IoT Course Project 1

IoT system development

- **Project**
  - Develop a small IoT system (1)
    - Connecting IoT sensors and actuators to Raspberry Pi
    - Write a program to read sensors and control the actuators according to the sensor output
    - C: [http://wiringpi.com/](http://wiringpi.com/)
    - Python: [https://sourceforge.net/p/raspberry-gpio-python/wiki/Inputs/](https://sourceforge.net/p/raspberry-gpio-python/wiki/Inputs/)
  - Use MQTT to publish and subscribe data (2)
    - Install a local MQTT broker: [https://mosquitto.org/](https://mosquitto.org/)
    - Write a publisher program on Raspberry Pi, which reads data from some sensors and publishes the data using MQTT
    - Use your own laptop to deploy an MQTT subscriber to get the published data
    - MQTT programming support: [https://pypi.org/project/paho-mqtt/](https://pypi.org/project/paho-mqtt/)
  - Develop a voice command interface for the IoT system (3)
    - Install Node-RED on your own laptop
    - Put together a Node-RED flow
      - [https://nodered.org/docs/user-guide/](https://nodered.org/docs/user-guide/)
    - The flow includes a node to capture voice from the mic of the laptop
      - Python code sample for capturing audio: [https://gist.github.com/mabdrabo/8678538](https://gist.github.com/mabdrabo/8678538)
      - Node-RED: [https://flows.nodered.org/node/node-red-contrib-dynamorse-audio-io](https://flows.nodered.org/node/node-red-contrib-dynamorse-audio-io)
    - You should have a speech-to-text node in the flow to process voice commands
      - Google Cloud stt service: [https://cloud.google.com/speech-to-text/docs/reference/rest/](https://cloud.google.com/speech-to-text/docs/reference/rest/)
      - Node-RED: [https://flows.nodered.org/node/node-red-contrib-viseo-google-speech](https://flows.nodered.org/node/node-red-contrib-viseo-google-speech)
    - The flow should include an MQTT node to perform the subscription and use the data in your flow
      - [https://cookbook.nodered.org/mqtt/](https://cookbook.nodered.org/mqtt/)
    - You need a node to communicate with Raspberry Pi
      - [https://flows.nodered.org/node/node-red-contrib-socketio](https://flows.nodered.org/node/node-red-contrib-socketio)

- **Objective**
  - Learn about various IoT platforms and tools and how to use them for constructing IoT systems

- **An example IoT system for your project (1)**
  - **Devices**
    - Sensors: PIR, sound, luminance, etc. sensors
    - Actuators: light bulbs
  - **Automated lighting control**
    - Get sensor inputs
      - Get sound sensor input event
      - Get PIR sensor input event
      - Read light sensor input
    - Activities
      - Upon PIR event, read light sensor, switch on one of the lights if the luminance is lower than a user pre-specified level
- E.g., if it is day time, the room is anyway luminated, no need to switch the light on (use lab lights to simulate daylights)
- Depending on the required lighting level, switch on the appropriate bulb(s)
  - After a pre-specified duration without PIR event
  - Turn off the lights

- Extend the project with voice commands (3)
  - Specifically command the lighting system to turn on lights
  - Use voice commands to setup the system parameters

- Extend the project with MQTT data flow (2)
  - When sound is detected, pass the event to the MQTT broker
  - Subscribe the sound event data
  - Feed the sound event signal to the Node-RED flow, and trigger the voice recognition node when sound is detected

Design of a more complex IoT system
- When designing your IoT system, first consider a large-scale system that is useful in real life
  - Try to pay more attention to multi-tenant IoT systems
  - Try to consider more interesting policies that requires more complex analysis
- For example
  - UTD campus management
    - Entry control using facial recognition
    - Track the individuals, providing on-demand transportation services, lighting control and AC control of individualized zones in buildings, customized information resources to be displayed on the available screens, …
    - Energy conservation policies for the buildings
    - Safety and security policies for the campus
    - User access control to rooms, facilities, equipment, information resources, …
- After designing a large-scale IoT system, scale it down to implement a small part of the system
- If you construct the large-scale IoT system you have designed, what features are needed but do not exist in some of the IoT platforms

Submission
- Report
  - Your system design and implementation
  - Efforts of the members
    - Discuss which member worked on which part of the project
  - Vision
    - Based on the project, assume that you can scale the small project up, what would be the large-scale IoT system you would like to construct
    - Try to pay more attention to multi-tenant IoT systems
    - Try to consider more interesting policies that requires more complex analysis
    - If you construct such an IoT system, what features are needed but not exist in some of the IoT platforms
- Demo
  - Discuss the system you designed and show it actually works
    - Goal of the system, middleware used, workflow realized in the middleware
  - Discuss the contributions of each group member
  - Discuss your vision of an extension to a large-scale IoT system