

BLE – Základní Informace / Komunikace

Petr Novák, novakpe@fel.cvut.cz (23.10.2020)

Úvod

(!!! V tomto textu jsou uvedeny pouze základní informace. !!!)

Základní pojmy

- Bluetooth Classic – Původní Bluetooth V2.x a V3.x.
- Bluetooth LE (Bluetooth Low Energy) – Standard Bluetooth pro přenos krátkých bloků dat na menší vzdálenosti s minimální energetickou náročností, V4.x a V5.x. BT Classic a BT LE jsou zcela nekompatibilní protokoly, ale některá zařízení / obvody umožňují současnou komunikaci v obou standardech, nazývají se tzv. „duální“.
- GAP (Generic Access Profile) – Definuje „Advertising pakety“ (informování / hledání) a způsob připojování BLE zařízení.
- GATT (Generic Attribute Profile) – Definuje způsob komunikace / přenosu dat mezi BLE zařízeními.
- ATT (Attribute Protocol) – Definuje „Services“, „Characteristic“ a „Data“ pro přenos mezi BLE zařízeními.
- BLE Device (zařízení) pracuje v jednom ze dvou režimů
 - o GAP roles (režim při hledání / připojování)
 - o Central (Observer) / Peripheral (Broadcaster) – „Centrál“ inicializuje / vykonává spojení na „Peripheral“.
 - Peripheral (master) – Poskytuje informace a připojení pro možnost čtení / zaslání dat (Broadcaster – poskytuje data vysíláním bez možnosti připojení, tedy pouze v ADVS paketu).
 - Centrál (slave) – Hledá zařízení pro připojení s možností čtení / zápisu data (Observer – sbírá data pomocí odposlechu vysílání bez připojení, tedy pouze čtením ADVS paketu).
 - Příklad: „Centrál“ je tablet a „Peripheral“ je BLE reproduktor.
 - o GATT rosel (režim při spojení / komunikace)
 - o Server / Client
 - Server (služba) – Poskytuje službu čtení / ukládání dat připojenému zařízení (clientu).
 - Client (žadatel) – Využívá služby pro čtení / ukládání dat na serveru.
- MAC address – Unikátní adresa BLE zařízení. Tato adresa však v naprosté většině případů není dána v HW, ale velmi často si ji BLE zařízení generuje náhodně a to i několikrát při své činnosti (například při přepínání režimů své činnosti).
- UUID (Universally unique identifier) – Pro některé BLE služby jsou tato čísla dána BLE standardem, pro vlastní služby je nutno si je (náhodně) vygenerovat.
- LE/DLE (Data Length Extension) – Rozšíření BLE standardu pro přenos většího množství dat v jednom datovém paketu. Jistá rozšíření obsahuje i velikost „Advertising paketu“.

Princip činnosti:

- BLE zařízení (server) vysílá tzv. „Advertising pakety“, aby se o něm věděli další BLE zařízení. Tyto „Advertising pakety“ obsahují jméno zařízení, jeho MAC adresu a další informace o vysílaném zařízení (možno i služby a data).

- Jiná BLE zařízení (clienti) tyto „Advertising pakety“ zachytávají, aby měli informace o BLE zařízeních v dosahu.
- Zařízení BLE Client se připojí k zařízení BLE Server podle MAC adresy cílového BLE zařízení.
- BLE Client zjistí / prohlédne všechny „Services“ a „Characteristics“ na cílovém zařízení, aby věděl jak s cílovým BLE zařízením komunikovat (pokud to neví již předem).
- Komunikace je uskutečněna pomocí „Services“ a „Characteristics“. V podstatě poskytovaných služeb ze strany serveru. BLE zařízení, zde server, může obsahovat i několik „Service“ a každá z nich může obsahovat i několik „Characteristic“. „Services“ tedy v podstatě sdružují „Characteristics“ do nějakých logických celků / skupin.
- Vlastní komunikace probíhá zápisem a čtením hodnot do „Characteristic(s)“ v jednotlivých „Service(s)“. V základní specifikaci BLE lze zapisovat / číst bloky dat maximálně 20bytes. V rozšířené specifikaci (avšak podle skutečných možností zařízení) může být blok dat až 251 bytes.
- Jedno BLE zařízení, například server, zapíše do „Characteristic“ data a druhé BLE zařízení, například client, si je může kdykoli přečíst. Samozřejmě je možný i opačný směr.
- Aby nemusel client stále kontrolovat zda server (již) zapsal do „Characteristic“ nová data, tak BLE standard poskytuje funkci tzv. „Notify“. Pokud client povolí / registruje „Notify“ na nějakou „Characteristic“, tak při zápisu dat serverem do této „Characteristic“, se na straně příjmu / klienta generuje událost (client se tedy okamžitě dozví o zápisu nových dat).

„Advertising paket“

- Vysílá zařízení, aby se o něm vědělo. Obsahuje základní informace o něm (v jistém případě i data).

Advertisement Data (přenášena v Advs paketu)

- Několik sekcí, libovolný počet.
- Sekce začíná délkou (1byte). Délka = identifikátor sekce + počet bytes v sekci. Do délky se nepočítá byte udávající délku sekce.
- Následuje označení sekce (1Byte).
- Dále jsou data sekce, až do jejího konce.
- Celková délka Advs paketu (všech sekcí s daty) je maximálně 32 bytes (pro základní verzi, pro některé rozšířené verze je delší).

Typy sekcí (seznam):

- 0x01 – Flags / příznaky.
- 0x03 – Complete List of 16-bit Service Class UUIDs
- 0x08 – Shortened Local Name
- 0x09 – Complete Local Name
- 0xFF – Manufacturer Specific Data

Sekce – Connect Flags (typ 0x01)

- Délka sekce 0x02 (1Byte) Snad vždy.
- Označení sekce 0x01. (1Byte)
- Data sekce / příznaky (1Byte).
 - o Bit0 – LE Limited Discoverable Mode (viditelnost na omezený čas)
 - o Bit1 – LE General Discoverable Mode (viditelnost na neomezený čas)
 - o Bit2 - BR/EDR is supported (iBeacon is Dual Mode device)

- Bit3 – LE and BR/EDR Controller operates simultaneously
- Bit4 – LE and BR/EDR Host operates simultaneously

Sekce – Manufacturer Specific Data (typ 0xFF)

- Označení sekce „0xFF“
- Následuje „Company ID“, tzv. „SIG“ (2Byte). Pokud nemá výrobce přidělen „SIG“, mělo by být uvedeno „0xFFFF“. První hodnota v sekci je LOW byte a druhá hodnota je HIGH byte (SIG=0x1234, v sekci zapsáno ... 0x34 0x12 ...).
- Další hodnoty / bytes zcela vlastní.
- Příklad 0x05 0xFF 0x34 0x12 0x55 0x66. Délka sekce 5 bytes, sekce typu „Manufacturer Specific Data“, SIG = 0x1234, vlastní data 0x55 a 0x66.

„Services“ a „Characteristics“

- Slouží pro komunikaci / výměnu dat mezi BLE zařízeními a představují v podstatě služby.
- Jsou vytvořeny / umístěny na straně serveru. Zapisuje do nich jak server, tak i client. Stejně platí i o čtení.
- „Service“ i „Characteristic“ jsou identifikovány podle tzv. UUID čísla. Podle UUID rovněž client hledá na serveru, zda obsahuje správný komunikační kanál / službu. UUID čísla jsou dvou typů:
 - Private (16bits) – Stanovená BLE standardem (BatteryLevel, ...). Pro rychlé hledání komunikačních kanálů a snadnou možnost umístění přímo v ADVS paketu.
 - Public (128bits) – Vlastní vytvořená. Nutno generovat správným programem (existuje jich mnoho).
- „Service“ pouze sdružuje i několik „Characteristic(s)“ je identifikována právě UUID číslem (ničím více).
- Každá „Characteristic“ je identifikována těmito parametry:
 - UUID číslo (většinou nějak odvozeným v UUID „Service“ změnou některých číslic / pozic / bytes)
 - Délka / velikost bloku dat, který je přenášen (běžně 20 bytes, ale některá rozšíření dovolují až 251 bytes).
 - Parametry / příznaky (ne všechny jsou zde uvedeny) přenosu dat (z pohledu klienta):
 - 0x02 Read – Hodnota se zapisuje serverem a čte clientem
 - 0x04 WriteNoResponse – Hodnota se zapisuje clientem a čte serverem. Přijetí hodnoty z klienta na server není serverem klientu potvrzeno.
 - 0x08 Write – Hodnota se zapisuje clientem a čte serverem. Přijetí hodnoty z klienta na server je (asi) vždy serverem klientu potvrzeno.
 - 0x10 Notify – Při zápisu hodnoty serverem se generuje na klientu událost „ValueChanged“. Není však žádná informace zda client skutečně hodnotu obdržel.
 - 0x20 Indicate – Stejně jako „Notify“, ale před zápisem další hodnoty serverem je očekáváno potvrzení o přijetí hodnoty clientem (pouze HW přijetí přes BLE komunikaci, nikoli vyzvednutí / zpracování hodnoty aplikací).

Příklady vytvořených „Service(s)“ s „Characteristic(s)“ na serveru:

- Vytvoření dvou „Service(s)“
 - První „Service“ obsahuje dvě „Characteristic(s)“
 - Druhá „Service“ obsahuje jednu „Characteristic“

- Základní UUID = A1FCXXYY78D340C29B6F3C5F7B2797DF. Pozice XXXX jsou využity pro identifikaci „Service(s)“ a „Characteristic(s)“
- 'Service' ...0101... = A1FC010178D340C29B6F3C5F7B2797DF
 - o 'Char' ...0102... (Read, Write, Notify, 20 bytes)
 - o A1FC010278D340C29B6F3C5F7B2797DF,1A,14
 - o 'Char' ...103... (Write, 10 bytes)
 - o A1FC010378D340C29B6F3C5F7B2797DF,02,0A
- 'Service' ...0111... = A1FC011178D340C29B6F3C5F7B2797DF
 - o 'Char' ...0112... (Read, Write, Notify, 5 bytes)
 - o A1FC011278D340C29B6F3C5F7B2797DF,1A,05

Celkové poznámky:

- Pokud server zapíše data do „Characteristic“, tak se nijak nekontroluje, zda si je client vyzvedl (to se server nijak automaticky podle BLE standardu nedozví). Server tedy může kdykoli zapsat nová data do „Characteristic“ a tím v podstatě původní přepsat aniž by si je client (již už ještě) vyzvedl.
 - o „Notify“ je pouze událost (na klientu) o zápisu nových dat do „Characteristic“ serverem. V tomto případě při špatném spojení není ani jistota, že data ze serveru na klienta skutečně dorazila a „Notify“ na klientu bylo vyvoláno.
 - o „Indicate“ je potvrzení z klienta na server, že data ze serveru přes BLE komunikaci na klienta skutečně dorazila (a na klientu tedy vyvolala událost), ale nikoli že si je aplikace (pro zpracování dat) vyzvedla.
- Pokud je potřeba mít jistotu vyzvednutí / zpracování dat na straně klienta než server zapíše do „Characteristic“ nová data, tak (asi) jediná možnost spočívá v potvrzování přijatých dat cílovou / uživatelskou aplikací. Tedy server zapíše data (do „Characteristic“ s „Notify“), client data vyzvedne, zpracuje a zapíše (klidně do stejné „Characteristic“) data / informace jako potvrzení o zpracování, teprve poté server zapíše další data. Při tomto stylu komunikace poněkud klesne celková přenosová rychlost, ale je zaručena spolehlivost doručení dat.
- BLE zařízení nemusí vždy sloužit pro obousměrnou komunikaci, takovému BLE zařízení (v jistém smyslu server) se říká „Beacon“ (maják) a vůbec neslouží pro připojení klientů (připojení neumožňuje), ale slouží pouze pro vysílání informací obsažených v „Advertising paketu“. Takové zařízení může sloužit například jako maják pro radiovou orientaci v prostoru (směr, případně vzdálenost), jako vysílač časových značek pro synchronizaci, nebo jako externí / vzdálené teplotní čidlo (nelze se připojit, ale každou minutu poskytuje aktuální teplotu v ADVS paketu).

Další poznámky:

- Na straně serveru (asi) není (žádné) „Notify“ / „Indicate“. V případě HW BLE modulu jako serveru jsou generovány události pokaždé, když client zapíše data do „Characteristic“ serveru. Takže „Notify“ / „Indicate“ (asi) není potřeba, aplikace na serveru (vystupující jako server) se o novém zápisu dat do „Characteristic“ klientem tedy vždy dozví.