Readme file for Arduino Time Library

Time is a library that provides timekeeping functionality for Arduino.

The code is derived from the Playground DateTime library but is updated

to provide an API that is more flexable and easier to use.

A primary goal was to enable date and time functionality that can be used with

a variety of external time sources with minimum differences required in sketch logic.

Example sketches illustrate how similar sketch code can be used with: a Real Time Clock,

internet NTP time service, GPS time data, and Serial time messages from a computer

for time synchronization.

The functions available in the library include:

hour(); // the hour now (0-23)

minute(); // the minute now (0-59)

second(); // the second now (0-59)

day(); // the day now (1-31)

weekday(); // day of the week, Sunday is day 0

month(); // the month now (1-12)

year(); // the full four digit year: (2009, 2010 etc)

there are also functions to return the hour in 12 hour format

hourFormat12(); // the hour now in 12 hour format

isAM(); // returns true if time now is AM

isPM(); // returns true if time now is PM

now(); // returns the current time as seconds since Jan 1 1970

The time and date functions can take an optional parameter for the time. This prevents

errors if the time rolls over between elements. For example, if a new minute begins

between getting the minute and second, the values will be inconsistent. Using the

following functions eliminates this probglem

time\_t t = now(); // store the current time in time variable t

hour(t); // returns the hour for the given time t

minute(t); // returns the minute for the given time t

second(t); // returns the second for the given time t

day(t); // the day for the given time t

weekday(t); // day of the week for the given time t

month(t); // the month for the given time t

year(t); // the year for the given time t

Functions for managing the timer services are:

setTime(t); // set the system time to the give time t

setTime(hr,min,sec,day,mnth,yr); // alternative to above, yr is 2 or 4 digit yr (2010 or 10 sets year to 2010)

adjustTime(adjustment); // adjust system time by adding the adjustment value

timeStatus(); // indicates if time has been set and recently synchronized

// returns one of the following enumerations:

timeNotSet // the time has never been set, the clock started at Jan 1 1970

timeNeedsSync // the time had been set but a sync attempt did not succeed

timeSet // the time is set and is synced

Time and Date values are not valid if the status is timeNotSet. Otherwise values can be used but

the returned time may have drifted if the status is timeNeedsSync.

setSyncProvider(getTimeFunction); // set the external time provider

setSyncInterval(interval); // set the number of seconds between re-sync

There are many convenience macros in the time.h file for time constants and conversion of time units.

To use the library, copy the download to the Library directory.

The Time directory contains the Time library and some example sketches

illustrating how the library can be used with various time sources:

- TimeSerial.pde shows Arduino as a clock without external hardware.

It is synchronized by time messages sent over the serial port.

A companion Processing sketch will automatically provide these messages

if it is running and connected to the Arduino serial port.

- TimeSerialDateStrings.pde adds day and month name strings to the sketch above

Short (3 character) and long strings are available to print the days of

the week and names of the months.

- TimeRTC uses a DS1307 real time clock to provide time synchronization.

A basic RTC library named DS1307RTC is included in the download.

To run this sketch the DS1307RTC library must be installed.

- TimeRTCSet is similar to the above and adds the ability to set the Real Time Clock

- TimeRTCLog demonstrates how to calculate the difference between times.

It is a vary simple logger application that monitors events on digtial pins

and prints (to the serial port) the time of an event and the time period since the previous event.

- TimeNTP uses the Arduino Ethernet shield to access time using the internet NTP time service.

The NTP protocol uses UDP and the UdpBytewise library is required, see:

http://bitbucket.org/bjoern/arduino\_osc/src/14667490521f/libraries/Ethernet/

- TimeGPS gets time from a GPS

This requires the TinyGPS library from Mikal Hart:

http://arduiniana.org/libraries/TinyGPS

Differences between this code and the playground DateTime library

although the Time library is based on the DateTime codebase, the API has changed.

Changes in the Time library API:

- time elements are functions returning int (they are variables in DateTime)

- Years start from 1970

- days of the week and months start from 1 (they start from 0 in DateTime)

- DateStrings do not require a seperate library

- time elements can be accessed non-atomically (in DateTime they are always atomic)

- function added to automatically sync time with extrnal source

- localTime and maketime parameters changed, localTime renamed to breakTime

Technical notes:

Internal system time is based on the standard Unix time\_t.

The value is the number of seconds since Jan 1 1970.

System time begins at zero when the sketch starts.

The internal time can be automatically synchronized at regular intervals to an external time source.

This is enabled by calling the setSyncProvider(provider) function - the provider argument is

the address of a function that returns the current time as a time\_t.

See the sketches in the examples directory for usage.

The default interval for re-syncing the time is 5 minutes but can be changed by calling the

setSyncInterval( interval) method to set the number of seconds between re-sync attempts.

The Time library defines a structure for holding time elements that is a compact version of the C tm structure.

All the members of the Arduino tm structure are bytes and the year is offset from 1970.

Convenience macros provide conversion to and from the Arduino format.

Low level functions to convert between system time and individual time elements are provided:

breakTime( time, &tm); // break time\_t into elements stored in tm struct

makeTime( &tm); // return time\_t from elements stored in tm struct

The DS1307RTC library included in the download provides an example of how a time provider

can use the low level functions to interface with the Time library.