

Project Assignment 3. Turning a TI CC13X0 sensor node into an actuator

General Instructions:

- Submission Deadline: 12:00 noon on May 4th, 2018. Please submit electronically to (rja150230@utdallas.edu) and consider that you may be required for a demonstration to be produced in person on the 4th, from 5:00pm to 7:45pm.
- Team members: A maximum of two students per project.
- Collection of Equipment: Stop by TA office ECSN 3.520 Wednesdays and Fridays from 2:30pm to 7:00pm.

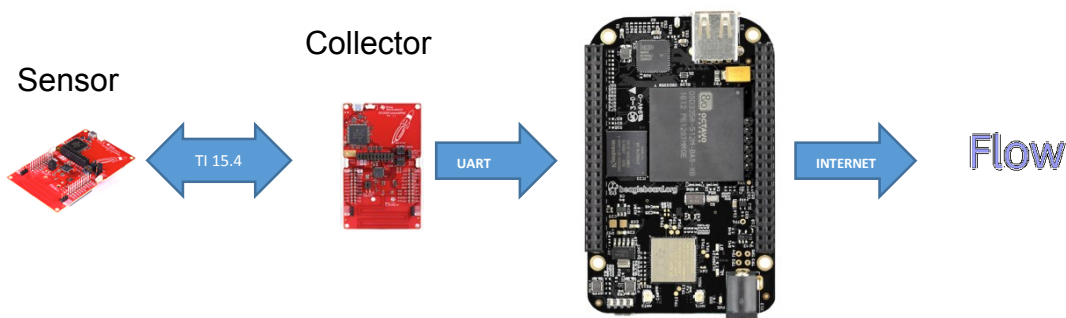
Problem Statement

Wireless sensor networks (WSN) are typically employed to collect data using well defined wireless protocols and analog/digital capabilities implemented in low power embedded systems such as the TI CC13X0 LaunchPads. However, user applications may require to take quick actions based on the data collected to react opportunely to the process being monitored.

In this project, you do need to set up a WSN using the CC13X0 LaunchPads running the TI Stack 15.4 implementation which will be connected to a main CPU using a BeagleBone Black Wireless board (BBW). You will be asked to measure the battery voltage and temperature of the sensor and report these values to the BBW. Then the BBW will send the data to the Flow Server where you will have to compare the measured value (Temperature) with a threshold value stored at the Flow. If the measured value is less than the threshold, you will be asked to turn off the onboard red led of the CC13X0 sensor node, otherwise the onboard red led should be turned on. By doing this assignment, you are essentially turning the sensor into an actuator that can react to its environment in a timely manner. Pick the threshold value at will to play with this condition.

Components

- Two (2) CC13X0 LaunchPads running the TI 15.4 Stack
- One (1) BeagleBone Black Wireless with Linux installed
- Internet connection over the WiFi built-in adapter of the Beaglebone Black
- AT&T Flow Design user account (it is free of charge at <http://flow.att.com>)



Procedure

For this project you will re-use your previous implementations in projects 1 and 2. The only two pieces missing are the following:

1. At the Flow server, sending a message back to the BBW. This can be done by placing a TCP/UDP out node to reply back to the socket already implemented when the threshold condition is met. At the BBW, every time after are sending data to the Flow server, you will need to listen the socket for a possible message back.
2. If you are receiving messages from the Flow server at the BBW, then you will need to send this message to the collector over the serial port.
3. Once the message is at the collector, you may use the following implementation to toggle the sensor red led in collector.c file.

```
/*!
Build and send the toggle led message to a device.

Public function defined in collector.h
*/
Collector_status_t Collector_sendToggleLedRequest(ApiMac_sAddr_t *pDstAddr)
{
    Collector_status_t status = Collector_status_invalid_state;

    /* Are we in the right state? */
    if(clicState >= Clic_states_started)
    {
        Llc_deviceListItem_t item;

        /* Is the device a known device? */
        if(Csf_getDevice(pDstAddr, &item))
        {
            uint8_t buffer[SMSGGS_TOGGLE_LED_REQUEST_MSG_LEN];

            /* Build the message */
            buffer[0] = (uint8_t)Smsggs_cmdlds_toggleLedReq;

            sendMsg(Smsggs_cmdlds_toggleLedReq, item.devInfo.shortAddress,
            item.capInfo.rxOnWhenIdle,
            SMSGGS_TOGGLE_LED_REQUEST_MSG_LEN,
            buffer);
            status = Collector_status_success;
        }
        else
        {
            status = Collector_status_deviceNotFound;
        }
    }

    return(status);
}
```

Analysis

1. How long does it take from the time the sensor sent the message and received the instruction to toggle its red led? To do this, use the Clock_getTicks() function to collect the two time-stamps and then compute the difference. Remember that this function gives you the number of ticks elapsed where every tick is 10µs.

Evaluation

- Full credit will be provided if you implement the requirements of this project all the way to the Flow server. That is, the Flow server is the entity evaluating the threshold condition.
- Partial credit will be provided if you implement the requirements of this project all the way to the BBW only. That is, the BBW is the entity evaluating the threshold condition.

- Partial credit will be provided if you implement the requirements of this project all the way to the collector only. That is, the Collector is the entity evaluating the threshold condition.