ABIS

Algae **B**loom **I**maging **S**atellite

Nevin, Ranvitha, Steven, Thomas

Issue

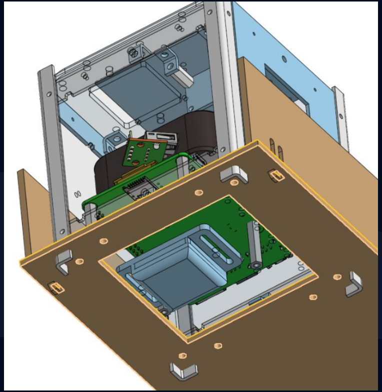
- Harmful Algae Blooms (HABs) impacting communities off the coast of New England
- Makes shellfish non-consumable due to Paralytic Shellfish Poisoning
- Impacts fisheries that depend on harvesting and selling shellfish
- Climate change has altered how HABs impact coastal ecosystems

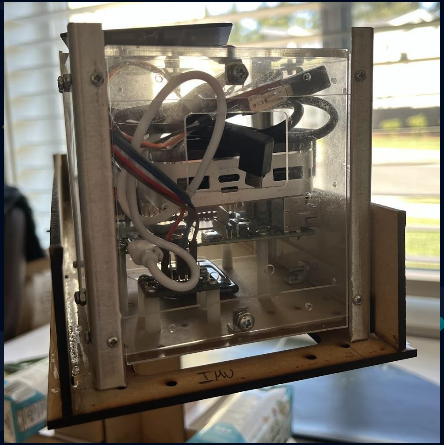
Mission

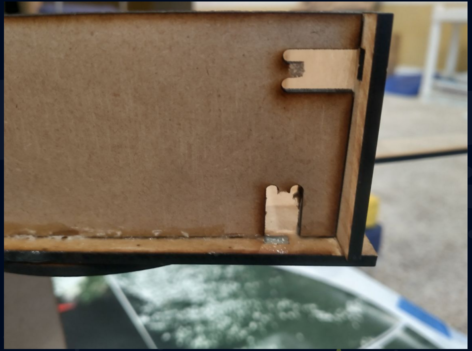
- Create a cubesat that is able to monitor the east coast region for HABs
- Allows researchers to use data to find and correlate long term trends to better forecast HABs
- Consequently allows coastal fisheries to be better prepared for and able to mitigate the impacts of HABs

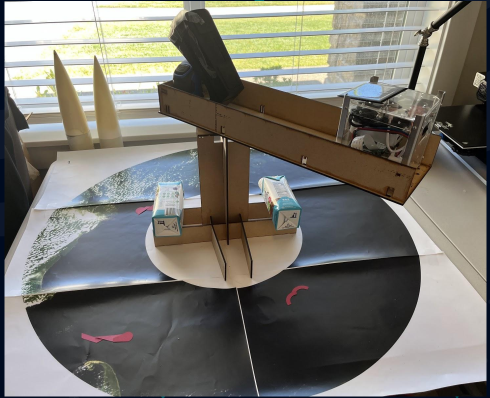
CubeSat Design - Structures

- Raspberry Pi 4 Flight Computer
- Acrylic body panels
- Battery mounted on Velcro
- Wooden Orbiter Parts
- Miscellaneous counterweight
- *Challenges*
  - *Orbiter platform bending*
  - *Orbiter part feet breaking*
  - *Metal pieces contacting exposed circuitry*



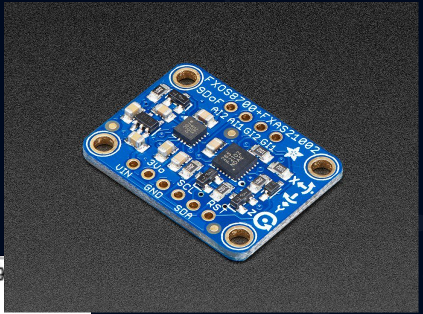


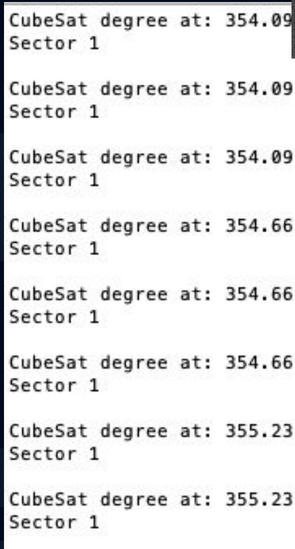




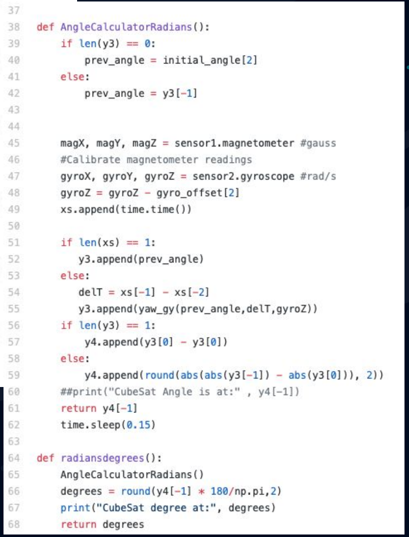
CubeSat Design - ADCS

- Accelerometer and magnetometer set initial position
- Gyroscope determined orbiter angle
- Returned orbiter angle and flight path sector
- Image capture every 40 degrees
- Writes angle and sector onto image
- *Challenges*
  - *Gyroscope measurements drift by ~12 degrees per rotation*
  - *Issues with readings going backwards*





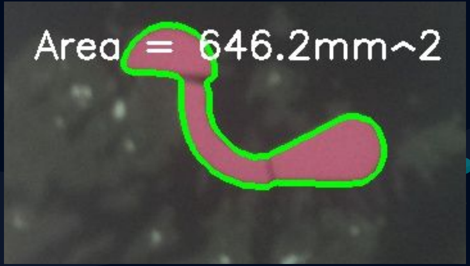
```
CubeSat degree at: 354.09
Sector 1

CubeSat degree at: 354.09
Sector 1

CubeSat degree at: 354.09
Sector 1

CubeSat degree at: 354.66
Sector 1

CubeSat degree at: 354.66
Sector 1

CubeSat degree at: 354.66
Sector 1

CubeSat degree at: 355.23
Sector 1

CubeSat degree at: 355.23
Sector 1
```



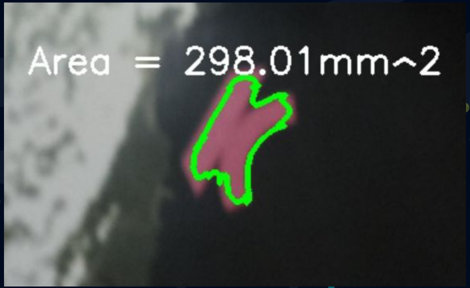
```

def AngleCalculatorRadians():
    if len(y3) == 0:
        prev_angle = initial_angle[2]
    else:
        prev_angle = y3[-1]


    magX, magY, magZ = sensor1.magnetometer #gauss
    #Calibrate magnetometer readings
    gyroX, gyroY, gyroZ = sensor2.gyroscope #rad/s
    gyroZ = gyroZ - gyro_offset[2]
    xs.append(time.time())

    if len(xs) == 1:
       y3.append(prev_angle)
    else:
       delT = xs[-1] - xs[-2]
       y3.append(yaw_gy(prev_angle,delT,gyroZ))
    if len(y3) == 1:
        y4.append(y3[0] - y3[0])
    else:
        y4.append(round(abs(abs(y3[-1]) - abs(y3[0])), 2))
    ##print("CubeSat Angle is at:" , y4[-1])
    return y4[-1]
    time.sleep(0.15)

def radiansdegrees():
    AngleCalculatorRadians()
    degrees = round(y4[-1] * 180/np.pi,2)
    print("CubeSat degree at:", degrees)
    return degrees
```

CubeSat Design - Image Processing

- Facilitated through OpenCV Python Library
- Image denoising to combat low camera quality
- Color masking to isolate algae blooms
- Edge intersection detection to avoid partial bloom processing
- Utilize proportional pixel count to calculate the area of the algae bloom
- Writing the data from ADCS and calculated area onto the photo
- *Challenges:*
  - *Finding the optimal color range*
  - *Having non-algae objects detected as algae*
  - *Partial algae detection*



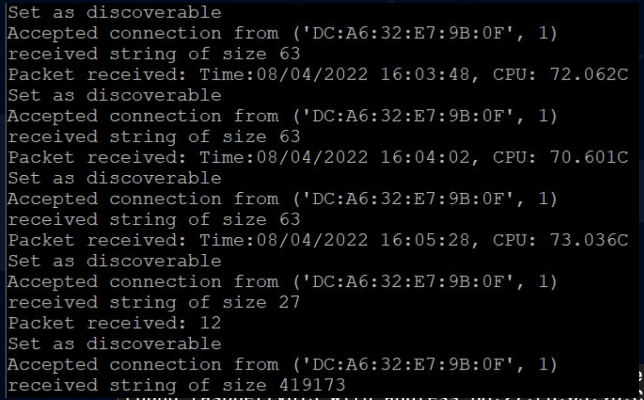
Complete Detection



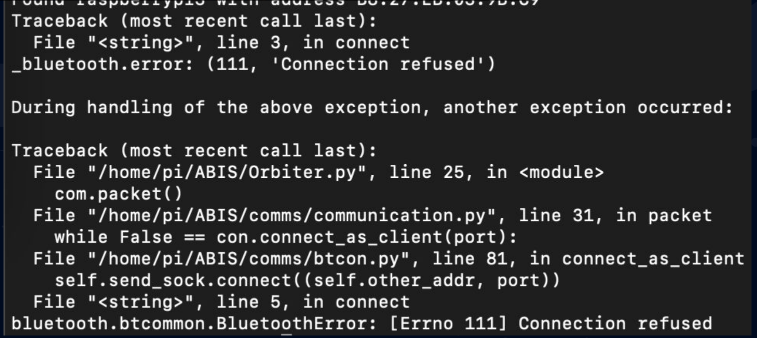
Partial Detection

CubeSat Design - Communications

- Two way connection between pis to facilitate data receive confirmation
- Timed packets to monitor cubesat health
- Missing-image checking and resolution
- Relative location and orientation between ground pi and cubesat
- Challenges
  - Occasionally searched for a long time
  - Froze midway through transmit
  - Incorrect send and receive timing resulting in data loss



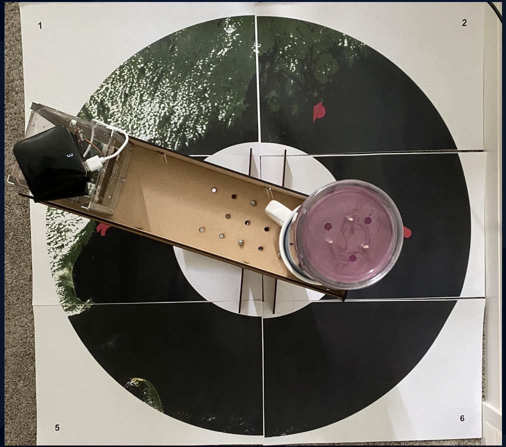
```
Set as discoverable
Accepted connection from ('DC:A6:32:E7:9B:0F', 1)
received string of size 63
Packet received: Time:08/04/2022 16:03:48, CPU: 72.062C
Set as discoverable
Accepted connection from ('DC:A6:32:E7:9B:0F', 1)
received string of size 63
Packet received: Time:08/04/2022 16:04:02, CPU: 70.601C
Set as discoverable
Accepted connection from ('DC:A6:32:E7:9B:0F', 1)
received string of size 63
Packet received: Time:08/04/2022 16:05:28, CPU: 73.036C
Set as discoverable
Accepted connection from ('DC:A6:32:E7:9B:0F', 1)
received string of size 27
Packet received: 12
Set as discoverable
Accepted connection from ('DC:A6:32:E7:9B:0F', 1)
received string of size 419173
```

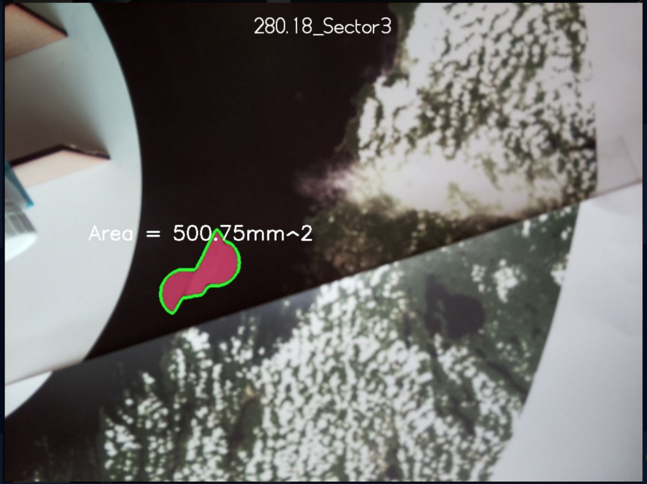


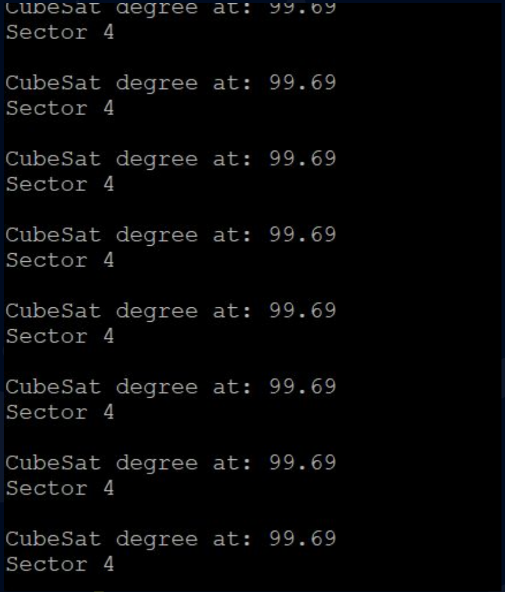
```
errypi3
Found raspberrypi3 with address B8:27:EB:03:7B:C9
Traceback (most recent call last):
  File "<string>", line 3, in connect
_bluetooth.error: (111, 'Connection refused')

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "/home/pi/ABIS/Orbiter.py", line 25, in <module>
    com.packet()
  File "/home/pi/ABIS/comms/communication.py", line 31, in packet
    while False == con.connect_as_client(port):
  File "/home/pi/ABIS/comms/btcon.py", line 81, in connect_as_client
    self.send_sock.connect((self.other_addr, port))
  File "<string>", line 5, in connect
bluetooth.btcommon.BluetoothError: [Errno 111] Connection refused
```

FLIGHT DAY VIDEO

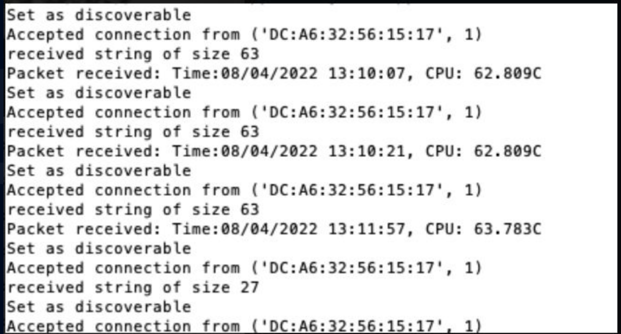
Flight Day Results







```
CubeSat degree at: 99.69
Sector 4

CubeSat degree at: 99.69
Sector 4

CubeSat degree at: 99.69
Sector 4

CubeSat degree at: 99.69
Sector 4

CubeSat degree at: 99.69
Sector 4

CubeSat degree at: 99.69
Sector 4

CubeSat degree at: 99.69
Sector 4

CubeSat degree at: 99.69
Sector 4
```



```
Set as discoverable
Accepted connection from ('DC:A6:32:56:15:17', 1)
received string of size 63
Packet received: Time:08/04/2022 13:10:07, CPU: 62.809C
Set as discoverable
Accepted connection from ('DC:A6:32:56:15:17', 1)
received string of size 63
Packet received: Time:08/04/2022 13:10:21, CPU: 62.809C
Set as discoverable
Accepted connection from ('DC:A6:32:56:15:17', 1)
received string of size 63
Packet received: Time:08/04/2022 13:11:57, CPU: 63.783C
Set as discoverable
Accepted connection from ('DC:A6:32:56:15:17', 1)
received string of size 27
Set as discoverable
Accepted connection from ('DC:A6:32:56:15:17', 1)
```

General Challenges

- Get the software to work correctly on everyone's Pi's
- Working with softwares and programs that many of us were not familiar with
- Putting together our orbiters and CubeSats without breaking anything
- Debugging errors that we have never seen before and don't have many online resources explaining what the error is and how to fix

Lessons Learned

- Implementing code freezes is critical
- Test code in its final runtime environment
- Sometimes reflashing your pi is the only way to solve an error
- Always have a way to revert to a previous version
- Minimize the number of bluetooth transmissions
    - Don’t strain your weakest link
- Minimize last minute edits

Real World Application

- Ground station image processing
- Color detection of other environmental changes such as deforestation, water quality, and wildlife
- Data gathering for remote locations
- More equitable space access

Acknowledgements

Emily McCarthy

Maddie Schroeder

Aidan Carrier