TRANSMISJA BLUETOOTH LOW ENERGY (LE)
- rodzaje,
- jak to przebiega,
- realizacja transmisji przez rozne urządzenia

Poniżej cytuję następujące strony (źródła):
https://www.bluetooth.com/specifications/archived-specifications/
https://en.wikipedia.org/wiki/Bluetooth#Implementation
https://developer.android.com/guide/topics/connectivity/bluetooth
https://developer.android.com/guide/topics/connectivity/bluetooth-le
http://www.alphamicro.net/franchises/u-blox/nina-b3.aspx?fbclid=IwAR0BzxNb1IQktxzuAqD_48jRp6we9ZSD3OzubwujFFP_qUAZsGF5wS-49L8

Bluetooth
We distinguish Bluetooth BR/EDR and Bluetooth LE. Like the BR/EDR radio, the LE radio operates in the unlicensed 2.4 GHz ISM band.

Bit rate
LE Uncoded PHYs
LE 1M PHY
The mandatory symbol rate is 1 megasymbol per second (Msym/s), where 1 megasymbol represents 1 megabit. The 1 Msym/s symbol rate may optionally support error correction coding.
LE 2M PHY
An optional symbol rate of 2 Msym/s may be supported, with a bit rate of 2 Mb/s.
LE Coded PHY
This may use either of two coding schemes, supporting a bit rate of 500 kb/s or supporting a bit rate of 125 kb/s.

Access schemes
LE employs two multiple access schemes: Frequency division multiple access (FDMA) and time division multiple access (TDMA).

FDMA (smillar to BR/EDR)
Forty (40) physical channels, separated by 2 MHz. Three (3) are used as primary advertising channels and 37 are used as secondary advertising channels and as data channels.

TDMA
A TDMA based polling scheme is used in which one device transmits a packet at a predetermined time and a corresponding device responds with a packet after a predetermined interval.

The physical channel is sub-divided into time units known as events. Data is transmitted between LE devices in packets that are positioned in these events.

There are four types of events: Advertising, Extended Advertising, Periodic Advertising, and Connection events.

Devices that transmit advertising packets on the advertising PHY channels are referred to as advertisers. Devices that receive advertising packets on the advertising channels without the intention to connect to the advertising device are referred to as scanners.

Transmissions on the advertising PHY channels occur in advertising events. At the start of each advertising event, the advertiser sends an advertising packet corresponding to the advertising event type.

On the same advertising PHY channel: The scanner may make a request to the advertiser which may be followed by a response from the advertiser.

The advertising PHY channel changes on the next advertising packet sent by the advertiser in the same advertising event.

advertising events – unidirectional or broadcast communication, to establish pair-wise bidirectional communication devices using data channels

Devices that need to form a connection to another device listen for connectable advertising packets. Such devices are referred to as initiators. If the advertiser is using a connectable advertising event, an initiator may make a connection request using the same advertising PHY channel. The advertising event is ended and connection events begin if the advertiser receives and accepts the request for a connection be initiated.

Once a connection is established, the initiator becomes the master device in what is referred to as a piconet and the advertising device becomes the slave device. Connection events are used to send data packets between the master and slave devices.

In connection events, channel hopping occurs at the start of each connection event. Within a connection event, the master and slave alternate sending data packets using the same data PHY channel. The master initiates the beginning of each connection event and can end each connection event at any time.

Devices in a piconet use a specific frequency hopping pattern, which is algorithmically determined by a field contained in the connection request sent by an initiating device.

The hopping pattern used in LE is a pseudo-random ordering of the 37 frequencies in the ISM band. The hopping pattern can be adapted to exclude a portion of the frequencies that are used by interfering devices. The adaptive hopping technique improves Bluetooth co-existence with static (non-hopping) ISM systems when these are co-located and have access to information about the local radio environment, or detected by other means.

The hierarchy is physical channel, physical link, logical transport, logical link, and L2CAP channel.

Physical traffic is not equal to logical traffic (traffic on logical links is multiplexed onto the physical link)
The physical link is used as a transport for one or more logical links that support asynchronous traffic. Traffic on logical links is multiplexed onto the physical link assigned by a scheduling function in the resource manager. A control protocol for the link and physical layers is carried over logical links in addition to user data. This is the link layer protocol (LL).

Devices that are active in a piconet have a default LE asynchronous connection logical transport (LE ACL) that is used to transport the LL protocol signaling.

The Link Layer function uses the LL protocol to control the operation of devices in the piconet and provide services to manage the lower architectural layers (PHY and LL).

Abstraction to applications and services L2CAP layer
Just as in BR/EDR, above the link layer the L2CAP layer provides a channelbased abstraction to applications and services.
s. It carries out fragmentation and de-fragmentation of application data and multiplexing and de-multiplexing of multiple channels over a shared logical link. L2CAP has a protocol control channel that is carried over the primary ACL logical transport.

Addition protocols over L2CAP
In addition to L2CAP, LE provides two additional protocol layers that reside on top of L2CAP. The Security Manager protocol (SMP) uses a fixed L2CAP channel to implement the security functions between devices. The other is the Attribute protocol (ATT) that provides a method to communicate small amounts of data over a fixed L2CAP channel. The Attribute protocol is also used by devices to determine the services and capabilities of other devices.

The Bluetooth profiles
A Bluetooth profile is a wireless interface specification for Bluetooth-based communication between devices. An example is the Hands-Free profile. For a mobile phone to connect to a wireless headset, both devices must support the Hands-Free profile.

To use Bluetooth wireless technology, a device must be able to interpret certain Bluetooth profiles, which are definitions of possible applications and specify general behaviors that Bluetooth-enabled devices use to communicate with other Bluetooth devices. These profiles include settings to parameterize and to control the communication from the start. Adherence to profiles saves the time for transmitting the parameters anew before the bi-directional link becomes effective. There are a wide range of Bluetooth profiles that describe many different types of applications or use cases for devices.

Implementation of the Bluetooth protocol stack
Linux has two popular Bluetooth stacks, BlueZ and Fluoride. The BlueZ stack is included with most Linux kernels. Fluoride is included in Android OS.

Bluetooth LE and Android
Android 4.3 (API level 18) introduces built-in platform support for Bluetooth Low Energy (BLE).

In order for Bluetooth-enabled devices to transmit data between each other, they must first form a channel of communication using a pairing process. One device, a discoverable device, makes itself available for incoming connection requests. Another device finds the discoverable device using a service discovery process. After the discoverable device accepts the pairing request, the two devices complete a bonding process where they exchange security keys. The devices cache these keys for later use. After the pairing and bonding processes are complete, the two devices exchange information. When the session is complete, the device that initiated the pairing request releases the channel that had linked it to the discoverable device. The two devices remain bonded, however, so they can reconnect automatically during a future session as long as they're in range of each other and neither device has removed the bond.

Here are the roles and responsibilities that apply when an Android device interacts with a BLE device
Central vs. peripheral. This applies to the BLE connection itself. The device in the central role scans, looking for advertisement, and the device in the peripheral role makes the advertisement.

GATT server vs. GATT client. This determines how two devices talk to each other once they've established the connection.
To understand the distinction, imagine that you have an Android phone and an activity tracker that is a BLE device. The phone supports the central role; the activity tracker supports the peripheral role (to establish a BLE connection you need one of each—two things that only support peripheral couldn't talk to each other, nor could two things that only support central).

In the example used in this document, the Android app (running on an Android device) is the GATT client. The app gets data from the GATT server, which is a BLE heart rate monitor that supports the Heart Rate Profile. But you could alternatively design your Android app to play the GATT server role.

NINA-B3 Series - Bluetooth 5 low energy modules (component of ARDUINO NANO 33 BLE)
NINA-B30x series are small, stand-alone Bluetooth low energy microcontroller unit (MCU) modules. NINA-B30 features full Bluetooth 5. Applications can include Bluetooth low energy services such as GATT, beacons, and mesh.