



USBoard-USS4

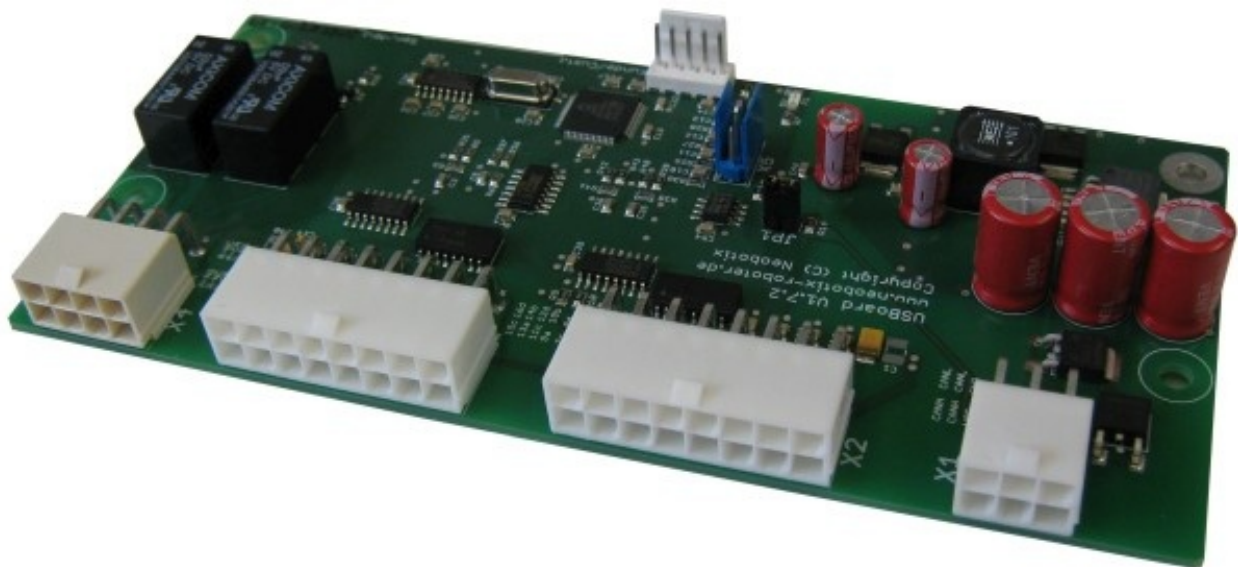
Neobotix GmbH

Nov 21, 2023

1	USBoard-USS4	1
1.1	Introduction	2
1.1.1	Intended Use	2
1.1.2	Improper Use	2
1.1.3	Qualified Personnel	2
1.2	Operating Principle	2
1.3	Technical Data	3
1.3.1	USBoard-USS4	3
1.3.2	Ultrasonic sensor Bosch Parkpilot USS4	3
1.4	Housing	4
1.5	Commissioning	4
1.6	Parameter Set	4
1.7	Command Set	5
1.8	CAN Communication	5
1.8.1	Addresses	5
1.8.2	Commands	6
1.9	RS-232 Communication	9
1.10	ROS Node	9
1.11	Graphical User Interface	10
1.11.1	Introduction	10
1.11.2	First Steps	10
1.11.3	Receiving data	11
1.11.4	Closing the Connection	11
1.11.5	Configuring the USBoard-USS4	11
1.12	Dimensions and Pinout	15
1.12.1	Dimensions	15
1.12.2	Pinout	16
1.13	Additional Parts	19
1.13.1	Connectors	19
1.13.2	Configuration Cable	19
1.13.3	Sensor Cable Sets	20
1.13.4	Ultrasonic Sensor USS4	21
1.13.5	Connector Kits for USS4	22
1.14	Legal Notes	22
1.14.1	EU Declaration of Conformity	22
1.14.2	RoHS Information	23

2	Connectors	24
2.1	TE Connectivity - HE14	24
2.2	Würth Elektronik - MPC4	25
2.3	Würth Elektronik - MPC3	26
3	Qualified Personnel	28
4	Legal Notes	29
4.1	Version Information	29
4.2	Liability	29
4.3	Downloads and Further Information	29

↓ Download as PDF¹



Ultrasonic sensors are used to measure the distance to objects within the sensor's metering range. The metering principle is based on measuring the time of flight of a sound pulse. This pulse is created by the sensor, reflected by an obstacle and then received by the sensor.

The USBoard-USS4 is capable of operating up to 16 ultrasonic sensors of type Bosch Parkpilot USS4 (also known as URF6 or URF7). It can be used for collision detection on big vehicles like buses, farm machines or construction machinery. The board features easy commissioning, comfortable parametrization and a wide range of custom settings. Defining warning and alarm distances allows an easy detection of possible collisions.

Furthermore four analogue inputs are available on the board. These inputs can be used to add other sensors to the system without the need for any additional boards.

¹ <https://neobotix-docs.de/hardware/en/USBoard-USS4.pdf>

Tip: The Graphical User Interface for the USBoard-USS4 can be found at *Graphical User Interface* (page 10).

1.1 Introduction

1.1.1 Intended Use

The USBoard-USS4 was designed for use in mobile robots and similar machines like wheeled drones or AGVs. It is meant for obstacle detection and collision avoidance. The USBoard-USS4 can also be used in stationary machines or equipment to provide distance measurement data.

The USBoard-USS4 was developed to provide exclusively non-safe data and information.

1.1.2 Improper Use

The USBoard-USS4 is explicitly **not a safety device** and must not be used as such. It must never be used to safeguard dangerous areas or as single system for collision avoidance.

The USBoard-USS4 may only be used within the specified *environmental conditions* (page 3). Operating the USBoard-USS4 outside these conditions may result in faulty measurements or a defect and is thus prohibited.

It is not recommended to use the USBoard-USS4 in applications with very high requirements on measurement accuracy, on reliability of the measurements or on data output speed. This for example includes motion control applications like parking.

1.1.3 Qualified Personnel

The USBoard-USS4 must only be installed, connected and brought into service by *qualified personnel* (page 28).

1.2 Operating Principle

An ultrasonic sensor creates a sound pulse that is reflected by obstacles and then returns to the sensor. By measuring the time between creating and receiving the sound pulse, the USBoard-USS4 is able to calculate the distance between the sensor and the closest obstacle.

Activating the sensors cyclically one after the other reduces interfering signals (Cross-Talk). The activation sequence of the sensors is 1, 5, 9, 13, 2, 6, 10, 14, 3, 7, 11, 15, 4, 8, 12, 16.

A CAN and an RS-232 communication interface can be used to acquire distances measured by the sensors as well as the data from the analogue inputs.

For each sensor a warning and an alarm distance can be defined by software. This allows to easily use the board for collision protection. As soon as the distance measured by a sensor falls below the according warning distance, an LED on the board lights up and a relay is switched. A second LED and another relay indicate that an obstacle was detected within the alarm distance of one or more sensors. Warning and alarm distances are defined in the parameter set.

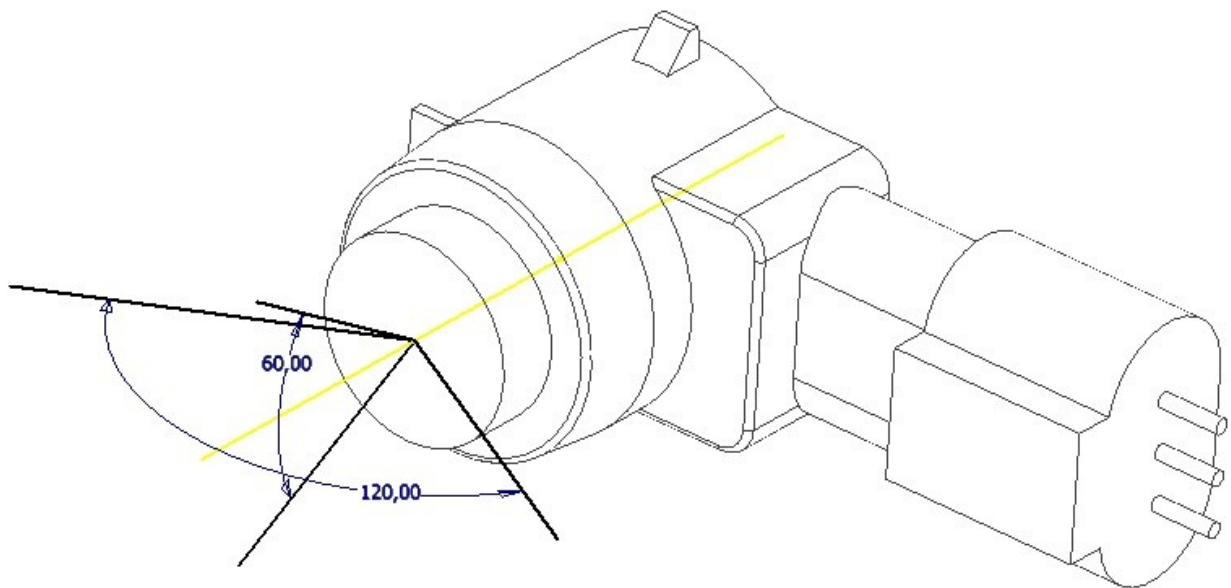
<p>Warning: The USBoard-USS4 is no safety system and can only provide supporting, non-safe information. Never use the USBoard-USS4 for safeguarding of dangerous areas or movements.</p>

1.3 Technical Data

1.3.1 USBoard-USS4

- Supply voltage from +9 VDC up to +60 VDC, max. 5 W
- Digital communication via CAN or RS-232
- Optical indicators for warning and alarm distance
- Relay outputs for warning and alarm distance
- Four analogue inputs (0 V – 5 V)
- Weight: 93 g
- Storage temperature range: -40 °C – 85 °C
- Operation temperature range: 0 °C – 70 °C
- Customs tariff number: 9031 9000
- Order number: X100

1.3.2 Ultrasonic sensor Bosch Parkpilot USS4



- Operating frequency: 42 kHz – 45 kHz
- Range with USBoard-USS4: 0.15 m up to 1.5 m
- Customs tariff number: 9031 8020
- Order number: X200

1.4 Housing

By default the USBoard-USS4 is delivered as OEM component without housing. If required, an appropriate housing can be supplied.

1.5 Commissioning

The USBoard-USS4 is delivered with the default settings to be found at *Parameter Set* (page 4) and is ready for instant use. An individual configuration can be done by Neobotix, if required, or by the customer. The included graphical parameter editor allows programming the board via an RS-232 port. After uploading the parameter set to the board, the USBoard-USS4 immediately uses the new settings and optionally stores them in the non-volatile on-board EEPROM.

The USBoard-USS4 can be used in four different ways:

1. Monitoring the warning and alarm distances with the LEDs on the board.
2. The relay outputs are connected to appropriate indicators and used for monitoring the fields.
3. The measured distances are acquired via the CAN or RS-232 interface and interpreted by an external computer.
4. A combination of the possibilities described above.

1.6 Parameter Set

The USBoard-USS4 uses a parameter set which allows a custom configuration. The graphical parameter editor can be used to easily manipulate the board's settings, store them on the host computer's hard drive, write them to the EEPROM on the USBoard-USS4 or read the current values from the board.

The parameter set consists of 54 bytes. Their meaning is described in the following.

Byte 1 CAN baud rate

0	1000 kBaud (default)
1	500 kBaud
2	250 kBaud
3	125 kBaud
4	100 kBaud
5	50 kBaud

Bytes 2-5 CAN base address (default 0×400), calculated as follows:

$(\text{byte}_5 \ll 24) \mid (\text{byte}_4 \ll 16) \mid (\text{byte}_3 \ll 8) \mid \text{byte}_2$
--

Byte 6 1 if the CAN extended ID is used, 0 if not (default)

Byte 7 Transmission mode for measurement data

0	Send on request (default)
1	Send continuously via CAN
2	Send continuously via RS-232
3	Send continuously via CAN and RS-232

Byte 8 Transmission interval for continuous sending

0	0.5 s
1	1.0 s
2	2.0 s
3	0.2 s

Bytes 9-10 Indicates active sensors (one bit per sensor, 1 = active, 0 = inactive), by default all sensors are active

Bytes 11-26 Warning distance of sensors 1-16 in cm, default is 100 cm

Bytes 27-42 Alarm distance of sensors 1-16 in cm, default is 30 cm

Bytes 43-51 Any value, obsolete

Bytes 52-54 Serial number, read only

1.7 Command Set

The following table shows the available commands of the USBoard-USS4.

Command	Value	Description
CMD_CONNECT	0	Check connection
CMD_SET_CHANNEL_ACTIVE	1	Activate channel for sending / receiving
CMD_GET_DATA_1TO8	2	Request data of sensors 1 to 8
CMD_GET_DATA_9TO16	3	Request data of sensors 9 to 16
CMD_WRITE_PARASET	4	Upload parameter set to board (volatile)
CMD_WRITE_PARASET_TO_EEPROM	5	Write parameter set to board's EEPROM (non-volatile)
CMD_READ_PARASET	6	Read current parameter set
CMD_GET_ANALOGIN	7	Read values of analog inputs
CMD_SET_DEBUG_PARA	8	For debugging only
CMD_GET_DEBUG_PARA	9	For debugging only
CMD_UNKNOWN	10	For debugging only

1.8 CAN Communication

You can find the command set at *Command Set* (page 5).

1.8.1 Addresses

The default base address is 0×400 . This base address can be changed in the parameter set.

The addresses used by the USBoard-USS4 are calculated from the base address by adding the following offsets.

Offset to base address	Message
+0	Receive commands
+1	Answer to CMD_CONNECT
+2	First answer to CMD_GET_DATA_1TO8
+3	Second answer to CMD_GET_DATA_1TO8
+4	First answer to CMD_GET_DATA_9TO16
+5	Second answer to CMD_GET_DATA_9TO16
+6	Answer to CMD_READ_PARASET
+7	Answer to CMD_GET_ANALOGIN
+8	Answer to CMD_WRITE_PARASET
+9	Answer to CMD_WRITE_PARASET_TO_EEPROM

1.8.2 Commands

In the following, the CAN IDs are given as an offset to the base address, so +3 means base address plus 3.

1.8.2.1 CMD_CONNECT

Use this command to establish and check the connection to the board.

Command ID: +0

CMD_CONNECT	0	0	0	0	0	0	0
-------------	---	---	---	---	---	---	---

Answer ID: +1

CMD_CONNECT	1	2	3	4	5	6	7
-------------	---	---	---	---	---	---	---

1.8.2.2 CMD_SET_CHANNEL_ACTIVE

This command can be used to activate or deactivate individual sensors without transmitting the complete parameter set. Two bytes for the channels 1 to 8 and 9 to 16 contain the information whether the sensors should be active. These bytes are bit-coded, each bit representing the state of one sensor. Every active channel is marked with a 1.

For example, $0 \times 1F$ as the first byte means channels 1 to 5 are active and channels 6 to 8 are not active.

Command ID: +0

CMD_SET_CHANNEL_ACTIVE	(Sensors 1 to 8)	(Sensors 9 to 16)	0	0	0	0	0
------------------------	------------------	-------------------	---	---	---	---	---

Answer No answer.

1.8.2.3 CMD_GET_DATA_1TO8

This command is used to request the readings of sensors 1 to 8.

Command ID: +0

CMD_GET_DATA_1TO8	0	0	0	0	0	0	0
-------------------	---	---	---	---	---	---	---

Answer (two parts) IDs: +2, +3

CMD_GET_DATA_1TO8	0	Values for sensors 1-4 (one byte each)	0	(reserved)
CMD_GET_DATA_1TO8	1	Values for sensors 5-8 (one byte each)	0	(reserved)

The measurement values are given in cm.

1.8.2.4 CMD_GET_DATA_9TO16

This command is used to request the readings of sensors 9 to 16.

Command ID: +0

CMD_GET_DATA_9TO16	0	0	0	0	0	0	0
--------------------	---	---	---	---	---	---	---

Answer (two parts) IDs: +4, +5

CMD_GET_DATA_9TO16	0	Values for sensors 9-12 (one byte each)	0	(reserved)
CMD_GET_DATA_9TO16	1	Values for sensors 13-16 (one byte each)	0	(reserved)

The measurement values are given in cm.

1.8.2.5 CMD_WRITE_PARASET

Use this command to transfer a complete parameter set to the USBoard-USS4. The parameters are stored volatile, which means that they will be lost when the board is switched off. To comfortably configure the board, use the graphical parameter editor.

After transmitting the parameter set, it will immediately be used by the board.

Command ID: +0

The command consists of nine messages sent one after the other, each containing some bytes of the parameter set.

CMD_WRITE_PARASET	0	Bytes 1 to 6
CMD_WRITE_PARASET	1	Bytes 7 to 12
CMD_WRITE_PARASET	2	Bytes 13 to 18
CMD_WRITE_PARASET	3	Bytes 19 to 24
CMD_WRITE_PARASET	4	Bytes 25 to 30
CMD_WRITE_PARASET	5	Bytes 31 to 36
CMD_WRITE_PARASET	6	Bytes 37 to 42
CMD_WRITE_PARASET	7	Bytes 43 to 48
CMD_WRITE_PARASET	8	Bytes 49 to 54

Answer ID: +8

Each message is answered individually. The first 8 answers are identical

CMD_WRITE_PARASET	0	0	0	0	0	0	0
-------------------	---	---	---	---	---	---	---

and the last answer contains the sum of all bytes in the parameter set.

CMD_WRITE_PARASET	low byte	high byte	0	0	0	0	0
-------------------	----------	-----------	---	---	---	---	---

1.8.2.6 CMD_WRITE_PARASET_TO_EEPROM

Use this command to write a complete parameter set into the board's EEPROM. The parameters are stored non-volatile, which means that they will be used again, the next time the USBoard-USS4 is switched on. To conveniently configure the board, use the graphical parameter editor.

The parameter set will immediately be used after transmission.

Command ID: +0

The command works identically to *CMD_WRITE_PARASET* (page 7), the only difference is the use of the command byte `CMD_WRITE_PARASET_TO_EEPROM`.

Answer ID: +9

The answer is identical to the answers to *CMD_WRITE_PARASET* (page 7), the only difference is the use of the command byte `CMD_WRITE_PARASET_TO_EEPROM`.

1.8.2.7 CMD_READ_PARASET

Use this command to read a complete parameter set from the USBoard-USS4.

Command ID: +0

CMD_READ_PARASET	0	0	0	0	0	0	0
------------------	---	---	---	---	---	---	---

Answer ID: +6

The answer consists of nine messages sent one after the other, each containing some bytes of the parameter set.

CMD_READ_PARASET	0	Bytes 1 to 6
CMD_READ_PARASET	1	Bytes 7 to 12
CMD_READ_PARASET	2	Bytes 13 to 18
CMD_READ_PARASET	3	Bytes 19 to 24
CMD_READ_PARASET	4	Bytes 25 to 30
CMD_READ_PARASET	5	Bytes 31 to 36
CMD_READ_PARASET	6	Bytes 37 to 42
CMD_READ_PARASET	7	Bytes 43 to 48
CMD_READ_PARASET	8	Bytes 49 to 54

1.8.2.8 CMD_GET_ANALOGIN

Use this command to acquire the data of the four analog inputs.

Command ID: +0

CMD_GET_ANALOGIN	0	0	0	0	0	0	0
------------------	---	---	---	---	---	---	---

Answer ID: +7

Because the resolution of the on-board AD-converter is 12 bit, the first part of the answer is made up of the four low bytes. Bytes 5 and 6 contain the upper 4 bits of the four channels.

CMD_GET_ANALOG	low byte channel 1	low byte channel 2	low byte channel 3	low byte channel 4	high bits of channel 2 and 1	high bits of channel 4 and 3	0
----------------	--------------------	--------------------	--------------------	--------------------	------------------------------	------------------------------	---

1.9 RS-232 Communication

Note: The RS-232 interface requires a common ground connection between the USBoard-USS4 and the computer.

The RS-232-interface runs at 19200 Baud. Communication via RS-232 uses the same format as the *CAN Communication* (page 5) except for the following addition:

Every message **from** the USBoard-USS4 begins with a start byte which must have the value 0xFF. After each message, made up of eight data bytes, a 16 bit checksum is sent. Because of this a complete message is 11 bytes long.

Byte 1	Bytes 2-9	Byte 10	Byte 11
0xFF	data bytes	checksum high byte	checksum low byte

The checksum is calculated with an CRC-CCITT algorithm using 8 databytes.

Messages **to** the USBoard-USS4 only contain the data bytes.

Example implementation of the checksum calculation (C code):

```

unsigned int getChecksum(unsigned char *c, size_t iNumBytes){
    unsigned int uCrc16;
    unsigned char ucData[2];
    size_t i;

    uCrc16 = 0;
    ucData[0] = 0;
    for(i=0; i<iNumBytes; i++){
        ucData[1] = ucData[0];
        ucData[0] = c[i];

        if(uCrc16 & 0x8000){
            uCrc16 = (uCrc16 & 0x7fff) << 1;
            uCrc16 ^= 0x1021;
        }else{
            uCrc16 <<= 1;
        }
        uCrc16 ^= (unsigned int) ucData[0] | ((unsigned int) ucData[1]) << 8;
    }

    return uCrc16;
}

```

1.10 ROS Node

You can find our ROS Node for the USBoard-USS4 on GitHub: https://github.com/neobotix/neo_usboard

1.11 Graphical User Interface

Download the GUI here: [32bit²](#), [64bit³](#)

Note: Please note that the 64bit version is no longer usable with the latest JRE version!

1.11.1 Introduction

The USBoard-USS4 is provided together with the *USBoard-USS4 configuration GUI*, a graphical parameter editor.

The USBoard-USS4-GUI requires a PC with:

- RS-232 interface (COM port) and
- Java® Runtime Environment 1.5 or later.

The connection between serial interface and USBoard-USS4 is established by a special cable which can be obtained from Neobotix or self-constructed using *the specifications* (page 15). This cable also connects the USBoard-USS4 with an *appropriate power supply* (page 3). A working supply is indicated by a green LED on the board.

To start the application, simply double-click on the `start.bat` file. Make sure to use the program version that matches your operating system.

The GUI can also be used on machines with Linux or Mac OS. This requires that the Java runtime environment can find the correct RXTX library. It is also necessary that the user has the permissions needed to access the serial interface. When using Ubuntu Linux please install the package `librxtx-java` and add the user to group `dialout`.

1.11.2 First Steps

1.11.2.1 Establishing a connection

Before starting a connection via the USBoard-USS4-GUI, please make sure that the USBoard-USS4 is connected

- to the PC's RS-232 interface and
- to a power supply (LED lights up).

Now select the correct communication port in the application and click the `Connect` button, to create the connection.

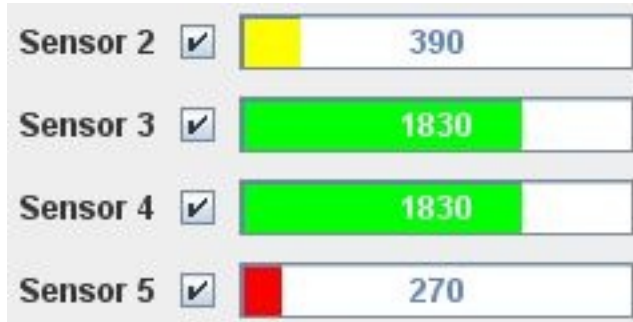


² https://neobotix-docs.de/files/USBoardGUI_v5802_32Bit.zip

³ https://neobotix-docs.de/files/USBoardGUI_v5802_64Bit.zip

1.11.3 Receiving data

After establishing a connection, first data will show up. Depending on currently set parameters, measured data are updated automatically in defined interval or can be read manually by clicking the Request buttons. If the measured distance of a sensor falls below its specified warning distance, the value is displayed yellow. In case of under-running the alarm distance, the colour changes to red. As long as any measured distance is below the according warning or alarm distance, the according LED on the USBoard-USS4 will light up and the relay is activated.



The measured values of the analogue inputs (voltage range 0V – 5V) are displayed as percentages below the distances.

1.11.4 Closing the Connection

To close the connection to the USBoard-USS4, just click the button `Disconnect`.

1.11.5 Configuring the USBoard-USS4

1.11.5.1 Communication Settings

In USBoard-USS4-GUI's left column several parameters regarding the communication settings of the USBoard-USS4 can be defined. The CAN base ID and transfer rate for communication via the CAN bus can be set in the `CAN` panel. The general transmission behaviour can be set in the `Transmission Mode` panel.

CAN

Extended ID:

Baudrate: ▼

Transmission Mode

Send on:

▼

Interval: ▼

The following table explains the different transmission modes. The interval parameter is only relevant for continuous transmission.

Transmission mode	Behaviour
Request	Send only on request
CAN cont.	Send continuously via CAN bus
RS232 cont.	Send continuously via RS-232 interface
RS232 + CAN cont.	Send continuously via both interfaces

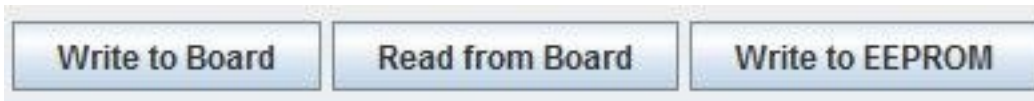
1.11.5.2 Configuring the Sensors

Various parameters of the ultrasonic measurement can be set in the right part of the GUI. Each sensor can be activated separately by marking the check-box beside its name. Deactivated sensors send neither ultrasonic impulses nor data. The warning and alarm distance of each sensor can be set in the edit fields to the right of the measurement indicator. The ID and analogue inputs of the USBoard-USS4 are not configurable.

Warning: The USBoard-USS4 is – despite warning and alarm functions – no safety system and can only provide supporting, non-safe information. Never use the USBoard-USS4 for safeguarding of dangerous areas or movements.

1.11.5.3 Transferring the Configuration

Changed configuration parameters have to be transmitted to the USBoard-USS4 to come into operation. The USBoard-USS4-GUI offers buttons for two available options:



- `Write to Board` transfers the parameters to the volatile memory of the USBoard-USS4. After the next restart, the old parameters from the non-volatile memory will be used again.
- `Write to EEPROM` writes the parameters into the non-volatile memory of the USBoard-USS4, so these parameters will be used until they are overwritten by using the same command.

To load the current parameter set from the USBoard-USS4 and display it in the -USS4-GUI, click the `Read from Board` button. Please note that all previous changes in the input mask will be lost.

1.11.5.4 Saving and Loading Configurations

The buttons `Save to File` and `Load from File` permit the user to save the current configuration into a file on the local computer or load it from one. Simply enter the file name or select an existing file in the dialogue that appears after clicking either button and confirm by clicking `OK`.

1.11.5.5 Logging

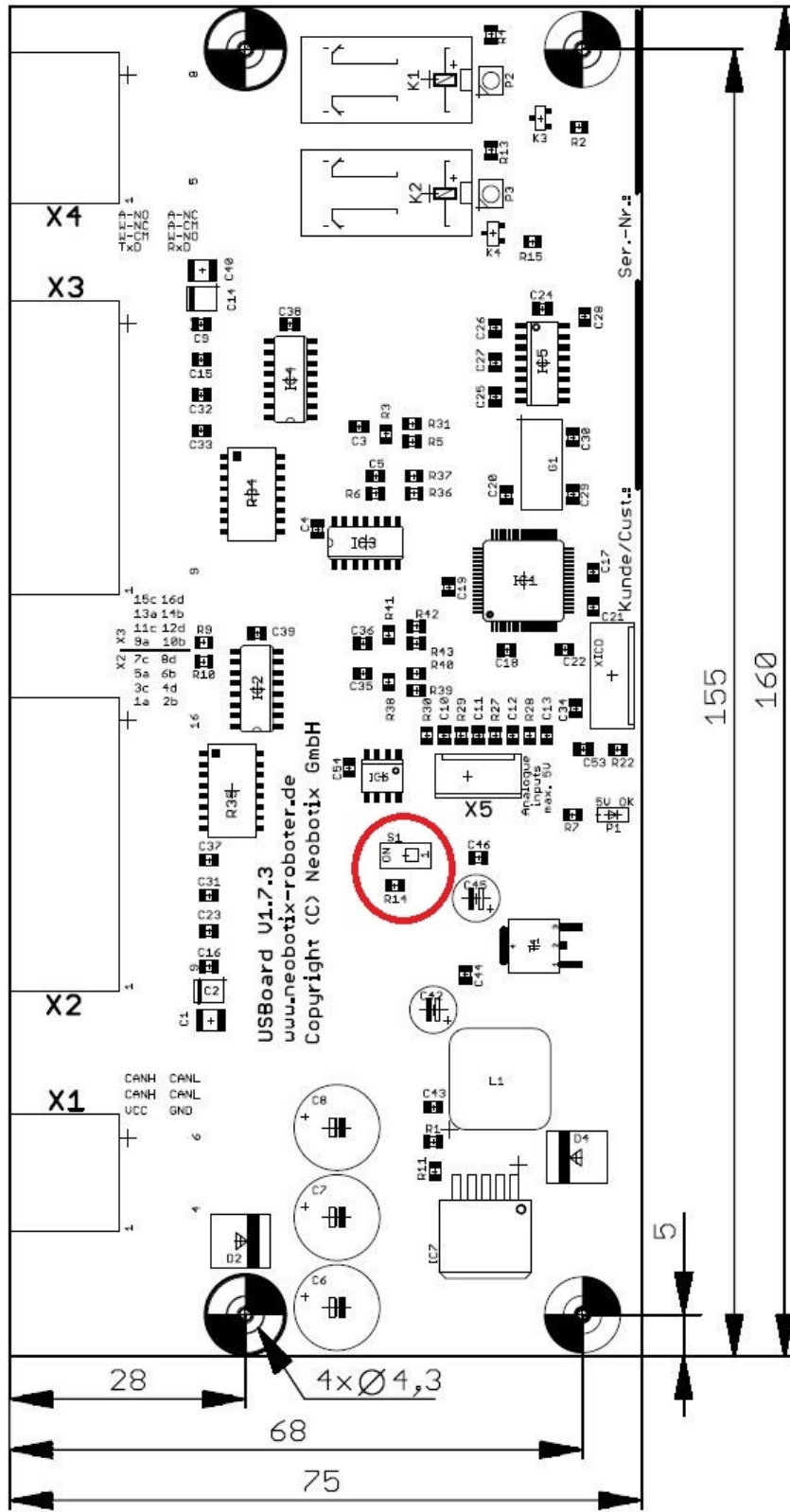
The USBoard-USS4-GUI writes log files for debugging purposes. The information that is to be logged while the GUI is running can be chosen from several different levels between `Severe` (critical problems only) and `Finest` (every action). The log files are located in the folder `log` in the same directory that the GUI is in.

A real time display of the currently active log can be opened with the button `Show / Hide`.



1.12 Dimensions and Pinout

1.12.1 Dimensions



Move S1 to position ON to activate the CAN terminating resistor.

The mounting hole at the bottom right corner is electrically connected to the ground plane of the USBoard-USS4. Please insulate this mounting point if necessary in your installation.

The USBoard-USS4 can be mounted both horizontally and on walls. Please note that when mounted upside down the relays might malfunction. The data output is not affected by the mounting orientation.

1.12.2 Pinout

Tip: More information about these connectors can be found at [Connectors](#) (page 24).

1.12.2.1 Connector X1 – WE MPC4, 6-pin

Pin	Description
1	Power supply input
2	CAN High
3	CAN High
4	Ground (for both power supply and RS-232)
5	CAN Low
6	CAN Low

Pins 2 and 3 are bridged, as well as pins 5 and 6. This allows two CAN cables (CAN-in and CAN-out) to be connected easily.

1.12.2.2 Connector X2 – WE MPC4, 16-pin

Pin	Description
1-4	Sensor power supply 0 V
5	Sensor 1
6	Sensor 3
7	Sensor 5
8	Sensor 7
9-12	Sensor power supply +8 V
13	Sensor 2
14	Sensor 4
15	Sensor 6
16	Sensor 8

1.12.2.3 Connector X3 – WE MPC4, 16-pin

Pin	Description
1	Sensor 9
2	Sensor 11
3	Sensor 13
4	Sensor 15
5-8	Sensor power supply 0 V
9	Sensor 10
10	Sensor 12
11	Sensor 14
12	Sensor 16
13-16	Sensor power supply +8 V

1.12.2.4 Connector X4 – WE MPC4, 8-pin

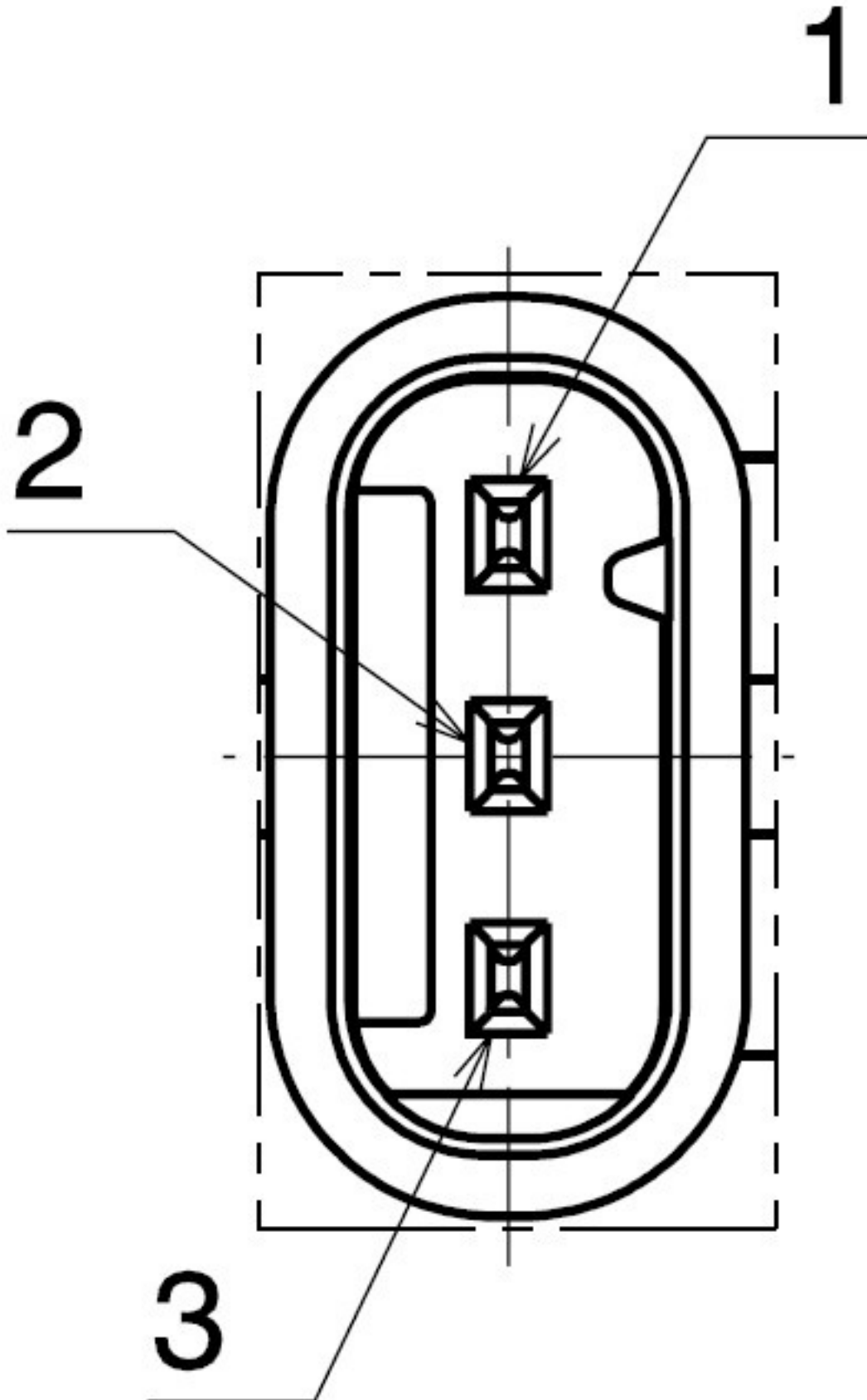
Pin	Description
1	RS-232 transmit
2	Relay “warning distance”, common contact
3	Relay “warning distance”, normally closed contact
4	Relay “alarm distance”, normally open contact
5	RS-232 receive
6	Relay “warning distance”, normally open contact
7	Relay “alarm distance”, common contact
8	Relay “alarm distance”, normally closed contact

1.12.2.5 Analogue input connector – TE Connectivity HE14, 4-pins

Pin	Description
1	Analogue input 1
2	Analogue input 2
3	Analogue input 3
4	Analogue input 4

1.12.2.6 Sensors – Housing: TE 1-1718644-1, Contacts: TE 1452668-1

Pin	Description
1	Send / receive
2	Sensor power supply 0 V
3	Sensor power supply +8 V



1.13 Additional Parts

1.13.1 Connectors

An overview of the connectors can be found [here](#) (page 24).

1.13.2 Configuration Cable

The configuration cable (order no. X201) provides a quick way to bring the USBoard-USS4 into operation. It can also be manufactured by the customer with the connectors described above. In that case please mind the ground connection of the RS-232 interface.

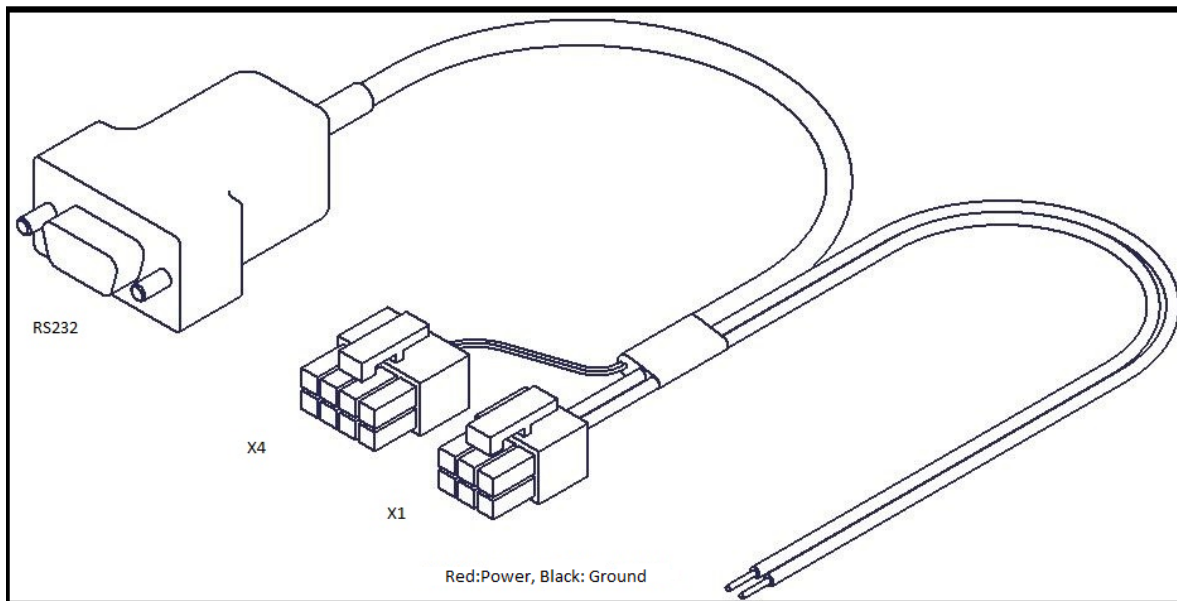


Fig. 1: USBoard-USS4 configuration cable (X201)

1.13.3 Sensor Cable Sets

Cable sets to connect the ultrasonic sensors to the USBoard-USS4 can be provided in small quantities for first tests. These cables are available for either connector X2 (order no. X204) or connector X3 (order no. X205). Both versions are 2 m long and shielded.

For your actual application please manufacture the cables according to the individual requirements and specifications.



Fig. 2: Cable set for USS4 sensors (X204/X205)

Small quantities of the connectors for the USBoard-USS4 (order no. X202) and for individual ultrasonic sensors (order no. X203) are available directly from Neobotix. Each set contains the connector housings and the matching number of crimp contacts.

In case you do not have the proper tools and also do not want to use a universal crimping tool, please contact Neobotix.

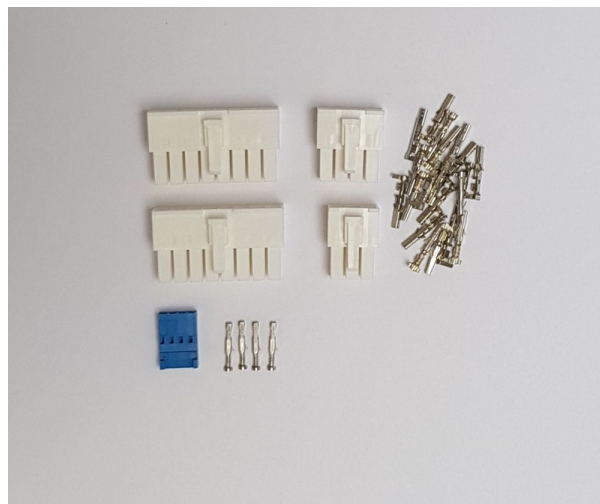


Fig. 3: Connector set for USBoard-USS4 (X202)

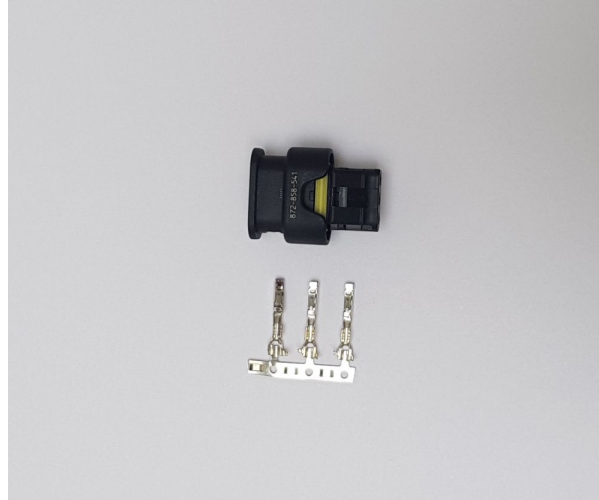


Fig. 4: Connector set for USS4 (X203)

1.13.4 Ultrasonic Sensor USS4

The ultrasonic sensors Bosch ParkPilot USS4 (Bosch part number 0 263 009 525) are available with order number X200.

Note: The bare sensors are called USS (Ultra Sonic Sensor), fourth generation. The fifth sensor generation is supported by our USBoard-USS5.

Bosch offers the URF system for private customers. It consists of eight sensors with a cable set and a small evaluation unit for mounting on the dashboard of cars. This system only uses a few LEDs and an acoustic signal to indicate that you are driving close to an obstacle. It does not offer data output. The URF is available in two versions (Gen. 6 and 7), which only differ in the firmware and both use the USS4.



Fig. 5: Ultrasonic sensor Bosch USS4 (X200)

1.13.5 Connector Kits for USS4

The *mounting set for USS4, straight* (X206) is used to mount the sensor on a front panel.

For an optimum sensor position, the *mounting set for USS4, 0°, 5°, 10°, 15°* (X207) includes four front rings for different mounting angles.



Fig. 6: Mounting kit for USS4, straight (X206)



Fig. 7: Mounting kit for USS4, 0°, 5°, 10°, 15° (X207)

1.14 Legal Notes

The general legal notes can be found at [Legal Notes](#) (page 29).

1.14.1 EU Declaration of Conformity



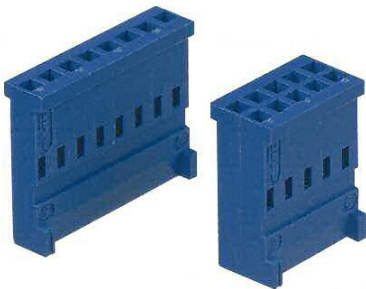
This product fulfils all relevant directives of the European Union.

1.14.2 RoHS Information



This product complies to the RoHS directives 2011/65/EU (RoHS 2) and 2015/863/EU of the European Parliament and the Council on the restriction of the use of hazardous substances in electrical and electronic equipment.

2.1 TE Connectivity - HE14

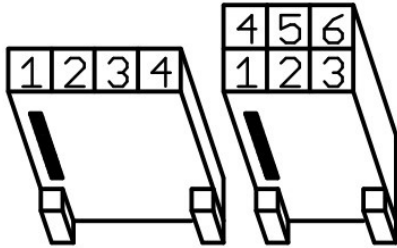


Pins	TE Connectivity	Farnell	RS Components
3 pins, 1 row	281838-3	429582	532-333
4 pins, 1 row	281838-4	429594	532-349
5 pins, 1 row	281838-5	429600	532-355
6 pins, 2 rows	281839-3	429650	532-406
8 pins, 2 rows	281839-4	429661	532-412
10 pins, 2 rows	281839-5	429673	532-428
12 pins, 2 rows	281839-6	429685	532-434



Crimp contacts	TE Connectivity	Farnell	RS Components
AWG 28-24	182734-2	429715	532-456

In Neobotix products the pin assignment of the HE14 connectors is as shown below.



2.2 Würth Elektronik - MPC4

Please check the [Würth Elektronik online catalogue](#)⁴ for details on the MPC4⁵.



Pins (in 2 rows)	Würth Elektronik
2	649002113322
4	649004113322
6	649006113322
8	649008113322
10	649010113322
16	649016113322

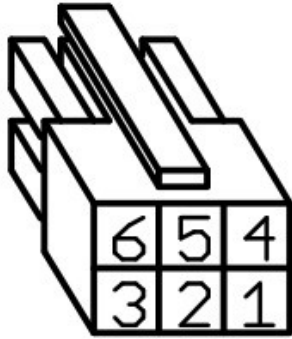


Crimp contacts	Würth Elektronik
AWG 24-18	64900613722

In Neobotix products the pin assignment of the MPC4 connectors is as shown below.

⁴ <https://www.we-online.com/en/products/components/overview>

⁵ https://www.we-online.de/katalog/de/em/connectors/wire-to-board/wr_mpc4/



2.3 Würth Elektronik - MPC3

Please check the Würth Elektronik online catalogue⁶ for details on the MPC3⁷.



Pins (in 2 rows)	Würth Elektronik
4	662004113322
6	662006113322
12	662012113322

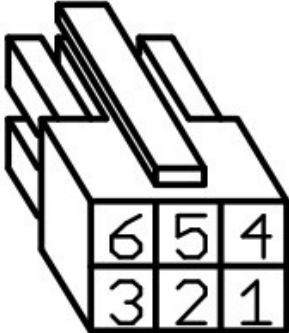


Crimp contacts	Würth Elektronik
AWG 24-20	66200113722

In Neobotix products the pin assignment of the MPC3 connectors is as shown below.

⁶ <https://www.we-online.com/en/products/components/overview>

⁷ https://www.we-online.de/katalog/de/em/connectors/wire-to-board/wr_mpc3/



Qualified Personnel

This product must only be modified, commissioned, operated and serviced by qualified personnel. Qualified personnel are defined as persons who

- due to their specialist training and experience have adequate knowledge for the work at hand,
- have been instructed by the responsible robot operator in the operation of the robot or its parts and the currently valid safety guidelines,
- are sufficiently familiar with the applicable official health and safety regulations, directives and generally recognised engineering practice (e.g. DIN standards, VDE stipulations, engineering regulations from other EC member states) that they can assess the work safety aspects of the product and
- have access to this manual and who have read it.

The following groups of persons are generally not considered qualified:

- Employees, interns or other academic staff not familiar with the product,
- visitors and guests,
- all members of other departments of the company or institution in which the product is operated.

This list is not intended to be exhaustive.

4.1 Version Information

The English part of this online documentation has been translated and is not the original. Please refer to the German version in case of uncertainties or questions.

4.2 Liability

Every care has been taken in the preparation of this manual which represents the state of technology at the time of its composing. However, inaccuracies or omissions might occur. Please inform Neobotix in case you notice any.

The Neobotix GmbH cannot be held responsible for any technical or typographical errors and reserves the right to make changes to the product and manual without prior notice. Neobotix makes no warranty of any kind with regard to the material contained within this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Neobotix GmbH shall not be liable or responsible for incidental or consequential damages in connection with the improper use of one or more of the products described in this manual.

4.3 Downloads and Further Information

Additional information, data sheets and documentations, also for other products of Neobotix, can be found in the download section on our homepage: <https://www.neobotix-robots.com/service/downloads>.