WHITEPAPER

GitHub Integration and Code Repositories

Introduction

This guide explains how to use GitHub with Innoslate to modify and store live code repositories by following the code created and used in a lunar rover prototype. GitHub is a web-based platform that allows developers to host, manage, and collaborate on software development projects using version control, issue tracking, and other features. Below discusses each feature of the GitHub interface and how it was used to interact with the code repositories for the lunar rover prototype.

Use GitHub Integrated in Innoslate

The GitHub integration in Innoslate provides an interface to exchange information between various users in an Innoslate project. Three actions can be performed to conduct the exchange of information between the two tools. These actions are Issues, Commits, and Pull Requests.

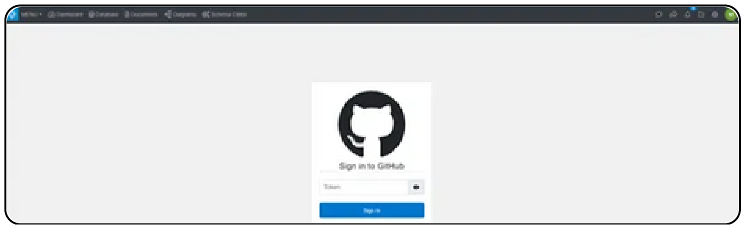
Issues are used to track work in GitHub. They can help the user organize and prioritize work that needs to be done within the current project. Issues can be differentiated using labels, assignees, and milestone relationships.

Commits are the history of a repository throughout development during the project. Commits tell a story through the progression of each repository in a project. Commits can be differentiated using assignees and timestamps.

Once a repository has been branched or forked, Pull Requests can be used to update and track its development in a project. The Pull Request is the final touch to tracing any Commits to their related Issues.

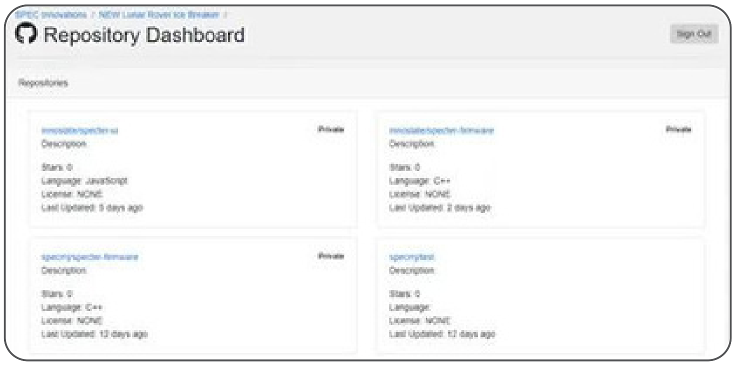
Login to GitHub View

A GitHub account, along with a specialized token, are required for login to access the GitHub View integrated with Innoslate.



GitHub View Login

When the unique token is used to access the GitHub View, any repositories and files attached to that token are opened and listed in the GitHub View dashboard. The figure below is a screenshot of the GitHub View maintained for the Lunar Rover Project.



innoslate/specter-ui Private
Description
Stars: 0
Language: JavaScript
License: NONE
Last Updated: 5 days ago

innoslate/specter-firmware Private
Description
Stars: 0
Language: C++
License: NONE
Last Updated: 2 days ago

specmj/specter-firmware Private
Description
Stars: 0
Language: C++
License: NONE
Last Updated: 12 days ago

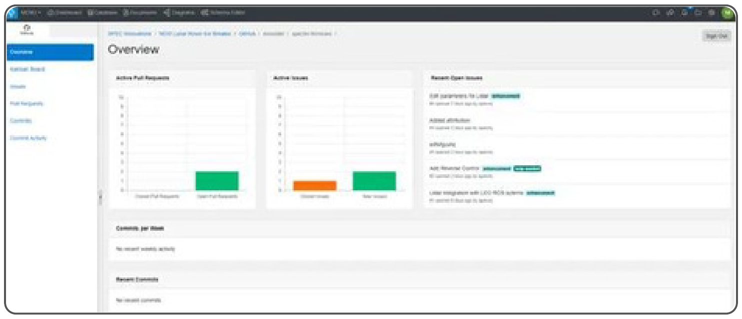
specmj/test
Description
Stars: 0
Language:
License: NONE
Last Updated: 12 days ago

GitHub Dashboard

There are four code repositories listed for the GitHub View in the Lunar Rover Project. Key features such as programming language, license, and the date the repository was last updated are listed for each repository in the GitHub Dashboard.

View the Repository Dashboard

Each repository also has its own dashboard, which displays recent activity and a list of the information exchanged between GitHub and Innoslate. The figure below is a screenshot of the SPECTER Firmware repository in the Lunar Rover Project.



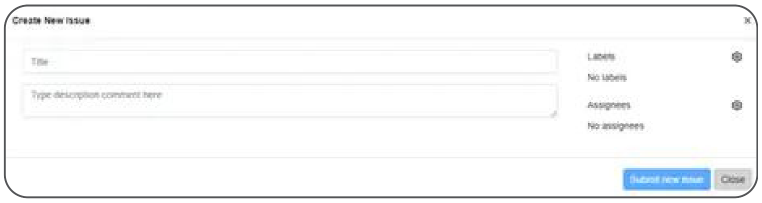
Repository Dashboard

The Innoslate Repository dashboard has widgets to display the following information:

- Number of active Pull Requests
- Number of active Issues
- Recently opened Issues
- Rate of Commits made per week
- Recent Commits

Create GitHub Issues

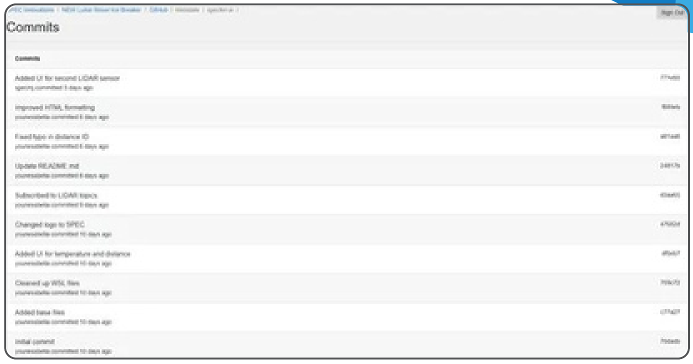
GitHub Issues can be created in each Innoslate repository by navigating to the 'Issues' tab on the left side panel. When creating a GitHub Issue, the following metadata is encouraged to differentiate the Issues: name, description, assignee, and labels. The figure below is a screenshot of the Issue Creation prompt. Once created, the GitHub Issue is stored in Innoslate and cloned to GitHub.



GitHub Issues Metadata

Create GitHub Commits

Each GitHub Commit in Innoslate is a clone of the Commit found in GitHub. Commits can be easily understood as edits made within a file or code in the repository. Metadata that can be added to the Issue in Innoslate includes a description and assignee. Use this information to track the progress made in the repository. A list of the Commits made to the SPECTER User Interface repository is on the right.



Commits

Added UI for second LIDAR sensor
specmj committed 3 days ago

Improved HTML formatting
younessbella committed 6 days ago

Fixed typo in distance ID
younessbella committed 6 days ago

Update README.md
younessbella committed 6 days ago

Subscribed to LIDAR topics
younessbella committed 9 days ago

Changed logo to SPEC
younessbella committed 10 days ago

Added UI for temperature and distance
younessbella committed 10 days ago

Cleaned up WSL files
younessbella committed 10 days ago

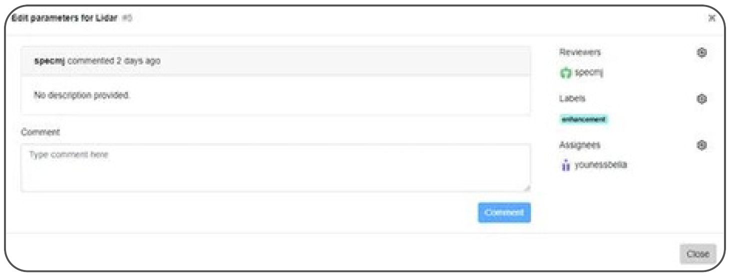
Added base files
younessbella committed 10 days ago

Initial commit
younessbella committed 10 days ago

GitHub Commits Metadata

Create GitHub Pull Requests

GitHub Pull Requests are the final submission of an Issue. Each Pull Request is also a clone from GitHub. Innoslate notifies the user of each Pull Request to allow them to check the GitHub platform for the changes to be approved. Within GitHub, the user can view all the Commits and Issues related to the Pull Request. If more information is required for the Pull Request, then comments can be added directly in Innoslate. Comments can be used to clarify any confusion in the Pull Request and provide feedback for the assignee or reviewer.



specmj commented 2 days ago

No description provided.

Reviewers: specmj

Labels: enhancement

Assignees: younessbella

GitHub Commits Metadata

Use GitHub Repositories

The two LEO Rover GitHub repositories for developing the prototype's firmware and user interface were cloned by the SPEC team to be modified for this Lunar Rover Project. The LEO Rover's firmware sends messages to components in the prototype using the connection between the processor and the component that performs the specified function. The user interface is used by operators to view and create commands for controlling the rover's actions and receiving diagnostics from the prototype's hardware.

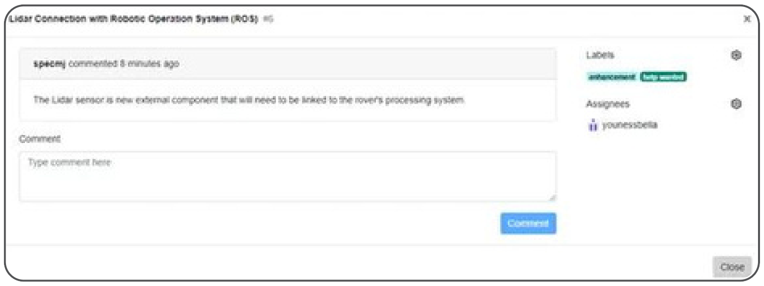
These two cloned repositories for the rover prototype are stored in Innoslate's GitHub Repository, and they contain all the code necessary to develop SPECTER's firmware and user interface. LEO Rover's default settings were kept for the firmware and user interface.

Modify the Firmware Repository

To gather data using SPECTER's lidar, modifications were made to the firmware to record the distances to obstacles from the rover body, as well as ambient temperatures. This information is used to provide data and diagnostics of the rover's surrounding area. The following sections step through creating Issues and submitting changes to develop an interface for streaming the above metrics directly to the rover operator.

Assign the Lidar to the Processing Port

First, an Issue was created to alert SPEC's software team of the new feature to be added to the user interface, including the lidar sensor in SPECTER's robot operating system (ROS). The figure below is a screenshot of this Issue.


Lidar Connection with Robotic Operation System (ROS) #5

specmj commented 8 minutes ago

The Lidar sensor is new external component that will need to be linked to the rover's processing system.

Comment

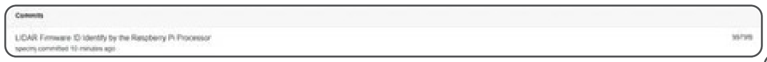
Type comment here

Labels: enhancement, help wanted

Assignees: younessbella

Lidar – ROS GitHub Issue

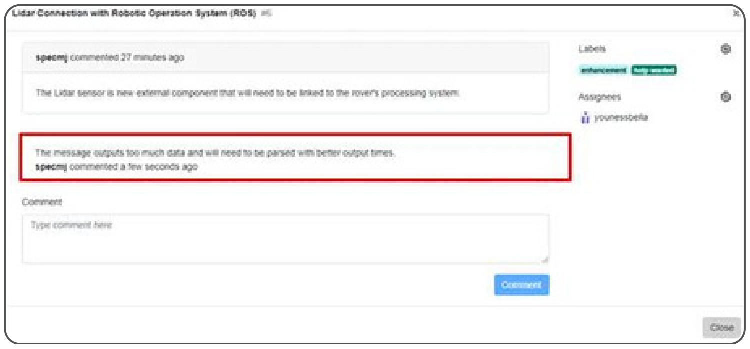
Once the Issue was published and cloned to the GitHub platform, it was assigned a SPEC software team member for the feature creation. Edits to the code were made in GitHub and submitted as Commits. These Commits exchanged the information from GitHub to Innoslate for publishing.


Commits

LIDAR Firmware ID identify by the Raspberry Pi Processor [illegible]
specmj committed 10 minutes ago

Lidar – ROS GitHub Commit

Once the Commit was made to allow SPECTER to communicate and send messages via the firmware, a reviewer from the SPEC software team examined the code written to verify it was ready for publishing. After reviewing the Commit, it was decided that the code needed continuing edits and would not be published yet. The figure below displays the reviewer's comment made to disapprove of the Commit.


Lidar Connection with Robotic Operation System (ROS) #6

specmj commented 27 minutes ago

The Lidar sensor is new external component that will need to be linked to the rover's processing system.

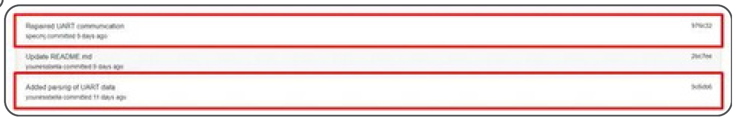
The message outputs too much data and will need to be parsed with better output times.
specmj commented a few seconds ago

Comment

Type comment here

Lidar – ROS GitHub Commit Reviewer Feedback

The original writer reviewed the reviewer's comments and continued editing the code to meet SPEC Innovation's software standards. These edits were also submitted as Commits. The figure below highlights the changes made.

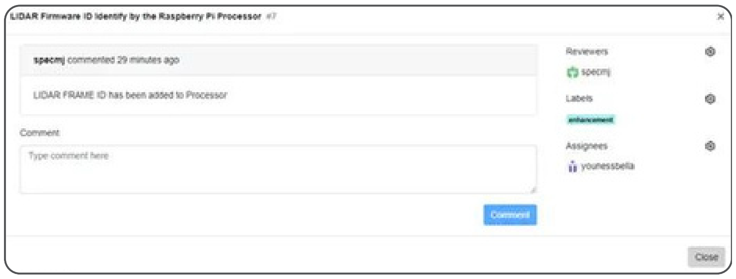

Repaired UART communication [illegible]
specmj committed 5 days ago

Update README.md [illegible]
younessbella committed 5 days ago

Added parsing of UART data [illegible]
younessbella committed 11 days ago

Lidar – ROS GitHub Commits

Once the code updates were complete, the reviewer read the new Commits and determined the code to be ready for publishing. The Commits were then submitted as a Pull Request, and the Issue was closed. The figure below is the final comment resolving the Issue.


LIDAR Firmware ID Identify by the Raspberry Pi Processor #7

specmj commented 29 minutes ago

LIDAR FRAME ID has been added to Processor

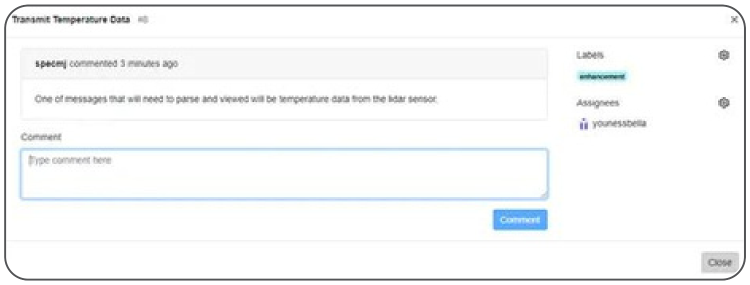
Comment

Type comment here

Lidar – ROS GitHub Pull Request

Transmit Temperature Data

An Issue was also created for developing an ambient temperature recording to be output from the Lidar sensor and displayed in the user interface. The figure below shows this GitHub Issue.



Transmit Temperature Data #8

specmj commented 3 minutes ago

One of messages that will need to parse and viewed will be temperature data from the lidar sensor.

Comment

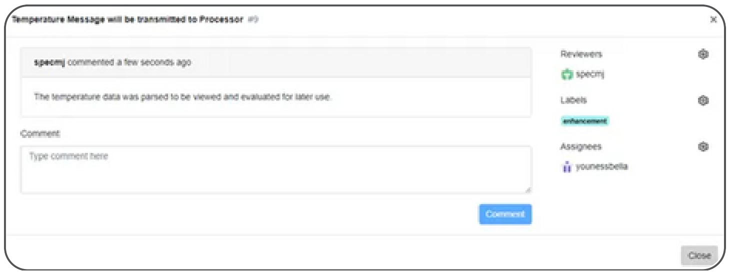
Type comment here

Labels: enhancement

Assignees: younessbella

Lidar – Temperature GitHub Issue

Once the Issue was published and cloned to the GitHub platform, it was assigned to a SPEC software team member for addressing. Any edits to the code were made on the GitHub platform and submitted as Commits to be exchanged with Innoslate. After the Commits were reviewed and approved, the edits were submitted as a Pull Request, and the Issue was closed.



Temperature Message will be transmitted to Processor #9

specmj commented a few seconds ago

The temperature data was parsed to be viewed and evaluated for later use.

Comment

Type comment here

Reviewers: specmj

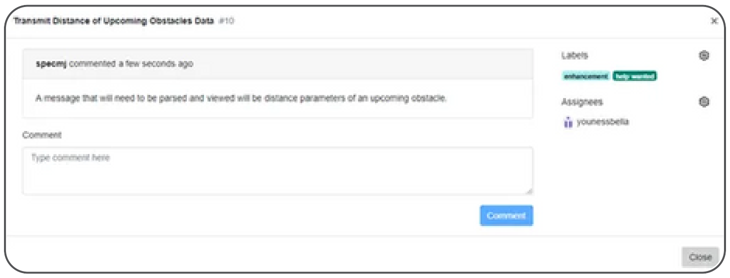
Labels: enhancement

Assignees: younessbella

Lidar – Temperature GitHub Pull Request

Transmit Distance Data

A final Issue was created to develop an output of distance data from the Lidar sensor, creating an obstacle avoidance warning on the user interface. The figure below is a screenshot of the Issue created for this feature.



Transmit Distance of Upcoming Obstacles Data #10

specmj commented a few seconds ago

A message that will need to be parsed and viewed will be distance parameters of an upcoming obstacle.

Comment

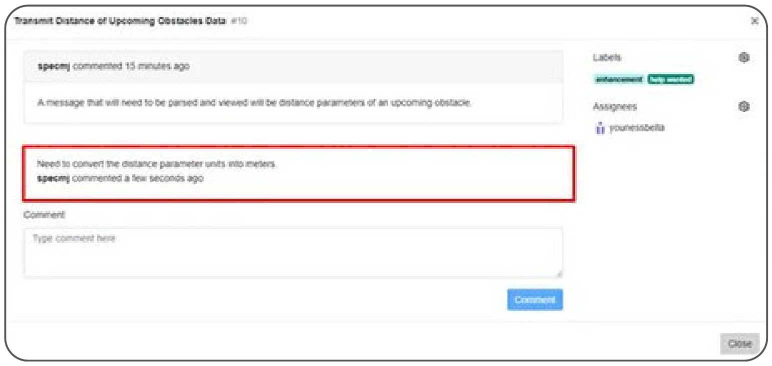
Type comment here

Labels: enhancement, help wanted

Assignees: younessbella

Lidar – Distance GitHub Issue

The same processes as mentioned with the other two issues were followed. However, the reviewer noticed the units used for distance parameters were incorrect and needed to be converted to meters. The figure below displays this comment made by the reviewer.


Transmit Distance of Upcoming Obstacles Data #10

specmj commented 15 minutes ago

A message that will need to be parsed and viewed will be distance parameters of an upcoming obstacle.

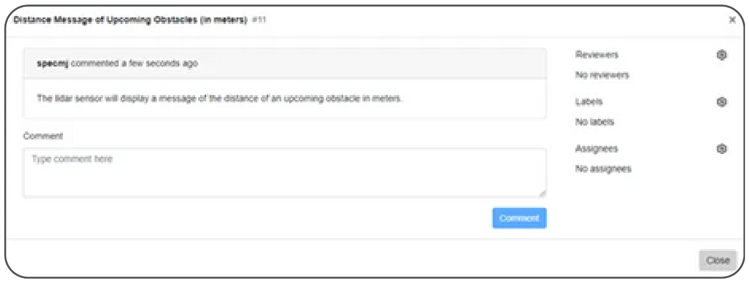
Need to convert the distance parameter units into meters.
specmj commented a few seconds ago

Labels: enhancement, help wanted

Assignees: younessbella

Lidar – Distance GitHub Commit Reviewer Feedback

Once the SPEC software team member made the edits and pushed the Commit, the Pull Request was submitted, and the Issue was closed.


Distance Message of Upcoming Obstacles (in meters) #11

specmj commented a few seconds ago

The lidar sensor will display a message of the distance of an upcoming obstacle in meters.

Reviewers: No reviewers

Labels: No labels

Assignees: No assignees

Lidar – Distance GitHub Pull Request

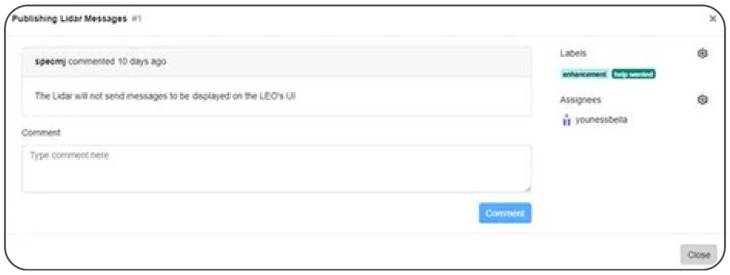
Modify the User Interface Repository

SPECTER's user interface allows the operator to view messages received from the prototype's processor. These messages and interface features include:

- Ambient temperature recording.
- Distance to obstacles from the rover body's front and rear display.
- Warning signal is triggered if a certain distance is reached from an obstacle.
- The user interface features provide diagnostic feedback to alert the operator of the rover's surrounding environment.

Display Lidar Data

An Issue was created to develop a display for the Lidar messages on the SPECTER's UI.


Publishing Lidar Messages #1

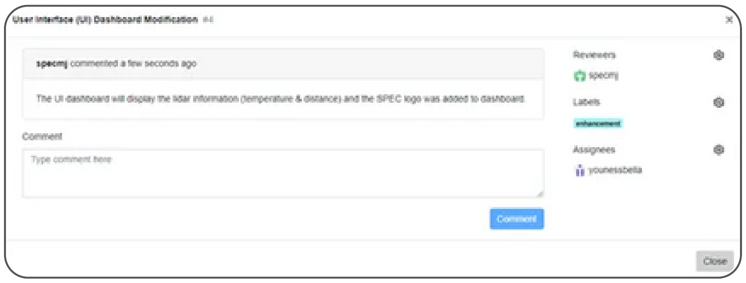
specmj commented 10 days ago

The Lidar will not send messages to be displayed on the LEO's UI

Labels: enhancement, help wanted

Assignees: younessbella

UI – Lidar GitHub Issue



User Interface (UI) Dashboard Modification #4

specmj commented a few seconds ago

The UI dashboard will display the lidar information (temperature & distance) and the SPEC logo was added to dashboard.

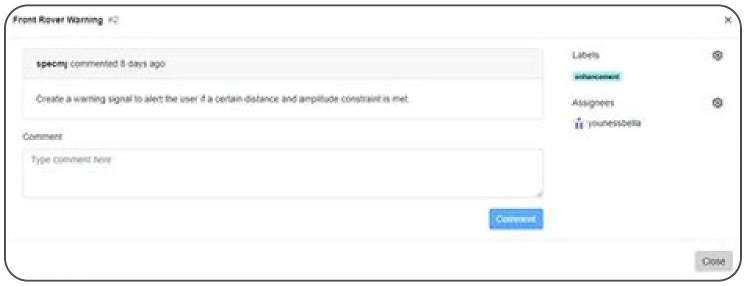
Comment

Type comment here

UI – Lidar GitHub Pull Request

Display Obstacle Warning Messages

Lastly, an Issue was also created to develop a warning alert display on the rover's UI for when SPECTER's body approaches an obstacle too closely. The figure below displays this Issue.



Front Rover Warning #2

specmj commented 8 days ago

Create a warning signal to alert the user if a certain distance and amplitude constraint is met.

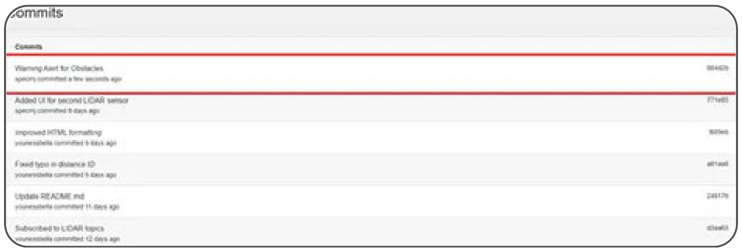
Comment

Type comment here

UI – Warning GitHub Issue

Once the Issue was published and cloned to the GitHub platform, it was assigned to a SPEC software team member for addressing. Any edits to the code were made in GitHub and then submitted as a Commits to be published on Innoslate.

The figure below displays a list of the Commits made in GitHub to create a warning message that displays on the user interface.



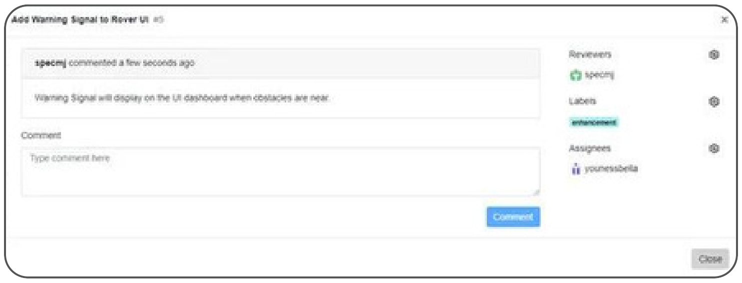
Commits

Commits	
Warning Alert for Obstacles specmj committed a few seconds ago	684429
Added UI for second LIDAR sensor specmj committed 8 days ago	771e85
Improved HTML formatting younessbella committed 9 days ago	f68feb
Fixed typo in distance ID younessbella committed 9 days ago	a81ae6
Update README.md younessbella committed 11 days ago	24817b
Subscribed to LIDAR topics younessbella committed 12 days ago	d3ea63

UI – Warning GitHub Commits

Once the code was updated and the changes approved by the reviewer, the Commits were submitted as a Pull Request, and the Issue was closed.

Lastly, an Issue was also created to develop a warning alert display on the rover's UI for when SPECTER's body approaches an obstacle too closely. The figure below displays this Issue.



Add Warning Signal to Rover UI #5

specmj commented a few seconds ago

Warning Signal will display on the UI dashboard when obstacles are near.

Comment

Type comment here

UI – Warning GitHub Issue