TimeLord Arduino Library

Please direct any questions or suggestions to support (at) swfltek.com

The library may be downloaded here.

This library provides some useful date and time functions. In order to be more compatible with typical Real Time Clock chips, this library contains date-time values in a 6 byte array, rather than using the traditional unix timestamp. The library defines labels for the array elements to help keep your code easy to read.

offset	Value	Label	Valid range
0	Seconds	tl_second	0-59
1	Minute	tl_minute	0-59
2	Hour	tl_hour	0-23
3	Day	tl_day	1-31
4	Month	tl_month	1-12
5	Year	tl_year	0-99

This library was designed for use in micro-controller applications. As such, various compromises are made to minimise memory use and processor load. A notable compromise is the two digit year. This library assumes years are between 2000 and 2099... you've been warned.
Other function limitations and notes are listed in italic after the function description.

Declaration

To use the library in your sketch, declare a TimeLord object...

```
TimeLord myLord;
```

You can declare multiple TimeLord objects, to work in different timezones for instance...

```
TimeLord london;
TimeLord paris;
```

Configuration

boolean TimeLord::Position(float latitude, float longitude);

Some functions require the geographic position be specified.
Latitude is positive for North, negative for South.
Longitude is positive for East, negative for West.
Returns false if the passed values make no sense.
Example:

```
TimeLord myLord;
myLord.Position(26.9, -81.8) // set position to SouthWest Florida
```

boolean TimeLord::TimeZone(int zone);

Some functions require the time zone be specified.
The passed value is in minutes before(positive) GMT, or after(negative).
Returns false if the zone value makes no sense.
Example:

```
TimeLord myLord;
myLord.TimeZone(-5 * 60); // set zone to USA Eastern
```

boolean DstRules(byte start_month, byte start_week, byte end_month, byte end_week, byte advance);

Daylight Savings begins and ends on certain Sundays, in certain months.
Use this call to set the rules for your locality.
The value of advance is in minutes.
This function returns false if the passed values make no sense.
Example for the USA...

```
TimeLord myLord;
myLord.DstRules(3,2,11,1, 60); // second sunday in march thru first sunday in november
```

Use

void TimeLord::GMT(byte date_time[6]);

Convert the passed date-time value to Greenwich Mean Time.

The date-time value passed is expected to be local standard time.
This function requires TimeZone() to be set.
Example:

```
TimeLord easternUsa.;
easternUsa.TimeZone(-5 * 60); // set zone to USA Eastern
byte the_time = {0, 0, 12, 1, 7, 9); // set the time to noon on July 1, 2009
easternUsa.GMT(the_time);
```

`void TimeLord::DST(byte date_time[6]);`

This subroutine converts the passed date-time value to Daylight Savings Time.
The date-time value passed is expected to be local standard time.
This function requires that DstRules() be set.
Example:

```
TimeLord easternUsa.;
easternUsa.DstRules(3,2,11,1, 60);
byte easternDST = {0, 0, 12, 1, 7, 9); // set the time to noon on July 1, 2009
easternUsa.DST(easternDST);
```

`boolean TimeLord::SunRise(byte date_time[6]);`

This function converts the passed date-time to the time of Sunrise on the given date.
The function returns false if the sun does not rise that day, which can happen in the Arctic or Antarctic.
This function requires that Position() be set.
Example: *`// what time does the sun rise on Xmas day`* *of 2009, in Southwest Florida?*

```
TimeLord myLord;
myLord.Position(26.9, -81.8) // set position to SouthWest Florida
byte sunRise  = {0, 0, 0, 25, 12, 9);
myLord.SunRise(sunRise);
```

This function is only accurate to within 5 minutes

boolean TimeLord::SunSet(byte date_time[6]);

Just like SunRise, except it computes the time of sun set.

void TimeLord::Sidereal(byte date_time[6], boolean local);

This subroutine converts the passed date-time into Mean Sidereal time.
If 'local' is false, Greenwich Mean Sidereal Time is computed.
If local is true, Local Mean Sidereal Time is computed. LMST requires that Position() be set.

Example:

```
TimeLord myLord;
myLord.Position(26.9, -81.8) // set position to SouthWest Florida
byte sidereal = {0, 0, 0, 1, 7, 9); // set the time to midnight on July 1, 2009
myLord.Sidereal(sidereal);
```

This function is only accurate to within +- 2 seconds GMST, which, for you astronomers, is about 30 arc-seconds.
LMST may be less accurate, depending on the accuracy of Position().

float TimeLord::MoonPhase(byte date_time[6]);

This subroutine returns the phase of the moon on the specified date.
The returned value ranges from 0 to (almost) 1
0.0 New moon
0.25 First Quarter
0.5 Full Moon
0.75 Last Quarter
0.99 Almost new
Example:

```
TimeLord myLord;
byte the_time = {0, 0, 0, 1, 7, 9); // set the time to midnight on July
```

```
1, 2009

float phase;
phase=myLord.MoonPhase(the_time);
```

This function is only accurate to 1 day

byte TimeLord::Season(byte date_time[6]);

Returns the season corresponding to the passed date.
0 = Winter
1 = Spring
2 = Summer
3 = Fall
Example

```
TimeLord myLord;
byte the_time = {0, 0, 0, 1, 7, 9); // set the time to midnight on July
1, 2009

byte season;
season=myLord.Season(the_time);
```

This function is only accurate to one day, and assumes the Northern Hemisphere. In the Southern Hemisphere, use

```
season = (season + 2) % 4
```

byte TimeLord::DayOfWeek(byte date_time[6]);

Returns the day of the week corresponding to the passed date, Sunday=1, Saturday=7.
Example:

```
TimeLord myLord;
byte the_time = {0, 0, 0, 1, 7, 9); // set the time to midnight on July
1, 2009
if (myLord.DayOfWeek(the_time) == 6) Serial.println("Thank God it's
Friday!");
```

byte TimeLord::LengthOfMonth(byte date_time[6]);

Returns the length of the month in the passed date.
Example:

```
TimeLord myLord;
byte the_time = {0, 0, 0, 1, 7, 9); // midnight on July 1, 2009
byte month_length;
month_length=myLord.LengthOfMonth(the_time);
```

boolean TimeLord::IsLeapYear(int year);

Returns true if the passed year is a leap year.
Example:

```
TimeLord myLord;
if (myLord.IsLeapYear(2012)) Serial.println("2012 is a leap year.");
```

Unlike most other functions where the year is limited to 0-99, IsLeapYear() works with with a 4 digit value.