Threading

Garrett Berg, Cloudform Design, garrett@cloudformdesign.com

Summary

The threading.h library was created to make threading simpler for microcontrollers, while still using a very small footprint. It can also be integreated with the User Interface (ui.h) library to create a simple user interface

This module supports threading through the *protothreads* library. Each function has an associated thread that can be called. This library can make it significantly easier to manage multiple processes. See the **threading_led_example2**. Also see **Userguide_protothreads** for details on how to use protothreads.

Overview

- **expose_threads(...) ::**
    - This is how you expose threads. Each argument must be a function that takes a single variable of type **(pthread *)** and returns a variable of type **uint8_t**.

```
uint8_t mythread(pthread *pt){
  // stuff
}

expose_threads(TH_T(mythread));
```

- **setup_threading() ::**
    - Must be called in the **setup** function, before any threads are scheduled or used. You must have already exposed the threads.
    - If you are using the ui.h module, use **setup_ui()** instead
- **schedule_thread(uint8_t thread_index) ::**
    - Schedules a thread to be called by thread_loop

- **thread_loop() ::**
    - This should be called in your loop function. It is what checks to see if there is new input and if any threads need to be run.
    - If you are using the ui.h module, call **ui_loop()** instead
- **kill_thread(uint8_t) ::**
    - the thread's pt is set to KILL. When the thread is next called, it will jump to PT_KILL if it is defined, or simply exit if it is not.

Document generated by eLyXer 1.2.3 (2011-08-31) on 2014-04-16T11:54:30.876871