Cat Call Demo

William Braga, Savannah Hearn

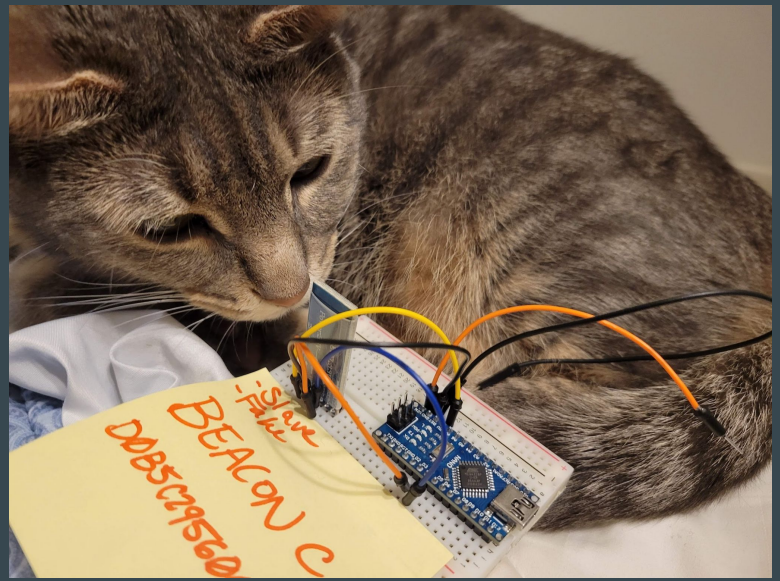
Georgia Tech

Motivation

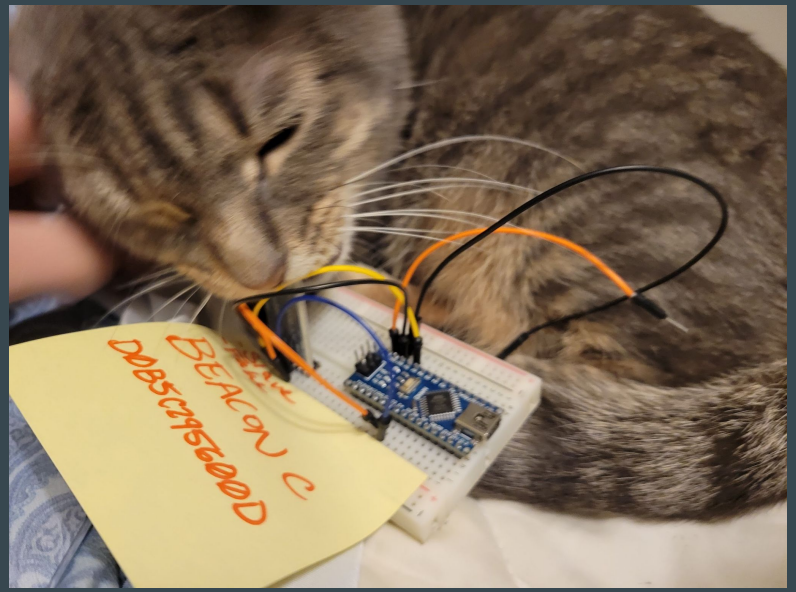
- A two-cat household with disagreements
- Cats are allowed in each other's territory under certain conditions

IACUC

This project has been approved by the IACUC for conducting research on Fe



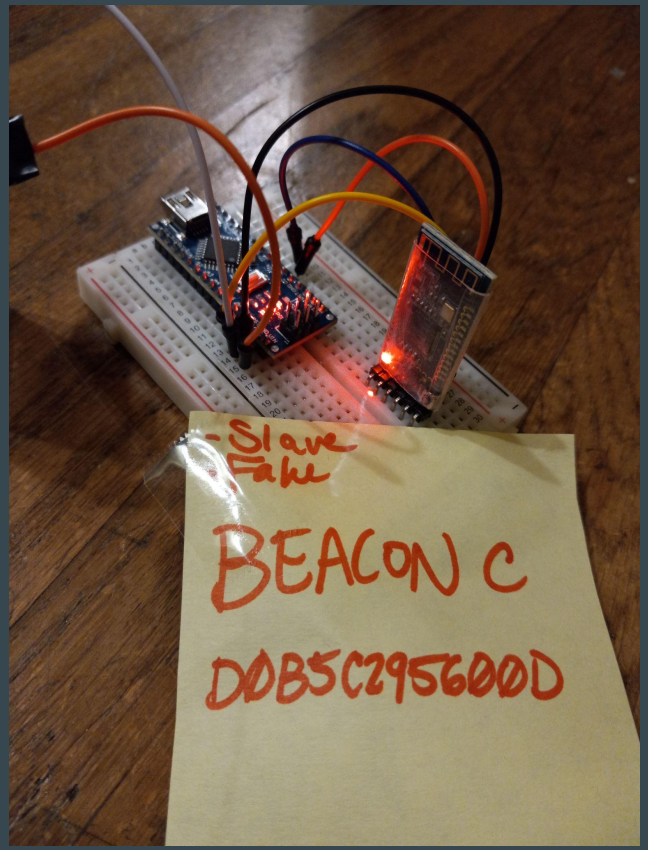
Feijoada (Fe) - Tracked Cat



Fe attempting to eat BLE module


ster
uthentic
COLLAR

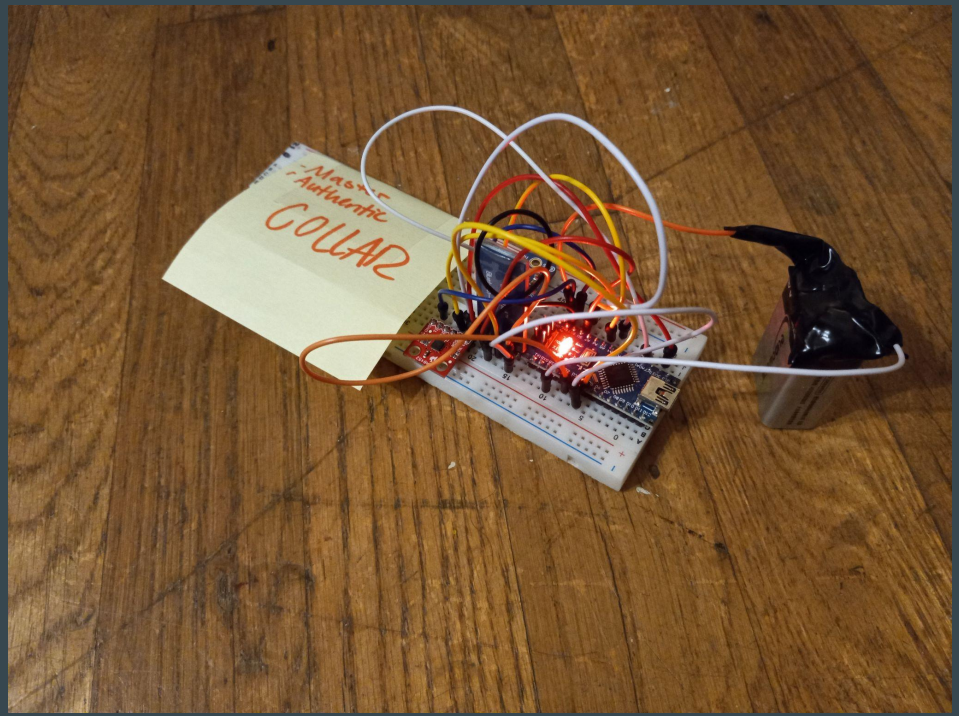
Beacons



Arduino Nano
(Off-brand) BLE module
Battery

Setup as a slave device to the harness circuit

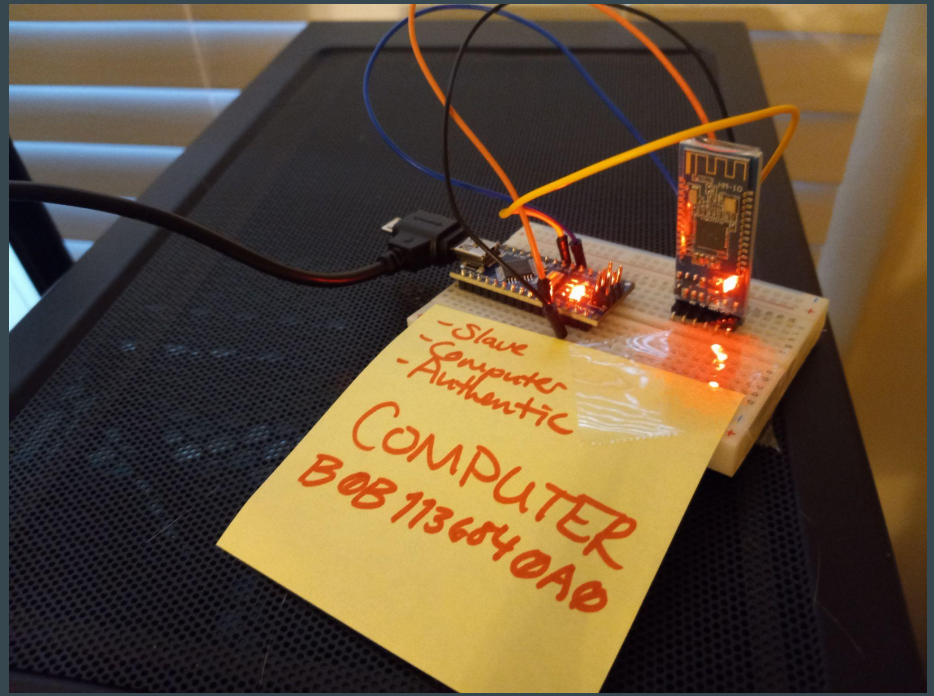
Harness Circuit



Arduino Nano
Accelerometer
Microphone
Battery

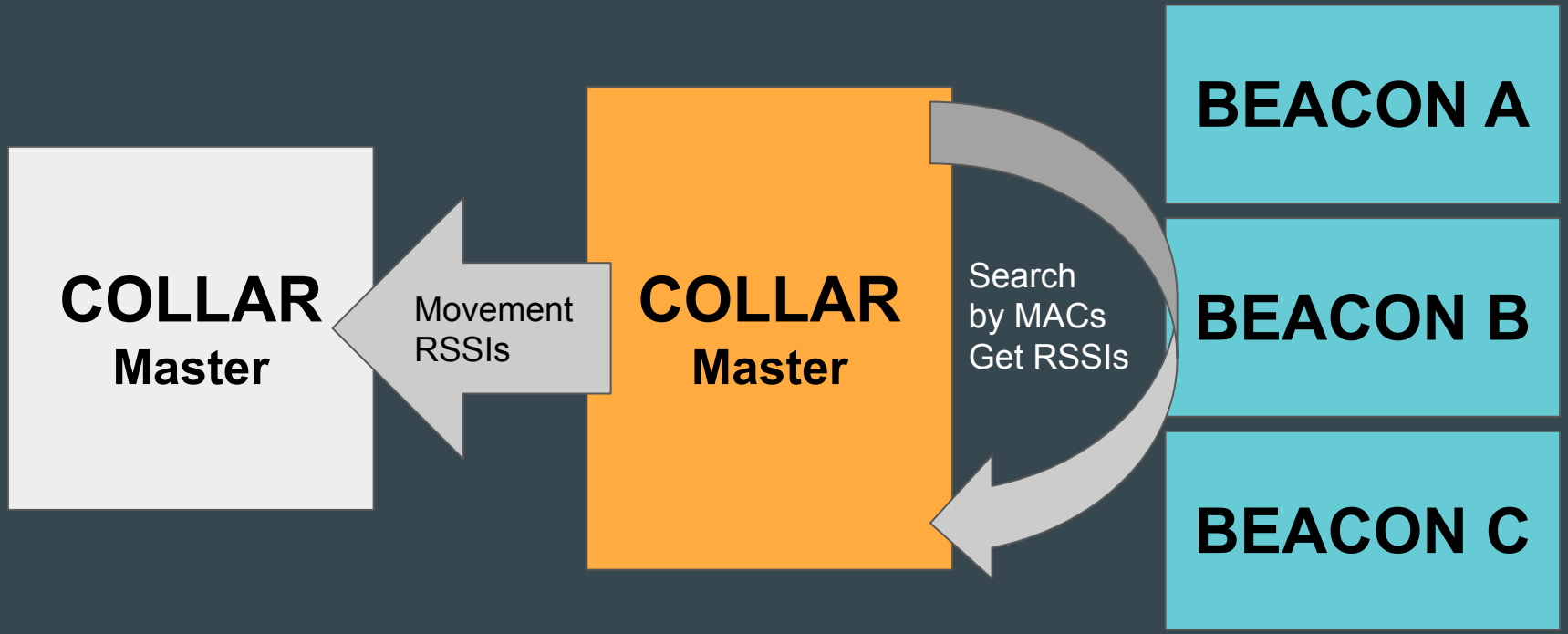
Setup as a master device, queries the beacons and sends data to the desktop arduino

Desktop Arduino Connection



Arduino Nano
BLE Module

Slave device; sends data to the python server using a serial connection


COLLAR
Master
Movement
RSSIs
COLLAR
Master
Search
by MACs
Get RSSIs
BEACON A
BEACON B
BEACON C

Server Await

```
#shared data queue between io and processer
queue = Queue()

#run io and processer in parallel
arduino_io = ArduinoIO(queue)
process_data = ProcessData(queue)
serial_io = Process(target=arduino_io.run)
serial_data = Process(target=process_data.run)

serial_io.start()
serial_data.start()

try:
    serial_io.join()
    serial_data.join()

except Exception:
    print('Disconnected')
    serial_io.terminate()
    serial_data.terminate()
    client.close()
```

Main Logic:

The server continually awaits a connection from a client. Once found, two threads are created: one to receive information from the arduino on the serial port and one to process the data

‘queue’ is a shared data structure between the threads to push and pop incoming IO.

Server Data Processing

```
if measure == 'position':
    pos_process = Process(target=position.run, args=(rssi, acc))
    pos_process.start()
    self.processes[measure] = pos_process

elif measure == 'motion':
    motion_process = Motion(target=motion.run, args=(position.pos,))
    motion_process.start()
    self.processes[measure] = motion_process

else:
    sound_process = Sound(target=sound.run, args=(amp,))
    sound_process.start()
    self.processes[measure] = sound_process
```

Create new processes to calculate position, motion, and volume in parallel
Parallelism will become critical when Kalman filters and ML models are added

Server - Finding Position

```
d = []
for i in range(len(self.beacons)):
    A = self.tx_power[i]
    n = self.n[i]
    d.append(10 ** ((A - rssi[i]) / (10 * n)))
    X = fsolve(self.func, [0, 0], args=(d,))

return X
```

$$RSSI = -(10 \times n) log_{10}(d) - A \qquad (4)$$

https://www.rn.inf.tu-dresden.de/dargie/papers/icwcuca.pdf

Converts the rssi values into distances using the formula from the research paper cited below.

The distances are then combined with the beacons positions into a set of equations, solved using scipy.optimize

Android - TCP Input

```java
@Override
public void run() {
    try {
        System.out.println("Connecting...");
        Socket socket = new Socket("192.168.0.112", 50000);
        System.out.println("Connected");
        BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
        String line;
        while (!Thread.currentThread().isInterrupted() && (line=in.readLine()) != null) {
            if (line != "") {
                listener.onData(this, line, this.main);
            }
        }
        socket.close();
        in.close();
    } catch (Exception e) {
        e.printStackTrace();
        System.out.println("Error!");
        System.exit(1);
    }
}
```

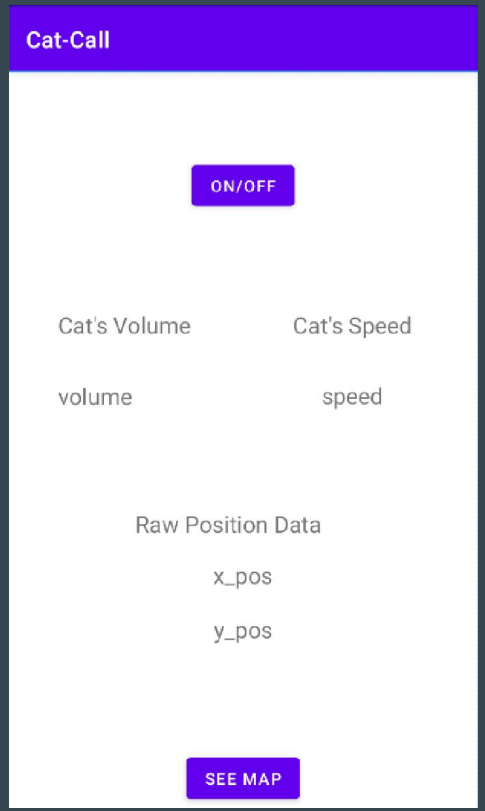
IO Thread awaits TCP messages sent from the server and then handles data parsing

float x coordinate
float y coordinate

String speed: {still, stalking, walking, running}

String volume: {quiet, moderate, loud}

Main Activity



On/Off button controls whether to receive TCP messages from the server and update the data on screen

The TextViews show the server-processed data in plain text

The See Map button opens a new activity that displays a map of the house and a cat icon for wherever Fe is located

Android Cat Map

```
if (room.equals("living room")) {
    changeConstraints(695, 435);
} else if (room.equals("kitchen")) {
    changeConstraints(300, 975);
} else if (room.equals("abed")) {
    changeConstraints(695, 975);
} else if (room.equals("corridor")) {
    changeConstraints(1200, 860);
} else if (room.equals("bathroom")) {
    changeConstraints(1230, 1100);
} else if (room.equals("wbed")) {
    changeConstraints(1650, 550);
} else if (room.equals("tbed")) {
    changeConstraints(1800, 1050);
} else if (room.equals("unknown")) {
    cat.setVisibility(View.INVISIBLE);
}
```

Using the x and y coordinates, the program predicts the room the cat is in. The room is then sent to the Map Activity and the position of the cat ImageView is updated