Specification of RS232 Serial Communication Interface for CG Devices Control

Version 1.7

11 August 2014

Table of Contents

List of Figure

List of Table

Revision History

Date	Version	Description
11th June 2009	1.0	Original Version
10th February 2010	1.1	Curtain control is supported Revised section: 4.1.3 Device Type 4.2 Device Control
17th March 2010	1.2	Enable and disable of device auto update status are supported Revised section: 4.1.1 Cmd 4.4 Auto Update of Device Status Version enquiry of ZDCP is supported Revised section: 4.1.1 Cmd 4.5 ZDCP Version Enquiry Format of status response is revised Revised section: 4.3 Device Status Request
15th April 2011	1.3	Revised section: 4.1.1 Cmd
20th March 2012	1.4	Command Update
15th April 2012	1.5	Revised section: 4.5 ZDCP Version Enquiry
5th June 2014	1.6	CG102IR infrared signal control is supported Revised section: 1 Introduction 4.1.3 DeviceType 4.2 Device Control 4.2.3 CG102IR Infrared Signal Control
11th August 2014	1.7	Diagrams updated

1 Introduction

Current document provides details description of specification and protocol of the interface between the external system and CG102RS232 through the RS232 interface.

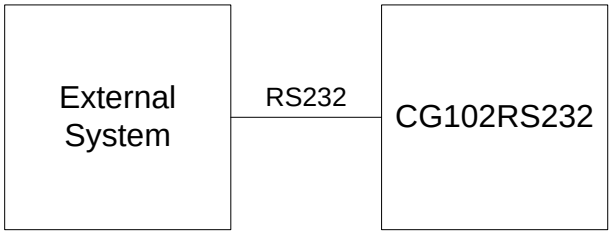

Figure 1-1 Communication between the MCU and IRC

The CG102RS232 is the device provides interface for external system to control and acquire status of CG ZigBee devices including:

- ZigBee Switch
- ZigBee Dimmer
- ZigBee Socket
- ZigBee Curtain Device
- CG102IR

The specification of physical layer and data layer of CG102RS232 are specified in Section 2 Serial Interface and Section 3 Serial Data Format respectively. The ZigBee Device Control Protocol (ZDCP) on RS232 is illustrated in Section 4.

2 RS232 Serial Interface

The CG102RS232 provides standard DB9 female connector for connection with external system. The pin assignment of the DB9 female connector is shown in Figure 2-1.

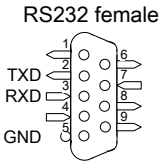

Figure 2-1 Pin assignment of the DB9 female connector

The connection between CG102RS232 and external system uses 3 pins of the RS232 interface, which are TXD, RXD and GND (Figure 2-2). The physical characteristics of the RS232 interface are shown in Table 2-1.

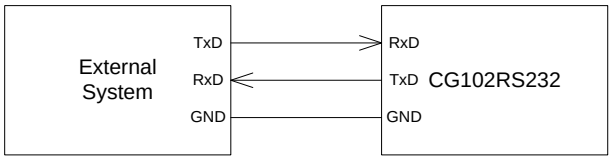

Figure 2-2 RS232 Interface connection between external system and CG102RS232

Table 2-1 Physical characteristics of the UART interface

PARAMETER	MIN	TYP	MAX	UNIT
Date Rate	-	57,600	-	bps
Framing	8N1			
Hardware flow control	Nil			
Voltage Level of bit ‘0’	+5	12	15	V
Voltage Level of bit ‘1’	-15	12	-5	V

In the RS232 serial communication protocol, the sender is able to request the receiver to acknowledge the sender. The timeout of the sender to received acknowledgment from receiver is 30ms. If the sender did not receive any acknowledgement from receiver within 30ms after transmission, sender will retransmit the data frame again.

3 Serial Data Format

The data format of serial interface is shown in Figure 3-1.

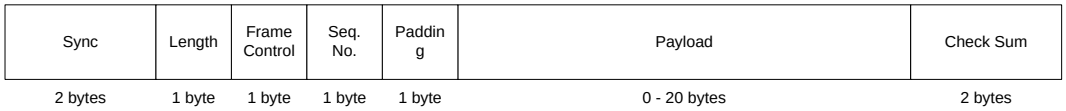

Figure 3-1 Data format of the serial interface

The data frame of the serial interface comprises of different fields, which are summarized in the following table (Table 3-1):

Table 3-1 Summary of fields in the data frame

Field	Description	Size
Sync	Sync-code	2 bytes
Length	Length of data frame	1 byte
Frame Control	Used for the frame acknowledgement	1 byte
Seq. No.	Sequence number of the data frame	1 byte
Padding	Padding field contain of ‘0’	1 byte
Payload	Payload reserved for Protocol Data Unit of ZDCP	1 – 20 bytes
Check Sum	Check sum code for error detection	2 bytes

3.1 Sync

The 2-byte sync-code is used to indicate the start of the data frame. The format of the sync-code is shown in Figure 3-2.

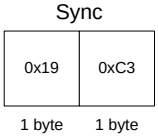

Figure 3-2 Format of the sync-code

3.2 Length

The 1-byte data frame length is used to indicate the total length of the data frame in term of number of byte. The length is counted start from the “Frame Control” to the end of the “Payload”.

3.3 Frame Control

The 1-byte frame control is used to indicate whether the data frame is required to be acknowledged by the receiver after the receiver received the data frame correctly. The format of the frame control is shown in Figure 3-3.

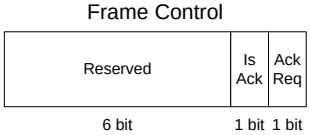

Figure 3-3 Data format of the frame control

When the sender is desired to receive an acknowledgment from the receiver to confirm the data is received correctly, the sender shall set the AckReq of the frame control field to ‘1’. Otherwise, the AckReq should be ‘0’.

When the receiver received a complete data frame correctly with AckReq is set to ‘1’, the receiver shall reply the sender with an acknowledgment frame with the IsAck set to ‘1’ immediately. The format of the acknowledgment frame is shown in Figure 3-4.

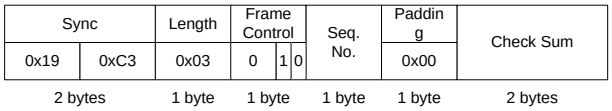

Figure 3-4 Frame format of the acknowledgment frame

Besides the acknowledgment frame, the IsAck shall always set to '0' in the "Frame Control" of any data frame.

3.4 Seq. No.

The 1-byte sequence number is used to identify different data frame during communication. The sequence number is generated by the sender and the value is started from 0. The value of the sequence number is increment by one for each new data frame and it should be reset to 0 after the value is larger than 255. When the data frame is required to be acknowledged, the receiver should fill the "Seq. No." of the acknowledgment frame with the received "Seq. No." from sender. The example shown in Figure 3-6 elaborates the general operation of the sequence number with the acknowledgement.

3.5 Padding

The 1-byte padding is reserved. In current version, the value of padding field should always be '0'.

3.6 Payload

The variable length payload is used to place the Protocol Data Unit of ZDCP.

3.7 Check Sum

The 2-byte check sum code is used to identify the correctness of the received data frame. The computation of the check sum is the summation of the entire byte which is counted from "Length" to the end of "Payload". The 2-byte summation result is placed into the field with Lower byte first and Upper byte end. For example, if the result is 0x1234, the check sum should be:

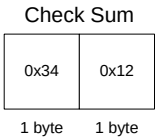

Figure 3-5 The contain of check sum field if the summation result is 0x1234

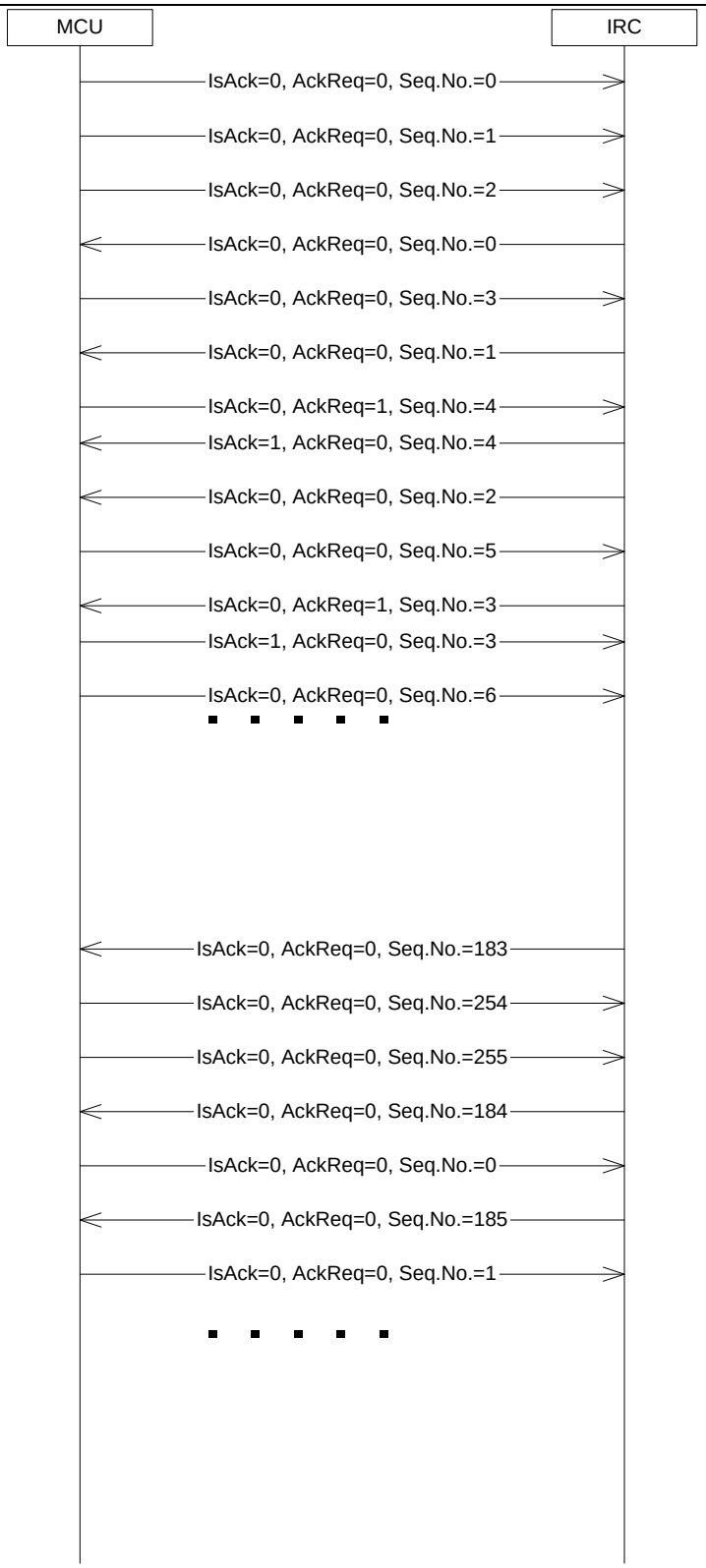

Figure 3-6 Operation example of the sequence number and acknowledgment.

4 ZigBee Device Control Protocol (ZDCP)

Current section provides details description on the communication protocol for external system to send and receive control and status from the CG ZigBee devices.

The CG102RS232 provide request-response communication approach for external system to control the CG ZigBee devices directly. The general concept is shown in Figure 4-1.

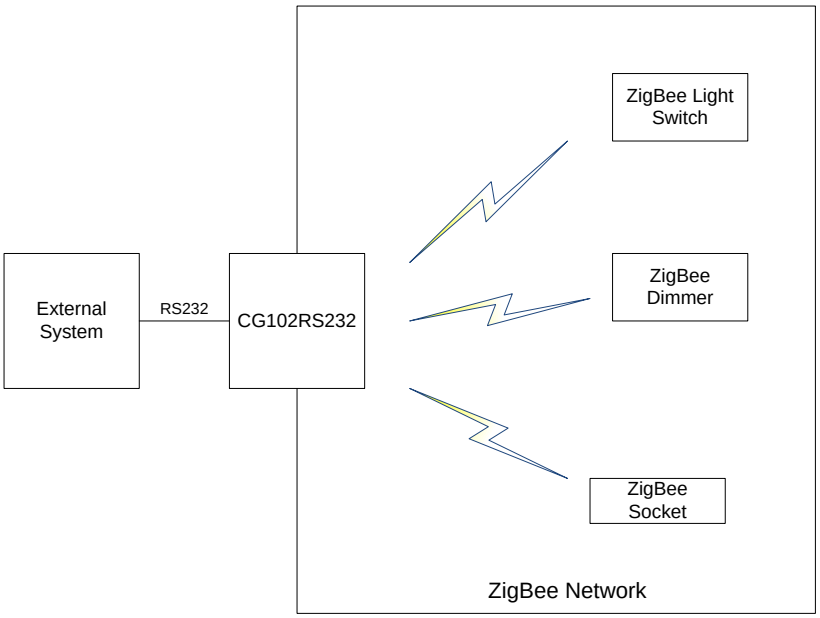

Figure 4-1 Request-response communication approach for ZigBee device control

According to Figure 4-1, when external system is desired to control a device within the ZigBee network, the external system may send a request command to the CG102RS232 with specified device ID, control command and control status through the RS232 interface. The CG102RS232 will send the control command to the target ZigBee device through the ZigBee network. If the control is accomplished, the target ZigBee device will send back a response to the CG102RS232. After the CG102RS232 received the response from ZigBee device, the CG102RS232 will send the response to the external system to inform the request is accomplished. If the target device is unreachable in the ZigBee network, the CG102RS232 will also send the response to the external system to inform the request is fail to accomplish. Details of the format of request-response commands and operation policy are described in current section.

4.1 Format of Protocol Data Unit (PDU)

The general protocol data unit (PDU) format of ZDCPl is shown in Figure 3-1.

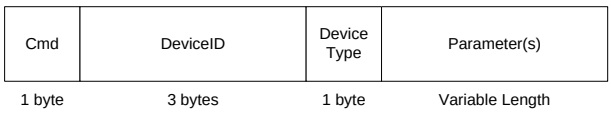

Figure 4-2 General PDU format of ZDCP

The PDU of the ZDCP comprises of different fields, which are summarized in the following table (Table 4-1):

Table 4-1 Summary of fields in the PDU of ZDCP

Field	Description	Size
Cmd	Command	1 byte
DeviceID	The device ID of target device	3 byte
DeviceType	The device type of the target device	1 byte
Parameter(s)	Parameter(s) for device control	Variable Length

4.1.1 Cmd

The command is used to specify the function of the PDU. The command also determines different parameters enclosed in the PDU. The summary of the command type and relative parameters is shown in the Table 4-2.

Table 4-2 Summary of the command type

Cmd		Description	Parameters
Name	Code		
CONTROL_REQ	0x01	Device Control Request	ControlCmd, ControlStatus (Set Dimmer Level Only)
CONTROL_RESP	0x02	Device Control Response	ControlCmd, ControlStatus (Set Dimmer Level Only), ControlResult
STATUS_REQ	0x03	Device Status Request	-
STATUS_RESP	0x04	Device Status Response	Refer to section 4.3
ENABLE_AU_REQ	0x81	Enable Auto Update Status Request	-
ENABLE_AU_RESP	0x82	Enable Auto Update Status Response	Result
DISABLE_AU_REQ	0x83	Disable Auto Update Status Request	-
DISABLE_AU_RESP	0x84	Disable Auto Update Status Response	Result
VERSION_REQ	0xF1	ZDCP version request	Refer to section 4.5
VERSION_RESP	0xF2	ZDCP version response	Refer to section 4.5

4.1.2 DeviceID

The 3-byte DeviceID is used to specify the identity of the target device in the ZigBee network. If the ZigBee devices are configured through the HomeNET planner, the DeviceID of the target device should be ID (2-byte) + EP (1-byte), which ID and EP please refer to HomeNET planner. For example:

- An one-gang dimmer device on the HomeNET planner which ID is assigned to 0001, the DeviceID for CG102RS232 should be 000101.
- A two-gang switch device on the HomeNET planner which ID is assigned to 0002, the DeviceID for CG102RS232 of the first gang should be 000201 and the second gang should be 000202.

4.1.3 DeviceType

The 1-byte DeviceType is used to specify the device type of the target device. The different types of device are shown in Table 4-3.

Table 4-3 Identification of device type

DeviceType		Description
Name	**Code**	
DEVICE_SWITCH	0x01	Light switch device type
DEVICE_SOCKET	0x02	Power socket device type
DEVICE_DIMMER	0x03	Dimmer light switch device type
DEVICE_CURTAIN	0x04	Curtain control device type
DEVICE_IR	0x05	CG102IR device type

4.2 Device Control

In the device Control, 1 byte of Control Command (ControlCmd) is required in the parameter field of PDU to identify the control function. The different types of ControlCmd are shown in Table 4-4

Table 4-4 Device control command

ControlCmd		Description
Name	**Code**	
DEVICE_ON	0x01	Control device ON
DEVICE_OFF	0x02	Control device OFF
DIM_UP_START	0x11	Start to dim-up (Dimmer only)
DIM_UP_STOP	0x12	Stop to dim-up (Dimmer only)
DIM_DOWN_START	0x13	Start to dim-down (Dimmer only)
DIM_DOWN_STOP	0x14	Stop to dim-down (Dimmer only)
DIM_TO_LEVEL	0x15	Dim to a provided dimming level (Dimmer only)
CURTAIN_UP	0x21	Control curtain to move up when curtain is stopped Control curtain stop when curtain is moving
CURTAIN_DOWN	0x22	Control curtain to move down when curtain is stopped Control curtain stop when curtain is moving
SEND_IR_CODE	0x31	Control CG102IR to send infrared signal according to the provided IR signal code

When the device control PDU is sent to CG102RS232, a device control response with control result (ControlResult) is expect to be returned from CG102RS232 after the device control is accomplished. The ControlResult is used to indicate the result of the control. Table 4-5 shows the format of the ControlResult.

Table 4-5 Format of ControlResult

ControlResult		Description
Name	**Code**	
SUCCESS	0x00	Control device success
FAILURE	0x01	Control device failure

4.2.1 Device ON/OFF Control

To control device ON, following PDU is required send to CG102RS232:

Cmd	DeviceID	Device Type	Control Cmd
0x01			0x01
1 byte	3 bytes	1 byte	1 byte

Figure 4-3 Request PDU of control device ON

If the control device ON is accomplished, following PDU is expected to be returned from CG102RS232:

Cmd	DeviceID	Device Type	Control Cmd	Control Result
0x02			0x01	0x00
1 byte	3 bytes	1 byte	1 byte	1 byte

Figure 4-4 Response PDU of control device ON

If the target device cannot be reach during the control process, following PDU is expected to be returned from CG102RS232:

Cmd	DeviceID	Device Type	Control Cmd	Control Result
0x02			0x01	0x01
1 byte	3 bytes	1 byte	1 byte	1 byte

Figure 4-5 Response PDU of device not found

To control device OFF, following PDU is required send to CG102RS232:

Cmd	DeviceID	Device Type	Control Cmd
0x01			0x02
1 byte	3 bytes	1 byte	1 byte

Figure 4-6 Request PDU of control device OFF

If the control device OFF is accomplished, following PDU is expected to be returned from CG102RS232:

Cmd	DeviceID	Device Type	Control Cmd	Control Result
0x02			0x02	0x00
1 byte	3 bytes	1 byte	1 byte	1 byte

Figure 4-7 Response PDU of control device OFF

If the target device cannot be reach during the control process, following PDU is expected to be returned from CG102RS232:

Cmd	DeviceID	Device Type	Control Cmd	Control Result
0x02			0x02	0x01
1 byte	3 bytes	1 byte	1 byte	1 byte

Figure 4-8 Response PDU of device not found

4.2.2 Device Dimming Level Control

Device dimming level control is only applied to the control of dimmer device.
To control the dimmer start to dim-up, following PDU is required send to CG102RS232:

Cmd	DeviceID	Device Type	Control Cmd
0x01			0x11
1 byte	3 bytes	1 byte	1 byte

Figure 4-9 Request PDU of dimmer start to dim-up

If the control dimmer start dim-up is accomplished, following PDU is expected to be returned from CG102RS232:

Cmd	DeviceID	Device Type	Control Cmd	Control Result
0x02			0x11	0x00
1 byte	3 bytes	1 byte	1 byte	1 byte

Figure 4-10 Response PDU of dimmer start to dim-up

To control the dimmer stop to dim-up, following PDU is required send to CG102RS232:

Cmd	DeviceID	Device Type	Control Cmd
0x01			0x12
1 byte	3 bytes	1 byte	1 byte

Figure 4-11 Request PDU of dimmer stop to dim-up

If the control dimmer stop dim-up is accomplished, following PDU is expected to be returned from CG102RS232:

Cmd	DeviceID	Device Type	Control Cmd	Control Result
0x02			0x12	0x00
1 byte	3 bytes	1 byte	1 byte	1 byte

Figure 4-12 Response PDU of dimmer stop to dim-up

To control the dimmer start to dim-down, following PDU is required send to CG102RS232:

Cmd	DeviceID	Device Type	Control Cmd
0x01			0x13
1 byte	3 bytes	1 byte	1 byte

Figure 4-13 Request PDU of dimmer start to dim-down

If the control dimmer start dim-down is accomplished, following PDU is expected to be returned from CG102RS232:

Cmd	DeviceID	Device Type	Control Cmd	Control Result
0x02			0x13	0x00
1 byte	3 bytes	1 byte	1 byte	1 byte

Figure 4-14 Response PDU of dimmer start to dim-down

To control the dimmer stop to dim-down, following PDU is required send to CG102RS232:

Cmd	DeviceID	Device Type	Control Cmd
0x01			0x14
1 byte	3 bytes	1 byte	1 byte

Figure 4-15 Request PDU of dimmer stop to dim-down

If the control dimmer stop dim-down is accomplished, following PDU is expected to be returned from CG102RS232:

Cmd	DeviceID	Device Type	Control Cmd	Control Result
0x02			0x14	0x00
1 byte	3 bytes	1 byte	1 byte	1 byte

Figure 4-16 Response PDU of dimmer stop to dim-down

To control dimmer to a desired dimming level, the dimming level is required to provide in the PDU. Following is the example of PDU when the desired dimming level is 70% (Control Status is required set to the value of 70, 0x46 in hexadecimal):

Cmd	DeviceID	Device Type	Control Cmd	Control Status
0x01			0x15	0x46
1 byte	3 bytes	1 byte	1 byte	1 byte

Figure 4-17 Example of set the dimmer level of 70%

If the control dimmer level (70%) is accomplished, following PDU is expected to be returned from CG102RS232:

Cmd	DeviceID	Device Type	Control Cmd	Control Status	Control Result
0x02			0x15	0x46	0x00
1 byte	3 bytes	1 byte	1 byte	1 byte	1 byte

Figure 4-18 Example of response PDU of set the dimmer level of 70%

If the target device cannot be reach during the control process, the control result in response PDU from CG102RS232 will be FAILURE.

4.2.3 CG102IR Infrared Signal Control

Note: Infrared signal control is only available for the firmware version of 1.4.

To control CG102IR to send IR signal, following PDU is required send to CG102RS232 (IRCode please refer to the IRSI of the “IR Learning Tools”):

Cmd	DeviceID	Device Type	Control Cmd	IRCode
0x01		0x05	0x31	
1 byte	3 bytes	1 byte	1 byte	1 byte

Figure 4-19 Request PDU of control CG102IR to send IR signal

If the CG102IR is able to send the IR signal, following PDU is expected to be returned from CG102RS232:

Cmd	DeviceID	Device Type	Control Cmd	IRCode	Control Result
0x02		0x05	0x31		0x00
1 byte	3 bytes	1 byte	1 byte	1 byte	1 byte

Figure 4-20 Response PDU of control CG102IR to send IR signal

If the target CG102IR cannot be reach during the control process, following PDU is expected to be returned from CG102RS232:

Cmd	DeviceID	Device Type	Control Cmd	IRCode	Control Result
0x02		0x05	0x31		0x01
1 byte	3 bytes	1 byte	1 byte	1 byte	1 byte

Figure 4-21 Response PDU of CG102IR not found

4.3 Device Status Request

The general PDU format of the status response is shown in Figure 4-19.

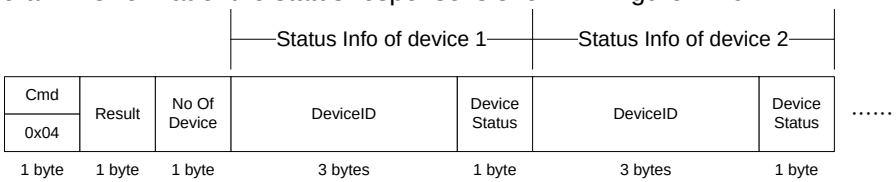

Figure 4-22 General PDU format of status response

The PDU of the status response comprises of different fields, which are summarized in the following table (Table 4-6):

Table 4-6 Summary of fields in the PDU of status response

Field	Description	Size
Cmd	Status Response Command	1 byte
Result	SUCCESS/FAILURE	1 byte
No of Device	Indicate number of device status	1 byte
DeviceID	The device ID of response device	3 byte
Device Status	The status of response device	1 byte

In the device status response PDU, multiple sets of device status information (DeviceID+DeviceStatus) may present in one PDU when the responding device has more than one control end-point (e.g. the two-gang switch). The number of set of device status information is indicated by the field “No Of Device”. Within a set of device status information, 1 byte of Device Status (DeviceStatus) is expected follow the DeviceID to indicate status of the target device. The status of device are shown in Table 4-7

Table 4-7 Device status

DeviceStatus		Description
Name	Code	
STATUS_OFF	0x00	Device status is OFF
STATUS_ON	0xFF	Device status is ON
STATUS_DIMMING_LEVEL	0x00 – 0x64	Indicate the dimming level from 0% to 100% (Dimmer only)

To acquire the device status, following PDU is required send to CG102RS232:

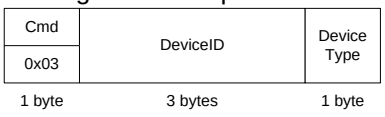

Figure 4-23 Request PDU of acquire device status

If the target device has single control end-point and such control end-point is turning ON, following PDU is expected to be returned from CG102RS232:

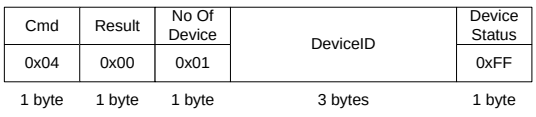

Figure 4-24 Response PDU of device status ON

If the target device has single control end-point and such control end-point is turning OFF, following PDU is expected to be returned from CG102RS232:

Cmd	Result	No Of Device	DeviceID	Device Status
0x04	0x00	0x01		0x00
1 byte	1 byte	1 byte	3 bytes	1 byte

Figure 4-25 Response PDU of device status OFF

If the target dimmer is turning ON at 40%, following PDU is expected to be returned from CG102RS232:

Cmd	DeviceID	Device Type	Device Status
0x03			0x28
1 byte	3 bytes	1 byte	1 byte

Figure 4-26 Response PDU of dimmer status at 40%

If the target device has two control end-points, and one of the control end-points is turning ON and another control end-point is turning OFF, following PDU is expected to be returned from CG102RS232:

Cmd	Result	No Of Device	DeviceID	Device Status	DeviceID	Device Status
0x04	0x00	0x02		0xFF		0x00
1 byte	1 byte	1 byte	3 bytes	1 byte	3 bytes	1 byte

Figure 4-27 Response PDU of device status which one of the end-points is ON and another en-point is OFF

4.4 Auto Update of Device Status

The auto status update is supported by device type of Switch, Socket and Dimmer. Such function allows the controller (Panel, CG102RS232, etc) to be updated once the status of device (Switch, Socket or Dimmer) is changed. The command ENABLE_AU_REQ is used to enable such functionality of a target device and the command DISABLE_AU_REQ is used to enable such functionality of a target device. When the ENABLE_AU_REQ is sent to a device from a controller, the controller will be registered to the device memory and the device will send status response (refer to section 4.3) to the controller automatically when the status of the device is changed. Please note that, each device (Switch, Socket or Dimmer) is able to register **maximum EIGHT (8) controller information**. Therefore, please ensure the number of registered controller (Panel, CG102RS232, etc) of a device is not exceeding EIGHT for each device.

Following is the example format of the PDU from a CG102RS232 to a device for register the controller for auto status update:

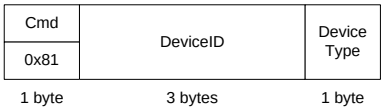

Figure 4-28 PDU format to enable a target device for auto status update

If the target device is found and such device is supporting auto status update, following PDU will be return from the CG102RS232:

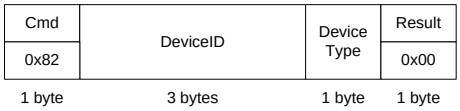

Figure 4-29 Response PDU format for enabling the auto status update successfully

Otherwise, following PDU will be return from the CG102RS232 to indicate the request action is failed:

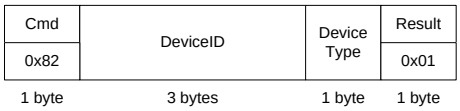

Figure 4-30 Response PDU format when enable the auto status update failed

Following is the example format of the PDU from a CG102RS232 to a device for deregistering the controller for auto status update:

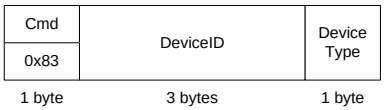

Figure 4-31 PDU format to disable a target device for auto status update

If the target device is found and such device is supporting auto status update, following PDU will be return from the CG102RS232:

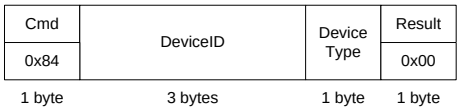

Figure 4-32 Response PDU format for disabling the auto status update successfully

Otherwise, following PDU will be return from the CG102RS232 to indicate the request action is failed:

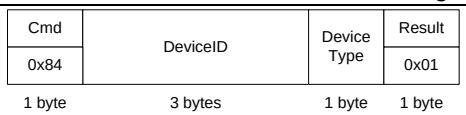

Figure 4-33 Response PDU format when disable the auto status update failed

4.5 ZDCP Version Enquiry

To enquire the version of ZDCP of a CG102RS232, following PDU (only comprises of Cmd field) is required send to CG102RS232:

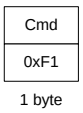

Figure 4-34 Request to enquire the ZDCP version of CG102RS232

If the ZDCP of the CG102RS232 is version 1.2 or above, following PDU is expected to be returned from CG102RS232:

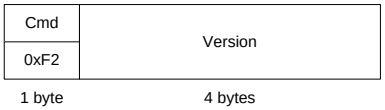

Figure 4-35 Responding PDU of ZDCP version enquiry

In the response of ZDCP version enquiry, 4 bytes of ASCII code are following the Cmd field. The 4-byte ASCII code is used to indicate the version number of the ZDCP within the CG102RS232. The first two ASCII code are the version number before the decimal point and the last two are the version number after the decimal point. For example, when the version of ZDCP is 1.2, the response ASCII code will be {‘0’,’1’,’0’,’2’} (Figure 4-27).

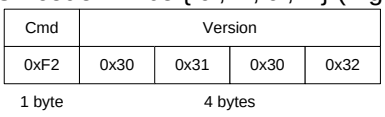

Figure 4-36 Responding PDU of ZDCP version enquiry form a CG102RS232 with ZDCP version 1.2

4.6 Operation Policy

In the operation of devices control and status request through the CG102RS232, following policies are strongly recommended.

In the operation of single device control/status request, the CG102RS232 will response to the external system within two seconds (refer as the Operation Period) when receiving the request from external system. The interval of Operation Period is depending on the traffic loading of ZigBee network and distance of the target device. The external system is not recommended to send any request to CG102RS232 within the Operation Period. Under normal operation:

- if the target device is reachable by the CG102RS232 directly, the CG102RS232 may response to the external system within 0.5 second;
- if the target device is reachable by the CG102RS232 through multiple ZigBee network hops, the CG102RS232 may response to the external system within two seconds;
- if the target device is unreachable by the ZigBee network, the CG102RS232 will response to the external system after two seconds with the FAILURE response.

After the external system received the response from CG102RS232, the external system may send another request to the CG102RS232 immediately.