

## **Part II**

# **GATT design for Genotyper**

CONFIDENTIAL

## *Introduction*

The smartphone will act as client (central) and the Genotyper reader will act as server (peripheral).

User Scenarios :

1. GAP service
2. Notification of the battery level of Genotyper.
3. Defining the Power mode: (a)Battery mode (b)Charging mode (c) Connected but not charging mode.
4. Reader ID connected or not
5. Notification of whether Swab connected or not.
6. To read the swab ID
7. To read the reaction step ID
8. To read the matrix
9. To read the Ambient Temperature
10. To read the controlled environment temperature.
11. To write the reaction Control (a) Start (b)stop (c) Pause

## *Trade-offs and Constraints*

### **1. Custom UUID vs SIG Adopted UUID**

- Using SIG adopted UUID would considerably reduce the packet size as SIG adopted UUID is of 16 bits whereas custom UUIDs are of 128-bits.
- With the header included a single 128-bit UUID will take up 18 bytes of your advertise packet, leaving only 13 bytes for other fields. Typically you need 3 bytes for the flags, leaving only 10 bytes for a local name etc.
- But there are limited SIG UUIDs which may or may not correspond to the information that we want to send through BLE for our product.
- Registering a SIG adopted UUID for our product could be an option but the cost of registering a SIG-adopted ID would be 2,500 USD.
- The one apparent solution would be using Custom based UUIDs.

- One possible solution might be using Descriptors for exactly defining our characteristic, but it will degrade the quality of our product.
- One other possible way would be to use enums for some characteristics. Like for power mode, we could just values 0,1,2 it will reduce three characteristics to one. For more readability, we could add what 0 actually means in our app interface.

## 2. SIG adopted that could be used instead of customized ones

- (a) SWAB Service : User Data
- (b) Temperature Service: User Data
- (c) Detection matrix service : User Data
- (d) SWAB ID characteristic : System ID/Model Number String / Serial Number String/ Object ID/ Object Name
- (e) Detection matrix result characteristic: String/ Report/Report Map/ RSC measurement/ Digital Output/ Aggregate/ User Index/Database Hash
- (f) Reaction Control Service : RSC measurement(Reaction System Control)/ RSC Feature (Reaction System Control)
- (g) Power mode characteristic : Battery Level state/ Battery Power State

## 3. Matrix Transfer

- For transferring the matrix we could consider it as an array and we would need only one characteristic for transferring the whole matrix, but there is size constraint of 23 bytes. You can also think of using write long (max 512 bytes) which is supported from Softdevice v6.0.0 For more information click [here](#).
- We could also send the matrix like an image (if matrix size is large). We can divide the matrix into fragments and then send them as characteristic notifications. According to spec, the order of the received packets should be the same as the order of the packets sent. so it should be fine using notification.

## 4. Compatibility with the older device

- Our device should be able to communicate with the older devices as well.
- We come across two terms : Legacy advertisement and extended advertisement.

Item	Description	
Fast	You can configure timing information of advertising event. This parameter will be configurable if [Enable Fast Advertising] is checked. If not checked, parameter will be ignored. You can set following items.	
	Advertising Interval	Set Advertising Interval.
	Advertising period	Set Advertising Period. Parameters set in [Fast] will be used for this period.
Slow	You can configure timing information of advertising event. If [Enable Fast Advertising] is checked, this parameter will be used after Fast Advertising period. If not checked, this parameter will be used from the beginning of advertising operation. You can set following items.	
	Advertising Interval	Set Advertising Interval.
	Advertising period	Set Advertising Period. This parameter will be configurable if [Set Advertising Period] is checked. If you want to send Advertising only for certain period, please set this parameter.

Figure 10: Configuring the Advertising Parameter

- Legacy advertisement is compatible with older versions of BLE. it consists of some protocols.
- Extended Advertisement is only compatible with version 5. The older versions won't be able to discover these advertisement.
- Extended Advertisements are a way to advertise more (offloaded) data than what's allowed with Legacy Advertisements. Offloading is accomplished by first advertising on the primary channel that points to an auxiliary packet on the secondary channel.
- Here is a useful [link](#). For now, we will not use Extended advertisements.

#### 5. Achieving long range of over 1 km

The benefit of using the LE Coded PHY is increased range with the trade-off of both higher power consumption and reduced speed (down to 125kbps or 500 kbps)

For this Extended advertisements will be used which means no compatibility with older versions.

To understand more refer to the following [link](#) Achieving more range is the priority for our product so LE coded PHY should be used.

#### 6. Low power Consumption

For optimization we need to set various parameters which includes Advertising interval, Connection interval, slave latency etc. as well as choosing the right hardware components like battery.

We can configure the advertising parameter of our device. We aim to find whether there is a way by which we could differentiate the advertising interval of our characteristics. I have attached [link](#) to give the exact data about all these parameters. Other than these, the new 2M PHY also reduces power consumption since the same

amount of data is transmitted in less time reducing the radio-on time. Coexistence is also improved because of the less radio-on time.( Increased speed )

7. **Increasing the Packet size** Data Length Extension can be enabled or disabled. This feature allows the LE controller to send data channel packet data units (PDUs) with payloads of up to 251 bytes of application data, while in the connected state instead of 27 bytes.

Legacy Bluetooth Core Specification Versions 4.0 and 4.1 peer Hosts or Controllers may run to inter-operability issues. To support these older peers, it is recommended to completely disable the feature as outlined in Disabling DLE at Runtime. For more detailed information click [here](#) For our product, compatibility with previous is more of concern instead of large data packets.

8. **For connection with more than one device** Periodic advertisement : In this the data/ advertising data will get updated as well as after a certain period the advertisements will be sent. These are used for broadcasting packets to devices at a set period between two unconnected devices, meaning that more than one device can listen and tune in on these periodic advertisements.

### *Defining Services and Characteristics*

- **Battery level Service**

Battery level

characteristics : "Battery level"

Access Permission : Read, Notify

- **Device Firmware Update**

The pre-defined service name is Secure DFU Service.

It has a 16 bit UUID "0xFE59"

It has two Characteristics:

- DFU Control Point

UUID: 0x8EC90001-F315-4F60-9FB8-838830DAEA50

Permissions: Write, Notify

- DFU Packet

UUID: 0x8EC90002-F315-4F60-9FB8-838830DAEA50

Permissions: Write Without Response, Notify

For more details click [here](#)

- **Power Mode Service**

1. Battery mode  
characteristic : "Battery mode ON"  
Access Permission : Notify
2. Charging mode  
characteristic : "Charging mode ON"  
Access Permission : Notify
3. Connected but not charging mode  
characteristic: "Connected but not charging"  
Access Permission : Notify

- **SWAB service**

1. SWAB connected  
Characteristic : "SWAB Connected"  
Access Permission : Notify
2. SWAB Reader ID  
Characteristic : "SWAB ID"  
Access Permission : Read, Notify

- **Detection** Characteristic : "Detection matrix"  
Access Permissions : Read, Notify

- **Temperature Service**

1. Ambient Temperature  
Characteristic : "Ambient Temperature"  
Access Permission : Read, Notify
2. Controlled Environment Temperature  
Characteristic : "Controlled Temperature"  
Access Permissions : Read, Notify

- **Reaction Control**

1. Reaction Step ID  
Characteristic : "Reaction Step ID"  
Access Permission : Read, Notify
2. Reaction Start  
Characteristics : " Reaction Start"  
Access Permission : Write, Notify
3. Reaction Stop  
Characteristics : "Reaction Stop"  
Access Permission : Write, notify
4. Reaction pause  
Characteristics : "Reaction Pause"  
Access Permission : Write, notify

Table 2: GATT Design

Service	Characteristic	UUID type
GAP		SIG adopted
	Device Name	SIG adopted
	Appearance	SIG adopted
Battery level		SIG adopted
	Battery level	SIG adopted
Power Mode Service		Custom
	Battery mode ON	Custom
	Charging mode ON	Custom
	Connected but not charging	Custom
SWAB service		Custom
	SWAB Connected	Custom
	SWAB ID	Custom
Reaction Control		Custom
	Reaction Step ID	Custom
	Reaction Start	Custom
	Reaction Stop	Custom
	Reaction Pause	Custom
Temperature Service		Custom
	Ambient Temperature	Custom
	Controlled Temperature	Custom
Secure DFU Service		SIG adopted
	DFU Control Point	SIG adopted
	DFU Control Packet	SIG adopted
Detection		Custom
Detection Matrix		custom

### Important links

1. **Device Firmware Update** The per-defined service name is Secure DFU Service. It has a 16 bit UUID "0xFE59" It has two Characteristics:

- (a) DFU Control Point

UUID: 0x8EC90001-F315-4F60-9FB8-838830DAEA50

Permissions: Write, Notify

- (b) DFU Packet

UUID: 0x8EC90002-F315-4F60-9FB8-838830DAEA50

Permissions: Write Without Response, Notify

For more details click [here](#)

### *Problem faced*

- **Error occurred adding service (NRF\_ERROR\_NO\_MEM):** Failed to add services

Error converting characteristic into driver

Error occurred adding service

It might be due to the limitaion of nrf connect.

One solution is to get a chip with larger RAM, i.e. the nRF51822 QFAC variant which has 32kB of RAM. [More Information](#)

Another solution, if possible, is to put the application on RAM diet.

The NRF\_ERROR\_NO\_MEM error can be generated by different things.

- **Memory allocation in nrf Dongle:** 1 MB FLASH and 256 kB RAM. The memory manager implements fixed-size block allocation. The fixed-size blocks are categorized into seven categories: xxsmall, xsmall, small, medium, large, xlarge, xxlarge. The application can configure how big each block should be and how many of these blocks should be allocated. Whenever the user requests a particular amount of memory, the module assigns a block usually the smallest possible block to the application instead of providing it the exact amount of memory. If the smallest block is not available then next possible free block is assigned.

The block allocator library enables the application to create a fixed-size block pool in RAM.

[Memory allocation](#)

[Block allocator library](#)

- The maximum length of an attribute (Services, Characteristics, and Descriptors) value shall be 512 octets" which leads us to the answer: 512 bytes.
- Using programmer app to see the distribution of memory inside the nrf Dongle. The memory is divided into 5 sections:
  1. **Bootloader section:** The bootloader is responsible for determining the start address of the application. The bootloader is also responsible for keeping track and verifying the integrity of the firmware, including the application, SoftDevice, and the bootloader itself.
  2. **Softdevice section:** The SoftDevice shares the available flash memory and RAM on the nRF52 System on Chip (SoC) with the application. The application must therefore be aware of the memory resources needed by the SoftDevice and leave the parts



of the memory used by the SoftDevice undisturbed for correct SoftDevice operation.

3. **MBR section** : It stands for Master boot record. The main functionality of the MBR is to provide an interface to allow in-system updates of the application, the SoftDevice, and boot-loader firmware. The MBR module occupies a defined region in the System on Chip (SoC) program memory where the System Vector table resides.
4. **Application Section**
5. **unknown**

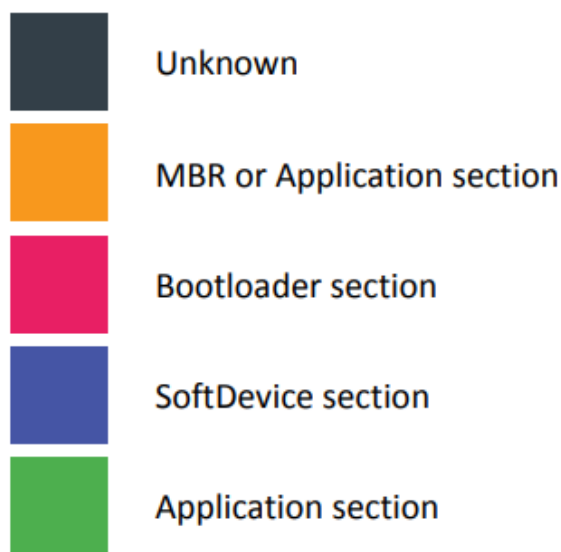


Figure 11: Memory layout section colors

- In nrf Connect, the size is in bytes.
- From the observations that I have done from my part, I have realised that once the error comes then even after deleting some services it will show the error again and again.  
If I tried to apply previous setup (that was successfully getting implemented before ), it would show the error. Also the memory of the application and softdevice remain same even after adding services.
- How to resolve unknown service and unknown characteristic in BLE?

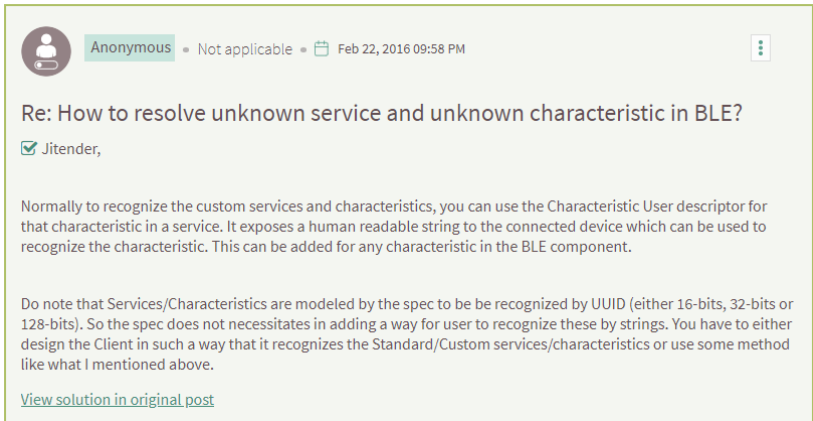


Figure 12: Possible solution for unknown services and unknown characteristics

Service	softdevice size (bytes)	Size in bytes
GAP	139264	184344
Battery level	139264	184344
Power Mode Service	139264	184344
SWAB service	139264	184344
Reaction Control	139264	184344
Temperature Service	139264	184344
Secure DFU Service	139264	184344
Detection	139264	184344
DFU, Batt	139264	184344
DFU, Batt, Temp	139264	184344
DFU, Batt, Temp, Detection	139264	184344
DFU, Batt, Temp, Detection, SWAB	139264	184344
DFU, Batt, Temp, Detection, SWAB, rxn(2)	error	error

Table 4: Memory usage of each service

*nrf52811*

The memory specification of this device: 192 KB Flash + 24 KB RAM

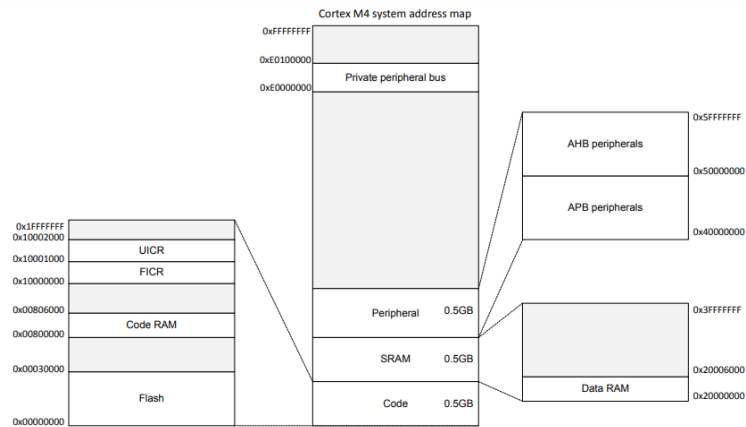


Figure 13: Memory map of nrf52811

Service	Characteristic	UUID type
GAP		SIG adopted
	Device Name	SIG adopted
	Appearance	SIG adopted
Battery level		SIG adopted
	Battery level	SIG adopted
	Ambient Temperature	Custom
	Controlled Temperature	Custom
User Data Service		SIG adopted
	Reaction Control	Custom
	SWAB Connected	Custom
	Report	SIG adopted
Device Information		SIG adopted
	Object ID	SIG adopted
	Power Mode	Custom
	Reaction Step ID	Custom
Secure DFU Service		SIG adopted
	DFU Control Point	SIG adopted
	DFU Control Packet	SIG adopted

Table 6: Final GATT Design