User Manual

DA16200 Wi-Fi Connection Notification

UM-WI-029

Abstract

This document describes how to use the Wi-Fi connection notification function of the DA16200.

Contents

1   Terms and Definitions

MCU   Microcontroller Unit

2   References

[1]   <DA16200>, SDK Programmer Guide, Dialog Semiconductor

3   Introduction

This document describes how to implement a customer's program in the DA16200.

The DA16200 SDK provides the Wi-Fi connection status notification function when run in the STA mode. Using this function, customer/developer can implement their own operation to let MCU or other hardwired device know the notified Wi-Fi connection status.

4   How to Create New F/W with Wi-Fi Notification Function

The DA16200 SDK provides Wi-Fi connection status notification function as a compile feature.

~/SDK/src/customer/config_generic_sdk.h

> *#undef __SUPPORT_WIFI_CONN_CB__*

Customer/developer can change the above compile feature as *#define* to create new F/W.

Also, user should implement their own operation. For example, sending specified event or data to MCU or other hardwired device using the defined protocol.

The released generic DA16200 SDK, only shows a notification message on console terminal.

```
### Customer Call-back : Wi-Fi disconnected ( reason_code = 0x3 ) ...

!!! No selected network !!!
Fast scan , freq = 2412. num_ssids = 1
Fast scan , freq = 2412. num_ssids = 1
!!! No selected network !!!
>>> Selected BSS aa:ff:88:11:ff:01 ssid='DA16200_11FF01' (-4)
>>> Network Interface (wlan0) : UP
>>> Associated with aa:ff:88:11:ff:01

Connection COMPLETE to aa:ff:88:11:ff:01

-- DHCP Client WLAN0: SEL


### Customer Call-back : Success to connect Wi-Fi ...

-- DHCP Client WLAN0: REQ
-- DHCP Client WLAN0: BOUND
        Assigned addr   : 10.0.0.2
            netmask   : 255.255.255.0
            gateway   : 10.0.0.1
            DNS addr   : 0.0.0.0

        DHCP Server IP : 10.0.0.1
        Lease Time     : 00h 30m 00s
        Renewal Time   : 00h 15m 00s
```

5 Sequence for Wi-Fi Connection Status Notification

The DA16200 Wi-Fi connection status notification function runs as a library and customer/developer needs to add their own code to send the result to MCU or another hardwired device.

To use this function, the DA16200 SDK already includes all source codes as the following sequences.

● Register call-back functions to use this feature
~/SDK/src/customer/system_start.c

```
Int system_start(void)
{
    … …

    /* Regist Wi-Fi connect/disconnect status notify call-back functions */
    Regist_wifi_notify_cb()
    … …
}
```

● Create a mutex-sema flags to prevent simultaneous access and register connect/disconnect callback function
~/SDK/common/main/util_api.c

```
void regist_wifi_notify_cb(void)
{
    ULONG status;

    wifi_conn_notify_mutex = malloc(sizeof(TX_MUTEX));
    if (wifi_conn_notify_mutex == NULL)
    {
        PRINTF("\n>>> Failed to allocate wifi_conn_notify_mutex buffer !\n");
        return;
    }
    memset(wifi_conn_notify_mutex, 0, sizeof(TX_MUTEX));

    status = tx_mutex_create(wifi_conn_notify_mutex,
                             "wifi_conn_cb_mutex", TX_NO_INHERIT);
    if (status != TX_SUCCESS)
    {
        PRINTF("\n>>> Failed to create Wi-Fi connection notify cb mutex !\n");
        return;
    }

    /* Wi-Fi connection call-back */
    wifi_conn_notify_cb_regist(wifi_conn_cb);

    /* Wi-Fi disconnection call-back */
    wifi_disconn_notify_cb_regist(wifi_disconn_cb);
}
```

● Wi-Fi connection function
When Wi-Fi is connected, this function sends notification "wifi_conn_flag = TRUE" and this flag is checked and used in customer/developer function.

```
static void wifi_conn_cb(void)
{
    ULONG status;

    /* Wait until 3 seconds to get mutex */
    status = tx_mutex_get(wifi_conn_notify_mutex, 300);
    if (status != TX_SUCCESS)
    {
        PRINTF("\nFailed to get wifi_conn_notify_mutex during 3 secs !\n");
```

```
        return;
    }

    wifi_conn_flag = TRUE;

    tx_mutex_put(wifi_conn_notify_mutex);
}
```

- Wi-Fi disconnection function
  When Wi-Fi is disconnected, this function notifies "wifi_disconn_flag = TRUE" and sends the reasone_code "wifi_disconn_reason".
  This flag and reason_code are checked and used in customer/developer function.

```
static void wifi_disconn_cb(ULONG reason_code)
{
    ULONG status;

    /* Wait until 3 seconds to get mutex */
    status = tx_mutex_get(wifi_conn_notify_mutex, 300);
    if (status != TX_SUCCESS)
    {
        PRINTF("\nFailed to get wifi_conn_notify_mutex during 3 sess !\n");
        return;
    }

    wifi_disconn_flag   = TRUE;
    wifi_disconn_reason = reason_code;

    tx_mutex_put(wifi_conn_notify_mutex);
}
```

NOTE
No other changes should be made to the above sequences and functions.

6 User Function for Wi-Fi Connection Status

For Wi-Fi connection status notification function, customer/developer should add their own operation in the DA16200 SDK before creating a new image.

In the DA16200 SDK, the customer/developer Wi-Fi connection status notification operation to MCU or to hardwired device are created as independent threads not to affect the basic Wi-Fi module operation.

~/SDK/src/customer/user_apps.c

```
const app_thread_info_t user_apps_table[] = {
/* name, func, stack_size, priority, net_chk_flag, dpm_flag, port_no, run_sys_mode
*/

    … …

#if defined ( __SUPPORT_WIFI_CONN_CB__ )
  { WIFI_CONN,      customer_wifi_conn,    1024,   USER_PRI_APP(0), FALSE, FALSE,
UNDEF_PORT, RUN_ALL_MODE    },
  { WIFI_DISCONN,  customer_wifi_disconn, 1024,   USER_PRI_APP(0), FALSE, FALSE,
UNDEF_PORT, RUN_ALL_MODE    },
#endif  // __SUPPORT_WIFI_CONN_CB__

    … …

    { NULL, NULL,   0, 0, FALSE, FALSE, UNDEF_PORT, 0   }
};
```

NOTE
No other changes should be made to the above two thread creation items in user_apps_table except for the stack_size and the thread running priority.

● Wi-Fi connection status function:
  ~/SDK/src/customer/user_apps.c
  In the provided source code, customer/developer should only change the **Event/Data TX** part to send notification to MCU or hardwired device. Don't clear the flag or change the mutex handling.

```
static void customer_wifi_conn(ULONG arg)
{
    while (1)
    {
        if (wifi_conn_flag == TRUE)
        {
#if 0
            //
            // Need customer's code about this event
            //
#else
            PRINTF("\n\n");
            PRINTF("### Customer Call-back : Success to connect Wi-Fi ...\n");
            PRINTF("\n");
#endif  // 0

            /*
             * Customer tunning value :
             *  Wait 100msec until sync with MCU
             */
            tx_thread_sleep(10);

            /* Clear event flag */
```

```
            tx_mutex_get(wifi_conn_notify_mutex, 300);

            wifi_conn_flag = FALSE;

            tx_mutex_put(wifi_conn_notify_mutex);
        }

        /* loop time delay : 10 msec */
        tx_thread_sleep(1);
    }
}
```

- Wi-Fi disconnection status function:
  ~/SDK/src/customer/user_apps.c
  In the provided source code, customer/developer should only change the **Event/Data TX** part to send notification to MCU or hardwired device. Don't clear the flag or change the mutex handling.

```
static void customer_wifi_disconn(ULONG arg)
{
    while (1)
    {
        if (wifi_disconn_flag == TRUE)
        {
#if 0
            //
            // Need customer's code about this event
            //
#else
            PRINTF("\n\n");
            PRINTF("### Customer Call-back : Wi-Fi disconnected ( reason_code =
0x%x ) ...\n", wifi_disconn_reason);
            PRINTF("\n");
#endif  // 0

            /*
             * Customer tunning value :
             *  Wait 100msec until sync with MCU
             */
            tx_thread_sleep(10);

            /* Clear event flag */
            tx_mutex_get(wifi_conn_notify_mutex, 300);

            wifi_disconn_reason = 0;
            wifi_disconn_flag = FALSE;

            tx_mutex_put(wifi_conn_notify_mutex);
        }

        /* loop time delay : 10 msec */
        tx_thread_sleep(1);
    }
}
```

Revision History

Revision	Date	Description
1.0	02-Sept-2020	Initial version.

Status Definitions

Status	Definition
DRAFT	The content of this document is under review and subject to formal approval, which may result in modifications or additions.
APPROVED or unmarked	The content of this document has been approved for publication.

Disclaimer

Unless otherwise agreed in writing, the Dialog Semiconductor products (and any associated software) referred to in this document are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of a Dialog Semiconductor product (or associated software) can reasonably be expected to result in personal injury, death or severe property or environmental damage. Dialog Semiconductor and its suppliers accept no liability for inclusion and/or use of Dialog Semiconductor products (and any associated software) in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Information in this document is believed to be accurate and reliable. However, Dialog Semiconductor does not give any representations or warranties, express or implied, as to the accuracy or completeness of such information. Dialog Semiconductor furthermore takes no responsibility whatsoever for the content in this document if provided by any information source outside of Dialog Semiconductor.

Dialog Semiconductor reserves the right to change without notice the information published in this document, including, without limitation, the specification and the design of the related semiconductor products, software and applications. Notwithstanding the foregoing, for any automotive grade version of the device, Dialog Semiconductor reserves the right to change the information published in this document, including, without limitation, the specification and the design of the related semiconductor products, software and applications, in accordance with its standard automotive change notification process.

Applications, software, and semiconductor products described in this document are for illustrative purposes only. Dialog Semiconductor makes no representation or warranty that such applications, software and semiconductor products will be suitable for the specified use without further testing or modification. Unless otherwise agreed in writing, such testing or modification is the sole responsibility of the customer and Dialog Semiconductor excludes all liability in this respect.

Nothing in this document may be construed as a license for customer to use the Dialog Semiconductor products, software and applications referred to in this document. Such license must be separately sought by customer with Dialog Semiconductor.

All use of Dialog Semiconductor products, software and applications referred to in this document is subject to Dialog Semiconductor's Standard Terms and Conditions of Sale, available on the company website (www.dialog-semiconductor.com) unless otherwise stated.

Dialog, Dialog Semiconductor and the Dialog logo are trademarks of Dialog Semiconductor Plc or its subsidiaries. All other product or service names and marks are the property of their respective owners.

© 2020 Dialog Semiconductor. All rights reserved.

RoHS Compliance

Dialog Semiconductor's suppliers certify that its products are in compliance with the requirements of Directive 2011/65/EU of the European Parliament on the restriction of the use of certain hazardous substances in electrical and electronic equipment. RoHS certificates from our suppliers are available on request.

Contacting Dialog Semiconductor

United Kingdom (Headquarters)	North America	Hong Kong	China (Shenzhen)
Dialog Semiconductor (UK) LTD	*Dialog Semiconductor Inc.*	*Dialog Semiconductor Hong Kong*	*Dialog Semiconductor China*
Phone: +44 1793 757700	Phone: +1 408 845 8500	Phone: +852 2607 4271	Phone: +86 755 2981 3669
Germany	Japan	Korea	China (Shanghai)
Dialog Semiconductor GmbH	*Dialog Semiconductor K. K.*	*Dialog Semiconductor Korea*	*Dialog Semiconductor China*
Phone: +49 7021 805-0	Phone: +81 3 5769 5100	Phone: +82 2 3469 8200	Phone: +86 21 5424 9058
The Netherlands	Taiwan		
Dialog Semiconductor B.V.	*Dialog Semiconductor Taiwan*		
Phone: +31 73 640 8822	Phone: +886 281 786 222		
Email:	Web site:		
enquiry@diasemi.com	www.dialog-semiconductor.com		