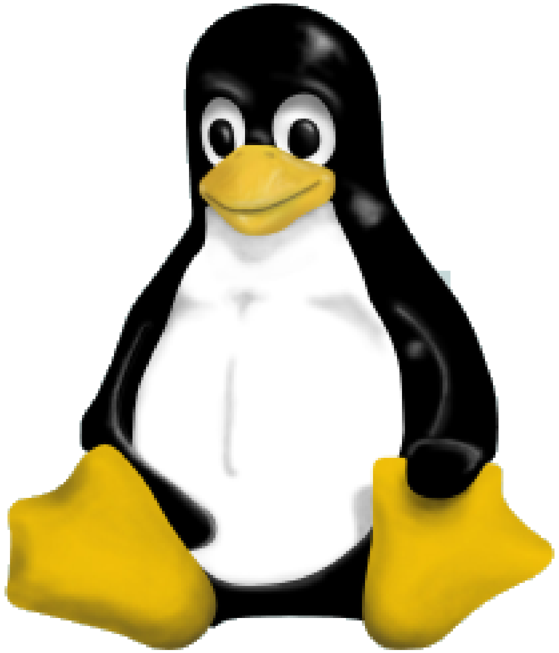
Linux Kernel Hacking Free Course, 3rd edition

Andrea Sarro
University of Rome “Tor Vergata”

HWMPS: Hardware Monitor & Protection System



April 5, 2006

Outline of the talk

- Project overview
- Developement phases and practical issues
- Hardware platform
- Microcontroller firmware
- Hardware management software
- Future plans

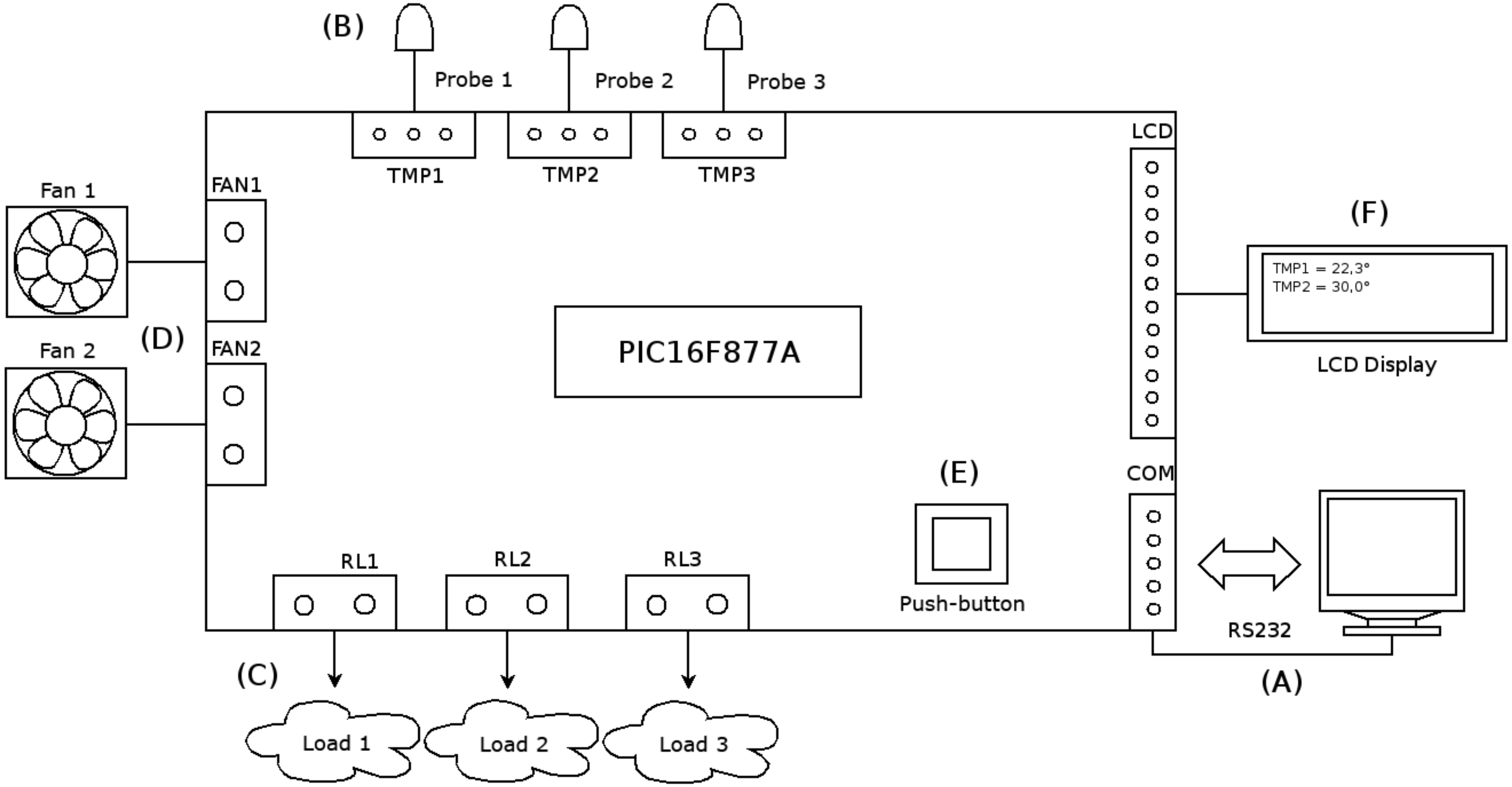
Project Overview

Project goals

- Realize a stand-alone hardware platform for protecting connected electrical devices and monitoring unpredictable and harmful operational states

- Realize a software platform able to:
  - Communicate with the hardware platform
  - Set operational parameters of the hardware platform
  - Receive status information about connected electrical devices

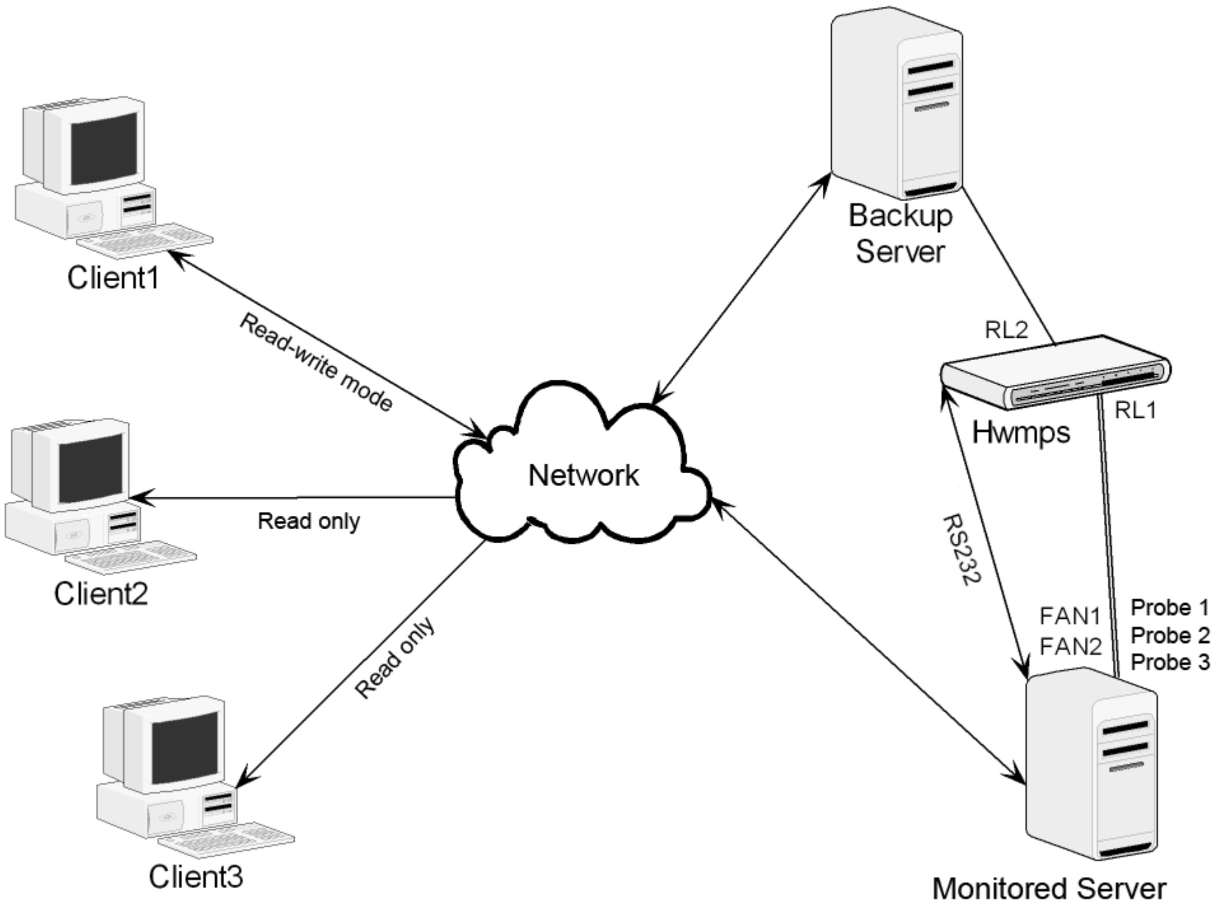
Hardware platform and interfaces



Hardware functions

- Monitoring functions:
  - Temperature readings from 3 thermal probes
  - Status visualization over a 2x16 LCD
  - Serial communication via RS232 (asynchronous)
- Protection and control functions:
  - Electrical devices control and protection via 3 realys
  - Warning messages via LCD and via software
  - Fan speed selection (2 channels)

Application background



Developement phases and practical issues

Developement phases

- Hardware developement:
  - Electrical schematic - EAGLE Schematic Editor
  - Printed circuit board - EAGLE Layout Editor
- Firmware developement:
  - PICMicro firmware - HI-TECH PICC + MPLAB IDE
  - Firmware simulation - MPLAB SIM
- Software developement:
  - Cross-platform RS232 Serial Library
  - Cross-platform client-server hardware management software

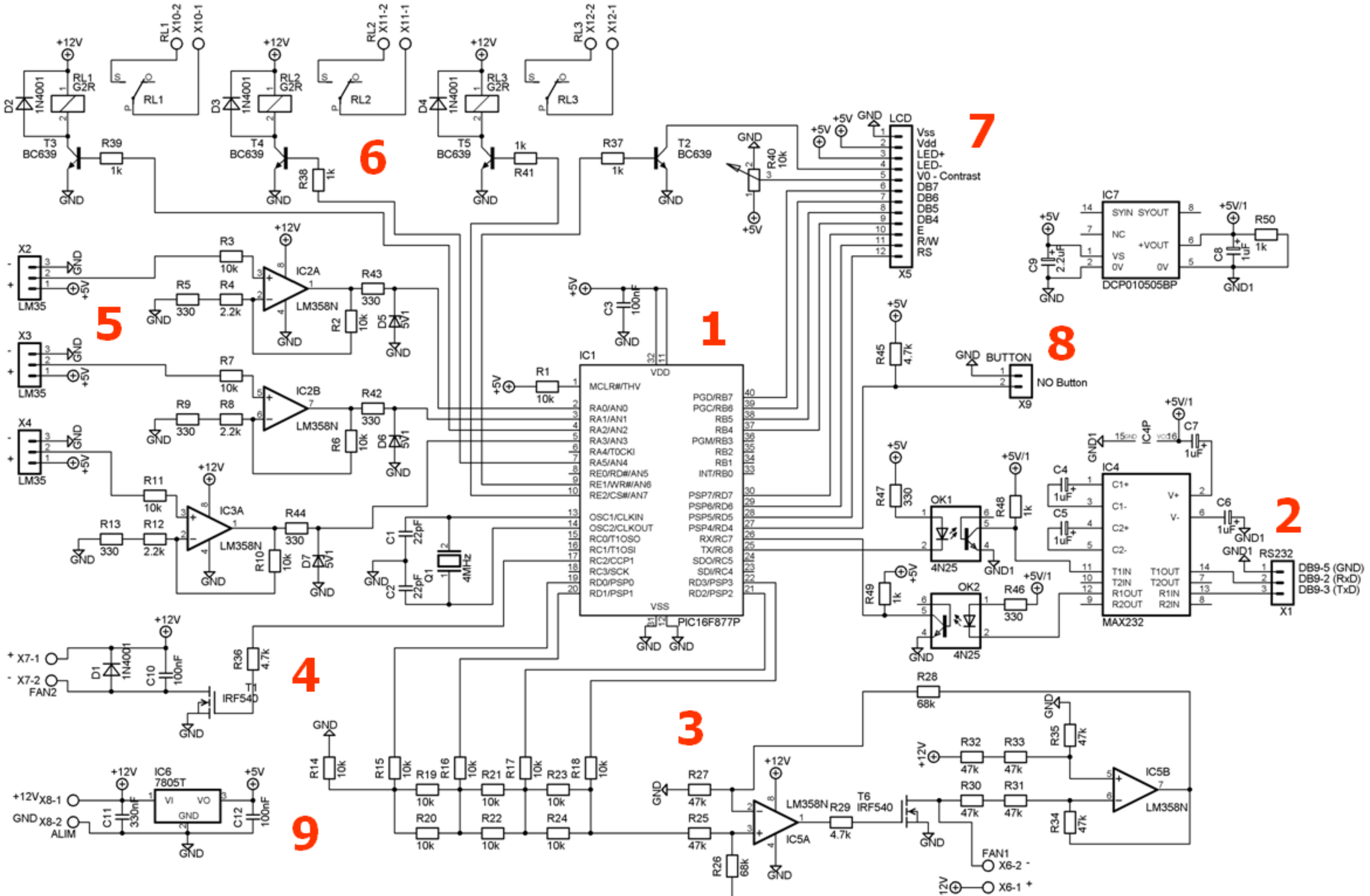
Practical issues

So what we need?

- Software:
  - Eagle 4.1: http://www.cadsoft.de
  - HI-TECH PICC C Compiler: http://www.htsoft.com
  - MPLAB IDE: http://www.microchip.com
  - GCC compiler + wxWidgets framework: http://www.wxwidgets.org
- Hardware:
  - PICMicro microcontroller and other electrical components
  - UV Light Box and a precoated Photoresist PCB laminate, tools for drilling and soldering
  - PICMicro hardware programmer, i.e. MPLAB-ICD2 (this is also an In Circuit Debugger)

Hardware platform

Electrical schematic



Rev. 1.0 RC3 - 19/06/2005

Electrical schematic - Modules

1. Microcontroller (PICmicro MCU)
2. RS232 opto-isolated interface
3. Digital-Analog Converter (DAC)
4. Pulse Width Modulation (PWM)
5. Thermal sensors
6. Relays
7. LCD display
8. Push button
9. Power supply

Bill of materials

The ”core” component is a Microchip PIC16F877A microcontroller

- To be operational it needs these components:
  - 4 MHz Quartz Xtal + two 22 uF ceramic capacitors
  - +5V/GND single power supply via LM7805 voltage regulator
  - Pull-up on MCLR pin via a 10 kOhm resistor

Bill of materials (cont'd)

For serial communication we use a MAX232 RS232 driver/receiver.

- For opto-isolation we also need:
  - DCP010505BP Isolated DC/DC converter (5V input / 5V unregulated output)
  - 4N35 optocoupler, one for RX line and one for TX line
  - Pull-ups and current limiting resistors for optocouplers, two low ESR capacitors for the DC/DC converter

Bill of materials (cont'd)

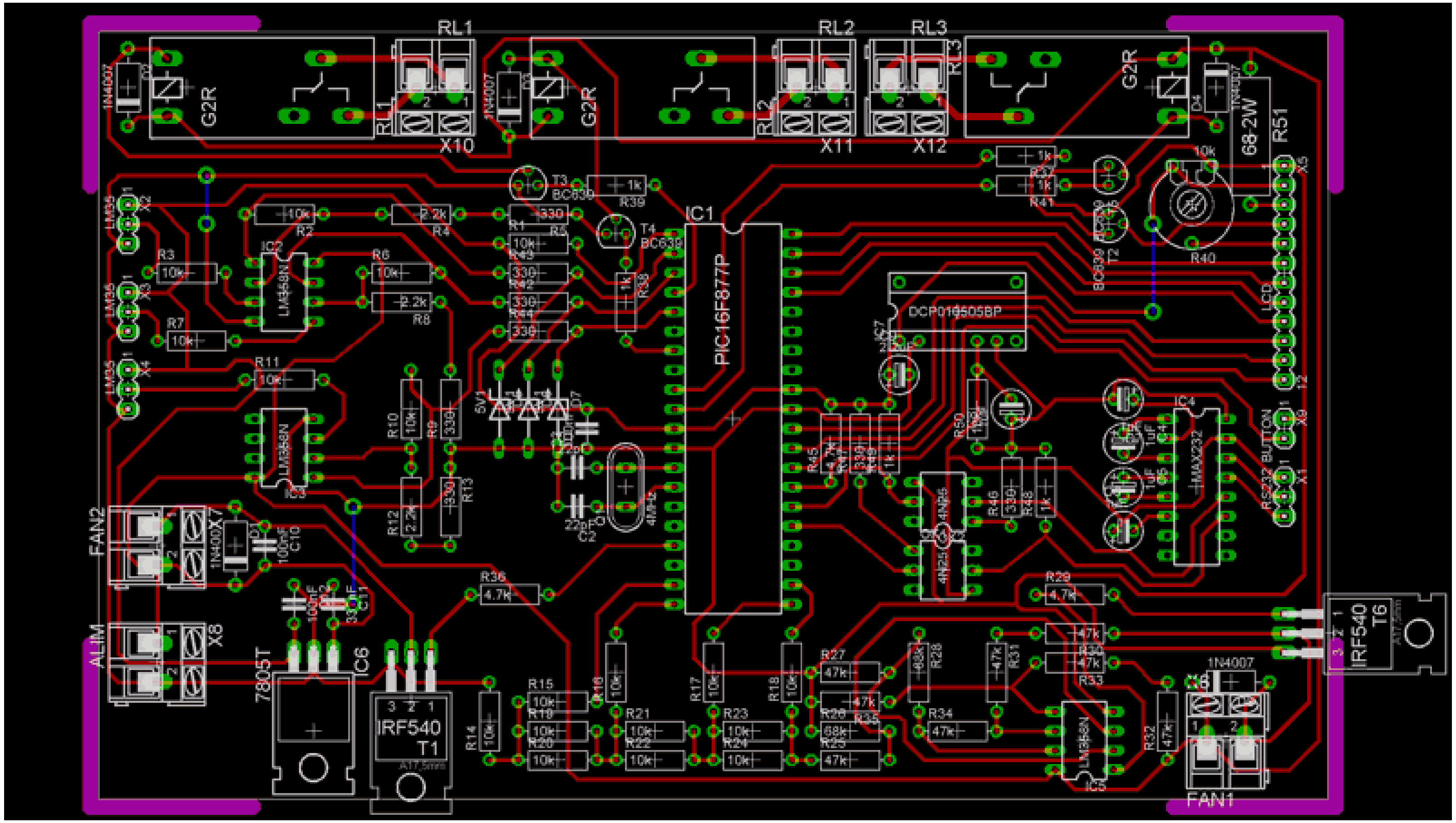
We also need these other components (oredered by their application field, omitting passive components like resistors)

- Thermal probing:
  - National Semiconductor's LM35 precision integrated-circuit sensors
  - LM358 operational amplifier
- Fan control:
  - IRF540 Power MOSFETs
  - LM358 operational amplifier

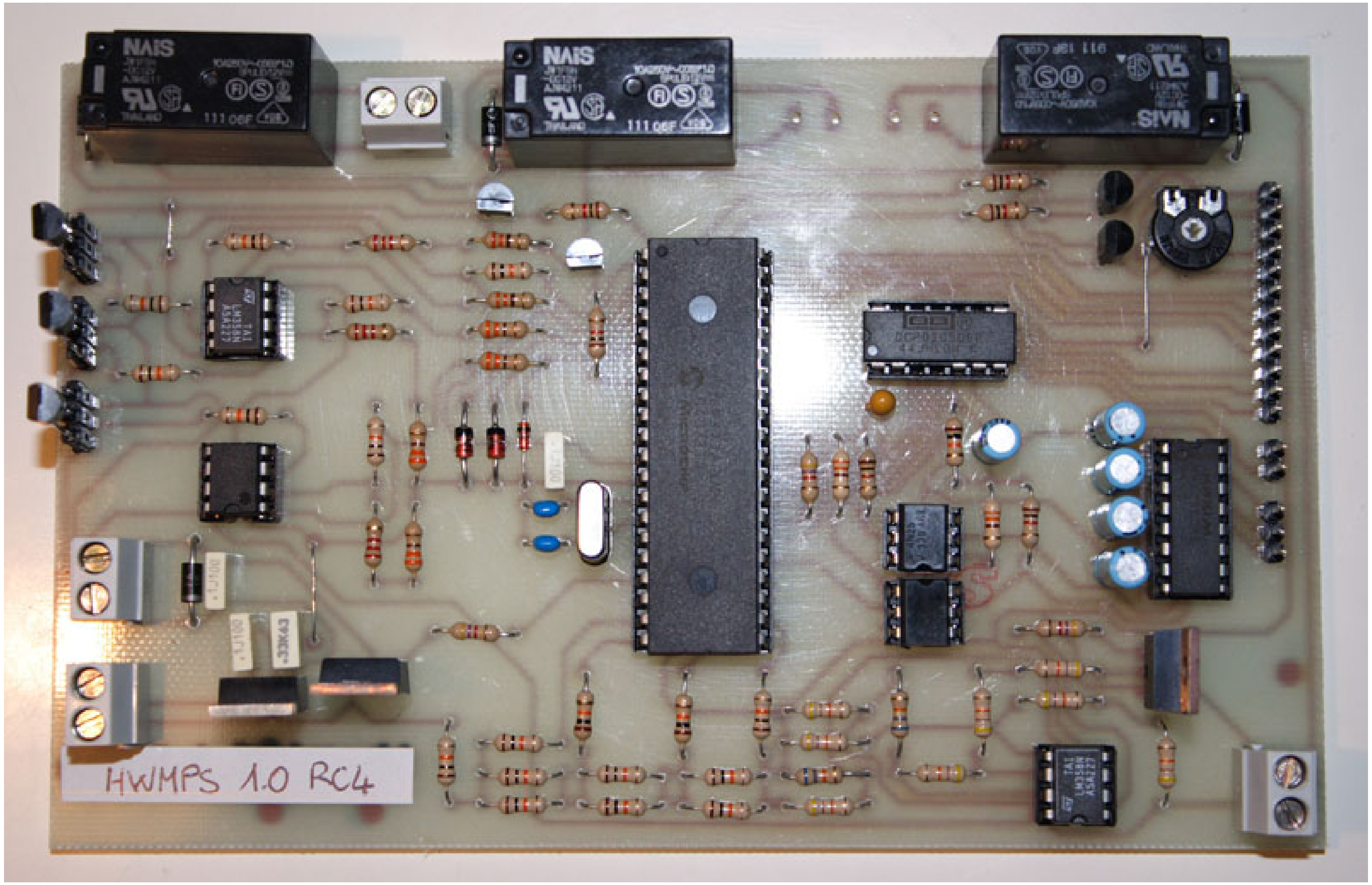
Bill of materials (cont'd)

- LCD Visualization:
  - Hitachi HD44780 based LCD or compatible (HD44780 clone)
  - BC639 NPN transistor for backlight ON/OFF software switching
- Electrical devices control:
  - 12V/230V-10A Relays
  - BC639 NPN transistors

Printed Circuit Board



HWMPS



Microcontroller firmware

Microcontroller

A microcontroller can be seen as: microprocessor + memory + peripherals

- Microchip PIC16F877A Mid-Range MCU:
  - 8 bit microcontroller available in PDIP/40 pin package
  - 8192 words of Program Memory
  - 368 byte of Data Memory (RAM)
  - 256 byte of EEPROM

- In this project we use these peripherals:
  - USART (Universal Synchronous Asynchronous Receiver Transmitter)
  - ADC (Analog/Digital Converter)
  - CCP Module (Capture/Compare/PWM)

Firmware

HWMPS Firmware has the role of realizing the whole platform features. It's written in C programming language.

- The firmware is composed of 4 modules (for more info see attached C sources):
  - Delay Library
  - LCD Library
  - Serial Library
  - (Real) Firmware

Firmware main roles

- Serial communication (via interrupts)
- Reading thermal sensors, converting 10 bit ADC values into human readable output
- Hardware protection and warning messages notification
- Fan speed thermoregulation
- LCD visualization
- Saving and loading settings from EEPROM

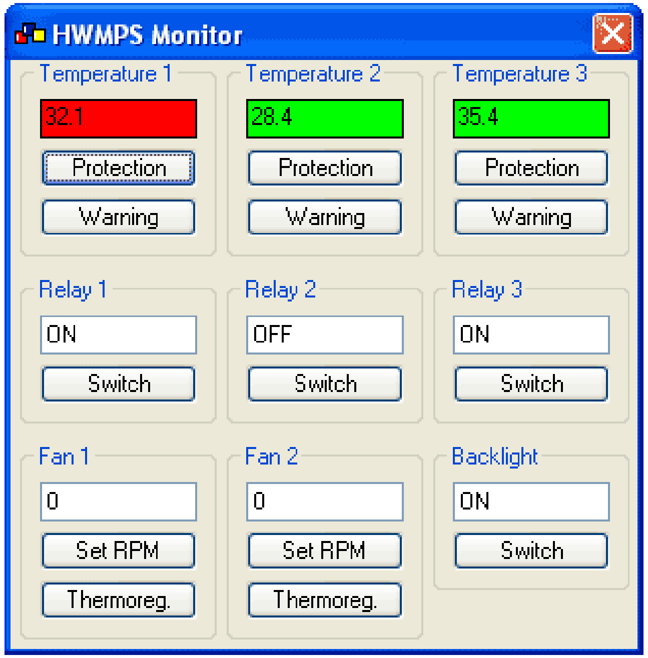
Hardware management software

HWMPS Serial Lib

- Cross-platform (Linux/Windows) library for serial communication
- It works in asynchronous - non blocking - mode
- Written in C++, it defines methods for:
  - Opening the serial port at a chosen baud-rate
  - Closing the serial port
  - Sending data
  - Receiving data in non blocking mode
  - Verifying serial port status

HWMPS Monitor

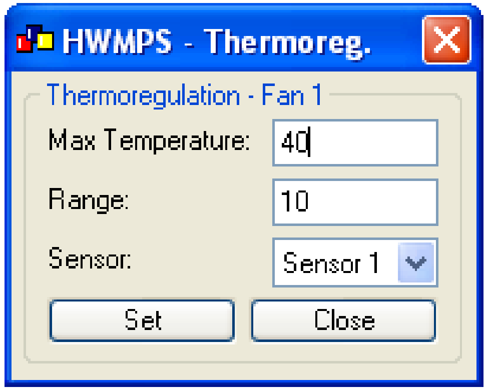
Cross-platform (Linux/Windows) software for HWMPS management. Written in C++ using wxWidgets library. It's client-server, so it's possible to monitor and control remote devices. The client interface on Windows looks like:



Temperature 1: 32.1
Temperature 2: 28.4
Temperature 3: 35.4

Relay 1: ON
Relay 2: OFF
Relay 3: ON

Fan 1: 0
Fan 2: 0
Backlight: ON



Thermoregulation - Fan 1

Max Temperature: 40
Range: 10
Sensor: Sensor 1

Future plans

Future plans and contacts

- Replacement of LM35 analog sensors with DS18B20 1-Wire digital sensors (in progress!)
- Ethernet interfacing for remote control (now using RS232)
- RTC for advanced time scheduling functions
- Custom boot-loader for on the fly firmware upgrade (via RS232)
- Project Web Site: http://hwmps.sourceforge.net