

Re: webcamasm

Source: <http://coding.derkeiler.com/Archive/Assembler/alt.lang.asm/2007-06/msg00887.html>

- *From:* JGCASEY <jgkcasey@xxxxxxxxxxxx>
 - *Date:* Sun, 24 Jun 2007 04:03:29 -0700
-

On Jun 24, 5:01 pm, Frank Kotler <fbkot...@xxxxxxxxxxxx> wrote:

I thought we'd get this out of Betov's Twelve Lessons thread...

John was saying...

I have seen C++ programmers struggle with the documentation! One thing I have always wanted to do was access webcam data in real time. Have a look at the efforts by C++ and C# programmers to see what a convoluted dogs breakfast that is. And even they don't explain how they did what they manage to do and some complain how hard it all was.

Frank:

=====

But, but, but... I read in the glossy brochure that it was *easy* with HLLs! :)

The last time you and I discussed this webcam issue, I promised to borrow a camera from a roommate and see what I could figure out. What I remember fooling with, some time ago, plugged into the parallel port.

John:

=====

The old Connectix ccd camera connected via the parallel port. Using the protocol specs provided by Connectix I wrote my own capture routines in QBasic, C and Assembler. I only had the monochrome model but it was a great little camera (I have two of them). Others have used them on little robots.

Since Logitech bought Connectix out they no longer make these models. I last used them with WindowsME on a computer that still had the parallel port. Now computers only come with usb ports etc. I just wanted to try the color webcams. I have done some experiments with Visual Basic, but it was too slow for real

Re: webcamasm

time processing, and a vc++ shell which was great in terms of processing speed but only captured about 3 frames per second.

Frank:

=====

I couldn't get as far as making the led blink with that one (from dos). What he came up with was usb connected. No usb ports in my old K6. :((I don't remember if I ever posted the end of that story, or just left you hanging... (I've got Adult Attention Deficit Hypo-activity Disorder, I think :)

However, that machine died, and this one *does* have usb! :) So I reacquired the camera. It is a "3Com Homeconnect" – apparently the same as "Vicom", which the old one was. First time I plugged anything into the usb port! (I *knew* there was something besides "booting from usb" that I wanted to try!)

So, armed with a hint from the webcam "howto", I looked in the output of "dmesg" for any mention of it. Sure enough, the camera was identified (as "Vicom") and is on /dev/video0. Okay... opening and reading that file turns the led on (That was easy!!!), and gives me "some bytes"!

Turning these bytes into an image isn't going quite as well. I'm closing in on it – I'm seeing 320*240*24, apparently. But not what I'd call "success" yet. I don't feel too bad about it – "xcam" and "xawtv" don't give a great image, either (better'n I've got!). "xawtv" complains that my X server doesn't have DGA support(???). I may need to upgrade software... or even RTFM! :)

But this won't help you. You do it entirely differently in Windows. According to that "fasmcam" demo, you use "avicap" – I suppose that's "avicap32.dll"? Do I understand you're getting images in a dialog-box? (and you're *complaining*???)

John:

=====

The idea is to get video data into memory where you can do what _you_ want to do with the video data. A program that just shows how to make the magical (unexplained) incantations to display a webcam image is pointless.

Frank:

Re: webcamasm

Re: webcamasm

=====

If you want to post that program here and "rip it to shreds"...
I don't know Windows, but those who do can probably help you
get it to do something different – image in its own window,
or whatever...

John:

=====

This might give you an idea of the "other stuff" that was of
interest to me ...

<http://www.generation5.org/content/2005/gpv.asp>

There is plenty of stuff out there but it requires you to
be proficient in c++ and windows programming. There are also
examples for c# and Java and of course visual Basic. I have
played with all these and have found the FreeBasic the best
option for my limited abilities, time and interests.

Frank:

=====

That's *if* you want to persue it in asm. To be honest, it
isn't going to make your cam run any faster. In terms of
manipulating the data between acquiring it from the camera
and displaying it (or whatever), there *may* be some place
where asm could make a difference – I suppose that's what
those mmx instructions are good for(?). Mostly, I think a
Basic app is going to work as well.

Some of us think asm is "fun"...

John:

=====

Yes I found it fun. But fun to do what Frank?

All this is not new, I first broached the subject back Aug 19 2005,
with the question, "what are your applications?" for which there
were three replies,

randy:

Device drivers, nuclear reactor console control software, USB device
drivers and protocol stacks, Embedded controller applications (e.g.,
measuring the contours of the bottom of a foot in order to build
custom
shoes), traffic controllers (those boxes you see on street corners),
and, of course, program development tools.

f0dder:

Re: webcamasm

Re: webcamasm

I've worked on a lot of miscellaneous things. TCP/IP networked stuff, toyed with operating system development, kernel mode drivers for windows (not for hardware yet, though), graphics programming (NOTHING like the cool stuff you can find over at <http://www.pouet.net> , though!), software protection (there's a couple of shareware apps out there protected by me), simplistic virtual machines (including an assembler-in-the-works for one of them), various tools and toys for my own use, a couple of in-house tools for the museum where I work, et cetera.

wolfgang kern:

My main fields are technical applications used in handicapped support and medical science/research.

Another smaller (but more regular) part are industrial production control (ie: automated pharmacy/food packaging).

Most of my solutions imply some hardware design, even lately just PC add-on cards.

And for my own pleasure I abuse my PCs as measurement-tools like a storing oscilloscope using the joy-stick input, frequency counters/generators on the LPT-connectors, and some more uncommon but funny stuff.

Frank Kotler:

???

Best,
John

Re: webcamasm