

```

/*****
/* Program      : MAIN.H
/* Function     : MAIN Header File
/* Author      : John F. Fitter B.E.
/*
/* Application specific stuff removed for web publishing.
/*
/* Copyright © 1998 Eagle Air Australia Pty. Ltd. All rights reserved
*****/

#ifndef _MAIN_H
#define _MAIN_H

/*****
/* Project specific defines
*****/

// #define _DEVELOPMENT // development on emulator - defined in main.cmd
// emulator speed is limited to 8MHz
#define _HVER 1 // hardware version number (0..15)
#define _HREV 0 // hardware revision number (0..15)
#define _SVER 1 // software version number (0..15)
#define _SREV 3 // software revision number (0..15)

/*****
/* Processor speed defines
*****/

#ifdef _DEVELOPMENT // for software development only
#define _CLOCK_ 800000L // development uses 8MHz xtal
#else
#define _CLOCK_ 16000000L // production units use 16MHz xtal
#define _FAST_CLOCK
#endif // _DEVELOPMENT
#define _MCLK _CLOCK_/400000L // cycles per microsecond

/*****
/* Macros
*****/

// Printing macros
#define print_to_printer() c_status.prt_to_lcd = false
#define print_to_lcd() c_status.prt_to_lcd = true

// All this defined for the HiTech compiler only - so this header file
// can be used by Borland C

#ifndef HI_TECH_C

/*****
/* Controller status defines
*****/

struct ctr_status {
    unsigned alternate :1; // interface alternate flag
    unsigned yield :1; // flag to yield control to system functions
    unsigned exp100ms :1; // flag that 100ms has expired
    unsigned sleep_en :1; // enable sleep mode flag
    unsigned awake :1; // controller is awake
    unsigned to_wait :1; // timed out waiting to time
    unsigned to_time :1; // timed out timing
    unsigned to_load :1; // timed out loading (motor load only)
    unsigned mtr_slow :1; // flag that the motor should be running slow
    unsigned prt_to_lcd :1; // print destination flag - printer/lcd
    unsigned serial_exists :1; // serial device is attached
    unsigned char_is_in :1; // serial character has been received
    unsigned ser_rx_err :1; // UART receive error status
    unsigned new_tstate :1; // new controller timing state flag
    unsigned new_istate :1; // new controller interface state flag
    unsigned new_field :1; // new controller interface field flag
    unsigned new_result :1; // new timing result flag
    unsigned en_timing :1; // enable timing flag
    unsigned tare :1; // tare flag - enables measurement
    unsigned accepted :1; // accepted flag - for new values
    unsigned accept_dlg :1; // accept dialog is active flag
    unsigned captured :1; // flag to indicate a time has been captured by CCP
    unsigned validresult :1; // flag to indicate a valid final timing result

```

```

    unsigned printvalid    :1;           // flag to enable printing of valid/invalid string
    unsigned autotimeout   :1;           // flag auto timeout has just occurred
    unsigned allowtimeout  :1;           // flag to allow timing state timeout
    unsigned islogged      :1;           // current measurement is logged
    unsigned gotaresult    :1;           // flag that there is a result - only set, never reset
    unsigned ext_key       :1;           // external simulated keypress (from the PC)
};

union tval_union{
    struct {
        unsigned int lw_tval;
        unsigned int hw_tval;
    } wtval;
    unsigned long ltval;
};

/*****
/* Variable and constant declarations
*****/

#ifdef _MAIN_C
unsigned char lin;
unsigned char pos;
unsigned char csr;
unsigned long unique;           // copy of unique reference number
unsigned char ser_data;        // serial input data

bank1 volatile unsigned char tmr_10ms; // 10ms timer
bank1 volatile unsigned char tmr_100ms; // 100ms timer
bank1 unsigned char sertimeout; // serial timeout (x10ms)
bank1 volatile unsigned char ser_char; // received serial character

bank1 union tval_union tmr_val; // captured value of tmr1 & tmr1_aux
bank1 unsigned int tmr_aux; // timer 1 bits 16 to 31

bank2 unsigned char tmr_1s; // 1 second timer
bank2 unsigned char tmr_100mm; // 6 second timer
bank2 unsigned char sleep_elaps; // sleep elapsed time (x100mm) - milliminutes !!!
bank2 unsigned char motor_time; // motor run timer (x100ms)
bank2 unsigned char motor_dc; // motor on-time at slow speed (ms/50ms)
bank2 unsigned char slow_timer; // timer for motor slow speed
bank2 unsigned char state_time; // state timeout timer (x100ms)
bank2 unsigned char auto_time; // auto-timeout timer (x100ms)
bank2 unsigned char old_sw; // saved optical switch states
bank2 unsigned char xl_super2; // translated superscript 2 for printer

#else
extern unsigned char lin;
extern unsigned char pos;
extern unsigned char csr;
extern unsigned long unique;
extern unsigned char ser_data;

extern bank1 volatile unsigned char tmr_10ms;
extern bank1 volatile unsigned char tmr_100ms;
extern bank1 unsigned char sertimeout;
extern bank1 volatile unsigned char ser_char;

extern bank1 union tval_union tmr_val;
extern bank1 unsigned int tmr_aux;

extern bank2 unsigned char tmr_1s;
extern bank2 unsigned char tmr_100mm;
extern bank2 unsigned char sleep_elaps;
extern bank2 unsigned char motor_time;
extern bank2 unsigned char motor_dc;
extern bank2 unsigned char slow_timer;
extern bank2 unsigned char state_time;
extern bank2 unsigned char auto_time;
extern bank2 unsigned char old_sw;
extern bank2 unsigned char xl_super2;

#endif // _MAIN_C

/*****
// Function prototypes
*****/

```

```
void init_proc();
void goto_sleep(unsigned char);
unsigned char dec_to_bcd(unsigned char);
unsigned char bcd_to_dec(unsigned char);
void put_bcd(unsigned char);
void yield();
void reset_proc(unsigned char);
signed char int_log10(double);
double pow10(signed char);
void start_motor();
```

```
#endif // HI_TECH_C
```

```
#endif // _MAIN_H
```

```
// ***** EOF MAIN.H *****
```