



USBoard-USS5

Neobotix GmbH

29.04.2024

1	Einleitung	1
1.1	Zu dieser Dokumentation	1
1.1.1	Symbole und Konventionen	1
1.1.2	Weiterführende Informationen	2
1.2	Rechtliche Anmerkungen	2
1.2.1	Versionsangabe	2
1.2.2	Haftung	2
1.2.3	Downloads und weitergehende Informationen	2
2	USBoard-USS5	3
2.1	Einleitung	4
2.1.1	Bestimmungsgemäße Verwendung	4
2.1.2	Unzulässige Verwendung	4
2.1.3	Sachkundiges Personal	4
2.2	Funktionsweise	4
2.2.1	Grundlegende Funktionsweise	4
2.2.2	Kreuzechomodus	5
2.3	Technische Daten	6
2.3.1	USBoard-USS5	6
2.3.2	Ultraschallsensor Bosch USS5	7
2.3.3	USS5 Erfassungsbereich	7
2.4	Parametersatz	9
2.5	Befehlssatz	12
2.5.1	Befehls-IDs	12
2.5.2	CAN-Kommunikation	12
2.5.3	Serielle Schnittstelle	12
2.6	Befehle	13
2.6.1	CMD_CONNECT	14
2.6.2	CMD_SET_CHANNEL_ACTIVE	14
2.6.3	CMD_GET_DATA_1TO8	14
2.6.4	CMD_GET_DATA_9TO16	15
2.6.5	CMD_WRITE_PARASET	15
2.6.6	CMD_WRITE_PARASET_TO_EEPROM	16
2.6.7	CMD_READ_PARASET	16
2.6.8	CMD_GET_ANALOGIN	16
2.6.9	CMD_GET_DATA	17

2.7	ROS-Node	17
2.7.1	Installation	18
2.7.2	Starten	18
2.7.3	Parameter	18
2.7.4	Topics	19
2.7.5	Mehrere USBoards	19
2.7.6	Hilfe	21
2.8	ROS 2 Node	21
2.8.1	Installation	21
2.8.2	Starten	22
2.8.3	Parameter	22
2.8.4	Topics	22
2.8.5	Hilfe	23
2.9	Grafische Bedienoberfläche	23
2.9.1	Einleitung	23
2.9.2	Erste Schritte	24
2.9.3	Konfiguration	26
2.10	Abmessungen	30
2.11	Steckerbelegung	31
2.11.1	X1	31
2.11.2	X2 - X5	31
2.11.3	X6	32
2.11.4	X7	32
2.11.5	USS5	32
2.12	Zukaufteile	32
2.12.1	Steckverbinder	32
2.12.2	USBoard-USS5-Konfigurationskabel	32
2.12.3	Sensorkabelsätze	33
2.12.4	Steckverbindersätze	33
2.12.5	Ultraschallsensoren	33
2.12.6	USS5-Einbaugehäuse	36
2.13	Häufige Fragen	36
2.13.1	Allgemeine Fragen	37
2.13.2	Verbindungsprobleme	37
2.14	Rechtliche Anmerkungen	37
2.14.1	EG-Konformitätserklärung	38
2.14.2	Informationen zu RoHS	38
3	Steckverbinder	39
3.1	TE Connectivity - HE14	39
3.2	Würth Elektronik - MPC4	40
3.3	Würth Elektronik - MPC3	41
4	Sachkundiges Personal	43

1.1 Zu dieser Dokumentation

1.1.1 Symbole und Konventionen

In dieser Dokumentation werden die folgenden Symbole und Hervorhebungen verwendet:

Gefahr: Weist auf eine Gefahrenstelle oder unmittelbar gefährliche Situation hin, die zu schweren Verletzungen bis hin zum Tod führen kann.

Warnung: Weist auf eine Gefahrenstelle oder potentiell gefährliche Situation hin, die zu schweren Verletzungen oder Schäden führen kann.

Achtung: Weist auf Gefahren oder Situationen hin, die zu leichten Verletzungen, Schäden oder anderen negativen Auswirkungen führen kann.

Bemerkung: Kennzeichnet wichtige Informationen, die für den sicheren Betrieb beachtet werden müssen.

Tipp: Gibt hilfreiche Hinweise, die die Arbeit mit dem mobilen Roboter einfacher und effizienter machen.

- Aufzählungen enthalten mehrere Informationen zum gleichen Thema.
- Sie sind, soweit möglich und sinnvoll, nach Priorität geordnet, mit dem jeweils wichtigsten Eintrag ganz oben.
- Aufzählungen erheben keinen Anspruch auf Vollständigkeit, sofern nichts anderes angegeben wurde.

1. Handlungsanleitungen sind nummeriert.
2. Nummerierte Anleitungen müssen in der gegebenen Reihenfolge befolgt werden.

1.1.2 Weiterführende Informationen

Weitere Informationen, insbesondere zu kundenspezifischen Konfigurationen und Systemen, erhalten Sie bei Auslieferung zusammen mit Ihrem Roboter oder [auf Anfrage](#)¹. In den meisten Fällen stehen Ihnen alle Unterlagen zu Ihrem individuellen Roboter auch [im Kundenbereich](#)² unserer Website zu Verfügung.

1.2 Rechtliche Anmerkungen

1.2.1 Versionsangabe

Der deutsche Teil dieser Online-Dokumentation ist das Original.

1.2.2 Haftung

Dieses Dokument wurde mit größtmöglicher Sorgfalt verfasst und repräsentiert den Stand der Technik zum Zeitpunkt seiner Erstellung. Fehler und Irrtümer sind jedoch nicht auszuschließen. Bitte informieren Sie Neobotix, sollten Sie solche im Dokument bemerken.

Die Neobotix GmbH ist nicht haftbar für technische oder schriftliche Fehler in diesem Dokument und behält sich das Recht vor, Änderungen seines Inhalts vorzunehmen, ohne diese vorher anzukündigen. Neobotix übernimmt keinerlei Garantie für die in diesem Dokument beschriebenen Produkteigenschaften. Insbesondere ergibt sich aus dem Inhalt kein Anspruch jedweder Art, weder auf Eigenschaften des Produkts noch auf seine Eignung für spezielle Anwendungsfälle. Die Neobotix GmbH kann nicht für Schäden haftbar gemacht werden, die aus der unsachgemäßen Nutzung eines oder mehrerer der beschriebenen Produkte resultieren.

1.2.3 Downloads und weitergehende Informationen

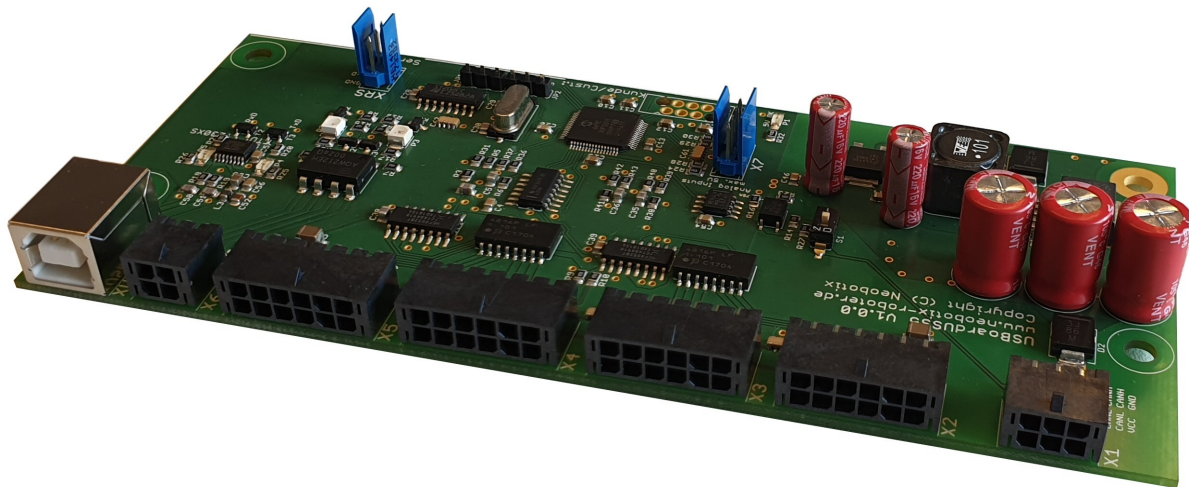
Weitergehende Informationen, Datenblätter und Dokumentationen, auch von weiteren Neobotix-Produkten, finden Sie im Downloadbereich unserer Website: <https://www.neobotix-roboter.de/service/downloads>.

¹ <https://www.neobotix-roboter.de/kontakt/daten>

² <https://www.neobotix-roboter.de/login>

USBoard-USS5

↓ Als PDF herunterladen³



Ultraschallsensoren messen die Entfernung zu Objekten, die sich innerhalb des Detektionsbereichs des Sensors befinden. Das physikalische Messprinzip beruht auf der Laufzeitmessung des Schalls, der vom Sensor ausgesandt, vom Hindernis reflektiert und vom Sensor wieder aufgefangen wird.

Das USBoard-USS5 ist für den Anschluss von bis zu 16 Ultraschallsensoren vom Typ Bosch USS5 ausgelegt. Es eignet sich beispielsweise für die Kollisionsabsicherung von autonomen Fahrzeugen wie mobilen Robotern, fahrenden Drohnen oder fahrerlosen Transportfahrzeugen. Das USBoard-USS5 bietet eine einfache Inbetriebnahme, komfortable Parametrierung und vielfältige Anwenderschnittstellen inklusive der Definition von Warn- und Alarmbereichen, die eine einfache Überwachung der Kollisionsgefahr ermöglichen.

Weiterhin stehen auf dem Board vier Analogeingänge zur Verfügung, die beispielsweise für den Anschluss von zusätzlichen Sensoren genutzt werden können.

³ <https://neobotix-docs.de/hardware/de/USBoard-USS5.pdf>

Tipp: Die graphische Benutzeroberfläche für das USBoard-USS5 finden Sie unter *Grafische Bedienoberfläche* (Seite 23).

2.1 Einleitung

Sehen Sie auch *Häufige Fragen* (Seite 36).

2.1.1 Bestimmungsgemäße Verwendung

Das USBoard-USS5 wurde für den Einsatz in mobilen Robotern und ähnlichen Geräten und Maschinen entwickelt, beispielsweise fahrenden Drohnen oder fahrerlosen Transportfahrzeugen. Es kann dort zur Erkennung von Hindernissen und zur Kollisionsvermeidung eingesetzt werden.

Darüber hinaus kann das USBoard-USS5 auch in stationären Aufbauten oder Maschinen eingesetzt werden, um etwa Messungen vorzunehmen.

Das USBoard-USS5 ist ausschließlich für die Bereitstellung nicht-sicherer Daten und Informationen entwickelt worden.

2.1.2 Unzulässige Verwendung

Das USBoard-USS5 ist explizit **kein Sicherheitsbauteil** und darf nicht als solches verwendet werden. Insbesondere darf es nicht zur Absicherung von Gefahrenstellen oder als alleiniges System zur Kollisionsvermeidung eingesetzt werden.

Ein Einsatz des USBoard-USS5 ist nicht empfehlenswert in Anwendungen, die sehr hohe Anforderungen an die Messgenauigkeit, an die Zuverlässigkeit der Messungen oder an die Geschwindigkeit der Messungen stellen. Dies sind zum Beispiel Anwendungen im Bereich Bewegungsregelung (Parken, Rangieren etc.).

2.1.3 Sachkundiges Personal

Das USBoard-USS5 darf nur von *sachkundigem Personal* (Seite 43) montiert, angeschlossen und in Betrieb genommen werden.

2.2 Funktionsweise

2.2.1 Grundlegende Funktionsweise

Das USBoard-USS5 misst die Laufzeit des Schallimpulses vom Sensor zum nächstgelegenen Hindernis und zurück. Aus der Laufzeit wird dann die Entfernung zum Hindernis berechnet.

Im Standardbetrieb werden die Sensoren nacheinander zyklisch abgefeuert und ausgewertet. Im *Kreuzechomodus* (Seite 5) arbeiten jeweils Gruppen von bis zu vier Sensoren zusammen. Nur einer der Sensoren erzeugt den Ultraschallpuls, aber alle vier Sensoren können das Echo empfangen. Wenn die geometrische Position aller Sensoren bekannt ist, lässt sich dann die Position des erkannten Objekts ermitteln.

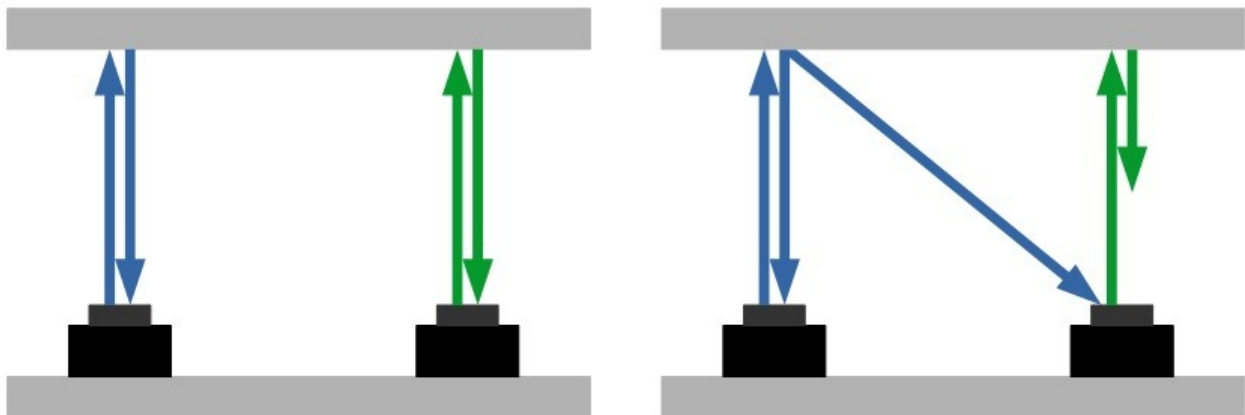
Die gemessenen Entfernungen können wahlweise zyklisch oder asynchron über CAN und / oder USB abgefragt werden. Auf Anfrage kann die USB-Schnittstelle auch durch RS-232 ersetzt werden.

Zur einfachen Verwendung als Kollisionsschutz können für jeden Sensor ein Warnbereich und ein Alarmbereich definiert werden. Wird bei mindestens einem Sensor die Warndistanz unterschritten, werden die entsprechende LED auf dem Board und ein Relaisausgang aktiviert. Eine zweite LED und ein weiterer Relaisausgang ermöglichen die Reaktion auf das Unterschreiten der Alarmdistanz. Warn- und Alarmbereiche werden komfortabel über die graphische Bedienoberfläche definiert und als Teil des Parametersatzes übertragen.

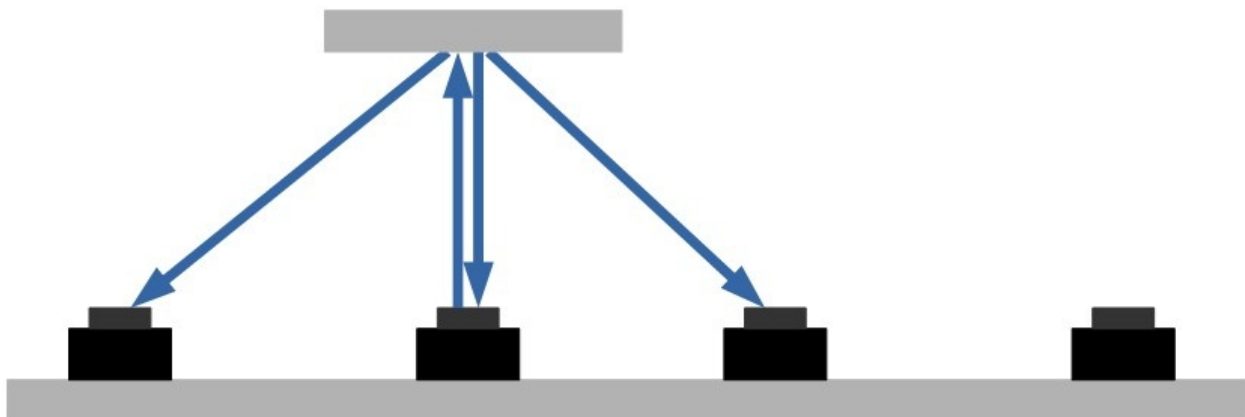
Warnung: Das USBoard-USS5 ist jedoch kein Sicherheitssystem und kann ausschließlich unterstützende, nicht sichere Daten liefern. Verwenden Sie das USBoard-USS5 niemals zur Absicherung von Gefahrenstellen.

2.2.2 Kreuzechomodus

Im klassischen Direktechomodus sind die Sensoren unabhängig voneinander. In einer Dauerschleife arbeiten die Sensoren nacheinander, wobei jeder aktive Sensor einen eigenen Ultraschallpuls aussendet und anschließend auf das Echo wartet, aus dem die Entfernung zum Objekt berechnet wird (linkes Bild unten). Falls ein Sensor zufällig ein Echo vom Puls eines früheren Sensors empfängt, kann dies nicht erkannt werden und es kommt zu einer Fehlmessung (rechtes Bild).



Im Kreuzechomodus dagegen sendet immer nur ein Sensor aus einer Vierergruppe den Ultraschallpuls aus aber alle vier Sensoren können das Echo empfangen. Aus den verschiedenen Schallaufzeiten vom Sender zum Objekt und zurück zu allen Sensoren können dann, wenn die Lage aller Sensoren zueinander bekannt ist, Rückschlüsse über die relative Position des erkannten Objekts gezogen werden.



Für die Positionsbestimmung des Objekts sind jedoch einige Aspekte zu beachten.

- Je nach Form und Oberfläche eines Objekts wird der Ultraschallpuls anders reflektiert, gestreut oder absorbiert.

Es ist deshalb möglich, dass direkte Messungen im normalen Modus ein Objekt erkennen, die Kreuzechos aber zu schwach sind, um im Kreuzechomodus genutzt zu werden.

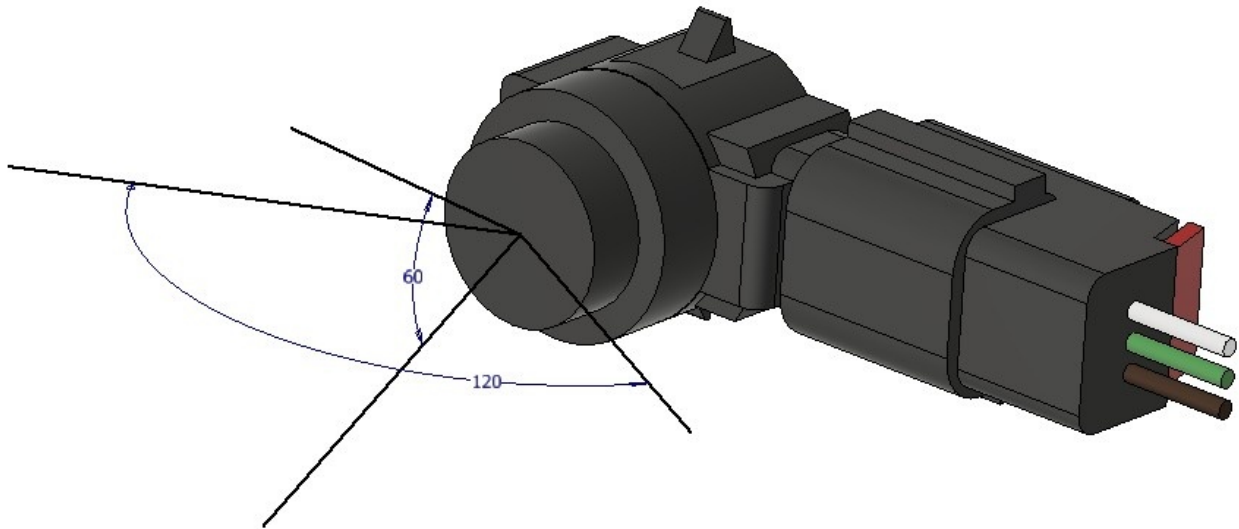
- Der reale Erfassungsbereich der Sensoren ist bei großen Reichweiten relativ schmal und zudem abhängig von Form und Material des Hindernisses. Es ist deshalb empfehlenswert, die Möglichkeiten des Kreuzechomodus vorab ausgiebig zu testen und sich mit den Eigenschaften und Fähigkeiten des USBoard-USS5 vertraut zu machen.
- Die auf dem USBoard-USS5 durchgeführte Entfernungsmessung ist in beiden Betriebsmodi identisch: die Schalllaufzeit wird halbiert und mit der Schallgeschwindigkeit multipliziert. Dies reduziert die Rechen- und Zykluszeit und vereinfacht den Datenaustausch. Bei relativ geringem Abstand zwischen den verwendeten Sensoren geben die Ausgabewerte in erster Näherung bereits einen guten Eindruck der Konstellation. Für eine exakte Berechnung auf Kundenseite müssen die Messwerte jedoch verdoppelt werden, um die komplette Distanz vom Sender zum Hindernis und weiter bis zum Empfänger zu erhalten.

2.3 Technische Daten

2.3.1 USBoard-USS5

- Versorgungsspannung von +9 VDC bis +60 VDC, max. 4 W
- Digitale Kommunikationsschnittstellen CAN und USB, optional RS-232
- Optische Anzeige einer Warn- / Alarmbereichsverletzung
- Solid-State-Relaisausgänge zur Anzeige von Warn- / Alarmbereichsverletzungen (max. 60 V, 0,5 A)
- Vier Analogeingänge (0 V – 5 V)
- Gewicht: 71 g
- Lagertemperaturbereich: -40 °C – 85 °C
- Betriebstemperaturbereich: 0 °C – 70 °C
- Zolltarifnummer: 9031 9000
- Bestellnummer: X101

2.3.2 Ultraschallsensor Bosch USS5



- Theoretischer Öffnungswinkel: $120^\circ \times 60^\circ$, symmetrisch
- Realer Öffnungswinkel bei Maximaldistanz: $10^\circ \times 10^\circ$, symmetrisch
- Unsere Testergebnisse zum Öffnungswinkel finden Sie [unten](#) (Seite 7).
- Messfrequenz: 48 kHz
- Messentfernung: 0,2 m bis 3,3 m
- Erkennung von Objekten im Nahbereich ($< 0,2$ m)
- Anzeige von leerem Messbereich (kein Echo)
- Zolltarifnummer: 9031 8080
- Bestellnummer: X210

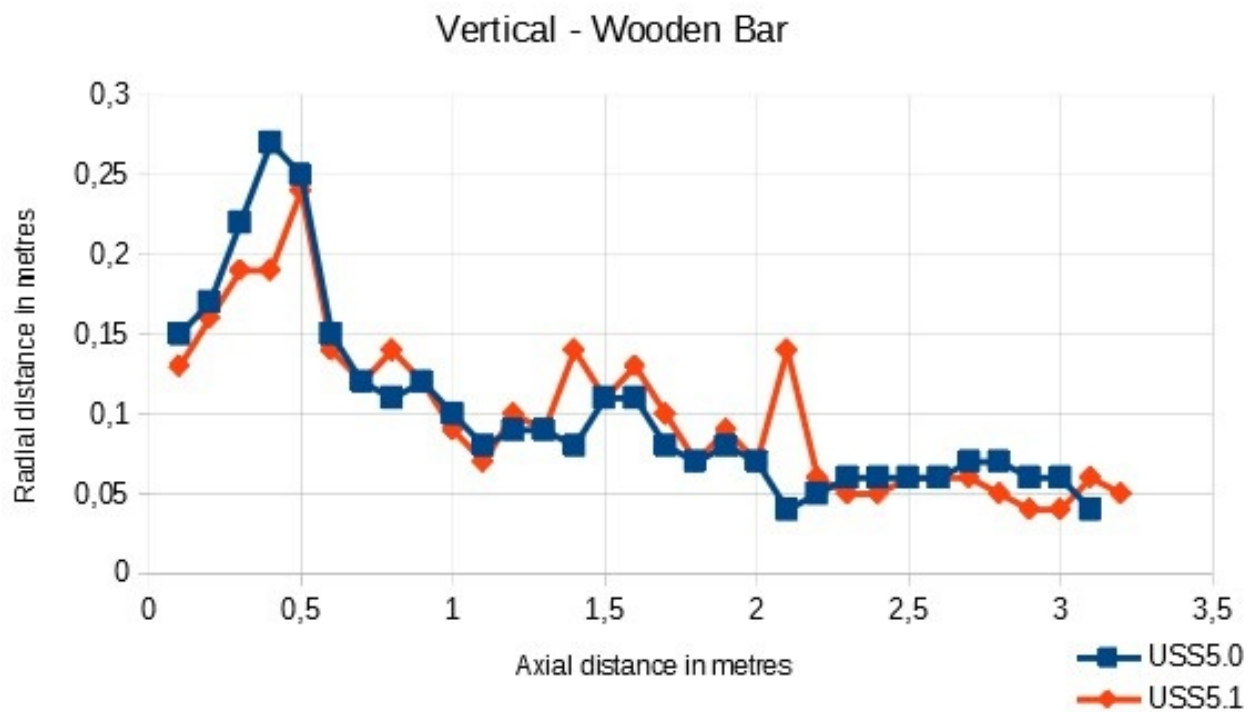
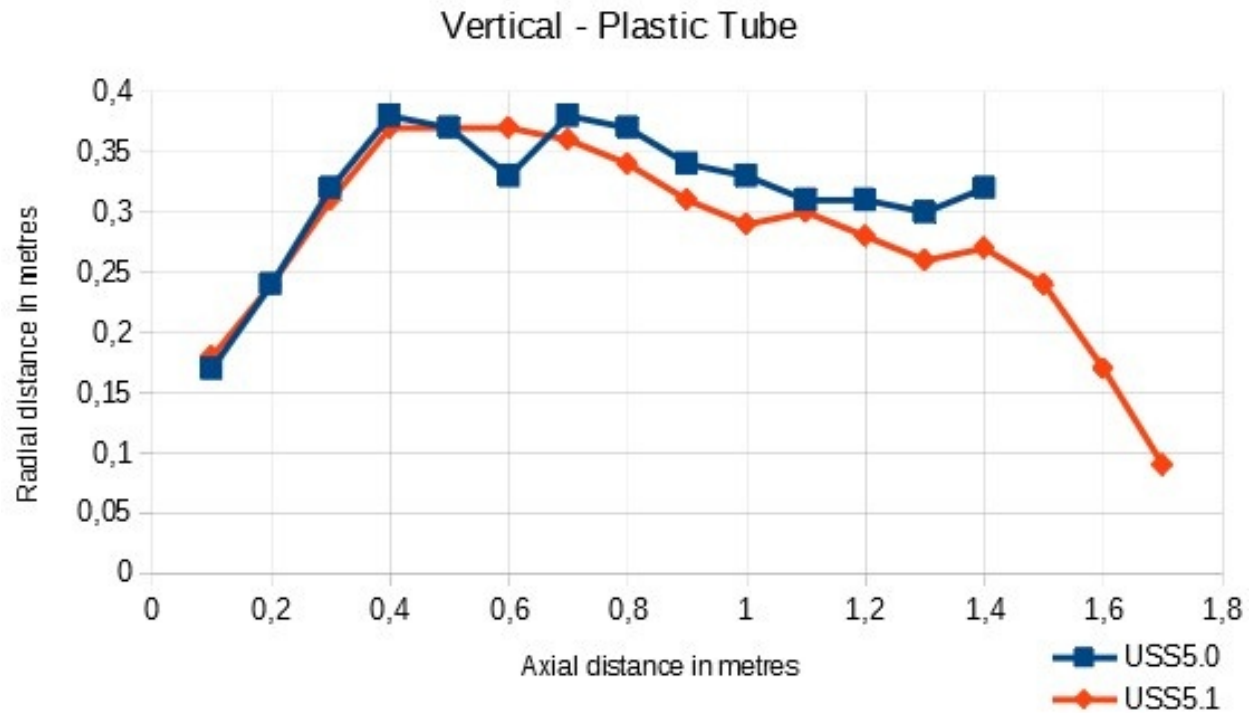
2.3.3 USS5 Erfassungsbereich

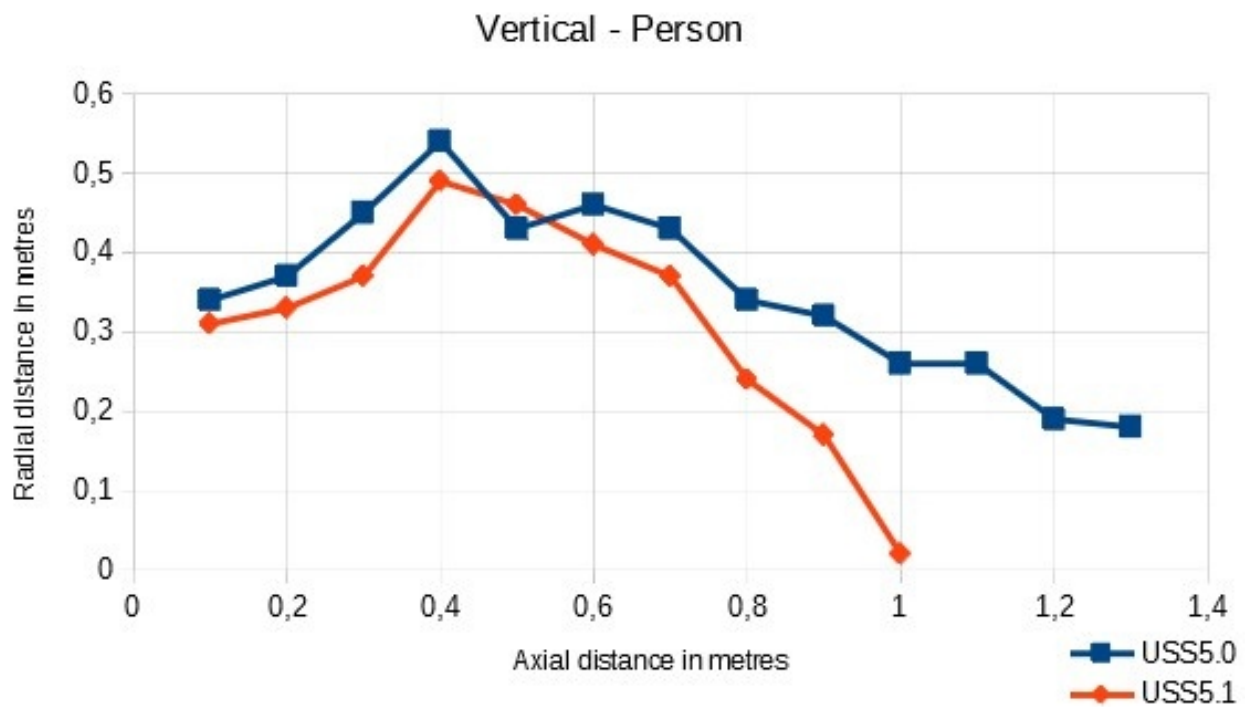
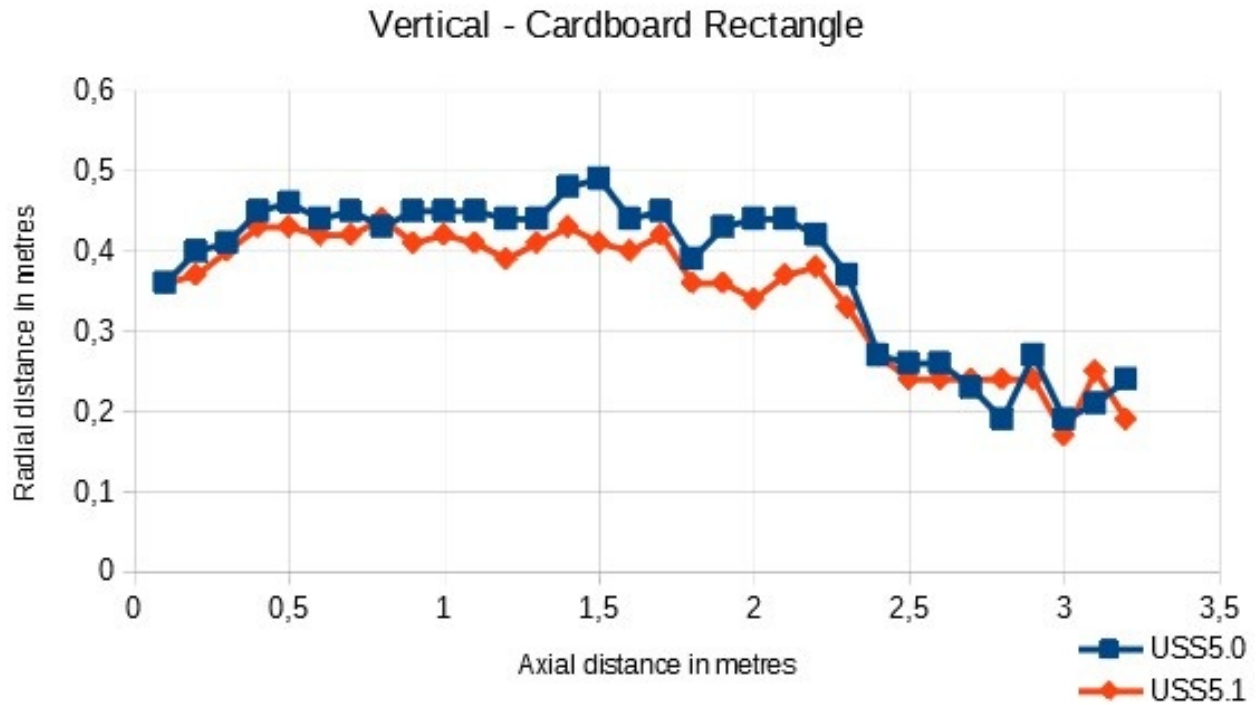
Wir haben den Erfassungsbereich der USS5 praktisch ermittelt und dazu verschiedene Objekte und Sensorkonfigurationen kombiniert. Die Testobjekte waren

- Kunststoffrohr (Durchmesser 75 mm, Höhe 1.75 m)
- Holzbalken (Breite 0.1 m, Höhe 1.77 m)
- Rechteckige Pappe (Breite 0.6 m, Höhe 0.4 m)
- Mensch (Höhe 1.8 m)

Unsere kompletten Messwerte können Sie [hier](#) herunterladen.

Bemerkung: Bitte beachten Sie, dass alle hier aufgeführten Daten lediglich zur Information dienen und keine Zusage von Produkteigenschaften darstellen. Bitte führen Sie eigene Tests durch, um die Eignung des USBoard-USS5 und der USS5-Sensoren für Ihre individuelle Anwendung zu prüfen.





2.4 Parametersatz

Das USBoard-USS5 verwendet einen Parametersatz, mit dem eine individuelle Konfiguration möglich ist. Der Parametersatz kann komfortabel mit der mitgelieferten grafischen Benutzeroberfläche editiert, auf der Festplatte des Konfigurations-Rechners gespeichert, in das EEPROM des USBoard-USS5 geschrieben und aus dem EEPROM gele-

sen werden.

Die folgende Tabelle zeigt den Aufbau des Parametersatzes, der aus 54 Bytes besteht.

Bemerkung: Viele Funktionen beeinflussen die Sensoren gruppenweise. Gruppe 0 besteht aus den Sensoren 1 bis 4, Gruppe 1 aus den Sensoren 5 bis 8 und so weiter.

Byte 0 CAN-Datenrate

0	1000 kBAud (Werkseinstellung)
1	500 kBAud
2	250 kBAud
3	125 kBAud
4	100 kBAud
5	50 kBAud

Bytes 1-4 CAN Basisadresse (standardmäßig 0x400), zu berechnen wie folgt:

```
(byte_4 << 24) | (byte_3 << 16) | (byte_2 << 8) | byte_1
```

Byte 5 Flags als Bitmaske

Bit	Bedeutung
0	Aktiviere CAN Extended ID (Werkseinstellung 0)
1	CAN-Abschlusswiderstand des USBoard-USS5-IP aktivieren (Werkseinstellung 0)
2	Analogeingänge aktivieren (Werkseinstellung 0)
3	Klassisches Datenformat verwenden (Werkseinstellung 0)
4	Warnrelais-Modus bei Verdeckung (Werkseinstellung 0) (siehe Konfiguration (Seite 26))
5	Alarmrelais-Modus bei Verdeckung (Werkseinstellung 0) (siehe Konfiguration (Seite 26))

Das Klassische Datenformat ist das selbe wie bei erstem USBoard, kompatibel zu RelayBoardV2. Dies betrifft nur automatischen Sendemodus.

Byte 6 Bits 0 bis 3 enthalten den Sendemodus für Messdaten.

0	Auf Anfrage
1	kontinuierliches Senden ohne Anforderung über CAN
2	Automatisch über USB / RS-232
3	Automatisch über CAN und USB / RS-232

Bits 4 bis 7 wählen aus, welche Gruppe Messdaten sendet (Bit 4 für Gruppe 0, Bit 5 für Gruppe 1 und so weiter). Wenn alle Bits auf 0 gesetzt sind, senden alle Gruppen.

Voreingestellt ist 0xFF, d.h. alle Gruppen senden auf Anfrage.

Byte 7 Bits 0 bis 3 enthalten das Sendeintervall für den automatischen Sendemodus.

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

Bits 4 bis 7 kodieren ein benutzerdefiniertes Sendeintervall in Inkrementen von 50 ms (0 für 50 ms, 1 für 100 ms und so weiter).

Byte 8 Aktive Sensoren 1 bis 8, bitkodiert. Voreinstellung ist 0xFF, also alle aktiv.

Deaktivierte Sensoren senden keine Ultraschallpulse. Durch das Deaktivieren von nicht verwendeten Sensor-kanälen kann die Zykluszeit des Systems reduziert werden.

Byte 9 Aktive Sensoren 9 bis 16, bitkodiert. Voreinstellung ist 0xFF, also alle aktiv.

Deaktivierte Sensoren senden keine Ultraschallpulse. Durch das Deaktivieren von nicht verwendeten Sensor-kanälen kann die Zykluszeit des Systems reduziert werden.

Bytes 10-25 Warnbereiche der Sensoren 1 – 16, in cm. Werkseinstellung ist 100 cm.

Bytes 26-41 Alarmbereiche der Sensoren 1 – 16, in cm. Werkseinstellung ist 30 cm.

Byte 42 Messauflösung, 2 Bits pro Gruppe

0	1 cm
1	0.5 cm (Werkseinstellung)
2	0.25 cm
3	0.125 cm

Byte 43 Kreuzechomodus aktiv, ein Bit pro Gruppe. Voreinstellung ist 0.

Byte 44 Kreuzechokonfiguration, 2 Bits pro Gruppe geben Sender an (0 bis 3). Voreinstellung ist 0, also der erste Sensor der Gruppe.

Bytes 45-46 Messintervall in Inkrementen von 10 ms, 4 Bits pro Gruppe (0 für 10 ms, 1 für 20 ms und so weiter). Voreinstellung ist 20 ms.

Byte 47 Tiefpassfilterparameter (siehe unten) für Entfernungsmessungen, in Inkrementen von 0.0078125. Voreinstellung ist 128, also Wert 1, Filterung deaktiviert.

Byte 48 Reserved, set to arbitrary values when sending.

Byte 49 Hardware-Version, nur lesen.

20	USBoard-USS5 V1.0.0
21	USBoard-USS5 V1.0.1
200	USBoard-USS5-IP V1.0.0
201	USBoard-USS5-IP V1.0.1

Bytes 50-53 Seriennummer, nur lesen.

(Im USBoard V1 werden nur die Bytes 51-53 benutzt.)

Der Tiefpassfilter kann genutzt werden, um die auszugebenden Messwerte zu glätten, oder um zu verhindern, dass Ausreißer sofort die Relais schalten, etwa durch einzelne fehlerhafte Messungen. Jeder neue Messwert eines Sensors wird entsprechend der Filtergewichtung mit dem aktuellen Ausgabewert verrechnet.

Ein Filterwert von 0.3 bedeutet, dass der neue Ausgabewert zu 70% aus dem letzten Wert und zu 30% aus dem neuen Messwert berechnet wird.

Ein Filterwert von 1.0 bedeutet, dass jeder neue Messwert unverändert als Ausgabewert übernommen, die Filterung also deaktiviert wird.

2.5 Befehlssatz

2.5.1 Befehls-IDs

Die folgende Tabelle listet die vom USBoard-USS5 unterstützten Befehle auf.

Kommando	Wert	Beschreibung
CMD_CONNECT	0	Verbindungstest ausführen
CMD_SET_CHANNEL_ACTIVE	1	Kanal für Senden/ Empfangen aktivieren
CMD_GET_DATA_1TO8	2	Messdaten Sensoren 1 bis 8 anfordern
CMD_GET_DATA_9TO16	3	Messdaten Sensoren 9 bis 16 anfordern
CMD_WRITE_PARASET	4	Parametersatz zum Board übertragen (flüchtig)
CMD_WRITE_PARASET_TO_EEPROM	5	Parametersatz ins EEPROM schreiben (nicht-flüchtig)
CMD_READ_PARASET	6	Parametersatz vom Board lesen
CMD_GET_ANALOGIN	7	Werte der 4 Analogeingänge lesen
–	8-12	Reserviert
CMD_GET_DATA	13	Messdaten aller Sensoren anfordern

2.5.2 CAN-Kommunikation

Die Werksvoreinstellung für die Basisadresse lautet 0×400 . Diese Basisadresse kann bei Bedarf innerhalb des Parametersatzes geändert werden.

Die vom USBoard-USS5 benutzten Adressen werden von der Basisadresse berechnet, indem jeweils die folgenden Offsets addiert werden.

Offset zur Basisadresse	Funktion
+0	Empfangen von Befehlen
+1	Antwort auf CMD_CONNECT
+2	Erste Antwort auf CMD_GET_DATA_1TO8
+3	Zweite Antwort auf CMD_GET_DATA_1TO8
+4	Erste Antwort auf CMD_GET_DATA_9TO16
+5	Zweite Antwort auf CMD_GET_DATA_9TO16
+6	Antwort auf CMD_READ_PARASET
+7	Antwort auf CMD_GET_ANALOGIN
+8	Antwort auf CMD_WRITE_PARASET
+9	Antwort auf CMD_WRITE_PARASET_TO_EEPROM
+13	Erste Antwort auf CMD_GET_DATA (Gruppe 0: 1 - 4)
+14	Zweite Antwort auf CMD_GET_DATA (Gruppe 1: 5 - 8)
+15	Dritte Antwort auf CMD_GET_DATA (Gruppe 2: 9 - 12)
+16	Vierte Antwort auf CMD_GET_DATA (Gruppe 3: 13 - 16)

2.5.3 Serielle Schnittstelle

Die Kommunikation per USB erfolgt über einen USB-Seriell-Konverter auf dem USBoard-USS5 und kann vom Betriebssystem wie eine klassische serielle Schnittstelle verwendet werden. Auf Anfrage kann der USB-Konverter durch einen RS-232-Konverter ersetzt werden. Das Kommunikationsprotokoll ändert sich dabei nicht.

Bemerkung: Bitte beachten Sie, dass die RS-232-Schnittstelle eine gemeinsame Masseverbindung zwischen USBoard-USS5 und Rechner erfordert.

Die serielle Schnittstelle arbeitet mit 19.200 Baud. Das Protokoll besitzt das gleiche Format wie die CAN-Kommunikation, jedoch mit den folgenden Erweiterungen:

Nachrichten **von** dem USBoard-USS5 beginnen mit dem Startbyte 0xFF. Am Ende jeder Nachricht, bestehend aus acht Datenbytes, wird eine 16 Bit lange Prüfsumme angehängt. Damit ist eine vollständige Nachricht 11 Bytes lang.

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

Die Checksumme berechnet sich nach CRC-CCITT über alle 8 Datenbytes.

Die **zum USBoard-USS5** verschickten Nachrichten enthalten lediglich die Datenbytes 1-8.

Implementierung der Checksummenberechnung (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;
}

```

2.6 Befehle

Den Befehlssatz finden Sie unter *Befehlssatz* (Seite 12).

Bemerkung: In dieser Dokumentation bezieht sich Byte 0 auf das erste Daten-Byte nach dem Header. Innerhalb der einzelnen Bytes wird das höchstwertige Bit zuletzt übertragen.

Im Folgenden werden die CAN IDs als Offset zur Basisadresse angegeben, d.h. +3 bedeutet Basisadresse plus 3.

2.6.1 CMD_CONNECT

Mit diesem Kommando kann die Kommunikationsverbindung zum Board hergestellt und getestet werden.

Kommando ID: +0

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

Antwort ID: +1

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

2.6.2 CMD_SET_CHANNEL_ACTIVE

Mit diesem Kommando werden aus den 16 verfügbaren Kanälen die Kanäle ausgewählt, auf denen gesendet und empfangen werden soll. Zwei Bytes D1 und D2 für die Kanäle 1 bis 8 und 9 bis 16 enthalten die Information, ob der Kanal entsprechend der Bitnummer aktiv sein soll. Die Bytes sind Bit-codiert, wobei eine 1 einen aktiven Kanal markiert. Der Befehl kann verwendet werden, wenn Sensoren aktiv oder passiv geschaltet werden sollen, ohne den vollständigen Parametersatz zu übertragen.

Beispiel: $0 \times 1F$ als erstes Byte bedeutet Sensoren 1 bis 5 sind aktiv, Sensoren 6 bis 8 sind deaktiviert.

Kommando ID: +0

CMD_SET_CHANNEL_ACTIVE	(Sensoren 1 bis 8)	(Sensoren 9 bis 16)	0	0	0	0	0
------------------------	--------------------	---------------------	---	---	---	---	---

Antwort Keine Antwort.

2.6.3 CMD_GET_DATA_1TO8

Mit diesem Kommando werden die Messwerte der Sensoren 1 bis 8 abgefragt.

Kommando ID: +0

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

Antwort (zwei Teile) IDs: +2, +3

CMD_GET_DATA_1TO8	0	Werte für Sensoren 1-4 (jeweils ein Byte)	0	(reserviert)
CMD_GET_DATA_1TO8	1	Werte für Sensoren 5-8 (jeweils ein Byte)	0	(reserviert)

Die Skalierung der Messwerte entspricht der konfigurierten Auflösung der Sensorgruppe (siehe [Parametersatz](#) (Seite 9)).

Bemerkung: Dieser Befehl ist veraltet. Der höchstmögliche Wert ist 255 (1 Byte), höhere Bits werden nicht übertragen.

2.6.4 CMD_GET_DATA_9TO16

Mit diesem Kommando werden die Messwerte der Sensoren 9 bis 16 abgefragt.

Kommando ID: +0

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

Antwort (zwei Teile) IDs: +4, +5

CMD_GET_DATA_9TO16	0	Werte für Sensoren 9-12 (jeweils ein Byte)	0	(reserviert)
CMD_GET_DATA_9TO16	1	Werte für Sensoren 13-16 (jeweils ein Byte)	0	(reserviert)

Die Skalierung der Messwerte entspricht der konfigurierten Auflösung der Sensorgruppe (siehe [Parametersatz](#) (Seite 9)).

Bemerkung: Dieser Befehl ist veraltet. Der höchstmögliche Wert ist 255 (1 Byte), höhere Bits werden nicht übertragen.

2.6.5 CMD_WRITE_PARASET

Mit diesem Kommando wird ein vollständiger Parametersatz an das USBoard-USS5 übermittelt (flüchtig, nach dem Ausschalten verloren). Der Parametersatz kann komfortabel mit dem mitgelieferten Parameter-Editor erstellt und übertragen werden.

Die neuen Parameter werden nach dem Schreiben sofort aktiviert.

Kommando ID: +0

Der Befehl besteht aus neun Nachrichten, die nacheinander gesendet werden, wobei jede Nachricht ein paar Bytes des Parametersatzes enthält.

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

Antwort ID: +8

Jede Nachricht wird einzeln beantwortet, wobei die ersten 8 Antworten identisch sind

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

und die letzte Antwort die Summe der ersten 48 Bytes des Parametersatzes enthält.

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

2.6.6 CMD_WRITE_PARASET_TO_EEPROM

Mit diesem Kommando wird ein vollständiger Parametersatz in das EEPROM des Boards geschrieben (nicht flüchtig). Ansonsten ist der Befehl mit CMD_WRITE_PARASET identisch.

Der Parametersatz wird nach dem Schreiben sofort übernommen.

Kommando ID: +0

Dieser Befehl ist identisch zu *CMD_WRITE_PARASET* (Seite 15), nur dass er das Befehls-Byte CMD_WRITE_PARASET_TO_EEPROM verwendet.

Antwort ID: +9

Die Antwort ist identisch zu den Antworten auf *CMD_WRITE_PARASET* (Seite 15), nur dass sie das Befehls-Byte CMD_WRITE_PARASET_TO_EEPROM verwendet.

2.6.7 CMD_READ_PARASET

Mit diesem Kommando wird der aktuelle, vollständige Parametersatz vom USBoard-USS5 gelesen. Die Antwort besteht aus neun Nachrichten, die hintereinander gesendet werden.

Kommando ID: +0

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

Antwort ID: +6

Die Antwort besteht aus neun Nachrichten, die nacheinander gesendet werden, wobei jede Nachricht ein paar Bytes des Parametersatzes enthält.

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

2.6.8 CMD_GET_ANALOGIN

Mit diesem Kommando werden die Messwerte der vier Analogeingänge abgefragt.

Kommando ID: +0

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

Antwort ID: +7

Die Auflösung beträgt 12 Bit, so dass innerhalb der CAN-Antwort zunächst das untere Byte des entsprechenden Kanals übertragen wird. In den Bytes 5 und 6 der CAN-Nachricht sind die oberen vier Bits der Kanäle 1 – 4 enthalten.

CMD_GET_DATA	low 8 bit channel 0	low 8 bit channel 1	low 8 bit channel 2	low 8 bit channel 3	high 4 bits of channel 0 and 1	high 4 bits of channel 3 and 4	0
--------------	---------------------	---------------------	---------------------	---------------------	--------------------------------	--------------------------------	---

2.6.9 CMD_GET_DATA

Mit diesem Kommando werden die Messwerte ausgewählter Sensoren abgefragt. Die Auswahl erfolgt durch eine Bitmaske, in der Bit 0 für Gruppe 0 steht, Bit 1 für Gruppe 1 und so weiter. Pro ausgewählter Gruppe wird eine Antwort gesendet.

Kommando ID: +0

CMD_GET_DATA	(ausgewählte Gruppen)	0	0	0	0	0	0
--------------	-----------------------	---	---	---	---	---	---

Antwort ID: +(13 + group_id)

CMD_GET_DATA	Info (info)	lower 8 bit sensor 0	lower 8 bit sensor 1	lower 8 bit sensor 2	lower 8 bit sensor 3	high 4 bits of sensors 0 and 1	high 4 bits of sensors 3 and 4
--------------	-------------	----------------------	----------------------	----------------------	----------------------	--------------------------------	--------------------------------

Das *info* byte enthält

- 2 Bits für die Gruppe (0 bis 3)
- 2 Bits für die Sensorauflösung (0 für 1cm, 1 für 0.5cm, 2 für 0.25cm, 3 für 0.125cm)
- 4 Bits für den sendenden Sensor, entweder `0xff` wenn alle Sensoren senden oder im Kreuzechomodus das Bit des aktiven Sensors auf 1 gesetzt.

Das *lower 8 bit* Byte eines Sensors kann auch einen Fehlercode enthalten:

- 0, wenn der Sensor nicht physisch an das USBoard-USS5 angeschlossen ist.
- 1 bedeutet, dass ein Objekt die Minimaldistanz des Sensors unterschreitet.
- 2 bedeutet, der Sensor ist aktiv und sendet Impulse, empfängt aber kein Echo. Das bedeutet normalerweise, dass das Objekt zu weit entfernt ist.

2.7 ROS-Node

Dieser Node übernimmt die Kommunikation des Neobotix USBoard-USS5.

Der USBoard-USS5-Node wurde getestet mit:

- ROS Kinetic auf Ubuntu 16.04
- ROS Melodic auf Ubuntu 18.04
- ROS Noetic auf Ubuntu 20.04

Sie finden den Quelltext der ROS-Node unter https://github.com/neobotix/neo_usboard_v2.

2.7.1 Installation

1. Klonen Sie das `neo_usboard_v2`-Repository in den source-Ordner Ihres Catkin Workspace:

```
cd your_catkin_workspace/src
git clone https://github.com/neobotix/neo_usboard_v2.git
```

2. Klonen Sie `neo_msgs` in den source-Ordner Ihres Catkin Workspace:

```
git clone https://github.com/neobotix/neo_msgs.git
```

3. Laden Sie die benötigten Submodule: `vnx-base`, `pilot-base` und `pilot-usboard`:

```
cd neo_usboard_v2
git submodule update --init
```

4. Installieren Sie `vnx-base`.

- ROS Kinetic:

```
sudo dpkg -i vnx-base/x86_64/vnx-base-1.9.3-x86_64-ubuntu-16.04.deb
```

- ROS Melodic:

```
sudo dpkg -i vnx-base/x86_64/vnx-base-1.9.3-x86_64-ubuntu-18.04.deb
```

- ROS Noetic:

```
sudo dpkg -i vnx-base/x86_64/vnx-base-1.9.3-x86_64-ubuntu-20.04.deb
```

5. Kompilieren Sie Ihren Workspace:

```
cd your_catkin_workspace
catkin_make
```

2.7.2 Starten

Wenn CAN verwendet wird muss der Bus zuerst konfiguriert werden:

```
sudo ip link set can0 down
sudo ip link set can0 type can bitrate 1000000
sudo ip link set can0 up
```

Um den USBoard-USS5 ROS Node zu starten, verwenden Sie:

```
roslaunch neo_usboard_v2 neo_usboard_v2.launch
```

2.7.3 Parameter

Die folgenden Parameter können Sie in `neo_usboard_v2.yaml` an Ihre Bedürfnisse anpassen:

Parameter	Wert	Hinweis
can_id	1024	Muss ein Vielfaches von 32 sein.
can_device	Nichts oder can0	
serial_port	/dev/ttyUSB0	
can_baud_rate	1000000 (bit/s)	Muss dem auf dem Board konfigurierten Wert entsprechen.
update_rate	5 (Hz)	Nur im „Request“-Sendemodus relevant.

Bemerkung: Falls `can_device` ein Wert zugewiesen wird, wird der CAN-Bus zur Kommunikation verwendet, andernfalls die mit `serial_port` spezifizierte serielle Schnittstelle.

Die folgenden Parameter können über den Parameterserver gesetzt werden, während die ROS-Node läuft.

Parameter	Wert	Hinweis
low_pass_gain	1	Verstärkung des Tiefpassfilters (1 = kein Filter)
enable_analog_input	false	Analogeingänge auslesen ein/aus
enable_legacy_format	false	Altes Nachrichtenformat benutzen (bei automatischem Senden)
enable_can_termination	false	CAN-Bus-Terminierung aktivieren
relay_warn_blocked_invert	false	Warn-Relay invertieren, wenn ein Sensor blockiert ist
relay_alarm_blocked_invert	false	Alarm-Relay invertieren, wenn ein Sensor blockiert ist
active_sensors(0 to 15)	true	Aktive Sensoren
warn_distance(0 to 15)	100 cm	Warndistanz pro Sensor
alarm_distance(0 to 15)	30 cm	Alarmdistanz pro Sensor
enable_transmission(0 to 4)	true	Gruppe in automatischem Modus aktiviert
fire_interval_ms	20 ms	Zeit zwischen zwei Impulsen
sending_sensor	0	Index des Sensors der im Kreuzechomodus den Impuls sendet
cross_echo_mode	false	Kreuzechomodus an/aus

Bemerkung: Um die ROS-Parameter aus der Anwendung heraus zu setzen, empfehlen wir den Service `/usboard_v2/set_parameters`. Wenn ein Setzen über die Kommandozeile nötig ist, benutzen Sie den Befehl `roslaunch dynamic_reconfigure dynparam set param value`. Nutzen Sie den Befehl `ros param list`, um die verfügbaren Parameter anzuzeigen.

2.7.4 Topics

Die folgenden ROS Topics sind verfügbar:

Name	Typ
/usboard_v2/measurements	neo_msgs/msgs/USBoardV2
/usboard_v2/sensor1	sensor_msgs/Range
/usboard_v2/sensor2	sensor_msgs/Range
...	
/usboard_v2/sensor16	sensor_msgs/Range

2.7.5 Mehrere USBoards

Im Fall von mehreren USBoard-USS5 ist es möglich, für jedes Board eine eigene ROS-Node zu starten.

2.7.5.1 Parameter

Jede ROS-Node braucht ihre eigene yaml-Konfigurationsdatei, sehen Sie zum Beispiel `neo_usboard_v2.yaml` und `neo_usboard_v2_1.yaml` in https://github.com/neobotix/neo_usboard_v2/tree/main/launch:

```
can_id: 1024
can_device: None
#can_device: can0
serial_port: /dev/ttyUSB0
can_baud_rate: 1000000
update_rate: 5
topic_path: /usboard_v2
```

```
can_id: 1056
can_device: None
#can_device: can1
serial_port: /dev/ttyUSB1
can_baud_rate: 1000000
update_rate: 5
topic_path: /usboard_v2_1
```

Bemerkung: CAN-IDs müssen ein Vielfaches von 0x20 sein, also 32 in Dezimalschreibweise.

2.7.5.2 Mehrere ROS-Nodes starten

Die Standard-Launchfile `neo_usboard_v2.launch` enthält ein Beispiel, um eine weitere ROS-Node zu starten:

```
<?xml version="1.0"?>
<launch>
  <!-- Load usboard_v2 params -->
  <rosparam command="load" ns="usboard_v2" file="$(find neo_usboard_v2)/launch/neo_
↪usboard_v2.yaml"/>
  <!-- start usboard_v2 node -->
  <node pkg="neo_usboard_v2" type="neo_usboard_v2" ns="usboard_v2" name="usboard_v2_
↪node" respawn="false" output="screen"/>

  <!-- Load usboard_v2 params for second board -->
  <rosparam command="load" ns="usboard_v2_1" file="$(find neo_usboard_v2)/launch/
↪neo_usboard_v2_1.yaml"/>
  <!-- start usboard_v2 node for second board -->
  <node pkg="neo_usboard_v2" type="neo_usboard_v2" ns="usboard_v2_1" name="usboard_
↪v2_node_1" respawn="false" output="screen"/>
</launch>
```

Entfernen Sie einfach die Kommentarzeichen beim zweiten Block und eine zweite ROS-Node zu starten.

Um alle konfigurierten Nodes zu starten:

```
roslaunch neo_usboard_v2 neo_usboard_v2.launch
```

2.7.6 Hilfe

Wenn Sie den Fehler `unable to connect to port /dev/ttyUSB0` erhalten, führen Sie folgenden Befehl aus:

```
sudo usermod -a -G dialout $USER
```

und starten Sie Ihren PC neu.

2.8 ROS 2 Node

Dieser Node übernimmt die Kommunikation des Neobotix USBoard-USS5.

Der USBoard-USS5-Node wurde getestet mit:

- ROS 2 Foxy auf Ubuntu 20.04, Rolling und Humble auf Ubuntu 22.04

Sie finden den Quelltext der ROS-Node unter https://github.com/neobotix/neo_usboard_v2-2.

Bemerkung: Wir unterstützen inzwischen auch die Distributionen Galactic und Humble. Bitte checken Sie bei Bedarf den entsprechenden Branch aus.

2.8.1 Installation

1. Clone the `neo_usboard_v2-2` repository into your `ros2` workspace source folder:

```
cd your_ros2_workspace/src
git clone https://github.com/neobotix/neo_usboard_v2-2.git
```

2. Klonen Sie `neo_msgs` in den source-Ordner Ihres Catkin Workspace:

```
git clone https://github.com/neobotix/neo_msgs2.git
```

3. Laden Sie die benötigten Submodule: `vnx-base`, `pilot-base` und `pilot-usboard`:

```
cd neo_usboard_v2-2
git submodule update --init
```

4. Installieren Sie `vnx-base`.

- ROS Foxy:

```
sudo dpkg -i vnx-base/x86_64/vnx-base-1.9.3-x86_64-ubuntu-20.04.deb
```

- ROS Humble:

```
sudo dpkg -i vnx-base/x86_64/vnx-base-1.9.6-x86_64-ubuntu-22.04.deb
```

5. Kompilieren Sie Ihren Workspace:

```
cd your_ros2_workspace
colcon build --symlink-install
. install/setup.bash
```


2.8.2 Starten

Wenn CAN verwendet wird muss der Bus zuerst konfiguriert werden:

```
sudo ip link set can0 down
sudo ip link set can0 type can bitrate 1000000
sudo ip link set can0 up
```

Um den USBoard-USS5 ROS Node zu starten, verwenden Sie:

```
ros2 launch neo_usboard_v2 neo_usboard_v2.launch.py
```

2.8.3 Parameter

Die folgenden Parameter können Sie in `neo_usboard_v2.yaml` an Ihre Bedürfnisse anpassen:

Parameter	Wert	Hinweis
<code>can_id</code>	1024	Muss ein Vielfaches von 32 sein.
<code>can_device</code>	Nichts oder <code>can0</code>	
<code>serial_port</code>	<code>/dev/ttyUSB0</code>	
<code>can_baud_rate</code>	1000000 (bit/s)	Muss dem auf dem Board konfigurierten Wert entsprechen.
<code>update_rate</code>	5 (Hz)	Nur im „Request“-Sendemodus relevant.

Bemerkung: Falls `can_device` ein Wert zugewiesen wird, wird der CAN-Bus zur Kommunikation verwendet, andernfalls die mit `serial_port` spezifizierte serielle Schnittstelle.

Die folgenden Parameter können über den Parameterserver gesetzt werden, während die ROS2-Node läuft.

Parameter	Wert	Hinweis
<code>low_pass_gain</code>	1	Verstärkung des Tiefpassfilters (1 = kein Filter)
<code>enable_analog_input</code>	false	Analogeingänge auslesen ein/aus
<code>enable_legacy_format</code>	false	Altes Nachrichtenformat benutzen (bei automatischem Senden)
<code>enable_can_termination</code>	false	CAN-Bus-Terminierung aktivieren
<code>relay_warn_blocked_invert</code>	false	Warn-Relay invertieren, wenn ein Sensor blockiert ist
<code>relay_alarm_blocked_invert</code>	false	Alarm-Relay invertieren, wenn ein Sensor blockiert ist
<code>active_sensors(0 to 15)</code>	true	Aktive Sensoren
<code>warn_distance(0 to 15)</code>	100 cm	Warndistanz pro Sensor
<code>alarm_distance(0 to 15)</code>	30 cm	Alarmdistanz pro Sensor
<code>enable_transmission(0 to 4)</code>	true	Gruppe in automatischem Modus aktiviert
<code>fire_interval_ms</code>	20 ms	Zeit zwischen zwei Impulsen
<code>sending_sensor</code>	0	Index des Sensors der im Kreuzechomodus den Impuls sendet
<code>cross_echo_mode</code>	false	Kreuzechomodus an/aus

2.8.4 Topics

Die folgenden ROS Topics sind verfügbar:

Name	Typ
/usboard_v2/measurements	neo_msgs/msg/USBoardV2
/usboard_v2/sensor1	sensor_msgs/msg/Range
/usboard_v2/sensor2	sensor_msgs/msg/Range
...	
/usboard_v2/sensor16	sensor_msgs/msg/Range

2.8.5 Hilfe

Wenn Sie den Fehler `unable to connect to port /dev/ttyUSB0` erhalten, führen Sie folgenden Befehl aus:

```
sudo usermod -a -G dialout $USER
```

und starten Sie Ihren PC neu.

2.9 Grafische Bedienoberfläche

Laden Sie die GUI hier herunter: [Windows⁴](#), [Linux⁵](#)

2.9.1 Einleitung

Das USBoard-USS5 wird zusammen mit einer grafischen Bedienoberfläche, kurz GUI (Graphical User Interface), ausgeliefert. Mit diesem Programm lassen sich die Einstellungen des USBoard-USS5 komfortabel ändern sowie die aktuellen Messwerte auslesen und anzeigen.

Für den Einsatz der GUI wird ein PC mit Microsoft Windows® oder Linux als Betriebssystem benötigt, der über einen freien USB-Anschluss verfügt. Für industrielle Anwendungen kann der USB-Anschluss auch durch eine klassische RS-232-Schnittstelle ersetzt werden. Der PC muss dann ebenfalls eine RS232-Schnittstelle (COM-Port) besitzen oder es muss ein USB-Seriell-Konverter verwendet werden.

Unter Windows wird für die GUI ein Installer geliefert. Führen Sie diesen durch einen Doppelklick aus und folgen Sie den Anweisungen. Der Installationsprozess sollte eine Verknüpfung auf Ihrem Desktop anlegen, über die Sie die GUI per Doppelklick starten können.

Die Linux-Version besteht aus Debian-Paketen für die neuesten Ubuntu LTS-Versionen. Installieren Sie das passende Paket mit:

```
sudo dpkg -i the_right_package.deb
```

wobei Sie `the_right_package` ersetzen mit der Datei, die zu Ihrem Betriebssystem passt. Nach der Installation starten sie die GUI mit folgendem Befehl:

```
/opt/neobotix/usboard2-gui/bin/usboard2_gui
```

⁴ https://neobotix-docs.de/files/USBoardUSS5-GUI_Windows.zip

⁵ https://neobotix-docs.de/files/USBoardUSS5-GUI_Linux.zip

2.9.2 Erste Schritte

2.9.2.1 Die Bedienoberfläche

Die Bedienoberfläche ist in drei Bereiche unterteilt: die Funktionsleiste ganz oben, den zentralen Datenbereich und die Fußzeile.

Die Funktionsleiste

Die Funktionsleiste ist in drei Bereiche gegliedert.

Ganz links finden Sie die Funktionen zum Aufbauen und Lösen der Datenverbindung zum USBoard-USS5.

Port: Hier können Sie die Schnittstelle auswählen, über die Sie mit dem USBoard-USS5 kommunizieren. Bei Einstecken des USB-Kabels wird automatisch ein virtueller COM-Port angelegt, den Sie hier auswählen können.

Connect Öffnet die ausgewählte Schnittstelle und baut eine Verbindung zum USBoard-USS5 auf.

Disconnect Trennt eine bestehende Verbindung.

Über die Schaltflächen in der Mitte können Sie USBoard-USS5-Konfigurationen lokal abspeichern bzw. diese Dateien wieder einlesen.

Save to File Speichert die aktuell in der GUI vorliegende Konfiguration auf dem Rechner.

Load from File Liest eine Konfigurationsdatei ein und trägt die Werte in die GUI ein.

Rechts befinden sich die Schaltfläche zum Übertragen der aktuellen Einstellungen zum USBoard-USS5 und zum Auslesen der im USBoard-USS5 vorliegenden Konfiguration.

Write to Board Schreibt die Parameter aus der GUI in den flüchtigen Speicher des USBoard-USS5, d.h. nach dem nächsten Neustart werden wieder die vorherigen Parameter aus dem EEPROM verwendet.

Read from Board Liest die momentan vom USBoard-USS5 verwendete Konfiguration aus und zeigt sie in der GUI an.

Write to EEPROM Schreibt die Parameter in den nicht-flüchtigen EEPROM-Speicher des USBoard-USS5, d.h. die Parameter werden auch nach einem Neustart verwendet.

Der Datenbereich

Der zentrale Datenbereich besitzt seinerseits drei Ansichten, die über die drei Reiter ausgewählt werden können.

Sensoren Hier werden die Messwerte der aktiven Sensoren visualisiert und die jeweiligen Warn- und Alarmgrenzwerte angezeigt.

Konfiguration In dieser Ansicht können die Kommunikationsparameter angepasst und die Sensoren konfiguriert werden.

Log Das Log listet die Systemmeldungen bei verschiedenen Aktionen auf.

Die Fußzeile

Hier finden Sie Informationen zum aktuellen Verbindungsstatus sowie die Seriennummer des Boards, mit dem Sie aktuell verbunden sind.

2.9.2.2 Messwerte anzeigen

Ultraschallsensoren

Die Ultraschallsensoren sind in Gruppen zu je vier Sensoren zusammengefasst, die entsprechend in vier identischen Feldern des Datenbereichs visualisiert werden.

Group 1

	Distance [cm]		Warn [cm]	Alarm [cm]
Sensor 1 <input type="checkbox"/>		N/A		
Sensor 2 <input type="checkbox"/>		N/A		
Sensor 3 <input type="checkbox"/>		N/A		
Sensor 4 <input type="checkbox"/>		N/A		

Die Kontrollkästchen direkt neben der Sensornummer zeigen an, ob dieser Sensor aktiv ist, also vom USBoard-USS5 angesteuert und ausgewertet wird.

Tipp: Wenn Sie nicht verwendete Kanäle deaktivieren, können Sie die Zykluszeit pro Durchlauf durch alle Sensoren reduzieren und dadurch schneller aktuelle Messdaten erhalten. Deaktivierte Kanäle werden vom USBoard-USS5 übersprungen, ohne auf ein Echo zu warten.

Ganz rechts werden für jeden Sensor der jeweilige Warn- und Alarmgrenzwert angezeigt. In der Mitte werden die aktuellen Messwerte als Farbbalken dargestellt und numerisch angezeigt.

Bemerkung: Die USS5-Sensoren besitzen eine untere und eine obere *Erfassungsgrenze* (Seite 6). Wenn ein Objekt detektiert wird, dass näher als die untere Erfassungsgrenze ist, wird anstelle des Messwerts die Meldung „BLOCKED“ angezeigt. Falls bis zur maximalen Reichweite überhaupt kein Objekt erkannt wird, wird die Meldung „TOO FAR“ ausgegeben

Analogeingänge

Analog

Input 1: <input type="text"/>	N/A	Input 2: <input type="text"/>	N/A	Input 3: <input type="text"/>	N/A	Input 4: <input type="text"/>	N/A
-------------------------------	-----	-------------------------------	-----	-------------------------------	-----	-------------------------------	-----

Die aktuellen Messwerte der vier Analogeingänge werden unterhalb der Anzeigen der Ultraschallsensoren dargestellt. Auch hier wird jeder Messwert numerisch und als Balkenanzeige ausgegeben. Die Balkenanzeige gibt dabei den Messwert relativ zur *maximalen Eingangsspannung* (Seite 6) an.

2.9.3 Konfiguration

2.9.3.1 Grundfunktionen

Sensoren konfigurieren

Die wichtigsten Einstellungen der Ultraschallsensoren können direkt in der Messwertanzeige vorgenommen werden.

Group 1

	Distance [cm]		Warn [cm]	Alarm [cm]
Sensor 1 <input type="checkbox"/>	<input type="text"/>	N/A	<input type="text"/>	<input type="text"/>
Sensor 2 <input type="checkbox"/>	<input type="text"/>	N/A	<input type="text"/>	<input type="text"/>
Sensor 3 <input type="checkbox"/>	<input type="text"/>	N/A	<input type="text"/>	<input type="text"/>
Sensor 4 <input type="checkbox"/>	<input type="text"/>	N/A	<input type="text"/>	<input type="text"/>

Aktivierung

Setzen Sie das Häkchen in der Auswahlbox, um den zugehörigen Sensor zu aktivieren. Deaktivierte Sensoren werden vom USBoard-USS5 nicht angesprochen, senden keinen Ultraschallpuls aus und es wird auch nicht auf ein Echo gewartet.

Tipp: Alle nicht verwendeten Sensorkanäle sollten deaktiviert werden, um die Zykluszeit für die Abfrage aller Sensoren zu reduzieren und schnellst möglich aktuelle Messwerte zu erhalten.

Wenn ein Kanal aktiviert ist aber kein Sensor angeschlossen wurde, wird dies von der GUI durch die Meldung „N/A“ anstelle des Messwerts angezeigt.

Schwellenwerte

Für jeden Sensor kann eine Warnungs- und eine Alarmschwelle festgelegt werden. Sobald der Messwert eines Sensors einen solchen Schwellenwert unterschreitet, wird das zugehörige Relais geschaltet. Die Relais bleiben jeweils so lange eingeschaltet, wie mindestens ein Sensor einen Messwert unterhalb seiner jeweiligen Schwelle liefert. Ein Schwellenwert von 0 deaktiviert die Funktion für diesen Sensor.

Es können nur ganzzahlige Werte in Zentimetern angegeben werden.

Mit Konfigurationen arbeiten

Mit den Schaltflächen in der Funktionsleiste können Sie Konfigurationen lokal abspeichern und später wiederverwenden und sie natürlich auch auf das USBoard-USS5 übertragen bzw. von dort auslesen, solange Sie verbunden sind.

Connection

Port: COM1 Connect Disconnect Save to File Load from File Write to Board Read from Board Write to EEPROM

Die meisten Funktionen sind selbsterklärend, einige Details sollten Sie aber beachten.

- Immer wenn Sie eine Konfiguration übertragen (aus einer Datei in die GUI, von der GUI auf das USBoard-USS5 oder anders herum), werden die auf der Empfängerseite vorhandenen Einstellungen überschrieben. Es gibt keine Rückgängig-Funktion.
- Mit Write to Board können Sie Einstellungen bequem testen. Nach einem Neustart des USBoard-USS5 wird aber wieder die im EEPROM gespeicherte Konfiguration verwendet.
- Eine mit Write to EEPROM übertragene Konfiguration wird im nicht-flüchtigen EEPROM-Speicher des USBoard-USS5 abgelegt und sofort aktiviert.
- Sie können mit Read from Board nur die aktuell verwendete Konfiguration auslesen. Diese kann sich von der im EEPROM hinterlegten Einstellung unterscheiden.

2.9.3.2 Komplexere Funktionen

Um weitergehende Einstellungen vorzunehmen, klicken Sie auf den Reiter Configuration, um die Ansicht des Datenbereichs umzuschalten.

Sensors **Configuration** Log

Global

CAN ID:

CAN Baudrate:

Transmit Mode:

Update Interval:

Low-Pass Gain:

CAN Termination: ☐

Enable Analog-In: ☐

Legacy Format: ☐

Warn Relay Mode: ☐

Alarm Relay Mode: ☐

Group 1

Resolution:

Fire Interval:

Sender:

Transmit: ☐

Cross-Echo: ☐

Group 2

Resolution:

Fire Interval:

Sender:

Transmit: ☐

Cross-Echo: ☐

Group 3

Resolution:

Fire Interval:

Sender:

Transmit: ☐

Cross-Echo: ☐

Group 4

Resolution:

Fire Interval:

Sender:

Transmit: ☐

Cross-Echo: ☐

Globale Einstellungen

Im Bereich Global ganz links können allgemeine Einstellungen vorgenommen werden.

CAN ID Die Basisadresse des USBoard-USS5 für die Kommunikation per CAN-Bus.

CAN Baudrate Die für den CAN-Bus zu verwendende Datenrate.

Transmit Mode Hier kann der Datenausgabemodus gewählt werden. Bei Request antwortet das USBoard-USS5 nur auf Anfragen, in allen anderen Fällen sendet es seine Messwerte automatisch über die ausgewählte Schnittstelle.

Update Interval Gibt die Zykluszeit an, mit der das USBoard-USS5 im automatischen Sendemodus Nachrichten verschickt.

Low-Pass Gain Der Tiefpassfilter kann genutzt werden, um die auszugebenden Messwerte zu glätten, oder um zu verhindern, dass Ausreißer sofort die Relais schalten, etwa durch einzelne fehlerhafte Messungen. Jeder neue Messwert eines Sensors wird entsprechend der Filtergewichtung mit dem aktuellen Ausgabewert verrechnet. Ein Filterwert von 0.3 bedeutet, dass der neue Ausgabewert zu 70% aus dem letzten Wert und zu 30% aus dem neuen Messwert berechnet wird. Ein Filterwert von 1.0 bedeutet, dass jeder neue Messwert unverändert als Ausgabewert übernommen, die Filterung also deaktiviert wird.

CAN Termination Beim USBoard-USS5-IP kann hier der CAN-Abschlusswiderstand aktiviert werden. Beim normalen USBoard-USS5 ist dieser Parameter ohne Funktion.

Enable Analog-In Hier kann die Auswertung der Analogeingänge aktiviert werden. Bei Anwendungen mit besonders kurzen Zykluszeiten und in denen die Eingänge nicht verwendet werden, kann es sinnvoll sein, die Analog-Digital-Wandlung komplett zu deaktivieren.

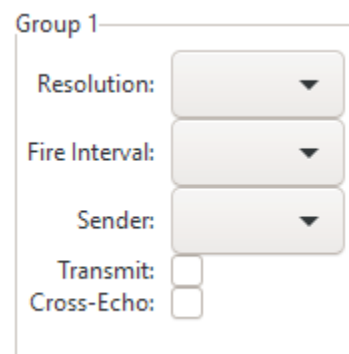
Legacy Format Hier kann auf das Datenformat des USBoard V1 umgeschaltet werden. Das USBoard-USS5 ist dann abwärtskompatibel zu älteren Anwendungen, büßt aber einige Funktionen ein.

Warn Relay Mode Wenn dieser Modus aktiv ist, haben blockierte Sensoren keine Auswirkung auf das Warnrelais. Unbeeinträchtigte Sensoren können das Warnrelais weiterhin abhängig von ihrem Messwert schalten. Verwenden Sie diesen Modus nur, wenn blockierte Sensoren nicht zu einer Reduzierung der Betriebssicherheit führen. Bei deaktivierter Funktion schaltet jeder blockierte Sensor das Warnrelais immer ein, unabhängig von den übrigen Messwerten. Verwenden Sie diesen Modus, um auf jeden blockierten Sensor zu reagieren.

Alarm Relay Mode Gleiche Funktion wie bei Warn Relay Mode, aber auf das Alarmrelais wirkend.

Einstellungen für Sensorengruppen

Die Ultraschallsensoren sind in Gruppen zu je vier Sensoren zusammengefasst. Die Einstellungen in diesem Bereich gelten immer für alle Sensoren einer Gruppe, die verschiedenen Gruppen können aber völlig individuell konfiguriert werden.



Group 1

Resolution:

Fire Interval:

Sender:

Transmit: ☐

Cross-Echo: ☐

Resolution Die Messdaten können in vier unterschiedlichen Auflösungen ausgegeben werden: 1 cm, 5 mm, 2,5 mm oder 1,25 mm.

Fire Interval Die Zeit zwischen den Ultraschallimpulsen zweier Sensoren kann angepasst werden, um unerwünschte Echos zu vermeiden. Für die komplette Auswertung eines Sensors sind aber mindestens 10 ms erforderlich. Diese Minimalzeit bietet zwar die kürzeste Zykluszeit, erlaubt aber nur Messungen bis 1,5 m. Bei größeren

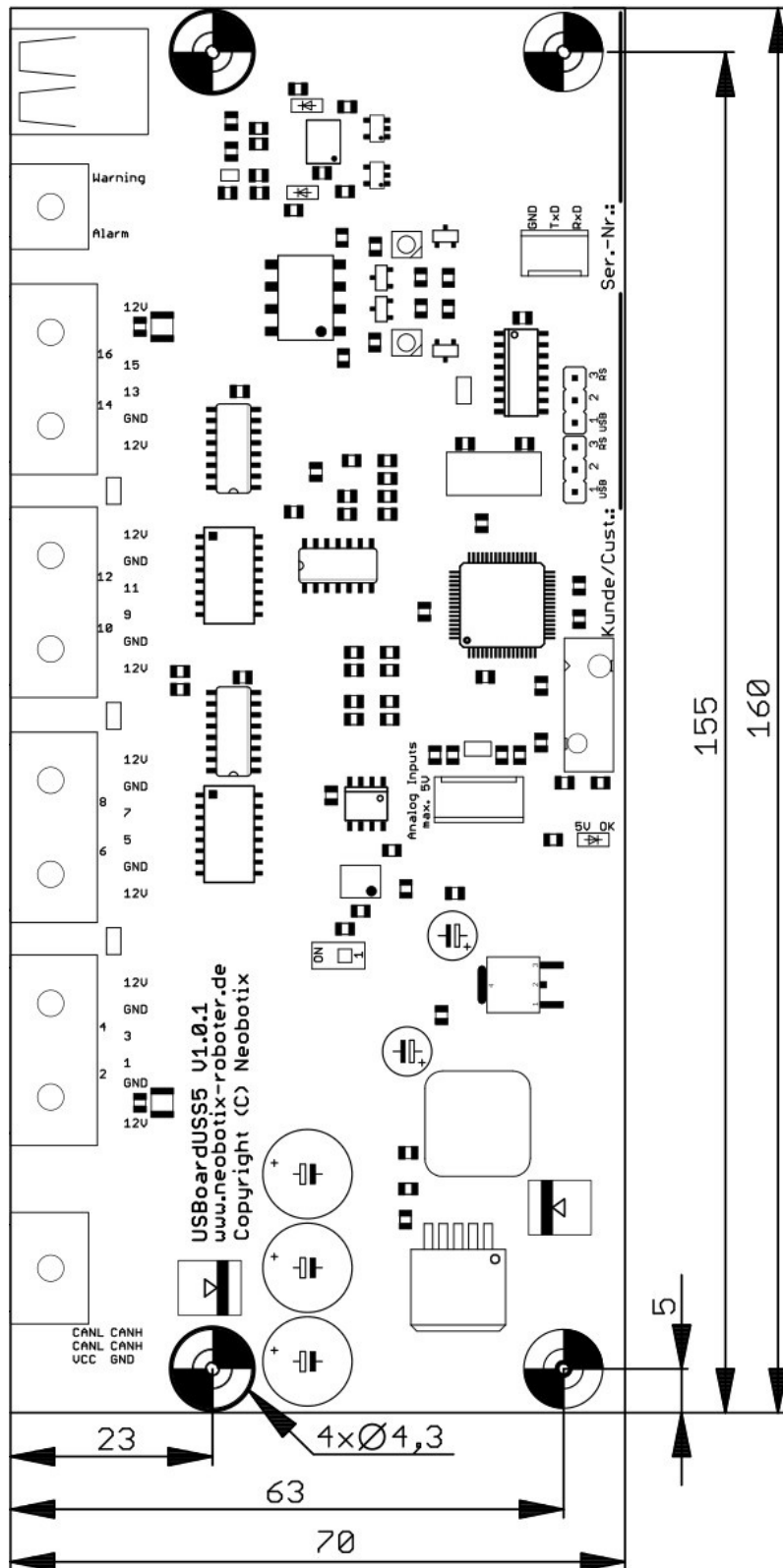
Abständen steigt die Schalllaufzeit über die verfügbare Messdauer. Um den kompletten Messbereich von 2,5 m abzudecken, beträgt die Werkseinstellung 20 ms.

Sender Hier kann für den Betrieb im Kreuzechomodus ausgewählt werden, welcher Sensor den Puls erzeugen soll.

Transmit Gibt an, ob die Mesdaten dieser Gruppe ausgegeben werden sollen.

Cross-Echo Schaltet die Gruppe vom normalen Direktechomodus auf Kreuzechobetrieb um.

2.10 Abmessungen



Bemerkung: Schieben Sie S1 auf Position ON, um den CAN-Abschlusswiderstand zu aktivieren.

Achtung: Die Befestigungsbohrungen in der Nähe des Versorgungssteckers X1 sind elektrisch mit der Massefläche des USBoard-USS5 verbunden. Bitte isolieren Sie diesen Befestigungspunkt, wenn Ihre Installation es erfordert.

Das USBoard-USS5 kann in beliebiger Orientierung montiert werden.

2.11 Steckerbelegung

Bemerkung: Weitere Informationen zu den verwendeten Steckverbindern finden Sie unter *Steckverbinder* (Seite 39).

2.11.1 X1

Würth Elektronik MPC3, 2-reihig, 6-polig

Pin	Beschreibung
1	Versorgungsspannung
2	CAN High
3	CAN High
4	Masse
5	CAN Low
6	CAN Low

Pins 2 und 3 bzw. 5 und 6 sind jeweils verbunden. Dadurch können zwei CAN-Leitungen (CAN-in und CAN-out) komfortabel angeschlossen werden.

2.11.2 X2 - X5

Würth Elektronik MPC3, 2-reihig, 12-polig

Pin	Beschreibung
1	Sensor-Versorgungsspannung
2	Masse
3	Sensordatenverbindung (X2: Sensor 2, X3: Sensor 6, X4: Sensor 10, X5: Sensor 14)
4	Sensordatenverbindung (X2: Sensor 4, X3: Sensor 8, X4: Sensor 12, X5: Sensor 16)
5	Masse
6	Sensor-Versorgungsspannung
7	Sensor-Versorgungsspannung
8	Masse
9	Sensordatenverbindung (X2: Sensor 1, X3: Sensor 5, X4: Sensor 9, X5: Sensor 13)
10	Sensordatenverbindung (X2: Sensor 3, X3: Sensor 7, X4: Sensor 11, X5: Sensor 15)
11	Masse
12	Sensor-Versorgungsspannung

2.11.3 X6

Würth Elektronik MPC3, 2-reihig, 4-polig

Pin	Beschreibung
1	Relaisausgang Alarm (Schließer)
2	Relaisausgang Warnung (Schließer)
3	Relaisausgang Alarm (Schließer)
4	Relaisausgang Warnung (Schließer)

2.11.4 X7

TE Connectivity HE14, 1-reihig, 4-polig

Pin	Beschreibung
1	Analogeingang 1
2	Analogeingang 2
3	Analogeingang 3
4	Analogeingang 4

2.11.5 USS5

Molex MX64, 3-polig, key A

Pin	Beschreibung
1	Versorgung
2	Signal
3	Masse

2.12 Zukaufteile

2.12.1 Steckverbinder

Eine Übersicht der verwendeten Steckverbinder finden Sie unter [Steckverbinder](#) (Seite 39).

2.12.2 USBoard-USS5-Konfigurationskabel

Der Basiskabelsatz (Best.-Nr. X211) ermöglicht eine schnelle und einfache Inbetriebnahme und erste Tests des USBoard-USS5. Er besteht aus:

- Stecker X1: Anschluss für Versorgungsspannung (1 m, offene Kabelenden, Rot = Plus, Schwarz = Minus) und CAN (1 m, D-Sub 9 Buchsenstecker, kein Abschlusswiderstand, Pin 2 - CAN Low, Pin 7 - CAN High)
- USB-Kabel: 1,5 m, USB 2.0, A-Stecker auf B-Stecker

Mit den unter [Steckerbelegung](#) (Seite 31) beschriebenen Steckverbindern kann er auch ohne Weiteres selbst hergestellt werden.



Abb. 1: USBBoard-USS5-Konfigurationskabel (X211)

2.12.3 Sensorkabelsätze

An das USBBoard-USS5 können vier Sensorkabelsätze für jeweils vier Sensoren angeschlossen werden.

Für erste Tests und Prototypen können diese Kabelsätze mit ungeschirmten Leitungen in verschiedenen Längen bei Neobotix bezogen werden. Die Standardvariante (Bestellnummer X214) ist 2 m lang und passt an alle vier Sensoranschlüsse X2 bis X5 des USBBoard-USS5. Andere Kabellängen (bis zu 25 m) und -konfigurationen sind auf Anfrage möglich.

Für die spätere Anwendung sind individuell gefertigte Kabelsätze empfehlenswert.

2.12.4 Steckverbindersätze

Falls Sie Ihre Anschlusskabel selbst fertigen möchten, aber die passenden Steckverbinder brauchen, bieten wir komplette Sets mit Crimpkontakten und Steckergehäusen für das USBBoard-USS5 und für die Ultraschallsensoren USS5 an. Die Kontakte lassen sich auch mit Universalcrimpzangen gut genug verarbeiten, um damit erste Tests durchzuführen und Prototypen auszustatten.

Das Set für das USBBoard-USS5 (Bestellnummer X212) enthält die Crimpkontakte und Steckergehäuse für alle Anschlüsse eines USBBoard-USS5 mit Ausnahme des USB-Kabels.

Das Set für die Ultraschallsensoren (Bestellnummer X213) enthält die Kontakte und das Steckergehäuse für jeweils einen Sensor.

2.12.5 Ultraschallsensoren

Die Ultraschallsensoren Bosch USS5.0 mit radialem Steckerabgang sind unter Bestellnummer X210 verfügbar.



Abb. 2: Kabelsatz für 4 x USS5-Sensoren (X214)



Abb. 3: Steckersatz für USBoard-USS5 (X212)

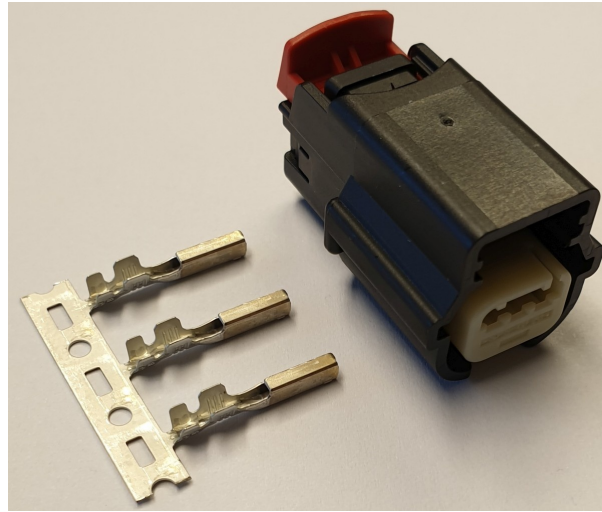


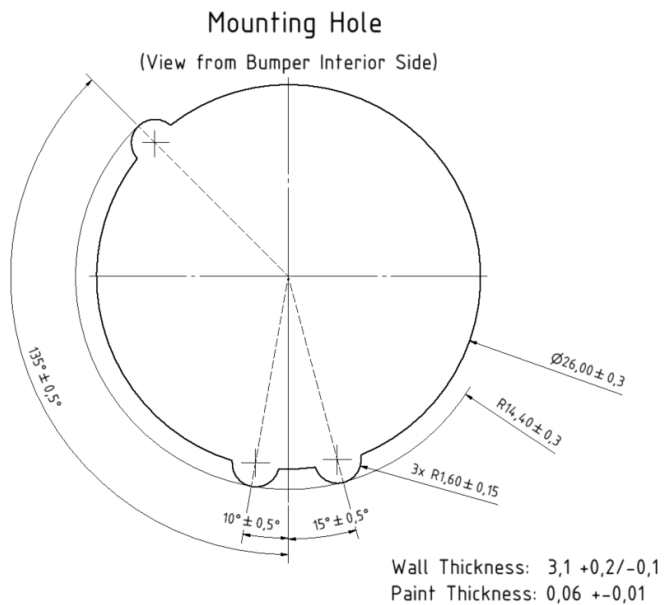
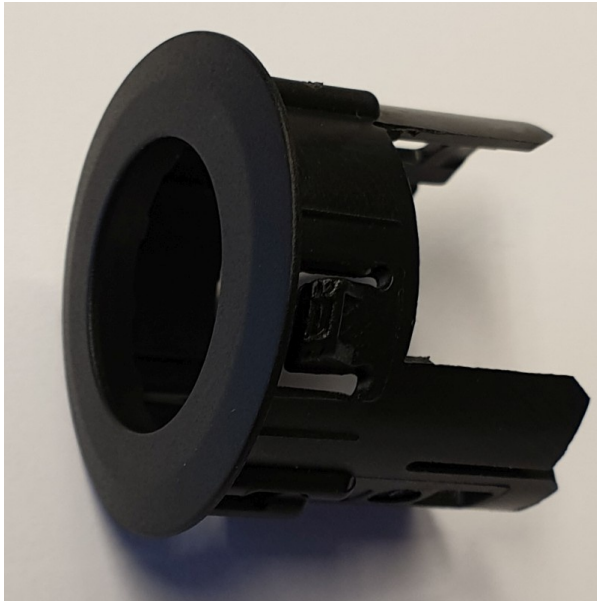
Abb. 4: Steckersatz für USS5 (X213)



Abb. 5: Ultraschallsensor Bosch USS5.0 (X210)

2.12.6 USS5-Einbaugehäuse

Mit dem USS5-Einbaugehäuse (Bestellnummer X216) kann der Sensor an einer Frontplatte befestigt werden.



2.13 Häufige Fragen

Bemerkung: Diese Seite wird laufend ergänzt, wenn wir Rückmeldung von unseren Kunden bekommen. Falls Ihre spezielle Frage hier noch nicht zu finden ist, sehen Sie sich bitte auf den anderen Seiten dieser Dokumentation um oder sprechen Sie uns an. Wir helfen Ihnen gerne!

2.13.1 Allgemeine Fragen

2.13.1.1 Wie funktioniert das USBoard-USS5?

Sowohl die grundlegende Funktionsweise von Ultraschallsensoren als auch der Kreuzechomodus des USBoard-USS5 werden unter *Funktionsweise* (Seite 4) detailliert beschrieben.

2.13.1.2 Die Messwerte sind nicht wie erwartet.

Leider ist dies eine Folge des physikalischen Messprinzips. Anders als zum Beispiel Laserlicht werden Ultraschallwellen leicht gestört oder abgelenkt, was in manchen Situationen zu falschen Messwerten führt. Zudem reagieren sie auf verschiedene Materialien und Oberflächen teils sehr unterschiedlich. Das ist der Grund, warum das USBoard-USS5 kein Sicherheitssystem ist.

Bitte sehen Sie sich unsere *FOV tests* (Seite 7). Wir empfehlen dringend, eigene Tests und Experimente mit dem System zu machen, bevor es produktiv eingesetzt wird.

2.13.1.3 Wie kann ich die Messdaten möglichst schnell erhalten?

Wenn Sensorkanäle deaktiviert werden, dann wird dort kein Ultraschallpuls erzeugt und auch nicht auf ein Echo gewartet. Dadurch kann die Zykluszeit des Gesamtsystems reduziert werden und die Messwerte aller real vorhandenen Sensoren stehen schneller zur Verfügung. Weitere Informationen finden Sie in der Beschreibung des Parametersatzes unter *Parametersatz* (Seite 9) und zum Befehl `CMD_SET_CHANNEL_ACTIVE` unter *Befehle* (Seite 13).

Bei einigen Anwendungen kann es auch sinnvoll sein, den *Kreuzechomodus* (Seite 5) zu verwenden.

2.13.2 Verbindungsprobleme

2.13.2.1 Fehler auf dem CAN-Bus

Bitte achten Sie darauf, dass mindestens ein Abschlusswiderstand vorhanden ist. Die empfohlene Konfiguration sieht je einen 120 Ohm Widerstand zwischen CAN High und CAN Low an jedem physikalischen Ende des Busses vor.

Das USBoard-USS5 bietet einen *kleinen Schalter* (Seite 30) mit dem der integrierte Abschlusswiderstand aktiviert werden kann. Um den Abschlusswiderstand des USBoard-USS5-IP zu aktivieren, verwenden Sie bitte die entsprechende Einstellung in der GUI.

2.13.2.2 Die GUI verbindet sich nicht mit dem USBoard-USS5

Bei einer direkten USB-Verbindung prüfen Sie bitte die folgenden Punkte:

- Wurde der richtige Port ausgewählt? Einige Computer besitzen interne COM Ports, die mit dem vom USBoard-USS5 erzeugten virtuellen Port verwechselt werden können. Probieren Sie alle erkannten Ports aus.
- Hat sich die Port-Nummer geändert? Verschiedenen USBoard-USS5 weist das Betriebssystem manchmal unterschiedliche Ports zu. Bitte starten Sie die GUI neu, um deren Port-Liste zu aktualisieren.

2.14 Rechtliche Anmerkungen

Die allgemeinen rechtlichen Anmerkungen finden Sie unter *Rechtliche Anmerkungen* (Seite 2).

2.14.1 EG-Konformitätserklärung



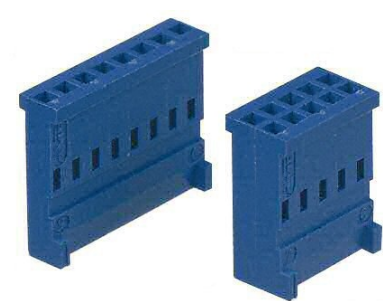
Hiermit bestätigt Neobotix, dass das beschriebene Produkt die relevanten EU-Richtlinien erfüllt.

2.14.2 Informationen zu RoHS



Hiermit bestätigt Neobotix, dass das beschriebene Produkt die RoHS-Richtlinien 2011/65/EU (RoHS 2) und 2015/863/EU des Europäischen Parlaments und des Rates zur Beschränkung der Verwendung gefährlicher Stoffe in Elektro- und Elektronikgeräten erfüllt.

3.1 TE Connectivity - HE14

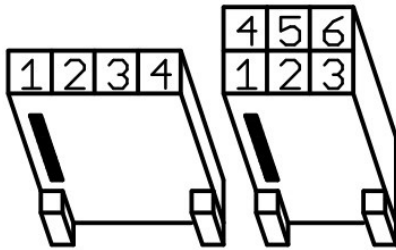


Pole	TE Connectivity	Farnell	RS Components
3-polig, 1-reihig	281838-3	429582	532-333
4-polig, 1-reihig	281838-4	429594	532-349
5-polig, 1-reihig	281838-5	429600	532-355
6-polig, 2-reihig	281839-3	429650	532-406
8-polig, 2-reihig	281839-4	429661	532-412
10-polig, 2-reihig	281839-5	429673	532-428
12-polig,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



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

Bei Neobotix-Produkten ist die Pinbelegung der HE14-Stecker wie unten dargestellt.



3.2 Würth Elektronik - MPC4

Für nähere Informationen zum MPC4⁶ konsultieren Sie bitte den Würth Elektronik Onlinekatalog⁷.



Pole (2-reihig)	Würth Elektronik
2	649002113322
4	649004113322
6	649006113322
8	649008113322
10	649010113322
16	649016113322

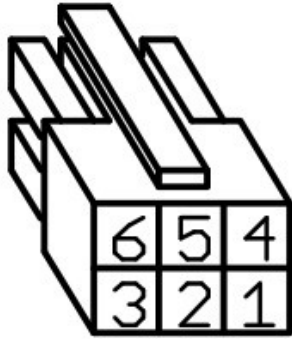


Crimpkontakte	Würth Elektronik
AWG 24-18	64900613722

Bei Neobotix-Produkten ist die Pinbelegung der MPC4-Stecker wie unten dargestellt.

⁶ https://www.we-online.de/katalog/de/em/connectors/wire-to-board/wr_mpc4/

⁷ <https://www.we-online.com/de/produkte/bauelemente/uebersicht>



3.3 Würth Elektronik - MPC3

Für nähere Informationen zum MPC3⁸ konsultieren Sie bitte den [Würth Elektronik Onlinekatalog](#)⁹.



Pole (2-reihig)	Würth Elektronik
4	662004113322
6	662006113322
12	662012113322

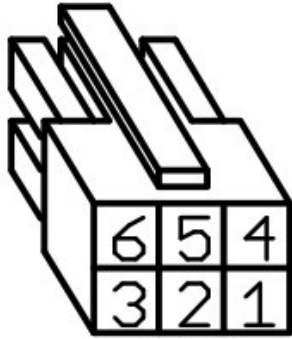


Crimpkontakte	Würth Elektronik
AWG 24-20	66200113722

Bei Neobotix-Produkten ist die Pinbelegung der MPC3-Stecker wie unten dargestellt.

⁸ https://www.we-online.de/katalog/de/em/connectors/wire-to-board/wr_mpc3/

⁹ <https://www.we-online.com/de/produkte/bauelemente/uebersicht>



Sachkundiges Personal

Dieses Produkt darf nur von sachkundigem Personal umgebaut, in Betrieb genommen und betrieben werden. Sachkundig ist, wer

- aufgrund seiner fachlichen Ausbildung und Erfahrung ausreichende Kenntnisse auf den für die jeweilige Arbeit erforderlichen Gebieten der Mechatronik und Robotik hat,
- vom Betreiber der Maschine in der Bedienung und den gültigen Sicherheitsrichtlinien unterwiesen wurde,
- mit den einschlägigen staatlichen Arbeitsschutzvorschriften, Unfallverhütungsvorschriften, Richtlinien und allgemein anerkannten Regeln der Technik (z. B. DIN-Normen, VDE-Bestimmungen, technische Regeln) so weit vertraut ist, dass er den arbeitssicheren Zustand des Produkts beurteilen kann und
- Zugriff auf diese Unterlagen und hat und diese gelesen hat.

Als nicht-sachkundig gelten im Allgemeinen, aber nicht ausschließlich:

- Nicht mit dem Produkt vertraute Praktikanten oder Mitarbeiter,
- Besucher und Gäste,
- alle Mitarbeiter anderer Abteilungen des Unternehmens oder der Einrichtung, in der das Produkt betrieben wird.

Diese Liste ist nicht abschließend.