/**
 * Function: ReadFrontRightBumper
 * @param None
 * @return a 10-bit value corresponding to the amount of light received.
 * @remark Returns the current light level. A higher value means less light is detected.
 * @author Max Dunne
 * @date 2012.01.06 */
unsigned int LightLevel(void);

/**
 * Function: ReadFrontLeftBumper
 * @param None
 * @return BUMPER_TRIPPED or BUMPER_NOT_TRIPPED
 * @remark Returns the state of the front left bumper
 * @author Max Dunne
 * @date 2012.01.06 */
unsigned char ReadFrontLeftBumper(void);

/**
 * Function: ReadFrontRightBumper
 * @param None
 * @return BUMPER_TRIPPED or BUMPER_NOT_TRIPPED
 * @remark Returns the state of the front right bumper
 * @author Max Dunne
 * @date 2012.01.06 */
unsigned char ReadFrontRightBumper(void);

/**
 * Function: ReadRearLeftBumper
 * @param None
 * @return None
 * @remark Returns the state of the rear left bumper
 * @author Max Dunne
 * @date 2012.01.06 */
unsigned char ReadRearLeftBumper(void);

/*
 * File: roach.h
 * Author: mdunne
 * 
 * Created on January 6, 2012, 10:59 AM
 */
#ifndef roach_H
#define roach_H

PUBLIC FUNCTION PROTOTYPES

*******************************************************************************/
#include SUCCESS
#define SUCCESS 0
#define ERROR -1
#endif
#define BUMPER_TRIPPED 1
#define BUMPER_NOT_TRIPPED 0

/**
 * Function: Roach_Init
 * @param None
 * @return None
 * @remark Performs all the initialization necessary for the roach.
 * This includes initializing the PWM module, the A/D converter, the
 * data directions on some pins, and setting the initial motor directions.
 * @author Max Dunne
 * @date 2012.01.06 */
char Roach_Init(void);

/**
 * Function: LeftMtrSpeed
 * @param newSpeed, A value between -10 and 10 which is the new speed
 * of the motor. 0 stops the motor. A negative value is reverse.
 * @return None
 * @remark This function is used to set the speed and direction
 * of the left motor.
 * @author Max Dunne
 * @date 2012.01.06 */
char LeftMtrSpeed(char newSpeed);
@param None
@return BUMPER_TRIPPED or BUMPER_NOT_TRIPPED
@remark Returns the state of the rear left bumper
@author Max Dunne
@date 2012.01.06 */
unsigned char ReadRearLeftBumper(void);

/**
 * Function: ReadRearRightBumper
 * @param None
 * @return BUMPER_TRIPPED or BUMPER_NOT_TRIPPED
 * @remark Returns the state of the rear right bumper
 * @author Max Dunne
 * @date 2012.01.06 */
unsigned char ReadRearRightBumper(void);

/**
 * Function: ReadBumpers
 * @param None
 * @return 4-bit value representing all four bumpers in following order:
 *         front left,front right, rear left, rear right
 * @remark Returns the state of all 4 bumpers
 * @author Max Dunne
 * @date 2012.01.06 */
unsigned char ReadBumpers(void);

#endif