List of Automated Scripts

Here are a list of useful scripts that can be used either to modify scripts in the server or to run certain features such as sending fake positions or vitals data. These scripts are in */data/var/www/html/indoor-localisation-2.0/scripts/intel-edison-setup/Automated Scripts*

- BLEScanner.js
    - Picking up RSSI from certain devices and averaging it over a period of time
    - can be a testing script for the calibration scripts
    - run it in rpi
- copyFiles_rpi.expect
    - copy files over to multiple rpis via sftp
    - running certain commands in multiple rpis via ssh
    - mass shutdown
    - need to input rpis ip addresses in *pssh_hosts*
    - con: processes run sequentially so it could be quite slow when doing sftp, need to download expect command
    - run it in local computer
- fakePulse.js
    - send fake vital data (bpm,spo2) from a particular anchor and gatt sensor
    - can send healthy and unhealthy vitals
    - run it in your local computer
- getAnchorIP.js
    - retrieve a list of ip addresses that you requested
    - run it with a list of anchors as the argument
    - the script will end by writing to pssh_hosts textfile
    - run it in local computer
- inputFixedPosition.py
    - you can input a fixed position in the server
    - this script uses POI points for a particular map as the fixed location
    - checks if the position is in a NavMesh and shift position to the side of the navMesh if it is not in a navMesh
    - need to create POIs and navMesh in the map before running the script
    - when you edit the map location, you will need to edit the coordinates in line 205
    - run it in local computer
- inputRawPosition.py
    - you can input a random position in a particular map
    - when you edit the map location, you will need to edit the coordinates in line 205
    - run it in local computer
- inputRawRSSI.js
    - send raw rssi packets to the server
    - ensure the transmitterid and anchorid are registered in the server
    - run it in local computer

- listenRawData.py
  - this script can be used to test if the anchors are sending the raw rssi to the algorithm server zmq port correctly
  - can uncomment lines 21-23 to test
  - can run it in the ec2 server
- log.js
  - this script is used to give envionment variables to the pulse.js
  - these variables are usually used for anchor status
  - You can modify this file if need to amend any variables or adding to them
  - this script uses *execAsync()* and a *Promise* to return the enviornment variables
  - this file is found in *~/node_client/config*
  - run it in the rpi
- nobleScanning.js
  - this script can be used to test the noble discovery, service discovery, characteristics discovery, basically the Gatt devices connection procedure
  - And depending on the Gatt sensor, you can also retrieve other features such as battery level
  - run it in the rpi
- pssh_examples.sh
  - pssh and pscp commands to transfer files and control rpis in parallel manner
  - pro: this is much faster than the expect script
  - do note that the ip addresses of the rpis need to be written in the pssh_hosts text file
  - the features that this script can do are: remote shutdown, replace pulse.js. updating log.js, aws_config.json, stop or start pulse.js, check pulse.js status, stop pulse.js and run transmit.js for calibration
  - run it in local computer
- pulse.js
  - main rpi script
  - modify here and transfer over to rpis via sftp, expect or pscp
  - run it in the rpi
- pulse_pox.js
  - main rpi script for pulse oximeter only
  - usually used for demo only
  - try not to change it too much since it is already working
  - run it in the rpi
- pulse.service
  - systemctl service script for pulse.js in the rpi
  - if need to modify, can change here and transfer to rpi
  - to run it: *sudo systemctl start pulse*
  - run it in the rpi
- pulse_sendIP.js
  - main rpi script that includes sending ip address of the rpi to the ipAddress topic in ES
  - because anchor status already sent IP to the ES, so this script isn't very useful

    - run it in the rpi
- rssi.py
    - this script is a dependent in the *process_barycentric_latest.py* in algorithm server
    - mainly helps to calculate distance from rssi and getting Measured Power
    - run it in ec2 server (algorithm server)
- sendVitals.py
    - script to send vitals to a zmq port firebase can pick up from
    - to simulate live reading of vitals
    - not currently used actively
    - run it in ec2 server
- storePOI.py
    - dependent in a number of scripts in server
    - basically queries graphql for information
    - retrieves information such as navMesh or POIs for certain maps
    - run it in local computer, ec2 server
- storeScale.py
    - similar to *storePOI.py*
    - retrieves information such as map scale, map coordinates, map image link, anchors measured power, devices location, map anchors
    - run it in local computer, ec2 server
- wpa_supplicant.conf
    - config file for wifi settings in rpi
    - refer to rpi documentation how to use
    - unable to transfer this file directly using sftp due to security reasons
    - but you can modify this file in your local computer and copy and paste when modifying the supplicant file via nano or vim
    - run it in the rpi