Augmented Reality Informational System (ARIS)

Authors/Co-Leads: Fisseha Tegegne, Salvador Baez, John Scovell
CS Graduate Members (2023): Venkata Vutukuru, Parth Akre, Karthik Mukka
CM/TM: Tomas Materdey

Design Validation Report
April 8th, 2024

University of Massachusetts Boston | *2023-2024*

I. Problem Definition

A. *Overview*

ARIS is an augmented reality “assistant” that will enable astronauts to access important biomedical data, such as BPM, body temperature, acceleration, oxygen levels, and fluid levels. This is accomplished via sensor nodes that are connected to sensor hubs wirelessly from an Arduino Nano to two pairs of receiving/transmitting XBee modules. The two sensor hubs are centralized by corresponding Raspberry Pi’s, which allow for all data and communications to be monitored, recorded, and transferred between system components via servers created with Python coding language. Machine learning algorithms are initiated to ensure the astronaut’s vitals are within safe and normal limits, and will alert them if abnormal data is collected.

All data shall be displayed and easily accessible on the user’s HMD (Head Mounted Display). For this project, a HoloLens will be utilized for this function, and a UI (User Interface) - hosted on a JavaScript server - will be created with Unity software. Importantly, the ARIS user will also control a rover equipped with a native USB camera, a RealSense D435i, and a Raspberry Pi. The method of communication and UI interaction must be decided, and could be implemented via methods such as voice commands or finger gestures. GPS navigation will be utilized alongside the UI to display a 2D map on the HoloLens, which will display moving “markers” that the user will follow.

B. *Motivations*

The motivation behind the project is to enhance the astronaut’s experience and improve their efficiency during space missions by utilizing augmented reality technology. This technology can provide real-time information, virtual guidance, and interactive visualizations, offering a more intuitive and immersive environment for astronauts to work in.

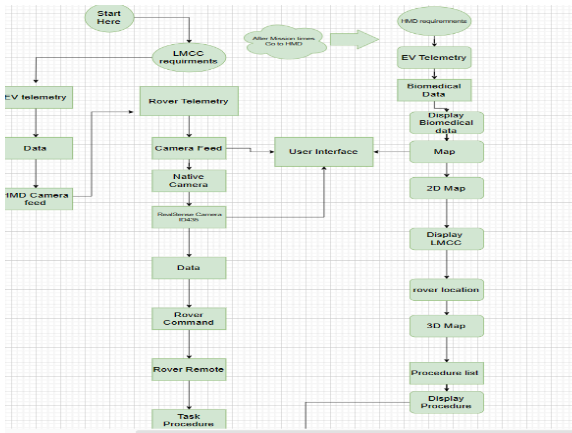
C. *Customer Requirements*

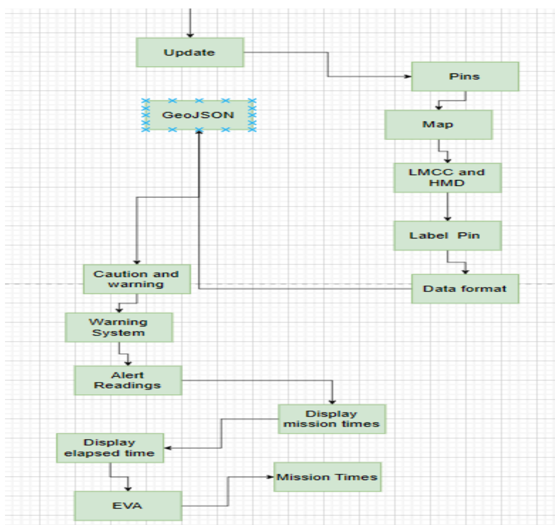
Customer requirements refers to the needs and expectations of the customers involved in the project. In the context of augmented reality for astronauts, customer requirements could include safety, user interface, real-time data, durability and reliability, compatibility, and power efficiency.

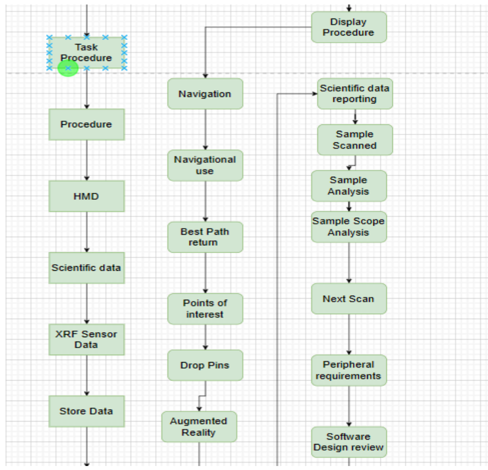
D. *Engineering Requirements*

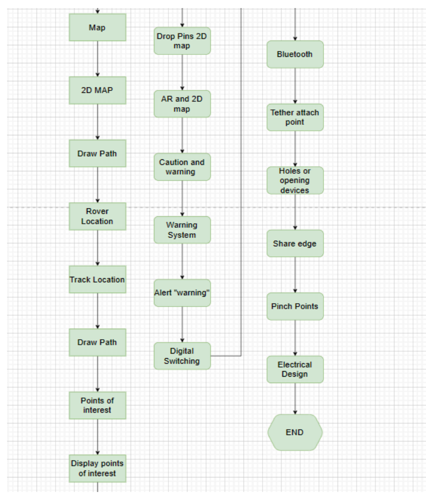
Engineering requirements, on the other hand, focus on the technical aspects of developing the augmented reality system. These requirements could include software development, integration with the spacecraft, calibration and accuracy, user experience optimization, and hardware specifications.

II. System Diagram









III. Project Management

A. *Overview*

Each week, a co-lead team member submitted weekly reports to recap our meetings and track progress. All documents are shared and updated on a one-drive with access to everyone. The project is managed in MS Project and focuses on five PSSC's (Project Specific Success Criteria).

B. *Team Roles*

- Sal Baez - *Co-Lead*
  - Machine learning, Software Engineering, Team Management.
- Fisseha Tegegne - *Co-Lead*
  - Battery and Power Management, Circuitry, Wiring.
- John Scovell - *Co-Lead/Junior Member*
  - Rover build and commanding, PCB, System Integration.

- Parth Akre - *CS Grad Member, Fall 2023*
    - HoloLens GUI Development, Augmented Reality, Unity.
- Venkata Vutukuru - *CS Grad Member, Fall 2023*
    - LMCC development, RealSense Camera Depth Sensing, System Integration.
- Karthik Mukka - *CS Grad Member, Fall 2023*
    - Machine learning, rover development, LMCC support.

IV. MS Project

Milestone 1	**21 days**	**Wed 9/20/23**	**Sun 10/15/23**				**100%**
PSSC 1: Data Comm	**21 days**	**Wed 9/20/23**	**Sun 10/15/23**		**Venkata,Sal,Fisl**		**100%**
Design	21 days	Wed 9/20/23	Sun 10/15/23				100%
Alternative Tasks	21 days	Wed 9/20/23	Sun 10/15/23				100%
PSSC 2: Map	**21 days**	**Wed 9/20/23**	**Sun 10/15/23**		**Karthik**		**100%**
Design	21 days	Wed 9/20/23	Sun 10/15/23				100%
Alternative Tasks	21 days	Wed 9/20/23	Sun 10/15/23				100%
PSSC 3: Caution	**21 days**	**Wed 9/20/23**	**Sun 10/15/23**		**Parth**		**100%**
Design	21 days	Wed 9/20/23	Sun 10/15/23				100%
Alternative Tasks	21 days	Wed 9/20/23	Sun 10/15/23				100%
PSSC 4: Rover	**21 days**	**Wed 9/20/23**	**Sun 10/15/23**		**Sal,Fish,John**		**100%**
Design	21 days	Wed 9/20/23	Sun 10/15/23				100%
Alternative Tasks	21 days	Wed 9/20/23	Sun 10/15/23				100%
PSSC 5: GUI	**21 days**	**Wed 9/20/23**	**Sun 10/15/23**		**Parth**		**100%**
Design	21 days	Wed 9/20/23	Sun 10/15/23				100%
Alternative Tasks	21 days	Wed 9/20/23	Sun 10/15/23				100%
Report: Problem Definition	**7 days**	**Sun 10/1/23**	**Mon 10/9/23**		**Sal,Fish,John**		**100%**
Draft	3 days	Sun 10/1/23	Tue 10/3/23				100%
Final	5 days	Tue 10/3/23	Mon 10/9/23				100%
Presentation	**7 days**	**Sun 10/1/23**	**Mon 10/9/23**		**Sal,Fish,John**		**100%**
Draft	3 days	Sun 10/1/23	Tue 10/3/23				100%
Final	5 days	Tue 10/3/23	Mon 10/9/23				100%

Milestone 2	**50 days?**	**Mon 10/16/23**	**Fri 12/22/23**				**100%**
PSSC 1: Data Comm	**42 days?**	**Mon 10/16/23**	**Tue 12/12/23**				**100%**
Design	**42 days**	**Mon 10/16/23**	**Tue 12/12/23**				**100%**
Understand previous team's progress	12 days	Mon 10/16/23	Tue 10/31/23		Fish,John, Karthik, Parth,Sal,Venkat		100%
Deciding add-ons	12 days	Mon 10/16/23	Tue 10/31/23		Fish,John,Sal		100%
Order camera	16 days	Mon 10/16/23	Mon 11/6/23		John		100%
Block Diagrams	24 days	Mon 10/16/23	Thu 11/16/23		Sal		100%
Camera feed	10 days	Fri 11/10/23	Thu 11/23/23		Fish,John,Sal,Ve		100%
~~Design PCB~~	~~8 days~~	~~Fri 11/10/23~~	~~Tue 11/21/23~~		~~Fish,John,Sal~~		0%
~~Order PCB~~	~~3 days~~	~~Wed 11/22/23~~	~~Fri 11/24/23~~	37	~~Fish,John,Sal~~		0%
Building	**42 days**	**Mon 10/16/23**	**Tue 12/12/23**				**100%**
Connect sensor hubs	6 days	Tue 10/31/23	Tue 11/7/23		Fish,Sal,John		100%
~~Connect RealSense Camera~~	~~6 days~~	~~Fri 12/1/23~~	~~Fri 12/8/23~~				0%
Connect Native Camera	4 days	Thu 11/16/23	Tue 11/21/23		Fish,John,Sal		100%
~~Mount PCB~~	~~2 days~~	~~Mon 11/27/23~~	~~Tue 11/28/23~~	38	~~Fish,John,Sal~~		0%
LMCC Development	10 days	Fri 11/10/23	Thu 11/23/23		Venkata		100%
Validation Test	**42 days**	**Mon 10/16/23**	**Tue 12/12/23**				**100%**
Confirm sensor hubs are working properly	8 days	Wed 11/8/23	Fri 11/17/23	40	Karthik		100%
~~Confirm sensor node data and machine learning~~	~~12 days~~	~~Wed 11/8/23~~	~~Thu 11/23/23~~		~~Karthik~~		0%
~~Confirm server is working~~	~~12 days~~	~~Thu 11/16/23~~	~~Fri 12/1/23~~		~~Venkata~~		0%
~~Confirm native camera feed~~	~~12 days~~	~~Thu 11/16/23~~	~~Fri 12/1/23~~		~~John,Karthik~~		0%
~~Confirm RealSense feed~~					~~Venkata~~		0%

~~PSSC 2: Map~~	~~35 days?~~	~~Mon 10/16/23~~	~~Fri 12/1/23~~	~~30~~		~~0%~~
~~Design~~	~~1 day?~~	~~Mon 10/16/23~~	~~Mon 10/16/23~~			~~0%~~
~~Building~~	~~1 day?~~	~~Mon 10/16/23~~	~~Mon 10/16/23~~			~~0%~~
~~Validation Test~~	~~1 day?~~	~~Mon 10/16/23~~	~~Mon 10/16/23~~			~~0%~~
~~PSSC 3: Caution~~	~~35 days?~~	~~Mon 10/16/23~~	~~Fri 12/1/23~~	~~30~~		~~0%~~
~~Design~~	~~1 day?~~	~~Mon 10/16/23~~	~~Mon 10/16/23~~			~~0%~~
~~Building~~	~~1 day?~~	~~Mon 10/16/23~~	~~Mon 10/16/23~~			~~0%~~
~~Validation Test~~	~~1 day?~~	~~Mon 10/16/23~~	~~Mon 10/16/23~~			~~0%~~
PSSC 4: Rover	**42 days?**	**Mon 10/16/23**	**Tue 12/12/23**			**100%**
Design	**42 days**	**Mon 10/16/23**	**Tue 12/12/23**			**100%**
Choose and order rover	16 days	Mon 10/16/23	Mon 11/6/23		John	100%
Utilizing Rasperry Pi to send/receive commands	11 days	Tue 11/14/23	Tue 11/28/23		John	100%
CAD model to mount RealSense on Rover	11 days	Tue 11/28/23	Tue 12/12/23		John	100%
Building	**19 days**	**Thu 11/16/23**	**Tue 12/12/23**			**100%**
Build Rover	9 days	Thu 11/16/23	Tue 11/28/23		John	100%
~~Print and attach CAD to Rover/RealSense~~	~~8 days~~	~~Fri 12/1/23~~	~~Tue 12/12/23~~		~~Sal,Fish,John~~	~~0%~~
Validation Test	**11 days?**	**Tue 11/28/23**	**Tue 12/12/23**			**100%**
~~Confirm both native camera and RealSense camera feed~~	~~11 days~~	~~Tue 11/28/23~~	~~Tue 12/12/23~~		~~Karthik,Venkata~~	~~0%~~
Send commands to move Rover	1 day	Tue 11/28/23	Tue 11/28/23		John	100%

PSSC 5: GUI	**42 days?**	**Mon 10/16/23**	**Tue 12/12/23**			**100%**
Design	**23 days**	**Mon 10/16/23**	**Wed 11/15/23**			**100%**
Design display for HMD	23 days	Mon 10/16/23	Wed 11/15/23		Parth	100%
Building	**20 days**	**Wed 11/15/23**	**Tue 12/12/23**			**100%**
Complete display on HMD	13 days	Wed 11/15/23	Fri 12/1/23		Parth	100%
Build interactive system to communicate with HMD	8 days	Fri 12/1/23	Tue 12/12/23		Parth	100%
~~Validation Test~~	~~8 days?~~	~~Fri 12/1/23~~	~~Tue 12/12/23~~			~~0%~~
~~Display real-time data~~	~~8 days~~	~~Fri 12/1/23~~	~~Tue 12/12/23~~		~~Karthik,Parth~~	~~0%~~
~~Interact with data and communicate with LMCC/Rover~~	~~8 days~~	~~Fri 12/1/23~~	~~Tue 12/12/23~~		~~Karthik,Parth, Venkata~~	~~0%~~
Report	**10 days**	**Fri 12/1/23**	**Thu 12/14/23**		**Sal,Fish,John**	**100%**
Draft	7 days	Fri 12/1/23	Mon 12/11/23			100%
Final	4 days	Mon 12/11/23	Thu 12/14/23			100%
Presentation	**10 days?**	**Mon 11/20/23**	**Fri 12/1/23**		**Sal,Fish,John**	**100%**
Draft	7 days	Mon 11/20/23	Tue 11/28/23			100%
Final	3 days	Wed 11/29/23	Fri 12/1/23			100%

Milestone 3	**35 days?**	**Mon 1/29/24**	**Fri 3/15/24**			**79%**
Action: LMCC to Rover Communication	**11 days?**	**Mon 1/29/24**	**Mon 2/12/24**		**John**	**73%**
Create server and establish connection	4 days	Mon 1/29/24	Thu 2/1/24			100%
Control rover from LMCC	1 day	Mon 1/29/24	Mon 1/29/24			100%
Send Video to LMCC		Mon 1/29/24				10%
Build LMCC GUI		Mon 1/29/24				50%
Data validation Test	1 day	Mon 1/29/24	Mon 1/29/24			100%
Connect/Send RealSense feed to LMCC						0%
Action: HoloLens to LMCC Communication	**11 days?**	**Mon 1/29/24**	**Mon 2/12/24**		**Sal**	**30%**
Install Unity	1 day	Mon 1/29/24	Mon 1/29/24			100%
Follow Parth's HoloLens Instructions		Mon 1/29/24				20%
Make HoloLens work						0%
Display sensor hub csv data on HoloLens						0%
Action: Sensor Data	**35 days?**	**Mon 1/29/24**	**Fri 3/15/24**		**Fish,John**	**95%**
Rebuild sensor hubs	23 days	Mon 1/29/24	Wed 2/28/24			100%
Debug code and fix errors	9 days	Wed 2/28/24	Sun 3/10/24			100%
Order new PCB	1 day	Mon 1/29/24	Mon 1/29/24			100%
Test sensors and replace bad ones		Mon 1/29/24				20%
Send csv data to HoloLens Unity project						0%

Action: Build team webpage	**31 days?**	**Mon 1/29/24**	**Sun 3/10/24**		**Sal,Fish,John**	**99%**
Email TA for password	1 day	Fri 2/2/24	Fri 2/2/24			100%
Watch/follow tutorial	1 day	Wed 2/7/24	Wed 2/7/24			100%
Upload first webpage requirements	1 day	Wed 2/7/24	Wed 2/7/24			100%
Upload second webpage requirements	1 day	Wed 3/6/24	Wed 3/6/24			100%
Action: Develop LMCC GUI	**10 days?**	**Tue 2/13/24**	**Mon 2/26/24**			**43%**
Choose interface	1 day	Tue 2/13/24	Tue 2/13/24			100%
Display all communications on GUI		Tue 2/13/24				20%
Validate all connections		Tue 2/13/24				10%
Design Validation Report	**11 days**	**Fri 3/1/24**	**Fri 3/15/24**			**0%**
Draft						0%
Final						0%
Design Validation Presentation	**11 days**	**Fri 3/1/24**	**Fri 3/15/24**			**20%**
Draft		Fri 3/1/24				40%
Final						0%

Milestone 4	38 days?	Sat 3/16/24	Tue 5/7/24			0%
PSSC 1: Data Comm	38 days?	Sat 3/16/24	Tue 5/7/24			0%
Design						0%
Building						0%
Validation Test						0%
PSSC 2: Map	38 days?	Sat 3/16/24	Tue 5/7/24			0%
Design						0%
Building						0%
Validation Test						0%
PSSC 3: Caution	38 days?	Sat 3/16/24	Tue 5/7/24			0%
Design						0%
Building						0%
Validation Test						0%
PSSC 4: Rover	38 days?	Sat 3/16/24	Tue 5/7/24			0%
Design						0%
Building						0%
Validation Test						0%
PSSC 5: GUI	38 days?	Sat 3/16/24	Tue 5/7/24			0%
Design						0%
Building						0%
Validation Test						0%
Report	38 days	Sat 3/16/24	Tue 5/7/24			0%
Draft						0%
Final						0%
Presentation	38 days	Sat 3/16/24	Tue 5/7/24			0%
Draft						0%
Final						0%

V. Parts, Budget, Resources, Tools & Skills

A. Parts
   1. RealSense Di435, HoloLens, Raspberry Pi, Sunfounder Rover, Module V2 Camera, XBee modules, GPS breakout, PCB

B. Budget
   1. Total Budget of $1,500
   2. Spent $900 on:
      a) RealSense D435i - $354.88
      b) Sunfounder Smart Video Car Kit - $454.87
      c) Raspberry Pi 4 Model B - $61.69
      d) PCB Print from OSHPark - $23

C. Resources
   1. NASA SUITS description document
   2. 2022-2023 team 4 project document and video
   3. Weekly CM/TM Meetings
   4. UMass Boston Tools and Supplies

D. Tools
   1. KiCad, Fusion 360, LTspice, Visual Studio, Unity,

E. Skills
   1. Python, server development, data communication, GUI design, Unity, C#, Raspberry Pi, Augmented Reality, Machine Learning

VI. Design Improvements/Updates since Project Readiness Presentation

A. PSSC 1 - Data Communication - 80%

Both sensor hubs are running and processing data. The machine learning Python scripts are running. Data is being sent and received to/from the

XBee Modules. The rover is communicating with a local server in order to control it. The sensor nodes are set-up to send biometric data to the sensor hubs.

B. *PSSC 2 - 2D Map - 65%*
   HoloLens has map and compass display, LMCC contains HTML/CSS GPS prototype, RealSense D435i camera maps a 2-dimensional space and calculates distances, GPS breakouts on Sensor Hubs provide coordinate data.
C. *PSSC 3 - Caution/Warning System - 75%*
   Sensor Hubs can detect abnormal biomedical conditions via machine learning from Sensor Node. The HoloLens GUI will be able to display this warning.
D. *PSSC 4 - Rover Commanding - 90%*
   Rover is commandable from a local server, which allows keystrokes to allow control from a user. Built and controllable from a Python-scripted server with a Raspberry Pi and servo motors.
E. *PSSC 5 - Fully Developed GUI - 90%*
   HoloLens has a well-developed graphical user interface. User interaction and menu selections.

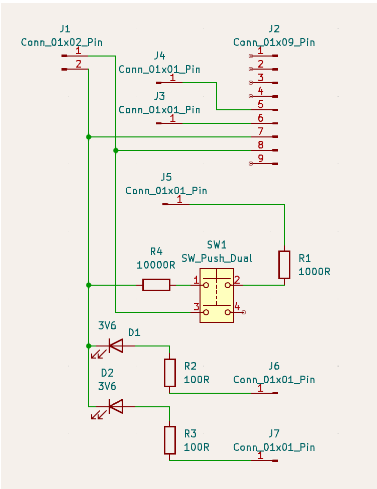
VII. Ethical Considerations

The HoloLens will not endanger the user with conditions such as blindness, hearing loss and no damage will come to the user using any component like rover and sensors. The privacy of the user will be top priority meaning when a user has the realsense camera it will not record anything about the user. The LMCC on hololens will not show a user bioinformatics with anyone useless the person wants to.

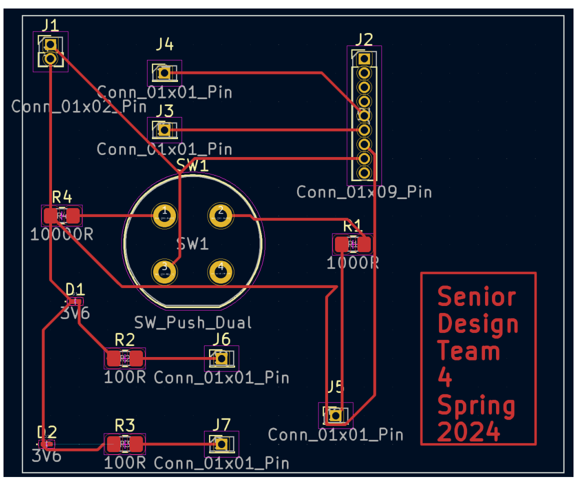
VIII. PCB - *Reasoning, Functionality, & Performance*

The PCB functionality is to provide GPS data to the sensor hubs, as well as provide an on-off button to test the connection between sensor hubs. The prototype for this design contains loose wires that easily come apart. The idea for creating a PCB to replace it is to provide safety measures and also improve functionality. The connector pins are intended for connection to the Raspberry PI, and the 9-pin connector is for the GPS breakout.

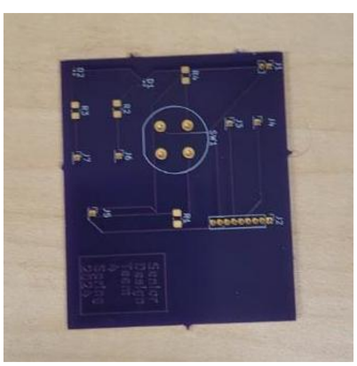
Upon receiving the PCB, we found the dimensions of the design were too small for the necessary components. It will need to be redesigned to fit the requirements. In addition, the resistors are embedded into the PCB, and might be the incorrect part number. The switch as well is a different design than the prototype's switch.

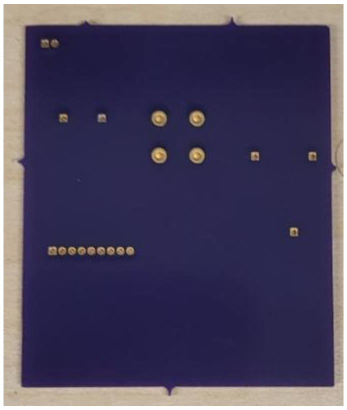


Circuit diagram for GPS breakout PCB.



KiCad schematic for GPS breakout PCB.





Received PCB.

IX. Machine Learning Data Collection, Training, and Prediction

A. Define the Problem
The problem is making a caution & warning system to detect abnormal or normal readings when an astronaut is doing an activity.

B. Data Collection
https://liveumb.sharepoint.com/:f:/r/sites/SD-F23-Team4/Shared%20Documents/General/Resources?csf=1&web=1&e=eOUj14

1	Accel_X	Accel_Y	Accel_Z	Body_Temp_F	Heart_rate	label
2	-0.08	0.01	9.93	91.74	0	resting
3	-0.07	0.02	9.97	91.74	0	resting
4	-0.06	0.02	9.94	91.85	0	resting
5	-0.08	0.02	9.95	91.85	0	resting
6	-0.06	0.01	9.96	91.74	0	resting
7	-0.07	0.01	9.96	91.74	0	resting
8	-0.07	0.01	9.93	91.74	0	resting
9	-0.07	0.02	9.97	91.74	0	resting
10	-0.06	0.02	9.94	91.74	0	resting
11	-0.07	0.02	9.96	91.74	0	resting
12	-0.07	0.01	9.96	91.74	0	resting
13	-0.07	0.01	9.95	91.74	0	resting
14	-0.07	0.02	9.94	91.74	0	resting
15	-0.07	0.02	9.97	91.74	0	resting
16	-0.06	0.01	9.94	91.74	0	resting

Figure 1. Training data set old

1	Accel_X	Accel_Y	Accel_Z	Body_Temp_F	Heart_rate	label
2	1.22	-1.39	11.78	75.54	0	running
3	1.46	-0.72	10.6	75.54	0	running
4	0.79	0.21	6.87	75.43	43	running
5	-0.16	-0.11	6.34	75.54	43	running
6	1.06	0.69	11.33	75.54	43	running
7	2.11	-0.61	7	75.54	0	running
8	-1.5	-1.28	9.65	75.54	0	running
9	-0.98	-1.57	8.04	75.54	165	running
10	0.98	1.18	10.14	75.54	164	running

Figure 2. Testing data set old

	A	B	C
1	accelnomoter x	accelrnometer y	status
2	0.05	1	normal
3	0.04	3	abnormal
4	0.04	4	abnormal
5	0.07	6	normal
6	0.1	7	abnormal
7	0.12	8	normal

Figure 3. New training set

	A	B	C
1	accelnomoter x	accelrnometer y	status
2	0.05	-0.5	normal
3	0.04	0.03	abnormal
4	0.04	0.2	abnormal
5	0.07	0.1	normal
6	0.1	0.4	abnormal
7	0.12	0.03	normal

Figure 4. New testing set

C. Feature Selection/Engineering - The activity is broken down into 3 categories running, walking and resting. If there are 86 BPM on resting is bad but if you get 96 BPM while running is good.

D. Training

Python was the programming language used along with Keras and Tensorflow libraries. Next, the sensor node was the hardware used to collect the data. Using the previous team 4 idea, meaning model.fit(), the number of iterations is 100 because of the time it takes for the machine to learn from patterns and if the number of iterations were set to 1000 then it would take longer to go through the training data to learn from.

E. Prediction

So our team basically used the same approach to the previous team's 4 ideas which is to use the predict function in python and display the loss and accuracy of the model.

```
Sending chunk 74 of 121 to 0013A20041F21DCD...
 1/29 [>.............................] - ETA: 1s - loss: 2.8405e-05 - accuracy:
14/29 [=============>................] - ETA: 0s - loss: 1.3266e-05 - accuracy:
28/29 [===========================>..] - ETA: 0s - loss: 2.7440e-04 - accuracy:
29/29 [==============================] - 0s 4ms/step - loss: 2.7381e-04 - accura
cy: 1.0000
Test Loss: 0.00027380927349440753
Test Accuracy: 1.0
```

Figure 5. PSSC #3 caution and warning system

X. Design Validation

A. PSSC 1: Data Communication

V&V: How reliable is the file sent to the hololens 1 from sensor hub

How fast it went (bits/sec)

1. IEEE

IEEE Standard for Biometric Open Protocol

IEEE 2010-2015

Identity assertion, role gathering, multilevel access control, assurance, and auditing are provided by the Biometric Open Protocol Standard (BOPS). The BOPS implementation includes software running on a client device (smartphone or mobile device), a trusted BOPS server, and an intrusion detection system. The BOPS implementation includes software running on a client device (smartphone or mobile device), a trusted BOPS server, and an intrusion detection system. The BOPS implementation allows pluggable components to replace existing components functionality, accepting integration into current operating environments in a short period of time.

2. ISO

ISO/IEC 20922:2016

Information Technology - Message Queuing Telemetry Transport (MQTT) V3.1.1

ISO/IEC 20922:2016 is a Client Server publish/subscribe messaging transport protocol. It is lightweight, open, simple, and designed so as to be easy to implement. These characteristics make it ideal for use in many situations, including constrained environments such as for communication in Machine to Machine (M2M) and Internet of Things (IoT) contexts where a small code footprint is required and/or network bandwidth is at a premium.

B. PSSC 2: 2D Map

V&V: Using Journey logistics (testing gps2.py on sensor hubs)

1. **IEEE**

   IEEE P1952: STANDARD FOR RESILIENT POSITIONING, NAVIGATION AND TIMING (PNT) USER EQUIPMENT

   Position, Navigation, and Time (PNT) services are essential for many critical applications across different sectors. Critical infrastructure, including communications, financial services, emergency services, power utilities, and transportation, all rely on PNT information to provide their services to users around the world. Threats, whether natural, accidental, or deliberate, can disrupt the delivery of PNT information and cause harm to individuals, businesses, and nations.

2. **ISO**
   ISO 24245:2023
   Space systems Global navigation satellite system (GNSS) receiver class codes
   This document specifies class codes to classify global navigation satellite system (GNSS) receivers. The class codes represent how signals transmitted from radionavigation satellites are processed. This document applies to all types of GNSS receiver devices.

C. **PSSC 3: Caution/Warning System**
   V&V: Write python test code - display on Hololens and LMCC
   - Test how fast data transferred and warning is display(bits/sec)

1. **IEEE**
   IEEE/ISO/IEC 11073-10201-2004
   ISO/IEEE International Standard for Health Informatics - Point-of-care medical device communication - Part 10201: Domain information model
   Within the context of the ISO/IEEE 11073 family of standards for point-of-care (POC) medical device communication (MDC), this standard provides an abstract object-oriented domain information model that specifies the structure of exchanged information, as well as the events and services that are supported by each object. All elements are specified using abstract syntax (ASN.1) and may be applied to many different implementation technologies, transfer syntaxes, and application service models.

2. **ISO**
   ISO/IEEE 11073-10407:2022
   Health informatics Device interoperability Part 10407: Personal health device communication Device specialization Blood pressure monitor
   Within the context of the ISO/IEEE 11073 family of standards for device communication, this document establishes a normative definition of communication between personal telehealth blood

pressure monitor devices and compute engines (e.g., cell phones, personal computers, personal health appliances, and set top boxes) in a manner that enables plug-and-play interoperability.

D. **PSSC 4: Rover Commanding**

V&V: Send commands from LMCC server to Rover with strings

- Confirm Rover client is receiving the commands

1. **IEEE**

   IEEE 1609.3-2020

   IEEE Standard for Wireless Access in Vehicular Environments (WAVE)--Networking Services

   Services to WAVE devices and systems are provided. Layer 3 and layer 4 of the open system interconnect (OSI) model and the Internet Protocol (IP), User Datagram Protocol (UDP), and Transmission Control Protocol (TCP) elements of the Internet model are represented. Management and data services within WAVE devices are provided.

2. **ISO**

   ISO/IEC 29341-8-10:2008

   ISO/IEC 29341-8-10:2008(E) enables the remote control, monitoring and configuration of a Dynamic Host Configuration Protocol (DHCP) and DNS server, serving a residential LAN. The series of ISO/IEC 29341 publications defines an architecture for pervasive peer-to-peer network connectivity of intelligent appliances, wireless devices and PCs. It is designed to bring easy to use, flexible, standards-based connectivity to ad-hoc or unmanaged networks whether in the home, in a small business, public spaces or attached to the Internet.

E. **PSSC 5: Fully Developed GUI**

V&V: How fast HMD and LMCC connection in bits/sec

1. **IEEE**
   IEEE 1471-2000
   IEEE Recommended Practice for Architectural Description for Software-Intensive Systems
   " This recommended practice addresses the activities of the creation, analysis, and sustainment of architectures of software-intensive systems, and the recording of such architectures in terms of architectural descriptions . A conceptual framework for architectural description is established. The content of an architectural description is defined. Annexes provide the rationale for key concepts and terminology, the relationships to other standards, and examples of usage. "

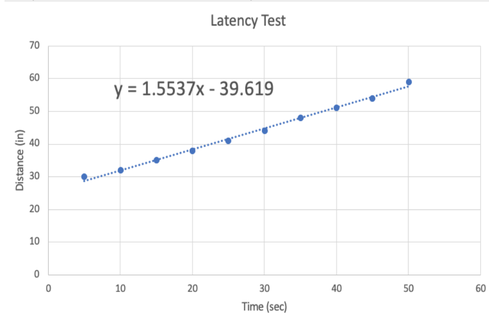
2. **ISO**
   ISO/IEC TS 20071-15:2017 provides guidance on various aspects of the user interface of applications that scan visual information that are used directly by humans, including: initiating the scanning application;
   - setting user's preferences and configuring the scanning application;
   - identifying the types of information currently of interest to the user;
   - locating visual objects of interest to the user; creating a static image via scanning the visual object;
   - identifying the information content provided by the visual object;
   - processing scanned information and outputting the results to the user.

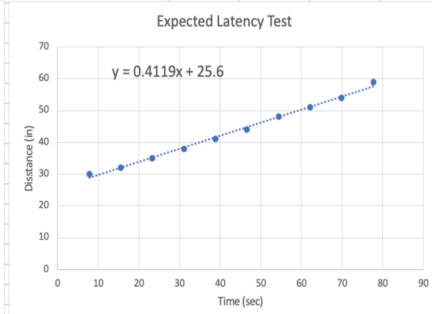
XI. Validation Data

A. PSSC 1 - Data Communication

Example data set to test latency from Hub 1 to Hub 2 to HoloLens. Testing distance from Hub 1 to HoloLens against the time it takes for the csv data file to populate the HoloLens Unity project

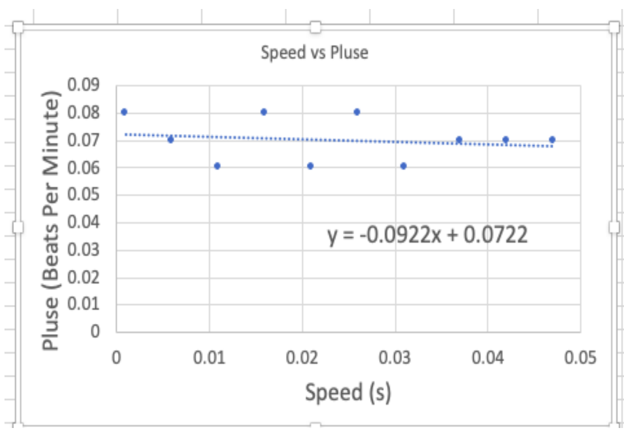
Hub1 to Hub2 to HoloLens (seconds)	Distance between Hub1 and HoloLens (in)	Expected Latency
30	5	7.7685
32	10	15.537
35	15	23.3055
38	20	31.074
41	25	38.8425
44	30	46.611
48	35	54.3795
51	40	62.148
54	45	69.9165
59	50	77.685

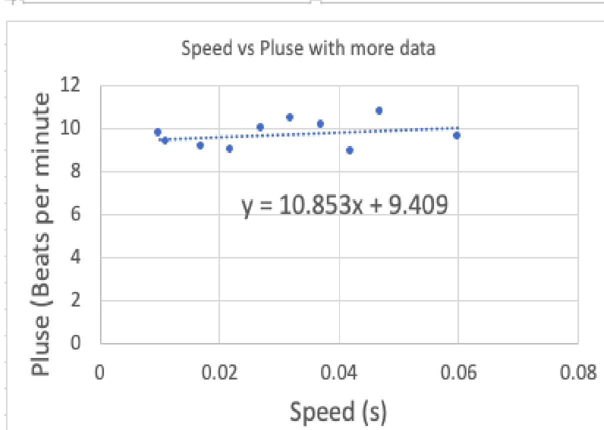




B. *PSSC 3 - Caution/Warning System*

The first graph is for running that uses pulse measurements from the pulse sensor. But the graph on the bottom is the machine learning prediction that has more data to the set.

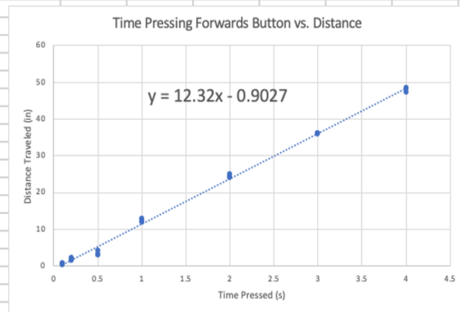


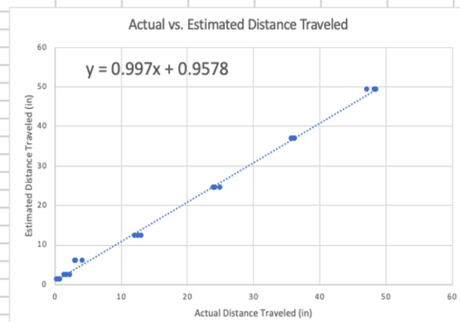


C. *PSSC 4 - Rover Commanding*

Testing the time pressed on the Rover controls vs. the distance it travels. Moves at 12.32 inches per second. Compared to estimated distance, the actual data is very close and has a linear relationship.

Distance Traveled (in)	Time Pressed (s)	Expected Distance Traveled based on 12.32 in/sec
0.8	0.1	1.232
0.5	0.1	1.232
0.3	0.1	1.232
1.8	0.2	2.464
2.3	0.2	2.464
1.4	0.2	2.464
3	0.5	6.16
3.2	0.5	6.16
4.1	0.5	6.16
13	1	12.32
12.5	1	12.32
12	1	12.32
24	2	24.64
24.2	2	24.64
25	2	24.64
36.1	3	36.96
35.8	3	36.96
36.2	3	36.96
48.3	4	49.28
47.2	4	49.28
48.5	4	49.28





XII. Deliverables

A. *Description of Final Product*

ARIS has many interconnected parts that need to be integrated together into a complete and functional system. This product will allow an astronaut to explore space terrain, collect data, and communicate with a central control console.

B. *Summary of Live Demo*

The live demonstration shows each component of ARIS. The sensor hubs, sensor nodes, rover, HoloLens, RealSense camera, and LMCC are displayed. Their functionalities are also illustrated, and samples of the machine learning algorithms are presented.

C. *Link to Live Demo:* https://youtu.be/8anOoq1N8Tc?feature=shared

XIII. Acknowledgement

Thank you to Tomas Materdey for acting as CM/TM for this project and hosting weekly meetings to keep the project on track. Thank you to the CS team for initiating and developing important aspects of this project, such as the LMCC server/webpage, HMD GUI, and RealSense Camera.

XIV. References

[1] NASA Spacesuit User Interface Technologies for Students (2023-2024)
[2] Augmented Reality Informational System (ARIS) for Astronauts (2022-2023)

- Written and Designed by Oussama Ourich (EE-Co-Lead), Felice Gabardi (EE), Kalvin Fonseca (EE), Maryam Aminu Mukhtar (CE-Team Lead), CM/TM - Tomas Materdey
- Video link: https://youtu.be/KLRMTx8ZErM?si=t8Z8bYhgAFOT8Mi4
- GitHub Repository: https://github.com/mamukhtar/Project-ARIS

[3] CLAWS Exit Pitch (2023)

- https://youtu.be/aLqUng8fB_o?si=k7Qbje0q9aRybdK6

[4] NASA SUITS Exit Pitch Presentations (2023)

- https://www.youtube.com/live/wddE3lTbnpQ?si=szTY3V2J50fs9Eqz

[5] “My experience at the NASA SUITS challenge…” by Ratnam, Avanash (2023)

- https://bootcamp.uxdesign.cc/my-experience-at-the-nasa-suits-challenge-as-a-ux-designer-16ea2c7c0b9d

[6] Magic leap 2 (hololens in the market)

- https://developer-docs.magicleap.cloud/docs/guides/developer-tools/app-sim/app-sim/
- https://developer-docs.magicleap.cloud/docs/guides/ml2-overview/index.html
- https://www.magicleap.com/developer

XV. Operation Manual

Use with caution. The sensor hubs both operate on Raspberry Pi’s. In order for each to run, a RealVNC server is needed to run the commands simultaneously. Run the scripts ‘Hub1.sh’ and ‘Hub2.sh’ to compile the necessary scripts. They will run one after the other and includes the machine learning scripts.

The rover runs by entering the main directory and deploying the ‘./start’ function in the terminal. This will boot up the local server, which will allow for the rover to be moved with a keyboard. Open up the local server to access the commands.

The HoloLens requires Unity for development and deployment. A local host is needed for the HoloLens to transmit and collect data. The LMCC is a HTML/CSS website script, with the development of a task initiator and map display. All documentation is provided in the programming directory.