Samson Kaller – 1277529
Lab #5 - Watchdog timer

Objectives:

- Improve the previous project by adding a watchdog
- Install and setup a Watch Dog timer in a RTOS environment

To hand in:

- Answers to all questions. Integrate with vending machine

Lab Work

This lab must be merged with the vending machine.

Part1: Configuring the WDT

- Configure the watchdog timer for a 8 second delay.

- Program the board and test your code. What do you observe? Comment.

After 8 seconds the board automatically resets. Whatever the code is currently executing will be interrupted and the program restarts.

- Modify your code to turn on a LED whenever the watchdog "bites" and resets the board.

- Test your code by programming the device in release mode.
- Now kick the dog in the IDLE HOOK task. Program the device in release mode and explain what happens:

With the watchdog timer being reset in the IdleHook, the board no longer resets and the program runs as expected.

Part 2: Hogging the processor

- In main.c , add a low priority task named vTaskHog. Inside this task add a while(1) loop that will hog the CPU and give it no slack. Program the device in release mode and explain what happens:

Since there is a while(1) loop in vTaskHog, the scheduler will run this task whenever the others are blocked. The IdleTask no longer runs, and consequently the WTD is never reset. This means the code will reset every 8 seconds, and the LED indicators for the WTD timeout are turned ON.

Approval _____

Questions:

Q1- Explain why we use watchdogs in embedded systems. Give examples.

Watchdog timers are useful for detecting software failures, which may occur if any task or operation performed by a microcontroller hangs for an indefinite amount of time, which might cause a system failure. Good system designers plan for software failure even if they have a quality code, so if the watchdog timer detects a process is taking too long to execute it will reset the system and hopefully un-bug the code. The watchdog time-out can be programmed for a desired time interval, taking into account the required intervals to perform tasks.

Examples:

- Space probes which aren't accessible for reset by humans implement a Watchdog time-out reset if a system failure occurs.
- Things with no user interface and which require little human supervision, if they do hang a watchdog would be useful to prevent performance issues.

Q2- After implementing vTaskHog task, explain why the board resets even though the watchdog was kicked in the Idle Hook task.

While the vTaskHog is implemented at the lowest priority, it still has priority over the IdleHook where the software "kicks the dog" and resets the Watchdog timer. vTaskHog never blocks and loops in an endless while(1) loop, and so no spare processing time is available to run the IdleHook, therefore the Watchdog timer is never cleared and therefore the board resets after the watchdog timer elapses.