Internet of Things - Lab Notes#1

Lab 1: Basic connection of the device to cloud - Registration and Discovery

Software needed on your computer for this lab exercise

The mbed IDE is web based and requires no installation on your computer. There is some software that you will need for diagnostics and debugging:

1. Terminal Emulator program: e.g. putty for windows; mac users can use the “screen” program. For receiving diagnostic messages from the embedded software.
2. REST client software, e.g. Advanced REST Client for Chrome. To test REST API operation.
3. Windows users will need to install a serial link driver for diagnostics monitoring: http://developer.mbed.org/handbook/Windows-serial-configuration

Step 1. Get an mbed account

If you don’t already have an mbed account, go to
http://mbed.org/
and create a free account for yourself.

ARM mbed OS is a web based IDE and platform for creating embedded and connected software projects for ARM Cortex-M based microcontrollers (MCU). The IDE runs entirely on the web and downloads a binary file to your computer that you drag and drop to your mbed device, which appears as a storage drive when connected via USB. There is no software to install on your computer to use mbed. A serial terminal emulator is used to monitor diagnostic information from the mbed libraries and from your embedded application program.

Step 2. Connect the mbed board to your computer

1. Use a micro USB cable to connect the USB port labeled “Debug” on the Arch Pro to a USB port on your computer.
2. A new device called “mbed” should appear in the storage manager. You will drag and drop files here to program the mbed board.

Step 3. Import, Compile, and load the Hello mDS demo application onto the mbed board

1. Go to the project page at: http://developer.mbed.org/teams/MBED_DEMOS/code/Hello-mDS-ArchPro-ethernet/
2. Import the demo program using the “Import this program” button on the project page.
3. Make sure the Platform Seeeduino-Arch-Pro is selected in the upper right part of the screen. Compile the project. A few compiler warnings may be generated. If successful, a

file named *Hello-mDS-ArchPro-ethernet-LPC1768.bin* will be stored in your default location, e.g. Downloads folder.

4. Drag and drop the compiler output file to the “mbed” device in your storage manager which represents the Arch Pro
5. When done copying (a few seconds), unplug and replug the USB connection to power the Arch Pro off and on again.
6. Status messages during startup and network connection can be monitored by connecting a terminal application to the USB serial port, baud rate = 9600. Pressing the reset button on the side of the mbed board at any time will cause a restart, enabling the diagnostic messages to be monitored from the beginning again.

Step 4. Connect your mbed board to the mbed Device Server instance.

After successfully installing the demo, the mbed board will automatically connect to the mbed Device Server instance at coen196.cloudapp.net

Monitor the USB serial port during startup; you should see output similar to the following:

```
[NSDL_DEBUG: /src/main.cpp:159]Hello mDS Demo Endpoint Application

[NSDL_DEBUG: /src/main.cpp:68]DHCP in use

[NSDL_DEBUG: /src/main.cpp:70]eth.init

Connect OK
[NSDL_DEBUG: /src/main.cpp:78]IP Address:10.0.0.45
MAC: 1E70FF0C071E
[NSDL_DEBUG: /src/main.cpp:104]name: mbedDEMO-1E70FF0C071E
[NSDL_DEBUG: /src/main.cpp:105]NSP=23.102.162.118 - port 5683

libNsdl init done
[NSDL_DEBUG: /src/main.cpp:117]Creating resources
TX callback!
Sending 81 bytes
NSP registering OK
Received 37 bytes
RX callback!
msg_code: 65
Payload length: 0 bytes
Payload:''
Location: /rd/domain/mbedDEMO-1E70FF0C071E
```

The return of the location information indicates that your mbed board is now connected to the mbed Device Server and can be accessed through the RESTful web API. Your MAC and IP addresses will be different from this example. Please make note of the NAME, which starts with

‘mbedDEMO-’ followed by the 12 digit concatenated MAC ID. This is your unique endpoint ID in the system will be used to construct the HTTP path to your resources.

5. Verify the registration with mbed Device Server

The mbed Device Server instance has 3 ports open for access:

http://coen296.cloudapp.net:8080/domain/endpoints
REST API resources representing endpoint data, see the User Guide page for more detail and instructions.
User=’app2’
Password=’secret’

https://coen296.cloudapp.net:8081
Administration interface to monitor endpoints, applications, and server status.
User=’admin’
Password=’admin’

coap://23.102.162.118:5683
CoAP endpoints connect to the ARM Sensinode platform and transfer data using the IETF CoAP protocol over UDP. The IP address is given here fof use in the embedded

Use your browser to navigate to https://coen296.cloudapp.net:8081
The username is “admin” and the password is “admin”. Select “Endpoints” and look in the list. You should find your unique endpoint name among the registered endpoints.

The resource schema for the demo follows the OMA LWM2M object model.

5. Use a REST Client to actuate the LED resource

The Base URL for all endpoints is:
http://coen296.cloudapp.net:8080/domain/endpoints

Endpoint discovery: Doing a GET to this URL returns a JSON formatted list of registered endpoints.

Using a REST API client, such as the “Advanced Rest Client” Chrome plugin, perform a GET to the base URL and find your endpoint name in the returned JSON formatted list. A JSON containing entries similar to the following should be returned.


```
0:
{
   name: "mbedDEMO-1E70FF0C071E"
```

```
    type: "DEMO"
    status: "ACTIVE"
}
```


The endpoint name is added to this base path to identify an individual endpoint:
http://coen296.cloudapp.net:8080/domain/endpoints/mbedDEMO-1E70FF0C071E/

Resource discovery: Doing a GET to this URL returns a JSON formatted list of Objects and Resources for the specified endpoint.

Using a REST API client, such as the "Advanced Rest Client" Chrome plugin, perform a GET to the URL that represents your endpoint (substitute your endpoint name above). A JSON similar to the following should be returned:


```
    0:
    {
        uri: "/11100/0/5900"
        obs: false
        type: ""
    }
    -
    1:
    {
        uri: "/3/0/1"
        obs: false
        type: ""
    }
    -
    2:
    {
        uri: "/3/0/0"
        obs: false
        type: ""
    }
```


Finally, the resource is identified by adding the resource path uri as per the following:

11100/0/5900

For example, the current LED setting for the above example board can be obtained by performing an http GET to the address:
http://coen296.cloudapp.net:8080/domain/endpoints/mbedDEMO-1E70FF0C071E/11100/0/5900?sync=true
Note ?sync=true to get a synchronous http response and see the value in your web browser or client. The server uses http basic authorization using the user name “app2” and password “secret”.

The data type is an LED Control String, 4 characters, 0 or 1, representing YBRG colors.
Storing a value of “0000” turns all LEDs off, “1111” turns them all on.
Storing a value of “0010” turns on the RED LED only, etc.

Using a REST API client, such as the “Advanced Rest Client” Chrome plugin, perform a GET to the URL that represents the LED resource (substitute your endpoint name above). A JSON STRING containing the current value should be returned:

0000

Using a REST API client, such as the “Advanced Rest Client” Chrome plugin, perform a PUT to the URL that represents the LED resource (substitute your endpoint name above). Supply a 4 character string of “1” and “0” to control the respective LEDs.

1010

Supplying the above string will illuminate the Yellow and Red LEDs only.

Your mbed board is now connected to mbed Device Server and you can control it using web applications through the mDS REST API.

The next lab note will explain and demonstrate the mbed library interfaces used to create new resources and embedded applications.

For more information:

ARM Nano Service Platform User Guide (mbed Device Server) - in the course materials

LWM2M Tutorial
http://community.arm.com/docs/DOC-8693

CoAP Tutorial
http://community.arm.com/docs/DOC-8633

General IoT Architecture and Standards
http://community.arm.com/docs/DOC-8649

White papers and free downloadable developer version of mbed Device Server platform
http://community.arm.com/groups/smart-and-connected/blog/2014/05/07/white-papers-on-arm-iot-software