digital-powerizer and i2c-powerizer: the missing rings in Domotics

Renzo Davoli
University of Bologna

Abstract

This paper describes two different circuits. The digital-powerizer is a mains switch driven by a digital signal. In my applications digital-powerizer come in units containing eight parallel digital-powerizers. The second i2c-powerizer is a circuit used to turn on and off eight digital signals. This latter circuit can drive eight digital-powerizers to turn on and off eight loads (e.g. eight lights).

1 digital-powerizer

Based on optoisolators and triacs, this circuit is able to convert a digital signal coming from some source (e.g. from a microcontroller) to a on/off operation for electric loads. The functionnality if the digital-powerizer is streightforward simple. When connected to a load, say a light, if the input signal is a “logic 1” the light is on, when the input is “logic 0” the light is off. A digital powerizer splits the low voltage part of the logic circuits from the high voltage section, it is an iterface between digital logic and electric utilities. It can be seen as a voltage decoupling unit, or, as you like, as a danger insulating unit: in a computer controlled domotics application this unit can be the only one where dangerous voltages are present. This unit itself must operate in a closed box as high voltages pass through the most part of the pcb.

The MOC3020 optoisolator with DIAC output drives a BT137 triac. Different triacs can be used for different loads, BT137 supports about 8A. A snubber network is used to protect the triac from spikes coming from inductive loads. It is suggested to protect the whole circuit with a 8A fuse or a magnetic switch to limit the current to the maximum supported by the triac. Four different input options are dipicted depending on the voltage of the input signal (3.3 or 5 volts) and the presence of a led to control the status of the digital powerizer.

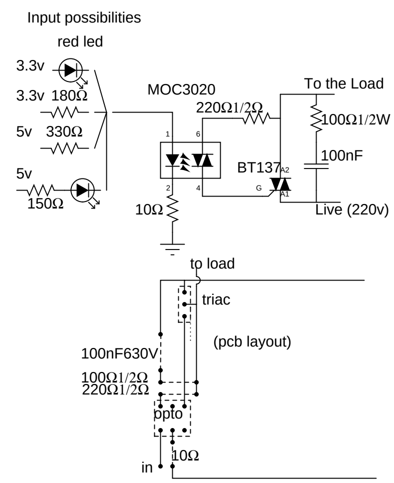

Figure 1: Digital Powerizer schematics

2 i2c-powerizer

Based on a ATmega8 or ATmega48/88/168 this circuit can turn on/off eight digital signals using eight push buttons. It works like a group of eight push-on push-off relay. The status of the eight “digital switches” can be inspected and changed Atmel two wire interface commands (compatible with i2c).

I2c-powerizer solve the “computer reliability dependance” for domotics applications. I2c-powerizer is very simple and completely independent from the controlling computer. If no data gets exchanged by the i2c network to the controlling unit, i2c-powerizer buttons still can be used to turn on and off all the circuits.

I2c-powerizer circuit is very simple, as it is the controlling software. Filters to avoid spourious signals on the push button lines have been implemented in software using shift registers, only signals stable for at least 8ms can pass the filter.

The circuit has just the microcontroller, the power supply capacitor, a reset pullup resistor, four dip switches (for the i2c address) and several sockets: power supply, input (to push button), output (e.g. to a digital powerizer unit), and i2c.

I2c-powerizer is able to work at different voltages. Using ATmega 48/88/168 both 3.3V and 5V are possible while ATmega8 requires 5V (provided it is not a L version). All the units on the i2c bus should work at the same voltage, otherwise there can be data loss.

I2c-powerizer is a i2c slave unit, and it can be connected to any i2c master. For example it can be connected to a serial port of a personal computer by the interface shown in Figure 5 (It is the same interface used by Guido Socher

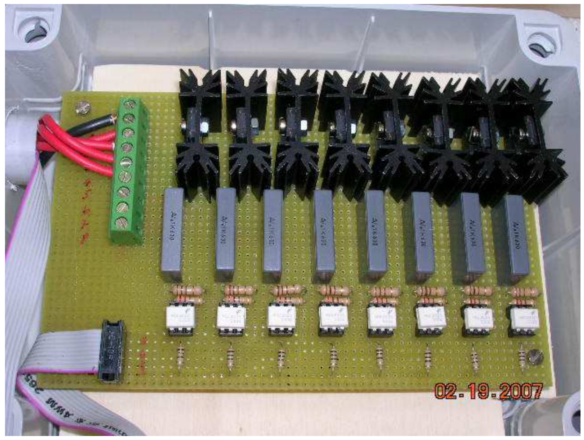

Figure 2: A picture of an eight channel Digital Powerizer

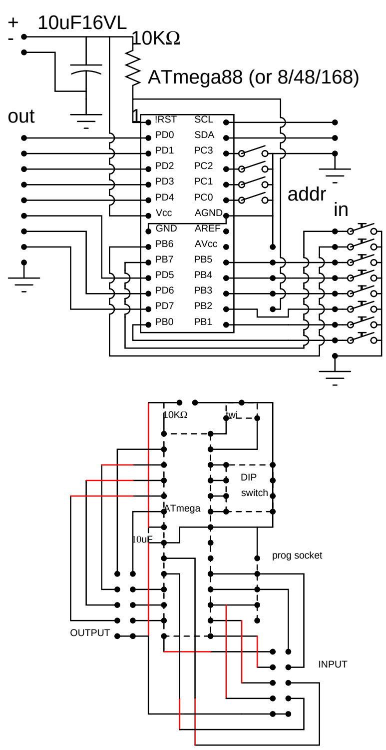

Figure 3: i2c Powerizer schematics

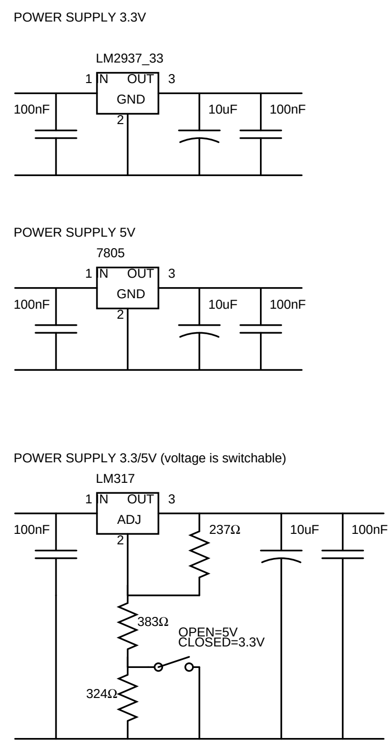

Figure 4: i2c powerizer power supply circuit options

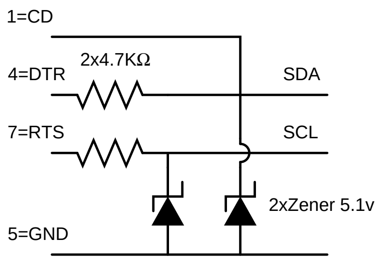

Figure 5: i2c to PC serial port interface

.

of Tuxgraphics for his i2c thermometer).

There is a program (i2x_linux), based on Guido's i2c_m library, to communicate with the powerizer. e.g.

```
$ ./i2c_linux -s 3
```

turn on channel 3 on client 0 (default, all dip switches open)

```
$ ./i2c_linux -c 1
```

print the status of the unit with i2c address 1

I2c-powerizer can be connected directly to the Ethernet device designed by Guido Socher[1]. The pcb has already the holes for SDA e SCL (microcontroller pins 27 and 28 respectively), two 4.7K pullup resistors to Vcc, one for each signal, are needed for i2c to work correctly. Figure 6 is a snapshot of an application using an i2c-powerizer with Guido's Ethernet unit.

The software loaded on the microcontroller has an interrupt based management of i2c and a main loop to control the input and change the outputs consistently.

The unit is able to recognize two types of commands (each one is one byte long) (write):

- `1 0 0 0 0 x x x` (hex 80 to 87), turn off the xxx (command & 0x7) circuit;
- `1 0 0 0 1 x x x` (hex 88 to 8F), turn on the xxx (command & 0x7) circuit;

It is possible to send several commands in one i2c message. All the bytes not matching any valid command are simply discarded.

When the unit is addressed for reading, one single byte contatining the current status is returned. Channel 0 status is the LSB while Channel 7 the MSB.

The pre-defined i2c addresses are in the range 0x10 to 0x1F. The dip-switch 0 connected to the pin PC0 (22) is the LSB. PC3 (pin 26) is the MSB of the address and must be set only if the i2c-powerizer is used in combination with external push-on push-off relay. When PC3 dip switch is closed the outputs are driven just by the i2c commands (not by the input) and the status returned by thw i2c command is the current situation of the inputs.

[1] see http://www.tuxgraphics.org/electronics/200606/article06061.shtml

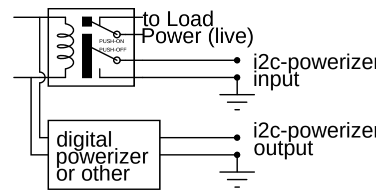

Figure 7: i2c powerizer connected to a push-on push-off relay (addr ¿= 8)

3 Conclusions

The two circuits presented here are two missing rings for domotics. The former (digital-powerizer) is the "actuator" of commands coming from computers or microcontrollers. Turning on and off high voltage equipments becomes as easy as turning on and off a led diode. The latter (i2c-powerizer) achieves two goals: (1) it is an interface between a domotics CPU (the i2c master) and the digital-powerizer, (2) it works as an independent unit. Standard (manual) functionnalities are preserved dispite of crashes of the network or of the controlling unit or complete reboot of the domotics system.

4 Legalese

-
-
- Any new idea included in this paper is **not patentable** as the schematics and details has already been filed with legal timestamp to be used as prior art.

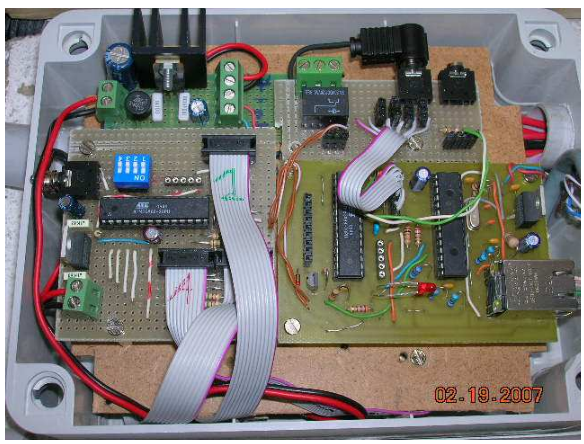

Figure 6: A picture of an i2c-Powerizer (lower left corner) in a box with an Ethernet controller