PROGRESS REVIEW 8: ILR07

Richa Varma
Team I
February 11, 2016
Team members: Dorothy Kirlew, Mohak Bhardwaj, Pranav Maheshwari, Shivam Gautam

1. Individual Progress

For Progress Review 8, one of the critical goals was to completely code and test the collaboration routine for our system. Dorothy and I continued working on it, building up on the routine skeleton we created two weeks ago. The subsystem is now complete and tested for full functionality.

The main tasks involved were:

a. Incorporating additional messages (XBee and ROS) for Visualization Tool functionality
b. Defining message formats and data structures for ROS messages
c. ROS integration

2. Mesh network of multiple Xbee devices

According to our revised scope and SVE plan, we had to revise our preliminary routine to introduce the concept of virtual vehicles and to address the loopholes we had identified in the logic sequence. The following figures depict the final start and wait sequences. There would be two mobile vehicles in the parking lot and one or more stationary (parked or in queue) vehicles. On one of the stationary vehicles, a visualization tool will be attached. The vehicle that has a UI attached to it would have to do some extra processing and exchange of messages.

Mainly, new message types both for ROS and XBee had to be added to aid the visualization tool in depicting the state of the parking lot and adding virtual vehicles.

ROS messages:

a. **PARK**: Collaboration node sends ‘park’ message to the multi-agent planner to obtain desired spot ID.
b. **RETURN**: Collaboration node sends ‘return’ message to navigation, directing it to the exit coordinates.
c. **UPDATE**: Multi-agent planner sends ‘update’ message collaboration node to update current vehicle dictionary with the selected spot.
d. **PARKED**: This message is necessary for the UI to know when the vehicle has reached its spot. Navigation sends a ‘parked’ message to collaboration node which relays this information to the UI to update the display status of the vehicle.
e. **RETURNED**: A ‘returned’ message is sent by navigation to the collaboration node which triggers a Goodbye message to be sent over the network.
f. **VV**: Virtual Vehicle messages are sent by the UI to the collaboration node on the platform to which the display is attached. These aid in updating the in_queue, parking, parked, returning and returned states of the vehicles in the lot in real time.

Xbee messages:

a. **HELLO**
b. **INTRO**
c. **UPDATE**
d. **PARKED**

e. **GOODBYE**

A PARKED message was added to convey vehicle status to the UI. The 'returned' status would be taken care of by the GOODBYE message.

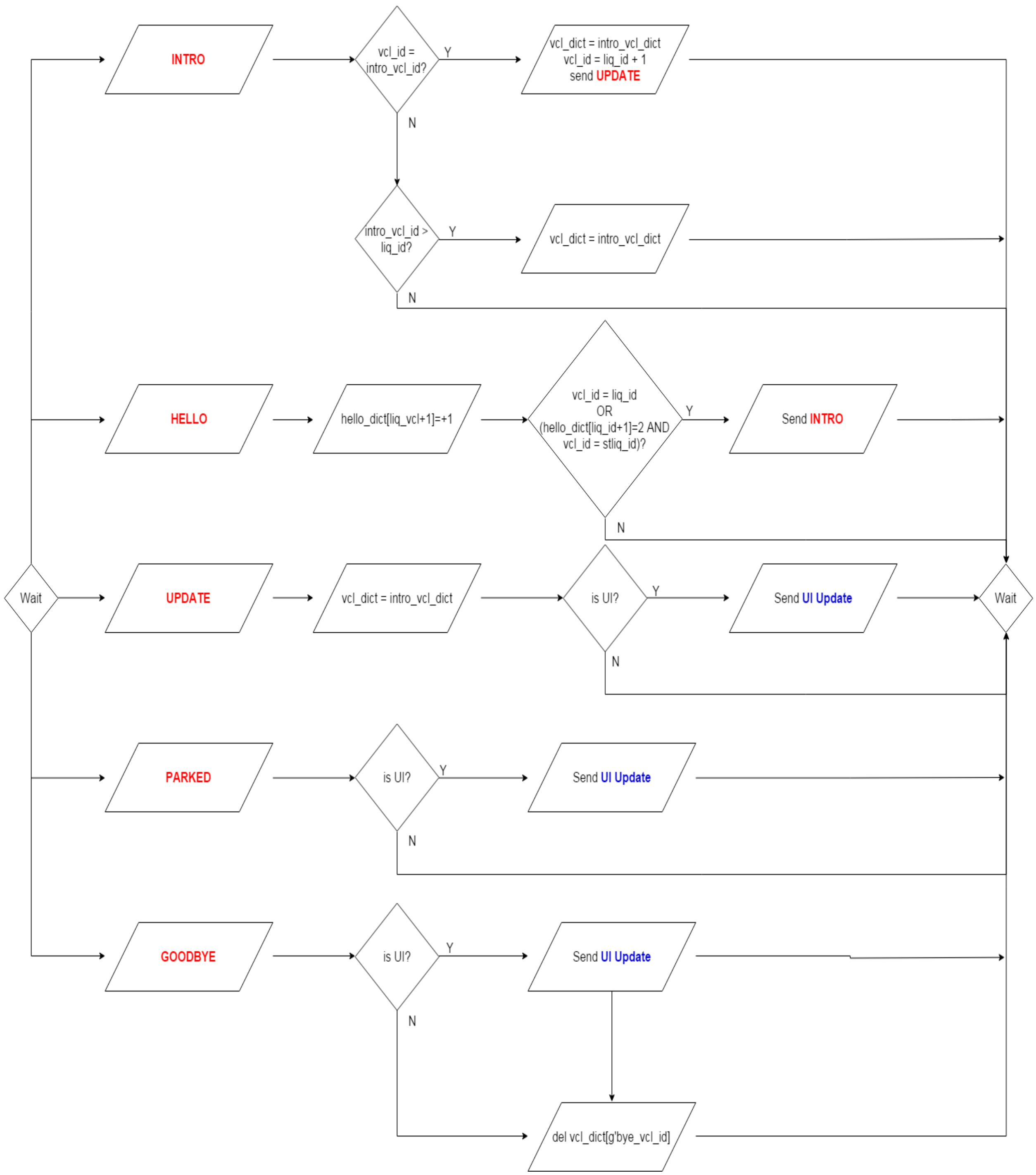

Figure 1. Xbee messages and wait routine (Figure Credits: Dorothy)

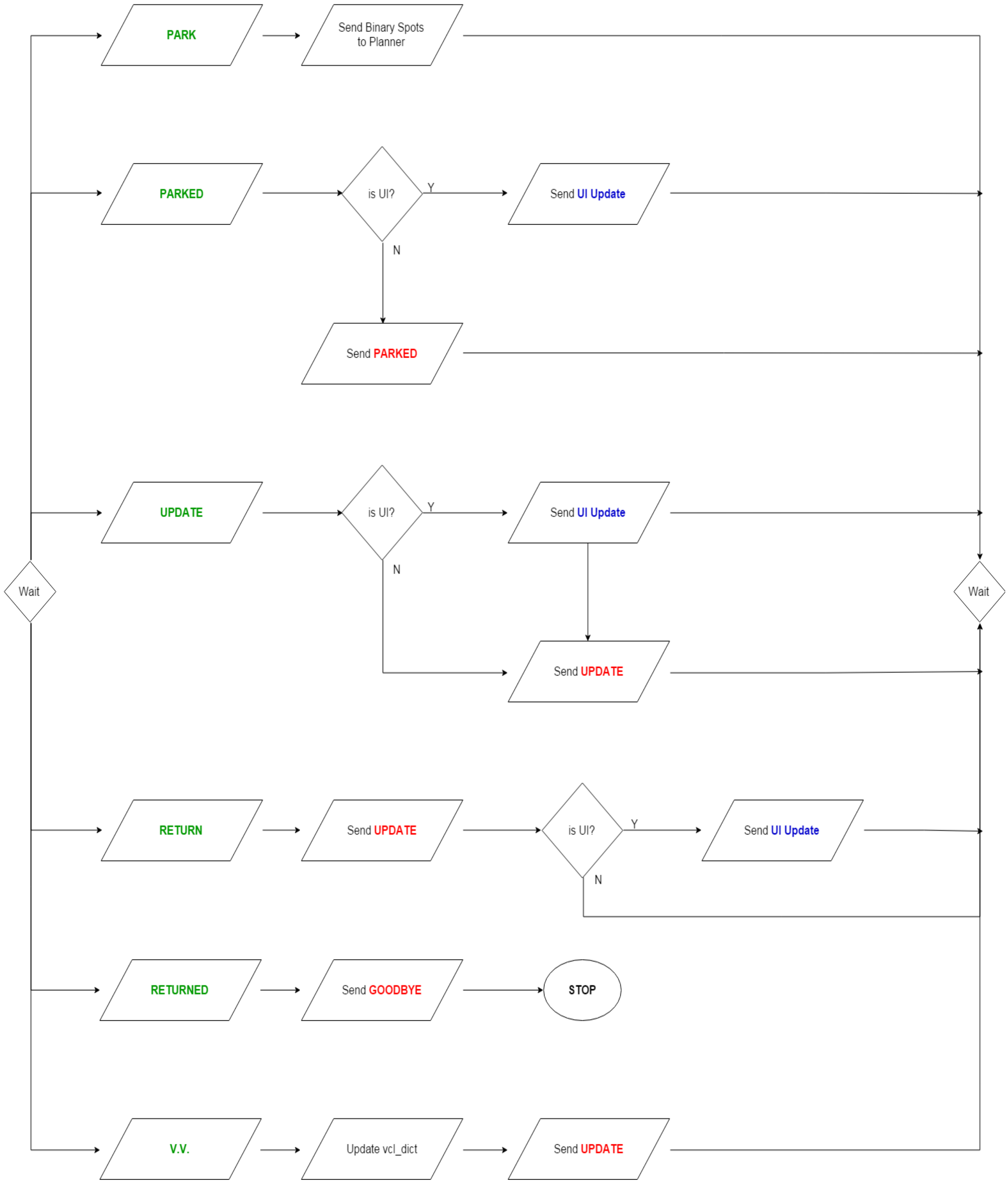

Figure 2. ROS messages and wait routine (Figure Credits: Dorothy)

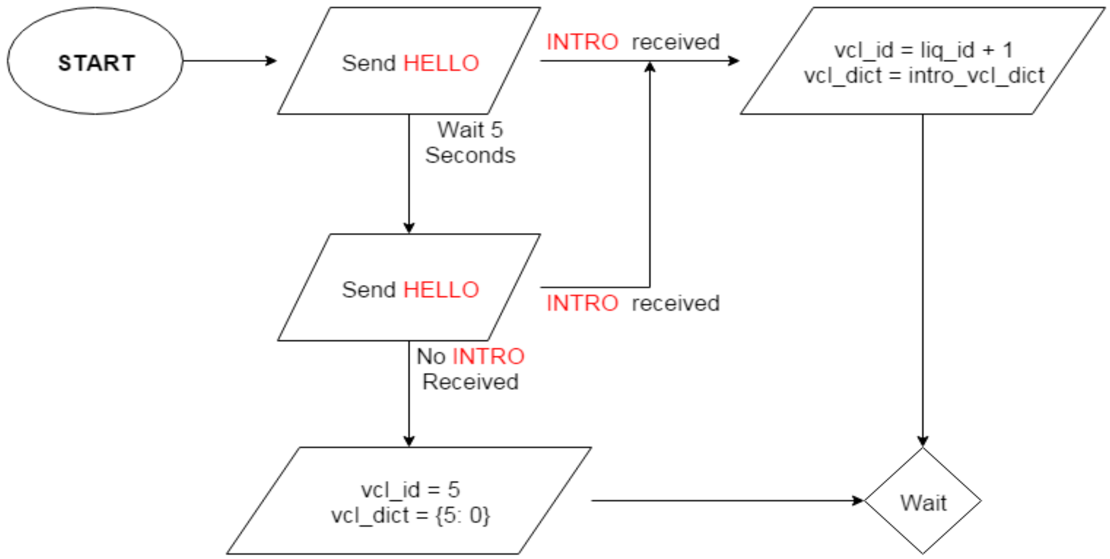

Figure 3. Start sequence (Figure Credits: Dorothy)

We also clearly defined the protocol and message data structures to ease the process of integration with other nodes.

Section Format

Node Name

- Message topic/purpose
    - Message type (string, list, etc.)
        - example message

Sent to XBees

Master

- Park Command
    - String (lowercase)
        - “park”
- Parked Status
    - String
        - “parked”
- Return Command
    - String
        - “return”
- Returned Status
    - String
        - “returned”

Planner

- Spot Chosen
    - String - SpotNumber
        - “5”

UI

- Virtual Vehicle Update
  - String - VehicleID, SpotNumber
    - “2,17”

<u>Sent by XBees</u>

Planner

- Binary Spots
  - Binary List Length 24
    - 0,1,0,0,1,0,1,0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,0,0

UI

- Vehicle State Updates: ROS Custom Message
  - string **status** (in_queue/parking/parked/returning/returned)
    uint16 **vehicle_ID**
    uint16 **spot_ID**
    - status : parking
    - vcl_id : 6
    - spot_id : 2

Navigation

- Exit Coordinates
  - String - X-Coordinate, Y-Coordinate (same format as what the Planner sends with parking spot coordinates)
    - “5.23,7.81”

For the virtual vehicle functionality, the first vehicle in the lot will be given a pre-fixed ID, e.g vcl_id=5, and the first four spots will be reserved for the virtual vehicles. This ensures that the virtual vehicles are never the last in queue, because they are purely virtual and do not have the capability to communicate. They are created just to fill up spots and change the state of the lot to demonstrate multiple scenarios.

1. Challenges

Resource issues with testing on SBCs was one of the challenges we faced. This is not a major challenge as now that we have the Intel processors that come with the Oculus platform, we can start testing on them as we incrementally integrate our system.

While coding, several unforeseen scenarios came to light which were taken into account as we went ahead. But still, we anticipate loopholes to come up and break cases to be found as we keep testing. It is very important to test robustness of message handling as we move further into integration.

2. Teamwork

For this Progress Review, I continued to work with Dorothy on the collaboration node. We completed the code and did the initial testing with two laptops. Eventually, we tested with two laptops and an Odroid XU4. Pranav completed the code for the visualization tool and worked on his code for navigation. He also worked with Shivam to CAD and print the mounts for the IR sensors and laser rangefinder. Mohak completed the code for multi-agent planner. Due to hardware issues with the platform, he was unable to work on the mapping subsystem as much as he had planned. Shivam worked on perception for obstacle detection using the laser rangefinder.

3. Plans

The team is on schedule with all the tasks and has planned to start integration this week. The subsystems that still need to be completed are the mapping and navigation. Pranav and Mohak will be working on them respectively, now that we have received a second fully integrated Oculus Prime platform. Shivam is planning to use a servo to scan the area around the vehicle with the laser rangefinder and will be working on the same. Dorothy and I will start working on the integration of the app, communication, UI and multi-agent planner.