

# Low Cost Four-Channel Network Video Server

*Zilog 2004 Flash Nets Cash Design Contest (Project Entry eZ2975)*

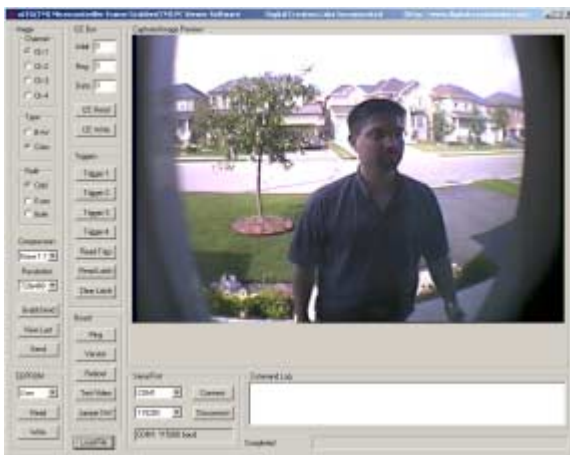
Ever wanted to setup a network of IP surveillance cameras around your house or cottage? If so, you have probably come to the realization that the commercial options available out there are simply too expensive for the average home user. For example, a single standalone IP camera can cost anywhere between 200\$ and 2000\$ depending on features. These costs quickly escalate when you need 4 or 5 cameras to cover a property. Part of the reason for this high cost is the inherent requirement to replicate the entire web server circuitry in each of the IP cameras.



*Figure 1: IP network video server allows the digitization of up to 4 channels of analog NTSC video from off-the-shelf CCTV cameras and serving out the data onto an 10/100BaseT Ethernet network.*

By contrast, conventional CCTV cameras, which put out analog NTSC signals, have become commodity items due to the exploding video surveillance market and are available for a mere fraction of their IP-enabled counterparts. The only thing missing is the ability to take the analog NTSC signal, digitize it, and then serve it out over an Ethernet interface. This is precisely the function of the network video server described in this project, which was realized for a few hundred dollars! Moreover, the video server was designed to accept inputs from up to 4 analog cameras to further reduce the per channel cost.

The Zilog eZ80F91 Acclaim! MCU was used at the heart of the system and is responsible for interfacing to a commercial OEM frame grabber module with a serial UART interface using an ASCII command set. The Zilog MCU is also responsible for detecting external triggers from a number of PIR motion sensors. It also buffers any captured images and serves out (pushes) the images via the TFTP protocol over a 10/100BaseT Ethernet LAN to a remote file server. The Zilog MCU also generates unique filenames for the image transfers, and can be remotely triggered via a custom TELNET command set.



*Figure 2: The images "pushed" by the video server through the LAN are collected on the hard drive of a remote PC running a TFTP server. The acquired images can then be reviewed with this image browser software.*

A remote PC running a TFTP file server gathers all the images which are uniquely identified and time-stamped on its hard drive for future review using special viewing software. System network access permissions can also be customized for added network security. The low cost nature of the system allows users to install a number of inexpensive CCTV cameras in their homes, offices, warehouses, and network-enable them by using the video server. Remote access is of course possible through the Internet, which allows remote monitoring of premises through a distant location. This can turn out to be fairly handy to keep an eye on the cottage during the winter months...

The low cost eZ80Acclaim! 8 bit MCU, as well as the free ZTP network stack, proved to be the key to the simplicity, ease of development, and low cost nature of this project.

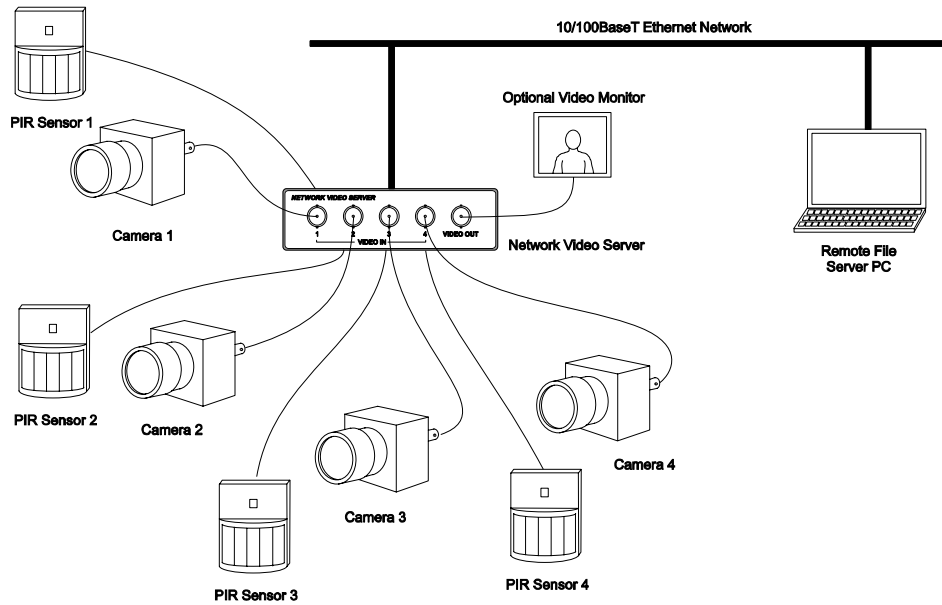


Figure 3: High-level architecture of a network video server system. 4 conventional NTSC video cameras connect directly to the video server. The server web-enables the video cameras and provides a standard Ethernet network interface. The images are transferred over the network to a central image repository. Optional triggers may also be used to trigger image capture.

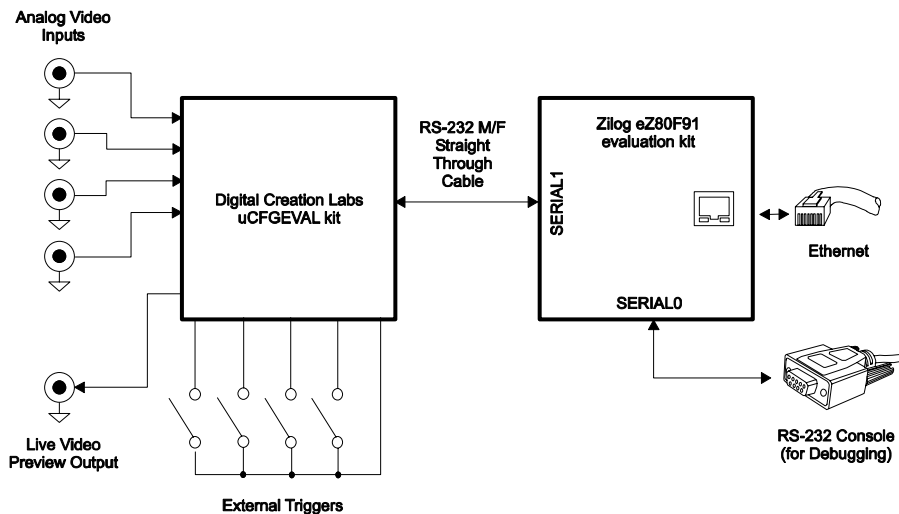


Figure 4: The overall network video server circuit simply involves interfacing a uCFGEVAL kit with the eZ80F91 evaluation board through an RS-232 straight-through cable...couldn't be simpler! Don't be deceived by the apparent simplicity, a lot is going on inside each of these modules... External trigger inputs are provided to trigger the uCFGEVAL by PIR intrusion sensors.

```

video_server_init();           // Initialize the video server
soft_trigger = 0;             // No soft triggers issued
for (;;)                       // Main thread
{
    if (soft_trigger)          // If a user issued a soft trigger command
    {                           // through telnet
        ucfg_soft_trigger(soft_trigger); // Send soft trigger command

        kprintf("Soft trigger command issued\n");

        KE_TaskSleep10(1);      // Sleep 100 msecs to give frame
                                // grabber a chance to grab a frame
        soft_trigger = 0;        // Clear soft trigger
    }

    if (video_server_triggered()) // If a trigger on uCFG was latched
    {
        kprintf("Trigger latch: %d\n", trigger);
                                // Transfer image from uCFG to local mem
        video_server_buffer_ucfg_image();
                                // Generate a sequential filename
        video_server_generate_filename(filename);
                                // Push image to TFTP repository server
        video_server_push_image_tftp(filename);

        ucfg_clear_trigger_latch(); // Clear trigger latch to re-enable uCFG
    }

    KE_TaskSleep(1);           // Sleep for 1 second
}

```

*Listing 1: Main thread of execution of the network video server. First, a check is performed to see if a soft trigger request was issued over the TELNET interface. The trigger latch on the uCFG is then polled for activity. If triggers are detected, the image is buffered locally and then sent to the remote file server via TFTP.*