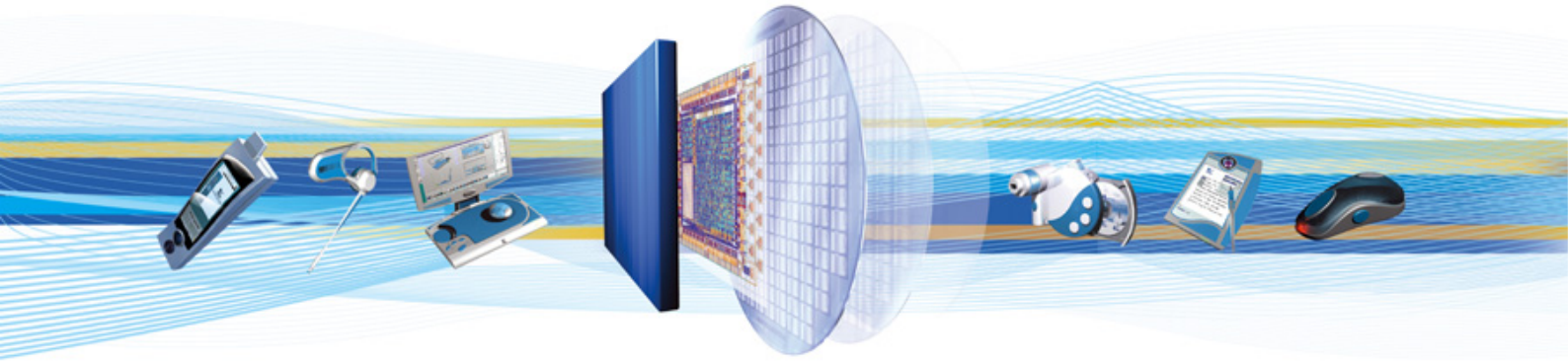
CSR Synergy Bluetooth 18.2.0

Dun Gateway AT

API Description

November 2011



Cambridge Silicon Radio Limited

Churchill House
Cambridge Business Park
Cowley Road
Cambridge CB4 0WZ
United Kingdom

Registered in England and Wales 3665875

Tel: +44 (0)1223 692000
Fax: +44 (0)1223 692001
www.csr.com

Contents

List of Figures

List of Tables

1 Introduction

1.1 Introduction and Scope

This document describes the message interface provided by the gateway part of the Dial-Up Profile as specified in [DUN]. The DUN GW in this document has dialing and control functionality included.

- The dialling and control functionality in DUN is implemented in a separate component called the AT-Interpreter. The AT-Interpreter is put on top of the DUN GW and all communication with the application is going through the AT-Interpreter. The application will therefore see the AT-Interpreter as a Dial up Network gateway with AT-Interpreter functionality. For convenience purpose the two components (DUN GW and AT-Interpreter) are in the following described as one component called CSR_BT_AT_DG. The AT-Interpreter is responsible for interpreting the AT-Commands that are used for setting up the modem connection
- The DUN GW is also available without the dialling and control functionality [DUN GW API]. The DUN GW without the dialling and control functionality will often be used if this functionality is already implemented in an existing application

The AT-Commands requirements are specified in [ITU-T V.250].

1.2 Assumptions

The following assumptions and preconditions are made in the following:

- There is a secure and reliable transport between the profile part, i.e. DUN gateway and the application
- Only one instance of the DUN gateway is active at any time

2 Description

2.1 Introduction

The AT-DG profile manager supplies the interface for the applications that should provide Dial-Up Networking functionality and conform to the gateway part of the Dial-Up Network profile. The AT-DG profile manager is implemented as specified in the Dial-Up Network Profile (K-7) [DUN].

The DUN profile layer provides functionality for:

- Establishing and maintaining a channel between the DUN profile manager and a terminal, which conforms to the terminal part of the Dial-Up Network profile in part K-7 of [DUN]
- Data from the terminal is interpreted and acted upon or concerted to a dedicated primitive for that specific command. To see a list of all supported AT-Commands and Result Codes please refer to part K-7, table 4.1 and 4.2 of [DUN]

2.2 Reference Model

2.2.1 Basic Model

The Dial-Up gateway functionality is based on the models outlined in Figure 1. The AT-Handler is responsible for interpretation of the AT-Commands and is thus extending the API in the sense that specific primitives are used for certain AT-Commands. This implies that the data from the terminal is intercepted by the AT-Handler and acted upon or converted to a dedicated primitive for that specific command.

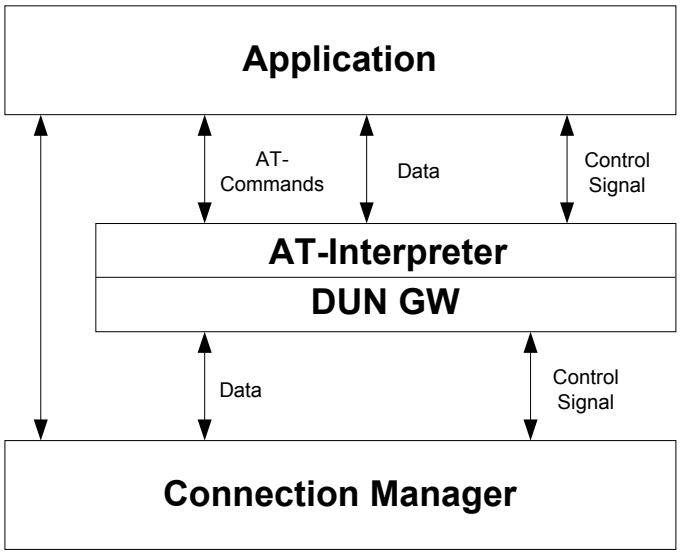


Figure 1: Basic DUN GW reference model

Figure 1 illustrates that the application can either receive control signals, such as modem control signal and connection management signals, AT-Commands or data. Depending on which state the AT-Interpreter is in data received will be processed in different ways.

2.3 Sequence Overview

A normal scenario is that the DUN gateway, usually a phone, will pair with a terminal, and hence create a bond between the pair of devices, in advance using the DUN functionality.

However, if this is not the case, the security manager will, depending on the security level, request the passkey from the application before accepting the connection. The bonding procedure is described in [SC].

A typical user scenario has the following phases:

- Bluetooth® connection set-up
- Network/modem connection set-up (AT-Commands)
- Data transfer
- Connection release (both Bluetooth® and modem)

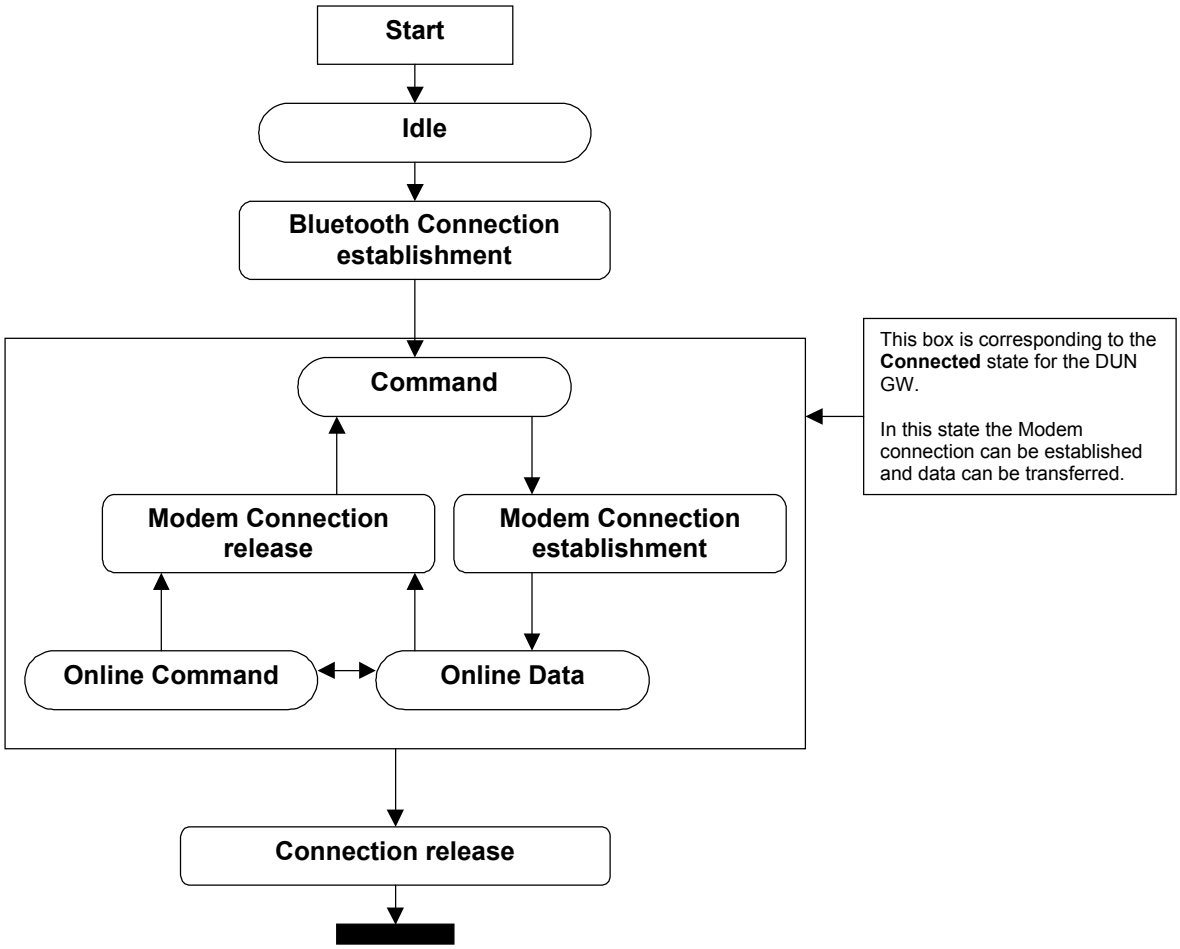


Figure 2: Sequence overview

The Bluetooth® connection will be established first. According to [DUN], the Bluetooth® connection is always initiated from the Data-Terminal side; this is described in the connection establishment (section 3.1.1).

When the Bluetooth® connection is established, the CSR_BT_AT_DG will enter the command state and be ready to receive data packages including AT-Commands from the Data terminal. In this state all data received will be interpreted as AT-Commands. When the modem connection is established, the CSR_BT_AT_DG will enter the online Data state where all data will be transferred directly to the application. If any changes are needed while the modem is online, the online command state is entered where data again will be interpreted as AT-Commands.

3 Interface Description

3.1 Connection Management

3.1.1 Connection Establishment

Connections are established from the terminal side, typically a laptop or PDA. The DUN profile manager and the local device must be set in a mode where incoming connections are accepted. This is accomplished by issuing the CSR_BT_AT_DG_ACTIVATE_REQ from the application towards the DUN profile manager. After sending the CSR_BT_AT_DG_ACTIVATE_REQ the local device is connectable.

Please note that whether or not the Bluetooth device will be discoverable, i.e. can be found by other Bluetooth devices, it must be controlled by the application. For more information, please refer to [CM]. After initialization of CSR Synergy Bluetooth the Bluetooth® device is set up to be discoverable.

The DUN profile manager remains in the idle state until the service is Deactivate from the application or the time expires.

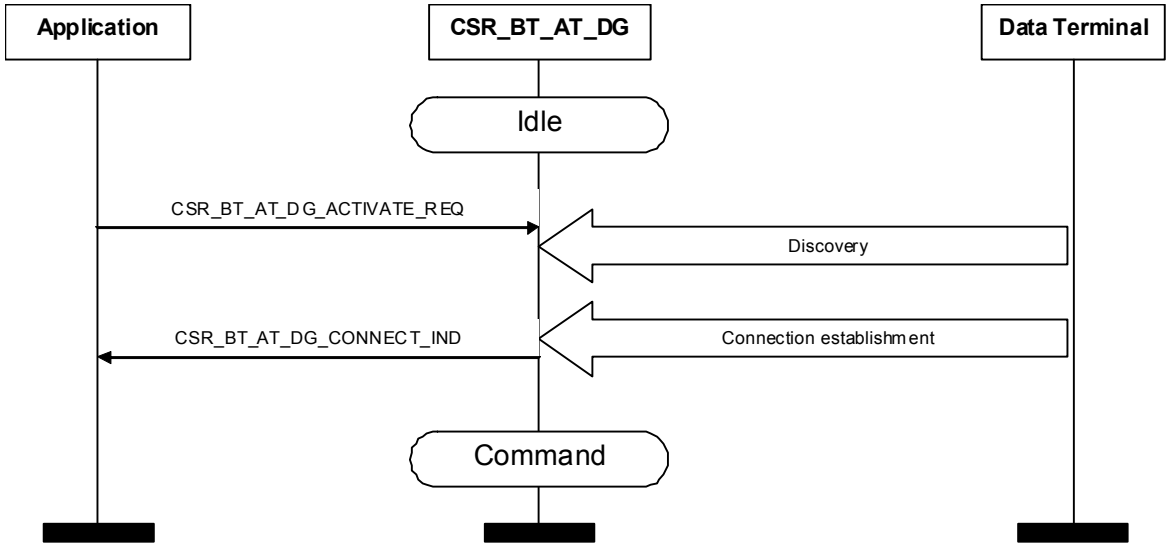


Figure 3: Connection establishment sequence

Before the DUN gateway will accept incoming connections a bond between the two devices must exist. If this is not the case the Security Control will ensure that the passkey is requested from the application, see [SC]. The CSR_BT_AT_DG_CONNECT_IND is an indication that the connection is established, authenticated and encryption is enabled for any subsequent data transfer.

3.1.2 Service Deactivation

If the application decides that the DUN service should no longer be connectable, the application may send a CSR_BT_AT_DG_DEACTIVATE_REQ and get a CSR_BT_AT_DG_DEACTIVATE_CFM return when deactivated.

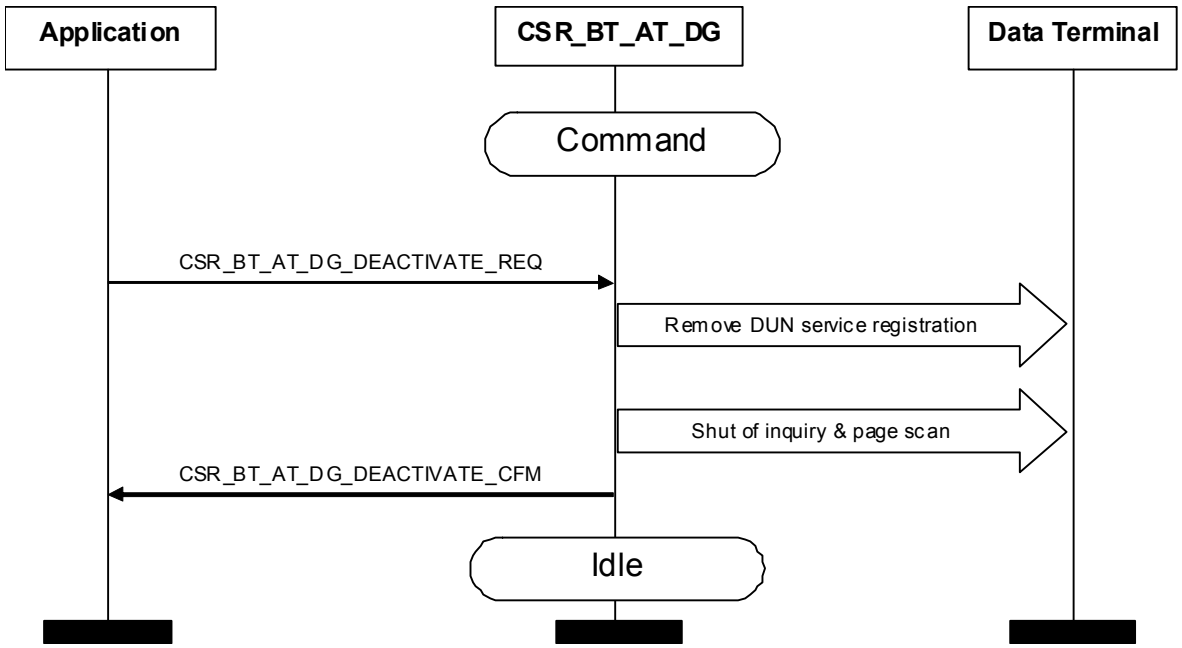


Figure 4: Connection establishment sequence

3.2 Connection Release

3.2.1 Normal Connection Release

The application layer may decide to release the established connection at any time. Release is done by sending a CSR_BT_AT_DG_DISCONNECT_REQ to the DUN profile manager.

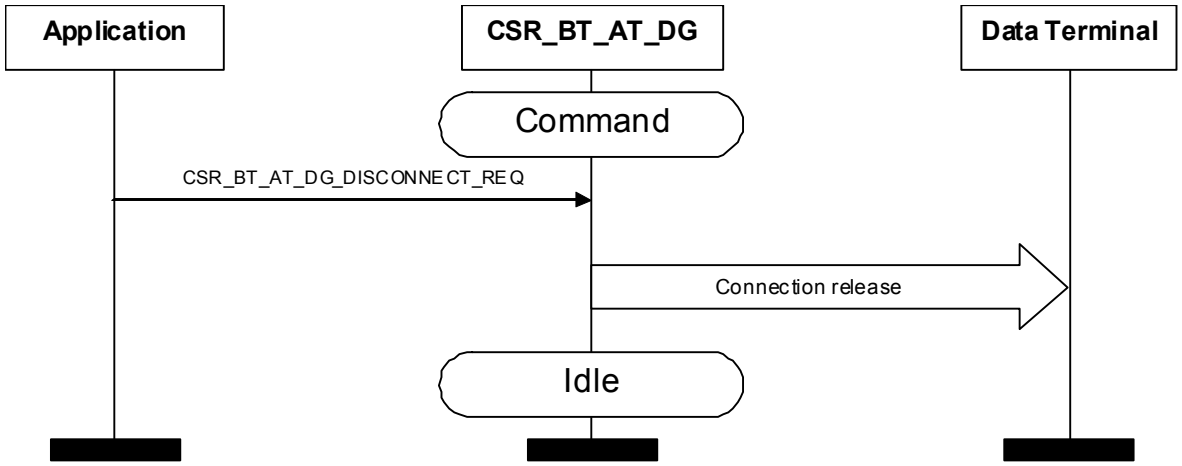


Figure 5: Normal connection release sequence

Depending on the actual state of the connections between the local and remote device, the physical link may be released as a consequence of the disconnection.

3.2.2 Abnormal Connection Release

The connected terminal at the remote end may at any time release the connection. Further, it is possible that the physical link is closed due to an abnormal situation, e.g. radio interference or the devices getting out of coverage from each other.

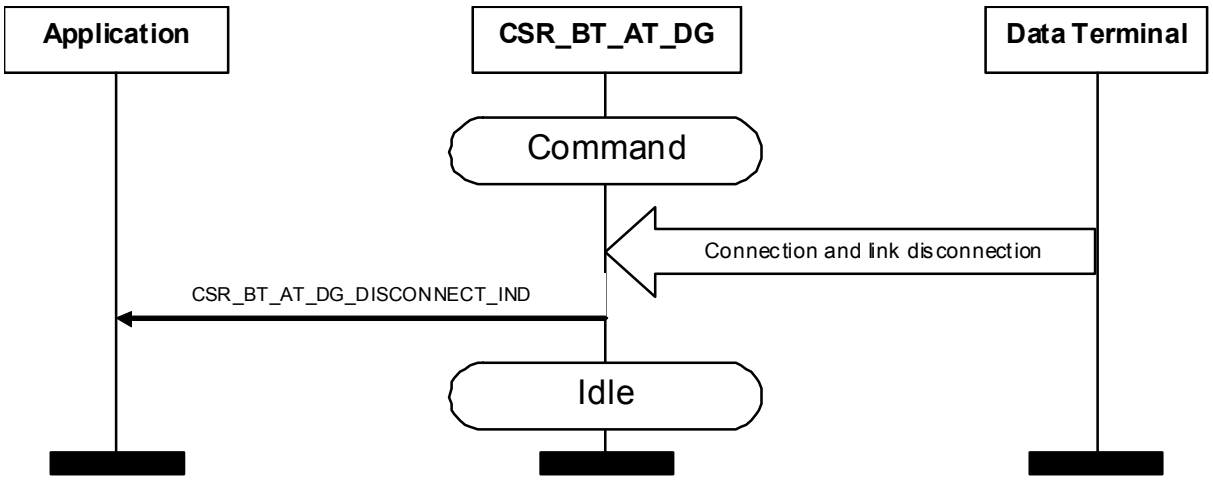


Figure 6: Abnormal connection release sequence

The DUN profile manager will handle the disconnect signal in a similar manner no matter the reason for the release. A new CSR_BT_AT_DG_ACTIVATE_REQ may be issued to make the service available again.

3.3 Data Transfer

In online data state the DUN gateway can transmit and receive data. The data sent and received are not interpreted by the DUN gateway.

3.3.1 Upstream Data

Data can be received from the peer Bluetooth® device and is forwarded to the application. Received data is forwarded using the CSR_BT_AT_DG_DATA_IND signal with a reference to the data payload.

Due to efficiency reasons it is preferred to apply credit based flow control between the local and remote device when data is transmitted.

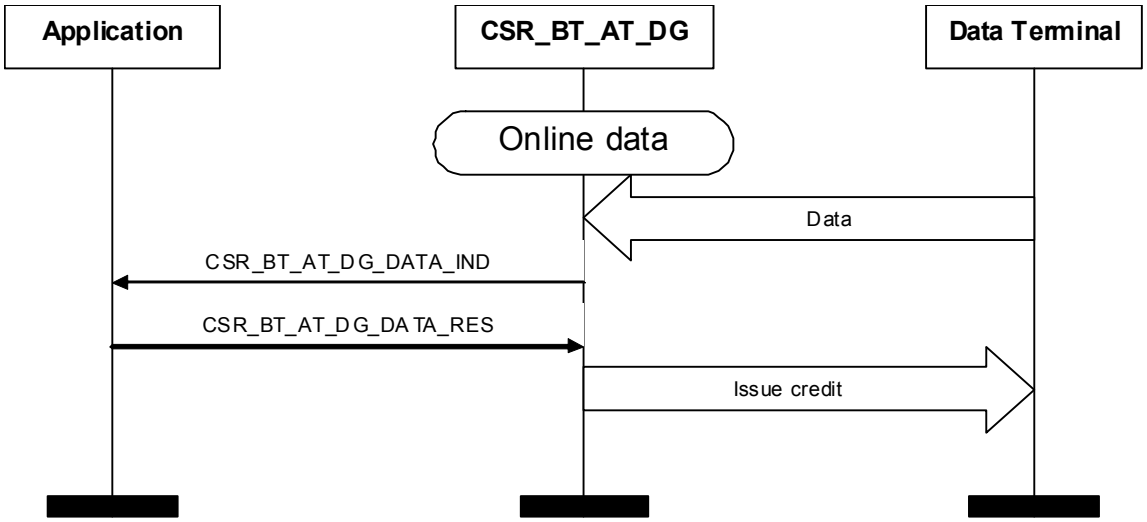


Figure 7: Upstream data transfer

Using credit based flow control implies that the Bluetooth® stack needs to issue credits back to the data terminal side when the received data is consumed and more data can be received. The CSR_BT_AT_DG_DATA_RES must be used by the application layer to inform the DUN profile manager that data is handled and that the application layer is able to receive more data payload. Thus, as long as the CSR_BT_AT_DG_DATA_RES has not been sent to the DUN profile manager no more data packets will be sent towards the application layer. It is the responsibility of the application layer to release the data payload when found appropriate by the application layer.

Once the CSR_BT_AT_DG_DATA_RES is sent the DUN profile manager will ensure that a credit information package is issued to the data terminal side.

3.3.2 Downstream Data

Using the CSR_BT_AT_DG_DATA_REQ signal the application layer can send data payload to the remote device. The CSR_BT_AT_DG_DATA_REQ must include a reference to the data payload to be issued.

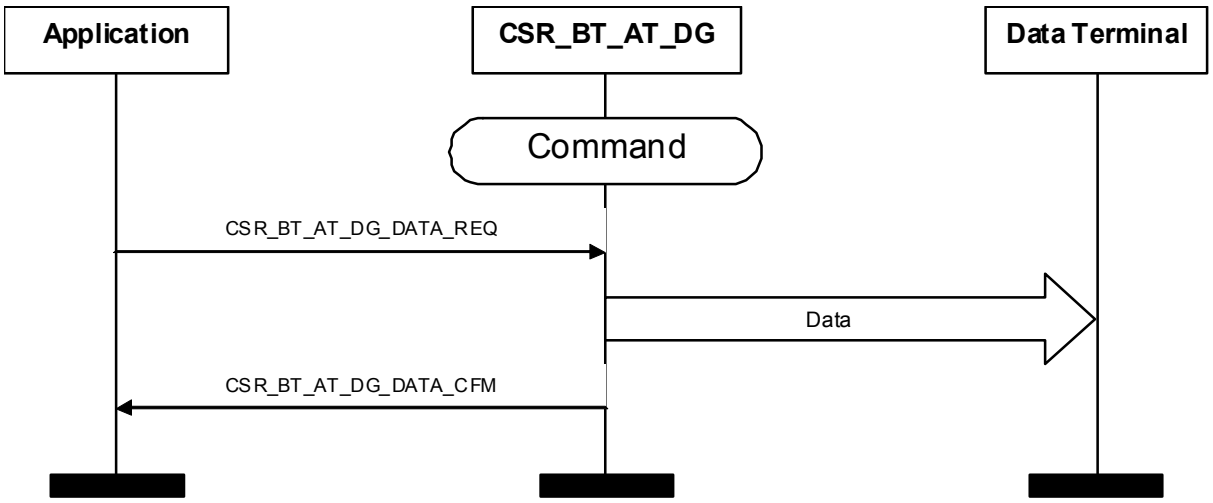


Figure 8: Downstream data

The DUN will forward the received data to the lower layers of the Bluetooth® stack for transmission to the terminal side. Once the data is sent to the lower layers a CSR_BT_AT_DG_DATA_CFM is issued to the application layer. Receiving the CSR_BT_AT_DG_DATA_CFM does not necessarily imply that the data has been received on the terminal side but only that the profile manager is capable of receiving more data.

Please note that the application layer is only allowed to send one data signal, i.e. one CSR_BT_AT_DG_DATA_REQ without receiving a CSR_BT_AT_DG_DATA_CFM and that the profile manager is responsible for the data payload reference; i.e. the application layer must not reuse or release the data payload memory.

3.4 Modem Control Signal

The modem line status can be controlled and read from the application.

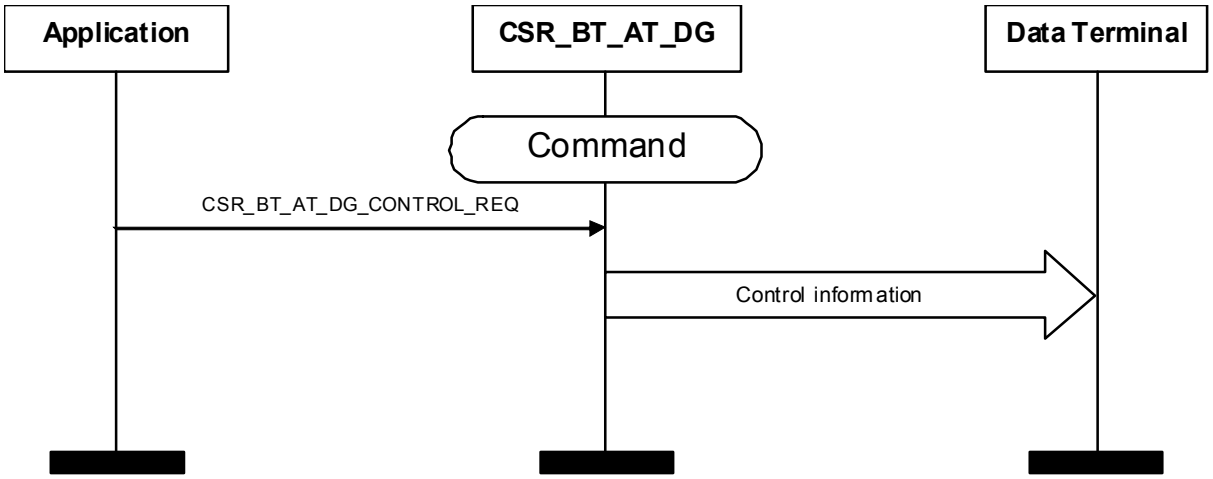


Figure 9: Set the modem status

Furthermore, it is possible to read state of the remote device.

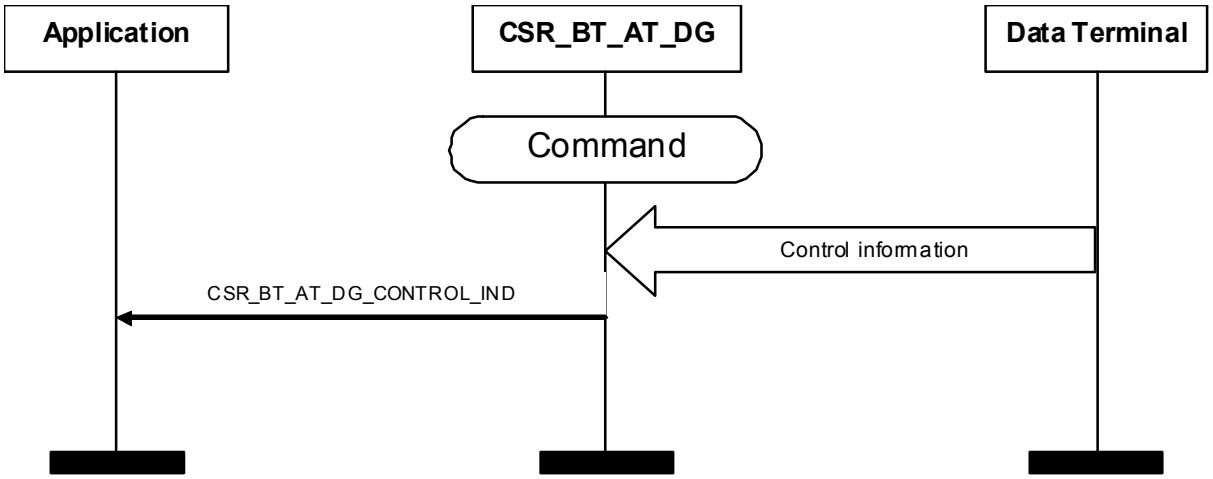


Figure 10: Read the modem status

3.5 AT-Commands

After having established the connection by the remote terminal, the DUN gateway is placed in a state where it is ready to interpret AT-Commands. CSR_BT_AT_DG_ATC_XX_IND represents the individual AT-Commands being supported by the AT-Command interpreter, and CSR_BT_AT_DG_ATC_XX_RES represents the response, which the application is expected to return. For more information about the commands and their define values, please refer to section 4.9 and 4.10.

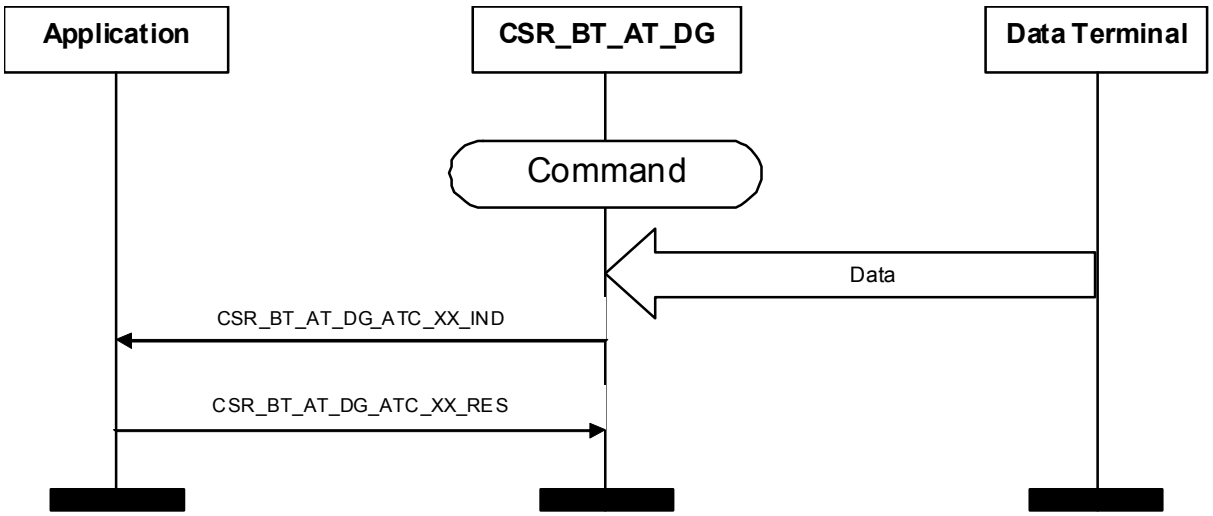


Figure 11: Data from the data terminal is interpreted as AT-Commands

In [ITU-T V.250] it is defined that responses consist of three parts: A header, text or result text, and a trailer. The characters transmitted for the header and trailer, are determined by a user setting, and handled by the AT-Handler. All AT-Commands are converted to primitives with the naming convention: CSR_BT_AT_DG_ATC_XX. **Please note that these signals do not follow the normal IND/RES terminology. An indication (IND) will always be followed by a response (RES), but the response will not always have the same name.** The following section shows what the correct response is to the indication.

3.5.1 Correct Response Signal for Each Indication Signal

Table 1 below points out the correct response signal for each indication signal. If a signal is marked with two responses it is up to the application to decide which one is used, depending on the parameter settings, see [ITU-T V.250]. The header file csr_bt_at_usr_config.h contains settings that may be changed at compile time for different defines used throughout the AT component. If it is desired to make the above application simpler, it is also possible to let the AT module give direct response signal to some indication signal, by changing settings in csr_bt_at_usr_config.h.

Response Primitives / Indication Primitives	CSR_BT_AT_DG_ATC_RESULT_CODE_RES	CSR_BT_AT_DG_ATC_CONNECT_TEXT_RES	CSR_BT_AT_DG_ATC_DEFAULT_SETTING_RES	CSR_BT_AT_DG_ATC_FACTORY_SETTING_RES	CSR_BT_AT_DG_ATC_GMI_RES	CSR_BT_AT_DG_ATC_GMM_RES	CSR_BT_AT_DG_ATC_GMR_RES	CSR_BT_AT_DG_ATC_GCAP_RES	CSR_BT_AT_DG_ATC_S0Q_RES	CSR_BT_AT_DG_ATC_S3Q_RES	CSR_BT_AT_DG_ATC_S4Q_RES	CSR_BT_AT_DG_ATC_S5Q_RES	CSR_BT_AT_DG_ATC_S6Q_RES	CSR_BT_AT_DG_ATC_S7Q_RES	CSR_BT_AT_DG_ATC_S8Q_RES	CSR_BT_AT_DG_ATC_S10Q_RES	CSR_BT_AT_DG_ATC_UNKNOWN_EXTENDED_CMD_RES
CSR_BT_AT_DG_ATC_AND_D_IND	✓																
CSR_BT_AT_DG_ATC_AND_F_IND				✓													
CSR_BT_AT_DG_ATC_AND_C_IND	✓																
CSR_BT_AT_DG_ATC_L_IND	✓																
CSR_BT_AT_DG_ATC_M_IND	✓																
CSR_BT_AT_DG_ATC_O_IND	✓	✓															
CSR_BT_AT_DG_ATC_X_IND	✓																
CSR_BT_AT_DG_ATC_Z_IND			✓														
CSR_BT_AT_DG_ATC_S0_IND	✓																
CSR_BT_AT_DG_ATC_S3_IND	✓																
CSR_BT_AT_DG_ATC_S4_IND	✓																
CSR_BT_AT_DG_ATC_S5_IND	✓																
CSR_BT_AT_DG_ATC_S6_IND	✓																
CSR_BT_AT_DG_ATC_S7_IND	✓																
CSR_BT_AT_DG_ATC_S8_IND	✓																
CSR_BT_AT_DG_ATC_S10_IND	✓																
CSR_BT_AT_DG_ATC_E_IND	✓																
CSR_BT_AT_DG_ATC_Q_IND	✓																
CSR_BT_AT_DG_ATC_V_IND	✓																
CSR_BT_AT_DG_ATC_P_IND	✓																
CSR_BT_AT_DG_ATC_H_IND	✓																
CSR_BT_AT_DG_ATC_T_IND	✓																
CSR_BT_AT_DG_ATC_A_IND	✓	✓															
CSR_BT_AT_DG_ATC_S0Q_IND									✓								
CSR_BT_AT_DG_ATC_S3Q_IND										✓							
CSR_BT_AT_DG_ATC_S4Q_IND											✓						

Response Primitives / Indication Primitives	CSR_BT_AT_DG_ATC_RESULT_CODE_RES	CSR_BT_AT_DG_ATC_CONNECT_TEXT_RES	CSR_BT_AT_DG_ATC_DEFAULT_SETTING_RES	CSR_BT_AT_DG_ATC_FACTORY_SETTING_RES	CSR_BT_AT_DG_ATC_GMI_RES	CSR_BT_AT_DG_ATC_GMM_RES	CSR_BT_AT_DG_ATC_GMR_RES	CSR_BT_AT_DG_ATC_GCAP_RES	CSR_BT_AT_DG_ATC_S0Q_RES	CSR_BT_AT_DG_ATC_S3Q_RES	CSR_BT_AT_DG_ATC_S4Q_RES	CSR_BT_AT_DG_ATC_S5Q_RES	CSR_BT_AT_DG_ATC_S6Q_RES	CSR_BT_AT_DG_ATC_S7Q_RES	CSR_BT_AT_DG_ATC_S8Q_RES	CSR_BT_AT_DG_ATC_S10Q_RES	CSR_BT_AT_DG_ATC_UNKNOWN_EXTENDED_CMD_RES
CSR_BT_AT_DG_ATC_S5Q_IND												✓					
CSR_BT_AT_DG_ATC_S6Q_IND													✓				
CSR_BT_AT_DG_ATC_S7Q_IND														✓			
CSR_BT_AT_DG_ATC_S8Q_IND															✓		
CSR_BT_AT_DG_ATC_S10Q_IND																✓	
CSR_BT_AT_DG_ATC_GMI_IND					✓												
CSR_BT_AT_DG_ATC_GMM_IND						✓											
CSR_BT_AT_DG_ATC_GMR_IND							✓										
CSR_BT_AT_DG_ATC_GCAP_IND								✓									
CSR_BT_AT_DG_ATC_ABORT_IND	✓																
CSR_BT_AT_DG_ATC_D_IND	✓	✓															
CSR_BT_AT_DG_ATC_UNKNOWN_EXTENDED_CMD_IND	✓																✓

Table 1: Correct response to an indication

4 CSR_BT_AT_DG Primitives

This section gives an overview of the primitives and parameters in the interface. Detailed information can be found in the corresponding csr_bt_at_prim.h file.

4.1 List of CSR_BT_AT_DG Primitives

Bluetooth® Connect and Management Primitives	**Reference**
CSR_BT_AT_DG_ACTIVATE_REQ	See section 4.2
CSR_BT_AT_DG_DEACTIVATE_REQ	See section 4.3
CSR_BT_AT_DG_DEACTIVATE_CFM	See section 4.3
CSR_BT_AT_DG_CONNECT_IND	See section 4.4
CSR_BT_AT_DG_PORTNEG_IND	See section 4.7
CSR_BT_AT_DG_PORTNEG_IND	See section 4.7
CSR_BT_AT_DG_DISCONNECT_REQ	See section 4.8
CSR_BT_AT_DG_DISCONNECT_IND	See section 4.8
CSR_BT_AT_DG_DATA_PATH_STATUS_IND	See section 4.11

Online Data Primitives	**Reference**
CSR_BT_AT_DG_DATA_REQ	See section 4.5
CSR_BT_AT_DG_DATA_CFM	See section 4.5
CSR_BT_AT_DG_DATA_IND	See section 4.5
CSR_BT_AT_DG_DATA_RES	See section 4.5

Modem Control Primitives	**Reference**
CSR_BT_AT_DG_CONTROL_REQ	See section 4.6
CSR_BT_AT_DG_CONTROL_IND	See section 4.6
CSR_BT_AT_DG_ ATC_XX_IND	See section 4.9
CSR_BT_AT_DG_ ATC_XX_RES	See section 4.10

Table 2: CSR_BT_AT_DG Primitives

4.2 CSR_BT_AT_DG_ACTIVATE

Primitives \ Parameters	type	phandle	timeout
CSR_BT_AT_DG_ACTIVATE_REQ	✓	✓	✓

Table 3: CSR_BT_AT_DG_ACTIVATE Primitive

Description

This signal is used for activating a service and making it connectable. The process includes:

- Enabling page scan

The service availability can be time limited or may run forever.

Parameters

type	Signal identity, CSR_BT_AT_DG_ACTIVATE_REQ.
phandle	The identity of the calling process. It is possible to initiate the procedure by any higher layer process as the response is returned to phandle.
timeout	The time in number of seconds being the service should be available for remote connections. The maximum Timeout time allowed is defined by the CSR_BT_MAX_TIME in the file “csr_bt_profiles.h”. A timeout value of 0 (zero) or the define CSR_BT_INFINITE_TIME in the file “csr_bt_profiles.h” indicates an infinity timeout time. When a connection is made, the service available stops.

4.3 CSR_BT_AT_DG_DEACTIVATE

Primitives \ Parameters	type	resultCode	resultSupplier
CSR_BT_AT_DG_DEACTIVATE_REQ	✓		
CSR_BT_AT_DG_DEACTIVATE_CFM	✓	✓	✓

Table 4: CSR_BT_AT_DG_DEACTIVATE Primitives

Description

Deactivate a service that has been activated previously. The service will no longer be connectable.

Parameters

type	Signal identity, CSR_BT_AT_DG_DEACTIVATE_REQ/CFM.
resultCode	The result code of the operation. Possible values depends on the value of resultSupplier. If e.g. the resultSupplier == CSR_BT_SUPPLIER_CM then the possible result codes can be found in csr_bt_cm_prim.h. All values which are currently not specified in the respective prim.h file are regarded as reserved and the application should consider them as errors.
resultSupplier	This parameter specifies the supplier of the result given in resultCode. Possible values can be found in csr_bt_result.h

4.4 CSR_BT_AT_DG_CONNECT

Primitives \ Parameters	type	serverChannel	profileMaxFrameSize	resultCode	resultSupplier	btConnId
CSR_BT_AT_DG_CONNECT_IND	✓	✓	✓	✓	✓	✓

Table 5: CSR_BT_AT_DG_CONNECT Primitives

Description

A connection has been established.

Parameters

Type	Signal identity, CSR_BT_AT_DG_CONNECT_IND.
serverChannel	The local server number, which is a reference ID used by the Connection Manager.
profileMaxFrameSize	The maximum payload length allowed in downstream data requests. Please note that it is possible to receive one octet (byte) more than indicated in the profileMaxFrameSize parameter.
resultCode	The result code of the operation. Possible values depend on the value of resultSupplier. If e.g. the resultSupplier == CSR_BT_SUPPLIER_CM then the possible result codes can be found in csr_bt_cm_prim.h. All values which are currently not specified in the respective prim.h file are regarded as reserved and the application should consider them as errors.
resultSupplier	This parameter specifies the supplier of the result given in resultCode. Possible values can be found in csr_bt_result.h.
btConnId	Connection Identifier

4.5 CSR_BT_AT_DG_DATA

Primitives \ Parameters	type	btConnId	payloadLength	*payload
CSR_BT_AT_DG_DATA_REQ	✓	✓	✓	✓
CSR_BT_AT_DG_DATA_CFM	✓	✓		
CSR_BT_AT_DG_DATA_IND	✓	✓	✓	✓
CSR_BT_AT_DG_DATA_RES	✓	✓		

Table 6: CSR_BT_AT_DG_DATA Primitives

Description

Send and receive data.

Parameters

type	Signal identity, CSR_BT_AT_DG_DATA_REQ/CFM/IND/RES.
btConnId	Connection Identifer
payloadLength	Length of data in number of octets.
*payload	Reference to the actual payload.

4.6 CSR_BT_AT_DG_CONTROL

Primitives \ Parameters	type	btConnId	modemstatus	break_signal
CSR_BT_AT_DG_CONTROL_REQ	✓	✓	✓	✓
CSR_BT_AT_DG_CONTROL_IND	✓	✓	✓	✓

Table 7: CSR_BT_AT_DG_CONTROL Primitives

Description

Send and receive modem status information.

Parameters

type — Signal identity, CSR_BT_AT_DG_CONTROL_REQ/IND.

btConnId — Connection Identifier

modemstatus — The modem status contains the following bit from the RS-232 interface:

Bit 0 CTS (Clear To Send)
Bit 1 RTS (Request To Send)
Bit 2 DSR (Data Set Ready)
Bit 3 DTR (Data Terminal Ready)
Bit 4 RI (Ring Indicator)
Bit 5 DCD (Data Carrier Detect)

There is mask code for this bit in the csr_bt_profiles.h.

break_signal — The `break_signal` is encoded as follows:

Bit 0 Not used.
Bit 1 0: No break signal encoded. 1: Break signal encoded.
Bit 2 Not used.
Bit 3 Not used.
Bit 4-7 Duration of break signal in 200mS increments.

4.7 CSR_BT_AT_DG_PORTNEG

Primitives \ Parameters	Type	btConnId	portPar	request
CSR_BT_AT_DG_PORTNEG_IND	✓	✓	✓	✓
CSR_BT_AT_DG_PORTNEG_RES	✓	✓	✓	

Table 8: CSR_BT_AT_DG_PORTNEG Primitives

Description

Send and receive port set-up information.

Parameters

type — Signal identity, CSR_BT_AT_DG_PORTNEG_IND/RES.

btConnId — Connection Identifier

portPar — The portPar is a structure defined as RFC_PORTNEG_VALUES_T. The RFC_PORTNEG_VALUES_T structure, shown below, is included into the PORTNEG primitive. In the library function call the RFC_PORTNEG_VALUES_T structure should be called as a pointer and the library function will copy the data into the PORTNEG primitive.

Request — TRUE: If the request is TRUE, the remote device requests the local RFC_PORTNEG_VALUES_T settings. The receiver must respond with current RFC_PORTNEG_VALUES_T settings.

FALSE: The remote device must confirm the settings or propose new ones. Setting new suggested values includes setting also the proper parameter mask.

```
typedef struct
{
   CsrUint8     baud_rate;      /* port speed indicator - see #defines above */
   CsrUint8     data_bits;      /* DATA_BITS_5, _6, _7 or _8 - see above */
   CsrUint8     stop_bits;      /* STOP_BITS_ONE or _ONE_AND_A_HALF - see above */
   CsrUint8     parity;         /* PARITY_OFF or PARITY_ON */
   CsrUint8     parity_type;    /* PARITY_TYPE_ODD, _EVEN, _MARK or _SPACE */
   CsrUint8     flow_ctrl_mask; /* 6 bits - use FLC_ #defines above (see 07.10) */
   CsrUint8     xon;            /* xon character  (default DC1 0x11) */
   CsrUint8     xoff;           /* xoff character (default DC3 0x13) */
   CsrUint16    parameter_mask; /* 16 bits (top two reserved) see PM_ #defines */
} RFC_PORTNEG_VALUES_T;
```

Baud_rate	Takes the form RFC_xxxx_BAUD, where `xx` is the port speed in bits per second. Encoded as: 0x00 = 2400 0x01 = 4800 0x02 = 7200 0x03 = 9600

	0x04 = 19200 0x05 = 38400 0x06 = 57600 0x07 = 115200 0x08 = 230400 0xFF = RFC_UNKNOWN_BAUD
data_bits	Number of data bits encoded as an unsigned integer. Valid values are 5, 6, 7 and 8. Encoded as: 0x00 = 5 bit 0x02 = 6 bit 0x01 = 7 bit 0x03 = 8 bit
stop_bits	Encoded as: 0x00 = 1 stop bit 0x01 = 1.5 stop bits
Parity	Encoded as: 0x00 = PARITY_OFF 0x01 = PARITY_ON
parity_type	Encoded as: 0x00 odd parity 0x02 even parity 0x01 mark parity 0x03 space parity
flow_ctrl_mask	Encoded as: Bit 0 XON / XOFF, input Bit 1 XON / XOFF, output Bit 2 RTR input Bit 3 RTR output Bit 4 RTC input Bit 5 RTC output
Xon	Xon character (default DC1, 0x11)
Xoff	Xoff character (default DC3, 0x13)
parameter_mask	`parameter_mask` is used for indicating which parameters in the Remote Port Negotiation command is negotiate able. For a command, `parameter_mask` is interpreted as follows: 0 no change 1 change For a response, `parameter_mask` is interpreted as follows: 0 not accepted / not supported 1 accepted proposal - the new values are used The bit mask is shown as: Bit0 bit rate Bit1 data bits Bit2 stop bits Bit3 Parity Bit4 parity type

	Bit5 XON character Bit6 XOFF character Bit7 Reserved Bit8 XON / XOFF, input Bit9 XON / XOFF, output Bit10 RTR input Bit11 RTR output Bit12 RTC input Bit13 RTC output Bit14 Reserved, set 0 by sender Bit15 Reserved, set 0 by sender

4.8 CSR_BT_AT_DG_DISCONNECT

Primitives \ Parameters	type	btConnId	localTerminated	reasonCode	reasonSupplier
CSR_BT_AT_DG_DISCONNECT_REQ	✓	✓			
CSR_BT_AT_DG_DISCONNECT_IND	✓	✓	✓	✓	✓

Table 9: CSR_BT_AT_DG_DISCONNECT Primitives

Description

Release a connection between the local and the remote device. Depending on the number of connections and other active services, the link may or may not be released.

Parameters

type	Signal identity, CSR_BT_AT_DG_DISCONNECT_REQ/IND.
btConnId	Connection Identifier
localTerminated	TRUE if termination of connection happened on request from the local host; FALSE otherwise.
reasonCode	The reason code of the operation. Possible values depend on the value of reasonSupplier. If e.g. the reasonSupplier == CSR_BT_SUPPLIER_CM then the possible reason codes can be found in csr_bt_cm_prim.h. All values which are currently not specified in the respective prim.h files are regarded as reserved and the application should consider them as errors.
reasonSupplier	This parameter specifies the supplier of the reason given in reasonCode. Possible values can be found in csr_bt_result.h

4.9 CSR_BT_AT_DG_ATC_XX_IND

Table 10 below shows the parameters for each AT-Command indication.

Parameters / Primitives	type	value	payload Length	*payload	Description
CSR_BT_AT_DG_ATC_AND_D_IND	✓	✓			Circuit 108 behaviour
CSR_BT_AT_DG_ATC_AND_F_IND	✓	✓			Set to factory defined configuration
CSR_BT_AT_DG_ATC_AND_C_IND	✓	✓			Circuit 109 behaviour
CSR_BT_AT_DG_ATC_L_IND	✓	✓			Monitor Speaker Loudness
CSR_BT_AT_DG_ATC_M_IND	✓	✓			Monitor Speaker Mode
CSR_BT_AT_DG_ATC_O_IND	✓	✓			Return to Online Data State
CSR_BT_AT_DG_ATC_X_IND	✓	✓			Result Code Selection and Call Progress Monitoring Control
CSR_BT_AT_DG_ATC_Z_IND	✓	✓			Reset to default configuration
CSR_BT_AT_DG_ATC_S0_IND	✓	✓			Set Automatic Answer
CSR_BT_AT_DG_ATC_S3_IND	✓	✓			Set Command Line Terminator Character
CSR_BT_AT_DG_ATC_S4_IND	✓	✓			Set Response Formatting Character
CSR_BT_AT_DG_ATC_S5_IND	✓	✓			Set Command Line Editing Character
CSR_BT_AT_DG_ATC_S6_IND	✓	✓			Set Pause Before Blind Dialing
CSR_BT_AT_DG_ATC_S7_IND	✓	✓			Set Connection Completion Timeout
CSR_BT_AT_DG_ATC_S8_IND	✓	✓			Set Comma Dial Modifier Time
CSR_BT_AT_DG_ATC_S10_IND	✓	✓			Set Automatic Disconnect Delay
CSR_BT_AT_DG_ATC_E_IND	✓	✓			Command Echo
CSR_BT_AT_DG_ATC_Q_IND	✓	✓			Result Code Suppression
CSR_BT_AT_DG_ATC_V_IND	✓	✓			DCE Response Format
CSR_BT_AT_DG_ATC_P_IND	✓				Select Pulse Dialing
CSR_BT_AT_DG_ATC_H_IND	✓				Hook control
CSR_BT_AT_DG_ATC_T_IND	✓				Select Tone Dialing
CSR_BT_AT_DG_ATC_A_IND	✓				Answer
CSR_BT_AT_DG_ATC_S0Q_IND	✓				Read Automatic Answer value
CSR_BT_AT_DG_ATC_S3Q_IND	✓				Read Command Line Terminator Character value
CSR_BT_AT_DG_ATC_S4Q_IND	✓				Read Response Formatting Character value
CSR_BT_AT_DG_ATC_S5Q_IND	✓				Read Command Line Editing Character value
CSR_BT_AT_DG_ATC_S6Q_IND	✓				Read Pause Before Blind Dialling value
CSR_BT_AT_DG_ATC_S7Q_IND	✓				Read Connection Completion Timeout value
CSR_BT_AT_DG_ATC_S8Q_IND	✓				Read Comma Dial Modifier Time value
CSR_BT_AT_DG_ATC_S10Q_IND	✓				Read Automatic Disconnect Delay value
CSR_BT_AT_DG_ATC_GMI_IND	✓				Request Manufacturer Identification
CSR_BT_AT_DG_ATC_GMM_IND	✓				Request Model Identification
CSR_BT_AT_DG_ATC_GMR_IND	✓				Request Revision Identification
CSR_BT_AT_DG_ATC_GCAP_IND	✓				Request Complete Capabilities list
CSR_BT_AT_DG_ATC_ABORT_IND	✓				Abort current AT Command
CSR_BT_AT_DG_ATC_D_IND	✓	✓	✓	✓	Dial
CSR_BT_AT_DG_ATC_UNKNOWN_EXTENDED_CMD_IND	✓		✓	✓	Interpret an unknown extended command

Table 10: Parameters for CSR_BT_AT_DG_ATC_XX_IND signals

Description

The CSR_BT_AT_DG is the Data Communication Endpoint (DCE) for the modem connection. In command state and online command state the DCE is ready to accept AT-Commands. These Command Primitives represent the individual AT-Commands that are supported by the AT-Command interpreter. For more information about the commands and their define values, please refer to [ITU-T V.250].

Parameters

type	Signal identity.
value	Representing a decimal integer parameter value, for the command. Except for CSR_BT_AT_DG_ATC_D_IND, where value = 1 means that the dial string is terminated with a semicolon, and value = 0 means the end of the command line.
payloadLength	Number of characters in the dial string.
*payload	Reference to the actual dial string.

4.10 CSR_BT_AT_DG_ATC_XX_RES

Table 11 below shows the parameters for each AT-Command response.

Primitives \ Parameters	type	value	payload Length	* payload	parameterSetting	atResult
CSR_BT_AT_DG_ATC_RESULT_CODE_RES	✓					✓
CSR_BT_AT_DG_ATC_CONNECT_TEXT_RES	✓		✓	✓		✓
CSR_BT_AT_DG_ATC_DEFAULT_SETTING_RES	✓				✓	
CSR_BT_AT_DG_ATC_FACTORY_SETTING_RES	✓				✓	
CSR_BT_AT_DG_ATC_GMI_RES	✓		✓	✓		
CSR_BT_AT_DG_ATC_GMM_RES	✓		✓	✓		
CSR_BT_AT_DG_ATC_GMR_RES	✓		✓	✓		
CSR_BT_AT_DG_ATC_GCAP_RES	✓		✓	✓		
CSR_BT_AT_DG_ATC_S0Q_RES	✓	✓				
CSR_BT_AT_DG_ATC_S3Q_RES	✓	✓				
CSR_BT_AT_DG_ATC_S4Q_RES	✓	✓				
CSR_BT_AT_DG_ATC_S5Q_RES	✓	✓				
CSR_BT_AT_DG_ATC_S6Q_RES	✓	✓				
CSR_BT_AT_DG_ATC_S7Q_RES	✓	✓				
CSR_BT_AT_DG_ATC_S8Q_RES	✓	✓				
CSR_BT_AT_DG_ATC_S10Q_RES	✓	✓				
CSR_BT_AT_DG_ATC_UNKNOWN_EXTENDED_CMD_RES	✓		✓	✓		

Table 11: Parameters for CSR_BT_AT_DG_ATC_XX_RES signals

The total number of characters in the payload shall not exceed 2044. All characters exceeding 2044 will be ignored. Note furthermore that the payload may not contain the sequence “0<CR>” or “OK <CR>”, so that the DTE can avoid false detection. Note that <CR> means the actual value of the S3 parameter, which per default must be CR.

The payload in GCAP shall use the extended syntax result code +<name>s, see [ITU-T V.250].

Description

In command state and online command state, the DCE shall issue responses. Responses consist of three parts: A header, text or result text, and a trailer. The characters transmitted for the header and trailer are determined by a user setting and handled by the AT-Handler. This means that the application layer only needs to send a numeric result code. Table 12: AT Numeric Result Codes, see 0, indicates the numeric result codes, their text equivalents, and a brief description of the use of each. Table 13: AT Result Code, see 0, indicates the basic syntax response commands issuing a CSR_BT_AT_DG_ATC_RESULT_CODE_RES, and also indicates which specific numeric result code each command may issue. Finally, in section 3.5.1, Table 1 points out the correct response signal for each indication signal. For more information about the responses for the individual commands, please refer to [ITU-T V.250].

Parameters

type	Signal identity.
value	Giving the value of the S-parameter, represented as a decimal integer value between 0 and 255.
payloadLength	Number of characters in the payload.
*payload	Reference to the actual payload.
parameterSetting	The values on the following parameters: Q, S3, S4, S5, &C, &D, E and V.
atResult	A result code, represented as a decimal integer value, see Table 13: AT Result Code (section 0), for valid values.

4.10.1 AT Numeric Result Codes

Numeric result code	Text result code	Description
0	OK	Acknowledges execution of a command.
1	**CONNECT**	A connection has been established; the DCE is moving from command state to online data state.
2	RING	The DCE has detected an incoming call from the network.
3	NO CARRIER	The connection has been terminated or the attempt to establish a connection failed.
4	ERROR	Command not recognized, command line exceeded maximum number of commands in one command line, parameter value invalid, or other problem with processing the command line.
6	NO DIAL TONE	No dial tone detected.
7	BUSY	Busy signal detected.
8	NO ANSWER	A “@” dial modifier was used, but remote ringing followed by five seconds of silence was not detected before expiration of the connection timer (S7).

Table 12: AT Numeric Result Codes

In [ITU-T V.250] the specific of each command includes the specific result codes that may be issued in the response (for the legal responses to an indication see section 3.5.1). AT result code 2 (RING) shall be issued by the application when it receives an incoming call.

Primitives \ AT result code	0	1	2	3	4	6	7	8
CSR_BT_AT_DG_ATC_AND_D_IND	✓				✓			
CSR_BT_AT_DG_ATC_AND_C_IND	✓				✓			
CSR_BT_AT_DG_ATC_L_IND	✓							
CSR_BT_AT_DG_ATC_M_IND	✓							
CSR_BT_AT_DG_ATC_O_IND		✓		✓	✓			
CSR_BT_AT_DG_ATC_X_IND	✓				✓			
CSR_BT_AT_DG_ATC_S0_IND	✓				✓			
CSR_BT_AT_DG_ATC_S3_IND	✓				✓			
CSR_BT_AT_DG_ATC_S4_IND	✓				✓			
CSR_BT_AT_DG_ATC_S5_IND	✓				✓			
CSR_BT_AT_DG_ATC_S6_IND	✓				✓			
CSR_BT_AT_DG_ATC_S7_IND	✓				✓			
CSR_BT_AT_DG_ATC_S8_IND	✓				✓			
CSR_BT_AT_DG_ATC_S10_IND	✓				✓			
CSR_BT_AT_DG_ATC_E_IND	✓				✓			
CSR_BT_AT_DG_ATC_Q_IND	✓				✓			
CSR_BT_AT_DG_ATC_V_IND	✓				✓			
CSR_BT_AT_DG_ATC_D_IND	✓	✓		✓	✓	✓	✓	✓
CSR_BT_AT_DG_ATC_H_IND	✓				✓			
CSR_BT_AT_DG_ATC_P_IND	✓							
CSR_BT_AT_DG_ATC_T_IND	✓							
CSR_BT_AT_DG_ATC_A_IND	✓	✓		✓	✓			
CSR_BT_AT_DG_ATC_UNKNOWN_EXTENDED_CMD_IND	✓				✓			

Table 13: AT Result Code

Note: When receiving a CSR_BT_AT_DG_ATC_UNKNOWN_EXTENDED_CMD_IND signal the application can choose between two response signals, e.g. CSR_BT_AT_DG_ATC_UNKNOWN_EXTENDED_CMD_RES and CSR_BT_AT_DG_ATC_RESULT_CODE_RES. CSR_BT_AT_ DG_ATC_RESULT_CODE_RES is used if the response code is an AT Numeric Result Code, e.g. OK or ERROR, while CSR_BT_AT_DG_ATC_UNKNOWN_EXTENDED_CMD_RES is used if the other signal is insufficient.

When using CSR_BT_AT_DG_ATC_UNKNOWN_EXTENDED_CMD_RES the payload may not contain the sequence “0<CR>” or “OK <CR>”, so that the DTE can avoid false detection. Note that <CR> means the actual value of the S3 parameter, which per default must be CR.

4.11 CSR_BT_AT_DG_DATA_PATH_STATUS

Primitives \ Parameters	type	dgInstanceQueue	status
CSR_BT_AT_DG_DATA_PATH_STATUS_IND	✓	✓	✓

Table 14: CSR_BT_AT_DG_DATA_PATH_STATUS Primitive

Description

Notify the application of changes to the DG data path. This signal is sent whenever the AT handler receives a CSR_BT_AT_DG_DATA_PATH_STATUS_IND.

Parameters

type	Signal identity, CSR_BT_AT_DG_DATA_PATH_STATUS_IND.
dgInstanceQueue	The queue ID of the DG instance, which has had its data path status changed.
status	Status of file path (open/closed/lost). See csr_bt_profiles.h.

5 Document References

Document	Reference
Bluetooth® Core Specification v.1.1, v.1.2 and v.2.0	[BT]
The Bluetooth® Specification, the Dial-Up Network Profile, profile section K:7	[DUN]
CSR Synergy Bluetooth, SC – Security Controller API Description	[SC]
Telecommunication Standardization Sector of ITU, profile section (05/99)	[ITU-T V.250]
api-0104-dg.pdf (part of CSR Synergy Bluetooth user documentation)	[DUN GW API]
CSR Synergy Bluetooth, CM – Connection Manager API Description, doc. no. api-0101-cm.pdf	[CM]

Terms and Definitions

DUN	Dial-Up Networking gateway. The gateway part of the dial-up networking profile as specified in [DUN]
DCE	Data Communication Equipment
DTE	Data Terminal Equipment
SIG	Special Interest Group
Data Terminal	This is the device that uses the dial-up service of the gateway. Typically this is a PC, laptop or PDA
CSR_BT_AT_DG	DUN GW and AT-interpreter
BlueCore®	Group term for CSR's range of Bluetooth wireless technology chips
Bluetooth®	Set of technologies providing audio and data transfer over short-range radio connections
CSR	Cambridge Silicon Radio
UniFi™	Group term for CSR's range of chips designed to meet IEEE 802.11 standards

Document History

Revision	Date	History
1	26 SEP 11	Ready for release 18.2.0

TradeMarks, Patents and Licences

Unless otherwise stated, words and logos marked with ™ or ® are trademarks registered or owned by CSR plc or its affiliates. Bluetooth® and the Bluetooth logos are trademarks owned by Bluetooth SIG, Inc. and licensed to CSR. Other products, services and names used in this document may have been trademarked by their respective owners.

The publication of this information does not imply that any licence is granted under any patent or other rights owned by CSR plc.

CSR reserves the right to make technical changes to its products as part of its development programme.

While every care has been taken to ensure the accuracy of the contents of this document, CSR cannot accept responsibility for any errors.

Life Support Policy and Use in Safety-critical Compliance

CSR's products are not authorised for use in life-support or safety-critical applications. Use in such applications is done at the sole discretion of the customer. CSR will not warrant the use of its devices in such applications.

Performance and Conformance

Refer to www.csrsupport.com for compliance and conformance to standards information.