CTIS - 477 System Engineering Project

Project Design Report

Ömer Levent Durdalı

21702600

Revision History

Date	Description	Author	Comments
18/03/2021	Version 0.1.0	Levent Durdalı	Initial Document

Table of Contents

Table of Figures

1. Identification

The projects name is, Smart Life and Home Security System, this document will refer to the project with following abbreviation; **SLS**.
SLS is a small Smart Home System that manages and tracks curtains, lights, temperature and motion in a given area.

2. Scope

2.1. System Overview

SLS's main purpose is to be a Decision support system for a home user, meaning the system can help with managing room temperature, adjust light levels and lastly helps the user to adjust the curtains in the specified room. The system will have 3 different sensors that is connected to a central hub. This hub will communicate with a personalized cloud server using the MQTT communication protocol. The end-user will be able to see the gathered data and control the light level and curtain positions from the local web dashboard panel.

2.2. Document Overview

The rest of this document gives the detailed specifications for SLS. It is organized as follows:

- **Section 3: System Wide Design Decisions**
  This section will present decisions that affect the system in any way.
- **Section 4: System Architectural Design**
  This section the defines the structure, behaviour, components and different levels of configurations of the system.
- **Section 5: System Models**
  This section will present Component, Package and Deployment diagrams of the system.
- **Section 6: Requirement Traceability**
  This section shows the traceability of the requirements.

2.3. Literature and Referenced Documents

[1]. Make-It.ca, "NodeMCU ESP8266 Detailed Review," [Online]. Available: https://www.makeit.ca/nodemcu-arduino/nodemcu-details-specifications/. [Accessed 3 March 2021].

[2]. Calder, Bruce, and Sara Santos. "Getting Started with Node-RED ". Random Nerd Tutorials, 2021, https://randomnerdtutorials.com/getting-started-with-node-red-dashboard/. [Accessed 18 Mar 2021].

3. System Wide Design Decisions

SLS provides a dashboard to observe the current condition of the specified area by showing the level of light, temperature and humidity. Also, in case it is dark SLS can automatically do two things if it is night time it opens the lights or if it is morning it opens the curtains by using the motor. SLS can also automatically toggle the light by using the motion sensor connected to the system.

There were 2 major design choices in design this system.

The first decision is NodeRED[2] sever since this is a hardware-software system the software to use this hardware should have been fully tested ore be fully custom written. As the size of the system isn't that big, I have chosen NodeRED as the main control panel and configuration software to control the hardware, it should be noted that this software doesn't fully control the sensor or device is acts as an interface for control and sensors.

The second decision was choosing ESP8266[1] development board as the main microcontrollers. The reasons for these are the following, it is very cheap and comes with an integrated Wi-Fi chipset so it can easily be configured to connected to Wi-Fi.

In terms of the user perspective these design choices offer, three main things, a cheap, flexible and maintainable (NodeRED has a very large community and its very modular) solution. The user will be able to open / close the motor and lights connected to the ESP8266 microcontroller.

3.1. Safety

The ESP8266 and the used devices are all widely used and generally well device to the intended job, so using well receive part will offer a complete safety net for the entire system.

3.2. Privacy and Security

The NodeRED controller is an opensource tool which is being developed by people around the globe as such it offers a very robust privacy setting and polices, they do not collect any user information except for system diagnostics. Along with these abilities NodeRED also offer secure communication with your preferred client and the microcontroller.

3.3. Component Choices

As stated in the Initial Document and in the first paragraph of section 3 of this document, the following component will be used

Name	Intended use	Material	Advisory Info	Voltage
ESP8266	Microcontroller	Standard PCB	-	5V
LDR	Light Level Measurement	Semiconductor Material	Need capacitor for use	3,3V to 5V
LED	Light	PCB, Silicon Lens		1,5V to 3V
Servo Motor	Assumed to be connected to a curtain. Then the motor is	Electromagnet, Plastic	Limited precision control, may need additional power supply	5V

	used open and close it			
Motion Sensor	Alert and Open Light	PCB, Infrared Diode, Lens	-	3,3V
Temperature and Humidity Sensor	Humidity and Temperature Measurement	PCB, Plastic, Thermistor	Verify working conditions	3,3V

3.4. System Behaviour

In terms of user perspective, the NodeRED will receive inputs from the user in form of buttons displayed in a dashboard. Any button pressed will control the microcontroller and the rest of the dashboard will show near real-time or average outputs sent by the microcontroller. This communication will be done with using MQTT.

The sensor reading will be stored in the NodeRED server and since this is simply sensor values not security is need to protect this, but the NodeRED configuration should be made so that only the server itself can write and modify the data.

3.5. Critical Requirements

The most important requirement is that the NodeRED's dashboard should be capable of displaying and controlling the defied components. If this does not work the server architecture will either be altered or completely change.

4. System Architectural Design

Modes	**Description**
Inactive	In inactive mode, the system is completely off. So, no connection can be made.
Ready	In ready mode, the system is booted up and started reading the sensors and completed its connection phase.
Active	In active mode, the system is performing the intended takes stated in this document.

4.1. System Components and Relations

Competent Identifier	Component Name
SLS-COM-01	NodeMCU V2 ESP8266 Development Board
SLS-COM-02	5 mm LDR Sensor
SLS-COM-03	DHT11 Temperature and Humidity Sensor
SLS-COM-04	HC-SR501 PIR Motion Detector
SLS-COM-05	5mm LED
SLS-COM-06	Tower Pro SG90 RC Mini (9gr) Servo Motor

- ESP8266 Development Board is a type of microcontroller and is the main component in the system.
- SLS-COM-2 trough SLS-COM-6 will be directly connected to SLS-COM-01.
- The algorithms that will run the all components will be executed and stored by SLS-COM-01

Competent Identifier	Purpose	Requirements ID
SLS-COM-01	Controller of the system	SRQ-SLS-02, SRQ-SLS-03,
SLS-COM-02	Will be used for checking light level in the area.	SRQ-SLS-01, SRQ-SLS-05, SRQ-SLS-06, SRQ-SLS-07,
SLS-COM-03	Will be used for checking the temperature and humidity of the area.	SRQ-SLS-01, SRQ-SLS-10, SRQ-SLS-11
SLS-COM-04	Will be used to automatically open and close the light. And show and alert of movement.	SRQ-SLS-01, SRQ-SLS-07
SLS-COM-05	Will be used as a light source for the user to open and close.	SRQ-SLS-05, SRQ-SLS-08
SLS-COM-06	Assumed to have control over curtains in a house. So, with this the user can open and close the curations by control the motor.	SRQ-SLS-06, SRQ-SLS-09

4.2. Concept of Execution

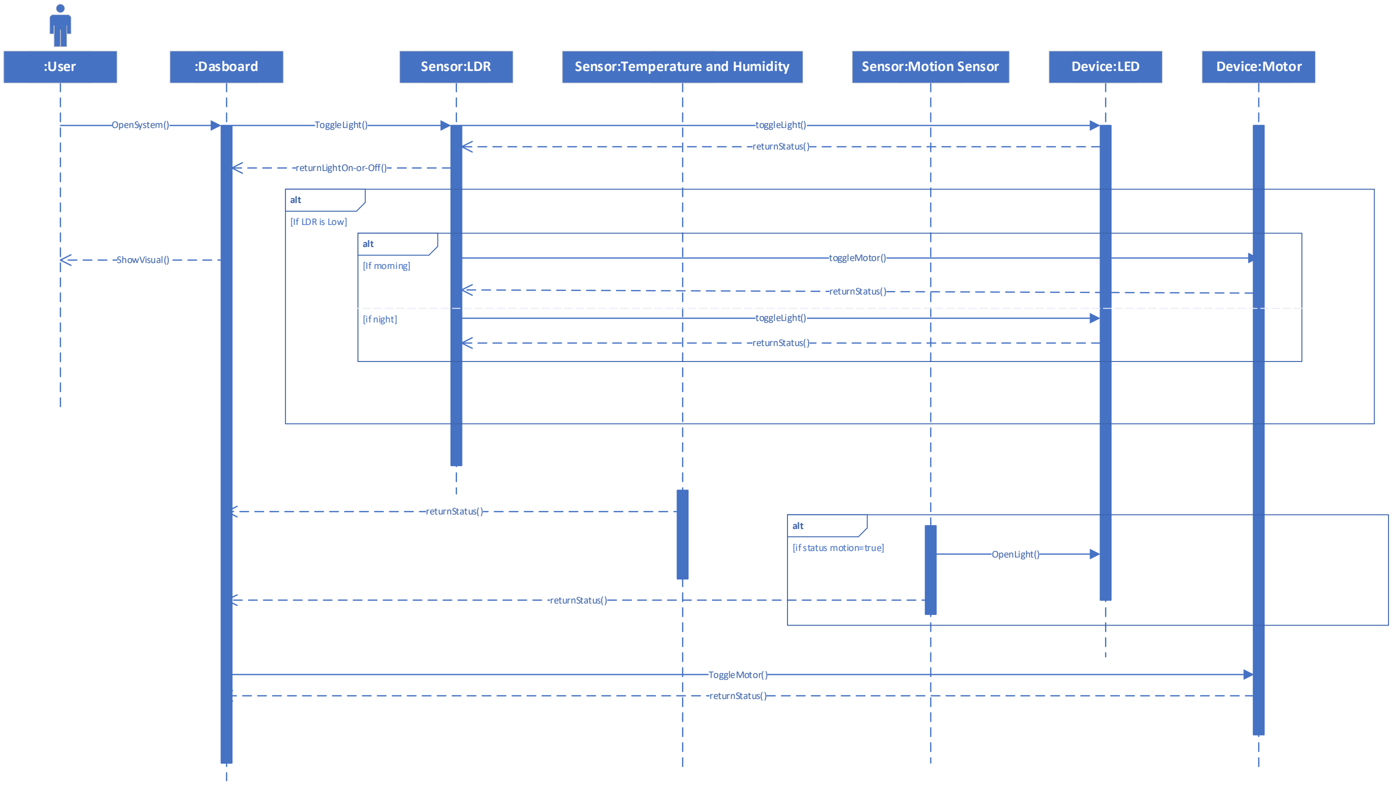

Figure 1: Sequence Diagram

SLS provides a dashboard to observe the current condition of the specified area by showing the level of light, temperature and humidity. Also, in case it is dark, which is known by the light level sensor, SLS can automatically do two things if it is night time it opens the lights or if it is morning it opens the curtains by using the motor. SLS can also automatically toggle the light by using the motion sensor connected to the system.

After the system is connected to a power source SLS will change its mode from Inactive to Ready.

After SLS's mode is Ready, SLS will start its connections and then start reading the sensors values. While reading these value SLS will also start sending these values over MQTT and change its mode to Active.

Finally, after its in active mode, SLS will be able to send MQTT data to the NodeRED Dashboard and also by the use of the NodeRED Dashboard SLS will be able to send commands to the connected microcontroller.

4.3. Interface Design

All external and internal interfaces are to be left to the development and test document.

5. System Models

5.1. Component Diagram

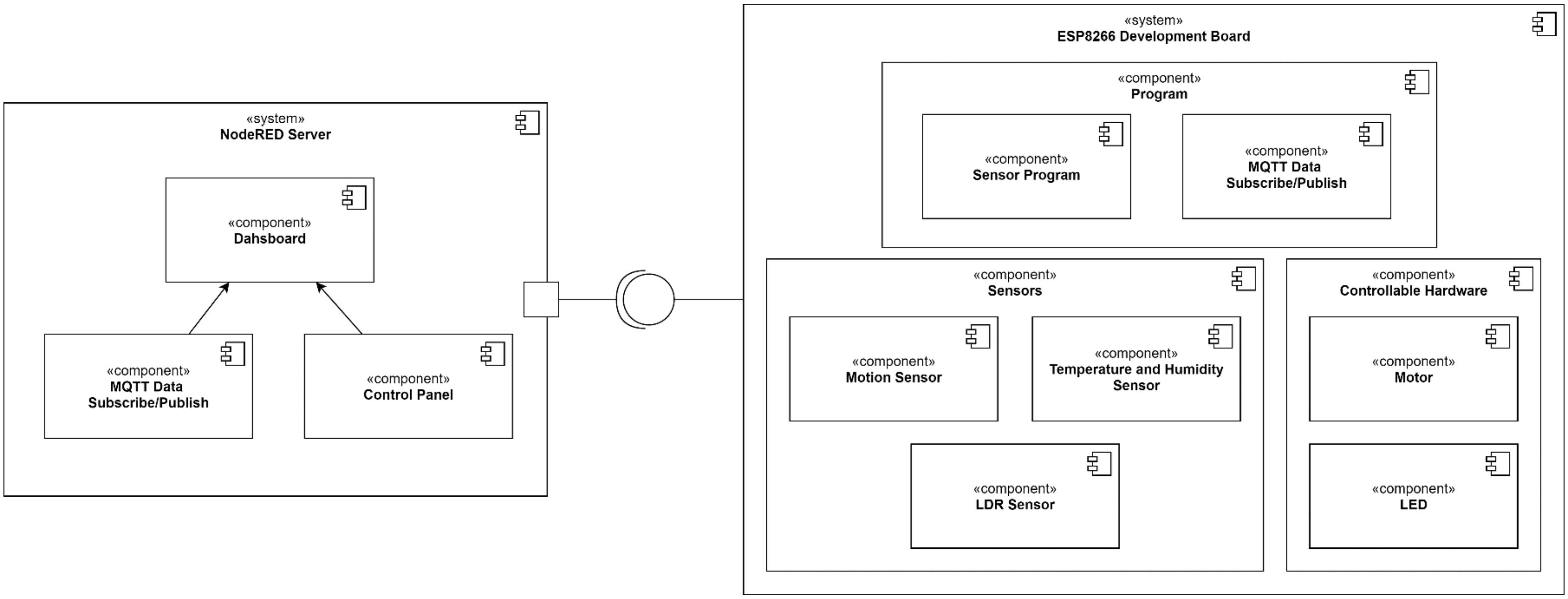

Figure 2: Component Diagram

5.2. Deployment Diagram

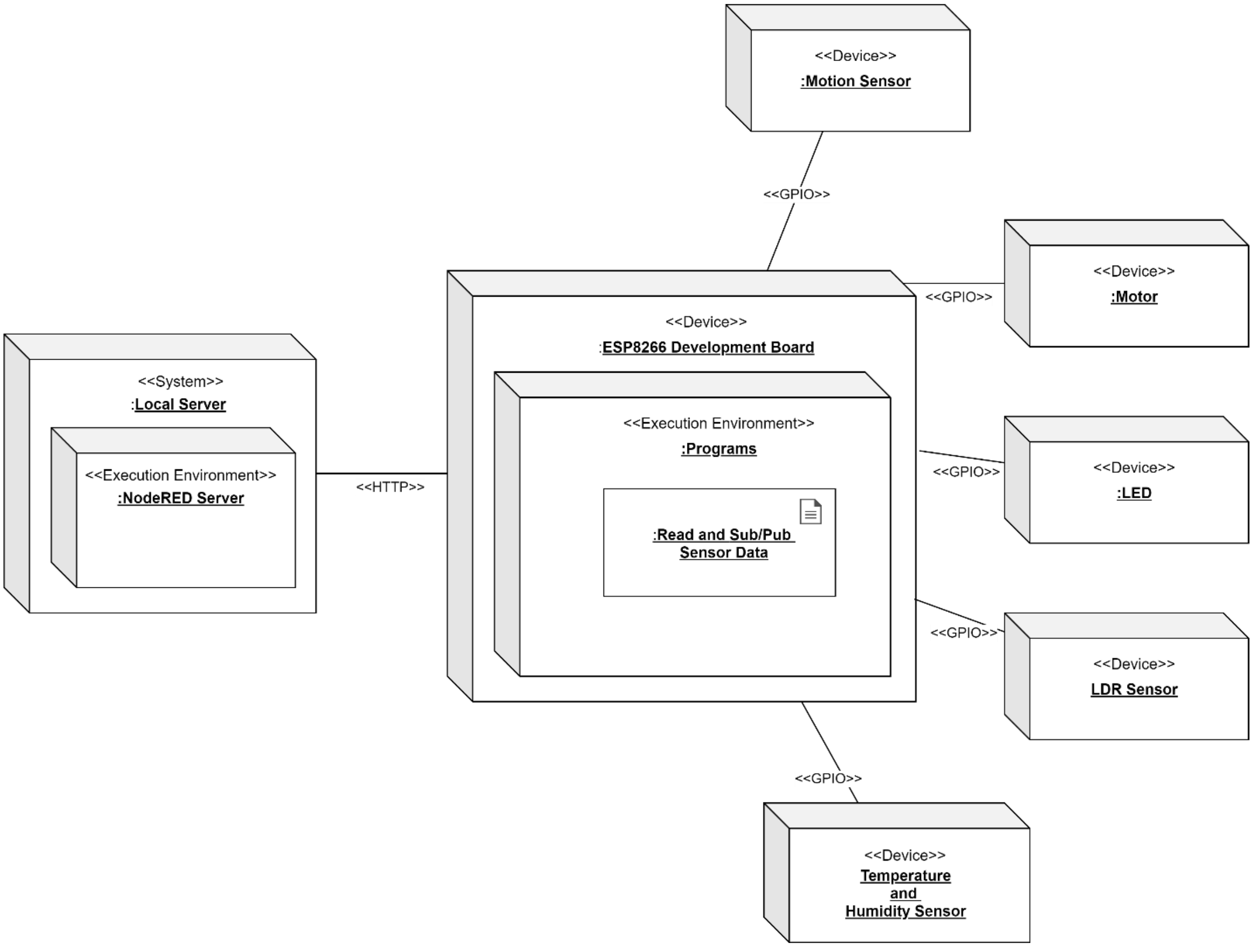

Figure 3: Deployment Diagram

5.3. Package Diagram

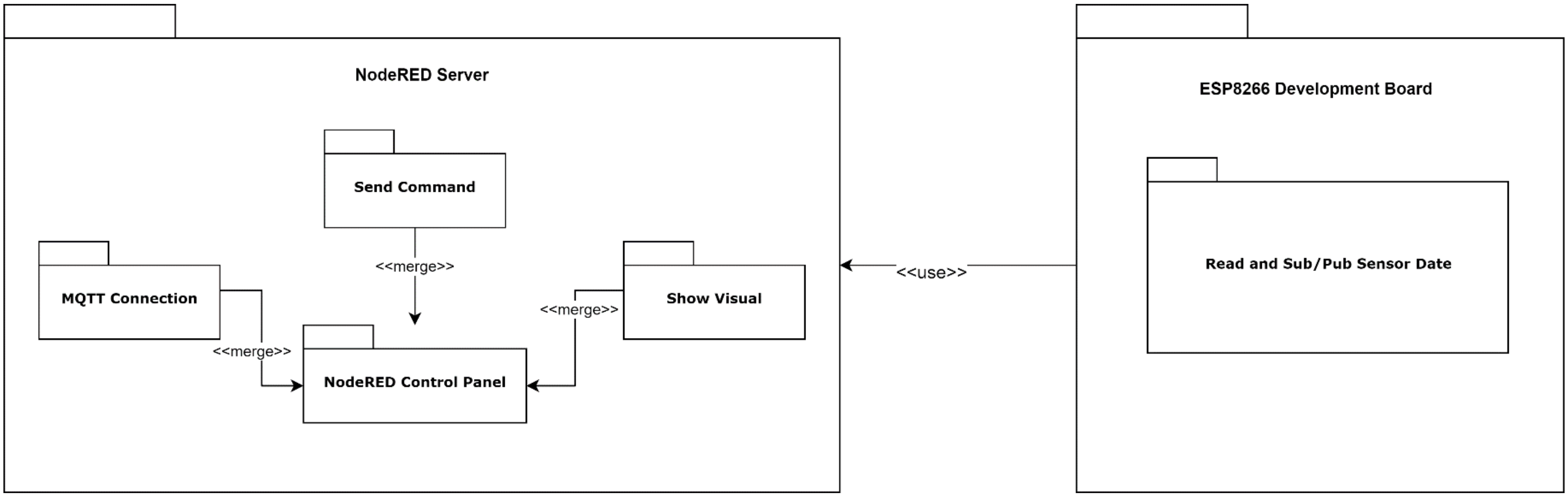

Figure 4: Package Diagram

6. Requirement Traceability

ID	User Requirement ID	Requirement ID	Requirement Name	Shot Description	Verification	Validation	Revision in Document
1	USR-SLS-01	SRQ-SLS-01	Get Sensor Info	Gather sensor data	Having readable values	System Testing	V0.1.0
2	USR-SLS-01	SRQ-SLS-02	Communication	Remote Usage	-	System Testing	V0.1.0
3	USR-SLS-02	SRQ-SLS-03	Send Command	Remote Usage	Change system function	System Testing	V0.1.0
4	USR-SLS-01	SRQ-SLS-04	Show Visual	Sensor output visualization	Having a working dashboard	Integration Testing	V0.1.0
5	USR-SLS-02	SRQ-SLS-05	Auto-Toggle Light	Self-toggle light	Change system function	Integration Testing	V0.1.0
6	USR-SLS-02	SRQ-SLS-06	Auto-Toggle Motor	Self-toggle motor	Change system function	Integration Testing	V0.1.0
7	USR-SLS-02	SRQ-SLS-07	Motion Detection	Alert for motion	Change system function	Integration Testing	V0.1.0
8	USR-SLS-02	SRQ-SLS-08	Toggle Light	User toggle light	Send command and toggle light	Integration Testing	V0.1.0
9	USR-SLS-02	SRQ-SLS-09	Toggle Motor	User toggle motor	Send command and toggle motor	Integration Testing	V0.1.0
10	USR-SLS-01	SRQ-SLS-010	Read Temperatures	Gather temperature data	Validating device output	Component Testing	V0.1.0
11	USR-SLS-01	SRQ-SLS-011	Read Humidity	Gather humidity data	Validating device output	Component Testing	V0.1.0

12	USR-SLS-03	SRQ-SLS-012	Login	Login to dashboard	Login to the system with valid credentials	Component Testing	V0.1.0