Stefano Benamati's previous experiences

Alarm system with graphic user interface.....2
Embedded system for exporting production parameters of industrial machines on the web.....3
Intelligent camera.....4
Embedded system to install in the side mirror for optical detection of overtaking cars.....5
User interface to display the output of a camera.....6
Interactive user interface for training, evaluation and validation of neural architectures based on the Totem processor.....7
Software infrastructure for an array of neural PC.....8
User interface for an information retrieval system based on neural technologies.....9

Project name	**Alarm system with graphic user interface**
Duration	from 04/04 to 08/04
Customer	MSLab / Microsystems / CFD
Working as	software developer (analyst and programmer)
Branch	security
References	
Technologies	Linux, C, C++, GNU, Nano-x, RS232, 8051, ARM

Project description

The system was composed by 2 main modules: a multimedia device based on Compulab´s ARMCore and an alarm device based on 8051 communicating through RS232.

My role was creating the routines for modules communication (using RS232), installing and configuring the environment for the multimedia device, implementing the code for handling/visualization of the user interface and realizing the user interface.

Tasks:

- Customized Linux kernel to work with our specific LCD display
- Customized Linux installation to work with our applications
- Created C++ classes for RS232 communication in a multi threaded environment.
- Defined Data Structures for inter threads communication.
- Created classes for easy thread management.
- Defined the architecture of the software
- Evaluated and chose the appropriate graphic libraries.
- Created the infrastructure to build the custom graphic user interface (class hierarchy)
- Implemented all the classes that define the widgets and screens of the interface.
- Implemented about 30% of the interface
- Created the functions for recording and playback vocal messages.
- Written the documentation.

Project name	**Embedded system for exporting production parameters of industrial machines on the web**
Duration	from 05/02 to 09/03
Customer	MSLab / Microsystems / Ilapak / HTP
Working as	software developer (analyst and programmer) / hardware debugger
Branch	industry automation
References	http://www.msares.it/CH_dettaglio.asp?id=45
Technologies	Linux, C, C++, GNU, Java, RS232, RS485, Modbus, Tektronix oscilloscope

Project description

The aim of this project was to create a small and cheap system to connect with industrial machines in order to remotize a part of its functionalities (read and modify parameters, read production parameters, upgrading the firmware, collecting statistics). The system was accessible through an ethernet connection (TCP/IP) or a RS232 (terminal emulation or PPP). The interface was web based.
The system was based on a Compulab´s 686Core running a Red Hat based Linux distribution and communicate with the machine through the Modbus protocol over an RS485 connection.

Tasks:

- Implemented the software to interface the industrial machine with the embedded system
- Debugged the hardware together with hardware developers.
- Defined test procedures
- Implemented test programs
- Implemented software for data collection, elaboration and presentation.
- Customized the software for two different industrial machines
- Written the documentation

Project name	**Intelligent camera**
Duration	from 02/02 to 05/02
Customer	Neuricam
Working as	software developer (analyst and programmer) / hardware debugger
Branch	artificial vision
References	www.neuricam.com (nc5100 ethercam)
Technologies	Linux, C++, Assembly, ARM7, GNU, Tektronix oscilloscope, RS232 sniffer, Flash memory.

Project description

The system was composed by a CMOS 320x256 logaritmic B/W sensor and an ARM7 processor. The operating system running on the camera was a customized Linux.
The camera was accessible through an ethernet connection using TCP/IP and was able to give the following services: FTP, Telnet and HTTP. Through some easy to use web pages it was possible to retrieve an image (plain or processed) and to modify camera parameters.

Tasks:

- Solved problem about Flash memory incompatibility.
- Implemented tools for downloading the firmware
- Implemented tools for flashing the firmware
- Implemented in assembly libraries for image processing
- Pipeline optimization for increasing system performances
- Created demo applications for the camera in C++ using the libraries
- Debugged the hardware together with hardware developers.

Project name	**Embedded system to install in the side mirror for optical detection of overtaking cars**
Duration	from 02/00 to 02/02
Customer	Neuricam / Ficosa
Working as	software developer (analyst and programmer) / hardware debugger / algorithms
Branch	automotive - safety - artificial vision
References	http://www.ficosa.com/eng/home_automocion.htm (follow the link “External mirrors” and then the banner “Overtake Detector” / ”Detector de Adelantamiento”)
Technologies	C++, Assembly (ARC), Totem code, Tektronix oscilloscope, Flash memory, EPP

Project description

It was an embedded system smaller than 3x3x3 cm for optical detection of overtaking cars. The heart of the system was a custom chip that includes a RISC processor, a neural processor and a 320x256 b/w logaritmic camera. This system was small enough to fit in the side mirror and was used to detect a car while it is overtaking. All the firmware for car detection was written in assembler for performance reasons.

The system was able to detect cars with more than 95% accurancy at 15fps, which I think is near to the limit without stereoscopic vision.

For a more detailed description of the system visit the website in the reference section.

Tasks:

- Contributed in the algorithms definition
- Implemented the firmware of the overtake detector 100% in assembly
- Pipeline optimization for increasing system performances
- Created the tools for interfacing the embedded system to a PC for diagnostic, programming and debugging purposes
- Created the tools for automatic performance evaluation based on a large database of videos.
- Created self diagnostic firmware
- Created the instruction vector for the test of the chip during the production
- Contributed to debug the chip by isolating, understanding and describing some errors in the camera sensor and in the neural processor.
- Debugged the hardware together with hardware developers.
- Proposed some new features to implement in the chip to increment system performances

Project name	**User interface to display the output of a camera**
Duration	from 12/99 to 02/00
Customer	Neuricam
Working as	software developer (analyst and programmer)
Branch	Vision
References	www.neuricam.com (NC1001 C1002)
Technologies	C ++, Borland Builder, DirectX, TCP/IP, EPP, Totem Code, TWAIN.

Project description

The user interface was implemented to give to Neuricam´s clients the opportunity to see how their cameras works, and, in addition, to give some code and programming examples to let them implement their own applications. For this reason were created (using Borland Builder) some components that permit to implement in a fast way an application. With these components it is possible to create a simple camera application just with a few mouse click and a few lines of code. Still now these components are the core of all the Neuricam's applications for their cameras.

Furthermore, within this project was created the software for TWAIN interface.

Tasks:

- Defined the graphic layout.
- Defined the classes.
- Written the code for fast image display using DirectX.
- Written the code for creating and modifying the palette.
- Written the code for recording and playback videos.
- Written the code for camera parameters setting.
- Implemented as a "Borland Builder Component" the software for retrieving the image from the camera (through a custom board, EPP, TCP/IP or emulation).
- Implemented as a "Borland Builder Component" the software for fast displaying the image from the camera (included image stretching/shrinking, false colors, mirror and rotation in real time).
- Implemented as a client/server architecture the code for remote control of the camera (parameter settings and image retrieving).
- Implemented the software for image retrieving through the TWAIN interface
- Written the documentation.

Project name	**Interactive user interface for training, evaluation and validation of neural architectures based on the Totem processor.**
Duration	from 05/99 to 12/99
Customer	Neuricam
Working as	software developer (analyst and programmer)
Branch	artificial intelligence
References	www.neuricam.com (WinTot32)
Technologies	C++, Totem Code, Borland Builder

Project description

The user interface was implemented to give to Neuricam´s clients the opportunity to see how their neural chips work, and in addition to have a fast and easy way to perform neural training. The software was able to deal with multiple neural chips (Totem) and perform parallel computations distributing the work on different Totems when available.
The software performs training of a user defined neural network through a set of examples and the RTS algorithm and was able to auto calculate the best parameters for the training.
Once the network is trained, the software can save it in binary format or as a C++ class that can be used in other programs. It is also available a software emulator for the neural processor.

Tasks:

- Defined the graphic layout.
- Defined the classes.
- Written the code for handling multiple neural processors.
- Written part of the code for the training subsystem.
- Written the subsystem for batch training.
- Implemented the editor and the compiler for neural architectures.
- Implemented the neural processor emulator
- Written the documentation.

Project name	**Software infrastructure for an array of neural PC**
Duration	from 03/99 to 05/99
Customer	”Università di Trento” and ”Istituto Nazionale di Fisica Nucleare (INFN)”
Working as	software developer (analyst and programmer)
Branch	distribute computing
References	
Technologies	C ++, Borland Builder, Totem Code, TCP/IP

Project description

For high energy event triggering was created a system based on an array of PC containing one or more neural coprocessors (Totem) connected by a fast Ethernet.

Tasks:

- Implemented the schema for load balance of the neural processors.

Project name	**User interface for an information retrieval system based on neural technologies**
Duration	from 01/99 to 03/99
Customer	“Università di Trento” and “Consiglio Nazionale delle Ricerche (CNR)”
Working as	software developer
Branch	
References	
Technologies	C++, Borland Builder.

Project description

As a research of the italian Consiglio Nazionale delle Ricerche (CNR), was created a new algorithm for information retrieval. A part of this research was to include neural technologies in the retrieval phase.

Tasks:

- Realized the user interface.