isel®
From Components to Systems

Operating Manual

Electronic handwheel RCS07 for CAN-CNC

isel®
www.isel.com

About this manual

In spite of all care, printing errors and mistake cannot be excluded.
We are very grateful for suggestions for improvement and information about errors.

Manufacturer: ***Isel*** Germany AG
Bürgermeister-Ebert-Straße 40
D-36124 Eichenzell

Phone: (06659) 981-0
Fax: (06659) 981-776
Email: automation@isel.com
http://www.isel.com

As of: 10/06/2017

Table of Contents

Introduction 4

General Remark 4
System prerequisites 4
Usage environment 4

Handwheel description 6

Hardware 6
Membrane keyboard 6
Handwheel status LED 7
Handwheel adjuster 7
Enabling switch 7
Emergency off switch 7
Housing 7
Spiral cable with plug 8
Change to the basic settings 8
Software 9
General Remark 9
CANOpen objects 10
Data object of the handwheel adjuster 10
Data object of the key status LEDs and the keys 11

Index 13

Introduction

General Remark

For setup, the machine axes can be moved manually or step by step via the buttons of a keyboard or via the alignment buttons of the machine operating panel. However, a handwheel is simpler and safer. A portable handwheel always keeps the machine operator close to where things happen, enabling him to keep an eye on the setup process and control the supply precisely and accurately.
In addition to the amplify of the IMD-series and the IO modules, the electronic handwheel RCS07 is another component of the CAN-CNC-control of Isel Germany AG. Generally, the handwheel is an IO-module. With its dedicated intelligence, the handwheel can collect information internally and process it. The CAN-bus and the CANOpen standards perform the information exchange with the control core running in the PC. Thanks to the data transfer via the serial bus, the number of electrical connections is limited to the absolute minimum. Therefore, the handwheel is also ideal for re-equipment. Contact us if you wish.
Although the handwheel is really only a CANOpen-IO module, you cannot just use our handwheel in a other control, since many software expansions around the control core are necessary to give this IO module the function of a handwheel. You can use a programming interface in order to integrate the handwheel into a operating interface.
The contents of this manual are also offered as online help. To open the online help, it is better that the online help is on a local hard disk rather than on a network hard drive. If the contents page remains empty when accessing help topics via the table of contents, there is a access problem to the help file. In this case, select the help file with the right mouse button. Selecting "Properties" displays the dialog box "... Properties". The button "Unblock" on the tab "General" should be clicked.

System prerequisites

If you want to use our electronic handwheel, you need to observe the following items.

- The CAN-CNC-control from version 01.60.00.90 onwards with the CAN-PCI- or CAN-PCI-Express-Interface is necessary. The handwheel can only target one axis that is placed on the same CAN-bus line. This means that if the two-channel CAN-interface is used, two hand wheels are needed if you want to move each axis with one handwheel.
- The handwheel currently only works with the end stages IMD20/40 for AC-motors as of firmware version 01.60.00.90. The handwheel cannot be used in combination with the amplify IMD10 and UVE8112 for DC motors or with the CPC12-module for controlling third-party amplify via a ±10V-signal.
- The CAN node number of the hand wheel is in the range of 64 … 79. The value 65 is pre-set at delivery. The pre-set value can be changed. But the device must be opened.
- Various transmission speeds are possible for the handwheel. By default, we work at the highest baud rate of 1MBit/s. The baud rate can only be changed by opening the housing.
- In the handwheel, a CAN-bus terminal resistor is firmly installed. Therefore, the handwheel must be the last bus participant. If necessary, the resistor can be deactivated.

Usage environment

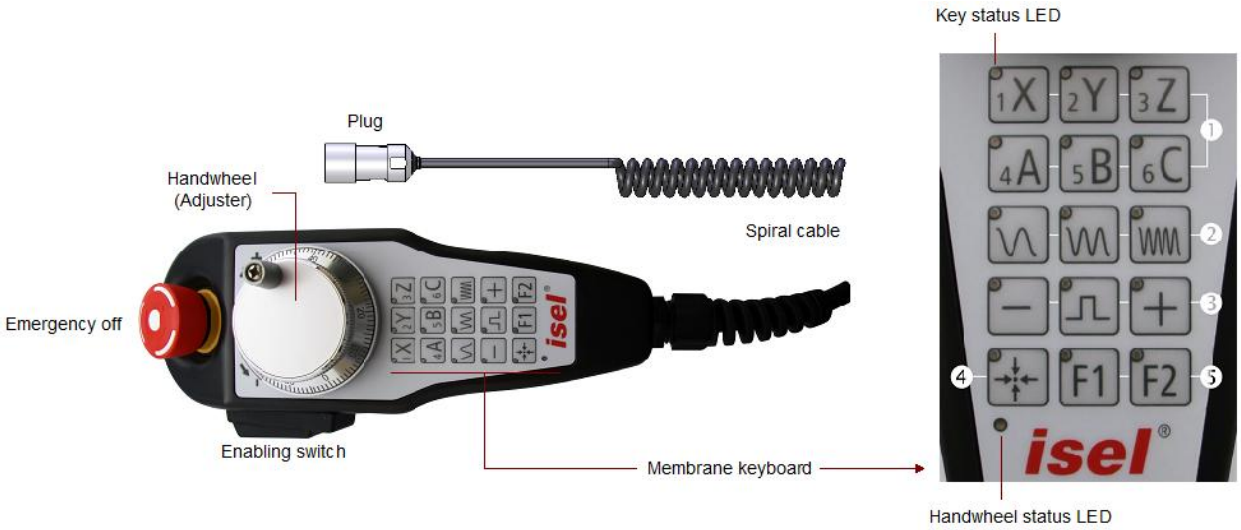
In order to ensure the handwheel's full function, the following ambience conditions must be ensured.

- Operating voltage 24V (+/-10%), polarity reversal protection 60V
- Power intake: approx. 2W

- Temperature range: -30 … 70°C
- Relative humidity: 0 … 95% non-condensing
- Weight: approx. 0.3kg without cable
- Protection type: IP65

Handwheel description

Hardware



Membrane keyboard

The keys are embossed and applied with snap-on discs in order to ensure a good feel. Each of the 15 keys of the dirt-repellent membrane keyboard is equipped with a status-LED for displaying the selection. The LED is on at selection. Otherwise, the LED is off. The operating interface can use the programming interface of the control core to shut down any of the keys or to switch it into one of the two modes.

- Key mode
  The key is only active when it is pushed. The key is a button here.
- Switch mode
  Every time the key is pushed, it switches from active to inactive or vice versa. Status is retained even when the key is released. In this case, the key is a switch.

The printed symbols make the key assignment self-explanatory.

1. The 6 switches allow the selection of an axis to be moved. A gantry axis is only one axis. The master axis is decisive. Each key has a letter and a number as a symbol. This makes it possible to designate a physical axis not only with a letter as usual, but also with a number. Furthermore, a key can be assigned to two axes by switching the two axes, e.g., with one of the freely available function keys F1 and F2. Only one axis must be active at any given time. If several keys are pushed at the same time, the first axis in the order X, Y, Z, ... is chosen. The assignment of a key to a specific axis is done in the operating interface. The available programming interface gives the operating interface absolute freedom here. Due to the printed symbols on the keys, it is, of course, a good idea to assign a key to the axis corresponding to the key symbol.
2. Use these 3 switches to select the feed for the movement (slow, normal, fast). The feeds are software settings in the operating interface. Logically, only one feed at a time can be active. If several keys are pushed at the same time here, the first feed becomes active in the order slow-medium-fast.

3. Instead of working with the handwheel adjuster, you can use the two buttons -/+ to consistently move the axes with the chosen feed. If the middle switch Π is active, the step-by-step movement is performed. The step width is a software-setting in the operating interface and equal to the length that the axis would move with one handwheel adjuster encoder increment. Movement with the buttons +/-, the so-called jog-movement, is always preferred to movement with the handwheel adjuster. If the two buttons +/- are pushed at the same time, only the button - will be evaluated. The axes cannot be moved without an active feed.
4. The button is intended for capture of the current positions of all axes. The positions are internally saved and can be read back. Positions not collected yet are simply overwritten.
5. The two keys F1 and F2 are freely available function keys. An operating interface can shut it down or be used as a button or switch in order to implement further functions.

Handwheel status LED

The handwheel uses this LED to signal its internal operating status outwards.

Not yet initialised or defective

Ready

Active or in use

Handwheel adjuster

The mechanical detent with 100 positions per revolution permits a very precise and accurate control of the movement. The incremental counting of the positions takes place internally in the handwheel. The current position is available as a 16-bit value and can be requested from the control on demand using the CAN bus. Turning the handwheel adjuster moves the axes. The axis speed is the product of the rotating speed and the chosen feed. If no feed switch is active, the axes also cannot be moved.

Enabling switch

The enabling switch is designed to function in 3 stags and 2 channels according to IEC/EN 60947-5-1. In contrast to a two-stage enabling switch, a 3-stage one ensures safety of the machine operator even if he forcefully holds on the enabling switch in emergency. Therefore, a three-stage enabling switch is express required in many C-standards.

Emergency off switch

The emergency-off-switch has 2 channels according to IEC/EN 60947-5-1 and two-pole-opening. It corresponds to the current machinery directive.

Housing

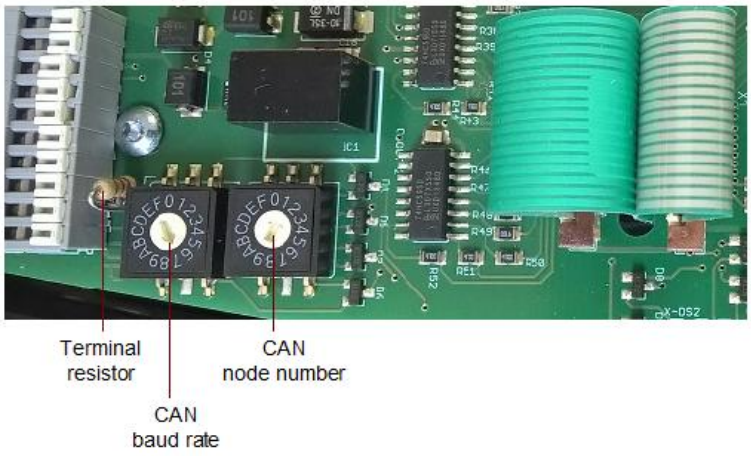
The housing (H223mm x W91mm x D92mm) is made of impact-resilient plastic ABS (UL94V0). On the reverse, there are rubberised adhesive magnets for attachment to metallic surfaces. A wall holder is not necessary or can be implemented with simple sheet strips.

Spiral cable with plug

The shielded cable (18 x 0.14 mm²) has a coil length of 1 m and therefore a stretched length of 4 m. The robust and industry-capable stainless steel plug has the following pin assignment.

Pin	Signal	Cable colour
1	VCC+	White
2	GND	Brown
3	Enabling Switch - Circuit 1 – Contact 1	Green
4	Enabling Switch - Circuit 1 – Contact 2	yellow
5	Enabling Switch - Circuit 2 – Contact 1	gray
6	Enabling Switch - Circuit 2 – Contact 2	Pink
7	Emergency-off-switch - Circuit 1 – Contact 1	Blue
8	Emergency-off-switch - Circuit 1 – Contact 2	Red
9	Emergency-off-switch - Circuit 2 – Contact 1	Black
10	Emergency-off-switch - Circuit 2 – Contact 2	Violet
11	CAN-L	Gray-Pink
12	CAN-H	Red-Blau
13	CAN-GND	White-Green
14	Diagnostic Enable	Brown-Green
15	Diagnostic TX	White - Yollow
16	Diagnostic RX	Yellow-Brown
17	Not connect	White-Gray/Gray-Brown
Housing	Shield	

Change to the basic settings



In the delivery status, the handwheel has the CAN-address 65 and a baud rate of 1MBits/s. The terminal resistor of 120 Ohm is active. This pre-setting can be adjusted to your needs. If you want to make any change here, you need to open the device housing. The address and the baud rate can be set via the rotary switches. The terminal resistor can be cut easily with a wire cutter.

Switch position	Node address
0	64=0x40
1	**65=0x41 (standard)**
2	66=0x42
3	67=0x43
4	68=0x44
5	69=0x45
6	70=0x46
7	71=0x47
8	72=0x48
9	73=0x49
A	74=0x4A
B	75=0x4B
C	76=0x4C
D	77=0x4D
E	78=0x4E
F	79=0x4F

Switch position	Baud rate
0	10kbit/s
1	20kbit/s
2	50kbit/s
3	125kbit/s
4	250kbit/s
5	500kbit/s
6	800kbit/s
7	**1Mbit/s (standard)**
8	unused
9	unused
A	unused
B	unused
C	unused
D	unused
E	unused
F	unused

Regarding the baud rate, you must observe that the entire line length from the CAN-bus is about 25 m at a baud rate of 1 MBits/s. This not only includes the cable length of the handwheel, but also the CAN bus lines between the other components and from the control cabinet to the control PC. For a spiral length of 1 m and a resulting stretched length of 4 m, the cable length of the handwheel actually is up to 7 m. If the entire length of the CAN bus line exceeds 25 m, you must work with a slower transmission rate. Of course, all CAN components, including the hand wheel, must have the same transmission rate. For the connection between the transmission rate and the length of the CAN bus line, see the manual ACSetup.

Software

General Remark

A programming interface is available for the handwheel. This interface can be used to integrate the handwheel into the operating interface. Personalisation of the handwheel, such as

- Assignment of the keys to the physical axes,
- Shut-down of a key or assignment of a function to a key,
- Feed values,
- Use of the function buttons, …

is the task of the corresponding operating interface. For the machine operators, the handwheel is part of the operating interface. In order to optimally adjust the handwheel to the respective needs, one must deal with the operating interface.

A number of items must be observed on the level of the control core CAN-CNC.

- The setting software CANSet can be used to check via the menu "CANSet\Extras\Scan CAN-Bus" whether the handwheel is connected or not. If the handwheel is there, it will return the line "Isel Handwheel - RCS07 - Version. V5.0".
- You can also report use of the handwheel to the control core in CANSet. This is done in the menu "CANSet\Configure CNC Control\Handwheel". You can learn more about this in the CANSet manual. When there is a handwheel, you can configure whether you want to use the handwheel or not. If the handwheel is not there, however, and you set this function to using the handwheel, you will be unable to successfully initialise the control anymore when starting.
- In the menu "CANSet\Configure CNC Control\Coordinates and Axes\Coordinates", use the software limit switches to specify the movement range of the individual axes. Note that the operating interface is able at any time using the programming interface to define its own movement range instead of the software limit switches set here. This means that your settings are then not used by the control core in this case.
- The diagnosis function can query the entire status of all components of the CAN-CNC control, including that of the handwheel, at any time. In the menu "CANSet\Extras\Display Diagnosis-File" you can learn more about what information you will receive both during the runtime and later from the log file for the handwheel.

CANOpen objects

The core of the handwheel is a CANOpen-IO module. Data are exchanged via the CAN-bus and based on the CANOpen standard 401. The handwheel-adjuster, the membrane keyboard with its keys and the status-LEDs are completely managed by the IO module and their current statuses are mapped in the corresponding data objects. The handwheel is operated by reading the data from these data objects and by writing the data into these data objects.

Data object of the handwheel adjuster

Index:	0x6401
Subindex:	None
Data type:	INTERGER16
Access:	reading/writing/PDOMapping-reading

This object contains the handwheel rotating encoder position. For each turn, the rotating encoder supplies 100 impulses with the prefix for the rotating direction. The incremental evaluation of the impulses then results in the signed 16-bit position data for the object. The position is 0 right after activation.

Data object of the key status LEDs and the keys

The two following objects are necessary for this.

LEDs object

Index: 0x2000

Subindex0: 2 - Number of subindexes

Subindex1 LEDs 1 to 8
Data type: UNSIGNED8
Access rights: reading/writing/PDOMapping-writing

Subindex2 LEDs 9 to 15
Data type: UNSIGNED8
Access rights: reading/writing/PDOMapping-writing

Key object

Index: 0x2001

Subindex0: 2 - Number of subindexes

Subindex1 Keys 1 to 8
Data type: UNSIGNED8
Access rights: reading/writing/PDOMapping-reading

Subindex2 Keys 9 to 15
Data type: UNSIGNED8
Access rights: reading/writing/PDOMapping-reading

Every LED is controlled by an output bit of the IO module. For output value 1, the LED is on and for output value 0, the LED is off. The LEDs are controlled externally and are to show whether the keys connected to them are active or not. Note that the handwheel status LED cannot be controlled from the outside. This LED shows the internal status of the handwheel (see section Handwheel status LED on page 7).
Every key is connected to an input bit of the IO-module. The input bit signals whether the corresponding key is actuated or not. While the key is actuated, the bit value is equal to 1. Otherwise, the bit value is equal to 0. From the outside, reading of this object shows which keys are currently pushed.
Assignment of the LEDs to the output bits (Index2000_Subindex1 and Index2000_Subindex2) and the assignment of the keys to the input bits (Index2001_Subindex1 und Index2001_Subindex2) are defined as follows.

Subindex1:

LED	1	2	3	4	5	6	7	8
Key	1	2	3	4	5	6	7	8
Position								
Bit	0	1	2	3	4	5	6	7

Subindex2:

LED	9	10	11	12	13	14	15	
Key	9	10	11	12	13	14	15	
Position								unused!
Bit	0	1	2	3	4	5	6	7

If you want to control the handwheel directly without using the programming interface, you should have a closer look at the electronic data sheet of the handwheel and the eds-file handwheel.eds. You will find all the information you need there.

Index

A

ACSetup 9
amplify
  IMD10 4
  IMD20/40 4
  UVE8112 4
axis
  feed 6, 7, 9
  gantry 6
  master 6
  movement range 10
  software limit switch 10
  speed 7

C

cable
  length 9
  spiral length 8, 9
  stretched length 8, 9
CAN
  address 9
  baud rate 4, 9
  cnc-control 4, 10
  interface 4
  io-module 10
  node number 4
  pci-card 4
  pci-express-card 4
  total line length 9
  two channels interface 4
CANOpen
  data object 10, 11
  io-module 4
  object 10
  PDO-mapping 10, 11
CANSet 10
CPC12-positioning module 4

D

data sheet
  electronic 12
diagnosis *See* handwheel

E

eds-file 12

H

handwheel
  adhesive magnet 7
  adjuster 7, 10
  baud rate 4
  detent 7
  diagnosis 10
  emergency off switch 7
  enabling switch 7
  function key 6, 7, 9
  hardware 6
  hardware adjustment 8
  housing 7
  membrane keyboard 6, 10
  node number 4
  operating voltage 4
  personalisation 9
  plug 8
  programming interface 4, 6, 9, 12
  retrofitting 4
  rotary switch 9
  shut down 10
  software 9
  spiral cable 8
  status *See* LED
  terminal resistor 4, 9
  wall holder 7
hardware *See* handwheel

I

impulse 10
incremental 7, 10
input bit 11

K

key 11
  assignment 9
  key mode 6
  shut down 7, 9
  status *See* LED
  switch mode 6

L

LED
  button status 6
  handwheel status 7, 11
  key status 11
log file 10

M

movement
  constant 7
  none 7
  step by step 7

O

operating interface 4, 6, 7, 9
operating voltage 4
output bit 11

P

plug 8
position capture 7
programming interface *See* handwheel

S

software *See* handwheel