الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

UNIVERSITÉ BADJI MOKHTAR - ANNABA
BADJI MOKHTAR – ANNABA UNIVERSITY

جامعة باجي مختار – عنابـــــــــــة

Faculté : Sciences de L'ingéniorat
Département : Electronique
Domaine : Sciences et Techniques
Filière : Télécommunications
Spécialité : Réseaux et Télécommunications

Mémoire

Présenté en vue de l'obtention du Diplôme de Master

Thème :

An acoustic wireless sensor network for remote monitoring of bird sounds

Présenté par : FERNANE Imed
CHAHI Rachida

Encadrant : *A. BOULMAIZ* *MCB* *UBM ANNABA*

Jury de Soutenance :

D. MESSADEG	Prof	UBM ANNABA	Président
A. BOULMAIZ	MCB	UBM ANNABA	Encadrant
S. NASRI	MCB	UBM ANNABA	Examinateur

Année Universitaire : 2020/2021

ACKNOWLEDGEMENTS

First of all, we would like to express our gratitude to our supervisor Mrs. BOULMAIZ Amira, thank you for been extremely supportive and helpful, you trusted us when the doubt found its way to our hearts, we truly appreciate the enormous amount of time that you gave as, you had lots of responsibilities but you made us one of your priorities, words cannot describe our appreciation enough, may God bless you in each step you step.

We are deeply grateful to all members of the jury Mr. MESSADEG and Mr.Nasri for agreeing to participate in the defense of this thesis. It is a pleasure and a great honor to have you as the jury members.

We are also grateful for all of the teachers of the Electronics department, thank you for the knowledge that you gave as in the past three years, we wish all the best for you all.

We thank also the administration including the head of the department, you were available at all times to solve our problems, the most important that you made an easy environment for studying keep on improving our department, so it will be one of the best departments at Badji Mokhtar university.

God bless you all.

DEDICATIONS

I *am honored to dedicate the fruit of my labor and efforts throughout the years of my study to the two souls dearest to my heart, to those who cultivated a love of science from an early age, my parents and grandparents, may God have mercy on them and put them in peace .*

To my sisters: *Nadia* and *Salima*

To my brothers: *Abd el Hamid* and *Seddik*

To our sister-in-law: *Cherifa*

To my nephew: *Ahmed*

To my nieces: *Dorssaf, Nihed, Mayssa* and our little Princess *Acile*

To my friends: *Zayneb* and *Achwak*

To all my colleagues

To a person to whom I can only express my gratitude and appreciation from my heart for the nobility of his morals and his knowledge**:** *HAMZA*

To all my respected professors, without exception, I dedicate this humble work to you. Thank you, gratitude, and appreciation for all the effort you have made in order to achieve your noble message.

To all who are close to my heart and love me
To Promoting networks and telecommunications 2021

RACHIDA

DEDICATION

I would like to dedicate this thesis to my parents my father Ali and my sweet mother Souad, thank God for blessing me with such parents I love you from the bottom of my heart.

I dedicate this thesis also to the closest person to my heart my grandfather Saiid the one that always believed in me, may God have mercy on him.

I would also like to dedicate this to my closest friends Hichem, Zaki, Fares, khaled and my brother Mohamed and my sister Malek

I am not forgetting my old colleagues Zayneb and Achwak we spent years of Gold at the university together and that makes you my other family.

I dedicate this also to the kindest person I have ever knew, Mr. Hamza, I thank you from the bottom of my heart for giving us your unconditional help, may God give you all the things that you wish in your life.

I would like to dedicate this thesis to my idol, Mm. BRIK it was a pleasure to be your student.

And finally I would like to thank all my colleagues as well; it was a pleasure to study with you and to receive your endless support.

FERNANE IMED

Abstract

Over the past five centuries, more than 150 species of birds have become extinct. According to the latest statistics, more than 1,400 bird species are now threatened with extinction. This represents 13% of all bird species.

The disappearance of bird species disrupts the ecological balance and has a direct negative impact on human beings. It is therefore essential to find a quick solution to protect these fragile species.

The key solution to study birds in their natural habitat is the continuous study using wireless sensor networks (WSN). In this work, we propose a real-time bird sound recognition system composed mainly of sensor nodes and a server.

We are specifically interested in the sensor node part. The node will capture bird sounds using a microphone which will then be processed using the ESP32 microcontroller. On the latter, we propose to implement an algorithm for extracting relevant features from the bird sounds. This algorithm based on the Hilbert envelope is named MHEC for "Mean Hilbert Envelope Coefficients". Once the features are extracted, they will be stored in a storage unit. The classification part will be done at the server level.
The experimental results show the feasibility of the proposed system by combining a remarkable classification rate and a lower implementation cost.

Key words: Real-time monitoring, bird sound recognition, ESP32, MHEC, WSN

ملخص

على مدى القرون الخمسة الماضية، انقرض أكثر من 150 نوعًا من الطيور. ووفقًا لآخر الإحصاءات، هناك أكثر من 1400 نوع من الطيور مهددة بالانقراض حاليا. وهذا يمثل 13٪ من جميع أنواع الطيور.

يؤدي اختفاء أنواع الطيور إلى اختلال التوازن البيئي وله تأثير سلبي مباشر على الإنسان. لذلك من الضروري إيجاد حل سريع لحماية هذه الأنواع الهشة.

الحل الرئيسي لدراسة الطيور في بيئتها الطبيعية هو الدراسة المستمرة باستخدام شبكات الاستشعار اللاسلكية.

في هذا العمل، نقترح نظامًا للتعرف على صوت الطيور في الوقت الحقيقي يتكون من عقد استشعار وخادم.

نحن مهتمون بشكل خاص بجزء عقدة الاستشعار. ستلتقط العقدة أصوات الطيور باستخدام ميكروفون سيتم معالجته بعد ذلك باستخدام متحكم ESP32

في الأخير، نقترح تنفيذ خوارزمية لاستخراج الميزات ذات الصلة من أصوات الطيور. سميت هذه الخوارزمية القائمة على . مغلف هيلبرت.MHEC "معاملات مغلف هيلبرت المتوسطة" وبمجرد استخراج الميزات، سيتم تخزينها في وحدة تخزين

سيتم إجراء جزء التصنيف على مستوى الخادم.

أثبتت النتائج التجريبية قابلية تجسيد النظام بأخذ معدل تصنيف مقبول و كلفة تنفيذ منخفضة بعين الاعتبار.

الكلمات المفتاحية:

المراقبة الآنية, التعرف على أصوات الطيور , ESP32, شبكات الإستشعار اللاسلكية,

MHEC معاملات غلاف هيلبرت المتوسطة

Résumé

Au cours des cinq derniers siècles, plus de 150 espèces d'oiseaux se sont éteintes. Selon les dernières statistiques, plus de 1 400 espèces d'oiseaux sont aujourd'hui menacées d'extinction. Cela représente 13 % de toutes les espèces d'oiseaux.
La disparition des espèces d'oiseaux perturbe l'équilibre écologique et a un impact négatif direct sur les êtres humains. Il est donc essentiel de trouver une solution rapide pour protéger ces espèces fragiles.

La solution clé pour étudier les oiseaux dans leur habitat naturel est l'étude continue à l'aide de réseaux de capteurs sans fil (RCSF). Dans ce travail, nous proposons un système de reconnaissance sonore d'oiseaux en temps réel composé principalement de nœuds de capteurs et d'un serveur.

Nous nous intéressons plus particulièrement à la partie nœud de capteurs. Le nœud capturera les sons d'oiseaux à l'aide d'un microphone qui sera ensuite traité à l'aide du microcontrôleur ESP32. Sur ce dernier, nous proposons d'implémenter un algorithme d'extraction de caractéristiques pertinentes à partir des sons d'oiseaux. Cet algorithme basé sur l'enveloppe de Hilbert est nommé MHEC pour " Mean Hilbert Envelope Coefficients ". Une fois les caractéristiques extraites, elles seront stockées dans une unité de stockage. La partie classification se fera au niveau du serveur.

Les résultats expérimentaux montrent la faisabilité du système proposé en combinant un taux de classification remarquable et un coût de mise en œuvre réduit.

Mots clés : Surveillance en temps réel, reconnaissance de sons d'oiseaux, ESP32, MHEC, RCSF

List of figures

List of Tables

List of Symbols

x	Multiplication
*	Convolution
δ	Magnitude
τ	Filter order
θ	Initial phase
$a(t, j)$	Instantaneous amplitude
$\emptyset(t, j)$	Instantaneous phase
$s(t, j)$	The sub band signal
$\hat{s}(t, j)$	The Hilbert transform
$e_s(t, j)$	The Hilbert envelope
$\approx$	Almost equal
ŋ	Smoothing factor
$\sum_{t=0}^{M-1}$	Summation over discrete index t from 0 to M-1
$S(l, j)$	The sample means

List of Abbreviations

SN	**S**ensor **N**ode
WSN	**W**ireless **S**ensor **N**etworks
WASN	**W**ireless **A**udio **S**ensor **N**etwork
CPU	**C**entral **P**rocessing **U**nit
RF	**R**adio **F**requency
WiFi	**Wi**reless **Fi**delity
EU	**E**uropean **U**nion
LPWAN	**L**ow-**P**ower **W**ide-**A**rea **N**etwork
Wi-SUN	**Wi**reless **S**mart **U**tility **N**etwork
IOT	**I**nternet **O**f **T**hings
LTE-M	**L**ong **T**erm **E**volution for **M**achines
Bluetooth LE	Bluetooth **L**ow **E**nergy
Tiny OS	Tiny **O**perating **S**ystem
MC	**M**icro-**C**ontroller
MCU	**M**icro **C**ontroller **U**nit
ROM	**R**ead **O**nly **M**emory
RAM	**R**andom **A**ccess **M**emory.
Atmel	**A**dvanced **T**echnology for **Me**mory and **L**ogic
Intel	Short form of **Intel**ligence
OS	**O**perating **S**ystem
SOS	**S**ophisticated **O**perating **S**ystem
NesC	**N**etwork **E**mbedded **S**ystems **C**
Solar PV	**P**hotovoltaic (**PV**)

GPS	**G**lobal **P**osition **S**ystem
PCN	**P**ersonal **C**ommunications **N**etwork
PASN	**P**ersonal **A**rea **S**ocial **N**etworking
OMS	**O**nline **M**onitoring **S**ystem
ISDA	**I**mproved **S**moke **D**etection **A**lgorithm
MHEC	**M**ean **H**ilbert **E**nvelope **C**oefficients
IDE	**I**ntegrated **D**evelopment **E**nvironment
LED	**L**ight-**E**mitting **D**iode
VCC	**V**oltage **C**ommon **C**ollector
GPIO	**G**eneral **P**urpose **I**nput/**O**utput
SPI	**S**erial **P**eripheral **I**nterface
I2S	**I**ntegrated **I**nter-**IC** **S**ound
I2C	**I**nter-**I**ntegrated **C**ircuit
ADC	**A**nalog-to-**D**igital **C**onverter.
DAC	**D**igital-to-**A**nalog **C**onverter
UART	**U**niversal **A**synchronous **R**eceiver-**T**ransmitter
SD /MMC	**S**ecure **D**igital (**SD**) **M**ultimedia **C**ard
TWAI	**T**reasury **W**eb **A**pplication **I**nfrastructure
VDDA	**V**ertical **D**isposition and **D**evelopment **A**greement
LNA_IN	**L**ow **N**oise **A**mplifier **IN**
SPIWP	**S**erial **P**eripheral **I**nterface **P**
SPICLK	**S**erial **P**eripheral **I**nterface **C**lock.
DB	**D**ata **B**ase
TRD	**T**onal **R**egion **D**etection
ERB	**E**quivalent **R**ectangular **B**andwidth

AM	**A**mplitude **M**odulation
FM	**F**requency **M**odulation
HTD	**H**ilbert **T**ransform **D**emodulation
TEO	**T**eager–Kaiser **E**nergy **O**perator
Mbps	**M**ega**b**it **p**er **s**econd
A-MPDU	**A**ggregate **M**AC **P**rotocol **D**ata **U**nit
A-MSDU	**A**ggregate **M**AC **S**ervice **D**ata **U**nit
AFH	**A**daptive **F**requency **H**opping
CVSD	**C**ontinuously **V**ariable **S**lope **D**elta

Table of Contents

CHAPTER II: A real-time bird sound recognition system using a low-cost microcontroller

CHAPTER III: Experimental Study

GENERAL INTRODUCTION

13% of the world's bird species are threatened, making them the 3rd most threatened animal species after amphibians and mammals. To this end, monitoring birds in their natural habitat is relevant to identify the species and to assess the progress made in efforts to safeguard biodiversity.

In order to monitor the bird species, several methods had been used such as: nest record, expert observations, wearable sensors, and Wireless Sensor Networks (WSN) …. Etc.

There are enormous numbers of methods for monitoring these birds. We can classify them into two main categories: the methods that relay on human's observation, these methods require the physical presence of the expert in the observation area, however due to many reasons such as hard weather conditions, this may be sometimes impossible. Another disadvantage that makes these methods less effective is that the presence of an individual can disturbs the animal and leads to behavior changes, we call this an active method and this is the last thing we want while monitoring an animal.

In the other hand, we have the methods and technics that rely on the electronic devices are passive methods. These methods are used for remote monitoring as well they do not disturb the monitored animal often, and they are more effective methods by experience. This class contains two popular technics that are the wearable sensors and the WSNs. However, the wearable sensors require certain financial abilities while the WSNs are less expensive.

In our study we use a WASN (Wireless Audio Sensors Network) to monitor the bird species.

In chapter I, we go through the network architecture, available hardware and software platforms and the previous works that used the WSNs for achieving their aims and finally we take a brief look at the challenges.

In the second chapter, we discuss the utilized hardware components, and mention the hardware capabilities that made the used platform suitable for our application, we expose the software utilized in our project as well and we mention the reasons for using such platform.

Also, we illustrate the followed steps for classifying the bird species. After taking all these factors in consideration we try to briefly highlight the weak and strong points in our system.

The last chapter is an evaluation of the system, by taking the hardware qualities and the followed algorithm for extracting what we call an audio prints on which the classification step is based. Finally, we compute the financial cost of our node and the node size as well, and see whether our system is applicable and feasible or not.

CHAPTER I

Wireless Sensor Networks: a Survey on Environmental Monitoring

I.1 INTRODUCTION

Due to various reasons such as global warming, climate change, pollution, species extinction and many other reasons, gathering data about the surrounding environment, which is known as environmental monitoring, is more needed than ever before.

In order to get these data collected and over the years, several methods had been used, including expert observations and even electronic devices and tools, and recently Wireless Sensor Networks are considered among the most effective methods for environmental monitoring, since they are stand-alone systems, low cost and easy to implement, yet monitoring the environment using WSNs still challenging for many reasons, such as processing speed, storage and battery limitations since they rely on low cost micro-controllers.

In the last years, new hardware and software platforms are appearing, for two main reasons, commercial and for improving the performance and getting more satisfying results.

I.2 WIRELESS SENSOR NETWORKS

I.2.1 Definition

Wireless sensor networks (WSNs) are interconnected sensor nodes that communicate wirelessly to collect data about the surrounding environment [1]. Nodes are generally low power and distributed in an ad hoc, decentralized fashion.

I.2.2 WSN architecture

A WSN is made up of set of sensor nodes (SN), These SN are organized into sensor fields (**figure I.1**). Each of these nodes has the ability to collect data and transfer it to the gateway then transmits this data by internet or by satellite to the central computer tasks to analyze it and make decisions [2].

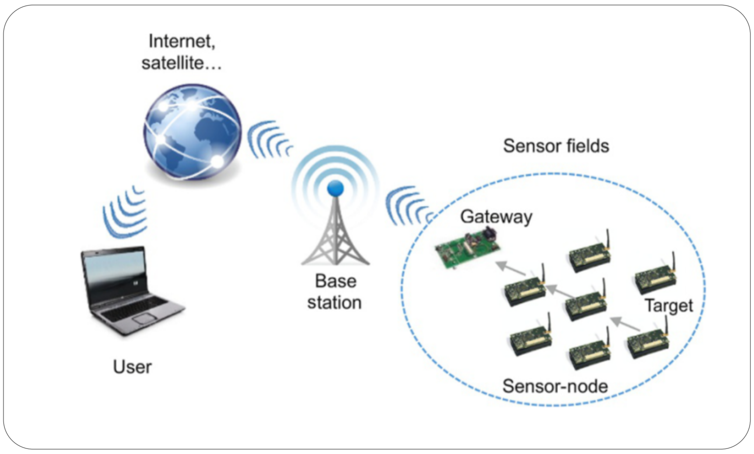

Figure I.1 - WSNs' Architecture

I.2.3 Sensor nodes

The Sensor Node (SN) is the main component in WSN, it captures the physical changes, and converts them into perceivable values, sensor nodes are cheap (low cost), programmable and they have small sizes

I.2.3.1 Senor nodes general architecture

The sensor node is the main part in the WSN, it consists generally of [3]:

- *A Sensing unit* that capture physical changes and converts them into perceivable values.
- *The processing unit* is generally associated with a small storage unit and it can manage the procedures that make the sensor node collaborate with the other nodes to carry out the assigned sensing tasks.
- *A transmission unit* connects the node to the network, there are a variety of options depending on the application the most common are Bluetooth, WIFI, LoRAWAN, etc. You can find more details in the next section.
- *A Power unit*, which is responsible for feeding the node's units with electric energy.

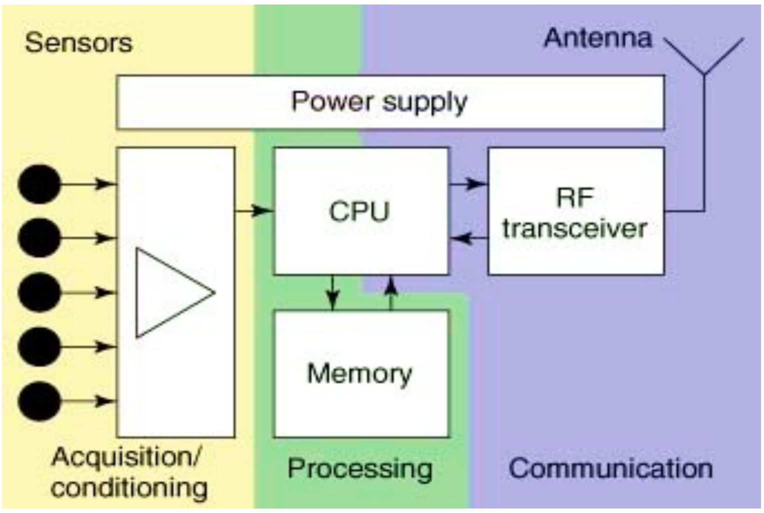

Figure I.2 – SN Architecture [4]

I.2.4 Wireless communication

There are several wireless standards and solutions for sensor node connectivity.

The most common are Bluetooth and WIFI, Bluetooth has a small range (10 meters) this problem is fixed by placing more nodes, the WIFI has a bigger cover range but the energy consumption is greater than the Bluetooth's.

Thread and ZigBee can connect sensors operating at 2.4 GHz with a data rate of 250kbit/s. Many use a lower frequency to increase radio range (typically 1 km), for example Z-wave operates at 915 MHz and in the EU 868 MHz has been widely used but these have a lower data rate (typically 50 kb/s).

The IEEE 802.15.4 working group provides a standard for low power device connectivity and commonly sensors and smart meters use one of these standards for connectivity. With the emergence of Internet of Things, many other proposals have been made to provide sensor connectivity. LORA is a form of LPWAN which provides long range low power wireless connectivity for devices, which has been used in smart meters and other long range sensor applications. Wi-SUN connects devices at home. Narrow Band IOT and LTE-M can connect up to millions of sensors and devices using cellular technology.

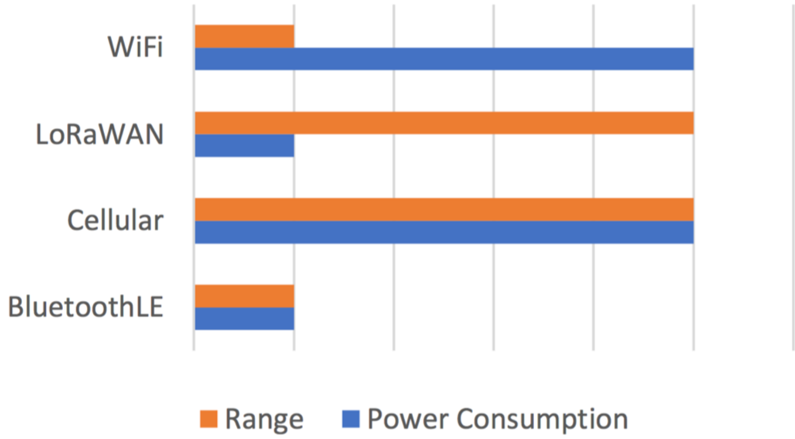

Figure I.3 - Comparison between the most common wireless communication platforms [7].

Figure I.3 compares the 3 platforms in terms of range and power Consumption, since WSNs have a huge number of sensor nodes the cost of each platform is an important factor as well.

I.2.5 Sensor Network Platforms

I.2.5.1 Hardware

The initial research into WSNs was mainly driven by military applications like battlefield reconnaissance and surveillance, nuclear, biological, and chemical attack detection, etc. These projects focused on *ad hoc*, multi-hop WSNs that consisted of thousands of immobile nodes randomly distributed over a large geographical area (e.g., Smart Dust) [4].

The nodes were tiny (hardly noticeable), severely resource constrained, and homogeneous (identical hard- and software). Subsequently, the emergence of civilian applications of WSNs in different fields (environmental monitoring, home automation, health applications, production, inventory, delivery control, etc.) produced a significant diversification of requirements with respect to deployment, mobility, size, cost, network topology, lifetime, etc., and therefore a flourishing of academic and commercial WSN platforms. To cope with these requirements, the platforms increased in size, computational resources, and hardware, as well as in software complexity.

The first commercial platforms appeared in the late 1990s. The most important platform was Crossbow's Rene mote, which emerged from the weC mote developed at the University of California, Berkeley, and which evolved later to the popular Mica platform. These platforms were the precursors of the recent Mica2 and MicaZ platforms (Table I.1). A major reason for the popularity of Crossbow's early mote platforms was their open source policy with both hard- and software design open to the public. This policy built the base for the widespread diffusion of TinyOS as operating system for WSNs. Today, various commercial platforms with different characteristics in terms of computing resources, sensor interfaces, software architecture, etc., are available, which allow to cope with a wide spectrum of civilian applications.

Name		**Tmote**	**Mica2**	**MicaZ**	**Imote2**	**Esp 32**
MCU	Chip manufacturer	Texas	Atmel	Atmel	Intel	Espressif
	Frequency	8Mz	7.38Mz	7.38Mz	13-416Mz	80-240Mz
	ROM	48KB	128KB	128KB	32MB	448 KB
	RAM	10KB	4KB	4KB	32MB	520KB
External memory		1024	512	512		4MB
Power	Supply voltage Min-max (V)	2.1-3.6	2.7-3.3	2.7-3.3	3.2-5	3-3.6
Dimensions	Cm x Cm x Cm	6.6x3.2x0.7	5.8x3.2x0.7	5.8x3.2x0.7	4.8x1.8x1.0	5.4x2.3x0.3

Table I.1 - Selection of Wireless Sensor Network Platforms (Hardware) [4]

I.2.5.2 Software

Unlike general-purpose operating systems for standard PCs such as Windows or Linux, the WSN software platforms are highly tailored to the limited node hardware. These WSN software frameworks are not full-blown operating systems, since they lack a powerful scheduler, memory management, and file system support. However, these frameworks are widely referred to as WSN operating systems. Therefore, this term is retained in the following section [4].

TinyOS, one of the most widespread operating systems, is presented in more detail in the following section. Other operating systems developed for WSNs are Contiki, Mantis, and SOS, Mongoose OS [5].

- ***TinyOS***

TinyOS [6] is written in nesC, an extension to the C language, which supports event-driven component-based programming. The basic concept of component-based programming is to decompose the program into functionally self-contained components. These components interact by exchanging messages through interfaces. The components are event-driven. Events can originate from the environment (a certain sensor reading exceeds a threshold) or from other components, triggering a specific action. The main advantage of this component-based approach is the reusability of components.

The nesC language extension introduces several additional keywords to describe a TinyOS component and its interfaces. nesC and TinyOS are both Open Source projects supported by a fast growing community.

TinyOS has been ported to over a dozen WSN platforms (Table I.1) and is also the native operating system of the presented Tmote platform. It provides a concurrency model and mechanisms for structuring, naming, and linking software components into a robust network embedded system. Today, TinyOS is a sort of de facto standard in WSN programming and widely used in the WSN community. As a result, a huge amount of software components for various sensors, network protocols, algorithms, and other WSN related topics is freely available on the Internet.

I.3 WSN ENVIRONMENTAL MONITORING: PREVIOUS WORKS

I.3.1 PANDI: A Hybrid Open Source Edge-based System for Environmental and Real-Time Passive Acoustic Monitoring -Prototype Design and Development [7]

Improving the accuracy of bird population estimation is crucial for determining their species concern. In Scotland, several species, including the Eurasian Bittern and the Corn Crake, pose challenges as their choice of habitat and behavior makes it extremely difficult for researchers to obtain accurate population numbers. Aims: To identify the targeted species individuals and aid researchers in pinpointing their location. Method a low-cost open-source Edge-based passive acoustic monitoring system was designed (PANDI). Results: The PANDI system design and initial validation are presented in this paper. Future work: Include increasing classification accuracy and scope and commencement of long-term field-trials.

I.2.5.3 Hardware platform

- Roseberry PI for acoustic monitoring (Master)
- ESP32 for monitoring the temperature, humidity and pinpointing birds' location (Slave)
- Power supply: Solar PV Array
- Wireless communication: LoRAWAN[1]

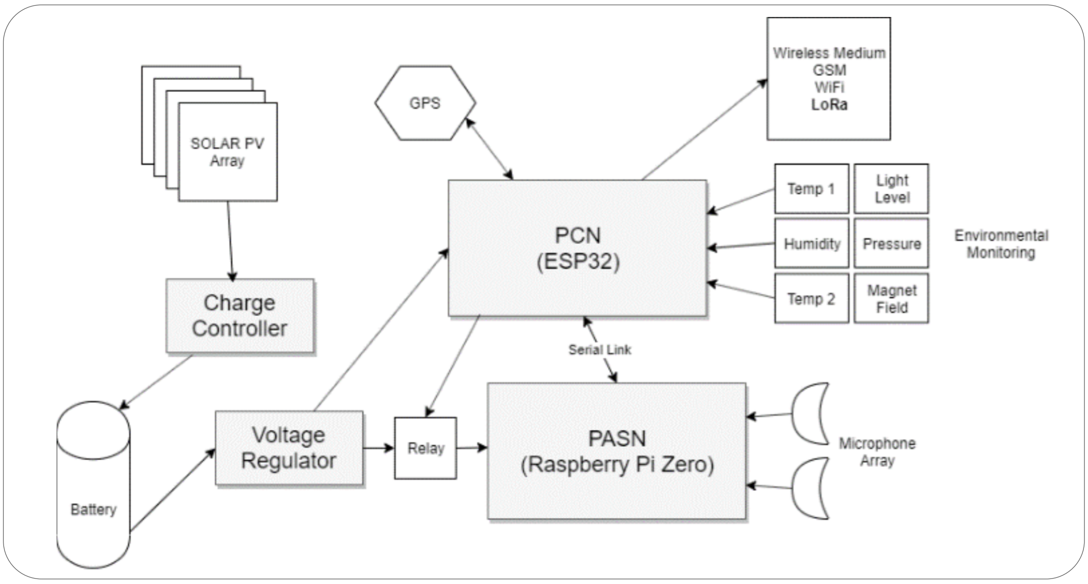

Figure I.4 - Block diagram of the PANDI System [7].

I.3.2 Fusion-Based Volcanic Earthquake Detection and Timing in Wireless Sensor Networks [8]

Volcano monitoring is of great interest to public safety and scientific explorations. However, traditional volcanic instrumentation such as broadband seismometers are expensive, power hungry, bulky, and difficult to install. Wireless sensor networks (WSNs) offer the potential to monitor volcanoes on unprecedented spatial and temporal scales. However, current volcanic WSN systems often yield poor monitoring quality due to the limited sensing capability of low-cost sensors and unpredictable dynamics of volcanic activities. This article proposes a novel quality-driven approach to achieving real-time, distributed, and long-lived volcanic earthquake detection and timing. By employing novel in-network collaborative signal processing algorithms, their approach can meet stringent requirements on sensing quality (i.e., low false alarm/missing rate, short detection delay, and precise earthquake onset time) at low power consumption. They have implemented their

[1] A LPWAN protocol designed to connect battery operated 'things' to the internet in regional, national or global networks.

algorithms in TinyOS and conducted extensive evaluation on a testbed of 24 TelosB motes as well as simulations based on real data traces collected during 5.5 months on an active volcano. they show that their approach yields near-zero false alarm/missing rate, less than one second of detection delay, and millisecond precision earthquake onset time while achieving up to six-fold energy reduction over the current data collection approach.

I.3.3 Design and testing of automated smoke monitoring sensors in vehicles [9]

This article emphasis an advanced method for controlling the pollution of vehicles through sensor integration process. All sensor nodes will be connected to a control center and it will be monitored using Online Monitoring System (OMS) with the help of an Improved Smoke Detection Algorithm (ISDA). The purpose of using ISDA is to monitor the vehicles which are located in different areas under different temperatures where, the ISDA divides the regions into circles. The data obtained in OMS will be aggregated and it will be plotted in MATLAB for better understanding. To test the efficiency of the projected method three test cases which include control of energy, minimization of range and cot have been performed. It is observed that in all three cases ISDA proves to be more efficient in terms of energy consumption, cost and distance when compared with existing methods.

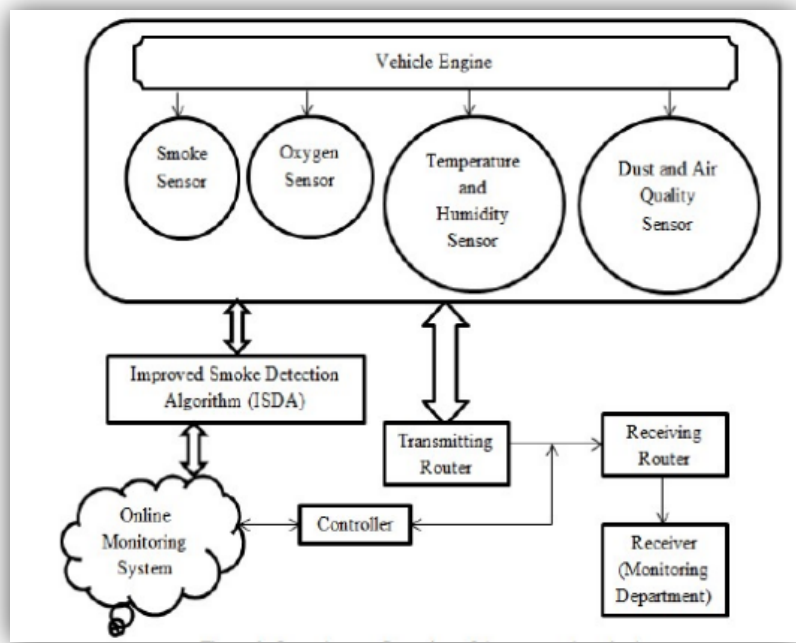

Figure 1.5 - Operation configuration of the proposed method in [9].

I.3.4 Design and Analysis of Wireless Sensor Networks for Animal Tracking in Large Monitoring Polar Regions Using Phase-Type Distributions and Single Sensor Model

Data gathering through wireless sensor networks (WSNs) has been used for the monitoring of endangered species. However, when it comes to animals that live in difficult access environments with large areas, where human access is extremely difficult, such as in Polar Regions, remote monitoring by traditional methods becomes complicated and even inefficient. This paper proposes the characterization of the animal random trajectories by means of the random walk model in order to select the appropriate detection range and number of nodes to guarantee a target detection probability. The animals are detected by static gatherer sensor nodes placed on land either by mobile sensor nodes attached to the animal or by land sensor nodes that detect them through movement, sound, or temperature among other methods. Due to their natural movement, the animal may be outside or inside the sensor nodes coverage radius. In order to reduce energy consumption, it is proposed that nodes be active and inactive, effectively increasing the system lifetime. As such, an inherent compromise between energy consumption and reporting efficiency is present in the design of the network. Building on this, careful network design is required in order to calculate the probability of successful detection and system lifetime. To this end, a mathematical model based on a Markov chain is proposed and developed. The model suitability is assessed via numerical simulations. The obtained results allow concluding that their trajectories can be modeled using specific phase type distributions. Finally, instead of considering that in Polar Regions, it is not feasible to have conventional WSNs, where many nodes are placed together, single nodes are placed in strategic locations isolated among them, and communication between nodes is not possible. As such, it proposes a novel and simplified model, where a single sensor is used to analyze the performance of the complete multi-sensor network.

I.4 CHALLENGES FOR ENVIRONMENTAL SENSOR NETWORKS

Over the last years, due to their benefits WSNs are being implemented in every possible field, but this does not deny the fact that WSN's have many imperfections because of hardware limitations as mentioned before, on top of that there are even external factors as the environmental noise, and this section discuss the common challenges while using WSN for environmental monitoring

I.4.1 Power management

Energy consumption is one of the key challenges in sensor networks, especially in long-term applications such as environmental monitoring. The sensor nodes are usually small, as result of that we are limited by small size batteries

Some of the proposed solutions are: using renewable energy sources [7], event driven sensors [11] which means that the sensor does not work the whole time another solution is using less complex algorithm.

I.4.2 Scalability

SNs differ in scale from some nodes to possibly several numbers. Furthermore, the deployment density is correspondingly adjustable. In the process of gathering data with high resolution, the node density could reach the extent where a node has numerous neighbors in their range of transmission. The protocols positioned in SNs should to be scalable to these extents and should be able to maintain and preserve performance effectively [12].

I.4.3 Remote management

Because researchers cannot regularly visit systems in isolated locations, remote access is necessary to fix bugs, shut down subsystems, change schedules, and so on.

I.4.4 Usability

The WSNs are to be deployed by users who buy them off the shelf. So, the WSN need to become easier to install, maintain and understand. It is necessary to propose new plug and play mechanisms and to develop more software modules with more user-friendly interface and to be practical, environmental sensor networks must primarily consist of off-the-shelf components that are relatively easy to deploy, maintain, and understand—much like the devices in a wireless home network. For example, the average earth scientist could not install the GlacsWeb[2] system because it requires a wide range of computer and electronics technologies with complex interfaces [13].

I.4.5 Standardization

The IEEE 802.15.4 represents a millstone in standardization efforts. Although, compatibility between of-the-shelf modules is in practice very low. It is important to specify standard interfaces to allow interoperability between different modules vendors in order to reduce the costs and to increase the available options. Mesh routing support.

[2] A sensor network for hostile environments

I.4.6 Size

Reducing the size is essential for many applications, in the other hand reducing the node size would limit the battery size and the ability to add more sensing components that will make the node more perfect, and recently micro-controllers (for eg. ESP32) are being produced with smaller sizes and more powerful processors.

I.4.7 Price

Due to several deployment models consider the SNs to be disposable devices, sensor networks could possibly contend with traditional information gathering methods only if the specific SN could be produced economically. The target price intended for a NS should preferably be very low in price [14].

The price depends on the three used platforms mentioned in the platforms section, for example it would be economical if we choose the Bluetooth as the communication unit, but this will force as to place more sensor nodes, and if we take the LoRAWAN platform, as they did in the PANDI System which is mentioned in section I.3, we can place just one SN every 1KM but this platform is very expensive and placing one node every 1 KM will provide more issues.

I.5 CONCLUSION

In this chapter, we have explained why monitoring the environment has been an obligation, and the role of Wireless sensor networks in this field, WSNs are easy to handle and implement, low cost and stand-alone systems.

Due to these advantages, Wireless sensor networks are usually the first option for environmental monitoring, they are not perfect though, WSNs have many hardware and software imperfection and so many challenges which are been solved every year by manufacturing new powerful hardware platforms for improving the performance. It is necessary also to make a compromise between production cost and the used sensor nodes depending on the application.

Next chapter will point the use of WSN in acoustic birds monitoring using ESP 32 as a hardware platform, Arduino IDE as the software platform and last but not least the WIFI as the communication platform.

Chapter II

A real-time bird sound recognition system using a low-cost microcontroller

II.1 INTRODUCTION

In this chapter, we go through the design details of the proposed system, including the utilized hardware and the used software for programming our MC in order to do the tasks that we mention later on the system implementation section

II.2 DESIGN DETAILS OF THE PROPOSED SYSTEM

In this section, we consider the basic design of the proposed system. We handle the hardware and software parts in separate sections, it should be noted that we propose a custom-made sound processing system such that new modules and properties can always be added as hardware allows.

II.2.1 Hardware details

Our real-time bird sound recognition system is composed of two main parts which are the node and the server.

The node itself is decomposed of three modules. The first module is the microphone circuitry responsible for acquiring bird calls. The second module is the micro-controller responsible for computation and processing. The third module is the storing unit responsible for storing the extracted features for the processed bird sound.

The server is responsible for classifying upcoming features using MATLAB classifier and pre-defined database.

II.2.1.1 The microphone circuitry

The acquiring unit in our system consists of two main parts as a mono channel microphone and amplifier. Since the energy consumption has to be minimum. We have chosen the low voltage LM 393 amplifier (**Figure II.1**).

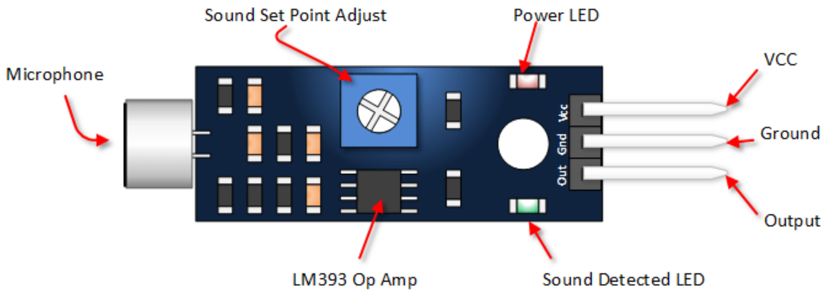

Figure II.1- Microphone Sound Detection Sensor Module LM393

II.2.1.2 The microcontroller

Our system is based on a low-cost and low power ESP 32 (**Figure II.2**). This microcontroller had been chosen for many reasons including his clock speed that can reach 240MHz, it supports a 2.4 GHz Wi-Fi (802.11 b/g/n) with 40 MHz of bandwidth support. The Bluetooth Low Energy. ESP32-S3 has 44 programmable GPIOs, SPI, I2S, I2C, and 12-bit ADC with Sampling frequency that can reach 27.1739 ksps [15], DAC and UART, SD /MMC host and TWAI [16]. The chosen microcontroller has lots of advantages and we have just mentioned those considered as criteria for more details you can check the ESP 32 data sheet [17].

The microcontroller is often the largest part in the node and since the node size is an important factor as mentioned in the first chapter we have chosen a small board with the following dimensions:

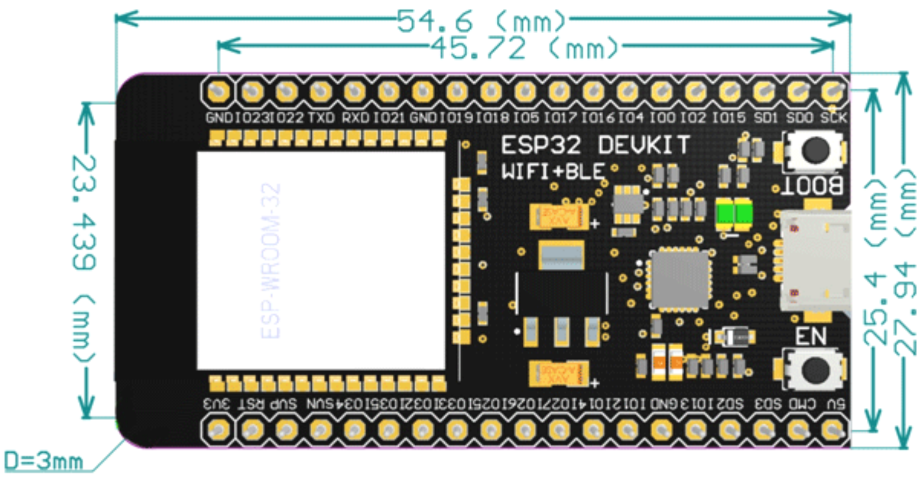

Figure II.2- ESP32-DevKitC-Dimensions

You can also find more details about the pins in **Figure II.3** and the following table (II.1):

Name	NO	Type	Power Domain
VDDA	1	P_A	----
LNA_IN	2	I/O/T	----
VDD3P3	3	P_A	----
VDD3P3	4	P_A	----
GPIO0	5	I/O/T	VDD3P3_RTC_IO
GPIO1	6	I/O/T	VDD3P3_RTC_IO
GPIO2	7	I/O/T	VDD3P3_RTC_IO

GPIO3	8	I/O/T	VDD3P3_RTC_IO
GPIO4	9	I/O/T	VDD3P3_RTC_IO
GPIO5	10	I/O/T	VDD3P3_RTC_IO
GPIO6	11	I/O/T	VDD3P3_RTC_IO
GPIO7	12	I/O/T	VDD3P3_RTC_IO
GPIO8	13	I/O/T	VDD3P3_RTC_IO
GPIO9	14	I/O/T	VDD3P3_RTC_IO
GPIO10	15	I/O/T	VDD3P3_RTC_IO
GPIO11	16	I/O/T	VDD3P3_RTC_IO
GPIO12	17	I/O/T	VDD3P3_RTC_IO
GPIO13	18	I/O/T	VDD3P3_RTC_IO
GPIO14	19	I/O/T	VDD3P3_RTC_IO
VDD3P3_RTC	20	P_A	----
XTAL_32K_P	21	I/O/T	VDD3P3_RTC_IO
XTAL_32K_N	22	I/O/T	VDD3P3_RTC_IO
DAC_1	23	I/O/T	VDD3P3_RTC_IO
DAC_2	24	I/O/T	VDD3P3_RTC_IO
GPIO19	25	I/O/T	VDD3P3_RTC_IO
GPIO20	26	I/O/T	VDD3P3_RTC_IO
VDD3P3_RTC_IO	27	P_D	VDD3P3_RTC_IO
GPIO21	28	I/O/T	VDD3P3_RTC_IO
SPICS1	29	I/O/T	VDD_SPI
VDD_SPI	30	P_D	----
SPIHD	31	I/O/T	VDD_SPI
SPIWP	32	I/O/T	VDD_SPI
SPICS0	33	I/O/T	VDD_SPI
SPICLK	34	I/O/T	VDD_SPI
SPIQ	35	I/O/T	VDD_SPI
SPID	36	I/O/T	VDD_SPI
GPIO33	37	I/O/T	VDD3P3_CPU/VDD_SPI
GPIO34	38	I/O/T	VDD3P3_CPU/VDD_SPI
GPIO35	39	I/O/T	VDD3P3_CPU/VDD_SPI
GPIO36	40	I/O/T	VDD3P3_CPU/VDD_SPI

GPIO37	41	I/O/T	VDD3P3_CPU/VDD_SPI
GPIO38	42	I/O/T	VDD3P3_CPU
MTCK	43	I/O/T	VDD3P3_CPU
MTDO	44	I/O/T	VDD3P3_CPU
VDD3P3_CPU	45	P_D	----
MTDI	46	I/O/T	VDD3P3_CPU
MTMS	47	I/O/T	VDD3P3_CPU
UOTXD	48	I/O/T	VDD3P3_CPU
UORXD	49	I/O/T	VDD3P3_CPU
GPIO45	50	I/O/T	VDD3P3_CPU
VDDA	51	P_A	----
XTAL_N	52	----	----
XTAL_P	53	----	----
VDDA	54	I/O/T	----
GPIO46	55	I	VDD3P3_CPU
CHIP_PU	56	I	VDD3P3_RTC_IO
GRN	57	G	----

Table II.1 – Pins description

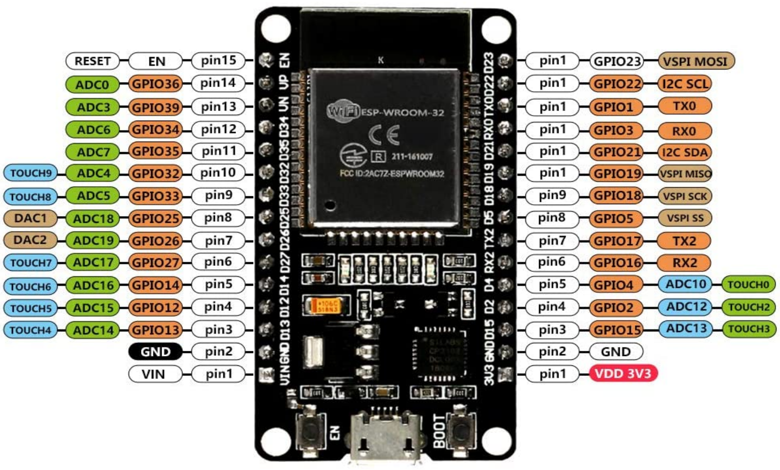

Figure II.3 – ESP 32 pins function

II.2.1.3 The storing unit

After processing the bird sound and getting the features we need to send these features to the server for classifying them, is not guaranteed that they will arrived perfectly at the destination though. Therefore, it is an obligation to add an extra memory; we used for that a 4 GB SD Card, And a SD Card Module.

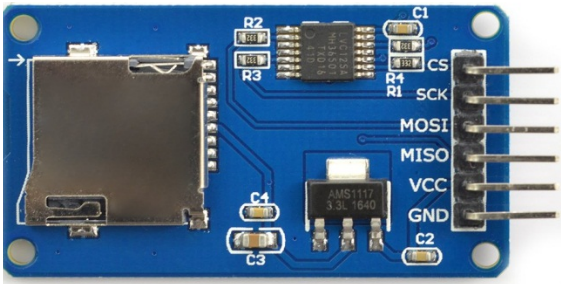

Figure II.4 – SD Card module

II.2.2 Software details

As mentioned in chapter I, our system is composed of a collection of SNs that capture the bird sounds, extract features, store a copy of them and send the extracted features to the server, then the server compares the received features to those who already exist in the DB.

We developed our node using Arduino IDE, which supports a variety of boards and MCs including Arduino and the ESP 32, but first we needed to install the ESP 32 driver in the Arduino IDE, to do so we followed these four steps [18].

- After opening Arduino IDE, we go to File > Preferences
- We have to enter following link: https://dl.espressif.com/dl/package_esp32_index.json into the “Additional Board Manager URLs” field.
- Open the Boards Manager. Go to Tools > Board > Boards Manager…
- Search for ESP32 and press install button for the “ESP32 by Espressif Systems”

We used MATLAB as programming platform for extracting bird sound features for the DB and for classifying detected bird calls. You can find more details about the DB in the third chapter.

II.3 SYSTEM IMPLEMENTATION

We present the block diagram of the proposed system in **Figure II.5**. As the diagram shows our system has six implementation stages. These are sampling and recording the sound signal, removing background noise from the sampled signal, detecting the sound parts of interest in recording, extracting features, store these features and finally classifying the bird species by a minimum distance classifier.

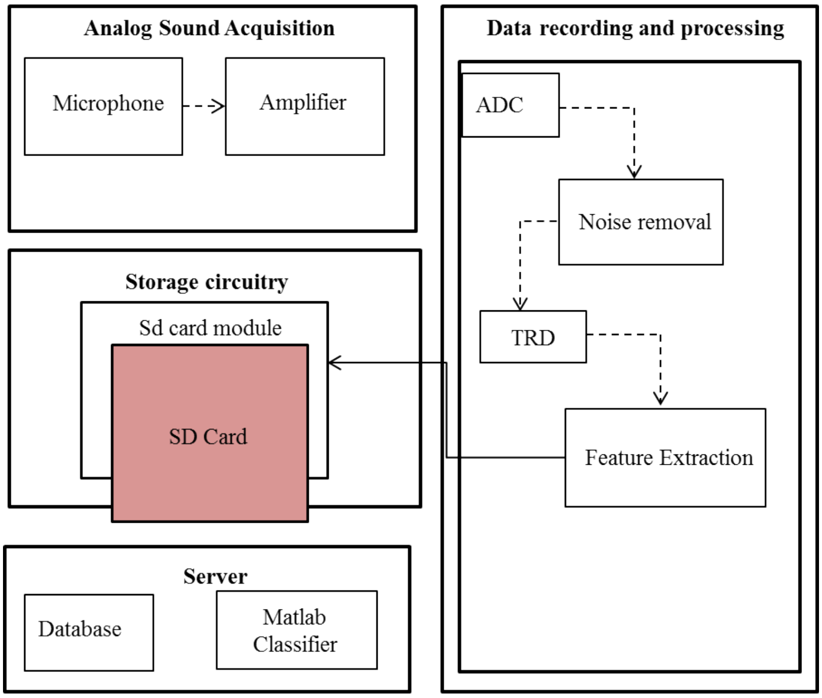

Figure II.5 – Block diagram of the proposed system

II.3.1 Sampling and recording

As the first stage of environmental sound processing, we apply analog signal sampling and recording. We acquire the analog audio signal by the microphone circuitry of our system. We also amplify the signal before feeding it to the microcontroller. The microcontroller samples the analog signal with its 12-bit ADC by a 20480 Hz sampling rate since most of the bird sounds are in frequency range 100 Hz to 10 kHz [19]. ADC of the microcontroller provides 12-bit samples which are stored in memory blocks of 16-bit integers. While processing the audio signal, 16-bit samples are converted to 32-bit floating-point numbers.

II.3.2 Noise removal

As mentioned in the sampling and recording section, the bird sounds are usually contained between 100 Hz and 10 KHz, therefore we use a simple band pass filter for removing sounds that are not included in this range.

II.3.3 Tonal region detection

The third stage of implementation is the detection of sound parts in the recording. There are two reasons to implement this stage. The first one is to be able to focus only on the necessary sound parts in the recording. After noise removal, the recording may contain no-sound and residual noise parts which are not necessary for feature extraction and storage. Therefore, we neglect these parts by detection. The second reason is to be able to observe the beginning and end of meaningful parts in recordings for feature extraction. To do so, we need first to eliminate certain values which are under predefined threshold, which is 0.20 of max peak, 0.20*max was chosen experimentally and works very nicely. The second step is differentiating to measure the power in order to find the sound changes, next we define the max distance between values to be defined inside one bird tweet. After that we define call breaks from max distance, include the beginning of recording. You can find more details in the codes section (extract calls code).

II.3.4 Feature extraction

The fourth stage of the proposed system is feature extraction from the de-noised and detected recordings. Mean Hilbert envelop coefficients was chosen as the feature extraction method. A block diagram illustrating the proposed feature extraction scheme is depicted in **Figure II.6** [20].

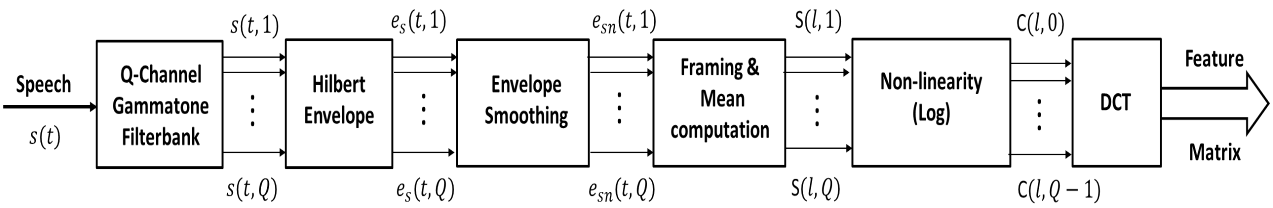

Figure II.6 – Block diagram of MHEC scheme

First, the pre-emphasized audio signal is decomposed into 32 bands through a 32-channel Gammatone filterbank [21]. The filterbank center frequencies are uniformly spaced on an equivalent rectangular bandwidth (ERB) scale between 200 and 3400 Hz (assuming a telephone bandwidth at a sampling rate of Fs = 8 kHz). The ERB for channel j=1; 2; ...; 32 is computed as,

$$\text{ERB}_j = \frac{f_j}{Q_{ear}} + B_{min} \quad (1)$$

Where Q_{ear}= 9.26449 and B_{min}= 24.7 are known as Glasberg and Moore parameters [8], and f_j is the center frequency in Hertz. The frequency response of the 32-channel Gammatone filterbank is illustrated in **Figure II.7**.

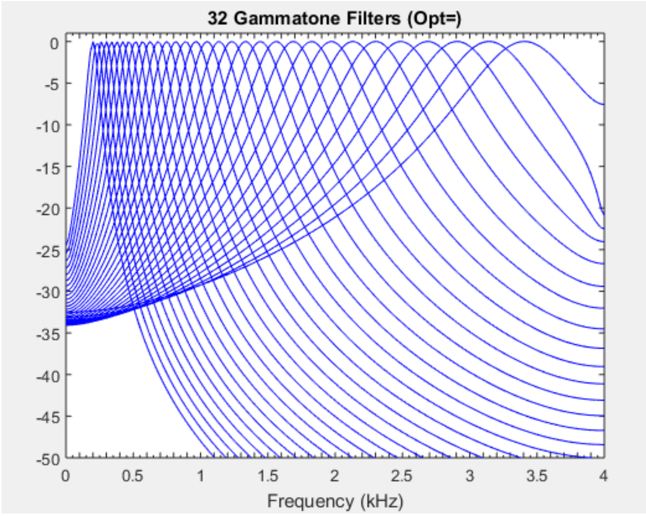

Figure II.7 – Frequency response of the 32-channel Gammatone Filterbank

As seen from the figure, the filterbank essentially consists of a set of band pass filters whose impulse responses are the product of a gamma function and a tone that represent the frequency response associated with a particular point on the basilar membrane of the cochlea. A Gammatone filter is defined by its time-domain impulse response which has a closed-form as [21],

$$h(t,j) = \delta t^{\tau-1} \exp\left(-2\pi b\left(f_j\right)t\right) \cos\left(2\pi f_j t + \theta\right) \tag{2}$$

Where δ and τ denote the magnitude of the response and the filter order, respectively, $b\left(f_j\right)$ is the filter bandwidth with $f_j$being the center frequency in Hz, and θ is the initial phase. The output signal at each channel is obtained as the convolution of the sound signal s(t) with the impulse response at that channel, i.e.

$$s(t,j) = s(t) * h(t,j) \tag{3}$$

The sub band signal $s(t,j)$ is known to have both an amplitude-modulation (AM) and a frequency-modulation (FM) structure as follows [23],

$$s(t,j) = a(t,j). \cos\left[\emptyset(t,j)\right] \tag{4}$$

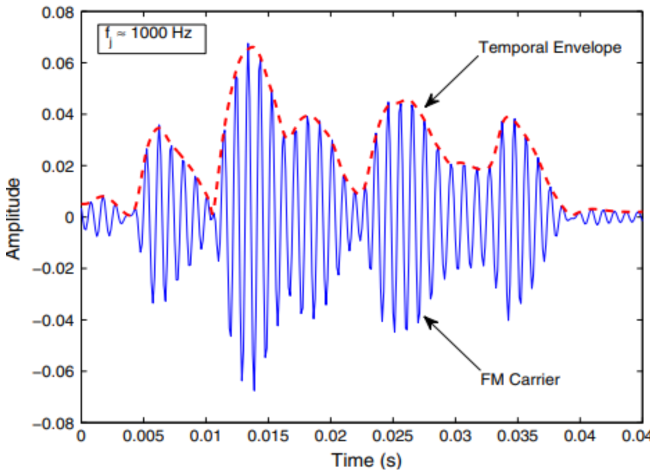

Figure II.8 – A sample sub-band audio signal at a center frequency of f j ≈ 1000 Hz. The instantaneous temporal envelope is shown in red (dashed).

Where $a(t,j)$ and $\emptyset(t,j)$ represent instantaneous amplitude and phase signals at the jth channel, respectively. This is illustrated in **Figure II.8** for a sample sub band audio signal at a center frequency of f j ≈ 1000 Hz, where the slowly varying instantaneous envelope (shown in dashed red) is superimposed over the FM carrier. The temporal envelope of sound is also known as the message which fluctuates at the rate of movements in human speech production apparatus including jaws, tongue, and lips (excluding the vocal folds). The instantaneous frequency of the carrier signal is a function of the vibration speed of vocal folds. Next, since we are primarily interested in slowly varying amplitude modulations $|a(t,j)|$ rather than the fine structure (i.e., instantaneous frequency), $\emptyset(t)$, the temporal envelope of the j^{th} channel output $s(t,j)$ is computed as the squared magnitude of the analytical signal obtained using the Hilbert transform. This approach is known as the Hilbert transform demodulation (HTD) which is employed to separate the AM signal from its FM counterpart. More specifically, let

$$s_a(t,j) = s(t,j) + i\hat{s}(t,j) \quad (5)$$

Denote the analytical signal, where $\hat{s}(t,j)$ is the Hilbert transform of $s(t,j)$, and i is the imaginary unit. The theoretical and computational advantages of the Hilbert transform over other alternative demodulation techniques such as the Teager–Kaiser energy operator (TEO) [23] have been discussed in [24], which describes the problem of AM–FM decomposition from the perspective that AM and FM components of a signal can be derived from the sum of the real signal and its conjugate. In [24], the author defines three physical conditions for the conjugation operator: (1) continuity, (2) homogeneity (i.e., invariance to scaling), and (3) harmonic correspondence (i.e., a single frequency tone must retain its constant amplitude and frequency after the operation, with only a 90-degree shift in phase). It was shown that the Hilbert operator is the only transform that satisfies all three conditions, while the TEO violates the continuity condition. Accordingly, when compared to the TEO, the Hilbert operator is more accurate for noisy signals, and enables elimination of distortion in special cases with wideband noise [25], which is a common case in channel distorted RATS data. Finally, Vakman [24] showed that the Hilbert transform is easier and faster to carry out regarding the computer processing.

The temporal envelope $e_s(t,j)$ is thus calculated as,

$$e_s(t,j) = s^2(t,j) + \hat{s}^2(t,j)\ . \tag{6}$$

With:

$$|a(t,j)|^2 \approx e_s(t,j). \tag{7}$$

Here, $e_s(t,j)$ is also called the Hilbert envelope of the signal $s(t,j)$. At the next stage, in order to further suppress the remaining redundant high frequency components, thereby reducing the approximation error in (7), the Hilbert envelope $e_s(t,j)$ is smoothed using a low-pass filter with a cut-off frequency of fc = 20 Hz as,

$$e_{sn}(t,j) = (1 - ŋ\ )e_s(t,j) + ŋe_{sn}(t-1,j) \tag{8}$$

Where the subscript n denotes the smoothed (or normalized) envelope, and ŋ is a smoothing factor (inversely) exponentially proportional to the cut-off frequency as,

$$ŋ = \exp\left(-\frac{2\pi f_c}{F_s}\right) \tag{9}$$

As noted earlier, the smoothed sub-band Hilbert envelopes which represent the envelope of the auditory nerve response, carry useful acoustic information [26, 27] and have been shown to be robust to signal degradations due to background noise and communication channel for different speech applications [21, 28, 29, 30]. Additionally, it is well-known that the sub-band AM signals at center frequencies above 1 kHz fluctuate at fundamental frequency (i.e., pitch) of sound [31]. Our hypothesis is that this property can help as an acoustic cue to better discriminate among different bird species.

At the next stage, the smoothed Hilbert envelope $e_{sn}(t,j)$ is blocked into frames of 25 ms duration with a skip rate of 10 ms. A Hamming window is applied to each frame to minimize discontinuities at the edges as well as the correlation between adjacent frames. To estimate the temporal envelope amplitude in frame l, the sample means are computed as,

$$S(l,j) = \frac{1}{M}\sum_{t=0}^{M-1} w(t).\, e_{sn}(t,j) \tag{10}$$

Where $w(t)$ denotes the Hamming window and M is the frame size in samples. Note that $S(l,j)$ is also a measure of the spectral modulation energy at the center frequency of the jth channel, and therefore provides a short-term spectral representation of the speech signal $s(t)$.

To compress the dynamic range of the estimated spectral parameters, $S(l,j)$, a nonlinear operator such as the natural logarithm [32] or root [33] is commonly applied in the extraction of cepstral features. It has been known in the ASR community that the logarithmic spectrum compression adopted in conventional cepstral analysis is susceptible to noise.

The spectral parameters $S(l,j)$ are either compressed using the natural logarithm or power-law nonlinearity with an exponent term ɤ as,

$$C(l,j) = |S(l,j)|^{ɤ} \tag{11}$$

Where ɤ is set to 1=15 as suggested in [34]. The motivation behind employing a nonlinear function as in (11), is twofold. First, the root compression with an exponent of ɤ = 1/15 provides a better fit to the psychophysical data observed from subjective experiments (i.e., involving human listeners), particularly for sound pressure levels below 0 dB SPL. In other words, it approximates more accurately the nonlinear relation between the input sound intensity and auditory nerve firing rate. Second, for frequency channels with energies much smaller than one, that is $S(l,j) \ll 1$, the compressed output has a value close to zero as opposed to negative infinity. Note that the logarithmic compression artificially increases the variance for such spectral components, resulting in a greater variability of estimated cepstral features which has been shown to be a major source of performance loss for speech applications.

After the nonlinearity stage, which uses either the log or root compression, the discrete cosine transform (DCT) is applied as,

$$c(l,q) = \sum_{j=0}^{Q-1} C(l,j).\, cos\left[\frac{\pi q}{2N}(2j+1)\right], \qquad q = 0, \ldots\ldots, 31. \tag{12}$$

The DCT is used to: (1) convert the spectrum to the cepstrum, and (2) decorrelate the various feature dimensions because, as shown in **Figure II.7**, there is significant overlap between adjacent filters. The latter is important because Gaussian mixture models (GMM) with diagonal covariance matrices of reasonable size can then be used to model the acoustic spaces of speakers (as opposed to full covariance matrices). A potential shortfall in using the DCT, which becomes apparent after a careful examination of eq. (12), is that it spreads any local distortion (due to background noise and/or transmission channel) across all resulting cepstral coefficients. This becomes even more problematic if the nonlinearity stage introduces undesirable and artificial variabilities as well (e.g., very large negative values for very small spectral energies after the log compression). This constitutes another important reason for adopting a nonlinear operator that is less susceptible to processing and environmental artifacts.

The output of the DCT stage is a matrix of 32-dimensional cepstral features$\boldsymbol{C}(\boldsymbol{l},\boldsymbol{j})$, entitled the mean Hilbert envelope coefficients (MHEC).

MHEC had achieved great results in speech recognition applications, and since the human sound production and the bird sound production are very similar, we propose to use this method for our application on bird sound recognition.

II.3.5 Storage

The fifth stage of the proposed system is for storing Extracted features on the SD card which communicates with the microcontroller by the SPI protocol. We use the Quad Synchronous Serial Interface (QSSI) module on the microcontroller as bi-SSI that serves SPI communication. FatFS library allows us to write the obtained features to SD card.

II.3.6 Classification

The sixth stage of the proposed system is the classification of bird species in the recordings. We use the minimum distance classifier for classification since it can be implemented easily. The training and cross-validation parts of the classification are carried out apart from the actual system implementation as off-line processes. When the sound is produced based on non-bird activity, corresponding MHEC features might be located far away from the MHEC feature vectors of bird species. For such conditions, we also set a distance threshold. Hence, if the distance between the received MHEC vector and MHEC feature vectors of the bird calls are greater than the threshold, the classifier concludes a non-bird activity. The MHEC feature vector of bird species and distance threshold are buffered in an array in the microcontroller, in order to classify the test samples.
Then the classification results are easily stored in the server, since the space is unlimited compared to the size of the features and the database as well.

II.4 Conclusion

In this chapter, we proposed a real-time bird sound recognition system consisting of a node and a server. The node consists of a microphone, microcontroller and data storage unit. The chosen microcontroller can handle the data recording and preliminary onboard signal processing, feature extraction and classification simultaneously. But as mentioned in the first chapter the network lifetime is a challenge itself, therefore our system is decomposed in two parts, the server handle the classification task, while the node handle the other tasks. And by doing this we limit the node processing tasks in order to extend its lifetime.

In the advanced parts of this chapter we discussed the processing tasks, the feature extraction section took the largest space of our work, in this section we have talked about the MHEC method, in our study we have just used the main steps of MHEC, this method can be more perfect if we add more steps according to our needs, in the other hand adding more tasks to the node will limit his lifetime, which is not amiable.

CHAPTER III

Experimental Study

III.1 DATASET

As mentioned before in chapter 2 our system is composed of two main entities the node and the server. The server classifies the bird species using the MATLAB classifier and pre-defined Database.

In our study we used 20 samples of 5 different species (**figure III.1-5**) which makes it 100 samples in total, in order to create the DB, we went though many steps, first we downloaded the audio samples from Xeno-Canto mp3 Format we removed the noise, detect the sound parts of interest and finally extracted the MHECs, and we used Matlab for doing these tasks.

It is important to understand that the Database is not composed of sounds the final database is composed of vectors each represents the audio print of the signal (**figure III.10 – 12**).

Figure III.1 – Flamingo (one the chosen species)

Figure III.3 – Great Bittern

Figure III.2 – Pied Avocet

Figure III.4 – Northern Shoveler

Figure III.5 – Gadwall

III.2 PERFORMANCE

In this section we evaluate our project from many perspectives by trying to answer the following questions:

- Is our hardware able to do the pre-mentioned tasks in the second chapter smoothly and without exhausting its processor so it will not over consume energy?
- Is the node size small enough so it will not disturb the studied birds?
- Is the financial cost of the node economical enough so we can offer the financial cost of the whole network?
- And the last question is about the system accuracy.

III.2.1 Hardware Performance

As known before the node is composed of a single channel microphone with an integrated amplifier, this give us the advantage of not adding an external amplifier because that will make the node even bigger, the second component is the micro-controller which is the ESP 32,this MC has an acceptable size for such applications and will not affect the final node size negatively, the ESP32 has two micro-processors the main one has 240MHz as a processing speed while the other processor is designed for smaller applications and that will be helpful if we want to add smaller tasks and will barley consume energy.

Briefly our system captures the bird sounds, converts the analog captured sound, removes the noise, extracted MHECs, stores them in a SD Card and finally send them to the server when the classification task will be done. These tasks will be easily done with the 240 MHz processing power, but first we need to sample the analog signal with sampling frequency Fs that is equal or twice greater than the greatest analog signal frequency (**Shannon**), the ESP 32 has the I2S interface that can be configured to operate with an 8-/16-/24-/32-bit resolution as an input or output channel, with sampling frequency that can reach 192Khz, which is more than enough for our application.

Unfortunately, the ESP 32 has just 320KB as flash memory, therefor we need and external SD card to store the extracted features.

III.2.2 Software performance

III.2.2.1 Noise removal

As mentioned in the second chapter we used a simple band pass filter that blocks the frequencies that are not included in the range 100Hz -10 000Hz. By doing this we cannot insure that we removed the noise since there are other sounds in the nature that are included in this range as well.

Here is an example for removing noise using this simple method:

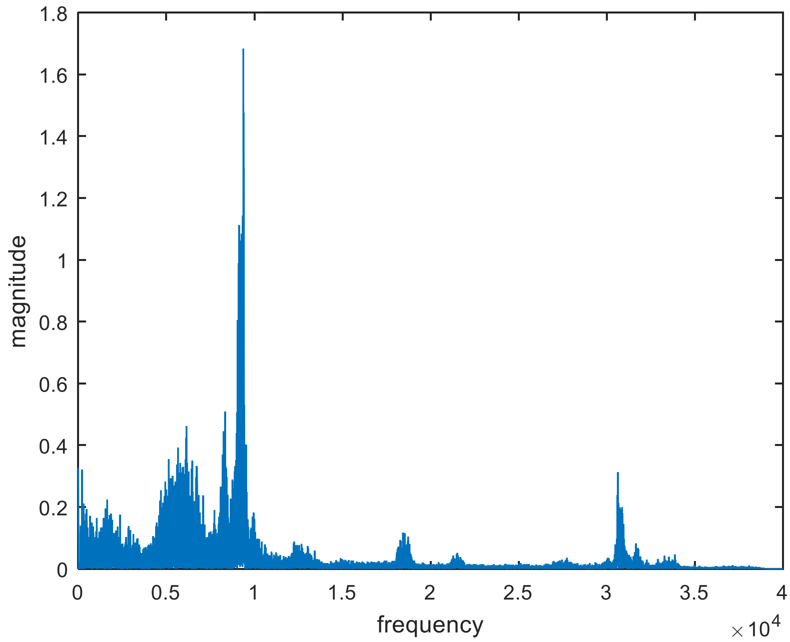

Figure III.6 – The spectrum of a noisy bird sound

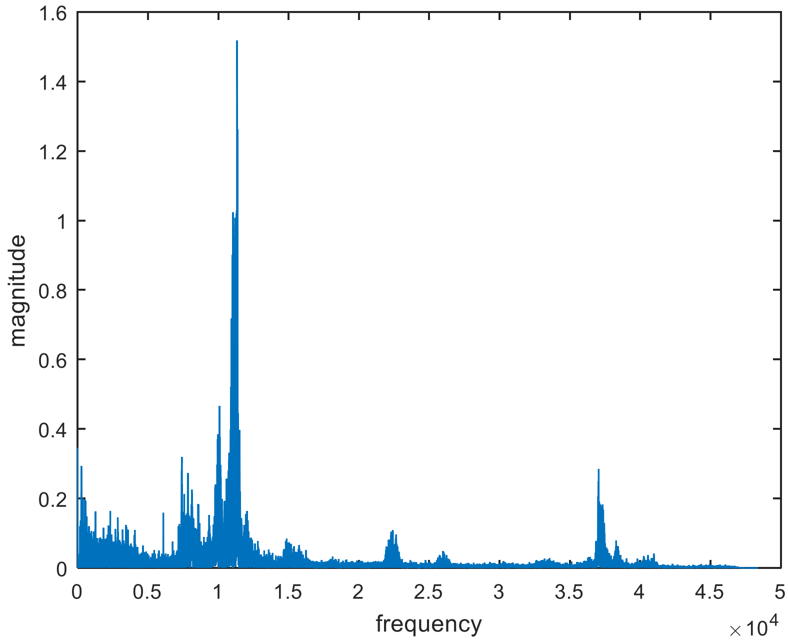

Figure III.7 – The spectrum of filter bird sound

III.2.2.2 Tonal region detection TRD

We have evaluated this part of the system manually, and the results were impressive, you can find the MATLAB code in the codes section and more details about it in the second chapter.

- The following figures represent the bird sound before and after extracting the sound parts of interest and removing the silent moments:

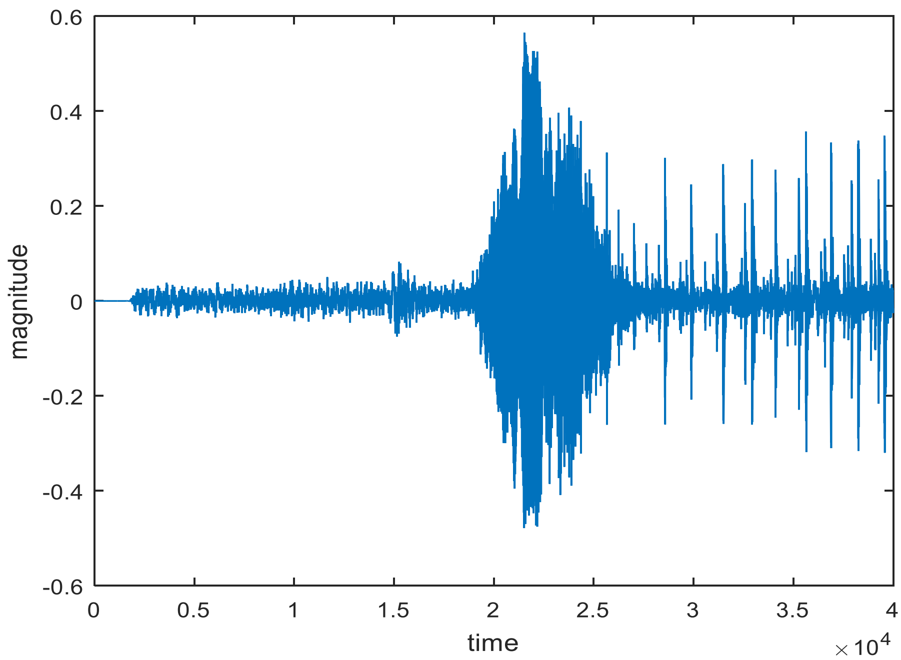

Figure III.8 – The waveform of noisy bird sound before removing the silence

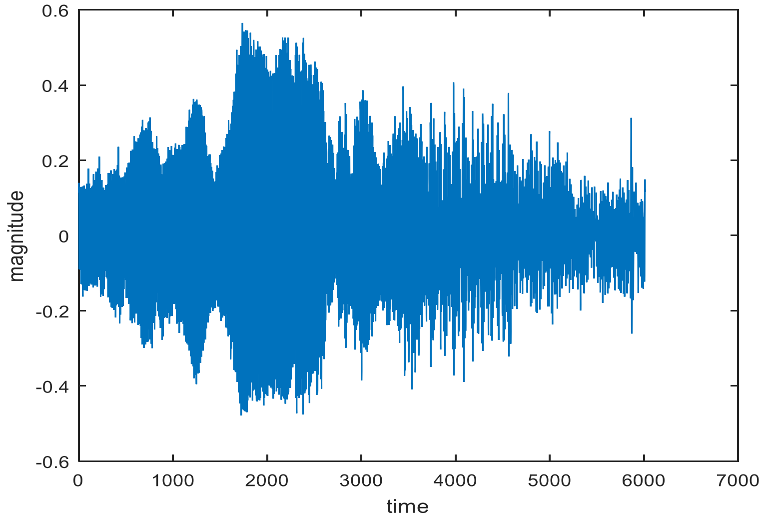

Figure III.9 - The waveform of noisy bird sound after removing the silence

III.2.2.3 Features extraction

Our system focus more on the feature extraction step we have used the MHEC that proved its quality in SID applications. We cannot ensure the work of the mentioned method till we put it into action though.

We have extracted the MHECs using MATLAB **figures III.10 – 12** represent the extracted MHECs for the chosen bird species

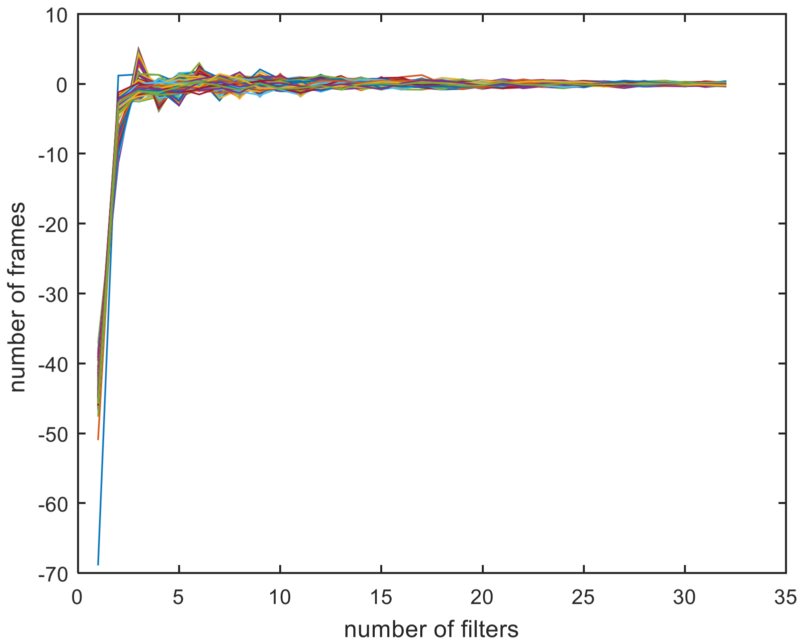

Figure III.10 – MHECs extracted from an audio sample of Pied Avocet

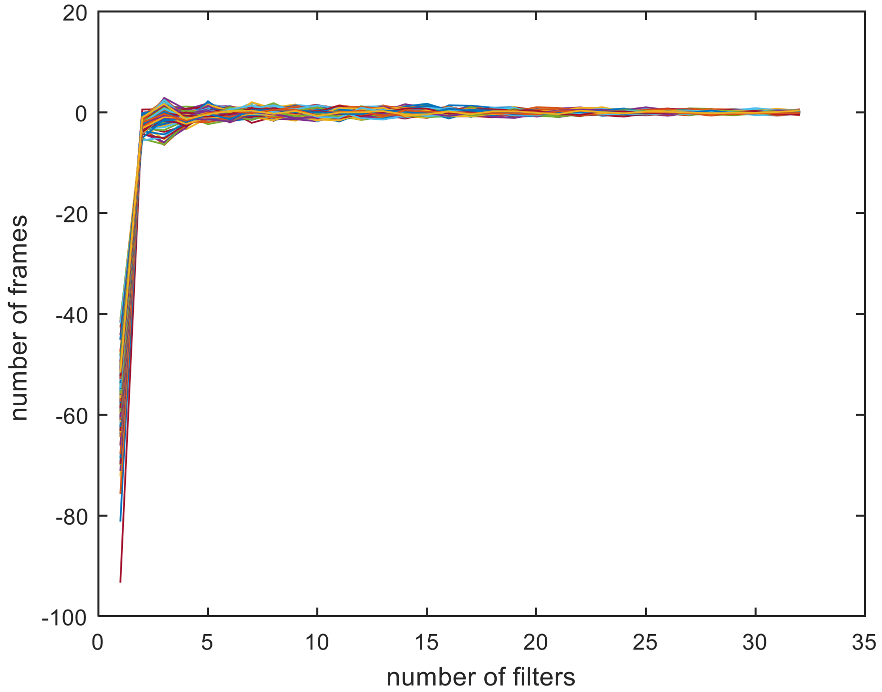

Figure III.11 – MHECs extracted from an audio sample of Great Bittern.

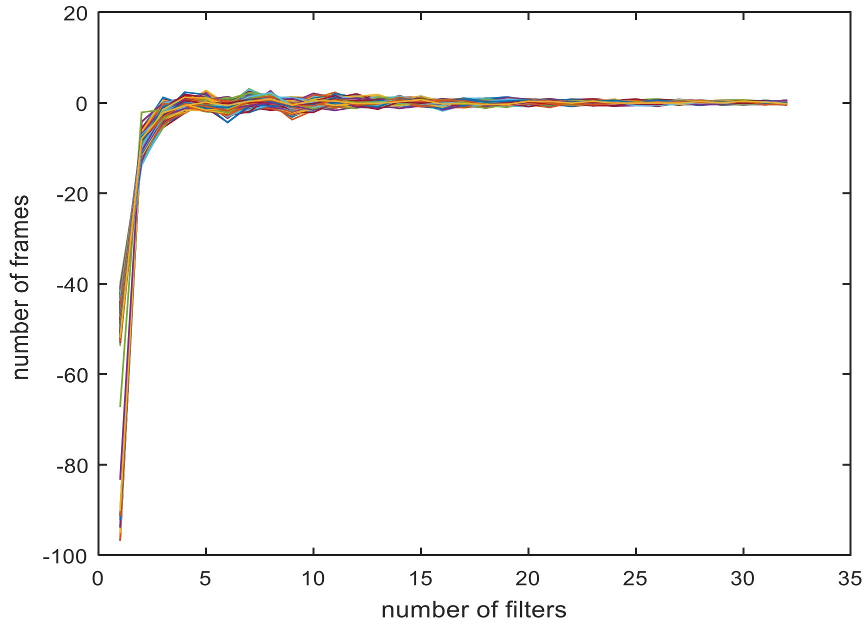

Figure III.12 – MHECs extracted from an audio sample of Flamingo bird.

The three figures above (**Figure III.10, 11, 12**) shows the extracted features (vector of coefficients) for three different bird species, it is clear that every bird have a different vector, we call these video the audio print.

III.2.2.4 Classification

This part of the system is very simple, easy to understand and effective as well, the accuracy of the classification results is going to determine whether our system is effective or not. The thing though is that the classification depends on the other steps, in this part of the system we use the minimum distance classifier which is easy to implement

In order to evaluate the work of our system under noisy conditions we have added an additive Gaussian noise to the database, and the table III.1 shows the classification rate with different SNRs.

SNR (dB)	Classification Rate
30	93,28%
25	88,54%
20	86,21%
15	84,98%
10	83,43%
5	74,62%

Table III.1 – Classification rate under noisy conditions

III.2.3 Financial cost

Usually a WSN is composed of hundreds if not thousands of SNs that is why the cost of each node is an important factor, otherwise it will not be realistic. In this section we are going to find out the final cost of each component and finally the cost of the whole sensor node:

- The microphone: 1€
- The micro-controller (ESP 32): 3,38€ + 2,48€ (shipping fee)
- The SD card module: 1€
- The SD card: 2€

After taking a look at the price of each component the final price of the whole node would be the sum: 1+1+2+5.86=9.86 €.

III.2.4 Node size

Our node size is optimal; we did not use big components the ESP32 which is considered as the largest component is very small (look at the chapter 2 for more details) so it will not disturb any bird. Otherwise the node will act like a scarecrow.

III.3 Conclusion

In this chapter we have analyzed our system, we found the weak points in the system, and the advantages as well. The most important advantage in our study is knowing the imperfections of our system which is necessary in order to improve the node. We have also answered the questions asked before in this chapter, by doing this we knew that the strongest points of our system are: the node size, the node cost and micro-controller abilities which is impressive considering its small size and cost as well. As any other project ours has many imperfections which are the noise removal part that depends on the individual knowledge in the ST and programing fields.

General conclusion

The aim of our thesis was to develop a system for bird species identification that has the less possible imperfection, so it will identify the birds precisely and effectively.

Therefore we gave a brief introduction to previous works that used the WSNs in different fields in the first chapter, and by doing this we observe the possible limitations for the WSNs, and the possible solutions as well, we have also took a look at the popular challenges, so we can handle them in our work whenever they appear, in the second chapter we went deeply into the details of our system which is composed as mentioned before of two entities, the server and the node, we have observed that many works, put a pressure on the node by giving it an enormous number of tasks which effects the node lifetime and exhaust it over time, one of the solutions that came in mind was to do the classification task in a server so we can extend the node lifetime by giving the it less tasks, but sending the extracted features, every time we process a sound signal, means that we will overuse the communication unit which consume a huge amount of energy, and the lifetime problem will even be worse.

For avoiding that we store the features in a SD Card and send them once a day to the server, the SD card were also necessary in case we lose the features while sending them. Another challenge was the financial cost, we have observed that in the PANDI system (previous works section) uses the Rasebery Pi, ESP32 and LORA as the main components which is not realistic because that will extend to cost of the node which will extend the network cost, another problem was the node size, using two micro-controllers will make it even bigger and will disturb the studied birds eventually, we have manage to solve these two problems in once by using the ESP32 as the processing unit, it has an integrated WIFI, Bluetooth and BLE and that will help us in financially and for reducing the node size as possible as we can.

Our node will not work as the PANDI system node but coming out with a compromise is an obligation, otherwise a single problem that may be the size, price, lifetime will ruin the whole system.

BIBLIOGRAPHY

[1]. Harsh Kupwade Patil, Thomas M. Chen, in Computer and Information Security Handbook (Third Edition), 2017.

[2]. Boulmaiz Amira, Processing and Automatic Recognition of bird calls and songs in wet zones Annaba-Taref, 2017.

[3]. M.A. Matin, M.M. Islam, Overview of Wireless Sensor Network, INTECH Open Access Publisher, 2012.

[4]. Reinhard Bischoff, Jonas Meyer, Glauco Feltrin, Wireless Sensor Network Platforms, 2009.

[5]. Website: https://www.espressif.com/en/tags/rtos, 2021

[6]. Levis, P., Madden, S., Polastre, J., Szewczyk, R., Whitehouse, K., Woo, A., ... & Culler, D. (2005). TinyOS: An operating system for sensor networks. In Ambient intelligence (pp. 115-148). Springer, Berlin, Heidelberg.

[7]. Sean Sturley, Santiago Matalonga, PANDI: A Hybrid Open Source Edge-based System for Environmental and Real-Time Passive Acoustic Monitoring -Prototype Design and Development, March 2020.

[8]. Rui Tan, Guoliang Xing, Jinzhu Chen, Wen-Zhan Song, Renjie Huang, Fusion-Based Volcanic Earthquake Detection and Timing in Wireless Sensor Networks, April 2013.

[9]. Pravin Kshirsagar, Hariprasath Manoharan, Kailash Kumar, Fadi Al-Turjman, DESIGN AND TESTING OF AUTOMATED SMOKE MONITORING SENSORS IN VEHICLES, March 2021.

[10]. Rodolfo Vera-Amaro, Mario E. Rivero Angeles, Alberto Luviano-Juarez, Design and Analysis of Wireless Sensor Networks for Animal Tracking in Large Monitoring Polar Regions Using Phase-Type Distributions and Single Sensor Model, Publisher: IEEE, March 2019.

[11]. Shih-Chii Liu, Bodo Rueckauer, Enea Ceolini, Adrian Huber, Tobi Delbruck, Event-Driven Sensing for Efficient Perception: Vision and Audition Algorithms, October 2019.

[12]. M.A. Matin, M.M. Islam, Overview of Wireless Sensor Network, September 2012.

[13]. Luís M. L. Oliveira, Joel J. P. C. Rodrigues, Wireless Sensor Networks: a Survey on Environmental Monitoring, OURNAL OF COMMUNICATIONS, VOL. 6, NO. 2, APRIL 2011.

[14].Kingsley Eghonghon Ukhurebor, Ituabhor Odesanya, Silas Soo Tyokighir, Rout George Kerry, Akinola Samson Olayinka and Ayodotun Oluwafemi Bobadoye, Wireless Sensor Networks: Applications and Challenges, October 2020.

[15]. labfruits, "ESP32 ADC Speed", www.labfruits.com. [Online].

[16].Espressif Systems, "ESP32-S2 Technical Reference Manual," espressif.com, 2019. [Online]. Available: https://www.espressif.com/en/support/documents/technical-documents

[17].Espressif Systems, "ESP32-S2 Technical Reference Manual," espressif.com, 2019. [Online]. Available: https://www.espressif.com/en/support/documents/Datasheet

[18].randomnerdtutorials : https://randomnerdtutorials.com/installing-the-esp32-board-in-arduino-ide-windows-instructions/

[19].Kwan C, Ho K, Mei G, Li Y, Ren Z, et al. An automated acoustic systemto monitor and classify birds. EURASIP J Appl Sig Process 2006;2006. 52-52.

[20].Seyed Omid Sadjadi, John H.L. Hansen, Mean Hilbert envelope coefficients (MHEC) for robust speaker and language identification, 2015 page: 3.

[21].Patterson, R.D., Robinson, K., Holdsworth, J., McKeown, D., Zhang, C., Allerhand, M., 1992. Complex sounds and auditory images. In: Cazals, Y., Demany, L., Horner, K. (Eds.), Auditory Physiology and Perception. Pergamon Press, Oxford, pp. 429–446.

[22].Glasberg, B.R., Moore, B.C., 1990. Derivation of auditory filter shapes from notched-noise data. Hearing Res. 47 (12), 103–138.

[23].Maragos, P., Kaiser, J.F., Quatieri, T.F., 1993. Energy separation in signal modulations with application to speech analysis. IEEE Trans. Signal Process. 41 (10), 3024–3051.

[24].Vakman, D., 1994. On the analytic signal, the Teager–Kaiser energy algorithm, and other methods for defining amplitude and frequency. IEEE Trans. Signal Process. 44 (4), 791–797.

[25].Schimmel, S.M., 2007. Theory of Modulation Frequency Analysis and Modulation Filtering, with Applications to Hearing Devices. Ph.D. Thesis, Dept. Elect. Eng., University of Washington.

[26].Drullman, R., Festen, J.M., Plomp, R., 1994. Effect of reducing slow temporal modulations on speech reception. J. Acoust. Soc. Am. 95 (5), 2670–2680.

[27].Assmann, P., Summerfield, A., 2004. The perception of speech under adverse conditions. In: Greenberg, S., Ainsworth, W.A., Popper, A.N., Fay, R.R. (Eds.), Speech Processing in the Auditory System. SpringerVerlag, New York.

[28].Kingsbury, B.E., Morgan, N., Greenberg, S., 1998. Robust speech recognition using the modulation spectrogram. Speech Commun. 25 (13), 117–132.

[29].Lawson, A., McLaren, M., Lei, Y., Mitra, V., Scheffer, N., Ferrer, L., Graciarena, M., 2013. Improving language identification robustness to highly channel-degraded speech through multiple system fusion. In: Proc. INTERSPEECH. Lyon, France, pp. 1507–1510

[30].Mitra, V., McLaren, M., Franco, H., Graciarena, M., Scheffer, N., 2013. Modulation features for noise robust speaker identification. In: Proc. INTERSPEECH. Lyon, France, pp. 3703–3707.

[31].Hu, G., Wang, D., 2004. Monaural speech segregation based on pitch tracking and amplitude modulation. IEEE Trans. Neural Networks 15 (5), 1135–1150.

[32].Weintraub, M., 1985. A Theory and Computational Model of Auditory Monaural Sound Separation. Ph.D. Thesis, Dept. Elect. Eng., Stanford University

[33].INTERSPEECH. Pittsburgh, PA, pp. 1471–1474. Hermansky, H., 1990. Perceptual linear predictive (PLP) analysis of speech. J. Acoust. Soc. Am. 87 (4), 1738–1752.

[34].Kim, C., Stern, R.M., 2012. Power-normalized cepstral coefficients (PNCC) for robust speech recognition. In: Proc. IEEE ICASSP. Kyoto, Japan, pp. 4101–4104