

# CC32xx Sensor Profile Application

## Overview

This application showcases the usage of lowest power mode(Hibernate) for CC3200. Exercising Hibernate can be quite rewarding in terms of power consumption. Especially, when the use case demands the device to be active for only fractional amount of time over long periods(like periodically sending some data over network every one hour). One can also calculate the hibernate current using this application.

[Return to CC31xx & CC32xx Home Page](#)
[Return to CC31xx Sample Applications](#)

## Application details

### Power Management Framework

CC3200 device have multiple power modes, which can be exercised by the user as per the requirement of their application. Power management framework makes it easier to specify the power policy for APPS MCU. All user has to configure is the lowest power mode the device should go into (Hibernate for this application). The framework tries to hide the intricacies of the underlying settings and helps the user to experience the low power mode capabilities of the device in a simple manner. If one wishes to have the custom settings for different power modes or use the framework for a different device, it can be accomplished with changing a couple of files. For more information regarding Power Management Framework, please refer to CC32xx Power Management Framework

### Current Measurement

This application specify the Hibernate as the lowest power mode. When the device is in hibernate, the current values can be as low as in the order of microseconds. The procedure for measuring current can be found at CC3200 Low Power Modes Current Measurement page.

### Program Flow

Most the parameters user will need to modify are specified as MACROS

- SSID\_NAME - Name of the Access Point used for this application.
- GPIO\_SRC\_WKUP - Gpio number used as wake up source.
- APP\_UDP\_PORT - Port number on which the device will send the udp packets.
- HIB\_DUR\_SEC - Time (in seconds) after which the device will come out of Hibernate.
- HIB\_DUR\_NSEC - Time (in addition to HIB\_DUR\_SEC) in nanoseconds after which the device will come out of Hibernate.
- TRAFFIC\_DUR\_SEC - Time (in seconds) for which the device will send the udp packets.
- TRAFFIC\_DUR\_NSEC - Time (in addition to TRAFFIC\_DUR\_SEC ) in nanoseconds for which the device will send the udp packets.
- SERVER\_IP\_ADDRESS - Address of the udp server(other machine on same network) to receive udp packets sent by device.

```
#define SSID_NAME          "cc3200demo"
#define GPIO_13            13
#define GPIO_17            17
#define GPIO_SRC_WKUP      GPIO_13
```

```
#define APP_UDP_PORT          5001
#define HIB_DUR_SEC           60
#define HIB_DUR_NSEC          0
#define TRAFFIC_DUR_SEC       5
#define TRAFFIC_DUR_NSEC      0
#define SERVER_IP_ADDRESS     0xC0A80066
```

**Note:** Security parameters for the AP can be configure inside **WlanConnect** function in "main.c".

**Note:** Whatever Gpio is used as wake up source has to added in **gpio\_list** array in "user\_app\_config.h" and pinmux for the same Gpio has to be added in "pinmux.c".

To start with, the board must be initialized and the power management framework have to be loaded.

```
//
// Board Initialisation
//
BoardInit();

//
// Configure the pinmux settings for the peripherals exercised
//
PinMuxConfig();

//
// Initialize the platform
//
platform_init();

//
// Configuring UART
//
g_tUartHndl = uart_open(PRCM_UARTA0);
```

Which will be followed by creation of task for setting timer and GPIO as wake source from Hibernate

```
//
// setting up timer and gpio as source for wake up from HIBERNATE
//
osi_TaskCreate(TimerGPIONTask,
               (const signed char *)"set wk_src for hibernate",
               OSI_STACK_SIZE, NULL, 1, NULL );
```

**Task: TimerGPIONTask**

1. Starts simplelink, switch to station mode(if not already),set NWP power policy and connects to the AP.
2. If the connection is unsuccessful, jump to 4th step.
3. send UDP packets to Server address(specified as MACRO) for configurable duration(MACRO)
4. Disconnects, stop simplelink, set timer and GPIO as wake sources from Hibernate.
5. Set Power policy for the APPS MCU.

```
//
// starting the simplelink
//
iRetVal = sl_Start(NULL, NULL, NULL);
...
//
// Switch to STA mode if device is not in this mode
//
SwitchToStaMode(iRetVal);

//
// Set the power management policy of NWP
//
iRetVal = sl_WlanPolicySet(SL_POLICY_PM, SL_NORMAL_POLICY, NULL, 0);
...
//
// connecting to the Access Point
//
if(-1 == WlanConnect())
{
    goto no_network_connection;
}else{
    UART_PRINT("Connected to AP\n\r");
}
...
while(g_ucTrafficEnable == 1)
{
    //
    // sending message
    //
    iRetVal = sendto(iSockDesc, pcSendBuff,BUFF_SIZE, 0,
                    (struct sockaddr *)&sServerAddr,sizeof(sServerAddr));
    if(iRetVal < 0)
    {
        UART_PRINT("send error\n\r");
        while(FOREVER);
    }
    ManageDelay(128,BUFF_SIZE);
}
...
```

```
//
// disconnect from the Access Point
//
WlanDisconnect();

//
// stop the simplelink with reqd. timeout value (30 ms)
//
sl_Stop(SL_STOP_TIMEOUT);

//
// setting Timer as one of the wakeup source
//
tTimerHndl = SetTimerAsWkUp();

//
// setting some GPIO as one of the wakeup source
//
tGPIOHndl = SetGPIOAsWkUp();
.
.
//
// Setting up HIBERNATE as the lowest power mode for the system.
//
lp3p0_setup_power_policy(POWER_POLICY_HIBERNATE);
```

## Porting to TI-RTOS

For porting this application on TI-RTOS, please refer to **Steps to build a new application on CC3200 with TI-RTOS:** and **Notes** section on CC32xx\_TI-RTOS page. **cc\_idle\_task\_pm()** is the function needed to be hooked to the idle task.

## Source Files briefly explained

- **main.c** - The main file implementing the sensor profile.
- **lp3p0\_board.c** - Board specific initialization for Power Management framework .
- **lp3p0\_plat\_ops.c** - Board specific APIs like IO parking and framework loading.

### Supporting Files

---

- **pinmux.c** - Generated by the PinMUX utility.

### Common Files

---

- **startup\_ccs.c** - CCS related functions
  - **startup\_ewarm.c** - IAR related functions
  - **wdt\_if.c** - Interface file for Watchdog Timer
-

## Prerequisites

This application facilitates the user to get the current numbers for Hibernate mode. The prerequisites for running this application are the following:

- Ensure that the device is in Station mode
- Delete all previously stored AP profiles.

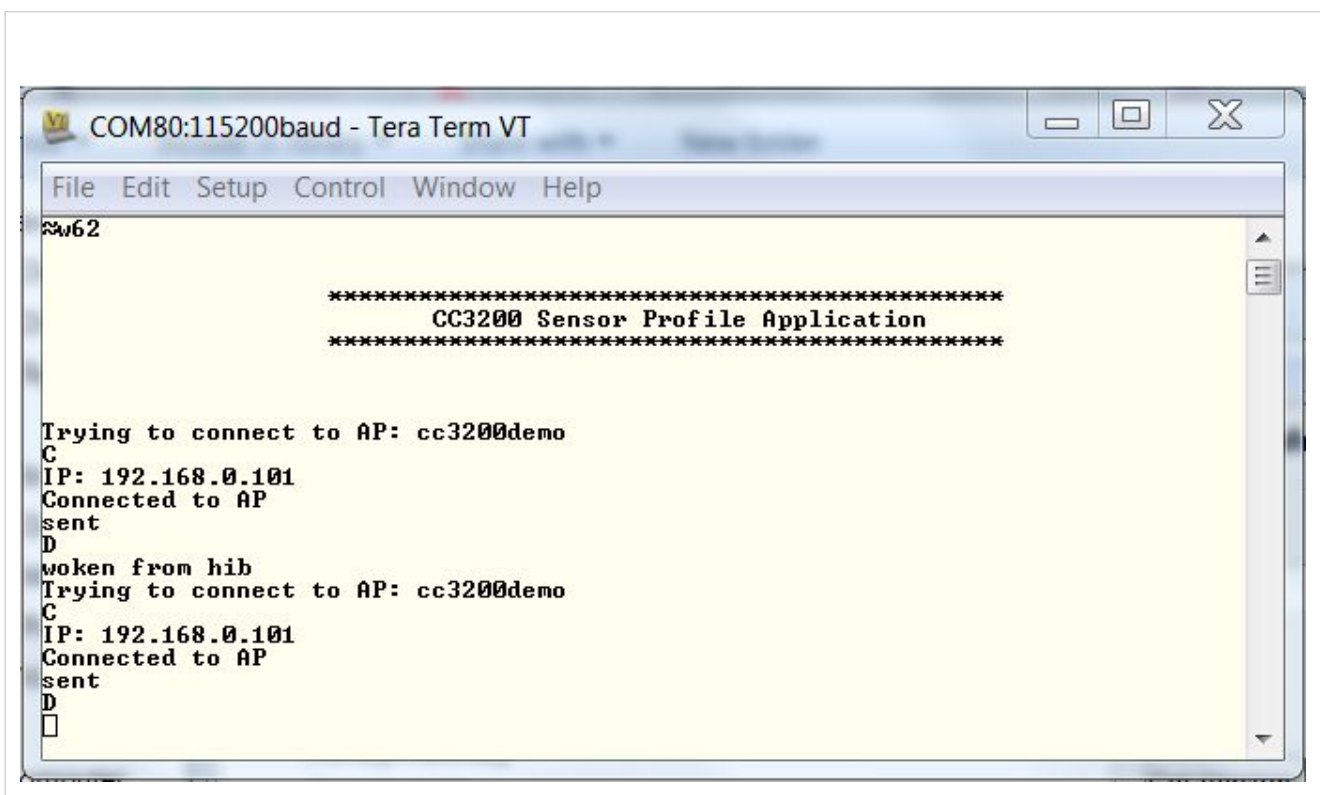
**Note:** One can simply run UDP Socket application before executing this application, that will take care of the above mentioned points.

## Usage

- Modify MACROS according to the requirement and recompile.
- Setup a serial communication application (HyperTerminal/TeraTerm). For detail info visit Terminal setup

On the host PC. The settings are:

- **Port:** Enumerated COM port
- **Baud rate:** 115200
- **Data:** 8 bit
- **Parity:** None
- **Stop:** 1 bit
- **Flow control:** None
- Run the application preferably from FLASH rather than from the debugger as debugger would disconnect in Hibernate.
- See the print on the serial communication program to understand the flow.



## **Limitations/Known Issues**

Refer to CC32xx\_Power\_Management\_Framework for limitations and known issues.

# Article Sources and Contributors

**CC32xx Sensor Profile Application** *Source:* <http://processors.wiki.ti.com/index.php?oldid=184788> *Contributors:* A0221015, Codycooke, Jitgupta, Malokyle

# Image Sources, Licenses and Contributors

**File:Cc31xx cc32xx return home.png** *Source:* [http://processors.wiki.ti.com/index.php?title=File:Cc31xx\\_cc32xx\\_return\\_home.png](http://processors.wiki.ti.com/index.php?title=File:Cc31xx_cc32xx_return_home.png) *License:* unknown *Contributors:* A0221015

**File:Cc32xx return sample apps.png** *Source:* [http://processors.wiki.ti.com/index.php?title=File:Cc32xx\\_return\\_sample\\_apps.png](http://processors.wiki.ti.com/index.php?title=File:Cc32xx_return_sample_apps.png) *License:* unknown *Contributors:* A0221015

**Image:Sensor\_profile\_screenshot.JPG** *Source:* [http://processors.wiki.ti.com/index.php?title=File:Sensor\\_profile\\_screenshot.JPG](http://processors.wiki.ti.com/index.php?title=File:Sensor_profile_screenshot.JPG) *License:* unknown *Contributors:* Jitgupta