

Hard Disk Activity Monitor

A 10-LED indicator to show how hard it's working

By Karsten Böhme (Germany)

Almost every PC case is fitted with an LED on the front panel which flickers whenever the hard disk(s) is accessed. A single flickering light is better than nothing at all but a 10-LED linear scale showing hard disk loading as a percentage would be really handy!



Features

- A display using 10 LEDs indicates hard drive activity
- Displays hard drive activity in 10 % steps
- Crash proof operation (independent of PC software)
- Neat compact module fits easily inside a PC case

It can sometimes happen, even using the most up to date PC with super fast hard disks and a powerful processor, that while some complex application is running you find yourself staring at the monitor, not quite knowing why the machine is performing slowly or has indeed decided to stop responding altogether. Sound familiar? You cannot be sure if the program has truly got itself into a pickle or intense hard disk activity is responsible for the sluggish behaviour. In such situations it would be helpful if you had a piece of add-on independent 'hardware' which gave some insight into what was happening inside your PC.

This design idea shows real-time hard disk activity in the form of a 10-bit linear display. One advantage of the design is that it runs independently of the PC software so it won't place any additional load on the PC's proces-

sor. It is also unaffected by internet-borne bugs or viruses (assuming there are none in the microcontroller firmware to start off with) and will function even if an application running on the PC has crashed. Despite the market for PC case modding and customisation being relatively large there doesn't appear to be any similar hard disk indicator available, but on second thoughts I guess that is what we have Elektor for!

More LEDs!

The average PC usually has just a single flickering LED giving you a rough indication that the hard disk(s) is being accessed. The modification described here shows hard disk activity in a much more informative linear display using ten LEDs (each step represents 10 %). The display is refreshed five times per second, giving a good responsive indication.

The question is how has the author managed to gain access to this information because it is not output from the hard disk controller in either analogue or digital form. His solution is to treat the on/off signal to the hard disk LED as a pulse-width modulated signal, simply measuring the on to off ratio of this signal gives an indication of how hard the disk is being used. This approach works surprisingly well. The wire from the motherboard normally connecting to the hard disk LED is instead connected to the

input on the hard disk monitor board. A microcontroller then reads the state of this signal during 200 ms windows to determine the amount of time that this signal is high during each period. A high for the entire period will result in all ten LEDs lit while a low for the entire period will turn them all off. The photo of the author's PC shows all ten LEDs lit. The first seven (the range from 1 to 70 %) uses green LEDs, the next two (71 to 90 %) are yellow and the final position is red (91 to 100 %).

Decoder and LED driver

From the functional description and circuit diagram you will be aware that the circuit uses a microcontroller. A low cost Atmel ATtiny2313 microcontroller together with an optocoupler and LEDs are connected as shown in the diagram in **Figure 1**. The wire which originally connected the motherboard to the hard disk indicator LED is instead connected to the monitor board input where it now drives an optocoupler. Its LED input stage forms a good substitute for the indicator LED and reduces the possibility of any signal mismatch. The original hard disk indicator LED does not go unused because the circuit provides an output to drive it in the same way that the signal from the motherboard did originally.

IC1 drives the indicator scale consisting of ten LEDs driven via 470 Ω series resistors.

Note. Readers' Projects are reproduced based on information supplied by the author(s) only. The use of Elektor style schematics and other illustrations in this article or the availability of project (software) downloads from the Elektor website does not imply the project having passed Elektor Labs for replication to verify claimed operation.

To increase the number of LEDs it would be necessary to use a larger controller with corresponding firmware. It is important to ensure that not only the maximum current output from each pin is within spec (10mA here) but also that the current drawn by the complete IC (60 mA for this device) is not exceeded. If a brighter display is required (which many users find unnecessarily distracting) more efficient low-current LEDs can be used.

The complete circuit is powered from a 5 V voltage source (use the 5 V from a spare power connector in the PC normally used to power a hard or floppy drive). A 10-way pin header is included on the PCB to allow in-circuit programming of the microcontroller.

Construction

For this project the author has designed a PCB for mounting the entire circuit including display LEDs. The PCB layouts for the component side (file: PCB_top.jpg [1]) and the underside (file: PCB_bottom.jpg [1]) of the board show that despite the use of SMD components the layout is quite well spaced out. The author has specified 1206 outline SMD resistors which are relatively easy to fit by hand. It is not anticipated that mounting any of the other components will present any great difficulty. All of the components except for the LEDs are fitted to the component side. The LEDs are mounted on the PCB underside at a suitable stand-off so that the LEDs fit snugly into the front panel holes when the PCB is fixed (on spacers) behind the front panel. A 1:1 scale photocopy of the PCB layout can be cut out and used as a template to drill the ten LED holes in the PC front panel. The LEDs can also be arranged as in the author's prototype board (see Prototype.jpg at [1]).

It is important to specify that the microcontroller is clocked by a 4 MHz internal clock when the firmware is flashed to memory. Without this step the controller runs noticeably slower and affects the display refresh rate. The screen shot in **Figure 2** shows how the clock fuses are configured in the Atmel AVR Studio 4 development environment. The author has written the software in C using the CodeVision compiler. The source code is well documented to allow simple modification should you be tempted to

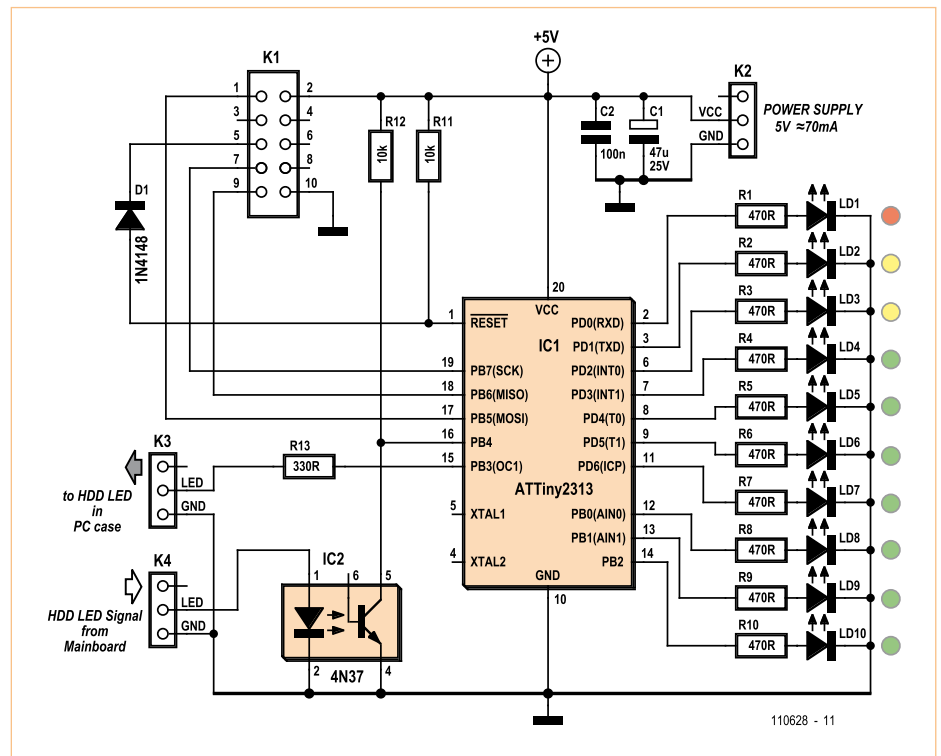


Figure 1. The hard disk monitor circuit diagram is quite simple.

add your own improvements. For the less adventurous there is also a complete Hex file available which can simply be flashed to the microcontroller memory. Both files form part of the free downloads for this article [1] together with the PCB design files in Eagle format. To see the display in action click on the video in the link below.

(110628)

Internet Link

[1] Video, software and author's PCB artwork files: www.elektor.com/110628

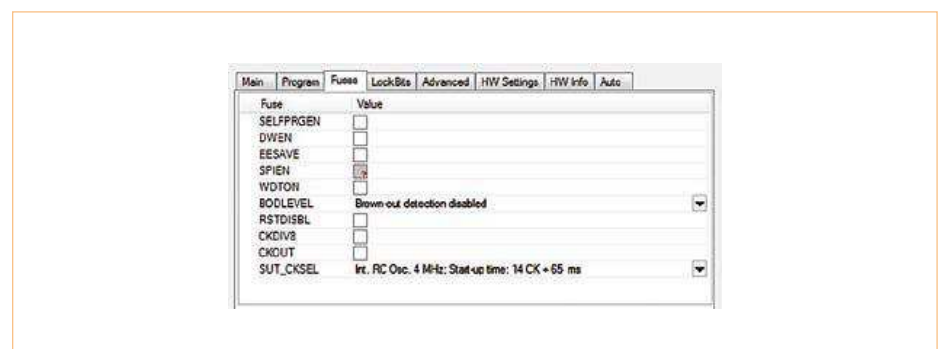


Figure 2. The fuse settings in AVR Studio 4 to give an internal 4 MHz clock.