

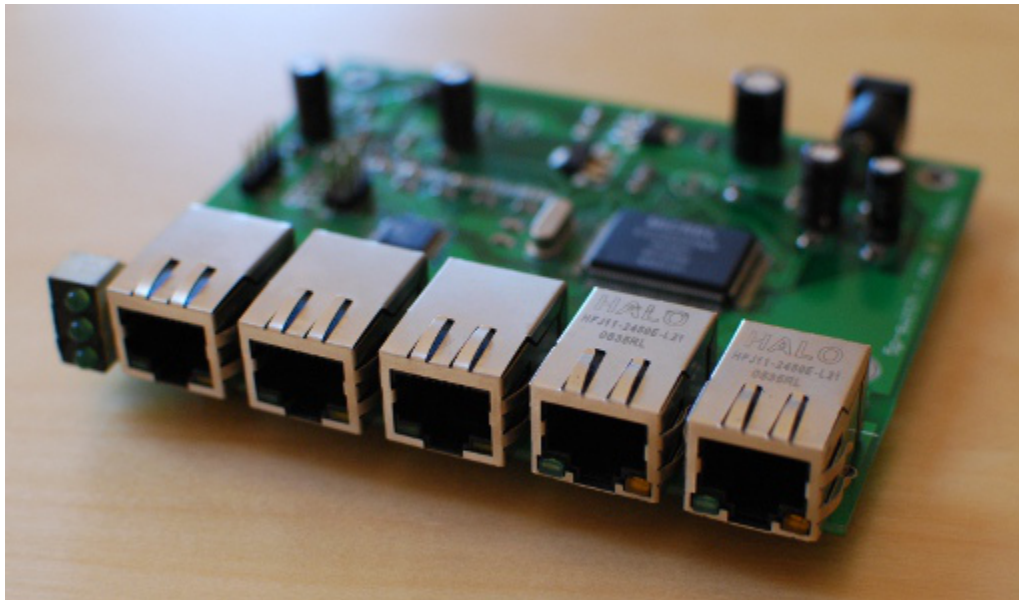
FIVE PORT FAST ETHERNET MANAGED SWITCH

This project is the complete design of a 5 port fast Ethernet switch, based on the Micrel KSZ8995M switch IC with integrated PHY and an ATmega168 8-bit microcontroller.

The hardware was originally developed in 2009, because I needed a compact, VLAN capable switch to expand the functionality of an [NSLU2](#), which has just a single Ethernet port and that I was using as router for my home network.

Features

- ▣ Complete VLAN support
- ▣ Auto MDIX on all ports
- ▣ Programmable port rate limiting
- ▣ Integrated MIB counters
- ▣ TTL UART interface for external access of switch configuration



Hardware

The main component of the Ethernet switch is... (surprise) the Switch! This project is based on a [Micrel KSZ8995MA](#), which is a 5 port switch, managed, with 5 integrated auto-MDIX capable PHYs.

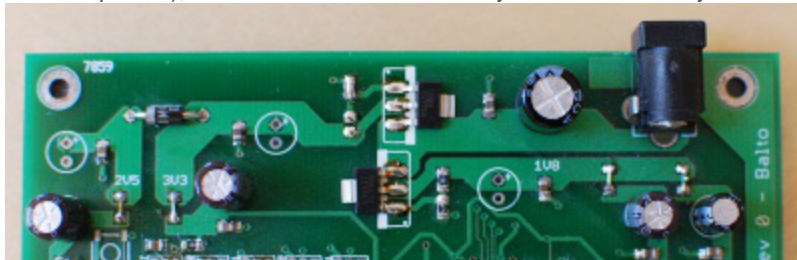
The device is available from major IC distributors in a 128 pin PQFP package, which can be easily hand-soldered if you have a good hand and the right tools.

If you work on electronic designs, you can also request the IC as a sample from Micrel itself... just don't abuse the sample service!



The switch requires three power rails: 3.3V for the digital I/O, 2.5 (or 3.3V) for the analog I/O and 1.8V for the core. As current requirements are pretty low, to keep the design simple and compact, the power supply has been designed with simple LM317 linear regulators and a series diode for the 2.5V.

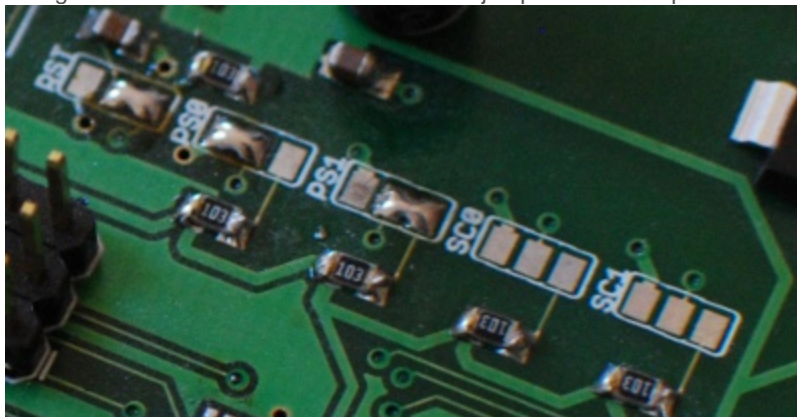
During the assembly of the PCB, the regulators have been changed to a mix of LD1117 and LD33 (as you can see from the photos), as I had these devices readily available and they fitted on the LM317 pads.



If you are redesigning the the power section, I suggest to use two LD1117 with a proper thermal pad, or a buck converter.

The switch IC has the basic configuration bits exposed as pinstrap on some 0R shortable jumper.

To use the switch in unmanaged mode just don't short any pinstrap, to use it in SPI salve mode so that it can be configured from the microcontroller short the jumpers as in the picture.



The switch IC is managed by the small Atmel ATmega168 microcontroller using the a TTL UART port on the host side, and an SPI port on the switch side.

Please note that to interface the TTL UART port with a standard RS232 port, you need a TTL-to-232 level shifter. The board has an [AVR910](#) compatible 2x3 2.54mm programming header for the microcontroller.

As for the Ethernet ports, there are five connectors with integrated magnetic. You can find compatible connectors from many manufacturers, just double check the integrated LEDs polarity (I designed for Pulse but found out that Halos are different).

Software

The firmware for the microcontroller implements a simple command-line interface to access the internal registers of the switch IC.

The interface can be used with a serial emulator application, like minicom, configured as 38400 bps, 8 data bit, 1 stop bit and no parity.

Available commands are:

- ▣ **help**: print available commands
- ▣ **rd offset count**: read registers
- ▣ **wr offset values...**: write registers
- ▣ **status**: print port status
- ▣ **sm_show and dm_show**: print internal MAC tables
- ▣ **vlan_show**: display current VLAN table
- ▣ **vlan_set addr valid membership fid vid**: set vlan entry
- ▣ **mib_show offset count**: show mib records



Refer to `commands.c` source file and switch IC documentation for further details.

Instructions to build and flash the firmware are provided in the README file in the repository.

If you need to store a specific configuration, you can hardcode that in the main function, before the "start switch" command, such as in the "start_mode_A()" function, disabled by default in the main.c file.