This application note describes the basics of sleep management with an example illustrating Talaria TWO wake-up from sleep mode.

# System Sleep Enable & Disable

## System Sleep APIs

### os\_suspend\_enable()

Suspends the system when idle.

|  |
| --- |
| void os\_suspend\_enable(void) |

Calling os\_suspend\_enable() will suspend the system or enable deep sleep, when the processor is idle. Enabling and disabling suspend mode takes additional time, which will affect the real-time response of the system. When an interrupt occurs, the system will wake up even if it is in a suspended state. However, the latency will be more as compared to when the system operates in a non-suspended mode.

### os\_suspend\_disable()

Disables system suspend.

|  |
| --- |
| void os\_suspend\_disable(void) |

When the system is idle, the kernel will place the CPU in low-power mode, ready to swiftly resume execution if an interrupt occurs.

# Code Walkthrough

## Sock\_wake.c

### Overview

The sample code is the path: examples/socket\_wakeup/src/sock\_wake.c is a simple application which demonstrates socket wake-up in sleep mode.

### Connect to a Wi-Fi network

To connect to a Wi-Fi network the following APIs from the Wi-Fi Connection Manager are used:

1. wcm\_create()

This function creates the Wi-Fi network interface using the wcm\_handle pointer.

1. wcm\_notify\_enable()

Enables callbacks of the link and IP address changes.

1. wcm\_add\_network\_profile ()

Asynchronously adds a Wi-Fi network to connect. Currently only one network can be added.

1. wcm\_auto\_connect ()

Enables start or stop auto connection of the device with Wi-Fi.

|  |
| --- |
| static int  wifi\_connection(void)  {  int rval;  struct network\_profile \*profile;  const char \*np\_conf\_path = os\_get\_boot\_arg\_str("np\_conf\_path") ?: NULL;  my\_wcm\_handle = wcm\_create(NULL);  if (my\_wcm\_handle != NULL) {  wcm\_notify\_enable(my\_wcm\_handle, wcm\_notifier, NULL);  if (np\_conf\_path != NULL) {  /\* Create a Network Profile from a configuration file in  \*the file system \*/  rval = network\_profile\_new\_from\_file\_system(&profile, np\_conf\_path);  } else {  /\* Create a Network Profile using BOOT ARGS \*/  rval = network\_profile\_new\_from\_boot\_args(&profile);  }  if (rval < 0) {  pr\_err("could not create network profile %d\n", rval);  return rval;  }  rval = wcm\_add\_network\_profile(my\_wcm\_handle, profile);  if (rval < 0) {  pr\_err("could not associate network profile to wcm %d\n", rval);  return rval;  }  rval = wcm\_auto\_connect(my\_wcm\_handle, 1);  if (rval != WCM\_SUCCESS) {  pr\_err("could enable auto connect for wcm %d\n", rval);  return rval;  }  xSemaphoreTake(connect\_lock, portMAX\_DELAY);  vTaskDelay(1000);  }  return WCM\_SUCCESS;  } |

### Wi-Fi Connection Callback Function

The app\_wcm\_notify\_cb() function enables the callbacks of link and IP address.

|  |
| --- |
| void  wcm\_notifier(void \*ctx, struct os\_msg \*msg)  {  switch (msg->msg\_type) {  case WCM\_NOTIFY\_MSG\_CONNECTED:  os\_printf("wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_CONNECTED\n");  xSemaphoreGive(connect\_lock);  break;  default:  break;  }  os\_msg\_release(msg);  } |

### Server Socket Function

On successfully connecting to the Wi-Fi, the device creates a UDP server in Talaria TWO with server port number 8000. The server will initially be in sleep mode. Here, we register a call- back function to wake up Talaria TWO from sleep mode.

|  |
| --- |
| static void  udp\_server()  {  struct netconn \*conn  = netconn\_new\_with\_callback(NETCONN\_UDP, sock\_event\_cb);  assert(conn);  if (netconn\_bind(conn, IP\_ADDR\_ANY, UDP\_SERVER\_PORT) == ERR\_OK) {  netconn\_set\_nonblocking(conn, 1);  os\_printf("udp server started at port %d\n", UDP\_SERVER\_PORT);  } else {  pr\_err("udp server at %d failed\n", UDP\_SERVER\_PORT);  }  } |

### Socket Event Callback Function

This is the callback function, where Talaria TWO is enabled from sleep and put back to sleep after a brief period of 500ms.

|  |
| --- |
| static void  sock\_event\_cb(struct netconn \*conn, enum netconn\_evt event, u16\_t len)  {  if (event == NETCONN\_EVT\_RCVPLUS) {  os\_printf("NETCONN\_EVT\_RCVPLUS\n");  os\_printf("Waking up\n");  os\_suspend\_disable();  vTaskDelay(500);  os\_printf("sleeping\n");  os\_suspend\_enable();  } } |

### Running the Application

Program sock\_wake.elf *(freertos\_sdk\_x.y\examples\socket\_wakeup\bin)* using the Download tool:

1. Launch the Download tool provided with InnoPhase Talaria TWO FreeRTOS SDK.
2. In the GUI window:
   1. Boot Target: Select the appropriate EVK from the drop-down
   2. ELF Input: Load the sock\_wake.elf by clicking on Select ELF File.
   3. AP Options: Provide the SSID and Passphrase under AP Options to connect to an Access Point.
   4. Programming: Prog RAM or Prog Flash as per requirement.

Launch the Hercules tool for Windows and provide the port number along with the IP address and send the data.

Graphical user interface, application

Description automatically generated

Figure : Hercules Tool - Data transfer

### Expected Output

sock\_wake.elf is created when the code is compiled. Following is the console output when the ELF is programmed onto Talaria TWO.

|  |
| --- |
| Y-BOOT 208ef13 2019-07-22 12:26:54 -0500 790da1-b-7  ROM yoda-h0-rom-16-0-gd5a8e586  FLASH:PWWWWWWAE  Build $Id: git-831e563 $  Flash detected. flash.hw.uuid: 39483937-3207-0061-00a2-ffffffffffff  Bootargs: ssid=test passphrase=12345678  $App:git-5a9f591  SDK Ver: FREERTOS\_SDK\_1.0  Wake From Sock App  addr e0:69:3a:00:15:a8  network profile created for ssid: test  [2.906,467] DISCONNECTED  [3.020,421] CONNECT:72:d7:92:3a:8a:71 Channel:6 rssi:-41 dBm  [3.070,394] MYIP 192.168.122.64  [3.070,558] IPv6 [fe80::e269:3aff:fe00:15a8]-link  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_CONNECTED  URI: udp://192.168.122.64:8000  udp server started at port 8000  NETCONN\_EVT\_RCVPLUS  Waking up  sleeping  NETCONN\_EVT\_RCVPLUS  Waking up  sleeping  NETCONN\_EVT\_RCVPLUS  Waking up  sleeping  NETCONN\_EVT\_RCVPLUS  Waking up  sleeping  NETCONN\_EVT\_RCVPLUS  Waking up  sleeping  NETCONN\_EVT\_RCVPLUS  Waking up  sleeping |