*/
* File:   LED.h
* Author: Elkaim
*
* Software module to access the 12 LEDs on the Uno32 I/O protection board. These
* are arranged in three banks (Bank 1, Bank2, and Bank3) with Bank 1 being red,
* Bank 2 yellow, and Bank 3 green.
* The individual LEDs can be addressed directly using the appropriate LAT pins
* as a bit level manipulation (0 turns LED on, 1 turns LED off), or the provided
* functions allow them to be used as banks or a whole 12bit array.
* The convention is that LED_BANKx_Y where x is 1,2, or 3 and Y is 0,1,2, or 3
* with the 0 bit being towards the interior of the I/O board and 3 being towards
* the edge.
* NOTE: In order for the LEDs to light, the LED CONTROL jumer for the corresponding
* bank must be in place. SPI_MASTER should be defined if both jumpers on the
* SPI CONTROL are towards the top of the board.
* LED_TEST (in the .c file) conditionally compiles the test harness for the code.
* Make sure it is commented out for module usage.
* Created on December 18, 2011, 7:58 PM
* Modified on October 21, 2013, 4:22 PM
*/

#ifndef LED_H
#define LED_H
/*******************************************************************************
* PUBLIC #Defines                                                           *
******************************************************************************/
#define JP_SPI_MASTER
//#define JP_SPI_SLAVE
#define LED_BANK1 0x01
#define LED_BANK2 0x02
#define LED_BANK3 0x04
#define LED_BANK1_3 LATDbits.LATD6
#define LED_BANK1_2 LATDbits.LATD11
#define LED_BANK1_1 LATDbits.LATD3
#define LED_BANK1_0 LATDbits.LATD5
#define LED_BANK2_3 LATFbits.LATF6
#define LED_BANK2_1 LATDbits.LATD7
#define LED_BANK2_0 LATGbits.LATG8
#define LED_BANK3_3 LATBbits.LATB0
#define LED_BANK3_2 LATFbits.LATF5
#define LED_BANK3_1 LATFbits.LATF4
#define LED_BANK3_0 LATGbits.LATG6

/*******************************************************************************
* PUBLIC FUNCTION PROTOTYPES                                                  *
******************************************************************************/
/**
* @Function LED_Init(void)
* @param
* @return SUCCESS or ERROR
* @brief Initializes the LED subsystem
* and turns the LED on.
* @author Gabriel Hugh Elkaim, 2011.12.25 01:16 */
char LED_Init(void);
/**
* @Function LED_AddBanks(unsigned char AddBanks)
* @param AddBanks - an unsigned char with a 1 in each position to set the bank as an
* LED bank, should be a bitwise OR of the #define'd LED_BANKx values.
* @return SUCCESS or ERROR
* @brief Adds selected banks to the system and sets them to be digital outputs
* @note J3 SPI Jumpers should be set in MASTER position.
* @author Gabriel Hugh Elkaim, 2011.12.25 01:16 */
char LED_AddBanks(unsigned char AddBanks);
/**
* @Function LED_RemoveBanks(unsigned char RemoveBanks)
* @param RemoveBanks - an unsigned char with a 1 in each position to remove the
* bank or banks as LED outputs. Should be a bitwise OR of the
* #define'd LED_BANKx values.
* @return SUCCESS or ERROR
* @brief Removes the indicated banks from the system and sets them to be digital inputs
* @note J3 SPI Jumpers should be set in MASTER position.
* @author Gabriel Hugh Elkaim, 2011.12.25 01:16 */
char LED_RemoveBanks(unsigned char RemoveBanks);
/**
* @Function LED_ActiveBanks(void)
* @param
* @return Listing of all LED Banks that are currently active.
* @brief Returns a variable of all active LED BANKS. An individual BANK can be
* determined active by "anding" with the LED_BANKx Macros
* @author Max Dunne, 2013.08.15 */
unsigned int LED_ActiveBanks(void);
/**
* @Function LED_GetBank(unsigned char bank)
* @param
* @return
* @brief
* @author
*/
*@param* bank - one of the #define'd LED_BANKx values.
* @return ERROR or state of BANK in lowest 4 bits
* @author Max Dunne, 203.10.21 01:16 */
char LED_GetBank(unsigned char bank);

/**
 * Function: LED_OnBank(unsigned char bank, unsigned char pattern);
 * @param bank - use #defined LED_BANKx [1,2, or 3] only one
 * @param pattern - which LEDs to turn ON
 * @return SUCCESS or ERROR
 * @brief Sets each LED in banks (LED_BANKx) to ON for every matching 1 on pattern.
 * 0's in the pattern are left unchanged.
 * @author Gabriel Hugh Elkaim, 2011.12.25 01:16 */
char LED_OnBank(unsigned char bank, unsigned char pattern);

/**
 * Function: LED_OffBank(unsigned char bank, unsigned char pattern);
 * @param bank - use #defined LED_BANKx [1,2, or 3] only one
 * @param pattern - which LEDs to turn OFF
 * @return SUCCESS or ERROR
 * @brief Sets each LED in banks (BANKx) to OFF for every matching 1 on pattern.
 * 0's in the pattern are left unchanged.
 * @author Gabriel Hugh Elkaim, 2011.12.25 01:16 */
char LED_OffBank(unsigned char bank, unsigned char pattern);

/**
 * Function LED_InvertBank(unsigned char bank, unsigned char pattern)
 * @param bank - one of the #define'd LED_BANKx values.
 * @param pattern - which LEDs to toggle.
 * @return SUCCESS or ERROR
 * @brief Forces the LEDs in (bank) to toggle for every matching 1 on pattern.
 * @author Gabriel Hugh Elkaim, 2011.12.25 01:16 */
char LED_InvertBank(unsigned char bank, unsigned char pattern);

/**
 * Function: LED_SetBank(unsigned char bank, unsigned char pattern)
 * @param bank - use #defined BANKx [1,2, or 3] only one
 * @return SUCCESS or ERROR
 * @brief Sets each LED in bank (BANKx) to match pattern, 1 turns LED on, 0 turns LED off. All pins in the bank are affected.
 * @author Gabriel Hugh Elkaim, 2011.12.25 01:16 */
char LED_SetBank(unsigned char bank, unsigned char pattern);

/**
 * Function LED_GetBank(unsigned char bank)
 * @param bank - one of the #define'd LED_BANKx values.
 * @return char: ERROR or state of BANK
 * @author Max Dunne, 203.10.21 01:16 */
char LED_GetBank(unsigned char bank);

/**
 * Function LED_End(void)
 * @param None
 * @return SUCCESS or ERROR
 * @brief Shuts down the LED subsystem
 * @author Gabriel Hugh Elkaim, 2011.12.25 01:16 */
char LED_End(void);