TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

ASE-9476 Factory Information Systems

Assignment 1: DPWS and REST deployment

16th of February, 2018

Objectives

1. Allowing students to work with actual industrial device
2. Provide a use case for working with DPWS and SOAP
3. Provide an understanding on RESTful services
4. Increase the students programming skills

Work description

As the ICT (Information and Communication Technology) evolves, Industrial section tends to adapt these advances in implementing manufacturing systems. This include information systems and networks. In this context, this assignment aims at consuming DPWS and RESTful service provided by service-enabled RTUs (Remote Terminal Units). The assignment requires the student to connect the pc to the S1000 via Ethernet socket. This will allow the students to communicate with S1000 and advance in the assignment.

The S1000 is configured to publish RESTful event each 5 second. Thus, the pc needs first to subscribe to the event. After that, the pc will start receiving the events containing the timestamp value. After that, the PC needs to extract the second digits and convert them to binary values. These binary values then will be send back the S1000 to change the outputs using the dedicated services. The students are required to use both REST and DPWS services for updating the output.

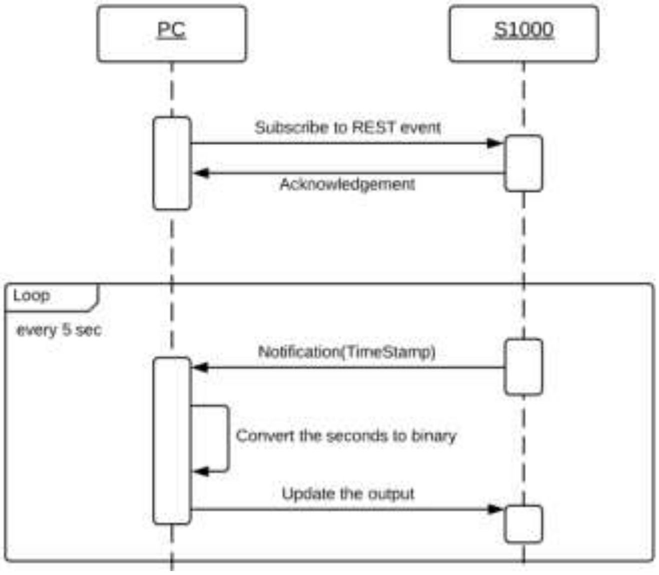

Figure 1. Sequence diagram of the assignment implementation

Steps

1. Turn ON the S1000.
2. Connect the S1000 to your PC.
3. Change the ip address to match the address that is written on the device. (see additional information section)
4. As operating systems have firewall, the TCP ports are by default closed. Therefore, you need to open the port where your nodejs application will listen to. To do so, see additional information section.

5. In case you are using body-parser module, don’t for get to fix the problem that was described in the lecture regarding the charset.
6. Now you are ready to go. You need first to use the web browser to see the rest event that you need to subscribe for. The link will be like this:

   http://192.168.100.xxx/rest/events

7. As an advice, use request module from npm to subscribe and invoke services.
8. After the application is subscribed, you need to build an http server where it will receive the event coming from the S1000. See additional information to learn how to subscribe to REST events.
9. Once you receive the events, you need to extract the second digits and convert them to binary values. After you need to send a request to the s1000 to change the outputs. The serves information is:
   a. Method: POST
   b. Link: http://192.168.100.xxx /rest/services/changeOutput
   c. Headers: 'Content-Type': 'application/json'
   d. Body: {"state0":true,"state1":true,"state2":true,"state3":true,"state4":true,"state5":true,"state6":true,"state7":true}
10. At this point, you can see the output changes in the front of the S1000.
11. You need to do the same thing but now using the DPWS and SOAP (for changing the output part only).
12. Start by reading the WSDL file in the S1000 using the following link: http://192.168.100.114:80/dpws/WS01
13. Then you need to construct the SOAP message to invoke the change output operation. This include the SOAP header and SOPA body.
14. Then you need to use the following message to send the message to the device
    a. Method: POST
    b. Link: http://192.168.100.xxx/dpws/WS01
    c. Headers:
       - accept: text/html,application/xhtml+xml,application/xml,text/xml;q=0.9,*/*;q=0.8
       - accept-encoding: none
       - accept-charset: utf-8
       - connection: close
       - host: 192.168.100.xxx:80
       - content-type: text/xml; charset=utf-8

```
<?xml version="1.0" encoding="ISO-8859-1"?>
 <s12:Envelope
         xmlns:s12="http://www.w3.org/2003/05/soap-envelope"
         xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
 <s12:Header>
         <wsa:Action>http://www.tut.fi/fast/Assignment/UpdateOutputs_Request</wsa:Action>
 </s12:Header>
 <s12:Body>
         <!—PUT YOU MESSAGE OF THE OUTPUTS HEREUSE THE WSDL FILE FOR THIS-->
 </s12:Body>
 </s12:Envelope>
```

Additional Information and Supporting documents

1. S1000 User Manual: http://www.inicotech.com/doc/S1000%20User%20Manual.pdf
2. FASTory description (use this document to learn how to subscribe to a REST event)
3. How to configure your IP address in your PC
   a. Windows: https://www.youtube.com/watch?v=BR1Z0jDauJ4
   b. Mac: https://www.youtube.com/watch?v=-l3l1KvRlTo
   c. Linux: https://www.youtube.com/watch?v=rTECO83sK_o
4. How to open a TCP port in your PC
   a. Windows: https://www.youtube.com/watch?v=cbFiWeeMUDI
   b. Mac: https://www.macworld.co.uk/how-to/mac-software/how-open-specific-ports-in-os-x-1010-firewall-3616405/
   c. Linux: https://www.youtube.com/watch?v=KGebpAPSeTs
5. In case you need to unsubscribe to an event, change the device from run to configure mode and then to run mode.

Tools and equipment

- INICO S1000 RTU
- NodeJS
- Wireshark
- Text editor
- Web browser

Questions

1. Describe the advantages and disadvantages of using HTTP-based protocols in industrial information systems? (Include in your response a reference to vertical and horizontal communications in the pyramid of automation)
2. Mention the importance of XSD when integrating systems
3. Explain what is parsing. In which part of the assignment do you use it?
4. In this assignment, are you using an Event Driven Architecture? In which part? Justify

5. Make a comparison between REST and SOAP (used in DPWS) Web Services. Generate a table (**pssst**!: this link (http://192.168.100.xxx/monitor/network.html) can help you to find out the number of packs sent and received.)

Deliverables

Each group should submit the following by **1st of March, 2018 23:59**:

- Zip file containing all developed code/scripts
- Report in PDF format (10 pages Max) including the following point:
  1. Code description
  2. Challenges and limitations
  3. Answers to the question in this document (questions' section)

It is important to be reminded that each group needs to present the results in person. This will be managed after the submission of the report. Further information will be sent via Moodle regarding this issue.

Grading

This assignment is graded out of 10 points.

Additional Notes

1 extra point as a bonus for the following additional requirements:

1. Build an HTML interface for the application showing the number of received events, the status of the outputs and the current time in the S1000 RTU
2. Error management is a very important topic when it comes to work with web services. Include a disconnection detection to you application so the user can be acknowledged if the S1000 is disconnected or turned off.

It is important to mention that the bonus marks will be ignored if the main requirement is not achieved.