

```

/*****
/* Program      : OPTICSW.H
/* Function     : Optical Switch Control Procedures Header File
/* Author      : John F. Fitter B.E.
/*
/*             Copyright © 1998 Eagle Air Australia Pty. Ltd. All rights reserved
/*****

#ifndef _OPTICSW_H
#define _OPTICSW_H

// Hardware defines
#define LED0PORT    PORTB           // led0 drive port
#define LED0DIR    TRISB           // led0 drive direction direction
#define LED0PIN    6               // led0 drive port pin
#define LED1PORT    PORTB           // led1 drive port
#define LED1DIR    TRISB           // led1 drive direction
#define LED1PIN    7               // led1 drive port pin
#define SENSE0     CCP1            // led0 sense pin
#define SENSE1     CCP2            // led1 sense pin
#define DRVPOSLOG  // define for led drive positive logic
#define SWITCH_0   // switch used for timing measurement

// Macros
#ifdef DRVPOSLOG
#define set_led0(state) led0=state // turn the led on or off
#define set_led1(state) led1=state
#define get_led0() led0           // get led status (on or off)
#define get_led1() led1
#else
#define set_led0(state) led0=!state
#define set_led1(state) led1=!state
#define get_led0() !led0
#define get_led1() !led1
#endif // DRVPOSLOG

// Macro to get the state of the sensor output. The logic is inverted so that a true result
// is returned from the sensor beam being interrupted. This makes more logical sense within
// the context of the program and avoids programming errors.
#define get_sense0() !SENSE0
#define get_sense1() !SENSE1

// Function prototypes
void init_opticsw();
unsigned char get_opticsw(unsigned char);
void set_opticsw();
void reset_opticsw();

// Variables
#ifdef _OPTICSW_C
#else
#endif // _OPTICSW_C

static volatile bit led0 @ (unsigned)&LED0PORT*8+LED0PIN;
static bit led0_dir @ (unsigned)&LED0DIR*8+LED0PIN;
static volatile bit led1 @ (unsigned)&LED1PORT*8+LED1PIN;
static bit led1_dir @ (unsigned)&LED1DIR*8+LED1PIN;

#endif // _OPTICSW_H

// ***** EOF OPTICSW.H *****

```