

RS-485 Switch Board

ElektorBus relay module

By Jens Nickel

Our ElektorBus series has shown how much interest there is in home automation applications. Here we describe a small circuit board that can switch two AC (230 VAC) loads. Also, two of the inputs to the on-board microcontroller are brought out to terminals so that the state of two switches can be read back. The board works with the ElektorBus and so is an ideal building-block for a home automation system controlled from a PC, tablet or smartphone.

So far the ElektorBus project has centred around hardware and software for experimental and development purposes. We will now start to move towards a more finished system, in particular looking at home automation applications, which have generated a lot of interest among our readers. The board we describe here is designed to switch two AC (230 V) loads and to read the state of two switches. The module is compact and based on an ATmega88 and an LT1785 RS-485 driver, the same ICs as we used in the ElektorBus experimental node [1]. Demonstration software that talks the ElektorBus protocol is also available for download, to run on microcontrollers, PCs and smartphones. The relay module, which is available ready-built from Elektor, can of course also be used in other projects.

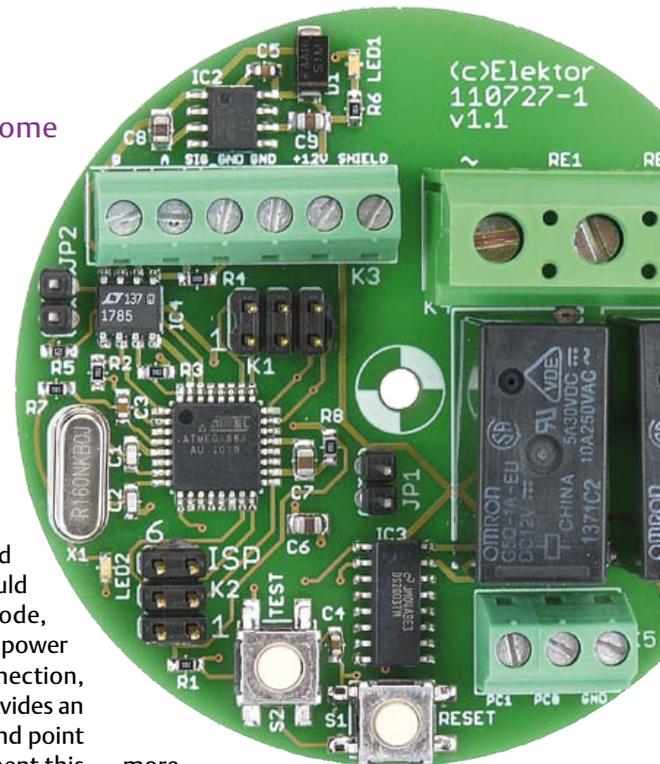
Bus connection

The circuit diagram is shown in **Figure 1**. The circuit receives its 12 V DC power supply over screw terminal connector K3. The RS-485 part of the circuit closely mirrors that of the experimental node, with data lines RO and DI, control lines DE and /RE, which allow half-duplex operation, and the RS-485 bus signals A and B. A 120 Ω termination resistor can optionally be wired across the bus lines using a jumper. The four wires of the ElektorBus are connected at K3: 12 V and GND to supply power and the A and B data lines. Besides those there are two extra connections. The first, 'Shield', is intended to be connected to

the cable's screen. This is connected to ground via jumper JP1: this should normally be done only on one node, for example the one that provides power to the bus. The second extra connection, 'SIG_GND' (for 'signal ground') provides an additional connection to the ground point of the RS-485 driver. For the moment this signal is simply connected to bus ground in the relay module. However, we wanted to allow for the future possibility of providing a separate ground for the RS-485 driver IC. When loads are switched it is possible that there will be local interference generated on the bus ground; and if the ground potentials at the drivers differ too widely this can result in spurious bits being detected on the bus. We have already observed this effect once [2]. One solution might be to connect signal ground directly to bus ground at only one point on the bus, while in the other bus nodes the two signals are linked via (for example) a 100 Ω resistor. Of course, this arrangement requires a fifth conductor in the bus cable. We have not yet tested this possibility, but keep the option open for when the bus is a little more mature: by all means experiment yourself!

Microcontroller pins

The test LED, test button, ISP (in-system programming) interface and clock and power supply circuitry will all be familiar from the experimental node design. We have replaced the single-in-line expansion connector with a 2-by-3 header, which is



more compact and for which it is easier to find suitable cables. The connector gives access to four of the microcontroller's pins, each of which can be used as a digital input or output or as an analogue input (ADC0 to ADC3). Two of these pins, along with ground, are also brought out to screw terminals, making it easy, for example, to connect a light switch. **It should go without saying that the switch and all associated cables should be kept well away from any AC powerline wiring!** The main feature of the module is the pair of relay-switched outputs. These are connected to two of the microcontroller's pins, PB0 and PB1, via relay driver IC3. The relays are SPST NO (single pole, single throw, normally open) types. With the circuit as shown these switch between the input labelled 'phase' and the outputs labelled 'S1' and 'S2'. These three signals are brought to larger screw terminals. The relay outputs can be used to switch low-voltage equipment or mains equipment such as lighting. If AC powerline voltages are used, the module must be mounted in an enclosure that prevents accidental finger contact in accordance with all applicable regulations

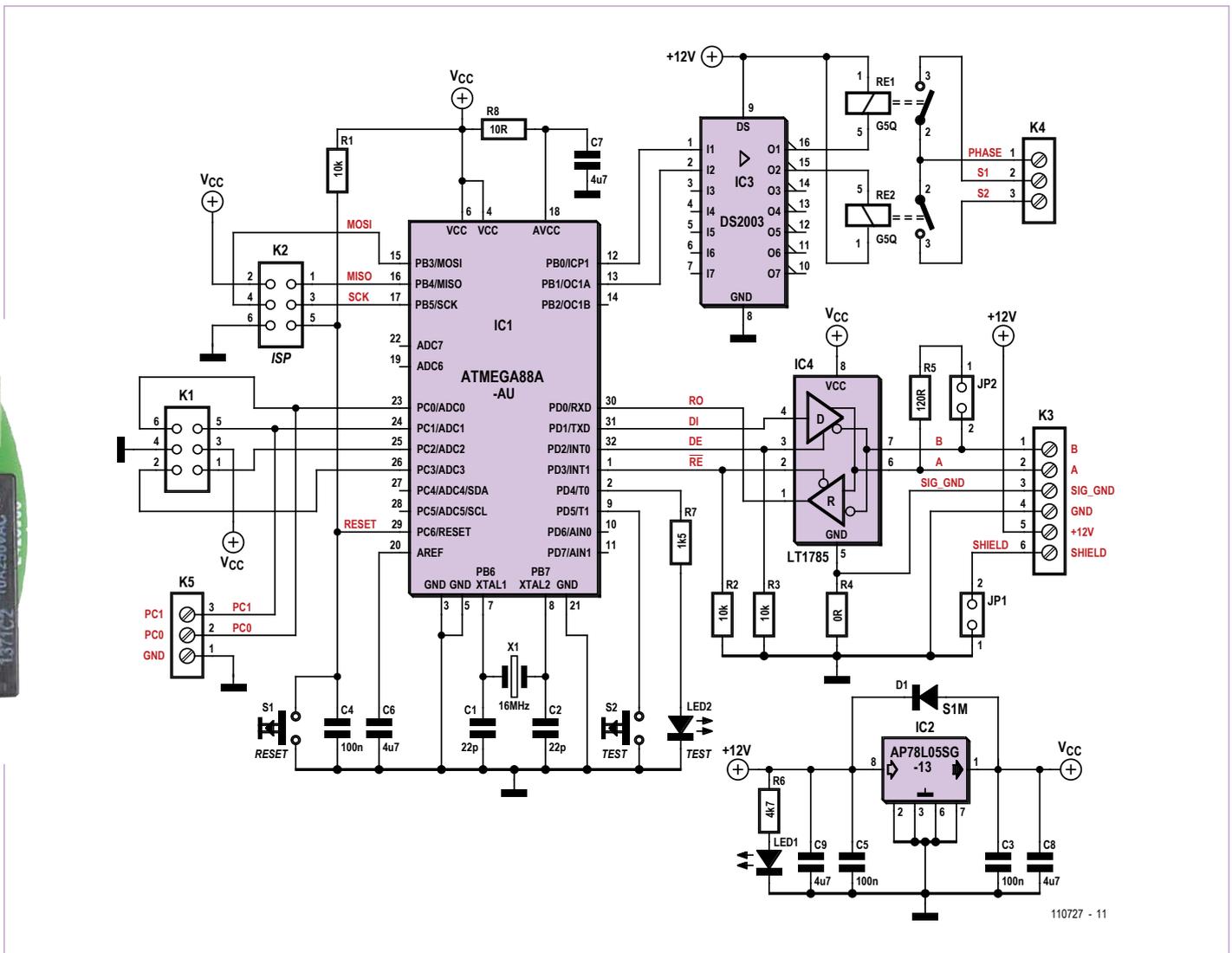


Figure 1. Up to two loads and two switches can be connected to the module using the screw terminals.

in your state or country.

The printed circuit board is round and compact, and can be fitted into an enclosure with a diameter of 60 mm (Figure 2). Do **not** fit the board in a junction box that already carries domestic AC grid wiring, and the bus wires should not be run parallel to AC wiring in trunking (see also the 'Cabling' text box). The switching module should be fitted in its own enclosure and the bus cable in its own trunking. In many countries electrical installations such as this must only be carried out by suitably qualified or competent people.

Software

With suitable firmware we can breathe life into the switching module. The ready-built modules are delivered with the microcontroller not programmed, as the board can be used for a wide range of applications. However, there is of course demonstration software available, which is compatible with the ElektorBus. The relays can be switched from a PC, using our USB-to-RS-485 converter and the ElektorBusBrowser.exe software. The same can be done from an Android smartphone or tablet, using the combination of the AndroPod board and the

free ElektorBusBrowserForAndroPod app described recently [3], [4]. We also provide a suitable user interface to allow the relays to be controlled. This is implemented in HTML and works equally well on PCs and Android devices.

The hardware is wired up as shown in Figure 3. As usual, the software is all available for free download from the project web page accompanying this article [5], and source code is of course provided.

After opening the zip archive the first thing to do is drag the 'UIBus' folder to the desktop. Then, if you will be using a smartphone

Elektor Products and Support

- RS-485 switch board (ready built and tested): # 110727-g1
- USB-to-RS-485 Converter (ready built and tested): # 110258-g1

- Free software download

All products and downloads are available via the web pages accompanying this article: www.elektor.com/110727

COMPONENT LIST

Resistors

SMD shape 0603
 R1,R2,R3 = 10kΩ
 R4 = 0Ω
 R5 = 120Ω
 R6 = 4.7kΩ
 R7 = 1.5kΩ
 R8 = 10Ω

Capacitors

C1,C2 = 22pF NPO (0603)
 C3,C4,C5 = 100nF X5R (0603)
 C6,C7,C8 = 4.7μF X5R 10V (0805)
 C9 = 4.7μF X5R 25V (0805)

Semiconductors

D1 = S1M (1000V / 1A)
 LED1,LED2 = LED, green (0603)
 IC1 = ATmega88A-AU, programmed
 IC2 = AP78L05SG-13
 IC3 = DS2003TM/NOPB
 IC4 = LT1785CS8#PBF

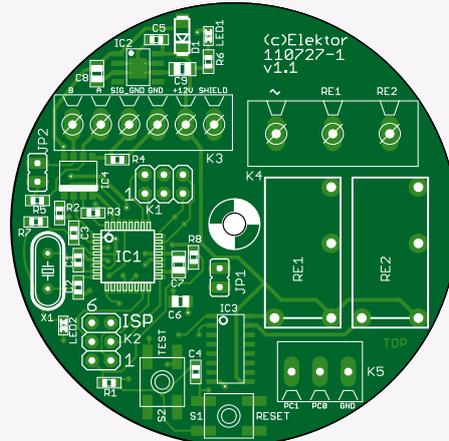


Figure 2. To keep the board compact we have used SMD components. The circular shape allows installation in an enclosure with a diameter of 60 mm.

Miscellaneous

RE1,RE2 = 12V, SPST-NO (e.g. G5Q-1A-EU DC12)
 JP1,JP2 = 2-pin pinheader, lead pitch 0.1 in. (2.54mm)
 K1,K2 = 6-pin pinheader, lead pitch 0.1 in. (2.54mm)
 K3 = 6-way screw terminal block, lead pitch 3.81mm
 K4 = 3-way screw terminal block, lead pitch 7.62mm
 K5 = 3-way screw terminal block, lead pitch 3.81mm
 X1 = 16MHz quartz crystal, HC-49S case
 S1,S2 = pushbutton, SPNO (e.g. B3S-1000)
 PCB # 110727-1

alternatively

Ready assembled and tested board # 110727-91

as a controller, transfer the required files to it, for example using the PC software Adif-Controller as described in [4]. The firmware needs to be programmed into the micro-controller: the firmware is written in C using the 'AVR Studio' environment, and a ready-compiled hex file is included in the download. In this example the data EEPROM in the ATmega88 is not used: the board's node address (which is '5') is hard-coded into the program.

Testing

After firing up the ElektorBusBrowser the HTML user interface will appear: see **Figure 4**. In the PC version the COM port address must be set up first using the combo box at the top of the screen, matching the address used by the USB-to-RS-485 converter. Then click on the button below. The ElektorBus scheduler can now be launched by clicking the 'on' button in the HTML user interface. The scheduler now polls the master (with node address 10) at regular intervals to prompt it to send any

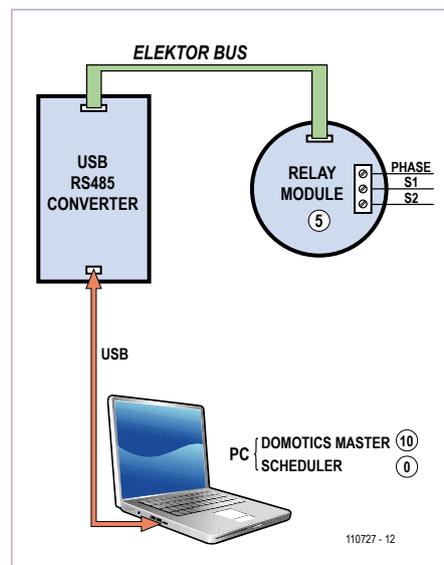


Figure 3. For a first test connect a USB-to-RS-485 converter and a relay module together over the bus. The relay module has node address 5 (hard-coded into the demonstration firmware).

switching command that might be available to the relay module. It is of course necessary to arrange for the command to be acknowledged: when the relay module receives a command, it replies with a message that contains the current state of its two relays. This means that the module is never directly addressed by the scheduler. As a consequence it is possible to have over a hundred relay modules connected to a single bus without the congestion that would arise from having the scheduler poll them all. In fact, the status message is sent during the so-called 'FreeBusPhase' which is initiated by the scheduler after each occasion when the master is polled. During a FreeBusPhase any bus participant that happens to have something to say is free to send a message. A collision can occur if there is more than one such participant, and so, if safe receipt is important, the acknowledge message from the relay module must in turn be acknowledged by the PC. This FreeBusPhase 'AcknowledgeMessage' is automatically sent by the JSBus Javascript

Cabling

As we reported in the 'E-Labs Inside' section of the March 2012 edition, we have been testing various cables for use with the bus. You can see the results in a YouTube video [7].

We started with a ten-core cable from the store room, which had one pair twisted. Next we tested some CAT 5e cable (four twisted pairs with an overall screen). We found cable lengths of 30 m caused no problems at all in either case, and so it seems that we do not need to recommend a specific cable. The only important aspect

is that the A and B signals should be carried on a twisted pair.

It is not normally allowed for the network cable to be carried in the same conduit as mains cables carrying 230 V. In some countries exceptions are made for special-purpose cables, of which one example is the YCYM 2x2x0.8 EIB/KNX bus cable designed for home automation applications. It should be possible to use such a cable with the ElektorBus, but we have not tested it. Such cable is somewhat pricey: it can cost several pounds per metre.

library that is loaded as part of the demonstration HTML file (see Listing). The library decodes the message from the relay node, and the user application code simply has to accept the two 'parts' (information units) which comprise the message. The first part represents the status of relay 1 (sent on channel 0) and the second part the status of relay 2 (on channel 1 within the same message). The function `ProcessPart`, which is defined in the javascript code in the HTML page, is called for each of the two parts. Within this function the status value (zero for 'off' or one for 'on') is used to update the status text suitably for the appropriate relay in the HTML user interface.

We have covered the above more thoroughly in previous articles on the Elektor-

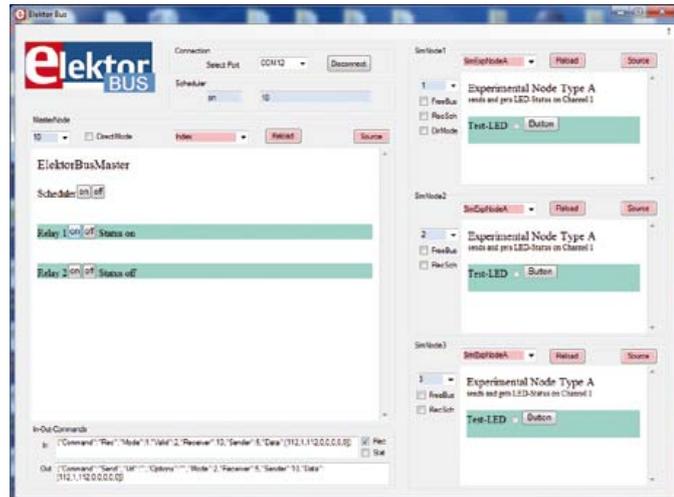


Figure 4. The ElektorBusBrowser PC software and the HTML user interface (left). In the window to the right you can see simulated bus nodes which can be brought into action for development purposes [8].

Advertisement

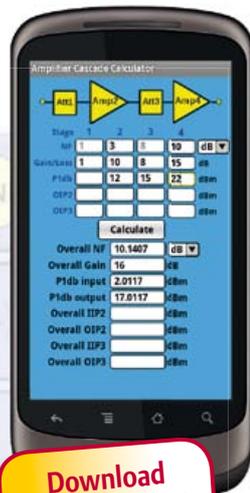
Elektor RF & Microwave Toolbox

for Android

The RF & Microwave Toolbox contains 34 calculation and conversion tools for RF, microwave and electronics in general. Whether you are an RF professional, radio-amateur, astronomer or hobbyist, this app puts some of the most important tools right at your fingertips. Literally!

Highlights:

- Amplifier cascade (NF, Gain, P1db, OIP2, OIP3)
- Field intensity and power density converter (W/m², V/m, A/m, Tesla, Gauss, dBm, W)
- PCB Trace calculator (impedance/dimensions)
- Pi and T attenuator
- Antenna temperature (Kelvin)
- EMC (EIRP, ERP, dBμV/m)
- Filter Design (Butterworth, Chebyshev, prototype)
- And much more



Download your app now!

Available in
Android Market

Further information at

www.elektor.com/rf-app

SiliconRay
Online Electronics Store

• Boards • Kits • Modules • Components • Tools • Instruments
FOR

• Companies • Professionals • Students • DIYers • Amateurs



USB STC 8051 MCU Programmer

Free For 8051 Developers

\$0

Add to cart



SEN1060 GPS Module With Antenna

Small, Low Cost, Easy To Develop

\$50

Add to cart



(HMC5883L) 3 Axis Compass Module

Integrate Compass Easily

\$12.5

Add to cart



LM3886 68Wx2 HiFi Audio Amplifier

HiFi Sound At Affordable Price

\$49.9

Add to cart

Simplify your electronics projects by visiting
<http://www.siliconray.com>

Listing: User interface in HTML and Javascript (excerpt)

```
<SCRIPT src='JSBus.txt' Language='javascript' ></SCRIPT>
<SCRIPT Language='javascript' >

function ProcessPart(part)
{
  if (part != null)
  {
    if(part.Sender == 5)
    {
      if(part.Channel == 0)
      {
        if (part.Numvalue == 0) {TextSetvalue ("StatusRelay1","off");}
        if (part.Numvalue == 1) {TextSetvalue ("StatusRelay1","on");}
      }

      if(part.Channel == 1)
      {
        if (part.Numvalue == 0) {TextSetvalue ("StatusRelay2","off");}
        if (part.Numvalue == 1) {TextSetvalue ("StatusRelay2","on");}
      }
    }
  }
}

function SwitchRelay1(Status)
{
  var parts = InitParts();
  parts = SetValue(parts, 10, 5, 0, 0, Status);
  SendParts(parts, true);
}

function SwitchRelay2(Status)
{
  var parts = InitParts();
  parts = SetValue(parts, 10, 5, 1, 0, Status);
  SendParts(parts, true);
}

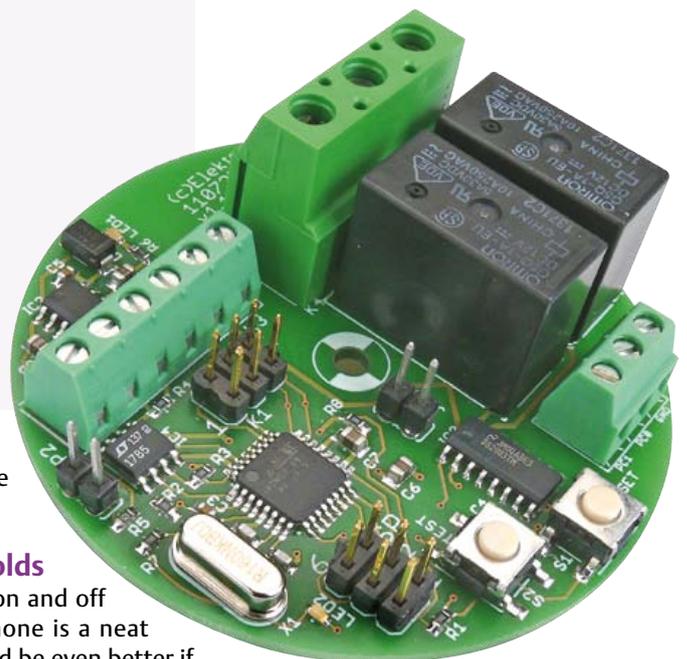
</SCRIPT>
...
```

everything could be operated remotely, even from the other side of the world. In the near future we will look at how this can be achieved: in the pipeline we have an expanded microcontroller board that includes not only an RS-485 interface (of course!) but also a socket for mounting a network interface module. This will let us use the communication facilities of smartphones and tablets to the full in conjunction with the ElektorBus. And the final destination of our bus? You will have to wait and see!

(110727)

Internet Links

- [1] www.elektor.com/110258
- [2] www.elektor.com/110225
- [3] www.elektor.com/110405
- [4] www.elektor.com/120097
- [5] www.elektor.com/110727
- [6] www.elektor.com/elektorbus
- [7] www.youtube.com/watch?v=rbDSTXNARmw
- [8] www.elektor.com/110708



Bus. However, from now on it won't be necessary to look back at those articles to check some detail: information about the message protocol, application protocol and the rapid development system built around HTML and Javascript can be found in the new ElektorBus Reference document,

which is available for free download at [6].

What the future holds

Switching equipment on and off using a PC or smartphone is a neat application, but it would be even better if