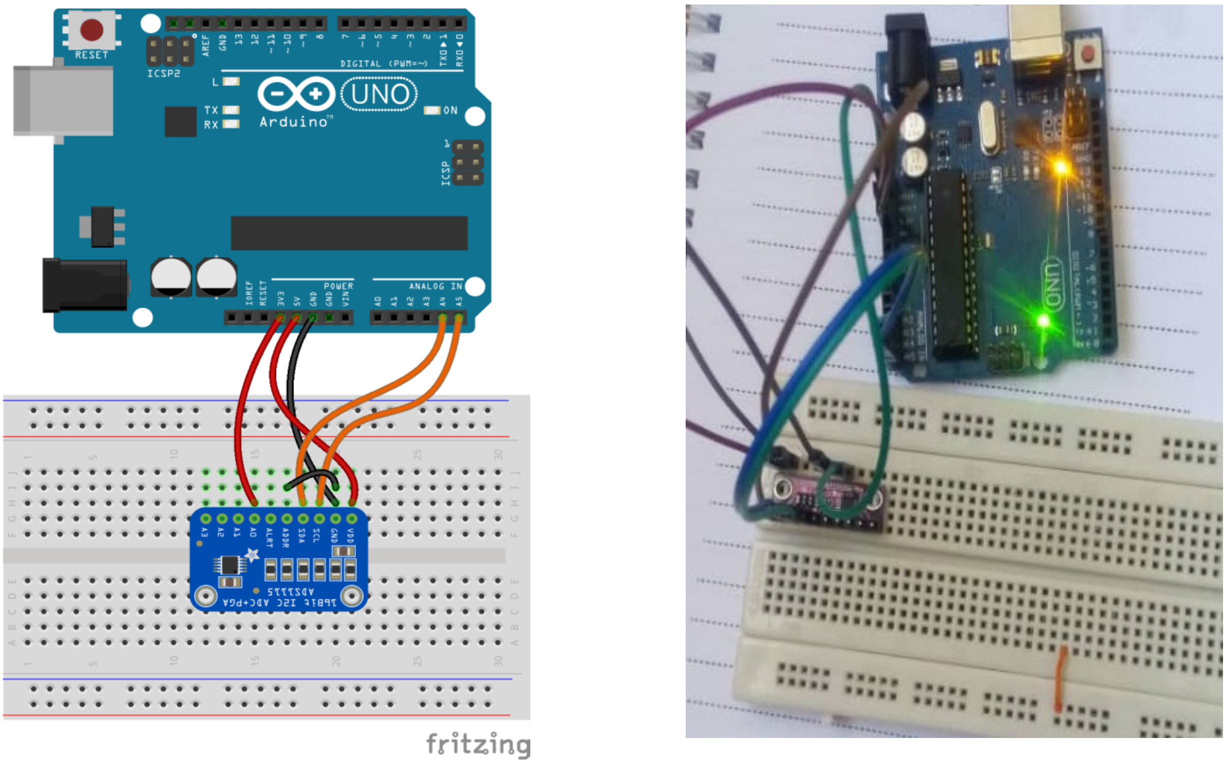


Figure 62: Wiring of ads 1115 with Arduino

b. **Wiring Arduino with MAX9814**

The microphone have three connections, so the wiring is simple:

- GND (Arduino) -> GND.
- AIN0 (Arduino) ->OUT.
- 3.3V (Arduino) -> VCC. VCC can be anywhere from 2.4 to five Volt (DC).

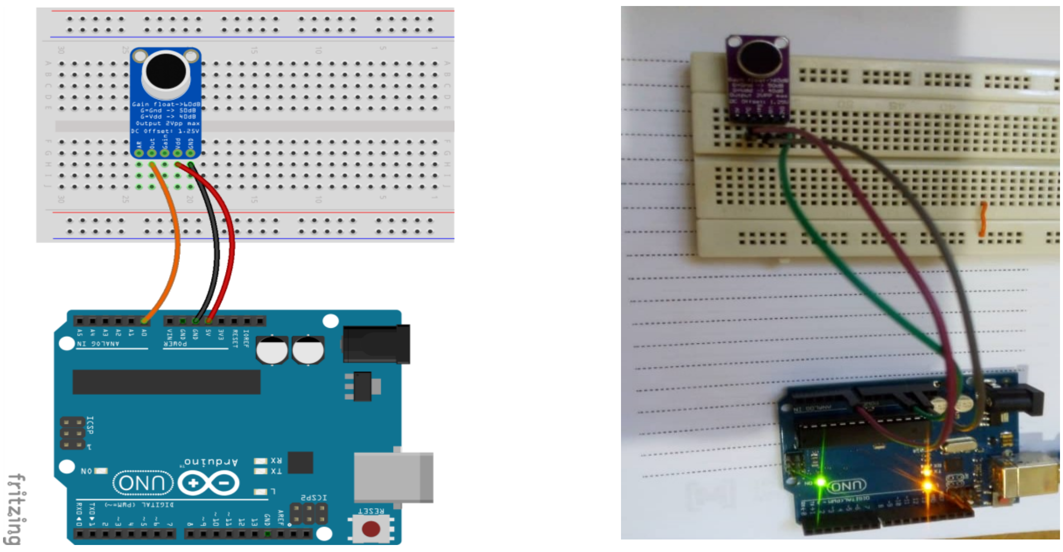


Figure 63: Wiring of microphone with Arduino

For better performance, we use the 3.3v pin because it is the adequate power supplier on the Arduino. The output will have a DC bias of VCC/2, so when it is quiet, the voltage will be a steady VCC/2 (1.65v) and when it low will reach the (3V).

c. System configuration in Arduino:

Start by testing the digital analogue converter with the arduino system in order to use the ADC we need to install required libraries in the arduino sketch guide it very important to upload same version, then by using OpenSource example from adafruit examples it about simple code in differential between AIN0 and AIN1 we have the following plot (we did not use gain function in this example) the results are very satisfactory:

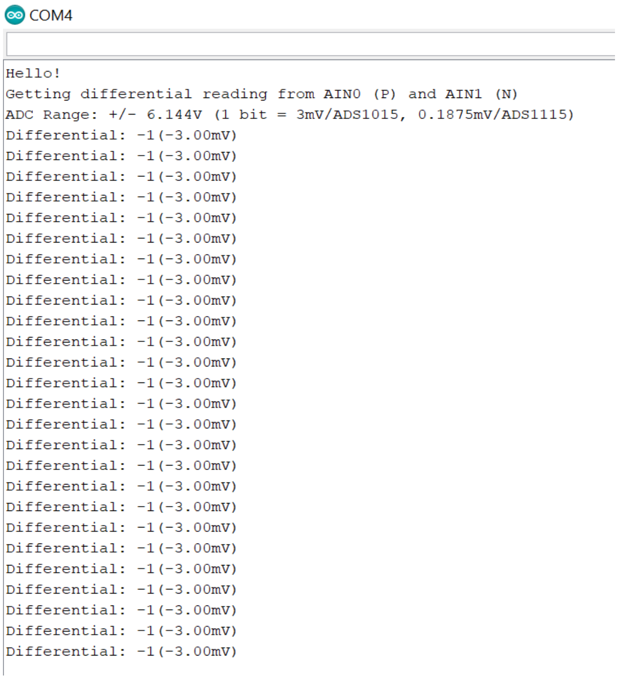


Figure 64: ADS1015 readings test

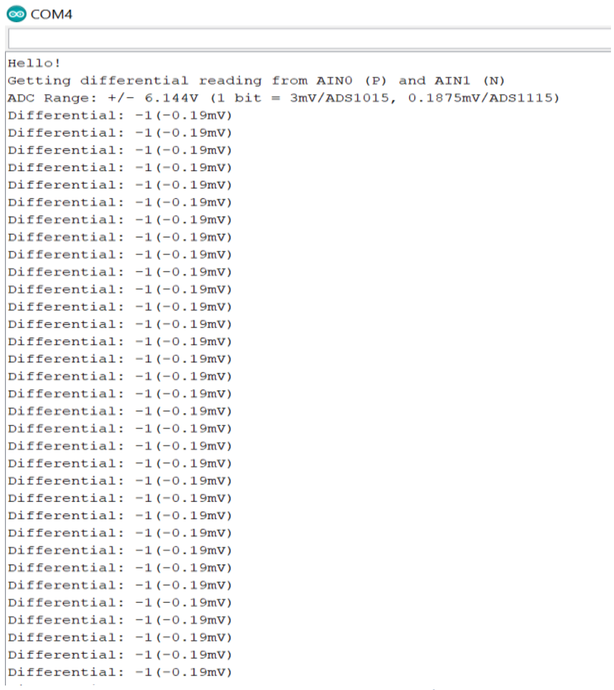


Figure 59: ADS1115 readings test

For testing the MAX9814 using same library of the ADAFRUIT we will implement a simple script from this guide, it about a sound tester manager we adjust the gain according to our needs we can see if it detect the sound will have an output of (0/0.1V) and it increase with the ambient sound of the surrounding.

2.2. Application phase with Raspberry Pi

Before working with the Raspberry Pi, we must do the setup for it by preparing the 16 GB SD card using a computer to install the operating system onto the SD card.

For this setup, we will use the program BalenaEtcher and Raspberry Pi imager.

After setting up the card, we must test it by connecting the Raspberry Pi to a keyboard, a mouse and a monitor, and then power it via a computer or USB wall charger.

Now we connect the Raspberry Pi to Wi-Fi, setting up the raspberry PI Wi-Fi via Ssh (Secure shell connection) / VNC Viewer.

Using VNC viewer allow us to see the Raspberry Pi's desktop remotely in a graphical way, using the mouse as if you were sitting in front of the Raspberry Pi.

Connecting to the Raspberry Pi like this can save on desktop clutter, and having multiple keyboards and mice all over the place.

It also means we can put the Raspberry Pi on different network, but still control it.

2.2.1. **Required components**

- Raspberry Pi 3 model B.
- ADS 1115/1015 adafruit.
- Microphone electret MAX9814 adafruit.
- Printing board.
- Wires.
- Testing board.

2.2.2. **System wiring for Raspberry Pi**

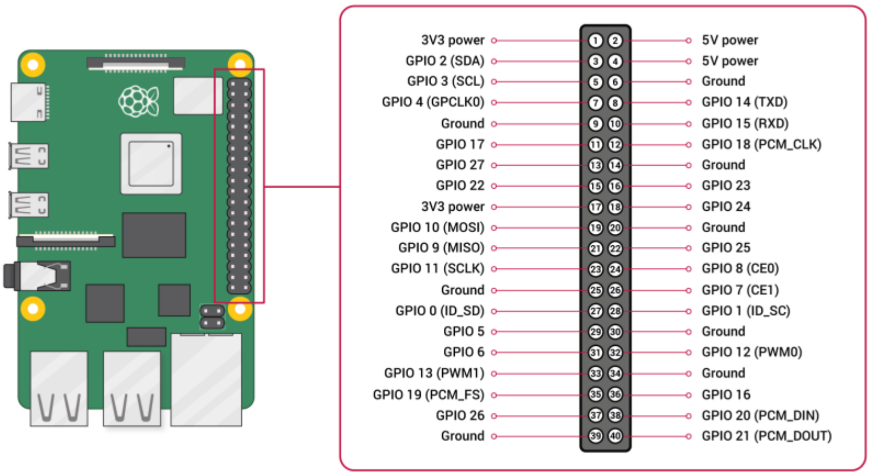


Figure 65: System of wiring for Raspberry Pi 3 model B

Therefore, it is two-step stage by wiring the digital converter ADS1115 to the computer in our case Raspberry Pi 3 model B, and then we wire the microphone to the ADC for further details look at (figure 65).

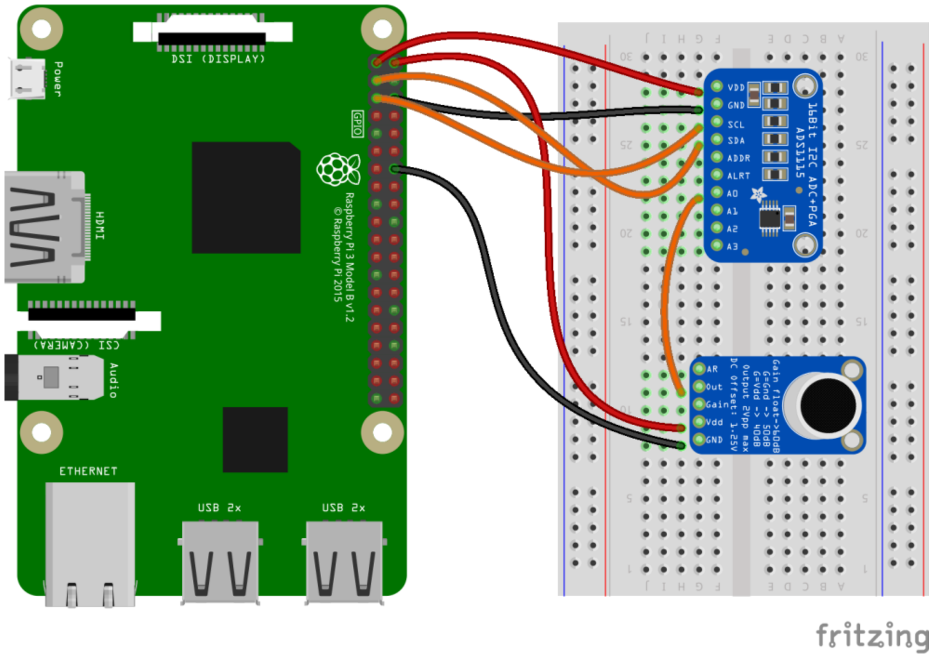


Figure 66: Wiring for the detection system

d. Raspberry Pi with ADC wiring

The wiring has four connections (Figure 66):

- PiPin1 (3V) to Pin1 of the ADS1115, we know every maximum voltage input power for an ADC must not exceed the VDD=3V value.
- GPIO2 ,Pipin3 (SDA) to pin4 SDA.
- GPIO3,Pipin5 (SCL) to pin3 SCL.
- PiPin6 (GND) to pin2 GND.

e. Wiring of the MAX9814 microphone with ADC

The wiring has one connection (Figure 66):

- Pin4 (out of MIC) to Pin7 (A0).

f. Wiring of MAX9814 with the Raspberry Pi

The wiring has two connections (Figure 66):

- PiPin14 (GND) to Pin1 (GND).
- PiPin2 (5V) to pin2 (VDD). It can be powered by 3V (Pin1).

g. **Connecting raspberry pi with microphone**

Since this hardware are not very optimal with low sensitivity for sound we are using different components : microphone and sound card connected with raspberry pi via USB port :

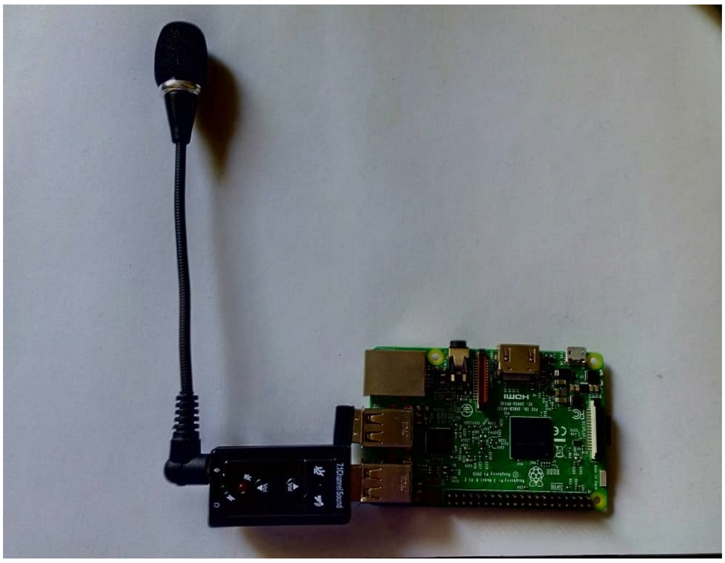


Figure 67: Final detection system

We must configure the USB entry from menu in audio settings as connected with the sound detector and it will be ready for operating.

We can record sound now using any application we have in our case we used node-red model and audacity to record some clips.

Check page annex for more technical information's about microphone type and sound card.

2.2.3. System configuration in Raspberry Pi:

In this part, we need to set the required configuration for the components with the raspberry pi to be able to read it with no issue starting by the configuration of the analogue digital converter.

The usage of the ADS1115 require installation of specific Python library ADAFRUIT_BLINKA, the adafruit_ads1x15 and the adafruit_bus_device, they need to be updated to the latest version adapted with the hardware, and we need to make sure the ADC is adapted to python3, so we must enable all the other statement running on the platform, at the same time make proper GPIO configuration.

3. Detection over real-time audio samples

3.1. System building

System is built using node-red software with collection of several nodes connected on local host in raspberry pi using same network for security measures ,the integrated user interface allow for user to control the device threw button “on” or “off” and same time showing sound level.in next title more detailed will be explained.

3.1.1. Node-red:

This phase we are using programming tool “NODE-RED” for wiring hardware and system API’s, it let us provide everything in one platform. To do so we need to install required module and nodes from node-red palette manager, and create a flow contains the needed hardware.

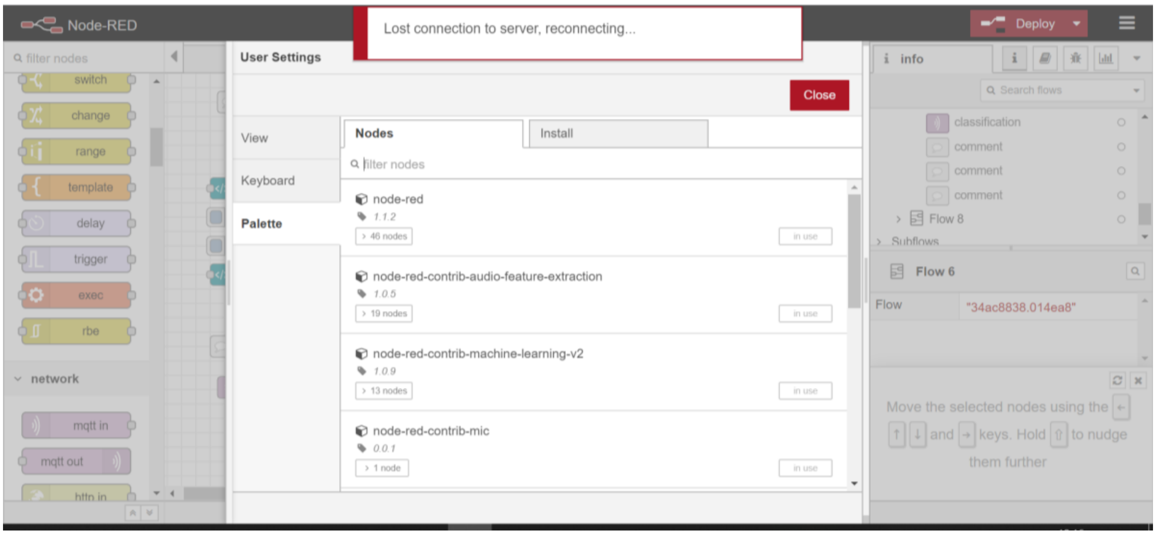


Figure 68: Palatte manager and nodes list in node red

Installed modules:

- **node-red-contrib-micropi 1.0.5** (module with node collection let us stream and record audio from microphone connected to raspberry pi only).
- **node-red-contrib-machine-learning-v2 1.0.9** (contains a set of nodes which offer machine learning functionalities. Such nodes have a python core that takes advantage of common ML libraries such as SciKit-Learn and Tensorflow. Classification and outlier detection can be performed through this package).
- **node-red-dashboard 2.23.2** (module that contain a set of nodes let us create a live data streamer on displaying dashboard).

General system require using 3 modules and 17 nodes: divide in two part,First part with detection system and second part for prediction.

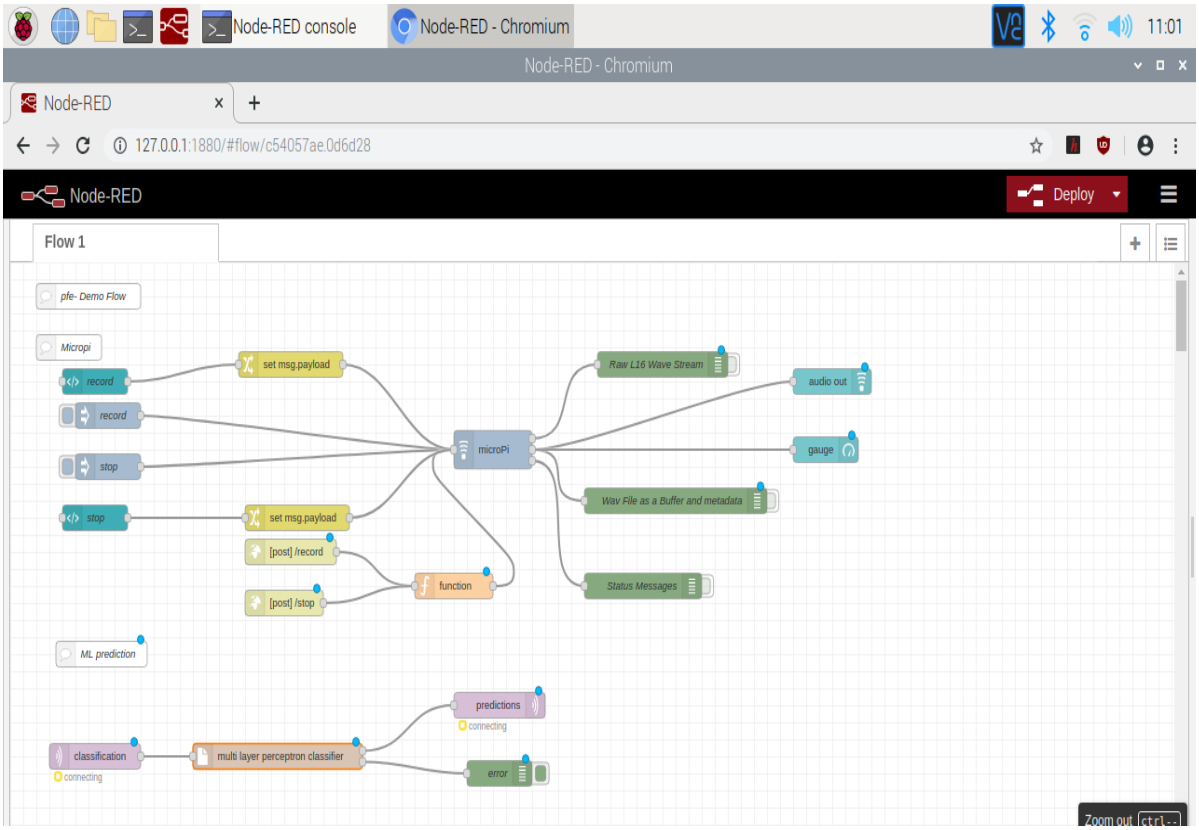


Figure 69: General system

h. Micropi first part

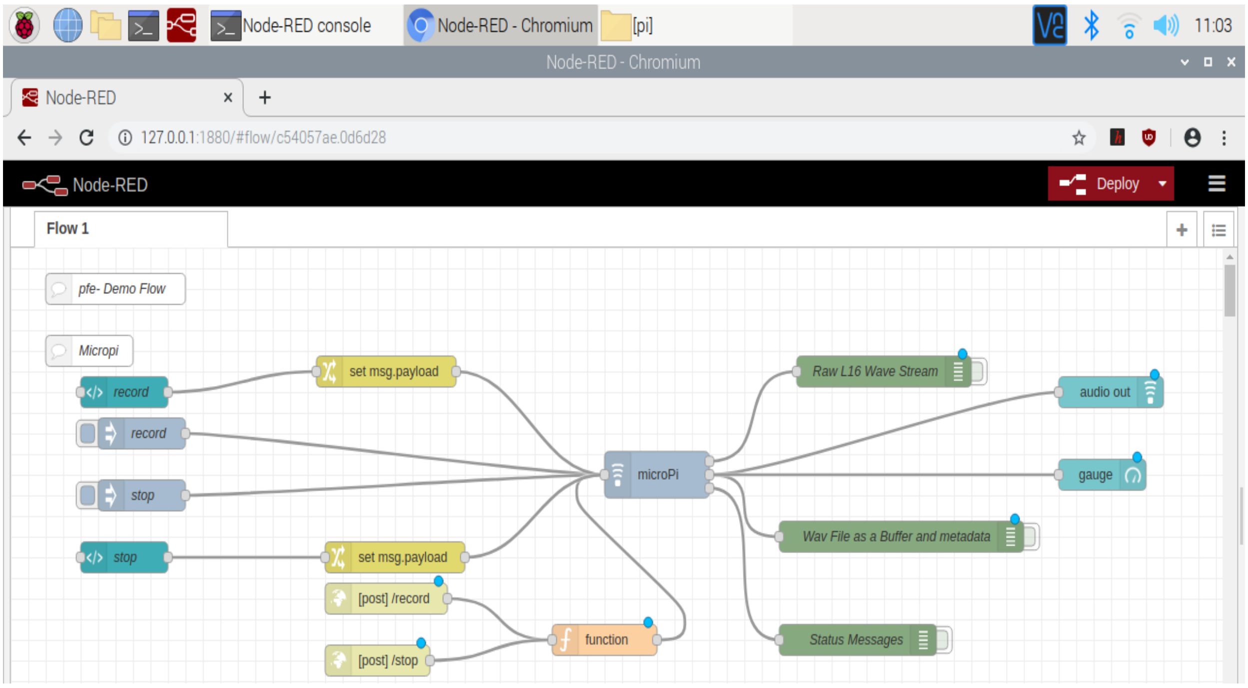


Figure 70: System detection with Micropi

Contain the detection sound system with some dashboard nodes let us control the record from the system nodes or from dashboard button explained as following:

- Micropi

for recording send boolean “true” as “msg.payload”, for “stoping” send “false”.we obtain 3 Outputs :

1. Streaming audio as raw L16
2. Wav file as a buffer with metadata
3. Status messages(print)

It is important to connect the second output with dashboard nodes else it won’t send any display . the micropi node is configured with path file to save it on sdcard locally(figure 71).

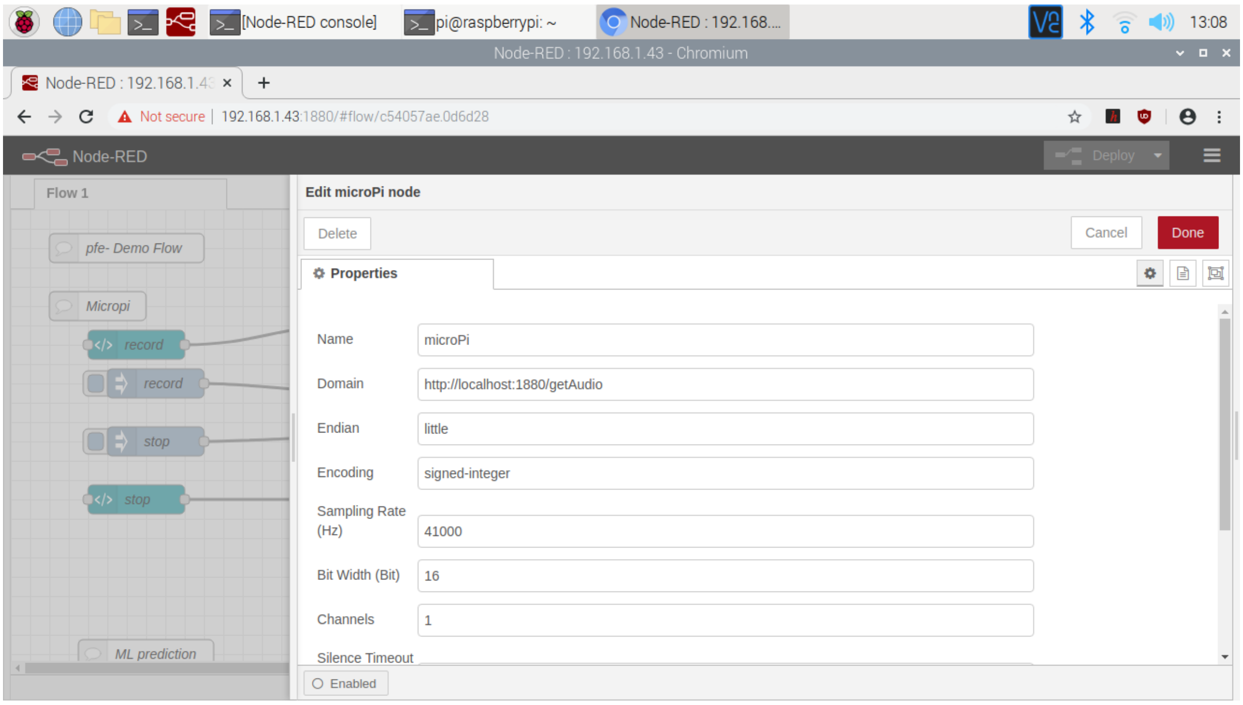


Figure 71: Micropi configuration

So, basically this step lets us record the sound on file [demo.wav] that is doesn’t save previous values but add on new values every time and surmount the old ones.

i. Prediction part

This part contain machine learning node that receive from MQTT node the predicted value and processed it threw “Multy layer perceptron model” and send back the results. Currently results available only at debug print.

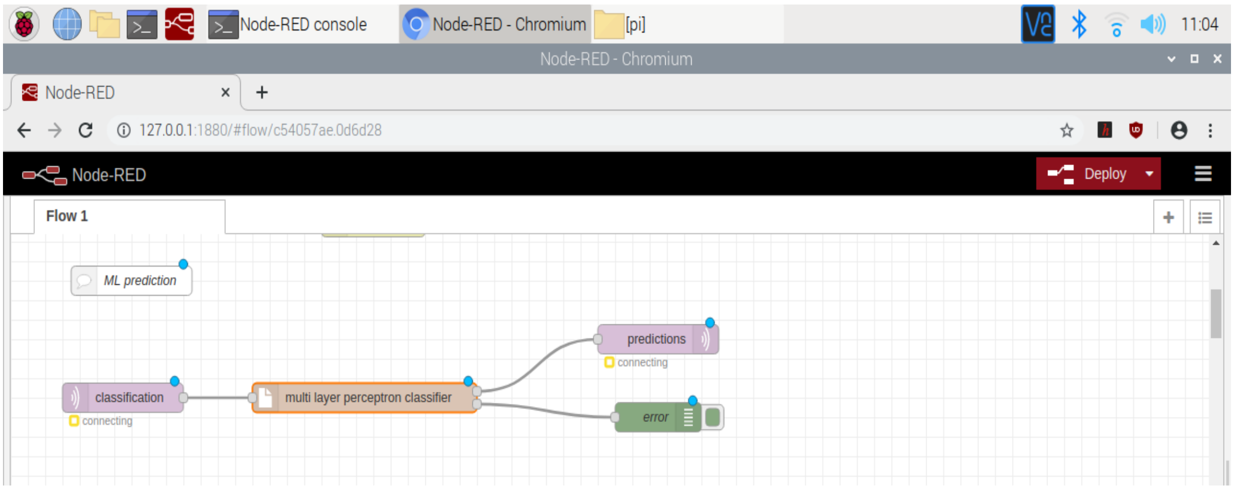


Figure 72: MLP model prediction node-red

j. **User interface :**

This step is for general user to use and how we want to display results built using html.

Showing three examples of sound state first one when device is turned off 'stop stat" (figure 73)and second one with 60 db sound level detecting the environment sounds (figure 74) and the last one shows the red color with 120 db explains the intensity of high sound level (figure 75,76).

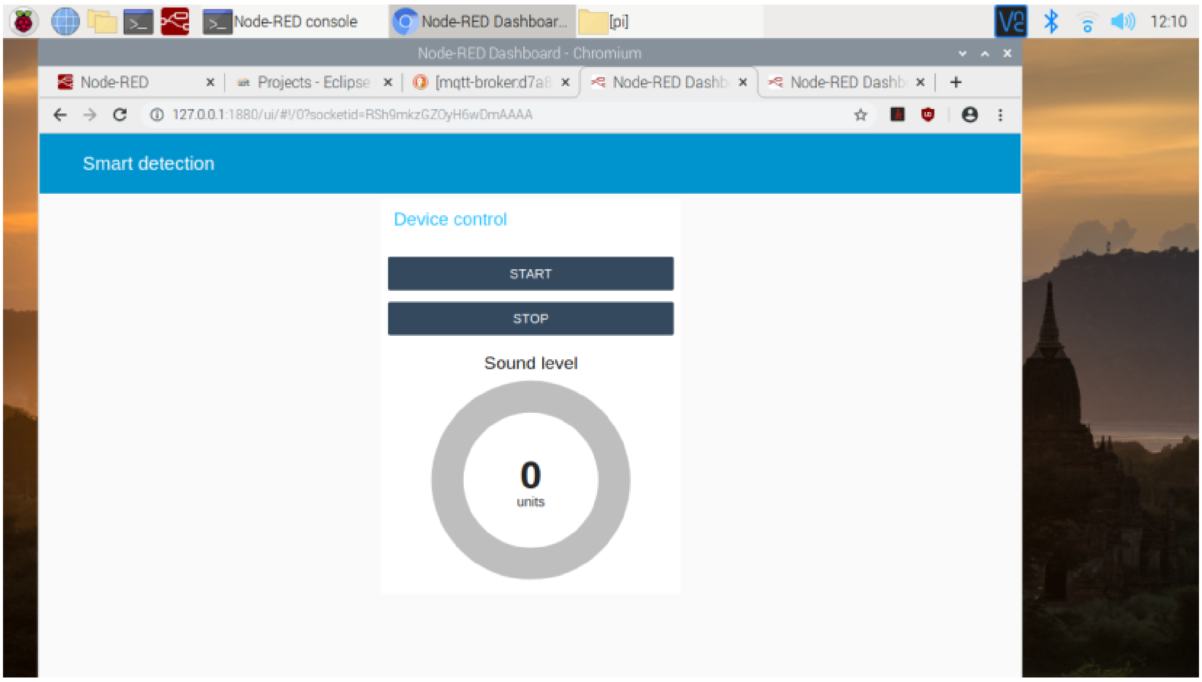


Figure 73: UI in stop mode

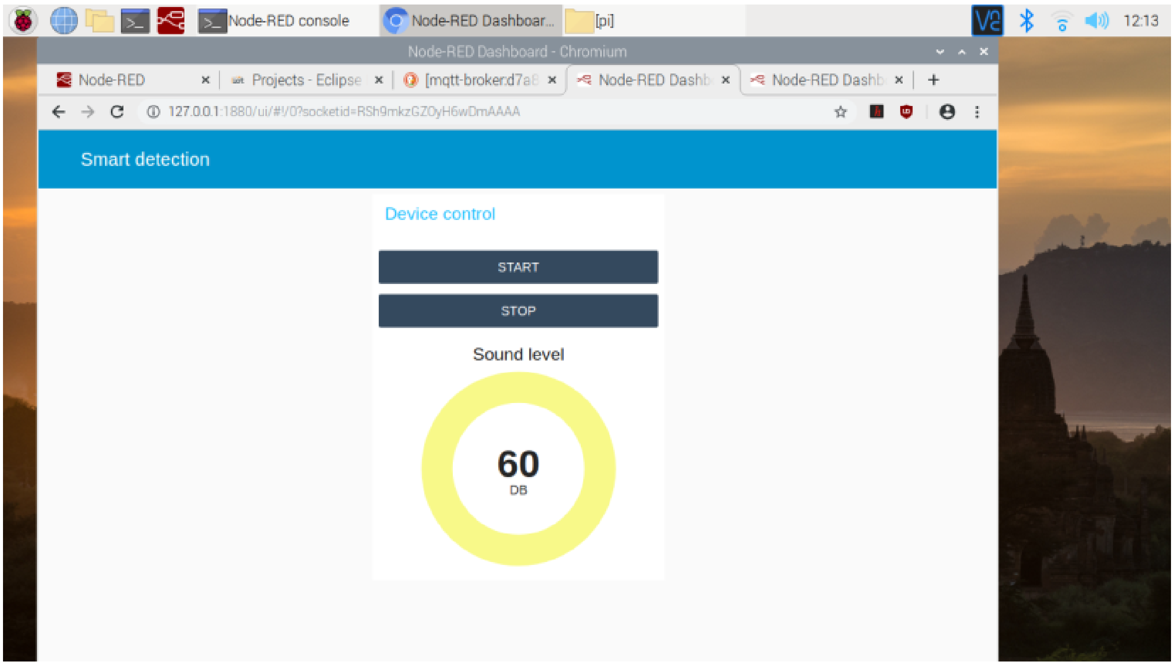


Figure 74: UI in enviroment sound

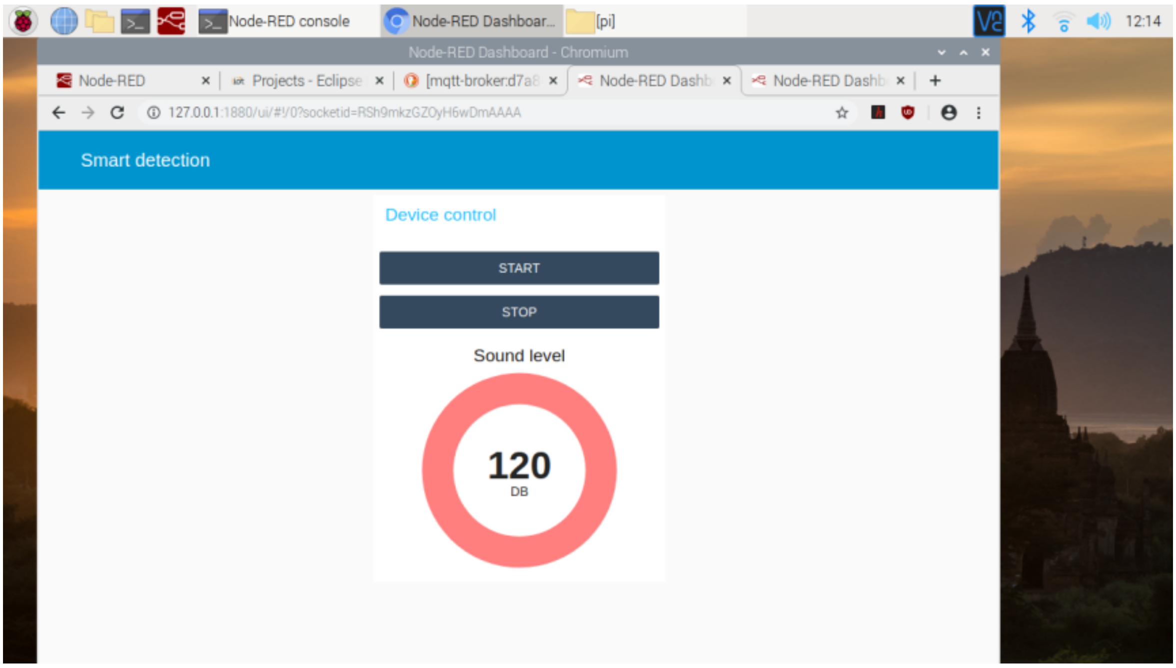


Figure 75: UI in high sound mode

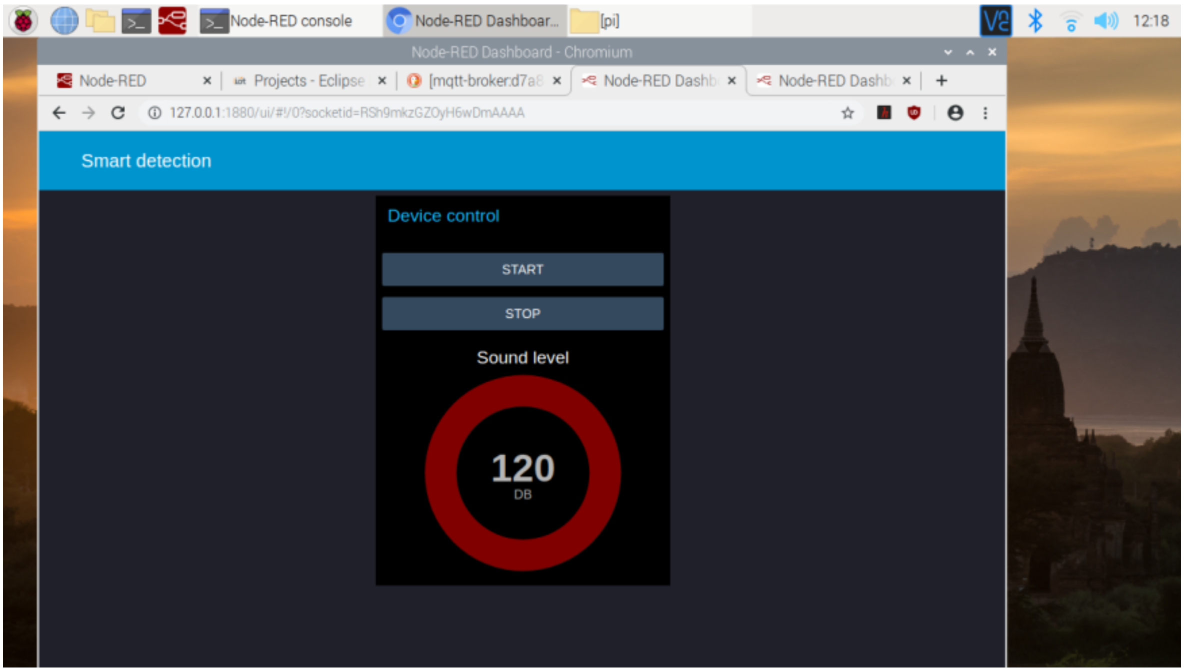


Figure 76: UI in dark mode for light sensitive users

That was the final step for this project the user interface is prototype alpha and in the near future we will be seeing more preformat UI/UX, and even better performance for detection, at the end there is general conclusion.

General conclusion

In this project, we realize a devise or a prototype based on the Raspberry Pi to detect baby cries and predict the cause of it using machine learning. Passing by the realization of the detection system (microphone, raspberry pi, etc.…), reorganize the database, test different technics of signal processing to have the best technic for our application.

Then we pass by a classification phase using Neural Networks (NN), Convolutional Neural Networks (CNN) and Support Vector Machine (SVM) and we chose the Neural Networks (NN) technic associate with MFCC method. The implantation of the deep learning to make the right prediction of the cry…

Despite all the difficulties we meet with during our work, we tried to combine the three specialties of biomedical engineering (biomedical electronics, biomedical informatics and telemedicine) in our project in order to obtain the best results.

The information is currently displayed via VNC viewer in phone, we could not finish the part of the transmission of the information or the results to the parents or the babysitter using one of the different techniques (Wi-Fi, Bluetooth or GSM)instead of that we just live show cast the detection results.

We are a little disappointed because of the lack of time and due to unforeseen events on top of it the Covid19 pandemic, that we could not finish our project with the way we wanted.

This work can always be developed; on the side of the device itself (the hardware), the collection of the database and on the software side (classification methods and the implantation of the machine learning), without forgetting the communication part for the improvement of the monitoring of the baby cries.

In addition, the experience gained and the results obtained during this study can be used for the future development of the device.

Finally, we remain very optimistic about obtaining in the near future a device that meets the expectations of people and help the mothers with hearing difficulties to monitor and know the needs of the baby.

This document is written with intention of being used by future researchers seeking information’s related to this field or any mentioned subjects.

Bibliography

[1] V. K. Mittal, « Discriminating features of infant cry acoustic signal for automated detection of cause of crying », in *2016 10th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, Tianjin, China, oct. 2016, p. 1-5, doi: 10.1109/ISCSLP.2016.7918391.

[2] S. Sharma, S. Asthana, et V. K. Mittal, « A Database of Infant Cry Sounds to Study the Likely Cause of Cry », p. 6.

[3] « Deafness ». https://www.who.int/news-room/facts-in-pictures/detail/deafness (consulté le mars 03, 2020).

[4] Douglas-Cowie, E., Cowie, R., & Schroder, M. (2000) A new emotion database: considerations, sources, and scope, Proc.ISC.

[5] W. A. Jabbar, H. K. Shang, S. N. I. S. Hamid, A. A. Almohammedi, R. M. Ramli, et M. A. H. Ali, « IoT-BBMS: Internet of Things-Based Baby Monitoring System for Smart Cradle », *IEEE Access*, vol. 7, p. 93791-93805, 2019, doi: 10.1109/ACCESS.2019.2928481.

[6] A. F. Symon, N. Hassan, H. Rashid, I. U. Ahmed, et S. M. Taslim Reza, « Design and development of a smart baby monitoring system based on Raspberry Pi and Pi camera », in *2017 4th International Conference on Advances in Electrical Engineering (ICAEE)*, Dhaka, sept. 2017, p. 117-122, doi: 10.1109/ICAEE.2017.8255338.

[7] H. I. Shahadi, D. H. Muhsen, H. T. Haider, et A. H. Taherinia, « Design and Implementation of a Smart Baby Crib », *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 671, p. 012050, janv. 2020, doi: 10.1088/1757-899X/671/1/012050.

[8] « Kshitij Lohekar, Sanket Deshmukh, Sahi Ambekar, Nayna Gole, L. Vina. Smart Baby Cradle. International Journal of Researc ». .

[9] L. Liu, Y. Li, et K. Kuo, « Infant cry signal detection, pattern extraction and recognition », in *2018 International Conference on Information and Computer Technologies (ICICT)*, DeKalb, IL, mars 2018, p. 159-163, doi: 10.1109/INFOCT.2018.8356861.

[10] R. I. Tuduce, M. S. Rusu, H. Cucu, et C. Burileanu, « Automated Baby Cry Classification on a Hospital-acquired Baby Cry Database », in *2019 42nd International Conference on Telecommunications and Signal Processing (TSP)*, Budapest, Hungary, juill. 2019, p. 343-346, doi: 10.1109/TSP.2019.8769075.

[11] F. Eyben, F. Weninger, F. Gross, et B. Schuller, « Recent developments in openSMILE, the munich open-source multimedia feature extractor », in *Proceedings of the 21st ACM international conference on Multimedia - MM '13*, Barcelona, Spain, 2013, p. 835-838, doi: 10.1145/2502081.2502224.

[12] B. Schuller, et al.. "Deception, Sincerity & Native Language",Interspeech ComParE, pp.2001-2005, 2016.

[13] Eibe Frank, Mark A. Hall, and Ian H. Witten. The WEKA workbench "Data Mining: Practical Machine Learning Tools and Techn .

[14] M. A. Hall, « Feature Selection for Discrete and Numeric Class Machine Learning », p. 16.

[15] Y. Lavner, R. Cohen, D. Ruinskiy, et H. IJzerman, « Baby Cry Detection in Domestic Environment using Deep Learning », p. 5, 2016.

[16] J. Hillenbrand, « The Physics of Sound », p. 44.

[17] « Sound », *Wikipedia*. févr. 17, 2020, Consulté le: févr. 18, 2020. [En ligne]. Disponible sur: https://en.wikipedia.org/w/index.php?title=Sound&oldid=941170005.

[18] Fundamentals of Telephone Communication Systems. Western Electrical Company. 1969. p. 2.1.

[19] « sound | Properties, Types, & Facts | Britannica ». https://www.britannica.com/science/sound-physics (consulté le févr. 21, 2020).

[20] « What are the characteristics of Sound Waves? », *Jagranjosh.com*, mai 07, 2018. https://www.jagranjosh.com/general-knowledge/what-are-the-characteristics-of-sound-waves-1525678871-1 (consulté le févr. 22, 2020).

[21] I.-A. Banica, H. Cucu, A. Buzo, D. Burileanu, et C. Burileanu, « Baby cry recognition in real-world conditions », in *2016 39th International Conference on Telecommunications and Signal Processing (TSP)*, Vienna, Austria, juin 2016, p. 315-318, doi: 10.1109/TSP.2016.7760887.

[22] S. Asthana, N. Varma, et V. K. Mittal, « Preliminary analysis of causes of infant cry », in *2014 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, Noida, India, déc. 2014, p. 000468-000473, doi: 10.1109/ISSPIT.2014.7300634.

[23] « human ear | Structure, Function, & Parts | Britannica ». https://www.britannica.com/science/ear (consulté le févr. 25, 2020).

[24] « Understanding how the ear works », *Hearing Link*. https://www.hearinglink.org/your-hearing/about-hearing/how-the-ear-works/ (consulté le févr. 22, 2020).

[25] « Deafness | Britannica ». https://www.britannica.com/science/deafness (consulté le mars 03, 2020).

[26] « Deaf People Parenting Hearing Children », *Verywell Health*. https://www.verywellhealth.com/deaf-parents-with-hearing-children-1046779 (consulté le mars 03, 2020).

[27] « WFD | World Federation of the Deaf ». https://wfdeaf.org/ (consulté le févr. 22, 2020).

[28] « Fédération Nationale des Sourds d'Algérie (FNSA) ». http://fnsa.e-monsite.com/ (consulté le févr. 25, 2020).

[29] « Algeria - AASL ». http://aasl.aacore.jp/wiki/Algeria#Organizations_and_associations_of_the_Deaf.2Fdeaf (consulté le févr. 25, 2020).

[30] « What Is A Condenser Microphone? (Detailed Answer + Examples) | My New Microphone ». https://mynewmicrophone.com/what-is-a-condenser-microphone-detailed-answer-examples/ (consulté le mars 09, 2020).

[31] « What is a filter in electronics? - Quora ». https://www.quora.com/What-is-a-filter-in-electronics (consulté le mars 09, 2020).

[32] « Active Low Pass Filter - Op-amp Low Pass Filter ». https://www.electronics-tutorials.ws/filter/filter_5.html (consulté le mars 09, 2020).

[33] « Scientific computing tools for Python — SciPy.org ». https://www.scipy.org/about.html (consulté le mars 08, 2020).

[34] B. McFee *et al.*, « librosa: Audio and Music Signal Analysis in Python », Austin, Texas, 2015, p. 18-24, doi: 10.25080/Majora-7b98e3ed-003.

[35] S. A. Alim et N. K. A. Rashid, « Some Commonly Used Speech Feature Extraction Algorithms », *Nat. Artif. Intell. - Algorithms Appl.*, déc. 2018, doi: 10.5772/intechopen.80419.

[36] « Continuous wavelet transform - Wikipedia ». https://en.wikipedia.org/wiki/Continuous_wavelet_transform (consulté le mars 08, 2020).

[37] S. P. Dewi, A. L. Prasasti, et B. Irawan, « The Study of Baby Crying Analysis Using MFCC and LFCC in Different Classification Methods », in *2019 IEEE International Conference on Signals and Systems (ICSigSys)*, Bandung, Indonesia, juill. 2019, p. 18-23, doi: 10.1109/ICSIGSYS.2019.8811070.

[38] H. Subramanian, "Audio Signal Classification," M. Tech CreditSemin. Rep., Pp. 1–17, 2004.

[39] G. Jin, B. Ye, Y. Wu, And F. Qu, "Vehicle Classification BasedOn Seismic Signatures Using Convolutional Neural Network," .

[40] S. Sharma, P. R. Myakala, R. Nalumachu, S. V. Gangashetty, andV. K. Mittal, "Acoustic analysis of infant cry signal towa .

[41] T. Khatib, A. Mohamed, K. Sopian, et M. Mahmoud, « Modeling of Solar Energy for Malaysia Using Artificial Neural Networks », p. 6.

[42] 2017 at 9:00am Posted by Amelia Matteson on August 11 et V. Blog, « Neural Networks: Crash Course On Multi-Layer Perceptron ». https://www.datasciencecentral.com/profiles/blogs/crash-course-on-multi-layer-perceptron-neural-networks-1 (consulté le août 16, 2020).

[43] « Deep Learning With Python: A Guide | Built In ». https://builtin.com/data-science/deep-learning-python (consulté le août 19, 2020).

[44] A. Hijazi et J.-M. Flaus, « A Deep Learning Approach for Intrusion Detection System in Industry Network », p. 8.

[45] V. J, « Tutorial on using Keras flow_from_directory and generators », *Medium*, déc. 02, 2019. https://medium.com/@vijayabhaskar96/tutorial-image-classification-with-keras-flow-from-directory-and-generators-95f75ebe5720 (consulté le mars 08, 2020).

[46] Taddy, Matt (2019). "Stochastic Gradient Descent". Business Data Science: Combining Machine Learning and Economics to O.

[47] « Machine à vecteurs de support », *Wikipédia*. juill. 08, 2020, Consulté le: juill. 10, 2020. [En ligne]. Disponible sur: https://fr.wikipedia.org/w/index.php?title=Machine_%C3%A0_vecteurs_de_support&oldid=172729764.

[48] « 1.4. Support Vector Machines — scikit-learn 0.23.1 documentation ». https://scikit-learn.org/stable/modules/svm.html (consulté le juill. 10, 2020).

[49] « Learning path: Getting started with Watson Studio – Build Smart. Build Secure. IBM Developer ». https://developer.ibm.com/technologies/data-science/series/learning-path-watson-studio/ (consulté le juill. 10, 2020).

[50] « sklearn.model_selection.train_test_split — scikit-learn 0.23.1 documentation ». https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html?highlight=test%20train#sklearn.model_selection.train_test_split (consulté le juill. 10, 2020).

[51] « sklearn.decomposition.PCA — scikit-learn 0.23.1 documentation ». https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html?highlight=pca#sklearn.decomposition.PCA (consulté le juill. 10, 2020).

Annex

Making an SD Card – Using Windows

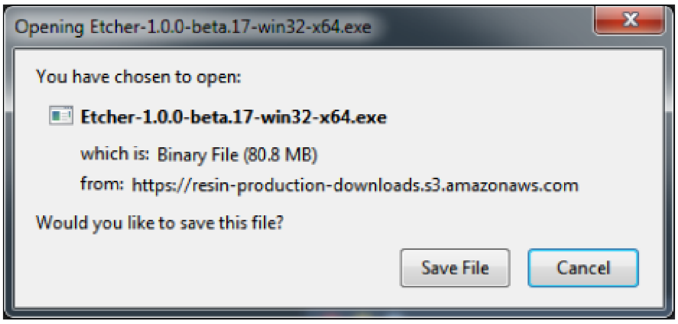
We really like using balenaEtcher for burning SD cards. Works great on any version of Windows, macOS and Linux. It will not over-write your backup disk drive, and can handle compressed images so you do not need to unzip them!

Step 1.

Download Etcher from: https://www.balena.io/etcher/ (https://adafru.it/EMc)

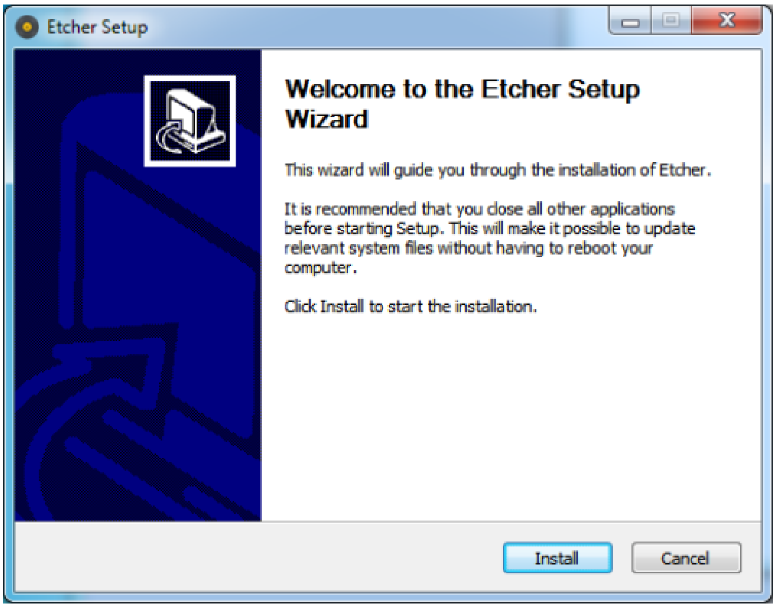
https://adafru.it/EMc

https://adafru.it/EMc



Step 2.

Run the downloaded app to install!



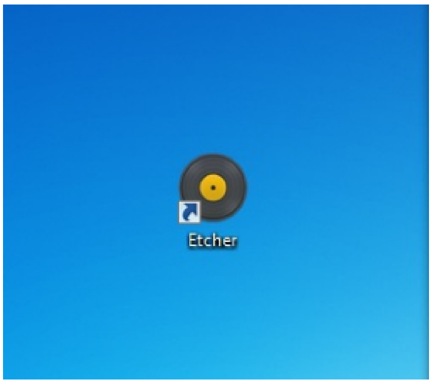
You can start immediately, doubleclick the Etcher desktop icon, or select it from the Start menu

Step 3.

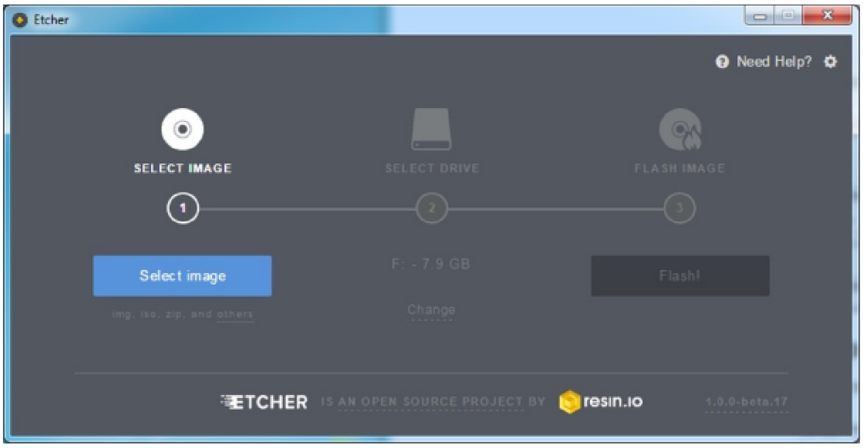
Eject any external storage devices such as USB flash drives and backup hard disks. This makes it easier to identify the SD card. Then insert the SD card into the slot on your computer or into the reader.

Step 4.

Run the Etcher program

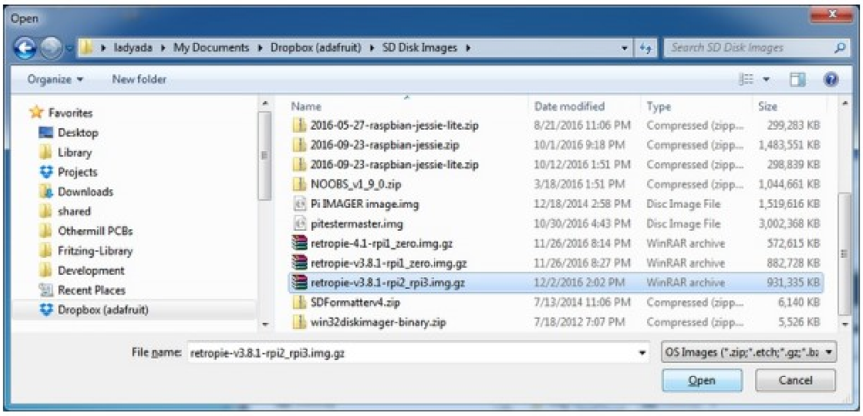


This will launch the following application.



Step 5.

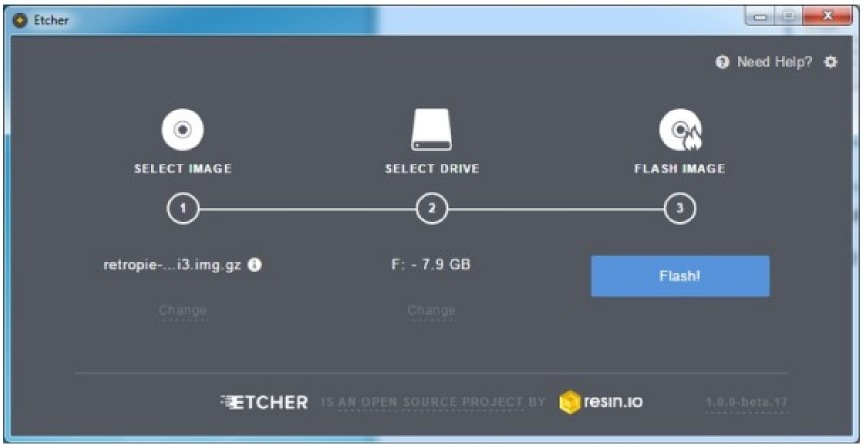
Select the image file by clicking **Select Image** you can select a compressed file such as a **.zip** or **.gz**



Step 6.

Etcher will automatically try to detect the SD drive, check the size to make sure its the right one

Then click **Flash!**



Check that you have the right device, as it will be reformatted, and then click Install.

It will take a few minutes to install, but once the SD card is ready, you will see the following.

Test & Configure

If you plan to use multiple SD cards, it is not a bad idea to label the card, or for microSD cards label the little plastic case they usually come in.

Testing the card is easy - insert it into your Pi, then connect a keyboard to the USB port and a NTSC/PAL TV to the composite port or an HDMI monitor to the HDMI Port. Then power it by connecting a Micro USB cable to the Pi and powering it via a computer or a USB wall charger.

For Raspbian, you should see something like the following, an Adafruit/Raspberry logo in the top left, and a ton of text filling up the screen:

In the next tutorial, you will find out how to configure your Raspberry Pi the first time you boot it up.

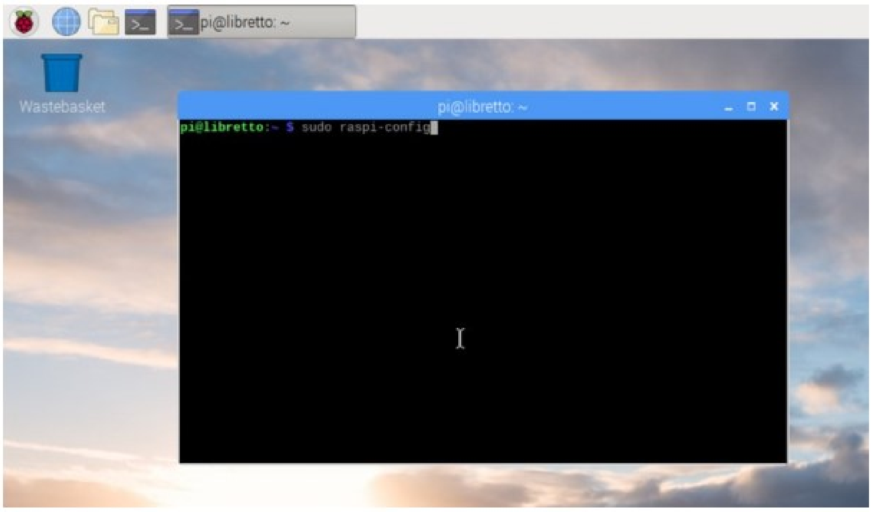
Setting up Wifi

raspi-config

raspi-config is the Raspberry Pi configuration tool that makes network configuration, remote access (eg. ssh / vnc), location settings and boot options a snap. This tool can be run from the command line and provides a simple interface that is easy to navigate with arrow keys. We highly recommend using this method over the GUI or manual command line file editing due to the flexibility of being able to run it remotely or in a console.

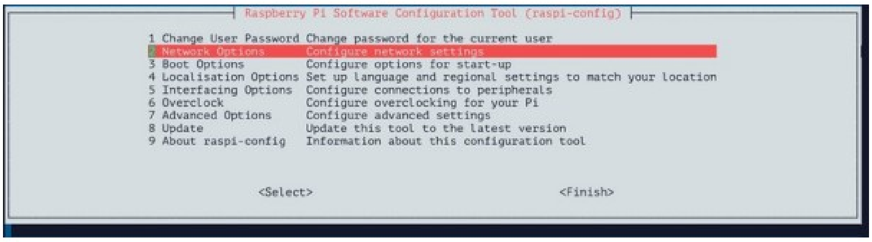
Launching raspi-config

When you first boot your newly configured Pi it will either drop you into a console or into a graphical environment. In either case we will run the following command. The GUI boot up will require us to launch a terminal.

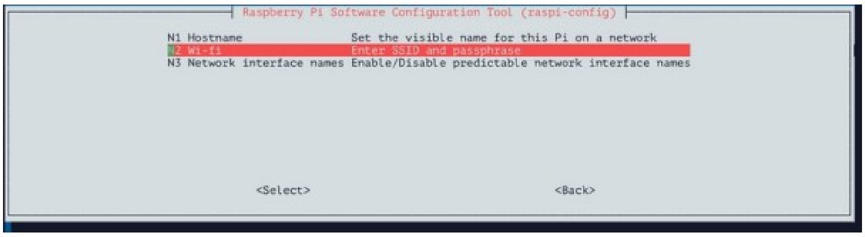



```
sudo raspi-config
```

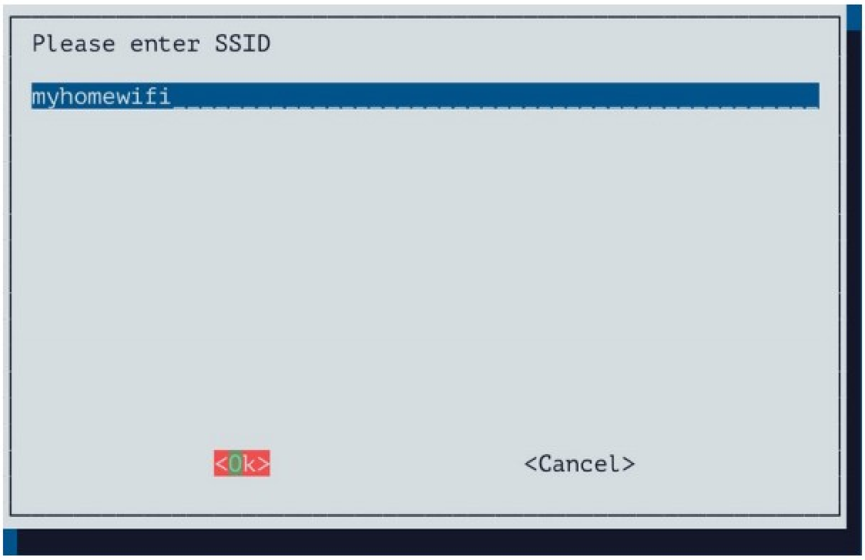
Select Network Options



Select Wi-Fi

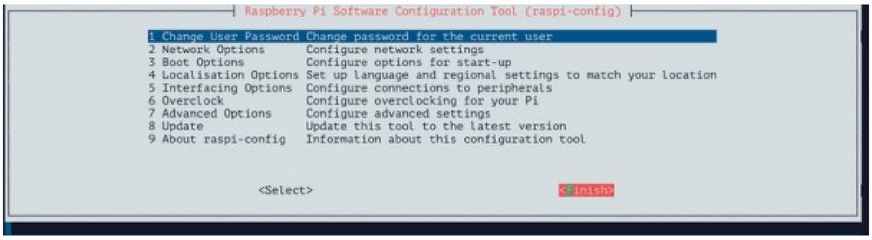


Enter WiFi Name (SSID)



Enter WiFi Password

Select Finish

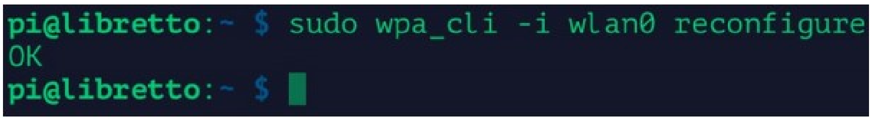


Bring Up the WiFi Adapter

We could just reboot and see if the WiFi automatically comes up. The following wpa_cli command will bring up the WiFi interface on-line without requiring a reboot.

If you have a modern Raspberry Pi and are using an external adapter it is likely that your WiFi device will be named wlan1.

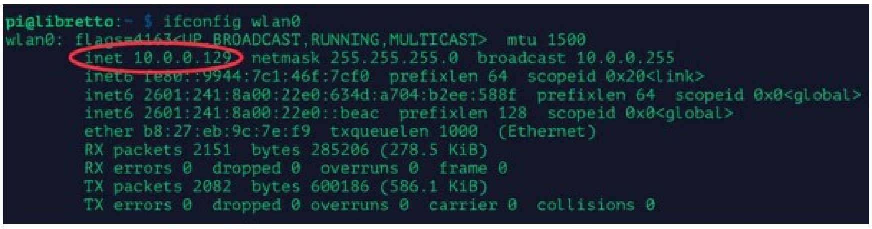
```
sudo wpa_cli -i wlan0 reconfigure
```



Verify IP Address

If everything worked properly we will see the IP address of our device after the "inet " argument. We are now on-line. If this does not work try repeating the steps above in raspi-config and entering the WiFi access point name (SSID) and password again.

```
ifconfig wlan0
```



Installing VNC

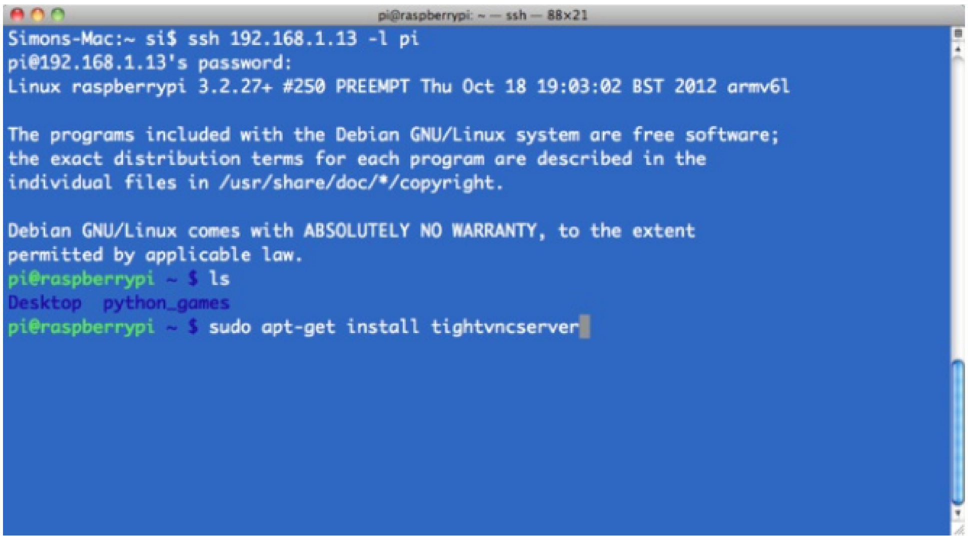
SSH (see http://learn.adafruit.com/adafruits-raspberry-pi-lesson-6-using-ssh/overview (https://adafru.it/aU3)) is often all you need to control your Raspberry Pi, however sometimes it is useful to be able to remote control your Raspberry Pi using the mouse and seeing just what you would see on the screen of the Raspberry Pi.

VNC (Virtual Network Connection) is a standard for doing just this. To use it, you have to install some software on your Pi. There are a number of VNC server applications, and the one we are going to use is called “tightvnc”.

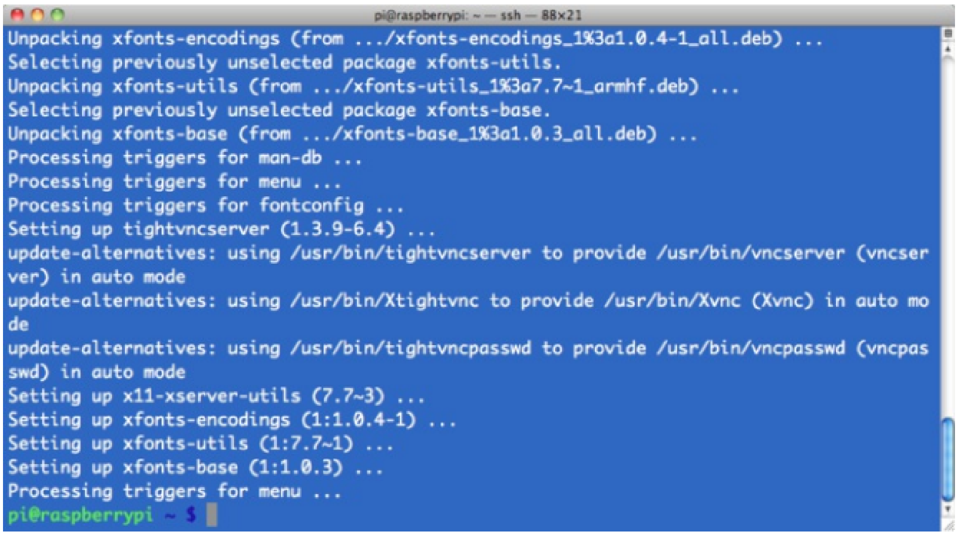
We can install the VNC server software using the SSH connection that we established earlier.

Enter the following command into your SSH terminal:

```
sudo apt-get update
sudo apt-get install tightvncserver
```

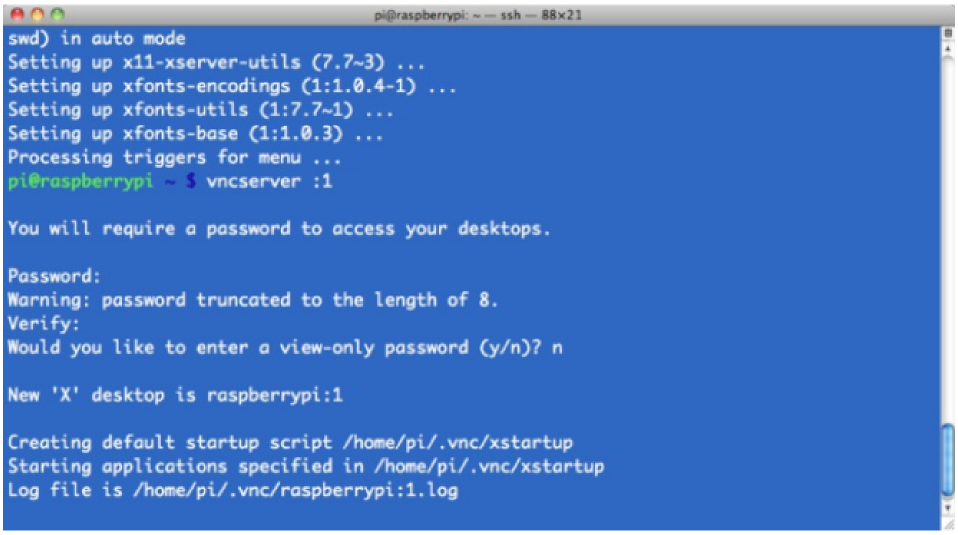


You will be prompted to confirm installation by typing “Y' and finally when installation is complete, you should see the following:



We now need to run the VNC Server, so enter the following command into your SSH window:

```
vncserver :1
```



You will be prompted to enter and confirm a password. It would make sense to use “raspberry” for this, but passwords are limited to 8 characters, so I use “raspberr”. Note that this is the password that you will need to use to connect to the Raspberry Pi remotely.

You will also be asked if you want to create a separate “read-only” password – say no.

From now on, the only command that you need to type within your SSH to start the VNC server will be:

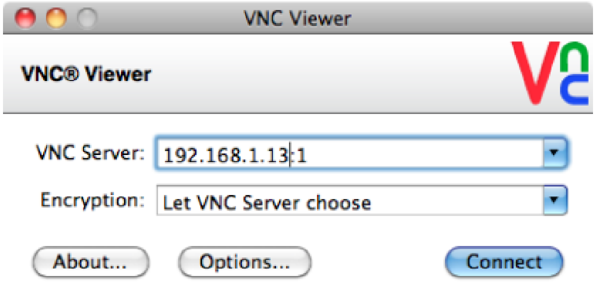
```
vncserver :1
```

The VNC server is now running and so we can attempt to connect to it, but first we must switch to the computer from which we want to control the Pi and setup a VNC client to connect to the Pi.

Using a VNC Client

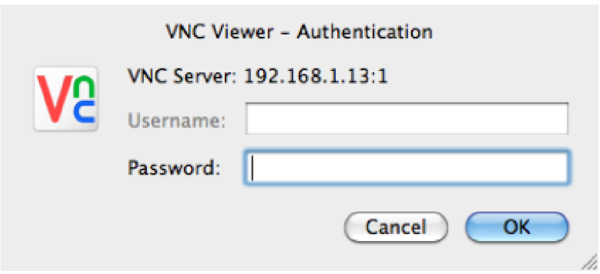
Again, there are many VNC clients, of which “VNCViewer” (http://www.realvnc.com (https://adafru.it/aU4)) is available for most platforms and I have found it to work well with TightVNC.

When you first run VNCViewer, you will see the following:



Enter the IP address of your Raspberry Pi, **append :1** (to indicate the port) and click on “Connect”. You will then get a warning message. Just click 'Continue'.

The following window will then popup for you to enter your password (“raspberr”).



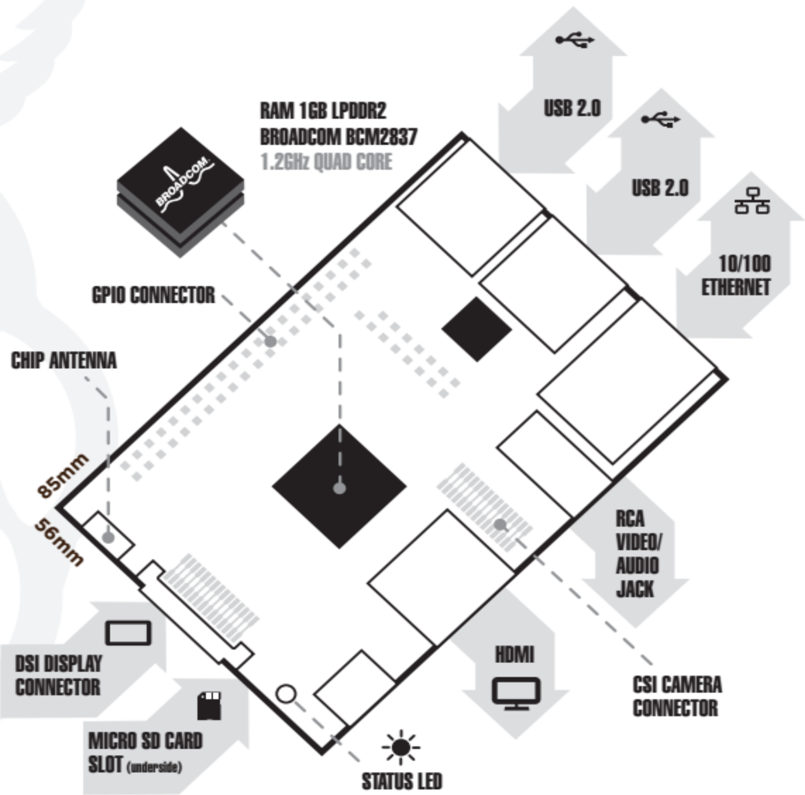
Finally, the VNC window itself should appear. You will be able to use the mouse and do everything as if you were using the Pi's keyboard mouse and monitor, except through your other computer.

As with SSH, since this is working over your network, your Pi could be situated anywhere, as long as it is connected to your network.

Raspberry Pi

Raspberry Pi 3 Model B

Product Name	**Raspberry Pi 3**
Product Description	The Raspberry Pi 3 Model B is the third generation Raspberry Pi. This powerful credit-card sized single board computer can be used for many applications and supersedes the original Raspberry Pi Model B+ and Raspberry Pi 2 Model B. Whilst maintaining the popular board format the Raspberry Pi 3 Model B brings you a more powerful processer, 10x faster than the first generation Raspberry Pi. Additionally it adds wireless LAN & Bluetooth connectivity making it the ideal solution for powerful connected designs.
RS Part Number	**896-8660**



Raspberry Pi

Raspberry Pi 3 Model B

Specifications	
Processor	Broadcom BCM2387 chipset. 1.2GHz Quad-Core ARM Cortex-A53 802.11 b/g/n Wireless LAN and Bluetooth 4.1 (Bluetooth Classic and LE)
GPU	Dual Core VideoCore IV® Multimedia Co-Processor. Provides Open GL ES 2.0, hardware-accelerated OpenVG, and 1080p30 H.264 high-profile decode. Capable of 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs with texture filtering and DMA infrastructure
Memory	1GB LPDDR2
Operating System	Boots from Micro SD card, running a version of the Linux operating system or Windows 10 IoT
Dimensions	85 x 56 x 17mm
Power	Micro USB socket 5V1, 2.5A
Connectors:	
Ethernet	10/100 BaseT Ethernet socket
Video Output	HDMI (rev 1.3 & 1.4 Composite RCA (PAL and NTSC)
Audio Output	Audio Output 3.5mm jack, HDMI USB 4 x USB 2.0 Connector
GPIO Connector	40-pin 2.54 mm (100 mil) expansion header: 2x20 strip Providing 27 GPIO pins as well as +3.3 V, +5 V and GND supply lines
Camera Connector	15-pin MIPI Camera Serial Interface (CSI-2)
Display Connector	Display Serial Interface (DSI) 15 way flat flex cable connector with two data lanes and a clock lane
Memory Card Slot	Push/pull Micro SDIO

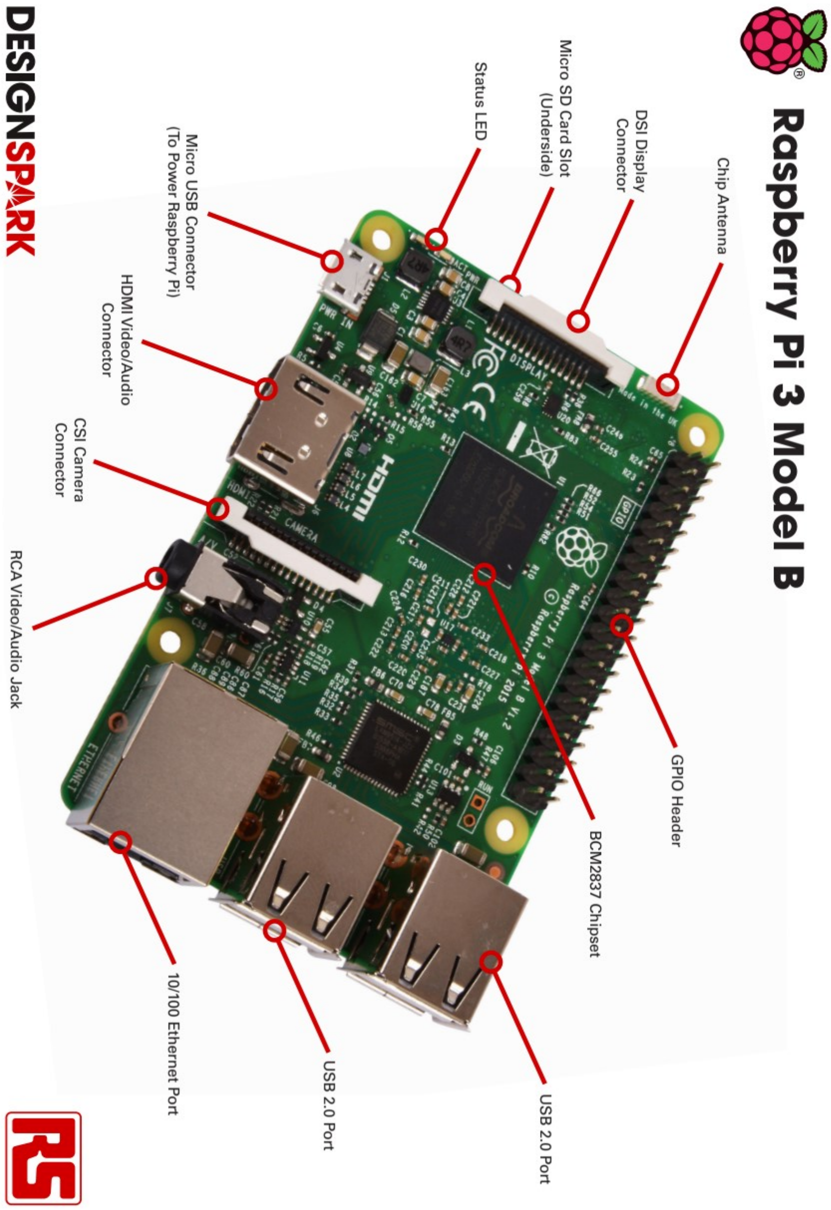
Key Benefits

- Low cost
- 10x faster processing
- Consistent board format
- Added connectivity

Key Applications

- Low cost PC/tablet/laptop
- Media centre
- Industrial/Home automation
- Print server
- Web camera
- Wireless access point
- Environmental sensing/monitoring (e.g. weather station)
- IoT applications
- Robotics
- Server/cloud server
- Security monitoring
- Gaming

www.rs-components.com/raspberrypi


Raspberry Pi 3 Model B
Chip Antenna
DSI Display Connector
Micro SD Card Slot (Underside)
Status LED
Micro USB Connector (To Power Raspberry Pi)
HDMI Video/Audio Connector
CSI Camera Connector
RCA Video/Audio Jack
10/100 Ethernet Port
USB 2.0 Port
USB 2.0 Port
BCM2837 Chipset
GPIO Header
DESIGNSPARK

Raspberry Pi®
Frequently Asked Questions

What is a Raspberry Pi?

Created by the Raspberry Pi Foundation, the Raspberry Pi is an open-source, Linux based, credit card sized computer board. The Pi is an exciting and accessible means of improving computing and programming skills for people of all ages. By connecting to your TV or monitor and a keyboard, and with the right programming, the Pi can do many things that a desktop computer can do such as surf the internet and play video. The Pi is also great for those innovative projects that you want to try out - newer models are ideal for Internet of Things projects due to their processing power. With Pi 3, Wireless LAN and Bluetooth Low Energy are on-board too.

What are the differences between the models?

Current versions of the Raspberry Pi are the Pi A+, Pi B+, Pi 2 B, Pi 3 B and Compute Module.

	Pi A+	Pi B+	Pi 2 B	Pi 3 B	Compute Module
Dimensions	66 x 56 x 14mm	85 x 56 x 17mm	85 x 56 x 17mm	85 x 56 x 17mm	67.5 x 30mm
SoC	BCM2835	BCM2835	BCM2836	BCM2837	BCM2835
Processor Core	ARM11	ARM11	ARM Cortex-A7	ARM Cortex-A53	ARM11
Processing Power	700 MHz	700 MHz	900 MHz	1.2 GHz	700 MHz
Memory	256 MB	512 MB	1 GB	1GB LPDDR2	512 MB
Ports	1x USB 2.0	4x USB 2.0 1x 10/100 Ethernet	4x USB 2.0 1x 10/100 Ethernet	4x USB 2.0 1x 10/100 Ethernet	N/A
GPIO	40	40	40	40	N/A

What do I get with my Raspberry Pi?

A Raspberry Pi board only.

Each Raspberry Pi customer is unique. You may already have cables, power supplies, keyboards, SD memory cards or monitors. However, if you do require additional products to start with your Pi or to really get creative, we can help.

Our expanding range of accessories includes:

Protective Cases	Power Supplies	NOOBS microSD Cards	Keyboards & Mice	Printers
Cables	Displays & Camera Boards	Wireless Connectivity	Add-on Boards	RS Pi Bundles

Raspberry Pi®

T5875DV
Raspberry Pi Power Supply

Features:

- Official Raspberry Pi Power Supply
- 1.5M Micro USB B Lead
- ErP Level 6 Efficiency Rating
- 50,000 Hour MTBF
- 1 Years Warranty

Output

Output Voltage	+5.1Vdc
Minimum Load Current	0A
Nominal Load Current	2.5A
Nominal Output Power	13W
Output Regulation	+/-5%
Line Regulation	+/-2%
Ripple & Noise	120mVp-p Maximum
Rise Time	100mS Maximum at nominal input
Turn-on Delay	3 Seconds Maximum at nominal input
Protection	Short circuit, over current, over voltage
Efficiency	80.86%
Output Cable	1500mm Micro USB B 5 Pin

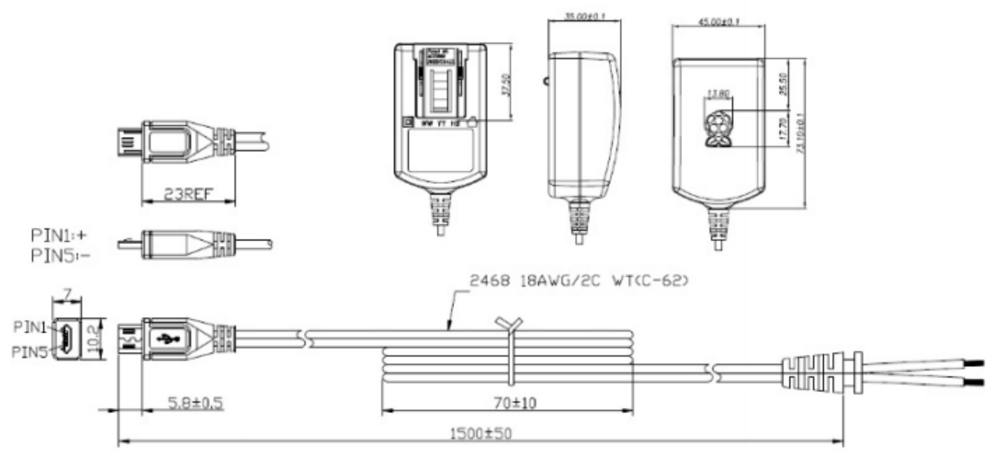
Input

Input Voltage Range	90-264VAC
Input Frequency	47-63Hz
Input Current	0.5A Max
Inrush Current	No damage and IP fuse will not blow
AC Inlet	UK, Euro, Aus & US changeable heads

Other

Dimensions	73.2 (L) * 45.1 (W) * 35.1 (H) mm
Weight	Approx 150g
Operating Temp.	0 °C to 40 °C
Storage Temp.	-20 °C to +60 °C
Opertating Humidity	20 ~ 85 % RH. Non-Condesning
MTBF	50,000 Hours

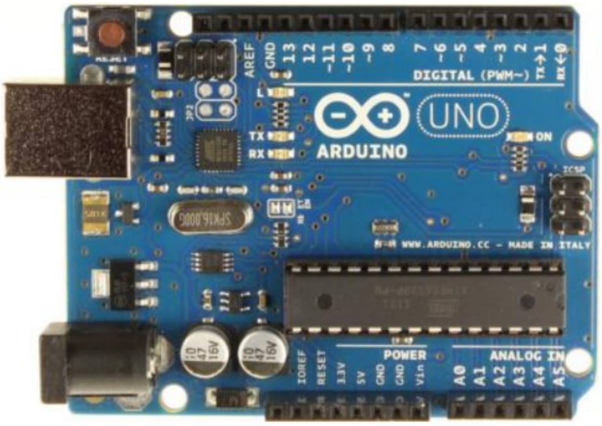
Diagrams



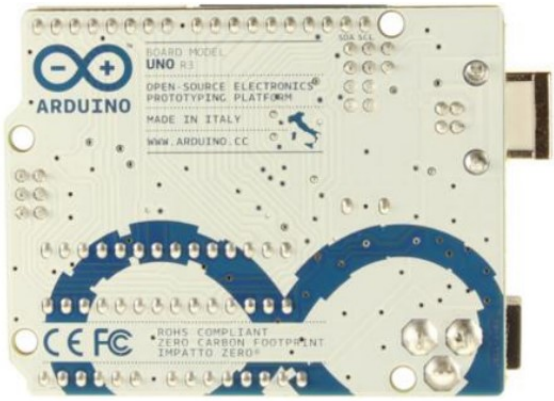
STONTRONICS
THE POWER TO SUPPLY
Chancerygate Business Centre, Cradock Road, Reading, Berkshire, RG2 0AH.
Tel: +44 (0) 118 931 1199 • Fax: +44 (0) 118 931 1145 • Email: info@stontronics.co.uk • Web: www.stontronics.co.uk

Please Note: Image shown is representative of entire range. Individual PSU image &/or drawings or data sheets available on request.

Arduino Uno

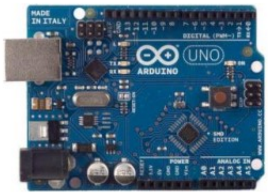


Arduino Uno R3 Front



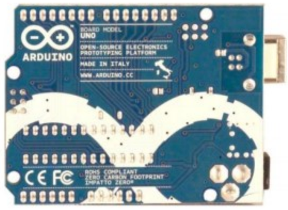
Arduino Uno R3 Back

Arduino Uno R2 Front



Arduino Uno SMD

Arduino Uno Front



Arduino Uno Back

Overview

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.
The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.
Revision 2 of the Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode.
Revision 3 of the board has the following new features:

- 1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the Arduino Due that operate with 3.3V. The second one is a not connected pin, that is reserved for future purposes.
- Stronger RESET circuit.
- Atmega 16U2 replace the 8U2.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the index of Arduino boards.

Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V

Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Schematic & Reference Design

EAGLE files: arduino-uno-Rev3-reference-design.zip (NOTE: works with Eagle 6.0 and newer)
Schematic: arduino-uno-Rev3-schematic.pdf
Note: The Arduino reference design can use an Atmega8, 168, or 328, Current models use an ATmega328, but an Atmega8 is shown in the schematic for reference. The pin configuration is identical on all three processors.

Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.
External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.
The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.
The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.**This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the analogWrite() function.

- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication using the SPI library.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function. Additionally, some pins have specialized functionality:

- **TWI: A4 or SDA pin and A5 or SCL pin.** Support TWI communication using the Wire library.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with analogReference().
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the mapping between Arduino pins and ATmega328 ports. The mapping for the Atmega8, 168, and 328 is identical.

Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).
A SoftwareSerial library allows for serial communication on any of the Uno's digital pins.
The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details. For SPI communication, use the SPI library.

Programming

The Arduino Uno can be programmed with the Arduino software (download). Select "Arduino Uno from the **Tools > Board** menu (according to the microcontroller on your board). For details, see the reference and tutorials.
The ATmega328 on the Arduino Uno comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).
You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.
The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available . The ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by:

- On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.
- On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See this user-contributed tutorial for more information.

Automatic (Software) Reset

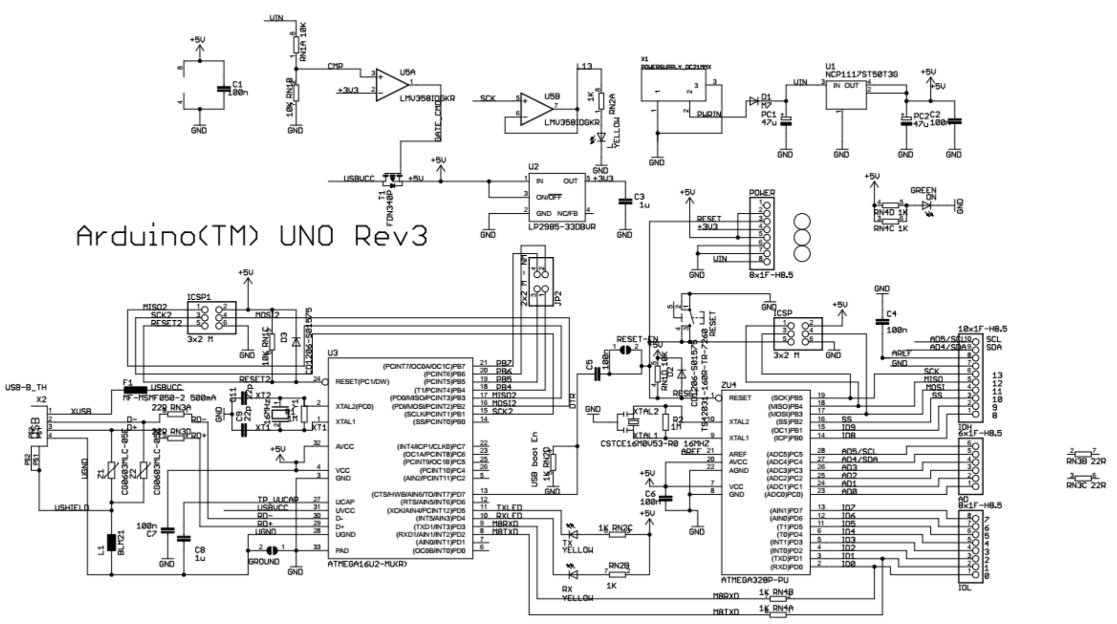
Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.
This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.
The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread for details.

USB Overcurrent Protection

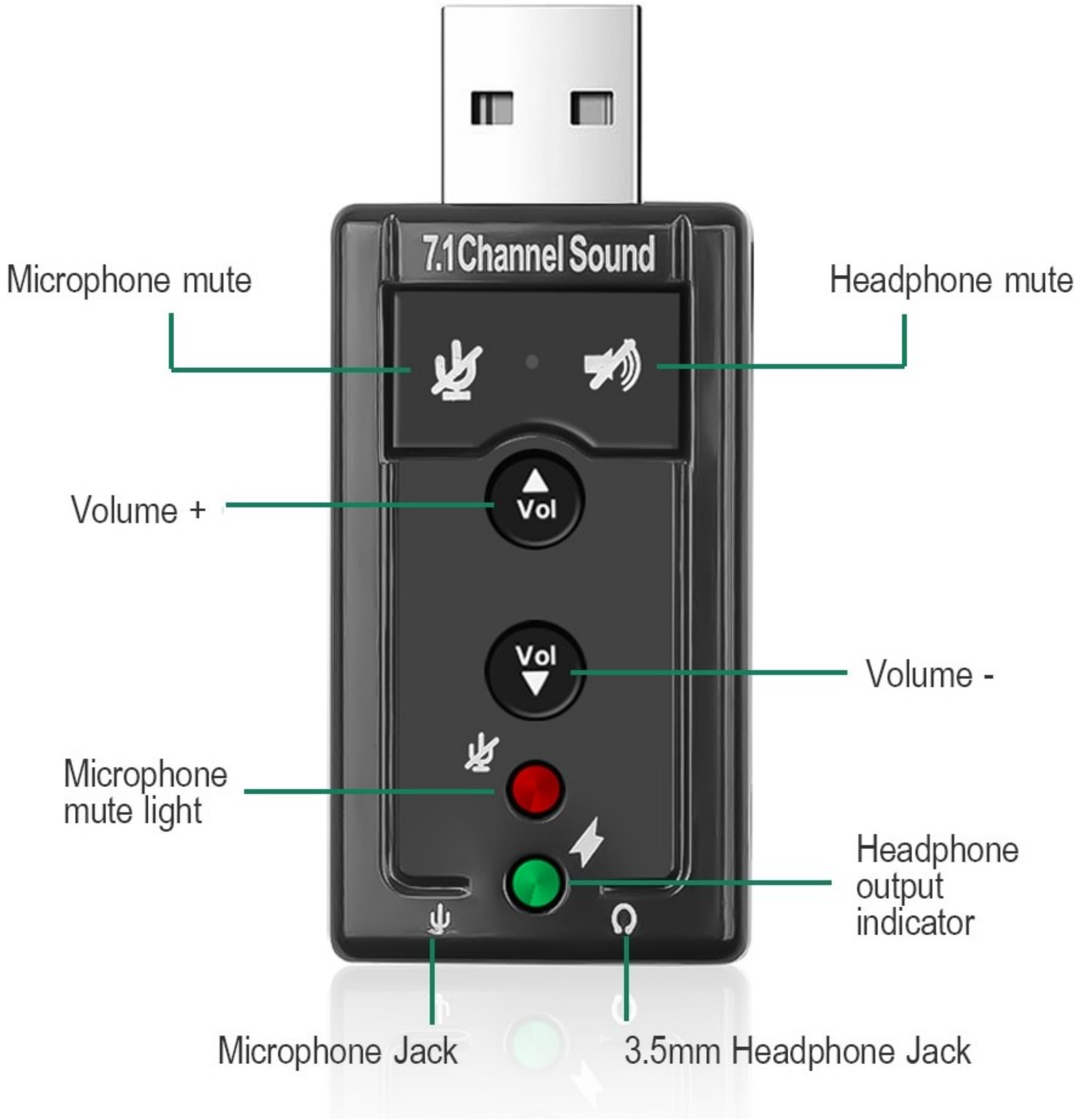
The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Four screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.



Reference Designs ARE PROVIDED "AS IS" AND "WITH ALL FAULTS. Arduino DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE
Arduino may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Arduino reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.
The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.
ARDUINO is a registered trademark.
Use of the ARDUINO name must be compliant with http://www.arduino.cc/en/Main/Policy



Features:

Size: approx 57 x 25 x 12mm
Weight: 12g
LED Indicators: Microphone - Mute Status, Activity
USB2.0 Full Speed (12Mbps) specification
USB HID 1.1 Class Specification
USB Audio Device Class Specification 1.0. USB powered bus, no external power required.
This is a USB 2.0 external 7.1 channel sound card sound adapter. It can work with either desktop or laptop computer. And with Xear's 3D sound simulation software, it extends the stereo speaker or earphone to the 7.1 channel environment, giving you high-quality sound enjoyment.
Connectors: USB Type A, Stereo Output Jack, Mono Microphone Input Jack
Xear 3D Virtual 7.1 Channel Sound Simulation Software is included for Windows XP / Vista
Keys functions: microphone-mute, speaker-mute, volume-up and down
Plug and play. No drivers required for Windows 2000 / XP / Server 2003 / Vista, Linux, Mac Os

EVALUATION KIT AVAILABLE

MAX9814

Microphone Amplifier with AGC and Low-Noise Microphone Bias

General Description

The MAX9814 is a low-cost, high-quality microphone amplifier with automatic gain control (AGC) and low-noise microphone bias. The device features a low-noise preamplifier, variable gain amplifier (VGA), output amplifier, microphone-bias-voltage generator and AGC control circuitry.

The low-noise preamplifier has a fixed 12dB gain, while the VGA gain automatically adjusts from 20dB to 0dB, depending on the output voltage and the AGC threshold. The output amplifier offers selectable gains of 8dB, 18dB, and 28dB. With no compression, the cascade of the amplifiers results in an overall gain of 40dB, 50dB, or 60dB. A trilevel digital input programs the output amplifier gain. An external resistive divider controls the AGC threshold and a single capacitor programs the attack/release times. A trilevel digital input programs the ratio of attack-to-release time. The hold time of the AGC is fixed at 30ms. The low-noise microphone-bias-voltage generator can bias most electret microphones.

The MAX9814 is available in the space-saving, 14-pin TDFN package. This device is specified over the -40°C to +85°C extended temperature range.

Applications

- Digital Still Cameras
- Digital Video Cameras
- PDAs
- Bluetooth Headsets
- Entertainment Systems (e.g., Karaoke)
- Two-Way Communicators
- High-Quality Portable Recorders
- IP Phones/Telephone Conferencing

Features

- ♦ Automatic Gain Control (AGC)
- ♦ Three Gain Settings (40dB, 50dB, 60dB)
- ♦ Programmable Attack Time
- ♦ Programmable Attack and Release Ratio
- ♦ 2.7V to 5.5V Supply Voltage Range
- ♦ Low Input-Referred Noise Density of 30nV/√Hz
- ♦ Low THD: 0.04% (typ)
- ♦ Low-Power Shutdown Mode
- ♦ Internal Low-Noise Microphone Bias, 2V
- ♦ Available in the Space-Saving, 14-Pin TDFN (3mm x 3mm) Package
- ♦ -40°C to +85°C Extended Temperature Range

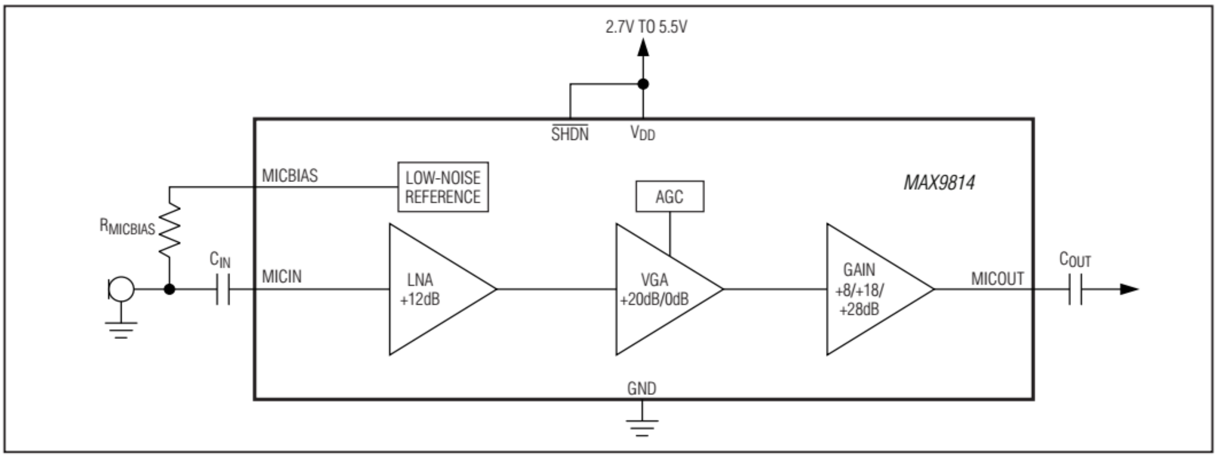
Ordering Information

PART	TEMP RANGE	PIN-PACKAGE
MAX9814ETD+T	-40°C to +85°C	14 TDFN-EP*

+Denotes a lead(Pb)-free/RoHS-compliant package.
T = Tape and reel.
**EP = Exposed pad.*

Pin Configurations appear at end of data sheet.

Simplified Block Diagram



For pricing, delivery, and ordering information, please contact Maxim Direct at 1-888-629-4642, or visit Maxim's website at www.maximintegrated.com.

19-0764; Rev 2; 6/09

Microphone Amplifier with AGC and Low-Noise Microphone Bias

ABSOLUTE MAXIMUM RATINGS

V_{DD} to GND ....-0.3V to +6V
All Other Pins to GND ....-0.3V to (V_{DD} + 0.3V)
Output Short-Circuit Duration ....Continuous
Continuous Current (MICOUT, MICBIAS) ....±100mA
All Other Pins ....±20mA
Continuous Power Dissipation (T_A = +70°C)
14-Pin TDFN-EP
(derate 16.7mW/°C above +70°C) 1481.5mW
Operating Temperature Range ....-40°C to +85°C
Junction Temperature ....+150°C
Lead Temperature (soldering, 10s) ....+300°C
Bump Temperature (soldering) Reflow ....+235°C

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ELECTRICAL CHARACTERISTICS

(V_{DD} = 3.3V, $\overline{SHDN}$ = V_{DD}, C_{CT} = 470nF, C_{CG} = 2µF, GAIN = V_{DD}, T_A = T_{MIN} to T_{MAX}, unless otherwise specified. Typical values are at T_A = +25°C.) (Note 1)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
GENERAL						
Operating Voltage	V_{DD}	Guaranteed by PSRR test	2.7		5.5	V
Supply Current	I_{DD}			3.1	6	mA
Shutdown Supply Current	$I_{\overline{SHDN}}$			0.01	1	µA
Input-Referred Noise Density	e_n	BW = 20kHz, all gain settings		30		nV/√Hz
Output Noise		BW = 20kHz		430		μV_{RMS}
Signal-to-Noise Ratio	SNR	BW = 22Hz to 22kHz (500mV_{RMS} output signal)		61		dB
		A-weighted		64		
Dynamic Range	DR	(Note 2)		60		dB
Total Harmonic Distortion Plus Noise	THD+N	f_{IN} = 1kHz, BW = 20Hz to 20kHz, R_L = 10kΩ, V_{TH} = 1V (threshold = $2V_{P-P}$), V_{IN} = 0.5mV_{RMS}, V_{CT} = 0V		0.04		%
		f_{IN} = 1kHz, BW = 20Hz to 20kHz, R_L = 10kΩ, V_{TH} = 0.1V (threshold = 200mV_{P-P}), V_{IN} = 30mV_{RMS}, V_{CT} = 2V		0.2		
Amplifier Input BIAS	V_{IN}		1.14	1.23	1.32	V
Maximum Input Voltage	V_{IN_MAX}	1% THD		100		mV_{P-P}
Input Impedance	Z_{IN}			100		kΩ
Maximum Gain	A	GAIN = V_{DD}	39.5	40	40.5	dB
		GAIN = GND	49.5	50	50.6	
		GAIN = unconnected	59.5	60	60.5	
Minimum Gain		GAIN = V_{DD}	18.7	20	20.5	dB
		GAIN = GND	29.0	30	30.8	
		GAIN = unconnected	38.7	40	40.5	
Maximum Output Level	V_{OUT_RMS}	1% THD+N, V_{TH} = MICBIAS		0.707		V_{RMS}
Regulated Output Level		AGC enabled, V_{TH} = 0.7V	1.26	1.40	1.54	V_{P-P}
AGC Attack Time	t_{ATTACK}	C_{CT} = 470nF (Note 3)		1.1		ms
Attack/Release Ratio	A/R	A/R = GND		1:500		ms/ms
		A/R = V_{DD}		1:2000		
		A/R = unconnected		1:4000		

ELECTRICAL CHARACTERISTICS (continued)

(V_{DD} = 3.3V, $\overline{SHDN}$ = V_{DD}, C_{CT} = 470nF, C_{CG} = 2μF, GAIN = V_{DD}, T_A = T_{MIN} to T_{MAX}, unless otherwise specified. Typical values are at T_A = +25°C.) (Note 1)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
MICOUT High Output Voltage	V_{OH}	I_{OUT} sourcing 1mA		2.45		V
MICOUT Low Output Voltage	V_{OL}	I_{OUT} sinking 1mA		3		mV
MICOUT Bias		MICOUT unconnected	1.14	1.23	1.32	V
Output Impedance	Z_{OUT}			50		Ω
Minimum Resistive Load	R_{LOAD_MIN}			5		kΩ
Maximum Capacitive Drive	C_{LOAD_MAX}			200		pF
Maximum Output Current	I_{OUT_MAX}	1% THD, R_L = 500Ω		1	2	mA
Output Short-Circuit Current	I_{SC}		3	8		mA
Power-Supply Rejection Ratio	PSRR	AGC mode; V_{DD} = 2.7V to 5.5V (Note 4)	35	50		dB
		f = 217Hz, V_{RIPPLE} = 100mV_{P-P} (Note 5)		55		
		f = 1kHz, V_{RIPPLE} = 100mV_{P-P} (Note 5)		52.5		
		f = 10kHz, V_{RIPPLE} = 100mV_{P-P} (Note 5)		43		
MICROPHONE BIAS						
Microphone Bias Voltage	$V_{MICBIAS}$	$I_{MICBIAS}$ = 0.5mA	1.84	2.0	2.18	V
Output Resistance	$R_{MICBIAS}$	$I_{MICBIAS}$ = 1mA		1		Ω
Output Noise Voltage	$V_{MICBIAS\_NOISE}$	$I_{MICBIAS}$ = 0.5mA, BW = 22Hz to 22kHz		5.5		μV_{RMS}
Power-Supply Rejection Ratio	PSRR	DC, V_{DD} = 2.7V to 5.5V	70	80		dB
		$I_{MICBIAS}$ = 0.5mA, V_{RIPPLE} = 100mV_{P-P}, f_{IN} = 1kHz		71		
TRILEVEL INPUTS (A/R, GAIN)						
Tri-Level Input Leakage Current		A/R or GAIN = V_{DD}	0.5V_{DD} / 180kΩ	0.5V_{DD} / 100kΩ	0.5V_{DD} / 50kΩ	mA
		A/R or GAIN = GND	0.5V_{DD} / 180kΩ	0.5V_{DD} / 100kΩ	0.5V_{DD} / 50kΩ	
Input High Voltage	V_{IH}		V_{DD} x 0.7			V
Input Low Voltage	V_{IL}				V_{DD} x 0.3	V
Shutdown Enable Time	t_{ON}			60		ms
Shutdown Disable Time	t_{OFF}			40		ms
DIGITAL INPUT ($\overline{SHDN}$)						
$\overline{SHDN}$ Input Leakage Current			-1		+1	μA
Input High Voltage	V_{IH}		1.3			V
Input Low Voltage	V_{IL}				0.5	V
AGC THRESHOLD INPUT (TH)						
TH Input Leakage Current			-1		+1	μA

Note 1: Devices are production tested at T_A = +25°C. Limits over temperature are guaranteed by design.
Note 2: Dynamic range is calculated using the EIAJ method. The input is applied at -60dBFS (0.707μV_{RMS}), f_{IN} = 1kHz.
Note 3: Attack time measured as time from AGC trigger to gain reaching 90% of its final value.
Note 4: CG is connected to an external DC voltage source, and adjusted until V_{MICOUT} = 1.23V.
Note 5: CG connected to GND with 2.2μF.

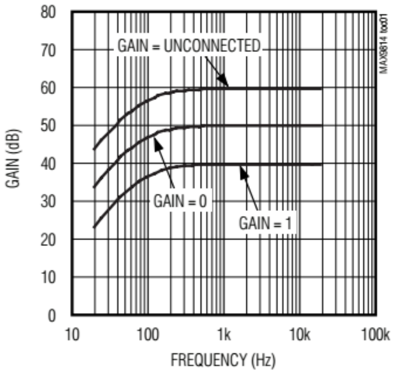
MAX9814

Microphone Amplifier with AGC and Low-Noise Microphone Bias

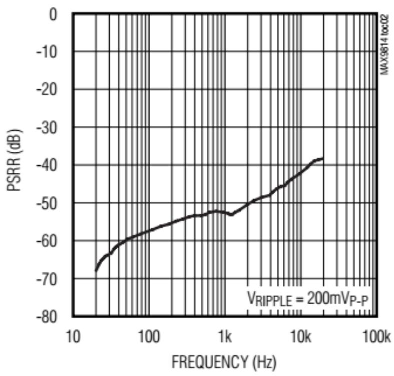
Typical Operating Characteristics

(V_{DD} = 5V, C_{CT} = 470nF, C_{CG} = 2.2µF, V_{TH} = $V_{MICBIAS}$ x 0.4, GAIN = V_{DD} (40dB), AGC disabled, no load, R_L = 10kΩ, C_{OUT} = 1µF, T_A = +25°C, unless otherwise noted.)

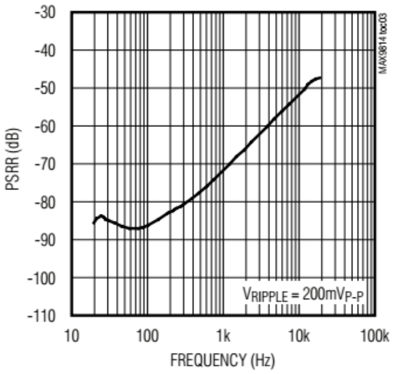
GAIN vs. FREQUENCY



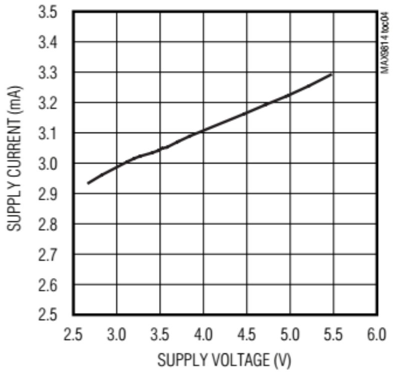
POWER-SUPPLY REJECTION RATIO vs. FREQUENCY



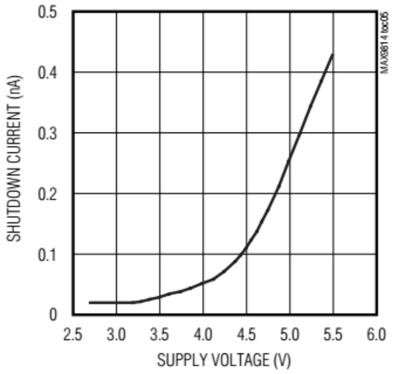
MICBIAS POWER-SUPPLY REJECTION RATIO vs. FREQUENCY



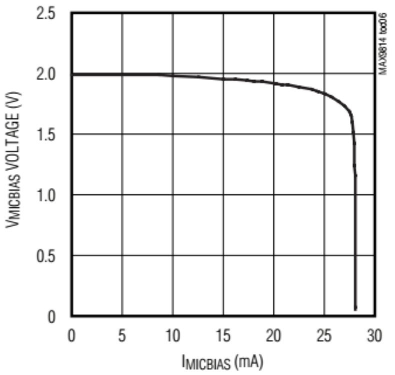
SUPPLY CURRENT vs. SUPPLY VOLTAGE



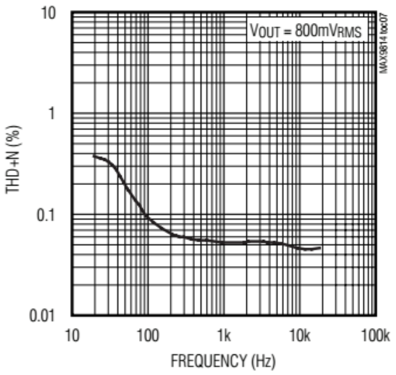
SHUTDOWN CURRENT vs. SUPPLY VOLTAGE



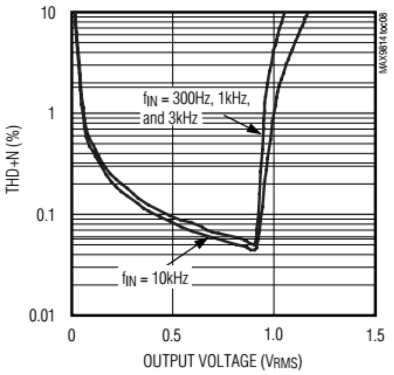
MICROPHONE BIAS VOLTAGE vs. MICROPHONE BIAS SOURCE CURRENT



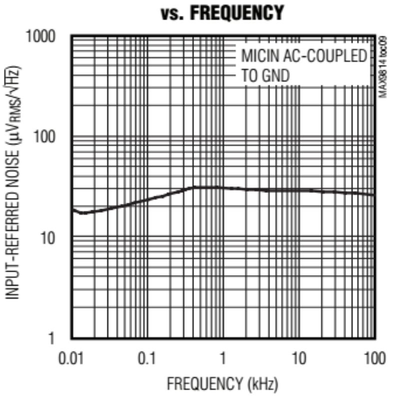
TOTAL HARMONIC DISTORTION PLUS NOISE vs. FREQUENCY



TOTAL HARMONIC DISTORTION PLUS NOISE vs. OUTPUT VOLTAGE



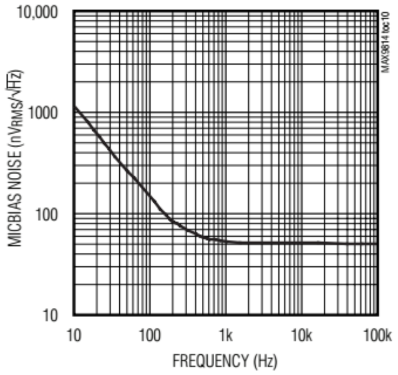
INPUT-REFERRED NOISE vs. FREQUENCY



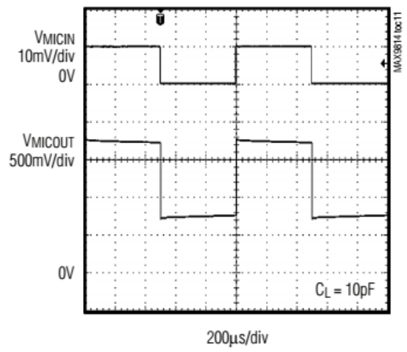
Typical Operating Characteristics (continued)

(V_{DD} = 5V, C_{CT} = 470nF, C_{CG} = 2.2µF, V_{TH} = $V_{MICBIAS}$ x 0.4, GAIN = V_{DD} (40dB), AGC disabled, no load, R_L = 10kΩ, C_{OUT} = 1µF, T_A = +25°C, unless otherwise noted.)

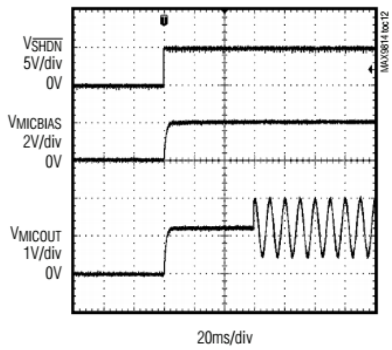
MICBIAS NOISE vs. FREQUENCY



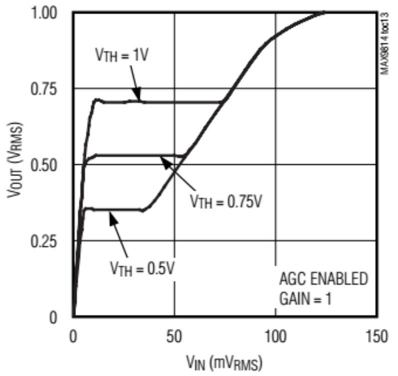
SMALL-SIGNAL PULSE RESPONSE



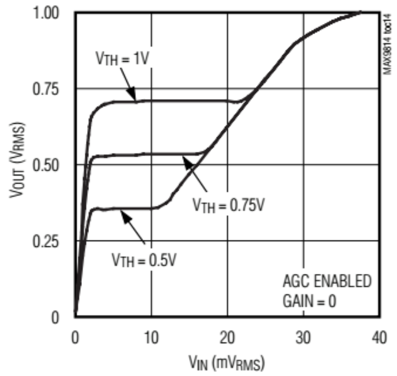
TURN-ON RESPONSE



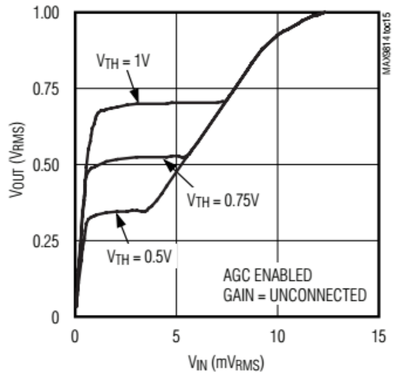
VOUT vs. VIN



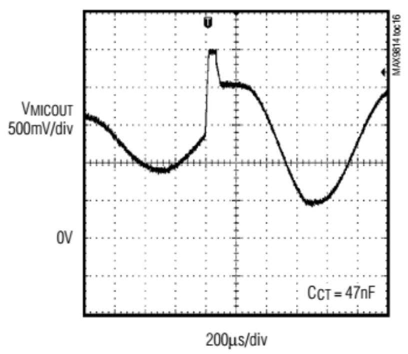
VOUT vs. VIN



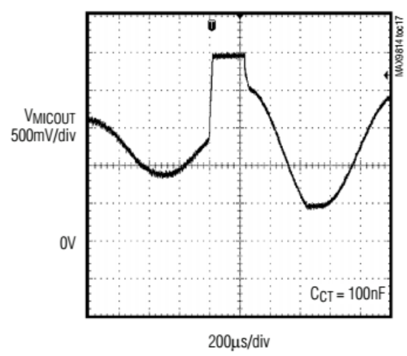
VOUT vs. VIN



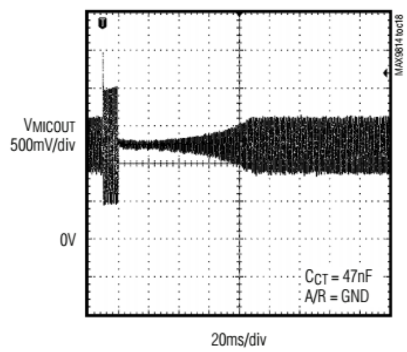
ATTACK TIME



ATTACK TIME



HOLD AND RELEASE TIME



Detailed Description

The MAX9814 is a low-cost, high-quality microphone amplifier with automatic gain control (AGC) and a low-noise microphone bias. The MAX9814 consists of several distinct circuits: a low-noise preamplifier, a variable gain amplifier (VGA), an output amplifier, a microphone-bias-voltage generator, and AGC control circuitry.

An internal microphone bias voltage generator provides a 2V bias that is suitable for most electret condenser microphones. The MAX9814 amplifies the input in three distinct stages. In the first stage, the input is buffered and amplified through the low-noise preamplifier with a gain of 12dB. The second stage consists of the VGA controlled by the AGC. The VGA/AGC combination is capable of varying the gain from 20dB to 0dB. The output amplifier is the final stage in which a fixed gain of 8dB, 18dB, 20dB is programmed through a single tri-level logic input. With no compression from the AGC, the MAX9814 is capable of providing 40dB, 50dB, or 60dB gain.

Automatic Gain Control (AGC)

A device without AGC experiences clipping at the output when too much gain is applied to the input. AGC prevents clipping at the output when too much gain is applied to the input, eliminating output clipping. Figure 1 shows a comparison of an over-gained microphone input with and without AGC.

The MAX9814's AGC controls the gain by first detecting that the output voltage has exceeded a preset limit. The microphone amplifier gain is then reduced with a selectable time constant to correct for the excessive output-voltage amplitude. This process is known as the attack time. When the output signal subsequently lowers in amplitude, the gain is held at the reduced state for a short period before slowly increasing to the normal value. This process is known as the hold and release time. The speed at which the amplifiers adjust to changing input signals is set by the external timing capacitor C_{CT} and the voltage applied to A/R. The AGC threshold can be set by adjusting V_{TH}. Gain reduction is a function of input signal amplitude with a maximum AGC attenuation of 20dB. Figure 2 shows the effect of an input burst exceeding the preset limit, output attack, hold and release times.

If the attack and release times are configured to respond too fast, audible artifacts often described as "pumping" or "breathing" can occur as the gain is rapidly adjusted to follow the dynamics of the signal. For best results, adjust the time constant of the AGC to accommodate the source material. For applications in which music CDs are the main audio source, a 160µs attack time with an 80ms release time is recommended. Music applications typically require a shorter release time than voice or movie content.

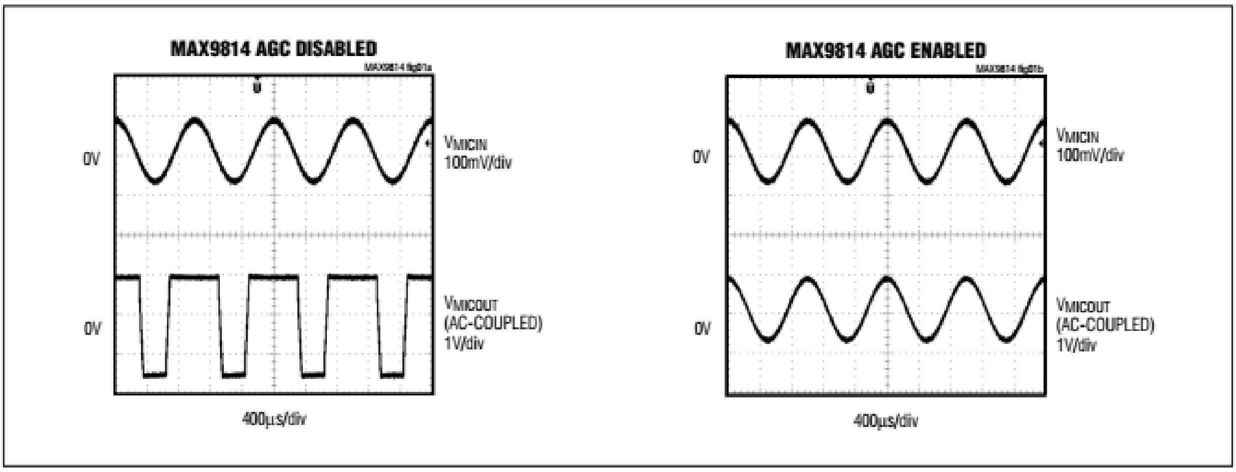


Figure 1. Microphone Input with and Without AGC

Setting the AGC Threshold

To set the output-voltage threshold at which the microphone output is clamped, an external resistor-divider must be connected from MICBIAS to ground with the output of the resistor-divider applied to TH. The voltage V_{TH} determines the peak output-voltage threshold at which the output becomes clamped. The maximum signal swing at the output is then limited to two times V_{TH} and remains at that level until the amplitude of the input signal is reduced. To disable AGC, connect TH to MICBIAS.

Microphone Bias Resistor

MICBIAS is capable of sourcing 20mA. Select a value for $R_{MICBIAS}$ that provides the desired bias current for the electret microphone. A value of 2.2kΩ is usually sufficient for a microphone of typical sensitivity. Consult the microphone data sheet for the recommended bias resistor.

Bias Capacitor

The BIAS output of the MAX9814 is internally buffered and provides a low-noise bias. Bypass BIAS with a 470nF capacitor to ground.

Input Capacitor

The input AC-coupling capacitor (C_{IN}) and the input resistance (R_{IN}) to the microphone amplifier form a highpass filter that removes any DC bias from an input signal (see the *Typical Application Circuit/Functional Diagram*). C_{IN} prevents any DC components from the input-signal source from appearing at the amplifier outputs. The -3dB point of the highpass filter, assuming zero source impedance due to the input signal source, is given by:

$$f_{-3dB\_IN} = \frac{1}{2\pi \times R_{IN} \times C_{IN}}$$

Choose C_{IN} such that f_{-3dB_IN} is well below the lowest frequency of interest. Setting f_{-3dB_IN} too high affects the amplifier's low-frequency response. Use capacitors with low-voltage coefficient dielectrics. Aluminum electrolytic, tantalum, or film dielectric capacitors are good choices for AC-coupling capacitors. Capacitors with high-voltage coefficients, such as ceramics (non-C0G dielectrics), can result in increased distortion at low frequencies.

Output Capacitor

The output of the MAX9814 is biased at 1.23V. To eliminate the DC offset, an AC-coupling capacitor (C_{OUT}) must be used. Depending on the input resistance (R_L) of the following stage, C_{OUT} and R_L effectively form a highpass filter. The -3dB point of the highpass filter, assuming zero output impedance, is given by:

$$f_{-3dB\_OUT} = \frac{1}{2\pi \times R_L \times C_{OUT}}$$

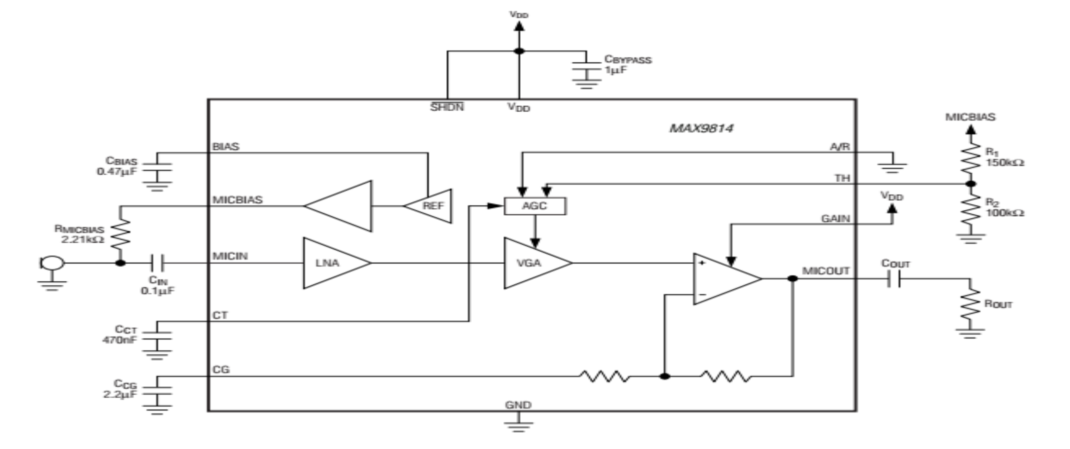
Shutdown

The MAX9814 features a low-power shutdown mode. When $\overline{SHDN}$ goes low, the supply current drops to 0.01µA, the output enters a high-impedance state, and the bias current to the microphone is switched off. Driving $\overline{SHDN}$ high enables the amplifier. Do not leave $\overline{SHDN}$ unconnected.

Power-Supply Bypassing and PCB Layout

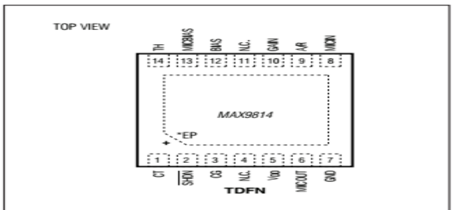
Bypass the power supply with a 0.1µF capacitor to ground. Reduce stray capacitance by minimizing trace lengths and place external components as close to the device as possible. Surface-mount components are recommended. In systems where analog and digital grounds are available, connect the MAX9814 to analog ground.

Typical Application Circuit/Functional Diagram



*THE DEVICE HAS BEEN CONFIGURED WITH AN ATTACK TIME OF 1.1ms, 40dB GAIN, AND AN ATTACK-AND-RELEASE RATIO OF 1:500.

Pin Configuration



Chip Information

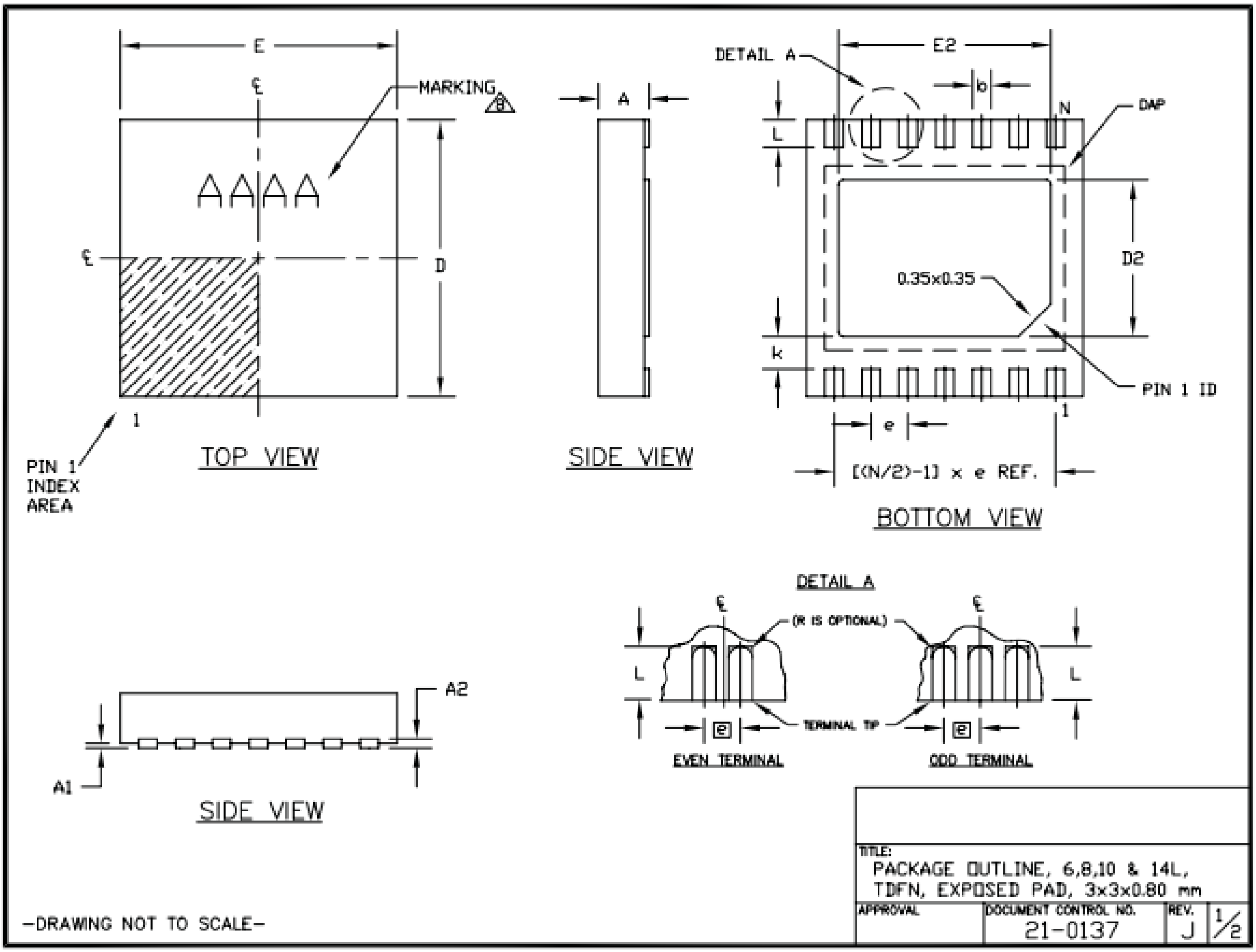
PROCESS: BiCMOS

Microphone Amplifier with AGC and Low-Noise Microphone Bias

Package Information

For the latest package outline information and land patterns, go to www.maxim-ic.com/packages.

PACKAGE TYPE	PACKAGE CODE	DOCUMENT NO.
14 TDFN-EP	T1433-2	21-0137



Package Information (continued)

For the latest package outline information and land patterns, go to www.maxim-ic.com/packages.

COMMON DIMENSIONS		
SYMBOL	MIN.	MAX.
A	0.70	0.80
D	2.90	3.10
E	2.90	3.10
A1	0.00	0.05
L	0.20	0.40
k	0.25 MIN.	
A2	0.20 REF.	

PACKAGE VARIATIONS							
PKG. CODE	N	D2	E2	e	JEDEC SPEC	b	[(N/2)-1] x e
T633-2	6	1.50±0.10	2.30±0.10	0.95 BSC	MO229 / WEEA	0.40±0.05	1.90 REF
T833-2	8	1.50±0.10	2.30±0.10	0.65 BSC	MO229 / WEEC	0.30±0.05	1.95 REF
T833-3	8	1.50±0.10	2.30±0.10	0.65 BSC	MO229 / WEEC	0.30±0.05	1.95 REF
T1033-1	10	1.50±0.10	2.30±0.10	0.50 BSC	MO229 / WEED-3	0.25±0.05	2.00 REF
T1033MK-1	10	1.50±0.10	2.30±0.10	0.50 BSC	MO229 / WEED-3	0.25±0.05	2.00 REF
T1033-2	10	1.50±0.10	2.30±0.10	0.50 BSC	MO229 / WEED-3	0.25±0.05	2.00 REF
T1433-1	14	1.70±0.10	2.30±0.10	0.40 BSC	----	0.20±0.05	2.40 REF
T1433-2	14	1.70±0.10	2.30±0.10	0.40 BSC	----	0.20±0.05	2.40 REF
T1433-3F	14	1.70±0.10	2.30±0.10	0.40 BSC	----	0.20±0.05	2.40 REF

NOTES:
1. ALL DIMENSIONS ARE IN mm. ANGLES IN DEGREES.
2. COPLANARITY SHALL NOT EXCEED 0.08 mm.
3. WARPAGE SHALL NOT EXCEED 0.10 mm.
4. PACKAGE LENGTH/PACKAGE WIDTH ARE CONSIDERED AS SPECIAL CHARACTERISTIC(S).
5. DRAWING CONFORMS TO JEDEC MO229, EXCEPT DIMENSIONS "D2" AND "E2", AND T1433-1 & T1433-2.
6. "N" IS THE TOTAL NUMBER OF LEADS.
7. NUMBER OF LEADS SHOWN ARE FOR REFERENCE ONLY.
8. MARKING IS FOR PACKAGE ORIENTATION REFERENCE ONLY.
9. ALL DIMENSIONS APPLY TO BOTH LEADED (–) AND PbFREE (+) PKG. CODES.

-DRAWING NOT TO SCALE-

TITLE: PACKAGE OUTLINE, 6,8,10 & 14L, TDFN, EXPOSED PAD, 3x3x0.80 mm			
APPROVAL	DOCUMENT CONTROL NO. 21-0137	REV. J	2/2

Ultra-Small, Low-Power, 16-Bit Analog-to-Digital Converter with Internal Reference

Check for Samples: ADS1113 ADS1114 ADS1115

FEATURES

- **ULTRA-SMALL QFN PACKAGE: 2mm × 1,5mm × 0,4mm**
- **WIDE SUPPLY RANGE: 2.0V to 5.5V**
- **LOW CURRENT CONSUMPTION: Continuous Mode: Only 150µA Single-Shot Mode: Auto Shut-Down**
- **PROGRAMMABLE DATA RATE: 8SPS to 860SPS**
- **INTERNAL LOW-DRIFT VOLTAGE REFERENCE**
- **INTERNAL OSCILLATOR**
- **INTERNAL PGA**
- **I²C™ INTERFACE: Pin-Selectable Addresses**
- **FOUR SINGLE-ENDED OR TWO DIFFERENTIAL INPUTS (ADS1115)**
- **PROGRAMMABLE COMPARATOR (ADS1114 and ADS1115)**

APPLICATIONS

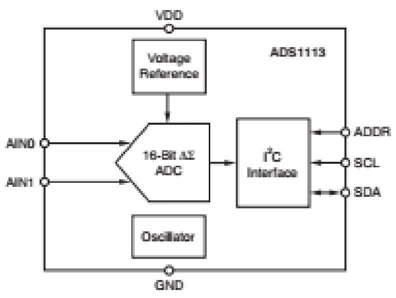
- **PORTABLE INSTRUMENTATION**
- **CONSUMER GOODS**
- **BATTERY MONITORING**
- **TEMPERATURE MEASUREMENT**
- **FACTORY AUTOMATION AND PROCESS CONTROLS**

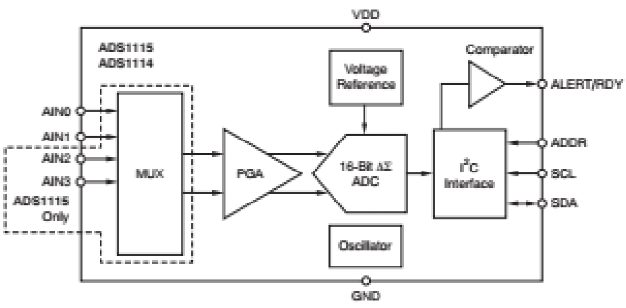
DESCRIPTION

The ADS1113, ADS1114, and ADS1115 are precision analog-to-digital converters (ADCs) with 16 bits of resolution offered in an ultra-small, leadless QFN-10 package or an MSOP-10 package. The ADS1113/4/5 are designed with precision, power, and ease of implementation in mind. The ADS1113/4/5 feature an onboard reference and oscillator. Data are transferred via an I²C-compatible serial interface; four I²C slave addresses can be selected. The ADS1113/4/5 operate from a single power supply ranging from 2.0V to 5.5V.

The ADS1113/4/5 can perform conversions at rates up to 860 samples per second (SPS). An onboard PGA is available on the ADS1114 and ADS1115 that offers input ranges from the supply to as low as ±256mV, allowing both large and small signals to be measured with high resolution. The ADS1115 also features an input multiplexer (MUX) that provides two differential or four single-ended inputs.

The ADS1113/4/5 operate either in continuous conversion mode or a single-shot mode that automatically powers down after a conversion and greatly reduces current consumption during idle periods. The ADS1113/4/5 are specified from –40°C to +125°C.





Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

I2C is a trademark of NXP Semiconductors.
All other trademarks are the property of their respective owners.

This integrated circuit can be damaged by ESD. Texas Instruments recommends that all integrated circuits be handled with appropriate precautions. Failure to observe proper handling and installation procedures can cause damage.

ESD damage can range from subtle performance degradation to complete device failure. Precision integrated circuits may be more susceptible to damage because very small parametric changes could cause the device not to meet its published specifications.

ORDERING INFORMATION

For the most current package and ordering information, see the Package Option Addendum at the end of this document, or see the TI web site at www.ti.com.

ABSOLUTE MAXIMUM RATINGS(1)

	ADS1113, ADS1114, ADS1115	UNIT
VDD to GND	–0.3 to +5.5	V
Analog input current	100, momentary	mA
Analog input current	10, continuous	mA
Analog input voltage to GND	–0.3 to VDD + 0.3	V
SDA, SCL, ADDR, ALERT/RDY voltage to GND	–0.5 to +5.5	V
Maximum junction temperature	+150	°C
Storage temperature range	–60 to +150	°C

(1) Stresses above those listed under *Absolute Maximum Ratings* may cause permanent damage to the device. Exposure to absolute maximum conditions for extended periods may affect device reliability.

PRODUCT FAMILY

DEVICE	PACKAGE DESIGNATOR MSOP/QFN	RESOLUTION (Bits)	MAXIMUM SAMPLE RATE (SPS)	COMPARATOR	PGA	INPUT CHANNELS (Differential/ Single-Ended)
ADS1113	BROI/N6J	16	860	No	No	1/1
ADS1114	BRNI/N5J	16	860	Yes	Yes	1/1
ADS1115	BOGI/N4J	16	860	Yes	Yes	2/4
ADS1013	BRMI/N9J	12	3300	No	No	1/1
ADS1014	BRQI/N8J	12	3300	Yes	Yes	1/1
ADS1015	BRPI/N7J	12	3300	Yes	Yes	2/4

ELECTRICAL CHARACTERISTICS

All specifications at –40°C to +125°C, VDD = 3.3V, and Full-Scale (FS) = ±2.048V, unless otherwise noted.
Typical values are at +25°C.

PARAMETER	TEST CONDITIONS	ADS1113, ADS1114, ADS1115 MIN	TYP	MAX	UNIT
ANALOG INPUT					
Full-scale input voltage(1)	V_{IN} = (AIN_P) – (AIN_N)		±4.096/PGA		V
Analog input voltage	AIN_P or AIN_N to GND	GND		VDD	V
Differential input impedance			See Table 2		
Common-mode input impedance	FS = ±6.144V(1)		10		MΩ
	FS = ±4.096V(1), ±2.048V		6		MΩ
	FS = ±1.024V		3		MΩ
	FS = ±0.512V, ±0.256V		100		MΩ
SYSTEM PERFORMANCE					
Resolution	No missing codes	16			Bits
Data rate (DR)			8, 16, 32, 64, 128, 250, 475, 860		SPS
Data rate variation	All data rates	–10		10	%
Output noise			See Typical Characteristics		
Integral nonlinearity	DR = 8SPS, FS = ±2.048V, best fit(2)			1	LSB
Offset error	FS = ±2.048V, differential inputs		±1	±3	LSB
	FS = ±2.048V, single-ended inputs		±3		LSB
Offset drift	FS = ±2.048V		0.005		LSB/°C
Offset power-supply rejection	FS = ±2.048V		1		LSB/V
Gain error(3)	FS = ±2.048V at 25°C		0.01	0.15	%
Gain drift(3)	FS = ±0.256V		7		ppm/°C
	FS = ±2.048V		5	40	ppm/°C
	FS = ±6.144V(1)		5		ppm/°C
Gain power-supply rejection			80		ppm/V
PGA gain match(3)	Match between any two PGA gains		0.02	0.1	%
Gain match	Match between any two inputs		0.05	0.1	%
Offset match	Match between any two inputs		3		LSB
Common-mode rejection	At dc and FS = ±0.256V		105		dB
	At dc and FS = ±2.048V		100		dB
	At dc and FS = ±6.144V(1)		90		dB
	f_{CM} = 60Hz, DR = 8SPS		105		dB
	f_{CM} = 50Hz, DR = 8SPS		105		dB
DIGITAL INPUT/OUTPUT					
Logic level					
V_{IH}		0.7VDD		5.5	V
V_{IL}		GND – 0.5		0.3VDD	V
V_{OL}	I_{OL} = 3mA	GND	0.15	0.4	V
Input leakage					
I_H	V_{IH} = 5.5V			10	μA
I_L	V_{IL} = GND	10			μA

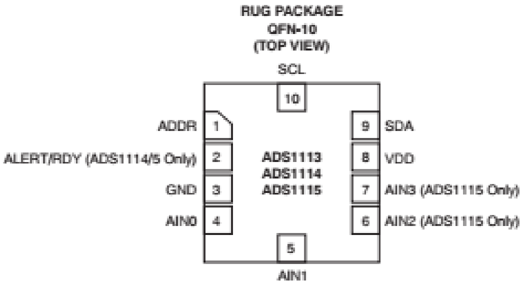
(1) This parameter expresses the full-scale range of the ADC scaling. In no event should more than VDD + 0.3V be applied to this device.
(2) 99% of full-scale.
(3) Includes all errors from onboard PGA and reference.

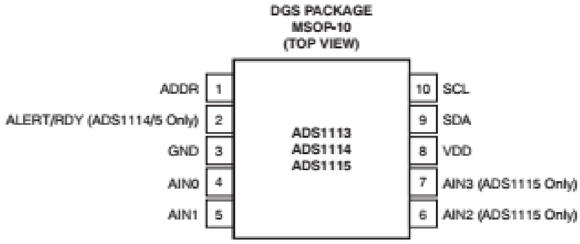
ELECTRICAL CHARACTERISTICS (continued)

All specifications at –40°C to +125°C, VDD = 3.3V, and Full-Scale (FS) = ±2.048V, unless otherwise noted.
Typical values are at +25°C.

PARAMETER	TEST CONDITIONS	ADS1113, ADS1114, ADS1115			UNIT
		MIN	TYP	MAX	
POWER-SUPPLY REQUIREMENTS					
Power-supply voltage		2		5.5	V
Supply current	Power-down current at 25°C		0.5	2	μA
	Power-down current up to 125°C			5	μA
	Operating current at 25°C		150	200	μA
	Operating current up to 125°C			300	μA
Power dissipation	VDD = 5.0V		0.9		mW
	VDD = 3.3V		0.5		mW
	VDD = 2.0V		0.3		mW
TEMPERATURE					
Storage temperature		–60		+150	°C
Specified temperature		–40		+125	°C

PIN CONFIGURATIONS





PIN DESCRIPTIONS

PIN #	DEVICE			ANALOG/ DIGITAL INPUT/ OUTPUT	DESCRIPTION
	ADS1113	ADS1114	ADS1115		
1	ADDR	ADDR	ADDR	Digital Input	I²C slave address select
2	NC(1)	ALERT/RDY	ALERT/RDY	Digital Output	Digital comparator output or conversion ready (NC for ADS1113)
3	GND	GND	GND	Analog	Ground
4	AIN0	AIN0	AIN0	Analog Input	Differential channel 1: Positive input or single-ended channel 1 input
5	AIN1	AIN1	AIN1	Analog Input	Differential channel 1: Negative input or single-ended channel 2 input
6	NC	NC	AIN2	Analog Input	Differential channel 2: Positive input or single-ended channel 3 input (NC for ADS1113/4)
7	NC	NC	AIN3	Analog Input	Differential channel 2: Negative input or single-ended channel 4 input (NC for ADS1113/4)
8	VDD	VDD	VDD	Analog	Power supply: 2.0V to 5.5V
9	SDA	SDA	SDA	Digital I/O	Serial data: Transmits and receives data
10	SCL	SCL	SCL	Digital Input	Serial clock input: Clocks data on SDA

(1) NC pins may be left floating or tied to ground.

TIMING REQUIREMENTS

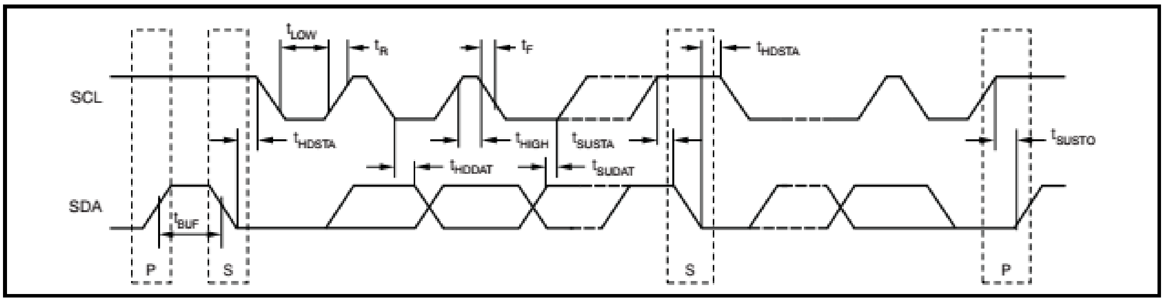


Figure 1. I²C Timing Diagram

Table 1. I²C Timing Definitions

PARAMETER		FAST MODE MIN	FAST MODE MAX	HIGH-SPEED MODE MIN	HIGH-SPEED MODE MAX	UNIT
SCL operating frequency	f_{SCL}	0.01	0.4	0.01	3.4	MHz
Bus free time between START and STOP condition	t_{BUF}	600		160		ns
Hold time after repeated START condition. After this period, the first clock is generated.	t_{HDSTA}	600		160		ns
Repeated START condition setup time	t_{SUSTA}	600		160		ns
Stop condition setup time	t_{SUSTO}	600		160		ns
Data hold time	t_{HDDAT}	0		0		ns
Data setup time	t_{SUDAT}	100		10		ns
SCL clock low period	t_{LOW}	1300		160		ns
SCL clock high period	t_{HIGH}	600		60		ns
Clock/data fall time	t_F		300		160	ns
Clock/data rise time	t_R		300		160	ns

TYPICAL CHARACTERISTICS

At T_A = +25°C and VDD = 3.3V, unless otherwise noted.

OPERATING CURRENT vs TEMPERATURE

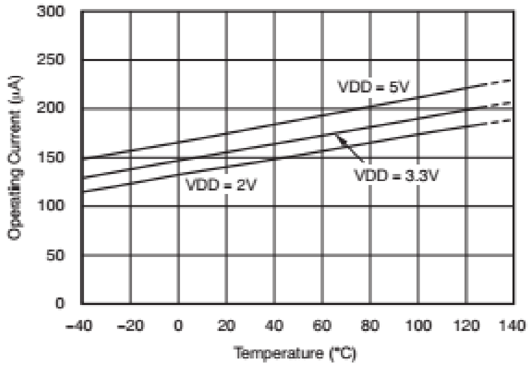


Figure 2.

SHUTDOWN CURRENT vs TEMPERATURE

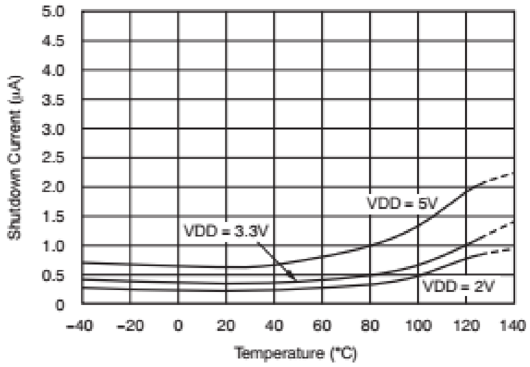


Figure 3.

SINGLE-ENDED OFFSET ERROR vs TEMPERATURE(1)

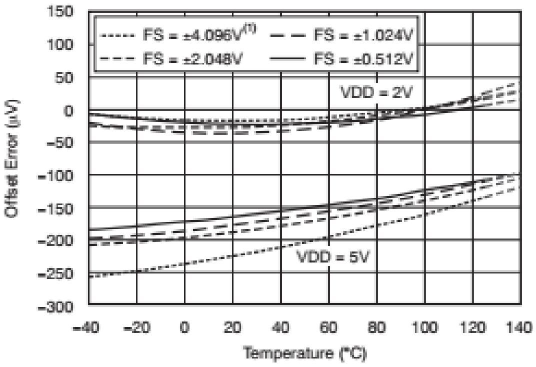


Figure 4.

DIFFERENTIAL OFFSET vs TEMPERATURE

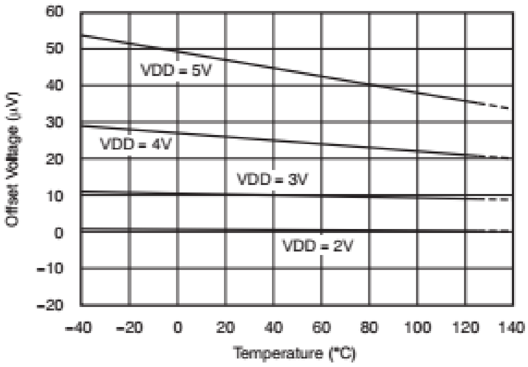


Figure 5.

GAIN ERROR vs TEMPERATURE

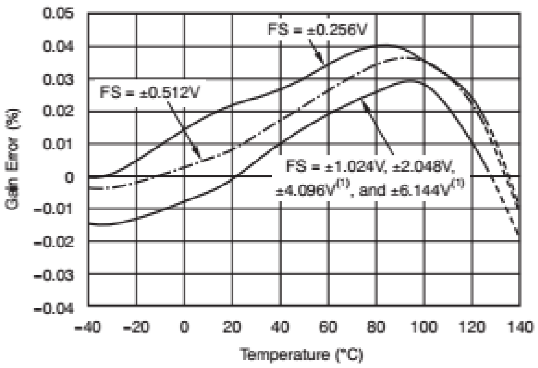


Figure 6.

GAIN ERROR vs SUPPLY

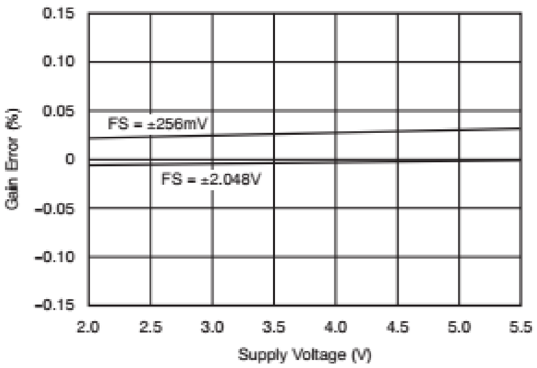


Figure 7.

(1) This parameter expresses the full-scale range of the ADC scaling. In no event should more than VDD + 0.3V be applied to this device.

TYPICAL CHARACTERISTICS (continued)

At T_A = +25°C and VDD = 3.3V, unless otherwise noted.

INL vs SUPPLY VOLTAGE(2)

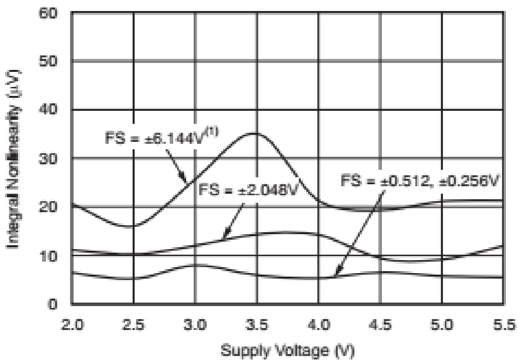


Figure 8.

INL vs INPUT SIGNAL

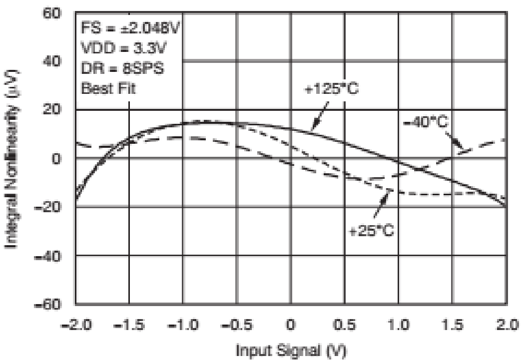


Figure 9.

INL vs INPUT SIGNAL

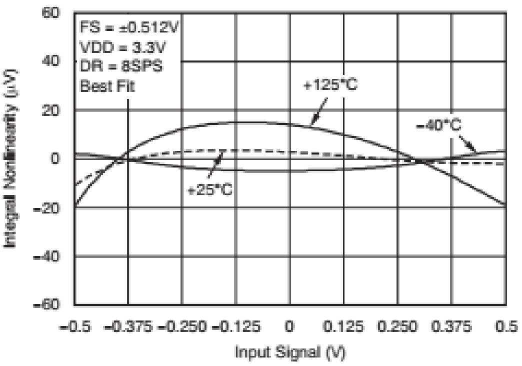


Figure 10.

INL vs INPUT SIGNAL

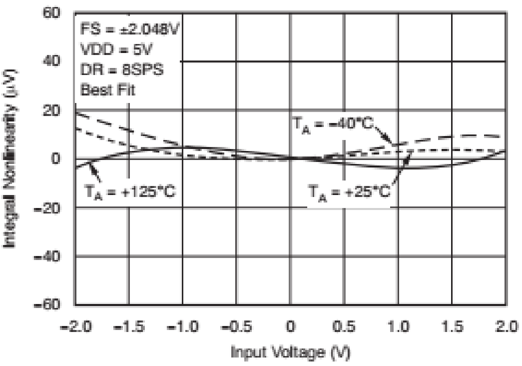


Figure 11.

INL vs INPUT SIGNAL

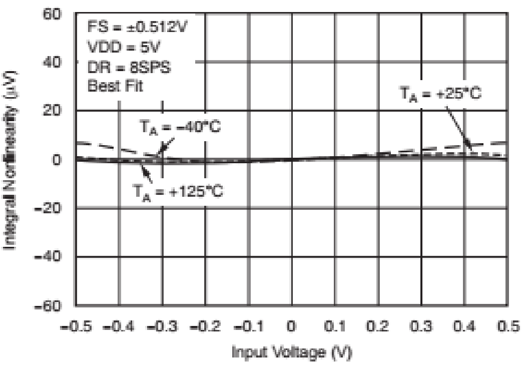


Figure 12.

INL vs TEMPERATURE

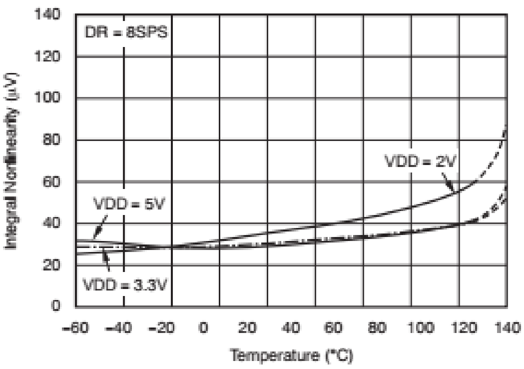


Figure 13.

(2) This parameter expresses the full-scale range of the ADC scaling. In no event should more than VDD + 0.3V be applied to this device.

TL081

General purpose JFET single operational amplifiers

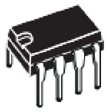
Features

- Wide common-mode (up to V_{CC}^{+}) and differential voltage range
- Low input bias and offset current
- Output short-circuit protection
- High input impedance JFET input stage
- Internal frequency compensation
- Latch-up free operation
- High slew rate: 16 V/µs (typ)

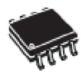
Description

The TL081, TL081A and TL081B are high-speed JFET input single operational amplifiers incorporating well matched, high-voltage JFET and bipolar transistors in a monolithic integrated circuit.

The devices feature high slew rates, low input bias and offset currents, and low offset voltage temperature coefficient.

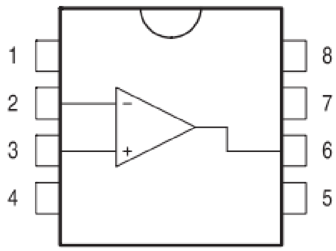


N
DIP8
(Plastic package)



D
SO-8
(Plastic micropackage)

Pin connections
(top view)



1 - Offset null 1
2 - Inverting input
3 - Non-inverting input
4 - V_{CC}^{-}
5 - Offset null 2
6 - Output
7 - V_{CC}^{+}
8 - N.C.

1 Schematic diagram

Figure 1. Schematic diagram

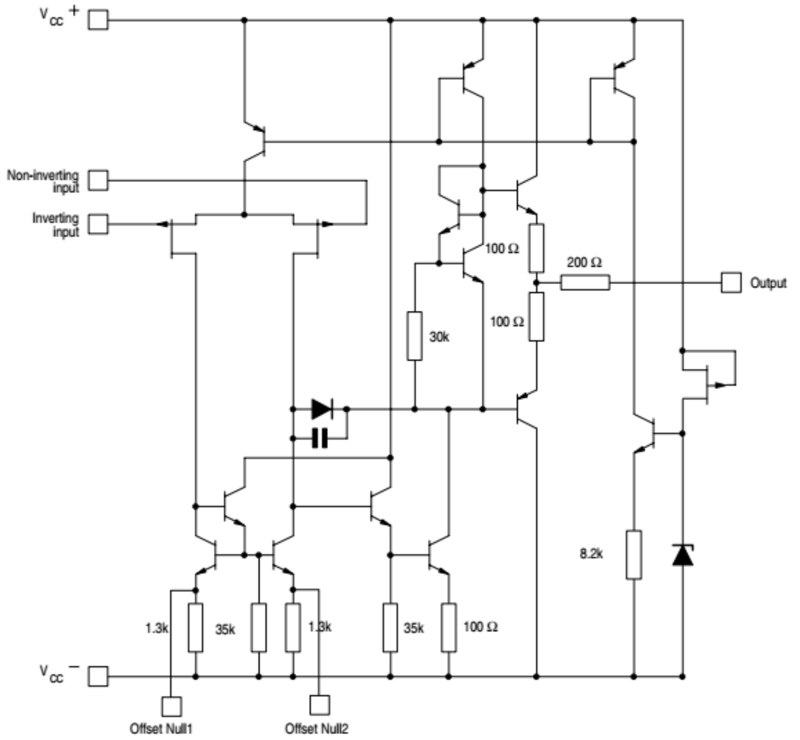
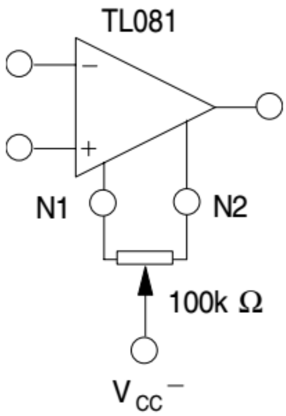


Figure 2. Input offset voltage null circuit



2 Absolute maximum ratings

Table 1. Absolute maximum ratings

Symbol	Parameter	TL081I, AI, BI	TL081C, AC, BC	Unit
V_{CC}	Supply voltage (1)	±18		V
V_{in}	Input voltage (2)	±15		V
V_{id}	Differential input voltage (3)	±30		V
P_{tot}	Power dissipation	680		mW
	Output short-circuit duration (4)	Infinite		
T_{stg}	Storage temperature range	-65 to +150		°C
R_{thja}	Thermal resistance junction to ambient(5) (6) SO-8 DIP8	 125 85		°C/W
R_{thjc}	Thermal resistance junction to case(5) (6) SO-8 DIP8	 40 41		°C/W
ESD	HBM: human body model(7)	500		V
	MM: machine model(8)	200		V
	CDM: charged device model(9)	1.5		kV

1. All voltage values, except differential voltage, are with respect to the zero reference level (ground) of the supply voltages where the zero reference level is the midpoint between V_{CC}^+ and V_{CC}^-.
2. The magnitude of the input voltage must never exceed the magnitude of the supply voltage or 15 volts, whichever is less.
3. Differential voltages are the non-inverting input terminal with respect to the inverting input terminal.
4. The output may be shorted to ground or to either supply. Temperature and/or supply voltages must be limited to ensure that the dissipation rating is not exceeded.
5. Short-circuits can cause excessive heating and destructive dissipation.
6. Rth are typical values.
7. Human body model: 100 pF discharged through a 1.5kΩ resistor between two pins of the device, done for all couples of pin combinations with other pins floating.
8. Machine model: a 200 pF cap is charged to the specified voltage, then discharged directly between two pins of the device with no external series resistor (internal resistor < 5 Ω), done for all couples of pin combinations with other pins floating.
9. Charged device model: all pins plus package are charged together to the specified voltage and then discharged directly to the ground.

Table 2. Operating conditions

Symbol	Parameter	TL081I, AI, BI	TL081C, AC, BC	Unit
V_{CC}	Supply voltage range	6 to 36		V
T_{oper}	Operating free-air temperature range	-40 to +105	0 to +70	°C

3 Electrical characteristics

Table 3. $V_{CC} = \pm15V$, $T_{amb} = +25°C$ (unless otherwise specified)

Symbol	Parameter	TL081I, AC, AI, BC, BI			TL081C			Unit
		Min.	Typ.	Max.	Min.	Typ.	Max.	
V_{io}	Input offset voltage ($R_s = 50\Omega$) $T_{amb} = +25°C$ TL081 TL081A TL081B $T_{min} \le T_{amb} \le T_{max}$ TL081 TL081A TL081B		 3 3 1	 10 6 3 13 7 5		 3	 10 13	mV
DV_{io}	Input offset voltage drift		10			10		μV/°C
I_{io}	Input offset current [1] $T_{amb} = +25°C$ $T_{min} \le T_{amb} \le T_{max}$		 5	 100 4		 5	 100 10	 pA nA
I_{ib}	Input bias current [1] $T_{amb} = +25°C$ $T_{min} \le T_{amb} \le T_{max}$		 20	 200 20		 20	 400 20	nA
A_{vd}	Large signal voltage gain ($R_L = 2k\Omega$, $V_o = \pm10V$) $T_{amb} = +25°C$ $T_{min} \le T_{amb} \le T_{max}$	 50 25	 200		 25 15	 200		V/mV
SVR	Supply voltage rejection ratio ($R_S = 50\Omega$) $T_{amb} = +25°C$ $T_{min} \le T_{amb} \le T_{max}$	 80 80	 86		 70 70	 86		dB
I_{CC}	Supply current, no load $T_{amb} = +25°C$ $T_{min} \le T_{amb} \le T_{max}$		 1.4	 2.5 2.5		 1.4	 2.5 2.5	mA
V_{icm}	Input common mode voltage range	±11	+15 -12		±11	+15 -12		V
CMR	Common mode rejection ratio ($R_S = 50\Omega$) $T_{amb} = +25°C$ $T_{min} \le T_{amb} \le T_{max}$	 80 80	 86		 70 70	 86		dB
I_{os}	Output short-circuit current $T_{amb} = +25°C$ $T_{min} \le T_{amb} \le T_{max}$	 10 10	 40	 60 60	 10 10	 40	 60 60	mA
$\pm V_{opp}$	Output voltage swing $T_{amb} = +25°C$ $R_L = 2k\Omega$ $R_L = 10k\Omega$ $T_{min} \le T_{amb} \le T_{max}$ $R_L = 2k\Omega$ $R_L = 10k\Omega$	 10 12 10 12	 12 13.5		 10 12 10 12	 12 13.5		V
SR	Slew rate ($T_{amb} = +25°C$) $V_{in} = 10V$, $R_L = 2k\Omega$, $C_L = 100pF$, unity gain	8	16		8	16		V/μs

Table 3. V_{CC} = ±15V, T_{amb} = +25°C (unless otherwise specified) (continued)

Symbol	Parameter	TL081I, AC, AI, BC, BI			TL081C			Unit
		Min.	Typ.	Max.	Min.	Typ.	Max.	
t_r	Rise time (T_{amb} = +25°C) V_{in} = 20mV, R_L = 2kΩ, C_L = 100pF, unity gain		0.1			0.1		µs
K_{ov}	Overshoot (T_{amb} = +25°C) V_{in} = 20mV, R_L = 2kΩ, C_L = 100pF, unity gain		10			10		%
GBP	Gain bandwidth product (T_{amb} = +25°C) V_{in} = 10mV, R_L = 2kΩ, C_L = 100pF, F= 100kHz	2.5	4		2.5	4		MHz
R_i	Input resistance		10^{12}			10^{12}		Ω
THD	Total harmonic distortion (T_{amb} = +25°C), F= 1kHz, R_L = 2kΩ, C_L = 100pF, A_V = 20dB, V_o = $2V_{pp}$		0.01			0.01		%
e_n	Equivalent input noise voltage R_S = 100Ω, F= 1kHz		15			15		$\frac{nV}{\sqrt{Hz}}$
∅m	Phase margin		45			45		degrees

1. The input bias currents are junction leakage currents which approximately double for every 10°C increase in the junction temperature.

Figure 3. Maximum peak-to-peak output voltage versus frequency

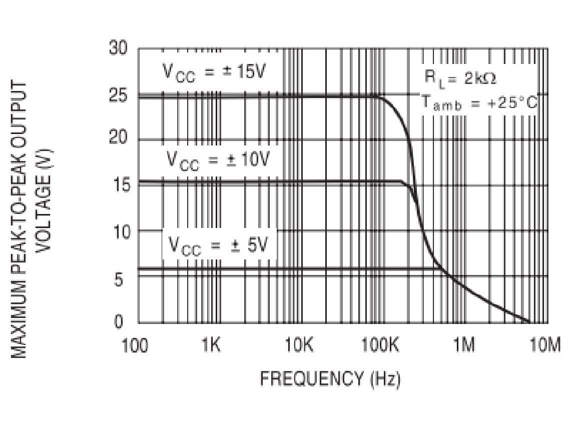


Figure 4. Maximum peak-to-peak output voltage versus frequency

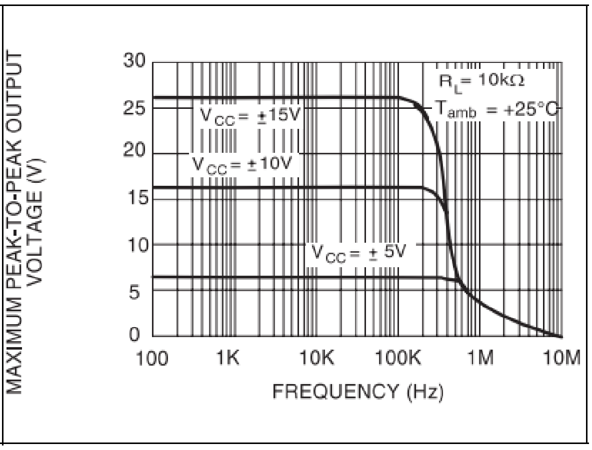


Figure 5. Maximum peak-to-peak output voltage versus frequency

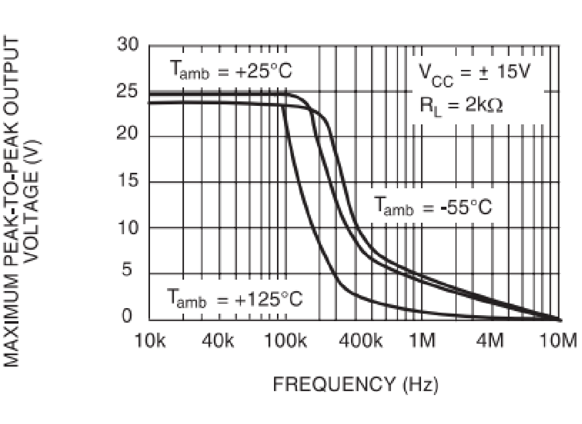


Figure 6. Maximum peak-to-peak output voltage versus free air temperature

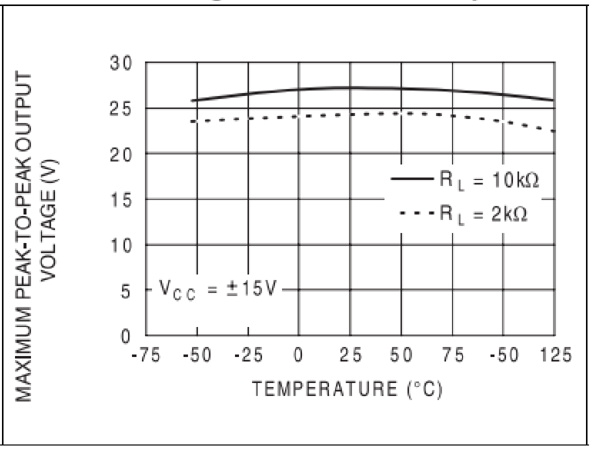


Figure 7. Maximum peak-to-peak output voltage versus load resistance

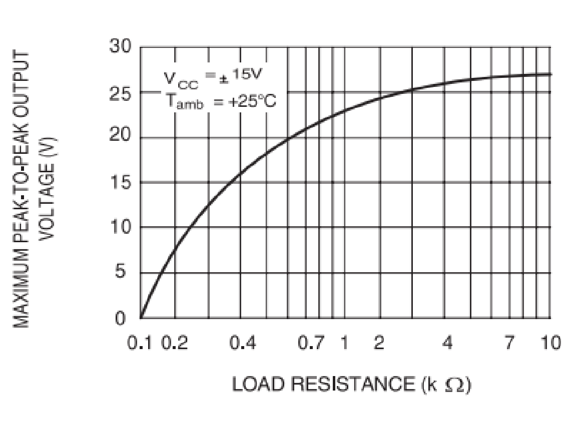


Figure 8. Maximum peak-to-peak output voltage versus supply voltage

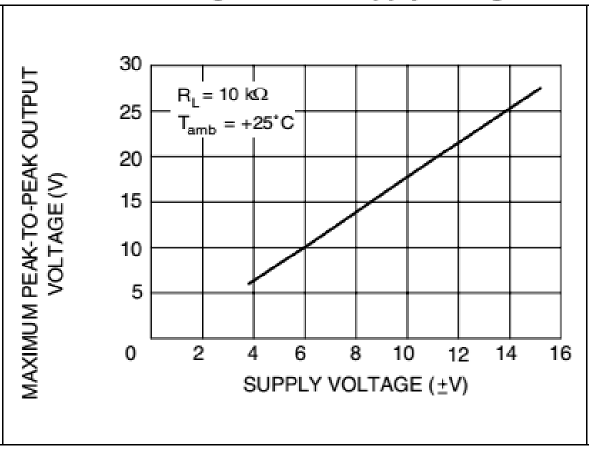


Figure 9. Input bias current versus free air temperature

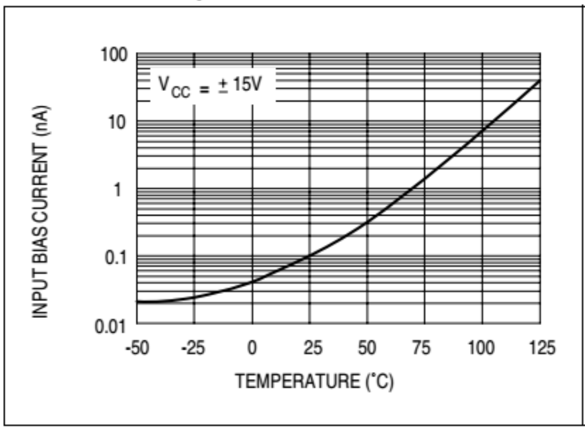


Figure 10. Large signal differential voltage amplification versus free air temp

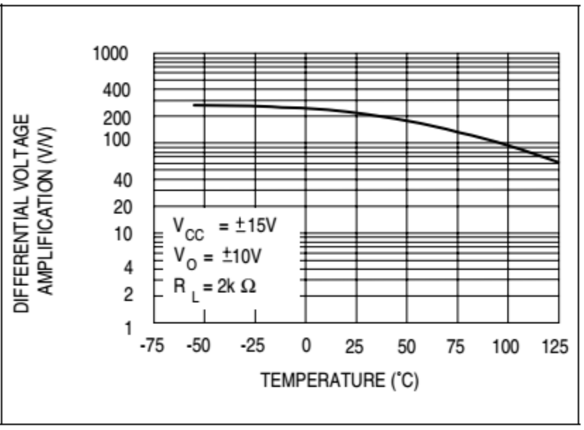


Figure 11. Large signal differential voltage amplification and phase shift versus frequency

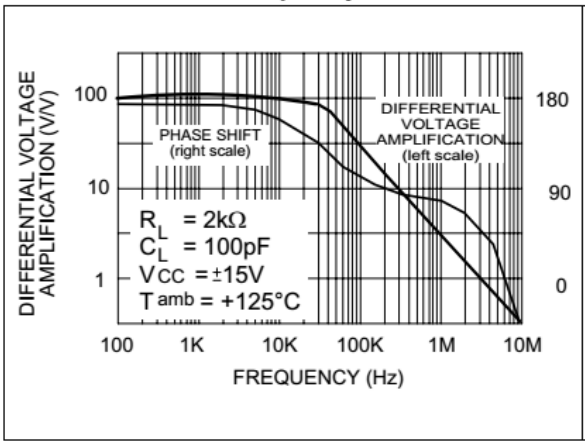


Figure 12. Total power dissipation versus free air temperature

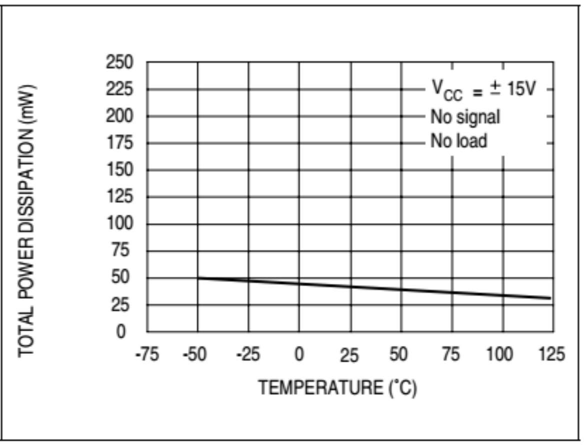


Figure 13. Supply current per amplifier versus free air temperature

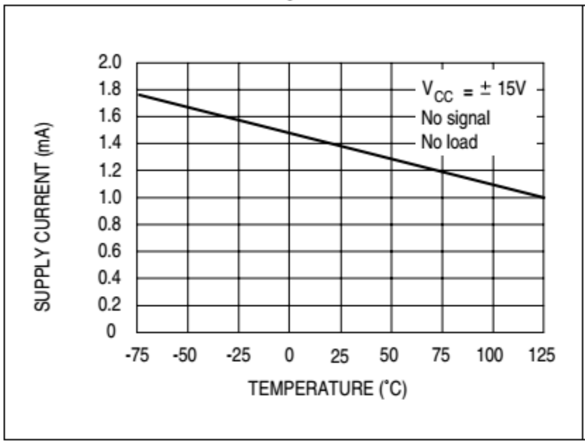


Figure 14. Supply current per amplifier versus supply voltage

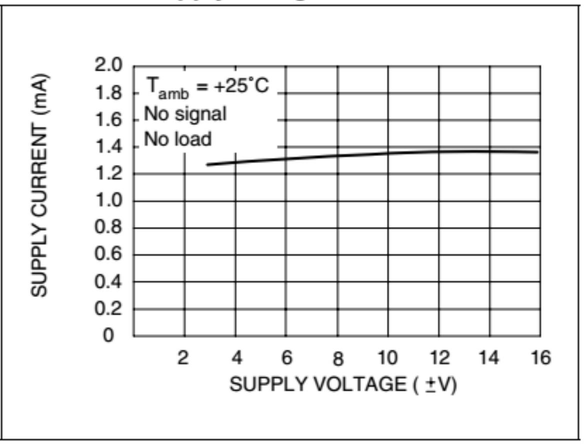


Figure 15. Common mode rejection ratio versus free air temperature

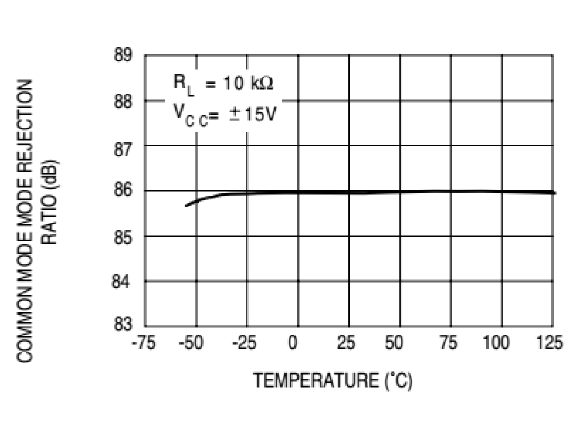


Figure 16. Equivalent input noise voltage versus frequency

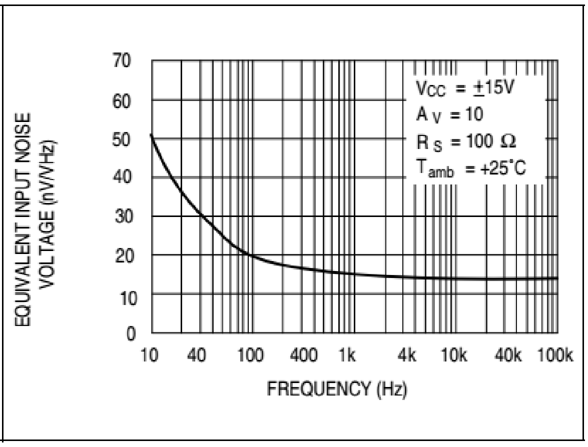


Figure 17. Output voltage versus elapsed time

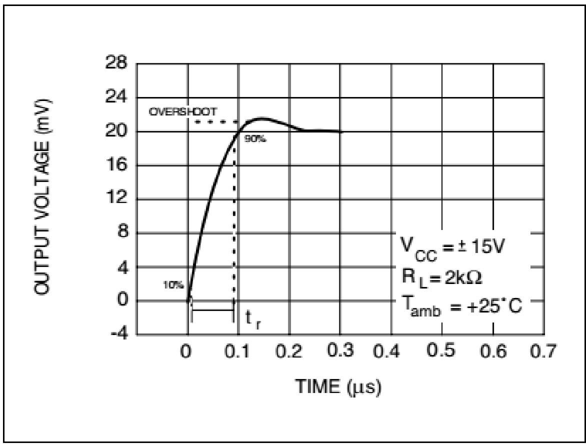


Figure 18. Total harmonic distortion versus frequency

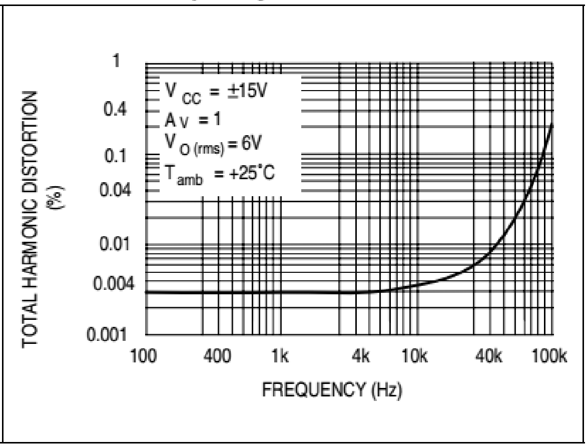
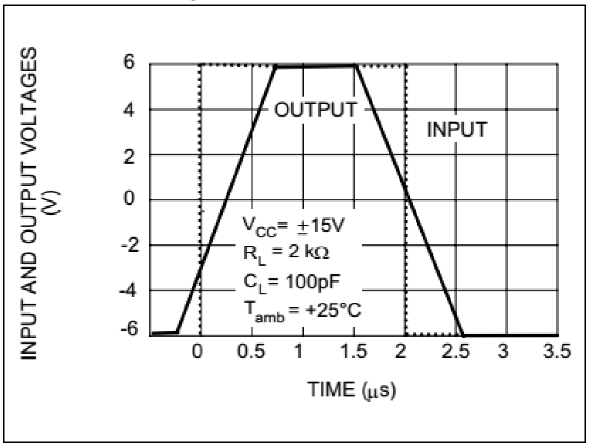


Figure 19. Voltage follower large signal pulse response



4 Parameter measurement information

Figure 20. Voltage follower

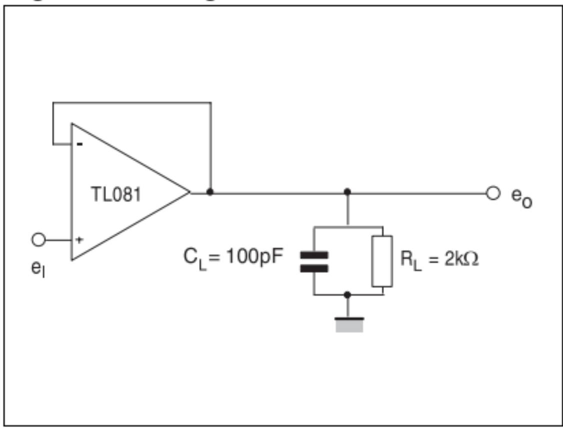
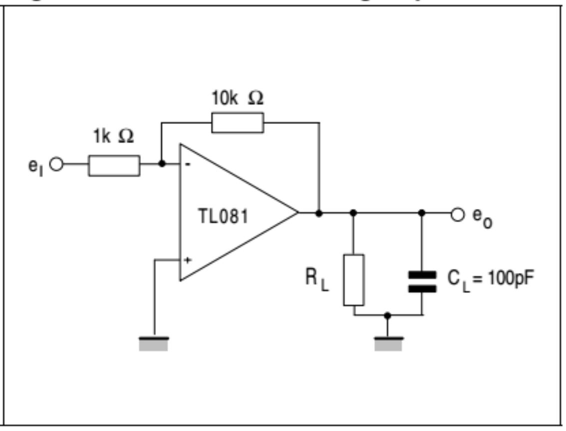


Figure 21. Gain-of-10 inverting amplifier



5 Typical applications

Figure 22. 0.5 Hz square wave oscillator

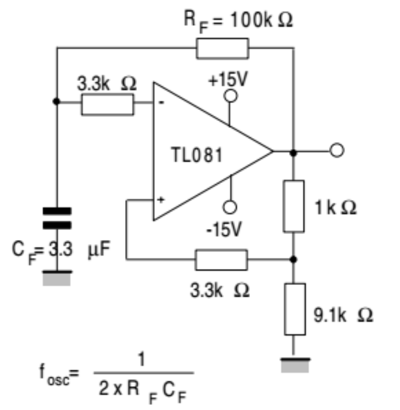
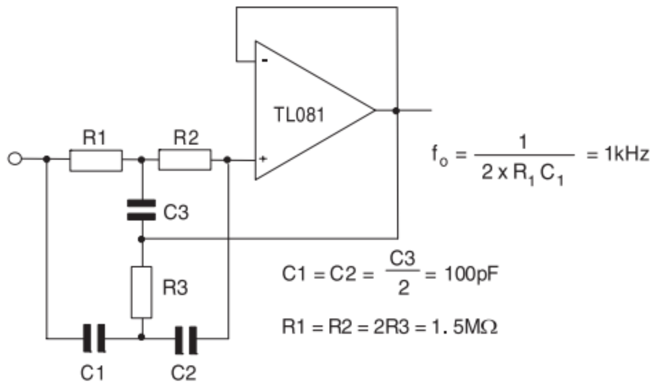


Figure 23. High Q notch filter



6 Package information

In order to meet environmental requirements, ST offers these devices in ECOPACK® packages. These packages have a lead-free second level interconnect. The category of second level interconnect is marked on the package and on the inner box label, in compliance with JEDEC Standard JESD97. The maximum ratings related to soldering conditions are also marked on the inner box label. ECOPACK is an ST trademark. ECOPACK specifications are available at: www.st.com.

6.1 DIP 8 package information

Figure 24. DIP8 package mechanical drawing

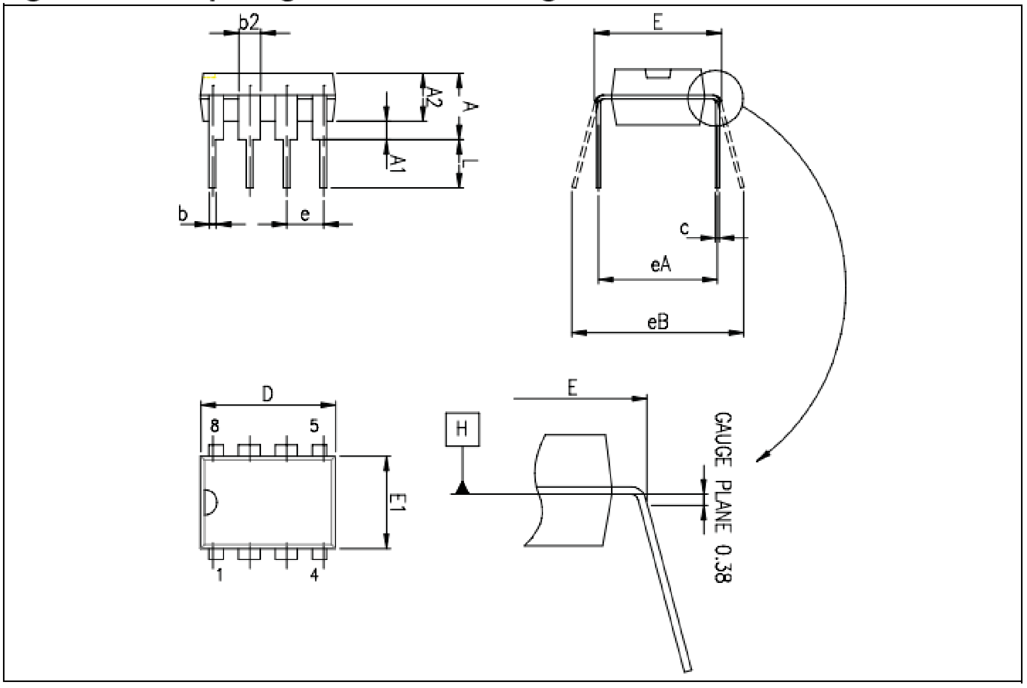


Table 4. DIP8 package mechanical data

Ref.	Dimensions					
	Millimeters			Inches		
	Min.	Typ.	Max.	Min.	Typ.	Max.
A			5.33			0.210
A1	0.38			0.015		
A2	2.92	3.30	4.95	0.115	0.130	0.195
b	0.36	0.46	0.56	0.014	0.018	0.022
b2	1.14	1.52	1.78	0.045	0.060	0.070
c	0.20	0.25	0.36	0.008	0.010	0.014
D	9.02	9.27	10.16	0.355	0.365	0.400
E	7.62	7.87	8.26	0.300	0.310	0.325
E1	6.10	6.35	7.11	0.240	0.250	0.280
e		2.54			0.100	
eA		7.62			0.300	
eB			10.92			0.430
L	2.92	3.30	3.81	0.115	0.130	0.150

6.2 SO-8 package information

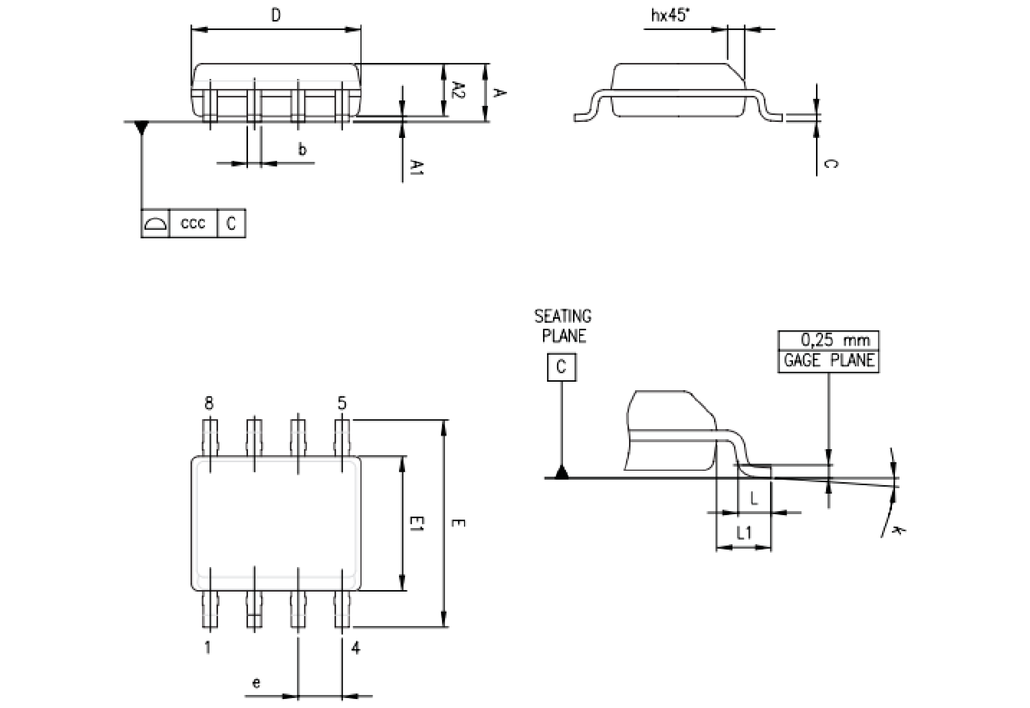


Figure 25. SO-8 package mechanical drawing

Table 5. SO-8 package mechanical data

Ref.	Dimensions					
	Millimeters			Inches		
	Min.	Typ.	Max.	Min.	Typ.	Max.
A			1.75			0.069
A1	0.10		0.25	0.004		0.010
A2	1.25			0.049		
b	0.28		0.48	0.011		0.019
c	0.17		0.23	0.007		0.010
D	4.80	4.90	5.00	0.189	0.193	0.197
E	5.80	6.00	6.20	0.228	0.236	0.244
E1	3.80	3.90	4.00	0.150	0.154	0.157
e		1.27			0.050	
h	0.25		0.50	0.010		0.020
L	0.40		1.27	0.016		0.050
k	1°		8°	1°		8°
ccc			0.10			0.004

August 2000

National Semiconductor

LM741 Operational Amplifier

General Description

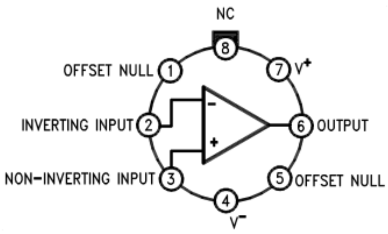
The LM741 series are general purpose operational amplifiers which feature improved performance over industry standards like the LM709. They are direct, plug-in replacements for the 709C, LM201, MC1439 and 748 in most applications.

The amplifiers offer many features which make their application nearly foolproof: overload protection on the input and output, no latch-up when the common mode range is exceeded, as well as freedom from oscillations.

The LM741C is identical to the LM741/LM741A except that the LM741C has their performance guaranteed over a 0˚C to +70˚C temperature range, instead of −55˚C to +125˚C.

Connection Diagrams

Metal Can Package

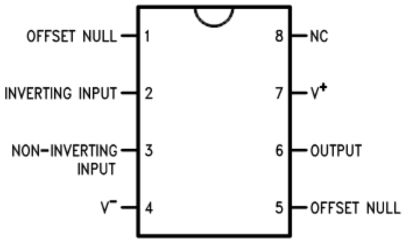


DS009341-2

Note 1: LM741H is available per JM38510/10101

Order Number LM741H, LM741H/883 (Note 1), **LM741AH/883 or LM741CH**
See NS Package Number H08C

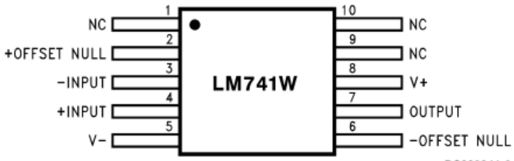
Dual-In-Line or S.O. Package



DS009341-3

Order Number LM741J, LM741J/883, LM741CN
See NS Package Number J08A, M08A or N08E

Ceramic Flatpak

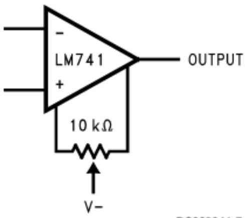


DS009341-6

Order Number LM741W/883
See NS Package Number W10A

Typical Application

Offset Nulling Circuit



DS009341-7

Absolute Maximum Ratings (Note 2)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/ Distributors for availability and specifications.

(Note 7)

	LM741A	LM741	LM741C
Supply Voltage	±22V	±22V	±18V
Power Dissipation (Note 3)	500 mW	500 mW	500 mW
Differential Input Voltage	±30V	±30V	±30V
Input Voltage (Note 4)	±15V	±15V	±15V
Output Short Circuit Duration	Continuous	Continuous	Continuous
Operating Temperature Range	−55°C to +125°C	−55°C to +125°C	0°C to +70°C
Storage Temperature Range	−65°C to +150°C	−65°C to +150°C	−65°C to +150°C
Junction Temperature	150°C	150°C	100°C
Soldering Information			
N-Package (10 seconds)	260°C	260°C	260°C
J- or H-Package (10 seconds)	300°C	300°C	300°C
M-Package			
Vapor Phase (60 seconds)	215°C	215°C	215°C
Infrared (15 seconds)	215°C	215°C	215°C

See AN-450 “Surface Mounting Methods and Their Effect on Product Reliability” for other methods of soldering surface mount devices.

	LM741A	LM741	LM741C
ESD Tolerance (Note 8)	400V	400V	400V

Electrical Characteristics (Note 5)

Parameter	Conditions	LM741A			LM741			LM741C			Units
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Input Offset Voltage	$T_A = 25°C$										
	$R_S \leq 10\ k\Omega$					1.0	5.0		2.0	6.0	mV
	$R_S \leq 50\Omega$		0.8	3.0							mV
	$T_{AMIN} \leq T_A \leq T_{AMAX}$										
	$R_S \leq 50\Omega$			4.0							mV
	$R_S \leq 10\ k\Omega$						6.0			7.5	mV
Average Input Offset Voltage Drift				15							μV/°C
Input Offset Voltage Adjustment Range	$T_A = 25°C$, $V_S = \pm 20V$	±10				±15			±15		mV
Input Offset Current	$T_A = 25°C$		3.0	30		20	200		20	200	nA
	$T_{AMIN} \leq T_A \leq T_{AMAX}$			70		85	500			300	nA
Average Input Offset Current Drift				0.5							nA/°C
Input Bias Current	$T_A = 25°C$		30	80		80	500		80	500	nA
	$T_{AMIN} \leq T_A \leq T_{AMAX}$			0.210			1.5			0.8	μA
Input Resistance	$T_A = 25°C$, $V_S = \pm 20V$	1.0	6.0		0.3	2.0		0.3	2.0		MΩ
	$T_{AMIN} \leq T_A \leq T_{AMAX}$, $V_S = \pm 20V$	0.5									MΩ
Input Voltage Range	$T_A = 25°C$							±12	±13		V
	$T_{AMIN} \leq T_A \leq T_{AMAX}$				±12	±13					V

Electrical Characteristics (Note 5) (Continued)

Parameter	Conditions	LM741A			LM741			LM741C			Units
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Large Signal Voltage Gain	T_A = 25˚C, $R_L \geq 2$ kΩ										
	V_S = ±20V, V_O = ±15V	50									V/mV
	V_S = ±15V, V_O = ±10V				50	200		20	200		V/mV
	$T_{AMIN} \leq T_A \leq T_{AMAX}$,										
	$R_L \geq 2$ kΩ,										
	V_S = ±20V, V_O = ±15V	32									V/mV
	V_S = ±15V, V_O = ±10V				25			15			V/mV
	V_S = ±5V, V_O = ±2V	10									V/mV
Output Voltage Swing	V_S = ±20V										
	$R_L \geq 10$ kΩ	±16									V
	$R_L \geq 2$ kΩ	±15									V
	V_S = ±15V										
	$R_L \geq 10$ kΩ				±12	±14		±12	±14		V
	$R_L \geq 2$ kΩ				±10	±13		±10	±13		V
Output Short Circuit	T_A = 25˚C	10	25	35		25			25		mA
Current	$T_{AMIN} \leq T_A \leq T_{AMAX}$	10		40							mA
Common-Mode	$T_{AMIN} \leq T_A \leq T_{AMAX}$										
Rejection Ratio	$R_S \leq 10$ kΩ, V_{CM} = ±12V				70	90		70	90		dB
	$R_S \leq 50Ω$, V_{CM} = ±12V	80	95								dB
Supply Voltage Rejection	$T_{AMIN} \leq T_A \leq T_{AMAX}$,										
Ratio	V_S = ±20V to V_S = ±5V										
	$R_S \leq 50Ω$	86	96								dB
	$R_S \leq 10$ kΩ				77	96		77	96		dB
Transient Response	T_A = 25˚C, Unity Gain										
Rise Time			0.25	0.8		0.3			0.3		µs
Overshoot			6.0	20		5			5		%
Bandwidth (Note 6)	T_A = 25˚C	0.437	1.5								MHz
Slew Rate	T_A = 25˚C, Unity Gain	0.3	0.7			0.5			0.5		V/µs
Supply Current	T_A = 25˚C					1.7	2.8		1.7	2.8	mA
Power Consumption	T_A = 25˚C										
	V_S = ±20V		80	150							mW
	V_S = ±15V					50	85		50	85	mW
LM741A	V_S = ±20V										
	T_A = T_{AMIN}			165							mW
	T_A = T_{AMAX}			135							mW
LM741	V_S = ±15V										
	T_A = T_{AMIN}					60	100				mW
	T_A = T_{AMAX}					45	75				mW

Note 2: "Absolute Maximum Ratings" indicate limits beyond which damage to the device may occur. Operating Ratings indicate conditions for which the device is functional, but do not guarantee specific performance limits.

Electrical Characteristics (Note 5) (Continued)

Note 3: For operation at elevated temperatures, these devices must be derated based on thermal resistance, and T_j max. (listed under "Absolute Maximum Ratings"). $T_j = T_A + (\theta_{jA} P_D)$.

Thermal Resistance	Cerdip (J)	DIP (N)	HO8 (H)	SO-8 (M)
θ_{jA} (Junction to Ambient)	100˚C/W	100˚C/W	170˚C/W	195˚C/W
θ_{jC} (Junction to Case)	N/A	N/A	25˚C/W	N/A

Note 4: For supply voltages less than ±15V, the absolute maximum input voltage is equal to the supply voltage.

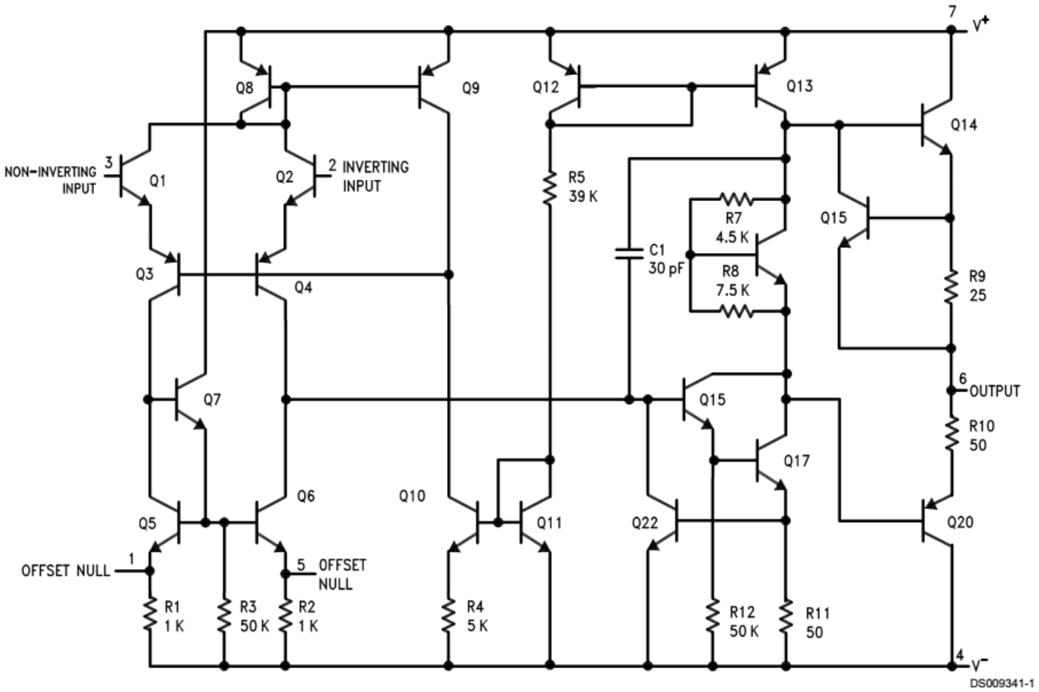
Note 5: Unless otherwise specified, these specifications apply for V_S = ±15V, −55˚C ≤ T_A ≤ +125˚C (LM741/LM741A). For the LM741C/LM741E, these specifications are limited to 0˚C ≤ T_A ≤ +70˚C.

Note 6: Calculated value from: BW (MHz) = 0.35/Rise Time(µs).

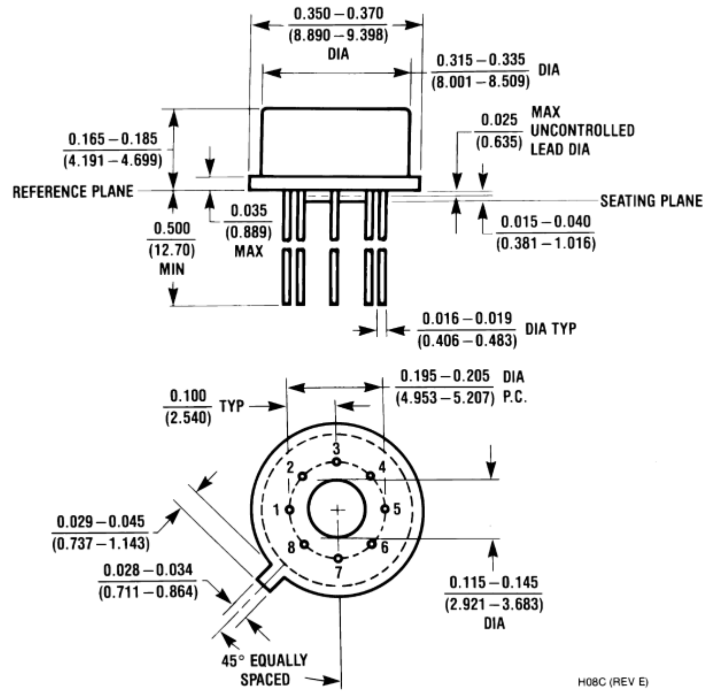
Note 7: For military specifications see RETS741X for LM741 and RETS741AX for LM741A.

Note 8: Human body model, 1.5 kΩ in series with 100 pF.

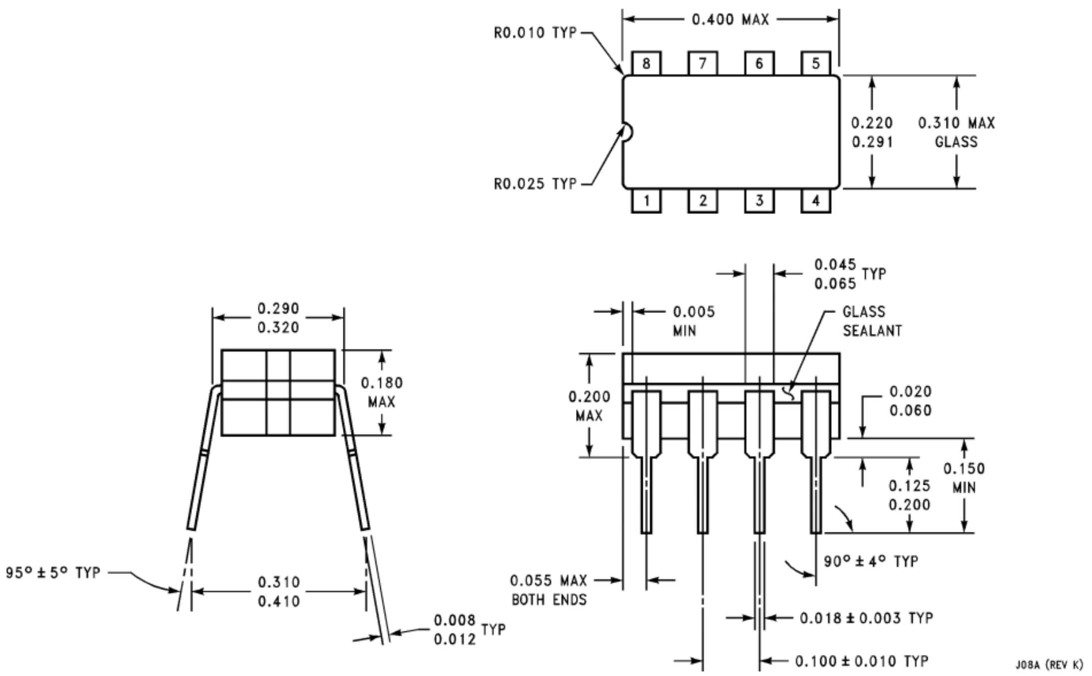
Schematic Diagram



Physical Dimensions inches (millimeters) unless otherwise noted

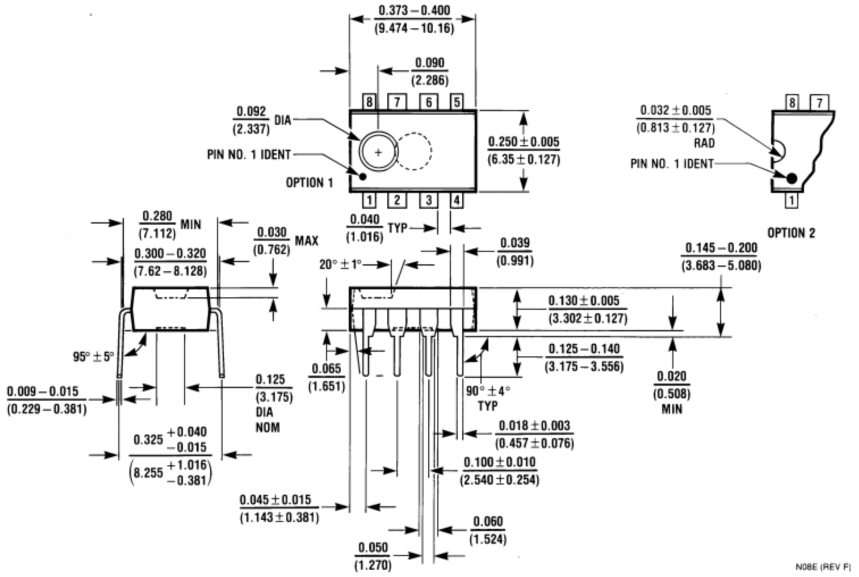


Metal Can Package (H)
Order Number LM741H, LM741H/883, LM741AH/883, LM741AH-MIL or LM741CH
NS Package Number H08C

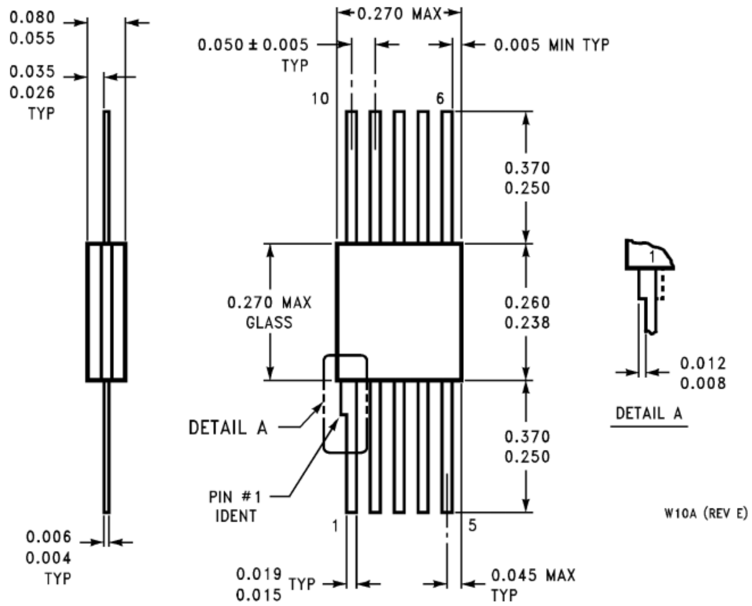


Ceramic Dual-In-Line Package (J)
Order Number LM741J/883
NS Package Number J08A

Physical Dimensions inches (millimeters) unless otherwise noted (Continued)



Dual-In-Line Package (N)
Order Number LM741CN
NS Package Number N08E

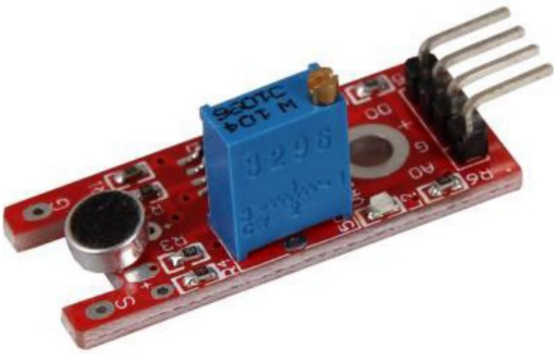


10-Lead Ceramic Flatpak (W)
Order Number LM741W/883, LM741WG-MPR or LM741WG/883
NS Package Number W10A

KY-038 Microphone sound sensor module

Contents

Picture



Technical data / Short description

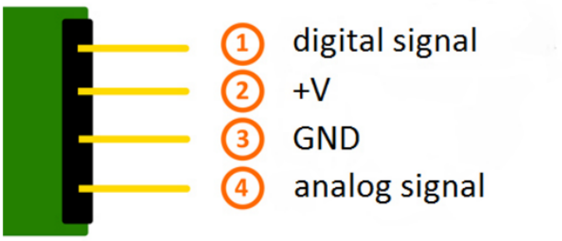
Digital Out: You can use a potentiometer to configure an extreme value for the sonic. IF the value exceeds the extreme value it will send a signal via digital out.

Analog Out: Direct microphone signal as voltage value

LED1: Shows that the sensor is supplied with voltage

LED2: Shows that a magnetic field was detected

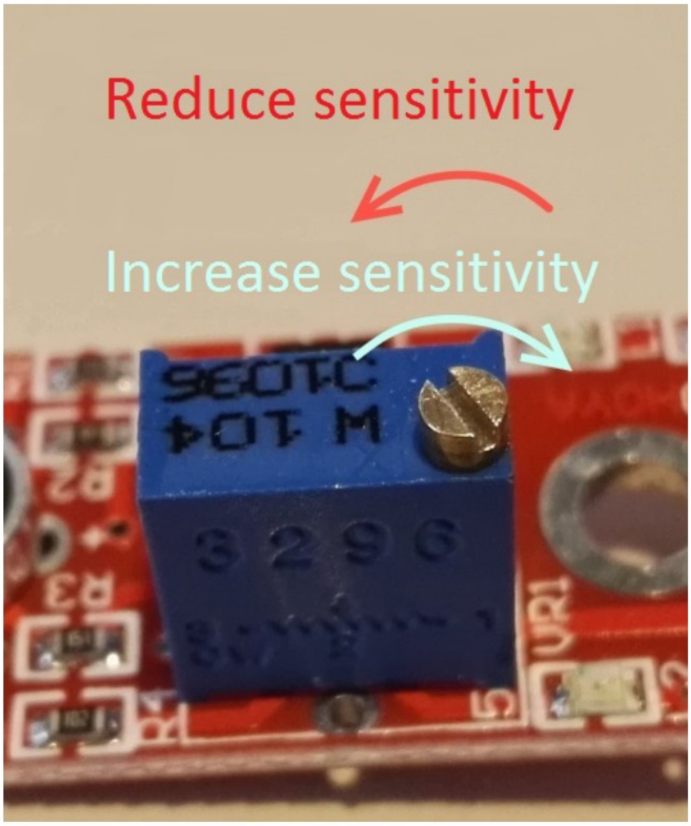
Pinout



Functionality of the sensor

The sensor has 3 main components on its circuit board. First, the sensor unit at the front of the module which measures the area physically and sends an analog signal to the second unit, the amplifier. The amplifier amplifies the signal, according to the resistant value of the potentiometer, and sends the signal to the analog output of the module.
The third component is a comparator which switches the digital out and the LED if the signal falls under a specific value.
You can control the sensitivity by adjusting the potentiometer.

Please notice: The signal will be inverted; that means that if you measure a high value, it is shown as a low voltage value at the analog output.



This sensor doesn't show absolute values (like exact temperature in °C or magneticfield strenght in mT). It is a relative measurement: you define an extreme value to a given normal environment situation and a signal will be send if the measurement exceeds the extreme value.

It is perfect for temperature control (KY-028), proximity switch (KY-024, KY-025, KY-036), detecting alarms (KY-037, KY-038) or rotary encoder (KY-026).