



## The Tuxgraphics AVR NTP clock



### *Abstract:*

The Network Time Protocol (NTP) has revolutionized the world. Suddenly one could have anywhere in the world accurate time and date. NTP is a simple UDP based protocol and can be implemented in a Microcontroller.

Using the tuxgraphics ethernet board and a LCD display we build a nice clock which gets time and date via NTP. Just attach it to you DSL router!

---

## A NTP client

So far we have only implemented UDP and TCP servers. A clock needs however to be a client. This is something new. A server just answers to incoming IP packets it is therefore a bit easier to

implement. However NTP is a short packet and it is therefore possible to add a NTP client to the existing network stack. We can therefore build a clock which has not only a LCD display but also a build-in webserver. This can then be used to configure the clock or see the current time.

The NTP protocol is described in [RFC958](#). Essentially it is just a 64 bit time-stamp. 32 bits of this time-stamp are the seconds in UTC (=GMT Greenwich mean time) since since Jan. 1st 1900. The other 32 bits are fractions of a second. In other words NTP can be very very accurate. For our purposes it is however enough if we just evaluate the seconds.

The AVR NTP clock synchronizes at startup with a NTP server and uses then a timer interrupt to maintain time locally. Every hour it tries then to synchronize again. If your DSL router is however off during the night then it's not a problem. The clock just continues. We use the on board crystal to maintain the clock locally. This will minimize the drift even if no Internet connection was possible for a couple of days.

## How to build a clock

A clock is essentially just a counter. Since NTP is already a 32-bit "counter" we just take a 32bit variable and increment it. That is: the initial setting of the counter comes via NTP and then we just count up every second.

For this we just generate a timer interrupt every second. The 16bit timer/counter of the atmega168 supports this already on hardware level.

The basic clock is therefore just this:

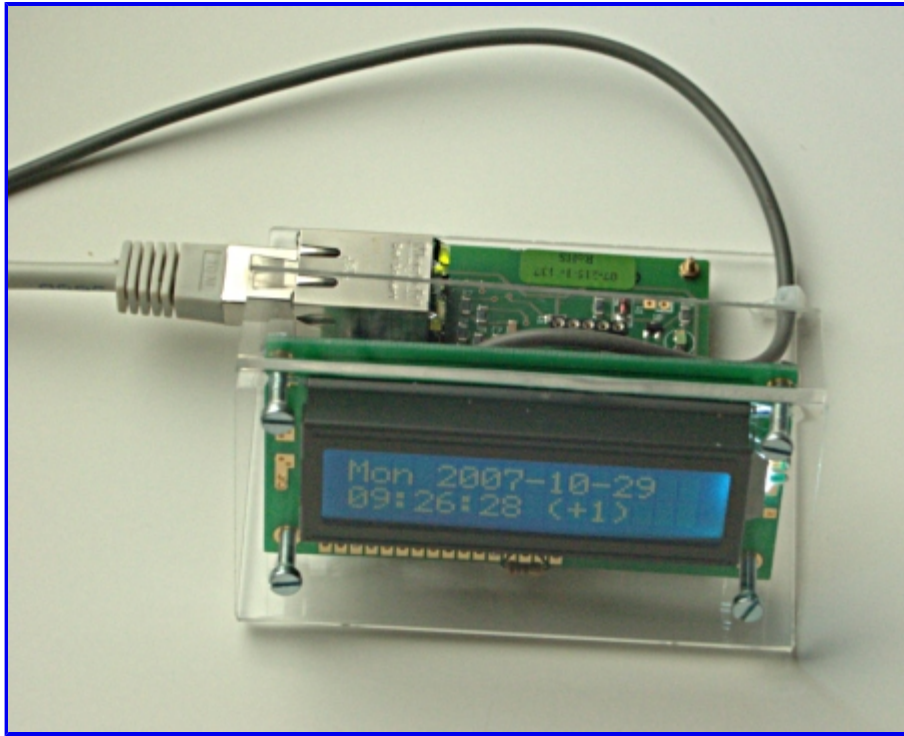
```
// interrupt, step seconds counter
ISR(TIMER1_COMPA_vect){
    time++;
}
```

The hardware interrupt is generated every second and we step the counter "time". Very easy.

If you had already a look at the code you might have noticed that the included README file says that one needs an atmega168 for this and a atmega88 not sufficient. Why should such a simple counter not fit into an atmega88 chip??.

The problem is the math to convert seconds since day-X into a human readable format. This requires quite heavy math for a microcontroller. AVR is a 8-bit processor therefore 32-bit math is expensive. Add the NTP client, the web-server and the LCD driver and you are above the 8Kb available in the atmega88. An atmega168 has however more than sufficient space. It fills not more than 2/3 of the atmega168. You can therefore easily add additional functions to this wall clock if you want.

## The tuxgraphics AVR NTP clock



LCD display + AVR webservice SMD board is all you need.

The 16x2 LCD display with blue backlight and the AVR webservice SMD board are mounted on an acrylic glass sheet. For the power supply I used an old Ericsson Mobile phone charger (not visible on the picture). It produces 5V DC and is very light as it is a switched power supply. It plugs directly into the wall socket and the 5V DC output is then connected via a 1.5m cable to the clock. I got the charger at ebay for 2 Euro.



NTP clock front view



NTP clock at night

## Using the clock

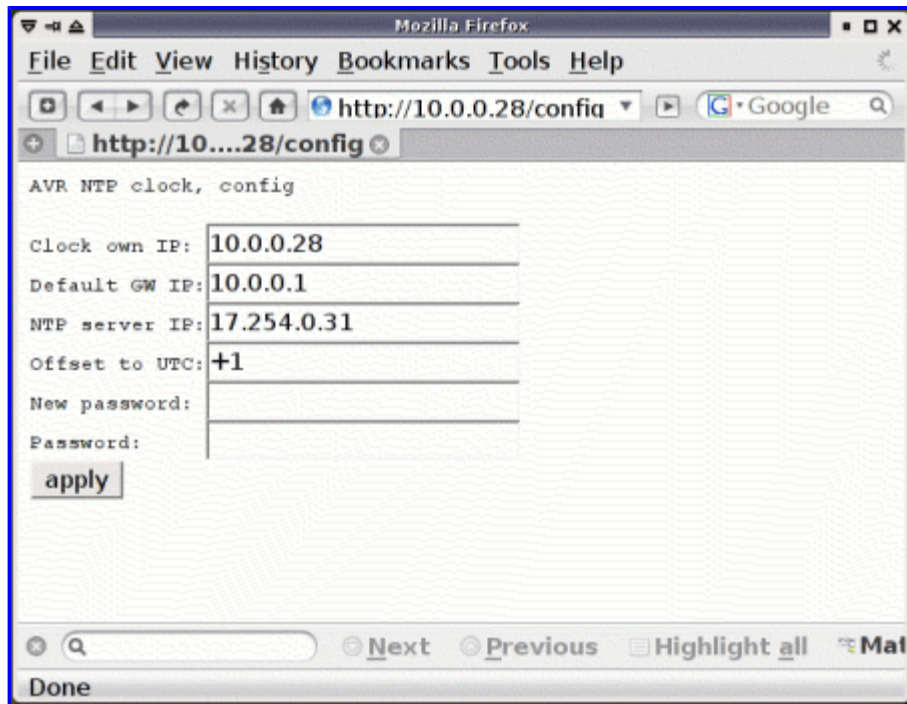
The clock has a LCD display with the following fields:

```
DayOfWeek      Date  X  
Time           (offset to utc)
```

The X in the right corner is a status indicator:

- nothing: ETH link ok, NTP time sync up-to date
- '/': ETH link down
- '|': ETH link up but NTP sever could not be reached yet. The clock tries to sync every hour. Thus if you cut your Internet connection during the night then you will see this character. Note that the clock will continue to run form the internal crystal.

With the build-in web-server you can easily configure the clock:



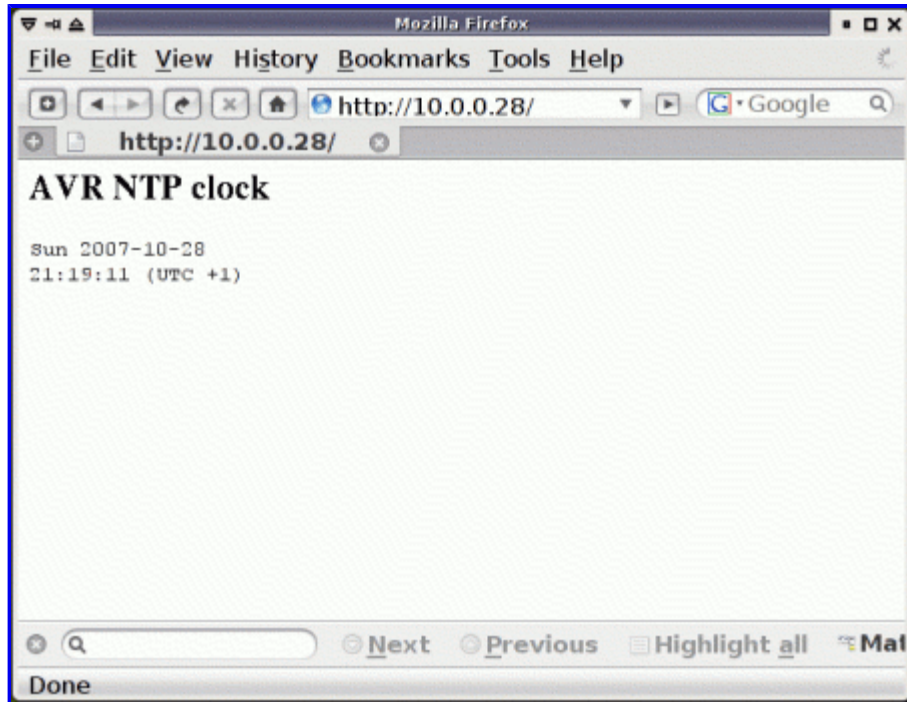
NTP clock config page

The fields on that configuration page (URL <http://ClockIP/config>) are as follows:

- Clock own IP: This is the clock's own IP address. The address under which you can reach it from your local network via web browser. It has to be a free address from your DSL router address range. You should reserve this address in the DSL router by binding it to a MAC address. This is to avoid that the router's DHCP server gives the address by accident to other computers in your network. The mac-address to IP configuration can usually be done under a menu point called "Lan IP settings" or "DHCP configuration" in your DSL router.
- Default GW: This is the internal IP address of your DSL router. Usually 192.168.0.1 or 10.0.0.1.
- NTP server IP: this is an IP address of a public NTP server on the Internet. You find some other examples of addresses in the source code (see download, file main.c) or from <http://www.pool.ntp.org/>.
- Offset to UTC: this is to configure your timezone. You can change this also to adjust to summer or winter time.
- New password: enter only something if you want to change the password
- Password: password to protect the configuration settings. The default is "secret". You have to enter the correct password to apply any changes. In case you forgot the password you can go back to default settings by connecting pin PB0 to GND during power on of the clock.

It is also possible to use this clock if the NTP server to query is not in the internet but on the local LAN. In this case you just specify the NTP server address as both "NTP server IP" and "Default GW".

The web server has also a second page (URL <http://ClockIP>) where you can just see the current time and date as shown on the LCD display.



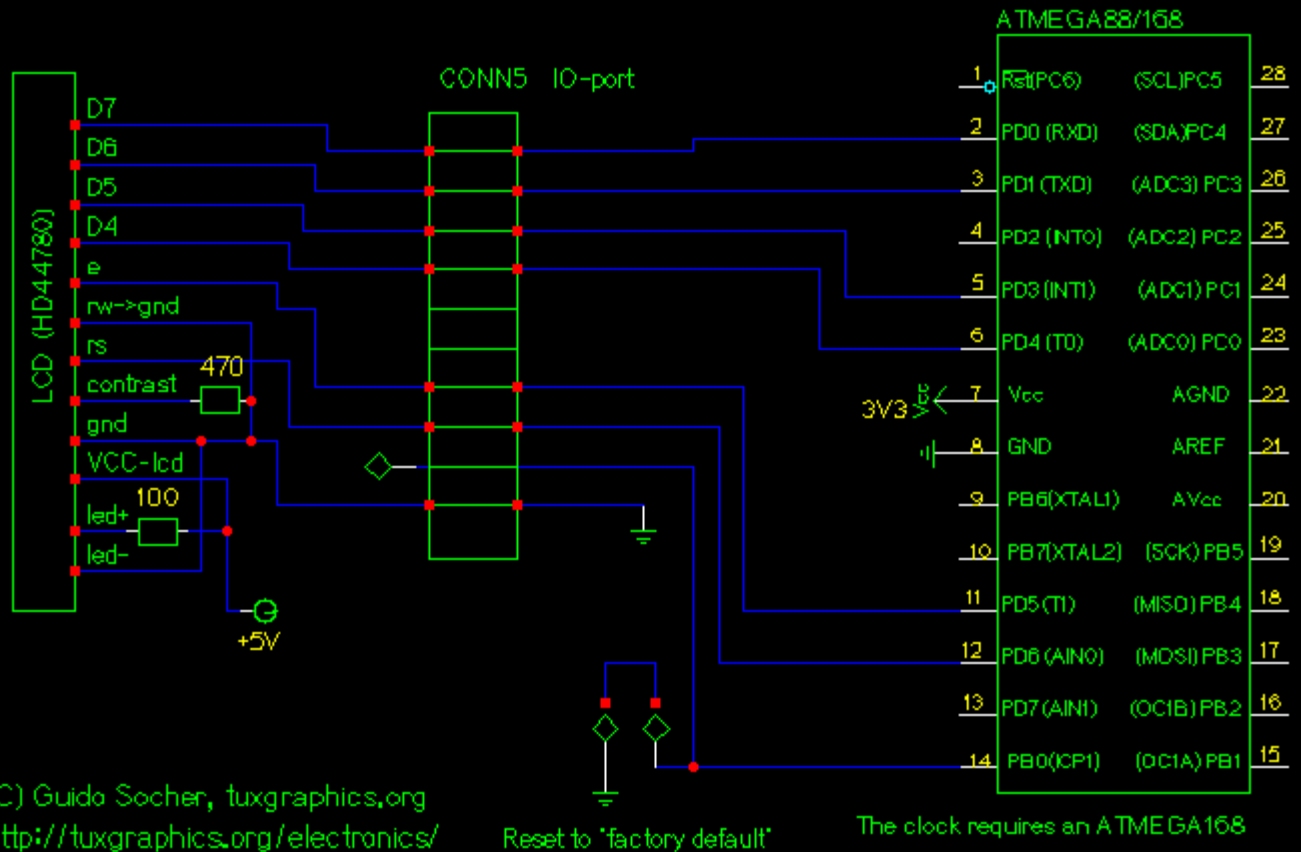
NTP clock via web-browser

## The NTP clock hardware

The hardware is the tuxgraphics AVR ethernet board (aka avr webservice board) and a 16x2 HD44780 compatible LCD display. I recommend to use for the ethernet board our SMD board as it fits nicely behind the display. A circuit diagram of the ethernet hardware is available at <http://tuxgraphics.org/electronics/200611/embedded-webservice.shtml>. The LCD display is then connected as follows:

## AVR NTP Clock

(only the connections of the LCD display to the atmega168 uC are shown here, For all other connections see the circuit diagram of the ethernet remote device circuit)



NTP clock circuit, LCD connection (click for a circuit diagram in PDF format)

You can power the 3.3V regulator of the ethernet board and the LCD display and the back light from a 5V DC power supply. I used an old Ericsson mobile phone charger for that purpose.

Between atmega168 pin PB1 and VCC (3.3V) you can optionally connect a 270 Ohm resistor and a LED. The LED will then go on and off every second (blink).

## FAQ section

Q: My clock works but it is almost 5min off before the next synchronization at the full hour.

A: You forgot to set the fuse bytes. Change the low fuse byte from 0x62 to 0x60: `avrdude -p m168 -c stk500v2 -u -v -U lfuse:w:0x60:m`

Q: When I access the clock via the web browser then everything is fine. The LCD display shows however no time. I see just a row of squares.

A: Make a table on paper and note down which ATmega pin name needs to be connected to which LCD pin name and number. After that check every wire between ethernet board and LCD. There is a mistake somewhere.

## Conclusion

A very nice clock and very pretty to look at ;-). There is space left on the flash memory of the atmega168. You can easily add functionality if you want.

Have fun!

## References

- Download page (source code): [NTP clock download page](#)
- Tuxgraphics electronics section: [tuxgraphics.org/electronics/](http://tuxgraphics.org/electronics/)
- The components for this NTP clock are available at: [shop.tuxgraphics.org](http://shop.tuxgraphics.org)