IUPUI ■ CSCI 43300: Internet of Things ■ Spring 2023

Project 2 Assignment

Due Date: March 29, 2023

General Instructions:

1. This assignment is to be carried out by a team of no more than two students.
2. Plagiarism is a violation of the IUPUI academic ethics and will not be tolerated. All material quoted directly from other sources must appear within quotation marks and appropriately referenced; all material that builds upon other sources must be appropriately referenced. This project must be strictly your team's work. Be sure to contact the instructor in case you have any questions as to what constitutes plagiarism or another academic ethics violation.
3. Your report will be graded for both content and form. It is important that the report be easy to read and understand (no misspellings, no grammatical mistakes, well labeled figures where appropriate, etc.). The report should contain no hand-written notes or hand-drawn figures.
4. Some milestone(s) might be set by TA to make sure your work in progress.

Objectives:

1. Learn how to connect a smart device to the Internet in a simple IoT application.
2. Learn how to use CoAP to develop IoT applications.

Problem Statement:

In this project, you are going to develop a simple server on your smart device for any Internet client (e.g., Google Chrome Copper or your own developed CoAP client on laptop) to query the sensor readings obtained on the smart device using CoAP protocol. For example, you may run *server.py* on your pi and *client.py* on your laptop, which depends on which library you use. You can use other sensors besides temperature sensor. Refer to *recitation.pdf* if you need to borrow sensors from lab.

Step 0: For Arduino, learn how to use WiFi Module (for Arduino teams only).

Step 1: Install the CoAP library and configure development environment. Run a "Hello World" application where client sends a request and <<Hello World>> will be printed in the payload at client end after server receives the request.

Step 2: Hook up sensors (e.g., temperature, acceleration, etc.) to Arduino/Raspberry Pi. Add the sensor readings as "resources" of the CoAP server, and provide sensor readings once a request is received from an Internet client. (That is, a GET request from a client.) Try repeating the request to get a different reading.

Step 3: Use Wireshark to analyze CoAP packets.

Report:

Your report needs to be complete, clear and concise. Presentation and demo (either live or prerecorded video demo) are required.

Your report should include a brief introduction, (1) a detailed description of your solution to Step 2 (e.g., circuit schematic, flowchart of your code logic, etc.), (2) your analysis (Step 3), (3) any issues encountered and how they are resolved; and (4) for team project, indicate which one works on which part of the project, and the conclusion. You may include other parts as well, for example, how to use a Web browser or your own developed CoAP client (including your development) to access your CoAP server.

You need to submit your circuit schema, all source codes, report and video demos in a single compressed file (i.e. zip) following project 1's naming convention.

Rubrics:

Step 1: 15 points

Step 2: 35 points (Server IP address and client IP address should be different to receive full credits.)

Step 3: 20 points

Report: 20 points

Demo: 10 points

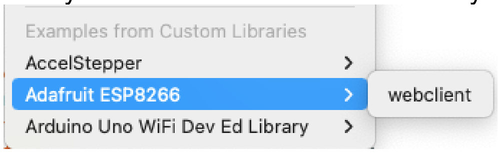
Bonus: 5 points (Add a different sensor for Step 2)

Useful Information and Websites:

0. **Common:**
   - (Tutorial video) https://www.youtube.com/watch?v=4bSr5x5gKvA
   - There are CoAP clients in iOS App Store as well.
   - (Google chrome copper as CoAP client) https://github.com/mkovatsc/Copper4Cr

- **Raspberry Pi:**
  - (Californium: Java CoAP implementation) https://github.com/eclipse/californium
  - Introducing the CoAP for Java (How to build a basic CoAP server and client): https://blogs.oracle.com/javamagazine/post/java-coap-constrained-application-protocol-introduction
  - (CoAPthon: Python CoAP library) https://github.com/Tanganelli/CoAPthon
  - (Aiocoap: Python CoAP library) https://github.com/chrysn/aiocoap
  - (txThings: Python CoAP library) https://github.com/mwasilak/txThings
  - (Set up an Apache Web Server for R Pi) https://www.raspberrypi.org/documentation/remote-access/web-server/apache.md

- **Arduino:**
  You will find <<webclient>> example code for simple ESP8266 test under <<Examples>> folder after you install Adafruit ESP8266 library.


Examples from Custom Libraries
AccelStepper >
Adafruit ESP8266 >
Arduino Uno WiFi Dev Ed Library >
webclient

  - Java Control of Arduino (RXTX Java library): https://playground.arduino.cc/Interfacing/Java/
  - Java Control of Arduino (jSerialComm library): https://fazecast.github.io/jSerialComm/ & https://www.youtube.com/watch?v=jLNoKa4JReY
  - (libcoap: C-Implementation of CoAP) https://libcoap.net/
  - CoAP client/server example code for ESP8266: https://github.com/automote/ESP-CoAP/blob/master/examples/
  - Example code (ESP8266-01 + Arduino + CoAP) : https://www.engineersgarage.com/client-server-communication-over-coap-protocol-iot-part-33/