Western Institute of Technology and Higher Education

Networks for Embedded Systems (Wireless)

O2023_ESI1118O

Estephania Martínez García

Practice 1 – IEEE 802.15.4

Nelida Paulina Hernández Moya

ie727824

Mauricio Peralta Osorio

ie728593

November 1st, 2023

Concepts

Throughout the development of this practice, we actively engaged with several key concepts, which not only facilitated a more robust implementation but also enhanced our understanding of the contextualization. These concepts served as fundamental building blocks for our work, contributing significantly to the overall success of the project:

IEEE 802.15.4 is a standard for low-power, short-range wireless communication, primarily used for enabling Internet of Things (IoT) and wireless sensor network (WSN) applications. In summary, IEEE 802.15.4 provides a set of specifications for devices to communicate wirelessly over relatively short distances (tens of meters to a few kilometers) while consuming minimal power. It is known for its energy efficiency, making it suitable for battery-operated and low-power devices, such as smart sensors, home automation, and industrial monitoring systems.

An **RTOS**, or Real-Time Operating System, is a specialized software system designed for applications that require precise and predictable timing and response. It manages hardware resources and tasks, ensuring that critical processes are executed within predefined time constraints. RTOS is commonly used in embedded systems, robotics, aerospace, and automotive applications.

The **PAN ID**, or Personal Area Network Identifier, in IEEE 802.15.4 is a unique identifier that distinguishes one wireless network or personal area network (PAN) from another. It helps devices within a PAN to identify and connect to the correct network. PAN IDs are a fundamental part of the addressing scheme in 802.15.4 networks, ensuring that devices communicate within the intended network.

In IEEE 802.15.4, a **channel** refers to a specific radio frequency on which devices within a wireless network communicate. The standard defines multiple channels with different frequencies, allowing for coexistence and reducing interference in wireless networks.

In IEEE 802.15.4, a **coordinator** is a special device within a wireless personal area network (WPAN) that serves as the central controller. It manages network resources, facilitates communication among devices, and may coordinate activities like channel selection, data routing, and network synchronization. The coordinator plays a key role in organizing and maintaining the network.

In IEEE 802.15.4, an **end device** is a node or device that communicates within a wireless personal area network (WPAN). End devices are typically sensor nodes, actuators, or other simple devices that gather or transmit data. They do not participate in managing the network and rely on a coordinator for network control and management.

The **MAC** (Media Access Control) stack in IEEE 802.15.4 is a part of the standard that defines the rules and protocols for accessing the shared wireless medium within a wireless network. It governs how devices transmit and receive data, handle channel access, and manage contention to ensure efficient and reliable communication.

Modifications in the Code

In the initial phase, we began by implementing the main task within the end device. This task is responsible for managing the LEDs based on the global account variable. The account variable is incremented by a timer, which triggers a callback function to update and reset it. In addition to this, we integrated keyboard interrupt handlers for the buttons to adjust the count according to the button pressed and reset the timer. This ensures that the count proceeds smoothly from that point onward.

As for the package transmission, we utilized the TransmitUART function, making minor adjustments as needed. Essentially, we transmitted the current account number each time it was updated.

In phase one for the coordinator, the approach involved accessing the value within a union by using a pointer and adding the address. Additionally, a function was implemented to manage the LEDs following the same logic as in the end device, which depended on the current account.

In the end, to output the requested data, we simply employ serial prints to access the data structure.

Regrettably, as a result of time limitations, we find ourselves unable to proceed with the full execution of phase 2 at this moment. Our current focus was dedicated to the completion of phase 1, alongside the subsections within the domain of phase 2.

In the context of phase two, we have successfully accomplished the first two objectives. In these tasks, we developed the requisite data structure and employed a flag and an index to ascertain whether the device already possessed an extended address. Subsequently, we either assigned a new address or retained the existing one, allowing us to integrate this information into the established data structure.

Furthermore, we tried to transmit the necessary message using the same UART function. However, what eluded us was the exact location from which to retrieve the "RxOnWhenIdle" parameter, whether for direct communication or polling, as it appeared solely within the GDP structure. Despite our thorough review of the documentation, the implementation details of this aspect, as well as the final procedure on the end device to signal its capabilities, remained unclear to us.

Conclusions Nelida

The development of this project was undeniably challenging, but it provided us with a wealth of valuable insights as we delved into the code, documentation, and their practical applications. We encountered the need to introduce new functionalities as initially planned and make adjustments to existing ones. While we were unable to successfully complete the entire project, we can confidently assert that we confronted and learned from the majority of the issues encountered during the process.

Initially, we embarked on the task of comprehending the utilization of tasks, timers, and events, although the latter were eventually excluded due to practicality concerns. However, grasping this information was critical not only for implementation but also for subsequent modifications. We also made use of the UART function for transmitting data, which, in our view, may not have been the most expedient choice but was deemed suitable for the purpose.

In the case of the coordinator, we encountered some challenges when attempting to access the account due to the union that rendered the reading invalid. Fortunately, we found pointers to be immensely helpful in resolving this issue, alongside the utilization of data structures to manage the information effectively.

In conclusion, it was a good experience to translate the theoretical knowledge of IEEE 802.15 into practical applications. This endeavor provided us with a rich learning opportunity, enabling us to gain valuable insights and tackle numerous challenges that arose during the process.

Conclusions Mauricio

The main challenge was to identify how to create and reset the timer correctly. Afterward, we had difficulties understanding the operation and synchronization of the end device with the coordinator. To overcome this, we had to read the information and observe the debug flow to understand how it works.

Additionally, identifying the structure where the received and transmitted data was located proved to be challenging. However, once we read the documentation, some things became much simpler, and we successfully completed most of the practice.

I learned how to configure message types, where to receive information such as the address source, device type, and message reception, among other things contained in the structure.

GitHub Repository

https://github.com/hanilizalo23/P1_RSE_E1_Wireless.git

Implementation Video

RSE Wireless - 802.15.4.MOV