BASE MoVE - A Basis for a Future-proof IoT Sensor

Jens-Peter Akelbein[1] Kai Beckmann[2] Mario Hoss[3] Samuel Schneider[4] Stefan Seyfarth[5]
Marcus Thoss[6]

Abstract: For a long time, the Internet of Things was considered the vision of interconnecting every device, leading to fundamentally new and pervasive application scenarios. In practice, however, the projected growth and realisation of IoT scenarios is often impeded by practical problems. The BASE MoVE research project, a cooperation between universities and industrial partners, took a holistic look at requirements and inhibitors for investing in IoT solutions, using Ambient Assisted Living as an application domain example. The perspectives of all stakeholders involved were taken into account during the design of a solution architecture, from the user to the manufacturer to the service provider and housing association. This paper presents the resulting modular base platform for IoT applications. Power supply through battery and energy harvesting reduces installation costs. The use of open source software and the support of several common smart home protocols also prevents a lock-in effect and dependency on a single manufacturer. This makes it possible to protect investments from market-driven changes from one manufacturer's ecosystem to another. Here, the paper takes an in-depth look into the choice of protocols. Over-the-air updates allow for secure operation as well as remote maintenance, no longer requiring expensive in-person maintenance. Finally, the manufacturing of the solution as a hardware module, as realised in BASE MoVE, also allows for easier creation and certification of new sensor devices in a company's product portfolio. To evaluate the developed solution, an apartment was equipped with different sensor devices, and a smart home scenario was implemented. The feasibility study could demonstrate that it is indeed possible to create a base platform that meets today's requirements of the stakeholders involved and allows for a sustainable, future-proof usage by offering adaptability to new technologies. In addition to the scientific results, the project also gave an assessment about component maturity and cost, which is valuable for the commercial project partner and its market entry strategy.

Keywords: Internet of Things; Ambient Assisted Living; Home Automation

1 Motivation

The vision of an „Internet of Things“ (IoT) describes a technological revolution to be created by operating interconnected devices that pervade environments of personal life, work, and nature. Although distribution of such devices has been realised to a notable degree already, and solutions for many application areas, commercial and research platforms

[1] Darmstadt University of Applied Sciences, Schoefferstraße 8b, 64295 Darmstadt, jens-peter.akelbein@h-da.de
[2] RheinMain University of Applied Sciences, Unter den Eichen 5, 65195 Wiesbaden, kai.beckmann@hs-rm.de
[3] Darmstadt University of Applied Sciences, Schoefferstraße 8b, 64295 Darmstadt, mario.hoss@h-da.de
[4] Thermokon Sensortechnik GmbH, Platanenweg 1, 35756 Mittenaar-Offenbach, samuel.schneider@thermokon.de
[5] Thermokon Sensortechnik GmbH, Platanenweg 1, 35756 Mittenaar-Offenbach, stefan.seyfarth@thermokon.de
[6] RheinMain University of Applied Sciences, Unter den Eichen 5, 65195 Wiesbaden, marcus.thoss@hs-rm.de

doi:10.18420/inf2020_85

abound, there are numerous problems hindering further growth and flawless operation. Foremost, sustainability is still questionable, as the market has not yet stabilised, and new protocols and interfaces are still emerging. This, and the existence of so many variants calls for adaptability as a major feature of a design if it is meant to prevail. Without it, the growth of the IoT landscape is bound to decrease, or even cease.

It is therefore necessary to identify the relevant impeding factors that could endanger the future of the IoT. Functional correctness or applicability to the solution domain is rarely a problem. Instead, non-functional and platform-level factors must be considered critical for the success of a market solution. As, with increasing distribution of sensor nodes, most of these cannot use wired powering any more, and regular battery changes become impractical and too costly, self-sustained energy harvesting must become a mainstream technology. To further support sustainability, and the success of a platform solution, technological changes must be reacted upon by providing easy integration into future scenarios, should the protocol landscape change. This requires a high degree of adaptability and the possibility to update the firmware, and thus, possibly, support new protocols.

The project did not attepmt to re-invent base technologies readily available today. Nor was there a focus on offering a consistent application-level protocol and modelling solution, as investigated thoroughly in previous research projects. Instead, this project evaluated how a viable, flexible IoT platform could be created by integrating existing hardware, (embedded) operating systems (OS), and communications technologies. For the evaluation, an application scenario was implemented. Emphasis was also put on rendering the approach more future-proof with regard to technological changes by facilitating the exchange of communications technologies and protocols.

The acronym contained in the project name BASE MoVE states the main aspects regarded in the project. A dedicated sensor node hardware platform should be created as the **b**asis for the solution. **A**daptability is to be achieved by allowing over-the-air (OTA) firmware updates. **S**ecurity should be considered as a first-class design objective and thus be regarded from the very start, and finally, design for **e**nergy awareness at hard- and software level must lead to a viable self-powered sensor node architecture utilising energy harvesting technology. Application scenarios regarded in the project were meant to serve as technology test show cases for the validation of the fulfilment of these goals, whereas application level modelling was not considered a prime objective.

2 Related Work

Enabling real multi-protocol support in IoT-scenarios depends on the capabilities of the underlying hardware. If the transceivers are locked to specific radio protocols, the freedom to change or replace a protocol is limited. Most flexibility can be achieved if the radio transceivers are generic and the implementation of a specific protocol is a matter of hardware configuration and software.

There are hardware platforms like the EFR32 series from Silicon Labs [Sib] providing System-on-a-Chip (SoC) solutions embedding generic radio transceivers which allow for the usage of different protocols with the same device. Moreover, a subset of the EFR32 series embeds transceivers for different radio frequencies, like 2.4GHz and the sub-GHz band (915 resp. 868MHz or 433MHz). With their RAIL library, Silicon Labs provides several proprietary Smart-Home protocol stacks for their SoC family, like ZigBee, Thread, BLE, Z-Wave etc., and supports the creation of firmware running two stacks in parallel, like BLE and Zigbee [Sia].

Most hardware vendors provide proprietary protocol stack implementations for the application areas they are targeting. For the broader area of the IoT this can only cover a subset. There are many surveys like [SJ17], [Za18] or [Di19] gathering and categorising the protocols proposed and used, and solutions in the IoT or the smart home sector. Regarding the term „IoT“, there is a consolidating trend towards IP-based protocols noticeable in recent surveys. The emergence of IP for IoT protocols becomes apparent, considering the Zigbee Cluster Library having been made available for IP [Do], or the work towards IP over BLE [Ni15]. This is a significant change from the former situation, with manufacturers selling products combining hardware and software. Open vendor-independent protocol stack implementations for the IoT are provided by open source IoT operating systems, like RIOT OS [Ba18], mbed, Zephyr, Contiki or others [Qu18].

3 Architecture

Figure 1 shows the architecture for the IoT platform developed. It consists of the modular hardware platform and the software layer composed of an IoT OS, exchangeable protocol stacks, management and firmware update functionality and the top-level applications. The aim is to provide a flexible platform for fast and cost-efficient development of smart home and related IoT devices, supporting operation based on energy harvesting. The IoT protocols are kept exchangeable within the base platform to prevent lock-in effects for the OEM and customers. Furthermore, the usage of an open source OS is proposed to protect sold and installed smart home devices in households from becoming abandon-ware.

One important requirement for the *hardware* is its featuring low power modes that allow for energy harvesting devices. The transceiver should support arbitrary protocols, like BLE and 802.15.4, and should be able to support future protocols by using a generic transceiver. The selected Mighty Gecko EFR32MG13P7-33F512GM48 from Silicon Labs meets these requirements. It was manufactured as a PCB module to simplify certification and re-usability. Several iterative versions of four different devices were created. 1) A relay actuator with a permanent power supply. 2) A window contact with environment sensors and a reed contact. 3) A room control unit additionally equipped with a low power display and control buttons. 4) An occupancy sensor, replacing the display with a passive infrared (PIR) sensor. The types 2) to 4) are passive (end devices), reacting on external or timer events. They feature an energy harvesting subsystem and a monocrystalline solar cell for indoor use.

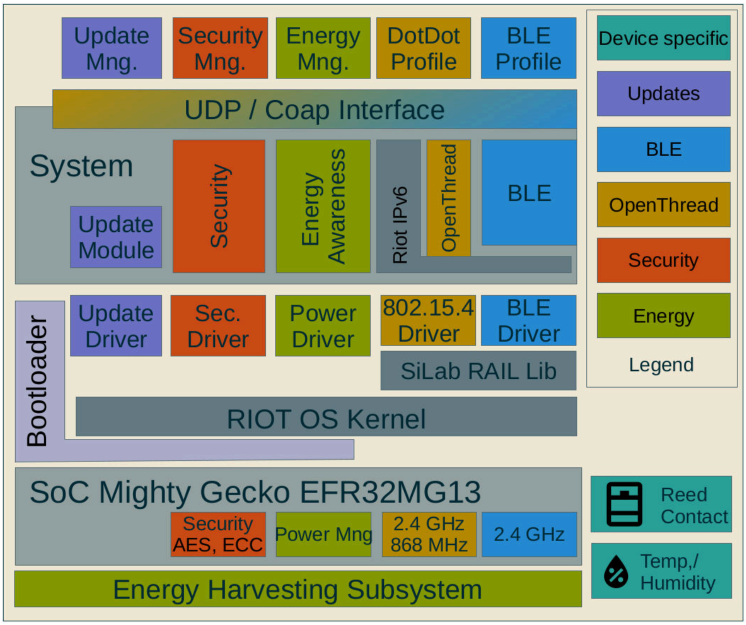

Fig. 1: Architecture of the "BASE MoVE" IoT platform

The OS as the *software* foundation was chosen based on a requirements analysis. Important aspects were the support of an open compiler like GCC, debugging support and it being an active project. Collaboration with the developer community and the ability to extend the OS with new features had to be possible. Satisfying these requirements, RIOT OS was chosen, even if it did not yet support all of the required hardware and protocol stacks. This was possible as the authors were familiar with RIOT OS, and estimated the development and integration of the missing functionality to be possible within the project.

The modular base platform supports exchangeable IoT protocols by flashing the device with different application images. The efforts to support multiple protocols within the same application, like Silicon Labs allows with their proprietary protocol stacks, were analysed and deferred due to the complexity and necessary modification of the OS. Furthermore, replacing the protocols through firmware update reduces the resource requirements to keep the devices viable, but requires firmware over the air (FOTA) functionality and the management of devices. To this aim, an additional management protocol was added, enabling the use of unaltered IoT protocol stacks. It is foreseeable that this approach will become obsolete, once a standard for FOTA is adopted by IoT protocols. On a device level an OS bootloader is used for a standard two slot approach.

The IoT platform supports wireless devices without a permanent energy source to reduce installation and maintenance costs. This comes at the cost of additional cross cutting concerns, limiting the possible functionality. In addition, the limited resources of the hardware platform and the required security level for OTA updates have to be considered throughout the system. To be a viable product, security requirements have to be met, while still allowing the operation with limited energy and memory requirements. Therefore, the platform requires both Security- and Energy-Awareness by Design.

For the management protocol, there was no established solution for constrained devices yet. Lightweight Machine to Machine (LWM2M) was evaluated[SA19], but could not be used due to resource requirements. Instead of a dedicated management protocol with the associated overhead, an update functionality is triggered over the Constrained Application Protocol (CoAP). A similar approach was also followed at the Software Updates for Internet of Things (SUIT)[So] IETF Taskforce which started work shorty before the beginning of the BASE MoVE project. The transmission of the image with parsing information in the form of a SUIT manifest should also enable protocol-independent transmission.

3.1 Multiprotocol

In the smart home sector, no consolidation of protocols is noticeable at this moment. On the contrary, new protocols are proposed, like the relatively new protocol „Thread“ developed by Google [Th]. To be able to demonstrate the multi-protocol approach of the BASE MoVE platform, a qualified subset of protocols had to be selected. This selection is differentiated between transport protocols, which distribute data between nodes (summarised OSI layers 1 - 6) and application protocols, defining the structure and semantics of the data exchanged (OSI layer 7). The requirements for this selection are: they have to be usable in smart home scenarios, support low power operations (potential utilisable with energy harvesting) and state-of-the-art security features. Furthermore, there have to be open source protocol stacks available, which can be used on the selected Mighty Gecko SoC hardware platform. For the application protocols, there should be standardised device profiles available enabling interoperability and connectivity between smart home devices.

The first selected group of protocols satisfying these requirements are IP-based. The 6LoWPAN protocol is the IETF standard to run „IPv6 over Low -Power Wireless Personal Area Networks“ [Mo07]. It is set on top of the IEEE 802.15.4 protocol, like ZigBee, and allows transparent mapping to standard IPv6 networks. There are several protocol stacks available, provided by different IoT OS. Using RIOT OS, its native 6LoWPAN stack is used. Additionally, the „Thread“ protocol is part of this group. Originally developed by Google Nest, it is now supported by different software and hardware vendors [Th]. Thread is based on 6LoWPAN, but replaces some parts and adds functionality, especially regarding security, deployment and routing. For this IoT platform, the open source stack „OpenThread“ [Op] is used, as there is a port to RIOT OS available. As second, different type of protocol BLE is selected. It is widely used in practice in many different smart home products, and virtually every smartphone offers connectivity and support. Furthermore, BLE Mesh, as a new standard extension, is going to provide the necessary mesh routing for more complex smart home scenarios in the future. For this IoT platform, the open source stack nimBLE from the Apache mynewt project is utilised, which was integrated by the RIOT OS community by the end of the BASE MoVE project. The selected application protocols are applied on top of the IP-based transport protocols. Again, there are several possibilities (see [SJ17]), but in this work, focus was on CoAP, another IETF standard. It was used for the application as well as the management part, because it is relatively lightweight and uses UDP. It can be integrated in edge and regular IT networks and there are several smart home and management protocols utilising CoAP.

For the support of application profiles based on CoAP, two approaches were incorporated. First, probably the most popular approach in practice is to define something new for a particular use case. As a first quick solution, the sensor data provided by the concrete devices were manually mapped to a REST structure and served by the RIOT OS CoAP implementation. The obvious limitation of the approach is that compatibility and interoperability are limited to the CoAP layer. As a second approach, a subset of Dotdot [Do] was prototypically implemented, which is a ZigBee Alliance standard mapping the well-established ZigBee Cluster Library (ZCL) and the ZigBee Device Profiles (ZDP) to a CoAP REST interface. Since the official specification of Dotdot was not openly accessible for most of the project time, the proof-of-concept realisation was based on information gathered from presentations, white papers and commercial implementations. It is limited to poll sensor data from end devices over CoAP and 6LoWPAN or the Thread protocol.

Regarding BLE, the application layer is part of the standard itself (Generic Attribute Profile - GATT). The Bluetooth Special Interest Group (SIG) defines the structure and semantics of data as Characteristics, the specific behaviour of a device functionality as Services, which are composed to Profiles defining the functionality of a type of device [GOP12]. Since the BLE stack within RIOT OS was only usable at the very end of the project time, very simple GATT services were implemented for the hardware platform as a proof-of-concept.

4 Application Scenario

To evaluate the base platform, an apartment was retrofitted to test its functionality in exemplary scenarios. The apartment was provided by one of the supporting industry partners, Vonovia SE.

The usage of BLE was planned, with IP over BLE for the OTA functionality, but due to delays for BLE support in RIOT, 6LoWPAN was used as a replacement protocol in the apartment instead. The behaviour and logic of the scenarios consist of a set of rules, executed on a smart home service platform. As such, the OpenHAB platform was chosen, necessitating the implementation of protocol bindings for CoAP and the application semantics defined for this project.

For the application scenarios, an exemplary ambient assisted living scenario (S1) as well as a smart home scenario (S2) were realised using the devices presented in chapter 3. For both scenarios, window sensors were attached to windows and doors reporting state changes (open/closed). In S1, a resident is warned when leaving the apartment if the windows are still open. So, if any window and the front door is open, a red warning light is switched on, and the forgotten window is displayed on a map next to the door. In S2, the room temperature is controlled depending on the resident's preferences, and the room lights are switched on or off depending on the room being occupied. Residents are using the room control unit to set a desired temperature, and, if the reported temperature from the window contacts surpasses this thresh-hold, an actor turns on a ventilator.

For a practical evaluation, these scenarios were implemented in three different rooms of the apartment. The first two rooms realised the scenarios with Thread and 6LoWPAN, respectively, to demonstrate the functionality. In the third room, the adaptability test case with FOTA functionality was evaluated, which needed two parallel OpenHAB deployments like in the first two rooms, each in a different IoT network and running a placeholder application. The devices were updated with firmware images, supporting the two different protocols, rotating them between the three applications.

5 Conclusion

The technical evaluation of the application scenarios has shown that the modular platform approach is feasible. The modular platform approach reduced the implementation effort for each new type of smart home device. In the end, the primary effort was on the hardware part. For the software, only drivers for new peripherals parts, some configuration and the additional application functionality were needed, which could largely be generated from ZigBee device profiles. Replacing an IoT protocol with another by firmware update was straightforward. The driver and protocol abstraction of the IoT OS allowed an easy creation of application images with new protocols. The issues encountered often stemmed from immature implementations. The early adaptation of Thread and DotDot, before the specification was available, provided some challenges. The implementation used, OpenThread, still had memory leak and energy consumption issues. While RIOT fulfilled the requirements set for this research project and offered great collaboration with the developer community, usage in a production environment would be premature. However, over the course of the project, clear improvements could be observed. The overall neglect of aspects regarding energy management throughout the design and the implementation, though, made the application in this scenario difficult. As a lesson learned, an adequate IoT-OS requires a holistic "Energy Awareness by Design". This topic is further explored in a new research project.

Furthermore, on a conceptual level, the evaluation of the application scenario has demonstrated that retrofitting existing living spaces with future-proof sensor devices is possible. It shows that sensor devices can be designed to adapt to new technologies, preserving the investment made despite being uncertain which technology will end up becoming the industry standard. Functional problems in the form of power consumption and stability issues with updates over Thread were caused by the use of immature implementations. In the apartment setup, an additional inhibitor for the general use of smart home scenarios was identified in the configuration time and complexity in OpenHAB. It is unlikely that the average end-user, facility manager or electrician installing the devices could implement the apartment-specific rules. Improvements in smart home service platforms are thus still needed for widespread adoption.

Bibliography

[Ba18] Baccelli, Emmanuel; Gundogan, Cenk; Hahm, Oliver; Kietzmann, Peter; Lenders, Martine S.; Petersen, Hauke; Schleiser, Kaspar; Schmidt, Thomas C.; Wahlisch, Matthias: RIOT: An Open Source Operating System for Low-End Embedded Devices in the IoT. IEEE Internet of Things Journal, 5(6):4428–4440, 2018.

[Di19] Dizdarević, Jasenka; Carpio, Francisco; Jukan, Admela; Masip-Bruin, Xavi: A survey of communication protocols for internet of things and related challenges of fog and cloud computing integration. ACM Computing Surveys, 51(6):1–29, 2019.

[Do] Dotdot, `https://zigbeealliance.org/solution/dotdot/`.

[GOP12] Gomez, Carles; Oller, Joaquim; Paradells, Josep: Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology. Sensors (Switzerland), 12(9):11734–11753, 2012.

[Mo07] Montenegro, G.; Kushalnagar, N.; Hui, J.; Culler, D.: Transmission of IPv6 Packets over IEEE 802.15.4 Networks. RFC 4944, September 2007.

[Ni15] Nieminen, J.; Savolainen, T.; Isomaki, M.; Patil, B.; Shelby, Z.; Gomez, C.: IPv6 over BLUETOOTH(R) Low Energy. RFC 7668, October 2015.

[Op] OpenThread, `https://openthread.io/`.

[Qu18] Qutqut, M. H.; Al-Sakran, A.; Almasalha, F.; Hassanein, H. S.: Comprehensive survey of the IoT open-source OSs. IET Wireless Sensor Systems, 8(6):323–339, 2018.

[SA19] Schmelzer, Paul; Akelbein, Jens-Peter: Evaluation of Hardware Requirements for Device Management of Constrained Nodes Based on the LWM2M Standard. In: CERC. 2019.

[Sia] Silicon Labs - Dynamic Multiprotocol, `https://silabs.com/wireless/multiprotocol`.

[Sib] Silicon Labs EFR32 Wireless Gecko Technology Features, `https://www.silabs.com/products/wireless/technology`.

[SJ17] Salman, Tara; Jain, Raj: A Survey of Protocols and Standards for Internet of Things. Advanced Computing and Communications, 1(1), March 2017.

[So] Software Updates for Internet of Things (suit), `https://datatracker.ietf.org/wg/suit/about/`.

[Th] Thread Stack Fundamentals, `https://threadgroup.org/ourresources`.

[Za18] Zaidan, A. A.; Zaidan, B. B.; Qahtan, M. Y.; Albahri, O. S.; Albahri, A. S.; Alaa, Mussab; Jumaah, F. M.; Talal, Mohammed; Tan, K. L.; Shir, W. L.; Lim, C. K.: A survey on communication components for IoT-based technologies in smart homes. Telecommunication Systems, 69(1), 2018.