Single-Gateway Multi-Node Mesh Network for a Natura Region.

1st Glykos Dimitris
Technical University of Crete
Electrical and Computer Engineering
Chania,Greece
dglykos@isc.tuc.gr

2nd Papadaki Aggeliki
Technical University of Crete
Electrical and Computer Engineering
Chania,Greece
apapadaki4@isc.tuc.gr

3rd Papadopoulos Georgios
Technical University of Crete
Electrical and Computer Engineering
Chania,Greece
gpapdopoulos7@isc.tuc.gr

5th Perakis Emmanouil
Technical University of Crete
Electrical and Computer Engineering
Chania,Greece
eperakis@isc.tuc.gr

4th Perakis Georgios
Technical University of Crete
Electrical and Computer Engineering
Chania,Greece
gperakis2@isc.tuc.gr

66th Siolis Panagiotis
Technical University of Crete
Electrical and Computer Engineering
Chania,Greece
psiolis@isc.tuc.gr

7th Stavrou Odysseas
Technical University of Crete
Electrical and Computer Engineering
Chania,Greece
ostavrou@isc.tuc.gr

8th Steiakakis Ioannis
Technical University of Crete
Electrical and Computer Engineering
Chania,Greece

9th Toganidis Nikos
Technical University of Crete
Electrical and Computer Engineering
Chania,Greece
ntoganidis@isc.tuc.gr

10th Vounatsos Vasilis
Technical University of Crete
Electrical and Computer Engineering
Chania,Greece

***Abstract*—Elafonisi beach, located in Chania Crete, is categorized as a Natura Region, containing more than 100 rare species of plants. The region of Chania and particularly Elafonisi, is a tourist attraction, visited by thousands of people annually. Although, the protected flora of Elafonisi has been fenced, people continue to pass through these fenced areas, damaging the flora. In order to protect these areas, we propose a Mesh Network of Nodes equipped with sensors, which is used to detect and count the number of people that access this region in a specific time interval. Our system can be utilised in order to record statistics of the people present in these areas, and notify the authorities, if necessary.**

***Index Terms*—Bluetooth Mesh Network, ThunderBoard Sense 2, Neural Network**

I. Introduction

Our team implemented a network of distributed nodes, using the projects Bluetooth Mesh - SoC Sensor Client[1] and the Bluetooth Mesh - SoC Sensor Server [2] provided by Silicon Labs [3]. For the deployment of the network of nodes, five Thunderboard Sense 2 development platforms and a single Raspberry Pi single board computer, were used. The Thunderboard Sense 2 platform, contains a plethora of available sensors (Relative Humidity and Temperature Sensor, UV and Ambient Light Sensor, Pressure Sensor, Indoor Air Quality and Gas Sensor, 6-axis Inertial Sensor, Digital Microphone, High brightness LEDs and Hall-effect Sensor). From the aforementioned sensors, the proposed system utilises the Ambient Light Sensor, 6-axis Inertial Sensor and the Digital Microphone.

II. Deployment

A. Client

The role of the Client [1] was assigned to a Thunderboard. This board makes sequential requests for data, to all the nodes in the network. Each node(server) [2], replies to their received request, by sending the requested data. These data are encapsulated in a packet and are sent to the gateway using serial communication. Each packet contains the header, the address of the sender, the type of the data and the data. The packet size and the frequency of transmission, was adjusted for the needs of this project.

B. Server-People entering/exiting

A mechanical construction fused with Thunderboard Sense 2 ICM-20648 6-Axis Inertial Sensor was devised, in order to count the number of people entering the protected areas on foot. The backbone of the mechanism is based on the concept of the angular motion of a wooden rectangular slate with respect to its center of rotation which refers to an axis parallel to the short dimension of the slate. In order to sense a person who steps on the slate, entering the area, the aforementioned motion sensor was used to convert mechanical force to angle.

Mechanical Section: The slate was designed to accommodate one person at a time, without them being able to enter the area without stepping on it. In particular the slate's dimensions are 120X60X3 (length, width, height). The slate stands on 4 legs, each leg consisting of a screw, a spiral that allows it to return back to its initial state, and an anti-vibration mounting that prevents bouncing. The axis(18 mm) is found underneath the middle of the slate, parallel to the short dimension of it, and rotates with the help of two ball bearings.

Electrical Section: The 6-Axis Inertial Sensor is embedded on the Thunderboard Sense2 which is fixed and aligned on the slate. The Thunderboard is powered by a power bank. This sensor, periodically measures the rate of change of the slate angle, by sampling every 10ms. These samples in turn, are used to calculate the integral of the angular velocity, to compute an angle in degrees. The chosen high sampling rate, results in high precision in the computation of the integral and thus provides our system, with better sensitivity. The last computed value of the angle is fed back to the system and is utilised to calculate the next value. Furthermore, a calibration is performed periodically, so as to have a consistent angle when the slate sits idle.

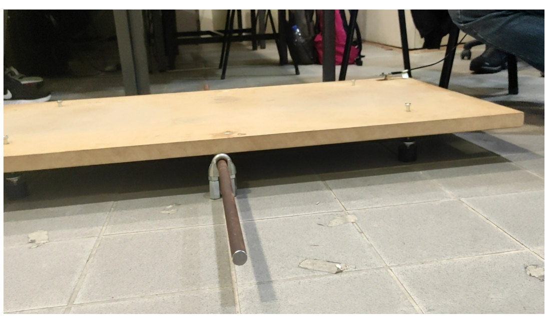

Fig. 1. Mechanical Section:

C. Server-Vehicles Entering/exiting

The vehicle detection was based on the ambient light and UV Sensor of the Thunderboard. More specifically, the sensor acquires a light measurement every 200 ms. This measurement is placed in the head of a queue that can store up to 10 measurements.The mean value of the measurements currently stored in the queue is calculated. A threshold is computed, equal to the mean value multiplied by a constant. This constant is used to avoid mistaken measurements, when a shadow covers the sensor. Its value was determined experimentally at the value of 0.1. Each new measurement, is compared to this threshold and if the value of the light is below that threshold(a vehicle is passing above the sensor) and depending on how many times the sensor measures the same value (how long the vehicle is above the sensor) the type of the vehicle can be defined. If the measured value is over it, that means that no vehicle has passed. The two categories of vehicles are buses and cars. An approximation of the number of people was made (the worst case as the vehicles are expected to be full), about 40 for the bus and 4 for the car. Each time the client makes a request the server sends the total number of people that have entered/exited. It is worth noting that two Thunderboards are going to be placed on the (double traffic- one on each lane) road of Elafonisi. The experiment was performed in a different case scenario, were ambient light existed in the atmosphere, as it took place between 3:00 to 5:00 A.M. The results were of an accuracy of about 90% .

D. Server-Present People

The microphone Sensor, present in the Thunder-board Sense, acquires buffers of microphone data and sends them to the client after each request.

E. Neural Network

For the neural network that identifies present people in a space, a significant amount of data of people talking near the Thunderboard were collected (divided in 3 classes, depending on the number of persons that are nearby) and also of ambient noise (4 classes overall) from the air and the sea. Note that some of the ambient noise data must include the sound of human talk in the distance, in order to imitate the real conditions. For the construction of the model, a division of the samples was made in windows of 1 second in size and with 0.1 second increments for each new sample. The MFE processing block on the Edge Impulse platform [4] was used to build the model, which extracts a spectrogram from audio signals, using Mel-filterbank energy features. As the site says, it is great for non-voice audio, which at a first glance might seem questionable, but given the fact that the purpose is not to recognise what the person says, but in reality to identify his presence in the area, it is the preferred solution. Also, a NN Classifier was used, which is the only available solution. Overall, approximately 30 minutes of data have been divided in the training and testing sets and the accuracy of the model is an acceptable 62.1%. The most notable and impactful error that the model has is misclassifying as "Ambient Noise", 59.3% of samples in "Humans Near 1-4" class. Beneath can be found the full Confusion Matrix from the current model.

	Ambient Noise	Humans Near 1-4	Humans Near 5-10	Humans Near 11 plus
Ambient Noise	96.3%	3.6%	0.1%	0%
Humans Near 1-4	59.3%	34.8%	2.1%	3.7%
Humans Near 5-10	18.4%	37.0%	34.5%	10.1%
Humans Near 11 plus	0.4%	10.5%	0.4%	88.7%
F1 Score	0.76	0.37	0.5	0.77

Fig. 2. Confussion Matrix

To conclude, there is still room for improvement and more data to be recorded. The recordings took place in the campus of TUC and in Kalathas beach in Chania, trying to simulate as much as possible, the conditions in Elafonisi.

F. Gateway

The connection between the gateway and the Client (in the Mesh Network) is a simple UART connection (that also can be configured with Parity, to avoid malformed data).

Each custom packet starts with a preliminary 2 byte hex code of 0xA0A0 and ends with 0x0D0A. This allows us to send an arbitrary amount of data and multiple data segments, without worrying about the size of each segment (As long as we know the structure of the packet). Each segment is separated with a value of 0x20 as this makes it easier to parse the packet later.

The packet consists of 3 useful segments (that come after the two starting bytes):

PID (Packet IDentifier) Number: 2 Bytes, Address: 2 Bytes, Data: Depends on PID.

Having a PID Number, allows a unique distinction among multiple Sensor Samples. In our case three PIDs were used (PEOPLECOUNT, MOTIONSENSED, VOICE).

In addition the usage of the Address segment allows us to use the same PID Number with a different processing method, For example, there is one sensor stationed in both the Entrance and Exit of the park that both transmit the same PID of PEOPLECOUNT but with a different address. Furthermore, with the right code modifications we can create hotspots based on the data of each sensor.

Something that we found out during the final testing phase of our project, was that the publishing process the boards initiated at startup, yielded false information across the UART connection. This was resolved by adjusting the code to parse packets only from valid (PID, Address) pairs.

After receiving a full packet, the server splits it to its useful segments and checks the validity of each section. Depending on the PID of the packet it's processed accordingly. For packets that carry information about the people entering / exiting the park a simple value is updated in a remote database. For packets that carry Voice samples, their data is appended into a buffer, and when the buffer reaches a specified length (the input window of the Neural Network) it is sent to the Impulse Runner(as input to the Neural Network) to classify the samples. The results of the classification process are updated in the same remote Database. The Impulse Runner is taken directly from Edge Impulse's Python Software development kit (SDK) examples located in their GitHub Repository.

For the Data Visualization(Fig. 3., a separate Flask Server is running alongside the UART Receiver that constantly probes the remote Database and creates a constant Data Stream that updates a simple website. The server hosts a website that grabs data from a Python Generator Object, which in turn probes the database each 250ms.

Technologies Used: VS Code, Python 3.10, Python Flask, Docker: PostgreSQL, PG Admin. Any Single Board Computer (SBC) with internet access,in this case a Raspberry Pi, can be used as a UART receiver (since the database can be anywhere).

G. Network

In order for the nodes of the network to communicate with each other, the act of provisioning was needed. More precisely, we constructed a network and assigned each node in the same group and with a specific functionality. This was a low complexity task, as an android application of Silicon Labs[3] was used, called Bluetooth Mesh.

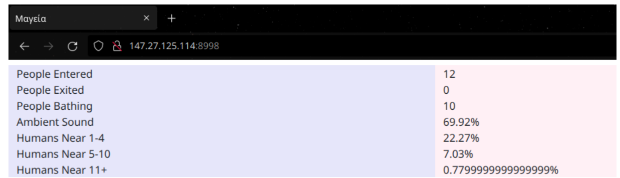

Fig. 3. Data Visualisation.

Acknowledgment

"Bletsas Aggelos, thanks for providing coordination and the Telecommunications Lab in the Technical University of Crete. Vardakis Iosif and Vougioukas Georgios thanks for assisting .

References

[1] Bluetooth Mesh - SoC Sensor Client
[2] Bluetooth Mesh - SoC Sensor Server
[3] https://www.silabs.com/
[4] https://www.edgeimpulse.com/