SoRTES Project: Time-Synchronized Embedded Device

October 2020

1 Assignment

Smart environmental monitoring consists of a network of small, embedded devices that monitor a physical property in their surroundings, e.g. temperature. These sensing devices are typically connected through a wireless network with a gateway, to which they regularly transmit their measurements. In order to minimize the frequency of battery replacements, which can be costly and tedious, the deployed sensors enter a sleep mode in between transmissions to reduce energy consumption. The gateway transmits beacons that inform the sensing devices when to wake up to transmit their next measurement. It is thus crucial that the sensing device respects this real-time deadline, since failure to wake up at the specified time causes the sensing device to lose its time synchronization with the gateway.

2 Requirements

When booted, your micro-controller (sensing device) must keep its radio open until it receives a beacon. A beacon is sent by the gateway between every 2 to 10 seconds. Each beacon specifies the exact transmission time of the next beacon (the first four bytes correspond to the gateway ID, and the rest of the beacon message defines the next transmission time in seconds. For example, a sample message can be "GW0710", where GW07 the gateway ID, and 10 seconds is the time until the next beacon transmission. The gateway sends in total 20 beacons and prints out via the serial port metadata such as the number of beacons sent and the number of acks. Note that you should set the correct frequency on your micro-controller to be able to receive the beacons from the gateway, and make sure the gateway's spreading factor (SF) is set to 7 (default).

Once it receives the beacon, your micro-controller should write its next wake up time to the database in the micro-controller's EEPROM memory, along with the current temperature which is acquired by reading the built-in temperature sensor. Do note that you will need to calibrate the temperature sensor. Your micro-controller then transmits that temperature value to the gateway. Upon

completing the above two tasks, the micro-controller should enter a sleep mode until the next beacon transmission interval, which is specified in the received beacon from the gateway. After receiving 20 beacons from the gateway, the sensing device must enter an ultra low-power operation mode, in which you should strive to **minimize** power consumption. You can measure your board's power consumption by following the steps described in the video "Lab2 - PowerMonitoring (Short)" on Toledo. Additionally, your micro-controller should support three commands listed below, through a serial interface:

1. A command to read the latest temperature value and beacon details from database and print the output to serial port.
2. A command to read all temperature values and beacon details from database and print the output to serial port.
3. To enable low power operation mode. In this mode, you should strive to minimize the power consumption.

These three commands will be sent through the serial port using numbers 1,2 and 3 for the first, second and third commands respectively. Note that your micro-controller should only be able to respond to these commands when the board is booted, until the first beacon is received. Access to the database needs to be synchronized by means of synchronization primitives (i.e. mutex, semaphores) in order to ensure its consistency at all times. Its mandatory to use FreeRTOS for implementing your system and different tasks carrying the system's responsibilities.

3 Report

As part of the assignment, we expect you to write a report of up to 4 pages that describes in detail how your system satisfies:

1. the varying real-time constraint, specified by the gateway
2. task synchronization to ensure database consistency
3. low power consumption when
    - in low-power operation mode, between consecutive transmissions
    - in ultra low-power mode, after receiving 20 beacons

4 Submission

This is a graded assignment. You can submit your project by sending an email containing a .zip file (named firstname_lastname.zip) that includes your report and source code to stefanos.peros@kuleuven.be. The subject of this email must be [SoRTES: Submission], to ensure that we don't miss any submissions.

The deadline for submission is Thursday 3 December 2020, at 23:59 PM. For other questions regarding the project, you should preferably post a question on the discussion board (under Project Questions), or alternatively send an email to any of the instructors: ashoksamraj.thangarajan@kuleuven.be, emek-can.aras@kuleuven.be, stefanos.peros@kuleuven.be. You can of course also ask your questions during the weekly Q&A sessions.

Good luck! The SoRTES team.