IT 08 016

Examensarbete 30 hp
Maj 2008

Video recording in mobile military systems

William Ekström

Institutionen för informationsteknologi
Department of Information Technology

UPPSALA
UNIVERSITET

Teknisk- naturvetenskaplig fakultet
UTH-enheten

Besöksadress:
Ångströmlaboratoriet
Lägerhyddsvägen 1
Hus 4, Plan 0

Postadress:
Box 536
751 21 Uppsala

Telefon:
018 – 471 30 03

Telefax:
018 – 471 30 00

Hemsida:
http://www.teknat.uu.se/student

Abstract

Video recording in mobile military systems

William Ekström

In this paper different approaches for digital video recording in vehicles are examined, with weight on video quality. Different video compression algorithms were gone through and the conclusion was that inter-frame compression, like MPEG video delivers better video quality than intra-frame algorithms, for example Wavelet. The main drawback is that it is difficult to extract single frames, which leads to difficulties when for example backwards-playability is needed.
To demonstrate how digital video recording can be done a demonstrator was built. Therefore, when the decision to use MPEG compression was made, a market research was done and a video grabber expansion card for PCs was bought. The report describes the hardware and software developments that had to be made to run the video recorder.
The result of the thesis was this report and a functioning recorder. The delivered video quality was fair and it was also remote controlled.

Handledare: Fredrik Wahlström
Ämnesgranskare: Bo Nordin
Examinator: Anders Jansson
IT 08 016
Sponsor: CC Systems

Tryckt av: Reprocentralen ITC

Sammanfattning

Militära uppdrag behöver alltid dokumenteras noggrant och att kunna samla bevismaterial för händelser är en viktigt. Därför är det intressant att kunna ha möjligheten att enkelt spela in videosekvenser.

I denna rapport undersöks olika sätt att spela in videofilm med från ett militärt fordon. Målet är att kunna montera en videoinspelningsenhet på ett skyddat ställe i fordonet och mata det med en analog videosignal för inspelning på lämpligt medium. Inspelningsenheten ska kunna fjärrstyras med fordonets inbyggda nätverk.

Metoder för att komprimera videoinformationen gås igenom och olika hårdvarulösningar diskuteras för att utröna vad som kan vara lämpligt för applikationen. Hänsyn måste tas till den utsatta miljö det innebär med temperaturvariationer och vibrationer.

Under arbetets gång tillverkades en demonstrationsenhet som komponerades ihop baserat på resultatet i den undersökande delen av rapporten. Enheten fjärrstyrs genom ett CAN-nätverk, vilket är en typ av nätverk designat för bruk inom fordonsindustrin. Demonstrationsenheten fungerande tillfredställande och videokvaliteten var god.

Contents

Introduction

When using a military vehicle the crew has to do many different tasks. They have to navigate, drive in difficult terrain and shoot at different types of targets. The targeting systems are these days very complex and the goal is to make it easy for the gunner to hit the targets with good precision. But even then things can go wrong. Sometimes a target can be missed without anyone knowing what went wrong. Was it the machine or was it the gunner that made a mistake. Did it hit to the left or to the right of the target?

Today war is very much about diplomacy. It is about convincing the enemy that they are inferior and giving proofs to the public that the military actions are made in a human way and that only strategic targets are hit. If anyone claim that civilians were attacked it is convenient to the government if proofs of what really happened can be delivered to the press.

The crew of a military vehicle has many things to think about in a combat situation. The enemy is doing its best to hide. Sometimes the gunner may just get a glance of something moving up front but can’t shoot because he is not sure if a target is a friend or a foe.

In all these situations it would be very useful to be able to record what really happened. After a training event, the instructor can look at the video and decide what did go wrong and give instructions to the gunner how to correct the errors. Evidence of what target that were hit can be broadcast on television, like the Americans did during the gulf war. Everyone has probably seen the videos of planes dropping bombs at the Iraqis. The gunner who could not judge if shooting the target up front is the right thing to do can rewind the video and look at it again, in slow motion if needed.

Considering the arguments above, it would be interesting to investigate the possibilities of installing a recording device in military tracked vehicles. It is a task with many different aspects to look at, i.e. how to integrate it into a current system, and reuse of as much equipment as possible. Which methods can be used? Is tape recording better than digital recording? Should the video data be compressed? How can digital equipment survive in such a harsh environment?

All these things will be looked upon in this report. A fully functional recording device, the demonstrator, will be built using the techniques discussed in the report. During the development of the demonstrator all steps taken will be noted in the report and followed by results.

Chapter 1 Background

On a military vehicle like the CV90, the crew consists of several persons: the most important are the captain, the gunner and the driver. Each of the crewmembers has a computer monitor on which they can watch one of a set of cameras on the outside of the vehicle. Moreover, each of the crewmembers is able to independently select which camera to watch. But they are not able to record anything. To enable this, a recording device must be installed.

The main issue in this report is to examine how a recording device could be made. It will cover both hardware and software aspects, which includes that a market analysis has been made and a demonstrator will be built. The demonstrator will consist of consumer products together with components made at CC Systems AB.

The market survey and the demonstrator will be targeted at a specific functionality, which is requested by the manufacturer of the vehicle. The functionality includes ability to:

- **Record video**
- **Replay video**
- **Store different video clips**
- **Jump in a video sequence**
- **Replay in slow motion**
- **Export data to an external device, like a portable hard disk**

Aside from these abilities, it should be possible to install the recorder anywhere in the vehicle, thus some kind of remote control is needed.

Although the demonstrator preferably will assemble a real recorder unit for a military vehicle, it will only be a bench test. It will never really be installed in a vehicle and does not need to be inside any black box. The reason for building a demonstrator is to show one way of creating such a device.

Chapter 2 Environment

2.1 Integrating into current system

2.1.1 CAN Network

The video recorder is supposed to be installed in a track vehicle with a crew of at least three, plus passengers. Each of the crewmembers has access to one PC and the passengers share one. All these computers are connected to a CAN network, to enable control of different devices and to transmit information like vehicle status in the vehicle.

The vehicle also contains a system of cameras, which produces analog video streams, which are fed to video switches. The switches are connected to the CAN network, which enables each PC user to independently choose which camera to watch.

To enable watching analog video on the PCs' video cards are installed in them. The video can then be watched in a window at the PC screen. At the same time control buttons, accessed by soft keys, is visible below the video window. The video switches are connected to the CAN network and controlled completely by software.

2.1.2 PC's to Control the Recorder

The preferable way of designing a video recording system is to make it as similar to the current systems as possible. Controlling the recorder should be done on the PC, just as the choice of camera is done today. The PC's could act as thin clients to access the Recorder, which may consist of a black box installable anywhere in the vehicle.

2.1.3 PC as Recorder

The preferences above calls for a CAN controlled video recorder. It should neither have a keyboard nor a screen. It ought to be a so-called black box type of component. One solution for designing a video recorder like this is to make it a PC. It is relatively easy to enable a PC to communicate on the

CAN bus due to the many CAN cards available. The fact that the manufacturer of the vehicle already uses PCs with CAN cards, is also an argument for this solution.

To perform the actual recording, a video capture card and a mass storage can be used. The capture card converts analog video signal into digital video, preferably compressed and stores it on some mass storage unit. To enable replay of the video clips, a digital to analog converter card is needed as well. The system is to be kept as simple as possible, thus a video card that can both record and replay is advantageous.

When recording, the recorder should act just as a PC, connected to the output of the video switch. This way it is possible to choose camera independently or switch between cameras during recording. When replaying, the recorder should act as a camera, connected to the input of the video switch. This will make it possible for any other onboard PC to display the video clips.

2.1.4 Operating System

The current operating system used in the vehicle is Windows based and to make the video recorder easier to manage it is good to use a similar operating system in it. This way the same development tools can be used for all computers onboard and the developers will not have to learn another set of tools.

As an alternative, Linux could be used, because it is a compact, stable and otherwise suitable operating system for embedded computers. But as stated above, using it would mean that the developers would have to learn new tools to get it working.

2.2 Size

It is easy to believe that it is plenty of room within a military track vehicle and that new devices could be installed without any effort. This is far from the truth, every corner in the vehicle is full of equipment. Therefore the recording unit must be as small as possible. No interaction-devices, like buttons and indicators should be mounted, because then it would have to be installed near a crewmember. In this way the Recorder can be installable anywhere in the vehicle.

2.3 Harsh environment

The environment on a military track vehicle is harsh. Inside, all things are vibrating when the tracks hit the ground and the temperature can range from freezing to hot. This calls for equipment that is made for rough conditions. Rugged components can often handle large temperature differences, vibrations and shocks. An ordinary home-PC would not survive long in these conditions.

Chapter 3 Software for Video Compression

3.1 Background

3.1.1 The Video Signal

A video signal received from a camera can be either digital or analog. An analog signal is interlaced and comes in a few different formats. The analog signal is very noise sensitive compared to the digital signal.

Composite Video

The Composite Video standard is most widely used in home-video applications, for example to connect the VCR to the TV. In this standard three types of signal are put together into one composite cable. These are brightness, color and synchronization signals. Usually Phono plugs are used as connectors.

A good thing with Composite Video is its wide availability of components and applications and a bad thing is that the signals can interfere with one another and cause bad picture quality.

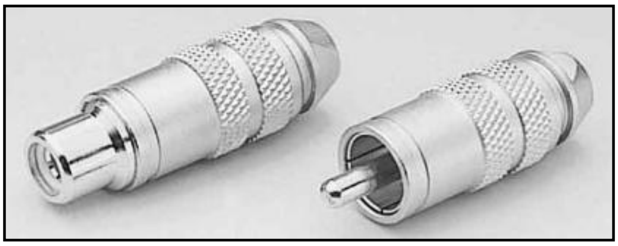


Figure 1: Phono connectors.

S-Video

S-Video works almost in the same way as Composite Video, but here the brightness and color information are separated. The format is sometimes called Y/C (Y for luminance, C for chrominance) or Hi-8. The result of separating the channels is a better picture quality. The connector used is a 4-pin mini DIN connector, the standard is available on most computer graphics adapters with TV-out and on newer VCRs.

S-Video is sometimes confused with S-VHS, which is a tape-standard used in VCRs that support S-Video. [1]

Figure 2: 4-pin mini DIN connector.

Pin	Description
1	Brightness ground
2	Color ground
3	Brightness + synch signal
4	Color signal

Table 1: Pin settings for 4-pin mini DIN.

SCART

SCART is a common standard connector and is used in virtually all VCR-to-TV interconnections. It transfers information in both ways, for example both the video in and video out signal. SCART supports stereo audio, RGB color information, S-Video and Composite Video. [2]

Pin	Description
1	Audio Out Right
2	Audio In Right
3	Audio Out Left + Mono
4	Audio Ground
5	RGB Blue Ground
6	Audio In Left + Mono
7	RGB Blue
8	Audio / RGB switch / 16:9
9	RGB Green Ground
10	Clock Out
11	RGB Green
12	Data Out
13	RGB Red Ground
14	Data Ground
15	RGB Red / Chrominance
16	Blanking Signal
17	Composite Video Ground
18	Blanking Signal Ground
19	Composite Video Out
20	Composite Video In / Luminance
21 *	Ground / Shield (Chassis)

Table 2: SCART pin settings. * Note that pin 21 is actually the metal housing of the Peritel SCART connector.

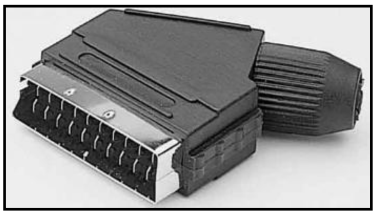


Figure 3: SCART connector

NTSC / PAL / SECAM

NTSC, PAL and SECAM are different broadcasting standards. NTSC (National Television Standards Committee) uses 525 lines with a refresh rate of about 30 fps and is used in North and Central America and Japan. PAL stands for Phase Alternation Line uses 625 lines but with 25 fps refresh rate. It is used in Western Europe, Middle East, South America and parts of Africa. SECAM is a French standard which is similar to PAL except for FM-modulated color signal. It is used in Eastern Europe, Russia and parts of Africa.

The difference between these standards makes replaying a video film recorded on an NTSC VCR impossible on a PAL VCR and vice versa. [3]

3.1.2 Large Data Streams

Handling digital video requires a lot of computer power and the reason is that video streams are very large.

Example: Consider a one-minute video clip with the resolution 640 x 480 pixels with a frame rate of 30 FPS and a color depth of 16 bits (65536 colors). One frame consists of 640 * 480 = 307200 pixels. Each pixel takes 16 bits, which means that one frame requires 4915200 bits (4800 Kbit).

In one second 30 frames are displayed and this requires a bandwidth of 144 Mbit/s. To store the one minute clip, a space of 144 Mbit * 60 seconds = 8640 Mbit = 1080 MB is needed.

Obviously video streams can be very large. The bandwidth needed is virtually impossible to obtain in a larger network. This fact presents the need to somehow compress the data. Fortunately, there exist many different techniques to compress data.

3.1.3 Real-Time Compression

Many specified compression algorithms are constructed in a way that data can easily be decrypted. The motivation of this is to make the video clips easily decodable by regular PCs without the need for special video cards. The price for this is that the compression becomes even heavier to perform. If real-time compression is vital, a good video card is needed.

If a system is too slow and can't manage all data in time, frames can be lost. This is not like an analog TV where the picture just gets a bit noisy if the bandwidth is temporarily poor. With MPEG for example, the whole clip gets impossible to replay if a few frames are lost. [4]

3.2 Some compression algorithms

There exist many compression algorithms and the most widespread will be looked upon. [5][6][7]

3.2.1 Intra-frame codecs

These algorithms take away the redundancy inside every frame of the video sequence. Simple areas of a picture can be compressed much more than complex areas. For example a 100 by 100 image in only one color can be coded as: "Paint the next 10000 pixels white". That is called run-length encoding. There exist many different algorithms for image compression, but all want to exclude information the eye can't see. Intra-frame codecs put independent compressed images in a row to obtain a video sequence. In this way a video sequence can be started and stopped anywhere between two frames which makes editing video easy.

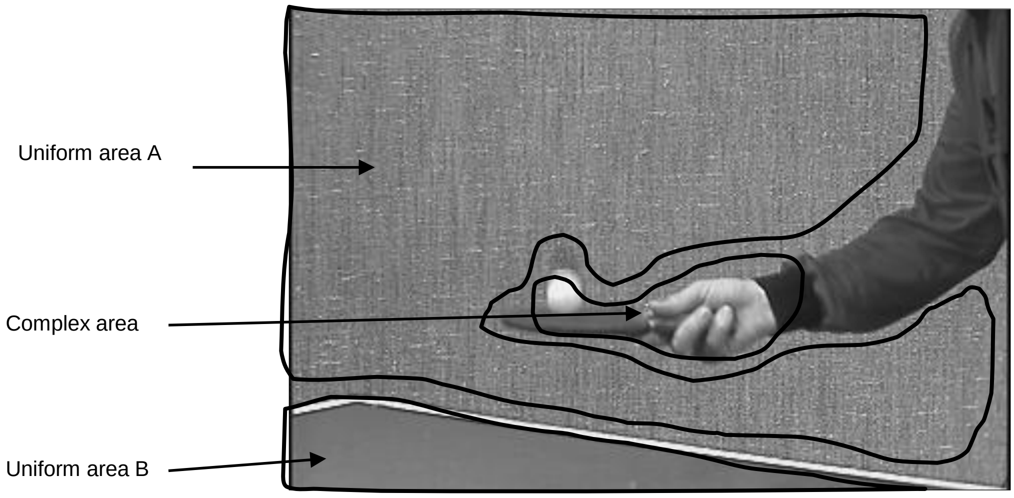


Figure 4: The simple areas can be more compressed than the more complex areas.

Motion-JPEG

Motion-JPEG or M-JPEG is based on compression on individual frames, called I-frames. The compression doesn't take advantage of similarities in sequential frames and every I-frame is compressed with the JPEG algorithm. This means a lot of work and special hardware is needed.

AVI

AVI is a wrapper video file format by Microsoft, which contains video data in any format. It has a theoretical maximum file size of 8 GB.

MOV

MOV is Apple's version of a wrapper video file format.

H.261 and H.263

H.261 and 262 are compression standard for ISDN. H.261 uses variable bit rate and is designed for 64 Kbit/s and 128 Kbit/s. H.263 is an extension to H.261. For fair quality the compression ratio is about 1:30.

Wavelet

Wavelet is an image compression algorithm like JPEG. But instead of dividing the image in small squares, the whole image is analyzed. This increases the image quality, particularly in images with one-colored smooth areas. When an image is compressed using Wavelet, few bytes are spent on areas with one color and more data on complex areas instead.

Figure 5: Wavelet compression, 1:71.9.

Figure 6: JPEG compression, 1:69.3.

In a video sequence compressed with Wavelet, the images are stored one by one in a sequence. Therefore wavelet is no inter-frame compression algorithm.

Indeo

Indeo is a compression format developed by Intel, which uses a Wavelet compression algorithm. It can be replayed using Apple's QuickTime. Indeo has a very high image quality. With good picture quality the compression ratio is about 1:10.

Cinepak

Cinepak is commonly used for movie compression when a CD-ROM is the storage media. This standard is very slow to compress but easy to decompress, which enables slow computers to replay a film. With good picture quality the compression ratio is about 1:10

YUV

YUV is a way, like RGB to represent image information. With RGB, each pixel in a digital image is represented by a red, a green and a blue color component. YUV, or YrYbY split the color signal into luminance, Y, red-luminance, rY and blue-luminance, bY. Since the eye is more sensitive to luminance than to colors, the color information can be compressed. With YUV 4:4:4, a four by four pixel sub-image uses four pixels to represent Y, four to represent rY and four to represent bY. In YUV 4:2:2, only two pixels represent red and two blue. And in 4:1:1 only one pixel for each 4-pixel sub-image represents red and blue. This way we lose some information, but it does not affect the quality much.

DV

DV uses YUV 4:2:2 or YUV 4:1:1 to represent the color information. The resulting image is then compressed further using the discrete cosine transform (DCT) that is also used in the JPEG algorithm. The compression ratio is about 1:5, which results in very good video quality.

3.2.2 Inter-frame codecs

Inter-frame compression algorithms not only compress a single frame in a video sequence. These algorithms also try to eliminate the redundancy between neighboring frames in the sequence. For example if a wall is recorded for a few seconds, many frames in a row will look exactly the same. This redundancy can be removed with intra-frame codecs.

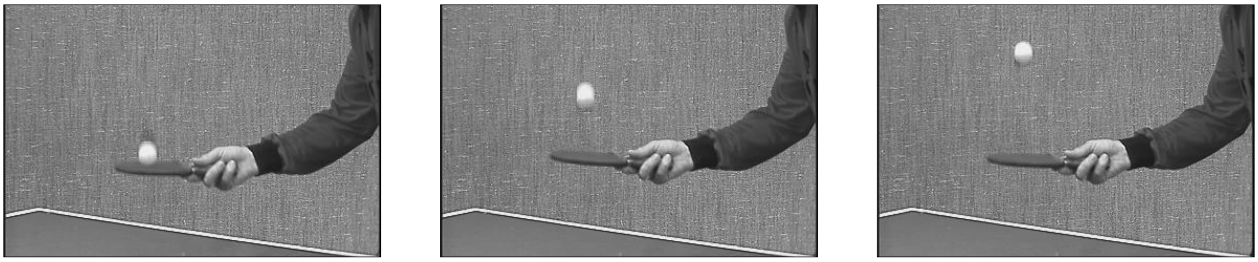


Figure 7: Here the ball is the only thing that differs between the images. When coding these three images only the first one needs to cover the background. The other frames can use the same background with a new ball position only.

In theory only the first frame in a movie is stored as a complete picture and all the other just represent the difference from the first one. But if every frame is based on the frame before, a tiny error in the beginning of the film can result in a huge error after a while. To help this control frames are stored every now and then.

MPEG-1

MPEG or Motion Picture Expert Group has a few different formats for video compression. One of them is MPEG-1 which was released 1993. It uses a bandwidth of 1 Mbit/s to 2 Mbit/s fixed bit rate, which results in a quality comparable to the VHS standard. This algorithm is asymmetric or non-isotropic, which means that it will take much longer to compress the video than to decompress it. The reason for this design is that the makers of a MPEG stream usually have quite powerful hardware and they want to reach a broader audience, with less powerful hardware.

MPEG-2

MPEG-2 was released 1995 and is an extension to MPEG-1 but with variable bit rate. It is used for DVD movies and can use a bit rate up to 100 Mbit/s. The most common bit rate used is 4 Mbit/s.

MPEG-4

This format is one of the latest and is specially designed for mobile and Internet use. That means low bandwidth, in the range between 10 Kbit/s and 1 Mbit/s. It has better tolerance to losses than MPEG-1 and MPEG-2. It is based on MPEG-2.

MPEG-7 and MPEG-21

MPEG-7 and -21 are formats under development. The algorithms are very similar to MPEG-2 and -4. The differences are mostly in the parameter settings.

3.2.3 Other codecs

ZIP, RAR, LZH

These are generic compression algorithms for all file types. They are loss less, but the compression ratio is in the 1:5 range which makes them a bad alternative in most video compression applications.

3.3 Operating System

An essential part of the recording system is the operating system (OS). The recorder is based on a PC, and as discussed in Chapter 2 PC requires an operating system to work in a simple manner. There exist many operating systems, but for this system only a few are usable. Firstly, the drivers for the components in the recorder must be supported by the OS. This requirement alone eliminates almost all operating systems on the market. Commonly supported operating systems are DOS, Linux and different versions of Windows.

The current computers onboard CV9030 use Windows and even if it would be possible to use other OS in the recorder, it is easiest for the developers if the same OS is used in every computer. Below is the most commonly supported operating systems briefly examined.

3.3.1 DOS

DOS, Disk Operating System, is the first OS from Microsoft and it is old and obsolete.

3.3.2 Windows

Microsoft's Windows is the most widespread operating system in the world. It is used in everything ranging from small pocket computers to embedded computers to workstations and servers. This is the biggest advantage for the OS since it follows that many products, both hardware and software supports it. Below, the different versions of Windows are briefly described.

3.3.3 Windows 3.1

Windows 3.1 was the first version of Windows that really became a hit on the market. With Windows 3.1 it was possible to move data between documents and connect the computer to a network. The OS was based on DOS and this resulted in a system that worked rather slowly, the opinion from many users was that it slowed down the computer too much to be usable. Many kept to the old DOS instead. But anyhow, Windows 3.1 brought the beginning of the use of graphical user interfaces to the PC world.

Nowadays Windows 3.1 is considered an old and obsolete OS and it is not an alternative for use in the video recorder. The two main reasons for this are the poor performance and the fact that very few hardware components support the OS today.

3.3.4 Windows 95/98/ME

When Windows 95 was introduced people understood that this is the way which computer interfaces will work for a long time. And indeed, the interface has lasted at least until now. Compared to the older Windows 3.1, this OS uses the hardware in a much more efficient way, thus making the system faster than before. However, the Windows 95 family still is based on DOS and, though performing quite well, does not completely bring the most out of the hardware.

A nice feature in the Windows 95 family is the Plug and Play functionality. This is essentially a big library of drivers for many types of hardware. When installing for example a new sound card, the OS recognize this and is able to use it without installing the drivers including the card. This makes managing of the system much easier.

The Windows 95 family is still one of the most used operating systems and is supported by a large variety of products. Therefore it may be a realistic choice when building the video recorder. On the downside for this OS is the stability issue. Windows 95 is not as stable as for example DOS, Linux or Windows 2000. The reason for this can be its large and complex structure and that it has been evolved through the years with improvements every now and then. As mentioned before, it is based on the rather old DOS, and getting a system consisting of both old 16 bit and new 32 bit programs to run stable is not easy.

3.3.5 Windows CE

Windows CE or Compact Edition is an OS specially made for embedded computers. It works much like Windows 95, but is slimmed to make it possible to use it in pocket computers and other small devices where only a subset of the functionality in Windows 95 is used. Plug and Play for example is not needed to such a large extent as in an OS for a desktop PC. Extra hardware is usually not installed in an embedded computer. When developing for Windows CE, a CE emulator can be used on a workstation. When the program is finished or there is a need to test it in its real environment, it is downloaded on the machine running Windows CE.

Windows CE needs only a fraction of the RAM memory and storage memory used in a desktop system. It is also much quicker to boot, since a search for hardware components is not needed. The OS is already designed for the particular hardware that is present in the embedded computer.

3.3.6 Windows NT

Windows NT is not based on DOS, like Windows 95 and it makes this operating system a better performer. It has a more rigorous security control and hardware management making the OS more stable. NT has been a popular OS for PC-based servers since the release and therefore most of the bugs have been sorted out. It does not have the Plug and Play functions like Windows 95, but it lets the user have quite good control of the hardware.

This is the OS that was chosen for the Recorder, both for its stability and the existence of hardware drivers.

3.3.7 Windows NT Embedded

Windows NT has a lot of built-in functionality. For example it has a lot of computer management programs and support for adding and configuring new hardware and so on. An embedded computer usually only runs one, or a few applications, and the hardware settings almost never change. Therefore much in Windows NT is unneeded. When installing Windows NT Embedded on a machine the needed parts only are put into a package and installed on the target computer. The OS then consists only of parts that are necessary. It also support headless mode, which enables the computer to work without any interaction devices like keyboard and monitor. This could be a good

OS for a Recorder, but in this project it was not worth the work to set up and use Windows NT Embedded.

3.3.8 Windows 2000

Windows 2000 could be called Windows NT 5, since it is based on Windows NT 4. The main difference between NT4 and NT5 is the Plug and Play support.

3.3.9 Windows XP

This is roughly a newer version of Windows 2000, with a new user interface and more support for internet.

3.3.10 Linux

Linux could have been a good candidate, but since no CAN card drivers was available for Linux it was not an alternative. This is mainly because there was not enough time to develop new drivers.

Chapter 4 Hardware

In this chapter the hardware possibilities for the video recorder is going to be examined. There are many ways of recording video, but all methods are not suitable for use in a vehicle.

4.1 The Parts of the Recorder

As stated in Chapter 2, a PC will probably be a good base for a video recording system. But what will the PC contain? An ordinary home PC does not have the ability to record video streams. It can't communicate with armored vehicle systems and it will break after little use in a vehicle on iron tracks. It may sound like the PC is not at all a good solution for video recording.

However, a PC is a flexible piece of machinery and it exists in many different shapes. It can be expanded with additional functionality due to the possibility to add circuit boards to the PC. Furthermore, PCs made for harsh environment, are available.

Now we will look at the different parts of the recorder.

4.1.1 The Base of the System

We have decided to use a PC as the base for the recorder. Then what different parts does a PC consist of?

The Processor

The processor lets the software control all other parts of the computer. It is important that the processor is powerful enough, otherwise recording high quality video is not possible. But it is not necessary to have the latest one for video recording.

When today's video compression methods were developed, the PCs were not as powerful as they are now. The processors in these computers were not powerful enough for the heavy compression work. To overcome this problem, special video cards for video compression were developed. The software is consequently built to use separate video cards. This means handling video doesn't require a high-speed processor. Once a video card is installed, an ordinary home computer gets capable of handling video streams.

The Memory

All computers need a memory, where the data is stored during computations. The most common memory in a PC is called Random Access Memory, or RAM. For video recording, a memory size of 256 MB should be enough. The reason for this is that the compression algorithms in most cases are made for older computers and can't take advantage of large amounts of memory.

The Main Board

A main board is needed to interconnect the processor, the memory and the other parts of the computer. The main board is a circuit board with connectors for the processor and memory. It also has a bus for communication with additional devices. There exist different bus standards for the PC, but the most common ones are ISA and PCI. Different devices can share both types of buses. This means, if many devices use the bus at the same time, the bus is overloaded and will be unable to handle all data.

The ISA bus

The ISA bus is 16 bits wide and runs at 8 MHz. This results in about 4 MB/s data transfer. [8] Extensions have been made to improve the ISA bus and one is called Extended ISA, or EISA. This bus features 32 bits width, and is backward compatible with ISA. EISA runs at 8 MHz as well and can transfer data with between 20 MB/s. [9]

The PCI bus

PCI replaces ISA and EISA. This bus runs at 33-66 MHz and is 32 or 64 bits wide. This results in a bandwidth of up to 133-266 MB/s depending on bus width. [10]

The PC/104 and PC/104+ buses

PC/104 and PC/104+ are technically the same buses as ISA and PCI respectively, only with different connectors. These buses are commonly used in embedded systems and make it possible to stack circuit boards on top of each other, resulting in compact packages. PC/104 is the bus standard used in the Recorder.

Which Bus Type is best for the Recorder?

It is not self-evident that the PCI is needed in a video recorder system. It depends on the components. If the video grabber board can both convert analog video into digital and compress it in real time, the data stream on the bus will be relatively small. If, on the other hand, the raw data from the frame grabber board must be fed to a compression board, the bus must have a very good performance.

Hard Disk Controllers

Usually a hard disk controller is built in the main board. If not, a controller card has to be installed either on the ISA or PCI bus. There exist a few standards for hard disk controllers; the most common are IDE, SCSI and SATA. [11]

IDE

The most widespread hard disk controller is called IDE. Virtually every new main board comes with an IDE connector onboard. There exist many different IDE standards, for example Ultra DMA-33, Ultra DMA-66 and Ultra DMA-100 with theoretical bus speeds of 33, 66 and 100 MB/s respectively. Any of these versions perform well enough for video recording, since compression will probably be used. [12]

An IDE controller can only handle two units, but in most PCs, two controllers are mounted, to enable up to four units at the same time. In the video recorder one hard drive is enough.

SCSI

SCSI is another common controller, which like IDE, also exists in many different versions. Common for all SCSI controllers is that they can handle a larger number of devices than IDE. Typically 7 or 15 devices can be used simultaneously. SCSI has better bandwidth than IDE and one big advantage is that the SCSI controller usually has a dedicated processor that handles the transfers. This helps the main processor to handle this many devices, which otherwise is a heavy job. This makes the SCSI controller a good choice for server applications, where large storage areas are needed and many users access hard drives at once. For video recording, on the other hand, it is not necessary to use a SCSI controller.

Serial ATA (SATA)

SATA replaces IDE in the modern PC. The performance is 150 MB/s and up. [13] In a SATA bus, data is transferred serially. The SATA interface supports hot-swapping of disks.

Display

A PC usually has some kind of display attached. This could be either an ordinary CRT monitor or a flat screen. This display is connected to a display output on the computer. Home PC’s generally manage the display by a graphics adapter mounted on the PCI or AGP bus. AGP is a bus specialized for graphics adapters. It almost works like a PCI bus. The main difference is, only one unit can be connected to the AGP bus at once.

A video recorder controlled by the CAN bus will not need any display. Preferably, it should be a little box that can be installed anywhere in the vehicle. Therefore, it does not matter if the PC has a display adapter or not.

4.1.2 Sampling the Analog Video Signal

Ok, now we have a PC main board with a processor and memory installed. We can just call this the CPU board. To be able to use the analog video signal a frame grabber is needed. A frame grabber grabs individual frames from an analog signal and converts each frame into digital images. These images are put together to a movie and if TV quality is enough, a frame rate of about 25 frames per second is required. This means that the frame grabber have to take a sample of the analog signal 25 times per second and convert them to digital images.

To connect the frame grabber to the CPU board we use either the ISA or the PCI bus. Through the bus it is possible to control the frame grabber and transfer video data to the hard disk controllers or RAM memory.

4.1.3 Encoding the Video

To get TV quality video we must capture at least 25 images per second each with a resolution of about 768 x 576 pixels. With a color depth of 16 bits, this requires a bandwidth of 25 x 768 x 576 x 16 = 177 Mbit/s which is roughly the same as 22 MB/s. This is more than, for example, the ISA bus can handle. The PCI can handle it, but to store one minute of video, 1.3 GB is needed. This is inefficient and to solve the problem the video stream must be compressed.

There exist many different compression algorithms, MPEG versions being the most common of these algorithms and reduces the video data size drastically. Due to the large data stream involved when dealing with digital video, the compression is very hard work for the processor. The CPU in a PC is usually unable to manage this work and to cope with this problem an extension to the PC has to be done. Adding a compression card to the PCI bus, where the data between the frame grabber and the compression card is transferred, often does this. The disadvantage of this configuration is, the PCI bus will be busy with transferring video, which may lead to bad handling of CAN massages, for example.

A good solution is to have the encoding device integrated on the frame grabber instead of having two separate boards for frame grabbing and compression. In this setting the PCI bus would only be transferring compressed data, which is much smaller in size. Additionally, it would probably be easier to develop the software for a board of that type and we would not have to take care of the communication between frame grabber and compressor at all.

4.1.4 Storing the Video

There is no point in installing a video recorder if it isn't possible to store the video data. Therefore some kind of storage device is needed. A number of different alternatives are possible but the storage device must fulfill a few requirements. Firstly it must be fast enough to store the video stream. If the video is compressed, the stream is relatively small, but if we want high quality movie the video stream grows larger. The storage device also must be big enough to store big sequences of video, contain an OS and needed applications.

Use of a hard disk drive for video recording is a good solution, but in this case the storage device has to handle a lot of vibrations and a hard environment. A hard disk simply cannot cope with this, so therefore some other solution is needed, and one is to use a flash drive. Flash drives are very durable and are available in many different sizes. The downside of that type of drive is they are very expensive. [14]

There is a need to review the video in an office environment. One way would be to replay the video with the video recorder and use an external analog tape recorder to store the video. This would make it easy to watch the video in other places than the vehicle, since tape recorders are available almost anywhere. Another way would be to connect an external hard drive to the recorder, either with some serial connection, like USB or FireWire. More of this problem is described in 4.7.

4.1.5 Decoding and Replaying the Video

Now the parts for recording and storing video sequences have been briefly described. The analog video signal is converted to a digital video stream, which is compressed and stored on a mass storage device. But it must also be possible to replay the video.

On the PC's in the vehicle are able to select one of the cameras and watch it. They all have video cards installed which takes an analog video signal and display it on the screen of the PC's and it works like this: The cameras are connected to the inputs of a video switch, which is also connected to the CAN bus. The PC's are also connected to the output of the video switch which means that the computers can control the input to their video cards by giving orders to the video switch via the CAN bus. The switch can serve a PC individually, which enables for example PC no. 1 to watch camera A, when PC no. 2 is watching camera B. Any other combination is also possible.

If the recorder gives an analog video output, it could be connected to the input of the switch and in this way work like a camera in the vehicle. While the video is being replayed into the switch, it could be watched on any PC. This presents another requirement of the video recorder. It must be able to decompress the stored video, and present it as an analog signal.

With the same arguments as with the recording, it is desirable to have both the decompression device and the analog output on the same board to minimize the use of the PCI bus.

4.2 Capture cards

There exist different kinds of capture cards on the market. Some have digital input and some have analog input. Certain cards are able to compress the video in real time and some grabs pictures with very high resolution.

Capture cards also vary a lot in pricing. In the cheaper segment the difference between prices is reflected in a difference in the resolution and the frame rate of the video output. From a certain price level and up they have roughly the same video quality, but more expensive cards are often faster than the cheaper ones. Cards with compression ability are in general more expensive than others.

4.2.1 Some Requirements of the Components

A video recorder should have certain functionality. Mainly it should be able to record video and later replay the film. Because an uncompressed video stream takes so much space, the recorder has to compress it. In 4.1, it was concluded that it is most suitable to have a capture card that is able to capture and compress the video as well as decompress and output it as analog video. The desirable bus type is PC/104 or PC/104+, since using PC/104, the recorder would have the same bus type as the current system in the vehicle. PC/104 devices also have a compact design and are usually made for use in harsh environments. [15]

4.2.2 Market Research

To decide which capture card is most suitable for video recording in the given environment, a market research was carried out. To get a grasp of how many manufacturers and their models of capture cards there are, a thorough search on the Internet[1] was done. Every found capture card was inspected regardless of suitability. The reason for this was to not discard a capture card too early.

4.2.3 The Usage of Capture Cards

In Table 3, the capture cards found in the first search are shown. They are sorted alphabetically in a table, which only tells if a card is able to compress or decompress and which form factor it has. Most capture cards don't handle decompression and replay of video. The reason for this might be that they are primarily made for security surveillance and traffic watch, where there is no need to replay the video. When watching traffic, the images are analyzed directly and then discarded without being stored. This might explain the great number of capture cards without the ability to compress the video.

Manufacturer	Model	Compression Ability	Decompression Ability	Form Factor
Advanced Micro Peripherals	MPEG1000	Yes	No	PC/104+
Ajeco	ANDI-FG	No	No	PC/104
Applied Integration Corp.	**AI-PC104/Wave**	**Yes**	**Yes**	**PC/104**
Applied Integration Corp.	**AI-PCI/Wave**	**Yes**	**Yes**	**PCI**
Applied Integration Corp.	**AI-PCI/Wave16**	**Yes**	**Yes**	**PCI**
EuroTech	CTR-1470	Yes	No	PC/104+
Darim	MG100	Yes	No	PCI
Darim	MPEGator Pro	Yes	No	PCI
Darim	MPEGator2	Yes	No	PCI
Digital Logic	MSMG104-Plus	No	No	PC/104+
DSP Design	TCVideo	No	No	PC/104+
Image Nation	CX Family	No	No	PC/104
Image Nation	PDX	No	No	PCI
Image Nation	PXC Family	No	No	PCI
Real Time Devices	CM7326	No	No	PC/104+
Sensoray	Model 311	No	No	PC/104+
Sensoray	**Model 512**	**Yes**	**Yes**	**PC/104**
Videonics	Python	Yes	No	Parallel port
Vitec Multimedia	**MPEG Profiler**	**Yes**	**Yes**	**PCI**
Vitec Multimedia	**VM2**	**Yes**	**Yes**	**PCI**
??	**MPEG-PCI**	**Yes**	**Yes**	**PCI**

Table 3: Frame grabber cards. The cards shown in boldface will be further examined while the rest are discarded. The motivation to keep the boldfaced capture cards is that they are able to both compress and decompress the video signal.

---

[1] Mainly with the Internet search engine Google, www.google.com

Some of the words used: mpeg, compression, video, capture, pc/104, frame, grabber

4.2.4 All capture cards does not fit into our application

To enable a deeper examination of the capture cards, some cards were discarded after just a quick glance on their specs because the lack of necessary functionality.

The capture card must be able to both compress and decompress the video, so therefore all other cards are sorted out. The result is presented in Table 4.

Manufacturer Model	**Applied Integration Corp. AI-PC104/Wave**	Applied Integration Corp. AI-PCI/Wave	Applied Integration Corp. AI-PCI/Wave16	**Sensoray Model 512**	Vitec Multimedia MPEG Profiler	Vitec Multimedia VM2	?? MPEG-PCI
Form Factor	**PC/104**	PCI	PCI	**PC/104**	PCI	PCI	PCI
Encoding	**Wavelet**	Wavelet	Wavelet	**MPEG-1, MPEG-2**	MPEG-1, MPEG-2	MPEG-1, MPEG-2	MPEG-1
Input	**Comp, S-Video**	Comp, S-Video	Comp, S-Video	**Comp, S-Video**	Comp, S-Video	Comp, S-Video	Comp.
Max Resolution	**768 x 288**	768 x 288	768 x 288	**768 x 576**	720 x 576	720 x 576	352 x 288
Output	**Comp, S-Video**	Comp, S-Video	Comp.	**Comp, S-Video**	SCART	Comp, S-Video	Comp.
Single Frame Capture	**No**	No	No	**No**	No	Yes	Yes, 720 x 576
Sound	**No**	No	No	**Yes**	Yes	Yes	Information Unavailable
Drivers	**Win95**	Win95	Win95	**DOS/Linux/Win**	WinNT	WinNT	Win95/98
SDK	**Instructions in how to develop drivers.**	Information Unavailable	Information Unavailable	**C/C++**	DirectShow	Information Unavailable	Information Unavailable
Price	**$699**	Information Unavailable	Information Unavailable	**$613**	Information Unavailable	Information Unavailable	1680SEK
Misc			Supports up to 16 inputs	**Dealer in Sweden**		Requires 800MHz due to software encoder.	Dealer in Sweden

Table 4: Capture cards with ability to both compress and decompress data.

4.2.5 Designed for home users

Many capture cards with ability to both compress and decompress video are targeted for editing home video or publishing video to the Internet. They come with complete programs and drivers for this use, but with no support for development of custom programs. This means that the cards are not suitable for use in an embedded system, which demands a tailor-made program. Cards designed for embedded systems, on the other hand comes mostly with development kits.

4.2.6 Embedded Components often more Rugged

A recording system for a track vehicle has to be rugged, which gives parts made for embedded systems an advantage; in Table 4 these are highlighted. These two capture cards seem the most fit for the given environment and they fulfill the requirements, except for single-image capturing.

Both the Applied Integration Corporation (AIC) and the Sensoray capture card supports analog composite video input and output and both have compression and decompression functionality. They have drivers for Windows and include support for application programmers. The main difference between the two is the compression algorithms used. The AIC card uses Wavelet compression while the Sensoray card uses MPEG-1 or MPEG-2.

4.2.7 Other aspects than performance

Not only the performance and features motivate the purchase of a certain component. Other parameters, as availability and support affect the choice as well. In this area the Sensoray card seems better. Sensoray have a retailer in Sweden, which often means that it is easier to get support and a quick delivery if the card is bought. The Sensoray card also has easily understandable C code with examples downloadable from their web site, which gives a hint on the service offered by that company.

Considering the facts described above and the suitability of MPEG encoding, the Sensoray 512 is the most suitable alternative for the video recording demonstrator.

4.3 Mass Storage

With the large bandwidth requirements we are dealing with in video capturing the storage units have to be fast and large. If the speed of the hard drive is insufficient, frames can be lost which means, in the worst case it is impossible to watch the video clip.

4.3.1 Hard Drives

Today EIDE hard drives are common. This type is specified for speeds up to 33 MB/s, but it only gives the performance in short bursts and under perfect conditions. For sustained writing they only manage a speed of 3 – 6 MB/s. If we are dealing with video we need better performance to maintain a good quality. Therefore it may be better to use a SCSI drive to store the information. They manage a speed of 6 – 13 MB/s and doesn't put as heavy load on the CPU as the EIDE drives do. SATA, which also is common, may also be a good alternative.

It is also important that the hard drives are large enough. Not only due to the size of an entire video sequence, but also because a big drive generally is faster than a smaller one. A larger disk drive has higher data density and therefore more data can be written in one revolution of the physical disk. Thus, a 20 GB drive at 7200 rpm is faster than a 10 GB disk drive at the same rotation speed.

Most hard drives need a little break every now and then to make a thermal synchronization. To make it possible to deliver a stable stream of data anyway, a cache memory is used. But if the needed bandwidth is too large, the hard drive will not be able to fill up the cache. Consequently when the drive needs a synchronization break the data stream will also brake. Luckily, there exist disk drives made for steady streams, which don't need the break. Those are called AV drives.

4.3.2 Flash Drives

One alternative to hard drives is the Flash drive, which is a drive without any mechanical devices inside. It is completely electronic. This means that it will function in a much harsher environment. Flash drives may vary much in performance but they are often comparable to hard drives.

4.3.3 Tape Drives

Tape drives can be very fast and large, but they have very long access times. If a video file would be in the other end of the tape we have to wind it to be able to access the file. That makes a tape drive a bad alternative.

4.4 File system

There are a few file systems available for ordinary home computers. Some are good for video recording and some are not. FAT-16, Linux NFS version 2 and Apple Macintosh (prior to OS X) can only be used with less than 2 GB disk space. It is more suitable to use FAT-32 that can address about 4 GB or better, NTFS that addresses up to 4 TB.

4.5 RAM Memory

128 MB should be enough to get good performance in Windows NT. [4]

4.6 CAN

CAN (Controller Area Network) is a serial bus-communication protocol. It is designed for real-time systems originally in the automotive industry. It is reliable, robust and supports high-speed short messages. Therefore it is suitable for sending control commands and status messages between for example a controller and the engine in a car.

The can network is a broadcast-bus, which means that each message is received by everyone in the network. The receivers must look in the message heather to know if a response is needed. The messaged transported in a CAN network have an ID and the lowest one has the highest priority. The CAN network can reach speeds of up to 1Mbit/s.

To control the Recorder remotely, CAN is a good alternative, especially since CAN is the control network in the vehicle. [16]

4.7 Transporting the video

4.7.1 Parallel port

The parallel port is an old interface, used mostly to connect a printer to a computer. It is much faster than the RS-232 port, but the maximum speed is between 5.5 and 6 Mbit/s. [17]

4.7.2 USB 1.1 / USB 2.0

USB, or Universal Serial Bus is very common in modern PCs. It is specified for a speed of 12 Mbit/s for the 1.1 version and 480 Mbit/s for the newer 2.0 version. [18]

4.7.3 FireWire

The FireWire, or IEEE 1394 can deliver 400 Mbps. It is commonly used in home DV cameras for transporting the video into a computer. There are quite a few external hard disks available, which uses FireWire, so this could be a smart way of transporting the videos from the Recorder in the vehicle to the office. The files could be copied to a hard drive, a flash drive or a tape drive through the FireWire interface. [19]

4.7.4 Backup Tapes

Backup tape drives are quite fast, typically 10 MB/s but many different types are available. The backup drive could either be used as a stationary module in the vehicle, or as a portable device connected through a FireWire cable. The drawback is that the mechanical tape drivers and the tapes are rather sensitive to dust and rough conditions.

4.7.5 CD-R

To burn CDs with the video files could be a good alternative if not for the very shock-sensitive burning process. CDs are otherwise compact and inexpensive.

4.7.6 Radio

It would be convenient if the files could be sent to the office with a radio modem, but it is too slow and would be sensitive for eavesdroppers. Encryption of the data might work, but transmitting by radio would anyway give away the position of the vehicle.

4.8 Other solutions

4.8.1 Tape recorders

It can seem like overkill to use a computer to compress video data and store it digitally when almost everyone has a functional analog VCR at home. They can both record and replay video, and are very inexpensive. The problem is that they are not generally controllable by a computer. But there are a few exceptions. Some tape recorders on the market have RS-232 connectors, and can be controlled by a computer. Mechanically they work like ordinary VCR’s for home use.

However, there are dark sides of this solution as well. Firstly, a tape recorder has slow search speeds. If we want to look a video shot filmed half an hour earlier, we have to rewind the tape, which takes time. Secondly, it is hard to find tape recorders for mobile use. They are bulky and not very durable. And we must have a PC anyway to control the tape recorder, which means that the system will consist of two units. A tape recorder will not have the same ability to store single images as a digital device either.

The conclusion about using tape recorders as the video recorder in a military vehicle is that it is not recommended.

4.8.2 Decoding on Client Computer

Why can’t the existing PCs handle the recording process by themselves, instead of using a separate recording device? The answer is that they already have a lot of processing to do. The computers can’t handle both the current programs and recording at the same time. If the PCs would have power enough to do it, it would not be a good solution anyway because the video clips would be spread over more than one computer, which makes it hard to export the clips from the vehicle. We also have more than one client on-board, so it would be more expensive to have recording ability in all of them. Now any client can connect to the Recorder and use it. They are also able to perform their current tasks without interference from video recording

Chapter 5 Implementation

As mentioned before, an important part of this thesis is the construction of a demonstrator, which is supposed to give some hints on how a video recorder could work. The demonstrator should use earlier evaluated parts to verify if the assumptions where accurate.

5.1 System Layout

In the current VC9030 vehicle system there are four clients that are able to watch the different video cameras independently. Using video switches does this. They are controlled by the clients using CAN and let the user select which camera to watch. The system is arranged to let two video switches be connected to each other, which enables any client to watch any camera.

The current system simplified.

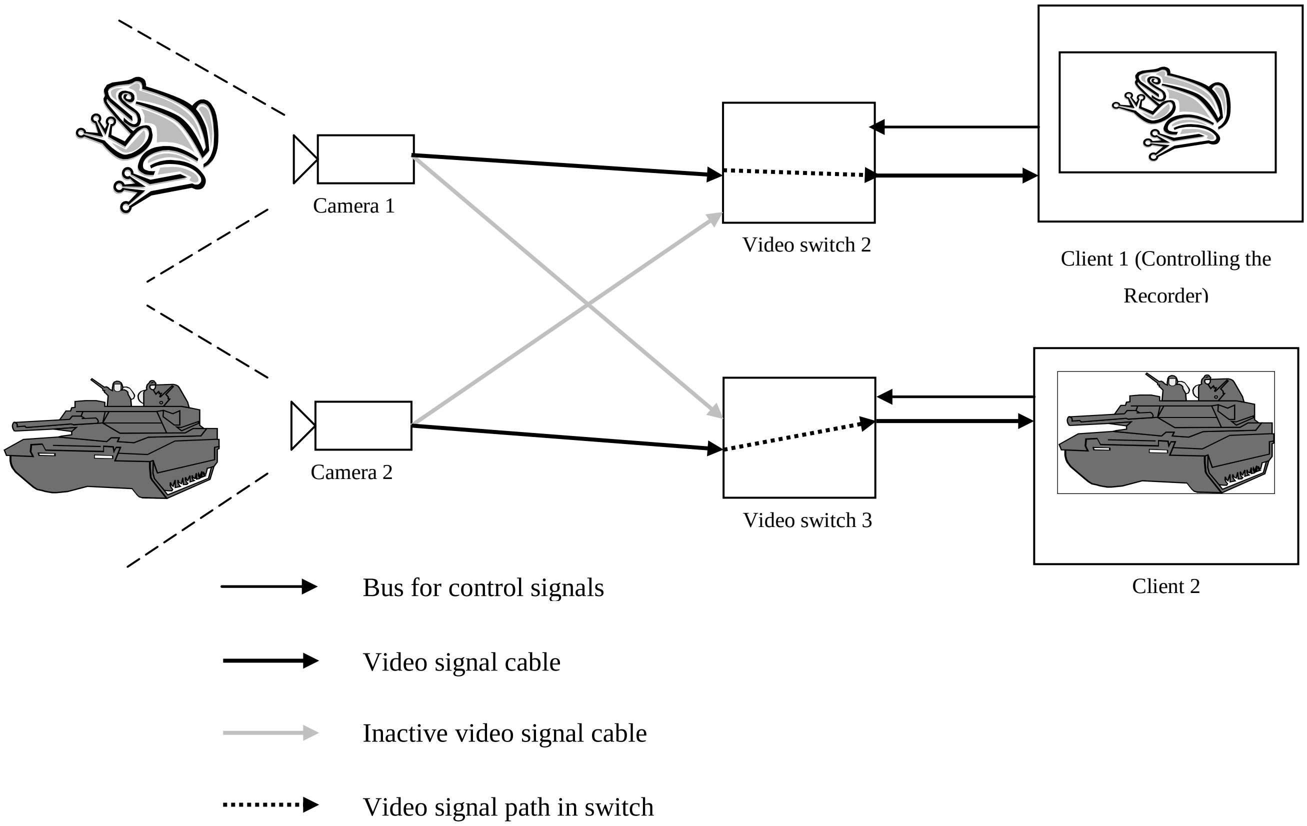


Figure 8: Client 1 is watching Camera 1 and Client 2 is watching Camera 2

The demonstrator is a simplified version of the existing system because there was no need to construct one that has all the parts in the existing system. Instead it has only the parts necessary for demonstrating video recording.

The demonstrator mainly consists of two computers, a client and the Recorder. They are connected with each other in two ways, with a CAN bus and with an analog video cable. The Recorder gets the analog video through a video composite cable, which is connected to a video source, for example a camera. The video is by-passed to the client at which a user can watch the video. The user must be able to control the Recorder, to start and stop recording and replaying video clips, for example. The user will not have any physical access to the Recorder, so it will be remotely controlled via the Client.

When the two devices are connected, software is needed to use the computers. The Client needs a program that lets the user give commands to the Recorder and a program that displays incoming video on the screen. The Recorder needs a routine that receives these commands and performs the recording. It has to use the drivers for the video grabber card to be able to record and replay.

Demonstrator system layout.

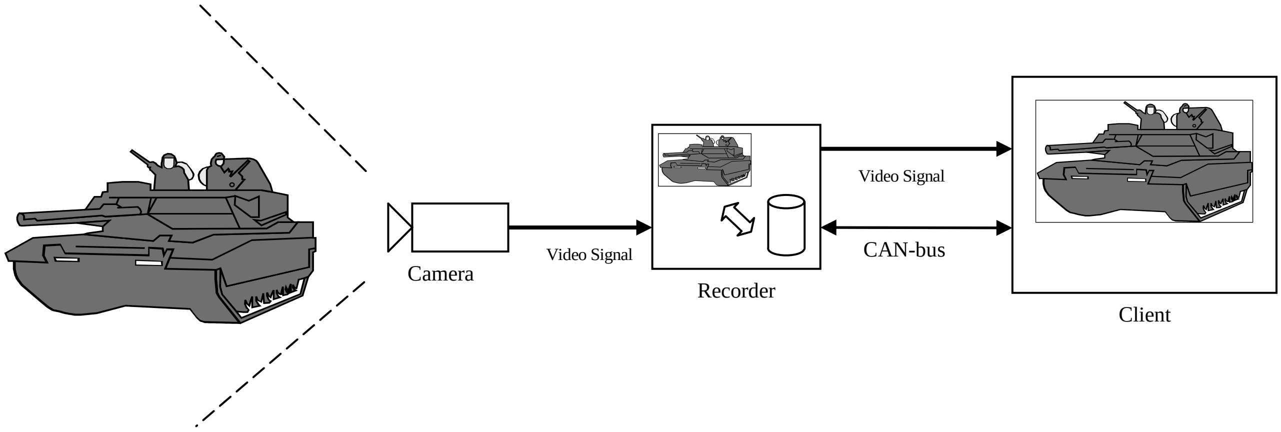


Figure 9: The Recorder can record the video signal and later replay it. While recording the video signal is by-passed to the Client which can show what is being recorded.

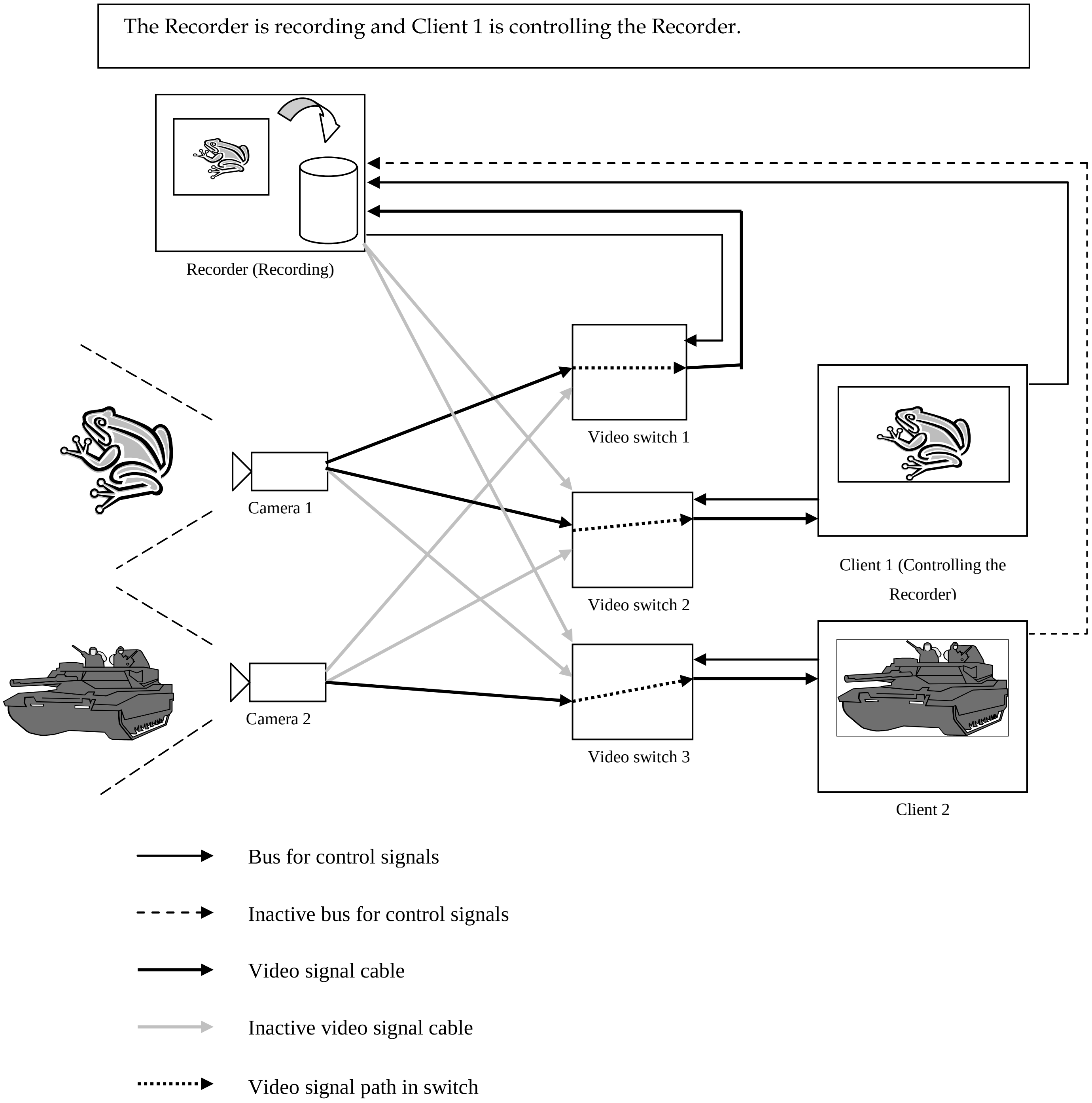


Figure 10: Possible system layout in the vehicle. The Recorder is in this state controlled by Client 1. It is recording and Camera 1 is the video source for both the Recorder and Client 1.

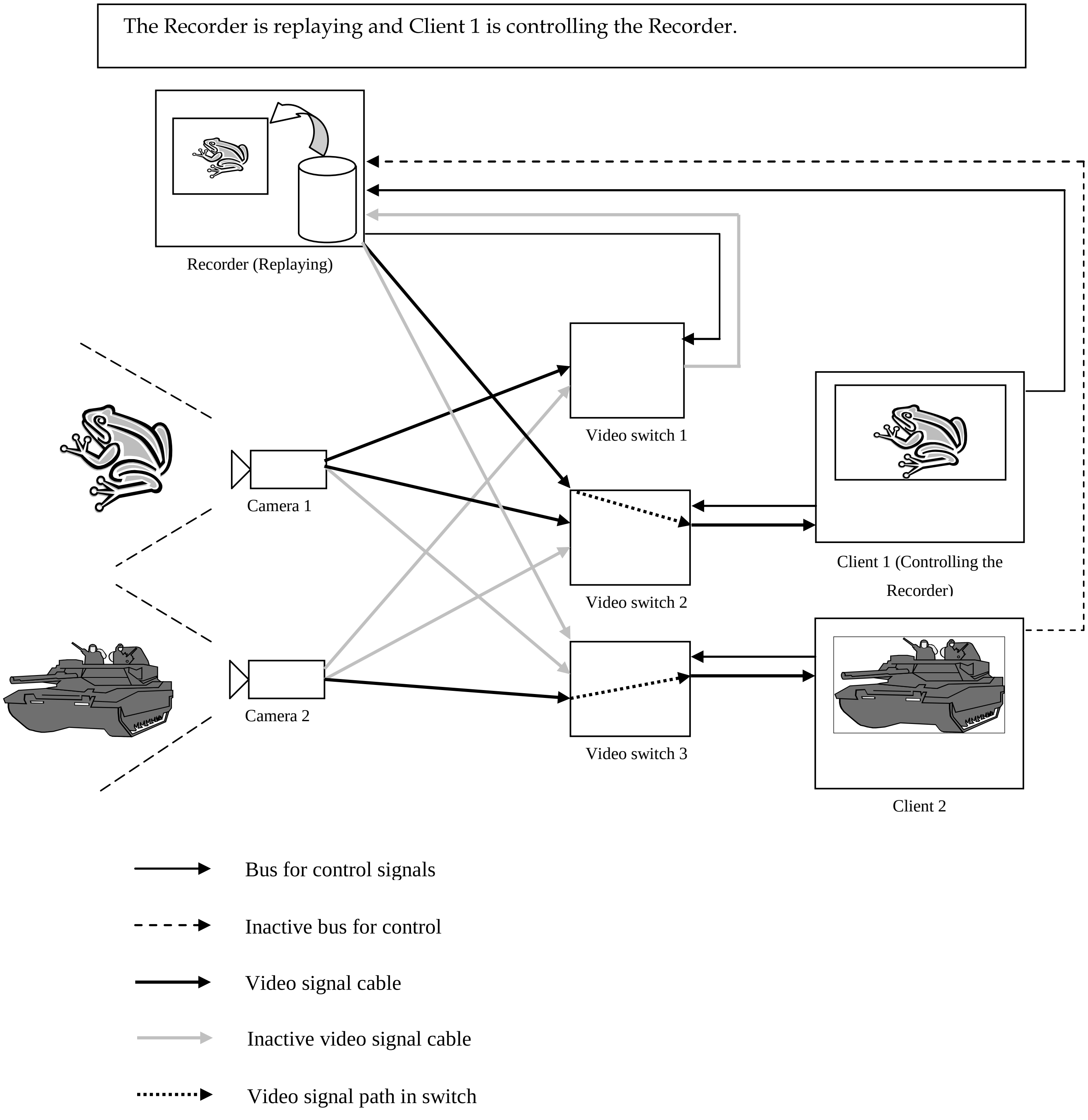


Figure 11: Here the Recorder is replaying and also a video source for Client 1.

5.2 Setup - The Client

As mentioned earlier, the Demonstrator will consist of both a Recorder and a client. The client in this system doesn’t have to be a rugged PC, like the real ones in the vehicle. Instead the client will be an ordinary workstation PC. It will run the client program, which lets the user, with a graphic user interface (GUI) control the Recorder. Controlling the Recorder means starting and stopping recording and replaying, deleting video files and a few other things. When a user hits a button with the mouse, a command is sent on the CAN bus telling the Recorder to perform the selected task.

Besides controlling the Recorder, the Client must be able to display live video on the screen. During the development of the Demonstrator, no camera was available, so a graphics adapter with composite video out was installed on the Client computer. This way, video could be fed to the Recorder.

5.2.1 Required Hardware

The Client in the Demonstrator system is an ordinary workstation and to be able to display analog video a TV card was installed. The choice of model was not that important since it simply has to display incoming video. It will be used primarily to examine different video qualities. A Hauppauge WinTV Go was selected because it was inexpensive and filled the one requirement that it is able to display incoming analog video.

5.3 Setup - The Recorder

5.3.1 Choice of Hardware

In Chapter 4 the hardware requirements were examined and the outcome was a few components needed for video recording. Firstly a base computer platform is needed. The best choice in this context is to use a PC based platform. These are widely available and are used in the vehicle currently as well.

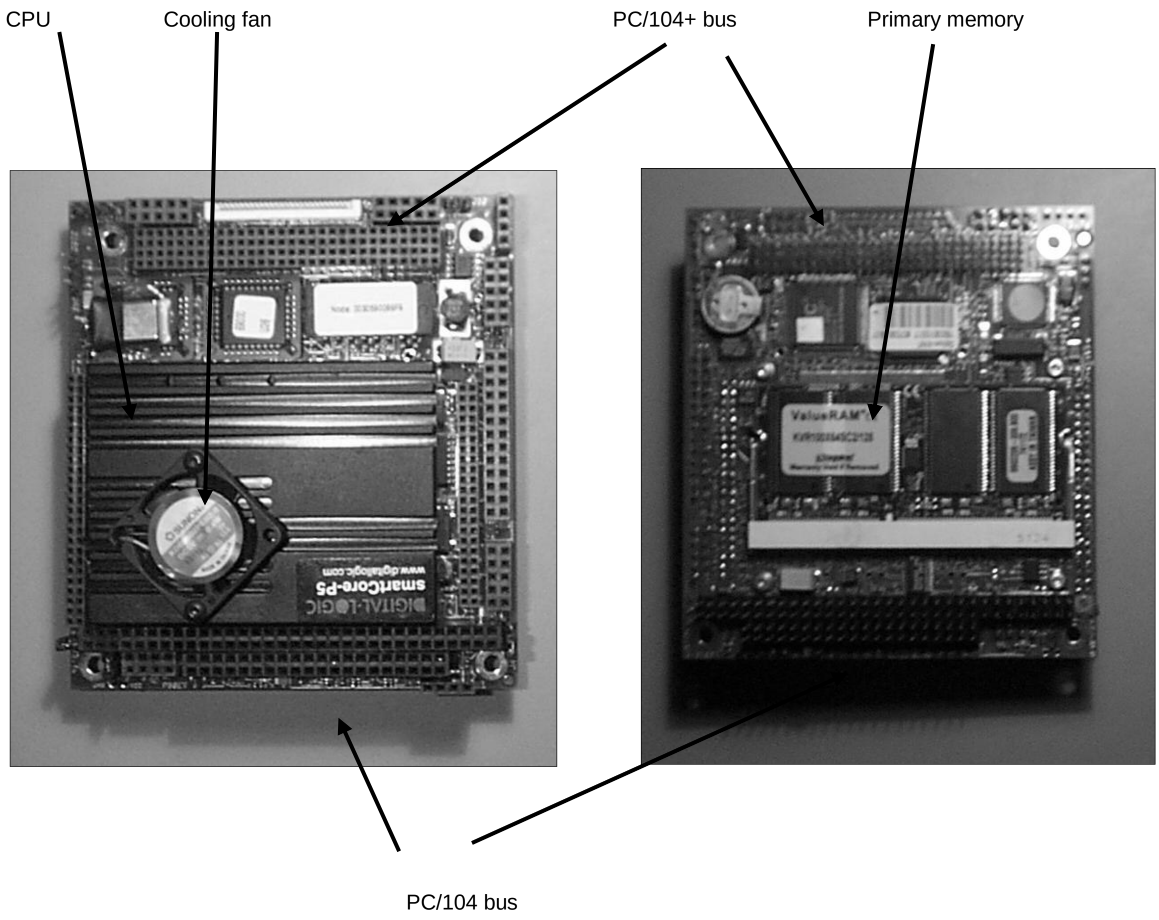


Figure 12: Digital Logic MSMP5SEV CPU Board. Left: Front side with a cooling fan for the processor. Right: Back side with the primary memory.

The selected platform uses the PC/104 standard, since it is the standard of the vehicle clients. The CPU unit is the same as in the current Clients as well, a Digital Logic MSMP5SEV unit with a 266 MHz processor. This CPU has support for video in and Ethernet. The CPU board is placed on a bottom card, which is an adapter for I/O ports like mouse, keyboard, monitor and Ethernet.

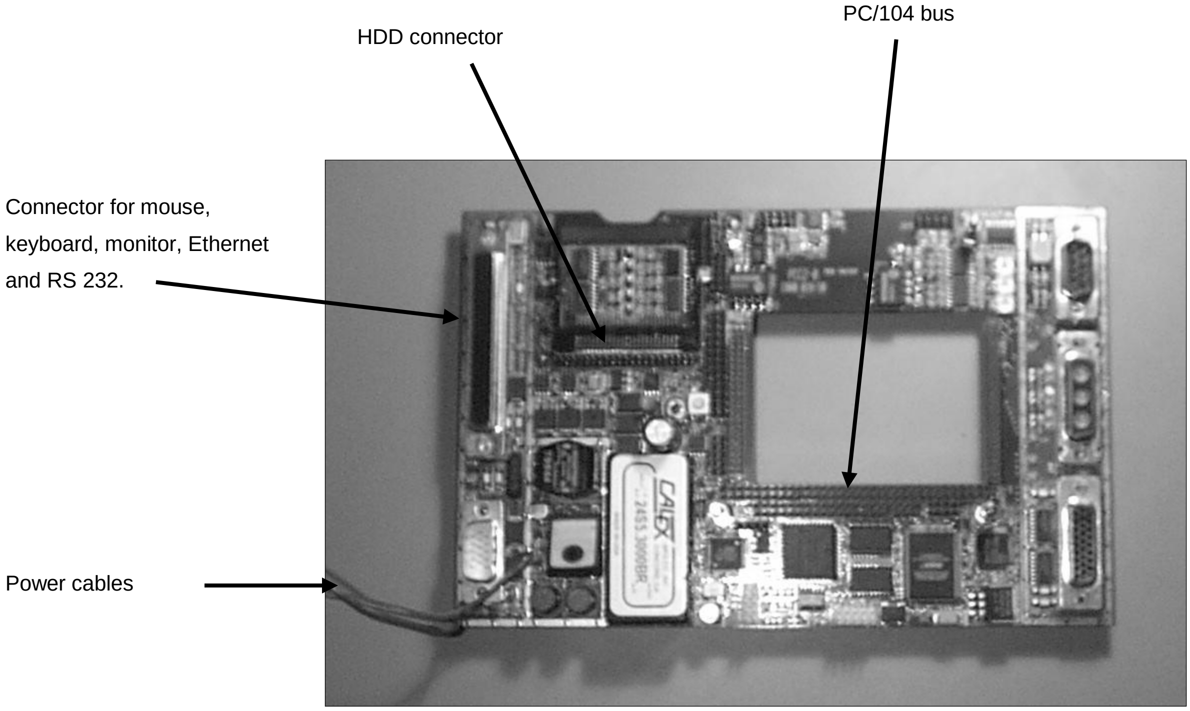


Figure 13: Bottom card for the Recorder. It is designed by CC Systems.

The bottom card only provides rows of pins with the different ports. Another important feature on the bottom card, besides stabilized power for the CPU board, is CAN functionality. This is a feature the CPU board lacks, which created a need for CAN support on the bottom card. A Mascot 24V power source is connected to the bottom card, which delivers stable power.

On top of the CPU board the video grabber board is installed. It uses the PC/104 standard as well and was therefore easy to put in place. In 4.2, the Sensoray Model 512 video grabber was selected as the best grabber in the context of this report.

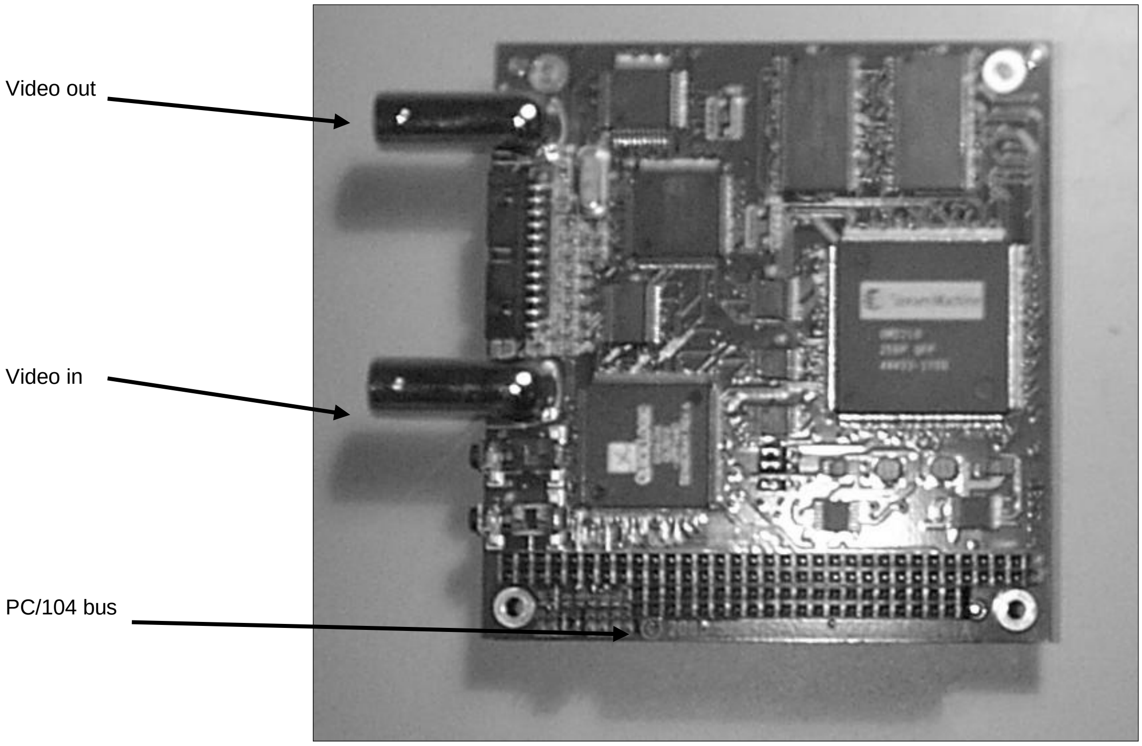


Figure 14: The heart of the video recorder. The Sensoray 512 frame grabber.

As stated in 4.3, a hard disk will probably not stand the harsh environment in a track vehicle and are therefore not suited for a real Recorder. But in the Demonstrator a hard disk will do just fine. A 2.5″ hard disk was bought, which is the format used in laptops.

Figure 15: The IBM 20 GB hard drive.

Other than the different components acquired for the Recorder, quite a few cables had to be made. The following is a list of these cables.

- Mouse
- Keyboard
- Monitor
- RS 232
- Ethernet
- CAN
- Video in
- Video out

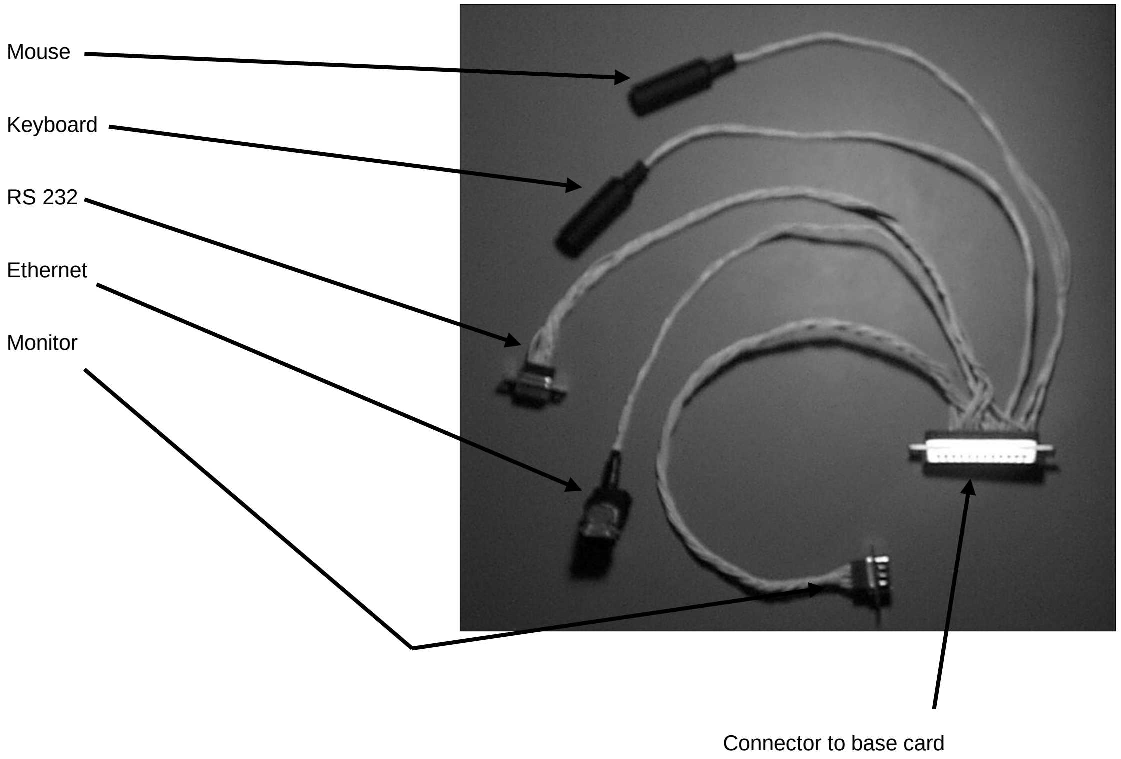


Figure 16: Cable for the base card.

Figure 17: Left: CAN cable. Middle: Video source to Recorder cable. Right: Recorder to Client cable.

5.3.2 Acquiring the Parts

When all the parts were selected we had to order the parts. Delivery times are often longer than expected, so when we had selected components, we started with the ordering.

Some components were available in the office. These include mouse, keyboard, monitor and power source. Even if the system really should be headless, which means without monitor, keyboard and mouse, these are needed during the software development and the driver installations.

The connectors for the made cables was ordered from ELFA and soldering and pressing put the cables together. Wires were available at CC Systems office.

The CPU board was ordered from the Swedish dealer About Industrial Computers and was delivered two weeks from the order date. The bottom card we could get in one day since one was available at CC Systems office in Alfta. The CPU board did not fit onto the bottom board directly so it was sent together with the bottom card to Alfta, where some of the sockets where replaced.

The Swedish dealer of the Sensoray board, ACTE Systems, let us borrow the board for evaluation for two months. Two months is not enough for this report, so CC Systems bought the card after the two months. They recommended using Linux instead of Windows because it was more stable, but Windows drivers were available, so we decided to use Windows.

The hard disk was ordered from Dustin and it was delivered within a few days. When it arrived it was sent to Alfta, where they mirrored it from another hard disk. The source hard disk was installed with Windows NT 4 and had all the needed drivers .

5.3.3 Putting the Parts Together

Once all the parts were delivered and all the cables were ready, everything was put together. As mentioned in 5.3.2, the CPU board and the bottom card was sent to CC Systems office in Alfta to be modified to fit together. On top of the CPU board the video grabber board was installed. All cards were then secured with screws. All cables were attached to the monitor, keyboard and other peripherals.

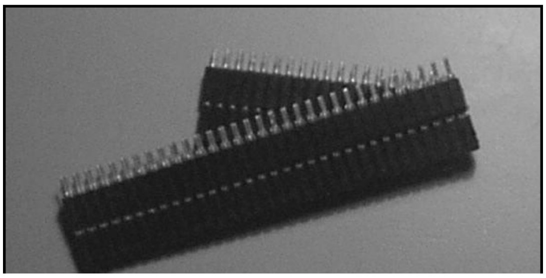


Figure 18: PC/104 distances which are used since the video connectors on the video grabber otherwise would touch the primary memory on the CPU board.

So came the moment of starting the Recorder. It was a disappointment because the recorder did not boot. Firstly a hardware failure was considered but after successful testing with a working version of Windows 98, by borrowing a laptop hard drive we were sure that the mirroring of the hard disk was unsuccessful.

The problem of installing an OS on the Recorder is that no floppy drive or CD-ROM is available. The mini-IDE cables used in laptops are also different from desktop IDE cables.

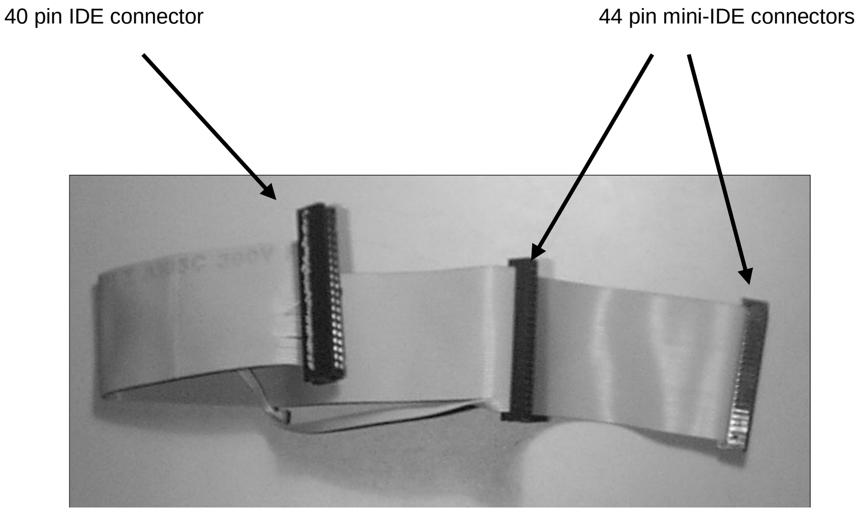


Figure 19: Modified IDE cable. A 40 pin connector was attached to enable CD-ROM access for the Recorder.

It is smaller in size and has four more contacts, where the power is supplied. Mini-IDE uses 44 pins and IDE uses 40, but the 40 first pins are exactly the same. Thus, a CD-ROM could be installed by putting an IDE contact on the mini-IDE cable. Power was fed from a desktop PC power supply. This way we were able to make a clean Windows NT 4 installation.

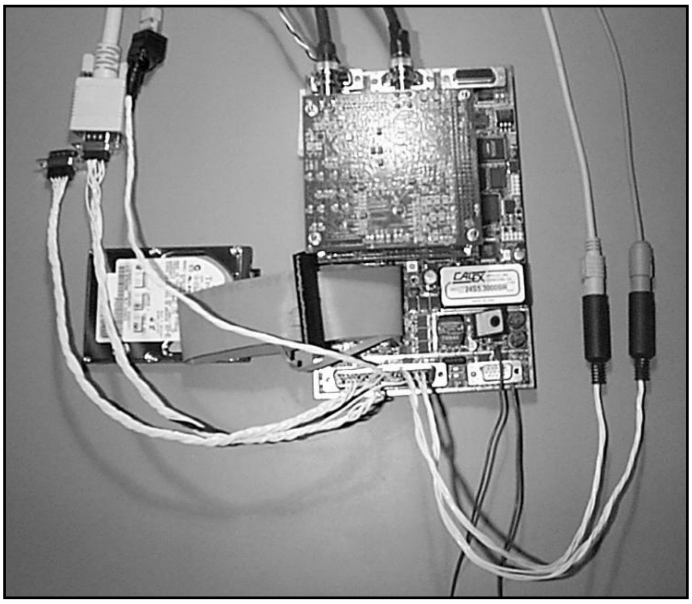


Figure 20: The complete Recorder.

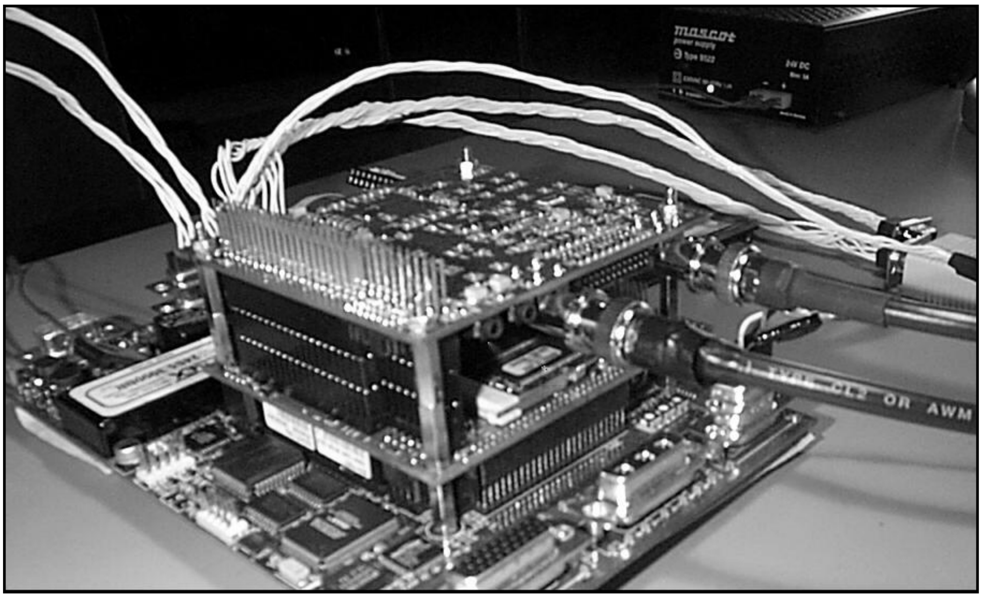
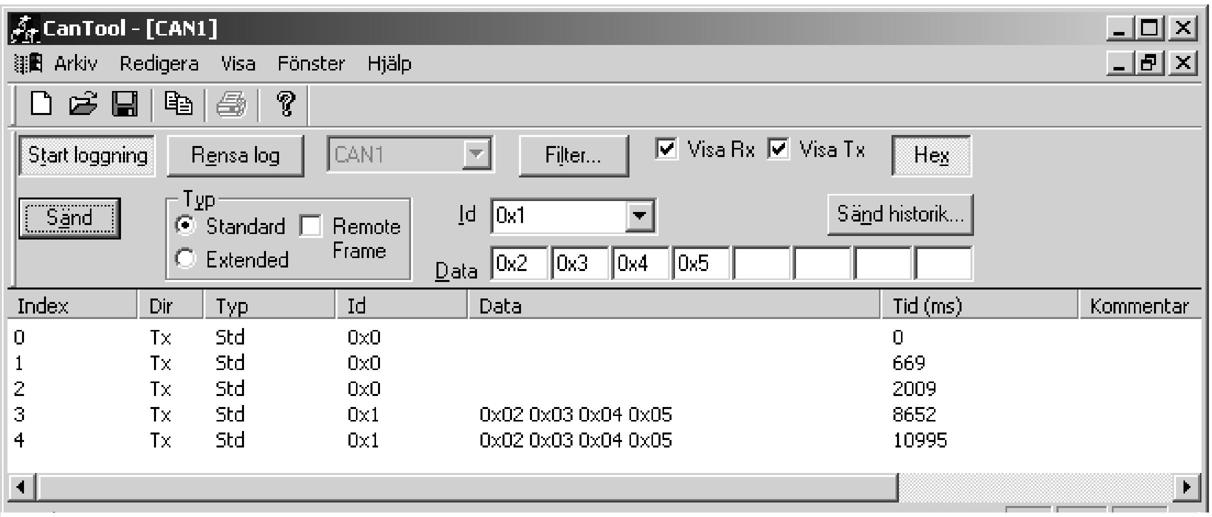


Figure 21: Different angle of the recorder.

5.3.4 Setting up the Communication

The Recorder needed a few drivers and without a floppy drive it would be convenient if a network communication could be established, especially when the software development was to begin. To start with, a serial communication, through the RS 232 interface, was set up. Using Stargate, files could be transferred to the Recorder. This made it possible to install drivers for the Ethernet controller on the Recorder. When the Ethernet was running, the hard disk was shared on the LAN.

When all the drivers were installed, the CAN network was configured. CanTool, which monitors the CAN traffic and is able to send messages, was used to test the CAN network. The speed of the CAN network cards had to be set to the same for all cards on the same network. The speed was chosen to 250000 baud, which is the bandwidth of the CV9030 vehicle.



CAN1

☑ Visa Rx ☑ Visa Tx

Typ: ● Standard ○ Extended ☐ Remote Frame

Id: 0x1

Data: 0x2 0x3 0x4 0x5

Index	Dir	Typ	Id	Data	Tid (ms)	Kommentar
0	Tx	Std	0x0		0	
1	Tx	Std	0x0		669	
2	Tx	Std	0x0		2009	
3	Tx	Std	0x1	0x02 0x03 0x04 0x05	8652	
4	Tx	Std	0x1	0x02 0x03 0x04 0x05	10995	

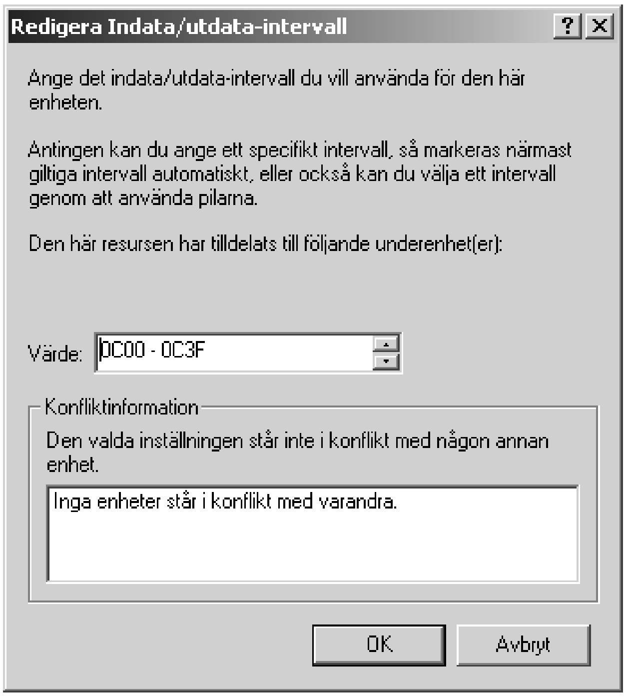
Figure 22: CanTool, by CC Systems.

5.3.5 Setting Up the Recorder Software

When Windows NT 4 was installed, drivers for the hardware had to be added. The display adapter and Ethernet are parts of the CPU board, drivers for these where found on Digital Logic's web site. The CAN functionality belongs to the bottom card and since it is designed by CC Systems, drivers where available in the office. All these were successfully installed and it was time to try the video grabber drivers. The information in the manual is vague about whether Windows NT 4 is supported or not. One page states that drivers are available for Windows 2000 and 98 only, but on another page, drivers for Windows NT and 95 are mentioned.

However, it turned out that the drivers did not work on Windows NT 4. To test if the drivers worked on Windows 2000, we installed the drivers on the developer PC.

At one point in the hardware install wizard, the option Have disk was selected and sx12.inf pointed to as the install file. This file came with the driver package and contains information about the installation. A few clicks later, the ISA IO address was selected. It should correspond to the jumper settings on the grabber board and we chose the default value.



Redigera Indata/utdata-intervall

Ange det indata/utdata-intervall du vill använda för den här enheten.

Antingen kan du ange ett specifikt intervall, så markeras närmast giltiga intervall automatiskt, eller också kan du välja ett intervall genom att använda pilarna.

Den här resursen har tilldelats till följande underenhet(er):

Värde: 0C00 - 0C3F

Konfliktinformation

Den valda inställningen står inte i konflikt med någon annan enhet.

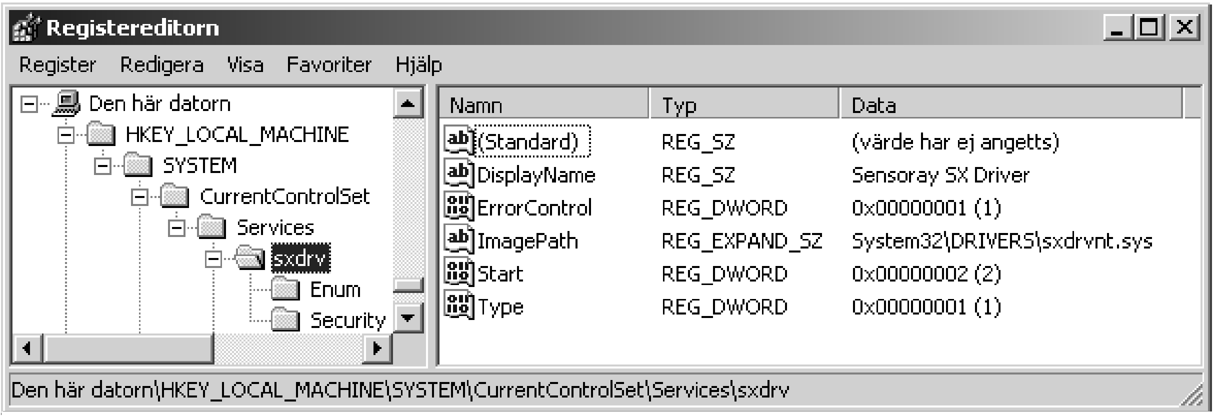
Inga enheter står i konflikt med varandra.

Figure 23: IO address selection when installing Sensoray drivers in Windows 2000.

After the Windows 2000 installation of the video grabber drivers were completed, we copied the driver files to the right directories on the NT 4 machine and copied the new Windows Registry key as well.

File	Location
sxdrvnt.sys	system32/drivers
x12.dll	Application Directory

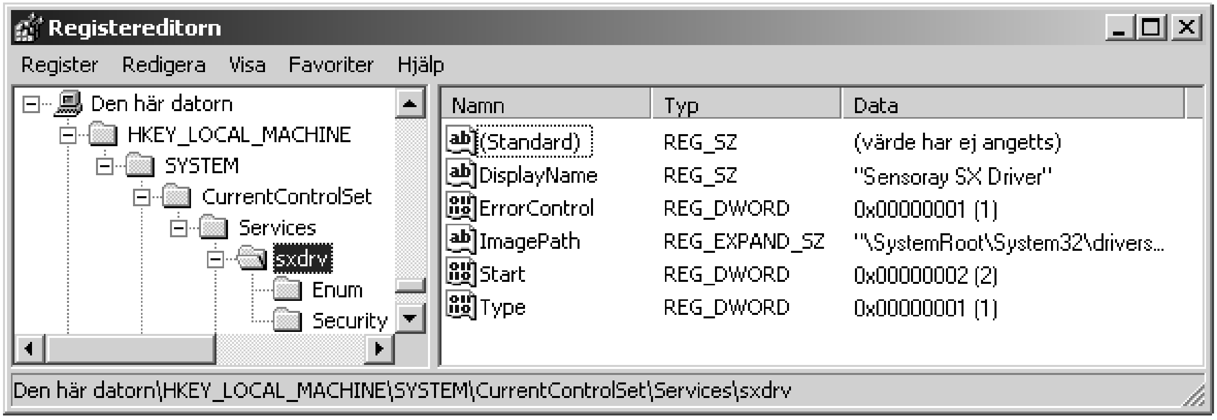
Table 5: Driver files and the directories they should be in.



Namn	Typ	Data
(Standard)	REG_SZ	(värde har ej angetts)
DisplayName	REG_SZ	Sensoray SX Driver
ErrorControl	REG_DWORD	0x00000001 (1)
ImagePath	REG_EXPAND_SZ	System32\DRIVERS\sxdrvnt.sys
Start	REG_DWORD	0x00000002 (2)
Type	REG_DWORD	0x00000001 (1)

Den här datorn\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\sxdrv

Figure 24: Registry key for the Sensoray board in Windows 2000.

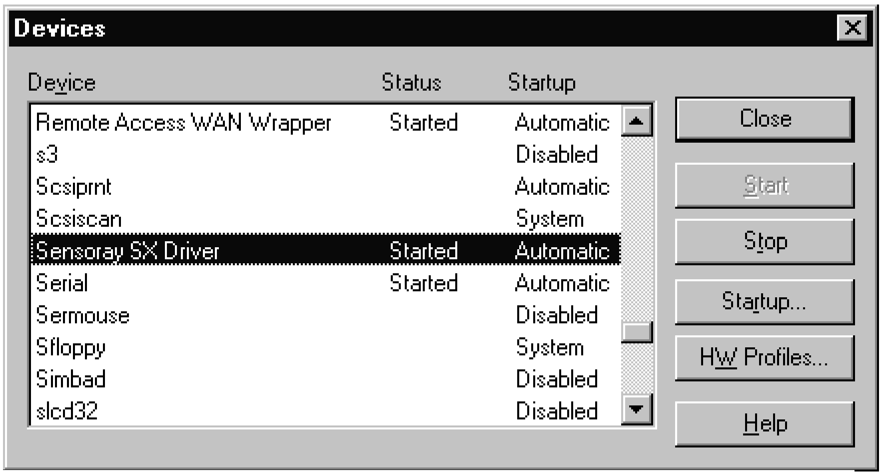


Namn	Typ	Data
(Standard)	REG_SZ	(värde har ej angetts)
DisplayName	REG_SZ	"Sensoray SX Driver"
ErrorControl	REG_DWORD	0x00000001 (1)
ImagePath	REG_EXPAND_SZ	"\SystemRoot\System32\drivers...
Start	REG_DWORD	0x00000002 (2)
Type	REG_DWORD	0x00000001 (1)

Den här datorn\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\sxdrv

Figure 25: Registry key in Windows NT. Notice the difference in how the operating systems specify their System32 path.

When this was done the driver was present and apparently working in the Devices list in Windows NT control panel. This means that the OS is able to find and run the driver, but does not guarantee that the hardware works. Even with the grabber board completely disconnected from the Recorder, the driver seemed to work.



Device	Status	Startup
Remote Access WAN Wrapper	Started	Automatic
s3		Disabled
Scsiprnt		Automatic
Scsiscan		System
Sensoray SX Driver	Started	Automatic
Serial	Started	Automatic
Sermouse		Disabled
Sfloppy		System
Simbad		Disabled
slcd32		Disabled

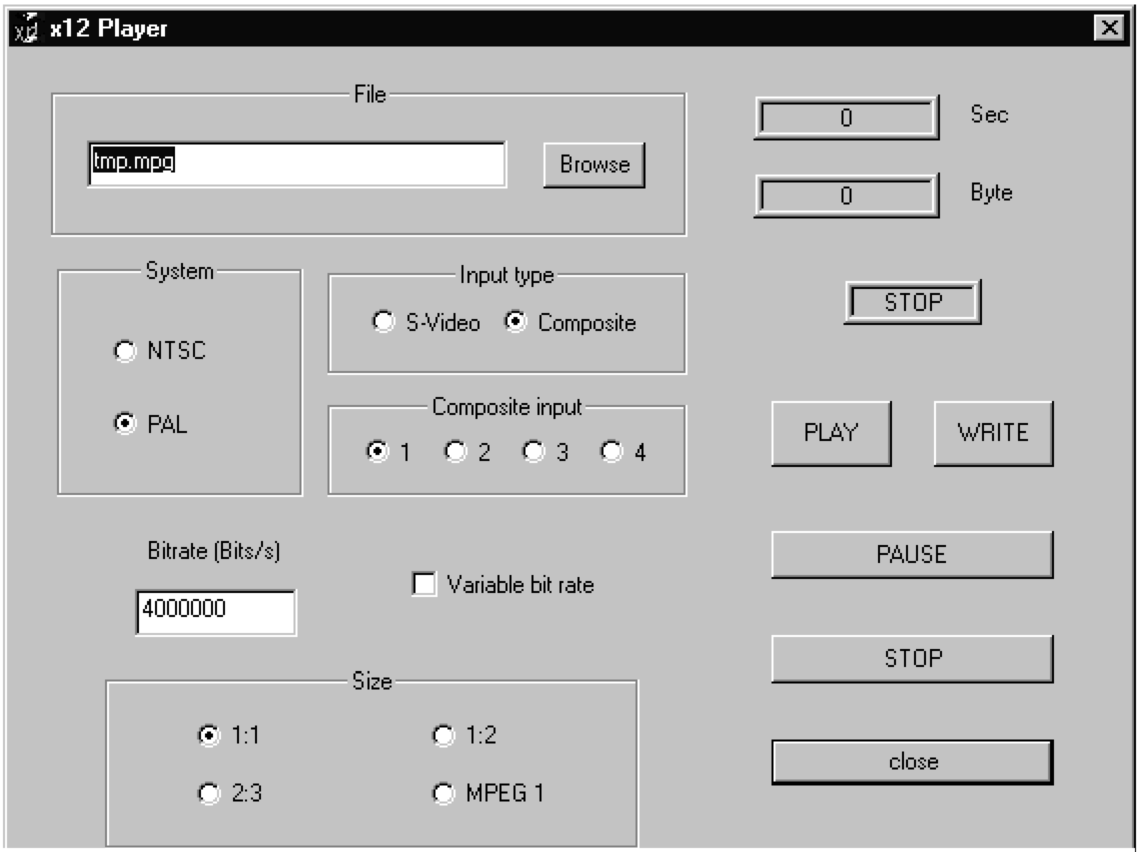
Figure 26: Windows NT. Apparently the driver is found. But this obviously does not guarantee that the device is working.

The problem was that when the sample program delivered with the Sensoray drivers was running, it stated that no grabber board was found. The reason was that we did not select any IO address for the unit. Windows 2000 prompted for it when we installed it, but on the NT 4 machine we manually copied the files and registry entries and thus never got any chance to select the address.

First we thought that the IO address information must be somewhere in the registry where we could add our grabber boards address, but no such registry key were found. By e-mailing the Sensoray support, we got the answer that the card does not work on Windows NT 4. But a few days later Sensoray sent new drivers, which worked on NT 4. The changes were that the x12.dll file was replaced and the sx12.ini file was added, which contained information about the IO address.

5.3.6 First Recording

Along with the drivers came a Win32 sample program with source code. This Win32 sample program doesn’t run in a client, only as a window. A new project was created in MS Visual Studio and the source code was added. The compilation worked when MFC was disabled. The program would not run at the beginning, as told in 5.3.5, but when all the drivers were installed correctly, the program worked.



x12 Player

File
tmp.mpg

Sec: 0

Byte: 0

System
○ NTSC
● PAL

Input type
○ S-Video ● Composite

Composite input
● 1 ○ 2 ○ 3 ○ 4

Bitrate (Bits/s)
4000000

☐ Variable bit rate

Size
● 1:1 ○ 1:2
○ 2:3 ○ MPEG 1

Figure 27: Demo program contained in the Sensoray driver package.

sx12app.c

x12demo.c

x12demo.rc

sx12.h

sx12f.h

x12demo.h

x12.ico

To test the program, we started by connecting the video output from the graphics adapter to the input on the Recorder. To enable both video and monitor output on the same time on the client, the program TVCC was started, so called Dual View. Then the output from the Recorder was connected to the input on the Hauppauge TV card and to display the incoming video on the Client screen, Hauppauge’s WinTV32 program was started. The Recorder did not output anything yet, even if it was running. When the demo program was started and the write-button was clicked, the Recorder became alive and the video signal was looped back from the Client output to the Client input, bypassing the Recorder. When the recording was stopped, the video continued to be fed through the Recorder.

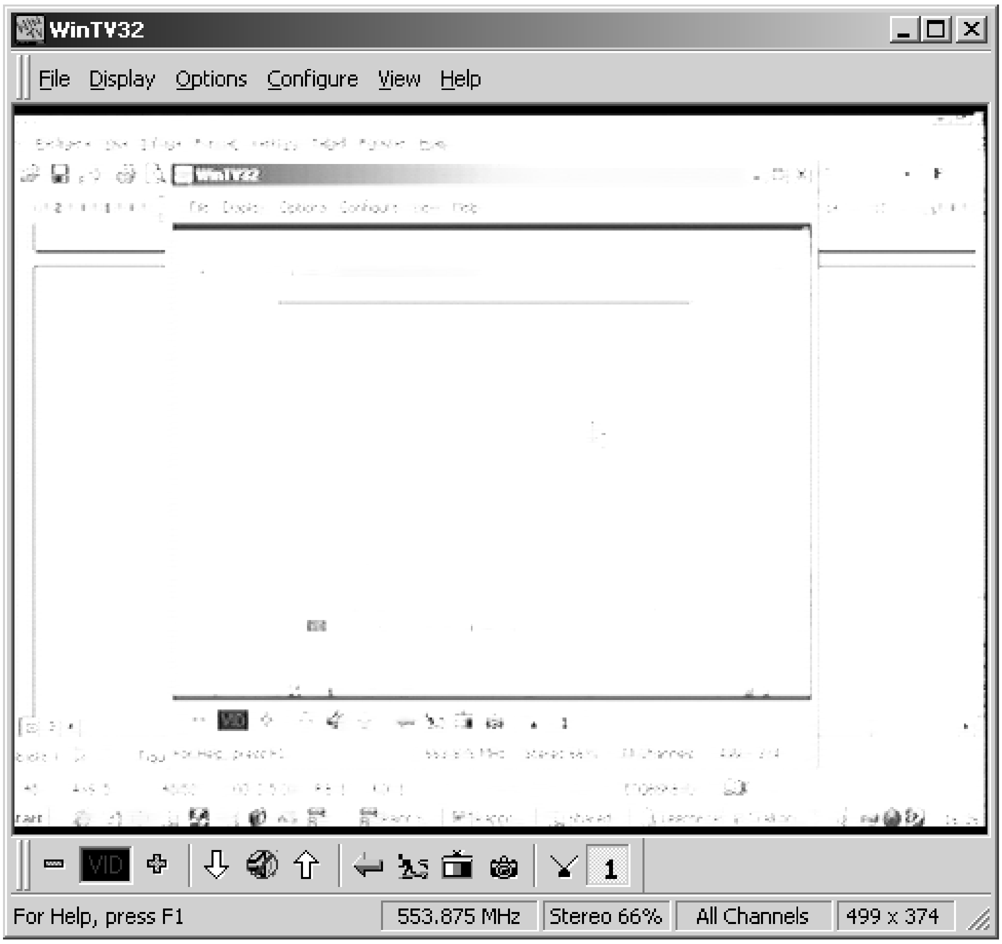


Figure 28: WinTV32 window on the Client without the resistance cable.

In the WinTV32 window, a smaller version of the Client screen was shown, and in that window, an even smaller one was shown, and so on. The problem was that every new, smaller version was brighter than the one before. By disconnecting the Recorder and directly loop back the signal from the Client output to the input on the same machine, it appeared that the error was not in the Recorder. By building a cable with a little resistance, the problem was solved, at least for the time being.

Figure 29: Resistance cable.

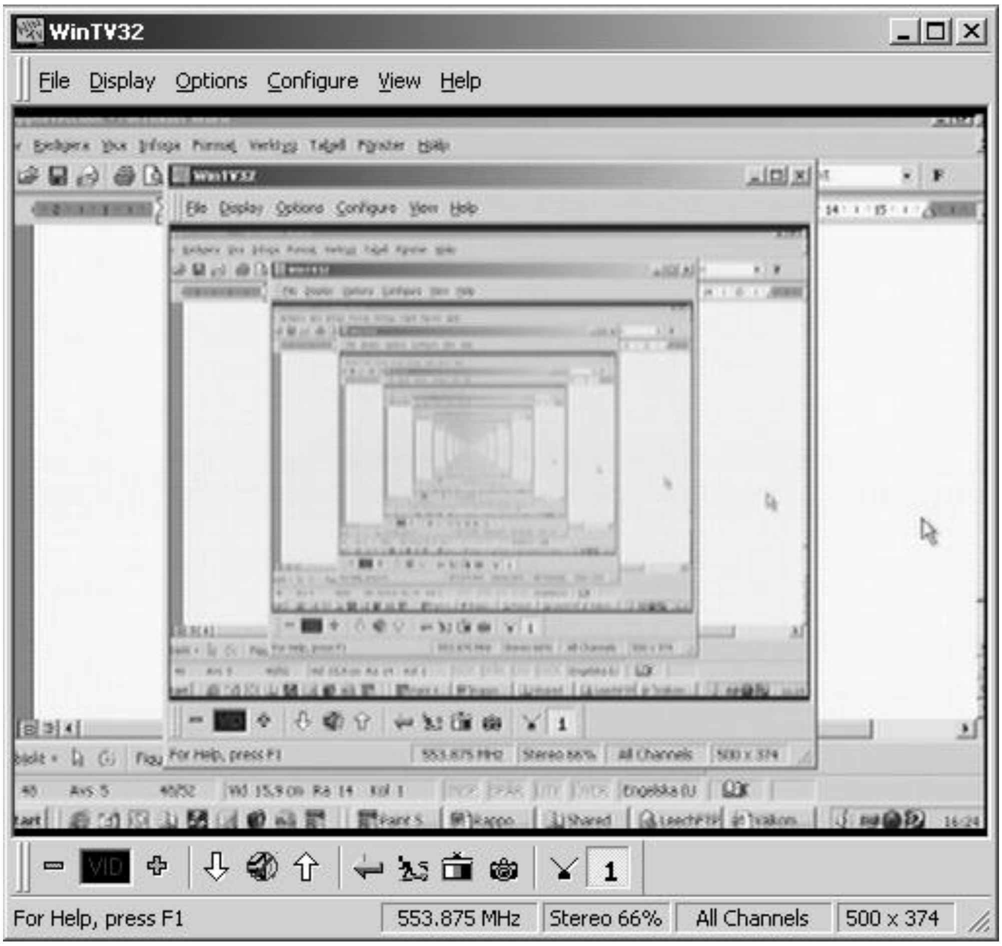


Figure 30: WinTV32 window on Client with the resistance cable connected.

When the loop-back picture looked ok it was time to test the recording. During recording the video grabber by-passes the signal back to the Client. The recording was stopped and when it was replayed it was confirmed that it worked. A few different bit rates were also tested and it was obvious that it affects the video quality a lot up to a certain level. The video file was also replayed with a DVD software decoder, PowerDVD to see if the file header information was valid, which it was.

5.4 The Client Software

5.4.1 Introduction

An important part in this report is the software development. Programs for video recording are available, but not for this particular application and there are a few different reasons for this. Firstly the hardware for the recording, the video grabber, is targeted for a rather small market. It is not compatible with desktop PCs because it uses the PC/104 standard and the manufacturer does not provide any software other than a rather simple sample program. Another reason for the lack of suitable software is that our video recorder is supposed to be controlled through the CAN bus. No human interaction devices are going to be connected to the final product. There are more requirements that make the application special, for example that it should be able to save the last minute of video data, meaning it will be recording continuously to a FIFO buffer, throwing away data older than one minute. The time span one minute is just an example, it could be everything between seconds and hours.

The requirements mean that the software in the Recorder is supposed to have functionality both to record video and to listen to the CAN bus. It should also be able to write large data blocks to the hard disk. It should have a protocol of communication, common with the Client software and be able to read CAN messages often enough, so it is quick in responding. A slow program can be a source of stress for the user.

5.4.2 Programming language

The drivers for the grabber card and the CAN controllers both in the client and the recorder are written for C or C++. This makes these candidates preferable for the Client software. A modern design is also wanted and therefore C++ is the most suitable language. For C++ there are also a few good development tools of which MS Visual C++ and Borland C++ Builder are used at CC Systems.

Since the operating system is made by Microsoft, MS Visual C++ was chosen as the developing environment. This way the chance of getting the programs to work on different versions of Windows was better. As a small bonus MS Visual C++ also supports MS Source Safe, which is used as a version management tool at CC Systems.

5.4.3 Layout

The strategy was to make functionality that was needed in both the Recorder and the Client available through a common component so that code could be reused. But there are a few differences in the type of programs used in the respective computers. The Recorder software will be run as a console application, without any GUI. The Client, on the other hand, will have a GUI with buttons and list-boxes to control the Recorder. That is another reason why the object oriented C++ language is suitable.

To get something to work from, a new project was created in MS Visual C++ using the MFC AppWizard. Two classes were generated, CClientApp and CClientDlg. The CClientApp class is the main class, which is first accessed when the program is started. It contains an instance of CClientDlg, which controls the dialog box. There the functions are located that are called when you press a button.

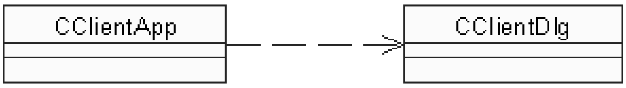


Figure 31: Class relationship after MFC AppWizard was run.

It is preferable to modify the methods in the classes generated automatically as little as possible as the code in these is rather messy compared to manually written code. The CClientApp class was left completely untouched, but CClientDlg was modified quite a bit. The modifications were mainly new functions, linked to buttons in the dialog window. The methods in the dialog class are all handling events from the dialog. For example if a button is pressed and the command needs information from a list-box, this functionality is contained in the method.

Other than displaying and handling the dialog window, the program must be able to send messages to the Recorder. This is done with two classes on two different layers. RecControl has a number of methods that are called by the CClientDlg methods. The RecControl methods mainly construct CAN messages from the information given and forward these messages to the Controller class which initiates the CAN network and sends the messages. The reason to have a separate class for the communication is that the same class is used in the Recorder and that if something in the communication layer needs a change, it is easier to have the functionality separated.

Figure 32: Relationship between CClientDlg, RecControl and Controller classes. When the user make a command in the dialog window CClientDlg passes the order to RecControl, which constructs a CAN message. This message is then sent to the Controller class for delivery on the CAN network.

The Client program is really only a control program, which does not bother at all with the actual recording. This means that it is a thin program and all functionality in it is to control the dialog and to send and receive messages through the CAN network. Apart from starting and stopping the user must have the possibility to change a few settings and to choose which file to replay and which to delete. Selecting files is done by a list-box, so when pressing for example the delete button the program must check which file was selected. It is mainly this kind of functionality in the Client program, which makes it relatively simple.

5.4.4 Classes

ClientApp

The ClientApp class was generated by the MFC AppWizard. This class starts up the application and initiates and opens the dialog window. This class is not modified at all.

ClientDlg

This class is generated by the MFC AppWizard as well, but it is not kept unchanged. ClientDlg controls the dialog window, all the buttons and other graphical components and the events generated when the user interacts with the interface. From the start it was just an empty window, but as functions were implemented at a lower level, buttons and controls for these were added in the window.

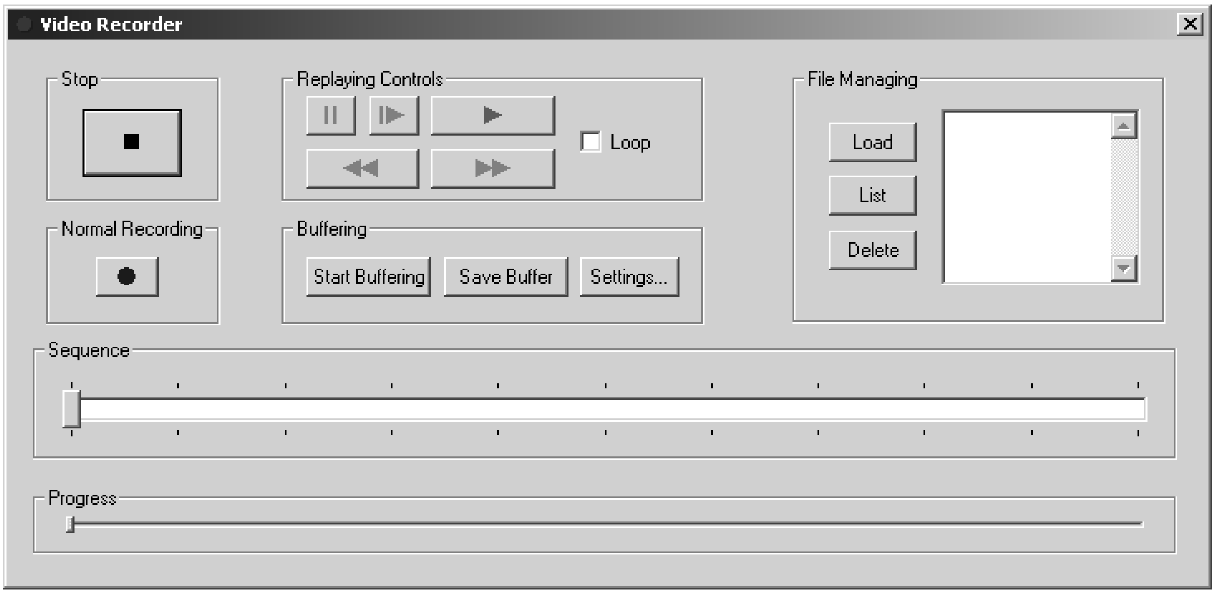


Figure 33: The ClientDlg window. This is the main GUI in the Client application.

Functions are connected to the buttons using the ClassWizard in MS Visual C++. When a new button is added to the window in the Resource Editor, this button does nothing. To bind a function to it, the button is double-clicked and then a new function is generated. If the buttons name is PlayButton, the function may be named OnBUTTONplay(). This function is generated as a member-function to the ClientDlg class and in it the functionality for the Play button is placed.

As mentioned before, the functionality should not preferably be in the dialog class. When the Play button is pressed, and the OnBUTTONplay function is called, it will take the information needed for replay and pass it to a lower-level class. In this case it will check which video file is selected, where it should start and stop and then pass it to the RecControl class. If no file is selected nothing is sent and if more than one file is selected only the first one is sent.

ClientDlg checks the state of the recorder every now and then. This is done by a timer which invokes an event every 100 milliseconds which in turn calls the OnTimer() function. This function asks the RecControl class if any message has been received from the Recorder. If the Recorder for example sends a message that a new file has been created, the OnTimer gets the message and updates the file-list in the GUI.

Controller

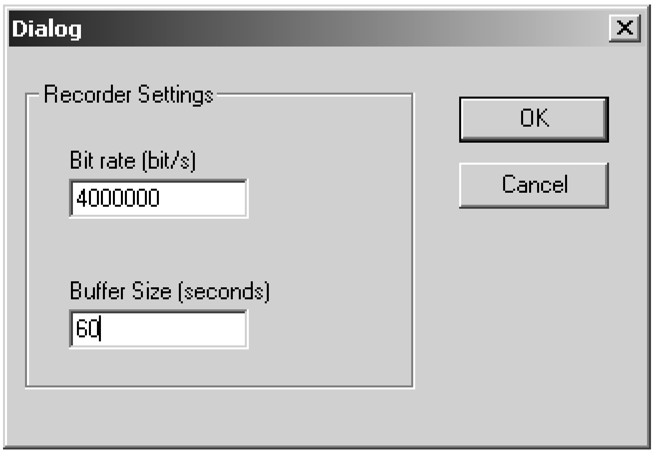
If anything in the communication requirements changes in a way that the functions that send the messages on the CAN network have to be modified, it is good to have this functionality in a separate class. The Controller class does just that. It has member functions to send and receive messages, which just sends the messages on the network. The Controller class also has an init function, which initiates the application for CAN networking.

RecControl

This class constructs all the CAN messages that are later sent to the Recorder. For example, if the user presses the Play button, ClientDlg's OnBUTTONplay() member function will be called, which in turn calls RecControl's PlayStart() function. In PlayStart() a CAN message will be created and sent to the Controller for delivery on the CAN network.

SettingsDlg

This is a class that controls a simple settings dialog. It was created using the Resource Editor in MS Visual C++. In the dialog the user can choose the length of the video buffer and the video capture bit rate.



Recorder Settings

Bit rate (bit/s): 4000000

Buffer Size (seconds): 60

Figure 34: The SettingsDlg window.

CSelectionSliderCtrl

This class was made by Pedro Pombeiro and was downloaded from the Web. It works mainly the same way as its base class, CSliderCtrl, except that it enables the user to select a range with the mouse or the keyboard. The class is instantiated and added to the ClientDlg as a control, which makes it possible to select a subset of the video clip to be replayed. [20]

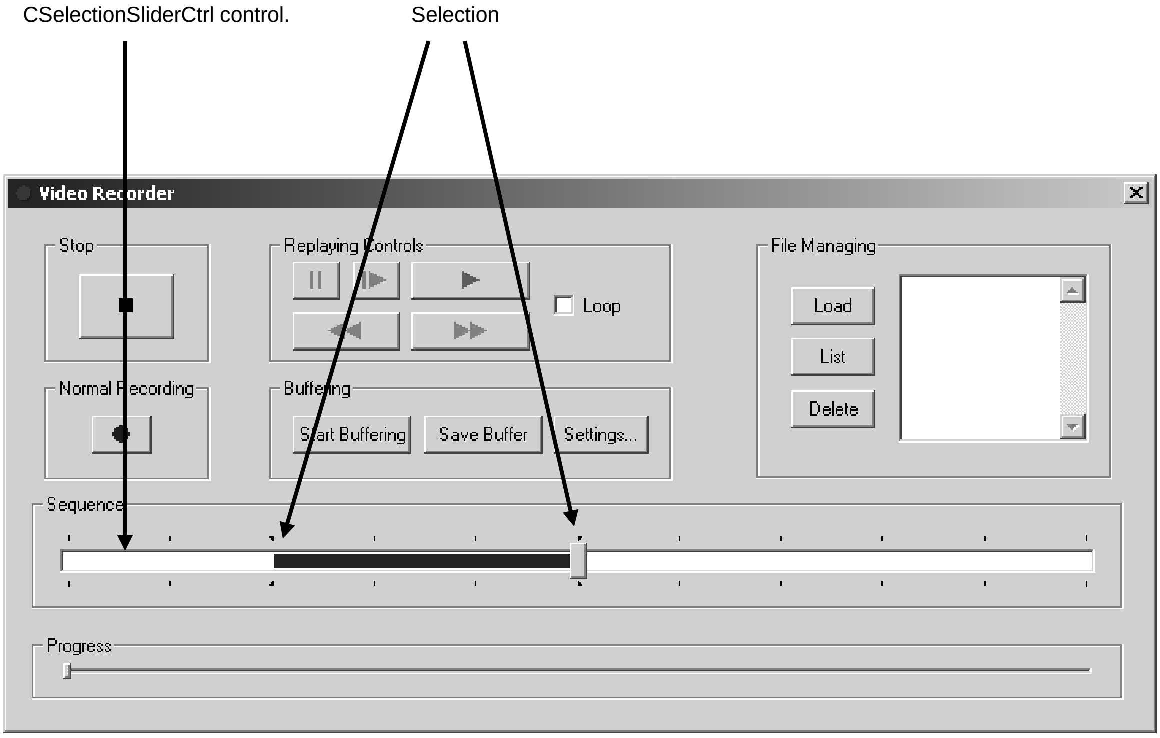


Figure 35: The CSelectionSliderCtrl as it appears in the dialog window.

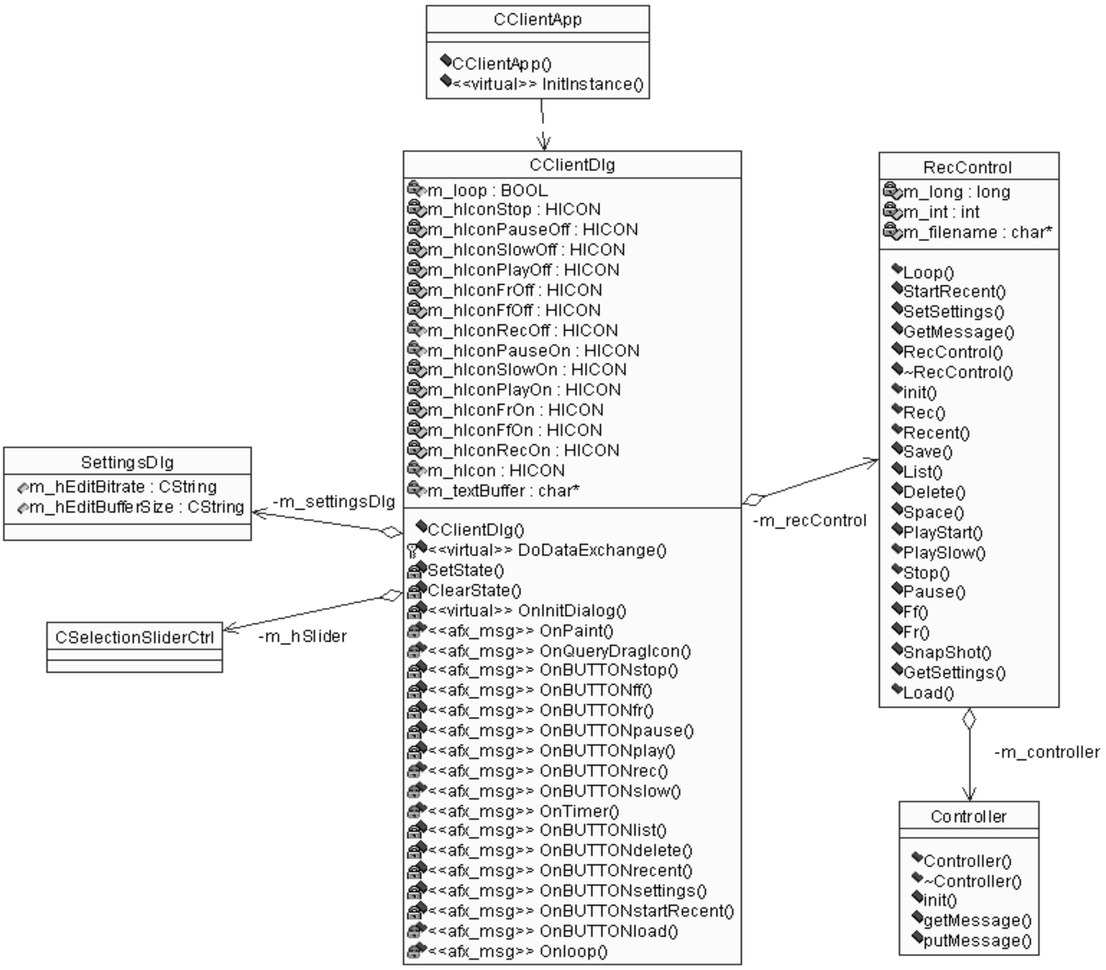


Figure 36: The Client class diagram.

5.5 The Recorder Software

5.5.1 Introduction

The Client program does not do anything more than handling the user interface. The rest of the intelligence in the Demonstrator is in the Recorder program. The Recorder is able to do the actual recording, replaying and file managing that is needed. It listens for CAN messages, which contains information about what to do. When an operation is completed, it will send a response message with the status of the Recorder. For example if the Play command is received, it will start replaying and response with the status message ‘Playing’.

Since the Recorder is supposed to be as head-less as possible, i.e. having no interaction devices connected, it doesn't need any GUI. Therefore the program is created as a Console Application in MS Visual C++. Furthermore, it will not return anything in the console window. All interaction with it is through the CAN network.

5.5.2 Layout

The Recorder consists of two classes, Controller and Recorder. The Controller is exactly the same as the Client's Controller class which manages the communication on the CAN network. It is the Recorder class that contains the recording intelligence. It consists of a main loop and a number of member functions. The loop, that is placed in the Run() function, listens to the CAN network and checks if there are any new messages. Then, depending on which state the Recorder is in, it will either read from or write to the video buffer, or simply sleep for a few milliseconds.

If the user clicks the Play-button, the Client will send a message to the Recorder. In the Recorder's Run() function, the message will be received and the function Play() will be called. In Play(), firstly the state of the Recorder is checked to make sure that replaying is possible, then the file is opened and the hardware is initialized. The reason for checking the state is that if recording is in progress, replaying is not allowed. The hardware can only do one thing at the time.

All the commands to the hardware are done through a set of functions that were included in the driver package. This way no hardware programming is needed and it really made getting the video grabber to work easy.

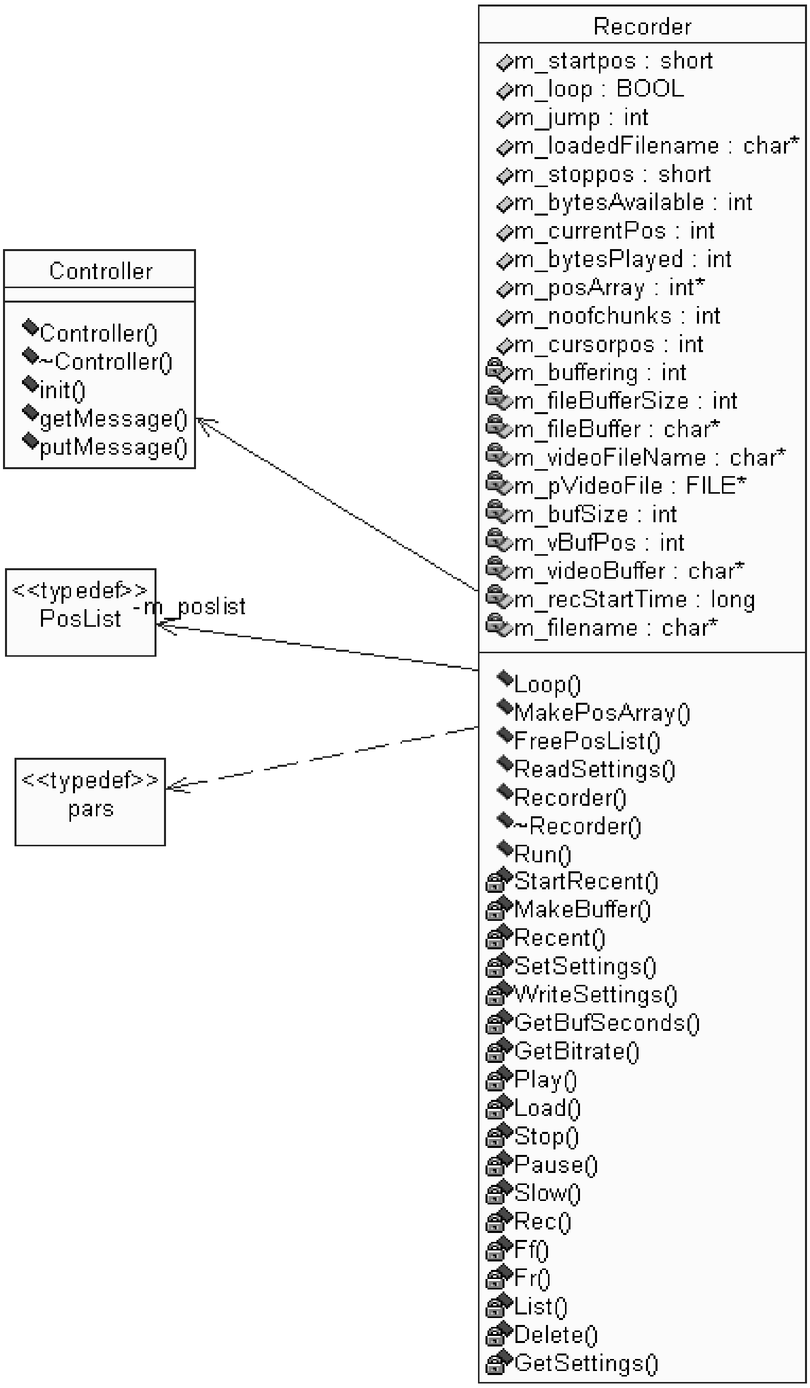


Figure 37: Class diagram for the Recorder application.

5.5.3 Classes

The Recorder is built as a console application, which reduces the number of classes needed, and the result is that only two classes were made. Their functionality is explained below.

Controller

The Controller class is exactly similar to the Controller class of the Client application (5.4.4) and so needs no further explanation.

Recorder

This is the class that takes care of the actual recording. It consists of a main loop, which calls other member functions when needed. In the loop, firstly the status of the CAN network is checked and if a new message has arrived the corresponding function will be called. In the next step of the loop the replaying or the recording takes place, depending on which state the recorder is in.

If the recorder is replaying, a chunk of video data is read into a memory buffer from a video file, then the data is written into the FIFO buffer of the video grabber. When the video grabber is in replaying mode and sees that new data is present in the buffer, it will decompress this data and display it on the video-out connector. Then the FIFO buffer is being emptied.

The recording mode works in a similar fashion, but in reverse order. Firstly the FIFO buffer is being read and if data is available, it is written to a video file. The recorder only uses one FIFO buffer and is so able to either record or replay one at a time only. If during replay, data is fed to the FIFO buffer, the recorder will halt or produce nonsense data.

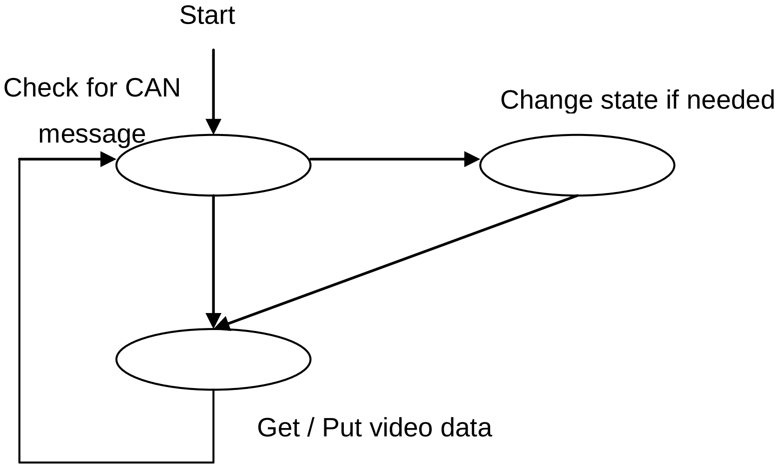


Figure 38: The main loop, simplified. If the CAN message 'start replay' is recieved, a function is called and the state is changed to 'replaying'.

5.6.1 Communication

Both the Client and the Recorder are continuously listening to the CAN network. This they accomplish in different ways. The Client uses a timer that invokes a function ten times a second that reads CAN messages. The reason for using a timer is that the Client application doesn't have any main loop, since it is an event-based application. The Recorder does have a main loop and therefore checks the message status during the loop-cycle. To illustrate how a message is sent through the system, a sequence diagram was made, see Figure 39. In the figure the user makes a Play command. The figure actually represents only half of the communication during this type of command. In reality a response message are sent as well, but it works according to the same principles, and was therefore excluded from the figure.

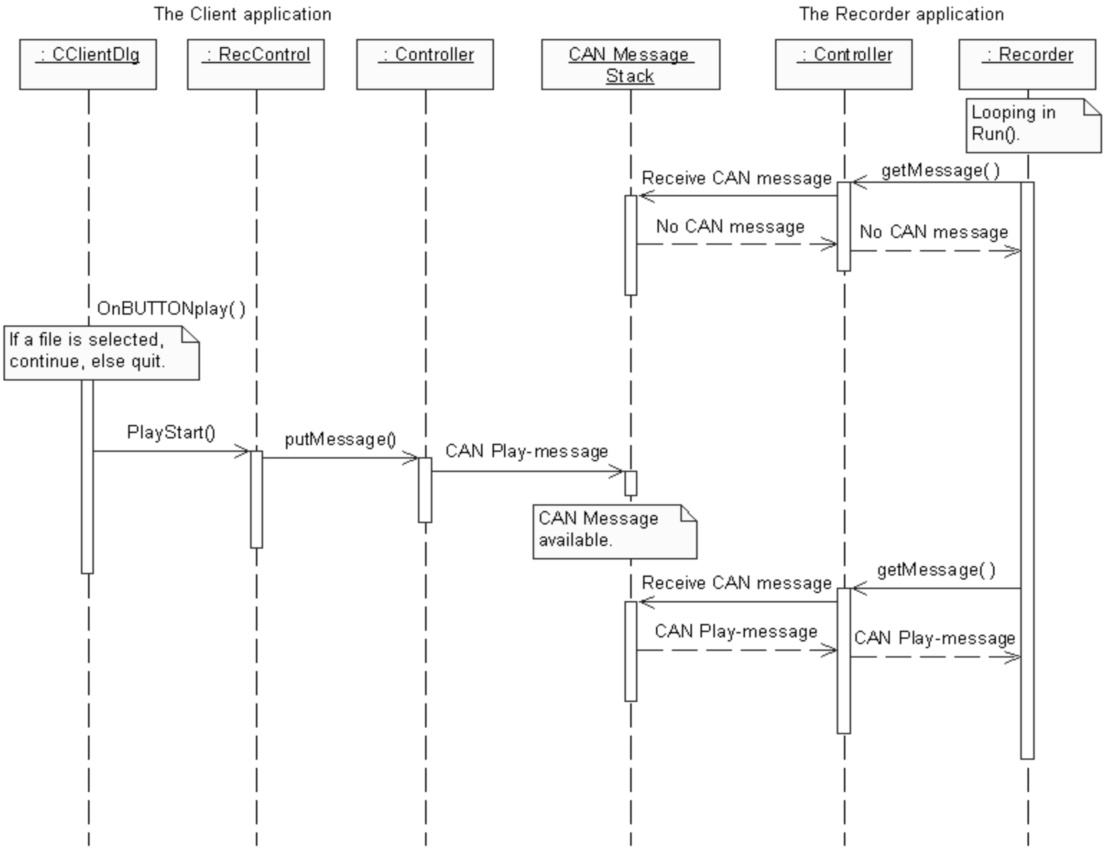


Figure 39: Sequence diagram for the use case where the Play button is clicked. It looks basically the same for the response sent back when the replaying has started.

5.6.2 Communication protocol

A protocol for communication between the Client and the Recorder was created. It is made as a C++ header file with defines for different messages and also contains a few different statuses for the Recorder.

```
// common.h
#define NODE_NAME "VREC"
// commands to the recorder...
#define CMD_VREC_REC_START             1       // Starts recording
#define CMD_VREC_REC_RECENT            2       // Stores last minute of video

#define CMD_VREC_SAVE                  3       // Save current video
#define CMD_VREC_LIST                  4       // List video clips
#define CMD_VREC_DELETE                5       // Delete video clip
#define CMD_VREC_SPACE                 6       // Report free space

#define CMD_VREC_PLAY_START            7       // Starts replay of a video clip
#define CMD_VREC_PLAY_SLOW_START       8       // Starts slow replay
#define CMD_VREC_STOP                  9       // Stops replay or record
```

Figure 40: A few of the commands in the protocol.

Chapter 6 Results and discussion

The result of this study was a working recording unit with software enabling it to be remotely controlled by another computer. Software for the Client computer was also made and the system works as planned. The requirements, suggested in Chapter 1, were fulfilled, except for the ability to export the data. It is now only possible to access the data through the Ethernet connection.

The Sensoray video grabber records and replays video with quite good quality. When recording at 4 Mbit/s bit rate, it is not possible to distinguish between original video and recorded video. But lowered to 2 Mbit/s, there was a visible change in quality.

Since the replaying works by feeding MPEG-data into a buffer it is not that easy to control the flow of the video. Slow motion was implemented by feeding data into the buffer slower, forcing the frame grabber to pause the replaying. But the pauses vary a little and the slow motion is not that smooth. It works however, and it is possible to pause the slow replay at any time, freezing the image for examination.

Sometimes when recording video, the image has strange colors and this is corrected by restarting the Recorder computer. This is probably a bug in the grabber card but could also be the result of the new drivers.

The bit rate settings seem to be correct, recording in 4 Mbit/s results in 400 KB/s data. The bit rate settings can be set between 100 Kbit/s and 10 Mbit/s, but the usable range is between 2 - 4 Mbit/s

Chapter 7 Suggestions for future work

The demonstrator that was made is a mere prototype and it is a long way to a final product. Decisions about which hardware to use had to be made and this sorted out potentially good alternatives. To be sure that the best road is taken when continuing to develop the Recorder some other alternatives may have to be examined as well. For this demonstrator inter-frame video compression was chosen, due to its superior movie quality, but it made for example freezing the video harder. With intra-frame compression it would be easier to extract single images and to replay the video in slow motion.

If the priority of the video quality remains high, the current MPEG-based compression is still the best alternative. If single image extraction is important an additional video grabber could be installed, in parallel with the current one. This grabber could be specialized in still images and would be able to deliver high quality images, while the other video grabber still would deliver good video.

The Recorder is supposed to work without keyboard or monitor and this calls for installing for example Windows NT Embedded instead of Windows NT. Since NT Embedded, when installed and configured, should work exactly as NT it should not be any problem.

The performance of the Recorder need to be better than it is right now, because of the frames that get lost when saving the video buffer. This could be accomplished by installing a faster CPU, but even better, the software could be optimized to handle the problem.

Chapter 8 References

[1] New Life for S-Video Connections Through DIY
Blackburn, 2000
http://www.soundstage.com/video/columns/200009_db_svideo.htm
Verified 2008-02-12

[2] Details of the SCART interface
Rebel, 1996
http://utopia.knoware.nl/users/eprebel/SoundAndVision/Engineering/SCART.html
Verified 2008-02-12

[3] TV Systems: A Comparison
King, 1995
http://www.ee.surrey.ac.uk/Contrib/WorldTV/compare.html
Verified 2008-02-12

[4] Building a Digital Video Capture System – Part I
Wright, 2001
http://www.tomshardware.com/2001/05/24/building_a_digital_video_capture_system_/
Verified 2008-02-12

[5] Understanding 4:1:1 and 4:2:2 Sampling Rates
Studio 1 Productions, 2002
http://www.studio1productions.com/Articles/411samp.htm
Verified 2008-02-12

[6] QuickTime File Format Specification, 2007
http://developer.apple.com/documentation/QuickTime/QTFF/qtff.pdf
Verified 2008-02-12

[7] H.261 Video Coding
Dr. Cherriman, 1996
http://www-mobile.ecs.soton.ac.uk/peter/h261/h261.html
Verified 2008-02-12

[8] Industry Standard Architecture
Wikipedia, 2007
http://en.wikipedia.org/wiki/Industry_Standard_Architecture
Verified 2008-02-26

[9] Extended Industry Standard Architecture
Wikipedia, 2008
http://en.wikipedia.org/wiki/Extended_Industry_Standard_Architecture
Verified 2008-02-26

[10] Peripheral Component Interconnect
Wikipedia, 2008
http://en.wikipedia.org/wiki/Extended_Industry_Standard_Architecture
Verified 2008-02-26

[11] SCSI and IDE: Defining the Differences
Windows IT Pro, June 1997 edition
Daily, 1997

[12] Ultra ATA/100 FAQ
Seagate, 2002
http://www.seagate.com/support/kb/disc/ultra_ata100.html
Verified 2008-02-26

[13] Serial ATA, A comparison with Ultra ATA Technology
The Serial ATA International Organization (SATA-IO)
http://www.sata-io.org/documents/serialata%20-
%20a%20comparison%20with%20ultra%20ata%20technology.pdf
Verified 2008-02-26

[14] Mechanical Hard Disks vs. Solid State Storage Solutions
Memory Digest, 2001
http://www.smartmodulartech.com/memdigest/sept2001/html/sept_2001_hot_topic.html
Verified 2008-02-26

[15] PC/104 Embedded Systems FAQ
Arrick Publishing
http://www.controlled.com/pc104faq/
Verified 2008-02-26

[16] Controller Area Network (CANbus)

Schofield, 2006

http://www.mjschofield.com/index.htm

Verified 2008-02-26

[17] Parallel port explained

B&B Electronics Mfg Co, 2002

http://www.bb-elec.com/tech_articles/parallel_port_explained.asp

Verified 2008-02-26

[18] USB 2.0 Specification

USB Implementers Forum, Inc.

http://www.usb.org/developers/docs/

Verified 2008-02-26

[19] FireWire: Getting fired up!

Filmandvideo.com, 2002

http://www.filmandvideo.com/New%20Pages/article6.html

Verified 2008-02-26

[19] A Slider Control For Range Selection

Pombeiro, 2002

http://www.codeguru.com/controls/SelectionSliderCtrl.shtml

Verified 2008-02-26