

# AN-1699 LMX9838 Software Users Guide

## User's Guide



Literature Number: SNOA498A  
March 2008—Revised February 2013

<b>1</b>	<b>Introduction</b>	<b>13</b>
<b>2</b>	<b>LMX9838 General Setup</b>	<b>13</b>
2.1	SYSTEM OVERVIEW	13
2.2	SYSTEM PARAMETERS	15
2.3	UART COMMUNICATION	19
2.4	AUDIO SUPPORT	21
<b>3</b>	<b>Initialization And Bluetooth Operation</b>	<b>22</b>
3.1	INITIALIZATION MODE	22
3.2	BLUETOOTH MODE	22
<b>4</b>	<b>Basic Link Establishment</b>	<b>29</b>
4.1	ACCEPTING INCOMING LINKS	29
4.2	SETTING UP A LINK USING THE COMMAND INTERFACE	31
4.3	USING TRANSPARENT MODE	39
4.4	EXAMPLES (SUMMARY)	40
<b>5</b>	<b>Advanced Usage</b>	<b>42</b>
5.1	LOCAL CONFIGURATION	42
5.2	SETTING UP MULTIPLE CONNECTIONS	50
5.3	DEFAULT CONNECTIONS	55
5.4	LOW POWER OPERATION	56
5.5	ESTABLISH AUDIO LINKS	64
5.6	SOFTWARE PATCHES	70
<b>6</b>	<b>Profile Support</b>	<b>72</b>
6.1	MULTIPLE SERVICE ENTRIES	73
6.2	SERIAL PORT PROFILE (SPP)	77
6.3	DIAL-UP NETWORKING (DUN)	77
6.4	HEADSET PROFILE	85
<b>7</b>	<b>LMX9838 Command Interface</b>	<b>105</b>
7.1	UART PROTOCOL PRINCIPLES	105
7.2	COMMAND SET	109
7.3	ERROR CODES	179
7.4	AT COMMANDS	183
<b>8</b>	<b>Annex</b>	<b>185</b>
8.1	INITIALIZATION IF EEPROM EMPTY	185
<b>9</b>	<b>Bibliography</b>	<b>186</b>

## List of Figures

1	LMX9838 Firmware Implementation . . . . .	14
2	LMX9838 Main Interface Blocks . . . . .	14
3	UART Null Modem Connection . . . . .	19
4	Receiving Data In Command Mode . . . . .	21
5	Receiving Data in Transparent Mode . . . . .	21
6	PCM Codec Connection Block Diagram . . . . .	22
7	Operation Flow After Boot-Up or Reset . . . . .	28
8	Relation Between Different Operation States . . . . .	29
9	Create SPPLink.dir . . . . .	31
10	Difference Between a Standard 0 Transmission and Break Signal . . . . .	40
11	Master Default Link Timeout . . . . .	47
12	Set Link Timeout at Slave . . . . .	47
13	Set Link Timeout at Master . . . . .	47
14	Poll Period . . . . .	48
15	Standard Point-To-Point SPP Connection . . . . .	50
16	Multiple SPP Links from LMX9838 . . . . .	50
17	Transition Between Different Hardware Power Modes . . . . .	58
18	Host Wake Up Hardware Connections . . . . .	58
19	Disable Transport Layer Timing . . . . .	59
20	LMX9838 Waking Up the Host in Command Mode . . . . .	60
21	No TL Enabled Event After Automatic Idle . . . . .	60
22	Host Waking Up LMX9838 . . . . .	61
23	SCO Link Bandwidth Requirements . . . . .	65
24	SCO and ACL/SPP Routing Within LMX9838 . . . . .	65
25	Influence of Audio Settings to LMX9838 Blocks and Formats . . . . .	66
26	AAI Generic Data Framing . . . . .	67
27	Theoretical AAI Interface Stream at Motorola Settings (Long Frame Sync) . . . . .	67
28	AAI Stream Overview for Motorola Codec (Short Frame Sync) . . . . .	67
29	AAI Slot Usage for Motorola with Audio Control set to 0x00 (Short Frame Sync) . . . . .	68
30	Difference Between Long and Short Frame Sync Signal . . . . .	69
31	Inverted Frame Sync Pulses . . . . .	69
32	Hardware Patch Mechanism . . . . .	71
33	Patch Command Flow . . . . .	72
34	Configuring the Service Database . . . . .	74
35	Example Differentiate Different Profiles . . . . .	76
36	Software Layer Implementations for DUN . . . . .	78
37	Command Flow: Requesting the RFCOMM Port Number from the Remote Device . . . . .	79
38	Command Flow: Establish a Link to a DUN Service of the Remote Device . . . . .	79
39	Flow Command: Sending AT Commands . . . . .	80
40	Sending AT Commands . . . . .	80
41	Using LMX9838 in a DUN GW Application . . . . .	81
42	Opening "Create Service Record" Dialog-box . . . . .	82
43	Selecting the Profile . . . . .	82
44	Configuring the Profile Settings for DUN . . . . .	83
45	Creating the Command String for the Audio Gateway Entry . . . . .	83
46	Send the Command to the LMX9838 . . . . .	84
47	Headset Data and Audio stream . . . . .	86

48	Opening “Create Service Record” Dialog-box . . . . .	87
49	Selecting the Profile . . . . .	88
50	Configuring the Profile Settings for Headset. . . . .	88
51	Creating the Command String for the Headset Entry. . . . .	89
52	Send the Command to the LMX9838. . . . .	89
53	“Incoming Audio Connection Establishment” as Defined in Bluetooth Specification . . . . .	91
54	Incoming Audio Connection with LMX9838 as HS. . . . .	91
55	Outgoing Audio Connection Establishment as Defined by the Bluetooth SIG . . . . .	92
56	Outgoing Link Establishment from the LMX9838 as HS . . . . .	93
57	Audio Link Release by the HS . . . . .	94
58	Audio Link Release from the LMX9838 as HS . . . . .	94
59	Audio Link Release by the HSAG . . . . .	94
60	Audio Link Release from the HSAG . . . . .	95
61	Audio Gateway Data and Audio Stream using LMX9838 in the HSAG . . . . .	95
62	Opening “Create Service Record” Dialog-box . . . . .	97
63	Selecting the Profile . . . . .	97
64	Configuring the Profile Settings for HSAG . . . . .	98
65	Creating the Command String for the Audio Gateway Entry . . . . .	98
66	Send the Command to the LMX9838. . . . .	98
67	“AG initiated Audio Connection Establishment” . . . . .	100
68	Outgoing Audio Connection with LMX9838 as HSAG . . . . .	101
69	Incoming Audio Connection on HSAG . . . . .	102
70	Incoming HS Connection as HSAG . . . . .	103
71	Audio Link Release by the HS . . . . .	104
72	Audio Link Release by the HS . . . . .	104
73	Audio Link Release Initiated by the HSAG . . . . .	105
74	Audio Link Release Initiated by the HSAG . . . . .	105
75	Inquiry Command Flow. . . . .	110
76	Get Remote Device Name Flow . . . . .	111
77	Requesting the Services of a Remote Device . . . . .	112
78	SPP Link Command Flow . . . . .	117
79	Transparent Mode; initiated by the Transparent Mode Request. . . . .	121
80	Transparent Mode; Automatically Activated in Transparent Slave . . . . .	122
81	Set SPP DTR. . . . .	147
82	Set Scan Mode . . . . .	159
83	Authentication Procedure with Fixed Pin . . . . .	165
84	Authentication Procedure with Pin Request . . . . .	165

## List of Tables

1	LMX9838 Information . . . . .	13
2	LMX9838 System Parameters, EEPROM Memory Map . . . . .	15
3	Overview of Operation States . . . . .	24
4	Incoming Link Established . . . . .	30
5	Example Incoming Link Established . . . . .	30
6	Standard Link Released Messages . . . . .	30
7	Example Transparent Mode Lost Indicator . . . . .	30
8	Example Link Released Indicator . . . . .	30
9	Inquiry Command Example . . . . .	31
10	Example Inquiry Command Package . . . . .	32
11	Example Device Found Indicator Package . . . . .	33
12	Example Inquiry Confirm Package . . . . .	33
13	Log of Create SDAP Command . . . . .	34
14	Example Create SDAP Connection . . . . .	34
15	Log of SDAP Browse for SPP . . . . .	34
16	Example SDAP Browse . . . . .	34
17	Example SDAP Browse Confirm . . . . .	35
18	Log of SDAP Disconnect . . . . .	35
19	Example Create SPP Link . . . . .	35
20	Log of Create SPP Connection . . . . .	36
21	Example Establish Link Confirm . . . . .	36
22	Example Link Established Indicator . . . . .	36
23	Log of Sending the Data "Test" . . . . .	37
24	Example Sending Data Package . . . . .	37
25	Example Sending Data "Testdata" . . . . .	37
26	Log of Incoming Data Event . . . . .	37
27	Example Incoming Data Event . . . . .	38
28	Log of Release Link Command . . . . .	38
29	Example Release Link Package . . . . .	38
30	Example Link Released Indicator Package . . . . .	38
31	Log of Set Transparent Mode . . . . .	39
32	Example Set Transparent Mode . . . . .	39
33	Example Set Transparent Mode Confirm . . . . .	39
34	Log of Transparent Mode Event . . . . .	40
35	Example Transparent Mode Indicator . . . . .	40
36	Log File of Incoming Link as Automatic Slave . . . . .	41
37	Log File of a Released Link as Automatic Slave . . . . .	41
38	Log File of a Complete Link Setup . . . . .	41
39	UART Settings . . . . .	42
40	Change NVS UART Speed to 57.6kbit/s . . . . .	43
41	Open 3 RFComm Ports . . . . .	49
42	Default SDP Entry on RFComm Port 1 . . . . .	49
43	Open RFComm Ports 1, 2 and 3 . . . . .	51
44	SDB Entry with Security on Outgoing Connection . . . . .	51
45	Example SDP Store SPP Record . . . . .	51
46	Establish Link on Local RFComm Port 1 . . . . .	52
47	Establish Link on Local RFComm Port 2 . . . . .	52

48	Establish Link on Local RFCOMM Port 3 . . . . .	52
49	Initialize 3 RFCOMM Ports . . . . .	53
50	Configure Additional SDB Entry to Enable Auth/encr. for Outgoing Connection. . . . .	53
51	Device Discovery . . . . .	53
52	Get Remote RFCOMM Port of First Device . . . . .	53
53	Get Remote RFCOMM Port of the Second Device. . . . .	54
54	Get Remote RFCOMM Port of the Third Device. . . . .	54
55	Establish Links to All Three Slaves . . . . .	55
56	Low Power Modes. . . . .	56
57	Patch Format at the Different Locations. . . . .	71
58	Default Service Database Configuration Index . . . . .	73
59	Configuring the Service Database . . . . .	74
60	Example Service Database Configuration . . . . .	74
61	Open the Necessary RFCOMM Ports . . . . .	75
62	Host Notification for Incoming Link on RFCOMM Port 2 . . . . .	76
63	Get Remote RFCOMM Port for DUN Profile of the Remote Device . . . . .	78
64	Establish Link to DUN Service on Remote Device. . . . .	79
65	Send init command “AT” and dialing command “ATDT01234567890” . . . . .	80
66	Send Init Command “AT” and Dialing Command “ATDT01234567890” . . . . .	81
67	Adding the DUN Service to the Service Database . . . . .	84
68	Service Database for COM1 and DUN . . . . .	84
69	Set RFCOMM Ports to Open . . . . .	84
70	Set Default Audio Settings to OKI Codec and CVSD. . . . .	86
71	Delete SDB Database . . . . .	87
72	Adding the Headset Service to the Service Database . . . . .	89
73	Service Database for COM1 and Headset . . . . .	90
74	Set RFCOMM Ports to Open . . . . .	90
75	Store Class of Device . . . . .	90
76	Incoming Audio Connection with LMX9838 as HS. . . . .	91
77	Requesting the RFCOMM Port Number for HSAG from the Remote Device . . . . .	92
78	Outgoing Link Establishment from the LMX9838 as HS . . . . .	93
79	Audio Link Release from the LMX9838 as HS . . . . .	94
80	Audio Link Release from the LMX9838 as HS . . . . .	95
81	Set Default Audio Settings to Motorola Codec and CVSD. . . . .	96
82	Adding the HSAG Service to the Service Database . . . . .	99
83	Service Database for COM1 and HSAG . . . . .	99
84	Set RFCOMM Ports to Open . . . . .	99
85	Store Class of Device . . . . .	99
86	Requesting the RFCOMM Port Number for HSAG from the Remote Device. . . . .	100
87	Outgoing Audio Connection with LMX9838 as HSAG . . . . .	101
88	Incoming Headset Connection . . . . .	103
89	Audio Link Release by the HS . . . . .	104
90	Audio Link Release by the HSAG . . . . .	105
91	Package Framing . . . . .	106
92	Packet Type Identification . . . . .	106
93	Opcode Values . . . . .	106
94	Inquiry Command . . . . .	110
95	Inquiry Confirm . . . . .	110
96	Device Found Indicator. . . . .	110

97	Get Remote Device Name Command . . . . .	111
98	Get Remote Device Name Confirm . . . . .	111
99	SDAP Connect Request . . . . .	112
100	SDAP Connect Confirm . . . . .	112
101	Disconnect Request . . . . .	113
102	Disconnect Confirm . . . . .	113
103	Connection Lost Indicator . . . . .	113
104	Example UUIDs for Service Classes . . . . .	113
105	SDAP Service Browse Request . . . . .	113
106	SDAP Service Browse Confirm . . . . .	114
107	SDAP Service Search Request . . . . .	115
108	SDAP Service Search Confirm . . . . .	115
109	SDAP Service Request . . . . .	115
110	SDAP Attribute Request Confirm . . . . .	116
111	SDAP Attribute Request . . . . .	116
112	SDAP Attribute Request Confirm . . . . .	116
113	Establish Link Request . . . . .	117
114	Establish Link Confirm . . . . .	117
115	SPP Link Established Indicator . . . . .	118
116	SPP Send Data . . . . .	118
117	SPP Send Data Confirm . . . . .	119
118	SPP Incoming Data Indicator . . . . .	119
119	SPP Release Link Request . . . . .	119
120	SPP Release Link Confirm . . . . .	119
121	SPP Link Released Indicator . . . . .	120
122	Incoming Link Established Indicator . . . . .	120
123	Read RSSI Request . . . . .	120
124	Read RSSI Confirm . . . . .	121
125	Transparent Mode Request . . . . .	122
126	Transparent Mode Confirm . . . . .	122
127	Transparent Mode Left Indicator . . . . .	122
128	Set Default Link Timeout Request . . . . .	123
129	Set Default Link Timeout Confirm . . . . .	123
130	Get Default Link Timeout Request . . . . .	124
131	Get Default Link Timeout Confirm . . . . .	124
132	Set Link Timeout Request . . . . .	124
133	Set Link Timeout Confirm . . . . .	124
134	Get Link Timeout Request . . . . .	125
135	Get Link Timeout Confirm . . . . .	125
136	Set Default Audio Settings Command . . . . .	125
137	Get Default Audio Settings Command . . . . .	126
138	Get Default Audio Settings Confirm . . . . .	126
139	Set PCM Slave Configuration Request . . . . .	127
140	Set PCM Slave Configuration Confirm . . . . .	127
141	Establish SCO Link Command . . . . .	128
142	Establish SCO Link Confirm . . . . .	128
143	SCO Link Established Indicator . . . . .	128
144	Release SCO Link Command . . . . .	129
145	Release SCO Link Confirm . . . . .	129

146	SCO Link Released Indicator	129
147	Set Audio Settings Command	130
148	Set Audio Settings Confirm	130
149	Get Audio Settings Command	130
150	Get Audio Settings Confirm	130
151	Change SCO Packet Type Command	131
152	Change SCO Packet Type Confirm	131
153	Change SCO Link Packet Type Indicator	131
154	Mute Command	132
155	Mute Confirm	132
156	Set Volume Command	132
157	Set Volume Confirm	133
158	Get Volume Command	133
159	Get Volume Confirm	133
160	Example Default Connection Table	134
161	Store Default Connection Confirm Request	134
162	Store Default Connection Confirm	134
163	Connect to Default Connection Request	135
164	Store Default Connection Confirm	135
165	Get List of Default Connections Request	135
166	Get List of Default Connections Confirm	135
167	Delete Default Connections Request	136
168	Delete Default Connections Confirm	136
169	Set Default Link Policy Command	137
170	Set Default Link Policy Confirm	137
171	Get Default Link Policy Command	137
172	Get Default Link Policy Confirm	137
173	Set Link Policy Request	138
174	Set Link Policy Confirm	138
175	Get Link Policy Request	138
176	Get Link Policy Confirm	138
177	Enter Sniff Mode Request	139
178	Enter Sniff Mode Confirm	139
179	Exit Sniff Mode Request	140
180	Exit Sniff Mode Confirm	140
181	Enter Park Mode Request	140
182	Enter Park Mode Confirm	140
183	Exit Park Mode Request	141
184	Enter Park Mode Confirm	141
185	Enter Hold Mode Request	141
186	Enter Hold Mode Confirm	141
187	Power Save Mode Changed Indicator	142
188	Disable Transport Layer Command	142
189	Disable Transport Layer Confirm	143
190	Transport Layer enabled indicator	143
191	SPP Set Port Config Request	143
192	SPP Set Port Configuration Confirm	145
193	SPP Get Port Configuration Request	145
194	SPP Get Port Configuration Confirm	145

195	SPP Port Configuration Changed Indicator . . . . .	146
196	SPP Get Port Status Confirm . . . . .	148
197	SPP Port Set DTR Request . . . . .	149
198	SPP Port Set DTR Confirm . . . . .	149
199	SPP Port Set RTS Request . . . . .	149
200	SPP Port Set RTS Confirm . . . . .	149
201	SPP Set Port Break Request . . . . .	150
202	SPP Set Port Break Confirm . . . . .	150
203	SPP Set Overrun Error Request . . . . .	150
204	SPP Port Overrun Error Confirm . . . . .	151
205	SPP Port Parity Error Request . . . . .	151
206	SPP Port Parity Error Confirm . . . . .	151
207	SPP Port Framing Error Request . . . . .	151
208	SPP Port Framing Errors Confirm . . . . .	152
209	SPP Port Status Changed Indicator . . . . .	152
210	SDP Set Ports to Open Request . . . . .	153
211	SDP Set Ports to Open Confirm . . . . .	153
212	SDP Get Ports to Open Request . . . . .	153
213	SDP Get Ports to Open Confirm . . . . .	153
214	Store Generic SDP Record Request . . . . .	154
215	Store Generic SDP Record Confirm . . . . .	155
216	SDP Enable SDP Record Request . . . . .	155
217	SDP Enable SDP Record Confirm . . . . .	155
218	SDP Delete SDP Record Request . . . . .	156
219	SDP Delete SDP Record Confirm . . . . .	156
220	Read Local Name Request . . . . .	156
221	Read Local Name Confirm . . . . .	156
222	Write Local Name Request . . . . .	157
223	Write Local Name Confirm . . . . .	157
224	Read Local BD_Addr Request . . . . .	157
225	Read Local BD_Addr Confirm . . . . .	157
226	Change Local BD_Address Request . . . . .	158
227	Change Local BD_Addr Confirm . . . . .	158
228	Store Class of Device Request . . . . .	158
229	Store Class of Device Confirm . . . . .	158
230	Set Scan Mode Request . . . . .	159
231	Set Scan Mode Confirm . . . . .	159
232	Automatic Limited Discoverable Ended Indicator . . . . .	160
233	Force Master Role Request . . . . .	160
234	Force Master Role Confirm . . . . .	160
235	Read Operation Mode Request . . . . .	160
236	Read Operation Mode Confirm . . . . .	161
237	Write Operation Mode Request . . . . .	161
238	Write Operation Mode Confirm . . . . .	161
239	Set Default Link Latency Request . . . . .	161
240	Set Default Link Latency Confirm . . . . .	162
241	Get Default Link Latency Request . . . . .	162
242	Get Default Link Latency Confirm . . . . .	162
243	ACL Established Indicator . . . . .	163

244	ACL Terminated Indicator . . . . .	163
245	Get Security Mode Request . . . . .	164
246	Get Security Mode Confirm . . . . .	164
247	Set Security Mode Request . . . . .	164
248	Set Security Mode Confirm . . . . .	164
249	Remove Pairing Command . . . . .	165
250	Remove Paired Device Confirm . . . . .	166
251	List Paired Devices Command . . . . .	166
252	List Paired Devices Confirm . . . . .	166
253	Get Fixed Pin Request . . . . .	166
254	Get Fixed Pin Confirm . . . . .	166
255	Set Fixed Pin Request . . . . .	167
256	Set Fixed Pin Confirm . . . . .	167
257	Pin request indicator . . . . .	168
258	Pin Request Response Command . . . . .	168
259	Pin Request Command Confirm . . . . .	168
260	Set Event Filter Command . . . . .	168
261	Set Event Filter Confirm . . . . .	169
262	Get Event Filter Command . . . . .	169
263	Get Event Filter Confirm . . . . .	169
264	Reset Request . . . . .	170
265	Simply Blue Ready Indicator . . . . .	170
266	Restore Factory Settings Request . . . . .	170
267	Restore Factory Settings Confirm . . . . .	170
268	Change UART Speed Request . . . . .	171
269	Change UART Speed Confirm . . . . .	171
270	Change UART Settings Request . . . . .	171
271	Change UART Settings Confirm . . . . .	172
272	Initiate Bluetooth Test Mode Request . . . . .	172
273	Initiate Bluetooth Test Mode Confirm . . . . .	172
274	Initiate RF Test Mode Request . . . . .	173
275	Initiate RF Test Mode Confirm . . . . .	174
276	Enter Bluetooth Mode Request . . . . .	174
277	Enter Bluetooth Mode Confirm . . . . .	174
278	Set Clock and Baudrate Request . . . . .	174
279	Set Clock and Baudrate Confirm . . . . .	174
280	Write ROM Patch Request . . . . .	175
281	Write ROM Patch Confirm . . . . .	175
282	Control Byte Pin Assignments . . . . .	176
283	Set GPIO WPU Request . . . . .	176
284	Set GPIO WPU Confirm . . . . .	176
285	Get GPIO Input State Request . . . . .	176
286	Get GPIO Status Confirm . . . . .	177
287	Set GPIO Direction Request . . . . .	177
288	Set GPIO Direction Confirm . . . . .	177
289	Set GPIO Output High Request . . . . .	177
290	Set GPIO Output High Confirm . . . . .	177
291	Set GPIO Output Low Request . . . . .	178
292	Set GPIO Output Low Confirm . . . . .	178

293	Read NVS Request . . . . .	178
294	Read NVS Confirm . . . . .	179
295	Write NVS Request . . . . .	179
296	Write NVS Confirm. . . . .	179
297	ACL Error Codes . . . . .	180
298	Generic Error Codes. . . . .	181
299	RFCOMM Error Codes . . . . .	183
300	RFCOMM Release Reasons . . . . .	183
301	Required AT Commands for DUN GW. . . . .	183
302	Result Codes for DUN. . . . .	184
303	AT Commands for Headset Profile . . . . .	184
304	Required Result Codes for Headset Implementations . . . . .	185
305	Await Initialization Indicator . . . . .	186

## ***AN-1699 LMX9838 Software Users Guide***

---

---

---

The Texas Instruments LMX9838 Bluetooth® Serial Port module is a fully integrated Bluetooth 2.0 baseband controller, 2.4 GHz radio, crystal, antenna, LDO and discrets combined to form a complete small form factor (10 mm x17 mm x 2.0 mm) Bluetooth node.

## 1 Introduction

The Texas Instruments LMX9838 Bluetooth® Serial Port module is a fully integrated Bluetooth 2.0 baseband controller, 2.4 GHz radio, crystal, antenna, LDO and discrets combined to form a complete small form factor (10 mm x17 mm x 2.0 mm) Bluetooth node.

All hardware and firmware is included to provide a complete solution from antenna through the complete lower and upper layers of the Bluetooth stack, up to the application including the Generic Access Profile (GAP), the Service Discovery Application Profile (SDAP), and the Serial Port Profile (SPP). The module includes a configurable service database to fulfill service requests for additional profiles on the host. Moreover, the LMX9838 is pre-qualified as a Bluetooth subsystem. Conformance testing through the Bluetooth qualification program enables a short time to market after system integration by ensuring a high probability of compliance and interoperability.

Based on TI's CompactRISC™ 16-bit processor architecture and Digital Smart Radio technology, the LMX9838 is optimized to handle the data and link management processing requirements of a Bluetooth node.

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

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

This document is a reference for implementing the LMX9838 module into a system. A getting started session gives a very detailed entry point for starting development. The advance usage section describes all features and configuration parameters in detail and gives example for using the LMX9838 as active bluetooth node. Finally all commands and events are listed and explained in the command section.

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

This document is based on:

**Table 1. LMX9838 Information**

Item	Version
Hardware	LMX9838
Firmware	V0212 or later
Actual Firmware Release	V0212
SimplyBlueCommander	1.6.0.1

## 2 LMX9838 General Setup

### 2.1 SYSTEM OVERVIEW

The LMX9838 is intended to be an add-on module to an existing microcontroller. In this function it either appears as cable like interface for the UART or can also be controlled with a simple application on the external microcontroller to establish links itself.

The LMX9838 includes the complete bluetooth stack including the following protocol layers:

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

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

- GAP (Generic Application Profile)
- SDAP (Service Discovery Application Profile)

- SPP (Serial Port Profile)

The application manages all profile related interactions to the stack but also offers a simplified command interface over the UART. The interface is used for configuring the device, setting up the link and receiving events from the module.

The interface can handle either packaged data transmission for multipoint support or is able to handle RAW data by setting it into a transparent UART mode.

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

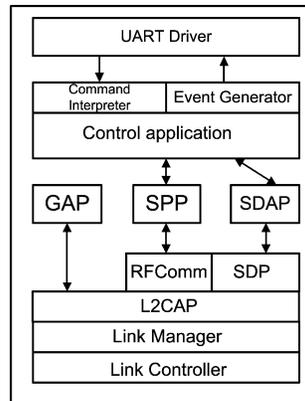


Figure 1. LMX9838 Firmware Implementation

### 2.1.1 Hardware Setup

Figure 2 shows a block diagram for the LMX9838. The block diagram shows all functional blocks connected to the device.

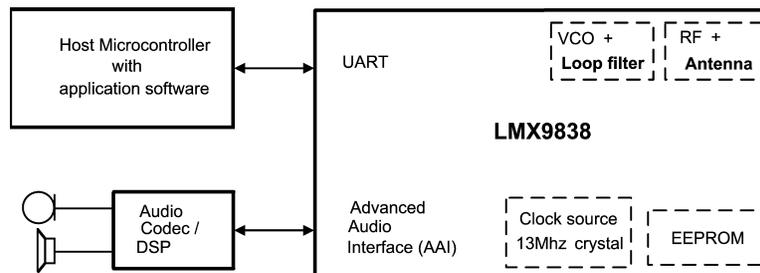


Figure 2. LMX9838 Main Interface Blocks

#### 2.1.1.1 Host Microcontroller Over UART

The host microcontroller can be any microcontroller, sending data or commands to the LMX9838 over the UART interface. In case the LMX9838 is used to actively establish outgoing links, the host controller needs to send the appropriate commands over the UART interface.

The LMX9838 offers bluetooth operation up to the Serial Port Profile (SPP), which is the basis for many other profiles like DUN or Headset. In case such profiles shall be supported by the end product, the additional profile needs to be implemented on the host application, which uses the LMX9838 as kind of "SPP gateway". See also Section 6 on how to support additional profiles.

### 2.1.1.2 System Parameters in Non-volatile Storage (NVS)

Since the LMX9838 is ROM based, all parameters used during a session will be stored in RAM and would be lost after power-down. For bluetooth operation as well as for internal configuration the LMX9838 has a number of System Parameters (Section 2.2), which contain parameters mandatory for bluetooth operation (e.g. Bluetooth Device Address (BD\_Addr)) as well as configuration values for the LMX9838. See Section 2.2 for a complete list of parameters.

To avoid reconfiguration of the LMX9838 on each power-up, all System Parameters (Section 2.2) are stored in the internal EEPROM. The LMX9838 automatically stores changed parameters within the EEPROM and restores them after boot-up and reset.

---

**NOTE: NEVER DELETE THE INTERNAL EEPROM.** The EEPROM contains important information such as the BD address which should never be changed. If the module does not work properly anymore you can use the "Restore Factory Settings" command (Section 7.2.18.5) to reset the EEPROM content to a known good state.

---

### 2.1.1.3 Clock Source

The LMX9838 includes an internal 13 MHz crystal, already tuned to give the optimal RF performances.

### 2.1.1.4 Audio Codec / DSP

As mentioned before, the LMX9838 can be used to transport control data required for additional profiles on the host. With this it is possible to use the device for audio profiles like Headset Profile (HSP) or Handsfree Profile (HFP). Audio data to be transported over the Bluetooth link will be handled through the Advanced Audio Interface (AAI), without any interaction required from the host.

The LMX9838 audio interface supports PCM master or slave operation. Master operation is used with typical codecs like the OKI, Motorola or Winbond codecs while PCM slave allows the flexible combination with DSPs or other host controllers. See also Section 5.5.

Please check "Texas Instruments: LMX9838 Datasheet" for details on which codecs are supported by LMX9838.

## 2.2 SYSTEM PARAMETERS

The LMX9838 is controlled by system parameters. Since these parameters are stored in an internal EEPROM or non-volatile memory, they are also called "NVS parameters". These parameters define the chip behavior during bluetooth operation but also after a software or hardware reset.

Since the LMX9838 is ROM based, these working parameters are stored in EEPROM and in a dedicated RAM area. These parameters and possible patches are loaded from EEPROM into RAM during boot up.

Table 2 lists the memory map of the System parameters. The address reflects the memory address of the internal EEPROM.

**Table 2. LMX9838 System Parameters, EEPROM Memory Map**

No.	Address	Parameter	Default Value	Description	SW Reset Rrequired
1	0000-0005	BD_Addr	<Unique address>	Bluetooth Device Address.	no
				LAP(lsb), LAP, LAP, UAP, NAP, NAP (msb)	
				Unique address programmed by TI, do not delete.	
2	0006	NVS	0x00	Indicates whether the EEPROM has been initialized or not.	yes
		Initialized		0x00: Initialized	
				0xFF: Not initialized	
3	0007	UnitKeyPresent	0xFF	Used by BT core, generated during pairing procedure.	no

**Table 2. LMX9838 System Parameters, EEPROM Memory Map (continued)**

No.	Address	Parameter	Default Value	Description	SW Reset Required
4	0008-0017	UnitKey	0xFF..0xFF	Used by BT core, generated during pairing procedure.	no
5	0018	DeviceNameLength	0xFF	Length of Parameter 6 "Devicename".	no
6	0019-0040	DeviceName	0xFF...0xFF	Friendly Name of the Bluetooth Device.	no
7	0041	CountryCode	0x00	Used by BT core.	yes
8	0042	PinLength	0x04	The length of parameter 9, "PinCode". In case set to 0, the LMX9838 will request pin from host.	no
9	0043-0052	PinCode	0000	Fixed PinCode used for pairing with other devices.	no
10	0053-0055	ClassOfDevice	0x000000	The "Class of Device" describes general functionality of the Bluetooth Device and is transmitted during the Inquiry process.	no
11	0056	SppPortsToOpen	0x00000001	Bitmask defining the RFCOMM channels to open. For each channel one RFCOMM instance will be created.	no
12	005A	PreferredMasterRole	0x00	Preferred Master forces the device to switch to Master Role after being connected. The device will reject the link if command could not be executed.	yes
13	005B	Automatic Operation	0x01	Configures the general behavior of the device. Please see <a href="#">Section 3.2</a> for details. 0x00: Automatic OFF (Non-automatic) 0x01: Automatic ON (Automatic)	yes
14	005C	PageScanMode	0x01	Configures the connectability of the device. 0x00: not connectable 0x01: normal scan 0x81: interlaced scan (faster connection time)	no
15	005D	InquiryScanMode	0x01	Configures the discoverability of the device. 0x00: not discoverable 0x01: normal scan 0x81: interlaced scan (faster response time)	no
16	005E	SecurityMode	0x02	Configures Service Level Security Mode.	no
17	005F-0060	DefaultLinkPolicy	0x000F	Configures the default link policy for incoming links.	no
18	0061	EventFilter	0x01	Configures the level of events reported to the host. 0x00: No filter, all events reported 0x01: ACL events filtered, only API events reported 0x02: All events filtered, only UART breaks indicated 0x03: All events filtered, including UART break	no
19	0062	PMM / GPIO usage	0xFF	Bitmask to configure enhanced power management (PMM) functions as well as the usage of dedicated GPIO pins. 0x01: enhanced PMM: 1 - disabled (default) 0 - enabled (requires 32 kHz crystal) 0x02: Use PG6 to signal SPP activity (Links) 0x04: Use PG7 to signal TL activity All other bits are reserved and should be set to 1.	yes
20	0063-0064	LinkTimeout	0x7D00	Configures the default link supervision timeout (in slots, 0.625ms) used for incoming and outgoing links.	no

**Table 2. LMX9838 System Parameters, EEPROM Memory Map (continued)**

No.	Address	Parameter	Default Value	Description	SW Reset Rrequired
21	0065	CodecType	0x00	The audio codec settings used on the PCM interface.	yes
				0x00: None connected	
				0x01: Motorola MC145483 / Winbond W681360	
				0x02: OKI MSM7717 / Winbond W681310	
				0x03: PCM slave, see PCMSlaveConfig	
				0x04-0xFF: reserved	
22	0066	AirFormat	0x00	The audio format used on the SCO link.	yes
				0x00: CVSD	
				0x01: $\mu$ -Law	
				0x02: A-Law	
				0x03-0xFF: reserved	
23	0067-0068	RfcommLatency	0x0000	Configures the default poll period of master to slave.	no
				0x0000: No requirement (default 40slots)	
				0x0002-0x0190: Valid link latency	
24	0069-006C	Frequency	0x00000000	The crystal frequency in Hz. The frequency parameter is only needed when the firmware start up in a mode with unknown crystal frequency (10-20MHz). OP pins are used to determine if the crystal frequency is unknown.	yes
25	006d	UartParityBit	0x00	Parity setting for the hardware UART interface.	yes
				0x00: No Parity	
				0x01: Even Parity	
				0x02: Odd Parity	
26	006E	UartStopBit	0x00	Stop bit settings for the hardware UART interface.	yes
				0x00: 1 Stop bit	
				0x01: 2 Stop bits	
27	006F	UartSpeed	0x03	Speed of the Hardware UART interface. This settings are only used, in case the OP pins are configured to Autobaudrate detect.	yes
				2400: 0x00	
				4800: 0x01	
				7200: 0x02	
				9600: 0x03	
				19200: 0x04	
				38400: 0x05	
				57600: 0x06	
				115200: 0x07	
				230400: 0x08	
				460800: 0x09	
				921600: 0x0A	
28	0070-00AE	RemoteDevices	0x00..0x00	Default connections database, to be connected during boot-up or by sending a command.	no
29	00AF	VtuneDesiredThreshold	0xFF	TI configured parameter, do not modify.	yes
30	00B0	VtuneCn	0xFF	TI configured parameter, do not modify.	yes
31	00B1	VtuneEnable	0xFF	TI configured parameter, do not modify.	yes

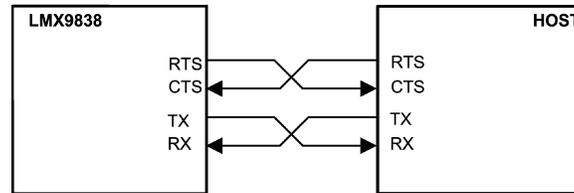
**Table 2. LMX9838 System Parameters, EEPROM Memory Map (continued)**

No.	Address	Parameter	Default Value	Description	SW Reset Rrequired
32	00B2-00B3	PcmSlaveConfig	0xFFFF	This 16-bit value (LSB first) is used to store the PCM format configuration for the PCM slave configuration. This setting is only use in case the PCM slave setting is activated (see No. 21).	yes
				BIT0-1: Slot selection	
				00: use slot 0	
				01: use slot 1	
				10: use slot 2	
				11: use slot 3	
				BIT2-3: Number of slots per frame	
				00: 1 slot	
				01: 2 slots	
				10: 3 slots	
				11: 4 slots	
				BIT4-6: PCM data format	
				000: Reserved	
				001: 8 bit A-Law	
				010: 8 bit $\mu$ -Law	
				011: 13 bit linear	
				100: 14 bit linear	
				101: 15 bit linear	
				110: 16 bit linear	
				111: Reserved	
BIT7: Frame sync length					
0: short frame sync					
1: long frame sync					
BIT8: Data word length					
0: 8-bit data word length					
1: 16-bit data word length					
BIT9: Frame sync polarity					
0: use inverted frame sync					
1: use normal frame sync					
BIT10-15: Unused, set to 0					
33	00B4	PcmFcprs	0xFF	Unsigned integer indicating the frame clock prescaler for generic PCM slave.	yes
34	00B5-00B8	RfSetupReg4	0xFFFF	TI configured parameter, do not modify.	yes
35	00B9-00BC	RfSetupReg15	0xFFFF	TI configured parameter, do not modify.	yes
36	00BD-00EE	Filler 1	0xFF ... 0xFF	Filler (not used).	
37	00EF-011E	ServiceRecords	0xFF..0xFF	SDP/Security info storage.	no
38	011F-0346	CoreNvsLinkKeys	0xFF..0xFF	Link key storage (24 keys).	no
39	0347-0355	AssertInfo	0xFF..0xFF	TI configured parameter, do not modify.	
40	0356-0367	RunErrorInfo	0xFF..0xFF	TI configured parameter, do not modify.	
41	0368-037F	Filler 2	0xFF..0xFF	Filler (not used).	
42	0380-137F	SdpRecords	0xFF..0xFF	SDP record storage.	no
43	1380-1FFF	PatchCode	0xFF..0xFF	ROM patch code storage.	yes

## 2.3 UART COMMUNICATION

The main communication interface between the LMX9838 and the host is the UART Interface.

The UART interface between host and LMX9838 needs to be connected in Null Modem configuration, meaning RTS/CTS and TX/RX are crossed.



**Figure 3. UART Null Modem Connection**

The command interface is based on a command/event based structure. Each command (also called "Request") will be acknowledged with the appropriate status event (also called "Confirm"). Unexpected events, like incoming link establishment or data, are also sent as events, but signed with a different simply blue package type called "Indicator". Please see [Section 7](#) for the detailed description of the command interface.

Since the LMX9838 acts like a gateway between the Bluetooth Link and the UART interface, the UART connection should use 4-wire hardware handshaking for optimal buffer handling. The LMX9838 uses the RTS lines to indicate low buffers and reacts on the CTS from the host, immediately stopping sending packages to the host.

In case the host microcontroller is not able to provide hardware handshaking, the UART should be used in "Command Mode" ([Section 2.3.2.1](#)) only, since the device would not be able to indicate full buffers by using the RTS signal. In the same way, the host needs to provide enough buffer space to be able to handle the incoming data, since it's not able to stop the flow from the LMX9838.

### 2.3.1 UART Configurations

The UART interface consists of four signals:

- TX: Transmit output
- RX: Receive input
- RTS: Ready-to-Send output, indicating the host "I am ready to receive data"
- CTS: Clear-to-Send input, allows the host to stop the transmission from the LMX9838 to the host

The LMX9838 will ALWAYS use the RTS to indicate to the host, that it is ready to receive data and it will ALWAYS sample the CTS input to check, if the host is able to receive data. Therefore, it has to be made sure that the CTS pin is pulled low in case the host is ready. Otherwise the LMX9838 will not start sending out data or events.

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

#### 2.3.1.1 UART 4-wire Connection for Hardware Handshaking

The 4-wire configuration is done as shown in [Figure 3](#), which is the recommended setting. This configuration allows full operation in "Command Mode" ([Section 2.3.2.1](#)) and "Transparent Mode" ([Section 2.3.2.2](#)).

The handshake functionality is based on the RTS / CTS signalling, which is used in both modes. The LMX9838 indicates with its RTS signal (RTS=low), that it is able to receive data and will raise it high in case the TX buffers are full. This indicator can get very important in "Transparent Mode" ([Section 2.3.2.2](#)) in case the host transmits a lot of data over the LMX9838. In case the remote bluetooth device runs into a buffer problem or the bluetooth link is slow, the LMX9838 will toggle the RTS as soon as its buffers are filled up.

The incoming CTS input can be used by the host to stop the flow from the LMX9838 to the host. This might get important in case the host processor is a slow performing device, not able to process the incoming data in sufficient time.

---

**NOTE:** The CTS signal is the only way to stop the flow from the LMX9838 to the host.

---

### 2.3.1.2 UART 2-wire Connection

In general, the LMX9838 does not absolutely require the RTS / CTS signals to be connected to the host. However, since the LMX9838 always assumes the signals to be used, the following issues have to be addressed or are recommended:

- CTS to GND:
  - The LMX9838 uses the CTS input to get the permission to send out data. If this signal is not pulled to GND, the device will not start sending data.
- For TX, use "Command Mode" ([Section 2.3.2.1](#)) only:
  - As described, the LMX9838 will indicate buffer issues by pulling RTS to high. Since the host does not check this pin it is recommended to use "Command Mode" ([Section 2.3.2.1](#)) only. In this mode, data have to be sent using the "SPP Send Data" command ([Section 7.2.3.3](#)). Each command is confirmed with a specific event including the command status. The LMX9838 will indicate the successful command processing with status 0x00. This can be seen as kind of software handshaking.
- **DO NOT USE "Transparent Mode" ([Section 2.3.2.2](#)) FOR TRANSMITTING DATA IN THIS CONFIGURATION.**
  - Using Transparent Mode without hardware handshake, it can not be guaranteed that data are not overwritten within the LMX9838. Use "Command Mode" ([Section 2.3.2.1](#)) instead.
- For RX, host needs to be fast enough:
  - Since the CTS signal is not used and fixed to GND, the host will not be able to tell the LMX9838 to stop the flow, in case he's running out of buffers. Therefore, only use 2-wire connection if the host is capable to process the incoming data in appropriate time.

## 2.3.2 UART Modes

### 2.3.2.1 Command Mode

The LMX9838 offers a wide range of commands to configure the hardware and the bluetooth operation. As the command set is on top of the profiles, Bluetooth operational commands are reduced to high level commands controlling general bluetooth operation.

In Command Mode, the LMX9838 will try to interpret all data sent over the UART to a known command. The commands have to be sent in a specific package format. The interface is based on an event mechanism. Any command sent will be confirmed by the appropriate confirmation event. Unexpected events (e.g. incoming links) will be reported by indication events.

Please refer to [Section 7](#) for a complete list of commands and their usage.

### 2.3.2.2 Transparent Mode

In case the LMX9838 has established a link to only one remote device and no configuration commands have to be sent to the LMX9838 "Command Mode" ([Section 2.3.2.1](#)), the UART interface can be switched to "Transparent mode".

This means data are directly routed to the bluetooth link and not interpreted. Also incoming data are not indicated as events, they are sent as RAW data to the UART.

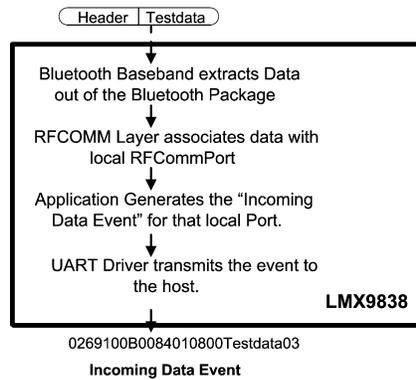


Figure 4. Receiving Data In Command Mode

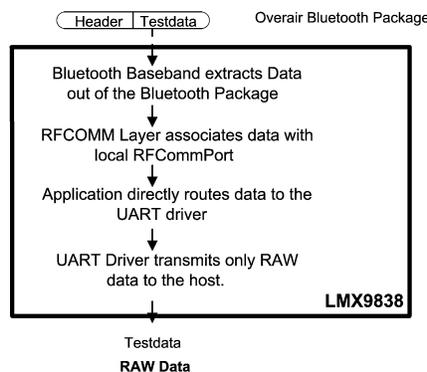


Figure 5. Receiving Data in Transparent Mode

**Leaving transparent mode:**

As the LMX9838 does not listen to commands, UART Break has to be used to tell the device to leave the transparent mode. See also [Section 7.2.4.1](#) for details.

**2.4 AUDIO SUPPORT**

The LMX9838 offers commands allowing to establish or to accept synchronous audio links. The audio data will be routed within the LMX9838 between the bluetooth baseband connection and the Advanced Audio interface. In order to encode or decode the PCM stream generated by the internal CVSD codec, an external codec or DSP has to be connected to the advanced audio interface. Alternatively the data can be used for further digital processing.

The AAI is configured by predefined codec settings, selected in NVS. The NVS setting configures the bit rate as well as the format like PCM log or linear. In addition, the interface can be configured to "PCM Slave" in which the external codec or DSP needs to provide the synchronous clock and frame sync signal. The settings for the PCM slave mode are done by specific commands. See also [Section 5.5.3.1](#) for detailed instructions on codec configuration.

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

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

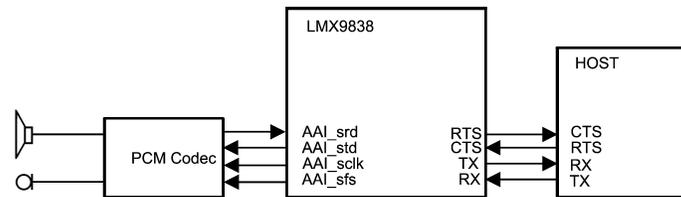


Figure 6. PCM Codec Connection Block Diagram

### 3 Initialization And Bluetooth Operation

The LMX9838 operation can be split into 2 modes, initialization mode and bluetooth mode. After boot or reset, the LMX9838 starts into the initialization mode, which checks for necessary operation parameters. The content of the internal EEPROM is copied in to the internal RAM, then once all necessary parameters have been checked, the device will enter bluetooth mode.

#### 3.1 INITIALIZATION MODE

TI programs an official BD address during fabrication. The module should therefore get into bluetooth mode immediately without any steps.

The initialization procedure is required only if the BD address is erased or not programmed. In this case please see [Section 8.1](#).

#### 3.2 BLUETOOTH MODE

The operation of the LMX9838 can be divided into different states. Each state represents special situations and describes the behavior of the module.

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

##### 3.2.1 NVS Settings Influencing the Operation State

Before actually describing the different states, we should have a look on the parameters, which influence the behavior of the LMX9838. The following parameters are stored in NVS and are checked during boot-up.

###### 3.2.1.1 Automatic Operation On/Off

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

###### 3.2.1.1.1 Automatic Operation On

- Connect to "Default connections stored in NVS" ([Section 3.2.1.3](#))
- Switch UART to "Transparent Mode" ([Section 2.3.2.2](#)) after first incoming link "Transparent Slave" ([Section 3.2.4.7](#)).

###### 3.2.1.1.2 Automatic Operation Off

- No automatic steps after Reset/Boot-up

###### 3.2.1.2 Force Master On/Off

This flag improves multipoint operation of the LMX9838. In normal operation, the LMX9838 will be slave for any incoming link, which limits the number of incoming connects to 2 devices [Section 3.2.4.6](#).

In case the Force Master Bit in the NVS is set to On, the LMX9838 will request a role switch on any incoming link to become Master.

### 3.2.1.2.1 Force Master On

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

### 3.2.1.2.2 Force Master Off

- Accept incoming link as slave.

### 3.2.1.3 Default Connections Stored in NVS

In case the automatic operation bit (see [Section 3.2.1.1](#)) is set to On, the default connections database is checked for any valid entry. If a valid entry is found, the LMX9838 tries three times to connect to each device stored in the database.

#### 3.2.1.3.1 Default Connection Transparent On/Off

Each default connection can be configured to switch the UART to "Transparent Mode" ([Section 2.3.2.2](#)) or to "Command Mode" ([Section 2.3.2.1](#)) after link establishment. In case the transparent flag is set, the LMX9838 will switch to "Transparent Mode" ([Section 2.3.2.2](#)) once the link is established.

---

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

---

## 3.2.2 Service Database

The service database is the source for a remote device to see, which services/profiles the LMX9838 and its host can offer. This database is usually stored in internal EEPROM and recovered after any Reset.

Since this database can be quite large, it is not part of the standard RAM NVS but is kept as instance in the main working RAM. This allows to keep flexibility and performance as high as possible.

If no configurations are made, the SDB is configured to its default setting. See [Section 5.1.2.9](#) for details on the configuration of the service database.

## 3.2.3 Patches

A patch allows to correct small parts of the firmware in ROM. In case a patch is available, the LMX9838 will use the patch code instead of the ROM code. The Patch will be stored in internal EEPROM and copied into RAM at each boot up. For more details on Patch RAM mechanism, see [Section 5.6](#).

## 3.2.4 Operation States

As combination out of the different parameters and usage scenarios, the following states can be defined for the LMX9838. The parameters and possibilities for this states are summarized in [Table 3](#).

How to read [Table 3](#). E.g. Scatternet Master (assuming default NVS settings).

In Scatternet Master, the device is in command mode (listening to commands).

The automatic flag in NVS has no influence. The device is discoverable and connectable for other devices.

The device is Master for x slaves and slave to 1 Master.

It is possible to search for other devices but it is not possible to be connected from another device. RAW data traffic is not possible.

No incoming link possible.

In case the command interface is used to establish a link to another device, the device stays in "Scatternet Master" ([Section 3.2.4.3](#)).

A UART BREAK has no influence on the functionality. Sending the command "Transparent Mode" ([Section 7.2.4](#)) is not applicable for this mode. A SCO link can be established to one of the remote devices.

**Table 3. Overview of Operation States**

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

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

### 3.2.4.1 Idle, Automatic and Idle, Non-automatic

After boot-up, reset or after successfully releasing the last link, the LMX9838 stays within one of the Idle states.

In case the Automatic Operation bit is set to "On", it is also called "Idle, Automatic". Otherwise, the state is called "Idle, Non-Automatic".

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

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

#### 3.2.4.1.1 Incoming Link at Idle Automatic

In case, the LMX9838 is not connected to any other device and gets an incoming link request, it will:

- ask for authentication or pin code exchange
- accept the link
- notify the application by an indicator

- turn off scanning (disabling page scan and inquiry scan)
- switch UART to "Transparent Mode" ([Section 2.3.2.2](#))

"Idle Automatic" is optimized for cable replacement applications, not requiring any interaction with the LMX9838 to accept an incoming link. As the LMX9838 automatically switches to transparent mode, the application can start sending Raw data immediately after receiving the notification.

#### 3.2.4.1.2 Incoming Link in Idle Non-Automatic

In case, the LMX9838 is not connected to any other device and gets an incoming link request, it will:

- ask for authentication or pin code exchange
- accept the link
- notify the application by an indicator
- change state to "Single Slave" ([Section 3.2.4.5](#))

After link establishment, the command interface is still active, delivering incoming data using the "Incoming Data Indicator" ([Section 7.2.3.4](#)). To send data the command "SPP Send Data" ([Section 7.2.3.3](#)) needs to be used.

"Idle Non-automatic" is optimized for multi-profile applications which need to manage multiple links or different profiles at the same time. As the LMX9838 stays in command mode, the application still has full control over the LMX9838, to establish links or do configurations.

In case "Transparent Mode" ([Section 2.3.2.2](#)) would be beneficial for some data transfer, it can be reached any time by sending the "Transparent Mode" command ([Section 7.2.4.1](#)).

#### 3.2.4.2 Piconet Master

In Piconet Master, the LMX9838 is in a link with one or more devices as Bluetooth Master. The bluetooth specification describes the Master as the controlling device for the piconet. The Master defines the hopping sequence and manages the connection to each slave.

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

The LMX9838 can be assumed as master of the bluetooth link and to be in "Piconet Master" when:

- the command interface has been used to actively establish one or more links by using one of the following commands
  - "Establish Link" ([Section 7.2.3.1](#))
  - "Connect to Default Connection" ([Section 7.2.7.2](#))
  - a default connection has been established after power-up or "Reset" ([Section 7.2.18.3](#))
- the LMX9838 accepted an incoming link while the Force Master bit in NVS was set to 0x01 ([Section 7.2.14.8](#))

After successfully establishing a link, the LMX9838 will stay in "Command Mode" ([Section 2.3.2.1](#)).

The benefit of being master instead of slave is, that the LMX9838 is actively managing the link to all devices, so can assign each device the bandwidth it requires. With this, the LMX9838 is able to support up to 7 active links.

In Piconet Master, the LMX9838 is able to handle one SCO link to one connected slave.

---

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

---

The LMX9838 has successfully been tested for 7 active links in combination with 1 SCO link, 7 open ports, and 7 service entries. Since a SCO link takes minimum 33% of the complete bluetooth bandwidth, reliability on link establishment can be reduced.

### 3.2.4.3 Scatternet Master

In case a bluetooth device is master for one or several slaves and in parallel slave to one master, the connection scenario is called "scatternet". The LMX9838 is able to be master to one or multiple slaves and in addition can be slave to maximum one master. Within the LMX9838 scenario, this state is called "Scatternet Master".

The LMX9838 can be assumed to be in this state, after one of the following connection situations:

- The LMX9838 accepts an incoming link as "Piconet Master" ([Section 3.2.4.2](#)) while the Force Master bit is set to 0x00.
- The device is "Single Slave" ([Section 3.2.4.5](#)) and actively establishes a link to another device by using one of the following commands:
  - "SDAP Connect" ([Section 7.2.2.1](#))
  - "Establish Link" ([Section 7.2.3.1](#))
  - "Connect to Default Connection" ([Section 7.2.7.2](#))

Once Scatternet Master has been reached LMX9838 is not able to accept another incoming link. However the device will still be discoverable and will still answer to service requests.

### 3.2.4.4 Transparent Master

In case the LMX9838 is only connected to one other device ("point-to-point" connection), it might be beneficial for the application to send data directly to the UART interface, without having to use the command "SPP Send Data" ([Section 7.2.3.3](#)). For this the LMX9838 offers the so called "Transparent Mode" on the UART (see [Section 2.3.2.2](#)), which allows to send data directly.

Since the LMX9838 in this case gets no information to which port to send this data to, transparent mode is only allowed on a point-to-point connection. Data will be routed directly from the UART interface to the remote bluetooth device.

Transparent Master means, the LMX9838 is master for the point-to-point connection to one other link and Transparent Mode is switched on. This state is reached by one of the following situations:

- Sending the "Transparent Mode" command ([Section 7.2.4.1](#)) in "Piconet Master" state ([Section 3.2.4.2](#)).
- In case Automatic operation is ON, Default Connection setup after Reset, in which the transparent flag is set to 0x01.
- Default Connection setup by using "Connect to Default Connection" ([Section 7.2.7.2](#)), in which the transparent flag is set to 0x01.

Leaving Transparent Master, initiated by a UART break, will lead into "Piconet Master" state ([Section 3.2.4.2](#)).

Since the LMX9838 can not send any events or react on incoming commands, scanning is switched off and therefore the LMX9838 is not discoverable or connectable for other devices.

### 3.2.4.5 Single Slave

The Bluetooth specification defines a bluetooth slave as the device which is connected by another device and adjusting to the timing of that device (Master). The slave synchronizes to the clock of Master and to its hopping sequence. In an active link, the master polls each slave (by default every 40 slots, see also [Section 7.2.14.11](#) to keep them synchronized but also to allow the slave to send data.

By default, the LMX9838 accepts any incoming link. Depending on the configuration of Security level ("Set Security Mode" [Section 7.2.16.2](#)) and the Service Database entry, the device will ask for authentication.

The LMX9838 can be assumed to be in Single Slave after one of the following actions appeared:

- The LMX9838 accepted an incoming link and reports it by the "SPP Link Established Indicator" ([Section 7.2.3.2](#)), while the Automatic Operation flag is set to 0x00 (Non-automatic).

- The host sends a UART Break to a LMX9838 in "Transparent Slave" ([Section 3.2.4.7](#)).

In theory, there's also the possibility for Single Slave, in case the LMX9838 actively establishes a link to another device, which requests a role switch. In this case the initiating device will be slave. Since the role switch is not reported to the command interface and therefore can't be proven, this scenario shall not be discussed in this document.

In Single slave the LMX9838 still listens to commands on the UART. The device will be discoverable and connectable for other devices. The existing link can be used as basis for a SCO link.

#### 3.2.4.6 Scatternet Slave

An advanced but not most efficient connection state is the Scatternet Slave. In this mode the LMX9838 is slave to two different masters. This means, the LMX9838 needs to switch between two different synchronization states over time, serving each Master only for a limited time. While it is synchronized to Master 1 it is not able to listen to Master 2, therefore might miss the poll packages.

The LMX9838 can be assumed to be in Scatternet Slave after the following action appeared:

- The LMX9838 accepted an incoming link as "Single Slave" ([Section 3.2.4.5](#)), reported by "SPP Link Established Indicator" ([Section 7.2.3.2](#)).

The LMX9838 is able to manage such a link without issues. However, since the switching between two piconets consumes significant bandwidth, "Piconet Master" ([Section 3.2.4.2](#)) or at least "Scatternet Master" ([Section 3.2.4.3](#)) should be used instead, which for example can be reached by setting the Force Master Flag within the NVS (see [Section 7.2.14.8](#)).

In Scatternet Slave, the LMX9838 can NOT handle a SCO link on one of the links. Discoverability and connectability are switched off.

#### 3.2.4.7 Transparent Slave

A typical scenario for the LMX9838 is the cable replacement, in which the LMX9838 just waits for an incoming connection and the host connected over UART starts transmitting data after an incoming link has been established. Since in many cases the LMX9838 just replaces a former cable connection, the "Transparent Mode" ([Section 2.3.2.2](#)) on the UART allows implementation without any software change on the data transmissions.

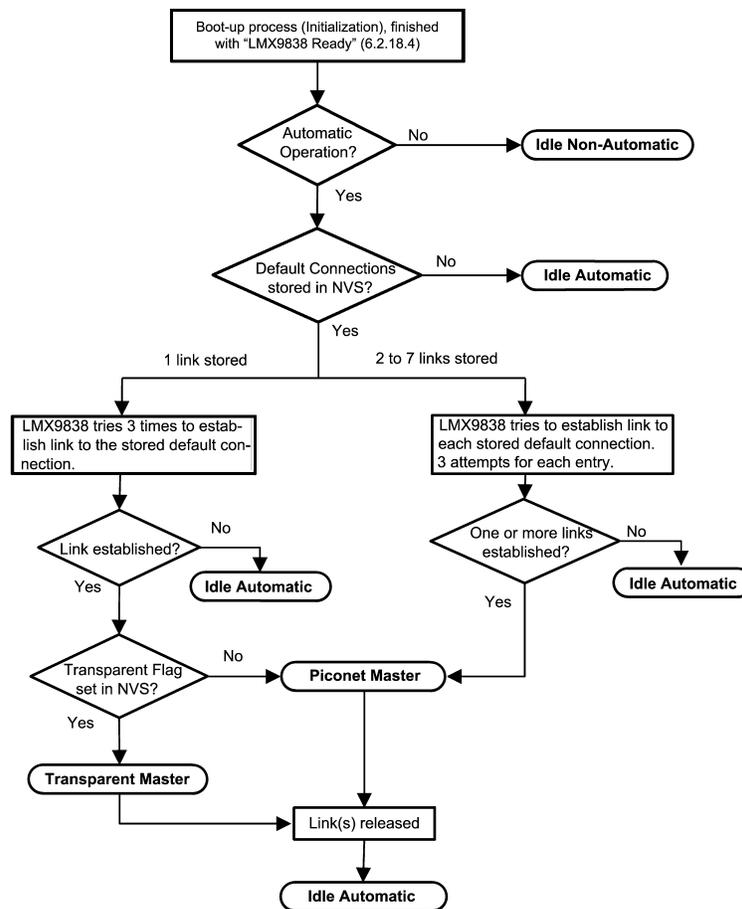
Transparent Slave means, the device is slave on the bluetooth link and the UART is switched to "Transparent Mode" ([Section 2.3.2.2](#)).

The LMX9838 can be assumed to be in "Transparent Slave" state after on of the following actions appeared:

- the LMX9838 accepted an incoming link in "Idle, Automatic" ([Section 3.2.4.1](#)), in which the Automatic Flag is switched to On.
- the host sent command "Transparent Mode" ([Section 7.2.4.1](#)) while the LMX9838 is in "Single Slave" state ([Section 3.2.4.5](#)).

Leaving Transparent Slave, initiated by a UART break, will lead into "Single Slave" state ([Section 3.2.4.5](#)).

Since the LMX9838 can not send any events or react on incoming commands, scanning is switched off and therefore the LMX9838 is not discoverable or connectable for other devices.


**Figure 7. Operation Flow After Boot-Up or Reset**

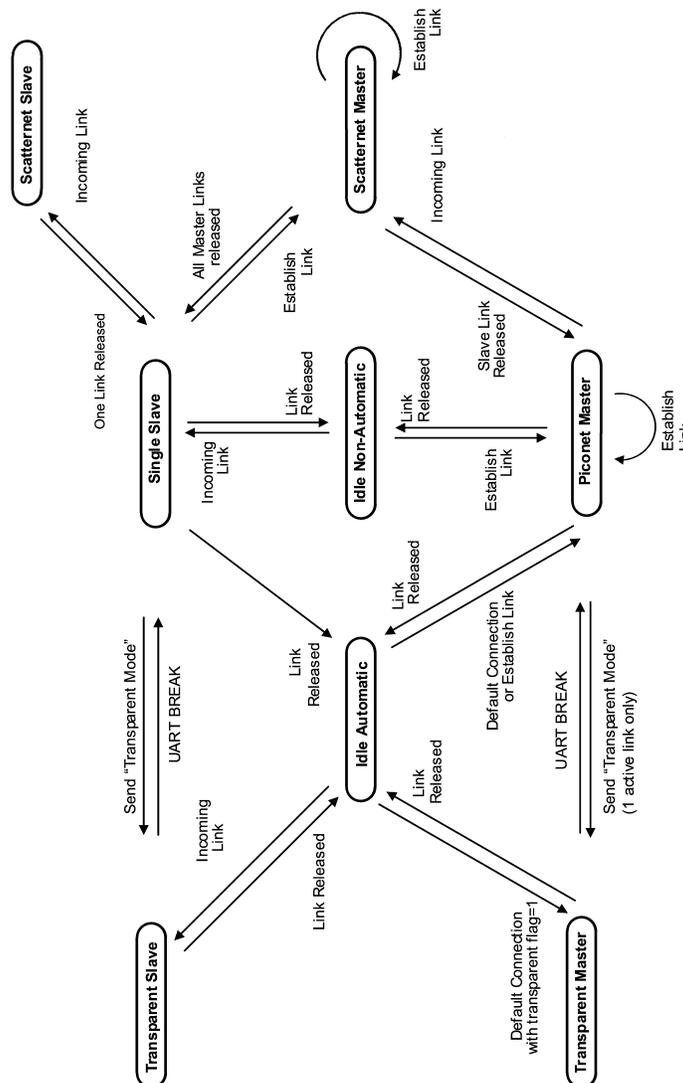


Figure 8. Relation Between Different Operation States

## 4 Basic Link Establishment

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

### 4.1 ACCEPTING INCOMING LINKS

As described in Section 3.2.4 the LMX9838 on default is in a waiting mode (Idle Automatic) after boot-up or reset. This means it waits for requests and automatically answers to connection requests.

If connected from a remote device, the LMX9838 establishes automatically a SPP link and indicates the established link to host by the "Link Established Event" and by setting PG6 pin to 1. By default the device enters "Transparent Slave" state (Section 3.2.4.7).

**Table 4. Incoming Link Established**

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

The event indicates the local RFCOMM Port and the BD\_Addr of the remote device.

**Table 5. Example Incoming Link Established**

Byte	Parameter	Value
01	Start Delimiter	02
02 - 06	Package Header	69 0C 07 00 7C
07- 12	BD_Addr of remote device	12 34 56 78 9A BC
13	Local RFCOMM Port	01
14	End Delimiter	03

After this event, the module automatically switches to "Transparent Mode" ([Section 2.3.2.2](#)) and routes all incoming and outgoing data from the RF side "unmodified" to the UART or vice versa. This switching process is not indicated to the host.

If the link is dropped, the LMX9838 will empty its buffers and

- send a UART break to the host
- send "Transparent Mode" Indicator ([Section 7.2.4.1](#))
  - Indicates on protocol level to the host that transparent mode has been left.
- send "SPP Release Link" Indicator ([Section 7.2.3.5](#))
  - Indicates that link has been released.
- PG6 pin will be set back to 0

**Table 6. Standard Link Released Messages**

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

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

Please see also [Section 7.1](#) for a complete description of the package and header format.

**Table 7. Example Transparent Mode Lost Indicator**

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

**Table 8. Example Link Released Indicator**

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

**Table 8. Example Link Released Indicator (continued)**

Byte	Parameter	Value
09	End Delimiter	03

## 4.2 SETTING UP A LINK USING THE COMMAND INTERFACE

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

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

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



**Figure 9. Create SPPLink.dir**

### 4.2.1 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 LMX9838 this process can be started with the command “Inquiry” ([Section 7.2.1.1](#)).

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.

**Table 9. Inquiry Command Example**

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

[Table 9](#) shows the package format used on the Command interface. The TX indicates the Inquiry command sent to the device, the two Rx lines the events from the LMX9838.

The following paragraphs explain the package format and usage in more detail. The complete package format is also described in [Section 7.1](#).

#### a.) The Inquiry Command.

Let's first have a look on the Inquiry command:

```
02 52 00 03 00 55 0A 00 00 03
```

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

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

In this case:

```
02 52 00 03 00 55
```

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

The **packet type id** for a request is 0x52. (see [Section 7.1.3](#) for the complete list).

The **opcode** for Inquiry is 00 (see [Section 7.1.4](#) for the complete list).

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

The payload for this package is:

```
0A 00 00
```

so the length is 0x0003 (bytes).

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

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

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

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

**Table 10. Example Inquiry Command Package**

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

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

### b.) The Device Found Indicator

The first response to the inquiry command from the LMX9838 is the Device\_Found\_Indicator. In hex:

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

**Package header:**

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

### The Payload:

46 95 28 D9 0A 00 04 02 52

- BD\_Addr - 46 95 28 D9 0A 00
- Class of Device - 04 02 52

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

BD\_Addr: 00 0A D9 28 95 46

Class of Device: 52 02 04 (Mobile Phone)

**Table 11. Example Device Found Indicator Package**

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

### c.) The Inquiry Confirm

Every command on the LMX9838 command interface is confirmed by an appropriate event. The confirmation always has the opcode as the command sent to the device. The event also indicates the success status of the command or any parameters requested. If no error occurred, status/error 0x00 will be returned. All other values have a specific reason. Please see [Table 298](#) and [Table 299](#) for a complete list of error codes.

The confirmation in hex:

02 43 00 01 00 44 00 03

Package header:

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

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

**Table 12. Example Inquiry Confirm Package**

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

## 4.2.2 Create SDAP Connection

To create a SPP connection to another device, the local RFCComm channel has to know which remote RFCComm Channel to address. Each service is registered to a specific RFCComm 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.

**Table 13. Log of Create SDAP Command**

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

**Table 14. Example Create SDAP Connection**

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

The only parameter of the command is the BD\_Addr to connect to:

46 95 28 D9 0A 00 (byte swapped)

The command is confirmed by the LMX9838 with the appropriate confirmation event. If status is 0x00 the link has been established.

#### 4.2.3 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 LMX9838 within the command.

**Table 15. Log of SDAP Browse for SPP**

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

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

**Table 16. Example SDAP Browse**

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

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

The full event includes the following parameters:

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

- Name of the service

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

**Table 17. Example SDAP Browse Confirm**

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

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

#### 4.2.4 SDAP Disconnect

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

**Table 18. Log of SDAP Disconnect**

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

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

#### 4.2.5 Create SPP Connection

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

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

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

**Table 19. Example Create SPP Link**

Byte	Parameter	Value
01	Start Delimiter	02
02 - 06	Package Header	52 0A 08 00 64
07	Local RFCOMM Port	01 (default at LMX9838)
08 - 13	Remote BD_Addr	46 95 28 D9 0A 00
14	Remote RFCOMM Port	04 (out of SDAP Request)
15	End Delimiter	03

**Table 20. Log of Create SPP Connection**

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

The Log Window shows 3 events returned by the LMX9838.

#### a) Establish Link Confirm

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

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

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

**Table 21. Example Establish Link Confirm**

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

#### b) Port Status Changed Indicator

This event indicates that during the RFCComm channel setup process the settings of the SPP link have changed.

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

Please see [Section 7.2.10.3](#) for the detailed description of the event.

#### c) Link Established Indicator

To indicate an established link on top of the SPP, the LMX9838 uses the “Link Established Indicator” ([Section 7.2.3.7](#)). The event returns:

- Status/Error code
- BD\_Addr of the remote device
- Local RFCComm port
- Remote RFCComm port

**Table 22. Example Link Established Indicator**

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

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

#### 4.2.6 Sending Data in Command Mode

After actively setting up a connection with the LMX9838, the device is in state "Piconet Master" (Section 3.2.4.2), still listening to commands and returning status changes by events.

To send data over the command interface the "Send Data" command (Section 7.2.3.3) has to be used.

Besides the data which have to be sent, the local RFCOMM Port parameter has also to be sent to the LMX9838. This enables the application to support multiple connections.

**Table 23. Log of Sending the Data "Test"**

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

**Table 24. Example Sending Data Package**

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

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

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

**Table 25. Example Sending Data "Testdata"**

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

#### 4.2.7 Receiving Data in Command Mode

In command mode, incoming data from a remote device are indicated by the "Incoming Data" (Section 7.2.3.4).

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

**Table 26. Log of Incoming Data Event**

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

**Table 26. Log of Incoming Data Event (continued)**

Direction	What	Hex Code	Interpreted by Simply Blue Commander
RX	Indicator	02,69,10,04,00,7D,01,01,00,65,03	Rx: Event: Incoming Data, Local Port: 01, Received Data: 65
RX	Indicator	02,69,10,04,00,7D,01,01,00,73,03	Rx: Event: Incoming Data, Local Port: 01, Received Data: 73
RX	Indicator	02,69,10,04,00,7D,01,01,00,74,03	Rx: Event: Incoming Data, Local Port: 01, Received Data: 74

Table 26 shows the log of 4 bytes received on local RfComm Port 01. The bytes together form the word “Test” again.

**Table 27. Example Incoming Data Event**

Byte	Parameter	Value
1	Start Delimiter	2
02-06	Package Header	69 10 04 00 7D
7	Local RfComm Port	1
08-09	Length of Data received	01 00 (byte swapped)
10	Received Data	54 (“T”)

#### 4.2.8 Releasing a SPP Connection

To release an existing SPP connection the “Release Link” Command (Section 7.2.3.5) is used. The command is referring to the local RfComm port the connection has been established on.

**Table 28. Log of Release Link Command**

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

**Table 29. Example Release Link Package**

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

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

##### a) Release Link Confirm

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

##### b) Link Released Indicator

The event indicates that the LMX9838 released the Link on the RfComm Port returned.

**Table 30. Example Link Released Indicator Package**

Byte	Parameter	Value
01	Start Delimiter	02
02-06	Package Header	69 0E 02 00 79
07	Reason Code	00

**Table 30. Example Link Released Indicator Package (continued)**

Byte	Parameter	Value
08	Local RfComm Port	01
09	End Delimiter	03

### 4.3 USING TRANSPARENT MODE

#### 4.3.1 Activating Transparent Mode

In Transparent Mode, described in [Section 2.3.2.2](#), the LMX9838 acts as cable replacement device. In this mode the LMX9838 does not interpret the packages sent to device. Instead, it is directly forwarding the data straight to the link previously set up.

If a bluetooth link to the LMX9838 has been established, this mode is automatically activated if the Automatic Operation Flag in NVS is set to 0x01 (default).

If the link was set up manually via the LMX9838 command interface (see [Section 4.2](#)), the LMX9838 still listens to commands and data have to be sent via the "Send Data" command ("Piconet Master")([Section 3.2.4.2](#)).

As the LMX9838 routes the data directly to the bluetooth link, "Transparent Mode" ([Section 2.3.2.2](#)) can only be activated if only one active link exists. The following command has to be used to switch into transparent mode:

- "Transparent Mode"

The "Transparent mode" command is referring to the local RfComm port the link was created on.

The command is confirmed by the appropriate event. Afterwards the LMX9838 is routing all incoming data directly to the remote device.

**Table 31. Log of Set Transparent Mode**

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

**Table 32. Example Set Transparent Mode**

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

**Table 33. Example Set Transparent Mode Confirm**

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

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

The LMX9838 is leaving the transparent mode, when a break signal is sent on the UART (see [Section 4.3.2](#)).

The break signal can also be used if the LMX9838 has been connected from a remote device and switched automatically to transparent (Automatic slave mode).

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

**Table 34. Log of Transparent Mode Event**

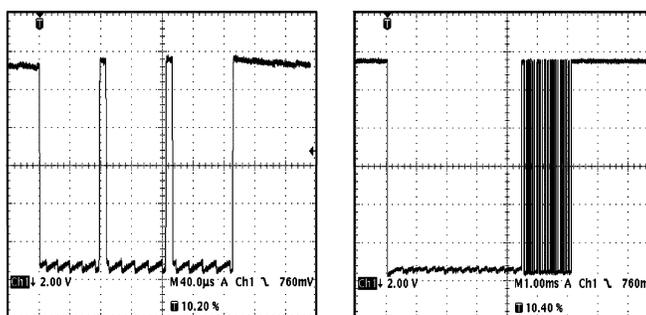
Direction	What	Hex Code	Interpreted by Simply Blue Commander
RX	Indicator	02,69,11,02,00,7C,01,00,03	Rx: Event: Transparent Mode, Local Port: 01, Mode: 00

**Table 35. Example Transparent Mode Indicator**

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

### 4.3.2 Leaving Transparent Mode with UART BREAK

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



**Figure 10. Difference Between a Standard 0 Transmission and Break Signal**

Figure 10 shows the difference between the signal of a normal 0 and the BREAK signal. The left picture shows the signalling of 3 Zeros at 115.2kbit/s. Each character is started and ended with a start bit and a Stop bit. The normal length of 1 byte is therefore about 86.8µs (1startbit + 8bit data + 1stopbit).

The picture on the right shows a BREAK signalled by the LMX9838 after a released link. The signal is held low for over 4 ms. Theoretical minimum value for a BREAK at this speed would be about 165µs.

## 4.4 EXAMPLES (SUMMARY)

The following log files show the typical hex values sent to respectively returned from the LMX9838. The level of events returned by the LMX9838 depends on the event filter level set within the NVS. The tables with the log entries also show the filter level, in which the messages are reported. Default filter setting is 01, so only those events will be reported.

Please see also [Section 5.1.1.6](#) for details on the Event Filter setting.

#### 4.4.1 Automatic Slave

Table 36 shows the event returned from the LMX9838 if it was connected from outside. The LMX9838 just returns one event indicating the BD\_Addr of the remote device and the local RFComm port it connected to.

**Table 36. Log File of Incoming Link as Automatic Slave**

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

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

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

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

#### 4.4.2 Setting up a Link

As documented in Section 4.2, 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 38 shows all commands necessary from scratch to establish a link to another device.

**Table 38. Log File of a Complete Link Setup**

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

**Table 38. Log File of a Complete Link Setup (continued)**

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

## 5 Advanced Usage

The LMX9838 offers a wide variety of functions for different usage models. This section points out the most important features and scenarios covered by the LMX9838.

### 5.1 LOCAL CONFIGURATION

#### 5.1.1 Hardware Configuration

The LMX9838 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 7.2.18](#) and [Section 7.2.19](#).

##### 5.1.1.1 Change UART Settings

The UART speed in general is determined by the choice of the OP pins of the LMX9838. The pins and UART settings in NVS are only checked during the software boot-up process so also after a Reset command.

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

**Table 39. UART Settings**

OP3	OP4	OP5	Function
1	0	0	UART speed read from NVS
1	0	1	UART speed 9.6 kbps
1	1	0	UART speed 115.2 kbps
1	1	1	UART speed 921.6 kbps

#### Example: Configuring LMX9838 for UART speed of 57.6kbit/s:

- Set OP3=1, OP4=0 and OP5=0
- Power up / Reset the device, the UART is configured on 9.6kbit/s on default. LMX9838 answers with "SimplyBlue LMX9838 Ready". If set to an unknown speed, please set the OP pins to a dedicated speed and try again.
- Send the following command:
  - — "Change NVS UART Speed" with parameter 0x06.

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

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

- Await the Confirmation Event from the LMX9838.
- Send the following command to do a hardware reset
  - — “Reset”

Afterwards the LMX9838 will communicate with a UART speed of 57.6kbit/s

### 5.1.1.2 Bluetooth Testmode

The LMX9838 supports the standard Bluetooth “Device Under Test” Mode and a “Local Loopback” Mode. The “Device under Test” Mode is the standard testmode used by any bluetooth tester. If activated the tester will be able to take control over the LMX9838 and put it into the specific testmodes needed for bluetooth qualification.

The “local loopback” mode is a simple UART loopback mode to test the UART communication interface. The testmodes are enabled by the command “Test Mode”.

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

Please see [Section 7.2.19.1](#) for details for the command.

### 5.1.1.3 RF Testmodes

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

The detailed command is described in [Section 7.2.19.2](#).

### 5.1.1.4 Restore Factory Settings

The LMX9838 is delivered with standard settings in NVS which can be seen in [Table 2](#). Those parameters are changing during usage or testing the part.

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

- Bluetooth Device Address (BD\_Addr)

---

**NOTE:** Please be aware that this command resets also the baud rate settings to 9600 bps. Command details can be found in [Section 7.2.18.5](#).

---

### 5.1.1.5 Read RSSI

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

This command offers the ability to read out the RSSI and indicate the current status of the signal strength. (See [Section 7.2.3.8](#)).

The value indicates:

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

- 0x01: Signal too strong

### 5.1.1.6 Event Filter

The LMX9838 indicates status changes and confirms commands with specific events. To address specific application requirements the level of reporting can be increased or decreased.

The following separation is possible:

- Report standard events including ACL indicators, which indicate the physical connection status.
- Report standard events only.
- Report no events, only UART break indicates lost link, LMX9838 still detects UART Break to leave "Transparent Mode" ([Section 2.3.2.2](#)).
- No reporting on the UART and "Transparent Mode" ([Section 2.3.2.2](#)) not left on UART BREAK.

The event filter can be configured by the following command:

- "Set Event Filter"
- "Get Event Filter"

#### 5.1.1.6.1 Report Standard Events Including ACL

In case the NVS setting is set to "report all events", the LMX9838 reports any status change and ACL link establishment to the host.

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

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

---

**NOTE:** Please be aware that any kind of established ACL link will be reported to the host. If the device is only waiting for connection, any attempt from another device to connect to the LMX9838 will be reported to the host. For a "slave only" usage, one of the other filtering settings would probably be more useful.

---

Available ACL Events:

- "ACL Established"
- "ACL Terminated"

#### 5.1.1.6.2 Report Standard Events (default)

In this reporting scheme the LMX9838 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.

#### 5.1.1.6.3 Report No Events

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

This filter setting is useful if the LMX9838 is used as cable replacement in front of a microcontroller, where no status event can be interpreted. But the device will still send a UART break when "Transparent Mode" ([Section 7.2.4.1](#)) is left. In addition the LMX9838 will still recognize a UART BREAK and leave "Transparent Mode" ([Section 7.2.4.1](#)) when detected. In this setting the pin PG6 can be used as hardware indicator about the link status of the LMX9838.

#### 5.1.1.6.4 Report No Events, UART Break Suppressed and Ignored

In addition to the level described in [Section 5.1.1.6.3](#), also the UART BREAK is suppressed and ignored.

This filter setting is useful if the LMX9838 is used as cable replacement in front of a microcontroller, where no status event and even no UART break can be interpreted. In addition the LMX9838 will not recognize a UART BREAK, therefore will not leave "Transparent Mode" (Section 7.2.4.1).

In this setting only pin PG6 can be used as hardware indicator about the link status of the LMX9838.

## 5.1.2 LMX9838 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.

### 5.1.2.1 Local Bluetooth Device Address

During production, TI stores in internal EEPROM the BD\_Addr which is an official and unique identifier assigned by the IEEE.

---

**NOTE:** **DO NOT CHANGE THE PROGRAMMED BD ADDRESS unless you are able to restore it afterwards! The BD address is not restored by Factory Settings!**

---

For prototyping and testing only, if necessary this value can be changed to any specific value. The commands available are:

- "Read Local Bluetooth Address"
- "Change Local Bluetooth Address"

---

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

---

### 5.1.2.2 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"
- "Write Local Name"

### 5.1.2.3 Class of Device

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

A complete list of numbers and can be found in the "Bluetooth Assigned Numbers" Document provided by the Bluetooth SIG at <https://programs.bluetooth.org/Technical/AssignedNumbers/baseband.htm>

Some Examples:

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

---

**NOTE:** the values should be seen as examples. Default value is 00 00 00. (no specific device)

---

Commands available:

- "Store Class of Device"

### 5.1.2.4 Automatic Operation

The Automatic Operation setting has influence on the behavior of the LMX9838 in different situations.

Please see [Section 3.2.4](#) for a detailed description of the different operation states.

Commands available:

- “Read Operation Mode”
- “Write Operation Mode”

#### 5.1.2.5 Fixed Pin

The LMX9838 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”
- “Set Fixed Pin”

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

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

#### 5.1.2.6 Configuring the Default Link Timeout

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

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

A slave will not be able to accept an incoming connection until the link is completely dropped. As it might be useful in certain application to give up a link earlier than 20 seconds, the LMX9838 gives the ability to configure the supervision timeout for each link. It can either be configured as default value in NVS or only for the existing SPP link.

---

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

---

##### 5.1.2.6.1 Setting the Default Link Timeout

The default Link Supervision timeout for all devices is 20 seconds. As the baseband calculates in slots, the value needs to be stored in number of slots. Therefore the default value used is 0x7D00. The default link supervision timeout is used for each incoming and outgoing link. The value stored in NVS can be changed anytime and will be active for the next link establishment without Reset.

- “Set Default Link Timeout”
- “Get Default Link Timeout”

##### 5.1.2.6.2 Changing the Link Timeout of an Existing Link

After a successful link establishment both parties of a link agreed on a specific link timeout. This timeout can be read back or changed anytime by the following commands. In case the local LMX9838 has the role slave, the parameter will only be set locally. In case the LMX9838 is master for the link, the timeout will also be communicated to the connected device, which then will adjust its timeout to that value as well. See also [Section 5.1.2.6.3](#) for the difference between master and slave.

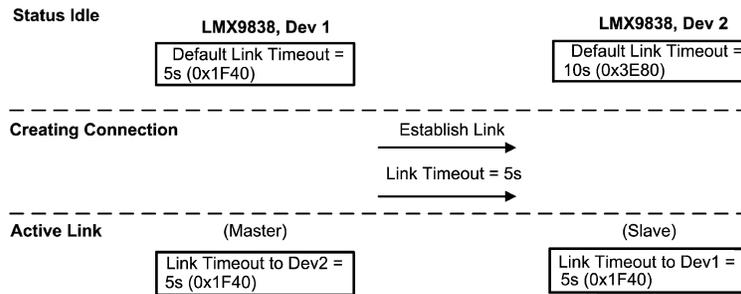
- “Set Link Timeout for an existing link”
- “Get Link Timeout of an existing link”

**5.1.2.6.3 Difference Between Master and Slave Role for the Link Timeout**

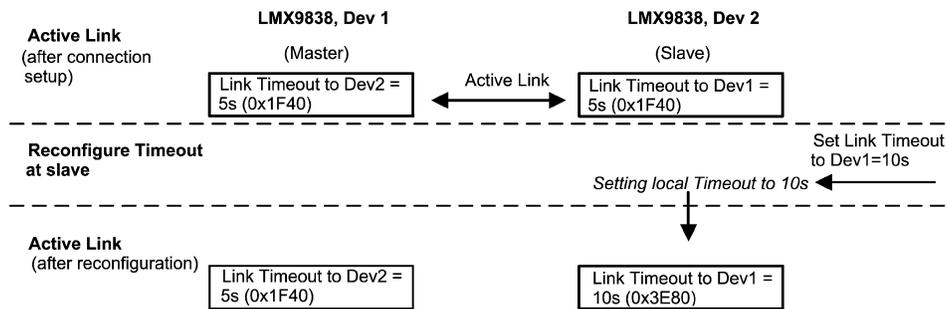
The link timeout is a local parameter stored for each specific link. Therefore a master, which is connected to several devices can have different link time-outs set for each of those links.

The difference between a master and slave is, that a master reports a change of the link timeout to the slave, which then adjust its own timeout to the reported value. In case the “Set link timeout” command will be sent over UART to a LMX9838 in slave role, the setting will only be active locally and no message will be sent to the master. This means the two devices will have different link timeout settings.

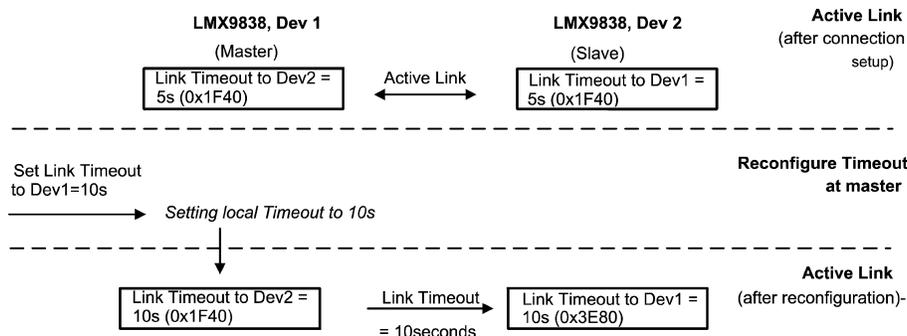
The following figures shall demonstrate the influence of default link timeout and set link timeout, in case they are set on master or slave.



**Figure 11. Master Default Link Timeout**



**Figure 12. Set Link Timeout at Slave**



**Figure 13. Set Link Timeout at Master**

### 5.1.2.7 Configuring the Link Latency

The link latency is part of the Quality of Service bluetooth is able to offer. The link latency results in the so called “poll period”.

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

In case the slave needs guaranteed data transmission lower than the default 40 slots, the link latency parameter for this link needs to be reduced. The LMX9838 offers the ability of configuring the default poll period used for each link. The parameter is stored within the NVS and will be requested for any incoming or outgoing link. Since a master might need to manage several slaves, the parameter has to be seen as request, it can not be 100% guaranteed that exactly this value will be used.

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

- “Set Default Link Latency”
- “Get Default Link Latency”

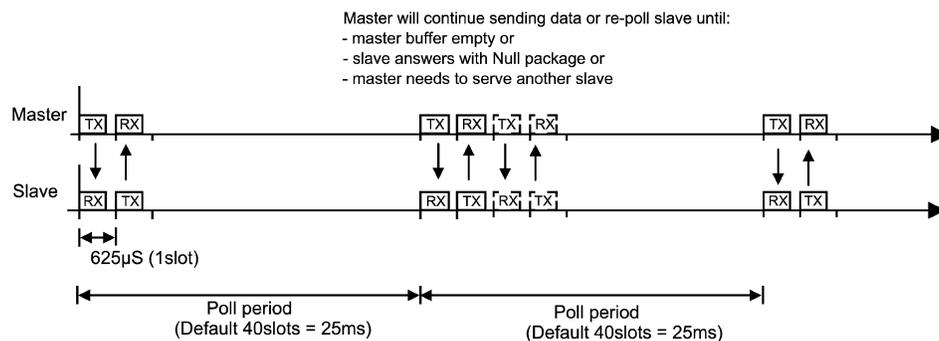
By default the link latency will be set to 0x0000, which means “No specific requirement”. With the master will start sending or requesting data from the slave in a 25ms period. Once the period is reached the device (master or slave) will send the data in a package large enough to carry the complete buffer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

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

---

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

---



**Figure 14. Poll Period**

### 5.1.2.8 RFComm Ports to Open

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

The LMX9838 can handle up to seven links simultaneously. Each link requires an RFComm port opened, which creates a buffer instance within the RAM to handle the upcoming data traffic. Because of that by default only RFComm port 1 is activated.

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

- [Section 7.2.12](#)

The ports in this command are expressed by a 32-bit mask indicating which RFCOMM ports the LMX9838 has to open. Bit 30 and 31 must be set to 0. Bit 0 is RFCOMM port 1 and bit 29 is port 30. The changes take effect as soon as the command has been confirmed.

Examples:

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

So to set up 3 Links, Ports to open could for example be configured to 0x00000007. The ports to be opened also depends, how many services are offered to other devices and which ports are bound to these services. Please see also [Section 6](#) for details.

**Table 41. Open 3 RFCOMM Ports**

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

### 5.1.2.9 Service Database

The LMX9838 contains a service database which configures the settings for authentication and encryption for that specific port. The configuration includes rules for incoming but also outgoing links. Once initialized the service database serves this service requests automatically, without any interaction required with the host.

In principle, the service database is only necessary to be able to offer services to remote devices. E.g. if a remote device creates a SDAP link to the LMX9838 and browses the services, it will only see the entries stored within this database. If for example a second profile needs to be offered, e.g. a second SPP called COM2, it needs to be entered into that database. See also [Section 7.2.13](#) for details modifying the "Service Database" ([Section 5.1.2.9](#)). For more details on SDP records parameters refer to the Bluetooth Specification.

By default the following entry is stored within the LMX9838 for RFCOMM Port 1:

**Table 42. Default SDP Entry on RFCOMM Port 1**

Parameter	Description	Value
Entry Index	Index at which the entry can be addressed for enabling or disabling it	0x00
RFCOMM Port	RFCOMM Port the settings refer to	0x01
Profile to be offered	The profile to be offered to the remote device. Profiles different from SPP need to be implemented on the host	SPP
Entry Name	Name which will be shown to the remote device	COM1
Authentication	Defines if authentication is required if port is used for an incoming or outgoing connection	0x02
	0x00 No authentication requirements	
	0x02 Authentication is only required for this profile for incoming connections	
	0x20 Authentication is only required for this profile for outgoing connections	
	0x22 Authentication is required for this profile for connections in both directions	
See also <a href="#">Section 7.2.13.1</a>		
Encryption	Defines if encryption is required if port is used for an incoming or outgoing connection	0x04
	0x00 No encryption requirements	
	0x04 Encryption is only required for this profile for incoming connections	
	0x40 Encryption is only required for this profile for outgoing connections	
	0x44 Encryption is required for this profile for connections in both directions	
See also <a href="#">Section 7.2.13.1</a>		

**NOTE:** The service database security settings are only used when the device is configured to security level 2 (default). Please see also [Section 7.2.16](#).

[Table 42](#) shows the default service database entry offered to a remote device. The configuration shows that an incoming link which will address RFComm port 1 will require authentication and encryption. In case the link establishment has not happened before, the LMX9838 will automatically initiate a pin code request to the remote device.

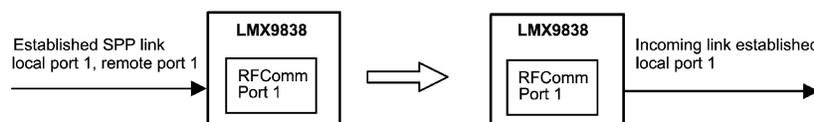
In general, for outgoing connections, there's no service database required for that specific RFComm port. For example, if a link is established from local RFComm port 2 to a remote port and no SDB entry has been made for that port, the default will be that no authentication and no encryption will be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

## 5.2 SETTING UP MULTIPLE CONNECTIONS

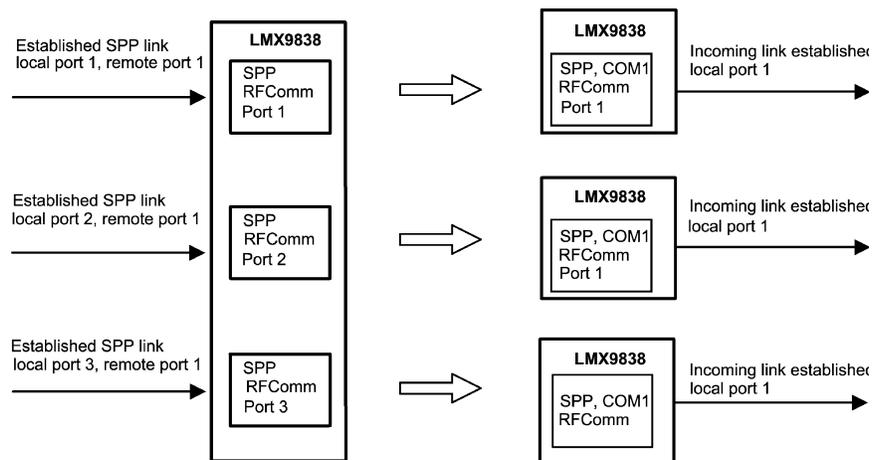
As already described in [Section 4.2](#), the LMX9838 command interface offers the ability to search for other devices, browse the services and to establish a link to another device.

The LMX9838 bluetooth operation is based on the Serial Port Profile. This profile emulates a serial port over a bluetooth link. As bluetooth is able to handle more than one links, the LMX9838 will also offer multiple communication ports, also called RFComm ports to the host. For each port a separate RFComm instance needs to be created.

[Figure 15](#) shows the standard link establishment as already described in [Section 4.2](#). This establishment uses the standard configurations for the LMX9838 with one RFComm port on the device.



**Figure 15. Standard Point-To-Point SPP Connection**



**Figure 16. Multiple SPP Links from LMX9838**

[Figure 16](#) shows the principle of establishing multiple links with the LMX9838. The LMX9838 on the left has to initialize 3 RFComm ports, which then can be addressed and bound to a remote device. Any data sent to that specific port are then sent to the device connected to it.

The figure also shows the term “SPP, COM1” which is an indicator for the service database entry stored for this specific port. By default the LMX9838 has configured a Serial Port Profile (SPP) to RFComm port 1. The name, which will be reported to a browsing device is set to “COM1”.

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

## 5.2.1 Configuration

The LMX9838 by default is configured for point-to-point operation only. Therefore some configuration settings have to be changed before it is possible to establish multiple links.

### 5.2.1.1 RFComm Ports to Open

As explained in [Section 5.1.2.8](#) the LMX9838 can handle up to seven links simultaneously. Each link is handled by a RFComm port which creates a buffer instance within the RAM to handle the upcoming data traffic. Because of that by default only RFComm port 1 is activated.

As we need to support three links in this example, ports 1,2 and 3 shall be used. For this, ports to open has to be configured to 0x00000007.

**Table 43. Open RFComm Ports 1, 2 and 3**

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

### 5.2.1.2 Configuring Service Database for Outgoing Connections

As explained in [Section 5.1.2.9](#) the LMX9838 contains a service database which configures the settings for authentication and encryption for that specific port. The configuration includes rules for incoming but also outgoing links.

In principle, for simple outgoing SPP connection on a port different to RFComm Port 1 it is not required to enter an additional service database entry. The LMX9838 will use the default setting, assuming no auth/encr required for the outgoing link. However, service entries for specific profiles, include also feature settings, which are also checked on outgoing links, e.g. Headset or handsfree profile entries include information on volume control or other features.

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

**Table 44. SDB Entry with Security on Outgoing Connection**

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

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

- "Store Generic SDP Record" ([Section 7.2.13.1](#))

The changes take effect immediately after getting the confirmation event.

**Table 45. Example SDP Store SPP Record**

Byte	Parameter	Value
01	Start Delimiter	02
02-06	Package Header	52 31 4C 00 CF
07	Local RFComm Port	02
08	Authentication setting	20

**Table 45. Example SDP Store SPP Record (continued)**

Byte	Parameter	Value
09	Encryption setting	40
10-00	Service Name length	47 00
12-82	Service Record	00 00 0A 00 00 00 00 01 00 35 03 19 01 11 04 00 35 0C 35 03 19 0 0 01 35 05 19 03 00 08 02 05 00 35 03 19 02 10 06 00 35 09 09 6E 65 09 6A 00 09 00 01 09 00 35 08 35 06 19 01 11 09 00 01 00 01 2 5 05 43 4F 4D 32 00
83	End Delimiter	03

### 5.2.2 Link Establishment

Establishing links to multiple slaves in principle is the same as a standard connection setup already described in [Section 4.2](#). The only difference which has to be considered is that the LMX9838 has enough RFCComm ports available (opened) and each link is assigned to one specific RFCComm port.

The link establishment uses the command "Establish Link" ([Section 7.2.3.1](#)) to create all links. Each link established will be confirmed by the appropriate confirmation event. The following three tables show the commands which have to be sent to establish the three links. The commands need to be adjusted for the local RFCComm Port, the remote BD\_ADDR and the remote RFCComm Port Number, derived out of the related SDAP request.

**Table 46. Establish Link on Local RFCComm Port 1**

Byte	Parameter	Value
01	Start Delimiter	02
02-06	Package Header	52 0A 08 00 64
<b>07</b>	<b>Local RFCComm Port</b>	<b>01</b>
08-13	Remote BD_Addr	12 34 56 78 9A BC
14	Remote RFCComm Port	01 (out of SDAP Request)
15	End Delimiter	3

**Table 47. Establish Link on Local RFCComm Port 2**

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

**Table 48. Establish Link on Local RFCComm Port 3**

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

### 5.2.3 Summary

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

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

#### 5.2.3.1 Device Configuration, Preparing for Multipoint Operation

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

**Table 49. Initialize 3 RFComm Ports**

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

**Table 50. Configure Additional SDB Entry to Enable Auth/encr. for Outgoing Connection**

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
	TX	Request	02,52,31,4C,00,CF,02,20,40,47,00,00,00,0A,00,00,00,00,01,00,35,03,19,01,11,04,00,35,0C,35,03,19,00,01,35,05,19,03,00,08,02,05,00,35,03,19,02,10,06,00,35,09,09,6E,65,09,6A,00,09,00,01,09,00,35,08,35,06,19,01,11,09,00,01,00,01,25,05,43,4F,4D,32,00,03	Tx: Cmd: Store SDP Record, Local Port: 02, Authentication: 20, Encryption: 40, SdpRecord: 00000A000000000010035031901110400350C3503190001350519030008020500350319021006003509096E65096A0009000109003508350619011109000100012505434F4D3200
00 / 01	RX	Confirm	02,43,31,02,00,76,00,01,03	Rx: Event: Store SDP Record, Status: 00, Identifier: 01

#### 5.2.3.2 Discover Devices

To get the devices BD\_Addresses an Inquiry has to be started.

**Table 51. Device Discovery**

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

#### 5.2.3.3 Get Remote RFComm Ports

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

**Table 52. Get Remote RFComm Port of First Device**

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

**Table 52. Get Remote RFComm Port of First Device (continued)**

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

**Table 53. Get Remote RFComm Port of the Second Device**

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

**Table 54. Get Remote RFComm Port of the Third Device**

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

### 5.2.3.4 Establish Links

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

**Table 55. Establish Links to All Three Slaves**

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

### 5.3 DEFAULT CONNECTIONS

The LMX9838 offers a default connection procedure which allows to store up to seven devices into a database-like system within the device. The stored connections are either connected after a reset or by sending the “Connect default connections” command. This allows an easy cable replacement setup but can also be used for automatic multipoint operations.

The successful or failed link establishment will be confirmed for each stored device. If set to Operation Mode “Automatic” the LMX9838 after reset or boot-up will try to connect to each of those devices 3 times then switch to the next (see [Section 3.2.4](#)).

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

Commands available:

- “Store Default Connection”
- “Connect to Default Connection”
- “Get List of Default Connections”
- “Delete Default Connection”

**NOTE:** For multiple connections please make sure that the NVS parameter "Ports to open" (See [Section 7.2.12.1](#)) is configured correctly to have the appropriate RFComm port instances initialized.

## 5.4 LOW POWER OPERATION

### 5.4.1 Hardware Power Management

The LMX9838 power management is a combination of the firmware and the hardware supported low power modes. Depending on the system activity it decides to switch off as many hardware blocks as possible to reduce the current consumption.

#### 5.4.1.1 Low Power Modes

The LMX9838 power management can be divided into six modes, which depend on the activity level of the UART interface and the bluetooth radio.

- UART Interface activity
  - — Enabled: UART fully active, RTS and CTS used for flow control (hardware handshaking)
  - — Disabled: UART disabled, RTS/CTS used for "Wake up functionality" ([Section 5.4.1.3](#))
- Bluetooth radio activity
  - — Active Link(s): Bluetooth radio and baseband handling active link, "Bluetooth Low Power Modes" ([Section 5.4.2](#)) can be used to reduce power consumption on radio.
  - — Page/Inquiry Scanning: LMX9838 only scanning, discoverable/connectable for other devices, no active link
  - — No Radio activity: no active link and scanning switched off

**Table 56. Low Power Modes**

UART / Radio	No Radio Activity	Page/Inquiry Scanning	Active Link(s)
Disabled	PM0	PM2	PM4
Enabled	PM1	PM3	PM5

- PM0:
  - — Lowest power consumption, Sleep Mode
  - — UART disabled (using "Disable Transport Layer" ([Section 7.2.9.1](#)), "Wake up functionality" ([Section 5.4.1.3](#)) enabled
  - — Scanning disabled (using "Set Scan Mode - Discoverability / Connectability" ([Section 7.2.14.6](#)), parameters 0x00,0x00), Device not discoverable/connectable for other devices
  - — No active bluetooth link
- PM1:
  - — UART enabled, device listening to commands ("Command Mode" ([Section 2.3.2.1](#)))
  - — Scanning disabled (using "Set Scan Mode - Discoverability / Connectability" ([Section 7.2.14.6](#)), parameters 0x00,0x00), Device not discoverable/connectable for other devices
  - — No active bluetooth link
- PM2:
  - — Typical standby mode
  - — UART disabled (using "Disable Transport Layer" ([Section 7.2.9.1](#))), "Wake up functionality" enabled ([Section 5.4.1.3](#))
  - — Scanning enabled (using "Set Scan Mode - Discoverability / Connectability" ([Section 7.2.14.6](#))), Device discoverable/connectable for other devices
  - — No active bluetooth link, LMX9838 waking up host on incoming link
- PM3:

- Default mode after boot-up
- — UART enabled, device listening to commands ("Command Mode"([Section 2.3.2.1](#)))
- — Scanning enabled (using "Set Scan Mode - Discoverability / Connectability" ([Section 7.2.14.6](#))), Device discoverable/connectable for other devices
- — No active bluetooth link
- PM4:
  - — UART disabled (using "Disable Transport Layer" ([Section 7.2.9.1](#))), "Wake up functionality" enabled ([Section 5.4.1.3](#))
  - — Scanning has no influence on power management
  - — Active bluetooth link, LMX9838 waking up host on another incoming link or incoming data
  - — Power consumption can be reduced by "Bluetooth Low Power Modes" ([Section 5.4.2](#))
- PM5:
  - — Full activity mode
  - — UART enabled, device listening to commands ("Command Mode" ([Section 2.3.2.1](#))) or in "Transparent Mode" ([Section 2.3.2.2](#))
  - — Scanning has no influence on power management
  - — Active bluetooth link(s)
  - — Power consumption can be reduced by "Bluetooth Low Power Modes" ([Section 5.4.2](#))

PM0 is lowest power, PM5 can be seen as highest power consumption mode. As indicated, the host is able to influence the power consumption by either switching off scanning using "Set Scan Mode - Discoverability / Connectability" ([Section 7.2.14.6](#)) or by disabling the UART, using "Disable Transport Layer" ([Section 7.2.9.1](#)) to enable the "Wake up functionality" ([Section 5.4.1.3](#)).

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

In order to save power in a waiting situation, the host might decide to put the device from PM3 to PM2, using the "Disable Transport Layer" command ([Section 7.2.9.1](#)). The device will still be available for connections and will wake up the host on an incoming link.

If the LMX9838 is already engaged in a link, the host can decide to use "Disable Transport Layer" ([Section 7.2.9.1](#)) to enable "Wake up functionality" ([Section 5.4.1.3](#)). This would allow the host to power down and just wait for incoming data even on an active link.

To reach a low power sleep mode (starting from PM3), the host needs to first shut down the radio activity by disabling the scanning using "Set Scan Mode - Discoverability / Connectability" ([Section 7.2.14.6](#)). This forces the device switching to PM1. After that the host can use "Disable Transport Layer" ([Section 7.2.9.1](#)) to switch off also the UART activity. The LMX9838 will activate the "Wake up functionality" ([Section 5.4.1.3](#)) on the UART and go to a very low power mode.

[Figure 17](#) gives an overview of all possible transitions between the different power modes and what action triggers the change.

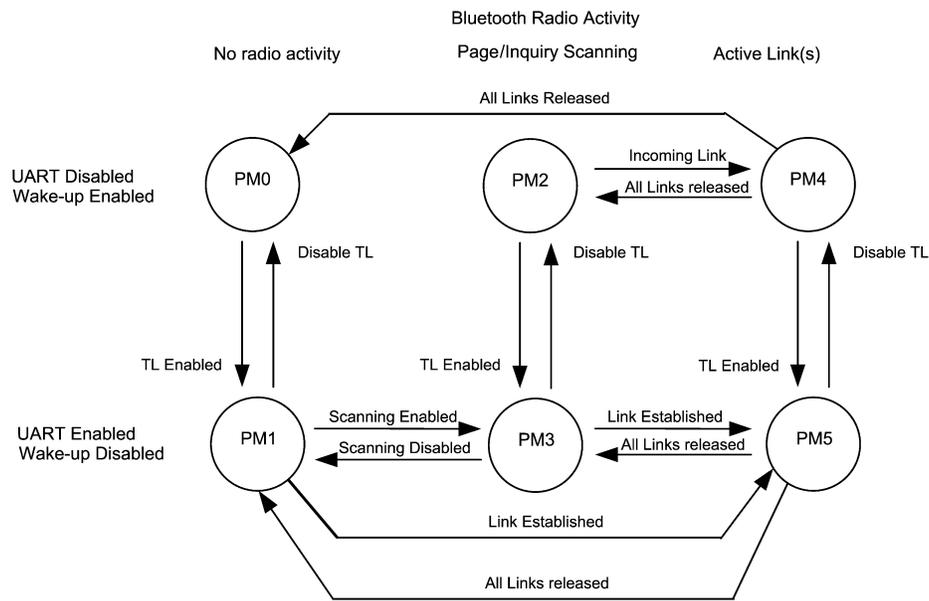


Figure 17. Transition Between Different Hardware Power Modes

### 5.4.1.2 Enhanced Power Management

The LMX9838 is able to reduce the power in the waiting state PM2, using an external 32.768 kHz crystal. If such a crystal is available and connected, the baseband is able to shut down the radio completely between the scanning intervals and with this to drop the power consumption to a few hundred Micro Ampere.

The usage of the 32.768 kHz is controlled by the Enhanced PowerManagement (PMM) bit in NVS. By default PMM is disabled, so the 32.768 kHz is not used even if connected. The PMM bit can be enabled by the "Write NVS" command (Section 7.2.22.2), configuring bit 1 of the system parameter at address 0x0062 to 0.

### 5.4.1.3 Wake Up Functionality

In certain applications the LMX9838 will be used most of the time in a waiting status, meaning it is waiting for being connected or listening to commands. To reduce power consumption of the system, the LMX9838 supports a specific Wake up functionality.

The LMX9838 supports to disable the UART transport layer (command "Disable Transport Layer" (Section 7.2.9.1) to switch off the command interpreter and all hardware components not needed for the current operation. The interface can be reactivated again by either side by using hardware pins.

The LMX9838 uses the RTS signal pin to wake up the host. The RTS / CTS signals are connected in a NULL-Modem fashion, meaning that RTS on the Host is connected to CTS on the LMX9838 and vice versa. Therefore the host would need to be able to monitor its CTS input or has to use a separate hardware pin. In case the LMX9838 has to be triggered by the host, the RTS pin is used as the hardware Wake-Up signal.

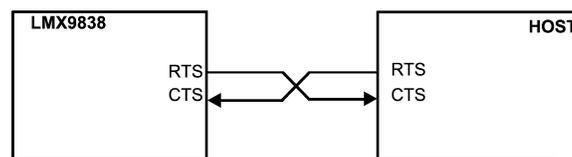


Figure 18. Host Wake Up Hardware Connections

### 5.4.1.3.1 Disabling the Transport Layer

The transport layer can be disabled in any operation state. The LMX9838 will try to wake up the host as soon as the interface would be needed again. To disable the transport layer the following command has to be used:

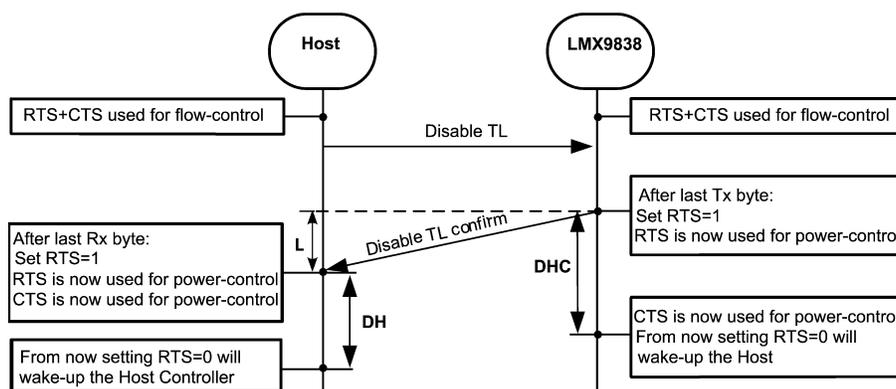
- “Disable Transport Layer”

The command will be confirmed with the standard “Disable Transport Layer” Confirm event. After the command has been sent, the LMX9838 will switch the functionality of the RTS and CTS pins from normal hardware flowcontrol to the wake-up functionality.

Both the Host and the LMX9838 shall set RTS=1 since they may be in a sleep mode and thus both are “Not Ready to Receive”. The HW Wake-Up signal is then defined as a falling edge on the CTS input i.e. a device wakes up the other device by asserting its own “Ready to Receive” output (i.e. setting RTS=0).

If the LMX9838 redefines the CTS input from “flow-control” input to “wake-up” input there will be a short period of time during which the signalling is ambiguous. To avoid this, delays are introduced as illustrated in Figure 19.

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



**Figure 19. Disable Transport Layer Timing**

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

**DHC** is the time period the LMX9838 must delay redefining CTS to Wake-Up input.

**DH** is the time period the Host must wait before attempting to send a Wake-Up signal to the LMX9838 by setting RTS = 0.

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

$$L_{max} \geq L \geq L_{min}$$

$$DHC \geq L_{max}$$

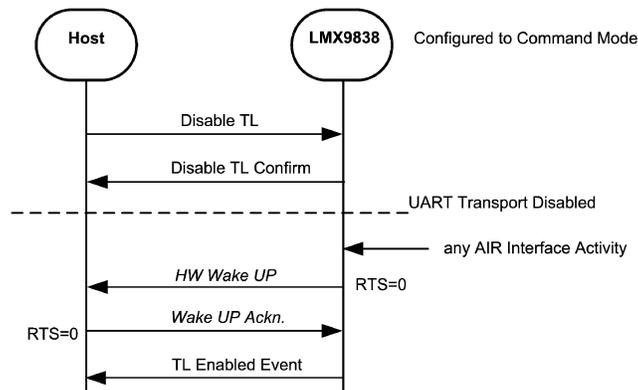
$$DH \geq DHC - L_{min}$$

### 5.4.1.3.2 LMX9838 Wakes Up Host

If the LMX9838 needs to send data to the host it must first make sure that the UART transport layer is enabled. If UART is disabled the LMX9838 assumes that the host is sleeping and starts the wake-up by setting RTS to 0. To be able to react on that Wake-up, the host has to monitor the CTS pin.

On receiving the CTS falling edge the host has to wake up its UART interface and switches the RTS/CTS pin functionality back to normal hardware handshake. As soon as RTS is 0, the LMX9838 will confirm the wake-up by sending the TL Enabled Event and start sending the pending events that triggered the wake-up.

Please see Figure 20 for the complete process.

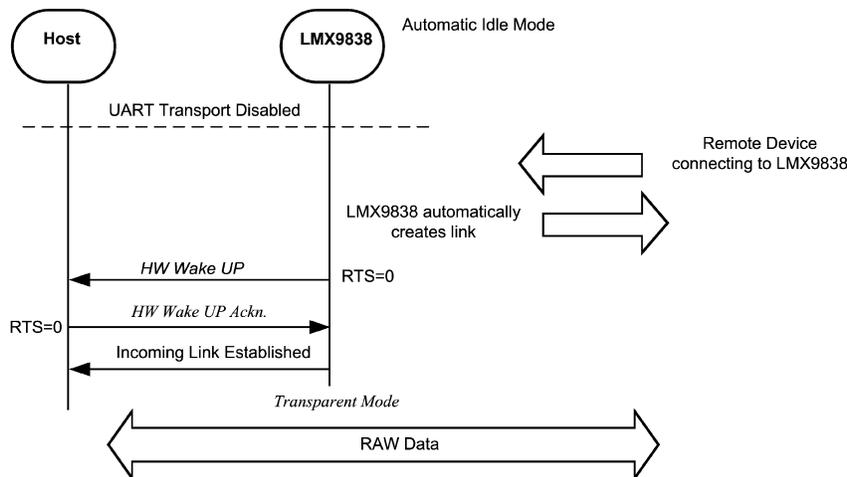


**Figure 20. LMX9838 Waking Up the Host in Command Mode**

**NOTE:** Even though the LMX9838 signals RTS=0 (i.e. “Ready to Receive”) the Host may not send any data or commands to the LMX9838 before it has received the “Transport Layer Enabled” event.

The event will always be received as soon as the LMX9838 re-enables the UART interface again. One exception is if the device is in “Idle Automatic” state. If the LMX9838 is operating in default idle automatic state waiting for being connected and the transport layer was switched off, it will not send the TL Enabled Event. As soon as it gets connected the LMX9838 will request the wake up and directly send the “Link Established” after successful connection establishment.

Please see [Figure 21](#) for that specific case.



**Figure 21. No TL Enabled Event After Automatic Idle**

### 5.4.1.3.3 Host Wakes Up LMX9838

If the host needs to send data or commands to the LMX9838 it must first make sure that the UART transport layer is enabled. If UART is disabled the host must assume that the host is sleeping and starts the wake-up by setting RTS to 0. The LMX9838 will wake up to the falling edge of its CTS pin.

When the LMX9838 detects the Wake-Up signal it activates the UART HW and acknowledges the Wake-Up signal by sending a “Transport Layer Enable” event. When the Host has received the “Transport Layer Enable” event, the LMX9838 is ready to receive commands.

This process is shown in the following figure.

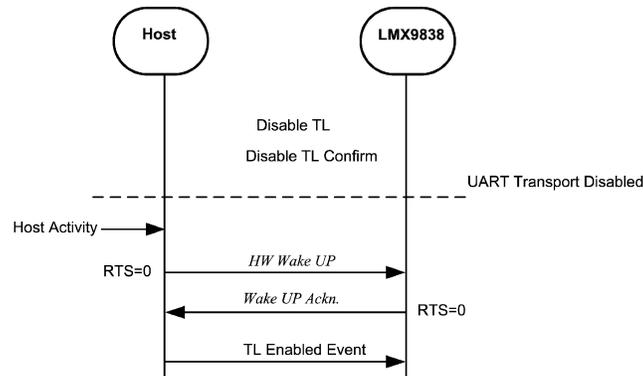


Figure 22. Host Waking Up LMX9838

**NOTE:** Even though the LMX9838 signals RTS=0 (i.e. “Ready to Receive”) the Host may not send any commands to the LMX9838 before receiving the Transport Layer Enabled event.

### 5.4.2 Bluetooth Low Power Modes

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

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

The LMX9838 supports all Low Power Modes.

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

#### 5.4.2.1 Link Policy

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

The following features can be switched on or off:

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

#### 5.4.2.1.1 **Default Link Policy Setting**

The LMX9838 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”
- “Get Default Link Policy”

#### 5.4.2.1.2 **Setting Link Policy in an Active Link**

In an active link, each of the settings can be switched on or off by the “Set Link Policy” Command. On default the LMX9838 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”
- “Get Link Policy”

#### 5.4.2.2 **Sniff Mode**

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

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

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

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

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

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

Commands available:

- “Enter Sniff Mode”
- “Exit Sniff Mode”

#### **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: 625 $\mu$ s.

$300\text{ms} / 625\mu\text{s} = 480 \text{ slots} = 1\text{E}0 \text{ hex}$

=> maximum Sniff interval: 01E0 hex

$50\text{ms} / 625\mu\text{s} = 80 \text{ slots} = 50 \text{ hex}$

minimum Sniff Interval: 0050 hex

#### b) Sniff Attempts and Timeout

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

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

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

Note that Nsniff attempt = 0 is not allowed.

For the example the following parameters have been successfully tested:

Sniff attempts: 5 slots

Sniff timeout: 3 slot

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

---

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

---

#### 5.4.2.3 Hold Mode

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

After that time they start normal transmission again.

The Hold Mode uses the following parameters:

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

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

Commands available:

- “Enter Hold Mode”

Please see [Section 5.4.2.2](#) for the calculation of the Hold mode parameters.

#### 5.4.2.4 Park Mode

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

The following parameters are used for Park Mode:

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

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

Available commands:

- “Enter Park Mode”
- “Exit Park Mode”

---

**NOTE:** Since broadcast packages are not acknowledged and they are not seen as data traffic, a parked linked will be dropped after the supervision timeout. To keep a parked device connected, it has to be unparked and parked within the supervision timeout. See also [Section 5.1.2.6](#) for details on the supervision timeout.

---

## 5.5 ESTABLISH AUDIO LINKS

### 5.5.1 Bluetooth Background

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

SCO package types:

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

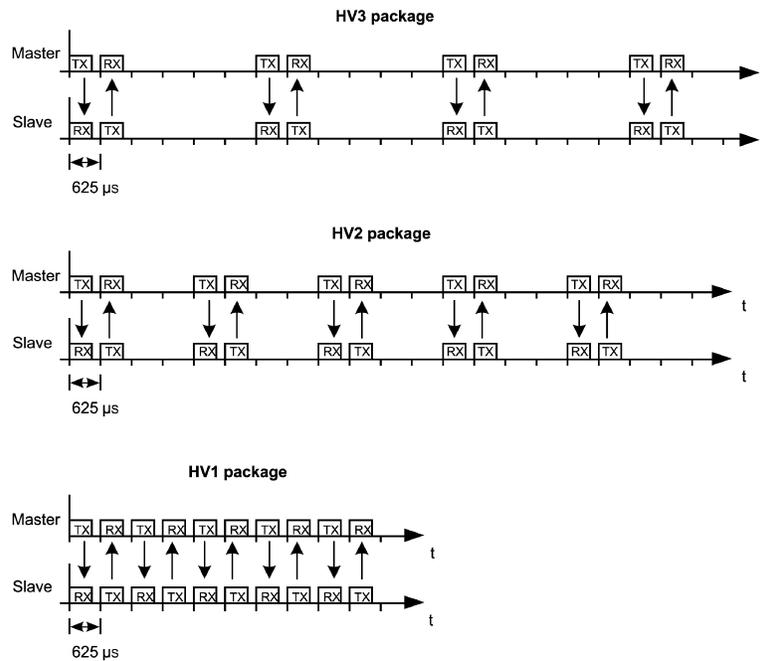


Figure 23. SCO Link Bandwidth Requirements

### 5.5.2 Routing of Audio Data

As mentioned in Section 2.4, the LMX9838 offers the Advanced audio interface (AAI), also called PCM interface, in order to connect an external PCM codec. The codec is used to convert the analog microphone signals into the digital PCM stream, which is then directly routed to an existing bluetooth SCO link. The same way, any incoming SCO data will be directly routed to the PCM interface to be converted into analog loudspeaker signals by the external codec.

Each SCO link is based on a previously established ACL link. The ACL link is used for the standard SPP profile. Because of this it is possible to have simultaneous data and voice transmissions. See Figure 24 for the split between SCO and ACL/SPP data routing.

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

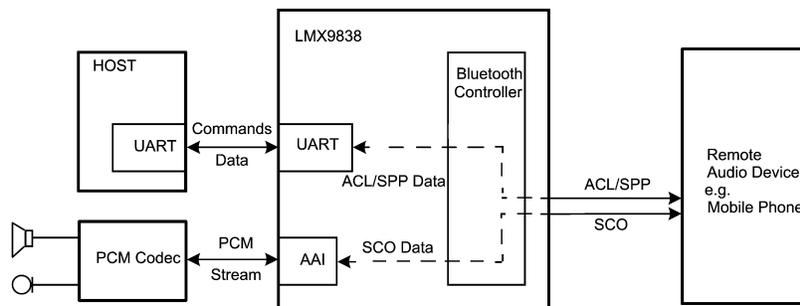


Figure 24. SCO and ACL/SPP Routing Within LMX9838

### 5.5.3 Audio Link Commands

The LMX9838 includes all functionality to establish and accept SCO links and routing the data to the PCM using the settings configured in the NVS. The settings for audio codec and air format can either be configured by default or for after link establishment.

#### 5.5.3.1 PCM Codec Configuration

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

- "Set audio settings"
- "Get audio settings"

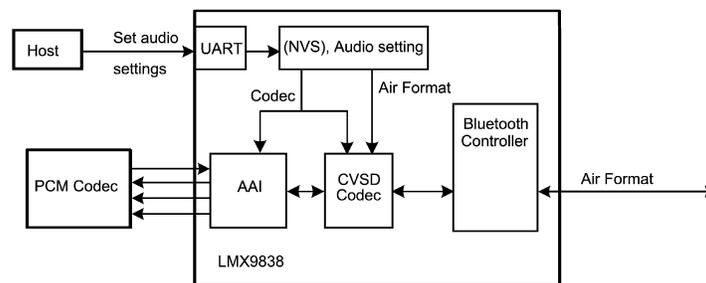
This command defines which format should be used at the PCM interface (codec driver or PCM slave) is used as well as the format used on the bluetooth link. This command only sets the parameters for the existing link.

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

- "Set Default Audio Settings"
- "Get Default Audio Settings"

Please see also [Figure 25](#) for the influence of both parameters to the blocks of the LMX9838. The codec setting configures the AAI interface timing to be able to interact with the external codec. The air format configures which format will be used on the bluetooth link. Both parameters influence the configuration of the internal CVSD codec, which will take care of the conversion to the appropriate formats.

In case PCM slave is used, these settings have to be stored separately. Please refer to [Section 5.5.3.1.2](#) for details.



**Figure 25. Influence of Audio Settings to LMX9838 Blocks and Formats**

#### 5.5.3.1.1 Using a Specific Codec Driver

The LMX9838 includes two specific drivers, which are optimized for the Motorola MC145483/Winbond W681360 codec or the OKI MSM7717 / Winbond W681310 codec.

The interface settings depend on the external clock used for the LMX9838. The following parameters have to be considered to interface with the AAI in that configurations:

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

In general these settings define the signalling on STD and SRD pins for the data. [Figure 26](#) shows the principle for the relation between frame sync signal and a data slot. The frame sync signal is typically raised in a frequency of 8 kHz. The data are sampled at the bit clock provided by the LMX9838 at pin SCLK.

A complete list of bit clock and format information for all codecs can be found in "Texas Instruments: LMX9838 Datasheet".

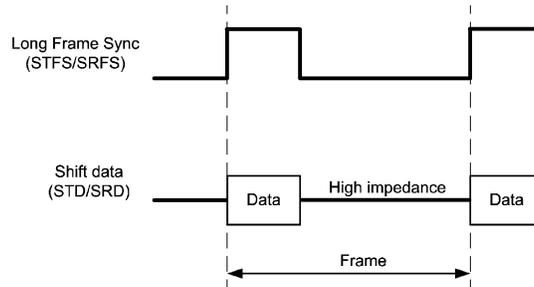


Figure 26. AAI Generic Data Framing

**Example:**

Motorola MC145483 at 13Mhz external clock.

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

The Motorola codec supports 3-bit audio control at the end of each data word. This information is used by the codec to control the volume. The codec requires a 13-bit linear data format. Figure 27 shows the generic setup of the 16-bit data word for the Motorola codec.

Figure 28 gives the overview and the relation between the actual SFS signal and the data slot. Figure 29 finally shows the real transmission, with actual bit clock and the 13-bit data word.

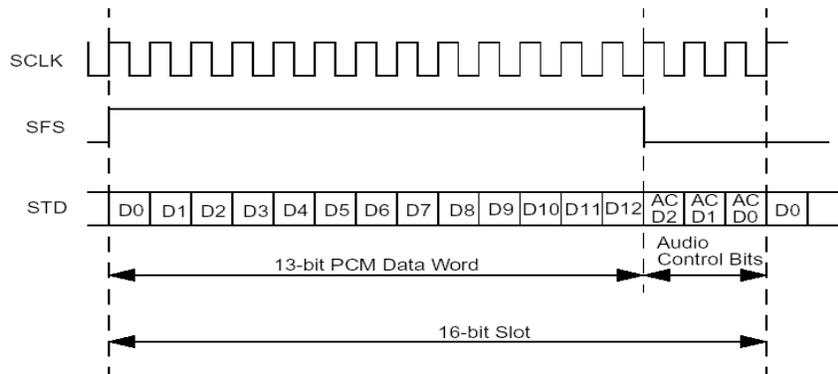


Figure 27. Theoretical AAI Interface Stream at Motorola Settings (Long Frame Sync)

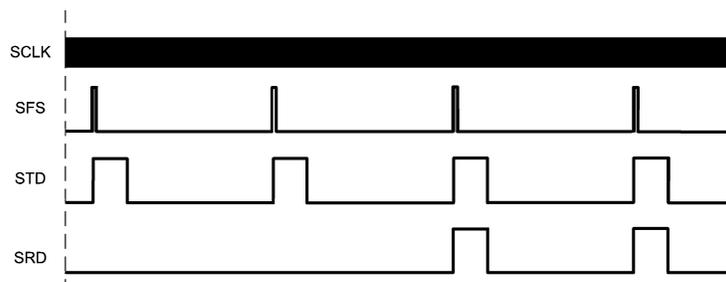
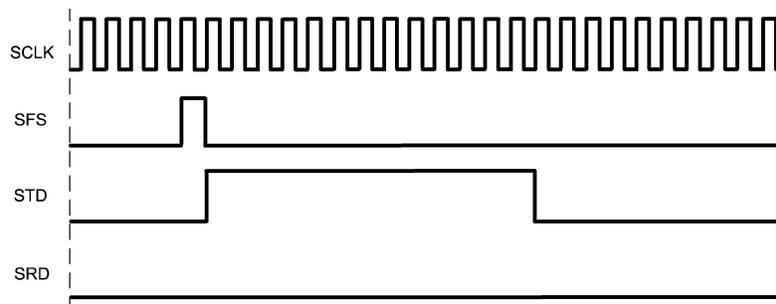


Figure 28. AAI Stream Overview for Motorola Codec (Short Frame Sync)



**Figure 29. AAI Slot Usage for Motorola with Audio Control set to 0x00 (Short Frame Sync)**

### 5.5.3.1.2 Using PCM Slave

Since the LMX9838 is also intended to be used in PDA like devices, the audio interface allows to be driven from an external controller or DSP. This means, the LMX9838 will sample the data from or to the DSP on the Bit clock and Frame clock provided by the DSP.

In order to be able to adjust the internal timing, the LMX9838 audio interface needs to be configured to the specific settings. The settings should be chosen so that the actual frame sync is configured to 8KHz. The bit-clock should not exceed the range of 128 kHz to 1024 kHz.

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

#### 1. Slot Selection

- The AAI supports up to four simultaneous slots of which the LMX9838 will extract and send the data. This allows a DSP to drive multiple codecs the same time, differentiating the codecs on the slot. In a single connection slot 1 should be used.

#### 2. Number of slots per frame

- This configures how many slots should be transmitted per frame. Maximum is four.

#### 3. PCM data format

The data format used. The on-chip codec supports:

- 8-bit A-Law
- 8-bit  $\mu$ -Law
- 13/14/15/16 bit linear

#### 4. Frame sync length

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

#### 5. Data word length

- The LMX9838 supports data word length of 8 or 16bit.

#### 6. Frame sync polarity

- The Frame pulse signal can either be used normal or inverted.

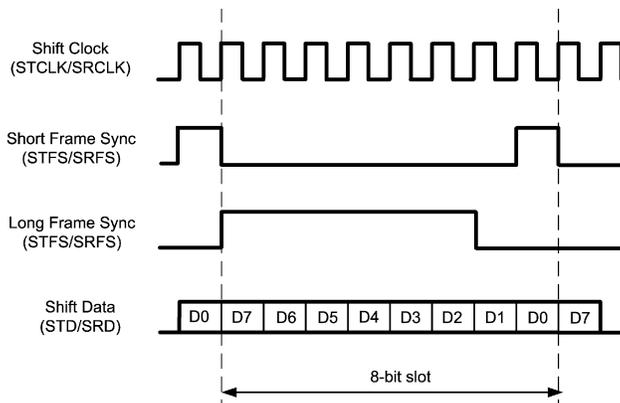


Figure 30. Difference Between Long and Short Frame Sync Signal

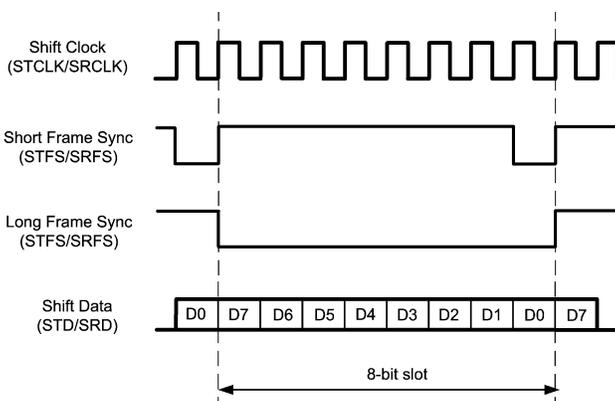


Figure 31. Inverted Frame Sync Pulses

### 5.5.3.2 SCO Link Establishment

In general a SCO link can only be established on top of an existing ACL link. So in order to establish a SCO link to another device, it first has to establish a standard SPP Link, using the "Establish Link" (Section 7.2.3.1) link command. Once this has been set up, a SCO link can be created to that device, using the following command.

- "Establish SCO Link"

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

### 5.5.3.3 Accept Incoming SCO Links

The LMX9838 will automatically accept incoming SCO link requests and confirm it with the indicator as follows. The firmware will use the default audio settings stored in NVS for the PCM codec driver and Bluetooth air format.

- "SCO Link Established Indicator"

### 5.5.3.4 Changing SCO Package Type

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

- "Change SCO Packet Type"

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

- "Change SCO Link Packet Type Indicator"

### 5.5.3.5 **Configuring Volume and Microphone level**

The LMX9838 commandset offers commands to configure the volume of the loudspeaker output on the PCM codec or to Mute the microphone. The commands can only be executed on existing links.

Please see details in the command section:

- "Mute"
- "Set Volume"
- "Get Volume"

### 5.5.3.6 **Releasing a SCO Link**

Releasing a SCO link only requires to send the following release command to the LMX9838. The command doesn't require any other parameter to be set, since the LMX9838 can only support one SCO link at a time anyway.

- "Release SCO Link"

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

- "SCO Link Released Indicator"

### 5.5.4 **Example**

To establish an audio link to other devices, normally the guidelines for a headset or handsfree profile have to be followed. Because of this please refer to [Section 6](#) for details, how to handle audio links.

## 5.6 **SOFTWARE PATCHES**

The LMX9838 firmware is included in ROM and therefore can not be updated. To still be able to correct small errors or to do minor modifications to the software, the LMX9838 offers the so called "Patch" mechanism, which basically allows to replace small parts of the ROM code. This patch code is held in RAM, read from the internal EEPROM of the LMX9838 during boot-up. To apply a new Patch, the Patch file needs to be provided after boot-up. This should be done only once, during the general initialization of the module.

### 5.6.1 **The Patch Engine**

The LMX9838 has HW support for patching the ROM code by replacing segments (512 bytes) of the ROM with segments of the 3kbyte patch RAM. In order to do patching the firmware must load the patch code into the patch RAM and update the patch look-up registers that define which part of the ROM should be patched. [Figure 32](#) shows the HW patch engine implemented in the LMX9838.

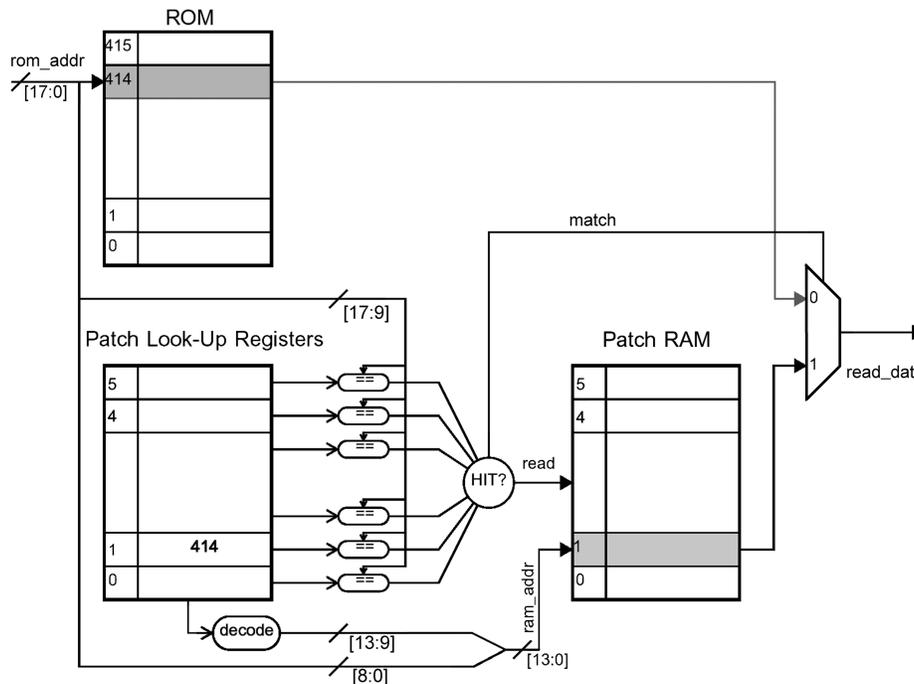


Figure 32. Hardware Patch Mechanism

### 5.6.2 Patch Format

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

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

- File: the patch file, usually delivered by TI. The file needs to be held on the host, which has to push it once into EEPROM.
- UART: indicates, which data are pushed by "Write ROM Patch" (Section 7.2.20.1)
  - X1 = All "global" meta data must be included as parameters in the first patch download command sent
  - X2 = All "segment" meta data for one patch segment must be included as parameter in one patch download command (i.e. the data shall not be split in two commands)
- EEPROM: An X indicates, which data are stored in EEPROM

Table 57. Patch Format at the Different Locations

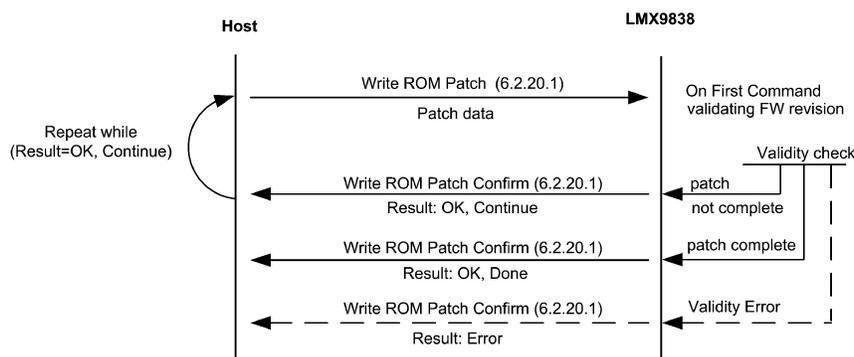
Size:	Name:	Description:	Present In		
			File	UART	EEPROM
uint8	PatchValid	Flag (bit0) indicating whether the NVS contains a valid patch.			X
uint32	PatchSize	Size of patch (including PatchSize field but excluding DescLen and Description fields).	X	X1	X
uint16	FormatRev	Patch format revision, the format specified in this table has revision 0x0100.	X	X1	X
uint16	NumSegs	Number of Patch Code Segments	X	X1	X
uint16	DescLen	Description length	X		
uint8[DescLen]	Description	Textual description of patch	X		
uint32	FWRev	Firmware revision to which this path applies	X	X1	X

**Table 57. Patch Format at the Different Locations (continued)**

Size:	Name:	Description:	Present In			
			File	UART	EEPROM	
uint32	FWPRev	New firmware revision when this path has been applied	X	X1	X	
for (l=0;l<NumSegs;l++)						
uint8	Type	Patch type:	X	X2	X	
		0x00 = HW patch.				
		0x01 = SW patch.				
uint32	RAMAddr	RAM Start address for patch segment.	X	X2	X	
if (Type=0x00) //HW patch						
	uint32	ROMAddr	ROM Start address for patch.	X	X2	X
else //SW patch						
	uint32	VarAddr	Address of function pointer variable for patch.	X	X2	X
uint16	CodeLen	Code length	X	X2	X	
uint16	MetaDataChk	Meta Data Checksum	X	X2	X	
uint16	CodeChk	Patch Code Checksum	X	X2	X	
uint8[CodeLen]	Code	Patch Code	X	X	X	
uint16	Checksum	Checksum (for everything to be present in NVS)	X	X	X	

### 5.6.3 Patch Command Flow

A patch needs to be pushed using the "Write ROM Patch" command. As indicated in [Section 5.6.2](#), the content of the file needs to be pushed in a specific way (X1 in one command, X2 in one command) to ensure the correct storage in NVS and Patch RAM.



**Figure 33. Patch Command Flow**

## 6 Profile Support

The LMX9838 is a full bluetooth node including the Generic Access profile (GAP), Service Discovery Profile (SDAP) and Serial Port Profile (SPP). In addition to those standard profiles, additional profiles can be implemented on host, using the LMX9838 as gateway to transport the higher layer protocols. A typical example for such a protocol is the AT-Commandset, which is used for Dial-up Networking (DUN) or also audio profiles like Headset (HSP) or Handsfree profiles (HFP).

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

## 6.1 MULTIPLE SERVICE ENTRIES

The LMX9838 includes a configurable database, which includes the services to be offered to remote devices. These services, which also include the dedicated RFCOMM port number, will be offered to a SDAP requests.

[Section 4.2](#) describes, how to establish a link using the LMX9838 command interface. This also includes the SDAP requests, which are necessary to get the available services and the dedicated RFCOMM port numbers.

### 6.1.1 The SDAP Service Database

By default the LMX9838 will offer one SPP profile to remote devices, which is dedicated to RFCOMM Port 1. The LMX9838 supports up to 10 database entries. See [Table 58](#) for the default service database configuration.

**Table 58. Default Service Database Configuration Index**

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

Once a remote device connects to the LMX9838, the firmware will first compare the RFCOMM port connected to the service database. In case no entry is found, the default settings as shown at index 0x00 in [Table 58](#) will be used. In case an appropriate entry for the addressed RFCOMM port exists, the LMX9838 will check the authentication and encryption settings and initiate the necessary procedures.

Independent of the stored profile, the LMX9838 will not do any additional profile specific procedures beyond the SPP link establishment including authentication and encryption. In case other profiles like DUN or OPP have been stored, the host has to monitor the RFCOMM port number which has been connected "Incoming Link Established Indicator" ([Section 7.2.3.7](#)) and has to react with the appropriate profile procedures.

### 6.1.2 Configuring the Service Database

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

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

- "Store Generic SDP Record" ([Section 7.2.13.1](#))

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

- "SDP Enable SDP Record"

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

- "SDP Delete All SDP Records"
- "Restore Factory Settings"

### 6.1.3 Example

As mentioned before the LMX9838 will only use the RFCOMM port and the Authentication and Encryption parameters of the stored profiles. In case profiles other than SPP shall be implemented the LMX9838 will only handle the SPP link establishment, the remaining profile specific actions like AT commands for DUN need to be implemented on the host.

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

The following sections describe an example for the configuration of the LMX9838 service database to offer 3 services and how the host will be notified on the addressed profile.

In this example, the LMX9838 shall be used as bluetooth node within a mobile phone, offering the following services.

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

### 6.1.3.1 Configuring the Service Database

The first step in enabling the phone for these services is to add the service record to the service database. For this the "Store Generic SDP Record" (Section 7.2.13.1) needs to be used.

For the setup the default SPP entry as set by "Restore Factory Settings" (Section 7.2.18.5) is assumed and will not be modified. Therefore only the DUN and the HSP profile entries need to be made. Table 59 shows the two commands to be sent to the LMX9838.

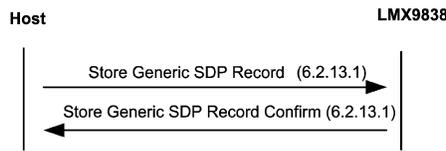


Figure 34. Configuring the Service Database

Table 59. Configuring the Service Database

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
	TX	Request	02,52,31,5E,00,E1,02,02,04,59,00,00,00,0A,00,00,00,00,01,00,35,03,19,03,11,04,00,35,0C,35,03,19,00,01,35,05,19,03,00,08,02,05,00,35,03,19,02,10,06,00,35,09,09,6E,65,09,6A,00,09,00,01,09,00,35,08,35,06,19,03,11,09,00,01,00,01,25,13,44,69,61,6C,2D,75,70,20,4E,65,74,77,6F,72,6B,69,6E,67,00,05,03,28,00,03	Tx: Cmd: Store SDP Record, Local Port: 02, Authentication: 02, Encryption: 04, SdpRecord: 00000A00000000010035031903110400350C3503190001350519030008020500350319021006003509096E65096A00090001090035083506190311090001000125134469616C2D7570204E6574776F726B696E67005032800
00 / 01	RX	Confirm	02,43,31,02,00,76,00,01,03	Rx: Event: Store SDP Record, Status: 00, Identifier: 01
	TX	Request	02,52,31,58,00,DB,03,22,44,53,00,00,00,0A,00,00,00,00,01,00,35,06,19,12,11,19,03,12,04,00,35,0C,35,03,19,00,01,35,05,19,03,00,08,03,05,00,35,03,19,02,10,06,00,35,09,09,6E,65,09,6A,00,09,00,01,09,00,35,08,35,06,19,08,11,09,00,01,00,01,25,0E,56,6F,69,63,65,20,47,61,74,65,77,61,79,00,03	Tx: Cmd: Store SDP Record, Local Port: 03, Authentication: 22, Encryption: 44, SdpRecord: 00000A00000000010035061912111903120400350C3503190001350519030008030500350319021006003509096E65096A000900010900350835061908110900010001250E566F69636520476174657761790003
00 / 01	RX	Confirm	02,43,31,02,00,76,00,02,03	Rx: Event: Store SDP Record, Status: 00, Identifier: 02

After these commands the service database will be filled as described in Table 60.

NOTE: This example shows each profile connected to a different RFCOMM port. This makes it easy for the host to differentiate between the different profiles. As the amount of RFCOMM ports can have influence on the overall performance of the LMX9838, it is also possible to connect one RFCOMM port to multiple profiles. In this case the application on the host needs to be able to differentiate between the different profiles on application level.

Table 60. Example Service Database Configuration

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

**Table 60. Example Service Database Configuration (continued)**

Index	Enabled	Local RFCOMM Port	Type	Name	Authentication	Encryption	Profile Specifics
02	1	03	HSAG	Voice Gateway	0x22	0x44	none

### 6.1.3.2 Configuring RFCOMM Ports to Open

The LMX9838 can handle up to seven RFCOMM ports simultaneously. Each RFCOMM port opened creates a buffer instance within the RAM to handle the upcoming data traffic. Because of that by default only RFCOMM port 1 is activated.

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

- [Section 7.2.12](#)

The ports in this command are expressed by a 32-bit mask indicating which RFCOMM ports the LMX9838 has to open. Bit 30 and 31 must be set to 0. Bit 0 is RFCOMM port 1 and bit 29 is port 30. The changes take effect as soon as the command has been confirmed.

Examples:

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

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

---

**NOTE:** The LMX9838 is optimized for a maximum of seven open RFCOMM ports. Any additional port can reduce the ability to establish ACL links.

---

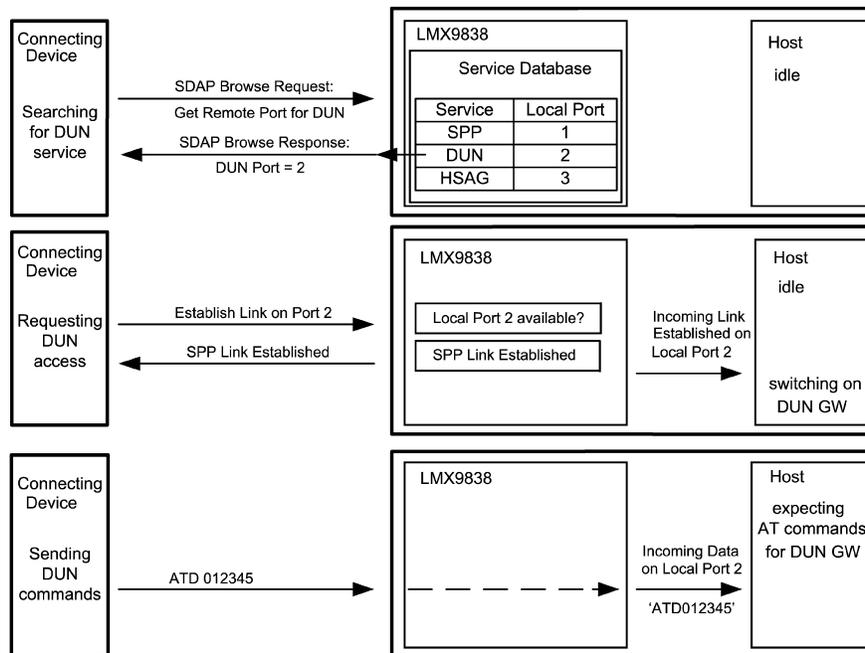
**Table 61. Open the Necessary RFCOMM Ports**

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

### 6.1.3.3 Handling Incoming Links

As described in [Section 6.1.3.1](#), for incoming links the LMX9838 will check the service database for configurations on authentication and encryption and, if appropriate, serve profile specific attributes. Once the link is established the LMX9838 will indicate the incoming link and RFCOMM port number to the host with the standard command "Incoming Link Established Indicator" ([Section 7.2.3.7](#)). Afterwards the host needs to decide according to the RFCOMM port number addressed, which service has been contacted and which profile specific procedure are necessary.

[Figure 35](#) shows a typical flow, in which the remote device selects the DUN profile out of a service database.


**Figure 35. Example Differentiate Different Profiles**

The link establishment will be handled as standard SPP link. As first action, the connecting device needs to ask for the RFCOMM port it has to use for the required service. So in this case, the connecting device will first create a SDAP connecting and send a service browse request including the UUID for the desired service. The LMX9838 will answer with RFCOMM port number stored for DUN in NVS, in this example port 2.

Afterwards the connecting device requests a standard SPP link to the port number extracted from the SDAP browse. In case the RFCOMM port is available the LMX9838 accepts the incoming link, considering the settings within the service database.

Finally the LMX9838 indicates the successful link establishment by sending the “Incoming Link Established Indicator”, to the host. The event includes also the port number which has been connected.

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

**Table 62. Host Notification for Incoming Link on RFCOMM Port 2**

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

**NOTE:** As shown in [Figure 35](#) the host will not be notified about the profile connected, it will only be notified about the RFCOMM port the remote port has established a link to. Because of this, the host needs to know the mapping between RFCOMM port and profile. This is necessary in case these profiles shall be available simultaneously.

Since more than one SPP based profile is based on AT commands, the differentiation between the profiles could also happen on application level, meaning all profiles use the same RFCOMM port. In this case, the host will not have to store the mapping between profile and RFCOMM port but would have to have a generic AT Commandset application, supporting all different profiles. The drawback of this is that all remaining profiles will be blocked as soon as one profile is in use.

## 6.2 SERIAL PORT PROFILE (SPP)

The SPP profile is the basic profile to be used for standard cable replacement. The SPP profile is fully included within the LMX9838. The SPP profile is basically covered by the SPP Link establishment command. Please see the examples in [Section 4.4](#).

## 6.3 DIAL-UP NETWORKING (DUN)

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

The following roles are defined for this profile:

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

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

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

The scenarios covered by this profile are the following:

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

The LMX9838 is capable of acting as both gateway or data terminal. In case of acting as DT, no further configurations are necessary. Acting as GW the LMX9838 needs to be configured to offer the DUN networking service to the DT, meaning an additional Service Record has to be stored within the device.

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

### 6.3.1 AT Commandset

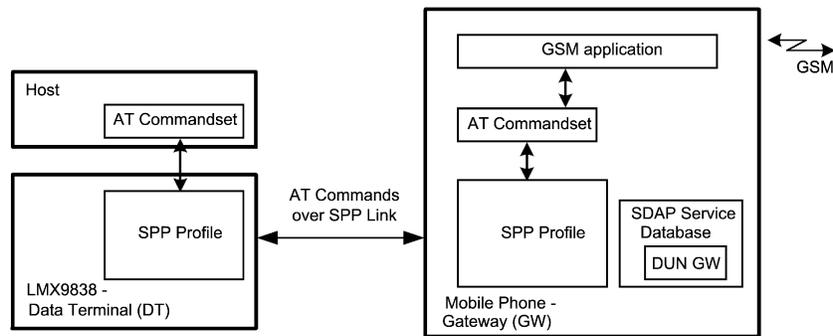
Dial up networking is based on a serial connection between two devices. The profile itself is handled by AT commands. So in order to create a DUN profile a standard SPP link has to be established, which then is used to transport the AT commands. For example using the LMX9838 to dial out with a mobile phone it first establishes a standard SPP link as done with SPP profile and then sending "ATDT0123456" as data to dial the specific number.

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

Please refer to to get the full of AT Commands to be supported by GW or DT devices. The commands are also listed in [Section 7.4](#).

### 6.3.2 Acting as DUN Data Terminal

Using the LMX9838 as data terminal (DT) means, it uses the DUN service of another device, e.g. a mobile phone, to dial out. The host needs to send the standard AT commands to establish the SPP link as described in [Section 4.2](#). There are no additional configurations necessary on the LMX9838.


**Figure 36. Software Layer Implementations for DUN**

The only difference for the link setup is that the DT has to connect to the DUN profile of the remote device instead of connecting to the SPP profile. To reach that, the correct remote RfComm port has to be used within the "Establish Link" command ([Section 7.2.3.1](#)). Once the correct RfComm port has been used, the remote device knows automatically that DUN has been addressed. The following chapters give a step by step description of connecting to the DUN service of a mobile phone.

### 6.3.2.1 Searching for the Device

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

### 6.3.2.2 Getting the RFCOMM Port for DUN on the Remote Device

As described in [Section 4.2.3](#), the remote RfComm port number for the service can be found out by using the Service Browse command. The command includes the parameter UUID, which defines the service to be looked for. The UUID for Dial Up Networking profile is 0x1103. See [Figure 37](#) for the command flow, [Table 63](#) shows the detailed parameters to be used for the commands.

**Table 63. Get Remote RfComm Port for DUN Profile of the Remote Device**

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

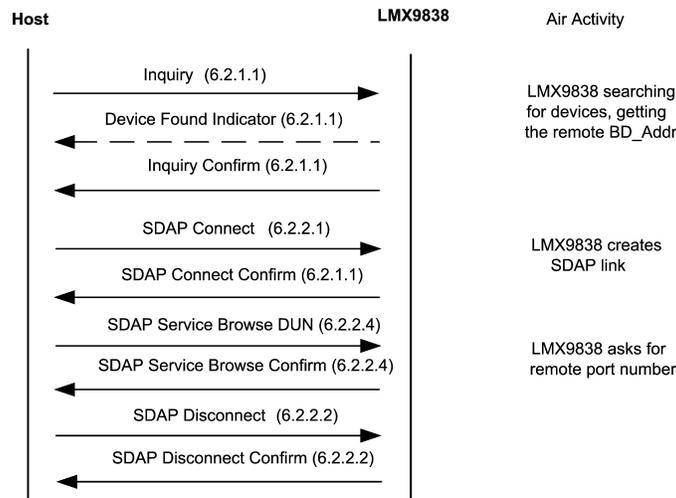


Figure 37. Command Flow: Requesting the RFCOMM Port Number from the Remote Device

### 6.3.2.3 Establishing a Link to a Dial Up Networking Service

The link establishment to a DUN profile is the same as to create a link to a standard SPP port. The remote device will recognize the profile to be used by the RFCOMM port, to which the SPP link has been created to. For this example, out of the procedure explained in Section 6.3.2.2, the RFCOMM port to be addressed for DUN is 0x03.

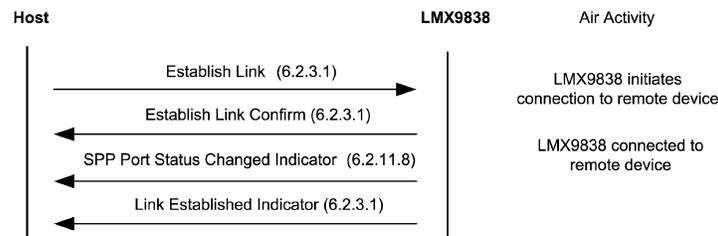


Figure 38. Command Flow: Establish a Link to a DUN Service of the Remote Device

Table 64. Establish Link to DUN Service on Remote Device

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
	TX	Request	02,52,0A,08,00,64,01,12,34,56,78,9A,BC,03,03	Tx: Cmd: Establish Link, Local Port: <b>01</b> , BdAddr: <b>123456789ABC</b> , Remote Port Number: <b>03</b>
00 / 01	RX	Confirm	02,43,0A,02,00,4F,00, <b>01</b> ,03	Rx: Event: Establish Link, Status: 00, Local Port: <b>01</b>
0	RX	Indicator	02,69,50,07,00,C0,12,34,56,78,9A,BC, <b>00</b> ,03	Rx: Event: ACL Established, BdAddr: 123456789ABC, Status: <b>00</b>
00 / 01	RX	Indicator	02,69,3E,04,00,AB,01,0C,00,00,03	Rx: Event: Port Status Changed, Local Port: 01, PortStatus: 0C, Break Length: 0000
00 / 01	RX	Indicator	02,69,0B,09,00,7D,00,12,34,56,78,9A,BC,0 <b>1</b> , <b>03</b> ,03	Rx: Event: Link Established, Status: 00, BdAddr: 123456789ABC, Local Port: 01, Remote Port Number: <b>03</b>

### 6.3.2.4 Sending AT Commands

Once the SPP link is established the GW will activate its AT commandset, expecting and accepting a specific AT commands as described in]. A full list of commands to be supported by the GW can also be found in Section 7.4.1.

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

So the commands from DT to GW will be thus:

AT<cmd>=<value><cr>

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

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

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

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

The AT commands can be sent by either using the "SPP Send Data" (Section 7.2.3.3) command or by first switching to transparent mode and then send the commands directly.

### 6.3.2.4.1 Using the Command Interface to Send and Receive AT Commands

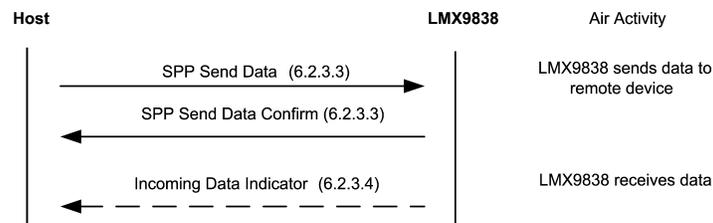


Figure 39. Flow Command: Sending AT Commands

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

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
	TX	Request	02,52,0F,07,00,68,01,04,00,41,54,0D,0A,03	Tx: Cmd: Send Data, Local Port: 01, Payload Data: 41540D0A (AT)
00 / 01	RX	Confirm	02,43,0F,02,00,54,00,01,03	Rx: Event: Send Data, Status: 00, Local Port: 01
00 / 01	RX	Indicator	02,69,10,09,00,82,01,06,00,0D,0A,4F,4B,0D,0A,03	Rx: Event: Incoming Data, Local Port: 01, Received Data: 0D0A4F4B0D0A (OK)
	TX	Request	02,52,0F,13,00,74,01,10,00,41,54,44,54,30,31,32,33,34,35,36,37,38,39,30,0D,03	Tx: Cmd: Send Data, Local Port: 01, Payload Data: 4154445430313233343536373839300D (ATDT01234567890)
00 / 01	RX	Confirm	02,43,0F,02,00,54,00,01,03	Rx: Event: Send Data, Status: 00, Local Port: 01

### 6.3.2.4.2 Using Transparent Mode to Send and Receive AT Commands

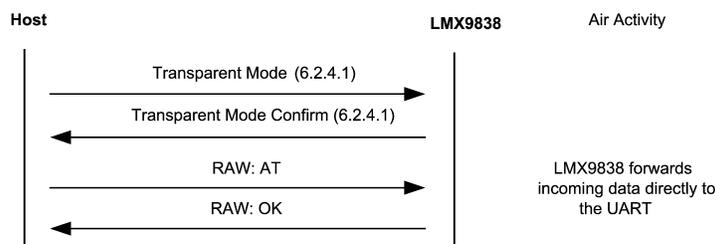


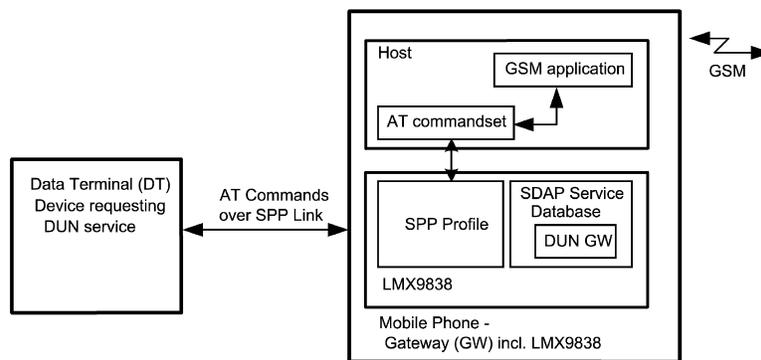
Figure 40. Sending AT Commands

**Table 66. Send Init Command “AT” and Dialing Command “ATDT01234567890”**

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

### 6.3.3 Acting as DUN Gateway

A DUN gateway offers the ability to access another network, e.g. GSM, routing the data between the data from the bluetooth link to the other network. The LMX9838 can be configured to offer a DUN networking service to other devices, e.g. as in a mobile phone or LAN access point. The host is able to differentiate the requested DUN service by checking the local port, to which has been connected to. Please see [Section 6.1.3.3](#) for details how to differentiate between different profiles.



**Figure 41. Using LMX9838 in a DUN GW Application**

#### 6.3.3.1 Configuring the Service Database

In order to offer an additional DUN service to other devices a new entry has been made into the service database. For this the "Store Generic SDP Record" ([Section 7.2.13.1](#)) can be used. The following example shows how to add the DUN entry to the default SDB database including only one SPP entry.

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

1. Select “ Tools/Create Service Record” within the Simply Blue Commander menu.
2. Select the service record required, in this case “Dial-up Networking (DUN)”
3. Select the settings desired for the DUN profile and choose the correct RFCOMM port. As this example device shall still be able to accept a second standard data links as well, RFCOMM port 2 should be used. Finally confirm the dialog with pressing “Create”.
4. The “Create” will fill the Hex/ASCII line of the Simply Blue Commander with the string necessary to send to the LMX9838. Afterwards just close the dialog with the “Close” Button.
5. The string can be sent directly to the board or stored as command within the directory. Once sent it can also be copied out of the log entry and copied into the development code when switching of the interpretation option.

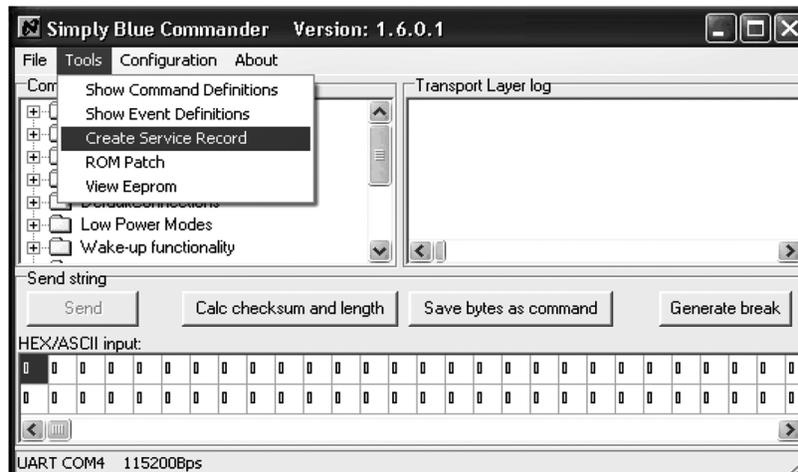


Figure 42. Opening “Create Service Record” Dialog-box

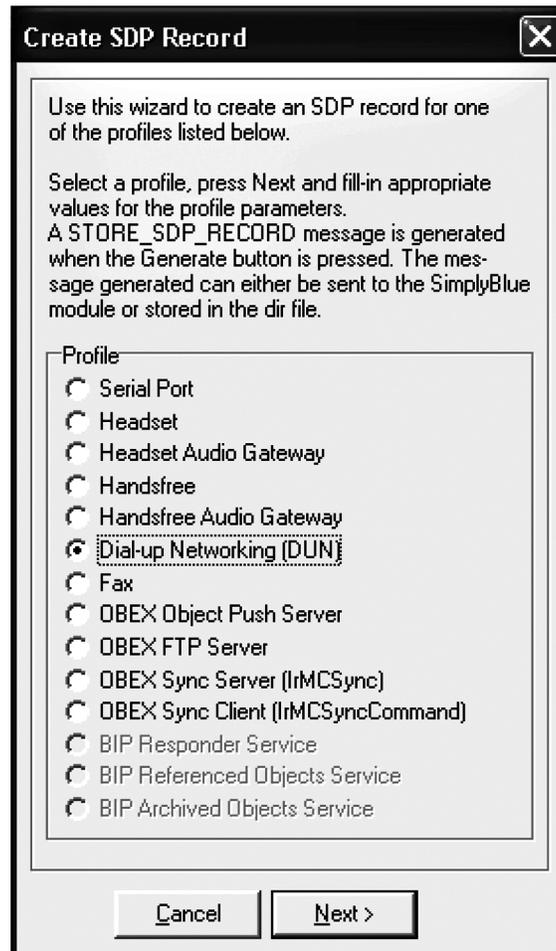


Figure 43. Selecting the Profile



Figure 44. Configuring the Profile Settings for DUN

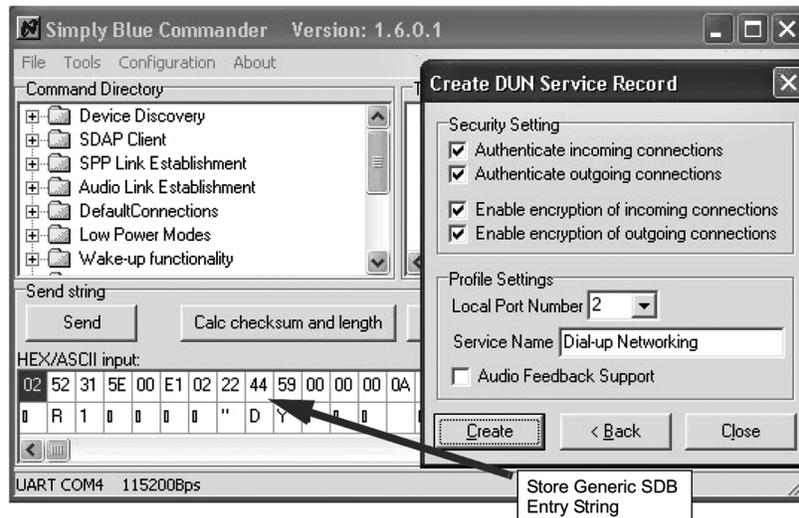


Figure 45. Creating the Command String for the Audio Gateway Entry

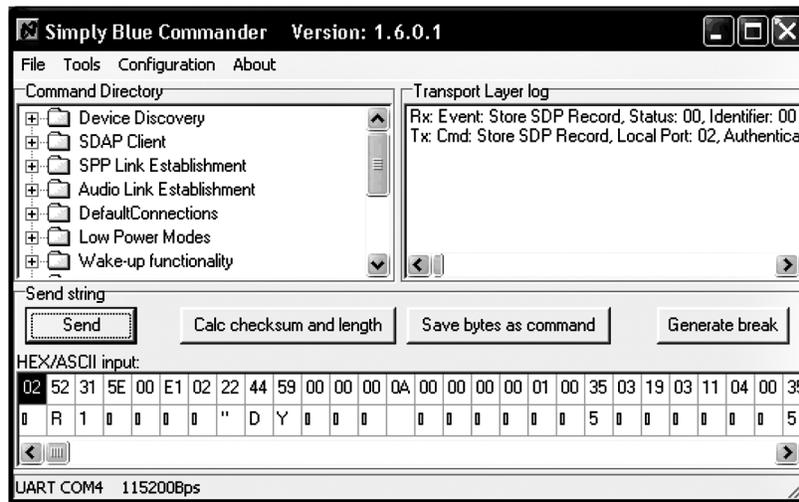


Figure 46. Send the Command to the LMX9838

Table 67. Adding the DUN Service to the Service Database

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
	TX	Request	02,52,31,5E,00,E1,02,02,04,59,00,00,00,0A,00,00,00,00,01,00,35,03,19,03,11,04,00,35,0C,35,03,19,00,01,35,05,19,03,00,08,02,05,00,35,03,19,02,10,06,00,35,09,09,6E,65,09,6A,00,09,00,01,09,00,35,08,35,06,19,03,11,09,00,01,00,01,25,13,44,69,61,6C,2D,75,70,20,4E,65,74,77,6F,72,6B,69,6E,67,00,05,03,28,00,03	Tx: Cmd: Store SDP Record, Local Port: 02, Authentication: 02, Encryption: 04, SdpRecord: 00000A000000000010035031903110400350C3503190001350519030008020500350319021006003509096E65096A00090001090035083506190311090001000125134469616C2D7570204E6574776F726B696E670005032800
00 / 01	RX	Confirm	02,43,31,02,00,76,00,01,03	Rx: Event: Store SDP Record, Status: 00, Identifier: 01

After this command the Service database has the following content:

Table 68. Service Database for COM1 and DUN

Index	Enabled	Local RFCOMM Port	Type	Name	Authentication	Encryption	Profile Specifics
0	1	1	SPP	COM1	0x02	0x04	none
1	1	2	DUN	DUN	0x02	0x04	none

### 6.3.3.2 Adapt RFCOMM Ports to Open

As both services shall be connectable independently, two separate RFCOMM ports need to be used. For this the "Set Ports To Open" command (Section 7.2.12.1) should be used.

Table 69. Set RFCOMM Ports to Open

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

### 6.3.3.3 Routing AT Commands

The DUN GW profile is based on the standard AT commandset as listed in [Section 7.4.1](#).

Once an incoming link has been established to the appropriate RFCOMM port, the host needs to be able to react to these AT commands. The LMX9838 will only route the AT commands from the host to the remote device.

Please refer to [Section 6.1.3.3](#) for how to manage incoming connections. The AT commands need to be sent and recognized as described in [Section 6.3.2.4](#).

## 6.4 HEADSET PROFILE

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

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

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

The following roles are defined for this profile:

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

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

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

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

The following sections describe, how to use the LMX9838 as HSAG or HS.

### 6.4.1 AT Commandset and Results

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

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

```
AT<cmd>=<value><cr>
```

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

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

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

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

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

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

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

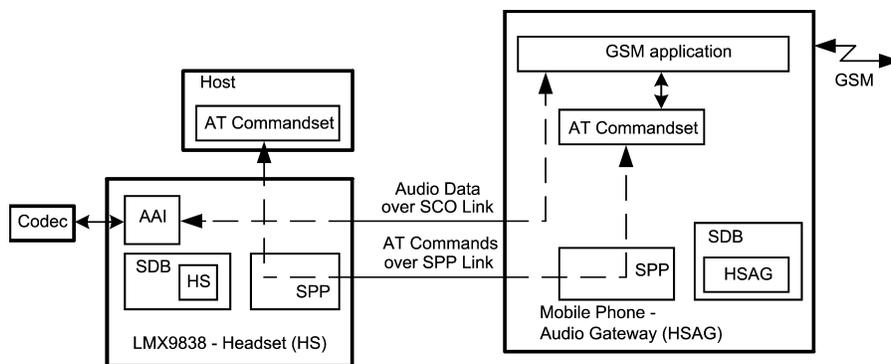
Please see [Section 7.4.2.1](#) for the complete list of commands necessary for the headset profile.

### 6.4.2 Acting as Headset

The headset is defined as the device, providing the input and output for audio data, e.g. microphone and speaker. The LMX9838 is able to connect an external codec via the on-chip advanced audio interface (AAI). Please see [Figure 6](#) how to connect the codec to the AAI.

The HS side of the headset profile is mostly controlled by Audio gateway. While the SPP link establishment can be initiated from both sides, the SCO link establishment will always be initiated by the HSAG. The LMX9838 will automatically accept incoming SCO links and report the successful establishment to the host.

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



**Figure 47. Headset Data and Audio stream**

#### 6.4.2.1 Configuring the Audio Path

In general the LMX9838 can be seen as gateway between the UART interface and the bluetooth link. The audio link is one additional interface for the LMX9838 to be routed. Please see also [Figure 24](#) for the two routing options.

The LMX9838 supports different configurations for the PCM codec interface and the air interface. The bluetooth specification defines the following formats to be used over the bluetooth link:

- CVSD (default)
- $\mu$ -Law
- A-Law

As indicated, the default and mostly used coding format used over a bluetooth link is CVSD. Since the headset most likely will always use the same settings, it is recommended to set the parameters to a default in the NVS, using the "Set audio settings" command ([Section 7.2.6.2.5](#)).

**Table 70. Set Default Audio Settings to OKI Codec and CVSD**

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

### 6.4.2.2 Configuring the Service Database for Headset

Since a headset usually is only used for exactly that purpose, the "SDP Delete All SDP Records" command ([Section 7.2.13.3](#)) could be used first, to delete the default SPP service database entry "COM1". The table below shows the command to be used to delete the complete database.

**Table 71. Delete SDB Database**

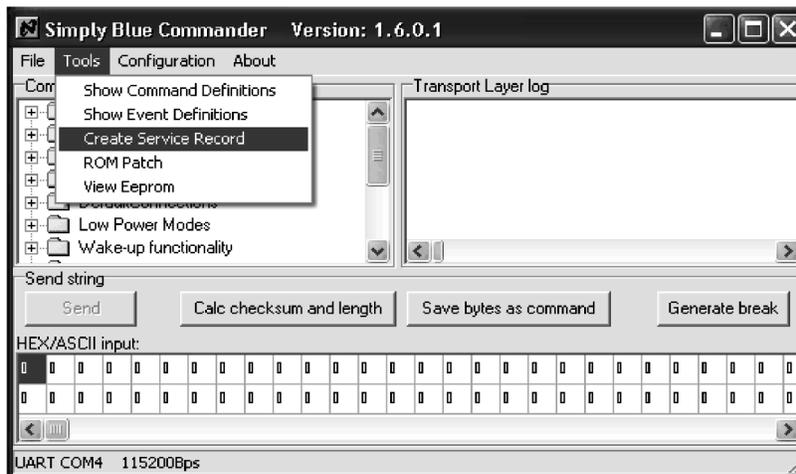
Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
	TX	Request	02,52,2A,00,00,7C,03	Tx: Cmd: Delete All SDP Records
00 / 01	RX	Confirm	02,43,2A,01,00,6E,00,03	Rx: Event: Delete All SDP Records, Status: 00

In order to offer an Headset service to other devices a new entry has been made into the service database. For this the "Store Generic SDP Record" ([Section 7.2.13.1](#)) needs to be used.

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

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

1. Select "Tools/Create Service Record" within the Simply Blue Commander menu.
2. Select the service record required, in this case "Headset"
3. Select the settings desired for the Headset profile and choose the correct RFCOMM port. Since the database has been deleted before, RFCOMM port 1 can be used. Finally confirm the dialog with pressing "Create".
4. The "Create" will fill the Hex/ASCII line of the Simply Blue Commander with the string necessary to send to the LMX9838. Afterwards just close the dialog with the "Close" Button.
5. The string can be sent directly to the board or stored as command within the directory. Once sent it can also be copied out of the log entry and copied into the development code when switching of the interpretation option.



**Figure 48. Opening "Create Service Record" Dialog-box**



**Figure 49. Selecting the Profile**



**Figure 50. Configuring the Profile Settings for Headset**

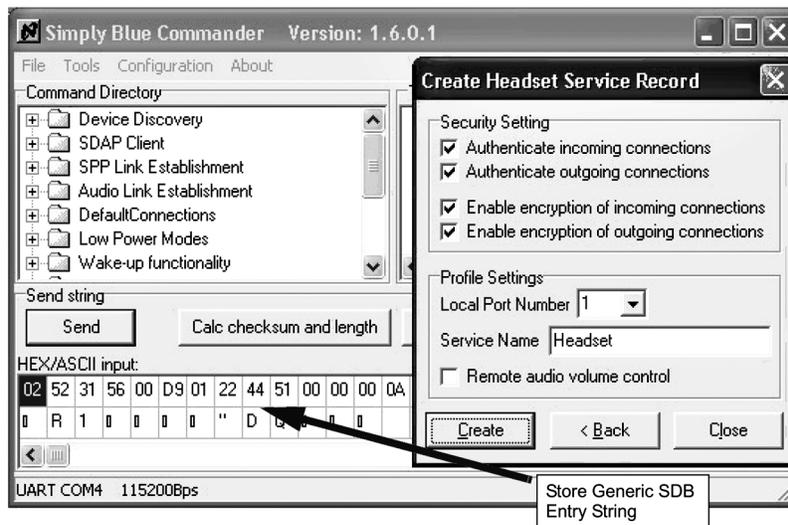


Figure 51. Creating the Command String for the Headset Entry

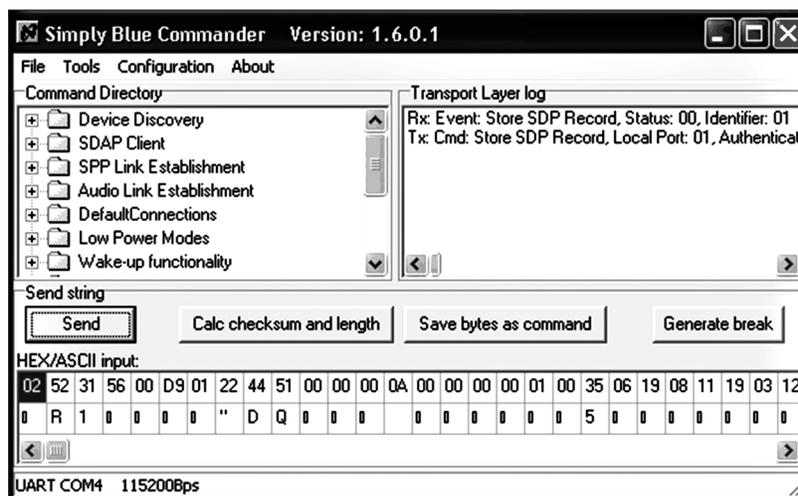


Figure 52. Send the Command to the LMX9838

Table 72 shows the command string being the result out of the steps just described.

Table 72. Adding the Headset Service to the Service Database

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
	TX	Request	02,52,31,56,00,D9,01,22,44,51,00,00,00,0A,00,00,00,00,01,00,35,06,19,08,11,19,03,12,04,00,35,0C,35,03,19,00,01,35,05,19,03,00,08,01,05,00,35,03,19,02,10,06,00,35,09,09,6E,65,09,6A,00,09,00,01,09,00,35,08,35,06,19,08,11,09,00,01,00,01,25,08,48,65,61,64,73,65,74,00,02,03,28,00,03	Tx: Cmd: Store SDP Record, Local Port: <b>01</b> , Authentication: <b>22</b> , Encryption: <b>44</b> , SdpRecord: 00000A00000000010035061908111903120400350C3503190001350519030008010500350319021006003509096E65096A000900010900350835061908110900010001250848656164736574000203280003
00 / 01	RX	Confirm	02,43,31,02,00,76,00,00,03	Rx: Event: Store SDP Record, Status: 00, Identifier: <b>00</b>

After this command the Service database has the following content:

**Table 73. Service Database for COM1 and Headset**

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

#### 6.4.2.3 Adapt RFCOMM Ports to Open

To only serve this one service only RFCOMM port one needs to be opened. In order to make only RFCOMM port 1 available for connection, the "Set Ports To Open" command ([Section 7.2.12.1](#)) should be used to make the appropriate configuration.

**Table 74. Set RFCOMM Ports to Open**

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

#### 6.4.2.4 Store Class of Device

During the Inquiry procedure the searching device will already receive the class of device. Please see also [Section 5.1.2.3](#) for more information on that parameter. In order to be recognized it is beneficial to already indicate the audio capability within the class of device. The following table shows the command how to set the "Class of Device" using the "Store Class of Device" command ([Section 7.2.14.5](#)). The class is set to 0x220404, which reflects the class for a headset.

**Table 75. Store Class of Device**

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

#### 6.4.2.5 Connection Handling as Headset

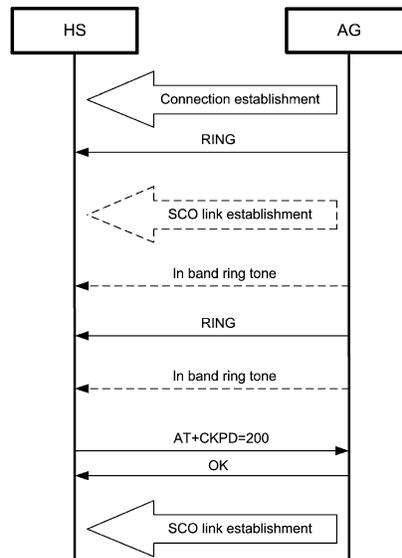
The Headset profile is based on some specific AT commands also listed in [Section 7.4.2](#).

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

The LMX9838 accepts incoming SCO links or SPP links automatically. It just indicates the successful establishment to the host. Therefore, for the HS implementation no further LMX9838 specific commands besides the already described configuration are required. The headset profile itself is controlled by a few AT commands. The following chapters give the generic guideline of how to act as headset within the different scenarios.

##### 6.4.2.5.1 Incoming Audio Connection

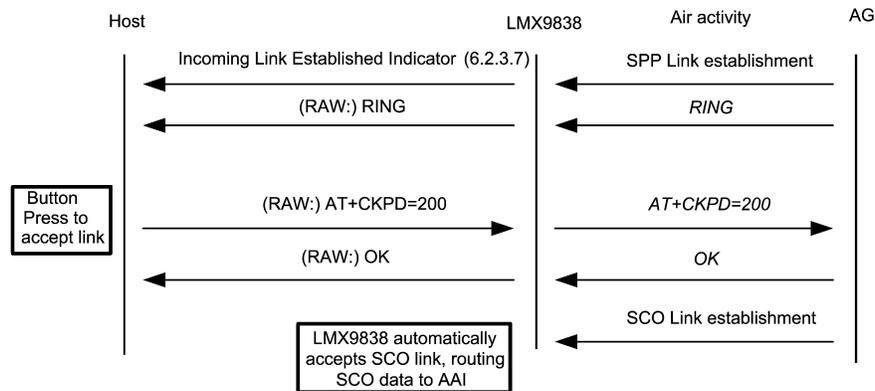
[Figure 53](#) shows the flow diagram of a headset link initiated by the HSAG as defined in .



**Figure 53. “Incoming Audio Connection Establishment” as Defined in Bluetooth Specification**

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

Please see [Figure 54](#) which shows the headset part of the profile for the incoming audio connection ‘translated’ to the LMX9838 interface.



**Figure 54. Incoming Audio Connection with LMX9838 as HS**

[Figure 54](#) shows the flow based on automatic mode, in which the LMX9838 will automatically switch to transparent mode on the UART. Please see the following table for the detailed description of the UART traffic between host and LMX9838.

**Table 76. Incoming Audio Connection with LMX9838 as HS**

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

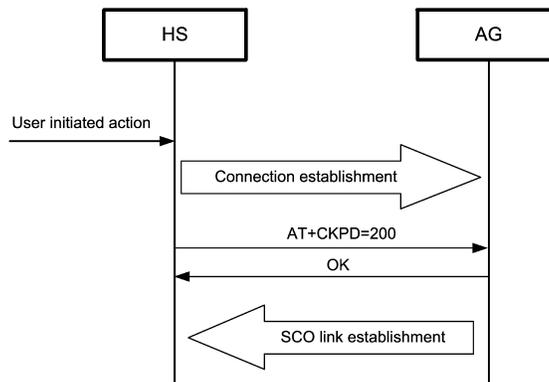
**Table 76. Incoming Audio Connection with LMX9838 as HS (continued)**

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
00 / 01 / 02 / 03	RX	RAW Data	0D,0A,4F,4B,0D,0A	Rx(RAW): 0D,0A,4F,4B,0D,0A (OK)

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

**6.4.2.5.2 Headset Initiated Connection Establishment**

The Headset profile also defines the scenario in which the headset initiates the link the HSAG. See [Figure 55](#) for the profile specification of the Bluetooth SIG.



**Figure 55. Outgoing Audio Connection Establishment as Defined by the Bluetooth SIG**

As [Figure 55](#) indicates, the headset basically only requests a SPP link and sends the AT+CKPD command. The rest is controlled by the HSAG. Therefore the link establishment is limited to the SPP link establishment and AT commands.

In order to establish a link to the HSAG, the headset needs to know the RFCOMM port number it has to connect on the remote device. Usually, these information are requested on first time of connection only, so only the "Establish Link" command ([Section 7.2.3.1](#)) will be necessary for the establishment.

[Table 77](#) gives an indication of the necessary SDAP request. The most important parameter is the UUID for the Headset Audio Gateway, which is 0x1112.

[Figure 56](#) and [Table 78](#) show the final profile flow necessary to establish a connection to the HSAG with the LMX9838.

**Table 77. Requesting the RFCOMM Port Number for HSAG from the Remote Device**

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

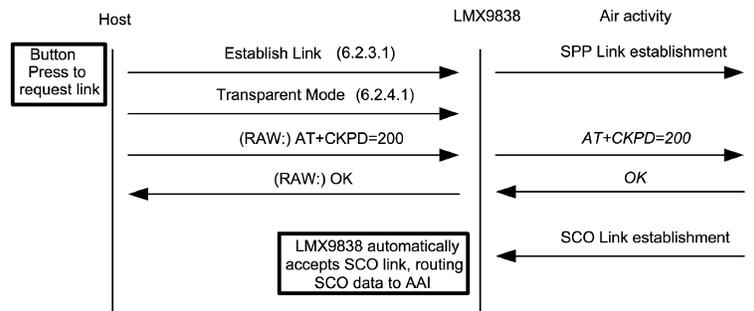


Figure 56. Outgoing Link Establishment from the LMX9838 as HS

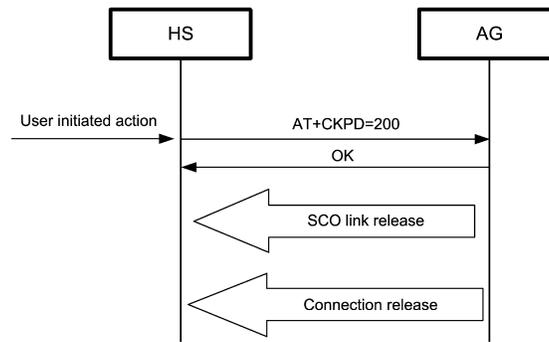
Table 78. Outgoing Link Establishment from the LMX9838 as HS

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

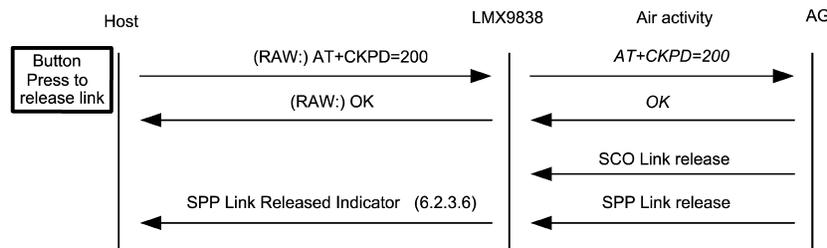
### 6.4.2.5.3 Audio Connection Release from the HS

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

The LMX9838 confirms the released link by the standard "SPP Link Released Indicator" (Section 7.2.3.6). See Figure 57 following the Bluetooth specification description for the realization with LMX9838. In case the HSAG releases the link the LMX9838 will send the same event .



**Figure 57. Audio Link Release by the HS**



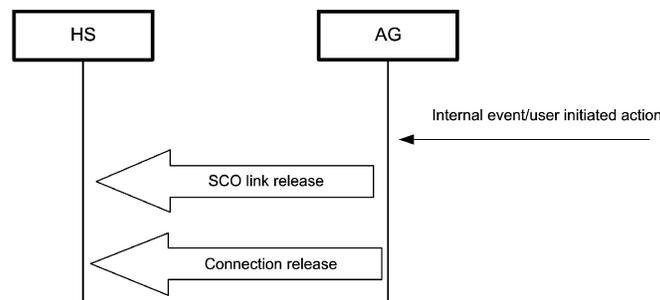
**Figure 58. Audio Link Release from the LMX9838 as HS**

**Table 79. Audio Link Release from the LMX9838 as HS**

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

**6.4.2.5.4 Audio Connection Release from the HSAG**

In case the HSAG releases the audio link, no further AT commands are required. The HSAG just releases the SPP and SCO link. The host will be notified by the standard "SPP Link Released Indicator" (Section 7.2.3.6) and "SCO Link Released Indicator" (Section 7.2.6.2.4).



**Figure 59. Audio Link Release by the HSAG**

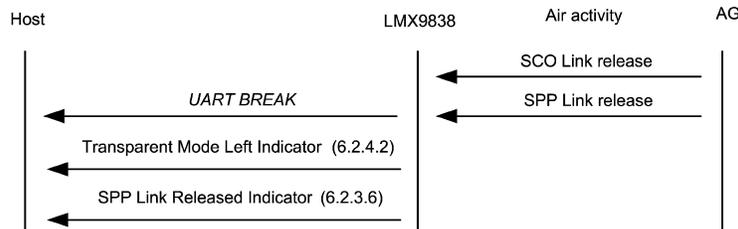


Figure 60. Audio Link Release from the HSAG

Table 80. Audio Link Release from the LMX9838 as HS

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

### 6.4.3 Acting as Audio Gateway

The audio gateway is defined as a device, handling audio data and using the HS as analog front end for both for input and output. Typical devices acting as Audio Gateways are cellular phones and personal computer. The LMX9838 can be used in an audio gateway by creating the required data link to the headset and routing all data from the Advanced Audio Interface (AAI) to the bluetooth link. The audio interface needs to be connected to the host PCM interface. Timing and electrical specification for the interface are derived out of the driver configuration set in NVS. Bit-clock and frame-clock settings for different codec options are described in [ "Texas Instruments: LMX9838 Datasheet". Please see [Figure 61](#) how to implement the LMX9838 in a audio gateway application.

The Audio Gateway (AG) controls the inter-operation with the headset in terms of establishing and releasing the required SCO link. While the SPP link establishment can be initiated from both sides, the SCO link establishment will always be initiated by the HSAG. The LMX9838 offers dedicated commands to control the audio link.

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

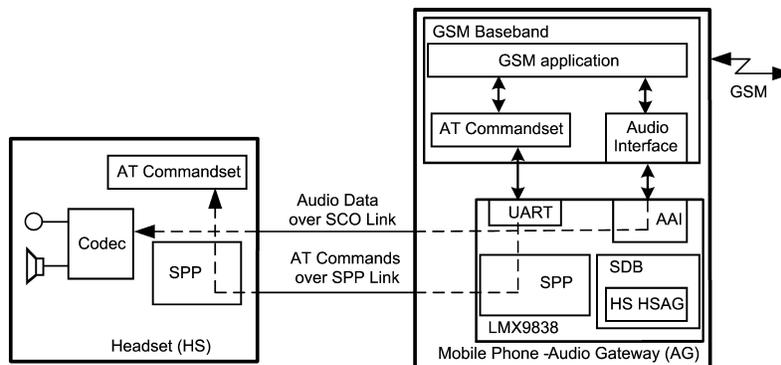


Figure 61. Audio Gateway Data and Audio Stream using LMX9838 in the HSAG

### 6.4.3.1 Configuring the Audio Path

In general the LMX9838 can be seen as gateway between the UART interface and the bluetooth link. The audio link is one additional interface for the LMX9838 to be routed. Please see also [Figure 24](#) for the two routing options.

The LMX9838 supports different configurations for the PCM codec interface and the air interface. The bluetooth specification defines the following formats to be used over the bluetooth link:

- CVSD (default)
- $\mu$ -Law
- A-Law

As indicated, the default and mostly used coding format used over a bluetooth link is CVSD. Since the headset most likely will always use the same settings, it is recommended to set the parameters to a default in the NVS, using the "Set audio settings" command ([Section 7.2.6.2.5](#)). Bitclock and frame-clock information for the different codecs are listed in .

**Table 81. Set Default Audio Settings to Motorola Codec and CVSD**

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

### 6.4.3.2 Configuring the Service Database for Audio Gateway

In order to offer an additional HSAG service to other devices a new entry has been made into the service database. For this the "Store Generic SDP Record" ([Section 7.2.13.1](#)) needs to be used. This command is a generic command to generate any possible profile within the device. To be able to do so, significant knowledge of the internal SDB structure would be required. In order to use this command, the "Simply Blue Commander" software needs to be used, to create this command.

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

1. Select "Tools/Create Service Record" within the Simply Blue Commander menu.
2. Select the service record required, in this case "Headset Audio Gateway"
3. Select the settings desired for the HSAG profile and choose the correct RFCOMM port. As this example device shall still be able to accept a second standard data links as well, RFCOMM port 2 should be used. Finally confirm the dialog with pressing "Create".
4. The "Create" will fill the Hex/ASCII line of the Simply Blue Commander with the string necessary to send to the LMX9838. Afterwards just close the dialog with the "Close" Button.
5. The string can be sent directly to the board or stored as command within the directory. Once sent it can also be copied out of the log entry and copied into the development code when switching of the interpretation option.

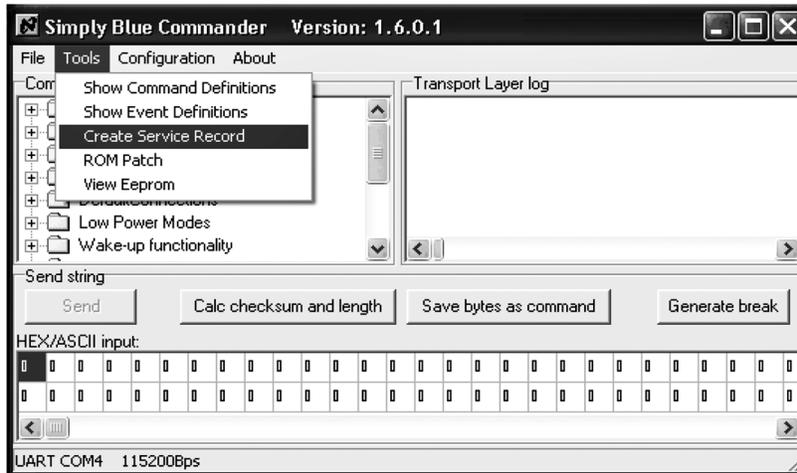


Figure 62. Opening “Create Service Record” Dialog-box

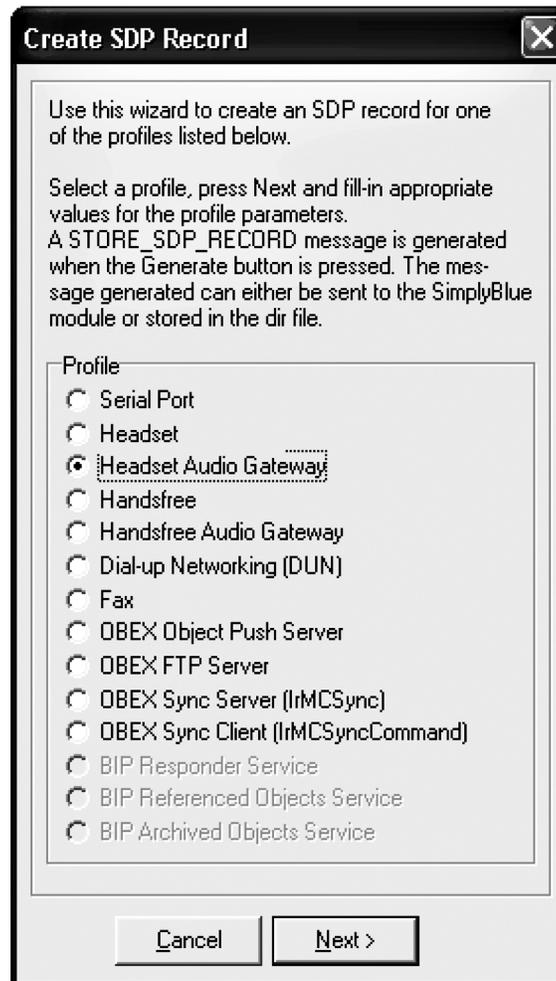


Figure 63. Selecting the Profile



Figure 64. Configuring the Profile Settings for HSAG

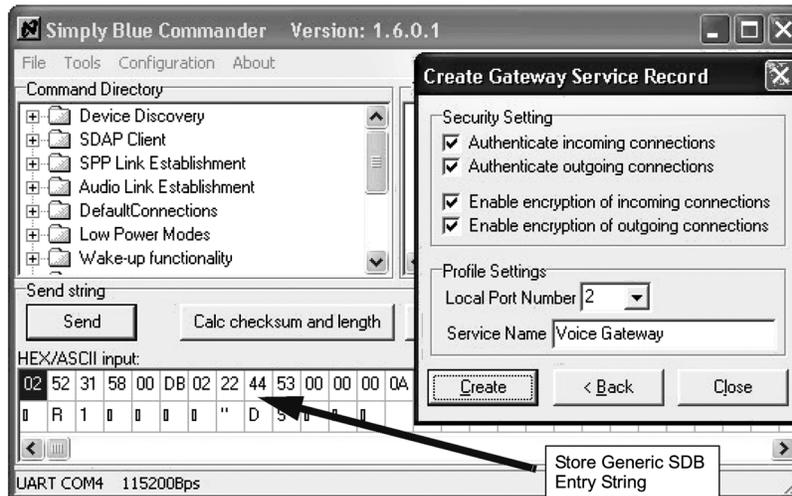


Figure 65. Creating the Command String for the Audio Gateway Entry

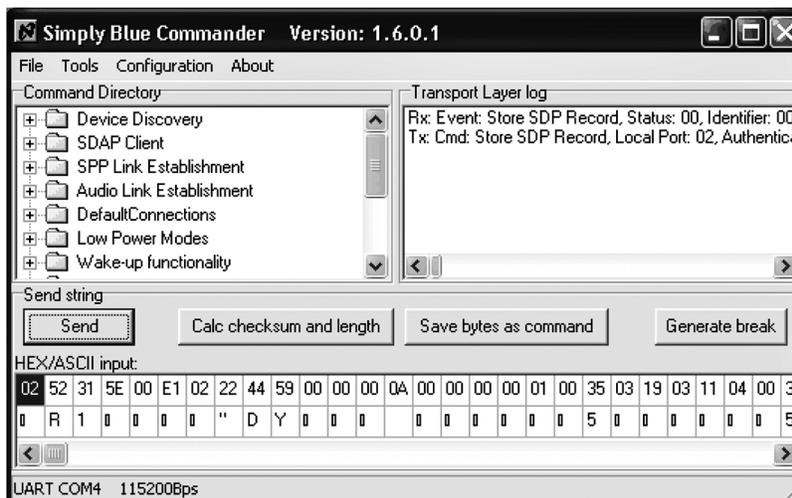


Figure 66. Send the Command to the LMX9838



The LMX9838 offers specific commands to establish and release SPP and SCO links. The first connection establishment is based on the standard SPP link. Once this is established the HSAG indicates the incoming call by sending the RING At command over the SPP link. Once the HS responds with the appropriate AT command, the HSAG needs to establish the SCO connection. In some implementations it is also possible to establish the SCO right after the SPP link to support inband ringtones.

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

### 6.4.3.5.1 Requesting the RFCOMM Port from the Headset

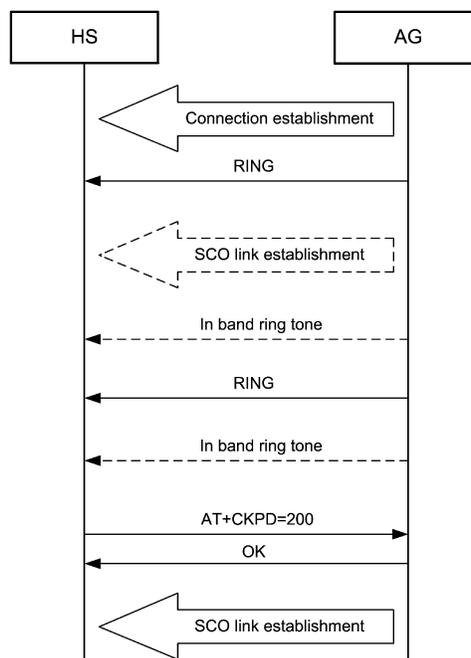
Table 86 gives an indication of the necessary SDAP request. The most important parameter is the UUID for the Headset Audio Gateway, which is 0x1108.

**Table 86. Requesting the RFCOMM Port Number for HSAG from the Remote Device**

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
	TX	Request	02,52,32,06,00,8A,12,34,56,78,9A,BC,03	Tx: Cmd: SDAP Connect, BdAddr: 123456789ABC
00	RX	Indicator	02,69,50,07,00,C0,12,34,56,78,9A,BC,00,03	Rx: Event: ACL Established, BdAddr: 123456789ABC, Status: 00
00 / 01	RX	Confirm	02,43,32,01,00,76,00,03	Rx: Event: SDAP Connect, Status: 00
	TX	Request	02,52,35,02,00,89,12,11,03	Tx: Cmd: Service Browse, Browse Group ID: 0811
00 / 01	RX	Confirm	02,43,35,16,00,8E,00,01,02,10,08,11,07,0E,56,6F,69,63,65,20,67,61,74,65,77,61,79,00,03	Rx: Event: Service Browse, Status: 00, Browse Group ID: 0210, Service ID: <b>0811</b> , PortNo: <b>07</b> , Service Name: Voice gateway.
	TX	Request	02,52,33,00,00,85,03	Tx: Cmd: SDAP Disconnect
00 / 01	RX	Confirm	02,43,33,01,00,77,00,03	Rx: Event: SDAP Disconnect, Status: 00
00	RX	Indicator	02,69,51,07,00,C1,12,34,56,78,9A,BC,13,03	Rx: Event: ACL Terminated, BdAddr: 123456789ABC, Reason: 13

### 6.4.3.5.2 Outgoing Audio Connection

Figure 67 shows the flow diagram of a HSAG link initiated by the HSAG as defined in .



**Figure 67. “AG initiated Audio Connection Establishment” as Defined in .**

As Figure 67 shows, the link is initiated by the HSAG, e.g. in case of an incoming call on a mobile phone. The HSAG will send the “RING” command to force an audio or visual signalling on the headset. The HSAG keeps on ringing until the HS user accepts the incoming link by pressing a button. On this the HS application will send the “AT+CKPD=200” command. The HSAG needs to confirm the successful reception of the command with the “OK” and finally sends the SCO establishment command.

Please see Figure 68 which shows the HSAG part of the profile for the outgoing audio connection ‘translated’ to the LMX9838 command interface.

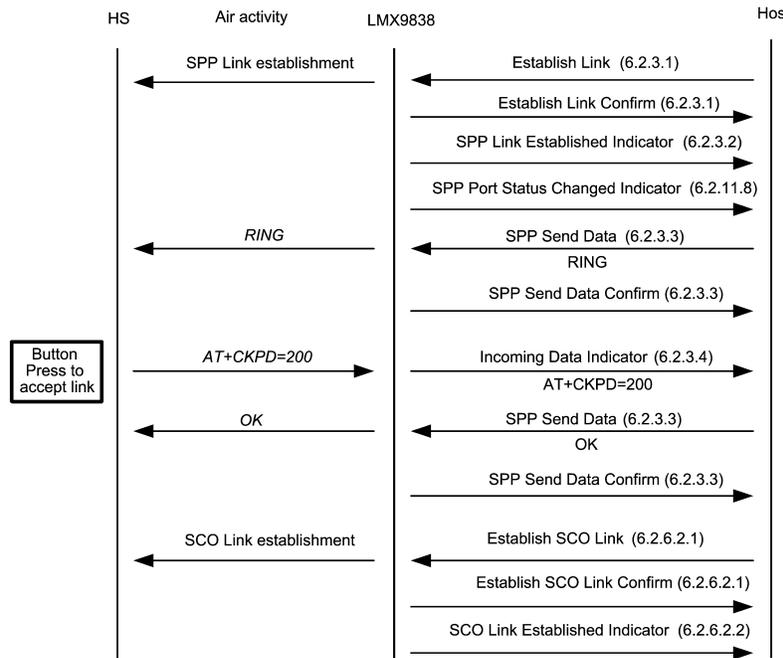


Figure 68. Outgoing Audio Connection with LMX9838 as HSAG

Please see the following table for the detailed description of the UART traffic between host and LMX9838.

Table 87. Outgoing Audio Connection with LMX9838 as HSAG

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
	TX	Request	02,52,0A,08,00,64,02,12,34,56,78,9A.BC,07,03	Tx: Cmd: Establish Link, Local Port: 02, BdAddr: 123456789ABC, Remote Port Number: 07
00 / 01	RX	Confirm	02,43,0A,02,00,4F,00,02,03	Rx: Event: Establish Link, Status: 00, Local Port: 02
00	RX	Indicator	02,69,50,07,00,C0,12,34,56,78,9A.BC,00,03	Rx: Event: ACL Established, BdAddr: 123456789ABC, Status: 00
00 / 01	RX	Indicator	02,69,0B,09,00,7D,00,12,34,56,78,9A.BC,02,07,03	Rx: Event: Link Established, Status: 00, BdAddr: 123456789ABC, Local Port: 02, Remote Port Number: 07
00 / 01	RX	Indicator	02,69,3E,04,00,AB,02,8C,00,00,03	Rx: Event: Port Status Changed, Local Port: 02, PortStatus: 8C, Break Length: 0000
	TX	Request	02,52,0F,0B,00,6C,02,08,00,0D,0A,52,49,4E,47,0D,0A,03	Tx: Cmd: Send Data, Local Port: 02, Payload Data: 0D0A52494E470D0A (RING)
00 / 01	RX	Confirm	02,43,0F,02,00,54,00,02,03	Rx: Event: Send Data, Status: 00, Local Port: 02
00 / 01	RX	Indicator	02,69,10,0F,00,88,02,0C,00,41,54,2B,43,4B,50,44,3D,32,30,30,0D,03	Rx: Event: Incoming Data, Local Port: 02, Received Data: 41542B434B50443D3230300D (AT+CKPD=200)
	TX	Request	02,52,0F,09,00,6A,02,06,00,0D,0A,4F,4B,0D,0A,03	Tx: Cmd: Send Data, Local Port: 02, Payload Data: 0D0A4F4B0D0A (OK)

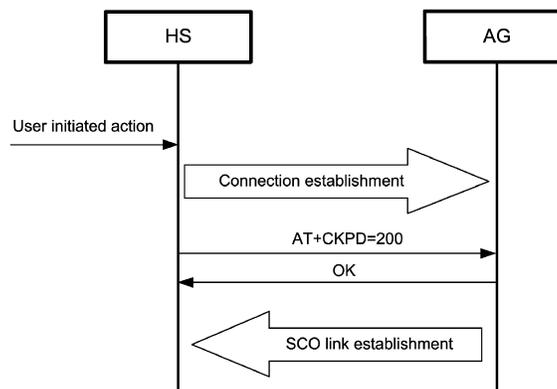
**Table 87. Outgoing Audio Connection with LMX9838 as HSAG (continued)**

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
00 / 01	RX	Confirm	02,43,0F,02,00,54,00,02,03	Rx: Event: Send Data, Status: 00, Local Port: 02
	TX	Request	02,52,5D,08,00,B7,12,34,56,78,9A.BC,80,00,03	Tx: Cmd: Establish SCO Link, BdAddr: 123456789ABC, Packet Type: 8000
00 / 01	RX	Confirm	02,43,5D,07,00,A7,00,12,34,56,78,9A.BC,03	Rx: Event: Establish SCO Link, Status: 00, BdAddr: 123456789ABC
00 / 01	RX	Indicator	02,69,5D,07,00,CD,00,12,34,56,78,9A.BC,03	Rx: Event: SCO Link Established, Status: 00, BdAddr: 123456789ABC

In some applications, the headset might send additional commands for volume control or status. Please refer to [Section 7.4.2](#) for the complete list of AT commands required by the headset profile.

### 6.4.3.5.3 Incoming Audio Connection from Headset

The Headset profile also defines the scenario in which the headset initiates the link to the HSAG.



**Figure 69. Incoming Audio Connection on HSAG as Defined .**

As [Figure 69](#) indicates, the headset requests a SPP link and sends the AT+CKPD=200 command. The HSAG needs to accept the incoming link and afterwards establish the SCO as already described for an outgoing link.

The headset will contact the HSAG on the RFCOMM port, the HSAG has been assigned to.

[Figure 70](#) and [Table 88](#) show the profile flow necessary to accept an incoming link from a headset using the LMX9838 in an HSAG application. The LMX9838 in this example is using the default setting, meaning configured for automatic mode. Because of this, it will switch to transparent mode as soon as the incoming SPP link has been established. Therefore a UART BREAK is sent before the SCO links can be sent.

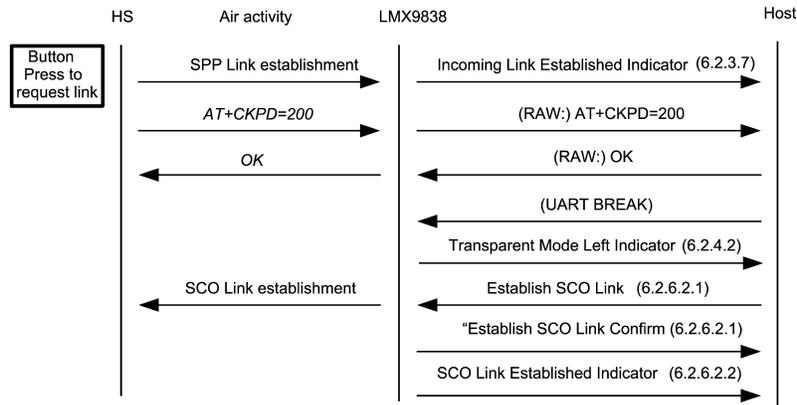


Figure 70. Incoming HS Connection as HSAG

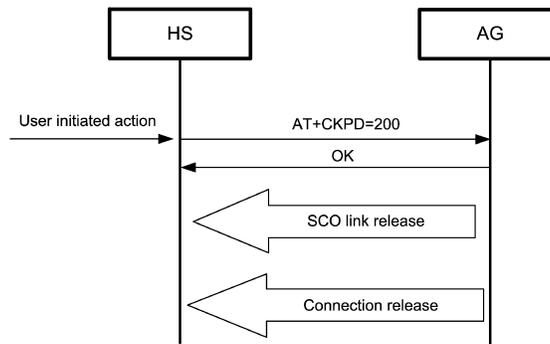
Table 88. Incoming Headset Connection

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
	TX	Request	02,69,0C,07,00,7C,12,34,56,78,9A,BC,02,03	Rx: Event: Incoming Link Established, BdAddr: 123456789ABC, Local Port: 02
00 / 01 / 02 / 03	RX	RAW Data	41,54,2B,43,4B,50,44,3D,32,30,30,0D	Rx(RAW): 41,54,2B,43,4B,50,44,3D,32,30,30,0D
	TX	RAW Data	0D,0A,4F,4B,0D,0A	Tx(RAW): 0D,0A,4F,4B,0D,0A
	TX	UART BREAK		Rx(RAW): 00
00 / 01	RX	Indicator	02,69,11,02,00,7C,02,00,03	Rx: Event: Transparent Mode, Local Port: 02, Mode: 00
	TX	Request	02,52,5D,08,00,B7,12,34,56,78,9A,BC,80,00,03	Tx: Cmd: Establish SCO Link, BdAddr: 123456789ABC, Packet Type: 8000
00 / 01	RX	Confirm	02,43,5D,07,00,A7,00,12,34,56,78,9A,BC,03	Rx: Event: Establish SCO Link, Status: 00, BdAddr: 123456789ABC
00 / 01	RX	Indicator	02,69,5D,07,00,CD,00,12,34,56,78,9A,BC,03	Rx: Event: SCO Link Established, Status: 00, BdAddr: 123456789ABC

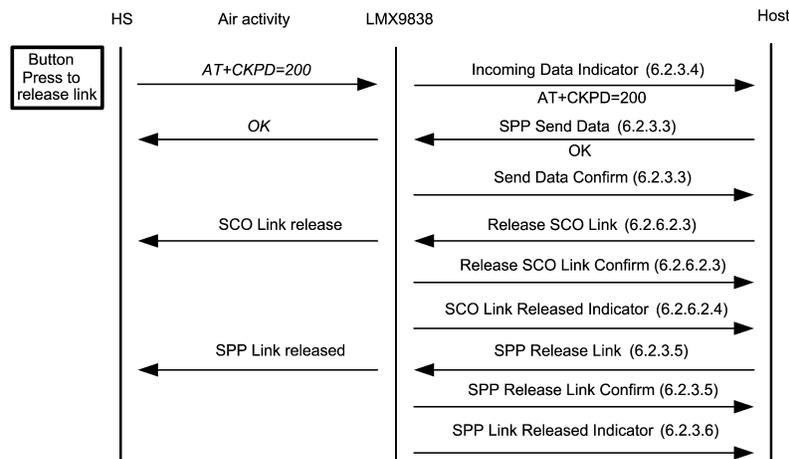
#### 6.4.3.5.4 Audio Connection Release from the HS

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

The LMX9838 confirms the released link by the standard "SPP Link Released Indicator" (Section 7.2.3.6). See following the Bluetooth specification description for the realization with LMX9838. In case the HSAG releases the link the LMX9838 will send the same event.



**Figure 71. Audio Link Release by the HS**



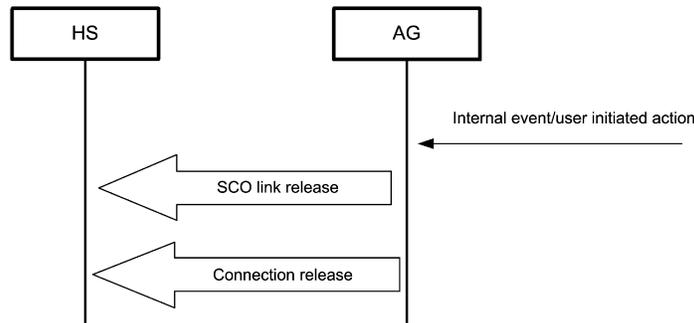
**Figure 72. Audio Link Release by the HS**

**Table 89. Audio Link Release by the HS**

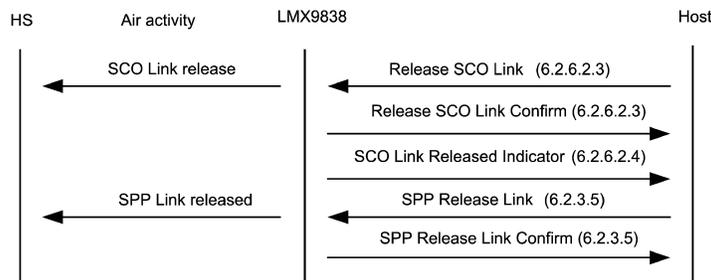
Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
00 / 01	RX	Indicator	02,69,10,0F,00,88,02,0C,00,41,54,2B,43,4B,50,44,3D,32,30,30,0D03	Rx: Event: Incoming Data, Local Port: 02, Received Data: 41542B434B50443D3230300D
	TX	Request	02,52,0F,09,00,6A,02,06,00,0D,0A,4F,4B,0D,0A,03	Tx: Cmd: Send Data, Local Port: 02, Payload Data: 0D0A4F4B0D0A
00 / 01	RX	Confirm	02,43,0F,02,00,54,00,02,03	Rx: Event: Send Data, Status: 00, Local Port: 02
	TX	Request	02,52,5E,00,00,B0,03	Tx: Cmd: Release SCO Link
00 / 01	RX	Confirm	02,43,5E,01,00,A2,00,03	Rx: Event: Release SCO Link, Status: 00
00 / 01	RX	Indicator	02,69,5E,02,00,C9,00,00,03	Rx: Event: SCO Link Released, Status: 00, HCI Reason: 00
	TX	Request	02,52,0D,01,00,60,02,03	Tx: Cmd: Release Link, Local Port: 02
00 / 01	RX	Confirm	02,43,0D,02,00,52,00,02,03	Rx: Event: Release Link, Status: 00, LocalPort: 02
00 / 01	RX	Indicator	02,69,0E,02,00,79,00,02,03	Rx: Event: Link Released, Reason: 00, Local Port: 02
00 / 01	RX	Indicator	02,69,51,07,00,C1,69,9A,01,A4,07,00,16,03	Rx: Event: ACL Terminated, BdAddr: 699A01A40700, Reason: 16

**6.4.3.5.5 Audio Connection Release Initiated by the HSAG**

The release of the audio connection uses the standard LMX9838 commandset, no further AT command is required.



**Figure 73. Audio Link Release Initiated by the HSAG**



**Figure 74. Audio Link Release Initiated by the HSAG**

**Table 90. Audio Link Release by the HSAG**

Filter	Direction	What	Hex Code	Interpreted by Simply Blue Commander
	TX	Request	02,52,5E,00,00,B0,03	Tx: Cmd: Release SCO Link
00 / 01	RX	Confirm	02,43,5E,01,00,A2,00,03	Rx: Event: Release SCO Link, Status: 00
00 / 01	RX	Indicator	02,69,5E,02,00,C9,00,00,03	Rx: Event: SCO Link Released, Status: 00, HCI Reason: 00
	TX	Request	02,52,0D,01,00,60,02,03	Tx: Cmd: Release Link, Local Port: 02
00 / 01	RX	Confirm	02,43,0D,02,00,52,00,02,03	Rx: Event: Release Link, Status: 00, LocalPort: 02
00 / 01	RX	Indicator	02,69,0E,02,00,79,00,02,03	Rx: Event: Link Released, Reason: 00, Local Port: 02
00 / 01	RX	Indicator	02,69,51,07,00,C1,69,9A,01,A4,07,00,16,03	Rx: Event: ACL Terminated, BdAddr: 699A01A40700, Reason: 16

**7 LMX9838 Command Interface**

**7.1 UART PROTOCOL PRINCIPLES**

The LMX9838 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.

All packets sent in both directions are constructed after the following model:

**Table 91. 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 -----					

### 7.1.1 Framing

The connection is considered “Error free”. But for packet recognition and synchronization, some framing is used.

### 7.1.2 Start Delimiter

The start delimiter indicates the LMX9838 the beginning of a new package. The “STX” char is used as start delimiter.

STX = 0x02

### 7.1.3 Packet Type Identification

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

**Table 92. Packet Type Identification**

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

All other values are reserved.

### 7.1.4 Opcode

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

**Table 93. 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
GAP_SET_FIXED_PIN	0x17
GAP_GET_PIN	0x75
GAP_GET_SECURITY_MODE	0x18
GAP_SET_SECURITY_MODE	0x19
GAP_REMOVE_PAIRING	0x1B

**Table 93. Opcode Values (continued)**

Opcode	Value
GAP_LIST_PAIRED_DEVICES	0x1C
GAP_ENTER_SNIFF_MODE	0x21
GAP_EXIT_SNIFF_MODE	0x37
GAP_ENTER_PARK_MODE	0x38
GAP_EXIT_PARK_MODE	0x39
GAP_ENTER_HOLD_MODE	0x3A
GAP_SET_LINK_POLICY	0x3B
GAP_GET_LINK_POLICY	0x3C
GAP_POWER_SAVE_MODE_CHANGED	0x3D
GAP_ACL_ESTABLISHED	0x50
GAP_ACL_TERMINATED	0x51
GAP_SET_AUDIO_CONFIG	0x59
GAP_GET_AUDIO_CONFIG	0x5A
GAP_ESTABLISH_SCO_LINK	0x5D
GAP_RELEASE_SCO_LINK	0x5E
GAP_MUTE_MIC	0x5F
GAP_SET_VOLUME	0x60
GAP_GET_VOLUME	0x61
GAP_CHANGE_SCO_PACKET_TYPE	0x62
SPP_SET_PORT_CONFIG	0x07
SPP_GET_PORT_CONFIG	0x08
SPP_PORT_CONFIG_CHANGED	0x09
SPP_ESTABLISH_LINK	0x0A
SPP_LINK_ESTABLISHED	0x0B
SPP_INCOMMING_LINK_ESTABLISHED	0x0C
SPP_RELEASE_LINK	0x0D
SPP_LINK_RELEASED	0x0E
SPP_SEND_DATA	0x0F
SPP_INCOMING_DATA	0x10
SPP_TRANSPARENT_MODE	0x11
SPP_CONNECT_DEFAULT_CON	0x12
SPP_STORE_DEFAULT_CON	0x13
SPP_GET_LIST_DEFAULT_CON	0x14
SPP_DELETE_DEFAULT_CON	0x15
SPP_SET_LINK_TIMEOUT	0x57
SPP_GET_LINK_TIMEOUT	0x58
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 93. Opcode Values (continued)**

Opcode	Value
SDAP_CONNECTION_LOST	0x34
SDAP_SERVICE_BROWSE	0x35
SDAP_SERVICE_SEARCH	0x36
SDAP_SERVICE_REQUEST	0x1E
SDAP_ATTRIBUTE_REQUEST	0x3F
CHANGE_LOCAL_BDADDRESS	0x27
CHANGE_NVS_UART_SPEED	0x23
CHANGE_UART_SETTINGS	0x48
SET_PORTS_TO_OPEN	0x22
GET_PORTS_TO_OPEN	0x1F
RESTORE_FACTORY_SETTINGS	0x1A
STORE_CLASS_OF_DEVICE	0x28
FORCE_MASTER_ROLE	0x1D
READ_OPERATION_MODE	0x49
WRITE_OPERATION_MODE	0x4A
SET_DEFAULT_LINK_POLICY	0x4C
GET_DEFAULT_LINK_POLICY	0x4D
SET_EVENT_FILTER	0x4E
GET_EVENT_FILTER	0x4F
SET_DEFAULT_LINK_TIMEOUT	0x55
GET_DEFAULT_LINK_TIMEOUT	0x56
SET_DEFAULT_AUDIO_CONFIG	0x5B
GET_DEFAULT_AUDIO_CONFIG	0x5C
SET_DEFAULT_LINK_LATENCY	0x63
GET_DEFAULT_LINK_LATENCY	0x64
SET_CLOCK_FREQUENCY	0x67
GET_CLOCK_FREQUENCY	0x68
SET_PCM_SLAVE_CONFIG	0x74
ENABLE_SDP_RECORD	0x29
DELETE_SDP_RECORDS	0x2A
STORE_SDP_RECORD	0x31
RESET	0x26
LMX9838_READY	0x25
TEST_MODE	0x24
WRITE_ROM_PATCH	0x47
READ_RSSI	0x20
RF_TEST_MODE	0x4B
DISABLE_TL	0x52
TL_ENABLED	0x53
HCI_COMMAND	0x65
AWAIT_INITIALIZATION_EVENT	0x66
ENTER_BLUETOOTH_MODE	0x66
SET_CLOCK_AND_BAUDRATE	0x69
SET_GPIO_WPU	0x6B
GET_GPIO_STATE	0x6C
SET_GPIO_DIRECTION	0x6D

**Table 93. Opcode Values (continued)**

Opcode	Value
SET_GPIO_OUTPUT_HIGH	0x6E
SET_GPIO_OUTPUT_LOW	0x6F
READ_NVS	0x72
WRITE_NVS	0x73

### 7.1.5 Data Length

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

### 7.1.6 Packet Data

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

### 7.1.7 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.

### 7.1.8 End Delimiter

The “ETX” char is used as end delimiter.

ETX = 0x03

### 7.1.9 Retransmission

The connection is considered “Error free”, hence no need for implementing time-outs and retransmissions.

### 7.1.10 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.

### 7.1.11 Byte Order

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

## 7.2 COMMAND SET

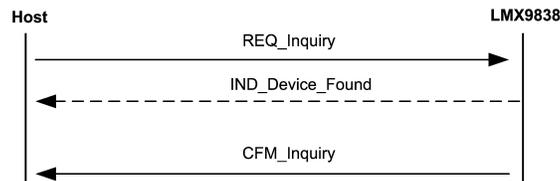
The LMX9838 implements a complete command set for bluetooth operation and local configuration.

The command set is based on a request/confirm scheme meaning any command will be confirmed by an appropriate event including the same opcode.

### 7.2.1 Searching for Remote Devices

The first step to establish a link to another device is to discover the devices in range. The discovering process is called “Inquiry”.

### 7.2.1.1 Inquiry



**Figure 75. Inquiry Command Flow**

**Table 94. Inquiry Command**

Description	Initiates a search for other Bluetooth devices.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	GAP_INQUIRY	
DataLength	3	
Data	Length	Duration of inquiry
	1 byte	Range: 0x01 -0x30 (1.28s - 61.44s)
	NumResponses	Maximum number of responses
	1 byte	Range: 0x00 - 0xFF
		0x00 = Unlimited number of responses.
	Mode	General Inquiry 0x00
	1 Byte	Limited Inquiry 0x01

**Table 95. Inquiry Confirm**

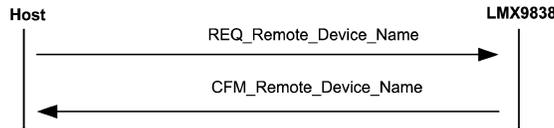
Description	Confirms that the search for other Bluetooth devices is complete.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_INQUIRY	
DataLength	1	
Data	Status	ERROR_OK
	1 byte	ERROR_DURATION_OUT_OF_RANGE
		ERROR_INVALID_MODE
		ERROR_INVALID_NO_OF_PARAMETERS

**Table 96. Device Found Indicator**

Description	Indicates that a device has been found.	
Firmware	0106, 02xx	
PacketType	IND	
Opcode	GAP_DEVICE_FOUND	
DataLength	9	
Data	BdAddr	Bluetooth device address of the found device.
	6 bytes	
	DeviceClass	Class of the found device.
	3 byte	

**7.2.1.2 Get Friendly Name of the Remote Device**

As seen the Device Found Indicator only delivers the BD\_Addr and the Class of Device of Remote Devices. To get the friendly name of the device a separate command has to be used.



**Figure 76. Get Remote Device Name Flow**

**Table 97. Get Remote Device Name Command**

Description	Request the user-friendly name from a known remote Bluetooth device.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	GAP_REMOTE_DEVICE_NAME	
DataLength	6	
Data	BdAddr 6 byte	Bluetooth device address for the remote device

**Table 98. Get Remote Device Name Confirm**

Description	Confirm to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_REMOTE_DEVICE_NAME	
DataLength	8+ NameLength if ok, otherwise 8	
Data	Status 1 byte	ERROR_OK ERROR_INVALID_NO_OF_PARAMETERS ERROR_TIMEOUT
	BdAddr 6 byte	Bluetooth device address for the remote device
	NameLength 1 byte	Number of bytes in device name
	DeviceName Length bytes	The user-friendly name of the remote device. NULL terminated. Maximum length is 40 bytes.

**7.2.2 SDAP Client Commands**

Establishing a link to another device requires that devices BD\_Addr but also the RFCOMM Port Number, the profile to connect to is registered at. If remote Com Port is unknown, it can be requested by a service discovery request using the SDAP client.

A SDAP request can only be done via an establish SDAP link. So the Command flow could be as following. To get information about the remote service, the host can use either “Service Browse”, “Service Search” or “Service Request” and “Attribute Request”. Usually the “Service Browse” is enough.

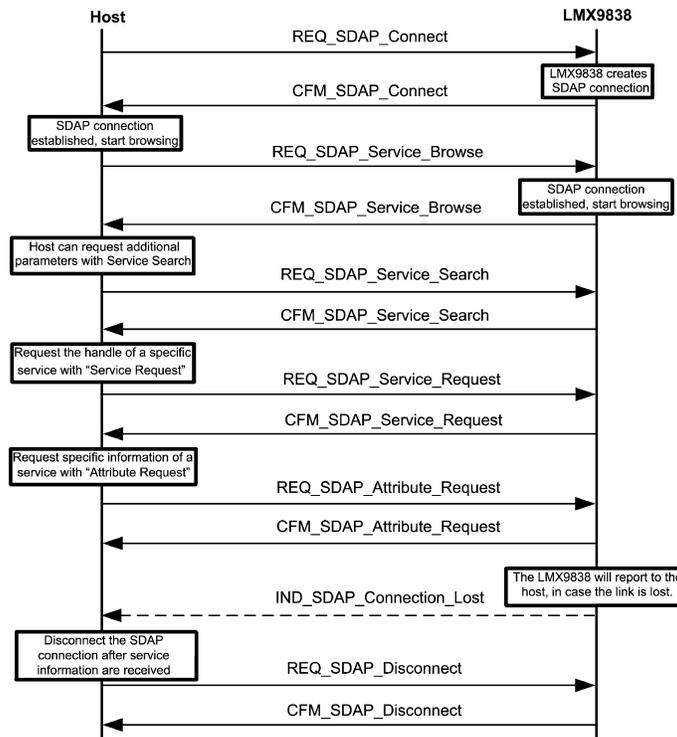


Figure 77. Requesting the Services of a Remote Device

7.2.2.1 SDAP Connect

The SDAP Connect Request forces the LMX9838 to create a SDAP link to another device. This command is required for further SDAP Service Requests.

Table 99. SDAP Connect Request

Description	Creates an SDP connection to a remote device. Only one SDP connection can be active.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SDAP_CONNECT	
DataLength	6	
Data	BdAddr	The Bluetooth address of the remote device to connect to.
	6 bytes	

Table 100. SDAP Connect Confirm

Description	Creates an SDP connection to a remote device. Only one SDP connection can be active.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SDAP_CONNECT	
DataLength	1	
Data	Status	ERROR_OK
	1 byte	ERROR_INVALID_NO_OF_PARAMETERS
		ERROR_CONNECTION_FAILED

### 7.2.2.2 SDAP Disconnect

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

**Table 101. Disconnect Request**

Description	This command disconnects the active SDP connection.
Firmware	0106, 02xx
PacketType	REQ
Opcode	SDAP_DISCONNECT
DataLength	0

**Table 102. Disconnect Confirm**

Description	Confirms the request above	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SDAP_DISCONNECT	
DataLength	1	
Data	Status	ERROR_OK
	1 byte	ERROR_INVALID_NO_OF_PARAMETERS

### 7.2.2.3 SDAP Connection Lost Indicator

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

**Table 103. Connection Lost Indicator**

Description	Notification sent to the application when a loss of the SDP connection is detected.
Firmware	0106, 02xx
PacketType	IND
Opcode	SDAP_CONNECTION_LOST
DataLength	0

### 7.2.2.4 SDAP Service Browse

The SDAP Service Browse can be used to get the RFCOMM Port Numbers of all or only specific Service Classes. The search mechanism is based on the 16bit-UUID for the services. The actual list of UUIDs can be found within the "Bluetooth Assigned Numbers" Document of the Bluetooth SIG.

**Table 104. 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 105. SDAP Service Browse Request**

Description	This command is used to browse the service record of the remote device.
Firmware	0106, 02xx
PacketType	REQ
Opcode	SDAP_SERVICE_BROWSE
DataLength	2

**Table 105. SDAP Service Browse Request (continued)**

Description	This command is used to browse the service record of the remote device.	
Data	BrowseGroupID	The requested browse group (16 bit UUID).
	2 bytes	The UUID has to be byte swapped within the command,
		e.g. to search for SPP entries the full command is
		02 52 35 02 00 89 01 11 03

**Table 106. SDAP Service Browse Confirm**

Description	Confirms the request above	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SDAP_SERVICE_BROWSE	
DataLength	2+NoOfServices*(6 + NameLength)	
Data	Status	ERROR_OK
	1 byte	ERROR_INVALID_NO_OF_PARAMETERS
		ERROR_RESULT_TOO_LARGE
		ERROR_TRUNCATED_ANSWER
	NoOfServices	Number of services found on remote device.
	1 byte	
	For each service	
	BrowseGroupID 2 bytes	The browse group UUID that the service belongs to.
	ServiceID	The service UUID.
	2 bytes	
	PortNr	RFCOMM port number.
	1 byte	The port which has to be used for link establishment to that service.
	NameLength	The number of bytes in the service name.
	1 byte	
	ServiceName	The name of the service.
	NameLength bytes	

### 7.2.2.5 SDAP Service Search

The SDAP Service Search command offers the ability to search for specific attributes for a service. The attribute IDs can be found at the Bluetooth SIG website under “Bluetooth Assigned numbers”.

Example: To request the “ProfileDescriptorList” of a SPP entry, the following command could be used:

02,52,36,06,00,8E,01,11,01,01,00,04,03

while:

- 11 01 reflects the UUID for the SPP service (0x1101) and
- 00 04 reflects the UUID for the “ProfileDescriptorList” (0x0004)

---

**NOTE:** As an exception to all other commands, this command does NOT required byte swapped parameters. So the UUIDs have to be filled in as defined in "Bluetooth SIG: Bluetooth Assigned Numbers, <https://programs.bluetooth.org/Technical/AssignedNumbers/baseband.htm>. See example above.

---

**Table 107. SDAP Service Search Request**

Description	This command is used to search for services in the service record of the remote device.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SDAP_SERVICE_SEARCH	
DataLength	2 + 2*SearchPatternLength + 2*AttributesLength	
Data	SearchPatternLength	Number of 16-bit UUIDs in Search pattern List. Note: Must be less than 86 elements.
	1 bytes	
	SearchPattern	List of the requested services. The search pattern list is a list of 16-bit UUIDs of the requested services.
	AttributesLength	Number of 16-bit UUIDs in attributeld list. Note: Must be less than 86 elements.
	1 byte	
	Attributes	List of requested attributes for the requested services. The attribute list is a list of 16-bit UUIDs for the requested attributes.

**Table 108. SDAP Service Search Confirm**

Description	Confirms the request above	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SDAP_SERVICE_SEARCH	
DataLength	3 + Length	
Data	Status	ERROR_OK
	1 byte	ERROR_INVALID_NO_OF_PARAMETERS
		ERROR_UNKNOWN_ERROR
		ERROR_RESULT_TOO_LARGE
	Length	Length of the result of the search. Maximum 330 bytes
	2 byte	
	Result	Result of the search
	Length bytes	

### 7.2.2.6 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 109. SDAP Service Request**

Description	This command is used the service record handles, from a remote device, for the given services in the search pattern.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SDAP_SERVICE_REQUEST	
DataLength	1 + 2* SearchPatternLength	
Data	SearchPatternLength	Number of 16-bit UUIDs in Search pattern List. <b>Note:</b> Must be less than 86 elements.
	1 bytes	
	SearchPattern	List of the requested services. The search pattern list is a list of 16-bit UUIDs of the requested services
	< 2* SearchPatternLength > bytes	

**Table 110. SDAP Service Request Confirm**

Description	Confirms the request above	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SDAP_SERVICE_REQUEST	
DataLength	2 + 4*Length	
Data	Status	ERROR_OK
	1 byte	ERROR_INVALID_NO_OF_PARAMETERS
		ERROR_UNKNOWN_ERROR
		ERROR_RESULT_TOO_LARGE
	Length	Number of 32-bit service record handles returned from remote device.
	1 byte	
	Result	The received 32-bit service record handles from the remote device.
	<4*Length> bytes	

### 7.2.2.7 SDAP Attribute Request

Instead of browsing the whole list of services within the remote database it is also possible to search only for specific attributes within a chosen entry. This command is based on the attribute “connection handle” of that specific entry.

Connection handles can be retrieved by the “SDAP Service Request” Command. (see [Section 7.2.2.6](#))

**Table 111. SDAP Attribute Request**

Description	This command is used to get the given attributes for a given service record handle.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SDAP_ATTRIBUTE_REQUEST	
DataLength	5 + 2*AttributesLength	
Data	Handle	The 32-bit service record handle returned for a given service by SDAP_SERVICE_REQUEST.
	4 bytes	
	AttributesLength	Number of 16-bit UUIDs in attributeld list. <b>Note:</b> Must be less than 86 elements.
	1 byte	
	Attributes	List of requested attributes for the requested services. The attribute list is a list of 16-bit UUIDs for the requested attributes.
	< 2 * AttributesLength> bytes	

**Table 112. SDAP Attribute Request Confirm**

Description	Confirms the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SDAP_ATTRIBUTE_REQUEST	
DataLength	3 + Length	
Data	Status	ERROR_OK
	1 byte	ERROR_INVALID_NO_OF_PARAMETERS
		ERROR_UNKNOWN_ERROR
		ERROR_RESULT_TOO_LARGE
	Length	Length of the result of the search. Maximum 330 bytes.
	2 byte	
	Result	Result of the search

### 7.2.3 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 RFCComm 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 LMX9838 will start to page and try to connect to the remote device. The SPP\_Link\_Establishment indicator returns an error code reporting the success of the link establishment.

The flow also shows the procedure of sending data to the remote device using the “Send Data” Command. Incoming data are indicated by the “Incoming data event”.

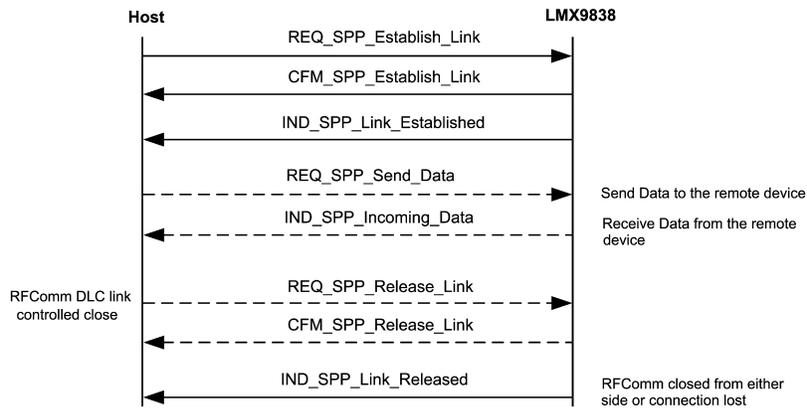


Figure 78. SPP Link Command Flow

#### 7.2.3.1 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 RFCComm Channel on the remote device is required.

Table 113. Establish Link Request

Description	Establish a DLC link to remote Bluetooth device .	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SPP_ESTABLISH_LINK	
DataLength	8	
Data	LocalPort	Local RFCOMM port number. Range 1-30
	1 byte	
	BdAddr	Bluetooth device address for the remote device
	6 byte	
	RemotePortNumber	Remote device RFCOMM port number.
	1 byte	(To be found using SDAP)

Table 114. Establish Link Confirm

Description	Confirm that the DLC link establishment is initiated. Note: This confirm does NOT indicate link establishment, only that link establishment is in progress. When link establishment response is received from the core, a SPP_LINK_ESTABLISHED indication is sent from the LMX9838.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SPP_ESTABLISH_LINK	

**Table 114. Establish Link Confirm (continued)**

<b>Description</b>	<b>Confirm that the DLC link establishment is initiated.</b> <b>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 LMX9838.</b>	
DataLength	2	
Data	Status	ERROR_OK
	1 byte	ERROR_SPP_PORT_BUSY
		ERROR_SPP_PORT_NOT_OPEN
		ERROR_SPP_INVALID_PORT
		ERROR_SPP_AUTOMATIC_CONNECTIONS_PROGRESSING
LocalPort	Local RFCOMM port number. Range 1-30	
	1 byte	

### 7.2.3.2 SPP Link Established Indicator

**Table 115. SPP Link Established Indicator**

<b>Description</b>	<b>Indication of establishment of a locally requested DLC link.</b>	
Firmware	0106, 02xx	
PacketType	IND	
Opcode	SPP_LINK_ESTABLISHED	
DataLength	9	
Data	Status	Refer to RFCOMM Error Codes.
	1 byte	
	BdAddr	Bluetooth device address for the remote device.
	6 byte	
	LocalPort	Local RFCOMM port number. Range 1-30.
	1 byte	
	RemotePortNumber	Remote device RFCOMM port number.
	1 byte	

### 7.2.3.3 SPP Send Data

If not switched to transparent, data have to be sent to a remote device using this command. The local RFCOMM Port is used to address the remote device.

**Table 116. SPP Send Data**

<b>Description</b>	<b>Send data on a SPP link to remote Bluetooth device</b>	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SPP_SEND_DATA	
DataLength	3 + <PayloadSize>	
Data	LocalPort	Local RFCOMM port number. Range 1-30
	1 byte	
	PayloadSize	Number of data bytes to send.
	2 bytes	Valid range is 1 to 330 bytes.
	PayloadData	The data to send.
	<PayloadSize> bytes	

**Table 117. SPP Send Data Confirm**

Description	Confirm to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SPP_SEND_DATA	
DataLength	2	
Data	Status	ERROR_OK
	1 byte	ERROR_LIMIT
		ERROR_UNABLE_TO_SEND
		ERROR_CURRENTLY_NO_BUFFER
		ERROR_NO_CONNECTION
		ERROR_SPP_INVALID_PORT
		ERROR_SPP_PORT_NOT_OPEN
LocalPort	Local RFCOMM port number. Range 1-30	
1 byte		

### 7.2.3.4 Incoming Data Indicator

**Table 118. SPP Incoming Data Indicator**

Description	Incoming data on a DLC link, from a remote Bluetooth device	
Firmware	0106, 02xx	
PacketType	IND	
Opcode	SPP_INCOMING_DATA	
DataLength	3 + <PayloadSize>	
Data	LocalPort	Local RFCOMM port number. Range 1-30
	1 byte	
	PayloadSize	Number of data bytes to send.
	2 bytes	Valid range is 1 to 330 bytes.
	PayloadData	The data to send.
	<PayloadSize> bytes	

### 7.2.3.5 SPP Release Link

**Table 119. SPP Release Link Request**

Description	Release a DLC link to remote Bluetooth device	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SPP_RELEASE_LINK	
DataLength	1	
Data	LocalPort	Local RFCOMM port number. Range 1-30
	1 byte	

**Table 120. 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 LMX9838.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SPP_RELEASE_LINK	

**Table 120. SPP Release Link Confirm (continued)**

Description	Confirm that the release is initiated. When the release is complete, a SPP_LINK_RELEASED indication is sent from the LMX9838.	
DataLength	2	
Data	Status	ERROR_OK
	1 byte	ERROR_NO_CONNECTION
		ERROR_SPP_INVALID_PORT
		ERROR_SPP_PORT_NOT_OPEN
LocalPort	LocalPort	Local RFCOMM port number. Range 1-30
	1 byte	

### 7.2.3.6 SPP Link Released Indicator

**Table 121. SPP Link Released Indicator**

Description	Indicates that a DLC link is released. The link release may have been initiated locally or remote, or could be caused by a loss of link (disturbance, dead device,... )	
Firmware	0106, 02xx	
PacketType	IND	
Opcode	SPP_LINK_RELEASED	
DataLength	2	
Data	Reason	Refer to <a href="#">Section 7.3.4</a>
	1 byte	
LocalPort	LocalPort	Local RFCOMM port number. Range 1-30
	1 byte	

### 7.2.3.7 Incoming Link Established Indicator

In case a remote device creates a link to the LMX9838, the device will indicate the successful link establishment by sending by the “Incoming Link Established Indicator”. The packet includes the BD\_Addr of the remote device and the local RFCOMM Port it connected to.

**Table 122. Incoming Link Established Indicator**

Description	Indication of establishment of a remotely requested DLC link.	
Firmware	0106, 02xx	
PacketType	IND	
Opcode	SPP_INCOMING_LINK_ESTABLISHED	
DataLength	7	
Data	BdAddr	Bluetooth device address for the remote device
	6 byte	
LocalPort	LocalPort	Local RFCOMM port number. Range 1-30
	1 byte	

### 7.2.3.8 Read RSSI

**Table 123. Read RSSI Request**

Description	This command will read out the current RSSI value for the existing link to a remote device.
Firmware	0106, 02xx

**Table 123. Read RSSI Request (continued)**

<b>Description</b>	<b>This command will read out the current RSSI value for the existing link to a remote device.</b>	
PacketType	REQ	
Opcode	READ_RSSI	
DataLength	6	
Data	BdAddress 6 byte	Bluetooth Address of the remote device for which to read out the current RSSI value.

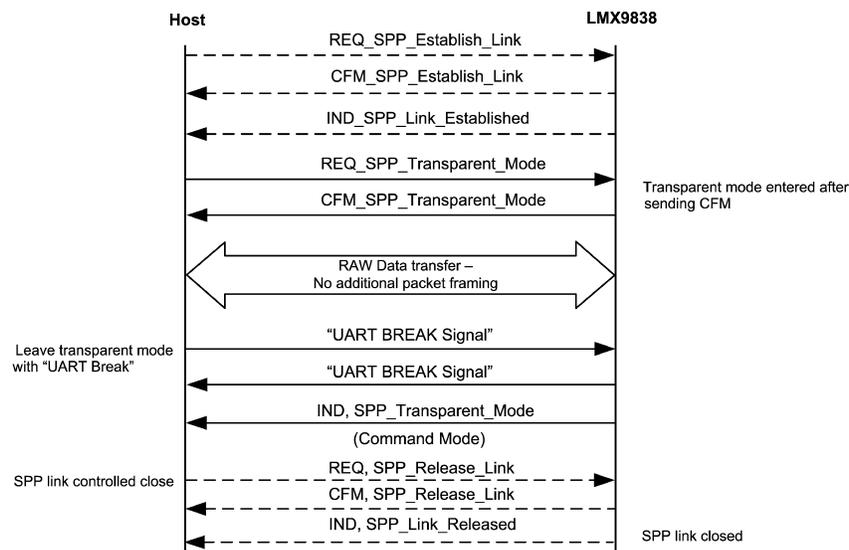
**Table 124. Read RSSI Confirm**

<b>Description</b>	<b>Confirms the request above</b>	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	READ_RSSI	
DataLength	2	
Data	Status 1 byte	ERROR_OK ERROR_UNKNOWN_ERROR ERROR_INVALID_NO_OF_PARAMETERS
	RSSI 1 byte	The RSSI value will be indicated in three steps. 0x01: Too High 0x00: Good 0xFF: Too Low

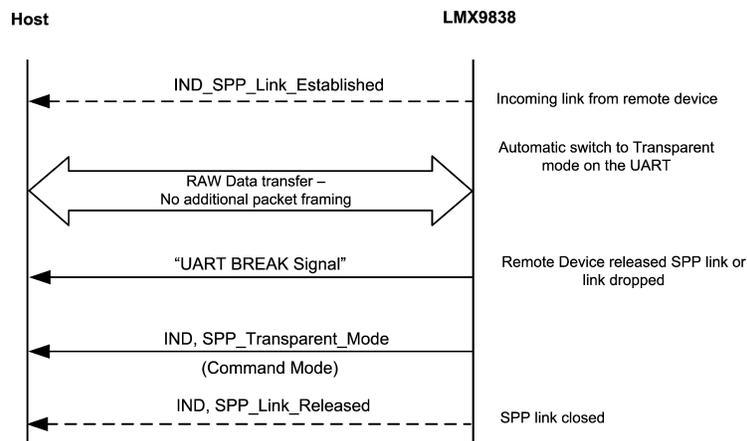
**7.2.4 Transparent Mode**

Transparent Mode offers the ability to switch off the Command Interface on the LMX9838 and use it as a cable replacement. This means data can be sent over a bluetooth link just by routing them to the LMX9838 without any package framing.

Transparent mode can only be enabled if one SPP link is established. Transparent mode can not be used if a device has two or more active SPP links.



**Figure 79. Transparent Mode; initiated by the Transparent Mode Request**



**Figure 80. Transparent Mode; Automatically Activated in Transparent Slave**

### 7.2.4.1 Transparent Mode Command

**Table 125. Transparent Mode Request**

Description	Switch to transparent mode on a SPP link to remote Bluetooth device	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SPP_TRANSPARENT_MODE	
DataLength	1	
Data	LocalPort	Local RFCOMM port number. Range 1-30
	1 byte	

**Table 126. Transparent Mode Confirm**

Description	Confirm to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SPP_TRANSPARENT_MODE	
DataLength	2	
Data	Status	ERROR_OK
	1 byte	ERROR_SPP_PORT_NOT_OPEN
		ERROR_SPP_INVALID_PORT
		ERROR_SPP_MULTIPLE_CONNECTIONS
		ERROR_NO_CONNECTION
LocalPort	Local RFCOMM port number. Range 1-30	

### 7.2.4.2 Transparent Mode Left Indicator

**Table 127. Transparent Mode Left Indicator**

Description	Indication from Simply Blue that transparent mode is left.
Firmware	0106, 02xx
PacketType	IND
Opcode	SPP_TRANSPARENT_MODE

**Table 127. Transparent Mode Left Indicator (continued)**

Description	Indication from Simply Blue that transparent mode is left.	
DataLength	2	
Data	LocalPort	Local RFCOMM port number. Range 1-30
	1 byte	
	Mode	Command Mode = 0
	1 byte	Transparent Mode = 1

## 7.2.5 Link Timeout

The Link Timeout commands report and set the Supervision timeout setting for an established link. The link timeout which is set during link establishment on all incoming and outgoing links is read out of the NVS. The timeout can either be changed on an existing link or for each link by configuring the default setting in NVS.

See also [Section 5.1.2.6](#) for a detailed description of the feature.

### 7.2.5.1 Set Default Link Timeout

**Table 128. Set Default Link Timeout Request**

Description	This command is used to change the default link supervision timeout. The default link supervision timeout is set during connection setup. The default link supervision timeout setting is stored in NVS.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SET_DEFAULT_LINK_TIMEOUT	
DataLength	2	
Data	LinkTimeout	The link supervision timeout in slots (0,625ms). The default value stored in the NVS after a factory reset is 20s.
	2 byte	0x0000: No link supervision timeout (the timer is disabled)
		0x0190-0xFFFF: Valid timeout range (in slots)
		0x7D00: The default value (20s)

**Table 129. Set Default Link Timeout Confirm**

Description	Response to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SET_DEFAULT_LINK_TIMEOUT	
DataLength	1	
Data	Status	ERROR_OK
	1 byte	ERROR_INVALID_NO_OF_PARAMETERS
		ERROR_ILLEGAL_LINK_TIMEOUT

### 7.2.5.2 Get Default Link Timeout

**Table 130. Get Default Link Timeout Request**

Description	This command is used to read the default link supervision timeout setting from NVS.
Firmware	0106, 02xx
PacketType	REQ
Opcode	GET_DEFAULT_LINK_TIMEOUT
DataLength	0

**Table 131. Get Default Link Timeout Confirm**

Description	Response to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GET_DEFAULT_LINK_TIMEOUT	
DataLength	2	
Data	LinkTimeout	The link supervision timeout in slots.
	2 byte	0x0000: No link supervision timeout (the timer is disabled)
		0x0190-0xFFFF: Valid timeout range (in slots)

### 7.2.5.3 Set Link Timeout for an Existing Link

**Table 132. Set Link Timeout Request**

Description	This command is used to change the current ACL link supervision timeout. This command will affect all physical links established to the specified device.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SPP_SET_LINK_TIMEOUT	
DataLength	8	
Data	BdAddr	The Bluetooth address of the remote device of the ACL link for which the Link Supervision Timeout is changed.
	6 bytes	
	LinkTimeout	The link supervision timeout in slots (0.625ms).
	2 byte	0x0000: No link supervision timeout (the timer is disabled)
		0x0190-0xFFFF: Valid timeout range (in slots)
		0x7D00: The default value (20s)

**Table 133. Set Link Timeout Confirm**

Description	Response to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SPP_SET_LINK_TIMEOUT	
DataLength	7	
Data	Status	ERROR_OK
	1 byte	ERROR_UNSPECIFIED_ERROR
		ERROR_INVALID_NO_OF_PARAMETERS
		ERROR_NO_CONNECTION
		ERROR_ILLEGAL_LINK_TIMEOUT
	BdAddr	The Bluetooth address of the remote device of the ACL link for which the Link Supervision Timeout is changed.
	6 bytes	

### 7.2.5.4 Get Link Timeout of an Existing Link

**Table 134. Get Link Timeout Request**

Description	This command is used to get the current link supervision timeout setting for the given ACL link.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SPP_GET_LINK_TIMEOUT	
DataLength	6	
Data	BdAddr 6 bytes	The Bluetooth address of the remote device of the ACL link for which the Link Supervision Timeout is requested.

**Table 135. Get Link Timeout Confirm**

Description	Response to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SPP_GET_LINK_TIMEOUT	
DataLength	9	
Data	Status 1 byte	ERROR_OK ERROR_UNSPECIFIED_ERROR ERROR_INVALID_NO_OF_PARAMETERS ERROR_NO_CONNECTION
	BdAddr 6 bytes	The Bluetooth address of the remote device of the ACL link for which the Link Supervision Timeout is requested.
	LinkTimeout 2 byte	The link supervision timeout in slots (0.625ms). 0x0000: No link supervision timeout (the timer is disabled) 0x0190-0xFFFF: Valid timeout range (in slots)

## 7.2.6 Audio (SCO) Link Support

### 7.2.6.1 Audio Path Configuration

The LMX9838 requires the configuration of the audio path in order to know, how the audio data need to be converted between the Advanced Audio Interface and the Bluetooth Link. The default audio settings allow to store this information in NVS, used for every SCO link.

#### 7.2.6.1.1 Set Default Audio Settings

**Table 136. Set Default Audio Settings Command**

Description	This command is used to set the default audio settings stored in NVS.
Firmware	0106, 02xx
PacketType	REQ
Opcode	SET_DEFAULT_AUDIO_CONFIG
DataLength	2

**Table 136. Set Default Audio Settings Command (continued)**

Description	This command is used to set the default audio settings stored in NVS.	
Data	CodecType	0x00: No codec available (default)
	1 byte	0x01: Motorola MC145483
		0x02: OKI MSM7717
		0x03: PCM slave, requires "PCM Slave Configuration" ( <a href="#">Section 7.2.6.1.3</a> )
		0x04-0xFF: Reserved.
AirFormat	0x00: CVSD (default)	
	1 byte	0x01: $\mu$ -Law
		0x02: A-Law
		0x03-0xFF: Reserved.

### 7.2.6.1.2 Get Default Audio Settings

**Table 137. Get Default Audio Settings Command**

Description	This command is used to get the default audio settings stored in NVS.
Firmware	0106, 02xx
PacketType	REQ
Opcode	GET_DEFAULT_AUDIO_CONFIG
DataLength	0

**Table 138. Get Default Audio Settings Confirm**

Description	Response to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GET_DEFAULT_AUDIO_CONFIG	
DataLength	3	
Data	Status	ERROR_OK
	1 byte	ERROR_INVALID_NO_OF_PARAMETERS
CodecType	1 byte	0x00: No codec available (default)
		0x01: Motorola MC145483
		0x02: OKI MSM7717
		0x03: PCM slave, requires "PCM Slave Configuration" ( <a href="#">Section 7.2.6.1.3</a> )
		0x04-0xFF: Reserved.
AirFormat	1 byte	0x00: CVSD (default)
		0x01: $\mu$ -Law
		0x02: A-Law
		0x03-0xFF: Reserved.

### 7.2.6.1.3 PCM Slave Configuration

The PCM Slave configuration gets active in case the audio settings, configured by "Set Default Audio Settings" ([Section 7.2.6.1.1](#)) or "Set audio settings" ([Section 7.2.6.2.5](#)) has been set to "PCM slave". For a detailed description of the settings, please refer to [Section 5.5.3.1.2](#).

**Table 139. Set PCM Slave Configuration Request**

Description	This command is used to set the PCM slave configuration in the NVS.	
PacketType	REQ	
Opcode	SET_PCM_SLAVE_CONFIG	
DataLength	3	
Data	PcmSlaveConfig	This 16-bit value (LSB first) is used to store the PCM format configuration for the PCM generic slave.
	2 bytes	BIT0-1, Slot selection
		00: use slot 0
		01: use slot 1
		10: use slot 2
		11: use slot 3
		BIT2-3: Number of slots per frame
		00: 1 slot
		01: 2 slots
		10: 3 slots
		11: 4 slots
		BIT4-6: PCM data format
		000: Reserved
		001: 8 bit A-Law
		010: 8 bit $\mu$ -Law
		011: 13 bit linear
		100: 14 bit linear
		101: 15 bit linear
		110: 16 bit linear
		111: Reserved
		BIT7: Frame sync length
		0: short frame sync
		1: long frame sync
	BIT8: Data word length	
	0: 8-bit data word length	
	1: 16-bit data word length	
	BIT9: Frame sync polarity	
	0: use inverted frame sync	
	1: use normal frame sync	
	BIT10-15: Unused, set to 0	
	Fcprs	This value is an unsigned integer indicating the frame clock prescaler for generic PCM slave. The ratio between the bit clock and the frame clock must be written into the ACCR.FCPRS register for the generic PCM slave to operate correctly.
	1 byte	The following equation must be true: $\text{bit\_clock}/(\text{Fcprs} + 1) = \text{frame\_clock}.$
		Example bit clock = 480000, frame sync rate = 8000, Fcprs must be set to 59 since $480000/(59 + 1) = 8000$

**Table 140. Set PCM Slave Configuration Confirm**

Description	Response to the request above.
PacketType	CFM
Opcode	SET_PCM_SLAVE_CONFIG
DataLength	1

**Table 140. Set PCM Slave Configuration Confirm (continued)**

Description	Response to the request above.	
Data	Status	ERROR_OK
	1 byte	ERROR_INVALID_NO_OF_PARAMETERS
		ERROR_INVALID_CODEC_SETTING

### 7.2.6.2 Link Establishment

#### 7.2.6.2.1 Establish SCO Link

**Table 141. Establish SCO Link Command**

Description	This command is used to establish a SCO link to a remote Bluetooth device. An ACL link must be established before a SCO link can be established.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	GAP_ESTABLISH_SCO_LINK	
DataLength	8	
Data	BdAddr	The Bluetooth address of the remote device to which the SCO link is established.
	6 bytes	
	PacketType	0x0020: HV1
	2 bytes	0x0040: HV2
		0x0080: HV3
	The packet types can be combined / ored together in order to enable multiple packet types.	

**Table 142. Establish SCO Link Confirm**

Description	Response to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_ESTABLISH_SCO_LINK	
DataLength	7	
Data	Status	ERROR_OK
	1 byte	ERROR_UNSPECIFIED_ERROR
	BdAddr	The Bluetooth address of the remote device.
	6 bytes	

#### 7.2.6.2.2 SCO Link Established Indicator

**Table 143. SCO Link Established Indicator**

Description	Indicates that a SCO link is established. Either the local device or the remote device may have initiated the link establishment.	
Firmware	0106, 02xx	
PacketType	IND	
Opcode	GAP_ESTABLISH_SCO_LINK	
DataLength	7	

**Table 143. SCO Link Established Indicator (continued)**

Description	Indicates that a SCO link is established. Either the local device or the remote device may have initiated the link establishment.	
Data	Status	ERROR_OK
	1 byte	ERROR_UNSPECIFIED_ERROR
		ERROR_NO_CONNECTION
		ERROR_CONNECTION_FAILED
	BdAddr	The Bluetooth address of the remote.
	6 bytes	

### 7.2.6.2.3 Release SCO Link

**Table 144. Release SCO Link Command**

Description	This command is used to disconnect a SCO link.
Firmware	0106, 02xx
PacketType	REQ
Opcode	GAP_RELEASE_SCO_LINK
DataLength	0

**Table 145. Release SCO Link Confirm**

Description	Response to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_RELEASE_SCO_LINK	
DataLength	1	
Data	Status	ERROR_OK
	1 byte	ERROR_NO_CONNECTION
		ERROR_INVALID_NO_OF_PARAMETERS

### 7.2.6.2.4 SCO Link Released Indicator

**Table 146. SCO Link Released Indicator**

Description	Indicates that a SCO link is released. The link release may have been initiated by the local device, the remote device or it could be caused by a loss of link (disturbance, dead device, etc.)	
Firmware	0106, 02xx	
PacketType	IND	
Opcode	GAP_RELEASE_SCO_LINK	
DataLength	2	
Data	Status	ERROR_OK
	1 byte	ERROR_UNKNOWN_ERROR
		ERROR_NO_CONNECTION
	HciReason	The HCI release reason.
	1 byte	The release reason indicated by the remote application.

### 7.2.6.2.5 Set audio settings

**Table 147. Set Audio Settings Command**

<b>Description</b>	<b>This command is used to set the audio settings the device shall use for the existing or next SCO (audio) link established. This setting will not be stored to NVS. To store a default setting use "Set audio settings" ( <a href="#">Section 7.2.6.2.5</a> )</b>	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	GAP_SET_AUDIO_CONFIG	
DataLength	2	
Data	CodecType	0x00: No codec available
	1 byte	0x01: Motorola MC145483
		0x02: OKI MSM7717
		0x03: PCM Slave, requires "PCM Slave Configuration" ( <a href="#">Section 7.2.6.1.3</a> )
		0x0-0xFF: Reserved.
AirFormat		0x00: CVSD
	1 byte	0x01: $\mu$ -Law
		0x02: A-Law
		0x03-0xFF: Reserved.

**Table 148. Set Audio Settings Confirm**

<b>Description</b>	<b>Response to the request above.</b>	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_SET_AUDIO_CONFIG	
DataLength	1	
Data	Status	ERROR_OK
	1 byte	ERROR_INVALID_NO_OF_PARAMETERS
		ERROR_ILLEGAL_AUDIO_CODEC_TYPE
		ERROR_ILLEGAL_AUDIO_AIR_FORMAT

### 7.2.6.2.6 Get Audio Settings

**Table 149. Get Audio Settings Command**

<b>Description</b>	<b>This command is used to get the audio settings the device uses at an existing link or it shall use for the next SCO (audio) link established.</b>	
	This setting will not read the default settings stored in NVS. To read the default setting use "Get Default Audio Settings" ( <a href="#">Section 7.2.6.1.2</a> )	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	GAP_GET_AUDIO_CONFIG	
DataLength	0	

**Table 150. Get Audio Settings Confirm**

<b>Description</b>	<b>Response to the request above.</b>	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_GET_AUDIO_CONFIG	

**Table 150. Get Audio Settings Confirm (continued)**

Description	Response to the request above.	
DataLength	3	
Data	Status	ERROR_OK
	1 byte	ERROR_INVALID_NO_OF_PARAMETERS
	CodecType	0x00: No codec available
	1 byte	0x01: Motorola MC145483
		0x02: OKI MSM7717
		0x03: PCM Slave, requires "PCM Slave Configuration" ( <a href="#">Section 7.2.6.1.3</a> )
		0x04-0xFF: Reserved.
AirFormat	0x00: CVSD	
	1 byte	0x01: $\mu$ -Law
		0x02: A-Law
		0x03-0xFF: Reserved.

### 7.2.6.2.7 Change SCO Packet Type

**Table 151. Change SCO Packet Type Command**

Description	This command is used to change the packet type used/enabled for the SCO link.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	GAP_CHANGE_SCO_PACKET_TYPE	
DataLength	2	
Data	PacketType	0x0020: HV1
	2 bytes	0x0040: HV2
		0x0080: HV3
		The packet types can be combined / ored together in order to enable multiple packet types.

**Table 152. Change SCO Packet Type Confirm**

Description	Response to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_CHANGE_SCO_PACKET_TYPE	
DataLength	1	
Data	Status	ERROR_OK
	1 byte	ERROR_UNSPECIFIED_ERROR

### 7.2.6.2.8 Change SCO Link Packet Type Indicator

**Table 153. Change SCO Link Packet Type Indicator**

Description	Indicates that the packet type of the SCO link has changed. Either the local device or the remote device may have initiated the change of the packet type of the SCO link.
Firmware	0106, 02xx
PacketType	IND
Opcode	GAP_CHANGE_SCO_PACKET_TYPE
DataLength	3

**Table 153. Change SCO Link Packet Type Indicator (continued)**

Description	Indicates that the packet type of the SCO link has changed. Either the local device or the remote device may have initiated the change of the packet type of the SCO link.	
Data	Status	ERROR_OK
	1 byte	ERROR_UNSPECIFIED_ERROR
		ERROR_NO_CONNECTION
		ERROR_CONNECTION_FAILED
PacketType	0x0020: HV1	
	2 bytes	0x0040: HV2
		0x0080: HV3
		The packet types can be combined / ored together in order to enable multiple packet types.

### 7.2.6.2.9 Mute

**Table 154. Mute Command**

Description	This command is used to mute the microphone.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	GAP_MUTE_MIC	
DataLength	1	
Data	Mute	0x01: Mute
	1 byte	0x00: Un mute

**Table 155. Mute Confirm**

Description	Response to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_SET_VOLUME	
DataLength	1	
Data	Status	ERROR_OK
	1 byte	ERROR_INVALID_NO_OF_PARAMETERS

### 7.2.6.2.10 Set Volume

**Table 156. Set Volume Command**

Description	This command is used to set the speaker volume.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	GAP_SET_VOLUME	
DataLength	1	

**Table 156. Set Volume Command (continued)**

Description	This command is used to set the speaker volume.	
Data	Volume	Volume control is only supported by the Motorola MC145483 codec.
	1 byte <sup>1</sup>	The following values are defined for MC145483:
		0x00: Mute
		0x01 - 0x1F: Step 1 (MIN)
		0x20 - 0x3F: Step 2
		0x40 - 0x5F: Step 3
		0x60 - 0x7F: Step 4
		0x80 - 0x9F: Step 5
		0xA0 - 0xBF: Step 6
		0xC0 - 0xDF: Step 7
		0xE0 - 0xFF: Step 8 (MAX)
		The following values are defined for MSM7717:
		0x00: Mute
		0x01 - 0xFF: Max volume

**Table 157. Set Volume Confirm**

Description	Response to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_SET_VOLUME	
DataLength	1	
Data	Status	ERROR_OK
	1 byte	ERROR_INVALID_NO_OF_PARAMETERS
		ERROR_SET_VOLUME_FAILED

### 7.2.6.2.11 Get Volume

**Table 158. Get Volume Command**

Description	This command is used to get the speaker volume.
Firmware	0106, 02xx
PacketType	REQ
Opcode	GAP_GET_VOLUME
DataLength	0

**Table 159. Get Volume Confirm**

Description	Response to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_GET_VOLUME	
DataLength	2	
Data	Status	ERROR_OK
	1 byte	ERROR_INVALID_NO_OF_PARAMETERS
	Volume	The actual volume used.
	1 byte	

## 7.2.7 Default Connections

A special feature for link setup is the storage of default connections. The LMX9838 can store up to three connections in its non-volatile data memory. If the “Operation Mode” parameter in NVS is set to automatic, the LMX9838 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 160. Example Default Connection Table**

Index	Local Port	BD_Addr	Remote Port	Transparent
0	1	12 34 56 78 90 12	1	no
1	2	98 76 54 32 10 01	3	no

### 7.2.7.1 Store Default Connection

**Table 161. Store Default Connection Confirm Request**

Description	Stores a default connection in NVS.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SPP_STORE_DEFAULT_CON	
DataLength	10	
Data	Index	Index in the default connection storage. Range 0x00 to 0x06.
	1 byte	
	LocalPort	Local RFCOMM port number. Range 1-30
	1 byte	
	RemotePort	RFCOMM port number on remote device
	1 byte	
	RemoteBdAddress	The BdAddress of the remote device
	6 bytes	
TransparentMode	1 byte	Which transparent mode to enter when the connection is established.
		0x00 Transparent mode off
		0x01 Transparent mode on (only for point-to-point).

**Table 162. Store Default Connection Confirm**

Description	Confirm to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SPP_STORE_DEFAULT_CON	
DataLength	1	
Data	Status	ERROR_OK
	1 byte	ERROR_LIMIT
		ERROR_SPP_INVALID_PORT
		ERROR_SPP_MULTIPLE_TRANSPARENT
		ERROR_SPP_PORT_BUSY

### 7.2.7.2 Connect to Default Connection

**Table 163. Connect to Default Connection Request**

Description	Connects the LMX9838 to a stored connection. Either a single or all connections can be established	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SPP_CONNECT_DEFAULT_CON	
DataLength	1	
Data	Index 1 byte	Index in the default connection storage. Range 0x00 to 0x06. If index is set to 0xFF, all default connections will be established.

**Table 164. Store Default Connection Confirm**

Description	Confirm to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SPP_CONNECT_DEFAULT_CON	
DataLength	1	
Data	Status 1 byte	ERROR_OK ERROR_SPP_AUTOMATIC_CONNECTIONS_PROGRESSING ERROR_LIMIT ERROR_SPP_DEFAULT_CONNECTION_NOT_STORED

### 7.2.7.3 Get List of Default Connections

**Table 165. Get List of Default Connections Request**

Description	Request a list of the default connections stored in NVS	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SPP_GET_LIST_DEFAULT_CON	
DataLength	1	
Data	Index 1 byte	Index in the default connection storage. Range 0x00 to 0x06.

**Table 166. Get List of Default Connections Confirm**

Description	Confirm to the request above.		
Firmware	0106, 02xx		
PacketType	CFM		
Opcode	SPP_GET_LIST_DEFAULT_CON		
DataLength	11		
Data	Status 1 byte	ERROR_OK ERROR_LIMIT	
	Index 1 byte	Index in the default connection storage. Range 0x00 to 0x06.	
	EntryStatus 1 byte	Bitfield: 0000000x	0: Entry is not stored. 1: Entry is stored.

**Table 166. Get List of Default Connections Confirm (continued)**

Description	Confirm to the request above.	
		Bitfield: 000000x0
		0: Command mode. 1: Transparent mode.
		Bitfield: xxxxxx00
		Reserved.
	RemoteBdAddress 6 bytes	The BdAddress of the remote device
	LocalPort 1 byte	Local RFCOMM port number. Range 1-30
	RemotePort 1 byte	RFCOMM port number on remote device

#### 7.2.7.4 Delete Default Connection

**Table 167. Delete Default Connections Request**

Description	Confirm to the request above.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SPP_GET_LIST_DEFAULT_CON	
DataLength	1	
Data	Index 1 byte	Index in the default connection storage. Range 0x00 to 0x06.

**Table 168. Delete Default Connections Confirm**

Description	Confirm to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SPP_DELETE_DEFAULT_CON	
DataLength	2	
Data	Index 1 byte	Index in the default connection storage. Range 0x00 to 0x06.
	Status 1 byte	0x00 OK

### 7.2.8 Bluetooth Low Power Modes

A bluetooth link is based on a physically synchronized connection, which means that the devices can only communicate after successful synchronization. For this, each package also includes some synchronization information. Also a specific polling scheme is in place to keep synchronization if no traffic is necessary.

As the slave has to actively listen to packages from the master, there are different methods to decrease the necessary active receive slots on devices.

The ability to switch to those specific modes is controlled by the Link Policy. To make sure both devices support the low power mode requested, Link Policy can be set first. It will only be successful if both sides support it.

#### 7.2.8.1 Set Default Link Policy

**Table 169. Set Default Link Policy Command**

Description	This command is used to change the default link policy. The default link policy is set during connection setup. The default link policy setting is stored in NVS.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SET_DEFAULT_LINK_POLICY	
DataLength	2	
Data	Link Policy	Bitfield:
	2 byte	0x0001 = Master-slave switch allowed
		0x0002 = Hold mode allowed
		0x0004 = Sniff mode allowed
		0x0008 = Park mode allowed

**Table 170. Set Default Link Policy Confirm**

Description	Response to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SET_DEFAULT_LINK_POLICY	
DataLength	1	
Data	Status	ERROR_OK
	1 byte	ERROR_INVALID_NO_OF_PARAMETERS
		ERROR_ILLEGAL_LINK_POLICY

### 7.2.8.2 Get Default Link Policy

**Table 171. Get Default Link Policy Command**

Description	This command is used to read the default link policy setting from NVS.
Firmware	0106, 02xx
PacketType	REQ
Opcode	GET_DEFAULT_LINK_POLICY
DataLength	0

**Table 172. Get Default Link Policy Confirm**

Description	Response to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GET_DEFAULT_LINK_POLICY	
DataLength	2	
Data	LinkPolicy	Bitfield:
	2 byte	0x0001 = Master-slave switch allowed
		0x0002 = Hold mode allowed
		0x0004 = Sniff mode allowed
		0x0008 = Park mode allowed

### 7.2.8.3 Set Link Policy

**Table 173. Set Link Policy Request**

Description	This command is used to change the current link policy setting for the given link.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	GAP_SET_LINK_POLICY	
DataLength	8	
Data	BdAddr	The Bluetooth address of the remote device of which to change the link policy settings for the link.
	6 bytes	
LinkPolicy	2 byte	Bitfield:
		0x0001 = Master-slave switch allowed
		0x0002 = Hold mode allowed
		0x0004 = Sniff mode allowed
		0x0008 = Park mode allowed

**Table 174. Set Link Policy Confirm**

Description	Response to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_SET_LINK_POLICY	
DataLength	1	
Data	Status	ERROR_OK
	1 byte	ERROR_UNSPECIFIED_ERROR
		ERROR_INVALID_NO_OF_PARAMETERS
		ERROR_NO_CONNECTION
		ERROR_ILLEGAL_LINK_POLICY

#### 7.2.8.4 Get Link Policy

**Table 175. Get Link Policy Request**

Description	This command is used to get the current link policy setting for the given link.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	GAP_GET_LINK_POLICY	
DataLength	6	
Data	BdAddr	The Bluetooth address of the remote device of which to get the current link policy settings for the link.
	6 bytes	

**Table 176. Get Link Policy Confirm**

Description	Response to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_GET_LINK_POLICY	
DataLength	3	
Data	Status	ERROR_OK
	1 byte	ERROR_UNSPECIFIED_ERROR
		ERROR_INVALID_NO_OF_PARAMETERS
		ERROR_NO_CONNECTION

**Table 176. Get Link Policy Confirm (continued)**

Description	Response to the request above.	
	LinkPolicy	Bitfield:
	2 byte	0x0001 = Master-slave switch allowed
		0x0002 = Hold mode allowed
		0x0004 = Sniff mode allowed
		0x0008 = Park mode allowed

### 7.2.8.5 Enter Sniff Mode

Command to enter the sniff mode. The command includes the maximum and minimum value for the sniff interval. After sending the command, Master and slave will calculate a reasonable sniff time and will switch into Sniff mode.

**Table 177. Enter Sniff Mode Request**

Description	This command is used to request sniff mode on a given link with user specified parameters.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	GAP_ENTER_SNIFF_MODE	
DataLength	14	
Data	BdAddr	The Bluetooth address of the remote device of which to put the link in sniff mode.
	6 bytes	
	SniffMaxInterval	Maximum sniff interval in slots.
	2 bytes	
	SniffMinInterval	Minimum sniff interval in slots
	2 bytes	
	SniffAttempt	Number of slots the slave must listen, beginning at the sniff slot, even if it does not receive a packet with its own AM.
	2 bytes	
	SniffTimeout	Number of additional slots the slave must listen if it continues to receive only packets with its own AM address.
	2 bytes	

**Table 178. Enter Sniff Mode Confirm**

Description	Response to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_ENTER_SNIFF_MODE	
DataLength	1	
Data	Status	ERROR_OK
	1 byte	ERROR_UNSPECIFIED_ERROR
		ERROR_INVALID_NO_OF_PARAMETERS
		ERROR_NO_CONNECTION

### 7.2.8.6 Exit Sniff Mode

**Table 179. Exit Sniff Mode Request**

Description	This command is used to exit a current sniff mode on a given link.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	GAP_EXIT_SNIFF_MODE	
DataLength	6	
Data	BdAddr	The Bluetooth address of the remote device of which to exit the current sniff mode.
	6 bytes	

**Table 180. Exit Sniff Mode Confirm**

Description	Response to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_EXIT_SNIFF_MODE	
DataLength	1	
Data	Status	ERROR_OK
	1 byte	ERROR_UNSPECIFIED_ERROR
		ERROR_INVALID_NO_OF_PARAMETERS
		ERROR_NO_CONNECTION

### 7.2.8.7 Enter Park Mode

In Park Mode the slave will lose its active member address and will not longer be part of the piconet. It will be kept synchronized by beacons within the specified interval range.

**Table 181. Enter Park Mode Request**

Description	This command is used to request park mode on a given link with user specified parameters.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	GAP_ENTER_PARK_MODE	
DataLength	10	
Data	BdAddr	The Bluetooth address of the remote device of which to put the link in park mode.
	6 bytes	
	BeaconMaxInterval	Acceptable longest length of the interval between beacons.
	2 bytes	
	BeaconMinInterval	Acceptable shortest length of the interval between beacons.
	2 bytes	

**Table 182. Enter Park Mode Confirm**

Description	Response to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_ENTER_PARK_MODE	
DataLength	1	
Data	Status	ERROR_OK
	1 byte	ERROR_UNSPECIFIED_ERROR
		ERROR_INVALID_NO_OF_PARAMETERS
		ERROR_NO_CONNECTION

### 7.2.8.8 Exit Park Mode

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

**Table 183. Exit Park Mode Request**

Description	This command is used to exit a current park mode on a given link.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	GAP_EXIT_PARK_MODE	
DataLength	6	
Data	BdAddr	The Bluetooth address of the remote device of which to exit the current park mode.
	6 bytes	

**Table 184. Enter Park Mode Confirm**

Description	Response to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_ENTER_PARK_MODE	
DataLength	1	
Data	Status	ERROR_OK
	1 byte	ERROR_UNSPECIFIED_ERROR
		ERROR_INVALID_NO_OF_PARAMETERS
		ERROR_NO_CONNECTION

### 7.2.8.9 Enter Hold Mode

**Table 185. Enter Hold Mode Request**

Description	This command is used to request Hold mode on a given link with user specified parameters.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	GAP_ENTER_HOLD_MODE	
DataLength	10	
Data	BdAddr	The Bluetooth address of the remote device of which to put the link in Hold mode.
	6 bytes	
	HoldMaxInterval	Maximum length of the Hold interval for which the Host may actually enter into the hold mode after negotiation with the remote device.
	2 bytes	
	HoldMinInterval	minimum length of the Hold interval for which the Host may actually enter into the hold mode after the negotiation with the remote device.
	2 bytes	

**Table 186. Enter Hold Mode Confirm**

Description	Response to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_ENTER_HOLD_MODE	
DataLength	1	

**Table 186. Enter Hold Mode Confirm (continued)**

Description	Response to the request above.	
Data	Status	ERROR_OK
	1 byte	ERROR_UNSPECIFIED_ERROR
		ERROR_INVALID_NO_OF_PARAMETERS
		ERROR_NO_CONNECTION

### 7.2.8.10 Power Save Mode Changed

In case the remote device changed the Power Mode for that link, this event will be returned by the LMX9838.

**Table 187. Power Save Mode Changed Indicator**

Description	This indication is sent to the host when changes the power save mode on a link occur.	
Firmware	0106, 02xx	
PacketType	IND	
Opcode	GAP_POWER_SAVE_MODE_CHANGED	
DataLength	8	
Data	Status	ERROR_OK
	1 byte	ERROR_ATTEMPT_FAILED
		ERROR_UNSPECIFIED_ERROR
	BdAddr	The Bluetooth address of the remote device for which the power save mode has changed on the link.
	6 bytes	
	Mode	0x00 = Active mode (Left power save mode)
	1 byte	0x01 = Hold mode (Hold mode entered)
		0x02 = Sniff mode (Sniff mode entered)
		0x03 = Park mode (Park mode entered)

### 7.2.9 Wake-Up Functionality

Wake-up functionality allows to stop the communication between host and LMX9838 in order to save power on both devices. The wake-up itself is done via the RTS/CTS pins or the Host\_WU pin. For a detailed description of the hardware wake-up please see also [Section 5.4.1.3](#).

The communication between host and LMX9838 can be stopped by sending the LMX9838 the Disable Transport Layer command. The command can be sent any time. The LMX9838 will continue its current operation status and try to wake up the host as soon as the interface is needed.

#### 7.2.9.1 Disable Transport Layer

**Table 188. Disable Transport Layer Command**

Description	This Command disables the transport layer and thereby allowing power-saving.
Firmware	0106, 02xx
PacketType	REQ
Opcode	DISABLE_TL
DataLength	0

**Table 189. Disable Transport Layer Confirm**

Description	Response to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	DISABLE_TL	
DataLength	1	
Data	Status	ERROR_OK
	1 byte	ERROR_COMMAND_DISALLOWED

### 7.2.9.2 Transport Layer Enabled Indicator

**Table 190. Transport Layer enabled indicator**

Description	This indication is sent when the transport layer is re-enabled. Transport layer is re-enabled by pulling RTS signal of the host or an event is ready to be sent from the Simply Blue to the host, normally due to an air-interface event.	
Firmware	0106, 02xx	
PacketType	IND	
Opcode	TL_ENABLED	
DataLength	1	
Data	Status	ERROR_OK
	1 byte	

### 7.2.10 SPP Port Configuration

An active SPP link appears as a virtual serial port connection between two devices. As any other serial connection it has different settings for that “virtual” serial port. The following settings enable the host to change specific port settings on that virtual port. The reference for all commands is the local RFCOMM port the link has been set up with.

All commands in this section are only handled at application level and have no direct impact on the bluetooth link.

#### 7.2.10.1 SPP Set Port Configuration

**Table 191. SPP Set Port Config Request**

Description	Write the configuration for the SPP port. These are virtual settings for the air connection, not the settings for the LMX9838 UART. Note: The baudrate in this configuration has no impact on the throughput on the bluetooth link itself or the LMX9838. It is not used by the RFCOMM layer.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SPP_SET_PORT_CONFIG	
DataLength	6	
Data	LocalPort	Local RFCOMM port number. Range 1-30
	1 byte	

**Table 191. SPP Set Port Config Request (continued)**

Description	Write the configuration for the SPP port. These are virtual settings for the air connection, not the settings for the LMX9838 UART. Note: The baudrate in this configuration has no impact on the throughput on the bluetooth link itself or the LMX9838. It is not used by the RFCOMM layer.			
BaudRate 1 byte	BaudRate	Baudrate		
	1 byte	0x00 = 2400 baud 0x01 = 4800 baud 0x02 = 7200 baud		
		0x03 = 9600 baud 0x04 = 19200 baud 0x05 = 38400 baud		
		0x06 = 57600 baud 0x07 = 115200 baud 0x08 = 230400 baud		
PortSettings 1 byte	PortSettings	Bitfield: 000000XX	Number of databits	00=5 bits
	1 byte			01=6 bits
				02=7 bits
				03=8 bits
		Bitfield: 00000X00	Number of stopbits	0 = 1
				1 = 1.5
		Bitfield: 0000X000	Parity	0 = No parity
				1 = Parity
		Bitfield: 00XX0000	ParityType	00=ODD
				01=EVEN
				02=MARK
				03=SPACE
		Bitfield: XX000000	Reserved	
FlowControl 1 byte	FlowControl	Bitfield: 0000000X	XonXoffOnInput	0=Disable
	1 byte			1=Enable
		Bitfield: 000000X0	XonXoffOnOutput	0=Disable
				1=Enable
		Bitfield: 00000X00	RtrOnInput	0=Disable
				1=Enable
		Bitfield: 0000X000	RtrOnOutput	0=Disable
				1=Enable
		Bitfield: 000X0000	RtcOnInput	0=Disable
				1=Enable
		Bitfield: 00X00000	RtcOnOutput	0=Disable
				1=Enable
		Bitfield: XX000000	Reserved	
XonChar 1 byte	XonChar	Char used for Xon, if Xon/Xoff flowcontrol is used.		
	1 byte			
XoffChar 1 byte	XoffChar	Char used for Xoff, if Xon/Xoff flowcontrol is used.		
	1 byte			

**Table 192. SPP Set Port Configuration Confirm**

Description	Confirm to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SPP_SET_PORT_CONFIG	
DataLength	2	
Data	Status	ERROR_OK
	1 byte	ERROR_UNSPECIFIED_ERROR
		ERROR_UNSPECIFIED_ERROR
		ERROR_SPP_INVALID_PORT
		ERROR_SPP_PORT_NOT_OPEN
		ERROR_UART_SPEED_OUT_OF_RANGE
	LocalPort	Local RFCOMM port number. Range 1-30
	1 byte	

### 7.2.10.2 SPP Get Port Configuration

**Table 193. 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 LMX9838 serial port.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SPP_GET_PORT_CONFIG	
DataLength	1	
Data	LocalPort	Local RFCOMM port number. Range 1-30
	1 byte	

**Table 194. SPP Get Port Configuration Confirm**

Description	Confirm to the request above.		
Firmware	0106, 02xx		
PacketType	CFM		
Opcode	SPP_GET_PORT_CONFIG		
DataLength	7		
Data	Status	ERROR_OK	
	1 byte	ERROR_UNSPECIFIED_ERROR	
		ERROR_SPP_INVALID_PORT	
		ERROR_SPP_PORT_NOT_OPEN	
	LocalPort	Local RFCOMM port number. Range 1-30	
	1 byte		
	BaudRate	Baudrate	
	1 byte	0x00 = 2400 baud 0x01 = 4800 baud 0x02 = 7200 baud	
		0x03 = 9600 baud 0x04 = 19200 baud 0x05 = 38400 baud	
		0x06 = 57600 baud 0x07 = 115200 baud 0x08 = 230400 baud	
	Portsettings	Bitfield: 000000XX	Number of databits
	1 byte		00=5 bits
			01=6 bits
			02=7 bits
			03=8 bits

**Table 194. SPP Get Port Configuration Confirm (continued)**

Description	Confirm to the request above.			
		Bitfield: 00000X00	Number of stopbits	0 = 1 1 = 1.5
		Bitfield: 0000X000	Parity	0 = No parity 1 = Parity
		Bitfield: 00XX0000	ParityType	00=ODD 01=EVEN 02=MARK 03=SPACE
		Bitfield: XX000000	Reserved	
	FlowControl 1 byte	Bitfield: 0000000X	XonXoffOnInput	0=Disable 1=Enable
		Bitfield: 000000X0	XonXoffOnOutput	0=Disable 1=Enable
		Bitfield: 00000X00	RtrOnInput	0=Disable 1=Enable
		Bitfield: 0000X000	RtrOnOutput	0=Disable 1=Enable
		Bitfield: 000X0000	RtcOnInput	0=Disable 1=Enable
		Bitfield: 00X00000	RtcOnOutput	0=Disable 1=Enable
		Bitfield: XX000000	Reserved	
	XonChar 1 byte	Char used for Xon, if Xon/Xoff flowcontrol is used.		
	XoffChar 1 byte	Char used for Xoff, if Xon/Xoff flowcontrol is used.		

### 7.2.10.3 SPP Port Configuration Changed Indicator

**Table 195. SPP Port Configuration Changed Indicator**

Description	Sent from the LMX9838 when remote device has changed the port configuration.			
Firmware	0106, 02xx			
PacketType	IND			
Opcode	SPP_PORT_CONFIG_CHANGED			
DataLength	6			
Data	LocalPort 1 byte	Local RFCOMM port number. Range 1-30		
	BaudRate 1 byte	Baudrate 0x00 = 2400 baud 0x01 = 4800 baud 0x02 = 7200 baud 0x03 = 9600 baud 0x04 = 19200 baud 0x05 = 38400 baud 0x06 = 57600 baud 0x07 = 115200 baud 0x08 = 230400 baud		
	Portsettings 1 byte	Bitfield: 000000XX	Number of databits	00=5 bits 01=6 bits 02=7 bits 03=8 bits

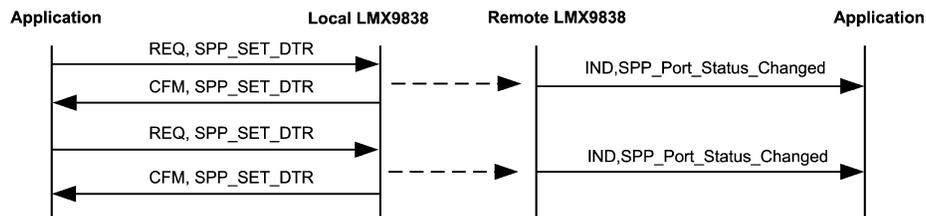
**Table 195. SPP Port Configuration Changed Indicator (continued)**

Description	Sent from the LMX9838 when remote device has changed the port configuration.			
		Bitfield: 00000X00	Number of stopbits	0 = 1 1 = 1.5
		Bitfield: 0000X000	Parity	0 = No parity 1 = Parity
		Bitfield: 00XX0000	ParityType	00=ODD 01=EVEN 02=MARK 03=SPACE
		Bitfield: XX000000	Reserved	
	FlowControl 1 byte	Bitfield: 0000000X	XonXoffOnInput	0=Disable 1=Enable
		Bitfield: 000000X0	XonXoffOnOutput	0=Disable 1=Enable
		Bitfield: 00000X00	RtrOnInput	0=Disable 1=Enable
		Bitfield: 0000X000	RtrOnOutput	0=Disable 1=Enable
		Bitfield: 000X0000	RtcOnInput	0=Disable 1=Enable
		Bitfield: 00X00000	RtcOnOutput	0=Disable 1=Enable
		Bitfield: XX000000	Reserved	
	XonChar 1 byte	Char used for Xon, if Xon/Xoff flowcontrol is used.		
	XoffChar 1 byte	Char used for Xoff, if Xon/Xoff flowcontrol is used.		

**7.2.11 SPP Port Status**

An active SPP link allows signalling of modem status and line status over the bluetooth link. The following commands and events describe how to change or get the status of those line parameters.

NOTE: This handshaking events have no impact on the LMX9838 flow control. These messages can only used by the host application on each side to either react on the flow themselves.



**Figure 81. Set SPP DTR**

**7.2.11.1 SPP Get Port Status**

<b>Description</b>	Get the current state of the modem status and line status at application level. This command resets the value (to 0) of the following members of the port status: <b>OverrunError</b> <b>ParityError</b> <b>FramingError</b> <b>BreakLength</b> The value of DSR and CTS are only changed when new values are received from the remote device!	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SPP_GET_PORT_STATUS	
DataLength	1	
Data	LocalPort	Local RFCOMM port number. Range 1-30
	1 byte	

**Table 196. SPP Get Port Status Confirm**

<b>Description</b>	<b>Confirm to the request above.</b>			
Firmware	0106, 02xx			
PacketType	CFM			
Opcode	SPP_GET_PORT_STATUS			
DataLength	5			
Data	Status	ERROR_OK		
	1 byte	ERROR_UNSPECIFIED_ERROR		
		ERROR_SPP_INVALID_PORT		
		ERROR_SPP_PORT_NOT_OPEN		
	LocalPort	Local RFCOMM port number. Range 1-30		
	1 byte			
	PortStatus	Bitfield: 0000000X	DTR	0 = Low
	1 byte			1 = High
		Bitfield: 000000X0	RTS	0 = Low
				1 = High
		Bitfield: 00000X00	DSR	0 = Low
				1 = High
		Bitfield: 0000X000	CTS	0 = Low
				1 = High
		Bitfield: 000X0000	Overrun Error	0 = No Error
				1 = Overrun Error
		Bitfield: 00X00000	Parity Error	0 = No Error
				1 = Parity Error
		Bitfield: 0X000000	Framing Error	0 = No Error
				1 = Framing Error
		Bitfield:X0000000	DLC established	0 = No DLC
				1 = DLC is available
	Break Length	The length in ms of the detected break. The value 0 is used to indicate that no break has been detected.		
	2 bytes			

### 7.2.11.2 SPP Port Set DTR

**Table 197. SPP Port Set DTR Request**

<b>Description</b>	<b>This command sets the state of the DTR. Since RFCOMM acts as a null modem where DTR and DSR are connected, the remote device will see this as a change of the state of the DSR signal. In case the remote device is a LMX9838, this remote LMX9838 will send a "SPP Port Status Changed Indicator" ( <a href="#">Section 7.2.11.8</a> ) to his application.</b>	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SPP_PORT_SET_DTR	
DataLength	2	
Data	LocalPort	Local RFCOMM port number. Range 1-30
	1 byte	
	State	0: False.
	1 byte	1: True.

**Table 198. SPP Port Set DTR Confirm**

<b>Description</b>	<b>Confirm to the request above.</b>	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SPP_PORT_SET_DTR	
DataLength	2	
Data	Status	ERROR_OK
	1 byte	ERROR_ILLEGAL_STATE_VALUE
		ERROR_SPP_INVALID_PORT
		ERROR_SPP_PORT_NOT_OPEN
	LocalPort	Local RFCOMM port number. Range 1-30
	1 byte	

### 7.2.11.3 SPP Port Set RTS

**Table 199. SPP Port Set RTS Request**

<b>Description</b>	<b>This command sets the state of the RTS bit. Since RFCOMM acts as a null modem where RTS and CTS are connected, the remote device will see this as a change of the state of the CTS signal. In case the remote device is a LMX9838, this remote LMX9838 will send a "SPP Port Status Changed Indicator" ( <a href="#">Section 7.2.11.8</a> ) to his application.</b>	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SPP_PORT_SET_RTS	
DataLength	2	
Data	LocalPort	Local RFCOMM port number. Range 1-30
	1 byte	
	State	0: False.
	1 byte	1: True.

**Table 200. SPP Port Set RTS Confirm**

<b>Description</b>	<b>Confirm to the request above.</b>	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SPP_PORT_SET_RTS	
DataLength	2	

**Table 200. SPP Port Set RTS Confirm (continued)**

Description	Confirm to the request above.	
Data	Status	ERROR_OK
	1 byte	ERROR_ILLEGAL_STATE_VALUE
		ERROR_SPP_INVALID_PORT
		ERROR_SPP_PORT_NOT_OPEN
	LocalPort	Local RFCOMM port number. Range 1-30
	1 byte	

#### 7.2.11.4 SPP Port Set BREAK

**Table 201. SPP Set Port Break Request**

Description	This command indicates a UART BREAK over the virtual serial link. In case the remote device is a LMX9838, this remote LMX9838 will send a "SPP Port Status Changed Indicator" ( <a href="#">Section 7.2.11.8</a> ) to his application.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SPP_PORT_BREAK	
DataLength	3	
Data	LocalPort	Local RFCOMM port number. Range 1-30
	1 byte	
	BreakLength	The length of the break in ms.
	2 bytes	

**Table 202. SPP Set Port Break Confirm**

Description	Confirm to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SPP_PORT_BREAK	
DataLength	2	
Data	Status	ERROR_OK
	1 byte	ERROR_SPP_INVALID_PORT
		ERROR_SPP_PORT_NOT_OPEN
	LocalPort	Local RFCOMM port number. Range 1-30
	1 byte	

#### 7.2.11.5 Set Overrun Error

**Table 203. SPP Set Overrun Error Request**

Description	This command is used to indicate that the host has detected an overrun error. In case the remote device is a LMX9838, this remote LMX9838 will send a "SPP Port Status Changed Indicator" ( <a href="#">Section 7.2.11.8</a> ) to his application.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SPP_PORT_OVERRUN_ERROR	
DataLength	1	
Data	LocalPort	Local RFCOMM port number. Range 1-30
	1 byte	

**Table 204. SPP Port Overrun Error Confirm**

Description	Confirm to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SPP_PORT_OVERRUN_ERROR	
DataLength	2	
Data	Status	ERROR_OK
	1 byte	ERROR_SPP_INVALID_PORT
		ERROR_SPP_PORT_NOT_OPEN
	LocalPort	Local RFCOMM port number. Range 1-30
	1 byte	

### 7.2.11.6 SPP Set Parity Error

**Table 205. SPP Port Parity Error Request**

Description	This command is used to indicate that the host has detected a parity error. In case the remote device is a LMX9838, this remote LMX9838 will send a "SPP Port Status Changed Indicator" ( <a href="#">Section 7.2.11.8</a> ) to his application.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SPP_PORT_PARITY_ERROR	
DataLength	1	
Data	LocalPort	Local RFCOMM port number. Range 1-30
	1 byte	

**Table 206. SPP Port Parity Error Confirm**

Description	Confirm to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SPP_PORT_PARITY_ERROR	
DataLength	2	
Data	Status	ERROR_OK
	1 byte	ERROR_SPP_INVALID_PORT
		ERROR_SPP_PORT_NOT_OPEN
	LocalPort	Local RFCOMM port number. Range 1-30
	1 byte	

### 7.2.11.7 SPP Set Framing Error

**Table 207. SPP Port Framing Error Request**

Description	This command is used to indicate that the host has detected a framing error. In case the remote device is a LMX9838, this remote LMX9838 will send a "SPP Port Status Changed Indicator" ( <a href="#">Section 7.2.11.8</a> ) to his application.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SPP_PORT_FRAMING_ERROR	
DataLength	1	
Data	LocalPort	Local RFCOMM port number. Range 1-30
	1 byte	

**Table 208. SPP Port Framing Errors Confirm**

Description	Confirm to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SPP_PORT_FRAMING_ERROR	
DataLength	2	
Data	Status	ERROR_OK
	1 byte	ERROR_SPP_INVALID_PORT
		ERROR_SPP_PORT_NOT_OPEN
LocalPort	LocalPort	Local RFCOMM port number. Range 1-30
	1 byte	

### 7.2.11.8 SPP Port Status Changed Indicator

**Table 209. SPP Port Status Changed Indicator**

Description	Send from the LMX9838 when remote device has changed the port.			
Firmware	0106, 02xx			
PacketType	IND			
Opcode	SPP_PORT_STATUS_CHANGED			
DataLength	4			
Data	LocalPort	Local RFCOMM port number. Range 1-30		
	1 byte			
	PortStatus	Bitfield: 0000000X	DTR	0 = Low
	1 byte			1 = High
		Bitfield: 000000X0	RTS	0 = Low
				1 = High
		Bitfield: 00000X00	DSR	0 = Low
				1 = High
		Bitfield: 0000X000	CTS	0 = Low
				1 = High
		Bitfield: 000X0000	Overrun Error	0 = No Error
				1 = Overrun Error
		Bitfield: 00X00000	Parity Error	0 = No Error
				1 = Parity Error
		Bitfield: 0X000000	Framing Error	0 = No Error
				1 = Framing Error
		Bitfield: X0000000	DLC established	0 = No DLC
				1 = DLC is available
Break Length	Break Length	The length in ms of the detected break. The value 0 is used to indicate that no break has been detected.		
	2 bytes			

### 7.2.12 RFComm Channels to Open

Each Service within the Service Database is registered to a specific RFComm channel. The configuration, which ports will be opened and initialized can be configured with the command "Set Ports To Open" ([Section 7.2.12.1](#)).

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

### 7.2.12.1 Set Ports To Open

**Table 210. SDP Set Ports to Open Request**

Description	This command will change which RFCOMM ports the LMX9838 initializes. The value gets effective immediately and is stored within the NVS to be restored after reset.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	SET_PORTS_TO_OPEN	
DataLength	4	
Data	PORTS	This field is a 32-bit mask indicating which RFCOMM ports the LMX9838 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.
	4 Bytes	

**Table 211. SDP Set Ports to Open Confirm**

Description	Confirms the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	PORTS_TO_OPEN	
DataLength	1	
Data	Status	ERROR_OK
	1 byte	ERROR_INVALID_PORT
		ERROR_INVALID_NO_OF_PARAMETERS

### 7.2.12.2 Get Ports To Open

**Table 212. SDP Get Ports to Open Request**

Description	This command reads back the currently opened RFCOMM ports.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	GET_PORTS_TO_OPEN	
DataLength	0	

**Table 213. SDP Get Ports to Open Confirm**

Description	Confirms the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	PORTS_TO_OPEN	
DataLength	5	
Data	Status	ERROR_OK
	1 byte	ERROR_INVALID_PORT
	PORTS 4 Bytes	This field is a 32-bit mask indicating which RFCOMM ports the LMX9838 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.

## 7.2.13 Local Service Database Configuration

The LMX9838 allows the modification of the Local Service Discovery Database. By default, the service database contains one entry configured at RFCComm port 1 for a Serial Port Profile, with Authentication and Encryption enabled.

If the application needs to offer a second service to another device, this service has to be entered within the service database and, if required, the RFCComm port has to be opened (see [Section 7.2.12.1](#)). The service entry in general includes information about the name of the service, which appears on a remote device after browsing, the port number and security settings.

Each registered service entry itself can be enabled or disabled. This allows to have different services registered to one specific RFCComm port enabling and disabling them by needs.

The command set includes just one command to store any SDP record available within the Service Database, called "Store Generic SDP Record" ([Section 7.2.13.1](#)). Each record includes the profile specific settings needed.

The service records are stored in a database like system within the NVS, so they are still available after reset. The storage of an entry is confirmed by the LMX9838 with a specific record identifier. This identifier is needed for the Enabling or Disabling command to address those specific entries.

### 7.2.13.1 Store Generic SDP Record

**Table 214. Store Generic SDP Record Request**

Description	This command will create a new service record in the local SDP database, store in the NVS. The Simply Blue Commander offers the ability to generate service records with all correct parameters automatically. See also <a href="#">Section 6.4.2.2</a> for an example.	
PacketType	REQ	
Opcode	STORE_SDP_RECORD	
DataLength	5 + <SdpRecordLength>	
Data	LocalPort	The local RFCOMM port used by this service. Range 1 to 30.
	1 byte	
	Authentication	0x00 No authentication requirements.
	1 byte	0x02 Authentication is only required for this profile for incoming connections.
		0x20 Authentication is only required for this profile for outgoing connections.
		0x22 Authentication is required for this profile for connections in both directions
	Encryption	0x00 No encryption requirements
	1 byte	0x04 Encryption is only required for this profile for incoming connections.
		0x40 Encryption is only required for this profile for outgoing connections.
		0x44 Encryption is required for this profile for connections in both directions
	SdpRecordLength	The number of bytes in the SDP record.
	2 bytes	
	SdpRecord	The SDP record data formatted as: < uint16:Attributeld, SdpDataElement:AttributeData, uint16:Attributeld, SdpDataElement:AttributeData, >. The maximum length for each service is 255 bytes.
	<SdpRecordLength> Bytes	

**Table 215. Store Generic SDP Record Confirm**

Description	Confirms the request above	
PacketType	CFM	
Opcode	STORE_SDP_RECORD	
DataLength	2	
Data	Status	ERROR_OK
	1 byte	ERROR_INVALID_SDP_RECORD
		ERROR_MAXIMUM_NO_OF_SERVICE_RECORDS_REACHED
		ERROR_WRITING_TO_NVS
	Identifier	A unique identifier, which must be used when dynamic enabling/disabling the record.
	1 byte	

### 7.2.13.2 SDP Enable SDP Record

**Table 216. SDP Enable SDP Record Request**

Description	This command is used to enable/disable stored SDP records in the LMX9838.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	ENABLE_SDP_RECORD	
DataLength	2	
Data	State	The new state of the SDP record.
	1 byte	0x00 Disable the record.
		0x01 Enable the record.
	Identifier	The identifier of the service record to address. This will be received when the record was stored in the LMX9838 by "Store Generic SDP Record" (7.2.13.1).
	1 byte	

**Table 217. SDP Enable SDP Record Confirm**

Description	Confirms the request above	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	ENABLE_SDP_RECORD	
DataLength	2	
Data	Status	ERROR_OK
	1 byte	ERROR_ILLEGAL_STATE_VALUE
		ERROR_IDENTIFIER_OUT_OF_RANGE
		ERROR_RECORD_ALREADY_IN_SELECTED_STATE
		ERROR_IDENTIFIER_NOT_IN_USE
		ERROR_INVALID_NO_OF_PARAMETERS
	Identifier	The identifier received when the record was stored in the LMX9838.
	1 byte	

### 7.2.13.3 SDP Delete All SDP Records

**Table 218. SDP Delete SDP Record Request**

Description	This command is used to delete all stored SDP records in the LMX9838.
Firmware	0106, 02xx
PacketType	REQ
Opcode	DELETE_SDP_RECORDS
DataLength	0
Data	None

**Table 219. SDP Delete SDP Record Confirm**

Description	Confirms the request above	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	DELETE_SDP_RECORDS	
DataLength	1	
Data	Status	ERROR_OK
	1 byte	ERROR_INVALID_NO_OF_PARAMETERS

## 7.2.14 Local Bluetooth Settings

The LMX9838 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.

### 7.2.14.1 Read Local Name

**Table 220. Read Local Name Request**

Description	Request the user-friendly name for the local Bluetooth device.
Firmware	0106, 02xx
PacketType	REQ
Opcode	GAP_READ_LOCAL_NAME
DataLength	0
Data	None

**Table 221. Read Local Name Confirm**

Description	Confirm to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_READ_LOCAL_NAME	
DataLength	2 + NameLength	
Data	Status	ERROR_OK
	1 byte	ERROR_INVALID_NO_OF_PARAMETERS
		ERROR_UNKNOWN_ERROR
	NameLength	Number of bytes in device name
	1 byte	
	DeviceName	The user-friendly name of the local device.
	NameLength bytes	The string is NULL terminated. Max length is 40 bytes.

### 7.2.14.2 Write Local Name

**Table 222. Write Local Name Request**

Description	Change the user-friendly name for the local Bluetooth device. The name is stored in NVS	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	GAP_WRITE_LOCAL_NAME	
DataLength	1+ NameLength	
Data	NameLength	Number of bytes in device name
	1 byte	
	DeviceName	The user-friendly name of the local device.
	Length bytes	(String must be NULL terminated). Max length is 40 bytes.

**Table 223. Write Local Name Confirm**

Description	Confirm to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_WRITE_LOCAL_NAME	
DataLength	1	
Data	Status	ERROR_OK
	1 byte	ERROR_NAME_TOO_LONG
		ERROR_INVALID_NO_OF_PARAMETERS

### 7.2.14.3 Read Local Bluetooth Address

**Table 224. Read Local BD\_Addr Request**

Description	Read the Bluetooth device address of the local Bluetooth device.
Firmware	0106, 02xx
PacketType	REQ
Opcode	GAP_READ_LOCAL_BDA
DataLength	0
Data	None

**Table 225. Read Local BD\_Addr Confirm**

Description	Confirm to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_READ_LOCAL_BDA	
DataLength	7	
Data	Status	ERROR_OK
	1 byte	ERROR_INVALID_NO_OF_PARAMETERS
		ERROR_UNKNOWN_ERROR
	BdAddr	Bluetooth device address for the local device
	6 byte	

### 7.2.14.4 Change Local Bluetooth Address

The Bluetooth Device Address (BD\_Addr) is a key parameter for Bluetooth operation. Officially assigned BD addresses are preprogrammed by Texas Instruments. The BD\_Addr should therefore never be changed.

**Table 226. Change Local BD\_Address Request**

<b>Description</b>	<b>The LMX9838 will store the local BdAddress within NVS. Changing the BdAddress requires a HW Reset or "Reset" ( Section 7.2.18.3 ). If the BdAddress has the value 0xFFFFFFFF, the LMX9838 will stay in "Initialization if EEPROM empty" after reset or power-up.</b>	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	CHANGE_LOCAL_BDADDRESS	
DataLength	6	
Data	BdAddress 6 Bytes	The new Bluetooth address for the local device. The address is stored in NVS.

**Table 227. Change Local BD\_Addr Confirm**

<b>Description</b>	<b>Confirms the request above</b>	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	CHANGE_LOCAL_BDADDRESS	
DataLength	1	
Data	Status	ERROR_OK
	1 byte	ERROR_RESET_TO_TI_BDADDRESS
		ERROR_INVALID_NO_OF_PARAMETERS

### 7.2.14.5 Store Class of Device

**Table 228. Store Class of Device Request**

<b>Description</b>	<b>This command will store the class of device for the LMX9838 in NVS. The proper value for the class of device parameter is specified by the Bluetooth SIG.</b>	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	STORE_CLASS_OF_DEVICE	
DataLength	3	
Data	ClassOfDevice 3 bytes	The class of device value to be stored in NVS.

**Table 229. Store Class of Device Confirm**

<b>Description</b>	<b>Confirms the request above</b>	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	STORE_CLASS_OF_DEVICE	
DataLength	1	
Data	Status	ERROR_OK
	1 byte	ERROR_INVALID_NO_OF_PARAMETERS

7.2.14.6 Set Scan Mode - Discoverability / Connectability

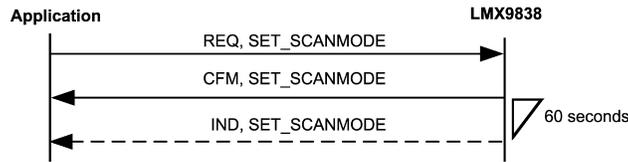


Figure 82. Set Scan Mode

Table 230. Set Scan Mode Request

Description	<p>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.                  Change the Bluetooth scan mode.                  Automatic limited discoverable mode automatically toggles between general and limited inquiry scanning. This mode defined by the Bluetooth GAP profile specification, refer to part K.1, section 6.2.1. The automatic Limited discoverable mode times out after 60 sec. At this point the LMX9838 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.</p>	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	GAP_SET_SCANMODE	
DataLength	2	
Data	Connectability	0x00 = Not connectable
	1 byte	0x01 = Connectable 0x81 = Connectable using Interlaced Scanning
Discoverability	1 byte	0x00 = Non discoverable
		0x01 = General discoverable
		0x81 = General discoverable using Interlaced Scanning
		0x02 = Limited discoverable
		0x82 = Limited discoverable using Interlaced Scanning
		0x03 = Automatic limited discoverable mode, see
	0x83 = Automatic limited discoverable mode using Interlaced Scanning	

Table 231. Set Scan Mode Confirm

Description	Confirm to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_SET_SCANMODE	
DataLength	1	
Data	Status	ERROR_OK
	1 byte	ERROR_INVALID_CONNECTABILITY_PARAMETER
		ERROR_INVALID_DISCOVERABILITY_PARAMETER
		ERROR_INVALID_NO_OF_PARAMETERS
		ERROR_UNKNOWN_ERROR

7.2.14.7 Automatic Limited Discoverable Mode Ended

**Table 232. Automatic Limited Discoverable Ended Indicator**

Description	Indication send from the device, when automatic limited discoverable mode has ended.	
Firmware	0106, 02xx	
PacketType	IND	
Opcode	GAP_SET_SCANMODE	
DataLength	1	
Data	Status	ERROR_OK
	1 byte	ERROR_UNKNOWN_ERROR

### 7.2.14.8 Force Master Role

The Force Master Role is initiated only after boot up or reset. The role is stored as parameter within NVS.

If Force Master is activated the LMX9838 tries to switch its role to master if connected from another device. For this the LMX9838 is sending a Master/Slave switch request after link establishment to the remote device. If the switch is successful, the link setup will be continued and the LMX9838 is open for other incoming connections.

If the switch fails, the link will be dropped. This functionality allows Access Point like applications on the LMX9838.

**Table 233. Force Master Role Request**

Description	This command will change the preferred role of the LMX9838 e.g. to force master role at any connections. The LMX9838 must be reset to let the changes take effect.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	FORCE_MASTER_ROLE	
DataLength	1	
Data	Role	0x00 Don't care about role
	1 bytes	0x01 Force master role at connection setup

**Table 234. Force Master Role Confirm**

Description	Confirms the request above	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	FORCE_MASTER_ROLE	
DataLength	1	
Data	Status	ERROR_OK
	1 byte	ERROR_INVALID_ROLE
		ERROR_INVALID_NO_OF_PARAMETERS

### 7.2.14.9 Read Operation Mode

The Operation Mode of the LMX9838 controls the behavior in different situations. Please see [Section 3.2.4](#) for details.

**Table 235. Read Operation Mode Request**

Description	This command will read out the current operation mode.
Firmware	0106, 02xx
PacketType	REQ
Opcode	READ_OPERATION_MODE
DataLength	0

**Table 235. Read Operation Mode Request (continued)**

Description	This command will read out the current operation mode.
Data	None

**Table 236. Read Operation Mode Confirm**

Description	Confirms the request above	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	READ_OPERATION_MODE	
DataLength	2	
Data	Status	ERROR_OK
	1 byte	ERROR_INVALID_NO_OF_PARAMETERS
	Mode	0x00 Automatic Operation OFF
	1 Byte	0x01 Automatic Operation ON

#### 7.2.14.10 Write Operation Mode

**Table 237. Write Operation Mode Request**

Description	This command will change the operation mode stored in NVS. The new setting will take effect after a reset.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	WRITE_OPERATION_MODE	
DataLength	1	
Data	Mode	0x00 Automatic Operation OFF
	1 Byte	0x01 Automatic Operation ON

**Table 238. Write Operation Mode Confirm**

Description	Confirms the request above	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	WRITE_OPERATION_MODE	
DataLength	1	
Data	Status	ERROR_OK
	1 byte	ERROR_INVALID_NO_OF_PARAMETERS
		ERROR_INVALID_MODE

#### 7.2.14.11 Set Default Link Latency

**Table 239. Set Default Link Latency Request**

Description	This command is used to change the default SPP link latency. The default link latency is set during SPP connection setup. The default link latency setting is stored in NVS. The link latency is used to calculate a poll interval for the ACL link. The calculated poll interval may be bigger than the latency specified because of e.g. an SCO link or another ACL link. See also <a href="#">Section 5.1.2.7</a> for the impact of the link latency.
Firmware	0106, 02xx
PacketType	REQ
Opcode	SET_DEFAULT_LINK_LATENCY

**Table 239. Set Default Link Latency Request (continued)**

<b>Description</b>	This command is used to change the default SPP link latency. The default link latency is set during SPP connection setup. The default link latency setting is stored in NVS. The link latency is used to calculate a poll interval for the ACL link. The calculated poll interval may be bigger than the latency specified because of e.g. an SCO link or another ACL link. See also <a href="#">Section 5.1.2.7</a> for the impact of the link latency.	
DataLength	2	
Data	Latency	The link latency in slots.
	2 byte	0x0000: No link latency requirement (default)
		0x0002 - 0x0190: Valid link latency
		In case 0x0000 is set the link latency is managed by the master, typically 40 slots (0x0028, 25ms) in a point-to-point connection are used

**Table 240. Set Default Link Latency Confirm**

<b>Description</b>	<b>Response to the request above.</b>	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SET_DEFAULT_LINK_LATENCY	
DataLength	1	
Data	Status	ERROR_OK
	1 byte	ERROR_INVALID_NO_OF_PARAMETERS
		ERROR_LIMIT

### 7.2.14.12 Get Default Link Latency

**Table 241. Get Default Link Latency Request**

<b>Description</b>	This command is used to read the default link latency setting from NVS.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	GET_DEFAULT_LINK_LATENCY	
DataLength	0	

**Table 242. Get Default Link Latency Confirm**

<b>Description</b>	<b>Response to the request above.</b>	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GET_DEFAULT_LINK_LATENCY	
DataLength	2	
Data	Latency	The link latency in slots.
	2 byte	0x0000: No link latency requirement (default)
		0x0002 - 0x0190: Valid link latency
		In case 0x0000 is set the link latency is managed by the master, typically 40 slots (0x0028, 25ms) in a point-to-point connection are used

## 7.2.15 ACL Indications

If the Event Filter is set to “Report all events”, the LMX9838 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 in [Table 297](#).

### 7.2.15.1 ACL Established

**Table 243. ACL Established Indicator**

Description	This indication is sent to the host when an ACL link is established.	
Firmware	0106, 02xx	
PacketType	IND	
Opcode	GAP_ACL_ESTABLISHED	
DataLength	7	
Data	BdAddr	The Bluetooth address of the remote device.
	6 bytes	
	Status	See <a href="#">Section 7.3.1</a>
	1 byte	

### 7.2.15.2 ACL Terminated

**Table 244. ACL Terminated Indicator**

Description	This indication is sent to the host when an ACL link is terminated.	
Firmware	0106, 02xx	
PacketType	IND	
Opcode	GAP_ACL_TERMINATED	
DataLength	7	
Data	BdAddr	The Bluetooth address of the remote device.
	6 bytes	
	Status	See <a href="#">Section 7.3.1</a>
	1 byte	

## 7.2.16 Bluetooth Security

Bluetooth security is part of the Generic Access Profile GAP. It is controlled by:

- Security Mode
  - Security Mode 1: No Security, the device never will ask for authentication or pairing.
  - Security Mode 2:
    - The level of security (Authorization, Authentication, Encryption) is determined by the setting in the service database entries. Each entry can have different security requirements.
    - No authentication necessary for SDAP links.
  - Security Mode 3:
    - Authentication already necessary on Link Manager level: SDAP links already require authentication (Service requests)
    - Device always asks for authentication
- Service Database Entry (only for Security Mode 2)
  - Each entry can specify the settings for authentication and encryption

The LMX9838 by default is in Security Mode 2.

### 7.2.16.1 Get Security Mode

**Table 245. Get Security Mode Request**

Description	Reads the current security mode of the Bluetooth device.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	GAP_GET_SECURITY_MODE	
DataLength	0	
Data	None	

**Table 246. Get Security Mode Confirm**

Description	Confirm to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_GET_SECURITY_MODE	
DataLength	2	
Data	Status	ERROR_OK
	1 byte	ERROR_INVALID_NO_OF_PARAMETERS
	Mode	The current Bluetooth security mode.
	1 byte	0x01 Security mode 1
		0x02 Security mode 2
		0x03 Security mode 3
		0x83 Security mode 3 with link level encryption

### 7.2.16.2 Set Security Mode

**Table 247. Set Security Mode Request**

Description	Changes the current security mode of the Bluetooth device. The security mode is stored in NVS and restored during power up.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	GAP_SET_SECURITY_MODE	
DataLength	1	
Data	Mode	The current Bluetooth security mode.
	1 byte	0x01 Security mode 1
		0x02 Security mode 2
		0x03 Security mode 3
		0x83 Security mode 3 with link level encryption

**Table 248. Set Security Mode Confirm**

Description	Confirm to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_SET_SECURITY_MODE	
DataLength	1	
Data	Status	ERROR_OK
	1 byte	ERROR_INVALID_NO_OF_PARAMETERS
		ERROR_INVALID_SECURITY_MODE

### 7.2.17 Pairing

The pairing process is part of the authentication procedure. If a local or remote service asks for authentication during link establishment, the authentication process will check if a link key already exists between the two devices. If not, the Link Manager initiates the pairing process. Within this, the two devices exchange a PIN code and create a secure link key which will be stored in each device.

During next link setup, the authentication routine takes the existing link key and proceeds without this pairing procedure.

The LMX9838 has a fixed PinCode which can be changed with the command “Change fixed Pin”. This pin is used during any pairing procedure. In case the stored PinCode has length 0x00, the LMX9838 will request a Pin from the application using the “Pin Request” indicator. The application needs to respond with the appropriate pin in the Pin request.

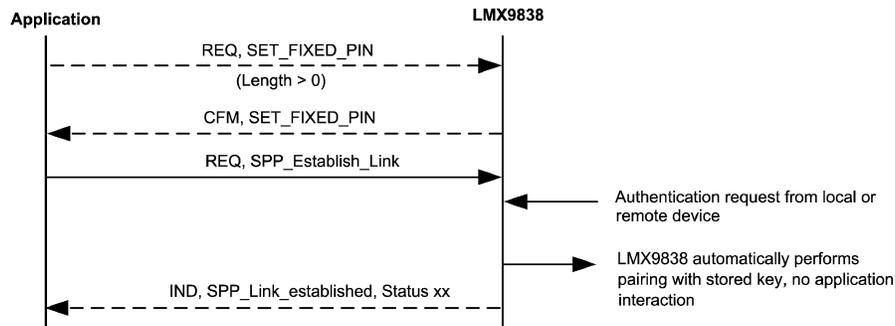


Figure 83. Authentication Procedure with Fixed Pin

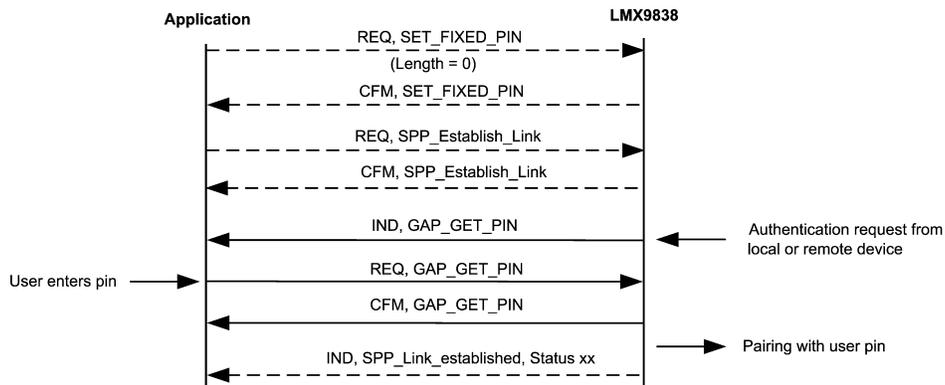


Figure 84. Authentication Procedure with Pin Request

#### 7.2.17.1 Remove Pairing

Table 249. Remove Pairing Command

Description	Remove pairing with a remote device.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	GAP_REMOVE_PAIRING	
DataLength	6	
	BdAddress	Remove pairing to the BdAddress.
	6 byte	

**Table 250. Remove Paired Device Confirm**

Description	Response to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_REMOVE_PAIRING	
DataLength	1	
Data	Status	ERROR_OK
	1 byte	ERROR_INVALID_NO_OF_PARAMETERS
		ERROR_LINKKEY_DOES_NOT_EXISTS

### 7.2.17.2 List Paired Devices

**Table 251. List Paired Devices Command**

Description	Request a list of paired devices from NVS.
Firmware	0106, 02xx
PacketType	REQ
Opcode	GAP_LIST_PAIRIED_DEVICES
DataLength	0

**Table 252. List Paired Devices Confirm**

Description	Response to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	LIST_PAIRIED_DEVICES	
DataLength	2 +6 * DeviceCount	
Data	Status	ERROR_OK
	1 byte	ERROR_INVALID_NO_OF_PARAMETERS
	DeviceCount	Number of devices in the list of paired devices
	1 byte	If 0 the device is not paired to any other devices. The maximum number of paired devices is 7
	BdAddresses	The list of paired devices.
	6 byte * DeviceCount	

### 7.2.17.3 Get Fixed Pin

**Table 253. Get Fixed Pin Request**

Description	Reads the current fixed pin code from NVS
Firmware	0106, 02xx
PacketType	REQ
Opcode	GAP_GET_FIXED_PIN
DataLength	0

**Table 254. Get Fixed Pin Confirm**

Description	Response to the request above.
Firmware	0106, 02xx
PacketType	CFM
Opcode	GAP_GET_FIXED_PIN

**Table 254. Get Fixed Pin Confirm (continued)**

Description	Response to the request above.	
DataLength	2+Pinlength	
Data	Status	ERROR_OK
	1 byte	ERROR_INVALID_NO_OF_PARAMETERS
	Pinlength	Length of pin code, in bytes.
	1 byte	0x00: No Fixed pin, LMX9838 will send "Pin request indicator" () in case a pin is required.
		Range: 0x01-0x10
	Pincode	PIN code used when the two Bluetooth devices are paired. The maximum length of a PIN code is 128 bits (16 bytes).
	Pinlength bytes	

### 7.2.17.4 Set Fixed Pin

**Table 255. Set Fixed Pin Request**

Description	Stores a new fixed pin code in NVS	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	GAP_SET_FIXED_PIN	
DataLength	1+ Pinlength	
	Pinlength	Length of pin code, in bytes.
	1 byte	0x00: No Fixed pin, LMX9838 will "Pin request indicator" () in case a pin is required.
		Range: 0x01-0x10
	Pincode	PIN code used when the two Bluetooth devices are paired. The maximum length of a PIN code is 128 bits (16 bytes).
	Pinlength bytes	

**Table 256. Set Fixed Pin Confirm**

Description	Response to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GAP_SET_FIXED_PIN	
DataLength	1	
Data	Status	ERROR_OK
	1 byte	ERROR_INVALID_NO_OF_PARAMETERS
		ERROR_PINCODE_TOO_LONG

### 7.2.17.5 PIN Request

The PIN request procedure will only be used, in case the Pincode length stored in NVS is 0. See also [Figure 82](#).

#### 7.2.17.5.1 Pin Request Indicator

**Table 257. Pin request indicator**

<b>Description</b>	<b>This event is used to inform the Host when a PIN code is requested during authentication of an ACL link. This event is only generated if the length of the Fixed pin parameter stored in the NVS is set to 0.</b>	
Firmware	02xx	
PacketType	IND	
Opcode	GAP_GET_PIN	
DataLength	6	
Data	BdAddr	The Bluetooth device address of the remote device.
	6 byte	

### 7.2.17.5.2 Pin Request Response

**Table 258. Pin Request Response Command**

<b>Description</b>	<b>This command is used to send a PIN code to the LMX9838 as response to a GAP_GET_PIN indication.</b>	
Firmware	02xx	
PacketType	REQ	
Opcode	GAP_GET_PIN	
DataLength	1	
DataLength	7 + Pinlength	
Data	BdAddr	The Bluetooth device address of the remote device.
	6 byte	
	Pinlength	Length of pin code, in bytes.
	1 byte	Range: 0x00-0x10
		0x00 indicates that the Host does not allow the authentication of the ACL link.
	Pincode	PIN code used when the two Bluetooth devices are paired. The maximum length of a PIN code is 128 bits (16 bytes).
	Pinlength bytes	

**Table 259. Pin Request Command Confirm**

<b>Description</b>	<b>Response to the request above.</b>	
Firmware	02xx	
PacketType	CFM	
Opcode	GAP_GET_PIN	
DataLength	1	
Data	Status	ERROR_OK
	1 byte	ERROR_INVALID_NO_OF_PARAMETERS
		ERROR_PINCODE_LENGTH

## 7.2.18 Hardware Commands

### 7.2.18.1 Set Event Filter

**Table 260. Set Event Filter Command**

<b>Description</b>	<b>This command is used to set the event filter. The setting is stored in NVS.</b>
Firmware	0106, 02xx
PacketType	REQ

**Table 260. Set Event Filter Command (continued)**

Description	This command is used to set the event filter. The setting is stored in NVS.	
Opcode	SET_EVENT_FILTER	
DataLength	1	
Data	Filter	0x00: All events reported
	1 byte	0x01: No ACL Link Indicators (default)
		0x02: No events reported, UART break still generated and detected.
		0x03: No events generated, UART break not generated or detected (full cable replacement)

**Table 261. Set Event Filter Confirm**

Description	Response to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	SET_EVENT_FILTER	
DataLength	1	
Data	Status	ERROR_OK
	1 byte	ERROR_INVALID_NO_OF_PARAMETERS
		ERROR_LIMIT

### 7.2.18.2 Get Event Filter

**Table 262. Get Event Filter Command**

Description	This command reads the current event filter setting.
Firmware	0106, 02xx
PacketType	REQ
Opcode	GET_EVENT_FILTER
DataLength	0

**Table 263. Get Event Filter Confirm**

Description	Response to the request above.	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	GET_EVENT_FILTER	
DataLength	1	
Data	Filter	0x00: All events reported
	1 byte	0x01: No ACL Link Indicators (default)
		0x02: No events reported, UART break still generated and detected.
		0x03: No events generated, UART break not generated or detected (full cable replacement)

### 7.2.18.3 Reset

**Table 264. Reset Request**

<b>Description</b>	<b>This command will perform a soft reset of the LMX9838. The LMX9838 will send a LMX9838_READY indication when it has performed the reset.</b>	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	RESET	
DataLength	0	
Data	None	

#### 7.2.18.4 LMX9838 Ready

**Table 265. Simply Blue Ready Indicator**

<b>Description</b>	<b>The LMX9838 will send this indication to the host when the device is fully initialized and ready to receive commands from the host.</b>	
Firmware	0106, 02xx	
PacketType	IND	
Opcode	DEVICE_READY	
DataLength	1+Length	
Data	Length	Number of bytes in software version string
	1 byte	
	version <Length> bytes	ASCII string containing the software version. e.g. 0210 indicating that the software version is version 2.10.

#### 7.2.18.5 Restore Factory Settings

**Table 266. Restore Factory Settings Request**

<b>Description</b>	<b>This command will restore the LMX9838 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 <a href="#">Table 2</a>. The LMX9838 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.</b>	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	RESTORE_FACTORY_SETTINGS	
DataLength	0	
Data	none	

**Table 267. Restore Factory Settings Confirm**

<b>Description</b>	<b>Confirms the request above</b>	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	RESTORE_FACTORY_SETTINGS	
DataLength	1	
Data	Status	ERROR_OK
	1 byte	ERROR_INVALID_NO_OF_PARAMETERS

### 7.2.18.6 Change NVS UART Speed

The UART Speed is only active if the OP pin configuration of the LMX9838 is set to “Read NVS Settings”.

**Table 268. Change UART Speed Request**

Description	This command will change the UART speed stored in NVS. The new UART speed will be used after a reset.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	CHANGE_NVS_UART_SPEED	
DataLength	1	
Data	UartSpeed	The UART speed to be stored in NVS
	1 Byte	0x00 = 2400
		0x01 = 4800
		0x02 = 7200
		0x03 = 9600
		0x04 = 19200
		0x05 = 38400
		0x06 = 57600
		0x07 = 115200
		0x08 = 230400
		0x09 = 460800
	0x0A = 921600	

**Table 269. Change UART Speed Confirm**

Description	Confirms the request above	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	CHANGE_NVS_UART_SPEED	
DataLength	1	
Data	Status	ERROR_OK
	1 byte	ERROR_UART_SPEED_OUT_OF_RANGE
		ERROR_INVALID_NO_OF_PARAMETERS

### 7.2.18.7 Change UART Settings

UART Settings will be stored in NVS and are valid for ALL UART speeds at LMX9838.

**Table 270. Change UART Settings Request**

Description	This command will change the UART settings stored in NVS. The new UART settings will be used after a reset.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	CHANGE_UART_SETTINGS	
DataLength	2	
Data	ParityType	0x00 None
	1 Byte	0x01 Even
		0x02 Odd
	StopBits	0x00 One Stop bit
	1 Byte	0x01 Two Stop bits

**Table 271. Change UART Settings Confirm**

Description	Confirms the request above	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	CHANGE_UART_SETTINGS	
DataLength	1	
Data	Status	ERROR_OK
	1 byte	ERROR_PARITY_BIT_OUT_OF_RANGE,
		ERROR_STOP_BITS_OUT_OF_RANGE,
		ERROR_INVALID_NO_OF_PARAMETERS

## 7.2.19 Test Modes

### 7.2.19.1 Bluetooth Test Mode

Bluetooth Qualification requires specific test modes to prove the functionality and quality of the bluetooth device.

The Test Mode Command offers the ability to enable either the bluetooth specified “Device Under Test” Mode and also a UART Loopback mode.

This modes can only be left by a reset.

**Table 272. Initiate Bluetooth Test Mode Request**

Description	This command will activate the test mode. To exit the test mode, a reset of the device must be performed. If local loopback mode is activated all data send to the device though the UART, are send back to the host. The device can only leave local loopback mode by performing a hardware reset.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	TEST_MODE	
DataLength	1	
Data	Mode	The Bluetooth test mode to enter:
	1 byte	0x01 Enable Bluetooth test mode
		0x02 Enable local loopback mode

**Table 273. Initiate Bluetooth Test Mode Confirm**

Description	Confirms the request above	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	TEST_MODE	
DataLength	1	
Data	Status	ERROR_OK
	1 byte	ERROR_ILLEGAL_TESTMODE
		ERROR_UNKNOWN_ERROR
		ERROR_INVALID_NO_OF_PARAMETERS

### 7.2.19.2 Initiate RF Test Mode

In addition to the standard Bluetooth Test Mode the LMX9838 offers a special RF Test Mode, switching the transmitter into a continuous transmit or receive mode. This is needed for Bluetooth qualification as well as regulatory testings for FCC and ETSI and will be needed for country specific qualification.

**Table 274. Initiate RF Test Mode Request**

Description	This command will activate the RF test mode. To exit the test mode, a reset of the device must be performed. The transmitter test must be stopped before a new test with changed parameters can be started.	
Firmware	0106, 02xx	
PacketType	REQ	
Opcode	RF_TEST_MODE	
DataLength	14	
Data	Test	Test Scenario:
	1 byte	0x00: Stop transmit
		0x01: Burst transmit, take payload from this command
		0x02: Burst transmit, use PRBS-9 sequence
		0x03: Continuous transmit, sends all-1 if ModulationPayload is nonzero, otherwise all-0
		0x04: Hopping receive
		0x05: continuous receive at fixed channel
Channel	Channel	Channel number 0 78
	1 byte	0: 2402 MHz
		78: 2480 MHz
		255 (0xFF): Hopping in connection state
PaCtrl	PaCtrl	Reserved for future use
	1 byte	
ModulationCtrl	ModulationCtrl	Modulation Control:
	1 byte	0: No modulation
		1: Access code only (68 us TX data every 1250 us note 1)
		2: Bluetooth DH1 packet with defined payload (note 2)
		3: Bluetooth DM1 packet (17 bytes)
		4: Bluetooth DH1 packet (27 bytes)
		5: Invalid
		6: Invalid
		7: Bluetooth HV3 packet (30 bytes)
		8: Invalid
		9: Bluetooth AUX1 packet (29 bytes)
		10: Bluetooth DM3 packet (121 bytes)
		11: Bluetooth DH3 packet (183 bytes)
		12: Invalid
		13: Invalid
		14: Bluetooth DM5 packet (224 bytes)
	15: Bluetooth DH5 packet (339 bytes)	
ModulationPattern	ModulationPattern	Defines Access Code modulation if ModulationCtrl = 1, else ignored.
	8 bytes	
ModulationPayload	ModulationPayload	Defines one byte of payload repeated through packet if
	1 byte	Test = 1, else ignored
Options	Options	Bit 0: Whitening: 1: enabled, 0: disabled
	1 byte	

**Table 275. Initiate RF Test Mode Confirm**

Description	Confirms the request above	
Firmware	0106, 02xx	
PacketType	CFM	
Opcode	RF_TEST_MODE	
DataLength	1	
Data	Status	ERROR_OK
	1 byte	ERROR_INVALID_NO_OF_PARAMETERS

### 7.2.19.3 Enter Bluetooth Mode

**Table 276. Enter Bluetooth Mode Request**

Description	The host uses this command to request the SimplyBlue module to enter Bluetooth mode after initialization parameters have been sent to the module.	
PacketType	REQ	
Opcode	ENTER_BLUETOOTH_MODE	
DataLength	0	
Data	None	

**Table 277. Enter Bluetooth Mode Confirm**

Description	Confirms the request above	
PacketType	CFM	
Opcode	ENTER_BLUETOOTH_MODE	
DataLength	1	
Data	Status	ERROR_OK
	1 byte	ERROR_COMMAND_DISALLOWED
		ERROR_INVALID_NO_OF_PARAMETERS

### 7.2.19.4 Set Clock And Baud Rate

**Table 278. Set Clock and Baudrate Request**

Description	This command is used to write the exact base band clock frequency and baud rate used. This command must be sent as the first command if UART auto baud rate detection is used. In case of first power-up with EEPROM, 9600 will be used.	
PacketType	REQ	
Opcode	SET_CLOCK_AND_BAUDRATE	
DataLength	8	
Data	ClockFrequency	The Clock Frequency to use.
	4 bytes	
	BaudRate	The baud rate to use.
	4 bytes	

**Table 279. Set Clock and Baudrate Confirm**

Description	Response to the request above.	
PacketType	CFM	
Opcode	SET_CLOCK_AND_BAUDRATE	
DataLength	1	

**Table 279. Set Clock and Baudrate Confirm (continued)**

Description	Response to the request above.	
Data	Status	ERROR_OK
	1 byte	ERROR_INVALID_NO_OF_PARAMETERS
		ERROR_COMMAND_DISALLOWED

## 7.2.20 Patching

Patching allows to correct small errors in ROM code. A patch has to be written into EEPROM. Please refer also to Section 5.6 "Software Patches" on page 58.

### 7.2.20.1 Write ROM Patch

**Table 280. Write ROM Patch Request**

Description	This command is used to store ROM patch code in the SimplyBlue module.	
PacketType	REQ	
Opcode	WRITE_ROM_PATCH	
DataLength	1 to 333 bytes	
Data	Patch data	Stream of Patch Data.

**Table 281. Write ROM Patch Confirm**

Description	Confirms the request above.	
PacketType	CFM	
Opcode	WRITE_ROM_PATCH	
DataLength	1	
Data	Status	0x00 = Ok, Done
	1 byte	0x01 = Ok, Continue
		0x80 = Error, Not enough info to continue download (a global or segment meta data group appears to be split)
		0x81 = Error, Not enough resources to continue download
		0x82 = Error, Patch too big
		0x83 = Error, Unsupported Patch format revision
		0x84 = Error, Patch not applicable to firmware in device.
		0x85 = Error, Patch CRC check failed.
		0x86 = Error, Patch NVS validation failed.
		0x87 = Error, RAMAddr or VarAddr out of RAM area.
	0xFF = Error, Unspecified error.	

## 7.2.21 GPIO handling

The LMX9838 offers three GPIOs which can be controlled and read back by the command interface. By default pins PG6 and PG7 are used by the firmware to indicate pin status and RF link activity. The following commands allow to change the usage of the pins and to configure them to the desired function.

Preparation:

In order to use pins PG6 and PG7 as GPIOs, EEPROM parameter #19 ([Table 2](#)) at address 0x0062 needs to be configured with the "Write NVS" command. This control byte configures the Enhanced Power management and the GPIO pin usage.

For example:

To use PG6 and PG7 as standard GPIOs and leave Enhanced Power Management disabled (default) use the following hex string:

```
02 52 73 04 00 C9 62 00 01 F9 03
```

The change requires a "Reset" to get active.

Using the GPIO commands:

Once this is done, the GPIOs can be configured with of the following GPIO commands. At each of the commands, the pin names are mapped to a specific bit position. Please see [Table 282](#) for the pin assignment.

**Table 282. Control Byte Pin Assignments**

Bit	7	6	5	4	3	2	1	0
Pin	PG7	PG6	(reserved)	PG4	(reserved)	(reserved)	(reserved)	(reserved)

### 7.2.21.1 Set GPIO Weak Pull-up

Each GPIO can be configured with internal Weak pull-ups, in case they are used as Input. With this a default input value of HIGH can be reached.

**Table 283. Set GPIO WPU Request**

Description	This command is used to enable or disable the weak pull up resistors on GPIO PG[4], PG[6] and PG[7].	
PacketType	REQ	
Opcode	SET_GPIO_WPU	
DataLength	1	
Data	WPU	If 1, the WPU resistor on PG[x] is enabled.
	1 bytes	If 0, the WPU resistor on PG[x] is disabled.
		See <a href="#">Table 281</a> for the bit assignments of the pins.

**Table 284. Set GPIO WPU Confirm**

Description	Confirms the request above.	
PacketType	CFM	
Opcode	SET_GPIO_WPU	
DataLength	1	
Data	Status	ERROR_OK
	1 byte	ERROR_UNEXPECTED
		ERROR_INVALID_NO_OF_PARAMETERS

### 7.2.21.2 Get GPIO Input State

This command allows to read back the status of the pins.

**Table 285. Get GPIO Input State Request**

Description	This command is used to read the PGDIN_REG register and return the state of PG[4], PG[6] and PG[7].
PacketType	REQ
Opcode	GET_GPIO_STATUS
DataLength	1
Data	None

**Table 286. Get GPIO Status Confirm**

Description	Confirms the request above.	
PacketType	CFM	
Opcode	GET_GPIO_STATUS	
DataLength	1	
Data	Status	ERROR_OK
	1 byte	ERROR_INVALID_NO_OF_PARAMETERS
	State	PG[x] is 1 if the input is high and 0 if the input is low.
	1 bytes	See <a href="#">Table 281</a> for the bit assignments of the pins.

### 7.2.21.3 Set GPIO Direction

This command allows to configure the direction of the GPIO pins in case they are not used by the bluetooth firmware.

**Table 287. Set GPIO Direction Request**

Description	This command is used to set the direction of GPIO PG[4], PG[6] and PG[7].	
PacketType	REQ	
Opcode	SET_GPIO_DIRECTION	
DataLength	1	
Data	Direction	If 1, PG[x] is output.
	1 bytes	If 0, PG[x] is input.

**Table 288. Set GPIO Direction Confirm**

Description	Confirms the request above.	
PacketType	CFM	
Opcode	SET_GPIO_DIRECTION	
DataLength	1	
Data	Status	ERROR_OK
	1 byte	ERROR_UNEXPECTED
		ERROR_INVALID_NO_OF_PARAMETERS

### 7.2.21.4 Set GPIO Output High

**Table 289. Set GPIO Output High Request**

Description	This command is used to set the output pins on PG[4], PG[6] and PG[7] to logical high.	
PacketType	REQ	
Opcode	SET_GPIO_OUTPUT_HIGH	
DataLength	1	
Data	Output	If 1, PG[x] output is set high.
	1 bytes	If 0, PG[x] output is unchanged.
		See <a href="#">Table 281</a> for the bit assignments of the pins.

**Table 290. Set GPIO Output High Confirm**

Description	Confirms the request above.	
PacketType	CFM	
Opcode	SET_GPIO_OUTPUT_HIGH	
DataLength	1	

**Table 290. Set GPIO Output High Confirm (continued)**

Description	Confirms the request above.	
Data	Status	ERROR_OK
	1 byte	ERROR_UNEXPECTED
		ERROR_INVALID_NO_OF_PARAMETERS

### 7.2.21.5 Set GPIO Output Low

**Table 291. Set GPIO Output Low Request**

Description	This command is used to set the output pins on PG[4], PG[6] and PG[7] to logical low.	
PacketType	REQ	
Opcode	SET_GPIO_OUTPUT_LOW	
DataLength	1	
Data	Output	If 1, PG[x] output is set low.
	1 bytes	If 0, PG[x] output is unchanged.

**Table 292. Set GPIO Output Low Confirm**

Description	Confirms the request above.	
PacketType	CFM	
Opcode	SET_GPIO_OUTPUT_LOW	
DataLength	1	
Data	Status	ERROR_OK
	1 byte	ERROR_UNEXPECTED
		ERROR_INVALID_NO_OF_PARAMETERS

## 7.2.22 EEPROM / NVS handling

On each boot-up or reset the LMX9838 tries to read all parameters required for operation from the internal EEPROM and copies them into a "RAM NVS" which is used during operation. The EEPROM (also called Non Volatile Storage) can be addressed by Read and Write commands to change configuration parameters directly. The EEPROM memory map is listed in [Table 2](#).

### 7.2.22.1 Read NVS

**Table 293. Read NVS Request**

Description	This command can be used to read from NVS which is either in EEPROM or from RAM, depending on the availability of an EEPROM.	
PacketType	REQ	
Opcode	READ_NVS	
DataLength	3	
Data	Address	The address.
	2 bytes	See <a href="#">Table 2</a> .
Length		The number of bytes that should be read.
	1 byte	

**Table 294. Read NVS Confirm**

Description	Confirms the request above.	
PacketType	CFM	
Opcode	READ_NVS	
DataLength	4 + Length	
Data	Status	ERROR_OK
	1 byte	ERROR_INVALID_NO_OF_PARAMETERS
	Address	The address.
	2 bytes	See <a href="#">Table 2</a> .
	Length	The number of bytes that has been read.
	1 byte	
Data	The data.	
	< Length > bytes	

### 7.2.22.2 Write NVS

**Table 295. Write NVS Request**

Description	This command can be used to write data to the NVS, which is either stored in EEPROM or in RAM depending on the availability of an EEPROM.	
PacketType	REQ	
Opcode	WRITE_NVS	
DataLength	3 + Length	
Data	Address	The address.
	2 bytes	See <a href="#">Table 2</a> .
	Length	The number of bytes that should be written.
	1 byte	
Data	The data.	
	< Length > bytes	

**Table 296. Write NVS Confirm**

Description	Confirms the request above.	
PacketType	CFM	
Opcode	WRITE_NVS	
DataLength	4	
Data	Status	ERROR_OK
	1 byte	ERROR_INVALID_NO_OF_PARAMETERS
	Address	The address.
	2 bytes	See <a href="#">Table 2</a> .
	Length	The number of bytes that has been written
	1 byte	

## 7.3 ERROR CODES

### 7.3.1 ACL Error Codes

The following table is copied out of.

**Table 297. ACL Error Codes**

Error Code	Description
0x01	Unknown HCI Command.
0x02	No Connection.
0x03	Hardware Failure.
0x04	Page Timeout.
0x05	Authentication Failure.
0x06	Key Missing.
0x07	Memory Full.
0x08	Connection Timeout.
0x09	Max Number Of Connections.
0x0A	Max Number Of SCO Connections To A Device.
0x0B	ACL connection already exists.
0x0C	Command Disallowed.
0x0D	Host Rejected due to limited resources.
0x0E	Host Rejected due to security reasons.
0x0F	Host Rejected due to remote device is only a personal device.
0x10	Host Timeout.
0x11	Unsupported Feature or Parameter Value.
0x12	Invalid HCI Command Parameters.
0x13	Other End Terminated Connection: User Ended Connection.
0x14	Other End Terminated Connection: Low Resources.
0x15	Other End Terminated Connection: About to Power Off.
0x16	Connection Terminated by Local Host.
0x17	Repeated Attempts.
0x18	Pairing Not Allowed.
0x19	Unknown LMP PDU.
0x1A	Unsupported Remote Feature.
0x1B	SCO Offset Rejected.
0x1C	SCO Interval Rejected.
0x1D	SCO Air Mode Rejected.
0x1E	Invalid LMP Parameters.
0x1F	Unspecified Error.
0x20	Unsupported LMP Parameter Value.
0x21	Role Change Not Allowed
0x22	LMP Response Timeout
0x23	LMP Error Transaction Collision
0x24	LMP PDU Not Allowed
0x25	Encryption Mode Not Acceptable
0x26	Unit Key Used
0x27	QoS is Not Supported
0x28	Instant Passed
0x29	Pairing with Unit Key Not Supported
0x2A	Different Transaction Collision
0x2B	Reserved
0x2C	QoS Unacceptable Parameter
0x2D	QoS Rejected
0x2E	Channel Classification Not Supported
0x2F	Insufficient Security
0x30	Parameter out of Mandatory Range

**Table 297. ACL Error Codes (continued)**

Error Code	Description
0x31	Reserved
0x32	Role Switch Pending
0x33	Reserved
0x34	Reserved Slot Violation
0x35	Role Switch Failed

### 7.3.2 Generic Error Codes

**Table 298. Generic Error Codes**

Error code	Macro	Description
0x00	ERROR_OK	No error.
0x01	ERROR_INVALID_NO_OF_PARAMETERS	The number of bytes in the request does not correspond to the protocol specification
0x02	ERROR_DURATION_OUT_OF_RANGE	The given duration value is not valid according to the specification.
0x03	ERROR_INVALID_MODE	The selected mode is not valid according to the specification
0x04	ERROR_TIMEOUT	A timeout occurred.
0x05	ERROR_UNKNOWN_ERROR	An unknown error occurred.
0x06	ERROR_NAME_TOO_LONG	The number of bytes in the name string is longer than the maximum specified value.
0x07	ERROR_INVALID_DISCOVERABILITY_PARAMETER	The given discoverability parameter does not contain a valid value according to the specification.
0x08	ERROR_INVALID_CONNECTABILITY_PARAMETER	The given connectability parameter does not contain a valid value according to the specification.
0x09	ERROR_INVALID_SECURITY_MODE	The given security mode is not a valid Bluetooth security mode.
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 LMX9838. The answer is truncated
0x0d	ERROR_RESULT_TOO_LARGE	The SDP result from the remote device is too large to be handled by the LMX9838 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_TI_BDADDRESS	The LMX9838 will change the Bluetooth address to the TI 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.

**Table 298. Generic Error Codes (continued)**

Error code	Macro	Description
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 LMX9838 is able to store, is reached.
0x19	ERROR_WRITING_TO_NV	An error occurred while writing to flash. The service record may not be stored.
0x1a	ERROR_INVALID_ROLE	The given role value is not a valid value according to the specification.
0x1b	ERROR_LIMIT	Limits exceeded (Parameter(s) violates limits).
0x1c	ERROR_UNEXPECTED	Unexpected at this moment.
0x1d	ERROR_UNABLE_TO_SEND	Could not send at this moment, no reason specified.
0x1e	ERROR_CURRENTLY_NO_BUFFER	Currently no room in buffer, try again later.
0x1f	ERROR_NO_CONNECTION	Trying to use an inexistent connection.
0x20	ERROR_SPP_INVALID_PORT	Port number out of range.
0x21	ERROR_SPP_PORT_NOT_OPEN	Port is closed.
0x22	ERROR_SPP_PORT_BUSY	Connection establishment on a PORT that has a connection.
0x23	ERROR_SPP_MULTIPLE_CONNECTIONS	Transparent mode attempted while more than 1 connection active.
0x24	ERROR_SPP_MULTIPLE_TRANSPARENT	Trying to store a default connection when a transparent default connection is already stored, or trying to store a transparent default connection when another connection is already stored.
0x25	ERROR_SPP_DEFAULT_CONNECTION_NOT_STORED	Trying to connection to a default connection, which is not stored.
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 exceeds the specified maximum number of supported formats.
0x2b	ERROR_TOO_MANY_DATASTORES	The given number of data stores excess the specified maximum number of data stores.
0x2C	ERROR_ATTEMPT_FAILED	Attempt to change low power mode failed
0x2D	ERROR_ILLEGAL_LINK_POLICY	The given link policy value is out of range
0x2E	ERROR_PINCODE_TOO_LONG	The pin code length field is too large.
0x2F	ERROR_PARITY_BIT_OUT_OF_RANGE	The given parity check is out of range
0x30	ERROR_STOP_BITS_OUT_OF_RANGE	The given number of stop bits is out of range
0x31	ERROR_ILLEGAL_LINK_TIMEOUT	The given link timeout value is out of range
0x32	ERROR_COMMAND_DISALLOWED	The command is not allowed.
0x33	ERROR_ILLEGAL_AUDIO_CODEC_TYPE	The given Audio CODEC type is out of range.
0x34	ERROR_ILLEGAL_AUDIO_AIR_FORMAT	The given Audio Air format is out of range.
0x35	ERROR_SDP_RECORD_TOO_LONG	The SDP record is too long.
0x36	ERROR_SDP_FAILED_TO_CREATE_RECORD	The SDP server failed to create the SDP record.
0x37	ERROR_SET_VOLUME_FAILED	The selected codec does not support volume control.
0x38	ERROR_ILLEGAL_PACKET_TYPE	The packet type specified in the request is not valid.
0x39	ERROR_INVALID_CODEC_SETTING	The codec (slave) settings is invalid.

### 7.3.3 RFCOMM Error Codes

**Table 299. RFCOMM Error Codes**

ErrorCode	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

### 7.3.4 RFCOMM Release Reasons

**Table 300. RFCOMM Release Reasons**

Error code	Macro	Description
0x00	RFDR_DLC_DISC_LOCAL_DEVICE	The local device has disconnected the DLC.
0x01	RFDR_DLC_DISC_REMOTE_DEVICE	The remote device has disconnected the DLC.
0x02	RFDR_DLC_DISC_ACL_FAILURE	ACL link failure/ link supervision timeout.
0x03	RFDR_DLC_DISC_LOWER_LAYER	Lower layer (e.g. L2CAP) has disconnected the DLC.

## 7.4 AT COMMANDS

The list of AT Commands is derived out of the “Bluetooth Profiles Book V1.1” from the Bluetooth SIG. All commands shall be implemented as described in International Telecommunication Union, “ITU-T Recommendation V.250”

### 7.4.1 DUN GW

#### 7.4.1.1 Required Commands

**Table 301. Required AT Commands for DUN GW**

Name	Description
&C	Circuit 109 (Received line signal detector) Behavior
&D	Circuit 108 (Data terminal ready) Behavior
&F	Set to Factory-defined Configuration
+GCAP	Request Complete Capabilities List
+GMI	Request Manufacturer Identification
+GMM	Request Model Identification
+GMR	Request Revision Identification
A	Answer
D	Dial
E	Command Echo
H	Hook Control
L	Monitor Speaker Loudness
M	Monitor Speaker Mode
O	Return to Online Data State
P	Select Pulse Dialling

**Table 301. Required AT Commands for DUN GW (continued)**

Name	Description
Q	Result Code Suppression
S0	Automatic Answer
S10	Automatic Disconnect Delay
S3	Command Line Termination Character
S4	Response Formatting Character
S5	Command Line Editing Character
S6	Pause Before Blind Dialling
S7	Connection Completion Timeout
S8	Comma Dial Modifier Time
T	Select Tone Dialling
V	DCE Response Format
X	Result Code Selection and Call
	Progress Monitoring Control
Z	Reset To Default Configuration

#### 7.4.1.2 Required Result Codes

**Table 302. Result Codes for DUN**

Name	Description
&C	Circuit 109 (Received line signal detector) Behavior
OK	Acknowledges execution of a command.
CONNECT	Connection has been established.
RING	The DCE has detected an incoming call signal from the network.
NO CARRIER	The connection has been terminated, or the attempt to establish a connection failed.
ERROR	Error.
NO DIALTONE	No dial-tone detected.
BUSY	Busy signal detected.

#### 7.4.2 Headset Profile

The following AT commands are defined for the headset profile.

##### 7.4.2.1 Headset AT Command

**Table 303. AT Commands for Headset Profile**

AT Capability	Syntax	Description	Values
RING		The Incoming call indication of ITU V.250, Section 6.3.4	
Microphone gain	+VGM=<gain>	Unsolicited result code issued by the HSAG to set the microphone gain of the HS. <gain> is a decimal numeric constant, relating to a particular (implementation-dependent) volume level controlled by the HS.	<gain>: 0-15
Speaker gain	+VGS=<gain>	Unsolicited result code issued by the HSAG to set the speaker gain of the HS. <gain> is a decimal numeric constant, relating to a particular (implementation-dependent) volume level controlled by the HS.	<gain>: 0-15
Microphone gain level report	+VGM=<gain>	Command issued by the HS to report the current microphone gain level setting to the HSAG. <gain> is a decimal numeric constant, relating to a particular (implementation-dependent) volume level controlled by the HS	<gain>: 0-15

**Table 303. AT Commands for Headset Profile (continued)**

AT Capability	Syntax	Description	Values
Speaker gain level indication report	+VGS=<gain>	Command issued by the HS to report the current speaker gain level setting to the HSAG. <gain> is a decimal numeric constant, relating to a particular (implementation-dependent) volume level controlled by the HS	<gain>: 0-15
Headset button press	+CKPD=200	Command issued by the HS to indicate that the button has been pressed.	

### 7.4.2.2 Required Result Codes

**Table 304. Required Result Codes for Headset Implementations**

Name	Description
OK	Acknowledges execution of a command.
RING	The DCE has detected an incoming call signal from the network.
ERROR	Error.

## 8 Annex

### 8.1 INITIALIZATION IF EEPROM EMPTY

THE INTERNAL EEPROM OF THE LMX9838 SHOULD NEVER BE EMPTY! NEVER DELETE ITS CONTENT !

If for whatever reason the content of the internal EEPROM is deleted, the LMX9838 needs to be initialized.

The LMX9838 requires several parameters to be able to enter Bluetooth Mode. Usually on power-up the device checks the its EEPROM for those parameters and copies them into the RAM mirror.

The following parameters need at least to be configured:

- Bluetooth Address (BD\_ADDR)
- UART Baudrate (required in case the OP pins are set to "Read from NVS" or "Autobaudrate")

Note: if the OP pin settings are set to "Read from NVS", the UART speed will be 9600 BPS at the startup. So please configure the host to 9600 BPS.

Depending on the OP pin settings, there is two ways to initialize the LMX9838.

#### 8.1.1 OP Pins are set to Hardware Selection

Below is the flow steps to ensure a successful startup in this case.

- Configure the UART speed by hardware using the OP4 and OP5 pins. For speed details see "Texas Instruments: LMX9838 Datasheet"
- Set the unique BD address by sending the command "Change Local Bluetooth Address"

Once those steps are completed, the module will leave the so-called "Initialization Mode" and should send the "SimplyBlue Ready" event. The LMX9838 is now in "Bluetooth operation" mode.

#### 8.1.2 OP Pins are set to NVS Selection

Below is the flow steps to ensure a successful startup in this case.

- If the EEPROM is completely empty, the UART speed in this case will be 9600 BPS. So set eventually the UART speed of the host device to 9600 BPS.
- Send the Autobaudrate by sending 0x01 to the LMX9838 module.
- The LMX9838 module will report the "Await Initialization Event".

- Set the unique BD address by sending the command "Change Local Bluetooth Address"

Once those steps are completed, the module will leave the so-called "Initialization Mode" and should send the "SimplyBlue Ready" event. The LMX9838 is now in "Bluetooth operation" mode.

### 8.1.2.1 *Await Initialization Event*

**Table 305. Await Initialization Indicator**

Description	The LMX9838 will send this indication to the host right after a reset when the device needs initialization parameters before it can continue.
Firmware	0106, 02xx
PacketType	IND
Opcode	AWAIT_INITIALIZATION_EVENT
DataLength	0
Data	None

## 9 Bibliography

(1)(2)(3)(4)

- (1) Texas Instruments: LMX9838 Datasheet
- (2) Bluetooth SIG: Specification of the Bluetooth System 1.2, November 05, 2003
- (3) Bluetooth SIG: Specification of the Bluetooth System 1.1, Volume 2 / Profiles, Version 1.1, February 22 2001
- (4) Bluetooth SIG: Bluetooth Assigned Numbers, <https://programs.bluetooth.org/Technical/AssignedNumbers/baseband.htm>

## IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as "components") are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI's terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers' products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers' products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI's goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or "enhanced plastic" are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have *not* been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

### Products

Audio	<a href="http://www.ti.com/audio">www.ti.com/audio</a>
Amplifiers	<a href="http://amplifier.ti.com">amplifier.ti.com</a>
Data Converters	<a href="http://dataconverter.ti.com">dataconverter.ti.com</a>
DLP® Products	<a href="http://www.dlp.com">www.dlp.com</a>
DSP	<a href="http://dsp.ti.com">dsp.ti.com</a>
Clocks and Timers	<a href="http://www.ti.com/clocks">www.ti.com/clocks</a>
Interface	<a href="http://interface.ti.com">interface.ti.com</a>
Logic	<a href="http://logic.ti.com">logic.ti.com</a>
Power Mgmt	<a href="http://power.ti.com">power.ti.com</a>
Microcontrollers	<a href="http://microcontroller.ti.com">microcontroller.ti.com</a>
RFID	<a href="http://www.ti-rfid.com">www.ti-rfid.com</a>
OMAP Applications Processors	<a href="http://www.ti.com/omap">www.ti.com/omap</a>
Wireless Connectivity	<a href="http://www.ti.com/wirelessconnectivity">www.ti.com/wirelessconnectivity</a>

### Applications

Automotive and Transportation	<a href="http://www.ti.com/automotive">www.ti.com/automotive</a>
Communications and Telecom	<a href="http://www.ti.com/communications">www.ti.com/communications</a>
Computers and Peripherals	<a href="http://www.ti.com/computers">www.ti.com/computers</a>
Consumer Electronics	<a href="http://www.ti.com/consumer-apps">www.ti.com/consumer-apps</a>
Energy and Lighting	<a href="http://www.ti.com/energy">www.ti.com/energy</a>
Industrial	<a href="http://www.ti.com/industrial">www.ti.com/industrial</a>
Medical	<a href="http://www.ti.com/medical">www.ti.com/medical</a>
Security	<a href="http://www.ti.com/security">www.ti.com/security</a>
Space, Avionics and Defense	<a href="http://www.ti.com/space-avionics-defense">www.ti.com/space-avionics-defense</a>
Video and Imaging	<a href="http://www.ti.com/video">www.ti.com/video</a>

### TI E2E Community

[e2e.ti.com](http://e2e.ti.com)