University of Gothenburg

DIT168

Project: Industrial IT and Embedded Systems

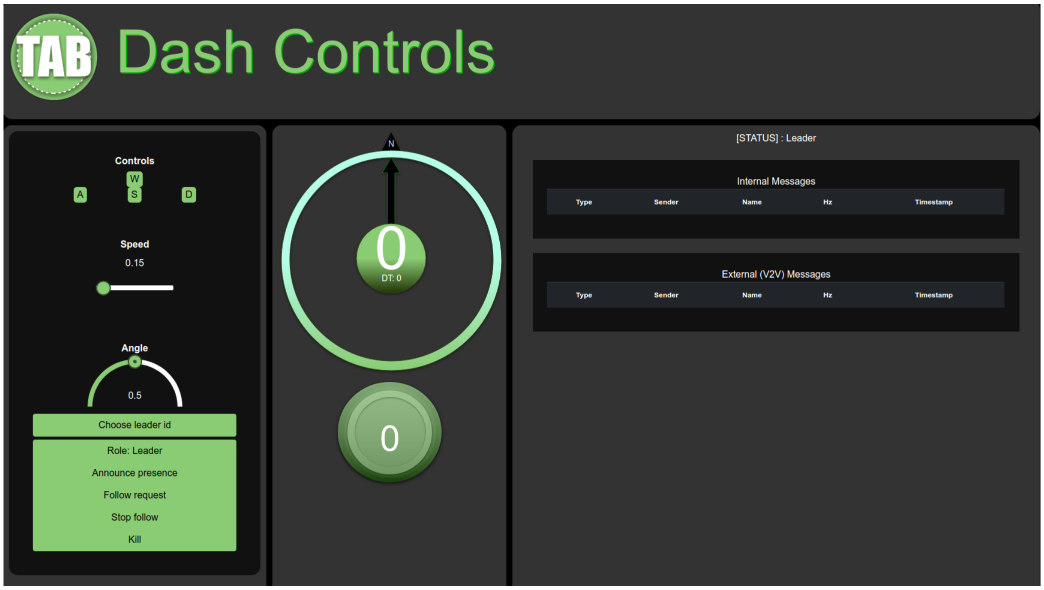
User Manual

Group 01
May 20th, 2018

Contents

1 Web Interface

The developed interface, pictured in Figure 1 can be divided to 3 sections; the remote control section, the sensor data visualization section and the message visualization section.



Internal Messages

Type	Sender	Name	Hz	Timestamp

External (V2V) Messages

Type	Sender	Name	Hz	Timestamp

Figure 1: Web Interface

The purpose of the interface, was to provided a nice and clean way of controlling the miniature vehicle. The development group though that the best way to achieve this is by using a single web page to provide both vehicle controls and data visualization.

2 Control Section

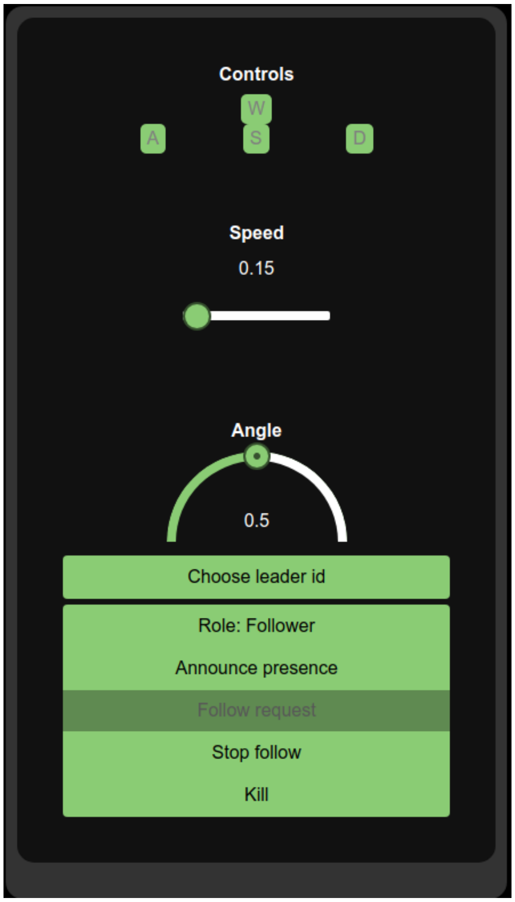


Figure 2: Control Section

The Control Section can be further divided into 2 separate sections or views.

The bottom section, discussed in section 2.1 is used to control what kind of messages are sent from the miniature car.

The top section, discussed in 2.2, is used to provide steering controls to the user.

This section of the manual is aimed to inform the user on how they can steer and control the car.

2.1 Message Controls

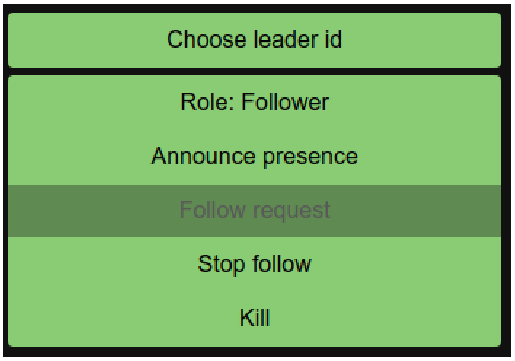
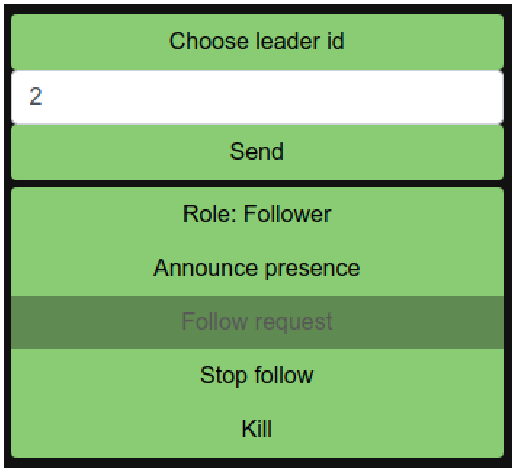


Figure 3: Message Controls



2

Figure 4: Message Controls

The small panel, pictured in Figures 3 & 4, is used to provide the user, or the Driver of the miniature car, with controls over the V2V-Protocol.

The first button, "Choose leader id" is a drop-down button that allows the Driver to select which other miniature vehicle to attempt to connect to. The above figures are used to depict the difference between the button's selection states. Once the button is a selected the user will see a message field to enter the other cars' IDs and a button to send the new leader ID to the car. For this feature to work the other cars must have had announced their presence. This is due to the miniature cars storing the vehicle IPs within a map, the ID being the key and IP the value. Without the AnnouncePresence being sent the UDP Senders, Receivers could not be established. Hence no V2V following.

The second button, in Figure 3 is used to swap between the miniature car's roles i.e. leader and follower. The current role is dispayed on the button itself and above the message visualization, seen in Figure 7. This button is significant, because the miniature car will not process some messages if it isn't in the correct role. For example, if the car is currently in Follower Mode the car will not execute LeaderStatus request if or when they are received.

The "Stop follow" and "Kill" button serve similar purposes. The Stop follow button, makes the miniature car send a StopFollow message disconnecting from the other car. After this a new connection can be established between the cars. The ”Kill” button takes it a step further and turns of the V2V-Protocol after sending StopFollow message.

The ”Announce presence” and ”Follow request” buttons are not as significant. At previous iterations these buttons were used to control the sending of AnnouncePresence and FollowRequest messages respectively. However, currently the sending of these messages has been automated. The buttons have been left for testing and verification purposes.

2.2 Steering Controls

The figure right, focuses on the top section of Figure 2. Moreover, it depicts the Remote Controller created for the project.

The Web Control provides the Driver control over 3 aspects; moving the car, changing car speed and altering turning angle.

The Driver can move the car by pressing one of the ”WASD” keys, pictured at the top of Figure 5. The choice for using ”WASD” was motivated by its widespread use in gaming. Thus, it was believed that by incorporating these keys the development team could improve the intuitiveness of the remote controller.

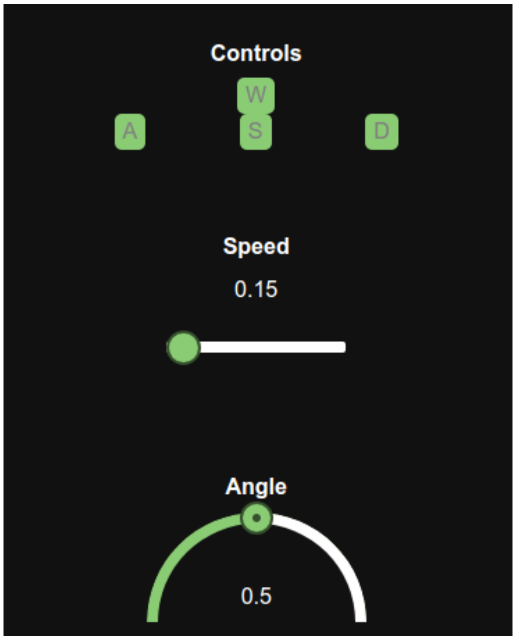


Figure 5: Steering Controls

To change the speed of the miniature car, the Driver can use the provided slider, with the current speed is displayed above the slider. The minimum speed is set to 0.15, due to it being the minimum required speed to physically move the car. The recommended speed range is between 0.15 and 0.20, as any higher could cause damage to the car’s axis and cause the wheels to spin off. A second, curved slider is used to change the car’s turning angle. The default is set to 0.5. These values correspond to percents.

3 Data Visualization

The Web Interface is also responsible for data visualization, which can be split up in two parts; Sensors and Messages.

3.1 Sensors

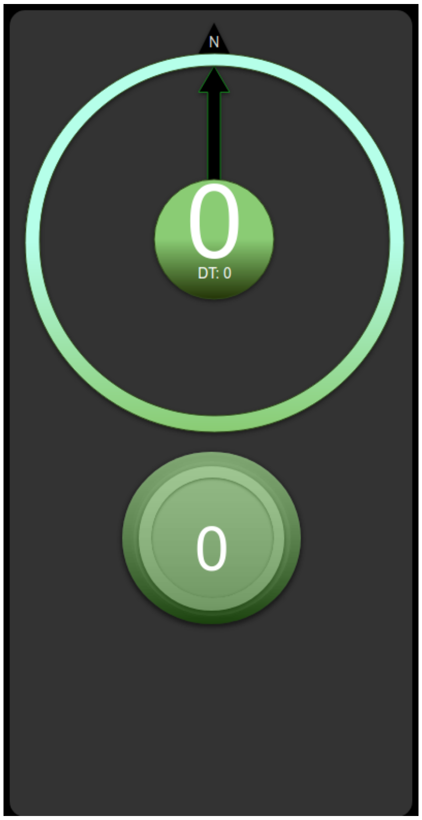


Figure 6: Sensor Section

The sensor section, pictured left, can be found in the middle of the interface as seen in Figure 1. This section is responsible for visualizing data from the IMU and the front Ultrasonic sensor.

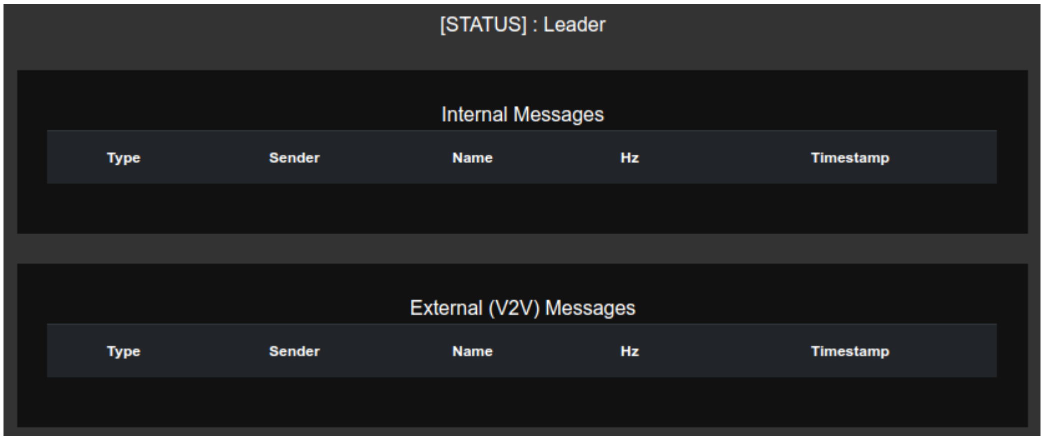
The larger circle, at the top of Figure 6, visualizes IMU readings. Specifically, 3 things. The first is the heading. These values are expressed as a compass with the black arrow pointing to the current car's heading. At the top of the circle a triangle with "N" meaning North. However, corresponds to the initial heading and is NOT affiliated with magnetic poles.

A second smaller circle can be seen within the compass. The larger number is used to depict the current speed of the car, whereas the smaller number under it expresses the distance travelled by the car, hence the "DT:" in front the smaller value.

The circle outside the compass is used to visualize Ultrasonic values. It changes colours from green to red. Red indicating that there is an object close to the car. The current distance to object is expressed as the number within the circle.

3.2 Messages

The messages being sent to and from the car are visualizesdin the right section on the interface. As mentioned in section 2.1, the current role of the car is depicted above the message visualization.



[STATUS] : Leader

Internal Messages

Type	Sender	Name	Hz	Timestamp

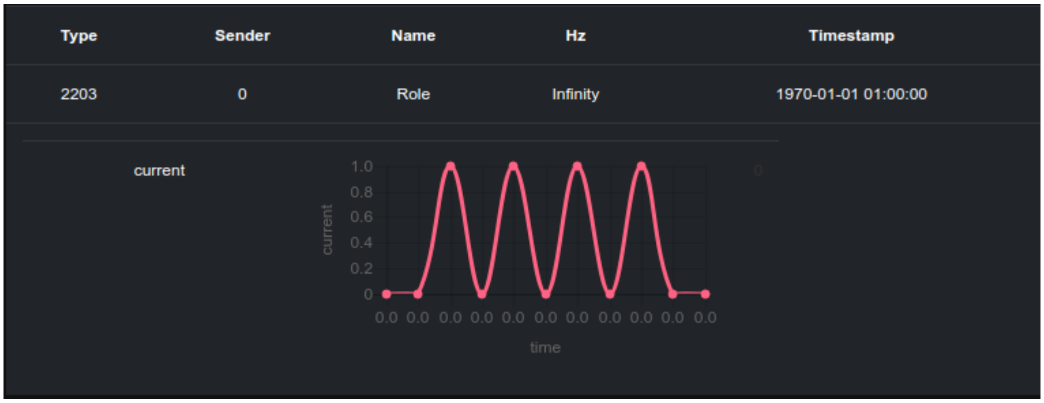
External (V2V) Messages

Type	Sender	Name	Hz	Timestamp

Figure 7: Message Visualization

The message visualization is divided into two sections. The first being "Internal Messages". Under this table messages from all the developed microservices, except the V2V, can be seen. In other words steering messages and sensor messages. The second table, "External (V2V) Messages", visualizes all the messages sent by V2V-Protocol.

Once a message is received it creates a graph, pictured in Figure 8 . The graph is used to better visualize the change in the value being sent, in this case the role, over time. Additionally, the message ID, name, sender as well as its frequency and timestamp.



Type	Sender	Name	Hz	Timestamp
2203	0	Role	Infinity	1970-01-01 01:00:00

Figure 8: Message Graph