0 Perface

About this demo,for QCA4020 need to connect Azure,therefore,QCA4020 development kit need to connnect Internet.The purport of this article is to elaborate that how to set QCA4020 and build connection with Internet automatically before the QCA4020 connecting to Azure.

1 Text

Firstly define a function "init_Wi-Fi" which will be called by "QCLI_Thread()" .The coding modify process is exicuted in "/quartz/demo/QCLI_demo/src/qcli/pal.c".

```
+int flag_connectwifi = 0;
+int flag_newwlan = 0;

+void init_WiFi()
+{
+   enable_Wlan();
+   get_IP();
+   add_DNS();
+   add_SNTP();
+}
```

1.1 Enable_Wlan

Enable WLAN and connect to AP.

Here now we need to add a judgement,and confirm whether it connect successfully or not.The code is as below:

```
+void enable_Wlan()
+{
+       enableWlan(0, NULL);
+
+       QCLI_Parameter_t param_pass[1];
+       param_pass[0].String_Value = "13657318740";
+       param_pass[0].Integer_Is_Valid = false;
+       setWpaPassphrase(1, param_pass);
```

```
+
+       QCLI_Parameter_t param_param[3];
+       param_param[0].String_Value = "WPA2";
+       param_param[0].Integer_Is_Valid = false;
+       param_param[1].String_Value = "CCMP";
+       param_param[1].Integer_Is_Valid = false;
+       param_param[2].String_Value = "CCMP";
+       param_param[2].Integer_Is_Valid = false;
+       setWpaParams(3, param_param);
+
+       QCLI_Parameter_t param_connect[1];
+       param_connect[0].String_Value = "xiaomi";
+       param_connect[0].Integer_Is_Valid = false;
+       connect(1, param_connect);
+
+
+       while(true)
+       {
+               if(flag_connectwifi == 1)
+               {
+                       break;
+               }
+               qurt_thread_sleep(30);
+               zigbee_printf("wait for connect wifi!");
+       }
+}
```

1.2 Get_IP

Aquire IP address after connecting successfully,here we still need to add a judgement to ensure whether it is succeed or not.

```
+void get_IP()
+{
+       QCLI_Parameter_t param_dhcp[2];
+       param_dhcp[0].String_Value = "wlan0";
+       param_dhcp[0].Integer_Is_Valid = false;
+       param_dhcp[1].String_Value = "new";
+       param_dhcp[1].Integer_Is_Valid = false;
+       d_dhcpv4c(2, param_dhcp);
```

```
+       while(true)
+       {
+               if(flag_newwlan == 1)
+               {
+                       break;
+               }
+               qurt_thread_sleep(30);
+               zigbee_printf("wait for new wlan!");
+       }
+}
```

1.3 Add_DNS

Enable DNS app and add DNS server.

```
+void add_DNS()
+{
+       QCLI_Parameter_t param_dnscstart[1];
+       param_dnscstart[0].String_Value = "start";
+       param_dnscstart[0].Integer_Is_Valid = false;
+       d_dnsc(1, param_dnscstart);
+
+       QCLI_Parameter_t param_dnscaddsrv[1];
+       param_dnscaddsrv[0].String_Value = "addsrv";
+       param_dnscaddsrv[0].Integer_Is_Valid = false;
+       param_dnscaddsrv[1].String_Value = "8.8.8.8";
+       param_dnscaddsrv[1].Integer_Is_Valid = false;
+       d_dnsc(2, param_dnscaddsrv);
+}
```

1.4 Add_SNTP

Enable SNTP app and add public SNTP server.

```
+void add_SNTP()
+{
+       QCLI_Parameter_t param_sntpcstart[1];
+       param_sntpcstart[0].String_Value = "start";
```

```
+       param_sntpcstart[0].Integer_Is_Valid = false;
+       d_sntpc(1, param_sntpcstart);
+
+       QCLI_Parameter_t param_sntpcaddsvr[1];
+       param_sntpcaddsvr[0].String_Value = "addsvr";
+       param_sntpcaddsvr[0].Integer_Is_Valid = false;
+       param_sntpcaddsvr[1].String_Value = "24.56.178.140";
+       param_sntpcaddsvr[1].Integer_Is_Valid = false;
+       d_sntpc(2, param_sntpcaddsvr);
+
+}
```

1.5 Set IP status

Wi-Fi function now is available,we need to set “flag_newwlan” to 1 once we get IP successfully.The location of code is ".../quartz/demo/QCLI_demo/src/net/netcmd.c".

```
+extern int flag_newwlan;

static int32_t ipconfig_dhcpc_success_cb()
{
     ...
+    QCLI_Printf(qcli_net_handle, "in ipconfig_dhcpc_success_cb ------
- \n");
+    flag_newwlan = 1;
     ...
}
```

1.6 Set the Wi-Fi status

When Wi-Fi connection is successfully,we need to set “flag_connect”Wi-Fi to 1,and the location of code is "...../quartz/demo/QCLI_demo/src/Wi-Fi/Wi-Fi_cmd_handler.c".

```
+extern int flag_connectwifi;

wifi_connect_handler()
{
        ...
```

```
+       QCLI_Printf(qcli_wlan_group, "test----------------------------
\r\n");
+       flag_connectwifi = 1;

+       QCLI_Printf(qcli_wlan_group, "-----in set_passphrase %s\n",
passphrase);
        ...
}
```

1.7 Function encapsulation

To realize “d_dhcpv4c”, “d_sntpc” and “d_dnsc”functions these three,actually it`s a encapsulation to existed functions.We can call them using external files.The location of code is "...../quartz/demo/QCLI_demo/src/net/netcmd.c".

```
+QCLI_Command_Status_t d_dhcpv4c(uint32_t Parameter_Count,
QCLI_Parameter_t *Parameter_List);
+QCLI_Command_Status_t d_sntpc(uint32_t Parameter_Count,
QCLI_Parameter_t *Parameter_List);
+QCLI_Command_Status_t d_dnsc(uint32_t Parameter_Count,
QCLI_Parameter_t *Parameter_List);




+QCLI_Command_Status_t d_dhcpv4c(uint32_t Parameter_Count,
QCLI_Parameter_t *Parameter_List)
+{
+       return dhcpv4c(Parameter_Count, Parameter_List);
+}
+

+QCLI_Command_Status_t d_sntpc(uint32_t Parameter_Count,
QCLI_Parameter_t *Parameter_List)
+{
+       return sntpc(Parameter_Count, Parameter_List);
+}
+

+QCLI_Command_Status_t d_dnsc(uint32_t Parameter_Count,
QCLI_Parameter_t *Parameter_List)
```

```
+{
+       return dnsc(Parameter_Count, Parameter_List);
+}
+
```