

```

/**
 * WireTouchDB.c
 * This file contains the service to debounce the wire game WireTouch event.
 */

#include "ES_Configure.h"
#include "ES_Framework.h"
#include "PWMTiva.h"
#include "WireTouchDB.h"
#include "Mummy.h"

#include "inc/hw_memmap.h"
#include "inc/hw_types.h"
#include "inc/hw_gpio.h"
#include "inc/hw_sysctl.h"

#define DB_TIME 1000 // WireTouch debounce time [ms]

static bool timerOn = false; // variable to store whether or not the debounce timer has been activated
static uint8_t MyPriority; // module-level variable for the VibrateMotor module's priority

/* Prototypes */
bool InitializeWireTouchDB(uint8_t Priority);
ES_Event RunWireTouchDB(ES_Event CurrentEvent);
bool PostWireTouchDB(ES_Event ThisEvent);
static uint8_t wireState(void);

/**
 * Initializes the WireTouchDB service.
 * @param Priority: priority of the WireTouchDB service
 */
bool InitializeWireTouchDB(uint8_t Priority) {
    ES_Event ThisEvent;
    MyPriority = Priority; // set the priority

    ThisEvent.EventType = ES_INIT; // post the initial transition event
    if (ES_PostToService(MyPriority, ThisEvent) == true) {
        return true;
    } else {
        return false;
    }
}

/**
 * Runs the WireTouchDB service.
 * @param CurrentEvent: the current event that has occurred
 */
ES_Event RunWireTouchDB(ES_Event CurrentEvent) {
    ES_Event ReturnEvent;
    ReturnEvent.EventType = ES_NO_EVENT; // assume no problems will occur

    if ((CurrentEvent.EventType == WireTouchDB) && (!timerOn)) {
        ES_Timer_InitTimer(WIRE_DB_TIMER, DB_TIME); // start the WireTouch debounce timer
        timerOn = true; // timer is now on
        //printf("Start debouncing...\r\n");
    } else if ((CurrentEvent.EventType == ES_TIMEOUT) && (CurrentEvent.EventParam == WIRE_DB_TIMER)) {
        timerOn = false; // timer has now expired
        //printf("Stop debouncing\r\n");
        if (wireState() != 0) {
            ES_Event ThisEvent;
            ThisEvent.EventType = WireTouch;
            PostMummySM(ThisEvent); // only post WireTouch event if wire is still high after debouncing
        }
    }
    return ReturnEvent;
}

```

```
/**
 * Posts events to the WireTouchDB service.
 * @param ThisEvent: event to post to WireTouchDB service
 * @return true if successfully posted; false otherwise
 */
bool PostWireTouchDB(ES_Event ThisEvent) {
    return ES_PostToService(MyPriority, ThisEvent);
}

/**
 * Samples the state of the wire.
 */
static uint8_t wireState(void) {
    return (WIRE_PORT & WIRE_PIN); // return the state of the wire
}
```