Project, Part 3 – Due Friday 12/6 11:59 PM

In this final part of the project you are to implement an end-to-end data collection application on the Telosb nodes and PC. Imagine two scenarios. In the first one the Telosb is an interactive probe that streams measurements live and in the second one it is a data logger that gets deployed for a long period of time. In both cases a second Telosb is used to interact with the first one. Your task is to implement as efficient and reliable as possible:

1. A script that encodes user input
2. Transmits it over serial to one Telosb
3. Which relays the message to a second Telosb over the radio
4. The second Telosb executes the command
5. Transmits the result all the way back to the PC and displays it

Purpose

- To understand cross platform optimization and reliability.

Deliverables

- Code:

A tar/zip containing the application folder, compilable when extracted to the cs450 directory.

- Report (at most 6 pages, either single or double column, using 10pt type):

A PDF containing: Design Choices, Bug Report, Evaluation, Future Work, Summary.

Design Choices: You often have multiple choices when writing code. Let us know the reasoning behind the choices you made. Discuss the pros and cons.

Bug Report: If something is not working, that you know of, let us know.

Evaluation: Show us your application is working as intended, e.g., screen shots, logs, etc.

Future Work: How would you improve the application? You are encouraged to include changes to the constraints set by the assignment.

Remember, the only way we can know about any particularly clever solutions you've made, and give you credit for them, is if you tell us about them in the report. Also, if your code is not working properly, please document what you have done and achieved.

- Groups:

You are allowed to form groups (max. 3 students) and share the source code. However, you have to write and hand-in an individual report each.

Sample Application

In the GIT cs450 folder you will find a solution for Project 2 named **tosProject2Solution** and a sample Python script in **tosProject3** that will show you how to interact with the serial port. If you are not comfortable with Python you are welcome to use any other language to communicate with the serial port on the PC as long as your code will run in the Virtual Machine with the proper Debian packages installed.

The channel assignments are the same as last time but double check that your Makefile is set correct:

	Channel
Dunyuan	11
Zihan	12
Renyuan	13
Gregory Chandler	14
Noah Lampel	15
Kunal Anil	16
Siwei	17
Luke Emmett	18
Nir Michael	19
Michael Anthony	20
Nicholas Jerome	21
Yanan	22

Assignment

1. Implement a reliable data collection application.

   Communication between the PC and the second Telosb should be reliable, meaning:
   a. Any UART corruption must be detected
   b. All commands and responses send over the UART must be received
   c. All commands and responses send over the radio must be received
   (note: you can rely on the radio to filter out corrupt radio packets)

Samples stored to FLASH should be protected against data corruption, meaning if the sample is different (when read from the FLASH) from what was written it should be discarded.

The implementation should support the command set from Project 2, meaning the user should be able to type in commands and parameters on the PC and then have the application encode it properly and send it to the Telosb.

2. Optimize your application.

   Use all the things covered in the class and exercises and papers to optimize the application:

   a. Minimize the code size (reported as ROM when compiled)
   b. Minimize the memory usage (reported as RAM when compiled)
   c. Minimize the radio duty cycle
   d. Maximize UART bandwidth
   e. Maximize radio bandwidth

   Hints:

   a. Mix regular C code into your TinyOS application
   b. Change the maximum packet size on the radio (it supports up to ~118 bytes)
   c. Profile the application with **romsize.pl** on the **main.exe** file in the **build/telosb** directory
   d. Human readable numbers are not stored very efficiently
   e. Ask questions on Piazza!

Grading

There are many different solutions to the assignment above and it is up to you to choose which one to implement and discuss your choice in the report. We will grade the assignment based on (1) how well it works, (2) how robust the implementation and communication protocol are, (3) the optimizations implemented, and (4) the written report.

For the optimizations, you should specifically state what you have done and how big the improvements are. You will get points for both *how many* optimizations and *how efficient* they are.

You will get *bonus* points for using your own **Project 2** solution but you will not get optimization points for things already implemented in the **tosProject2Solution** application. It is still possible to get full points using the sample application.