Inside a Bluetooth mesh lighting network

a comprehensive guide for lighting manufacturers

INTEGRATION WITH COMPONENTS

GOOD PRACTICES AND NEW POSSIBILITIES

NEW APPROACH TO LIGHTING CONTROL

SILVAIR

silvair.com

Table of Contents

Introduction

This guide explains how component manufacturers and luminaire makers can integrate Bluetooth mesh within their products, and what good practices should be followed when implementing Bluetooth mesh lighting networks. It also outlines some of the most innovative concepts underlying Bluetooth mesh networking, as well as the essentials of decentralized lighting control introduced by this wireless technology. Last but not least, it gives a glimpse at advanced lighting control strategies included in the Bluetooth mesh specification.

Since its adoption in 2017, the Bluetooth mesh standard continues to gain momentum in professional lighting applications. Designed specifically to address the challenging requirements of wireless lighting control, it is the first low-power communication technology that solves the three pain points of today's connected lighting systems: reliability, scalability and simplicity. It also democratizes lighting control, allowing manufacturers of all sizes and geographies to join a globally interoperable ecosystem of devices that work with each other out of the box. Learn how you can put Bluetooth mesh into your products, and what makes this technology such a powerful engine for wireless lighting control systems.

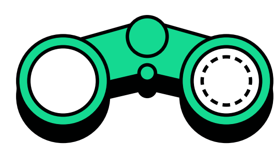
Software – a new lighting component

Before a device can become part of a Bluetooth mesh lighting network, it needs to implement the core Bluetooth mesh software. This is where all fundamental networking principles are defined, enabling a device to join the network and communicate with other network nodes. In addition, it needs to implement the so-called models. They are defined in the Bluetooth Mesh Model specification. By defining functionalities of network nodes, mesh models standardize the operation of lighting and sensing devices within a Bluetooth mesh network. This standardization is one of the pillars of the global interoperability of Bluetooth mesh.

A core Bluetooth mesh stack with relevant mesh models implemented on top of it is what we call complete lighting firmware. Such firmware has to be deployed into a device in order to make it "smart". This is done using the system-on-chip (SoC) technology. An SoC consists of a radio transceiver and a microcontroller. The latter contains the firmware necessary to turn an LED lighting or sensing component into a Bluetooth mesh enabled device.

Despite its miniature size, an SoC is a complex and multi-component computing platform. It integrates multiple specialized chips and functions into a single piece of silicon. This concept has set new standards in computer miniaturization, driving the smartphone revolution and paving the way for new applications. With the advent of the LED era, SoCs entered also the lighting component market. And they are here to stay, so let's get a bit more familiar with their basics.

There are several ways in which manufacturers can transfer relevant Bluetooth mesh firmware onto SoCs that are to be integrated with their products. This can be done at the chip supplier's production line, but there are other alternatives. At Silvair, we've designed and built the **Silvair Manufacturing and Testing Environment (MaTE)**. It is a system that allows our customers to conveniently download the Silvair Firmware from the Silvair manufacturing cloud, activate the license, and configure the settings in accordance with their needs. We believe this gives our partners maximum flexibility in terms of production logistics and planning.



" With the advent of the LED era, SoCs entered also the lighting component market. And they are here to stay"

1 CPU

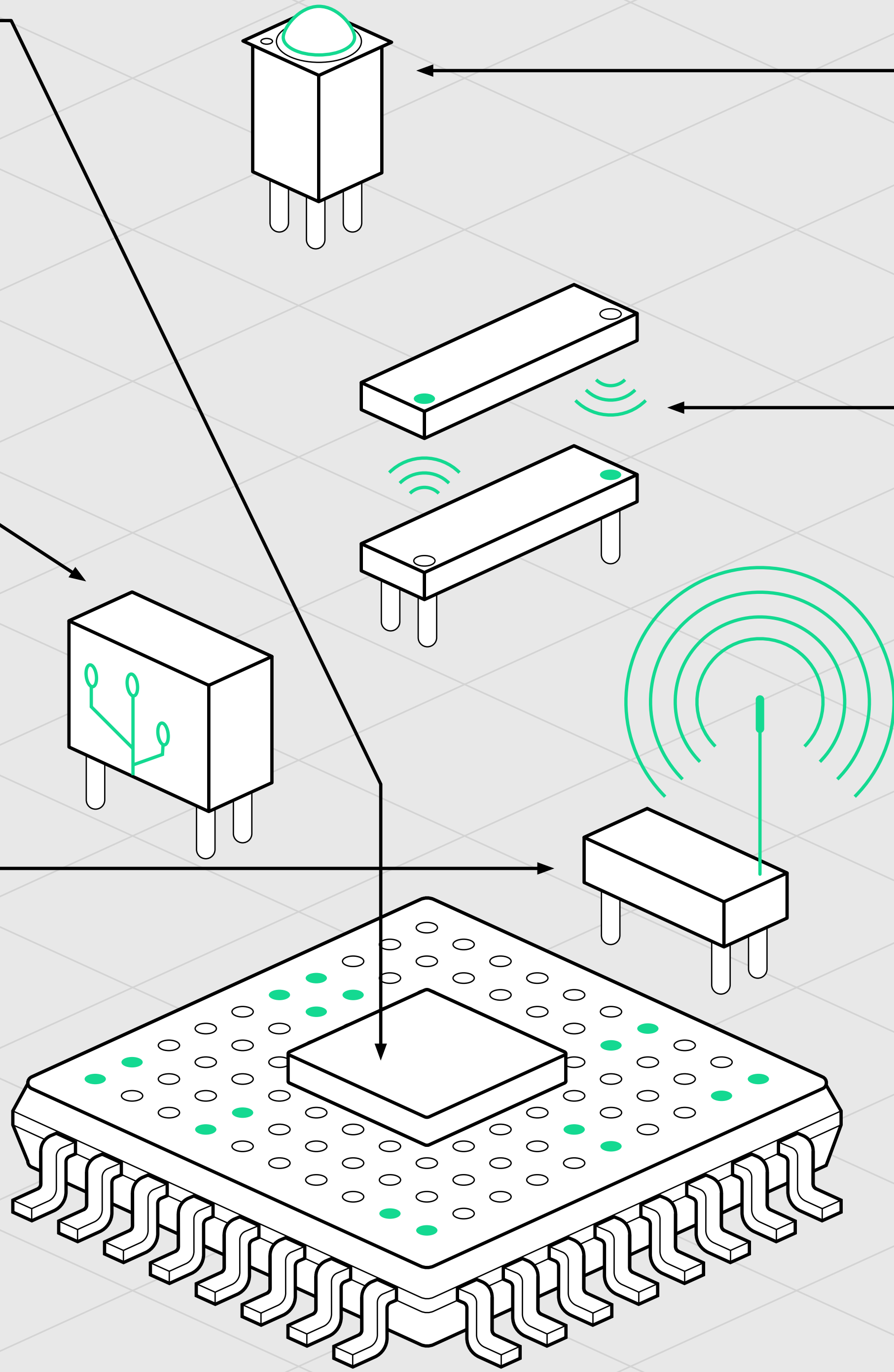
The backbone of every system-on-chip. This is where information processing takes place. The efficiency of this process and the amount of available computing resources are extremely important. This is because connected lighting systems require SoCs to process significantly more information than typical IoT applications. More recent system-on-chip solutions come with at least 32-bit processors, while enabling the usage of advanced programming techniques, such as thread isolation.

2 Memory block

Memory capabilities are equally important when it comes to the efficiency of data processing. For connected lighting applications, SoCs with at least 64kB RAM should be used to ensure sufficient efficiency and design flexibility. In addition to RAM memory, SoCs also include flash memory.

3 Radio module

A component enabling wireless connectivity. In hopes of reaching a broader set of market applications, silicon vendors often provide support for more than one radio protocols in their chips. This seems like a reasonable response to technological fragmentation in the wireless communication landscape, however, handling multiple protocol stacks requires more resources. And in the world of the IoT – and commercial smart lighting in particular – there is never enough processing power or memory.



4 Integrated peripherals and sensors

These can include a variety of solutions, including counter-timers, real-time timers or temperature sensors. Some of them are crucial for enabling specific smart lighting functionalities, such as precise scheduling or detailed operational feedback from connected luminaires

5 External interfaces

These are used for establishing connections with external devices and systems. Interfaces commonly found in today's SoC solutions include USB, FireWire or Ethernet, but also UART and PWM. Both are very relevant when it comes to development of products for connected lighting applications.

In traditional computing, all major components are separate chips interconnected via data magistrale. SoC components are, on the other hand, integrated into single chip and connected internally.

Issues to keep an eye on

Security

The problem with security in the Internet of Things is that efficient security mechanisms need to be run on small devices with very limited resources. At the same time, attempts to break these mechanisms can be carried out on powerful computing farms. And smart devices have to withstand such attempts. This is why SoCs need to support advanced encryption and authentication procedures. Strong processors found in the latest generations of chips are capable of handling modern and effective encryption algorithms without a significant trade off in overall SoC performance. To support encryption mechanisms even more efficiently, systems-on-chip often include dedicated hardware security modules.

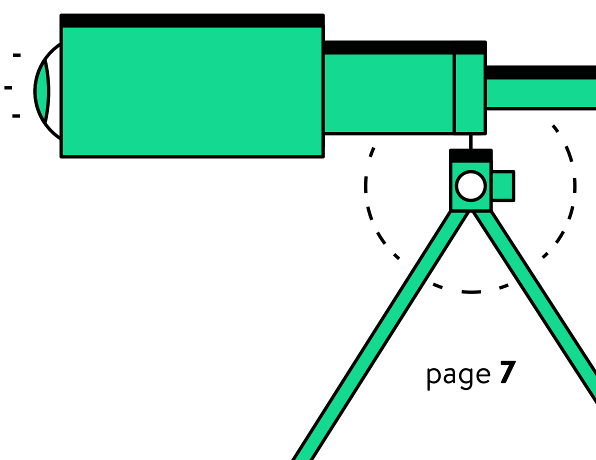
It should be remembered that the SoC's flash memory stores information that is shared by all of the nodes of a wireless mesh network. This includes sensitive data, such as security keys. **It is extremely important that SoCs used in professional lighting applications have built-in flash modules. This is what makes them resistant to the so-called "trash can attack"**, a serious security threat that is very often overlooked. If a discarded device has a system-on-chip with external flash memory, the memory unit can be unsoldered and security keys can be retrieved from it. This would allow unauthorized persons to eventually access the network.

Over-the-air update

Flash memory resources are crucial for flawless over-the-air firmware update capability. SoCs with sufficient flash can carry out this process without interrupting device operation because they can store two copies of firmware. A device can keep using the original copy while the update is being downloaded and implemented, and then switch to a new version once everything is ready. SoCs with smaller flash resources need to overwrite the original firmware during the over-the-air update process, which makes it impossible to perform such updates automatically over a mesh network (a device is no longer a network node as its firmware is being overwritten). Software stacks for connected lighting are particularly heavy so only few SoC solutions are capable of storing two copies of firmware at the same time.

Energy efficiency

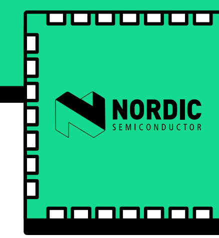
This is one of the fields where SoC technologies have made the biggest progress over recent years. For lighting and sensing applications, these advancements are particularly important. Nodes of a connected lighting network need to be able to process information on an ongoing basis even though some of them have no fixed power supply, which is why support for efficient battery operation is essential. In addition, even when LED fixtures are turned off, their radios are constantly turned on - so optimization in power consumption helps prevent excessive energy drainage. Over the years, silicon vendors have developed advanced power and resource management mechanisms to maximize energy efficiency and battery life. In the latest chips, peripherals have independent and automated clock and power management, so that they can be powered down when not required for task operation. This allows for keeping power consumption to a minimum.

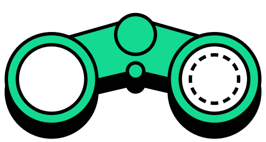


The state of SoC technology

System-on-chip technologies keep developing gradually but today we already have access to SoCs that can handle even the most demanding software stacks. **Silicon solutiions are already mature and advanced, providing us with all the tools we need to build wireless lighting control systems with wire-like reliability and multiple powerful features.** These tools are necessary, but we should not forget about the crucial role of software. It is the software that defines what these systems will do and how we'll be using them. At technology companies, individuals responsible for key decisions regarding SoC design implementation usually have much more to do with software than hardware. Because in the end, it is the silicon that adjusts to software needs, not the other way around.

There is a number of advanced SoCs available on the market, but for the **Silvair Firmware** we recommend our partners to use the **nRF52 series** chip from **Nordic Semiconductor.** We believe it is currently the best option – not only because of its powerful computing and memory resources, but also because it is perfectly suited for lighting applications and their specific requirements.





“ In the end, it is the silicon that adjusts to software needs, not the other way around ”

SILVAIR

Many ways to build a smart luminaire

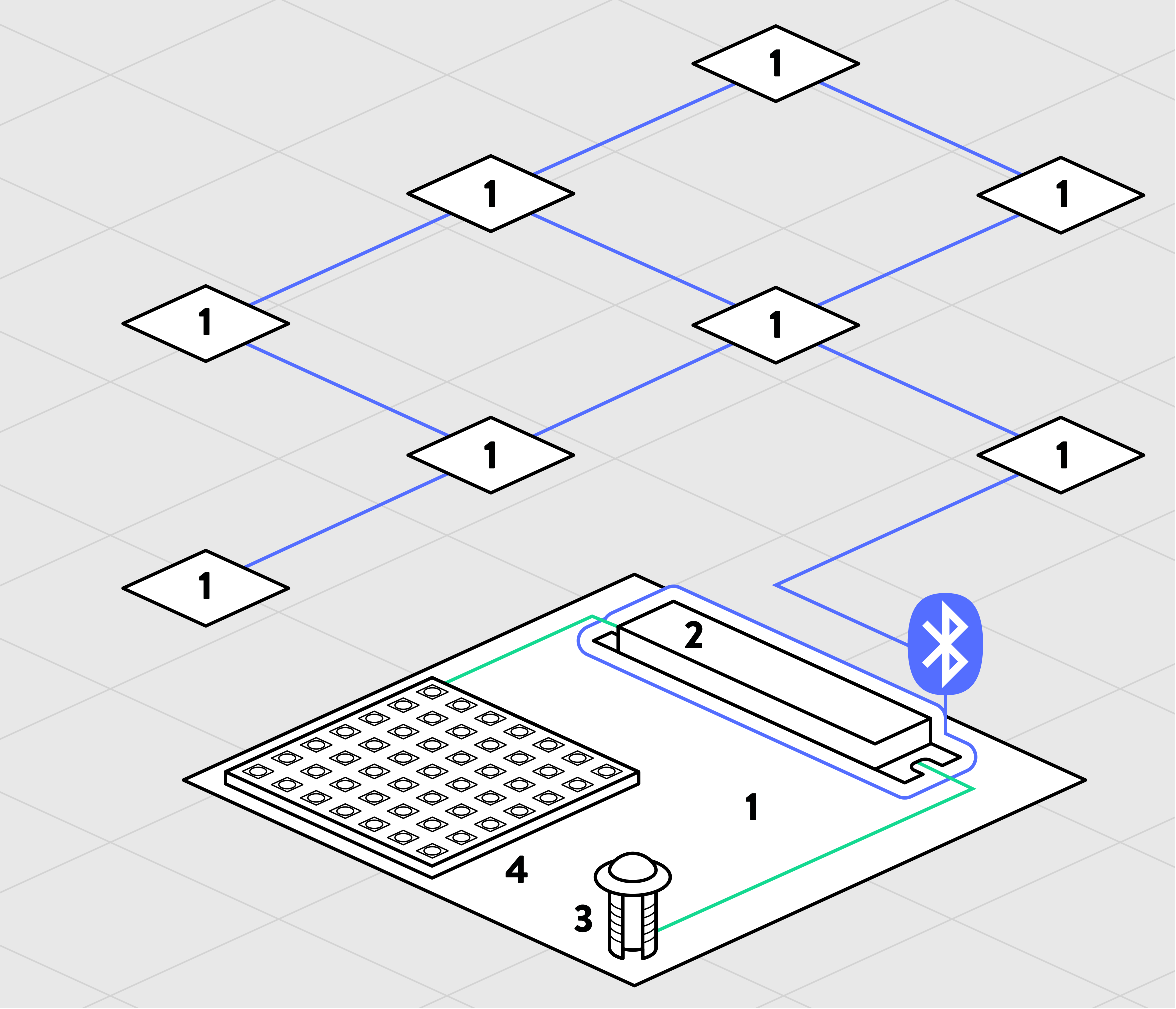
A system-on-chip with implemented lighting firmware needs to be integrated into each device that is supposed to be part of a Bluetooth mesh lighting network. In the case of simple one-component devices, such as wireless switches or standalone sensors, this is pretty straightforward. An SoC with appropriate firmware – including a relevant set of mesh models – has to be integrated with a device to enable contextual communication between that device and other nodes of a mesh lighting network. Things get a bit more complicated in the case of luminaires that consist of multiple different components. Since these components communicate with each other via local wired connections, there is no need to put an SoC into each of them. In fact, a Bluetooth mesh chipset needs to be integrated only into one of the components. **There is a number of different ways of implementing Bluetooth mesh connectivity into a luminaire.** These include the following arrangements:

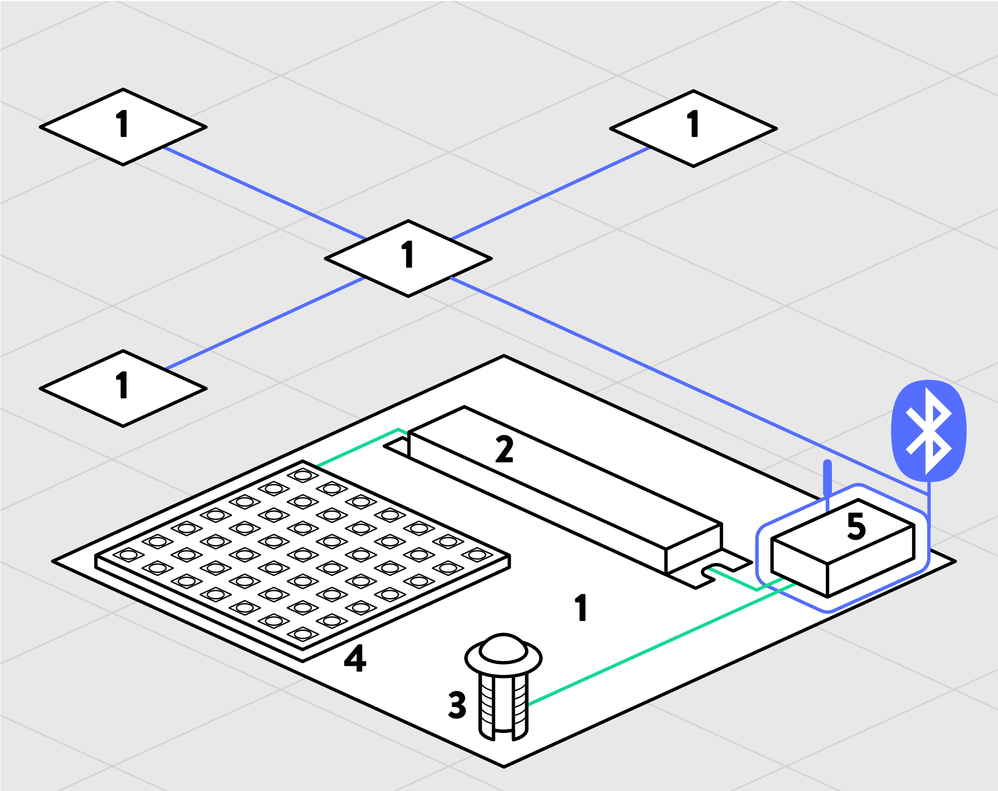
Legend:

1. Luminaire
2. LED driver
3. Sensor
4. Light source
— Wireless communication
— Wired connection
☐ Component with mesh firmware

Sample luminaire implementation #1

An SoC with mesh lighting firmware is installed inside an LED driver (marked in blue) with a sensor input port. The **green lines** represent local wired connections between luminaire components. The blue lines represent wireless communication within a Bluetooth mesh lighting network.



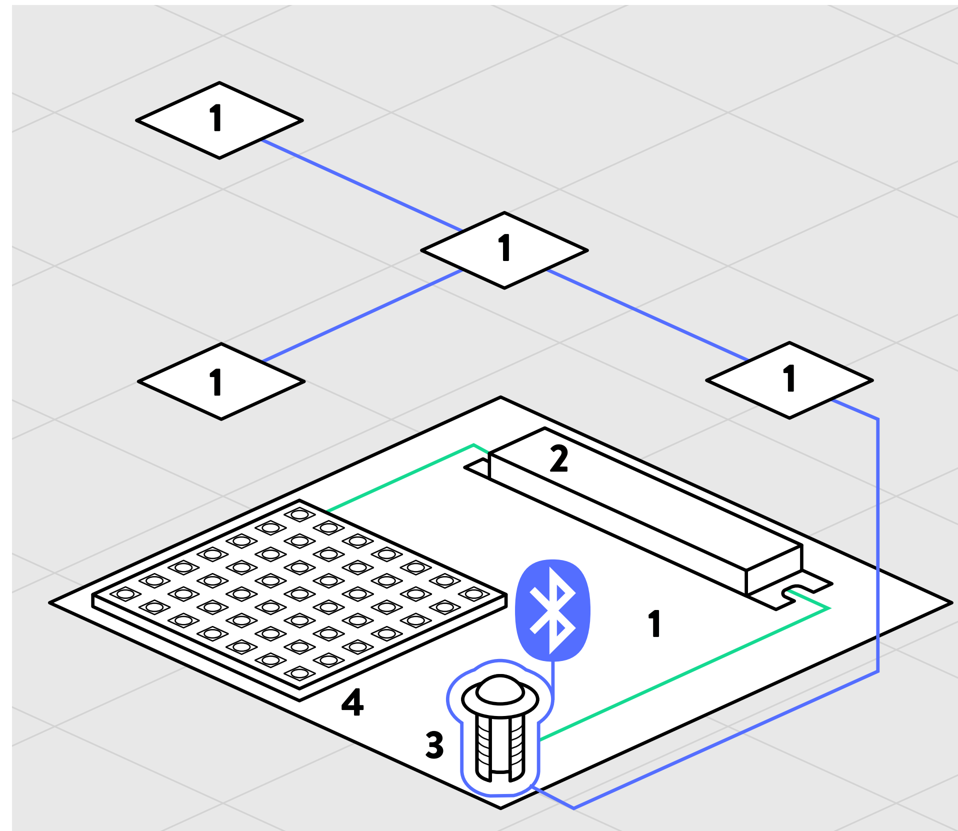


Sample luminaire implementation #2

An SoC with mesh lighting firmware is installed inside a fixture controller (marked in blue) with a sensor input port and a driver output port. **The fixture controller is a device that converts Bluetooth mesh commands to a 0-10V or DALI output**, enabling easy retrofit of existing LED systems into Bluetooth mesh lighting networks. The **green lines** represent local wired connections between luminaire components. The **blue lines** represent wireless communication within a Bluetooth mesh lighting network.

Sample luminaire implementation #3

An SoC with mesh lighting firmware is installed inside a sensor device (marked in blue). The **green lines** represent local wired connections between luminaire components. The blue lines represent wireless communication within a Bluetooth mesh lighting network.



Legend:

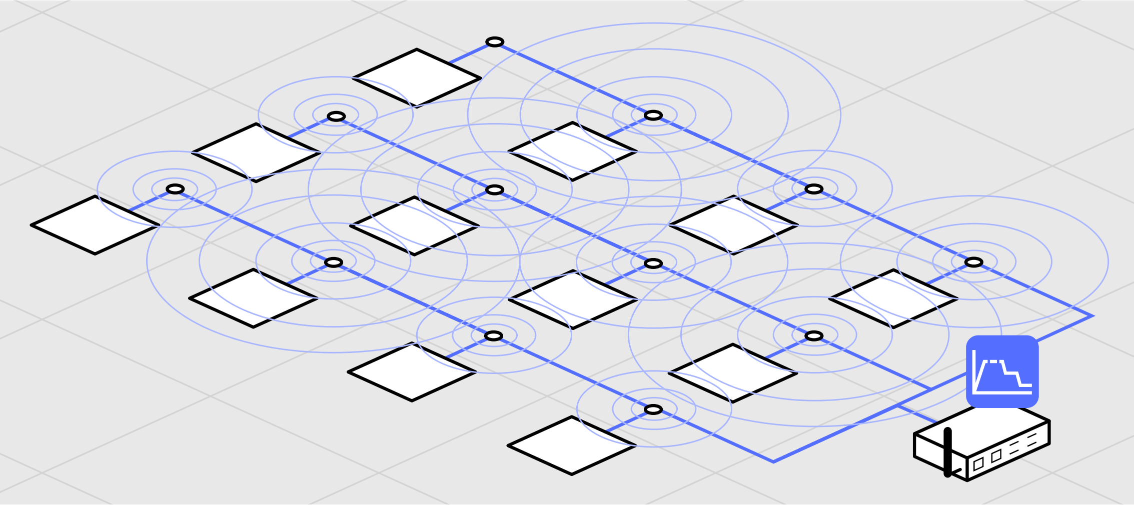
1. Luminaire	**5. Fixture controller**
2. LED driver	**— Wireless communication**
3. Sensor	**— Wired connection**
4. Light source	**☐ Component with mesh firmware**

SILVAIR

The challenges of wireless lighting control

Once we have Bluetooth mesh enabled luminaires and switches, we can start building a fully operational Bluetooth mesh lighting control system. **Since Bluetooth mesh is natively supported by all smartphones and laptops on the market today, such devices can be used to perform all necessary commissioning activities.** Of course, relevant software tools need to be installed onto them to handle these processes. These tools must allow for setting up the network, provisioning individual devices, grouping them in accordance with their arrangement across the space, and implementing the desired lighting control scenarios. But before we dive into control scenarios themselves, let's see how such a system works. There are a couple of things about Bluetooth mesh lighting networks that are different from anything the market had seen before.

Lighting control systems we've used so far - wired and wireless alike - had one thing in common. A central control unit that governs the entire installation. This kind of network architecture has very specific implications. The figure below presents communication patterns within a typical, centralized wireless control system:



We can see 12 lighting fixtures and 12 sensors, with communication happening through a gateway device that also serves as a controller. The controller stores the logic configuration for individual nodes, making sure that the desired scenarios are realized in accordance with the plan. **Such a centralized control paradigm is the approach that we've seen in all connected lighting solutions so far. It has evolved from classic IT networks where it has been used for ages. But the nature of mesh lighting networks is very different, and this legacy architecture proved not efficient enough.**

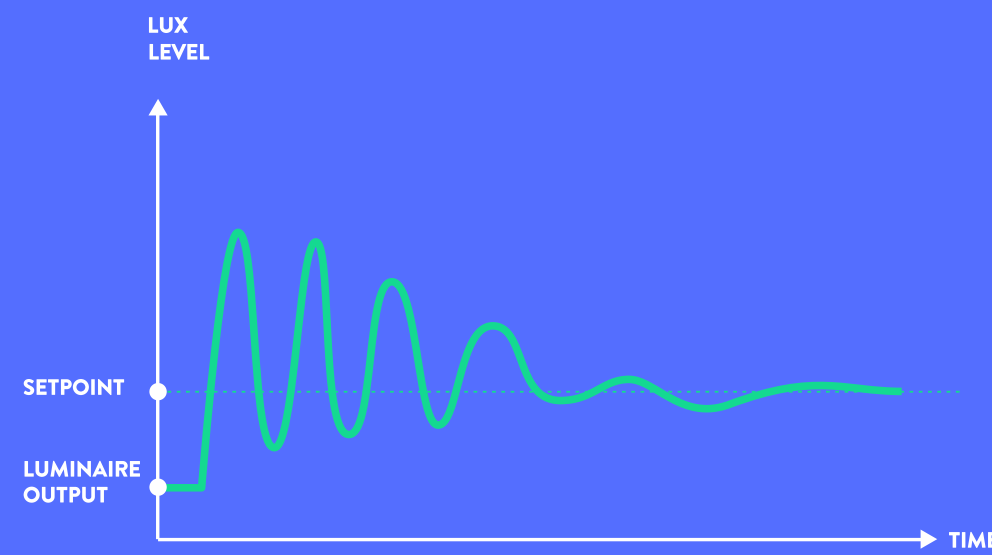
With bottlenecks typical for centralized control networks, wireless technologies with a centralized controller struggle to deliver performance that is expected in professional lighting applications. In the figure above, all data keeps travelling to the controller and back. The controller not only becomes a single point of failure, but also generates bottlenecks that affect the packet delivery rate and, ultimately, the throughput of the network. Sensors keep pushing information to the controller and the controller pushes commands back to each of the associated fixtures, multiplying traffic volume. In large-scale installations, this communication saturates the network. As a result, the efficiency of the entire system is very low, which is a serious issue for lighting applications where instant responsiveness is taken for granted. Such heavy traffic is not what low-power wireless technologies were designed to deal with, so it shouldn't be surprising that most of them fail to provide industry-grade reliability and scalability.

Oscillations in closed-loop daylight harvesting

Things get even more complicated when we get deeper into specific processes that take place within a sensor-driven lighting control system. **Closed-loop daylight harvesting is an example of a feature that is particularly challenging for wireless technologies. This is because of how the light level is stabilized in closed-loop installations**.

When ambient light level decreases significantly (a cloud shows up, a shutter is pulled down, etc.), the controller starts increasing a luminaire's output in response to sensor readings. However, it doesn't know what output should be reached - it just keeps pushing the light level up until the sensor says "it's enough!". In the meantime, sensor readings keep arriving. These sensor reports must be very frequent to ensure that the controller learns as quickly as possible about the fact that the desired set point has been reached. So we're talking about a constant stream of sensor data here.

But even with a high frequency of such readings, it is very unlikely that a reading precisely matching the desired set point will be delivered. A reading slightly exceeding that set point will probably arrive to the controller, urging it to stop increasing the luminaire output. In addition, by the time this message is delivered to the controller, the light level will have increased even further. So the next thing the controller needs to do is start decreasing the output, trying to precisely reach the predefined set point. This time, however, the output adjustment will be taking place at a lower pace (since there is much smaller difference between the current output and the output required to match the predefined set point). And again, the controller will learn a bit too late about the reached set point - so a reverse adjustment will once again has to be made. This cycle is repeated multiple times, as shown in the following figure:



Efficient, low-latency delivery of sensor readings is the key to enable smooth and eye-friendly daylight harvesting. If these readings are delayed at any point of the light level stabilization process, the luminaire's output increases/decreases too much versus the desired set point, resulting in strong oscillations that are visible to the human eye.

Ensuring low-latency packet delivery under these circumstances - when a constant stream of packets is being sent from multiple sensors - is a major challenge for low-power wireless technologies. **Bluetooth is designed to provide low-latency concurrent multicast, which allows daylight harvesting systems to perform as smoothly as in wired installations.** Due to their centralized architectures, other low-power technologies struggle with that. Low-latency message delivery is a specific requirement of the lighting industry, and only Bluetooth mesh (with its small data packets, high data rate, robust concurrent multicast and other unique properties) is currently able to satisfy it.

Decentralized control – a new approach to connected lighting

To address the challenges described above, the Bluetooth mesh specification provides exceptional support for lighting and sensing applications, delivering multiple innovative concepts that have not yet been seen in the low-power wireless environment. First off, it challenges the architecture of control systems. Until now, building automation was all about centralized controls: centralized lighting controller, centralized heating / ventilation controller, etc. This is the legacy of wired installations where centralized solutions work perfectly fine. But wireless technologies need a different approach to provide adequate performance, because centralized systems do not scale in the wireless world. Fortunately, Moore's Law today allows for moving the control function to the very edge - to the leaf nodes. This cuts half of the traffic instantly.

Bluetooth mesh networking introduces a fully decentralized architecture, removing bottlenecks and single points of failure. It doesn't require central controllers, since it puts a software controller into each luminaire. It doesn't require gateways because it's already in every personal computing device – so a smartphone can connect with each node directly when commissioning or configuration activities need to be performed. With such a dispersed logic configuration and multi-path packet delivery, mesh lighting networks based on Bluetooth mesh are very resilient.

Another crucial factor that increases the reliability and scalability of communication is the size of data packets. In Bluetooth mesh, they are very compact because the protocol is designed to transport large amounts of small packets as fast as possible. This compactness contributes to the spectral efficiency and throughput of wireless lighting control networks. **Radio is a shared medium and collisions are one of the major problems to address. The math is simple: a shorter packet means less collisions.** With both payload and overhead kept under very strict limits, Bluetooth's message occupies less than 400µs in the shared radio spectrum. This means that more than 2,500 such messages can theoretically be sent over 1 second.

When combined, all these features give Bluetooth mesh a very unique capability – concurrent multicast. Multicasting, the ability to distribute messages to multiple nodes in a single transmission, is an essential requirement in lighting applications. There is simply no connected lighting without multicast. In commercial settings, dozens or more network nodes must be able to respond to lighting control commands at the same time, with complete synchronization and minimal latency – because this is how lighting systems have always worked. There are low-power wireless technologies on the market that support multicast, but Bluetooth mesh is the only solution that can handle systems where multiple devices simultaneously send streams of multicast messages.

ICN – a new approach to wireless lighting control

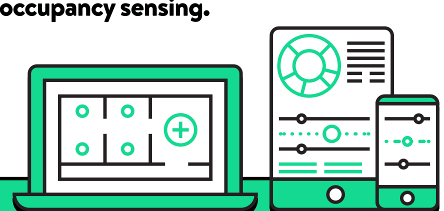
The communication paradigms introduced by Bluetooth mesh are in line with the Information Centric Networking (ICN) concept. This is a direction where the Internet is said to be heading in order to deal with constantly increasing traffic volume. It is an approach to evolve the Internet infrastructure by introducing uniquely named data as a core Internet principle. Moving away from a host-centric paradigm, the ICN model doesn't care about senders, recipients, addresses. Instead, named information is its focal point, making data independent from location, storage or application. In Bluetooth mesh lighting networks, this is realized through the publish-subscribe paradigm, one of the core concepts of the Bluetooth mesh specification. It resembles the way people communicate with each other on Twitter. An update is published and whoever has been subscribed - receives it. Addresses are assigned to information, not to specific devices. Luminaires subscribe to these addresses so that they can respond to relevant messages. Other devices (sensors, switches) publish to these addresses, distributing lighting control commands among all interested parties at the same time.

From manual operation to **full automation**

As already mentioned, Bluetooth mesh puts a software controller into each luminaire. This way, the luminaires themselves become intelligent and form a complete lighting control system without the need for a central control box. **The controller has been designed in such a way as to allow for plug-and-play implementation of advanced lighting control strategies, including occupancy sensing and daylight harvesting.** Of course, dedicated software tools are required to deploy such strategies - and it's the role of companies like Silvair to provide such tools to lighting manufacturers, facility managers or commissioning specialists.

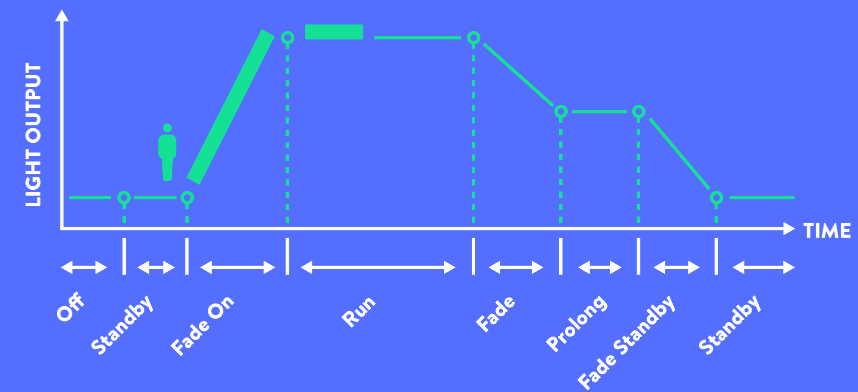
Lighting control scenarios enabled by the software controller can drive energy efficiencies in commercial spaces, while facilitating compliance with building energy codes. They can also be fine-tuned to address specific lighting needs. Once implemented, each scenario can be overridden at any time using e.g. a wireless wall switch.

The controller allows for implementing three different types of lighting control scenarios: **switch control, vacancy sensing and occupancy sensing.**



To streamline and accelerate the commissioning of commercial spaces, Silvair has developed a set of software tools for setting up and managing wireless lighting control networks based on Bluetooth mesh. Instead of using the one-by-one approach, **Silvair Commissioning** implements desired groups, lighting control scenarios and sensor associations all at once for multiple devices within a particular zone. Typical network formation processes are automatically performed in the background.

Forget the control box – software will take care of your luminaires

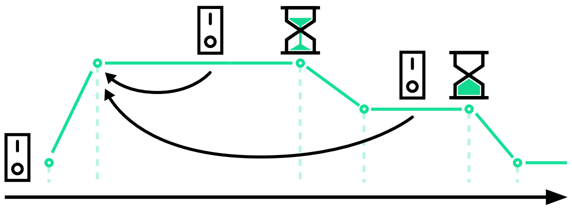


The figure above presents a typical operation of the software controller based on occupancy sensor readings. When turned on, the controller enters the Standby mode, in which it is ready to respond to events from sensors or switches. Once occupancy is detected, the controller enters the Fade On phase, increasing the luminaire's output to a predefined level. This is when the Run phase begins, triggering the Run timer. Its duration can also be flexibly adjusted to meet lighting requirements in a given space. Each occupancy detection resets the timer. On the chart above, the thick line represents the period during which occupancy is being detected. It is followed by a thin horizontal line which reflects the duration of the Run timer. Over this period, the luminaire's output remains unchanged. Once the Run timer has expired, the controller enters the Fade phase, during which the luminaire's output is decreased to a predefined level. Once this level is reached, the Prolong phase begins. This is a transitional phase that, again, lasts for a predefined period of time. If occupancy is detected during the Prolong phase, the controller returns to the Run phase, and the Run timer is triggered once again. However, if no occupancy is detected during the Prolong phase, then the controller starts decreasing the light level even further (the Fade Standby phase). Once the lights fade out, the controller enters the Standby mode.

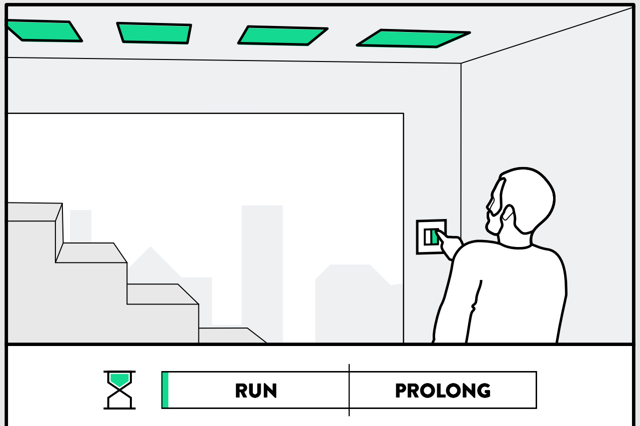
Switch control

This is the most straightforward scenario which features manual switch control and does not involve any sensors. It does use the controller, though, to introduce a certain degree of automation. Switches are used to trigger the controller and turn the luminaires on for a predefined period of time (the Run phase). Once the Run timer expires, the output of the luminaires is decreased as the controller enters the Prolong phase. Once the Prolong phase ends, the luminaires fade off. However, a switch can be used at any point to make the controller return to the beginning of the Run phase, extending the duration of the period over which the luminaires stay full-on. The switch control scenario is commonly implemented in stairwells, when no sensors are involved. **It can also support daylight harvesting to utilize natural daylight and reduce energy consumption even further.** Of course, ambient light sensors will have to be installed to feed the controller with precise information on natural daylight availability.

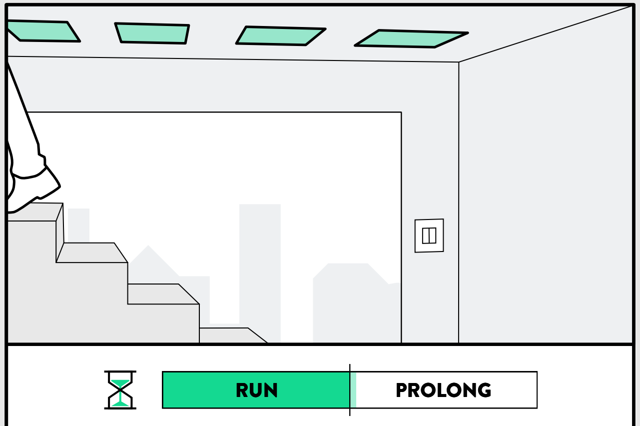
Referring to the controller's state diagram, this scenario can be presented as follows:



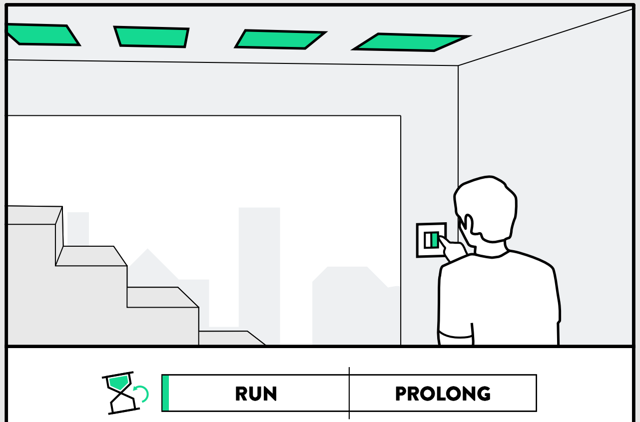
In practice, the switch control scenario could look like this:



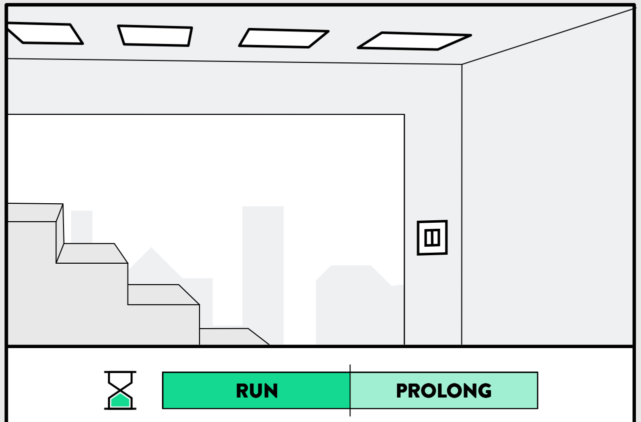
1 The lights are turned on manually with a momentary switch. This also triggers the Run timer.



2 Once the Run timer has expired, the light level gradually decreases to the Prolong phase (e.g. 20% of the full output).



3 If the switch is pressed again during the Prolong phase, the lights return to the full output and the Run timer is restarted.

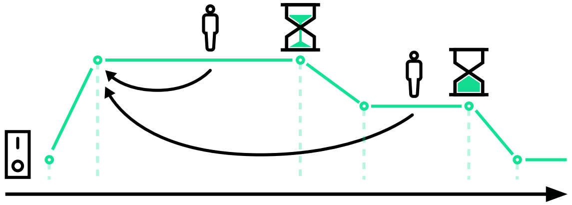


4 If the Run timer expires and the switch is not used again during the Prolong phase, the lights fade off.

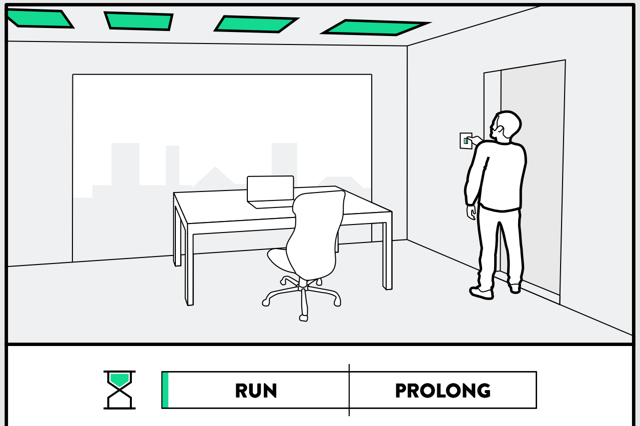
Vacancy sensing

In this scenario, switches are also used to trigger the controller. However, it introduces occupancy sensors to reduce energy consumption by turning the luminaires off when the space is vacant. Hitting a switch turns the luminaires on for a predefined period of time. Once the Run timer expires, the output of the luminaires is decreased as the controller enters the Prolong phase. Once the Prolong phase ends, the luminaires fade off. However, if occupancy is detected at any point during the Run phase or the Prolong phase, the controller returns to the beginning of the Run phase and the Run timer is restarted. This means that once the luminaires are turned on, their output will remain unchanged for as long as occupancy is detected in a given area. **The vacancy sensing scenario can also support daylight harvesting to utilize natural daylight and drive further energy efficiencies.**

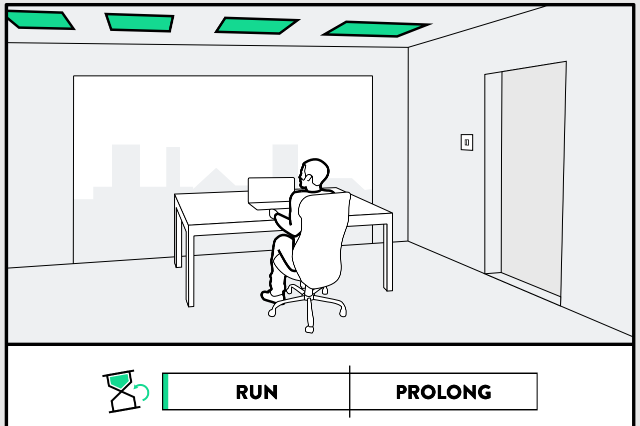
Referring to the controller's state diagram, this scenario can be presented as follows:



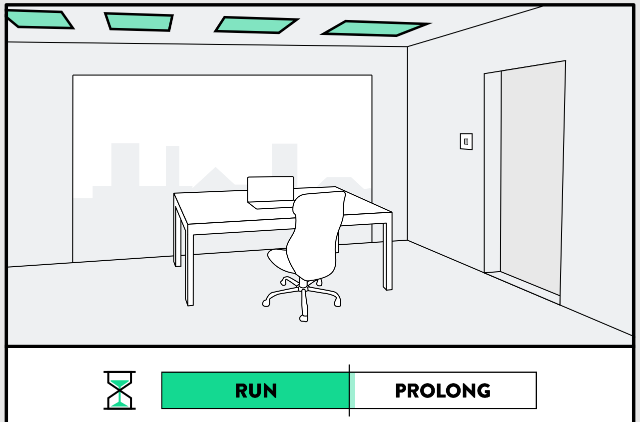
In practice, the vacancy sensing scenario could look like this:



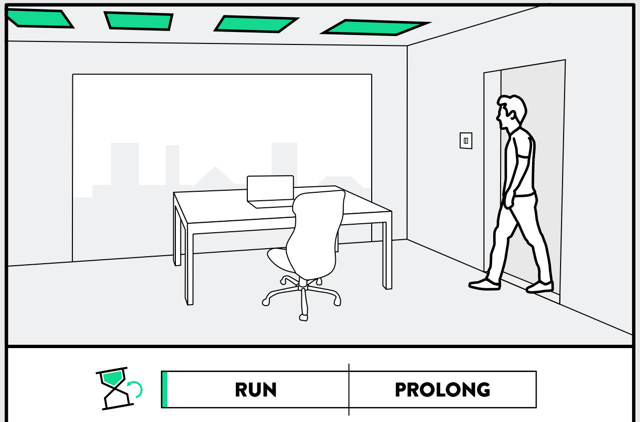
1 The lights are turned on manually with a momentary switch. This also triggers the Run timer.



2 Each occupancy detection restarts the timer, so the light stays at the configured level for as long as occupancy is detected in the room.



3 The lights automatically switch to the Prolong phase once the Run timer expires. This means that no occupancy has been detected for a predefined period of time (the duration of the Run timer).

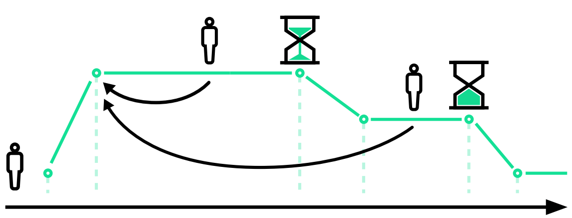


4 If occupancy is detected during the Prolong phase, the lights return to the full output and the Run timer is restarted.

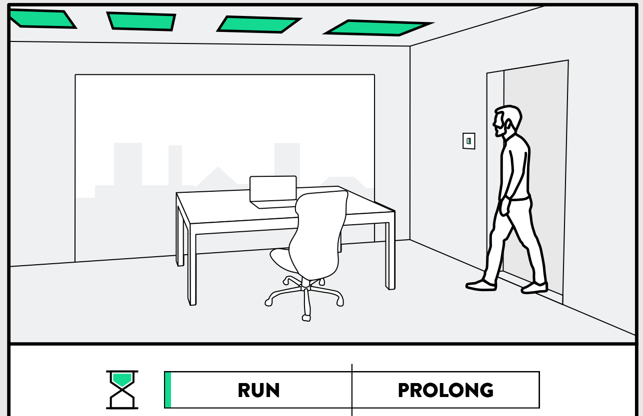
Occupancy sensing

Occupancy sensing is the most autonomous scenario as it uses occupancy sensors to both trigger the controller and keep it running. Switches are not needed, although they can be deployed if necessary. The luminaires turn on automatically whenever occupancy is detected in a given space. Once the Run timer expires, the output of the luminaires is decreased as the controller enters the Prolong phase. Once the Prolong phase ends, the luminaires fade off. However, if occupancy is detected at any point during the Run phase or the Prolong phase, the controller returns to the beginning of the Run phase and the Run timer is restarted. **The occupancy sensing scenario can also support daylight harvesting to utilize natural daylight and reduce energy consumption even further.**

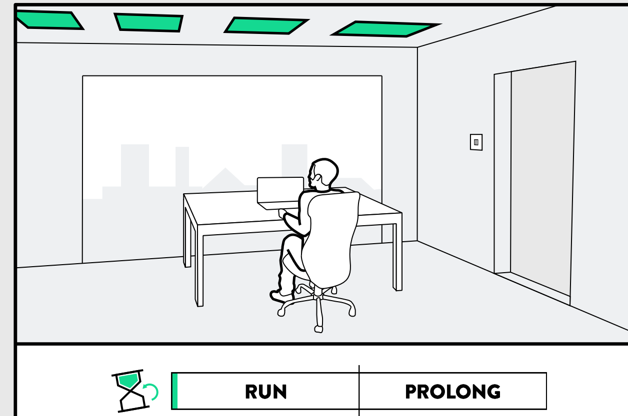
Referring to the controller's state diagram, this scenario can be presented as follows:



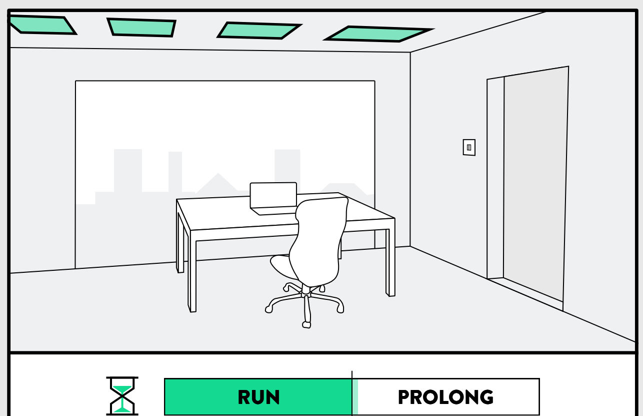
In practice, the occupancy sensing scenario could look like this:



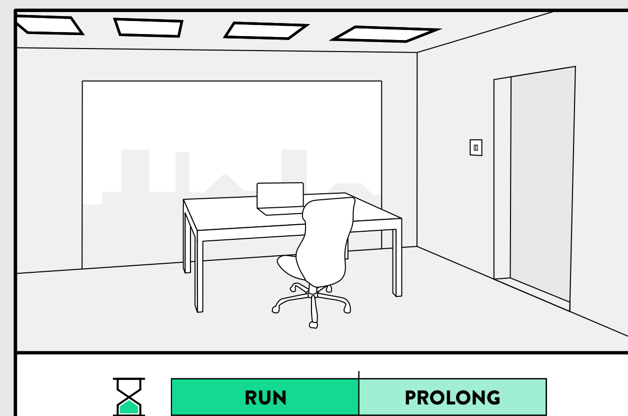
1 The lights turn on automatically whenever occupancy is detected. This also triggers the Run timer.



2 Each occupancy detection restarts the timer, so the light stays at the configured level for as long as occupancy is detected in the room.



3 The lights automatically switch to the Prolong phase once the Run timer expires. This means that no occupancy has been detected for a predefined period of time (the duration of the Run timer).



4 If no occupancy is detected during the Run phase and the following Prolong phase, the lights fade off.

Conclusion

On a final note, we encourage lighting professionals of all disciplines to explore the possibilities enabled by Bluetooth mesh. With a total of around 1,000 pages, the **Bluetooth mesh specifications** are not a casual evening read. But they dive very deep into the roots of the problems that wireless technologies are struggling with in lighting applications. Bluetooth mesh networking is a complex technology that requires a significant development effort from stack vendors, probably bigger than in the case of other low-power radio solutions. But to solve complex challenges, you need smart and powerful tools.

The good news is that lighting companies can go wireless without having to develop any software by themselves. Bluetooth mesh firmware solutions - including the ones dedicated for lighting applications – are already available on the market. **Here** you can find a list of Bluetooth mesh products that have been qualified by the Bluetooth SIG. It's updated regularly and it's growing fast, reflecting the rapidly expanding ecosystem of globally interoperable lighting and sensing devices that use Bluetooth mesh as their wireless engine. If you decide to use Bluetooth, make sure that the solutions provided by your firmware vendor can be found on that list. Also, make sure that you are provided with a complete set of software tools for commissioning, monitoring, or whatever you and your customers will need. This way, you won't have to bear the costs of developing adequate sotware solutions on your own.

If you have any questions, visit **www.silvair.com/contact**
We'll be glad to help!

If you would like to learn more about Bluetooth mesh and wireless communication in lighting applications, we recommend the following white papers and e-books published by Silvair:

A tale of five protocols
will guide you through the diverse wireless connectivity landscape, outlining the major strengths and weaknesses of the five leading technologies: Wi-Fi, Z-Wave, ZigBee, Thread and Bluetooth. The e-book describes the differences between them, explaining how they tackle the challenges of today's IoT, with a particular emphasis on the building automation segment.

How to build a wireless sensor-driven lighting control system based on Bluetooth mesh
is a deep dive into the most important lighting and sensing models included in the Bluetooth mesh specification. The white paper explains which models should be used in different types of devices, and how these models interact with each other to fuel a robust and autonomous lighting control system.

Information-centric networking: a revolutionary approach to wireless lighting control
explains how information-centric networking (ICN), a revolutionary concept developed to address the challenges relating to the exponential growth of data traffic across the Internet, can be used to address the main pain points of today's wireless lighting control systems. The white paper also describes how Bluetooth mesh embraces the ICN architecture to enable wire-like reliability in professional lighting applications.

Inside a Bluetooth mesh lighting network: a comprehensive guide for lighting manufacturers

Rev. 1, January 2019

Text / content: **Szymon Rzadkosz** (Silvair)
Illustration / editorial design: **Jerzy Nosek & Milena Rózga** (Silvair)
Concept / compliance: **Katarzyna Meier** (Silvair)

SILVAIR

silvair.com