

```

/*****
/* Program      : BOOM.H
/* Function     : Boomerang Light Controller Header File
/* Author      : John F. Fitter B.E.
/*
/* Copyright © 1998 Eagle Air Australia Pty. Ltd. All rights reserved
*****/

#ifndef _BOOM_H
#define _BOOM_H

#ifdef _DEVELOPMENT

// Logical defines

#define TRUE      1
#define FALSE    0
#define true     TRUE
#define false    FALSE

// Application specific defines

#define SLEEPTIME    30 // Time in seconds to go to sleep

// Constant arrays

// The led_on array is arranged as follows;

// led 1 {time 1 sequence 1, time 2 sequence 1, time 1 sequence 2, time 2 sequence 2, etc.}
// led 2 {time 1 sequence 1, time 2 sequence 1, time 1 sequence 2, time 2 sequence 2, etc.}
// etc.

// The led_off array is structurally the same but specifies led off times.
// The times are multiples of 5mS. Typical boomerang angular speeds are 6 revolutions/S thus
// 5mS represents 11 degrees of arc. There are 33.3 5mS intervals in one revolution.

const unsigned char patt_mask[5] = {1, 2, 4, 0x10, 0x20};
const unsigned char led_on[][20] =
  {{ 1, 16, 4, 1, 2, 1, 2, 1, 20, 10, 1, 1, 1, 1, 2, 1, 1, 16, 2},
   {16, 16, 4, 1, 4, 2, 2, 1, 20, 10, 1, 1, 20, 50, 1, 2, 1, 1, 17, 2},
   { 1, 16, 4, 1, 6, 3, 2, 1, 10, 10, 1, 1, 50, 20, 1, 2, 1, 1, 18, 2},
   {16, 16, 4, 1, 8, 4, 2, 1, 10, 10, 1, 1, 100, 100, 1, 2, 1, 1, 19, 2},
   { 1, 16, 4, 1, 10, 5, 2, 1, 1, 1, 1, 1, 100, 100, 1, 2, 1, 1, 20, 2}};

const unsigned char led_off[][20] =
  {{16, 16, 4, 4, 2, 1, 3, 3, 10, 5, 1, 1, 1, 1, 7, 6, 1, 5, 2, 15},
   {16, 16, 4, 4, 4, 2, 3, 3, 5, 5, 1, 1, 20, 20, 7, 6, 1, 5, 2, 16},
   {16, 16, 4, 4, 4, 2, 3, 3, 10, 5, 1, 1, 20, 20, 7, 6, 1, 5, 2, 17},
   {16, 16, 4, 4, 6, 3, 3, 3, 5, 5, 1, 1, 1, 1, 7, 6, 1, 5, 2, 18},
   {16, 16, 4, 4, 6, 3, 3, 3, 1, 1, 1, 1, 1, 1, 7, 6, 1, 5, 2, 19}};

// Variable declarations

unsigned char output_mask; // output mask
unsigned char output_pattern;
unsigned char count_ls; // 1S counter
unsigned char count_sleep; // sleep time counter
unsigned char led[5]; // array of led timing values
unsigned char index;
unsigned char sequence; // led sequence number
bit buttonup; // up state of user button
bit new_sequence; // flag for new sequence

#endif // _BOOM_H

// ***** EOF BOOM.H *****

```