ECE 445

SENIOR DESIGN LABORATORY

Design Document

Distributed Species Tracker

Team No. 10

Ryan Day
(rmday2@illinois.edu)
Jonathan Yuen
(yuen9@illinois.edu)
Max Shepherd
(maxes2@illinois.edu)

TA: Hanyin Shao
Professor: Olga Mironenko

February 23, 2023

1. Introduction

1.1. Problem

Invasive species are organisms that find their way into an environment of which they are not a native. They are capable of inflicting great harm on their new ecosystems leading to the death of native species as well as significant economic damage in some cases. Removing invasive species is an incredibly intensive and difficult task, the burden of which sometimes even falls on civilians who are called to look out for the invading species in order to provide intel on their location and help prevent any further spreading. Some other common methods for invasive species control include chemical control, bringing in new predators, or even uprooting parts of ecosystems in a desperate attempt to prevent the spread of the invasive species [1].

Endangered species are creatures that are on the brink of extinction. A lot of conservation efforts are made in order to restore the population of the species, including gathering the animals and breeding them in a controlled environment, as well as monitoring them via a tracking chip or satellite [2].

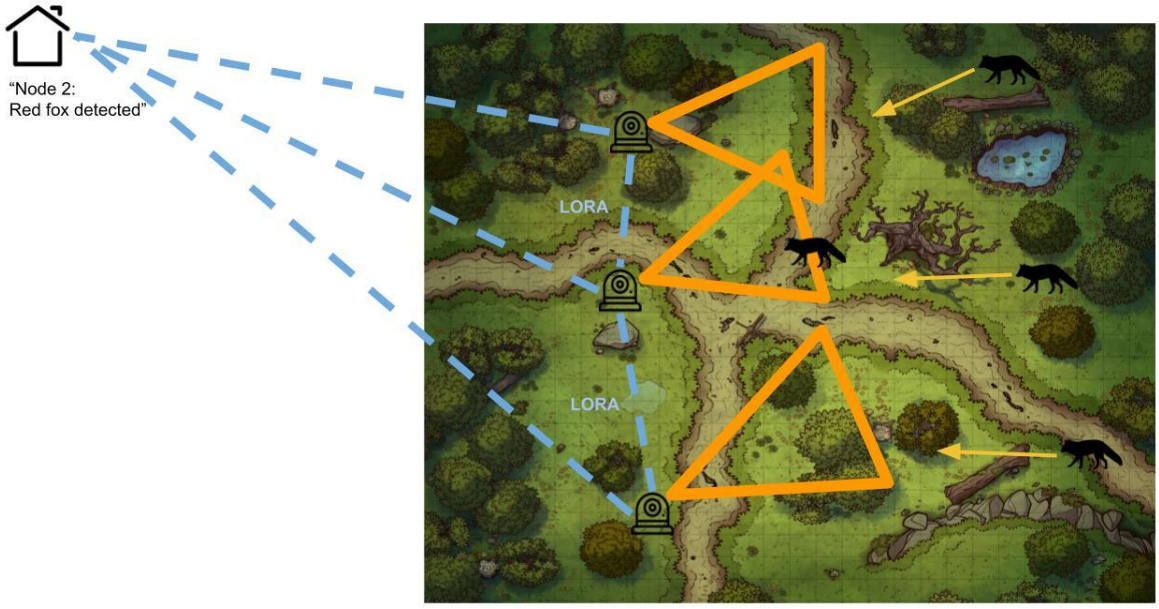
1.2. Solution

We propose a network of nodes that, once deployed in the wild, can capture images and process them to determine whether or not a species of interest has been in a certain area. The nodes will communicate with one another in order to compile a report of all of the places and times that an animal was seen. This can be an improvement on satellite imaging that is hindered by trees and overbrush and is also an improvement over the manual scouring of wilderness that is often used in the hunt of invasive and endangered species. The network, if deployed for long enough, can offer valuable data and present a comprehensive view of a species' behavior.

This semester, we aim to provide a proof of concept for this idea by building a small set of these nodes and demonstrating their ability to recognize an animal and log its whereabouts in a way that is redundant and node-failure-tolerant.

In order to do this, we will fit each node with a camera that will take images to be processed. If the species being monitored is detected, its location will be sent over the network of nodes via a routing subsystem. A power subsystem will supply and regulate power to the modules in each node. A sensor subsystem will provide GPS data and infrared detection.

1.3. Visual Aid



1.4. High Level Requirements

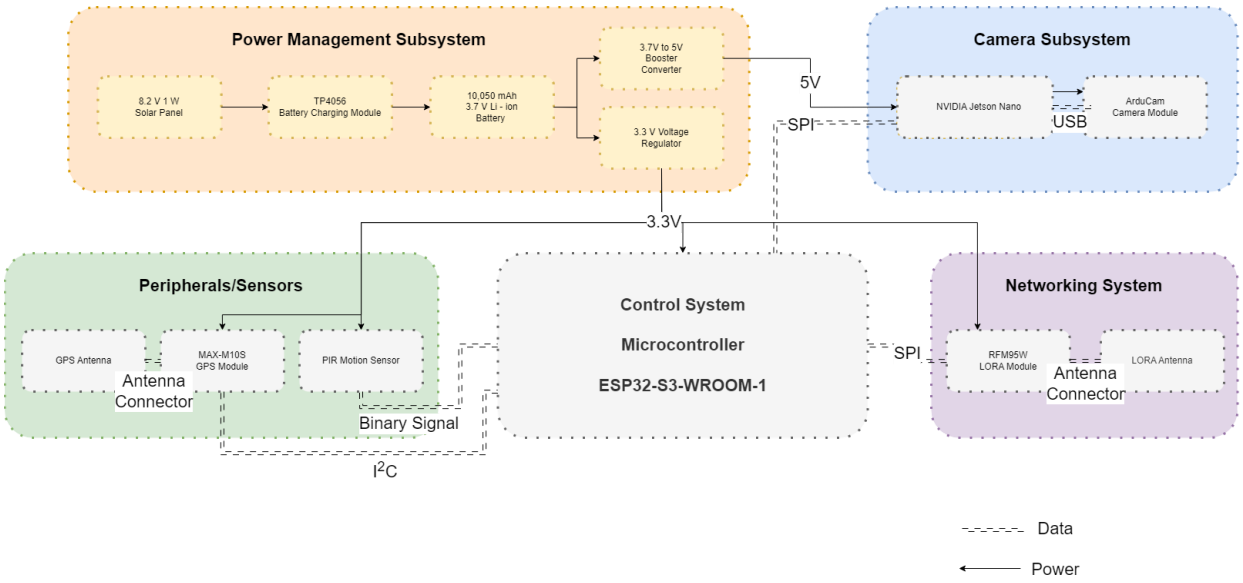
Data redundancy - We should be able to demonstrate that data gathered on any arbitrary node is reflected on the rest of the nodes in the network.

Detection accuracy - The system should be able to identify the presence of an animal with the infrared sensor and classify the animal we are monitoring with an accuracy of 70% or higher.

Battery life - The system should be able operate at constant use for at least 4 hours.

2. Design

2.1. Block Diagram

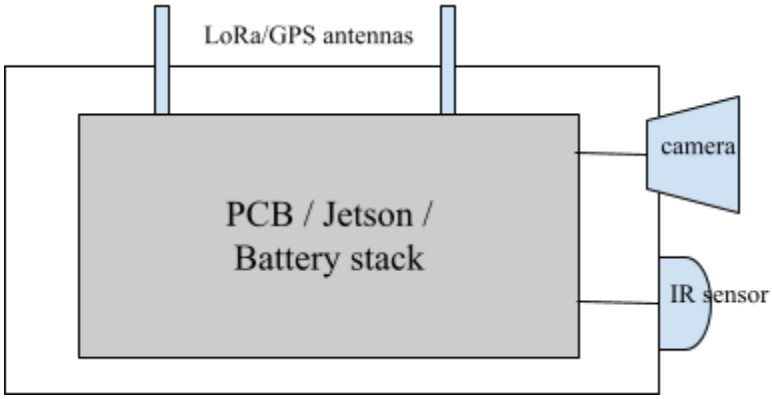


This block diagram consists of a power subsystem which will charge a battery and regulate the power supply to the rest of the node. The networking subsystem will send and receive GPS and timestamp packets and replicate data throughout the mesh network. The camera subsystem will take pictures of species and classify them. The sensor subsystem will gather GPS data and detect the presence of wildlife with an infrared sensor. These subsystems will be described in great detail in section 2.3.

2.2. Physical Design

Packaged in each node will be a Jetson Nano, our PCB board, an infrared sensor, a camera, 2 antennas, and a battery pack. The dimensions of the Jetson nano and the battery pack are 100 x 79 x 30 mm and 66.6 x 55.3 x 18.7 mm respectively. Our PCB board will be 6.4 mm at its thickest due to the edge launch SMA connector. Therefore, the overall dimensions are 100 x 79 x 55.1mm. We will need to stack these components side by side and leave room for the camera and infrared sensor to go up against the edge of the box, facing the same direction.

We wish to encase each node in a box that will need to have openings for the IR sensor, the camera, and the antennas.



2.3. Subsystem Overviews and Requirements

2.3.1. Networking Subsystem

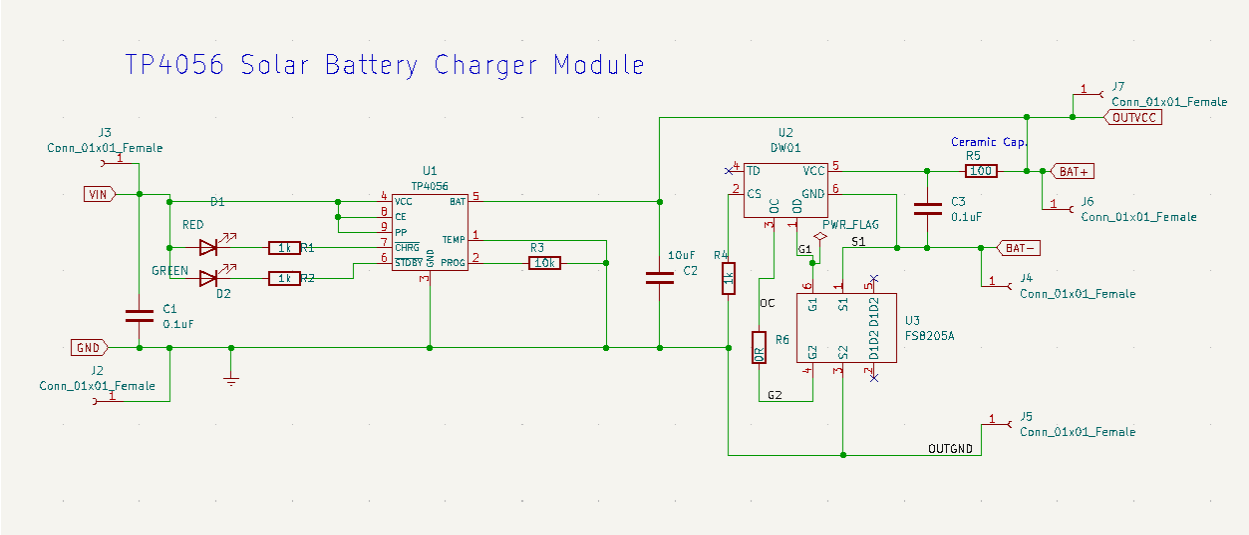
This subsystem will establish the network over which the nodes will communicate. These nodes will replicate local GPS data amongst themselves. We will use a LoRa transceiver in order to achieve the reliable transfer of relatively small byte packets over a long range. A driver for the LoRa module and algorithms for the mesh network will be implemented in software and run on the microcontroller. The LoRa module will communicate with the microcontroller over SPI to receive the data that it should handle the transmission of. This module needs a 3.3V supply from the power module and draws at least 150 mA of current.

Requirements	Verification
Each node must be able to successfully discover and connect with the network of Species Trackers.	When testing the networking modules, we will power on our LoRa modules one at a time using a DC power supply and program an “introduction” message to be sent out once the LoRa module is initialized via the ESP32 microcontroller. This introduction message should then be interpreted by another LoRa module before adding them to their membership list. All messages that an RFM95W receives goes into a buffer that can be printed and viewed over the USB serial connection we have established with our MCU. This is how we will verify our LoRa module’s ability to join and connect. These results can be compiled in a list that tracks successful message transactions.

Each node must be capable of sending the required data (GPS location, node number, timestamp) over the network or forwarding data to its neighbors to achieve data replication and redundancy. If a packet is dropped, the nodes should re-send the data so that it is received properly.	We will use a similar verification method for this requirement as the “discover and connect” requirement. We will test this requirement by sending test data packets from our RFM95W modules and verifying that the same packet arrives on the other RFM95W modules in the exact same condition in which it was sent. Again, we can verify a received message simply by probing the receive message buffer and printing its contents over USB serial. We can test the re-send functionality by sending a packet from one module, disconnecting a receiver from its power source, waiting a few seconds, and then bringing the receiver back up to power to verify that the message was received.
The range of the communication between nodes should be at least 1 kilometer.	For this verification, we can use two RFM95W modules and their respective ESP32-S3 microcontrollers and some LED’s or laptops with USB serial connection to the MCU. We can run a transceive thread on one RFM95W and a receive thread on the other. The transceiver will send a message to the receiver every second. The receiver should blink an LED, or relay the received message over serial when it receives a message. We continue this process while walking the modules apart from one another until 1 km of distance is between them. Then we can verify that messages are still being received.

2.3.2. Power Management Subsystem

This subsystem will handle the charging and discharging of a 3.7V battery to supply a steady 3.3V supply to the MCU, infrared sensor, LoRa module, and GPS module and a 5V supply to the Jetson Nano while supporting up to 2.5 A of current draw for up to 4 hours. The subsystem will handle the charging of the battery via a solar cell that is regulated with a battery charging module. The output of the battery will be regulated down to 3.3V with a voltage regulator and stepped up to 5V with a boost converter.



Requirements	Verification
The power subsystem should be able to power each node for a minimum of 4 hours if there is no sun available and support a much longer lifespan with a steady supply of sunlight.	We are using a 10,050mAh battery pack. We estimate that the total current draw of our node is 2.5A. Since there is no way of knowing how often our system will encounter animals, we assume a worst case scenario incident in which each module is operating at max current draw non-stop. This means that the battery should last 4 hours. We can verify this requirement by measuring the current draw and voltage level using a multimeter at the RFM95W when it is sending a message, the GPS module when it is packaging and sending GPS data, the infrared sensor when it is being triggered by our movement, the MCU when it is running each of its usual threads, and the Jetson when it is processing an image and verifying that these values add up to

	2.5A. We can then measure the voltage and current across our battery over the span of a few minutes of these trials and apply a regression to make sure that it is on track for 4 hours of battery life. These results will be compiled in an equation and relayed in our notebook.
The solar panel must be able to power the 3.7 V lithium ion battery.	To verify that the solar panel can power a 3.7 V lithium ion battery, we can measure the open-circuit voltage and the short-circuit current of the solar panel with a multimeter under the conditions of moderate sunlight. If the power output of the solar panel is higher than the power required to charge the battery, then we will know that the solar panel can power the battery. We can also directly measure the voltage at the terminals of our partially-drained battery after connecting it to a sun-exposed solar panel and verify that its voltage level is 3.7. These results will be stated in our notebook.
This subsystem must provide a stable 3.3 ± 0.1 V power source that supports up to 0.5A of current draw to the microcontroller, infrared sensor, the GPS module, and the LoRa module.	We can verify this requirement by measuring the current draw of each of these components while they are completing respective tasks and measuring the voltage across the output terminals of the 3.3 V regulator.
The power system must provide a stable 5V power source that supports 2 A of current draw to the Nvidia Jetson nano.	We can verify this requirement by measuring the current draw of the Jetson Nano when it is processing an image and measuring the voltage across the output terminals of the 5 V boost converter. These results can be stated in our notebook in the form of the observed equation.

2.3.3. Camera Subsystem

This subsystem will take images of its surroundings once triggered by the infrared sensor. The image will then be processed and run through a classifier that will determine whether or not the image contains the species of interest. If the classifier outputs a positive classification, the microcontroller will compile a packet of data to be sent to the networking submodule. The only

components in the camera subsystem are a camera and an Nvidia Jetson embedded GPU computer which will run the deep learning model. This subsystem connects to other subsystems through an SPI interface with the MCU, on which the MCU notifies the camera subsystem when the IR sensor has been triggered and the Jetson notifies the MCU if it has made a detection.

Requirements	Verification
The deep learning model should have an accuracy of >80% in the wild.	We can run 50 test cases where a person walks in front of the IR sensor and the detection is made and then 50 test cases where the IR sensor is triggered and a person is not within view of the camera. We can then measure the classification accuracy of our model based on this. We will redo any test cases where the IR sensor is not triggered as this does not relate to the model accuracy. These results can be compiled in a table.
The average time between the IR sensor being triggered and the corresponding message being sent should be below 100 ms.	During the experiment above, we can timestamp when the IR sensor is triggered and when the notification of a finished classification is sent using the MCU's clock. Then, we can compute this metric using this data. These results can be compiled in a table.

2.3.4. Sensor Subsystem

This subsystem will be responsible for gathering GPS data and hosting the infrared sensor. The GPS data will be processed in the MCU and packaged into the data to be transmitted by the networking submodule when needed. The infrared sensor will detect living creatures in the proximity of the node and trigger a photo to be taken by the camera.

Requirements	Verification
The GPS module must provide a location that is 10 meters away from the true location of the node	We can verify this requirement by testing the GPS module in isolation with the MCU. We will gather 20 unique GPS locations by moving the GPS/MCU system and logging the data it sends to the MCU via the I2C buses. We will simultaneously take note of the true GPS location and compare the values

	for all 20 data points, verifying that they are within 10 meters. These results can be compiled in a table.
The IR sensor must be able to trigger the camera to take a photo if a living animal is within 5 feet (on a bigger budget, we would invest in sensors with much greater range).	We can easily verify this requirement by testing the IR sensor in isolation with the camera module and hooking the system up to a debugging LED. We will measure a 5 foot distance from the IR sensor and push inanimate objects in front of it to verify that the IR sensor is not triggered. Then we will walk past the infrared sensor ourselves to then verify that the LED lights up, and that a photo is taken. These results will be stated in our notebook.

2.3.5. Microcontroller

The microcontroller will host the RFM95W drivers, as well as the software that handles communication with the GPS sensor, infrared sensor, and camera.

Requirements	Verification
The microcontroller should be able to run code that can parallelize the processing of messages coming in from the LoRa module and messages from the camera module regarding a specific sighting.	This can be tested with the ESP32-S3-DevkitC-, the Jetson GPU, and the RFM95W breakouts by flashing this software to the MCU and setting print statements or flashing an LED when a message comes in from the camera module, when a message is outbound via the transceiver, and when a message is inbound via the receiver. We can pin the transceiver handling and the camera handling threads to their own separate cores in the MCU to explicitly determine how these peripherals are interacting with the MCU. We can send relevant confirmation messages over the USB serial port to verify correct threads of execution. These results can be compiled in a table.
The microcontroller should be able to poll the GPS module or request that it send updated GPS coordinates.	We can verify this requirement by flashing a program on our MCU that interacts with the GPS module by periodically logging the input

	from the GPS module and by sending it an interrupt indicating that a location should be sent. We can mark these events with serial print statements and send the received GPS information over serial as well in order to verify that updated GPS coordinates can be sent. These results can be compiled in a table.
The microcontroller should be capable of executing the code that manages the LoRa module, the sensors, and the image processing such that the latency for gathering and processing data from the sensors does not exceed 1s. Therefore, when a picture is taken, the packet of data composed of GPS location and timestamp should be on the way to the LoRa module within 1 second so that the information pertaining to the animal sighting is as accurate as possible.	To verify this requirement, we will simply take timestamps at the point when the GPIO pin connected to the IR sensor's data line goes high, and compare it to the timestamp at the point where the next outbound message from the RFM95W is sent. These results can be compiled in a table.

2.4. Tolerance Analysis

One of the challenges of our project is reliably replicating data throughout the network of our LoRa nodes. Since the use case of our project is to drop nodes off in the wild and leave them there for an extended period of time and intermittently pull compiled data from the network, we do not have heavy restrictions on the minimum time a packet should spend in transit between two nodes ("airtime"). Instead, we care more about data transfer reliability across a long distance of at least 1 kilometer. Therefore, we need to pay attention to our link budget, which is essentially a way for us to account for all of the gains and losses of our communication system.

We can summarize all of the gains and losses in the communication between two LoRa modules with this equation.

$$RP = TP + TAG - TL - PL - ML + RAG - RL$$

RP = Received Power
TP = Transmitted Power
TAG = Transmitter Antenna Gain
TL = Transmitter Losses (transmission line, connector)
PL = Path Loss

ML = Miscellaneous Loss
RAG = Receiver Antenna Gain
RL = Receiver Loss

In order for a signal to be interpreted correctly, the Received Power needs to be greater than the received sensitivity, which for our RFM95W module, is -144 dBm. The Transmitted Power for an RFM95W is +20 dBm and the gain of our antennas is 1.2 dBm. This means that (TL + PL + ML + RL $<$ 166 dB). We will ensure that our transmission line is impedance matched with our antennas, which should minimize the losses TL and RL. The biggest contributors to our loss will be the path loss.

The equation for free space loss based on distance is $20log_{10}(4\pi d/\lambda)$, where d is the distance between the two antennas and λ is the wavelength [3]. The wavelength of our 915 Mhz signal is 33cm. To satisfy our 1 km range requirement, our free space loss will be 91.6139 dB. This still leaves a budget of roughly 75 dB to be lost through external factors such as physical obstructions.

We can also make our data transfer more reliable and improve our robustness in the face of noise and interference, by configuring values like the spreading factor and the coding rate of our packets. The spreading factor represents the number of spreading chips per data bit and a higher spreading factor means that more chips are used to spread each data bit, which results in a lower data rate but a higher processing gain. The processing gain is the ratio of the signal's bandwidth to the data rate and it represents the enhancement of the signal-to-noise ratio (SNR) that results from the spreading process. The higher the processing gain, the higher the signal's SNR and the higher the reliability of the transmission [4]. LoRa has the added ability of being able to demodulate signals -7.5 dB to -20 dB below the noise floor, making it an excellent choice for reliable transmission.

The coding rate is also configurable and denotes the ratio of data bits to redundant bits that can be used to help correct messages being received. A higher coding rate means that there are more redundant bits in the packet for each data bit, increasing the packet's durability in the face of interference.

We can configure our LoRa modules so that they have a spreading factor of 9, a coding rate of 4/8, and a bandwidth of 125 kHz. Assuming an average payload size of 50 bytes (16 bytes for latitude and longitude coordinates, 25 bytes for the timestamp, a byte for the node number, remaining bytes for a description of the classified animal), this yields an airtime of 476.16 ms [5]. This configuration is not incredibly fast but will make sure that our packets are sent reliably.

3. Cost and Schedule

3.1. Cost Analysis

Component	Manufacturer	Quantity	Unit Price	Total Price
RFM95W LORA RADIO TRANSCEIVER BR 1528-1667-ND	Adafruit	3	$19.50	$58.50
LoRa module RFM95W-915S2	RF Solutions	3	$14.50	$43.50
ESP32-S3-DevKitC-1-N8R8	Espressif Systems	2	$15.00	$30.00
ESP32-S3-WROOM-1U-N16R8	Espressif Systems	3	$4.20	$12.60
RF ANT 916MHZ WHIP STR SMA MALE ANT-916-CW-HW-SMA-ND	LINX TECHNOLOGIES INC	4	$9.89	$29.67
Edge-Launch SMA Connector for 1.6mm	Adafruit	3	$2.50	$7.50
USB micro connector	Adafruit	3	$2.95	$8.85
ESD protection USBLC6-2P6	STMicroelectronics	3	$1.00	$3.00
MAX-M10S MAX-M10S-00B	U-blox	3	$21.00	$63.00
PIR motion sensor	Adafruit	3	$9.95	$29.85
22uF capacitor GRM21BR60J226ME39L	Murata	3	$0.29	$0.87
1 uF capacitor TCKIX105CT	Cal-Chip Electronics, Inc.	3	$0.29	$0.87
0.1 uF capacitor T491A104K035AT	KEMET	9	$0.62	$5.58
10k Ohm resistor RT0603FRE0710KL	Yageo	3	$0.10	$0.30
Battery charging IC TP4056	Seeed Technology Co., Ltd	3	$4.90	$14.70
3.3V regulator 579-TC1264-3.3VDBTR	Microchip Technology / Atmel	3	$0.92	$2.76
5V boost converter LM2585S-12/NOPB	National Semiconductor	3	$6.17	$18.51
Barrel connector 10-01935	Tensility International Corp	3	$3.23	$9.69

Battery pack Adafruit Lithium Ion Polymer Battery - 3.7V 10050mAh (10 Ah)	Adafruit	3	$29.95	$89.85
Solar panel 313070005	Seeed Technology Co., Ltd	3	$12.30	$36.90
Jetson Nano Developer Kit	Nvidia	3	$149.00	$447.00
Total Components Cost				**$913.50**

The average UIUC computer engineering graduate makes $105,352 per year, which equates to an hourly salary of $50.65. We estimate that we are spending and will spend roughly 10 hours a week on this project over the course of 10 weeks. Therefore, each of our team members will command $5,065 for a total team salary of $15,195.

The total costs associated with this project come out to $16,108.50.

3.2. Schedule

Below is a tentative schedule of deadlines, tasks, and goals. Tasks for the weeks in the more immediate future are divided into finer grain subtasks. After our PCB is ordered and delivered, our team will have to work more as a single unit in order to assemble and debug the fully compiled system.

WEEK	Deadlines	Tasks
Week of 2/20:	Design Document due 2/23 Team contract due 2/24	**Max** : Finalize what board will be used for image processing and what camera will be used **Jon** : Confirm IR component and order, fix block diagram, familiarize with MAX-M10S **Ryan** : Continue development on LoRa communication; send, receive, reply messages between two boards; step back distance **Together** : Finish first PCB design iteration over the weekend
Week of 2/27:	Design Review on 2/27-2/29 PCB Review on 2/28	**Jon** : Get the MCU to acknowledge IR trigger **Ryan** : Send messages from Jetson to the MCU and from the MCU to the Jetson **Max** : Take picture with camera, process on

		Jetson, and send classification to MCU **Together** : Respond to criticism from PCB review and fix design
Week of 3/6:	First round of PCBs ordered on 3/7 Teamwork Evaluation due 3/8 Finalize machine shop specs by 3/10	**Ryan** : Write the software that allows us to form a mesh network between multiple LoRa modules **Jon** : Work on GPS software. **Together** : - Finalize functionality of LoRa network, Jetson interface, and IR sensor with our breakout board/breadboard configuration - Have every component ordered
Week of 3/13:	Spring Break	
Week of 3/20:		**Together:** - Solder PCB and debug PCB design - Verify that we can flash the MCU - Verify that we can provide power to the components of our board - Flash software and test functionality (LoRa, GPS, camera, IR sensor)
Week of 3/27	Have second round of PCB's ordered by 3/28 Progress report on 3/29	**Together:** - Continue debugging and fixing issues - Finalize updated PCB design
Week of 4/3:		- Solder new PCB and debug
Week of 4/10:	Teamwork evaluation due 4/14	**Together:** - Run through the functionality of the project within our team and prepare for mock demos
Week of 4/17	Mock demo on 4/18	**Together:** - Make tweaks for the final demo - Work on presentation
Week of 4/24:	Final Demo and Mock presentation	**Together:** - Finalize presentation and report

4. Ethics and Safety

1. Our project is designed to interact with nature and stay outdoors for extended periods of time. It is important that the contents of our tracker nodes, including lithium ion batteries, are well-contained in order to avoid pollution or the harming of animals. We will ensure that our batteries, PCBs, and LoRa modules are encapsulated safely in a container so that it can be deployed and then recovered in a way that leaves no trace on the environment in which it was stationed. By doing so, we are doing our best to comply with the IEEE code of ethics that calls for "ethical design and sustainable development practices" [6].

2. When using radio frequency transmission, there are a number of issues that can arise such as interference with other signals. When designing our networking module, we will make sure that we are transmitting at an unrestricted frequency and acting in full compliance with the FCC. Our project's use case is to be deployed in relatively remote areas, so the risk of interfering with outside signals is low.

3. When working on our project, we will abide by the IEEE code of ethics by being open to criticism of our work from our teammates, TA's, and professors. We organized set meeting times throughout the week designed to keep ourselves up to date on the progress made by our teammates and created a shared Gitlab project that will host our code in an effort to maintain technical transparency. We will also always be ready to pivot and embrace different design parameters and restrictions if our research turns up ethically-binding reasons to do so. Before starting work on our project, we completed lab safety tutorials and received nominal training in the areas of PCB design and soldering to make sure that we are undertaking only the tasks for which we are qualified. Finally, while working in a group, we will treat each other fairly and respectfully, culturing an environment that welcomes the exchange of ideas and promotes productive, enjoyable work.

REFERENCES

[1] “Control Mechanisms | National Invasive Species Information Center,” *Invasivespeciesinfo.gov*, 2019. https://www.invasivespeciesinfo.gov/subject/control-mechanisms [Accessed:09-Feb-2023].

[2] A. New, “A New Way to Track Endangered Wildlife Populations from Space,” *Yale E360*, 2021.https://e360.yale.edu/digest/a-new-way-to-track-endangered-wildlife-populations-from-space [Accessed:09-Feb-2023].

[3] “Friis Free Space Loss Equation,” *Gmu.edu*, 2023. http://mason.gmu.edu/~rmorika2/Friis_Free_Space_Loss_Equation_.htm [Accessed:23-Feb-2023].

[4] A. Jebril, A. Sali, A. Ismail, and M. Rasid, “Overcoming Limitations of LoRa Physical Layer in Image Transmission,” *Sensors*, vol. 18, no. 10, p. 3257, Sep. 2018, doi: https://doi.org/10.3390/s18103257 [Accessed:09-Feb-2023].

[5] “LoRaTools,” *Loratools.nl*, 2023. https://loratools.nl/#/airtime [Accessed:09-Feb-2023].

[6] “IEEE code of Ethics,” *IEEE*. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html [Accessed:09-Feb-2023].