The USB I/O-Board VELLEMAN K8055 working with eCS

by **Uwe Hinz**, 2009
uhdrelb@t-online.de

Introduction

- The number of USB devices out there is huge !
  For OS/2 or eCS only a small number of them is usable.
  A lot of USB drivers are still missing...
  In the commercially available OS/2 - DAC software SCADA ONSPEC 6.1 [4][5] some Digital I/O may be supported... It has not been tried yet !

- This presentation is a successor to
  „**Digital I/O with eComStation via the USB-Interface meM-PIO**“,
  held at DWS2008 in Düsseldorf.

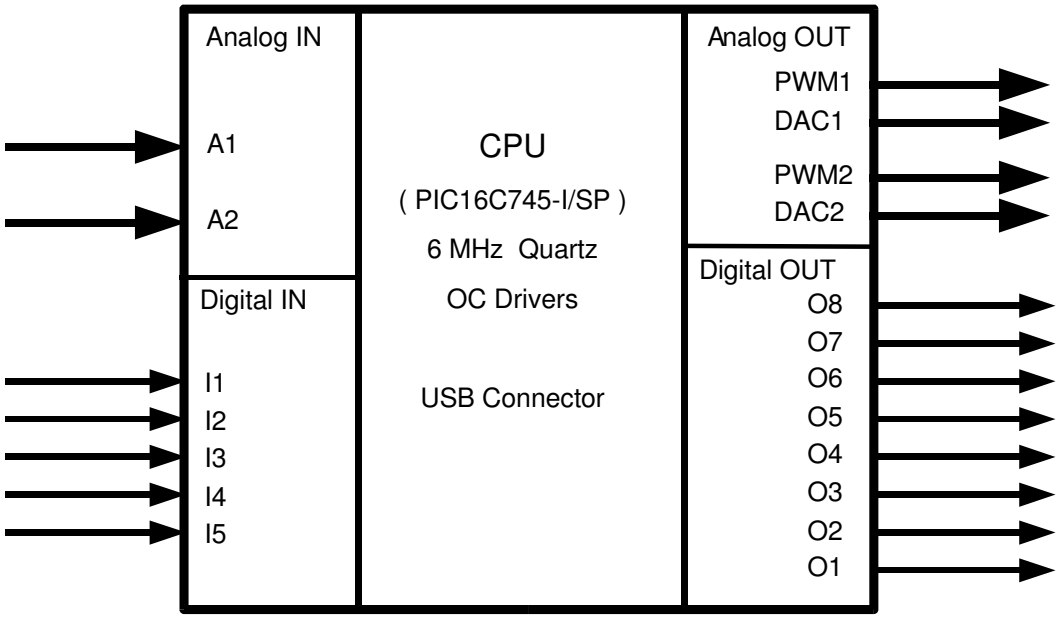
- It still puts the focus on DAC hardware for OS/2 or eCS,
  but right here the focus is on the:

USB Experimental Interface Board VELLEMAN K8055.
aka VM110

The USB I/O-Board VELLEMAN K8055 working with eCS

by Uwe Hinz, 2009

What does the Interface Board offer ?



- Designed for very common tasks, the K8055 can deal with TTL and CMOS on its digital inputs.
- Open Collector outputs allow **the designer** to go beyond TTL and CMOS.
  Loads that can be driven by up to 24 Volts can be connected.

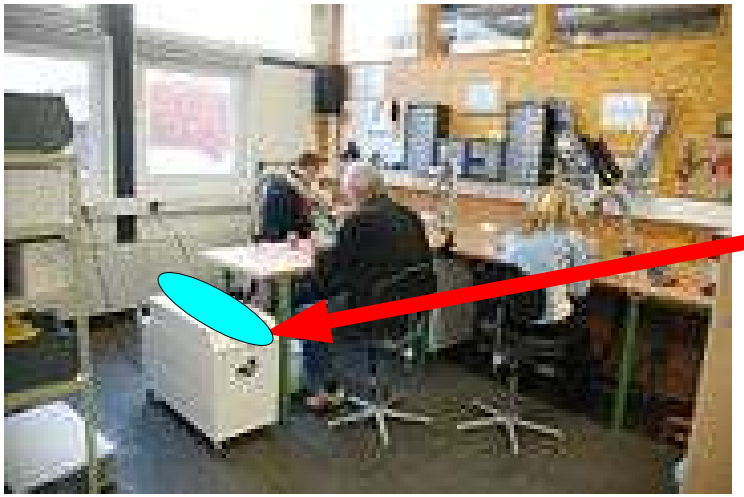
VELLEMAN

K8055

aka VM100

90 x 150 mm

PC as a Design Tool running with eCS



The USB I/O-Board VELLEMAN K8055 working with eCS

by Uwe Hinz, 2009

What can be done with K8055 ?

Three lists of possible tasks. Some items work **with extra hardware** only and some do it **Out of the Box**

Electronic's Lab

- Sensing Digital Signals
- Counting Events
- Small Logic Analysers
- Simple ON and OFF switching
- Stepper Motor Controllers
- (slow) Square Wave Generators

- Simple Digital Voltmeter (up to 5 Volts)
- Thermometer with USB
- Driving a Relay Bank
- Dimming LEDs and little Light Bulbs
- Driving Gauges

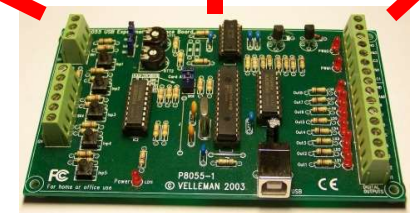
- USB – Free Style Interface Converter
(Controlling Devices with old Interfaces)

House Control

- Sensing Doors or Windows
- Admittance Check
(Counting visitors or cars)
- Controlling Blinds
- Thermometer
- Simple Weather Station
- Light Control
- Water Sensor
- Simple Gas Sensor

Education, Science

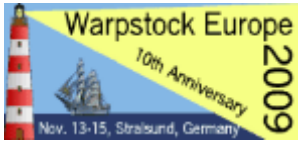
- Experiment Control
- Data Acquisition
- Science Fair Equipment
- Games with USB Interface
- Design Competition
- Programming Lessons
- Web Applications with
Remote Control Functions
- Computer AddOns (any OS)
- Integrating K8055 into
extendable devices
- Robotics

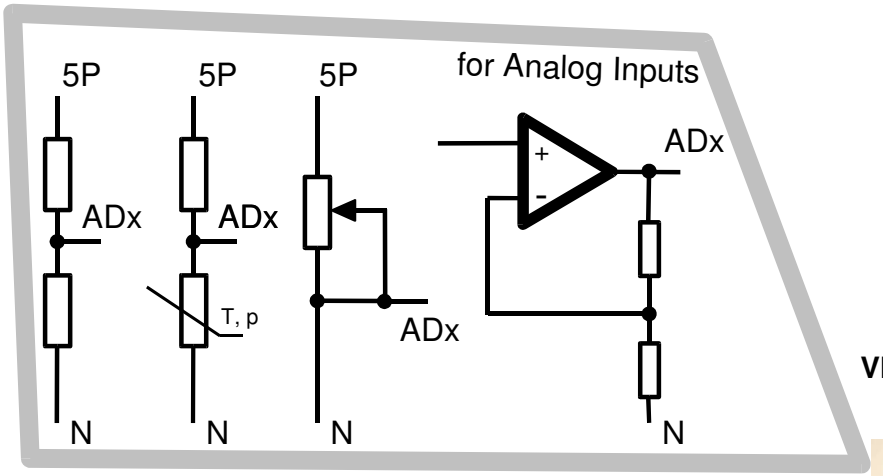


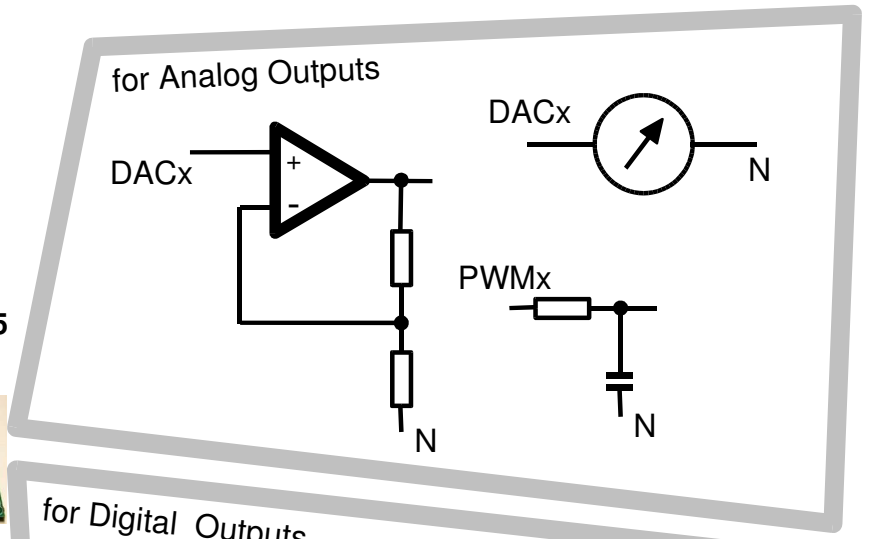
The USB I/O-Board VELLEMAN K8055 working with eCS

by Uwe Hinz, 2009

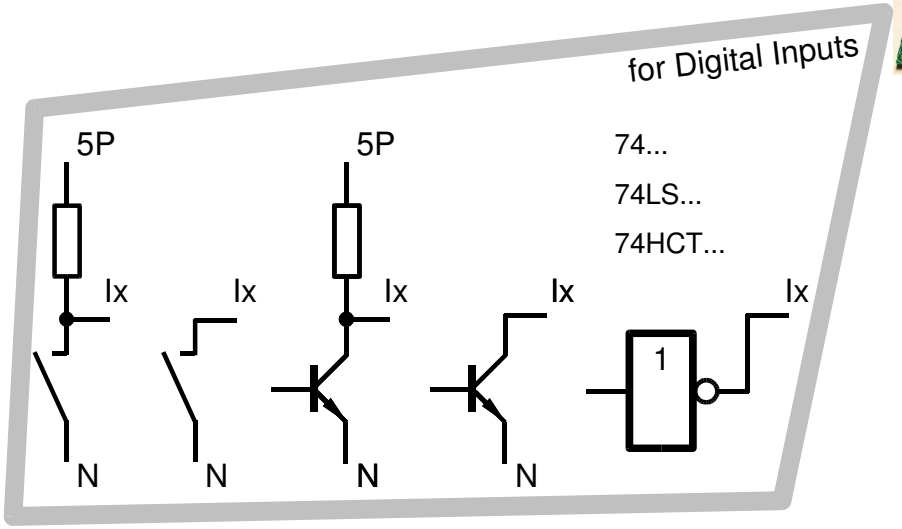
Some circuitry that can be used with K8055

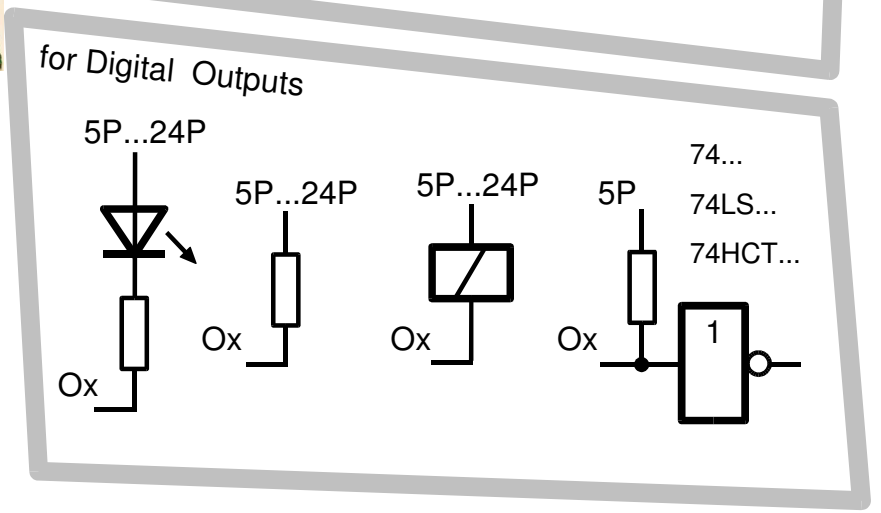






VELLEMAN K8055



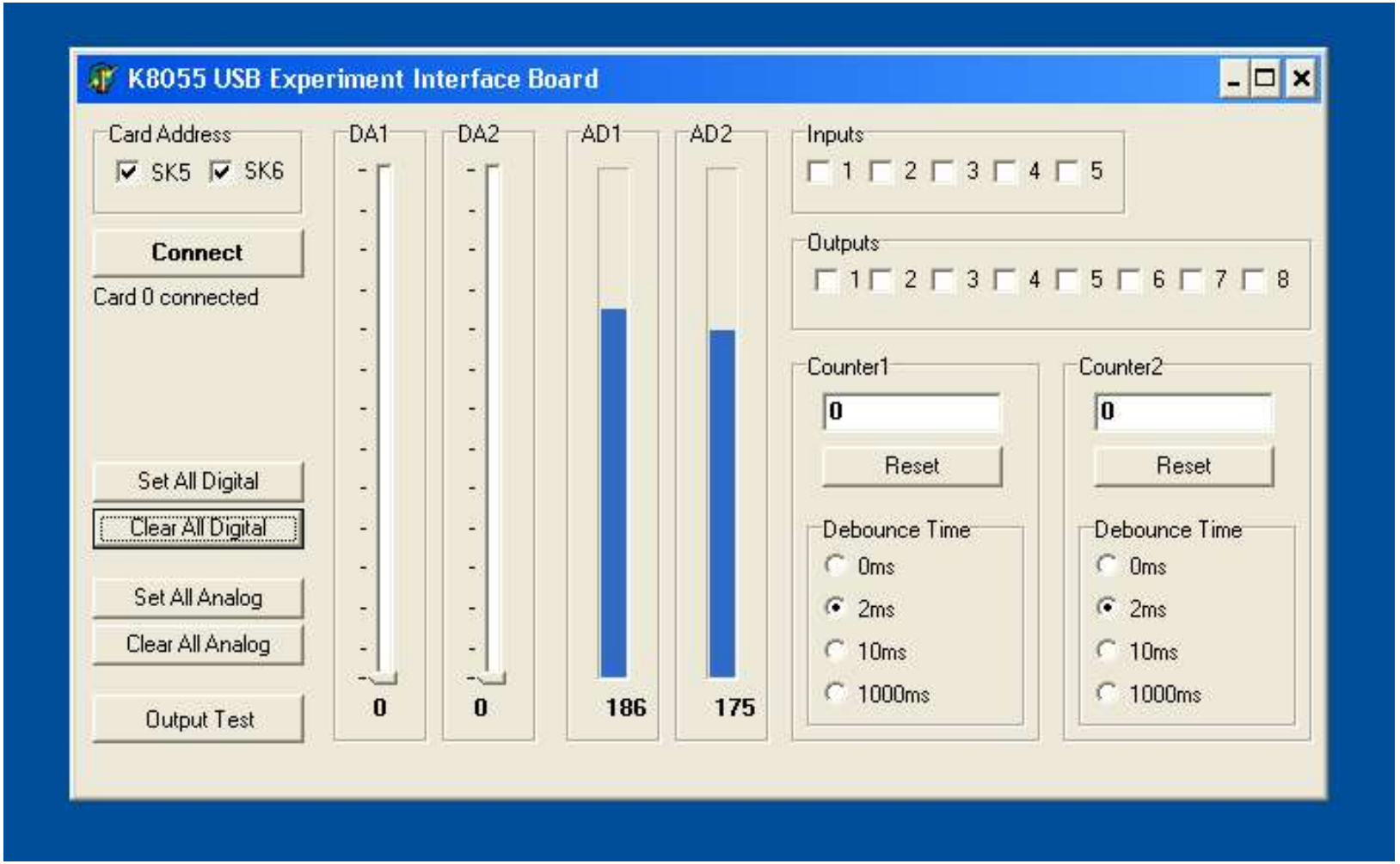


Don't ignore components that are already on the K8055 PCB !

Don't overload anything on the K8055 PCB !

by Uwe Hinz, 2009

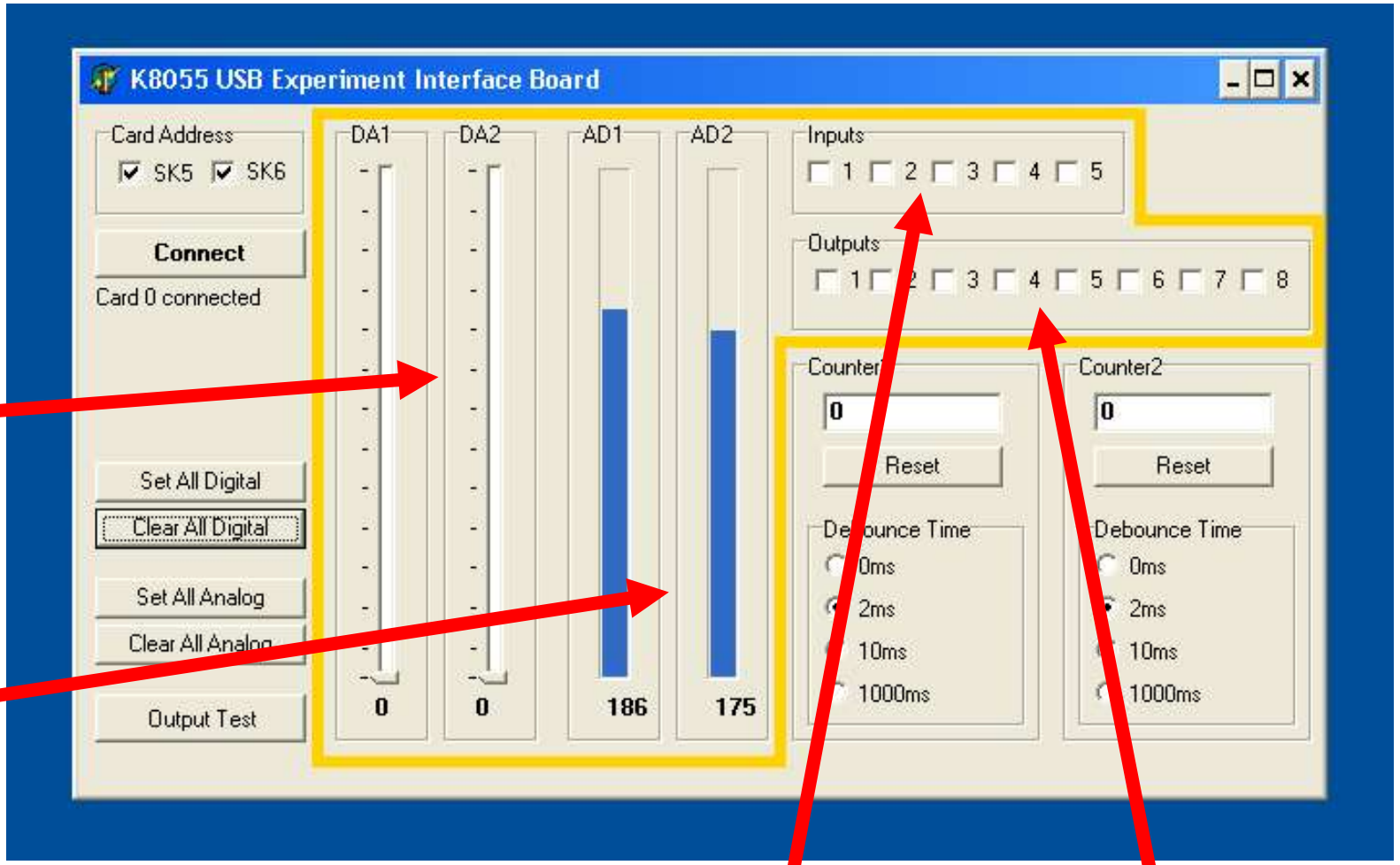
The K8055-Demo application in Windows as a template



The USB I/O-Board VELLEMAN K8055 working with eCS

by Uwe Hinz, 2009

The K8055-Demo application essentials



2 Analog Outputs

2 Analog Outputs

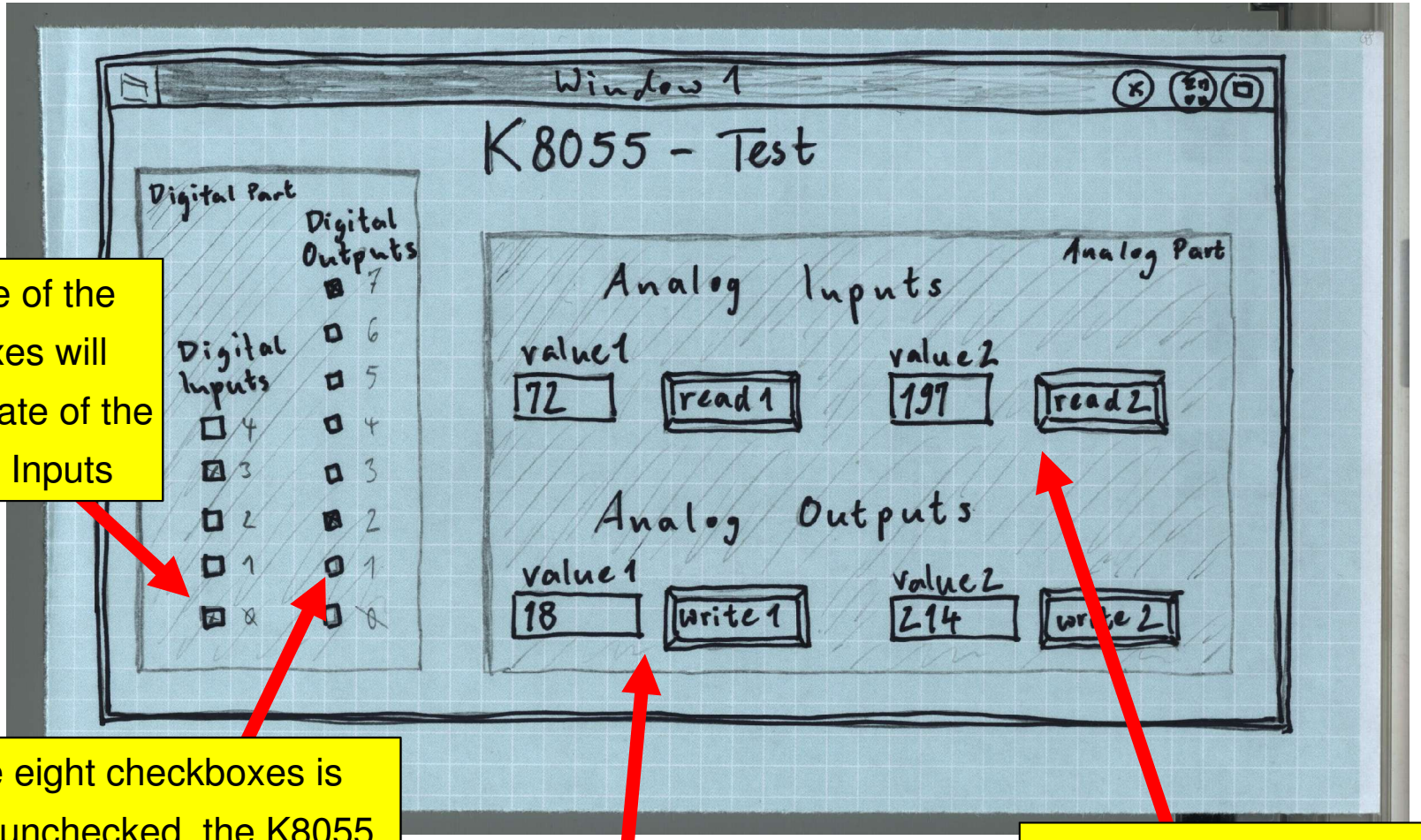
5 Digital Inputs

8 Digital Outputs

The USB I/O-Board VELLEMAN K8055 working with eCS

by Uwe Hinz, 2009

Just some pice of paper from the sketch-book



A click on one of the five checkboxes will display the state of the K8055 Digital Inputs

If one of the eight checkboxes is checked or unchecked, the K8055 Digital Output shall change level

A value from 0..255 is typed into **value1** and clicking on **write1** must send it to K8055

Clicking on **read2** should read from K8055 and show the reading in **value2**

Warpstock Europe 2009 10th Anniversary Nov. 13-15, Stralsund, Germany

by Uwe Hinz, 2009

The Driver Problem

Without a Device Driver, no application can get in contact with the

USB I/O-Board VELLEMAN K8055 ,

everyone knows that !

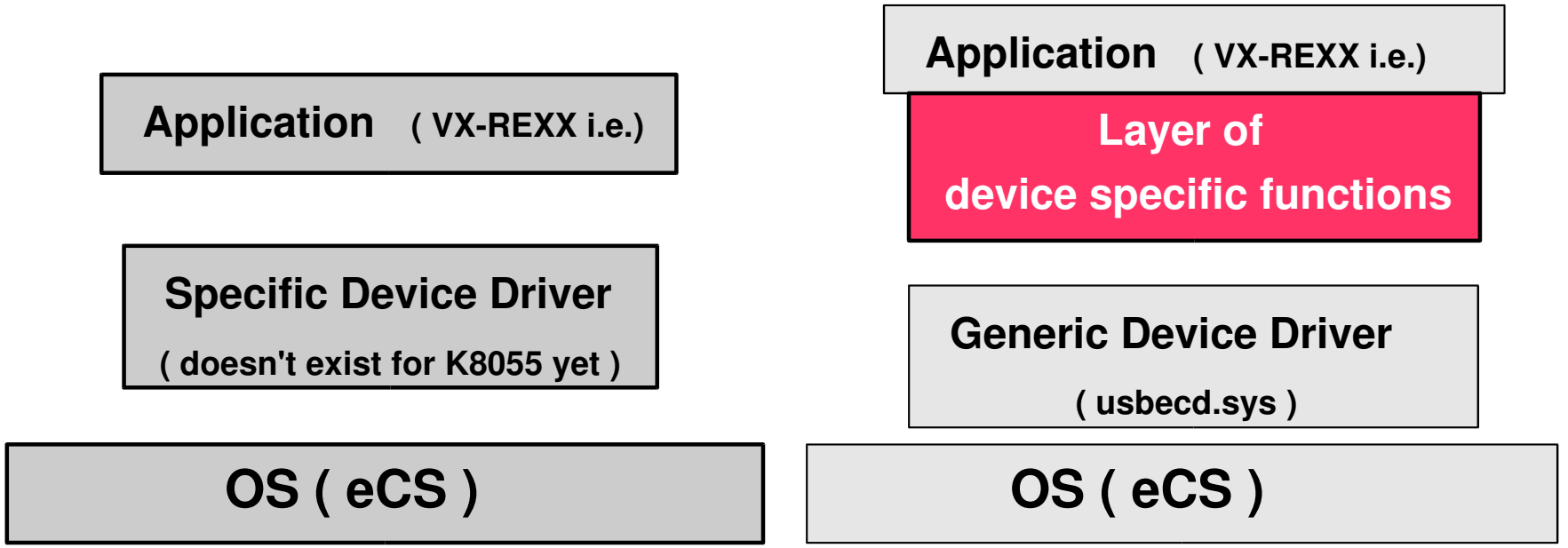
How to get a driver is still the most critical question !

by Uwe Hinz, 2009

Wim Brul's USB device driver a good choice

K8055 shall be used with the legendary **Wim Brul Driver [1]**
'usbecd.sys' .

This driver is generic !

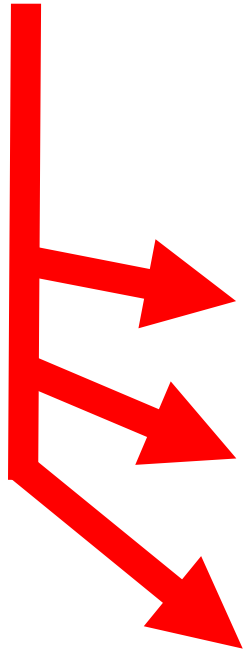


The USB I/O-Board VELLEMAN K8055 working with eCS

by Uwe Hinz, 2009

Wim Brul's device driver 'usbecd.sys' downloadable from hobbes

/pub/os2/system/drivers/misc



usb_uhci_fix01.zip	USB host controller driver (VIA and Intel chipsets) [More info]	1999/11/03	Compressed archive, 15.70Kb
usbecd00.zip	USB 1.1 Expanded Control Driver [More info]	2005/03/18	Compressed archive, 7.32Kb
usbecd10.zip	USB 1.1 Expanded Control Driver [More info]	2005/12/12	Compressed archive, 8.47Kb
usbecd10s.zip	USB 1.1 Expanded Control Driver - source/samples [More info]	2005/12/12	Compressed archive, 63.76Kb

by Uwe Hinz, 2009

List of files in 'usbecd10.zip'

```
The volume label in drive C is VOLUME 3.
The Volume Serial Number is 2933:BC14.
Directory of C:\Desktop\work\USBecdDL\usbecd10

26.05.07 23.18          <DIR>      0 ----  .
26.05.07 23.18          <DIR>      0 ----  ..
11.04.05 10.31           1.875     0 a---  usbecd.sys
 6.12.05 14.25          18.668     0 a---  usbecd.txt
 6.12.05 19.14           2.062     0 a---  usbread.cmd
 6.12.05 19.13           9.635     0 a---  usbwrite.cmd
        6 file(s)      32.240 bytes used
                    1.499.275 K bytes free
```

the driver

the readme

the examples

by Uwe Hinz, 2009

Starting with no knowledge about K8055

Three fundamental questions

1. How does the operating system (eCS) know K8055 is there ?
2. How does K8055 start working ?
3. What kind of information is needed to write to K8055 and what is read from it ?

by Uwe Hinz, 2009

Wim Brul's USB device driver and eCS

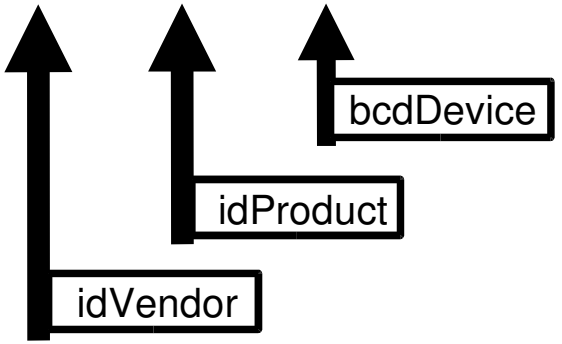
1. How does the operating system (eCS) know K8055 is there ?

The way to use the driver goes here:

Put in **config.sys**, a set of three numbers complete the D parameter.
The three hex numbers make the device identifiable.

```
DEVICE=C:\USBDRVS\USBECD.SYS /D:XXXX:XXXX:XXXX /N:$$$$$$$$ /S /V
```

All parameters explained in **usbecd.txt**




```
DEVICE=C:\USBECD10\USBECD.SYS /D:10CF:5500:0000 /N:$ /S /V
```

Obtained from USB Device Monitor for **K8055**

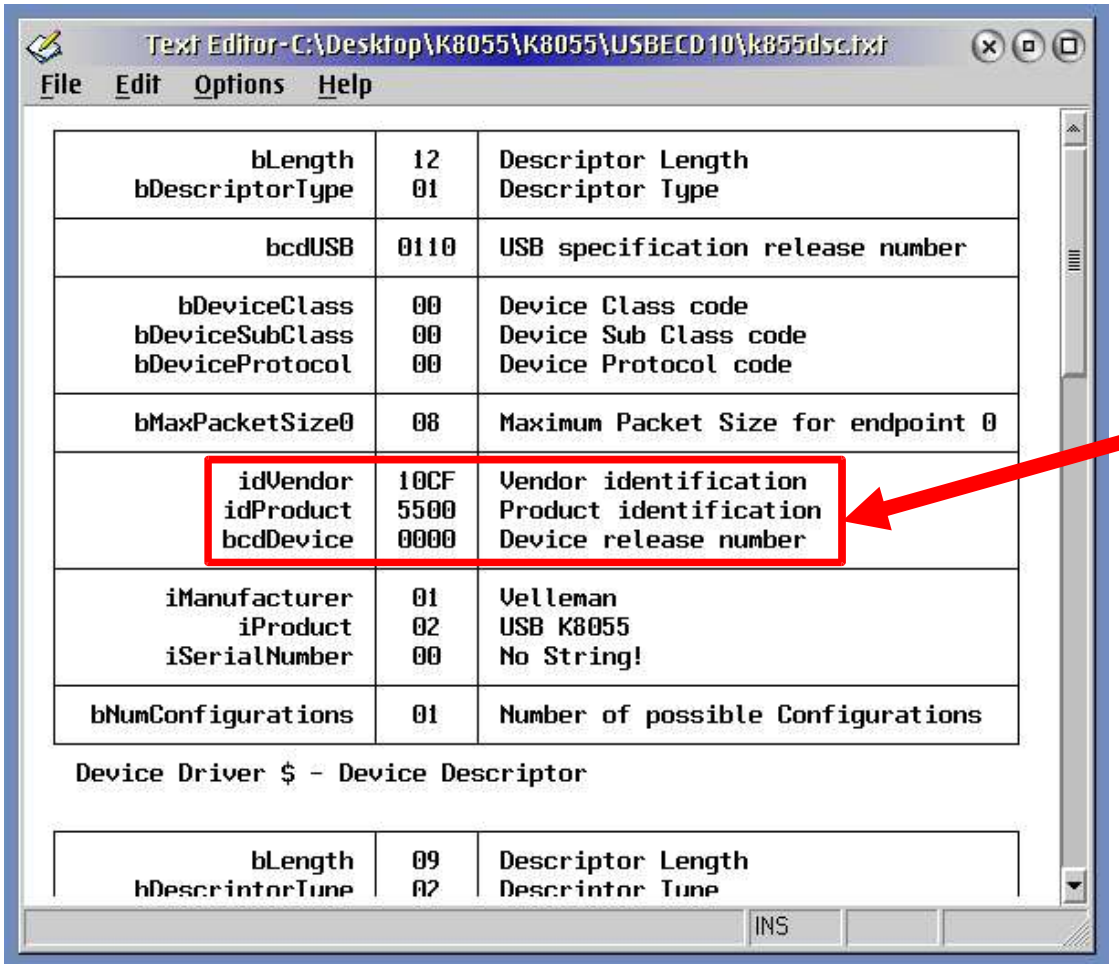
Minimum device name in all Wim Brul Examples

The USB I/O-Board VELLEMAN K8055 working with eCS

by Uwe Hinz, 2009

Wim Brul's example - output of the REXX script 'usbwrite.cmd'

1. How does the operating system (eCS) know K8055 is there ?



After Reboot an instance of Wim Brul's driver is loaded. It can be checked with the '**usbwrite.cmd**' example. The screenshot shows the same three numbers as they were used for the D parameter.

Caution ! The name '**usbwrite.cmd**' is **misleading** ! Only general setup data are written to the USB device !

by Uwe Hinz, 2009

The next level

2. How does K8055 start working ?

After the Wim Brul examples did it, the next question is:

- What shall be sent to the USB device to make it work ?

USB devices can have one or more configurations. A very common example is a digital photo camera. One configuration is its appearence as a Mass Storage Device while transfering data to a PC. An alternative or second configuration is its ability to react as TWAIN compatible hardware.

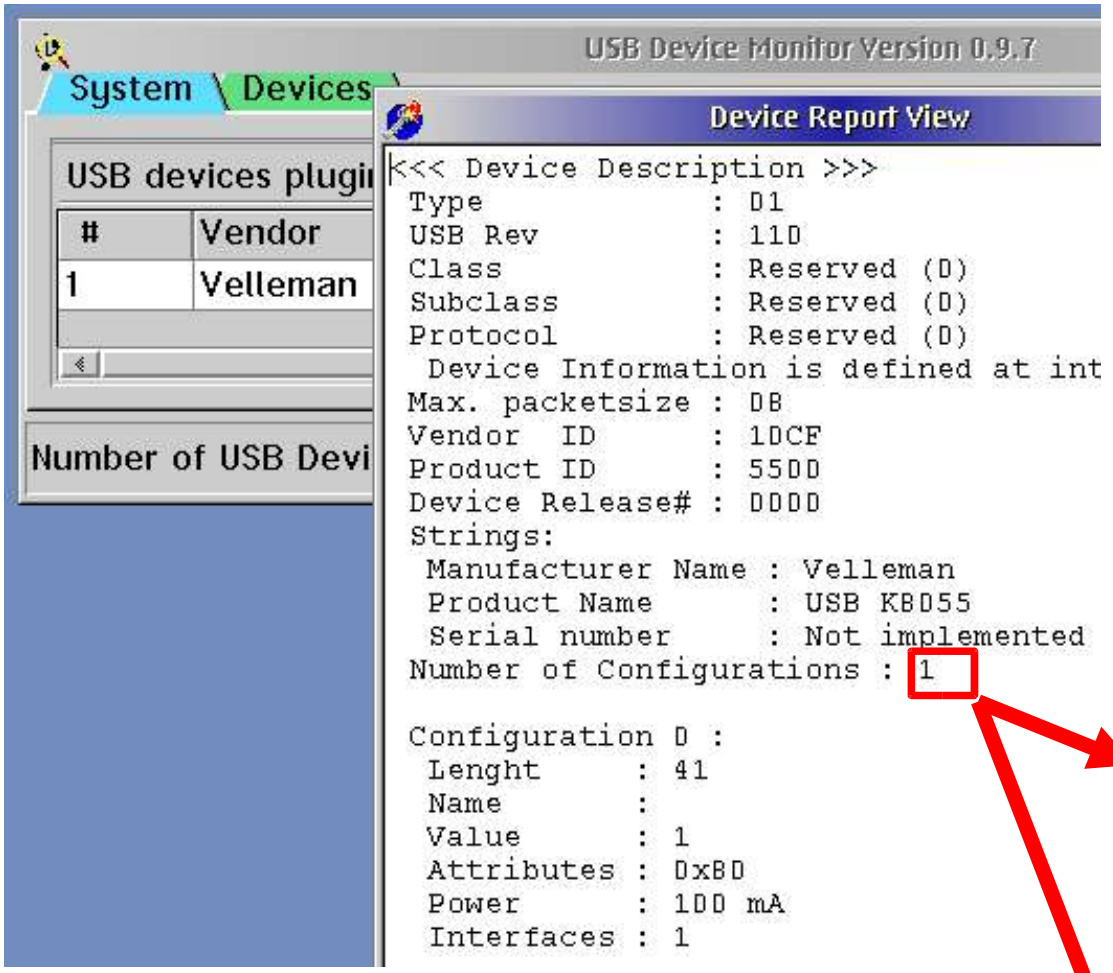
1. One must know **how many configurations** a K8055 has got !
2. Even if K8055 has got only one configuration, is it active by default ?
3. How can this configuration be activated ?
4. The examples, provided by **Wim Brul,** do not set anything concerning device properties.
   Being generic, **the driver 'usbecd.sys' knows nothing** in particular **about** K8055 and **any** other **device** !
5. What does the Setup Packet look like, that can activate configurations ? [3]
6. Setting a configuration is the same as transfering a Setup Packet to the particular USB device !
   How a Setup Packet is sent, can be found in 'usbwrite.cmd' lines 31, 32 and 33.

The USB I/O-Board VELLEMAN K8055 working with eCS

by Uwe Hinz, 2009

How many configurations K8055 has got ?

2. How does K8055 start working ?



The tool
'USB Device Monitor'

can tell the developer how many configurations **K8055** has got.

Answer: Only one !

The Setup Packet 'Set configuration 1' looks like **00 09 01 00 00 00 00 00** [3]

by Uwe Hinz, 2009

The next level

3. What kind of information is needed to write to K8055 and what is read from it ?

After the Wim Brul examples run, and the only configuration is set active, the next question is:

- How do I know what must be sent to the USB device in order to use its functionality ?

- What will the USB device send me back ?

If the manufacturer does not publish any appropriate specification, an OS different from eCS and a test application have to be used in order to observe the data exchange with

K8055 !

Moreover, what tool can help to observe the data exchange ?

by Uwe Hinz, 2009

The next level

3. What kind of information is needed to write to K8055 and what is read from it ?

After the Wim Brul examples run, and the only configuration is set active, the next question is:

- How do I know what must be sent to the USB device in order to use its functionality ?

- What will the USB device send me back ?

If the manufacturer does not publish any appropriate specification, an OS different from eCS and a test application have to be used in order to observe the data exchange with

K8055 !

Moreover, what tool can help to observe the data exchange ?

Hardware: Ellisys

Software: usbsniffer

by Uwe Hinz, 2009

The USB analyser hardware Ellisys 110

3. What kind of information is needed to write to K8055 and what is read from it ? Hardware: Ellisys

An USB analyser had to be ordered !

Market investigation
lead to the

Ellisys 110 .

Including the analyser
programme
Ellisys Visual USB

for Win2000, it promised
professional support.

The USB I/O-Board VELLEMAN K8055 working with eCS

by Uwe Hinz, 2009

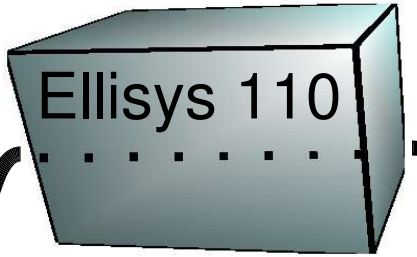
Watching the data exchange Ellisys 110

3. What kind of information is needed to write to K8055 and what is read from it ?

No additional wires or settings are needed.
K8055 data exchange will be investigated with Ellisys 110

Hardware: Ellisys

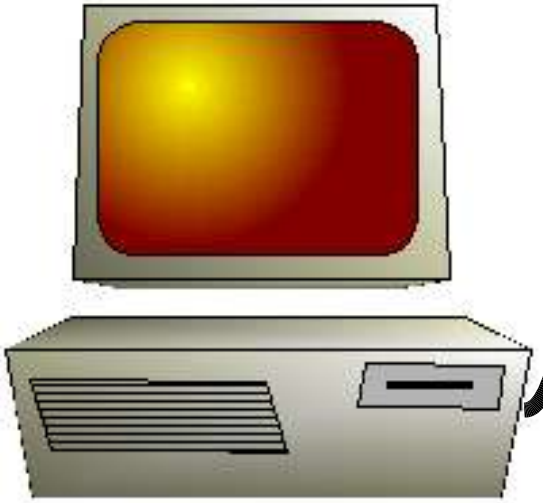
[2]



USB cable

PC (any OS)

development takes place here



USB cable

USB cable

The analyser application

Ellisys Visual USB

(Win2000, XP)

Warpstock Europe 10th Anniversary 2009
Nov. 13-15, Stralsund, Germany

The USB I/O-Board VELLEMAN K8055 working with eCS

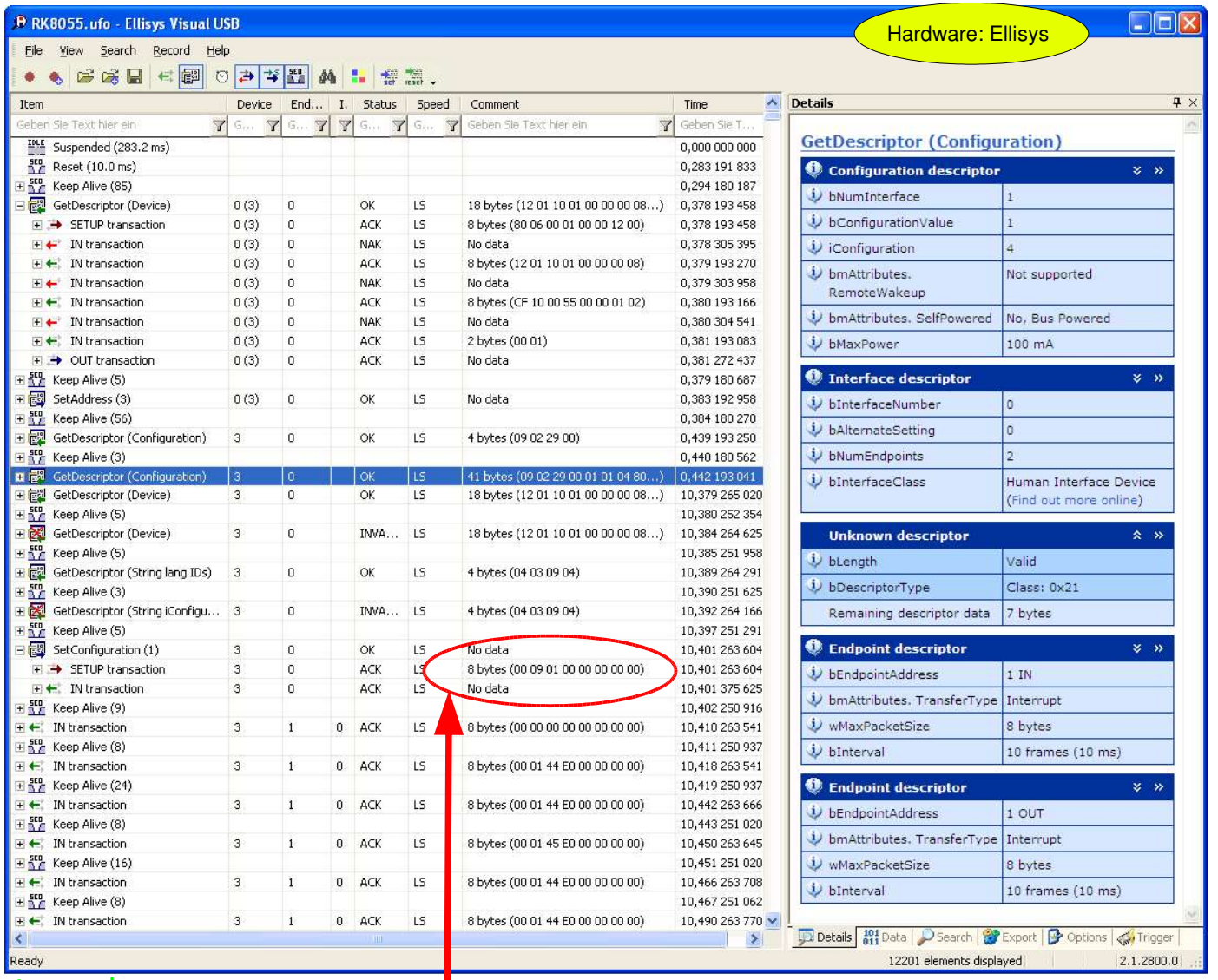
by Uwe Hinz, 2009

K8055 data exchange log made by the analyser hardware Ellisys 110

3. What kind of information is needed to write to K8055 and what is read from it ?

1. Joining the Bus

2. Initialising K8055



3. Data exchange

Configuration 1 set active

For details see [3].

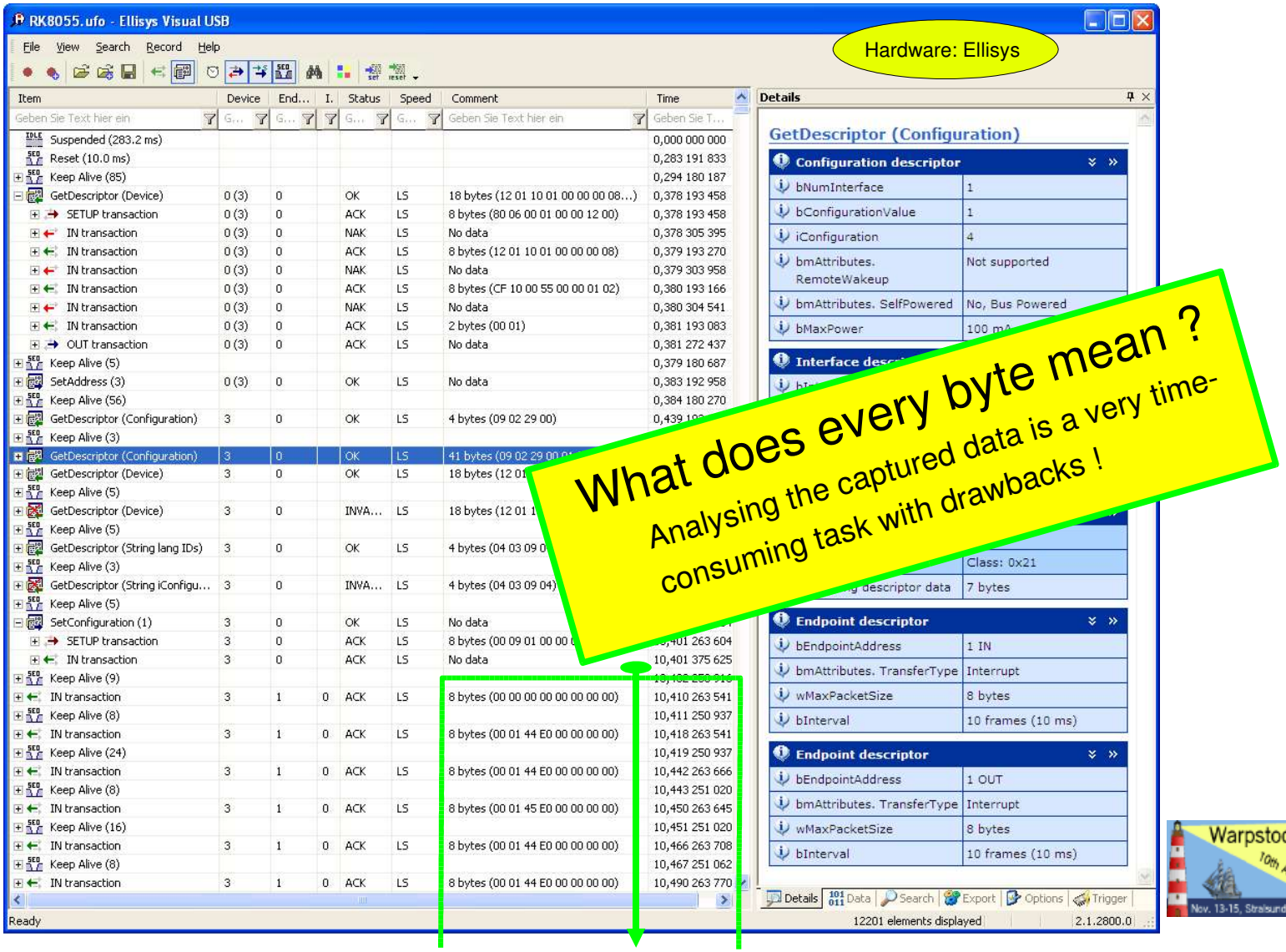
The USB I/O-Board VELLEMAN K8055 working with eCS

by Uwe Hinz, 2009

Thorough investigation must reveal the K8055 data traffic

3. What kind of information is needed to write to K8055 and what is read from it ?

1. Joining the Bus
2. Initialising K8055
3. Data exchange



The USB I/O-Board VELLEMAN K8055 working with eCS

by Uwe Hinz, 2009

Thorough investigation must reveal the K8055 data traffic

3. What kind of information is needed to write to K8055 and what is read from it ?

'**usbsniffer**' is a software tool for Windows (Win98, Win2000), that capture all USB Traffic

Text Editor-D:\Desktop\K8055\K8055\usbecd00\K8055-01.LOG

File Edit Options Help

Software: usbsniffer

```
00126779   21:22:00       Status = 00000000
00126780   21:22:00     -- URB_FUNCTION_BULK_OR_INTERRUPT_TRANSFER:
00126781   21:22:00       PipeHandle            = 0xC19CC518
00126782   21:22:00       TransferFlags         = 00000003 (USBD_TRANSFER_DIRECTION_IN, USBD_SHORT_TRANSFER_OK)
00126783   21:22:00       TransferBufferLength  = 00000008
00126784   21:22:00       TransferBuffer        = c19ef310
00126785   21:22:00       TransferBufferMDL     = c19efba0
00126786   21:22:00
00126787   21:22:00         0000:
00126788   21:22:00      71 01 ba 8e 00 00 00 00
00126789   21:22:00       UrbLink               = 00000000
00126790   21:22:00     UsbSnoop - DispatchAny : IRP_MJ_INTERNAL_DE
00126791   21:22:00     UsbSnoop - MyDispatchInternalIOCTL(FF02BD87
00126792   21:22:00      >>>  URB 5282 going down  >>>
00126793   21:22:00       Status = 00000000
00126794   21:22:00     -- URB_FUNCTION_BULK_OR_INTERRUPT_TRANSFER:
00126795   21:22:00       PipeHandle            = 0xC19CC518
00126796   21:22:00       TransferFlags         = 00000003 (USBD_TRA
00126797   21:22:00       TransferBufferLength  = 00000008
00126798   21:22:00       TransferBuffer        = c19ef310
00126799   21:22:00       TransferBufferMDL     = 00000000
00126800   21:22:00       UrbLink               = 00000000
00126801   21:22:00     UsbSnoop - DispatchAny : IRP_MJ_INTERNAL_DEVICE_CONTROL
00126802   21:22:00     UsbSnoop - MyDispatchInternalIOCTL(FF02BD87) : fdo=C18A38B8, Irp=C19E3360
00126803   21:22:00      >>>  URB 5283 going down  >>>
00126804   21:22:00       Status = 00000000
00126805   21:22:00     -- URB_FUNCTION_BULK_OR_INTERRUPT_TRANSFER:
00126806   21:22:00       PipeHandle            = 0xC19CC52C
00126807   21:22:00       TransferFlags         = 00000002 (USBD_TRANSFER_DIRECTION_OUT, USBD_SHORT_TRANSFER_OK)
00126808   21:22:00       TransferBufferLength  = 00000008
00126809   21:22:00       TransferBuffer        = e6a18f49
00126810   21:22:00       TransferBufferMDL     = 00000000
00126811   21:22:00
00126812   21:22:00         0000:
00126813   21:22:00      05 1f 00 00 00 00 00 00
00126814   21:22:00       UrbLink               = 00000000
00126815   21:22:00     UsbSnoop - MyInternalIOCTLCompletion(FF02BC08) : fido=C1A05778, Irp=C19C0C40, Context=C1CB5B30
00126816   21:22:00      <<<  URB 5281 coming back  <<<
```

Found INS

What does every byte mean ?

Going through the Log File is the analysing job.
One must gess, assume, compare or predict over a big number of cycles.

The USB I/O-Board VELLEMAN K8055 working with eCS

by Uwe Hinz, 2009

Ellisys or usbsniffer ?

3. What kind of information is needed to write to K8055 and what is read from it ?

	Hardware: Ellisys	Software: usbsniffer
- Price	800 EUR	free (downloaded from the internet)
- Output	Binary (Exportfilter txt, xls.. - extra costs)	txt format
- OS	Win2000, WinXP	Win98, Win2000, WinXP
- Equipment	Little Box (50 x 80 x 80 mm), cables	no extra hardware, cables
- File size / minute	appx. 250KByte per minute (K8055-Demo conditions (Win98))	appx. 5MByte per minute (K8055-Demo conditions (Win98))
- Readability	very good	poor
- Assistance	excellent ! Help with citations of USB 1.1 Specifications	nothing alike

by Uwe Hinz, 2009

A walk through the desert

3. What kind of information is needed to write to K8055 and what is read from it ?

by Uwe Hinz, 2009

Discovered K8055 commands and answers

3. What kind of information is needed to write to K8055 and what is read from it ?

Writing – allways 8 bytes going to K8055

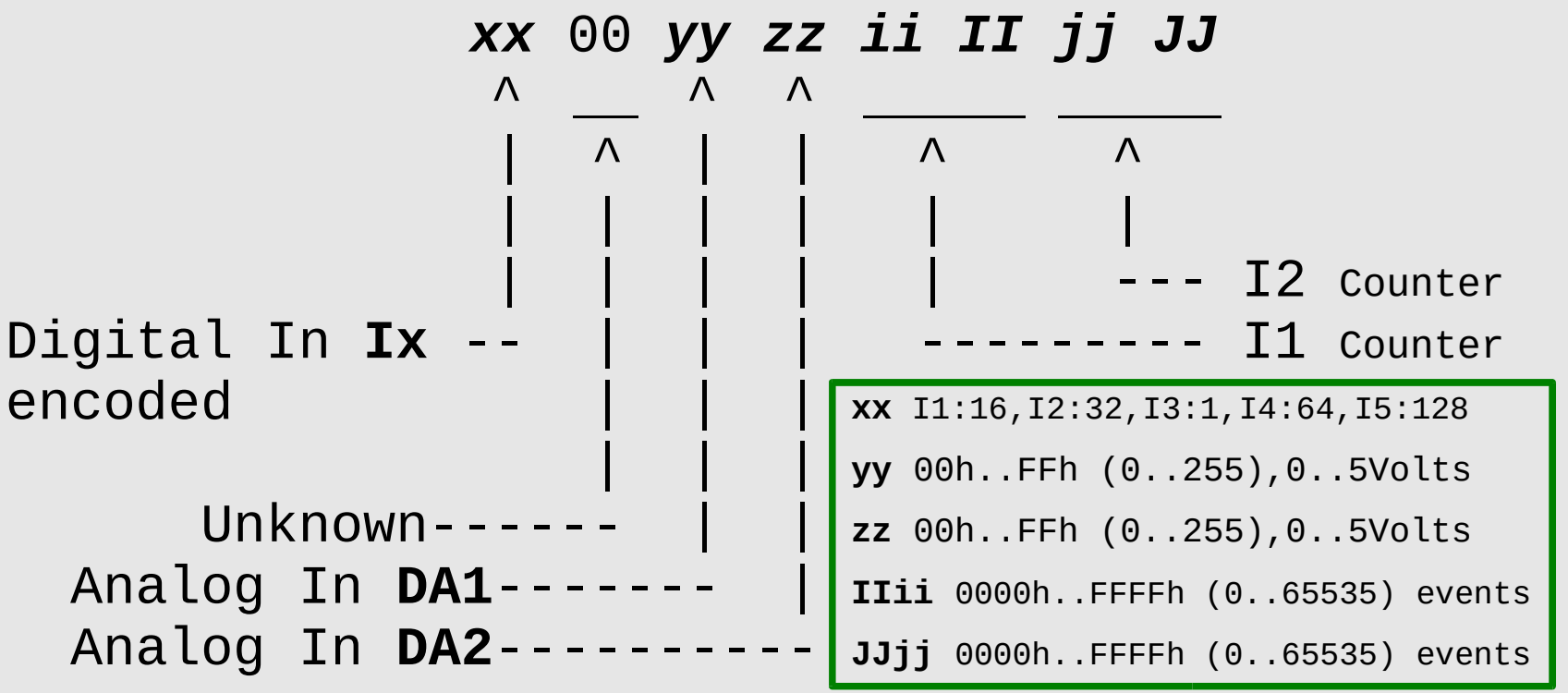
```
                  05 xx yy zz 00 00 00 00
                  __  ^  ^  ^  ___________
                   ^  |  |  |        ^
                   |  |  |  |        |
                   |  |  |  |        |
                   |  |  |  |        |
                   |  |  |  |        --- Unknown
Always 05, -----   |  |  |
Reason unknown     |  |  |
                   |  |  |
  Digital Out Ox---   |  |  | xx 00h..FFh (0..255)
  Analog Out AD1------   |  | yy 00h..FFh (0..255),0..5Volts
  Analog Out AD2---------   | zz 00h..FFh (0..255),0..5Volts
```

by Uwe Hinz, 2009

Discovered K8055 commands and answers

3. What kind of information is needed to write to K8055 and what is read from it ?

Reading – allways 8 bytes coming from K8055



The USB I/O-Board VELLEMAN K8055 working with eCS

by Uwe Hinz, 2009

The difference between Setup Packets and Parameter Packets

Setup Packets can be observed on the Universal Serial Bus. They travel from the OS to the USB Device.

```
00 09 01 00 00 00 00 00
```

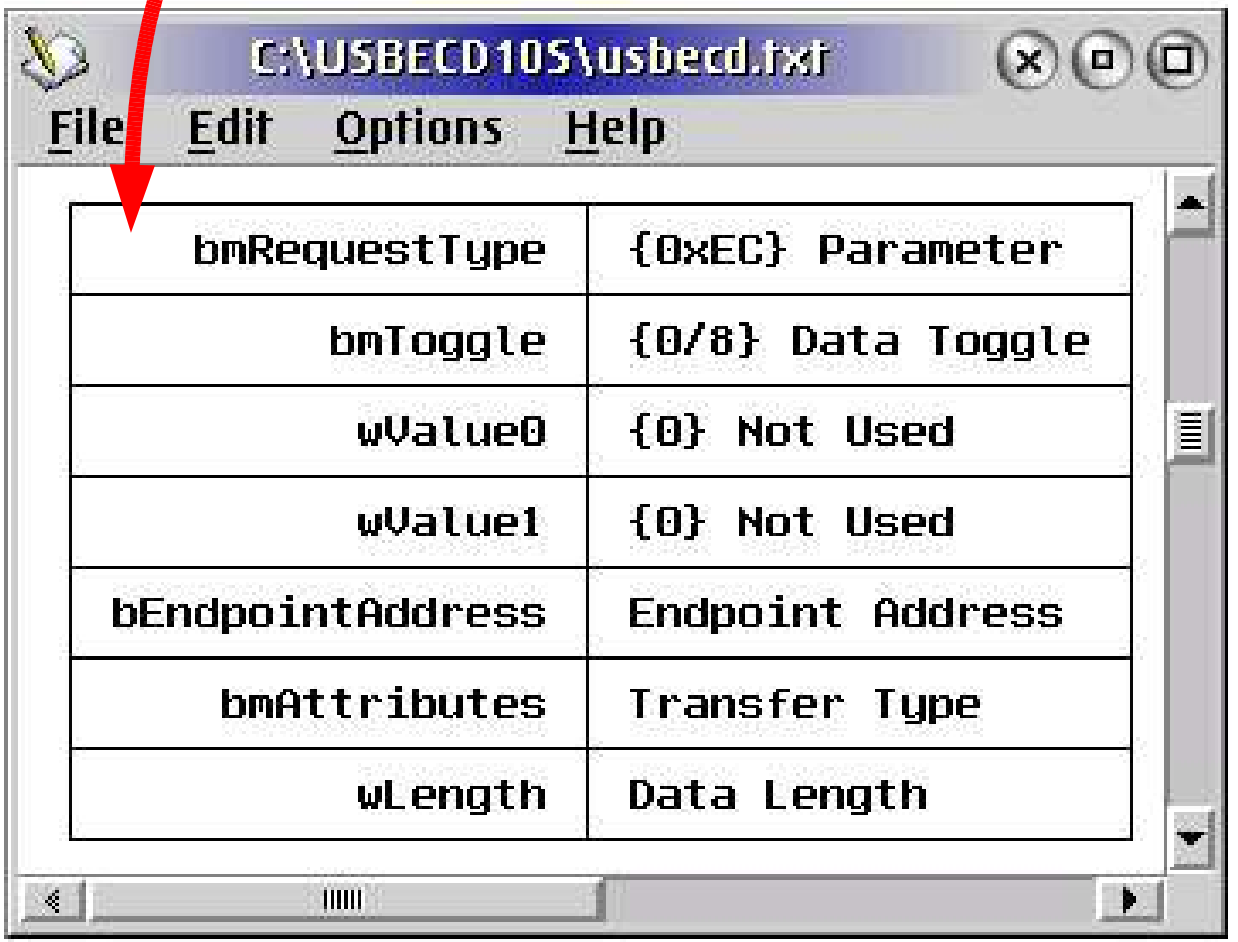
for setup (SetConfiguration 1)

Parameter Packets are to carry the data. They do not appear on the bus. Just the data they carry do so.

```
EC 0x 00 00 xx xx xx xx
```

for data exchange

The USB driver wants **0xEC** in the request type byte to do data exchange instead of setup.



by Uwe Hinz, 2009

Basic form of Parameter Packets and how to complete it

- Basic type of Parameter Packets :

For sending or getting data

EC 0x 00 00 xx xx xx xx

xx is the unknown part.
It must be extracted from the
the Configuration Descriptor.
See next pages....

by Uwe Hinz, 2009

Two types of Parameter Packets are needed


```
- Two types of Parameter Packets :


 Sending data to the device
           EC 0x 00 00 01 03 xx xx

 Getting data from the device
           EC 0x 00 00 81 03 xx xx
```

Endpoint addresses (0x81, 0x01) and transfer type (0x03) are completed.
The remaining xx indicates the length of the trailing data. The maximum length is passed in the Configuration Descriptor too.
See next pages....

by Uwe Hinz, 2009

Parameter Packets ready for K8055

- Parameter packets with data buffer lengths inserted:

(data buffer not shown)

Sending data to the device (K8055)

EC 0x 00 00 01 03 ***08 00***

Getting data from the device (K8055)

EC 0x 00 00 81 03 ***08 00***

Parameter packets in usable form. Trailing data are not mentioned here.
See next two pages that show complete parameter packets for sending and reading...

The USB I/O-Board VELLEMAN K8055 working with eCS

by Uwe Hinz, 2009

How to put Parameter Packet and K8055 data together (writing)

Parameter Packet with data to be written

```
EC 18 00 00 01 03 08 00  05 55 80 0E 00 00 00 00  before call
EC 10 00 00 01 03 08 00  05 55 80 0E 00 00 00 00  after call
EC 18 ..                    ^  ^  ^
EC 10 ..                    |  |  |
                            |  |  |
                            |  |  |
Always 05 -----             |  |  |
Reason unknown              |  |  |
                            |  |  |
 Digital Out Ox---          |  |  |
 Analog Out AD1------       |  |  |
 Analog Out AD2----------   |  |  |
```

Ox 00h..FFh (0..255)
AD1 00h..FFh (0..255),0..5Volts
AD2 00h..FFh (0..255),0..5Volts

by Uwe Hinz, 2009

How to put Parameter Packet and K8055 data together (reading)

Parameter Packet with data to be read

```
EC 18 00 00 81 03 08 00  2E 2E 2E 2E 2E 2E 2E 2E  before call
EC 10 00 00 81 03 08 00  40 01 F0 0F 01 00 02 00  after call
EC 18 ..                  ^     ^  ^  ^^^^^^^^^^
EC 10 ..                  |  ^  |  |  |  |  |
                          |  |  |  |  |  |  |
                          |  |  |  |  |  |  --- I2 Counter
                          |  |  |  |  |  ------ I1 Counter
Digital In Ix --          |  |  |  |
                          |  |  |  |
     Unknown------        |  |  |  |
  Analog In DA1-------    |  |  |  |
  Analog In DA2---------- |  |  |  |
```

Ix encoded

DA1 00h..FFh (0..255),0..5Volts

DA2 00h..FFh (0..255),0..5Volts

by Uwe Hinz, 2009

Parameter Packet and data put together for sending


```
/*-- rexx fragment ------------------------------------------------------*/
/*   DataFromApplicationToDeviceK8055
sTrailingData = X2C(05 55 80 0E 00 00 00 00 )   /* 8 bytes */
oiBuffer = X2C(EC 08 00 00 01 03 08 00) || sTrailingData
```

8 bytes prepaired to be written

EC 08 00 00 01 03 08 00 **05 55 80 0E 00 00 00 00**

Variable: oiBuffer

```
CALL WriteAndReadProcedure
```

DataToggleBit before CALL

...

EC 00 00 00 01 03 08 00 **05 55 80 0E 00 00 00 00**

DataToggleBit after CALL

```
/* ------------------------------------------------------*/
/*                                                       */
   WriteAndReadProcedure:

     RC = CHAROUT(ddName, oiBuffer)
          ...
     RETURN
```

The 8 last bytes in **oiBuffer** have to be treated as a block.

05 – Data Output via Digital and Analog Ports, correct meaning unknown !

55 – O1 **LOW**, O2 **HIGH**, O3 **LOW**, O4 **HIGH** O5 **LOW**, O6 **HIGH**, O7 **LOW**, O8 **HIGH**

80 - Analog Ouput **DA1** at **128 * 0.01960** -> 2.59 Volts

0E - Analog Output **DA2** at **14 * 0.01960** -> 0.2740 Volts

00 00 00 00 – unknown !

by Uwe Hinz, 2009

. . . and for reading the answer


```
/*-- rexx fragment ------------------------------------------------------------*/
/*   DataFromDeviceK8055ToApplication
sTrailingData = '........'    /* 8 bytes empty, – phantasy characters */
oiBuffer = X2C(EC 00 00 00 81 03 08 00) || sTrailingData
                    EC 00 00 00 81 03 08 00  2E 2E 2E 2E 2E 2E 2E 2E
CALL WriteAndReadProcedure
                    EC 08 00 00 81 03 08 00  40 01 F0 0F 01 00 02 00
...
```

8 bytes waiting to be read

Variable: oiBuffer

DataToggleBit before CALL

DataToggleBit after CALL

```
/* ---------------------------------------------------*/
/*                                                    */
   WriteAndReadProcedure:

      RC = CHAROUT(ddName, oiBuffer)
           ...
      RETURN
```

The 8 last bytes in **oiBuffer** have to be extracted and used separately.

- 40 - Digital Input **I4** is **LOW**
- 01 - unknown
- F0 - Analog Input **AD1** at **240 * 0.01960** -> 4.704 Volts
- 0F - Analog Input **AD2** at **15 * 0.01960** -> 0.294 Volts
- 0001 – One pulse counted on Digital Input **I1**
- 0002 – Two pulses counted on Digital Input **I2**

The USB I/O-Board VELLEMAN K8055 working with eCS

by Uwe Hinz, 2009

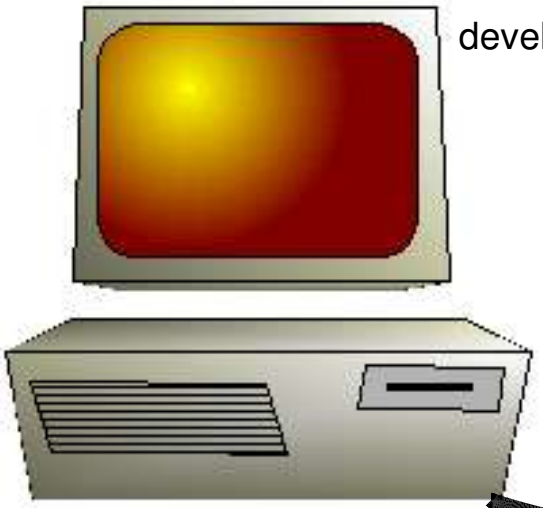
Further Developing with eCS after the data exchange is known

After data exchange between the OS and eCS is analysed and the meanings of messages are clear, development can go ahead with less equipment.

K8055

PC (eCS)

development takes place here



USB port

USB cable

The USB I/O-Board VELLEMAN K8055 working with eCS

by Uwe Hinz, 2009

From Phase 1 to Phase 2

Phase 1

- Collecting knowlege about K8055 using Wim Brul's driver

- Testing the knowlege with a number of small REXX scripts

Phase 2

- Creating a Test Application with C using USBCALLS [7] ?

- Creating a Test Application from the small REXX scripts ?

- Creating a new K8055 driver ? → Phase 6..7 ?

The USB I/O-Board VELLEMAN K8055 working with eCS

by Uwe Hinz, 2009

Wim Brul's 'usbwrite.cmd' changed to work with K8055

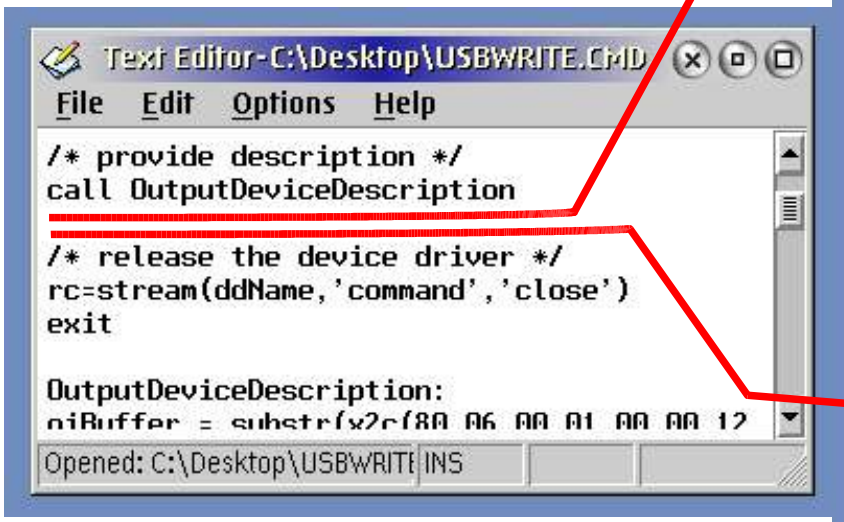
With 20 extra lines of source code **usbwrite.cmd** can be turned into a little programm that will work with **K8055** for output operations:

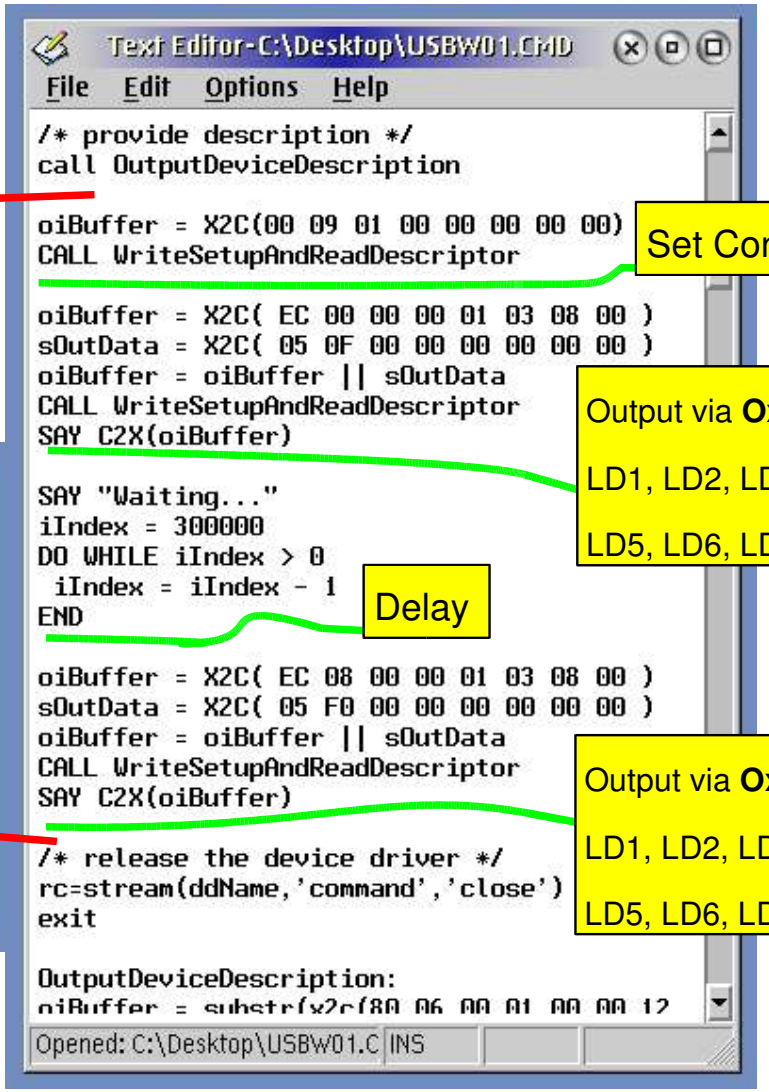
usbw01.cmd

Line 1
↓
Line 24
Line 25

Line 27
Line 28
Line 29

Line 31
Line 32
↓
Line 225





Set Configuration 1

Output via **Ox**:

LD1, LD2, LD3, LD4 = ON

LD5, LD6, LD7, LD8 = OFF

Delay

Output via **Ox**:

LD1, LD2, LD3, LD4 = OFF

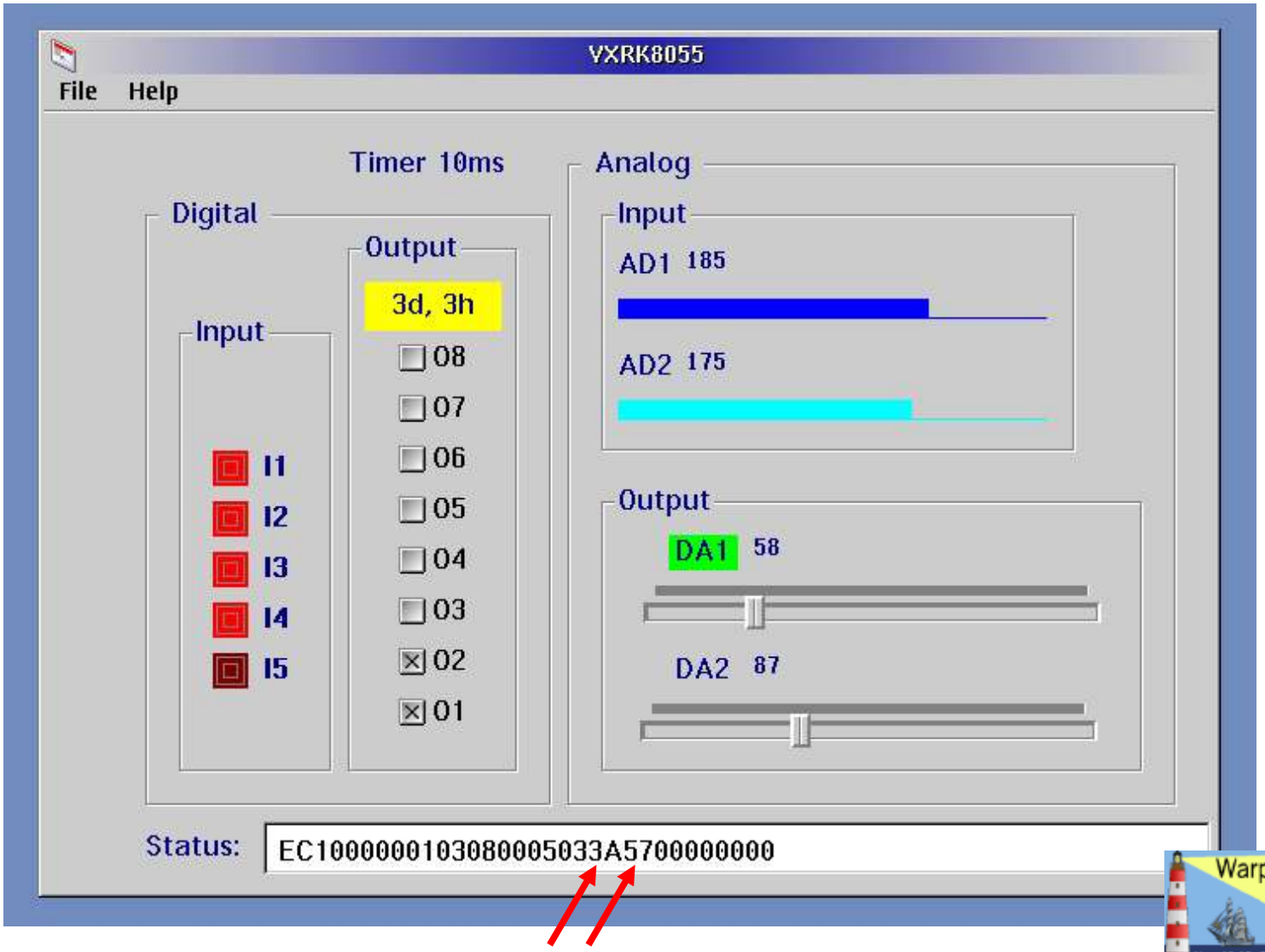
LD5, LD6, LD7, LD8 = ON

A recipe

The USB I/O-Board VELLEMAN K8055 working with eCS

by Uwe Hinz, 2009

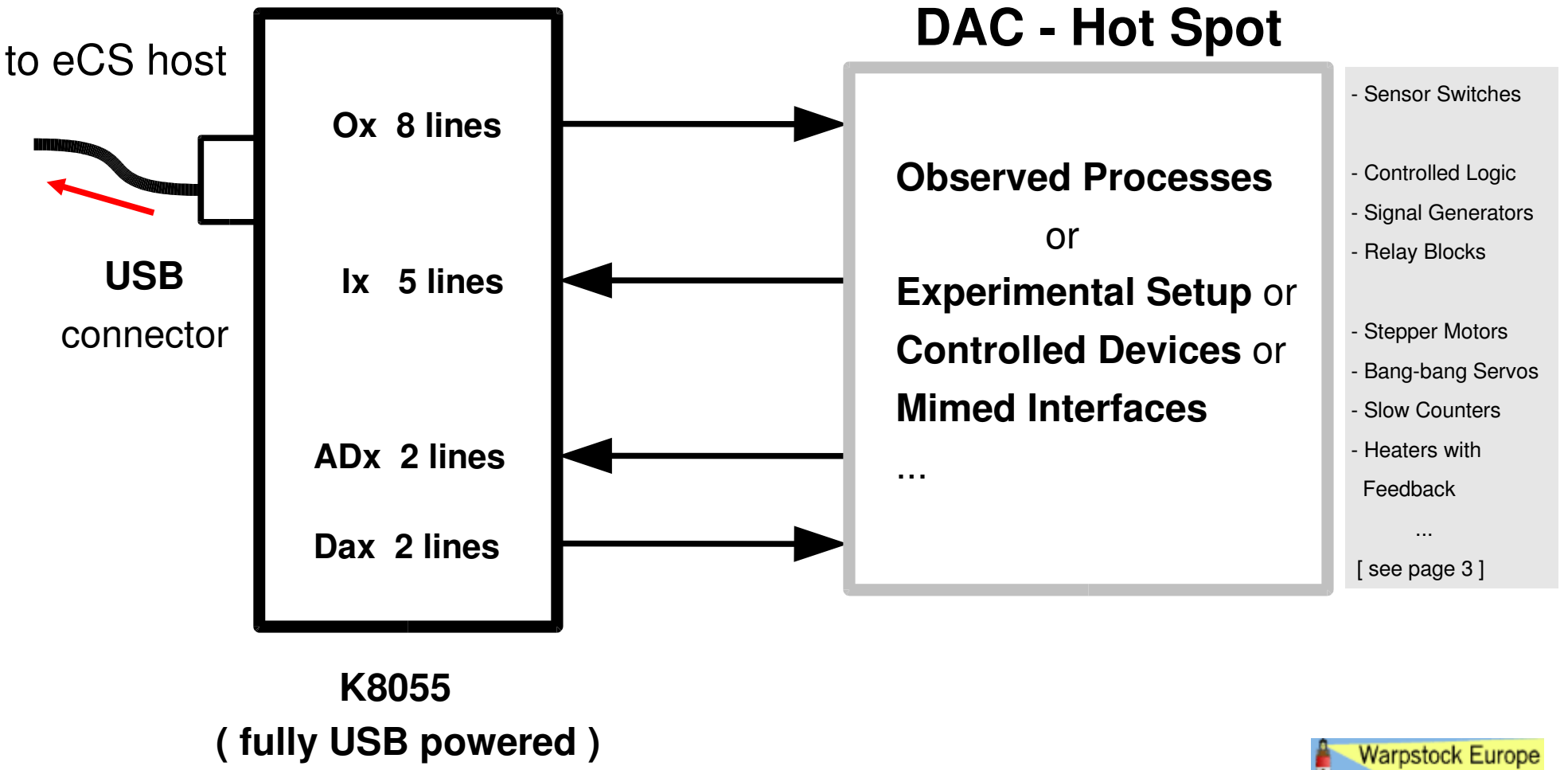
K8055 GUI written in VX-REXX



The USB I/O-Board VELLEMAN K8055 working with eCS

by Uwe Hinz, 2009

Hypothetic use of K8055



by Uwe Hinz, 2009

From Phase 2 to Phase 3

Phase 2

- Test Applications with REXX or VX-REXX created !

Phase 3

Turning the collected knowledge into a Device Control Application

' K8055.cmd '

to interface the Digital Thermometer

DTM 2010

by Uwe Hinz, 2009

DTM 2010 - The Device, that can be connected to K8055

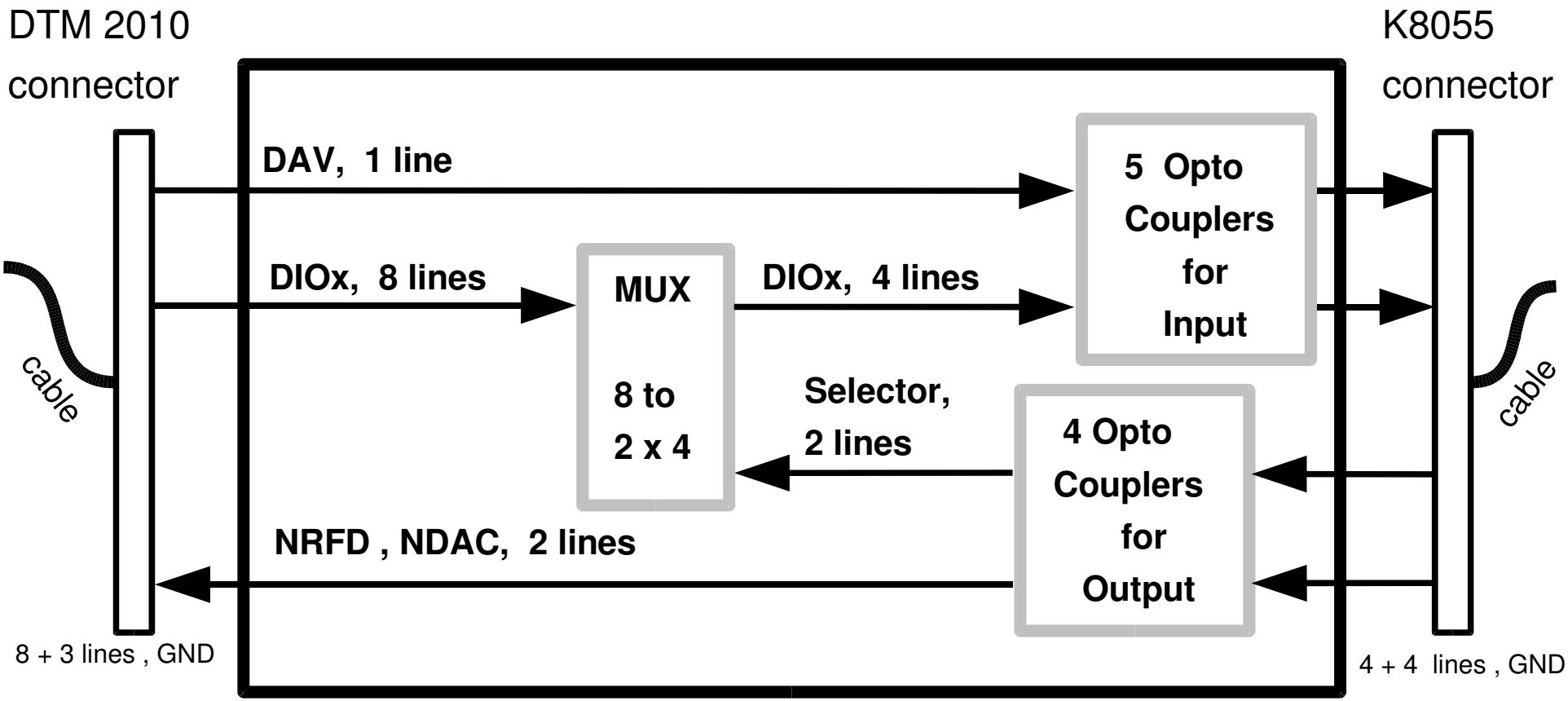
This historic digital thermometer was manufactured before 1989 by ' VEB Thermometerwerk Geraberg '.

It is equipped with a reduced **IEEE 488 Interface** (8 Data Lines + 3 Handshake Lines, no Control Lines) [6] [8].

A bit of extra electronics is needed in order to attach the device to **K8055**.

by Uwe Hinz, 2009

The extra hardware for connecting the DTM 2010 to K8055

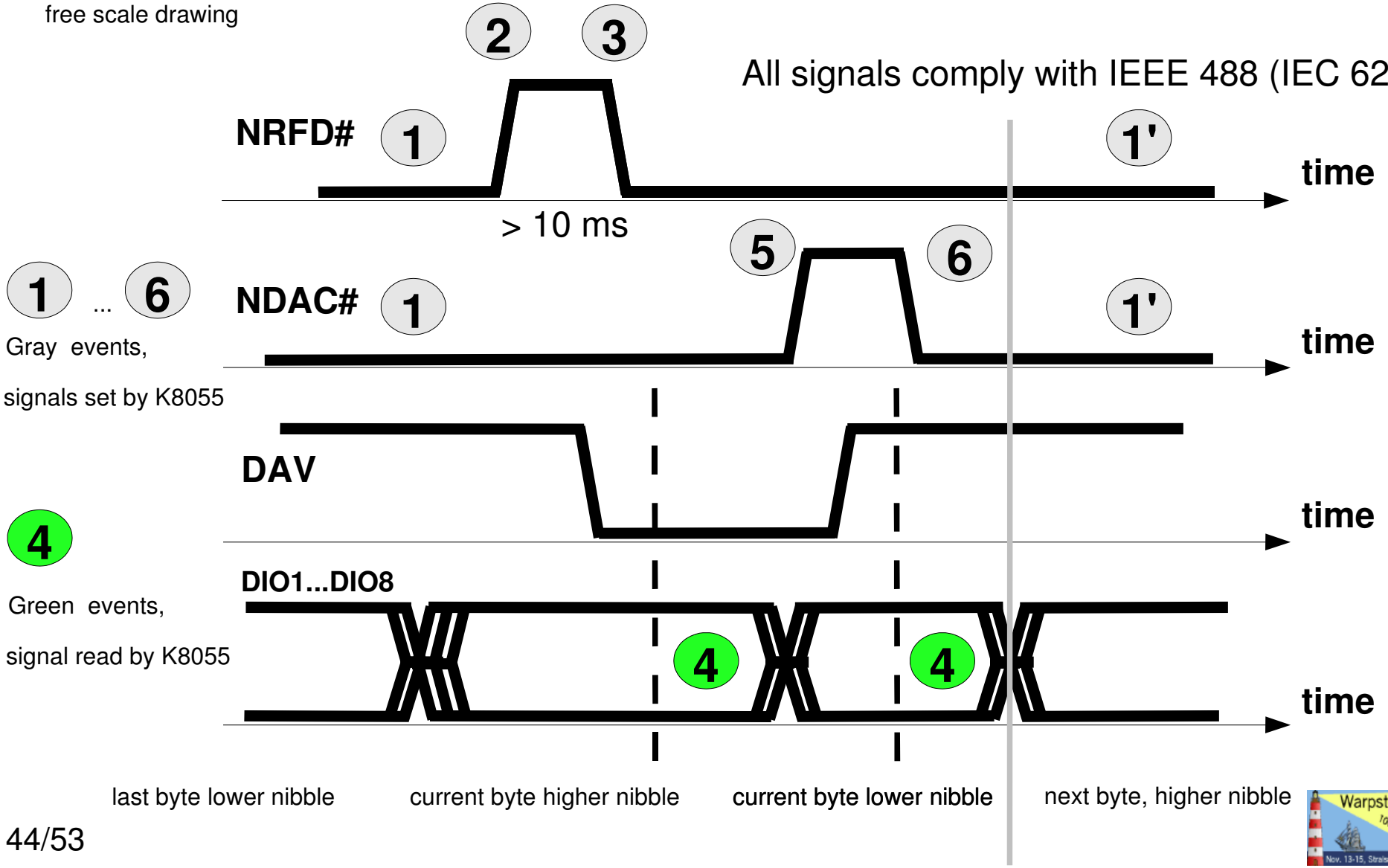


by Uwe Hinz, 2009

DTM 2010 signals and interaction with K8055

free scale drawing

All signals comply with IEEE 488 (IEC 625) [6]



The USB I/O-Board VELLEMAN K8055 working with eCS

by Uwe Hinz, 2009

An Egg Boiler Example

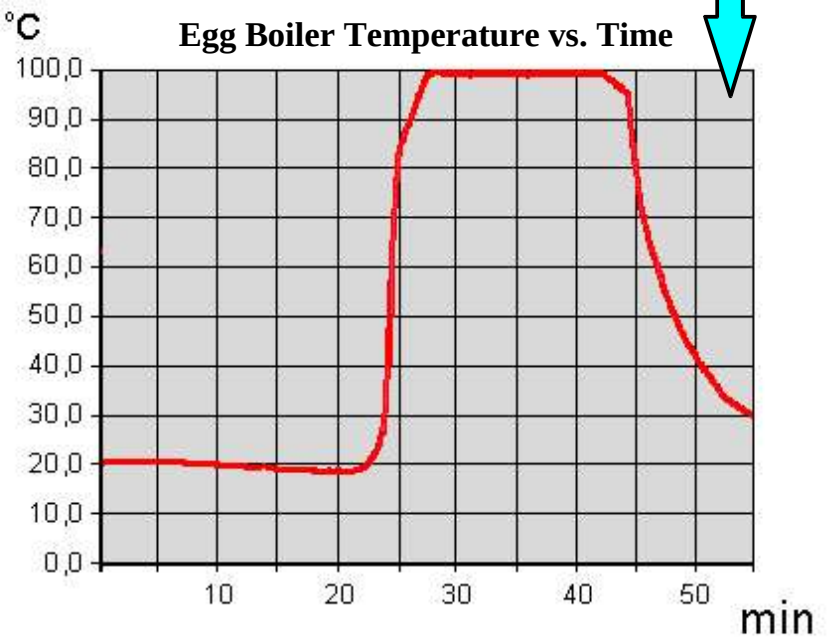
Thermometer ' DTM 2010 ' and

REXX script ' K8055.cmd '

Warpstock Europe 2009 10th Anniversary Nov. 13-15, Stralsund, Germany

Data file ' K8055.csv '

OpenOffice Calc file ' EIK01.scx '



by Uwe Hinz, 2009

From Phase 3 to Phase 4

Phase 3

- Trying to get rid of VX-REXX.
  It is very buggy and not maintained any more.

A way to create PM-Applications with GUI is needed !

Phase 4

Project Team **Björn Hennig** and **Uwe Hinz**

Turning the collected knowledge into a

DLL

for K8055

with **Open Watcom C**

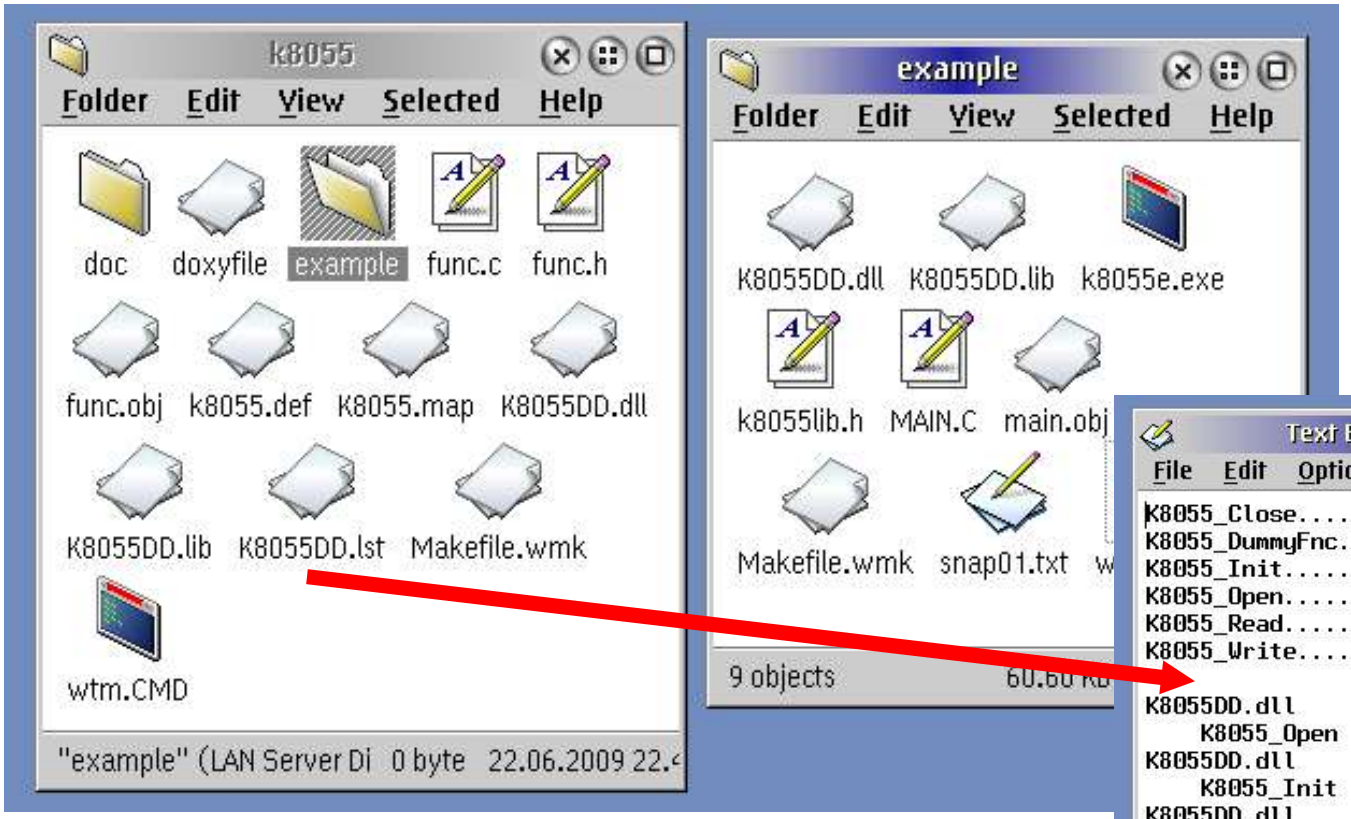
1. Step

The USB I/O-Board VELLEMAN K8055 working with eCS

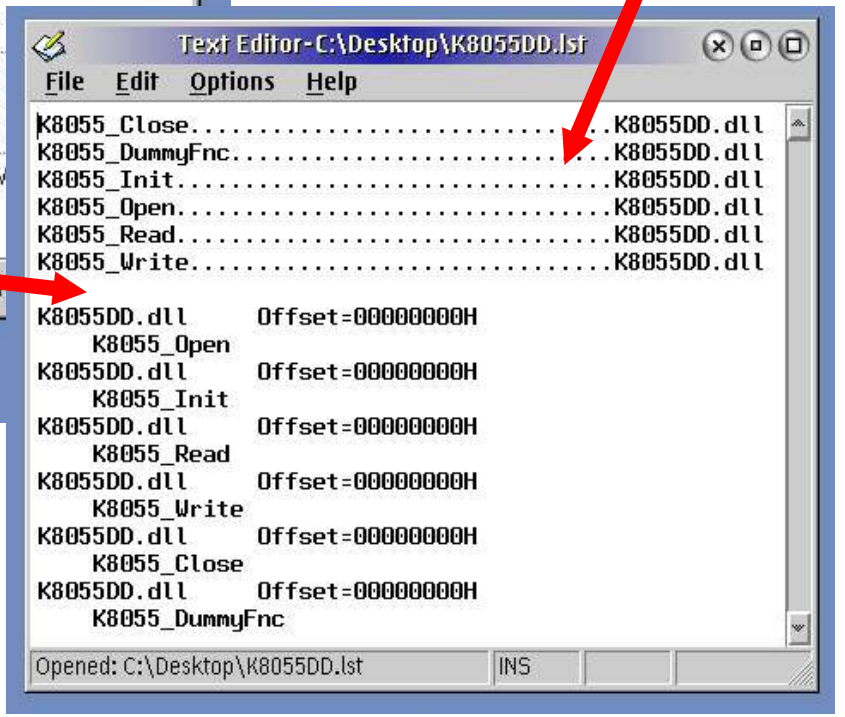
by Uwe Hinz, 2009

Creating a DLL with Watcom C

Done by Björn Hennig



6 functions exported by **K8055DD.dll**



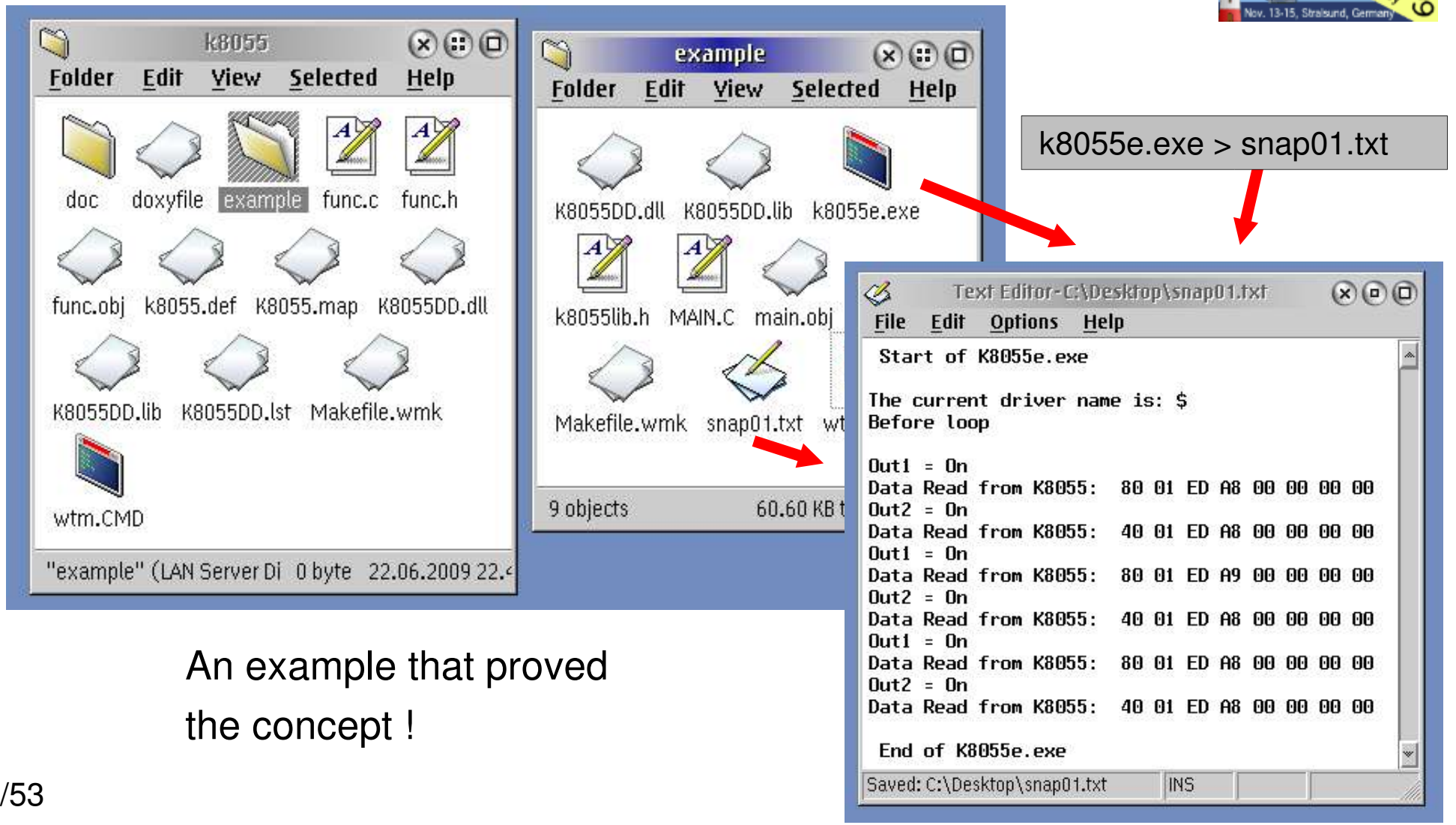
Just a glimpse of the programmer's workshop

The USB I/O-Board VELLEMAN K8055 working with eCS

by Uwe Hinz, 2009

Building an example with Watcom C

Done by Björn Hennig



An example that proved the concept !

by Uwe Hinz, 2009

The old To Do List

from: **„Digital I/O with eComStation via the USB-Interface meM-PIO“**

- Speed measurement and extended tests, YES, done
- Using a language different from REXX, YES, done
- Extending error checks within the applications, No, not done
- Doing the same project with USBCALLS, No, not done, project continued Wim Brul's driver
- Tests with more than one meM-PIO, No , not on the market any more, working with K8055 instead
- Adding a Serial Number check to bind an individual meM-PIO to a future application, dito
- Getting a new meM-PIO and looking for the differences that occurred over time, No, see above
- Connecting meM-PIO with a device different from thermometer DTM 2010, No, not done
- Creating a project that is run by more than one person, YES, done (**Björn Hennig** and **Uwe Hinz**)
- Turning Wim Brul's driver into a special meM-PIO driver, No, not done, idea suspended
- and ...

by Uwe Hinz, 2009

To Do List

- Speed measurement and extended tests
- Going over from REXX to Python, **Qt4-GUI**, Watcom C...
- Tests with more than one K8055
- Adding a Serial Number check to bind an individual K8055 to a future application
- Extending error checks within the applications
- Connecting K8055 with a device, different from thermometer DTM 2010
- Continuing work in the project team

by Uwe Hinz, 2009

References 1

[1] http://home.hccnet.nl/w.m.brul/usbprobe/index.html

[2] http://www.velleman.eu/distributor/products/

[3] **Compaq, Intel, Microsoft, NEC** 'Universal Serial Bus Specification' Revision 1.1, September 23, 1998

[4] http://AutomationONSPEC.com/AOSbrochure.pdf

[5] http://automationonspec.com/

[6] http://www.techsoft.de/htbasic/tutgpibm.htm?tutgpib.htm/

by Uwe Hinz, 2009

References 2

[7] http://en.ecomstation.ru/projects/usbtools/download/usbcalls-20060807.zip , ftp://ftp.netlabs.org/pub/events/DWS2005/DWS2005_USBStack.pdf

[8] http://www.juengling-online.de/Data/DTM_2010.pdf

Acknowledgement

I thank

All members of the OS/2 User Group Dresden,
Björn Hennig in particular

and

Heidi Fritzlar, my wife, for her help and understanding...

Photos by:

Annemarie Koebel	(page 42)
Björn Hennig	(pages 2, 3, 4, 20, 36)
Jens Eickhoff	(page 2)
Thomas Schoch	(page 25)
Uwe Hinz	(pages 7, 9, 45)