Jett Guerrero
Github root directory: https://github.com/guerrj1/Advanced-Embedded-Systems

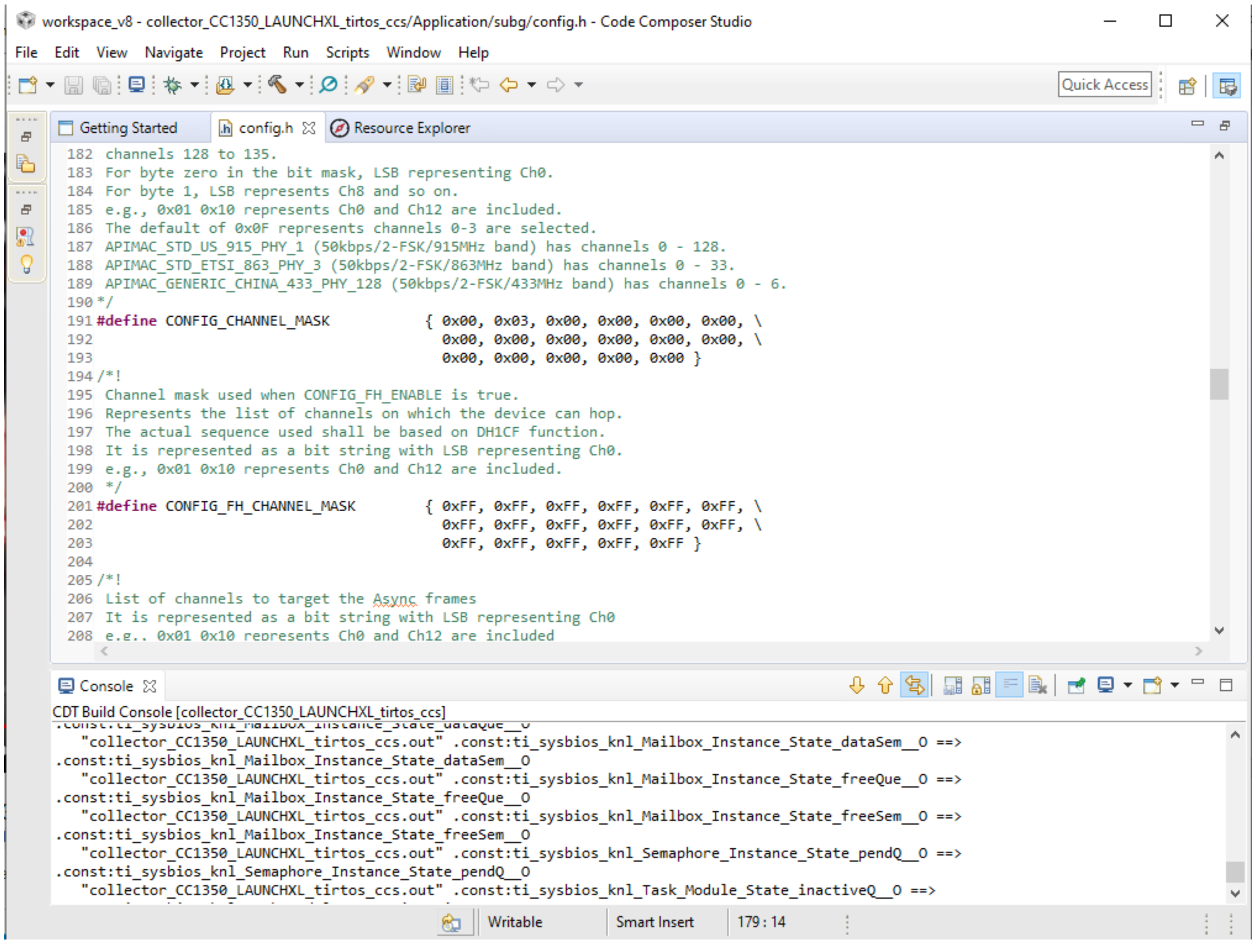
Date Submitted: **12/12/2019**

PART 1

Task 01:

Youtube Link: No Demo

Modified Code:


workspace_v8 - collector_CC1350_LAUNCHXL_tirtos_ccs/Application/subg/config.h - Code Composer Studio
File Edit View Navigate Project Run Scripts Window Help
Quick Access
Getting Started
config.h
Resource Explorer
Console
CDT Build Console [collector_CC1350_LAUNCHXL_tirtos_ccs]
Writable
Smart Insert
179 : 14

```
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           { 0x00, 0x03, 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.
The actual sequence used shall be based on DH1CF function.
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, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, \
                                        0xFF, 0xFF, 0xFF, 0xFF, 0xFF }

/*!
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
```

```
   "collector_CC1350_LAUNCHXL_tirtos_ccs.out" .const:ti_sysbios_knl_Mailbox_Instance_State_dataSem__O ==>
.const:ti_sysbios_knl_Mailbox_Instance_State_dataSem__O
   "collector_CC1350_LAUNCHXL_tirtos_ccs.out" .const:ti_sysbios_knl_Mailbox_Instance_State_freeQue__O ==>
.const:ti_sysbios_knl_Mailbox_Instance_State_freeQue__O
   "collector_CC1350_LAUNCHXL_tirtos_ccs.out" .const:ti_sysbios_knl_Mailbox_Instance_State_freeSem__O ==>
.const:ti_sysbios_knl_Mailbox_Instance_State_freeSem__O
   "collector_CC1350_LAUNCHXL_tirtos_ccs.out" .const:ti_sysbios_knl_Semaphore_Instance_State_pendQ__O ==>
.const:ti_sysbios_knl_Semaphore_Instance_State_pendQ__O
   "collector_CC1350_LAUNCHXL_tirtos_ccs.out" .const:ti_sysbios_knl_Task_Module_State_inactiveQ__O ==>
```

Partner's CCS collector project of the channel mask set to 0x03

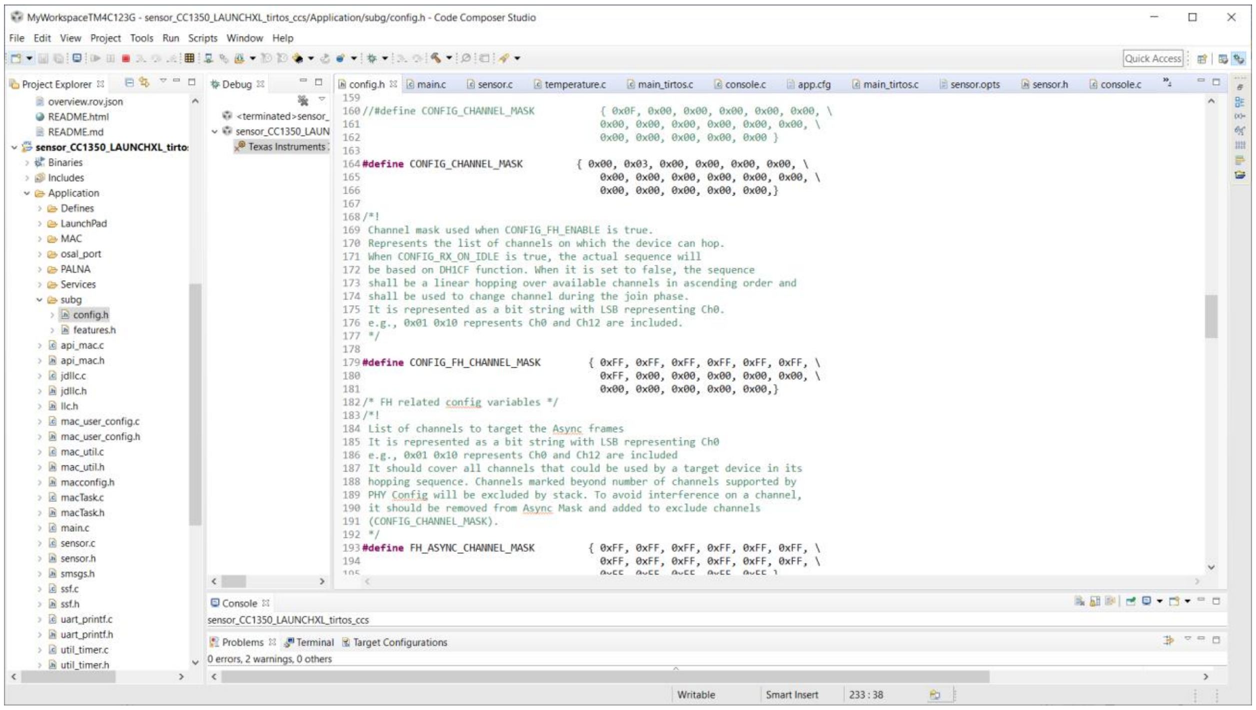
Jett Guerrero
Github root directory: https://github.com/guerrj1/Advanced-Embedded-Systems

----

Task 02:

Youtube Link: No Demo

Modified Code:



```
//#define CONFIG_CHANNEL_MASK           { 0x0F, 0x00, 0x00, 0x00, 0x00, 0x00, \
                                          0x00, 0x00, 0x00, 0x00, 0x00, 0x00, \
                                          0x00, 0x00, 0x00, 0x00, 0x00 }

#define CONFIG_CHANNEL_MASK         { 0x00, 0x03, 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, \
```

```
0 errors, 2 warnings, 0 others
```

CCS of Sensor Code with channel mask configured to 0x03

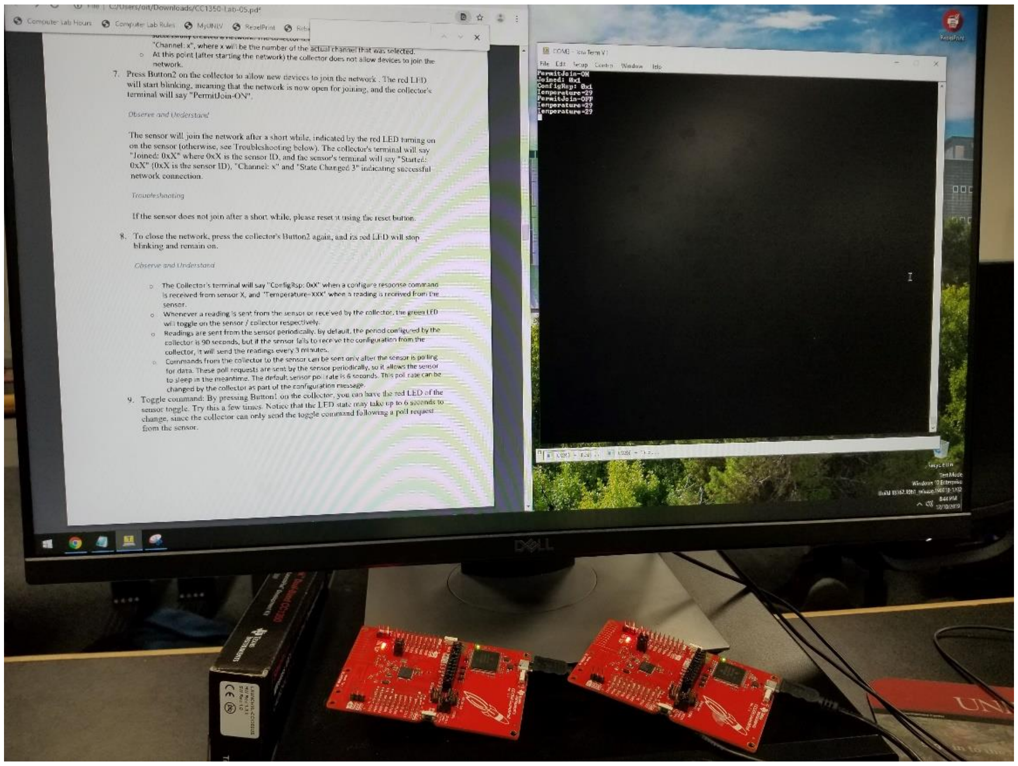
Grading scheme: 30% Coding, 30% Documentation, 40% Execution/Video.

Jett Guerrero
Github root directory: https://github.com/guerrj1/Advanced-Embedded-Systems

----

Task 03:

Youtube Link: https://youtu.be/es2oBuANhC0



```
PermitJoin-ON
Joined: 0x1
ConfigRsp: 0x1
Temperature=29
PermitJoin-OFF
Temperature=29
Temperature=29
```

Set up of the sensor board and the collector board with terminal window

Modified Code:

```
/******************************************************************************

 @file config.h

 @brief TI-15.4 Stack configuration parameters for Sensor applications

 Group: WCS LPC
 Target Device: cc13x0
 ******************************************************************************


 *****************************************************************************/
#ifndef CONFIG_H
#define CONFIG_H

/******************************************************************************
 Includes
 *****************************************************************************/
#include "api_mac.h"

#ifdef __cplusplus
extern "C"
```

Grading scheme: 30% Coding, 30% Documentation, 40% Execution/Video.

```
{
#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           { 0x0F, 0x00, 0x00, 0x00, 0x00, 0x00, \
                                        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, \
                                        0x00, 0x00, 0x00, 0x00, 0x00 }

#define CONFIG_CHANNEL_MASK         { 0x00, 0x03, 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  180000
#define CONFIG_REPORTING_INTERVAL  45000
//#define CONFIG_REPORTING_INTERVAL  500

#else
/*! Default Polling interval in milliseconds. It will get updated upon reception
 of a config request message */
#define CONFIG_POLLING_INTERVAL      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  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))
```

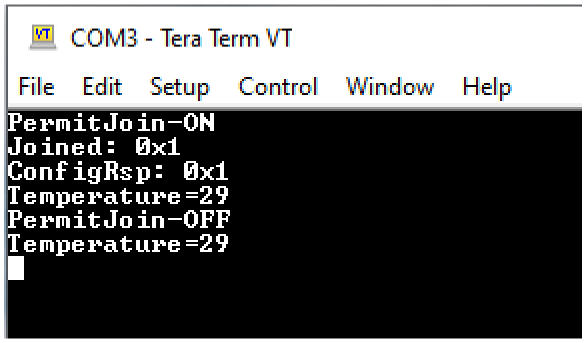
```
#define CONFIG_TRANSMIT_POWER        14
#else
#define CONFIG_TRANSMIT_POWER        12
#endif
#endif
#else /* DeviceFamily_CC13X2 */
#define CONFIG_TRANSMIT_POWER        12
#endif

#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 */
```

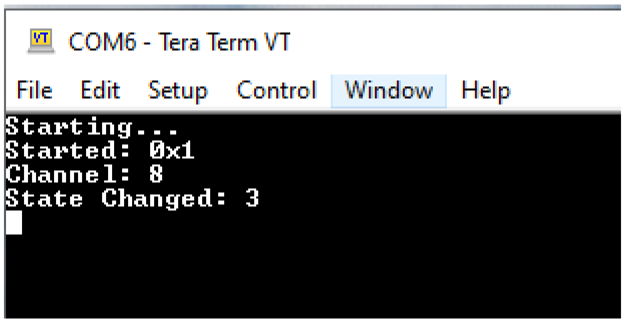
Task 04:

Youtube Link: https://youtu.be/TtSnbjiMraA

Grading scheme: 30% Coding, 30% Documentation, 40% Execution/Video.



```
PermitJoin-ON
Joined: 0x1
ConfigRsp: 0x1
Temperature=29
PermitJoin-OFF
Temperature=29
```



```
Starting...
Started: 0x1
Channel: 8
State Changed: 3
```

Modified Code:

```
/* Polling Interval Min and Max (in milliseconds) */
//#define MIN_POLLING_INTERVAL 1000
#define MIN_POLLING_INTERVAL 250
#define MAX_POLLING_INTERVAL 10000
```

sensor.c

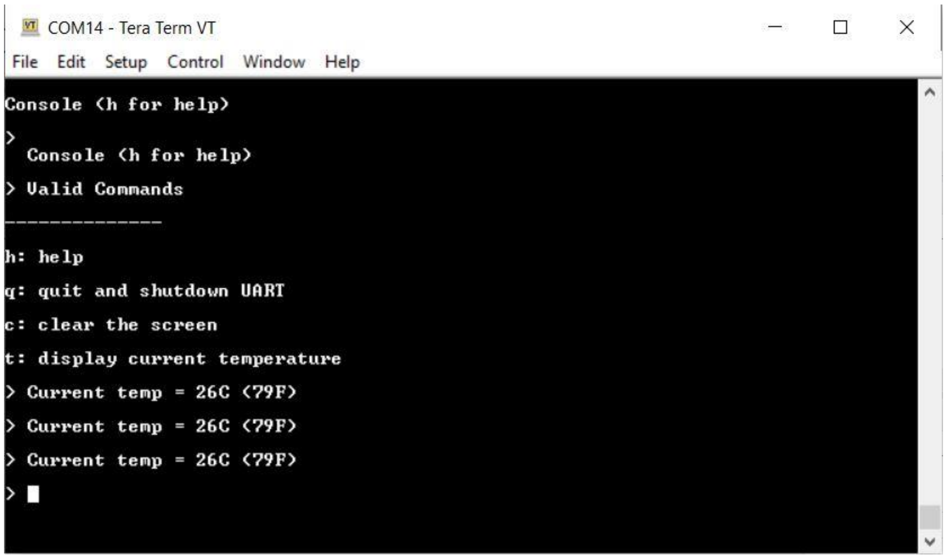
```
/*! Default Reporting Interval - in milliseconds. It will get updated upon
 reception of a config request message */
//#define CONFIG_REPORTING_INTERVAL  180000
#define CONFIG_REPORTING_INTERVAL  45000
//#define CONFIG_REPORTING_INTERVAL  500
```

config.h

PART 2

Task 01:

Youtube Link: https://youtu.be/W19BwKl0_Aw



```
Console (h for help)

>
 Console (h for help)
> Valid Commands
--------------
h: help
q: quit and shutdown UART
c: clear the screen
t: display current temperature
> Current temp = 26C (79F)
> Current temp = 26C (79F)
> Current temp = 26C (79F)
>
```

Jett Guerrero
Github root directory: https://github.com/guerrj1/Advanced-Embedded-Systems

Modified Code:

```
/* 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 < 100; sample++)
    {
        if (I2C_transfer(i2c, &i2cTransaction))
        {
            /*
             Extract degrees C from the received data;
             see Si7021 datasheet
            */
            temp = (rxBuffer[0] << 8) | (rxBuffer[1]);
            temperature = (((175.72 * temp)/ 65536) - 46.85); // celsius
            temperaturef = (temperature * (1.8)) + 32; //farenheit
            Display_printf(display, 0, 0, "Sample %u: %d (C)", sample, temperaturef);
        }

        else
        {
            Display_printf(display, 0, 0, "I2C Bus fault.");
        }
```

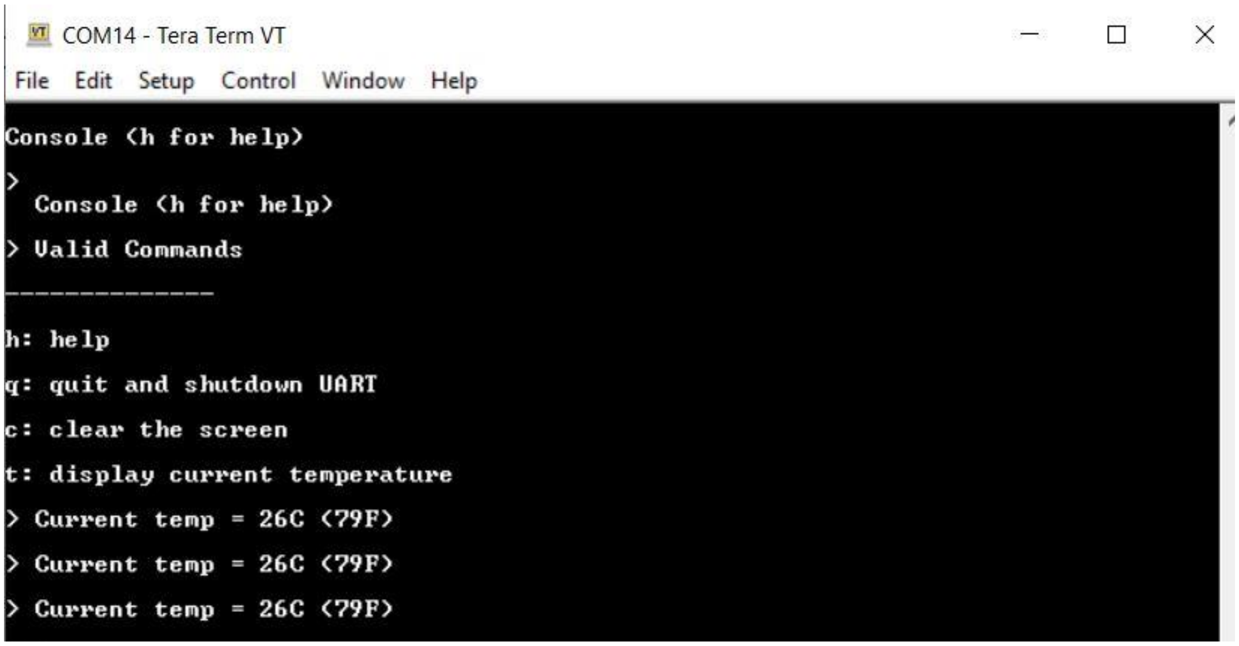
Grading scheme: 30% Coding, 30% Documentation, 40% Execution/Video.

Jett Guerrero
Github root directory: https://github.com/guerrj1/Advanced-Embedded-Systems

---

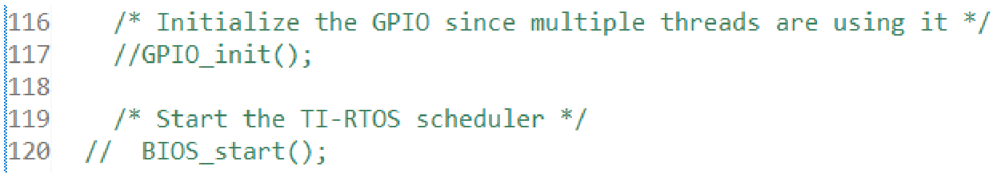
Task 02:

Youtube Link: https://youtu.be/QjsRt_1VV6Y



```
Console (h for help)
>
  Console (h for help)
> Valid Commands
--------------
h: help
q: quit and shutdown UART
c: clear the screen
t: display current temperature
> Current temp = 26C (79F)
> Current temp = 26C (79F)
> Current temp = 26C (79F)
```

Modified Code:



```
    /* Initialize the GPIO since multiple threads are using it */
    //GPIO_init();

    /* Start the TI-RTOS scheduler */
//  BIOS_start();
```
Commented these lines in main_tirtos.c

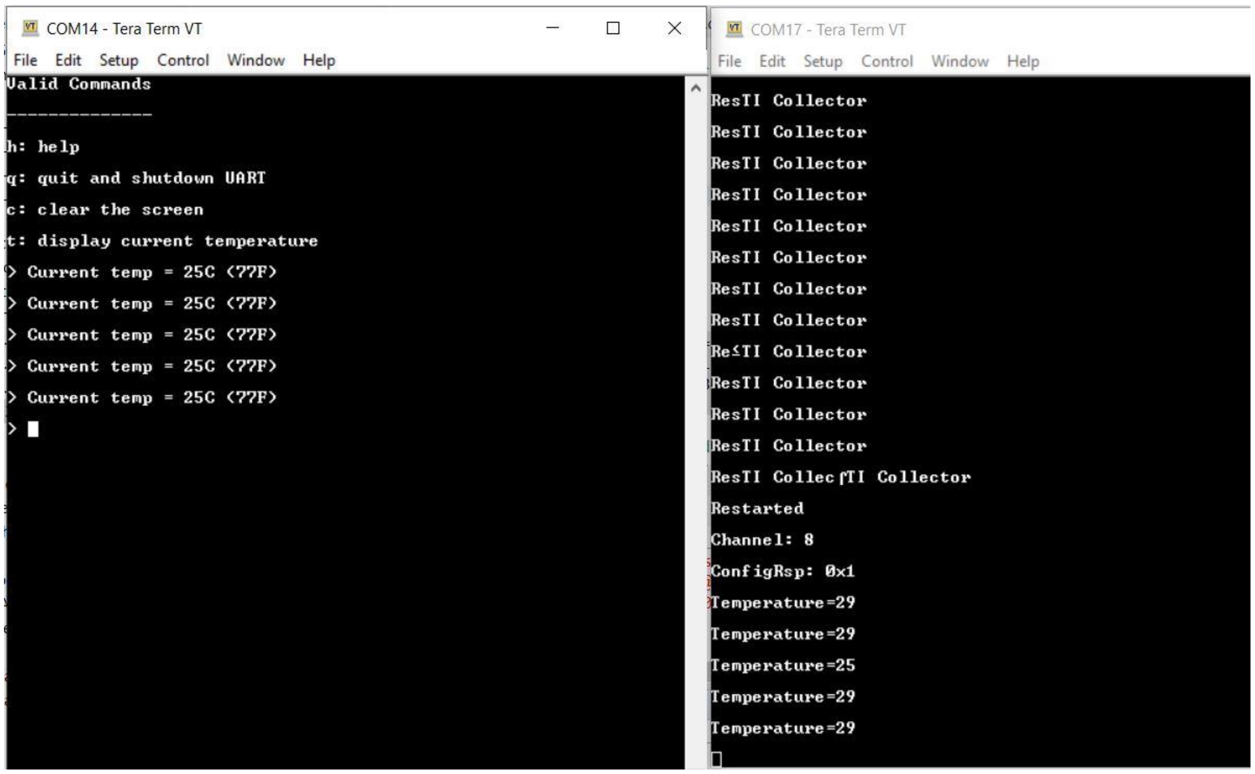
Grading scheme: 30% Coding, 30% Documentation, 40% Execution/Video.

Jett Guerrero
Github root directory: https://github.com/guerrj1/Advanced-Embedded-Systems

---

Task 03:

Youtube Link: https://youtu.be/k_kQ4Ab0tTA


COM14 - Tera Term VT
File Edit Setup Control Window Help
COM17 - Tera Term VT
File Edit Setup Control Window Help

```
Valid Commands
--------------
h: help
q: quit and shutdown UART
c: clear the screen
t: display current temperature
> Current temp = 25C (77F)
> Current temp = 25C (77F)
> Current temp = 25C (77F)
> Current temp = 25C (77F)
> Current temp = 25C (77F)
>
```

```
ResTI Collector
ResTI Collector
ResTI Collector
ResTI Collector
ResTI Collector
ResTI Collector
ResTI Collector
ResTI Collector
Re≤TI Collector
ResTI Collector
ResTI Collector
ResTI Collector
ResTI CollecƒTI Collector
Restarted
Channel: 8
ConfigRsp: 0x1
Temperature=29
Temperature=29
Temperature=25
Temperature=29
Temperature=29
```

Modified Code:

```
//Added libraries
#include "smsgs.h"
#include "mac_util.h"
#include "api_mac.h"
#include "sensor.h"
extern Smsgs_tempSensorField_t tempSensor;
```

```
switch (cmd) {
    case 't':

        //added lines
        tempSensor.objectTemp = localTemperatureC;
        tempSensor.ambienceTemp = localTemperatureC;
        Util_setEvent(&Sensor_events, EXT_SENSOR_READING_TIMEOUT_EVT);
```

Added lines in console.c

Grading scheme: 30% Coding, 30% Documentation, 40% Execution/Video.