MScEE final year project:
Self tracking webcam
Project proposal

DEMAINE Benoît-Pierre

wed 28th May 2003

This document has been written for

1 Abstract

This paper will briefly explains how DEMAINE Benoît-Pierre intends to realize a self tracking camera, as being his final-year project.

2 General idea

The general idea of this project, is to built a self moving camera which will be able to track a moving object. The project will use 2 computer USB ports:

- one port will be dedicated to a webcam
- one port will be dedicated to an array of servo-motors which will move the webcam

A standard webcam driver will acquire the video stream. A smart Graphical User Interface will let the user track an object. The software will compute video filtering, and determine how to move the webcam in order to recenter the picture around the wanted object. Then a USB microchip will monitor the servos and recenter the field of the webcam on the desired object.

This project involves a lot of different skills:

- video processing and pattern recognition
- GUI design

- USB (a deep understanding)
- microchip and ASM language abilities
- electronic servo understanding

3 Main problems expected

The first problem was to find a USB capable microchip. Because very few are available, I choose the cheapest one : MC68HC908JB8. In order to make the prototype easy to build, I choose the model MC68HC908JB8JP available in DIP20 package, easily solderable on a standard strip board.

Then, I had to choose a development platform: I choose GNU/Linux because it offers a huge amount of open source software what will allow me to reuse the work of other people so as to build the video processing part. GUI design should not be very difficult under Linux since there are a lot of different GUI projects (PHP-qt, Perl, python-qt, GTK ...)

Programming the MC68HC908JB8 will require the development a new driver. The chip is sold by Motorola, two companies sell programming GUI for Windows98. Since this would imply the use of expensive and proprietary softwares, I decided to write my own development GUI and distribute it as a free software afterwords; for that I will have to design the GUI, and a default driver for the chip. This driver is not available in source form; thus I will have to write it from scratch. Once the default driver is written, I can program and test a home made firmware, this will enable the firmware to read the queries of the video processing programs, and monitor the servos.

Writing the default driver will require:

- USB1.0 understanding
- Linux security layers understanding
- perfect GNU-C99 knowledge
- full understanding of the 68HC908JB8 series
- compliant design of a software interface so that other people can re-use my driver for different work

After that, writing the second driver will be a light version of the primary driver. I will only have to modify the VID/PID of the device, and listen for commands from a different device file.

The chip will require two firmwares: one compliant with the other development tools from other companies, and one to monitor the servos. The second firmware will have to use some internal clock, and produce the appropriate square waves on I/O pins to tell the servo-motors how to move. I expect some difficulties about the use of the clock.

The last problem is to make everything work (chip+servos) **only consuming 500mA as peak current.**

4 Work done

I could contact some people how started to write the default driver, but stopped their work when they had to face the security-layers design of the Linux kernel. I can reuse some parts of their work.

I have already read 60% of the technical documentation of the 68HC908JB8, and drawn some schematics of a prototype board using the DIP20 package.

I don't worry at all about the GUIs, nor the video processing, since the xawtv project is available from source, and provide a fluent video stream. If working on the stream is too difficult, or requires too much computing, I can reduce the video processing by using picture processing. In that case I can use some other program to provide me with shots from the webcam. Since most GNU projects are available from source, I know that will be easy.

I already wrote to Motorola to ask them some more technical specifications about the chip; I am waiting for their response.

I also downloaded the free monitor program for MC68HC08 series from Pemicro website (http://www.pemicron.com). That software provides a free source code firmware for the chip. I will mix that code with some other ones to produce a specific firmware for my prototype board.

A friend of mine already designed the webcam support, with the servo control; I only have to fit it with my own webcam, and check that the servos do not consume more than 500mA.

An other company offered to provide me some boards to help me developing the default driver, and contribute to the driver development.