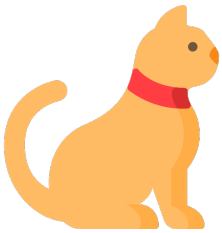


EXERCISE 3

Internet of Things

Shelby Benton | Spring 2020

EXERCISE OVERVIEW

Goal & Requirements

The goal of this exercise was to **capture data from the environment**, and **create outputs** based on that data. The exercise had several requirements:

- Capture data from at least two different sources using the Arduino or other technologies
- Output the data to two different networked services

Scope

Because I was familiar with the Arduino and its capabilities, I chose to use its sensors to capture my desired data. My sensors included a **simple button**, and the **ultrasonic sensor** to detect motion. I used the web-based applet service, https://ifttt.com, to output the data.

Context & Final Result

I based my project around a **realistic scenario** that helped me imagine a context in which this might be applicable. My context was as follows:

★ Rusty the cat goes in and outside the house on a daily basis. His owners **need a way to know if and when he comes home**. They installed a mat at the entrance of the house that has a **hidden button** underneath. When Rusty enters through the doorway, IFTTT will be triggered and **send a notification to their phones** that he has stepped in. Because Rusty may not step on the mat with the button when he comes inside, his owners installed a **motion sensor that will detect Rusty's presence** upon entry. The motion sensor will trigger IFTTT to **send a GroupMe message** that he is inside to all members of the household that are in the chat. These devices make Rusty's **owners feel secure** in knowing that their cat is safely inside.

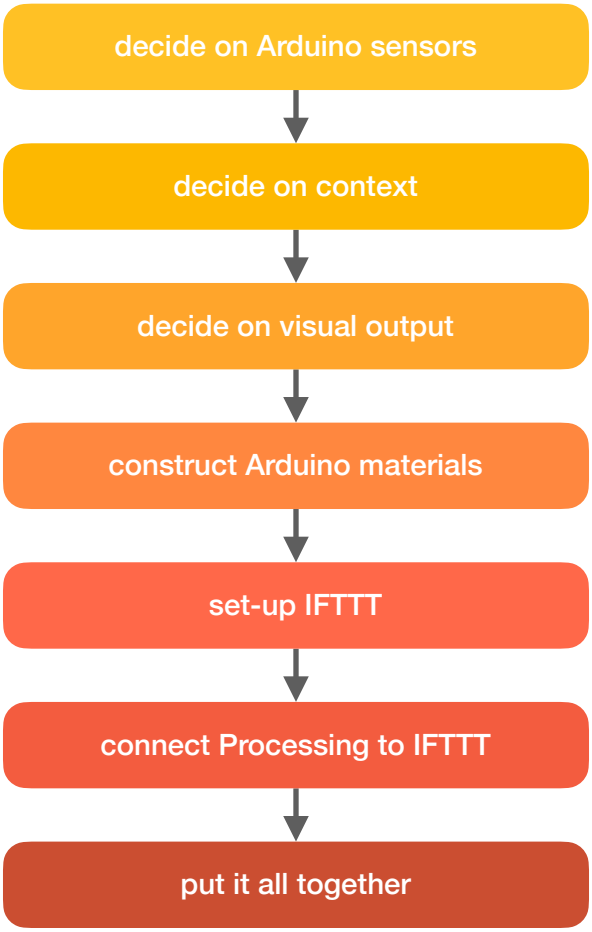
View a video of the final results with this link:
https://drive.google.com/file/d/1c9Z38YurhHMrtI7v9Geb5IQerTqWouBH/view?usp=sharing

THE PROCESS

Overview

In beginning this exercise, I knew I wanted to use the Arduino sensors so that I could **utilize my existing skills**, while also **learning how they applied to this new realm** of networked devices. I first decided on the sensors I wanted to use and had at my disposal. A lot can be done with a simple button, and the ultrasonic sensor had interested me for quite some time, so I chose to utilize these to **create an interesting and relevant context** in which to design. From there, it was just about constructing everything and **connecting the separate parts**: the Arduino, Processing, and IFTTT.

One of the main challenges of the process was trying to get **Processing to connect to IFTTT**. I watched several tutorials, consulted with students in other majors, and it finally came down to me asking for help from my UX peers. I had to admit that **I had no idea what I was doing**, and I ended up learning a lot through their examples and guidance. I feel confident that I would be able to **successfully complete this exercise again** with different sensors and a different context.



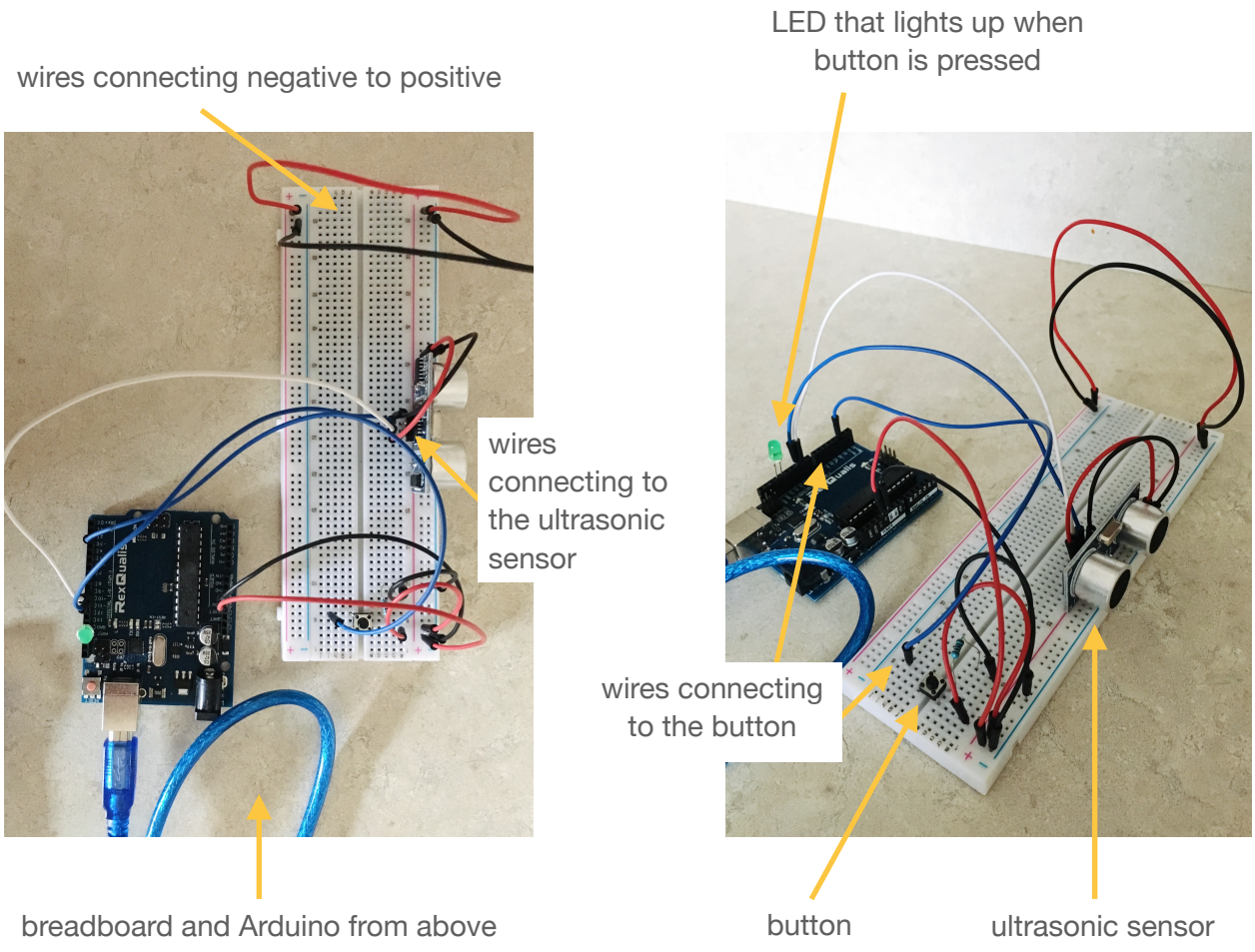
THE PROCESS

Arduino

I first connected the **ultrasonic sensor to detect distance and motion**. When the sensor detected motion **15 centimeters away** or less, it would **send that data to Processing**. It was interesting to see the ways in which the sensor could be utilized. For my purposes, it was easy enough to figure out the necessary wiring and distance.

I next connected the button and LED. The LED **was not used within my context** as an output; I used it to know that my button was working when I pressed it. I was familiar with how to wire these components because of examples I had done in class. It was **exciting to see the button have a purpose**, though.

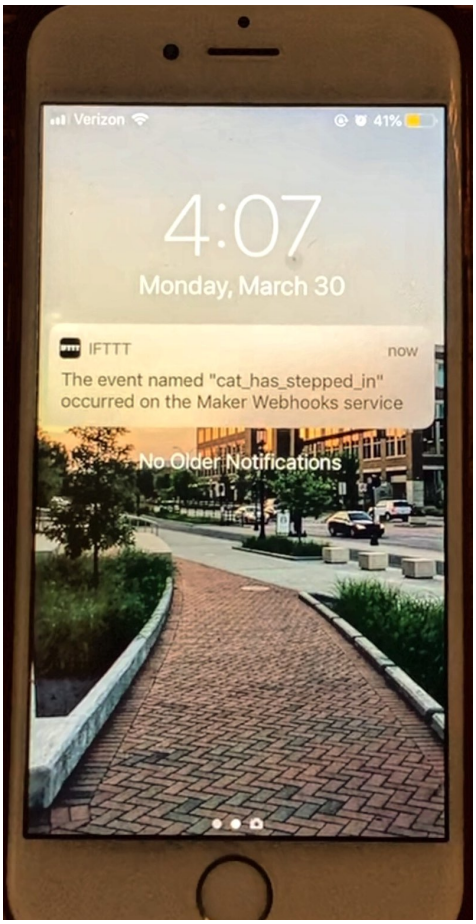
Below is the set-up of my Arduino and the sensors on the breadboard:



THE PROCESS

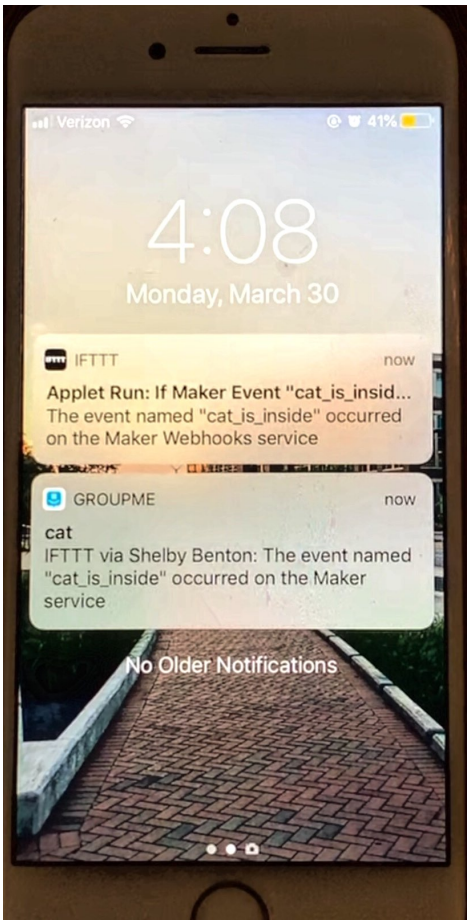
Output Visuals

The output visuals **varied depending on which sensor detects** Rusty's presence first. Both visuals showed up as **notifications** on the owner's phone; one from IFTTT and the other from GroupMe. In an ideal situation, it **would only send one notification** to the phone so that the user does not become overwhelmed with the amount of notifications. Because I was limited in my simulation, it oftentimes sent both notifications at the same time. Pictured below are the notifications the user would get:



IFTTT notification

This notification shows up when Rusty the cat enters through the doorway and steps on the **button**



GroupMe message notification

This notification shows up when Rusty the cat enters through the doorway and the **ultrasonic sensor** detects his motion

THE CODE

Arduino

```
// ULTRASONIC
// defines pins numbers
const int trigPin = 9;
const int echoPin = 10;

// defines variables
long duration;
int distance;
```

```
// BUTTON
// constants won't change. they're used here to set pin numbers:
const int buttonPin = 2;     // the number of the pushbutton pin
const int ledPin =  13;      // the number of the LED pin

// variables will change:
int buttonState = 0;         // variable for reading the pushbutton status
```

```
void setup() {
```

```
// ULTRASONIC
pinMode(trigPin, OUTPUT);   // sets the trigPin as an output
pinMode(echoPin, INPUT);    // sets the echoPin as an input
Serial.begin(9600);         // starts the serial communication
pinMode(buttonPin, INPUT);

// BUTTON
pinMode(ledPin, OUTPUT);    // initialize the LED pin as an output:
pinMode(buttonPin, INPUT);  // initialize the pushbutton pin as an input:
}
```

```
void loop() {
```

```
// ULTRASONIC
// clears the trigPin
digitalWrite(trigPin, LOW);
delayMicroseconds(2);

// sets the trigPin on HIGH state for 10 microseconds
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);

// reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);

// calculating the distance
  distance= duration*0.034/2;

// prints the distance on the serial monitor
Serial.print("Distance: ");
Serial.println(distance);

// read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);
  if (buttonState == HIGH) {
    Serial.println("button_pressed");
  }
```

```
// BUTTON
  // if the pushbutton is pressed, the buttonState is HIGH:
  if (buttonState == HIGH) {      // turn LED on:
  digitalWrite(ledPin, HIGH);
  } else {                        // turn LED off:
  digitalWrite(ledPin, LOW);
}
```

code for the ultrasonic sensor

code for the button

THE CODE

Processing

```
import processing.serial.*; // getting data from the Arduino serial monitor
Serial myPort;
String serialValue;

void setup()
{
  myPort = new Serial(this, "/dev/cu.usbmodem14401", 9600); // 14401 is top left port on MacBook, make sure to always plug Arduino in here
  // this is how Processing knows where to retrieve the data from
}

void draw()
{
  if (myPort.available() > 0){
    {
      serialValue = myPort.readString();
      if (serialValue.contains("15")){ //if the ultrasonic sensor reads 15 or less, then a GroupMe message is sent
        link("https://maker.ifttt.com/trigger/cat_is_inside/with/key/mDtSx42be6Ns3u6FOjLDSmYChoxD09qQ0RjKeppfdkN"); // link from IFTTT
        delay(10000); // delay the output for 10 seconds
      }
      if (serialValue.contains("button_pressed")); //if the button is pressed, then an IFTTT notification is sent
        link("https://maker.ifttt.com/trigger/cat_has_stepped_in/with/key/mDtSx42be6Ns3u6FOjLDSmYChoxD09qQ0RjKeppfdkN"); // link from IFTTT
        delay(10000); // delay the output for 10 secons
    }
  }
}
```

REFLECTION

After using the Arduino and Processing on several occasions before completing this exercise, I was **excited to be able to expand my skillset** and learn new ways that each could be used for prototyping. I had zero knowledge about networked devices and the 'internet of things,' so I **needed to look into examples** of smart home devices and other connected physical devices. After finding examples like the obvious Google Home and Nest, I found more obscure examples like an alarm clock that is connected to a coffee pot, unique garage door openers, and more. These concepts allowed me to think out of the box and at more specific scenarios. My roommate introduced me to a Twitter account, @PepitoTheCat. A Tweet is sent with Pepito's picture when he leaves or enters the house. This gave me the idea to create my kitty context.

As I have mentioned previously, the hardest part of this exercise was connecting Processing to IFTTT. After I figured out how to set up my services on IFTTT, I was **unfamiliar with what to code in Processing** besides the URL given to me in IFTTT. With **help from my peers**, it became clear that I would not have been able to figure that out on my own. I'm thankful that they were willing to help and explain the code.

I had **fun connecting my sensors to the Arduino**. Messing with the wires and checking my work against examples was satisfying, and I was excited to be able to see immediate progress. I hope that I will **feel confident in using the Arduino** to prototype in the future. I have come to appreciate and understand how it can be an extremely **helpful, insightful, and invigorating** way of evaluating an idea.