# Bluetooth Ingestible Capsule

Group: sddec23-19
Members: Chase Thompson, Cutler Thayer, Jon Thomas, Robert Zukowski
Website: https://sddec23-19.sd.ece.iastate.edu/

# Problem Statement

- We want to diagnose Gastrointestinal (GI) Ailments in affordable, non-invasive ways using ingestible capsules
- Our solutions
  - Wirelessly capture and transmit GI images
  - Automatically detect points of interest within GI images
  - Create a GUI to aid with viewing points of interest
- Our hopes
  - Reduce labor of gastroenterologists diagnosing colon ailments
  - Reduce cost of ingestible imaging capsules

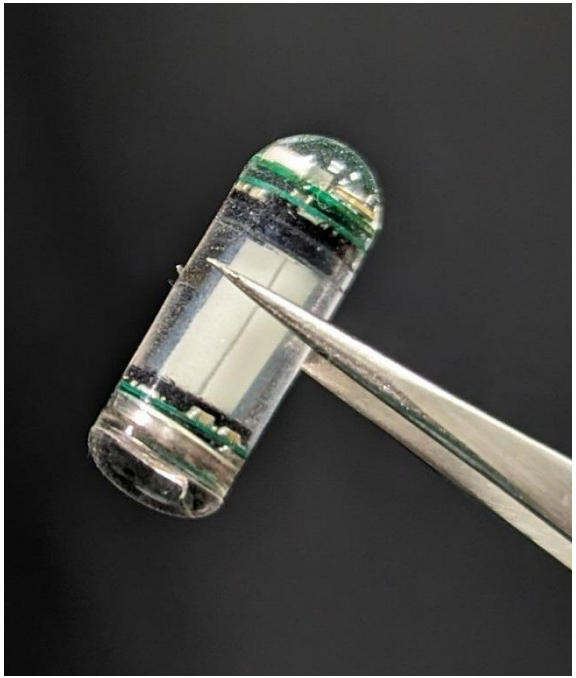
# Design Requirements

**Functional Requirements:**

- Analyze images for signs of abnormalities
- Display the image data and analysis findings
- Receive data from an image sensor,
- Send/Receive images via Bluetooth/USB

**Non-Functional Requirements:**

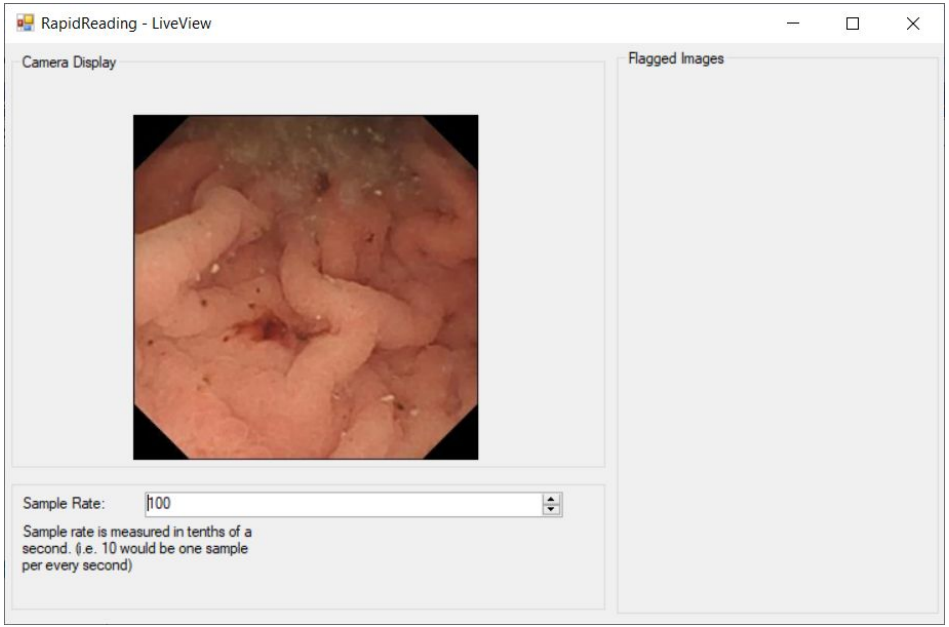
- Electronics must fit in a standard 000 size capsule
- Analysis of an image frame can be completed before another frame is received



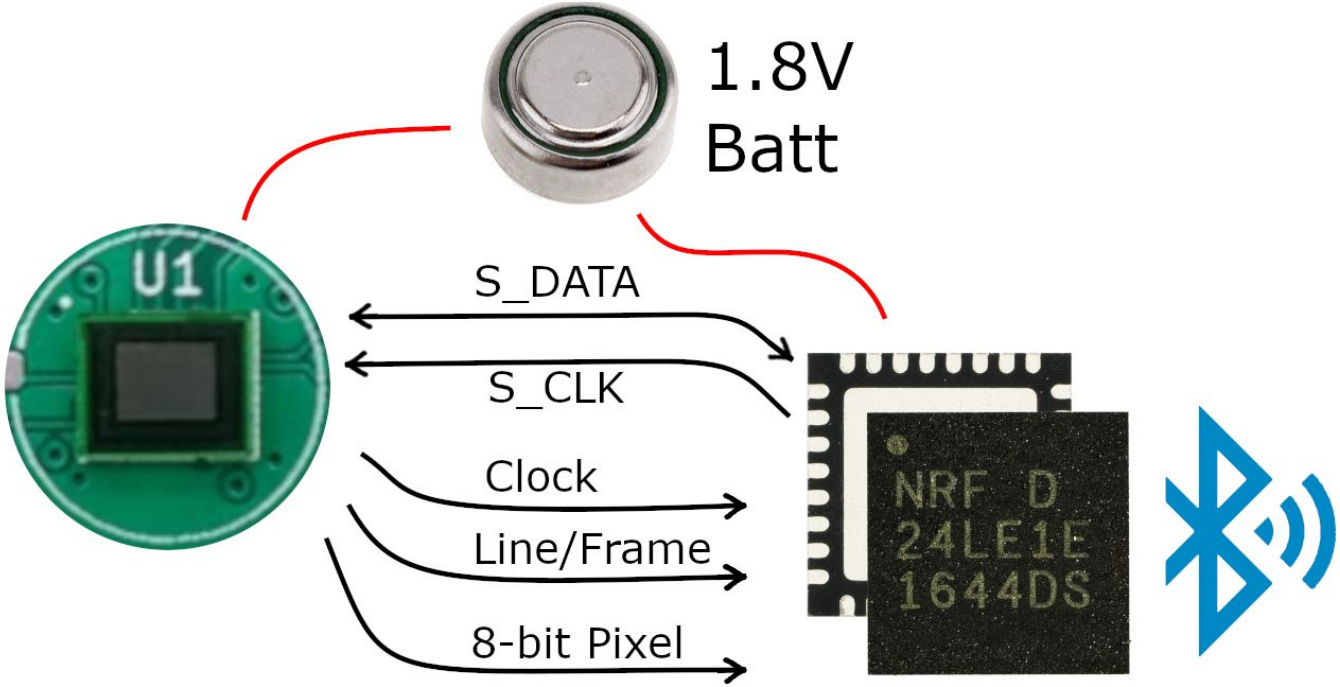
*Credit: National University of Singapore*

# Accomplishments

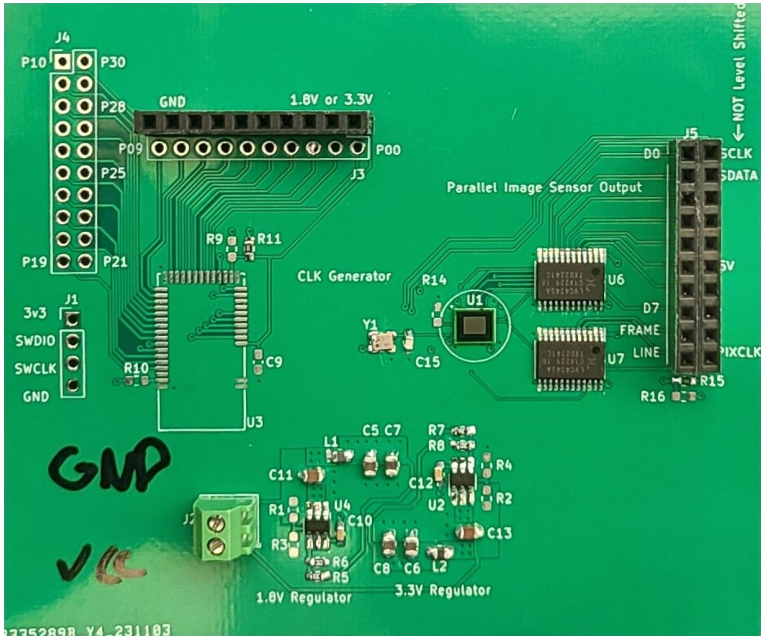
- Successfully detects abnormalities in GI images
- Acquired an image dataset of GI irregularities
- Stores flagged images in a labeled folder and shows thumbnails in GUI
- Successfully send image data over Bluetooth and USB
- Researched an image sensor that meets size specifications

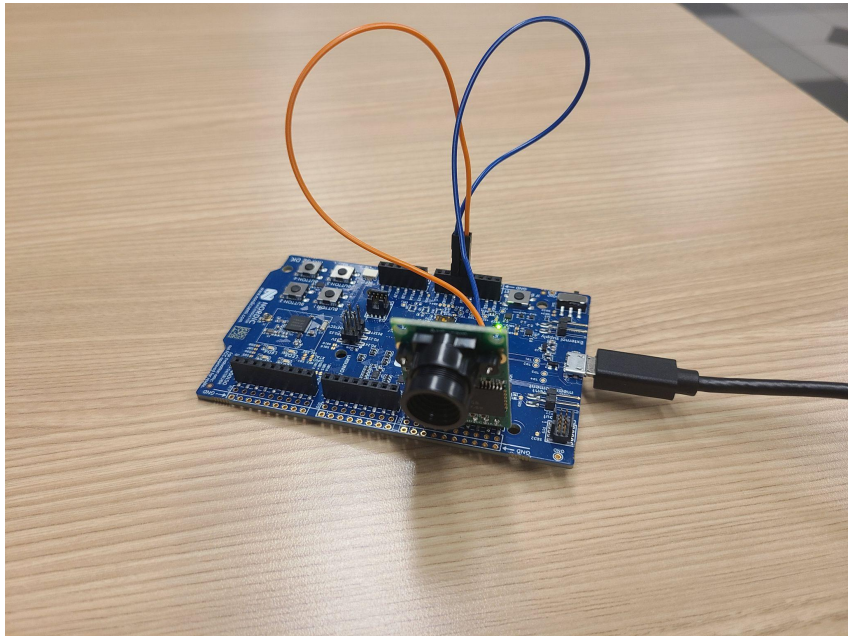


# Capsule Breakdown



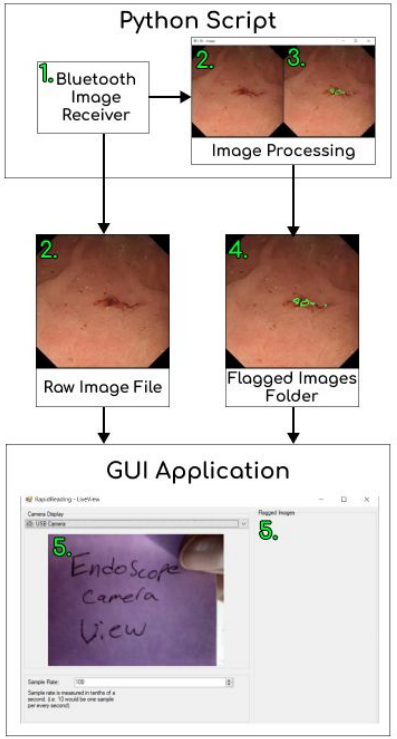
# Prototypes


J4
P10
P30
P28
GND
1.8V or 3.3V
P09
P00
J3
P25
P19
P21
R9
R11
CLK Generator
R14
U1
U6
Y1
C15
C9
R10
3v3
J1
SWDIO
SWCLK
GND
U3
Parallel Image Sensor Output
J5
D0
SCLK
SDATA
NOT Level Shifted
5V
D7
FRAME
LINE
PIXCLK
U7
R15
R16
GND
L1
C5 C7
R7
R8
R4
C11
C12
U2
R2
R1
U4
C10
C13
R3
R6
R5
C8
C6
L2
1.8V Regulator
3.3V Regulator



# Software Breakdown

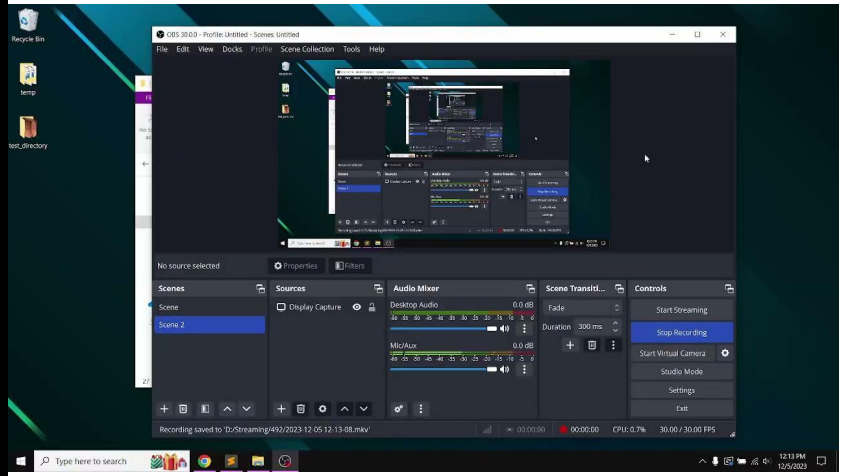
1. Python script receives image data from Bluetooth
2. Image data is sent to image analysis script while also being saved under “liveview.jpg”
3. Image analysis script detects if ailments are present in the image
4. If ailments are detected, a contoured image is generated and saved under the flagged_images directory
5. GUI loads “liveview.jpg” and displays it to the user while also checking for flagged images



# Demo

LiveView Demonstration

Bluetooth Demonstration



# Testing

Image Analysis Testing

- Unit Tests
- 50 tests, 86% coverage
  - 25/25 tests passed for saving images
  - 17/25 tests passed for successfully analyzing images

```
=================================== short test summary info ====================================
FAILED test_image.py::test_img_analysis[1.jpg] - AssertionError: assert 'False' == 'True'
FAILED test_image.py::test_img_analysis[18.jpg] - AssertionError: assert 'False' == 'True'
FAILED test_image.py::test_img_analysis[2.jpg] - AssertionError: assert 'False' == 'True'
FAILED test_image.py::test_img_analysis[3.jpg] - AssertionError: assert 'False' == 'True'
FAILED test_image.py::test_img_analysis[4.jpg] - AssertionError: assert 'False' == 'True'
FAILED test_image.py::test_img_analysis[5.jpg] - AssertionError: assert 'False' == 'True'
FAILED test_image.py::test_img_analysis[6.jpg] - AssertionError: assert 'False' == 'True'
FAILED test_image.py::test_img_analysis[7.jpg] - AssertionError: assert 'False' == 'True'
================================= 8 failed, 42 passed in 0.50s =================================
```

Coverage report: 86%

*coverage.py v7.3.2, created at 2023-12-03 10:46 -0600*

| Module ↑ | statements | missing | excluded | coverage |
|---|---|---|---|---|
| detect.py | 52 | 10 | 0 | 81% |
| test_image.py | 18 | 0 | 0 | 100% |
| **Total** | **70** | **10** | **0** | **86%** |

*coverage.py v7.3.2, created at 2023-12-03 10:46 -0600*

# Software Challenges & Solutions

| Challenges | Solutions |
| --- | --- |
| • Difficulty integrating C# WinForm with a Python script<br>  ○ GUI runs in C# WinForm<br>  ○ GUI needs to start Image Analysis Script<br>  ○ Image Analysis runs on Python | • Using Windows processes to start the script from the User Interface Application<br>• Adding Argparse for Python to pass different arguments from the UI |
| • False readings on images<br>  ○ Folds within the colon can look similar to some ailments like blood<br>  ○ Some ailments are too subtle to notice without causing false positives | • Focusing on the center of the images<br>• Calibrating the image threshold |
| • Sending images over Bluetooth<br>  ○ Connection hangs<br>  ○ New images are sent before old ones are fully received | • Compressing image data more<br>• Only send images after so much time has elapsed |

# Hardware Challenges & Solutions

| Challenges | Solutions |
| --- | --- |
| ● Image Sensor Integration<br>○ 96 MHz pixel clock output<br>○ No existing embedded software libraries to handle image sensor | ● Direct connection to high-speed MCU |
| ● Bluetooth Low Energy<br>○ Bluetooth classic not supported by modern nRF devboards<br>○ Reliance on Nordic support for troubleshooting | ● Use BLE |
| ● Capsule integration<br>○ Board could not fit into standard-sized capsules | ● Separate “blocks” wired with flex-PCBs |

# Key Team Contributions

**Chase Thompson**

- Image Analysis using OpenCV
- Developed the User Interface
- Defined test cases for image analysis
- Created test scripts for the image analysis using pytest
- Wrote documentation for building the project

**Cutler Thayer**

- Researched what kinds of ailments are covered in endoscopy
- Acquired training data to test for each kind of ailment

**Robert Zukowski**

- Drafted schematic and layout for image sensor test-board
- Researched project components used
- Began implementation/troubleshooting for image sensing over BLE

**Jon Thomas**

- Researched existing technology
- Facilitated advisor meetings
- Aided in solution design

# Future Work

**Software Plans**

- Set up CI/CD Pipeline
- Improve image analysis accuracy
- Add support for detection of different ailments
- Add integration testing to fully test the LiveView segment
- Implement machine learning for image analysis
- Add support for prerecorded video scanning

**Hardware Plans**

- Integrate capsule
  - Requires wireless capability
- Source microcontroller with Bluetooth capabilities and high internal clock speed
- Integrate a 1.8V battery back into the design

# Conclusion

- Most of the original project requirements have been met
  - Software that analyzes and categorizes images between healthy and unhealthy
  - User interface that allows the client to easily analyze already acquired images as well as a live feed from a camera
- Future Enhancements
  - Support for more ailments
  - Larger set of training data
  - Improvement to the code base

## Chase Thompson

*Software Engineering*

## Robert Zukowski

*Electrical Engineering*

## Cutler Thayer

*Computer Engineering*

## Jon Thomas

*Electrical Engineering*