Arduino Time library

The Time library adds timekeeping functionality to Arduino with or without external timekeeping hardware. It allows a ske to get the time and date as: second, minute, hour, day, month and year. It also provides time as a standard C time_t so elapsed times can be easily calculated and time values shared across different platforms.

> Update: newer versions of Time, TimeAlarms, and DS1307RTC are available, featuring more updates for newer versio Arduino and compatibility with Arduino Due.

The code is derived from the earlier Playground DateTime library but is updated to provide an API that is more flexible an easier to use.

> The download includes example sketches illustrating how similar sketch code can be used with: a Real Time Clock,Int NTP time service, GPS time data, DCF77 radio signal, and Serial time messages from a computer. To use all of the features in the library, you'll need the UDPbitewise library, found here.

Additional documentation is included in a readme.txt file

Functional Overview

```
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 1

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 betwe elements. For example, if a new minute begins between getting the minute and second, the values will be inconsistent. Us

```
                // 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,month,yr); // Another way to set
                                  // the time
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.

Using the Library Copy the download to the Library directory. The Time directory contains the Time library and some examp 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.

- TimeRTC uses a DS1307 Real Time Clock (or DS3231 ChronoDot RTC*) 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. * The I2C 'DS3231' interface is very straight forward and virtually ider

- 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/146674905211/libraries/Ethernet/ Note: This is out of date. For Arduino 0022, see the bui

-TimeGPS gets time from a GPS

This requires the TinyGPS and NewSoftSerial libraries from Mikal Hart: http://arduiniana.org/libraries/TinyGPS and http://arduiniana.org/libraries/newsoftserial/

Example sketch:

The test sketch uses a message on the serial port to set the time. A Processing sketch that sends these messsages is inclu in the download but you can test this sketch by sending T1262347200 using the serial monitor (this sets the time to noon c Jan 1 2010). On a unix system, you can set the time with the shell command:

```
TZ_adjust=-8;  echo T$(($(date +%s)+60*60*$TZ_adjust)) > /dev/tty.usbserial-A8008pym
```

[Get

-- edit: The above code did not work for me in my linux terminal, I wrote the below script which worked. (I was using zsh and UnoR3)

```
TZ_adjust=5.5;d=$(date +%s);t=$(echo "60*60*$TZ_adjust/1" | bc);echo T$(echo $d+$t |bc ) > /dev/ttyACM0
```

[Get

Adjust the TZ_adjust value to your timezone and the serial port to your Arduino.

```
#include <Time.h>

#define TIME_MSG_LEN  11   // time sync to PC is HEADER followed by Unix time_t as ten ASCII digits
#define TIME_HEADER  'T'   // Header tag for serial time sync message
#define TIME_REQUEST  7    // ASCII bell character requests a time sync message

// T1262347200  //noon Jan 1 2010

void setup()  {
  Serial.begin(9600);
}

void loop(){
  if(Serial.available() )
```

```
  if(timeStatus() == timeNotSet)
    Serial.println("waiting for sync message");
  else
     digitalClockDisplay();
  delay(1000);
}

void digitalClockDisplay(){
  // digital clock display of the time
  Serial.print(hour());
  printDigits(minute());
  printDigits(second());
  Serial.print(" ");
  Serial.print(day());
  Serial.print(" ");
  Serial.print(month());
  Serial.print(" ");
  Serial.print(year());
  Serial.println();
}

void printDigits(int digits){
  // utility function for digital clock display: prints preceding colon and leading 0
  Serial.print(":");
  if(digits < 10)
    Serial.print('0');
  Serial.print(digits);
}

void processSyncMessage() {
  // if time sync available from serial port, update time and return true
  while(Serial.available() >=  TIME_MSG_LEN ){  // time message consists of header & 10 ASCII digits
    char c = Serial.read() ;
    Serial.print(c);
    if( c == TIME_HEADER ) {
      time_t pctime = 0;
      for(int i=0; i < TIME_MSG_LEN -1; i++){
        c = Serial.read();
        if( c >= '0' && c <= '9'){
          pctime = (10 * pctime) + (c - '0') ; // convert digits to a number
        }
      }
      setTime(pctime);   // Sync Arduino clock to the time received on the serial port
    }
  }
}
```

code. This sketch gets time from an Internet time provider (NTP) using the Arduino Ethernet shield. Note that the loop coc does not require any logic to maintain time sync. The Time library will automatically monitor NTP and sync the time as necessary.

```
void setup()
{
  Serial.begin(9600);
  Ethernet.begin(mac,ip,gateway);
  Serial.println("waiting for sync");
  setSyncProvider(getNtpTime);
  while(timeStatus()== timeNotSet)
     ; // wait until the time is set by the sync provider
}

void loop()
{
  if( now() != prevDisplay) //update the display only if the time has changed
  {
    prevDisplay = now();
    digitalClockDisplay();
  }
}

void digitalClockDisplay(){
  // digital clock display of the time
  Serial.print(hour());
  printDigits(minute());
  printDigits(second());
  Serial.print(" ");
  Serial.print(day());
  Serial.print(" ");
  Serial.print(month());
  Serial.print(" ");
  Serial.print(year());
  Serial.println();
}

void printDigits(int digits){
  // utility function for digital clock display: prints preceding colon and leading 0
  Serial.print(":");
  if(digits < 10)
    Serial.print('0');
  Serial.print(digits);
}

/*-------- NTP code ----------*/
```

Errata

The macro to convert days to time_t was corrected as of July 22 2011