



Разгоняем BLE

Nikita Kulikov

Android/Mobile @ Flipper Devices

Expert: Egor Markov

notAnotherOne



Куликов Никита

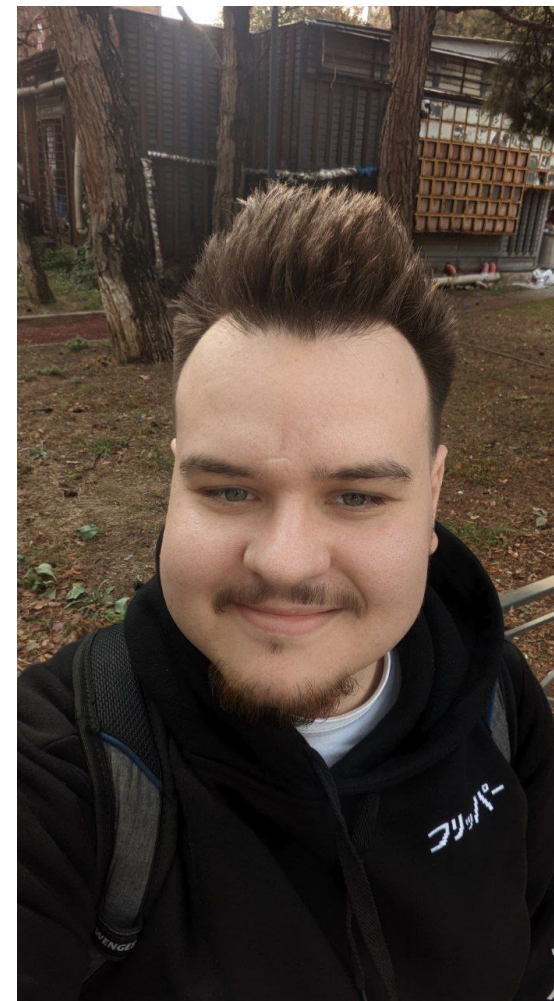
- Android/Mobile - Flipper Devices
- Cameos AR - Snapchat
- libVerify/libNotify - Mail.ru
- Yandex.Browser



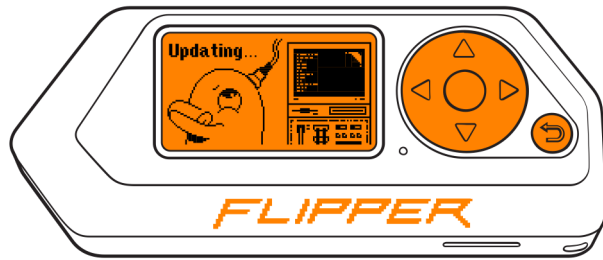
@LionZXY



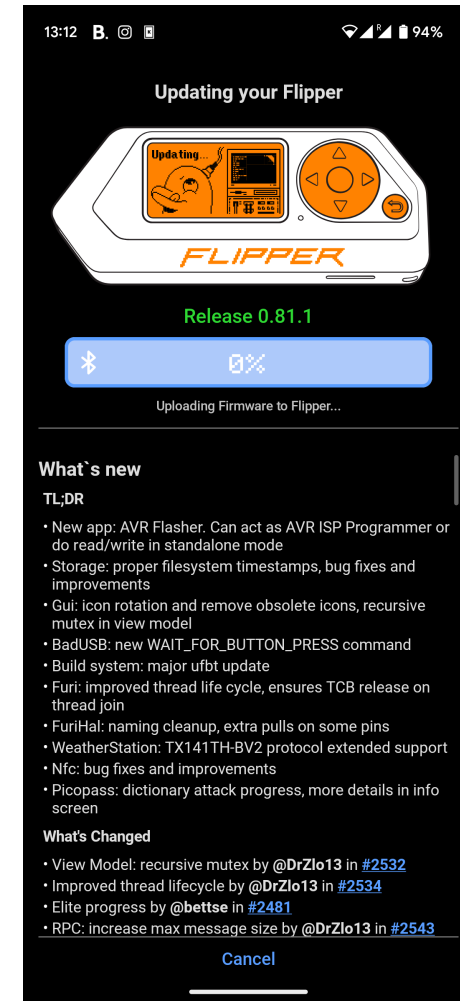
@localhost_ru



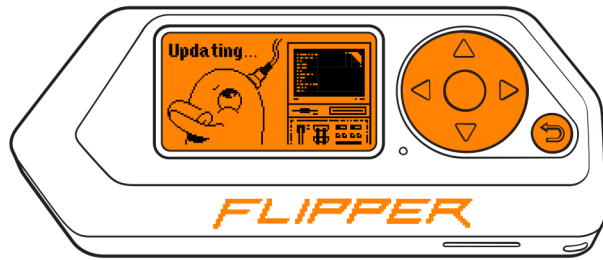
FLIPPER



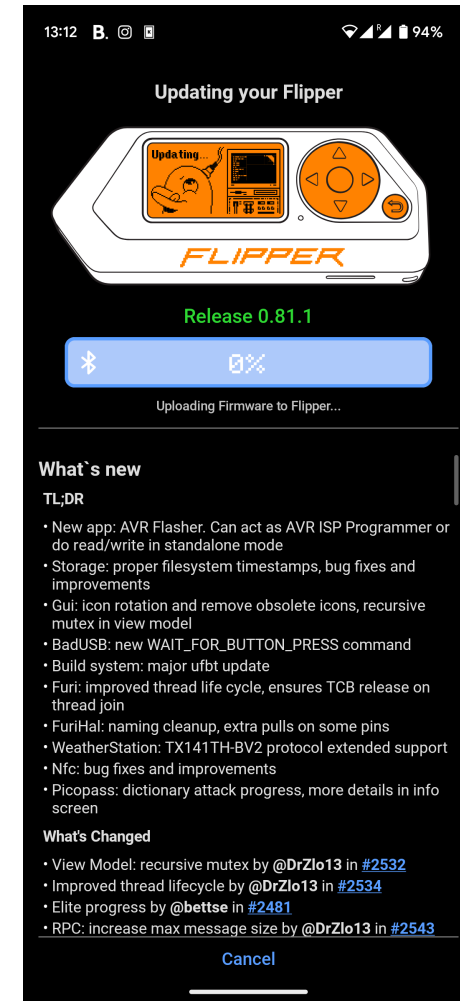
1.1 MB



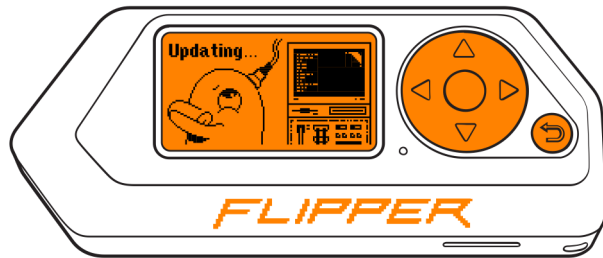
FLIPPER



1 115 323 bytes

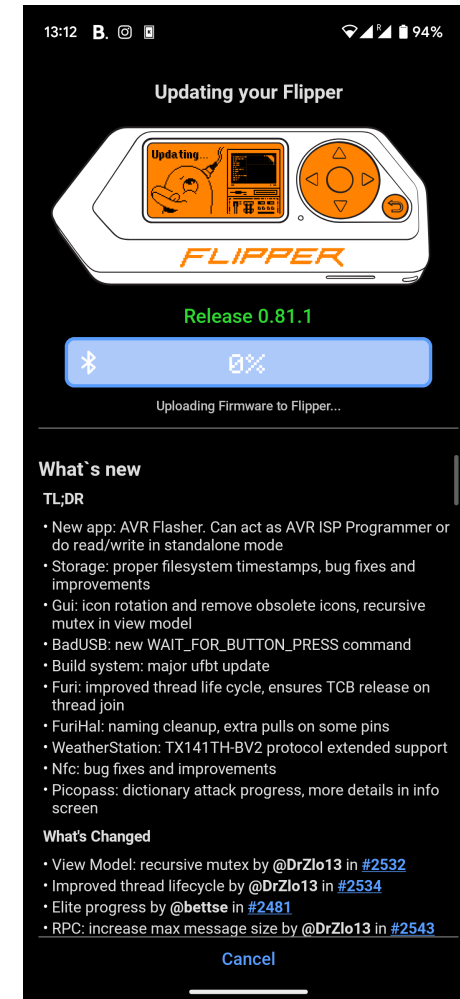


FLIPPER

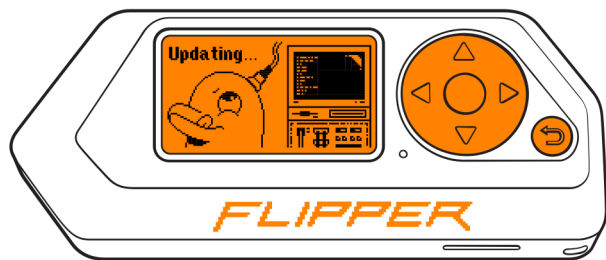


1 115 323 bytes

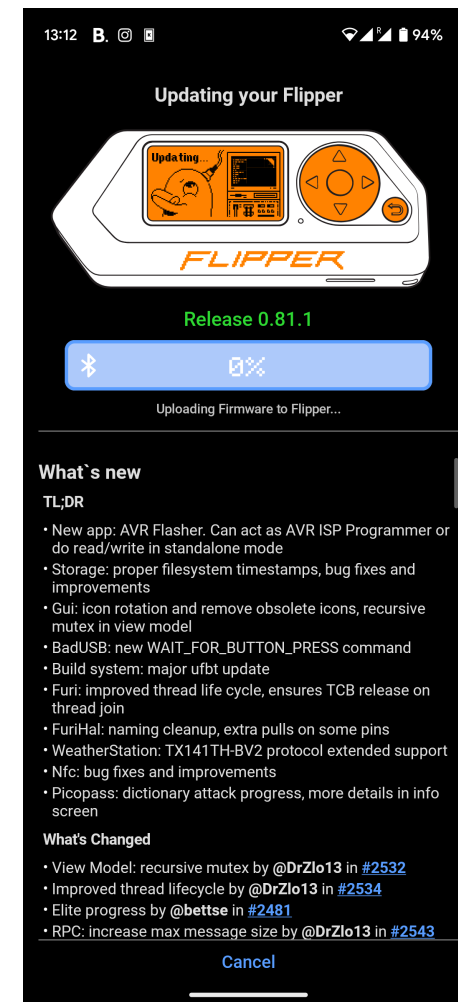
200 byte/s



FLIPPER

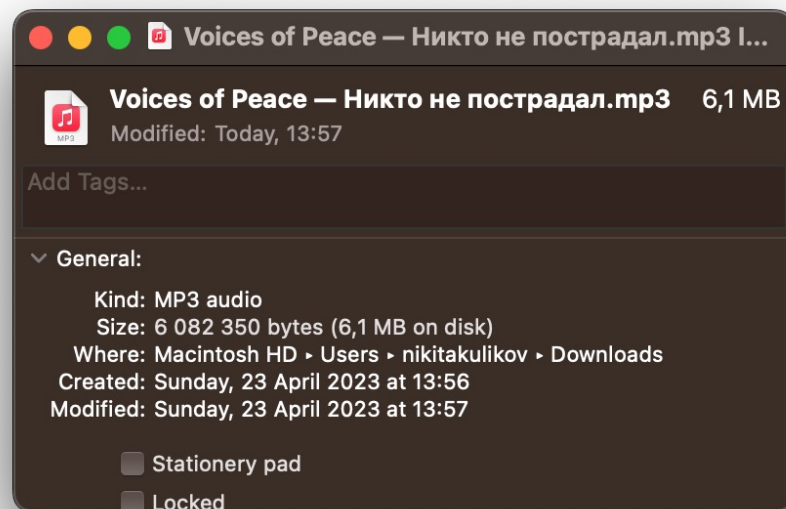


1 115 323 bytes
200 byte/s
1 час, 32 минуты





~ 3 часа



~ 8 часов
x168



~ 5 минут



Кому будет полезен доклад

- Вы хотите ускорить BLE
- Вы планируете работать с BLE
- Вы работаете с BLE
- Вы хотите понять как работает BLE

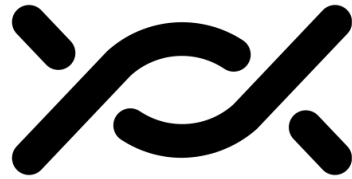


Содержание:

Что такое BLE	9
Как устроено BLE	14
Логический слой BLE	27
Физический слой BLE	48

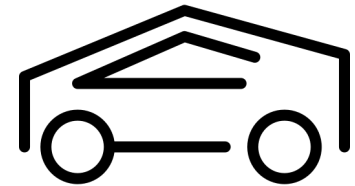
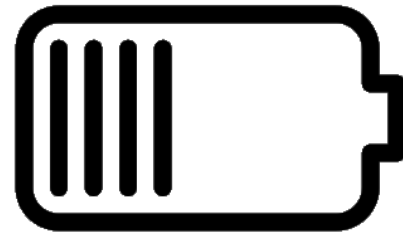
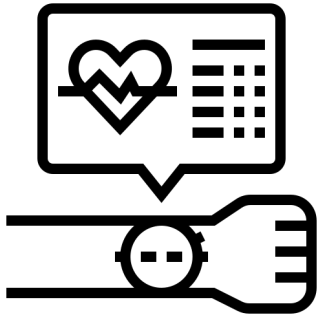


Bluetooth



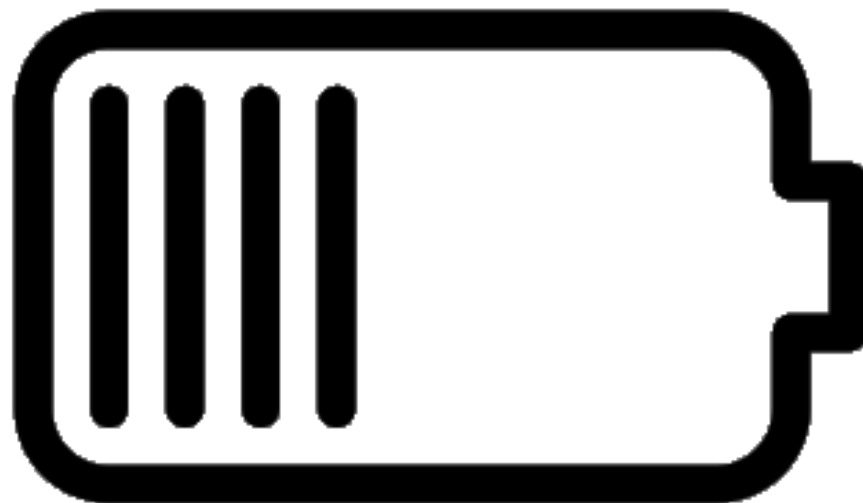


Bluetooth Low Energy





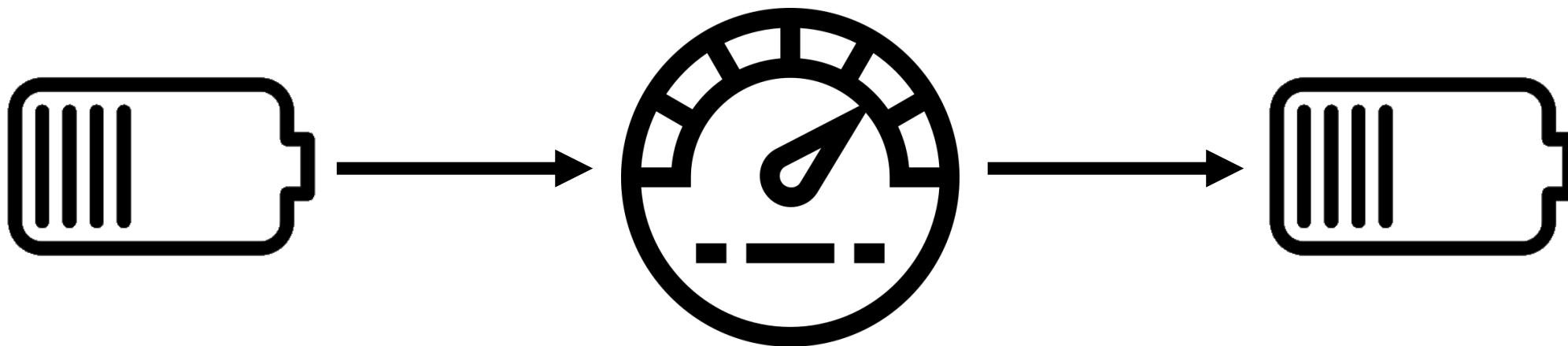
Bluetooth Low Energy



1 месяц автономности



Задача: поднять временно скорость



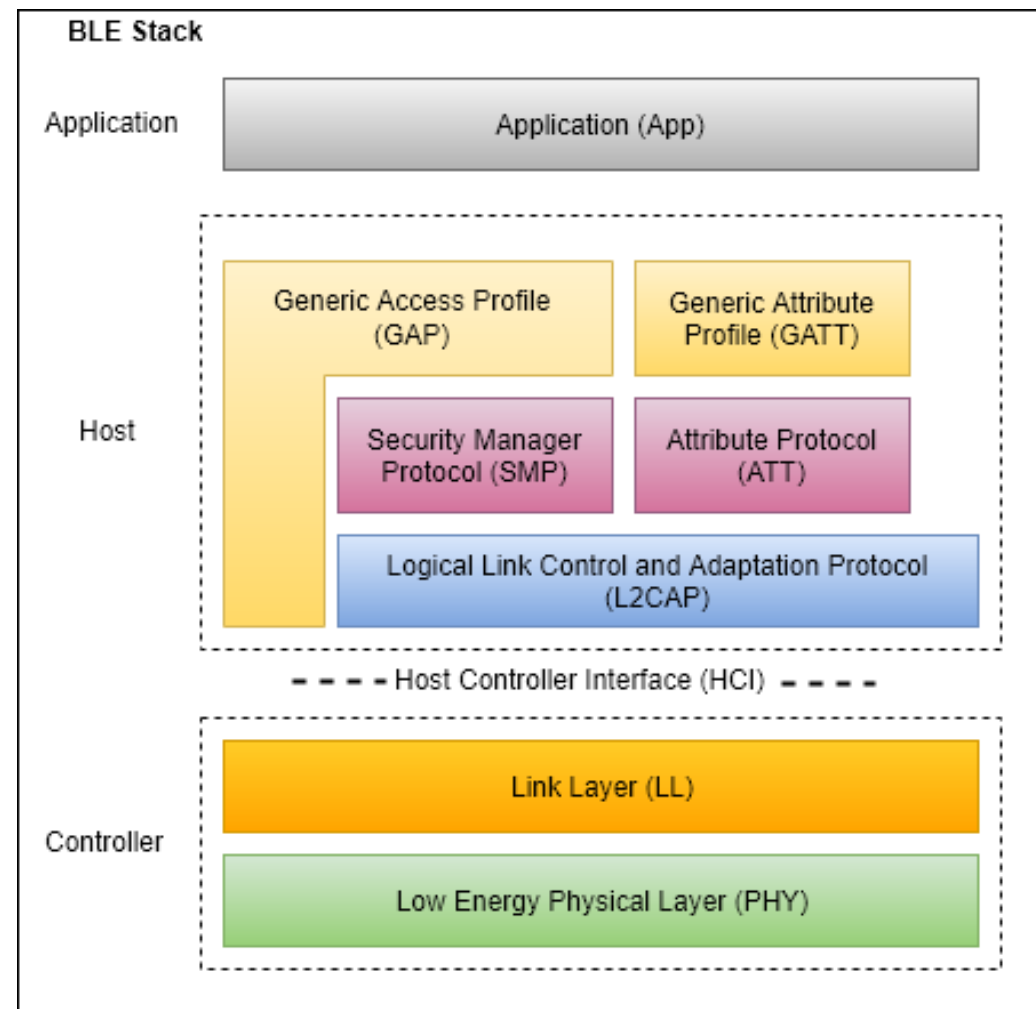


Содержание:

Что такое BLE	9
Как устроено BLE	14
Логический слой BLE	27
Физический слой BLE	48



BLE Stack



**HRSTM**

DISCONNECT

Status: CONNECTED

BONDED

DEVICE INFORMATION**0x180A**

PRIMARY SERVICE

MANUFACTURER NAME STRING

**UUID: 00002A29-0000-1000-8000-00805F9B34FB**

Properties: READ

Value:STM

Hex: 0x53544D00



Service

```
service 0000180d-0000-1000-8000-00805f9b34fb(  
    read char 19ed82ae-ed21-4c9d-4145-228e62fe0000,  
    read char 19ed82ae-ed21-4c9d-4145-228e61fe0000,  
    read write char 19ed82ae-ed21-4c9d-4145-228e63fe0000  
)
```



BLE API

```
struct Flipper(  
    let name: String,  
    let heartrate: Int,  
    var ownerName: String  
)
```

```
data class Flipper(  
    val name: String,  
    val heartrate: Int,  
    var ownerName: String  
)
```

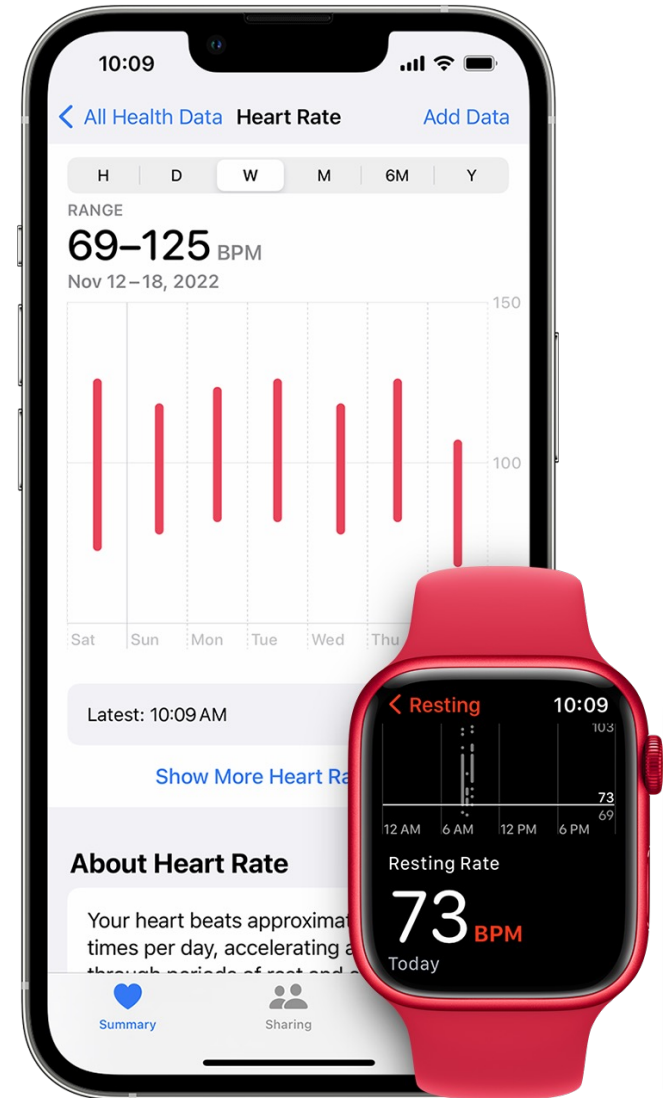


BLE API

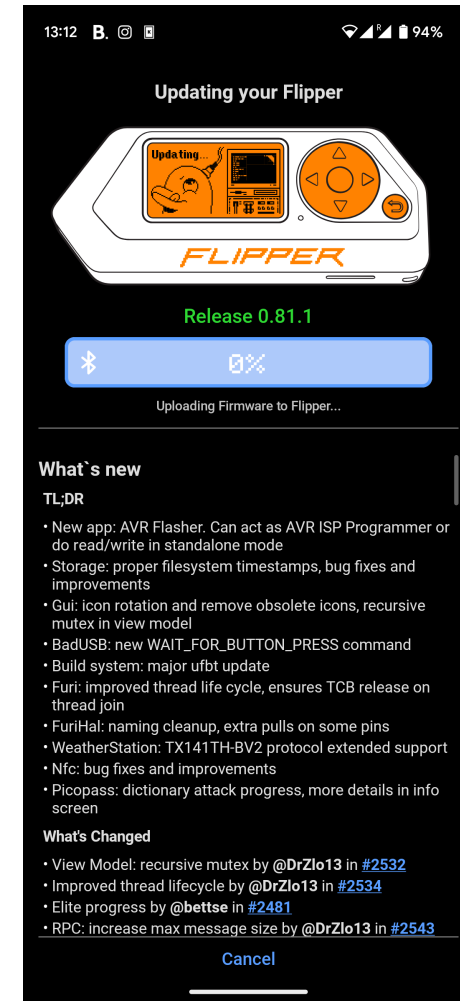
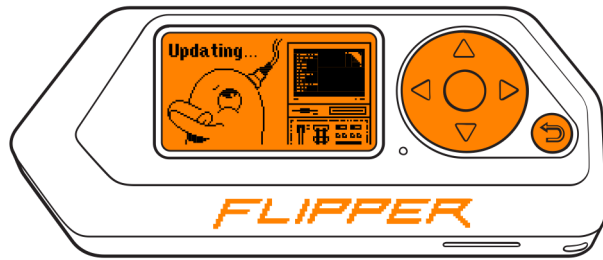
```
data class Flipper(  
    val name: String,  
    val heartrate: Int,  
    var ownerName: String  
)
```



Heart Rate



FLIPPER





Serial API

```
val data = byteArrayOf(1_000_000)
flipper.tx = data.split(0, 509)
flipper.tx = data.split(509, 1018)
flipper.tx = data.split(1018, 1527)
flipper.tx = data.split(1527, 2036)
```

FLIPPER



Serial Bluetooth Terminal

Kai Morich

Покупки в приложении

4,6★

2,45 тыс. отзывов

500 тыс.+

Количество скачиваний

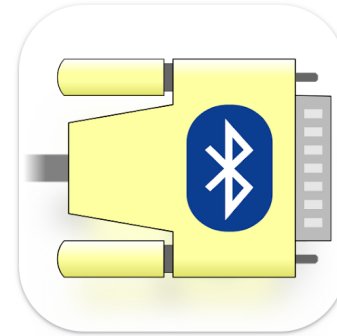
E

Для всех ⓘ

Установить на устройствах

📱 Это приложение можно скачать на некоторые из ваших устройств.

Open the Mac App Store to buy and download apps.



BLE Serial Pro 4+

nRF & CC2541 BLE UART terminal

Ed Nieuwenhuys

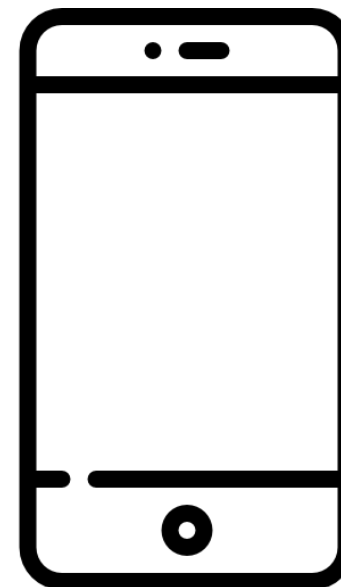
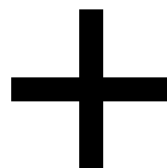
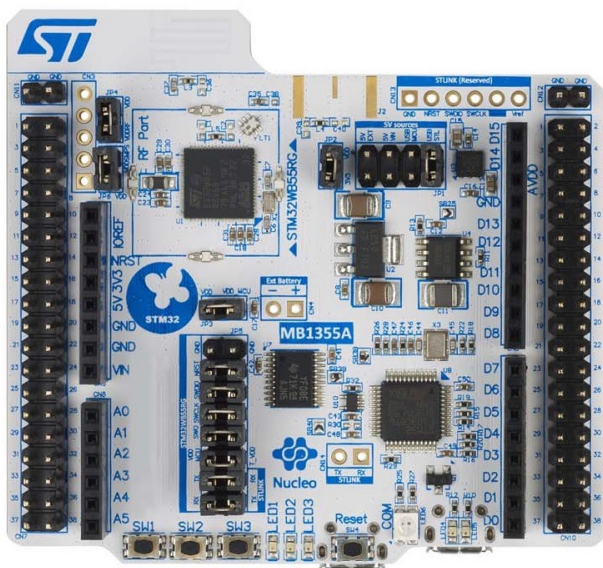
Designed for iPad

19,99 lei

[View in Mac App Store ↗](#)



Sample





LionZXY/MobiusBLEPerformance

mobile

stm32





Замеры Init



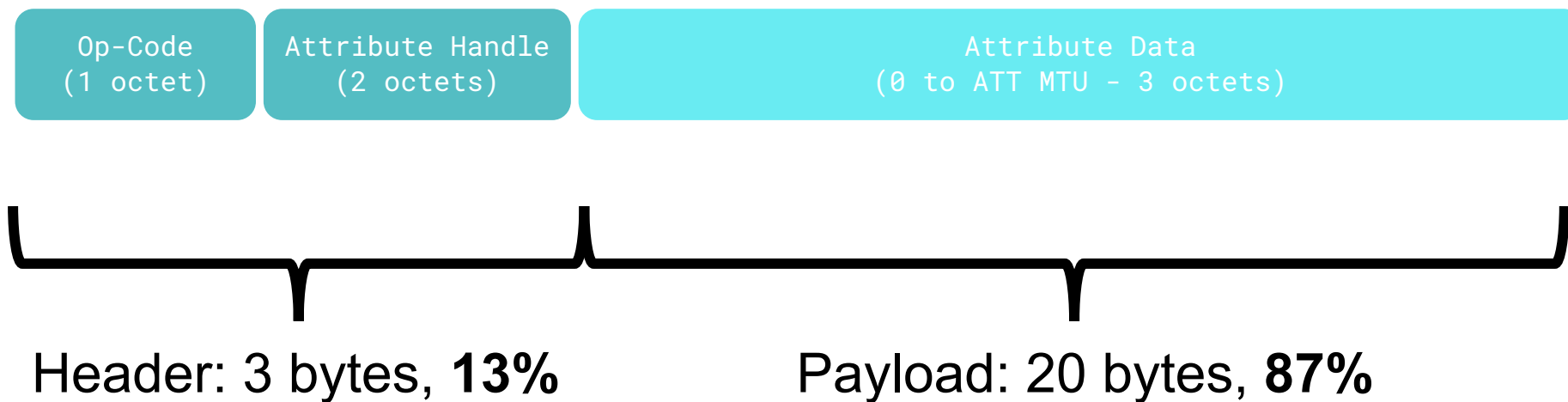


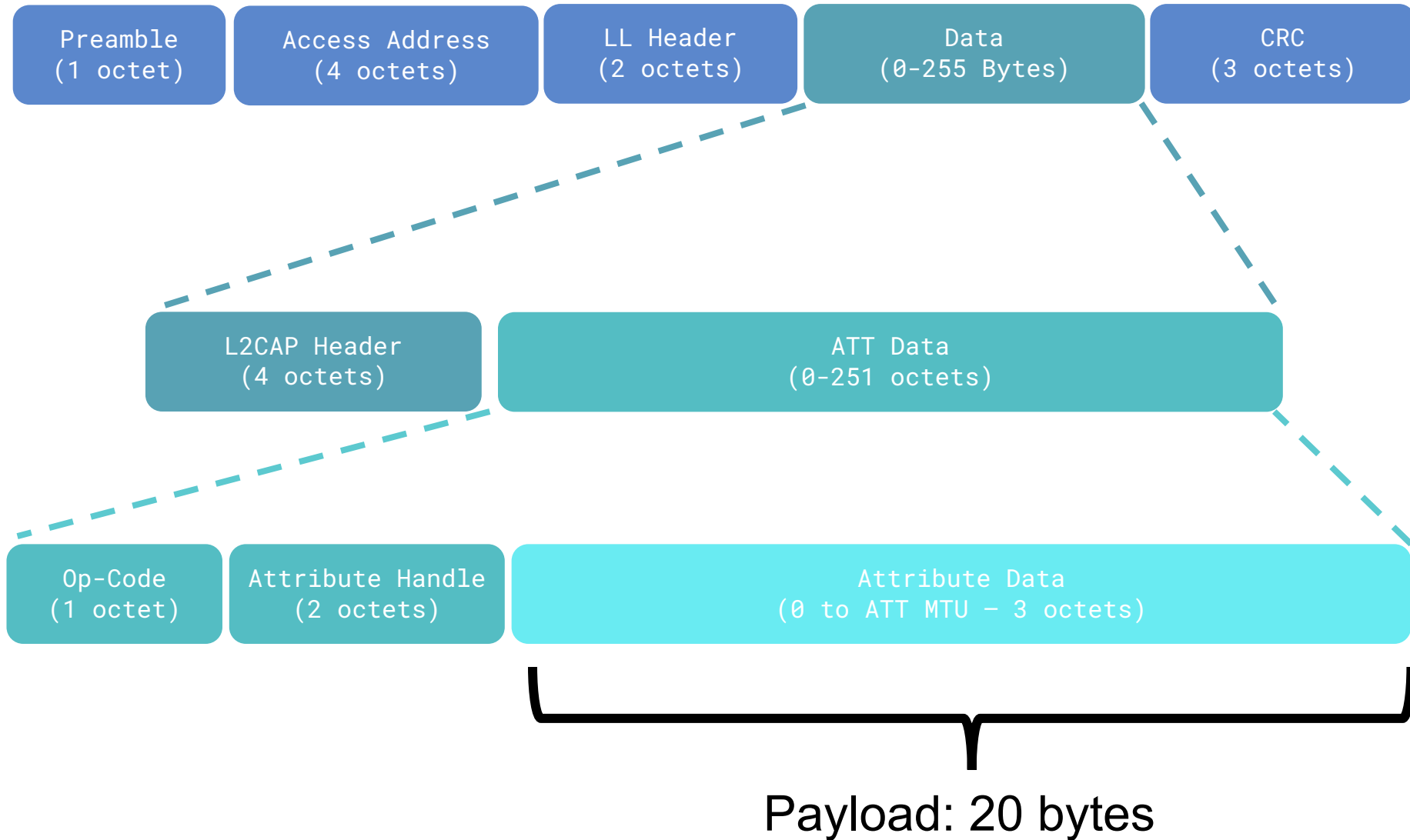
Содержание:

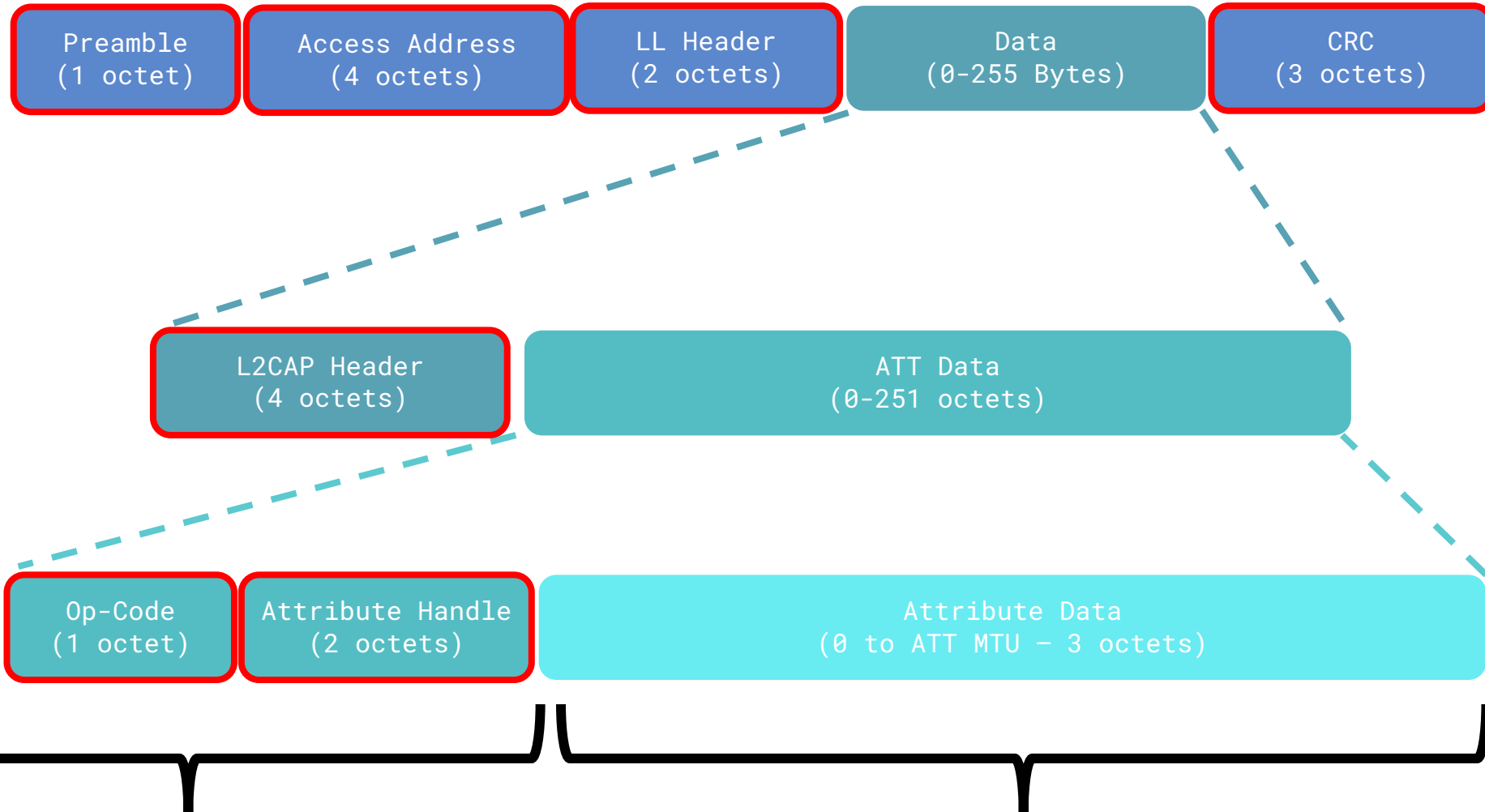
Что такое BLE	9
Как устроено BLE	14
Логический слой BLE	27
Физический слой BLE	48



Устройство пакета BLE

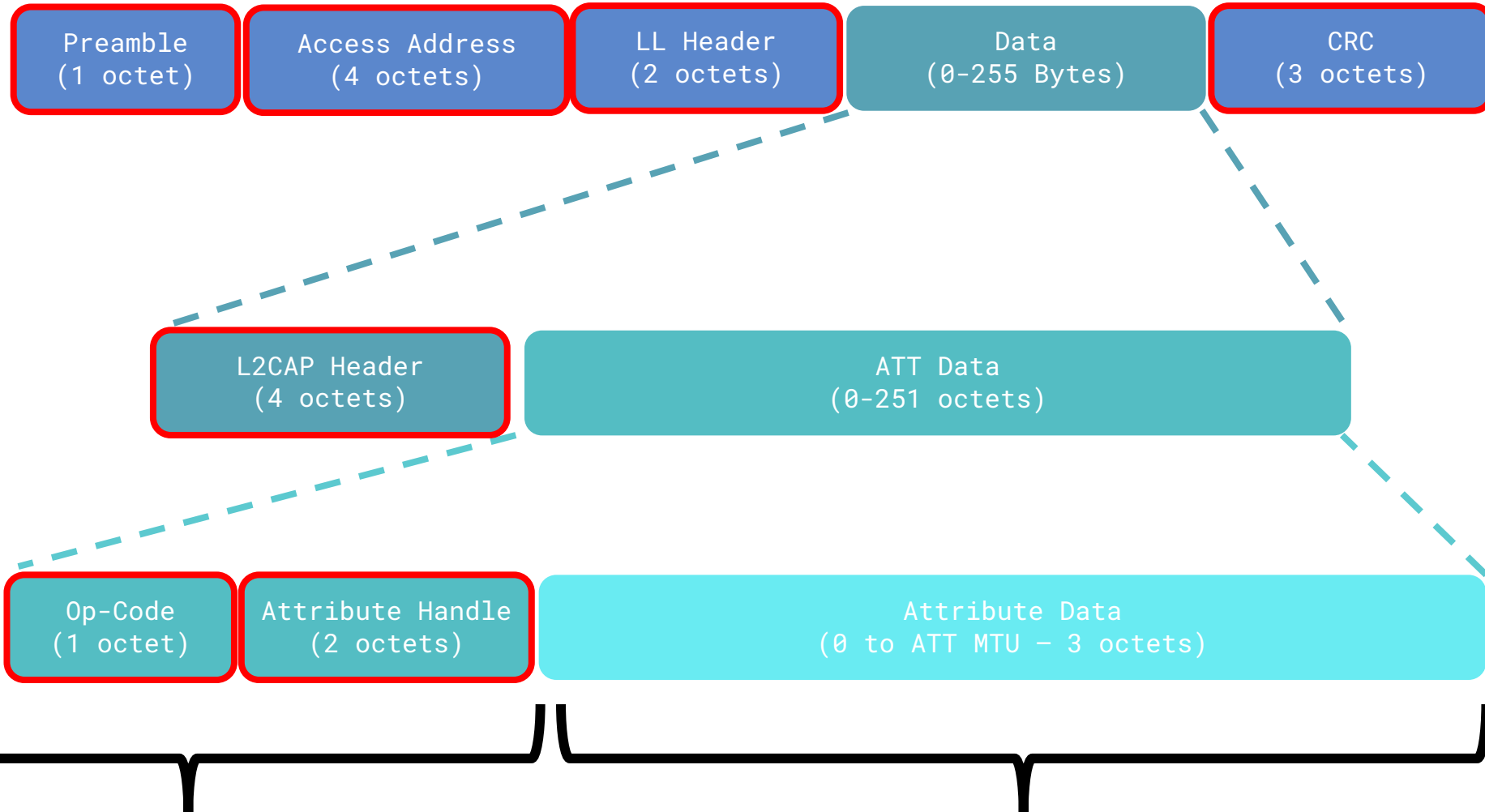






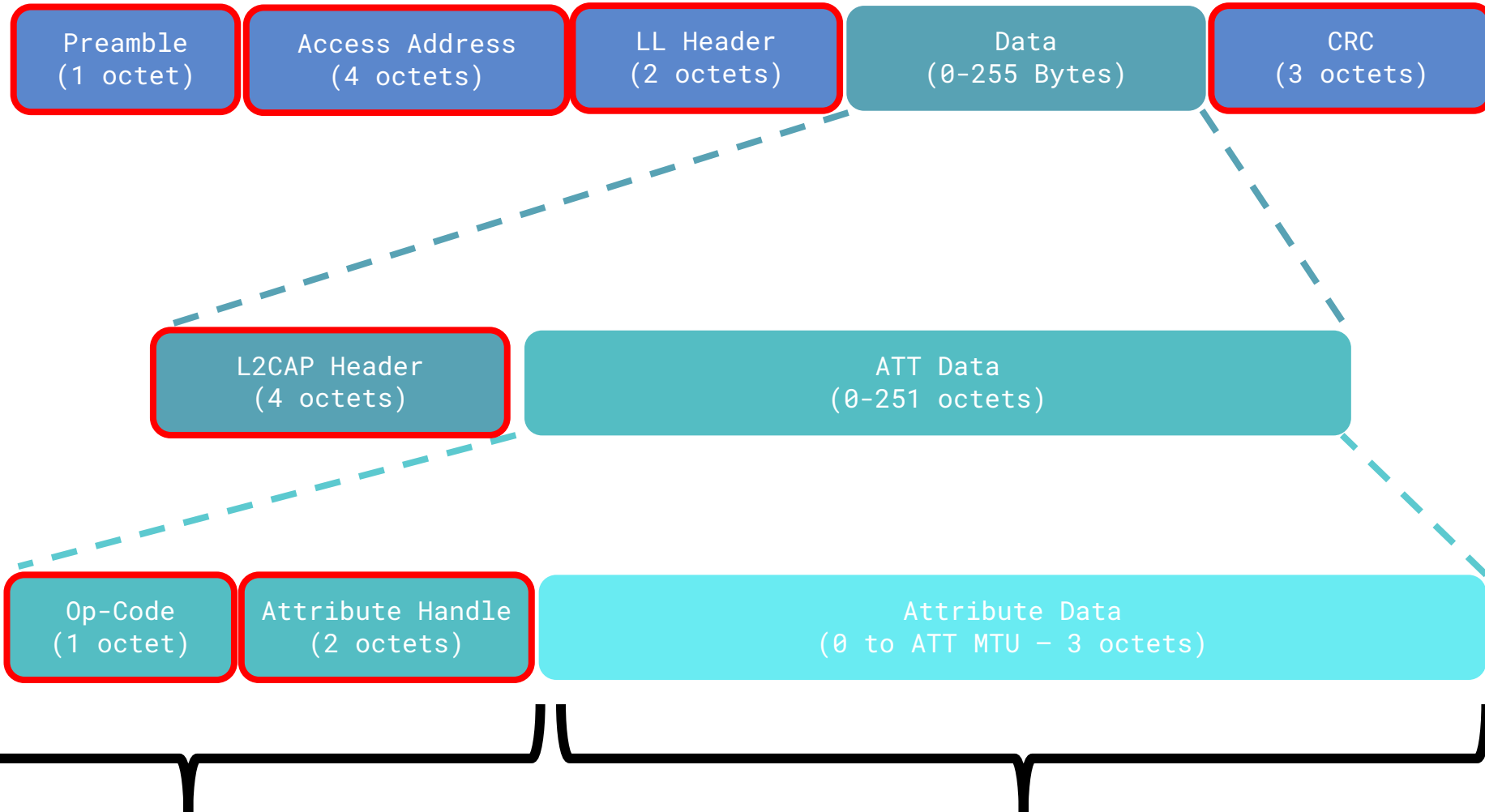
PHY header: 17 bytes, **3.2%**

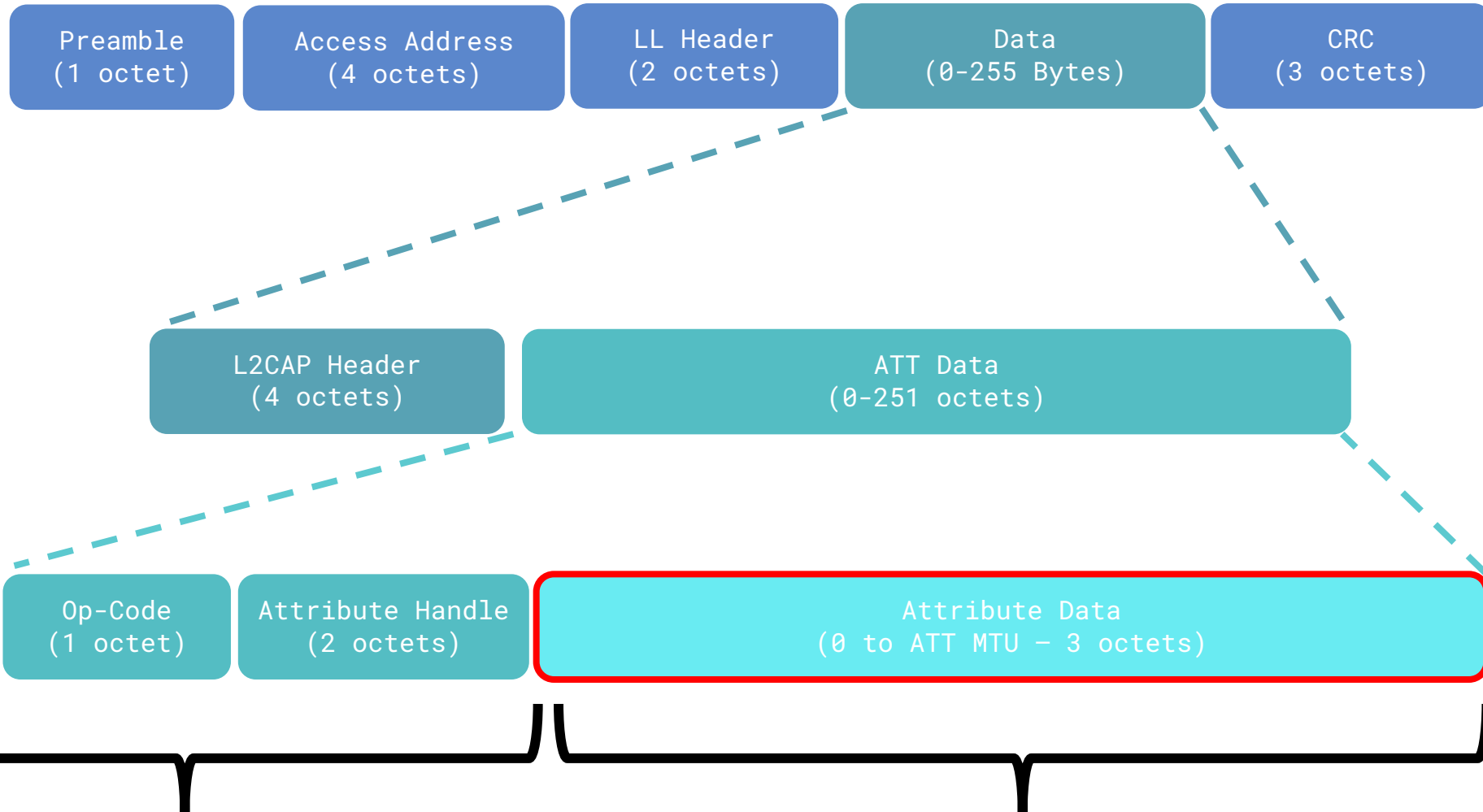
Payload: 20 bytes, **54%**



PHY header: 17 bytes, 3.2%

Payload: 20 bytes, 54%





PHY header: 17 bytes, **3.2%**

Payload: 509 bytes, **96.8%**



```
/* USER CODE END APP_BLE_Init_1 */
SHCI_C2_Ble_Init_Cmd_Packet_t ble_init_cmd_packet =
{
    {{0,0,0}},                                /**< Header unused */
    {pBleBufferAddress = 0,                    /** pBleBufferAddress not used */
      BleBufferSize = 0,                       /** BleBufferSize not used */
      NumAttrRecord = CFG_BLE_NUM_GATT_ATTRIBUTES,
      NumAttrServ = CFG_BLE_NUM_GATT_SERVICES,
      AttrValueArrSize = CFG_BLE_ATT_VALUE_ARRAY_SIZE,
      NumOfLinks = CFG_BLE_NUM_LINK,
      ExtendedPacketLengthEnable = CFG_BLE_DATA_LENGTH_EXTENSION,
      PrWriteListSize = CFG_BLE_PREPARE_WRITE_LIST_SIZE,
      MblockCount = CFG_BLE_MBLOCK_COUNT,
      AttMtu = MOBIUS_MTU,
      SlaveSca = CFG_BLE_SLAVE_SCA,
      MasterSca = CFG_BLE_MASTER_SCA,
      LsSource = CFG_BLE_LS_SOURCE,
      MaxConnEventLength = CFG_BLE_MAX_CONN_EVENT_LENGTH,
      HsStartupTime = CFG_BLE_HSE_STARTUP_TIME,
      ViterbiEnable = CFG_BLE_VITERBI_MODE,
      Options = CFG_BLE_OPTIONS,
      HwVersion = 0,
      max_coc_initiator_nbr = CFG_BLE_MAX_COC_INITIATOR_NBR,
      min_tx_power = CFG_BLE_MIN_TX_POWER,
      max_tx_power = CFG_BLE_MAX_TX_POWER,
      rx_model_config = CFG_BLE_RX_MODEL_CONFIG,
      CFG_BLE_MAX_ADV_SET_NBR,
      CFG_BLE_MAX_ADV_DATA_LEN,
      CFG_BLE_TX_PATH_COMPENS,
      CFG_BLE_RX_PATH_COMPENS,
      CFG_BLE_CORE_VERSION,
      CFG_BLE_OPTIONS_EXT
    }
};
```



```
/* USER CODE END APP_BLE_Init_1 */
SHCI_C2_Ble_Init_Cmd_Packet_t ble_init_cmd_packet =
{
    {{0,0,0}},                                /**< Header unused */
    {.pBleBufferAddress = 0,                    /** pBleBufferAddress not used */
      .BleBufferSize = 0,                      /** BleBufferSize not used */
      .NumAttrRecord = CFG_BLE_NUM_GATT_ATTRIBUTES,
      .NumAttrServ = CFG_BLE_NUM_GATT_SERVICES,
      .AttrValueArrSize = CFG_BLE_ATT_VALUE_ARRAY_SIZE,
      .NumOfLinks = CFG_BLE_NUM_LINK,
      .ExtendedPacketLengthEnable = CFG_BLE_DATA_LENGTH_EXTENSION,
      .PrWriteListSize = CFG_BLE_PREPARE_WRITE_LIST_SIZE,
      .MblockCount = CFG_BLE_MBLOCK_COUNT,
      .AttMtu = MOBIUS_MTU,
      .SlaveSca = CFG_BLE_SLAVE_SCA,
      .MasterSca = CFG_BLE_MASTER_SCA,
      .LsSource = CFG_BLE_LS_SOURCE,
      .MaxConnEventLength = CFG_BLE_MAX_CONN_EVENT_LENGTH,
      .HsStartupTime = CFG_BLE_HSE_STARTUP_TIME,
      .ViterbiEnable = CFG_BLE_VITERBI_MODE,
      .Options = CFG_BLE_OPTIONS,
      .HwVersion = 0,
      .max_coc_initiator_nbr = CFG_BLE_MAX_COC_INITIATOR_NBR,
      .min_tx_power = CFG_BLE_MIN_TX_POWER,
      .max_tx_power = CFG_BLE_MAX_TX_POWER,
      .rx_model_config = CFG_BLE_RX_MODEL_CONFIG,
      CFG_BLE_MAX_ADV_SET_NBR,
      CFG_BLE_MAX_ADV_DATA_LEN,
      CFG_BLE_TX_PATH_COMPENS,
      CFG_BLE_RX_PATH_COMPENS,
      CFG_BLE_CORE_VERSION,
      CFG_BLE_OPTIONS_EXT
    }
};
```



Android

requestMtu

Added in API level 21

```
public boolean requestMtu (int mtu)
```



Request an MTU size used for a given connection.

When performing a write request operation (write without response), the data sent is truncated to the MTU size. This function may be used to request a larger MTU size to be able to send more data at once.



iOS

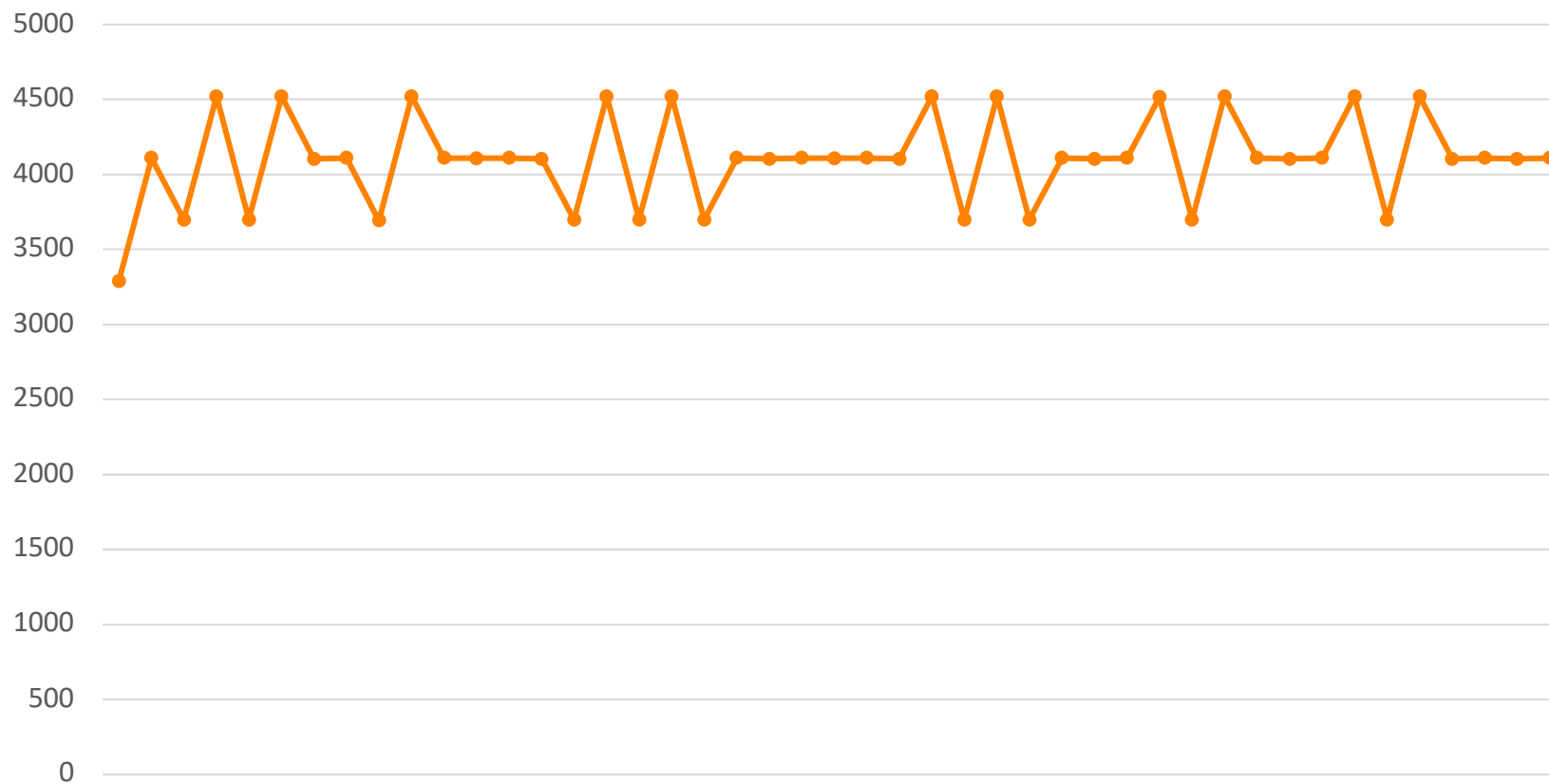
< iOS 10: 158 bytes

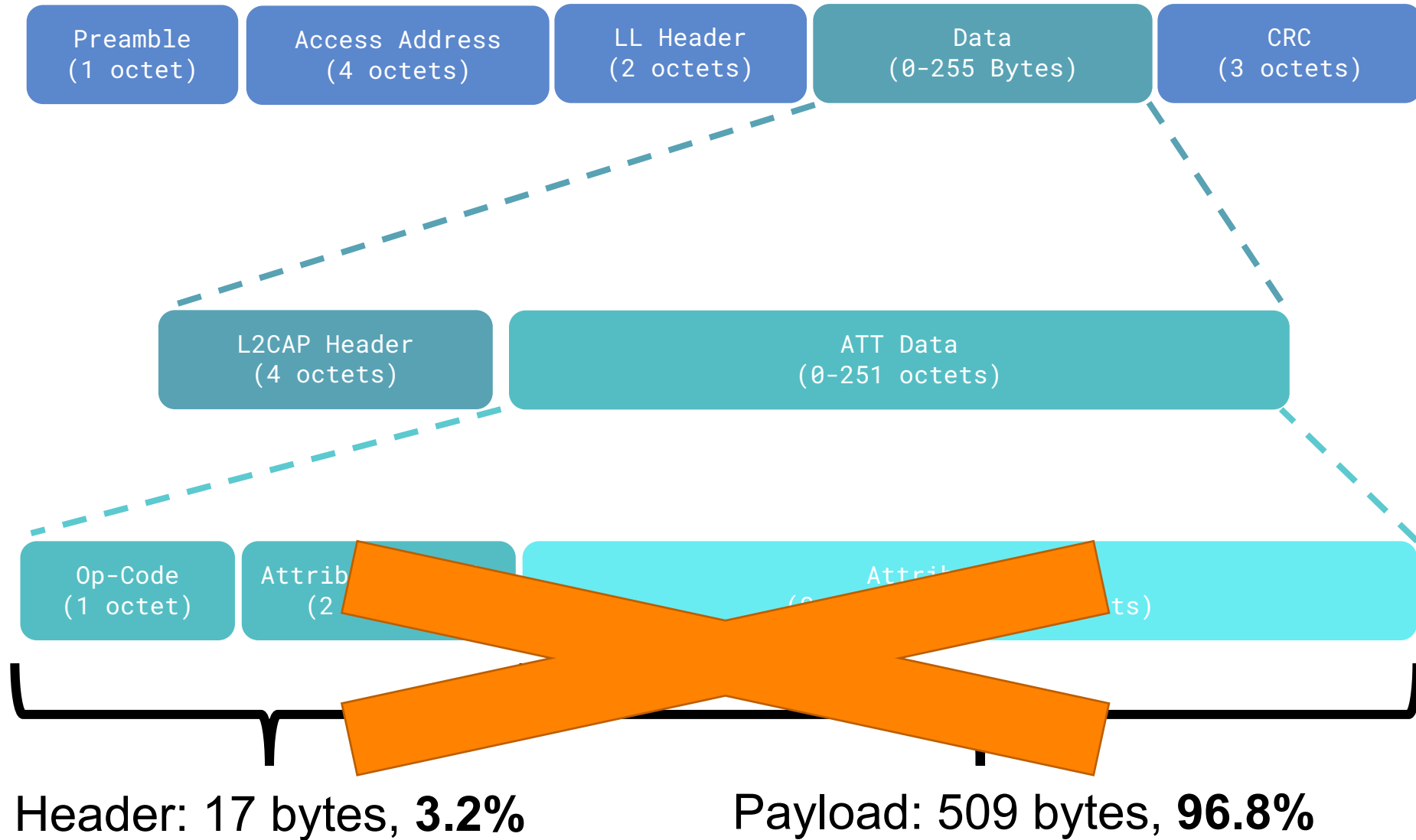
> iOS 10: 185+ bytes

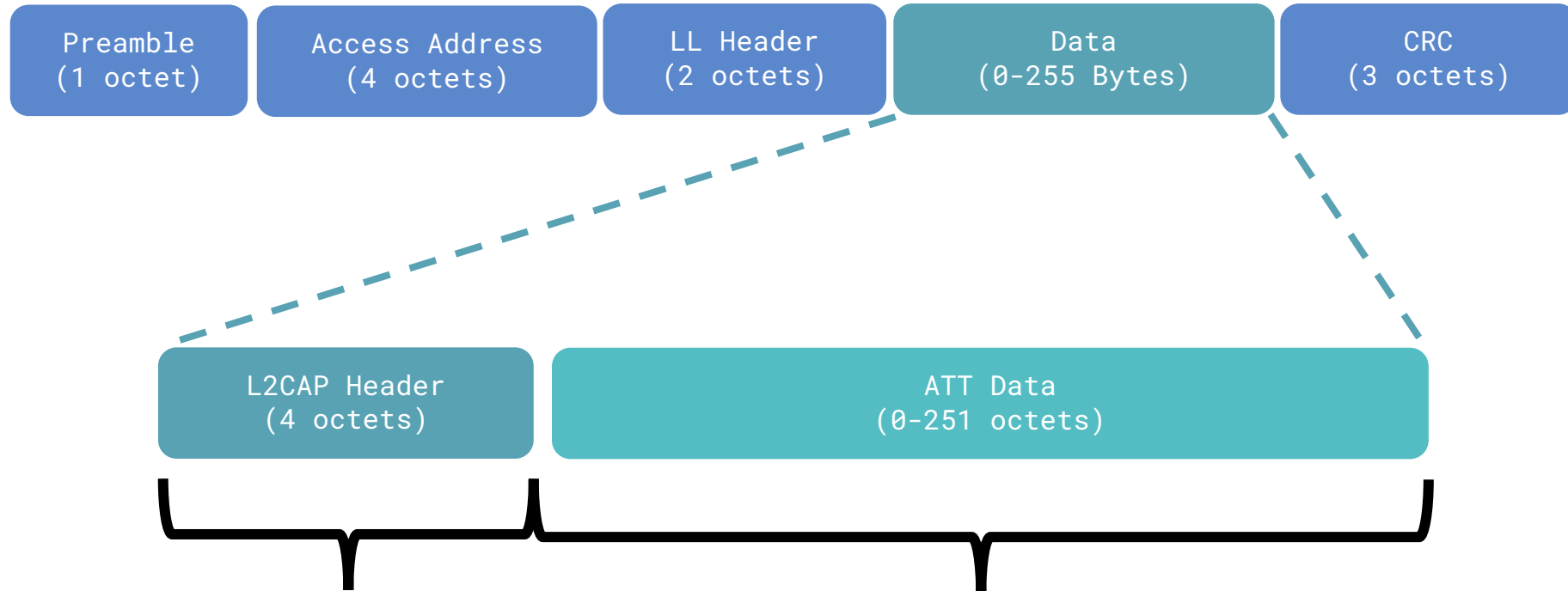
```
var maximumWriteValueLength: Int {  
    min(freeSpace,  
        peripheral.maximumWriteValueLength(for: .withoutResponse))  
}
```



Замеры с увеличенным MTU







Header: 17 bytes
3.23% -> 3.21%

Payload: 509 bytes, **96.8%**



createL2capChannel

Added in API level 29

```
public BluetoothSocket createL2capChannel (int psm)
```



Class

CBL2CAPChannel

A live L2CAP connection to a remote device.

iOS 11.0+

iPadOS 11.0+

macOS 10.13+

Mac Catalyst 13.1+

tvOS 11.0+

watchOS 4.0+

FLIPPER



createL2capChannel

Added in API level 29

```
public BluetoothSocket createL2capChannel (int psm)
```



Class

CBL2CAPChannel

A live L2CAP connection to a remote device.

iOS 11.0+

iPadOS 11.0+

macOS 10.13+

Mac Catalyst 13.1+

tvOS 11.0+

watchOS 4.0+



Write without Response

```
* Add write characteristic
*/
hciCmdResult = aci_gatt_add_char(HRS_Context.HeartRateSvcHdle,
                                UUID_TYPE_128,
                                (const Char_UUID_t*)char_rx_uuid,
                                MOBIUS_RX_MAXDATA_LEN,
                                CHAR_PROP_WRITE_WITHOUT_RESP | CHAR_PROP_WRITE,
                                ATTR_PERMISSION_NONE,
                                GATT_NOTIFY_ATTRIBUTE_WRITE, /* gattEvtMask */
                                10, /* encryKeySize */
                                CHAR_VALUE_LEN_VARIABLE, /* isVariable */
                                &(HRS_Context.SerialRxCharHdle));

if (hciCmdResult == BLE_STATUS_SUCCESS) {
    BLE_DBG_DIS_MSG ("Serial RX Characteristics Added Successfully %04X \n",
                    HRS_Context.SerialRxCharHdle);
}
else
{
    BLE_DBG_DIS_MSG ("FAILED to add Serial RX Characteristics, Error: %02X !!\n",
                    hciCmdResult);
}
```



Write without Response

```
/* Add write characteristic
*/
hciCmdResult = aci_gatt_add_char(HRS_Context.HeartRateSvcHdle,
                                UUID_TYPE_128,
                                (const Char_UUID_t*)char_rx_uuid,
                                MOBIUS_RX_MAXDATA_LEN,
                                CHAR_PROP_WRITE_WITHOUT_RESP | CHAR_PROP_WRITE,
                                ATTR_PERMISSION_NONE,
                                GATT_NOTIFY_ATTRIBUTE_WRITE, /* gattEvtMask */
                                10, /* encryKeySize */
                                CHAR_VALUE_LEN_VARIABLE, /* isVariable */
                                &(HRS_Context.SerialRxCharHdle));

if (hciCmdResult == BLE_STATUS_SUCCESS) {
    BLE_DBG_DIS_MSG ("Serial RX Characteristics Added Successfully %04X \n",
                    HRS_Context.SerialRxCharHdle);
}
else
{
    BLE_DBG_DIS_MSG ("FAILED to add Serial RX Characteristics, Error: %02X !!\n",
                    hciCmdResult);
}
```



Write without Response

Android:

```
log(Log.DEBUG, () -> characteristic.setValue(  
characteristic.setValue(data);  
log(Log.DEBUG, () -> "characteristic.setWriteType(  
characteristic.setWriteType(writeType);  
log(Log.DEBUG, () -> "gatt.writeCharacteristic("  
return gatt.writeCharacteristic(characteristic);
```

iOS:

```
log(Log.DEBUG, () -> "writeValue(  
return  
}  
peripheral.writeValue(data, for: serialWrite, type: .withResponse)  
freeSpace -= data.count  
}
```



Write without Response

Android:

```
log(Log.DEBUG, () -> characteristic.setValue(  
characteristic.setValue(data);  
log(Log.DEBUG, () -> "characteristic.setWriteType(  
characteristic.setWriteType(writeType);  
log(Log.DEBUG, () -> "gatt.writeCharacteristic("  
return gatt.writeCharacteristic(characteristic);
```

iOS:

```
log(Log.DEBUG, () -> "writeValue", data);  
return  
}  
peripheral.writeValue(data, for: serialWrite, type: .withResponse)  
freeSpace -= data.count  
}
```



Замеры WW/R



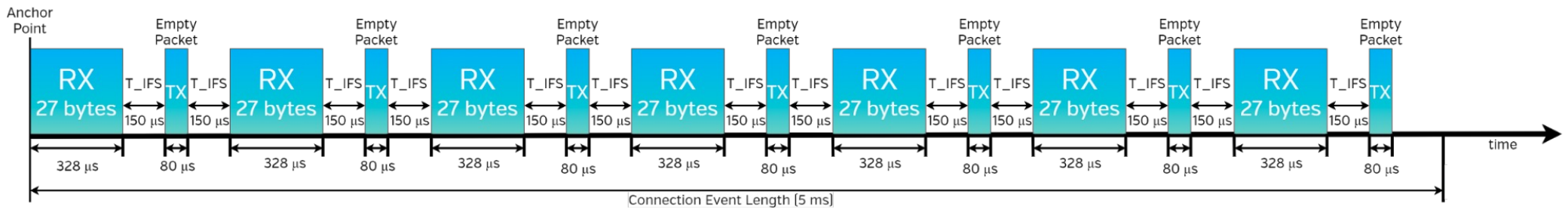


Содержание:

Что такое BLE	9
Как устроено BLE	14
Логический слой BLE	27
Физический слой BLE	48



Connection Interval





STM32

```
if(aci_l2cap_connection_parameter_update_req(  
    gap->service.connection_handle,  
    params->conn_int_min,  
    params->conn_int_max,  
    gap->connection_params.slave_latency,  
    gap->connection_params.supervisor_timeout)) {  
    FURI_LOG_E(TAG, "Failed to request connection parameters update");  
}
```



Android

requestConnectionPriority

Added in API level 21

```
public boolean requestConnectionPriority (int connectionPriority)
```



Request a connection parameter update.

This function will send a connection parameter update request to the remote device.

For apps targeting `Build.VERSION_CODES#S` or or higher, this requires the

`Manifest.permission#BLUETOOTH_CONNECT` permission which can be gained with

`Activity.requestPermissions(String[], int)`.

Requires `Manifest.permission.BLUETOOTH_CONNECT`



Interval: 30 - 50 ms,

Interval: 11.25 - 15 ms

Interval: 100 - 125 ms,

Connection parameter update - Use the connection parameters recommended by the Bluetooth SIG. This is the default value if no connection parameter update is requested.

Interval: 30 - 50 ms, latency: 0, supervision timeout: 5 sec (Android 8+) or 20 sec (before).

See Also: [commit 673c5903c4a920510c371af26e5870857a584ead](#) ↗

7 usages

```
public static final int CONNECTION_PRIORITY_BALANCED = 0;
```

Connection parameter update - Request a high priority, low latency connection. An application should only request high priority connection parameters to transfer large amounts of data over LE quickly. Once the transfer is complete, the application should request `CONNECTION_PRIORITY_BALANCED` connection parameters to reduce energy use.

Interval: 11.25 - 15 ms (Android 6+) or 7.5 - 10 ms (Android 4.3 - 5.1), latency: 0, supervision timeout: 5 sec (Android 8+) or 20 sec (before).

See Also: [commit 4bc7c7e877c9d18f2781229c553b6144f9fd7236](#) ↗,
[commit 673c5903c4a920510c371af26e5870857a584ead](#) ↗

6 usages

```
public static final int CONNECTION_PRIORITY_HIGH = 1;
```

Connection parameter update - Request low power, reduced data rate connection parameters.

Interval: 100 - 125 ms, latency: 2, supervision timeout: 5 sec (Android 8+) or 20 sec (before).

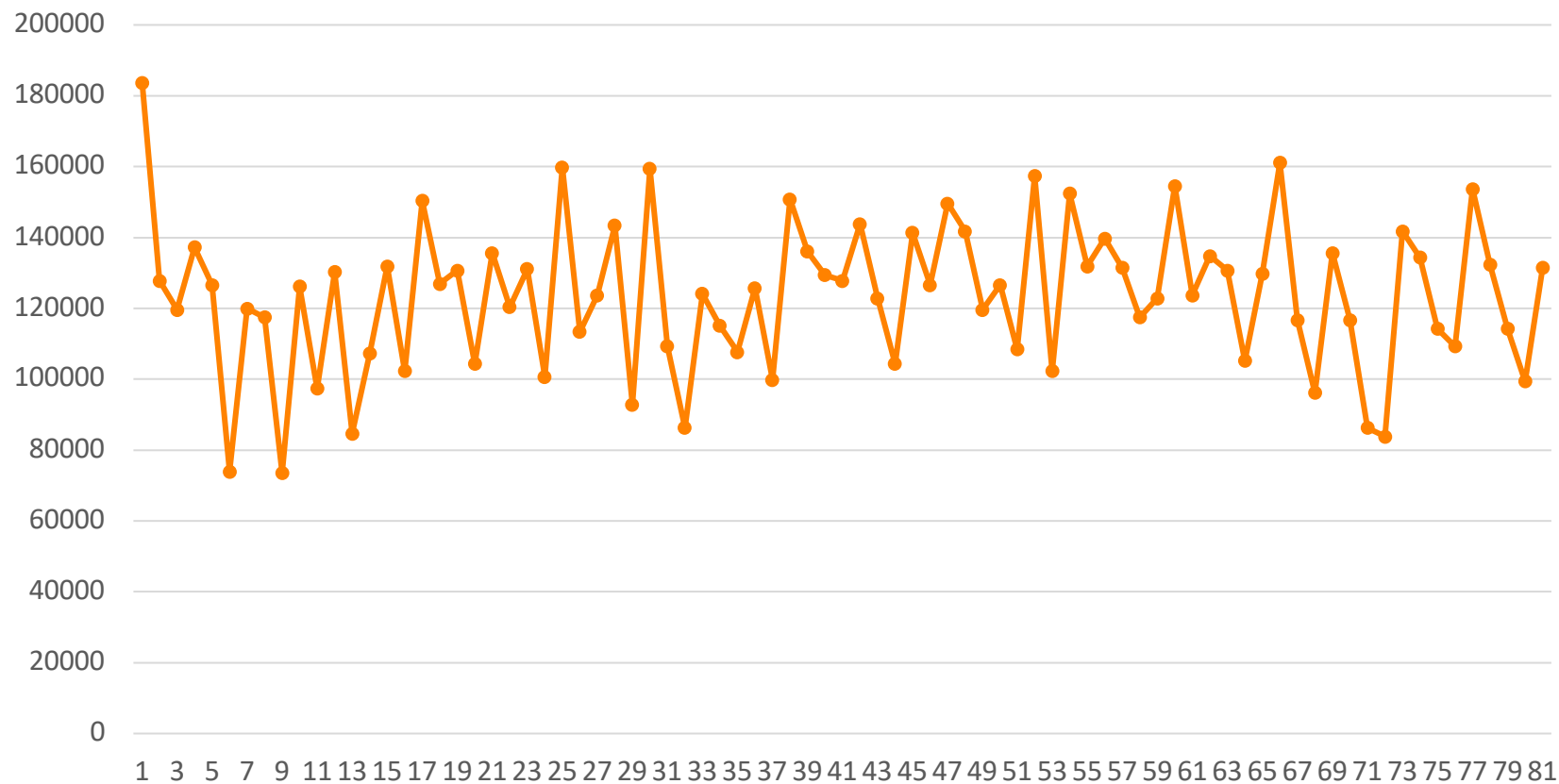
See Also: [commit 673c5903c4a920510c371af26e5870857a584ead](#) ↗

5 usages

```
public static final int CONNECTION_PRIORITY_LOW_POWER = 2;
```

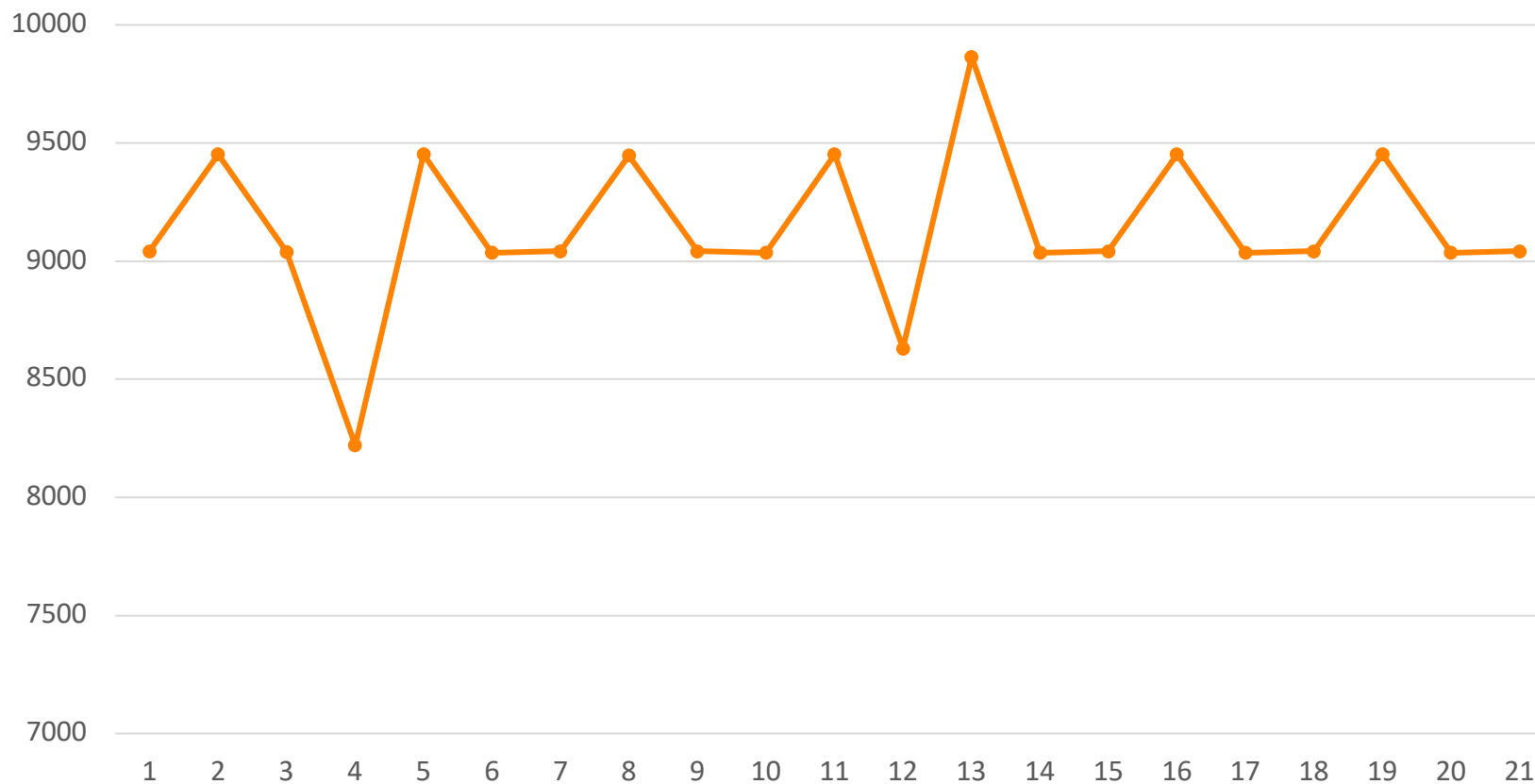


Замеры WW/R, ConnectionInterval



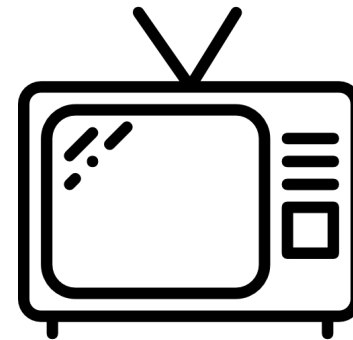
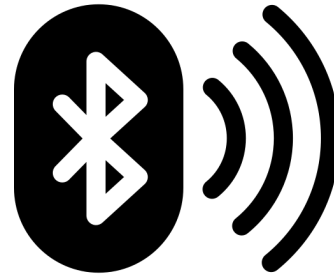
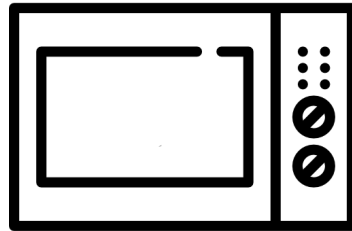


Замеры WWR, ConnectionInterval



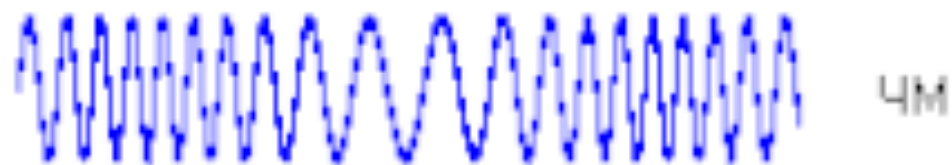
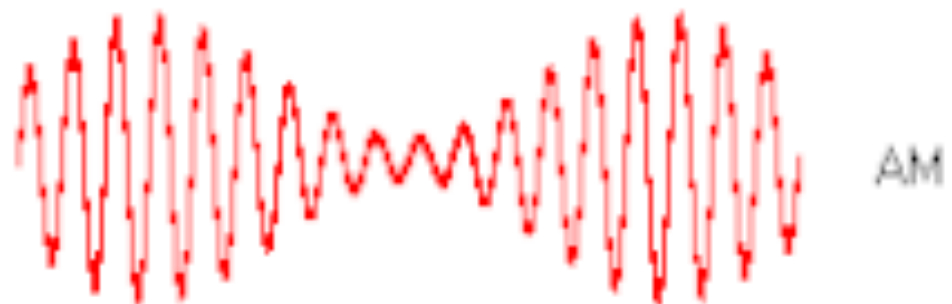


2.4 Ghz





Частотная модуляция





1M PHY

-185 KHz

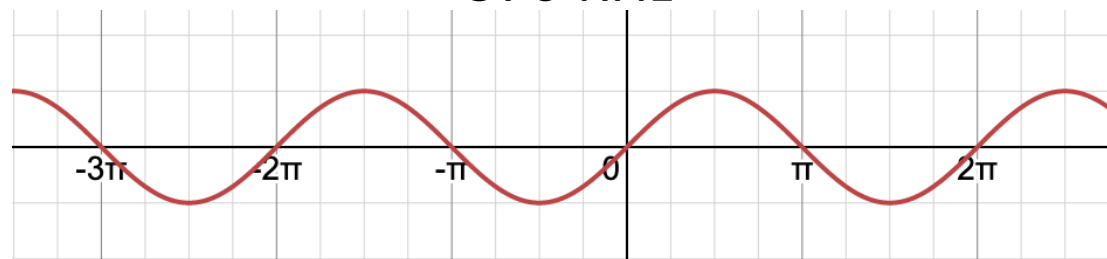


+185 KHz



2M PHY

-370 KHz



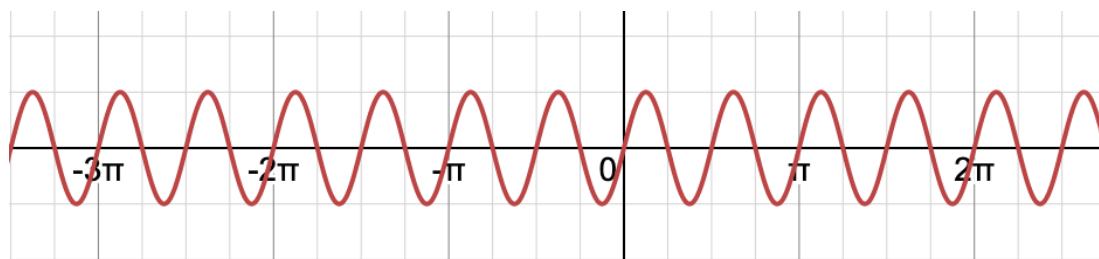
-185 KHz



+185 KHz



+370 KHz





STM32

```
/**
 * Initialize Default PHY
 */
ret = hci_le_set_default_phy(ALL_PHYS_PREFERENCE,MOBIUS_PHY_TX,MOBIUS_PHY_RX);
if (ret != BLE_STATUS_SUCCESS)
{
    APP_DBG_MSG("  Fail   : hci_le_set_default_phy command, result: 0x%x \n", ret);
}
else
{
    APP_DBG_MSG("  Success: hci_le_set_default_phy command\n");
}
```



Android

setPreferredPhy

Added in API level 26

```
public void setPreferredPhy (int txPhy,  
                             int rxPhy,  
                             int phyOptions)
```



Set the preferred connection PHY for this app. Please note that this is just a recommendation, whether the PHY change will happen depends on other applications preferences, local and remote controller capabilities. Controller can override these settings.

`BluetoothGattCallback#onPhyUpdate` will be triggered as a result of this call, even if no PHY change happens. It is also triggered when remote device updates the PHY.

For apps targeting `Build.VERSION_CODES#S` or or higher, this requires the

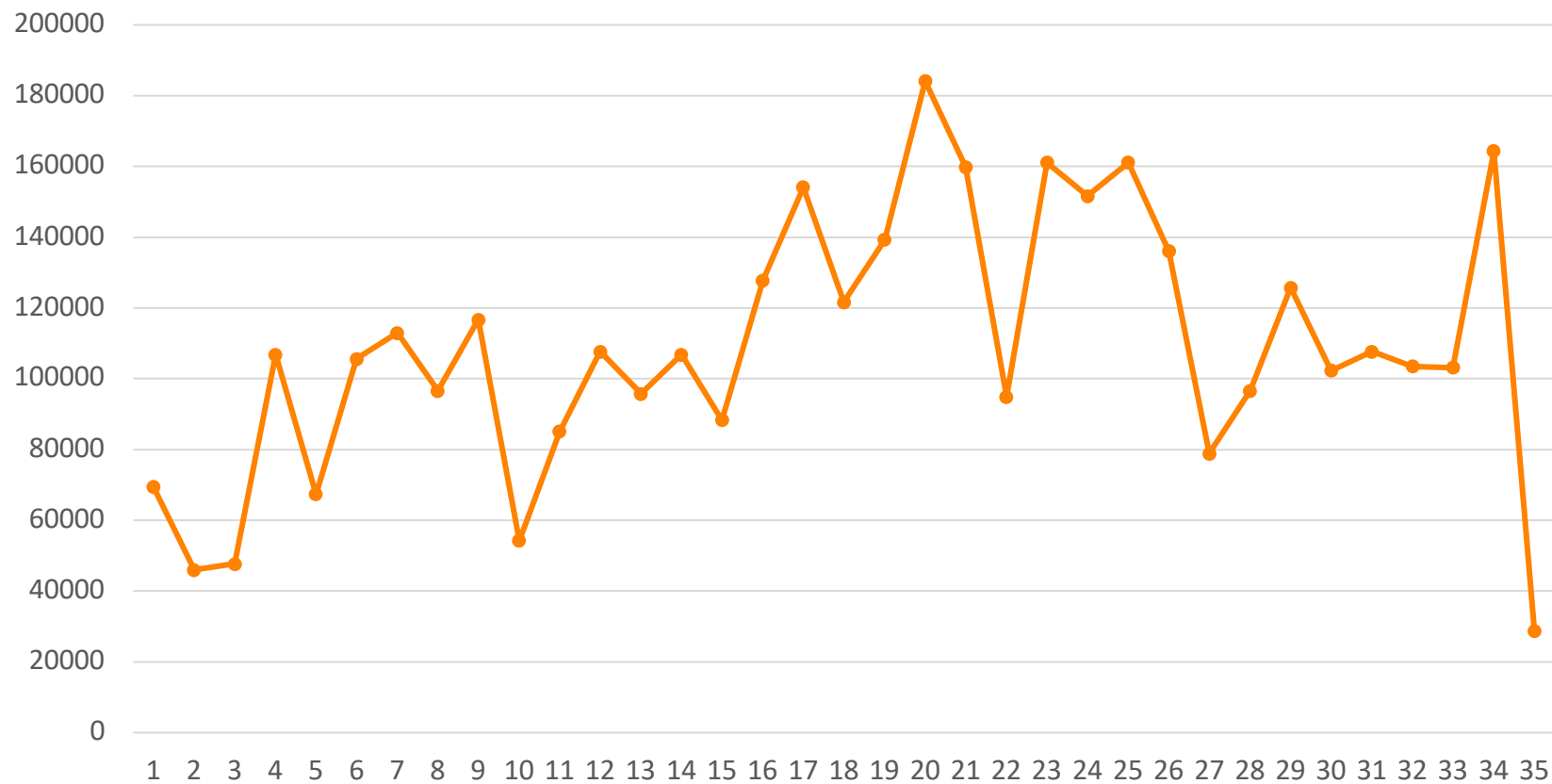
`Manifest.permission#BLUETOOTH_CONNECT` permission which can be gained with

`Activity.requestPermissions(String[], int)`.

Requires `Manifest.permission.BLUETOOTH_CONNECT`

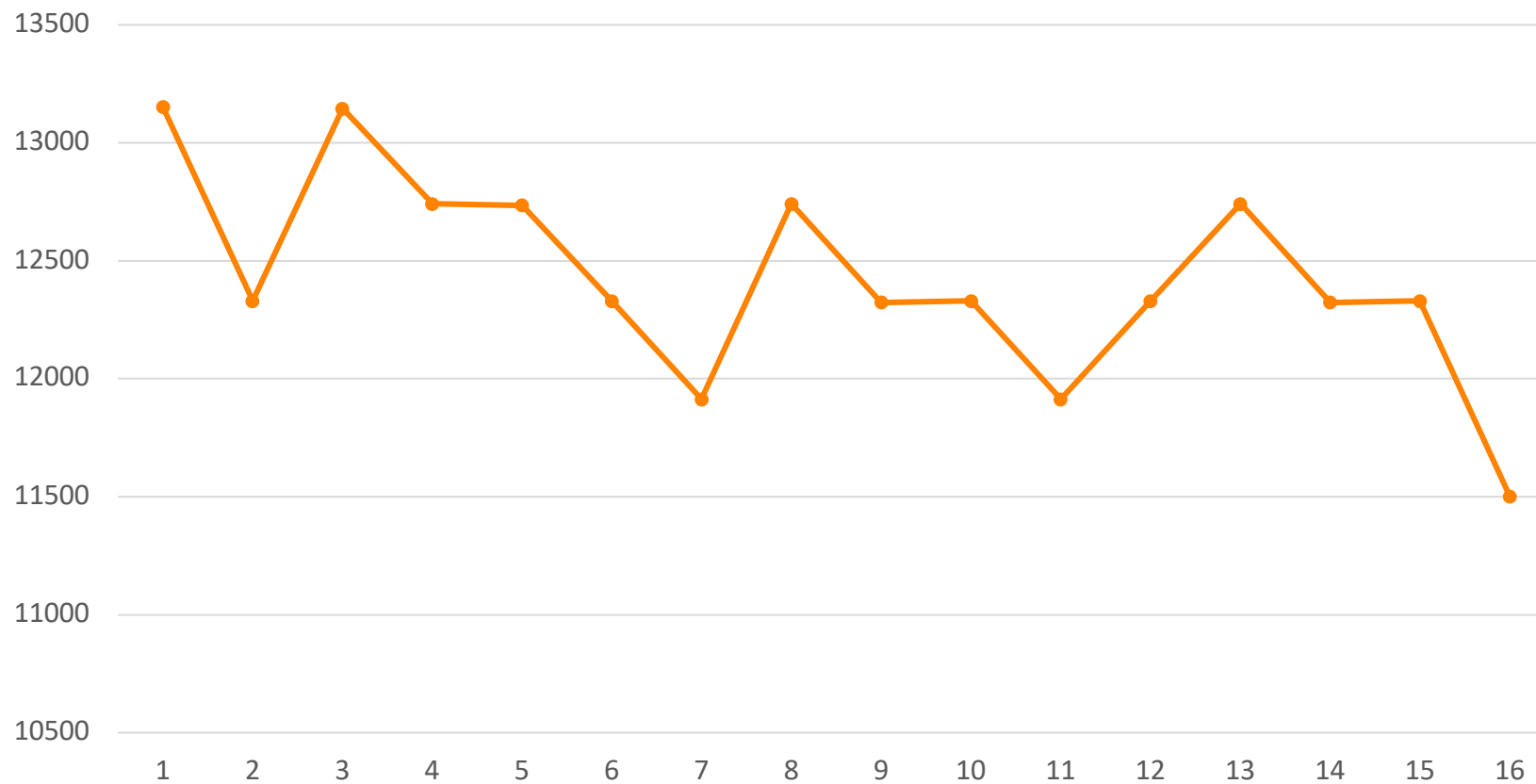


2M PHY, WW/R





2M PHY, WWR



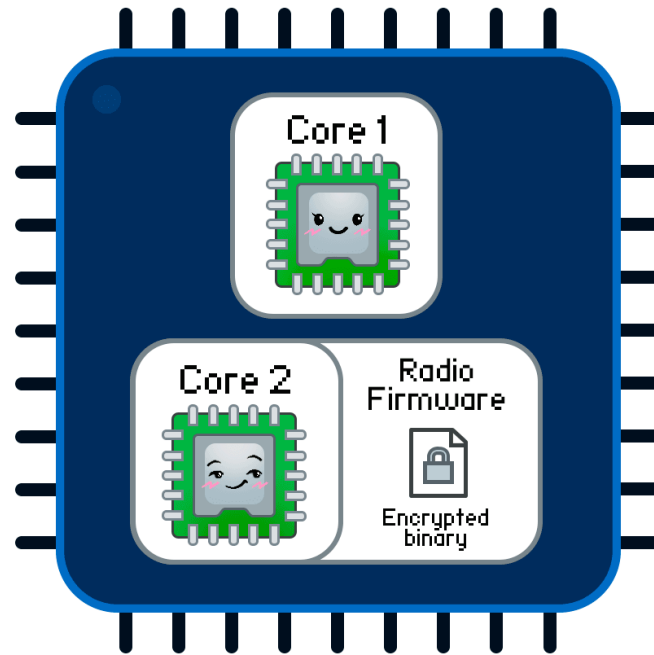


Итоговая таблица

Ускорение	WWR	WW/R
Изначально	~200 b/s	
Увеличение MTU	~4кб/с	~140кб/с
Уменьшение Interval	~9кб/с	~140кб/с
2M PHY	~12кб/с	~140кб/с



STM32WB55





Итоговая таблица

Ускорение	WWR	WW/R
Изначально	~200 b/s	
Увеличение MTU	~4кб/с	~140кб/с
Уменьшение Interval	~9кб/с	~140кб/с
2M PHY	~12кб/с	~140кб/с