Raspberry Pi Based Recording System for Acoustics Monitoring of Bird Species.

Gabriel KIARIE[1] and Ciira wa MAINA[2]

[1,2] *Dedan Kimathi University of Technology, Department of Electrical and Electronic Engineering, Centre for Data Science and Artificial Intelligence (DSAIL), Private Bag-10143, Nyeri, Kenya.*

[1]*Email: gabriel.kiarie@dkut.ac.ke*

[2]*Email: ciira.maina@dkut.ac.ke*

DSAIL Technical Report

DSAIL-2020-001

December 2020

ABSTRACT

Severe changes in ecosystems due to human encroachment and climate change call for close monitoring of ecosystems in order to conserve them. Bioacoustics monitoring provides a means to remotely monitor ecosystems of interest especially in the remote and inaccessible parts. Birds are one of the most vocal species and they respond rapidly to changes in the ecosystems they are found in. Therefore, by carrying out acoustic monitoring of the birds present in a given ecosystem, we can learn about the changes taking place in it. In this paper, we present an acoustic system that is based on the Raspberry Pi and is used to collect audio recordings for use in acoustic monitoring of birds. The audio data collected will be used to develop machine learning models to perform automatic classification of birds. This will offer a tool to provide continuous monitoring of ecosystems.

Keywords: Spectrograms, acoustics, conservation.

List of figures

List of tables

Contents

1. Introduction

Birds respond quickly to changes in the ecosystems they are found in. This makes them to be one of the best species to use in studying ecosystems [1] [2]. Increase in human activities and climate change in recent years have resulted in severe degradation of ecosystems [3]. It is projected that by the year 2100, 600-900 land bird species may become extinct due to global warming. Majority of these, about 89%, are species in the tropics [2]. Therefore, conservation efforts need to be increased especially in the tropics.

Old methods of carrying out surveys to study ecosystems are expensive and limited by other factors like time, weather and inaccessibility of some parts of the ecosystems [4] [5]. However, use of automated systems for data acquisition and classification can help overcome these problems.

Different species of animals, especially birds, vocalize frequently [6]. The sound produced by an animal contains the 'acoustic fingerprint' of its species [7]. Therefore, we can classify organisms by listening to the sound they produce. It is easy to detect many birds from the call they make hence acoustic classification of birds is a good method of studying birds [8].

To completely understand the effects of climate change on ecosystems, we require to process data collected over a reasonable amount of time. There exists a gap in the research of birds with the aim to conserve ecosystems. In this paper, we present an inexpensive DSAIL Bioacoustics System that will help address this gap. The system will be deployed in ecosystems of interest to collect audio data that will be used to develop a machine learning model to classify birds automatically from their vocalization.

The system is based on the Raspberry Pi single board computer. The Raspberry Pi was chosen since it provides a platform to record, perform classification and store the time stamped audio recordings for retrieval. It is operated using a Unix-based operating system that is freely distributed and can be powered from DC sources like batteries [9]. The system also comprises a power management board designed at our lab to help power the Raspberry Pi intelligently.

Section 2 outlines the objectives of the project. Section 3 describes the methodology used in carrying out the project. Section 4 presents and discusses the results obtained from the project. Section 5 talks about the future work of this project. Section 6 concludes the paper.

2. Objectives

The objectives of this project are:

i. Design an acoustic system to record birds' acoustic activities.
ii. Deploy the system in various ecosystems to collect data.
iii. Retrieve the recordings and label them with the help of an ornithologist.
iv. Use the data obtained from the system in studying the bird species trend in ecosystems

3. Methodology

To achieve automatic acoustic classification of birds, we need enough acoustic data to train machine learning models to do the classification. The first step of this project was to design the

DSAIL Bioacoustics System that will enable us to collect acoustic data of birds in the ecosystems. The system is based on the Raspberry Pi 3 single board computer. The DSAIL Bioacoustics System comprises two major components i.e. the hardware and the software. We will go through each of this in the following sections.

3.1. Hardware

The hardware of the acoustic system is the physical parts of the system. The main hardware in the acoustic system are the Raspberry Pi 3, USB microphone, Power management board, lithium battery, solar panel and a charging module. The system was designed and developed at the DSAIL lab. We will look at each of these components in the following sections.

3.1.1. The Raspberry Pi

The Raspberry Pi is the central device in the acoustic system. It controls the performance of the setup. The Raspberry Pi is used to record and store the time stamped birds' sounds in an external storage or in its SD card. It also monitors the battery voltage and makes informed decisions of shutting down the system whenever the voltage drops to a certain predetermined value.

3.1.2. USB microphone

Our system uses a USB microphone to record sound. The microphone is equipped with an analog to digital converter, ADC, since the Raspberry Pi lacks an onboard ADC hence cannot be fed with analog signals. The USB microphone records sound and sample it at a frequency of $48kHz$ and feed the sampled sound to the Raspberry Pi through a USB port. The USB microphone used is sensitive enough for our setup but it lacks the wind cancelling feature.

3.1.3. Power management board

The Raspberry Pi operates like a normal computer. There is constant reading and writing on to its storage. This means care needs to be taken when shutting it down just like is expected in the common computers we use every day. Abrupt cutting of power may result in data loss due to corruption of the storage. In order to manage power, a lot of attention was directed towards designing a power management board for the system. The board acts as a power supply to the Raspberry Pi and performs intelligent shutdown and wake up of the Pi.

The main section of the power management board is the switching section. It is responsible for switching different sections of the circuit or the entire circuit on or off. Figure 1 is the circuit of the switching section of the board:

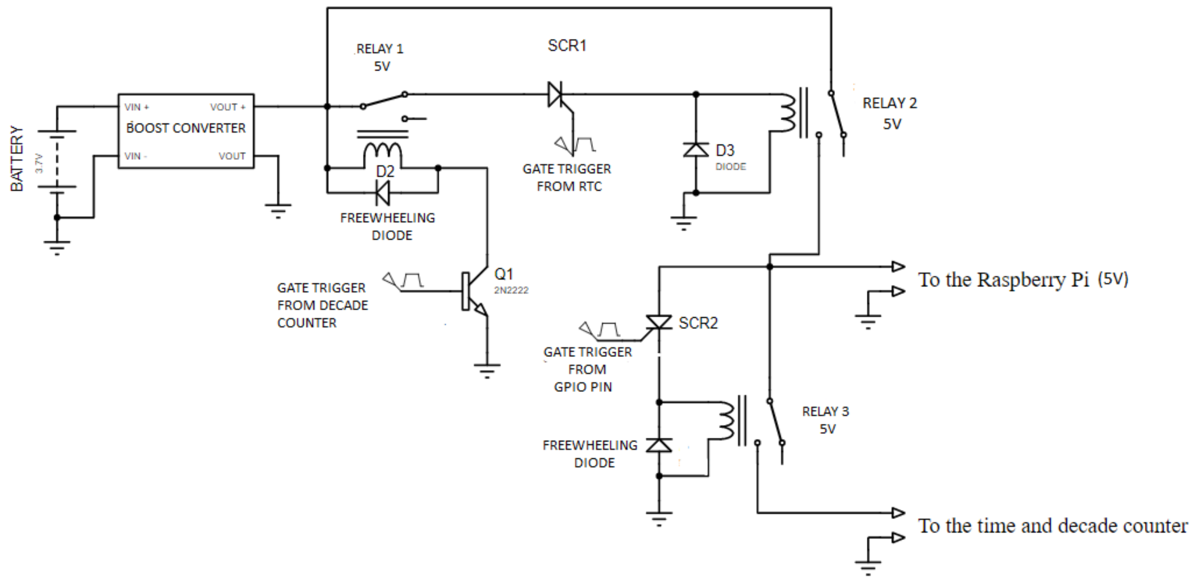

Figure 1: The circuit of the switching section

The circuit is based on the latching effect of the silicon controlled rectifier (SCR). Once a forward biased silicon controlled rectifier's gate is triggered by a sufficient voltage, it enters into a latching state where it conducts continuously provided the anode current is above the rated holding current. When the anode current drops below the holding current or the anode to cathode current flow is removed, the SCR stops conducting and enters off state [10]. This makes it easier to switch on and off our circuits using single triggers.

A single gate trigger of SCR1 from the RTC introduces a latch to SCR1 and this makes RELAY 2 to switch to the normally open (N/O) terminal connecting the Raspberry Pi to power. Similarly, a single trigger from the GPIO pin of the Raspberry Pi will switch the timer and decade counter circuits on by latching SCR2. To switch off the entire circuit, we need to cut off the anode current of SCR1 momentarily. This is achieved by triggering the NPN transistor. This opens RELAY 1 cutting the anode current of SCR1 momentarily hence switching the entire circuit off.

Every battery has the maximum allowed depth of discharge(MDOD) beyond which it should not be discharged. This is meant to protect the battery's lifespan. Also, for our system, we do not intend to have the system shutting down out of battery drainage since this may result in loss of data due to corruption of the SD card. For this reason, we need to continuously monitor the battery voltage to ensure we do not go above the allowed MDOD. In our setup, we used a $3.7V$ lithium battery that has a MDOD of 80% or the battery should not drop below a cut-off voltage of $2.8V$. Should the battery voltage get to the cut off voltage, the system has been designed to initiate a shutdown procedure.

The power management board enables the Raspberry Pi to continuously monitor the depth of discharge of the battery by reading the voltage level of the battery. The Raspberry Pi lacks analog pins hence cannot read the battery voltage level directly. An analog to digital converter (ADC) and a potential divider have been added on the board to read the voltage and feed it to the Pi in digital form. This enables the Raspberry Pi to make an informed decision of shutting down the whole system whenever the voltage of the battery reaches the cut off voltage.

The voltage reading after every five minutes is stored in a csv file together with the time of that reading. From these readings we can assess the power performance of the system and determine the health of the battery. Figure 2 shows the circuit used to enable the Raspberry Pi to monitor the voltage:

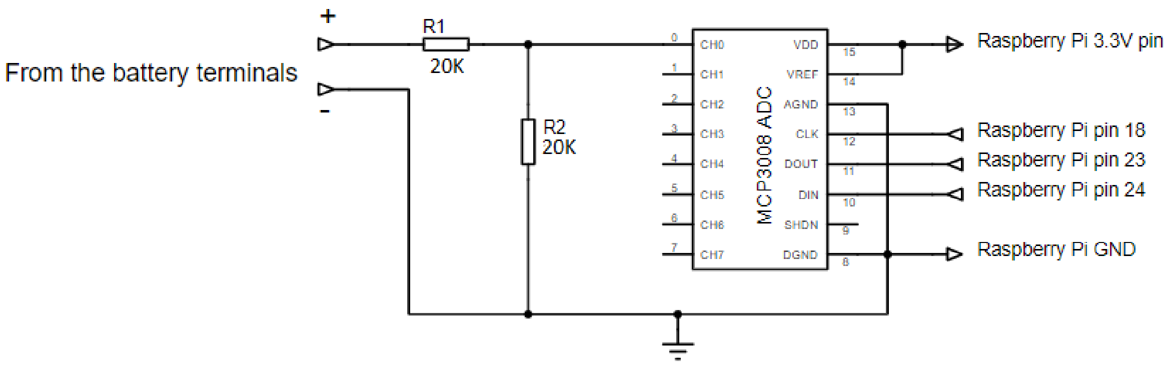

Figure 2: Voltage reading circuit

Additionally, the board has a DS3231 real time clock (RTC). The RTC plays very important roles in the system. The first and the main role is scheduling wake up of the Pi. The DS32131 RTC is equipped with an alarm interrupt that is used to switch the entire system back on. Before the Pi shuts down, it normally schedules the time it plans to wake up. This is done by setting an alarm on the RTC whose interrupt is used to trigger the waking of the system. The RTC also is used to set the Raspberry Pi time and date on boot since the Raspberry Pi lacks an on board RTC. This ensures that the saved recordings are time stamped correctly. Figure 3 shows the RTC setup circuit:

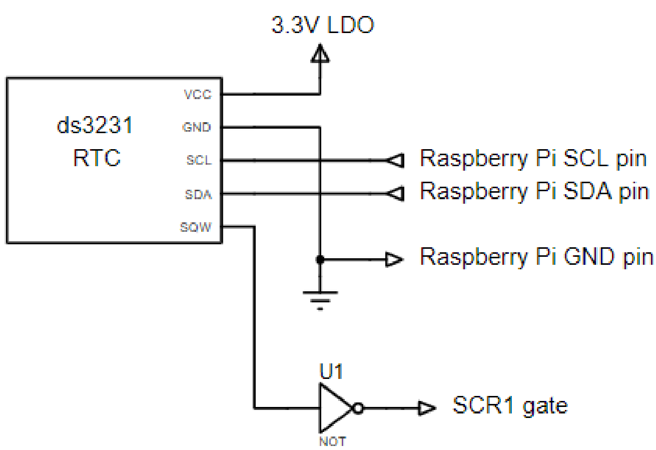

Figure 3: The RTC setup circuit.

The DS3231 alarm interrupt is active low. This means that the SQW pin goes low when the set alarm time matches with the RTC time. To use this interrupt to trigger the main SCR in the switching section we feed the output of the SQW pin to a not gate. Whenever the output of the SQW pin goes low due to the alarm interrupt, the output of the NOT gate goes high hence triggering the main SCR to switch on the entire circuit.

Whenever it is time for the Raspberry Pi to shutdown, the correct procedure of shutting down needs to be done before the whole system is disconnected from power. To achieve this, a timer circuit has been incorporated in the board to allow the Raspberry Pi to shut down before power is disconnected. The time circuit consists of a 555 timer connected in astable mode and a decade counter. The time produces a pulse waveform of period of about *4s* that is fed to a decade counter. The decade counter triggers power disconnection at the tenth pulse. This gives the Raspberry Pi about *40s* to safely shutdown which is more than safe time. The Raspberry Pi normally triggers the timer circuit to start counting whenever it has made a decision to shutdown.

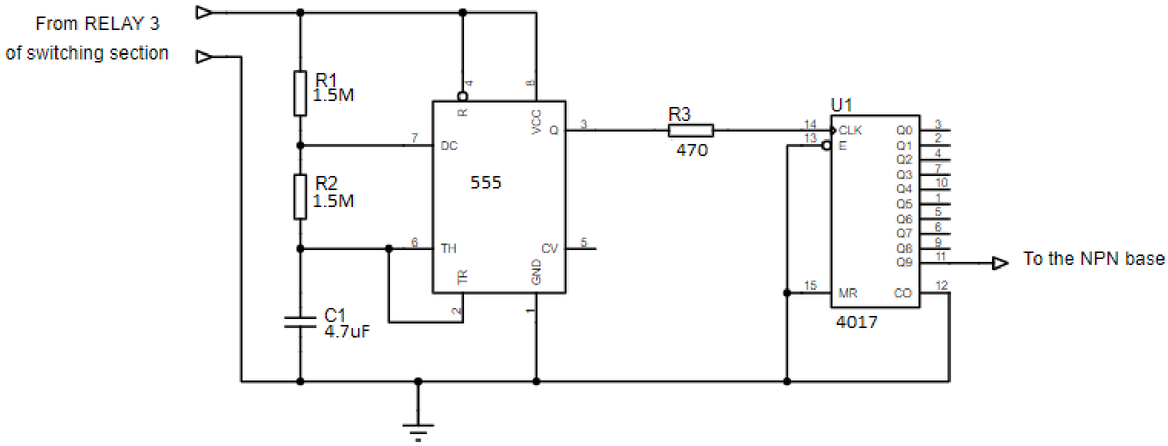

Figure 4: Circuit of the timer section.

N/B: The gates of the SCR and the base of the NPN need to be connected in series with a resistor and the pins triggering them. The resistor limits the current drawn hence protecting both the sourcing and sinking devices. The resistor values may range from $220\Omega - 1K\Omega$. In this system, 470Ω resistors were used.

3.1.4. Power source

Power is a crucial consideration when designing any system particularly those that are meant to be deployed in the field. For power, our system uses a solar panel that is used to charge a lithium battery. Figure 5 shows the power system setup:

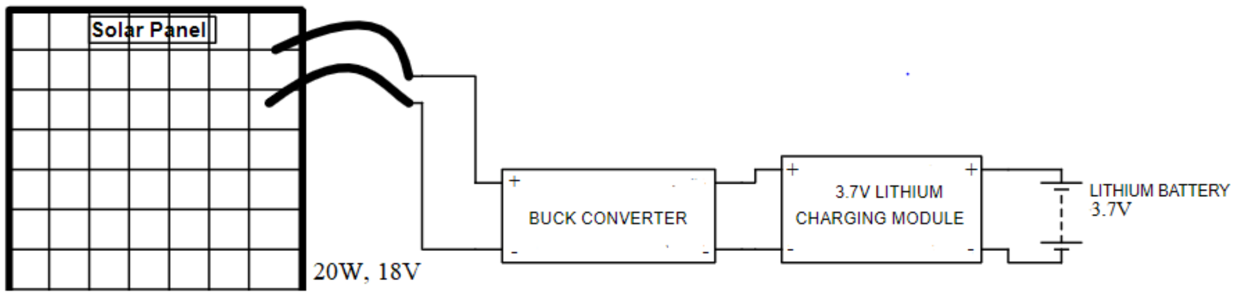

Figure 5: Power source.

The setup above is used to charge the battery voltage. The buck converter is used to step down the solar voltage from $18V$ to $5V$ which is then fed to the lithium cell charging module. Due to the intermittent nature of solar power, we need to continually monitor the state of the battery. Figure 6 shows the circuit of the power management board.

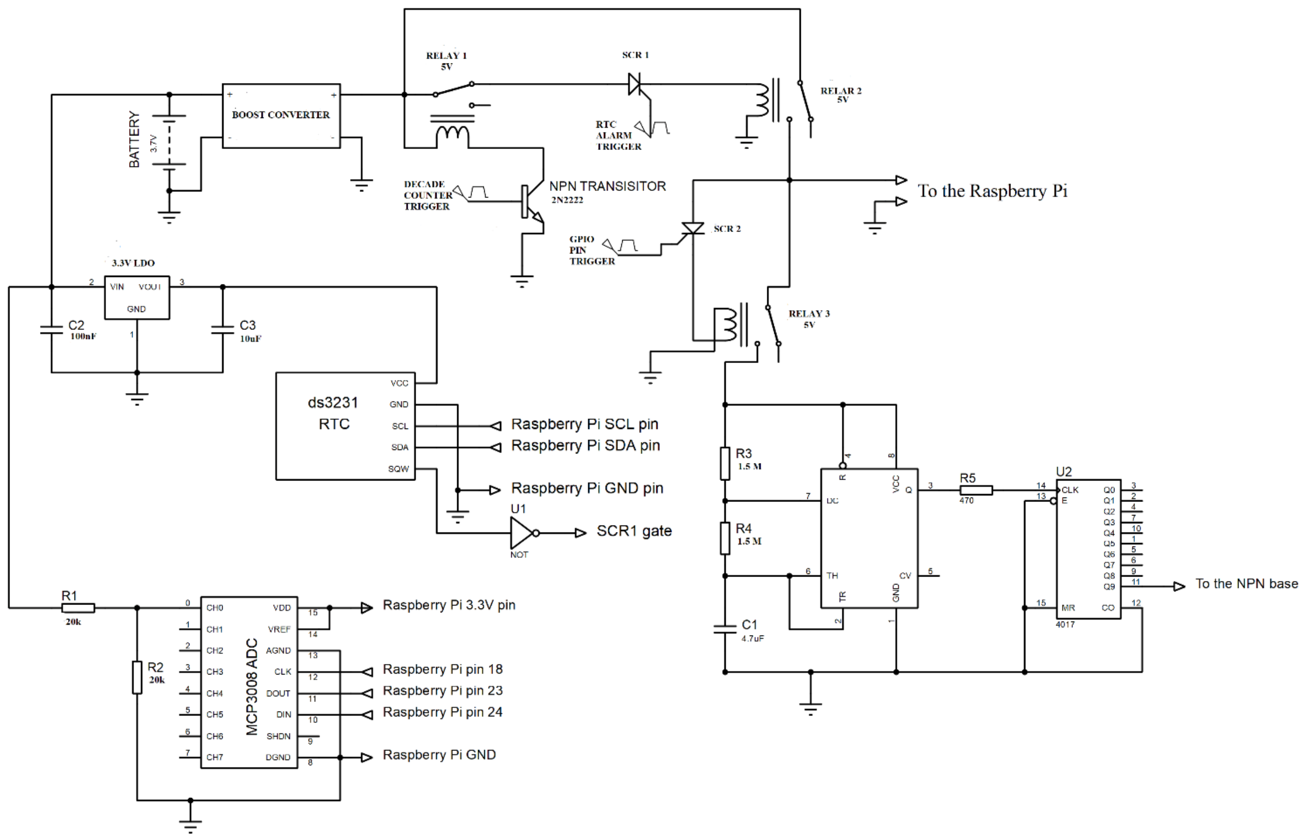

Figure 6: Circuit of the power management board.

3.2. Software

The Raspberry Pi in the acoustic system has been loaded with Raspbian Buster operating system (OS). Raspbian is a freely distributed OS that is based on Debian and is optimized to work with the Raspberry Pi hardware [11]. To achieve power management and recording of birds' sounds, two major programs were developed. In the following sections, we will look at each of these programs:

3.2.1. Power management program

The power management program is used to control the shutting down and waking up of the system. The program controls the modules on the power management board to achieve this. Our current setup is designed to work during certain hours of the day due to power constraint. The system wakes up at 5.00 am every morning and shuts down at 11.00 am when birds are normally most active.

During this window of operation, the power management program monitors the battery voltage to ensure it is above the cutoff voltage of the battery, 2.8V. The program reads the voltage reading of the ADC after every thirty seconds. The voltage reading corresponding to each five minutes' interval is saved in a CSV file to be used in power analysis of the system. The program, therefore, shuts down the system depending on the time of the day or the voltage of the battery. The program is also responsible for setting the alarm of the RTC to wake the system at 5.00 am the following day and then triggering the timer circuit to disconnect the entire system from power after the Raspberry Pi shuts down.

3.2.2. Audio data collection program

This program is used to enable our system to collect audio data. The program is written in Python programming language and makes use of some third party libraries like librosa and sounddevice. Librosa is a python module that is used for music and audio analysis [12]. The program records continuously and passes blocks of equal length to a queue. The blocks are then fetched from the queue and analyzed to check for audio activity. Blocks that possess high energy compared to a preset still condition are saved as time stamped audio files to be used as the data. With the help of an ornithologist, the files will be labeled and used in machine learning. The same principle of detecting activity will be used when classifying birds. Figure 7 shows the flowchart of the audio data collection program:

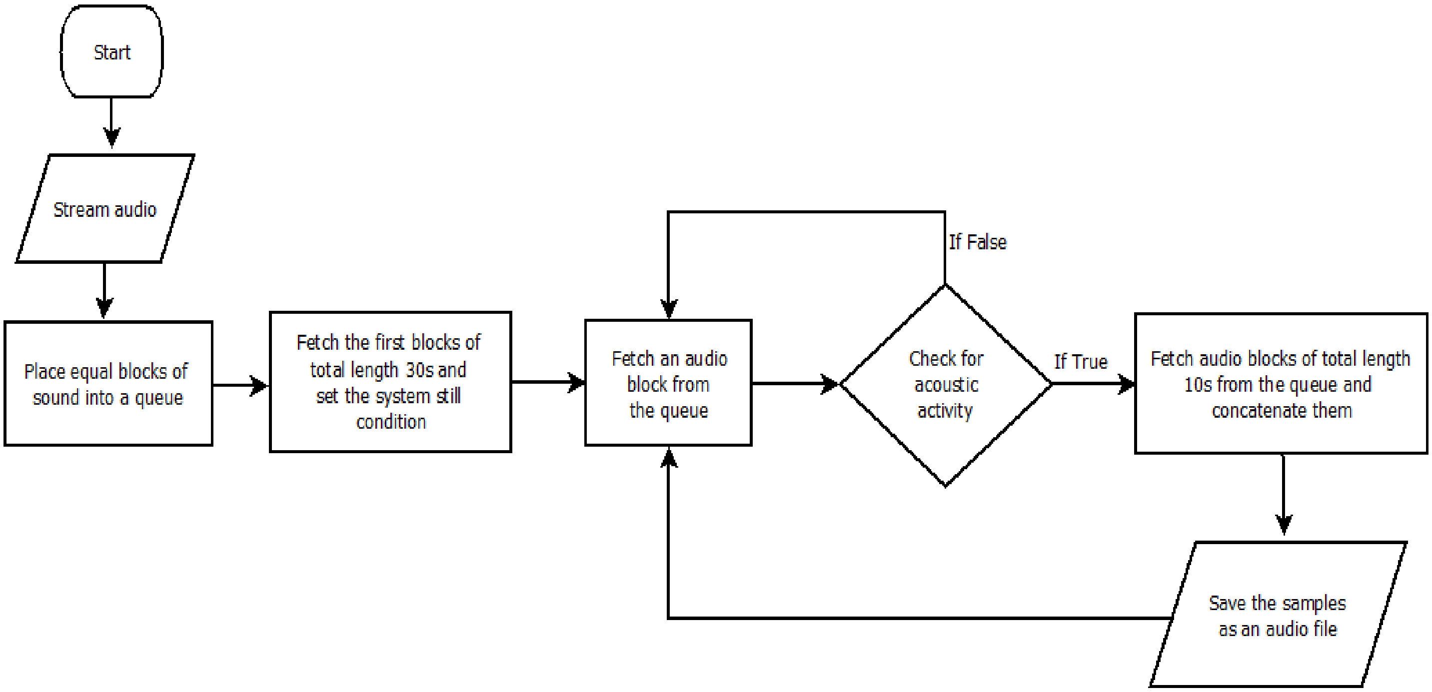

Figure 7: Flowchart of the audio data collection program.

During development of the system, several deployments were done to test the performance of the system. Early deployments were carried out at Dedan Kimathi University of Technology with an aim to study the efficiency of the power management board and program. After several iterations we were able to achieve a system that met our needs. Later the system was deployed with the audio data collection program at Dedan Kimathi University Conservancy and we were able to collect audio data of birds.

Figure 8: DSAIL Bioacoustics System deployment outside the university Resource Centre for power analysis of the system (left) and at the university conservancy to collect acoustic data of birds (right).

4. Results and Discussion

In this section, we will present and discuss power analysis and audio data that were collected during development and deployment of the DSAIL Bioacoustic System.

4.1. Power analysis

One of the challenges encountered when designing sensors is increasing the time of their operation since most of them are battery powered [13]. In our design, we used the available materials to design a solar powered system that will be working during a given time of the day when birds are most active and then shutdown for the battery to recharge. Using the measured power consumption of the system and an estimate of the power provided by the solar, we can plan the time the system should operate before shutting down for the battery to recharge. In this section, we will present and discuss the data collected for power analysis of our system.

The ratings of the battery and the solar panel used in this experiment are:

1. Battery - $10{,}000mAh, 3.7V$
2. Solar panel - $20W, 18V$

The currents being drawn by the Raspberry Pi 3 when connected at to a power supply at are as follows:

Table 1: Current drawn by the Raspberry Pi.

S/N	State/Activity	Current(A)
1	Idling without the microphone	0.22
2	Idling with the Microphone connected	0.24
3	When recording sound(program running)	0.25

The Power board consumes a constant current of $0.1A$ so the maximum current the system draws on average is $0.35A$.

From this data, we can perform the power analysis of the system as follows:

The system watt-hour consumption is given by:

$$\text{Watt-hour} = voltage \times current \times time$$

The system operates for six hours in a day. Therefore, the energy, E_{sy}, used by the system is:

$$E_{sy} = 5.3 \times 0.35 \times 6$$

$$= 11.13Wh$$

The battery's capacity in watt-hour is given be:

$$E_b = 3.7 \times 10$$

$$= 37Wh$$

From the above calculation, we can see the battery has more capacity than the system draws in a single day. Taking a maximum depth of discharge of 80% that is allowed for lithium cells, the safe energy to draw from the battery, E_{sb}, will be:

$$E_{sb} = 0.8 \times 37$$

$$= 29.6Wh$$

From the above calculations, we can see that we can meet the energy requirement of our system for more than one day without recharging the battery. The time that a battery can meet the requirement of its load without being recharged is called autonomy. The autonomy of the battery used in the system is:

$$autonomy = \frac{E_{sb}}{E_{sy}}$$

$$= \frac{29.6}{11.13}$$

$$\cong 2\ days$$

The energy supplied by the solar panel, E_{so}, is given by:

$$Average\ peak\ sunshine\ hours\ in\ Kenya = 6hours$$

$$Charging\ voltage = 4.2V$$

$$Charging\ current = 0.97A$$

$$E_{so} = 4.2 \times 0.97 \times 6$$

$$= 24.44Wh$$

This is sufficient to top up the battery after the system has drawn power in the morning. A plot of the battery voltage against time shows the battery drainage profile.

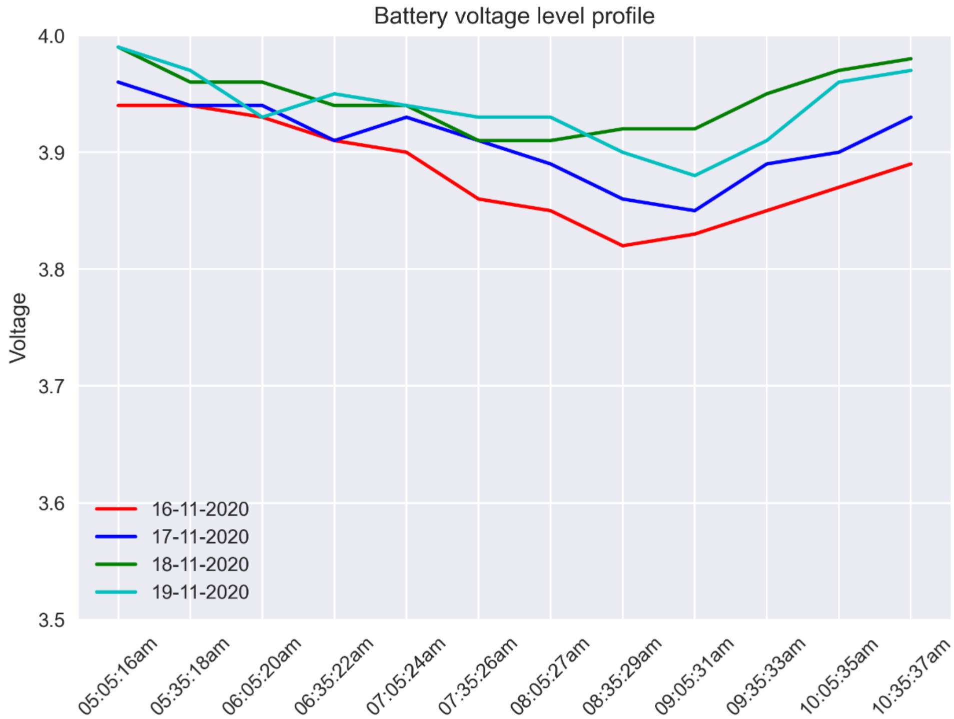

Figure 9: Voltage profiles of the battery for four consecutive days.

From figure 9, we can see the discharge profile of the battery from 5 am to around 9 am as the system draws power from the battery. From 9 am, the solar panel can provide more power than is being drawn by the system hence the voltage of the battery starts rising as it is being charged. From the plots we can also observe that the voltage of the battery starts on a higher value each morning than it was on the previous day before the system shut down. This means the battery is being recharged by the solar during the day. This profile can also be used to check the health of the battery.

4.2. Audio data collection

After deploying the acoustic system at the conservancy, we managed to collect audio recordings of birds. From the saved recordings, we can analyze the distribution of detected acoustic activities during the time the acoustic system is up. Figure 10 are plots of the number of detected acoustic activities after every thirty minutes interval from the time the system wakes up to the time it shuts down.

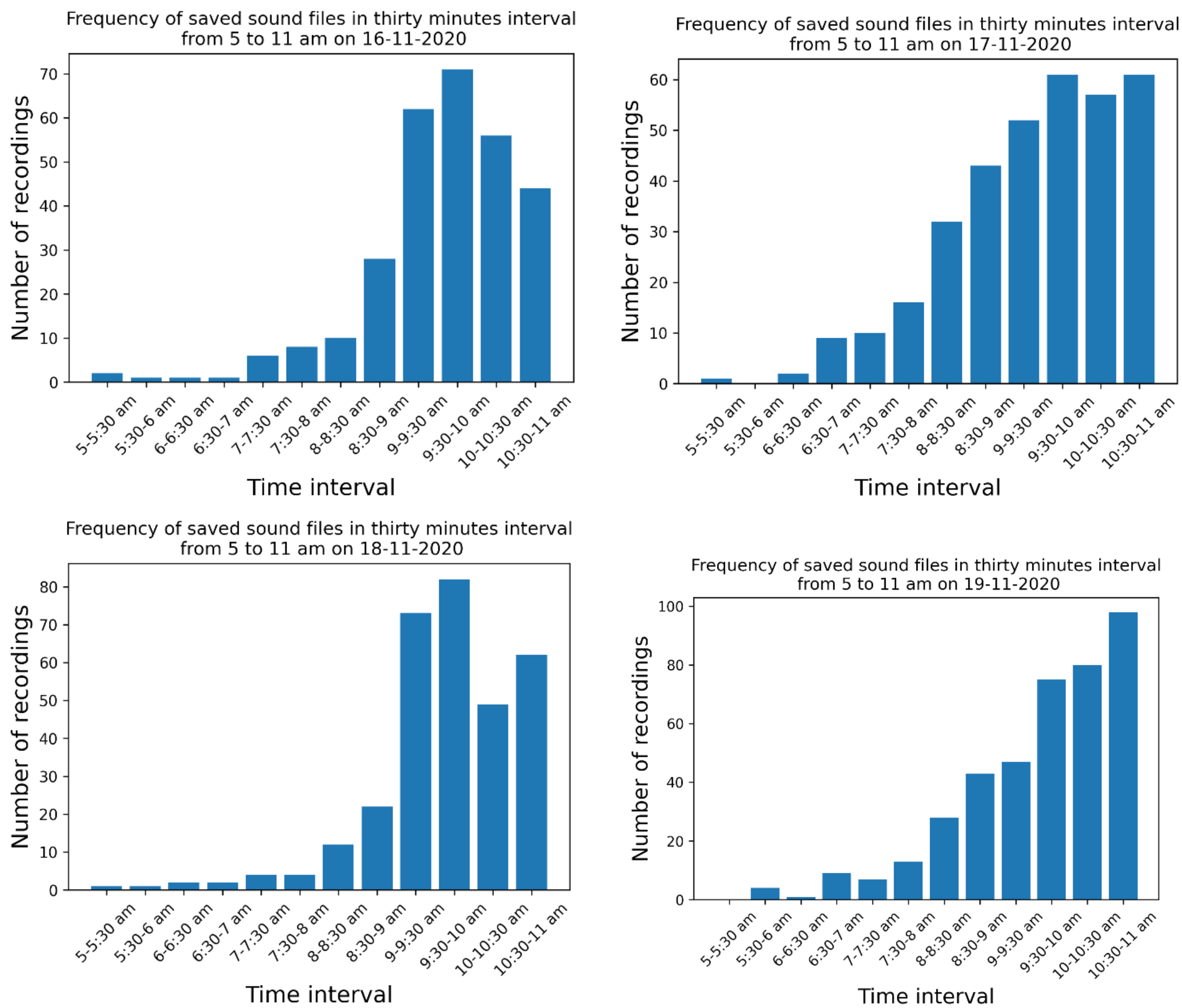

Figure 10: Frequency distribution of sound recordings during the window of operation of the acoustic system.

From figure 9, we can observe that there is less acoustic activity from 5 am to around 8 am after which there is an increase in the acoustic activities being detected by the system.

Several birds can be identified from listening to the saved recordings. Some of the identified birds are the Hartlaub's Turacos and the Grey-backed Camaroptera. A comparison between the spectrograms of the recordings of these birds and sample recordings from Xeno-canto show a lot of similarity. Xeno-canto is a website that offers a platform for sharing birds' sound recording from all over the world [14]. Figure 11 and 12 show a comparison of three seconds long spectrograms of Hartlaub's Turacos and Grey-backed Camaroptera's recordings from our system and those from Xeno-canto:

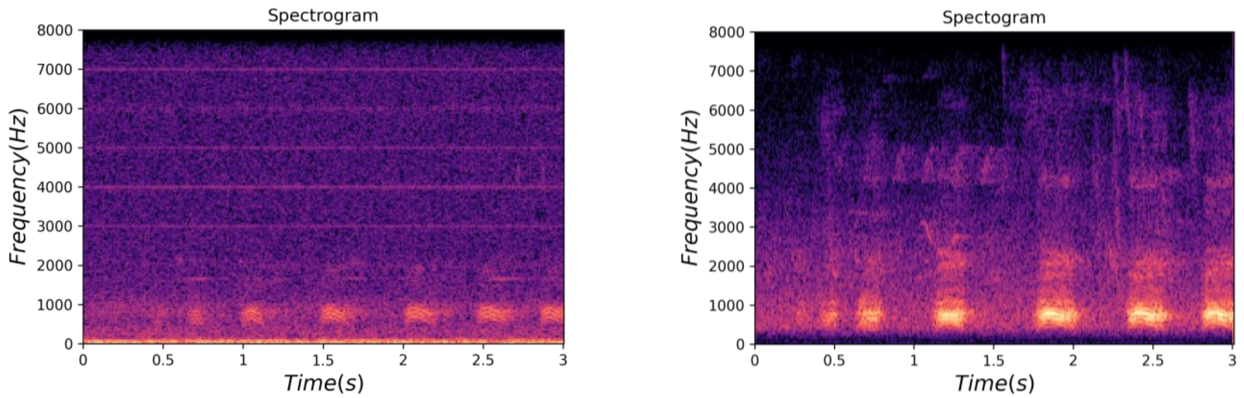

Figure 11: Spectrograms of Hartlaub's Turacos recordings from the conservancy (left) and from Xeno-canto (right).

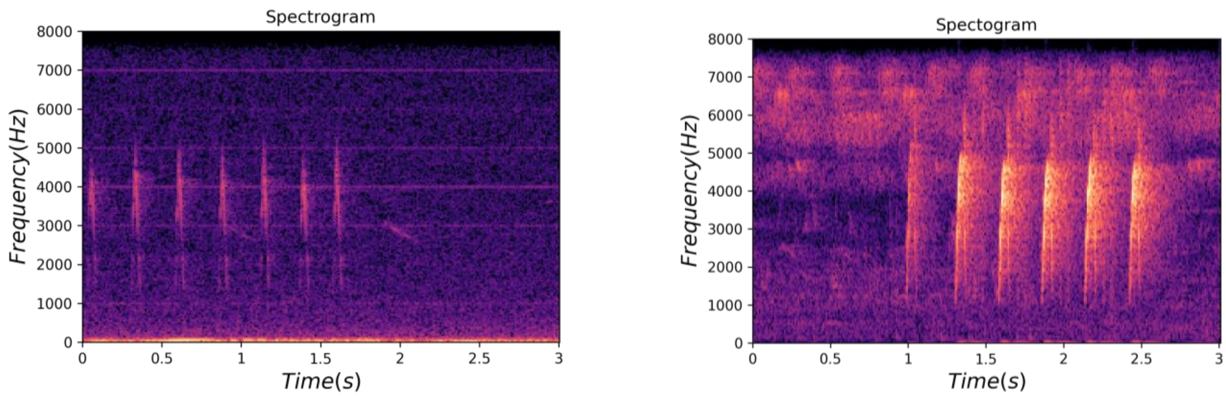

Figure 12: Spectrograms of Grey-backed Camaroptera recordings from the conservancy (left) and the spectrogram from Xeno-canto (right).

From the spectrograms above, we can see some similarity between the recordings made by our system and those from Xeno-canto. Also, the spectrogram of each species of bird is unique. Therefore, we can classify birds by recording them and then perform spectral analysis of the recordings and then treat the spectrogram as an image. This means that our system will be able to give live streams of the birds in a given ecosystem. This will greatly improve the efforts of studying and conserving ecosystems.

5. Future Work

After collecting audio samples, we intend to have them labeled with the help of an ornithologist. From there we will develop a machine learning model, train it to classify birds using the data and then deploy it in our system. We will then deploy the system in the field for automatic classification of birds from their vocalizations.

6. Conclusion

The DSAIL Bioacoustics System is a robust system for collecting birds' acoustic data for monitoring ecosystems. The system is based on the Raspberry Pi 3 single board computer that has proven to be reliable in recording good quality birds' sounds. The system makes use of solar power which is ample in the tropics. This means that the system can run for a long period of time hence will be a reliable tool for carrying out acoustic monitoring of ecosystems. Additionally, the development of the power management board that intelligently powers the Raspberry Pi and the power management program improves the reliability of the system. The DSAIL Bioacoustics System has the potential to promote use of bioacoustics in monitoring ecosystems for conservation.

7. Acknowledgment

We would like to thank the National Research Fund (NRF) for funding through the Multidisciplinary Research Award. We would also like to thank Google for funding through the Artists + Machine Intelligence (AMI) Focused Research Award. We also express our sincere gratitude to DSAIL research interns Jason Kabi and Jared Makario for their support in developing the DSAIL Bioacoustics System. We would also like to thank the wardens at the Dedan Kimathi University Conservancy for their willingness to let us set up the acoustic system at the conservancy.

References

[1] F. Briggs, Y. Huang and R. Raich, "The 9th annual MLSP competition: New methods for acoustic classification of multiple simultaneous bird species in a noisy environment," in *2013 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, Southampton, 2013.

[2] Ç. H. Şekercioğlu, R. B. Primack and J. Wormworthd, "The effects of climate change on tropical birds," *Biological Conservation*, vol. 148, p. 18, 2012.

[3] A. Brown, S. Garg and J. Montgomery, "Automatic and Efficient Denoising of Bioacoustics Recordings Using MMSE STSA," vol. 6, pp. 5010-5022, 2018.

[4] T. D. Ganchev, O. Jahn, M. I. Marques, J. M. Figueiredo and K.-L. Schuchmann, "Automated acoustic detection of Vanellus chilensis lampronotus," vol. 42, pp. 6098-6111, 2015.

[5] M. Aide, C. Corrada-Bravo, M. Campos-Cerqueira, C. Milan, G. Vega and R. Alvarez, "Real-time bioacoustics monitoring and automated species identification," vol. 1, 2013.

[6] J. Salamon, J. P. Bello, A. Farnsworth, M. Robbins, S. Keen, H. Klinck and S. Kelling, "Towards the Automatic Classification of Avian Flight Calls for Bioacoustic Monitoring," vol. 11, 2016.

[7] J. Cramer, V. Lostanlen, A. Farnsworth, J. Salamon and J. P. Bello, "Chirping up the Right Tree: Incorporating Biological Taxonomies into Deep Bioacoustic Classifiers," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[8] A. M. Solomes and D. Stowell, "Efficient Bird Sound Detection on the Bela Embedded System," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[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, pp. 308-312, 2016.

[10] P. Scherz and S. Monk, Practical Electronics for Inventors, McGraw Hill, 2016.

[11] "Frontpage - Raspbian," [Online]. Available: https://www.raspbian.org/. [Accessed 5 12 2020].

[12] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg and O. Nieto, "librosa: Audio and music signal analysis in python.," *In Proceedings of the 14th python in science conference*, pp. 18-25, 2015.

[13] S. Tang, X.-Y. Li, X. Shen, J. Zhang, G. Dai and S. K. Das, "Cool: On Coverage with Solar-Powered Sensors," in *2011 31st International Conference on Distributed Computing Systems*, Minneapolis, 2011.

[14] "About :: xeno-canto.," [Online]. Available: https://www.xeno-canto.org/about/xeno-canto. [Accessed 9 12 2020].