EAGLE SYSTEMS
Microprocessor Based Designs

10908 33rd Street
Santa Fe, Texas 77510
Phone: (409) 925-2753

1.0 Introduction

Daylight Saving Time begins for most of the United States at 2 a.m. on the first Sunday of April. Time reverts to standard time at 2 a.m. on the last Sunday of October. (Interesting tidbit: Note that it is Daylight Saving (singular) Time, NOT Daylight Saving**S** Time. We are saving daylight, so it is singular and not plural.)

There are many great articles available on the web that describes the history and purpose of Daylight Saving Time (which is beyond the scope of this article).

2.0 DST Algorithm

The following algorithm assumes the computer system has access to date/time information. To adjust any clock for Daylight Saving Time, only one reliable piece of information is needed: *What day of the week is it?*

There are two ways to gather the day-of-week information:

- Available as part of the date/time information from a real time clock (*of course, in this case, you will not need to calculate the day-of-week*).
- Calculated based on the current month/day/year information.

2.1 Day-Of-Week Calculation

To calculate the day-of-week, use the following formula (integer math only):

a = (14-month)/12
y = year – a
m = (month + (12 * a)) – 2
d = (day + y + (y / 4) – (y / 100) + (y / 400) + ((31 * m) / 12)) mod 7

The result of "d" is 0 for Sunday, 1 for Monday, 2 for Tuesday, etc.. To avoid confusion, Sunday will be changed from 0 to 7 using a software algorithm shown in the C code example.

Remember: "mod 7" simply means to take the remainder, as in %7 written in C.

2.2 Day-Of-Week Example: December 05, 2002

a = (14-12)/12 = *0 (integer math)*
y = 2002 – 0 = *2002*
m = (12 + (12 * 0)) – 2 = *10*
d = (5 + 2002 + (2002 / 4) – (2002 / 100) + (2002 / 400) + ((31 * 10) / 12)) mod 7
d = (2507) – (20) + (5) + (25) mod 7
d = 2517 mod 7 (*2517 / 7 = 359 with a remainder of 4*)
d = 4 (*Thursday*)

December 5, 2002 falls on a Thursday!

2.3 Time/Date Websites:

http://www.timeanddate.com/time/aboutdst.html - Daylight Saving Time history (nice article!)

http://www.timeanddate.com - Will generate calendars for any year to double check your day-of-week calculation.

10908 33rd Street
Santa Fe, Texas 77510
Phone: (409) 925-2753

3.0 C Program Example

There are 3 functions listed in the following code example:

calc_day_of_week() = Used to calculate the day-of-week based on the current day-month-year.

dst_check() = Used to detect daylight saving time change (2:00am, first Sunday in April, and 2:00am last Sunday in October. Flags will be set/cleared to indicate an hour add/subtract from the current time.

dst_clock_adjust() = Used to adjust the system clock back/ahead one hour based on the flags set/cleared in the dst_check() function.

The function example expects to find the system date/time located in the following structure variables:

Time.hr_counter	Current Hour military time
Time.day	Current day of month
Time.day_of_week	Current day of week (1-7, 1= Monday 7=Sunday)
Time.month	Current month (1-12)
Time.year	Current year (00-99) Must add 2000 to year if your clock does not support 4 digit years.

The C code examples were copied from actual project files that use the Dallas 1644 Real time clock. The code examples can be easily adapted to other types of real time clocks.

10908 33rd Street
Santa Fe, Texas 77510
Phone: (409) 925-2753

3.1 calc_day_of_week()

```
/*>>>>>>>>>>>>>>>>>>>>>>>>>> FUNCTION: calc_day_of_week() <<<<<<<<<<<<<<<<<<<<<<    */
/* Purpose: Calculate Day-Of-Week based on current date.                            */
/*                                                                                  */
/*   Notes:  The Dallas DS1644 Clock/Calendar will not initialize the day of        */
/*           the week based on the current date. It will allow you to initialize    */
/*           it to a value of 1-7. The clock will then increment the day of week    */
/*           on each transition across midnight. We must initialize the day of      */
/*           week on each reboot of the SA-CPU, or when ever there is a             */
/*           date/time change by the user.                                          */
/*                                                                                  */
/*           Entry occurs once/second or upon each user update of the date.         */
/*                                                                                  */
/* Programmer:  Rick Vaughn (Eagle Systems, Santa Fe, Texas 77510)                  */
/*                                                                                  */
/* Revisions: 4/16/02 (Orig)  ---Rick Vaughn---                                     */
/*                                                                                  */
/*>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>    */

void calc_day_of_week(void)
       {
       uint calc_a,calc_y,calc_m,calc_d;

       // Formula to calculate the day-of-week:
       //
       // a= (14-month)/12
       // y= year - a
       // m= (month+(12*a))-2
       // d= (day+y+(y/4)-(y/100)+(y/400)+((31*m)/12)) mod 7
       //
       // Remember: "mod 7" simply means to take the remainder, as in %7 in C.
       //
       // The following variables are assigned:
       //
       //     time.hr_counter              Current Hour
       //     time.day              Current Day
       //     time.day_of_week      Current day of the week (1-7, 1=Monday)
       //     time.month            Current month (1-12)
       //     time.year             Current year (00-99) Must add 2000

       calc_a = (14-time.month) / 12;
       calc_y = (time.year + 2000) - calc_a;
       calc_m = (time.month+(12 * calc_a)) - 2;
       calc_d = (time.day+calc_y+(calc_y/4)-(calc_y/100)+(calc_y/400)+((31*calc_m)/12)) % 7;

       // The day of week now resides in the calc_d variable. Remember, the day-of-week
       // calculated is in the range 0-6, with Sunday = 0. We want Sunday = 7.

       if(calc_d)
              {
              // Non-zero - store day-of-week 1-6
              time.day_of_week = (uchar)calc_d;
              }
       else
              {
              // Equal to zero - this is Sunday - reset to 7
              time.day_of_week = 7;
              }

       }
```

3.2 dst_check()

```
/*>>>>>>>>>>>>>>>>>>>>>>>>>>> FUNCTION: dst_check() <<<<<<<<<<<<<<<<<<<<<<<<<<<<    */
/* Purpose: Check for Daylight Saving Time.                                         */
/*                                                                                  */
/*                                                                                  */
/*    Notes:  To calculate daylight saving time, it will always begin the first     */
/*            Sunday in April, and end the last Sunday in October. We can check     */
/*            to see if we have the first Sunday by testing for the first           */
/*            occurance of a "7" day-of-week.                                       */
/*            The time will change at 2:00am. A flag will be set to indicate the    */
/*            time is in DST and will need to be incremented in October.            */
/*                                                                                  */
/*            If enabled, the daylight saving change will only occur if the         */
/*            system is powered up and actually "sees" the transition to 2:00am     */
/*            on the first Sunday in April and the last Sunday in October.          */
/*                                                                                  */
/*            If the user changes the date to jump into or out-of daylight          */
/*            saving time, the system will not automatically adjust the time.       */
/*            If a manual change to the date is made, the system will assume a      */
/*            manual change to the time is required also.                           */
/*                                                                                  */
/*                                                                                  */
/* Programmer:  Rick Vaughn (Eagle Systems, Santa Fe, Texas 77510)                  */
/*                                                                                  */
/* Revisions: 4/16/02 (Orig)  ---Rick Vaughn---                                     */
/*                                                                                  */
/*>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<    */

void dst_check(void)
        {

        // Flags tested:
        //
        // dst_auto_adjust_flag: Set if we need to automatically adjust for daylight saving.

        if(flags.dst_auto_adjust_flag)
                {
                // Daylight Saving Time auto adjust is enabled - Check to see if we need
                // to adjust the time. First Sunday in April, and Last Sunday in October.
                //
                //*** APRIL - SPRING FORWARD - ADD 1 HOUR ****
                if(time.month == 4)
                        {
                        // We are in April - see if this is the first Sunday
                        if(time.day <= 7)
                                {
                                // We are in the first week of April - see which day is equal
                                // to the 7th day of the week (Sunday)

                                if(time.day_of_week == 7)
                                        {
                                        // Its April, we are currently in the first Sunday, see if
                                        // its 2:00am - Adjust the time forward when
                                        // 2:00am arrives.
                                        if((time.hr_counter == 2) && (!flags.dst_adjusted))
                                                {
                                                // Its 2:00am - increment the hour and set the
                                                // dst_active_flag
                                                // Call the dst_clock_adjust() function to add one
                                                // hour.
                                                dst_clock_adjust(1);
                                                flags.dst_adjusted=TRUE;  // Only increment one time
                                                }
                                        }
                                }
                        }// end April check
```

EAGLE SYSTEMS
Microprocessor Based Designs

10908 33rd Street
Santa Fe, Texas 77510
Phone: (409) 925-2753

3.2 dst_check() cont…

```
        //*** OCTOBER - FALL BACK - SUBTRACT 1 HOUR ****
        if(time.month == 10)
                {
                // We are in October - see if this is the last Sunday
                if(time.day >= 25)
                        {
                        // We are in the last week of October - see which day is equal to
                        // the last Sunday.
                        if(time.day_of_week == 7)
                                {
                                // Its October, we are currently in the last Sunday, see if
                                // its 2:00am - Adjust the time backwards when 2:00am
                                // arrives.
                                if((time.hr_counter == 2) && (!flags.dst_adjusted))
                                        {
                                        // Its 2:00am - subtract an hour and set the
                                        // dst_active_flag
                                        // Call the dst_clock_adjust() function to subtract
                                        // one hour.
                                        dst_clock_adjust(0);
                                        flags.dst_adjusted=TRUE;  // Only increment one time
                                        }
                                }
                        }
                }// end October check


        }

//** Clear the flag.dst_adjusted flag if the hour is NOT equal to 1am or 2am (remember,
//   in October the clock is adjusted back an hour to 1:00am - we don't want to adjust
//   again when 2:00am is reached for the second time.

if((time.hr_counter < 1) || (time.hr_counter > 2))
        {
        // It's not 1:00am or 2:00am - clear the flag
        flags.dst_adjusted = FALSE;
        }

}
```

3.3 dst_clock_adjust()

```
/*>>>>>>>>>>>>>>>>>>>>>>>>>>> FUNCTION: dst_clock_adjust() <<<<<<<<<<<<<<<<<<<<    */
/* Purpose: Adjust the clock back/ahead an hour.                                  */
/*                                                                                */
/*                                                                                */
/*   Notes:  This routine is used to add or subtract an hour from the system      */
/*           clock (used for daylight saving time).  A "direction flag" will      */
/*           be passed to this routine to indicate an addition or subtraction     */
/*           to the current system time.                                          */
/*                                                                                */
/*                direction_flag: 0= subtract, 1 = add.                           */
/*                                                                                */
/*           This routine can be called anytime the user resets the date.         */
/*           (it may not always change the time at 2:00am - if the user changes   */
/*           the date to jump inside/outside of daylight saving time, the clock   */
/*           should automatically adjust (if dst is enabled)                      */
/*                                                                                */
/*                                                                                */
/*                                                                                */
/* Programmer:  Rick Vaughn (Eagle Systems, Santa Fe, Texas 77510)                */
/*                                                                                */
/* Revisions: 4/16/02 (Orig)  ---Rick Vaughn---                                   */
/*                                                                                */
/*>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  */

void dst_clock_adjust(uchar direction_flag)
	{
	// Direction flag: 0=subtract, 1=add
	// Get the current date/time

	read_realtime_clock();
	time.temp_minute = time.min_counter;
	time.temp_hour = time.hr_counter;
	time.temp_second = time.sec_counter;

	/* Store the current date */
	time.temp_day = time.day;
	time.temp_month = time.month;
	time.temp_year = time.year;

	if(direction_flag)
		{
		// Non-Zero - must be increment
		// Make sure we will not cross the 23:00 to 0:00 boundary
		if(time.temp_hour == 23)
			{
			// Set for 0:00
			time.temp_hour = 0;
			}
		else
			{
			// Normal increment
			time.temp_hour++;
			}

		}
	else
		{
		// direction flag = 0, subtract an hour from the current time
		// Make sure we will not cross the 0:00 to 23:00 boundry
		if(time.temp_hour == 0)
			{
			// Set for 23:00
			time.temp_hour = 23;
			}
		else
```

EAGLE SYSTEMS
Microprocessor Based Designs

10908 33rd Street
Santa Fe, Texas 77510
Phone: (409) 925-2753

3.3 dst_clock_adjust() cont…

```
                {
                // Normal decrement
                time.temp_hour--;
                }
        }

    set_realtime_clock();
    read_realtime_clock();
    }
```