

Baby ThermoSense

Remote temperature monitoring and reporting system

Project Summary



Submitted by **Gal Ron & Dvir Netanel**

Advanced Computer Systems Course

Prof. Sivan Toledo

April 2018, Tel Aviv University

Introduction

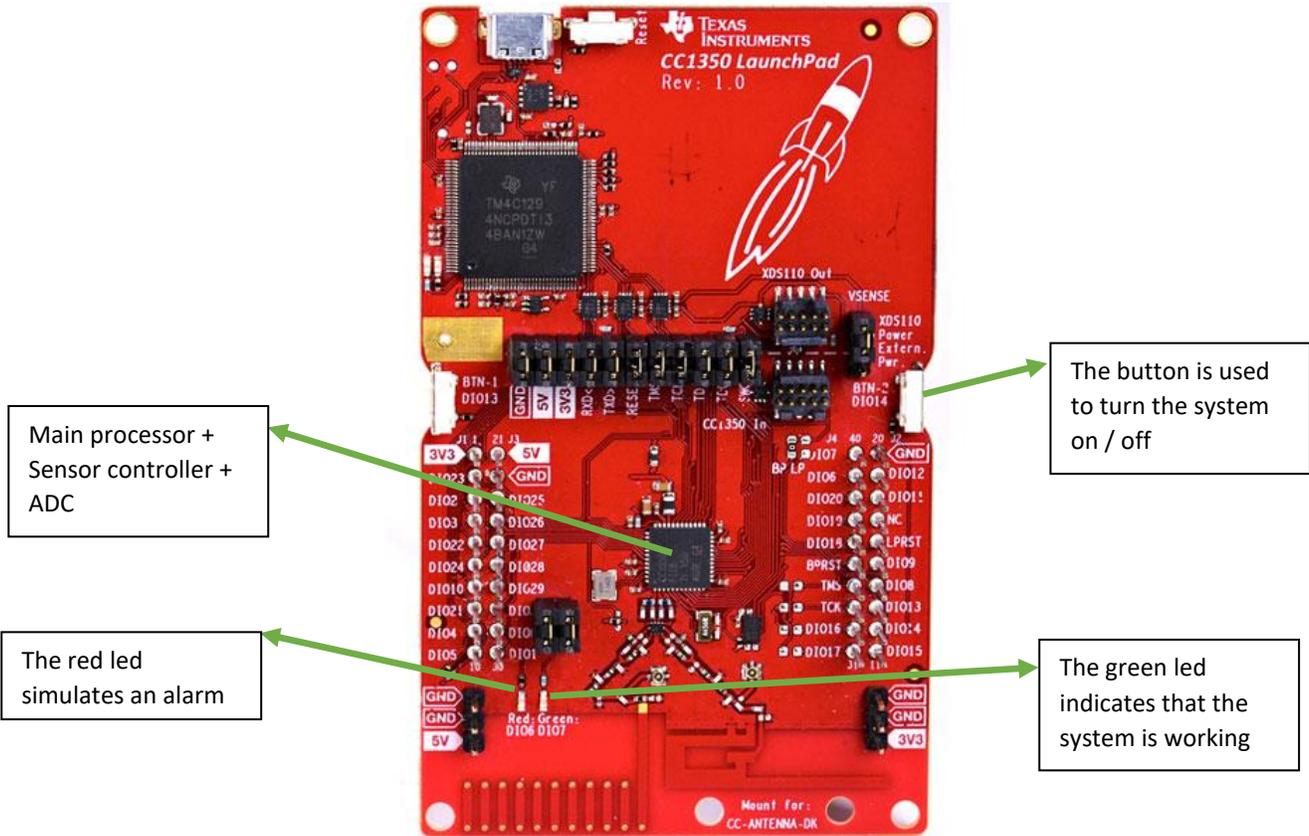
In this project we have implemented a BLE based temperature monitoring system. Our system can be used by parents to monitor the temperature of the room where their baby sleeps, and alert in case of extreme temperature changes (e.g. fire, cold breeze).

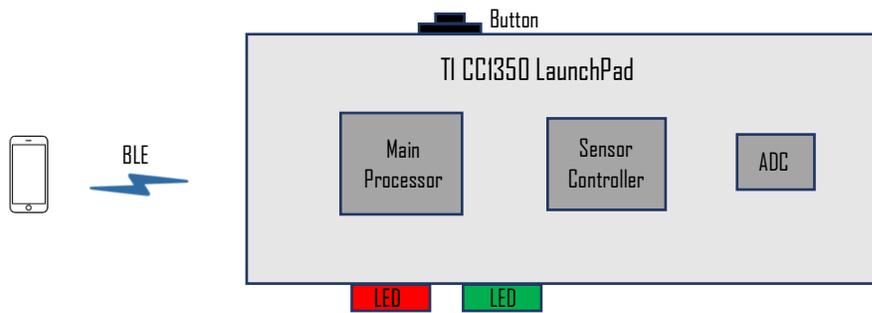
System overview

The system makes use of the **BLE** feature, **LEDs**, and **buttons** of the TI CC1350 microcontroller, and also implements **ambient temperature measurement** using the sensor controller (see implementation details).

The system allows the user to read the current room temperature, and also configure a range of “normal temperatures” (or use the pre-configured lower and upper bounds). The system constantly samples the temperature, and notifies the user when the measured temperature deviates from the normal bound by turning on the **red led**; this is to simulate the activation of an alarm. The user may turn-off the led (the alarm) by pressing a button, which turns off the whole system. The **green led** is used to indicate whether the system is turned on or off.

System diagram





Using the system

1. The user starts a temperature monitoring session by placing the launchpad in the room to be monitored and by setting a temperature range in his phone.
2. The temperature range is sent from the smartphone to the LaunchPad via BLE.
3. The LaunchPad starts monitoring the room's temperature and issues an alert in case the temperature exceeds the defined range (by lighting the red led, representing an alert).
4. Pressing the LaunchPad's button will cancel the alert.

Implementation Details

When implementing this project, we faced several challenges, such as:

- Creating a driver using the Sensor Controller Studio (SCS)
- Converting current to temperature in order to measure the room temperature in a precise manner
- Using the SCS driver and compiling it together with a project that uses BLE and other peripherals

Temperature Measurement

There are two main methods for measuring ambient temperature on the CC1350 LaunchPad:

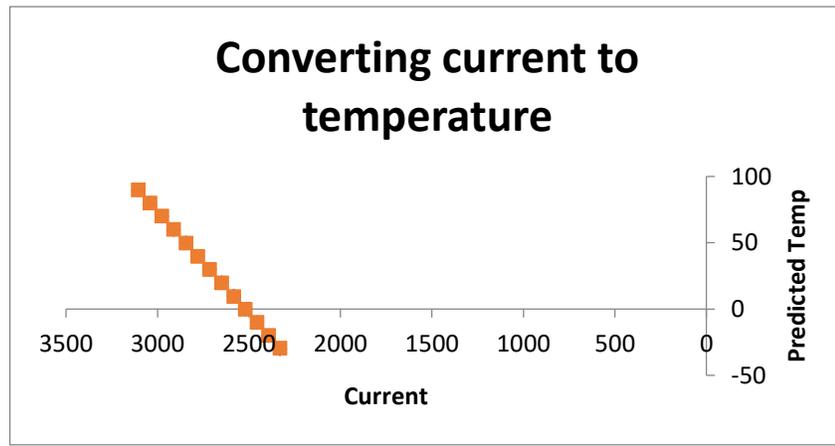
1. Reading the on-chip battery monitoring and temperature sensor, BATMON which performs alternate 6-bit measurements of both temperature and supply voltage.
2. Using the integrated 12-bit analog-to-digital converter (ADC) to measure ambient temperature by sampling an internal current source (ISRC). The ISRC is Proportional to the Absolute Temperature (PTAT), and thus gives a linear response as the ambient temperature changes.

We chose to use the second method due to its increased resolution and accuracy, which is critical for our project's functionality (The BATMON's temperature reading may deviate +/- 5 degrees Celsius from actual ambient temperature).

For querying the internal current source, we have used the Sensor Controller Studio for generating a driver which saves to RAM the 12-bit value of current read from the ADC. This current reading is converted to temperature based on a linear regression formula we deduced from a conversion table found in TI's documentation.

Linear regression:

90	3105
80	3041
70	2977
60	2912
50	2845
40	2780
30	2715
20	2650
10	2583
0	2521
-10	2457
-20	2394
-30	2331



$$\text{Temp} = 0.154412 * \text{Current} - 389.42$$

Generating the Sensor Controller Interface Driver

Using Sensor Controller Studio, we generated an interface driver which includes a single task for sampling the ADC periodically and saving the reading to a designated data structure on the board's RAM.

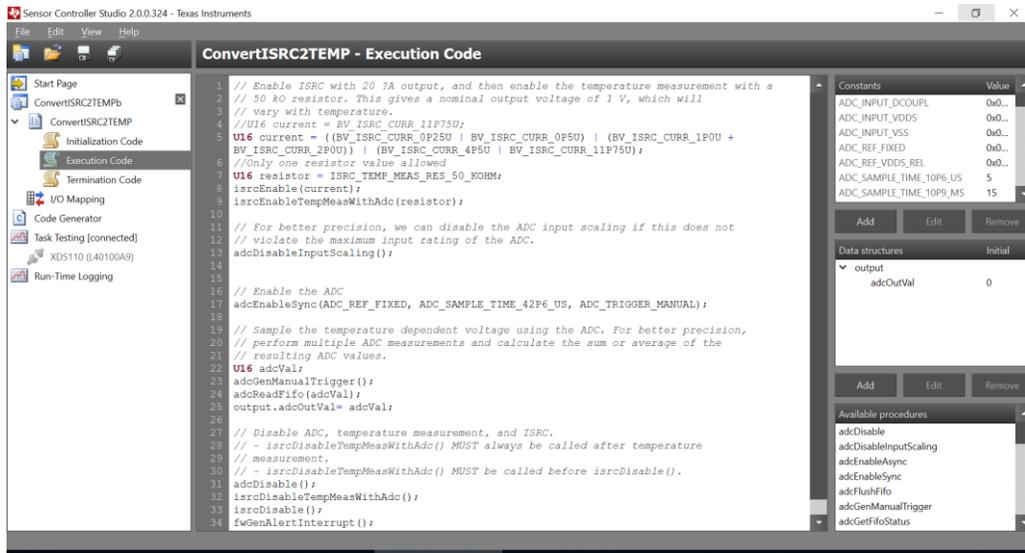


Figure – Sensor Controller Studio

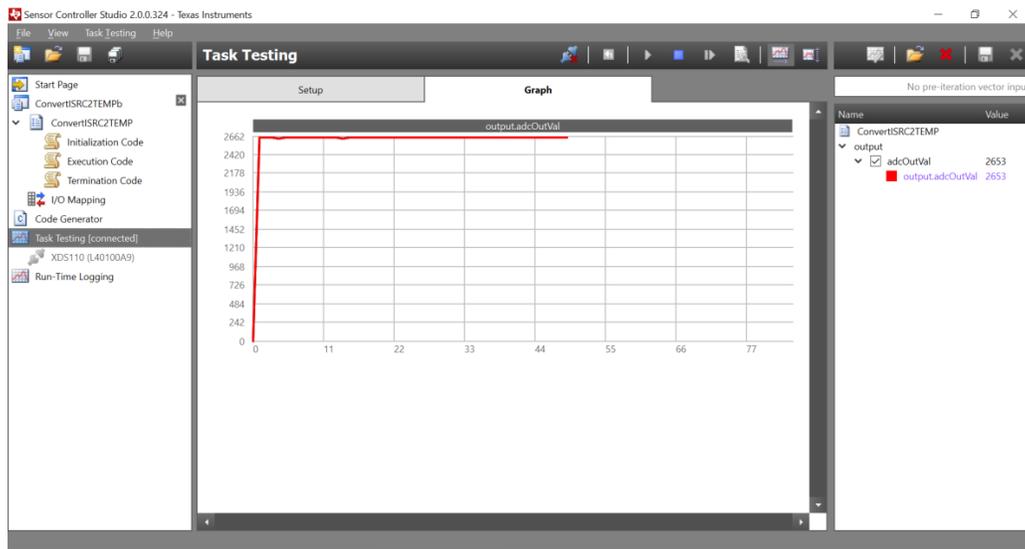


Figure – Temperature graph in SCS (Sensor Controller Studio)

The driver was tested on SCS’s Task Testing screen (see the screenshot) and compiled to c files that were added to the Code Composer Studio project containing the main controller’s code.

BLE communication

We created three GATT attributes to implement our system: one read-only attribute that allows the user to read the current temperature, and two read/write attributes that allow the user to set the upper and lower bounds for the normal temperature range (measuring a temperature out of this bound would cause an “alarm”).

The following screenshots demonstrate the use of our system using the new BLE functionality in Windows 10, with the “Bluetooth LE Explorer” Windows app.

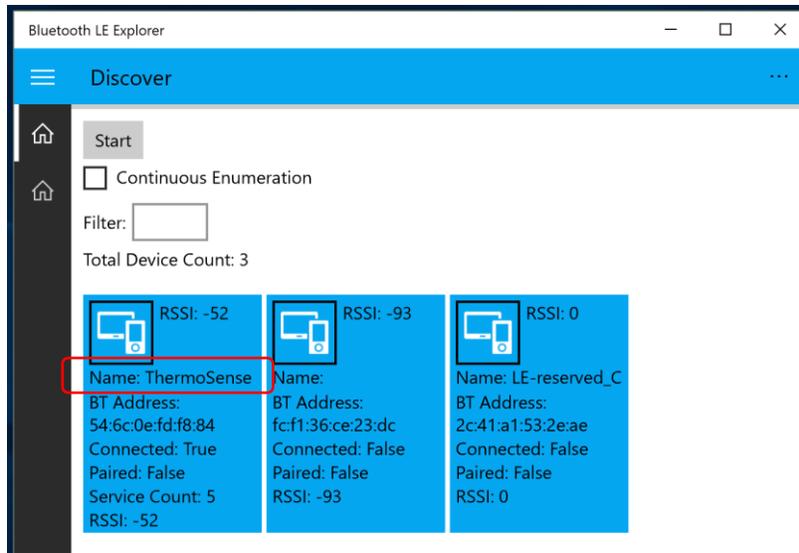


Figure – Our device in Windows 10 Bluetooth LE Explorer App

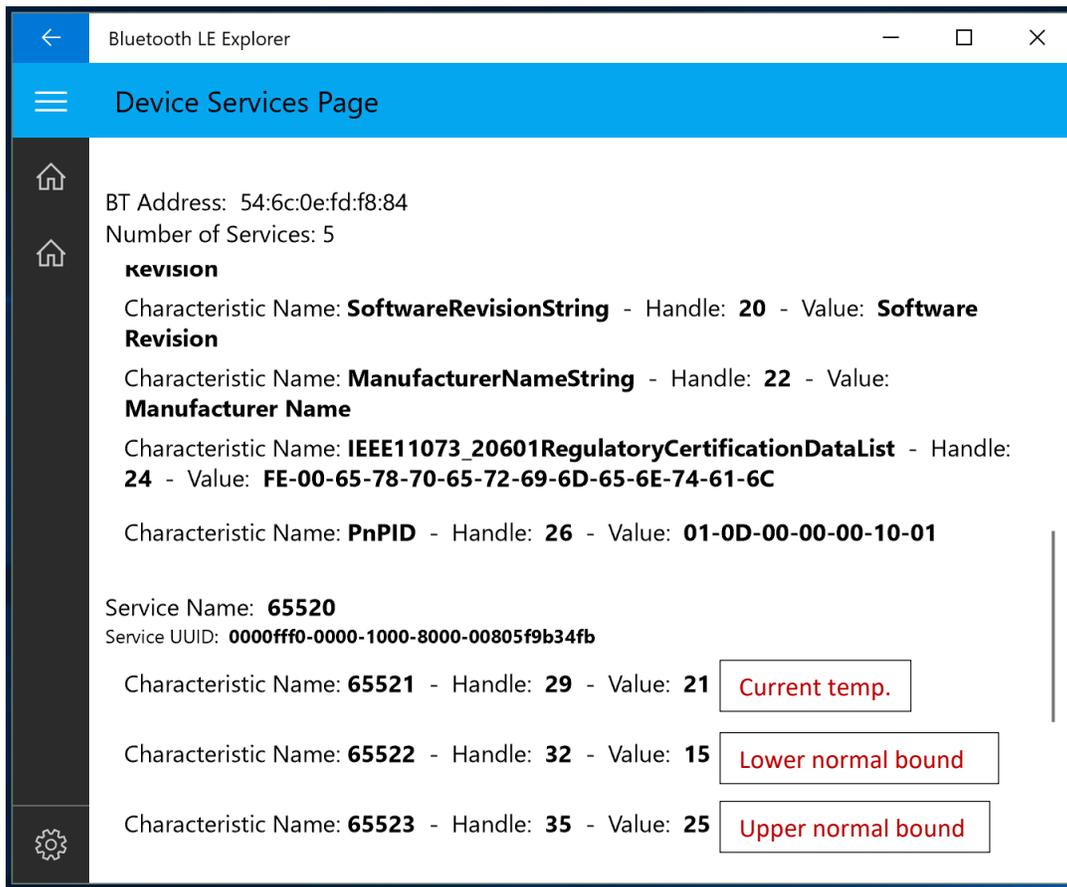


Figure – The GATT Attributes

Summary

In this project we used several features of the CC1350 LaunchPad by TI, including BLE communication, sensor sampling, main controller and sensor controller interaction, task scheduling and control of peripheral components like LEDs and buttons. The project demonstrates how the LaunchPad can be used to develop a prototype for a remote temperature monitoring system. Further development of the prototype into a mature product would involve adding a more accurate temperature sensor, adding a strong buzzer for alerting in case of a significant temperature deviation (instead of just turning on the red led) and packing the controller into a cool case that would allow its positioning in a baby room.