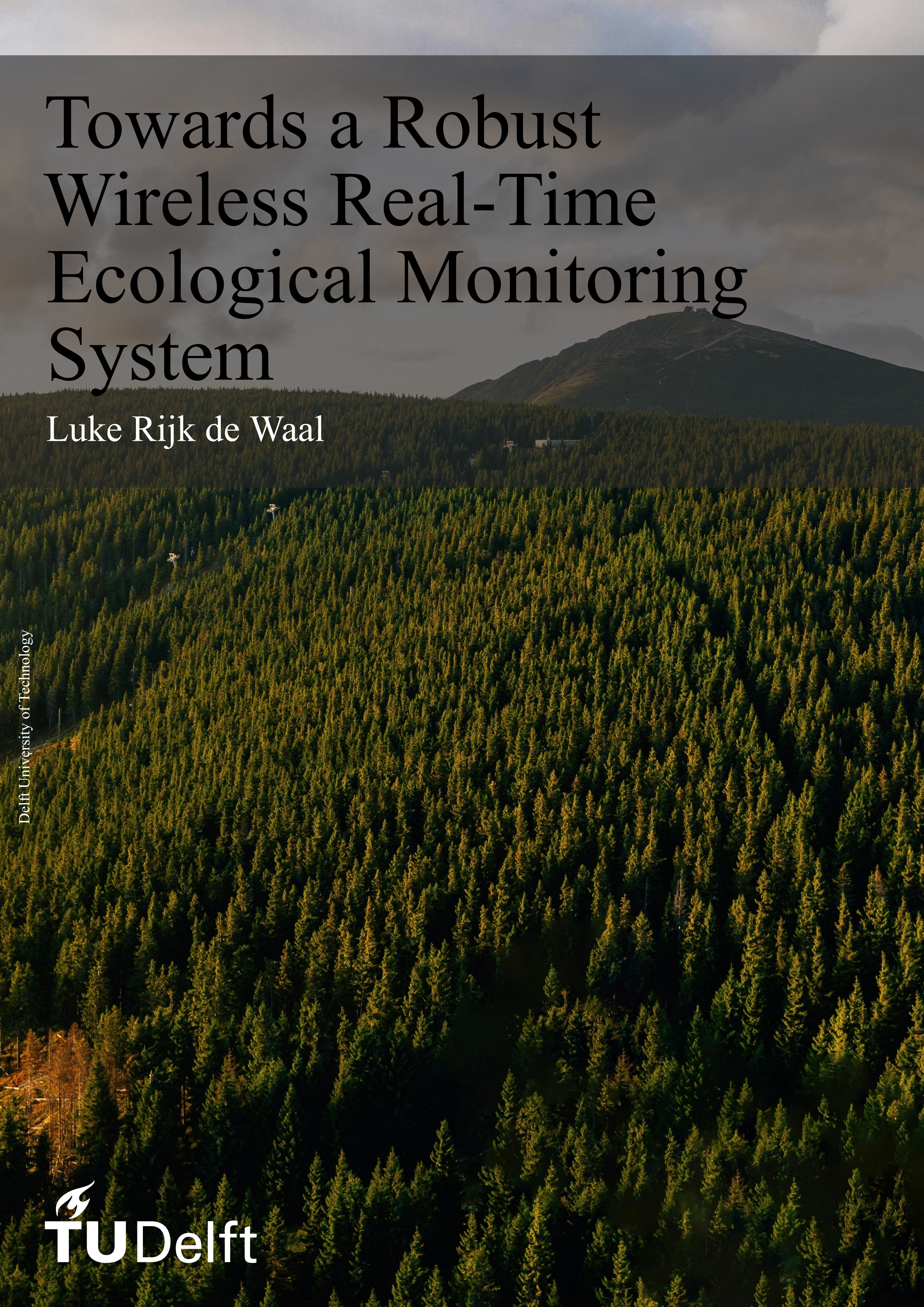
Towards a Robust Wireless Real-Time Ecological Monitoring System

Luke Rijk de Waal

Delft University of Technology



TU Delft

Towards a Robust Wireless Real-Time Ecological Monitoring System

by

Luke Rijk de Waal

to obtain the Master of Science dual-degree between
Aerospace Engineering & Embedded Systems
at the Delft University of Technology
to be defended publicly on the 30th of April, 2024

Faculties:	Aerospace Engineering	
	Electrical Engineering, Math & Computer Science	
Degrees & Tracks:	Control & Operations	Control & Simulation
	Embedded Systems	Software & Networking
Student Number:	4560000	
Project Duration:	February 2023 - April 2024	
Thesis Committee:	Dr. Ewoud Jan Jacob Smeur	(AE) Chair
	Dr. Ir. Salua Hamaza	(AE) Supervisor
	Dr. Raj Thilak Rajan	(EEMCS) Supervisor
	Dr. Ir. Richard Hendriks	(EEMCS) Examiner

Cover: Aerial Photo of Forest Across Mountain. Pec pod Sněžkou, Hradec Králové Region, Czechia. Photo by Marcin Jozwiak

TU Delft

Towards a Robust Wireless Real-Time Ecological Monitoring System

Luke Rijk de Waal
Embedded Systems and
Aerospace Control & Simulation
Delft University of Technology
Delft, The Netherlands, 2628CD

***Abstract*—Climate change poses a serious threat to ecosystems and increases the need for accurate and rigorous monitoring of ecosystems. Current monitoring solutions are often bulky, expensive, and lack critical functionalities such as on-board inference capabilities, robust wireless connections, and a diverse sensor suite. Ecological monitoring projects often suffer from inefficiencies caused by the large time delays between collecting data and analyzing said data, as well as having to spend large amounts of time in the field setting up the sensors manually. This thesis addresses many of these issues by designing a sensor with an extensive sensor suite, robust wireless capabilities and an on-board audio classifier able to perform real-time inference. Furthermore, attention is paid to making the system extendable in the future and allow for potentially integrating the sensors with a drone delivery- and retrieval system. The system tests performed indicate that the system has great potential given more time to tweak some of its identified shortcomings.**

Keywords: Ecological Monitoring, IoT, Edge Computing

Contributions

This thesis not only showcases the skills that have been taught during the master programs in terms of embedded programming and hardware design, it also highlights the required focus and dedication needed in order to complete the design- and implementation of any engineering product from start to finish through iterative design, refinement, and trial & error.

What sets this thesis apart is its direct relevance to current global challenges. Recognizing the limitations of existing monitoring tools, this thesis demonstrates a solution that not only addresses these shortcomings but also offers innovative features to enhance current ecological monitoring efforts. The set of common shortcomings and obstacles in current state-of-the-art environmental monitoring systems are:

- The exceedingly high acquisition costs. Most systems cost an exorbitant amount to acquire with comparatively very few features available. Or, in order to use the system, proprietary software must be used with a monthly subscription. While profitable for the system developers, this discourages widespread use.
- A lack of on-board processing capabilities. Many environmental sensing solutions are passive in the sense that data is collected, but no further processing is performed on the data until retrieval, resulting in large delays between sampling and analysis.
- The dependence on wireless infrastructure or outright absence of wireless capabilities. Many systems either require mobile networks to provide coverage in the monitored area, which is not always realistic in remote areas. Or, no wireless capabilities are offered at all, meaning any data or inference results are only available for to the user after retrieving the sensors from the field.

In essence, this thesis encapsulates technical expertise, practical innovation, and environmental consciousness. Overall, it demonstrates the potential for interdisciplinary solutions to address pressing global issues.

Introduction

It is well known and widely accepted in the scientific community that human activity is largely to blame for the trends in climate change seen today. The Intergovernmental Panel on Climate Change (IPCC) has stated that *"Human activities, principally through emissions of greenhouse gases, have unequivocally caused global warming"* [1]. The consequences of climate change for habitats, both terrestrial and aquatic, can only be speculated. Estimates of the IPCC state that a large number of terrestrial species will have a high risk of extinction if these climate change trends continue the way they do currently [1].

In order to maintain an accurate estimate of the state of ecosystems, it is not uncommon for researchers, conservationists and scientists to partake in Ecological Monitoring (EM) activities. These activities can range from collecting samples to be analyzed in a lab, taking pictures or video of animals, recording animal sounds, analyzing soil or air quality, etc. The most renowned EM project is that of Charles D. Keeling in 1976, where the atmospheric CO_2

contents were measured over a prolonged period of time. His findings sparked the debate of the relationship between atmospheric CO_2 contents and climate change [2]. Without his findings, this link may not have been made until much later, demonstrating that without such data being collected, it would be extremely difficult to understand how and why ecosystems are changing, and more importantly, how to respond to these changes. Likens and Lindenmayer aptly state in their book *"Effective Ecological Monitoring"* (2018) that without frequently collecting ecological information for prolonged periods of time, changes to ecosystems may go unnoticed until it is far too late [3].

It is clear that, without reliable data, lawmakers and governments cannot make informed decisions regarding conservation policies, and cannot accurately determine the efficacy of current conservation programs. A problem that plagues the field of EM however, is cost, availability, and usability of currently available monitoring systems, as well as a data-pipeline pestered with inefficiencies and hard-drives full of untouched data.

The purpose of this thesis is to address these issues by developing a monitoring system that is capable of logging valuable data in real-time while remaining a cheaper alternative to the current state-of-the-art. This is to be achieved by integrating the data collection and data analysis on the same chip, saving researchers and analysts time and effort, while simultaneously increasing the amount of valuable data collected.

1. Background Information

In this section a more in-depth description of the current state of the field of EM is given, together with the problems the field is currently facing. Furthermore, a more detailed description of the types of data that are to be collected is given.

1.1. Ecological Data & Variables

The goal of EM activities are to sample and collect data that help describing the environment and its inhabitants. What this data exactly entails depends on the study that is being performed. Generally speaking, there are two types of ecological variables that describe an ecosystem: Biotic- and Abiotic Ecological Variables.

An Abiotic Ecologial Variable (AEV) describes the physical state of an ecosystem over time through well-defined physical quantities. Commonly sampled AEVs are air temperature, air pressure, relative humidity, soil pH, atmospheric particle contents, radiation levels, solar intensity and wind-speeds. These variables are usually straightforward to measure and the required equipment is widely available.

A Biotic Ecological Variable (BEV) on the other hand describes a quantity or quality related to a biological component (flora & fauna) of an ecosystem. These are in general much harder to accurately measure using automated systems due to the often qualitative nature of these variables. Examples of such variables are best summarized by Besson et al. (2022) [4] to be the individual behaviours and traits of an animal, or the abundance and distributions of species in a given area. These variables have historically been monitored solely by humans due to the absence of the required technology. Without artificial intelligence it is near impossible to determine, in an automated fashion, what animal has been spotted in a video or recorded in an audio recording. Furthermore, the computational power required for neural networks to operate have always been much higher than most low-power Micro-Controller Unit (MCU)s/Micro-Processor Unit (MPU)s could offer. This is now starting to change, and many fields are starting to employ edge computing, which is a shift away from the centralized cloud-computing model and towards a more decentralized data model, with computations happening "at the edge", with the result of reducing bandwidth requirements and increasing response times [5].

1.2. Problems with Ecological Monitoring

With EM evidently playing a critical role today and in the future regarding nature preservation, an effective method for gathering and analyzing data is essential. There are a multitude problems with current EM methods however. Many of these issues are discussed by Besson et al. (2022) [4] and Lovett et al. (2007) [6].

1.2.1. Data Consistency

First and foremost is the issue of data consistency. Many researchers and institutions will have their own methodology, and own data formats, making data-sharing difficult and time-consuming. This is not aided by the fact that much of the ecological variables are complicated to numerically represent, could be measured at different spatial and temporal scales for different species, and may be measured at different locations and different times of year. This is also called the problem of **Data Variety** in the field of Big Data, and prevents effectively scaling up any data analysis efforts.

1.2.2. Data Reliability

Data reliability, or the lack thereof, is another problem that plagues the field of EM. Particularly, when manual sampling is employed as data-collection method, the reliability of the measurements is solely dependent on the one taking the measurements and thus a perfect opportunity for human error to be introduced. Limiting the possibility of human error should be a top priority for any ecologist taking measurements, as erroneous data will lead to erroneous conclusions. In the field of Big Data, this is also referred to as the problem of **Data Veracity**.

1.2.3. Data Velocity

Time is a scarce resource in almost any possible sense. For ecological research this is no different: EM is a time-consuming activity and in the past often required researchers to go out in the field and collect data manually. Time spent in the field is time not spent analyzing collected data however. Lovett et al. (2007) states that one of the most important habits for effective ecological monitoring is the "*Continual examination, interpretation and presentation of monitoring data*" [6], as this allows for catching errors early on, and the ability to quickly detect patterns and trends, in turn allowing quick responses to alleviate changes in ecosystems. As such, minimizing the time between measurement and analysis should be a priority for any researcher. This is also known as **Data Velocity** in Big Data.

1.2.4. Required Infrastructure

Nowadays, it is more common to (partially) automate the data collection tasks through the use of a MCU or MPU, electronic sensors, and wireless communication, as opposed to manual data gathering. All of this equipment requires the appropriate infrastructure to support itself, such as power and a wireless connection.

When selecting equipment, the provided wireless communication medium matters significantly. If the equipment makes use of mobile data networks, the area where the equipment will be located must have coverage. Similarly, equipment utilizing technology such as Wi-Fi requires range extenders to function correctly. These infrastructure requirements significantly influence the usability of the sensors in different parts of the world and restricts usage to certain areas only, or to people who have the financial means to invest in the required infrastructure. It should be noted that the areas where monitoring is the most valuable, tend to be the most remote regions with the generally poorer mobile coverage and little to no possibility for high-speed internet connections.

In a similar fashion, these remote areas must be reached by people in order to position the sensors and retrieve them after the monitoring has been completed. When the area that is to be monitored is remote and not easy to reach for people, this will naturally introduce larger delays and costs to set the system up and maintain it during the mission.

1.2.5. System Costs

Ecological monitoring equipment varies quite significantly in the attached costs. As previously mentioned, infrastructure is a large potential investment that needs to be made depending on the equipment used. Other than the physical and wireless infrastructure, the initial investment cost can be quite high too for relatively simple equipment. For example, a €1,635.00 (2024) investment yields the TRAMEX-5 monitoring system. This includes 5 sensors that can monitor a set of two AEVs (Temperature and Relative Humidity) with a long-range wireless medium (3km line-of-sight) and send multiple messages per hour [7]. This is not an investment many researchers can make nor justify as such a system is unable to record any BEVs. Other products such as the wildlife audio recorders by Wildlife Acoustics do allow for recording high-quality audio, but instead lack any sensors to record AEVs [8].

1.2.6. Synopsis

The shortcomings and missed opportunities discussed in this section is where this project will aim to make improvements. In particular, addressing the issues regarding data variety, data veracity, data velocity, investment costs, and time spent in the field will be the main focus on this project, as will be explained in more detail in section 3. In section 2 similar and related projects will be showcased together with the respective advancements made and concepts explored.

2. Related Work

In recent years, due to widespread availability of consumer electronics and the large increase in capabilities of these consumer electronics, a myriad of hobbyists projects as well as research-oriented projects have spawned a wave of EM equipment with the main goal of making this equipment affordable and more accessible than the currently available commercial equipment.

One of the more influential of such projects is Solo: An open-source, customizable and inexpensive audio recorder for bioacoustic research (Whytock & Christie, 2017) [9]. This solution is based on the Raspberry Pi, a hobbyist Single-Board Computer (SBC) mainly intended for educational purposes. The main selling-points of Solo are the comparatively low price combined with the highly-customizable hard- and software, as opposed to closed-source commercial products with limited to no customizability. The AURITA project by Beason et al. [10] combines Solo with the ability to record both audible and ultrasonic audio in order to extend the use-cases of Solo to bats, amphibians and insects as well.

Continuing this trend of open-source low-cost equipment, Sethi et al. (2018) introduces a robust Real-Time (RT) autonomous acoustic monitoring device with networking capabilities [11]. The main difference between this implementation and Solo, is the added networking capabilities as well as the addition of a solar panel, allowing the system to autonomously record data indefinitely given enough sunlight and given a stable mobile network connection.

The AudioMoth is another open-source project by Hill et al. (2018) [12], [13]. The AudioMoth differs significantly from the aforementioned two projects in the sense that the AudioMoth is a custom-designed Printed Cicruit Board (PCB) and runs custom-made firmware. The design is completely transparent and open-source to allow for easy customization by the user. It furthermore boasts recording capabilities in the ultrasonic range, allowing recordings to capture bat-calls, amphibians, and certain insects, applications where standard recording equipment falls short. The main drawback of the audiomoth however is its lack of wireless capabilities, meaning it has to be retrieved before any data can be inspected.

An honorable mention is the TeensyBat project by Edwin Houwertjes and Cor Berrevoets [14], [15]. The TeensyBat is a hobbyist project that allows electronics enthousiasts and acoustic monitoring enthousiasts alike to create their own handheld bat monitoring device. It is entirely open-source and can only be bought in parts to be assembled manually, with no intention by the creators to commercialize their product.

Finally, a topic that is still heavily being researched is not just the autonomous collection of data through affordable electronic devices, but also the autonomous deployment of these devices. Many devices can record for days, weeks, or even months at a time. Eventually however, the devices must be retrieved and redeployed. This is still a very time-consuming task. In order to reduce this time-investment, the usage of drones is becoming more and more of a reality. Hamaza et al. (2019) [16] discusses the usage of an unmanned aerial manipulator to position and retrieve sensors in both indoor and outdoor environments. Many other methodologies for delivering sensors in arboreal environments have been explored by Hamaza et al. (2020) [17] and Kocer et al. (2021) [18]. Integrating such a delivery and retrieval system may prove invaluable for EM projects of the future.

3. Goal of the Project

The current issues and problems with EM have been identified in section 1, with other projects discussed in section 2 attempting to address some of these issues. It is these aforementioned problems that will be the focus of this project. Inspired by the EM data-analysis pipeline proposed by [4], there will be five main points of improvement that this project will attempt to address:

1) **Low System Acquisition Cost**
   The system should be affordable for anyone to allow widespread use and more effective use of the networking capabilities.
2) **Data Reliability**
   The ability to continually and reliably sample important ecological variables.
3) **Robust Wireless Communication**
   The ability to reliably communicate with a central hub to relay data or receive control commands to minimize time spent in the field.
4) **Onboard Inference**
   The ability to instantly perform inference on acquired data to minimize the time between measurement and analysis.
5) **Drone Deployment- & Retrieval**
   The ability to deploy and retrieve the sensor nodes by the use of drones to minimize time spent in the field.

As a result, the goal of this project can be summarized by the following statement:

The design and development of a cheap sensor node capable of reliably measuring ecological variables, perform inference on said variables, and relay any data and/or findings over a wireless network with little to no infrastructure requirements, all while being lightweight and small enough to be deployable by drones.

4. System Design

In this section, the design overview of the system is given, followed by a description of the mode of operation of the system, as well as functional block diagrams of the subsystems. The detailed hardware- and software designs are presented in the sections afterwards.

4.1. Design Overview

The high-level system diagram can be seen in Figure 2 where every module and its function is shown. The SBC on which the design builds is the BeagleBone Black [19] and can be seen in Figure 1. The system will be able to measure AEVs as well as infer BEVs through audio. In particular, the system will attempt to detect and classify birds by using the BirdNet classifier [20].

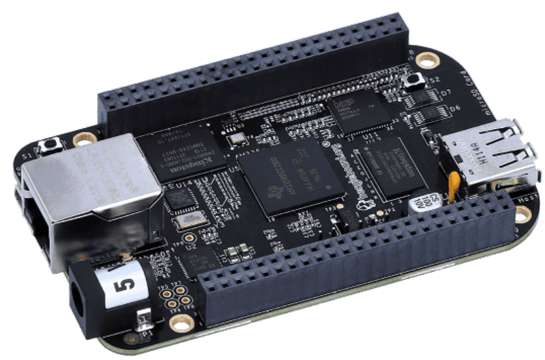


Figure 1: BeagleBone Black SBC (image source)

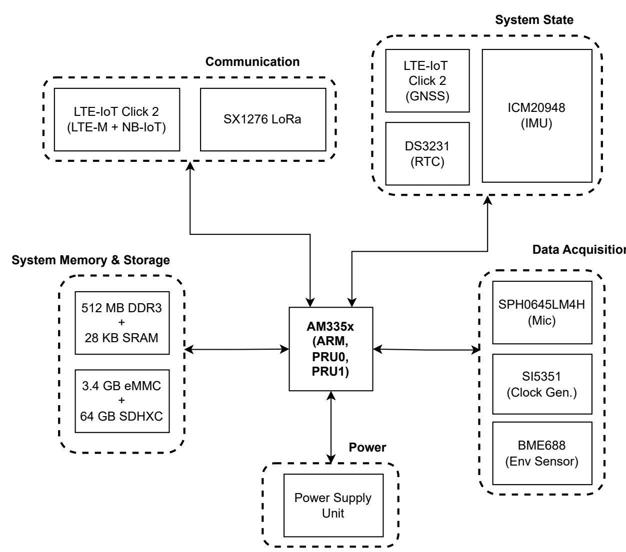


Figure 2: High-Level System

4.1.1. Data Acquisition

The data acquisition module consists of two sensors: the BME688 environmental sensor, and the SPH0645LM4H digital microphone. It also contains the SI5351 clock generator, which is required for the correct operation of the digital microphone.

The BME688 samples four AEVs: temperature, pressure, relative humidity, and gas resistance. These variables can be used to determine the presence of certain chemicals in the air, as well as keep track of the physical state of the environment.

The SPH0645LM46 digital microphone samples a single-channel audio stream at 48 KHz. This sampling rate is chosen due to it being the required sample rate to use BirdNet [20]. BirdNet is a classifier trained to detect- and classify bird-sounds. As a result, the microphone and BirdNet classifier combination yields multiple BEVs. In particular, it can yield an estimate of the species' abundances in the area as well as its distribution over time given long-enough monitoring times. Furthermore, when multiple sensors are deployed over an area, it can produce an estimate of the physical distributions of the bird species at any given point in time.

4.1.2. System State Monitor

The system state monitoring module consists of three components: the DS3231 Real-Time Clock (RTC), the LTE-IoT-Click-2 Global Navigation Satellite System (GNSS) chip, and the ICM20948 Inertial Measurement Unit (IMU) with builtin Digital Motion Processor (DMP).

The DS3231 RTC ensures that the board will always have an accurate estimate of the time, even when no internet or GNSS connection is available or when the board is powered down. The RTC has an external rechargeable battery in case no power can be provided by the board itself.

The LTE-IoT-Click-2 is based on the BG96 chip and possesses both GNSS and Long Term Evolution - Machine Type Communication (LTE-M) / Narrowband IoT (NB-IoT) capabilities, though the LTE-M / NB-IoT functionalities remain unused within this project.

Finally, the ICM20948 IMU uses its internal DMP to obtain an absolute orientation as a quaternion by incorporating accelerometer measurements, gyroscope measurements, and magnetometer measurements. The DMP requires custom firmware to be uploaded before use, after which it will run internally at 200 Hz, but only outputs data at 5 Hz to not overflow the data buffers, allowing it to maintain high accuracy with little to no computational power required from the host processor. The IMU furthermore features a Wake-On-Motion (WOM) interrupt pin which is used to detect if the sensor has significantly moved to detect a fall or being picked up.

4.1.3. Wireless Communication

The wireless communication module consists of two components, though as previously mentioned, the LTE-M / NB-IoT capabilities of the LTE-IoT-Click-2 remain unused. The SX1276 LoRa module is the main form of wireless communication to- and from the system. With a theoretical maximum range of over 10 km in rural areas, large areas can easily be covered without any extra infrastructure required to operate. It can operate in a wide frequency range from 137 MHz to 1020 MHz [21], making it flexible and able to broadcast anywhere within the sub-GHz Industrial-Scientific-Medical (ISM) bands, with extra requirements attached [22], [23].

4.1.4. Memory & Storage

The SBC features 3.4 GB of Embedded Multi Media Card (eMMC), 512 MB of DDR3 Random Access Memory (RAM), and an Secure Digital Extended Capacity (SDXC) card slot for extra storage. The SD card slot allows for almost 48 Hours of raw audio to be recorded on a 32 GB card.

4.1.5. Power Supply

Power can be supplied via three distinct methods. The SBC possesses a mini-Universal Serial Bus (USB) port from which the board can be powered, though this limits the available current to 500 mA and may cause problems when many peripherals are attached. The board also possesses a 5V/2A 5.5mm/2.1mm barrel jack which is the recommended way to power the board when possible. Finally, the board has pins available where it is possible to attach a battery. The board can be powered with a 3.6V supply via a battery, unless the USB connector is in use, which requires a 5V supply. As a result, the maximum possible power consumption of the board is $5\text{ V} \cdot 2\text{ A} = 10\text{ W}$ if the USB is in use, or 7.4 W without the USB.

4.2. Functional Description

As described in subsection 4.1, the system contains multiple functional modules. In this subsection the functional behaviour of the system as a whole, as well as the functional behaviour of every module separately, will be described.

4.2.1. Mode of Operation

The system, upon starting, will first enter the initialization phase. Here, the GNSS peripheral will try to obtain a fix. Depending on how long ago the last fix occurred and how many satellites are visible, this may take up to two minutes. The radio is initialized shortly after obtaining a fix and put into receiver mode, followed by the initialization of the data acquisition peripherals. The data acquisition peripherals are either initialized through the ARM processor, through one of the Programmable Real-Time Unit (PRU)s, or both. Finally, the BirdNet model is loaded into memory. After all the initialization steps have been executed successfully will the main phase commence.

During the main phase, the PRUs will start the data acquisition. Simultaneously, the ARM processor will poll whether any of the data buffers is ready to be processed. When a data buffer is ready, the ARM processor will extract the data and apply any necessary transformations to the data before logging it to disk and/or passing it along to other components who require the data, such as the radio or BirdNet. Meanwhile, the GNSS peripheral will continue obtaining position fixes at a fixed interval and the radio peripheral will listen for messages. Simultaneously, if large movements of the system are detected (ie. it has fallen or is picked up by an unauthorized person) by the IMU, a WOM signal is sent from the IMU to the ARM processor, where the ARM processor will get the radio to transmit a help message to the base station, such that appropriate actions can be undertaken.

Finally, to terminate the program and finalize the mission, all the peripherals are deactivated or put into low-power mode, the PRUs are told to halt, and all the collected data not written to file yet are written to file.

In the following paragraphs, the functional description of the core components and the corresponding data flow is explained in more detail.

4.2.2. PRU0 Data Flow

The first PRU, PRU0, handles the environmental data acquisition and IMU data acquisition. In Figure 3 the data-flow is shown. The PRU samples the environmental data and IMU data at fixed rates, with the IMU data being sampled an integer-multiple of the environmental data sample rate to simplify the sampling schedule and reduce code complexity. The PRU passes the raw data collected from the sensors into their respective buffers in shared memory, where the ARM processor can retrieve the data ans apply any necessary transformations. Finally, the processed data is stored on disk.

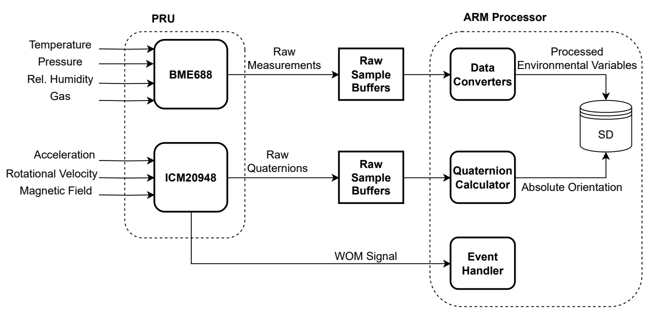


Figure 3: Functional Flow: Environmental- & IMU Data

4.2.3. PRU1 Data Flow

The first PRU, PRU1, handles the audio data acquisition. In Figure 4 the data flow can be seen. The PRU needs to ensure that no samples are skipped, and therefore performs no tasks other than ensuring the samples are extracted from the audio data register and putting the samples in a buffer in the main memory region. The memory region is a cyclical buffer divided into multiple chunks from where the ARM processor can extract the data and apply any necessary transformations. In particular, the audio must be scaled appropriately and the audio has a certain DC offset which needs to be corrected before the audio can be used for analysis. The DC offset is calculated from the first set of buffers, until the estimate has converged, after which it is subtracted from all audio samples. After the audio is processed, it can be used by BirdNet to perform an inference run, after which it is also stored to disk.

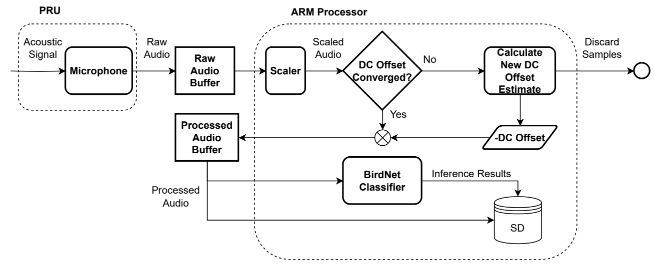


Figure 4: Functional Flow: Audio Data

4.2.4. GNSS Data Flow

The GNSS data flow is very straightforward, as seen in Figure 5. A command is sent over the Universal Asynchronous Receiver / Transmitter (UART) to return the current position and any extra information regarding the fix, such as the number of satellites visible and the Horizontal Degree of Precision (HDOP) associated to the position fix. This raw serial message is parsed, after which the data is stored to disk.

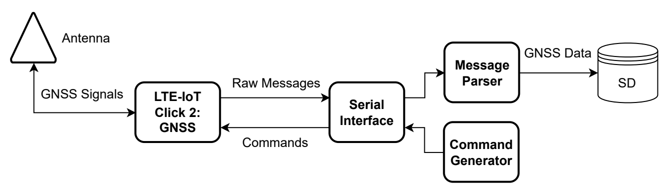


Figure 5: Functional Flow: GNSS Data

4.2.5. Radio Communication

One of the more complex mechanisms is that of the Long-Range (LoRa) radio, which can be found in Figure 6. The radio module, upon receiving a message, triggers an interrupt, triggering the message to be extracted from the module First-In First-Out (FIFO) buffer and put into the incoming message queue where it is parsed and any subsequent actions are executed. Such actions may include changing program settings or collecting data from other threads. When transmitting a message instead, an interrupt is triggered upon finishing the transmission, after which the radio module will be put back into receiver mode. Every message that is transmitted or received has some associated diagnostic data, such as Signal-to-Noise Ratio (SNR), Received Signal Strength Indicator (RSSI) or Time-On-Air (TOA) values, which are logged for performance evaluation after program termination.

4.3. Hardware Design

The system runs on the AM335X processor and has 5 main branches of modules required to operate: Data Acquisition Modules, System State Monitoring Modules, Wireless Communication Modules, Memory and Storage Modules, and a Power Supply module. In this section the finalized detailed hardware design is showcased. Firstly a

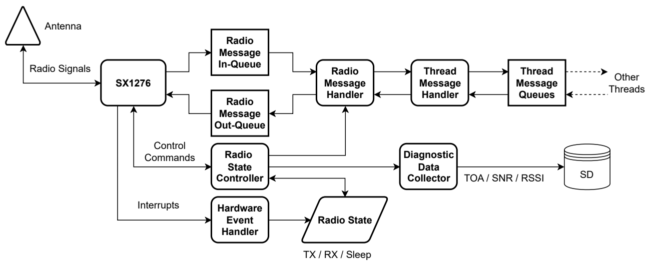


Figure 6: Functional Flow: Radio Communication

high-level description of the hardware is given, followed by a more in-depth description and specification of the hardware. Finally, a brief description of a custom-made audio sampling component is given.

4.3.1. Hardware Architecture: High-Level

In Figure 7 a high-level overview of the hardware and its connections is given. Here, all the interconnections between the sensors, peripherals, memory regions, and processing units is given.

The ARM core is the main coordinator of the system, while the two PRUs are controlled by the ARM core. To communicate between the PRUs and ARM, a region of the shared memory bank is reserved. Furthermore, the AM335X has a set of interrupts dedicated to the PRUs, allowing the ARM and PRUs to indicate events have occurred and react accordingly. The PRUs have access to the same peripherals as the ARM core, as well as all the memory regions available on the board.

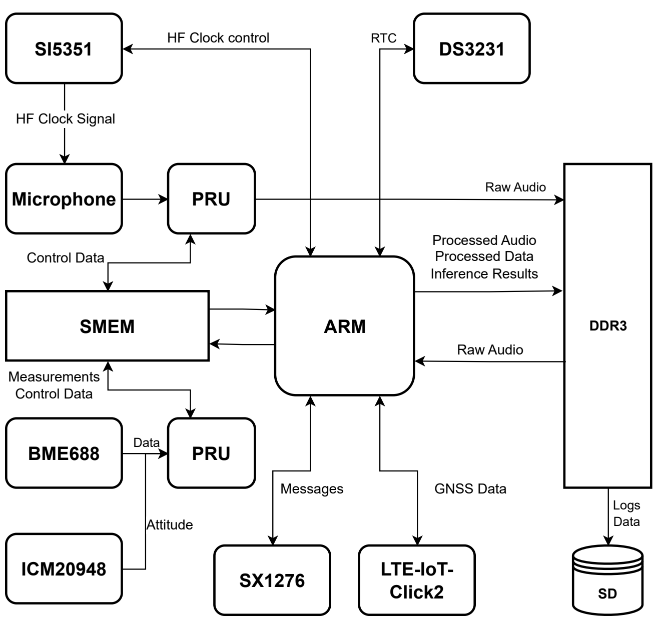
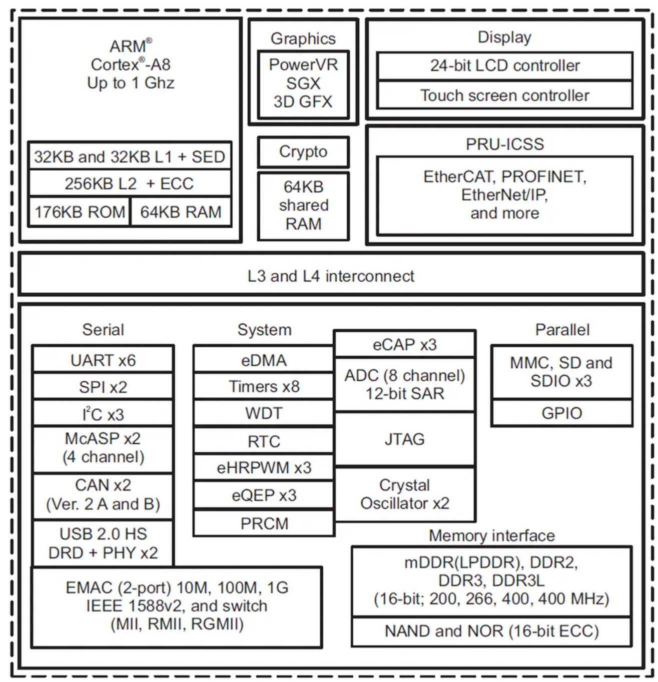


Figure 7: High-Level Hardware Connections & Data Flow

4.3.2. Hardware Architecture: SBC Specifications

The AM335X processor is an ARM-based heterogeneous multiprocessor with 3 cores by Texas Instruments [24]. The primary core runs at a maximum of 1 GHz, while the two remaining cores run at 200 MHz. Each of these two smaller cores is a PRU, and very distinct from the main core. Each PRU has limited Data RAM (DRAM) and Instruction RAM (IRAM) (8 KB of each), as well as 12 KB of Shared RAM. PRUs are extremely consistent due to their simple nature. There are no cache misses or context-switches to induce unexpected delays, and the clock frequency is extremely consistent. The SBC furthermore features 3.4 GB of eMMC, 512 MB of DDR3 RAM, and an SDXC card slot for extra storage. In Figure 8 an overview of the AM335X and its functionalities is given.

Figure 8: AM335X Functionality Specification [25]



4.3.3. Hardware Architecture: Peripherals

In Figure 25 the hardware schematic is given. The devices connected to the SBC make use of different types of hardware communication protocols such as Inter-Integrated Circuit (I^2C), Inter-Integrated Circuit Sound (I^2S), Serial Peripheral Interface (SPI) and UART, alongside digital General-Purpose Input-Output (GPIO). In Table 1 an overview of the utilized protocols and the corresponding peripherals is given.

Out of all the aforementioned protocols, the most commonplace is I^2C with a total of 4 peripherals being attached to one of two I^2C busses. In particular, the DS3231 RTC possesses its own dedicated I^2C bus. Meanwhile, the BME688, ICM20948, and SI5351 are all connected to the same I^2C bus. All I^2C busses run on the high-speed mode with a clock of 400 KHz and use 7-bit addressing.

The I^2S protocol is a specific case of Time-Division Multiplexing (TDM), where TDM is a way of transmitting and/or receiving signals over a common signal path. In the case of I^2S, two channels (time-slots) are used; the left and right channel. The SBC contains two Multichannel Audio Serial Port (McASP) modules, capable of a wide variety of TDM configurations among other protocols. The I^2S protocol is only used by the microphone and is currently set-up to sample at 48 KHz with a bit-clock of 3.072 MHz and samples of 32 bits wide with 18 bits of data precision. The resulting sampling frequency and the bit-clock are related by Equation 1:

$$f_s = \frac{f_b}{N_c \cdot w_s} \tag{1}$$

where f_s is the audio sampling frequency, f_b is the bit-clock, N_c is the number of channels, which is always 2 in I^2S, and finally, w_s is the number of bits per sample per channel, which usually is 16-, 24- or 32-bits. The required bit-clock could not be achieved using the internal clock dividers, and as such the external clock generator, SI5351, had to be used to generate the exact bit-clock of 3.072 MHz.

To communicate with the SX1276 LoRa radio module, the SPI interface is used together with multiple GPIO lines. A clock-speed of up to 10 MHz is supported by the radio chip for the SPI transfers, and up to 48 MHz on the SBC. The GPIO lines are used to indicate events / interrupts by the chip.

Finally, the UART interface is utilized by the LTE-IoT-Click-2 module to obtain GNSS data and information. This interface can furthermore be used to communicate through a mobile network given coverage and an appropriate SIM-card is present in the module, though this feature is left for future design iterations. The SBC supports Baudrates up to 3.6864 Mbps, while the LTE-IoT-Click-2 module supports Baudrates up to 3 Mbps, making it suitable for streaming live audio data given appropriate network coverage.

4.3.4. Bill-Of-Materials

In Table 2 the major components and their online price-ranges are given. The total amounts to around €225 for a sensor with LTE-M / NB-IoT capabilities (excluding SIM and network subscription costs), and €175 is an estimate of the costs of the system without the networking capabilities based on the price of components with GNSS capabilities but no networking capabilities. These prices are excluding batteries, housing, drone and attachment mechanism for the sensors as these features are left for future design iterations.

While the prototype with LTE-M networking capabilities exceeds the cost set in requirement NR-4 by €25 or %12.5 of the intended price, these costs can be significantly decreased in the future when the components are bought in bulk or when custom-designed PCBs with directly-placed components can be printed in bulk. If no LTE-M networking capabilities are required, the system costs can be reduced even further to be within the bound set in non-functional requirement **NR-1** by default.

TABLE 1: Summarizing Table: Hardware Connections

Module	**Interface**	**Adapter**	**Core**
BME688	I^2C	I^2C-1	PRU0
ICM20948			
SI5351			ARM
DS3231		I^2C-2	
SX1276	SPI	SPI-1.0	
LTE-IoT-Click 2	UART	UART4	
SPH0645LM4H	I^2S	McASP0	PRU1

4.3.5. Ultrasonic Audio Sampler Design

A hybrid bird- and bat-detector would maximize the use of the system for both day- and night-times. A notable motivation for pursuing bat-detection within this project is the very high prices and bulky equipment associated with current state-of-the-art bat monitoring equipment [10], [12].

For this reason, a full-spectrum audio-sampling PCB is designed to be used with the monitoring system. In Figure 10 a 3D model of the PCB is shown. The detailed schematic and PCB design diagrams can be found in Appendix C (Figure 26 and Figure 27).

The components were selected based on an existing solution, the TeensyBat [14]. It uses the SPU0410LR5H analog Micro-Electromechanical System (MEMS) microphone together with a high-speed Analog-to-Digital Converter (ADC) (ADS8883) able to sample up to 680 KHz with an 18-bit sample-width. Furthermore, the PCB features

Figure 9: Driver Interaction Diagram

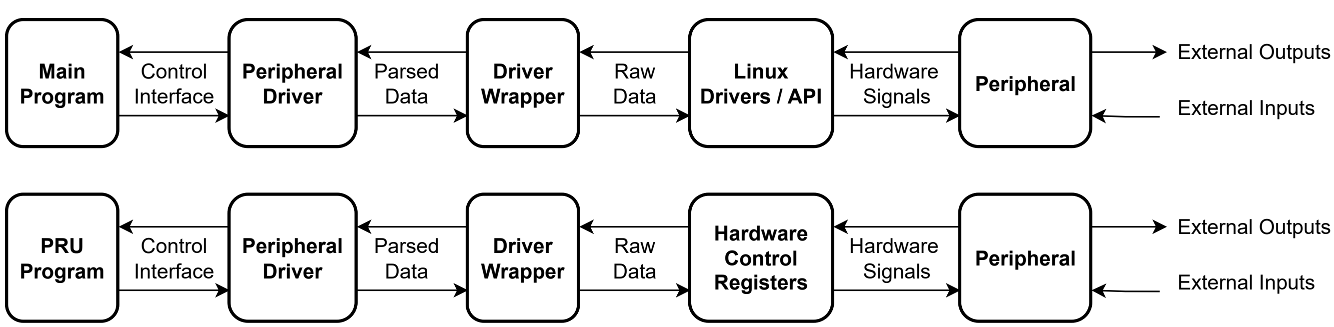


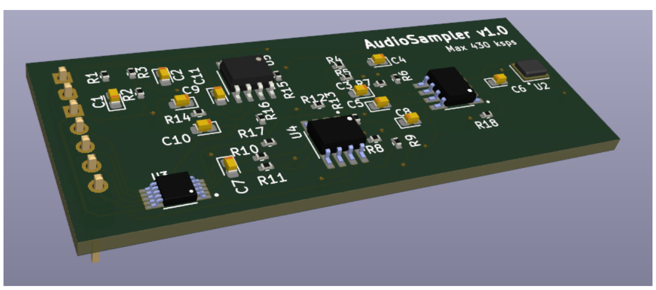
TABLE 2: Component Price Breakdown

Component	Description	Price Range [€]
Beaglebone Black	SBC	47-70
ICM20948	IMU	17-22
BME688	Environmental Sensor	20-26
DS3231	RTC	3-5
LTE-IoT-Click 2	LTE-M / NB-IoT GNSS	72-74*
SX1276	LoRa	12-15
SI5351	Clock Generator	9-12
SPH0645LM4H	Digital Microphone	10-16
Total	(with LTE-M)	190-240
	(without LTE-M)	150-200

* Price excludes the cost of the SIM card, antennas, and mobile subscription needed to operate on the LTE-M / NB-IoT networks

an active low-pass filter with a cutoff frequency of 215 KHz, meaning a sample-rate of 430 KHz can be sustained. The ADC interface uses the SPI protocol and can therefore easily be interfaced with.

Figure 10: Audio Sampler 3D Model



Sadly, due to time-constraints and supply-chain limitations, the time it took for the PCB to be printed, assembled and shipped would be too large and as such this design is left unused in the final product.

4.4. Software Design

In this subsection the software design and development process is explained. In particular, the relationship between hardware and software and how this dynamic influenced the software design over time. Furthermore, the software components that are part of the system are explained, followed by a more detailed explanation of how the components interact, and how they fit into the overall architecture.

4.4.1. Development & Design Process

Software design can only truly commence once the underlying hardware as well as any peripherals that need to be interfaced with are known. As such, software design only started after the finalization of the system's Bill Of Materials (BOM).

With all the underlying hardware and required peripherals known, the first decision to make is the Operating System (OS). The Beaglebone Black SBC has multiple supported OSes, though due to the board being over a decade old, many OSes are no longer maintained or updated. The most up-to-date OS at the time of writing is Linux Debian 11, with kernel version 5.10.168. For better embedded and real-time performance, the kernel with RT extensions and preemptible threads is utilized. While this does not offer the deterministic performance guarantees of a true Real-Time Operating System (RTOS), development and debugging is significantly simpler on a standard OS. Additionally, if the system is able to run within its timing requirements, given some margin, on a standard OS, then it is most certainly going to be able to do so on a true RTOS.

The next steps in the design & development process of the software is the implementation of the drivers needed to interface with the system's peripherals. In Figure 9 the general layout of the drivers developed to interact with a peripheral is given for both ARM and PRU programs. The peripheral is interacted with using a communication protocol such as SPI or I^2C. In the case of the ARM processor, this is done through the corresponding Linux library and Application Programming Interface (API). For the PRU, this is done by directly interacting with the hardware registers dedicated to the communication protocol. From here, raw data can be transmitted or received to and from the peripheral. A communication driver wrapper is created to simplify interacting with the Linux API or hardware registers. In turn, this communication driver wrapper is used by an overarching peripheral wrapper used to interact with a specific peripheral. As a result, the communication driver wrapper can be reused for multiple peripherals using the same communication protocol. Finally, the main programs use this peripheral wrapper to interact with the peripheral in question. It is logical to develop the drivers from the lowest-level up, and reuse as much code as possible.

After the development of all the required drivers, the main program and the PRU programs can be developed. The development process of these programs are wildly different due to the different resources available to the cores, as well as the different tasks that are to be executed.

In particular, the PRUs are to execute their programs with stricter timing requirements while possessing very little memory of their own. This lead to developing drivers with a memory footprint as small as possible, and a consistent execution time. Furthermore, the drivers are created by directly manipulating registers defined in the AM335X Technical Reference Manual [24]. On the other hand, the drivers to be run on the ARM processor utilize the provided Linux libraries and have no strict requirements on memory footprint.

With all peripheral drivers implemented for both the ARM and PRUs, the next step is the development of any software components that are not related to peripheral interactions, but still required for correct functioning of the system. These include components such as common datastructures, the BirdNet inference model, and the data logging system.

Finally, the software integration phase commences where all the components are connected and any additional software such as messaging queues is written to connect the components correctly.

4.4.2. Software Architecture

The software architecture seeks to be modular, extendable, scalable, configurable, maintainable and reliable while ensuring the minimum performance requirements of the system are met.

The software is divided between the three cores, where the PRU cores have a simple single-threaded program to execute with strict timing requirements, and the ARM processor instead runs a complex multithreaded program with tight timing requirements coordinating the whole system.

In the following paragraphs the functionalities and interactions of the different threads, components and PRU programs are discussed.

4.4.2.1 PRU Manager Thread

The PRU manager does what its name suggests; it interacts with the PRU cores through interrupts and memory-mapped regions in the shared memory bank. In particular, the PRU manager uploads the firmware to the PRU cores, as well as stopping the PRUs upon program termination. Furthermore, the data collected by the PRUs is taken from the buffers in shared memory and processed by the PRU manager. Examples of such processing include transforming raw data to floating-point data, calculating the DC-offset of the audio signal and correcting for it, and amplifying the audio signal in software. In order to prevent a buffer overflow, the PRU manager must run at a small-enough period to be able to extract all the data from the buffers before it is overwritten.

4.4.2.2 GNSS Thread

The GNSS thread is the simplest thread by nature, as it has a single task: retrieving GNSS data from the LTE-IoT-Click 2 through the UART interface. This task does not require the thread to run at a high frequency, as a GNSS fix is only taken every 20-30 seconds depending on the desired setting.

4.4.2.3 Radio Thread

The radio thread, or LoRa thread, is the highest-priority thread in the system. It runs at the highest frequency as well, due to its more strict RT requirements. This thread is, with the absence of an internet connection, the only way to communicate with the outside world. Furthermore, this thread monitors the WOM pin and needs to react to a WOM event in a timely fashion in order to minimize the risk of losing a sensor-node to theft or mishap. As such, it runs with a period of 100 ms, making this the theoretical worst-case response time in case of a WOM event.

4.4.2.4 BirdNet Thread

Finally, the BirdNet thread is the thread that requires the most Central Processing Unit (CPU) time, with a median CPU utilization percentage of 79.66%. For this reason, the BirdNet thread is also the lowest priority thread out of all threads, to ensure that higher priority threads can preempt it to stay on schedule. Its tasks include reading the processed audio data obtained from the PRU-manager thread and writing it to file. A raw binary file-format is used to reduce any required overhead. After saving the raw audio to file, it is passed on to the BirdNet classifier. The inference results are then put into the log-file, after which the thread is suspended until the next audio buffer is available.

4.4.2.5 Inter-Thread Communication

Every thread has an incoming priority queue which is check at the start of every thread's period. The queue is divided into 4 different priorities, from `URGENT` to `LOW` priority. These priority queues contain messages or requests for data. For example, the radio thread may receive a wireless message requesting the latest available GNSS data. The radio thread requests this data through the priority queue of the GNSS thread, after which the GNSS thread puts the latest obtained sample into the radio thread's incoming priority queue.

4.4.2.6 Thread Safety

There are multiple shared resources in the system, which could cause severe issues in a multithreaded environment. The most contended-for resource is the log file. Every thread has certain information it needs to put in the log file, which may be more or less depending on the chosen logging verbosity level. To prevent race conditions and thread starvation, mutexes with inherited priorities are utilized. Anytime a thread of priority p_1 requests the mutex when a thread of priority $p_2 < p_1$ holds it, the priority p_2 is temporarily boosted to p_1 to allow the thread to finish its work before dissolving ownership of the mutex. This prevents dead-lock and allows low-priority threads to finish their logging task gracefully. The inter-thread priority queues feature a similar system with mutexes, where every queue has its own mutex which must be locked before being able to read-or-write to and from the priority queue in question.

4.4.2.7 PRU0 Program - Environmental & IMU Data

The firmware running on PRU0 is responsible for collecting data from the peripherals connected on the I^2C bus. This includes the BME688 Environmental Sensor, and the ICM20948 IMU. Due to the constrained memory, the whole program does not fit into IRAM. Therefore, the program is split into 2 parts, each responsible for a different phase: the initialization phase, and the main phase.

In the initialization phase, drivers are initialized and sensor calibration for the BME688 is performed, after which the PRU halts. The ARM processor then transfers the firmware for the DMP over the I^2C bus, the firmware being too large to fit in PRU memory.

The main PRU firmware is then uploaded, which contains the last remaining setup and the main program loop. In this last setup stage, the heater-profile and sampling configuration settings are uploaded to the BME688. The sampling settings allow the selection of ADC oversampling factor and the Output Data Rate (ODR) of the sensor. The heater profile allows the user to determine how long the heating-element needs to heat up for, and what temperature it needs to hold. These parameters can be varied depending on what substances are to be detected [26].

In the main part of the firmware, the BME688 is sampled with a fixed period of T_{BME}. In-between samples, data is read from the FIFO buffer of the IMU with a sample-period satisfying $N \cdot T_{\text{IMU}} = T_{\text{BME}}$ with $N \in \mathbb{N}$, allowing for a simple inner- and outer-loop structure of the code. A value of $N = 20$ is found to be suitable, with $T_{\text{BME}} = 4$ seconds, and $T_{\text{IMU}} = 0.2$ seconds.

Samples are stored in dual-buffers in shared memory from where the PRU-Manager thread can extract it. Other than raw data, the PRU stores status information in shared memory as well, in order for the PRU-Manager to know when an unrecoverable error has occurred in the PRU core and may reboot the core or I^2C interface if required.

4.4.2.8 PRU1 Program - Audio Data

The firmware running on PRU0 is responsible for collecting raw audio samples over the I^2S bus. This program, having a much smaller memory footprint compared to PRU0, fits entirely into IRAM and does not require to be split-up.

The program initializes a reserved chunk of DDR3 memory and splits it into multiple buffers the size of the BirdNet input. The first 3 buffers are discarded, as the audio signal needs time to settle, after which the program continuously reads audio samples and stores them directly into DDR3 memory where the PRU-Manager extracts them. As with PRU0, other information regarding the status of the program is stored in shared memory.

5. Experimental Results

This section presents the outcomes of a multitude of different tests conducted focusing on every subsystem, as well as the system's performance as a whole.

5.1. Environmental Data

The collection of AEVs is one of the important capabilities of the system. The AEVs that are collected are temperature, pressure, relative humidity, and gas resistances, sampled at a constant interval of 4 seconds. Two experiments were performed, the first experiment being a long-term data-collection experiment, with the system logging data for approximately 15 hours. The second experiment involved testing the response to the presence of specific substances in the air.

5.1.1. Experiment I

In Figure 11, the AEV data collected overnight during the experiment is shown. In the early stages of the experiment, two distinct spikes in relative humidity and gas resistances are visible. These spikes correspond to a person breathing directly onto the sensor, indicating the sensor functions as expected.

Furthermore, the fluctuations in temperature correspond to the window of the room where the sensor is located being opened (where temperature drops occur) and closed (where the temperature starts to rise again).

Finally, the large drop in gas resistances towards the end of the experiment correspond to aromatic food products being placed in the same room as the sensor.

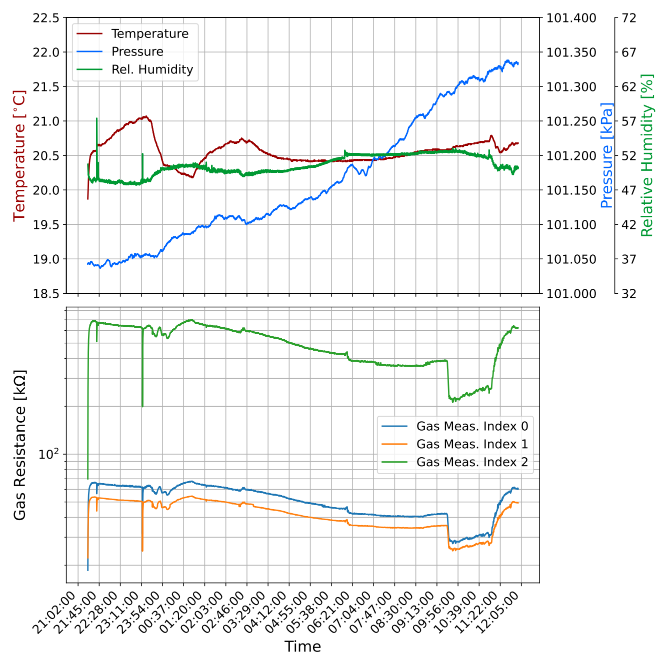


Figure 11: Environmental Data Plot

5.1.2. Experiment II

In this experiment, three different chemical substances were put in close proximity to the BME688. The chemicals used for this experiment are alcohol, bleach, and ammonia respectively. In Figure 12 the results are shown. Every large drop in gas resistance indicates a new substance being held in close proximity. The substances were used in high concentrations to showcase a drastic change in the AEVs. This did in turn significantly influence the relative humidity and temperature due to the large amount of evaporation occurring.

Every chemical has a unique gas-resistance footprint as can be seen in Figure 12. A classifier can be developed by taking known substances and extracting their gas-resistance footprint. This is made possible by using the Bosch Sensortec software [27], though integrating this software is left for future design iterations. In these experiments, a gas-heater profile of three steps was used, but up to ten steps are available on the sensor, allowing for a more detailed footprint to be established for different chemicals.

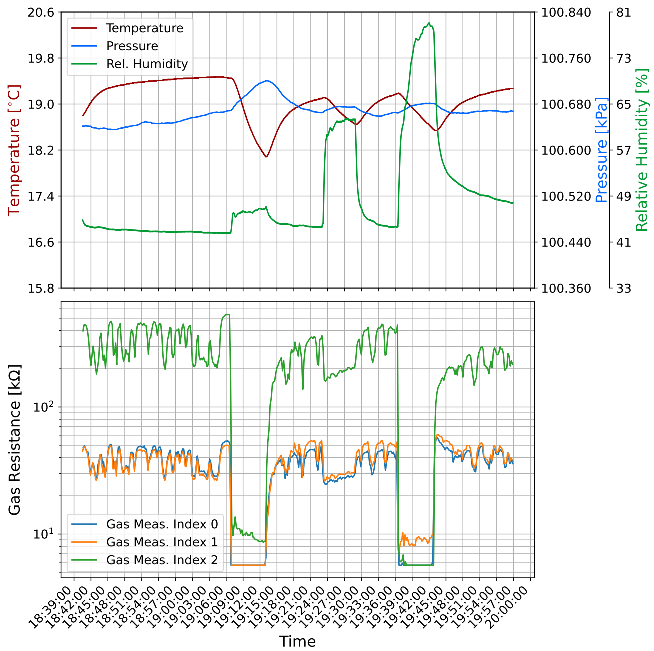


Figure 12: Gas Test Data Plot

5.2. System State Data

The system state data consists of both the position, as well as the attitude. Both serve an important purpose, with the GNSS data providing the user with the information required to pick up the sensor either by hand or by drone, while the attitude data can be used to determine what direction the microphone and radio antenna are pointed in.

5.2.1. GNSS Performance Analysis

Figure 13 shows the absolute error of the GNSS signal over time in both horizontal and vertical directions, as well as the HDOP values corresponding to the measurements. The measurements were taken from a stationary position on the balcony of a building on the 7th floor, about 25 meters from the ground with a 30 second interval.

The samples are filtered using exponential smoothing, according to Equation 2, in order to retain an estimate of the positioning without the sudden fluctuations.

$$
\begin{aligned}
s_0 &= x_0 \\
s_t &= \lambda x_t + (1 - \lambda) s_{t-1}, \quad \lambda \in (0, 1)
\end{aligned}
\tag{2}
$$

The cause for these fluctuations can have many different causes, ranging from unfavourable satellite geometry to small changes in the propagation delay [28].

Unfortunately, no Vertical Degree of Precision (VDOP) or Geometric Degree of Precision (GDOP) values are provided by the chip. Lower HDOP values indicate a better estimate of the lateral positioning, a statement that can be verified by inspecting the lateral estimate offset and the HDOP signals: a clear correlation exists between the HDOP signal and the lateral estimate.

The accuracy of the GNSS system during these experiments is summarized in Table 3. The majority of measurements have a HDOP value under 2%, which is considered an excellent estimate as stated by Isik et al. (2020) [29]. Furthermore, the lateral offset is practically bounded by 40 meters, and vertical offset by 20 meters, with $2/3$ measurements being within 20 meters laterally and 10 meters vertically.

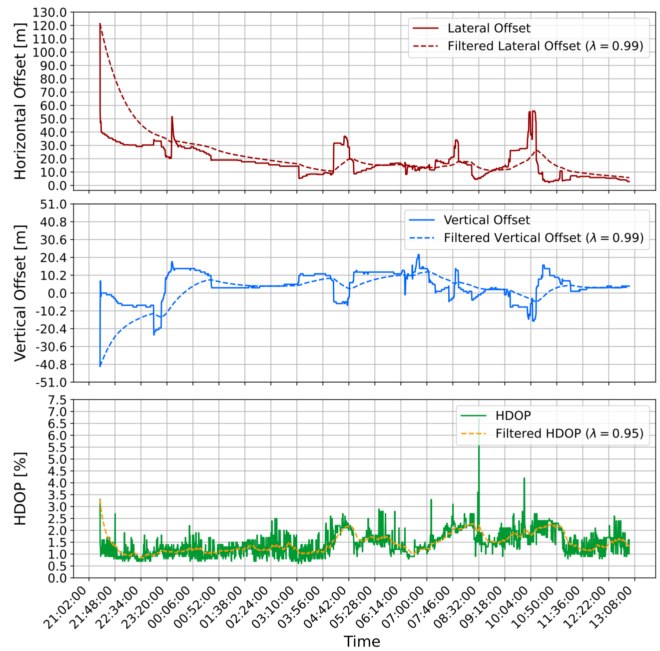


Figure 13: Horizontal- and Vertical Offsets in meters, together with the provided HDOP values

TABLE 3: GNSS Measured Accuracy Metrics

Distances	< 10 m	< 20 m	< 30 m	< 40 m
Lateral	27.8%	69.6%	83.4%	98.3%
Vertical	66.5%	98.2%	99.9%	99.9%

Dilution	$< 1\%$	$< 2\%$	$< 3\%$	$< 4\%$
HDOP	18.1%	82.8%	99.6%	99.8%

5.2.2. IMU Performance Analysis

The IMU's coordinate system is shown in Figure 14, with positive rotations about the X, Y and Z axes corresponding to positive roll (ϕ), pitch (θ), and yaw (ψ). With these definitions, it is important to remember that a pitch-down is a positive rotation about the Y-axis, and thus yields a positive pitch angle θ.

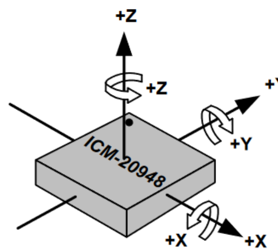


Figure 14: IMU Coordinate System [30]

The test result is shown in Figure 15. During the test, the IMU is first allowed to settle. In particular the heading angle ψ requires some time to settle due to the magnetometer within the IMU requiring calibration. After the signal has settled, the roll- and pitch angles, ϕ and θ respectively, are rotated by approximately $45°$ in both positive and negative directions. Then, the heading angle is rotated by approximately $90°$ in both directions. All rotations were done while attempting to keep the other angles near $0°$. After the separate rotations, the whole IMU is rotated around all axes at about $30°$. The logged data corresponds to the movements made during the test, verifying its correct functioning.

Furthermore, the $\psi = 0°$, which should correspond to magnetic north, is off by $2-5°$ from a smartphone compass. The magnetometer used in the smartphone is the AK09911, while the magnetometer used within the ICM20948 is the newer AK09916. The measurements being within a few degrees of one another is acceptable performance, though it is difficult to quantitatively compare these results as either could be off by some degrees, and magnetic readings can easily be influenced by electromagnetic interference.

5.3. Wireless Communication Performance

The long-range radio is the only means for the user to communicate with the sensors deployed in the field, and as such it is important to have a reliable up- and down-link. The LoRa radio has multiple parameters, such as Spreading Factor, Coding rate and Bandwidth, that can be customized depending on the application of the radio and any regulations surrounding the use of the sub-GHz band in the region where the radio is to be deployed. The most common of these restrictions are regarding the TOA, signal power, bandwidth, and may require the usage of Frequency-Hopping Spread Spectrum (FHSS), Listen-Before-Talk (LBT), and/or Adaptive Frequency Agility (AFA). All the regulations for the European Union are set-up by the European Research Council (ERC), Electronic Communications Committee (ECC) and European Telecommunications Standards Institute (ETSI) [22], [23], though even within the EU countries may deviate from these regulations.

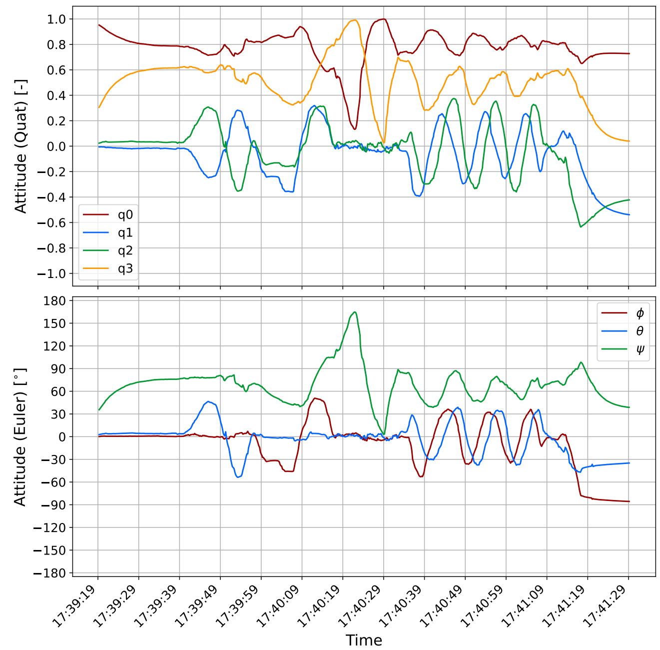


Figure 15: IMU Sample Data Plot

Other than following regulations, the link must be able to robustly operate to be used within the monitoring system. Received wireless signals have certain properties that help categorize the quality of the signal between two transceivers, such as SNR, RSSI, and the frequency deviation or frequency error, Δf.

In the following experiments, the two most influential parameters, Spreading Factor (SF) and Bandwidth (BW), were chosen to be analyzed in more detail together with practical range measurements in order to be able to determine the optimal selection of both of these parameters. Both these parameters influence the symbol-rate R_s according to:

$$R_s = \frac{BW}{2^{SF}} \quad \left[\frac{\text{symb}}{s}\right] \tag{3}$$

For every experiment, the SNR, RSSI, Δf, TOA, and distance between transceivers are logged. The experiments are performed by placing one transceiver on the balcony on the 7th floor, and the other moving around the park out front of the building. During the experiment, the moving transceiver finds itself in a forested area where no line-of-sight is guaranteed. Furthermore, the park finds itself in an urban area, meaning there is electromagnetic interference. A carrier frequency of 865.730 MHz is used during all of the experiments. An experiment utilizing a carrier frequency of 433 MHz was performed as well, however due to the electromagnetic interference present this band was rendered unusable and thus not shown here.

5.3.1. Effect of Spreading Factor

LoRa is a form of a Chirp Spread Spectrum (CSS) modulated signal, meaning that bits are sent through chirps. A chirp is when the signal moves up or down the frequency spectrum, usually in a linear fashion. The Spreading Factor (SF) of a LoRa signal is best explained using Figure 16. Higher SF values indicate a longer chirp, which take longer to transmit but in return yield a much more robust signal that require a lower SNR to demodulate the signal, as described in Table 4. In return, higher SF signals use more total power to transmit the same message, and the bitrate is much lower [31].

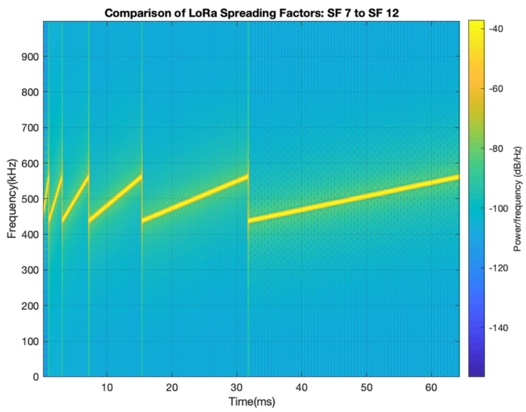


Figure 16: Visualization of Spreading Factor [32]

TABLE 4: Minimum SNR levels per SF [21]

SF	6	7	8	9	10	11	12
Min. SNR [dB]	-5	-7.5	-10	-12.5	-15	-17.5	-20

The results of the experiment are shown in Figure 17 for a selection of three different SF values and a bandwidth fixed at 250 kHz. The minimum SNR levels described in Table 4 closely follow the trends shown in Figure 17. Some signals are received with SNR levels below the defined minimum, though these are the exception rather than the rule.

5.3.2. Effect of Bandwidth

The signal bandwidth, when increased, results in a higher bitrate, but will reduce the receiver sensitivity, in turn reducing range. As such, to maximize the range, the bandwidth must be made more narrow, while trying to maximize bitrate or minimize TOA, the bandwidth must be widened. In Figure 18 the results of the experiment are shown, with the SF fixed at 9.

As expected, the lower bandwidth signals suffer less from signal degradation over larger distances and reach the farthest. A bandwidth of 62.5 kHz is the smallest possible bandwidth that can be used when no Temperature Compensated Crystal Oscillator (TCXO) is used, as temperature differences between the radio chips result in larger frequency deviations. As a rule of thumb [21]:

$$\Delta f_{\max}(BW) = \pm\frac{BW}{4} \tag{4}$$

Meaning that for a bandwidth of 32.25 kHz, a maximum frequency deviation of ±8.06 kHz is allowed. Inspecting

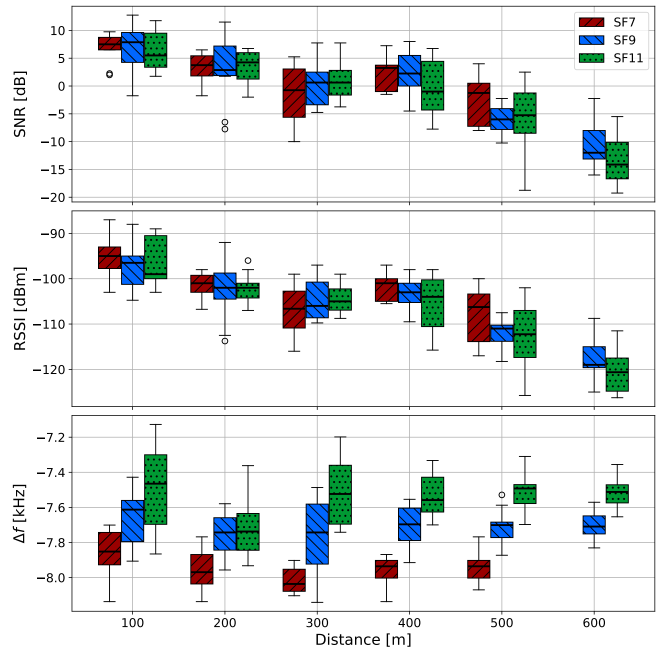


Figure 17: Influence of SF on wireless signal

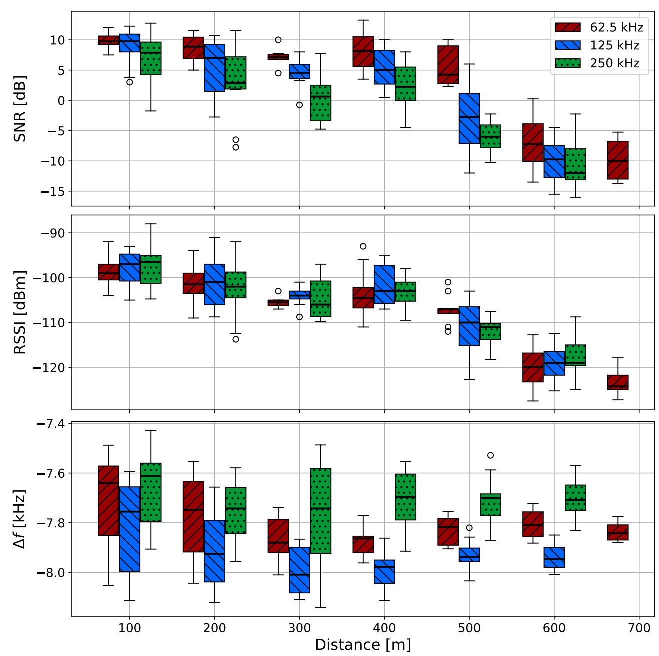


Figure 18: Influence of bandwidth on wireless signal

the Δf plots of both Figure 17 and Figure 18, this cannot be guaranteed with the current setup.

5.3.3. Resulting Time-on-Air

From both experiments, the resulting TOA for different messages is logged. The TOA can be calculated analytically using the following formulae described in the datasheet [21]. First, a place-holder function, $T(M)$, encapsulating the effect of message size M in bytes, is defined as:

$$T(M) = \left\lceil \frac{8M - 4 \cdot \text{SF} + 16 \cdot \text{CRC} - 20 \cdot \text{IH} + 28}{4 \cdot (\text{SF} - 2 \cdot \text{DE})} \right\rceil \quad (5)$$

where SF is the spreading factor, CRC is 1 if Cyclic Redundancy Check (CRC) is used, IH is 0 if a header is present, and DE is 1 if low-data rate optimization is active. Using this, the number of payload symbols can be calculated as:

$$N_{\text{payload}}(M) = 8 + \max\left[T(M) \cdot (\text{CR} + 4), 0\right] \quad (6)$$

where CR is the Coding Rate, a measure for the number of redundant bits present per 4 bits. The number of preamble symbols is determined by the developer and can be set to any number between 6 and 2^{16}, to which 4.25 is added:

$$N_{\text{preamble}} = n_{\text{preamble}} + 4.25 \quad (7)$$

Finally, the TOA is calculated by combining the above equations with Equation 3:

$$\text{TOA}(M) = \frac{N_{\text{payload}}(M) + N_{\text{preamble}}}{R_s} \quad (8)$$

The TOA curves for varying payload sizes and for different combinations of spreading factors and bandwidths are shown in Figure 19. The analytical model and the measured experimental TOA values coincide, with deviations in the order of microseconds.

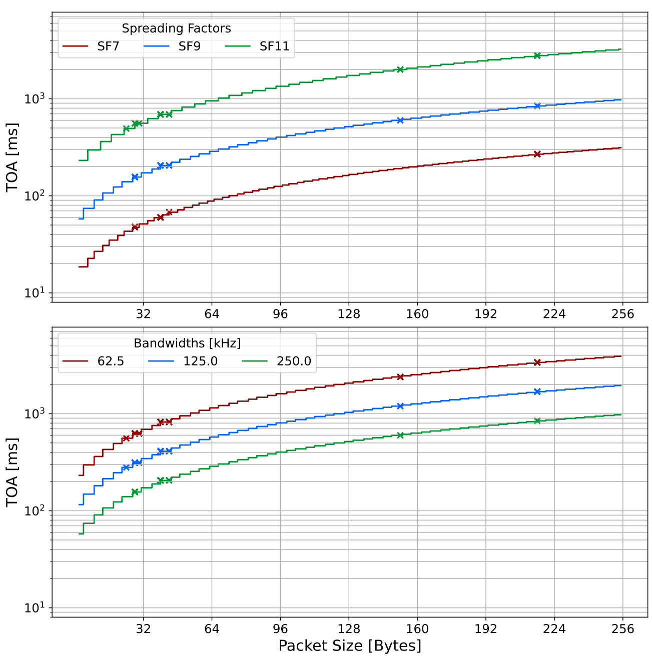


Figure 19: TOA: Analytical and Experimental

From the aforementioned formulae and the plot in Figure 19, the effect of bandwidth and SF on the TOA can clearly be seen. Larger spreading factors introduce larger transmission times, while larger bandwidths reduce the required transmission time instead. Furthermore, the analytical formula can be used to predict the TOA in order to adhere to any regulations that limit the TOA or impose a duty cycle on transmissions.

5.4. Audio Data & Classifier Performance

The collection of BEVs is a crucial component of the system, and in this case this comes in the form of audio and an on-board classifier. To test the performance, an experiment to test the effective range of the microphone and the corresponding classifier performance is executed. In particular, a portable speaker with a recording of a sedge warbler is positioned at varying distances from the microphone. At every distance the speaker volume is kept at the same level and the same recording is played back. At known times the recording stops and starts, and within these times the classifier should theoretically be able to detect the sedge warbler at every 3-second segment, as there are no silent gaps over 1 second long. The result of this experiment is shown in Figure 20, with the audio waveform and Mel-spectrum shown in Figure 22. The sudden spikes are purposefully made by clapping hands together close to the microphone, to indicate the transition to a new distance.

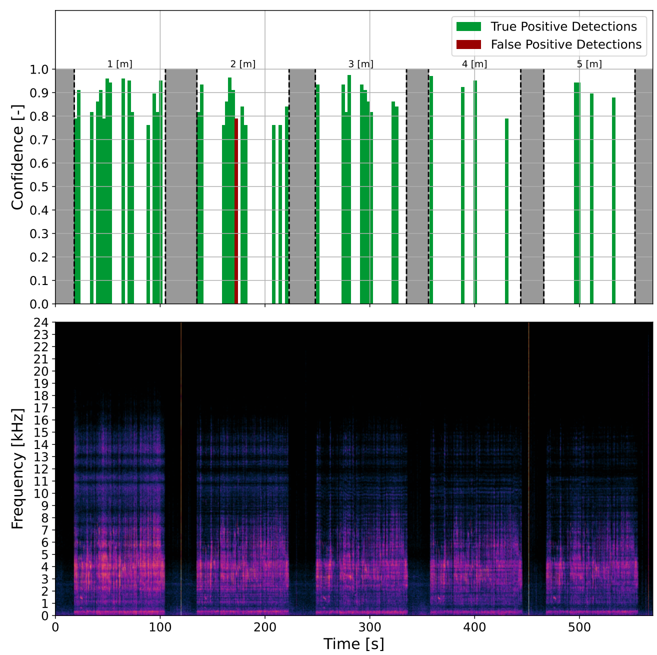


Figure 20: BirdNet Detections

The ratio of the number of 3-second segments where a detection was made over the total number of 3-second segments is given in Table 5 for every distance. Unlike the theoretical value of obtaining a 100% detection rate at close distances, a maximum rate of only 51.72% is reached at 1 meter distance. A likely cause is the relatively low volume of the recording. By amplifying the audio signal, the detection rate is likely significantly improved, particularly for larger distances.

5.5. Multi-Threaded Performance

Ensuring the system can run efficiently and predictably is important to ensure data integrity and consistency. For

TABLE 5: BirdNet detection rate with distance

Distance [m]	1	2	3	4	5
Detection Rate [%]	51.72	37.5	34.48	13.64	13.79

TABLE 6: Thread Performance Overview, all units in ms

Thread	**Default Period**	**Min**	**Max**	μ	$\bar{\mu}$	σ
BirdNet	3000	0.1647	2661.9465	2413.6302	2471.6807	379.0780
PRU Manager	500	0.0139	101.1554	10.1735	0.0282	21.1017
LoRa	100	0.6594	56.8799	0.9874	0.7253	3.2804
GNSS	2000	0.0095	1013.6886	0.9215	0.0252	21.5295

this purpose, the threads are traced such that any timing issues can be identified and the overall performance of the thread-scheduler can be quantified. In Figure 21 a sample of the thread trace is shown with thread preemptions, ranked according to thread priority. Moreover, the statistics are collected and summarized in Table 6, where for every thread the standard constant period is given, followed by the real minimum and maximum recorded thread period, the mean and median thread period, and the standard deviation of all timing samples collected.

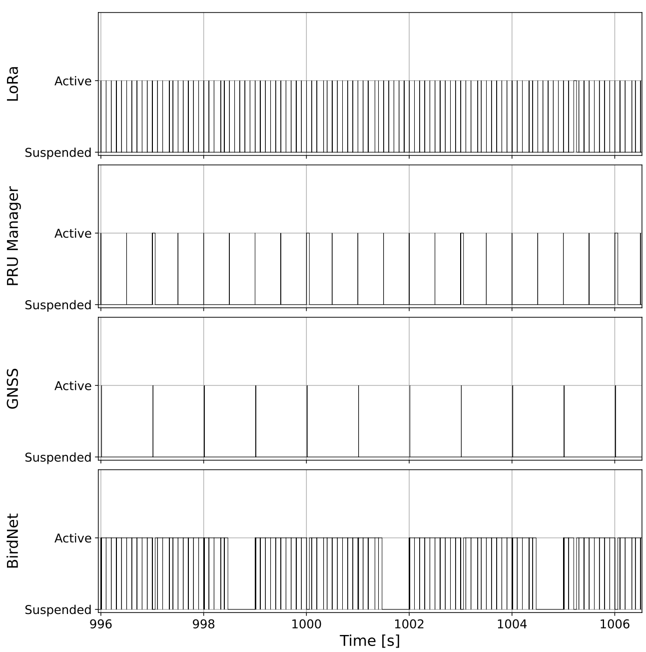


Figure 21: Thread Trace Sample Plot

Inspecting Table 6, the worst-case thread execution times fall within the period of every thread. Moreover, the mean- and median thread execution times are considerably lower than the worst-case execution times, leading to the conclusion that the system could potentially run at faster rates, i.e. more GNSS samples can be taken, and the LoRa thread could run at an even smaller period to allow for faster radio response times. It should be noted that by reducing thread periods, the number of context-switches will increase as well, which may reduce the overall performance of the system, or worse, result in an infeasible task schedule. Instead of attempting to speed up the current system, the system could be extended instead by adding more sensors or processing more of the collected data on-board.

Conclusion

The goal of the project, ie. ”The design and development of a cheap sensor node capable of reliably measuring ecological variables, perform inference on said variables, and relay any data and/or findings over a wireless network with little to no infrastructure requirements, all while being lightweight and small enough to be deployable by drones”, can be said to have been achieved with this project. All requirements listed in Appendix B have been met, with a few exceptions which are to be addressed in the recommendations section.

Returning to the five main points of improvement listed in section 3, we can assert whether the goals have been met and to what extent. The first goal is a reduction in price, actualized through non-functional requirement **NR-4**. As discussed in subsubsection 4.3.4, while the networked version exceeds the cost-limit set in non-functional requirement **NR-4**, the non-networked version would be below the cost limit. It should be noted however that these costs will be lower if the system is to be produced in larger numbers.

The second goal is the improvement of data reliability, actualized through functional requirements **FR-4** to **FR-6**. This is achieved through the use of the SBC’s PRU-cores working in parallel with the ARM core. The PRUs allow for consistent sampling irrespective of the ARM core’s current tasks due to the consistent clock frequency of the PRUs and absence of delay-inducing effects such as context switching or cache misses. With the AEV-collecting sensors on one of the PRU cores and the audio collection done by the other PRU core, all the environmental data is sampled at very consistent rates.

The third goal addressed by this system is the robust wireless capabilities to allow data and detections to be relayed to a central hub while the sensors are still out in the field, with relevant requirement functional requirement **FR-3**. As demonstrated, the board is able to broadcast messages in electromagnetically noisy environments and without direct line of sight up to 600 meters given the appropriate SF and bandwidth settings. While this is not an appropriate up-link for data streaming, it is sufficient for control messages and infrequent data packets containing sampled AEVs and/or BEVs. Moreover, while the LTE-M / NB-IoT networking capabilities are currently unused, the module is present, which provides the possibility for real-time data streaming.

The next goal stated is the ability to perform inference on the collected data on-board in (near) real-time. The related requirement is functional requirement **FR-7**. The experiments demonstrated that the system is able to continuously record audio and perform inference on the audio in buffers of 3-seconds long. The inference takes a shorter amount of

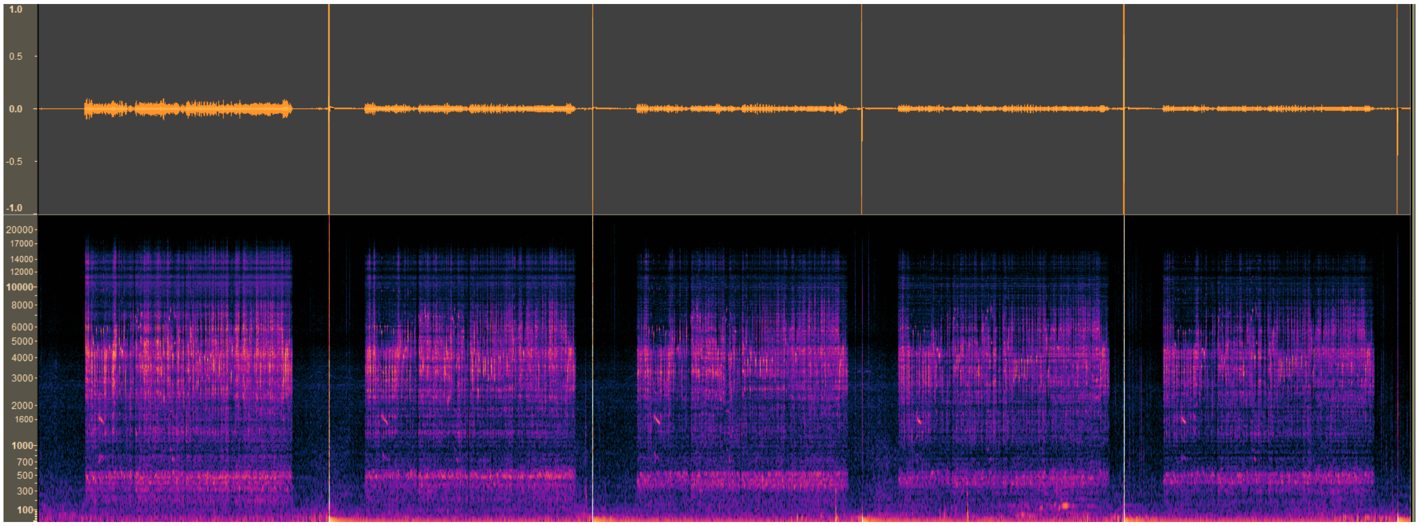


Figure 22: Recorded Audio Sample: Screenshot from Audacity

time than the input-length of the audio signal, meaning a continuous audio analysis is possible.

The final goal is the deployability of the system with drones. This goal received less attention than the others, as it required too much time to be worked out in detail. This goal was kept in mind throughout the whole design process however, with functional requirements **FR-2** and **FR-13** and non-functional requirements **NR-2** to **NR-3** being relevant to this goal. The details of realizing this goal however are left for future design iterations as this was not feasible within the given timeline for this project.

Recommendations

Given the experimental results and experience with using the system to perform the experiments, the following set of improvements to the system and valuable additions are suggested. Some of the requirements listed in Appendix B are not yet met due to a shift in focus of the project or a lack of time to develop the necessary features in a timely manner. These requirements will also be mentioned and addressed in this section.

Firstly, in terms of overall system improvements and additions, the development of a drone attachment and deployment mechanism to reduce the time spent in the field is essential. To do so, functional requirement **FR-2** must be met as well, which currently is not the case.

Additionally, integrating the software within an RTOS could further optimize system performance and responsiveness, considering that while the threads run within their timing constraints, certain features can currently not effectively be used. For example, the lag-time between an interrupt signal coming in on a GPIO pin and the software handling the interrupt is currently much too large and limited by the interrupt polling-speed and the scheduling of the thread assigned to handling the interrupt.

Regarding wireless communication, upgrading to a more capable LoRa chip that allows for simultaneous transmission and reception would be beneficial and significantly simplify the implementation of mesh-networking, in-turn enhancing network resilience and coverage. Moreover, utilizing a TCXO for the LoRa chip would allow the usage of smaller bandwidths and thus increase the maximum potential range. Implementing FHSS can improve signal stability and reliability, and is a requirement within certain regions to be allowed to operate, satisfying functional requirement FR-3.1.3. Consideration should also be given to leveraging LTE-M network technology for specific communication requirements, particularly given the fact that the hardware is already present, and just needs to be interfaced with together with a SIM-card that can access the LTE-M network.

In terms of advancing sensor integration and utilizing the system to its fullest extent, integrating a gas classifier would allow for more AEVs to effectively be measured, significantly enhancing the environmental monitoring capabilities. For example, detecting smoke or high concentrations of carbon mono- or dioxide would be possible through a custom-trained gas classifier, and doing so would satisfy the functional requirement FR-8. Furthermore, upgrading to more accurate sensors for temperature, pressure, and humidity can improve data accuracy and reliability, as the BME688 influences these measurements when taking gas-measurements with the heaters and thus may lead to erroneous measurements, particularly when sampling at higher frequencies.

Regarding audio classifiers, exploring the use of alternative classifiers such as those designed for bat detection in combination with the designed ultrasonic sound sampler can extend the system's use-cases.

Overall, these recommendations aim to address current shortcomings of the system as well as the unmet requirements, and introduce performance-enhancing features which could increase the potential value of the system to the end-user significantly.

Acknowledgments

The author would like to thank Dr. Ir. Salua Hamaza and Dr. Raj Thilak Rajan for their guidance, patience and time during this project. Furthermore, special thanks to the members of the BeagleBoard Forum, in particular Robert C. Nelson, for helping out with many of my questions regarding the hardware platform.

References

[1] H. Lee, J. Romero, and Core Writing Team IPCC, "AR6 Synthesis Report: Climate Change 2023," Mar. 2023. [Online]. Available: https://www.ipcc.ch/report/ar6/syr/

[2] C. D. Keeling, R. B. Bacastow, A. E. Bainbridge, C. A. Ekdahl Jr., P. R. Guenther, L. S. Waterman, and J. F. S. Chin, "Atmospheric carbon dioxide variations at Mauna Loa Observatory, Hawaii," *Tellus*, vol. 28, no. 6, pp. 538–551, Jan. 1976, publisher: Taylor & Francis. [Online]. Available: https://doi.org/10.3402/tellusa.v28i6.11322

[3] G. Likens and D. Lindenmayer, *Effective Ecological Monitoring*. Csiro Publishing, May 2018, google-Books-ID: QDBZDwAAQBAJ.

[4] M. Besson, J. Alison, K. Bjerge, T. E. Gorochowski, T. T. Høye, T. Jucker, H. M. R. Mann, and C. F. Clements, "Towards the fully automated monitoring of ecological communities," *Ecology Letters*, vol. 25, no. 12, pp. 2753–2775, 2022. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/ele.14123

[5] K. Cao, Y. Liu, G. Meng, and Q. Sun, "An Overview on Edge Computing Research," *IEEE Access*, vol. 8, pp. 85 714–85 728, 2020, conference Name: IEEE Access. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9083958

[6] G. M. Lovett, D. A. Burns, C. T. Driscoll, J. C. Jenkins, M. J. Mitchell, L. Rustad, J. B. Shanley, G. E. Likens, and R. Haeuber, "Who needs environmental monitoring?" *Frontiers in Ecology and the Environment*, vol. 5, no. 5, pp. 253–260, 2007. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1890/1540-9295%282007%295%5B253%3AWNEM%5D2.0.CO%3B2

[7] "Tramex Remote Environmental Monitoring System 5 | TREMS-5." [Online]. Available: http://tramexmeters.com/kits/remote-environmental-monitoring-system/remote-environmental-monitoring-system-5

[8] "Wildlife acoustics: Wildlife Audio Recording Equipment." [Online]. Available: https://www.wildlifeacoustics.com/

[9] R. C. Whytock and J. Christie, "Solo: an open source, customizable and inexpensive audio recorder for bioacoustic research," *Methods in Ecology and Evolution*, vol. 8, no. 3, pp. 308–312, 2017, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/2041-210X.12678. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/2041-210X.12678

[10] R. D. Beason, R. Riesch, and J. Koricheva, "AURITA: an affordable, autonomous recording device for acoustic monitoring of audible and ultrasonic frequencies," *Bioacoustics*, vol. 28, no. 4, pp. 381–396, Jul. 2019, publisher: Taylor & Francis _eprint: https://doi.org/10.1080/09524622.2018.1463293. [Online]. Available: https://doi.org/10.1080/09524622.2018.1463293

[11] S. S. Sethi, R. M. Ewers, N. S. Jones, C. D. L. Orme, and L. Picinali, "Robust, real-time and autonomous monitoring of ecosystems with an open, low-cost, networked device," *Methods in Ecology and Evolution*, vol. 9, no. 12, pp. 2383–2387, 2018, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/2041-210X.13089. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/2041-210X.13089

[12] A. P. Hill, P. Prince, E. Piña Covarrubias, C. P. Doncaster, J. L. Snaddon, and A. Rogers, "AudioMoth: Evaluation of a smart open acoustic device for monitoring biodiversity and the environment," *Methods in Ecology and Evolution*, vol. 9, no. 5, pp. 1199–1211, 2018, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/2041-210X.12955. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/2041-210X.12955

[13] A. P. Hill, P. Prince, J. L. Snaddon, C. P. Doncaster, and A. Rogers, "AudioMoth: A low-cost acoustic device for monitoring biodiversity and the environment," *HardwareX*, vol. 6, p. e00073, Oct. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2468067219300306

[14] CorBer, "Teensy_batdetector (update 2023 03 17)," Feb. 2023, original-date: 2020-07-27T12:07:38Z. [Online]. Available: https://github.com/CorBer/teensy_batdetector

[15] Edwin Houwertjes and Cor Berrevoets, "Home | teensybat.com," 2023. [Online]. Available: https://www.teensybat.com/

[16] S. Hamaza, I. Georgilas, M. Fernandez, P. Sanchez, T. Richardson, G. Heredia, and A. Ollero, "Sensor Installation and Retrieval Operations Using an Unmanned Aerial Manipulator," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2793–2800, Jul. 2019, conference Name: IEEE Robotics and Automation Letters.

[17] S. Hamaza, A. Farinha, H.-N. Nguyen, and M. Kovac, "Sensor Delivery in Forests with Aerial Robots: A New Paradigm for Environmental Monitoring," Nov. 2020.

[18] B. B. Kocer, B. Ho, X. Zhu, P. Zheng, A. Farinha, F. Xiao, B. Stephens, F. Wiesemüller, L. Orr, and M. Kovac, "Forest Drones for Environmental Sensing and Nature Conservation," in *2021 Aerial Robotic Systems Physically Interacting with the Environment (AIRPHARO)*, Oct. 2021, pp. 1–8.

[19] "BeagleBone Black — BeagleBoard Documentation." [Online]. Available: https://docs.beagleboard.org/latest/boards/beaglebone/black/index.html

[20] S. Kahl, C. M. Wood, M. Eibl, and H. Klinck, "BirdNET: A deep learning solution for avian diversity monitoring," *Ecological Informatics*, vol. 61, p. 101236, Mar. 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574954121000273

[21] SemTech, "SX1276." [Online]. Available: https://www.semtech.com/products/wireless-rf/lora-connect/sx1276

[22] E. (ETSI), "300 220-1 (v2. 4.1): Electromagnetic compatibility and Radio spectrum Matters (ERM); Short Range Devices (SRD)," May 2012. [Online]. Available: https://www.etsi.org/deliver/etsi_en/300200_300299/30022001/02.04.01_60/en_30022001v020401p.pdf

[23] Electronic Communications Committee, "ERC/REC 70-03," Jun. 2023. [Online]. Available: https://docdb.cept.org/document/845

[24] *AM335x Sitara Processor Technical Reference Manual*, Texas Instruments, 12 2019, revision Q. [Online]. Available: https://www.ti.com/lit/ug/spruh73q/spruh73q.pdf

[25] *AM335x Sitara Processor Datasheet*, Texas Instruments, 3 2020, revision L. [Online]. Available: https://www.ti.com/lit/ds/symlink/am3358.pdf?ts=1707100008342

[26] Bosch, "Gas sensor BME688." [Online]. Available: https://www.bosch-sensortec.com/products/environmental-sensors/gas-sensors/bme688/

[27] ——, "BME688 SensorTec Software." [Online]. Available: https://www.bosch-sensortec.com/software-tools/software/bme688-software/

[28] P. Ranacher, R. Brunauer, W. Trutschnig, S. V. d. Spek, and S. Reich, "Why GPS makes distances bigger than they are," *International Journal of Geographical Information Science*, Feb. 2016, publisher: Taylor and Francis. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/13658816.2015.1086924

[29] O. K. Isik, J. Hong, I. Petrunin, and A. Tsourdos, “Integrity Analysis for GPS-Based Navigation of UAVs in Urban Environment,” *Robotics*, vol. 9, no. 3, p. 66, Sep. 2020, number: 3 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/2218-6581/9/3/66

[30] TDK InvenSense, “ICM-20948 Datasheet.” [Online]. Available: https://invensense.tdk.com/download-pdf/icm-20948-datasheet/

[31] Semtech, “LoRa and LoRaWAN: Technical overview | DEVELOPER PORTAL.” [Online]. Available: https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lora-and-lorawan/

[32] D.-H. Kim, E.-K. Lee, and J. Kim, “Experiencing LoRa Network Establishment on a Smart Energy Campus Testbed,” *Sustainability*, vol. 11, p. 1917, Mar. 2019.

[33] H.-N. Nguyen, R. Siddall, B. Stephens, A. Navarro-Rubio, and M. Kovač, “A Passively Adaptive Microspine Grapple for Robust, Controllable Perching,” in *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*, Apr. 2019, pp. 80–87.

[34] M. Kovac, “Learning from nature how to land aerial robots,” *Science*, vol. 352, no. 6288, pp. 895–896, May 2016, publisher: American Association for the Advancement of Science. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.aaf6605

[35] R. Berto, P. Napoletano, and M. Savi, “A LoRa-Based Mesh Network for Peer-to-Peer Long-Range Communication,” *Sensors (Basel, Switzerland)*, vol. 21, no. 13, p. 4314, Jun. 2021. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8272137/

[36] Y. Triwidyastuti, M. Musayyanah, F. Ernawati, and C. D. Affandi, “Multi-hop Communication between LoRa End Devices,” *Scientific Journal of Informatics*, vol. 7, no. 1, pp. 125–135, Jun. 2020. [Online]. Available: https://journal.unnes.ac.id/nju/index.php/sji/article/view/21855

[37] J. Pérez, J. Díaz, J. Berrocal, R. López-Viana, and Á. González-Prieto, “Edge computing,” *Computing*, vol. 104, no. 12, pp. 2711–2747, Dec. 2022. [Online]. Available: https://doi.org/10.1007/s00607-022-01104-2

[38] J. Pan and J. McElhannon, “Future Edge Cloud and Edge Computing for Internet of Things Applications,” *IEEE Internet of Things Journal*, vol. PP, pp. 1–1, Oct. 2017.

[39] S. Kahl, “BirdNET-Analyzer,” Sep. 2023, original-date: 2021-09-22T13:29:56Z. [Online]. Available: https://github.com/kahst/BirdNET-Analyzer

[40] S. Kestur, J. D. Davis, and O. Williams, “BLAS Comparison on FPGA, CPU and GPU,” in *2010 IEEE Computer Society Annual Symposium on VLSI*, Jul. 2010, pp. 288–293, iSSN: 2159-3477. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/5572788

[41] Tramex, “Tramex Remote Environmental Monitoring System 5 | TREMS-5.” [Online]. Available: http://tramexmeters.com/kits/remote-environmental-monitoring-system/remote-environmental-monitoring-system-5

Acronyms

ADC Analog-to-Digital Converter. 10, 11, 13
AEV Abiotic Ecologial Variable. 4–7, 13, 18, 19, 25–27
AFA Adaptive Frequency Agility. 15
API Application Programming Interface. 11

BEV Biotic Ecological Variable. 4–7, 17, 18, 23, 25–27
BOM Bill Of Materials. 11
BW Bandwidth. 15

CPU Central Processing Unit. 12, 25, 26
CRC Cyclic Redundancy Check. 17
CSS Chirp Spread Spectrum. 15

DMP Digital Motion Processor. 7, 13
DRAM Data RAM. 9
DSP Digital Signal Processor. 25, 26

ECC Electronic Communications Committee. 15
EM Ecological Monitoring. 3–6
eMMC Embedded Multi Media Card. 7, 9
ERC European Research Council. 15
ETSI European Telecommunications Standards Institute. 15

FHSS Frequency-Hopping Spread Spectrum. 15, 19
FIFO First-In First-Out. 8, 13
FLOPS Floating-Point Operations Per Second. 26
FPGA Field-Programmable Gate Array. 25, 26

GDOP Geometric Degree of Precision. 14
GNSS Global Navigation Satellite System. 7, 8, 10–12, 14, 18, 26
GPIO General-Purpose Input-Output. 9, 10, 19

HDOP Horizontal Degree of Precision. 8, 14

I^{2}C Inter-Integrated Circuit. 9–13
I^{2}S Inter-Integrated Circuit Sound. 9, 10, 13
IMU Inertial Measurement Unit. 7, 8, 11–15, 26
IPCC Intergovernmental Panel on Climate Change. 3
IRAM Instruction RAM. 9, 12, 13
ISM Industrial-Scientific-Medical. 7

LBT Listen-Before-Talk. 15, 24
LoRa Long-Range. 8, 10–12, 15, 16, 18, 19, 24–26
LoRaWAN Long-Range Wide Area Network. 24
LTE-M Long Term Evolution - Machine Type Communication. 7, 10, 11, 18, 19, 25

McASP Multichannel Audio Serial Port. 10
MCU Micro-Controller Unit. 4, 5
MEMS Micro-Electromechanical System. 10
MPU Micro-Processor Unit. 4, 5

NB-IoT Narrowband IoT. 7, 10, 11, 18

ODR Output Data Rate. 13, 25
OS Operating System. 11

P2P Peer-to-Peer. 24, 26
PCB Printed Cicruit Board. 5, 10, 11
PRU Programmable Real-Time Unit. 7–9, 11–13, 18

RAM Random Access Memory. 7, 9
RSSI Received Signal Strength Indicator. 8, 15
RT Real-Time. 5, 11, 12
RTC Real-Time Clock. 7, 9, 11
RTOS Real-Time Operating System. 11, 19

SBC Single-Board Computer. 5–7, 9–11, 18

SDXC Secure Digital Extended Capacity. 7, 9
SF Spreading Factor. 15–18
SNR Signal-to-Noise Ratio. 8, 15, 16
SPI Serial Peripheral Interface. 9–11

TCXO Temperature Compensated Crystal Oscillator. 16, 19
TDM Time-Division Multiplexing. 9, 10
TOA Time-On-Air. 8, 15–17, 24

UART Universal Asynchronous Receiver / Transmitter. 8–10, 12
USB Universial Serial Bus. 7

VDOP Vertical Degree of Precision. 14

WOM Wake-On-Motion. 7, 8, 12

Appendix A.
Product Vision

In this chapter, the product concept is outlined. In particular, the identified gap in the market is translated into a product concept which is to be developed into a functioning prototype. The development of this concept is split into multiple steps. Firstly, the stakeholders are identified and classified in order to better understand what to prioritize in the design. After the stakeholder analysis the concept exploration phase is presented. In this phase all ideas and concepts are pooled together in order to identify potential solutions at a high level. This is followed by a brief feasibility study, where the best concept is taken and analyzed from multiple standpoints, including economic, political, legal and environmental standpoints. Finally, the preferred concept is then expanded upon by detailing its concept of operation: a high-level description of a single mission cycle of the product.

A.1. Stakeholder Analysis

For the system, a total of five stakeholder groups are identified: Researchers, Conservationists, (Local) Governments, Hobbyists, and Local Inhabitants. A simple overview of every group's needs, contributions and potential obstructions is summarized in Table 7. From this overview, it is clear that it is important to prioritize cost and reliability in order to gain the favour of researchers, conservationists, and any potential government willing to sponsor the use of this system.

A.2. Concept Exploration

Before the design of the system commences, a concept exploration phase is needed to identify feasible designs, components and functionalities. There are five main categories where critical design choices are to be made. These categories are the level of integration with the delivery/retrieval drones, the primary BEV sensor(s), the wireless networking architecture, the computational- and data-flows, and finally, the type of hardware everything is based on- and connected to.

A.2.1. Drone Integration Level

The drone integration level determines whether the sensors are physically integrated with the drone and use the drone's resources such as memory and processor (high integration level), if the sensors have their own resources but are still physically attached to the drone throughout the mission duration (medium integration level), or if the sensor and drones are entirely separated and thus the drone only serves as delivery/retrieval equipment (low integration level). Instead, for medium and high integration levels, the drone will land/perch at the desired location and start the data logging.

The high- and medium levels of integration have the advantage of being nearly fully autonomous by design, and have the least amount of human intervention required. The system would fly to its location, log data, and fly back, all autonomously, removing any difficulties that come with positioning and retrieving sensor nodes in arboreal environments, and instead translates the problem into developing a perching drone. Drone perching is a technology that is currently undergoing heavy development and research, and multiple prototype drones have been demonstrated to be able to perch on- or attach it self to surfaces [17], [33], [34].

The higher levels of integration also have drawbacks however. In particular, the fact that if the sensor cannot operate independently of the drone, the end users are forced to acquire a drone for every sensor node, resulting in an increased price-per-unit.

Furthermore, drones require a lot of power to operate, leading to a shorter maximum mission duration, considering the drones need to have enough power to fly to- and from the mission location. The reduction in mission time may be partially alleviated with the medium level of integration by splitting the power supply of the sensor subsystem and the drone. This does in turn lead to problems due to increased weight and in turn increased power requirements for the drone.

The low level of integration has a big advantage due to the reduced required initial investment, allowing for larger areas to be monitored with the same initial investment. The mission time of a sensor node is also no longer restricted by the drone's power supply an can instead be positioned and left there to be picked up at a later time. The amount of drones required would also be reduced, as a single drone could deploy multiple sensors throughout the mission.

The drawback however, is the fact that the drones will require a mechanism to be able to deploy and retrieve the sensors. Such systems are being developed and researched extensively, the solution with the most potential being the aerial arm manipulator, as demonstrated by [16], [17], [18].

A.2.2. Primary Sensors

The primary sensors are intended to be able to sample data that can be used to determine certain BEVs. An extensive overview of the advantages and disadvantages of many available sensors is given by Besson et al. (2022) [4], from which it can be found that the two most viable sources of data are acoustic data and digital video.

The resulting comparison can be found in Table 8, with the main takeaway being that video generates significantly more data, and often analysis algorithms require many more computational resources than usually available in low-cost systems. Furthermore, the use of the camera is restricted to the area it is pointed at, and only during daytime, unless the camera has night-time abilities. The microphone generates a much more manageable amount of data, and is not limited

TABLE 7: Stakeholder Analysis Matrix

Stakeholder	Influence	Interest	Stakeholder Need	Contributions	Obstructions	Notes
Researchers	High	High	Cheap, fast data-analysis pipeline	Agree to use and test the system.	May refuse to use system, particularly the older generation of researchers may object.	Primary target audience
Conservationists	High	High	Real-time disaster monitoring	Allow for testing of the system in areas of interest	May object to have drones enter ecosystems on a regular basis.	Secondary target audience
Government	High	Medium	Better data for policy-making	Sponsor the use and development of the system.	Regulations regarding drones, radio, and privacy.	
Locals	Low	Medium	Unintrusive methods for monitoring local environmental health	Insights into effective use of the system in their local environments.	May object to drones and audio recording devices to be present in the local area.	
Hobbyists	Low	Low	Cheap and easy to use equipment	May contribute to open-source development.	None	

by its direction, nor by night time. The main drawback of a microphone however is the fact that background noise may influence the recordings significantly.

TABLE 8: Comparison Overview: Video vs Acoustic Data

Data	Data Rate [mb/s]	Compute Load [GFLOPS]	Directionality	Restrictions
Acoustic	0.1 - 2	0.1 - 1	Omni	Background Noise
Video	1 - 100	1-100	Uni	Daytime light View Obstructions

A.2.3. Networking Architecture

There are many types of wireless connection types, and each type comes with a multitude of potential network layouts and network densities, as well as regulatory- and operational restrictions.

In terms of wireless connection types, there are five that can be considered for a wireless sensor application. Each connection type has a limit on the bandwidth and range and may require additional infrastructure to operate. Some connection types are also limited in network size or have restrictions on the amount of data that can be transmitted. Common restrictions for operating in the sub-GHz band for networks such as LoRa Peer-to-Peer (P2P) and Long-Range Wide Area Network (LoRaWAN) are limits on TOA and/or requiring an LBT protocol [23]. Table 9 summarizes the comparison between he different wireless connection types.

The range parameter is self-explanatory. Naturally, the connections with lower range will require a denser network to achieve coverage, while longer range communications will allow for sparser networks. The network size parameter indicates the theoretical maximum amount of nodes that can be connected to a single gateway, and the depth parameter indicates the longest possible chain of devices a message would have to pass through to reach the gateway. Larger allowable network depths increase coverage, but reduces the effective data-rates and introduces large delays between the end-nodes and the gateway [35], [36]. The bitrate column indicates the theoretical range of achievable bitrates, depending on other factors such as range, interference, and line-of-sight. Finally, the restrictions column briefly summarizes any potential requirements imposed on the wireless communication, from both legal and practical aspects.

TABLE 9: Comparison Overview: Wireless Communication

Type	Range [m]	Max. Size / Gateway	Max. Depth	Bitrate [kbps]	Restrictions
BlueTooth	5-50	Any	Any	100-900	None
WiFi	10-100	Any	2	100-10000	Need Extenders
ZigBee	10-100	Any	Any	20-250	None
LoRaWAN	100-10000	10-50	2	0.1-250	TOA / LBT
LoRa P2P	100-10000	Any	Any	0.1-250	TOA / LBT
Mobile/WAN	N/A[1]	Any	1	1000-50000	Infrastructure

A.2.4. Computational- & Data-Flow

All the data that is gathered by the sensor nodes must be processed somewhere. Cloud computing is the framework where all the raw data goes to a central location to be processed, while Edge Computing is the framework where data gets (partially) processed at the nodes themselves [5], [37]. Both have advantages and disadvantages, mainly pertaining to power consumption, processing power, and bandwidth

1. Dependent on network operator, terrain, weather, plan, and coverage

bottlenecks as described by Pan and McElhannon (2017) [38].

In particular, cloud computing requires high-bandwidth connections to be able to receive the data, in turn requiring the sensor nodes to also be able to sustain bitrates at a high-enough level for long periods of time. This is not always realistic, and in many cases impossible. This does reduce processing power requirements on the sensor nodes themselves.

Edge computing reduces the required bandwidth as it is possible to only send any potential findings back to a base station while saving the raw data locally. The sensor nodes should be able to run the analysis algorithm(s) in a timely fashion in order to function, which usually requires more processing power. An audiostream sampled at 48 kHz and 4 byte-wide samples generate 192 kB/s of data. On the other hand, a classifier may generate one of N detectable classes, which can be represented with a single integer, with the number of bytes needed given by:

$$n_{\text{bytes}} = \left\lceil \frac{\log_2(N)}{8} \right\rceil \tag{9}$$

In the case of BirdNet, the latest version supports 6522 different classes [39], meaning the audiostream does not need to be transmitted, instead a single 16-bit (2-byte) integer is enough to represent a single detection. If, for every 3 seconds of audio, the 10 highest-confidence detections are transmitted, that requires 20 bytes per 3 seconds, or 6.67 B/s: an information compression ratio of approximately 28800 compared to the raw audio-stream.

A.2.5. Underlying Hardware

The final concept category is the underlying hardware selection. There are four common possible choices for comparable systems, namely developing the system on an Field-Programmable Gate Array (FPGA), Digital Signal Processor (DSP), or on a CPU. Furthermore, systems can have many possible CPU configurations and combinations such as single-core, homogeneous multi-core, or heterogeneous multi-core processors. Table 10 summarizes the pros and cons of every configuration in a qualitative manner. It should moreover be noted that for running machine learning algorithms, the performance on an FPGA is unclear as it is still fairly experimental, though experiments show its great potential, particularly regarding power-consumption [40]. Furthermore, many of the common machine learning frameworks do not readily support FPGAs. Finally, the development effort estimate is based on a personal assessment of skill-levels for developing on the given hardware and as such is mostly subjective.

A.3. Functional Decomposition

In Figure 24 the functional decomposition of the system is shown, with five main branches, each representing a phase of the mission: the mission planning phase, the sensor deployment phase, the main data gathering phase, the sensor retrieval phase, and finally the finalization phase.

TABLE 10: Qualitative Comparison of Underlying Hardware

Type	Development Effort	Power Consumption	Price
FPGA	High	Medium-High	High
DSP	Medium-High	Low	Low
CPU (Single Core)	Low	Low	Low
CPU (Hom. Multi Core)	Medium	Medium	Low-Medium
CPU (Het. Multi Core)	Medium	Low-Medium	Medium

A.3.1. Mission Planning

The mission planning phase, as the name suggests, pertains to the planning of the mission and the preparation of the sensors. In the decomposition, the sensor preparation branch is the only branched expanded upon, as the other branches do not directly influence the capabilities of the system. The sensor preparation phase concerns the calibration of sensors, as well as checking data storage availability and checking whether the wireless modules function as expected before sending the sensor into the field.

A.3.2. Deployment & Retrieval

The deployment phase relates to the sensors being deployed either by drone or manually, and this phase is left for future design iterations. Similarly, the retrieval phase is left for future iterations as well.

A.3.3. Main Mission

The main mission is split into four sections: the initialization of the system, the collection of environmental variables, the monitoring of the system status, and finalizing the mission.

The initialization section is mainly concerned with setting up the wireless connections while out in the field. In particular, checking the LTE-M connection status, if used, and determining whether a direct LoRa connection to base-station is possible, or whether a multi-hop path is required to reach the base-station. Furthermore, the log-files and data-files are created on disk.

Next, the data-collection commences, where all AEV and BEVs are sampled at regular intervals. For the gas sensor, the heater profile can be programmed with up to 10 steps and temperatures, as well as the sampling settings such as the ODR and oversampling settings.

The status monitoring section mainly concerns itself with monitoring available power, available storage, wireless connection status, and the device's position and attitude. Furthermore, commands to change any system parameters can be received over the wireless connections. Errors are also logged continuously and can be inspected in the log file. The number of critical errors can be requested and sent over the wireless medium of choice. Moreover, the latest data samples can be requested too over the wireless medium in order to be analyzed in near real-time.

The final section is mainly concerned with ensuring the log- and data files are not corrupted and closed appropriately. The sensors are then stopped from logging further data, aside from the GNSS and IMU modules, which remain logging the system state until retrieval.

A.3.4. Mission Finalization

The final phase consists of extracting the logs and data files from the board and analyzing all of the data. This analysis is done off-board, therefore once the data is taken from disk, the sensor can be sent out into the field again with a new battery.

A.4. Requirements Generation

From the functional decomposition the system requirements flow naturally, with both the functional and non-functional requirements given in Appendix B.

A.5. Concept Tradeoff

Considering the goal of the project, as mentioned in section 3, states the need of a cheap, reliable and versatile system, the trade-offs made during the design of the system reflect this goal. The concept branches explored in subsection A.2 are pruned such that the design can converge, with the choices made explained in this section.

A.5.1. Drone Integration Level

As discussed in subsection A.2, there are three levels of drone integration we may consider for the design: high-, medium- and low levels of drone integration. For the final design, a low level of integration is determined to be the most suitable choice given the project goal. In particular, the increased cost per unit that comes with higher levels of integration could not be justified, regardless of any other advantages they may give. This will give much more freedom to the final users of the system in determining in how they wish to deploy the sensors.

A.5.2. Primary Sensor(s)

The primary sensors could be either a microphone, or a camera, as discussed in subsection A.2. The main trade-off is between the potential of the gathered data in extracting BEVs, as well as the rate at which it is generated and the associated analysis algorithm Floating-Point Operations Per Second (FLOPS). In order to minimize the cost of the hardware needed for the on-board analysis and data storage as well as power requirements, an approach utilizing audio data is more appropriate.

A.5.3. Networking Architecture

There are many potential network layouts, architectures, and protocols to choose from, each with their own distinct pros and cons. In order to save power and bandwidth, it is not desired to stream all of the recorded data continuously. Furthermore, it is beneficial to the end user to have freedom in determining how densely an area is to be populated with sensor nodes. By choosing a longer-range medium, this choice is fully left to the end-user. By looking at Table 9, the only viable choice left after ruling out short-range media and media with infrastructure requirements, is LoRa P2P.

A.5.4. Computational- & Data Flow

Regarding computational- and data flow, there are two main directions the system can go: cloud computing and edge computing. As mentioned in subsection A.2, cloud computing introduces delays and bottlenecks and increases the performance requirements on the networking capabilities, while edge computing solely increases the requirements on the underlying hardware. Considering the sensors are likely to be used in areas with little to no high-bandwidth connections available and the delay between sample and analysis is to be minimized, edge computing is the logical choice for this system.

A.5.5. Underlying Hardware

The final trade-off made for the system is the type of underlying hardware which will execute all the tasks. As mentioned, the main contenders are FPGAs, DSPs, or CPUs. The CPUs are furthermore divided into multiple categories. After considering the main use case of the system to run an analysis algorithm on-board while also sampling data and communicating wirelessly, a heterogeneous multi-core processor is deemed to be the most appropriate solution given the project time frame, budget, and power requirements.

A.6. Feasibility Study & Risk Identification

A brief feasibility study is performed before diving into the development of the system, addressing concerns and identifying any risks from economic, environmental, and legal aspects.

The design is intended to be cheaper than the current state-of-the-art, both in terms of initial investment costs and maintenance costs. Current state-of-the-art from companies such as Tramex provide wireless sensors such as the TREMS-5 [41] which can measure certain AEVs, with a price of over €1600 for 5 sensors and a base station. The AudioMoth is priced at €60 a piece, but offers no wireless capabilities and no option to sample AEVs instead. This product aims to provide the user with both AEV and BEV data as well as wireless capabilities and online inference for a price that is around the €100-200 mark per sensor node for initial investments, and a maintenance cost that is to be minimized. Maintenance costs may include battery replacements, repairing drone components such as rotor blades, and replacing any malfunctioning or worn-out sensors.

Considering the system is designed in part to address environmental concerns, the environmental impact of the system itself should be taken into careful consideration. Factors such as the energy consumption and the sustainability of the materials used in the system are to be taken into account, as well as any electronic waste generated throughout the system's life-cycle must be accounted for. The system moreover introduces risks through the introduction of electronics and in particular batteries into fragile ecosystems which may be at elevated risks of wildfires. The introduction of drones also poses a risk to safety to both humans and wildlife, as accidents are not an impossibility. These risks must be assessed at the site where the system is to be used and must follow local laws and regulations.

Finally, laws and regulations regarding the use of certain radio frequencies is another issue that must be taken into consideration. Many countries allow the use of different radio frequencies and bandwidths, in turn influencing the usability of the system in these regions. For example, many EU countries allow the use of the 433 MHz and 868 MHz bands with certain restrictions, while in the US the 915 MHz band is used instead. Similarly, adhering to regulations around drone usage is important when the system is to be deployed using drones, particularly in urban areas and areas in the vicinity of airports or military installations.

A.7. Concept of Operation

The system's concept of (ideal) operation can be derived from the previously discussed functional decomposition and the generated requirements. First, the sensor nodes must be prepared by ensuring the batteries are charged, sensors are calibrated, the wireless communication functions appropriately, and the positioning of the sensors in the field is determined. In Figure 23 an example of a deployment is shown where the wireless medium with mesh-network capabilities is assumed to have a range of 500 meters. After these initial steps, the sensor nodes are ready to be deployed, either manually or with drones.

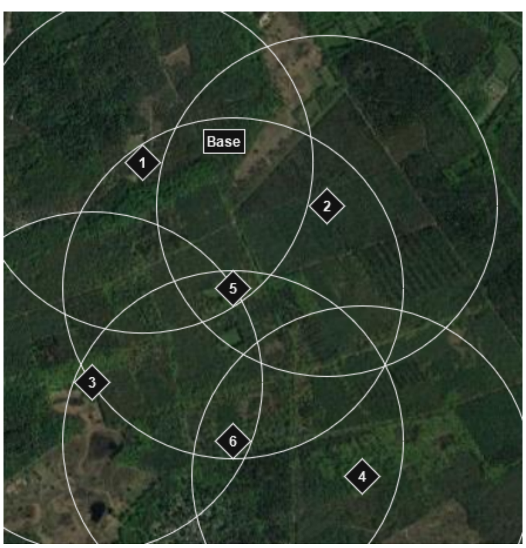


Figure 23: Map-View of a possible sensor deployment spatial layout ($r \approx 500$ m)

After deployment, a multitude of tasks are executed in concurrently. Data is collected through the sensors carried on-board, with on-line inference being performed on this data in (near) real-time. Ideally classifiers are present for both the AEVs (eg. gas classifier) and BEVs (eg. bird sound classifier), though the presence of a BEV classifier is more valuable due to its ability to reduce the amount of data that has to be sent over a wireless medium.

The wireless communication medium handles any incoming messages and send out messages pertaining to the sensor's operational status as well as part of the data collected. Furthermore, in case of a mesh network, messages from other sensors are to be relayed to the base station. Moreover, in the case of any unusual events such as detecting large movements of the sensor, a notification is sent to the base-station such that action can be undertaken to recover the sensor.

After completion of the sensor's data logging tasks, it is to be retrieved, either manually or with a drone, such that all the raw data can be extracted from the sensor such that it can be sent out into the field again to repeat the process. The raw data can in turn be used for full-scale data analysis at the base station.

Appendix B. Requirements

B.1. Functional Requirements

FR-1: The sensor node shall be able to perform diagnostics on its internal functioning
- FR-1.1 The sensor node shall be able to calibrate all sensors
- FR-1.2 The sensor node shall be able to check the available battery level
- FR-1.3 The sensor node shall be able to check the available disk storage

FR-2: The sensor node shall be able to attach/fasten itself to the tree where it is positioned

FR-3: The sensor node shall be able to communicate with a central base station
- FR-3.1 The sensor node shall possess at least one unlicensed LPWAN communication medium
  - FR-3.1.1 The sensor node shall be able to communicate with its neighbours
  - FR-3.1.2 The sensor node shall be able to pass along messages to its neighbours
  - FR-3.1.3 The sensor node shall adhere to the ISM band regulations of the region
- FR-3.2 The sensor node should possess a licensed LPWAN module
  - FR-3.2.1 Sensor nodes with a licensed LPWAN connection shall be able to stream data in real-time
  - FR-3.2.2 Sensor nodes with a licensed LPWAN connection should act as data sink
- FR-3.3 The sensor node shall be able to select the transmission bit rate

FR-4: The sensor node shall be able to record audio
- FR-4.1 The audio sensor shall be able to sustain a sampling rate of at least 44.1 KHz
- FR-4.2 The sampling rate of the audio sensor shall be variable
- FR-4.3 The audio gain shall be variable

FR-5: The sensor node shall be able to record multiple environmental variables
- FR-5.1 The sensor node shall be able to record ambient temperature
- FR-5.2 The sensor node shall be able to record ambient pressure
- FR-5.3 The sensor node shall be able to record ambient humidity
- FR-5.4 The sensor node shall be able to record atmospheric gas resistance
- FR-5.5 The sensor node should be able to record solar intensity
- FR-5.6 The sensor node should be able to record wind speed
- FR-5.7 The sensor node should be able to record ambient radiation levels

FR-6: The sensor node shall be able to have a programmable sampling schedule
- FR-6.1 The sensor node should be able to deactivate/reactivate sensors at will
- FR-6.2 The sensor node should be able to change the sampling rate of any onboard sensor

FR-7: The sensor node shall be able to perform online inference on the audio data
- FR-7.1 The sensor node shall be able to run BirdNET inference within 3 seconds

FR-8: The sensor node should be able to detect forest fires through its environmental sensors

FR-9: The sensor node shall be able to deduce its current physical state
- FR-9.1 The sensor node shall possess GNSS capabilities
- FR-9.2 The sensor node shall possess an accelerometer
- FR-9.3 The sensor node shall possess a gyroscope
- FR-9.4 The sensor node shall possess an RTC

FR-10: The sensor node shall be able to store all collected data on-board

FR-11: The sensor node shall operate using an operating system

FR-12: The sensor node shall keep an event log where all system events are noted

FR-13: The system shall not leave behind any non-biodegradable materials in the environment

B.2. Non-Functional Requirements

NR-1: The sensor node shall be able to run continuously for at least 6 hours

NR-2: The sensor node shall not weigh more than 200 grams

NR-3: The sensor node shall not exceed dimensions of 12 cm $\times$ 8 cm $\times$ 8 cm

NR-4: A sensor node unit should not exceed a cost of €200

Appendix C.
Large Figures

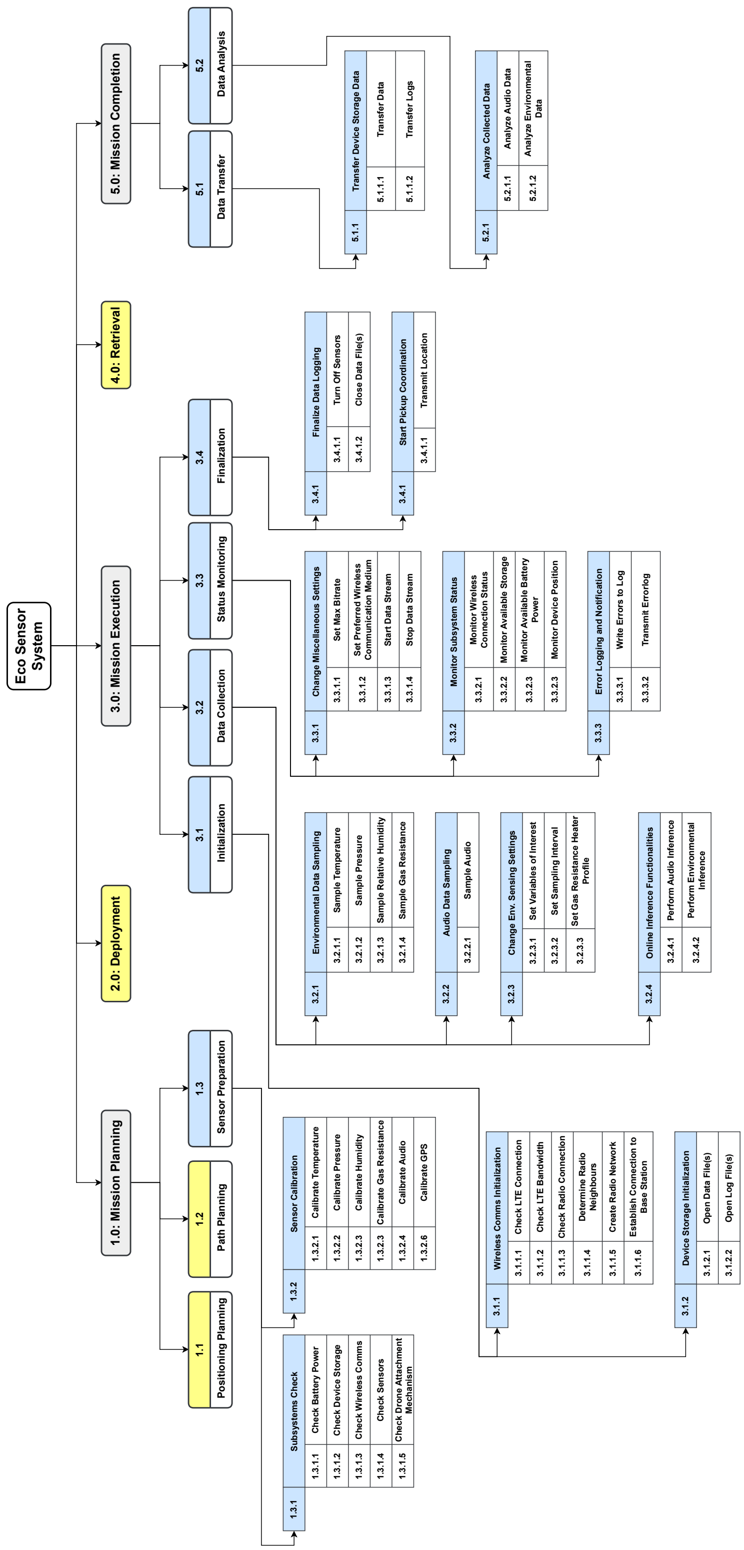


Figure 24: Functional Decomposition Diagram

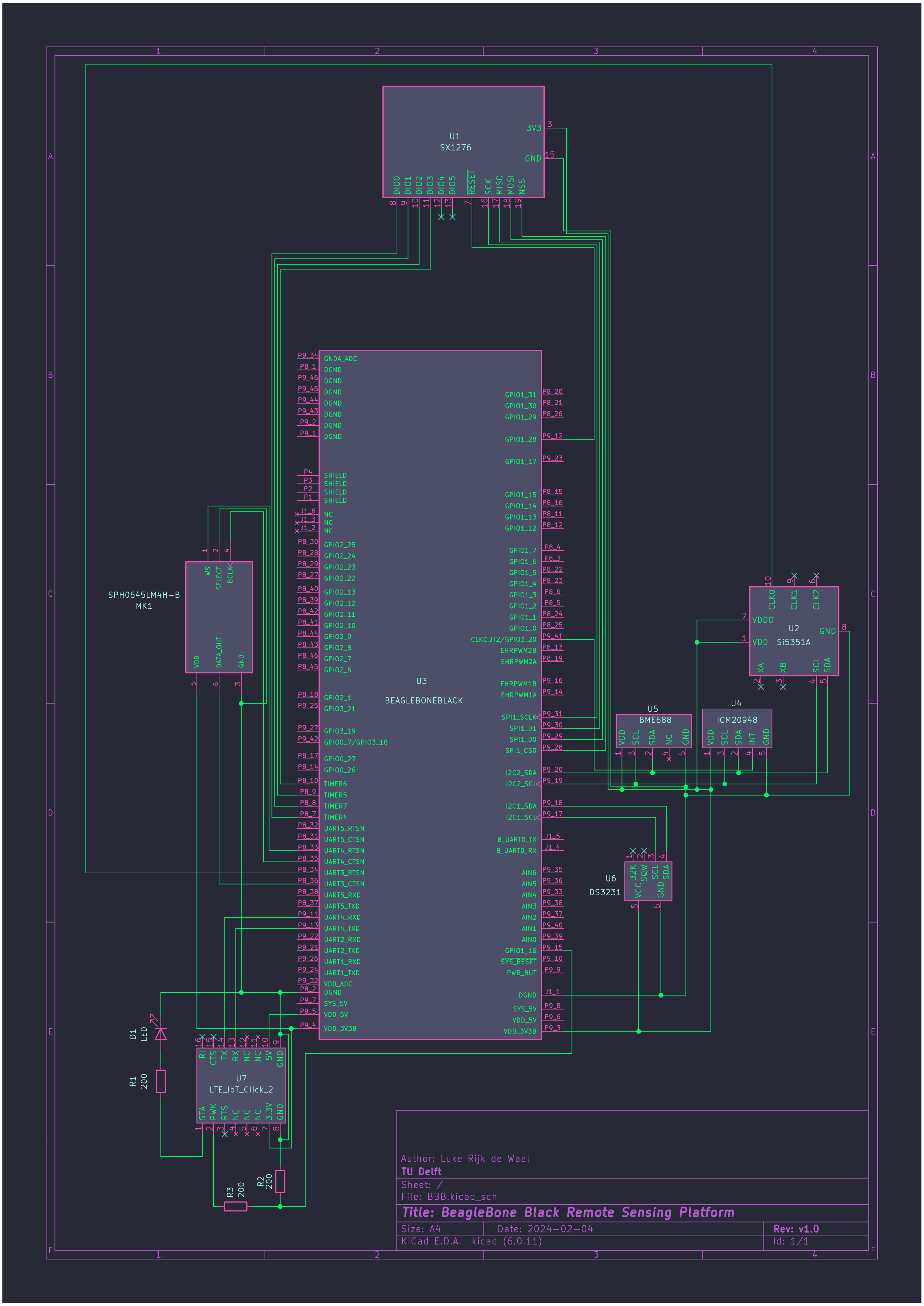


Figure 25: System Hardware Schematic

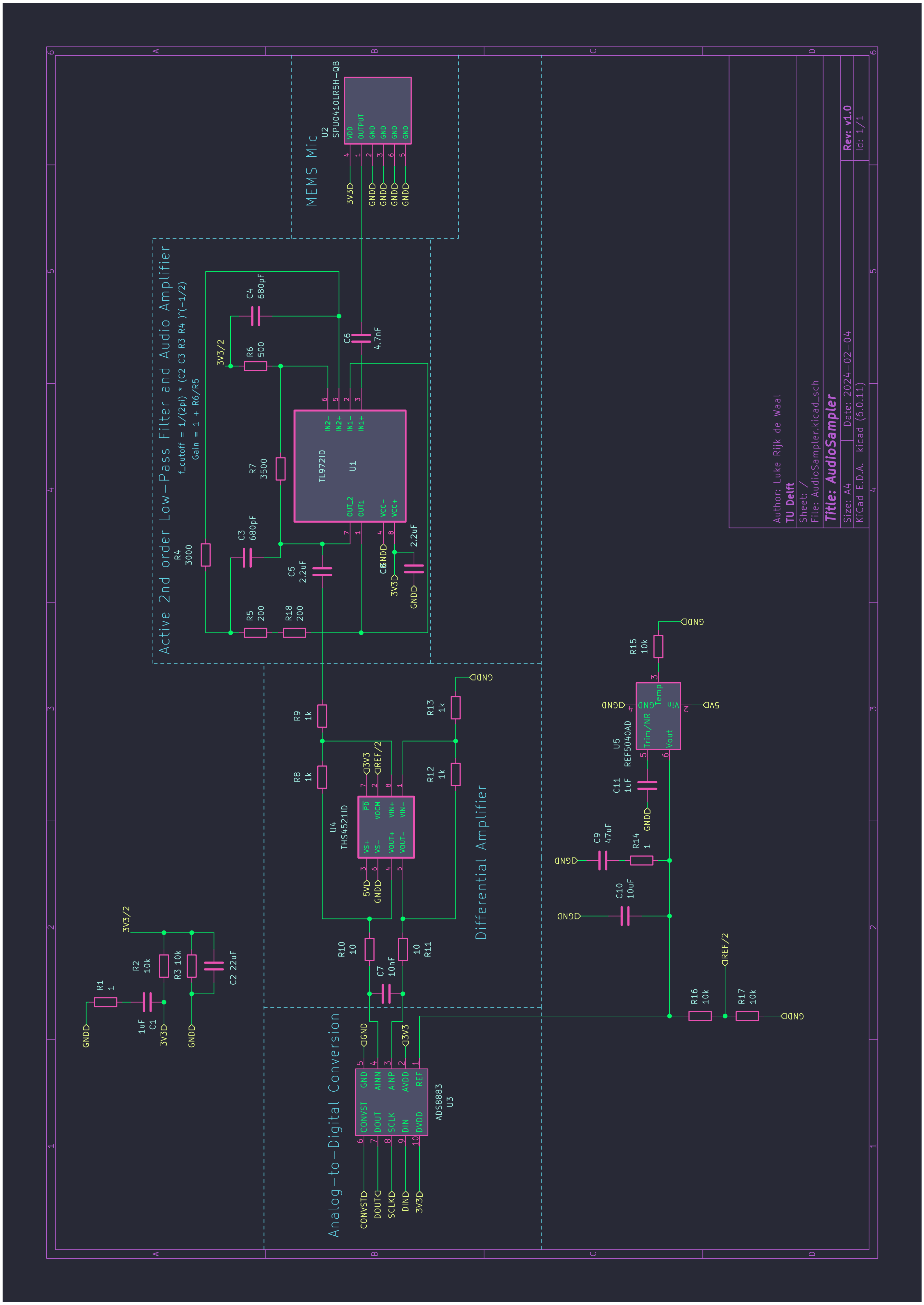


Figure 26: AudioSampler Hardware Schematic

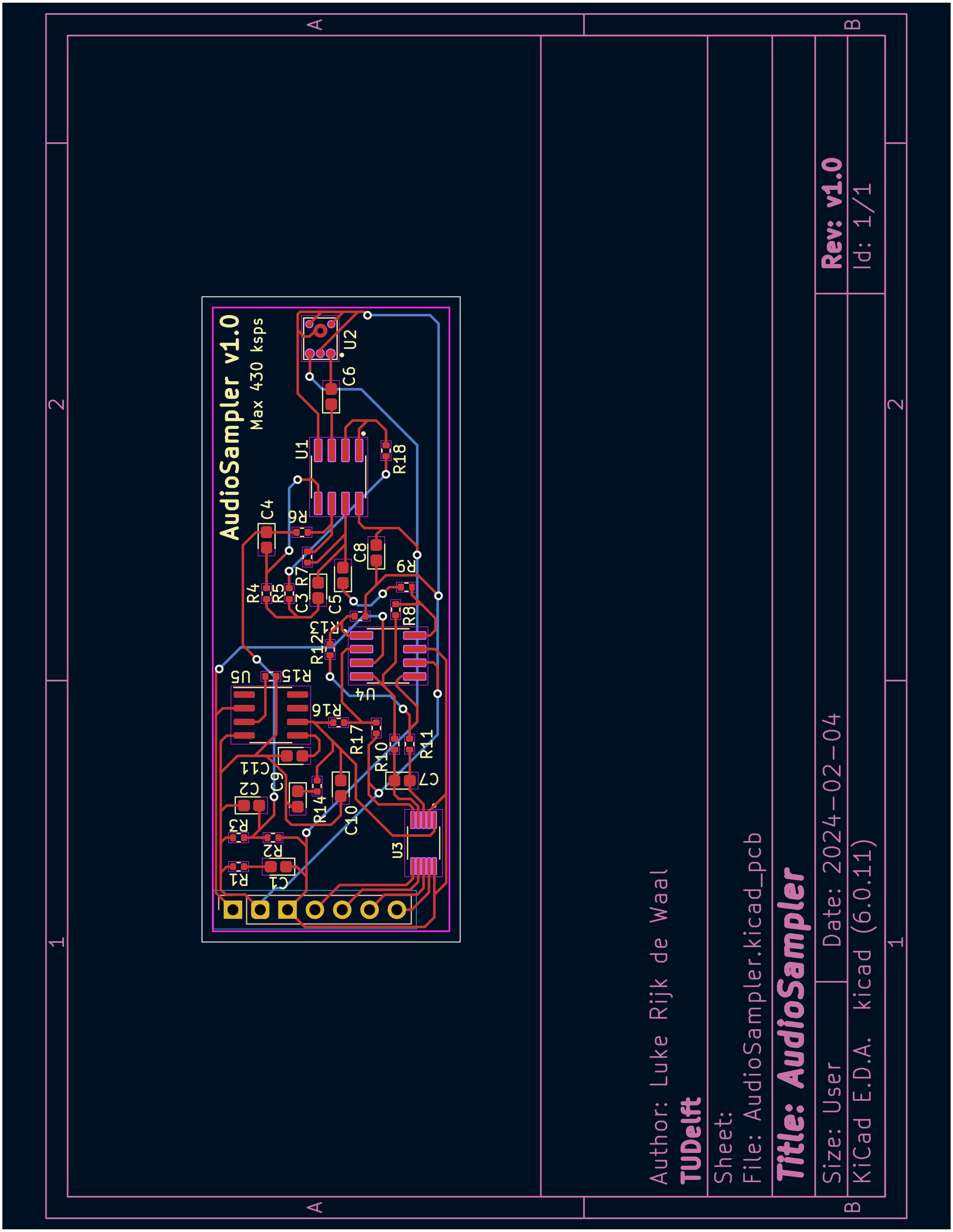


Figure 27: AudioSampler PCB layout