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.

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        0xC0A80067
```

💡 **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(TimerGPIOTask,
            (const signed char *)"set wk_src for hibernate",
            OSI_STACK_SIZE, NULL, 1, NULL );
```

Task: TimerGPIOTask

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
//
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);
```

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

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:
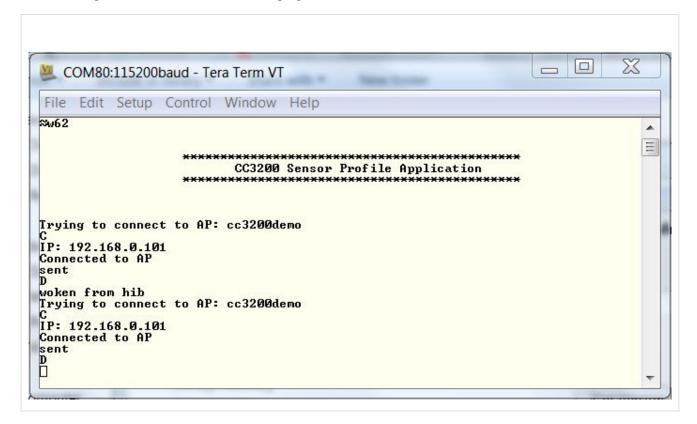
- **Port**: Enumerated COM port
- **Baud rate**: 115200
- **Data**: 8 bit
- **Parity**: None
- **Stop**: 1 bit
- **Flow control**: None

- 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

None

Links

Template:CC32xx Sample Apps Map

Article Sources and Contributors

Image Sources, Licenses and Contributors

License

a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Adaptations or Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

a. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

b. Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

c. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

d. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

e. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

f. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.