

Bluetooth
&
Symbian

Adalbert Prokop

5. Juli 2000

Inhaltsverzeichnis

1 Was ist Bluetooth?	2
1.1 Funktionsweise	2
1.2 Kosten	2
1.3 Vorteile	2
2 Geschichte von Bluetooth	3
2.1 Gründer der SIG Bluetooth	3
2.2 Mitglieder der SIG Bluetooth	3
3 Hardware-Spezifikation	3
3.1 Geräteklassen	3
4 Radio-Spezifikation	3
4.1 Frequenzen	3
4.2 Kanäle	4
4.3 Slots	4
4.4 Verbindungsarten	5
4.4.1 SCO-Verbindungen	5
4.4.2 ACL-Verbindungen	5
4.5 Paketformate	5
4.5.1 Access Code	6
4.5.2 Preamble	6
4.5.3 Sync Word	7
4.5.4 Trailer	7
4.5.5 Header	7
4.5.6 AM_ADDR	7
4.5.7 TYPE	7
4.5.8 FLOW	8
4.5.9 ARQN	8
4.5.10 SEQN	8
4.5.11 HEC	8
5 Fehlerkorrektur	8
5.1 1/3 FEC	8
5.2 2/3 FEC	8
5.3 ARQ	9
6 Sicherheitsaspekte	9
7 Service-Discovery Protokoll	10
7.1 Was SDP (nicht) kann	10
7.2 Wie sehen SDP-Informationen aus	11
7.3 Attributbeispiele	12
8 Bluetooth vs. IrDA	12
9 Symbian	12
9.1 EPOC	12

1 Was ist Bluetooth?

1.1 Funktionsweise

Bluetooth ist Kurzstrecken-Funktechnik, mit der Distanzen von 0,1 m bis zu 10 m überwunden werden können. Für Spezialanwendungen sind Distanzen bis zu 100 m möglich.

Bluetooth-Geräte verwenden das ISM-Band (Industrial – Scientific – Medical) auf der Frequenz von 2,4 GHz. Der Frequenzbereich reicht von 2.400 bis 2.483,5 MHz. Dieser Bereich ist in den meisten Ländern lizenzfrei. Allerdings gibt es einige Einschränkungen. So haben z. B. Spanien, Frankreich und Japan das ISM-Band teilweise belegt.

Die Bluetooth-Spezifikation teilt das ISM-Band in 79 Kanäle auf, die einen Abstand von 1 MHz angeordnet sind. Somit lassen sich die verfügbaren Kanal-Frequenzen wie folgt ermitteln: $f = 2402 + k$ MHz, $k = 0, \dots, 78$. MUm die Verbindungen weitgehend störungsunempfindlich zu machen (immerhin konzentrieren sich im ISM-Band viele Störquelle, z. B. Mikrowellenherde, etc.), wird der Verbindungskanal ständig, nach einem pseudo-zufälligen Muster, gewechselt. Die Wechsel (sog. *Channel hops*) erfolgen mit einer Frequenz von 1.600 Hz.

Bluetooth-Geräte bilden selbständig sogenannte *Pico*-Netze (PicoNet), die ihrerseits wieder zu sog. *Scatter*-Netz (ScatterNet) zusammengefasst werden können. Jedes Pico-Netz besteht aus einem *master* und bis zu 7 *slaves*. Ein Pico-Netz hat nur einen *master*. Jeder *master* oder *slave* kann Teilnehmer in weiteren Pico-Netzen sein. Dadurch entsteht ein Scatter-Netz

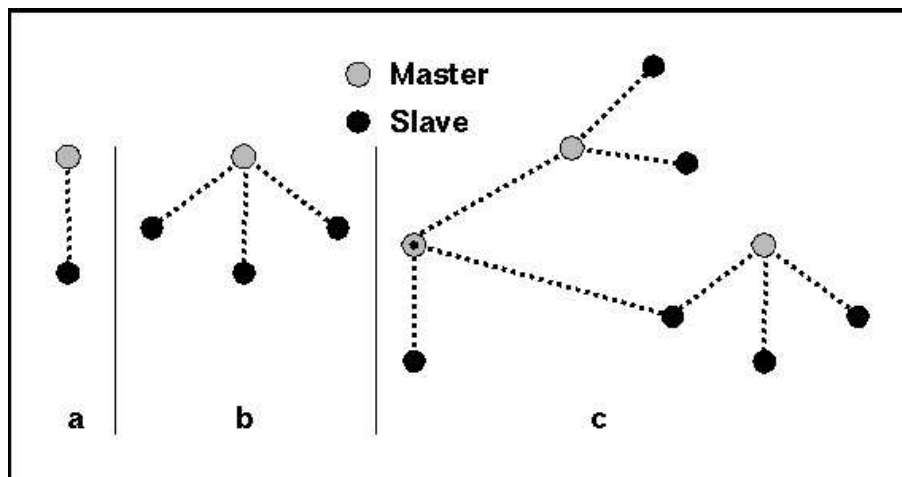


Abbildung 1: PicoNets (a), (b) und ScatterNet (c)

1.2 Kosten

Die Kosten für einen Sende-Chip betragen heute noch \$ 20–\$ 30, sind aber durch höhere Integration und Massenproduktion auf weniger als \$ 5 zu reduzieren.

Für Entwickler gibt es derzeit von Ericsson einen **Bluetooth Developer Kit** für £ 9.000 (ca. 25.000,- DM) und einen **Bluetooth Starter Kit** für \$ 3.000

1.3 Vorteile

Die Bluetooth-Technik bringt einige Vorteile mit sich:

- Keine direkte Sichtverbindung zwischen den Geräten notwendig, geschweige denn Kabel,
- auch Verbindungen durch Wände etc. sind möglich,
- keine Ausrichtung notwendig – Funk strahlt in alle Richtungen ab,

- sehr störungsunempfindlich durch schnelle Kanalwechsel,
- bringt eine Datenverschlüsselung mit,
- kann das TCP/IP Protokoll weiterleiten
- ist eine billige Lösung.

2 Geschichte von Bluetooth

2.1 Gründer der SIG Bluetooth

Der Name *Bluetooth* geht zurück auf den dänischen König *Harald Blåtand* (mit dem Beinamen *Blauzahn*), der im 10. Jahrhundert Dänemark und Norwegen vereinen wollte.

Die Gründer der *Special Interest Group* (SIG) Bluetooth waren Ericsson, IBM, Intel, Nokia, und Toshiba. Im Mai 1998 haben sie sich als Ziel gesetzt „die drahtlose Verbindungstechnik für die privaten und kommerziellen Mobilgeräte zu revolutionieren“. (*revolutionize wireless connectivity for personal and business mobile devices*)

2.2 Mitglieder der SIG Bluetooth

Am 30. August 1998 waren neben den Gründungsmitgliedern mehr als 60 Firmen Mitglieder in der SIG Bluetooth, u. a. so bedeutende Namen wie 3Com, Casio, Compaq, Logitech, Motorola, Psion, Epson oder JavaSoft.

Im Juni 1999 zählte die SIG mehr als 750 Mitglieder, im September 1999 waren es 1000, im Dezember 1999 gab es 1.300 Mitglieder und im April 2000 waren es über 1.700 Mitglieder. Der aktuelle Anzahl beträgt mehr als 1.800 Mitglieder.

Interessanterweise gehört Microsoft weder zu den Gründungsmitgliedern, noch zu den „Mitgliedern der Ersten Stunde“. Im Jahr 1999 ließ Microsoft verlauten: „Microsoft is currently weighting the importance of providing Bluetooth support in Windows. Regardless of the result of this process, Microsoft does not plan to provide any specific support for this technology in the upcoming release of Windows 2000“. Kürzlich jedoch haben Intel und Microsoft angekündigt, zusammen an einer Bluetooth-Integration in Windows arbeiten zu wollen. Jawad Khaki, stellvertretender Leiter der Netzwerk Gruppe von Microsoft kündigte das Erscheinungsdatum auf die erste Hälfte vom Jahr 2001 an.

3 Hardware-Spezifikation

3.1 Geräteklassen

Die gesamte Bluetooth-Einheit lässt sich auf einem Breich von der Größe eines 2 DM Stückes unterbringen. Desweiteren kann man natürlich größere und leistungsfähigere Geräte bauen. Die Bluetooth-Spezifikation sieht Einteilung der Geräte in drei Klassen vor.

Nur Geräte der Klasse 1 müssen eine Sendeleistungsregelung besitzen, für Geräte der Klassen 2 und 3 ist sie optional.

4 Radio-Spezifikation

4.1 Frequenzen

Das zur Verfügung stehende Frequenzbereich in USA und Europa¹ ist 2.400 – 2.483,5 MHz. Das Spektrum teilt sich auf in 79 RF-Kanäle mit einem Abstand von 1 MHz, beginnend bei 2.402 MHz.

¹bis auf Frankreich und Spanien

Geräte-Klasse	Maximale Leistung	Nominale Leistung	Minimale Leistung	Leistungsregelung
1	100 mW (20 dBm)	—	1 mW (0dBm)	in Schritten zwischen 2 dB und 8 dB für Sendeleistung über 0 dBm
2	2,5 mW (4 dBm)	1 mW (0 dBm)	0,25 mW (-6 dBm)	optional
3	1 mW (0 dBm)	—	—	optional

Tabelle 1: Bluetooth-Geräteklassen

Für Spanien, Frankreich und Japan steht nur ein eingeschränkter Frequenzbereich zur Verfügung. In diesen Ländern wird nur mit 23 RF-Kanälen gearbeitet, die auch jeweils im Abstand von 1 MHz angeordnet sind.

Geräte, die für den eingeschränkten Frequenzbereich entwickelt wurden sind nicht mit Geräten kompatibel, die das gesamte ISM-Band nutzen. Vielmehr stellen sie eine lokale Version für den heimischen Markt dar.

4.2 Kanäle

Die Bluetooth-Geräte kommunizieren auf sog. Kanälen (*channels*), die durch einen pseudo-zufälligen Wechsel der RF-Kanäle (79 oder 23) definiert werden. Die Wechselfolge ist individuell für jedes Pico-Netz und wird aus der Geräteadresse des *masters* ermittelt. Der Offset (Phasenverschiebung) der Wechselfolge wird aus dem *clock*-Wert der *masters* bestimmt.

Ein Kanal ist in *slots* unterteilt, wobei jeder *slot* einem RF-Kanal entspricht. Alle Geräte im Pico-Netz bleiben zu ihrem Kanal Zeit- und Wechselsynchronisiert.

Die nominale Wechselgeschwindigkeit des Kanals ist 1.600 Hz

4.3 Slots

Die Slots haben eine Länge von $625 \mu\text{s}$ ($= \frac{1}{1600}\text{s}$). Die Slots sind nach der Uhr des Masters durchnummeriert, von 0 bis $2^{27} - 1$, was etwa einer Durchlaufzeit von einem Tag entspricht. ($625 \mu\text{s} \cdot (2^{27} - 1) \approx 83886\text{s} \approx 23,3\text{h}$) Die Nummerierung erfolgt zyklisch.

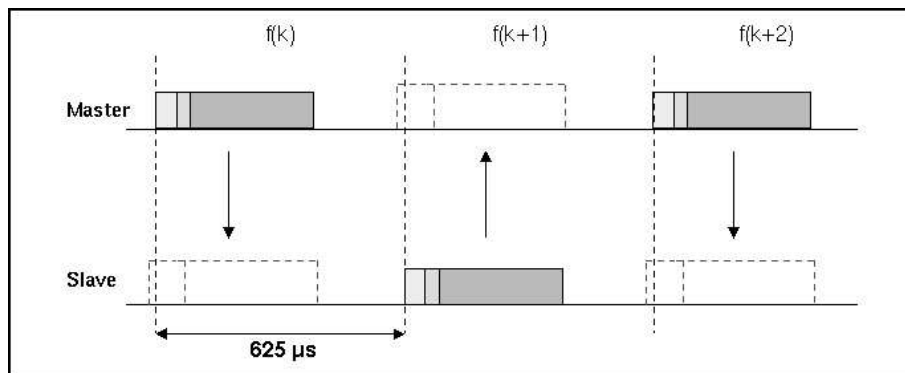


Abbildung 2: Timing bei Single-Slot Paketen

In den Slots können der Master und die Slaves Daten austauschen. Der Master darf die Sendung nur an geraden Slots beginnen, die Slaves nur an ungeraden Slots. Der RF-Kanal (der *physikalische* Kanal) darf während des Sendens eines Datenpaketes nicht geändert werden.

Multi-Slot Pakete haben die Länge von 3 oder 5 Slots. Da der RF-Kanal während der Paketübermittlung nicht geändert werden darf, wird das Multi-Slot Paket auf *einer* Frequenz gesendet ($f(k)$)

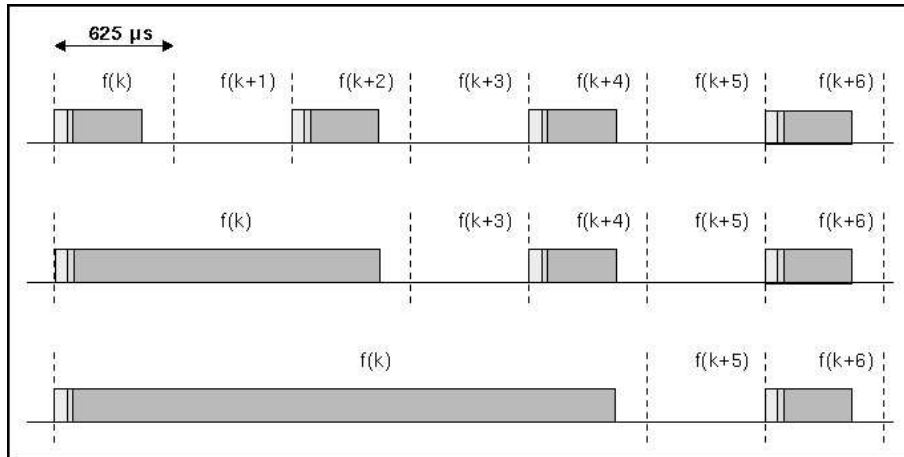


Abbildung 3: Multi-Slot Pakete

und die anschließenden Pakete werden auf den entsprechenden $f(k+4)$ oder $f(k+6)$ Frequenzen gesendet (s. Abbildung).

4.4 Verbindungsarten

Bluetooth bietet zwei verschiedene Verbindungsarten: synchrone (*synchronous connection-oriented, SCO*) und asynchrone (*asynchronous connection-less, ACL*). Die SCO-Verbindungen werden überwiegend zur Sprachübermittlung genutzt, während die ACL-Verbindungen meist für Datenübertragung genutzt werden. Für die SCO-Verbindung werden Slots in einem festen Intervall reserviert. In den nicht reservierten Slots kann eine ACL-Verbindung aufgebaut werden.

4.4.1 SCO-Verbindungen

Eine SCO-Verbindung kann nur zwischen einem Master und einem Slave aufgebaut werden. Der Master kann bis zu drei SCO-Verbindungen aufbauen – entweder zu demselben oder zu verschiedenen Slaves. Ein Slave unterstützt bis zu drei SCO-Verbindungen zu demselben Master oder zwei SCO-Verbindungen zu unterschiedlichen Masters.

Der Slave darf auf ein SCO-Paket im folgenden Slot antworten, es sei denn, es wurde ein anderes Gerät adressiert. Konnte die Geräteadresse nicht dekodiert werden, darf der Slave trotzdem im folgenden Slot antworten.

Fehlerhafte Pakete werden grundsätzlich *nicht* wiederholt, da die Daten ihre zeitliche Relevanz nicht verlieren dürfen.

4.4.2 ACL-Verbindungen

Eine ACL-Verbindung wird aufgebaut zwischen einem Master und *allen* Slaves eines Pico-Netzes. Es darf nur eine ACL-Verbindung zwischen einem Master und einem Slave bestehen. Für die meisten ACL-Pakete ist eine Wiederholung der Sendung vorgesehen, um Datenintegrität zu gewährleisten.

Ein Slave darf nur dann auf ein ACL-Paket antworten, falls es im vorhergehenden Slot adressiert war. Konnte die Geräteadresse nicht dekodiert werden, darf der Slave nicht antworten.

4.5 Paketformate

Die Bit-Interpretation folgt dem *little endian format*, d. h.:

- Das niederwertigste Bit (*least significant bit, LSB*) ist das b_0 (s. Abbildung);

- Das LSB ist das erste Bit, das in den Äther gesendet wird;
- In Abbildungen ist das LSB auf der linken Seite zu finden.

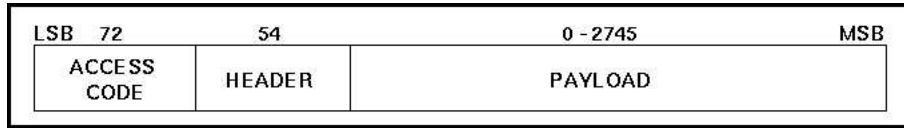


Abbildung 4: Das reguläre Paketformat

Jedes Paket besteht aus 3 Teilen: *access code*, *header* und *payload*. Ein Paket kann nur aus dem Access Code, aus Access Code und Header, sowie aus allen drei Teilen bestehen. Besteht das Paket nur aus dem Access Code, ist seine Länge auf 68 Bit beschränkt.

4.5.1 Access Code

Der Access Code wird zu Synchronisation, DC Offset-Ausgleich und Identifikation verwendet.

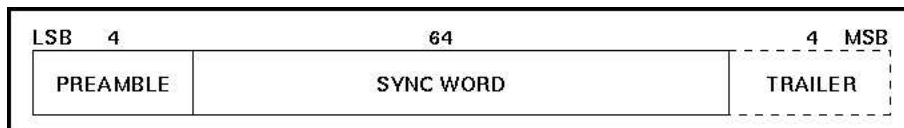


Abbildung 5: Das Access-Code Format

Es gibt drei verschiedene Arten der Access Codes:

- *Channel Access Code* (CAC)
- *Device Access Code* (DAC)
- *Inquiry Access Code* (IAC)

CAC wird in allen Datenpaketen verwendet, die in einem Pico-Netz ausgetauscht werden. DAC wird in Verwaltungspaketen zur Signalisationszwecken verwendet. IAC (es gibt zwei Arten: GIAC *general IAC* und DIAC *dedicated IAC*) werden verwendet, um alle Bluetooth-Geräte in der Nähe (GIAC) oder um Geräte mit bestimmten Eigenschaften (z. B. alle Drucker, Modems, etc.) (DIAC) auffindig zu machen.

Das CAC enthält **preamble**, **sync word** und **trailer**, während DAC und IAC meistens nur **preamble** und **sync word** enthalten, falls kein **header** folgt.

4.5.2 Preamble

Die **preamble** ist eine feste Folge von Nullen und Einsen, entweder **1010** (falls LSB des **sync words** 1 ist) oder **0101** (falls LSB des **sync words** 0 ist).



Abbildung 6: **preamble**

4.5.3 Sync Word

Die 64 Bit des **sync words** werden aus den 24 niederwertigsten Bits der Geräte-Nummer abgeleitet (*lower address part, LAP*). Im CAC wird die LAP des Masters verwendet, bei IAC werden spezielle, reservierte LAPs verwendet und bei DAC wird die LAP des angesprochenen Slaves verwendet.

4.5.4 Trailer

Trailer besteht – wie auch die **preamble** – aus einer festen Bitfolge, entweder **1010** oder **0101**, vom höchwertigen Bit (*most significant bit, MSB*) des **sync words** abhängig.

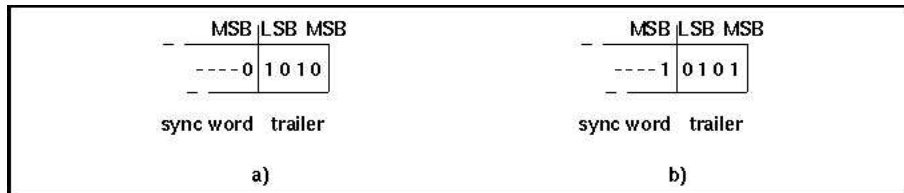


Abbildung 7: trailer

4.5.5 Header

Der Header besteht aus 6 Feldern, die folgende Informationen tragen:

- Adresse eines aktiven Gerätes;
- Typ-Code;
- Flußkontrollfeld;
- Bestätigungsindikator;
- Sequenznummer;
- Prüfsumme;

Der Header besteht aus 18 Bits, die mit einer 1/3 FEC codiert werden (anders gesagt: jedes Bit wird zwei mal wiederholt).

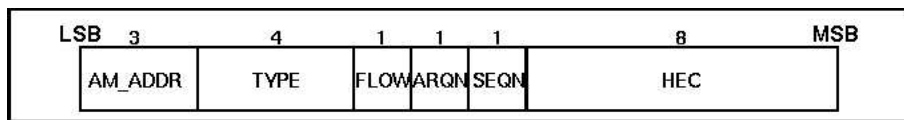


Abbildung 8: Header-Format

4.5.6 AM_ADDR

AM_ADDR ist ein 3 Bit lange, temporäre Mitgliedskennung in einem Pico-Netz (deswegen hat ein Pico-Netz höchstens 7 Slaves). Die AM_ADDR-Kennung des Slaves wird in allen Master-Slave und Slave-Master Paketen verwendet. AM_ADDR=0 ist für allgemeine Nachrichten an *alle* Slaves reserviert.

4.5.7 TYPE

Da das TYPE-Feld 4 Bit lang ist, können 16 verschiedene Paketarten definiert werden. Das TYPE-Feld definiert in erster Line, ob es sich um ein SCO oder ACL Paket handelt.

4.5.8 FLOW

Dient der Flußkontrolle. Sobald der Empfangspuffer eines Gerätes voll ist, wird im folgenden Paket FLOW=0 gesendet. Ab dann sind nur Signalpakete (NULL, POLL, ID) erlaubt. Ein leerer Empfangspuffer wird durch FLOW=1 (GO) angezeigt.

4.5.9 ARQN

ARQN wird verwendet, um den Sender über den Zustand der Daten (im **payload**-Feld) zu informieren. War CRC des letzten Paketes OK, wird ARQN=1 (ACK) gesendet, anderenfalls ARQN=0 (NAK).

4.5.10 SEQN

Das SEQN-Bit wird bei jedem gesendetem Paket invertiert. Sollte ein Paket-Wiederholung notwendig sein (z. B. wegen aufgetretener Übertragungsfehler) und es kommen zwei Pakete mit identischen Daten an, so läßt sich über dieses Bit feststellen, ob es sich um eine Wiederholung handelt, oder ob es ein neues Paket mit gleichen Daten ist.

4.5.11 HEC

Aus den vorhergehenden 10 Bits des Headers werden nach einem bestimmten Muster 8 Bit der HEC-Prüfsumme berechnet.

5 Fehlerkorrektur

Bluetooth-Spezifikation sieht 3 Fehlerkorrekturmöglichkeiten vor:

- 1/3 FEC
- 2/3 FEC
- ARQ-Schema

5.1 1/3 FEC

1/3 FEC bedeutet, dass aus n Eingabe-Bits $3n$ Ausgabe-Bits folgen. Es wird die denkbar einfachste Lösung verwendet: jedes Bit wird 3 mal wiederholt...

5.2 2/3 FEC

Dieses Korrekturschema operiert nur auf $n \cdot 10$ Bits, so das von Fall zu Fall das Auffüllen mit Nullen notwendig ist.

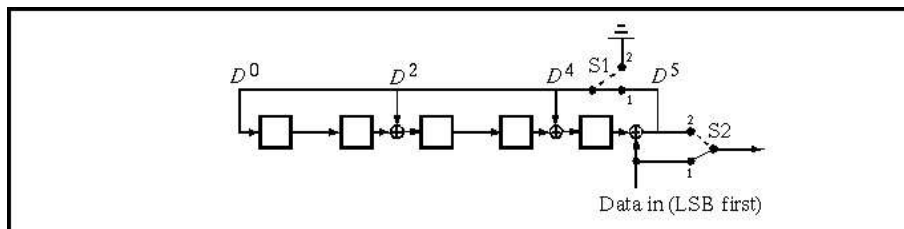


Abbildung 9: LFSR

Zu Anfang werden alle Register mit Nullen initialisiert. Die Schalter befinden sich in der Stellung 1. Nach und nach werden die 10 Informationsbits in den LFSR geschoben. Sind 10 Informationsbits eingegeben worden, werden die Schalter in Position 2 gelegt und 5 Paritätsbits werden ausgegeben.

5.3 ARQ

Mit dem ARQ-Schema (*automatic repeat request*) werden diejenigen **payload**-Felder der Pakete geschützt, die eine CRC-Prüfung vorsehen. Der Header wird nicht durch das ARQ-Schema geschützt. Das versenden des Paketes wird wiederholt, bis eine Bestätigung erhalten wurde oder ein *time out* erreicht wurde.

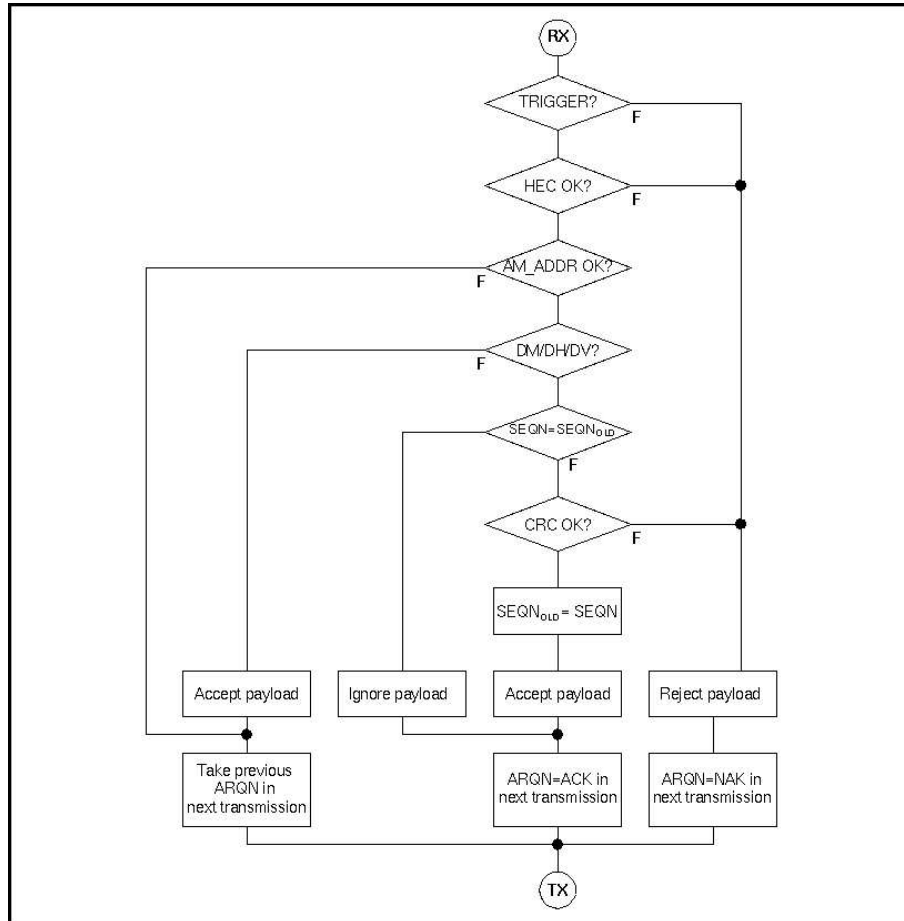


Abbildung 10: So wird das ARQN Bit gesetzt

6 Sicherheitsaspekte

Da Bluetooth eine Funktechnik ist – also theoretisch mit sehr geringem Aufwand abgehört werden kann – ist eine Datenverschlüsselung zwingend erforderlich. Hierzu existieren in Bluetooth vier verschiedene Größen, die bei der Datenverschlüsselung eine Rolle spielen:

- *BD_ADDR Bluetooth device address* mit 48 Bit Länge;
- Privater Authentizierungs-Schlüssel mit 128 Bit Länge;

- Privater Schlüssel variabler Länge mit 8–128 Bit (1-16 Byte) Länge;
- RAND, eine Zufallszahl

Die BD_ADDR ist öffentlich bekannt und wird bei Anfrage oder automatisch verschickt. Sie ist einzigartig für jedes Bluetooth-Gerät – was die Anzahl aller Geräte auf etwa 10^{14} einschränkt... ;) Das sind „nur“ 20.000 Geräte pro Erdbewohner, wenn man von 10 Milliarden Menschen ausgeht...

Der Authentifizierungsschlüssel wird während der Initialisierung erzeugt und ändert sich anschließend nicht mehr. Die Authentifizierung wird immer mit 128 Bit verschlüsselt.

Der private Schlüssel wird bei jeder angeforderten Verschlüsselung neu generiert, deswegen ist seine Lebensdauer ziemlich kurz. Seine Generierung kann – bei Bedarf – von höheren Protokollschichten neu angestoßen werden.

Die Zufallszahl RAND soll eine (pseudo-)zufällige, nichtperiodische Zahl sein. In Praxis bedeutet *zufällig*, dass die Wahrscheinlichkeit, die nächste Stelle richtig vorherzusagen, kleiner als $(\frac{1}{2})^L$, wobei L die Länge der Zahl ist.

7 Service-Discovery Protokoll

Das SDP (*service discovery protocol*) ist ein Dienst, der Bluetooth-Geräte befähigt, sich über verfügbare Dienste in der Umgebung zu informieren.

7.1 Was SDP (nicht) kann

Das SDP erfüllt folgende Voraussetzungen:

- Es muß möglich sein, nach Diensten mit bestimmten Attributen zu suchen.
- Dienste, die zu bestimmten Klassen gehören müssen sich finden lassen.
- Es ist möglich eine Übersicht der Dienste zu bekommen, ohne vorherige Kenntniss der Dienst-Eigenschaften.
- Da es sich um mobile Geräte handelt, muß es möglich sein neue Dienste, die im Sende- / Empfangsbereich auftauchen zu erkennen.
- Ebenso muß man erkennen können, welche Dienste nicht mehr verfügbar sind.
- Dienste, Dienstattribute und -klassen müssen sich eindeutig identifizieren lassen.
- Das erkennen von Diensten muß auf direktem Weg, ohne Umweg über ein drittes Gerät möglich sein.
- SDP muß auch auf Geräten mit geringer Komplexität einsetzbar sein.
- SDP bietet eine Möglichkeit Dienste *inkrementell* abzufragen, um Datenverkehr zwischen Bluetoothgeräten zu minimieren.

Folgende Dienste werden *nicht* vom SDP unterstützt:

- SDP bietet keinen Zugriff auf Dienste, es dient nur der Erkennung.
- SDP handelt keine Verbindungsparameter aus.
- SDP unterstützt keine *automatische* Benachrichtigung, wenn Dienste oder Dienst-Attribute sich ändern oder nicht mehr erreichbar sind.

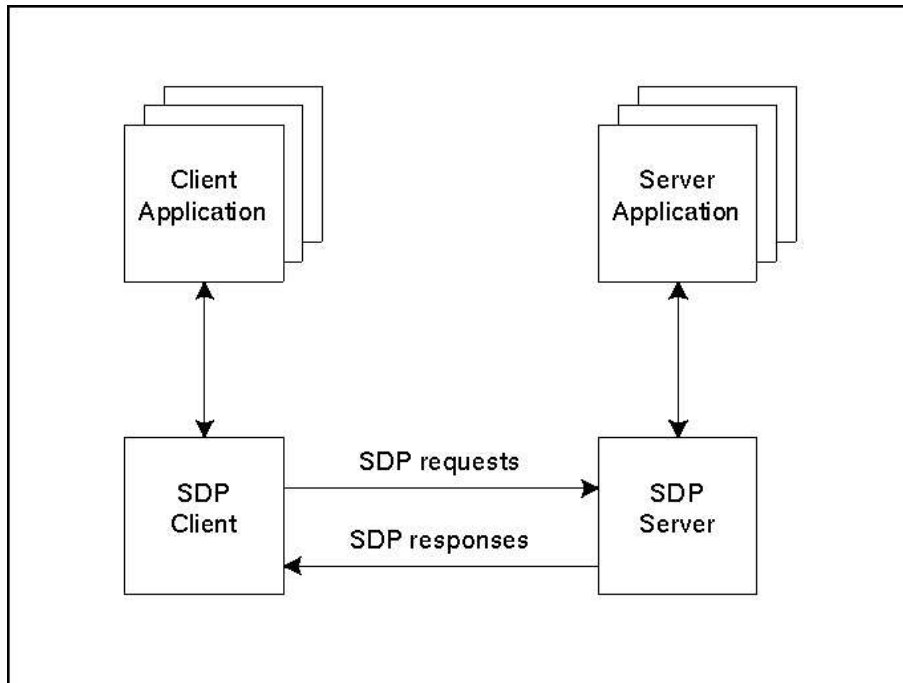


Abbildung 11: Das SDP-Modell

7.2 Wie sehen SDP-Informationen aus

Das SDP kann vereinfacht in diesem Modell dargestellt werden:

Jedes Bluetooth-Gerät kann höchstens einen SDP-Server besitzen, der alle verfügbaren Dienste des Gerätes kennt. Gibt es keine verfügbaren Dienste, so erübrigt sich auch ein Server.

Ebenfalls benutzen alle Anwendungen eines Bluetooth-Gerätes nur einen SDP-Client, um Informationen über verfügbare Dienste zu bekommen.

Das Informationsfeld sieht wie folgt aus:

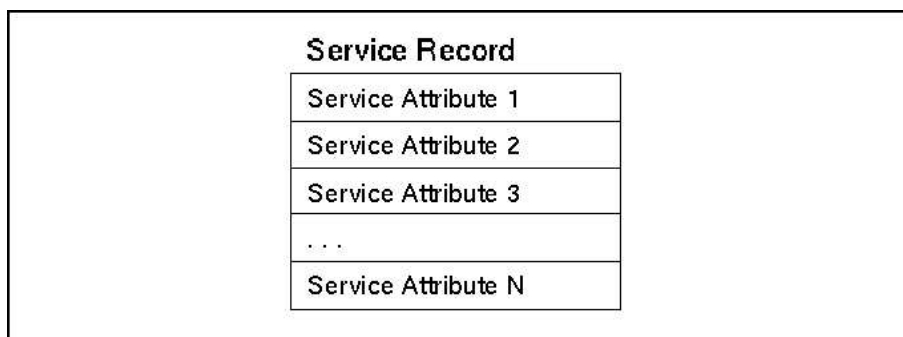


Abbildung 12: Format des *service record* von SDP

Der *service record* mit dem Wert 0x0000 0000 bezeichnet das SDP-Protokoll selbst, die Attribute bezeichnen die Features, die vom SDP-Server und -Client unterstützt werden, z. B. die Protokoll-Version.

Werte zwischen 0x0000 0001 und 0x0000 FFFF sind reserviert.

Service Class ID List	Identifiziert den Typ des Dienstes. Mit anderen Worten handelt es sich um eine Liste der Klassen, von den der Service abgeleitet ist
Service ID	Die spezielle, unverwechselbare Dienst-Instanz
Protocol Descriptor List	Beschreibt die Protokoll-Stacks, die dieser Dienst unterstützt
Provider Name	Name der Dienstanbieter
Icon URL	Definiert eine URL eines Icon-Bildes, das den Service repräsentiert
Service Name	Name des Dienstes im Klartext
Service Description	Eine Beschreibung des Dienstes im Klartext

7.3 Attributbeispiele

Damit man sich mehr unter dem Stichwort *service attribute* vorstellen kann, die einige Beispiele:

8 Bluetooth vs. IrDA

Obwohl Bluetooth und IrDA scheinbar dem gleichen Zweck dienen – dem Kabellosen Datentransfer – ist eine Koexistenz beider Techniken möglich. Beide Techniken haben ihre Vor- und Nachteile. Die Vorteile der einen Technik sind Nachteile der Anderen. So wären bei den Vorteilen der IrDA-Technik zu nennen:

- Lineare Signalausbreitung (*point and shoot*)
- Beschränkter Wirkungsbereich (1 m)
- Hohe Transferrate (4-16Mb/s)
- Bereits jetzt schon weit verbreitet
- Sehr billig zu implementieren

Als Beispiel stelle man sich ein Geschäfts-Meeting von, bei dem die Teilnehmer über PDAs verfügen. Angenommen, man möchte *vor* dem Meeting Visitenkarten austauschen. Für diese Anwendung ist Bluetooth die denkbar schlechteste Wahl. Die Geräte müssen sich gegenseitig finden, die Personen müssen aus einer Liste die Geräte ihrer jeweiligen Gesprächspartner wählen und können erst dann Daten übertragen.

Hier wäre IrDA besser geeignet. Wegen der kurzen Reichweite von 1 m und dem schmalen Winkel von 30° könnten die Personen einfach ihre Geräte aufeinander richten und die Datenübertragung starten. Dies könnte simultan passieren, ohne dass sich die Teilnehmer gegenseitig stören.

Möchte man weiterhin die Präsentation, die im Meeting vorgeführt werden soll, jedem zugänglich machen, so ist IrDA sicher ungeeignet und unbequem. Bluetooth punktet hier eindeutig, wegen der omnidirektionalen Datenübertragung, auch durch Gegenstände hindurch; etwa durch Stühle, Tische oder OH-Projektoren, die im Weg sind.

9 Symbian

Symbian stellt eine große, übergreifende Plattform für mobile Geräte dar. Symbian's EPOC Betriebssystem wurde entsprechend konzipiert, so dass es auf PDA's wie auch auf Mobilfunktelefonen funktioniert.

9.1 EPOC

Symbian unterstützt eine große Palette von Protokollen zur Netzwerk-Kommunikation, wie auch höhere Protokolle für den Datenaustausch.



Abbildung 13: Symbian-Geräte

So werden TCP/IP, PPP, CHAP und PAP im Kernel implementiert. Ebenso sind Protokolle wie IMAP, POP, SMTP und SMS vorhanden.

Das Spektrum wird auf der Sicherheitsseite durch S/MIME, SSL v3.0, DES, 3DES, RSA, DSA und MD5 abgerundet.

Die Symbian-Plattform unterstützt nicht nur für sich kompilierte Software, sondern bringt auch eine *Java Virtual Machine* mit sich, die zu nach eigenen Angaben gänzlich zu JDK 1.1.4 mit AWT kompatibel ist. Die Unterstützung von **JAVA2** soll bald verfügbar sein. Für Darstellung der AWT-Objekte erfolgt durch *EIKON*, EPOCs grafische Oberfläche.

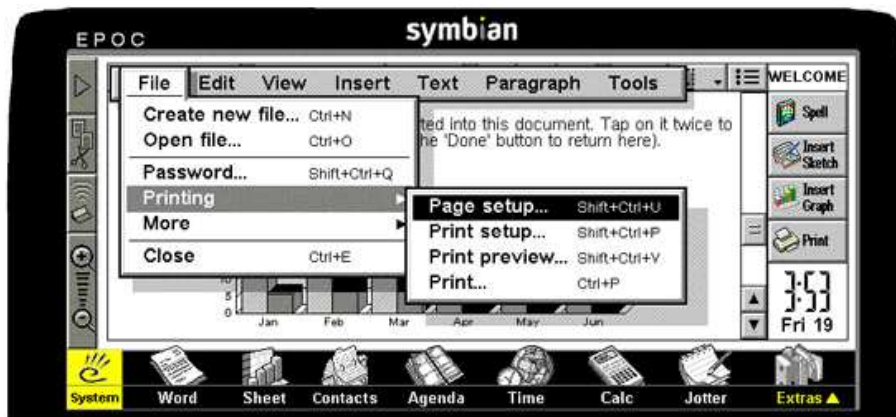


Abbildung 14: EPOCs GUI: **EIKON**

EPOC bringt recht viele Anwendungen mit: Neben den Standard PIM Programmen (Adressbuch, Terminkalender, Weltkarte und Wecker), die man in einem BS für Mobilgeräte erwarten kann:

Dem Besitzer steht ein Office-Paket zur Verfügung, mit Textverarbeitung, Tabellenkalkulation, Datenbank und Notizheft.

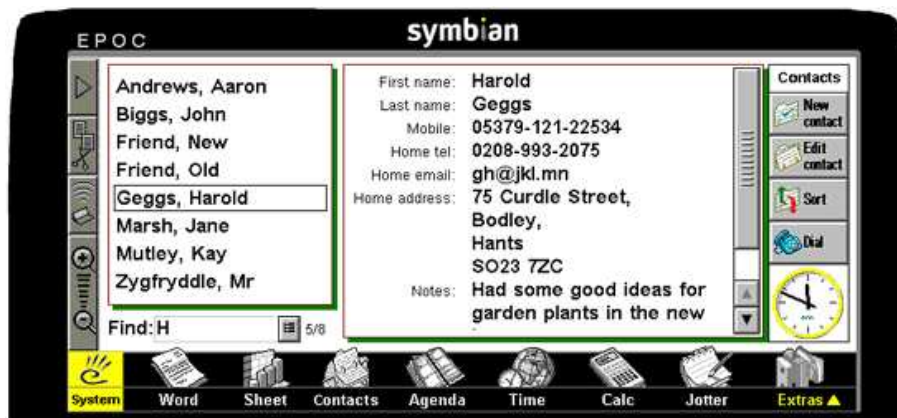


Abbildung 15: Adressverwaltung auf EPOC

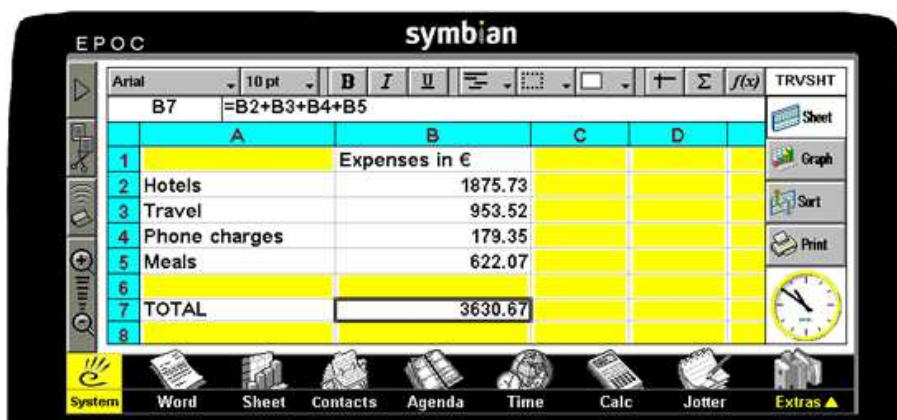


Abbildung 16: Tabellenkalkulation auf EPOC