DRLTS app

Wednesday, February 26, 2020 9:13 PM

Useful information:

- Dictionary for all tag/node related values: \app\src\main\res\layout\ node_deatils.xml

- Main files may need to check/reference/modify :
  NetworkOverviewNodeListAdapter.java
  NetworkNodeManagerImpl.java
  NodeDetailFragment.java

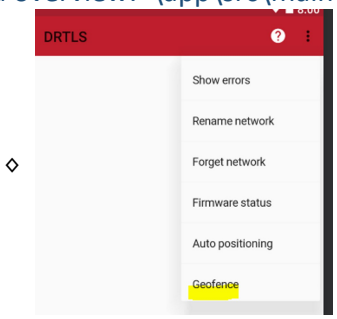
For reading the node id in edit button:

Log.i("NodeIDDebug", "NetworkNodeListItemHolder: clicked node ID is: " + Util.formatAsHexa(nodeId));
long nodeId to Hex (nodeId) is correct, issue is at going to the NodeDetailFragment page.

==========> Temporarily ignoring the edit node crash

Current ideas for Geofence: [Basic Ver]

- Add a tab at menu overview: \app\src\main\res\menu\menu_overview.xml



Write Onclick action in either: NetworkOverviewNodeListAdapter.java or NetworkNodeManagerImpl.java

- Go to a new scene allow user to input coordinates of Geofence.
- Take user input, check if the tag's X & Y position is in the fenced area every time when the tag position is refreshed. (Need to check the ble trigger, find a good place to insert the coordinate compare function.)
- Generate pop up warnings or other ways to send feedback to user/the tag

Further ideas for Geofence: [Maybe later]

- Get a storyboard and basic schematics of the current DRLTS Android application. (understand how this app work in more detail)

- Recorded current tag position and create geofence according all points recorded. (add-on function to geofence)

- Upload geofence location to database and synchronize the geofence data to all platform (add-on to geofence) Figure out how to transfer a tag from one network to another. (maybe reassign the PAN id?) [something need to figure out for sure if want to apply this to large area with multi network use case]

- Maybe also take a look to directly program the 1001 device and see how to fully use/program the device. (acceleration sensor etc.)