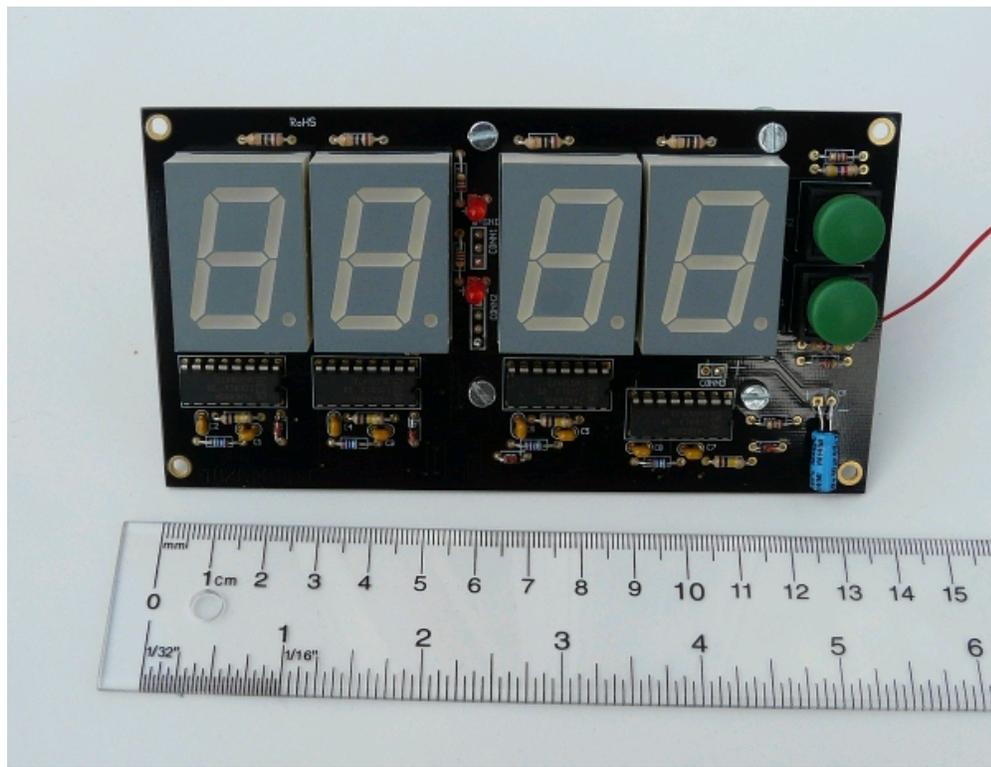




## Large 4-digit LED display, network connected



*Abstract:*

This is a large network connected LED display with 4 digits. The digits are one inch (2.54cm) high, big enough to be used in public places such as waiting rooms or small sports events. This LED display is completely flicker free and the display can be shown on a running camera with no flicker at all.

The setup is extremely easy. You just connect this to the computer network and it's a web server. You point your browser to it and you change the display as needed. It has as well a UDP based machine to machine interface.

---

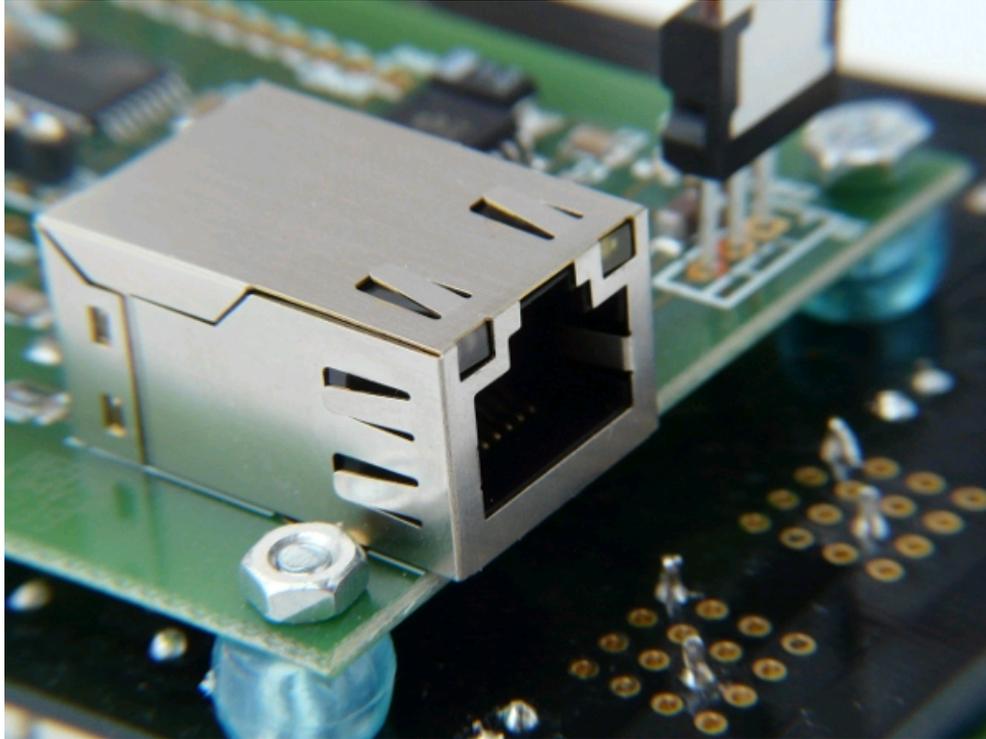
# How does it work?

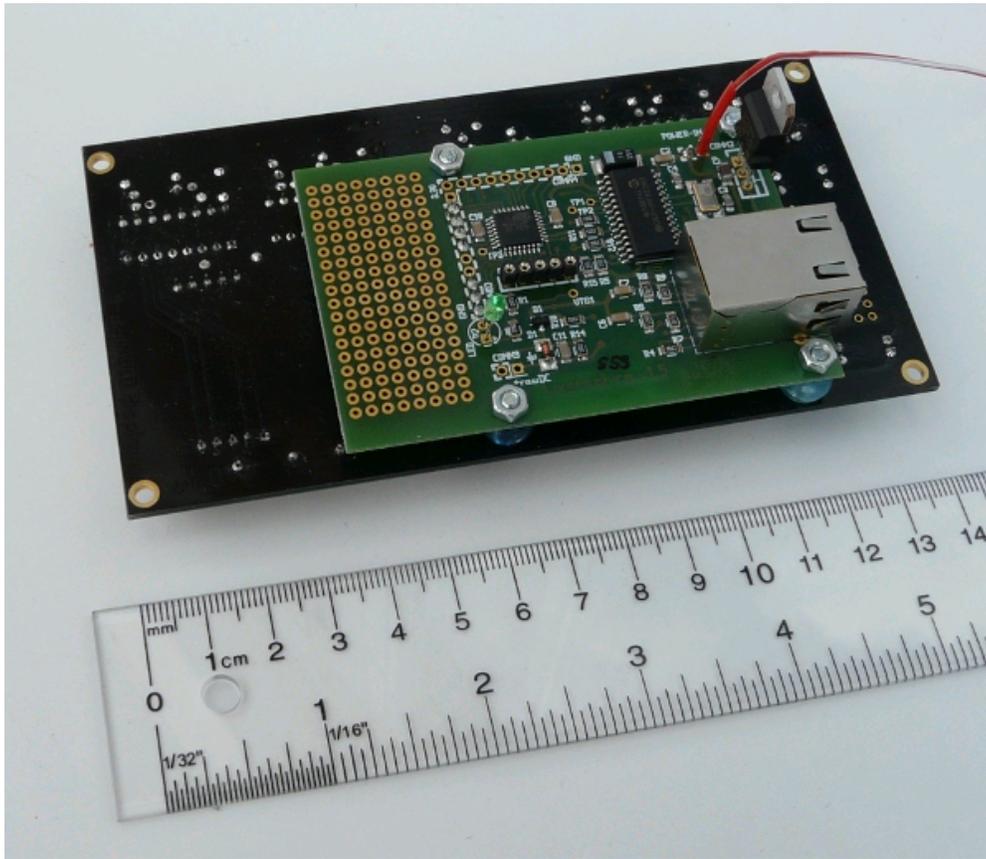
Seven segment displays need 8 IO-pins (7 digit segments + decimal point). That's 32 IO-pins. It would be cumbersome to connect this to a microcontroller.

We use a shift register chip with output buffers. The data is loaded in serial and can then be loaded into the output buffer with one single clock puls. In other words the display update is instant and there is no flicker at all.

# Using the display

This large for digit LED display is network connected.





You just need an ethernet cable and you can control it from any other computer in the network. It's simple but very generic.

The display can be controlled in a number of ways:

1. From a desktop web browser such as firefox, MSIE, safari.... Just point the browser to the IP address of the display and write the number into the web-form:



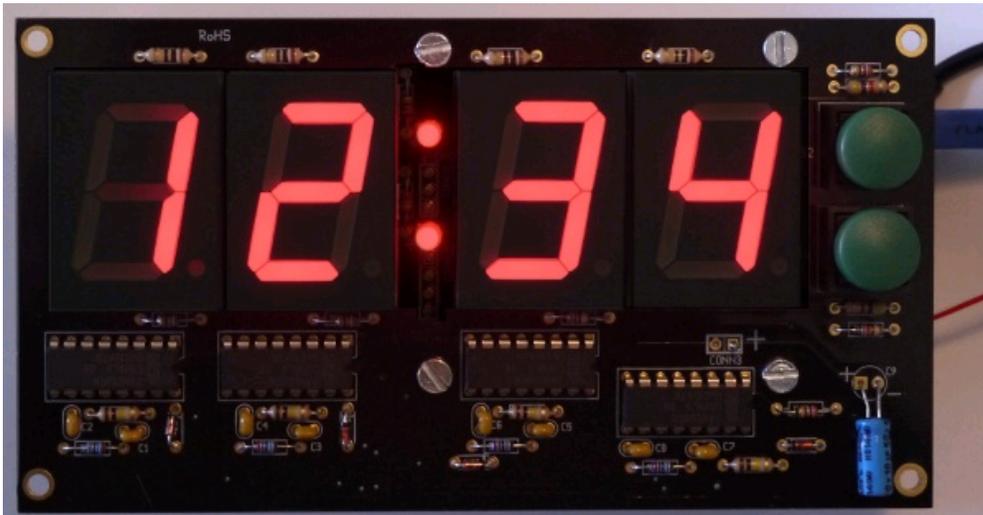
2. You can use a command-line web browser such as lynx:

```
> lynx -dump "http://10.0.0.29/set?n=12:34"  
This would cause the display to show "12:34"
```

3. You can use the udpcorn (provided together with the software for this display) command:

```
> udpcorn n=12:34 10.0.0.29  
II: data: n=12:34, ip: 10.0.0.29 port: 1200  
OK: 10.0.0.29: n=12:34
```

This would cause the display to show "12:34". The udpcorn command is available for Linux, windows and Mac.



The display support besides the digits 0-9 the following special characters:

" " space=display nothing (as of software version 1.0)  
: = the colon in the middle (as of software version 1.0)  
. = a dot (as of software version 1.0)  
- = a dash (as of software version 1.0)

Support for more special characters could be added in future versions.

Examples:

"2 :3.3" would display as:

2 :3.3

"8--1" would display as:

3 - - 1

Note that a space must be entered in the web form as a literal space character and the browser will to the proper encoding. However if you use a commandline web browser such as lynx, curl or w3m then you must encode it since space is not a valid URL character. It can be encoded as either "+" or "%20". The following example sends the string "1 2" (a space between the one and the two) to the display:

```
# curl "http://10.0.0.29/set?n=1+2"  
or  
# curl "http://10.0.0.29/set?n=1%202"
```

# What can I do with it?

The display has 4 digits with an optional decimal point behind each digit and there are two dots in the middle.

You can display things like:

- Any number up to 4 digits (e.g a ticket number: " 812")
- The score in a game between two teams (e.g: "1 : 2")
- Floating point numbers (e.g "4.231")
- Display time in hours and minutes
- A start stop timer

This large LED display has both an easy to use machine to machine interface and a web interface for human interaction. The machine to machine interface allows you to write a little script or a program to update the display.

The board has 2 push buttons on the side next to the display. They are for "future use" because we have some even more ideas for this display.

# Give me an example: a wall clock

This is a unix shell script (runs on a MAC or a linux PC) and it reads the current time periodically from your PC and displays it on the large for digit LED display.

```
#!/bin/sh
# Change this to the IP address of your display:
ip_of_display="10.0.0.29"
curr_time=`date "+%H:%M"`
prev_time=""
while true; do
    if [ "$curr_time" != "$prev_time" ]; then
        udpcmd "n=$curr_time" $ip_of_display
        prev_time="$curr_time";
    fi
    sleep 1 # block loop for one sec then check again
    curr_time=`date "+%H:%M"`
done
```

The script makes use of the udpcmd command to "send" the time every minute to the display. You need to install udpcmd in /usr/bin before you use the script.

Have fun!

# References

## **Technical specifications:**

Power: 5V DC, about 0.3A

Display height: 1 inch (2.54cm), this is the size of the actual numbers.

Size of the board: 13.6cm X 7.3cm

Push buttons: 2

The display uses the tuxgraphics ethernet board (SMD version)

- Download page (firmware source code): [4-digit large LED display, download page](#)
- This 4-digit network connected LED display is available as a kit at: [shop.tuxgraphics.org](http://shop.tuxgraphics.org)