0 Preface

About this demo,QCA4020 control on or off status of bulb, color temperature and dim via Zigbee protocol. The purport of this article is to elaborate how to build Zigbee network, and Zigbee bulb can connect it automatically once it searched the network. You will find we have complished various of controlling interface in this demo, and we can control different status of bulb via these interfaces.

1 Text

1.1 Initialize Zigbee protocol, and build Zigbee network.

Global variable "flag_form" is used to judge whether the Zigbee network is successfull or not.

```
/*....../quartz/demo/QCLI_demo/src/qcli/pal.c*/

+int flag_form = 0;

+void init_Zigbee()
+{
+       d_cmd_ZB_Initialize(0,NULL);
+
+       QCLI_Parameter_t param_setBIB[4];
+       param_setBIB[0].Integer_Value = 0x100f;
+       param_setBIB[0].Integer_Is_Valid = true;
+       param_setBIB[1].Integer_Value = 1;
+       param_setBIB[1].Integer_Is_Valid = true;
+       param_setBIB[2].Integer_Value = 4;
+       param_setBIB[2].Integer_Is_Valid = true;
+       param_setBIB[3].Integer_Value = 0;
+       param_setBIB[3].Integer_Is_Valid = true;
+       d_cmd_ZB_SetBIB(4, param_setBIB);
+
+       QCLI_Parameter_t param_CE[2];
+       param_CE[0].Integer_Value = 1;
+       param_CE[0].Integer_Is_Valid = true;
+       param_CE[1].Integer_Value = 6;
+       param_CE[1].Integer_Is_Valid = true;
+       d_cmd_ZB_CL_CreateEndpoint(2, param_CE);
+
```

```
+       QCLI_Parameter_t param_form[1];
+       param_form[0].Integer_Value = 1;
+       param_form[0].Integer_Is_Valid = true;
+       d_cmd_ZB_Form(1, param_form);
+
+       while(true)
+       {
+               if(flag_form == 1)
+               {
+                       zigbee_printf("Form Zigbee network success!");
+                       break;
+               }
+               qurt_thread_sleep(50);
+               zigbee_printf("wait for forming Zigbee network!");
+       }
+}
+
```

1.2 Zigbee devices connection

The function "Connect_ZigbeeDevice()" is to wait connection of Zigbee bulb, "flag_connect" is used to judge whether the bulb join into the network successfully or not.

```
/*....../quartz/demo/QCLI_demo/src/qcli/pal.c*/
+int flag_connect = 0;

+void Connect_ZigbeeDevice()
+{
+       while(true)
+       {
+               if(flag_connect == 1)
+               {
+                       zigbee_printf("Device join success!");
+                       break;
+               }
+               qurt_thread_sleep(50);
+               zigbee_printf("wait that blub join in Zigbee
network!");
+       }
+}
```

1.3 Zigbee network architecture building

We can set "flag_form" to 1 after we have built network architecture successfully.

```
/*..../quartz/demo/QCLI_demo/src/zigbee/zigbee_demo.c*/
+extern int flag_form;

static void ZB_Event_CB()
{
    ...
+               flag_form = 1;
    ...
}
```

1.4 Zigbee network devices connection

We can set "flag_form" to 1 after the devices connect successfully.

```
/*....../quartz/demo/QCLI_demo/src/zigbee/zdp_demo.c*/

+extern int flag_connect;

static void ZB_ZDP_Event_CB(qapi_ZB_Handle_t ZB_Handle, const
qapi_ZB_ZDP_Event_t *ZDP_Event_Data, uint32_t CB_Param)
{
    ...
+                       QCLI_Parameter_t param[3];
+                       param[0].Integer_Value = 2;
+                       param[0].Integer_Is_Valid = true;
+                       param[1].Integer_Value = ZDP_Event_Data-
>Event_Data.Device_Annce.NwkAddr;
+                       device_addr = ZDP_Event_Data-
>Event_Data.Device_Annce.NwkAddr;
+                       QCLI_Printf(ZDP_Demo_Context.QCLI_Handle, "
NetworkAddress:  %d\n",
                                        ZDP_Event_Data-
>Event_Data.Device_Annce.NwkAddr);
+                       param[1].Integer_Is_Valid = true;
```

```
+                       param[2].Integer_Value = 1;
+                       param[2].Integer_Is_Valid = true;
+                       d_cmd_ZB_AddDevice(3, param);
+                       flag_connect = 1;
    ...
}
```

1.5 Realizing on/off function of Zigbee bulb

The function "Control_ZigbeeLightOnOff" is to control on/off of the bulb.

```
/*....../quartz/demo/QCLI_demo/src/qcli/pal.c*/
+int Control_ZigbeeLightOnOff(int on)
+{
+       if(on > 1)
+       {
+               QCLI_Parameter_t param_on[2];
+               param_on[0].Integer_Value = 1;
+               param_on[0].Integer_Is_Valid = true;
+               param_on[1].Integer_Value = 1;
+               param_on[1].Integer_Is_Valid = true;
+               d_cmd_ZCL_OnOff_On(2, param_on);
+       }
+       else{
+               QCLI_Parameter_t param_off[2];
+               param_off[0].Integer_Value = 1;
+               param_off[0].Integer_Is_Valid = true;
+               param_off[1].Integer_Value = 1;
+               param_off[1].Integer_Is_Valid = true;
+               d_cmd_ZCL_OnOff_Off(2, param_off);
+       }
+
+       return 0;
+}
```

1.6 Realizing to adjust the dim of bulb

Controlling the dim of bulb via the function "Control_ZigbeeLightLevel".

```
/*....../quartz/demo/QCLI_demo/src/qcli/pal.c*/

+int Control_ZigbeeLightLevel(int level)
+{
+               QCLI_Parameter_t param_level[5];
+               param_level[0].Integer_Value = 1;
+               param_level[0].Integer_Is_Valid = true;
+               param_level[1].Integer_Value = 1;
+               param_level[1].Integer_Is_Valid = true;
+               param_level[2].Integer_Value = 1;
+               param_level[2].Integer_Is_Valid = true;
+               param_level[3].Integer_Value = level;
+               param_level[3].Integer_Is_Valid = true;
+               param_level[4].Integer_Value = 5;
+               param_level[4].Integer_Is_Valid = true;
+               d_cmd_ZCL_LevelControl_MoveToLevel(5, param_level);
+
+               return 0;
+}
```

1.7 Realizing to adjust color temperature

Controlling color temperature of bulb via function "Control_ZigbeeColorTemperaturn".

```
/*....../quartz/demo/QCLI_demo/src/qcli/pal.c*/

+int Control_ZigbeeColorTemperaturn(int temp)
+{
+               QCLI_Parameter_t param_temp[4];
+               param_temp[0].Integer_Value = 1;
+               param_temp[0].Integer_Is_Valid = true;
+               param_temp[1].Integer_Value = 1;
+               param_temp[1].Integer_Is_Valid = true;
+               param_temp[2].Integer_Value = temp;
+               param_temp[2].Integer_Is_Valid = true;
+               param_temp[3].Integer_Value = 5;
+               param_temp[3].Integer_Is_Valid = true;
+               d_cmd_ZCL_ColorControl_MoveToColorTemp(4, param_temp);
+
```

```
+                return 0;
+}
```

1.8 Saving Internet address of bulb

Here is to save the network address,and the location of code is ".....quartz/demo/QCLI_demo/src/zigbee/zigbee_demo.c".

```
+uint64_t  device_addr = 0;
static QCLI_Command_Status_t cmd_ZB_AddDevice(uint32_t Parameter_Count,  QCLI_Parameter_t *Parameter_List)
{
    ...
-         (Verify_Integer_Parameter(&Parameter_List[0], QAPI_ZB_ADDRESS_MODE_GROUP_ADDRESS_E, QAPI_ZB_ADDRESS_MODE_EXTENDED_ADDRESS_E)) &&
-         (Hex_String_To_ULL(Parameter_List[1].String_Value, &DevAddr)))
+         (Verify_Integer_Parameter(&Parameter_List[0], QAPI_ZB_ADDRESS_MODE_GROUP_ADDRESS_E, +QAPI_ZB_ADDRESS_MODE_EXTENDED_ADDRESS_E))/*&&
+         (Hex_String_To_ULL(Parameter_List[1].String_Value, &DevAddr))*/)
       {
+                DevAddr = device_addr;
    ...
}
```

1.9 Function encapsulation

Last but not least, we need to do function encapsulation process,the code related to color temperature is located at "..../quartz/demo/QCLI_demo/src/zigbee/clusters/zcl_colorcontrol_demo.c".

```
+QCLI_Command_Status_t d_cmd_ZCL_ColorControl_MoveToColorTemp(uint32_t Parameter_Count, QCLI_Parameter_t *Parameter_List);

+QCLI_Command_Status_t d_cmd_ZCL_ColorControl_MoveToColorTemp(uint32_t
```

```
Parameter_Count, QCLI_Parameter_t *Parameter_List)
+{
+       return cmd_ZCL_ColorControl_MoveToColorTemp(Parameter_Count,
Parameter_List);
+}
```

The code related to color temperature is located at ".....quartz/demo/QCLI_demo/src/zigbee/clusters/zcl_levelcontrol_demo.c".

```
+QCLI_Command_Status_t d_cmd_ZCL_LevelControl_MoveToLevel(uint32_t
Parameter_Count, QCLI_Parameter_t *Parameter_List);

+QCLI_Command_Status_t d_cmd_ZCL_LevelControl_MoveToLevel(uint32_t
Parameter_Count, QCLI_Parameter_t *Parameter_List)
+{
+       return cmd_ZCL_LevelControl_MoveToLevel(Parameter_Count,
Parameter_List);
+}
+
    ...
+               param_temp[3].Integer_Is_Valid = true;
+               d_cmd_ZCL_ColorControl_MoveToColorTemp(4, param_temp);
+
+               return 0;
+}
```

The code related to color on or off status is located at ".../quartz/demo/QCLI_demo/src/zigbee/clusters/zcl_onoff_demo.c"

```
+QCLI_Command_Status_t   d_cmd_ZCL_OnOff_On(uint32_t   Parameter_Count,
QCLI_Parameter_t *Parameter_List);
+QCLI_Command_Status_t  d_cmd_ZCL_OnOff_Off(uint32_t  Parameter_Count,
QCLI_Parameter_t *Parameter_List);

+QCLI_Command_Status_t   d_cmd_ZCL_OnOff_On(uint32_t   Parameter_Count,
QCLI_Parameter_t *Parameter_List)
+{
```

```
+       return cmd_ZCL_OnOff_On(Parameter_Count, Parameter_List);
+}

+QCLI_Command_Status_t  d_cmd_ZCL_OnOff_Off(uint32_t  Parameter_Count, QCLI_Parameter_t *Parameter_List)
+{
+       return cmd_ZCL_OnOff_Off(Parameter_Count, Parameter_List);
+}
```

The code related to Endpoint creation is located at ".....quartz/demo/QCLI_demo/src/zigbee/zcl_demo.c".

```
+QCLI_Command_Status_t d_cmd_ZB_CL_CreateEndpoint(uint32_t Parameter_Count, QCLI_Parameter_t *Parameter_List);

+QCLI_Command_Status_t d_cmd_ZB_CL_CreateEndpoint(uint32_t Parameter_Count, QCLI_Parameter_t *Parameter_List)
+{
+       return cmd_ZB_CL_CreateEndpoint(Parameter_Count, Parameter_List);
+}
```

Encapsulate below functions "d_cmd_ZB_AddDevice", "d_cmd_ZB_AddDevice", "d_cmd_ZB_SetBIB", "d_cmd_ZB_Initialize".

```
+QCLI_Command_Status_t d_cmd_ZB_AddDevice(uint32_t Parameter_Count,
 QCLI_Parameter_t *Parameter_List);
+QCLI_Command_Status_t d_cmd_ZB_AddDevice(uint32_t Parameter_Count, QCLI_Parameter_t *Parameter_List);
+QCLI_Command_Status_t d_cmd_ZB_SetBIB(uint32_t Parameter_Count, QCLI_Parameter_t *Parameter_List);
+QCLI_Command_Status_t d_cmd_ZB_Initialize(uint32_t Parameter_Count, QCLI_Parameter_t *Parameter_List);

+QCLI_Command_Status_t d_cmd_ZB_Initialize(uint32_t Parameter_Count, QCLI_Parameter_t *Parameter_List)
+{
```

```
8
9  +       return cmd_ZB_Initialize(Parameter_Count, Parameter_List);
10 +}
11
12 +QCLI_Command_Status_t d_cmd_ZB_AddDevice(uint32_t Parameter_Count,
    QCLI_Parameter_t *Parameter_List)
13 +{
14 +       return cmd_ZB_AddDevice(Parameter_Count, Parameter_List);
15 +}
16
17 +QCLI_Command_Status_t d_cmd_ZB_Form(uint32_t Parameter_Count,
   QCLI_Parameter_t *Parameter_List)
18 +{
19 +       return cmd_ZB_Form(Parameter_Count, Parameter_List);
20 +}
21
22 +QCLI_Command_Status_t d_cmd_ZB_SetBIB(uint32_t Parameter_Count,
   QCLI_Parameter_t *Parameter_List)
23 +{
24 +       return cmd_ZB_SetBIB(Parameter_Count, Parameter_List);
25 +}
26
```