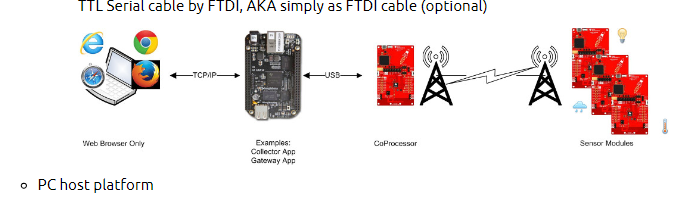
**Date Submitted: 12/13/19**

**Youtube Link:**

<https://www.youtube.com/watch?v=BwWgzvQXfO4>

**Modified Schematic (if applicable):**

**Code:**

**temperature.c (from sensor) :**

/\*

\* Copyright (c) 2018-2019, Texas Instruments Incorporated

\* All rights reserved.

\*

\* Redistribution and use in source and binary forms, with or without

\* modification, are permitted provided that the following conditions

\* are met:

\*

\* \* Redistributions of source code must retain the above copyright

\* notice, this list of conditions and the following disclaimer.

\*

\* \* Redistributions in binary form must reproduce the above copyright

\* notice, this list of conditions and the following disclaimer in the

\* documentation and/or other materials provided with the distribution.

\*

\* \* Neither the name of Texas Instruments Incorporated nor the names of

\* its contributors may be used to endorse or promote products derived

\* from this software without specific prior written permission.

\*

\* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"

\* AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,

\* THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR

\* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR

\* CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,

\* EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,

\* PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS;

\* OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,

\* WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR

\* OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE,

\* EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

\*/

/\*

\* ======== i2ctmp116.c ========

\*/

**#include** <stdint.h>

**#include** <stddef.h>

**#include** <unistd.h>

**#include** "smsgs.h"

**#include** "mac\_util.h"

**#include** "api\_mac.h"

**#include** "sensor.h"

**extern** Smsgs\_tempSensorField\_t tempSensor;

/\* Driver Header files \*/

**#include** <ti/drivers/GPIO.h>

**#include** <ti/drivers/I2C.h>

**#include** <ti/display/Display.h>

/\* Example/Board Header files \*/

**#include** "Board.h"

**#define** TASKSTACKSIZE 640

/\*

\* ======== TMP Registers ========

\*/

**#define** Si7021\_TMP\_REG 0xE3

**#define** Si7021\_HUM\_REG 0xE5

**#define** Si7021\_ADDR 0x40;

**static** Display\_Handle display;

/\*

\* ======== mainThread ========

\*/

**void** \***mainThread**(**void** \*arg0)

{

uint16\_t sample;

uint16\_t temperature,temperaturef;

uint8\_t txBuffer[1];

uint8\_t rxBuffer[2];

I2C\_Handle i2c;

I2C\_Params i2cParams;

I2C\_Transaction i2cTransaction;

/\* Call driver init functions \*/

Display\_init();

**GPIO\_init**();

**I2C\_init**();

/\* Configure the LED and if applicable, the TMP116\_EN pin \*/

**GPIO\_setConfig**(Board\_GPIO\_LED0, GPIO\_CFG\_OUT\_STD | GPIO\_CFG\_OUT\_LOW);

**#ifdef** Board\_GPIO\_TMP116\_EN

**GPIO\_setConfig**(Board\_GPIO\_TMP116\_EN, GPIO\_CFG\_OUT\_STD | GPIO\_CFG\_OUT\_HIGH);

/\* 1.5 ms reset time for the TMP116 \*/

**sleep**(1);

**#endif**

/\* Open the HOST display for output \*/

display = Display\_open(Display\_Type\_UART, NULL);

**if** (display == NULL) {

**while** (1);

}

/\* Turn on user LED \*/

**GPIO\_write**(Board\_GPIO\_LED0, Board\_GPIO\_LED\_ON);

Display\_printf(display, 0, 0, "Starting the i2ctmp example.");

/\* Create I2C for usage \*/

**I2C\_Params\_init**(&i2cParams);

i2cParams.bitRate = *I2C\_400kHz*;

i2c = **I2C\_open**(Board\_I2C\_TMP, &i2cParams);

**if** (i2c == NULL) {

Display\_printf(display, 0, 0, "Error Initializing I2C\n");

**while** (1);

}

**else** {

Display\_printf(display, 0, 0, "I2C Initialized!\n");

}

/\* Common I2C transaction setup \*/

i2cTransaction.writeBuf = txBuffer;

i2cTransaction.writeCount = 1;

i2cTransaction.readBuf = rxBuffer;

i2cTransaction.readCount = 2;

/\* Try Si7021 \*/

txBuffer[0] = Si7021\_TMP\_REG;

i2cTransaction.slaveAddress = Si7021\_ADDR;

**if** (!**I2C\_transfer**(i2c, &i2cTransaction)) {

/\* Could not resolve a sensor, error \*/

Display\_printf(display, 0, 0, "Error. No TMP sensor found!");

**while**(1);

}

**else** {

Display\_printf(display, 0, 0, "Detected Si7021 sensor.");

}

/\* Take 20 samples and print them out onto the console \*/

**for** (sample = 0; sample < 20; sample++) {

**if** (**I2C\_transfer**(i2c, &i2cTransaction)) {

/\*

\* Extract degrees C from the received data;

\* see Si7021 datasheet

\*/

temperature = (rxBuffer[0] << 8) | (rxBuffer[1]);

temperaturef = (((175.72 \* temperature)/ 65536) - 46.85);

Display\_printf(display, 0, 0, "Sample %u: %d (C)",

sample, temperaturef);

}

**else** {

Display\_printf(display, 0, 0, "I2C Bus fault.");

}

tempSensor.objectTemp = temperaturef;

tempSensor.ambienceTemp = temperaturef;

Util\_setEvent(&Sensor\_events, EXT\_SENSOR\_READING\_TIMEOUT\_EVT);

/\* Sleep for 1 second \*/

**sleep**(1);

}

**I2C\_close**(i2c);

Display\_printf(display, 0, 0, "I2C closed!");

**return** (NULL);

}

**------------------------------------------------------------------------------------**

**Config.h (from sensor) :**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

@file config.h

@brief TI-15.4 Stack configuration parameters for Sensor applications

Group: WCS LPC

Target Device: cc13x0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Copyright (c) 2016-2019, Texas Instruments Incorporated

All rights reserved.

Redistribution and use in source and binary forms, with or without

modification, are permitted provided that the following conditions

are met:

\* Redistributions of source code must retain the above copyright

notice, this list of conditions and the following disclaimer.

\* Redistributions in binary form must reproduce the above copyright

notice, this list of conditions and the following disclaimer in the

documentation and/or other materials provided with the distribution.

\* Neither the name of Texas Instruments Incorporated nor the names of

its contributors may be used to endorse or promote products derived

from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"

AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,

THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR

PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR

CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,

EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,

PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS;

OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,

WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR

OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE,

EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**#ifndef** CONFIG\_H

**#define** CONFIG\_H

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Includes

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**#include** "api\_mac.h"

**#ifdef** \_\_cplusplus

**extern** "C"

{

**#endif**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Constants and definitions

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* config parameters \*/

/\*! Security Enable - set to true to turn on security \*/

**#define** CONFIG\_SECURE **true**

/\*! PAN ID \*/

**#define** CONFIG\_PAN\_ID 0xFFFF

/\*! FH disabled as default \*/

**#define** CONFIG\_FH\_ENABLE **false**

/\*! link quality \*/

**#define** CONFIG\_LINKQUALITY 1

/\*! percent filter \*/

**#define** CONFIG\_PERCENTFILTER 0xFF

/\*!

Beacon order, value of 15 indicates non beacon mode,

8 is a good value for beacon mode

\*/

**#define** CONFIG\_MAC\_BEACON\_ORDER 15

/\*!

Superframe order, value of 15 indicates non beacon mode,

8 is a good value for beacon mode

\*/

**#define** CONFIG\_MAC\_SUPERFRAME\_ORDER 15

/\*! Maximum number of message failure, to indicate sync loss \*/

**#define** CONFIG\_MAX\_DATA\_FAILURES 3

/\*!

Maximum number of attempts for association in FH mode

after reception of a PAN Config frame

\*/

**#define** CONFIG\_FH\_MAX\_ASSOCIATION\_ATTEMPTS 3

/\* Interval for scan backoff \*/

**#define** CONFIG\_SCAN\_BACKOFF\_INTERVAL 5000

/\* Interval for delay between orphan notifications \*/

**#define** CONFIG\_ORPHAN\_BACKOFF\_INTERVAL 300000

/\*! Setting for Phy ID \*/

**#define** CONFIG\_PHY\_ID (APIMAC\_STD\_US\_915\_PHY\_1)

/\*! MAC Parameter \*/

/\*! Min BE - Minimum Backoff Exponent \*/

**#define** CONFIG\_MIN\_BE 3

/\*! Max BE - Maximum Backoff Exponent \*/

**#define** CONFIG\_MAX\_BE 5

/\*! MAC MAX CSMA Backoffs \*/

**#define** CONFIG\_MAC\_MAX\_CSMA\_BACKOFFS 4

/\*! macMaxFrameRetries - Maximum Frame Retries \*/

**#define** CONFIG\_MAX\_RETRIES 3

**#if** ((CONFIG\_PHY\_ID >= APIMAC\_MRFSK\_STD\_PHY\_ID\_BEGIN) && (CONFIG\_PHY\_ID <= APIMAC\_MRFSK\_STD\_PHY\_ID\_END))

/\*! Setting for channel page \*/

**#define** CONFIG\_CHANNEL\_PAGE (APIMAC\_CHANNEL\_PAGE\_9)

**#elif** ((CONFIG\_PHY\_ID >= APIMAC\_MRFSK\_GENERIC\_PHY\_ID\_BEGIN) && (CONFIG\_PHY\_ID <= APIMAC\_MRFSK\_GENERIC\_PHY\_ID\_END))

/\*! Setting for channel page \*/

**#define** CONFIG\_CHANNEL\_PAGE (APIMAC\_CHANNEL\_PAGE\_10)

**#else**

**#error** "PHY ID is wrong."

**#endif**

**#if** (defined(CC1312R1\_LAUNCHXL))

**#if**((CONFIG\_PHY\_ID == APIMAC\_GENERIC\_CHINA\_433\_PHY\_128) || (CONFIG\_PHY\_ID == APIMAC\_GENERIC\_CHINA\_LRM\_433\_PHY\_130))

**#error** "Error: 433 MHz Operation is not supported on 1312 board!"

**#endif**

**#endif**

/\*! scan duration in seconds\*/

**#define** CONFIG\_SCAN\_DURATION 5

/\*!

Coordinator Short Address When Operating with FH Enabled.

\*/

**#define** FH\_COORD\_SHORT\_ADDR 0xAABB

/\*!

Range Extender Mode setting.

The following modes are available.

APIMAC\_NO\_EXTENDER - does not have PA/LNA

APIMAC\_HIGH\_GAIN\_MODE - high gain mode

To enable CC1190, use

#define CONFIG\_RANGE\_EXT\_MODE APIMAC\_HIGH\_GAIN\_MODE

\*/

**#define** CONFIG\_RANGE\_EXT\_MODE APIMAC\_NO\_EXTENDER

/\*! Setting Default Key\*/

**#define** KEY\_TABLE\_DEFAULT\_KEY {0x12, 0x34, 0x56, 0x78, 0x9a, 0xbc, 0xde, 0xf0,\

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00}

/\*!

Channel mask used when CONFIG\_FH\_ENABLE is false.

Each bit indicates if the corresponding channel is to be scanned

First byte represents channels 0 to 7 and the last byte represents

channels 128 to 135.

For byte zero in the bit mask, LSB representing Ch0.

For byte 1, LSB represents Ch8 and so on.

e.g., 0x01 0x10 represents Ch0 and Ch12 are included.

The default of 0x0F represents channels 0-3 are selected.

APIMAC\_STD\_US\_915\_PHY\_1 (50kbps/2-FSK/915MHz band) has channels 0 - 128.

APIMAC\_STD\_ETSI\_863\_PHY\_3 (50kbps/2-FSK/863MHz band) has channels 0 - 33.

APIMAC\_GENERIC\_CHINA\_433\_PHY\_128 (50kbps/2-FSK/433MHz band) has channels 0 - 6.

\*/

**#define** CONFIG\_CHANNEL\_MASK { 0x04, 0x00, 0x00, 0x00, 0x00, 0x00, \

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, \

0x00, 0x00, 0x00, 0x00, 0x00 }

/\*!

Channel mask used when CONFIG\_FH\_ENABLE is true.

Represents the list of channels on which the device can hop.

When CONFIG\_RX\_ON\_IDLE is true, the actual sequence will

be based on DH1CF function. When it is set to false, the sequence

shall be a linear hopping over available channels in ascending order and

shall be used to change channel during the join phase.

It is represented as a bit string with LSB representing Ch0.

e.g., 0x01 0x10 represents Ch0 and Ch12 are included.

\*/

**#define** CONFIG\_FH\_CHANNEL\_MASK { 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, \

0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, \

0x00, 0x00, 0x00, 0x00, 0x00,}

/\* FH related config variables \*/

/\*!

List of channels to target the Async frames

It is represented as a bit string with LSB representing Ch0

e.g., 0x01 0x10 represents Ch0 and Ch12 are included

It should cover all channels that could be used by a target device in its

hopping sequence. Channels marked beyond number of channels supported by

PHY Config will be excluded by stack. To avoid interference on a channel,

it should be removed from Async Mask and added to exclude channels

(CONFIG\_CHANNEL\_MASK).

\*/

**#define** FH\_ASYNC\_CHANNEL\_MASK { 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, \

0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, \

0xFF, 0xFF, 0xFF, 0xFF, 0xFF }

/\*! Rx on when idle, false for sleepy device, true for non sleepy device \*/

**#define** CONFIG\_RX\_ON\_IDLE **false**

/\*!

The number of non sleepy channel hopping end devices to be supported.

It is to be noted that the total number of non sleepy devices supported

must be less than 50. Stack will allocate memory proportional

to the number of end devices requested.

\*/

**#define** FH\_NUM\_NON\_SLEEPY\_HOPPING\_NEIGHBORS 2

/\*!

The number of non sleepy fixed channel end devices to be supported.

It is to be noted that the total number of non sleepy devices supported

must be less than 50. Stack will allocate memory proportional

to the number of end devices requested.

\*/

**#define** FH\_NUM\_NON\_SLEEPY\_FIXED\_CHANNEL\_NEIGHBORS 2

/\*!

Dwell Time: The duration for which a non sleepy end device shall

stay on a specific channel before hopping to next channel.

\*/

**#define** CONFIG\_DWELL\_TIME 250

**#if** (((CONFIG\_PHY\_ID >= APIMAC\_MRFSK\_STD\_PHY\_ID\_BEGIN) && (CONFIG\_PHY\_ID <= APIMAC\_MRFSK\_GENERIC\_PHY\_ID\_BEGIN)) || \

((CONFIG\_PHY\_ID >= APIMAC\_GENERIC\_US\_915\_PHY\_132) && (CONFIG\_PHY\_ID <= APIMAC\_GENERIC\_ETSI\_863\_PHY\_133)))

/\*! Default Polling interval in milliseconds. It will get updated upon reception

of a config request message \*/

**#define** CONFIG\_POLLING\_INTERVAL 6000

/\*! PAN Advertisement Solicit trickle timer duration in milliseconds \*/

**#define** CONFIG\_PAN\_ADVERT\_SOLICIT\_CLK\_DURATION 6000

/\*! PAN Config Solicit trickle timer duration in milliseconds \*/

**#define** CONFIG\_PAN\_CONFIG\_SOLICIT\_CLK\_DURATION 6000

/\*! Default Reporting Interval - in milliseconds. It will get updated upon

reception of a config request message \*/

**#define** CONFIG\_REPORTING\_INTERVAL 500 //set to a more responsive rate

**#else**

/\*! Default Polling interval in milliseconds. It will get updated upon reception

of a config request message \*/

**#define** CONFIG\_POLLING\_INTERVAL 6000 //previously 60000

/\*! PAN Advertisement Solicit trickle timer duration in milliseconds \*/

**#define** CONFIG\_PAN\_ADVERT\_SOLICIT\_CLK\_DURATION 60000

/\*! PAN Config Solicit trickle timer duration in milliseconds \*/

**#define** CONFIG\_PAN\_CONFIG\_SOLICIT\_CLK\_DURATION 60000

/\*! Default Reporting Interval - in milliseconds. It will get updated upon

reception of a config request message \*/

**#define** CONFIG\_REPORTING\_INTERVAL 500 //previous: 600000

**#endif**

/\*! FH Poll/Sensor msg start time randomization window \*/

**#define** CONFIG\_FH\_START\_POLL\_DATA\_RAND\_WINDOW 10000

/\*! If enabled, the periodic sensor message shall be sent as a fixed size

\* packet of specified size. If set to 0, the periodic sensor message shall be

\* of type sensor data specified in smsgs.h

\*/

**#define** SENSOR\_TEST\_RAMP\_DATA\_SIZE 0

/\*! value for ApiMac\_FHAttribute\_netName \*/

**#define** CONFIG\_FH\_NETNAME {"FHTest"}

/\*! Range Extender is not supported in uBLE project \*/

**#ifdef** FEATURE\_UBLE

**#if** CONFIG\_RANGE\_EXT\_MODE

**#error** "CONFIG\_RANGE\_EXT\_MODE should be APIMAC\_NO\_EXTENDER"

**#endif**

**#endif**

/\*!

Value for Transmit Power in dBm

For US and ETSI band, Default value is 10, allowed values are

-10, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 and 14dBm.

For China band, allowed values are 6, 10, 13, 14 and 15dBm.

For CC1190, allowed values are between 18, 23, 25, 26 and 27dBm.

When the nodes in the network are close to each other

lowering this value will help reduce saturation \*/

**#ifndef** DeviceFamily\_CC13X2

**#if** CONFIG\_RANGE\_EXT\_MODE

**#define** CONFIG\_TRANSMIT\_POWER 26

**#else**

**#if** ((CONFIG\_PHY\_ID == APIMAC\_GENERIC\_CHINA\_433\_PHY\_128) || (CONFIG\_PHY\_ID == APIMAC\_GENERIC\_CHINA\_LRM\_433\_PHY\_130))

**#define** CONFIG\_TRANSMIT\_POWER 14

**#else**

**#define** CONFIG\_TRANSMIT\_POWER 12

**#endif**

**#endif**

**#else** /\* DeviceFamily\_CC13X2 \*/

**#define** CONFIG\_TRANSMIT\_POWER 12

**#endif**

**#ifndef** DeviceFamily\_CC13X2

**#if** CONFIG\_RANGE\_EXT\_MODE

**#if** (CCFG\_FORCE\_VDDR\_HH == 1)

**#error** "CCFG\_FORCE\_VDDR\_HH should be 0"

**#endif**

**#else**

**#if** ((CONFIG\_PHY\_ID == APIMAC\_GENERIC\_CHINA\_433\_PHY\_128) || (CONFIG\_PHY\_ID == APIMAC\_GENERIC\_CHINA\_LRM\_433\_PHY\_130))

**#if** (CCFG\_FORCE\_VDDR\_HH == 0)

**#if** (CONFIG\_TRANSMIT\_POWER >= 15)

**#error** "CONFIG\_TRANSMIT\_POWER should be less than 15"

**#endif**

**#else**

**#if** (CONFIG\_TRANSMIT\_POWER < 15)

/\* In 433 MHz band when CCFG\_FORCE\_VDDR\_HH = 1, only possible value of transmit power is 15 \*/

**#error** "CONFIG\_TRANSMIT\_POWER should be 15"

**#endif**

**#endif**

**#else**

**#if** (CCFG\_FORCE\_VDDR\_HH == 0)

**#if** (CONFIG\_TRANSMIT\_POWER >= 14)

**#error** "CONFIG\_TRANSMIT\_POWER should be less than 14"

**#endif**

**#else**

**#if** (CONFIG\_TRANSMIT\_POWER < 14)

/\* In US and ETSI band when CCFG\_FORCE\_VDDR\_HH = 1, only possible value of transmit power is 14 \*/

**#error** "CONFIG\_TRANSMIT\_POWER should be 14"

**#endif**

**#endif**

**#endif**

**#endif**

**#else**

**#if** (CCFG\_FORCE\_VDDR\_HH == 1)

**#if** (CONFIG\_TRANSMIT\_POWER != 14)

/\* In US and ETSI band when CCFG\_FORCE\_VDDR\_HH = 1, only possible value of transmit power is 14 \*/

**#error** "CONFIG\_TRANSMIT\_POWER should be 14"

**#endif**

**#endif**

**#endif**

/\*!

\* Enable this mode for certfication.

\* For FH certification, CONFIG\_FH\_ENABLE should

\* also be enabled

\*/

**#define** CERTIFICATION\_TEST\_MODE **false**

**#ifdef** POWER\_MEAS

/\*!

Power profile to be used when Power MEAS is enabled.

Profile 1 - POLL\_ACK - Polling Only

Profile 2 - DATA\_ACK - 20 byte application data + ACK from sensor to collector

Profile 3 - POLL\_DATA - Poll + received Data from collector

Profile 4 - SLEEP - No Poll or Data. In Beacon mode, beacon RX would occur

\*/

**#define** POWER\_TEST\_PROFILE DATA\_ACK

**#endif**

/\* Check if all the necessary parameters have been set for FH mode \*/

**#if** CONFIG\_FH\_ENABLE

**#if** !defined(FEATURE\_ALL\_MODES) && !defined(FEATURE\_FREQ\_HOP\_MODE)

**#error** "Do you want to build image with frequency hopping mode? \

Define either FEATURE\_FREQ\_HOP\_MODE or FEATURE\_ALL\_MODES in features.h"

**#endif**

**#endif**

/\* Check if stack level security is enabled if application security is enabled \*/

**#if** CONFIG\_SECURE

**#if** !defined(FEATURE\_MAC\_SECURITY)

**#error** "Define FEATURE\_MAC\_SECURITY or FEATURE\_ALL\_MODES in features.h to \

be able to use security at application level"

**#endif**

**#endif**

/\* Set beacon order and superframe order to 15 for FH mode to avoid user error \*/

**#if** CONFIG\_FH\_ENABLE

**#if** (CONFIG\_MAC\_BEACON\_ORDER != 15) && (CONFIG\_MAC\_SUPERFRAME\_ORDER != 15)

**#error** "Do you want to build image with frequency hopping mode? \

If yes, CONFIG\_MAC\_BEACON\_ORDER and CONFIG\_MAC\_SUPERFRAME\_ORDER \

should both be set to 15"

**#endif**

**#if** (FH\_NUM\_NON\_SLEEPY\_HOPPING\_NEIGHBORS < 2) || (FH\_NUM\_NON\_SLEEPY\_FIXED\_CHANNEL\_NEIGHBORS < 2)

**#error** "You have an invalid value for FH neighbors. Set the values \

for FH\_NUM\_NON\_SLEEPY\_HOPPING\_NEIGHBORS and FH\_NUM\_NON\_SLEEPY\_FIXED\_CHANNEL\_NEIGHBORS to at least 2"

**#endif**

**#endif**

**#ifdef** \_\_cplusplus

}

**#endif**

**#endif** /\* CONFIG\_H \*/