A Wireless Wearable Solution for Monitoring and Analyzing Stroke Recovery

Josh Duha, Emma Tholl, Julien Campbell, Alexis Hamrak, Rafal Dzwonek, Daphne Hong
Sponsor Representative: Dr. Svetlana Yanushkevich, Academic Supervisor : Dr. Abraham Fapojuwo

A wireless, wearable system for monitoring movement in recovering stroke patients over 72-hours. This system provides caregivers with data and visualizations to help them understand if the patient's stroke resulted in a loss of movement on either side of the body. The system uses two small accelerometers worn on both wrists to continuously stream X, Y, and Z data. Data is then transferred to the Raspberry Pi hub and displayed on two applications. The first application displays live data as it is collected from the two sensors, whereas the second application displays an analytic summary of the data collected over the past 24 hours.

Introduction

Strokes commonly cause numbness or weakness on one side of the body [1]. Stroke patients recover lost strength with the help of physiotherapy while caregivers clinically monitor for asymmetry. An issue with this process is that patient movement is only monitored while someone is physically present. The quality of tracking patient progress would be significantly improved if the healthcare team was able to continuously monitor movement for up to 72 hours after they experience a stroke [2].

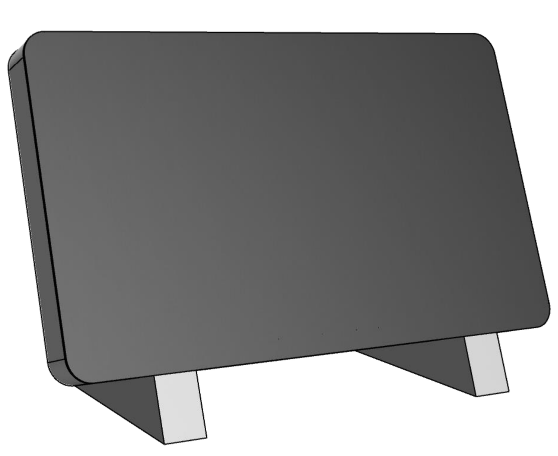
Hardware

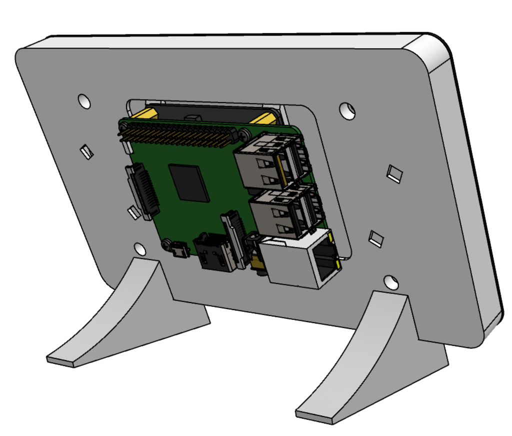
Sensors – Two MetaMotionS Accelerometers were used since they could meet the following constraints:

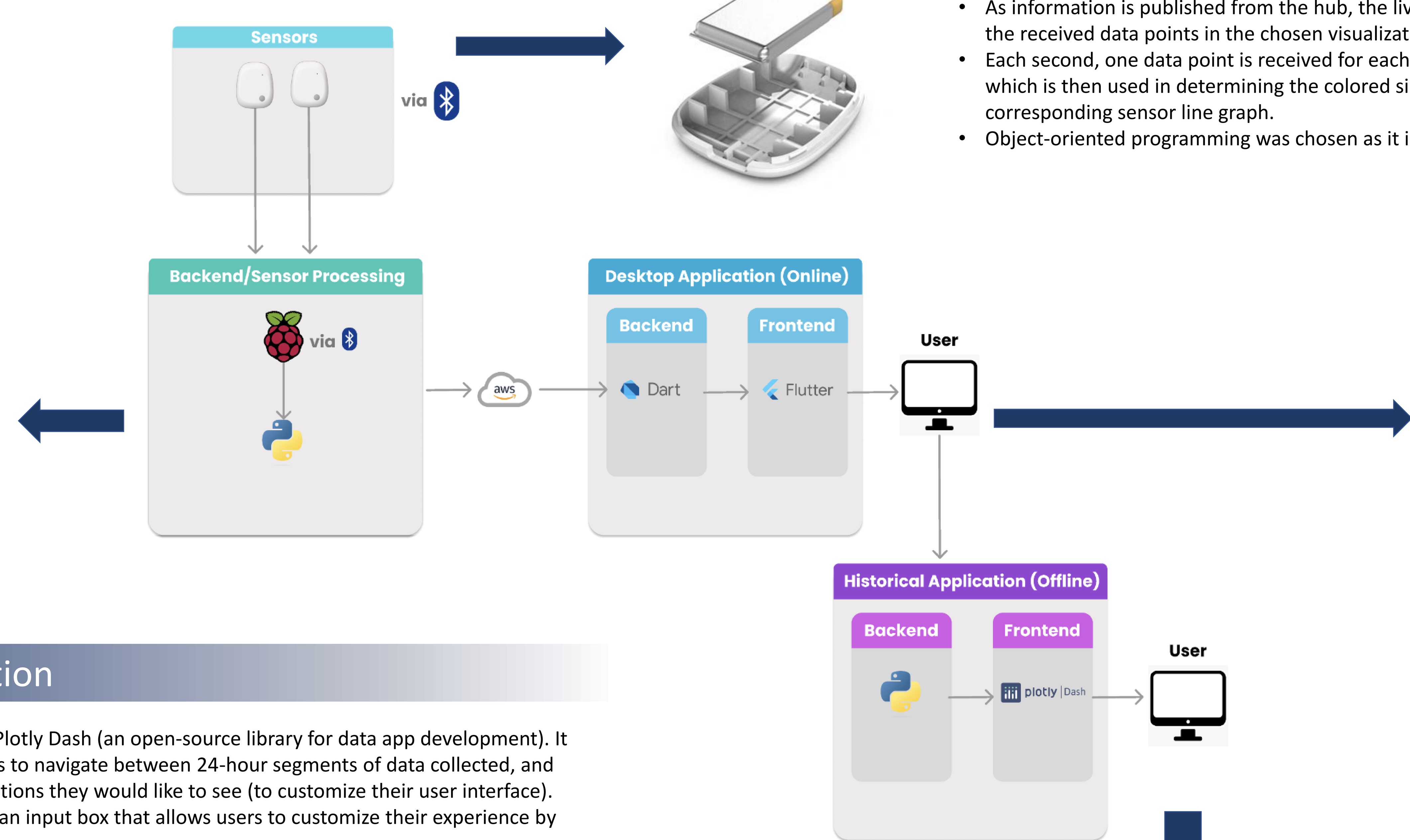
- Must last from at least 7am to 10pm (15h) on a single charge.
- Must have an extensive enough API to retrieve accelerometer data in the form of X, Y, Z vectors in floating point format.
- Must maintain connection across a typical hospital room (10m range).
- Must be readily available enough to replace in case of damage (shipping < 2 weeks).

Hub – A Raspberry Pi was used since it could meet the following constraints:

- Maintains simultaneous Bluetooth connections with up to 4 sensors. This makes the system expandible from the current 2 sensor setup. A third sensor was connected as a proof of concept that more sensors can be accommodated.
- Runs first level of data parsing and packaging of sensor data into JSONs.
- Publishes data to desktop via serial Bluetooth dongle instead of AWS.
- Must be small (capable of fitting on bedside table), powerful (able to run an operating system that supports our backend development), and easy to sanitize (minimal surfaces and crevices for microbes to live on).







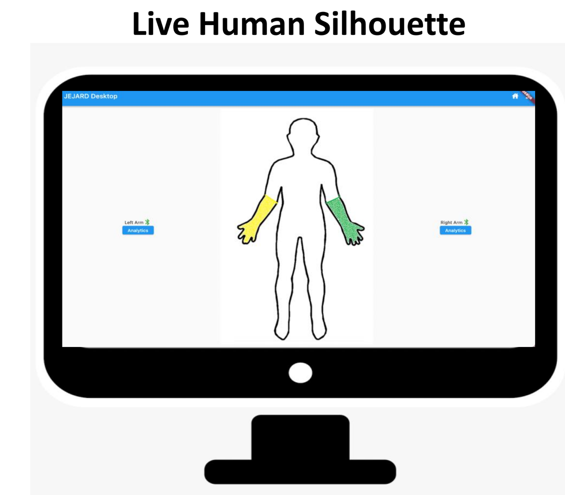
Software – Live Application

- Utilizes Flutter (frontend) and Dart (backend) (open-source UI SDK produced by Google [7]).
- Data Visualization 1: Live graph displaying the x-, y-, and z-accelerometer data for each sensor using Syncfusion library [8].
- Data Visualization 2: Live color-coded silhouette representing the movement of each limb. The color combinations were manually coded using the vector magnitude equation below.

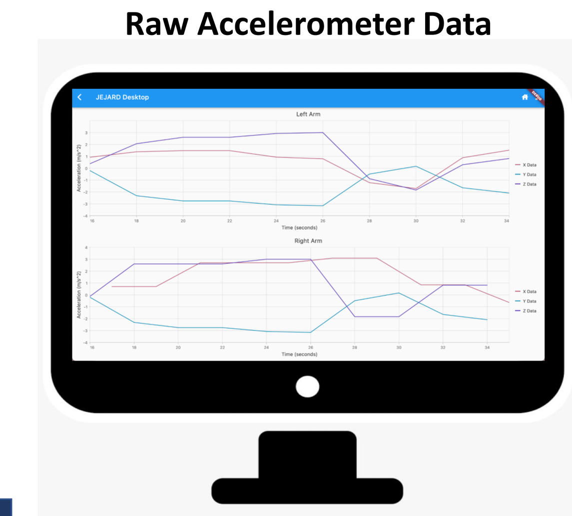
$$v = \sqrt{x^2 + y^2 + z^2}$$

- Each of the accelerometer parameters (x, y, z) had to be considered when determining what category of movement the limb should be considered under. Calculating the magnitude allowed us to determine the overall limb movement.
- Amazon Web Services' (AWS) IoT Core [9] was integrated to receive the sensor data via cloud computing. It then writes the data into a csv file that the live application could use later.
- AWS was integrated into the live application using the MQTT protocol and the MQTT Client library [10] from Dart.
- When the application starts its run, it establishes a "subscriber" to an AWS data stream using a private key, protocol, and topic name.
- As information is published from the hub, the live application receives it in real-time and displays the received data points in the chosen visualizations.
- Each second, one data point is received for each sensor and the data is parsed into a sensor object, which is then used in determining the colored silhouette to display and plot that data point on its corresponding sensor line graph.
- Object-oriented programming was chosen as it is reusable, encapsulated, and abstract.

Live Human Silhouette



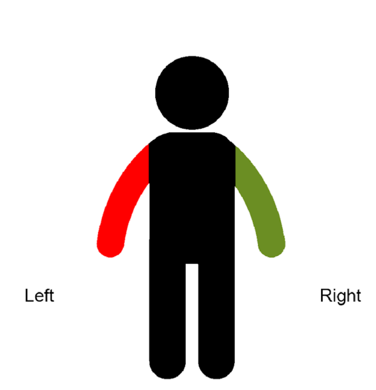
Raw Accelerometer Data



Data Analytics – Historical Application

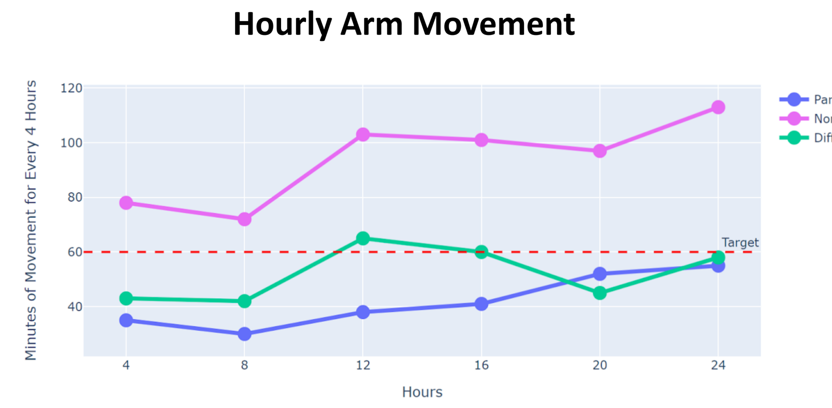
- The historical application was programmed in Python with Plotly Dash (an open-source library for data app development). It has a header that contains the name of the application, tabs to navigate between 24-hour segments of data collected, and checkboxes that allow users to select the subset of visualizations they would like to see (to customize their user interface).
- In the left pane, patient information is displayed, as well as an input box that allows users to customize their experience by modifying thresholds used in the application.
- The body of the application consists of four main visualizations. These include the following: a human silhouette that shows the severity of limb impairment, a time series graph that shows hourly arm movement, a scatter plot that shows the patient's use ratio, and a bar plot that shows the activity counts of the upper limbs.
- This application uses data collected by Washington University, retrieved from SimTK [3]. The 24-hour 30Hz raw accelerometer data was converted into activity counts using a Python package called Agcounts [4]. The data was then downsampled to 10-second epochs and sectioned into 4-hour intervals. Vector magnitudes of the activity counts were computed for the paretic and non-paretic limb use, activity counts, and use ratio calculations.

Severity of Arm Impairment



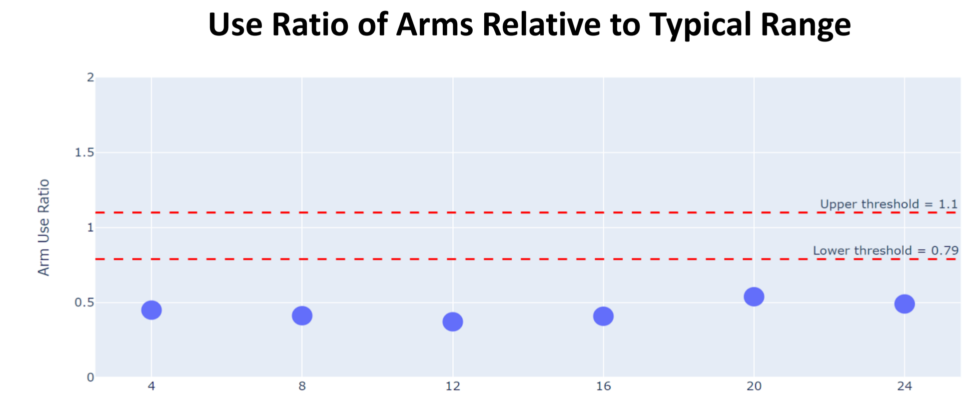
Human Silhouette - Has color-changing arms to represent the movement seen from each limb over 24 hours. Red represents little to no movement by the limb whereas green represents a lot of movement. The threshold that is being used to determine the color of the limb was determined through literature studies [5]. By creating this visualization, users can quickly understand if one arm has been moving less than the other, which may indicate loss of function/numbness following the patient's stroke.

Hourly Arm Movement



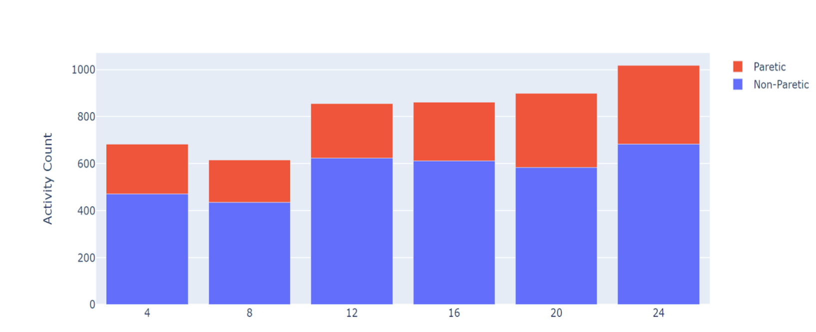
Time Series Graph - Shows the hourly arm movement. This graph has three lines representing the paretic arm movement (blue), non-paretic arm movement (purple), as well as the difference between the two arms (green). A red threshold line was also added to indicate how much movement the doctors would like to see over 4 hours. The x-axis of this graph represents the 24 hours of data collected, segmented into 4-hour chunks. On the y-axis, this graph displays the minutes of movement for every 4 hours. This graph helps doctors quickly understand how many minutes of movement each arm has produced as well as the difference between them.

Use Ratio of Arms Relative to Typical Range



Scatter Plot - Plots 24 hours of the use ratio between the upper extremities. The blue dots represent the ratio of paretic arm movement to non-paretic arm movement. Values close to one indicate symmetrical limb use, values less than one indicate greater use of the non-paretic limb, and values more than one indicate greater use of the paretic limb. It also includes an upper and lower threshold range to represent where the expected use ratio should be, which was decided based on research [6]. This graph will help doctors quickly understand the difference in use between limbs in comparison to the expected range.

Activity Count of Paretic and Non-Paretic Arms



Bar Graph - Demonstrates the number of activity counts from the paretic (red) and non-paretic (blue) limb over 24 hours. This graph helps users quickly understand the level of activity from each limb over 4-hour segments. Data collected from a patient without imbalance would show blue and red bars of the same height.

Informative Graphics: Allow healthcare providers to quickly and efficiently interpret how their patient's mobility was affected by their stroke.
Scalable: Patient data screen allows doctors to quickly pull up important information. This app can be scaled to include multiple patients.
Clear + Concise: Easily understandable by patients, doctors, and nurses that makes the application useful in a busy hospital setting.

References:
[1] "Treat and recover from stroke," *Centers for Disease Control and Prevention*, 05-Apr-2022. [Online]. Available: https://www.cdc.gov/stroke/treatments.htm. [Accessed: 11-Nov-2022].
[2] H. C. Persson, M. Parziali, A. Danielsson, and K. S. Sunnerhagen, "Outcome and upper extremity function within 72 hours after first occasion of stroke in an unselected population at a stroke unit. A part of the SALGOT study - BMC neurology," *BioMed Central*, 29-Dec-2012. [Online]. Available: https://bmcneurol.biomedcentral.com/articles/10.1186/1471-2377-12-162. [Accessed: 11-Nov-2022].
[3] C. E. Lang, K. J. Waddell, J. Barth, C. L. Holleran, M. J. Strube, and M. D. Bland, "Upper limb performance in daily life approaches plateau around three to six weeks post-stroke," *Neurorehabilitation and Neural Repair*, vol. 35, no. 10, pp. 903–914, 12-Sep-2021, doi: https://doi.org/10.1177/15459683211041302.
[4] LLC, "agcounts," *PyPI*, 15-Dec-2022. [Online]. Available: https://pypi.org/project/agcounts/. [Accessed: 03-Mar-2023]
[5] M. Wallich, "Decision Support Using Accelerometry and Machine Learning," University of Calgary, Calgary, Alberta, Canada, 2022.
[6] S. Zhang, A. V. Rowlands, P. Murray, and T. L. Hurst, "Physical activity classification using the genea wrist-worn accelerometer," *Medicine and Science in Sports and Exercise*, vol. 44, no. 4, pp. 742– 748, 01-Apr-2012, doi: https://doi.org/10.1249/mss.0b013e31823bf95c.
[7] Flutter. Accessed: March 19, 2023. [Online]. Available: https://flutter.dev/
[8] "Syncfusion_flutter_charts: Flutter Package," *Dart packages*, 15-Mar-2023. [Online]. Available: https://pub.dev/packages/syncfusion_flutter_charts. [Accessed: 19-Mar-2023].
[9] AWS IoT-Core. Accessed: March 19, 2023. [Online]. Available: https://aws.amazon.com/iot-core/?nc=sn&loc=0
[10] "Mqtt_client: Dart Package," *Dart packages*, 05-Mar-2023. [Online]. Available: https://pub.dev/packages/mqtt_client. [Accessed: 19-Mar-2023].
[11] Dr. Mohammed Almekhlafi, Craig Doram, Cumming School of Medicine, Foothills Hosptial

Biometric Technologies Laboratory

SCHULICH School of Engineering