

## UNAX<sup>®</sup> Ux4

### Instruction Manual

### Multi-axis motion coordinator for XENAX<sup>®</sup> servo controllers

Edition June 2008



### Compact multi-axis motion coordinator with TCP/IP web technology

Commissioning and parameterisation of  
servo axes by web browser.  
Axis coordinator free programmable in BASIC  
syntax for standalone applications.

## General

This instruction manual describes the multi-axis motion coordinator UNAX<sup>®</sup> Ux4 with XENAX<sup>®</sup> slave servo controller.

The document contains necessary information about commissioning, set up, electrical connections, programming and error handling.

The operating system of the multi-axis motion coordinator in addition to the axis user interface WebMotion<sup>®</sup> is installed, so the UNAX<sup>®</sup> system is instantly ready to use.

We will gladly answer your questions you may have or supply additional information.

Alois Jenny  
Jenny Science AG

## Contents

1 System overview	4
2 Electrical Connections	5
2.1 Pin configuration	6
2.2 System cabling	9
3 Development Environment / Tools	10
4 Programming XENAX <sup>®</sup> slave controller	11
4.1 Ethernet connect	11
4.2 Test IP with >IPCONFIG	11
4.3 Test connection with >PING	11
4.4 IP address search / change	12
4.5 JAVA Plugin check	13
4.6 Applet Cache	14
4.7 WebMotion <sup>®</sup>	15
4.8 Set up XENAX <sup>®</sup> for operation with UNAX <sup>®</sup>	16
5 Programming the UNAX <sup>®</sup> multi-axis coordinator	17
5.1 Motion Perfect 2	17
5.2 Axis configuration	17
5.3 BASIC command set	17
5.4 Tasks and motion profile generator	17
5.5 System module UNAX_SYS.bas	18
5.6 User I/O UNAX	19
5.7 XENAX <sup>®</sup> handshake with UNAX <sup>®</sup>	20
5.8 COM 1 / COM 2 Interface	23
5.9 External Encoder	24
6 Application example	25
6.1 Exercise	25
6.2 Parameterisation XENAX <sup>®</sup> slave servo controller	26
6.3 UNAX <sup>®</sup> application	28
6.4 Test the user application	37
6.5 Completion, commissioning	38
7 Communication with UNAX <sup>®</sup> system	39
7.1 Standard	39
7.2 Ethernet TCP/IP Gateway	39
8 Frequently asked questions (FAQ)	41
9 Technical data	44
9.1 Electronics, Firmware	44
9.2 Dimensions UNAX <sup>®</sup> Ux4	44
9.3 Drilling plan of electrical cabinet	44

## 1 System overview

The UNAX<sup>®</sup> system stands for clear structured control hierarchic.

The multi-axis motion coordinator is able to coordinate and interpolate up to four servo axes.

Over the internal TCP/IP communication gateway of UNAX<sup>®</sup>, programming and parameterisation of servo axis operates by web application WebMotion<sup>®</sup>.

Set IP address in browser, select servo axis and the WebMotion<sup>®</sup> graphic user interface looks like a peer to peer connection to a certain XENAX<sup>®</sup> servo controller.

UNAX<sup>®</sup> multi-axis motion coordinator



XENAX<sup>®</sup> slave servo controller



LINAX<sup>®</sup> linear motor



## 2 Electrical Connections

front

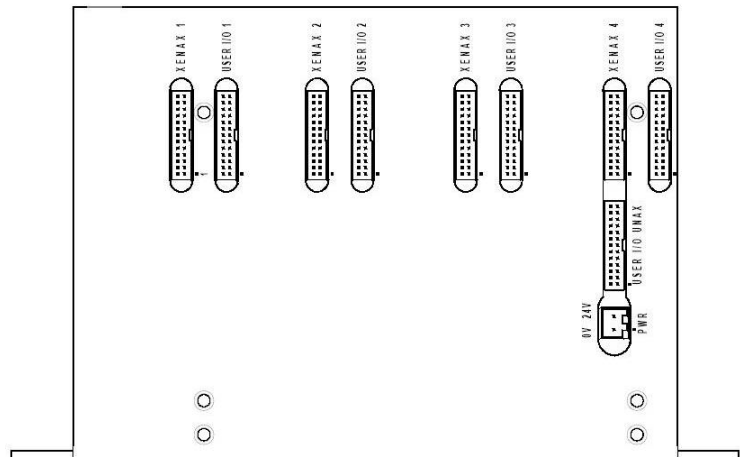


### DESCRIPTION

### PLUG

XENAX 1..XENAX 4	8 Pole Jack RJ45 with status LED
ENCODER EXT	8 Pole Jack RJ45 with status LED
TCP/IP	8 Pole Jack RJ45 with status LED
COM1	USB Type B
COM2	USB Type B

underside



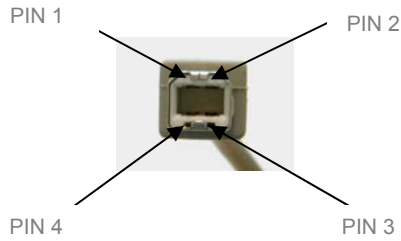
XENAX 1 ..4	26 Pole FCI connector for XENAX <sup>®</sup> control
USER I/O 1 .. 4	26 Pole FCI connector free PLC of XENAX <sup>®</sup>
USER_I/O_UNAX	26 Pole FCI connector application free programmable
PWR	2 Pole WAGO power UNAX <sup>®</sup>

## 2.1 Pin configuration

### COM 1 / COM 2

The two serial interfaces RS232 are wired to USB B connectors.

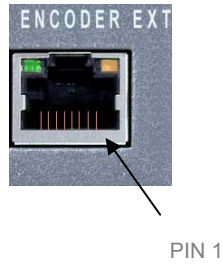
USB B connector	COM connector PC/Laptop
Pin 2	Pin 3
Pin 3	Pin 2
Pin 4	Pin 5



Communication cable is part of shipment

### ENCODER EXT

RJ45 Jack	Description
Pin 1	GND
Pin 2	+5V
Pin 3	A
Pin 4	B
Pin 5	-B
Pin 6	-A
Pin 7	Z
Pin 8	-Z



### PWR

POWER -	Pin 1	white	0, GND
POWER +	Pin 2	brown	24VDC

### XENAX 1 .. 4

PLC connection via ribbon cable to XENAX<sup>®</sup> slave servo controller.

Ribbon cables are part of shipment

USER I/O 1 .. 4

Unused PLC I/O's from XENAX®  
slave servo controller

**Note:**

Description "Input" and "Output" is from XENAX®'s  
point of view.

	Pin 1		NC, reserved for UNAX® handshake
	Pin 2		NC, reserved for UNAX® handshake
	Pin 3		NC, reserved for UNAX® handshake
Active low, NPN open collect. 50V/350mA, freewheel. diode	Pin 4	yellow	Output 4
Active low, NPN open collect. 50V/350mA, freewheel. diode	Pin 5	grey	Output 5
Active low, NPN open collect. 50V/350mA, freewheel. diode	Pin 6	pink	Output 6
Active low, NPN open collect. 50V/350mA, freewheel. diode	Pin 7	blue	Output 7
Active low, NPN open collect. 50V/350mA, freewheel. diode	Pin 8	red	Output 8
Power supply voltage protected with 100mA poly-switch With 24V supply usable for input signal level	Pin 9	black	PWR
2A	Pin 10	violet	GND
2A	Pin 11	grey-pink	GND
250mA	Pin 12	red-blue	5V
24V Pull Down *	Pin 13	white-green	Input 9
24V Pull Down *	Pin 14	brown-green	Input 10
24V Pull Down *	Pin 15	white-yellow	Input 11
24V Pull Down *	Pin 16	brown-yellow	Input 12
	Pin 17		reserved for UNAX® handshake
	Pin 18		reserved for UNAX® handshake
	Pin 19		reserved for UNAX® handshake
	Pin 20		reserved for UNAX® handshake
24V Pull Down *	Pin 21	white-blue	Input 5
24V Pull Down *	Pin 22	brown-blue	Input 6
24V Pull Down *	Pin 23	white-red	Input 7
24V Pull Down *	Pin 24	brown-red	Input 8
2A	Pin 25	white-black	GND
250mA	Pin 26	brown-black	5V

\* 24V Input with 2.7 kΩ pull down, Ri = 12,7 kΩ

USER I/O UNAX

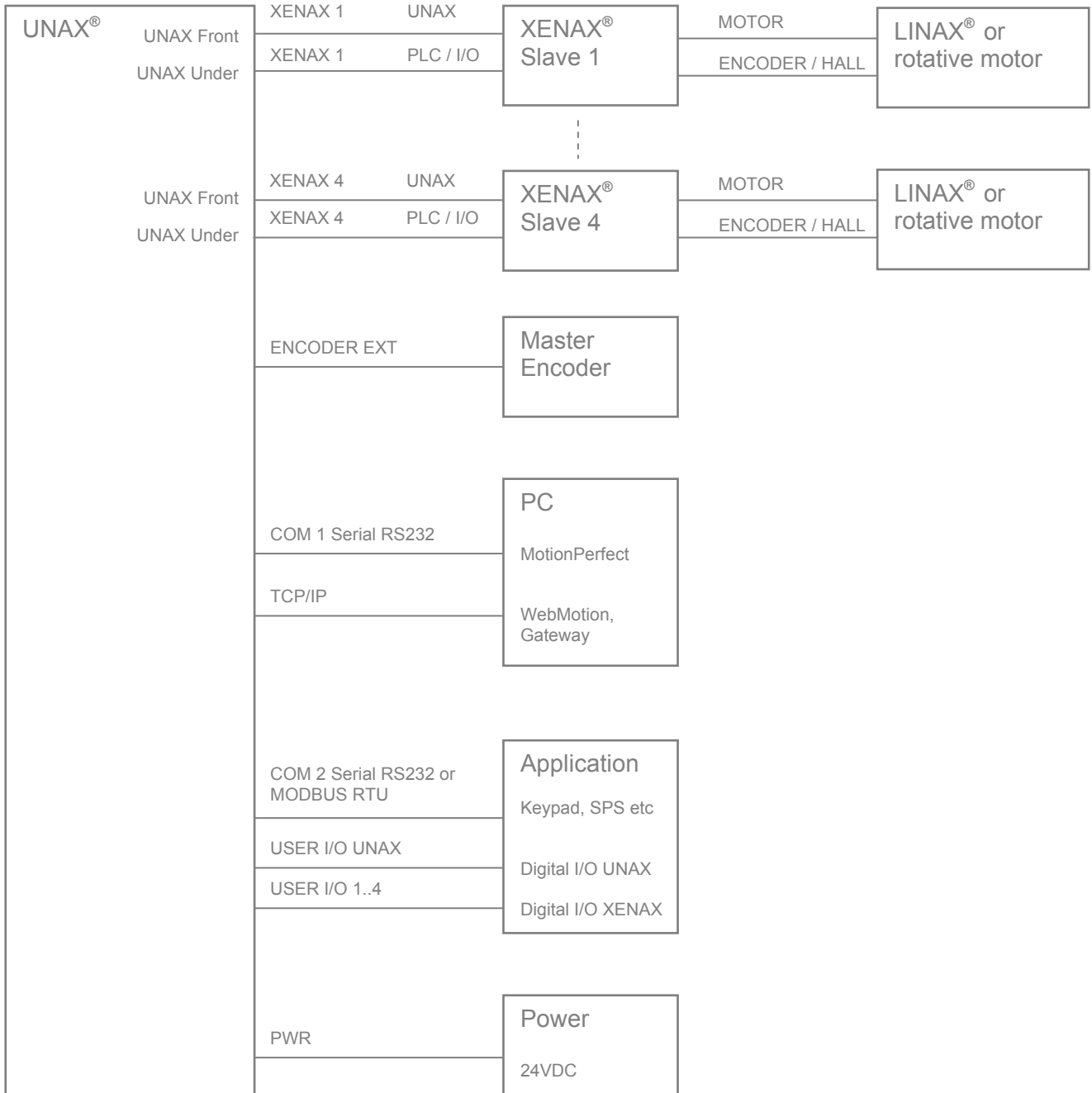
UNAX<sup>®</sup> I/O's, free programmable by application

Active low, NPN open collect. 24V/350mA, freewheel. diode	Pin 1	white	Output 1
Active low, NPN open collect. 24V/350mA, freewheel. diode	Pin 2	brown	Output 2
Active low, NPN open collect. 24V/350mA, freewheel. diode	Pin 3	green	Output 3
Active low, NPN open collect. 24V/350mA, freewheel. diode	Pin 4	yellow	Output 4
Active low, NPN open collect. 24V/350mA, freewheel. diode	Pin 5	grey	Output 5
Active low, NPN open collect. 24V/350mA, freewheel. diode	Pin 6	pink	Output 6
Active low, NPN open collect. 24V/350mA, freewheel. diode	Pin 7	blue	Output 7
Active low, NPN open collect. 24V/350mA, freewheel. diode	Pin 8	red	Output 8
24V Power supply voltage protected with 100mA poly-switch usable for input signal level	Pin 9	black	PWR
2A	Pin 10	violet	GND
2A	Pin 11	grey-pink	GND
250mA	Pin 12	red-blue	5V
	Pin 13		NC
	Pin 14		NC
	Pin 15		NC
	Pin 16		NC
24V Rin = 6.8k	Pin 17	white-grey	Input 0 / R0 <sup>1</sup>
24V Rin = 6.8k	Pin 18	brown -grey	Input 1 / R1 <sup>1</sup>
24V Rin = 6.8k	Pin 19	white-pink	Input 2 / R2 <sup>1</sup>
24V Rin = 6.8k	Pin 20	brown-pink	Input 3 / R3 <sup>1</sup>
24V Rin = 6.8k	Pin 21	white-blue	Input 4 / XENAX 3, OUT 3 <sup>2</sup>
24V Rin = 6.8k	Pin 22	brown-blue	Input 5 / XENAX 4, OUT 1 <sup>2</sup>
24V Rin = 6.8k	Pin 23	white-red	Input 6 / XENAX 4, OUT 2 <sup>2</sup>
24V Rin = 6.8k	Pin 24	brown-red	Input 7 / XENAX 4, OUT 3 <sup>2</sup>
2A	Pin 25	white-black	GND
250mA	Pin 26	brown-black	5V

<sup>1</sup> Free programmable inputs, registration inputs for position detection with an external encoder

<sup>2</sup> Free programmable inputs if XENAX 3 resp. XENAX 4 are not present.

## 2.2 System cabling



**Note :**

If the master encoder is connected (ENCODER EXT), connection XENAX 4 is not usable and must be disconnected.

### 3 Development Environment / Tools

The following tools are involved into programming a UNAX<sup>®</sup> system.

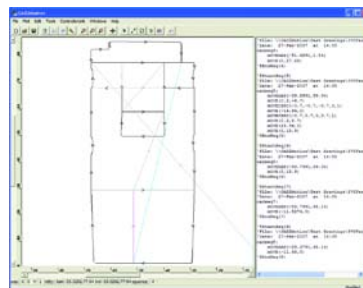
**WebMotion<sup>®</sup>**  
Parameterisation and programming of XENAX<sup>®</sup> slave servo controllers



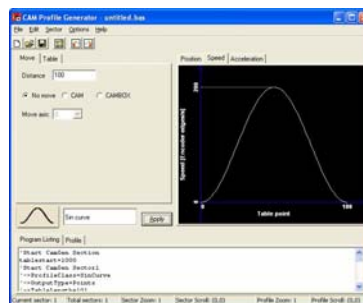
**Motion Perfect 2**  
Integrated development environment for application programming of the UNAX<sup>®</sup> multi-axis motion coordinator in BASIC syntax



**CAD to Motion**  
Import of CAD files with DXF-format. Automatically conversion to BASIC code for UNAX<sup>®</sup> application.



**Cam Gen**  
Tool to create CAM tables



## 4 Programming XENAX® slave controller

### 4.1 Ethernet connect

Connection of UNAX® to laptop/PC via Ethernet-switch with normal network cable RJ45.

For connection of the laptop / PC directly to UNAX® use a crossed cable RJ45. With newer network cards a crossed cable is not necessary any more.

LED status on Ethernet connector

Colour	LED left	Colour	LED right
off	no connection	off	no data
orange	10Mbps	orange	half duplex
green	100Mbps	green	full duplex



### 4.2 Test IP with >IPCONFIG

IPCONFIG command in DOS window

Check the TCP/IP address range:  
IP address of the PC must be in range of 192.168.2.xxx  
If necessary adjust IP address manually via „network set up“, e.g. 192.168.2.200  
xxx = 001 – 255 ≠ address UNAX®

```
Verbindungsspezifisches DNS-Suffix:
IP-Adresse (Autokonfig.) . . . . . : 192.168.2.200
Subnetzmaske . . . . . : 255.255.255.0
Standardgateway . . . . . :
```

### 4.3 Test connection with >PING

PING command in DOS window

The IP address you will find at the back of UNAX®

If there is no answer, check direct connection with a crossed cable.

If you do not know the IP address of UNAX®, you can look for it with the DeviceInstaller tool. Make sure, that the Ethernet connection is operational.

```
C:\Dokumente und Einstellungen>ping 192.168.2.100
Ping wird ausgeführt für 192.168.2.100 mit 32 Bytes Daten:
Antwort von 192.168.2.100: Bytes=32 Zeit<1ms TTL=64
Antwort von 192.168.2.100: Bytes=32 Zeit<1ms TTL=64
Antwort von 192.168.2.100: Bytes=32 Zeit<1ms TTL=64
Antwort von 192.168.2.100: Bytes=32 Zeit<1ms TTL=64
Ping-Statistik für 192.168.2.100:
    Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0 (0% Verlust)
    Ca. Zeitangaben in Millisek.:
        Minimum = 0ms, Maximum = 0ms, Mittelwert = 0ms
```

#### 4.4 IP address search / change

To view and change the IP address as well as to update WebMotion®, the **DeviceInstaller** tool is used.

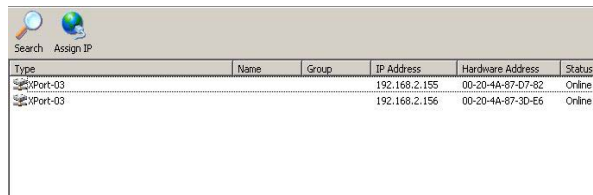
Use the adjacent link and click **DeviceInstaller**.  
Select the component **XPort** and install it.

<http://www.lantronix.com/device-networking/utilities-tools/device-installer.html>

The **DeviceInstaller** needs the Microsoft .NET framework driver. Should this driver not exist it can be downloaded by returning to the adjacent link **Microsoft .NET Framework**.

#### Search IP address of UNAX®

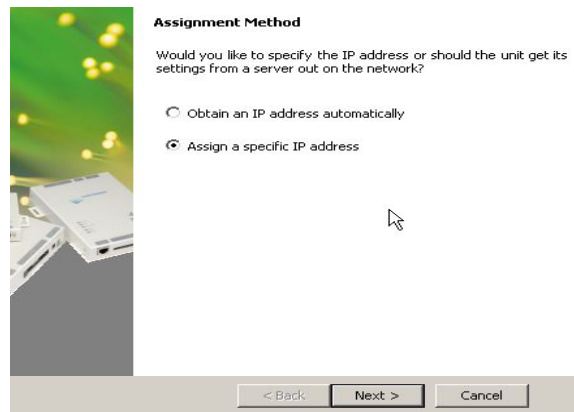
Start the **DeviceInstaller** tool of Lantronix and choose **Search** to look for existing IP addresses of XPorts.



Type	Name	Group	IP Address	Hardware Address	Status
XPort-03			192.168.2.155	00-20-4A-87-07-82	Online
XPort-03			192.168.2.156	00-20-4A-87-30-E6	Online

#### Change IP addresses

Via **Assign IP** you choose **Assign a specific IP address**. Now it is possible to set up a new IP address.

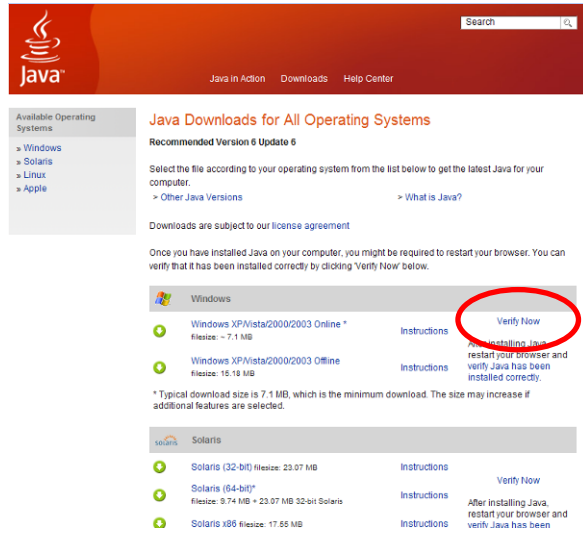


## 4.5 JAVA Plugin check

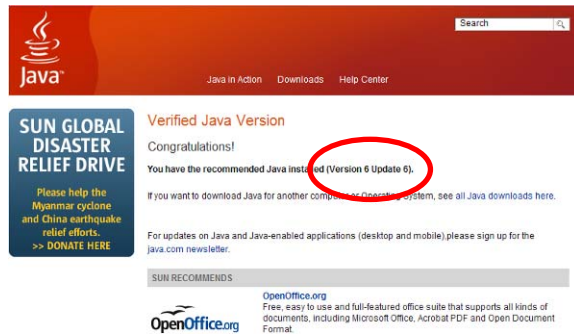
In case of problems with the JAVA software you can execute a check with the link on right side.

<http://www.java.com/de/download/manual.jsp>

By selecting *Verify now* (marked red on right), your installed JAVA Plugin version will appear.



The version of JAVA Plugin Software must be 1.4 or higher. The verified JAVA version is the first figure after the dot. The example on the right shows the version 1.6.



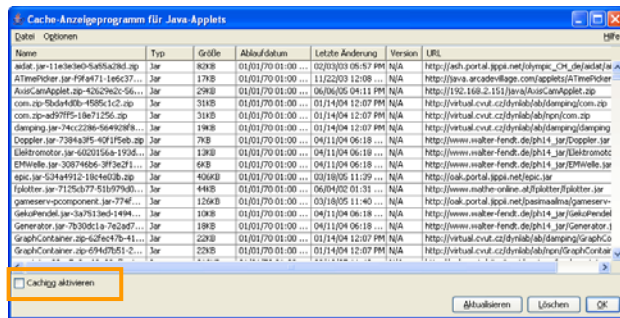
If a lower version is installed, it can be updated with a Java software download. The current version of JAVA Runtime Environment (JRE) is also available on the enclosed CD.

## 4.6 Applet Cache

To provide a correct loading procedure of WebMotion® applet into the browser, caching functionality of java plugin must be switched off.

Otherwise the loading procedure can be unreliable.

Open Java Control Panel with:  
start / systemcontrol (classic view) / java  
temporary internet files / setup  
change view to „show applets“



The checkbox must be switched off

## 4.7 WebMotion®

WebMotion® is an integrated graphic user interface into UNAX® device, based on web technology (applet), to parameterisation and programming of XENAX® slave servo controllers.

Standard web browsers like Internet Explorer, Mozilla, Firefox, Opera and so on are able to load and execute the WebMotion® applet.

Start web browser with the IP address (back side of UNAX®) with extension /unax.html

<http://192.168.2.xxx/unax.html>

The main site of UNAX® shows the available XENAX® servo controllers (XENAX 1..n).

After selection of a servo controller, the graphic user interface looks like a peer to peer connection to a standard XENAX® servo controller. Parameterisation and programming can be done.

Return to UNAX® main site occurs by the button exit in menu ident/main of WebMotion®



### Note:

After switching on the UNAX® power supply, wait at least 10 seconds until the UNAX® web server is ready for operating. Only afterwards start the browser with WebMotion®.

On interruption of the UNAX® power supply, please exit the browser. When turning on wait 10 seconds again before restarting the browser with WebMotion®.

Further information of WebMotion® is included into XENAX® user manual.

## 4.8 Set up XENAX<sup>®</sup> for operation with UNAX<sup>®</sup>

For operation of the XENAX<sup>®</sup> servo controller with the UNAX<sup>®</sup> multi-axis coordinator, following parameters must be set:

MODE	2	Operation mode stepper control
INC PER PULSE	1	Increment per Pulse count. Together with the definition UNIT = 16 into UNAX <sup>®</sup> application, all move commands into UNAX <sup>®</sup> program are based on µm units. 1 INC = 1 µm

### Input Functions

The UNAX<sup>®</sup> multi-axis motion coordinator is able to control four digital inputs (Input 1..4) of each connected XENAX<sup>®</sup> slave servo controller.

This inputs must be defined as Input Functions on XENAX<sup>®</sup> slave servo controller to take effect (for detailed information, see manual XENAX<sup>®</sup>).

Recommended programming:

Initialisation of the XENAX<sup>®</sup> slave servo axis (HORM) and motion to a defined start position (IX).

HORM must be executed just once after switching on the servo controller. The successful HORM sequence can be sent to UNAX<sup>®</sup> via digital Output of XENAX<sup>®</sup> slave servo controller

Stop with Emergency Exit (EE). The power stage of XENAX<sup>®</sup> slave servo controller will be switched off.

free programmable, additional motion indexes for example

Input Function 1 (PG1)

Input Function 2 (EE)

Input Function 3, 4

### Output Functions

Each XENAX<sup>®</sup> slave servo controller can send status information to the UNAX<sup>®</sup> by three digital outputs (Output 1..3)

Examples:  
Homing sequence done, error, position reached

## 5 Programming the UNAX® multi-axis coordinator

### 5.1 Motion Perfect 2

Integrated development environment for programming the UNAX multi-axis motion coordinator.

Up to five multi threading tasks are programmable in Trio BASIC Syntax.

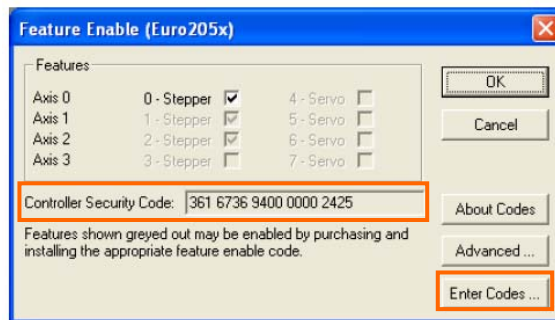
To get detailed information about installation and handling, please see the manual *MotionPerfect.pdf* on CD.



### 5.2 Axis configuration

The UNAX® multi-axis motion coordinator is able to coordinate up to four axes. The actual axis configuration is listed in the menu *Controller / enable features*

To enable additional axes (up to 4), send the controller security code to Jenny Science. Then you will get an activation code to enter in the menu *Enable / Enter Codes*



### 5.3 BASIC command set

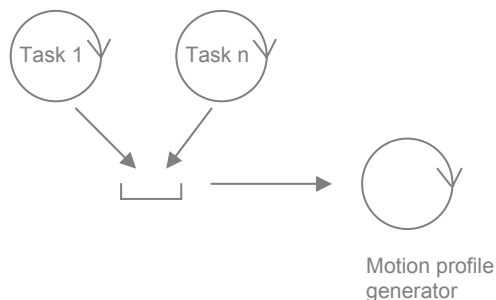
The programming language of UNAX® is a BASIC Syntax with specific command set.

For detailed information, please see the document *Programming.pdf* and chapter 8 *Trio Basic commands* in full manual

### 5.4 Tasks and motion profile generator

Each task writes the motion commands into an input buffer system of the motion profile generator. (MTYPE, NTYPE, PMOVE, see TrioBasic commands)

To synchronise on a specific task program line, the command WAIT IDLE is used. For example, set an output after a motion profile is finished



## 5.5 System module UNAX\_SYS.bas

Software task with defined user interface to control the hardware outputs of UNAX<sup>®</sup>

The control interface includes the global variables VR(0..3), see also document *Programming.pdf*

VR(0)	USER I/O UNAX
VR(1)	XENAX 1 / 2
VR(2)	XENAX 3 / 4
VR(3)	7 segment LED

### **Important:**

The command OP (Output) is reserved for the software task UNAX\_SYS. Do not use OP in the user application.

## 5.6 User I/O UNAX

I/O interface of UNAX<sup>®</sup>, freely programmable into user application.

Make sure that UNAX\_SYS is running

Output No	Pin No	VR(0) Bit No	valency
1	1	0	1
2	2	1	2
3	3	2	4
4	4	3	8
5	5	4	16
6	6	5	32
7	7	6	64
8	8	7	128

### Set Outputs

Output control works via global variable VR(0).  
Depending on output combination, the sum of valency must be written into VR(0).

#### Example 1

Set outputs 1,6,7 for a duty cycle of 10 ms.  
VR(0) = 97  
WA(10)  
VR(0) = 0

#### Example 2

Set output 4 only:  
Output 4 means Bit Nr.3 of VR(0), so  
SET\_BIT(3,0), in the same way  
CLEAR\_BIT(3,0) in case to clear output.

Inputs 0..7 are mirrored in BASIC variable IN.  
With the bit number as parameter, test of a specific input is possible.

### Read Inputs

#### Example:

```

Program branch through logic combination of
Input 3 (Pin 20) AND Input 5 (Pin 22)
IF (IN(3)=ON AND IN(5)=ON) THEN
    GOTO inp_on
ELSE
    GOTO inp_off
ENDIF
    
```

#### Important:

If XENAX 3 is connected, IN Bit 4 is used for UNAX<sup>®</sup> handshake with XENAX<sup>®</sup> slave servo controller. Therefore, USER I/O UNAX input 4 is busy. In the same way, inputs 5..7 of USER I/O UNAX are busy, if XENAX 4 is connected.

### 5.7 XENAX® handshake with UNAX®

Each XENAX® slave servo controller is connected via ribbon cable with UNAX® multi-axis motion coordinator.  
(Connector XENAX 1..4 on underside of UNAX®).

Make sure that UNAX\_SYS is running

#### Set inputs on XENAX® slave servo controller

UNAX® is able to control Input 1..4 of each connected XENAX® slave servo controller. So, input functions on XENAX® can be triggered from UNAX® (homing, programs or indexes for example)

XENAX Nr	Input Nr	VR(1) Bit Nr	valency
1	1	0	1
	2	1	2
	3	2	4
	4	3	8
2	1	4	16
	2	5	32
	3	6	64
	4	7	128

XENAX Nr	Input Nr	VR(2) Bit Nr	valency
3	1	0	1
	2	1	2
	3	2	4
	4	3	8
4	1	4	16
	2	5	32
	3	6	64
	4	7	128

#### Example

Assumption: Two XENAX® slave servo controller (XENAX 1, 2) are connected with UNAX®. The homing function has to be triggered via Input 1 on each controller at the same time.

VR(1)=17  
WA(100)  
VR(1)=0

In case of three XENAX® slave servo controller (XENAX 1,2,3)

VR(1)=17  
VR(2)=1  
WA(100)  
VR(1)=0  
VR(2)=0

Read outputs from XENAX® slave servo controller

UNAX® can read the digital outputs 1..3 of each connected XENAX® slave servo controller.

XENAX Nr	Output Nr	IN Bit Nr
1	1	8
	2	9
	3	10
2	1	11
	2	12
	3	13
3	1	14
	2	15
	3	4*
4	1	5*
	2	6*
	3	7*

**Example:**

Assumption: Each XENAX® slave servo controller sets output function 1 (HOME), if the homing sequence is finished successfully. The application has to wait, until the homing sequence of XENAX 1 and XENAX 2 is finished.

```

wait_ho:
IF (IN(8) = OFF OR IN(11) = OFF) THEN
GOTO wait_ho
ENDIF
    
```

**\* Important:**

If XENAX 3 is connected, IN Bit 4 is used for UNAX® handshake with XENAX® slave servo controller. Therefore, USER I/O UNAX input 4 is busy. In the same way, inputs 5..7 of USER I/O UNAX are busy, if XENAX 4 is connected.

### Control 7-segment LED display

The 7-segment display of UNAX<sup>®</sup> is free programmable by the user application. It can be used for application specific status information.

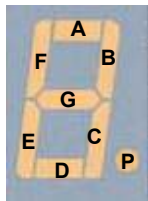
#### display 0..9, A..F

LED digits/character	VR(3) value
0..9	0..9
A	10
B	11
C	12
D	13
E	14
F	15

#### segment control

If value VR(3)>15 then segment control with offset value = 16 is activated.

LED segment	val
blank	0
segment F	1
segment G	2
segment E	4
segment D	8
segment C	16
segment B	32
segment A	64
point	128



$$VR(3) = val + 16$$

#### Examples:

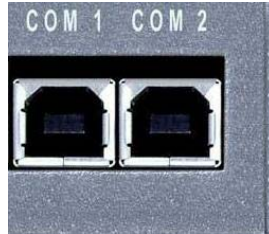
Display	VR(3)
digit 5	5
character E	14
segment G	18
character P	119

## 5.8 COM 1 / COM 2 Interface

### COM 1

Internal connected to Euro 205x, port 0.  
Serial ASCII command interface of Euro205x,  
MotionPerfect2 interface

All commands of TRIO BASIC command set  
can be used (see document  
*CommandReference.pdf*)



### Protocol

Each response of Euro205x is terminated with  
[CR][LF]>>  
Values of variables can be requested by prefix „?“

### Examples

Request global variable VR(5):  
?VR(5)  
1.2345[CR][LF]>>

Set global variable VR(3) with value=12:  
VR(3)=12

Read axis 0 parameter acceleration:  
BASE(0)  
?ACCEL

### COM 2

Internal connected to Euro 205x, port 1.  
This serial interface is free programmable by the  
application.

PRINT  
Send string

INPUT  
Waits for a string to be received, terminated with  
a carriage return <CR>

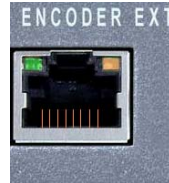
Optionally, MODBUS RTU can be defined with  
command SETCOM to connect adapted  
keypads or HMI.  
(see also document *full manual6.7.pdf* chapter  
13-3)

## 5.9 External Encoder

The UNAX<sup>®</sup> application can read a master axis via external encoder connector.

The command MPOS returns the current encoder position.

Example:  
CurrentPos = MPOS AXIS(3)



**Note:**

If less than four stepper axes are configured (see chapter 5.2 *axis configuration*), the fourth axis automatically works in ENCODER EXT mode, ATYPE AXIS(3) = 3.

Make sure that connector XENAX 4 is not connected.

If the fourth axis is enabled, the axis type must be set explicit into UNAX<sup>®</sup> application with ATYPE AXIS(3) = 1 (stepper axis)

## 6 Application example

An easy example shows the basic mechanism of programming a UNAX<sup>®</sup> system.

The user gets a quick and efficient insight of the UNAX<sup>®</sup> programming and is able to develop real applications with the aid of a substantial documentation.

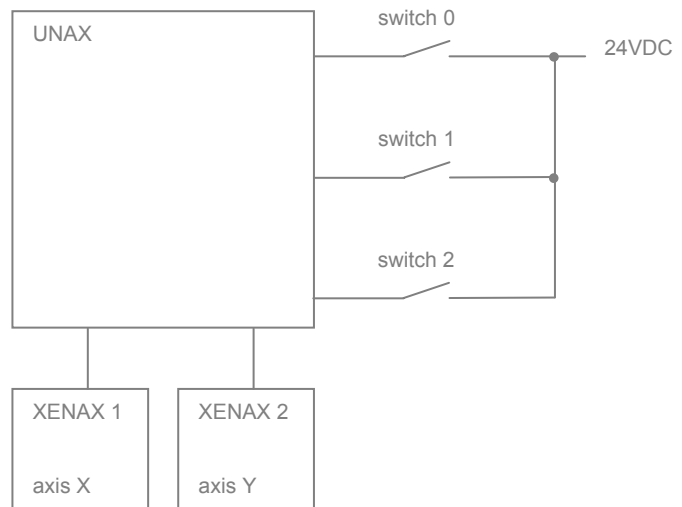
### 6.1 Exercise

The UNAX<sup>®</sup> system includes two XENAX<sup>®</sup> slave servo controllers with LINAX<sup>®</sup> linear stages in XY-cross table configuration.

XENAX 1 axis X  
XENAX 2 axis Y

Three switches are connected to the USER I/O UNAX<sup>®</sup> interface.

switch 0	Pin 17	0=	start
		1=	stop
switch 1	Pin 18	0=	curve-8
		1=	CAD-curve
switch 2	Pin 19	1=	reset



### Functionality of the program example

After power ON of the UNAX<sup>®</sup> system, reset and homing sequence of both axes must be done.

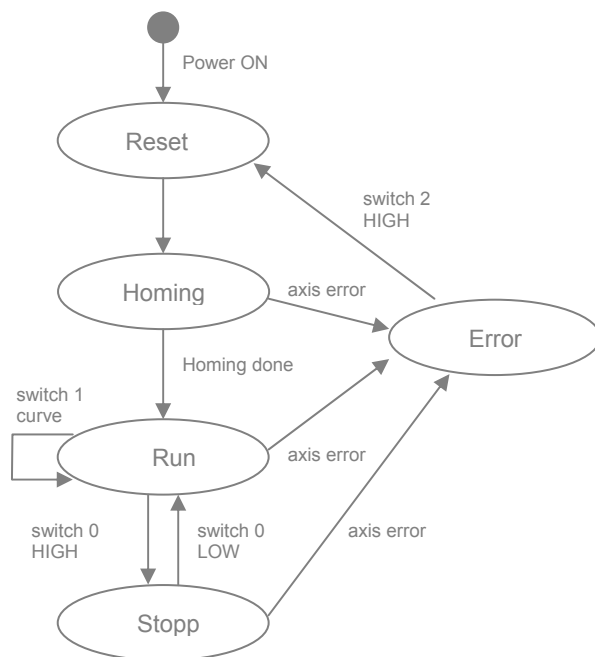
The homing sequence includes the initialisation of the axis (HORM) as well as positioning the slide in middle position.

Afterwards, the motion starts according selected curve through switch 1.

0	curve of an „8“
1	imported CAD curve

Switch 0 stops the motion in middle position.

In case of an axis error (display „E“ on UNAX<sup>®</sup>), switch 2 starts the reset sequence and the application restarts again.



## 6.2 Parameterisation XENAX® slave servo controller

The parameterisation and programming of the XENAX® slave servo controllers can be done centrally via integrated web application WebMotion® inside of UNAX®.

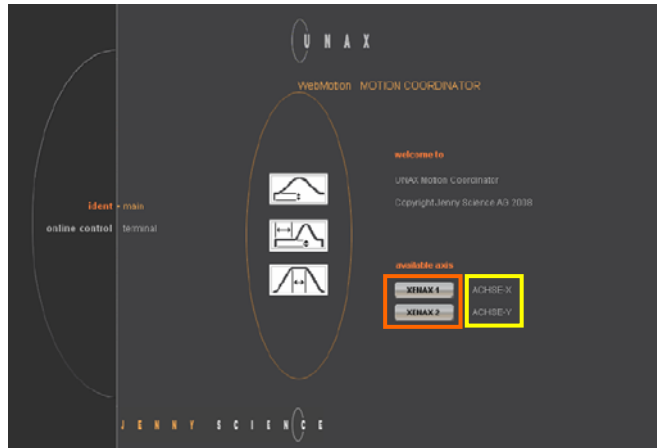
start browser with URL:  
<http://192.168.2.100/Unax.html>

**Note:**

The UNAX® probably does not have the standard IP address 192.168.2.100, therefore the URL has to be adapted to the current IP address of UNAX®.

If the IP address of UNAX® is unknown, the tool *Device Installer* is able to scan for it.

The two available axes into the UNAX® system are shown on the start site with buttons XENAX 1 and XENAX 2.



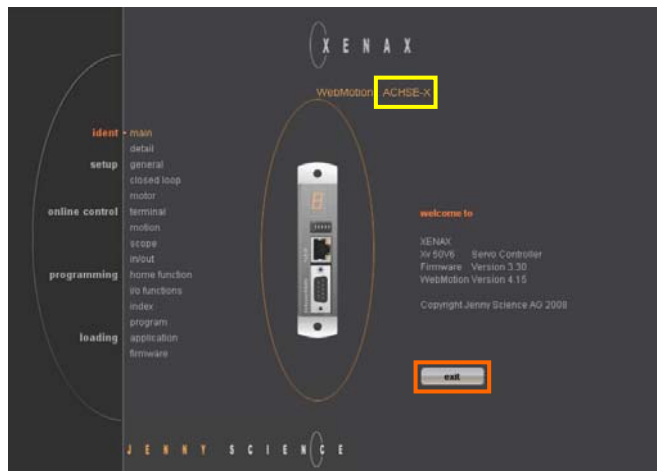
With the selection of an axis, WebMotion® looks like a peer to peer connection to a standard XENAX® servo controller. In the menu *ident/main*, the button *exit* returns to UNAX® start site.

Each connected XENAX® slave servo controller can be labelled with a user defined string.

The corresponding command is SID (Servo IDentification), entered in menu *online control / terminal* (see also Xenax® manual chapter *ASCII command set*)

**Example**

Define axis identification „ACHSE-X“  
 command: SIDACHSE-X



The Homing Sequence has to be defined on both XENAX<sup>®</sup> slave servo controller with Program 1.

Afterwards index 1 drives to the absolute middle position of the LINAX<sup>®</sup> stage (assumption Linax LX-44)

Output 1 indicates the completion of the homing sequence to UNAX<sup>®</sup>.

## Homing

### Programm 1

Line 1	CO	1	Clear Output 1
Line 2	HORM		Home Ref Mark
Line 3	IX	1	Index 1
Line 4	SO	1	Set Output 1

### Index 1

Acceleration	1000000
Speed	10000
Distance	22000
Abs/Rel	Abs

### Input Function 1

PG1 (Programm 1)

Reset of XENAX<sup>®</sup> slave servo controller is programmed as an Input function EE (Emergency Exit). The power stage is switched off.

## Reset

### Input Function 2

EE (Emergency Exit)

UNAX<sup>®</sup> must be able to detect axis errors. Therefore, the Output Function 2 (Error) of each XENAX<sup>®</sup> slave servo controller must be supervised by UNAX<sup>®</sup>.

## Error

### Output Function 2

ERR (Error)

To support the Pulse/Direction interface of UNAX<sup>®</sup> correctly, XENAX<sup>®</sup> slave controllers must be set into stepper mode control (WebMotion menu *setup / general*):

Mode	2
INC per Pulse	1

## Parameterisation for UNAX<sup>®</sup> operation

### 6.3 UNAX® application

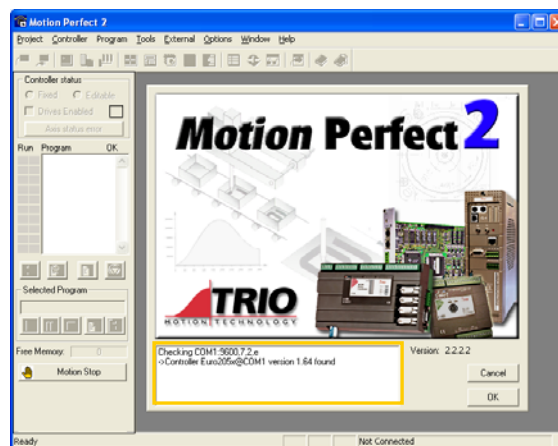
The programming of UNAX® multi-axis motion coordinator is supported by the integrated development environment *Motion Perfect 2*. For detailed information, please see manual *MotionPerfect.pdf* on CD.

#### Create new project

The start screen of Motion Perfect 2 shows hardware- and firmware identification of Euro205x board inside of UNAX®.

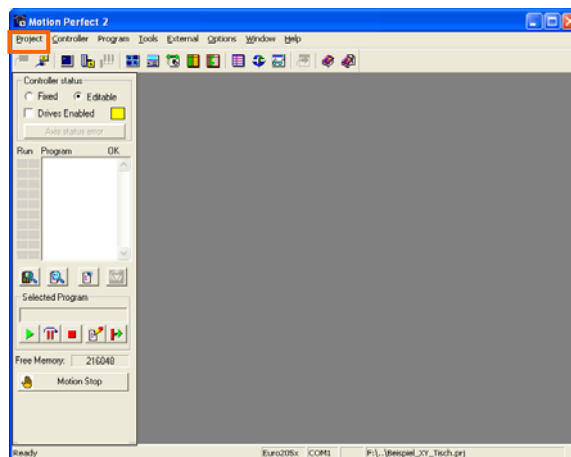
In this case, the communication is established and the tool ready for operation.

Otherwise, check the cabling and baudrate settings.



Create a new Project folder in Project / New project

The project file has the extension \*.prj



## Include UNAX<sup>®</sup> systemtask

UNAX<sup>®</sup> systemtask is used for hardware control of UNAX<sup>®</sup>.

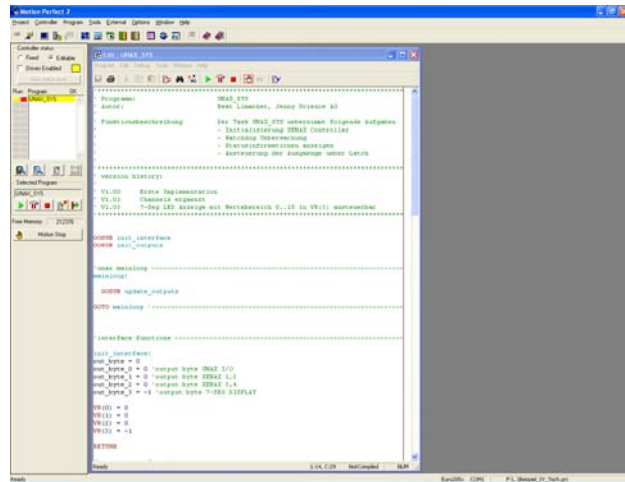
Copy *UNAX\_SYS.bas* into project folder

Load *UNAX\_SYS.bas* in menu *Project / Load Program File* into the project

With double click on *UNAX\_SYS*, the source code will be shown.

### Note:

Please do not modify the file *UNAX\_SYS.BAS*. It is supported by JennyScience AG.

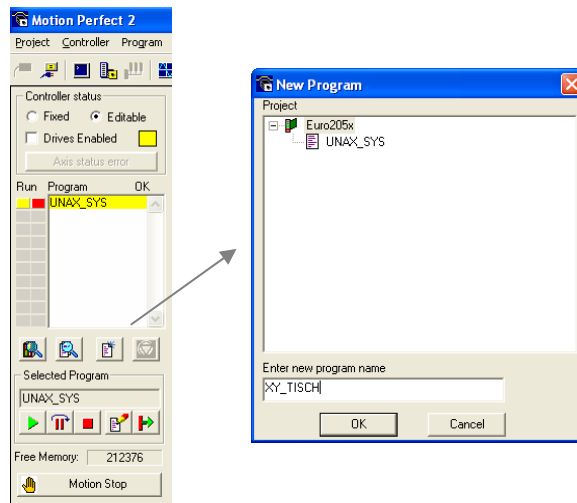


## Create application task

In the following chapters, the application is build up step by step with the matching descriptions.

First step, we add a new task into the project.

Click on the button  to open the filedialog and create a new application task "XY\_TISCH.BAS".



### Program framework (XY\_TISCH\_1.bas)

The framework can be written into the program editor.

**Note:**

The program steps are included on CD in UNAX\_Programmierung / Beispielprogramm / Beispiel\_XY\_Tisch

The files can be loaded in the menu *Project / load program file*

**Description**

**RUN „UNAX\_SYS“**

Make sure the systemtask UNAX\_SYS is running before the control variables VR(0..3) are modified.

**MICROSTEP=OFF**

Defined for stepper control, do not modify

**UNITS AXIS(x) = 16**

Setting needed for general UNAX® distance unit of 1 µm

**MERGE = ON**

Merge of consecutive moves. The axis will not ramp down to zero between the moves. The functionality can disabled by MERGE = OFF

**Start program**

Click on the red colored square of XY\_TISCH resp. XY\_TISCH\_1, if the color changes to green, the task is running.

Task UNAX\_SYS starts through XY\_TISCH

The functionality is shown by a „0“ on the UNAX® 7-segment display. Tests with other values are possible, values 0..15 into VR(3).

**Stop program**

To stop all the tasks, click on the green squares. The source code can be modified only if all tasks are stopped.

```

*****
' Programm:                Beispielapplikation XY Tisch
' Autor:                   Jenny Science AG
*****

RUN "UNAX_SYS"
WA(500) 'warten auf systemtask

MICROSTEP=OFF 'Mikrostep Modus

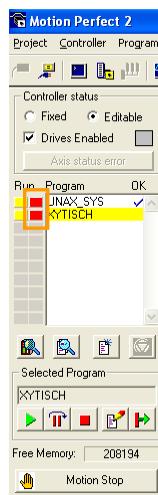
MERGE=ON , Aneinanderreihung von Fahrbefehlen ohne Verzögerungsrampen

UNITS AXIS(0) = 16 'Achspanparameter Achse 0 setzen
SPEED AXIS(0) = 20000
ACCEL AXIS(0) = 10000000
DECEL AXIS(0) = 10000000

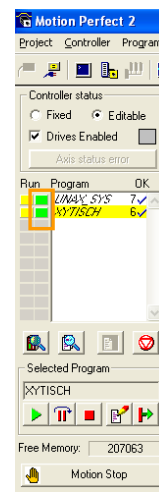
UNITS AXIS(1) = 16 'Achspanparameter Achse 1 setzen
SPEED AXIS(1) = 20000
ACCEL AXIS(1) = 10000000
DECEL AXIS(1) = 10000000

VR(0)= 0 'alle Ausgaenge von USER I/O UNAX loeschen
VR(1)= 0 'alle Inputs zu XENAX 1,2 loeschen
VR(2)= 0 'alle Inputs zu XENAX 3,4 loeschen
VR(3)= 0 '7-Seg Anzeige auf 0 setzen

WHILE 1=1 '-----HAUPTLOOP-----
WEND '-----
    
```



Tasks stopped



Tasks running

## Reset / Homing (XY\_TISCH\_2.bas)

The input function 2 of XENAX® slave servo controller is defined as Emergency Exit (EE). To trigger this reset function, a signal pulse is required on input 2 of both servo controllers.

In the same way, the homing sequence is defined on input function 1.

See also chapter 5.7 XENAX® handshake with UNAX® / Set inputs on XENAX® slave servo controller

```

*****
' Programm:           Beispielapplikation XY Tisch
' Autor:             Jenny Science AG
*****
CONSTANT "state_reset",1
CONSTANT "state_homing",2
CONSTANT "state_run",3

RUN "UNAX_SYS"
WA(500) 'warten auf systemtask

MICROSTEP=OFF 'Mikrostep Modus

MERGE=ON 'Fahrbefehle werden aneinandergereicht ohne Verzögerungsrampen

UNITS AXIS(0) = 16 'Achsparemeter Achse 0 setzen
SPEED AXIS(0) = 20000
ACCEL AXIS(0) = 10000000
DECEL AXIS(0) = 10000000

UNITS AXIS(1) = 16 'Achsparemeter Achse 1 setzen
SPEED AXIS(1) = 20000
ACCEL AXIS(1) = 10000000
DECEL AXIS(1) = 10000000

VR(0)= 0 'alle Ausgaenge von USER I/O UNAX loeschen
VR(1)= 0 'alle Inputs zu XENAX 1,2 loeschen
VR(2)= 0 'alle Inputs zu XENAX 3,4 loeschen
VR(3)= 0 '7-Seg Anzeige auf 0 setzen

VR(10) = state_reset ' Status der Applikation

WHILE 1=1 '-----HAUPTLOOP-----

    ON VR(10) GOSUB do_reset, do_homing, do_run

WEND '-----

do_reset:
    VR(1)=34 'Input 2 setzen bei XENAX 1,2 -> Input Function EE ausloesen
    WA(100)
    VR(1)=0 'alle Inputs zu XENAX 1,2 deaktivieren
    WA(1000)

    VR(1)=VR(1) OR $11 'Input 1 setzen bei XENAX 1,2 -> Input Funct. PG1, Homing
    WA(100)
    VR(1)=VR(1) AND $EE 'Input 1 loeschen bei XENAX 1,2
    VR(3) = 1 '7-Seg Anzeige UNAX auf 1 setzen
    VR(10)=state_homing
RETURN

do_homing:
    IF (IN(8)=ON AND IN(11)=ON) THEN 'Output 1 lesen von XENAX 1,2 Homing fertig ?
        VR(10)=state_run
    ENDIF
RETURN

do_run:
RETURN

```

## Drive motion path of an "8" (XY\_TISCH\_3.bas)

**DEFPOS(0,0)**  
Sets internal position counter of the interpolator to zero.

**WDOG = ON**  
Enables power stage of UNAX® pulse / direction interface.

**BASE(0,1)**  
Used to direct subsequent motion commands and axis parameter reads/writes to a particular axis, or group of axes.

**MOVEABS(0,0)**  
Absolute position move to start position

**MOVECIRC()**  
Circular move with two orthogonal axes.  
(see command reference for detailed information)

**WAIT IDLE**  
Suspend program execution until the base axis has finished executing its current move and any further buffered move.

```

*****
' Programm:                Beispielapplikation XY Tisch
' Autor:                   Jenny Science AG
*****

CONSTANT "state_reset",1
CONSTANT "state_homing",2
CONSTANT "state_run",3

RUN "UNAX_SYS"
WA(500) 'warten auf systemtask

MICROSTEP=OFF 'Mikrostep Modus

MERGE=ON 'Fahrbefehle werden aneinandergereicht ohne Verzoegerungsrampen

UNITS AXIS(0) = 16 'Achspanparameter Achse 0 setzen
SPEED AXIS(0) = 20000
ACCEL AXIS(0) = 10000000
DECEL AXIS(0) = 10000000

UNITS AXIS(1) = 16 'Achspanparameter Achse 1 setzen
SPEED AXIS(1) = 20000
ACCEL AXIS(1) = 10000000
DECEL AXIS(1) = 10000000

VR(0)= 0 'alle Ausgaenge von USER I/O UNAX loeschen
VR(1)= 0 'alle Inputs zu XENAX 1,2 loeschen
VR(2)= 0 'alle Inputs zu XENAX 3,4 loeschen
VR(3)= 0 '7-Seg Anzeige auf 0 setzen

VR(10) = state_reset ' Status der Applikation

WHILE 1=1 '-----HAUPTLOOP-----
    ON VR(10) GOSUB do_reset, do_homing, do_run
WEND '-----

do_reset:
    VR(1)=34 'Input 2 setzen bei XENAX 1,2 -> Input Function EE ausloesen
    WA(100)
    VR(1)=0 'alle Inputs zu XENAX 1,2 deaktivieren
    WA(1000)

    VR(1)=VR(1) OR $11 'Input 1 setzen bei XENAX 1,2 -> Input Funct. PGI, Homing
    WA(100)
    VR(1)=VR(1) AND $EE 'Input 1 loeschen bei XENAX 1,2
    VR(3) = 1 '7-Seg Anzeige UNAX auf 1 setzen
    VR(10)=state_homing
    RETURN

do_homing:
    IF (IN(8)=ON AND IN(11)=ON) THEN 'Output 1 lesen von XENAX 1,2 Homing fertig ?
        DEFPOS (0,0) 'Homing abgeschlossen, absolute Position in UNAX nullen
        WDOG=ON 'Endstufe UNAX freischalten fuer Puls-Richtungssteuerung
        BASE(0,1) 'Definition der Achsgruppe fuer Fahrbefehle
        VR(10)=state_run
    ENDF
    RETURN

do_run:
    VR(3) = 2 '7-Seg Anzeige UNAX auf 2 setzen
    MOVEABS(0,0) '8-Kurve
    MOVECIRC(0,0,0,10000,0)
    MOVECIRC(0,0,0,-10000,1)
    WAIT IDLE
    RETURN
    
```

## Start / Stop functionality (XY\_TISCH\_4.bas)

The motion execution can be stopped by input 0 (Pin17) of USER I/O UNAX interface.

IN returns the value of the digital inputs of the UNAX®.

For detailed information, see chapter 5.6 User I/O UNAX/ Read inputs

```

*****
' Programm:      Beispielapplikation XY Tisch
' Autor:        Jenny Science AG
*****

CONSTANT "state_reset",1
CONSTANT "state_homing",2
CONSTANT "state_run",3
CONSTANT "state_stop",4

RUN "UNAX_SYS"
WA(500) 'warten auf systemtask

MICROSTEP=OFF 'Mikrostep Modus

MERGE=ON 'Fahrbefehle werden aneinandergereiht ohne Verzoegerungsrampen

UNITS AXIS(0) = 16 'Achsparemeter Achse 0 setzen
SPEED AXIS(0) = 20000
ACCEL AXIS(0) = 10000000
DECEL AXIS(0) = 10000000

UNITS AXIS(1) = 16 'Achsparemeter Achse 1 setzen
SPEED AXIS(1) = 20000
ACCEL AXIS(1) = 10000000
DECEL AXIS(1) = 10000000

VR(0)= 0 'alle Ausgaenge von USER I/O UNAX loeschen
VR(1)= 0 'alle Inputs zu XENAX 1,2 loeschen
VR(2)= 0 'alle Inputs zu XENAX 3,4 loeschen
VR(3)= 0 '7-Seg Anzeige auf 0 setzen

VR(10) = state_reset ' Status der Applikation

WHILE 1=1 '-----HAUPTLOOP-----
    ON VR(10) GOSUB do_reset, do_homing, do_run, do_stop
WEND '-----

do_reset:
    VR(1)=34 'Input 2 setzen bei XENAX 1,2 -> Input Function EE ausloesen
    WA(100)
    VR(1)=0 'alle Inputs zu XENAX 1,2 deaktivieren
    WA(1000)

    VR(1)=VR(1) OR $11 'Input 1 setzen bei XENAX 1,2 -> Input Funct. PGI,Homing
    WA(100)
    VR(1)=VR(1) AND $EE 'Input 1 loeschen bei XENAX 1,2
    VR(3) = 1 '7-Seg Anzeige UNAX auf 1 setzen
    VR(10)=state_homing
RETURN

do_homing:
    IF (IN(8)=ON AND IN(11)=ON) THEN 'Output 1 lesen von XENAX 1,2 Homing fertig ?
        DEFPOS (0,0) 'Homing abgeschlossen, absolute Position in UNAX nullen
        WDOG=ON 'Endstufe UNAX freischalten fuer Puls-Richtungsansteuerung
        BASE(0,1) 'Definition der Achsgruppe fuer Fahrbefehle
        VR(10)=state_run
    ENDIF
RETURN

do_run:
    IF (IN(0)=ON) THEN 'USER I/O UNAX Eingang 0 (Stoptaste) gesetzt ?
        VR(10)=state_stop
    ELSE
        VR(3) = 2 '7-Seg Anzeige UNAX auf 2 setzen
        MOVEABS(0,0) '8-Kurve
        MOVECIRC(0,0,0,10000,0)
        MOVECIRC(0,0,0,-10000,1)
        WAIT IDLE
    ENDIF 'Stoptaste
RETURN

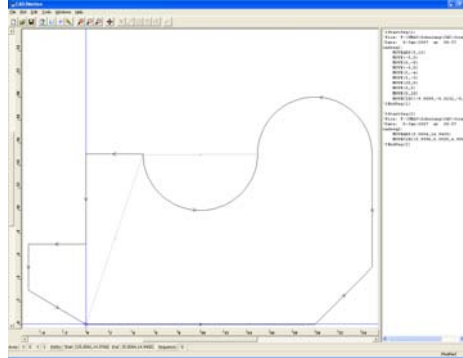
do_stop:
    VR(3)=1 '7-Seg Anzeige UNAX auf 1 setzen
    IF (IN(0)=ON) THEN 'USER I/O UNAX Eingang 0, Stoptaste aktiv ?
        VR(10)=state_stop
    ELSE
        VR(10)=state_run
    ENDIF
RETURN

```

### Import DXF- file (XY\_TISCH\_5.bas)

CAD2Motion is a program designed to allow users to translate CAD generated two dimensional motion paths in DXF format into Trio BASIC programs.


DXF-Import:  
File / Import / DXF  
the example file *form\_2.dxf* is located on CD.

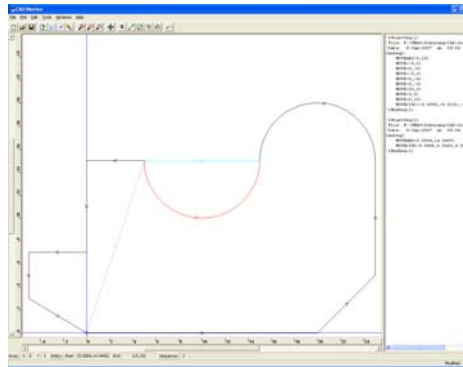


### Adjustments, scaling

The following adjustments are examples that demonstrate the modification possibilities of an imported file.

The path is divided into sections with marked motion direction. Often, the suggested sequence is not ideal and must be modified manually.


For example, the direction of a section must be changed. Select section and click on button . The motion direction changes to the opposite.

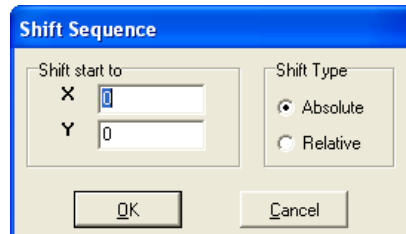


The tool has created two path sequences automatically. Now we want to reduce it to just one sequence. So append MOVECIRC command of sequence 2 at the end of sequence 1. Then delete the complete sequence 2.

Redraw display in menu *Plot / Redraw* (F5)

```
'$StartSeq(1)
'File: F:\UNAX\Schulung\CAD\form_2.dxf
'Date: 8-Jan-2007 at 09:04
cadseq1:
:
MOVE (20,0)
MOVE (5,5)
MOVE (0,10)
MOVECIRC(-9.9999,-0.0232,-5,0,0)
MOVECIRC(-9.9996,-0.0595,-5,0,1)
'$EndSeq(1)
```

With the button shift sequence, the path origin can be shifted to zero position. Select *cadseq1* into code editor, click the button  and set X and Y coordinates to zero.





## Complete application with error handling (XY\_TISCH\_6.bas)

Definition of an error state

```

*****
' Programm:                Beispielapplikation XY Tisch
' Autor:                   Jenny Science AG
*****

CONSTANT "state_reset",1
CONSTANT "state_homing",2
CONSTANT "state_run",3
CONSTANT "state_stop",4
CONSTANT "state_error",5

RUN "UNAX_SYS"
WA(500) 'warten auf systemtask

MICROSTEP=OFF 'Mikrostep Modus

MERGE=ON 'Fahrbefehle werden aneinandergereiht ohne Verzoegerungsrampen

UNITS AXIS(0) = 16 'Achspanparameter Achse 0 setzen
SPEED AXIS(0) = 20000
ACCEL AXIS(0) = 10000000
DECEL AXIS(0) = 10000000

UNITS AXIS(1) = 16 'Achspanparameter Achse 1 setzen
SPEED AXIS(1) = 20000
ACCEL AXIS(1) = 10000000
DECEL AXIS(1) = 10000000

VR(0)= 0 'alle Ausgaenge von USER I/O UNAX loeschen
VR(1)= 0 'alle Inputs zu XENAX 1,2 loeschen
VR(2)= 0 'alle Inputs zu XENAX 3,4 loeschen
VR(3)= 0 '7-Seg Anzeige auf 0 setzen

VR(10) = state_reset ' Status der Applikation

WHILE 1=1 '-----HAUPTLOOP-----
    ON VR(10) GOSUB do_reset, do_homing, do_run, do_stop, do_error
WEND '-----

do_reset:
    VR(1)=34 'Input 2 setzen bei XENAX 1,2 -> Input Function EE ausloesen
    WA(100)
    VR(1)=0 'alle Inputs zu XENAX 1,2 deaktivieren
    WA(1000)

    VR(1)=VR(1) OR $11 'Input 1 setzen bei XENAX 1,2 -> Input Function PGI, Homing
    WA(100)
    VR(1)=VR(1) AND $EE 'Input 1 loeschen bei XENAX 1,2
    VR(3) = 1 '7-Seg Anzeige UNAX auf 1 setzen
    VR(10)=state_homing
RETURN

do_homing:
    IF (IN(8)=ON AND IN(11)=ON) THEN 'Output 1 lesen von XENAX 1,2 Homing fertig ?
    DEFPOS (0,0) 'Homing abgeschlossen, absolute Position in UNAX nullen
    WDOG=ON 'Endstufe UNAX freischalten fuer Puls-Richtungsansteuerung
    BASE(0,1) 'Definition der Achsgruppe fuer Fahrbefehle
    VR(10)=state_run
    ELSE
    IF (IN(9)=ON OR IN(12)=ON) THEN 'Output 2 lesen von XENAX 1,2 Achsfehler ?
    VR(10)=state_error
    ENDIF
    ENDIF
RETURN

do_run:
    IF (IN(9)=ON OR IN(12)=ON) THEN 'Output 2 lesen von XENAX 1,2 Achsfehler ?
    VR(10)=state_error
    ELSE
    IF (IN(0)=ON) THEN 'USER I/O UNAX Eingang 0 (Stoptaste) gesetzt ?
    VR(10)=state_stop
    ELSE
    VR(3) = 2 '7-Seg Anzeige UNAX auf 2 setzen
    IF (IN(1)=ON) THEN 'USER I/O UNAX Eingang 1 (Kurvenform)
    MOVEABS(0,0) 'CAD-Kurve
    MOVE(-5000,0)
    MOVE(0,-8000)
    MOVE(-5000,0)
    MOVE(0,-4000)
    MOVE(5000,-3000)
    MOVE(20000,0)
    MOVE(5000,5000)
    MOVE(0,10000)
    MOVECIRC(-10000,0,-5000,0,0)
    MOVECIRC(-10000,0,-5000,0,1)
    WAIT IDLE
    ELSE
    MOVEABS(0,0) '8-Kurve
    MOVECIRC(0,0,0,10000,0)
    MOVECIRC(0,0,0,-10000,1)
    WAIT IDLE
    ENDIF 'Kurvenform
    ENDIF 'Stoptaste
    ENDIF 'Achsfehler
RETURN

```

Check error state

*exceptional motion controls*

```

do_stop:
  VR(3)=1 '7-Seg Anzeige UNAX auf 1 setzen
  IF (IN(9)=ON OR IN(12)=ON) THEN 'Output 2 lesen von XENAX 1,2 Achsfehler ?
  VR(10)=state_error
  ELSE
    IF (IN(0)=ON) THEN 'USER I/O UNAX Eingang 0, Stoptaste aktiv ?
    VR(10)=state_stop
  ELSE
    VR(10)=state_run
  ENDIF
  ENDF
  RETURN

do_error:
  VR(3)=95 '7-Seg Anzeige UNAX auf "E" setzen
  IF (IN(2)=ON) THEN 'USER I/O UNAX Eingang 2, Resettaste gedruckt ?
  VR(10)=state_reset
  ENDF
  RETURN

```

If the user application is in error state, switch 2 starts the reset sequence and restarts the application

### 6.4 Test the user application

To test the application, set the switches as follows:

switch 0	stop	OFF
switch 1	curve	OFF
switch 2	reset	OFF

Start the application (see step 1, XY-TISCH\_1.bas)

Both LINAX<sup>®</sup> axes execute the HORM sequence in addition to centering their slides.

Afterwards, the orthogonal axis executes a motion path like an “8” repetitively.

UNAX<sup>®</sup> writes “2” into its 7-segment display, the XENAX<sup>®</sup> slave controllers “1”.

Set switch 1 (curve) to ON

If the path „8“ is finished, the motion path changes to DXF imported sequence.

Set switch 0 (stop) to ON

If the DXF motion path is finished, the axes stop operating. The UNAX writes “1” into its 7-segment display.

If an error occurs, the application state changes to ERROR, the UNAX shows „E“ on display.

A signal pulse on switch 2 starts the reset sequence and restarts the application again.

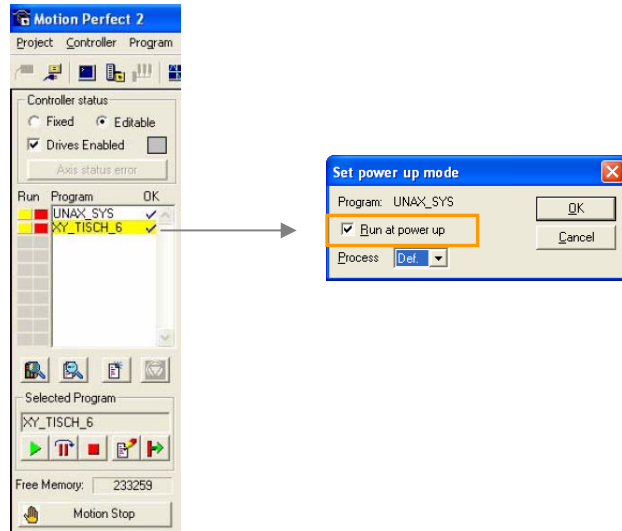
## 6.5 Completion, commissioning

A finished and tested program can be stored nonvolatile into the UNAX<sup>®</sup> flash memory.

With task option *Run at power up*, a task starts automatically after power up the UNAX<sup>®</sup> system.

Set cursor on XY\_TISCH\_6 resp. XY\_TISCH in the program list, press the right mouse button and set the checkbox *Run at power up* in dialog.

After power up the UNAX<sup>®</sup> system, the application starts automatically.



## 7 Communication with UNAX<sup>®</sup> system

### 7.1 Standard

The UNAX<sup>®</sup> system is designed to operate in standalone mode. Synchronisation with host computers can be realised with different mechanisms.

#### Hardware handshake with digital I/O

The USER I/O UNAX interface supports eight digital outputs and up to 8 digital inputs to trigger and control actions on the UNAX<sup>®</sup> system.

Please take care about reservation of input channels in case of larger system configuration. See also chapter 5.6 *User I/O UNAX / Read Inputs*

#### COM 1 (port 0)

Serial interface to internal Euro205x axis coordinator with TRIO commandset and system protocol.

#### COM 2 (port 1)

Serial interface, free programmable by an application. MODBUS RTU protocol optional

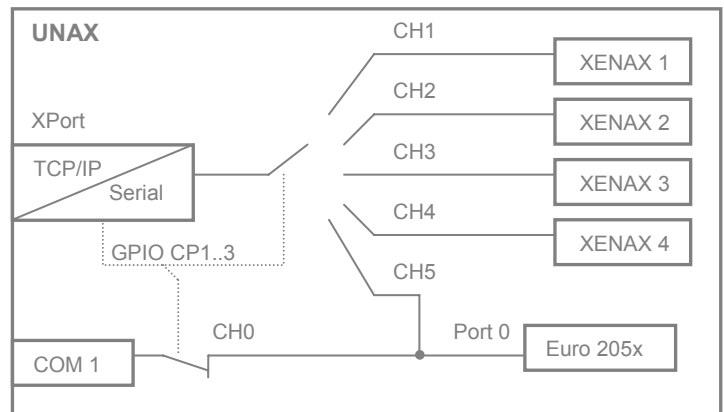
Please consider chapter 5.8 COM 1 / COM 2 Interface for detailed information.

### 7.2 Ethernet TCP/IP Gateway

In general, the TCP/IP interface is used for parameterisation and programming of the XENAX<sup>®</sup> slave servo controllers with the tool WebMotion<sup>®</sup>.

If the host system controls the complete UNAX<sup>®</sup> system over the TCP/IP interface, the host application has to support the channel selection mechanism to communicate with the different devices.

The internal TCP/IP Module XPort is able to select the communication channel to a target device with binary coded outputs CP1..CP3



-CP3 <sup>1</sup>	-CP2	-CP1	CH0	CH1	CH2	CH3	CH4	CH5
0	0	0	1	0	0	0	0	0
0	0	1	1	1	0	0	0	0
0	1	0	1	0	1	0	0	0
0	1	1	1	0	0	1	0	0
1	0	0	1	0	0	0	1	0
1	0	1	0	0	0	0	0	1
1	1	0	1	0	0	0	0	0
1	1	1	1	0	0	0	0	0

<sup>1</sup> The caption -CP3..1 means, outputs CP3..1 must be defined as LOW-active

**Channel CH0**

The connection from UNAX COM 1 interface to the internal axis coordinator Euro205x is always active, except channel 5 is selected.

**Channel CH1 .. CH4**

Communication channel to a XENAX<sup>®</sup> slave servo controller. Active channel is marked with green LED placed by XENAX 1..4 connector.

**Channel CH5**

Communication channel to internal Euro205x axis coordinator. The connection from COM 1 to Euro205x will be interrupted.

**Important**

The serial interface parameters of all involved devices must be identical.  
(XPort, XENAX<sup>®</sup> slave servo controller, Euro205x)

For detailed information about XPort CP1..CP3 activation see the document XPort\_User\_Guide.pdf (page 45).  
The document is located on CD in the folder *UNAX Programmierung / Kanalselektion*.  
A delphi example application UNAXswitch.exe with source code is also included.

## 8 Frequently asked questions (FAQ)

*Are XENAX<sup>®</sup> slave servo controllers used for the UNAX<sup>®</sup> system compatible to standard versions of XENAX<sup>®</sup> servo controllers ? Are this devices exchangeable ?*

Each standard XENAX<sup>®</sup> servo controller includes a XPort TCP/IP module. The XENAX<sup>®</sup> slave servo controller has just a RJ45 connector with proprietary pinout for UNAX<sup>®</sup> communication and control. Therefore, the devices are not compatible and can not be exchanged.

*Can UNAX<sup>®</sup> be mounted displaced from the XENAX<sup>®</sup> slave servo controllers ?*

We recommend strongly to mount the UNAX<sup>®</sup> system according chapter 9.3 *Drilling plan of electrical cabinet*. The cabling is adapted and the electromagnetic compatibility is noncritical.

*Is it possible to connect expansion boards from TRIO to the UNAX<sup>®</sup> device ?*

The hardware connectors for expansion boards are not accessible.

*Why does the servo parameters D\_GAIN, I\_GAIN and so on take no effect on drive behaviour of the axis ?*

The XENAX<sup>®</sup> slave servo controllers operate as stepper axis from UNAX<sup>®</sup>'s point of view. The position loop is located inside of each XENAX<sup>®</sup> device.

The servo parameters D\_GAIN etc. are designated for servo mode with position loop inside of the Euro205x device.

*What is the resulting speed of a multi-axis motion command ?*

In a UNAX<sup>®</sup> application, each axis can be defined with axis parameters like speed, acceleration and so on.

Linear motion with one axis

Speed and acceleration depends on the specific axis parameters.

Multi-axis linear motion, circular motion

The resulting speed and acceleration is taken from the first axis defined in the BASE command.

Example 1:  
 BASE (0,1)  
 MOVE (50000, 30000)  
 resulting speed is defined by axis 0

Example 2:  
 BASE (0,1)  
 MOVECIRC(0,0,0,10000,0)  
 resulting circular speed is defined by axis 0

*How can a host system change variables to control an UNAX<sup>®</sup> application ?*

Through the Euro205x protocol via serial interface COM 1 or the TCP/IP gateway channel 5.

Via serial interface COM 2 with INPUT command, or a customer defined protocol or MODBUS RT protocol.

*Is it possible to teach positions ?*

Go to a defined origin position with an Index

Clear internal position counter of UNAX<sup>®</sup> with DEFPOS(0)

Drive the axis with the FORWARD or MOVE command to a specific position, controlled by inputs for example

Read out internal position counter of UNAX<sup>®</sup>: Teachpos = MPOS

Position values can be stored fixed into the flash memory of UNAX<sup>®</sup> with the command FLASHVR

*The fourth axis does not work properly*

A UNAX<sup>®</sup> standard system does not include four axes. In general, the fourth axis is defined as an external encoder input.

To use the fourth axis in stepper control mode, the axis type must be set explicit into the user application.

```
ATYPE AXIS(3) = 1
```

See also chapter 5.9 External Encoder

## 9 Technical data

### 9.1 Electronics, Firmware

Description	Data
Power supply	24VDC
Current consumption	300mA
Interfaces	Ethernet TCP/IP, integrated http Web Server 2 x RS 232, Development Environment, Application Pulse/Direction interpolator with $f_{max} = 2\text{MHz}$ Master Encoder
Status display	7-Segment LED
Input digital	Up to 8 x 24V Pull-down, free programmable
Output digital	8 x 350mA, 24V open drain, free programmable

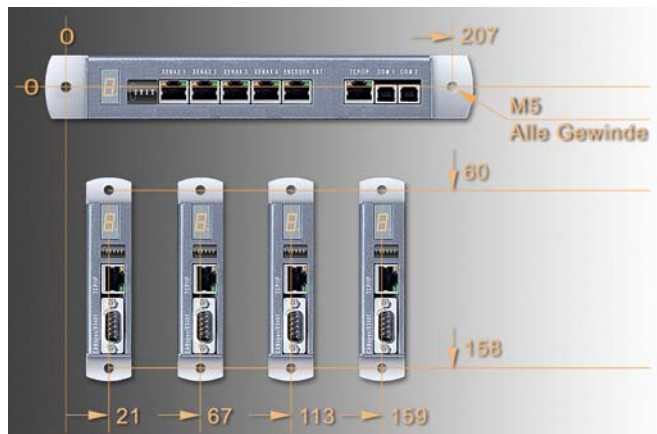
### 9.2 Dimensions UNAX<sup>®</sup> Ux4

Outside dimensions 190 x 140 x 35 mm

### 9.3 Drilling plan of electrical cabinet

Recommended system layout inside of electrical cabinet.

All screw thread M5



## Notes

This instruction manual contains copyright protected information. All rights are reserved.

This document may not be in its entirety or partially copied, duplicated or translated without the prior consent of Jenny Science AG.

Jenny Science AG grants no guarantee on, or will be held responsible for, any incidents resulting from false information.

Information in this instruction manual is subject to change.

Jenny Science AG  
D4 Platz 4  
CH-6039 Root Längenbold, Switzerland

Phone +41 (0) 41 455 44 55  
Fax +41 (0) 41 455 44 50

[www.jennyscience.ch](http://www.jennyscience.ch)  
[info@jennyscience.ch](mailto:info@jennyscience.ch)