Light

Service Description

Abstract
This document provides service description for the **Light** service.

Contents

1 Overview

This document describes the **Light** service, which enables basic control of light emission within the lamp system. The primary purpose of this service is to switch the light on or off. It provides a simple interface for toggling the light state. This service focuses on binary control, with no adjustments for intensity or color, ensuring reliable and straightforward operation.

The rest of this document is organized as follows. In Section 2, we describe the abstract message operations provided by the service. In Section 3, we end the document by presenting the data types used by the mentioned operations.

1.1 Significant Prior Art

The light is generated by an efficient LED, which is very energy efficient and can shine very brightly. The power source is a standard AA battery with a minimum voltage of 1.5V.

1.2 How This Service Is Meant to Be Used

The given system should consume the service. Use the controls provided by the service to toggle the state, whether light should be emitted or not.

1.3 Important Delimitations and Dependencies

The service's primary function is to provide illumination for a single room, with no additional features or capabilities. It is designed specifically for use as a stationary lamp within a room, though it can be repositioned within the same space. The light does not support dimming or color adjustment. This simplifies the requirements for the AI system, which only needs to determine whether the light should be switched on or off.

- It does not support dimming
- It does not support transitions
- It does not support colors
- It does not support movement

1.4 Access policy

Available for anyone within the local cloud.

2 Service Interface

This section describes the interfaces to the Light service. The service is used to control the emission of light. In particular, each subsection names an abstract operation, an input type and an output type, in that order. The input type is named inside parentheses, while a colon precedes the output type. Input and output types are only denoted when accepted or returned, respectively, by the interface in question. All abstract data types named in the section are defined in Section 3.

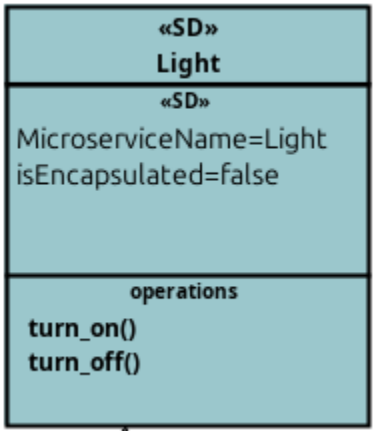

Figure 1: SysML block description diagram of the Light and its interface

The following interface operations are available.

2.1 operation turn off (LightstateForm) : LightstateResult

The turn_off operation is used to turn the light off. The services will contain various metadata as well as a physical endpoint. The parameters are representing the necessary service input information.

2.2 operation turn on (LightstateForm) : LightstateResult

The turn_on operation is used to turn the light on. The services will contain various metadata as well as a physical endpoint. The parameters are representing the necessary service input information.

3 Information Model

Here, all data objects that can be part of the **Light** Service provided to the hosting System are listed in alphabetic order. Note that each subsection, which describes one type of object, begins with the *struct* keyword, which is used to denote a collection of named fields, each with its own data type. As a complement to the explicitly defined types in this section, there is also a list of implicit primitive types in Section 3.3, which are used to represent things like hashes and identifiers.

3.1 struct LightstateForm

Field	Type	Mandatory	Description
lightEmission	Boolean	yes	State of light emission. Default false

3.2 struct LightstateResult

Field	Type	Description
emissionOld	Boolean	State of light emission before change.
emissionNew	Boolean	State of light emission after change.

3.3 Primitives

Types and structures mentioned throughout this document that are assumed to be available to implementations of this service. The concrete interpretations of each of these types and structures must be provided by any IDD document claiming to implement this service.

Type	Description
Address	A string representation of the address
Boolean	One out of `true` or `false`.
Interface	Any suitable type chosen by the implementor of the service.
DateTime	Pinpoints a specific moment in time.
List<A>	An *array* of a known number of items, each having type A.
Name	A string identifier that is intended to be both human and machine-readable.
PortNumber	Decimal number in the range of 0-65535
Version	Specifies a service version.

4 References

5 Revision History

5.1 Amendments

No.	Date	Version	Subject of Amendments	Author
1	2024-10-12	4.6.2		Max Lütkemeyer

5.2 Quality Assurance

No.	Date	Version	Approved by
1	2024-10-12	4.6.2	