COE 4DN4 Lab1 -
A Simple 'Internet-of-Everything' Smart-Home Server

Lab 1 Posted: Tuesday, Jan. 13, 2015
Labs: M,T,W,Th - JHE-234 EOW

TA Supervision & Demo Times (for Labs 1 and 2):
Labs: M,T,W,Th - JHE-234 EOW

TA Supervision & Demo Times (for Labs 1):
(These labs will be supervised if 2+ groups reserve time on each day)
Wed, Thu - Jan 21,22
Wed, Thu - Feb 4,5
(or by appointment)

Lab 1 Brief Report Due: Mon, Feb. 9, 2015 in Avenue-to-Learn drop-box (by 10 am)
(check email for updates on this lab)

Goal:

The Internet is undergoing a fundamental transformation and evolving into the "*Internet-of-Things*" (IOT) and more recently the "*Internet-of-Everything*" (IOE). The IOE will interconnect people, processes and devices, to yield a better and more efficient world. It will allow for Smart-Homes, Smart-Buildings, Smart-Cities, Smart-Manufacturing, Smart-Supply-Chains, etc. You will not find this material in any textbook since it is too new, but it was the hottest topic at the 2014 and 2015 CES and other industry tradeshows. Please see these URLs for nice easy-read summaries:

http://blogs.cisco.com/ioe/beyond-things-the-internet-of-everything-takes-connections-to-the-power-of-four/

http://www.cisco.com/web/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf

At the 2014 CES, the Cisco CEO estimated the commercial value of the IOE to be $16 Trillion US over the period 2014-2022. To realize this value, the world will need to create a tremendous amount of easy-to-use intuitive software, to enable all people to easily interact with the IOE to achieve their goals.

Please download the TCP Echo Client-Server code. (For C code, use the Donahoo site; for Python code, you can use the Echo software on the class web-site). Change the TCPServer code to act as a IOE Smart-Home Server.

No organization like IEEE has yet defined a standard for what a Smart-Home Server might look like since the IOE is so new, so we will define our own simple server. The Smart-Home server will allow any smart devices to 'register themselves' to the server. Once registered, the home-owner can then read the current-read-value and write the current-write-value. The IOE is expected to interconnect 50 billion smart-devices by 2022. The startup NEST builds smart thermostats, and was recently bought by Google for $3.2 Billion. Each year, the CES demonstrates smart thermostats, smart slow-cookers, smart refrigerators, smart moisture-sensors for your lawn/garden, smart-cars, etc. Each year there are even more smart-devices announced.

The Smart-Home IOEserver maintains a data-base with many rows (say 64) and 4 columns: In column A is a text string for the smart-device name, with length = 128 bytes. In column B is a number with the current-value of some variable, read from a sensor, with length = 32 bits. In column C is another number with the target-value of the variable, with length = 32 bits. The target-value is the value the user sets the smart-device too (ie a thermostat temperature). You need to create a C, Java or Python data-structure for this data-base. (If you use Python, there are no strict 'data-types', so the data-base can store generic text and numbers.)

You may initialize the first 2 rows of this data-base in your IOEserver code for testing as follows (here I use commas to separate the fields):

Device-Name (Text-String)	Read-value (number)	Target-Value (number)	IP-address
Thermostat-Main Room	19	23	177.68.25.17
Thermostat-Living-Room	18	22	177.68.25.18

The IOEserver responds to 4 commands from the IOEclient. An IOEclient connects to the IOEserver, and can execute the following commands. The IOEserver is running on a given IP-address and port, associated with the Smart-Home.

Command	Parameters to send
CONNECT	IP_address, Port
ADD	'device-name', device-IP-address
REMOVE	'device-name'
READ-value	'device-name'
WRITE-target	'device-name', target value (32 bits)
QUIT	

The READ-value command will return 2 values associated with the smart-device to the IOEclient, the read-value and the target-value. The WRITE-target command will write a value to the smart device.

You may work in teams of 1-3 students. The working system should be demonstrated to a TA during the laboratory demonstration hours, by all members of the team. The TA will record a presentation mark for the entire team. You must be present during the team's presentation to

answer questions about the software, to be considered part of a team. The TA(s) will ask questions during the demonstration to assess your software, and your contributions.

WHAT TO SUBMIT:

The lab report should be brief and have these sections:

(1) Objective and brief introduction - a few sentences (1 page tops)
(2) IOEServer flow-chart* (thorough self-explanatory flow-charts)
(3) IOEClient flow-chart*
(4 Well-documented C, Java or Python code, referring back to flow-chart when necessary
(5) Experimental results (include a few sentences and a few screen snap-shots to demonstrate working software)
(6) Issues / Problems that were encountered & needed to be solved. Enumerate any issues. If we know common problems, we can address them for next year's class.
(7) Record name of TA to whom you demo-ed the system
(8) Conclusion - brief summary

If you have demonstrated the software, then the report is very brief and simply documents what you have demonstrated. The flow-charts are important and worth marks, so please follow our class flow-chart format. The class flow-chart format is explained next:

Flow-chart format: : All variables, vectors, arrays, data-structures should be clearly defined. The definitions show executable software code, in your language of choice. All access to the OS or to the socket library functions are complete and show executable software code, in your language of choice. The logical flow of the program (if-the-else, while loops, etc), can be written in words that refer to your variables; the logical flow does not need to be executable code.

Please submit a electronic-copy of the lab report along with professionally documented software on Avenue-to-Learn.

Create a folder called 4DN4-Lab1-Submission-XX-YY (xx-yy are the initials of your group members). The folder contains the brief report (PDF format), and the flow-charts (PDF format). Also create a folder for the IOEServer code and a folder for the IOEClient code, and include the documented C, Java or Python code in appropriate folders. Only 1 team member needs to upload a report, but the report must mention all team members and student numbers in the front page.

I encourage everyone to start and finish the 4DN4 labs early. They should be fun to do and there is no need to wait until the last week before they are due. The time-sots for lab. demos are shown above. Each team needs to reserve a 15 minute demo-slot with the TA before the lab. session, on a First-Come-First-Served (FCFS) basis. If you finish 4DN4 lab 1 early, please start work on 4DN4 lab 2. The prof or TAs should be able to help you with either lab.

Hints: Defining Data-Structures in Python

structures2.py (no symbol selected)

```
1
2   # Create a data-structure in Python
3
4   # This code works, but there may be more elegant ways to create data-structures.
5
6   # Note: I used tabs to indent this text.
7   # Python seems to complain if you mix tabs with spaces
8   # so try to be consistent in how you indent your code.
9
10  class  Pet(object):
11      name = None
12      type = None
13      food = None
14
15  # create an array [0 ... 9] called Pets
16  # note the use of square brackets to create a null-list, like Matlab
17
18  Pets = []  # a null list
19  for i in range(0, 10):
20      nextpet = Pet()
21      Pets.append(nextpet)
22
23  # note the use of square brackets to select one element from an array
24  y = Pets[1]
25
26  Pets[1].name = 'Dude'
27  Pets[1].type = 'Dog'
28  Pets[1].food = 'Performatrin dog food'
29
30  Pets[2].name = 'kitten'
31  Pets[2].type = 'Cat'
32  Pets[1].food = 'Performatrin low-fat cat food'
33
34  print 'My 1st pet ' , y.name, ' is a ', y.type
35  print 'My 2nd pet ' , Pets[2].name, ' is a ', Pets[2].type
36
```

A Voice-Operated Internet-of-Things ?

Here is an interesting article. Facebook just bought a startup company that hopes to use voice-translation to control the Internet of Things.

Facebook Acquires Voice Recognition Startup Wit.ai $FB

Jan 05, 2015 | facebook, google, Voice Recognition, Wit.ai | 0 Comments

January 5, 2015
Thomas Halleck
Posted with permission from International Business Times

Republish
Reprint

Facebook is acquiring Wit.ai, a voice recognition startup that looks to offer voice interaction with the growing Internet of Things.Facebook Inc.

Facebook Inc. is acquiring a speech-to-text startup called Wit.ai, it announced Monday. Wit.ai's software can translate spoken words into usable data much like Apple Inc.'s Siri or Google Now, and can be added to existing mobile apps.

Wit.ai says it is powering "hundreds" of apps and devices after it was founded 18 months ago, and has over 6,000 developers using its platform. In addition to mobile applications, Facebook's latest acquisition sees a place for its voice recognition software among the Internet of Things, a myriad of connected sensors, including product categories like home automation, wearable technology and robotics, that Gartner predicts will number 4.9 billion by the end of 2015.

"Facebook has the resources and talent to help us take the next step," Wit.ai said in a company blog post. "Facebook's mission is to connect everyone and build amazing experiences for the over 1.3 billion people on the platform -- technology that understands natural language is a big part of that, and we think we can help."