LMX9830 - Software Users Guide

Texas Instruments
Software Users Guide
February 2007
Revised February 2013
Revision 1.4

Introduction

The Texas Instruments LMX9830 Bluetooth Serial Port module is a highly integrated Bluetooth 1.2 baseband controller and 2.4 GHz radio, combined to form a complete small form factor (6.1 mm x 9.1 mm x 1.2 mm) Bluetooth node.

All hardware and firmware is included to provide a complete solution from antenna through the complete lower and upper layers of the Bluetooth stack, up to the application including the Generic Access Profile (GAP), the Service Discovery Application Profile (SDAP), and the Serial Port Profile (SPP). The module includes a configurable service database to fulfill service requests for additional profiles on the host. Moreover, the LMX9830 is pre-qualified as a Bluetooth Integrated Component. Conformance testing through the Bluetooth qualification program enables a short time to market after system integration by ensuring a high probability of compliance and interoperability. Based on TI's CompactRISC™ 16-bit processor architecture and Digital Smart Radio technology, the LMX9830 is optimized to handle the data and link management

The on-chip memory, ROM and Patch RAM, are optimized for lowest cost and risk, allowing the correction of small firmware errors by the patch mechanism.

The module offers an automatic slave mode without any configuration necessary from an external host. Additionally it offers a command set for hardware configuration and full bluetooth operation over SPP.

This document is a reference for implementing the LMX9830 module into a system. A getting started session gives a very detailed entry point for starting development. The advance usage section describes all features and configuration parameters in detail and gives example for using the LMX9830 as active bluetooth node. Finally all commands and events are listed and explained in the command section.

IMPORTANT:
Please also always check for Release Notes for possible errors and software workarounds for the firmware.

This document is based on:

Table 0-1.

Item	Version
Hardware	LMX9830
Firmware	V0106 or later
Actual Firmware Release	V0212
SimplyBlueCommander	1.6.0.1

Table of Contents

1.0 LMX9830 General Setup

1.1 SYSTEM OVERVIEW

The LMX9830 is intended to be an add-on module to an existing microcontroller. In this function it either appears as cable like interface for the UART or can also be controlled with a simple application on the external microcontroller to establish links itself.

The LMX9830 includes the complete bluetooth stack including the following protocol layers.

- Link Controller
- Link Manager
- L2CAP (Logic Link Control and Adaptation)
- RFCOMM
- SDP (Service Discovery Protocol)

An on-chip application together with those protocol layers offers the following profiles:

- GAP (Generic Application Profile)
- SDAP (Service Discovery Application Profile)
- SPP (Serial Port Profile)

The application manages all profile related interactions to the stack but also offers a simplified command interface over the UART. The interface is used for configuring the device, setting up the link and receiving events from the module.
The interface can handle either packaged data transmission for multipoint support or is able to handle RAW data by setting it into a transparent UART mode.

The firmware can also accept or establish synchronous links (SCO) to transmit audio data. Once the link has been established the firmware routes the synchronous data to and from the PCM interface, using predefined driver settings. No further host action is required.

.

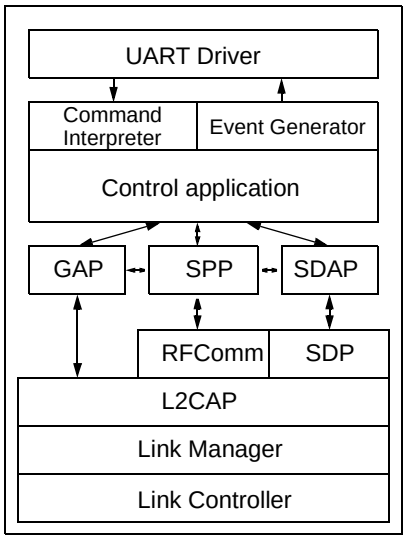


Figure 1-1. LMX9830 Firmware Implementation

1.1.1 Hardware setup

Figure 1-2 shows a block diagram for the LMX9830. The block diagram shows all functional blocks connected to the device.

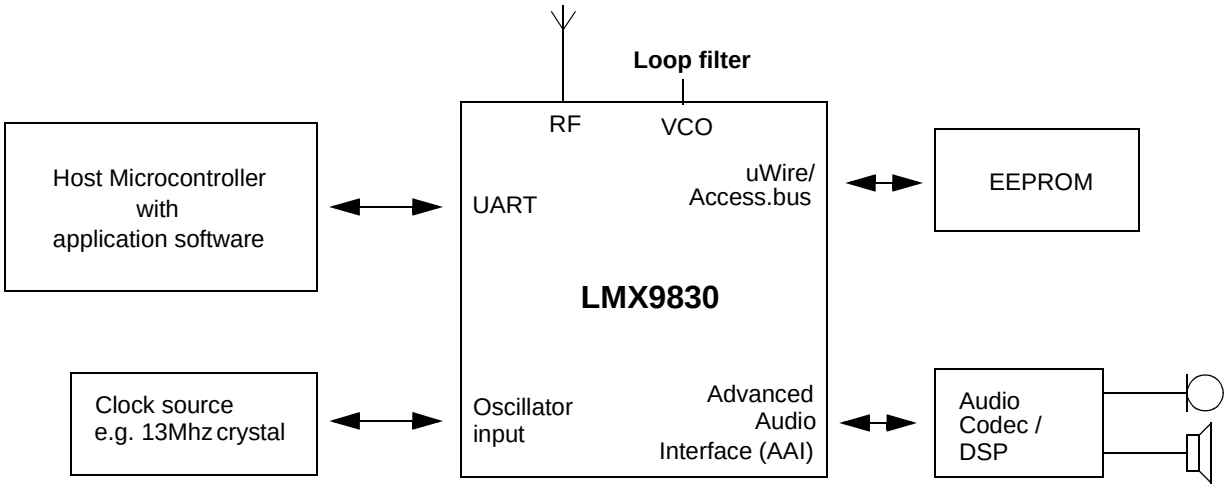


Figure 1-2. LMX9830 main interface blocks

1.1.1.1 Host Microcontroller over UART

The host microcontroller can be any microcontroller, sending data or commands to the LMX9830 over the UART interface. In case the LMX9830 is used to actively establish outgoing links, the host controller needs to send the appropriate commands over the UART interface.

The LMX9830 offers bluetooth operation up to the Serial Port Profile (SPP), which is the basis for many other profiles like DUN or Headset. In case such profiles shall be supported by the end product, the additional profile needs to be implemented on the host application, which uses the LMX9830 as kind of “SPP gateway”. See also Section 5.0 "Profile Support" on page 79 how to support additional profiles.

1.1.1.2 System Parameters in Non-volatile Storage (NVS)

Since the LMX9830 is ROM based, all parameters used during a session will be stored in RAM and would be lost after power-down. For bluetooth operation as well as for internal configuration the LMX9830 has a number of "System Parameters" (1.2), which contain parameters mandatory for bluetooth operation (e.g. Bluetooth Device Address (BD_Addr)) as well as configuration values for the LMX9830. See Section 1.2 "System Parameters" on page 10 for a complete list of parameters.

1.1.1.2.1 Operation with EEPROM

To avoid reconfiguration of the LMX9830 on each power-up, all"System Parameters" (1.2) can be stored within an external EEPROM. The LMX9830 will automatically store changed parameters within the EEPROM and will restore them after boot-up and reset.

The first programming of the EEPROM (in case it is detected as “empty”) is automatically done by the LMX9830 by using the "Restore Factory Settings" (6.2.18.5) function, which automatically sets or resets all values to operation default. This command usually is only used once for initialization, but can also be used anytime to reset the EEPROM to a known good state.

1.1.1.2.2 Operation without EEPROM

The LMX9830 can also be used without external EEPROM. In this case the LMX9830 on power-up will run into an “Initialization Mode”, in which the host needs to provide mandatory parameters before entering the bluetooth operation by using the "Enter Bluetooth Mode" (6.2.20.2) command.

Once the device is in "Bluetooth Mode" (2.2), the LMX9830 will operate from the “RAM System Parameters” (also called “RAM NVS”). The parameters can even be addressed by writing with the same commands. There’s no difference on the interface.

The only difference appears on a software reset, which resets the Service Database to the default (one SPP service, see Section 2.3.3 on page 29). All other values (e.g. BD_Addr or clock settings) are unchanged/restored.

1.1.1.3 Clock Source

The clock source usually is done by either a 12 or 13Mhz crystal or by using an external clock source. The LMX9830 is able to run from various external clock speeds. Please check also [1] "Texas Instruments: LMX9830 Datasheet" for a list of supported frequencies.

In case a speed different to 12 or 13Mhz is used, the clock value needs to be stored as "System Parameters" (1.2).

1.1.1.4 Audio Codec / DSP

As mentioned before, the LMX9830 can be used to transport control data required for additional profiles on the host. With this it is possible to use the device for audio profiles like Headset Profile (HSP) or Handsfree Profile (HFP). Audio data to be transported over the Bluetooth link will be handled through the Advanced Audio Interface (AAI), without any interaction required from the host.

The LMX9830 audio interface supports PCM master or slave operation. Master operation is used with typical codecs like the OKI or Motorola codecs while PCM slave allows the flexible combination with DSPs or other host controllers. See also Section 4.5 "Establish Audio Links" on page 69.

Please check [1] "Texas Instruments: LMX9830 Datasheet" for details on which codecs are supported by LMX9830.

1.2 SYSTEM PARAMETERS

The LMX9830 is controlled by system parameters. Since these parameters are usually stored in a non-volatile memory like the EEPROM, they are also called “NVS parameters”. These parameters define the chip behavior during bluetooth operation but also after a software or hardware reset.

Since the LMX9830 is ROM based, these working parameters are stored in a dedicated RAM area and set to factory default after power cycle or hard reset. In case an EEPROM is connected, these parameters and possible patches are loaded into RAM before entering bluetooth mode.

In case no EEPROM is connected, the firmware will enter the “Initialization Mode”, which requires the programming of minimum required parameters (BD_ADDR). Please see also Section 2.1 "Initialization Mode" on page 18 for the Boot-up sequence.

Table 1-1 lists the memory map of the System parameters. The address reflects the memory address of the optionally connected EEPROM.

Table 1-1. LMX9830 System Parameters, EEPROM Memory Map

No.	Address	Parameter	Default Value	Description	SW Reset required
1	0000-0005	BD_Addr	<empty>	Bluetooth Device Address LAP(lsb), LAP, LAP, UAP, NAP, NAP (msb) Required for Bluetooth mode	no
2	0006	NVS Initialized	0x00	Indicates whether the eeprom has been initialized or not. 0x00: Initialized 0xFF Not initialized.	yes
3	0007	Unit-KeyPresent	0xFF	Used by BT core, generated during pairing procedure.	no
4	0008-0017	UnitKey	0xFF..0xFF	Used by BT core, generated during pairing procedure.	no
5	0018	Device-Name-Length	0xFF	Length of Parameter 6 “Devicename”	no
6	0019-0040	Device-Name	0xFF...0xFF	Friendly Name of the Bluetooth Device	no
7	0041	Country-Code	0x00	Used by BT core	yes
8	0042	PinLength	0x04	The length of parameter 9, “PinCode”. In case set to 0, the LMX9830 will request pin from host.	no
9	0043-0052	PinCode	“0000”	Fixed PinCode used for pairing with other devices	no

Table 1-1. LMX9830 System Parameters, EEPROM Memory Map

No.	Address	Parameter	Default Value	Description	SW Reset required
10	0053-0055	ClassOfDevice	0x000000	The 'Class of Device' describes general functionality of the Bluetooth Device and is transmitted during the Inquiry process.	no
11	0056	SppPortsToOpen	0x00000001	Bitmask defining the RFCOMM channels to open. For each channel one RFCOMM instance will be created.	no
12	005A	PreferredMasterRole	0x00	Preferred Master forces the device to switch to Master Role after being connected. The device will reject the link if command could not be executed.	yes
13	005B	Automatic Operation	0x01	Configures the general behavior of the device. Please see Section 2.2 "Bluetooth Mode" for details. 0x00: Automatic OFF (Non-automatic) 0x01: Automatic ON (Automatic)	yes
14	005C	PageScanMode	0x01	Configures the connectability of the device 0x00: not connectable 0x01: normal scan 0x81: interlaced scan (faster connection time)	no
15	005D	InquiryScanMode	0x01	Configures the discoverability of the device 0x00: not discoverable 0x01: normal scan 0x81: interlaced scan (faster response time)	no
16	005E	SecurityMode	0x02	Configures Service Level Security Mode.	no
17	005F-0060	DefaultLinkPolicy	0x000F	Configures the default link policy for incoming links.	no
18	0061	EventFilter	0x01	Configures the level of events reported to the host. 0x00: No filter, all events reported 0x01: ACL events filtered, only API events reported. 0x02: All events filtered, only UART breaks indicated 0x03: All events filtered, including UART break.	no
19	0062	PMM / GPIO usage	0xFF	Bitmask to configure enhanced power management (PMM) functions as well as the usage of dedicated GPIO pins 0x01: enhanced PMM: 1 - disabled (default); 0 - enabled (requires 32khz crystal connected) 0x02: Use PG6 to signal SPP activity (Links) 0x04: Use PG7 to signal TL activity All other bits are reserved and should be set to 1	yes
20	0063-0064	LinkTimeout	0x7D00	Configures the default link supervision timeout (in slots, 0.625ms) used for incoming and outgoing links.	no

Table 1-1. LMX9830 System Parameters, EEPROM Memory Map

No.	Address	Parameter	Default Value	Description	SW Reset required
21	0065	CodecType	0x00	The audio codec settings used on the PCM interface: 0x00: None connected 0x01: Motorola MC145483 0x02: OKI MSM7717 0x03: PCM slave, see PCMSlaveConfig 0x04-0xFF: reserved	yes
22	0066	AirFormat	0x00	The audio format used on the SCO link 0x00: CVSD 0x01: u-Law 0x02: A-Law 0x03-0xFF: reserved	yes
23	0067-0068	RfcommLatency	0x0000	Configures the default poll period of master to slave. 0x0000: No requirement (default 40slots) 0x0002-0x0190: Valid link latency	no
24	0069-006C	Frequency	0x00000000	The crystal frequency in Hz. The frequency parameter is only needed when the firmware start up in a mode with unknown crystal frequency (10-20MHz). OP pins are used to determine if the crystal frequency is unknown	yes
25	006d	UartParityBit	0x00	Parity setting for the hardware UART interface. 0x00: No Parity 0x01: Even Parity 0x02: Odd Parity	yes
26	006E	UartStopBit	0x00	Stop bit settings for the hardware UART interface 0x00: 1 Stop bit 0x01: 2 Stop bits	yes
27	006F	UartSpeed	0x03	Speed of the Hardware UART interface. This settings are only used, in case the FSEL pins are configured to “Autobaudrate detect” 2400: 0x00 4800: 0x01 7200: 0x02 9600: 0x03 19200: 0x04 38400: 0x05 57600: 0x06 115200: 0x07 230400: 0x08 460800: 0x09 921600: 0x0A	yes
28	0070-00AE	RemoteDevices	0x00..0x00	Default connections database, to be connected during boot-up or by sending a command.	no
29	00AF	VtuneDesiredThreshold	0xFF	NSC configured parameter, do not modify	yes
30	00B0	VtuneCn	0xFF	NSC configured parameter, do not modify	yes
31	00B1	VtuneEnable	0xFF	NSC configured parameter, do not modify	yes

Table 1-1. LMX9830 System Parameters, EEPROM Memory Map

No.	Address	Parameter	Default Value	Description	SW Reset required
32	00B2-00B3	PcmSlave-Config	0xFFFF	This 16-bit value (LSB first) is used to store the PCM format configuration for the PCM slave configuration. This setting is only use in case the PCM slave setting is activated (see No. 21). BIT0-1, Slot selection 00: use slot 0 01: use slot 1 10: use slot 2 11: use slot 3 BIT2-3: Number of slots per frame 00: 1 slot 01: 2 slots 10: 3 slots 11: 4 slots BIT4-6: PCM data format 000: Reserved 001: 8 bit A-law 010: 8 bit u-law 011: 13 bit linear 100: 14 bit linear 101: 15 bit linear 110: 16 bit linear 111: Reserved BIT7:Frame sync length 0: short frame sync 1: long frame sync BIT8: Data word length 0: 8-bit data word length 1: 16-bit data word length BIT9: Frame sync polarity 0: use inverted frame sync 1: use normal frame sync BIT10-15: Unused, set to 0	yes
33	00B4	PcmFcprs	0xFF	Unsigned integer indicating the frame clock prescaler for generic PCM slave	yes
34	00B5-00B8	RfSetupReg4	0xFFFF	NSC configured parameter, do not modify	yes
35	00B9-00BC	RfSetupReg15	0xFFFF	NSC configured parameter, do not modify	yes
36	00BD-00EE	Filler 1	0xFF ... 0xFF	Filler (not used)	
37	00EF-011E	Service-Records	0xFF..0xFF	SDP/Security info storage	no
38	011F-0346	CoreNvs-LinkKeys	0xFF..0xFF	Link key storage (24 keys)	no
39	0347-0355	AssertInfo	0xFF..0xFF	NSC configured parameter, do not modify	
40	0356-0367	RunError-Info	0xFF..0xFF	NSC configured parameter, do not modify	
41	0368-037F	Filler2	0xFF..0xFF	Filler, not used	
42	0380-137F	SdpRecords	0xFF..0xFF	SDP record storage	no

Table 1-1. LMX9830 System Parameters, EEPROM Memory Map

No.	Address	Parameter	Default Value	Description	SW Reset required
43	1380-1FFF	PatchCode	0xFF..0xFF	ROM patch code storage	yes

1.3 UART COMMUNICATION

The main communication interface between the LMX9830 and the host is the UART Interface.

The UART interface between host and LMX9830 needs to be connected in Null Modem configuration, meaning RTS/CTS and TX/RX are crossed.

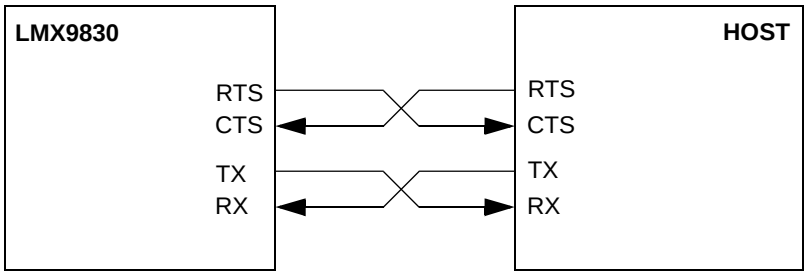


Figure 1-3. UART Null Modem connection

The command interface is based on a command/event based structure. Each command (also called “Request”) will be acknowledged with the appropriate status event (also called “Confirm”). Unexpected events, like incoming link establishment or data, are also sent as events, but signed with a different simply blue package type called “Indicator”. Please see Section 6.0 "LMX9830 Command Interface" on page 117 for the detailed description of the command interface.

Since the LMX9830 acts like a gateway between the Bluetooth Link and the UART interface, the UART connection should use 4-wire hardware handshaking for optimal buffer handling. The LMX9830 uses the RTS lines to indicate low buffers and reacts on the CTS from the host, immediately stopping sending packages to the host.

In case the host microcontroller is not able to provide hardware handshaking, the UART should be used in "Command Mode" (1.3.2.1) only, since the device would not be able to indicate full buffers by using the RTS signal. In the same way, the host needs to provide enough buffer space to be able to handle the incoming data, since it’s not able to stop the flow from the LMX9830.

1.3.1 UART configurations

The UART interface consists of four signals:

- TX: Transmit output
- RX: Receive input
- RTS: Ready-to-Send output, indicating the host “i am ready to receive data”
- CTS: Clear-to-Send input, allows the host to stop the transmission from the LMX9830 to the host.

The LMX9830 will ALWAYS use the RTS to indicate to the host, that it is ready to receive data and it will ALWAYS sample the CTS input to check, if the host is able to receive data. Therefore, it has to be made sure that the CTS pin is pulled low in case the host is ready. Otherwise the LMX9830 will not start sending out data or events.

However, some applications don’t have the ability to use the RTS/CTS handshaking. The following two chapters show the differences between using full 4-wire configuration or 2-wire configuration.

1.3.1.1 UART 4-wire connection for hardware handshaking

The 4-wire configuration is done as shown in Figure 1-3, which is the recommended setting. This configuration allows full operation in "Command Mode" (1.3.2.1) and "Transparent Mode" (1.3.2.2).

The handshake functionality is based on the RTS / CTS signalling, which is used in both modes. The LMX9830 indicates with its RTS signal (RTS=low), that it is able to receive data and will high it low in case the TX buffers are full.
This indicator can get very important in "Transparent Mode" (1.3.2.2) in case the host transmits a lot of data over the LMX9830. In case the remote bluetooth device runs into a buffer problem or the bluetooth link is slow, the LMX9830 will toggle the RTS as soon as its buffers are filled up.

The incoming CTS input can be used by the host to stop the flow from the LMX9830 to the host. This might get important in case the host processor is a slow performing device, not able to process the incoming data in sufficient time.

NOTE: The CTS signal is the only way to stop the flow from the LMX9830 to the host.

1.3.1.2 UART 2-wire connection

In general, the LMX9830 does not absolutely require the RTS / CTS signals to be connected to the host. However, since the LMX9830 always assumes the signals to be used, the following issues have to be addressed or are recommended:

- CTS to GND:
  The LMX9830 uses the CTS input to get the permission to send out data. If this signal is not pulled to GND, the device will not start sending data.
- For TX, use "Command Mode" (1.3.2.1) only:
  As described, the LMX9830 will indicate buffer issues by pulling RTS to high. Since the host does not check this pin it is recommended to use "Command Mode" (1.3.2.1) only. In this mode, data have to be sent using the "SPP Send Data" (6.2.3.3) command. Each command is confirmed with a specific event including the command status. The LMX9830 will indicate the successful command processing with status 0x00. This can be seen as kind of software handshaking.
- **DO NOT USE "Transparent Mode" (1.3.2.2) FOR TRANSMITTING DATA IN THIS CONFIGURATION.**
  Using Transparent Mode without hardware handshake, it can not be guaranteed that data are not overwritten within the LMX9830. Use "Command Mode" (1.3.2.1) instead.
- For RX, host needs to be fast enough:
  Since the CTS signal is not used and fixed to GND, the host will not be able to tell the LMX9830 to stop the flow, in case he's running out of buffers. Therefore, only use 2-wire connection if the host is capable to process the incoming data in appropriate time.

1.3.2 UART Modes

1.3.2.1 Command Mode

The LMX9830 offers a wide range of commands to configure the hardware and the bluetooth operation. As the command set is on top of the profiles, Bluetooth operational commands are reduced to high level commands controlling general bluetooth operation.

In Command Mode, the LMX9830 will try to interpret all data sent over the UART to a known command. The commands have to be sent in a specific package format. The interface is based on an event mechanism. Any command sent will be confirmed by the appropriate confirmation event. Unexpected events (e.g. incoming links) will be reported by indication events.

Please refer to Section 6.0 “LMX9830 Command Interface” on page 117 for a complete list of commands and their usage.

1.3.2.2 Transparent Mode

In case the LMX9830 has established a link to only one remote device and no configuration commands have to be sent to the LMX9830 ("Command Mode" (1.3.2.1)), the UART interface can be switched to “Transparent mode”.
This means data are directly routed to the bluetooth link and not interpreted. Also incoming data are not indicated as events, they are sent as RAW data to the UART.

.

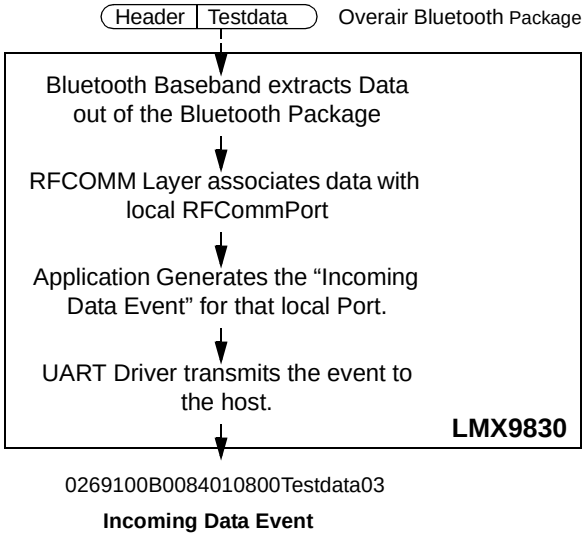


Figure 1-4. Receiving data in command mode

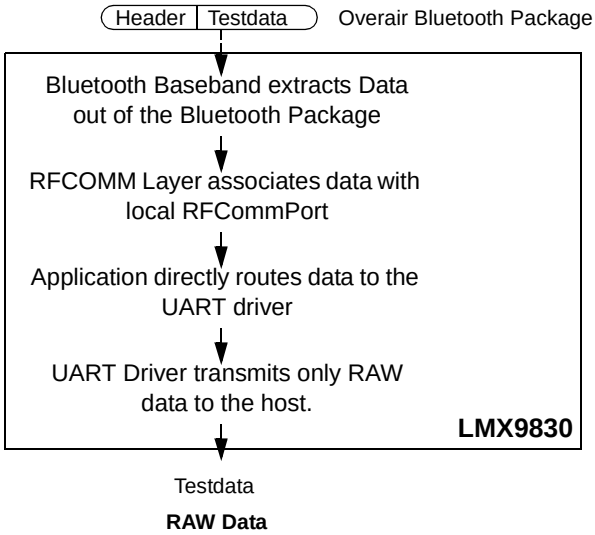


Figure 1-5. Receiving Data in Transparent Mode

Leaving transparent mode:

As the LMX9830 does not listen to commands, UART Break has to be used to tell the device to leave the transparent mode. See also Section 6.2.4.1 “Transparent Mode” on page 135 for details.

1.4 AUDIO SUPPORT

The LMX9830 offers commands allowing to establish or to accept synchronous audio links. The audio data will be routed within the LMX9830 between the bluetooth baseband connection and the Advanced Audio interface. In order to encode or decode the PCM stream generated by the internal CVSD codec, an external codec or DSP has to be connected to the advanced audio interface. Alternatively the data can be used to be for further digital processing.

The AAI is configured by predefined codec settings, selected in NVS. The NVS setting configures the bit rate as well as the format like PCM log or linear. In addition, the interface can be configured to “PCM Slave” in which the external codec or DSP needs to provide the synchronous clock and frame sync signal. The settings for the PCM slave mode are done by specific commands. See also Section 4.5.3.1 "PCM codec configuration" on page 70 for detailed instructions on codec con-

figuration.

The Bluetooth standard defines CVSD, u-Law and A-Law to be used as format on the bluetooth link. This settings is also configured within the NVS.

Figure 1-6 shows a typical application block diagram, how to connect the audio codec.

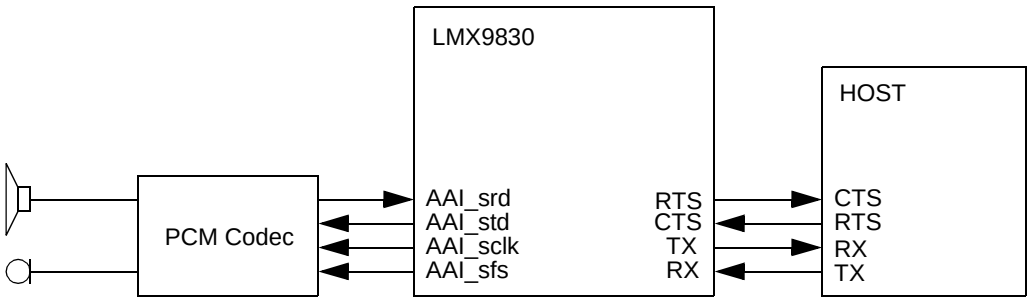


Figure 1-6. PCM Codec connection block diagram

2.0 Initialization and Bluetooth operation

The LMX9830 operation can be split into 2 modes, initialization mode and bluetooth mode. After boot or reset, the LMX9830 starts into the initialization mode, which checks for necessary operation parameters. If all parameters are found in EEPROM or have been submitted from the host (over UART), the device will enter bluetooth mode. In case an EEPROM is used, initialization mode usually only gets visible on first boot-up if some parameters are missing. Afterwards the device will enter bluetooth mode automatically.

2.1 INITIALIZATION MODE

The LMX9830 follows a defined startup procedure, in order to check the availability of minimum required parameters for bluetooth operation. The device will stay in the so-called “Initialization Mode” until the following parameters are defined.

- External Clock Speed
- UART Baud rate
- Bluetooth Device Address (BD_Addr)

Clock and Baud rate can either be defined by the pins OP3, OP4 and OP5 or can be handled through NVS settings. In case one of the parameters is missing, the LMX9830 sends the "Await Initialization Event" (6.2.20.1). In case either clock or baud rate are not defined, the host needs to initiate the UART auto baud rate procedure, sending character 0x01.

If all parameters are defined, the initialization mode needs to be ended by the "Enter Bluetooth Mode" (6.2.20.2) command. After this the device is in bluetooth operation.

The initialization mode will not be visible during Reset of boot-up if all parameters are set correctly in EEPROM or RAM.

Figure 2-7 on page 19 shows the generic initialization flow after boot-up or hardware reset. The initialization mode is similar for EEPROM and No-EEPROM operation. Please see also Section 2.3 "Operation without EEPROM" on page 28 for details, on the specific reset behavior with and without EEPROM.

Figure 2-8 on page 20 shows the general Initialization communication on the UART interface between host and LMX9830.

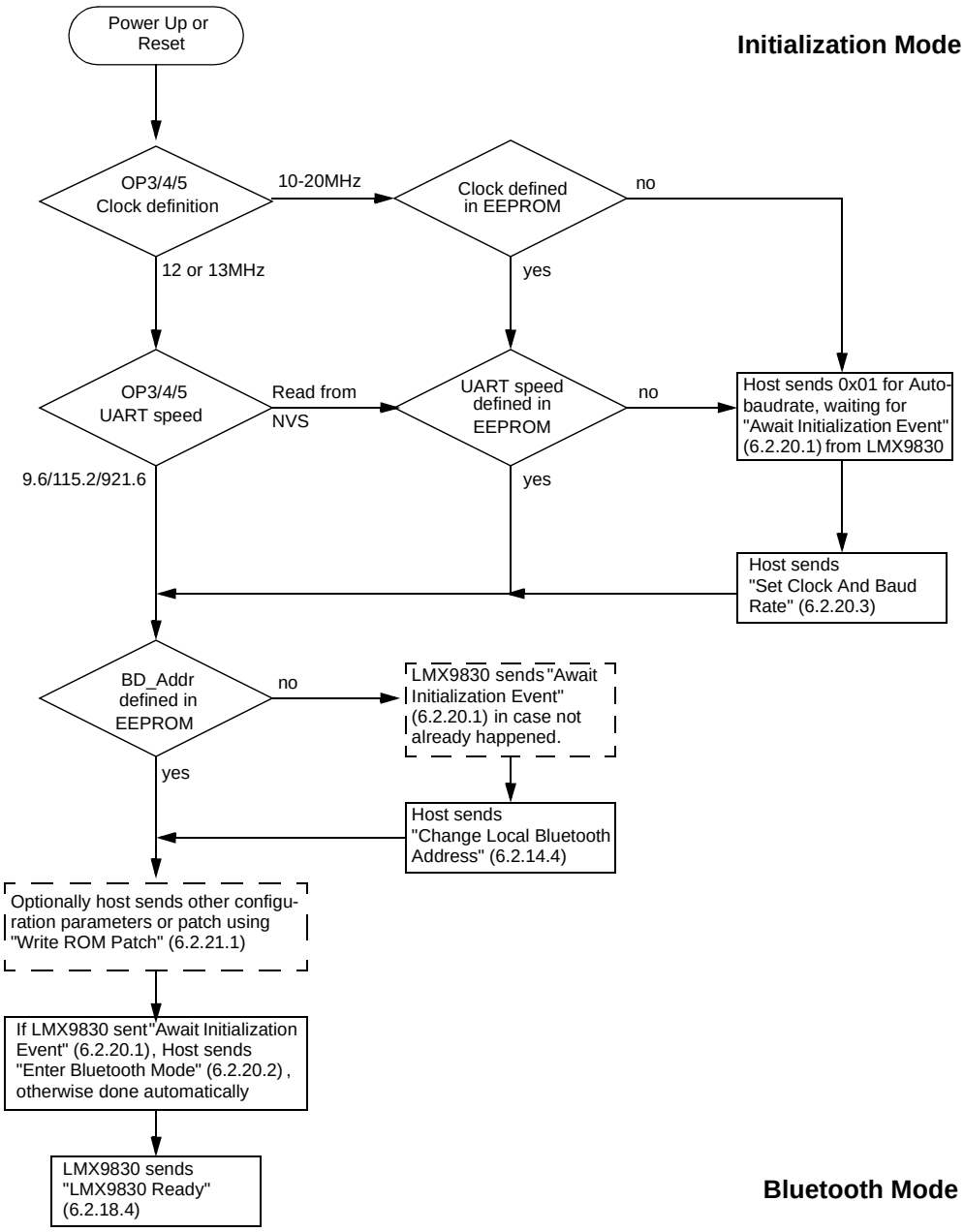


Figure 2-7. LMX9830 Initialization

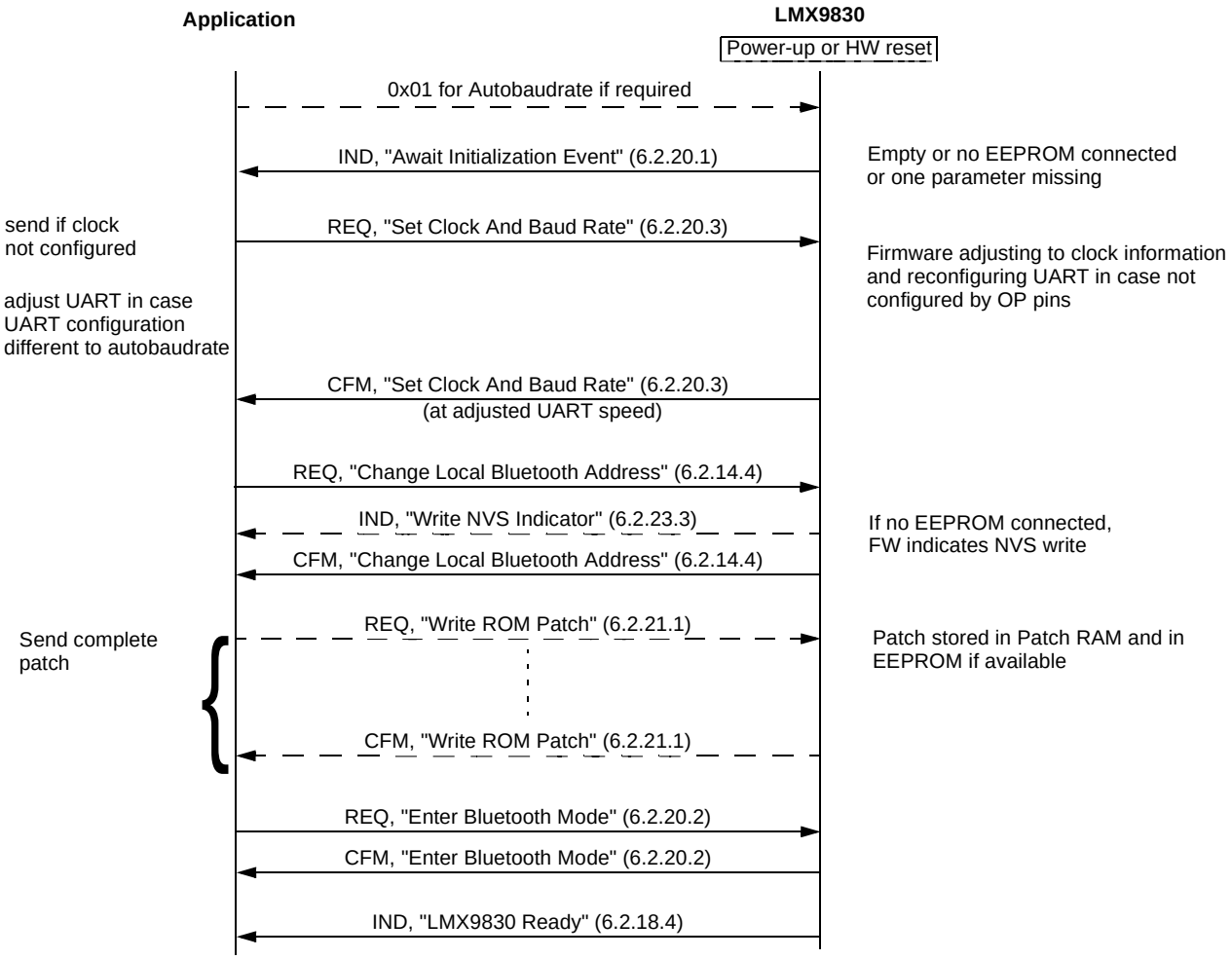


Figure 2-8. General initialization sequence

2.2 BLUETOOTH MODE

The operation of the LMX9830 can be divided into different states. Each state represents special situations and describes the behavior of the module.

The operation states are influenced by a few features, stored in the NVS (RAM and EEPROM). Depending on those, the device will come up to different modes and act according to those settings:

2.2.1 NVS Settings influencing the operation state

Before actually describing the different states, we should have a look on the parameters, which influence the behavior of the LMX9830. The following parameters are stored in NVS and are checked during boot-up.

2.2.1.1 Automatic operation on/off

The Automatic operation flag is checked after Reset/Boot-up and on incoming links. According to this flag the firmware will initiate automatic steps.

Automatic Operation On:

- Connect to "Default connections stored in NVS" (2.2.1.3)
- Switch UART to "Transparent Mode" (1.3.2.2) after first incoming link ("Transparent Slave" (2.2.2.7))

Automatic Operation Off:

- No automatic steps after Reset/Boot-up

2.2.1.2 Force Master on/off

This flag improves multipoint operation of the LMX9830. In normal operation, the LMX9830 will be slave for any incoming link, which limits the number of incoming connects to 2 devices (see "Scatternet Slave" (2.2.2.6)).

In case the Force Master Bit in the NVS is set to On, the LMX9830 will request a role switch on any incoming link to become Master.

Force Master On:

- Request Role Switch on incoming link to become Bluetooth Master. Link will be dropped if role switch not successful.

Force Master Off:

- Accept incoming link as slave

2.2.1.3 Default connections stored in NVS

In case the automatic operation bit (see Section 2.2.1.1) is set to On, the default connections database is checked for any valid entry. If a valid entry is found, the LMX9830 tries three times to connect to each device stored in the database.

Default connection transparent On/Off

Each default connection can be configured to switch the UART to"Transparent Mode" (1.3.2.2) or to "Command Mode" (1.3.2.1) after link establishment. In case the transparent flag is set, the LMX9830 will switch to "Transparent Mode" (1.3.2.2) once the link is established.

NOTE: The transparent flag can only be activated if only one default connection is stored in the database

2.2.2 Operation States

As combination out of the different parameters and usage scenarios, the following states can be defined for the LMX9830. The parameters and possibilities for this states are summarized in Table 2-2 "Overview of Operation States" on page 22.

How to read Table 2-2 on page 22: E.g. Scatternet Master (assuming default NVS settings)

In Scatternet Master, the device is in command mode (listening to commands).
The automatic flag in NVS has no influence. The device is discoverable and connectable for other devices.
The device is Master for x slaves and slave to 1 Master.
It is possible to search for other devices but it is not possible to be connected from another device. RAW data traffic is not possible.
No Incoming link possible.
In case the command interface is used to establish a link to another device, the device stays in "Scatternet Master" (2.2.2.3).
A UART BREAK has no influence on the functionality. Sending the command "Transparent Mode" (6.2.4.1) is not applicable for this mode. A SCO link can be established to one of the remote devices.

Table 2-2. Overview of Operation States

Parameter / State	Idle Automatic	Idle Non-Automatic	Piconet Master	Scatternet Master	Transparent Master	Single Slave	Scatternet Slave	Transparent Slave
UART Mode	Command	Command	Command	Command	Transparent	Command	Command	Transparent
Automatic Operation	0x01 (On)	0x00 (Off)	-	-	-	-	-	-
Discoverable[1]	yes	yes	yes	yes	no	yes	no	no
Connectable[2]	yes	yes	yes	no	no	yes	no	no
Bluetooth Role	-	-	Master to x Slaves[3]	Master to x Slaves[4] Slave to 1 Master	Master to 1 Slave	Slave to 1 Master	Slave to 2 Masters	Slave to 1 Master
Possible to search for devices (Inquiry)	yes	yes	yes	yes	no	yes	no	no
Connect to remote devices (actively)	yes	yes	yes[5]	yes[4]	no	yes[4]	no	no
Send Raw Data	-	-	no	no	yes	no	no	yes
State after incoming link	Transparent Slave	Single Slave	Scatternet Master	-	-	Scatternet Slave	-	-
State after outgoing link	Piconet Master	Piconet Master	Piconet Master	Scatternet Master	-	Scatternet Master	-	-
State after sending "Transparent Mode" (6.2.4.1)[6]	-	-	Transparent Master	-	-	Transparent Slave	-	-
State after UART BREAK	-	-	-	-	Piconet Master	-	-	Single Slave
SCO Link possible	-	-	yes	yes	yes	yes	no	yes

1. Depending on parameter #14 in the non-volatile storage (default setting assumed), see Table 1-1 on page 10
2. Depending on parameter #15 in the non-volatile storage (default setting assumed) see Table 1-1 on page 10
3. With a value of $1 \leq x \leq 7$
4. With a value of $1 \leq x \leq 7$
5. Maximum number of slaves is 7
6. Transparent Mode can only be activated, if one and only one SPP link is active.

2.2.2.1 Idle, Automatic and Idle, Non-automatic

After boot-up, reset or after successfully releasing the last link, the LMX9830 stays within one of the Idle states.
In case the Automatic Operation bit is set to “On”, it is also called “Idle, Automatic”. Otherwise, the state is called “Idle, Non-Automatic”

In both modes, the device is able to accept incoming links or the application can use the command interface to actively inquire or establish a link.

The difference between Automatic “On” and “Off” just shows up at an incoming connection:

2.2.2.1.1 Incoming Link at Idle Automatic

In case, the LMX9830 is not connected to any other device and gets an incoming link request, it will

- ask for authentication or pin code exchange
- accept the link
- notify the application by an indicator
- turn off scanning (disabling page scan and inquiry scan)
- switch UART to "Transparent Mode" (1.3.2.2))

“Idle Automatic” is optimized for cable replacement applications, not requiring any interaction with the LMX9830 to accept an incoming link. As the LMX9830 automatically switches to transparent mode, the application can start sending Raw data immediately after receiving the notification.

2.2.2.1.2 Incoming Link in Idle Non-Automatic

In case, the LMX9830 is not connected to any other device and gets an incoming link request, it will

- ask for authentication or pin code exchange
- accept the link
- notify the application by an indicator
- change state to "Single Slave" (2.2.2.5)

After link establishment, the command interface is still active, delivering incoming data using the "Incoming Data Indicator" (6.2.3.4). To send data the command "SPP Send Data" (6.2.3.3) needs to be used.

“Idle Non-automatic” is optimized for multi-profile applications which need to manage multiple links or different profiles at the same time. As the LMX9830 stays in command mode, the application still has full control over the LMX9830, to establish links or do configurations.

In case "Transparent Mode" (1.3.2.2) would be beneficial for some data transfer, it can be reached anytime by sending the "Transparent Mode" (6.2.4.1) command.

2.2.2.2 Piconet Master

In Piconet Master, the LMX9830 is in a link with one or more devices as Bluetooth Master. The bluetooth specification describes the Master as the controlling device for the piconet. The Master defines the hopping sequence and manages the connection to each slave.

In general, a bluetooth device will be master of the link, when it initiated the link (using the paging procedure). A device accepting an incoming link is called slave. However, every bluetooth device is able to request to a change of role (also called “role switch”) during link setup, therefore this general rule may not apply in 100% of the cases.

The LMX9830 can be assumed as master of the bluetooth link and to be in "Piconet Master" when

- the command interface has been used to actively establish one or more links by using one of the following commands
  - "Establish Link" (6.2.3.1)
  - "Connect to Default Connection" (6.2.7.2)
  - a default connection has been established after power-up or "Reset" (6.2.18.3)
- the LMX9830 accepted an incoming link while the Force Master bit in NVS was set to 0x01 ("Force Master Role" (6.2.14.8))

After successfully establishing a link, the LMX9830 will stay in "Command Mode" (1.3.2.1).

The benefit of being master instead of slave is, that the LMX9830 is actively managing the link to all devices, so can assign each device the bandwidth it requires. With this, the LMX9830 is able to support up to 7 active links.

In Piconet Master, the LMX9830 is able to handle one SCO link to one connected slave.

NOTE: The maximum number of links is limited by the available RAM, which is basically reduced by buffers for open RFCOMM ports, active ACL and SCO links and the number of service entries made. For example, the more Service Database entries are made, the less RFComm ports can be opened or the less ACL links can be established.

The LMX9830 has successfully been tested for 7 active links in combination with 1 SCO link, 7 open ports, and 7 service entries. Since a SCO link takes minimum 33% of the complete bluetooth bandwidth, reliability on link establishment can be reduced.

2.2.2.3 Scatternet Master

In case a bluetooth device is master for one or several slaves and in parallel slave to one master, the connection scenario is called “scatternet”. The LMX9830 is able to be master to one or multiple slaves and in addition can be slave to maximum one master. Within the LMX9830 scenario, this state is called “Scatternet Master”.

The LMX9830 can be assumed to be in this state, after one of the following connection situations

- The LMX9830 accepts an incoming link as "Piconet Master" (2.2.2.2) while the Force Master bit is set to 0x00.
- The device is "Single Slave" (2.2.2.5) and actively establishes a link to another device by using one of the following commands
  — "SDAP Connect" (6.2.2.1)
  — "Establish Link" (6.2.3.1)
  — "Connect to Default Connection" (6.2.7.2)

Once Scatternet Master has been reached LMX9830 is not able to accept another incoming link. However the device will still be discoverable and will still answer to service requests.

2.2.2.4 Transparent Master

In case the LMX9830 is only connected to one other device (“point-to-point” connection), it might be beneficial for the application to send data directly to the UART interface, without having to use the command "SPP Send Data" (6.2.3.3). For this the LMX9830 offers the so call “Transparent Mode” on the UART (see Section 1.3.2.2 "Transparent Mode" on page 15), which allows to send data directly.

Since the LMX9830 in this case gets no information to which port to send this data to, transparent mode is only allowed on a point-to-point connection. Data will be routed directly from the UART interface to the remote bluetooth device.

Transparent Master means, the LMX9830 is master for the point-to-point connection to one other link and Transparent Mode is switched on. This state is reached by one of the following situations

- Sending the "Transparent Mode" (6.2.4.1) command in "Piconet Master" (2.2.2.2) state.
- In case Automatic operation is ON, Default Connection setup after Reset, in which the transparent flag is set to 0x01.
- Default Connection setup by using "Connect to Default Connection" (6.2.7.2) in which the transparent flag is set to 0x01.

Leaving Transparent Master, initiated by a UART break, will lead into "Piconet Master" (2.2.2.2) state.

Since the LMX9830 can not send any events or react on incoming commands, scanning is switched off and therefore the LMX9830 is not discoverable or connectable for other devices.

2.2.2.5 Single Slave

The Bluetooth specification [2] defines a bluetooth slave as the device which is connected by another device and adjusting to the timing of that device (Master). The slave synchronizes to the clock of Master and to its hopping sequence. In an active link, the master polls each slave (by default every 40 slots, see also "Set Default Link Latency" (6.2.14.11)) to keep them synchronized but also to allow the slave to send data.

By default, the LMX9830 accepts any incoming link. Depending on the configuration of Security level ("Set Security Mode" (6.2.16.2)) and the Service Database entry, the device will ask for authentication.

The LMX9830 can be assumed to be in Single Slave after one of the following actions appeared:

- The LMX9830 accepted an incoming link and reports it by the "SPP Link Established Indicator" (6.2.3.2), while the Automatic Operation flag is set to 0x00 (Non-automatic).
- The host sends a UART Break to a LMX9830 in "Transparent Slave" (2.2.2.7)

In theory, there's also the possibility for Single Slave, in case the LMX9830 actively establishes a link to another device, which requests a role switch. In this case the initiating device will be slave. Since the role switch is not reported to the command interface and therefore can't be proven, this scenario shall not be discussed in this document.

In Single slave the LMX9830 still listens to commands on the UART. The device will be discoverable and connectable for other devices. The existing link can be used as basis for a SCO link.

2.2.2.6 Scatternet Slave

An advanced but not most efficient connection state is the Scatternet Slave. In this mode the LMX9830 is slave to two different masters. This means, the LMX9830 needs to switch between two different synchronization states over time, serving each Master only for a limited time. While it is synchronized to Master 1 it is not able to listen to Master 2, therefore might miss the poll packages.

The LMX9830 can be assumed to be in Scatternet Slave after the following action appeared:

- The LMX9830 accepted an incoming link as "Single Slave" (2.2.2.5), reported by "SPP Link Established Indicator" (6.2.3.2).

The LMX9830 is able to manage such a link without issues. However, since the switching between two piconets consumes significant bandwidth, "Piconet Master" (2.2.2.2) or at least "Scatternet Master" (2.2.2.3) should be used instead, which for example can be reached by setting the Force Master Flag within the NVS (see "Force Master Role" (6.2.14.8)).

In Scatternet Slave, the LMX9830 can NOT handle a SCO link on one of the links. Discoverability and connectability are switched off.

2.2.2.7 Transparent Slave

A typical scenario for the LMX9830 is the cable replacement, in which the LMX9830 just waits for an incoming connection and the host connected over UART starts transmitting data after an incoming link has been established. Since in many cases the LMX9830 just replaces a former cable connection, the "Transparent Mode" (1.3.2.2) on the UART allows implementation without any software change on the data transmissions.

Transparent Slave means, the device is slave on the bluetooth link and the UART is switched to "Transparent Mode" (1.3.2.2).

The LMX9830 can be assumed to be in "Transparent Slave" state after on of the following actions appeared:

- the LMX9830 accepted an incoming link in "Idle, Automatic" (2.2.2.1), in which the Automatic Flag is switched to On.
- the host sent command "Transparent Mode" (6.2.4.1) while the LMX9830 is in "Single Slave" (2.2.2.5) state.

Leaving Transparent Slave, initiated by a UART break, will lead into"Single Slave" (2.2.2.5) state.

Since the LMX9830 can not send any events or react on incoming commands, scanning is switched off and therefore the LMX9830 is not discoverable or connectable for other devices.

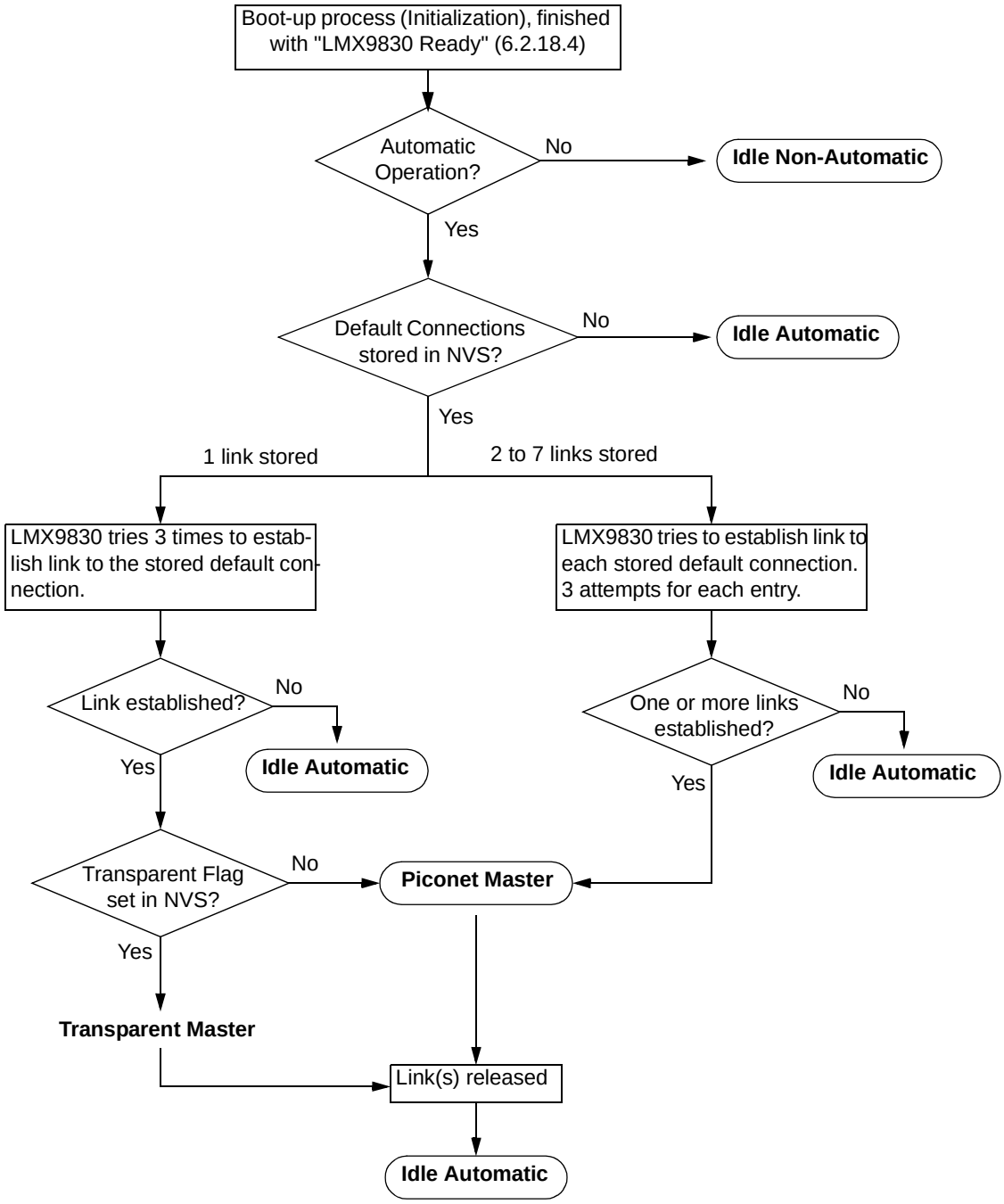


Figure 2-9. Operation Flow after boot-up or Reset

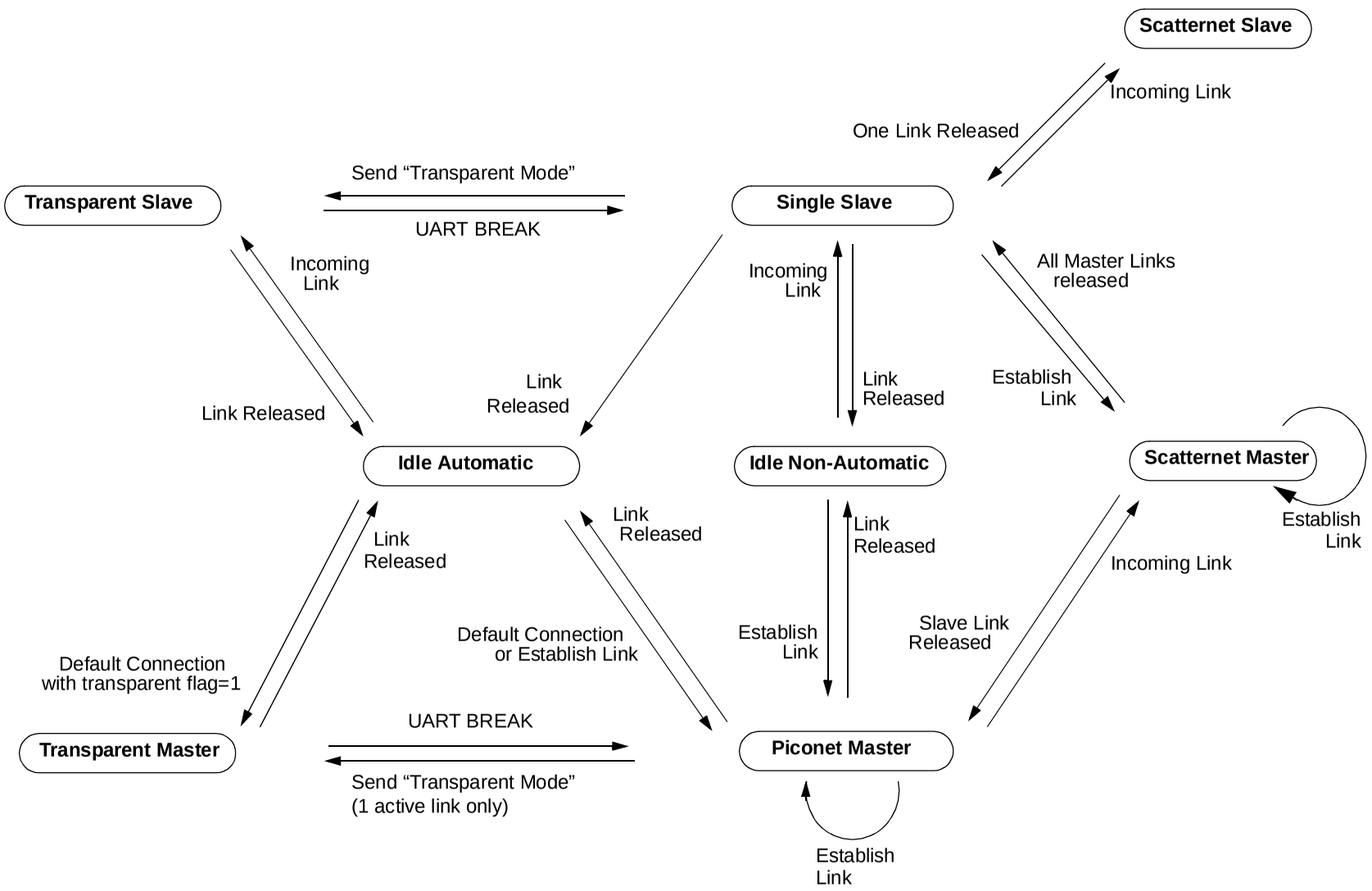


Figure 2-10. Relation between different Operation states

2.3 OPERATION WITHOUT EEPROM

As mentioned, the LMX9830 has a number of "System Parameters" (1.2), which are usually stored in a dedicated area in RAM, also called NVS. In case an EEPROM is available, the LMX9830 will store these parameters also on this. During "Initialization Mode" (2.1), on executing "Enter Bluetooth Mode" (6.2.20.2), the LMX9830 copies the EEPROM content into the RAM NVS.

This allows to operate either with or without EEPROM without any difference in bluetooth mode. But since the RAM NVS is lost on a hardware reset or power-down, the behavior differs in various situations.

Acting without EEPROM means that these "System Parameters" (1.2) have to be managed by the host application and restored after hardware or some of them even software reset.

In general, without EEPROM, the LMX9830 will indicate each change in the RAM NVS by the "Write NVS Indicator" (6.2.23.3). The Indicator includes the address, the length of the data and the data. This indicator allows the host to restore the RAM NVS changes after a hardware reset or power-cycle.

Figure 2-11 shows an example of writing the BD_Addr, if no EEPROM is connected. Table 2-3 gives a detailed description of the UART communication for this example.

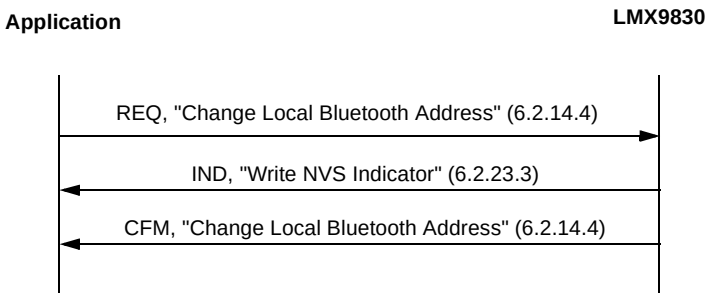


Figure 2-11. Write_NVS Indicator if no EEPROM connected

Table 2-3. Write_NVS Indicator example

Direction	What	Hex Code	Interpreted by Simply Blue Commander
TX	Request	02,52,27,06,00,7F,12,34,56,78,9A, BC,03	Tx: Cmd: Change Local BdAddress, BD_Addr: 123456789ABC
RX	Indicator	02,69,73,09,00,E5,00,00,06,12,34, 56,78,9A,BC,03	Rx: Event: Write NVS, Address: 0000, Data: 123456789ABC
TX	Confirm	02,43,27,01,00,6B,00,03	Rx: Event: Change Local Bd Address, Status: 00

The parameters to be taken care of on the host can be separated into the following categories:

2.3.1 Main configuration parameters

The following parameters are mandatory to be stored in the device before entering "Bluetooth Mode" (2.2).

- BD_Addr
- Clock (if not set by OP pins)
- Baud rate (if not set by OP pins)

A Hardware reset or power cycle has the effect that the complete RAM including NVS section is reset by the hardware. Due to this the LMX9830 will always come up with in "Initialization Mode" (2.1) with the "Await Initialization Event" (6.2.20.1), since the BD_Addr has to be set as a minimum. Clock and UART Baud rate might already be set by the OP pins.

A software reset ("Reset" (6.2.18.3)) does not affect these parameters so the LMX9830 will stay in "Bluetooth Mode" (2.2).

2.3.2 Bluetooth configuration parameters

Most of the "System Parameters" (1.2) influence the bluetooth behaviour of the device and are either stored by the host or generated by the device itself:

- Static configuration settings, e.g.

  - — Default Audio settings
  - — Default link latency
  - — Local Name
  - — Class of Device
  - — Fixed pin
- Dynamic parameters
  - — Link keys stored during pairing procedure

If a static parameter is not changed by the host, the LMX9830 will use the default setting as defined in Table 1-1 "LMX9830 System Parameters, EEPROM Memory Map" on page 10.

Dynamic parameters as the link keys are of special interest, since they are generated during pairing process and are used during the authentication procedure with a remote device. To make sure, the bluetooth devices do not have to pair again after a hardware reset of the LMX9830, these link keys should be restored.

Both, static and dynamic parameters are not affected by software reset but need to be restored after power-cycle or hardware reset.

2.3.3 Service Database

The service database is the source for a remote device to see, which services/profiles the LMX9830 and its host can offer. This database is usually stored in EEPROM and recovered after any Reset.

Since this database can be quite large, it is not part of the standard RAM NVS but is kept as instance in the main working RAM. This allows to keep flexibility and performance as high as possible.
Therefore, if no EEPROM is connected, **the service database has to be initialized after each reset, software or hardware**.
If no configurations are made, the SDB is configured to its default setting. See Section 4.1.2.9 "Service Database" on page 53 for details on the configuration of the service database.

2.3.4 Patches

A patch allows to correct small parts of the firmware in ROM. In case a patch is available, the LMX9830 will use the patch code instead of the ROM code.

A patch has to be provided during "Initialization Mode" (2.1) and is cleared by any reset, hardware or software reset.
In case a "Reset" (6.2.18.3) has been sent to the LMX9830, it will not use a patch previously stored during the initialization.

The general initialization is shown in Figure 2-7 on page 19.

To still be able to apply the patch without hardware reset, the BD_Addr parameter can be used for a workaround. As mentioned in Section 2.3.1 the LMX9830 will stay in "Initialization Mode" (2.1) if one of the mandatory parameters is missing. This behaviour can be used by just changing the BD_Addr to 0xFFFFFFFFFFFF before doing the reset. This will force the LMX9830 to stay in "Initialization Mode" (2.1). Afterwards the BD_Addr needs to be set back to the original value and the patch can be applied before entering bluetooth mode by "Enter Bluetooth Mode" (6.2.20.2).

Figure 2-12 on page 30 shows the workaround to be able to patch after a "Reset" (6.2.18.3). The patch has to be provided before the "Enter Bluetooth Mode" (6.2.20.2) command.

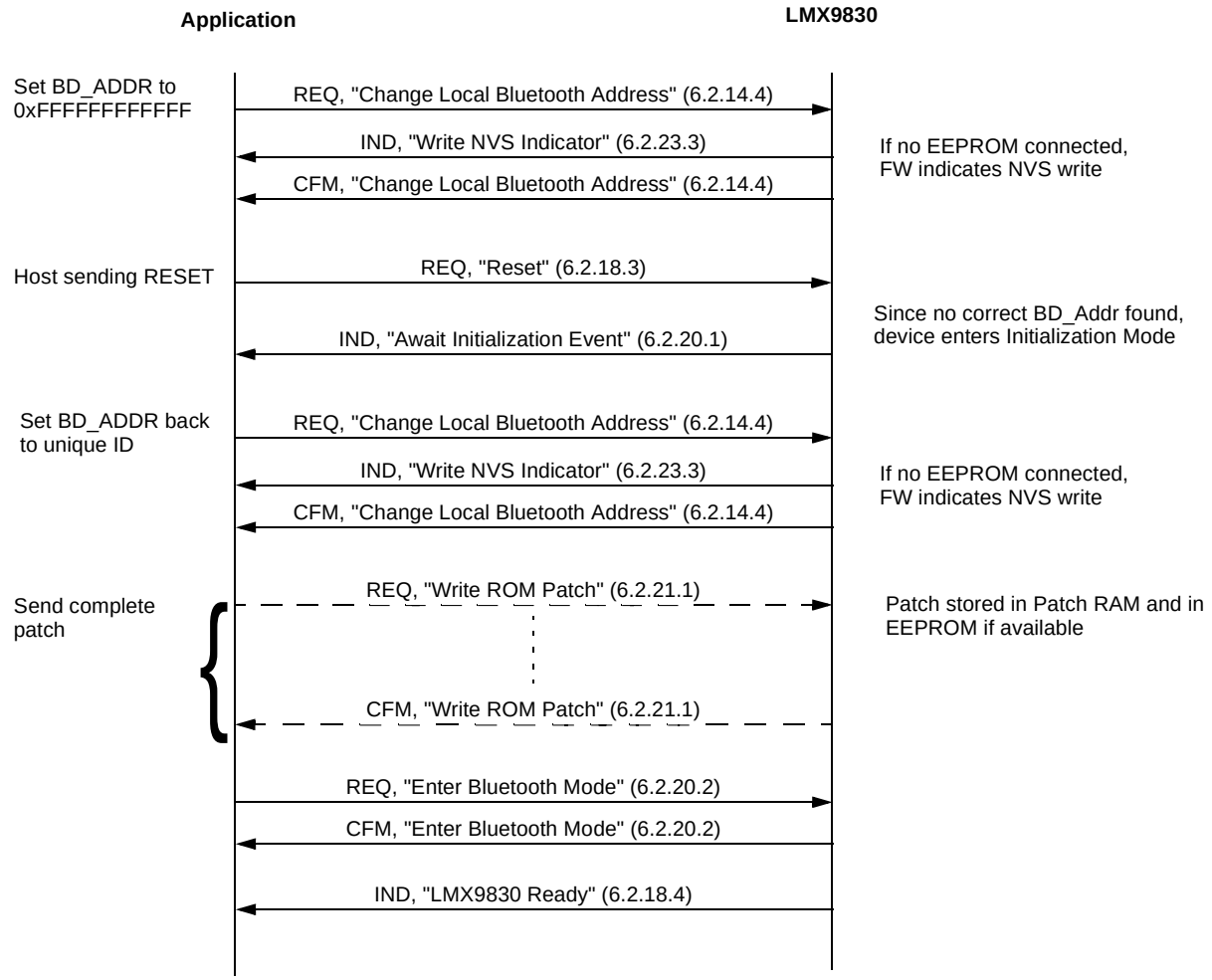


Figure 2-12. Workaround to enable patching after software reset if no EEPROM connected

3.0 Basic Link Establishment

This description is using Simply Blue Commander (SBC) Log Entries to show the command structure and their results. The SBC figures show the commands from bottom to top. The table show the events as interpreted by the SBC as well as Raw Hex string, to directly see the real UART communication.

3.1 ACCEPTING INCOMING LINKS

As described in Section 2.2.2 "Operation States" on page 21 the LMX9830 on default is in a waiting mode (Idle Automatic) after boot-up or reset. This means it waits for requests and automatically answers to connection requests.

If connected from a remote device, the LMX9830 establishes automatically a SPP link and indicates the established link to host by the “Link Established Event” and by setting LSTAT1 pin to 1. By default the device enters "Transparent Slave" (2.2.2.7) state.

Rx: Event: Incoming Link Established, BdAddr: 123456789ABC, Local Port: 01

Figure 3-13. Incoming Link Established

Table 3-4. Incoming Link Established

Direction	What	Hex Code	Interpreted by Simply Blue Commander
RX	Indicator	02,69,0C,07,00,7C,22,22,22,22,22,22,01,03	Rx: Event: Incoming Link Established, BDAddr: 222222222222, Local Port: 01

The event indicates the local RFComm Port and the BD_Addr of the remote device.

Table 3-5. Example Incoming Link Established

Byte	Parameter	Value
01	Start Delimiter	02
02 - 06	Package Header	69 0C 07 00 7C
07- 12	BD_Addr of remote device	BF 8C 03 02 90 00
13	Local RFComm Port	01
14	End Delimiter	03

After this event, the module automatically switches to"Transparent Mode" (1.3.2.2) and routes all incoming and outgoing data from the RF side “unmodified” to the UART or vice versa. This switching process is not indicated to the host.

If the link is dropped, the LMX9830 will empty its buffers and

- send a UART break to the host
- send “Transparent Mode” Indicator (Section 6.2.4.1 on page 135)
  — Indicates on protocol level to the host that transparent mode has been left.
- send “SPP Release Link” Indicator (Section 6.2.3.5 on page 131)
  — Indicates that link has been released.
- LSTAT1 pin will be set back to 0

Rx: Event: Link Released, Reason: 01, Local Port: 01
Rx: Event: Transparent Mode, Local Port: 01, Mode: 00

Figure 3-14. Standard Link Released Messages

Table 3-6. Standard Link Released Messages

Direction	What	Hex Code	Interpreted by Simply Blue Commander
RX	Indicator	02,69,11,02,00,7C,01,00,03	Rx: Event: Transparent Mode, Local Port: 01, Mode: 00
RX	Indicator	02,69,0E,02,00,79,01,01,03	Rx: Event: Link Released, Reason: 01, Local Port: 01

The indicators “transparent mode” and “link released” report within their package the local RFComm port and the current mode respectively the reason of releasing the link.

Please see also Section 6.1 "UART Protocol principles" on page 117 for a complete description of the package and header format.

Table 3-7. Example Transparent Mode Lost Indicator

Byte	Parameter	Value
01	Start Delimiter	02
02 - 06	Package Header	69 11 02 00 7C
07	Local RFComm Port	01
08	Mode	00 (Command Mode)
09	End Delimiter	03

Table 3-8. Example Link Released Indicator

Byte	Parameter	Value
01	Start Delimiter	02
02 - 06	Package Header	69 0E 02 00 79
07	Reason byte	01 (Remote device disconn.)
08	Local RFComm Port	01
09	End Delimiter	03

3.2 SETTING UP A LINK USING THE COMMAND INTERFACE

Setting up a bluetooth link between devices requires that the devices know specific parameters of each other. To get those parameters several steps have to be processed before a SPP link can be established to the device.

All commands necessary for this section can be found in “CreateSPPLink.dir” of the Simply Blue Commander.

The first commands will be explained very detailed for better understanding of the syntax and logic of the command interface.

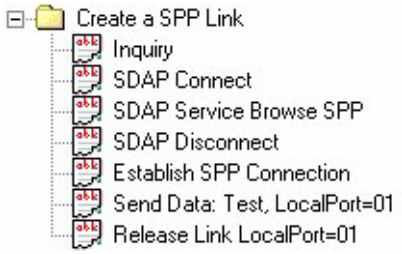


Figure 3-15. CreateSPPLink.dir

3.2.1 Inquiry

The Inquiry process searches for devices in range and gets its BD_Addr (Bluetooth Device Address). This address is a unique address for each Bluetooth device on the market.
Also transmitted with it is the “Class of Device” of this device.

With LMX9830 this process can be started with the command “Inquiry” (Section 6.2.1.1).

The command results in two different events:

- Device Found Indicator
- Inquiry Complete Confirmation

Each found device will be indicated by the Device Found Indicator, including its BD_Addr and Class of Device.

The Inquiry Complete indicates the end of the Inquiry process. Figure 3-16 shows the log as interpreted by the Simply Blue Commander.

```
Rx: Event: Inquiry, Status: 00
Rx: Event: Device Found, BdAddr: 469528D90A00, DeviceClass: 040252
Tx: Cmd: Inquiry, Length: 0A, NumResponces: 00, Mode: 00
```

Figure 3-16. Inquiry command example

Table 3-9. Inquiry Command Example

Direction	What	Hex Code	Interpreted by Simply Blue Commander
TX	Request	02,52,00,03,00,55,0A,00,00,03	Tx: Cmd: Inquiry, Length: 0A, NumResponces: 00, Mode: 00
RX	Indicator	02,69,01,09,00,73,46,95,28,D9,0A,00,04,02,52,03	Rx: Event: Device Found, BdAddr: 469528D90A00, DeviceClass: 040252
RX	Confirm	02,43,00,01,00,44,00,03	Rx: Event: Inquiry, Status: 00

Table 3-9 shows the package format used on the Command interface. The TX indicates the Inquiry command sent to the device, the two Rx lines the events from the LMX9830.

The following paragraphs explain the package format and usage in more detail. The complete package format is also described in Section 6.1.

a.) The Inquiry Command
Let’s first have a look on the Inquiry command:

02 52 00 03 00 55 0A 00 00 03

Any package, request or event, has a 6 byte header

- Start delimiter (1 byte)
- Packet Type Identification (1 byte)
- Opcode (the actual command, 1 byte)
- Payload length (2 bytes)
- Checksum (1 byte)

In this case:

02 52 00 03 00 55

The **start delimiter** is always 0x02.

The **packet type id** for a request is 0x52. (see Section 6.1.3 for the complete list of packet types)

The **opcode** for Inquiry is 00 (see Section 6.1.4 for the complete list of opcodes)

The **payload length** indicates the length of the payload after the checksum.

The payload for this package is

0A 00 00

so the length is 0x0003 (bytes).

The **checksum** is calculated as sum of packet type id, opcode and packet length,

0x52 + 0x00 + 0x03 + 0x00 = 0x55

The **payload** for this command consists of three parameters:

- Inquiry length - 0x0A (10 seconds)
- Number of responses - 0x00 (no limitation)
- Inquiry Mode - 0x00 (General Inquiry)

Table 3-10. Example Inquiry Command Package

Byte	Parameter	Value
01	Start Delimiter	02
02 - 06	Package Header	52 00 03 00 55
07	Inquiry Length	0A (10 seconds)
08	Number of Responses	00
09	Inquiry Mode	00 (General Inquiry)
10	End Delimiter	03

All packages have to end with the **end delimiter** 0x03.

b.) The Device Found Indicator

The first response to the inquiry command from the LMX9830 is the Device_Found_Indicator. In hex:

02 69 01 09 00 73 46 95 28 D9 0A 00 04 02 52 03

Package header:

- Start delimiter - 0x02
- Packet Type - Indicator: 0x69
- Opcode - 0x01 (Indicator opcode, different from command opcode)
- Payload Length - 0x0009 (byte swapped in the package)
- Checksum - 0x69 + 0x01 + 0x09 + 0x00 = 0x73

The **Payload**:

46 95 28 D9 0A 00 04 02 52

- BD_Addr - 46 95 28 D9 0A 00
- Class of Device - 04 02 52

Because of the Little Endian format, both parameters have to be byte swapped. So the “real” values are:

BD_Addr: 00 0A D9 28 95 46

Class of Device: 52 02 04 (Mobile Phone)

Table 3-11. Example Device Found Indicator Package

Byte	Parameter	Value
01	Start Delimiter	02
02 - 06	Package Header	69 01 09 00 73
07 - 12	BD_Addr	46 95 28 D9 0A 00
13 - 15	Class of Device	04 02 52
16	End Delimiter	03

c.) The Inquiry Confirm

Every command on the LMX9830 command interface is confirmed by an appropriate event. The confirmation always has the opcode as the command sent to the device. The event also indicates the success status of the command or any parameters requested. If no error occurred, status/error 0x00 will be returned. All other values have a specific reason. Please see Table 6-298 "Generic Error Codes" on page 198 and Table 6-299 "RFCOMM Error Codes" on page 200 for a complete list of error codes.

The confirmation in hex:

02 43 00 01 00 44 00 03

Package header:

- Start delimiter - 0x02
- Packet type - confirm: 0x43
- Opcode - 0x00 (confirmation, same as command)
- Payload length - 0x0001 (byte swapped in the package)
- Checksum - 0x43 + 0x01 + 0x00 + 0x00 = 0x44

The payload of a confirmation consists at least of the status byte. In this case 0x00.

Table 3-12. Example Inquiry Confirm Package

Byte	Parameter	Value
01	Start Delimiter	02
02 - 06	Package Header	43 00 01 00 44
07	Status/Error Code	00
08	End Delimiter	03

3.2.2 Create SDAP Connection

To create a SPP connection to another device, the local RFComm channel has to know which remote RFComm Channel to address. Each service is registered to a specific RFComm channel number. To get this number the local device has to do a Service Request on the remote device and get the service entry.

The first command necessary for this is the “Create SDAP Connection”. This command establishes a SDP based connection to the other device.

Rx: Event: SDAP Connect, Status: 00
Tx: Cmd: SDAP Connect, BdAddr: 469528D90A00

Figure 3-17. Log of the Create SDAP Command.

Table 3-13. Log of Create SDAP Command

Direction	What	Hex Code	Interpreted by Simply Blue Commander
TX	Request	02,52,32,06,00,8A,46,95,28,D9,0A,00,03	Tx: Cmd: SDAP Connect, BdAddr: 469528D90A00
RX	CFM	02,43,32,01,00,76,00,03	Rx: Event: SDAP Connect, Status: 00

Table 3-14. Example Create SDAP Connection

Byte	Parameter	Value
01	Start Delimiter	02
02 - 06	Packet Header	52 32 06 00 8A
07 - 12	Remote BD_Addr	46 95 28 0D 0A 00
13	End delimiter	03

The only parameter of the command is the BD_Addr to connect to:

46 95 28 D9 0A 00 (byte swapped)

The command is confirmed by the LMX9830 with the appropriate confirmation event. If status is 0x00 the link has been established.

3.2.3 SDAP Service Browse for SPP

After the SDAP connection is established, the service request can be sent. To search for a remote SPP entry, UUID 1101 can be used.

As any multi-byte parameter the UUID has to be sent byte swapped to the LMX9830 within the command.

.

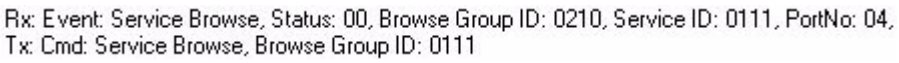


Figure 3-18. Log SDAP Browse for SPP

Table 3-15. Log of SDAP Browse for SPP

Direction	What	Hex Code	Interpreted by Simply Blue Commander
TX	Request	02,52,35,02,00,89,01,11,03	Tx: Cmd: Service Browse, Browse Group ID: 0111
RX	CFM	02,43,35,0D,00,85,00,01,02,10,01,11,04,05,43,4F,4D,31,00,03	Rx: Event: Service Browse, Status: 00, Browse Group ID: 0210, Service ID: 0111, PortNo: 04, Service Name: COM1.

- Opcode - 0x35
- Parameters:
  — UUID for the requested Service: 0x1101

Table 3-16. Example SDAP Browse

Byte	Parameter	Value
01	Start Delimiter	02
02 - 06	Package Header	52 35 02 00 89
08 - 09	UUID	01 11
10	End Delimiter	03

The confirmation of the command includes all information about the registered services on the remote device for the requested UUID.

The full event includes the following parameters:

- Status byte (Error code) - 0x00
- Number of services - 0x02 (Number of services found)
- BrowseGroupID - 0x1002 (Public Browse Group)
- ServiceUUID - 0x1101 (The service found)
- RFComm Port Number - 0x04
- Number of bytes in the service name
- Name of the service

The following table shows the full confirm package for one SPP entry.

Table 3-17. Example SDAP Browse Confirm

Byte	Parameter	Value
01	Start Delimiter	02
02 - 06	Package Header	43 35 0D 00 85
07	Status Byte	00
08	Number of Services	01
09 - 10	Browse Group ID	02 10
11 - 12	Service UUID	01 11
13	Remote RFComm Port Number	04
14	Number of bytes in name	05
15 - 18	Service Name	43 4F 4D 31 00 (COM1)
19	End Delimiter	03

The most important parameter out this event is parameter byte number 13, the RFComm Port Number. This will be needed to create a SPP Link to the other device.

3.2.4 SDAP Disconnect

After a successful Service Browse the connection has to be released again. As there can only be made one SDAP link at the time, the SDAP Disconnect command has no parameters.

Figure 3-19. Log of SDAP Disconnect

Table 3-18. Log of SDAP Disconnect

Direction	What	Hex Code	Interpreted by Simply Blue Commander
TX	Request	02,52,33,00,00,85,03	Tx: Cmd: SDAP Disconnect
RX	CFM	02,43,33,01,00,77,00,03	Rx: Event: SDAP Disconnect, Status: 00

The confirmation of the command just returns the error/status code and is 0x00 on successful disconnection.

3.2.5 Create SPP Connection

Based on the information out of the Inquiry and the service request, a SPP connection can be established to the remote device. (assuming a SPP entry was found).

The following parameters are needed to establish a SPP link to a remote device.

- Command Opcode: 0x0A
- Local RFComm Port: Depending on local configuration, on default RFComm Port 1 is enabled
- Remote BD_Addr: out of Inquiry process
- Remote RFComm Port: out of SDAP Request

.

Table 3-19. Example Create SPP Link

Byte	Parameter	Value
01	Start Delimiter	02
02 - 06	Package Header	52 0A 08 00 64
07	Local RFComm Port	01 (default at LMX9830)
08 - 13	Remote BD_Addr	46 95 28 D9 0A 00
14	Remote RFComm Port	04 (out of SDAP Request)
15	End Delimiter	03

Rx: Event: Link Established, Status: 00, BdAddr: 469528D90A00, Local Port: 01, Remote Port Number: 04
Rx: Event: Port Status Changed, Local Port: 01, PortStatus: 0C, Break Length: 0000
Rx: Event: Establish Link, Status: 00, Local Port: 01
Tx: Cmd: Establish Link, Local Port: 01, BdAddr: 469528D90A00, Remote Port Number: 04

Figure 3-20. Log of Create SPP Connection

Table 3-20. Log of Create SPP Connection

Direction	What	Hex Code	Interpreted by Simply Blue Commander
TX	Request	02,52,0A,08,00,64,01,46,95,28,D9,0A,00,04, 03	Tx: Cmd: Establish Link, Local Port: 01, BdAddr: 469528D90A00, Remote Port Number: 04
RX	Confirm	02,43,0A,02,00,4F,00,01,03	Rx: Event: Establish Link, Status: 00, Local Port: 01
RX	Indicator	02,69,3E,04,00,AB,01,0C,00,00,03	Rx: Event: Port Status Changed, Local Port: 01, PortStatus: 0C, Break Length: 0000
RX	Indicator	02,69,0B,09,00,7D,00,46,95,28,D9,0A,00,01, 04,03	Rx: Event: Link Established, Status: 00, BdAddr: 469528D90A00, Local Port: 01, Remote Port Number: 04

The Log Window shows 3 events returned by the LMX9830.

a) Establish Link Confirm
As any confirm the “Establish Link Confirm” (Section 6.2.3.1) has the same Opcode as the command sent. It includes the following parameters:

- Status/Error Code - 0x00
- Local RFComm Port - 0x01

The event means: “Got the request, trying to set up the link on port 1”.

Table 3-21. Example Establish Link Confirm

Byte	Parameter	Value
01	Start Delimiter	02
02 - 06	Package Header	43 02 0A 00 4F
07	Status/Error Code	00 (for success)
08	Local RFComm Port	01
09	End delimiter	03

b) Port Status Changed Indicator
This event indicates that during the RFComm channel setup process the settings of the SPP link have changed.

The RFComm channel behaves like a virtual serial port with emulated handshaking and flow control.

Please see Section 6.2.10.3 for the detailed description of the event.

c) Link Established Indicator
To indicate an established link on top of the SPP, the LMX9830 uses the “Link Established Indicator” (Section 6.2.3.7). The event returns

- Status/Error code
- BD_Addr of the remote device
- Local RFComm port
- Remote RFComm port

Table 3-22. Example Link Established Indicator

Byte	Parameter	Value
01	Start Delimiter	02
02 - 06	Package Header	69 0B 09 00 7D
07	Status/Error Code	00 (for success)
08 - 13	BD_Addr	46 95 28 0D 0A 00
14	Local RFComm Port	01
15	Remote RFComm Port	04
16	End delimiter	03

The package indicates which local RFComm port is now bound to a specific link.

3.2.6 Sending Data in Command Mode

After actively setting up a connection with the LMX9830, the device is in state "Piconet Master" (2.2.2.2), still listening to commands and returning status changes by events.

To send data over the command interface the “Send Data” (Section 6.2.3.3) command has to be used.

Besides the data which have to be sent, the local RFComm Port parameter has also to be sent to the LMX9830. This enables the application to support multiple connections.

The Log windows in Figure 3-21 show the transmission of the word “Test” over an established SPP link. The data is displayed in hex values.

Rx: Event: Send Data, Status: 00, Local Port: 01
Tx: Cmd: Send Data, Local Port: 01, Payload Data: 54657374

Figure 3-21. Log of sending the Data “Test”

Table 3-23. Log of sending the Data “Test”

Direction	What	Hex Code	Interpreted by Simply Blue Commander
TX	Request	02,52,0F,07,00,68,01,04,00,54,65,73,74,03	Tx: Cmd: Send Data, Local Port: 01, Payload Data: 54657374
RX	Confirm	02,43,0F,02,00,54,00,01,03	Rx: Event: Send Data, Status: 00, Local Port: 01

Table 3-24. Example Sending Data Package

Byte	Parameter	Value
01	Start Delimiter	02
02 - 06	Package Header	52 0F 07 00 68
07	Local RFComm Port	01
08 - 09	Length of Data to send	04 00 (byte swapped)
10 - 13	Data to send	54 65 73 74 (“Test”)
14	End Delimiter	03

NOTE: The length of the data in the payload has influence on the package length within the package header and the length parameter within the payload itself.

Table 3-25 shows a second example with a longer data package, marking changed parameters in bold.

Table 3-25. Example Sending Data “Testdata”

Byte	Parameter	Value
01	Start Delimiter	02
02 - 06	Package Header	52 0F **0B** 00 **6C**
07	Local Port	01
08 - 09	Length of Data	**08 00** (byte swapped)
10 - 13	Data to send	54 65 73 74 64 61 74 61 (“Testdata”)
14	End Delimiter	03

3.2.7 Receiving Data in Command Mode

In command mode, incoming data from a remote device are indicated by the “Incoming Data” (Section 6.2.3.3) Event.

Besides the received data, the event also includes the local RFComm Port, on which the device has received the data.

```
Rx: Event: Incoming Data, Local Port: 01, Received Data: 74
Rx: Event: Incoming Data, Local Port: 01, Received Data: 73
Rx: Event: Incoming Data, Local Port: 01, Received Data: 65
Rx: Event: Incoming Data, Local Port: 01, Received Data: 54
```

Figure 3-22. Log of Incoming Data Event

Table 3-26. Log of Incoming Data Event

Direction	What	Hex Code	Interpreted by Simply Blue Commander
RX	Indicator	02,69,10,04,00,7D,01,01,00,54,03	Rx: Event: Incoming Data, Local Port: 01, Received Data: 54
RX	Indicator	02,69,10,04,00,7D,01,01,00,65,03	Rx: Event: Incoming Data, Local Port: 01, Received Data: 65

Table 3-26. Log of Incoming Data Event

Direction	What	Hex Code	Interpreted by Simply Blue Commander
RX	Indicator	02,69,10,04,00,7D,01,01,00,73,03	Rx: Event: Incoming Data, Local Port: 01, Received Data: 73
RX	Indicator	02,69,10,04,00,7D,01,01,00,74,03	Rx: Event: Incoming Data, Local Port: 01, Received Data: 74

Figure 3-22 and Figure 3-22 show the log of 4 bytes received on local RFComm Port 01. The bytes together form the word “Test” again.

Table 3-27. Example Incoming Data Event

Byte	Parameter	Value
01	Start Delimiter	02
02 - 06	Package Header	69 10 04 00 7D
07	Local RFComm Port	01
08 - 09	Length of Data received	01 00 (byte swapped)
10	Received Data	54 (“T”)
11	End Delimiter	03

3.2.8 Releasing a SPP connection

To release an existing SPP connection the “Release Link” Command (Section 6.2.3.5) is used. The command is referring to the local RFComm port the connection has been established on.

Rx: Event: Link Released, Reason: 00, Local Port: 01
Rx: Event: Release Link, Status: 00, LocalPort: 01
Tx: Cmd: Release Link, Local Port: 01

Figure 3-23. Log of Release Link Command

Table 3-28. Log of Release Link Command

Direction	What	Hex Code	Interpreted by Simply Blue Commander
TX	Request	02,52,0D,01,00,60,01,03	Tx: Cmd: Release Link, Local Port: 01
RX	Confirm	02,43,0D,02,00,52,00,01,03	Rx: Event: Release Link, Status: 00, LocalPort: 01
RX	Indicator	02,69,0E,02,00,79,00,01,03	Rx: Event: Link Released, Reason: 00, Local Port: 01

Table 3-29. Example Release Link Package

Byte	Parameter	Value
01	Start Delimiter	02
02 - 06	Package Header	52 0D 01 00 60
07	Local RFComm Port	01
08	End Delimiter	03

The “Release Link” is confirmed by two events. Both include a status/error byte and the port number.

a) Release Link Confirm

The event confirms to the host that the command has been received and release is initiated.

b) Link Released Indicator

The event indicates that the LMX9830 released the Link on the RFComm Port returned.

Table 3-30. Example Link Released Indicator Package

Byte	Parameter	Value
01	Start Delimiter	02
02 - 06	Package Header	69 0E 02 00 79
07	Reason Code	00
08	Local RFComm Port	01
09	End Delimiter	03

3.3 USING TRANSPARENT MODE

3.3.1 Activating Transparent Mode

In Transparent Mode, described in Section 1.3.2.2, the LMX9830 acts as cable replacement device. In this mode the LMX9830 does not interpret the packages sent to device. Instead, it is directly forwarding the data straight to the link previously set up.

If a bluetooth link to the LMX9830 has been established, this mode is automatically activated if the Automatic Operation Flag in NVS is set to 0x01 (default).

If the link was set up manually via the LMX9830 command interface (see Section 3.2), the LMX9830 still listens to commands and data have to be sent via the “Send Data” command ("Piconet Master" (2.2.2.2)).

As the LMX9830 routes the data directly to the bluetooth link,"Transparent Mode" (1.3.2.2)can only be activated if only one active link exists. The following command has to be used to switch into transparent mode:

- “Transparent Mode” (page 135)

The “Transparent mode” command is referring to the local RFComm port the link was created on.

The command is confirmed by the appropriate event. Afterwards the LMX9830 is routing all incoming data directly to the remote device.

Rx: Event: Transparent Mode, Status: 00, Local Port: 01
Tx: Cmd: Transparent Mode, Local Port: 01

Figure 3-24. Log of Set Transparent Mode

Table 3-31. Log of Set Transparent Mode

Direction	What	Hex Code	Interpreted by Simply Blue Commander
TX	Request	02,52,11,01,00,64,01,03	Tx: Cmd: Transparent Mode, Local Port: 01
RX	Confirm	02,43,11,02,00,56,00,01,03	Rx: Event: Transparent Mode, Status: 00, Local Port: 01

Table 3-32. Example Set Transparent Mode

Byte	Parameter	Value
01	Start Delimiter	02
02 - 06	Package Header	52 11 01 00 64
07	Local RFComm Port	01
08	End Delimiter	03

Table 3-33. Example Set Transparent Mode Confirm

Byte	Parameter	Value
01	Start Delimiter	02
02 - 06	Package Header	43 11 02 00 56
07	Status/Error Code	00
08	Local RFComm Port	01
09	End Delimiter	03

Afterwards any data received will be routed unmodified to the UART.

```
Tx(RAW): 54,65,73,74
Rx(RAW): 54,65,73,74
```

Figure 3-25. Sending/Receiving “Test” in transparent mode

The LMX9830 is leaving the transparent mode, when a break signal is sent on the UART (seeSection 3.3.2).

The break signal can also be used if the LMX9830 has been connected from a remote device and switched automatically to transparent (Automatic slave mode).

The recognized BREAK is confirmed by the “Transparent Mode” Event.

```
Rx: Event: Transparent Mode, Local Port: 01, Mode: 00
```

Figure 3-26. Log of Transparent Mode Event

Table 3-34. Log of Transparent Mode Event

Direction	What	Hex Code	Interpreted by Simply Blue Commander
RX	Indicator	02,69,11,02,00,7C,01,00,03	Rx: Event: Transparent Mode, Local Port: 01, Mode: 00

Table 3-35. Example Transparent Mode Indicator

Byte	Parameter	Value
01	Start Delimiter	02
02 - 06	Package Header	69 11 02 00 7C
07	Local RFComm Port	01
08	Operation Mode	00 (Command)
09	End Delimiter	03

3.3.2 Leaving transparent mode with UART BREAK

The UART Break is defined as the contiguous transmission of “0” (space) for a certain length of time. The CCITT “blue book” specification states that the time duration for this is larger than 2M+3 bit time (where M is the character length). After the break sequence, another 2M+3 bit time consisting of the contiguous transmission of “1” (mark) is required to start the next character.

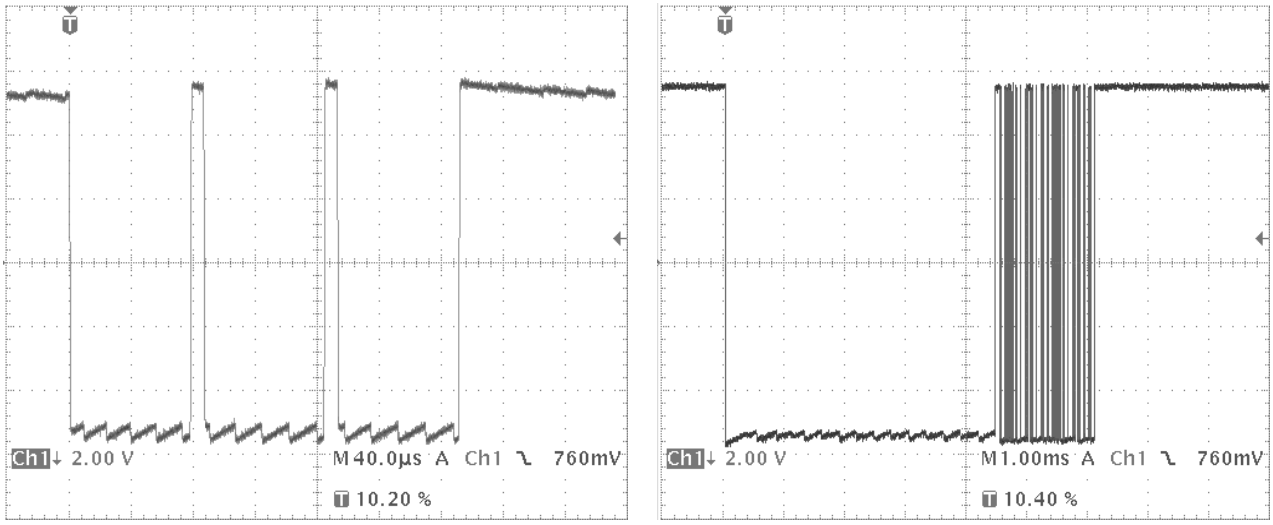


Figure 3-27. Difference between a Standard 0 transmission and BREAK signal

Figure 3-27 shows the difference between the signal of a normal 0 and the BREAK signal. The left picture shows the signalling of 3 Zeros at 115.2kbit/s. Each character is started and ended with a start bit and a Stop bit. The normal length of 1 byte is therefore about 86.8µs (1startbit + 8bit data + 1stopbit).

The picture on the right shows a BREAK signalled by the LMX9830 after a released link. The signal is held low for over 4 ms. Theoretical minimum value for a BREAK at this speed would be about 165µS.

3.4 EXAMPLES (SUMMARY)

The following log files show the typical hex values sent to respectively returned from the LMX9830. The level of events returned by the LMX9830 depends on the event filter level set within the NVS. The tables with the log entries also show the filter level, in which the messages are reported. Default filter setting is 01, so only those events will be reported.

Please see also Section 4.1.1.6 "Event Filter" on page 47 for details on the Event Filter setting.

3.4.1 Automatic Slave

Table 3-36 shows the event returned from the LMX9830 if it was connected from outside. The LMX9830 just returns one event indicating the BD_Addr of the remote device and the local RFComm port it connected to.

Table 3-36. Log File of Incoming Link as automatic slave

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
00	RX	Indicator	02,69,50,07,00,C0,12,34,56,78,9A, BC,00,03	Rx: Event: ACL Established, BdAddr: 123456789ABC, Status: 00
00 / 01	RX	Indicator	02,69,0C,07,00,7C,12,34,56,78,9A ,BC,01,03	Rx: Event: Incoming Link Established, BdAddr: 123456789ABC, Local Port: 01

Table 3-37 shows the events of typical procedure if a link was released from the other device.

Table 3-37. Log File of a Released Link as Automatic Slave

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
00 / 01	RX	UART Break		Rx(RAW): 00
00 / 01	RX	Indicator	02,69,11,02,00,7C,01,00,03	Rx: Event: Transparent Mode, Local Port: 01, Mode: 00

Table 3-37. Log File of a Released Link as Automatic Slave

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
00 / 01	RX	Indicator	02,69,0E,02,00,79,01,01,03	Rx: Event: Link Released, Reason: 01, Local Port: 01
00	RX	Indicator	02,69,51,07,00,C1,12,34,56,78, 9A,BC,13,03	Rx: Event: ACL Terminated, BdAddr: 123456789ABC, Reason: 13

3.4.2 Setting up a link

As documented in Section 3.2, setting up one or more links to another device in general requires the knowledge of the BD_Addr and the RFComm Port to connect to. Table 3-38 shows all commands necessary from scratch to establish a link to another device.

Table 3-38. Log File of a complete link setup

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
	TX	Request	02,52,00,03,00,55,0A,00,00,03	Tx: Cmd: Inquiry, Length: 0A, NumResponces: 00, Mode: 00
00 / 01	RX	Indicator	02,69,01,09,00,73,12,34,56,78, 9A,BC,00,00,00,03	Rx: Event: Device Found, BdAddr: 123456789ABC, DeviceClass: 000000
00 / 01	RX	Confirm	02,43,00,01,00,44,00,03	Rx: Event: Inquiry, Status: 00
	TX	Request	02,52,32,06,00,8A,12,34,56,78, 9A,BC,03	Tx: Cmd: SDAP Connect, BdAddr: 123456789ABC
00	RX	Indicator	02,69,50,07,00,C0,12,34,56,78, 9A,BC,00,03	Rx: Event: ACL Established, BdAddr: 123456789ABC, Status: 00
00 / 01	RX	Confirm	02,43,32,01,00,76,00,03	Rx: Event: SDAP Connect, Status: 00
	TX	Request	02,52,35,02,00,89,01,11,03	Tx: Cmd: Service Browse, Browse Group ID: 0111
00 / 01	RX	Confirm	02,43,35,0D,00,85,00,01,02,10, 01,11,01,05,43,4F,4D,31,00,03	Rx: Event: Service Browse, Status: 00, Browse Group ID: 0210, Service ID: 0111, PortNo: 01, Service Name: COM1.
	TX	Request	02,52,33,00,00,85,03	Tx: Cmd: SDAP Disconnect
00	RX	Indicator	02,69,51,07,00,C1,12,34,56,78, 9A,BC,16,03	Rx: Event: ACL Terminated, BdAddr: 123456789ABC, Reason: 16
00 / 01	RX	Confirm	02,43,33,01,00,77,00,03	Rx: Event: SDAP Disconnect, Status: 00
	TX	Request	02,52,0A,08,00,64,01,12,34,56, 78,9A,BC,01,03	Tx: Cmd: Establish Link, Local Port: 01, BdAddr: 123456789ABC, Remote Port Number: 01
00 / 01	RX	Confirm	02,43,0A,02,00,4F,00,01,03	Rx: Event: Establish Link, Status: 00, Local Port: 01
00	RX	Indicator	02,69,50,07,00,C0,12,34,56,78, 9A,BC,00,03	Rx: Event: ACL Established, BdAddr: 123456789ABC, Status: 00
00 / 01	RX	Indicator	02,69,3E,04,00,AB,01,0C,00,00 ,03	Rx: Event: Port Status Changed, Local Port: 01, PortStatus: 0C, Break Length: 0000
00 / 01	RX	Indicator	02,69,0B,09,00,7D,00,12,34,56, 78,9A,BC,01,01,03	Rx: Event: Link Established, Status: 00, BdAddr: 123456789ABC, Local Port: 01, Remote Port Number: 01
	TX	Request	02,52,11,01,00,64,01,03	Tx: Cmd: Transparent Mode, Local Port: 01
00 / 01	RX	Confirm	02,43,11,02,00,56,00,01,03	Rx: Event: Transparent Mode, Status: 00, Local Port: 01

4.0 Advanced Usage

The LMX9830 offers a wide variety of functions for different usage models. This section points out the most important features and scenarios covered by the LMX9830.

4.1 LOCAL CONFIGURATION

4.1.1 Hardware Configuration

The LMX9830 has several commands to configure the local hardware. Those include settings for the UART speed and configuration, a soft reset and also include settings to set the device into special test modes. Please check also Section 6.2.18 “Hardware Commands” on page 183 and Section 6.2.19 “Test Modes” on page 187.

4.1.1.1 Change UART settings

The UART speed in general is determined by the choice of the OP pins of the LMX9830. The pins and UART settings in NVS are only checked during the software boot-up process so also after a Reset command.

In case another speed than the default needs to be used, the speed configuration has to be done in NVS. By default a speed of 9.6kbit/s is stored.

Table 4-39. UART Settings

Osc Freq. (MHz)	BBCLK (MHz)	PLL (48 MHz)	OP3	OP4	OP5	Function
12	12	OFF	0	0	0	UART speed read from NVS
10-20	10-20	ON	0	1	0	Clock and UART baudrate detection
13	13	OFF	1	0	0	UART speed read from NVS
13	13	OFF	1	0	1	UART speed 9.6 kbps
13	13	OFF	1	1	0	UART speed 115.2 kbps
13	13	OFF	1	1	1	UART speed 921.6 kbps

Example: Configuring LMX9830 for UART speed of 57.6kbit/s with 13MHz external clock:

- Set OP3=1, OP4=0 and OP5=0
- Power up / Reset the device, the UART is configured on 9.6kbit/s on default. LMX9830 answers with “SimplyBlue LMX9830 Ready”. If set to an unknown speed, please set the OP pins to a dedicated speed and try again.
- Send the following command:
  — “Change NVS UART Speed” (page 186) with parameter 0x06.

Table 4-40. Change NVS UART Speed to 57.6kbit/s

Byte	Parameter	Value
01	Start Delimiter	02
02 - 06	Package Header	52 23 01 00 76
07	UART speed	06
08	End Delimiter	03

- Await the Confirmation Event from the LMX9830.
- Send the following command do a hardware reset
  — “Reset” (page 185)

Afterwards the LMX9830 will communicate with a UART speed of 57.6kbit/s

4.1.1.2 Bluetooth Testmode

The LMX9830 supports the standard Bluetooth “Device Under Test” Mode and a “Local Loopback” Mode. The “Device under Test” Mode is the standard testmode is used by any bluetooth tester. If activated the tester will be able to take control over the LMX9830 and put it into the specific testmodes needed for bluetooth qualification.

The “local loopback” mode is a simple UART loopback mode to test the UART communication interface.

The testmodes are enabled by the command “Test Mode”.

“Bluetooth Device Under Test Mode” can be left by software Reset, “Local Loopback test” requires a hardware reset.

Please see Section 6.2.19.1 “Bluetooth Test Mode” on page 187 for details for the command.

4.1.1.3 RF Testmodes

Bluetooth qualification, FCC and CEPT qualifications also require continuos transmit modes. For this the transmitter has to be set to a specific transmit only or receive only status, with which test houses can make for instance spurious emission testings.

The detailed command is described in Section 6.2.19.2 “Initiate RF Test Mode” on page 188.

4.1.1.4 Restore Factory Settings

The LMX9830 is delivered with standard settings in NVS which can be seen in Table 1-1 "LMX9830 System Parameters, EEPROM Memory Map" on page 10. Those parameters are changing during usage or testing the part.

The restore to factory settings gives the ability to restore all default values as listed in the table and deletes all additional entries. The following parameters are NOT affected by the restore to factory settings:

- Bluetooth Device Address (BD_Addr)

NOTE: Please be aware that this command resets also the clock and baud rate settings. In case the NVS settings are required for one of those parameters, the device will require initialization as described in Section 2.1 “Initialization Mode” on page 18.

This command is not required if operated without EEPROM, since the LMX9830 will use the default settings anyway after each power-up or hardware reset.

Command details can be found in Section 6.2.18.5 “Restore Factory Settings” on page 185.

4.1.1.5 Read RSSI

After link establishment the radio is measuring the “Receive Signal Strength Indicator”, a parameter indicating the signal strength of the incoming packages. Typically the Link Manager forces the remote devices to decrease or increase its output power to improve the receiving performance.

This command offers the ability to read out the RSSI and indicate the current status of the signal strength. (See Section 6.2.3.8 on page 133).

The value indicates:

- 0xFF: Signal too low
- 0x00: Signal OK
- 0x01: Signal too strong

4.1.1.6 Event Filter

The LMX9830 indicates status changes and confirms commands with specific events. To address specific application requirements the level of reporting can be increased or decreased.

The following separation is possible:

- Report standard events including ACL indicators, which indicate the physical connection status.
- Report standard events only
- Report no events, only UART break indicates lost link, LMX9830 still detects UART Break to leave "Transparent Mode" (1.3.2.2)
- No reporting on the UART and "Transparent Mode" (1.3.2.2) not left on UART BREAK

The event filter can be configured by the following command:

- “Set Event Filter” (page 184)
- “Get Event Filter” (page 184)

4.1.1.6.1 Report standard events including ACL

In case the NVS setting is set to “report all events”, the LMX9830 reports any status change and ACL link establishment to the host.

The main difference to the other filter settings are the reported ACL indicators.

This reporting can be necessary if the system is actively trying to connect to another device. The ACL indicator includes an error code, which gives information about the reason of a failed connection. (E.g. failed authentication).

NOTE: Please be aware that any kind of established ACL link will be reported to the host. If the device is only waiting for connection, any attempt from another device to connect to the LMX9830 will be reported to the host. For a “slave only” usage, one of the other filtering settings would probably be more useful.

Available ACL Events:

- “ACL Established” (page 177)
- “ACL Terminated” (page 177)

4.1.1.6.2 Report standard events (default)

In this reporting scheme the LMX9830 reports all events and indicators except the ACL indications. This mode is set as default and is basically allowing backwards compatibility to earlier firmware versions.

4.1.1.6.3 Report no events

In case the UART does not send back any event to the host the Event filter has to be set to “Report no events”. The only “event” is the UART Break, which still indicates the loss of the bluetooth link.

This filter setting is useful if the LMX9830 is used as cable replacement in front of a microcontroller, where no status event can be interpreted. But the device will still send a UART break when"Transparent Mode" (6.2.4.1) is left. In addition the LMX9830 will still recognize a UART BREAK and leave "Transparent Mode" (6.2.4.1) when detected.

In this setting the pin LSTAT1 can be used as hardware indicator about the link status of the LMX9830.

4.1.1.6.4 Report no events, UART Break suppressed and ignored

In addition to the level described in Section 4.1.1.6.3, also the UART BREAK is suppressed and ignored.

This filter setting is useful if the LMX9830 is used as cable replacement in front of a microcontroller, where no status event and even no UART break can be interpreted. In addition the LMX9830 will not recognize a UART BREAK, therefore will not leave "Transparent Mode" (6.2.4.1).

In this setting only pin LSTAT1 can be used as hardware indicator about the link status of the LMX9830.

4.1.2 LMX9830 Bluetooth Configuration

The Local Bluetooth Configuration includes commands for changing parameters which have influence on if or how the device will answer to requests and how it behaves in different situations.

4.1.2.1 Local Bluetooth Device Address

The BD_Addr is a unique identifier for each bluetooth product. It is neither delivered nor pre programmed by TI on the LMX9830. The BD_Addr parameter is stored in NVS and EEPROM in case an EEPROM is connected to the LMX9830.

If necessary this value can be changed to any specific value.

The commands available are:

- “Read Local Bluetooth Address” (page 171)
- “Change Local Bluetooth Address” (page 172)

Note:
Please be aware that by overwriting this address the uniqueness of the device address cannot be guaranteed anymore.

4.1.2.2 Local Name

The Local Name is transmitted on “Remote Name Requests” from other devices. It just represents a friendly name of the device. Default value is “Serial Port Device”.

- “Read Local Name” (page 170)
- “Write Local Name” (page 171)

4.1.2.3 Class of Device

The Class of Device is based on a numbering scheme of the Bluetooth SIG and is returned on Inquiry requests from other devices. The Class of Device indicates the basic functionality of a device like Mobile Phone, Printer, Headset. This number can be used by the main application to already filter the devices in range for certain functionality.

A complete list of numbers and can be found in the “Bluetooth Assigned Numbers” Document provided by the Bluetooth SIG at https://www.bluetooth.org/foundry/assignnumb/document/baseband.

Some Examples:

- Desktop Computer: 00 01 04
- Handheld PDA: 00 01 14
- Cellular Phone: 70 02 04

Note: the values should be seen as examples.

Default value is 00 00 00. (no specific device)

Commands available:

- “Store Class of Device” (page 172)

4.1.2.4 Automatic Operation

The Automatic Operation setting has influence on the behavior of the LMX9830 in different situations.
Please see Section 2.2.2 “Operation States” on page 21 for a detailed description of the different operation states.

Commands available:

- “Read Operation Mode” (page 175)
- “Write Operation Mode” (page 175)

4.1.2.5 Fixed Pin

The LMX9830 stores a fixed pin which will be used during pairing processes. The pin is stored in NVS and can be changed by the following commands:

- “Get Fixed Pin” (page 181)
- “Set Fixed Pin” (page 182)

The pin stores the hex value for the ASCII character used as pin. E.g. the pin “1 2 3 4” will be stored as “31 32 33 34”.

Default value is in ASCII “0 0 0 0”, in hex “30 30 30 30”.

4.1.2.6 Configuring the Default Link Timeout

The bluetooth specification defines a specific timeout, which causes the baseband to drop the link if no packages have been received on a link for a specific period of time. In a standard active bluetooth link which is not used by the application to send data at the moment, the master sends out “poll’ packages in agreed intervals, to keep the slaves synchronized. The default poll period is 40slots (or 25ms). The slaves acknowledge each package with a “Null” package.

In case those poll packages are not received by the slave or the master does not get the acknowledgement from the slave, both devices will still try to send or receive packages from each other until the “supervision timeout” is reached. After that the link is indicated as lost. This “supervision timeout” is set by default to 20seconds.

A slave will not be able to accept an incoming connection until the link is completely dropped. As it might be useful in certain application to give up a link earlier than 20seconds, the LMX9830 gives the ability to configure the supervision timeout for each link. It can either be configured as default value in NVS or only for the existing SPP link.

NOTE: The Link Supervision Timeout should not be set too low. The value shall also guarantee the quality of service of an existing link. Due to the fact that packages can get lost due to noisy environment or a master might need to share his bandwidth between multiple slaves, the timeout also ensures that the devices don't treat the link as lost if a few packages are not received or acknowledged. E.g. a link timeout of only 1 second could even be too short to guarantee a stable link.

4.1.2.6.1 Setting the Default Link Timeout

The default Link Supervision timeout for all devices is 20seconds. As the baseband calculates in slots, the value needs to be stored in number of slots. Therefore the default value used is 0x7D00.

The default link supervision timeout is used for each incoming and outgoing link. The value stored in NVS can be changed anytime and will be active for the next link establishment without Reset.

- "Set Default Link Timeout" (page 136)
- "Get Default Link Timeout" (page 136)

4.1.2.6.2 Changing the Link Timeout of an existing link

After a successful link establishment both parties of a link agreed on a specific link timeout. This timeout can be read back or changed anytime by the following commands. In case the local LMX9830 has the role slave, the parameter will only be set locally. In case the LMX9830 is master for the link, the timeout will also be communicated to the connected device, which then will adjust its timeout to that value as well. See also Section 4.1.2.6.3 for the difference between master and slave.

- "Set Link Timeout for an existing link" (page 137)
- "Get Link Timeout of an existing link" (page 137)

4.1.2.6.3 Difference between Master and slave role for the link timeout

The link timeout is a local parameter stored for each specific link. Therefore a master, which is connected to several devices can have different link time-outs set for each of those links.

The difference between a master and slave is, that a master reports a change of the link timeout to the slave, which then adjust its own timeout to the reported value. In case the "Set link timeout" command will be sent over UART to a LMX9830 in slave role, the setting will only be active locally and no message will be sent to the master. This means the two devices will have different link timeout settings.

The following figures shall demonstrate the influence of default link timeout and set link timeout, in case they are set on master or slave.

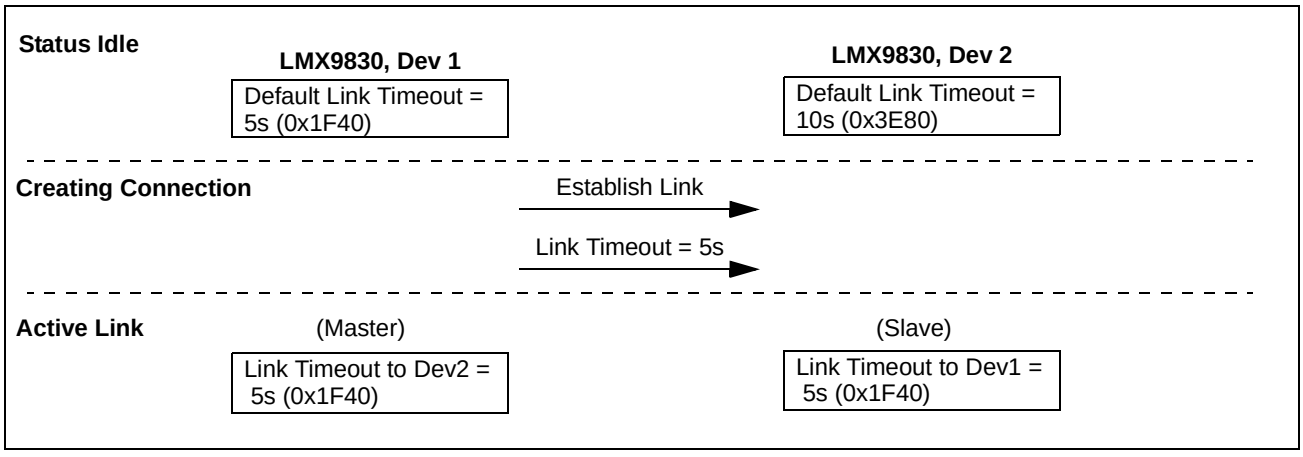


Figure 4-28. Master Default Link Timeout

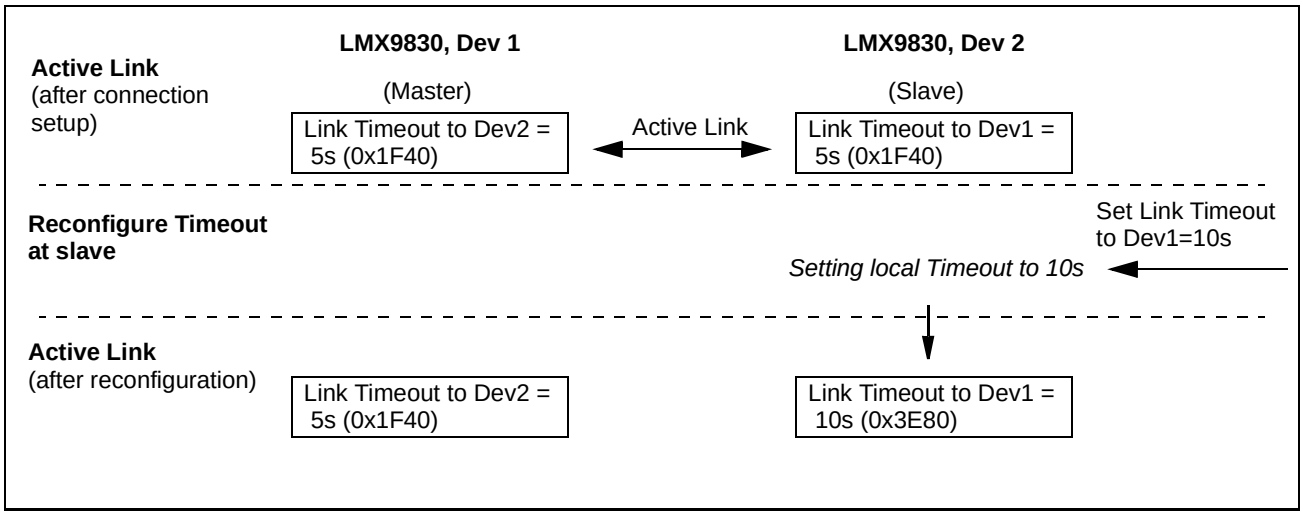


Figure 4-29. Set Link Timeout at Slave

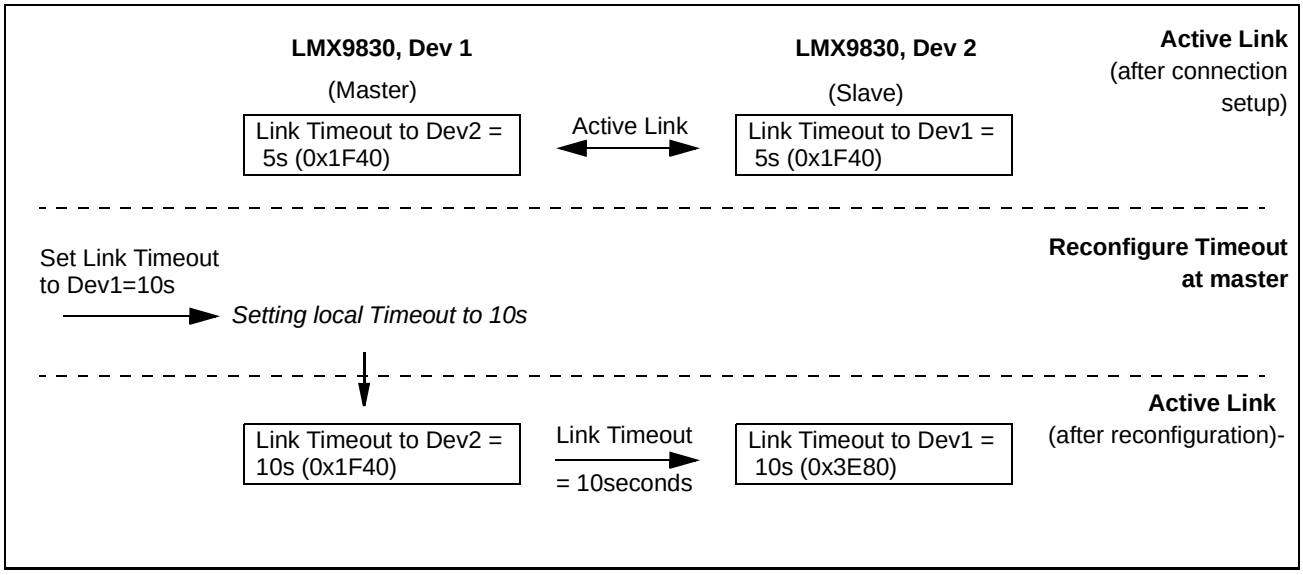


Figure 4-30. Set Link Timeout at Master

4.1.2.7 Configuring the link latency

The link latency is part of the Quality of Service bluetooth is able to offer.

The link latency results in the so called “poll period”.

A standard bluetooth link controlled by the master device, polling each of the slaves connected to it in a predefined period. The polling is necessary to keep the slaves synchronized but also to enable them to send data to the master. Since the master controllers the link it is able to send data immediately to the slave to be addressed. The reaction time for sending data to a slave or a slave device to send data is limited to the poll period agreed with the master. The default poll period for any Bluetooth link is 40 slots (25ms).

In case the slave needs guaranteed data transmission lower than the default 40 slots, the link latency parameter for this link needs to be reduced.

The LMX9830 offers the ability of configuring the default poll period used for each link. The parameter is stored within the NVS and will be requested for any incoming or outgoing link. Since a master might need to manage several slaves, the parameter has to be seen as request, it can not be 100% guaranteed that exactly this value will be used.

The parameter can be set and reviewed by the following commands:

- “Set Default Link Latency” (page 176)

- “Get Default Link Latency” (page 177)

By default the link latency will be set to 0x0000, which means “No specific requirement”. With the master will start sending or requesting data from the slave in a 25ms period. Once the period is reached the device (master or slave) will send the data in a package large enough to carry the complete buffer content. If the buffers are filled up again during the bluetooth transmission, the next slot will be used again.

The minimum value which can be configured is 2 slots, which configures the master to poll the slave at every slot.

NOTE: Using very small poll periods will heavily increase the power consumption of an active link, since the devices exchange packages in lower intervals.

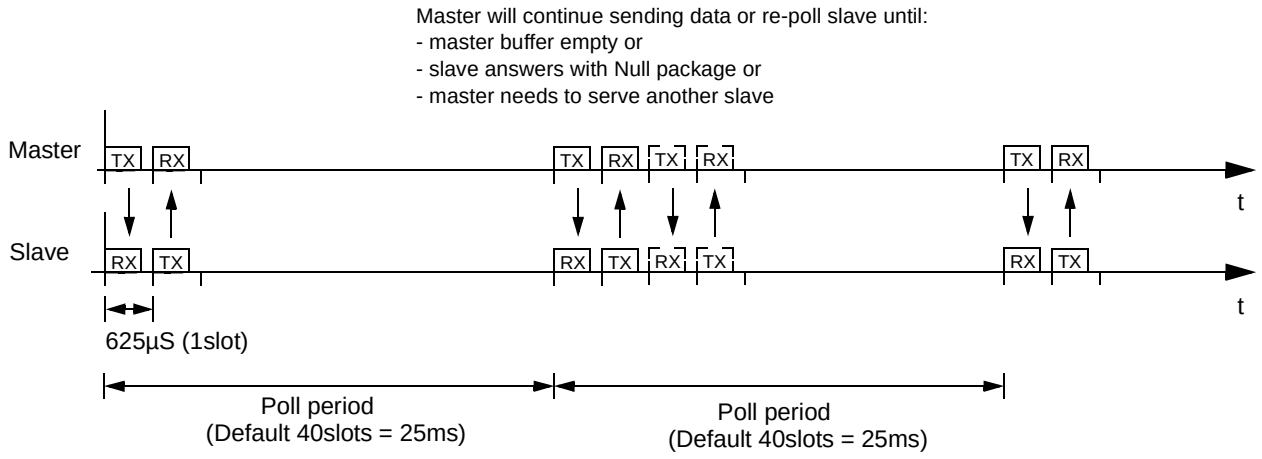


Figure 4-31. Poll period

4.1.2.8 RFComm Ports to open

The Serial Port profile is based on the protocol layer RFCOMM, which offers a serial port emulation to the application interface. Each virtual serial port can be seen as virtual cable between two devices. All data sent to this port will be routed over bluetooth to the remote device.

The LMX9830 can handle up to seven links simultaneously. Each link requires an RFComm port opened, which creates a buffer instance within the RAM to handle the upcoming data traffic. Because of that by default only RFComm port 1 is activated.

The RFComm ports opened and initialized for operation can be configured by the following command:

- Section 6.2.12 "RFcomm Channels to open" on page 166

The ports in this command are expressed by a 32-bit mask indicating which RFCOMM ports the LMX9830 has to open. Bit 30 and 31 must be set to 0. Bit 0 is RFCOMM port 1 and bit 29 is port 30. The changes take effect as soon as the command has been confirmed.

Examples:

Open RFComm port 1: Set Ports to open to 0x00000001
Open RFComm port 1 and 3: Set Ports to open to 0x00000005
Open RFComm port 1, 2 and 3: Set Ports to open to 0x00000007

So to set up 3 Links, Ports to open could for example be configured to 0x00000007. The ports to be opened also depends, how many services are offered to other devices and which ports are bound to these services. Please see also section Section 5.0 "Profile Support" on page 79 for details.

Table 4-41. Open 3 RFComm Ports

Byte	Parameter	Value
01	Start Delimiter	02
02 - 06	Package Header	52 22 04 00 78

Table 4-41. Open 3 RFComm Ports

Byte	Parameter	Value
07 - 10	RFComm Ports to open	07 00 00 00
11	End Delimiter	03

4.1.2.9 Service Database

The LMX9830 contains a service database which configures the settings for authentication and encryption for that specific port. The configuration includes rules for incoming but also outgoing links. Once initialized the service database serves this service requests automatically, without any interaction required with the host.

In principle, the service database is only necessary to be able to offer services to remote devices. E.g. if a remote device creates a SDAP link to the LMX9830 and browses the services, it will only see the entries stored within this database. If for example a second profile needs to be offered, e.g. a second SPP called COM2, it needs to be entered into that database. See also Section 6.2.13 on page 168 for details modifying the "Service Database" (2.3.3).

By default the following entry is stored within the LMX9830 for RFComm Port 1:

Table 4-42. Default SDP entry on RFComm Port 1

Parameter	Description	Value
Entry Index	Index at which the entry can be addressed for enabling or disabling it	0x00
RFComm Port	RFComm Port the settings refer to	0x01
Profile to be offered	The profile to be offered to the remote device. Profiles different from SPP need to be implemented on the host.	SPP
Entry Name	Name which will be shown to the remote device	COM1
Authentication	Defines if authentication is required if port is used for an incoming or outgoing connection. 0x00 No authentication requirements 0x02 Authentication is only required for this profile for incoming connections. 0x20 Authentication is only required for this profile for outgoing connections. 0x22 Authentication is required for this profile for connections in both directions See also "Store Generic SDP Record" (6.2.13.1).	0x02
Encryption	Defines if encryption is required if port is used for an incoming or outgoing connection. 0x00 No encryption requirements 0x04 Encryption is only required for this profile for incoming connections. 0x40 Encryption is only required for this profile for outgoing connections. 0x44 Encryption is required for this profile for connections in both directions See also "Store Generic SDP Record" (6.2.13.1)	0x04

NOTE: The service database security settings are only used when the device is configured to security level 2 (default). Please see also Section 6.2.16 on page 178.

Table 4-42 shows the default service database entry offered to a remote device. The configuration shows that an incoming link which will address RFComm port 1 will require authentication and encryption. In case the link establishment has not happened before, the LMX9830 will automatically initiate a pin code request to the remote device.

In general, for outgoing connections, there's no service database required for that specific RFComm port. For example, if a link is established from local RFComm port 2 to a remote port and no SDB entry has been made for that port, the default will be that no authentication and no encryption will be required.

4.2 SETTING UP MULTIPLE CONNECTIONS

As already described in Section 3.2, the LMX9830 command interface offers the ability to search for other devices, browse the services and to establish a link to another device.

The LMX9830 bluetooth operation is based on the Serial Port Profile. This profile emulates a serial port over a bluetooth link. As bluetooth is able to handle more than one links, the LMX9830 will also offer multiple communication ports, also called RFComm ports to the host. For each port a separate RFComm instance needs to be created.

Figure 4-32 on page 54 shows the standard link establishment as already described in Section 3.2 on page 32. This establishment uses the standard configurations for the LMX9830 with one RFcomm port on the device.

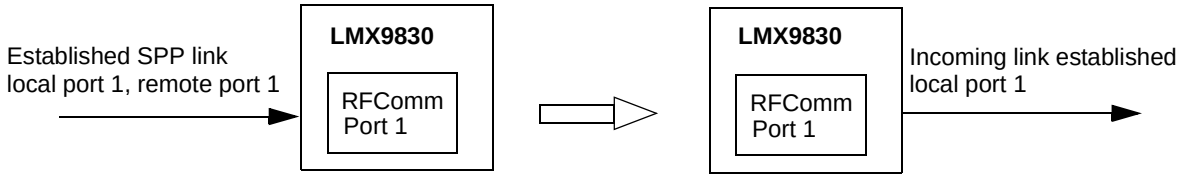


Figure 4-32. Standard point-to-point SPP connection

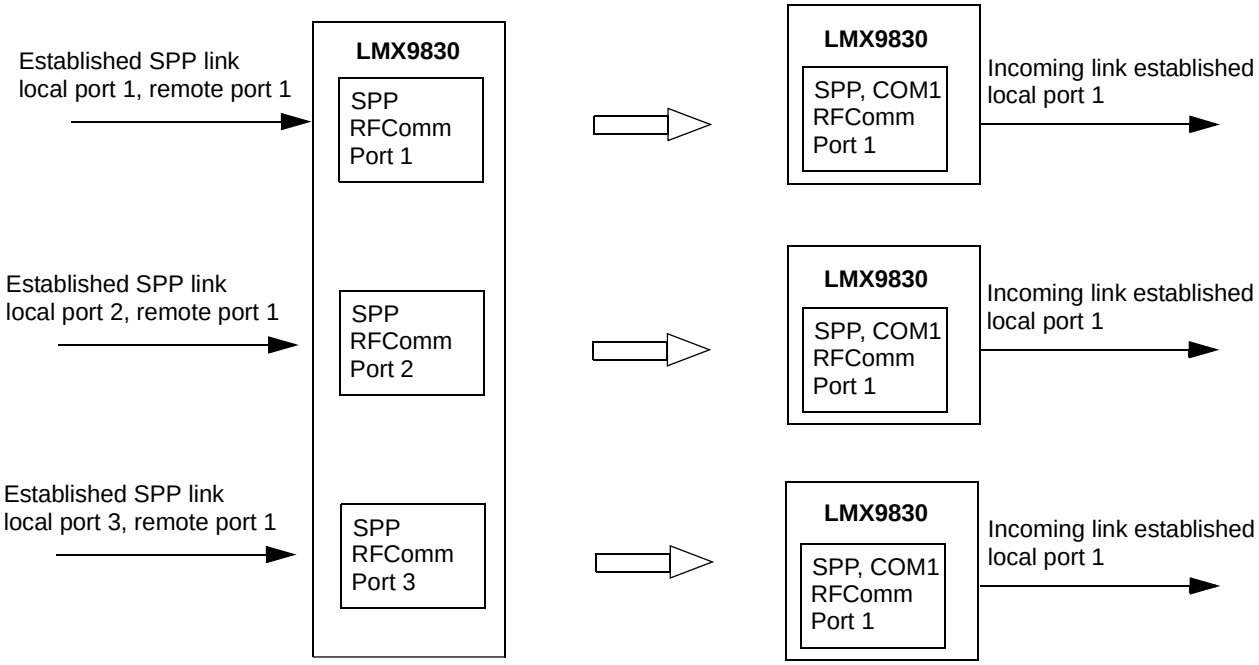


Figure 4-33. Multiple SPP links from LMX9830

Figure 4-33 on page 54shows the principle of establishing multiple links with the LMX9830. The LMX9830 on the left has to initialize 3 RFComm ports, which then can be addressed and bound to a remote device. Any data sent to that specific port are then sent to the device connected to it.

The figure also shows the term “SPP, COM1” which is an indicator for the service database entry stored for this specific port. By default the LMX9830 has configure a Serial Port Profile (SPP) to RFComm port 1. The name, which will be reported to a browsing device is set to “COM1”.

The following sections will guide you through the configuration settings and link setup commands to establish link to three slaves.

4.2.1 Configuration

The LMX9830 by default is configured for point-to-point operation only. Therefore some configuration settings have to be changed before it is possible to establish multiple links.

4.2.1.1 RFComm Ports to open

As explained in Section 4.1.2.8 on page 52 the LMX9830 can handle up to seven links simultaneously. Each link is handled by a RFComm port which creates a buffer instance within the RAM to handle the upcoming data traffic. Because of that by default only RFComm port 1 is activated.

As we need to support three links in this example, ports 1,2 and 3 shall be used.

For this, ports to open has to be configured to 0x00000007.

Table 4-43. Open RFComm Ports 1,2 and 3

Byte	Parameter	Value
01	Start Delimiter	02
02 - 06	Package Header	52 22 04 00 78
07 - 10	RFComm Ports to open	07 00 00 00
11	End Delimiter	03

4.2.1.2 Configuring Service Database for outgoing connections

As explained in Section 4.1.2.9 on page 53 the LMX9830 contains a service database which configures the settings for authentication and encryption for that specific port. The configuration includes rules for incoming but also outgoing links.

In principle, for simple outgoing SPP connection on a port different to RFComm Port 1 it is not required to enter an additional service database entry. The LMX9830 will use the default setting, assuming no auth/encr required for the outgoing link. However, service entries for specific profiles, include also feature settings, which are also checked on outgoing links, e.g. Headset or handsfree profile entries include information on volume control or other features.

To demonstrate the configuration for a secure outgoing connection, RFComm Port 2 shall be entered into the service database as SPP profile, with authentication enabled for outgoing connections.:

Table 4-44. SDB entry with security on outgoing connection

Parameter	**Description**	**Value**
Entry Index	Index at which the entry can be addressed for enabling or disabling it	0x01
RFComm Port	RFComm Port the settings refer to	0x02
Profile to be offered	The profile to be offered to the remote device. Profiles different from SPP need to be implemented on the host.	SPP
Entry Name	Name which will be shown to the remote device	COM2
Authentication	0x20 Authentication is only required for this profile for outgoing connections.	0x20
Encryption	0x40 Encryption is only required for this profile for outgoing connections.	0x40

The following can be used to store the additional entry into the service database.

- "Store Generic SDP Record" (6.2.13.1)

The changes take effect immediately after getting the confirmation event.

Table 4-45. Example SDP Store SPP Record

Byte	Parameter	Value
01	Start Delimiter	02
02 - 06	Package Header	52 31 4C 00 CF
07	Local RFComm Port	02
08	Authentication setting	20
09	Encryption setting	40
10 - 11	Service Name length	47 00

Table 4-45. Example SDP Store SPP Record

Byte	Parameter	Value
12 - 82	Service Record	00 00 0A 00 00 00 00 01 00 35 03 19 01 11 04 00 35 0C 35 03 19 0 0 01 35 05 19 03 00 08 02 05 00 35 03 19 02 10 06 00 35 09 09 6E 65 09 6A 00 09 00 01 09 00 35 08 35 06 19 01 11 09 00 01 00 01 2 5 05 43 4F 4D 32 00
83	End Delimiter	03

4.2.2 Link Establishment

Establishing links to multiple slaves in principle is the same as a standard connection setup already described in Section 3.2 "Setting up a link using the Command Interface" on page 32. The only difference which has to be considered is that the LMX9830 has enough RFComm ports available (opened) and each link is assigned to one specific RFComm port.

The link establishment uses the command "Establish Link" (6.2.3.1) to create all links. Each link established will be confirmed by the appropriate confirmation event. The following three tables show the commands which have to be sent to establish the three links. The commands need to be adjusted for the local RFComm Port, the remote BD_ADDR and the remote RFComm Port Number, derived out of the related SDAP request.

Table 4-46. Establish Link on Local RFComm Port 1

Byte	**Parameter**	**Value**
01	Start Delimiter	02
02 - 06	Package Header	52 0A 08 00 64
07	**Local RFComm Port**	**01**
08 - 13	Remote BD_Addr	12 34 56 78 9A BC
14	Remote RFComm Port	01 (out of SDAP Request)
15	End Delimiter	03

Table 4-47. Establish Link on Local RFComm Port 2

Byte	**Parameter**	**Value**
01	Start Delimiter	02
02 - 06	Package Header	52 0A 08 00 64
07	**Local RFComm Port**	**02**
08 - 13	Remote BD_Addr	34 56 78 9A BC 12
14	Remote RFComm Port	01 (out of SDAP Request)
15	End Delimiter	03

Table 4-48. Establish Link on Local RFComm Port 3

Byte	**Parameter**	**Value**
01	Start Delimiter	02
02 - 06	Package Header	52 0A 08 00 64
07	**Local RFComm Port**	**03**
08 - 13	Remote BD_Addr	56 78 9A BC 12 34
14	Remote RFComm Port	01 (out of SDAP Request)
15	End Delimiter	03

4.2.3 Summary

The following tables show a complete example of configuring a device to establish three links to 3 different slaves.

The example is based on the assumption that the other devices are not known and have never been explored before, so all information have to be collected.

4.2.3.1 Device Configuration, preparing for multipoint operation

In order to create more than one link, the device needs to initialize multiple connections. In case, authentication and encryption are necessary for one of the outgoing links an additional service database entry needs to be made. The example shows a new service entry for Local RFComm Port 2 with Authentication and Encryption enabled.

Table 4-49. Initialize 3 RFComm Ports

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
	TX	Request	02,52,22,04,00,78,07,00,00,00,03	Tx: Cmd: Set Ports To Open, Ports: 07000000
00 / 01	RX	Confirm	02,43,22,01,00,66,00,03	Rx: Event: Set Ports To Open, Status: 00

Table 4-50. Configure additional SDB Entry to enable auth/encr. for outgoing connection

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
	TX	Request	02,52,31,4C,00,CF,02,20,40,47,00, 00,00,0A,00,00,00,00,01,00,35,03, 19,01,11,04,00,35,0C,35,03,19,00, 01,35,05,19,03,00,08,02,05,00,35, 03,19,02,10,06,00,35,09,09,6E,65, 09,6A,00,09,00,01,09,00,35,08,35, 06,19,01,11,09,00,01,00,01,25,05, 43,4F,4D,32,00,03	Tx: Cmd: Store SDP Record, Local Port: 02, Authentication: 20, Encryption: 40, SdpRecord: 00000A000000001003503190111040035 0C350319000135051903000802050035 031902100600350909 6E65096A0009000 1090035083506190111090001000125 05434F4D3200.
00 / 01	RX	Confirm	02,43,31,02,00,76,00,01,03	Rx: Event: Store SDP Record, Status: 00, Identifier: 01

4.2.3.2 Discover Devices

To get the devices BD_Addresses an Inquiry has to be started.

Table 4-51. Device Discovery

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
	TX	Request	02,52,00,03,00,55,0A,00,00,03	Tx: Cmd: Inquiry, Length: 0A, NumResponces: 00, Mode: 00
00 / 01	RX	Indicator	02,69,01,09,00,73,12,34,56,78,9A, BC,00,00,00,03	Rx: Event: Device Found, BdAddr: 123456789ABC, DeviceClass: 000000
00 / 01	RX	Indicator	02,69,01,09,00,73,34,56,78,9A,BC, 12,00,00,00,03	Rx: Event: Device Found, BdAddr: 3456789ABC12, DeviceClass: 000000
00 / 01	RX	Indicator	02,69,01,09,00,73,56,78,9A,BC,12, 34,00,00,00,03	Rx: Event: Device Found, BdAddr: 56789ABC1234, DeviceClass: 000000
00 / 01	RX	Confirm	02,43,00,01,00,44,00,03	Rx: Event: Inquiry, Status: 00

4.2.3.3 Get Remote RFComm Ports

To get the remote Comports, for each of the links a SDAP Browse has to be done.

Table 4-52. Get Remote RFComm Port of first device

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
	TX	Request	02,52,32,06,00,8A,**12,34,56,78,9A, BC**,03	Tx: Cmd: SDAP Connect, BdAddr: 123456789ABC

Table 4-52. Get Remote RFComm Port of first device

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
01	RX	Indicator	02,69,50,07,00,C0,**12,34,56,78,9A, BC**,00,03	Rx: Event: ACL Established, BdAddr: 123456789ABC, Status: 00
00 / 01	RX	Confirm	02,43,32,01,00,76,00,03	Rx: Event: SDAP Connect, Status: 00
	TX	Request	02,52,35,02,00,89,**01,11**,03	Tx: Cmd: Service Browse, Browse Group ID: 0111
00 / 01	RX	Confirm	02,43,35,1E,00,96,00,01,02,10,01, 11,**01**,16,42,6C,75,65,74,6F,6F,74, 68,20,53,65,72,69,61,6C,20,50,6F, 72,74,00,03	Rx: Event: Service Browse, Status: 00, Browse Group ID: 0210, Service ID: 0111, PortNo: 01, Service Name: Bluetooth Serial Port.
	TX	Request	02,52,33,00,00,85,03	Tx: Cmd: SDAP Disconnect
00 / 01	RX	Confirm	02,43,33,01,00,77,00,03	Rx: Event: SDAP Disconnect, Status: 00
00	RX	Indicator	02,69,51,07,00,C1,**12,34,56,78,9A, BC**,16,03	Rx: Event: ACL Terminated, BdAddr: 123456789ABC, Reason: 16

Table 4-53. Get Remote RFComm Port of the second device

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
	TX	Request	02,52,32,06,00,8A,**34,56,78,9A,BC ,12**,03	Tx: Cmd: SDAP Connect, BdAddr: 3456789ABC12
00	RX	Indicator	02,69,50,07,00,C0,**34,56,78,9A,BC ,12**,00,03	Rx: Event: ACL Established, BdAddr: 3456789ABC12, Status: 00
00 / 01	RX	Confirm	02,43,32,01,00,76,00,03	Rx: Event: SDAP Connect, Status: 00
00 / 01	RX	Request	02,52,35,02,00,89,01,11,03	Rx: Event: Service Browse, Browse Group ID: 0111
00 / 01	RX	Confirm	02,43,35,1E,00,96,00,01,02,10,01, 11,**01**,16,42,6C,75,65,74,6F,6F,74, 68,20,53,65,72,69,61,6C,20,50,6F, 72,74,00,03	Rx: Event: Service Browse, Status: 00, Browse Group ID: 0210, Service ID: 0111, PortNo: 01, Service Name: Bluetooth Serial Port.
	TX	Request	02,52,33,00,00,85,03	Tx: Cmd: SDAP Disconnect
00 / 01	RX	Confirm	02,43,33,01,00,77,00,03	Rx: Event: SDAP Disconnect, Status: 00
00	RX	Indicator	02,69,51,07,00,C1,**34,56,78,9A,BC ,12**,16,03	Rx: Event: ACL Terminated, BdAddr: 56789ABC1234, Reason: 16

Table 4-54. Get Remote RFComm Port of the third device

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
	TX	Request	02,52,32,06,00,8A,**56,78,9A,BC,12 ,34**,03	Tx: Cmd: SDAP Connect, BdAddr: 56789ABC1234
00	RX	Indicator	02,69,50,07,00,C0,**56,78,9A,BC,12 ,34**,00,03	Rx: Event: ACL Established, BdAddr: 56789ABC1234, Status: 00
00 / 01	RX	Confirm	02,43,32,01,00,76,00,03	Rx: Event: SDAP Connect, Status: 00
00 / 01	RX	Request	02,52,35,02,00,89,01,11,03	Rx: Event: Service Browse, Browse Group ID: 0111

Table 4-54. Get Remote RFComm Port of the third device

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
00 / 01	RX	Confirm	02,43,35,1E,00,96,00,01,02,10,01,11,**01**,16,42,6C,75,65,74,6F,6F,74,68,20,53,65,72,69,61,6C,20,50,6F,72,74,00,03	Rx: Event: Service Browse, Status: 00, Browse Group ID: 0210, Service ID: 0111, PortNo: 01, Service Name: Bluetooth Serial Port.
00 / 01	TX	Request	02,52,33,00,00,85,03	Tx: Cmd: SDAP Disconnect
00 / 01	RX	Confirm	02,43,33,01,00,77,00,03	Rx: Event: SDAP Disconnect, Status: 00
00	RX	Indicator	02,69,51,07,00,C1,**56,78,9A,BC,12,34**,16,03	Rx: Event: ACL Terminated, BdAddr: 56789ABC1234, Reason: 16

4.2.3.4 Establish Links

The link establishment is always performed with the same command “Establish Link”, just referring to different local ports.

Table 4-55. Establish Links to all three slaves

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
	TX	Request	02,52,0A,08,00,64,**01,12,34,56,78,9A,BC,01**,03	Tx: Cmd: Establish Link, Local Port: 01, BdAddr: 123456789ABC, Remote Port Number: 01
00 / 01	RX	Confirm	02,43,0A,02,00,4F,**00,01**,03	Rx: Event: Establish Link, Status: 00, Local Port: 01
00	RX	Indicator	02,69,50,07,00,C0,**12,34,56,78,9A,BC,00**,03	Rx: Event: ACL Established, BdAddr: 123456789ABC, Status: 00
00 / 01	RX	Indicator	02,69,3E,04,00,AB,**01,0C,00,00**,03	Rx: Event: Port Status Changed, Local Port: 01, PortStatus: 0C, Break Length: 0000
00 / 01	RX	Indicator	02,69,0B,09,00,7D,**00,12,34,56,78,9A,BC,01,01**,03	Rx: Event: Link Established, Status: 00, BdAddr: 123456789ABC, Local Port: 01, Remote Port Number: 01
	TX	Request	02,52,0A,08,00,64,**02,34,56,78,9A,BC,12,01**,03	Tx: Cmd: Establish Link, Local Port: 02, BdAddr: 123456789ABC, Remote Port Number: 01
00 / 01	RX	Confirm	02,43,0A,02,00,4F,**00,02**,03	Rx: Event: Establish Link, Status: 00, Local Port: 02
00	RX	Indicator	02,69,50,07,00,C0,**34,56,78,9A,BC,12,00**,03	Rx: Event: ACL Established, BdAddr: 3456789ABC12, Status: 00
00 / 01	RX	Indicator	02,69,3E,04,00,AB,**02,0C,00,00**,03	Rx: Event: Port Status Changed, Local Port: 02, PortStatus: 0C, Break Length: 0000
00 / 01	RX	Indicator	02,69,0B,09,00,7D,**00,34,56,78,9A,BC,12,02,01**,03	Rx: Event: Link Established, Status: 00, BdAddr: 3456789ABC12, Local Port: 02, Remote Port Number: 01
	TX	Request	02,52,0A,08,00,64,**03,56,78,9A,BC,12,34,01**,03	Tx: Cmd: Establish Link, Local Port: 03, BdAddr: 56789ABC1234, Remote Port Number: 01
00 / 01	RX	Confirm	02,43,0A,02,00,4F,00,01,03	Rx: Event: Establish Link, Status: 00, Local Port: 03
00	RX	Indicator	02,69,50,07,00,C0,**56,78,9A,BC,12,34,00**,03	Rx: Event: ACL Established, BdAddr: 56789ABC1234, Status: 00
00 / 01	RX	Indicator	02,69,3E,04,00,AB,**03,0C,00,00**,03	Rx: Event: Port Status Changed, Local Port: 03, PortStatus: 0C, Break Length: 0000

Table 4-55. Establish Links to all three slaves

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
00 / 01	RX	Indicator	02,69,0B,09,00,7D,**00,56,78,9A,BC,12,34,03,01,**03	Rx: Event: Link Established, Status: 00, BdAddr: 56789ABC1234, Local Port: 03, Remote Port Number: 01

4.3 DEFAULT CONNECTIONS

The LMX9830 offers a default connection procedure which allows to store up to seven devices into a database-like system within the device. The stored connections are either connected after a reset or by sending the “Connect default connections” command. This allows an easy cable replacement setup but can also be used for automatic multipoint operations.

The successful or failed link establishment will be confirmed for each stored device.

If set to Operation Mode “Automatic” the LMX9830 after reset or boot-up will try to connect to each of those devices 3 times then switch to the next (see Section 2.2.2 "Operation States" on page 21).

Each connection storage includes the local and remote RFComm port, the BD_Addr and a transparent flag, which, if set, forces the device to switch to Transparent Mode after link establishment.

Commands available:

- “Store Default Connection” (page 147)
- “Connect to Default Connection” (page 148)
- “Get List of Default Connections” (page 148)
- “Delete Default Connection” (page 149)

NOTE: For multiple connections please make sure that the NVS parameter “Ports to open” (See"Set Ports To Open" (6.2.12.1)) is configured correctly to have the appropriate RFComm port instances initialized.

4.4 LOW POWER OPERATION

4.4.1 Hardware Power Management

The LMX9830 power management is a combination of the firmware and the hardware supported low power modes. Depending on the system activity it decides to switch off as many hardware blocks as possible to reduce the current consumption.

4.4.1.1 Low Power Modes

The LMX9830 power management can be divided into six modes, which depend on the activity level of the UART interface and the bluetooth radio.

- UART Interface activity
  - — Enabled: UART fully active, RTS and CTS used for flow control (hardware handshaking)
  - — Disabled: UART disabled, RTS/CTS used for "Wake up functionality" (4.4.1.3)
- Bluetooth radio activity
  - — Active Link(s): Bluetooth radio and baseband handling active link, "Bluetooth Low Power Modes" (4.4.2) can be used to reduce power consumption on radio.
  - — Page/Inquiry Scanning: LMX9830 only scanning, discoverable/connectable for other devices, no active link
  - — No Radio activity: no active link and scanning switched off

Table 4-56. Low Power Modes

UART / Radio	No radio activity	Page/Inquiry scanning	Active Link(s)
Disabled	PM0	PM2	PM4
Enabled	PM1	PM3	PM5

- PM0:
  - Lowest power consumption, Sleep Mode
  - UART disabled (using "Disable Transport Layer" (6.2.9.1)), "Wake up functionality" (4.4.1.3) enabled
  - Scanning disabled (using "Set Scan Mode - Discoverability / Connectability" (6.2.14.6) parameters 0x00,0x00), Device not discoverable/connectable for other devices
  - No active bluetooth link
- PM1:
  - UART enabled, device listening to commands ("Command Mode" (1.3.2.1))
  - Scanning disabled (using "Set Scan Mode - Discoverability / Connectability" (6.2.14.6) parameters 0x00,0x00), Device not discoverable/connectable for other devices
  - No active bluetooth link
- PM2:
  - Typical standby mode
  - UART disabled (using "Disable Transport Layer" (6.2.9.1)), "Wake up functionality" (4.4.1.3) enabled
  - Scanning enabled (using "Set Scan Mode - Discoverability / Connectability" (6.2.14.6)), Device discoverable/connectable for other devices
  - No active bluetooth link, LMX9830 waking up host on incoming link
- PM3:
  - Default mode after boot-up
  - UART enabled, device listening to commands ("Command Mode" (1.3.2.1))
  - Scanning enabled (using "Set Scan Mode - Discoverability / Connectability" (6.2.14.6)), Device discoverable/connectable for other devices
  - No active bluetooth link
- PM4:
  - UART disabled (using "Disable Transport Layer" (6.2.9.1)), "Wake up functionality" (4.4.1.3) enabled
  - Scanning has no influence on power management
  - Active bluetooth link, LMX9830 waking up host on another incoming link or incoming data
  - Power consumption can be reduced by "Bluetooth Low Power Modes" (4.4.2)
- PM5:
  - Full activity mode
  - UART enabled, device listening to commands ("Command Mode" (1.3.2.1)) or in "Transparent Mode" (1.3.2.2)
  - Scanning has no influence on power management
  - Active bluetooth link(s)
  - Power consumption can be reduced by "Bluetooth Low Power Modes" (4.4.2)

PM0 is lowest power, PM5 can be seen as highest power consumption mode. As indicated, the host is able to influence the power consumption by either switching off scanning using "Set Scan Mode - Discoverability / Connectability" (6.2.14.6) or by disabling the UART, using "Disable Transport Layer" (6.2.9.1) to enable the "Wake up functionality" (4.4.1.3).

Default mode after boot-up is usually PM3, in which the UART is enabled and the device is discoverable and connectable. In case the device is connected from another device or actively establishes a link, it switches to PM5.

In order to save power in a waiting situation, the host might decide to put the device from PM3 to PM2, using the "Disable Transport Layer" (6.2.9.1) command. The device will still be available for connections and will wake up the host on an incoming link.

If the LMX9830 is already engaged in a link, the host can decide to use "Disable Transport Layer" (6.2.9.1) to enable "Wake up functionality" (4.4.1.3). This would allow the host to power down and just wait for incoming data even on an active link.

To reach a low power sleep mode (starting from PM3), the host needs to first shut down the radio activity by disabling the scanning using "Set Scan Mode - Discoverability / Connectability" (6.2.14.6). This forces the device switching to PM1. After that the host can use "Disable Transport Layer" (6.2.9.1) to switch off also the UART activity. The LMX9830 will activate the "Wake up functionality" (4.4.1.3) on the UART and go to a very low power mode.

Figure 4-34 on page 62 gives an overview of all possible transitions between the different power modes and what action triggers the change.

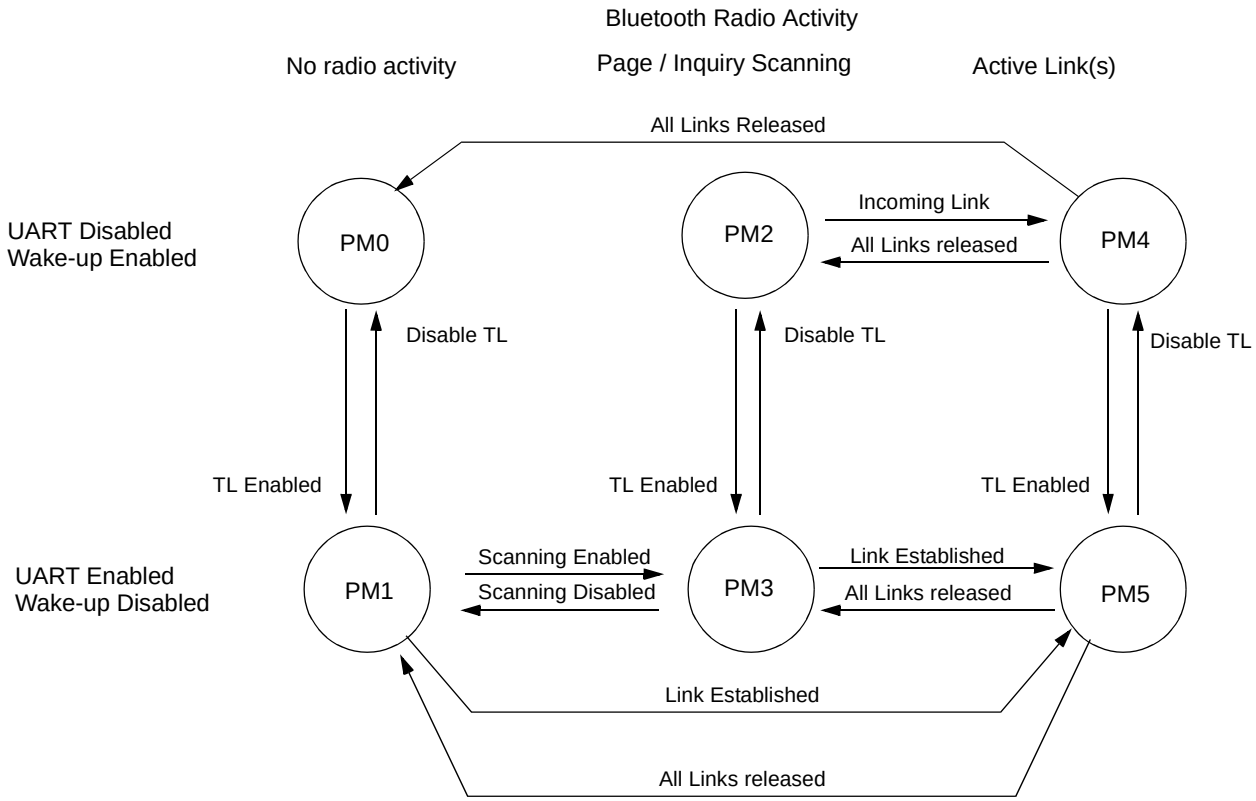


Figure 4-34. Transition between different Hardware Power Modes

4.4.1.2 Enhanced Power Management

The LMX9830 is able to reduce the power in the waiting state PM2, using an external 32.768khz crystal. If such a crystal is available and connected, the baseband is able to shut down the radio completely between the scanning intervals and with this to drop the power consumption to a few hundred Micro Ampere.

The usage of the 32.768 khz is controlled by the Enhanced PowerManagement (PMM) bit in NVS. By default PMM is disabled, so the 32.768kHz is not used even if connected. The PMM bit can be enabled by the "Write NVS" (6.2.23.2) command, configuring bit 1 of the system parameter at address 0x0062 to 0.

4.4.1.3 Wake up functionality

In certain applications the LMX9830 will be used most of the time in a waiting status, meaning it is waiting for being connected or listening to commands. To reduce power consumption of the system, the LMX9830 supports a specific Wake up functionality.

The LMX9830 supports to disable the UART transport layer (command "Disable Transport Layer" (6.2.9.1**)** to switch off the command interpreter and all hardware components not needed for the current operation. The interface can be reactivated again by either side by using hardware pins.

The LMX9830 uses the RTS signal pin to wake up the host. The RTS / CTS signals are connected in a NULL-Modem fashion, meaning that RTS on the Host is connected to CTS on the LMX9830 and vice versa. Therefore the host would need to be able to monitor its CTS input or has to use a separate hardware pin. In case the LMX9830 has to be triggered by the host, the RTS pin is used as the hardware Wake-Up signal.

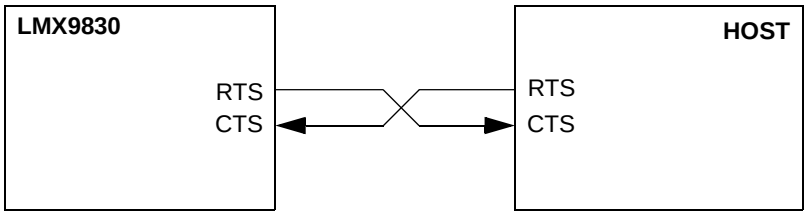


Figure 4-35. Host Wake Up hardware connections

4.4.1.3.1 Disabling the transport layer

The transport layer can be disabled in any operation state. The LMX9830 will try to wake up the host as soon as the interface would be needed again.

To disable the transport layer the following command has to be used:

- "Disable Transport Layer" (page 156)

The command will be confirmed with the standard "Disable Transport Layer" Confirm event. After the command has been sent, the LMX9830 will switch the functionality of the RTS and CTS pins from normal hardware flowcontrol to the wake-up functionality.

Both the Host and the LMX9830 shall set RTS=1 since they may be in a sleep mode and thus both are "Not Ready to Receive". The HW Wake-Up signal is then defined as a falling edge on the CTS input i.e. a device wakes up the other device by asserting its own "Ready to Receive" output (i.e. setting RTS=0).

If the LMX9830 redefines the CTS input from "flow-control" input to "wake-up" input there will be a short period of time during which the signalling is ambiguous. To avoid this, delays are introduced as illustrated in Figure 4-36.

To guarantee no loss of data the UART shifts out the last byte of the confirm event and the Host redefines its RTS output when it has received the last byte of the "Disable transport layer event".

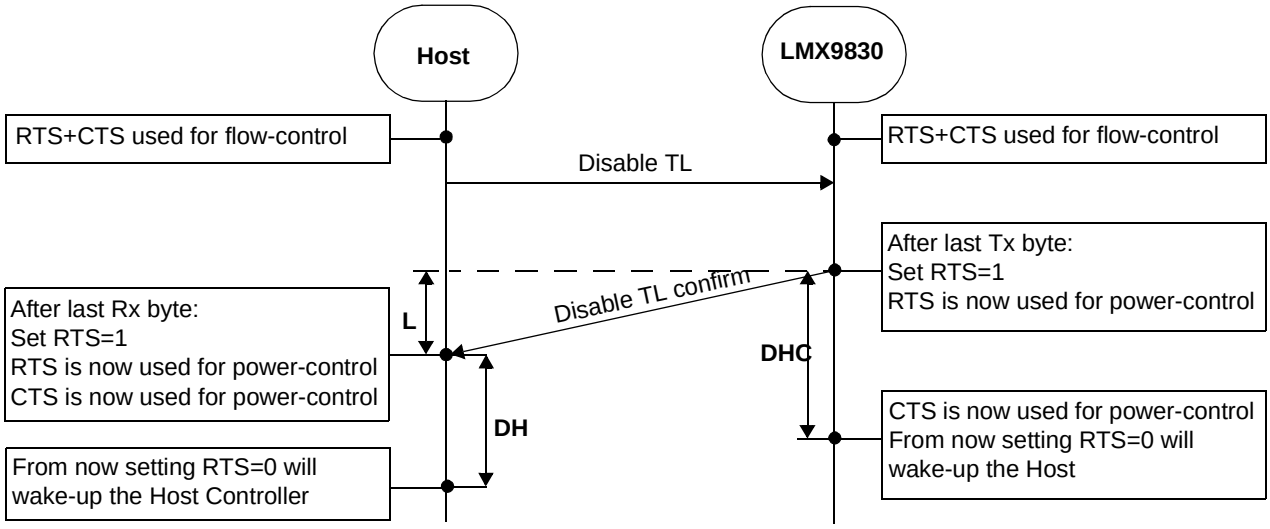


Figure 4-36. Disable Transport Layer Timing

L is the time period in which the RTS/CTS signalling is ambiguous.

DHC is the time period the LMX9830 must delay redefining CTS to Wake-Up input.

DH is the time period the Host must wait before attempting to send a Wake-Up signal to the LMX9830 by setting RTS = 0.

In order to make the mechanism work the following relations must be true:

Lmax ≥ **L** ≥ **Lmin**

DHC ≥ **Lmax**

DH ≥ **DHC - Lmin**

4.4.1.3.2 LMX9830 wakes up host

If the LMX9830 needs to send data to the host it must first make sure that the UART transport layer is enabled. If UART is disabled the LMX9830 assumes that the host is sleeping and starts the wake-up by setting RTS to 0. To be able to react on that Wake-up, the host has to monitor the CTS pin.

On receiving the CTS falling edge the host has to wake up its UART interface and switches the RTS/CTS pin functionality back to normal hardware handshake. As soon as RTS is 0, the LMX9830 will confirm the wake-up by sending the TL Enabled Event and start sending the pending events that triggered the wake-up.

Please see Figure 4-37 for the complete process.

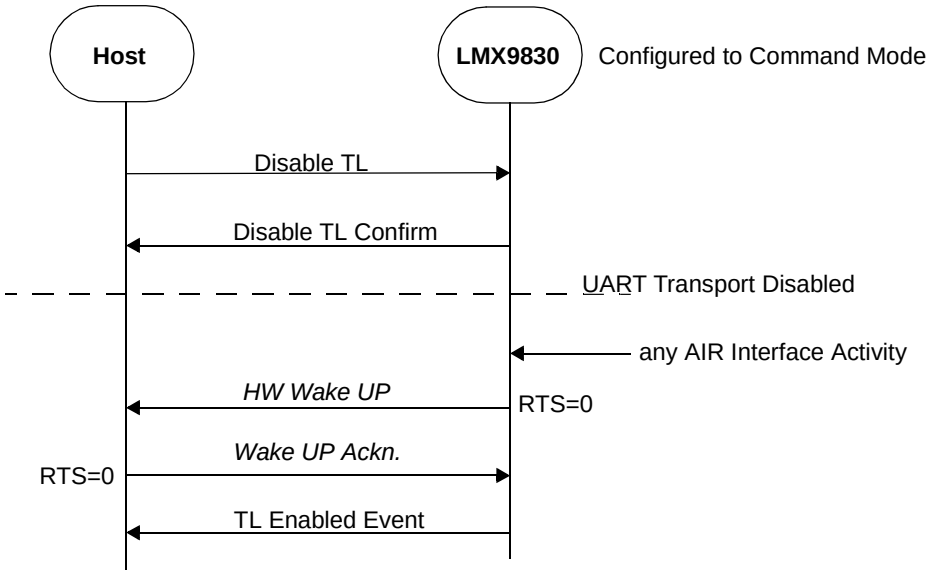


Figure 4-37. LMX9830 waking up the host in command mode

Note:

Even though the LMX9830 signals RTS=0 (i.e. “Ready to Receive”) the Host may not send any data or commands to the LMX9830 before it has received the “Transport Layer Enabled” event.

The event will always be received as soon as the LMX9830 re-enables the UART interface again. One exception is if the device is in “Idle Automatic” state. If the LMX9830 is operating in default idle automatic state waiting for being connected and the transport layer was switched off, it will not send the TL Enabled Event. As soon as it gets connected the LMX9830 will request the wake up and directly send the “Link Established” after successful connection establishment.

Please see Figure 4-38 for that specific case.

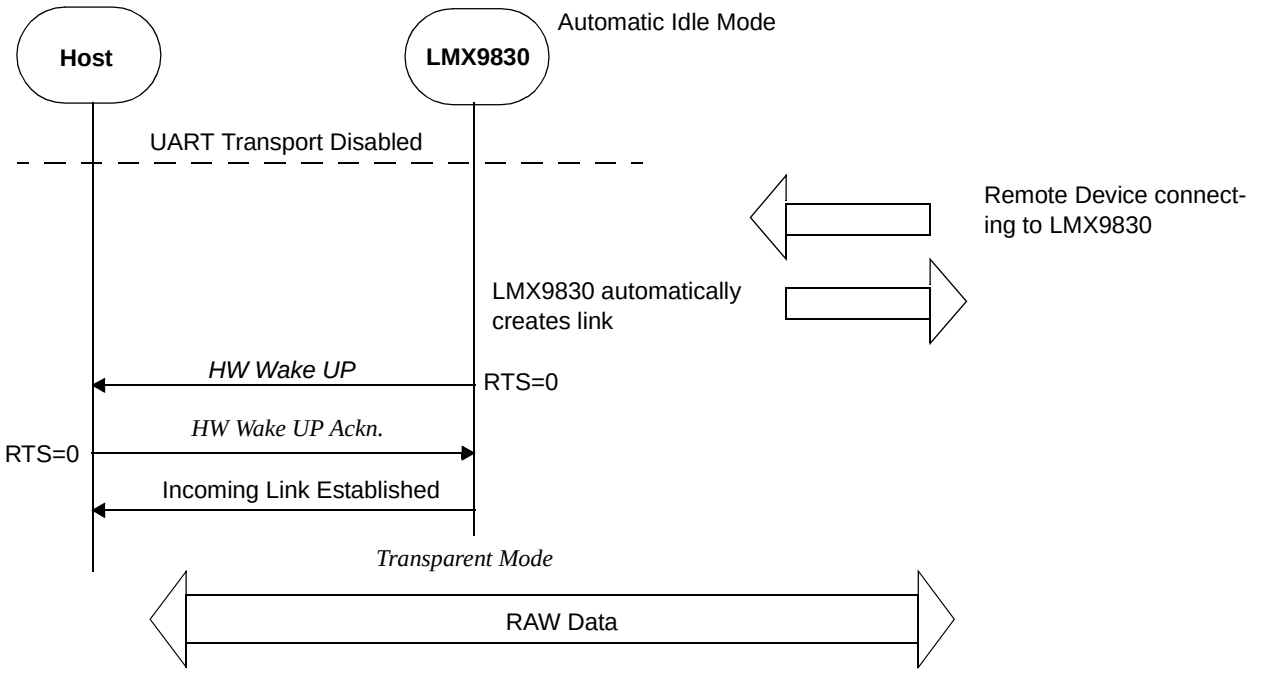


Figure 4-38. No TL enabled event after Automatic Idle

4.4.1.3.3 Host wakes up LMX9830

If the host needs to send data or commands to the LMX9830 it must first make sure that the UART transport layer is enabled. If UART is disabled the host must assume that the host is sleeping and starts the wake-up by setting RTS to 0. The LMX9830 will wake up to the falling edge of its CTS pin.

When the LMX9830 detects the Wake-Up signal it activates the UART HW and acknowledges the Wake-Up signal by sending a “Transport Layer Enable” event. When the Host has received the “Transport Layer Enable” event, the LMX9830 is ready to receive commands.

This process is shown in the following figure

.

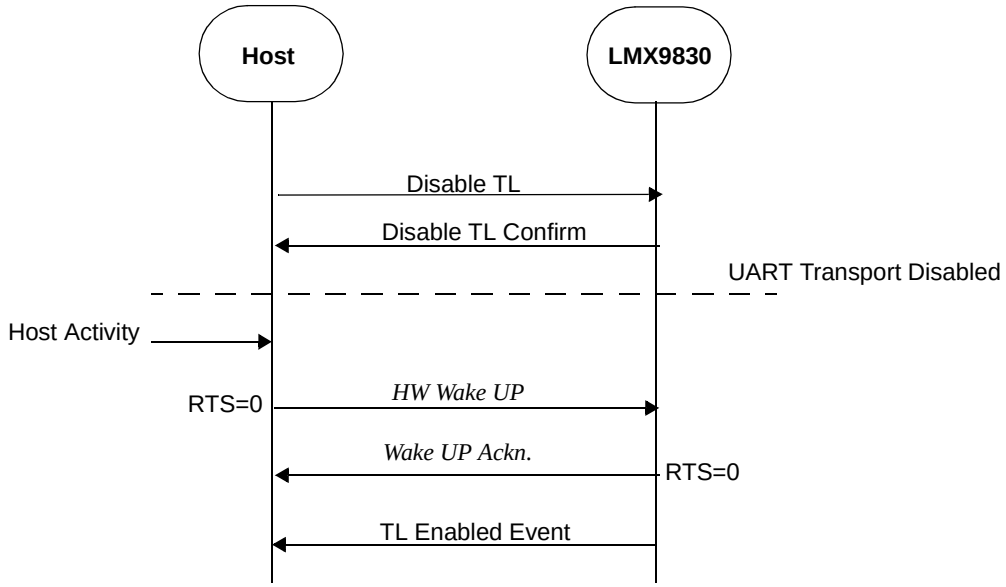


Figure 4-39. Host waking up LMX9830

NOTE:

Even though the LMX9830 signals RTS=0 (i.e. “Ready to Receive”) the Host may not send any commands to the LMX9830 before receiving the Transport Layer Enabled event.

4.4.2 Bluetooth Low Power Modes

The Bluetooth standard offers several different Low Power Modes to reduce the “active” time for the transceiver. The following modes are currently defined within the bluetooth specification:

- Sniff Mode
  - — Master and Slave device arrange certain interval to talk to each other. In between the “Sniff Slots” the slave does not listen to Master transmissions and is able to switch off the radio.
  - — Slave is still an active member of the piconet.
  - — Data transmissions reduced to the Sniff slots.
- Hold Mode
  - — Master and Slave device agree to stop transmission for a specific time.
  - — Slave is still an active member of the piconet
  - — Slave and Master are not able to exchange data within the hold time.
- Park Mode
  - — Slave is not part of active piconet anymore
  - — Slave kept synchronized by “Beacons”
  - — Slave has to be unparked before data can be transmitted between devices again.

The LMX9830 supports all Low Power Modes.

One important parameter for switching to a Low Power Mode is the “Link Policy”. The Link policy defines, which modes are accepted for this specific link.

4.4.2.1 Link Policy

The Link Policy defines which Low Power Modes are allowed for one specific link. The settings indicate, which modes are allowed on the local device for this specific link. To use one of the features, both devices in the link have to allow the mode.

The following features can be switched on or off:

- Master/Slave Switch
- Sniff Mode
- Hold Mode
- Park Mode

4.4.2.1.1 Default Link Policy Setting

The LMX9830 includes the parameter “Default Link Policy” in NVS, which automatically is set for all incoming and outgoing links.

On default, the link policy is configured to accept Master/Slave switch as well as all Low Power Modes.

The Default Link Policy can be checked and changed by the following commands:

- “Set Default Link Policy” (page 150)
- “Get Default Link Policy” (page 150)

4.4.2.1.2 Setting Link Policy in an active link

In an active link, each of the settings can be switched on or off by the “Set Link Policy” Command.

On default the LMX9830 will use the “Default Link Policy” stored in NVS. The Link Policy of an active link can be checked and changed by the following commands:

- “Set Link Policy” (page 151)
- “Get Link Policy” (page 151)

4.4.2.2 Sniff Mode

The Sniff Mode allows to reduce the transmission slots to a specific interval, defined by parameters sent with the command. In standard operation, the slave is listening continuously to the master. This causes high power consumption on the slave. By setting a Sniff interval, the Slave does not expect packages from the Master for a specific time and therefore can switch of the radio receiver to reduce power or is able to actively talk to other devices.

The “Enter Sniff Mode” command has the following parameters:

- BD_Addr
  — The BD_Addr of the remote device
- Maximal Sniff Interval
  — The maximum interval of slots the two devices are not exchanging packages.
  — Range: 0x0006 to 0x1000 (3.725ms to 2.56s)
- Minimum Sniff Interval
  — The minimum interval of slots the two devices are not exchanging packages
  — Range: 0x0006 to 0x1000 (3.725ms to 2.56s)
- Sniff Attempts
  — Number of slots the slave has to listen to incoming packages from the master, beginning at the sniff slot.
  — Range: 0x0001 to 0x07FF
- Sniff Timeout
  — Number of slots the slave has to listen even if he still receives packages for itself.
  — Range: 0x0000 to 0x0028

As it could be possible that master or slave has to manage more than one links, the parameters give a range of sniff slots in which the “real” sniff timing will be.

The devices will agree on the parameter and confirm the mode change by an indicator (“Power Save Mode Changed” (page 155)).

After this the Sniff Mode will be active until it is released by the “Exit Sniff Mode” command.

Commands available:

- “Enter Sniff Mode” (page 152)
- “Exit Sniff Mode” (page 153)

EXAMPLE:

Calculation of Sniff parameters:

The communication should be reduced to send 1 package each 300ms. Minimum should be 50ms.

a) Calculation of Sniff interval

1 time slot: 625µs.

300ms / 625µs = 480 slots = 1E0 hex

=> maximum Sniff interval: 01E0 hex

50ms / 625µs = 80 slots = 50 hex

minimum Sniff Interval: 0050 hex

b) Sniff Attempts and Timeout

The slave starts listening at the sniff slots for Nsniff attempt consecutive receive slots unless a packet is received. After every reception of a packet, the slave continues listening at the subsequent Nsniff timeout or remaining of the receive slots, whichever is greater.

For Nsniff timeout > 0 the slave continues listening as long as it receives packets.

Note that Nsniff attempt =1 and Nsniff timeout =0 cause the slave to listen only at the first sniff slot, irrespective of packets received from the master.

Note that Nsniff attempt = 0 is not allowed.

For the example the following parameters have been successfully tested:

Sniff attempts: 5 slots

Sniff timeout: 3 slot

This means the slave listens for a minimum of 5 slots for packets from the master. After a received package it will listen for three more slots.

NOTE: In case a device has to manage multiple connections, Sniff attempt and Sniff timeout need to be large enough to make sure the devices have enough slots available to send their data. (e.g. 10slots each).

4.4.2.3 Hold Mode

The Hold Mode provides the ability to stop package transmission between two devices for one specific time range.

After that time they start normal transmission again.

The Hold Mode uses the following parameters:

- BD_Addr
  — The BD_Addr of the remote device
- Maximal Hold Interval
  — The maximum number of slots the two devices are not exchanging packages.
- Minimum Hold Interval
  — The minimum number of slots the two devices are not exchanging packages

The Hold Mode will be confirmed on entering and on leaving the Mode with the “Power Save Mode Changed” Indicator.

Commands available:

- “Enter Hold Mode” (page 155)

Please see Section 4.4.2.2 "Sniff Mode" for the calculation of the Hold mode parameters.

4.4.2.4 Park Mode

Park Mode enables devices to completely disconnect from each other and just keep synchronized. In that case a Master has all resources available for setting up other links or managing its current piconet. If necessary, the link to the parked slave can be re-established by a special unpark procedure.

The following parameters are used for Park Mode:

- BD_Addr
  — The BD_Addr of the remote device
- Maximal Beacon Interval
  — Acceptable longest length between beacons.
- Minimum Beacon Interval
  — Shortest length between beacons.

As the clocks of Master and slave have to be kept synchronous, the maximal beacon interval should be not to high, otherwise a reestablishment could fail.

Available commands:

- “Enter Park Mode” (page 153)
- “Exit Park Mode” (page 154)

NOTE: Since broadcast packages are not acknowledged and they are not seen as data traffic, a parked linked will be dropped after the supervision timeout. To keep a parked device connected, it has to be unparked and parked within the supervision timeout. See also Section 4.1.2.6 “Configuring the Default Link Timeout” on page 49 for details on the supervision timeout.

4.5 ESTABLISH AUDIO LINKS

4.5.1 Bluetooth Background

A standard bluetooth link consists of asynchronous connectionless (ACL) links, meaning, data are sent by request from the master to slave. Since the slave has to wait for the master polls to be able to transmit data, these ACL links are not suitable for audio links. For this, the bluetooth specification defines the synchronous connection-oriented (SCO) links., which are used to transport real-time audio data. On SCO links, master and slave communicate on dedicated reserved slots. The frequency on which the master and slave will exchange packages is defined by the package type.

SCO package types:

- HV3:
  - 30bytes per package
  - no FEC
  - Master and slave exchange data every 6 slots; link consumes about 33% of the complete bluetooth bandwidth
  - mostly used since it leaves most flexibility for other bluetooth links.
- HV2:
  - 20bytes per package
  - 2/3 FEC
  - Master and slave exchange data every 4 slots; link consumes 50% of the complete bluetooth bandwidth
- HV1:
  - 10bytes per package
  - 1/3 FEC
  - Master and slave exchange data every 2nd slot; link consumes 100% of the complete bluetooth bandwidth, no other link possible

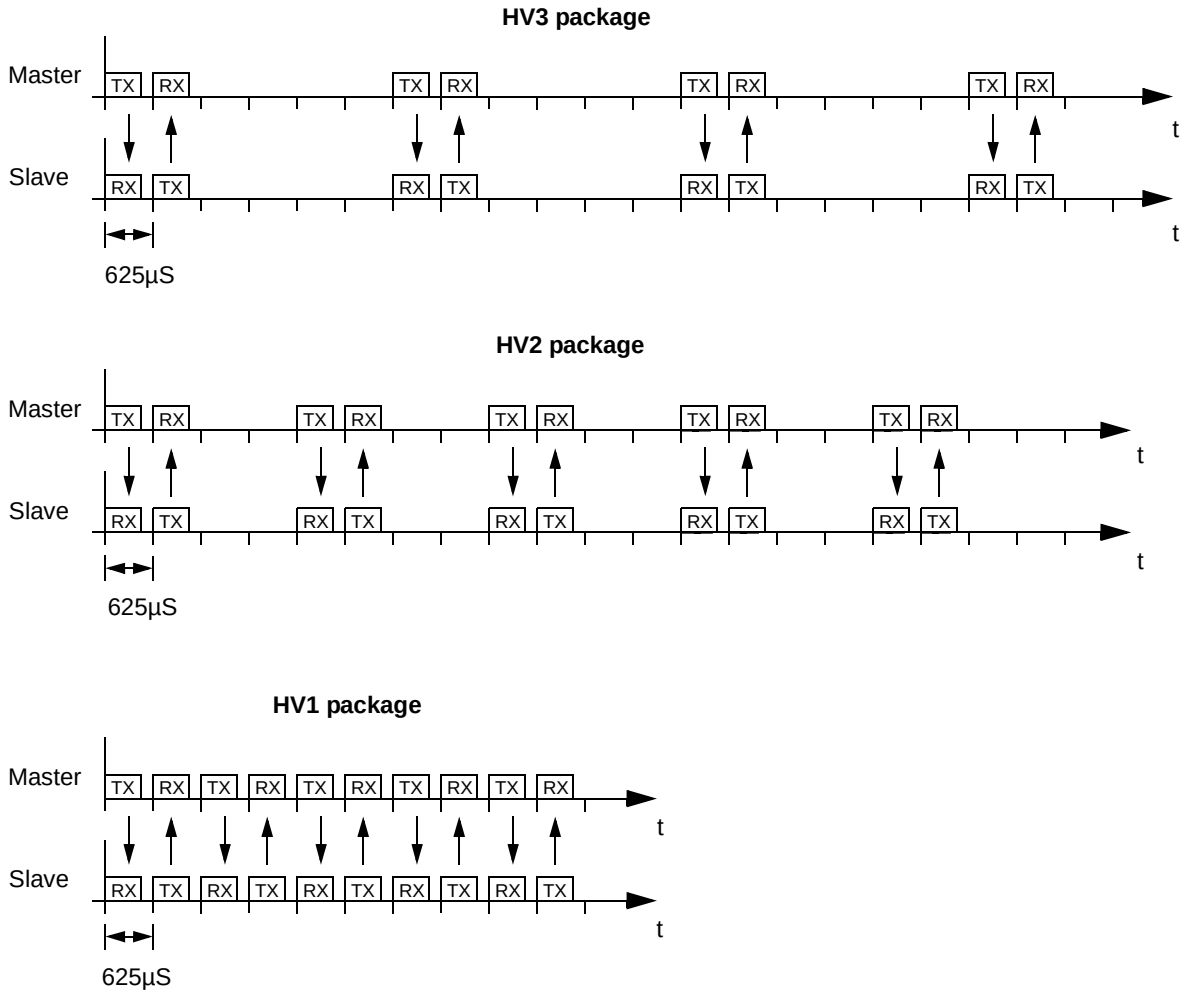


Figure 4-40. SCO link bandwidth requirements

4.5.2 Routing of audio data

As mentioned in Section 1.4 “Audio support” on page 16, the LMX9830 offers the Advanced audio interface (AAI), also called PCM interface, in order to connect an external PCM codec. The codec is used to convert the analog microphone signals into the digital PCM stream, which is then directly routed to an existing bluetooth SCO link. The same way, any incoming SCO data will be directly routed to the PCM interface to be converted into analog loudspeaker signals by the external codec.

Each SCO link is based on a previously established ACL link. The ACL link is used for the standard SPP profile. Because of this it is possible to have simultaneous data and voice transmissions. See Figure 4-41 on page 70 for the split between SCO and ACL/SPP data routing.

NOTE: Simultaneous data transmission is always possible even if SCO package type HV1 is used. Invisible to the user, the bluetooth link will use the mixed data/voice package DV, which allows to transport both data and synchronous audio.

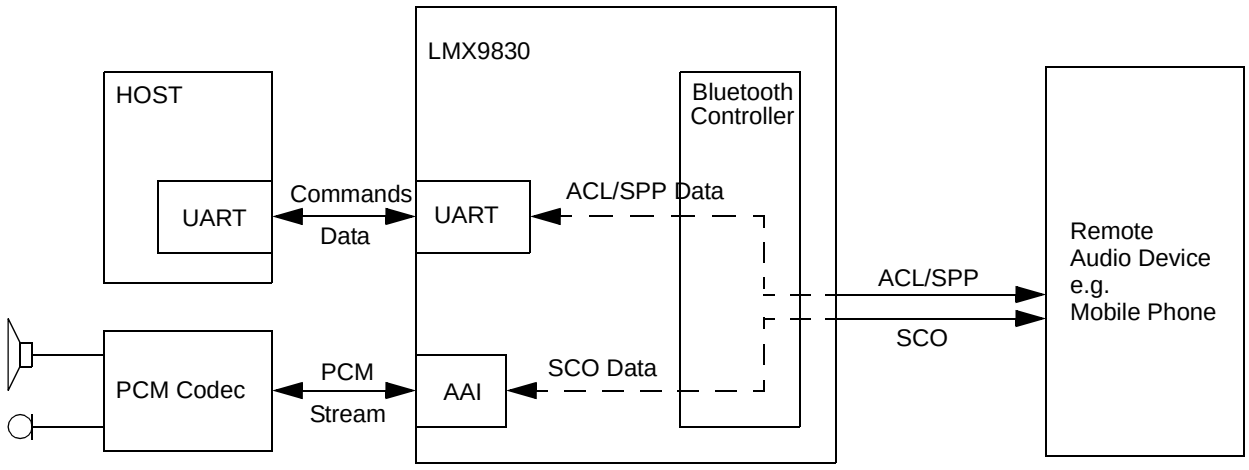


Figure 4-41. SCO and ACL/SPP routing within LMX9830

4.5.3 Audio Link commands

The LMX9830 includes all functionality to establish and accept SCO links and routing the data to the PCM using the settings configured in the NVS. The settings for audio codec and air format can either be configured by default or for after link establishment

4.5.3.1 PCM codec configuration

In order to route the information of the SCO link to the PCM codec in the right format and timing, the firmware needs to know the codec and air format to be used. Those parameters can be configured after link establishment by the following commands:

- "Set audio settings" on page 142
- "Get audio settings" on page 143

This command defines which format should be used at the PCM interface (codec driver or PCM slave) is used as well as the format used on the bluetooth link. This command only sets the parameters for the existing link.

The settings can also be stored in NVS as default for future incoming or outgoing SCO link establishments by the following command:

- "Set Default Audio Settings" on page 138
- "Get Default Audio Settings" on page 138

Please see also Figure 4-42 on page 71 for the influence of both parameters to the blocks of the LMX9830. The codec setting configures the AAI interface timing to be able to interact with the external codec. The air format configures which format will be used on the bluetooth link. Both parameters influence the configuration of the internal CVSD codec, which will take care of the conversion to the appropriate formats.

In case PCM slave is used, these settings have to be stored separately. Please refer to Section 4.5.3.1.2 "Using PCM slave" on page 73 for details.

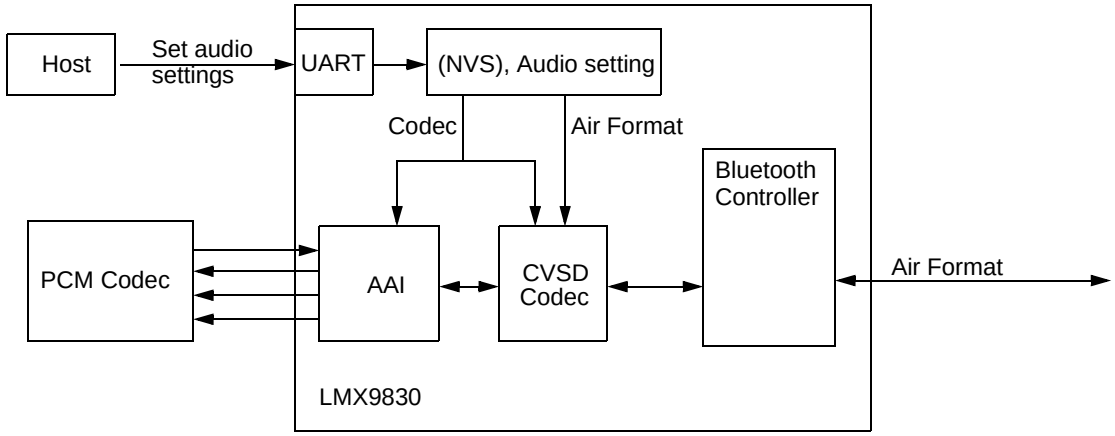


Figure 4-42. Influence of audio settings to LMX9830 blocks and formats

4.5.3.1.1 Using a specific codec driver

The LMX9830 includes two specific drivers, which are optimized for the Motorola MC145483/Winbond W681360 codec or the OKI MSM7717 / Winbond W681310 codec.

The interface settings depend on the external clock used for the LMX9830. The following parameters have to be considered to interface with the AAI in that configurations:

- Bit Clock (Pin SCLK)
- Frame Clock (Pin SFS)
- Data format (13-bit linear, 8-log PCM a-Law)

In general these settings define the signalling on STD and SRD pins for the data.Figure 4-43 shows the principle for the relation between frame sync signal and a data slot. The frame sync signal is typically raised in a frequency of 8kHz. The data are sampled at the bit clock provided by the LMX9830 at pin SCLK.

A complete list of bit clock and format information for all codecs can be found in[1] "Texas Instruments: LMX9830 Datasheet".

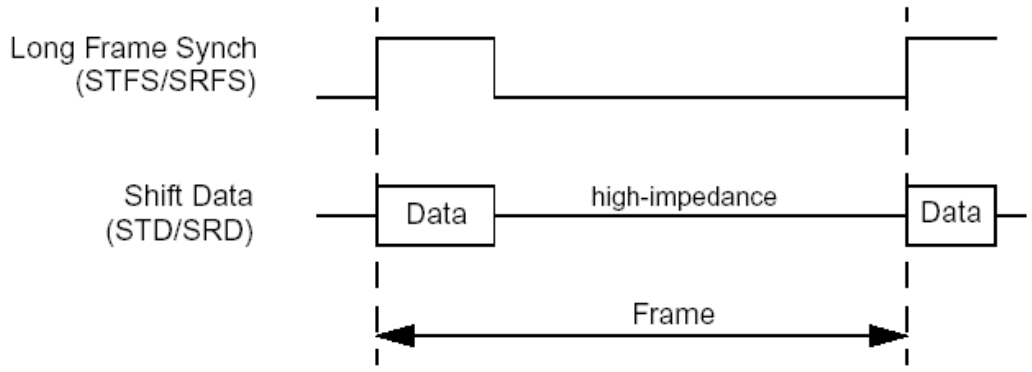


Figure 4-43. AAI generic data framing

Example:

Motorola MC145483 at 13Mhz external clock.

- Bit Clock:520kHz • Frame Clock:8kHz • AAI Frame Sync Pulse: Long

The Motorola codec supports 3-bit audio control at the end of each data word. This information is used by the codec to con trol the volume. The codec requires a 13-bit linear data format.Figure 4-44 on page 72 shows the generic setup of the 16-bit data word for the Motorola codec.

Figure 4-45 on page 72gives the overview and the relation between the actual SFS signal and the data slot. Figure 4-46 on page 73 finally shows the real transmission, with actual bit clock and the 13-bit data word.

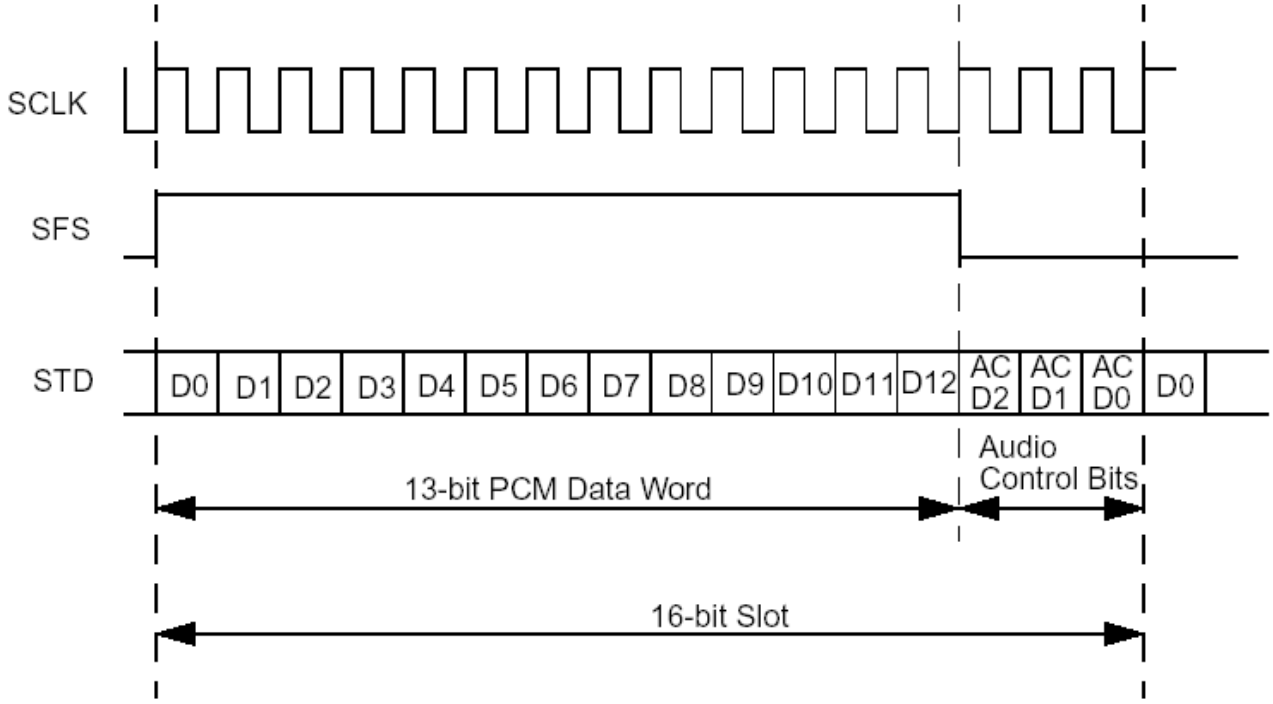


Figure 4-44. Theoretical AAI Interface stream at Motorola settings

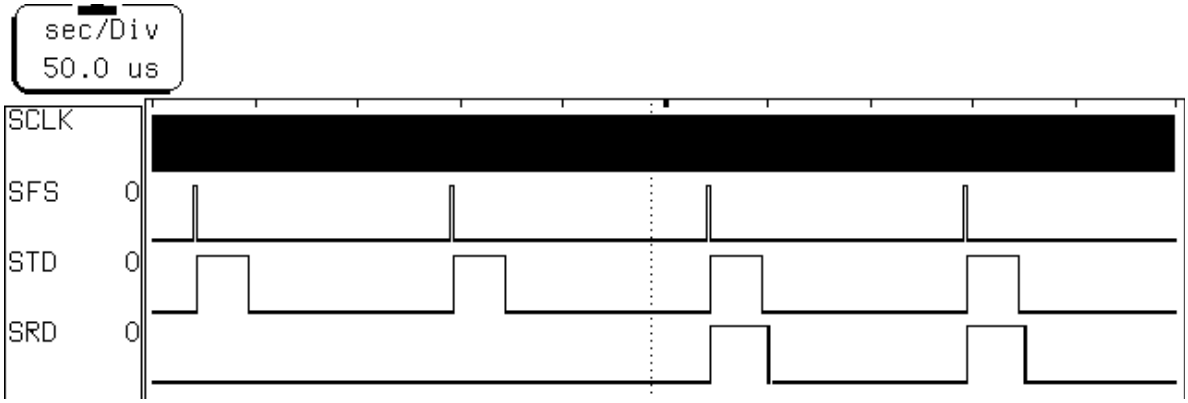


Figure 4-45. AAI stream overview for Motorola codec

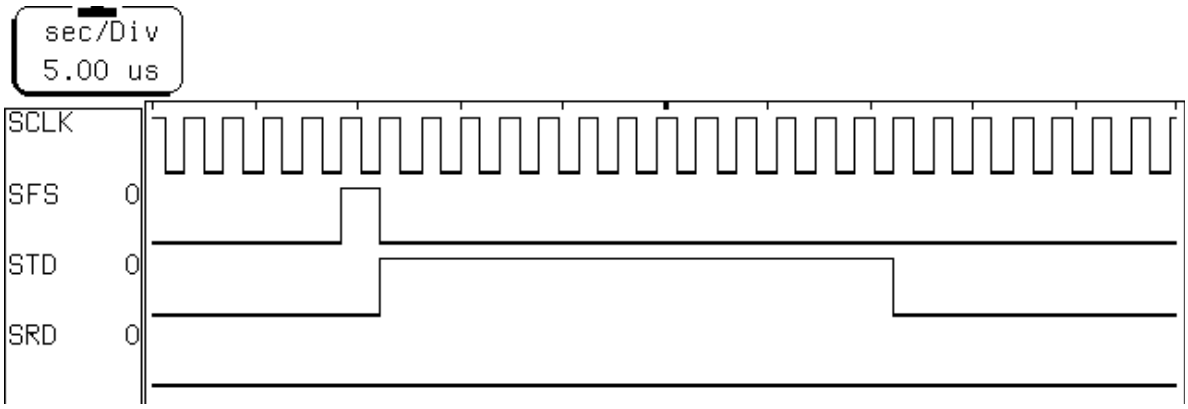


Figure 4-46. AAI slot usage for Motorola with audio control set to 0xFF

4.5.3.1.2 Using PCM slave

Since the LMX9830 is also intended to be used in PDA like devices, the audio interface allows to be driven from an external controller or DSP. This means, the LMX9830 will sample the data from or to the DSP on the Bit clock and Frame clock provided by the DSP.

In order to be able to adjust the internal timing, the LMX9830 audio interface needs to be configured to the specific settings. The settings should be chosen so that the actual frame sync is configured to 8Khz. The bit-clock should not exceed the range of 128khz to 1024khz.

The following parameters need to be configured in order to get a correct PCM stream setup. The settings have to be configured by the command.

a.) Slot Selection
The AAI supports up to four simultaneous slots of which the LMX9830 will extract and send the data. This allows a DSP to drive multiple codecs the same time, differentiating the codecs on the slot. In a single connection slot 1 should be used

b.) Number of slots per frame
This configures how many slots should be transmitted per frame. Maximum is four.

c.) PCM data format
The data format used. The on-chip codec supports

- 8-bit A-law
- 8-bit u-law
- 13/14/15/16 bit linear

d.) Frame sync length
Defines, if a short or a long frame sync signal is used.

Short frame sync pulse:
A short frame sync pulse has the length of one data bit. Using short fame sync pulses, the transfer of the first data bit or the first slot begins at the first positive edge of the shift clock after the frame sync pulse (negative edge).

Long frame sync pulse:
For 8 bit data the frame sync generated will be 6 bits long and for 16 bit data the frame sync can be configured to 13, 14, 15 or 16 bits long. When receiving frame syncs it should be active on the first bit of data and stay active for a least two bits and it needs to go low for at least one clock cycle prior to starting a new frame. Using long frame sync pulses, the transfer of the first word (first slot) begins at the first positive edge of the shift clock after the positive edge of frame sync pulse.

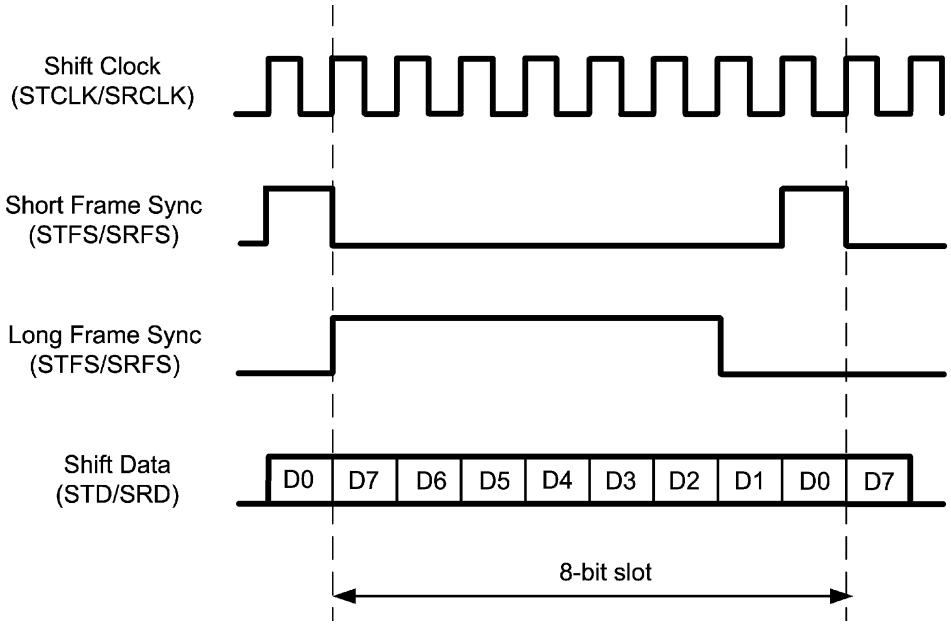


Figure 4-47. Difference between long and short frame sync signal

e.) Data word length
The LMX9830 supports data word length of 8 or 16bit.

f.) Frame sync polarity
The Frame pulse signal can either be used normal or inverted.

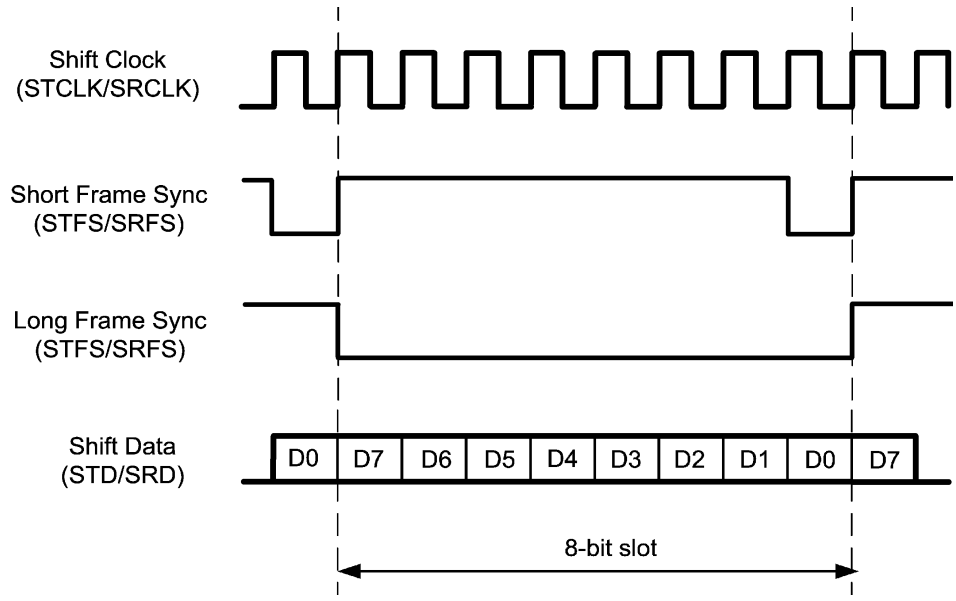


Figure 4-48. Inverted Frame sync pulses

4.5.3.2 SCO Link Establishment

In general a SCO link can only be established on top of an existing ACL link. So in order to establish a SCO link to another device, it first has to establish a standard SPP Link, using the"Establish Link" (6.2.3.1) link command. Once this has been set up, a SCO link can be created to that device, using the following command.

- "Establish SCO Link" on page 141

The command requires the BD_Addr as well as the SCO link package type (HV1, HV2 or HV3) to be used.

4.5.3.3 Accept incoming SCO Links

The LMX9830 will automatically accept incoming SCO link requests and confirm it with the indicator as follows. The firmware will use the default audio settings stored in NVS for the PCM codec driver and Bluetooth air format.

- "SCO Link Established Indicator" on page 141

4.5.3.4 Changing SCO package type

Once a link has been established the devices are able to switch the package type in order to optimize their bandwidth or audio quality requirements. The commands will require the BD_Addr as well as the new packet type to be used.

- "Change SCO Packet Type" on page 144

In case the remote device changed the package type, it will be indicated to the host by the following indicator.

- "Change SCO Link Packet Type Indicator" on page 144

4.5.3.5 Configuring Volume and Microphone level

The LMX9830 commandset offers commands to configure the volume of the loudspeaker output on the PCM codec or to Mute the microphone. The commands can only be executed on existing links.

Please see details in the command section:

- "Mute" on page 145
- "Set Volume" on page 145
- "Get Volume" on page 146

4.5.3.6 Releasing a SCO link

Releasing a SCO link only requires to send the following release command to the LMX9830. The command doesn't require any other parameter to be set, since the LMX9830 can only support one SCO link at a time anyway.

- "Release SCO Link" on page 142

As final confirmation that the SCO link is released, it will be indicated by the following message.

- "SCO Link Released Indicator" on page 142

4.5.4 Example

To establish an audio link to other devices, normally the guidelines for a headset or handsfree profile have to be followed. Because of this please refer to Section 5.0 "Profile Support" for details, how to handle audio links.

4.6 SOFTWARE PATCHES

The LMX9830 firmware is included in ROM and therefore can not be updated. To still be able to correct small errors or to do minor modifications to the software, the LMX9830 offers the so called “Patch“ mechanism, which basically allows to replace small parts of the ROM code. This patch code is held in RAM and needs to be provided during "Initialization Mode" (2.1), in which it is either read from EEPROM or pushed by the host.

4.6.1 The Patch Engine

The LMX9830 has HW support for patching the ROM code by replacing segments (512 bytes) of the ROM with segments of the 3kbyte patch RAM. In order to do patching the firmware must load the patch code into the patch RAM and update the patch look-up registers that define which part of the ROM should be patched. Figure 4-49 on page 76 shows the HW patch engine implemented in the LMX9830.

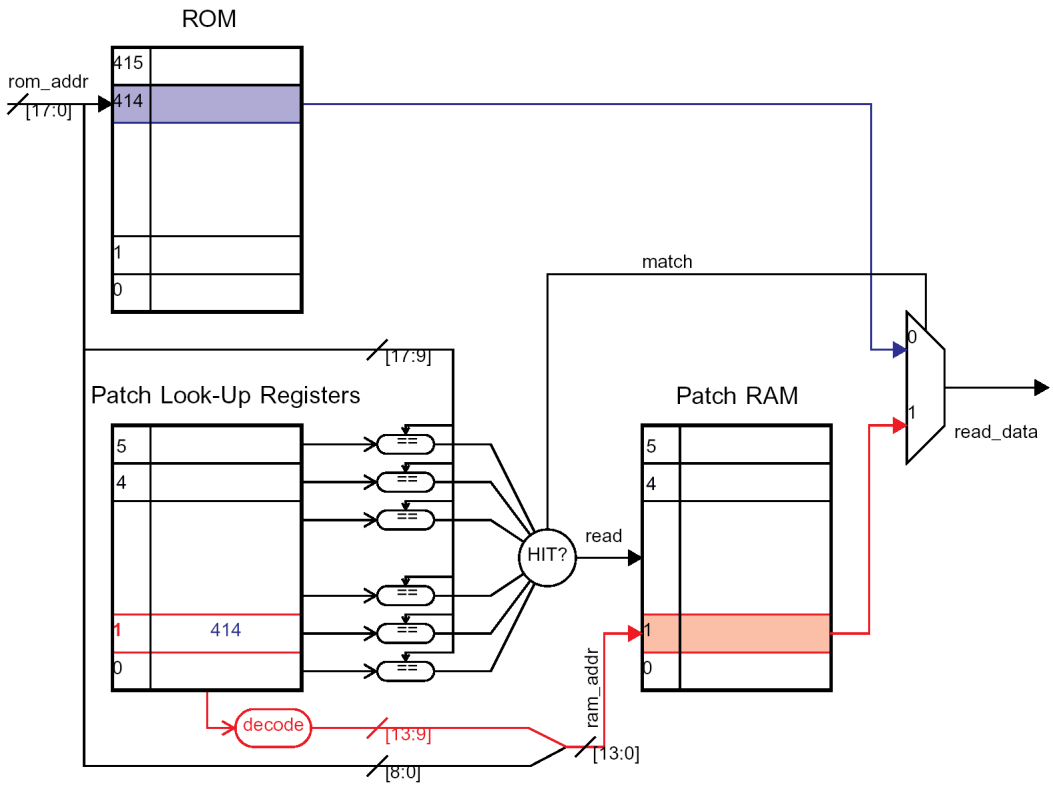


Figure 4-49. Hardware Patch mechanism

4.6.2 Patch format

A patch is basically a piece of code which needs to follow a specific format and is usually provided by Texas Instruments. Table 4-57 on page 77 shows the format of a patch.

The table shows all parts of a patch. Since not all information of the patch-file are actually used by commands or stored in NVS, the table indicates the availability for each location.

- File: the patch file, usually delivered by TI. The file needs to be held on the host, which has to push it either once into EEPROM or during each initialization without EEPROM.
- UART: indicates, which data are pushed by"Write ROM Patch" (6.2.21.1)
  - X1 = All “global” meta data must be included as parameters in the first patch download command sent
  - X2 = All “segment” meta data for one patch segment must be included as parameter in one patch download command (i.e. the data shall not be split in two commands)
- EEPROM: An X indicates, which data are stored in EEPROM

Table 4-57. Patch format at the different locations

Size:	Name:	Description:	Present In		
			File	UART	EEPROM
uint8	PatchValid	Flag (bit0) indicating whether the NVS contains a valid patch.			X
uint32	PatchSize	Size of patch (including PatchSize field but excluding “DescLen” and “Description” fields).	X	X1	X
uint16	FormatRev	Patch format revision, the format specified in this table has revision 0x0100.	X	X1	X
uint16	NumSegs	Number of Patch Code Segments	X	X1	X
uint16	DescLen	Description length	X		
uint8[DescLen]	Description	Textual description of patch	X		
uint32	FWRev	Firmware revision to which this path applies	X	X1	X
uint32	FWPRev	New firmware revision when this path has been applied	X	X1	X
for (I=0;I<NumSegs;I++)					
uint8	Type	Patch type: 0x00 = HW patch. 0x01 = SW patch.	X	X2	X
uint32	RAMAddr	RAM Start address for patch segment.	X	X2	X
if (Type=0x00) //HW patch					
uint32	ROMAddr	ROM Start address for patch	X	X2	X
else //SW patch					
uint32	VarAddr	Address of function pointer variable for patch	X	X2	X
uint16	CodeLen	Code length	X	X2	X
uint16	MetaDataChk	Meta Data Checksum	X	X2	X
uint16	CodeChk	Patch Code Checksum	X	X2	X
uint8[CodeLen]	Code	Patch Code	X	X	X
uint16	Checksum	Checksum (for everything to be present in NVS)	X	X	X

4.6.3 Patch Command Flow

A patch needs to be pushed using the "Write ROM Patch" (6.2.21.1) command. As indicated in Section 4.6.2, the content of the file needs to be pushed in a specific way (X1 in one command, X2 in one command) to ensure the correct storage in NVS and Patch RAM. In case an EEPROM is connected, the patch will automatically be stored.

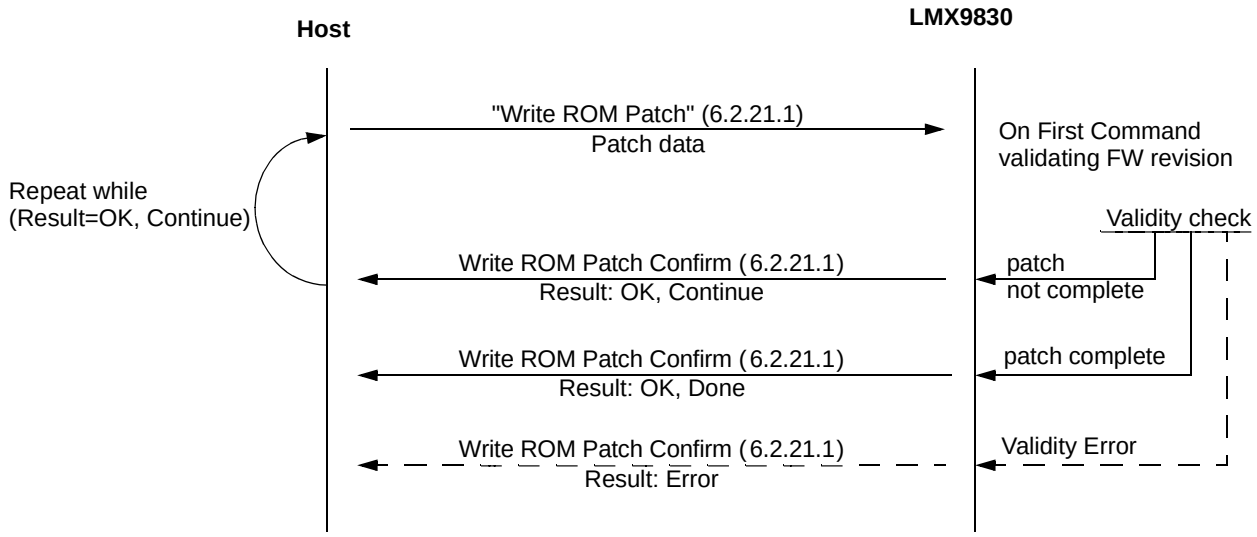


Figure 4-50. Patch Command flow

5.0 Profile Support

The LMX9830 is a full bluetooth node including the Generic Access profile (GAP), Service Discovery Profile (SDAP) and Serial Port Profile (SPP). In addition to those standard profiles, additional profiles can be implemented on host, using the LMX9830 as gateway to transport the higher layer protocols. A typical example for such a protocol is the AT-Commandset, which is used for Dial-up Networking (DUN) or also audio profiles like Headset (HSP) or Handsfree profiles (HFP).

The following information are based on [3] "Bluetooth SIG: Specification of the Bluetooth System 1.1, Volume 2 / Profiles, Version 1.1, February 22 2001".

5.1 MULTIPLE SERVICE ENTRIES

The LMX9830 includes a configurable database, which includes the services to be offered to remote devices. These services, which also include the dedicated RFCOMM port number, will be offered to a SDAP requests.

Section 3.2 "Setting up a link using the Command Interface" on page 32describes, how to establish a link using the LMX9830 command interface. This also includes the SDAP requests, which are necessary to get the available services and the dedicated RFComm port numbers.

5.1.1 The SDAP Service Database

By default the LMX9830 will offer one SPP profile to remote devices, which is dedicated to RFComm Port 1. The LMX9830 supports up to 10 database entries. See Table 5-58 for the default service database configuration.

Table 5-58. Default Service Database Configuration

Index	Enabled	Local RFComm Port	Type	Name	Authentication	Encryption	Profile Specifics
00	1	01	SPP	COM1	0x02	0x04	none
01							
09							

Once a remote device connects to the LMX9830, the firmware will first compare the RFCOMM port connected to the service database. In case no entry is found, the default settings as shown at index 0x00 in Table 5-58 will be used. In case an appropriate entry for the addressed RFCOMM port exists, the LMX9830 will check the authentication and encryption settings and initiate the necessary procedures.

Independent of the stored profile, the LMX9830 will not do any additional profile specific procedures beyond the SPP link establishment including authentication and encryption. In case other profiles like DUN or OPP have been stored, the host has to monitor the RFCOMM port number which has been connected ("Incoming Link Established Indicator" (6.2.3.7)) and has to react with the appropriate profile procedures.

5.1.2 Configuring the service database

The service database can be configured by one command. The command will add a specific entry to the database. Once it has been sent, the new entry will automatically be stored to the first unused index.

The index number will be confirmed within the appropriate command confirmation events.

- "Store Generic SDP Record" (6.2.13.1)

The index can be used to enable/disable the different entries during operation.

- “SDP Enable SDP Record” (page 169)

The entries can not be deleted separately. The database will only be deleted or reset to default by the following commands.

- “SDP Delete All SDP Records” (page 170)
- “Restore Factory Settings” (page 185)

5.1.3 Example

As mentioned before the LMX9830 will only use the RFCOMM port and the Authentication and Encryption parameters of the stored profiles. In case profiles other than SPP shall be implemented the LMX9830 will only handle the SPP link establishment, the remaining profile specific actions like AT commands for DUN need to be implemented on the host.

For this the host needs to know, which profile has been connected.

The following sections describe an example for the configuration of the LMX9830 service database to offer 3 services and how the host will be notified on the addressed profile.

In this example, the LMX9830 shall be used as bluetooth node within a mobile phone, offering the following services

- Serial Port Profile, SPP
- Dial Up Networking, DUN
- Headset Audio Gateway, HSAG

5.1.3.1 Configuring the service database

The first step in enabling the phone for these services is to add the service record to the service database. For this the "Store Generic SDP Record" (6.2.13.1) needs to be used.

For the setup the default SPP entry as set by "Restore Factory Settings" (6.2.18.5) is assumed and will not be modified. Therefore only the DUN and the HSP profile entries need to be made. Table 5-59 shows the two commands to be sent to the LMX9830.

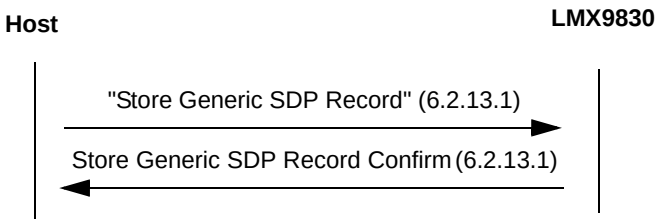


Figure 5-51. Configuring the service database

Table 5-59. Configuring the service database

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
	TX	Request	02,52,31,5E,00,E1,02,02,04,59,00,00,00,0A,00,00,00,00,01,00,35,03,19,03,11,04,00,35,0C,35,03,19,00,01,35,05,19,03,00,08,02,05,00,35,03,19,02,10,06,00,35,09,09,6E,65,09,6A,00,09,00,01,09,00,35,08,35,06,19,03,11,09,00,01,00,01,25,13,44,69,61,6C,2D,75,70,20,4E,65,74,77,6F,72,6B,69,6E,67,00,05,03,28,00,03	Tx: Cmd: Store SDP Record, Local Port: 02, Authentication: 02, Encryption: 04, SdpRecord: 00000A000000000100350319031104003350C3503190001350519030008020500350319021006003509096E65096A0009000109003508350619031109000100012513 4469616C2D7570204E6574776F726B696E670005032800
00 / 01	RX	Confirm	02,43,31,02,00,76,00,01,03	Rx: Event: Store SDP Record, Status: 00, Identifier: 01
	TX	Request	02,52,31,58,00,DB,03,22,44,53,00,00,00,0A,00,00,00,00,01,00,35,06,19,12,11,19,03,12,04,00,35,0C,35,03,19,00,01,35,05,19,03,00,08,03,05,00,35,03,19,02,10,06,00,35,09,09,6E,65,09,6A,00,09,00,01,09,00,35,08,35,06,19,08,11,09,00,01,00,01,25,0E,56,6F,69,63,65,20,47,61,74,65,77,61,79,00,03	Tx: Cmd: Store SDP Record, Local Port: 03, Authentication: 22, Encryption: 44, SdpRecord: 00000A000000000100350619121119031 20400350C3503190001350519030008030 5003503190210060035090 96E65096A0 0090001090035083506190811090001000 01250E566F6963652047617465776179 0
00 / 01	RX	Confirm	02,43,31,02,00,76,00,02,03	Rx: Event: Store SDP Record, Status: 00, Identifier: 02

After these commands the service database will be filled as described inTable 5-60.

NOTE: This example shows each profile connected to a different RFCOMM port. This makes it easy for the host to differentiate between the different profiles. As the amount of RFCOMM ports can have influence on the overall performance of the LMX9830, it is also possible to connect one RFCOMM port to multiple profiles. In this case the application on the host needs to be able to differentiate between the different profiles on application level.

Table 5-60. Example Service Database Configuration

Index	Enabled	Local RFComm Port	Type	Name	Authentication	Encryption	Profile Specifics
00	1	01	SPP	COM1	0x02	0x04	none
01	1	02	DUN	DUN	0x02	0x04	none
02	1	03	HSAG	Voice Gateway	0x22	0x44	none

5.1.3.2 Configuring RFCOMM ports to open

The LMX9830 can handle up to seven RFComm ports simultaneously. Each RFComm port opened creates a buffer instance within the RAM to handle the upcoming data traffic. Because of that by default only RFComm port 1 is activated.

The RFComm ports opened and initialized for operation can be configured by the following command:

- Section 6.2.12 "RFcomm Channels to open" on page 166

The ports in this command are expressed by a 32-bit mask indicating which RFCOMM ports the LMX9830 has to open. Bit 30 and 31 must be set to 0. Bit 0 is RFCOMM port 1 and bit 29 is port 30. The changes take effect as soon as the command has been confirmed.

Examples:

Open RFComm port 1: Set Ports to open to 0x00000001
Open RFComm port 1 and 3: Set Ports to open to 0x00000005
Open RFComm port 1, 2 and 3: Set Ports to open to 0x00000007

In this example, ports 1 to 3 would need to be opened, so the parameter has to be configured to 0x00000007.

NOTE: The LMX9830 is optimized for a maximum of seven open RFCOMM ports. Any additional port can reduce the ability to establish ACL links.

Table 5-61. Open the necessary RFCOMM Ports

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
	TX	Request	02,52,22,04,00,78,07,00,00,00,03	Tx: Cmd: Set Ports To Open, Ports: 07000000
00 / 01	RX	Confirm	02,43,22,01,00,66,00,03	Rx: Event: Set Ports To Open, Status: 00

5.1.3.3 Handling incoming links

As described in Section 5.1.3.1 on page 80, for incoming links the LMX9830 will check the service database for configurations on authentication and encryption and, if appropriate, serve profile specific attributes. Once the link is established the LMX9830 will indicate the incoming link and RFCOMM port number to the host with the standard command "Incoming Link Established Indicator" (6.2.3.7). Afterwards the host needs to decide according to the RFCOMM port number addressed, which service has been contacted and which profile specific procedure are necessary.

Figure 5-52 shows a typical flow, in which the remote device selects the DUN profile out of a service database.

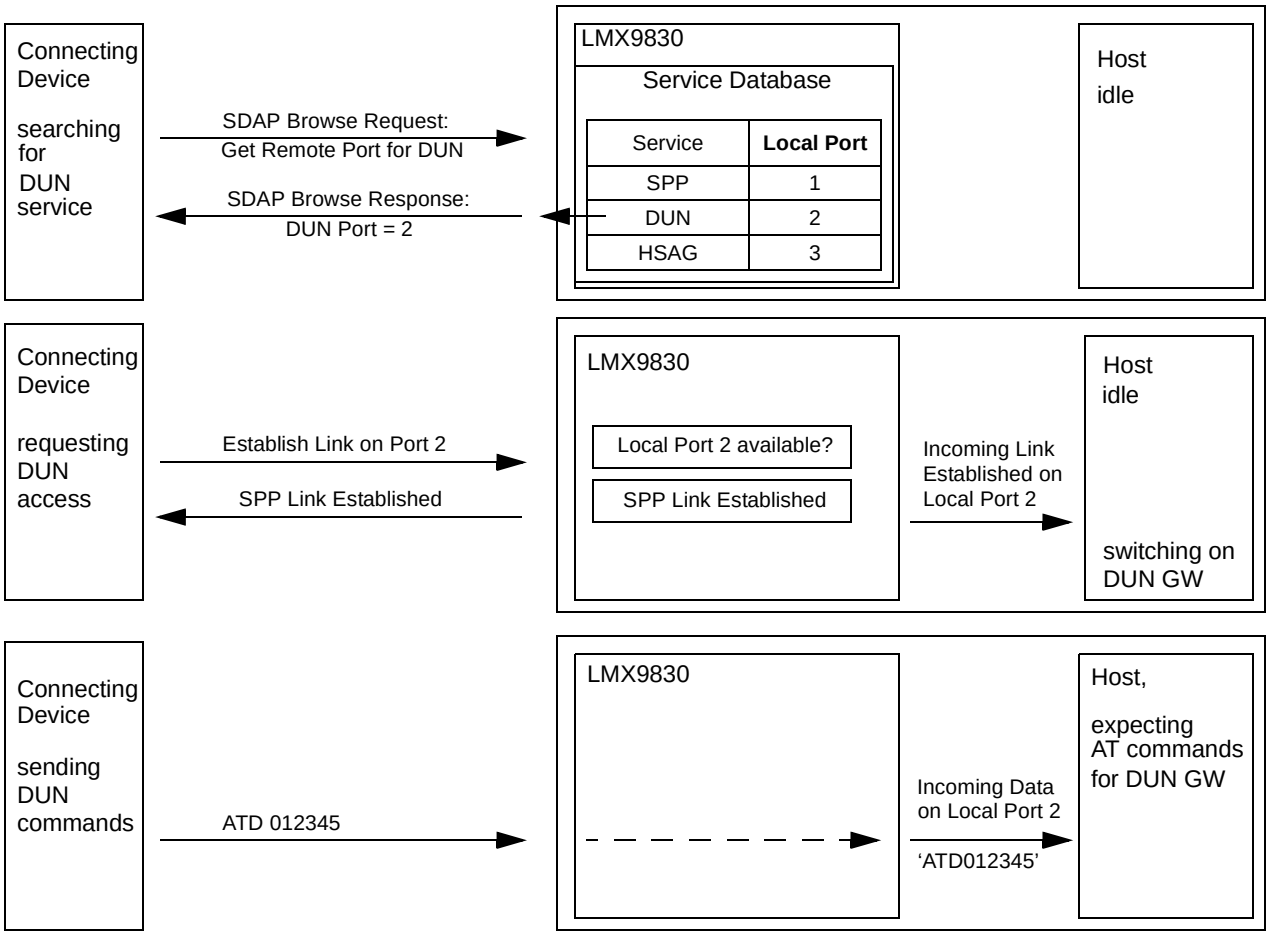


Figure 5-52. Example differentiate different profiles

The link establishment will be handled as standard SPP link. As first action, the connecting device needs to ask for the RFCOMM port it has to use for the required service. So in this case, the connecting device will first create a SDAP connecting and send a service browse request including the UUID for the desired service. The LMX9830 will answer with RFCOMM port number stored for DUN in NVS, in this example port 2.

Afterwards the connecting device requests a standard SPP link to the port number extracted from the SDAP browse. In case the RFCOMM port is available the LMX9830 accepts the incoming link, considering the settings within the service database.

Finally the LMX9830 indicates the successful link establishment by sending the "Incoming Link Established Indicator" (page 132), to the host. The event includes also the port number which has been connected.

According to the port number reported, the host needs to decide which profile procedures will have to be initiated.

Table 5-62. Host notification for incoming link on RFCOMM port 2

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
00 / 01	RX	Indicator	02,69,0C,07,00,7C,12,34,56,78,9A ,BC,**02**,03	Rx: Event: Incoming Link Established, BdAddr: 123456789ABC, Local Port: **02**

NOTE: As shown in Figure 5-52 the host will not be notified about the profile connected, it will only be notified about the RFCOMM port the remote port has established a link to. Because of this, the host needs to know the mapping between RFCOMM port and profile. This is necessary in case these profiles shall be available simultaneously.

Since more than one SPP based profile is based on AT commands, the differentiation between the profiles could also happen on application level, meaning all profiles use the same RFCOMM port. In this case, the host will not have to store the

mapping between profile and RFCOMM port but would have to have a generic AT Commandset application, supporting all different profiles. The drawback of this is that all remaining profiles will be blocked as soon as one profile is in use.

5.2 SERIAL PORT PROFILE (SPP)

The SPP profile the basic profile to be used for standard cable replacement. The SPP profile is fully included within the LMX9830. The SPP profile is basically covered by the SPP Link establishment command. Please see the examples in Section 3.4 "Examples (Summary)" on page 44.

5.3 DIAL-UP NETWORKING (DUN)

The Dial-up Networking Profile defines the protocols and procedures that shall be used by devices implementing the usage model called ‘Internet Bridge’. The most common examples of such devices are modems and cellular phones.

The following roles are defined for this profile:

Gateway (GW) – This is the device that provides access to the public network. Typical devices acting as gateways are cellular phones and modems.

Data Terminal (DT) – This is the device that uses the dial-up services of the gateway. Typical devices acting as data terminals are laptops and desktop PCs.

In the rest of this document, these terms are only used to designate these roles.

The scenarios covered by this profile are the following:

- Usage of a GW by a DT as a wireless modem for connecting to a dial-up internet access server or using other dial-up services
- Usage of a GW by a DT to receive data calls

The LMX9830 is capable of acting as both gateway or data terminal. In case of acting as DT, no further configurations are necessary. Acting as GW the LMX9830 needs to be configured to offer the DUN networking service to the DT, meaning an additional Service Record has to be stored within the device.

Both configuration scenarios are explained in more detail within the following sections.

5.3.1 AT Commandset

Dial up networking is based on a serial connection between two devices. The profile itself is handled by AT commands. So in order to create a DUN profile a standard SPP link has to be established, which then is used to transport the AT commands. For example using the LMX9830 to dial out with a mobile phone it first establishes a standard SPP link as done with SPP profile and then sending “ATDT0123456” as data to dial the specific number.

The AT commandset together with the SPP Profile form the Dial up networking profile.

Please refer to [3] to get the full of AT Commands to be supported by GW or DT devices. The commands are also listed in Section 6.4 “AT Commands” on page 202.

5.3.2 Acting as DUN data terminal

Using the LMX9830 as data terminal (DT) means, it uses the DUN service of another device, e.g. a mobile phone, to dial out. The host needs to send the standard AT commands to establish the SPP link as described in Section 3.2 “Setting up a link using the Command Interface” on page 32. There are no additional configurations necessary on the LMX9830.

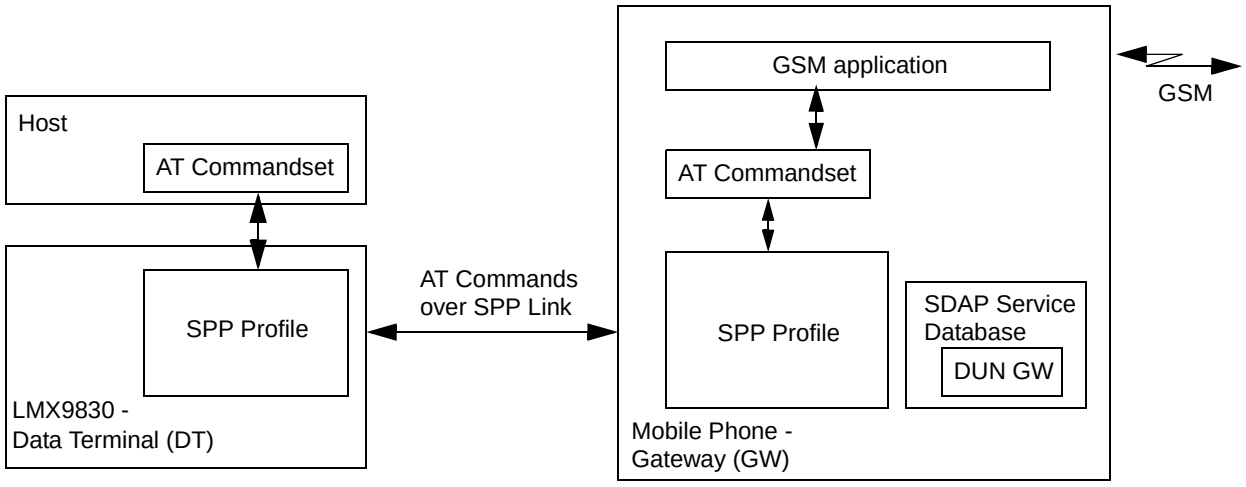


Figure 5-53. Software Layer implementations for DUN

The only difference for the link setup is that the DT has to connect to the DUN profile of the remote device instead of connecting to the SPP profile. To reach that, the correct remote RFComm port has to be used within the "Establish Link" (6.2.3.1) command. Once the correct RFComm port has been used, the remote device knows automatically that DUN has been addressed.

The following chapters give a step by step description of connecting to the DUN service of a mobile phone.

5.3.2.1 Searching for the device

The Inquiry is a standard procedure which is not related to any profile. With this the BD_Addr of the remote device will be known. The remote device needs to have inquiry scan switched on.

5.3.2.2 Getting the RFCOMM port for DUN on the remote device

As described in Section 3.2.3 “SDAP Service Browse for SPP” on page 36, the remote RFComm port number for the service can be found out by using the Service Browse command. The command includes the parameter UUID, which defines the service to be looked for. The UUID for Dial Up Networking profile is 0x1103. See Figure 5-54 for the command flow, Table 5-63 shows the detailed parameters to be used for the commands.

Table 5-63. Get Remote RFComm Port for DUN profile of the remote device

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
	TX	Request	02,52,32,06,00,8A,12,34,56,78,9A, BC,03	Tx: Cmd: SDAP Connect, BdAddr: 123456789ABC
00	RX	Indicator	02,69,50,07,00,C0,12,34,56,78,9A, BC,00,03	Rx: Event: ACL Established, BdAddr: 123456789ABC, Status: 00
00 / 01	RX	Confirm	02,43,32,01,00,76,00,03	Rx: Event: SDAP Connect, Status: 00
00 / 01	RX	Request	02,52,35,02,00,89,**03,11**,03	Rx: Event: Service Browse, Browse Group ID: **0311**
00 / 01	RX	Confirm	02,43,35,1B,00,93,00,01,02,10,03, 11,**03**,13,44,69,61,6C,2D,75,70,20, 4E,65,74,77,6F,72,6B,69,6E,67,00, 03	Rx: Event: Service Browse, Status: 00, Browse Group ID: 0210, Service ID: 0311, PortNo: **03**, Service Name: Dial-up Networking.
	TX	Request	02,52,33,00,00,85,03	Tx: Cmd: SDAP Disconnect
00 / 01	RX	Confirm	02,43,33,01,00,77,00,03	Rx: Event: SDAP Disconnect, Status: 00
00	RX	Indicator	02,69,51,07,00,C1,12,34,56,78,9A, BC,16,03	Rx: Event: ACL Terminated, BdAddr: 123456789ABC, Reason: 16

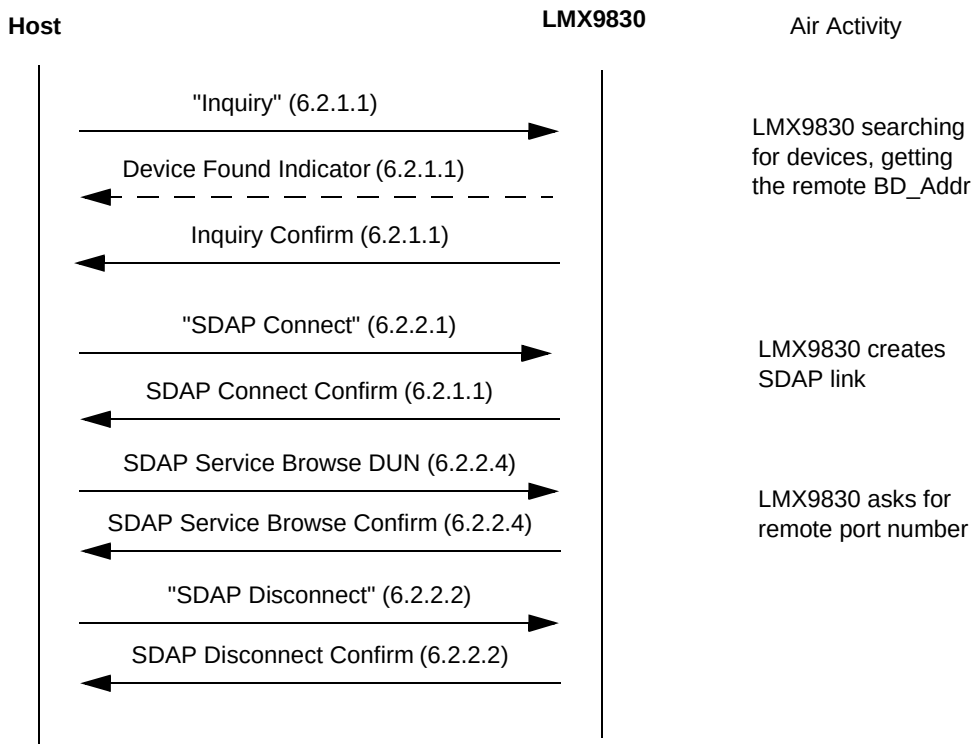


Figure 5-54. Command Flow: Requesting the RFComm Port Number from the remote device

5.3.2.3 Establishing a Link to a Dial Up Networking service

The link establishment to a DUN profile is the same as to create a link to a standard SPP port. The remote device will recognize the profile to be used by the RFComm port, to which the SPP link has been created to. For this example, out of the procedure explained in Section 5.3.2.2, the RFComm port to be addressed for DUN is 0x03.

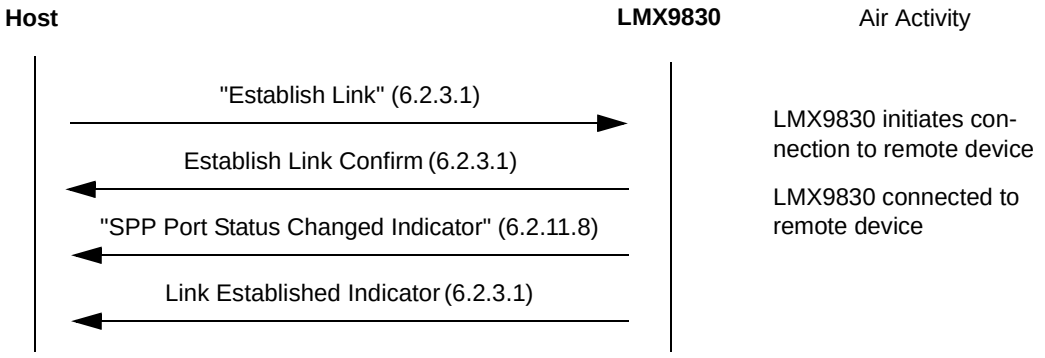


Figure 5-55. Command Flow: Establish a link to a DUN service of the remote device

Table 5-64. Establish Link to DUN service on remote device

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
	TX	Request	02,52,0A,08,00,64,01,12,34,56,78, 9A,BC,03,03	Tx: Cmd: Establish Link, Local Port: **01**, BdAddr: **123456789ABC**, Remote Port Number: **03**

Table 5-64. Establish Link to DUN service on remote device

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
00 / 01	RX	Confirm	02,43,0A,02,00,4F,00**,01**,03	Rx: Event: Establish Link, Status: 00, Local Port: **01**
00	RX	Indicator	02,69,50,07,00,C0,12,34,56,78,9A, BC,**00**,03	Rx: Event: ACL Established, BdAddr: 123456789ABC, Status: **00**
00 / 01	RX	Indicator	02,69,3E,04,00,AB,01,0C,00,00,03	Rx: Event: Port Status Changed, Local Port: 01, PortStatus: 0C, Break Length: 0000
00 / 01	RX	Indicator	02,69,0B,09,00,7D,00,12,34,56,78, 9A,BC,01,**03**,03	Rx: Event: Link Established, Status: 00, BdAddr: 123456789ABC, Local Port: 01, Remote Port Number: **03**

5.3.2.4 Sending AT Commands

Once the SPP link is established the GW will activate it's AT commandset, expecting and accepting a specific AT commands as described in [3]. A full list of commands to be supported by the GW can also be found in Section 6.4.1 on page 202.

AT commands are basically ASCII characters to be sent to the remote device. The commands additionally make use of Carriage Return (0x0D) and Line Feed (0x0A).

So the commands from DT to GW will be thus:

AT<cmd>=<value><cr>

If the command is processed successfully, the resulting response from the GW to the DT is:

<cr><lf>OK<cr><lf>

If the command is not processed successfully, the resulting response from the GW to the DT is:

<cr><lf>ERROR<cr><lf>

The AT commands can be sent by either using the"SPP Send Data" (6.2.3.3) command or by first switching to transparent mode and then send the commands directly.

5.3.2.4.1 Using the command interface to send and receive AT commands

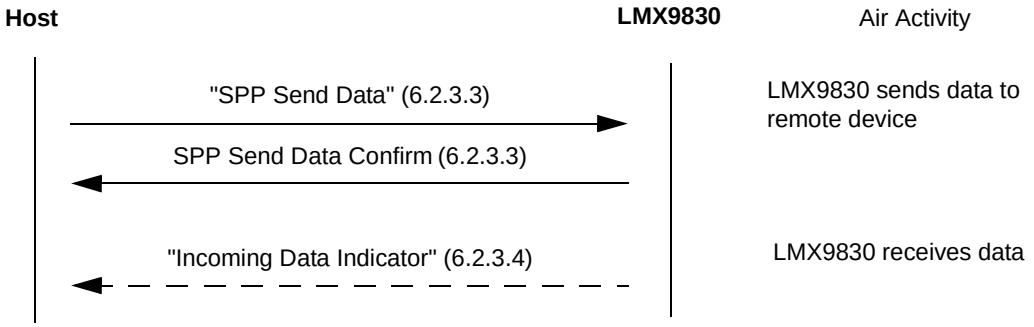


Figure 5-56. Flow Command: Sending AT Commands

Table 5-65. Send init command "AT" and dialing command "ATDT01234567890"

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
	TX	Request	02,52,0F,07,00,68,01,04,00,41,54, 0D,0A,03	Tx: Cmd: Send Data, Local Port: 01, Payload Data: 41540D0A (AT)

Table 5-65. Send init command “AT” and dialing command “ATDT01234567890”

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
00 / 01	RX	Confirm	02,43,0F,02,00,54,00,01,03	Rx: Event: Send Data, Status: 00, Local Port: 01
00 / 01	RX	Indicator	02,69,10,09,00,82,01,06,00,0D,0A, 4F,4B,0D,0A,03	Rx: Event: Incoming Data, Local Port: 01, Received Data: 0D0A4F4B0D0A *(OK)*
	TX	Request	02,52,0F,13,00,74,01,10,00,41,54, 44,54,30,31,32,33,34,35,36,37,38, 39,30,0D,03	Tx: Cmd: Send Data, Local Port: 01, Payload Data: 4154445430313233343536373839300D *(ATDT01234567890)*
00 / 01	RX	Confirm	02,43,0F,02,00,54,00,01,03	Rx: Event: Send Data, Status: 00, Local Port: 01

5.3.2.4.2 Using transparent mode to send and receive AT commands

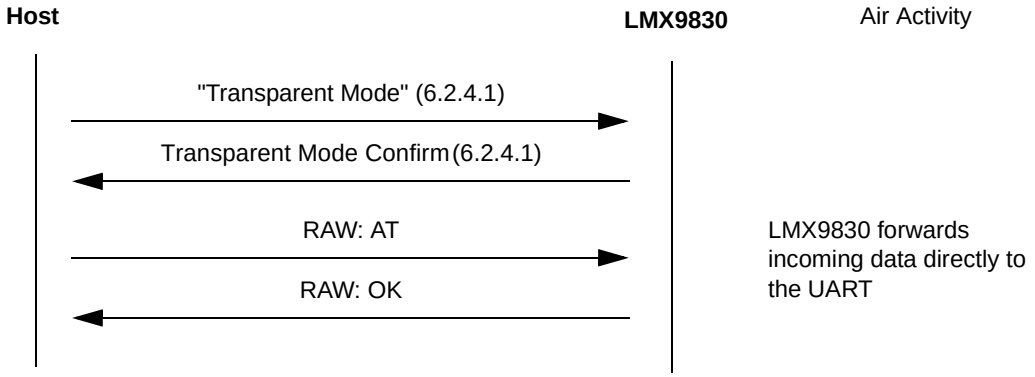


Figure 5-57. Sending AT Commands

Table 5-66. Send init command “AT” and dialing command “ATDT01234567890”

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
	TX	Request	02,52,11,01,00,64,01,03	Tx: Cmd: Transparent Mode, Local Port: 01
00 / 01	RX	Confirm	02,43,11,02,00,56,00,01,03	Rx: Event: Transparent Mode, Status: 00, Local Port: 01
	TX	RAW Data	41,54,0D,0A	Tx(RAW): 41,54,0D *(AT)*
00 / 01 / 02 / 03	RX	RAW Data	0D,0A,4F,4B,0D,0A	Rx(RAW): 0D,0A,4F,4B,0D,0A *(OK)*
	TX	RAW Data	41,54,44,54,30,31,32,33,34,35,36, 37,38,39,30,0D	Tx(RAW): 02,52,0F,13,00,74,01,10,00,41,54,44,54,30,31,32,33,34,35,36,37,38,39,30,0D *(ATDT01234567890)*

5.3.3 Acting as DUN gateway

A DUN gateway offers the ability to access another network, e.g. GSM, routing the data between the data from the bluetooth link to the other network. The LMX9830 can be configured to offer a DUN networking service to other devices, e.g. as in a mobile phone or LAN access point. The host is able to differentiate the requested DUN service by checking the local port, to which has been connected to. Please see Section 5.1.3.3 on page 81 for details how to differentiate between different profiles.

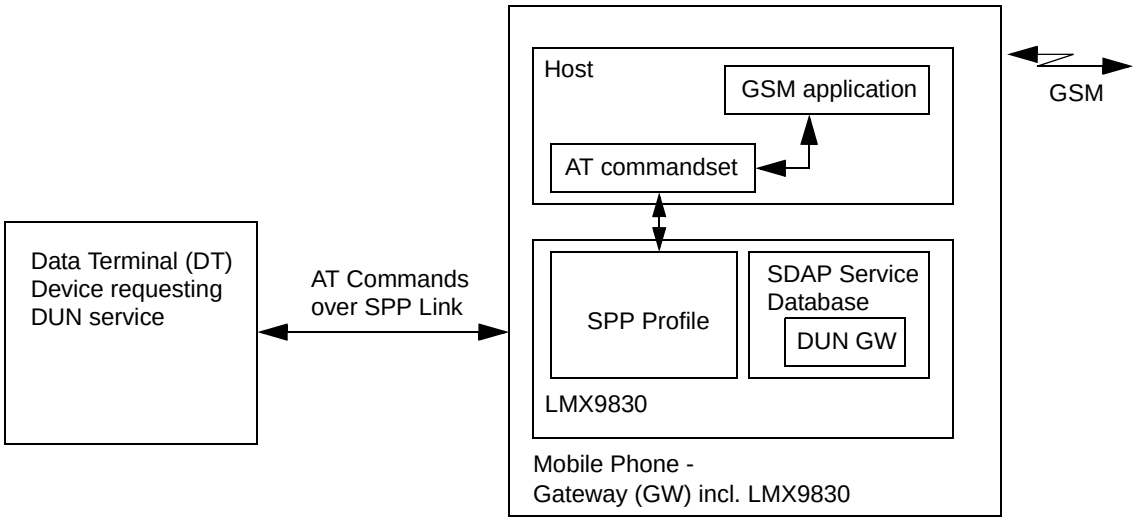


Figure 5-58. Using LMX9830 in a DUN GW application

5.3.3.1 Configuring the Service Database

In order to offer an additional DUN service to other devices a new entry has been made into the service database. For this the "Store Generic SDP Record" (6.2.13.1) can be used. The following example shows how to add the DUN entry to the default SDB database including only one SPP entry.

The generation of profile entries is supported in Simply Blue Commander versions later than 1.3.0.3. Please follow the following figures to create a DUN service database entry:

1) Select “Definitions/Create Service Record” within the Simply Blue Commander menu.

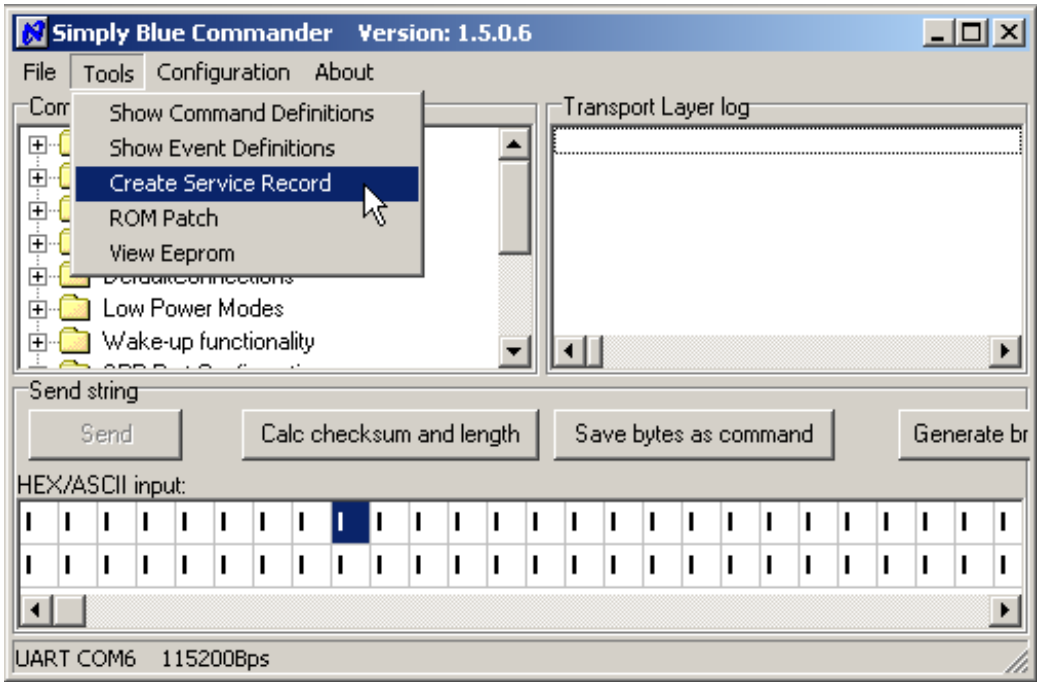


Figure 5-59. Opening “Create Service Record” Dialog-box

2) Select the service record required, in this case “Dial-up Networking (DUN)”

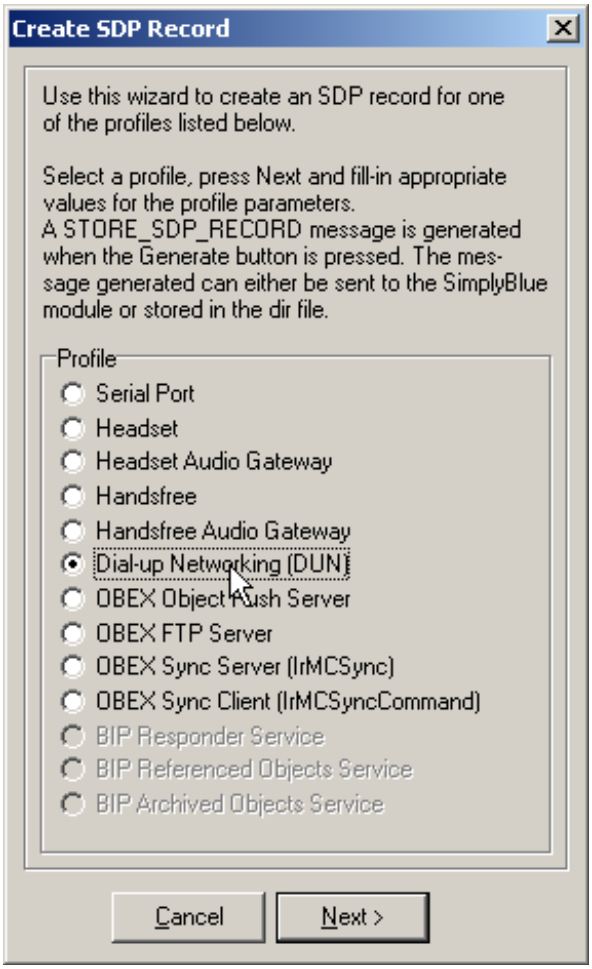


Figure 5-60. Selecting the profile

3) Select the settings desired for the DUN profile and choose the correct RFCOMM port. As this example device shall still be able to accept a second standard data links as well, RFCOMM port 2 should be used. Finally confirm the dialog with pressing “Create”.

Figure 5-61. Configuring the profile settings for DUN

4) The “Create” will fill the Hex/ASCII line of the Simply Blue Commander with the string necessary to send to the LMX9830. Afterwards just close the dialog with the “Close” Button.

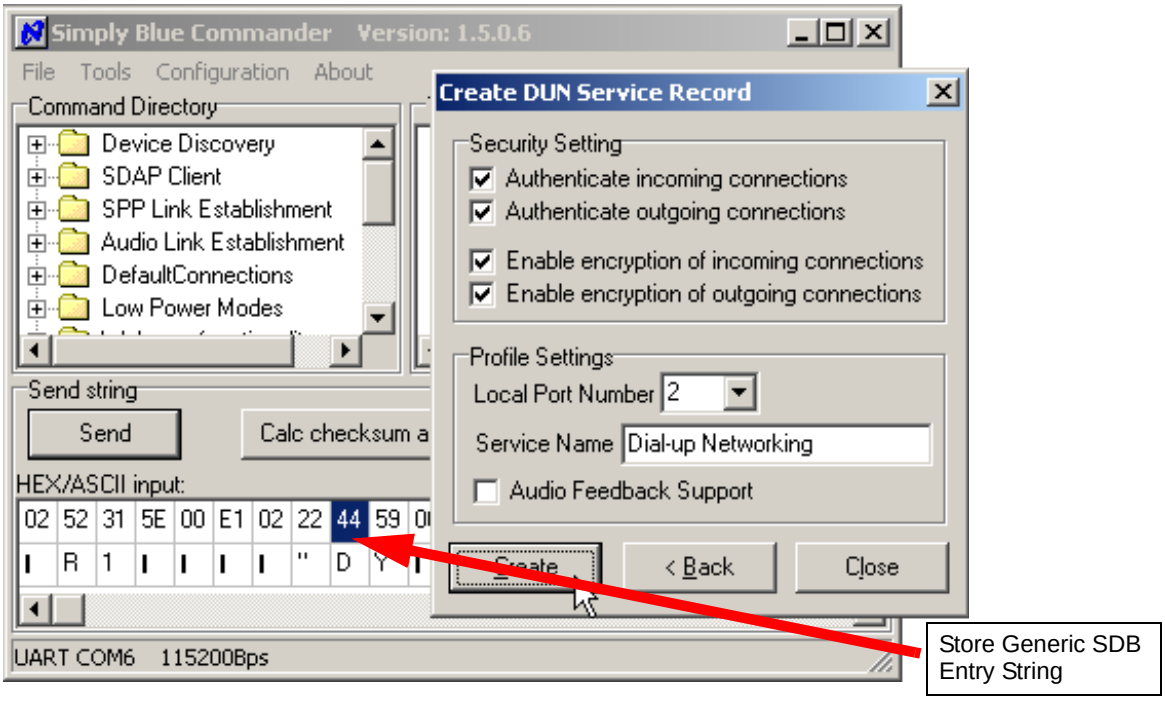


Figure 5-62. Creating the command string for the audio gateway entry

5) The string can be sent directly to the board or stored as command within the directory. Once sent it can also be copied out of the log entry and copied into the development code when switching of the interpretation option.

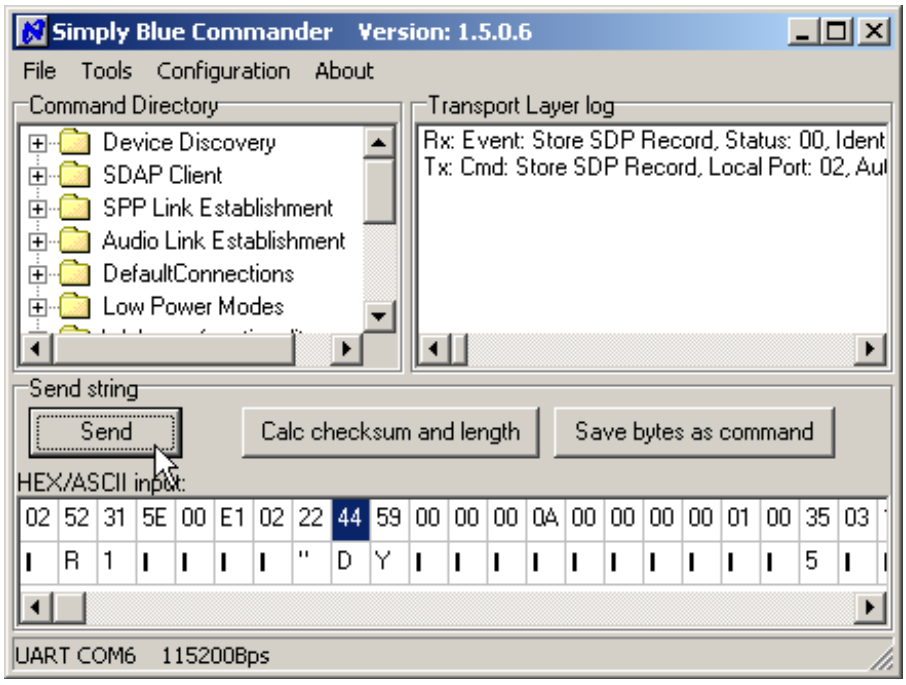


Figure 5-63. Send the command to the LMX9830

.

Table 5-67. Adding the DUN service to the Service Database

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
	TX	Request	02,52,31,5E,00,E1,02,02,04,59,00,00,00,0A,00,00,00,00,01,00,35,03,19,03,11,04,00,35,0C,35,03,19,00,01,35,05,19,03,00,08,02,05,00,35,03,19,02,10,06,00,35,09,09,6E,65,09,6A,00,09,00,01,09,00,35,08,35,06,19,03,11,09,00,01,00,01,25,13,44,69,61,6C,2D,75,70,20,4E,65,74,77,6F,72,6B,69,6E,67,00,05,03,28,00,03	Tx: Cmd: Store SDP Record, Local Port: 02, Authentication: 02, Encryption: 04, SdpRecord: 00000A000000001003503190311040035 0C3503190001350519030008020500350319021006003509096E65096A0009000109003508350619031109000010001251344469616C2D7570204E6574776F726B696E670005032800
00 / 01	RX	Confirm	02,43,31,02,00,76,00,01,03	Rx: Event: Store SDP Record, Status: 00, Identifier: 01

After this command the Service database has the following content:

Table 5-68. Service Database for COM1 and DUN

Index	Enabled	Local RFComm Port	Type	Name	Authentication	Encryption	Profile Specifics
00	1	01	SPP	COM1	0x02	0x04	none
01	1	02	DUN	DUN	0x02	0x04	none

5.3.3.2 Adapt RFCOMM ports to open

As both services shall be connectable independently, two separate RFCOMM ports need to be used. For this the "Set Ports To Open" (6.2.12.1) command should be used.

Table 5-69. Set RFCOMM ports to open

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
	TX	Request	02,52,22,04,00,78,03,00,00,00,03	Tx: Cmd: Set Ports To Open, Ports: **03000000**
00 / 01	RX	Confirm	02,43,22,01,00,66,00,03	Rx: Event: Set Ports To Open, Status: 00

5.3.3.3 Routing AT Commands

The DUN GW profile is based on the standard AT commandset as listed in Section 6.4.1 “DUN GW” on page 202.

Once an incoming link has been established to the appropriate RFCOMM port, the host needs to be able react to these AT commands. The LMX9830 will only route the AT commands from the host to the remote device.

Please refer to Section 5.1.3.3 “Handling incoming links” on page 81 for how to manage incoming connections. The AT commands need to be sent and recognized as described in Section 5.3.2.4 “Sending AT Commands” on page 86

5.4 HEADSET PROFILE

The following description is based on [3]. Please refer to that document for complete description of the profile.

This Headset profile defines the protocols and procedures that shall be used by devices implementing the usage model called ‘Ultimate Headset’. The most common examples of such devices are headsets, personal computers, and cellular phones.

The headset can be wirelessly connected for the purposes of acting as the device’s audio input and output mechanism, providing full duplex audio.

The following roles are defined for this profile:

Audio Gateway (HSAG) – This is the device that is the gateway of the audio, both for input and output. Typical devices acting as Audio Gateways are cellular phones and personal computer.

Headset (HS) – This is the device acting as the Audio Gateway's remote audio input and output mechanism.

The profile specifies guidelines for incoming and outgoing connections for both profile roles, covering the following scenarios:

- AG initiated connection establishment
- HS initiated connection establishment
- Audio connection release
  — HS initiated
  — HSAG initiated
- Audio connection transfer
  — Transfer from HSAG to HS
  — Transfer from HS to HSAG
- Remote Volume Control

The following sections describe, how to use the LMX9830 as HSAG or HS.

5.4.1 AT Commandset and Results

The command line termination character shall be carriage return (0x0D). The response formatting character shall be line feed (0x0A). The HSAG shall not echo command characters (Opposite to default recommendation by ITU V.250). The HSAG shall transmit result codes, using the verbose (rather than numeric) format.

The format for a command from the HS to the HSAG is thus:

AT<cmd>=<value><cr>

If the command is processed successfully, the resulting response from the HSAG to the HS is:

<cr><lf>OK<cr><lf>

If the command is not processed successfully, the resulting response from the HSAG to the HS is:

<cr><lf>ERROR<cr><lf>

The format for an unsolicited result code (such as RING) from the HSAG to the HS is:

<cr><lf><result code><cr><lf>

The headset profile uses a subset of AT commands and result codes from existing standards.

Please see Section 6.4.2.1 on page 203 for the complete list of commands necessary for the headset profile.

5.4.2 Acting as Headset

The headset is defined as the device, providing the input and output for audio data, e.g. microphone and speaker. The LMX9830 is able to connect an external codec via the on-chip advanced audio interface (AAI). Please see Figure 1-6 "PCM Codec connection block diagram" on page 17 how to connect the codec to the AAI.

The HS side of the headset profile is mostly controlled by Audio gateway. While the SPP link establishment can be initiated from both sides, the SCO link establishment will always be initiated by the HSAG. The LMX9830 will automatically accept incoming SCO links and report the successful establishment to the host.

The only configuration necessary is to create the service database record and to configure the default audio settings for the PCM codec driver.

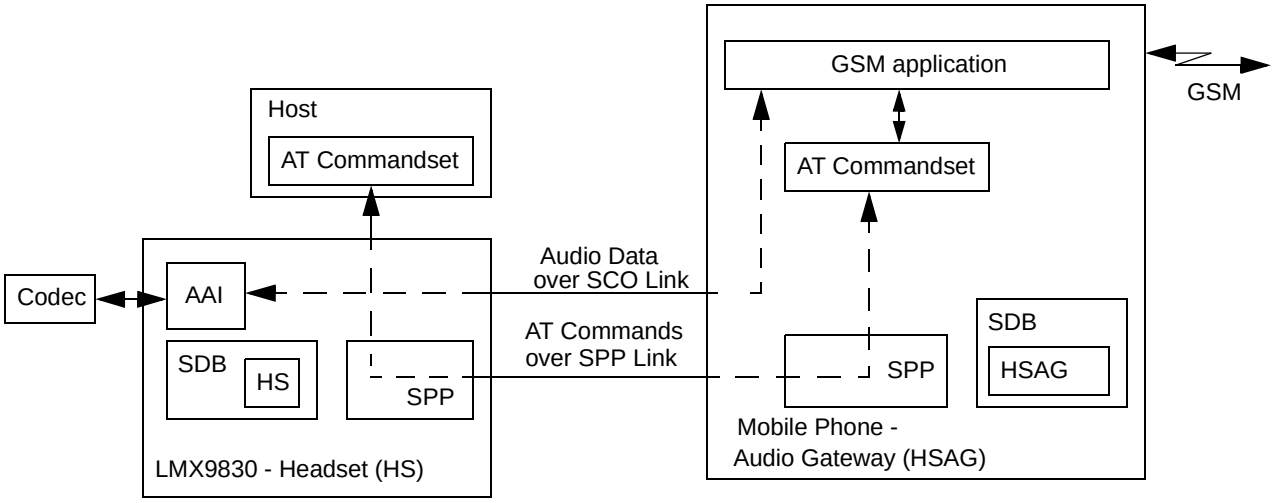


Figure 5-64. Headset Data and Audio stream

5.4.2.1 Configuring the audio path

In general the LMX9830 can be seen as gateway between the UART interface and the bluetooth link. The audio link is one additional interface for the LMX9830 to be routed. Please see also Figure 4-41 on page 70 for the two routing options.

The LMX9830 supports different configurations for the PCM codec interface and the air interface. The bluetooth specification defines the following formats to be used over the bluetooth link:

- CVSD (default)
- µ-Law
- A-Law

As indicated, the default and mostly used coding format used over a bluetooth link is CVSD. Since the headset most likely will always use the same settings, it is recommended to set the parameters to a default in the NVS, using the "Set audio settings" (6.2.6.2.5) command.

Table 5-70. Set Default Audio Settings to OKI codec and CVSD

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
	TX	Request	02,52,5B,02,00,AF,**02**,**00**,03	Tx: Cmd: Set Default Audio Settings, Codec Type: **02**, Air Format: **00**
00 / 01	RX	Confirm	02,43,5B,01,00,9F,00,03	Rx: Event: Set Default Audio Settings, Status: 00

5.4.2.2 Configuring the Service Database for Headset

Since a headset usually is only used for exactly that purpose, the "SDP Delete All SDP Records" (6.2.13.3) command could be used first, to delete the default SPP service database entry “COM1”. Table ## shows the command to be used to delete the complete database.

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
	TX	Request	02,52,2A,00,00,7C,03	Tx: Cmd: Delete All SDP Records
00 / 01	RX	Confirm	02,43,2A,01,00,6E,00,03	Rx: Event: Delete All SDP Records, Status: 00

In order to offer an Headset service to other devices a new entry has been made into the service database. For this the "Store Generic SDP Record" (6.2.13.1) needs to be used.

This command is a generic command to generate any possible profile within the device. To be able to do so, significant knowledge of the internal SDB structure would be required. In order to use this command, the “Simply Blue Commander” software needs to be used, to create this command.

The generation of profile entries is supported in Simply Blue Commander versions later than 1.3.0.3. Please follow the following figures to create a headset service database entry:

1) Select “Definitions/Create Service Record” within the Simply Blue Commander menu.

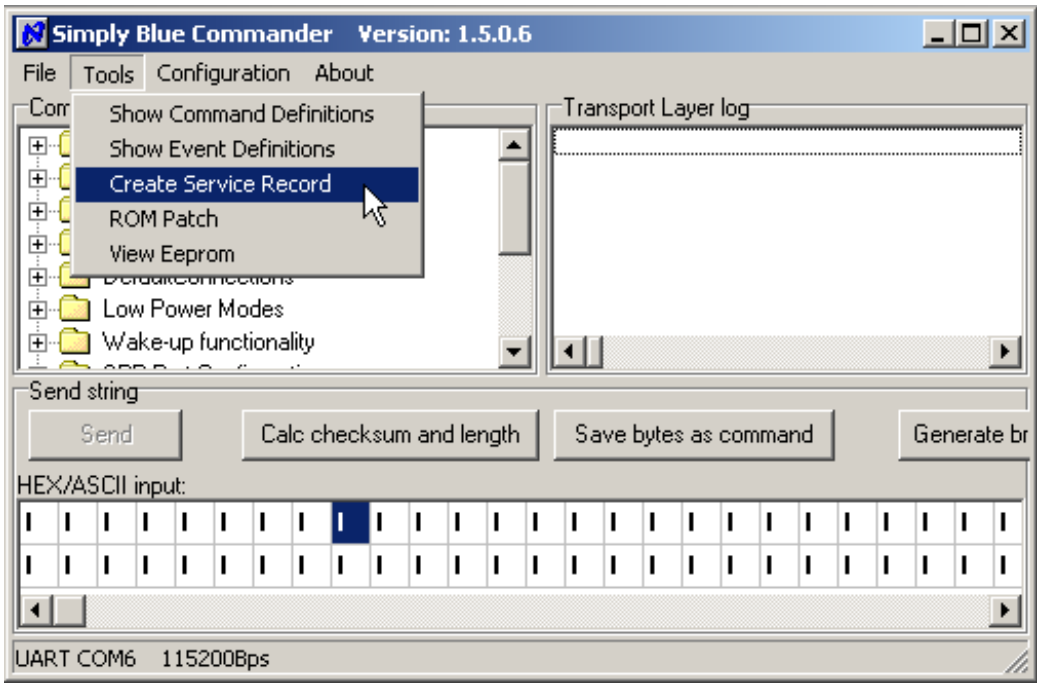


Figure 5-65. Opening “Create Service Record” Dialog-box

2) Select the service record required, in this case “Headset”

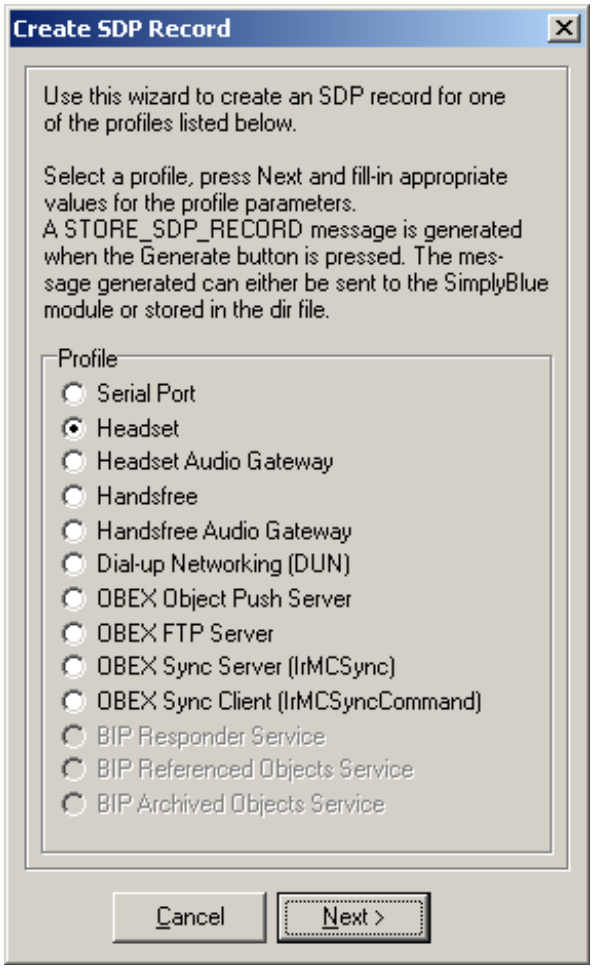


Figure 5-66. Selecting the profile

3) Select the settings desired for the Headset profile and choose the correct RFCOMM port. Since the database has been deleted before, RFCOMM port 1 can be used. Finally confirm the dialog with pressing “Create”.

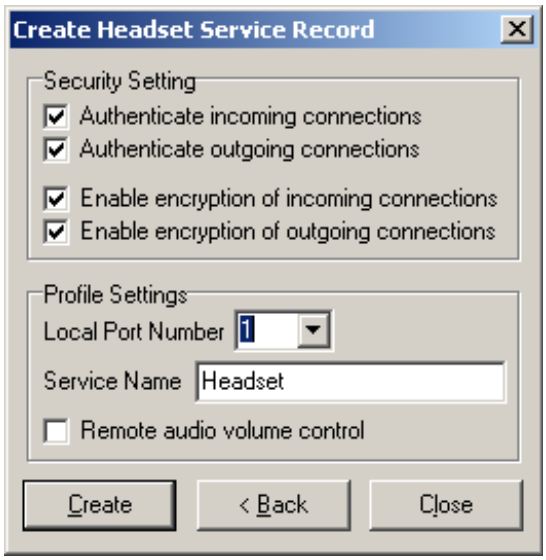


Figure 5-67. Configuring the profile settings for headset

4) The “Create” will fill the Hex/ASCII line of the Simply Blue Commander with the string necessary to send to the LMX9830. Afterwards just close the dialog with the “Close” Button.

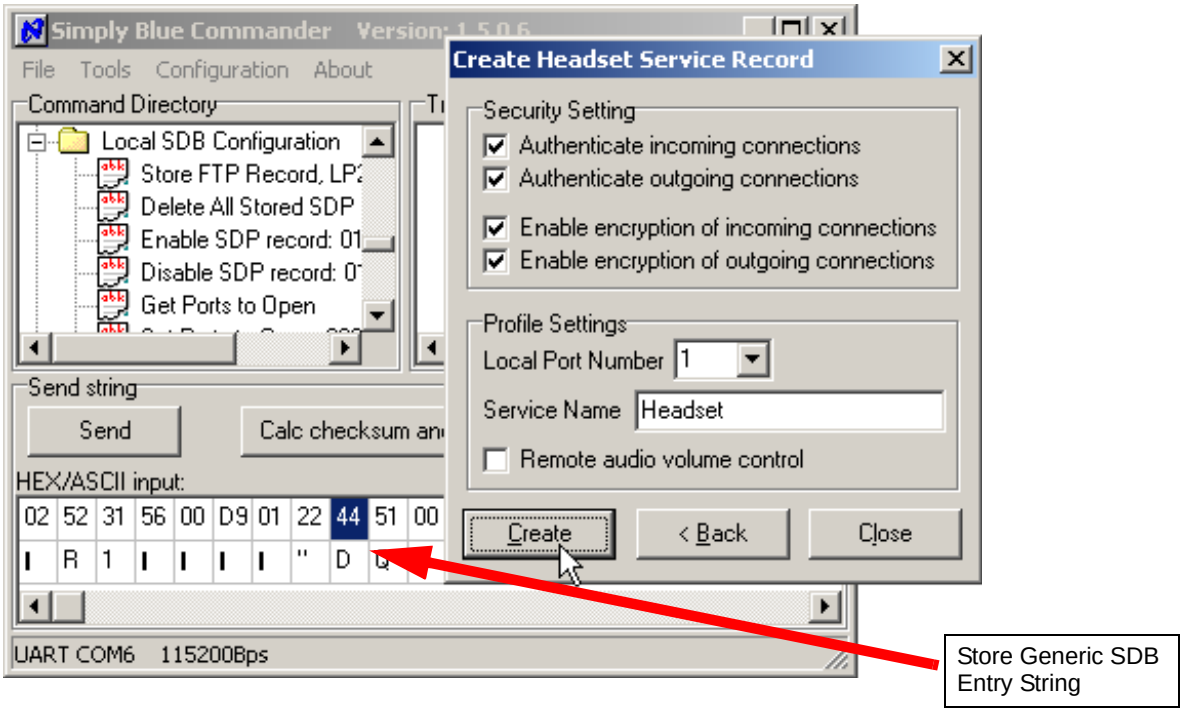


Figure 5-68. Creating the command string for the headset entry

5) The string can be sent directly to the board or stored as command within the directory. Once sent it can also be copied out of the log entry and copied into the development code when switching of the interpretation option.

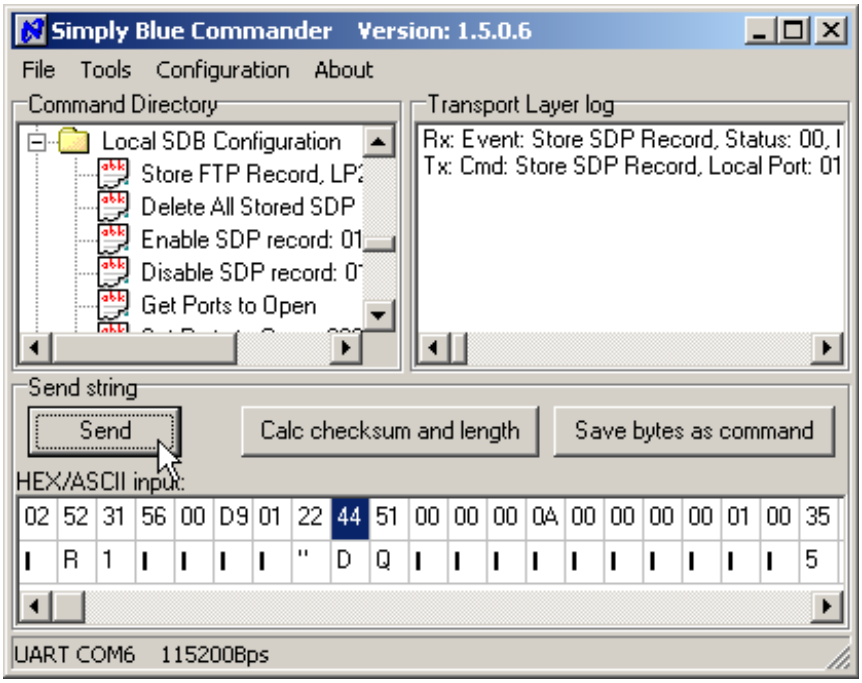


Figure 5-69. Send the command to the LMX9830

Table 5-71 shows the command string being the result out of the steps just described.

Table 5-71. Adding the headset service to the Service Database

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
	TX	Request	02,52,31,56,00,D9,01,22,44,51,00, 00,00,0A,00,00,00,00,01,00,35,06, 19,08,11,19,03,12,04,00,35,0C,35, 03,19,00,01,35,05,19,03,00,08,01, 05,00,35,03,19,02,10,06,00,35,09, 09,6E,65,09,6A,00,09,00,01,09,00, 35,08,35,06,19,08,11,09,00,01,00, 01,25,08,48,65,61,64,73,65,74,00, 02,03,28,00,03	Tx: Cmd: Store SDP Record, Local Port: **01**, Authentication: **22**, Encryption: **44**, SdpRecord: 00000A000000001003506190811190312040035 0C350319000135051903000801050035031902100600350909 6E65096A0009000109003508350619081109000100125084865616473657400020328 00
00 / 01	RX	Confirm	02,43,31,02,00,76,00,00,03	Rx: Event: Store SDP Record, Status: 00, Identifier: **00**

After this command the Service database has the following content:

Table 5-72. Service Database for COM1 and Headset

Index	Enabled	Local RFComm Port	Type	Name	Authentication	Encryption	Profile Specifics
00	1	01	HS	Headset	0x22	0x44	none

5.4.2.3 Adapt RFCOMM ports to open

To only serve this one service only RFComm port one needs to be opened. In order to make only RFComm port 1 available for connection, the "Set Ports To Open" (6.2.12.1) command should be used to make the appropriate configuration.

Table 5-73. Set RFCOMM ports to open

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
	TX	Request	02,52,22,04,00,78,01,00,00,00,03	Tx: Cmd: Set Ports To Open, Ports: **01000000**
00 / 01	RX	Confirm	02,43,22,01,00,66,00,03	Rx: Event: Set Ports To Open, Status: 00

5.4.2.4 Store Class of Device

During the Inquiry procedure the searching device will already receive the class of device. Please see also Section 4.1.2.3 "Class of Device" on page 49 for more information on that parameter. In order to be recognized it is beneficial to already indicate the audio capability within the class of device. The following table shows the command how to set the “Class of Device” using the "Store Class of Device" (6.2.14.5) command. The class is set to 0x220404, which reflects the class for a headset.

Table 5-74. Store Class of Device

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
	TX	Request	02,52,28,03,00,7D,**04**,**04**,**22**,03	Tx: Cmd: Store Class of Device, Class of Device: **040422**
00 / 01	RX	Confirm	02,43,22,01,00,66,00,03	Rx: Event: Store Class of Device, Status: 00

5.4.2.5 Connection handling as headset

The Headset profile is based on some specific AT commands also listed in Section 6.4.2 on page 203.

In a headset application the link will either be initiated from the HSAG or the HS itself. In both cases the HSAG will control the SCO link establishment. The HS will only react on incoming AT commands and maybe send the request for establishment or release.

The LMX9830 accepts incoming SCO links or SPP links automatically. It just indicates the successful establishment to the host. Therefore, for the HS implementation no further LMX9830 specific commands besides the already described configuration are required. The headset profile itself is controlled by a few AT commands.

The following chapters give the generic guideline of how to act as headset within the different scenarios.

5.4.2.5.1 Incoming audio connection

Figure 5-70 on page 98 shows the flow diagram of a headset link initiated by the HSAG as defined in [3].

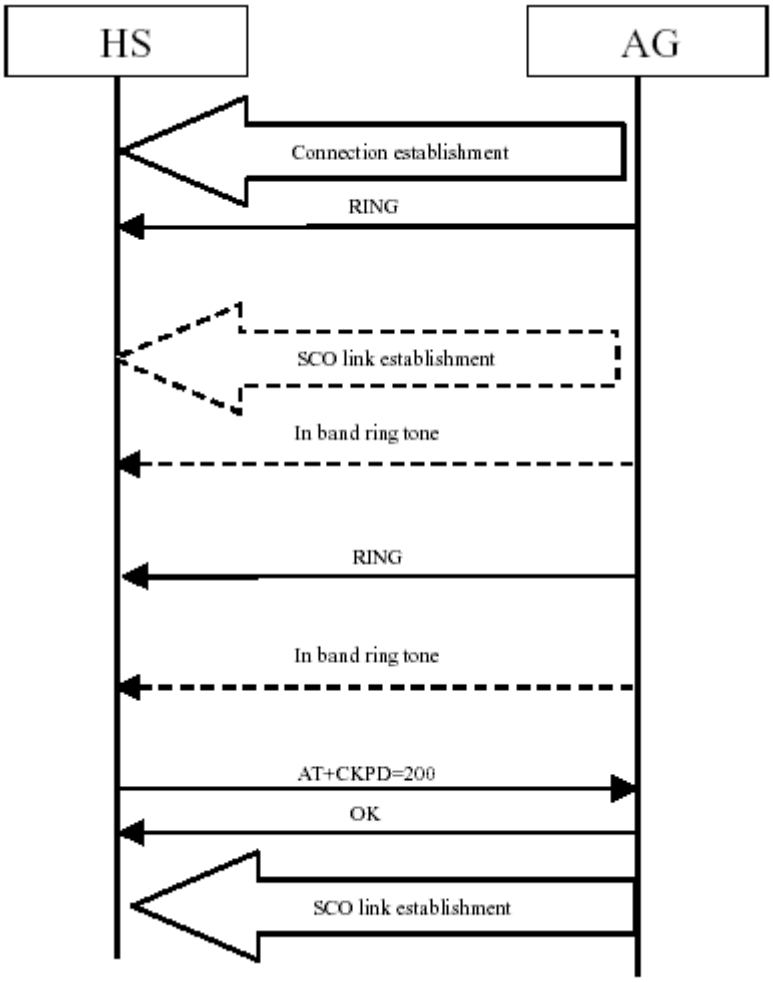


Figure 5-70. “Incoming audio connection establishment” as defined in bluetooth specification

As the specification figure already shows, the HS only needs to accept the incoming SPP and audio link and to indicate a button press by sending the “AT+CKPD=200” command. In addition the HS application needs to be able to recognize the RING command and the OK result code.

Please see Figure 5-71 which shows the headset part of the profile for the incoming audio connection ‘translated’ to the LMX9830 interface.

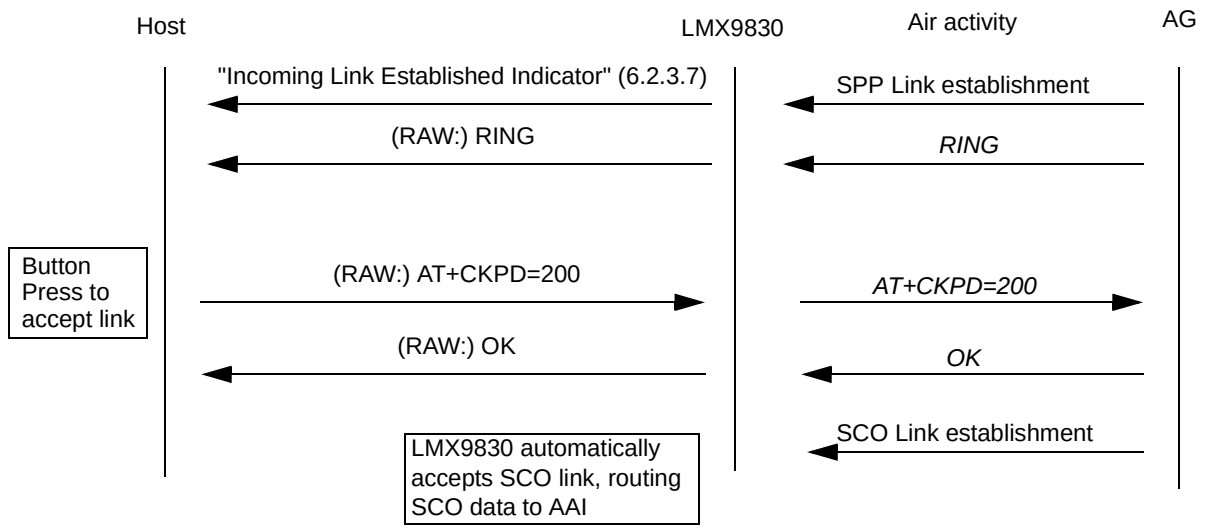


Figure 5-71. Incoming audio connection with LMX9830 as HS

Figure 5-71 shows the flow based on automatic mode, in which the LMX9830 will automatically switch to transparent mode on the UART. Please see the following table for the detailed description of the UART traffic between host and LMX9830.

Table 5-75. Incoming audio connection with LMX9830 as HS

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
00 / 01	RX	Indicator	02,69,50,07,00,C0,B0,CF,22,17,00, 08,00,03	Rx: Event: Incoming Link Established, BdAddr:123456789ABC, Local Port: 02
00 / 01 / 02 / 03	RX	RAW Data	0D,0A,52,49,4E,47,0D,0A	Rx(RAW): 0D,0A,52,49,4E,47,0D,0A *(RING)*
00 / 01 / 02 / 03	TX	RAW Data	41,54,2B,43,4B,50,44,3D,32,30,30, 0D	Tx(RAW): 41,54,2B,43,4B,50,44,3D,32,30,30,0D *(AT+CKPD=200)*
00 / 01 / 02 / 03	RX	RAW Data	0D,0A,4F,4B,0D,0A	Rx(RAW): 0D,0A,4F,4B,0D,0A *(OK)*

After the final OK is received the audio link is established.

5.4.2.5.2 Headset initiated connection establishment

The Headset profile also defines the scenario in which the headset initiates the link the HSAG. See Figure 5-72 on page 100 for the profile specification of the Bluetooth SIG.

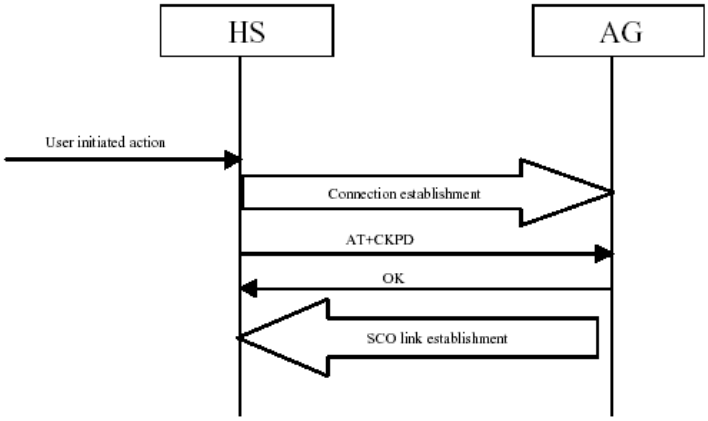


Figure 5-72. Outgoing audio connection establishment as defined by the Bluetooth SIG

As Figure 5-72 indicates, the headset basically only requests a SPP link and sends the AT+CKPD command. The rest is controlled by the HSAG. Therefore the link establishment is limited to the SPP link establishment and AT commands.

In order to establish a link to the HSAG, the headset needs to know the RFCOMM port number it has to connect on the remote device. Usually, these information are requested on first time of connection only, so only the "Establish Link" (6.2.3.1) command will be necessary for the establishment.

Table 5-76 gives an indication of the necessary SDAP request. The most important parameter is the UUID for the Headset Audio Gateway, which is 0x1112.

Figure 5-73 and Table 5-77 show the final profile flow necessary to establish a connection to the HSAG with the LMX9830.

Table 5-76. Requesting the RFCOMM port number for HSAG from the remote device

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
	TX	Request	02,52,32,06,00,8A,12,34,56,78,9A, BC,03	Tx: Cmd: SDAP Connect, BdAddr: 123456789ABC
00	RX	Indicator	02,69,50,07,00,C0,12,34,56,78,9A, BC,00,03	Rx: Event: ACL Established, BdAddr: 123456789ABC, Status: 00
00 / 01	RX	Confirm	02,43,32,01,00,76,00,03	Rx: Event: SDAP Connect, Status: 00
	TX	Request	02,52,35,02,00,89,12,11,03	Tx: Cmd: Service Browse, Browse Group ID: 0811
00 / 01	RX	Confirm	02,43,35,16,00,8E,00,01,02,10,**12**, **11**,**03**,0E,56,6F,69,63,65,20,67,61, 74,65,77,61,79,00,03	Rx: Event: Service Browse, Status: 00, Browse Group ID: 0210, Service ID: **1211**, PortNo: **03**, Service Name: Voice gateway.
	TX	Request	02,52,33,00,00,85,03	Tx: Cmd: SDAP Disconnect
00 / 01	RX	Confirm	02,43,33,01,00,77,00,03	Rx: Event: SDAP Disconnect, Status: 00
00	RX	Indicator	02,69,51,07,00,C1,12,34,56,78,9A, BC,13,03	Rx: Event: ACL Terminated, BdAddr: 123456789ABC, Reason: 13

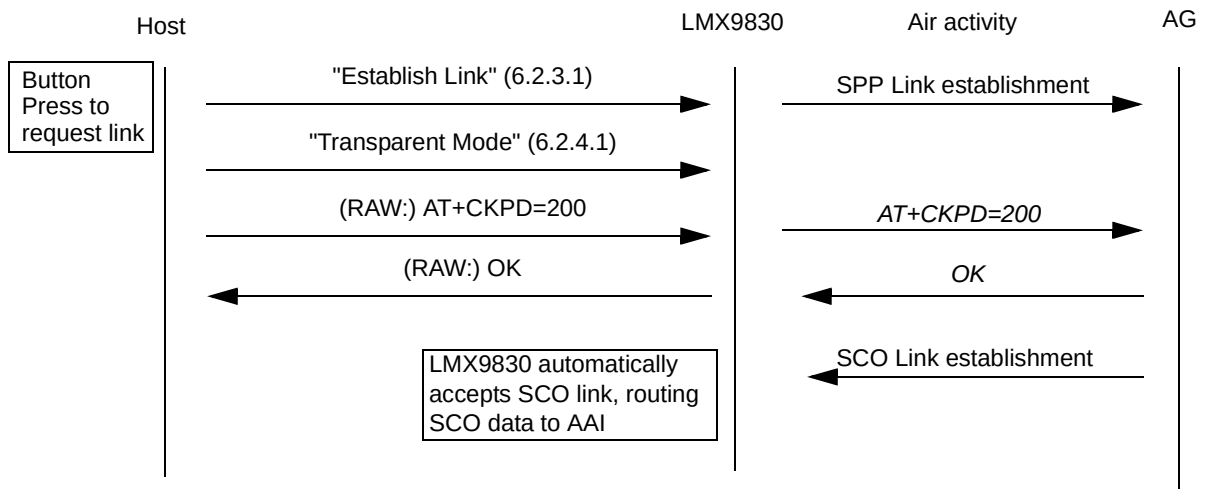


Figure 5-73. Outgoing link establishment from the LMX9830 as HS

Table 5-77. Outgoing link establishment from the LMX9830 as HS

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
	TX	Request	02,52,0A,08,00,64,01,12,34,56,78, 9A,BC,**03**,03	Tx: Cmd: Establish Link, Local Port: 01, BdAddr: 123456789ABC, Remote Port Number: **03**
00 / 01	RX	Confirm	02,43,0A,02,00,4F,00,01,03	Rx: Event: Establish Link, Status: 00, Local Port: 01
00	RX	Indicator	02,69,50,07,00,C0,12,34,56,78,9A, BC,00,03	Rx: Event: ACL Established, BdAddr: 123456789ABC, Status: 00
00 / 01	RX	Indicator	02,69,0B,09,00,7D,00,12,34,56,78, 9A,BC,01,**03**,03	Rx: Event: Link Established, Status: 00, BdAddr: 123456789ABC, Local Port: 01, Remote Port Number: **03**
00 / 01	RX	Indicator	02,69,3E,04,00,AB,01,8C,00,00,03	Rx: Event: Port Status Changed, Local Port: 01, PortStatus: 8C, Break Length: 0000
	TX	Request	02,52,11,01,00,64,01,03	Tx: Cmd: Transparent Mode, Local Port: 01
00 / 01	RX	Confirm	02,43,11,02,00,56,00,01,03	Rx: Event: Transparent Mode, Status: 00, Local Port: 01
	TX	RAW Data	41,54,2B,43,4B,50,44,3D,32,30,30, 0D	Tx(RAW): 41,54,2B,43,4B,50,44,3D,32,30,30,0D *(AT+CKPD=200)*
00 / 01 / 02 / 03	RX	RAW Data	0D,0A,4F,4B,0D,0A	Rx(RAW): 0D,0A,4F,4B,0D,0A *(OK)*

5.4.2.5.3 Audio connection release from the HS

The connection release is based on the same command as the connection confirmation, AT+CKPD=200. In case the user wants to release the link from the HS to the HSAG, it presses the same button again, which will send the same AT command. The HSAG will confirm by an OK and release the link.

The LMX9830 confirms the released link by the standard"SPP Link Released Indicator" (6.2.3.6). See following the Bluetooth specification description for the realization with LMX9830. In case the HSAG releases the link the LMX9830 will send the same event.

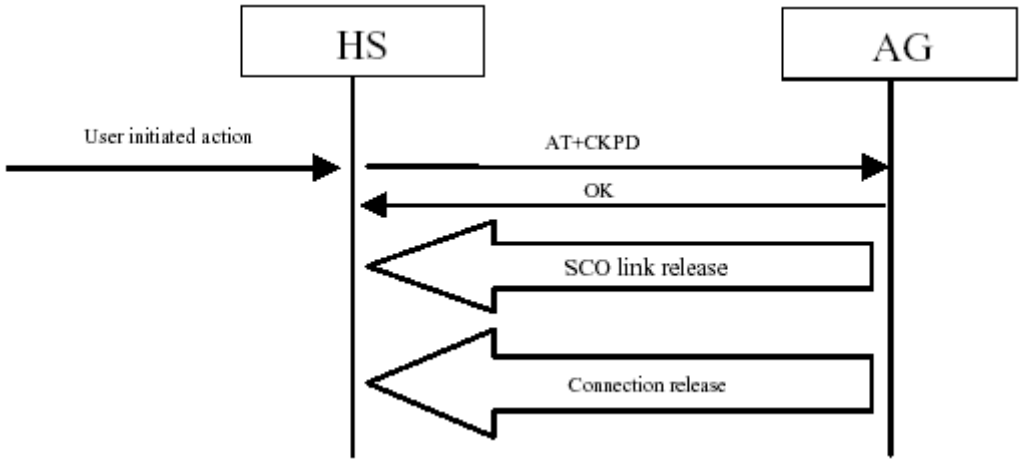


Figure 5-74. Audio link release by the HS

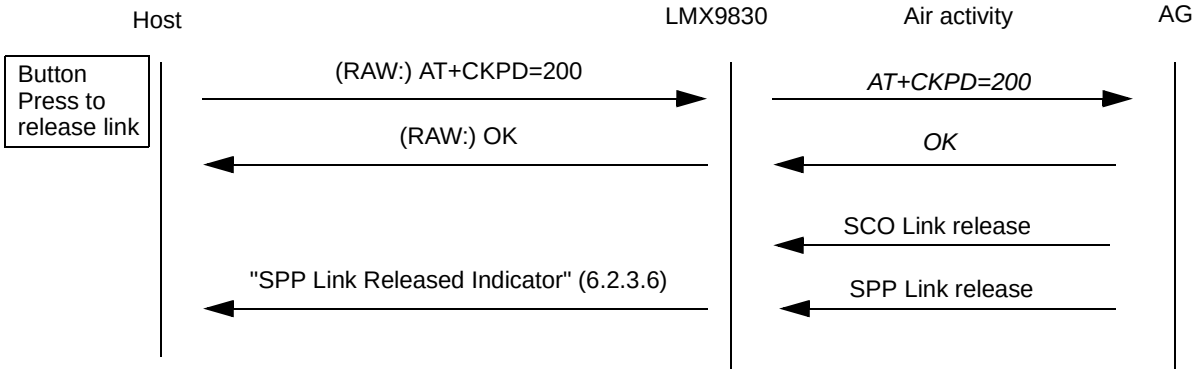


Figure 5-75. Audio link release from the LMX9830 as HS

Table 5-78. Audio link release from the LMX9830 as HS

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
	TX	RAW Data	41,54,2B,43,4B,50,44,3D,32,30,30, 0D	Tx(RAW): 41,54,2B,43,4B,50,44,3D,32,30,30,0D
00 / 01 / 02 / 03	RX	RAW Data	0D,0A,4F,4B,0D,0A	Rx(RAW): 0D,0A,4F,4B,0D,0A
00 / 01 / 02	RX	UART BREAK		Rx(RAW): 00
00 / 01	RX	Indicator	02,69,11,02,00,7C,01,00,03	Rx: Event: Transparent Mode, Local Port: 01, Mode: 00
00 / 01	RX	Indicator	02,69,0E,02,00,79,01,01,03	Rx: Event: Link Released, Reason: 01, Local Port: 01
00	RX	Indicator	02,69,51,07,00,C1,12,34,56,78,9A, BC,13,03	Rx: Event: ACL Terminated, BdAddr: 123456789ABC, Reason: 13

5.4.2.5.4 Audio connection release from the HSAG

In case the HSAG releases the audio link, no further AT commands are required. The HSAG just releases the SPP and SCO link. The host will be notified by the standard "SPP Link Released Indicator" (6.2.3.6) and "SCO Link Released Indicator" (6.2.6.2.4).

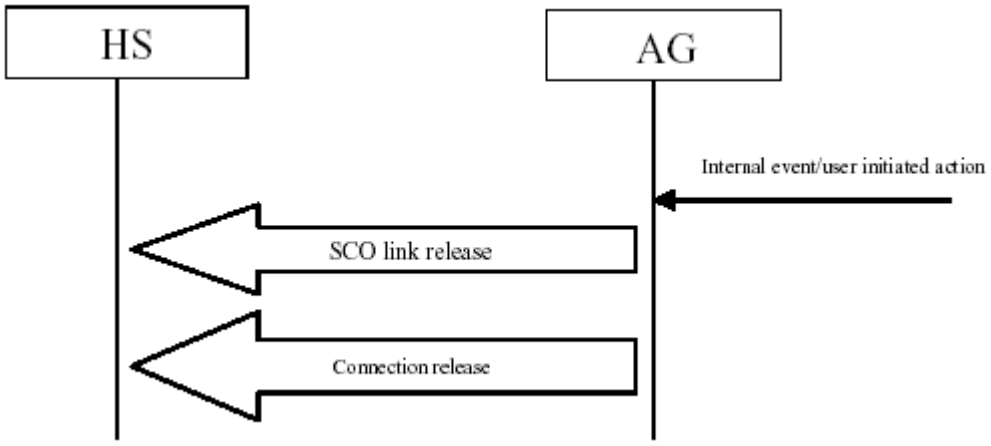


Figure 5-76. Audio link release by the HSAG

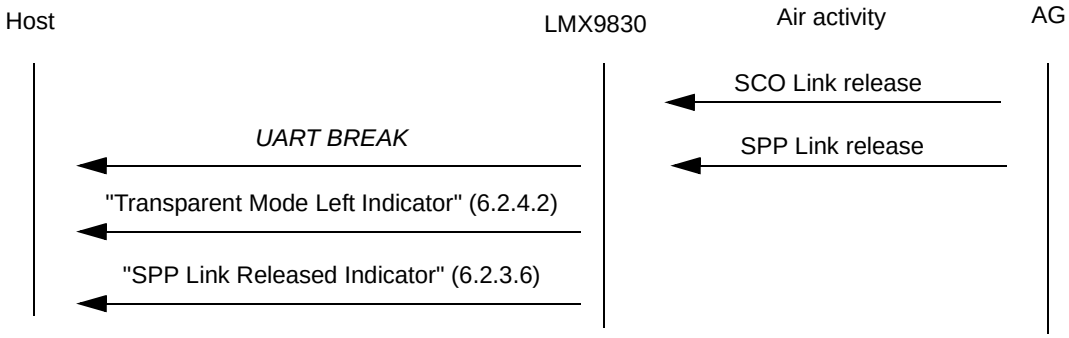


Figure 5-77. Audio link release by the HSAG

Table 5-79. Audio link release from the LMX9830 as HS

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
00 / 01 / 02	RX	UART BREAK		Rx(RAW): 00
00 / 01	RX	Indicator	02,69,11,02,00,7C,01,00,03	Rx: Event: Transparent Mode, Local Port: 01, Mode: 00
00 / 01	RX	Indicator	02,69,0E,02,00,79,01,01,03	Rx: Event: Link Released, Reason: 01, Local Port: 01
00	RX	Indicator	02,69,51,07,00,C1,12,34,56,78,9A, BC,13,03	Rx: Event: ACL Terminated, BdAddr: 123456789ABC, Reason: 13

5.4.3 Acting as Audio Gateway

The audio gateway is defined as a device, handling audio data and using the HS as analog front end for both for input and output. Typical devices acting as Audio Gateways are cellular phones and personal computer. The LMX9830 can be used in an audio gateway by creating the required data link to the headset and routing all data from the Advanced Audio Inter-

face (AAI) to the bluetooth link. The audio interface needs to be connected to the host PCM interface. Timing and electrical specification for the interface are derived out of the driver configuration set in NVS. Bit-clock and frame-clock settings for different codec options are described in [1]. Please see Figure 5-64 "Headset Data and Audio stream" on page 93 how to implement the LMX9830 in a audio gateway application.

The Audio Gateway (AG) controls the inter-operation with the headset in terms of establishing and releasing the required SCO link. While the SPP link establishment can be initiated from both sides, the SCO link establishment will always be initiated by the HSAG. The LMX9830 offers dedicated commands to control the audio link.

In order to be able to create a link to a headset, it is required to create the service database record and to configure the default audio settings for the PCM codec driver.

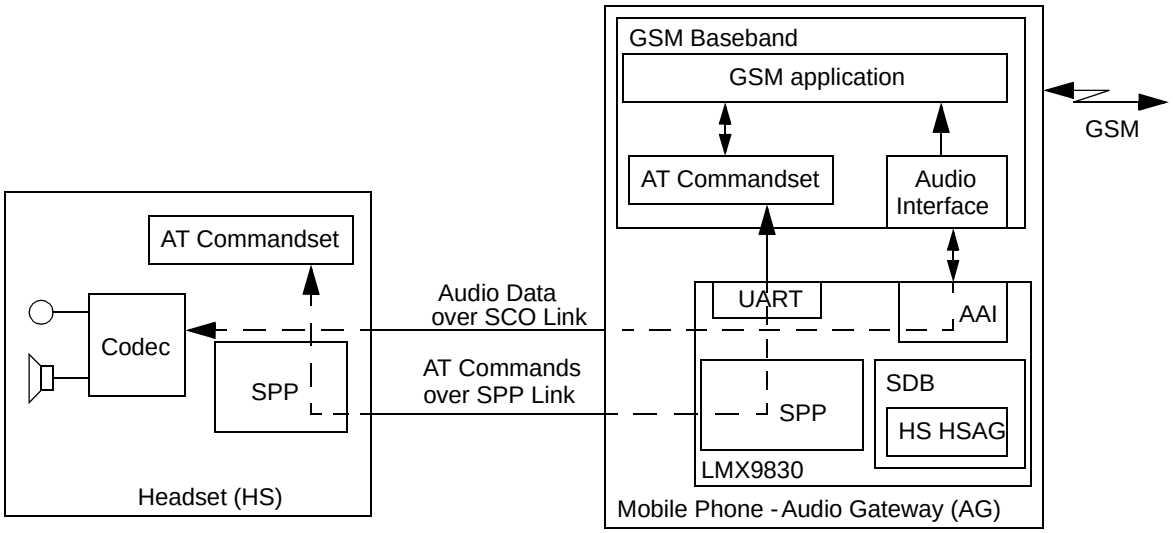


Figure 5-78. Audio Gateway Data and Audio stream using LMX9830 in the HSAG

5.4.3.1 Configuring the audio path

In general the LMX9830 can be seen as gateway between the UART interface and the bluetooth link. The audio link is one additional interface for the LMX9830 to be routed. Please see also Figure 4-41 on page 70 for the two routing options.

The LMX9830 supports different configurations for the PCM codec interface and the air interface. The bluetooth specification defines the following formats to be used over the bluetooth link:

- CVSD (default)
- µ-Law
- A-Law

As indicated, the default and mostly used coding format used over a bluetooth link is CVSD. Since the headset most likely will always use the same settings, it is recommended to set the parameters to a default in the NVS, using the "Set audio settings" (6.2.6.2.5) command. Bitclock and frame-clock information for the different codecs are listed in [1].

Table 5-80. Set Default Audio Settings to Motorola codec and CVSD

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
	TX	Request	02,52,5B,02,00,AF,**01**,**00**,03	Tx: Cmd: Set Default Audio Settings, Codec Type: **01**, Air Format: **00**
00 / 01	RX	Confirm	02,43,5B,01,00,9F,00,03	Rx: Event: Set Default Audio Settings, Status: 00

5.4.3.2 Configuring the Service Database for Audio Gateway

In order to offer an additional HSAG service to other devices a new entry has been made into the service database. For this the "Store Generic SDP Record" (6.2.13.1) needs to be used. This command is a generic command to generate any possi-

ble profile within the device. To be able to do so, significant knowledge of the internal SDB structure would be required. In order to use this command, the “Simply Blue Commander” software needs to be used, to create this command.

The generation of profile entries is supported in Simply Blue Commander versions later than 1.3.0.3. Please follow the following figures to create a headset service database entry:

1) Select “Definitions/Create Service Record” within the Simply Blue Commander menu.

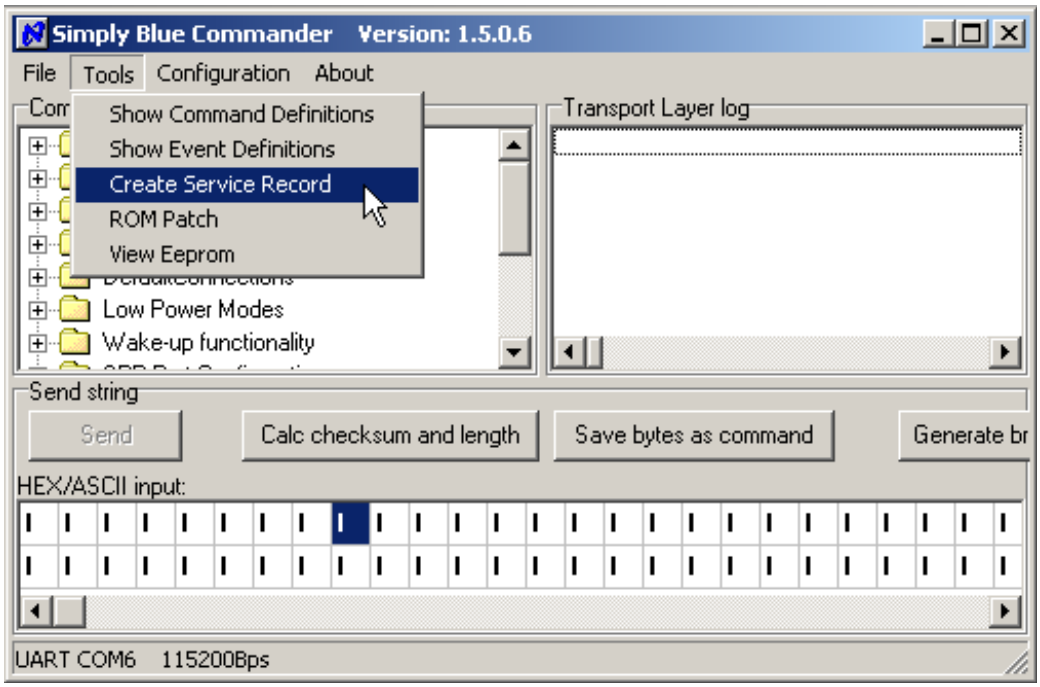


Figure 5-79. Opening “Create Service Record” Dialog-box

2) Select the service record required, in this case “Headset Audio Gateway”

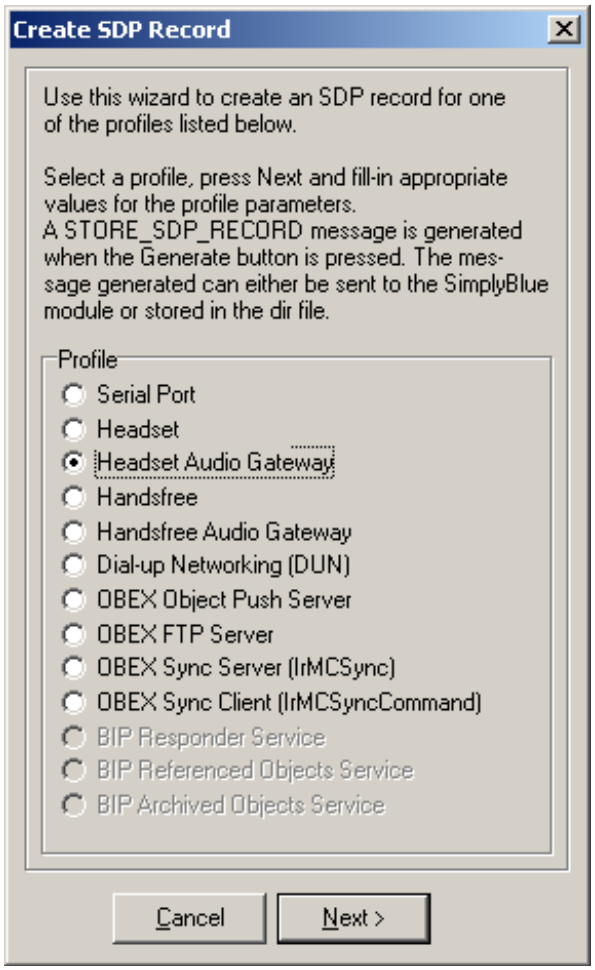


Figure 5-80. Selecting the profile

3) Select the settings desired for the HSAG profile and choose the correct RFCOMM port. As this example device shall still be able to accept a second standard data links as well, RFCOMM port 2 should be used. Finally confirm the dialog with pressing “Create”.

Figure 5-81. Configuring the profile settings for HSAG

4) The “Create” will fill the Hex/ASCII line of the Simply Blue Commander with the string necessary to send to the LMX9830. Afterwards just close the dialog with the “Close” Button.

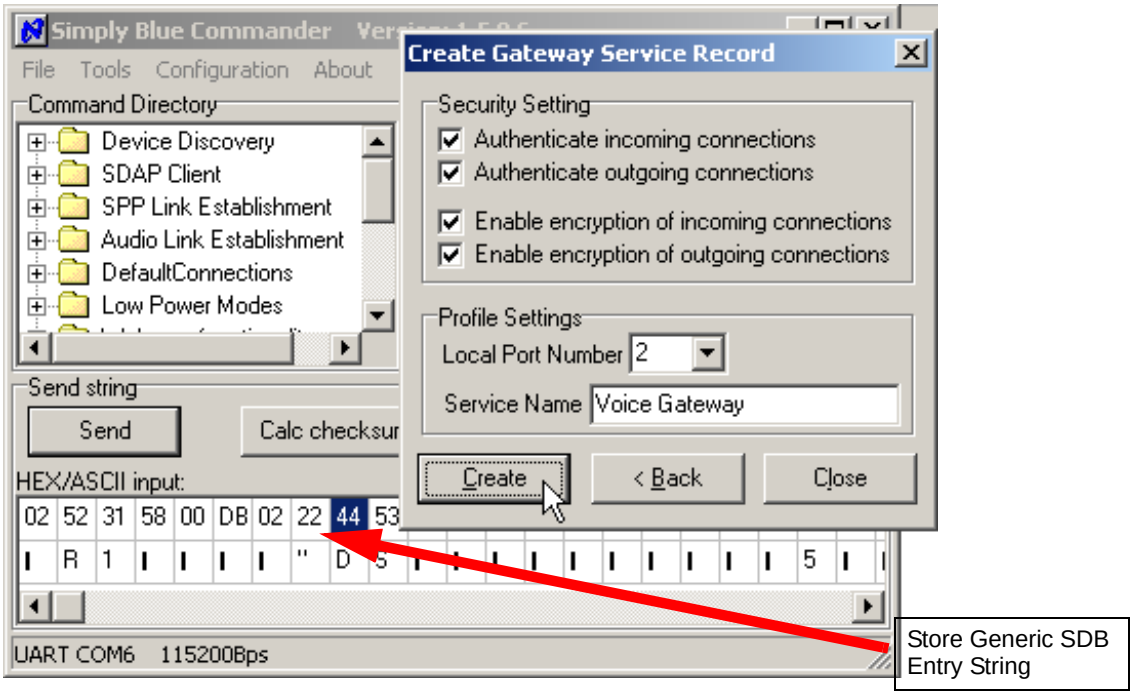


Figure 5-82. Creating the command string for the audio gateway entry

5) The string can be sent directly to the board or stored as command within the directory. Once sent it can also be copied out of the log entry and copied into the development code when switching of the interpretation option.

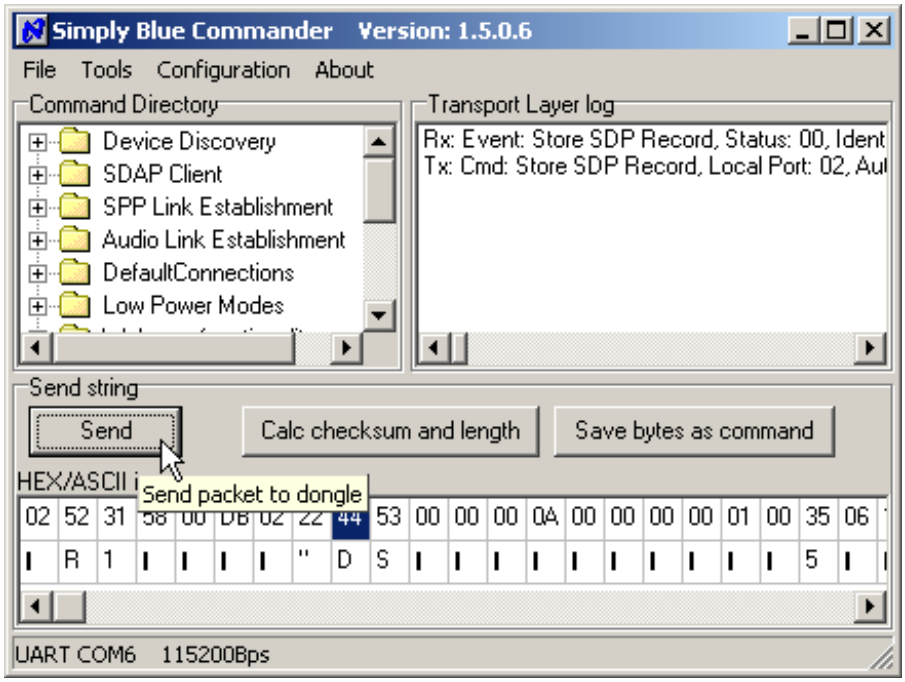


Figure 5-83. Send the command to the LMX9830

Table 5-71 shows the command string being the result out of the steps just described.

Table 5-81. Adding the HSAG service to the Service Database

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
	TX	Request	02,52,31,58,00,DB,**02**,**22**,**44**,53,00,00,00,0A,00,00,00,00,01,00,35,06,19,12,11,19,03,12,04,00,35,0C,35,03,19,00,01,35,05,19,03,00,08,02,05,00,35,03,19,02,10,06,00,35,09,09,6E,65,09,6A,00,09,00,01,09,00,35,08,35,06,19,08,11,09,00,01,00,01,25,0E,56,6F,69,63,65,20,47,61,74,65,77,61,79,00,03	Tx: Cmd: Store SDP Record, Local Port: **02**, Authentication: **22**, Encryption: **44**, SdpRecord: 00000A000000001003506191211190312040035 0C350319000135051903000802050035031902100600350909 6E65096A00090001090035083506190811090001000125 0E566F6963652047617465776179 00
00 / 01	RX	Confirm	02,43,31,02,00,76,00,**01**,03	Rx: Event: Store SDP Record, Status: 00, Identifier: **01**

After this command the Service database has the following content:

Table 5-82. Service Database for COM1 and HSAG

Index	Enabled	Local RFComm Port	Type	Name	Authentication	Encryption	Profile Specifics
00	1	01	SPP	COM1	0x02	0x04	none
01	1	02	HSAG	Voice Gateway	0x22	0x44	none

5.4.3.3 Adapt RFCOMM ports to open

For a simpler differentiation on the host the services have been set to two separate RFCOMM port. In order to make both ports available for connection, the "Set Ports To Open" (6.2.12.1) command should be used to make the appropriate configuration.

Table 5-83. Set RFCOMM ports to open

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
	TX	Request	02,52,22,04,00,78,03,00,00,00,03	Tx: Cmd: Set Ports To Open, Ports: **03000000**
00 / 01	RX	Confirm	02,43,22,01,00,66,00,03	Rx: Event: Set Ports To Open, Status: 00

5.4.3.4 Store Class of Device

During the Inquiry procedure the searching device will receive the class of device. Please see also Section 4.1.2.3 "Class of Device" on page 49 for more information on that parameter. In order to be recognized as audio device it is beneficial to indicate the audio capability within the class of device. The following table shows the command how to set the “Class of Device” using the "Store Class of Device" (6.2.14.5) command. The class is set to 522204, which reflects the class for a mobile phone.

Table 5-84. Store Class of Device

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
	TX	Request	02,52,28,03,00,7D,**04**,**22,52**,03	Tx: Cmd: Store Class of Device, Class of Device: **042252**
00 / 01	RX	Confirm	02,43,22,01,00,66,00,03	Rx: Event: Store Class of Device, Status: 00

5.4.3.5 Connection handling as audio gateway

The Headset profile is based on some specific AT commands also listed inSection 6.4.2 “Headset Profile” on page 203.

In a headset application the link will either be initiated from the HSAG or the HS itself. In both cases the HSAG will control the SCO link establishment. The HS will only react on incoming AT commands and maybe send the request establishment or release.

The LMX9830 offers specific commands to establish and release SPP and SCO links. The first connection establishment is based on the standard SPP link. Once this is established the HSAG indicates the incoming call by sending the RING At command over the SPP link. Once the HS responds with the appropriate AT command, the HSAG needs to establish the SCO connection. In some implementations it is also possible to establish the SCO right after the SPP link to support inband ringtones.

The following chapters give the generic guideline of how to act as HSAG within the different scenarios.

5.4.3.5.1 Requesting the RFCOMM port from the Headset

Table 5-85 gives an indication of the necessary SDAP request. The most important parameter is the UUID for the Headset Audio Gateway, which is 0x0811.

Table 5-85. Requesting the RFCOMM port number for HSAG from the remote device

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
	TX	Request	02,52,32,06,00,8A,12,34,56,78,9A, BC,03	Tx: Cmd: SDAP Connect, BdAddr: 123456789ABC
00	RX	Indicator	02,69,50,07,00,C0,12,34,56,78,9A, BC,00,03	Rx: Event: ACL Established, BdAddr: 123456789ABC, Status: 00
00 / 01	RX	Confirm	02,43,32,01,00,76,00,03	Rx: Event: SDAP Connect, Status: 00
	TX	Request	02,52,35,02,00,89,12,11,03	Tx: Cmd: Service Browse, Browse Group ID: 0811
00 / 01	RX	Confirm	02,43,35,16,00,8E,00,01,02,10,**08**, **11**,**07**,0E,56,6F,69,63,65,20,67,61, 74,65,77,61,79,00,03	Rx: Event: Service Browse, Status: 00, Browse Group ID: 0210, Service ID: **0811**, PortNo: **07**, Service Name: Voice gateway.
	TX	Request	02,52,33,00,00,85,03	Tx: Cmd: SDAP Disconnect
00 / 01	RX	Confirm	02,43,33,01,00,77,00,03	Rx: Event: SDAP Disconnect, Status: 00
00	RX	Indicator	02,69,51,07,00,C1,12,34,56,78,9A, BC,13,03	Rx: Event: ACL Terminated, BdAddr: 123456789ABC, Reason: 13

5.4.3.5.2 Outgoing audio connection

Figure 5-70 on page 98 shows the flow diagram of a HSAG link initiated by the HSAG as defined in [3].

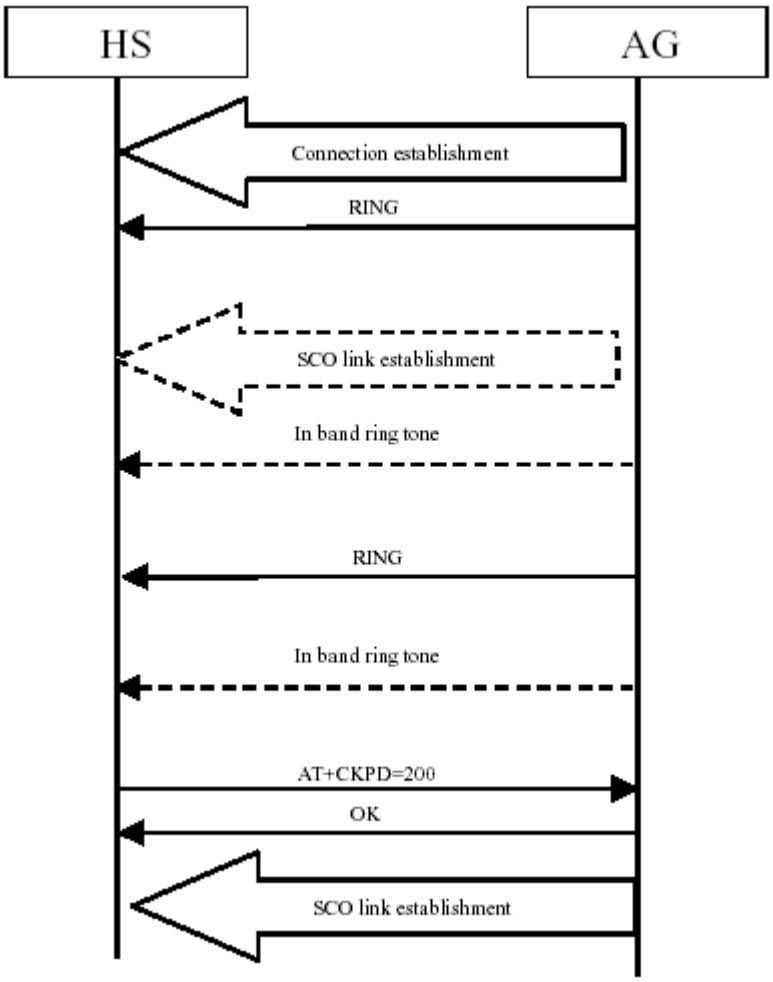


Figure 5-84. "AG initiated audio connection establishment" as defined in[3]

As Figure 5-84 shows, the link is initiated by the HSAG, e.g. in case of an incoming call on a mobile phone. The HSAG will send the "RING" command to force an audio or visual signalling on the headset. The HSAG keeps on ringing until the HS user accepts the incoming link by pressing a button. On this the HS application will send the "AT+CKPD=200" command. The HSAG needs to confirm the successful reception of the command with the "OK" and finally sends the SCO establishment command.

Please see Figure 5-71 which shows the HSAG part of the profile for the outgoing audio connection 'translated' to the LMX9830 command interface.

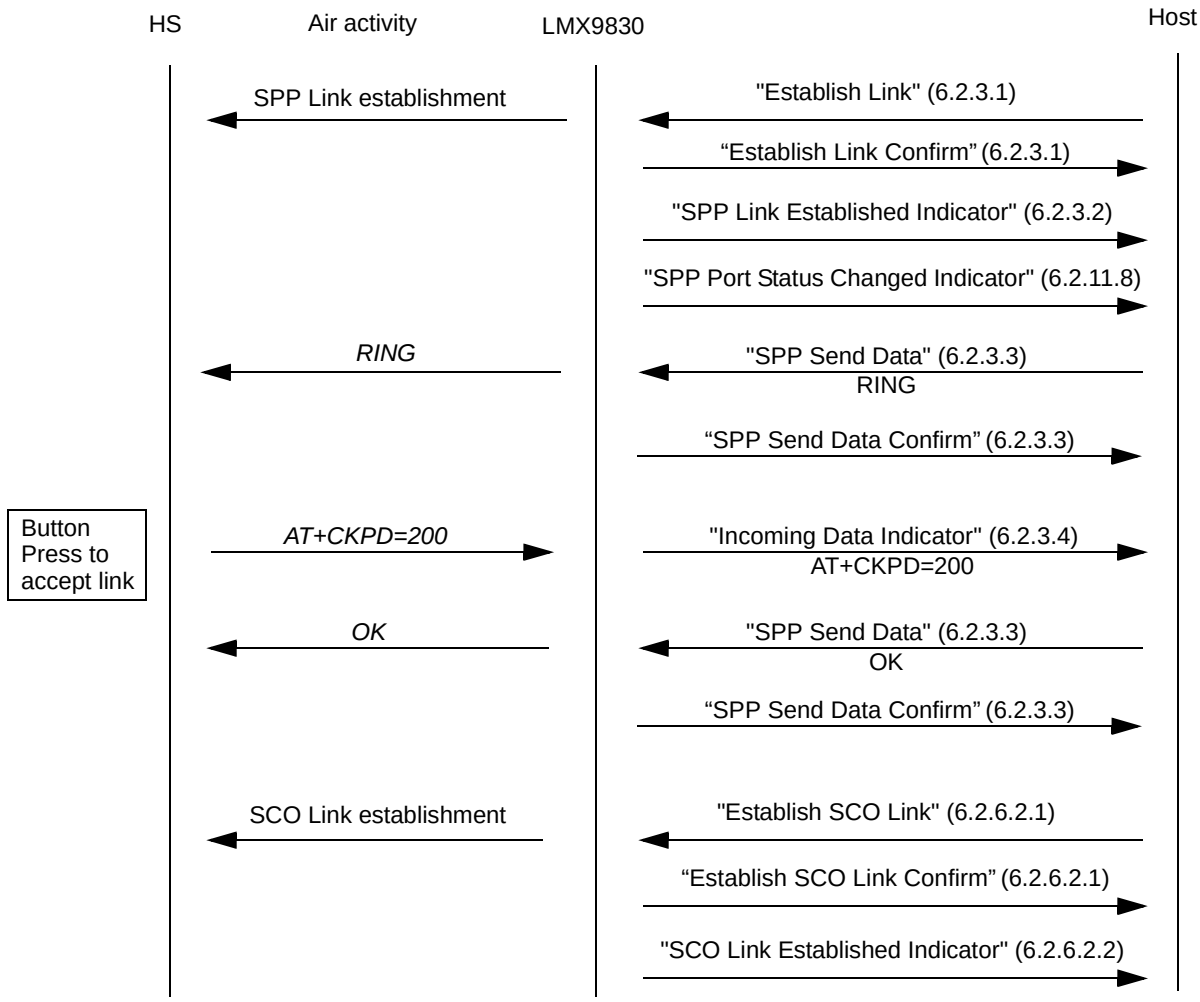


Figure 5-85. Outgoing audio connection with LMX9830 as HSAG

Please see the following table for the detailed description of the UART traffic between host and LMX9830.

Table 5-86. Outgoing audio connection with LMX9830 as HSAG

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
	TX	Request	02,52,0A,08,00,64,**02**,12,34,56,78, 9A.BC,**07**,03	Tx: Cmd: Establish Link, Local Port: **02**, BdAddr: 123456789ABC, Remote Port Number: **07**
00 / 01	RX	Confirm	02,43,0A,02,00,4F,00,**02**,03	Rx: Event: Establish Link, Status: 00, Local Port: **02**
00	RX	Indicator	02,69,50,07,00,C0,12,34,56,78,9A. BC,00,03	Rx: Event: ACL Established, BdAddr: 123456789ABC, Status: 00
00 / 01	RX	Indicator	02,69,0B,09,00,7D,00,12,34,56,78, 9A.BC,**02**,**07**,03	Rx: Event: Link Established, Status: 00, BdAddr: 123456789ABC, Local Port: **02**, Remote Port Number: **07**
00 / 01	RX	Indicator	02,69,3E,04,00,AB,02,8C,00,00,03	Rx: Event: Port Status Changed, Local Port: 02, PortStatus: 8C, Break Length: 0000
	TX	Request	02,52,0F,0B,00,6C,02,08,00,0D,0A ,52,49,4E,47,0D,0A,03	Tx: Cmd: Send Data, Local Port: 02, Payload Data: 0D0A52494E470D0A *(RING)*

Table 5-86. Outgoing audio connection with LMX9830 as HSAG

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
00 / 01	RX	Confirm	02,43,0F,02,00,54,00,02,03	Rx: Event: Send Data, Status: 00, Local Port: 02
00 / 01	RX	Indicator	02,69,10,0F,00,88,02,0C,00,41,54, 2B,43,4B,50,44,3D,32,30,30,0D,03	Rx: Event: Incoming Data, Local Port: 02, Received Data: 41542B434B50443D3230300D *(AT+CKPD=200)*
	TX	Request	02,52,0F,09,00,6A,02,06,00,0D,0A, 4F,4B,0D,0A,03	Tx: Cmd: Send Data, Local Port: 02, Payload Data: 0D0A4F4B0D0A *(OK)*
00 / 01	RX	Confirm	02,43,0F,02,00,54,00,02,03	Rx: Event: Send Data, Status: 00, Local Port: 02
	TX	Request	02,52,5D,08,00,B7,12,34,56,78,9A. BC,80,00,03	Tx: Cmd: Establish SCO Link, BdAddr: 123456789ABC, Packet Type: 8000
00 / 01	RX	Confirm	02,43,5D,07,00,A7,00,12,34,56,78, 9A.BC,03	Rx: Event: Establish SCO Link, Status: 00, BdAddr: 123456789ABC
00 / 01	RX	Indicator	02,69,5D,07,00,CD,00,12,34,56,78 ,9A.BC,03	Rx: Event: SCO Link Established, Status: 00, BdAddr: 123456789ABC

In some applications, the headset might send additional commands for volume control or status. Please refer to Section 6.4.2 on page 203 for the complete list of AT commands required by the headset profile.

5.4.3.5.3 Incoming audio connection from Headset

The Headset profile also defines the scenario in which the headset initiates the link to the HSAG.

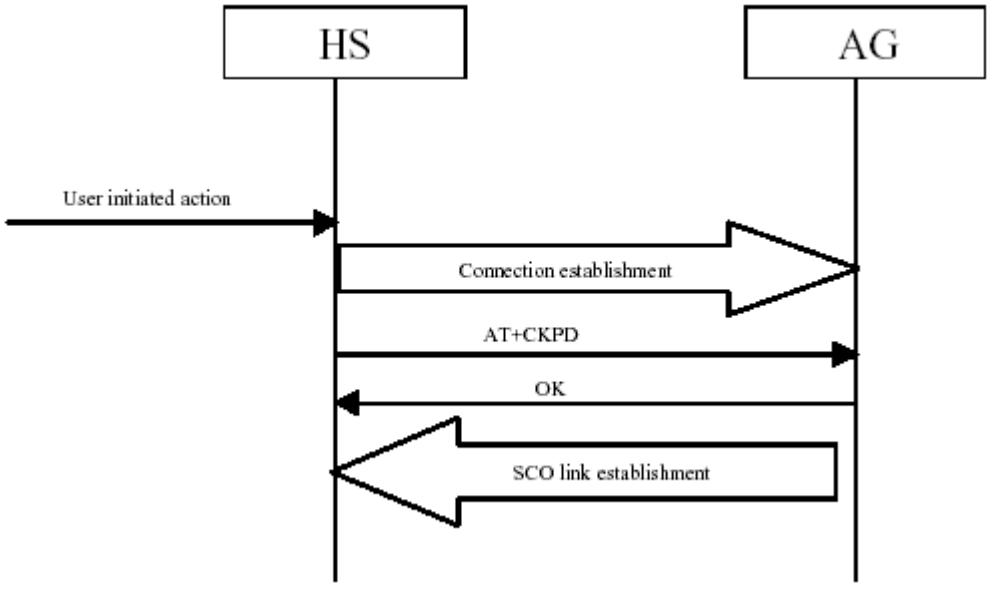


Figure 5-86. Incoming audio connection on HSAG as defined in [3]

As Figure 5-72 indicates, the headset requests a SPP link and sends the AT+CKPD=200 command. The HSAG needs to accept the incoming link and afterwards establish the SCO as already described for an outgoing link.

The headset will contact the HSAG on the RFCOMM port, the HSAG has been assigned to.

Figure 5-87 and Table 5-87 show the profile flow necessary to accept an incoming link from a headset using the LMX9830 in an HSAG application. The LMX9830 in this example is using the default setting, meaning configured for automatic mode. Because of this, it will switch to transparent mode as soon as the incoming SPP link has been established. Therefore a UART BREAK is sent before the SCO links can be sent.

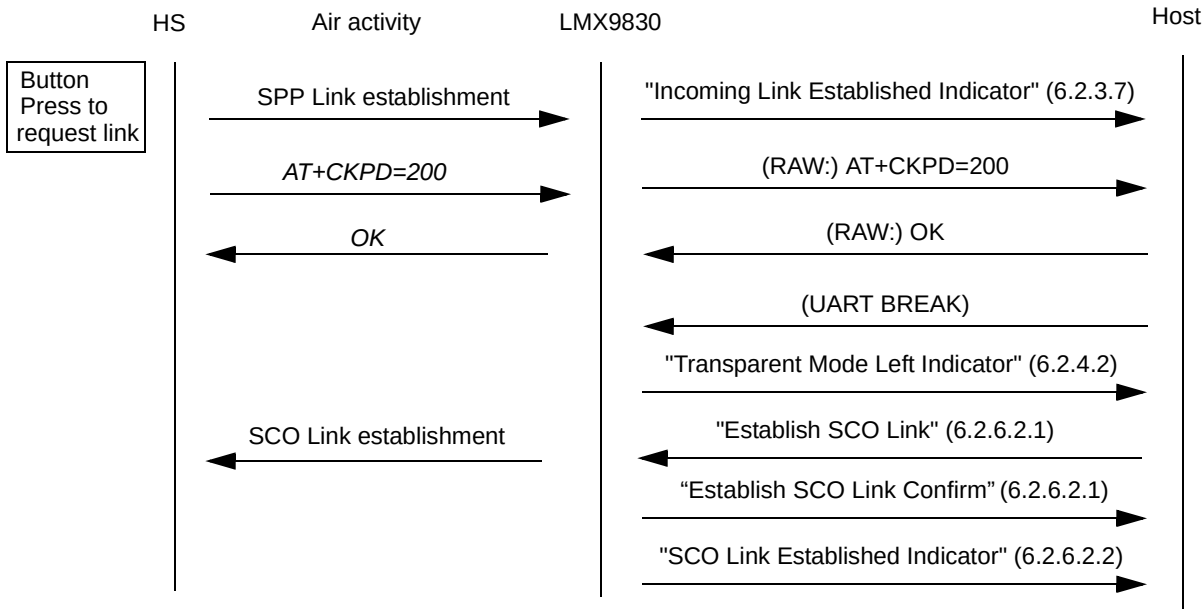


Figure 5-87. Incoming HS connection as HSAG

Table 5-87. Incoming headset connection

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
	TX	Request	02,69,0C,07,00,7C,12,34,56,78,9A ,BC,02,03	Rx: Event: Incoming Link Established, BdAddr: 123456789ABC, Local Port: 02
00 / 01 / 02 / 03	RX	RAW Data	41,54,2B,43,4B,50,44,3D,32,30,30, 0D	Rx(RAW): 41,54,2B,43,4B,50,44,3D,32,30,30,0D
	TX	RAW Data	0D,0A,4F,4B,0D,0A	Tx(RAW): 0D,0A,4F,4B,0D,0A
	TX	UART BREAK		Rx(RAW): 00
00 / 01	RX	Indicator	02,69,11,02,00,7C,02,00,03	Rx: Event: Transparent Mode, Local Port: 02, Mode: 00
	TX	Request	02,52,5D,08,00,B7,12,34,56,78,9A. BC,80,00,03	Tx: Cmd: Establish SCO Link, BdAddr: 123456789ABC, Packet Type: 8000
00 / 01	RX	Confirm	02,43,5D,07,00,A7,00,12,34,56,78, 9A.BC,03	Rx: Event: Establish SCO Link, Status: 00, BdAddr: 123456789ABC
00 / 01	RX	Indicator	02,69,5D,07,00,CD,00,12,34,56,78 ,9A.BC,03	Rx: Event: SCO Link Established, Status: 00, BdAddr: 123456789ABC

5.4.3.5.4 Audio connection release from the HS

The connection release is based on the same command as the connection confirmation, AT+CKPD=200. In case the user wants to release the link from the HS to the HSAG, it presses the same button again, which will send the same AT command. The HSAG will confirm by an OK and release the link.

The LMX9830 confirms the released link by the standard"SPP Link Released Indicator" (6.2.3.6). See following the Bluetooth specification description for the realization with LMX9830. In case the HSAG releases the link the LMX9830 will send the same event.

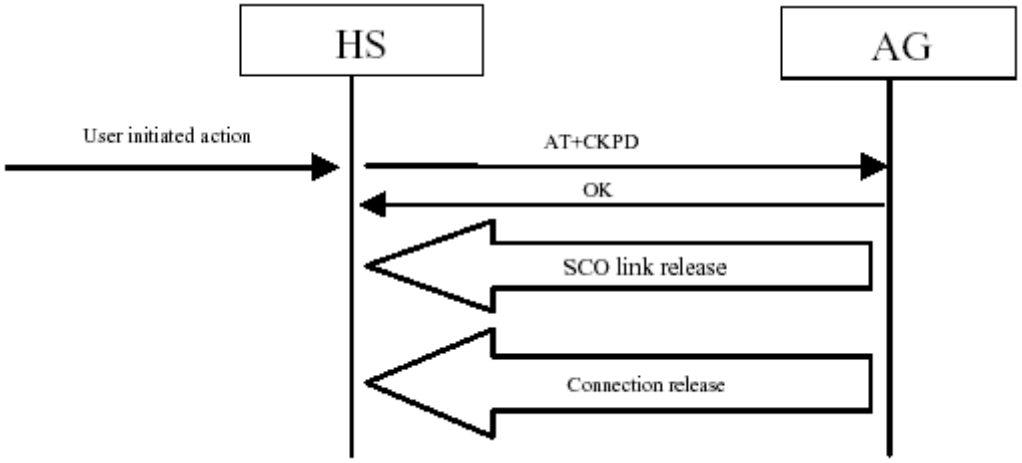


Figure 5-88. Audio link release by the HS

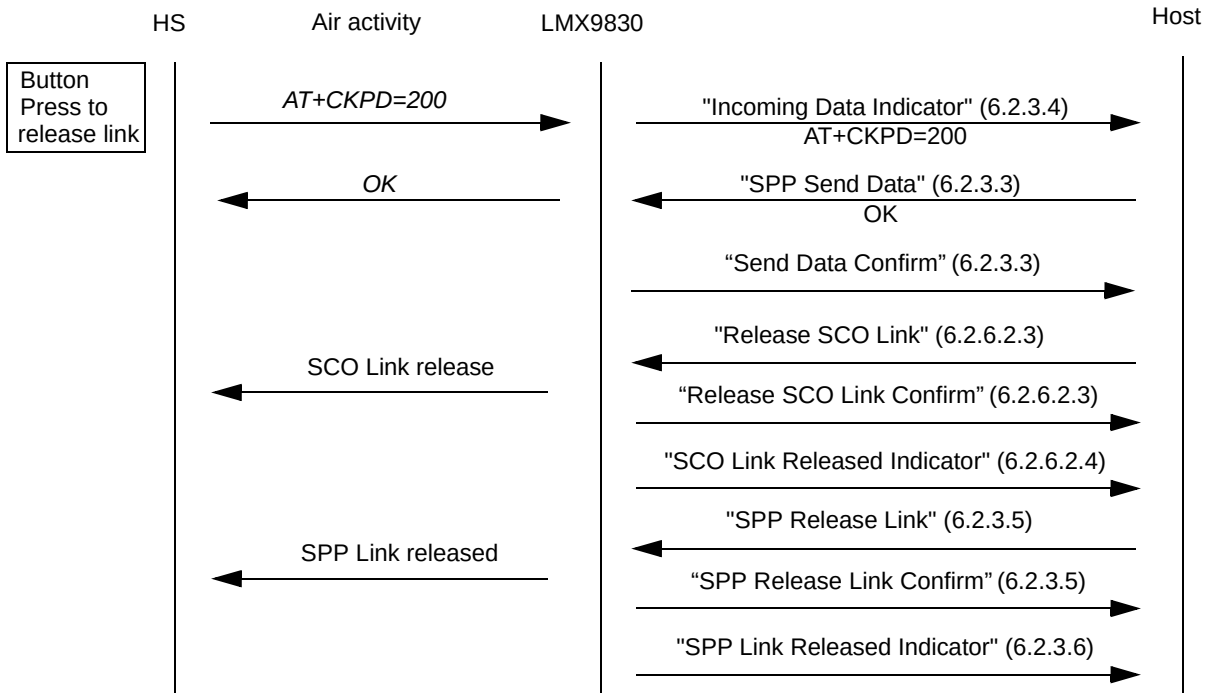


Figure 5-89. Audio link release by the HS

Table 5-88. Audio link release by the HS

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
00 / 01	RX	Indicator	02,69,10,0F,00,88,02,0C,00,41,54, 2B,43,4B,50,44,3D,32,30,30,0D03	Rx: Event: Incoming Data, Local Port: 02, Received Data: 41542B434B50443D3230300D
	TX	Request	02,52,0F,09,00,6A,02,06,00,0D,0A, 4F,4B,0D,0A,03	Tx: Cmd: Send Data, Local Port: 02, Payload Data: 0D0A4F4B0D0A

Table 5-88. Audio link release by the HS

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
00 / 01	RX	Confirm	02,43,0F,02,00,54,00,02,03	Rx: Event: Send Data, Status: 00, Local Port: 02
	TX	Request	02,52,5E,00,00,B0,03	Tx: Cmd: Release SCO Link
00 / 01	RX	Confirm	02,43,5E,01,00,A2,00,03	Rx: Event: Release SCO Link, Status: 00
00 / 01	RX	Indicator	02,69,5E,02,00,C9,00,00,03	Rx: Event: SCO Link Released, Status: 00, HCI Reason: 00
	TX	Request	02,52,0D,01,00,60,02,03	Tx: Cmd: Release Link, Local Port: 02
00 / 01	RX	Confirm	02,43,0D,02,00,52,00,02,03	Rx: Event: Release Link, Status: 00, LocalPort: 02
00 / 01	RX	Indicator	02,69,0E,02,00,79,00,02,03	Rx: Event: Link Released, Reason: 00, Local Port: 02
00 / 01	RX	Indicator	02,69,51,07,00,C1,69,9A,01,A4,07, 00,16,03	Rx: Event: ACL Terminated, BdAddr: 699A01A40700, Reason: 16

5.4.3.5.5 Audio connection release initiated by the HSAG

The release of the audio connection uses the standard LMX9830 commandset, no further AT command is required.

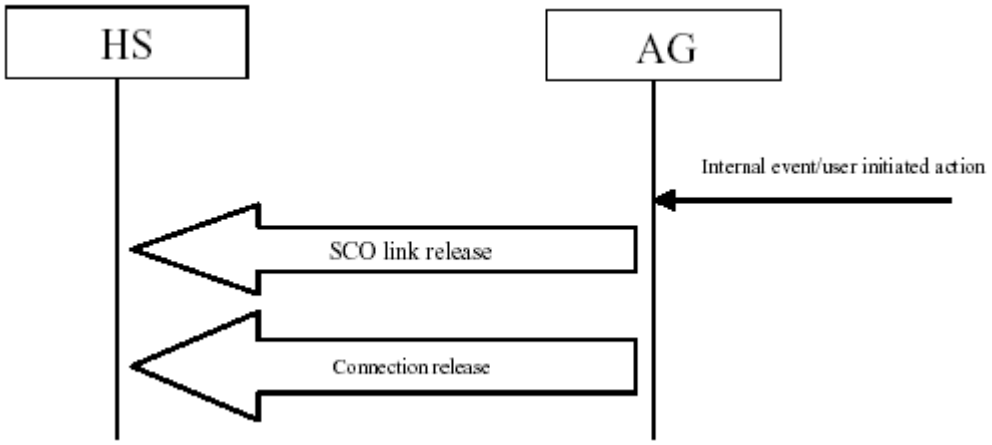


Figure 5-90. Audio link release initiated by the HSAG

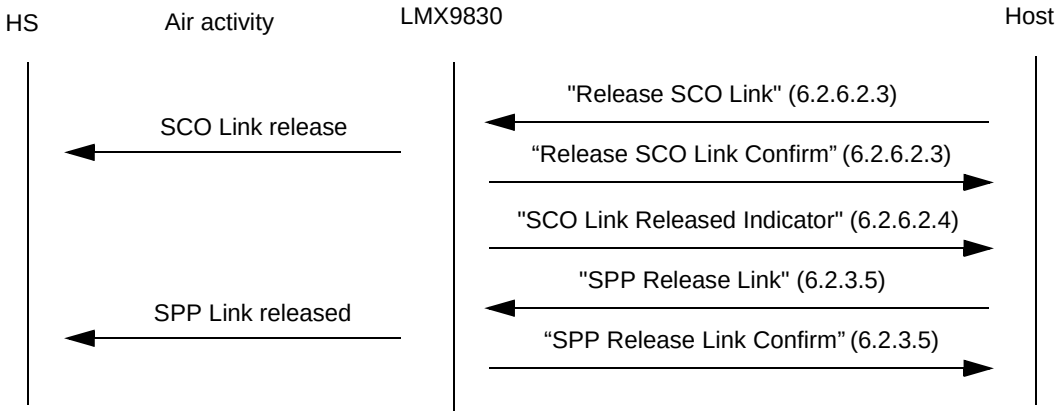


Figure 5-91. Audio link release initiated by the HSAG

Table 5-89. Audio link release by the HSAG

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
	TX	Request	02,52,5E,00,00,B0,03	Tx: Cmd: Release SCO Link
00 / 01	RX	Confirm	02,43,5E,01,00,A2,00,03	Rx: Event: Release SCO Link, Status: 00
00 / 01	RX	Indicator	02,69,5E,02,00,C9,00,00,03	Rx: Event: SCO Link Released, Status: 00, HCI Reason: 00
	TX	Request	02,52,0D,01,00,60,02,03	Tx: Cmd: Release Link, Local Port: 02
00 / 01	RX	Confirm	02,43,0D,02,00,52,00,02,03	Rx: Event: Release Link, Status: 00, LocalPort: 02
00 / 01	RX	Indicator	02,69,0E,02,00,79,00,02,03	Rx: Event: Link Released, Reason: 00, Local Port: 02
00 / 01	RX	Indicator	02,69,51,07,00,C1,69,9A,01,A4,07, 00,16,03	Rx: Event: ACL Terminated, BdAddr: 699A01A40700, Reason: 16

6.0 LMX9830 Command Interface

6.1 UART PROTOCOL PRINCIPLES

The LMX9830 can be controller with simple commands on the UART interface. The commands have to be sent within a special package format. The following sections describe the format of the command set packages.

6.1.1 Framing

The connection is considered “Error free”. But for packet recognition and synchronization, some framing is used.
All packets sent in both directions are constructed after the following model:

Table 6-90. Package Framing

Start delimiter	Packet Type identification	Op code	Data length	Check-sum	Packet Data	End delimiter
1 byte	1 byte	1 byte	2 bytes	1 byte	<Data length> bytes	1 byte
	|---------- Checksum ----------|					

6.1.2 Start delimiter

The start delimiter indicates the LMX9830 the beginning of a new package. The “STX” char is used as start delimiter.

STX = 0x02

6.1.3 Packet type identification

This byte identifies the type of packet. The following types are valid:

Table 6-91. Packet Type Identification

Code	Packet Type	Description
0x52 'R'	Request (REQ)	A request sent to the Bluetooth module. All request are answered by exactly one confirm.
0x43 'C'	Confirm (CFM)	The Bluetooth modules confirm to a request. All request are answered by exactly one confirm.
0x69 'i'	Indication (IND)	Information sent from the Bluetooth module, that is not a direct confirm to a request.
0x72 'r'	Response (RES)	An optional response to an indication. This is used to respond to some type of indication messaged.

All other values are reserved.

6.1.4 Opcode

The opcode is a command specifier. Each command is represented by this one byte identifier.

Table 6-92. Opcode Values

Opcode	Value
GAP_INQUIRY	0x00
GAP_DEVICE_FOUND	0x01
GAP_REMOTE_DEVICE_NAME	0x02
GAP_READ_LOCAL_NAME	0x03
GAP_WRITE_LOCAL_NAME	0x04
GAP_READ_LOCAL_BDA	0x05
GAP_SET_SCANMODE	0x06
GAP_GET_FIXED_PIN	0x16

Table 6-92. Opcode Values

GAP_SET_FIXED_PIN	0x17
GAP_GET_PIN	0x75
GAP_GET_SECURITY_MODE	0x18
GAP_SET_SECURITY_MODE	0x19
GAP_REMOVE_PAIRING	0x1B
GAP_LIST_PAIRED_DEVICES	0x1C
GAP_ENTER_SNIFF_MODE	0x21
GAP_EXIT_SNIFF_MODE	0x37
GAP_ENTER_PARK_MODE	0x38
GAP_EXIT_PARK_MODE	0x39
GAP_ENTER_HOLD_MODE	0x3A
GAP_SET_LINK_POLICY	0x3B
GAP_GET_LINK_POLICY	0x3C
GAP_POWER_SAVE_MODE_CHANGED	0x3D
GAP_ACL_ESTABLISHED	0x50
GAP_ACL_TERMINATED	0x51
GAP_SET_AUDIO_CONFIG	0x59
GAP_GET_AUDIO_CONFIG	0x5A
GAP_ESTABLISH_SCO_LINK	0x5D
GAP_RELEASE_SCO_LINK	0x5E
GAP_MUTE_MIC	0x5F
GAP_SET_VOLUME	0x60
GAP_GET_VOLUME	0x61
GAP_CHANGE_SCO_PACKET_TYPE	0x62
SPP_SET_PORT_CONFIG	0x07
SPP_GET_PORT_CONFIG	0x08
SPP_PORT_CONFIG_CHANGED	0x09
SPP_ESTABLISH_LINK	0x0A
SPP_LINK_ESTABLISHED	0x0B
SPP_INCOMMING_LINK_ESTABLISHED	0x0C
SPP_RELEASE_LINK	0x0D
SPP_LINK_RELEASED	0x0E
SPP_SEND_DATA	0x0F
SPP_INCOMING_DATA	0x10
SPP_TRANSPARENT_MODE	0x11
SPP_CONNECT_DEFAULT_CON	0x12
SPP_STORE_DEFAULT_CON	0x13
SPP_GET_LIST_DEFAULT_CON	0x14
SPP_DELETE_DEFAULT_CON	0x15
SPP_SET_LINK_TIMEOUT	0x57
SPP_GET_LINK_TIMEOUT	0x58

Table 6-92. Opcode Values

SPP_PORT_STATUS_CHANGED	0x3E
SPP_GET_PORT_STATUS	0x40
SPP_PORT_SET_DTR	0x41
SPP_PORT_SET_RTS	0x42
SPP_PORT_BREAK	0x43
SPP_PORT_OVERRUN_ERROR	0x44
SPP_PORT_PARITY_ERROR	0x45
SPP_PORT_FRAMING_ERROR	0x46
SDAP_CONNECT	0x32
SDAP_DISCONNECT	0x33
SDAP_CONNECTION_LOST	0x34
SDAP_SERVICE_BROWSE	0x35
SDAP_SERVICE_SEARCH	0x36
SDAP_SERVICE_REQUEST	0x1E
SDAP_ATTRIBUTE_REQUEST	0x3F
CHANGE_LOCAL_BDADDRESS	0x27
CHANGE_NVS_UART_SPEED	0x23
CHANGE_UART_SETTINGS	0x48
SET_PORTS_TO_OPEN	0x22
GET_PORTS_TO_OPEN	0x1F
RESTORE_FACTORY_SETTINGS	0x1A
STORE_CLASS_OF_DEVICE	0x28
FORCE_MASTER_ROLE	0x1D
READ_OPERATION_MODE	0x49
WRITE_OPERATION_MODE	0x4A
SET_DEFAULT_LINK_POLICY	0x4C
GET_DEFAULT_LINK_POLICY	0x4D
SET_EVENT_FILTER	0x4E
GET_EVENT_FILTER	0x4F
SET_DEFAULT_LINK_TIMEOUT	0x55
GET_DEFAULT_LINK_TIMEOUT	0x56
SET_DEFAULT_AUDIO_CONFIG	0x5B
GET_DEFAULT_AUDIO_CONFIG	0x5C
SET_DEFAULT_LINK_LATENCY	0x63
GET_DEFAULT_LINK_LATENCY	0x64
SET_CLOCK_FREQUENCY	0x67
GET_CLOCK_FREQUENCY	0x68
SET_PCM_SLAVE_CONFIG	0x74
ENABLE_SDP_RECORD	0x29
DELETE_SDP_RECORDS	0x2A

Table 6-92. Opcode Values

STORE_SDP_RECORD	0x31
RESET	0x26
LMX9830_READY	0x25
TEST_MODE	0x24
WRITE_ROM_PATCH	0x47
READ_RSSI	0x20
RF_TEST_MODE	0x4B
DISABLE_TL	0x52
TL_ENABLED	0x53
HCI_COMMAND	0x65
AWAIT_INITIALIZATION_EVENT	0x66
ENTER_BLUETOOTH_MODE	0x66
SET_CLOCK_AND_BAUDRATE	0x69
SET_GPIO_WPU	0x6B
GET_GPIO_STATE	0x6C
SET_GPIO_DIRECTION	0x6D
SET_GPIO_OUTPUT_HIGH	0x6E
SET_GPIO_OUTPUT_LOW	0x6F
READ_NVS	0x72
WRITE_NVS	0x73

6.1.5 Data length

Number of bytes in the “Packet data” area. The maximum size is 333 bytes.

6.1.6 Packet data

The data fields hold binary data; hence both 0x02 (=STX) and 0x03 (=ETX) are allowed as data.

6.1.7 Checksum

This is a simple Block Check Character (BCC) checksum of the bytes from “Packet type” to, and including, “data length”. The BCC checksum is calculated as the low byte of the sum of all bytes.

E.g. if the sum of all bytes are 0x3724, the checksum is 0x24.

6.1.8 End delimiter

The “ETX” char is used as end delimiter.
ETX = 0x03

6.1.9 Retransmission

The connection is considered “Error free”, hence no need for implementing time-outs and retransmissions.

6.1.10 Flow control

A transparent data-mode is supported for RFCOMM communication. When using this transparent mode, full hardware handshake is needed.

When not in transparent mode, the protocol principle of REQ-CFM, limits the need of buffer capacity. As IND's can come out of REQ-CFM sequence, and is unconfirmed, the user device has to be able to read these data fast enough / have enough buffer capacity.

6.1.11 Byte Order

The byte order of the protocol is Little Endian, if nothing else is specified.

6.2 COMMAND SET

The LMX9830 implements a complete command set for bluetooth operation and local configuration.

The command set is based on a request/confirm scheme meaning any command will be confirmed by an appropriate event including the same opcode.

6.2.1 Searching for remote devices

The first step to establish a link to another device is to discover the devices in range. The discovering process in called “Inquiry”.

6.2.1.1 Inquiry

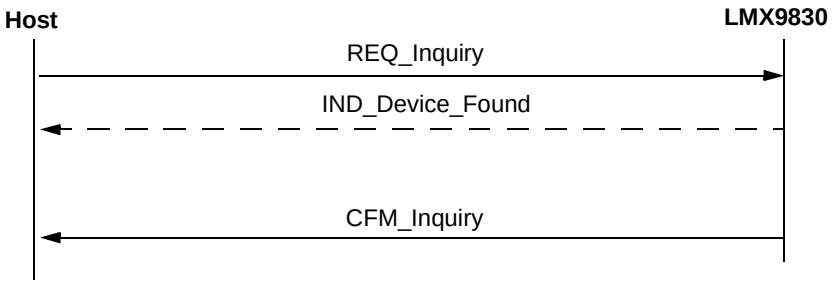


Figure 6-92. Inquiry Command Flow

Table 6-93. Inquiry Command

Description	**Initiates a search for other Bluetooth devices.**	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	GAP_INQUIRY	
DataLength	3	
Data	Length 1 byte	Duration of inquiry Range: 0x01 -0x30 (1.28s - 61.44s)
	NumResponses 1 byte	Maximum number of responses Range: 0x00 - 0xFF 0x00 = Unlimited number of responses.
	Mode 1 Byte	General Inquiry 0x00 Limited Inquiry 0x01

Table 6-94. Inquiry Confirm

Description	**Confirms that the search for other Bluetooth devices is complete.**
Firmware	0106, 02xx
PacketType	CFM
Opcode	GAP_INQUIRY
DataLength	1

Table 6-94. Inquiry Confirm

Data	Status 1 byte	ERROR_OK ERROR_DURATION_OUT_OF_RANGE ERROR_INVALID_MODE ERROR_INVALID_NO_OF_PARAMETERS

Table 6-95. Device Found Indicator

Description	**Indicates that a device has been found.**	
Firmware	0106, 02xx	
PacketType	IND	
Opcode	GAP_DEVICE_FOUND	
DataLength	9	
Data	BdAddr 6 bytes	Bluetooth device address of the found device.
	DeviceClass 3 byte	Class of the found device.

6.2.1.2 Get Friendly Name of the Remote Device

As seen the Device Found Indicator only delivers the BD_Addr and the Class of Device of Remote Devices. To get the friendly name of the device a separate command has to be used.

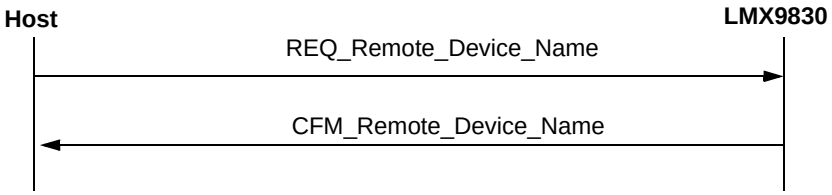


Figure 6-93. Get Remote Device Name Flow

Table 6-96. Get Remote Device Name Command

Description	**Request the user-friendly name from a known remote Bluetooth device.**	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	GAP_REMOTE_DEVICE_NAME	
DataLength	6	
Data	BdAddr 6 byte	Bluetooth device address for the remote device

Table 6-97. Get Remote Device Name Confirm

Description	**Confirm to the request above.**
Firmware	0106, 02xx
PacketType	CFM
Opcode	GAP_REMOTE_DEVICE_NAME
DataLength	8+ NameLength if ok, otherwise 8

Table 6-97. Get Remote Device Name Confirm

Data	Status 1 byte	ERROR_OK ERROR_INVALID_NO_OF_PARAMETERS ERROR_TIMEOUT
	BdAddr 6 byte	Bluetooth device address for the remote device
	NameLength 1 byte	Number of bytes in device name
	DeviceName Length bytes	The user-friendly name of the remote device. NULL terminated. Maximum length is 40 bytes.

6.2.2 SDAP Client Commands

Establishing a link to another device requires that devices BD_Addr but also the RFComm Port Number, the profile to connect to is registered at. If remote Com Port is unknown, it can be requested by a service discovery request using the SDAP client.

A SDAP request can only be done via an establish SDAP link. So the Command flow could be as following. To get information about the remote service, the host can use either “Service Browse”, “Service Search” or “Service Request” and “Attribute Request”. Usually the “Service Browse” is enough.

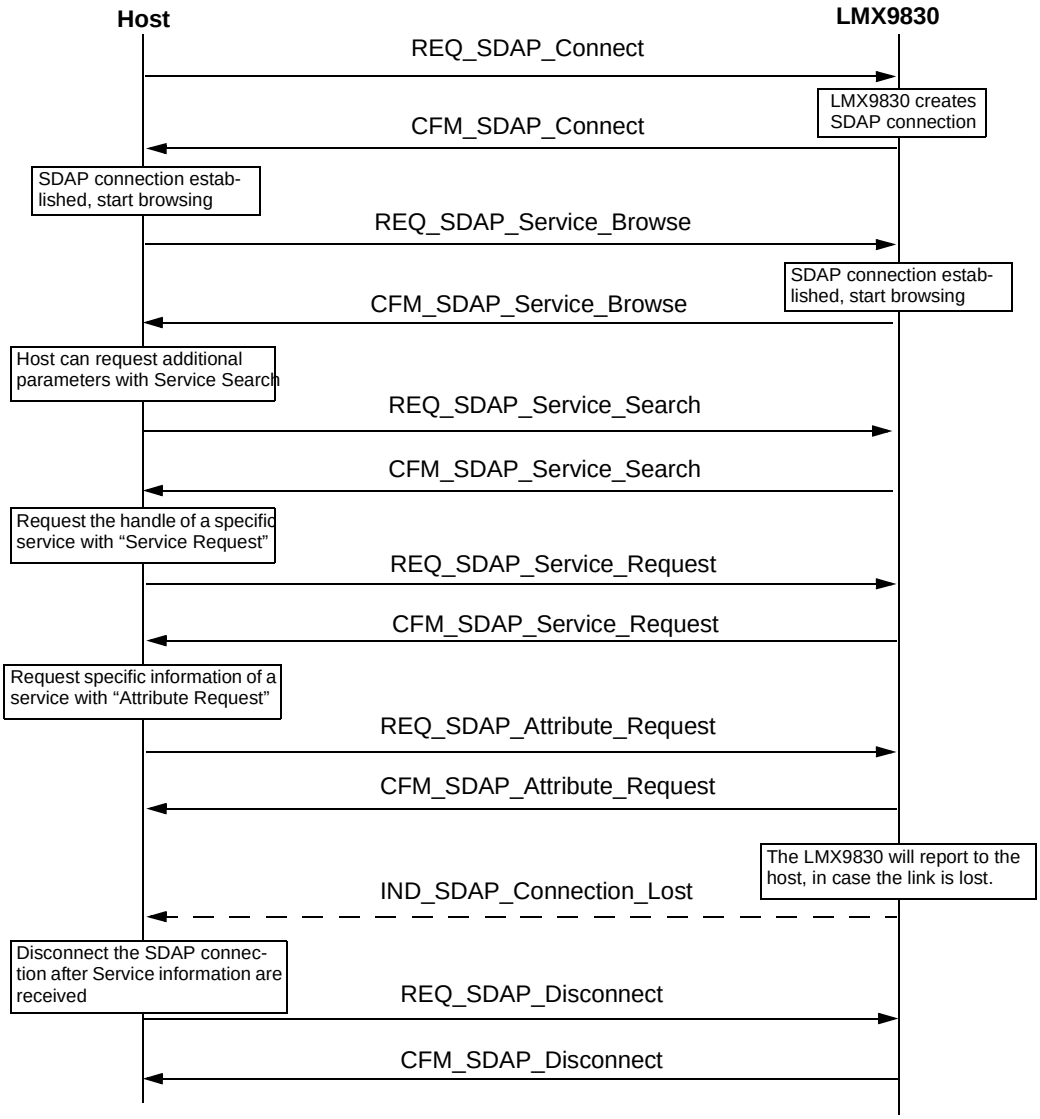


Figure 6-94. Requesting the services of a remote device

6.2.2.1 SDAP Connect

The SDAP Connect Request forces the LMX9830 to create a SDAP link to another device. This command is required for further SDAP Service Requests

Table 6-98. SDAP Connect Request

Description	Creates an SDP connection to a remote device. Only one SDP connection can be active.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SDAP_CONNECT	
DataLength	6	
Data	BdAddr 6 bytes	The Bluetooth address of the remote device to connect to.

Table 6-99. SDAP Connect Confirm

Description	Confirms the request above	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SDAP_CONNECT	
DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_INVALID_NO_OF_PARAMETERS ERROR_CONNECTION_FAILED

6.2.2.2 SDAP Disconnect

The SDAP link has to be disconnected after finishing the service browse/search.r

Table 6-100. Disconnect Request

Description	This command disconnects the active SDP connection.
Firmware	0106, 02xx
PacketType	REQ
Opcode	SDAP_DISCONNECT
DataLength	0

Table 6-101. Disconnect Confirm

Description	Confirms the request above	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SDAP_DISCONNECT	
DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_INVALID_NO_OF_PARAMETERS

6.2.2.3 SDAP Connection Lost Indicator

This indicator appears after an unexpected loss of the SDAP link.

Table 6-102. Connection Lost Indicator

Description	Notification sent to the application when a loss of the SDP connection is detected.
Firmware	0106, 02xx
PacketType	IND
Opcode	SDAP_CONNECTION_LOST
DataLength	0

6.2.2.4 SDAP Service Browse

The SDAP Service Browse can be used to get the RFComm Port Numbers of all or only specific Service Classes. The search mechanism is based on the 16bit-UUID for the services. The actual list of UUIDs can be found within the “Bluetooth Assigned Numbers” Document of the Bluetooth SIG.

Table 6-103. Example UUIDs for Service Classes

Service Class	UUID	Description
PublicBrowseGroup	0x1002	Returns the list of all registered services
SPP	0x1101	Serial Port Profile
DUN	0x1103	Dial-Up Networking Profile

Table 6-104. SDAP Service Browse Request

Description	This command is used to browse the service record of the remote device.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SDAP_SERVICE_BROWSE	
DataLength	2	
Data	BrowseGroupID 2 bytes	The requested browse group (16 bit UUID). The UUID has to be byte swapped within the command, e.g. to search for SPP entries the full command is 02 52 35 02 00 89 **01 11** 03

Table 6-105. SDAP Service Browse Confirm

Description	Confirms the request above	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SDAP_SERVICE_BROWSE	
DataLength	2+NoOfServices*(6 + NameLength)	
Data	Status 1 byte	ERROR_OK ERROR_INVALID_NO_OF_PARAMETERS ERROR_RESULT_TOO_LARGE ERROR_TRUNCATED_ANSWER
	NoOfServices 1 byte	Number of services found on remote device.
	For each service	
	BrowseGroupID 2 bytes	The browse group UUID that the service belongs to.

Table 6-105. SDAP Service Browse Confirm

		ServiceID 2 bytes	The service UUID.
		PortNr 1 byte	RFCOMM port number. The port which has to be used for link establishment to that service.
		NameLength 1 byte	The number of bytes in the service name
		ServiceName NameLength bytes	The name of the service.

6.2.2.5 SDAP Service Search

The SDAP Service Search command offers the ability to search for specific attributes for a service. The attribute IDs can be found at the Bluetooth SIG website under “Bluetooth Assigned numbers”.

Example: To request the “ProfileDescriptorList” of a SPP entry, the following command could be used:

02,52,36,06,00,8E,01,**11,01**,01,**00,04**,03

while:

- 11 01 reflects the UUID for the SPP service (0x1101) and
- 00 04 reflects the UUID for the “ProfileDescriptorList” (0x0004)

NOTE: As an exception to all other commands, this command does NOT required byte swapped parameters. So the UUIDs have to be filled in as defined in [4] "Bluetooth SIG: Bluetooth Assigned Numbers, https://www.bluetooth.org/foundry/assignnumb/document/assigned_numbers". See example above.

Table 6-106. SDAP Service Search Request

Description	**This command is used to search for services in the service record of the remote device.**	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SDAP_SERVICE_SEARCH	
DataLength	2 + 2*SearchPatternLength + 2*AttributesLength	
Data	SearchPatternLength 1 bytes	Number of 16-bit UUIDs in Search pattern List. **Note**: Must be less than 86 elements.
	SearchPattern	List of the requested services. The search pattern list is a list of 16-bit UUIDs of the requested services.
	AttributesLength 1 byte	Number of 16-bit UUID’s in attributeId list. **Note**: Must be less than 86 elements.
	Attributes	List of requested attributes for the requested services. The attribute list is a list of 16-bit UUID’s for the requested attributes.

Table 6-107. SDAP Service Search Confirm

Description	**Confirms the request above**
Firmware	0106, 02xx
PacketType	CFM
Opcode	SDAP_SERVICE_SEARCH
DataLength	3 + Length

Table 6-107. SDAP Service Search Confirm

Data	Status 1 byte	ERROR_OK ERROR_INVALID_NO_OF_PARAMETERS ERROR_UNKNOWN_ERROR ERROR_RESULT_TOO_LARGE
	Length 2 byte	Length of the result of the search. Maximum 330 bytes
	Result Length bytes	Result of the search

6.2.2.6 SDAP Service Request

Each Service Entry has a unique number called “Service Record Handle”. This command is used to get the record handle for stored entries for specific UUIDs.

Table 6-108. SDAP Service Request

Description	**This command is used the service record handles, from a remote device, for the given services in the search pattern.**	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SDAP_SERVICE_REQUEST	
DataLength	1 +2* SearchPatternLength	
Data	SearchPatternLength 1 bytes	Number of 16-bit UUID’s in Search pattern List. **Note**: Must be less than 86 elements.
	SearchPattern < 2* SearchPattern-Length > bytes	List of the requested services. The search pattern list is a list of 16-bit UUID’s of the requested services

Table 6-109. SDAP Service Request Confirm

Description	**Confirms the request above**	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SDAP_SERVICE_REQUEST	
DataLength	2 + 4*Length	
Data	Status 1 byte	ERROR_OK ERROR_INVALID_NO_OF_PARAMETERS ERROR_UNKNOWN_ERROR ERROR_RESULT_TOO_LARGE
	Length 1 byte	Number of 32-bit service record handles returned from remote device.
	Result <4*Length> bytes	The received 32-bit service record handles from the remote device.

6.2.2.7 SDAP Attribute Request

Instead of browsing the whole list of services within the remote database it is also possible to search only for specific attributes within a chosen entry. This command is based on the attribute “connection handle” of that specific entry.

Connection handles can be retrieved by the “SDAP Service Request” Command. (seeSection 6.2.2.6 on page 127)

Table 6-110. SDAP Attribute Request

Description	This command is used to get the given attributes for a given service record handle.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SDAP_ATTRIBUTE_REQUEST	
DataLength	5 + 2*AttributesLength	
Data	Handle 4 bytes	The 32-bit service record handle returned for a given service by SDAP_SERVICE_REQUEST.
	AttributesLength 1 byte	Number of 16-bit UUID’s in attributeId list. **Note**: Must be less than 86 elements.
	Attributes < 2 * AttributesLength> bytes	List of requested attributes for the requested services. The attribute list is a list of 16-bit UUID’s for the requested attributes.

Table 6-111. SDAP Attribute Request Confirm

Description	Confirms the request above	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SDAP_ATTRIBUTE_REQUEST	
DataLength	3 + Length	
Data	Status 1 byte	ERROR_OK ERROR_INVALID_NO_OF_PARAMETERS ERROR_UNKNOWN_ERROR ERROR_RESULT_TOO_LARGE
	Length 2 byte	Length of the result of the search. Maximum 330 bytes.
	Result	Result of the search

6.2.3 SPP Link Establishment

This section describes the basic functionality of creating a full SPP link to a remote device. Basically only one single command is needed to create the connection. The command “Establish Link” requires the BD_Addr and the RFComm port of the remote device, determined out of the Inquiry and the SDAP connection. The command and event flow can be found within the following flowchart.

The command will first be confirmed by a standard confirmation package. Afterwards the LMX9830 will start to page and try to connect to the remote device. The SPP_Link_Establishment indicator returns an error code reporting the success of the link establishment.

The flow also shows the procedure of sending data to the remote device using the “Send Data” Command. Incoming data are indicated by the “Incoming data event”.

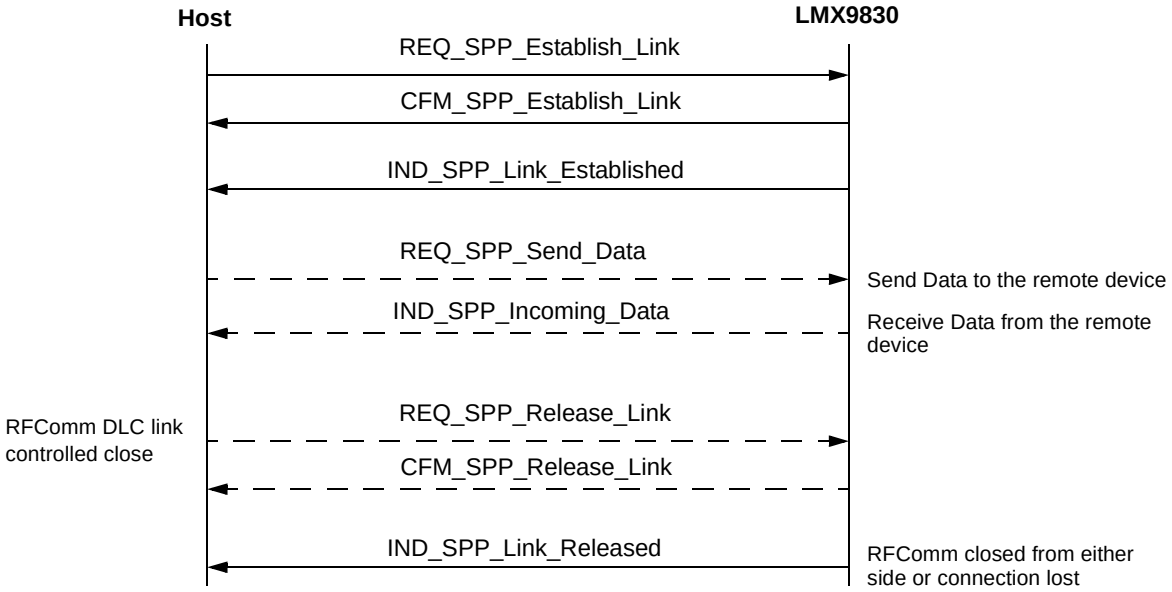


Figure 6-95. SPP Link Command Flow

6.2.3.1 Establish Link

The Establish Link command is the major command to establish a link to a remote device. To create a link the BD_Addr and the RFComm Channel on the remote device is required.

Table 6-112. Establish Link Request

Description	**Establish a DLC link to remote Bluetooth device**	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SPP_ESTABLISH_LINK	
DataLength	8	
Data	LocalPort 1 byte	Local RFCOMM port number. Range 1-30
	BdAddr 6 byte	Bluetooth device address for the remote device
	RemotePortNumber 1 byte	Remote device RFCOMM port number. (To be found using SDAP)

Table 6-113. Establish Link Confirm

Description	**Confirm that the DLC link establishment is initiated. Note: This confirm does NOT indicate link establishment, only that link establishment is in progress. When link establishment response is received from the core, a SPP_LINK_ESTABLISHED indication is sent from the LMX9830.**
Firmware	0106, 02xx
PacketType	CFM
Opcode	SPP_ESTABLISH_LINK
DataLength	2

Table 6-113. Establish Link Confirm

Data	Status 1 byte	ERROR_OK ERROR_SPP_PORT_BUSY ERROR_SPP_PORT_NOT_OPEN ERROR_SPP_INVALID_PORT ERROR_SPP_AUTOMATIC_CONNECTIONS_PROGRESSING
	LocalPort 1 byte	Local RFCOMM port number. Range 1-30

6.2.3.2 SPP Link Established Indicator

Table 6-114. SPP Link Established Indicator

Description	**Indication of establishment of a locally requested DLC link.**	
Firmware	0106, 02xx	
PacketType	IND	
Opcode	SPP_LINK_ESTABLISHED	
DataLength	9	
Data	Status 1 byte	Refer to “RFCOMM Error Codes” (page 200).
	BdAddr 6 byte	Bluetooth device address for the remote device.
	LocalPort 1 byte	Local RFCOMM port number. Range 1-30.
	RemotePortNumber 1 byte	Remote device RFCOMM port number.

6.2.3.3 SPP Send Data

If not switched to transparent, data have to be sent to a remote device using this command. The local RFComm Port is used to address the remote device.

Table 6-115. SPP Send Data

Description	**Send data on a SPP link to remote Bluetooth device**	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SPP_SEND_DATA	
DataLength	3 + <PayloadSize>	
Data	LocalPort 1 byte	Local RFCOMM port number. Range 1-30
	PayloadSize 2 bytes	Number of data bytes to send. Valid range is 1 to 330 bytes.
	PayloadData <PayloadSize> bytes	The data to send.

Description	**Confirm to the request above.**

Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SPP_SEND_DATA	
DataLength	2	
Data	Status 1 byte	ERROR_OK ERROR_LIMIT ERROR_UNABLE_TO_SEND ERROR_CURRENTLY_NO_BUFFER ERROR_NO_CONNECTION ERROR_SPP_INVALID_PORT ERROR_SPP_PORT_NOT_OPEN
	LocalPort 1 byte	Local RFCOMM port number. Range 1-30

6.2.3.4 Incoming Data Indicator

Table 6-116. SPP Incoming Data Indicator

Description	**Incoming data on a DLC link, from a remote Bluetooth device**	
Firmware	0106, 02xx	
PacketType	IND	
Opcode	SPP_INCOMING_DATA	
DataLength	3 + <PayloadSize>	
Data	LocalPort 1 byte	Local RFCOMM port number. Range 1-30
	PayloadSize 2 bytes	Number of data bytes to send. Valid range is 1 to 330 bytes.
	PayloadData <PayloadSize> bytes	The data to send.

6.2.3.5 SPP Release Link

Table 6-117. SPP Release Link Request

Description	**Release a DLC link to remote Bluetooth device**	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SPP_RELEASE_LINK	
DataLength	1	
Data	LocalPort 1 byte	Local RFCOMM port number. Range 1-30

Table 6-118. SPP Release Link Confirm

Description	**Confirm that the release is initiated. When the release is complete, a SPP_LINK_RELEASED indication is sent from the LMX9830.**
Firmware	0106, 02xx

Table 6-118. SPP Release Link Confirm

PacketType	CFM	
Opcode	SPP_RELEASE_LINK	
DataLength	2	
Data	Status 1 byte	ERROR_OK ERROR_NO_CONNECTION ERROR_SPP_INVALID_PORT ERROR_SPP_PORT_NOT_OPEN
	LocalPort 1 byte	Local RFCOMM port number. Range 1-30

6.2.3.6 SPP Link Released Indicator

Table 6-119. SPP Link Released Indicator

Description	**Indicates that a DLC link is released. The link release may have been initiated locally or remote, or could be caused by a loss of link (disturbance, dead device, ….)**	
Firmware	0106, 02xx	
PacketType	IND	
Opcode	SPP_LINK_RELEASED	
DataLength	2	
Data	Reason 1 byte	Refer to “RFCOMM Release Reasons” (page 201)
	LocalPort 1 byte	Local RFCOMM port number. Range 1-30

6.2.3.7 Incoming Link Established Indicator

In case a remote device creates a link to the LMX9830, the device will indicate the successful link establishment by sending by the “Incoming Link Established Indicator”. The packet includes the BD_Addr of the remote device and the local RFComm Port it connected to.

Table 6-120. Incoming Link Established Indicator

Description	**Indication of establishment of a remotely requested DLC link.**	
Firmware	0106, 02xx	
PacketType	IND	
Opcode	SPP_ INCOMING_LINK_ESTABLISHED	
DataLength	7	
Data	BdAddr 6 byte	Bluetooth device address for the remote device
	LocalPort 1 byte	Local RFCOMM port number. Range 1-30

6.2.3.8 Read RSSI

Table 6-121. Read RSSI Request

Description	This command will read out the current RSSI value for the existing link to a remote device.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	READ_RSSI	
DataLength	6	
Data	BdAddress 6 byte	Bluetooth Address of the remote device for which to read out the current RSSI value.

Table 6-122. Read RSSI Confirm

Description	Confirms the request above	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	READ_RSSI	
DataLength	2	
Data	Status 1 byte	ERROR_OK ERROR_UNKNOWN_ERROR ERROR_INVALID_NO_OF_PARAMETERS
	RSSI 1 byte	The RSSI value will be indicated in three steps. 0x01: Too High 0x00: Good 0xFF: Too Low

6.2.4 Transparent Mode

Transparent Mode offers the ability to switch off the Command Interface on the LMX9830 and use it as a cable replacement. This means data can be sent over a bluetooth link just by routing them to the LMX9830 without any package framing. Transparent mode can only be enabled if one SPP link is established. Transparent mode can not be used if a device has two or more active SPP links.

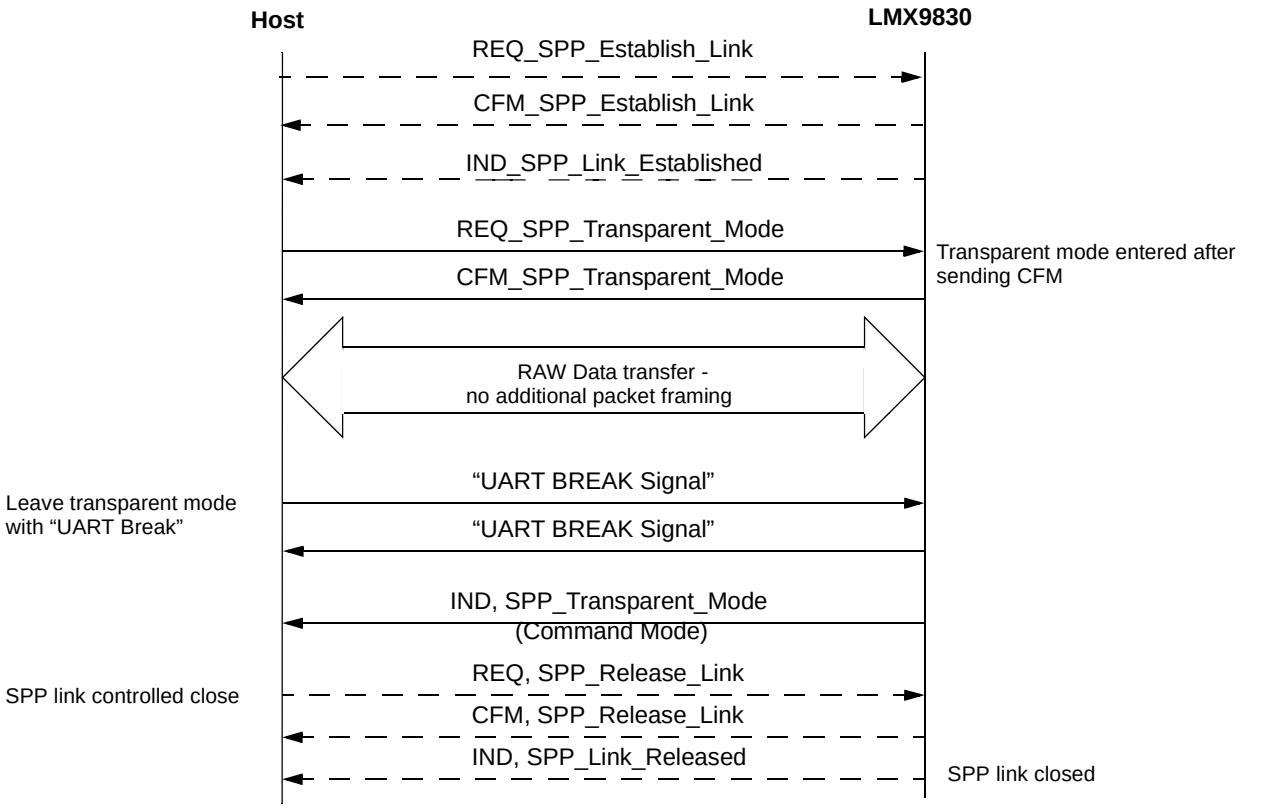


Figure 6-96. Transparent Mode; initiated by the Transparent Mode Request

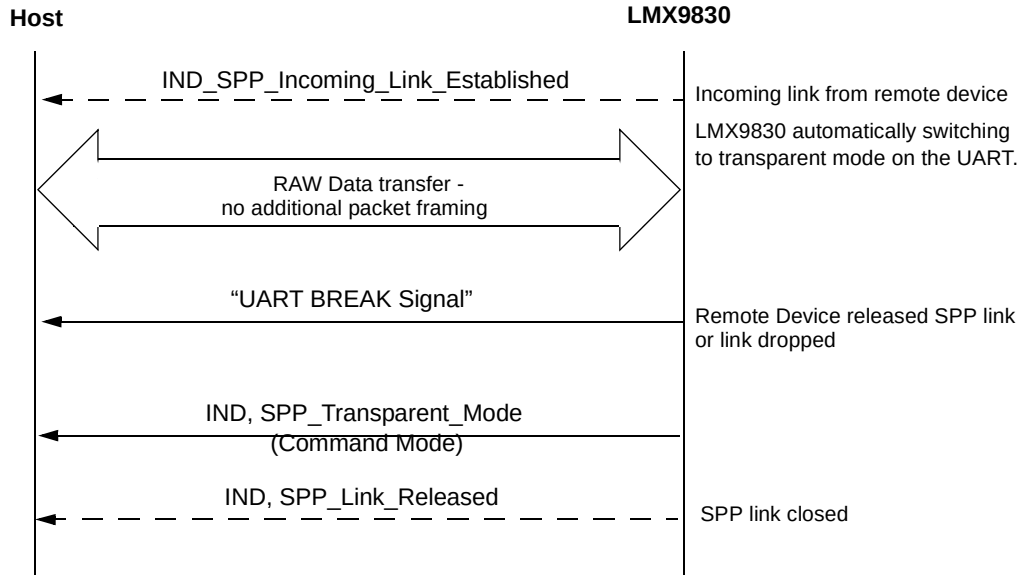


Figure 6-97. Transparent Mode; automatically activated in transparent slave

6.2.4.1 Transparent Mode

Table 6-123. Transparent Mode Request

Description	Switch to transparent mode on a SPP link to remote Bluetooth device	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SPP_TRANSPARENT_MODE	
DataLength	1	
Data	LocalPort 1 byte	Local RFCOMM port number. Range 1-30

Table 6-124. Transparent Mode Confirm

Description	Confirm to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SPP_TRANSPARENT_MODE	
DataLength	2	
Data	Status 1 byte	ERROR_OK ERROR_SPP_PORT_NOT_OPEN ERROR_SPP_INVALID_PORT ERROR_SPP_MULTIPLE_CONNECTIONS ERROR_NO_CONNECTION
	LocalPort 1 byte	Local RFCOMM port number. Range 1-30

6.2.4.2 Transparent Mode Left Indicator

Table 6-125. Transparent Mode Left Indicator

Description	Indication from Simply Blue that transparent mode is left.	
Firmware	0106, 02xx	
PacketType	IND	
Opcode	SPP_TRANSPARENT_MODE	
DataLength	2	
Data	LocalPort 1 byte	Local RFCOMM port number. Range 1-30
	Mode 1 byte	Command Mode = 0 Transparent Mode = 1

6.2.5 Link Timeout

The Link Timeout commands report and set the Supervision timeout setting for an established link. The link timeout which is set during link establishment on all incoming and outgoing links is read out of the NVS. The timeout can either be changed on an existing link or for each link by configuring the default setting in NVS.

See also Section 4.1.2.6 “Configuring the Default Link Timeout” on page 49 for a detailed description of the feature.

6.2.5.1 Set Default Link Timeout

Table 6-126. Set Default Link Timeout Request

Description	**This command is used to change the default link supervision timeout. The default link supervision timeout is set during connection setup. The default link supervision timeout setting is stored in NVS.**	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SET_DEFAULT_LINK_TIMEOUT	
DataLength	2	
Data	LinkTimeout 2 byte	The link supervision timeout in slots (0,625ms). The default value stored in the NVS after a factory reset is 20s. 0x0000: No link supervision timeout (the timer is disabled) 0x0190-0xFFFF: Valid timeout range (in slots) 0x7D00: The default value (20s)

Table 6-127. Set Default Link Timeout Confirm

Description	Response to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SET_DEFAULT_LINK_TIMEOUT	
DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_INVALID_NO_OF_PARAMETERS ERROR_ILLEGAL_LINK_TIMEOUT

6.2.5.2 Get Default Link Timeout

Table 6-128. Get Default Link Timeout Request

Description	**This command is used to read the default link supervision timeout setting from NVS.**
Firmware	0106, 02xx
PacketType	REQ
Opcode	GET_DEFAULT_LINK_TIMEOUT
DataLength	0

Table 6-129. Get Default Link Timeout Confirm

Description	**Response to the request above.**	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GET_DEFAULT_LINK_TIMEOUT	
DataLength	2	
Data	LinkTimeout 2 byte	The link supervision timeout in slots. 0x0000: No link supervision timeout (the timer is disabled) 0x0190-0xFFFF: Valid timeout range (in slots)

6.2.5.3 Set Link Timeout for an existing link

Table 6-130. Set Link Timeout request

Description	This command is used to change the current ACL link supervision timeout. This command will affect all physical links established to the specified device.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SPP_SET_LINK_TIMEOUT	
DataLength	8	
Data	BdAddr 6 bytes	The Bluetooth address of the remote device of the ACL link for which the Link Supervision Timeout is changed.
	LinkTimeout 2 byte	The link supervision timeout in slots (0.625ms). 0x0000: No link supervision timeout (the timer is disabled) 0x0190-0xFFFF: Valid timeout range (in slots) 0x7D00: The default value (20s)

Table 6-131. Set Link Timeout Confirm

Description	Response to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SPP_SET_LINK_TIMEOUT	
DataLength	7	
Data	Status 1 byte	ERROR_OK ERROR_UNSPECIFIED_ERROR ERROR_INVALID_NO_OF_PARAMETERS ERROR_NO_CONNECTION ERROR_ILLEGAL_LINK_TIMEOUT
	BdAddr 6 bytes	The Bluetooth address of the remote device of the ACL link for which the Link Supervision Timeout is changed.

6.2.5.4 Get Link Timeout of an existing link

Table 6-132. Get Link Timeout request

Description	This command is used to get the current link supervision timeout setting for the given ACL link.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SPP_GET_LINK_TIMEOUT	
DataLength	6	
Data	BdAddr 6 bytes	The Bluetooth address of the remote device of the ACL link for which the Link Supervision Timeout is requested.

Table 6-133. Get Link Timeout confirm

Description	Response to the request above.

Table 6-133. Get Link Timeout confirm

Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SPP_GET_LINK_TIMEOUT	
DataLength	9	
Data	Status 1 byte	ERROR_OK ERROR_UNSPECIFIED_ERROR ERROR_INVALID_NO_OF_PARAMETERS ERROR_NO_CONNECTION
	BdAddr 6 bytes	The Bluetooth address of the remote device of the ACL link for which the Link Supervision Timeout is requested.
	LinkTimeout 2 byte	The link supervision timeout in slots (0.625ms). 0x0000: No link supervision timeout (the timer is disabled) 0x0190-0xFFFF: Valid timeout range (in slots)

6.2.6 Audio (SCO) Link Support

6.2.6.1 Audio Path Configuration

The LMX9830 requires the configuration of the audio path in order to know, how the audio data need to be converted between the Advanced Audio Interface and the Bluetooth Link. The default audio settings allow to store this information in NVS, used for every SCO link.

6.2.6.1.1 Set Default Audio Settings

Table 6-134. Set Default Audio Settings Command

Description	**This command is used to set the default audio settings stored in NVS.**	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SET_DEFAULT_AUDIO_CONFIG	
DataLength	2	
Data	CodecType 1 byte	0x00: No codec available (default) 0x01: Motorola MC145483 0x02: OKI MSM7717 0x03: PCM slave, requires "PCM Slave Configuration" (6.2.6.1.3) 0x04-0xFF: Reserved.
	AirFormat 1 byte	0x00: CVSD (default) 0x01: µ-law 0x02: A-law 0x03-0xFF: Reserved.

6.2.6.1.2 Get Default Audio Settings

Table 6-135. Get Default Audio Settings Command

Description	**This command is used to get the default audio settings stored in NVS.**
Firmware	0106, 02xx
PacketType	REQ
Opcode	GET_DEFAULT_AUDIO_CONFIG

Table 6-135. Get Default Audio Settings Command

DataLength	0

Table 6-136. Get Default Audio Settings Confirm

Description	Response to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GET_DEFAULT_AUDIO_CONFIG	
DataLength	3	
Data	Status 1 byte	ERROR_OK ERROR_INVALID_NO_OF_PARAMETERS
	CodecType 1 byte	0x00: No codec available (default) 0x01: Motorola MC145483 0x02: OKI MSM7717 0x03: PCM slave, requires "PCM Slave Configuration" (6.2.6.1.3) 0x04-0xFF: Reserved.
	AirFormat 1 byte	0x00: CVSD (default) 0x01: µ-law 0x02: A-law 0x03-0xFF: Reserved.

6.2.6.1.3 PCM Slave Configuration

The PCM Slave configuration gets active in case the audio settings, configured by"Set Default Audio Settings" (6.2.6.1.1) or "Set audio settings" (6.2.6.2.5) has been set to “PCM slave”. For a detailed description of the settings, please refer to Section 4.5.3.1.2 "Using PCM slave" on page 73.

Table 6-137. Set PCM Slave Configuration Request

Description	This command is used to set the PCM slave configuration in the NVS.
PacketType	REQ
Opcode	SET_PCM_SLAVE_CONFIG
DataLength	3

Table 6-137. Set PCM Slave Configuration Request

Data	PcmSlaveConfig 2 bytes	This 16-bit value (LSB first) is used to store the PCM format configuration for the PCM generic slave. BIT0-1, Slot selection 00: use slot 0 01: use slot 1 10: use slot 2 11: use slot 3 BIT2-3: Number of slots per frame 00: 1 slot 01: 2 slots 10: 3 slots 11: 4 slots BIT4-6: PCM data format 000: Reserved 001: 8 bit A-law 010: 8 bit u-law 011: 13 bit linear 100: 14 bit linear 101: 15 bit linear 110: 16 bit linear 111: Reserved BIT7:Frame sync length 0: short frame sync 1: long frame sync BIT8: Data word length 0: 8-bit data word length 1: 16-bit data word length BIT9: Frame sync polarity 0: use inverted frame sync 1: use normal frame sync BIT10-15: Unused, set to 0
	Fcprs 1 byte	This value is an unsigned integer indicating the frame clock prescaler for generic PCM slave. The ratio between the bit clock and the frame clock must be written into the ACCR.FCPRS register for the generic PCM slave to operate correctly. The following equation must be true: bit_clock/(Fcprs + 1) = frame_clock. Example bit clock = 480000, frame sync rate = 8000, Fcprs must be set to 59 since 480000/(59 + 1) = 8000

Table 6-138. Set PCM Slave Configuration Confirm

Description	Response to the request above.	
PacketType	CFM	
Opcode	SET_PCM_SLAVE_CONFIG	
DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_INVALID_NO_OF_PARAMETERS ERROR_INVALID_CODEC_SETTING

6.2.6.2 Link Establishment

6.2.6.2.1 Establish SCO Link

Table 6-139. Establish SCO Link Command

Description	This command is used to establish a SCO link to a remote Bluetooth device. An ACL link must be established before a SCO link can be established.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	GAP_ESTABLISH_SCO_LINK	
DataLength	8	
Data	BdAddr 6 bytes	The Bluetooth address of the remote device to which the SCO link is established.
	PacketType 2 bytes	0x0020: HV1 0x0040: HV2 0x0080: HV3 The packet types can be combined / or'ed together in order to enable multiple packet types.

Table 6-140. Establish SCO Link Confirm

Description	Response to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_ESTABLISH_SCO_LINK	
DataLength	7	
Data	Status 1 byte	ERROR_OK ERROR_UNSPECIFIED_ERROR
	BdAddr 6 bytes	The Bluetooth address of the remote device.

6.2.6.2.2 SCO Link Established Indicator

Table 6-141. SCO Link Established Indicator

Description	Indicates that a SCO link is established. Either the local device or the remote device may have initiated the link establishment.	
Firmware	0106, 02xx	
PacketType	IND	
Opcode	GAP_ESTABLISH_SCO_LINK	
DataLength	7	
Data	Status 1 byte	ERROR_OK ERROR_UNSPECIFIED_ERROR ERROR_NO_CONNECTION ERROR_CONNECTION_FAILED
	BdAddr 6 bytes	The Bluetooth address of the remote.

6.2.6.2.3 Release SCO Link

Table 6-142. Release SCO Link Command

Description	**This command is used to disconnect a SCO link.**
Firmware	0106, 02xx
PacketType	REQ
Opcode	GAP_RELEASE_SCO_LINK
DataLength	0

Table 6-143. Release SCO Link Confirm

Description	**Response to the request above.**	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_RELEASE_SCO_LINK	
DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_NO_CONNECTION ERROR_INVALID_NO_OF_PARAMETERS

6.2.6.2.4 SCO Link Released Indicator

Table 6-144. SCO Link Released Indicator

Description	**Indicates that a SCO link is released. The link release may have been initiated by the local device, the remote device or it could be caused by a loss of link (disturbance, dead device, etc.)**	
Firmware	0106, 02xx	
PacketType	IND	
Opcode	GAP_RELEASE_SCO_LINK	
DataLength	2	
Data	Status 1 byte	ERROR_OK ERROR_UNKNOWN_ERROR ERROR_NO_CONNECTION
	HciReason 1 byte	The HCI release reason. The release reason indicated by the remote application.

6.2.6.2.5 Set audio settings

Table 6-145. Set audio settings command

Description	**This command is used to set the audio settings the device shall use for the existing or next SCO (audio) link established. This setting will not be stored to NVS. To store a default setting use "Set audio settings" (6.2.6.2.5)**
Firmware	0106, 02xx
PacketType	REQ
Opcode	GAP_SET_AUDIO_CONFIG
DataLength	2

Table 6-145. Set audio settings command

Data	CodecType 1 byte	0x00: No codec available 0x01: Motorola MC145483 0x02: OKI MSM7717 0x03: PCM Slave, requires "PCM Slave Configuration" (6.2.6.1.3) 0x0-0xFF: Reserved.
	AirFormat 1 byte	0x00: CVSD 0x01: µ-law 0x02: A-law 0x03-0xFF: Reserved.

Table 6-146. Set audio settings confirm

Description	**Response to the request above.**	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_SET_AUDIO_CONFIG	
DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_INVALID_NO_OF_PARAMETERS ERROR_ILLEGAL_AUDIO_CODEC_TYPE ERROR_ILLEGAL_AUDIO_AIR_FORMAT

6.2.6.2.6 Get audio settings

Table 6-147. Get audio settings command

Description	**This command is used to get the audio settings the device uses at an existing link or it shall use for the next SCO (audio) link established. This setting will not read the default settings stored in NVS. To read the default setting use "Get Default Audio Settings" (6.2.6.1.2)**
Firmware	0106, 02xx
PacketType	REQ
Opcode	GAP_GET_AUDIO_CONFIG
DataLength	0

Table 6-148. Get audio settings confirm

Description	**Response to the request above.**	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_GET_AUDIO_CONFIG	
DataLength	3	
Data	Status 1 byte	ERROR_OK ERROR_INVALID_NO_OF_PARAMETERS

Table 6-148. Get audio settings confirm

	CodecType 1 byte	0x00: No codec available 0x01: Motorola MC145483 0x02: OKI MSM7717 0x03: PCM Slave, requires "PCM Slave Configuration" (6.2.6.1.3) 0x04-0xFF: Reserved.
	AirFormat 1 byte	0x00: CVSD 0x01: µ-law 0x02: A-law 0x03-0xFF: Reserved.

6.2.6.2.7 Change SCO Packet Type

Table 6-149. Change SCO Packet Type Command

Description	**This command is used to change the packet type used/enabled for the SCO link.**	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	GAP_CHANGE_SCO_PACKET_TYPE	
DataLength	2	
Data	PacketType 2 bytes	0x0020: HV1 0x0040: HV2 0x0080: HV3 The packet types can be combined / or'ed together in order to enable multiple packet types.

Table 6-150. Change SCO Packet Type Confirm

Description	**Response to the request above.**	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_CHANGE_SCO_PACKET_TYPE	
DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_UNSPECIFIED_ERROR

6.2.6.2.8 Change SCO Link Packet Type Indicator

Table 6-151. Change SCO Link Packet Type Indicator

Description	**Indicates that the packet type of the SCO link has changed. Either the local device or the remote device may have initiated the change of the packet type of the SCO link.**
Firmware	0106, 02xx
PacketType	IND
Opcode	GAP_CHANGE_SCO_PACKET_TYPE
DataLength	3

Table 6-151. Change SCO Link Packet Type Indicator

Data	Status 1 byte	ERROR_OK ERROR_UNSPECIFIED_ERROR ERROR_NO_CONNECTION ERROR_CONNECTION_FAILED
	PacketType 2 bytes	0x0020: HV1 0x0040: HV2 0x0080: HV3 The packet types can be combined / or'ed together in order to enable multiple packet types.

6.2.6.2.9 Mute

Table 6-152. Mute Command

Description	**This command is used to mute the microphone.**	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	GAP_MUTE_MIC	
DataLength	1	
Data	Mute 1 byte	0x01: Mute 0x00: Un mute

Table 6-153. Mute Confirm

Description	**Response to the request above.**	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_SET_VOLUME	
DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_INVALID_NO_OF_PARAMETERS

6.2.6.2.10 Set Volume

Table 6-154. Set Volume Command

Description	**This command is used to set the speaker volume.**
Firmware	0106, 02xx
PacketType	REQ
Opcode	GAP_SET_VOLUME
DataLength	1

Table 6-154. Set Volume Command

Data	Volume 1 byte1	Volume control is only supported by the Motorola MC145483 codec. The following values are defined for MC145483: 0x00: Mute 0x01 - 0x1F: Step 1 (MIN) 0x20 - 0x3F: Step 2 0x40 - 0x5F: Step 3 0x60 - 0x7F: Step 4 0x80 - 0x9F: Step 5 0xA0 - 0xBF: Step 6 0xC0 - 0xDF: Step 7 0xE0 - 0xFF: Step 8 (MAX) The following values are defined for MSM7717: 0x00: Mute 0x01 - 0xFF: Max volume

Table 6-155. Set Volume Confirm

Description	Response to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_SET_VOLUME	
DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_INVALID_NO_OF_PARAMETERS ERROR_SET_VOLUME_FAILED

6.2.6.2.11 Get Volume

Table 6-156. Get Volume Command

Description	This command is used to get the speaker volume.
Firmware	0106, 02xx
PacketType	REQ
Opcode	GAP_GET_VOLUME
DataLength	0

Table 6-157. Get Volume Confirm

Description	Response to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_GET_VOLUME	
DataLength	2	
Data	Status 1 byte	ERROR_OK ERROR_INVALID_NO_OF_PARAMETERS

Table 6-157. Get Volume Confirm

	Volume 1 byte	The actual volume used.

6.2.7 Default Connections

A special feature for link setup is the storage of default connections. The LMX9830 can store up to three connections in its non-volatile data memory. If the “Operation Mode” parameter in NVS is set to automatic, the LMX9830 tries to connect to every device stored within the memory. After three attempts the device will give up and try the next one or stay in idle mode.

The connections can also be established during runtime by the “Establish default connection” command.

The connections are stored in a table like structure. Each connection is entered with an index number. This index is used to that device within the Establish or Delete commands. The transparent option can only be set if only 1 connection is stored.

Table 6-158. Example Default Connection Table

Index	Local Port	BD_Addr	Remote Port	Transparent
00	01	12 34 56 78 90 12	01	no
01	02	98 76 54 32 10 01	03	no

6.2.7.1 Store Default Connection

Table 6-159. Store Default Connection Confirm Request

Description	**Stores a default connection in NVS.**	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SPP_STORE_DEFAULT_CON	
DataLength	10	
Data	Index 1 byte	Index in the default connection storage. Range 0x00 to 0x06.
	LocalPort 1 byte	Local RFCOMM port number. Range 1-30
	RemotePort 1 byte	RFCOMM port number on remote device
	RemoteBdAddress 6 bytes	The BdAddress of the remote device
	TransparentMode 1 byte	Which transparent mode to enter when the connection is established. 0x00 Transparent mode off 0x01 Transparent mode on (only for point-to-point).

Table 6-160. Store Default Connection Confirm

Description	**Confirm to the request above.**
Firmware	0106, 02xx
PacketType	CFM
Opcode	SPP_STORE_DEFAULT_CON
DataLength	1

Table 6-160. Store Default Connection Confirm

Data	Status 1 byte	ERROR_OK ERROR_LIMIT ERROR_SPP_INVALID_PORT ERROR_SPP_MULTIPLE_TRANSPARENT ERROR_SPP_PORT_BUSY

6.2.7.2 Connect to Default Connection

Table 6-161. Connect to Default Connection Request

Description	**Connects the LMX9830 to a stored connection. Either a single or all connections can be established**	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SPP_CONNECT_DEFAULT_CON	
DataLength	1	
Data	Index 1 byte	Index in the default connection storage. Range 0x00 to 0x06. If index is set to 0xFF, all default connections will be established.

Table 6-162. Store Default Connection Confirm

Description	**Confirm to the request above.**	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SPP_ CONNECT _DEFAULT_CON	
DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_SPP_AUTOMATIC_CONNECTIONS_PROGRESSING ERROR_LIMIT ERROR_SPP_DEFAULT_CONNECTION_NOT_STORED

6.2.7.3 Get List of Default Connections

Table 6-163. Get List of Default Connections Request

Description	**Request a list of the default connections stored in NVS**	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SPP_GET_LIST_DEFAULT_CON	
DataLength	1	
Data	Index 1 byte	Index in the default connection storage. Range 0x00 to 0x06.

Table 6-164. Get List of Default Connections Confirm

Description	**Confirm to the request above.**
Firmware	0106, 02xx

Table 6-164. Get List of Default Connections Confirm

PacketType	CFM		
Opcode	SPP_GET_LIST_DEFAULT_CON		
DataLength	11		
Data	Status 1 byte	ERROR_OK ERROR_LIMIT	
	Index 1 byte	Index in the default connection storage. Range 0x00 to 0x06.	
	EntryStatus 1 byte	Bitfield: 0000000x	0: Entry is not stored. 1: Entry is stored.
		Bitfield: 000000x0	0: Command mode. 1: Transparent mode.
		Bitfield: xxxxxx00	Reserved.
	RemoteBdAddress 6 bytes	The BdAddress of the remote device	
	LocalPort 1 byte	Local RFCOMM port number. Range 1-30	
	RemotePort 1 byte	RFCOMM port number on remote device	

6.2.7.4 Delete Default Connection

Table 6-165. Delete Default Connections Request

Description	**Deletes a stored default connection in NVS**	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SPP_DELETE_DEFAULT_CON	
DataLength	1	
Data	Index 1 byte	Index in the default connection storage. Range 0x00 to 0x06.

Table 6-166. Delete Default Connections Confirm

Description	**Confirm to the request above.**	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SPP_DELETE_DEFAULT_CON	
DataLength	2	
Data	Index 1 byte	Index in the default connection storage. Range 0x00 to 0x06.
	Status 1 byte	0x00 OK

6.2.8 Bluetooth Low Power Modes

A bluetooth link is a based on a physically synchronized connection, which means that the devices can only communicate after successful synchronization. For this, each package also includes some synchronization information. Also a specific polling scheme is in place to keep synchronization if no traffic is necessary.

As the slave has to actively listen to packages from the master, there are different methods to decrease the necessary active receive slots on devices.

The ability to switch to those specific modes is controlled by the Link Policy. To make sure both devices support the low power mode requested, Link Policy can be set first. It will only be successful if both sides support it.

6.2.8.1 Set Default Link Policy

Table 6-167. Set Default Link Policy Command

Description	**This command is used to change the default link policy. The default link policy is set during connection setup. The default link policy setting is stored in NVS.**	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SET_DEFAULT_LINK_POLICY	
DataLength	2	
Data	Link Policy 2 byte	Bitfield: 0x0001 = Master-slave switch allowed 0x0002 = Hold mode allowed 0x0004 = Sniff mode allowed 0x0008 = Park mode allowed

Table 6-168. Set Default Link Policy Confirm

Description	**Response to the request above.**	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SET_DEFAULT_LINK_POLICY	
DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_INVALID_NO_OF_PARAMETERS ERROR_ILLEGAL_LINK_POLICY

6.2.8.2 Get Default Link Policy

Table 6-169. Get Default Link Policy Command

Description	**This command is used to read the default link policy setting from NVS.**
Firmware	0106, 02xx
PacketType	REQ
Opcode	GET_DEFAULT_LINK_POLICY
DataLength	0

Table 6-170. Get Default Link Policy Confirm

Description	**Response to the request above.**
Firmware	0106, 02xx
PacketType	CFM

Table 6-170. Get Default Link Policy Confirm

Opcode	GET_DEFAULT_LINK_POLICY	
DataLength	2	
Data	LinkPolicy 2 byte	Bitfield: 0x0001 = Master-slave switch allowed 0x0002 = Hold mode allowed 0x0004 = Sniff mode allowed 0x0008 = Park mode allowed

6.2.8.3 Set Link Policy

Table 6-171. Set Link Policy Request

Description	**This command is used to change the current link policy setting for the given link.**	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	GAP_SET_LINK_POLICY	
DataLength	8	
Data	BdAddr 6 bytes	The Bluetooth address of the remote device of which to change the link policy settings for the link.
	LinkPolicy 2 byte	Bitfield: 0x0001 = Master-slave switch allowed 0x0002 = Hold mode allowed 0x0004 = Sniff mode allowed 0x0008 = Park mode allowed

Table 6-172. Set Link Policy Confirm

Description	**Response to the request above.**	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_SET_LINK_POLICY	
DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_UNSPECIFIED_ERROR ERROR_INVALID_NO_OF_PARAMETERS ERROR_NO_CONNECTION ERROR_ILLEGAL_LINK_POLICY

6.2.8.4 Get Link Policy

Table 6-173. Get Link Policy Request

Description	**This command is used to get the current link policy setting for the given link.**
Firmware	0106, 02xx
PacketType	REQ
Opcode	GAP_GET_LINK_POLICY

Table 6-173. Get Link Policy Request

DataLength	6	
Data	BdAddr 6 bytes	The Bluetooth address of the remote device of which to get the current link policy settings for the link.

Table 6-174. Get Link Policy Confirm

Description	**Response to the request above.**	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_GET_LINK_POLICY	
DataLength	3	
Data	Status 1 byte	ERROR_OK ERROR_UNSPECIFIED_ERROR ERROR_INVALID_NO_OF_PARAMETERS ERROR_NO_CONNECTION
	LinkPolicy 2 byte	Bitfield: 0x0001 = Master-slave switch allowed 0x0002 = Hold mode allowed 0x0004 = Sniff mode allowed 0x0008 = Park mode allowed

6.2.8.5 Enter Sniff Mode

Command to enter the sniff mode. The command includes the maximum and minimum value for the sniff interval. After sending the command, Master and slave will calculate a reasonable sniff time and will switch into Sniff mode.

Table 6-175. Enter Sniff Mode Request

Description	**This command is used to request sniff mode on a given link with user specified parameters.**	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	GAP_ENTER_SNIFF_MODE	
DataLength	14	
Data	BdAddr 6 bytes	The Bluetooth address of the remote device of which to put the link in sniff mode.
	SniffMaxInterval 2 bytes	Maximum sniff interval in slots.
	SniffMinInterval 2 bytes	Minimum sniff interval in slots
	SniffAttempt 2 bytes	Number of slots the slave must listen, beginning at the sniff slot, even if it does not receive a packet with its own AM.
	SniffTimeout 2 bytes	Number of additional slots the slave must listen if it continues to receive only packets with its own AM address.

Table 6-176. Enter Sniff Mode Confirm

Description	**Response to the request above.**

Table 6-176. Enter Sniff Mode Confirm

Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_ENTER_SNIFF_MODE	
DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_UNSPECIFIED_ERROR ERROR_INVALID_NO_OF_PARAMETERS ERROR_NO_CONNECTION

6.2.8.6 Exit Sniff Mode

Table 6-177. Exit Sniff Mode Request

Description	**This command is used to exit a current sniff mode on a given link.**	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	GAP_EXIT_SNIFF_MODE	
DataLength	6	
Data	BdAddr 6 bytes	The Bluetooth address of the remote device of which to exit the current sniff mode.

Table 6-178. Exit Sniff Mode Confirm

Description	**Response to the request above.**	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_EXIT_SNIFF_MODE	
DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_UNSPECIFIED_ERROR ERROR_INVALID_NO_OF_PARAMETERS ERROR_NO_CONNECTION

6.2.8.7 Enter Park Mode

In Park Mode the slave will lose its active member address and will not longer be part of the piconet. It will be kept synchronized by beacons within the specified interval range.

Table 6-179. Enter Park Mode Request

Description	**This command is used to request park mode on a given link with user specified parameters.**
Firmware	0106, 02xx
PacketType	REQ
Opcode	GAP_ENTER_PARK_MODE
DataLength	10

Table 6-179. Enter Park Mode Request

Data	BdAddr 6 bytes	The Bluetooth address of the remote device of which to put the link in park mode.
	BeaconMaxInterval 2 bytes	Acceptable longest length of the interval between beacons.
	BeaconMinInterval 2 bytes	Acceptable shortest length of the interval between beacons.

Table 6-180. Enter Park Mode Confirm

Description	**Response to the request above.**	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_ENTER_PARK_MODE	
DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_UNSPECIFIED_ERROR ERROR_INVALID_NO_OF_PARAMETERS ERROR_NO_CONNECTION

6.2.8.8 Exit Park Mode

This commands forces the devices getting the parked slave back as active member of the piconet.

Table 6-181. Exit Park Mode Request

Description	**This command is used to exit a current park mode on a given link.**	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	GAP_EXIT_PARK_MODE	
DataLength	6	
Data	BdAddr 6 bytes	The Bluetooth address of the remote device of which to exit the current park mode.

Table 6-182. Enter Park Mode Confirm

Description	**Response to the request above.**	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_ENTER_PARK_MODE	
DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_UNSPECIFIED_ERROR ERROR_INVALID_NO_OF_PARAMETERS ERROR_NO_CONNECTION

6.2.8.9 Enter Hold Mode

Table 6-183. Enter Hold Mode Request

Description	This command is used to request Hold mode on a given link with user specified parameters.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	GAP_ENTER_HOLD_MODE	
DataLength	10	
Data	BdAddr 6 bytes	The Bluetooth address of the remote device of which to put the link in Hold mode.
	HoldMaxInterval 2 bytes	Maximum length of the Hold interval for which the Host may actually enter into the hold mode after negotiation with the remote device.
	HoldMinInterval 2 bytes	minimum length of the Hold interval for which the Host may actually enter into the hold mode after the negotiation with the remote device.

Table 6-184. Enter Hold Mode Confirm

Description	Response to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_ENTER_HOLD_MODE	
DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_UNSPECIFIED_ERROR ERROR_INVALID_NO_OF_PARAMETERS ERROR_NO_CONNECTION

6.2.8.10 Power Save Mode Changed

In case the remote device changed the Power Mode for that link, this event will be returned by the LMX9830.

Table 6-185. Power Save Mode Changed Indicator

Description	This indication is sent to the host when changes the power save mode on a link occur.	
Firmware	0106, 02xx	
PacketType	IND	
Opcode	GAP_POWER_SAVE_MODE_CHANGED	
DataLength	8	
Data	Status 1 byte	ERROR_OK ERROR_ATTEMPT_FAILED ERROR_UNSPECIFIED_ERROR
	BdAddr 6 bytes	The Bluetooth address of the remote device for which the power save mode has changed on the link.

Table 6-185. Power Save Mode Changed Indicator

	Mode 1 byte	0x00 = Active mode (Left power save mode) 0x01 = Hold mode (Hold mode entered) 0x02 = Sniff mode (Sniff mode entered) 0x03 = Park mode (Park mode entered)

6.2.9 Wake-Up Functionality

Wake-up functionality allows to stop the communication between host and LMX9830 in order to save power on both devices. The wake-up itself is done via the RTS/CTS pins or the Host_WU pin. For a detailed description of the hardware wake-up please see also Section 4.4.1.3 “Wake up functionality” on page 62.

The communication between host and LMX9830 can be stopped by sending the LMX9830 the Disable Transport Layer command. The command can be sent any time. The LMX9830 will continue its current operation status and try to wake up the host as soon as the interface is needed.

6.2.9.1 Disable Transport Layer

Table 6-186. Disable Transport Layer Command

Description	**This Command disables the transport layer and thereby allowing power-saving.**
Firmware	0106, 02xx
PacketType	REQ
Opcode	DISABLE_TL
DataLength	0

Table 6-187. Disable Transport Layer Confirm

Description	**Response to the request above.**	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	DISABLE_TL	
DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_COMMAND_DISALLOWED

6.2.9.2 Transport layer enabled indicator

Table 6-188. Transport Layer enabled indicator

Description	**This indication is sent when the transport layer is re-enabled. Transport layer is re-enabled by pulling RTS signal of the host or an event is ready to be sent from the Simply Blue to the host, normally due to an air-interface event.**	
Firmware	0106, 02xx	
PacketType	IND	
Opcode	TL_ENABLED	
DataLength	1	
Data	Status 1 byte	ERROR_OK

6.2.10 SPP Port Configuration

An active SPP link appears as a virtual serial port connection between two devices. As any other serial connection it has different settings for that “virtual” serial port. The following settings enable the host to change specific port settings on that virtual port. The reference for all commands is the local RFcomm port the link has been set up with.

All commands in this section are only handled at application level and have no direct impact on the bluetooth link.

6.2.10.1 SPP Set Port Configuration

Table 6-189. SPP Set Port Config Request

Description	**Write the configuration for the SPP port. These are “virtual” settings for the air connection, not the settings for the LMX9830 UART. Note: The baudrate in this configuration has no impact on the throughput on the bluetooth link itself or the LMX9830. It is not used by the RFCOMM layer.**			
Firmware	0106, 02xx			
PacketType	REQ			
Opcode	SPP_SET_PORT_CONFIG			
DataLength	6			
Data	LocalPort 1 byte	Local RFCOMM port number. Range 1-30		
	BaudRate 1 byte	Baudrate 0x00 = 2400 baud 0x01 = 4800 baud 0x02 = 7200 baud 0x03 = 9600 baud 0x04 = 19200 baud 0x05 = 38400 baud 0x06 = 57600 baud 0x07 = 115200 baud 0x08 = 230400 baud **Note: The baudrate in this configuration has no impact on the throughput on the bluetooth link itself or the LMX9830. It is not used by the RFCOMM layer. The parameter might be used as informative parameter on the application level.**		
	Portsettings 1 byte	Bitfield: 000000XX	Number of databits	00=5 bits 01=6 bits 02=7 bits 03=8 bits
		Bitfield: 00000X00	Number of stopbits	0 = 1 1 = 1.5
		Bitfield: 0000X000	Parity	0 = No parity 1 = Parity
		Bitfield: 00XX0000	ParityType	00=ODD 01=EVEN 02=MARK 03=SPACE
		Bitfield: XX000000	Reserved	
	FlowControl 1 byte	Bitfield: 0000000X	XonXoffOnInput	0=Disable 1=Enable
		Bitfield: 000000X0	XonXoffOnOutput	0=Disable 1=Enable
		Bitfield: 00000X00	RtrOnInput	0=Disable 1=Enable
		Bitfield: 0000X000	RtrOnOutput	0=Disable 1=Enable

Table 6-189. SPP Set Port Config Request

		Bitfield: 000X0000	RtcOnInput	0=Disable 1=Enable
		Bitfield: 00X00000	RtcOnOutput	0=Disable 1=Enable
		Bitfield: XX000000	Reserved	
	XonChar 1 byte	Char used for Xon, if Xon/Xoff flowcontrol is used.		
	XoffChar 1 byte	Char used for Xoff, if Xon/Xoff flowcontrol is used.		

Table 6-190. SPP Set Port Configuration Confirm

Description	**Confirm to the request above.**	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SPP_SET_PORT_CONFIG	
DataLength	2	
Data	Status 1 byte	ERROR_OK ERROR_UNSPECIFIED_ERROR ERROR_UNSPECIFIED_ERROR ERROR_SPP_INVALID_PORT ERROR_SPP_PORT_NOT_OPEN ERROR_UART_SPEED_OUT_OF_RANGE
	LocalPort 1 byte	Local RFCOMM port number. Range 1-30

6.2.10.2 SPP Get Port Configuration

Table 6-191. SPP Get Port Configuration Request

Description	**Read the configuration for the SPP port. This is “virtual” settings for the air connection, not settings for the LMX9830 serial port.**	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SPP_GET_PORT_CONFIG	
DataLength	1	
Data	LocalPort 1 byte	Local RFCOMM port number. Range 1-30

Table 6-192. SPP Get Port Configuration Confirm

Description	**Confirm to the request above.**
Firmware	0106, 02xx
PacketType	CFM
Opcode	SPP_GET_PORT_CONFIG
DataLength	7

Table 6-192. SPP Get Port Configuration Confirm

Data	Status 1 byte	ERROR_OK ERROR_UNSPECIFIED_ERROR ERROR_SPP_INVALID_PORT ERROR_SPP_PORT_NOT_OPEN		
	LocalPort 1 byte	Local RFCOMM port number. Range 1-30		
	BaudRate 1 byte	Baudrate 0x00 = 2400 baud 0x01 = 4800 baud 0x02 = 7200 baud 0x03 = 9600 baud 0x04 = 19200 baud 0x05 = 38400 baud 0x06 = 57600 baud 0x07 = 115200 baud 0x08 = 230400 baud		
	Portsettings 1 byte	Bitfield: 000000XX	Number of databits	00=5 bits 01=6 bits 02=7 bits 03=8 bits
		Bitfield: 00000X00	Number of stopbits	0 = 1 1 = 1.5
		Bitfield: 0000X000	Parity	0 = No parity 1 = Parity
		Bitfield: 00XX0000	ParityType	00=ODD 01=EVEN 02=MARK 03=SPACE
		Bitfield: XX000000	Reserved	
	FlowControl 1 byte	Bitfield: 0000000X	XonXoffOnInput	0=Disable 1=Enable
		Bitfield: 000000X0	XonXoffOnOutput	0=Disable 1=Enable
		Bitfield: 00000X00	RtrOnInput	0=Disable 1=Enable
		Bitfield: 0000X000	RtrOnOutput	0=Disable 1=Enable
		Bitfield: 000X0000	RtcOnInput	0=Disable 1=Enable
		Bitfield: 00X00000	RtcOnOutput	0=Disable 1=Enable
		Bitfield: XX000000	Reserved	
	XonChar 1 byte	Char used for Xon, if Xon/Xoff flowcontrol is used.		
	XoffChar 1 byte	Char used for Xoff, if Xon/Xoff flowcontrol is used.		

6.2.10.3 SPP Port Configuration Changed Indicator

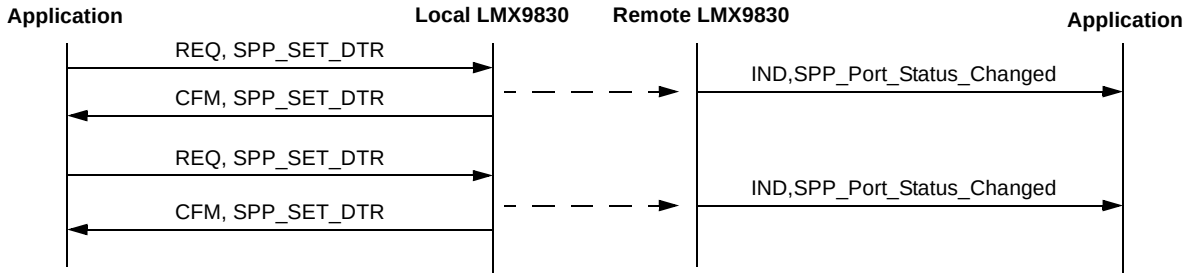
Table 6-193. SPP Port Configuration Changed Indicator

Description	**Sent from the LMX9830 when remote device has changed the port configuration.**			
Firmware	0106, 02xx			
PacketType	IND			
Opcode	SPP_PORT_CONFIG_CHANGED			
DataLength	6			
Data	LocalPort 1 byte	Local RFCOMM port number. Range 1-30		
	BaudRate 1 byte	Baudrate 0x00 = 2400 baud 0x01 = 4800 baud 0x02 = 7200 baud 0x03 = 9600 baud 0x04 = 19200 baud 0x05 = 38400 baud 0x06 = 57600 baud 0x07 = 115200 baud 0x08 = 230400 baud		
	Portsettings 1 byte	Bitfield: 000000XX	Number of databits	00=5 bits 01=6 bits 02=7 bits 03=8 bits
		Bitfield: 00000X00	Number of stopbits	0 = 1 1 = 1.5
		Bitfield: 0000X000	Parity	0 = No parity 1 = Parity
		Bitfield: 00XX0000	ParityType	00=ODD 01=EVEN 02=MARK 03=SPACE
		Bitfield: XX000000	Reserved	
	FlowControl 1 byte	Bitfield: 0000000X	XonXoffOnInput	0=Disable 1=Enable
		Bitfield: 000000X0	XonXoffOnOutput	0=Disable 1=Enable
		Bitfield: 00000X00	RtrOnInput	0=Disable 1=Enable
		Bitfield: 0000X000	RtrOnOutput	0=Disable 1=Enable
		Bitfield: 000X0000	RtcOnInput	0=Disable 1=Enable
		Bitfield: 00X00000	RtcOnOutput	0=Disable 1=Enable
		Bitfield: XX000000	Reserved	
	XonChar 1 byte	Char used for Xon, if Xon/Xoff flowcontrol is used.		
	XoffChar 1 byte	Char used for Xoff, if Xon/Xoff flowcontrol is used.		

6.2.11 SPP Port Status

An active SPP link allows signalling of modem status and line status over the bluetooth link. The following commands and events describe how to change or get the status of those line parameters.

NOTE: This handshaking events have no impact on the LMX9830 flow control. These messages can only used by the host application on each side to either react on the flow themselves.



6.2.11.1 SPP Get Port Status

Description	**Get the current state of the modem status and line status at application level. This command resets the value (to 0) of the following members of the port status:** **OverrunError** **ParityError** **FramingError** **BreakLength** **The value of DSR and CTS are only changed when new values are received from the remote device!**	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SPP_GET_PORT_STATUS	
DataLength	1	
Data	LocalPort 1 byte	Local RFCOMM port number. Range 1-30

Table 6-194. SPP Get Port Status Confirm

Description	**Confirm to the request above.**			
Firmware	0106, 02xx			
PacketType	CFM			
Opcode	SPP_GET_PORT_STATUS			
DataLength	5			
Data	Status 1 byte	ERROR_OK ERROR_UNSPECIFIED_ERROR ERROR_SPP_INVALID_PORT ERROR_SPP_PORT_NOT_OPEN		
	LocalPort 1 byte	Local RFCOMM port number. Range 1-30		
	PortStatus 1 byte	Bitfield: 0000000X	DTR	0 = Low 1 = High

Table 6-194. SPP Get Port Status Confirm

		Bitfield: 000000X0	RTS	0 = Low 1 = High
		Bitfield: 00000X00	DSR	0 = Low 1 = High
		Bitfield: 0000X000	CTS	0 = Low 1 = High
		Bitfield: 000X0000	Overrun Error	0 = No Error 1 = Overrun Error
		Bitfield: 00X00000	Parity Error	0 = No Error 1 = Parity Error
		Bitfield: 0X000000	Framing Error	0 = No Error 1 = Framing Error
		Bitfield:X0000000	DLC established	0 = No DLC 1 = DLC is available
	Break Length 2 bytes	The length in ms of the detected break. The value 0 is used to indicate that no break has been detected.		

6.2.11.2 SPP Port Set DTR

Table 6-195. SPP Port Set DTR Request

Description	**This command sets the state of the DTR. Since RFCOMM acts as a "null modem" where DTR and DSR are connected, the remote device will see this as a change of the state of the DSR signal. In case the remote device is a LMX9830, this remote LMX9830 will send a "SPP Port Status Changed Indicator" (6.2.11.8) to his application.**	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SPP_PORT_SET_DTR	
DataLength	2	
Data	LocalPort 1 byte	Local RFCOMM port number. Range 1-30
	State 1 byte	0: False. 1: True.

Table 6-196. SPP Port Set DTR Confirm

Description	**Confirm to the request above.**
Firmware	0106, 02xx
PacketType	CFM
Opcode	SPP_PORT_SET_DTR
DataLength	2

Table 6-196. SPP Port Set DTR Confirm

Data	Status 1 byte	ERROR_OK ERROR_ILLEGAL_STATE_VALUE ERROR_SPP_INVALID_PORT ERROR_SPP_PORT_NOT_OPEN
	LocalPort 1 byte	Local RFCOMM port number. Range 1-30

6.2.11.3 SPP Port Set RTS

Table 6-197. SPP Port Set RTS Request

Description	**This command sets the state of the RTS bit. Since RFCOMM acts as a “null modem” where RTS and CTS are connected, the remote device will see this as a change of the state of the CTS signal. In case the remote device is a LMX9830, this remote LMX9830 will send a "SPP Port Status Changed Indicator" (6.2.11.8) to his application.**	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SPP_PORT_SET_RTS	
DataLength	2	
Data	LocalPort 1 byte	Local RFCOMM port number. Range 1-30
	State 1 byte	0: False. 1: True.

Table 6-198. SPP Port Set RTS Confirm

Description	**Confirm to the request above.**	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SPP_PORT_SET_RTS	
DataLength	2	
Data	Status 1 byte	ERROR_OK ERROR_ILLEGAL_STATE_VALUE ERROR_SPP_INVALID_PORT ERROR_SPP_PORT_NOT_OPEN
	LocalPort 1 byte	Local RFCOMM port number. Range 1-30

6.2.11.4 SPP Port Set BREAK

Table 6-199. SPP Set Port Break Request

Description	**This command indicates a UART BREAK over the virtual serial link. In case the remote device is a LMX9830, this remote LMX9830 will send a "SPP Port Status Changed Indicator" (6.2.11.8) to his application.**
Firmware	0106, 02xx

Table 6-199. SPP Set Port Break Request

PacketType	REQ	
Opcode	SPP_PORT_BREAK	
DataLength	3	
Data	LocalPort 1 byte	Local RFCOMM port number. Range 1-30
	BreakLength 2 bytes	The length of the break in ms.

Table 6-200. SPP Set Port Break Confirm

Description	**Confirm to the request above.**	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SPP_PORT_BREAK	
DataLength	2	
Data	Status 1 byte	ERROR_OK ERROR_SPP_INVALID_PORT ERROR_SPP_PORT_NOT_OPEN
	LocalPort 1 byte	Local RFCOMM port number. Range 1-30

6.2.11.5 Set Overrun Error

Table 6-201. SPP Set Overrun Error Request

Description	**This command is used to indicate that the host has detected an overrun error. In case the remote device is a LMX9830, this remote LMX9830 will send a "SPP Port Status Changed Indicator" (6.2.11.8) to his application.**	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SPP_PORT_OVERRUN_ERROR	
DataLength	1	
Data	LocalPort 1 byte	Local RFCOMM port number. Range 1-30

Table 6-202. SPP Port Overrun Error Confirm

Description	**Confirm to the request above.**	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SPP_PORT_OVERRUN_ERROR	
DataLength	2	
Data	Status 1 byte	ERROR_OK ERROR_SPP_INVALID_PORT ERROR_SPP_PORT_NOT_OPEN

Table 6-202. SPP Port Overrun Error Confirm

	LocalPort 1 byte	Local RFCOMM port number. Range 1-30

6.2.11.6 SPP Set Parity Error

Table 6-203. SPP Port Parity Error Request

Description	**This command is used to indicate that the host has detected a parity error. In case the remote device is a LMX9830, this remote LMX9830 will send a "SPP Port Status Changed Indicator" (6.2.11.8) to his application.**	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SPP_PORT_PARITY_ERROR	
DataLength	1	
Data	LocalPort 1 byte	Local RFCOMM port number. Range 1-30

Table 6-204. SPP Port Parity Error Confirm

Description	**Confirm to the request above.**	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SPP_PORT_PARITY_ERROR	
DataLength	2	
Data	Status 1 byte	ERROR_OK ERROR_SPP_INVALID_PORT ERROR_SPP_PORT_NOT_OPEN
	LocalPort 1 byte	Local RFCOMM port number. Range 1-30

6.2.11.7 SPP Set Framing Error

Table 6-205. SPP Port Framing Error Request

Description	**This command is used to indicate that the host has detected a framing error. In case the remote device is a LMX9830, this remote LMX9830 will send a "SPP Port Status Changed Indicator" (6.2.11.8) to his application.**	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SPP_PORT_FRAMING_ERROR	
DataLength	1	
Data	LocalPort 1 byte	Local RFCOMM port number. Range 1-30

Table 6-206. SPP Port Framing Errors Confirm

Description	**Confirm to the request above.**
Firmware	0106, 02xx

Table 6-206. SPP Port Framing Errors Confirm

PacketType	CFM	
Opcode	SPP_PORT_FRAMING_ERROR	
DataLength	2	
Data	Status 1 byte	ERROR_OK ERROR_SPP_INVALID_PORT ERROR_SPP_PORT_NOT_OPEN
	LocalPort 1 byte	Local RFCOMM port number. Range 1-30

6.2.11.8 SPP Port Status Changed Indicator

Table 6-207. SPP Port Status Changed Indicator

Description	**Send from the LMX9830 when remote device has changed the port.**		
Firmware	0106, 02xx		
PacketType	IND		
Opcode	SPP_PORT_STATUS_CHANGED		
DataLength	4		
Data	LocalPort 1 byte	Local RFCOMM port number. Range 1-30	
	PortStatus 1 byte	Bitfield: 0000000X	DTR 0 = Low 1 = High
		Bitfield: 000000X0	RTS 0 = Low 1 = High
		Bitfield: 00000X00	DSR 0 = Low 1 = High
		Bitfield: 0000X000	CTS 0 = Low 1 = High
		Bitfield: 000X0000	Overrun Error 0 = No Error 1 = Overrun Error
		Bitfield: 00X00000	Parity Error 0 = No Error 1 = Parity Error
		Bitfield: 0X000000	Framing Error 0 = No Error 1 = Framing Error
		Bitfield:X0000000	DLC established 0 = No DLC 1 = DLC is available
	Break Length 2 bytes	The length in ms of the detected break. The value 0 is used to indicate that no break has been detected.	

6.2.12 RFcomm Channels to open

Each Service within the Service Database is registered to a specific RFComm channel. The configuration, which ports will be opened and initialized can be configured with the command "Set Ports To Open" (6.2.12.1).

If a RFComm port has not been opened, it is not connectable from outside or can not be used for an outgoing link.

6.2.12.1 Set Ports To Open

Table 6-208. SDP Set Ports to Open Request

Description	This command will change which RFCOMM ports the LMX9830 initializes. The value gets effective immediately and is stored within the NVS to be restored after reset.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SET_PORTS_TO_OPEN	
DataLength	4	
Data	PORTS 4 Bytes	This field is a 32-bit mask indicating which RFCOMM ports the LMX9830 has to open. Bit 30 and 31 must be set to 0. Bit 0 is RFCOMM port 1 and bit 29 is port 30 e.g. if this field has the value 0x00000007, port 1 to 3 will be opened. All other ports will be closed if open.

Table 6-209. SDP Set Ports to Open Confirm

Description	Confirms the request above	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	PORTS_TO_OPEN	
DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_INVALID_PORT ERROR_INVALID_NO_OF_PARAMETERS

6.2.12.2 Get Ports To Open

Table 6-210. SDP Get Ports to open Request

Description	This command reads back the currently opened RFComm ports.
Firmware	0106, 02xx
PacketType	REQ
Opcode	GET_PORTS_TO_OPEN
DataLength	0

Table 6-211. SDP Get Ports to open Confirm

Description	Confirms the request above	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	PORTS_TO_OPEN	
DataLength	5	
Data	Status 1 byte	ERROR_OK ERROR_INVALID_NO_OF_PARAMETERS
	PORTS 4 Bytes	This field is a 32-bit mask indicating which RFCOMM ports the LMX9830 has to open. Bit 30 and 31 must be set to 0. Bit 0 is RFCOMM port 1 and bit 29 is port 30 e.g. if this field has the value 0x00000007, port 1 to 3 will be opened.

6.2.13 Local Service Database Configuration

The LMX9830 allows the modification of the Local Service Discovery Database. By default, the service database contains one entry configured at RFComm port 1 for a Serial Port Profile, with Authentication and Encryption enabled.

If the application needs to offer a second service to another device, this service has to be entered within the service database and, if required, the RFComm port has to be opened (see "Set Ports To Open" (6.2.12.1)). The service entry in general includes information about the name of the service, which appears on a remote device after browsing, the port number and security settings.

Each registered service entry itself can be enabled or disabled. This allows to have different services registered to one specific RFComm port enabling and disabling them by needs.

The command set includes just one command to store any SDP record available within the Service Database, called "Store Generic SDP Record" (6.2.13.1). Each record includes the profile specific settings needed.

The service records are stored in a database like system within the NVS, so they are still available after reset. The storage of an entry is confirmed by the LMX9830 with a specific record identifier. This identifier is needed for the Enabling or Disabling command to address those specific entries.

NOTE: The service database is reset to the default factory setting, in case the LMX9830 is used without EEPROM. In this case all entries have to be initiated after the "Await Initialization Event" (6.2.20.1) or "LMX9830 Ready" (6.2.18.4) indicator.

6.2.13.1 Store Generic SDP Record

Table 6-212. Store Generic SDP Record Request

Description	**This command will create a new service record in the local SDP database, store in the NVS. The "Simply Blue Commander" offers the ability to generate service records with all correct parameters automatically. See also "Configuring the Service Database for Headset" (5.4.2.2) for an example.**	
PacketType	REQ	
Opcode	STORE_SDP_RECORD	
DataLength	5 + <SdpRecordLength>	
Data	LocalPort 1 byte	The local RFCOMM port used by this service. Range 1 to 30.
	Authentication 1 byte	0x00 No authentication requirements. 0x02 Authentication is only required for this profile for incoming connections. 0x20 Authentication is only required for this profile for outgoing connections. 0x22 Authentication is required for this profile for connections in both directions
	Encryption 1 byte	0x00 No encryption requirements 0x04 Encryption is only required for this profile for incoming connections. 0x40 Encryption is only required for this profile for outgoing connections. 0x44 Encryption is required for this profile for connections in both directions
	SdpRecordLength 2 bytes	The number of bytes in the SDP record.
	SdpRecord <SdpRecord-Length> Bytes	The SDP record data formatted as: < uint16:AttributeId, SdpDataElement:AttributeData, uint16:AttributeId, SdpDataElement:AttributeData, …>. The maximum length for each service is 255 bytes.

Table 6-213. Store Generic SDP Record Confirm

Description	**Confirms the request above**

Table 6-213. Store Generic SDP Record Confirm

PacketType	CFM	
Opcode	STORE_SDP_RECORD	
DataLength	2	
Data	Status 1 byte	ERROR_OK ERROR_INVALID_SDP_RECORD ERROR_MAXIMUM_NO_OF_SERVICE_RECORDS_REACHED ERROR_WRITING_TO_NVS
	Identifier 1 byte	A unique identifier, which must be used when dynamic enabling/disabling the record.

6.2.13.2 SDP Enable SDP Record

Table 6-214. SDP Enable SDP Record Request

Description	**This command is used to enable/disable stored SDP records in the LMX9830.**	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	ENABLE_SDP_RECORD	
DataLength	2	
Data	State 1 byte	The new state of the SDP record. 0x00 Disable the record. 0x01 Enable the record.
	Identifier 1 byte	The identifier of the service record to address. This will be received when the record was stored in the LMX9830 by "Store Generic SDP Record" (6.2.13.1).

Table 6-215. SDP Enable SDP Record Confirm

Description	**Confirms the request above**	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	ENABLE_SDP_RECORD	
DataLength	2	
Data	Status 1 byte	ERROR_OK ERROR_ILLEGAL_STATE_VALUE ERROR_IDENTIFIER_OUT_OF_RANGE ERROR_RECORD_ALREADY_IN_SELECTED_STATE ERROR_IDENTIFIER_NOT_IN_USE ERROR_INVALID_NO_OF_PARAMETERS
	Identifier 1 byte	The identifier received when the record was stored in the LMX9830.

6.2.13.3 SDP Delete All SDP Records

Table 6-216. SDP Delete SDP Record Request

Description	This command is used to delete all stored SDP records in the LMX9830.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	DELETE_SDP_RECORDS	
DataLength	0	
Data	None	

Table 6-217. SDP Delete SDP Record Confirm

Description	Confirms the request above	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	DELETE_SDP_RECORDS	
DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_INVALID_NO_OF_PARAMETERS

6.2.14 Local Bluetooth Settings

The LMX9830 uses the NVS Data memory to store all parameters specific for the local device. All bluetooth settings are checked during runtime and read out directly from this memory area.

6.2.14.1 Read Local Name

Table 6-218. Read Local Name Request

Description	Request the user-friendly name for the local Bluetooth device.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	GAP_READ_LOCAL_NAME	
DataLength	0	
Data	None	

Table 6-219. Read Local Name Confirm

Description	Confirm to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_READ_LOCAL_NAME	
DataLength	2 + NameLength	
Data	Status 1 byte	ERROR_OK ERROR_INVALID_NO_OF_PARAMETERS ERROR_UNKNOWN_ERROR
	NameLength 1 byte	Number of bytes in device name

Table 6-219. Read Local Name Confirm

	DeviceName NameLength bytes	The user-friendly name of the local device. The string is NULL terminated. Max length is 40 bytes.

6.2.14.2 Write Local Name

Table 6-220. Write Local Name Request

Description	Change the user-friendly name for the local Bluetooth device. The name is stored in NVS	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	GAP_WRITE_LOCAL_NAME	
DataLength	1+ NameLength	
Data	NameLength 1 byte	Number of bytes in device name
	DeviceName Length bytes	The user-friendly name of the local device. (String must be NULL terminated). Max length is 40 bytes.

Table 6-221. Write Local Name Confirm

Description	Confirm to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_WRITE_LOCAL_NAME	
DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_NAME_TOO_LONG ERROR_INVALID_NO_OF_PARAMETERS

6.2.14.3 Read Local Bluetooth Address

Table 6-222. Read Local BD_Addr Request

Description	Read the Bluetooth device address of the local Bluetooth device.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	GAP_READ_LOCAL_BDA	
DataLength	0	
Data	None	

Table 6-223. Read Local BD_Addr Confirm

Description	Confirm to the request above.
Firmware	0106, 02xx
PacketType	CFM
Opcode	GAP_READ_LOCAL_BDA
DataLength	7

Table 6-223. Read Local BD_Addr Confirm

Data	Status 1 byte	ERROR_OK ERROR_INVALID_NO_OF_PARAMETERS ERROR_UNKNOWN_ERROR
	BdAddr 6 byte	Bluetooth device address for the local device

6.2.14.4 Change Local Bluetooth Address

The Bluetooth Device Address (BD_Addr) is a key parameter for bluetooth operation. Therefore, the LMX9830 will only allow to switch to bluetooth mode ("Enter Bluetooth Mode" (6.2.20.2)), if this value is different to 0xFFFFFFFFFFFF. Please see also "Initialization Mode" (2.1) for details on the BD_Addr.

Usually the BD_Addr is stored in the external EEPROM. If no EEPROM is connected, this parameter has to be pushed to the LMX9830 after each power-up.

Table 6-224. Change Local BD_Address Request

Description	**The LMX9830 will store the local BdAddress within NVS. Changing the BDAddress requires a HW Reset or "Reset" (6.2.18.3). If the BdAddress has the value 0xFFFFFFFFFFFF, the LMX9830 will stay in "Initialization Mode" (2.1) after reset or power-up.**	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	CHANGE_LOCAL_BDADDRESS	
DataLength	6	
Data	BdAddress 6 Bytes	The new Bluetooth address for the local device. The address is stored in NVS. In case no EEPROM

Table 6-225. Change Local BD_Addr Confirm

Description	**Confirms the request above**	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	CHANGE_LOCAL_BDADDRESS	
DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_RESET_TO_NSC_BDADDRESS ERROR_INVALID_NO_OF_PARAMETERS

6.2.14.5 Store Class of Device

Table 6-226. Store Class of Device Request

Description	**This command will store the class of device for the LMX9830 in NVS. The proper value for the class of device parameter is specified by the Bluetooth SIG.**
Firmware	0106, 02xx
PacketType	REQ
Opcode	STORE_CLASS_OF_DEVICE
DataLength	3

Table 6-226. Store Class of Device Request

Data	ClassOfDevice 3 bytes	The class of device value to be stored in NVS.

Table 6-227. Store Class of Device Confirm

Description	**Confirms the request above**	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	STORE_CLASS_OF_DEVICE	
DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_INVALID_NO_OF_PARAMETERS

6.2.14.6 Set Scan Mode - Discoverability / Connectability

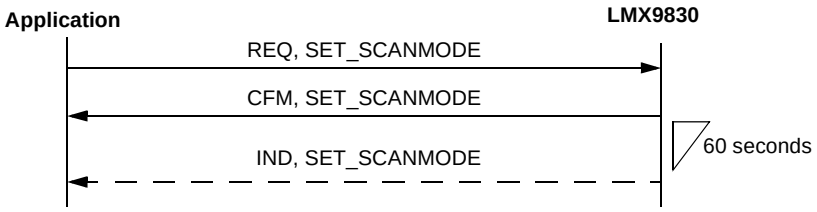


Table 6-228. Set Scan Mode Request

Description	**Change the Bluetooth scan mode. Automatic limited discoverable mode automatically toggles between general and limited inquiry scanning. This mode defined by the Bluetooth GAP profile specification, refer to part K.1, section 6.2.1. The automatic Limited discoverable mode times out after 60 sec. At this point the LMX9830 sends the GAP_SET_SCANMODE indication and resets page and inquiry scan settings to the value that was stored before the automatic limited discoverable mode was entered. The Connectability mode and discoverability modes are stored in NVS and restored during startup. An exception are the limited discoverable mode and automatic limited discoverable modes are selected, in this case neither connectability mode or discoverability mode are stored in NVS.**	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	GAP_SET_SCANMODE	
DataLength	2	
Data	Connectability 1 byte	0x00 = Not connectable 0x01 = Connectable 0x81 = Connectable using Interlaced Scanning

Table 6-228. Set Scan Mode Request

	Discoverability 1 byte	0x00 = Non discoverable 0x01 = General discoverable 0x81 = General discoverable using Interlaced Scanning 0x02 = Limited discoverable 0x82 = Limited discoverable using Interlaced Scanning 0x03 = Automatic limited discoverable mode, see[3] 0x83 = Automatic limited discoverable mode using Interlaced Scanning

Table 6-229. Set Scan Mode Confirm

Description	**Confirm to the request above.**	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_SET_SCANMODE	
DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_INVALID_CONNECTABILITY_PARAMETER ERROR_INVALID_DISCOVERABILITY_PARAMETER ERROR_INVALID_NO_OF_PARAMETERS ERROR_UNKNOWN_ERROR

6.2.14.7 Automatic Limited Discoverable Mode Ended

Table 6-230. Automatic limited discoverable ended Indicator

Description	**Indication send from the device, when automatic limited discoverable mode has ended.**	
Firmware	0106, 02xx	
PacketType	IND	
Opcode	GAP_SET_SCANMODE	
DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_UNKNOWN_ERROR

6.2.14.8 Force Master Role

The Force Master Role is initiated only after boot up or reset. The role is stored as parameter within NVS.

If Force Master is activated the LMX9830 tries to switch its role to master if connected from another device. For this the LMX9830 is sending a Master/Slave switch request after link establishment to the remote device. If the switch is successful, the link setup will be continued and the LMX9830 is open for other incoming connections.
If the switch fails, the link will be dropped.

This functionality allows Access Point like applications on the LMX9830.

Table 6-231. Force Master Role Request

Description	**This command will change the preferred role of the LMX9830 e.g. to force master role at any connections. The LMX9830 must be reset to let the changes take effect.**
Firmware	0106, 02xx
PacketType	REQ

Table 6-231. Force Master Role Request

Opcode	FORCE_MASTER_ROLE	
DataLength	1	
Data	Role 1 bytes	0x00 Don't care about role 0x01 Force master role at connection setup

Table 6-232. Force Master Role Confirm

Description	**Confirms the request above**	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	FORCE_MASTER_ROLE	
DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_INVALID_ROLE ERROR_INVALID_NO_OF_PARAMETERS

6.2.14.9 Read Operation Mode

The Operation Mode of the LMX9830 controls the behavior in different situations. Please seeSection 2.2.2 "Operation States" on page 21 for details.

Table 6-233. Read Operation Mode Request

Description	**This command will read out the current operation mode.**	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	READ_OPERATION_MODE	
DataLength	0	
Data	None	

Table 6-234. Read Operation Mode Confirm

Description	**Confirms the request above**	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	READ_OPERATION_MODE	
DataLength	2	
Data	Status 1 byte	ERROR_OK ERROR_INVALID_NO_OF_PARAMETERS
	Mode 1 Byte	0x00 Automatic Operation OFF 0x01 Automatic Operation ON

6.2.14.10 Write Operation Mode

Table 6-235. Write Operation Mode Request

Description	**This command will change the operation mode stored in NVS. The new setting will take effect after a reset.**

Table 6-235. Write Operation Mode Request

Firmware	0106, 02xx	
PacketType	REQ	
Opcode	WRITE_OPERATION_MODE	
DataLength	1	
Data	Mode 1 Byte	0x00 Automatic Operation OFF 0x01 Automatic Operation ON

Table 6-236. Write Operation Mode Confirm

Description	**Confirms the request above**	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	WRITE_OPERATION_MODE	
DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_INVALID_NO_OF_PARAMETERS ERROR_INVALID_MODE

6.2.14.11 Set Default Link Latency

Table 6-237. Set Default Link Latency Request

Description	**This command is used to change the default SPP link latency. The default link latency is set during SPP connection setup. The default link latency setting is stored in NVS. The link latency is used to calculate a poll interval for the ACL link. The calculated poll interval may be bigger than the latency specified because of e.g. an SCO link or another ACL link. See also Section 4.1.2.7 on page 51 for the impact of the link latency.**	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SET_DEFAULT_LINK_LATENCY	
DataLength	2	
Data	Latency 2 byte	The link latency in slots. 0x0000: No link latency requirement (default) 0x0002 - 0x0190: Valid link latency In case 0x0000 is set the link latency is managed by the master, typically 40 slots (0x0028, 25ms) in a point-to-point connection are used

Table 6-238. Set Default Link Latency Confirm

Description	**Response to the request above.**	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SET_DEFAULT_LINK_LATENCY	
DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_INVALID_NO_OF_PARAMETERS ERROR_LIMIT

6.2.14.12 Get Default Link Latency

Table 6-239. Get Default Link Latency Request

Description	This command is used to read the default link latency setting from NVS.
Firmware	0106, 02xx
PacketType	REQ
Opcode	GET_DEFAULT_LINK_LATENCY
DataLength	0

Table 6-240. Get Default Link Latency Confirm

Description	Response to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GET_DEFAULT_LINK_LATENCY	
DataLength	2	
Data	Latency 2 byte	The link latency in slots. 0x0000: No link latency requirement (default) 0x0002 - 0x0190: Valid link latency In case 0x0000 is set the link latency is managed by the master, typically 40 slots (0x0028, 25ms) in a point-to-point connection are used

6.2.15 ACL Indications

If the Event Filter is set to “Report all events”, the LMX9830 indicates any established and terminated ACL link to the host. This feature enables the user to monitor the bluetooth physical interface. In case the establishment failed the indicators report the reason for terminating or not establishing the link. The ACL error events can be found inTable 6-297 "ACL Error Codes" on page 197.

6.2.15.1 ACL Established

Table 6-241. ACL Established Indicator

Description	This indication is sent to the host when an ACL link is established.	
Firmware	0106, 02xx	
PacketType	IND	
Opcode	GAP_ACL_ESTABLISHED	
DataLength	7	
Data	BdAddr 6 bytes	The Bluetooth address of the remote device.
	Status 1 byte	See “ACL Error Codes” (page 197)

6.2.15.2 ACL Terminated

Table 6-242. ACL Terminated Indicator

Description	This indication is sent to the host when an ACL link is terminated.
Firmware	0106, 02xx
PacketType	IND

Table 6-242. ACL Terminated Indicator

Opcode	GAP_ACL_TERMINATED	
DataLength	7	
Data	BdAddr 6 bytes	The Bluetooth address of the remote device.
	Status 1 byte	See “ACL Error Codes” (page 197)

6.2.16 Bluetooth Security

Bluetooth security is part of the Generic Access Profile GAP. It is controlled by:

- Security Mode
  - — Security Mode 1:
    - – No Security, the device never will ask for authentication or pairing.
  - — Security Mode 2:
    - – The level of security (Authorization, Authentication, Encryption) is determined by the setting in the service database entries. Each entry can have different security requirements.
    - – no authentication necessary for SDAP links
  - — Security Mode 3:
    - – Authentication already necessary on Link Manager level: SDAP links already require authentication (Service requests)
    - – Device always asks for authentication
- Service Database Entry (only for Security Mode 2)
  - — Each entry can specify the settings for authentication and encryption

The LMX9830 by default is in Security Mode 2.

6.2.16.1 Get Security Mode

Table 6-243. Get Security Mode Request

Description	**Reads the current security mode of the Bluetooth device.**	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	GAP_GET_SECURITY_MODE	
DataLength	0	
Data	None	

Table 6-244. Get Security Mode Confirm

Description	**Confirm to the request above.**	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_GET_SECURITY_MODE	
DataLength	2	
Data	Status 1 byte	ERROR_OK ERROR_INVALID_NO_OF_PARAMETERS

Table 6-244. Get Security Mode Confirm

	Mode 1 byte	The current Bluetooth security mode. 0x01 Security mode 1 0x02 Security mode 2 0x03 Security mode 3 0x83 Security mode 3 with link level encryption

6.2.16.2 Set Security Mode

Table 6-245. Set Security Mode Request

Description	**Changes the current security mode of the Bluetooth device. The security mode is stored in NVS and restored during power up.**	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	GAP_SET_SECURITY_MODE	
DataLength	1	
Data	Mode 1 byte	The current Bluetooth security mode. 0x01 Security mode 1 0x02 Security mode 2 0x03 Security mode 3 0x83 Security mode 3 with link level encryption

Table 6-246. Set Security Mode Confirm

Description	**Confirm to the request above.**	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_SET_SECURITY_MODE	
DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_INVALID_NO_OF_PARAMETERS ERROR_INVALID_SECURITY_MODE

6.2.17 Pairing

The pairing process is part of the authentication procedure. If a local or remote service asks for authentication during link establishment, the authentication process will check if a link key already exists between the two devices. If not, the Link Manager initiates the pairing process. Within this, the two devices exchange a PIN code and create a secure link key which will be stored in each device.

During next link setup, the authentication routine takes the existing link key and proceeds without this pairing procedure.

The LMX9830 has a fixed PinCode which can be changed with the command “Change fixed Pin”. This pin is used during any pairing procedure. In case the stored PinCode has length 0x00, the LMX9830 will request a Pin from the application using the “Pin Request” indicator. The application needs to respond with the appropriate pin in the Pin request.

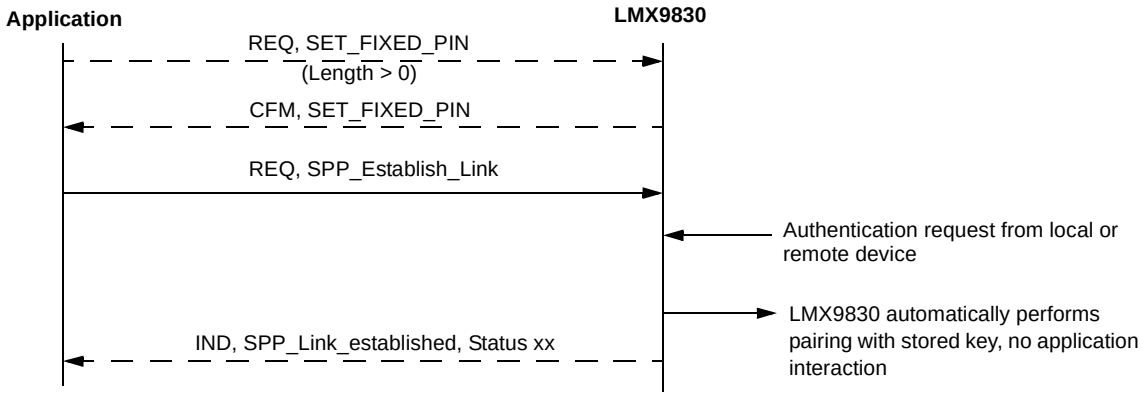


Figure 6-98. Authentication procedure with fixed pin

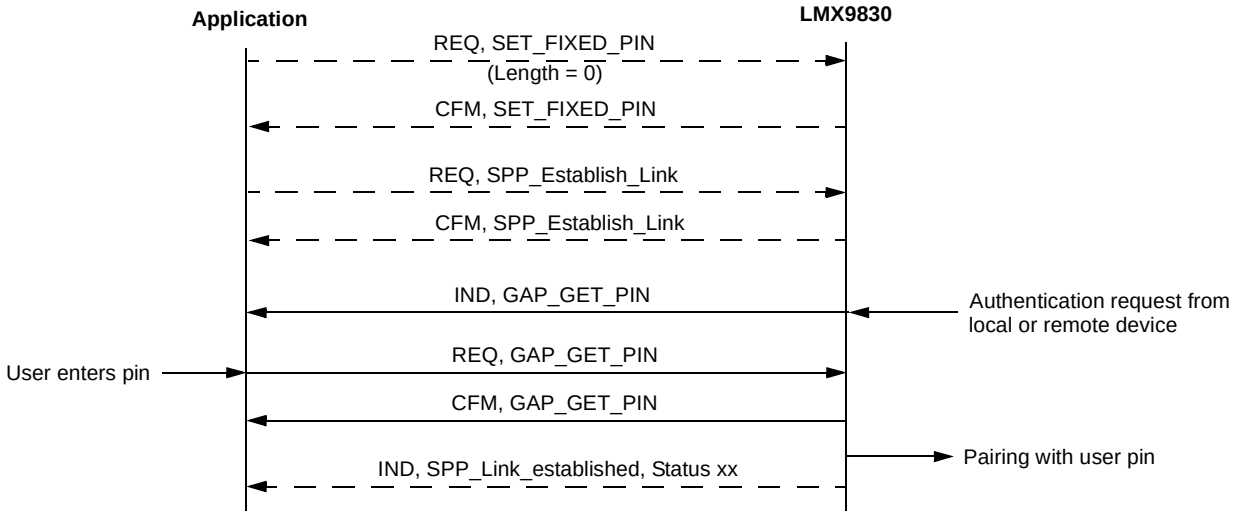


Figure 6-99. Authentication procedure with Pin request

6.2.17.1 Remove Pairing

Table 6-247. Remove Pairing Command

Description	**Remove pairing with a remote device.**	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	GAP_REMOVE_PAIRING	
DataLength	6	
	BdAddress 6 byte	Remove pairing to the BdAddress.

Table 6-248. Remove Paired Device Confirm

Description	**Response to the request above.**

Table 6-248. Remove Paired Device Confirm

Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_REMOVE_PAIRING	
DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_INVALID_NO_OF_PARAMETERS ERROR_LINKKEY_DOES_NOT_EXISTS

6.2.17.2 List Paired Devices

Table 6-249. List Paired Devices Command

Description	**Request a list of paired devices from NVS**
Firmware	0106, 02xx
PacketType	REQ
Opcode	GAP_LIST_PAIRED_DEVICES
DataLength	0

Table 6-250. List paired devices Confirm

Description	**Response to the request above.**	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	LIST_PAIRED_DEVICES	
DataLength	2 +6 * DeviceCount	
Data	Status 1 byte	ERROR_OK ERROR_INVALID_NO_OF_PARAMETERS
	DeviceCount 1 byte	Number of devices in the list of paired devices If 0 the device is not paired to any other devices. The maximum number of paired devices is 7.
	BdAddresses 6 byte * DeviceCount	The list of paired devices

6.2.17.3 Get Fixed Pin

Table 6-251. Get Fixed Pin Request

Description	**Reads the current fixed pin code from NVS**
Firmware	0106, 02xx
PacketType	REQ
Opcode	GAP_GET_FIXED_PIN
DataLength	0

Table 6-252. Get Fixed Pin Confirm

Description	**Response to the request above.**
Firmware	0106, 02xx

Table 6-252. Get Fixed Pin Confirm

PacketType	CFM	
Opcode	GAP_GET_ FIXED _PIN	
DataLength	2+Pinlength	
Data	Status 1 byte	ERROR_OK ERROR_INVALID_NO_OF_PARAMETERS
	Pinlength 1 byte	Length of pin code, in bytes. 0x00: No Fixed pin, LMX9830 will send "Pin request indicator" (6.2.17.5.1) in case a pin is required. Range: 0x01-0x10
	Pincode Pinlength bytes	PIN code used when the two Bluetooth devices are paired. The maximum length of a PIN code is 128 bits (16 bytes).

6.2.17.4 Set Fixed Pin

Table 6-253. Set Fixed Pin Request

Description	**Stores a new fixed pin code in NVS**	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	GAP_SET_FIXED_PIN	
DataLength	1+ Pinlength	
	Pinlength 1 byte	Length of pin code, in bytes. 0x00: No Fixed pin, LMX9830 will "Pin request indicator" (6.2.17.5.1) in case a pin is required. Range: 0x01-0x10
	Pincode Pinlength bytes	PIN code used when the two Bluetooth devices are paired. The maximum length of a PIN code is 128 bits (16 bytes).

Table 6-254. Set Fixed Pin Confirm

Description	**Response to the request above.**	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_SET_ FIXED _PIN	
DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_INVALID_NO_OF_PARAMETERS ERROR_PINCODE_TOO_LONG

6.2.17.5 PIN request

The PIN request procedure will only be used, in case the Pincode length stored in NVS is 0. See also Figure 6-99 on page 180.

6.2.17.5.1 Pin request indicator

Table 6-255. Pin request indicator

Description	This event is used to inform the Host when a PIN code is requested during authentication of an ACL link. This event is only generated if the length of the Fixed pin parameter stored in the NVS is set to 0.	
Firmware	02xx	
PacketType	IND	
Opcode	GAP_GET_PIN	
DataLength	6	
Data	BdAddr 6 byte	The Bluetooth device address of the remote device.

6.2.17.5.2 Pin request response

Table 6-256. Pin request response command

Description	This command is used to send a PIN code to the LMX9830 as response to a GAP_GET_PIN indication.	
Firmware	02xx	
PacketType	REQ	
Opcode	GAP_GET_PIN	
DataLength	1	
DataLength	7 + Pinlength	
Data	BdAddr 6 byte	The Bluetooth device address of the remote device.
	Pinlength 1 byte	Length of pin code, in bytes. Range: 0x00-0x10 0x00 indicates that the Host does not allow the authentication of the ACL link.
	Pincode Pinlength bytes	PIN code used when the two Bluetooth devices are paired. The maximum length of a PIN code is 128 bits (16 bytes).

Table 6-257. Pin request command confirm

Description	Response to the request above.	
Firmware	02xx	
PacketType	CFM	
Opcode	GAP_GET _PIN	
DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_INVALID_NO_OF_PARAMETERS ERROR_PINCODE_LENGTH

6.2.18 Hardware Commands

6.2.18.1 Set Event Filter

Table 6-258. Set Event Filter Command

Description	This command is used to set the event filter. The setting is stored in NVS.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SET_EVENT_FILTER	
DataLength	1	
Data	Filter 1 byte	0x00: All events reported 0x01: No ACL Link Indicators (default) 0x02: No events reported, UART break still generated and detected. 0x03: No events generated, UART break not generated or detected (full cable replacement)

Table 6-259. Set Event Filter Confirm

Description	Response to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SET_EVENT_FILTER	
DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_INVALID_NO_OF_PARAMETERS ERROR_LIMIT

6.2.18.2 Get Event Filter

Table 6-260. Get Event Filter Command

Description	This command reads the current event filter setting.
Firmware	0106, 02xx
PacketType	REQ
Opcode	GET_EVENT_FILTER
DataLength	0

Table 6-261. Get Event Filter Confirm

Description	Response to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GET_EVENT_FILTER	
DataLength	1	
Data	Filter 1 byte	0x00: All events reported 0x01: No ACL Link Indicators (default) 0x02: No events reported, UART break still generated and detected. 0x03: No events generated, UART break not generated or detected (full cable replacement)

6.2.18.3 Reset

Table 6-262. Reset Request

Description	**This command will perform a soft reset of the LMX9830. The LMX9830 will send a LMX9830_READY indication when it has performed the reset.**	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	RESET	
DataLength	0	
Data	None	

6.2.18.4 LMX9830 Ready

Table 6-263. Simply Blue Ready Indicator

Description	**The LMX9830 will send this indication to the host when the device is fully initialized and ready to receive commands from the host.**	
Firmware	0106, 02xx	
PacketType	IND	
Opcode	DEVICE_READY	
DataLength	1+Length	
Data	Length 1 byte	Number of bytes in software version string
	version <Length> bytes	ASCII string containing the software version. e.g. “0210” indicating that the software version is version 2.10.

6.2.18.5 Restore Factory Settings

Table 6-264. Restore Factory Settings Request

Description	**This command will restore the LMX9830 configuration in NVS to factory settings except the Bluetooth address of the device, which can be restored otherwise. The factory settings are similar to all default settings listed in Table 1-1 "LMX9830 System Parameters, EEPROM Memory Map" on page 10. The LMX9830 needs to be restarted in order to let the changes take effect. No commands that write to NVS should be sent to the device in between RESTORE_FACTORY_SETTINGS REQ and CFM. When CFM is sent all the factory settings are completed.**	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	RESTORE_FACTORY_SETTINGS	
DataLength	0	
Data	none	

Table 6-265. Restore Factory Settings Confirm

Description	**Confirms the request above**
Firmware	0106, 02xx
PacketType	CFM

Table 6-265. Restore Factory Settings Confirm

Opcode	RESTORE_FACTORY_SETTINGS	
DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_INVALID_NO_OF_PARAMETERS

6.2.18.6 Change NVS UART Speed

The UART Speed is only active if the OP pin configuration of the LMX9830 is set to “Read NVS Settings” or “Autobaudrate”. Changed UART speeds get active after reset.

Table 6-266. Change UART Speed Request

Description	**This command will change the UART speed stored in NVS. The new UART speed will be used after a reset.**	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	CHANGE_NVS_UART_SPEED	
DataLength	1	
Data	UartSpeed 1 Byte	The UART speed to be stored in NVS 0x00 = 2400 0x01 = 4800 0x02 = 7200 0x03 = 9600 0x04 = 19200 0x05 = 38400 0x06 = 57600 0x07 = 115200 0x08 = 230400 0x09 = 460800 0x0A = 921600

Table 6-267. Change UART Speed Confirm

Description	**Confirms the request above**	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	CHANGE_NVS_UART_SPEED	
DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_UART_SPEED_OUT_OF_RANGE ERROR_INVALID_NO_OF_PARAMETERS

6.2.18.7 Change UART Settings

UART Settings will be stored in NVS and are valid for ALL UART speeds at LMX9830.

Table 6-268. Change UART Settings Request

Description	**This command will change the UART settings stored in NVS. The new UART settings will be used after a reset.**
Firmware	0106, 02xx
PacketType	REQ

Table 6-268. Change UART Settings Request

Opcode	CHANGE_UART_SETTINGS	
DataLength	2	
Data	ParityType 1 Byte	0x00 None 0x01 Even 0x02 Odd
	StopBits 1 Byte	0x00 One Stop bit 0x01 Two Stop bits

Table 6-269. Change UART Settings Confirm

Description	**Confirms the request above**	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	CHANGE_UART_SETTINGS	
DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_PARITY_BIT_OUT_OF_RANGE, ERROR_STOP_BITS_OUT_OF_RANGE, ERROR_INVALID_NO_OF_PARAMETERS

6.2.19 Test Modes

6.2.19.1 Bluetooth Test Mode

Bluetooth Qualification requires specific test modes to prove the functionality and quality of the bluetooth device.

The Test Mode Command offers the ability to enable either the bluetooth specified “Device Under Test” Mode and also a UART Loopback mode.

This modes can only be left by a reset.

Table 6-270. Initiate Bluetooth Test Mode Request

Description	**This command will activate the test mode. To exit the test mode, a reset of the device must be performed. If local loopback mode is activated all data send to the device though the UART, are send back to the host. The device can only leave local loopback mode by performing a hardware reset.**	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	TEST_MODE	
DataLength	1	
Data	Mode 1 byte	The Bluetooth test mode to enter: 0x01 Enable Bluetooth test mode 0x02 Enable local loopback mode

Table 6-271. Initiate Bluetooth Test Mode Confirm

Description	**Confirms the request above**
Firmware	0106, 02xx
PacketType	CFM
Opcode	TEST_MODE

Table 6-271. Initiate Bluetooth Test Mode Confirm

DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_ILLEGAL_TESTMODE ERROR_UNKNOWN_ERROR ERROR_INVALID_NO_OF_PARAMETERS

6.2.19.2 Initiate RF Test Mode

In addition to the standard Bluetooth Test Mode the LMX9830 offers a special RF Test Mode, switching the transmitter into a continuos transmit or receive mode. This is needed for Bluetooth qualification as well as regulatory testings for FCC and ETSI and will be needed for country specific qualification.

Table 6-272. Initiate RF Test Mode Request

Description	**This command will activate the RF test mode. To exit the test mode, a reset of the device must be performed. The transmitter test must be stopped before a new test with changed parameters can be started.**	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	RF_TEST_MODE	
DataLength	14	
Data	Test 1 byte	Test Scenario: 0x00: Stop transmit 0x01: Burst transmit, take payload from this command 0x02: Burst transmit, use PRBS-9 sequence 0x03: Continuos transmit, sends all-1 if ModulationPayload is non-zero, otherwise all-0 0x04: Hopping receive 0x05: continuos receive at fixed channel
	Channel 1 byte	Channel number 0 – 78 0: 2402 MHz 78: 2480 MHz 255 (0xFF): Hopping in connection state
	PaCtrl 1 byte	Reserved for future use

Table 6-272. Initiate RF Test Mode Request

	ModulationCtrl 1 byte	Modulation Control: 0: No modulation 1: Access code only (68 us TX data every 1250 us – note 1) 2: Bluetooth DH1 packet with defined payload (note 2) 3: Bluetooth DM1 packet (17 bytes) 4: Bluetooth DH1 packet (27 bytes) 5: Invalid 6: Invalid 7: Bluetooth HV3 packet (30 bytes) 8: Invalid 9: Bluetooth AUX1 packet (29 bytes) 10: Bluetooth DM3 packet (121 bytes) 11: Bluetooth DH3 packet (183 bytes) 12: Invalid 13: Invalid 14: Bluetooth DM5 packet (224 bytes) 15: Bluetooth DH5 packet (339 bytes)
	ModulationPattern 8 bytes	Defines Access Code modulation if ModulationCtrl = 1, else ignored.
	ModulationPayload 1 byte	Defines one byte of payload repeated through packet if Test = 1, else ignored
	Options 1byte	Bit 0: Whitening: 1: enabled, 0: disabled

Table 6-273. Initiate RF Test Mode Confirm

Description	Confirms the request above	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	RF_TEST_MODE	
DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_INVALID_NO_OF_PARAMETERS

6.2.20 LMX9830 Initialization

The LMX9830 requires several parameters to be able to enter Bluetooth Mode. Usually on power-up the device checks the EEPROM for those parameters and copies them into the RAM mirror. If no EEPROM is connected or one of the parameters is not set, the device will stay in "Initialization Mode" (2.1) until the parameters are available in RAM.

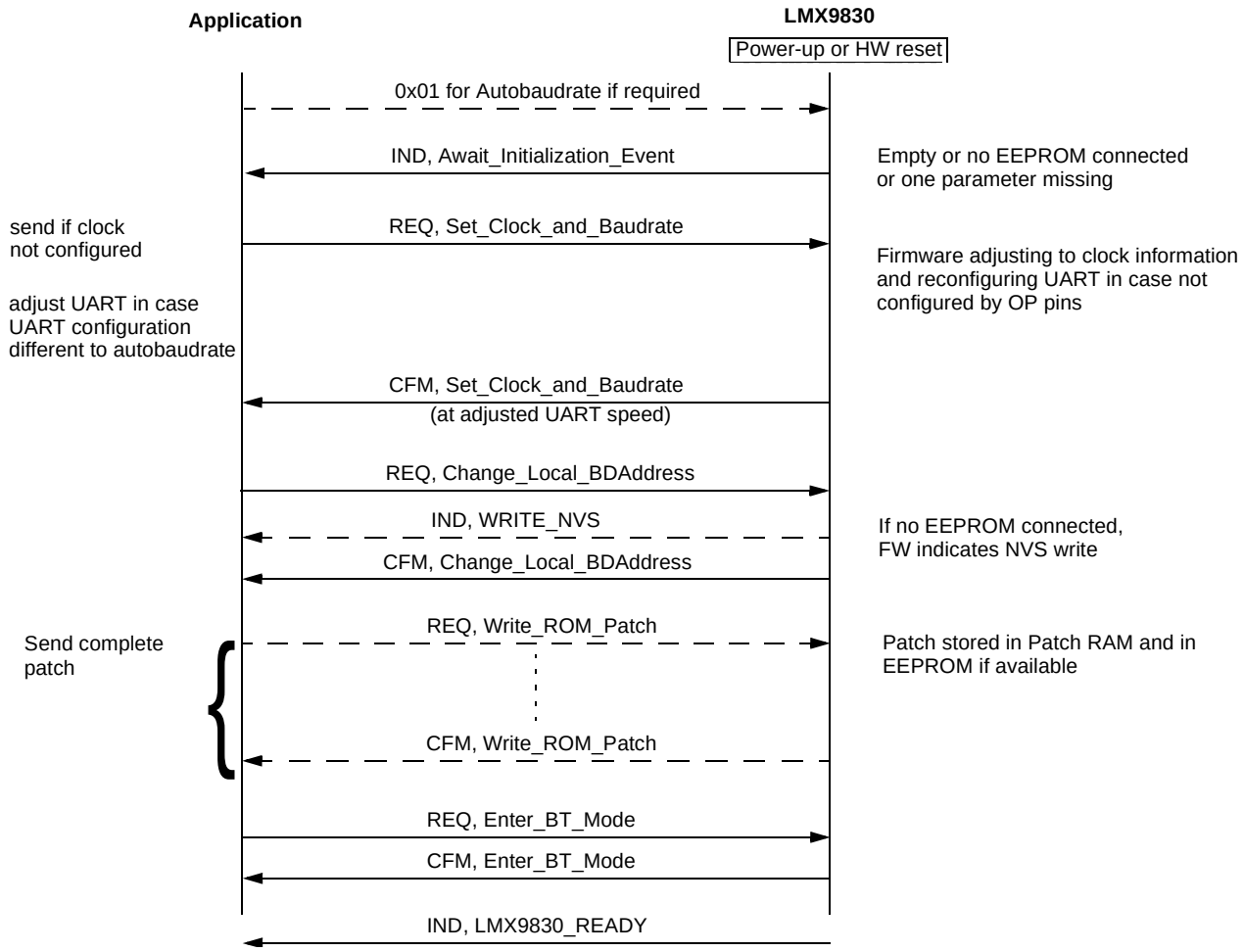
The following parameters need at least to be configured:

- Bluetooth Address (BD_ADDR)
- Clock Information (required in case the OP pins are set to 10-20MHz[1])
- UART Baudrate (required in case the OP pins are set to “Read from NVS” or “Autobaudrate”[1])

If one of these parameters is missing the device will report the"Await Initialization Event" (6.2.20.1)which then requires to set BD_ADDR and possibly clock and UART baudrate. In addition, the initialization phase can be used for patches and

1.See [1] "Texas Instruments: LMX9830 Datasheet" for the OP pin configurations.

configurations, which require a reset anyway. The initialization phase finally is closed by the "Enter Bluetooth Mode" (6.2.20.2).



6.2.20.1 Await Initialization Event

Table 6-274. Await Initialization Indicator

Description	**The LMX9830 will send this indication to the host right after a reset when the device needs initialization parameters before it can continue.**	
Firmware	0106, 02xx	
PacketType	IND	
Opcode	AWAIT_INITIALIZATION_EVENT	
DataLength	0	
Data	None	

6.2.20.2 Enter Bluetooth Mode

Table 6-275. Enter Bluetooth Mode Request

Description	**The host uses this command to request the SimplyBlue module to enter Bluetooth mode after initialization parameters have been sent to the module.**
PacketType	REQ
Opcode	ENTER_BLUETOOTH_MODE

Table 6-275. Enter Bluetooth Mode Request

DataLength	0	
Data	None	

Table 6-276. Enter Bluetooth Mode Confirm

Description	**Confirms the request above**	
PacketType	CFM	
Opcode	ENTER_BLUETOOTH_MODE	
DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_COMMAND_DISALLOWED ERROR_INVALID_NO_OF_PARAMETERS

6.2.20.3 Set Clock And Baud Rate

Table 6-277. Set Clock and Baudrate Request

Description	**This command is used to write the exact base band clock frequency and baud rate used. This command must be sent as the first command if UART auto baud rate detection is used. In case of first power-up with EEPROM, 9600 will be used. In case**	
PacketType	REQ	
Opcode	SET_CLOCK_AND_BAUDRATE	
DataLength	8	
Data	ClockFrequency 4 bytes	The Clock Frequency to use.
	BaudRate 4 bytes	The baud rate to use.

Table 6-278. Set Clock and Baudrate Confirm

Description	**Response to the request above.**	
PacketType	CFM	
Opcode	SET_CLOCK_AND_BAUDRATE	
DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_INVALID_NO_OF_PARAMETERS ERROR_COMMAND_DISALLOWED

6.2.21 Patching

Patching allows to correct small errors in ROM code. A patch has to be downloaded to RAM NVS from the host or fetched by the EEPROM before entering "Initialization Mode" (2.1). Please refer also to Section 4.6 "Software Patches" on page 75.

6.2.21.1 Write ROM Patch

Table 6-279. Write ROM Patch Request

Description	**This command is used to store ROM patch code in the SimplyBlue module.**
PacketType	REQ

Table 6-279. Write ROM Patch Request

Opcode	WRITE_ROM_PATCH	
DataLength	1 to 333 bytes	
Data	Patch data	Stream of Patch Data.

Table 6-280. Write ROM Patch Confirm

Description	**Confirms the request above.**	
PacketType	CFM	
Opcode	WRITE_ROM_PATCH	
DataLength	1	
Data	Status 1 byte	0x00 = Ok, Done 0x01 = Ok, Continue 0x80 = Error, Not enough info to continue download (a “global” or “segment” meta data group appears to be split) 0x81 = Error, Not enough resources to continue download 0x82 = Error, Patch too big 0x83 = Error, Unsupported Patch format revision 0x84 = Error, Patch not applicable to firmware in device. 0x85 = Error, Patch CRC check failed. 0x86 = Error, Patch NVS validation failed. 0x87 = Error, RAMAddr or VarAddr out of RAM area. 0xFF = Error, Unspecified error.

6.2.22 GPIO handling

The LMX9830 offers three GPIOs which can be controlled and read back by the command interface. By default pins PG6 and PG7 are used by the firmware to indicate pin status and RF link activity. The following commands allow to change the usage of the pins and to configure them to the desired function.

Preparation:

In order to use pins PG6 and PG7 as GPIOs, EEPROM parameter #19 (Table 1-1 on page 10) at address 0x0062 needs to be configured with the "Write NVS" (6.2.23.2) command. This control byte configures the Enhanced Power management and the GPIO pin usage.

For example:
To use PG6 and PG7 as standard GPIOs and leave Enhanced Power Management disabled (default) use the following hex string:

02 52 73 04 00 C9 62 00 01 **F9** 03

The change requires a "Reset" (6.2.18.3) to get active.

Using the GPIO commands:

Once this is done, the GPIOs can be configured with of the following GPIO commands. At each of the commands, the pin names are mapped to a specific bit position. Please see Table 6-281 for the pin assignment.

Table 6-281. Control byte pin assignments

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Pin	PG7	PG6	(reserved)	PG4	(reserved)	(reserved)	(reserved)	(reserved)

6.2.22.1 Set GPIO Weak Pull-up

Each GPIO can be configured with internal Weak pull-ups, in case they are used as Input. With this a default input value of HIGH can be reached.

Table 6-282. Set GPIO WPU Request

Description	This command is used to enable or disable the weak pull up resistors on GPIO PG[4], PG[6] and PG[7].	
PacketType	REQ	
Opcode	SET_GPIO_WPU	
DataLength	1	
Data	WPU 1 bytes	If 1, the WPU resistor on PG[x] is enabled. If 0, the WPU resistor on PG[x] is disabled. See Table 6-281 on page 192 for the bit assignments of the pins.

Table 6-283. Set GPIO WPU Confirm

Description	Confirms the request above.	
PacketType	CFM	
Opcode	SET_GPIO_WPU	
DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_UNEXPECTED ERROR_INVALID_NO_OF_PARAMETERS

6.2.22.2 Get GPIO Input State

This command allows to read back the status of the pins.

Table 6-284. Get GPIO Input State Request

Description	This command is used to read the PGDIN_REG register and return the state of PG[4], PG[6] and PG[7].	
PacketType	REQ	
Opcode	GET_GPIO_STATUS	
DataLength	1	
Data	None	

Table 6-285. Get GPIO Status Confirm

Description	Confirms the request above.	
PacketType	CFM	
Opcode	GET_GPIO_STATUS	
DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_INVALID_NO_OF_PARAMETERS
	State 1 bytes	PG[x] is 1 if the input is high and 0 if the input is low. See Table 6-281 on page 192 for the bit assignments of the pins.

6.2.22.3 Set GPIO Direction

This command allows to configure the direction of the GPIO pins in case they are not used by the bluetooth firmware.

Table 6-286. Set GPIO Direction Request

Description	**This command is used to set the direction of GPIO PG[4], PG[6] and PG[7].**	
PacketType	REQ	
Opcode	SET_GPIO_DIRECTION	
DataLength	1	
Data	Direction 1 bytes	If 1, PG[x] is output. If 0, PG[x] is input.

Table 6-287. Set GPIO Direction Confirm

Description	**Confirms the request above.**	
PacketType	CFM	
Opcode	SET_GPIO_DIRECTION	
DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_UNEXPECTED ERROR_INVALID_NO_OF_PARAMETERS

6.2.22.4 Set GPIO Output High

Table 6-288. Set GPIO Output High Request

Description	**This command is used to set the output pins on PG[4], PG[6] and PG[7] to logical high.**	
PacketType	REQ	
Opcode	SET_GPIO_OUTPUT_HIGH	
DataLength	1	
Data	Output 1 bytes	If 1, PG[x] output is set high. If 0, PG[x] output is unchanged. See Table 6-281 on page 192 for the bit assignments of the pins.

Table 6-289. Set GPIO Output High Confirm

Description	**Confirms the request above.**	
PacketType	CFM	
Opcode	SET_GPIO_OUTPUT_HIGH	
DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_UNEXPECTED ERROR_INVALID_NO_OF_PARAMETERS

6.2.22.5 Set GPIO Output Low

Table 6-290. Set GPIO Output Low Request

Description	**This command is used to set the output pins on PG[4], PG[6] and PG[7] to logical low.**

Table 6-290. Set GPIO Output Low Request

PacketType	REQ	
Opcode	SET_GPIO_OUTPUT_LOW	
DataLength	1	
Data	Output 1 bytes	If 1, PG[x] output is set low. If 0, PG[x] output is unchanged.

Table 6-291. Set GPIO Output Low Confirm

Description	**Confirms the request above.**	
PacketType	CFM	
Opcode	SET_GPIO_OUTPUT_LOW	
DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_UNEXPECTED ERROR_INVALID_NO_OF_PARAMETERS

6.2.23 EEPROM / NVS handling

On each boot-up or reset the LMX9830 tries to read all parameters required for operation from an external EEPROM and copies them into a “RAM NVS” which is used during operation. The EEPROM (also called **N**on **V**olatile **S**torage) can be addressed by Read and Write commands to change configuration parameters directly. The EEPROM memory map is listed in Table 1-1 "LMX9830 System Parameters, EEPROM Memory Map" on page 10.

In case no EEPROM is connected, these commands will just change/read the RAM NVS. The handling is the same. In addition, since the user doesn’t know possible changes, the firmware reports each change to the RAM NVS by the "Write NVS Indicator" (6.2.23.3). This allows the host to keep track of all NVS parameters and supersedes the need to completely read back the NVS before power down. A typical example is the storage of a link key after an pairing procedure. The link key will be reported to the host and can be restored after a power-down.

6.2.23.1 Read NVS

Table 6-292. Read NVS Request

Description	**This command can be used to read from NVS which is either in EEPROM or from RAM, depending on the availability of an EEPROM.**	
PacketType	REQ	
Opcode	READ_NVS	
DataLength	3	
Data	Address 2 bytes	The address. See Table 1-1 on page 10 for the memory map.
	Length 1 byte	The number of bytes that should be read.

Table 6-293. Read NVS Confirm

Description	**Confirms the request above.**	
PacketType	CFM	
Opcode	READ_NVS	
DataLength	4 + Length	
Data	Status 1 byte	ERROR_OK ERROR_INVALID_NO_OF_PARAMETERS

Table 6-293. Read NVS Confirm

	Address 2 bytes	The address. See Table 1-1 on page 10 for the memory map.
	Length 1 byte	The number of bytes that has been read.
	Data < Length > bytes	The data.

6.2.23.2 Write NVS

Table 6-294. Write NVS Request

Description	**This command can be used to write data to the NVS, which is either stored in EEPROM or in RAM depending on the availability of an EEPROM.**	
PacketType	REQ	
Opcode	WRITE_NVS	
DataLength	3 + Length	
Data	Address 2 bytes	The address. See Table 1-1 on page 10 for the memory map.
	Length 1 byte	The number of bytes that should be written.
	Data < Length > bytes	The data.

Table 6-295. Write NVS Confirm

Description	**Confirms the request above.**	
PacketType	CFM	
Opcode	WRITE_NVS	
DataLength	4	
Data	Status 1 byte	ERROR_OK ERROR_INVALID_NO_OF_PARAMETERS
	Address 2 bytes	The address. See Table 1-1 on page 10 for the memory map.
	Length 1 byte	The number of bytes that has been written

6.2.23.3 Write NVS Indicator

Table 6-296. Write NVS Indicator

Description	**This event is used to inform the Host about the data written to the NVS in the LMX9830. The device generates this event when no external EEPROM is connected.**	
PacketType	IND	
Opcode	WRITE_NVS	
DataLength	3 + Length	
Data	Address 2 bytes	The address. See Table 1-1 on page 10 for the memory map.

Table 6-296. Write NVS Indicator

	Length 1 byte	The number of bytes that should be written.
	Data < Length > bytes	The data.

6.3 ERROR CODES

6.3.1 ACL Error Codes

The following table is copied out of.

Table 6-297. ACL Error Codes

Error Code	Description
0x01	Unknown HCI Command.
0x02	No Connection.
0x03	Hardware Failure.
0x04	Page Timeout.
0x05	Authentication Failure.
0x06	Key Missing.
0x07	Memory Full.
0x08	Connection Timeout.
0x09	Max Number Of Connections.
0x0A	Max Number Of SCO Connections To A Device.
0x0B	ACL connection already exists.
0x0C	Command Disallowed.
0x0D	Host Rejected due to limited resources.
0x0E	Host Rejected due to security reasons.
0x0F	Host Rejected due to remote device is only a personal device.
0x10	Host Timeout.
0x11	Unsupported Feature or Parameter Value.
0x12	Invalid HCI Command Parameters.
0x13	Other End Terminated Connection: User Ended Connection.
0x14	Other End Terminated Connection: Low Resources.
0x15	Other End Terminated Connection: About to Power Off.
0x16	Connection Terminated by Local Host.
0x17	Repeated Attempts.
0x18	Pairing Not Allowed.
0x19	Unknown LMP PDU.
0x1A	Unsupported Remote Feature.
0x1B	SCO Offset Rejected.
0x1C	SCO Interval Rejected.
0x1D	SCO Air Mode Rejected.
0x1E	Invalid LMP Parameters.

Table 6-297. ACL Error Codes

0x1F	Unspecified Error.
0x20	Unsupported LMP Parameter Value.
0x21	Role Change Not Allowed
0x22	LMP Response Timeout
0x23	LMP Error Transaction Collision
0x24	LMP PDU Not Allowed
0x25	Encryption Mode Not Acceptable
0x26	Unit Key Used
0x27	QoS is Not Supported
0x28	Instant Passed
0x29	Pairing with Unit Key Not Supported
0x2A	Different Transaction Collision
0x2B	Reserved
0x2C	QoS Unacceptable Parameter
0x2D	QoS Rejected
0x2E	Channel Classification Not Supported
0x2F	Insufficient Security
0x30	Parameter out of Mandatory Range
0x31	Reserved
0x32	Role Switch Pending
0x33	Reserved
0x34	Reserved Slot Violation
0x35	Role Switch Failed

6.3.2 Generic error codes

Table 6-298. Generic Error Codes

Error code	Macro	Description
0x00	ERROR_OK	No error.
0x01	ERROR_INVALID_NO_OF_PARAMETERS	The number of bytes in the request does not correspond to the protocol specification
0x02	ERROR_DURATION_OUT_OF_RANGE	The given duration value is not valid according to the specification.
0x03	ERROR_INVALID_MODE	The selected mode is not valid according to the specification
0x04	ERROR_TIMEOUT	A timeout occurred.
0x05	ERROR_UNKNOWN_ERROR	An unknown error occurred.
0x06	ERROR_NAME_TOO_LONG	The number of bytes in the name string is longer than the maximum specified value.
0x07	ERROR_INVALID_DISCOVERABILITY_PARAMETER	The given discoverability parameter does not contain a valid value according to the specification.
0x08	ERROR_INVALID_CONNECTABILITY_PARAMETER	The given connectability parameter does not contain a valid value according to the specification.
0x09	ERROR_INVALID_SECURITY_MODE	The given security mode is not a valid Bluetooth security mode.

Table 6-298. Generic Error Codes

0x0a	ERROR_LINKKEY_DOES_NOT_EXISTS	No link key exists for the given Bluetooth address
0x0b	ERROR_CONNECTION_FAILED	The connection setup failed due to unknown reasons.
0x0c	ERROR_TRUNCATED_ANSWER	The returned number of services is too large to be handled by the LMX9830. The answer is truncated
0x0d	ERROR_RESULT_TOO_LARGE	The SDP result from the remote device is too large to be handled by the LMX9830 due to ram limitations
0x0e	ERROR_NOT_POSSIBLE_TO_ENTER_TESTMODE	It is currently not possible to enter the selected test mode.
0x0f	ERROR_ILLEGAL_TESTMODE	The given test mode is not a valid test mode according to the specification
0x10	ERROR_RESET_TO_NSC_BDADDRESS	The LMX9830 will change the Bluetooth address to the NSC address.
0x11	ERROR_UART_SPEED_OUT_OF_RANGE	The selected UART speed value is not valid according to the specification.
0x12	ERROR_INVALID_PORT	The given port value is larger than the maximum specified value.
0x13	ERROR_ILLEGAL_STATE_VALUE	The given state value is not a valid state according to the specification
0x14	ERROR_IDENTIFIER_OUT_OF_RANGE	The given identifier is larger than the maximum specified value.
0x15	ERROR_RECORD_ALREADY_IN_SELECTED_STATE	The service record is already enabled/disabled.
0x16	ERROR_INVALID_AUTHENTICATION_VALUE	The given authentication value is not a valid value according to the specification.
0x17	ERROR_INVALID_ENCRYPTION_VALUE	The given encryption value is not a valid value according to the specification.
0x18	ERROR_MAXIMUM_NO_OF_SERVICE_RECORDS_REACHED	The maximum number of service records, which the LMX9830 is able to store, is reached.
0x19	ERROR_WRITING_TO_NVS	An error occurred while writing to flash. The service record may not be stored.
0x1a	ERROR_INVALID_ROLE	The given role value is not a valid value according to the specification.
0x1b	ERROR_LIMIT	Limits exceeded (Parameter(s) violates limits).
0x1c	ERROR_UNEXPECTED	Unexpected at this moment.
0x1d	ERROR_UNABLE_TO_SEND	Could not send at this moment, no reason specified.
0x1e	ERROR_CURRENTLY_NO_BUFFER	Currently no room in buffer, try again later.
0x1f	ERROR_NO_CONNECTION	Trying to use an inexistent connection.
0x20	ERROR_SPP_INVALID_PORT	Port number out of range.
0x21	ERROR_SPP_PORT_NOT_OPEN	Port is closed.
0x22	ERROR_SPP_PORT_BUSY	Connection establishment on a PORT that has a connection.
0x23	ERROR_SPP_MULTIPLE_CONNECTIONS	Transparent mode attempted while more than 1 connection active.
0x24	ERROR_SPP_MULTIPLE_TRANSPARENT	Trying to store a default connection when a transparent default connection is already stored, or trying to store a transparent default connection when another connection is already stored.
0x25	ERROR_SPP_DEFAULT_CONNECTION_NOT_STORED	Trying to connection to a default connection, which is not stored.

Table 6-298. Generic Error Codes

0x26	ERROR_SPP_AUTOMATIC_CONNECTIONS_PROGRESSING	Trying to start connecting to default connections when default connection establishment is already progressing.
0x27	ERROR_UNSPECIFIED_ERROR	Other error.
0x28	ERROR_IDENTIFIER_NOT_IN_USE	Trying to enable a SDP record which is not stored. Wrong identifier.
0x29	ERROR_INVALID_SUPPORTED_FAXCLASS_VALUE	Faxclass parameter must be 0 or 1.
0x2a	ERROR_TOO_MANY_SUPPORTED_FORMATS	The given number of supported formats excesses the specified maximum number of supported formats.
0x2b	ERROR_TOO_MANY_DATASTORES	The given number of data stores excess the specified maximum number of data stores.
0x2C	ERROR_ATTEMPT_FAILED	Attempt to change low power mode failed
0x2D	ERROR_ILLEGAL_LINK_POLICY	The given link policy value is out of range
0x2E	ERROR_PINCODE_TOO_LONG	The pin code length field is too large.
0x2F	ERROR_PARITY_BIT_OUT_OF_RANGE	The given parity check is out of range
0x30	ERROR_STOP_BITS_OUT_OF_RANGE	The given number of stop bits is out of range
0x31	ERROR_ILLEGAL_LINK_TIMEOUT	The given link timeout value is out of range
0x32	ERROR_COMMAND_DISALLOWED	The command is not allowed.
0x33	ERROR_ILLEGAL_AUDIO_CODEC_TYPE	The given Audio CODEC type is out of range.
0x34	ERROR_ILLEGAL_AUDIO_AIR_FORMAT	The given Audio Air format is out of range.
0x35	ERROR_SDP_RECORD_TOO_LONG	The SDP record is too long.
0x36	ERROR_SDP_FAILED_TO_CREATE_RECORD	The SDP server failed to create the SDP record.
0x37	ERROR_SET_VOLUME_FAILED	The selected codec does not support volume control.
0x38	ERROR_ILLEGAL_PACKET_TYPE	The packet type specified in the request is not valid.
0x39	ERROR_INVALID_CODEC_SETTING	The codec (slave) settings is invalid.

6.3.3 RFCOMM Error Codes

Table 6-299. RFCOMM Error Codes

Error code	Macro	Description
0x00	RFCS_NO_ERROR	No error
0x01	RFCS_INVALID_DLC	The DLC does not exist
0x02	RFCS_INVALID_PORT	The port does not exist
0x03	RFCS_DLC_ESTABLISH_FAILED	The DLC establishment failed
0x04	RFCS_ACCESS_REJECTED	SECM did not authorize access to the requested service (DLC)
0x05	RFCS_INVALID_CONNECTION	There does not exist a DLC/L2CAP connection to the device
0xFF	RFCS_UNSPECIFIED_ERROR	Not used

6.3.4 RFCOMM Release Reasons

Table 6-300. RFComm Release Reasons

Error code	Macro	Description
0x00	RFCR_DLC_DISC_LOCAL_DEVICE	The local device has disconnected the DLC.
0x01	RFCR_DLC_DISC_REMOTE_DEVICE	The remote device has disconnected the DLC.
0x02	RFCR_DLC_DISC_ACL_FAILURE	ACL link failure/ link supervision timeout.
0x03	RFCR_DLC_DISC_LOWER_LAYER	Lower layer (e.g. L2CAP) has disconnected the DLC.

6.4 AT COMMANDS

The list of AT Commands is derived out of the “Bluetooth Profiles Book V1.1” from the Bluetooth SIG. All commands shall be implemented as described in International Telecommunication Union, “ITU-T Recommendation V.250”

6.4.1 DUN GW

6.4.1.1 Required commands

Table 6-301. Required AT Commands for DUN GW

Name	Description
&C	Circuit 109 (Received line signal detector) Behavior
&D	Circuit 108 (Data terminal ready) Behavior
&F	Set to Factory-defined Configuration
+GCAP	Request Complete Capabilities List
+GMI	Request Manufacturer Identification
+GMM	Request Model Identification
+GMR	Request Revision Identification
A	Answer
D	Dial
E	Command Echo
H	Hook Control
L	Monitor Speaker Loudness
M	Monitor Speaker Mode
O	Return to Online Data State
P	Select Pulse Dialling
Q	Result Code Suppression
S0	Automatic Answer
S10	Automatic Disconnect Delay
S3	Command Line Termination Character
S4	Response Formatting Character
S5	Command Line Editing Character
S6	Pause Before Blind Dialling
S7	Connection Completion Timeout
S8	Comma Dial Modifier Time
T	Select Tone Dialling
V	DCE Response Format
X	Result Code Selection and Call Progress Monitoring Control
Z	Reset To Default Configuration

6.4.1.2 Required Result Codes

Table 6-302. Result Codes for DUN

Name	Description
&C	Circuit 109 (Received line signal detector) Behavior

Table 6-302. Result Codes for DUN

Name	Description
OK	Acknowledges execution of a command.
CONNECT	Connection has been established.
RING	The DCE has detected an incoming call signal from the network.
NO CARRIER	The connection has been terminated, or the attempt to establish a connection failed.
ERROR	Error.
NO DIALTONE	No dial-tone detected.
BUSY	Busy signal detected.

6.4.2 Headset Profile

The following AT commands are defined for the headset profile.

6.4.2.1 Headset AT Command

Table 6-303. AT Commands for Headset profile

AT Capability	Syntax	Description	Values
RING		The Incoming call indication of ITU V.250, Section 6.3.4	
Microphone gain	+VGM=<gain>	Unsolicited result code issued by the HSAG to set the microphone gain of the HS. <gain> is a decimal numeric constant, relating to a particular (implementation-dependent) volume level controlled by the HS.	<gain>: 0-15
Speaker gain	+VGS=<gain>	Unsolicited result code issued by the HSAG to set the speaker gain of the HS. <gain> is a decimal numeric constant, relating to a particular (implementation-dependent) volume level controlled by the HS.	<gain>: 0-15
Microphone gain level report	+VGM=<gain>	Command issued by the HS to report the current microphone gain level setting to the HSAG. <gain> is a decimal numeric constant, relating to a particular (implementation-dependent) volume level controlled by the HS	<gain>: 0-15
Speaker gain level indication report	+VGS=<gain>	Command issued by the HS to report the current speaker gain level setting to the HSAG. <gain> is a decimal numeric constant, relating to a particular (implementation-dependent) volume level controlled by the HS	<gain>: 0-15
Headset button press	+CKPD=200	Command issued by the HS to indicate that the button has been pressed.	

6.4.2.2 Required Result Codes

Table 6-304. Required result codes for Headset implementations

Name	Description
OK	Acknowledges execution of a command.
RING	The DCE has detected an incoming call signal from the network.
ERROR	Error.

7.0 Bibliography

[1] Texas Instruments: LMX9830 Datasheet

[2] Bluetooth SIG: Specification of the Bluetooth System 1.2, November 05 2003

[3] Bluetooth SIG: Specification of the Bluetooth System 1.1, Volume 2 / Profiles, Version 1.1, February 22 2001

[4] Bluetooth SIG: Bluetooth Assigned Numbers, https://www.bluetooth.org/foundry/assignnumb/document/assigned_numbers

8.0 Revision History

This is a report of the revision/creation process of the LMX9830 - Software Users Guide. Any revisions (i.e., additions, deletions, parameter corrections, etc.) are recorded in the table(s) below.

Revision # (PDF Date)	Revisions / Comments
1.0 (08/09/05)	Initial release.
1.1 (10/17/05)	Updated Introduction on Page 1. Updated Section 4.4.1.3 “Wake up functionality” on page 62 for RTS/CTS levels, adapted to “Real voltage levels”. Updated Table 6-279 "Write ROM Patch Request" on page 191
1.2 (02/08/06)	Changed Name Length to 40 Bytes supported instead of 30 Bytes. Fixed UART parity settings (Odd and Even inverted) in NVS and command.
1.3 (02/01/07)	Corrected: - SDAP_ATTRIBUTE_REQUEST - SDAP_SERVICE_REQUEST length field (2 to 1) - SET_DEFAULT_LINK_POLICY - RF_TEST payload size (13 to 14) - SET_DEFAULT_AUDIO_SETTING (PCM slave config) Removed MEM_WRITE, MEM_READ Opcodes. Corrected BD adress oversight (not included in LMX9830)
1.4 (02/26/13)	Corrected: - Power Management - Example description and figure 4-44 on page 72 - Serial bits numbering in figures 4-47 and 4-48

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as “components”) are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers’ products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers’ products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265