

Bluetooth Defense Axolotl - Project Timeline

William McAlister

2024-04-15

- Came up with the idea of using the axolotl head from our CPS stories assignment.
- Decided to use a pump as the actuator to advance my project for presentation on demo day.

2024-04-16

- Attended the Build studio workshop and wrote code to control the neopixels and a servo motor via the pinout interfaces of the Bluefruit.
- Read Adafruit's tutorial on Bluetooth proximity sensing and decided to use bluetooth as the sensor.

2024-04-17

- Followed Adafruit's tutorial to understand code that detects the proximity of Bluetooth devices.
- Wrote code to list all Bluetooth devices in the environment.
- Learnt to identify specific Bluetooth devices from their advertisements.
- Based on adafruit's tutorial. Wrote code to detect the proximity of a specific Bluetooth device,
- Struggled with stack overflow errors.

2024-04-18

- Tested how the pump works using a laboratory power supply.
- Figured out how to connect hoses to the pump.

2024-04-21

- Bought some plumbing fittings.
- Borrowed a 12v battery from my uncle.

2024-04-22

- Talked with Muhammed Asfour about the best way to control a DC motor.
- Used the laboratory power supply to test how a relay switch functioned.
- Progressed the code. Struggled with stack overflow errors (see image below).

2024-04-23

- Read Adafruit's documentation on the pinout interfaces.
- Wired the pump and the Bluefruit together via the relay switch.
- Wrote simple code so that the Bluefruit would periodically switch the pump on and off (see test_relay_switch.py)

2024-04-24

- Combined the sensor and actuators together so that the pump ran when the axolotl detected the stork in close proximity.
- Tested using a battery to power my pump.
- Completed a viable set of code for both the axolotl (see axolotl_combined.py) and the stork.

```

File Edit View Settings Shell Run Tools Help
stork.py CIRCUITPY/code.py CircuitPython 8.x/code.py
30 # Advertise your signal so that the other creature knows you are nearby
31 print('advertising')
32 ble.start_advertising(advertisement)
33 time.sleep(2)
34 ble.stop_advertising()
35
36 #check if there are other creatures nearby
37 print("Scanning for devices")
38 cpb.pixels.fill(0x111111)
39 advertisements = ble.start_scan(minimum_rssi=-200, timeout=0.3)
40 ads = [(e.complete_name, e.rssi) for e in advertisements]
41 ble.stop_scan()
42 names = [x for x, y in ads]
43 rssi = [y for x, y in ads]
44 print(names)
45 #print(stork in names)
46 if stork in names:
47     stork_rssi = rssi[names.index(stork)]
48     print('stork rssi: ', stork_rssi)
49     # This discrete strength code was taken from an Adafruit tutorial
50     # 2019 John Edgar Park for Adafruit Industries MIT licence
51     discrete_strength = min((100 + stork_rssi) // 5, 10)
52     for i in range(0, discrete_strength):
53         cpb.pixels[i] = 0x110000
54     cpb.pixels.show()
55     #if stork_rssi < 70:
56     # spray water
57     #elif stork_rssi < 80:
58     # warn the user
59     del(stork_rssi)
60     del(discrete_strength)
61 print('YEWwW!')
62 del(advertisements)
63 del(ads)
64 del(names)
65 del(rssi)
66 #del(stork_rssi)
67 #del(discrete_strength)
68 print(locals())
69 gc.collect()
70 #time.sleep(3)

, None, None, None]
YEWwW!
{'time': <module 'time'>, '__file__': 'code.py', 'stork': 'CIRCUITPY1013', '
', '__name__': '__main__', 'cpb': <Bluefruit object at 0x20035c60>, 'advert
ff\x22\x08\x06\x00\x00\x00\x00\x11\x00'), 'ble': <BLERadio object at 0x20017
'CIRCUITPY1013', 'BLERadio': <class 'BLERadio'>}
advertising
Scanning for devices
[None, None, None, None, None, None, None, None, None, None, None, None, No
, None, None, None, None, None, None, None, None, None, None, None, No
0A7AY|_7&v#VE_P1AqWGP', None, None, None, None, None, None, None, None, No
None, None, None, None, None, None, None, None, None, None, None, None, No
, None, None, None, None, None]
YEWwW!
{'time': <module 'time'>, '__file__': 'code.py', 'stork': 'CIRCUITPY1013', '
', '__name__': '__main__', 'cpb': <Bluefruit object at 0x20035c60>, 'advert
ff\x22\x08\x06\x00\x00\x00\x00\x11\x00'), 'ble': <BLERadio object at 0x20017
'CIRCUITPY1013', 'BLERadio': <class 'BLERadio'>}
advertising
Scanning for devices
[None, None, None, None, None, None, None, None, None, None, None, None, No
, None, None, None, None, None, None, None, None, None, None, None, No
e, None, None, None, None, None, None, None, None, None, None, None, No
YEWwW!
{'time': <module 'time'>, '__file__': 'code.py', 'stork': 'CIRCUITPY1013', '
', '__name__': '__main__', 'cpb': <Bluefruit object at 0x20035c60>, 'advert
ff\x22\x08\x06\x00\x00\x00\x00\x11\x00'), 'ble': <BLERadio object at 0x20017
'CIRCUITPY1013', 'BLERadio': <class 'BLERadio'>}
advertising
Traceback (most recent call last):
  File "code.py", line 40, in <module>
    File "code.py", line 40, in <listcomp>
    File "adafruit_ble/_init_.py", line 274, in start_scan
MemoryError: memory allocation failed, allocating 58313 bytes

Code done running.

Press any key to enter the REPL. Use CTRL-D to reload.

```

Figure 1: Too much Bluetooth, not enough RAM

- Major struggles with stack overflow errors in the lab environment presumably because of the number of Bluetooth devices around.

To-Do 2024-04-25

- ☒ Draw the Axolotl eye
- ☒ Solder the axolotl eye
- ☒ Fit the eye to the head
- ☒ Attach the battery into the head
- ☒ Attach the relay to the side of the head
- ☒ Fix the stork code to broadcast reliably
- ☒ Code the warning into the axolotl
- ☒ Code the eyes for the stork
- ☒ Tidy up the wiring
- ☒ Do a full assembled test outdoors
- ☒ Get a video of me playing with it
- ☐ Eliminate stack overflow issues in the code

Video

Why?

- ☐ I've made two creatures that interact via Bluetooth.
- ☐ I've learnt to use the Bluetooth sensor.
- ☐ I've also learnt how to control a 12v circuit with a 4v circuit and a to control a flow of water with a flow of electricity.
- ☐ I chose to make an axolotl and a stork because of an in-joke that started with our CPS stories presentation.
- ☐ I didn't have much time this fortnight so I decided to re-purpose an existing creature rather than starting from scratch.
- ☐ I chose to use these sensors and actuators because I am aiming to employ the same components in my maker project.
- ☐ The choice of Bluetooth caused some headaches. I think it was pushing the limits of the Bluefruit hardware.

What?

- ☐ Explain how the stork works.
- ☐ Explain how the Axolotl works.

- ☐ Explain how to setup the axolotl to see it working.
- ☐ Explain what can go wrong.
- ☐ Explain the challenges: Stack overflow from BLE, electrical connections are a bit flimsy, the pump can stall with air bubbles.

How?

- ☐ Talk about the sensor, actuator and physical components as three separate streams of development that were brought together at the end.
- ☐ Refer the user to the supplementary docs to see a couple of iterations of the code.
- ☐ To make the Bluetooth sensor, I followed a tutorial from Adafruit.
- ☐ It was a really satisfying experience. The code worked immediately when loaded onto my two Bluefruits and I easily understood how the code worked and was able to make changes. I learnt a lot about bluetooth from the tutorial.
- ☐ I think Adafruit does a phenomenal job with their tutorials and I really like that they release their code open-source.

Acknowledgements

- ☐ I basically need to thank everyone this fortnight.
- ☐ Biggest thanks to Tianee for making the axolotl costume in the first place and for being my videographer.

Supplementary Docs

- ☐ Include this to-do list
- ☐ Augment this to-do list to show the timeline of what I did on each day and a couple of reflections.
- ☐ Include some of the intermediate python code linked from the text.
- ☐ Include a screenshot of some errors.