



SAP und IoT - Die Anbindung von Bluetooth Low Energy (BLE) an SAP

Thomas Randl
Fakultät für Informatik

WS 2020/21

ToDo

Inhaltsverzeichnis

1	Einleitung	4
2	Vorgestellte Technologien	4
2.1	IoT	4
2.2	Industrie 4.0	4
2.3	SAP	4
3	Allgemeine Funktionsweise Bluetooth Low Energy	4
3.1	Protokollstack	4
3.2	Kommunikation	7
3.2.1	Advertisement	7
3.2.2	Verbindung	7
3.3	Anwendungsszenarien	7
4	Schnittstellenbeschreibung	7
4.1	SAP	7
4.2	BLE	7
4.3	Anbindungsmöglichkeiten	7
4.3.1	Bewertung der Möglichkeiten	7
4.3.2	Administration	7
5	Vergleich mit anderen gängigen IoT Kommunikationsprotokollen	7
5.1	Vorteile	7
5.2	Nachteile	7
6	Fazit	7

Abkürzungsverzeichnis

BLE	Bluetooth Low Energy
GAP	Generic Access Profile
GATT	Generic Attribute Profile
HCI	Host Controller Interface

1 Einleitung

2 Vorgestellte Technologien

Im folgenden Kapitel wird einleitend eine kurze Erläuterung über die drei zentralen Begriffe dieser Arbeit gegeben. Besonderer Schwerpunkt ist dabei die Aufschlüsselung der Begriffe und die Erklärung warum Diese in dieser Arbeit einen derartigen Stellenwert besitzen.

2.1 IoT

2.2 Industrie 4.0

2.3 SAP

3 Allgemeine Funktionsweise Bluetooth Low Energy

Im nachfolgenden Kapitel wird nun ein kurzer Überblick über die Technologie Bluetooth Low Energy (BLE) gegeben. Dabei wird zum einen die Architektur anhand des Protokollstacks und zum anderen die Funktionsweise erläutert.

3.1 Protokollstack

Die Architektur von BLE lässt sich anhand des Protokollstacks erkennen. in Abbildung 1 ist dieser dargestellt. Besonders auffällig ist, dass er in drei Ebenen unterteilt ist. Der „Controller“ ist dabei der Bereich, welcher am nächsten an der Hardware sitzt. Hier befinden sich die zwei Lagen „LE Physical“ und „Link“.

Diese beiden Protokolle sind in einer Großzahl von Gerätearchitekturen beheimatet. Das Physical Layer ist dafür vorgesehen, digitale Signale (Bitfolgen) in analoge umzuwandeln. Dieser Schritt wird benötigt, um die BLE Signale für etwaige Empfänger zugänglich zu machen. Natürlich werden im Physical Layer auch empfangene analoge Signale in digitale umgewandelt. Dieser werden dann im Protokollstack nach oben ins Link Layer weitergereicht. Im Fall von BLE ist die Schnittstelle für das Senden der analogen Signale die Luft. BLE sendet daher im Frequenzbereich beginnend bei 2,4GHz bis 2,4835GHz [Ran20, Seite 7]. Diesen Bereich teilt sich das Protokoll mit anderen wie zum Beispiel Wifi. Um Kollisionen zu vermeiden teilt BLE den Bereich in 40 Kanäle auf und wechselt während der Verbindung in regelmäßigen Abständen oder bei Übertragungsproblemen den Kanal. Dieser Ansatz nennt sich Frequency Hopping Spread Spectrum [Tow14, Seite 17].

Das Link Layer unterscheidet sich in seiner Funktionsweise nicht großartig von dem anderer Kommunikationsprotokolle. Hier werden die Nachrichten die für das Versenden

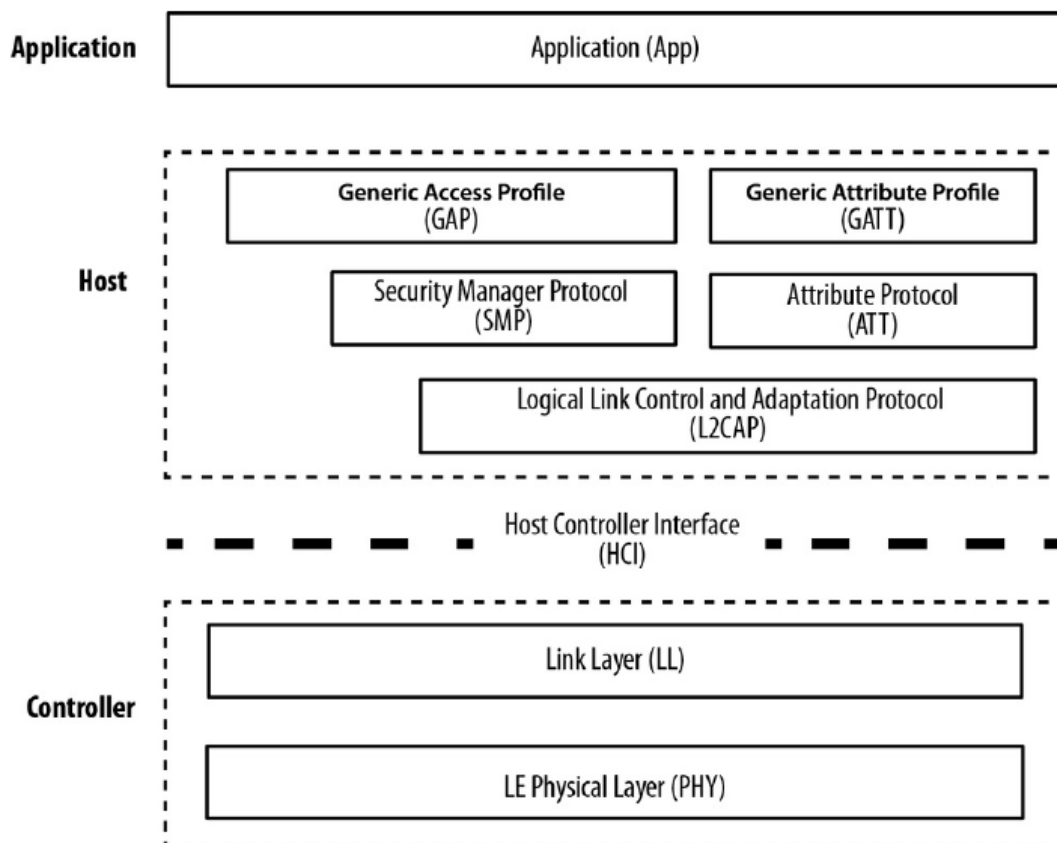


Abbildung 1: BLE Protokollstack [Tow14, Seite 16]

aus den oberen Schichten ankommen in Pakete gepackt und an das Physical Layer weitergereicht. Dieser Prozess ist für ankommende Pakete natürlich vice versa [Tan14, Seit 194].

Im Link Layer von BLE wird die Paketgröße festgelegt. Diese lässt seit Version 4.2 eine Payload von 251 Byte pro Paket zu [Gup]. Zum Vergleich bei WLAN (IEEE 802.11) kann ein Paket bis zu 2312 Byte groß sein [MTU]. Daraus lässt sich schließen, dass BLE einen weitaus geringeren Datendurchsatz als WLAN hat. Allerdings liefert BLE wiederum andere Vorteile. Auf diese wird in Kapitel 5 näher eingegangen.

Das Herzstück des BLE Protokollstacks bilden die beiden Profile Generic Access Profile (GAP) und Generic Attribute Profile (GATT). Diese befinden sich wie in Abbildung 1 zu erkennen im Host Bereich der Architektur. Die beiden Protokolle bilden die Schnittstelle zur tatsächlichen Anwendung mit welcher der Nutzer letzten Endes interagiert.

Das GAP ist dafür vorgesehen sämtliche Parameter der Verbindung zwischen den Geräten zu verwalten. Vom Verbindungsaufbau bis hin zur Kommunikation werden sämtliche Funktionen von diesem Profil bereitgestellt und abgehandelt.

Im Zuge der jeweiligen Konfiguration kann ein Gerät in BLE eine der folgenden vier

Rollen annehmen:

- Broadcaster (Keine Verbindung)
- Observer (Keine Verbindung)
- Central (Verbindung)
- Peripheral (Verbindung)

In BLE ist es nicht festgeschrieben, dass Geräte eine Verbindung eingehen müssen um Informationen zu erhalten. Die Rollen des Broadcasters und Observers sind sogar ausschließlich ohne feste Verbindung zwischen den Geräten vorgesehen. Diese Funktion wird im allgemeinen gerne von BLE Beacons verwendet.

Ein Gerät welches als Broadcaster definiert ist sendet dauerhaft einen bestimmten Datensatz. Dabei ist zu keinem Zeitpunkt klar, ob Geräte in Reichweite sind, welche den Datensatz empfangen. Ein Gerät welches diese Daten lesen kann muss als Observer konfiguriert sein. Ein solcher scannt die drei Advertisement Kanäle von BLE dauerhaft nach Broadcastnachrichten. Falls er eine erhält liest er diese und verwendet sie. Wichtig ist hierbei, dass der Observer keine Antwort auf eine Nachricht sendet.

Sollte ein Gerät allerdings eine Verbindung eingehen, dann muss dieses als Central konfiguriert sein. Dies ist die gängigste Form des BLE Gerätes. So ist beispielsweise jedes Smartphone in der Regel als Central konfiguriert und kann Verbindungen zu Peripherals wie zum Beispiel BLE Kopfhörern aufnehmen. Dabei ist ein Central in der Regel sogar in der Lage mehrere Verbindungen zu selben Zeit einzugehen.

Ein Peripheral wiederum ist das Gegenstück zum Central, welches seine Verbindungsbereitschaft an sämtliche Geräte in Reichweite signalisiert. Im Fall einer aktiven Verbindung übernimmt das Central die Steuerung des Gerätes unter Berücksichtigung des Funktionsumfangs des Peripherals. [Ran20, Seite 15]

Das GAP ermöglicht es einem BLE Gerät zusätzlich seine Sichtbarkeit und Verbindungsbereitschaft gegenüber anderen Geräten über die Advertisement Kanäle mitzuteilen. Dafür wird auf dem Gerät ein Modus eingestellt welcher anschließend an alle Geräte in Reichweite mitgeteilt wird. Ein Modus ist dabei eng mit der Geräterolle verbunden. Ein Gerät kann folgende Modi annehmen [Ran20, Seite 16]:

- Broadcast (Rolle: Broadcaster)
- Nicht zu entdecken (Rolle: Peripheral)
- Eingeschränkt zu entdecken (Rolle: Peripheral)
- Normal zu entdecken (Rolle: Peripheral)
- Nicht verbindbar (Rolle: Alle)

- Verbindbar (Rolle: Central, Peripheral)

Im GATT wird definiert, ob es sich bei dem Gerät um einen Client oder Server handelt. Zweiterer verarbeitet die Kommunikationsanfragen des Clients und liefert die gewünschten Antworten oder führt entsprechende Aktionen aus. Der Server ist in der Regel ein Peripheral auf dem Services hinterlegt sind [Usa17, Seite 30]. Welche das sind und was für eine Aktion mit diesen verbunden ist wird in der Regel durch die Nutzeranwendung festgelegt. Der Client ist dementsprechend ein Central, welches die gesamte Verbindung steuert.

An oberster Stelle des Protokollstacks befindet sich die Nutzeranwendung. Diese ist nach dem entsprechenden Use Case programmiert und variiert von Anwendung zu Anwendung. Ausschließlich der Stack unterhalb ist für alle Applikationen gleich.

3.2 Kommunikation

Todo: Rollen Verbindung und Kommunikation

3.2.1 Advertisement

3.2.2 Verbindung

3.3 Anwendungsszenarien

4 Schnittstellenbeschreibung

4.1 SAP

4.2 BLE

4.3 Anbindungsmöglichkeiten

4.3.1 Bewertung der Möglichkeiten

4.3.2 Administration

5 Vergleich mit anderen gängigen IoT Kommunikationsprotokollen

5.1 Vorteile

5.2 Nachteile

6 Fazit

Literatur

- [Gup] S. Gupta und R. Kumar. BLE v4.2: Creating Faster, More Secure, Power-Efficient Designs - Part 1. <https://www.electronicdesign.com/communications/ble-v42-creating-faster-more-secure-power-efficient-designs-part-1>. Last visit: 15 Okt 2020.
- [MTU] MTU - Maximum Transfer Unit. <https://www.elektronik-kompodium.de/sites/net/0812211.htm>. Last visit: 15 Okt 2020.
- [Ran20] T. Randl. Funktionsweise und Einordnung in den Bereich der IOT Kommunikationsprotokolle. In *Bluetooth Low Energy*, S. 7, 15 – 16. 2020.
- [Tan14] A. Tanenbaum und D. Wetherall. In *Computer Networks*, Bd. 5, S. 194. Pearson, Harlow (Vereinigtes Koenigreich), 2014.
- [Tow14] K. Townsend, C. Cufi, Akiba und R.Davidson. Tools and techniques for lowpower networking. In *Getting Started with Bluetooth Low Energy*, S. 16 – 18. O'Reilly Media Inc., Sebastopol (Vereinigte Staaten von Amerika), 2014.
- [Usa17] M. Usama und B. Aftab. Take your first steps in IoT. In *Building Bluetooth Low Energy Systems*, S. 30. Packt, Birmingham (Vereinigtes Koenigreich), Mumbai (Indien), 2017.