

Instituto Tecnológico y de Estudios Superiores de Occidente

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018,
publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Departamento de Electrónica, Sistemas e Informática
Especialidad en Sistemas Embebidos



Comparison between Silicon Labs and NXP development platforms for Matter

TRABAJO RECEPCIONAL que para obtener el **DIPLOMA** de
ESPECIALISTA EN SISTEMAS EMBEBIDOS

Presenta: **MIGUEL ANGEL CARDENAS ORTEGA**

Asesor **SERGIO NICOLÁS SANTANA SÁNCHEZ**

Tlaquepaque, Jalisco. 8 de agosto de 2024.

Comparison between Silicon Labs and NXP development platforms for Matter

Cardenas Ortega, Miguel Angel
Departamento de Electrónica, Sistemas e Informática
Instituto Tecnológico y de Estudios Superiores de Occidente
Guadalajara, México
miguel.cardenaso@iteso.mx

Santana Sánchez, Sergio Nicolás
Departamento de Electrónica, Sistemas e Informática
Instituto Tecnológico y de Estudios Superiores de Occidente
Guadalajara, México
sergioni@iteso.mx

Abstract— **Matter is a new application and transport protocol that can be used over IP networks to solve the interoperability issues of the home automation industry. The objective of this research is to evaluate the development complexity of Matter for embedded systems developers. A market survey of the available semiconductors that support Matter was performed to select two Matter chips from Silicon Labs and NXP for a proof-of-concept test. The test simulated the development of two basic smart home products, a smart light bulb and a smart light switch, to compare the development platforms and processes of each chip. Findings revealed that Silicon Labs offers a more robust and complete development platform for Matter than NXP. In conclusion, the current state of Matter development platforms and tools offers enough support for developers to integrate Matter into an embedded device.**

Keywords— *Matter, Home Automation, Thread, Border Router, Mesh Networks.*

I. INTRODUCTION

In the early days of the home automation industry, smart home technology companies started developing their own ecosystems independently and with no regard for the interoperability between products from different vendors. Furthermore, manufacturers implemented their own proprietary solutions for wireless communication between devices. This created a fragmented market that made it difficult for consumers to know which products would be compatible with each other, while also severely limiting the consumer's options to a single ecosystem, brand, or manufacturer [1].

Finally in 2019, the interoperability concerns in the home automation industry were addressed by a group of the largest technology companies involved in the market, including Apple, Google, Amazon and Samsung. These companies came together to create the Connectivity Standards Alliance, or CSA, whose main objective was to create, develop, and promote the use of a new open communication protocol specifically designed to improve the lack of compatibility between smart home products and to integrate more robust security features [2]. CSA's efforts finalized with the creation of the Matter protocol, which was released in late 2022.

The design of the Matter standard is based on the following four principles: 1) *Interoperability*: to ensure proper communication among devices from different manufacturers; 2) *Security*: to integrate robust security mechanisms aimed at protecting user's data and privacy; 3) *Simplicity*: to facilitate the

handling of smart home products and the user experience; and lastly, 4) *Reliability*: to enable a consistent and responsive local connectivity in any home network [3].

Matter is an open-source, royalty free, communication protocol focused on home automation applications. This protocol is IP-based since it relies on other IP networking protocols for the routing and transmission of data such as Wi-Fi, Thread or Ethernet [3]. By making Matter compatible with IP protocols, the CSA provided manufacturers with an already proven and standard technology, facilitating wireless communication among their products. Moreover, Matter has become a key technological breakthrough in the home automation industry with a rapid adoption rate facilitated by the support of the largest technological companies in the market. According to ABI research more than 5.5 billion smart home devices with Matter support will be sold by 2030 [4].

Due to the novelty of Matter, there is a lack of documentation to support its development process in embedded systems. The available documentation on Matter is provided by the manufacturers of the few semiconductor chips that support this protocol. Hence, it is important to expand on the Matter literature specifically from the perspective of the development process for a new embedded device.

The main objective of this research is to evaluate the feasibility and complexity of integrating Matter in an embedded system and to extend the knowledge on the topic to assist future developers in making informed decisions when designing smart home products. To achieve this objective a market survey and proof-of-concept test were performed to compare the development process of two different chips and their development platforms. This work focuses on platforms using Thread and BLE protocols. Matter can also use Wi-Fi or Ethernet; however, it will not be discussed in the content of this document.

The organization of this paper is as follows: Section II and III describe the hardware and software characteristics of the Matter protocol. Section IV describes the Matter semiconductor market survey. Section V addresses the proof-of-concept tests and the Matter development evaluation. Lastly, Section VI concludes the paper and describes how to expand the research in the future.

II. MATTER HARDWARE CHARACTERISTICS

Matter is an open-source communication standard that resides entirely on the application layer of the standard OSI model. By being an application-level protocol, it relies on the IP protocol for the networking layer and from other protocols for the physical and data link layers. Matter currently supports Ethernet, Wi-Fi and Thread as its physical layer protocols [3]. The representation of Matter using the OSI model is shown in “Fig. 1”.

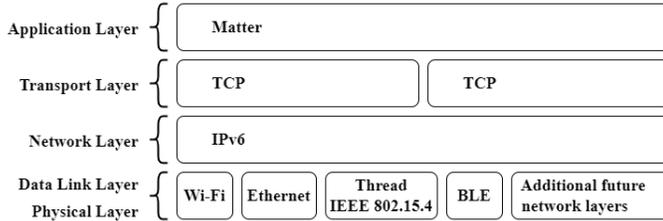


Figure 1. Representation of the Matter protocol using the OSI model.

A. Thread

Thread is a low-power, wireless mesh networking protocol used by Matter. Thread is based on the IEEE 802.15.4 standard, which defines the physical and data link layers for wireless personal area networks, or WPANs. It operates in the 2.4 GHz frequency band, and offers support for mesh networks, as opposed to Wi-Fi or Bluetooth which also operate in the same frequency. Thread is IP based and uses a version of the IPv6 protocol for low powered and low memory devices named 6LOWPAN, which makes Thread compatible with other IP based networks such as the internet [5].

B. Bluetooth Low Energy

Matter also utilizes Bluetooth Low Energy, or BLE, for the commissioning of new devices. Commissioning in this context refers to the process of assigning the necessary security credentials to a new device to access a secured network. Bluetooth is not used by Matter to transmit sensitive data to or from devices; it is only used for the commissioning process [6].

C. Network Topology

Matter supports the use of mesh networks only when using Thread. Mesh networks enable communications between all the nodes within a network and allow the creation of multiple efficient paths to route data packets across the network. Moreover, mesh networks have multiple benefits over the *star* type networks used by Wi-Fi; their self-healing characteristics make them more reliable, and if configured correctly, they can have a greater range by adding more nodes [5]. This is especially important in a home automated environment where sometimes the wireless range of a Wi-Fi router is not enough to cover the entire household.

Matter also supports the *star* topology for multiple networks. This is required to communicate Matter devices that use Wi-Fi or Ethernet with those that use Thread. The *star* topology network consists of multiple peripheral networks joined together by a central network. The central network will typically be the user’s home network using Wi-Fi or Ethernet. The peripheral networks can be any of the supported types by the Matter specification, but it will usually be a Thread network. To route

packets between the central and peripheral networks, a Border Router must be used [6]. “Fig. 2” shows the star topology of a Matter network.

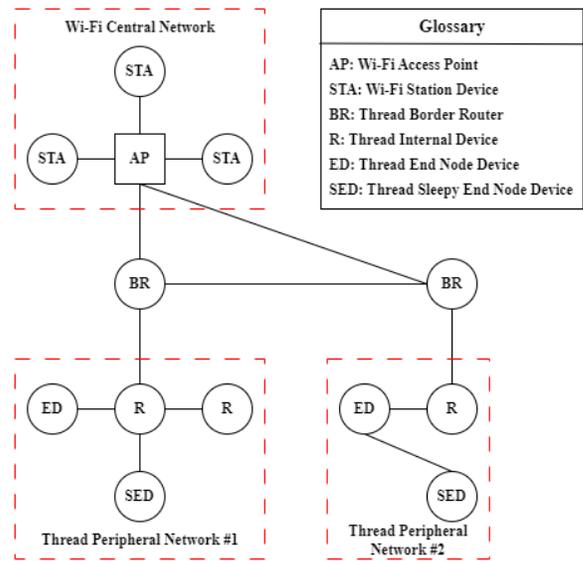


Figure 2. The star topology of a Matter network.

D. Border Router

Border Routers enable communications between two Matter subnetworks. These devices provide the routing of packets between a Wi-Fi subnetwork and a Thread subnetwork and vice versa. Border routers essentially translate among Matter devices that use Wi-Fi or Ethernet and those that use Thread. Basically, all the information that crosses between any two subnetworks will flow through a border router device [6].

III. MATTER SOFTWARE CHARACTERISTICS

A. Device Data Model

The Matter 1.3 core specification defines a particular device data model that implements a hierarchal description for the features of a device [7]. The device data model is shown in “Fig. 3”.

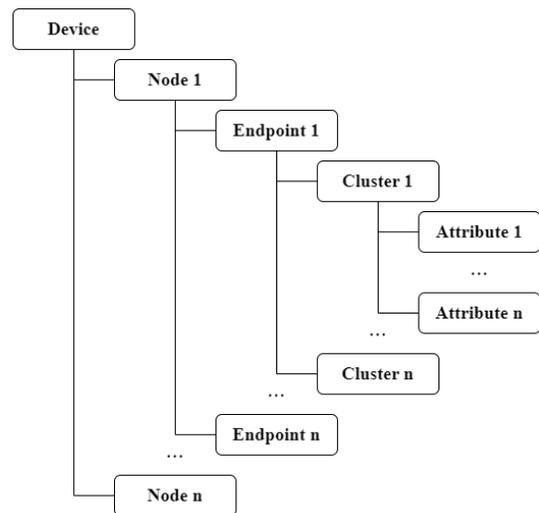


Figure 3. Matter’s device data model.

1) *Device*

A device is the highest level on the hierarchy of Matter's data model and it is composed of one or more nodes. A device entails all the available features of a single product [7].

2) *Node*

A node is what a user can perceive as a functionally distinct and whole unit, encapsulating all features. A node has a unique identifier and can be addressed by other nodes on the network, either from the same device or from external devices. This means that all communications inside a Matter network starts and terminates at a node [7]. For example, in a smart light switch plate that contains two switches, the whole product, meaning the whole switch plate, would be the device, and each single switch would work as a separate node.

3) *Endpoint*

Endpoints encapsulate a feature or functionality set of a product. Nodes are made of a collection of endpoints [7]. Referring to the switch plate analogy mentioned previously, if each switch on the plate contains a motion sensor, the switch functionality would be one endpoint, while the motion detection functionality would work as another endpoint.

4) *Cluster*

Clusters group a single feature or functionality of a device. Endpoints are made of a collection of clusters [7]. For instance, the smart switch could use the on/off cluster to control a light or the level control cluster to control a dimmable light.

5) *Attribute*

This is the last level of the hierarchal data model. Attributes are the characteristics and states exposed by a cluster. Attributes can be of different data types such as uint8, strings or arrays [7]. For example, the lighting cluster of a smart light will store the light's state attribute as on or off.

IV. SEMICONDUCTOR MARKET SURVEY FOR MATTER

A survey of the available semiconductors on the market that support the Matter Protocol was conducted. Only chips or semiconductors that met the following criteria were considered in the survey:

- 1) The chip is already in production. Semiconductors in development were not considered.
- 2) The chip has support for wireless communication using the IEEE 802.15.4 standard.
- 3) The chip has support for wireless communication using Bluetooth Low Energy or BLE

- 4) The chip supports the Thread protocol stack.

- 5) The chip supports the Matter protocol stack.

The survey included the following manufacturers: Microchip, Nordic Semiconductors, NXP, Silicon Labs, and Texas Instruments. Moreover, the survey was conducted with the available website documentation from each manufacturer.

Survey results found that the industry offers two solution types to integrate Matter into an embedded system. The first solution is called standalone; this is a single chip solution capable of storing and running both Matter and Thread stacks, as well as the user application. The second is a two-chip solution where there is a host processor and a Radio Co-Processor, or RCP. The main or host processor stores and runs both the Matter stack and the user application, while the RCP handles the Thread stack and has the required radiofrequency hardware needed for wireless communication. There are inherent benefits and drawbacks for each type of solution. Standalone solutions are usually cheaper and easier to implement because they require less hardware components. As a disadvantage, standalone solutions can struggle with memory space because everything must be stored in a single chip. However, if supported, this disadvantage could be alleviated by using an external Flash chip. On the other hand, the two-chip solution offers more flexibility, since the main processor does not require specialized radio hardware. This allows the developer to choose from a wider range of microcontrollers. The survey shows the hardware differences between standalone and RCP chips.

Table I compares the characteristics of the available Matter semiconductors in the market. Microchip was the only manufacturer that lacked a Matter chip in production.

The survey shows that all the chips use the popular Cortex-M core from ARM. Further findings demonstrate that the standalone chips offer more memory space in both Flash and RAM compared to the RCP chips, since standalone solutions must store more data than only the Thread stack. Also, the current consumption of Matter chips ranges from around 5 to 10 mA for both wireless transmission and reception. Lastly, the nRF5340 chip from Nordic Semiconductors stands out as the only chip that offers a dual core processor, which may be advantageous if used by an application that could benefit from a multicore architecture.

Furthermore, the survey also found that the semiconductor industry for Matter is in constant growth and evolution, since most chip manufacturers are developing new Matter chip families. However, manufacturers are not expected to introduce new or replace their Matter development platforms, but to just update them to add support for their new chips. Meaning that even with the continuous growth of the Matter chip market, the comparisons and evaluations of development platforms realized in this paper are still valuable for future developers.

TABLE I. COMPARISON OF AVAILABLE SEMICONDUCTORS THAT SUPPORT MATTER

Part Number	Manufacturer	Solution Type	Protocol Support	Flash [kB]	RAM [kB]	Core	Max Tx Output Power [dBm]	Tx Current (@0 dBm) [mA]	Rx current [mA]	Operating Voltage [V]
MG24	Silicon Labs	Standalone	Matter, Thread, BLE	1536	256	Cortex-M33 (78 MHz)	19.5	5.0	5.1	1.71-3.8
MG21	Silicon Labs	Thread RCP	Matter, Thread, BLE, Zigbee	1024	96	Cortex-M33 (80 MHz)	20	9.3	9.5	1.71-3.8
MR21	Silicon Labs	Thread RCP	Matter, Thread, BLE	512	64	Cortex-M33 (80 MHz)	20	9.3	9.5	1.71-3.8
nRF52840	Nordic Semiconductors	Standalone	Matter, Thread, BLE, Zigbee, NFC	1024	256	Cortex-M4 (64 MHz)	8	6.4	6.3	1.7-5.5
nRF5340	Nordic Semiconductors	Standalone	Matter, Thread, BLE, Zigbee, NFC	1024 + 256	512 + 64	Cortex-M33 (128 MHz) + Cortex-M33 (64 MHz)	3	4.2	3.7	1.7-5.5
CC2674x10	Texas Instrument	Standalone	Matter, Thread, BLE, Zigbee	1024	296	Cortex-M33 (48 MHz)	5	7.3	6.4	1.8-3.8
CC2652x7	Texas Instrument	Standalone	Matter, Thread, BLE, Zigbee	704	152	Cortex-M4F (64 MHz)	5	7.3	6.4	1.8-3.8
CC2652x	Texas Instrument	Thread RCP	Matter, Thread, BLE, Zigbee	352	88	Cortex-M4F (64 MHz)	6	7.3	6.9	1.8-3.8
K32W0x	NXP	Standalone	Matter, Thread, BLE, Zigbee	640	152	Cortex-M4 (48 MHz)	11	7.4	4.3	1.9-3.6
K32W148	NXP	Standalone	Matter, Thread, BLE, Zigbee	1024	128	Cortex-M33 (96 MHz)	10	4.6	4.7	1.8-3.6

V. EVALUATING THE DEVELOPMENT PROCESS FOR MATTER

A proof-of-concept test was performed to evaluate the complexity of developing software using Matter after selecting two chips, from the previous survey, alongside their development platforms. Two of the most basic smart home products were developed with each selected chip: a smart light bulb and a smart light switch. Evaluation board kits from each chip were obtained to run the tests. Lastly, a comparison was performed between the performance of each chip and the quality of their development platforms and processes. The selected chips for the tests were the MG24 from Silicon Labs and the K32W0x from NXP.

A. Thread Border Router with a Raspberry Pi

A Thread Border Router is required to create a Matter network using Thread. A simple way to create a Thread Border Router is to use a Raspberry Pi with a Thread RCP, since the Raspberry Pi lacks the necessary radio hardware to use Thread. Fortunately, Silicon Labs offers all the necessary tools to create this border router device.

The first step is to download the Matter Hub Raspberry Pi Image from the Silicon Labs website [8]. This image is Linux based and contains all the software tools to convert a Raspberry Pi into a Matter controller and commissioner. The Raspberry Pi Imager software was used to flash the image into an SD card that was later inserted into the Raspberry Pi. The second step is to configure the RCP for the Raspberry Pi. The XG24-EK2703A board from Silicon Labs was selected for its compatibility with the previously downloaded Raspberry Pi image. The board was easily flashed using the Simplicity Studio software from Silicon

Labs, which provides an already compiled binary for the RCP firmware. The XG24-EK2703A board was later connected to a USB socket of the Raspberry Pi. The border router device requires an ethernet cable with internet access to work properly. Finally, the command `"mattertool startThread"` can be used to start a new Matter network using this border router device.

B. Developing Matter devices with MG24 from Silicon Labs

To develop and simulate the smart light bulb and smart light switch, the `"EFR32xG24 Pro Kit +10 dBm"` evaluation board was selected. The development ecosystem of Silicon Labs runs on Windows and consists of their proprietary integrated development environment, or IDE, named Simplicity Studio 5. This software provides useful tools such as an energy profiler to measure the current consumption of the MG24 chip, as well as an extended list of example projects, and a graphical tool to configure the endpoints, clusters, and attributes of a Matter device.

The Simplicity Studio IDE was used to download an example project, named `"MatterLightOverThreadSolution"`, to serve as a base for the light bulb device. This example project is available for multiple chips and evaluation boards from Silicon Labs and contains all the necessary source code to simulate a smart light bulb. The example project was compiled and flashed into the development board using the Simplicity Studio IDE. Similarly, to develop the smart light switch device the example project named `"MatterLightSwitchThreadSolution"` was downloaded, compiled, and flashed using the Simplicity Studio IDE.

To commission the new smart devices into the Matter network first each device must be power up, one at a time, before

running the command “*mattertool bleThread*” in the Raspberry Pi configured as a Thread Border Router. It is important to write down the Node ID of each device that appears at the end of the commissioning process in the command line of the Raspberry Pi. The last step was to bind both devices, so a press of *button 1* on the light switch board would toggle the state of the smart light bulb. To bind the devices the following two commands were used:

- *mattertool accesscontrol write acl* '[{ "fabricIndex" : 1 , "privilege" : 5 , "authMode" : 2 , "subjects" : [112233] , "targets" : null } , { "fabricIndex" : 1 , "privilege" : 3 , "authMode" : 2 , "subjects" : {Light Switch Node ID} , "targets" : null }]' {Light Bulb Node ID} 0
- *mattertool binding write binding* '[{ "fabricIndex" : 1 , "node" : {Light Bulb Node ID} , "endpoint" : 1 , "cluster" : 6 }]' {Light Switch Node ID} 1

C. Developing with K32W0x from NXP

To develop and simulate the smart light bulb and smart light switch the “*IOTZTB-DK006*” advanced development kit was selected. The development ecosystem of NXP runs primarily on Linux and it does not provide a proprietary IDE as opposed to Silicon Labs.

NXP provides a useful guide that was followed closely during the development process [9]. The development process consisted of cloning the NXP-Matter repository from GitHub and downloading the current version (2.6.13) of the Software Development Kit (SDK) for the K32W061DK6 development board. Inside the cloned repository NXP provides a list of example projects ready to be compiled, similar to what Silicon Labs does with its Simplicity Studio IDE. For more explicit details on how to perform these steps follow the “*Getting started with K32W0x1 in Matter*” guide from NXP [9].

To compile the smart light bulb project, first navigate to the example project’s path “*matter/examples/lighting-app/nxp/k32w/k32w0*” and then run the following commands:

1. *gn gen out/debug --args="chip_with_OM15082=0 chip_with_ot_cli=0 is_debug=false chip_crypto=\"platform\" chip_with_se05x=0 chip_pw_tokenizer_logging=true chip_enable_ota_requestor=false"*
2. *ninja -C out/debug*

The compilation process generates a binary file named “*chip-k32w0x-light-example.bin*” in the path “*matter/examples/lighting-app/nxp/k32w/k32w0/out/debug*”. For the flashing process of the binary file into the K32W061DK6 board a Windows environment was needed to install the DK6 Programmer tool provided by NXP. To flash the firmware to the board the following command was used in a command prompt on a Windows system:

```
DK6Programmer.exe -s COM9 -p chip-k32w0x-light-example.bin
```

It is important to mention that NXP does not provide an example project for a smart light switch device, which is unusual considering every other manufacturer does. NXP only provides an example project that combines in a single device both a smart light switch and a smart light bulb. The path for this example project is “*examples/light-switch-combo-app/nxp/k32w/k32w0*” and the compilation and flashing processes are the same as with the light bulb device with the only difference being that the name of the output binary file is “*chip-k32w0x-light-switch-combo-example.bin*”.

To commission both devices into the Matter network, first each device must be powered up, one at a time, and then, the user interface button on the board must be pressed before running the “*mattertool bleThread*” command on the Raspberry Pi or Thread Border Router. To bind the devices, the same binding commands from the previous section must be used, making sure to change the Node IDs for the new IDs obtained during the commissioning process.

D. Comparing both developing platforms

The development platform of Silicon Labs is the better option compared with the development platform of NXP. The Simplicity Studio IDE from Silicon Labs offers a better way to download example projects, to organize the files of a Matter project while also offering a powerful graphical tool to configure the endpoints and clusters of a device. In comparison, NXP provides a huge repository of around 25GB that the developer must download in its entirety to be able to compile any example project. In comparison, both example projects from Silicon Labs had a combined size of 1GB. Moreover, the way the NXP repository is organized causes the compilation process to be confusing. It is unclear which files from the entire NXP repository are part of a project, making it difficult for a developer to modify any existing projects. Lastly, developing with NXP required switching between Linux and Windows which adds an extra step of complexity. Table II compares the development characteristics of the platforms tested.

TABLE II. COMPARISON OF MATTER DEVELOPMENT PLATFORMS

Manufacturer	NXP		Silicon Labs	
	Light bulb	Light switch	Light bulb	Light switch
Chip	K32W0x	K32W0x	MG24	MG24
IDE Support	Does not support an IDE		Simplicity Studio 5 IDE	
Operating System Support	Linux and Windows		Windows	
Project Size	25.5 GB		899 MB	

Manufacturer	NXP		Silicon Labs	
Device type	Light bulb	Light switch	Light bulb	Light switch
Codebase Complexity	Hard	Hard	Medium	Medium
Compilation Complexity	Medium	Medium	Easy	Easy
Flashing complexity	Medium	Medium	Easy	Easy
Modification of Project Complexity	Hard	Hard	Easy	Easy
Guides and Documentation	Outdated	Outdated	Complete	Complete
Compatibility with Apple's Homekit	Complete	Partial	Complete	Complete

To compare the current consumption of both Matter chips, an energy meter software tool provided by both manufacturers was used. In the case of the MG24 chip, the energy profiler tool inside the Simplicity Studio 5 IDE was used to measure the current consumption of the chip when the device was idle and when transmitting wirelessly. The obtained results are shown in “Fig. 4” and “Fig. 5”.

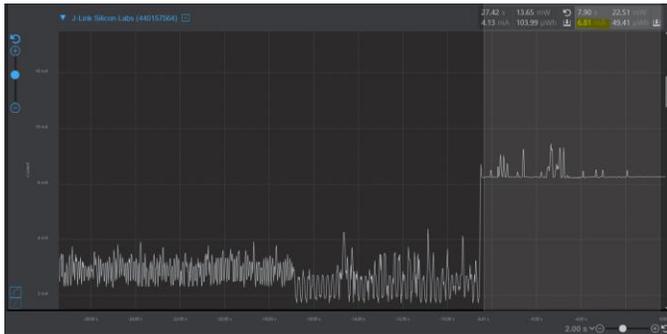


Figure 4. MG24 average current consumption of 6.81 mA while idle



Figure 5. MG24 peak current consumption of 8.75 mA when transmitting

In the case of the K32W0x chip, the energy measurement tool inside the MCUXpresso IDE, from NXP was used to measure the current consumption of the chip in the same scenarios as with the MG24 chip. The obtained results are shown in “Fig. 6” and “Fig. 7”.

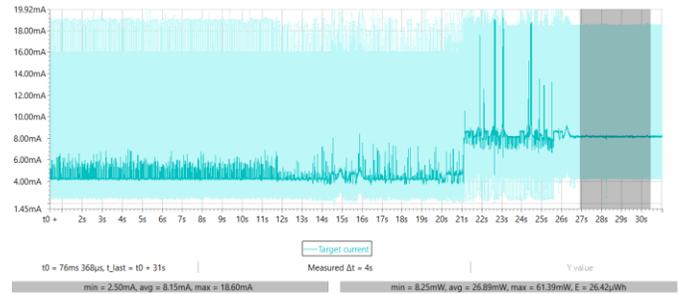


Figure 6. K32W0x average current consumption of 8.15 mA while idle

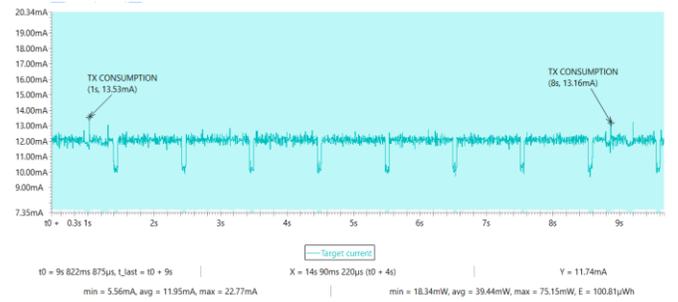


Figure 7. K32W0x peak current consumption of 13.53 mA when transmitting

To compare the wireless range of both chips, a line-of-sight test was performed where the light bulb and light switch of each chip were separated in increasing steps of five meters until they were no longer able to communicate with each other. Unfortunately, the testing site ran out of space at 65 meters before any of the paired devices lost communication.

Moreover, to compare the interoperability of the chips, and the Matter protocol, a third ecosystem was selected to evaluate how the developed devices performed while interacting with third party devices. The selected ecosystem was Apple’s Homekit, and as a Thread Border Router for this environment the Apple TV 4K was selected since this device has all the hardware requirements to communicate to other Matter devices using Thread. Both devices from Silicon Labs work seamlessly with Apple’s Homekit, and were added to the Homekit app following the simple instructions provided by the app. Once they were added to the ecosystem, the app granted full access to the devices features such as turning the light bulb on and off, and binding both devices together. The smart light bulb developed

with the NXP chip also succeeded in this test and was added to the network and controlled by the Homekit app in the same manner as with the Silicon Labs devices. The only device that presented an issue was the smart light switch from NXP because this device was developed using an example project that combined a light bulb with a light switch in a single device. The

issue was that the Homekit app detected this device as a light bulb only and did not expose an interface to bind or to control the light switch inside the device.

Table III summarizes the hardware and performance characteristics of the four developed devices.

TABLE III. COMPARISON OF CHARACTERISTICS BETWEEN MG24 AND K32W0X

Manufacturer	NXP		Silicon Labs	
	Light bulb	Light switch	Light bulb	Light switch
Device type	K32W0x	K32W0x	MG24	MG24
Chip	K32W0x	K32W0x	MG24	MG24
Core	Cortex-M4 (48 MHz)	Cortex-M4 (48 MHz)	Cortex-M33 (78 MHz)	Cortex-M33 (78 MHz)
Flash	640 kB	640 kB	1536 kB	1536 kB
RAM	152 kB	152 kB	256 kB	256 kB
Average current consumption when advertising using BLE	4.43 mA	3.96 mA	3.14 mA	2.86 mA
Average current consumption when commissioned and idle	8.15 mA	11.93 mA	6.81 mA	2.23 mA
Peak TX current consumption	NA	~13.53 mA	NA	~6 mA
Peak RX current consumption	~16 mA	NA	~8.75 mA	NA
Wireless Range	>65 m	>65 m	>65 m	>65 m
Development Kit Cost	\$525 USD		\$358 USD	
Single Chip Cost	\$5.73 USD		\$9.96 USD	

VI. CONCLUSIONS

In conclusion, the current state of Matter development platforms offers enough documentation and tools to support developers for integrating Matter in an embedded device. Furthermore, findings showed that Silicon Labs provides a superior development platform for Matter compared to NXP. Finally, the research done in this paper could be expanded in the future by evaluating the Matter development platforms from other manufacturers and by updating the survey of the available Matter chips.

REFERENCES

[1] J. Greenough, "The US smart home market has been struggling - here's how and why the market will take off," Business Insider, <https://www.businessinsider.com/the-us-smart-home-market-report-adoption-forecasts-top-products-and-the-cost-and-fragmentation-problems-that-could-hinder-growth-2015-9?r=MX&IR=T> (accessed May 29, 2024).

[2] "Amazon, Apple, Google, and the Alliance and its board members form Industry Working Group to develop a new, open standard for Smart Home Device Connectivity," CSA, <https://csa-iot.org/newsroom/connectedhomeip/> (accessed May 29, 2024).

[3] "Build with matter: Smart home device solution," CSA, <https://csa-iot.org/all-solutions/matter/> (accessed Jun. 20, 2024).

[4] "More than 5.5 billion smart home matter-compliant devices will ship between 2022 and 2030," ABI Research: The Tech Intelligence Experts, <https://www.abiresearch.com/press/more-than-55-billion-smart-home->

[matter-compliant-devices-will-ship-between-2022-and-2030/](https://www.abiresearch.com/press/more-than-55-billion-smart-home-matter-compliant-devices-will-ship-between-2022-and-2030/) (accessed May 29, 2024)

[5] I. Unwala, Z. Taqvi, and J. Lu, "Thread: An IOT protocol," *2018 IEEE Green Technologies Conference (GreenTech)*, Apr. 2018. doi:10.1109/greentech.2018.00037

[6] Matter Specification Version 1.3, 23-27349, 2024

[7] "The Device Data Model," Google Developer Center, <https://developers.home.google.com/matter/primer/device-data-model> (accessed Jun. 30, 2024)

[8] "Using the Pre-Built Raspberry Pi "Matter Hub" Image," Silicon Labs, <https://docs.silabs.com/matter/1.0.1/matter-wifi/build-pi-env> (accessed Jul. 1, 2024)

[9] "Getting started with K32W0x1 in Matter," NXP, <https://www.nxp.com/docs/en/user-guide/K32W0X1-MATTER-QSG.pdf> (accessed Jul. 1, 2024)