The Cook Manager System

Shiri Haim, ID:302633722

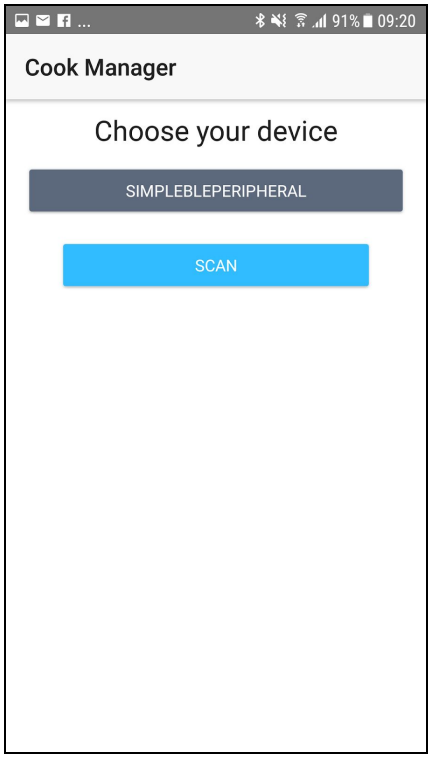
As a cooking and baking enthusiast and as someone who always tries to cook a few different dishes at the same time, I wanted to create an app that will manage the cooking and baking in the kitchen. The application allows you to define for each pot or pan on the stove, what is cooked on it. In addition, you can set a timer for the food that is put into the oven. Once the timer is finished or any of the materials in the pots reaches its boiling point (for each different material, there is a different boiling point), the application will alert. In this way there is one manager which handle all the different baking and cooking dishes in the kitchen and allows you to focus on the cooking itself.

Procedure

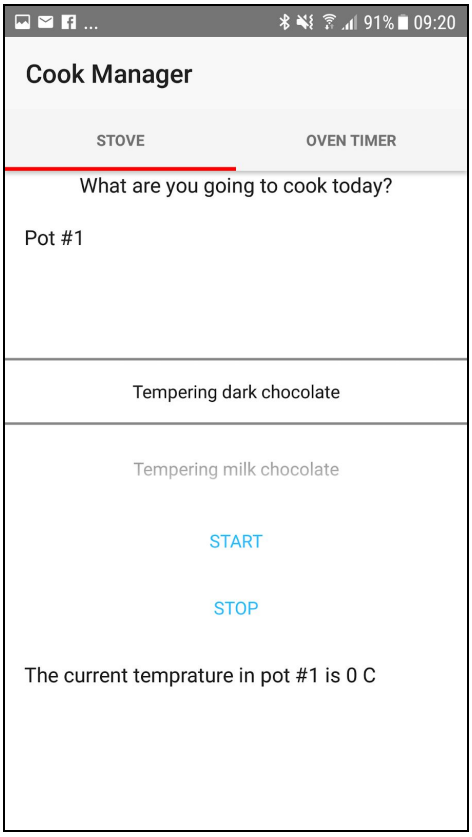
There are 3 main components involved in this system. The first, is the DS18B20 1-Wire Digital Thermometer. This is a waterproof temperature sensor which measures temperatures from -55℃ to +125℃. The second, is the CC1350 Launchpad. The third is an android application (written in NativeScript and tested on Galaxy S7). The following diagram describes the communication between those components:

Application description

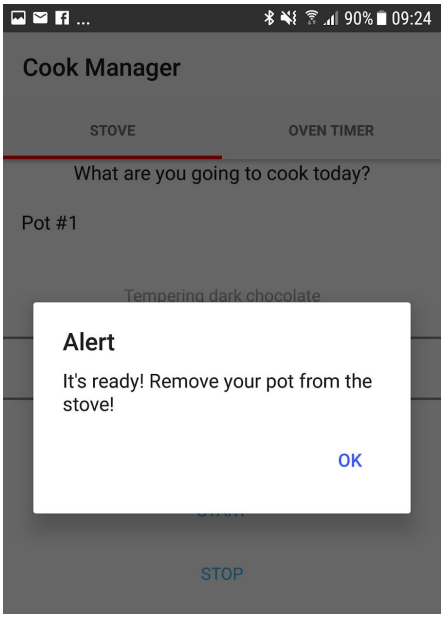
Assumes a client wants to get alert from the Cook Manager app when a dark chocolate in his pot are getting to 48℃ (the required temperature for dark chocolate tempering). He inserts the temperature sensor to the pot and put the pot on the stove.
Than he open the Cook Manager application and choose the appropriate bluetooth device (simple peripheral).



After the connection to the bluetooth device succeed, the client will routed to the manager screen. Then he will choose the 'Tempering dark chocolate' option, from the given list, and will tap on start button. From this point, the application will read the temperature from the sensor via the launchpad every 3 seconds and update the bottom label.

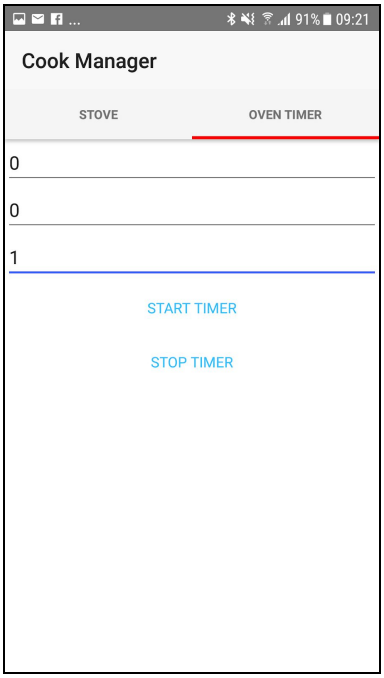


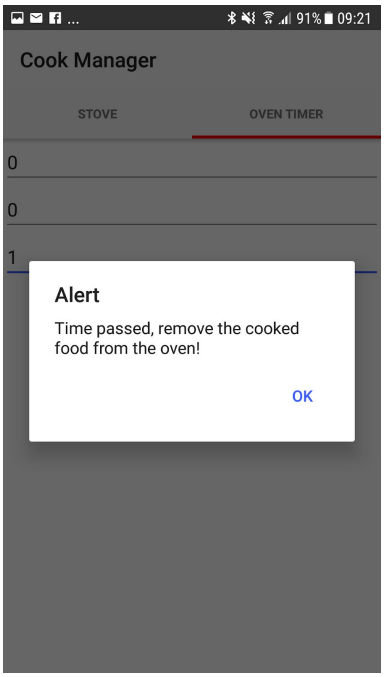
If the readed temperature is over the 48℃, the application alert to the client for removing the pot from the stove.



Additional application features

1. Simple timer for the oven. It alerts when the time is up.

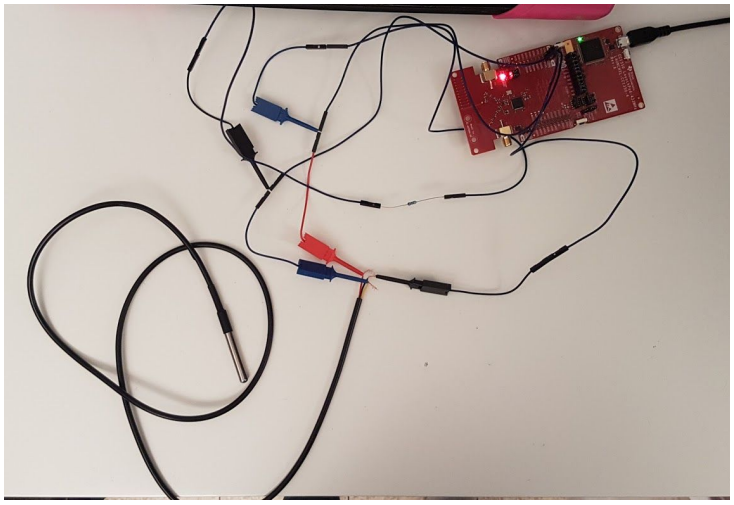


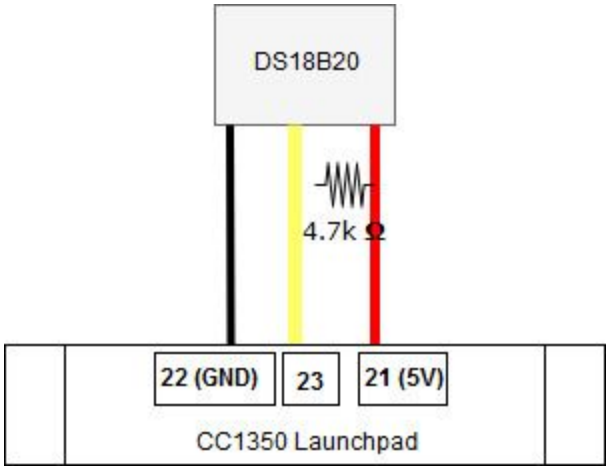


2. Stopping measure temperature in any time.
3. Choosing different materials and appropriate temperatures accordingly.

Implementation details

(1) *The hardware diagram and a real picture of the prototype*





(2) The communication between the DS18B20 sensor and the launchpad

The DS18B20 send the measured temperature to the launchpad over a 1-Wire interface. I had to implement this interface for our launchpad (see the DS18B20.c file), since I didn't found any library that should work with this launchpad. This file also contains a public getTemperature() function which do all the required process for reading values and returning the low and high values from the sensore. Those values should returned only when the temperature is stable (this is an optimization that I choose to do after I found that when starting measure the material, the first reads return values which growing fast).

Measuring temperature with 1-wire interface based on 3 steps: reset-convert-read. The reset function should consists of a reset pulse transmitted by the bus master (the launchpad in our case) which will followed by presence pulse(s) transmitted by the slave(s) (the sensor in our case). Then a temperature conversion is required. Sending this command to the sensor will convert the temperature and stored the thermal data

in the scratchpad memory in a 16-bit. Than a read command should be send twice, one for getting the low value, and the other for getting the high value.

(3) The communication between the launchpad and the android application

The launchpad firmware communicate with an android application over bluetooth. The readable characteristic 2 (FFF2) retrieve the measured temperature (invoking the getTemparture(..) function) from the sensor and send the results to the client. The client convert those 2 values to unsigned int.

The firmware based mainly on the simple_peripheral example project that we also used in the homeworks. A change has been made for the returned type of the characteristic 2. Instead of returning value of type uint8_t, an array of size 2 of type uint8_t is returned. This change has been made since in this implementation, characteristic 2 returns the getTemperature function returned value.

Results

For testing the results, I compared them to the results getting from a real digital temperature for food. I found that the differences were less than 2℃.

Difficulties and challenges

The big challenge in this project for me, was to understand how to compose the sensor to the launchpad and to implement the 1-wire interface for this launchpad. Since I had never compose electrical components, and I have no background in Electrical Engineering, I had to learn a lot for composing the sensor to the board in the

appropriate way. After getting some composition of the hardware, I tried to use some implementation of the 1-wire interface for the CC1350 Launchpad, which I found in the ti rtos forums (uploaded be a customer, not something official). This try failed and I got the same number from the sensor without relation to the measured temperature. The difficult was in analysys where are the problems, in the hardware or in the firmware. Since I got some value from the sensor, which was different from the value I got when the sensor was disconnected from the board, I decided to focus first on the interface implementation. After reading again and again the DS18B20 specifications, I found that I didn't wait enough time between the sending of the commands and that the reset function was not properly implemented for this board. I found some examples of implementations of this interface for arduino, and another example of implementation for SK40C + PIC16F877A board and used them and some information I read in the ti rtos documentation of this board for fixing those bugs in my implementation, and finally it worked.

Movie:

https://www.dropbox.com/s/6dyedhown91uw1z/TheCookManagerSystem.mp4?dl=0

Bibliography

DS18B20 specifications, https://www.4project.co.il/documents/doc_2761_2633.pdf

TI RTOS site and documentations for CC1350 Launchpad,

http://www.ti.com/tool/LAUNCHXL-CC1350

NativeScript site, https://www.nativescript.org/

Inspiration projects:

https://create.arduino.cc/projecthub/TheGadgetBoy/ds18b20-digital-temperature-sensor-and-arduino-9cc806

https://tutorial.cytron.io/2012/11/01/ds18b20-temperature-sensor/

https://e2e.ti.com/support/wireless_connectivity/simplelink_wifi_cc31xx_cc32xx/f/968/t/456513