#ifndef pwm_H
#define pwm_H

/******************************************************************************
* PUBLIC #DEFINES
******************************************************************************/
#define MIN_PWM_FREQ 100
#define PWM_500HZ 500
#define PWM_1KHZ 1000
#define PWM_2KHZ 2000
#define PWM_5KHZ 5000
#define PWM_10KHZ 10000
#define PWM_20KHZ 20000
#define PWM_30KHZ 30000
#define PWM_40KHZ 40000
#define MAX_PWM_FREQ 100000
#define PWM_DEFAULT_FREQUENCY PWM_1KHZ

#define PWM_PORTZ06 (1<<0)
#define PWM_PORTY12 (1<<1)
#define PWM_PORTY10 (1<<2)
#define PWM_PORTY04 (1<<3)
#define PWM_PORTX11 (1<<4)
#define MIN_PWM 0
#define MAX_PWM 1000

/******************************************************************************
* PUBLIC FUNCTION PROTOTYPES
******************************************************************************/

/**
* @Function PWM_Init(void)
* @param None
* @return SUCCESS or ERROR
* @brief Initializes the timer for the PWM system and set to is to the default frequency
* @note None.
* @author Max Dunne
* @date November 12, 2011, 9:27 AM
*/
char PWM_Init(void);

/**
* @Function PWM_SetFrequency(unsigned int NewFrequency)
* @param NewFrequency - new frequency to set. best to use #defined from header
* @return SUCCESS OR ERROR
* @brief Changes the frequency of the PWM system.
* @note Behavior of PWM channels during Frequency change is undocumented
* @author Max Dunne
* @date 2013.08.19
*/
char PWM_SetFrequency(unsigned int NewFrequency);

/**
* @Function PWM_GetFrequency(void)
* @return Frequency of system in Hertz
* @brief gets the frequency of the PWM system.
* @author Max Dunne
* @date 2013.08.19
*/
unsigned int PWM_GetFrequency(void);

/**
* @Function PWM_AddPins(unsigned short int AddPins)
* @param AddPins - use #defined PWM_PORTxxx OR'd together for each A/D Pin you wish to add
* @return SUCCESS OR ERROR
* @brief Adds new pins to the PWM system. If any pin is already active it errors out
* @author Max Dunne
* @date 2013.08.15
*/
char PWM_AddPins(unsigned short int AddPins);

/**
* @Function PWM_RemovePins(unsigned int PWMPins)
* @param RemovePins - use #defined AD_PORTxxx OR'd together for each A/D Pin you wish to remove
* @return SUCCESS OR ERROR
* @brief Remove pins from the PWM system. If any pin is not active it errors out
* @author Max Dunne
* @date 2013.08.15
*/
char PWM_RemovePins(unsigned int PWMPins);

/**
* @Function PWM_ListPins(void)
* @param None
* @return Listing of all PWM pins that are active.
* @brief Returns a variable of all active pwm pins. An individual pin can be determined
* if active by "anding" with the pwm_PORTXX Macros
* @author Max Dunne
* @date 2013.08.19
*/
unsigned int PWM_ListPins(void);

/******************************************************************************
* PUBLIC FUNCTION PROTOTYPES
******************************************************************************/

/**
* @Function PWM_Init(void)
* @param None
* @return SUCCESS or ERROR
* @brief Initializes the timer for the PWM system and set to is to the default frequency
* @note None.
* @author Max Dunne
* @date 2013.08.25
*/
char PWM_Init(void);

/**
* @Function PWM_SetFrequency(unsigned int NewFrequency)
* @param NewFrequency - new frequency to set. best to use #defined from header
* @return SUCCESS OR ERROR
* @brief Changes the frequency of the PWM system.
* @note Behavior of PWM channels during Frequency change is undocumented
* @author Max Dunne
* @date 2013.08.19
*/
char PWM_SetFrequency(unsigned int NewFrequency);

/**
* @Function PWM_GetFrequency(void)
* @return Frequency of system in Hertz
* @brief gets the frequency of the PWM system.
* @author Max Dunne
* @date 2013.08.19
*/
unsigned int PWM_GetFrequency(void);

/**
* @Function PWM_AddPins(unsigned short int AddPins)
* @param AddPins - use #defined PWM_PORTxxx OR'd together for each A/D Pin you wish to add
* @return SUCCESS OR ERROR
* @brief Adds new pins to the PWM system. If any pin is already active it errors out
* @author Max Dunne
* @date 2013.08.15
*/
char PWM_AddPins(unsigned short int AddPins);

/**
* @Function PWM_RemovePins(unsigned int PWMPins)
* @param RemovePins - use #defined AD_PORTxxx OR'd together for each A/D Pin you wish to remove
* @return SUCCESS OR ERROR
* @brief Remove pins from the PWM system. If any pin is not active it errors out
* @author Max Dunne
* @date 2013.08.15
*/
char PWM_RemovePins(unsigned int PWMPins);

/**
* @Function PWM_ListPins(void)
* @param None
* @return Listing of all PWM pins that are active.
* @brief Returns a variable of all active pwm pins. An individual pin can be determined
* if active by "anding" with the pwm_PORTXX Macros
* @author Max Dunne
* @date 2013.08.19
*/
unsigned int PWM_ListPins(void);
char PWM_SetDutyCycle(unsigned char Channel, unsigned int Duty);

/**
 * Function PWM_GetDutyCycle
 *
 * @param Channels, use #defined PWM_PORTxxx
 *
 * @return Duty cycle
 *
 * @remark Gets the Duty Cycle for a Single Channel and returns error if that channel is not enabled
 *
 * @author Max Dunne
 * @date 2011.11.12 */

unsigned int PWM_GetDutyCycle(char Channel);

/**
 * Function: PWM_End
 * @param None
 * @return SUCCESS or ERROR
 *
 * @remark Disables the PWM sub-system and releases all pins.
 *
 * @author Max Dunne
 * @date 2011.11.12 */

char PWM_End(void);

#endif