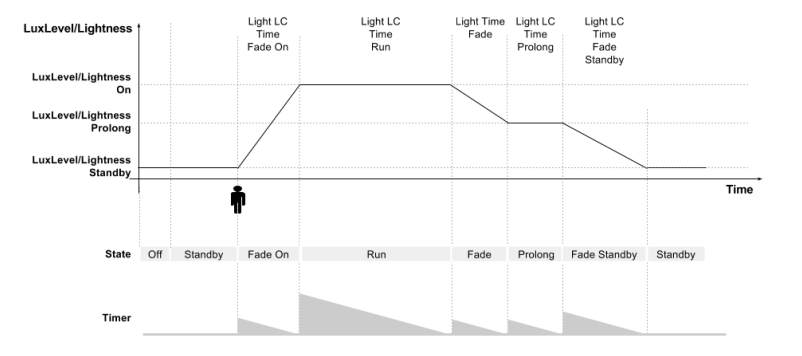
## Purpose

The purpose of this training is clearly described in the [confluence page](https://confluence.silabs.com/display/SalesTraining/2020+Q2+FAE+Training?focusedCommentId=120858018&#comment-120858018), we should focus on the new lighting model(s). So, I think the Light Control Model (LC model) should be a good topic to be covered in this hands-on.

## Introduction

The LC model is introduced and defined in the Mesh Model Specification and was first supported by BT Mesh 1.6.0 SDK. The definition of the model in the SPEC is complex, but in general, it aims at smart lighting. It implements a state machine to tune the lightness level based on the environment status, so basically it could work without the human intervention. Below is a picture from the Model SPEC which describes how it works.



### Necessary Inputs for LC server feature demonstration

* Ambient light – which is an important feedback to feed the PI regulator so that the state machine could tune the lightness output against it.
* Occupancy state – Ignore the lightness adjustment, basically the state machine will turn on the light if there is occupancy reported, turn off the light when no occupancy for a certain time or the occupancy is cleared explicitly.

## Devices & Roles

Essentially, only one device could work for demo purpose, but to make the concept clearer, I would suggest following the typical setup – one provisioner (smartphone App), one switch (LC client) and one light (LC server).

**Smartphone with Bluetooth Mesh App**

This will be acting as the provisioner, its responsibilities:

* Create network and group (appkey)
* Provision the light and switch node to the network and configure them properly.
* Adjust the LC states and the LC properties, which could reflect the behavior changes for the state machine to tune the lightness.

**Light – LC Server**

Responsibilities

* It’s the most intuitional part to indicate if the hands-on is successful or not – the lightness should change according to the ambient light and occupancy sensors’ status.
* Output some data to indicating the change on lightness linearly, which could indicate how the state machine is regulating the lightness.

**Switch – LC Client**

In the setup, the switch is not actually functional as a switch to control the lightbulbs on the light node. Instead, it sends out the sensor messages according to the sensor status or manual input, which will result in the LC server to tune the lightness based on the sensor status.

Responsibilities:

* [TODO] Because there isn’t an occupancy sensor on the Thunderboard sense BG22, we need to figure out the best way to simulate it. Currently, one option is to use the buttons on the board, one to increment the occupancy state value (e.g. people count), the other one to decrement.
* There is an ambient light sensor on it, so it can measure the ambient light and send result to the light node periodically.

## Items need to be done by the trainees

**Smartphone App (provisioner)**

* No coding is required.
* Provision the light and switch node
* Configure the PUB/SUB if applicable, if they decide to use unicast, then this is not necessary
* Change the LC states and properties to see how it reflects in the behaviors that the state machine tune the light.
  + Lightness level on each state.
  + Ambient level on each state.

**Light node**

* Add some logging information to indicate the change of the light status. The light example already has logging, but it would be very hard to use because it’s too redundant.

**Switch node**

* Ambient/occupancy sensor sampling, this is not the scope of BT Mesh model or BT Mesh itself, maybe we can provide the code directly.
* Sending the sensor status to the light node.