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

Outline of the talk

- Project overview

Outline of the talk

- Project overview
- Developement phases and practical issues

Outline of the talk

- Project overview
- Developement phases and practical issues
- Hardware platform

Outline of the talk

- Project overview
- Developement phases and practical issues
- Hardware platform
- Microcontroller firmware

Outline of the talk

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

Outline of the talk

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

Project Overview

Project goals

Project goals

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

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:

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

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

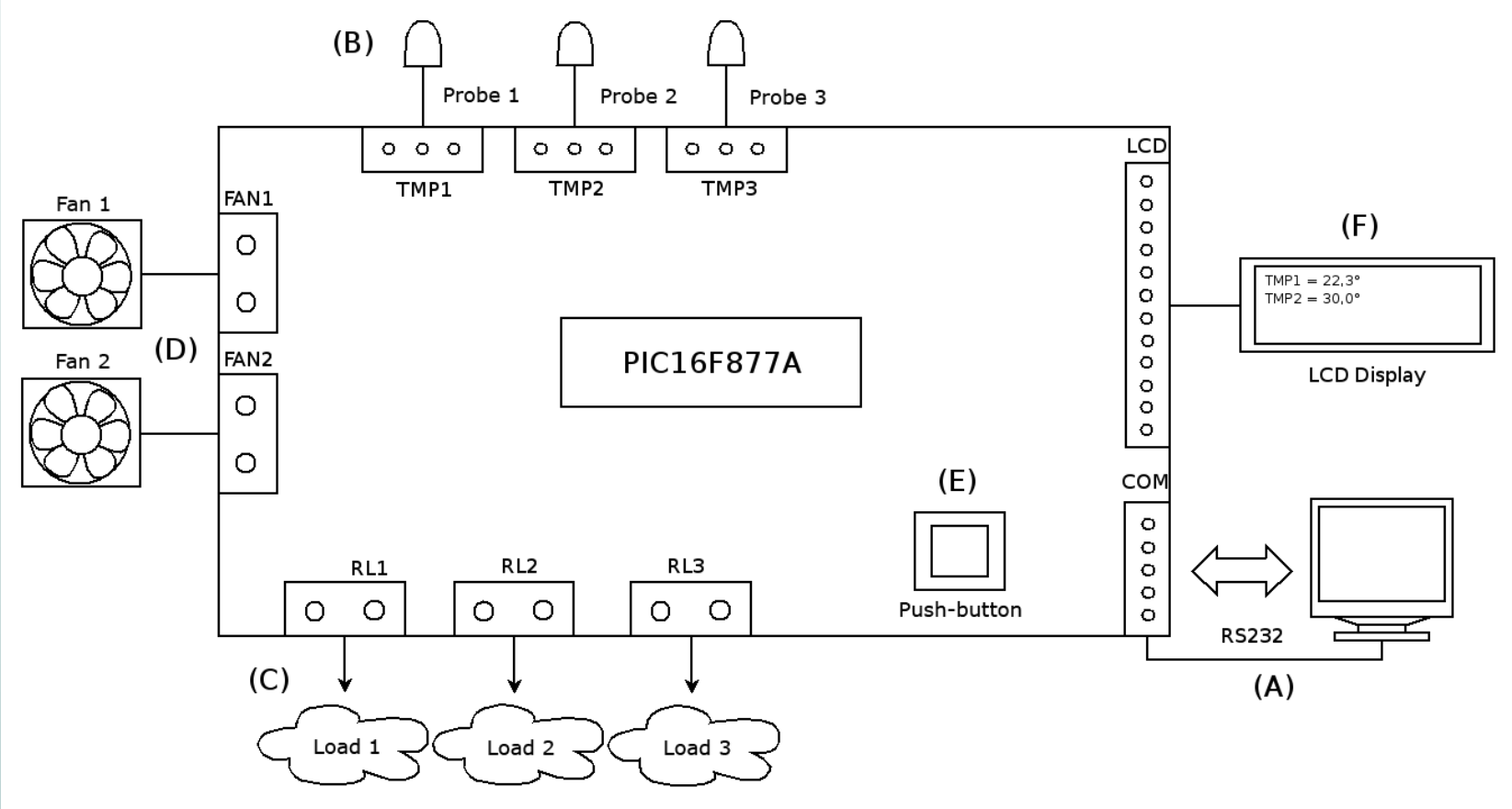
Project goals

- 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 platform and interfaces



Hardware functions

Hardware functions

- Monitoring functions:

Hardware functions

- Monitoring functions:
  - Temperature readings from 3 thermal probes

Hardware functions

- Monitoring functions:
  - Temperature readings from 3 thermal probes
  - Status visualization over a 2x16 LCD

Hardware functions

- Monitoring functions:
  - Temperature readings from 3 thermal probes
  - Status visualization over a 2x16 LCD
  - Serial communication via RS232 (asynchronous)

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:

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

Hardware functions

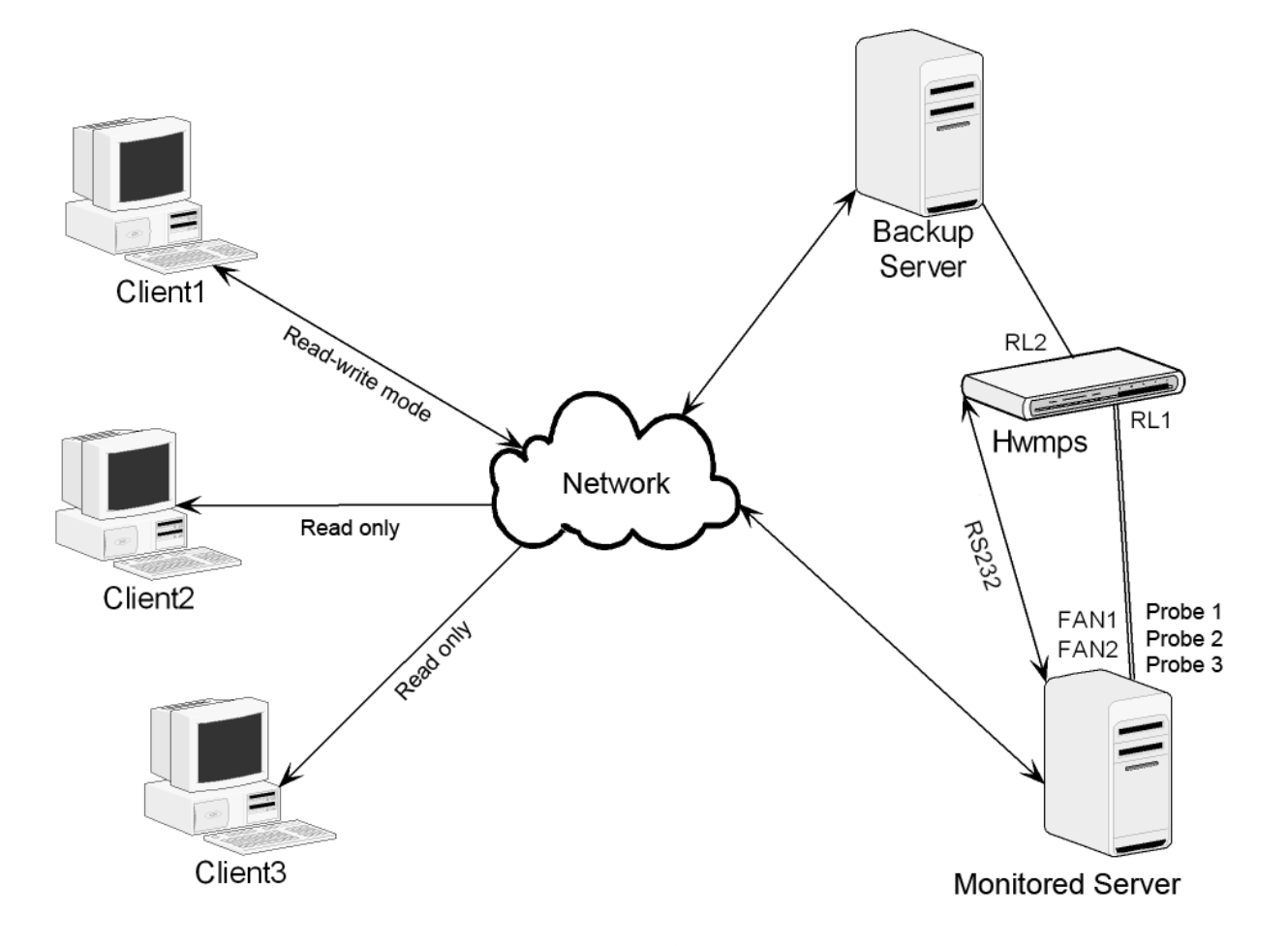
- Monitoring functions:
  - Temperature readings from 3 thermal probes
  - Status visualization over a 2x16 LCD
  - Serial communication via RS232 (asynchronous)
- Protection and control functions:
  - Electrical devices control and protection via 3 realys
  - Warning messages via LCD and via software

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

Application background



Developement phases and practical issues

Developement phases

Developement phases

- Hardware developement:

Developement phases

- Hardware developement:
  - Electrical schematic

Developement phases

- Hardware developement:
  - Electrical schematic - EAGLE Schematic Editor

Developement phases

- Hardware developement:
  - Electrical schematic - EAGLE Schematic Editor
  - Printed circuit board

Developement phases

- Hardware developement:
  - Electrical schematic - EAGLE Schematic Editor
  - Printed circuit board - EAGLE Layout Editor

Developement phases

- Hardware developement:
  - Electrical schematic - EAGLE Schematic Editor
  - Printed circuit board - EAGLE Layout Editor
- Firmware developement:

Developement phases

- Hardware developement:
  - Electrical schematic - EAGLE Schematic Editor
  - Printed circuit board - EAGLE Layout Editor
- Firmware developement:
  - PICMicro firmware

Developement phases

- Hardware developement:
  - Electrical schematic - EAGLE Schematic Editor
  - Printed circuit board - EAGLE Layout Editor
- Firmware developement:
  - PICMicro firmware - HI-TECH PICC + MPLAB IDE

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

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

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:

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

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

Practical issues

So what we need?

Practical issues

So what we need?

- Software:

Practical issues

So what we need?

- Software:
  - Eagle 4.1: http://www.cadsoft.de

Practical issues

So what we need?

- Software:
  - Eagle 4.1: http://www.cadsoft.de
  - HI-TECH PICC C Compiler: http://www.htsoft.com

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

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

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:

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

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

Practical issues

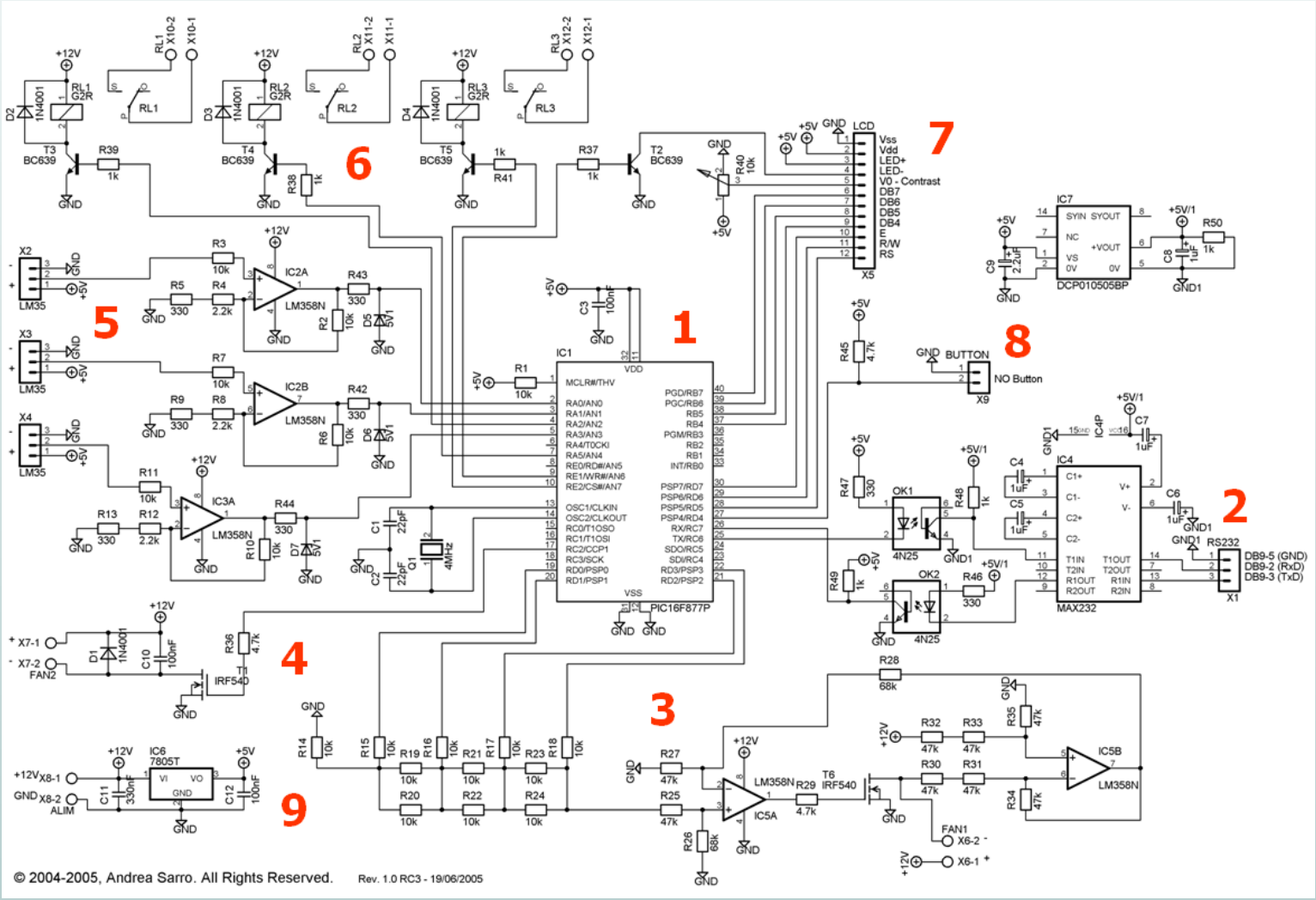
So what we need?

- 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

Electrical schematic



Electrical schematic - Modules

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

Bill of materials

The ”core” component is a Microchip PIC16F877A microcontroller

Bill of materials

The ”core” component is a Microchip PIC16F877A microcontroller

- To be operational it needs these components:

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

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

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)

Bill of materials (cont'd)

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

Bill of materials (cont'd)

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

- For opto-isolation we also need:

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)

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

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)

Bill of materials (cont'd)

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

Bill of materials (cont'd)

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

- Thermal probing:

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

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

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:

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

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)

Bill of materials (cont'd)

- LCD Visualization:

Bill of materials (cont'd)

- LCD Visualization:
  - Hitachi HD44780 based LCD or compatible (HD44780 clone)

Bill of materials (cont'd)

- LCD Visualization:
  - Hitachi HD44780 based LCD or compatible (HD44780 clone)
  - BC639 NPN transistor for backlight ON/OFF software switching

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:

Bill of materials (cont'd)

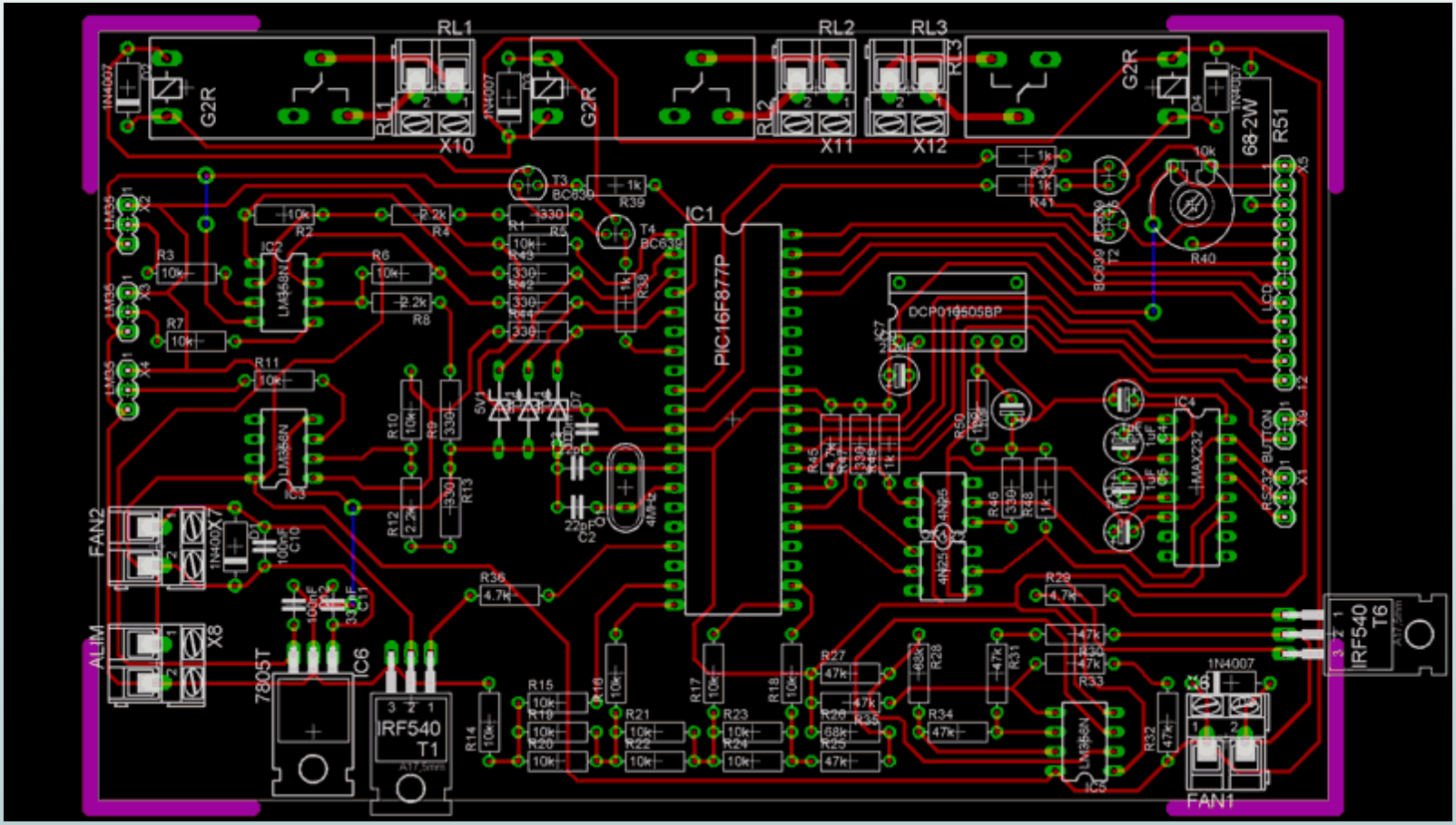
- LCD Visualization:
  - Hitachi HD44780 based LCD or compatible (HD44780 clone)
  - BC639 NPN transistor for backlight ON/OFF software switching
- Electrical devices control:
  - 12V/230V-10A Relays

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

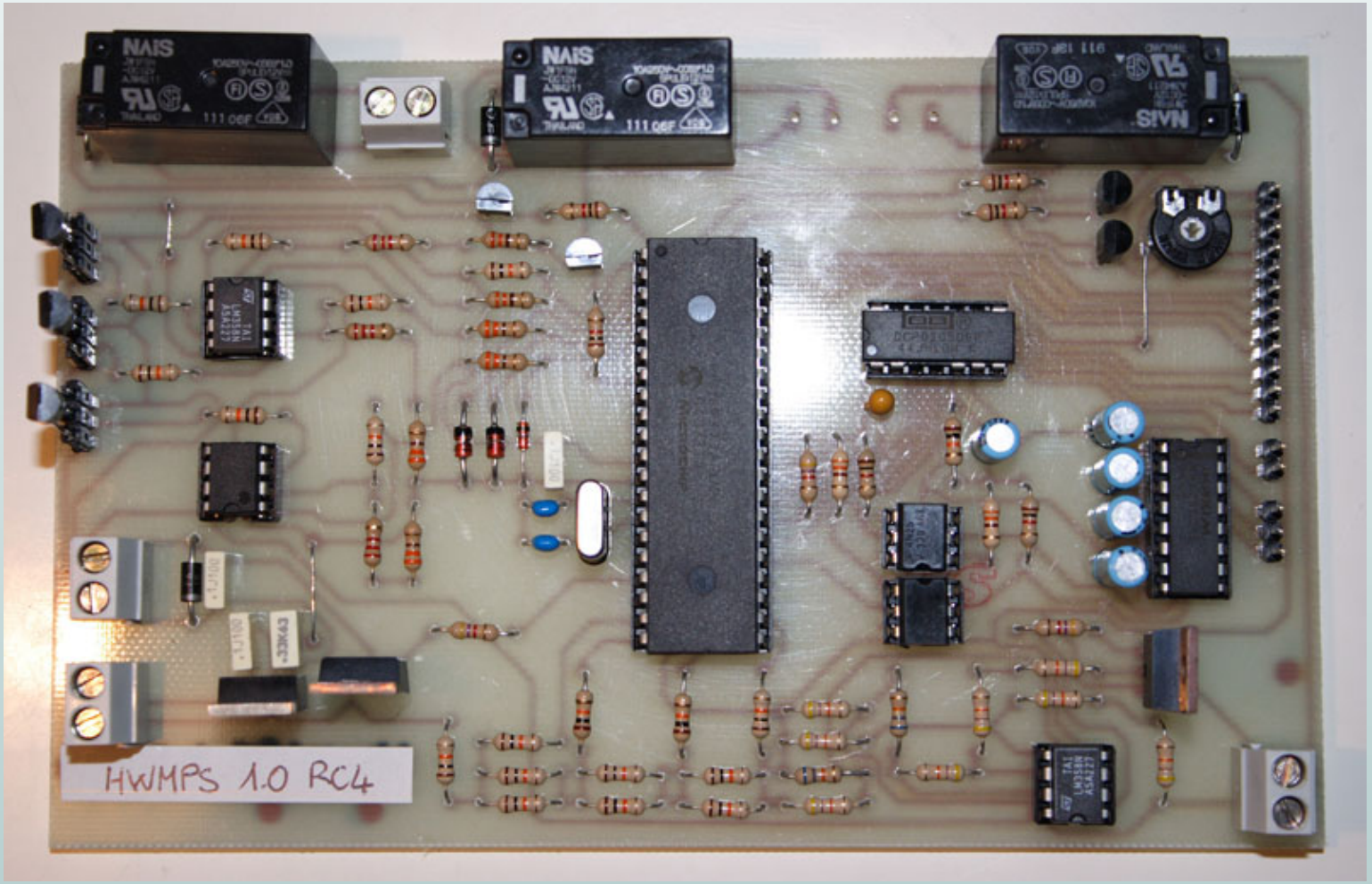
Printed Circuit Board

Printed Circuit Board



HWMPS

HWMPS



Microcontroller firmware

Microcontroller

Microcontroller

A microcontroller can be seen as:

Microcontroller

A microcontroller can be seen as: microprocessor

Microcontroller

A microcontroller can be seen as: microprocessor +

Microcontroller

A microcontroller can be seen as: microprocessor + memory

Microcontroller

A microcontroller can be seen as: microprocessor + memory +

Microcontroller

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

Microcontroller

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

- Microchip PIC16F877A Mid-Range MCU:

Microcontroller

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

- Microchip PIC16F877A Mid-Range MCU:
  - 8 bit microcontroller available in PDIP/40 pin package

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

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)

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

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:

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)

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)

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

Firmware

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

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):

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

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

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

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

Firmware main roles

- Serial communication (via interrupts)

Firmware main roles

- Serial communication (via interrupts)
- Reading thermal sensors, converting 10 bit ADC values into human readable output

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

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

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

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

HWMPS Serial Lib

- Cross-platform (Linux/Windows) library for serial communication

HWMPS Serial Lib

- Cross-platform (Linux/Windows) library for serial communication
- It works in asynchronous - non blocking - mode

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:

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

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

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

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

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

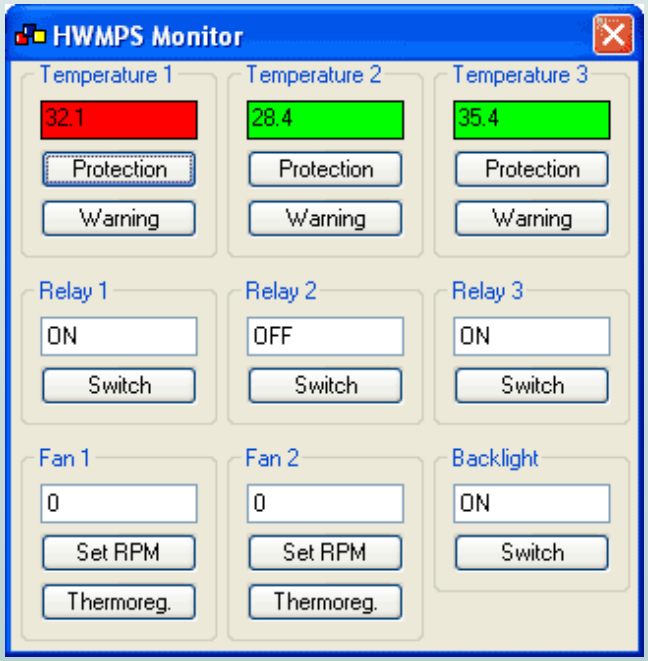
HWMPS Monitor

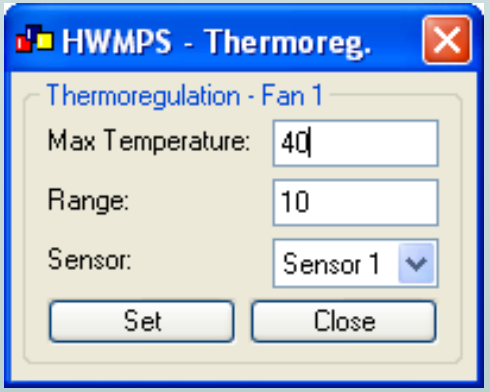
HWMPS Monitor

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.

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:





Future plans

Future plans and contacts

Future plans and contacts

- Replacement of LM35 analog sensors with DS18B20 1-Wire digital sensors (in progress!)

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)

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

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)

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