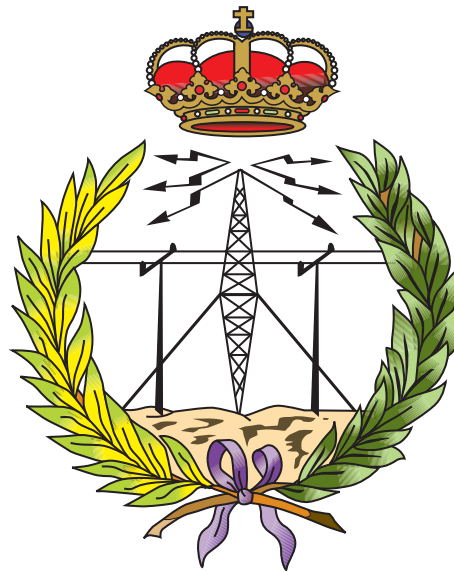


TECHNICAL UNIVERSITY OF MADRID
SCHOOL OF TELECOMMUNICATIONS SYSTEMS AND ENGINEERING

Semester Project



Bc. et Bc. Jaroslav Svoboda
Jaime Sancho
Diallo Elhadj Sadou

Smart Lighting

UBIQUITOUS AND SECURE NETWORKS AND SERVICES

Lecturer: Jesus Rodriguez Molina
Subject coordinator: Jose Fernan Martinez Ortega
Degree programme: Master in Systems and Services Engineering for the Information society

Madrid 2018

Contents

| | | |
|----------|---|-----------|
| 1 | Introduction and Executive Summary | 2 |
| 1.1 | Introduction | 2 |
| 1.2 | Benefits of smart lighting | 2 |
| 1.3 | Existing implementations | 2 |
| 1.3.1 | Chicago | 2 |
| 1.3.2 | Amsterdam | 2 |
| 1.3.3 | Project Implementation | 3 |
| 2 | Scheduling of workload | 4 |
| 3 | Requirement Analysis | 6 |
| 3.1 | General use | 6 |
| 3.2 | Receive Periodic Data | 7 |
| 3.3 | Request data | 8 |
| 3.4 | Change parameters | 9 |
| 4 | Project Design | 10 |
| 4.1 | UML Component Diagrams | 10 |
| 4.2 | UML Class Diagrams | 11 |
| 4.3 | Sequence Diagrams | 14 |
| 4.4 | Protocol diagram | 18 |
| 5 | Testing | 19 |
| 6 | Conclusion | 24 |

1. Introduction and Executive Summary

1.1 Introduction

Smart or intelligent lighting is part of the recent IoT and Smart Cities trend. It is being installed in many cities of developed world improving the safety and efficiency of cities and expanding the services provided to citizens by cities. Our project demonstrates simple system of smart lighting which reacts to movement.

1.2 Benefits of smart lighting

Main benefits are:

- Improved Service
- Improved Safety
- Energy Savings
- Reduced light pollution

New installations of smart lighting system in vast majority of cases makes use of LED lights which have improved lifetime period. This lowers costs of lightbulbs and also workforce needed for bulb replacements.

Because the lights dim or turn off, they reduce the needed energy saving costs and environment. This with better light design helps to lower light pollution which is a waste of energy and also has negative impact on fauna. More progress in this area can be achieved by regulation to prevent unnecessary lighting and smart planning of street lighting.

In areas with low quality or reliability of lighting can newly installed lighting system lower the crime level especially when used with camera system.

1.3 Existing implementations

1.3.1 Chicago

In the largest city of Midwest lives approximately 2.7 million people on area of 606 km². This area has 353000 of outdoor lighting fixtures which in 92 % of cases use High-Pressure Sodium (HPS) bulbs. Majority of light poles were installed in 1950's and 60's. City of Chicago decided to renovate its lighting system in fall 2017. Project goal is to replace 270000 of existing HPS light fixtures with new energy-efficient LED lights which are now installed in only 2 % of all street lights. City will be also able to use new lighting management system enabling real-time monitoring, control of the fixtures and in future support smart city applications. This will help to save 50 % of the city's lighting electric bill, that is approximately €8.5 million every year. In terms of saved energy, the estimation is 181 679 358 kWh. City also hopes to lower the use of 311 line. The project should finish in 2021 and use at least 50 % of local workforce. The program is part of Better Building Initiative which aim is to convert 1.5 million of outdoor lights nation-wide. [2, 3]

1.3.2 Amsterdam

In 2012 cities of Amsterdam, Rotterdam and Eindhoven decided to create a trial of Smart Lights in Metropolitan Areas project. Amsterdam decided to create safe and pleasant living and operating environment for newly renovated square Hoekenrodeplein in Amsterdam's Zuidooost district. The city collaborated with Cisco, A2, Phillips, Alliander and KPN. Installed system combines adaptive lighting, cameras and public Wi-Fi. Each of 144 LED lights has its own IP

address and can be controlled individually or automatically adapt thanks to sensors. The lights can not only change intensity but also change their colour. The saved energy is used for Wi-Fi network and air quality measuring.[4, 1]

1.3.3 Project Implementation

For the project implementation we are using Sun SPOT JAVA Development Kit with 4 motes. Mote characteristics are:

- 2.4 GHz IEEE 802.15.4 radio with an integrated antenna and a USB interface
- ARM architecture 32 bit CPU with ARM920T@180 MHz, 512 kB RAM and 4 MB flash memory.
- mote board includes a three-axis accelerometer, temperature sensor, light sensor, 8 tricolor LEDs, analogue and digital inputs, two momentary switches, and 4 high current output pins.
- unit uses 3.7 V rechargeable 750 mA h lithium-ion battery, has 30 μ A deep sleep mode and battery management is provided by software.

The software was developed on virtual machines running Ubuntu 10.04, Solarium simulation environment and Eclipse IDE.

2. Scheduling of workload

Table 2.1: My caption

| TASK | AUTHOR | WORK HOURS |
|--|----------------|------------|
| T1. Contextualization of the project | All | 1 h |
| T2. Requirements analysis | All | 5 h |
| T3. Design | All | 5 h |
| T3.1 Use case diagrams | All | 3 h |
| T3.2 Sequence diagrams | Diallo, Chaime | 3 h |
| T3.3 Components diagrams | | |
| T3.3.1 Components diagrams: normal and light sensor | Jaime | 1 h |
| T3.3.1 Components diagrams: aggregator | Jaroslav | 1 h |
| T3.5 Class diagrams | | |
| T3.5.1 Class diagrams: normal and light sensor | Jaime | 1 h |
| T3.5.1 Class diagrams: aggregator | Jaroslav | 1 h |
| T3.6 Other diagrams: Protocol diagram | Jaime | 1 h |
| T4. Implementation | | |
| T4.1 Implementation of MIDLET in mote 1: normal sensor | Jaime | 3 h |
| T4.2 Implementation of MIDLET in mote 2: light sensor | Jaime | 3 h |
| T4.4 Implementation of aggregation-oriented MIDLET | Jaroslav | 5 h |
| T4.4 Implementation of user interface | Diallo | 6 h |
| T5. Testing | Jaime | 4 h |
| T6. Document edition | All | 5 h |

3. Requirement Analysis

3.1 General use

Figure 3.1: General use

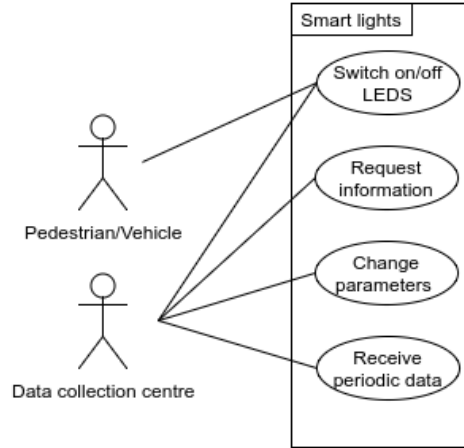


Table 3.1: General use

| | |
|----------------------|---|
| Scope and Objectives | The lights on the lampposts switch on/off depending on the information of the movement and light sensor of the nodes (if the light is low and the movement sensor detects something, the lights turn on, otherwise they turn off). The data collection centre receives periodically this data and it is capable to change some parameters or request some information of the nodes. |
| Actors | <ul style="list-style-type: none"> • Pedestrians/vehicles • Data control department of the city |
| Preconditions | <ul style="list-style-type: none"> • The sensor nodes are on • A pedestrian/vehicle comes/goes away from the node |
| Post-conditions | <ul style="list-style-type: none"> • The lights will turn on/off • All the measured information will be sent to the city data manager |
| Sequence Description | <ol style="list-style-type: none"> 1. Obtain measurements of movement, temperature, humidity and light. 2. If there is a change in the movement sensor and the light measured is low, toggle the lights on the node. 3. Send the status of light to the data control department of the city. |
| Exceptions | The node is not able to communicate with the data control department of the city. |

3.2 Receive Periodic Data

Figure 3.2: Receive Periodic Data

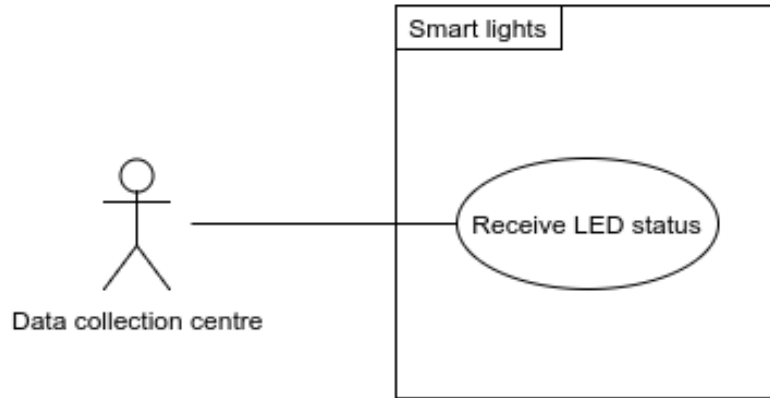


Table 3.2: Receive Periodic Data

| | |
|----------------------|--|
| Scope and Objectives | The nodes are placed in the lampposts of the street getting information of the movement and the light. This information added to the LED status of the lamppost is sent to the data collection centre periodically. |
| Actors | Data control department of the city |
| Preconditions | <ul style="list-style-type: none"> • The sensor nodes are on • The sensor nodes have sufficient memory to store measured values when connection is off |
| Post-conditions | <ul style="list-style-type: none"> • The measured value is stored in memory • The measured value is sent to the data control department of the city |
| Sequence Description | <ol style="list-style-type: none"> 1. Obtain measurement 2. Check the network 3. If network not available store data 4. Send all data to the data control department of the city |
| Exceptions | In case of error, report to the data control department of the city |

3.3 Request data

Figure 3.3: Request Data

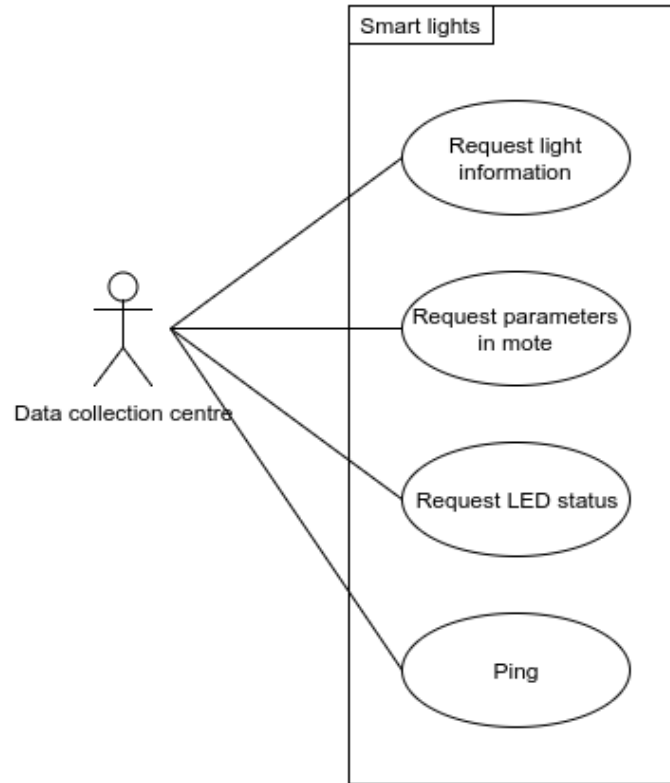


Table 3.3: Request Data

| | |
|----------------------|---|
| Scope and Objectives | The data collection centre have the possibility to request the information of the light and LED status of the motes and also the parameters that are using the motes. Furthermore, a ping functionality exists to check if the mote is alive. |
| Actors | Data control department of the city |
| Preconditions | <ul style="list-style-type: none">• The sensor nodes are on• The sensor node is online to get the request |
| Post-conditions | The measured value and statuses are sent to the data control department of the city |
| Sequence Description | <ol style="list-style-type: none">1. Check if node is online2. If not then repeat x times3. Request current data from node |
| Exceptions | The sensor node does not respond after x requests |

3.4 Change parameters

Figure 3.4: Change parameters

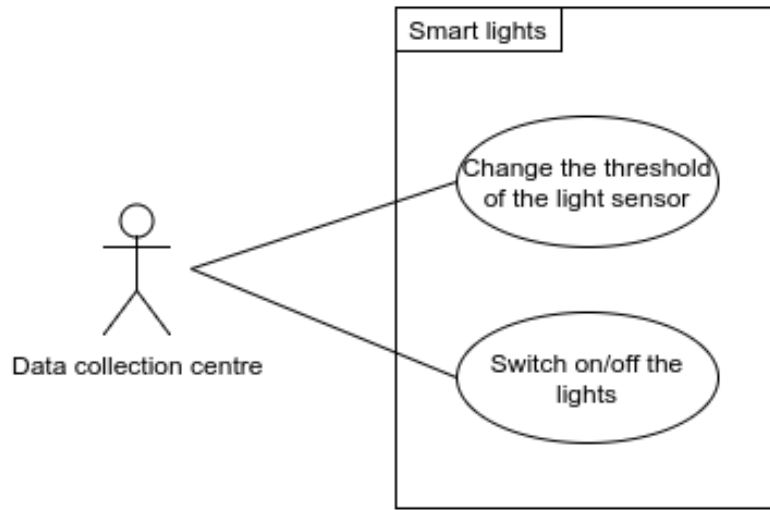


Table 3.4: Change parameters

| | |
|----------------------|---|
| Scope and Objectives | The data collection center is able to change the threshold of the light sensor and also has the possibility of switch on/off the lights in case of emergency. |
| Actors | Data control department of the city |
| Preconditions | <ul style="list-style-type: none"> • The sensor nodes are on • The sensor node is online to get the request • Led light is working |
| Post-conditions | <ul style="list-style-type: none"> • The light is turned on/off • Parameters are modified • The status is sent to the data control department of the city |
| Sequence Description | <ol style="list-style-type: none"> 1. Check if node is online 2. If not then repeat x times 3. Request switching on/off the light or change of parameters 4. Send ack when light turns on or parameters are changed |
| Exceptions | <ul style="list-style-type: none"> • The sensor node does not respond after x requests • The light does not turn on/off |

4. Project Design

4.1 UML Component Diagrams

Figure 4.1: Movement mote

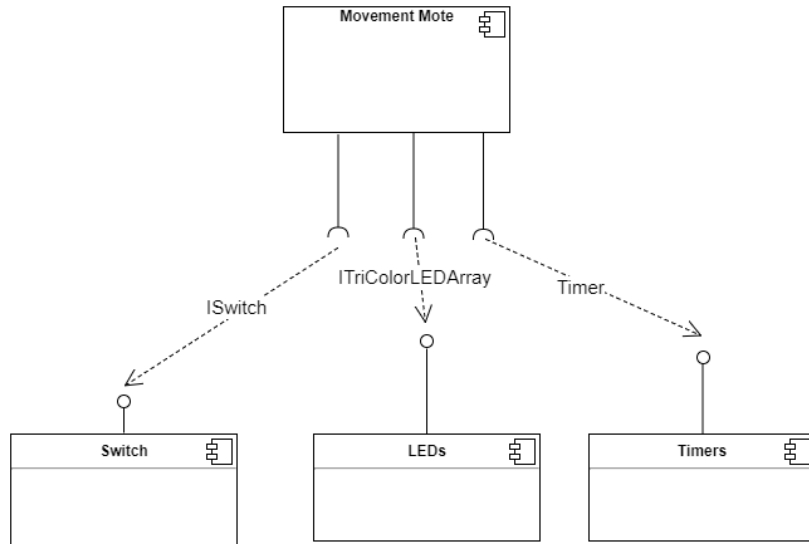


Figure 4.2: Light mote

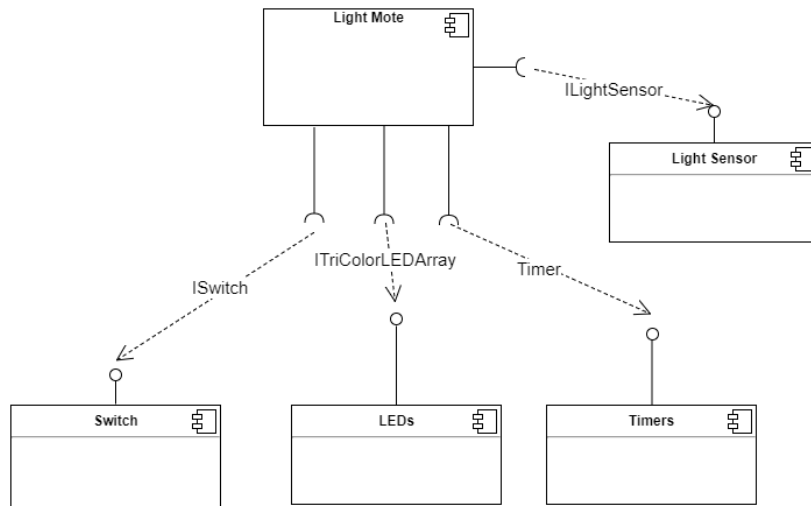


Figure 4.3: Data collection centre

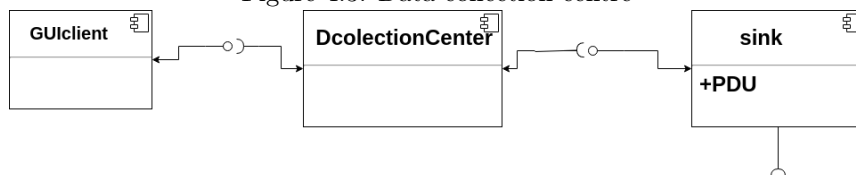
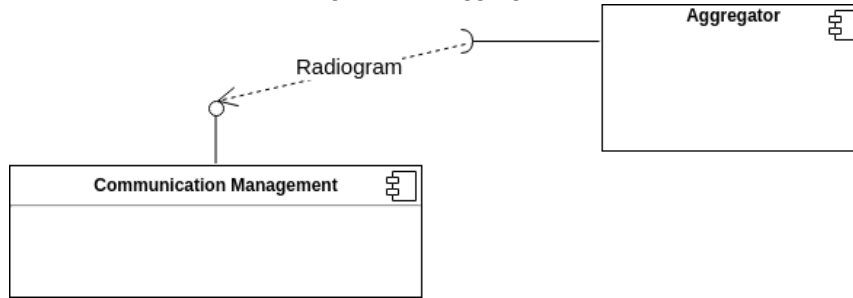


Figure 4.4: Aggregator



4.2 UML Class Diagrams

Figure 4.5: Movement mote

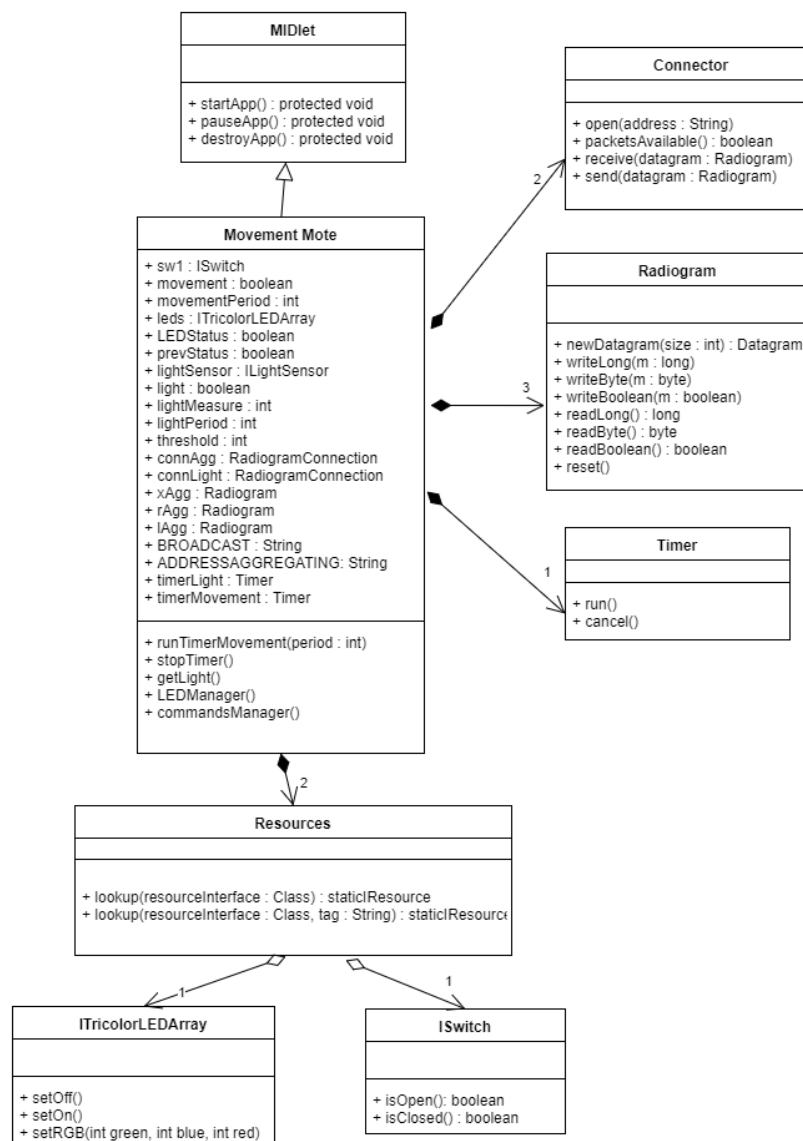


Figure 4.6: Light mote

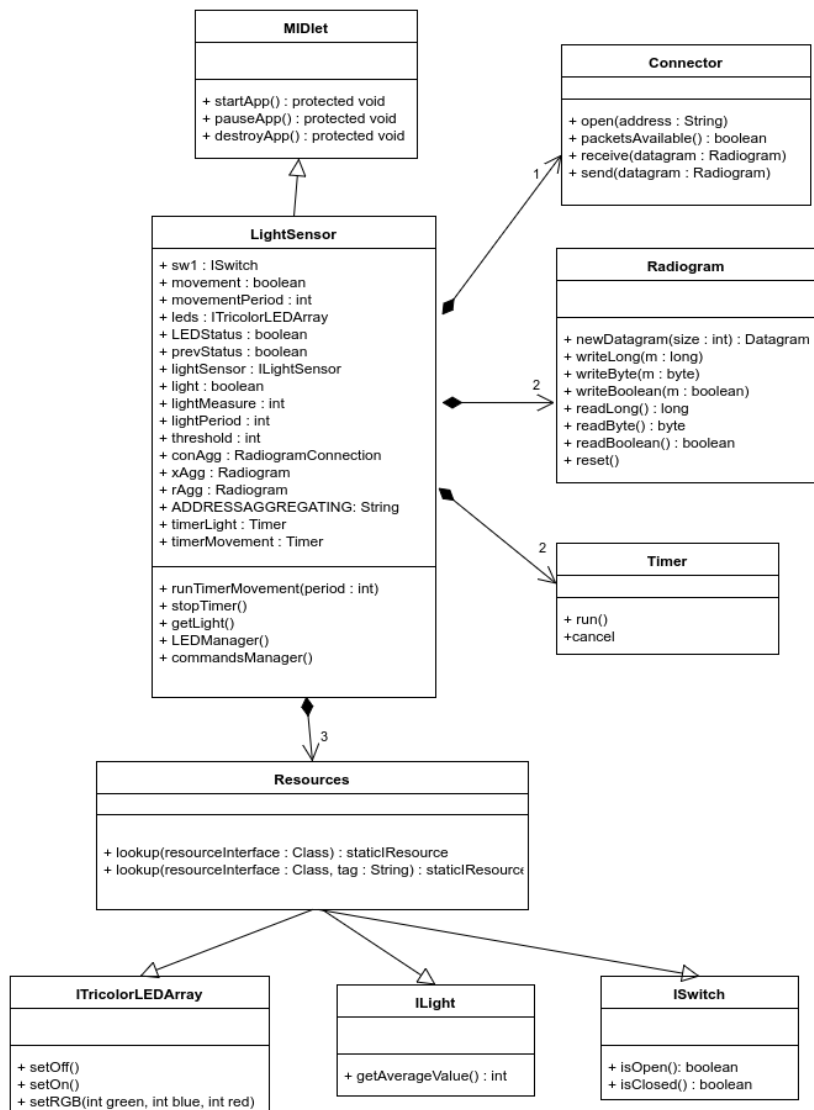


Figure 4.7: Sink

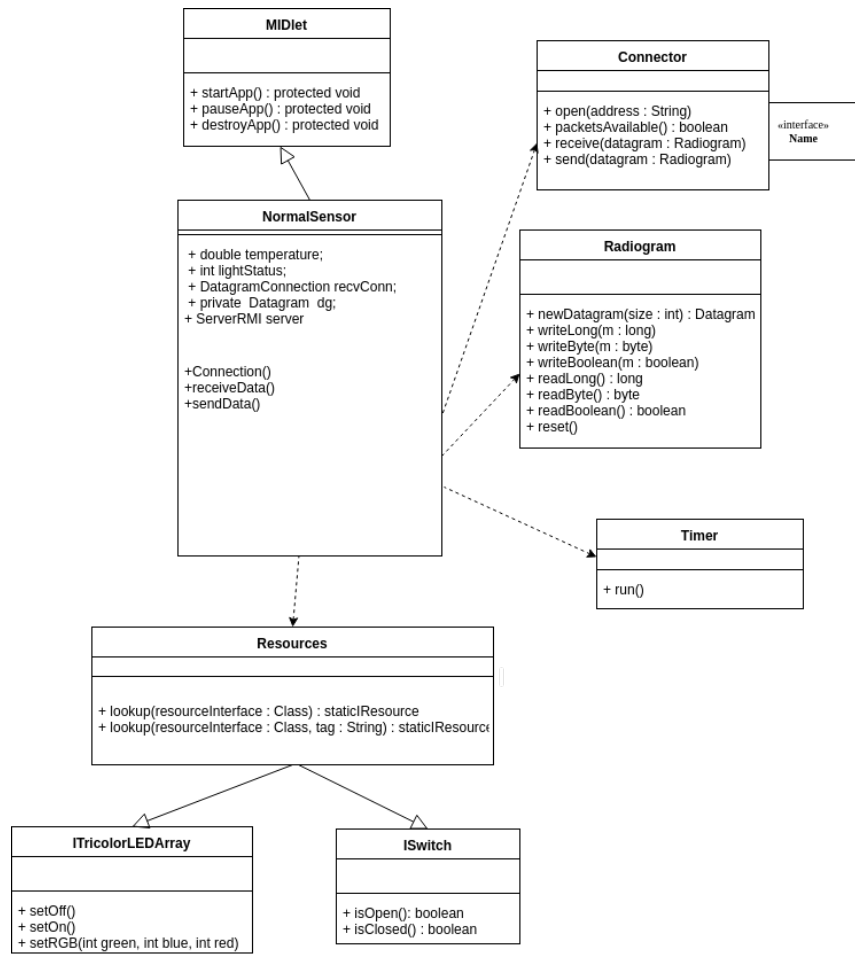
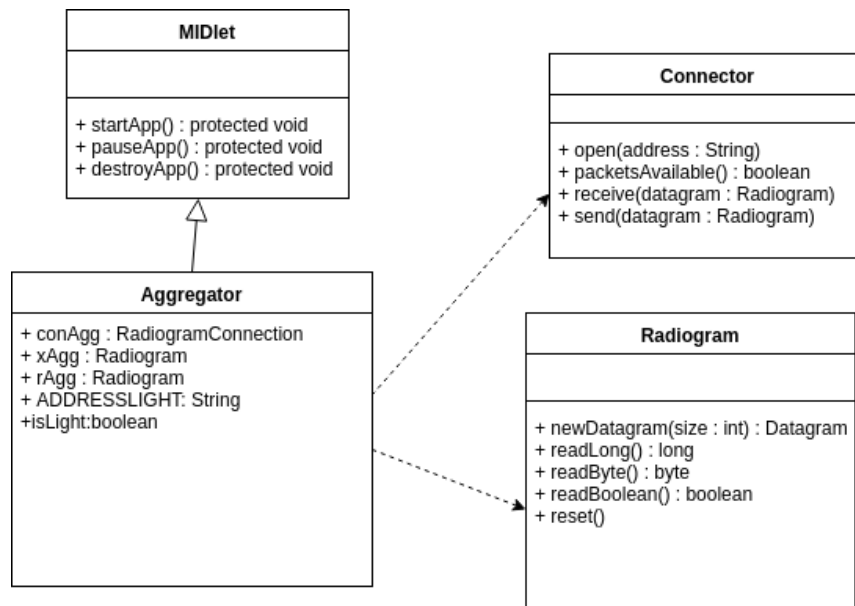


Figure 4.8: Aggregator



4.3 Sequence Diagrams

Figure 4.9: Light Broadcast

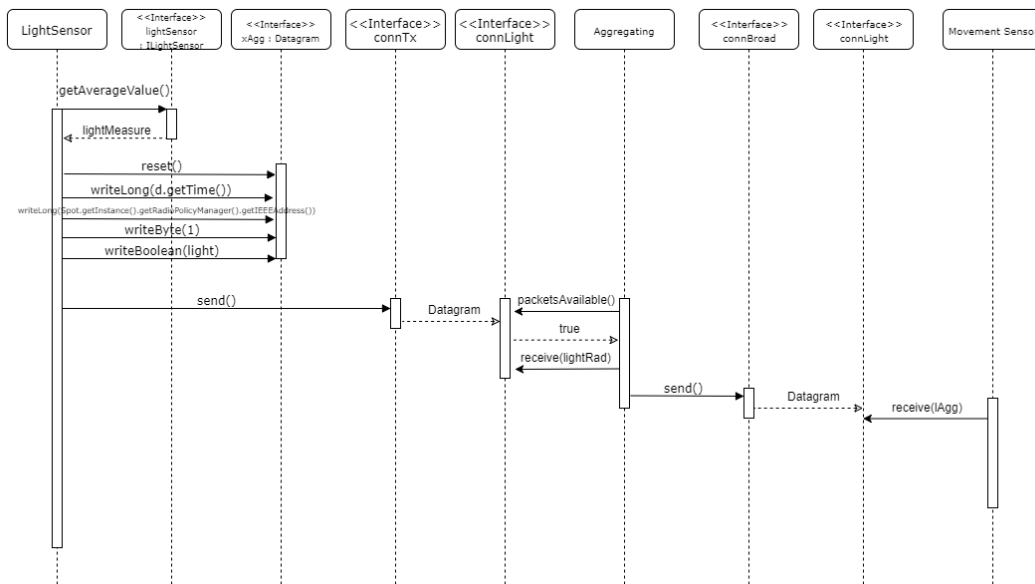


Figure 4.10: Example of use of protocol

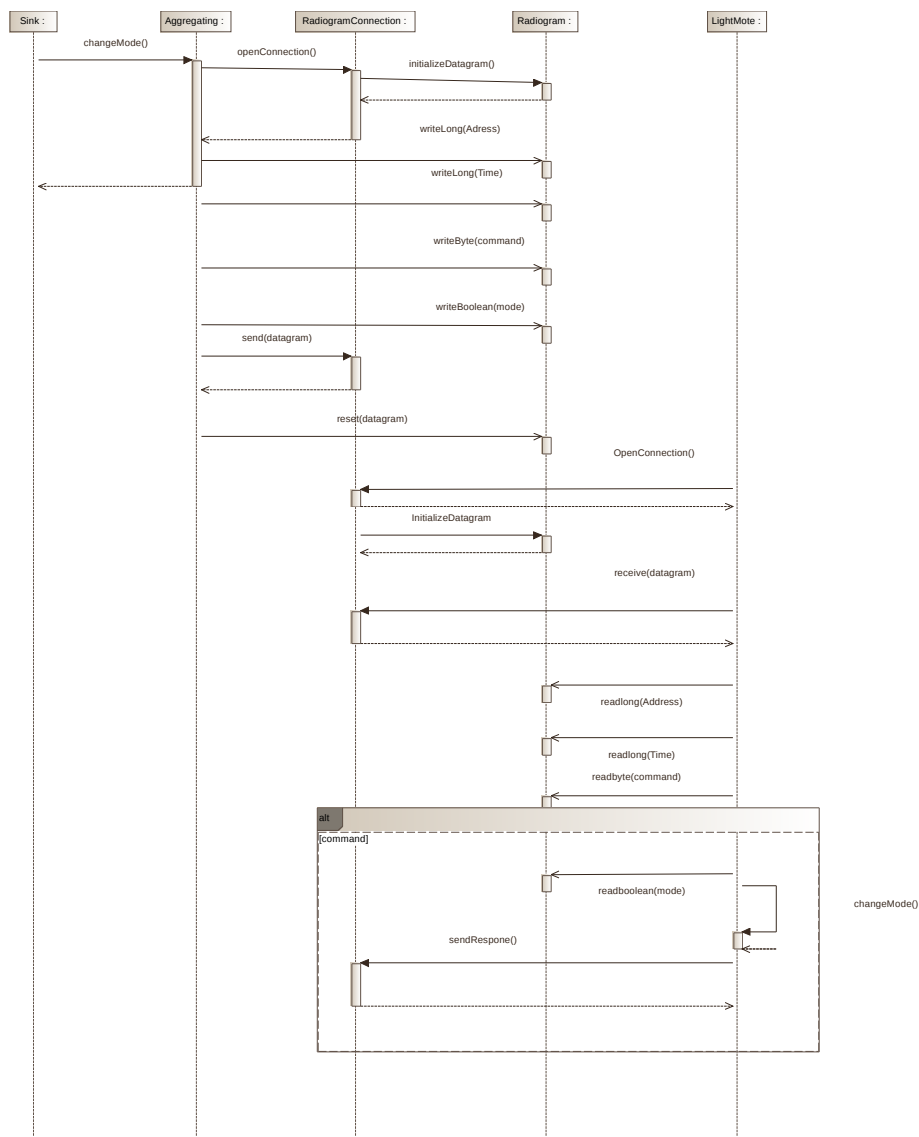


Figure 4.11: Base station

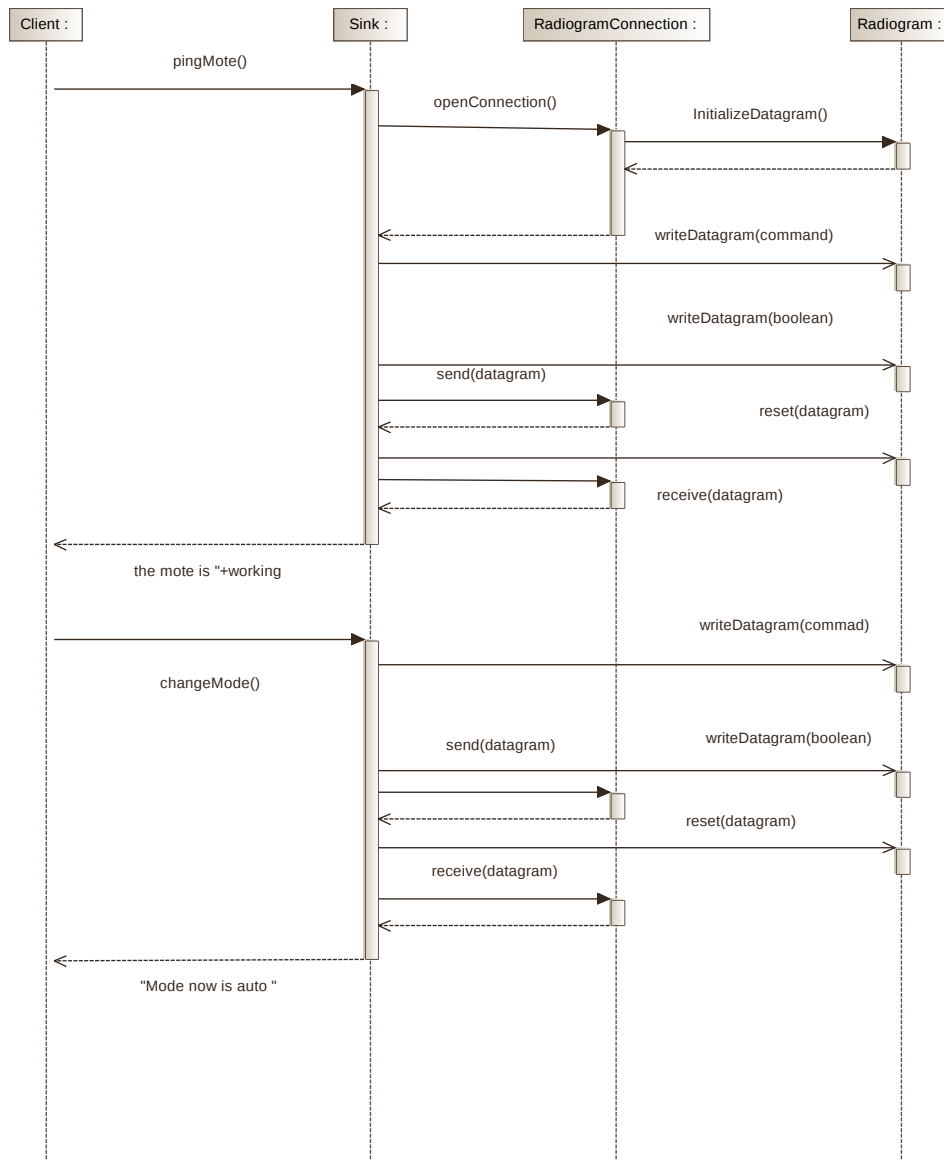
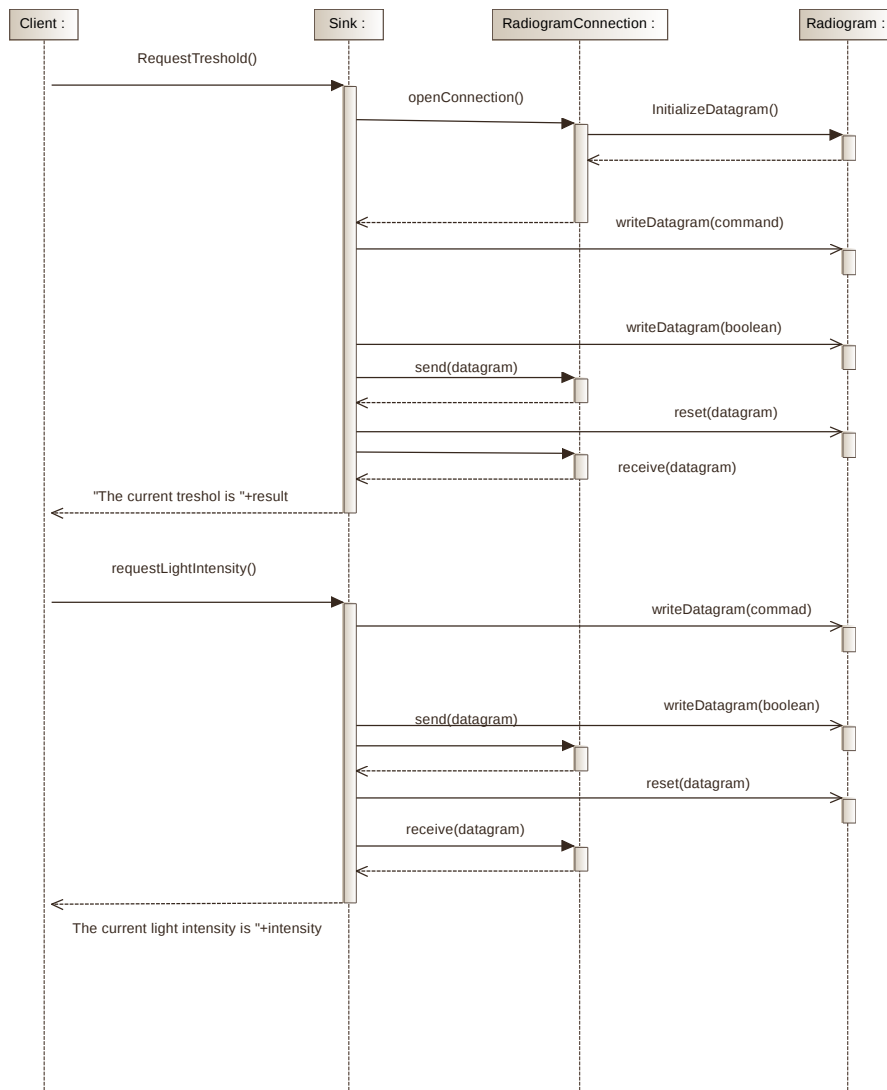
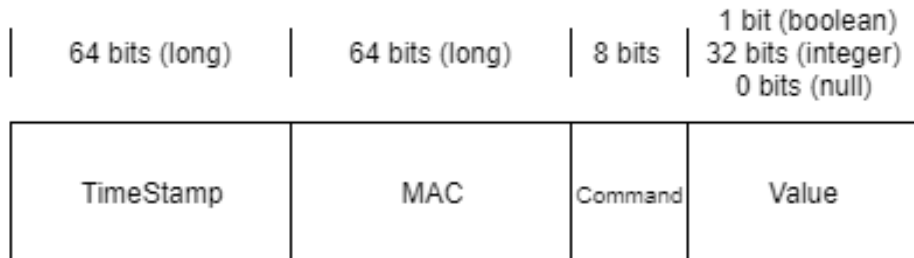


Figure 4.12: Base station with light intensity sensor



4.4 Protocol diagram

Figure 4.13: Protocol diagram



Commands:

1. Light intensity (true/false) send or receive
2. Led status whenever it changes (true/false) or change led status from
3. Request/send the value of light intensity threshold
4. Request/send the current value of light intensity
5. Receive ping, answer ping (null)
6. Special state (sink changes the state of the LEDs) (true/false)

5. Testing

In order to test the project, in first place the Java Solarium Emulator was used. Once the project was working well, it was tested on the real motes using the graphical interface. For doing that, three motes has been used: one for the light sensor, other for the movement sensor and another one for the aggregating. Furthermore, the base station was used. The following tests were done in order to verify the functioning of the project (the MAC address selector shows the MAC of the motes in Solarium, the first corresponds with the light mote and the second with the movement mote):

Figure 5.1: Motes used



Figure 5.2: Graphical interface

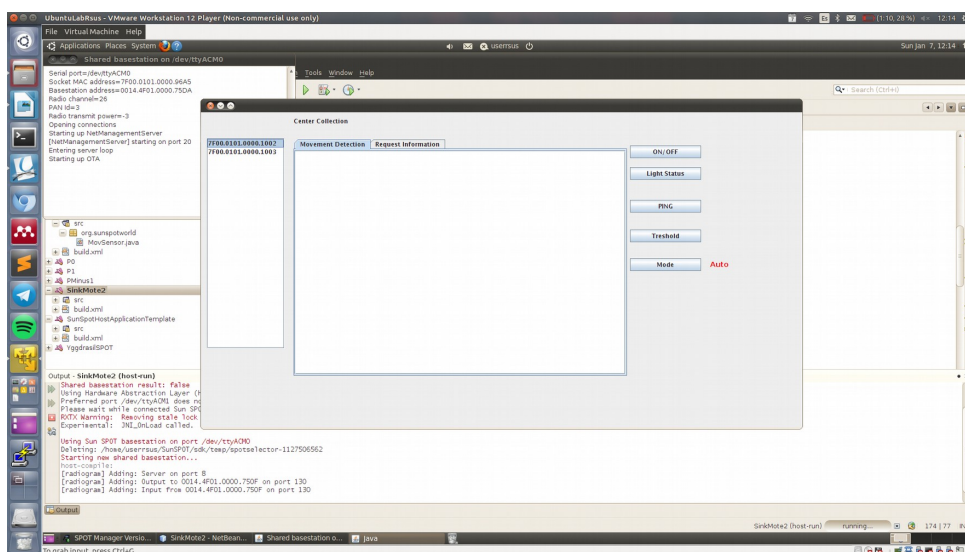


Figure 5.5



Figure 5.6: If the mode is set again to Automatic, it is possible to see the change of the light status whenever the switch (simulating a movement sensor) is pressed. The information registered includes the MAC of the mote, and the time.

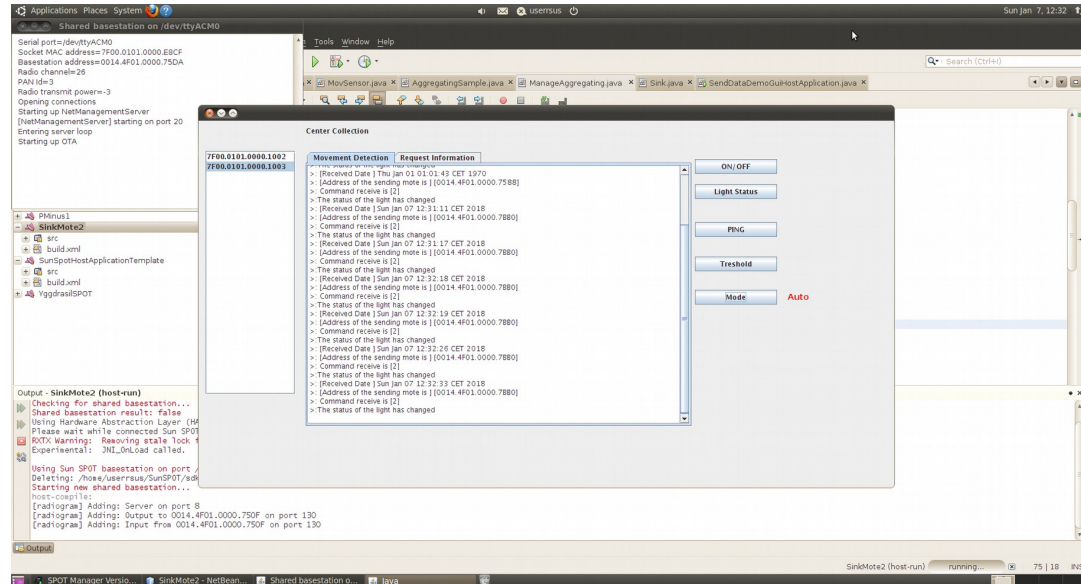


Figure 5.7

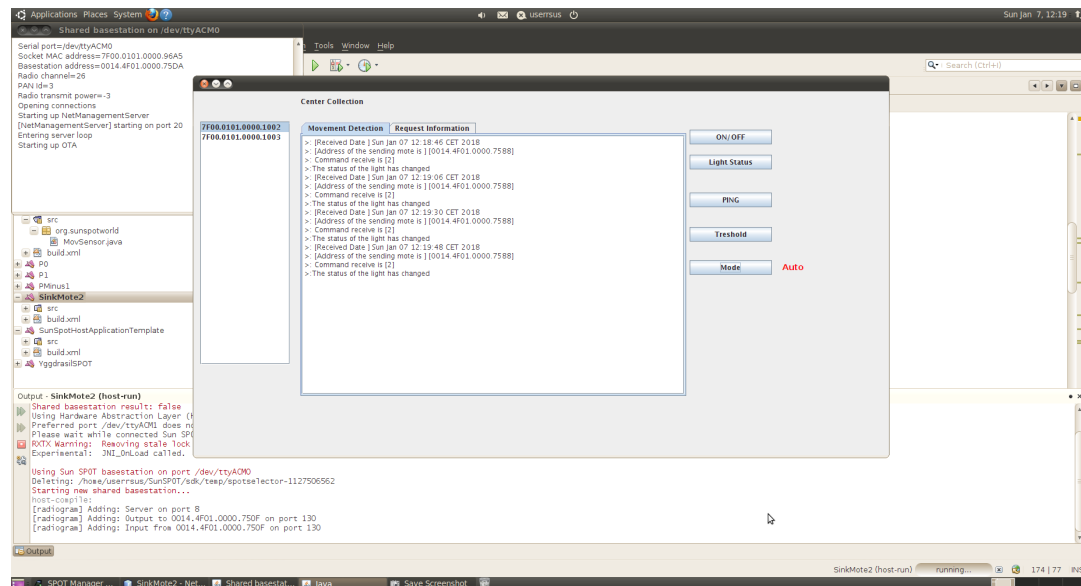


Figure 5.8



6. Conclusion

In this project we created a smart lighting system which reacts dynamically on movement of vehicles or pedestrians, detects night and day, and is also managed remotely using GUI. Operator can turn the lights on or off, check if the mote works, get current light intensity, change the threshold for night and day mode and also get the statistical data for use analysis; everything of course remotely using JAVA application.

The system, if implemented, could be vastly improved by better newer motes which would not have to be wireless. The system might either be connected to metro-optical network or use power-line communication. As a product the system would be probably an indivisible part of public lighting and might incorporate other sensors and functions as explained in real life examples.

Bibliography

- [1] Lucy Fisher and Nikita Junagade. *Smart Lighting for Hoekenrodeplein square*. English. LUCI Association. Aug. 2016. URL: <http://www.luciassociation.org/smart-lighting-for-amsterdams-hoekenrodeplein-square/> (visited on 14/12/2017).
- [2] Rose Jordan. *Chicago Smart Lighting Project. Implementation Model*. English. Version initial. Midwest Energy Efficiency Alliance. 31st Mar. 2017. URL: https://betterbuildingssolutioncenter.energy.gov/sites/default/files/tools/Chicago_Smart_Lighting_Project_MEEA.pdf (visited on 14/12/2017).
- [3] Unknown. *Chicago Smart Lighting Program*. English. 17th Sept. 2017. URL: <http://chicago.maps.arcgis.com/sharing/rest/content/items/d8eaff2a1d0f4e0a834773cd792a5ae7/data> (visited on 14/12/2017).
- [4] Joyce van Veen and Pim Stevens. *Smart Light*. English. Amsterdam Smart City. 7th Sept. 2017. URL: <https://amsterdamsmartcity.com/projects/smart-light> (visited on 14/12/2017).