Updated List of projects:

1 Time synchronization support  
  
This project provides a service in LiteOS that synchronizes between  
multiple nodes. There have been several papers on time synchronization  
 in the literature, but the current implementation has been limited to  
TinyOS. This project will be helpful to time related services in the  
LiteOS system. Technically, this project requires reading time  
synchronization papers and learning how to implement them using timers  
in LiteOS.  
  
2 Implementing EnvrioTrack based on LiteOS  
  
This project implements the EnviroTrack as a library in the LiteOS  
environment. It requires reading the original EnviroTrack paper,  
reading its implementation in the TinyOS operating system environment,  
and porting this implementation to the LiteOS system environment.  
  
3 RPC Support in LiteOS  
  
RPC support in LiteOS allows the user to implement applications on different nodes, where one node may remotely invoke another node’s service via procedural calls. Similar to common RPCs, the design of this service need to consider the scenarios of node failures as well as message losses. The delay of RPCs will also be significantly longer than local functional calls.   
  
  
4 Support for Tmote and Telos  
  
Similar to above, but requires changes to the kernel as well. So this  
project is much more difficult compared to 3.  Requires learning the  
toolchain of MSP430 MCU and its sensors.  
  
  
5 Adding Eclipse front-end to LiteOS (implemented to some extent, but)  
  
In this project, we want to provide a front-end for LiteOS that allows  
IDE based programming and operations. This will require extending  
Eclipse and a lot of Java programming. The LiteOS toolchain is ready  
to be used, and should be able to directly invoked from the Eclipse  
environment.  
  
  
6 GUI-like front-end for LiteOS  
  
This project aims to represent the terminal like environment with a  
File Explorer like environment, where the user may use the mouse to  
drag between icons to install, kill, start, and stop applications on  
the sensor nodes. Therefore, a sensor network appears as a sensor  
network drive. Not sure what language to use to implement this  
Explorer like environment, but it should require substantial  
programming to make this feasible.  
  
7 Security features  
  
Adding security for the shell, including encrypting the log-in,  
log-out, messages, and protect against common types of attacks such as  
replay attacks and wormhole attacks.