```c
/*
 * clockControl.c
 *
 *  Created on: May 19, 2015
 *      Author: Taylor Cowley
 */

#include "clockControl.h"
#include "clockDisplay.h"
#include "supportFiles/display.h"
#include <stdio.h>


// States for the controller state machine.
enum clockControl_st_t {
    init_st,                // Start here, stay in this state for just one tick.
    never_touched_st,       // Wait here until the first touch - clock is disabled until set.
    waiting_for_touch_st,   // waiting for touch, clock is enabled and running.
    ad_timer_running_st,    // waiting for the touch-controller ADC to settle.
    auto_timer_running_st,  // waiting for the auto-update delay to expire
    // (user is holding down button for auto-inc/dec)
    rate_timer_running_st,  // waiting for the rate-timer to expire to know when to perform the
auto inc/dec.
    rate_timer_expired_st,  // when the rate-timer expires, perform the inc/dec function.  NOT
USED
    add_second_to_clock_st  // add a second to the clock time and reset the ms counter.    NOT
USED
} currentState = init_st;

uint16_t adTimer = 0;
uint16_t autoTimer = 0;
uint16_t rateTimer = 0;
uint16_t clockTimer = 0;

void clockControl_tick() {
    // Perform state action first.
    switch(currentState) {
    case init_st:               //we init everything!
        clockDisplay_init();    //init Our clock display!
        adTimer = 0;            //zero the values (just in case)
        autoTimer = 0;          //zero the values (just in case)
        rateTimer = 0;          //zero the values (just in case)
        clockTimer = 0;         //zero the values (just in case)
        break;
    case never_touched_st:      //uh, nothing? we just idle
        break;
    case waiting_for_touch_st:  //Our seconds should be ticking! :)
        clockTimer = clockTimer + 1;//tick our clock
        break;
    case ad_timer_running_st:   //Waiting for the analog/digital systems to cool down
        adTimer = adTimer + 1;      //increment the timer
        break;
    case auto_timer_running_st: //Waiting the 0.5 seconds to start counting up fast
        autoTimer = autoTimer + 1;  //increment the timer
        break;
    case rate_timer_running_st: //Waiting the 0.1 seconds to tick up one fast!
        rateTimer = rateTimer + 1;  //increment the timer
```

```
        break;
    default:            //This signifies an error
        printf("clockControl_tick state action: hit default\n\r");      //print the error
        break;
    }

    // Perform state update next.
    switch(currentState) {
    case init_st:
        currentState = never_touched_st;    //we only spend one tick in init
        break;
    case never_touched_st:                      //we wait until a touch
        if(display_isTouched()){                //are we touched? if yes
            display_clearOldTouchData();        //clear the data for good measure
            currentState = waiting_for_touch_st;//move to the next state
        }
        break;
    case waiting_for_touch_st:              //waiting for the touch to increment something
        if(clockTimer >= CLOCK_TIMER_EXPIRED){  //have we ticked to a second? if yes
            clockTimer = 0;                     //reset our clock
            clockDisplay_advanceTimeOneSecond();//and we advance a second!
        }
        if(display_isTouched()){                //are we touched? if yes
            display_clearOldTouchData();        //clear the data for good measure
            currentState = ad_timer_running_st; //on to waiting for the analog/digital cooldown
        }
        break;
    case ad_timer_running_st:               //waiting for the analog/digital cooldown
        if(adTimer >= AD_TIMER_EXPIRED){    //we only update the state after the analog/digital
cooldown
            adTimer = 0;                            //reset the timer for the next time
            if(!display_isTouched()){               //we aren't touched anymore
                clockDisplay_performIncDec();       //update the display for the very short touch
                currentState = waiting_for_touch_st;//and we wait for the next touch
            } else {                                //we are still being touched!
                currentState = auto_timer_running_st;//time to wait for the 0.5 seconds
            }
        }
        break;
    case auto_timer_running_st:
        if(!display_isTouched()) {                  //Aw they let go during the 0.5 secs
            clockDisplay_performIncDec();           //update just one time for the short touch
            currentState = waiting_for_touch_st;    //and we wait for the next touch
        } else if(autoTimer >= AUTO_TIMER_EXPIRED) {//still being touched and we have reached 0.5
secs!
            autoTimer = 0;
            clockDisplay_performIncDec();           //start our fast add with an update!
            currentState = rate_timer_running_st;   //and we start updating really fast!
        }
        break;
    case rate_timer_running_st:
        if(!display_isTouched()) {                  //aw they let go
            currentState = waiting_for_touch_st;    //and wait for the next touch
        } else if(rateTimer >= RATE_TIMER_EXPIRED) {//Still touching us and we made it to 0.1
secs!
            rateTimer = 0;                          //WOO! Time to count up again!
            clockDisplay_performIncDec();           //and update the screen!
```

```
        }
        break;
    default:        //This is an error
        printf("clockControl_tick state update: hit default\n\r");  //print the error
        break;
    }
}
```