ILab 08: Raspberry Pi Server

UCR CS 122A

Pre-lab

Your Raspberry Pi and ATMega1284 should already be wired together through SPI from the previous lab. You can complete this lab either keeping the LCD and photoresistor on separate microcontrollers or you can combine them to the same microcontroller if you would like.

Introduction

Now that you are able to retrieve data from the sensor on the microcontroller and send it to be displayed, we will now work on communicating with a server through an API call. First, you are going to spin up a similar server locally on your raspberry pi for development purposes.

Setting up the local environment (server)

1. You are going to create a new directory on your RPi for this lab
   ```
   $ mkdir Lab08-RPiServer; cd Lab08-RPiServer
   ```
2. Create a virtual environment in this top-level directory
   ```
   $ virtualenv -p /usr/bin/python3.7 lab8_env
   $ source lab8_env/bin/activate
   ```
3. Create a directory for the server code (RPiServer).
   ```
   $ mkdir RPiDevelopmentServer
   ```
4. You are going to need to install flask to run the server
   ```
   $ pip install flask
   ```
5. A development server has been provided here.
6. Start the local server
   ```
   $ python lightServer.py
   ```
   Note: This will boot up a development server. If you make any changes to the script the server will reload it automatically.
7. Check if the server is running. In another terminal (ssh into your raspberry pi) type:
   ```
   $ curl 127.0.0.1:5000
   ```
   And you should see the response below in the terminal:
   ```
   {
         "Average": 0,
         "Requests": 0
   }
   ```

Setting up the client

1. Create a directory for the client code (RPiClient) at the same level as RPiDevelopmentServer
   `$ mkdir RPiClient;`
2. Install the requests package to be able to make requests to the server
   `$ pip install requests`
3. You will want to copy your code from Lab 7 over (and verify it still works) to start working from. At the top of the file, you will need to add some additional imports for the requests library
   ```
   import requests
   from requests.exceptions import HTTPError
   import time
   ```
4. Next, you will want to complete the following function for the Raspberry Pi Zero W. (You will need to change the IP based on your local development server)

```
def server_sendLightData(newData):
    # Send data to server (http://0.0.0.0:5000/submitLightValue/<newData>
    # If there is an error, print the error (and context)
    return average
```

Exercises

1. Complete the system above with the Raspberry Pi running a local server that will calculate the running average. The client script will submit new light values using the provided API and send the averages to the LCD.
   Note: Get the TA to check you off before moving on.
2. Replace the local server with the class server (`pi-wireless:8000`)
3. (Challenge) Set this script up to run on boot. If you did this challenge for the last lab, make sure only one script runs on boot.

Submission

Each student must submit their source files (`.c, .py`) and test files (`.gdb`) according to instructions in the lab submission guidelines.

`$ tar -czvf [cslogin]_lab3.tgz turnin/`

Don't forget to commit and push to Github before you logout!