# LMX9820 Bluetooth Serial Port Module - Software Users Guide

National Semiconductor
Software Users Guide, **Confidential**
September 2003
Revision 1.3

LMX9820 Bluetooth Serial Port Module - Software Users Guide

## Introduction

The National Semiconductor® LMX9820 Bluetooth Serial Port Module is a highly integrated radio, baseband, and memory implemented on an LTCC (Low Temperature Co-Fired Ceramic) substrate. 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 fullfill service requests for additional profiles on the host.

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 SPP full bluetooth operation.

This document is a reference for implementing the LMX9820 module into a system. Included are a first introduction for bluetooth technology and a detailed description to of the usage models for the bluetooth module. A getting started session gives a very detailed entry point for starting development. Finally all commands and features are listed and explained.

This document is based on:

**Table 0-1.**

| Item | Version |
|---|---|
| Hardware | LMX9820V5.2 |
| Firmware | V5.05 and later |
| SimplyBlueCommander | 1.2.0.1 |

**IMPORTANT**:

LMX9820 products delivered with software versions earlier than v5.07 require initialization to factory default settings (see Section 3.1.1.4 “Restore Factory Settings” on page 22).

Afterwards the parameters in flash are programmed for proper functionality.

This procedure only has to be done once before first usage of the device.

## Table of Contents

# 1.0 LMX9820 Features

## 1.1 SYSTEM OVERVIEW

The LMX9820 is intended to be an add-on module to an existing microcontroller. This means 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 LMX9820 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 offers together with those protocol layers the following profiles:

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

The application is managing all profile related interactions to the stack but also offers a simplified command interface over the UART. The interface is used for configuration of 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.

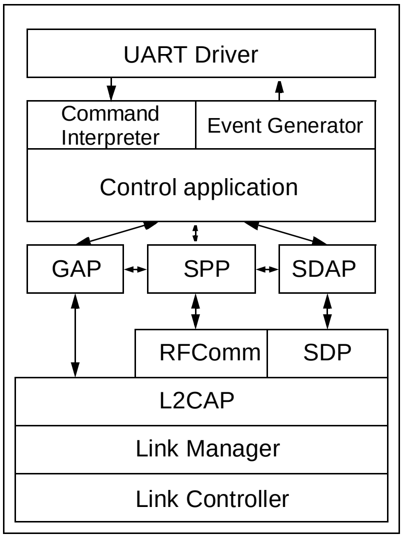


**Figure 1-1. LMX9820 Firmware Implementation**

## 1.2 SYSTEM PARAMETERS

The LMX9820 includes a Non-volatile memory area to store system and bluetooth parameters. The parameters define the chip behaviour during operation but also after hardware or software reset. The values are configured by special commands through the UART interface. Please see Table 1-1 for a complete list of parameters.

**Table 1-1. LMX9820 System Parameters**

| No. | Parameter | Default Value | Description | Checked |
|---|---|---|---|---|
| 1 | BD_Addr | <preprogrammed> | Bluetooth Device Address. On default preprogrammed by National. | Runtime |
| 2 | Device Name | Serial Port Device | Friendly Name of the Bluetooth Device | Runtime |
| 3 | Link Keys | <empty> | Link Keys for paired devices | Runtime |
| 4 | Pin Code | 0000 | Fixed PinCode used for pairing with other devices | Runtime |
| 5 | Class of Device | 000000 | The ’Class of Device’ describes general functionalty of the Bluetooth Device and is transmitted during the Inquiry process. | Runtime |
| 6 | SDP Records | 1 SPP Entry, Auth and Encr. Enabled | The Service Discovery Database of the Device. The database can be configured via Configuration Commands. On default one SPP port configured. | Runtime |
| 7 | SPP Ports to Open | 0000 0001 | Defines the RFCOMM channels to open. For each channel one RFCOMM instance will be created. | Runtime |
| 8 | Default Connections | <empty> | Database for devices to be connected during boot-up or by sending a command. | Runtime |
| 9 | Preferred Master | False | Preferred Master forces the device to switch to Master Role after beeing connected. The device will reject the link if command could not be executed. | Boot-up |
| 10 | UART speed | 9600 | Speed of the Hardware UART Interface | Boot-up |

Table 1-1. LMX9820 System Parameters

| No. | Parameter | Default Value | Description | Checked |
|---|---|---|---|---|
| 11 | UART Parity | None | Parity Setting for the Hardware UART Interface | Boot-up |
| 12 | UARTNoofStopbits | 1 | Number of Stop Bits used on Hardware UART Interface | Boot-up |
| 13 | Operation Mode | Automatic | Configures the general behaviour of the device. Please see Section 1.6 "Operation Modes" for details. | Boot-up |
| 14 | Inquiry Scan Mode | General Inquriy | Sets the discoverability mode of the device. | Runtime |
| 15 | Page Scan Mode | Enabled | Sets the connectablity of the device. (enabled/disabled) | Runtime |
| 16 | Security Mode | 2 | Configures Service Level Security Mode. | Runtime |
| 17 | Event Filter | 1 | Configures the level of events reported to the host | Runtime |
| 18 | Default Link Policy | 000F | Configures the default link policy for incoming and outgoing links. | Runtime |

## 1.3 UART TRANSPARENT MODE

On default the LMX9820 is listening on the UART to a special package format described in chapter 4. Any incoming data will be analyzed and data or parameters will be extracted.

In case the LMX9820 has established a link to only one remote device and no configuration commands have to be sent to the LMX9820, 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.

.

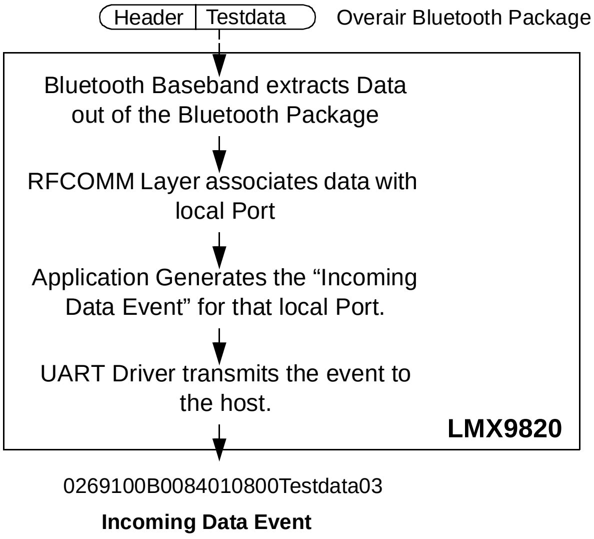

Header
Testdata
Overair Bluetooth Package
Bluetooth Baseband extracts Data out of the Bluetooth Package
RFCOMM Layer associates data with local Port
Application Generates the “Incoming Data Event” for that local Port.
UART Driver transmits the event to the host.
LMX9820
0269100B0084010800Testdata03
Incoming Data Event

Figure 1-2. Receiving data in command mode

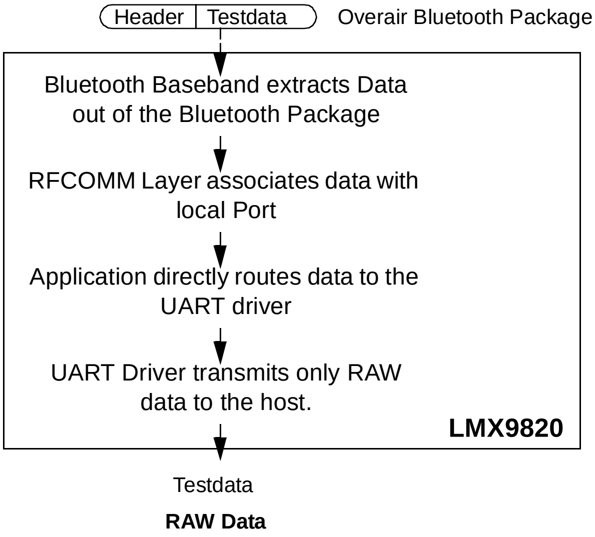

Header
Testdata
Overair Bluetooth Package
Bluetooth Baseband extracts Data out of the Bluetooth Package
RFCOMM Layer associates data with local Port
Application directly routes data to the UART driver
UART Driver transmits only RAW data to the host.
LMX9820
Testdata
RAW Data

Figure 1-3. Receiving Data in Transparent Mode

**Leaving transparent mode:**

As the LMX9820 does not listen to commands, UART Break has to be used to tell the device to leave the transparent mode.

## 1.4 DEFAULT CONNECTIONS

The LMX9820 is able to store up to 3 connections into its own database. The parameters stored within the device are:

- Index
- local RFCommPort
- BD_Addr
- Remote COMPort
- Transparent Mode (yes/no)

Each connection can be addressed by its index. The connection to those devices can be established by the “Connect Default Connection” command.

If the device is in Automatic mode (parameter 13 in Table 1-1), the device will try to connect to those devices after boot-up or reset.

## 1.5 THE COMMAND INTERFACE

The LMX9820 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.

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 will be reported by indication events or requests.

Please refer to Section 4.0 for a complete list of commands and their usage

## 1.6 OPERATION MODES

The operation of the LMX9820 can be devided into different modes. Each mode represents special situations and describes the behaviour of the module.

The following modes can be defined for the LMX9820:

- Automatic Idle
- Command Idle
- Command Mode Master
- Command Mode Slave
- Transparent Mode

The definition for the operation mode “automatic or command” mode is determined out of the parameter “operation mode” in the NVS (see Table 1-1) and has influence on the behaviour of the LMX9820 in different sitiuations.

Figure 1-4 on page 9 describes the behaviour of the LMX9820 after boot-up or reset.
Figure 1-5 on page 10 and Figure 1-6 show the operation flow, if the LMX9820 is contacted by a remote device either in automatic or in command operation.
Figure 1-7 finally shows the operation as master using the command interface. For this operation there’s no difference if the device is configured to automatic or command mode.

### 1.6.1 Automatic/Command Idle

On default, e.g. after boot-up with no connection established, the LMX9820 stays within one of the Idle Modes. This means the device is is visible for other devices, connectable and is answering to service requests with the service database stored within the NVS.

### 1.6.2 Command Master Mode

The command interface can be used to Inquire, doing service requests on the remote device and establishing connections.
If the link is establish by the LMX9820 commands, it will be in the “Command Master” Mode. This means the device acts as master within the link, beeing able to build up a piconet with other devices. Depending on the parameters for Scanmode stored in NVS, the device still listens to requests from outside.

### 1.6.3 Automatic/Command Slave Mode

If connected by another device, the LMX9820 will ask for authentication and if necessary pairing will be processed automatically using the stored Pincode.
After connection setup the LMX9820 indicates the link establishment by sending the event “Incoming Link Established” (see Section 4.2.3.4) and by setting pin LSTAT1 to 0.

Depending on the Operation Mode (Parameter #13 in Table 1-1) the module will switch to transparent mode or not:

- Operation Mode Automatic:
  LMX9820 automatically switches to Transparent Mode on UART
- Operation Mode Command:
  LMX9820 still tries to interpret the data on the UART as commands

If the link is lost, the module will indicate the lost link by setting LSTAT1 to 1 and sending a Link Released indicator to the host. In case the UART was in transparent mode, an UART BREAK signal will be sent first.

### 1.6.4 Transparent Mode

If the LMX9820 established a link to only one remote device, it can be switched to UART transparent mode. (see also "UART Transparent Mode" on page 6).

In this mode the LMX9820 will not listen to commands anymore. The device will also not respond to inquiry or connection requests from remote devices..

**Table 1-2. Operation Mode Features**

| Parameter | Automatic Idle | Command Idle | Command Master | Command Slave | Transparent |
|---|---|---|---|---|---|
| P. #13 in NVS | Automatic | Command | both | both | both |
| Discoverable[1] | yes | yes | yes | no | no |
| Connectable[1] | yes | yes | yes | no | no |
| Master/Slave | - | - | master | slave | both |
| Search for devices | yes | yes | yes | yes[2] | no |

Table 1-2. Operation Mode Features

| Parameter | Automatic Idle | Command Idle | Command Master | Command Slave | Transparent |
|---|---|---|---|---|---|
| Connect to remote devices (actively) | yes | yes | yes | yes[2] | no |
| Send Raw Data | - | - | no | no | yes |
| Multipoint connections | - | - | yes | yes[2] | no |

1. Depending on the setting in Non-volatile storage (default setting assumed)
2. Only if the active link is switched to on e of the Low Power Modes (Sniffmode, Holdmode or Parkmode)

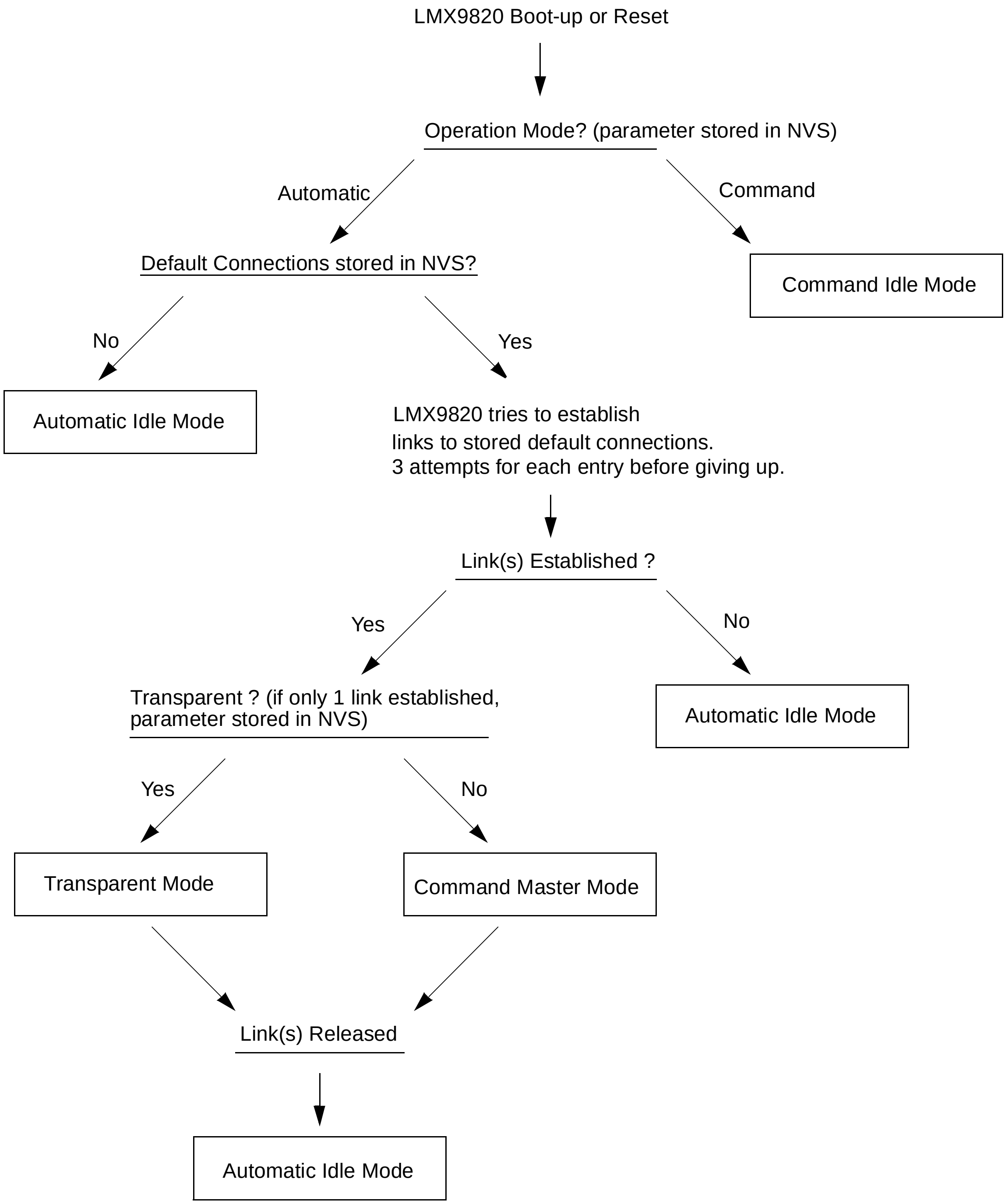


**Figure 1-4. Operation Flow after boot-up or Reset**

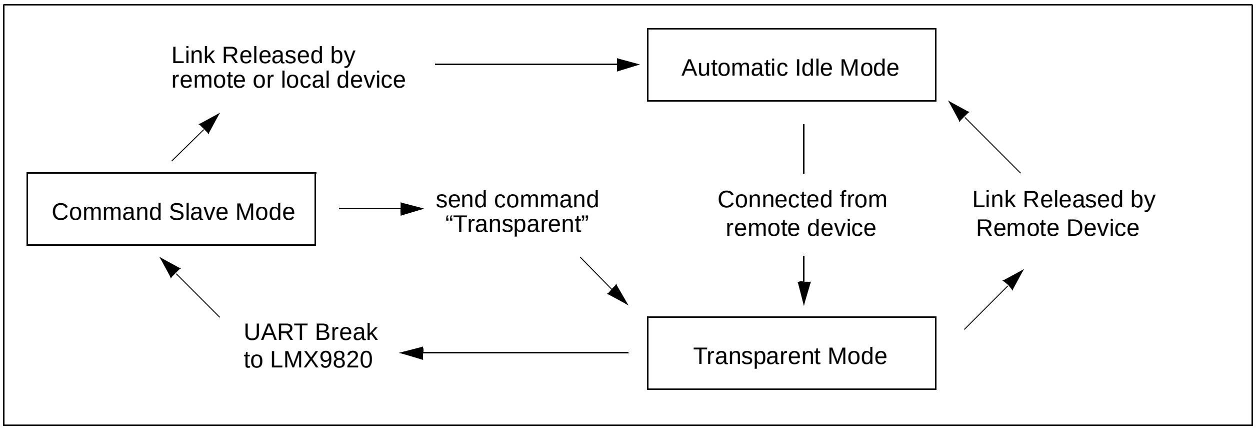


**Figure 1-5. Automatic Slave Operation Flow**

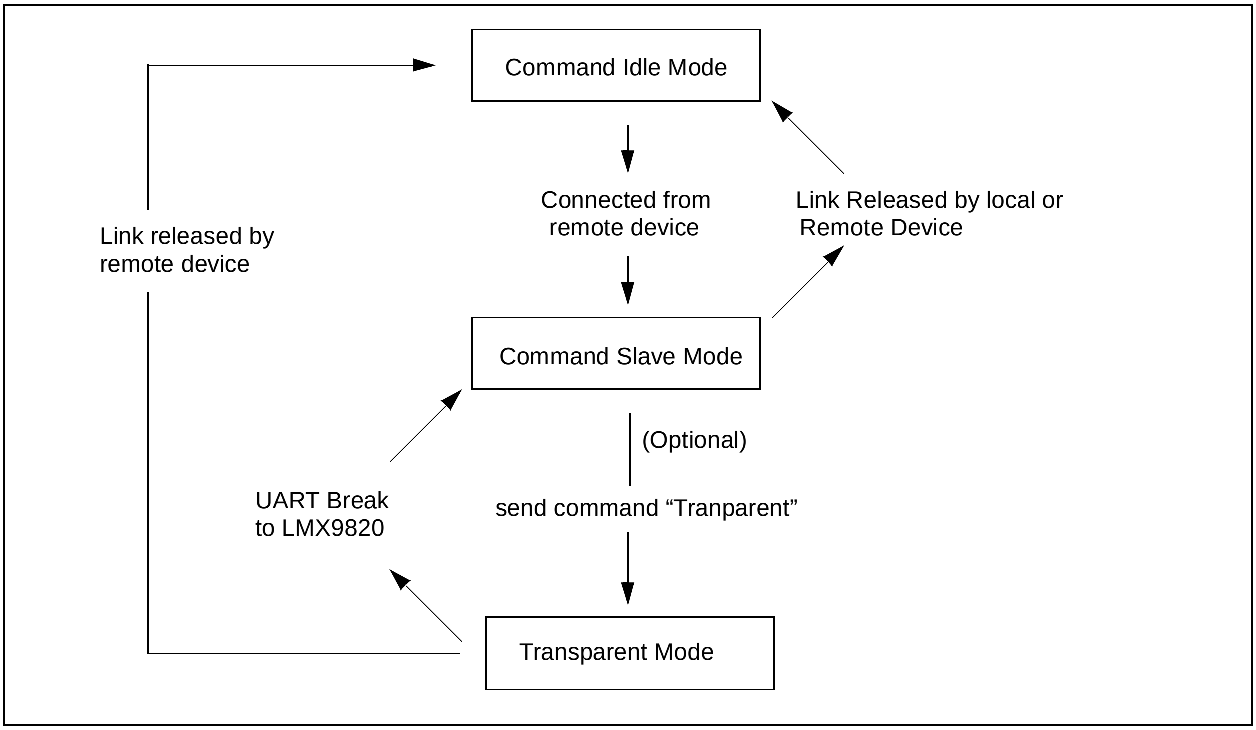


**Figure 1-6. Command Slave Operation Flow**

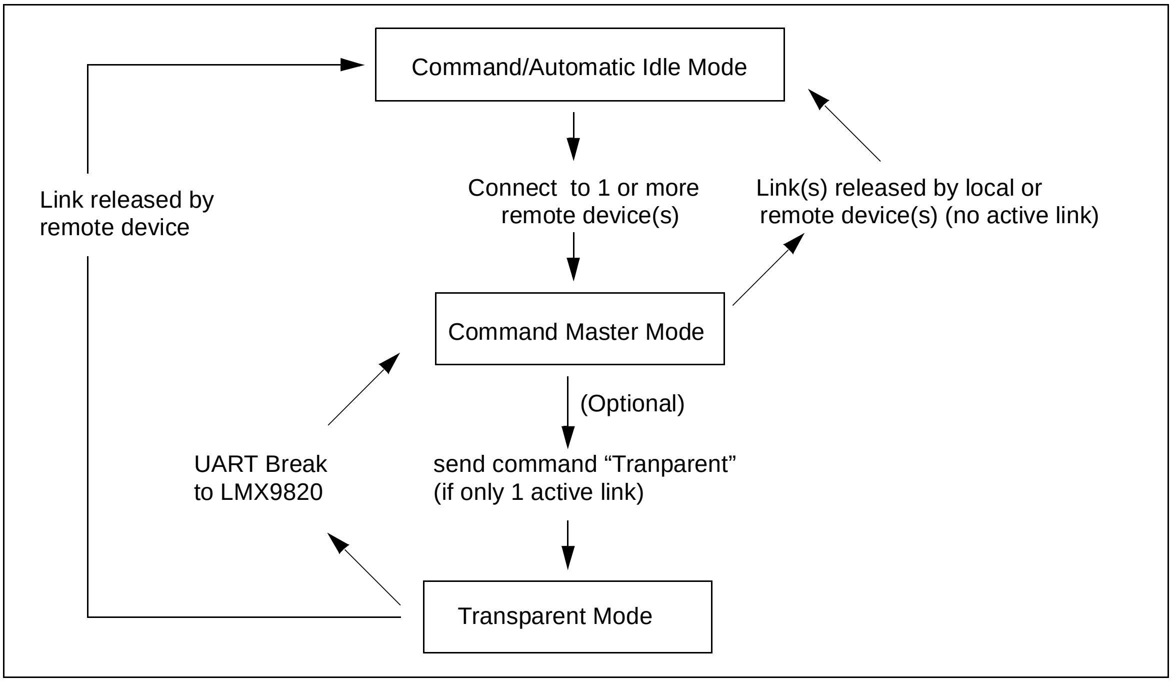

Command/Automatic Idle Mode
Link released by remote device
Connect to 1 or more remote device(s)
Link(s) released by local or remote device(s) (no active link)
Command Master Mode
(Optional)
UART Break to LMX9820
send command “Tranparent” (if only 1 active link)
Transparent Mode

**Figure 1-7. Command Master Operation Flow**

## 2.0 Getting Started

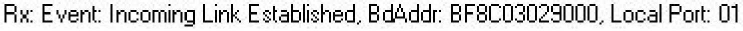
This description is using Simply Blue Commander Log Entries to show the command structure and their results.

The log file is adding new commands on top of the screen, so all log file pictures show the first entry at the bottom of the picture, the last entry at the top.

### 2.1 USING THE AUTOMATIC SLAVE OPERATION

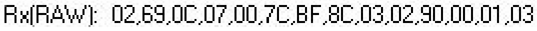
As described in Section 1.6 the LMX9820 on default is in a waiting mode (Automatic idle mode) after boot-up or reset. This means it is waiting for requests and automatically is answering to connection requests.

If connected from a remote device, the LMX9820 establishes automatically a SPP link and indicates the established link to host by the “Link Established Event” and by pulling down the LSTAT1 pin.



```
Rx: Event: Incoming Link Established, BdAddr: BF8C03029000, Local Port: 01
```

**Figure 2-1. Log of Incoming Link Established**



```
Rx(RAW):  02,69,0C,07,00,7C,BF,8C,03,02,90,00,01,03
```

**Figure 2-2. Hex Log of Incoming Link Established**

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

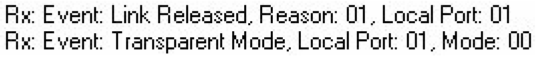
**Table 2-1. Example Incoming Link Established**

| 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 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.

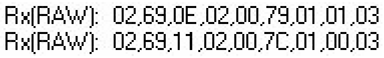
If the link is dropped, the LMX9820 will empty its buffers and

- send a UART break to the host,
- send “Transparent Mode Indicator” (Section 4.2.4.1)
  — Indicates on protocol level to the host that transparent mode has been left.
- send “Link Released Event” (Section 4.2.3.3)
  — Indicates that link has been released.
- LSTAT1 pin will be set back to 1



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

**Figure 2-3. Log of Link Released Event**



```
Rx(RAW):  02,69,0E,02,00,79,01,01,03
Rx(RAW):  02,69,11,02,00,7C,01,00,03
```

**Figure 2-4. Hex Log of Link Released Event**

The events “transparent mode” and “link released” indicate within their package the local RFComm port and the current mode respectivly the reason of releasing the link.

Please see also Section 4.1 for a complete description of the package and header format.

**Table 2-2. Example Transparent Mode Lost**

| 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 2-3. Example Link Released**

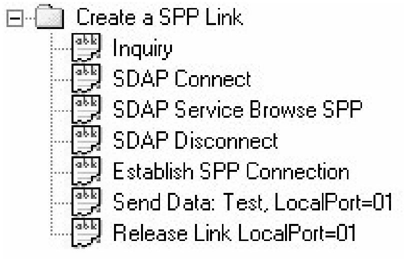
| Byte | Parameter | Value |
|---|---|---|
| 01 | Start Delimiter | 02 |
| 02 - 06 | Package Header | 69 0E 02 00 79 |
| 07 | Reason byte | 01 (Remote device disconn.) |
| 08 | Local Port | 01 |
| 09 | End Delimiter | 03 |

## 2.2 SETTING UP A LINK USING THE COMMAND INTERFACE

Setting up a bluetooth link between devices requires that the devices know specifc parameters of each other. To get those parameters several steps have to be processed before a real 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 to fully understand the syntax and logic of the command interface.



```
Create a SPP Link
  Inquiry
  SDAP Connect
  SDAP Service Browse SPP
  SDAP Disconnect
  Establish SPP Connection
  Send Data: Test, LocalPort=01
  Release Link LocalPort=01
```

**Figure 2-5. CreateSPPLink.dir**

### 2.2.1 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.

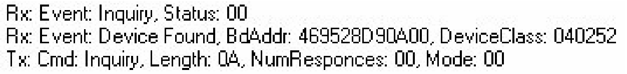
With LMX9820 this process can be startet with the command “Inquiry” (Section 4.2.1.1).

The command results in two different events:

- Device Found Indicator
- Inquiry Complete Confirmation

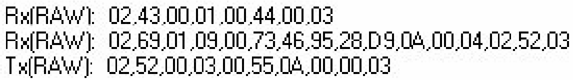
Each found device will be indicated by the Device Found Indicator, including its BD_Addr and Class of Device.

The Inquiry Complete indicates the end of the Inquiry process. Figure 2-6 shows the log as interpreted by the Simply Blue Commander and Figure 2-7 the actual hex traffic on the UART.



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

**Figure 2-6. Interpreted Log of an Inquiry**



```
Rx(RAW):  02,43,00,01,00,44,00,03
Rx(RAW):  02,69,01,09,00,73,46,95,28,D9,0A,00,04,02,52,03
Tx(RAW):  02,52,00,03,00,55,0A,00,00,03
```

**Figure 2-7. Hex Log of an Inquiry**

Figure 2-7 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 LMX9820.

The following paragraphs explain the package format and usage in more detail. The complete package format is also described in Section 4.1.

**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

- Startdelimiter (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 **startdelimiter** is always 0x02.

The **packet type id** for a request is 0x52. (see Section 4.1.3 for the complete list)

The **Opcode** for Inquiry is 00 (see Section 4.1.4 for the complete list of opcodes)

The **payload length** indicates literally 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 2-4. 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 enddelimiter 0x03.

**b.) The Device Found Indicator**

The first response to the inquiry command from the LMX9820 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:**

- Startdelimiter - 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 2-5. 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 LMX9820 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 occured, error 0x00 will be returned. All other values have a specific reason. Please see Table 4-155 "Generic Error Codes" on page 89 for a complete list of error codes.

The confirmation in hex:

02 43 00 01 00 44 00 03

Package header:

- Startdelimiter - 0x02
- Packet Type - Indicator: 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 always consists only of the status byte. In this case 0x00.

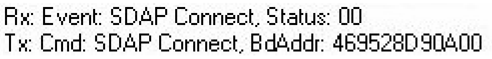
**Table 2-6. Example Inquiry Confirm Package**

| 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 |

### 2.2.2 Create SDAP Connection

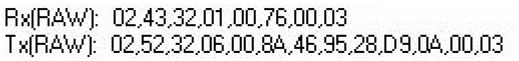
To create a SPP connection to another device, the local RFComm channel has to know which remote RFComm Channel to address. Each service is registered to a specific RFComm channel number. To get this number the local device has to do a Service Request on the remote device and get the service entry.

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 2-8. Log of the Create SDAP Command**



```
Rx(RAW):  02,43,32,01,00,76,00,03
Tx(RAW):  02,52,32,06,00,8A,46,95,28,D9,0A,00,03
```

**Figure 2-9. Hexadecimal Log of the Create SDAP Command.**

**Table 2-7. 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)

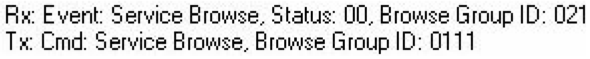
The command is confirmed by the LMX9820 with the appropriate confirmation event. If status is 0x00 the link has been established.

### 2.2.3 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.

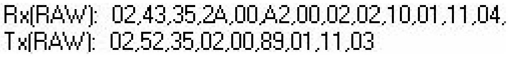
As any multi-byte parameter the UUID has to be sent byte swapped to the LMX9820 within the command.

.



```
Rx: Event: Service Browse, Status: 00, Browse Group ID: 021
Tx: Cmd: Service Browse, Browse Group ID: 0111
```

**Figure 2-10. Log SDAP Browse for SPP**



```
Rx(RAW):  02,43,35,2A,00,A2,00,02,02,10,01,11,04,
Tx(RAW):  02,52,35,02,00,89,01,11,03
```

**Figure 2-11. Hex Log of SDAP Browse for SPP**

Opcode - 0x35

Parameters:

- UUID for the requested Service: 0x1101

**Table 2-8. Example SDAP Browse**

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

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

The full event includes back 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, should be the service requested)
- RFComm Port Number - 0x04
- Number of bytes in the service name
- Name of the service

The following table shows the full confirm package for two SPP entries.

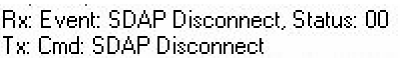
**Table 2-9. Example SDAP Browse Confirm**

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

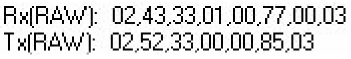
### 2.2.4 SDAP Disconnect

After a successfull Service Browse the connection has to be released again.



```
Rx: Event: SDAP Disconnect, Status: 00
Tx: Cmd: SDAP Disconnect
```

**Figure 2-12. Log of SDAP Disconnect**



```
Rx(RAW):  02,43,33,01,00,77,00,03
Tx(RAW):  02,52,33,00,00,85,03
```

**Figure 2-13. Hex Log of SDAP Disconnect**

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

### 2.2.5 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).

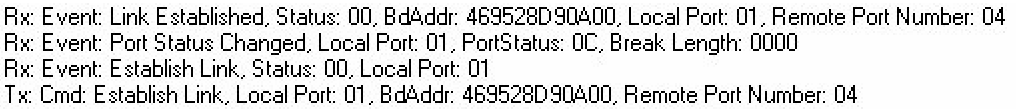
The following parameters are needed to establish a SPP link to a remote device.

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

see also Figure 2-15 for the SPP Command (TX).

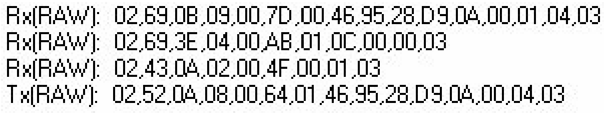
**Table 2-10. Example Create SPP Link**

| Byte | Parameter | Value |
|---|---|---|
| 01 | Start Delimiter | 02 |
| 02 - 06 | Package Header | 52 0A 08 00 64 |
| 07 | Local COM Port | 01 (default at LMX9820) |
| 08 - 13 | Remote BD_Addr | 46 95 28 D9 0A 00 |
| 14 | Remote COM 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 2-14. Log of Create SPP Connection**



```
Rx(RAW):  02,69,0B,09,00,7D,00,46,95,28,D9,0A,00,01,04,03
Rx(RAW):  02,69,3E,04,00,AB,01,0C,00,00,03
Rx(RAW):  02,43,0A,02,00,4F,00,01,03
Tx(RAW):  02,52,0A,08,00,64,01,46,95,28,D9,0A,00,04,03
```

**Figure 2-15. Hex Log of Create SPP Connection**

The Log Window shows 3 events returned by the LMX9820.

**a) Establish Link Confirm**
As any confirm the “Establish Link Confirm” (Section 4.2.3.1) has the same Opcode as the command sent. It includes the following parameters:

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

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

**Table 2-11. 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 Port Number | 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 4.2.7.3 for the detailed description of the event.

**c) Link Established Indicator**
To indicate an established link on top of the SPP, the LMX9820 uses the “Link Established Indicator” (Section 4.2.3.4). The event returns

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

**Table 2-12. 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) |

**Table 2-12. Example Link Established Indicator**

| Byte | Parameter | Value |
|---|---|---|
| 08 - 13 | BD_Addr | 46 95 28 0D 0A 00 |
| 14 | Local Port Number | 01 |
| 15 | Remote Port Number | 04 |
| 16 | End delimiter | 03 |

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

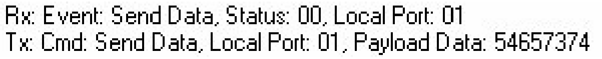
### 2.2.6 Sending Data in Command Mode

After actively setting up a connection with the LMX9820, the device is still listening to commands and returning status changes by events.

So to send data over the command interface the “Send Data” (Section 4.2.3.2) Command has to be used.

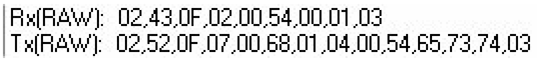
Besides the data which have to be sent, the local RFComm Port parameter has also to be sent to the LMX9820. This enables the application to support multiple connections.

The Log windows in Figure 2-16 and Figure 2-17 show the transmission of the word “Test” over an established SPP link. The data is displayed in hex as ASCII values.



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

**Figure 2-16. Log of sending the Data “Test”**



```
Rx(RAW):  02,43,0F,02,00,54,00,01,03
Tx(RAW):  02,52,0F,07,00,68,01,04,00,54,65,73,74,03
```

**Figure 2-17. Hex log of sending the data “Test”**

**Table 2-13. Example Sending Data Package**

| Byte | Parameter | Value |
|---|---|---|
| 01 | Start Delimiter | 02 |
| 02 - 06 | Package Header | 52 0F 07 00 68 |
| 07 | Local 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 2-14 shows a second example with a longer Datapackage, marking changed parameters in bold.

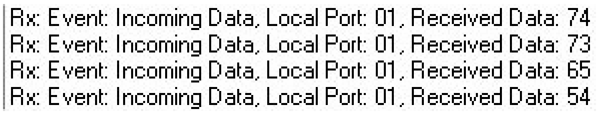
**Table 2-14. Example Sending Data “Testdata”**

| 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 |

### 2.2.7 Receiving Data in Command Mode

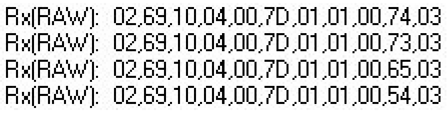
In command mode, incoming data from a remote device are indicated by the “Incoming Data” (Section 4.2.3.2) Event.

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 2-18. Log of Incoming Data Event**



```
Rx(RAW):  02,69,10,04,00,7D,01,01,00,74,03
Rx(RAW):  02,69,10,04,00,7D,01,01,00,73,03
Rx(RAW):  02,69,10,04,00,7D,01,01,00,65,03
Rx(RAW):  02,69,10,04,00,7D,01,01,00,54,03
```

**Figure 2-19. Hex Log of Incoming Data Event**

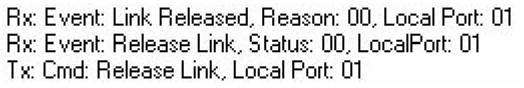
Figure 2-19 and Figure 2-18 show the log of 4 bytes received on local Port 01. The bytes together form the word “Test” again.

**Table 2-15. 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 |

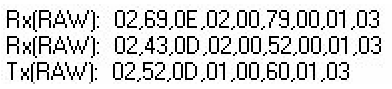
### 2.2.8 Releasing a SPP connection

To release an existing SPP connection the “Release Link” Command (Section 4.2.3.3) 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 2-20. Log for Release Link Command**



```
Rx(RAW):  02,69,0E,02,00,79,00,01,03
Rx(RAW):  02,43,0D,02,00,52,00,01,03
Tx(RAW):  02,52,0D,01,00,60,01,03
```

**Figure 2-21. Hex Log for Release Link Command**

**Table 2-16. 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 |

Table 2-16. Example Release Link Package

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 LMX9820 released the Link on the Com Port returned.

**Table 2-17. 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 |

## 2.3 USING TRANSPARENT MODE

In Transparent Mode, described in Section 1.3, can be sent to the LMX9820 as they are. This means, the LMX9820 does not interpret the packages sent to device. Instead, it is routing the data straight to the link previously set up.

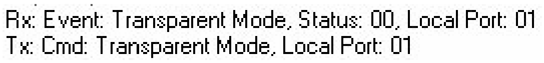
If the LMX9820 was connected from another device, this mode is automatically activated if the Operation Mode “Automatic” is set in NVS (default).

If the link was set up manually via commands (see Section 2.2), the LMX9820 still listens to commands and data have to be sent via the “Send Data” command.

As the LMX9820 routes the data directly to the bluetooth link, transparent mode can only be activated if only one active link exists.

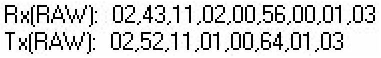
The “Transparent mode” command is referring to the local port the link was created.

The command is confirmed by the appropriate event. Afterwards the LMX9820 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 2-22. Log of Set Transparent Mode**



```
Rx(RAW):  02,43,11,02,00,56,00,01,03
Tx(RAW):  02,52,11,01,00,64,01,03
```

**Figure 2-23. Hex Log of Set Transparent Mode**

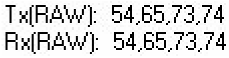
**Table 2-18. Example Set Transparent Mode**

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

**Table 2-19. 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 Port | 01 |
| 09 | End Delimiter | 03 |

Any data received will be routed directly to the UART.



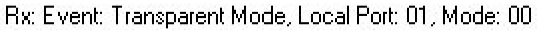
```
Tx(RAW):  54,65,73,74
Rx(RAW):  54,65,73,74
```

**Figure 2-24. Sending/Receiving “Test” in transparent mode**

The LMX9820 is leaving the transparent mode, when a Break signal is sent on the UART.

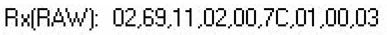
The break signal can also be used if the LMX9820 has been connected from a remote device (Automatic slave transparent) and switched automatically to transparent.

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



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

**Figure 2-25. Log of Transparent Mode Event**



```
Rx(RAW):  02,69,11,02,00,7C,01,00,03
```

**Figure 2-26. Hex Log of Transparent Mode Indicator**

**Table 2-20. Example Transparent Mode Indicator**

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

## 2.4 EXAMPLES

The following log files show the typical hex values sent to respectively returned from the LMX9820. The level of events returned by the 9820 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.

### 2.4.1 Automatic Slave

Table 2-21 shows the event returned from the LMX9820 if it was connected from outside. The LMX9820 just returns one event indicating the BD_Addr of the remote device and the local port it connected to.

**Table 2-21. Log File of Incoming Link as automatic slave**

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

Table 2-22 shows the events of typical procedure if a link was released from the other device.

**Table 2-22. Log File of a Released Link as Automatic Slave**

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

### 2.4.2 Setting up a link

As documented in Section 2.2, 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 2-23 shows all commands necessary from scratch to establish a link to another device.

**Table 2-23. Log File of a complete link setup**

| Filter | Direction | What | Hex Code | Interpreted by Simply Blue Commander |
|---|---|---|---|---|
| 01 / 02 | TX | Request | 02,52,00,03,00,55,0A,00,00,03 | Inquiry, Length: 0A, NumResponces: 00, Mode: 00 |
| 01 / 02 | RX | Indicator | 02,69,01,09,00,73,12,34,56,78,9A,BC,00,00,00,03 | Device Found, BdAddr: 123456789ABC, DeviceClass: 000000 |
| 01 / 02 | RX | Event | 02,43,00,01,00,44,00,03 | Inquiry, Status: 00 |
| 01 / 02 | TX | Request | 02,52,32,06,00,8A,12,34,56,78,9A,BC,03 | SDAP Connect, BdAddr: 123456789ABC |
| 02 | RX | Indicator | 02,69,50,07,00,C0,12,34,56,78,9A,BC,00,03 | ACL Established, BdAddr: 123456789ABC, Status: 00 |
| 01 / 02 | RX | Event | 02,43,32,01,00,76,00,03 | SDAP Connect, Status: 00 |
| 01 / 02 | TX | Request | 02,52,35,02,00,89,01,11,03 | Service Browse, Browse Group ID: 0111 |
| 01 / 02 | RX | Event | 02,43,35,0D,00,85,00,01,02,10,01,11,01,05,43,4F,4D,31,00,03 | Service Browse, Status: 00, Browse Group ID: 0210, Service ID: 0111, PortNo: 01, Service Name: COM1. |
| 01 / 02 | TX | Request | 02,52,33,00,00,85,03 | SDAP Disconnect |

Table 2-23.  Log File of a complete link setup

| Filter | Direction | What | Hex Code | Interpreted by Simply Blue Commander |
|---|---|---|---|---|
| 02 | RX | Indicator | 02,69,51,07,00,C1,12,34,56,78, 9A,BC,16,03 | ACL Terminated, BdAddr: 123456789ABC, Reason: 16 |
| 01 / 02 | RX | Event | 02,43,33,01,00,77,00,03 | SDAP Disconnect, Status: 00 |
| 01 / 02 | TX | Request | 02,52,0A,08,00,64,01,12,34,56, 78,9A,BC,01,03 | Establish Link, Local Port: 01, BdAddr: 123456789ABC, Remote Port Number: 01 |
| 01 / 02 | RX | Event | 02,43,0A,02,00,4F,00,01,03 | Establish Link, Status: 00, Local Port: 01 |
| 02 | RX | Indicator | 02,69,50,07,00,C0,12,34,56,78, 9A,BC,00,03 | ACL Established, BdAddr: 123456789ABC, Status: 00 |
| 01 / 02 | RX | Indicator | 02,69,3E,04,00,AB,01,0C,00,00 ,03 | Port Status Changed, Local Port: 01, PortStatus: 0C, Break Length: 0000 |
| 01 / 02 | RX | Indicator | 02,69,0B,09,00,7D,00,12,34,56, 78,9A,BC,01,01,03 | Link Established, Status: 00, BdAddr: 123456789ABC, Local Port: 01, Remote Port Number: 01 |
| 01 / 02 | TX | Request | 02,52,11,01,00,64,01,03 | Transparent Mode, Local Port: 01 |
| 01 / 02 | RX | Event | 02,43,11,02,00,56,00,01,03 | Transparent Mode, Status: 00, Local Port: 01 |

# 3.0 Advanced Usage

The LMX9820 offers a wide variety of functions for different usage models. This section should point out the most important features and scenarious covered by the LMX9820.

## 3.1 LOCAL CONFIGURATION

### 3.1.1 Hardware Configuration

The LMX9820 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 4.2.16 “Local Hardware Configuration” on page 82.

#### 3.1.1.1 Change UART settings

The UART speed in general is determined by the choice of the ISEL pins of the LMX9820. The pins and UART settings in NVS are only checked during the software boot-up process so also after a Reset command.

**Table 3-1. UART settings**

| ISEL1 | ISEL2 | Interface Speed (baud) | UART Settings |
|---|---|---|---|
| 1 | 1 | 921.6k | Check NVS |
| 0 | 1 | 115.2k | Check NVS |
| 1 | 0 | 9.6k | No Parity,<br>One Stop bit |
| 0 | 0 | Check NVS | Check NVS |

#### 3.1.1.2 Bluetooth Testmode

The LMX9820 supports the standard Bluetooth “Device Under Test” Mode and a “Local Loopback” Mode.

Those 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 4.2.16.8 “Bluetooth Test Mode” on page 85 for details for the command.

#### 3.1.1.3 RF Testmodes

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

The detailed command is described in Section 4.2.16.9 “Initiate RF Test Mode” on page 86.

#### 3.1.1.4 Restore Factory Settings

The LMX9820 is delivered with standard settings in NVS which can be seen in Table 1-1 "LMX9820 System Parameters" on page 5. 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 changes will be made.

- Reset of the Service Database entry to the default
- Deleting Link Keys (all paired devices)
- UART Speed Settings
- Event Filter
- Ports to open
- Master Role off
- Operation Mode
- Pin
- Local Name
- Default Link Policy

Command details can be found in Section 4.2.16.5 “Restore Factory Settings” on page 83.

#### 3.1.1.5 Read RSSI

After link establishment the radio is measuring the “Receive Signal Strenght 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 4.2.16.10 on page 87).

The value indicates:

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

#### 3.1.1.6 Start Firmware Upgrade

The LMX9820 includes an “In-System-Programming” (ISP) code which allows to update the on-chip firmware over the UART interface. The code is activated by either the “Environment Pins” ENV or by sending this command to the chip. The command will initiate a software Jump to the ISP code, which can only be left by a hardware reset. In ISP no bluetooth functionality will be provided.

Commands available:

- “Firmware Upgrade” on page 88

NOTE: The Command “Firmware Upgrade” sets the empty flag on flash memory. Therefore it is necessary to reprogram the flash and to reset the this flag in ISP mode, other wise the module will stay in ISP mode after boot-up.

#### 3.1.1.7 Event Filter

The LMX9820 indicates status changes and confirms commands with specific events. To address specifc application requirements the level of reporting can be increased or decreased.

The following separation is possible:

- Expected Command confirmations (CFM)
- Indicators for status changes or incoming data
- ACL indicators, reporting any incoming or outgoing physical bluetooth link.

The filtering can either suppress no event, only the ACL indicators or all events.

##### 3.1.1.7.1 Report all events

In case the NVS setting is set to “report all events”, the LMX9820 reports any status change and ACL link establishment to the host.

The main difference to the other fitler 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 LMX9820 will be reported to the host. For a “slave only” usage, one of the other filtering settings would propably be more useful.

Available ACL Events:

- “ACL Established” on page 77
- “ACL Terminated” on page 78

##### 3.1.1.7.2 Report all events except ACL events

In this reporting scheme the LMX9820 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.

##### 3.1.1.7.3 Report no events

In case the UART should not send back any event to the host the Event filter has to be set to “Report no events”.

This filter setting is useful if the LMX9820 is used as cable replacement in front of a microcontroller, where no status event can be interpreted.

In this setting the LSTAT pin is the only indicator for the status of the LMX9820.

### 3.1.2 LMX9820 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.

#### 3.1.2.1 Local Bluetooth Device Address

The BD_Addr is a unique identifier for each bluetooth product. The LMX9820 is delivered by National with a preprogrammed BD_Addr which is stored in a write protected area in the flash and copied to NVS if this is empty.

If necessary this value can be changed to any specific value. A custom value will be reset by the “Restore Factory settings” command.

The commands available are:

- “Read Local Bluetooth Address” on page 73
- “Change Local Bluetooth Address” on page 74

#### 3.1.2.2 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” on page 72
- “Write Local Name” on page 73

#### 3.1.2.3 Class of Device

The Class of Device is based on a numbering scheme of the Bluetooth SIG and is returned on and Inquriy request 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 User Interface 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.

Some Examples:

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

Note: the values should just be seen as examples.

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

Commands available:

- “Store Class of Device” on page 74

#### 3.1.2.4 Operation Mode

The Operation Mode as described in Section 1.6 “Operation Modes” on page 7 has influence on the behaviour of the LMX9820 in different situations. Please check the previous section for details.

Commands available:

- “Read Operation Mode” on page 76
- “Write Operation Mode” on page 77

#### 3.1.2.5 Fixed Pin

The LMX9820 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” on page 81
- “Set Fixed Pin” on page 81

The pin stores the hex value for the ASCII character used as pin. Eg. 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”.

## 3.2 DEFAULT CONNECTIONS

The LMX9820 offers a default connection procedure which allows to store several 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 successfull or failed link establishment will be confirmed for each stored device.

If set to Operation Mode “Automatic” the LMX9820 will try to connect to each of those devices 3 times then switch to the next. (see Section 1.6 "Operation Modes")

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

Commands available:

- “Store Default Connection” on page 46
- “Connect to Default Connection” on page 46
- “Get List of Default Connections” on page 47
- “Delete Default Connection” on page 48

NOTE:

For multiple connections please make sure that the NVS parameter “Ports to open” (See “Set Ports To Open” on page 63) is configured correctly to have the appropriate number of RFComm instances initialised.

Eg. For 2 connections set “ports to open” to 0x00000003,
for 3 connections set “ports to open” to 0x00000007.

## 3.3 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.
  - Datatransmissions 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 synchronised by “Beacons”
  - Slave has be unparked before data can be transmitted between devices again.

The LMX9820 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 specifc link.

### 3.3.1 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

#### 3.3.1.1 Default Link Policy Setting

The LMX9820 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” on page 48
- “Get Default Link Policy” on page 49

#### 3.3.1.2 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 LMX9820 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” on page 49
- “Get Link Policy” on page 50

### 3.3.2 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 continously to the master. This causes high power consumption on the slave. By setting a Sniff intervall, 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.
- Minimum Sniff Interval
  — The minimum interval of slots the two devices are not exchanging packages
- Sniff Attempts
  — Number of slots the slave has to listen to incoming packages from the master, beginning at the sniff slot.
- Sniff Timeout
  — Number of slots the slave has to listen even if he still receives packages for itself.

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” on page 53).

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

Commands available:

- “Enter Sniff Mode” on page 51
- “Exit Sniff Mode” on page 51

**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.

So, 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: | 3 slots |
| Sniff timeout: | 1 slot |

This means the slave listens for a minimum of 3 slots for packets from the master. After a received package it will at least listen for one more slot.

### 3.3.3 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” on page 53

Please see Section 3.3.2 "Sniff Mode" for the calculation of the Hold mode parameters.

### 3.3.4 Park Mode

Park Mode enables devices to completely disconnect from each other and just keep synchronised. 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” on page 52
- “Exit Park Mode” on page 52

# 4.0 LMX9820 Command Interface

## 4.1 UART PROTOCOL PRINCIPLES

The LMX9820 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.

### 4.1.1 Framing

The connection is considered "Error free". But for packet recognition and synchronisation, some framing is used.
All packets sent in both directions are constructed after the following model:

**Table 4-1. 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 ---------------\| | | | | | |

### 4.1.2 Start delimiter

The start delimiter indicates the LMX9820 the beginning of a new package. The "STX" char is used as start delimiter.

STX = 0x02

### 4.1.3 Packet type identification

This byte identifies the type of packet. The following types are valid:

**Table 4-2. Packet Type Identification**

| Code | Packet Type | Description |
|---|---|---|
| 0x52<br>'R' | Request<br>(REQ) | A request sent to the Bluetooth module.<br>All request are answered by exactly one confirm. |
| 0x43<br>'C' | Confirm<br>(CFM) | The Bluetooth modules confirm to a request.<br>All request are answered by exactly one confirm. |
| 0x69<br>'i' | Indication<br>(IND) | Information sent from the Bluetooth module, that is not a direct confirm to a request. |
| 0x72<br>'r' | Response<br>(RES) | An optional response to an indication.<br>This is used to respond to some type of indication messaged. |

All other values are reserved.

### 4.1.4 Opcode

The opcode is a command specifier. Each command is represented by this one byte identifier.

**Table 4-3. 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 4-3. Opcode Values

| | |
|---|---|
| GAP_SET_FIXED_PIN | 0x17 |
| 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 |
| | |
| 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_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 |

**Table 4-3. Opcode Values**

| | |
|---|---|
| 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 |
| | |
| ENABLE_SDP_RECORD | 0x29 |
| DELETE_SDP_RECORDS | 0x2A |
| STORE_SPP_RECORD | 0x2B |
| STORE_DUN_RECORD | 0x2C |
| STORE_FAX_RECORD | 0x2D |
| STORE_OPP_RECORD | 0x2E |
| STORE_FTP_RECORD | 0x2F |
| STORE_SYNC_RECORD | 0x30 |
| | |
| RESET | 0x26 |
| LMX9820_READY | 0x25 |
| TEST_MODE | 0x24 |
| FIRMWARE_UPGRADE | 0x47 |
| READ_RSSI | 0x20 |
| RF_TEST_MODE | 0x4B |

### 4.1.5 Data length

Number of bytes in the "Packet data" area. The maximum size is 333 bytes.

### 4.1.6 Packet data

The data fields hold binary data; hence both 0x02 (=STX) and 0x03 (=ETX) are allowed as data.

### 4.1.7 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.

### 4.1.8 End delimiter

The "ETX" char is used as end delimiter.
ETX = 0x03

### 4.1.9 Retransmission

The connections is considered "Error free", hence no need for implementing timeouts and retransmissions.

### 4.1.10 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.

### 4.1.11 Byte Order

The byte order of the protocol is Little Endian, if nothing else is specified.

## 4.2 COMMAND SET

The LMX9820 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.

### 4.2.1 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”.

#### 4.2.1.1 Inquiry

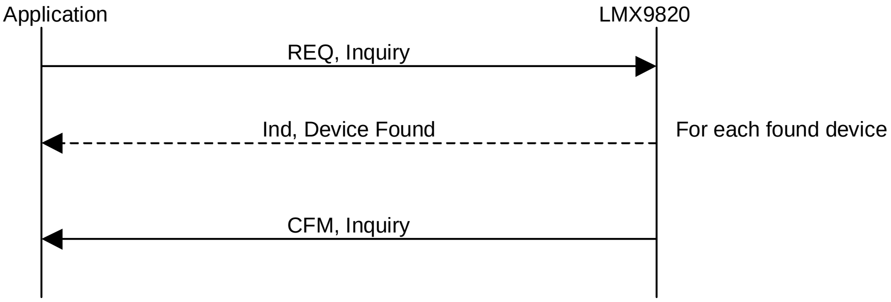


**Figure 4-1. Inquiry Command Flow**

**Table 4-4. Inquiry Command**

| **Description** | **Initiates a search for other Bluetooth devices.** | |
|---|---|---|
| PacketType | REQ | |
| Opcode | GAP_INQUIRY | |
| DataLength | 3 | |
| Data | Length<br>1 byte | Duration of inquiry<br>Range: 0x01 -0x30 (1.28s - 61.44s) |
| | NumResponses<br>1 byte | Maximum number of responses<br>Range: 0x00 - 0xFF<br>0x00 = Unlimited number of responses. |
| | Mode<br>1 Byte | General Inquiry 0x00<br>Limited Inquiry 0x01 |

**Table 4-5. Inquiry Confirm**

| **Description** | **Confirms that the search for other Bluetooth devices is complete.** | |
|---|---|---|
| PacketType | CFM | |
| Opcode | GAP_INQUIRY | |
| DataLength | 1 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_DURATION_OUT_OF_RANGE<br>ERROR_INVALID_MODE<br>ERROR_INVALID_NO_OF_PARAMETERS |

**Table 4-6. Device Found Indicator**

| Description | Indicates that a device has been found. | |
|---|---|---|
| PacketType | IND | |
| Opcode | GAP_DEVICE_FOUND | |
| DataLength | 9 | |
| Data | BdAddr<br>6 bytes | Bluetooth device address of the found device. |
| | DeviceClass<br>3 byte | Class of the found device. |

#### 4.2.1.2 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 seperate command has to be used.

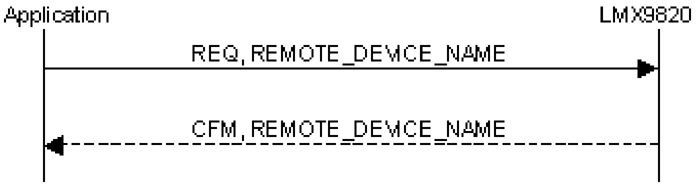


**Figure 4-2. Get Remote Device Name Flow**

**Table 4-7. Get Remote Device Name Command**

| Description | Request the user-friendly name from a known remote Bluetooth device. | |
|---|---|---|
| PacketType | REQ | |
| Opcode | GAP_REMOTE_DEVICE_NAME | |
| DataLength | 6 | |
| Data | BdAddr<br>6 byte | Bluetooth device address for the remote device |

**Table 4-8. Get Remote Device Name Confirm**

| Description | Confirm to the request above. | |
|---|---|---|
| PacketType | CFM | |
| Opcode | GAP_REMOTE_DEVICE_NAME | |
| DataLength | 8+ NameLength if ok, otherwise 8 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_INVALID_NO_OF_PARAMETERS<br>ERROR_TIMEOUT |
| | BdAddr<br>6 byte | Bluetooth device address for the remote device |
| | NameLength<br>1 byte | Number of bytes in device name |
| | DeviceName<br>Length bytes | The user-friendly name of the remote device.<br>NULL terminated. Maximum length is 30 bytes. |

### 4.2.2 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 the SDAP client.

A SDAP request can only be done via an establish SDAP link. So the Command flow would be as following:

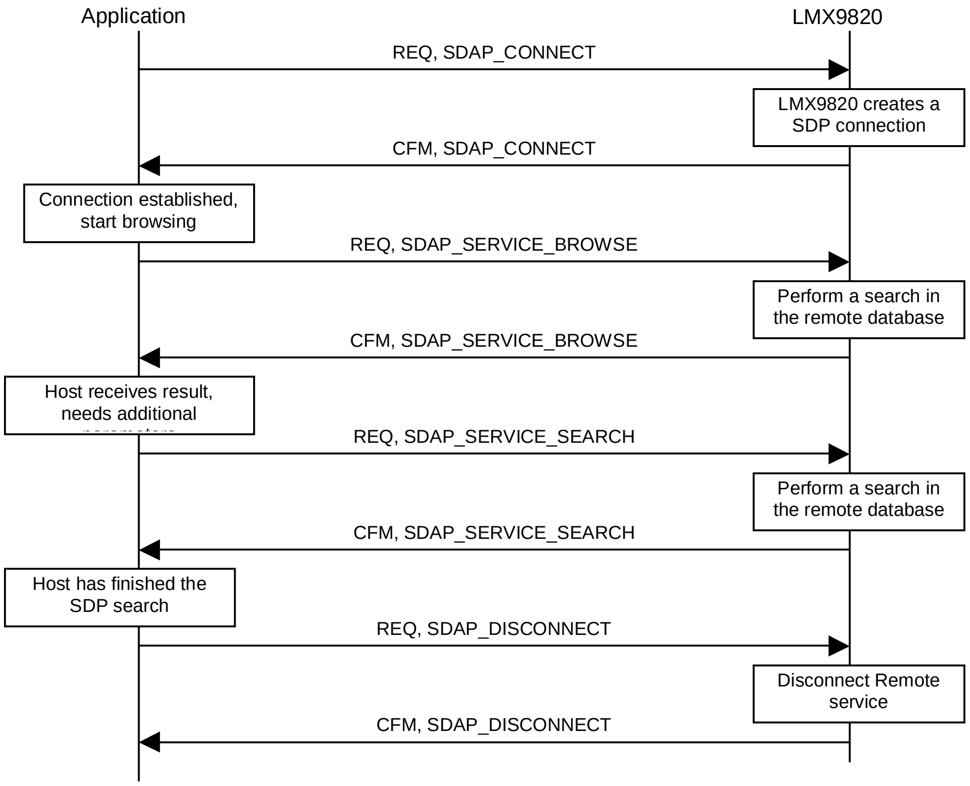

Application
LMX9820
REQ, SDAP_CONNECT
LMX9820 creates a SDP connection
CFM, SDAP_CONNECT
Connection established, start browsing
REQ, SDAP_SERVICE_BROWSE
Perform a search in the remote database
CFM, SDAP_SERVICE_BROWSE
Host receives result, needs additional
REQ, SDAP_SERVICE_SEARCH
Perform a search in the remote database
CFM, SDAP_SERVICE_SEARCH
Host has finished the SDP search
REQ, SDAP_DISCONNECT
Disconnect Remote service
CFM, SDAP_DISCONNECT

**Figure 4-3. Requesting the services of a remote device**

#### 4.2.2.1 SDAP Connect

The SDAP Connect Request forces the LMX9820 to create a SDAP link to another device. This command is required for further SDAP Service Requests

**Table 4-9. SDAP Connect Request**

| Description | Creates an SDP connection to a remote device. Only one SDP connection can be active. | |
|---|---|---|
| PacketType | REQ | |
| Opcode | SDAP_CONNECT | |
| DataLength | 6 | |
| Data | BdAddr<br>6 bytes | The Bluetooth address of the remote device to connect to. |

**Table 4-10. SDAP Connect Confirm**

| Description | Confirms the request above |
|---|---|
| PacketType | CFM |
| Opcode | SDAP_CONNECT |
| DataLength | 1 |

**Table 4-10. SDAP Connect Confirm**

| | | |
|---|---|---|
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_INVALID_NO_OF_PARAMETERS<br>ERROR_CONNECTION_FAILED |

#### 4.2.2.2 SDAP Disconnect

The SDAP link has to be disconnected after finishing the service browse/search.r

**Table 4-11. Disconnect Request**

| Description | This command disconnects the active SDP connection. |
|---|---|
| PacketType | REQ |
| Opcode | SDAP_DISCONNECT |
| DataLength | 0 |

**Table 4-12. Disconnect Confirm**

| Description | Confirms the request above | |
|---|---|---|
| PacketType | CFM | |
| Opcode | SDAP_DISCONNECT | |
| DataLength | 1 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_INVALID_NO_OF_PARAMETERS |

#### 4.2.2.3 SDAP Connection Lost Indicator

This indicator appears after an unexpected loss of the SDAP link.

**Table 4-13. Connection Lost Indicator**

| Description | Notification sent to the application when a loss of the SDP connection is detected. |
|---|---|
| PacketType | IND |
| Opcode | SDAP_CONNECTION_LOST |
| DataLength | 0 |

#### 4.2.2.4 SDAP Service Browse

The SDAP Service Browse can be used to get the RFComm Portnumbers 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 4-14. 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 4-15. SDAP Service Browse Request**

| Description | This command is used to browse the service record of the remote device. |
|---|---|
| PacketType | REQ |

Table 4-15. SDAP Service Browse Request

| | | |
|---|---|---|
| Opcode | SDAP_SERVICE_BROWSE | |
| DataLength | 2 | |
| Data | BrowseGroupID<br>2 bytes | The requested browse group (16 bit UUID).<br>The UUID has to be byte swapped within the command, e.g. to search for SPP entries the full command is<br>02 52 35 02 00 89 **01 11** 03 |

Table 4-16. SDAP Service Browse Confirm

| Description | Confirms the request above | | |
|---|---|---|---|
| PacketType | CFM | | |
| Opcode | SDAP_SERVICE_BROWSE | | |
| DataLength | 2+NoOfServices*(6 + NameLength) | | |
| Data | Status<br>1 byte | | ERROR_OK<br>ERROR_INVALID_NO_OF_PARAMETERS<br>ERROR_RESULT_TOO_LARGE<br>ERROR_TRUNCATED_ANSWER |
| | NoOfServices<br>1 byte | | Number of services found on remote device. |
| | For each service | | |
| | | BrowseGroupID<br>2 bytes | The browse group UUID that the service belongs to. |
| | | ServiceID<br>2 bytes | The service UUID. |
| | | PortNr<br>1 byte | RFCOMM port number.<br>The port which has to be used for link establishment to that service. |
| | | NameLength<br>1 byte | The number of bytes in the service name |
| | | ServiceName<br>NameLength<br>bytes | The name of the service. |

### 4.2.2.5 SDAP Service Search

The SDAP Service Search command offers the ability to search for specific attributes for a service. The attributes are defined within the SDP Specification.

Table 4-17. SDAP Service Search Request

| Description | This command is used to search for services in the service record of the remote device. | |
|---|---|---|
| PacketType | REQ | |
| Opcode | SDAP_SERVICE_SEARCH | |
| DataLength | 2 + 2*SearchPatternLength + 2*AttributesLength | |
| Data | SearchPatternLength<br>1 bytes | Number of 16-bit UUID's 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 UUID's of the requested services. |
| | AttributesLength<br>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 4-18. SDAP Service Search Confirm**

| Description | Confirms the request above | |
|---|---|---|
| PacketType | CFM | |
| Opcode | SDAP_SERVICE_SEARCH | |
| DataLength | 3 + Length | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_INVALID_NO_OF_PARAMETERS<br>ERROR_UNKNOWN_ERROR<br>ERROR_RESULT_TOO_LARGE |
| | Length<br>2 byte | Length of the result of the search. Maximum 330 bytes |
| | Result<br>Length bytes | Result of the search |

#### 4.2.2.6 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 4-19. SDAP Service Request**

| Description | This command is used the service record handles, from a remote device, for the given services in the search pattern. | |
|---|---|---|
| PacketType | REQ | |
| Opcode | SDAP_SERVICE_REQUEST | |
| DataLength | 1 +2* SearchPatternLength | |
| Data | SearchPatternLength<br>1 bytes | Number of 16-bit UUID’s in Search pattern List. **Note**: Must be less than 86 elements. |
| | SearchPattern<br>< 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 4-20. SDAP Service Request Confirm**

| Description | Confirms the request above | |
|---|---|---|
| PacketType | CFM | |
| Opcode | SDAP_SERVICE_REQUEST | |
| DataLength | 3 + 4*Length | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_INVALID_NO_OF_PARAMETERS<br>ERROR_UNKNOWN_ERROR<br>ERROR_RESULT_TOO_LARGE |
| | Length<br>2 byte | Number of 32-bit service record handles returned from remote device. |
| | Result<br><4*Length> bytes | The received 32-bit service record handles from the remote device. |

#### 4.2.2.7 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 Search” Command. (see Section 4.2.2.5 on page 37)

**Table 4-21. SDAP Attribute Request**

| **Description** | **This command is used to get the given attributes for a given service record handle.** | |
|---|---|---|
| PacketType | REQ | |
| Opcode | SDAP_ATTRIBUTE_REQUEST | |
| DataLength | 5 + 2*AttributesLength | |
| Data | Handle<br>4 bytes | The 32-bit service record handle returned for a given service by SDAP_ATTRIBUTE_REQUEST. |
| | AttributesLength<br>1 byte | Number of 16-bit UUID’s in attributeId list. **Note**: Must be less than 86 elements. |
| | Attributes<br>< 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 4-22. SDAP Attribute Request Confirm**

| **Description** | **Confirms the request above** | |
|---|---|---|
| PacketType | CFM | |
| Opcode | SDAP_ATTRIBUTE_REQUEST | |
| DataLength | 3 + Length | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_INVALID_NO_OF_PARAMETERS<br>ERROR_UNKNOWN_ERROR<br>ERROR_RESULT_TOO_LARGE |
| | Length<br>2 byte | Length of the result of the search. Maximum 330 bytes. |
| | Result | Result of the search |

### 4.2.3 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 LMX9820 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”.

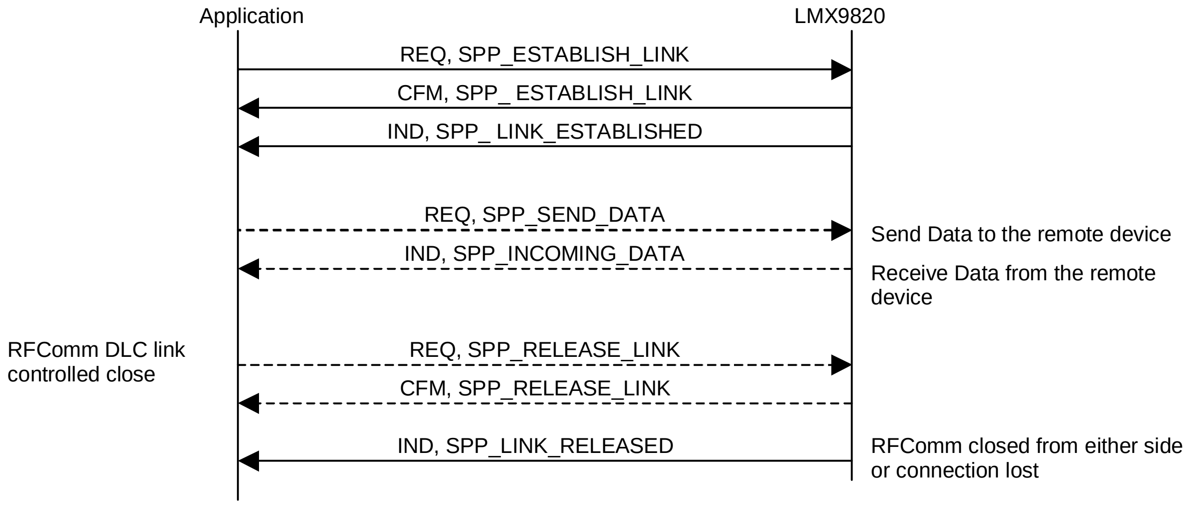

Application
LMX9820
REQ, SPP_ESTABLISH_LINK
CFM, SPP_ ESTABLISH_LINK
IND, SPP_ LINK_ESTABLISHED
REQ, SPP_SEND_DATA
Send Data to the remote device
IND, SPP_INCOMING_DATA
Receive Data from the remote device
RFComm DLC link controlled close
REQ, SPP_RELEASE_LINK
CFM, SPP_RELEASE_LINK
IND, SPP_LINK_RELEASED
RFComm closed from either side or connection lost

**Figure 4-4. SPP Link Handling Flow**

#### 4.2.3.1 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 4-23. Establish Link Request**

| Description | Establish a DLC link to remote Bluetooth device | |
|---|---|---|
| PacketType | REQ | |
| Opcode | SPP_ESTABLISH_LINK | |
| DataLength | 8 | |
| Data | LocalPort<br>1 byte | Local RFCOMM port number. Range 1-30 |
| | BdAddr<br>6 byte | Bluetooth device address for the remote device |
| | RemotePortNumber<br>1 byte | Remote device RFCOMM port number.<br>(Must be found using SDAP) |

**Table 4-24. Establish Link Confirm**

| Description | Confirm that the DLC link establishment is initiated.<br>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 LMX9820. |
|---|---|
| PacketType | CFM |

**Table 4-24. Establish Link Confirm**

| | | |
|---|---|---|
| Opcode | SPP_ESTABLISH_LINK | |
| DataLength | 2 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_SPP_PORT_BUSY<br>ERROR_SPP_PORT_NOT_OPEN<br>ERROR_SPP_INVALID_PORT<br>ERROR_SPP_AUTOMATIC_CONNECTIONS_PROGRESSING |
| | LocalPort<br>1 byte | Local RFCOMM port number. Range 1-30 |

**Table 4-25. SPP Link Established Indicator**

| Description | Indication of establishment of a locally requested DLC link. | |
|---|---|---|
| PacketType | IND | |
| Opcode | SPP_LINK_ESTABLISHED | |
| DataLength | 9 | |
| Data | Status<br>1 byte | Refer to “RFCOMM Error Codes” on page 91. |
| | BdAddr<br>6 byte | Bluetooth device address for the remote device. |
| | LocalPort<br>1 byte | Local RFCOMM port number. Range 1-30. |
| | RemotePortNumber<br>1 byte | Remote device RFCOMM port number. |

#### 4.2.3.2 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.I

**Table 4-26. SPP Send Data**

| Description | Send data on a SPP link to remote Bluetooth device | |
|---|---|---|
| PacketType | REQ | |
| Opcode | SPP_SEND_DATA | |
| DataLength | 3 + <PayloadSize> | |
| Data | LocalPort<br>1 byte | Local RFCOMM port number. Range 1-30 |
| | PayloadSize<br>2 bytes | Number of data bytes to send.<br>Valid range is 1 to 330 bytes. |
| | PayloadData<br><PayloadSize> bytes | The data to send. |

| Description | Confirm to the request above. |
|---|---|
| PacketType | CFM |
| Opcode | SPP_SEND_DATA |
| DataLength | 2 |

| | | |
|---|---|---|
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_LIMIT<br>ERROR_UNABLE_TO_SEND<br>ERROR_CURRENTLY_NO_BUFFER<br>ERROR_NO_CONNECTION<br>ERROR_SPP_INVALID_PORT<br>ERROR_SPP_PORT_NOT_OPEN |
| | LocalPort<br>1 byte | Local RFCOMM port number. Range 1-30 |

**Table 4-27. SPP Incoming Data Event**

| **Description** | **Incoming data on a DLC link, from a remote Bluetooth device** | |
|---|---|---|
| PacketType | IND | |
| Opcode | SPP_INCOMING_DATA | |
| DataLength | 3 + <PayloadSize> | |
| Data | LocalPort<br>1 byte | Local RFCOMM port number. Range 1-30 |
| | PayloadSize<br>2 bytes | Number of data bytes to send.<br>Valid range is 1 to 330 bytes. |
| | PayloadData<br><PayloadSize> bytes | The data to send. |

#### 4.2.3.3 SPP Release Link

**Table 4-28. SPP Release Link Request**

| **Description** | **Release a DLC link to remote Bluetooth device** | |
|---|---|---|
| PacketType | REQ | |
| Opcode | SPP_RELEASE_LINK | |
| DataLength | 1 | |
| Data | LocalPort<br>1 byte | Local RFCOMM port number. Range 1-30 |

**Table 4-29. 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 LMX9820.** | |
|---|---|---|
| PacketType | CFM | |
| Opcode | SPP_RELEASE_LINK | |
| DataLength | 2 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_NO_CONNECTION<br>ERROR_SPP_INVALID_PORT<br>ERROR_SPP_PORT_NOT_OPEN |

**Table 4-29. SPP Release Link Confirm**

| | | |
|---|---|---|
| | LocalPort<br>1 byte | Local RFCOMM port number. Range 1-30 |

**Table 4-30. SPP Link Released Indicator**

| **Description** | **Indicates that a DLC link is released.<br>The link release may have been initiated locally or remote, or could be caused by a loss of link (disturbance, dead device, ….)** | |
|---|---|---|
| PacketType | IND | |
| Opcode | SPP_LINK_RELEASED | |
| DataLength | 2 | |
| Data | Reason<br>1 byte | Refer to “RFCOMM Release Reasons” on page 91 |
| | LocalPort<br>1 byte | Local RFCOMM port number. Range 1-30 |

### 4.2.3.4 Incoming Link Established Indicator

In case a remote device is connecting to the LMX9820, the device will indicate the successfull link establishment by sending by the “Incoming Link Established Indicator”. The packet includes the BD_Addr of the remote device and the local RFCommport it connected to.

**Table 4-31. Incoming Link Established Indicator**

| **Description** | **Indication of establishment of a remotely requested DLC link.** | |
|---|---|---|
| PacketType | IND | |
| Opcode | SPP_ INCOMING_LINK_ESTABLISHED | |
| DataLength | 7 | |
| Data | BdAddr<br>6 byte | Bluetooth device address for the remote device |
| | LocalPort<br>1 byte | Local RFCOMM port number. Range 1-30 |

### 4.2.4 Transparent Mode

Transparent Mode offers the ability to switch off the Command Interface on the LMX9820 and use it as a cable replacement. This means data can be sent over a bluetooth link just by routing them to the LMX9820 without any package framing. Transparent mode can only be enabled if one SPP link is establised. Transparent mode can’t be used if a device has two or more active SPP links.

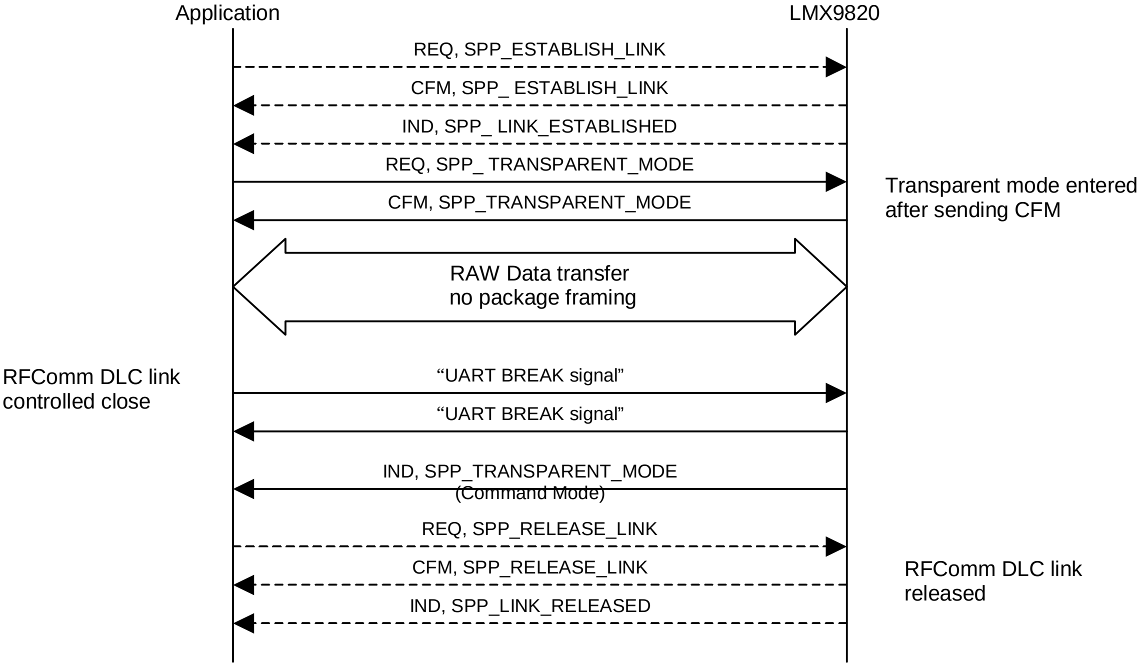

Application
LMX9820
REQ, SPP_ESTABLISH_LINK
CFM, SPP_ ESTABLISH_LINK
IND, SPP_ LINK_ESTABLISHED
REQ, SPP_ TRANSPARENT_MODE
CFM, SPP_TRANSPARENT_MODE
Transparent mode entered after sending CFM
RAW Data transfer no package framing
RFComm DLC link controlled close
"UART BREAK signal"
"UART BREAK signal"
IND, SPP_TRANSPARENT_MODE (Command Mode)
REQ, SPP_RELEASE_LINK
CFM, SPP_RELEASE_LINK
IND, SPP_LINK_RELEASED
RFComm DLC link released

**Figure 4-5. Transparent Mode; initiated by the Transparent Mode Request**

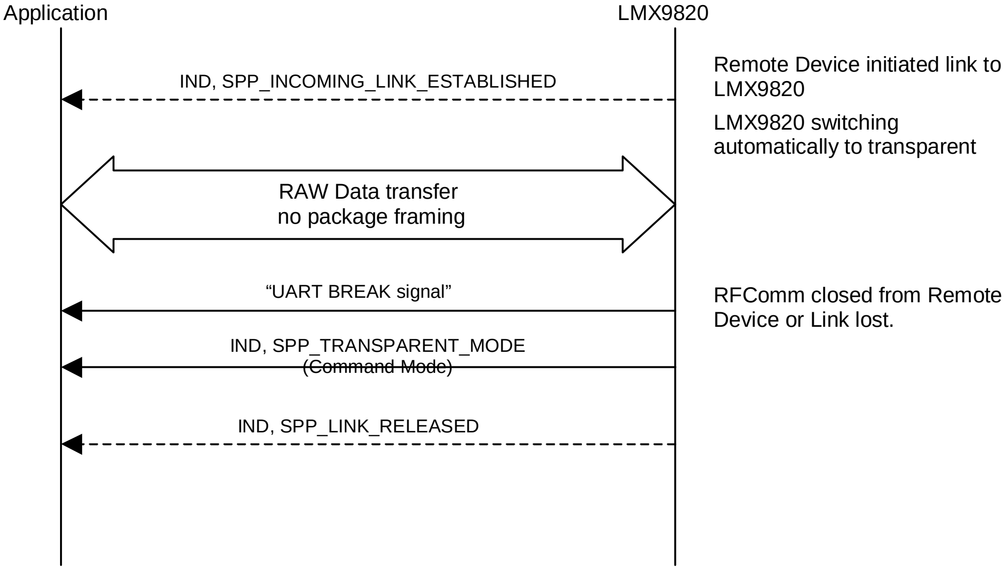

Application
LMX9820
IND, SPP_INCOMING_LINK_ESTABLISHED
Remote Device initiated link to LMX9820
LMX9820 switching automatically to transparent
RAW Data transfer no package framing
"UART BREAK signal"
RFComm closed from Remote Device or Link lost.
IND, SPP_TRANSPARENT_MODE (Command Mode)
IND, SPP_LINK_RELEASED

**Figure 4-6. Transparent Mode; automatically activated in automatic slave operation**

#### 4.2.4.1 Transparent Mode

Table 4-32. Transparent Mode Request

| Description | Switch to transparent mode on a SPP link to remote Bluetooth device | |
|---|---|---|
| PacketType | REQ | |
| Opcode | SPP_TRANSPARENT_MODE | |
| DataLength | 1 | |
| Data | LocalPort<br>1 byte | Local RFCOMM port number. Range 1-30 |

Table 4-33. Transparent Mode Confirm

| Description | Confirm to the request above. | |
|---|---|---|
| PacketType | CFM | |
| Opcode | SPP_TRANSPARENT_MODE | |
| DataLength | 2 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_SPP_PORT_NOT_OPEN<br>ERROR_SPP_INVALID_PORT<br>ERROR_SPP_MULTIPLE_CONNECTIONS<br>ERROR_NO_CONNECTION |
| | LocalPort<br>1 byte | Local RFCOMM port number. Range 1-30 |

Table 4-34. Transparent Mode Left Indicator

| Description | Indication from Simply Blue that transparent mode is left. | |
|---|---|---|
| PacketType | IND | |
| Opcode | SPP_TRANSPARENT_MODE | |
| DataLength | 2 | |
| Data | LocalPort<br>1 byte | Local RFCOMM port number. Range 1-30 |
| | Mode<br>1 byte | Command Mode = 0<br>Transparent Mode = 1 |

### 4.2.5 Default Connections

A special feature for link setup is the storage of default connections. The LMX9820 can store a limited amount of connections in its non-volatile data memory. If the “Operation Mode” parameter in NVS is set to automatic, the LMX9820 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 4-35. Example Default Connection Table

| Index | Local Port | BD_Addr | Remote Port | Transparent |
|---|---|---|---|---|
| 01 | 01 | 12 34 56 78 90 12 | 01 | no |
| 02 | 02 | 98 76 54 32 10 01 | 03 | no |

#### 4.2.5.1 Store Default Connection

Table 4-36. Store Default Connection Confirm Request

| **Description** | **Stores a default connection in NVS.** | |
|---|---|---|
| PacketType | REQ | |
| Opcode | SPP_STORE_DEFAULT_CON | |
| DataLength | 10 | |
| Data | Index<br>1 byte | Index in the default connection storage. Range 0x00 to 0x02. |
| | LocalPort<br>1 byte | Local RFCOMM port number. Range 1-30 |
| | RemotePort<br>1 byte | RFCOMM port number on remote device |
| | RemoteBdAddress<br>6 bytes | The BdAddress of the remote device |
| | TransparentMode<br>1 byte | Which transparent mode to enter when the connection is established.<br>0x00 Transparent mode off<br>0x01 Transparent mode on (only for point-to-point). |

Table 4-37. Store Default Connection Confirm

| **Description** | **Confirm to the request above.** | |
|---|---|---|
| PacketType | CFM | |
| Opcode | SPP_STORE_DEFAULT_CON | |
| DataLength | 1 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_LIMIT<br>ERROR_SPP_INVALID_PORT<br>ERROR_SPP_MULTIPLE_TRANSPARENT<br>ERROR_SPP_PORT_BUSY |

#### 4.2.5.2 Connect to Default Connection

Table 4-38. Connect to Default Connection Request

| **Description** | **Connects the LMX9820 to a stored connection. Either a single or all connections can be established** |
|---|---|
| PacketType | REQ |

Table 4-38. Connect to Default Connection Request

| | | |
|---|---|---|
| Opcode | SPP_CONNECT_DEFAULT_CON | |
| DataLength | 1 | |
| Data | Index<br>1 byte | Index in the default connection storage. Range 0x00 to 0x02. If index is set to 0xFF, all default connections will be established. |

Table 4-39. Store Default Connection Confirm

| **Description** | **Confirm to the request above.** | |
|---|---|---|
| PacketType | CFM | |
| Opcode | SPP_ CONNECT _DEFAULT_CON | |
| DataLength | 1 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_SPP_AUTOMATIC_CONNECTIONS_PROGRESSING<br>ERROR_LIMIT<br>ERROR_SPP_DEFAULT_CONNECTION_NOT_STORED |

### 4.2.5.3 Get List of Default Connections

Table 4-40. Get List of Default Connections Request

| **Description** | **Request a list of the default connections stored in NVS** | |
|---|---|---|
| PacketType | REQ | |
| Opcode | SPP_GET_LIST_DEFAULT_CON | |
| DataLength | 1 | |
| Data | Index<br>1 byte | Index in the default connection storage. Range 0x00 to 0x02. |

Table 4-41. Get List of Default Connections Confirm

| **Description** | **Confirm to the request above.** | | |
|---|---|---|---|
| PacketType | CFM | | |
| Opcode | SPP_GET_LIST_DEFAULT_CON | | |
| DataLength | 11 | | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_LIMIT | |
| | Index<br>1 byte | Index in the default connection storage. Range 0x00 to 0x02. | |
| | EntryStatus<br>1 byte | Bitfield: 0000000x | 0: Entry is not stored.<br>1: Entry is stored. |
| | | Bitfield: 000000x0 | 0: Command mode.<br>1: Transparent mode. |
| | | Bitfield: xxxxxx00 | Reserved. |
| | RemoteBdAddress<br>6 bytes | The BdAddress of the remote device | |
| | LocalPort<br>1 byte | Local RFCOMM port number. Range 1-30 | |

Table 4-41. Get List of Default Connections Confirm

| | | |
|---|---|---|
| | RemotePort<br>1 byte | RFCOMM port number on remote device |

### 4.2.5.4 Delete Default Connection

Table 4-42. Delete Default Connections Request

| **Description** | **Deletes a stored default connection in NVS** | |
|---|---|---|
| PacketType | REQ | |
| Opcode | SPP_DELETE_DEFAULT_CON | |
| DataLength | 1 | |
| Data | Index<br>1 byte | Index in the default connection storage. Range 0x00 to 0x02. |

Table 4-43. Delete Default Connections Confirm

| **Description** | **Confirm to the request above.** | |
|---|---|---|
| PacketType | CFM | |
| Opcode | SPP_DELETE_DEFAULT_CON | |
| DataLength | 2 | |
| Data | Index<br>1 byte | Index in the default connection storage. Range 0x00 to 0x02. |
| | Status<br>1 byte | 0x00 OK |

### 4.2.6 Bluetooth Low Power Modes

A bluetooth link is a based on a physically synchronized connection, which means that the devices can only communicate after successfull synchronisation. For this, each package also includes some synchronisation information. Also a specific polling scheme is in place to keep synchronisation 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 successfull if both sides support it.

### 4.2.6.1 Set Default Link Policy

Table 4-44. 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.** |
|---|---|
| PacketType | REQ |
| Opcode | GAP_SET_DEFAULT_LINK_POLICY |
| DataLength | 2 |

Table 4-44.  Set Default Link Policy Command

| | | |
|---|---|---|
| Data | LinkPolicy<br>2 byte | Bitfield:<br>0x0001 = Master-slave switch allowed<br>0x0002 = Hold mode allowed<br>0x0004 = Sniff mode allowed<br>0x0008 = Park mode allowed |

Table 4-45.  Set Default Link Policy Confirm

| **Description** | **Response to the request above.** | |
|---|---|---|
| PacketType | CFM | |
| Opcode | GAP_SET_DEFAULT_LINK_POLICY | |
| DataLength | 1 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_INVALID_NO_OF_PARAMETERS<br>ERROR_ILLEGAL_LINK_POLICY |

#### 4.2.6.2 Get Default Link Policy

Table 4-46.  Get Default Link Policy Command

| **Description** | **This command is used to read the default link policy setting from NVS.** |
|---|---|
| PacketType | REQ |
| Opcode | GET_DEFAULT_LINK_POLICY |
| DataLength | 0 |

Table 4-47.  Get Default Link Policy Confirm

| **Description** | **Response to the request above.** | |
|---|---|---|
| PacketType | CFM | |
| Opcode | GET_DEFAULT_LINK_POLICY | |
| DataLength | 2 | |
| Data | LinkPolicy<br>2 byte | Bitfield:<br>0x0001 = Master-slave switch allowed<br>0x0002 = Hold mode allowed<br>0x0004 = Sniff mode allowed<br>0x0008 = Park mode allowed |

#### 4.2.6.3 Set Link Policy

Table 4-48.  Set Link Policy Request

| **Description** | **This command is used to change the current link policy setting for the given link.** | |
|---|---|---|
| PacketType | REQ | |
| Opcode | GAP_SET_LINK_POLICY | |
| DataLength | 8 | |
| Data | BdAddr<br>6 bytes | The Bluetooth address of the remote device of which to change the link policy settings for the link. |

**Table 4-48. Set Link Policy Request**

| | | |
|---|---|---|
| | LinkPolicy<br>2 byte | Bitfield:<br>0x0001 = Master-slave switch allowed<br>0x0002 = Hold mode allowed<br>0x0004 = Sniff mode allowed<br>0x0008 = Park mode allowed |

**Table 4-49. Set Link Policy Confirm**

| Description | Response to the request above. | |
|---|---|---|
| PacketType | CFM | |
| Opcode | GAP_SET_LINK_POLICY | |
| DataLength | 1 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_UNSPECIFIED_ERROR<br>ERROR_INVALID_NO_OF_PARAMETERS<br>ERROR_NO_CONNECTION<br>ERROR_ILLEGAL_LINK_POLICY |

### 4.2.6.4 Get Link Policy

**Table 4-50. Get Link Policy Request**

| Description | This command is used to get the current link policy setting for the given link. | |
|---|---|---|
| PacketType | REQ | |
| Opcode | GAP_GET_LINK_POLICY | |
| DataLength | 6 | |
| Data | BdAddr<br>6 bytes | The Bluetooth address of the remote device of which to get the current link policy settings for the link. |

**Table 4-51. Get Link Policy Confirm**

| Description | Response to the request above. | |
|---|---|---|
| PacketType | CFM | |
| Opcode | GAP_GET_LINK_POLICY | |
| DataLength | 3 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_UNSPECIFIED_ERROR<br>ERROR_INVALID_NO_OF_PARAMETERS<br>ERROR_NO_CONNECTION |
| | LinkPolicy<br>2 byte | Bitfield:<br>0x0001 = Master-slave switch allowed<br>0x0002 = Hold mode allowed<br>0x0004 = Sniff mode allowed<br>0x0008 = Park mode allowed |

#### 4.2.6.5 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 4-52. Enter Sniff Mode Request**

| Description | This command is used to request sniff mode on a given link with user specified parameters. | |
|---|---|---|
| PacketType | REQ | |
| Opcode | GAP_ENTER_SNIFF_MODE | |
| DataLength | 14 | |
| Data | BdAddr<br>6 bytes | The Bluetooth address of the remote device of which to put the link in sniff mode. |
| | SniffMaxInterval<br>2 bytes | Maximum sniff interval in slots. |
| | SniffMinInterval<br>2 bytes | Minimum sniff interval in slots |
| | SniffAttempt<br>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<br>2 bytes | Number of additional slots the slave must listen if it continues to receive only packets with its own AM address. |

**Table 4-53. Enter Sniff Mode Confirm**

| Description | Response to the request above. | |
|---|---|---|
| PacketType | CFM | |
| Opcode | GAP_ENTER_SNIFF_MODE | |
| DataLength | 1 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_UNSPECIFIED_ERROR<br>ERROR_INVALID_NO_OF_PARAMETERS<br>ERROR_NO_CONNECTION |

#### 4.2.6.6 Exit Sniff Mode

**Table 4-54. Exit Sniff Mode Request**

| Description | This command is used to exit a current sniff mode on a given link. | |
|---|---|---|
| PacketType | REQ | |
| Opcode | GAP_EXIT_SNIFF_MODE | |
| DataLength | 6 | |
| Data | BdAddr<br>6 bytes | The Bluetooth address of the remote device of which to exit the current sniff mode. |

**Table 4-55. Exit Sniff Mode Confirm**

| Description | Response to the request above. |
|---|---|
| PacketType | CFM |
| Opcode | GAP_EXIT_SNIFF_MODE |
| DataLength | 1 |

Table 4-55. Exit Sniff Mode Confirm

| | | |
|---|---|---|
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_UNSPECIFIED_ERROR<br>ERROR_INVALID_NO_OF_PARAMETERS<br>ERROR_NO_CONNECTION |

#### 4.2.6.7 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 synchronised by beacons within the specified interval range.

Table 4-56. Enter Park Mode Request

| Description | This command is used to request park mode on a given link with user specified parameters. | |
|---|---|---|
| PacketType | REQ | |
| Opcode | GAP_ENTER_PARK_MODE | |
| DataLength | 10 | |
| Data | BdAddr<br>6 bytes | The Bluetooth address of the remote device of which to put the link in park mode. |
| | BeaconMaxInterval<br>2 bytes | Acceptable longest length of the interval between beacons. |
| | BeaconMinInterval<br>2 bytes | Acceptable shortest length of the interval between beacons. |

Table 4-57. Enter Park Mode Confirm

| Description | Response to the request above. | |
|---|---|---|
| PacketType | CFM | |
| Opcode | GAP_ENTER_PARK_MODE | |
| DataLength | 1 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_UNSPECIFIED_ERROR<br>ERROR_INVALID_NO_OF_PARAMETERS<br>ERROR_NO_CONNECTION |

#### 4.2.6.8 Exit Park Mode

This commands forces the devices getting the parked slave back as active member of the piconet.

Table 4-58. Exit Park Mode Request

| Description | This command is used to exit a current park mode on a given link. | |
|---|---|---|
| PacketType | REQ | |
| Opcode | GAP_EXIT_PARK_MODE | |
| DataLength | 6 | |
| Data | BdAddr<br>6 bytes | The Bluetooth address of the remote device of which to exit the current park mode. |

Table 4-59. Enter Park Mode Confirm

| **Description** | **Response to the request above.** | |
|---|---|---|
| PacketType | CFM | |
| Opcode | GAP_ENTER_PARK_MODE | |
| DataLength | 1 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_UNSPECIFIED_ERROR<br>ERROR_INVALID_NO_OF_PARAMETERS<br>ERROR_NO_CONNECTION |

#### 4.2.6.9 Enter Hold Mode

Table 4-60. Enter Hold Mode Request

| **Description** | **This command is used to request Hold mode on a given link with user specified parameters.** | |
|---|---|---|
| PacketType | REQ | |
| Opcode | GAP_ENTER_HOLD_MODE | |
| DataLength | 10 | |
| Data | BdAddr<br>6 bytes | The Bluetooth address of the remote device of which to put the link in Hold mode. |
| | HoldMaxInterval<br>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<br>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 4-61. Enter Hold Mode Confirm

| **Description** | **Response to the request above.** | |
|---|---|---|
| PacketType | CFM | |
| Opcode | GAP_ENTER_HOLD_MODE | |
| DataLength | 1 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_UNSPECIFIED_ERROR<br>ERROR_INVALID_NO_OF_PARAMETERS<br>ERROR_NO_CONNECTION |

#### 4.2.6.10 Power Save Mode Changed

In case the remote device changed the Power Mode for that link, this event will be returned by the LMX9820.

Table 4-62. Power Save Mode Changed Indicator

| **Description** | **This indication is sent to the host when changes the power save mode on a link occur.** |
|---|---|

**Table 4-62. Power Save Mode Changed Indicator**

| | | |
|---|---|---|
| PacketType | IND | |
| Opcode | GAP_POWER_SAVE_MODE_CHANGED | |
| DataLength | 8 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_ATTEMPT_FAILED<br>ERROR_UNSPECIFIED_ERROR |
| | BdAddr<br>6 bytes | The Bluetooth address of the remote device for which the power save mode has changed on the link. |
| | Mode<br>1 byte | 0x00 = Active mode (Left power save mode)<br>0x01 = Hold mode (Hold mode entered)<br>0x02 = Sniff mode (Sniff mode entered)<br>0x03 = Park mode (Park mode entered) |

### 4.2.7 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.

#### 4.2.7.1 SPP Set Port Configuration

**Table 4-63. SPP Set Port Config Request**

| **Description** | **Write the configuration for the SPP port. This is "virtual" settings for the air connection, not settings for the LMX9820 serial port.** | | | |
|---|---|---|---|---|
| PacketType | REQ | | | |
| Opcode | SPP_SET_PORT_CONFIG | | | |
| DataLength | 6 | | | |
| Data | LocalPort<br>1 byte | Local RFCOMM port number. Range 1-30 | | |
| | BaudRate<br>1 byte | Baudrate<br>0x00 = 2400 baud 0x01 = 4800 baud 0x02 = 7200 baud<br>0x03 = 9600 baud 0x04 = 19200 baud 0x05 = 38400 baud<br>0x06 = 57600 baud 0x07 = 115200 baud 0x08 = 230400 baud | | |
| | Portsettings<br>1 byte | Bitfield: 000000XX | Number of databits | 00=5 bits<br>01=6 bits<br>02=7 bits<br>03=8 bits |
| | | Bitfield: 00000X00 | Number of stopbits | 0 = 1<br>1 = 1.5 |
| | | Bitfield: 0000X000 | Parity | 0 = No parity<br>1 = Parity |
| | | Bitfield: 00XX0000 | ParityType | 00=ODD<br>01=EVEN<br>02=MARK<br>03=SPACE |
| | | Bitfield: XX000000 | Reserved | |

Table 4-63. SPP Set Port Config Request

| | | | | |
|---|---|---|---|---|
| | FlowControl<br>1 byte | Bitfield: 0000000X | XonXoffOnInput | 0=Disable<br>1=Enable |
| | | Bitfield: 000000X0 | XonXoffOnOutput | 0=Disable<br>1=Enable |
| | | Bitfield: 00000X00 | RtrOnInput | 0=Disable<br>1=Enable |
| | | Bitfield: 0000X000 | RtrOnOutput | 0=Disable<br>1=Enable |
| | | Bitfield: 000X0000 | RtcOnInput | 0=Disable<br>1=Enable |
| | | Bitfield: 00X00000 | RtcOnOutput | 0=Disable<br>1=Enable |
| | | Bitfield: XX000000 | Reserved | |
| | XonChar<br>1 byte | Char used for Xon, if Xon/Xoff flowcontrol is used. | | |
| | XoffChar<br>1 byte | Char used for Xoff, if Xon/Xoff flowcontrol is used. | | |

Table 4-64. SPP Set Port Configuration Confirm

| **Description** | **Confirm to the request above.** | |
|---|---|---|
| PacketType | CFM | |
| Opcode | SPP_SET_PORT_CONFIG | |
| DataLength | 2 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_UNSPECIFIED_ERROR<br>ERROR_UNSPECIFIED_ERROR<br>ERROR_SPP_INVALID_PORT<br>ERROR_SPP_PORT_NOT_OPEN<br>ERROR_UART_SPEED_OUT_OF_RANGE |
| | LocalPort<br>1 byte | Local RFCOMM port number. Range 1-30 |

### 4.2.7.2 SPP Get Port Configuration

Table 4-65. 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 LMX9820 serial port.** | |
|---|---|---|
| PacketType | REQ | |
| Opcode | SPP_GET_PORT_CONFIG | |
| DataLength | 1 | |
| Data | LocalPort<br>1 byte | Local RFCOMM port number. Range 1-30 |

Table 4-66. SPP Get Port Configuration Confirm

| Description | Confirm to the request above. | | | |
|---|---|---|---|---|
| PacketType | CFM | | | |
| Opcode | SPP_GET_PORT_CONFIG | | | |
| DataLength | 7 | | | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_UNSPECIFIED_ERROR<br>ERROR_SPP_INVALID_PORT<br>ERROR_SPP_PORT_NOT_OPEN | | |
| | LocalPort<br>1 byte | Local RFCOMM port number. Range 1-30 | | |
| | BaudRate<br>1 byte | Baudrate<br>0x00 = 2400 baud 0x01 = 4800 baud 0x02 = 7200 baud<br>0x03 = 9600 baud 0x04 = 19200 baud 0x05 = 38400 baud<br>0x06 = 57600 baud 0x07 = 115200 baud 0x08 = 230400 baud | | |
| | Portsettings<br>1 byte | Bitfield: 000000XX | Number of databits | 00=5 bits<br>01=6 bits<br>02=7 bits<br>03=8 bits |
| | | Bitfield: 00000X00 | Number of stopbits | 0 = 1<br>1 = 1.5 |
| | | Bitfield: 0000X000 | Parity | 0 = No parity<br>1 = Parity |
| | | Bitfield: 00XX0000 | ParityType | 00=ODD<br>01=EVEN<br>02=MARK<br>03=SPACE |
| | | Bitfield: XX000000 | Reserved | |
| | FlowControl<br>1 byte | Bitfield: 0000000X | XonXoffOnInput | 0=Disable<br>1=Enable |
| | | Bitfield: 000000X0 | XonXoffOnOutput | 0=Disable<br>1=Enable |
| | | Bitfield: 00000X00 | RtrOnInput | 0=Disable<br>1=Enable |
| | | Bitfield: 0000X000 | RtrOnOutput | 0=Disable<br>1=Enable |
| | | Bitfield: 000X0000 | RtcOnInput | 0=Disable<br>1=Enable |
| | | Bitfield: 00X00000 | RtcOnOutput | 0=Disable<br>1=Enable |
| | | Bitfield: XX000000 | Reserved | |

**Table 4-66. SPP Get Port Configuration Confirm**

| | | |
|---|---|---|
| | XonChar<br>1 byte | Char used for Xon, if Xon/Xoff flowcontrol is used. |
| | XoffChar<br>1 byte | Char used for Xoff, if Xon/Xoff flowcontrol is used. |

### 4.2.7.3 SPP Port Configuration Changed Indicator

**Table 4-67. SPP Port Configuration Changed Indicator**

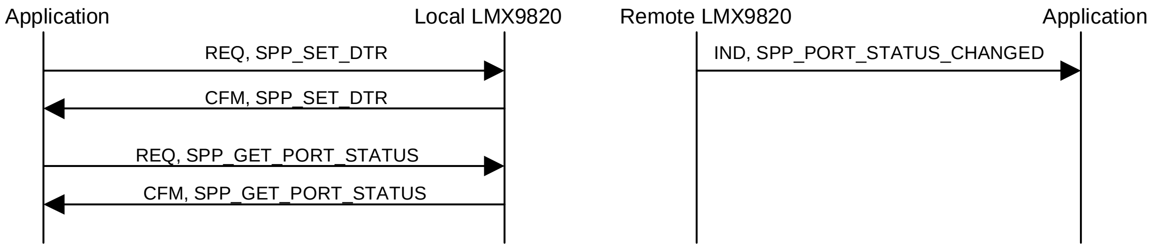
| Description | Send from the LMX9820 when remote device has changed the port configuration. | | | |
|---|---|---|---|---|
| PacketType | IND | | | |
| Opcode | SPP_PORT_CONFIG_CHANGED | | | |
| DataLength | 6 | | | |
| Data | LocalPort<br>1 byte | Local RFCOMM port number. Range 1-30 | | |
| | BaudRate<br>1 byte | Baudrate<br>0x00 = 2400 baud 0x01 = 4800 baud 0x02 = 7200 baud<br>0x03 = 9600 baud 0x04 = 19200 baud 0x05 = 38400 baud<br>0x06 = 57600 baud 0x07 = 115200 baud 0x08 = 230400 baud | | |
| | Portsettings<br>1 byte | Bitfield: 000000XX | Number of databits | 00=5 bits<br>01=6 bits<br>02=7 bits<br>03=8 bits |
| | | Bitfield: 00000X00 | Number of stopbits | 0 = 1<br>1 = 1.5 |
| | | Bitfield: 0000X000 | Parity | 0 = No parity<br>1 = Parity |
| | | Bitfield: 00XX0000 | ParityType | 00=ODD<br>01=EVEN<br>02=MARK<br>03=SPACE |
| | | Bitfield: XX000000 | Reserved | |
| | FlowControl<br>1 byte | Bitfield: 0000000X | XonXoffOnInput | 0=Disable<br>1=Enable |
| | | Bitfield: 000000X0 | XonXoffOnOutput | 0=Disable<br>1=Enable |
| | | Bitfield: 00000X00 | RtrOnInput | 0=Disable<br>1=Enable |
| | | Bitfield: 0000X000 | RtrOnOutput | 0=Disable<br>1=Enable |
| | | Bitfield: 000X0000 | RtcOnInput | 0=Disable<br>1=Enable |
| | | Bitfield: 00X00000 | RtcOnOutput | 0=Disable<br>1=Enable |
| | | Bitfield: XX000000 | Reserved | |

Table 4-67. SPP Port Configuration Changed Indicator

| | | |
|---|---|---|
| | XonChar<br>1 byte | Char used for Xon, if Xon/Xoff flowcontrol is used. |
| | XoffChar<br>1 byte | Char used for Xoff, if Xon/Xoff flowcontrol is used. |

### 4.2.8 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.



#### 4.2.8.1 SPP Get Port Status

| **Description** | **Get the current state of the modem status and line status. This command resets the value (to 0) of the following members of the port status:**<br>**OverrunError**<br>**ParityError**<br>**FramingError**<br>**BreakLength**<br>**The value of DSR and CTS are only changed when new values are received from the remote device!** | |
|---|---|---|
| PacketType | REQ | |
| Opcode | SPP_GET_PORT_STATUS | |
| DataLength | 1 | |
| Data | LocalPort<br>1 byte | Local RFCOMM port number. Range 1-30 |

Table 4-68. SPP Get Port Status Confirm

| **Description** | **Confirm to the request above.** | | | |
|---|---|---|---|---|
| PacketType | CFM | | | |
| Opcode | SPP_GET_PORT_STATUS | | | |
| DataLength | 5 | | | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_UNSPECIFIED_ERROR<br>ERROR_SPP_INVALID_PORT<br>ERROR_SPP_PORT_NOT_OPEN | | |
| | LocalPort<br>1 byte | Local RFCOMM port number. Range 1-30 | | |
| | PortStatus<br>1 byte | Bitfield: 0000000X | DTR | 0 = Low<br>1 = High |

**Table 4-68. SPP Get Port Status Confirm**

| | | | | |
|---|---|---|---|---|
| | | Bitfield: 000000X0 | RTS | 0 = Low<br>1 = High |
| | | Bitfield: 00000X00 | DSR | 0 = Low<br>1 = High |
| | | Bitfield: 0000X000 | CTS | 0 = Low<br>1 = High |
| | | Bitfield: 000X0000 | Overrun Error | 0 = No Error<br>1 = Overrun Error |
| | | Bitfield: 00X00000 | Parity Error | 0 = No Error<br>1 = Parity Error |
| | | Bitfield: 0X000000 | Framing Error | 0 = No Error<br>1 = Framing Error |
| | | Bitfield:X0000000 | DLC established | 0 = No DLC<br>1 = DLC is available |
| | Break Length<br>2 bytes | The length in ms of the detected break. The value 0 is used to indicate that no break has been detected. | | |

### 4.2.8.2 SPP Port Set DTR

**Table 4-69. SPP Port Set DTR Request**

| Description | This command sets the state of the DTR bit. 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. | |
|---|---|---|
| PacketType | REQ | |
| Opcode | SPP_PORT_SET_DTR | |
| DataLength | 2 | |
| Data | LocalPort<br>1 byte | Local RFCOMM port number. Range 1-30 |
| | State | 0: False.<br>1: True. |

**Table 4-70. SPP Port Set DTR Confirm**

| Description | Confirm to the request above. | |
|---|---|---|
| PacketType | CFM | |
| Opcode | SPP_PORT_SET_DTR | |
| DataLength | 2 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_ILLEGAL_STATE_VALUE<br>ERROR_SPP_INVALID_PORT<br>ERROR_SPP_PORT_NOT_OPEN |
| | LocalPort<br>1 byte | Local RFCOMM port number. Range 1-30 |

### 4.2.9 SPP Port Set RTS

Table 4-71. 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.** | |
|---|---|---|
| PacketType | REQ | |
| Opcode | SPP_PORT_SET_RTS | |
| DataLength | 2 | |
| Data | LocalPort<br>1 byte | Local RFCOMM port number. Range 1-30 |
| | State | 0: False.<br>1: True. |

Table 4-72. SPP Port Set RTS Confirm

| **Description** | **Confirm to the request above.** | |
|---|---|---|
| PacketType | CFM | |
| Opcode | SPP_PORT_SET_RTS | |
| DataLength | 1 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_ILLEGAL_STATE_VALUE<br>ERROR_SPP_INVALID_PORT<br>ERROR_SPP_PORT_NOT_OPEN |
| | LocalPort<br>1 byte | Local RFCOMM port number. Range 1-30 |

#### 4.2.9.1 SPP Port Set BREAK

Table 4-73. SPP Set Port Break Request

| **Description** | **This command indicates that the host has detected a break.** | |
|---|---|---|
| PacketType | REQ | |
| Opcode | SPP_PORT_BREAK | |
| DataLength | 2 | |
| Data | LocalPort<br>1 byte | Local RFCOMM port number. Range 1-30 |
| | BreakLength | The length of the break in ms. |

Table 4-74. SPP Set Port Break Confirm

| **Description** | **Confirm to the request above.** |
|---|---|
| PacketType | CFM |
| Opcode | SPP_PORT_BREAK |
| DataLength | 2 |

Table 4-74. SPP Set Port Break Confirm

| | | |
|---|---|---|
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_SPP_INVALID_PORT<br>ERROR_SPP_PORT_NOT_OPEN |
| | LocalPort<br>1 byte | Local RFCOMM port number. Range 1-30 |

#### 4.2.9.2 Set Overrun Error

Table 4-75. SPP Set Overrun Erro Request

| Description | This command is used to indicate that the host has detected an overrun error. This command is only used if the "application" is an UART / UART driver. | |
|---|---|---|
| PacketType | REQ | |
| Opcode | SPP_PORT_OVERRUN_ERROR | |
| DataLength | 1 | |
| Data | LocalPort<br>1 byte | Local RFCOMM port number. Range 1-30 |

Table 4-76. SPP Port Overrun Error Confirm

| Description | Confirm to the request above. | |
|---|---|---|
| PacketType | CFM | |
| Opcode | SPP_PORT_OVERRUN_ERROR | |
| DataLength | 2 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_SPP_INVALID_PORT<br>ERROR_SPP_PORT_NOT_OPEN |
| | LocalPort<br>1 byte | Local RFCOMM port number. Range 1-30 |

#### 4.2.9.3 SPP Set Parity Error

Table 4-77. SPP Port Parity Error Request

| Description | This command is used to indicate that the host has detected a parity error. This command is only used if the "application" is an UART / UART driver. | |
|---|---|---|
| PacketType | REQ | |
| Opcode | SPP_PORT_PARITY_ERROR | |
| DataLength | 1 | |
| Data | LocalPort<br>1 byte | Local RFCOMM port number. Range 1-30 |

Table 4-78. SPP Port Parity Error Confirm

| Description | Confirm to the request above. |
|---|---|
| PacketType | CFM |

**Table 4-78. SPP Port Parity Error Confirm**

| | | |
|---|---|---|
| Opcode | SPP_PORT_PARITY_ERROR | |
| DataLength | 1 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_SPP_INVALID_PORT<br>ERROR_SPP_PORT_NOT_OPEN |
| | LocalPort<br>1 byte | Local RFCOMM port number. Range 1-30 |

### 4.2.9.4 SPP Set Framing Error

**Table 4-79. SPP Port Framing Error Request**

| **Description** | **This command is used to indicate that the host has detected a framing error. This command is only used if the "application" is an UART / UART driver.** | |
|---|---|---|
| PacketType | REQ | |
| Opcode | SPP_PORT_FRAMING_ERROR | |
| DataLength | 1 | |
| Data | LocalPort<br>1 byte | Local RFCOMM port number. Range 1-30 |

**Table 4-80. SPP Port Framing Errors Confirm**

| **Description** | **Confirm to the request above.** | |
|---|---|---|
| PacketType | CFM | |
| Opcode | SPP_PORT_FRAMING_ERROR | |
| DataLength | 2 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_SPP_INVALID_PORT<br>ERROR_SPP_PORT_NOT_OPEN |
| | LocalPort<br>1 byte | Local RFCOMM port number. Range 1-30 |

### 4.2.9.5 SPP Port Status Changed Indicator

**Table 4-81. SPP Port Status Changed Indicator**

| **Description** | **Send from the LMX9820 when remote device has changed the port.** | | | |
|---|---|---|---|---|
| PacketType | IND | | | |
| Opcode | SPP_PORT_STATUS_CHANGED | | | |
| DataLength | 4 | | | |
| Data | LocalPort<br>1 byte | Local RFCOMM port number. Range 1-30 | | |
| | PortStatus<br>1 byte | Bitfield: 0000000X | DTR | 0 = Low<br>1 = High |
| | | Bitfield: 000000X0 | RTS | 0 = Low<br>1 = High |

Table 4-81. SPP Port Status Changed Indicator

| | | | | |
|---|---|---|---|---|
| | | Bitfield: 00000X00 | DSR | 0 = Low<br>1 = High |
| | | Bitfield: 0000X000 | CTS | 0 = Low<br>1 = High |
| | | Bitfield: 000X0000 | Overrun Error | 0 = No Error<br>1 = Overrun Error |
| | | Bitfield: 00X00000 | Parity Error | 0 = No Error<br>1 = Parity Error |
| | | Bitfield: 0X000000 | Framing Error | 0 = No Error<br>1 = Framing Error |
| | | Bitfield:X0000000 | DLC established | 0 = No DLC<br>1 = DLC is available |
| | Break Length<br>2 bytes | The length in ms of the detected break. The value 0 is used to indicate that no break has been detected. | | |

### 4.2.10 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 initialised can be configured with the command Set Ports to Open.

If a RFComm has not been opened, it is not connectable from outside or can not be used for setting up a link.

#### 4.2.10.1 Set Ports To Open

Table 4-82. SDP Set Ports to Open Request

| Description | This command will change which RFCOMM ports the LMX9820 will open both at start-up and runtime. | |
|---|---|---|
| PacketType | REQ | |
| Opcode | SET_PORTS_TO_OPEN | |
| DataLength | 4 | |
| Data | PORTS<br>4 Bytes | This field is a 32-bit mask indicating which RFCOMM ports the LMX9820 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 4-83. SDP Set Ports to Open Confirm

| Description | Confirms the request above | |
|---|---|---|
| PacketType | CFM | |
| Opcode | PORTS_TO_OPEN | |
| DataLength | 1 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_INVALID_PORT<br>ERROR_INVALID_NO_OF_PARAMETERS |

#### 4.2.10.2 Get Ports To Open

Table 4-84. SDP Get Ports to open Request

| Description | This command will get the value of which RFCOMM ports the LMX9820 will open both at start-up and runtime. | |
|---|---|---|
| PacketType | REQ | |
| Opcode | GET_PORTS_TO_OPEN | |
| DataLength | 0 | |
| Data | | |

Table 4-85. SDP Get Ports to open Confirm

| Description | Confirms the request above | |
|---|---|---|
| PacketType | CFM | |
| Opcode | PORTS_TO_OPEN | |
| DataLength | 5 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_INVALID_NO_OF_PARAMETERS |
| | PORTS<br>4 Bytes | This field is a 32-bit mask indicating which RFCOMM ports the LMX9820 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. |

### 4.2.11 Local Service Database Configuration

The LMX9820 allows the modification of the Local Service Discovery Database. On default, the service database contains one entry configured at RFComm port 1 for a Serial Port Profile, Authentication and Encryption enabled.

If the application needs to open a second connection to another device, the dedicated service has to be registered into the database and the registered RFComm port has to be opened (see Section 4.2.10.1). 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 com port enabling and disabling them by needs.

In addition the command set allows to add besides SPP also DUN, FAX, OPP, FTP and SYNC profiles by one command. Each with 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 LMX9820 with a specific record identifier. This identifier is needed for the Enabling or Disabling command to address those specific entries.

#### 4.2.11.1 SDP Enable SDP Record

Table 4-86. SDP Enable SDP Record Request

| Description | This command is used to enable/disable stored SDP records in the LMX9820. | |
|---|---|---|
| PacketType | REQ | |
| Opcode | ENABLE_SDP_RECORD | |
| DataLength | 2 | |
| Data | State<br>1 byte | The new state of the SDP record.<br>0x00 Disable the record.<br>0x01 Enable the record. |

Table 4-86. SDP Enable SDP Record Request

| | | |
|---|---|---|
| | Identifier<br>1 byte | The identifier of the service record to address. This will be received when the record was stored in the LMX9820. |

Table 4-87. SDP Enable SDP Record Confirm

| Description | Confirms the request above | |
|---|---|---|
| PacketType | CFM | |
| Opcode | ENABLE_SDP_RECORD | |
| DataLength | 2 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_ILLEGAL_STATE_VALUE<br>ERROR_IDENTIFIER_OUT_OF_RANGE<br>ERROR_RECORD_ALREADY_IN_SELECTED_STATE<br>ERROR_IDENTIFIER_NOT_IN_USE<br>ERROR_INVALID_NO_OF_PARAMETERS |
| | Identifier<br>1 byte | The identifier received when the record was stored in the LMX9820. |

#### 4.2.11.2 SDP Delete All SDP Records

Table 4-88. SDP Delete SDP Record Request

| Description | This command is used to delete all stored SDP records in the LMX9820. | |
|---|---|---|
| PacketType | REQ | |
| Opcode | DELETE_SDP_RECORDS | |
| DataLength | 0 | |
| Data | None | |

Table 4-89. SDP Delete SDP Record Confirm

| Description | Confirms the request above | |
|---|---|---|
| PacketType | CFM | |
| Opcode | DELETE_SDP_RECORDS | |
| DataLength | 1 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_INVALID_NO_OF_PARAMETERS |

#### 4.2.11.3 SDP Store SPP Record

Table 4-90. SDP Store SPP Record Request

| Description | This command will create a new SPP record in the local SDP database, stored in NVS. |
|---|---|
| PacketType | REQ |
| Opcode | STORE_SPP_RECORD |

**Table 4-90. SDP Store SPP Record Request**

| | | |
|---|---|---|
| DataLength | 4 + NameLength | |
| Data | LocalPort<br>1 byte | The local RFCOMM port used by this service. Range 1 to 30. |
| | Authentication<br>1 byte | 0x00 No authentication requirements<br>0x02 Authentication is only required for this profile for incoming connections.<br>0x20 Authentication is only required for this profile for outgoing connections.<br>0x22 Authentication is required for this profile for connections in both directions |
| | Encryption<br>1 byte | 0x00 No encryption requirements<br>0x04 Encryption is only required for this profile for incoming connections.<br>0x40 Encryption is only required for this profile for outgoing connections.<br>0x44 Encryption is required for this profile for connections in both directions |
| | NameLength<br>1 byte | The number of bytes in the service name including the NULL termination. |
| | ServiceName<br><NameLength> bytes | A text string containing a human-readable name for the service. Must be NULL terminated and must maximum contain 15 bytes. The string is placed at language base 0x0100. |

**Table 4-91. SDP Store SPP Record Confirm**

| **Description** | **Confirms the request above** | |
|---|---|---|
| PacketType | CFM | |
| Opcode | STORE_SPP_RECORD | |
| DataLength | 2 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_INVALID_PORT<br>ERROR_NAME_TOO_LONG<br>ERROR_INVALID_AUTHENTICATION_VALUE<br>ERROR_INVALID_ENCRYPTION_VALUE<br>ERROR_MAXIMUM_NO_OF_SERVICE_RECORDS_REACHED<br>ERROR_WRITING_TO_NVS |
| | Identifier<br>1 byte | A unique identifier, which must be used when dynamic enabling/disabling the record. |

#### 4.2.11.4 SDP Store DUN Record

**Table 4-92. SDP Store DUN Record Request**

| **Description** | **This command will create a new DUN record in the local SDP database, stored in NVS.** |
|---|---|
| PacketType | REQ |
| Opcode | STORE_DUN_RECORD |
| DataLength | 4 + <Length> |

Table 4-92. SDP Store DUN Record Request

| | | |
|---|---|---|
| Data | LocalPort<br>1 byte | The local RFCOMM port used by this service. Range 1 to 30. |
| | Authentication<br>1 byte | 0x00 No authentication requirements<br>0x02 Authentication is only required for this profile for incoming connections.<br>0x20 Authentication is only required for this profile for outgoing connections.<br>0x22 Authentication is required for this profile for connections in both directions |
| | Encryption<br>1 byte | 0x00 No encryption requirements<br>0x04 Encryption is only required for this profile for incoming connections.<br>0x40 Encryption is only required for this profile for outgoing connections.<br>0x44 Encryption is required for this profile for connections in both directions |
| | NameLength<br>1 byte | The number of bytes in the service name including the NULL termination. |
| | ServiceName<br><NameLength> Bytes | A text string containing a human-readable name for the service. Must be NULL terminated and must maximum contain 15 bytes. The string is placed at language base 0x0100. |

Table 4-93. SDP Store DUN Record Confirm

| Description | Confirms the request above | |
|---|---|---|
| PacketType | CFM | |
| Opcode | STORE_DUN_RECORD | |
| DataLength | 2 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_INVALID_PORT<br>ERROR_NAME_TOO_LONG<br>ERROR_INVALID_AUTHENTICATION_VALUE<br>ERROR_INVALID_ENCRYPTION_VALUE<br>ERROR_MAXIMUM_NO_OF_SERVICE_RECORDS_REACHED<br>ERROR_WRITING_TO_NVS |
| | Identifier<br>1 byte | A unique identifier, which must be used when dynamic enabling/disabling the record. |

#### 4.2.11.5 SDP Store FAX Record

Table 4-94. SDP Store Fax Record Request

| Description | This command will create a new FAX record in the local SDP database, stored in NVS. |
|---|---|
| PacketType | REQ |

Table 4-94. SDP Store Fax Record Request

| | | |
|---|---|---|
| Opcode | STORE_FAX_RECORD | |
| DataLength | 7 + <NameLength> | |
| Data | LocalPort<br>1 byte | The local RFCOMM port used by this service. Range 1 to 30. |
| | FaxClass1<br>1 byte | Fax Class 1 Support.<br>0x00 = False<br>0x01 = True |
| | FaxClass20<br>1 byte | Fax Class 2.0 Support.<br>0x00 = False<br>0x01 = True |
| | FaxClass2<br>1 byte | Fax Class 2 Support.<br>0x00 = False<br>0x01 = True |
| | Authentication<br>1 byte | 0x00 No authentication requirements<br>0x02 Authentication is only required for this profile for incoming connections.<br>0x20 Authentication is only required for this profile for outgoing connections.<br>0x22 Authentication is required for this profile for connections in both directions |
| | Encryption<br>1 byte | 0x00 No encryption requirements<br>0x04 Encryption is only required for this profile for incoming connections.<br>0x40 Encryption is only required for this profile for outgoing connections.<br>0x44 Encryption is required for this profile for connections in both directions |
| | NameLength<br>1 byte | The number of bytes in the service name including the NULL termination. |
| | ServiceName<br><NameLength> Bytes | A text string containing a human-readable name for the service. Must be NULL terminated. The string is placed at language base 0x0100. |

Table 4-95. SDP Store FAX Record Confirm

| Description | Confirms the request above |
|---|---|
| PacketType | CFM |
| Opcode | STORE_FAX_RECORD |
| DataLength | 2 |

Table 4-95. SDP Store FAX Record Confirm

| | | |
|---|---|---|
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_INVALID_PORT<br>ERROR_NAME_TOO_LONG<br>ERROR_INVALID_AUTHENTICATION_VALUE<br>ERROR_INVALID_ENCRYPTION_VALUE<br>ERROR_MAXIMUM_NO_OF_SERVICE_RECORDS_REACHED<br>ERROR_WRITING_TO_NVS<br>ERROR_INVALID_SUPPORTED_FAXCLASS_VALUE |
| | Identifier<br>1 byte | A unique identifier, which must be used when dynamic enabling/disabling the record. |

### 4.2.11.6 SDP Store OPP Record

Table 4-96. SDP Store OPP Record Request

| Description | This command will create a new OPP record in the local SDP database, stored in NVS. | |
|---|---|---|
| PacketType | REQ | |
| Opcode | STORE_OPP_RECORD | |
| DataLength | 5 + <NameLength> + <NoOfFormats> | |
| Data | LocalPort<br>1 byte | The local RFCOMM port used by this service. Range 1 to 30. |
| | Authentication<br>1 byte | 0x00 No authentication requirements<br>0x02 Authentication is only required for this profile for incoming connections.<br>0x20 Authentication is only required for this profile for outgoing connections.<br>0x22 Authentication is required for this profile for connections in both directions |
| | Encryption<br>1 byte | 0x00 No encryption requirements<br>0x04 Encryption is only required for this profile for incoming connections.<br>0x40 Encryption is only required for this profile for outgoing connections.<br>0x44 Encryption is required for this profile for connections in both directions |
| | NoOfFormats<br>1 byte | The number of bytes in the supported format field. Range 1 to 7. |
| | SupportedFormats<br><NoOfFormats> bytes | A list of supported formats, see profile specification for further details. |
| | NameLength<br>1 byte | The number of bytes in the service name including the NULL termination. |
| | ServiceName<br><NameLength> Bytes | A text string containing a human-readable name for the service. Must be NULL terminated. The string is placed at language base 0x0100. |

Table 4-97. SDP Store OPP Record Confirm

| Description | Confirms the request above | |
|---|---|---|
| PacketType | CFM | |
| Opcode | STORE_OPP_RECORD | |
| DataLength | 2 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_INVALID_PORT<br>ERROR_NAME_TOO_LONG<br>ERROR_INVALID_AUTHENTICATION_VALUE<br>ERROR_INVALID_ENCRYPTION_VALUE<br>ERROR_MAXIMUM_NO_OF_SERVICE_RECORDS_REACHED<br>ERROR_WRITING_TO_NVS<br>ERROR_TOO_ MANY_SUPPORTED_FORMATS |
| | Identifier<br>1 byte | A unique identifier, which must be used when dynamic enabling/disabling the record. |

#### 4.2.11.7 SDP Store FTP Record

Table 4-98. SDP Store FTP Record Request

| Description | This command will create a new FTP record in the local SDP database, stored in NVS. | |
|---|---|---|
| PacketType | REQ | |
| Opcode | STORE_FTP_RECORD | |
| DataLength | 4 + <NameLength> | |
| Data | LocalPort<br>1 byte | The local RFCOMM port used by this service. Range 1 to 30. |
| | Authentication<br>1 byte | 0x00 No authentication requirements<br>0x02 Authentication is only required for this profile for incoming connections.<br>0x20 Authentication is only required for this profile for outgoing connections.<br>0x22 Authentication is required for this profile for connections in both directions |
| | Encryption<br>1 byte | 0x00 No encryption requirements<br>0x04 Encryption is only required for this profile for incoming connections.<br>0x40 Encryption is only required for this profile for outgoing connections.<br>0x44 Encryption is required for this profile for connections in both directions |
| | NameLength<br>1 byte | The number of bytes in the service name including the NULL termination. |
| | ServiceName<br><Length> Bytes | A text string containing a human-readable name for the service. Must be NULL terminated. The string is placed at language base 0x0100. |

Table 4-99. SDP Store FTP Record Confirm

| Description | Confirms the request above | |
|---|---|---|
| PacketType | CFM | |
| Opcode | STORE_FTP_RECORD | |
| DataLength | 2 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_INVALID_PORT<br>ERROR_NAME_TOO_LONG<br>ERROR_INVALID_AUTHENTICATION_VALUE<br>ERROR_INVALID_ENCRYPTION_VALUE<br>ERROR_MAXIMUM_NO_OF_SERVICE_RECORDS_REACHED<br>ERROR_WRITING_TO_NVS |
| | Identifier<br>1 byte | A unique identifier, which must be used when dynamic enabling/disabling the record. |

#### 4.2.11.8 SDP Store IrMCSync Record

Table 4-100. SDP Store SYNC Record Request

| Description | This command will create a new IrMCSync record in the local SDP database, stored in NVS. | |
|---|---|---|
| PacketType | REQ | |
| Opcode | STORE_SYNC_RECORD | |
| DataLength | 5+ <NameLength> + <NoOfDataStores> | |
| Data | LocalPort<br>1 byte | The local RFCOMM port used by this service. Range 1 to 30. |
| | Authentication<br>1 byte | 0x00 No authentication requirements<br>0x02 Authentication is only required for this profile for incoming connections.<br>0x20 Authentication is only required for this profile for outgoing connections.<br>0x22 Authentication is required for this profile for connections in both directions |
| | Encryption<br>1 byte | 0x00 No encryption requirements<br>0x04 Encryption is only required for this profile for incoming connections.<br>0x40 Encryption is only required for this profile for outgoing connections.<br>0x44 Encryption is required for this profile for connections in both directions |
| | NoOfDataStores<br>1 byte | The number of bytes in the supported data stores. Range 1 to 5 |

**Table 4-100. SDP Store SYNC Record Request**

| | | |
|---|---|---|
| | SupportedDataStores<br><NoOfDataStores> bytes | A list of supported data stores, see profile specification for further details. |
| | NameLength<br>1 byte | The number of bytes in the service name including the NULL termination. |
| | ServiceName<br><NameLength> Bytes | A text string containing a human-readable name for the service. Must be NULL terminated. The string is placed at language base 0x0100. |

**Table 4-101. SDP Store SYNC Record Confirm**

| Description | Confirms the request above | |
|---|---|---|
| PacketType | CFM | |
| Opcode | STORE_SYNC_RECORD | |
| DataLength | 2 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_INVALID_PORT<br>ERROR_NAME_TOO_LONG<br>ERROR_INVALID_AUTHENTICATION_VALUE<br>ERROR_INVALID_ENCRYPTION_VALUE<br>ERROR_MAXIMUM_NO_OF_SERVICE_RECORDS_REACHED<br>ERROR_WRITING_TO_NVS<br>ERROR_TOO_ MANY_DATASTORES |
| | Identifier<br>1 byte | A unique identifier, which must be used when dynamic enabling/disabling the record. |

### 4.2.12 Local Bluetooth Settings

The LMX9820 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.

#### 4.2.12.1 Read Local Name

**Table 4-102. Read Local Name Request**

| Description | Request the user-friendly name for the local Bluetooth device. | |
|---|---|---|
| PacketType | REQ | |
| Opcode | GAP_READ_LOCAL_NAME | |
| DataLength | 0 | |
| Data | None | |

**Table 4-103. Read Local Name Confirm**

| Description | Confirm to the request above. |
|---|---|
| PacketType | CFM |
| Opcode | GAP_READ_LOCAL_NAME |
| DataLength | 2 + NameLength |

Table 4-103. Read Local Name Confirm

| | | |
|---|---|---|
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_INVALID_NO_OF_PARAMETERS<br>ERROR_UNKNOWN_ERROR |
| | NameLength<br>1 byte | Number of bytes in device name |
| | DeviceName<br>NameLength bytes | The user-friendly name of the local device.<br>The string is NULL terminated. Max length is 30 bytes. |

#### 4.2.12.2 Write Local Name

Table 4-104. Write Local Name Request

| **Description** | **Change the user-friendly name for the local Bluetooth device. The name is stored in NVS** | |
|---|---|---|
| PacketType | REQ | |
| Opcode | GAP_WRITE_LOCAL_NAME | |
| DataLength | 1+ NameLength | |
| Data | NameLength<br>1 byte | Number of bytes in device name |
| | DeviceName<br>Length bytes | The user-friendly name of the local device.<br>(String must be NULL terminated). Max length is 30 bytes. |

Table 4-105. Write Local Name Confirm

| **Description** | **Confirm to the request above.** | |
|---|---|---|
| PacketType | CFM | |
| Opcode | GAP_WRITE_LOCAL_NAME | |
| DataLength | 1 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_NAME_TOO_LONG<br>ERROR_INVALID_NO_OF_PARAMETERS |

#### 4.2.12.3 Read Local Bluetooth Address

Table 4-106. Read Local BD_Addr Request

| **Description** | **Read the Bluetooth device address of the local Bluetooth device.** | |
|---|---|---|
| PacketType | REQ | |
| Opcode | GAP_READ_LOCAL_BDA | |
| DataLength | 0 | |
| Data | None | |

Table 4-107. Read Local BD_Addr Confirm

| **Description** | **Confirm to the request above.** |
|---|---|

Table 4-107. Read Local BD_Addr Confirm

| | | |
|---|---|---|
| PacketType | CFM | |
| Opcode | GAP_READ_LOCAL_BDA | |
| DataLength | 7 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_INVALID_NO_OF_PARAMETERS<br>ERROR_UNKNOWN_ERROR |
| | BdAddr<br>6 byte | Bluetooth device address for the local device |

#### 4.2.12.4 Change Local Bluetooth Address

Table 4-108. Change Local BD_Address Request

| **Description** | **The LMX9820 will change the local BdAddress with the one specified in this command. If the new BdAddress has the value 0xFFFFFFFFFFFF, the LMX9820 will use the default Bluetooth address specified by NSC and return ERROR_RESET_TO_NSC_BDADDRESS. A delay of at least 1 sec should be inserted between this command and the reset operation.** | |
|---|---|---|
| PacketType | REQ | |
| Opcode | CHANGE_LOCAL_BDADDRESS | |
| DataLength | 6 | |
| Data | BdAddress<br>6 Bytes | The new Bluetooth address for the local device. The address is stored in NVS. |

Table 4-109. Change Local BD_Addr Confirm

| **Description** | **Confirms the request above** | |
|---|---|---|
| PacketType | CFM | |
| Opcode | CHANGE_LOCAL_BDADDRESS | |
| DataLength | 1 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_RESET_TO_NSC_BDADDRESS<br>ERROR_INVALID_NO_OF_PARAMETERS |

#### 4.2.12.5 Store Class of Device

Table 4-110. Store Class of Device Request

| **Description** | **This command will store the class of device for the LMX9820 in NVS. The proper value for the class of device parameter is specified by the Bluetooth SIG.** | |
|---|---|---|
| PacketType | REQ | |
| Opcode | STORE_CLASS_OF_DEVICE | |
| DataLength | 3 | |
| Data | ClassOfDevice<br>3 bytes | The class of device value to be stored in NVS. |

**Table 4-111. Store Class of Device Confirm**

| **Description** | **Confirms the request above** | |
|---|---|---|
| PacketType | CFM | |
| Opcode | STORE_CLASS_OF_DEVICE | |
| DataLength | 1 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_INVALID_NO_OF_PARAMETERS |

### 4.2.12.6 Set Scan Mode

**Table 4-112. Set Scan Mode Request**

| **Description** | **Change the Bluetooth scan mode.**<br>**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 LMX9820 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.**<br>**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.** | |
|---|---|---|
| PacketType | REQ | |
| Opcode | GAP_SET_SCANMODE | |
| DataLength | 2 | |
| Data | Connectability<br>1 byte | 0 = Not connectable<br>1 = Connectable |
| | Discoverability<br>1 byte | 0 = Non discoverable<br>1 = General discoverable<br>2 = Limited discoverable<br>3 = Automatic limited discoverable mode (see Bluetooth GAP profile) |

**Table 4-113. Set Scan Mode Confirm**

| **Description** | **Confirm to the request above.** | |
|---|---|---|
| PacketType | CFM | |
| Opcode | GAP_SET_SCANMODE | |
| DataLength | 1 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_INVALID_CONNECTABILITY_PARAMETER<br>ERROR_INVALID_DISCOVERABILITY_PARAMETER<br>ERROR_INVALID_NO_OF_PARAMETERS<br>ERROR_UNKNOWN_ERROR |

#### 4.2.12.7 Automatic Limited Discoverable Mode Ended

**Table 4-114. Automatic limited discoverable ended Indicator**

| **Description** | **Indication send from the device, when automatic limited discoverable mode has ended.** | |
|---|---|---|
| PacketType | IND | |
| Opcode | GAP_SET_SCANMODE | |
| DataLength | 1 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_UNKNOWN_ERROR |

#### 4.2.12.8 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 LMX9820 tries to switch its role to master if connected from another device. For this the LMX9820 is sending a Master/Slave switch request after link establishment to the remote device. If the switch is successfull, the link setup will be continued and the LMX9820 is open for other incoming connections.
If the switch fails, the link will be dropped.

This functionality allows Access Point like applications on the LMX9820.

**Table 4-115. Force Master Role Request**

| **Description** | **This command will change the preferred role of the LMX9820 e.g. to force master role at any connections. The LMX9820 must be reset to let the changes take effect. A delay of at least 1 sec should be inserted between this command and the reset operation.** | |
|---|---|---|
| PacketType | REQ | |
| Opcode | FORCE_MASTER_ROLE | |
| DataLength | 1 | |
| Data | Role<br>1 bytes | 0x00 Don't care about role<br>0x01 Force master role at connection setup |

**Table 4-116. Force Master Role Confirm**

| **Description** | **Confirms the request above** | |
|---|---|---|
| PacketType | CFM | |
| Opcode | FORCE_MASTER_ROLE | |
| DataLength | 1 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_INVALID_ROLE<br>ERROR_INVALID_NO_OF_PARAMETERS |

#### 4.2.12.9 Read Operation Mode

The Operation Mode of the LMX9820 controls the behavious in different situations. Please see Section 1.6 for details.

**Table 4-117.**

| **Description** | **This command will read out the current operation mode.** |
|---|---|
| PacketType | REQ |
| Opcode | READ_OPERATION_MODE |

Table 4-117.

| | | |
|---|---|---|
| DataLength | 0 | |
| Data | None | |

Table 4-118. Read Operation Mode Confirm

| **Description** | **Confirms the request above** | |
|---|---|---|
| PacketType | CFM | |
| Opcode | READ_OPERATION_MODE | |
| DataLength | 2 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_INVALID_NO_OF_PARAMETERS |
| | Mode<br>1 Byte | 0x00 Command Mode<br>0x01 Automatic Mode |

#### 4.2.12.10 Write Operation Mode

Table 4-119. Write Operation Mode Request

| **Description** | **This command will change the operation mode stored in NVS. The new setting will take effect after a reset.** | |
|---|---|---|
| PacketType | REQ | |
| Opcode | WRITE_OPERATION_MODE | |
| DataLength | 1 | |
| Data | Mode<br>1 Byte | 0x00 Command Mode<br>0x01 Automatic Mode |

Table 4-120. Write Operation Mode Confirm

| **Description** | **Confirms the request above** | |
|---|---|---|
| PacketType | CFM | |
| Opcode | WRITE_OPERATION_MODE | |
| DataLength | 1 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_INVALID_NO_OF_PARAMETERS<br>ERROR_INVALID_MODE |

### 4.2.13 ACL Indications

If the Event Filter is set to “Report all events”, the LMX9820 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 4-154 "ACL Error Codes" on page 88.

#### 4.2.13.1 ACL Established

Table 4-121. ACL Established Indicator

| **Description** | **This indication is sent to the host when an ACL link is established.** |
|---|---|
| PacketType | IND |
| Opcode | GAP_ACL_ESTABLISHED |

**Table 4-121. ACL Established Indicator**

| | | |
|---|---|---|
| DataLength | 7 | |
| Data | BdAddr<br>6 bytes | The Bluetooth address of the remote device. |
| | Status<br>1 byte | See “ACL Error Codes” on page 88 |

#### 4.2.13.2 ACL Terminated

**Table 4-122. ACL Terminated Indicator**

| **Description** | **This indication is sent to the host when an ACL link is terminated.** | |
|---|---|---|
| PacketType | IND | |
| Opcode | GAP_ACL_TERMINATED | |
| DataLength | 8 | |
| Data | BdAddr<br>6 bytes | The Bluetooth address of the remote device. |
| | Status<br>1 byte | See “ACL Error Codes” on page 88 |

### 4.2.14 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 LMX9820 on default is in Security Mode 2.

#### 4.2.14.1 Get Security Mode

**Table 4-123. Get Security Mode Request**

| **Description** | **Reads the current security mode of the Bluetooth device.** | |
|---|---|---|
| PacketType | REQ | |
| Opcode | GAP_GET_SECURITY_MODE | |
| DataLength | 0 | |
| Data | None | |

Table 4-124.  Get Security Mode Confirm

| Description | Confirm to the request above. | |
|---|---|---|
| PacketType | CFM | |
| Opcode | GAP_GET_SECURITY_MODE | |
| DataLength | 2 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_INVALID_NO_OF_PARAMETERS |
| | Mode<br>1 byte | The current Bluetooth security mode.<br>0x01 Security mode 1<br>0x02 Security mode 2<br>0x03 Security mode 3<br>0x83 Security mode 3 with link level encryption |

### 4.2.14.2   Set Security Mode

Table 4-125.  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. | |
|---|---|---|
| PacketType | REQ | |
| Opcode | GAP_SET_SECURITY_MODE | |
| DataLength | 1 | |
| Data | Mode<br>1 byte | The current Bluetooth security mode.<br>0x01 Security mode 1<br>0x02 Security mode 2<br>0x03 Security mode 3<br>0x83 Security mode 3 with link level encryption |

Table 4-126.  Set Security Mode Confirm

| Description | Confirm to the request above. | |
|---|---|---|
| PacketType | CFM | |
| Opcode | GAP_SET_SECURITY_MODE | |
| DataLength | 1 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_INVALID_NO_OF_PARAMETERS<br>ERROR_INVALID_SECURITY_MODE |

### 4.2.15   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 LMX9820 has a fixed PinCode which can be changed with the command “Change fixed Pin”. This pin is used during any pairing procedure.

The following commands allow to remove those stored link keys or change the PINCode within the device.

#### 4.2.15.1 Remove Pairing

Table 4-127. Remove Pairing Command

| Description | Remove pairing with a remote device. | |
|---|---|---|
| PacketType | REQ | |
| Opcode | GAP_REMOVE_PAIRING | |
| DataLength | 6 | |
| | BdAddress<br>6 byte | Remove pairing to the BdAddress. |

Table 4-128. Remove Paired Device Confirm

| Description | Response to the request above. | |
|---|---|---|
| PacketType | CFM | |
| Opcode | GAP_REMOVE_PAIRING | |
| DataLength | 1 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_INVALID_NO_OF_PARAMETERS<br>ERROR_LINKKEY_DOES_NOT_EXISTS |

#### 4.2.15.2 List Paired Devices

Table 4-129. List Paired Devices Command

| Description | Request a list of paired devices from NVS |
|---|---|
| PacketType | REQ |
| Opcode | GAP_LIST_PAIRED_DEVICES |
| DataLength | 0 |

Table 4-130. List paired devices Confirm

| Description | Response to the request above. | |
|---|---|---|
| PacketType | CFM | |
| Opcode | LIST_PAIRED_DEVICES | |
| DataLength | 2 +6 * DeviceCount | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_INVALID_NO_OF_PARAMETERS |
| | DeviceCount<br>1 byte | Number of devices in the list of paired devices<br>If 0 the device is not paired to any other devices.<br>The maximum number of paired devices is 7. |
| | BdAddresses<br>6 byte * DeviceCount | The list of paired devices |

#### 4.2.15.3 Get Fixed Pin

Table 4-131. Get Fixed Pin Request

| Description | Reads the current fixed pin code from NVS |
|---|---|
| PacketType | REQ |
| Opcode | GAP_GET_FIXED_PIN |
| DataLength | 0 |

Table 4-132. Get Fixed Pin Confirm

| Description | Response to the request above. | |
|---|---|---|
| PacketType | CFM | |
| Opcode | GAP_GET_ FIXED _PIN | |
| DataLength | 2+Pinlength | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_INVALID_NO_OF_PARAMETERS |
| | Pinlength<br>1 byte | Length of pin code, in bytes.<br>Range: 0x01-0x10 |
| | Pincode<br>Pinlength bytes | PIN code used when the two Bluetooth devices are paired. The maximum length of a PIN code is 128 bits (16 bytes). |
| | | |

#### 4.2.15.4 Set Fixed Pin

Table 4-133. Set Fixed Pin Request

| Description | Stores a new fixed pin code in NVS | |
|---|---|---|
| PacketType | REQ | |
| Opcode | GAP_SET_FIXED_PIN | |
| DataLength | 1+ Pinlength | |
| | Pinlength<br>1 byte | Length of pin code, in bytes.<br>Range: 0x01-0x10 |
| | Pincode<br>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 4-134. Set Fixed Pin Confirm

| Description | Response to the request above. | |
|---|---|---|
| PacketType | CFM | |
| Opcode | GAP_SET_ FIXED _PIN | |
| DataLength | 1 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_INVALID_NO_OF_PARAMETERS<br>ERROR_PINCODE_TOO_LONG |

### 4.2.16 Local Hardware Configuration

The LMX9820 offers a set of commands to configure general hardware specific parameters.

#### 4.2.16.1 Set Event Filter

Table 4-135. Set Event Filter Command

| Description | This command is used to set the event filter. The setting is stored in NVS. | |
|---|---|---|
| PacketType | REQ | |
| Opcode | SET_EVENT_FILTER | |
| DataLength | 1 | |
| Data | Filter<br>1 byte | 0x00: All events reported<br>0x01: No ACL Link Indicators (default)<br>0x02: No events reported |

Table 4-136. Set Event Filter Confirm

| Description | Response to the request above. | |
|---|---|---|
| PacketType | CFM | |
| Opcode | SET_EVENT_FILTER | |
| DataLength | 1 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_INVALID_NO_OF_PARAMETERS<br>ERROR_LIMIT |

#### 4.2.16.2 Get Event Filter

Table 4-137. Get Event Filter Command

| Description | This command reads the current event filter setting. |
|---|---|
| PacketType | REQ |
| Opcode | GET_EVENT_FILTER |
| DataLength | 0 |

Table 4-138. Get Event Filter Confirm

| Description | Response to the request above. | |
|---|---|---|
| PacketType | CFM | |
| Opcode | GET_EVENT_FILTER | |
| DataLength | 1 | |
| Data | Filter<br>1 byte | 0x00: All events reported<br>0x01: No ACL Link Indicators (default)<br>0x02: No events reported |

#### 4.2.16.3 Reset

Table 4-139. Reset Request

| **Description** | **This command will perform a soft reset of the LMX9820. The LMX9820 will send a LMX9820_READY indication when it has performed the reset.<br>Note: A delay of at least 1 sec should be inserted between this NVS write commands and the reset operation.** | |
|---|---|---|
| PacketType | REQ | |
| Opcode | RESET | |
| DataLength | 0 | |
| Data | None | |

#### 4.2.16.4 LMX9820 Ready

Table 4-140. Simply Blue Ready Indicator

| **Description** | **The LMX9820 will send this indication to the host when the device is fully initialised and ready to receive commands from the host.** | |
|---|---|---|
| PacketType | IND | |
| Opcode | DEVICE_READY | |
| DataLength | 1+Length | |
| Data | Length<br>1 byte | Number of bytes in software version string |
| | version<br>&lt;Length&gt; bytes | ASCII string containing the software version. e.g. "0200" indicating that the software version is version 2.00. |

#### 4.2.16.5 Restore Factory Settings

Table 4-141. Restore Factory Settings Request

| **Description** | **This command will restore the LMX9820 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 "LMX9820 System Parameters" on page 5.<br>The LMX9820 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.** | |
|---|---|---|
| PacketType | REQ | |
| Opcode | RESTORE_FACTORY_SETTINGS | |
| DataLength | 0 | |
| Data | none | |

Table 4-142.

| **Description** | **Confirms the request above** |
|---|---|
| PacketType | CFM |
| Opcode | RESTORE_FACTORY_SETTINGS |
| DataLength | 1 |

**Table 4-142.**

| | | |
|---|---|---|
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_INVALID_NO_OF_PARAMETERS |

### 4.2.16.6 Change NVS UART Speed

The UART Speed is only active if the ISEL pin configuration of the LMX9820 is set to “Check NVS settings”. Changed UART speeds get active after reset.

**Table 4-143. 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. A delay of at least 1 sec should be inserted between this command and the reset operation.** | |
|---|---|---|
| PacketType | REQ | |
| Opcode | CHANGE_NVS_UART_SPEED | |
| DataLength | 1 | |
| Data | UartSpeed<br>1 Byte | The UART speed to be stored in NVS<br>0x00 = 2400<br>0x01 = 4800<br>0x02 = 7200<br>0x03 = 9600<br>0x04 = 19200<br>0x05 = 38400<br>0x06 = 57600<br>0x07 = 115200<br>0x08 = 230400<br>0x09 = 460800<br>0x0A = 921600 |

**Table 4-144. Change UART Speed Confirm**

| **Description** | **Confirms the request above** | |
|---|---|---|
| PacketType | CFM | |
| Opcode | CHANGE_NVS_UART_SPEED | |
| DataLength | 1 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_UART_SPEED_OUT_OF_RANGE<br>ERROR_INVALID_NO_OF_PARAMETERS |

### 4.2.16.7 Change UART Settings

UART Settings will be stored in NVS and are valid for ALL UART speeds at LMX9820, except for ISEL pin selected 9600baud. That special settings always uses No parity and One Stop bit.

**Table 4-145. 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. A delay of at least 1 sec should be inserted between this command and the reset operation.** |
|---|---|

**Table 4-145. Change UART Settings Request**

| | | |
|---|---|---|
| PacketType | REQ | |
| Opcode | CHANGE_UART_SETTINGS | |
| DataLength | 2 | |
| Data | PartityBit<br>1 Byte | 0x00 None<br>0x01 Odd<br>0x02 Even |
| | StopBits<br>1 Byte | 0x00 One Stop bit<br>0x01 Two Stop bits |

**Table 4-146. Change UART Settings Confirm**

| **Description** | **Confirms the request above** | |
|---|---|---|
| PacketType | CFM | |
| Opcode | CHANGE_UART_SETTINGS | |
| DataLength | 1 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_PARITY_BIT_OUT_OF_RANGE,<br>ERROR_STOP_BITS_OUT_OF_RANGE,<br>ERROR_INVALID_NO_OF_PARAMETERS |

### 4.2.16.8 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 4-147. 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.** | |
|---|---|---|
| PacketType | REQ | |
| Opcode | TEST_MODE | |
| DataLength | 1 | |
| Data | Mode<br>1 byte | The Bluetooth test mode to enter:<br>0x01 Enable Bluetooth test mode<br>0x02 Enable local loopback mode |

**Table 4-148. Initiate Bluetooth Test Mode Confirm**

| **Description** | **Confirms the request above** |
|---|---|
| PacketType | CFM |
| Opcode | TEST_MODE |
| DataLength | 1 |

Table 4-148.  Initiate Bluetooth Test Mode Confirm

| | | |
|---|---|---|
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_ILLEGAL_TESTMODE<br>ERROR_UNKNOWN_ERROR<br>ERROR_INVALID_NO_OF_PARAMETERS |

#### 4.2.16.9 Initiate RF Test Mode

In addition to the standard Bluetooth Test Mode the LMX9820 offers a special RF Test Mode, switching the transmitter into a continous transmit mode. This is needed for Bluetooth qualification as well as regulatory testings for FCC and ETSI and will be needed for country specific qualification.

This mode can only be left with a hardware reset.

Table 4-149.  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.** | |
|---|---|---|
| PacketType | REQ | |
| Opcode | RF_TEST_MODE | |
| DataLength | 13 | |
| Data | Test<br>1 byte | Test Scenario:<br>0: Stop transmit<br>1: Burst transmit, take payload from this HCI Command<br>2: Burst transmit, use PRBS-9 sequence |
| | Channel<br>1 byte | Channel number 0 – 78<br>0: 2402 MHz<br>78: 2480 MHz<br>255 (0xFF): Hopping in connection state |
| | PaCtrl<br>1 byte | Reserved for future use |

Table 4-149. Initiate RF Test Mode Request

| | | |
|---|---|---|
| | ModulationCtrl<br>1 byte | Modulation Control:<br>0: No modulation<br>1: Access code only (68 us TX data every 1250 us – note 1)<br>2: Bluetooth DH1 packet with defined payload (note 2)<br>3: Bluetooth DM1 packet (17 bytes)<br>4: Bluetooth DH1 packet (27 bytes)<br>5: Invalid<br>6: Invalid<br>7: Bluetooth HV3 packet (30 bytes)<br>8: Invalid<br>9: Bluetooth AUX1 packet (29 bytes)<br>10: Bluetooth DM3 packet (121 bytes)<br>11: Bluetooth DH3 packet (183 bytes)<br>12: Invalid<br>13: Invalid<br>14: Bluetooth DM5 packet (224 bytes)<br>15: Bluetooth DH5 packet (339 bytes) |
| | ModulationPattern<br>8 bytes | Defines Access Code modulation if ModulationCtrl = 1, else ignored. |
| | ModulationPayload<br>1 byte | Defines one byte of payload repeated through packet if<br>Test = 1, else ignored |

Table 4-150. Initiate RF Test Mode Confirm

| Description | Confirms the request above | |
|---|---|---|
| PacketType | CFM | |
| Opcode | RF_TEST_MODE | |
| DataLength | 1 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_INVALID_NO_OF_PARAMETERS |

#### 4.2.16.10 Read RSSI

Table 4-151. Read RSSI Request

| Description | This command will read out the current RSSI value for given link a remote device. | |
|---|---|---|
| PacketType | REQ | |
| Opcode | READ_RSSI | |
| DataLength | 6 | |
| Data | BdAddress<br>6 byte | The link to the bdaddress for which to read out the current RSSI value. |

Table 4-152. Read RSSI Confirm

| Description | Confirms the request above |
|---|---|
| PacketType | CFM |
| Opcode | READ_RSSI |

**Table 4-152. Read RSSI Confirm**

| | | |
|---|---|---|
| DataLength | 2 | |
| Data | Status<br>1 byte | ERROR_OK<br>ERROR_UNKNOWN_ERROR<br>ERROR_INVALID_NO_OF_PARAMETERS |
| | RSSI<br>1 byte | The RSSI value<br>Range: -128 <= N <= 127<br>Units: dB |

#### 4.2.16.11 Firmware Upgrade

The LMX9820 offers a special code to update the firmware of the UART interface. The ISP (In-System-Programming) Code is located in a write protected area of the memory.

ISP can either be activated by setting the pin ENV0 to 0 and reset, or by sending the Firmware Upgrade command.

As the chip will leave the standard command interface after that command, it is not confirmed.

**Table 4-153. Firmware Upgrade Request**

| Description | This command is used to start the ISP code for firmware upgrade. | |
|---|---|---|
| PacketType | REQ | |
| Opcode | FIRMWARE_UPGRADE | |
| DataLength | 0 | |
| Data | | |

## 4.3 ERROR CODES

### 4.3.1 ACL Error Codes

The following table is copied out Bluetooth Specification 1.1 Part H6.1.

**Table 4-154. 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. |

Table 4-154. ACL Error Codes

| | |
|---|---|
| 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. |
| 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-0xFF | Reserved for Future Use. |

### 4.3.2 Generic error codes

Table 4-155. 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. |

Table 4-155. Generic Error Codes

| | | |
|---|---|---|
| 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. |
| 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 LMX9820. The answer is truncated |
| 0x0d | ERROR_RESULT_TOO_LARGE | The SDP result from the remote device is too large to be handled by the LMX9820 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 LMX9820 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 LMX9820 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. |

**Table 4-155. Generic Error Codes**

| | | |
|---|---|---|
| 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. |
| 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 |

### 4.3.3 RFCOMM Error Codes

**Table 4-156. 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 |

### 4.3.4 RFCOMM Release Reasons

**Table 4-157. 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. |

## 5.0 Revision History

This is a report of the revision/creation process of the LMX9820 Bluetooth Serial Port Module - Software Users Guide. Any revisions (i.e., additions, deletions, parameter corrections, etc.) are recorded in the table(s) below.

| **Revision #<br>(PDF Date)** | **Revisions / Comments** |
|---|---|
| 1.0 (05/16/03) | Initial release. |
| 1.1 (06/20/03) | • Updated Table 0-1<br>— Changed SW Version to 5.05<br>— Changed SBC Version to 1.2.0.1<br>• Updated Table 1-1 "LMX9820 System Parameters"<br>— Added Event Filter<br>— Added Default Link Policy Settings<br>• Added Section 3.1.1.7 "Event Filter"<br>• Updated Section 3.3 "Low Power Modes"<br>— Changed Content<br>— Added Section 3.3.1.1 "Default Link Policy Setting"<br>• Added the following commands in Section 4.2<br>— Get Event Filter<br>— Set Event Filter<br>— Get Default Link Policy<br>— Set Default Link Policy<br>— ACL Indications<br>• Added Section 4.3.1 "ACL Error Codes" |
| 1.2 (08/19/03) | • Added Table 2-21 "Log File of Incoming Link as automatic slave"<br>• Added Table 2-22 "Log File of a Released Link as Automatic Slave"<br>• Updated Section 2.1 "Using the automatic slave operation"<br>• Updated Section 3.3.2 "Sniff Mode"<br>• Updated Section 3.3.3 "Hold Mode"<br>• Updated Table 4-3 "Opcode Values" |