quectel_bc66_drv

Generated by Doxygen 1.9.1

1 Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

bc66_at_cmd_t **BC66 Command struct**	**2**
bc66_obj_t	**3**

2 File Index

2.1 File List

Here is a list of all files with brief descriptions:

3 Data Structure Documentation

3.1 bc66_at_cmd_t Struct Reference

BC66 Command struct.

Data Fields

- const char ∗ cmd

    at command sentence
- cmd_flgs_t cmd_flags

    flags for command implementation (see
- char ∗ cmd_rsp

    expected command response
- uint32_t rsp_timeout

    response timeout [ms]

3.1.1 Detailed Description

BC66 Command struct.

3.1.2 Field Documentation

3.1.2.1 cmd `const char* cmd`

at command sentence

3.1.2.2 cmd_flags `cmd_flgs_t cmd_flags`

flags for command implementation (see
`flags enum)`

3.1.2.3 cmd_rsp `char* cmd_rsp`

expected command response

3.1.2.4 rsp_timeout `uint32_t rsp_timeout`

response timeout [ms]

The documentation for this struct was generated from the following file:

- /Users/jcbecerra/dev/fw/iot/quectel_bc66_driver/src/bc66_drv.c

3.2 bc66_obj_t Struct Reference

```
#include <bc66_drv.h>
```

Data Fields

- void(∗ func_init_ptr)()
  uart initialize function pointer
- void(∗ func_delay)(uint32_t t)
  delay function pointer
- int(∗ func_w_bytes_ptr)(uint8_t ∗txc, uint16_t len)
  write bytes function pointer
- int(∗ func_r_bytes_ptr)(uint8_t ∗rxc, uint16_t size)
  read one-byte function pointer
- struct {
  void(∗ MDM_PSM_EINT_N)(size_t pin_value)
  delay function pointer
  void(∗ MDM_PWRKEY_N)(size_t pin_value)
  modem power key function pointer
  void(∗ MDM_RESET_N)(size_t pin_value)
  modem reset function pointer
  void(∗ MDM_RI)()
  modem ring interrupt function pointer
  } control_lines

3.2.1 Field Documentation

3.2.1.1 `struct { ... } control_lines`

3.2.1.2 func_delay `void(* func_delay) (uint32_t t)`

delay function pointer

3.2.1.3 func_init_ptr `void(* func_init_ptr) ()`

uart initialize function pointer

3.2.1.4 func_r_bytes_ptr `int(* func_r_bytes_ptr) (uint8_t *rxc, uint16_t size)`

read one-byte function pointer

3.2.1.5 func_w_bytes_ptr `int(* func_w_bytes_ptr) (uint8_t *txc, uint16_t len)`

write bytes function pointer

3.2.1.6 MDM_PSM_EINT_N `void(* MDM_PSM_EINT_N) (size_t pin_value)`

delay function pointer

3.2.1.7 MDM_PWRKEY_N `void(* MDM_PWRKEY_N) (size_t pin_value)`

modem power key function pointer

3.2.1.8 MDM_RESET_N `void(* MDM_RESET_N) (size_t pin_value)`

modem reset function pointer

3.2.1.9 MDM_RI `void(* MDM_RI) ()`

modem ring interrupt function pointer

The documentation for this struct was generated from the following file:

- /Users/jcbecerra/dev/fw/iot/quectel_bc66_driver/src/bc66_drv.h

4 File Documentation

4.1 /Users/jcbecerra/dev/fw/iot/quectel_bc66_driver/src/bc66_drv.c File Reference

MIT License.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdarg.h>
#include "bc66_drv.h"
```

Data Structures

- struct bc66_at_cmd_t

  BC66 Command struct.

Macros

- #define CMD_END_LINE "\r\n"

  End of line command chars.
- #define RSP_OK "\r\nOK\r\n"

  Ok response.
- #define RSP_ERROR "\r\nERROR\r\n"

  Error response.
- #define RSP_END_OF_LINE "\r\n"

  End of line response chars.
- #define RSP_TIMEOUT "BC66_TIMEOUT\r\n"

  Answer when a timeout is occurred.
- #define RSP_NO_CMD_IMPEMENTED "BC66_NO_CMD\r\n"

  The command is not implemented.
- #define MAX_RSP_SIZE 64

  Max AT response size.

Enumerations

- enum cmd_flgs_t { TEST = 0x1 , READ = 0x2 , WRITE = 0x4 , EXE = 0x8 }

  Command possibilities indicator flags.

Functions

- void bc66_init (bc66_obj_t ∗bc66_obj)

  Function to initialize bc66 object.
- void bc66_deinit (bc66_obj_t ∗bc66_obj)

  Function to initialize bc66 object.
- bc66_ret_t bc66_send_at_command (bc66_cmd_type_t cmd_type, const bc66_cmd_list_t cmd_lst, const char ∗exp_rsp, const char ∗arg_fmt,...)

  Function to send at command sentence to bc66 module through an external function communication.
- char ∗ bc66_get_at_response (char ∗rsp)

  Function to get any response stored in the RX buffer.
- void bc66_reset (void)

  Reset the module when PIN is low.
- void bc66_power_on ()

  Pull down PWRKEY to turn on the module.
- void bc66_power_off ()

  Pull up PWRKEY to turn off the module.
- char ∗ bc66_get_last_response (void)

  Function to get last modem response.
- bool bc66_send_cmd_AT (void)

  Send AT command to sync baud rate.
- bc66_ret_t bc66_set_echo_mode (bool echo)

  Set Command Echo Mode.
- bc66_ret_t bc66_set_power_saving_mode (int mode)

  Power Saving Mode Setting (PSM).
- bc66_ret_t bc66_set_psd_conn (pdp_type_t pdp_type, const char ∗apn, const char ∗user, const char ∗pass)

  Set Default PSD Connection.
- bc66_ret_t bc66_set_mqtt_parameters (uint16_t keepalive, bool dataformat, bool session, bool version)

  Used to configure optional parameters of MQTT.
- bc66_ret_t bc66_open_net_mqtt_client (const char ∗server_ip, uint16_t server_port)

  Open a Network for MQTT Client.
- bc66_ret_t bc66_connect_mqtt_client (const char ∗client_id, const char ∗user, const char ∗pass)

  Connect a Client to MQTT Server.
- bc66_ret_t bc66_disconn_mqtt_client (void)

  Disconnect a Client from MQTT Server.
- bc66_ret_t bc66_publish_msg_mqtt (const char ∗topic, const char ∗msg, int qos)

  Publish Messages.

Variables

- const bc66_at_cmd_t bc66_cmds_list [ ]

  Define AT commands list: order must be equal to commands definition enum bc66_cmd_list_t.

4.1.1 Detailed Description

BC66 NB-IoT modem driver. (`https://www.quectel.com/product/bc66.htm`)
AT Command Syntax The AT or at prefix must be set at the beginning of each command line. Entering <CR> will terminate a command line. Commands are usually followed by a response that includes <CR><LF><response><CR><LF>. Throughout this document, only the responses are presented, <CR><LF> are omitted intentionally.
Types of AT Commands and Responses

- Test Command AT+<x>=?
- Read Command AT+<x>?
- Write Command AT+<x>=<n>
- Execution Command AT+<x>

Date

03/15/2021

Author

Eng. Juan Cruz Becerra

Version

1.0.0

4.1.2 Macro Definition Documentation

4.1.2.1 CMD_END_LINE `#define CMD_END_LINE "\r\n"`
End of line command chars.

4.1.2.2 MAX_RSP_SIZE `#define MAX_RSP_SIZE 64`
Max AT response size.

4.1.2.3 RSP_END_OF_LINE `#define RSP_END_OF_LINE "\r\n"`

End of line response chars.

4.1.2.4 RSP_ERROR `#define RSP_ERROR "\r\nERROR\r\n"`

Error response.

4.1.2.5 RSP_NO_CMD_IMPEMENTED `#define RSP_NO_CMD_IMPEMENTED "BC66_NO_CMD\r\n"`

The command is not implemented.

4.1.2.6 RSP_OK `#define RSP_OK "\r\nOK\r\n"`

Ok response.

4.1.2.7 RSP_TIMEOUT `#define RSP_TIMEOUT "BC66_TIMEOUT\r\n"`

Answer when a timeout is occurred.

4.1.3 Enumeration Type Documentation

4.1.3.1 cmd_flgs_t `enum cmd_flgs_t`

Command possibilities indicator flags.

Enumerator

TEST	Command has test posibility.
READ	Command has read posibility.
WRITE	Command has write posibility.
EXE	Command has execute posibility.

4.1.4 Function Documentation

4.1.4.1 bc66_connect_mqtt_client()

```
bc66_ret_t bc66_connect_mqtt_client (
        const char * client_id,
        const char * user,
        const char * pass )
```

Connect a Client to MQTT Server.

Parameters

client_id	: The client identifier. The max length is 128 bytes.
user	: User name of the client. It can be used for authentication. The max length is 256 bytes.
pass	: Password corresponding to the user name of the client. It can be used for authentication. The max length is 256 bytes.

Returns

See `bc66_ret_t` return codes.

4.1.4.2 bc66_deinit() `void bc66_deinit ( bc66_obj_t * bc66_obj )`

Function to initialize bc66 object.

Parameters

bc66_obj	

4.1.4.3 bc66_disconn_mqtt_client() `bc66_ret_t bc66_disconn_mqtt_client ( void )`

Disconnect a Client from MQTT Server.
Used when a client requests a disconnection from MQTT server. A DISCONNECT message is sent from the client to the server to indicate that it is about to close its TCP/IP connection.

Returns

See `bc66_ret_t` return codes.

4.1.4.4 bc66_get_at_response() `char* bc66_get_at_response ( char * rsp )`

Function to get any response stored in the RX buffer.

Parameters

rsp	: response to get

Returns

Response if found, NULL otherwise

4.1.4.5 bc66_get_last_response() `char* bc66_get_last_response ( void )`

Function to get last modem response.
If send a new AT command, the buffer which contain the last response will be erased.

Returns

Pointer to RX buffer with last response.

4.1.4.6 bc66_init() `void bc66_init ( bc66_obj_t * bc66_obj )`

Function to initialize bc66 object.

Parameters

bc66_obj	

4.1.4.7 bc66_open_net_mqtt_client() `bc66_ret_t bc66_open_net_mqtt_client ( const char * server_ip,`

```
uint16_t server_port )
```

Open a Network for MQTT Client.

Parameters

server_ip	: server ip (string)
server_port	: server port (0 to 65535)

Returns

See `bc66_ret_t` return codes.

4.1.4.8 bc66_power_off() `void bc66_power_off ( )`

Pull up PWRKEY to turn off the module.

4.1.4.9 bc66_power_on() `void bc66_power_on ( )`

Pull down PWRKEY to turn on the module.

4.1.4.10 bc66_publish_msg_mqtt() `bc66_ret_t bc66_publish_msg_mqtt (`

```
const char * topic,
const char * msg,
int qos )
```

Publish Messages.
Used to publish messages by a client to a server for distribution to interested subscribers.

Parameters

topic	: Topic that the client wants to subscribe to or unsubscribe from. The maximum length is 255 bytes.
msg	: The message that needs to be published. The maximum length is 700 bytes. If in data mode (after > is responded), the maximum length is 1024 bytes
qos	: Integer type. The QoS level at which the client wants to publish the messages. • 0 At most once • 1 At least once • 2 Exactly once

Returns

See `bc66_ret_t` return codes.

4.1.4.11 bc66_reset() `void bc66_reset (`

```
void )
```

Reset the module when PIN is low.

4.1.4.12 bc66_send_at_command() `bc66_ret_t bc66_send_at_command (`

```
bc66_cmd_type_t cmd_type,
const bc66_cmd_list_t cmd_lst,
const char * exp_rsp,
const char * arg_fmt,
 ... )
```

Function to send at command sentence to bc66 module through an external function communication.

Parameters

cmd_type	: BC66_CMD_TEST, BC66_CMD_READ, BC66_CMD_WRITE or BC66_CMD_EXE type.
cmd_lst	: command to send (see command list).
rsp	: pointer to expected response text.
arg_fmt	: arguments format (like printf function) and must be sended all arguments too.

Returns

See `bc66_ret_t` return codes.

4.1.4.13 bc66_send_cmd_AT()

```
bool bc66_send_cmd_AT (
            void )
```

Send AT command to sync baud rate.

Returns

See `bc66_ret_t` return codes.

4.1.4.14 bc66_set_echo_mode()

```
bc66_ret_t bc66_set_echo_mode (
            bool echo )
```

Set Command Echo Mode.
This Execution Command determines whether or not the UE echoes characters received from external MCU during command state.
The command takes effect immediately. Remain valid after deep-sleep wakeup. The configuration will be saved to NVRAM (should execute AT&W after this command is issued).

Parameters

echo	• false: Echo mode OFF • true: Echo mode ON

Returns

See `bc66_ret_t` return codes.

4.1.4.15 bc66_set_mqtt_parameters()

```
bc66_ret_t bc66_set_mqtt_parameters (
            uint16_t keepalive,
            bool dataformat,
            bool session,
            bool version )
```

Used to configure optional parameters of MQTT.

Parameters

keepalive	: Configure the keep-alive time. The range is 0-3600. The default value is 120. Unit: second. It defines the maximum time interval between messages received from a client. If the server does not receive a message from the client within 1.5 times of the keep-alive time period, it disconnects the client as if the client has sent a DISCONNECT message. 0 The client is not disconnected
dataformat	: The format of sent and received data. • 0 Text format • 1 Hex format
session	: The session type. • 0 The server must store the subscriptions of the client after it is disconnected. • 1 The server must discard any previously maintained information about the client and treat the connection as "clean".
version	: The version of MQTT protocol. • 0 MQTT v3.1 • 1 MQTT v3.1.1

Returns

See `bc66_ret_t` return codes.

4.1.4.16 bc66_set_power_saving_mode()

```
bc66_ret_t bc66_set_power_saving_mode (
        int mode )
```

Power Saving Mode Setting (PSM).

Power Saving Mode Setting.

Parameters

mode	Integer type. Disable or enable the use of PSM in the UE • 0 Disable the use of PSM • 1 Enable the use of PSM • 2 Disable the use of PSM and discard all parameters for PSM or, if available, reset to the default values.

Returns

See `bc66_ret_t` return codes.

4.1.4.17 bc66_set_psd_conn()

```
bc66_ret_t bc66_set_psd_conn (
        pdp_type_t pdp_type,
        const char * apn,
        const char * user,
        const char * pass )
```

Set Default PSD Connection.

This command sets the PSD connection settings for PDN connection on power-up. When attaching to the NB-IoT network on power-on, a PDN connection setup must be performed. In order to allow this to happen, PDN connection settings must be stored in NVRAM, thus making it to be used by the modem during the attach procedure.

Parameters

pdp_type	: Specify the type of packet data protocol.
apn	: A logical name that is used to select the GGSN or the external packet data network. The maximum configurable APN length is 99 bytes.
user	: The user name for accessing to the IP network. (Optional)
pass	: The password for accessing to the IP network. (Optional)

Returns

See `bc66_ret_t` return codes.

4.1.5 Variable Documentation

4.1.5.1 bc66_cmds_list `const bc66_at_cmd_t bc66_cmds_list[]`

Define AT commands list: order must be equal to commands definition enum bc66_cmd_list_t.

4.2 /Users/jcbecerra/dev/fw/iot/quectel_bc66_driver/src/bc66_drv.h File Reference

MIT License.

```
#include <stdint.h>
#include <stdbool.h>
#include <stddef.h>
```

Data Structures

- struct bc66_obj_t

Enumerations

- enum bc66_cmd_type_t { BC66_CMD_TEST , BC66_CMD_READ , BC66_CMD_WRITE , BC66_CMD_EXE }

  AT command posibility. Erch command can test and/or read and/or write and/or execute. Use with `bc66_send_at_command(...)` function.
- enum bc66_cmd_list_t {
  bc66_cmd_list_AT , bc66_cmd_list_ATI , bc66_cmd_list_ATE , bc66_cmd_list_CEREG ,
  bc66_cmd_list_CESQ , bc66_cmd_list_CGATT , bc66_cmd_list_CGPADDR , bc66_cmd_list_QCGDEFCONT ,
  bc66_cmd_list_CIMI , bc66_cmd_list_CPIN , bc66_cmd_list_CPSMS , bc66_cmd_list_QNBIOTEVENT ,
  bc66_cmd_list_QMTCFG , bc66_cmd_list_QMTOPEN , bc66_cmd_list_QMTCLOSE , bc66_cmd_list_QMTCONN ,
  bc66_cmd_list_QMTDISC , bc66_cmd_list_QMTSUB , bc66_cmd_list_QMTUNS , bc66_cmd_list_QMTPUB ,
  bc66_cmd_list_size }

  This is the commands implemented list.
- enum bc66_ret_t {
  bc66_ret_success , bc66_ret_timeout , bc66_ret_error , bc66_ret_out_of_range ,
  bc66_ret_not_init , bc66_ret_no_cmd_implemented }

  bc66 library api return
- enum pdp_type_t { pdp_type_ip , pdp_type_ipv6 , pdp_type_ipv4v6 , pdp_type_non_ip }

  Enumeration to specify the type of packet data protocol.

Functions

- void bc66_init (bc66_obj_t ∗bc66_obj)

  Function to initialize bc66 object.
- char ∗ bc66_get_at_response (char ∗rsp)

  Function to get any response stored in the RX buffer.
- bc66_ret_t bc66_send_at_command (bc66_cmd_type_t cmd_type, const bc66_cmd_list_t cmd_lst, const char ∗exp_rsp, const char ∗arg_fmt,...)

  Function to send at command sentence to bc66 module through an external function communication.
- void bc66_reset (void)

  Reset the module when PIN is low.
- void bc66_power_on ()

  Pull down PWRKEY to turn on the module.
- void bc66_power_off ()

  Pull up PWRKEY to turn off the module.
- char ∗ bc66_get_last_response (void)

  Function to get last modem response.
- bool bc66_send_cmd_AT (void)

  Send AT command to sync baud rate.
- bc66_ret_t bc66_set_echo_mode (bool echo)

  Set Command Echo Mode.
- bc66_ret_t bc66_set_power_saving_mode (int mode)

  Power Saving Mode Setting.
- bc66_ret_t bc66_set_psd_conn (pdp_type_t pdp_type, const char ∗apn, const char ∗user, const char ∗pass)

  Set Default PSD Connection.
- bc66_ret_t bc66_set_mqtt_parameters (uint16_t keepalive, bool dataformat, bool session, bool version)

  Used to configure optional parameters of MQTT.
- bc66_ret_t bc66_open_net_mqtt_client (const char ∗server_ip, uint16_t server_port)

  Open a Network for MQTT Client.
- bc66_ret_t bc66_connect_mqtt_client (const char ∗client_id, const char ∗user, const char ∗pass)

  Connect a Client to MQTT Server.
- bc66_ret_t bc66_disconn_mqtt_client (void)

  Disconnect a Client from MQTT Server.
- bc66_ret_t bc66_publish_msg_mqtt (const char ∗topic, const char ∗msg, int qos)

  Publish Messages.

4.2.1 Detailed Description

Copyright

Juan Cruz Becerra

BC66 NB-IoT modem driver. (https://www.quectel.com/product/bc66.htm)
AT Command Syntax The AT or at prefix must be set at the beginning of each command line. Entering <↩CR> will terminate a command line. Commands are usually followed by a response that includes <CR><↩LF><response><CR><LF>. Throughout this document, only the responses are presented, <CR><LF> are omitted intentionally.
Types of AT Commands and Responses

- Test Command AT+<x>=?
- Read Command AT+<x>?
- Write Command AT+<x>=<n>
- Execution Command AT+<x>

Date

03/15/2021

Author

Eng. Juan Cruz Becerra

Version

1.0.0

4.2.2 Enumeration Type Documentation

4.2.2.1 bc66_cmd_list_t `enum bc66_cmd_list_t`

This is the commands implemented list.

Enumerator

bc66_cmd_list_AT	AT command. Use to sync baud rate.
bc66_cmd_list_ATI	Display Product Identification Information.
bc66_cmd_list_ATE	Set Command Echo Mode.
bc66_cmd_list_CEREG	EPS Network Registration Status.
bc66_cmd_list_CESQ	Extended Signal Quality.
bc66_cmd_list_CGATT	PS Attachment or Detachment.
bc66_cmd_list_CGPADDR	Show PDP Addresses.
bc66_cmd_list_QCGDEFCONT	Set Default PSD Connection Settings.
bc66_cmd_list_CIMI	Request International Mobile Subscriber Identity.
bc66_cmd_list_CPIN	Enter PIN.
bc66_cmd_list_CPSMS	Power Saving Mode Setting.
bc66_cmd_list_QNBIOTEVENT	Enable/Disable NB-IoT Related Event Report.
bc66_cmd_list_QMTCFG	Configure Optional Parameters of MQTT.
bc66_cmd_list_QMTOPEN	Open a Network for MQTT Client.
bc66_cmd_list_QMTCLOSE	Close a Network for MQTT Client.
bc66_cmd_list_QMTCONN	Connect a Client to MQTT Server.
bc66_cmd_list_QMTDISC	Disconnect a Client from MQTT Server.
bc66_cmd_list_QMTSUB	Subscribe to Topics.
bc66_cmd_list_QMTUNS	Unsubscribe from Topics.
bc66_cmd_list_QMTPUB	Publish Messages.
bc66_cmd_list_size	Is not a command. Only to know commands quantity.

4.2.2.2 bc66_cmd_type_t `enum bc66_cmd_type_t`

AT command posibility. Erch command can test and/or read and/or write and/or execute. Use with `bc66_send↩ _at_command(...)` function.

Enumerator

BC66_CMD_TEST	Send AT TEST command.
BC66_CMD_READ	Send AT READ command.
BC66_CMD_WRITE	Send AT WRITE command.
BC66_CMD_EXE	Send AT TEST command.

4.2.2.3 bc66_ret_t `enum bc66_ret_t`

bc66 library api return

Enumerator

bc66_ret_success	Modem data process successful.
bc66_ret_timeout	Response timeout.
bc66_ret_error	Modem response with error message.
bc66_ret_out_of_range	At least some argument is out of range.
bc66_ret_not_init	
bc66_ret_no_cmd_implemented	RSP_NO_CMD_IMPEMENTED.

4.2.2.4 pdp_type_t `enum pdp_type_t`

Enumeration to specify the type of packet data protocol.

Enumerator

pdp_type_ip	Internet Protocol (IETF STD 5).
pdp_type_ipv6	Internet Protocol version 6 (IETF RFC 2460).
pdp_type_ipv4v6	Dual IP stack (see 3GPP TS 24.301).
pdp_type_non_ip	Transfer of Non-IP data to external packet network (see 3GPP TS 24.301).

4.2.3 Function Documentation

4.2.3.1 bc66_connect_mqtt_client()

```
bc66_ret_t bc66_connect_mqtt_client (
        const char * client_id,
        const char * user,
        const char * pass )
```

Connect a Client to MQTT Server.

Parameters

client↩ _id	: The client identifier. The max length is 128 bytes.

Parameters

user	: User name of the client. It can be used for authentication. The max length is 256 bytes.
pass	: Password corresponding to the user name of the client. It can be used for authentication. The max length is 256 bytes.

Returns

See `bc66_ret_t` return codes.

4.2.3.2 bc66_disconn_mqtt_client()

```
bc66_ret_t bc66_disconn_mqtt_client (
        void )
```

Disconnect a Client from MQTT Server.
Used when a client requests a disconnection from MQTT server. A DISCONNECT message is sent from the client to the server to indicate that it is about to close its TCP/IP connection.

Returns

See `bc66_ret_t` return codes.

4.2.3.3 bc66_get_at_response()

```
char* bc66_get_at_response (
        char * rsp )
```

Function to get any response stored in the RX buffer.

Parameters

rsp	: response to get

Returns

Response if found, NULL otherwise

4.2.3.4 bc66_get_last_response()

```
char* bc66_get_last_response (
        void )
```

Function to get last modem response.
If send a new AT command, the buffer which contain the last response will be erased.

Returns

Pointer to RX buffer with last response.

4.2.3.5 bc66_init()

```
void bc66_init (
        bc66_obj_t * bc66_obj )
```

Function to initialize bc66 object.

Parameters

bc66_obj	

4.2.3.6 bc66_open_net_mqtt_client() `bc66_ret_t bc66_open_net_mqtt_client (`

```
const char * server_ip,
uint16_t server_port )
```

Open a Network for MQTT Client.

Parameters

server_ip	: server ip (string)
server_port	: server port (0 to 65535)

Returns

See `bc66_ret_t` return codes.

4.2.3.7 bc66_power_off() `void bc66_power_off ( )`

Pull up PWRKEY to turn off the module.

4.2.3.8 bc66_power_on() `void bc66_power_on ( )`

Pull down PWRKEY to turn on the module.

4.2.3.9 bc66_publish_msg_mqtt() `bc66_ret_t bc66_publish_msg_mqtt (`

```
const char * topic,
const char * msg,
int qos )
```

Publish Messages.
Used to publish messages by a client to a server for distribution to interested subscribers.

Parameters

topic	: Topic that the client wants to subscribe to or unsubscribe from. The maximum length is 255 bytes.
msg	: The message that needs to be published. The maximum length is 700 bytes. If in data mode (after > is responded), the maximum length is 1024 bytes
qos	: Integer type. The QoS level at which the client wants to publish the messages. • 0 At most once • 1 At least once • 2 Exactly once

Returns

See `bc66_ret_t` return codes.

4.2.3.10 bc66_reset() `void bc66_reset (`

```
void )
```

Reset the module when PIN is low.

4.2.3.11 bc66_send_at_command() `bc66_ret_t bc66_send_at_command (`

```
bc66_cmd_type_t cmd_type,
const bc66_cmd_list_t cmd_lst,
const char * exp_rsp,
const char * arg_fmt,
 ... )
```

Function to send at command sentence to bc66 module through an external function communication.

Parameters

cmd_type	: BC66_CMD_TEST, BC66_CMD_READ, BC66_CMD_WRITE or BC66_CMD_EXE type.
cmd_lst	: command to send (see command list).
rsp	: pointer to expected response text.
arg_fmt	: arguments format (like printf function) and must be sended all arguments too.

Returns

See `bc66_ret_t` return codes.

4.2.3.12 bc66_send_cmd_AT() `bool bc66_send_cmd_AT (`

```
void )
```

Send AT command to sync baud rate.

Returns

See `bc66_ret_t` return codes.

4.2.3.13 bc66_set_echo_mode() `bc66_ret_t bc66_set_echo_mode (`

```
bool echo )
```

Set Command Echo Mode.
This Execution Command determines whether or not the UE echoes characters received from external MCU during command state.
The command takes effect immediately. Remain valid after deep-sleep wakeup. The configuration will be saved to NVRAM (should execute AT&W after this command is issued).

Parameters

echo	• false: Echo mode OFF • true: Echo mode ON

Returns

See `bc66_ret_t` return codes.

4.2.3.14 bc66_set_mqtt_parameters() `bc66_ret_t bc66_set_mqtt_parameters (`

```
uint16_t keepalive,
bool dataformat,
bool session,
bool version )
```

Used to configure optional parameters of MQTT.

Parameters

keepalive	: Configure the keep-alive time. The range is 0-3600. The default value is 120. Unit: second. It defines the maximum time interval between messages received from a client. If the server does not receive a message from the client within 1.5 times of the keep-alive time period, it disconnects the client as if the client has sent a DISCONNECT message. 0 The client is not disconnected
dataformat	: The format of sent and received data. • 0 Text format • 1 Hex format
session	: The session type. • 0 The server must store the subscriptions of the client after it is disconnected. • 1 The server must discard any previously maintained information about the client and treat the connection as "clean".
version	: The version of MQTT protocol. • 0 MQTT v3.1 • 1 MQTT v3.1.1

Returns

See `bc66_ret_t` return codes.

4.2.3.15 bc66_set_power_saving_mode()

```
bc66_ret_t bc66_set_power_saving_mode (
        int mode )
```

Power Saving Mode Setting.

Parameters

mode	Integer type. Disable or enable the use of PSM in the UE • 0 Disable the use of PSM • 1 Enable the use of PSM • 2 Disable the use of PSM and discard all parameters for PSM or, if available, reset to the default values.

Returns

See `bc66_ret_t` return codes.

Power Saving Mode Setting.

Parameters

mode	Integer type. Disable or enable the use of PSM in the UE • 0 Disable the use of PSM • 1 Enable the use of PSM • 2 Disable the use of PSM and discard all parameters for PSM or, if available, reset to the default values.

Returns

See `bc66_ret_t` return codes.

4.2.3.16 bc66_set_psd_conn()

```
bc66_ret_t bc66_set_psd_conn (
        pdp_type_t pdp_type,
        const char * apn,
        const char * user,
        const char * pass )
```

Set Default PSD Connection.

This command sets the PSD connection settings for PDN connection on power-up. When attaching to the NB-IoT network on power-on, a PDN connection setup must be performed. In order to allow this to happen, PDN connection settings must be stored in NVRAM, thus making it to be used by the modem during the attach procedure.

Parameters

pdp_type	: Specify the type of packet data protocol.
apn	: A logical name that is used to select the GGSN or the external packet data network. The maximum configurable APN length is 99 bytes.
user	: The user name for accessing to the IP network. (Optional)
pass	: The password for accessing to the IP network. (Optional)

Returns

See `bc66_ret_t` return codes.

Index