ECE180DB : Lab 4 Report

Thomas Kost

UID : 504989794

Tasks Planned

- clarify best method for integration of modules
- Stop and wait ARQ implementation completed
- continue to familiarize with pyQT to allow for UI integration of modules

Tasks Completed

- Finished modifying MQTT server to accept a sender reciever architecture. User identification and applicable message filtering has been added as well as the ARQ architecture.
  - blocking threads were removed to allow for more seamless integration with other computationally demanding modules
  - message IDS are garunteed unique
  - testing scripts written
  - working on final debugging and plan to make a PR
  - supports individual messaging and group messaging
- Gesture recognition
  - hardware setup was completed
  - discussed issues with integration with current method (steaming data from IMU)
  - designed new architecture that should reduce possible jitter in sampling, allow for a higher sampling rate, and reduce effort required to integrate with pyQT
  - this method is offloading all relevant processing and sampling to the raspberry pi. From there we will perform our classification–this result will trigger a mqtt server to send a keyword triggering an event in out state machine.

Future Tasks

- pull request for mqtt server changes
- add user identification manager to allow it to be more expandable and add more functionality
- expand gesture recognition data set, work on further validation and add mqtt server to existing classifier. Classification is capable of being run on a set of data being live read in. This will preserve a high sampling rate
  - add testing script for live performance
  - more validation is always a good thing, but goes hand and hand with expanding data set
  - if possible we can take advantage of the already built one vs one classifier and add multiple gestures