

Diego and Ankur  
EE 250  
05/05/2023

We developed an Internet of Things (IoT) system that measures distance using a Grove Ultrasonic Ranger and plots the data over time. The system also calculates velocity and acceleration from the distance measurements and publishes all three data types to a public remote MQTT broker. The system utilizes a Raspberry Pi connected to the Grove sensors via an I2C bus and requires the GrovePi library to interface with the sensors and the Paho MQTT library to publish the data. To facilitate publishing, the system converts the distance, velocity, and acceleration data into JSON format.

The system starts by setting up the I2C bus, connecting the Ultrasonic Ranger and Grove Button to the digital pins on the Raspberry Pi, and initializing the lower and upper distance thresholds. It then creates empty lists to store the distance and time data, assigns a garbage value to "value\_to\_remove," and initializes "garbage\_indices" as an empty list.

The system enters an infinite loop that continuously reads the distance data from the Ultrasonic Ranger, checking for the presence of the garbage value in the distance list, and removing it along with the corresponding time value. The garbage value is set based on a value that is collected when the grove-pi sensor experiences an error. When the button is pressed, the system starts recording the distance data and the current time. When the button is pressed again, the system stops recording data and calculates the velocity and acceleration from the distance and time data. The data is then converted to JSON format and published to the remote MQTT broker.

To visualize the data, a subscriber receives the published data and uses the matplotlib library to create three subplots showing the distance, velocity, and acceleration data plotted against time. The plot is saved as a JPEG file.

One limitation of this system is that it only works with a single distance sensor, which limits its application. To address these limitations, the system could be modified to support multiple sensors and incorporate error correction algorithms. Additionally, the system requires a continuous connection to the MQTT broker, which may be problematic if the connection is lost. To address this limitation, the system could be modified to include a caching mechanism that stores the data until the connection is re-established.

This project would serve as a useful addition to sports medicine. These sensors can be used to track the performance of athletes, monitor their movements and identify any potential injuries or issues based on comparisons to previous data sets. A difficulty in this project could be the accurate calculation of velocity and acceleration, which requires precise timing and distance measurements. Overall, Diego and Ankur learned valuable lessons about working with IoT systems, such as the importance of error handling and the challenges of implementing real-time data processing.

Block Diagram:

