Lab1 – Simple IoT Application

NCTU Introduction to IoT

October 23, 2019
Outline

• Objective

• Device side
  • Controlling LED with Raspberry Pi / Switch / Light Sensor (LDR)
  • Temperature & Humidity monitoring

• Server side
  • Sending sensor values to a server
  • Controlling actuators via Website
Objective

• Simulating a smart-home environment (automatic lamp, environment monitoring):
  • Lamp will be turned on automatically if the room gets dark and turned off automatically if the room have enough light.
  • Controlling the lamp on the website.
  • Monitoring temperature and humidity on the website.
Device side
Basic

• Breadboard Connections

• Raspberry pi 4 pins
Controlling LED with Raspberry Pi (1)

• Diagram
Controlling LED with Raspberry Pi (2)

• Components:
  • LED
  • A resistor 220.

• Using file LEDAndRaspberry.py in source code
Controlling LED with Switch (1)

• Diagram
Controlling LED with Switch (2)

- Components:
  - LED
  - Switch
  - 2 resistors 220.

- Using file `LEDAndSwitch.py` in source code
Controlling LED with Light Sensor (1)

• Diagram
Controlling LED with Light Sensor (2)

• Components:
  • LED
  • Resistor 220
  • LDR (Light Dependent Resistor)
  • Capacitor 1μF

• Using file LEDAndLDR.py in source code
Temperature & Humidity monitoring

- sudo apt-get update
- sudo apt-get upgrade
- sudo apt-get install python3-dev python3-pip
- sudo python3 -m pip install --upgrade pip setuptools wheel
- sudo pip3 install Adafruit_DHT
Temperature & Humidity monitoring

• Components:
  • AM2302 DHT22 Sensor
    • VCC (+)
    • GND (-)
    • DAT (data)

• Using file LEDAndLDR.py in source code
Server side
• **Note:** we build the server on your laptop or PC by using Django. But we also can use some public clouds on the internet.

• Install python3

• Install virtual environment
  • `pip install virtualenv`

• Create virtual environment
  • `virtualenv <venv_name>`

• Active virtual environment
  • `<direction_of_environment>\Scripts\active`
Build A Server (2)

• Install Django
  • pip install django

• Install Django REST framework
  • pip install djangorestframework

• Create a project \textit{(from now you run all commands at this folder)}
  • django-admin startproject \texttt{SmartHomeProject}

• Create an application inside the project
  • python manage.py startapp \texttt{myApp}
Build A Server (3)

• Tell Django to use the application
  • `mysite/settings.py` → find `INSTALLED_APPS` → add 'myApp.apps.MyappConfig', 'rest_framework',

• Create model
  • `myApp/models.py` (into source code)

• Create serializers
  • `myApp/serializers.py` (source code)

• Add models to Django admin
  • `myApp/admin.py` (source code)
Build A Server (4)

- Create superuser (admin)
  - python manage.py createsuperuser
- Create view
  - myApp/views.py
- Create url
  - SmartHomeProject/urls.py (source code)
  - myApp/urls.py (source code)
Build A Server (5)

• Create HTML file
  • myApp/templates/index.html (source code)

• Create a database
  • python manage.py migrate

• Run server
  • python manage.py runserver 0.0.0.0:8000
Using website

• Access website
  • http://<serverIP>:8000/

• Django administration website
  • http://<serverIP>:8000/admin/
  • Go to this website to create an object of the Light model before the first time you use the home website.
  • Also create an object of the TemperatureData model by using admin website or Web API before the first time you use the home website.

• NOTE:
  • Please turn off your server’s firewall.
  • Add ALLOWED_HOSTS = ['*'] inside setting.py
Using web api

• url for temperature api
  • http://<serverIP>:8000/api/temperature

• url for light api
  • http://<serverIP>:8000/api/light

• Methods (install “requests” first: pip install requests)
  • GET: return data
    • Requests.get(‘url’)
  • POST: create data
    • Requests.get(‘url’, {‘key1’:‘data1’, ‘key2’:‘data2’})
Sending Sensor Values To A Server

• Components (using “Temperature & Humidity monitoring” slide)
• Using file InterfacingWithServer\TempHumidity.py in source code.
Controlling Actuators Via Website

• Components (using “Controlling LED with Light Sensor” slide)

• Using file `InterfacingWithServer\LEDAndLDR.py` in source code.
References

• Internet of things a hands-on approach by arshdeep bahga and vijay madisetti
  • https://pimylifeup.com/raspberry-pi-humidity-sensor-dht22/
  • https://projectiot123.com/2019/02/01/raspberry-pi-gpio-programming-example-for-servo-motor-using-python/
  • https://tutorial.djangogirls.org/en/