

BLOOD COAGULATION MONITORING SYSTEM

Artem Maslov, Student; Marta Mashevska, Ph. D., As.

Lviv Polytechnic National University, Ukraine, e-mail: marta.v.mashevska@lpnu.ua

<https://doi.org/10.23939/istcmtm2020.03.024>

Abstract. This article describes the creation of a communication system between smartphones and various medical devices, also there is an analysis of different approaches to data transfer between devices, such as manual data entry, screen scanning using a seven-segment display recognition system, and Bluetooth low energy. The practical part demonstrates a developed communication system using Bluetooth technology. There is also implemented an automatic communication between devices that requires no user actions.

Key words: Medical device, Low energetic Bluetooth, MVP-architecture.

1. Introduction

Implementing fast and user-friendly communication between different peripheral devices is essential in program development. That is why it is necessary to automate this process as much as possible.

2. The current state of the problem

Now there are few ways to implement communication between peripheral devices that come in handy in different situations. The most popular of them are:

1. Screen scanning – often seen when short and simple communication is needed between devices. Some messengers have it as a way to quickly give access to other devices or when scanning QR code. In terms of medical devices – it is usually used to read test results from the screen and write according to data to smartphones. The main advantage of such a method is that it works even if the medical device is relatively old and doesn't have Bluetooth, Wi-Fi, or other communication technologies. The main disadvantage is that text recognition doesn't have 100 % accuracy and sometimes scanning the screen may be inconvenient because of devices like tabs which are a bit bulky.

2. Bluetooth low energy (BLE) – one of the most up-to-date technologies. Just with a few clicks, the user can transfer any data between devices (Fig. 1). The only issue here is that not all medical devices have Bluetooth support. Otherwise, it has practically no downsides while dealing with medical devices. Hundreds of records are transmitted in a fraction of a second and handling this process from the developer side has gotten much easier over the past years.

3. Manual data entry – sometimes we have to deal with really old devices that have no Bluetooth support and screen recognition may not always work. So we have to foresee such cases and approach of a simple manual data entry should be implemented. The idea is simple – the user fills all the required data himself and an appropriate data record is added to the database. That approach is rather planned C at best since the process of communication should be as automated as possible.

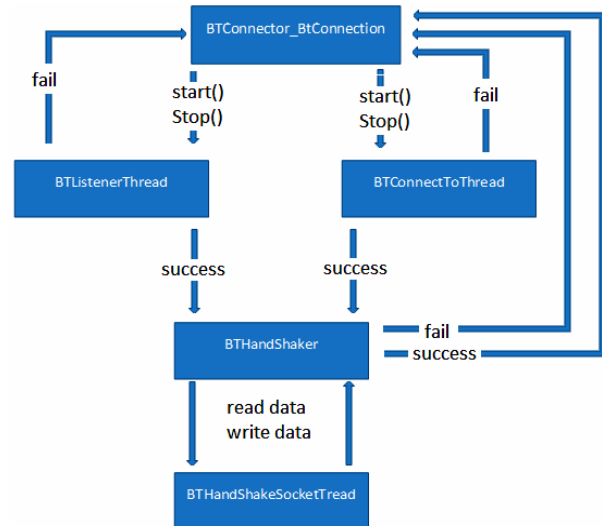


Fig. 1. BLE structure

Every technology has its benefits and limitations, and BLE is no exception. It's important to know these pros and cons to be able to determine whether BLE is suitable for your specific application and use case or not.

Benefits of BLE

- As the name indicates BLE has relatively low energy consumption compared to other low power technologies. BLE achieves optimized and low power consumption by keeping the radio off as much as possible and sending small amounts of data at low transfer speeds.
- No cost to access the official specification documents. To use most other wireless protocols and technologies, you would have to be a member of the official group or consortium for that technology to access the specification.
- Lower cost of modules and chipsets, even when compared to other similar technologies.
- Most importantly, its existence in most smartphones in the market.

Limitations of BLE

- **Permissions:** one of the issues both for the developer and for the user is that BLE requires few permissions – Bluetooth_ADMIN and ACCESS_FINE_LOCATION.

- **Data Throughput:** the data throughput of BLE is limited by the physical radio layer (PHY) data rate, which is the rate at which the radio transmits data. This rate depends on the current Bluetooth version. For Bluetooth 4.2 and earlier, the rate is fixed at 1 Mbps. For Bluetooth 5 and later, however, the rate varies depending on the mode and PHY used. The rate can be 1 Mbps like earlier versions, or 2 Mbps when utilizing the high-speed feature.

- **Range:** Bluetooth Low Energy (and Bluetooth in general) was designed for short-range applications and hence its range of operation is limited. There are a few factors that limit the range of BLE:

- BLE operates in the 2.4 GHz ISM spectrum which is greatly affected by obstacles that exist all around us such as metal objects, walls, and water (especially human bodies).

- Performance and design of the antenna of the BLE device.

- The physical enclosure of the device.
 - Device orientation.

- **Gateway Requirement for Internet Connectivity:** To transfer data from a BLE-only device to the Internet, another BLE device that has an IP connection is needed to receive this data and then, in turn, relay it to another IP device (or to the Internet).

3. The goal of the work

The goal of this work is to develop an approach that allows users to comfortably utilize and transfer data between the medical device and android applications.

Statement of the main material

Since we are going to review BLE technology as a preferred communication way here is a summary of key BLE terms and concepts (Fig. 2):

- **Generic Attribute Profile (GATT)** – The GATT profile is a general specification for sending and receiving short pieces of data known as “attributes” over a BLE link. All current Low Energy application profiles are based on GATT.

- The Bluetooth SIG defines many profiles for Low Energy devices. A profile is a specification for how a device works in a particular application. Note that a device can implement more than one profile. For example, a device could contain a heart rate monitor and a battery level detector.

- **Attribute Protocol (ATT)** – GATT is built on top of the Attribute Protocol (ATT). This is also referred to as GATT/ATT. ATT is optimized to run on BLE devices. To this end, it utilizes as few bytes as possible. Each attribute is uniquely identified by a Universally Unique Identifier (UUID), which is a

standardized 128-bit format for a string ID used to uniquely identify information. The *attributes* transported by ATT are formatted as *characteristics* and *services*.

- **Characteristic** – A characteristic contains a single value and 0-n descriptors that describe the characteristic’s value. A characteristic can be thought of as a type, analogous to a class.

- **Descriptor** – Descriptors are defined attributes that describe a characteristic value. For example, a descriptor might specify a human-readable description, an acceptable range for a characteristic’s value, or a unit of measure that is specific to a characteristic’s value.

- **Service** – A service is a collection of characteristics. For example, you could have a service called “Heart Rate Monitor” that includes characteristics such as “heart rate measurement”.

From the developer side, some things require understanding but overall the underlying structure of BLE was simplified compared to prior approaches. Here is a review of the main inner layers (Fig. 3, 4):

- The physical layer (PHY) refers to the physical radio used for communication and for modulating/demodulating the data. It operates in the ISM band (2.4 GHz spectrum).

- The Link Layer is the layer that interfaces with the Physical Layer (Radio) and provides the higher levels an abstraction and a way to interact with the radio (through an intermediary level called the HCI layer which we’ll discuss shortly). It is responsible for managing the state of the radio as well as the timing requirements for adhering to the Bluetooth Low Energy specification.

- **Direct Test Mode:** the purpose of this mode is to test the operation of the radio at the physical level (such as transmission power, receiver sensitivity, etc.).

- The Host Controller Interface (HCI) layer is a standard protocol defined by the Bluetooth specification that allows the Host layer to communicate with the Controller layer. These layers could exist on separate chips, or they could exist on the same chip.

- The Logical Link Control and Adaptation Protocol (L2CAP) layer act as a protocol multiplexing layer. It takes multiple protocols from the upper layers and places them in standard BLE packets that are passed down to the lower layers beneath it.

- **Read:** The client (app) reads the value of a characteristic or descriptor on the server (BLE device) and interprets it based upon a protocol that has been established beforehand. A medical device may have a character whose value represents the current blood coagulation value.

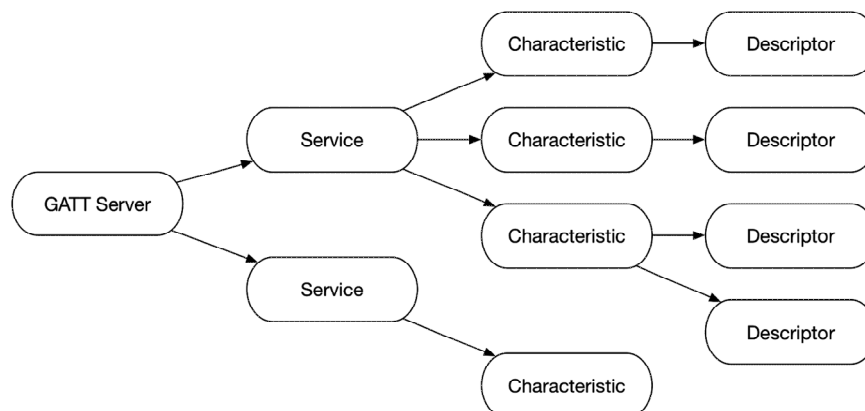


Fig. 2. BLE communication process

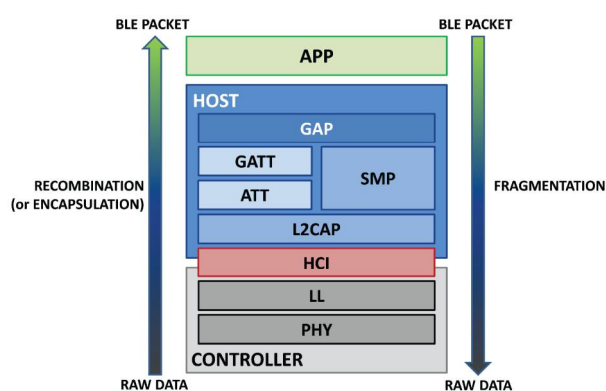


Fig. 3. BLE core architecture

All in all, there are 3 common BLE operations:

- **Write:** The client writes some bytes to a characteristic or descriptor on the server (BLE device). The server's firmware processes the write and performs some server-side operation in response to it. For example, a medical device may have a characteristic that changes database records when written to.

- **Notify/Indicate:** The client subscribes to a characteristic for notifications or indications, and is notified by the server when the value of the characteristic changes. A medical device may have a notifiable characteristic that will report the need to perform a test when it's subscribed to.

To develop such a system, Android Studio was chosen as a coding environment because it is where most Android applications are created and it is the most convenient and reliable tool given the amount of documentation written for developers. Android Studio is a free environment with an extremely large number of users, which simplifies the solution of many problems related to creating applications and setting up the environment itself.

The application development language is Java. Java has been a leader in this market for over 20 years

and is very user-friendly in combination with the aforementioned development environment.

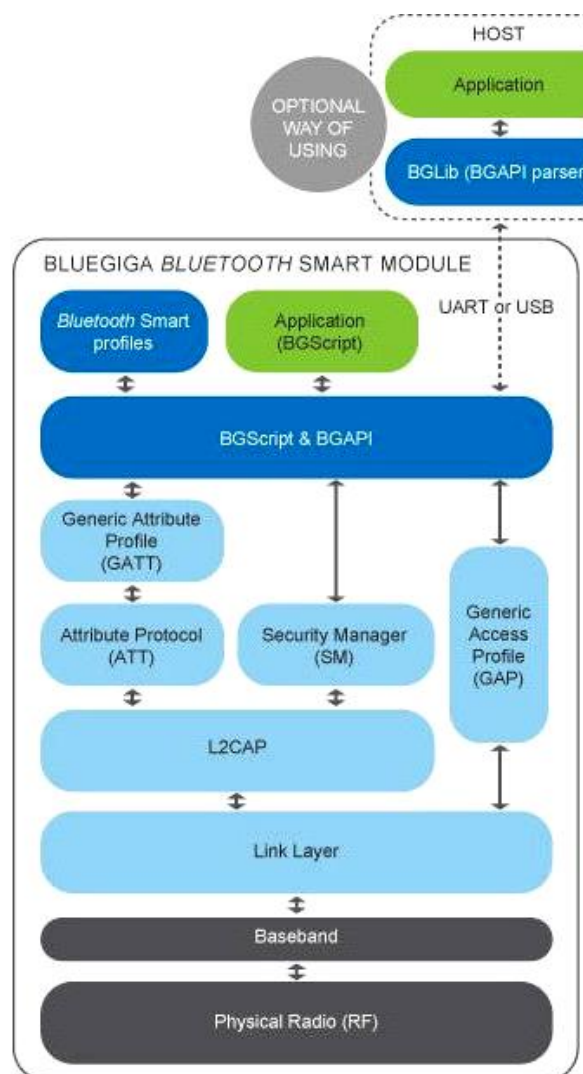


Fig. 4. Complete BLE stack

Practical implementation

BLE implementation fits nicely in MVP (Model View Presenter) app architecture which allows developers to make any changes to communication flow without affecting other app components. Also to reduce energy consumption both on a smartphone and medical device, the connection is canceled right after all data records have been transferred. After that smartphone checks, if there is a need to start looking for new results and whenever a medical device has new tests to advertise, the android device automatically connects and reads all necessary data.



Fig. 5. Appearance of the connected devices

Conclusions

Various methods of communicating between peripheral devices were reviewed here. As a conclusion, it is worth to say that technologies are developing lightning-fast and we all want to be up-to-date and use the newest and most convenient technologies. In that case, BLE is far superior compared to all others in terms

of user-friendly approach, speed, simplicity both for the developer and for clients, and most of all, reliability. Screen scanning does not guarantee an accurate result just as manual data entry, besides both these approaches require more actions from the user side which should be avoided. And yet implementing multiple methods is recommended to cover all cases like old devices without Bluetooth support etc.

References

- [1] Bluetooth Low Energy (BLE) [Online]. Available: <https://www.novelbits.io/basics-Bluetooth-low-energy/#:~:text=Benefits%20of%20BLE,access%20the%20official%20specification%20documents..>
- [2] Android BLE official documentation [Online]. Available: <https://developer.android.com/guide/topics/connectivity/Bluetooth-le>
- [3] Smart software stack [Online]. Available: <https://www.silabs.com/products/development-tools/software/bluegiga-Bluetooth-smart-software-stack>
- [4] BLE intensive introduction [Online]. Available: <https://atadiat.com/en/e-Bluetooth-low-energy-ble-101-tutorial-intensive-introduction/>
- [5] K. Townsend, C. Cufi, Akiba, R. Davidson, *Getting Started with Bluetooth Low Energy: Tools and Techniques for Low-Power Networking*. O'Reilly Media; 1st Edition, 2014.
- [6] Bluetooth vs. Bluetooth Low Energy [Online]. Available: <https://www.link-labs.com/blog/Bluetooth-vs-Bluetooth-low-energy>
- [7] BLE intensive introduction [Online]. Available: <https://blog.beaconstac.com/2018/08/ble-made-simple-a-complete-guide-to-ble-Bluetooth-beacons/>
- [8] BLE 101 [Online]. Available: <https://www.ostusa.com/blog/introduction-basics-of-Bluetooth-low-energy-ble/>
- [9] M. Afaneh, *The state of Bluetooth industry*, Belmont, CA: Wadsworth, 2016.
- [10] M. Julian, *Practical Monitoring: Effective Strategies for the Real World*. O'Reilly Media; 1st Edition, 2017.