VRIJE
UNIVERSITEIT
BRUSSEL

Software and Engineering for Embedded Systems

Controlling Multiple BLE Devices using a BLE-WiFi Gateway

Group Members:
Mehdi Hedayatrasa
Mikhail Korotych

Table of Contents

1 Introduction

The utilization of the Internet of Things (IoT) has transformed the area of consumer and industrial technology these days. This transformation is particularly evident in the development and deployment of smart devices within various sectors such as home automation, healthcare, industrial control, and environmental monitoring. The BLE—Wi-Fi gateway project can also be totally constructive regarding these developments. It aims to build a gateway that bridges the communication gap between low-power Bluetooth Low Energy (BLE) devices and the broader Wi-Fi network. Utilizing the Cypress PSoC6 microcontroller, this project aims to create a seamless interface that not only enables remote control and monitoring but also ensures scalability and robustness.

This initiative is anticipated to bring up significant advancements in how BLE devices integrate into larger network infrastructures, ultimately enhancing operational efficiency and user accessibility.

2 Background

2.1 Technical Context

Bluetooth Low Energy (BLE) is an essential technology in the IoT area, known for its low energy consumption and capacity to operate on small, battery-powered devices such as wearables, sensors, and home automation controls [1] Integrating BLE devices with Wi-Fi networks extends their utility by connecting them to the Internet, thereby amplifying their range and functionality. The Cypress PSoC 6 platform is selected for its dual-core architecture, which effectively manages computational and communication tasks, and its integrated BLE and CapSense capabilities, making it ideal for diverse IoT applications [2].

2.2 Previous Work

Various implementations of BLE–Wi-Fi gateways currently exist in specific areas like health monitoring systems and smart wearables. These implementations often focus on particular aspects such as data security in health devices or enhancing connectivity in consumer electronics [3]. However, these solutions commonly illustrate limitations in adaptability, power efficiency, and cost, which restrict their applicability across different sectors. The proposed project could address these shortcomings by developing a gateway that not only supports extensive customization but also optimizes both power efficiency and operational cost, thus broadening the potential for widespread IoT integration.

2.3 Significance

This project's significance is underscored by its potential to streamline the integration of BLE devices with existing network infrastructures, simplifying the creation of expansive IoT ecosystems. The gateway facilitates both remote and automated control over BLE devices through Wi-Fi connectivity, enhancing device interoperability and user experience. The expected outcomes include more efficient resource management and improved functionalities in smart environments, contributing to the ongoing evolution of smart technology applications [4].

3 Description

As mentioned in previous sections, the main objective of this project is utilizing a BLE—Wi-Fi gateway that is designed to bridge the communication gap between BLE devices and a Wi-Fi network, enabling the control and monitoring of BLE-equipped devices remotely.

To demonstrate the feasibility of this control, remote switching of the red and green LEDs mounted on the board will be performed. The connection between the Bluetooth devices and the computer will be established via a gateway. Here, it follows more specific descriptions for this process.

3.1 Hardware Description

- *The PSoC™ 6 BLE Prototyping Kit (CY8CPROTO-063-BLE)*. The Cypress PSoC 6 microcontroller offers a dual-core architecture with one core dedicated to applications and the other to managing wireless communications. Its integrated BLE capability makes it ideal for IoT applications. [5] Here, the green and red LEDs (`LED4` and `LED3` respectively) are interesting. They will be toggled after the chip programming with the help of the PSoC™ Creator [6].
- *The Dusun DSGW-021-5-EU wireless router and gateway*. This device was selected for its dual-band capabilities and support for both Wi-Fi, Bluetooth protocols and UART interface. [7] It is also able to directly connect with BLE devices, local networks and transmit data to local servers.

The general scheme of the communication within the network is on Figure 1.

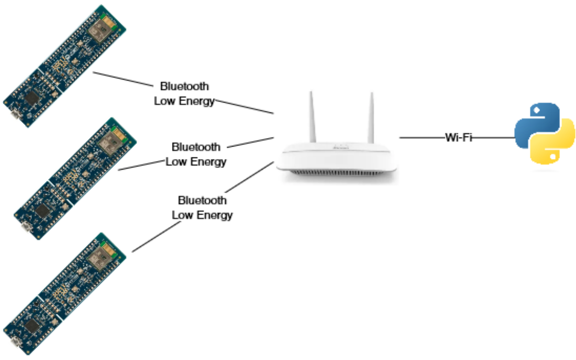

Figure 1. The BLE–Wi-Fi communication.

3.2 Software Description

- *The PSoC® Creator™*. This IDE is needed to properly program the microcontroller. This stage includes: enabling the BLE module, advertising the name and MAC address of a device, publication of the so-called "services" and "characteristics" which provides the outer control over the corresponding chip. [8]
- *The managing application*. The application that is used in this work is written in the Python programming language [9] and behaves as a client that can easily access the gateway and connected Bluetooth devices via its web API in order to manage the PSoC. All the API requests are based on the HTTP protocol, sometimes the parsing of HTML pages takes place.

The successful exchange process is demonstrated on Figure 2.

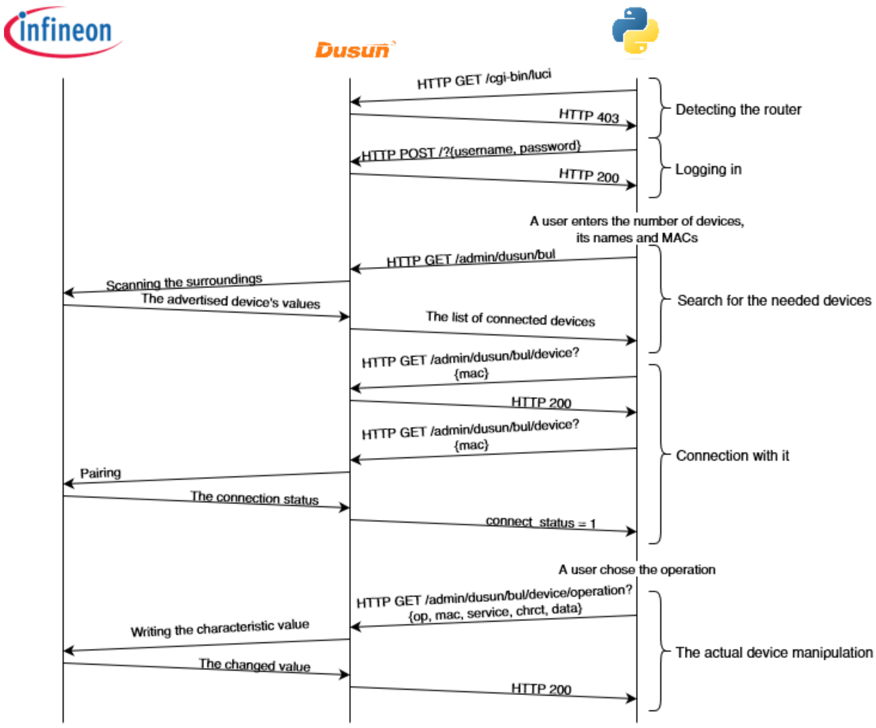

Figure 2. The diagram of the PSoC–client communication based on the HTTP.

4 Experimental Results

The solution we have implemented consists of two independent modules. The first module is a project in the PSoC® Creator™ programme (the design diagram and pins of the components is shown below).

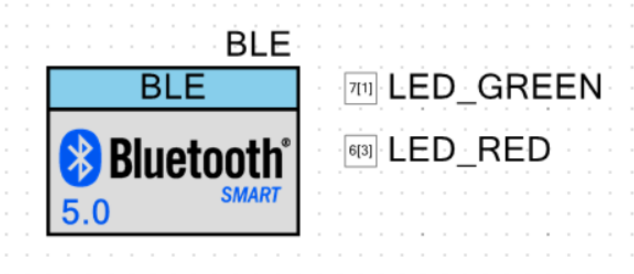

Figure 3. The top design of the project.

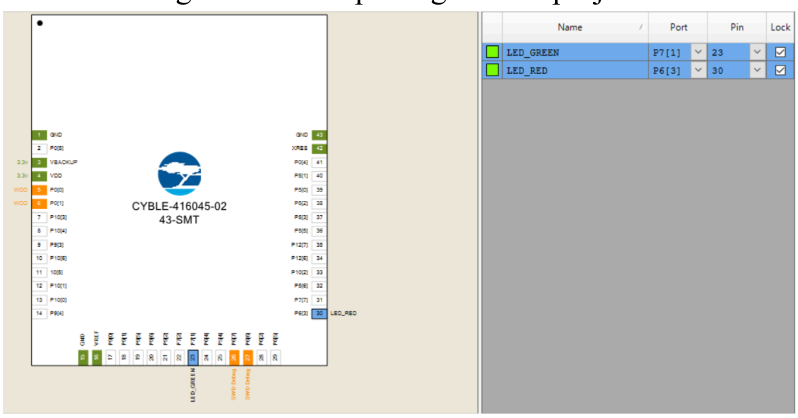

Figure 4. The pin design of the project.

For the required behaviour of the microcontroller, the handler of external requests to write characteristics has been modified (fig. 5).

```c
case CY_BLE_EVT_GATTS_WRITE_REQ: // When the other side sends the write request
    writeReqParameter = (cy_stc_ble_gatts_write_cmd_req_param_t *)eventParameter;

    /* Toggles the green LED when the value "true" is sent through the BLE GREEN_LED_TOGGLE characteristic.*/
    if (CY_BLE_DEVICE_INTERFACE_GREEN_LED_TOGGLE_CHAR_HANDLE == writeReqParameter->handleValPair.attrHandle) {
        bool toggle = writeReqParameter->handleValPair.value.val[0];
        if (toggle) {
            Cy_GPIO_Inv(LED_GREEN_PORT, LED_GREEN_NUM);
            toggle = false;
        }
    }

    /* Toggles the red LED when the value "true" is sent through the BLE RED_LED_TOGGLE characteristic.*/
    if (CY_BLE_DEVICE_INTERFACE_RED_LED_TOGGLE_CHAR_HANDLE == writeReqParameter->handleValPair.attrHandle) {
        bool toggle = writeReqParameter->handleValPair.value.val[0];
        if (toggle) {
            Cy_GPIO_Inv(LED_RED_PORT, LED_RED_NUM);
            toggle = false;
        }
    }

```

Figure 5. The `main_cm0p.c` file fragment that has been changed.

As for the Python project, it contains one file main.py with 12 functions and 4 constants. Their list in the appearance order is given below:

- `ROOT_PATH` — the URL of the router main web page in a local network;
- `UNKNOWN_SERVICE_UUID` — the generated by the PSoC® Creator™ universally unique identifier of the service;
- `GREEN_LED_TOGGLE_CHRCT`, `RED_LED_TOGGLE_CHRCT` — the generated by the PSoC® Creator™ universally unique identifiers of the characteristics;
- `check_connection` — detection of the router in the local network;
- `logging` — entering the user name and the password to control devices connected to the router;
- `print_dic` — printing the device name and its MAC address;
- `prepare_devices_info` — obtaining the device names and its MAC addresses (optionally) to control to;
- `search_for_devices` — trying to detect devices from the previous function;
- `search_for_mac` — searching for the MAC addresses which have not been entered in the previous function;
- `connect_to_devices` — parallel connection to the devices by its MAC;
- `get_device_by_mac` — connection to a single device, is used by the previous function;
- `toggle_led` — sending the request to rewrite the characteristic of the corresponding LED on the board;
- `control_menu` — the main menu to choose devices to manipulate;
- `enter_device_number` — entering the existing number of the device to manipulate.

5 Discussion and Conclusion

In the process of work many variants of interaction of Wi-Fi and BLE protocols were tested. It should be noted that Infineon provides a mobile application “AIROC™ Bluetooth® Connect - Mobile App” [10] for communication with Bluetooth Low Energy devices, but the goal of the project was to communicate with them through the gateway, not directly.

Initially, we proposed two approaches:

1. to use the gateway vendor's SDK to create an embedded C solution. We rejected it because of outdated documentation of the vendor, impossibility to compile the SDK for the target system, and also because of the catastrophic lack of space in the overlay section (a feature of OpenWRT OS);
2. Also, due to the small amount of free permanent memory it was not possible to implement a Python programme running remotely on the gateway from the main device.

As a result, despite the fact that no software is embedded into the router's operating system during the process, the project goals were achieved. The PSoCs are managed completely remotely via a programme on the host device, while improving the portability of the solution and making it compatible with any gateway that has the same—or at least similar—GUI and API.

References

1. Townsend, K., Cufí, C., & Akiba. (2014). Getting Started with Bluetooth Low Energy: Tools and Techniques for Low-Power Networking. O'Reilly Media, Inc.
2. Cypress Semiconductor. (2018). PSoC 6: Purpose-Built for the IoT. Retrieved from Cypress Semiconductor
3. Lee, I., & Lee, K. (2015). The Internet of Things (IoT): Applications, investments, and challenges for enterprises. Business Horizons, 58(4), 431-440.
4. Zachariah, T., Klugman, M., Campbell, B., & Adkins, J. (2014). The Internet of Things Has a Gateway Problem. Retrieved from Link to source
5. https://www.infineon.com/cms/en/product/evaluation-boards/cy8cproto-063-ble/
6. https://www.infineon.com/cms/en/design-support/tools/sdk/psoc-software/psoc-creator/
7. https://www.dusuniot.com/product-specification/dsgw-021-lte-4g-dual-band-wifi-router-gateway/
8. https://infineon.github.io/bless/ble_api_reference_manual/html/index.html
9. https://www.python.org/
10. https://play.google.com/store/apps/details?id=com.infineon.airocbluetoothconnect