Lunar Exploration using Augmented Reality (LEAR)

Group 6:

Teo Malendevych	Photonic Engineering
Sammy Lee	Photonic Engineering
David Brown	Computer Engineering
Yongsheng Xu	Electrical Engineering

Sponsored By:

George Jackson Foundation
NASA Florida Space Grant Consortium

Project Overview

Inspiration

Teo

Goals and Objectives

General:

- Display of Extravehicular activity (EVA) tasks
- Communication with Mission Control
- System Tutorial

Navigation and Task Assistance:

- Guidance from point A to point B
- Location of points of interest, lander, partners
- Available notes and directions for tasks
- Ability to capture video and images
- Assistance with high-contrast areas on the moon

Peripheral Devices:

- Wrist-mounted optical heart rate monitor and pulse oximeter
- Outside temperature sensor
- Wireless communication with headset

Teo

Requirements and Specifications

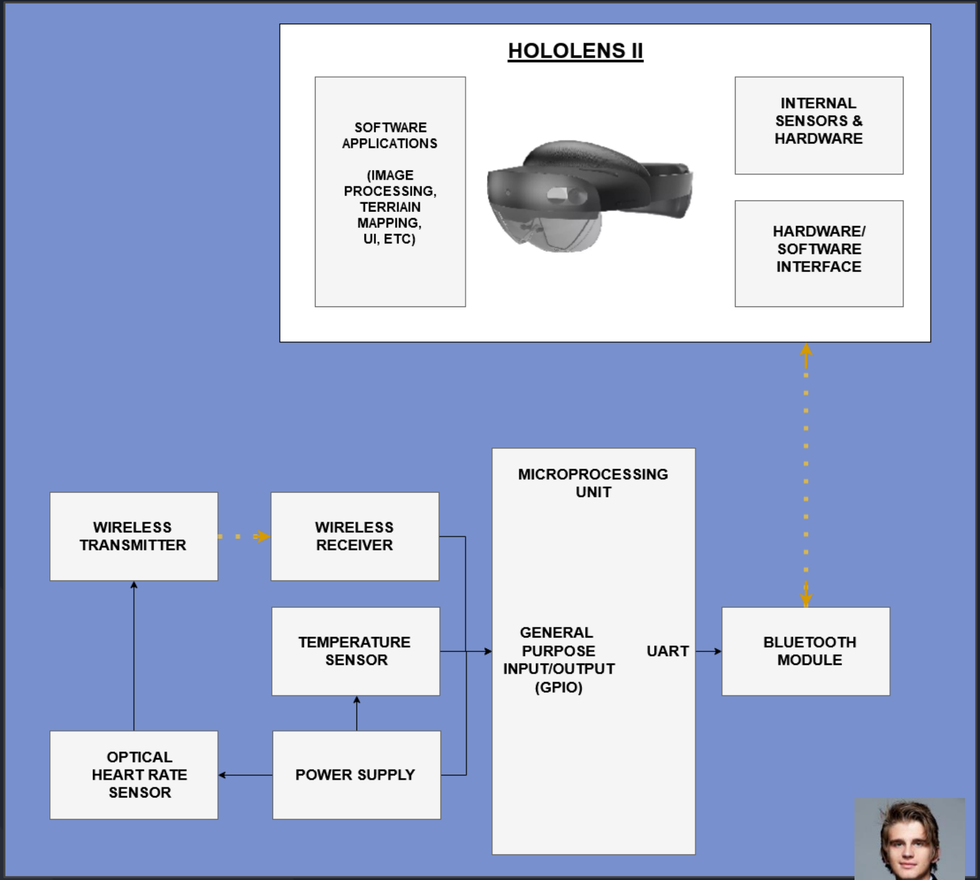
- Guide user in real-time to any marked coordinates of interest within a range of 2 kilometers from starting position.
- Monitor suit vitals (O2, H2O, etc) and alert user if values are abnormal or approach dangerous levels.
- Provide a method for taking images and videos
- Monitor user pulse and alert if value approaches abnormal levels (>100 bpm or <60 bpm
- Image Processing should not be less than 30 frames per second.
- Heart rate monitor PCB: 5 cm x 5 cm
- Measurement of heart rate and SO_2 accurate to 10%
- Wireless transmission of 10 ft
- Temperature measurements accurate to 2%

Teo

External System Design

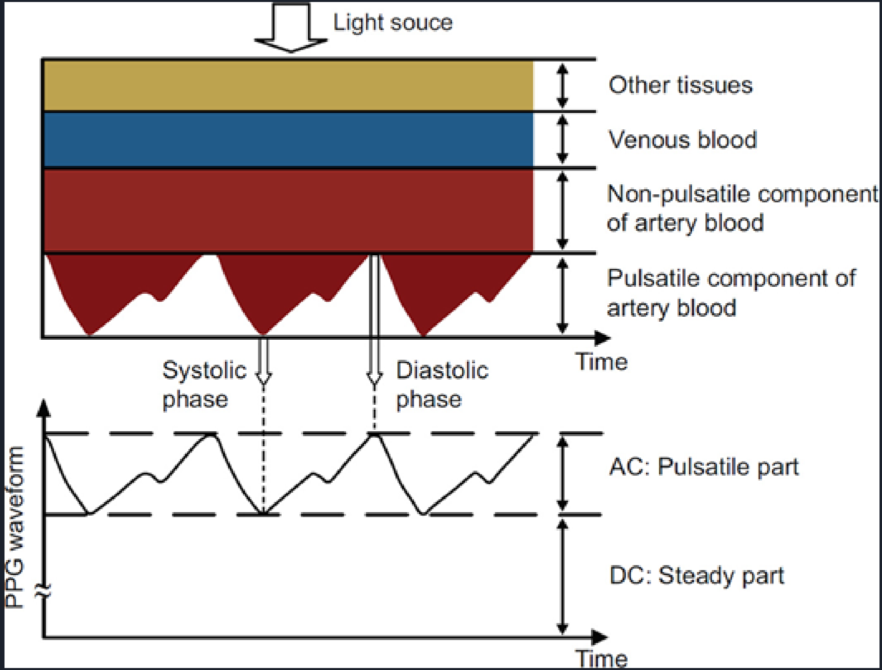
Hardware Design

- External sensors to monitor suit (and astronaut) vitals
  - Temperature
  - Heart Rate
- Microprocessor receives sensor data
- Wirelessly transmitted via bluetooth



Teo

Heart Rate Monitor and Pulse Oximeter Theory

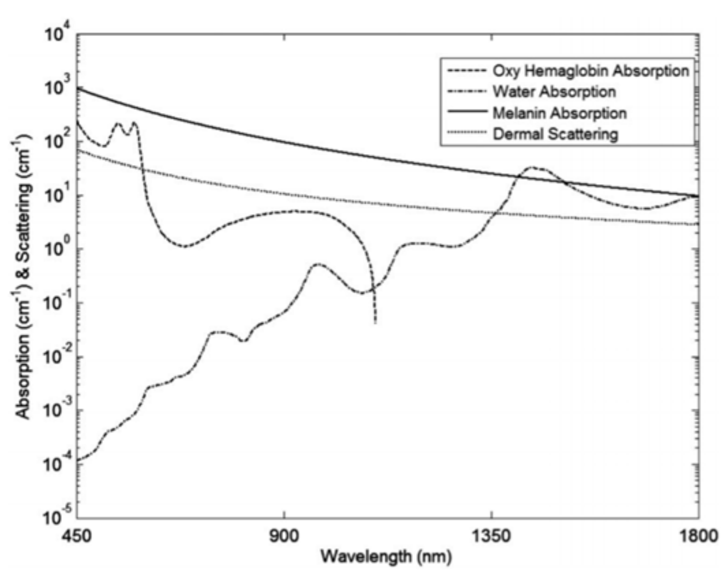


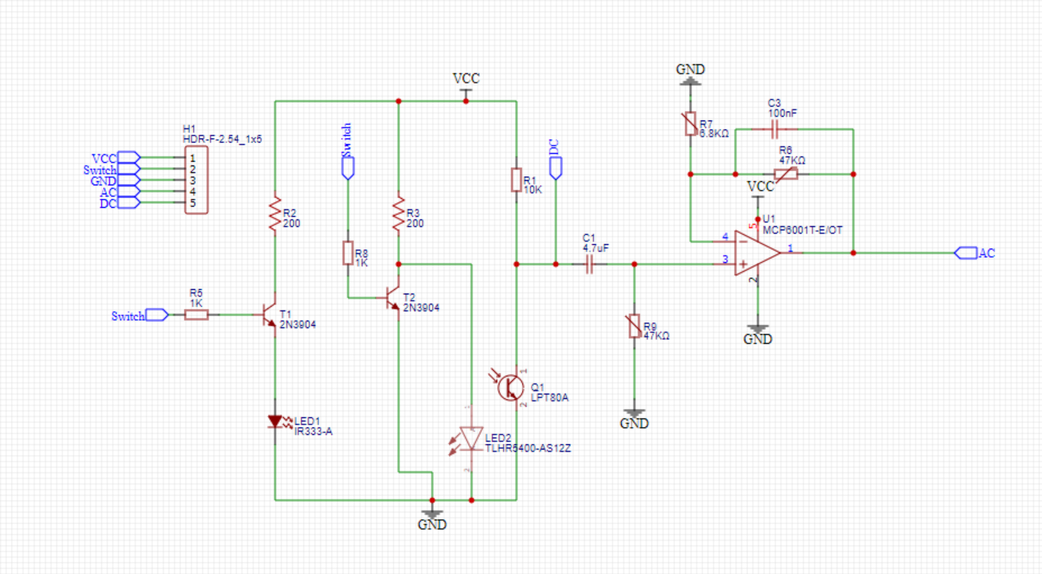
$$R = \frac{\ln \frac{I_{\max}(\lambda_1)}{I_{\min}(\lambda_1)}}{\ln \frac{I_{\max}(\lambda_2)}{I_{\min}(\lambda_2)}}$$

$$\frac{[\mathrm{AC/DC}]_{\lambda_1}}{[\mathrm{AC/DC}]_{\lambda_2}} = \frac{\left[\left(\mathrm{SpO_2} \cdot \varepsilon_{\mathrm{HbO_2}} + (1-\mathrm{SpO_2})\varepsilon_{\mathrm{Hb}}\right) \cdot \Delta[\mathrm{HbT}]_a \cdot L\right]_{\lambda_1}}{\left[\left(\mathrm{SpO_2} \cdot \varepsilon_{\mathrm{HbO_2}} + (1-\mathrm{SpO_2})\varepsilon_{\mathrm{Hb}}\right) \cdot \Delta[\mathrm{HbT}]_a \cdot L\right]_{\lambda_2}}$$

Teo

Heart Rate Monitor and Pulse Oximeter
Part Selection & Design





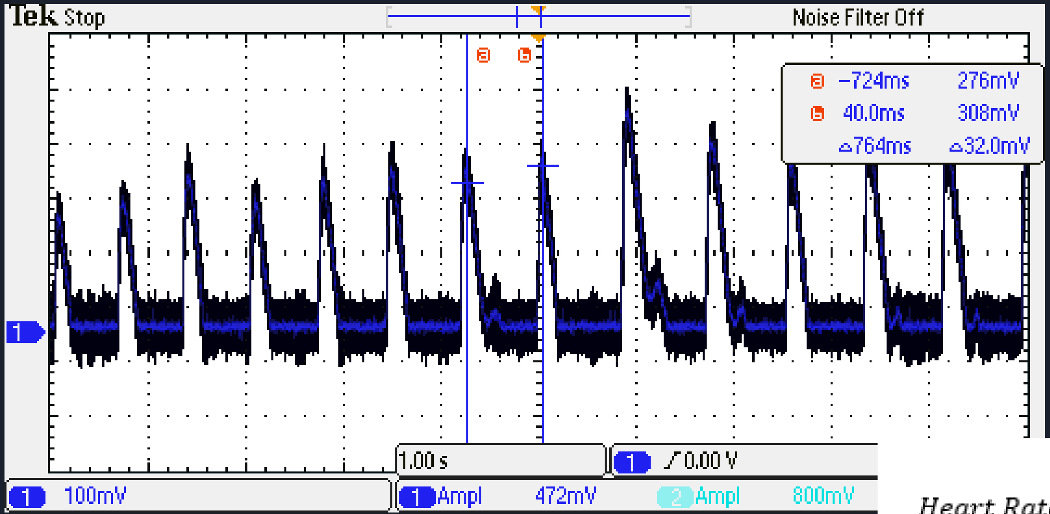
IR LED: Vishay Semiconductors VSMY2943G, Emitter Wavelength: 940 nm, LxW: 5.8 mm by 2.3 mm

Red LED: Cree XLamp XP-E2 LED, Emitter Wavelength: 650-670 nm, LxW: 3.5 mm by 3.5 mm,

Photodiodes: Osram LPT 80A Phototransistor, Spectral Range of Sensitivity: 450 - 1100 nm, LxW: 5 mm by 6 mm

Teo

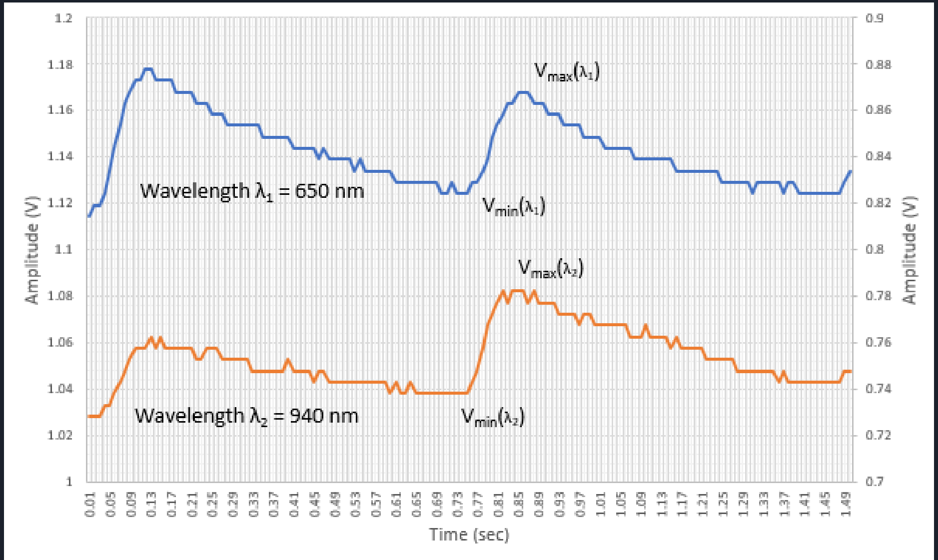
Heart Rate Monitor and Pulse Oximeter Testing



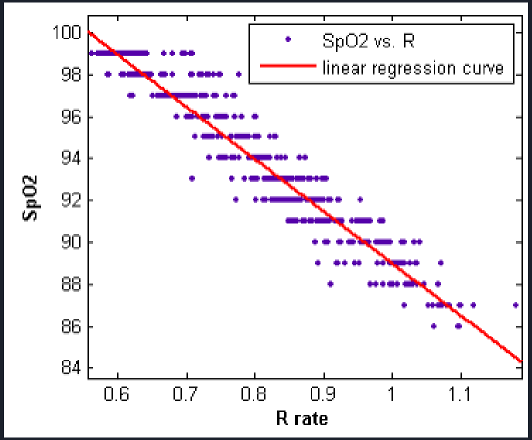
$$Heart\ Rate = \frac{60}{764\ ms} = 78.53\ BPM$$

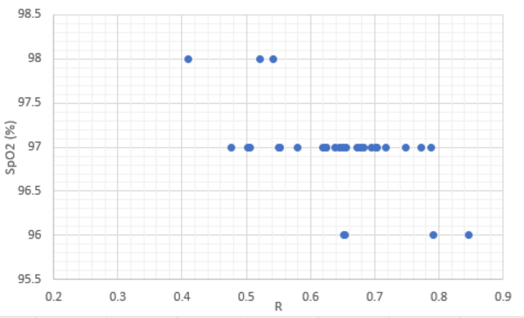
Teo

$SP0_2$ Calibration



$$R = \frac{\ln \frac{I_{\max}(\lambda_1)}{I_{\min}(\lambda_1)}}{\ln \frac{I_{\max}(\lambda_2)}{I_{\min}(\lambda_2)}}$$





Teo

Peak Detection

Algorithm: Finding Min and Subsequent Max Real-Time

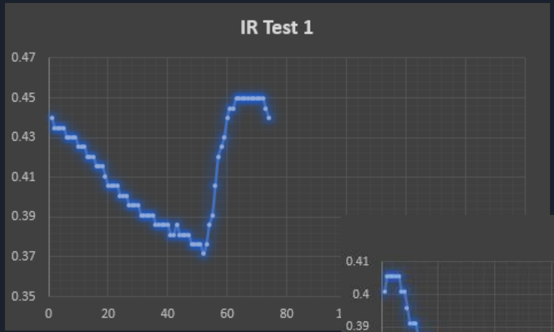
Input: Voltage values

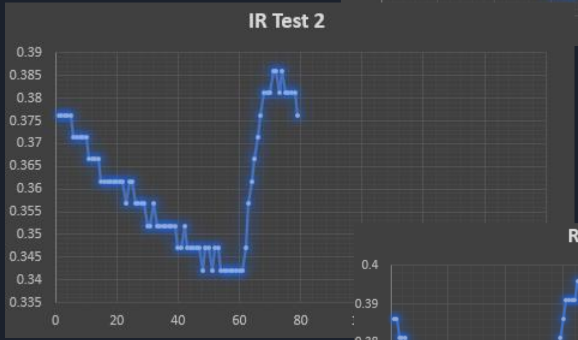
Output: min and max

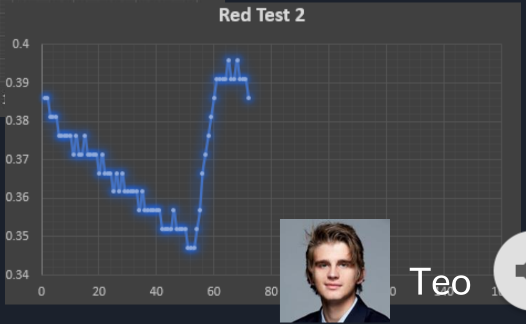
```
temp = 0;
flag = 0;

for index, value in signal do
    if value < temp
        if value <= min
            min = value;
        else if flag == 1
            break;
        else
            flag = 1;
        end
    else if value > temp
        max = value;
        flag = 0;
    end
    temp = value;
end

return max, min
```







Teo

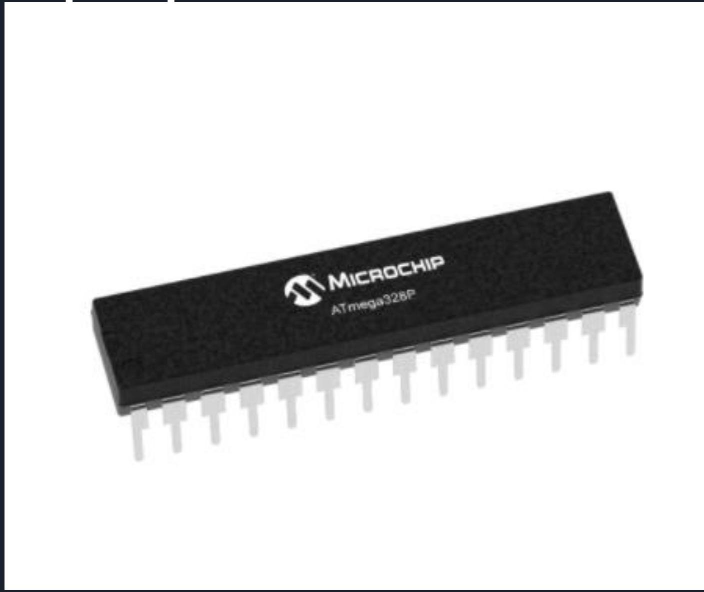
Heart Rate Accuracy

	SAMPLE SIZE	AVERAGE ERROR
70-75	6	2.63
76-80	31	2.58
81-85	19	4.55
86-90	26	2.27
91-95	18	2.49
96-100	2	4.05
101-105	1	4.76

Teo

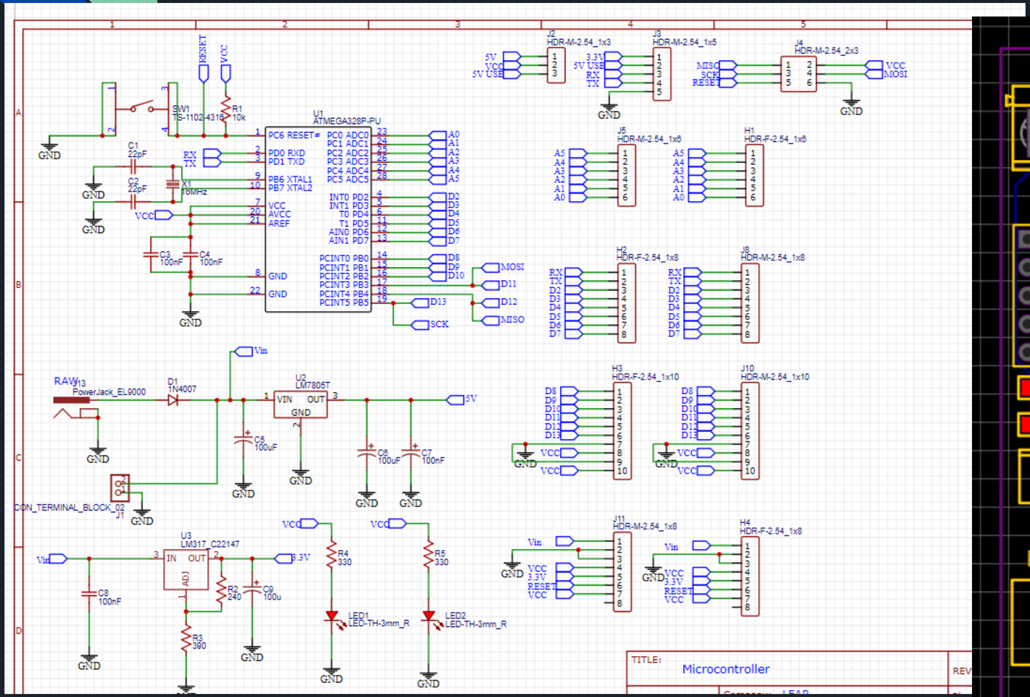
Microcontroller
-Atmega328p-pu

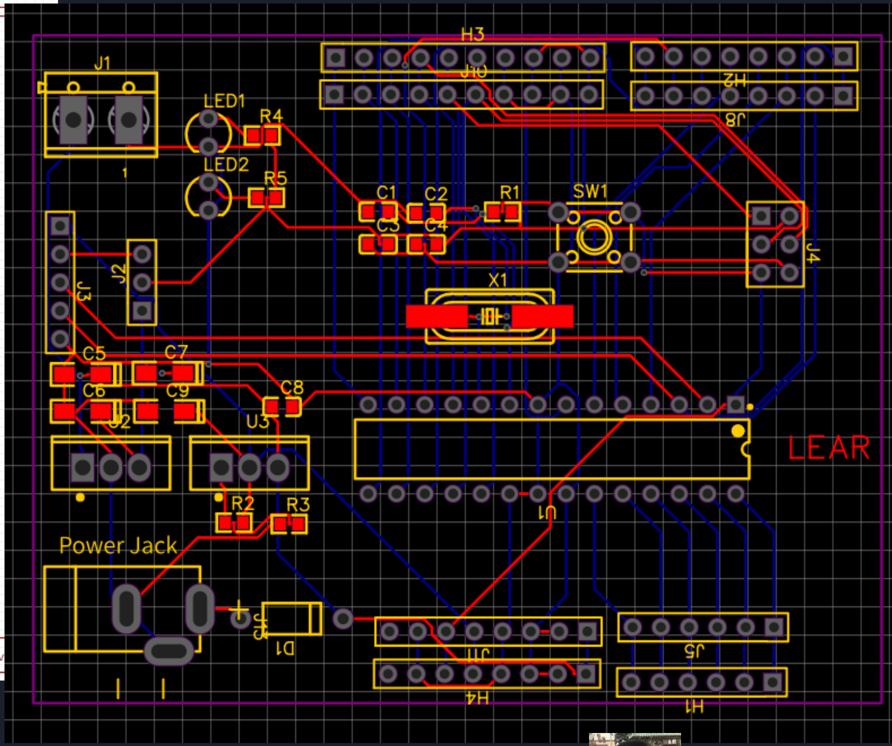
- Same with MCU on Arduino
- Program by Arduino IDE(Burn the bootloader)
- Programe with C/C++
- Easy to find documentation we need



Yong

PCB for Microcontroller Module





Wireless Communication Module

Goal: Serial communication streams between all sensors, microprocessing units, and the HoloLens

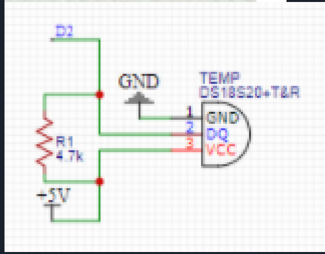
Parts we used: FS1000A Wireless RF Module, HC-05 Bluetooth Module,

nRF24L01 Transceiver, Zigbee Wireless Communication Module, DS18S20 Digital Temperature Sensor

Temperature Sensor

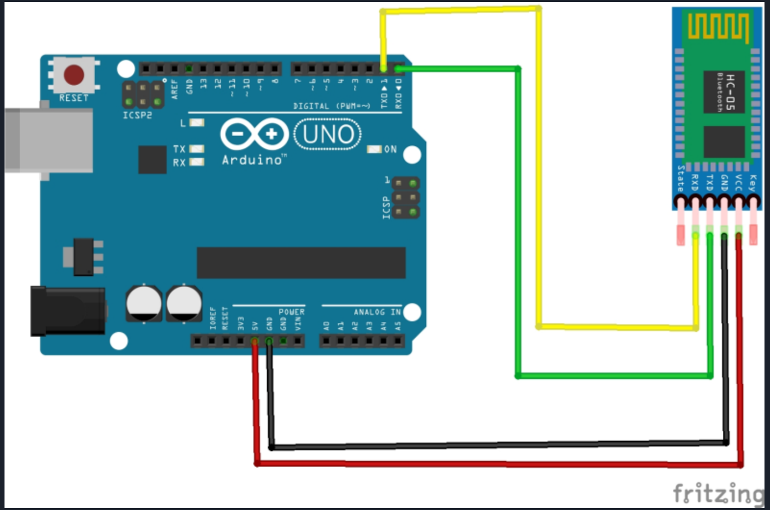
DS18S20 Temperature Sensor

- 3-pin connection
- One-wire structure (use Dallas Temperature Library)
- 4.7K ohm pull-up resistor
- 5V operating voltage



Yong

HC-05 Bluetooth Module



- 4-pin connection
- Works on 5V or 3.3V with resistor
- Easily interface with other bluetooth device

Theory use in project:

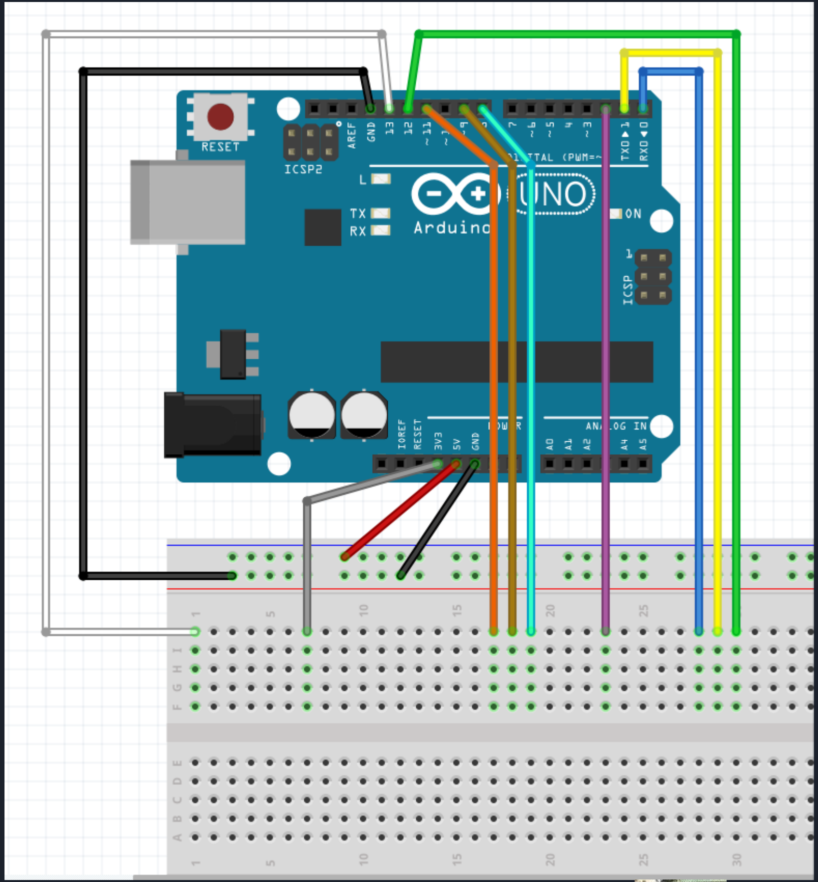
HC-05 will send data (temperature, heart rate,

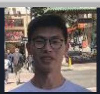
oxygen saturation to the computer that

connected to the HoloLens server, then server send data to HoloLens)

PCB Testing

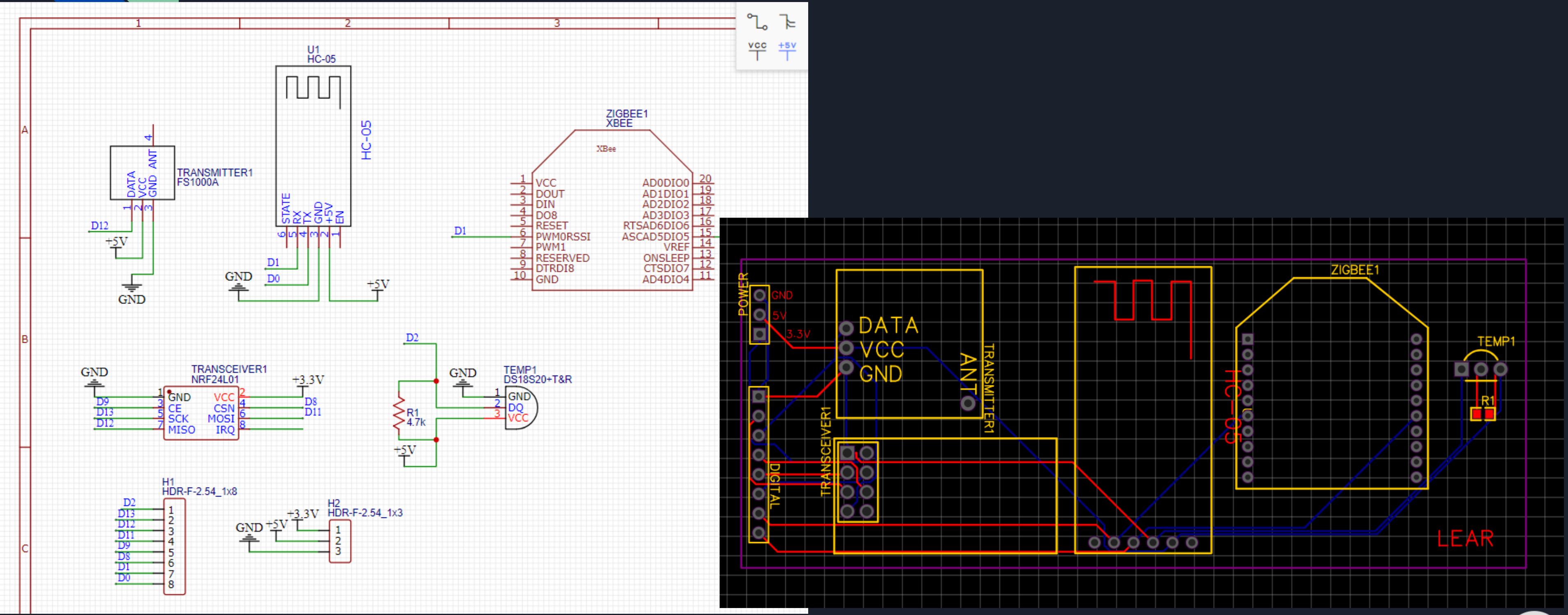
Before making PCB, we need to decide what ports we need and how can we connect it, so we make a schematic diagram to show all ports needed be connected to a breadboard.





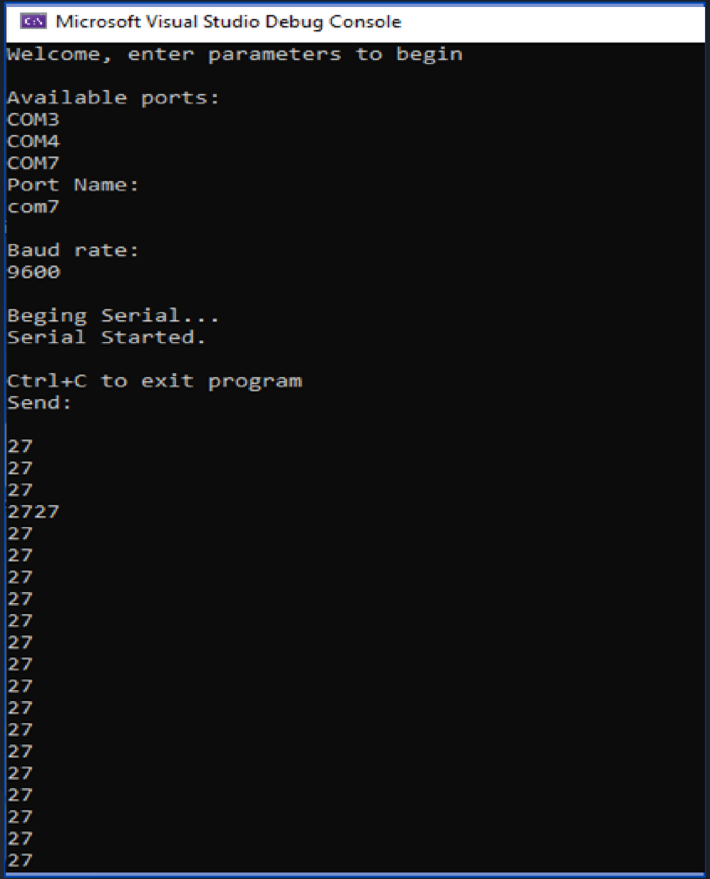
Yong

PCB for Wireless Communication Module



Yong

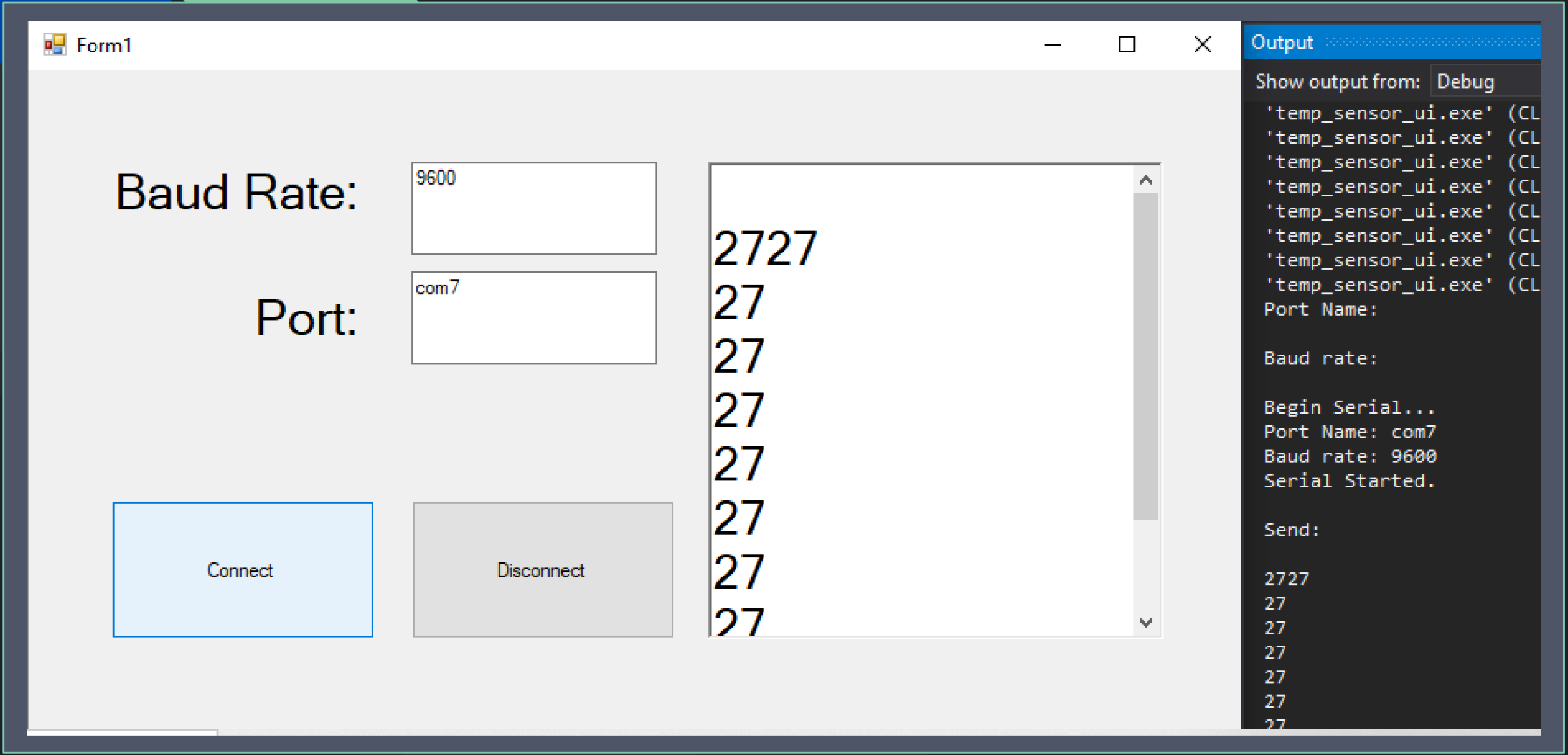
Sensor data using C# console application



- Desktop computer receives data via C# console application
- C# console application prints temperature to terminal

David

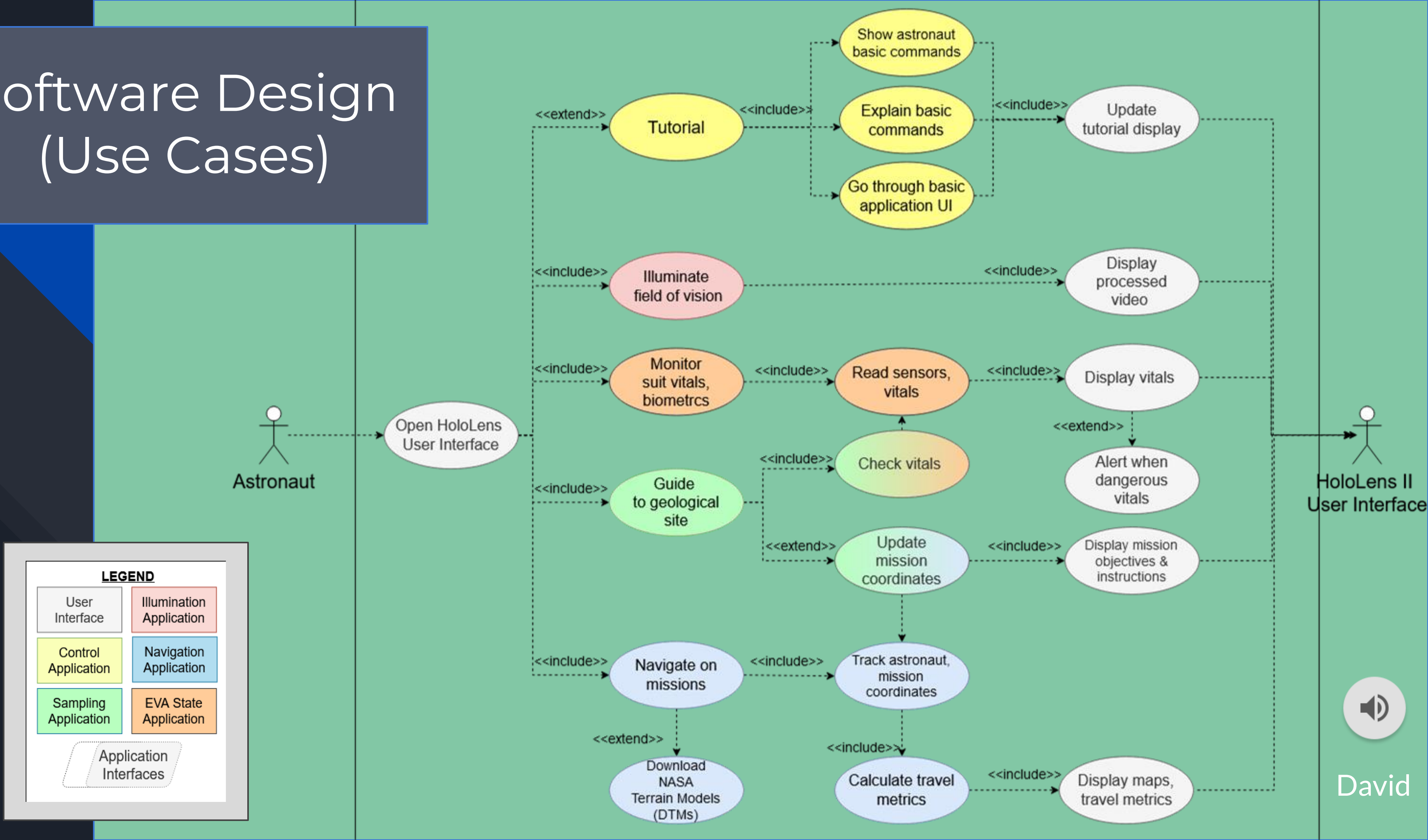
Sensor data with C# User Interface Application



David

Software Design

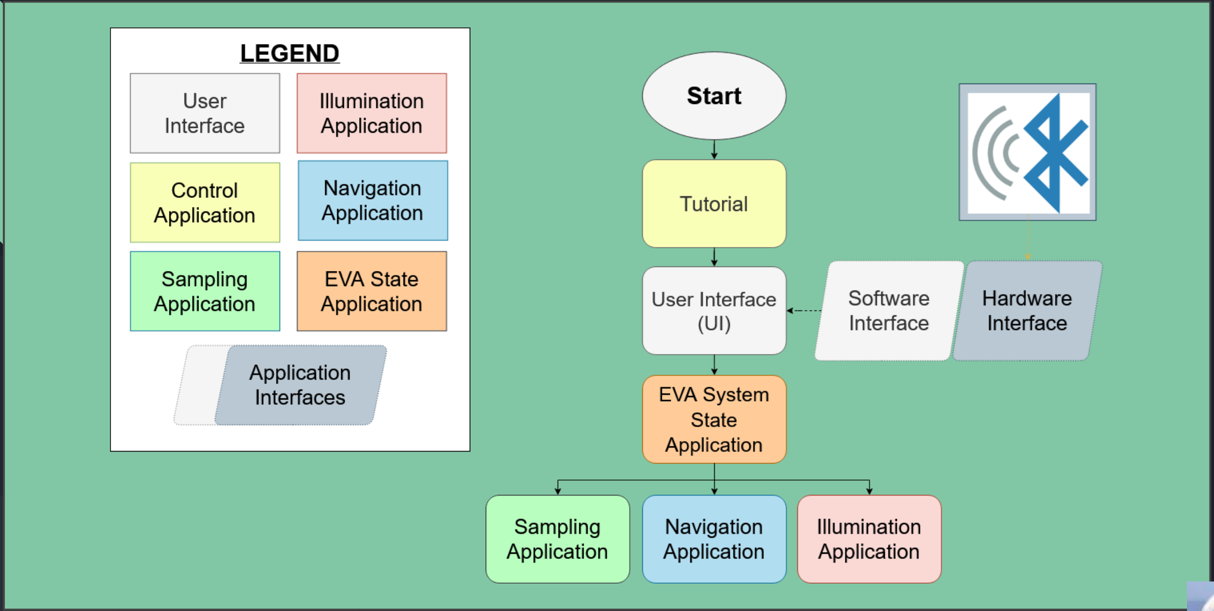
Software Design (Use Cases)



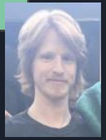
David

Software Design High-level Overview

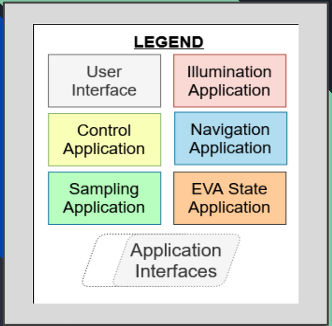
- NASA objectives:
  - Control/tutorial
  - EVA System State
  - Illumination
  - Navigation
  - Geological Sampling



David

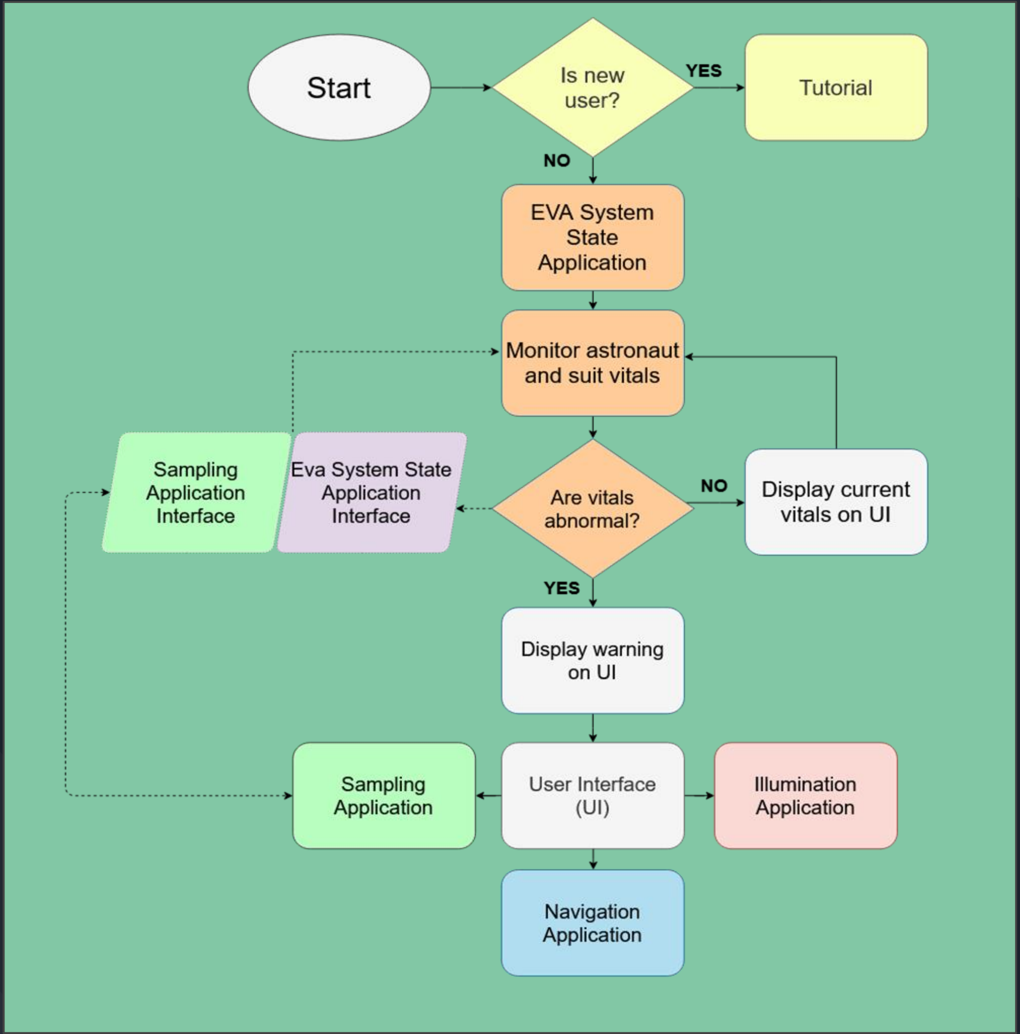


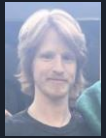
EVA System State



- User starts UI

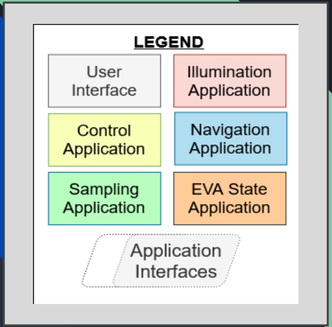
- Background processes record suit vitals (if mission starts)
  - Interfaces with Sampling application to warn astronaut if necessary



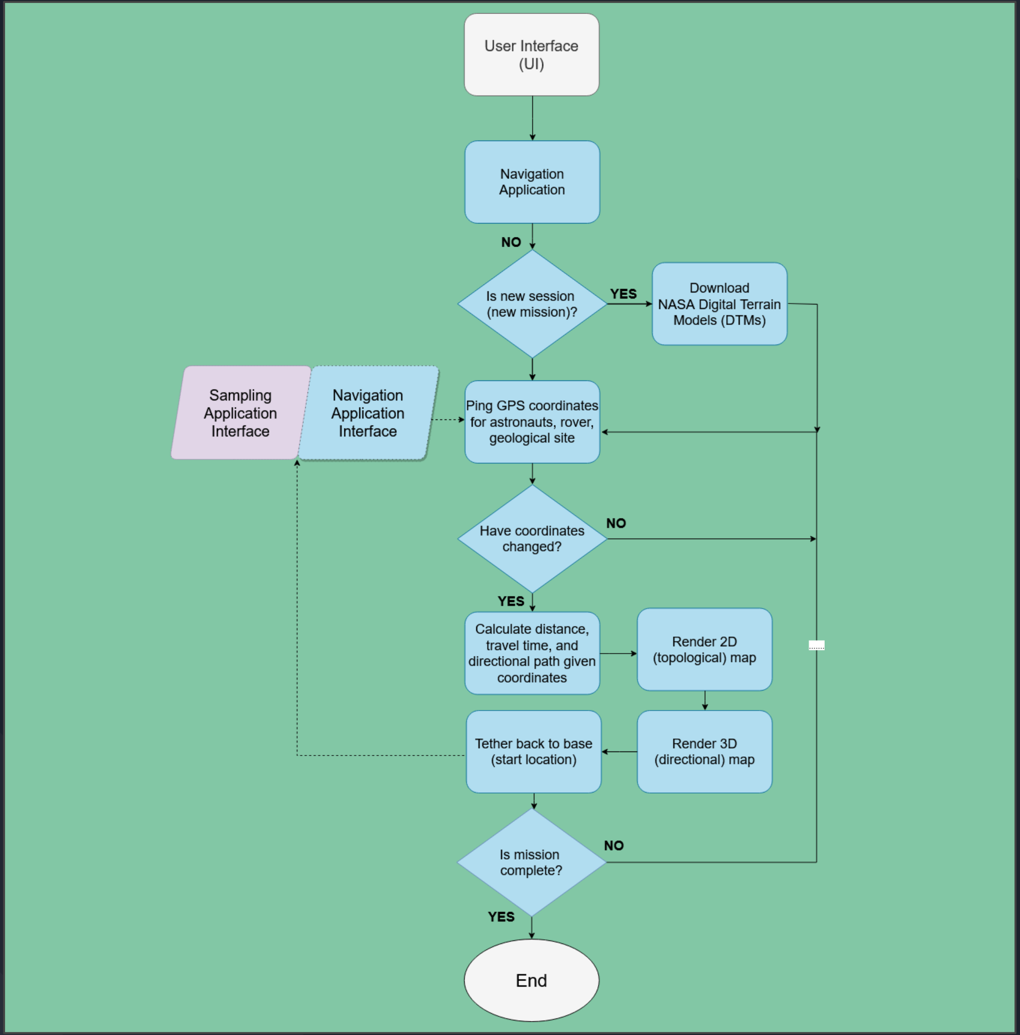


David

Navigation

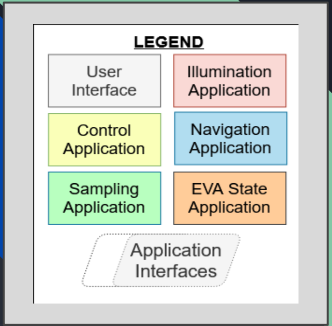


- Navigation application tethers astronaut to "home" location while they engage on mission
- Updates 2D navigation map in real-time relative to geological site
- Tracks current coordinates
- 3D directional arrow directs astronaut to geological site, home

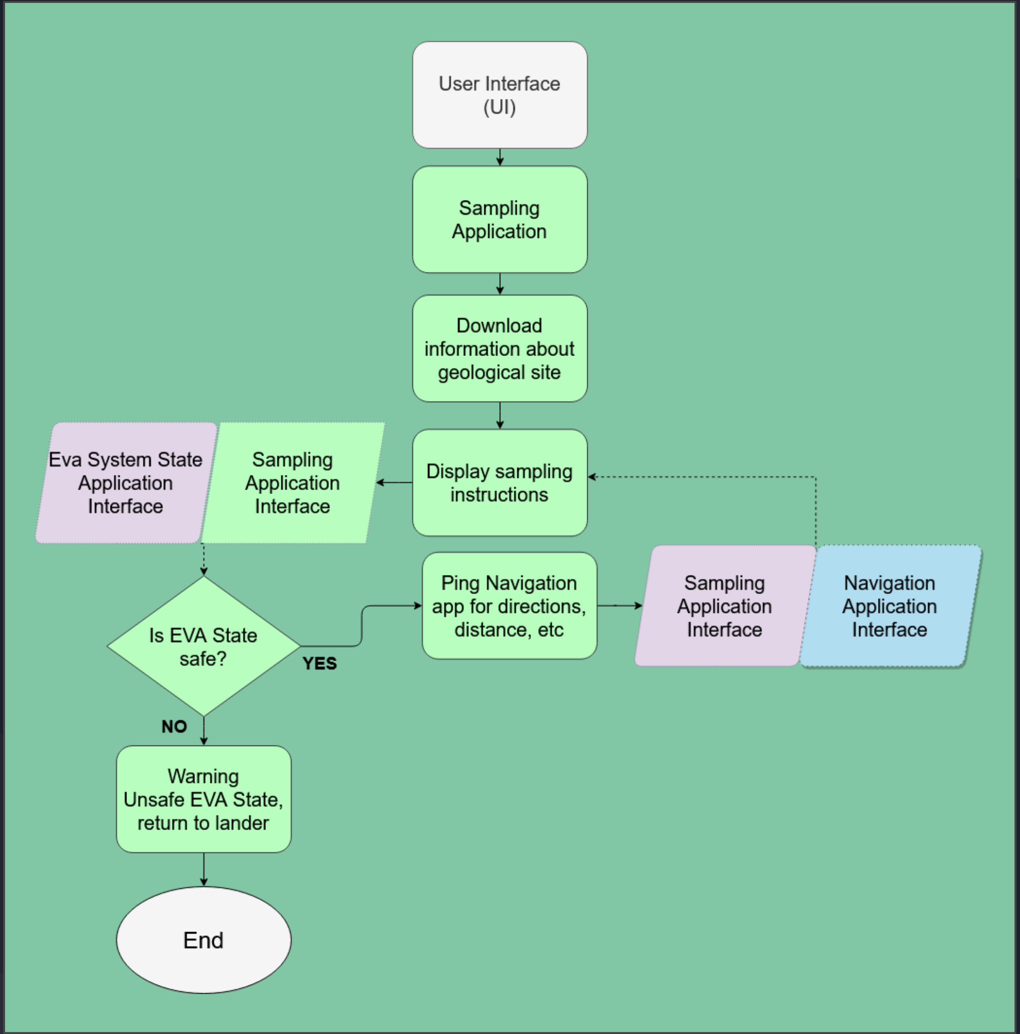



David

Sampling

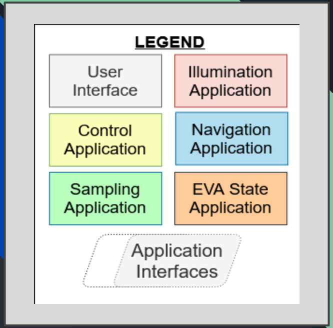


- Sampling (or “Mission”) application downloads all details about mission
- Displays notebook for mission containing instructions, tasks, tools, and other important information
- Interfaces with both Navigation and EVA State components

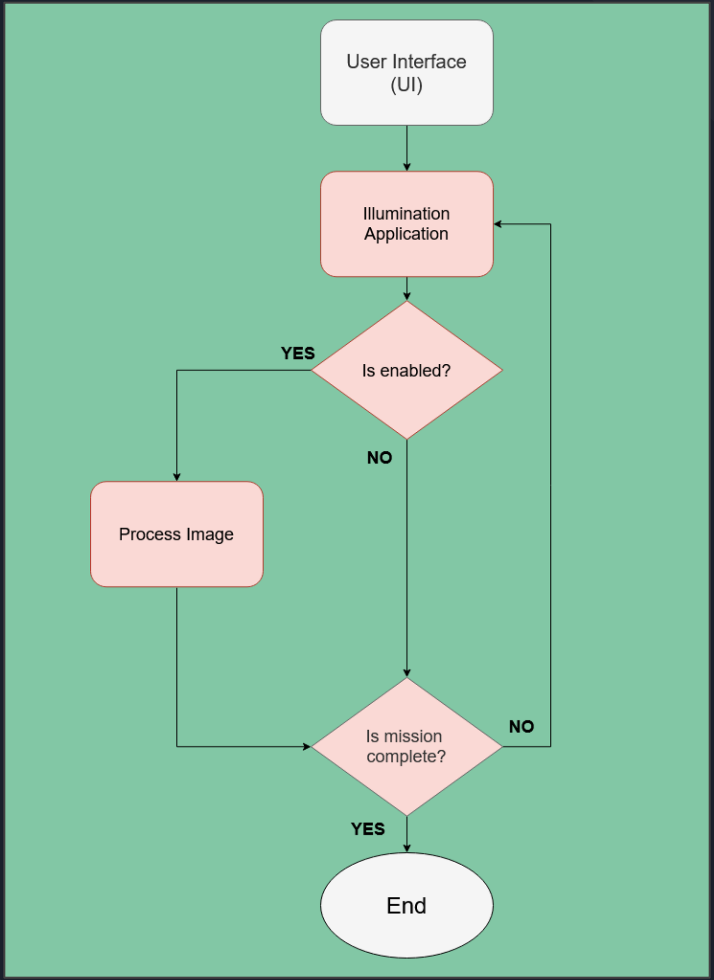


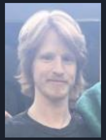
David

Illumination



- Illumination component is toggled on/off
- If on, the Hololens 2 video stream is processed to deal with high contrast environment





David

HoloLens Development Environment

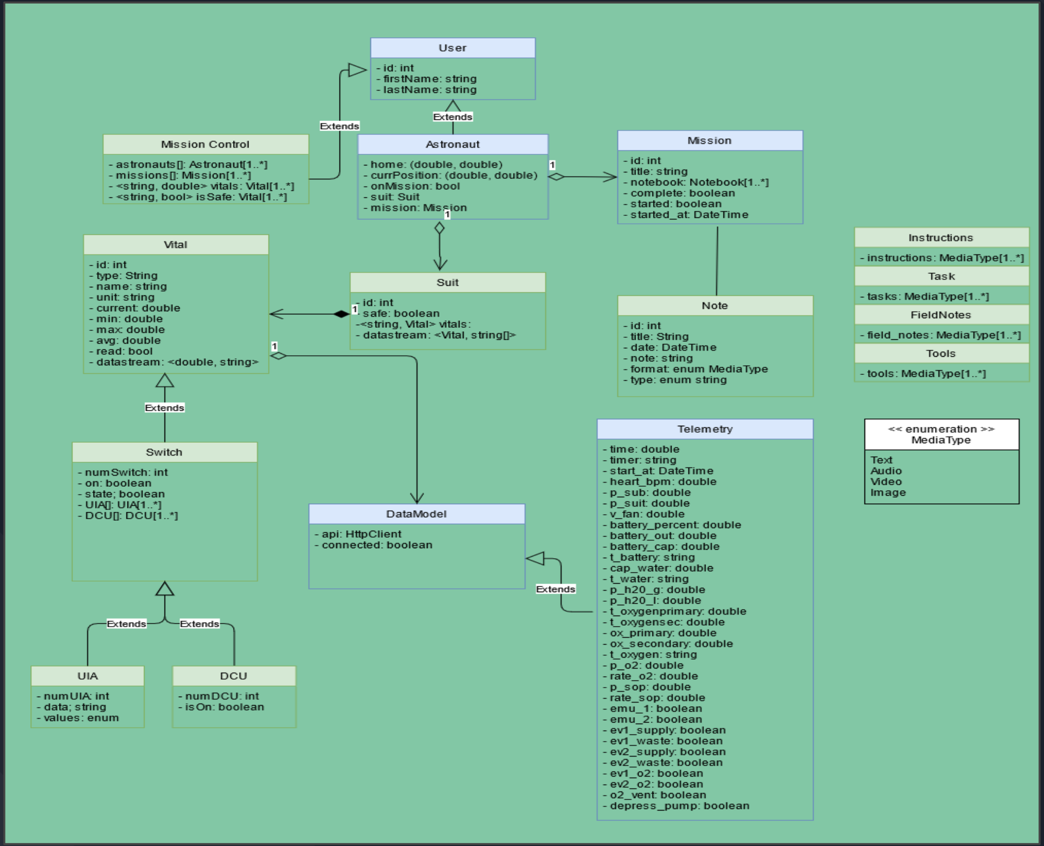
- Windows 10 OS, Visual Studio IDE, and Unity 3D engine
- Microsoft-mixed reality toolkit

David

Software Organization

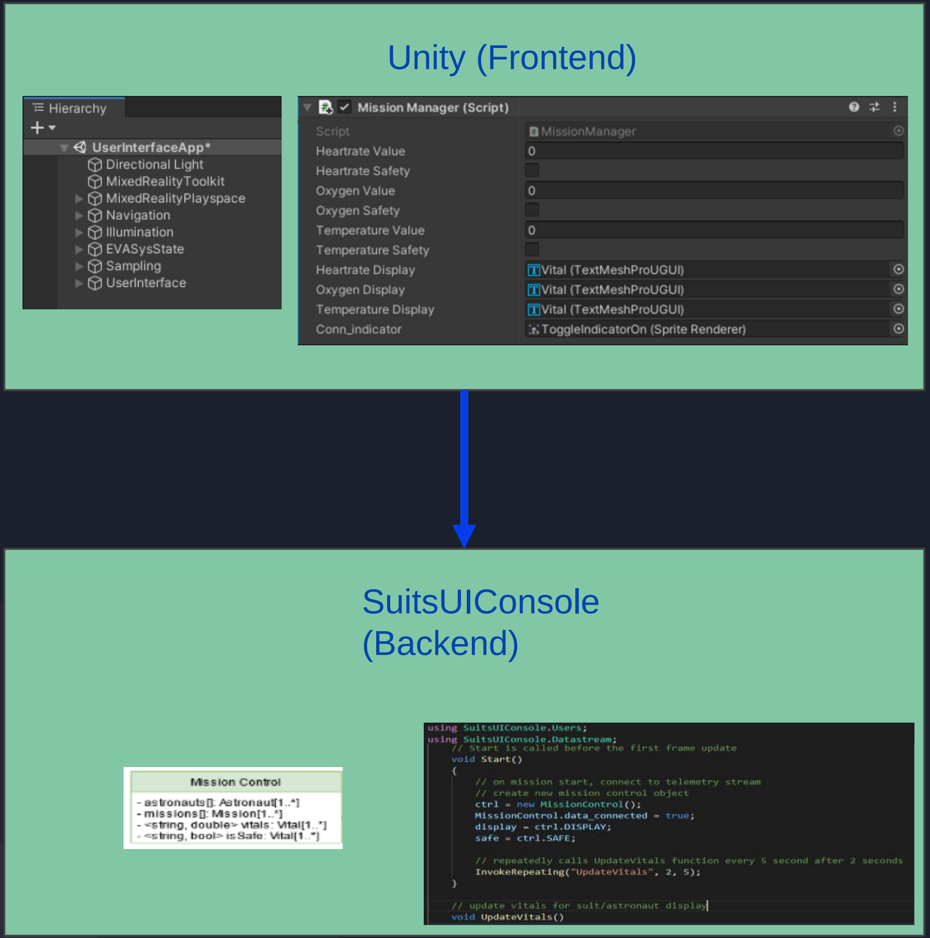
- Backend -- processes data
    - SuitsUIConsole namespace

- Frontend -- sends and receives data from backend, updates UI
    - Unity 3D engine



David

Dataflow between UI and codebase



- Each application is an individual “Game Object” in Unity
  - Encapsulate outlined classes
- Unity frontend instantiates a backend Mission Control object at run-time
- Game Objects use Mission Control object to process data and update UI

David

Illumination feature

David

Illumination Outline

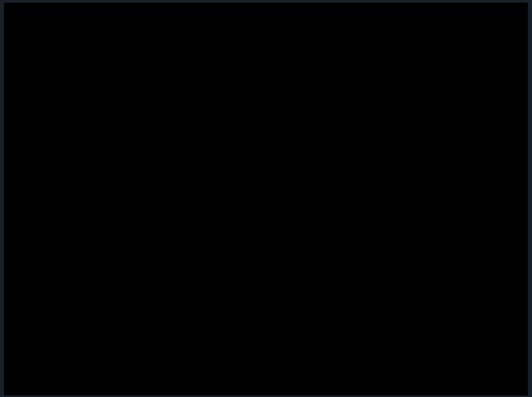
- Requirements
- Thresholding
- Contrast Limited Adaptive Histogram Equalization
- Implementation
- Tested Systems and Devices
- Issues

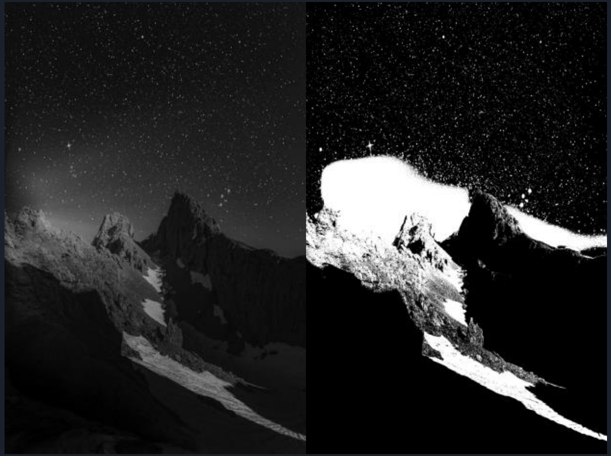
Illumination Requirements

- Image Processing through a camera in real time.
- Frames per second (fps) should be a common digital standard being 24-30 fps for movies or 60 fps.
- Gray Scale Processing was chosen for speed and simplicity.
- Main objective is increasing contrast in areas of interest

Thresholding

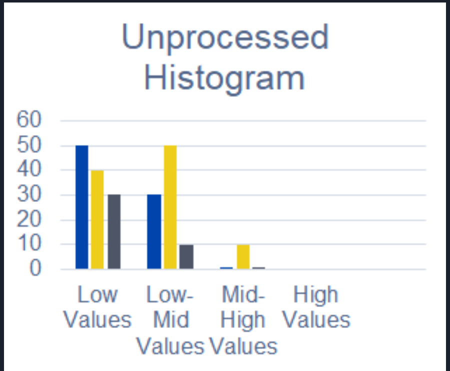
- Original Solution based on processing times
- Theory
  - Choosing a value and adjusting the image based on their relation to the chosen value.
  - Binary
  - Gray Level
- Not incredibly useful for our problem

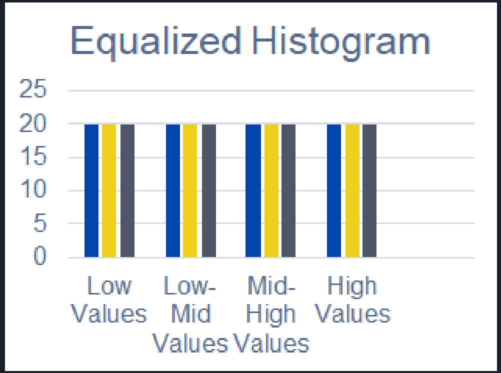




Contrast Limited Adaptive Histogram Equalization (CLAHE)

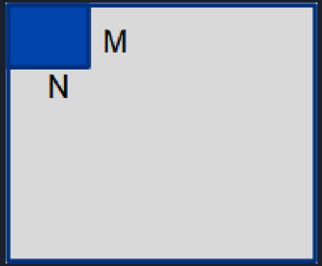
- Utilizes the Histogram Equalization Technique
- Histogram Equalization
  - Takes the values of the images and tries to equalize the frequency in which they appear.
  - Useful when there might be glare or washed-out images.
  - Global Transform

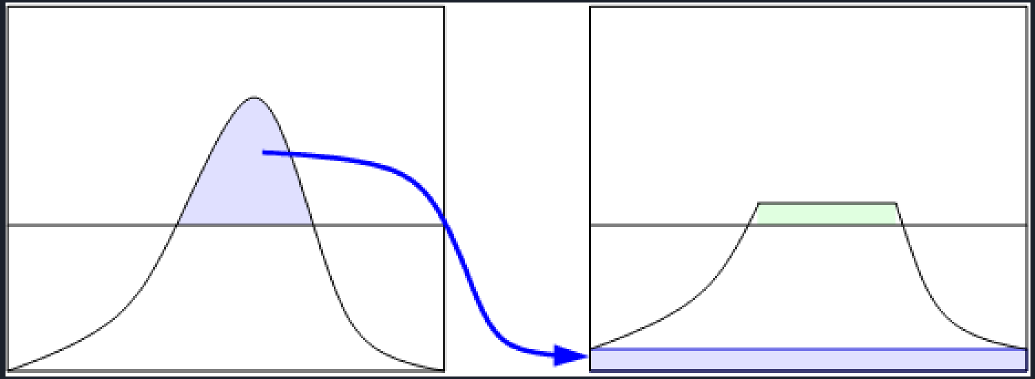




CLAHE cont.

- Adaptive Histogram Equalization
  - Tile Size (Neighborhood)
    - NxM grid
  - At Borders Bilateral Interpolation
- Contrast Limited
  - Clip Limit
  - Noise Limiting



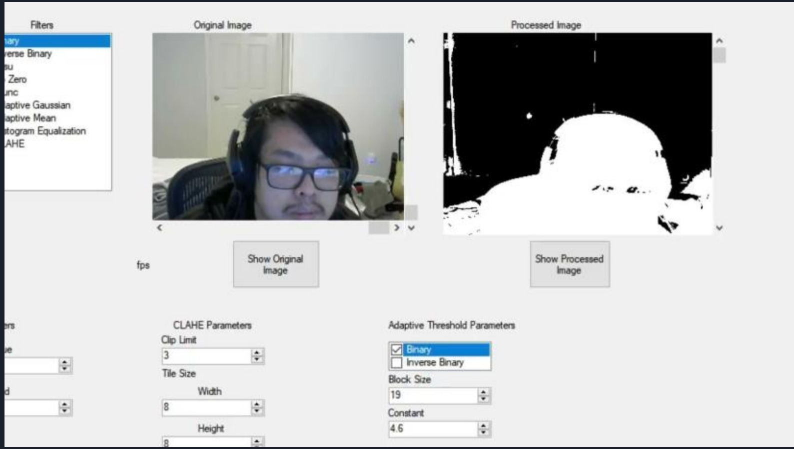


Implementation

- Python (Prototyping)
  - OpenCV
  - Google Colaboratory (Jupyter Notebook)
  - Anaconda Spyder
  - Didn’t display updated matrixes well
- C#
  - Emgu.CV (.Net wrapper of OpenCV)
  - Visual Studio
  - Better Graphical User Interface (GUI)
  - Testing.

Issues

- The original python programming did not display updated frames very well
- Multiple Filters at once created stuttering of the frame
- Preprocessing Technique
- Permissions from Microsoft for HoloLens Access

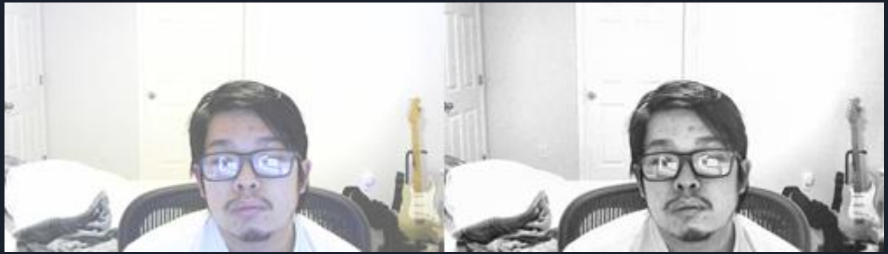


Additional Applications

- Post Processing Technique
  - Can be used in conjunction with a post processing technique to clear isolate certain parts of the image
- Reconstruction
  - 3D mapping for future exploration

Systems and Devices

- Webcam with Windows (Video)
- Webcam with OSX (Video)
- Pi Camera with Linux (Video)
- Dash Cam Data (Video)
- Camera Images (.jpeg, .tiff, …)



Budget and Finances

Budget			
Item	Price/per unit	Quantity	Price/total
Microsoft Hololens II	$ 3,500.00	1	$ 3,500.00
PCB Board	$ 20.00	5	$ 100.00
Photosensor	$ 10.00	1	$ 10.00
Microcontroller	$ 30.00	1	$ 30.00
Temperature Sensor	$ 11.00	1	$ 11.00
Push Button	$ 1.00	10	$ 10.00
Wireless Communication Module	$ 9.00	1	$ 9.00
Pressure Sensor	$ 9.00	1	$ 9.00
LED Sources	$ 2.00	10	$ 20.00
Wiring	$ 0.50	20	$ 10.00
Housing for External Sensors	$ 15.00	1	$ 15.00
Wristwatch Housing	$ 15.00	2	$ 30.00
LED Detector	$ 5.00	2	$ 10.00
Total Expenses			$ 3,764.00
Funding			
Florida Space Grant Consortium			$ 2,000.00
George Jackson Foundation			$ 1,000.00
Personal Contributions			$ 1,000.00
Total Funding			$ 4,000.00
Surplus			$ 236.00

Teo

Questions?

Thank you!