Date Submitted: **December 10, 2019**

A CC1352R1 will be set up as a collector module and a CC1350 will be setup as a sensor module.

Task 01 :

Youtube Link: Task 3's video showing both collector and sensor running
https://youtu.be/Fz9JTOFhRGI

Modified Code:
Code is not modified as the function shown below already displays a temperature value received from the sensor module.

```
void Csf_deviceSensorDataUpdate(ApiMac_sAddr_t *pSrcAddr, int8_t rssi,
                                Smsgs_sensorMsg_t *pMsg)
{
    CUI_ledToggle(csfCuiHndl, CONFIG_LED_GREEN);
#ifndef POWER_MEAS
    CUI_statusLinePrintf(csfCuiHndl, deviceStatusLine, "Sensor - Addr=0x%04x,
Temp=%d, RSSI=%d",
                        pSrcAddr->addr.shortAddr, pMsg->tempSensor.ambienceTemp,
rssi);
    CUI_statusLinePrintf(csfCuiHndl, numJoinDevStatusLine, "%x",
getNumActiveDevices());

#endif /* endif for POWER_MEAS */

#if defined(MT_CSF)
    MTCSF_sensorUpdateIndCB(pSrcAddr, rssi, pMsg);
#endif /* endif for MT_CSF */
}
```
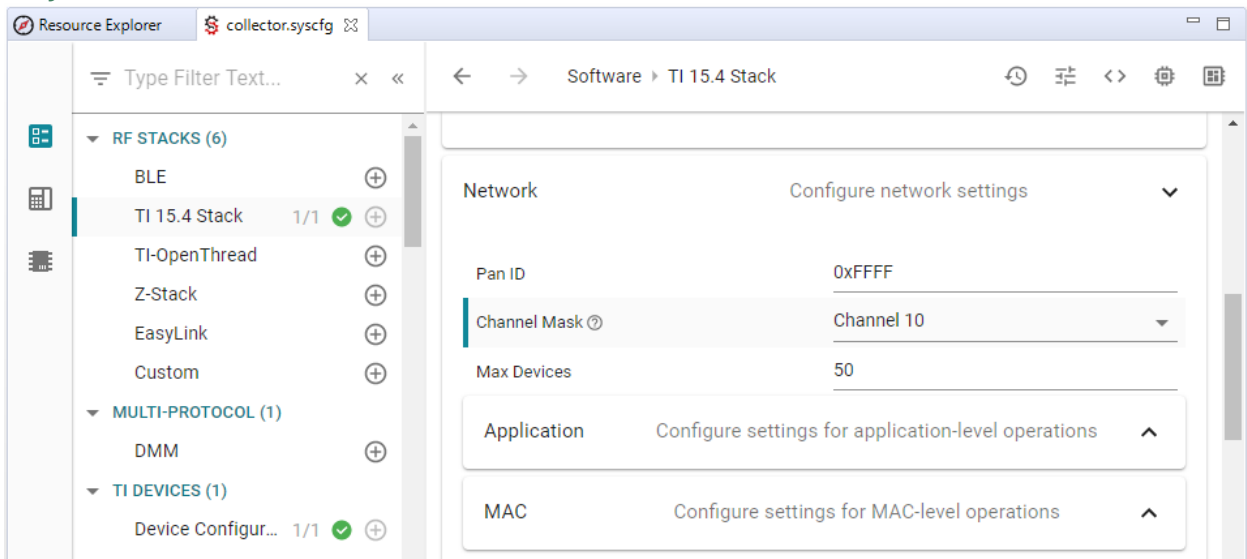
Screenshots

Collector is set for 50kbps and US-compatible PHY

Grading scheme: 30% Coding, 30% Documentation, 40% Execution/Video.
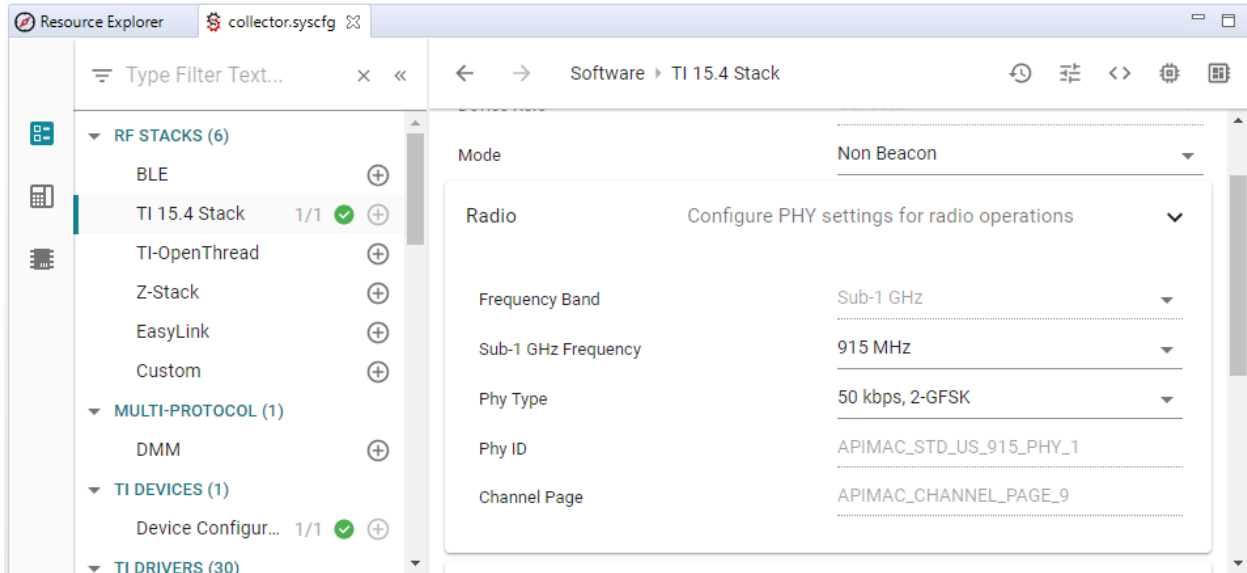
Only Channel 10 is enabled.

Task 02 :

Youtube Link: Task 3's video showing both collector and sensor running
https://youtu.be/Fz9JTOFhRGI

Modified Code:
Channel mask set to channel 10 using
```
#define CONFIG_CHANNEL_MASK             { 0x00, 0x04, 0x00, 0x00, 0x00, 0x00, \
```

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

```
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, \
0x00, 0x00, 0x00, 0x00, 0x00 }
```

Screenshots:

Task 03 :

Youtube Link: https://youtu.be/Fz9JTOFhRGI

Modified Code:

Screenshots:
UART Screen capture
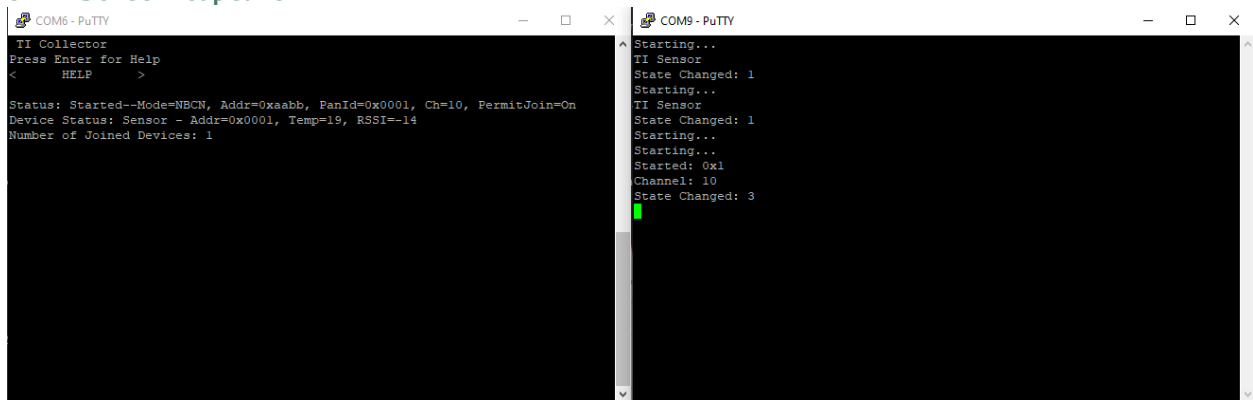
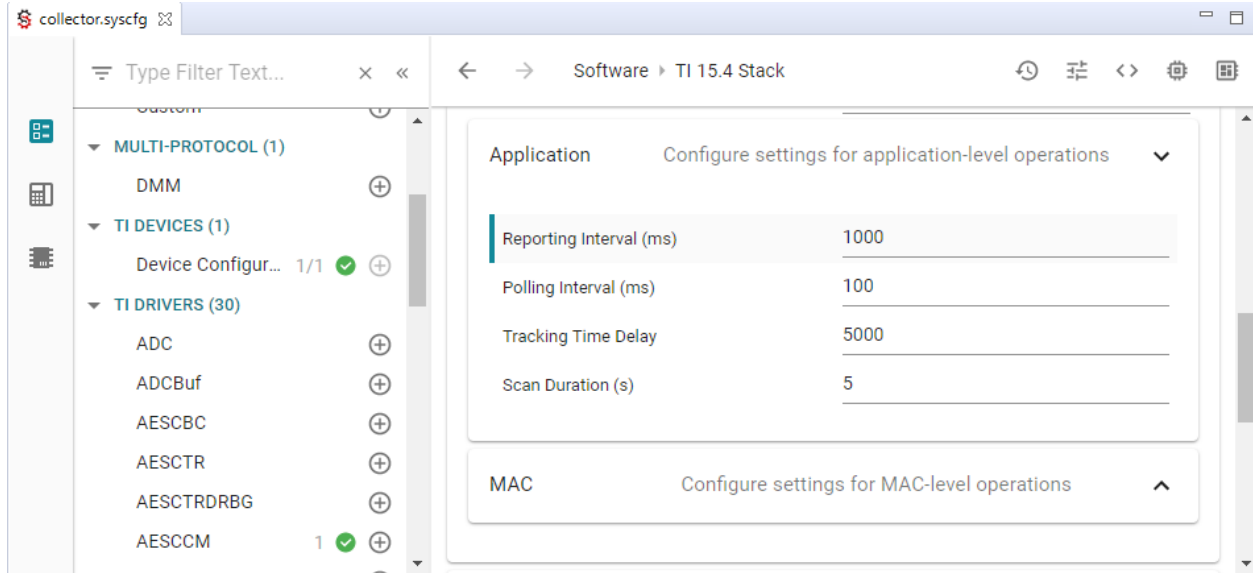
Task 04:

Youtube Link: https://youtu.be/b_lpkOnEAV4
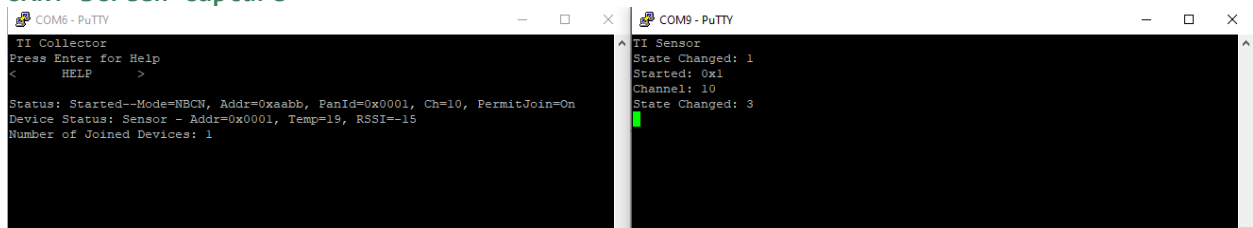
Sensor Modification:
#define CONFIG_REPORTING_INTERVAL 500

Collector Modification:
Changed so that collector reports every second and polls ever 100ms.

Screenshots:
UART Screen capture

Task 05 (Remote Sensor 1):

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

Temperature.c Modifications:

Added global definitions
```
/* ======== Si7021 Registers ======== */
#define Si7021_TMP_REG 0xE3
#define Si7021_HUM_REG 0xE5
#define Si7021_ADDR 0x40;
```

Searches for Si7021 sensore
```
    /* 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.");
```

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

```
        }

Changed how temperature is gathered from Si7021
while (1) {
        if (I2C_transfer(i2c, &i2cTransaction)) {
            /*
             *  Extract degrees C from the received data; see sensor datasheet.
             *  Make sure we are updating the global temperature variables
             *  in a thread-safe manner.
             */
            pthread_mutex_lock(&temperatureMutex);
            /*
             * Extract degrees C from the received data;
             * see Si7021 datasheet
             */
            temperatureC = (rxBuffer[0] << 8) | (rxBuffer[1]);
            temperatureC = (((175.72 * temperatureC)/ 65536) - 46.85);
            temperatureF = (temperatureC*5)/9+32;
            pthread_mutex_unlock(&temperatureMutex);

            /*  Send an alert if the temperature is too high!! */
            if ((int)temperatureC >= HIGH_TEMP) {
                sendAlert(temperatureC);
            }
            else {
                clearAlert(temperatureC);
            }
        }
```
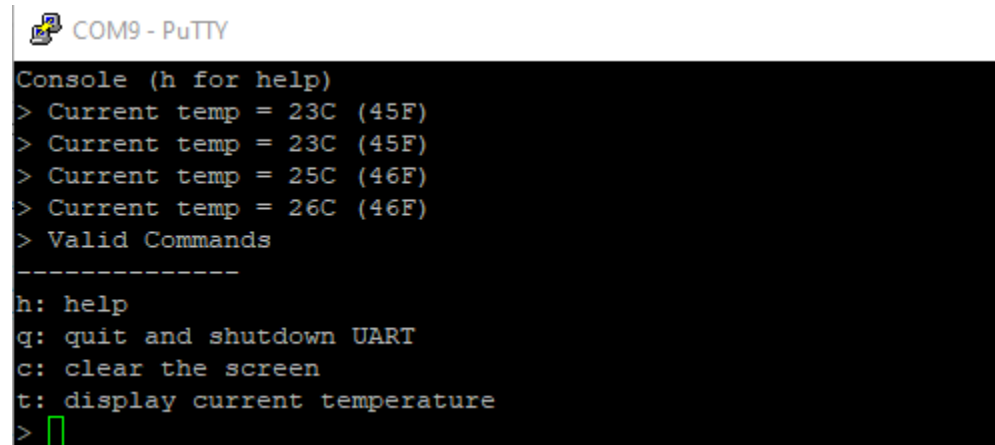
Screenshots:

UART


```
COM9 - PuTTY
Console (h for help)
> Current temp = 23C (45F)
> Current temp = 23C (45F)
> Current temp = 25C (46F)
> Current temp = 26C (46F)
> Valid Commands
--------------
h: help
q: quit and shutdown UART
c: clear the screen
t: display current temperature
>
```

<mark>Task 06:</mark>

Youtube Link: https://youtu.be/lJHNSgNNXW8

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

Modified Code:

Added following code to bottom of app.cfg

```
/* Include POSIX Support */
var Settings = xdc.useModule('ti.sysbios.posix.Settings');
```

In main_tirtos.c

Renamed main() to main_app() and commented out calls to Board_init(), GPIO_init() and BIOS_start().
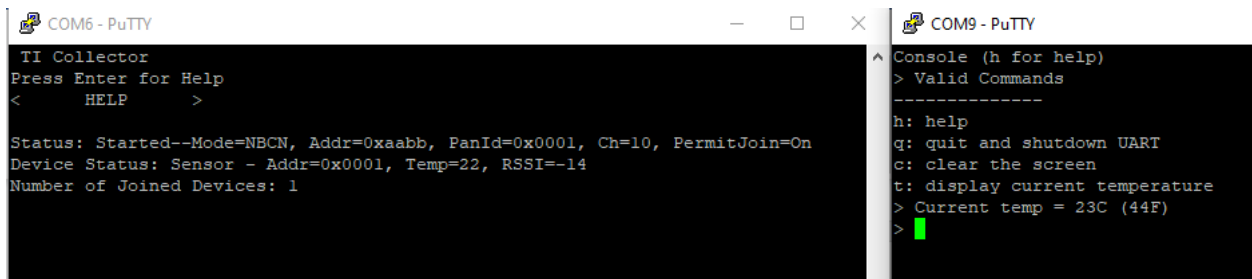
In main.c

Added main_app(); towards the end and added the following line before the start of main()

Extern int main_app(void);

In sensor.opts, removed DBOAD_DISPLAY_USE_UART and DxBOARD_DISPLAY_USE_LCD

Screenshots:

UART Output

Task 07:

Youtube Link: https://youtu.be/HG6ytxO1sbE

Modified Code:

In sensor.h, added #define EXT_SENSOR_READING_TIMEOUT_EVT 0x0004 to global definitions

In sensor.c, added the following coded inside the Sensor_process() function

```
if(Sensor_events & EXT_SENSOR_READING_TIMEOUT_EVT)
{
    /* Process Sensor Reading Message Event */
    processSensorMsgEvt();

    /* Clear the event */
    Util_clearEvent(&Sensor_events, EXT_SENSOR_READING_TIMEOUT_EVT);
}
```

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

Also removed 'STATIC' from the line STATIC Smsgs_tempSensorField_t tempSensor =

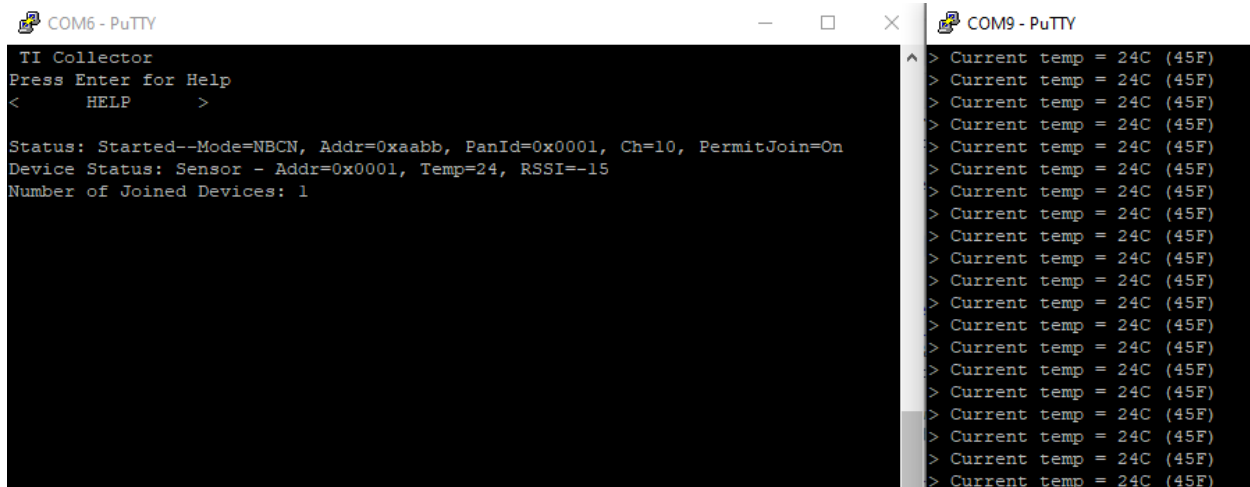In console.c, added following lines under global context

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

Also added the following lines in simpleConsole(), after the line "case t"

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

Screenshots:

UART, t needs to be held down has the collector update the temperature frequently

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