**UML Physical View**

**Refrigerator**

**Server**

**User**

Database

Pc/Workstation

Computer / Workstation

TCP/IP

TCP/IP

**Internet**

Temp. Sensor

Monitor

Control module

Web Server

**Assume**: User can control system by using PC or via the internet

**UML Class Diagram**

Assuming system has a lot of sensor, 1 monitor and 1 control module.

Temp. Sensor

**UML Use Case**

- Current Temp.

- Current Temp.

- Sensor Id

- Sensor Id

- Current Temp.

- Sensor Id

- Current Temp.

Receives

Sends to

1

1

1

\*

\*

1

Control Module

Monitor

Database

|  |  |
| --- | --- |
| **Use-Case** | Adding temperature sensor |
| **Actors** | User, Server, Database and Router |
| **Description** | * In the beginning, power is supplied for temperature sensor module. User will press WPS button on the router which allow user auto connect to temperature sensor making LED to be blink mode, and then user press setup button. This button tells the unit to tell the server that a user is setting up a new temperature sensor. * In this time, the sensor module will send data including unit ID and setting temperature to user, after that this data continue to transfer to server module. In the server module, it will send status demand to clear the “setup” mode in sensor module, which makes the LED to be changed solid mode. After this process, temperature sensor will send data packages including current temperature, unit ID and new status to server module. In the end, server module will store data containing unit ID and current temperature in database if valid data returned to server form unit. |
| **Stimulus** | Connecting the sensor into the refrigerator for regulating temperature to maintain acceptable temperature for vaccine storage. |
| **Response** | For the stimulus information, with the current temperature confirmation, a corrective action is maintained for acceptable temperature or changed to acceptable temperature range. |

**HTTP GET STRING**

* **Assumption:**
* No real-time clock on sensors and monitor.
* Resolution temperature: 0.1˚ C.
* Min: -20 ˚C and Max: 50 ˚C, Δ = 70 ˚C
* 700 points needed, 10-bit number needed
* 12-bit 2’s complement signed number
* Temperature = 12-bit 2’s complement signed number
* STATUS = 4-bit binary number for any errors that may arise during operation.
* Unit ID:1 million units => 20-bit number
* **Assign values:**
* **Value 1 = 12-bit signed 2’s complement number**
* **Value 2 = 20-bit unsigned number**
* **Value 3 = 4-bit unsigned number**
* **Value 4 = 12-bit signed 2’s complement number**
* **Write HTTP URL:**
* Get data from temperature sensor to database:

<http://tempmonitoring.com/tempchecking?temp> **= value1&ID=value2&status=value3&tempset=value4**

* Get data from the server to the temperature sensor:

**Assume**: Controller ignores value1 and value3 when received value2 and value4

<http://tempmonitoring.com/tempset?temp> **=value2&status=value4&tempset**

**UML Interaction View:**

* Assuming the sensor will periodically wake up every one hour to sense temperature and send to the database. In the other case, the sensor also wake up to sense temperature when the refrigerator is open**,** then send the data to database so that server can control monitor to adjust the temperature inside.

