



WiFi smart switch design

Version: 1.0

Release date: 2016-08-11

© 2015 - 2016 MediaTek Inc.

This document contains information that is proprietary to MediaTek Inc. ("MediaTek") and/or its licensor(s). MediaTek cannot grant you permission for any material that is owned by third parties. You may only use or reproduce this document if you have agreed to and been bound by the applicable license agreement with MediaTek ("License Agreement") and been granted explicit permission within the License Agreement ("Permitted User"). If you are not a Permitted User, please cease any access or use of this document immediately. Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited. THIS DOCUMENT IS PROVIDED ON AN "AS-IS" BASIS ONLY. MEDIATEK EXPRESSLY DISCLAIMS ANY AND ALL WARRANTIES OF ANY KIND AND SHALL IN NO EVENT BE LIABLE FOR ANY CLAIMS RELATING TO OR ARISING OUT OF THIS DOCUMENT OR ANY USE OR INABILITY TO USE THEREOF. Specifications contained herein are subject to change without notice.

Document Revision History

Revision	Date	Description
1.0	30 July 2016	Draft Version

Table of contents

- 1. Introduction..... 3**
 - 1.1. Objective.....3
- 2. Hardware..... 4**
 - 2.1. Component4
 - 2.2. Interface4
 - 2.3. Power Scheme4
 - 2.4. Schematic Diagram4
- 3. Software 5**
 - 3.1. GPIO5
 - 3.1.1. GPIO config5
 - 3.1.2. GPIO control5
 - 3.2. TCP client in device6
 - 3.3. TCP server in PC7
- 4. Test & Debug 8**
 - 4.1. Device8
 - 4.2. PC.....8
- 5. Limitation 9**
 - 5.1. Cloud.....9

1 . Introduction

1.1. Objective

The purpose of this document is to describe the design of WIFI smart switch.

In this document, you can create your own smart switch project by using MT7687 and a relay, and controlled by socket cmd send from PC after connection. This design can be used in smart home or other scene that need a remote digital signal control.

In this document, HW design and SW coding should be introduced for reader. And reader could build the project in a short time

2. Hardware

2.1. Component

MCU: MediaTek MT7687

WiFi: embedded in MediaTek MT7687 (WiFi SOC)

Peripheral: a relay

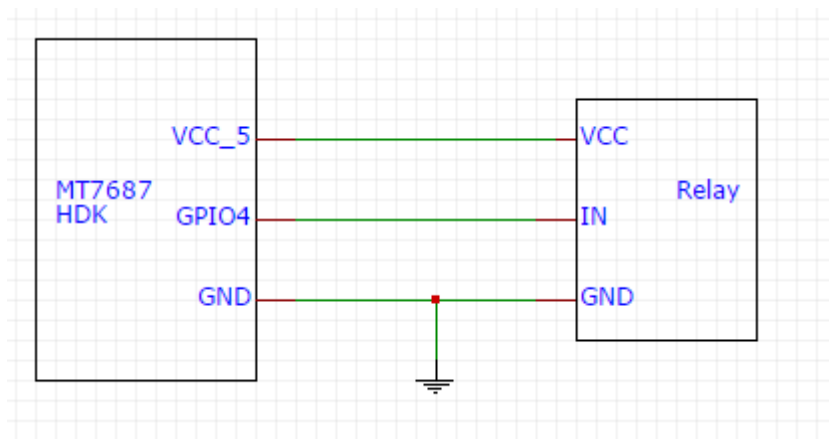
2.2. Interface

Pin Name	76x7 Pin Function	Peripheral Pin Name	Pin description	I/O	level shift
GPIO4	GPIO4	Relay input	GPIO for relay control	output	no need

2.3. Power Scheme

Power name	Power voltage	Power supply
VBUS(for mt7687)	5V	USB
VCC(for relay)	5V	76X7_5V

2.4. Schematic Diagram



3. Software

Software is contain GPIO driver and tcp client. Based on SDK 3.3.1 .

3.1. GPIO

3.1.1. GPIO config

```
void GPIO_Switch_Init()
{
    hal_gpio_status_t ret;
    hal_pinmux_status_t ret_pinmux_status;
    ret = hal_gpio_init(HAL_GPIO_4);
    ret_pinmux_status = hal_pinmux_set_function(HAL_GPIO_4, HAL_GPIO_4_GPIO4); //set the pin to
work in GPIO mode
    ret = hal_gpio_set_direction(HAL_GPIO_4, HAL_GPIO_DIRECTION_OUTPUT);
}
}
```

3.1.2. GPIO control

```
void switch_Ctrl_For_On()
{
    while(1)
    {
        if( xSemaphoreTake( xSemaphore_on, portMAX_DELAY ) == pdTRUE )
        {
            if(0 == hal_gpio_set_output(HAL_GPIO_4, HAL_GPIO_DATA_HIGH))
            {
                LOG_I(common, "On signal. Set output High");
            }
        }
    }
}

void switch_Ctrl_For_Off()
{
    while(1)
    {
        if( xSemaphoreTake( xSemaphore_off, portMAX_DELAY ) == pdTRUE )
        {
            if(0 == hal_gpio_set_output(HAL_GPIO_4, HAL_GPIO_DATA_LOW))
            {
                LOG_I(common, "Off signal. Set output Low");
            }
        }
    }
}
```

```

    }
  }
}

```

3.2. TCP client in device

There is a sample project for TCP/UDP in MT7687 SDK3.3.1, project root as follow:

SDK_V3.3.1\project\mt7687_hdk\apps\lwip_socket

You can use this sample to complete the socket connection, modification as follow:

Set the TCP socket connection port as what you want, here I set the sample as 8800.

And the port setting must be the same as TCP server

```

#define SOCK_TCP_SRV_PORT    6500
#define SOCK_UDP_SRV_PORT    6600
#define LWIP_SOCKET_STACK_SIZE (1000/sizeof(portSTACK_TYPE))
#define LWIP_SOCKET_TASK_PRIO 1

```

Set the TCP server IP as what you connect to. The IP address is little-endian.

```

static void tcp_client_test(void)
{
  int s;
  int ret;
  struct sockaddr_in addr;
  int count = 0;
  int rcv_len, rlen;
  char rcv_buf[32] = {0};
  char switch_data = 0;
  struct netif *sta_if = netif_find_by_type(NETIF_TYPE_STA);

  vTaskDelay(5000);
  char send_data[] = "Hello Server!";

  LOG_I(common, "tcp_client_test starts");

  os_memset(&addr, 0, sizeof(addr));
  addr.sin_len = sizeof(addr);
  addr.sin_family = AF_INET;
  addr.sin_port = lwip_htons(SOCK_TCP_SRV_PORT);
  inet_addr_from_ipaddr(&addr.sin_addr, netif_ip4_addr(sta_if));
  addr.sin_addr.s_addr = 0x6C01A8C0;
}

```

For data receive part, you can receive the switch data(send part reserve)

```
while (1) {
    /* Write something */
    ret = lwip_write(s, send_data, sizeof(send_data));
    LOG_I(common, "TCP client write:ret = %d", ret);

    LOG_I(common, "TCP client waiting for data...");
    lwip_recv(s, &switch_data, 1, 0);
    LOG_I(common, "TCP client received data:%s", rcv_buf);

        if('0' == switch_data)
            xSemaphoreGive(xSemaphore_off);
        else if('1' == switch_data)
            xSemaphoreGive(xSemaphore_on);
    vTaskDelay(2000);
}
```

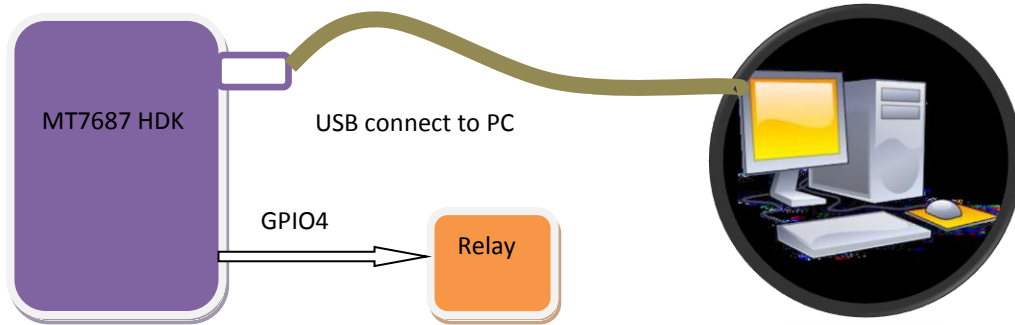
3.3. TCP server in PC

You can use net assist to test the socket connection as a TCP server.

4. Test & Debug

4.1. Device

Connect the device like the following picture:



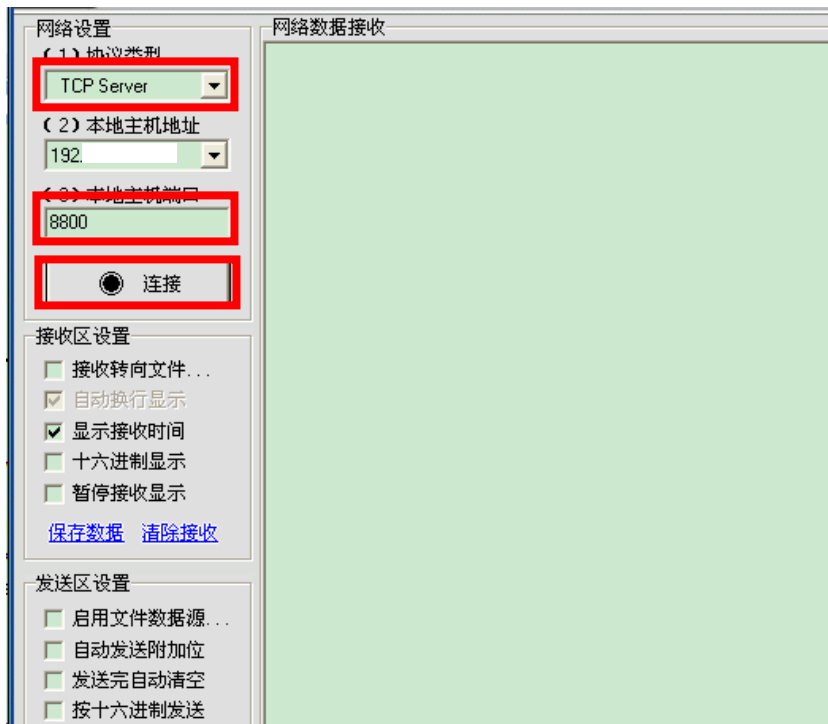
Download the software BIN into MT7687

Then power on the MT7687 HDK

NOTE: Before power on, TCP server in PC side must be setup firstly

4.2. PC

Run TCP net assist and make it work in TCP sever mode. Net assist can get the IP by itself. And set the socket port as setting in MT7687. Then click "connect" button.



Then you can send the "0" or "1" in net assist to MT7687 to control the relay.

5. Limitation

5.1. Cloud

There is only socket sample code in this design. If cloud service is needed, reader can link the cloud service by socket or http service.