

Bridging light applications to the IP domain

Citation for published version (APA):

Bui, T. V., Lukkien, J. J., Frimout, E., & Broeksteeg, G. (2011). Bridging light applications to the IP domain. In *29th International Conference on Consumer Electronics (ICCE 2011, Las Vegas NV, USA, January 9-12, 2011)* (pp. 235-236). Institute of Electrical and Electronics Engineers. <https://doi.org/10.1109/ICCE.2011.5722558>

DOI:

[10.1109/ICCE.2011.5722558](https://doi.org/10.1109/ICCE.2011.5722558)

Document status and date:

Published: 01/01/2011

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Bridging Light Applications to the IP Domain

The Vinh Bui*, Johan J. Lukkien*, Emmanuel Frimout†, and Gerard Broeksteeg†

*Department of Mathematics and Computer Science, Eindhoven University of Technology, The Netherlands

†Philips Lighting, Eindhoven, The Netherlands

Abstract—With the advent of LED lights as well as lighting systems that have communication capabilities, new lighting setups and scenarios have become possible. New standards of the ZigBee Cluster Library (ZCL) in lighting systems describe the protocols for controlling and monitoring devices in such systems. In this paper we examine the problem of bridging this ZigBee domain to the IP domain such as allowing standardized access and easy integration of advanced lighting scenarios. We define two access levels for the bridging: a message-based API, which translates the protocol and which allows integration in arbitrary networked environments and an HTTP interface that gives access to an embedded web application for direct control purposes. We show a realization of this design on a prototype board of ZigBee and IP modules along with measurements on response times.

I. INTRODUCTION

Philips Lighting develops many radio frequency controlled lighting products, for the office and home domains. Interoperability among these devices is achieved through specific use of the ZigBee Pro standard. For example, a SmartLink system consists of lights and remote controllers, where remote controllers are used to change the appearance of lights. Besides the integrated control use cases, a large diversity of use cases that have been identified require opening up the lighting control system to devices outside the ZigBee domain such as PDAs in the IP domain.

Related work to connect ZigBee networks with the TCP/IP network can be categorized into a gateway based approach [1][2] and an overlay based approach [3][4]. In this paper we describe a gateway approach to bridge light applications to the IP domain. In order to prove that the approach can successfully support our system, we design and implement a prototype called *SmartBridge*.

II. MAIN REQUIREMENTS AND DESIGN CHALLENGES

The following main requirements and design challenges are considered for our approach, final design, and implementation.

Compatibility: The SmartBridge works and communicates with SmartLink devices based on ZCL messages.

API and Web application: An API of the SmartBridge enables IP applications to interact with SmartLink applications. The API must be extensible to support complex control use cases such as schedules and dynamic effects. An embedded web application supports a set of standardized controls.

Plug and Play: It must be easy for users to set up the SmartBridge. Besides, a mechanism to discover its services such as the embedded web application is needed.

Performance: The latency, i.e. the time elapsing from the moment that a user issues a control command until a light executes that command, should be below 200ms in a local network.

Reasonable cost: The cost of the bridge device must be low enough for mass deployment in consumer homes.

III. SMARTBRIDGE AND SYSTEM DESIGN

Taking related work and the requirements into account, we propose that our SmartBridge uses the gateway approach. The main role of this gateway is to translate the protocol of different applications between the networks. The SmartBridge hardware consists of two main modules: the IP module to interface with the TCP/IP network and the ZigBee module to interface with ZigBee networks. These modules communicate via a serial interface. In order to support user's direct control commands, we implement an embedded web application for requests in the local network. We also design and implement an experimental web portal, which supports services such as controlling lights from the Internet and finding the SmartBridge's local IP address. Figure 1 shows a lighting control system, which has a SmartBridge connected to a ZigBee network at home and to the web portal through the Internet.

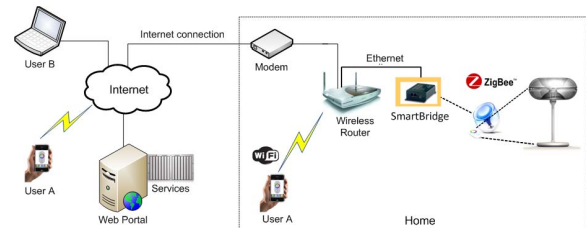


Fig. 1. Lighting control system uses the SmartBridge and a web portal

Software on the SmartBridge supports two access interfaces. First, the HTTP interface gives access to an embedded web application. The web application supports configuration and control functions, much alike the regular remote controller. Second, a message-based interface, to which PDA applications and the web portal services can communicate. The latter interface is necessary to integrate lighting into other applications. For example, a PC application could use the light as an output medium to warn the user about an upcoming appointment.

A layered architecture showing two hardware modules is presented in Figure 2. Commands received from the mentioned interfaces are processed in a service layer into corresponding ZCL messages and sent to the ZigBee module via the serial interface. The SmartLink application on the ZigBee module has responsibilities of processing these messages and sending them to lights. The service discovery protocol is realized through the UPnP. The module *SW Updater* is implemented for the purpose of remote software update. There are two protocols in our system:

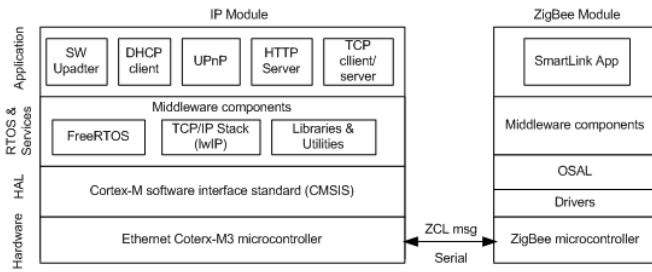


Fig. 2. Layered architecture overview of the SmartBridge prototype

SmartBridge-IP protocol: The protocol is based on IP text messages, and it supports a number of commands handled 'first come first serve'. Messages that are processed later are buffered. When the IP interface receives a command, it forwards the command to a command processor service. This service parses the command and calls a related function to process it, i.e., translating it into a ZCL message. Our protocol is stateless to make connection management simple. Examples of command sequences are shown in Figure 3. In both (a) and (b) a lighting application sends a command to control or to monitor lights, then an acknowledgment is returned from the SmartBridge. In (b), an attribute value of light 2 then is returned to the user's application.

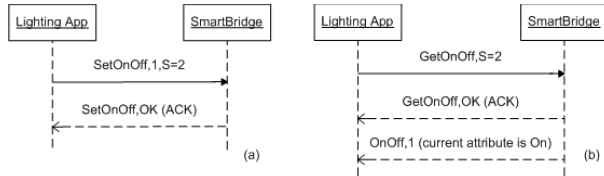


Fig. 3. Examples of command sequences of the SmartBridge-IP protocol: (a) Set light 2 on; (b) Get a current on/off attribute of light 2

SmartBridge-ZCL protocol: The protocol is based on ZCL messages and is used to communicate between IP and ZigBee modules via the serial interface. The messages come from the IP module or from the ZigBee module. The control commands received at the IP module are translated into ZCL messages and sent to the ZigBee module. The ZCL messages then are processed into ZigBee's packets and sent to lights. The message format is: $[Zcl, \langle address \rangle, \langle command, sub-command, parameters \rangle]$, where addresses and commands are defined in the ZCL for the lighting system.

To support future commands, the command field could contain a binary ZCL command that can be sent directly to lights without a translation on the ZigBee module. The messages sent from the ZigBee module can be response messages containing light's attributes or the commissioning information. In order to account for the speed between Ethernet and UART, we implement message buffers between the application and the UART controller and a software flow control mechanism on the serial interface.

There are many scenarios in which users want to control lights through the Internet while they are away from home. We propose an approach of using a web portal as a central point to communicate between SmartBridges and IP applications. A TCP client of the SmartBridge establishes a connection to the web portal, such avoiding problems with a firewall or

NAT. Through this connection, the web portal can forward users' requests to the SmartBridge. The web portal is also necessary to support services such as updating software and finding the SmartBridge's local IP address. Figure 4 shows main components of the web portal.

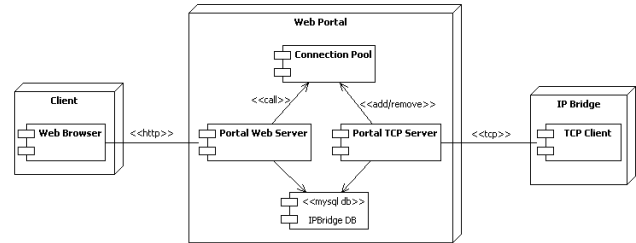


Fig. 4. Component based overview of the web portal

IV. IMPLEMENTATION

A hardware platform of the SmartBridge is shown in Figure 5. We use FreeRTOS as a multi-threading operating system and lwIP TCP/IP stack in the IP module. The SmartLink software platform is employed in the ZigBee module.

	IP Module	ZigBee Module
MCU	TI LM3S6965	TI CC2530
SRAM / Flash	64KB / 256KB	8KB / 256KB
Network	10/100 Ethernet	2.4GHz ZigBee
Serial interface	UART	UART

Fig. 5. Hardware platform of the SmartBridge

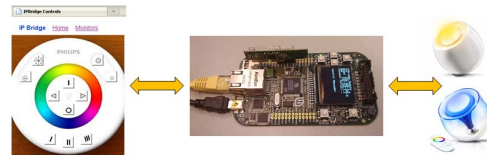


Fig. 6. Web application implemented on the SmartBridge to control lights.

Using a multi-threading RTOS is easier in term of design and implementation for future extensions of the interfaces and of applications. Figure 6 shows an implementation of the SmartBridge prototype and a web interface to control lights. The first evaluation shows that the average command latency is less than 200ms in a local network and is less than one second through the web portal.

V. CONCLUSIONS

In this paper we present the design and implementation of the SmartBridge prototype to bridge the light applications to the IP domain. The first testing and evaluation show that the prototype works and meets the main requirements. Future work includes security which we did not consider until now.

REFERENCES

- [1] P. Kinnery, "Gateways: Beyond the sensor network," ZigBee Alliance.
- [2] M. R. Trchalk, D. D. Programme, I. P. Oc(enek, D. D. Programme, S. Prof, and M. Lvda, "Abstract zigbee gateways."
- [3] A. Dunkels, J. Alonso, T. Voigt, H. Ritter, and J. Schiller, "Connecting wireless sensor networks with tcp/ip networks," in *Proceedings of the Second International Conference on WWIC*, 2004, pp. 143–152.
- [4] H. Dai and R. Han, "Unifying micro sensor networks with the internet via overlay networking," in *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks*, 2004, pp. 571–572.