

```

#include "PE_Types.h"
#ifndef DEF_FIOFCNDEF_H
#define DEF_FIOFCNDEF_H

/*=====
    $Author: r w lewis $
=====*/
/*-----FcnDef.h -----*/
//
// Function:          function prototypes
// Notes:
//
//-----*/
#pragma CODE_SEG DEFAULT

void    SystemInterrupt          (uint8  interrupt_state);

void    TimeFcns                 (int8   font_size);

void    TimeMenu                 (struct lcdTimeDateStatic *lcd_time_date_static_ptr,
                                struct lcdTimeDateDyn   *lcd_time_date_dyn_ptr,
                                int8   font_size);

uint8   UpdateWeekDay           (struct TimeDate          *system_time_ptr,
                                struct lcdTimeDateDyn   *lcd_time_date_dyn_ptr,
                                int8   font_size);

uint8   MenuNextSelection       (struct lcdMenuCntrl      *lcd_menu_cntrl_ptrs,
                                uint16  selection_type);

void    ExitOperationPrompt     (int8   font_size);

//-----
void    rtcClk                  (int8   font_size);
void    rtcClockDisp            (int8   font_size);
void    rtcStartClock           (struct TimeDate          *system_time_ptr);

//-----
void    SystemTimeUpdate        (void);
int8    sdRxTx_spi              (int8   write_byte);
int8    rtcRxTx_spi             (int8   write_byte);
void    rtcWriteTime_spi        (struct TimeDate          *system_time_ptr);
void    rtcWriteDate_spi        (struct TimeDate          *system_time_ptr);
uint8   rtcReadCntrlReg_spi     (void);
void    rtcEnable_spi           (void);
void    rtcDisable_spi          (void);
void    rtcResetAlarm_0_spi     (void);
void    rtcReset_spi            (void);
void    rtcReadStatus_spi       (void);
void    rtcWrite_1_Sec_Alarm_spi (void);
sint8   rtcReadByte_spi         (void);
void    rtcReadDateTime_spi     (struct TimeDate *system_time_ptr);

//-----
void    InitSetup                (void);
void    Init_sci                 (void);
void    Init_tmr                 (void);
void    rtcInit_spi              (void);

//-----
void    MessWrite                (int8   font_size);
void    MessRead                 (int8   font_size);

//-----
void    ledIndicator             (uint16  led_state,
                                uint16  led_color,
                                uint8   blink_num);

```

```

void SetOutPut          (uint8 output_mode,
                        uint8 lcd_scroll_cntrl);

void rtcInterruptOn    (void);
void sdWaveOut         (sint8 *SoundFile);
//-----
uint8 BCDDouble2Hex    (uint8 bcd_num);
uint8 Hex2BCDDouble    (uint8 hex_num);
sint16 ValidateTimeDate (struct TimeDate *system_time_ptr);

void Gear_atd         (void);

void itoa              (sint8 *buf,
                        uint16 i,
                        uint16 base);

void Char2AsciiDigits (uint8 char_2_cnv,
                        sint8 *destptr,
                        sint8 cnv_size);
char* xstrcat          (char *des,
                        char *src,
                        ...);
//-----
void swHandlerInterruptDisable (uint8 lft,
                                uint8 rt,
                                uint8 up,
                                uint8 dwn,
                                uint8 sel,
                                uint8 lap,
                                uint8 race_start_end);

void swHandlerInterruptEnable (uint8 lft,
                                uint8 rt,
                                uint8 up,
                                uint8 dwn,
                                uint8 sel,
                                uint8 lap,
                                uint8 race_start_end);

void swPortInterruptDisable (uint8 lft,
                              uint8 rt,
                              uint8 up,
                              uint8 dwn,
                              uint8 sel,
                              uint8 lap,
                              uint8 race_start_end);

void swPortInterruptEnable (uint8 lft,
                              uint8 rt,
                              uint8 up,
                              uint8 dwn,
                              uint8 sel,
                              uint8 lap,
                              uint8 race_start_end);

void EnableSwitchInterrupt (sint8 switch_name);
void DisableSwitchInterrupt (sint8 switch_name);

void swResetFlags        (uint8 lft,
                          uint8 rt,
                          uint8 up,
                          uint8 dwn,
                          uint8 sel,
                          uint8 lap,
                          uint8 race_start_end);

void WaitOnAnySwitch     (void);
void swDebounce          (void);

```

```

//-----
uint8   IncrIndex           (uint8   incr_index,
                             uint8   lowest_value,
                             uint8   highest_value);

uint8   IncrDispOneDigit   (uint8   line_num,
                             uint8   digit_addr,
                             uint8   highest_value,
                             uint8   lowest_value,
                             uint8   digit_value,
                             int8    font_size);

void    IncrDispDigit      (int8    *digit_ptr,           // pointer to hex digit pair
                             uint8   line_addr,
                             uint8   char_addr,
                             uint8   lowest_value,
                             uint8   highest_value,
                             uint8   digit_position,
                             int8    font_size);

void    IncrDispHoursLSDigit (int8    *digit_ptr,           // pointer to hex digit pair
                             uint8   line_addr,
                             uint8   char_addr,
                             uint8   lowest_value,
                             uint8   highest_value,
                             int8    font_size);

void    IncrDispMonth      (struct TimeDate      *system_time_ptr,
                             struct lcdTimeDateDyn *lcd_time_date_dyn_ptr,
                             int8    font_size);

void    IncrDispDateMSDigit (struct TimeDate      *system_time_ptr,
                             uint8   lcd_clk_date_line,
                             uint8   lcd_clk_date_addr,
                             int8    font_size);

void    IncrDispDateLSDigit (struct TimeDate      *system_time_ptr,
                             uint8   lcd_clk_date_line,
                             uint8   lcd_clk_date_addr,
                             int8    font_size);

uint8   IncrDateCheck      (struct TimeDate      *system_time_ptr);

//-----
void    lcdDispUpdate      (uint8   curr_ascii_char,
                             int8    font_size);

void    DispLogo           (void);

void    DispRTClkStatic    (struct lcdTimeDateStatic *lcd_time_date_static_ptr,
                             uint8   time_enabled,
                             uint8   date_enabled,
                             int8   font_size);

void    DispRTClkDyn       (struct TimeDate      *system_time_ptr,
                             struct lcdTimeDateDyn *lcd_time_date_dyn_ptr,
                             uint8   time_enabled,
                             uint8   date_enabled,
                             int8   font_size);

void    DispTimeStringStatic (struct lcdTimeDateStatic *lcd_time_date_static_ptr,
                             int8   font_size);

void    DispTimeStringDyn   (struct TimeDate      *system_time_ptr,
                             struct lcdTimeDateDyn *lcd_time_date_dyn_ptr,
                             int8   font_size);

void    DispDateStringStatic (struct lcdTimeDateStatic *lcd_time_date_static_ptr,

```

```

        int8 font_size);

void DispDateStringDyn      (struct TimeDate      *system_time_ptr,
                             struct lcdTimeDateDyn *lcd_time_date_dyn_ptr,
                             int8 font_size);

void DispTimeHoursDyn      (struct TimeDate      *system_time_ptr,
                             struct lcdTimeDateDyn *lcd_time_date_dyn_ptr,
                             int8 font_size);

void DispTimeMinutesDyn    (struct TimeDate      *system_time_ptr,
                             struct lcdTimeDateDyn *lcd_time_date_dyn_ptr,
                             int8 font_size);

void DispTimeSecondsDyn    (struct TimeDate      *system_time_ptr,
                             struct lcdTimeDateDyn *lcd_time_date_dyn_ptr,
                             int8 font_size);

void DispDateYearDyn       (struct TimeDate      *system_time_ptr,
                             struct lcdTimeDateDyn *lcd_time_date_dyn_ptr,
                             int8 font_size);

void DispDateMonthDyn      (struct TimeDate      *system_time_ptr,
                             struct lcdTimeDateDyn *lcd_time_date_dyn_ptr,
                             uint8 cursor_status,
                             int8 font_size);

void DispDateWeekDayDyn    (struct TimeDate      *system_time_ptr,
                             struct lcdTimeDateDyn *lcd_time_date_dyn_ptr,
                             int8 font_size);

void DispDateDateDyn       (struct TimeDate      *system_time_ptr,
                             struct lcdTimeDateDyn *lcd_time_date_dyn_ptr,
                             int8 font_size);

void DispTimeStatic        (struct lcdTimeDateStatic *lcd_time_date_static_ptr,
                             int8 font_size);

void DispDateStatic        (struct lcdTimeDateStatic *lcd_time_date_static_ptr,
                             int8 font_size);
//-----
void DispPromptExit        (uint8 lft,
                             uint8 rt,
                             uint8 up,
                             uint8 dwn,
                             uint8 sel,
                             uint8 lap,
                             uint8 race_start_end,
                             int8 font_size);
//-----
void DispChar              (uint8 disp_char);

void DispCursorOff        (void);
void DispCursor           (uint8 lcd_line_num,
                             uint8 lcd_char_addr,
                             uint8 cursor_cntrl,
                             int8 font_size,
                             uint8 string_length);

void DispString            (uint8 lcd_line_num,
                             uint8 lcd_char_addr,
                             int8 *buffer,
                             uint8 clear_line,
                             uint8 cursor_cntrl,
                             uint8 cursor_line_num,
                             uint8 cursor_addr,
                             int8 font_size);

void DispStringNoCursor   (uint8 lcd_line_num,

```

```

        uint8 lcd_char_addr,
        int8 *buffer,
        uint8 clear_line,
        int8 font_size);

void grphClrLine (uint8 lcd_line_num,
                int8 font_size);

void DispBuffLine (uint8 lcd_line_num,
                  uint8 lcd_char_addr,
                  int8 *buffer,
                  int8 font_size);

void DispAndIncrAddr (uint8 curr_ascii_char,
                     uint8 scroll_mode,
                     int8 font_size,
                     uint16 pause_time,
                     uint16 time_delay_EOL,
                     uint8 new_page);

uint8 DispPrompt (uint8 lft,
                  uint8 rt,
                  uint8 up,
                  uint8 dwn,
                  uint8 sel,
                  uint8 lap,
                  uint8 race_start_end,
                  uint8 line_addr,
                  uint8 char_addr,
                  uint8 ascii_char1,
                  uint8 ascii_char2,
                  int8 *current_char_ptr,
                  uint16 bypass_prompt,
                  int8 font_size);

void DispSetCharAddr (uint8 lcd_line_num,
                     uint8 lcd_char_addr);

void DispPrintf (void);

uint8 lcdIncrLineAddr (uint8 scroll_mode);

//-----
sint8 lcdBusyCheck (void);
void lcdWriteCmdWithBusyCheck (uint8 lcd_cmd);
void lcdWriteCmd (uint8 lcd_cmd);
void lcdWriteCmdWithDelay (uint8 lcd_cmd,
                           uint8 cmd_delay);

//-----
void TimeDelay (uint16 timer_delay);
void TimerDelay (void);
void TimeDelayScript (uint16 time_delay_EOL);

//-----
void grphClrView (void);
void grphOutline (void);

void grphUpdate (uint8 top,
                uint8 bottom);

void grphSetFont (int8 font_size);
void grphSaveFont (void);
void grphRestoreFont (void);

//-----
void Print_iMn (uint8 *print_char);

#endif

```