

1. The command interface of the Cyclus2

The Cyclus 2 is provided with a serial interface and an Ethernet adapter. The ergometer can be fitted with a WiFi-module as an additional option. The Cyclus2 can be parameterised and controlled via these interfaces using a specified set of commands; a complete data query can be carried out.

After switching on the Cyclus2 or after a reset the serial interface is initialised as follows:

Speed:	4800 Baud
Data bits:	8
Parity:	no
Stop bits:	One
Flow control:	no

The network adapters retrieve their IP address from a DHCP server, which is located on the network. The DHCP-Server setup is to be implemented in such a manner that the Cyclus2 always gets allocated the same IP address. On special request the network adapters can get allocated a static IP address. The Cyclus2 can respond to TCP/IP connections on port 25000. The command set, described in the following, can be fully applied for both the serial interface and the network interface.

The command syntax is based on ASCII-code; the Cyclus2 responds with ASCII-code. The commands have to be concluded with `<CR>` or `<CR><LF>`. The Cyclus2 accepts both termination codes for commands. The replies get the termination code set by the `eol` command. The `<CR>` code is the default setting. Values with decimal places use a point as decimal separator. Multiple values to be sent are separated by a comma. If the Cyclus2 is to be controlled via an external system, e.g. software on a personal computer, the Cyclus2 can be switched to slave mode. Once the Cyclus2 is in the slave mode it can be controlled via the interface only. The device can not be operated manually. The Cyclus2 can be set to the slave mode with the commands `slave`, `ergo` or alternatively on the device in the menu item `System/ Connect`. Please note the instructions referring to the commands described below.

Write commands are concluded with `ok` after successful completion. The required data are retrieved once a query is implemented. In case of an error the response `error: <message>` is sent.

1.1 The command set of Cyclus2

	vers	Query of software version
	Query command vers?	Reply vers: Cyclus2, Version <val>
	Parameter:	<val> Version number in format d.ddd (e.g. 3.100) or from Version 4 in format d.d.d.d (e.g. 4.0.2392.23489)
	Notes:	Version 3.100 Note: Changed in Version 4

	sn	Query of serial number
	Query command sn?	Reply sn:<val>
	Parameter:	<val> Serial number in format dddd-ddddd-ddddd (e.g. 0297-10020-00046)
	Notes:	Version 3.100

	os	Query of version of operating system
	Query command os?	Reply os:<val>
	Parameter:	<val> Versions number in format x.x.x.x, Date+Time (e.g. 5.0.1.16, 30.06.2006 14:51:50,00)
	Notes:	Version 4

	eol	Configuration of the end of line						
	Query command eol?	Reply eol:<val1>,<val2>						
	Configuration command eol=<val1>,<val2>	Reply ok or error:<message>						
	Parameter:	<table border="0"> <tr> <td><val1></td> <td>1. character 13 only is accepted for <CR></td> <td>// unsigned short int</td> </tr> <tr> <td><val2></td> <td>2. character Possible settings are 0 (no second character) and 10 for <LF></td> <td>// unsigned short int</td> </tr> </table>	<val1>	1. character 13 only is accepted for <CR>	// unsigned short int	<val2>	2. character Possible settings are 0 (no second character) and 10 for <LF>	// unsigned short int
<val1>	1. character 13 only is accepted for <CR>	// unsigned short int						
<val2>	2. character Possible settings are 0 (no second character) and 10 for <LF>	// unsigned short int						
	Notes:	Version 3.100						

	br	Configuration of baud rate			
	Query command br?	Reply br:<val>			
	Configuration command br=<val>	Reply ok The response is sent with the so far preset baud rate or error:<message>			
	Parameter:	<table border="0"> <tr> <td><val></td> <td>baud rate Permitted presets are: 1200, 2400, 4800 (default), 9600, 19200, 38400 and 56000. The 56000 setting does not exist any longer as from version 4 and the settings 57600 and 115200 are added instead. The next command must be sent with the new baud rate!</td> <td>// unsigned short int</td> </tr> </table>	<val>	baud rate Permitted presets are: 1200, 2400, 4800 (default), 9600, 19200, 38400 and 56000. The 56000 setting does not exist any longer as from version 4 and the settings 57600 and 115200 are added instead. The next command must be sent with the new baud rate!	// unsigned short int
<val>	baud rate Permitted presets are: 1200, 2400, 4800 (default), 9600, 19200, 38400 and 56000. The 56000 setting does not exist any longer as from version 4 and the settings 57600 and 115200 are added instead. The next command must be sent with the new baud rate!	// unsigned short int			
	Notes:	Version 3.100 Note: Changed in version 4			

	text	Write text to the info bar
	Query command text?	Reply text:<data>
	Configuration command text=<data>	Reply ok or error:<message>
	Parameter:	<data> Text with maximal 63 characters // char [64] The text is displayed in slave mode at the upper margin of the display. The default is „Cyclus2 Interface“. If the Cyclus2 is switched to slave mode with the parameter 2, no info line will be available (see above).
	Notes:	Version 3.100

	data	Query of ergometry data
	Query command data?	Reply data:<val>, <data>
	Configuration command data=<val>	Reply ok or error:<message>
	Parameter:	<val> Data format and data flow // unsigned short int 0 Com Format 1 after query command 1 Com Format 2 after query command 10 Com Format 1 continued 11 Com Format 1 continued. New as from version 4 4 Winsock Format 1 after query command. 6 Winsock Format 1 continued 12 Winsock +Com Format 1 after query command.

		<p>14 Winsock +Com Format 1 continued</p> <p>New as from version 4.2.4155</p> <p>2 Com Format 3 after query command. 3 Com Format 3 continued 5 Winsock Format 3 after query command. 7 Winsock Format 3 continued 13 Winsock +Com Format 3 after query command. 15 Winsock +Com Format 3 continued</p> <p>As from version 4 the format 2 is not supported anymore! Additionally the format 3 with the data for the virtual gear shift is introduced as from version 4.2.4155.</p> <p><data> Format 1 <Time counted from ergometry start in Milliseconds/10>, // unsigned int <Distance counted from ergometry start in Meters>, // float <Crank rotations counted from ergometry start>, // float <Work counted from ergometry start in Joules>, // float <Cadence in rpm>, // float <Heart Rate bpm>, // float <Speed in kms/h>, // float <Transmission in Meters>, // float <Pedal Force in Newtons>, // float <Power in Watts>, // float <Inclination in %>, // float <Work in Heart Rate Beat in Joules> // float</p> <p><data> Format 2 <Time stamp in Milliseconds/10>, // unsigned int <Torque in Newton/1000>, // unsigned short int <Periodic time of cadence (Bit 0..15)>, // unsigned short int < Periodic time of belt pully(Bit 0..15)>, // unsigned short int < Periodic time of cadence (Bit 16..23)>, // unsigned short int < Periodic time of belt pully (Bit 16..23)>, // unsigned short int <Heart rate in bpm>, // unsigned short int <Gear ratio in 1/1000>, // unsigned short int <Load value>, // unsigned short int</p>
--	--	--

		<p><manual break>, // unsigned short int <Stage id>, // unsigned short int <Target value> // float</p> <p><data> Format 3 (the same like format 1 but with additional fields for the virtual gear transmission)</p> <p><Time counted from ergometry start in Milliseconds/10>, // unsigned int <Distance counted from ergometry start in Meters>, // float <Crank rotations counted from ergometry start>, // float <Work counted from ergometry start in Joules>, // float <Cadence in rpm>, // float <Heart Rate bpm>, // float <Speed in kms/h>, // float <Transmission in Meters>, // float <Pedal Force in Newtons>, // float <Power in Watts>, // float <Inclination in %>, // float <Work in Heart Rate Beat in Joules> // float <virtual chain ring>, // unsigned int <virtual rear sprocket> // unsigned int</p>
	Notes:	<p>Version 3.100 Note: Changed in version 4 Note: Changed in version 4.2.4155</p>

	cycle1	Configuration of the mounted bike (new version)
	Query command cycle1?	Reply cycle1:<data>
	Configuration command cycle1=<data> (in slave mode only)	Reply ok or error:<message>
	Parameter:	<data> Parameters of the mounted bike <Version for later extensions, 0 for the structure below> //unsigned short int <Wheel size in Meters>, // float <Crank length in Meters>, // float <Weight in Kilogramms >, // float <Real front chain ring>, // unsigned short int <Real rear sprocket>, // unsigned short int <Virtual front chain ring>, // unsigned short int <Virtual rear sprocket>, // unsigned short int
	Notes:	Version 4.2.4155, new version for <i>Cycle</i>

	cassette	Configuration of the cassette with the available rear sprockets for virtual gear shifting
	Query command cassette?	Reply cassette:<data>
	Configuration command cassette=<data> (in slave mode only)	Reply ok or error:<message>
	Parameter:	<data> preset sprockets separated by commas. If no parameters have been explicitly set, 0 is rendered. In this case stepless gear shift is applied. The number of parameters is variable. The command <i>cycle</i> and <i>cycle1</i> delete the presets! Examples: <i>cassette=0</i> (presets will be deleted, stepless gear shift) <i>cassette= 12,13,14,15,16,17,19,21,23,25</i>
	Notes:	Version 4.2.4155

	rings	Configuration of the front chain rings for virtual gear shifting
	Query command rings?	Reply rings:<data>
	Configuration command rings=<data> (in slave mode only)	Reply ok or error:<message>
	Parameter:	<data> set chain rings separated by comma. If no parameters are explicitly set, 0 will be rendered. In this case stepless gear shift is applied. The number of parameters is variable. The command <i>cycle</i> and <i>cycle1</i> delete the presets! Example: rings=0 (Settings will be deleted, stepless gear shift) rings=39,53
	Notes:	Version 4.2.4155

	cond	Configuration of external field conditions
	Query command cond?	Reply cond:<data>
	Configuration command cond=<data> (in slave mode only)	Reply ok or error:<message>
	Parameter:	<data> field parameter <Air density in kg/m ³ >, // float <Road surface>, // unsigned short int 0 paved road 1 Asphaltic road 2 Cement cycling track 3 Wood cycling track
	Notes:	Version 3.100 Parameters must not be changed during a running training programme.

	heap	Query of available heap size
	Query command heap?	Reply heap:<val>
	Parameter:	<val> available heap size in bytes // float
	Notes:	Version 3.100

	time	Configuration of the local time
	Query command time?	Reply time:<data>
	Configuration command time=<data>	Reply ok or error:<message>
	Parameter:	<data> local time with format dd.mm.yyyy hh:nn:ss
	Notes:	Version 3.100

	ctrl	Control of ergometry
	Query command ctrl?	Reply ctrl:<val>
	Configuration command ctrl=<val> (in slave mode only)	Reply ok or error:<message>
	Parameter:	<val> Status of ergometry // unsigned short int 0 breaks ergometry or resume after pause, while status is "no ergometry" 1 starts ergometry or resume after pause, while ergometry is running 2 pause
	Notes:	Version 3.100

	mpt	Configuration of Maximum Strength Test
	Query command mpt?	Reply mpt:<val>[,<data>]
	Configuration command mpt=<val>,<data> (in slave mode only)	Reply ok or error:<message>
	Parameter:	<p><val> Type of ergometry // unsigned short int 5 Data <data> are available, Maximum Strength Test else no parameter of Maximum Strength Test available</p> <p><data> Parameter of Maximum Strength Test</p> <p><LimitId>, // unsigned short int <LimitValue>, // float <InitialLoad> // float <Len>, // unsigned long <UnitId>, // unsigned short int</p> <p>New as from Version 4</p> <p><StartValue>, // float</p> <p>A description of the individual parameters is shown in chapter 2.4. When writing always set for <val> = 5.</p>
	Notes:	Version 3.100 Parameters must not be changed during a running training programme. Note: Changed in version 4 see chapter 2.4

	mct	Configuration of Maximum Cadence Test
	Query command mct?	Reply mct:<val>[,<data>]
	Configuration command mct=<val>,<data> (in slave mode only)	Reply ok or error:<message>
	Parameter:	<p><val> Type of ergometry // unsigned short int 4 Data <data>are available, Maximum Cadence Test else no parameter of Maximum Cadence Test available</p> <p><data> Parameter of Maximum Cadence Test</p> <p><StartCadence>, // unsigned short int <BasicLoad> // float <Len>, // unsigned long <UnitId>, // unsigned short int</p> <p>A description of the individual parameters is shown in chapter 2.4. When writing always set for <val> = 4.</p>
	Notes:	Version 3.100 Parameters must not be changed during a running training programme.

	want	Configuration of Wingate Anaerobic Test
	Query command want?	Reply want:<val>[,<data>]
	Configuration command want=<val>,<data> (in slave mode only)	Reply ok or error:<message>
	Parameter:	<p><val> Type of training programme // unsigned short int 12 Data <data> are available, Wingate Anaerobic Test else no parameter of Maximum Cadence Test available</p> <p><data> Parameter of Wingate Anaerobic Test</p> <p><Profile >, // int <Factor> // float <Time >, // float <StartCadence>, // float</p> <p>A description of the individual parameters is shown in chapter 2.6. When writing always set for <val> = 12.</p>
	Notes:	Version 4.0 Parameters must not be changed during a running training programme.

	pwc	Configuration of PWC Test (Physical Working Capacity Test)
	Query command pwc?	Reply pwc:<val>[,<data>]
	Configuration command pwc=<val>,<data> (in slave mode only)	Reply ok or error:<message>
	Parameter:	<p><val> Type of ergometry // unsigned short int 13 Data <data> are available, PWC Test else no parameter of PWC Test available</p> <p><data> Parameter of PWC Test</p> <p><Protocol >, // int <Start> // float <Step>, // float <Cadence>, // float <LenType>, // int <Len>, // float <CoolDownLenType>, // int <CoolDownLen>, // float <CoolDownPower>, // float</p> <p>A description of the individual parameters is shown in chapter 2.8. When writing always set for <val> = 13.</p>
	Notes:	Version 4.0 Parameters must not be changed during a running training programme.

	prog	Query of the current type of ergometry
	Query command prog?	Reply prog:<val>
	Parameter:	<val> Type of ergometry // unsigned short int 0: manual control, without control of load duration 1: ergometry with some stages 4: Maximum Cadence Test 5: Maximum Strength Test 8: Source of ergometry is the load generator 9: Conconi Test 10: OBLA Test 11: Slave mode without control of the duration new with version 4 12: Wingate Anaerobic Test 13: PWC Test 14: Real life track
	Notes:	Version 3.100 Note: Changed in version 4

	load	Configuration of load value(the same like the manual control mode)
	Query command load?	Reply load:<CtrlId>,<Val> If no manually controlled ergometry has been set (e.g. during a stage-controlled ergometry), the value 255 is rendered as CtrlId. In this case no further value will be sent for Val.
	Configuration command load=<CtrlId>,<val> (in slave mode only)	Reply ok or error:<message>
	Parameter:	<p><CtrlId> Id of load value // unsigned short int</p> <p>4: Pedal Force in Newtons</p> <p>5: Power in Watts</p> <p>6: Inclination in %</p> <p>255: will be sent on request (see above)</p> <p><Val> Force value in dependency of <i>CtrlId</i> in N, W or % // float</p> <p>Co-domains: Pedal Force 50..1500 Newtons</p> <p> Power 10..3000 Watts</p> <p> Inclination -15..15 %</p> <p>An example in chapter 3.4 demonstrates the application of this command.</p>
	Notes:	Version 4

	calc	Configuration of calculation parameter
	Query command calc?	Reply calc:<interval>, <avg>
	Configuration command calc=<interval>,<avg> (in slave mode only)	Reply ok or error:<message>
	Parameter:	<p><interval> Minimum time in ms/10 (10..1000) // unsigned short int between two data requests exchanged between control device and brake aggregate of the Cyclus2. For standard programmes the minimum time is always set to 500ms <interval>=50. For the Maximum-Force-Test and Maximum-Cadence-Test it is always set to 250ms <interval>=25.</p> <p><avg> Number of data sets (1..20) // unsigned short int For the calculation of the floating mean value during the calculation of the training data on the Cyclus2 For standard programmes the parameter is always set to 10 data sets. For the tests, which respectively determine the Maximal Cadence and the Maximum Force <avg> = 5</p>
	Notes:	Version 3.100 Note: as from version 4 no longer existent

	mon	Configuration of monitoring (cf. check)																					
	Query command mon? Query command mon? <Id>	Reply mon:<Id-Flags> Reply mon:<Id>, <Min>, <Max>																					
	Configuration command mon=<Id>, <Min>, <Max>,<State>,<Band> (in slave mode only)	Reply ok or error:<message>																					
	Parameter:	<table border="0"> <tr> <td data-bbox="864 576 996 608"><Id-Flags></td> <td data-bbox="996 576 1971 639">Flags of available monitoring items</td> <td data-bbox="1971 576 2051 608">// unsigned short int</td> </tr> <tr> <td colspan="3" data-bbox="864 608 2051 639">The output is in hexadecimal format. For better distinction the flag 0x8000 is additionally set.</td> </tr> <tr> <td data-bbox="864 639 996 671"><Id></td> <td data-bbox="996 639 1971 671">Id of monitoring item</td> <td data-bbox="1971 639 2051 671"></td> </tr> <tr> <td data-bbox="864 671 996 703"><Min></td> <td data-bbox="996 671 1971 703">lower limit of training range</td> <td data-bbox="1971 671 2051 703">// float</td> </tr> <tr> <td data-bbox="864 703 996 735"><Max></td> <td data-bbox="996 703 1971 735">upper limit of training range</td> <td data-bbox="1971 703 2051 735">// float</td> </tr> <tr> <td data-bbox="864 735 996 767"><State></td> <td data-bbox="996 735 1971 767">monitoring enabled</td> <td data-bbox="1971 735 2051 767">// int</td> </tr> <tr> <td data-bbox="864 767 996 799"><Band></td> <td data-bbox="996 767 1971 799">with colour band</td> <td data-bbox="1971 767 2051 799">// int</td> </tr> </table> <p data-bbox="864 879 2051 1082">A description of the flags and IDs is shown in chapter 2.1. The unit for the parameters <Min> and <Max> is also documented there. If a monitoring is to be deleted, the parameter <State> must be set to 0. The drawing of the colour strip in the diagram can be initiated by setting <State>. Please note that a colour strip can be drawn only for one parameter.</p>	<Id-Flags>	Flags of available monitoring items	// unsigned short int	The output is in hexadecimal format. For better distinction the flag 0x8000 is additionally set.			<Id>	Id of monitoring item		<Min>	lower limit of training range	// float	<Max>	upper limit of training range	// float	<State>	monitoring enabled	// int	<Band>	with colour band	// int
<Id-Flags>	Flags of available monitoring items	// unsigned short int																					
The output is in hexadecimal format. For better distinction the flag 0x8000 is additionally set.																							
<Id>	Id of monitoring item																						
<Min>	lower limit of training range	// float																					
<Max>	upper limit of training range	// float																					
<State>	monitoring enabled	// int																					
<Band>	with colour band	// int																					
	Notes:	Version 4.0																					

1.2 The commands compatible with the Ergoline Ergometer

The Ergoline800 compatible commands serve as interface for software products, which do not support a protocol for the Cyclus2. In order to remote control the Cyclus2 with these application programmes the Cyclus2 can be initialised as *Ergoline* or *ergoSelect* from within this software. Before hand the Cyclus2 must be set to Ergoline mode to enable the ergometry to be controlled with these commands. In order to do so the user has to select the *System/Connect* menu item and confirm the dialogue box, which is displayed, with *ok*. This mode can also be set via the interface using the *ergo* command (see above).

See example in chapter 3.5. The commands for the inquiry of the blood pressure render only dummies!

Control programmes are not to use Cyclus2 standard commands and Ergoline commands together in one protocol driver for the Cyclus2!

	w	Set the target value during ergometry
	Configuration command w<val> (in slave mode only)	no reply
	Parameter:	<val> Target value (0..2000 Watts) // unsigned short int
	Notes:	Version 2.200

	a	Initialize the target value of ergometry before start
	Configuration command a<val> (in slave mode only)	no reply
	Parameter:	<val> Target value (0..2000 Watts) // unsigned short int
	Notes:	Version 2.200

	l	Set load changes in time interval
	Configuration command l<val> (in slave mode only)	no reply
	Parameter:	<val> Load changes per minute in Watts (0..1000) // unsigned short int Reset after commands <i>w</i> or <i>f</i> .
	Notes:	Version 2.200

	s	Start ergometry
	Configuration command s (in slave mode only)	no reply
	Notes:	Version 2.200

	f	Break ergometry
	Configuration command f (in slave mode only)	no reply
	Notes:	Version 2.200

	b	Query of current power
	Query command b	Reply B<val>
	Parameter:	<val> Current Power in Watts // unsigned short int
	Notes:	Version 2.200

	d	Query of current cadence
	Query command d	Reply n<val>
	Parameter:	<val> Current cadence in rpm // unsigned short int
	Notes:	Version 2.200

	h	Query of current heart rate
	Query command h	Reply H<val>
	Parameter:	<val> Current heart rate in bpm // unsigned short int
	Notes:	Version 2.200

	i	Query of device id
	Query command i	Reply er800P10V243
	Notes:	Version 2.200 Note: Changed in version 4, the reply is equivalent with the Ergoline 800 to SW-Version 2.9

	x	Leave slave mode
	Query command x	No reply
	Notes:	Version 2.200

	o	Query of systole (dummy irrelevant)
	Query command o	Reply O000
	Notes:	Version 2.200

	u	Query of diastole (dummy irrelevant)
	Query command u	Reply U999
	Notes:	Version 2.200

2. Annotations regarding the data structures of the Cyclus2

2.1. General data types, indices and flags

Training parameters: The training parameters, which are analysed on the Cyclus2 are characterised by IDs and flags. The IDs are used e.g. to parameterise analysis functions (e.g. monitoring routines with the *check* command, graphic settings with the *graph* command). This also defines the unit, which determines the scale. The allocation is as follows:

Training value	Id	Flag
Cadence in rpm	0	0x0001
Heart rate in bpm	1	0x0002
Speed in kms/h	2	0x0004
Transmission in Meters	3	0x0008
Pedal force in Newtons	4	0x0010
Power in Watts	5	0x0020
Inclination in %	6	0x0040
Work in heart beat in Joules	7	0x0080
none (for chart)	9	none

Load parameter: Reference values can be defined on the Cyclus2 either for the performance, for the pedal force (torque) or for the inclination. These IDs are used e.g. for the generation of programmes or for the setting of loads (*load*). They furthermore determine the unit for the actual value. The allocation is as follows:

Load value	Id
Pedal force in Newtons	4
Power in Watts	5
Inclination in %	6

Length type: The duration of a load on the Cyclus2 can be set in time-, distance- and labour-units. Furthermore is the x-axis defined by the length type. The units are allocated as follows:

Type of length	Id	Cyclus2 type in format	Basic unit for commands: <i>graph, stage, mct, mpt</i>	Basic unit for command: <i>gen</i>
Time	0	Seconds	s	ms/10
Time	1	Minutes	min	ms/10
Distance	2	Meters	m	m
Distance	3	Kilometers	km	m
Work	4	Kilojoules	kJ	J
Work	5	Joules	J	J
Time	9	hh:mm:ss	n.d.	ms/10

2.2 The data format of load stages

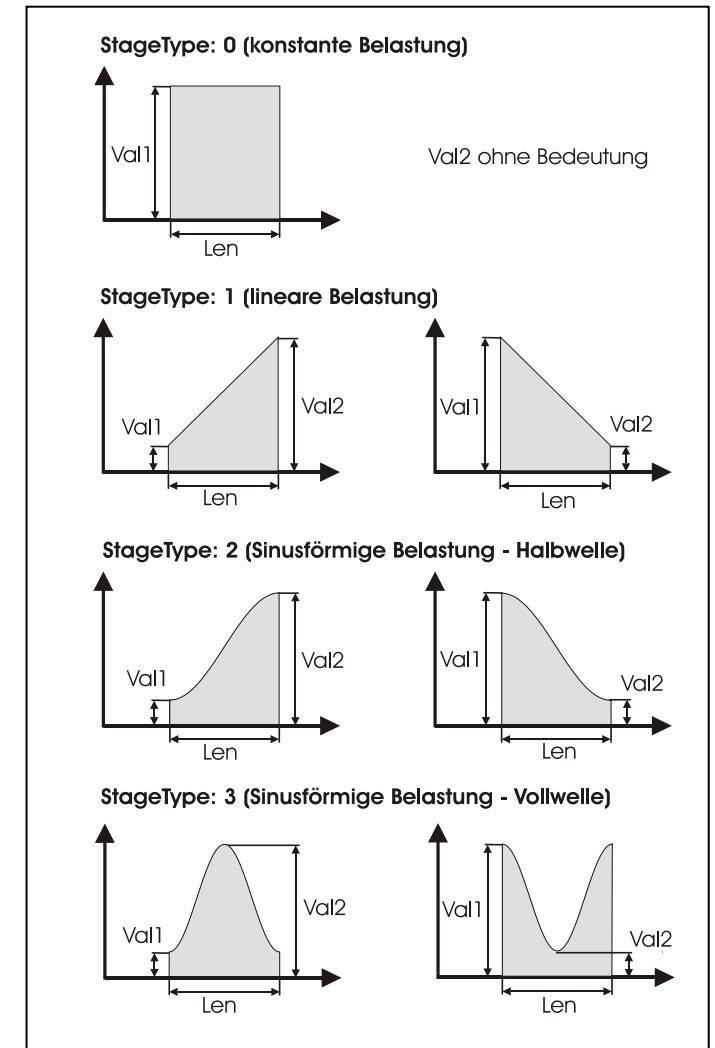
On the Cyclus2 the load programmes are defined by stages. A programme can be assembled using up to 2000 stages. As shown in the figure, 5 different stage types (*StageType*), which, except the type track profile, can be freely combined in a programme, are available.

StageType	Id
Constant load	0
linear load	1
Half wave - sinus load	2
wave – sinus load	3
Real life track (ab Version 4)	4

The parameters *Val1* and *Val2* render the absolute values of the load. The parameter *ControlId* allocates the respective load parameter to the absolute values (see 2.1). When applying track profiles the parameter *Val1* relates to the initial height and *Val2* to the final height.

The parameter *Len* defines the duration of a stage. The unit of the duration is defined by the parameter *UnitId*. The respective units are listed in chapter.2.1.

When using track profiles only *UnitIds*, which define a distance (2 and 3), are permitted.

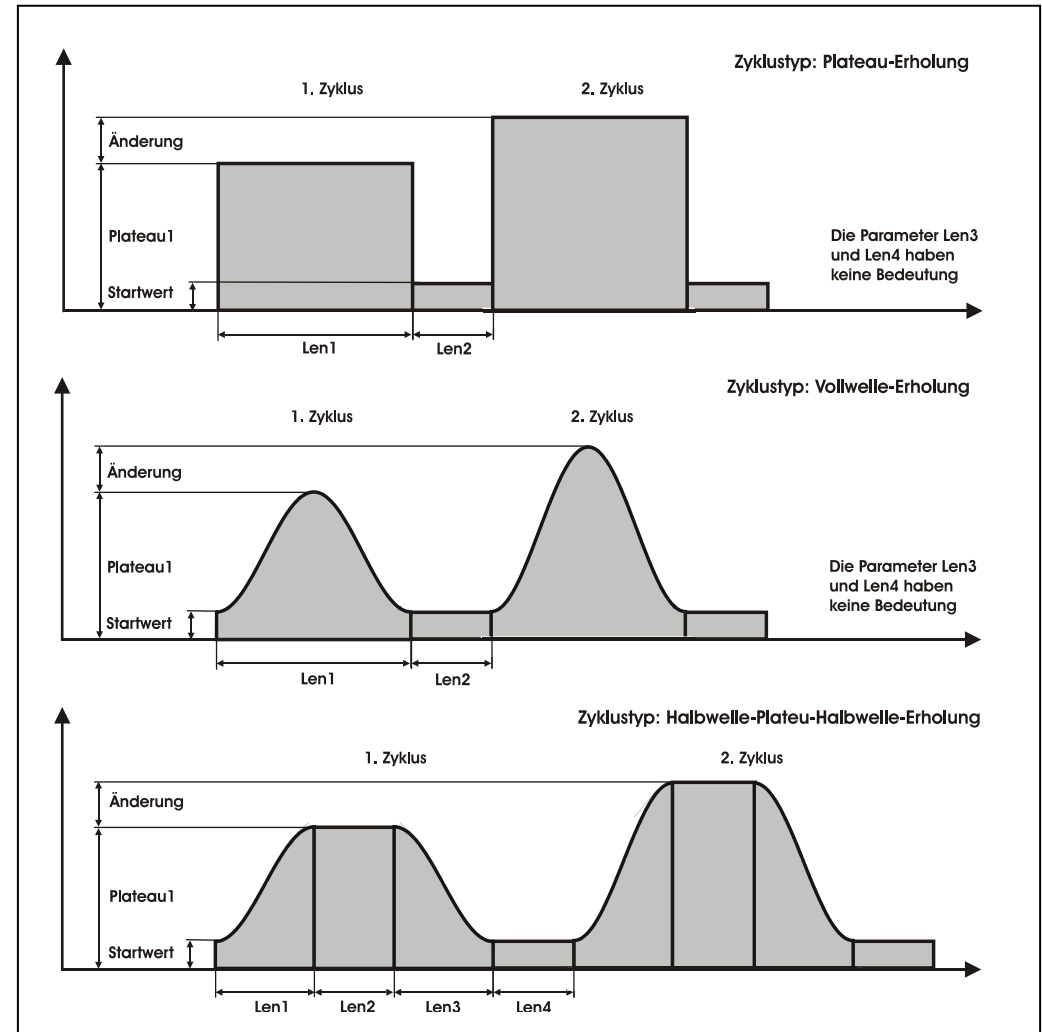


2.3 The data format of the load generator

The Cyclus2 is equipped with a generator, which, when initialised with a few parameters, can generate programmes with typical load patterns. With the parameters a detailed cycle is described, which repeats according to the presets. The Cyclus2 distinguishes between the following types of cycles:

Type of cyclus	Id
Plateau-Cool down	0
Wave-Cool down	1
Half wave-Plateau-Half wave-Cool down	2

In the figure the parameters, which relate to the type of cycle that has been set, are graphically illustrated. The parameters *Len1* to *Len4* relate to the parameter length type in this context. The parameter load value is for the allocation of the parameters *StartValue*, *Plateau1* and *Change*. Please refer to chapter.2.1 for further details.



2.4 The data format of Maximum Strength Test

The Maximum Strength Test is part of the Cyclus2 standard features. It is initialised with the command *mpt* and the following parameters: *LimitId* defines the training parameter and *LimitValue* the absolute value of this training parameter which the maximum strength of the athlete is to be measured with. As for the training parameters for the test it can be selected between cadence and speed. *StartPower* sets the initial load for the test. The duration of the test is set by the parameters *Len* and *UnitId*.

As from version 4 an additional parameter *StartValue* applies. It sets the starting condition for the test and relates to the value defined by *LimitId*.

2.5 The data format of Maximum Cadence Test

On the Cyclus2 the maximal cadence that is achieved by an athlete is determined with the Maximal Cadence Test. The test settings are implemented with the *mct* command. The parameter *StartCadence* (20..200) determines the cadence that the test is automatically started with. *BasicPower* (0..1500) determines the basic load with which the test is to be carried out with. The parameters *Len* and *UnitId* define the duration of the test.

2.6 The data format of Wingate Anaerobic Test

The command *want* initialises the Wingate-Test on the Cyclus2. The parameter *StartCadence* (20..200) determines the cadence that the test is automatically started with. The duration of the test is set by the parameter *Time* (in milliseconds). The load during the test is calculated using the parameters *Profil* und *Factor*. Please note that for the load calculation (*Profil* 0 and 1) the body weight of the athlete is part of the equation. The weight can be set with the command *user* (page 8).

	Profil	Belastung während des Tests (Pedalkraft F in N)
Relative torque	0	$F = \frac{Factor \times G}{l}$ G: Body weight in kg l: Crank length in m
Relative pedal force	1	$F = Factor \times G$ G: Body weight in kg
Absolute pedal force	2	$F = Factor$
kg/kg body weight (same like Monark)	3	
Percent body weight (same like Monark)	4	

2.7 The data structure of the OBLA Test

The command *obla* prompts the settings of the OBLA Test. The parameters are as follows:

<i>Start</i>	Load during the 1. load level
<i>Step</i>	Load that the initial load is increased by from load level to load level
<i>Cadence</i>	Cadence that the test is to be carried out with
<i>LenType</i>	Unit that duration was edited with
<i>Len</i>	Load duration for one level (meters, milliseconds, J)
<i>CoolDownLenType</i>	Unit that the duration of the recreation after the test was edited with
<i>CoolLenLen</i>	Load duration for the recreation (meters, milliseconds, J),
<i>CoolDownPower</i>	Preset of the performance power during the recreation

2.8 The data structure of the PWC-Test (Physical Working Capacity)

The PWC Test is set up with the command *pwc*. The parameter *Protocol* sets the cancelation criterion (0 -> PWC130, 1 -> PWC150, 2 -> PWC170). All further parameters relate to the load levels and have the same reference as in chapter 2.7. ,The data structure of the OBLA Test’.

3. Examples

3.1 Preparation and settings

Command to Cyclus2	Reply from Cyclus2	Comments
vers?	vers:Cyclus2,Version 3.100	Check connectivity, query of software version on the Cyclus2,
sn?	sn:0297-10020-00100	Query of the serial number of the Cyclus2
slave=1	ok	Switch Cyclus2 to slave mode
cycle=2.115,0.172,8.5,1,53,12	ok	Set parameter of the actual bike used (wheel perimeter 2,115m, crank length 172mm, weight 8,5kg, fix ratio 53/12)
user=Jens,Meyer,10,10,75,72.5,0.44,0.715	ok	Set parameters of athlete (Jens Meyer born 10/10/1975, weight 72,5kg, resistance area 0,44m ² , fanning friction factor 0,715)
cond=1.202,1	ok	Preset conditions are cycling on asphalt track with an air density of 1,202kg/m ³
check?	check:8000	No monitoring is set.
check=0,70,80	ok	Set monitoring of cadence. The athlete is to exercise within the range of 70 to 80 1/min.
check?	check:8001	Flag for cadence is set.
check=0,0,0	ok	Delete monitoring of cadence.
check?	check:8000	No monitoring is set.
graph=1,0,20,1,0,250,5,0,300,1	ok	Diagram settings (x-axis 20 minutes, left y-axis heart rate from 0-250 1/min, right y-axis performance from 0-300 W, with auxiliary grid)

3.2 Programme generation and ergometry control

Command to Cyclus2	Reply from Cyclus2	Comments
slave?	slave:0	Cyclus2 is in standard mode
slave=1	ok	Switch Cyclus2 to slave mode
gen=8,3000,0,0,0,100,120,20,1,5,0,10	ok	Sinus programme with 10 hills with a load continuation of 30 seconds each. The first hill has an amplitude of 120 W, for each of the following hills the amplitude will be increased by 20 W. The load in the valley is 100W.
data=10	ok	Setting of a continuous data output once new training data have been retrieved.
	data:10,<data>	Data are continuously being sent without prompt. In this case approx. 2 sets of data per second are sent via the serial port. If the data are to be sent via the LAN, the command 'data=6' must be used.
ctrl=1	ok	Start of ergometry,
ctrl=0	ok	Cancellation of ergometry.
data=0	ok	Stop continuous data output
slave=0	ok	Set Cyclus2 back to standard mode. Cyclus2 can be operated manually.

3.3 Programme initialisation with stages and ergometry control

Command to Cyclus2	Reply from Cyclus2	Comments
slave?	slave:0	Cyclus2 is in standard mode.
slave=1	ok	Switch Cyclus2 to slave mode.
stage?	Stage:30000	No programme loaded on the Cyclus2.
stage=0,30,200,0,0,5,0	ok	Initialise first stage of the programme (30 seconds, constant 200 Watt).
stage=1,20,200,150,1,5,0	ok	Initialise second stage of the programme (the load is decreased from 200 to 150 Watt within 20 seconds).
stage=1,20,150,0,0,5,0	ok	Initialise third stage of the programme (20 seconds, constant 150 Watt)
stage=1,20,100,0,0,5,0	ok	Initialise fourth stage of the programme (20 seconds, constant 100 Watt).
stage=1,30,100,200,2,5,0	ok	Initialise fifth stage of the programme (30 seconds, Sinus from 100 up to 200 Watt).
stage=1,10,200,100,2,5,0	ok	Initialise sixth stage of the programme (10 seconds, Sinus from 200 down to 100 Watt).
stage=2,20,100,300,3,5,0	ok	Initialise sixth stage of the programme (20 seconds, Sinus wave with valley 100 and amplitude 300 Watt), and programme preview update.
stage?	Stage:30007	Programme with seven stages loaded.
data=10	ok	Setting of a continuous data output once new training data have been retrieved.
	data:10,<data>	Data are continuously being sent without prompt. In this case approx. 2 sets of data per second are sent via the serial port. If the data are to be sent via the LAN, the command I 'data=6' must be used.
ctrl=1	ok	Start of ergometry.
save=3	ok	Save ergometry data on USB memory stick or if not present on the network drive.
ctrl=0	ok	Cancellation of ergometry.
data=0	ok	Stop continuous data output.

3.4 Ergometry control from an external system (Example track pattern)

Command to Cyclus2	Reply from Cyclus2	Comments
slave?	slave:0	Cyclus2 is in standard mode.
slave=1	ok	Switch Cyclus2 to slave mode.
ctrl?	ctrl:0	There is no ergometry running.
load=6,0	ok	Preparation of track pattern (Load parameter inclination, initial load 0 %), previously set up ergometry programmes will be deleted.
data=10	ok	Setting of a continuous data output once new training data have been retrieved.
	data:10,<data>	Data are continuously being sent without prompt. In this case approx. 2 sets of data per second are sent via the serial port. If the data are to be sent via the LAN, the command I 'data=6' must be used.
ctrl=1	ok	Start of ergometry.
...		
load=6,0.75	ok	Set new inclination (0.75 % uphill) Load parameter must not be changed!
...		
load=6,-1.25	ok	Set new inclination (-1.25 % downhill) Load parameter must not be changed!
...		
save=3	ok	Save ergometry data on USB memory stick or if not present on the network drive.
ctrl=0	ok	Cancellation of ergometry.
data=0	ok	Stop continuous data output
slave=0	ok	Set Cyclus2 clear for operation

3.5 Ergometry control from an external system with the Ergoline commands

Command to Cyclus2	Reply from Cyclus2	Comments
vers?	vers: Cyclus2, Version 4.0.2895.23809	Connectivity check and version query.
ergo=1	ok	Switch Cyclus2 to Ergolinemode. Any ergometry already running will be cancelled. If the Cyclus2 is in analysis mode, it will be switched to standard operation mode.
text=Cyclus2 im Ergoline-Modus	ok	Text content on the display of the Cyclus2.
graph=1,0,20,1,0,250,5,0,300,1	ok	Diagram settings (x-axis 20 minutes, left y-axis heart rate from 0-250 1/min, right y-axis performance from 0-300 W, with auxiliary grid).
a90		Set initial load 90W.
s		Start ergometry.
b	B090	Enquire actual power (result 90 W).
h	H102	Enquire actual heart rate (result 102 1/min).
d	n081	Enquire actual cadence (result 81 1/min).
w120		New load preset 120W.
...		
save=3	ok	Save ergometry data on USB memory stick or if not present on the network drive.
f		Cancellation of ergometry.
ergo=0	ok	Set Cyclus2 clear for operation

3.6 Control of track simulation

Command to Cyclus2	Reply from Cyclus2	Comments
vers?	vers: Cyclus2, Version 4.2.4155.23809	Connectivity check and version query. Note, some of the commands listed below are available only as from version 4.2.4155.23809!
sn?	sn:0297-10020-00100	Query of the serial number of the Cyclus2.
slave?	slave:0	Cyclus2 is in standard mode (no slave mode).
ctrl?	ctrl:0	There is no ergometry running.
slave=3	ok	Switches the Cyclus2 to slave mode. Only the virtual gear shift can be operated on the Cyclus2.
user1=0,Max,Superman,11.11.1980,1,78.3,1.805,0.44,0.715	ok	Initialisation of the athlete. Note: important as body weight and cwA are factors in the calculation of the load reference preset.
cond=1.202,1	ok	Terms that co-determine the load reference preset.
cycle1=0,2.113,0.1725,8,53,12,53,17	ok	Initialisation of the bike, Note: important as the parameters of the bike are factors in the calculation of the load reference preset..
cassette=12,13,14,15,16,17,19,21,23,25	ok	Setting up of the crown gear of cassette at rear wheel hub for virtual gear shifting. Send the command after <i>cycle1</i> everytime!
rings=39,53	ok	Setting up of front chain rings on crank for virtual gear shifting. Send the command after <i>cycle1</i> everytime!
graph=3,0,25,1,0,250,5,0,500,1	ok	Diagram settings (x-axis 25 kilometers, left y-axis heart rate from 0-250 1/min, right y-axis power from 0-500 W, with auxiliary grid)
stage=0,0.2,152,154,4,6,3	ok	Define 1. stage (existing programme will be deleted) 0,2Km from height 152m to height 154m
stage=1,0.3,154,155,4,6,3	ok	Add 2. stage 0,3km from height 154m to height 155m
stage=1,0.5,155,155,4,6,3	ok	Add 3. stage 0,5km from height 155m to height 155m
stage=1,1.2,155,149,4,6,3	ok	Add 4.stage 1,2km from height 155m to height 149m
stage=1,.....		Add further load stretches.
stage=2,0.5,149,152,4,6,3	ok	Add final stage and draw load preview 0,5km from height 149m to height 152m

data=7	ok	Data are being sent continuously every 500ms in format 3 via the Winsocket interface.
	data:7,0,0,0,0,0,67,0,6.59,0,0,0,0,53,17	Athlete is not cycling and has a heart rate of 67 1/min.
	data:7,0,0,0,0,.....	A new set of data every 500ms.
	...	
ctrl=1	ok	Start ergometry.
	data:7,50,.....	1. data set during the ergometry.
	data:7,102,.....	2. data set during the ergometry.
	...	A new set of data every 500ms.
ctrl=0	ok	Cancel track simulation.
data=5		Stop continuous data output.
slave=0	ok	Set Cyclus2 clear for operation.