USER MANUAL

Force Sensor DAQs
with USB, UART or CAN Interfaces

Edition E2

OnRobot DAQ Version 5.3

June 2018

Content

Copyright © 2017-2018 OnRobot A/S. All rights Reserved. No part of this publication may be reproduced, in any form or by any means, without the prior written permission of OnRobot A/S.

Information provided within this document is accurate to the best of our knowledge at the time of its publication. There may be differences between this document and the product if the product has been modified after the edition date.

OnRobot A/S. does not assume any responsibility for any errors or omissions in this document. In no event shall OnRobot A/S. be liable for losses or damages to persons or property arising from the use of this document.

The information within this document is subject to change without notice. You can find the latest version on our webpage at: **https://onrobot.com/**.

The original language for this publication is English. Any other languages that are supplied have been translated from English.

All trademarks belong to their respective owners. The indications of (R) and TM are omitted.

1 Preface

1.1 Target Audience

This document is intended for persons working with OnRobot sensors and DAQs. Personnel working with the DAQ are expected to have the following expertise:

A. Basic knowledge of electronic and electrical systems

1.2 Intended Use

The DAQs are designed for gathering data from OnRobot sensors. OnRobot is not liable for any damage or injury resulting from misuse.

1.3 Important safety notice

The DAQ is *partly completed machinery* and a risk assessment is required for each application the DAQ is a part of. It is important that all safety instructions herein are followed. The safety instructions are limited to the sensor only and do not cover the safety precautions of a complete application.

The complete application must be designed and installed, in accordance with the safety requirements specified in the standards and regulations of the country where the application is installed.

1.4 Warning Symbols

DANGER:

This indicates a very dangerous situation which, if not avoided, could result in injury or death.

WARNING:

This indicates a potentially hazardous electrical situation which, if not avoided, could result in injury or damage to the equipment.

WARNING:

This indicates a potentially hazardous situation which, if not avoided, could result in injury or major damage to the equipment.

CAUTION:

This indicates a situation which, if not avoided, could result in damage to the equipment.

NOTE:

This indicates additional information such as tips or recommendations.

1.5 Typographic Conventions

The following typographic conventions are used in this document.

Table 1: Conventions

`Courier Text`	File paths and file names, code, user input and computer output.
Italicized text	Citations and marking image callouts in text.
Bold text	UI elements, including text appearing on buttons and menu options.
Bold, blue text	External links, or internal cross-references.
<angle brackets>	Variable names that must be substituted by real values or strings.
1. Numbered lists	Steps of a procedure.
A. Alphabetical lists	Image callout descriptions.

2 Getting Started

There are three DAQ (Data Acquisition) types and all are covered in this document:

- Single-channel for 3-axis force sensor (code name “31”)
- Multi-channel for 3-axis force sensors (code name “34”)
- Single-channel for 6-axis force/torque sensor (code name “64”)

It is recommended that you first identify which DAQ type is used.

There are three interface types that all use the same protocol (same bytes have to be transmitted and received in order to communicate with the device):

- USB (CDC - virtual serial port)
- UART (3.3V TTL Rx/Tx with 1M Baud)
- CAN (Standard CAN according to the ISO 11898)

> This document covers OnRobot proprietary protocols only.
>
> For Ethernet or EtherCAT EDS (Electronic Data Sheet) files contact OnRobot.

2.1 USB and UART

Both USB and UART are serial ports with the following communication settings:

Baud rate:	1 000 000
Stop bit:	1
Parity:	none
Data bits:	8
Flow control:	none

In this case, bidirectional communication can be used to read and write to the device.

2.2 CAN

The CAN interface uses two CAN channels for this purpose:

- 0x100 (default) for reading (Rx) to the device
- 0x101 (default) for writing (Tx) to the device

The CAN channel IDs (11bits) can be reconfigured any time. The process of the reconfiguration is discussed in **CAN ID Reconfiguration**.

3 Using the URCap Plugin

Once the device is powered on, it cyclically (default rate is 100Hz) starts to transfer a frame of 16, 34 or 22 bytes:

- 16 bytes for a single-channel 3 axis force sensor

Data packet bytes						
1 2 3 4	5 6	7 8	9 10	11 12	13 14	15 16
Header (170,7,8,10)	Sample Counter	Status	Fx	Fy	Fz	Checksum

- 34 bytes for a multi-channel 3 axis force sensor (4 channel is supported)

Data packet bytes										
1 2 3 4	5 6	7 8	9 10	11 12	13 14	…	27 28	29 30	31 32	33 34
Header (170,7,8,28)	Sample Counter	Status	Fx 1	Fy 1	Fz 1	…	Fx 4	Fy 4	Fz 4	Checksum

- 22 bytes for a single-channel 6 axis force/torque sensor

Data packet bytes									
1 2 3 4	5 6	7 8	9 10	11 12	13 14	15 16	17 18	19 20	21 22
Header (170,7,8,16)	Sample Counter	Status	Fx	Fy	Fz	Tx	Ty	Tz	Checksum

The force (or torque) values are in dimensionless counts that can be converted to N (or Nm) by using the values in the sensitivity report.

For example:

- Reading from Fx = 532 [Counts]
- Sensitivity (Counts@N.C.) = 6100
- Nominal Capacity for Fx(N.C.) =150N

```
Fx [N] = Fx[Counts] / (Counts@N.C.) * (N.C.) = 532 / 6100
* 150N = 13.08N
```

The Sample counter is a UINT16 type value that is incremented each time the DAQ completes an internal sampling (fixed at 1 kHz).

The Status UINT16 indicates the current status of the sensor. The detailed description is described in **Status**.

The Checksum (UINT16) can be used to check the integrity of the packet. The checksum is a sum of the preceding bytes starting from the header (170 + 7+ 8 + …).

> The force (and torque) values are all signed INT16 values.

4 Configuration

The sensor can be configured by sending a 9 byte Configuration packet.

Configuration packet bytes								
1	2	3	4	5	6	7	8	9
Header (170,0,50,3)				Speed	Filter	Zero	Checksum	

> For CAN interface, a full frame (8 byte) and a partial frame (1 byte) should be sent.
>
> If only full frames can be sent, please sure that the last byte of the Checksum Word are sent as the first byte of the second full frame.

The Checksum (UINT16) needs to be calculated according to the value of the Speed, Filter and Zero.

```
Checksum=170+0+50+3+Speed+Filter+Zero
```

The Speed byte sets the update speed.

Speed (decimal)	Update frequency
0	Stops the transmission
1	1000 Hz
3	333 Hz
10	100 Hz (default)
33	30 Hz
100	10 Hz

The internal filtering can be configured by setting the Filter byte.

Filter (decimal)	Cut-off frequency
0	No filtering
1	500 Hz
2	150Hz
3	50 Hz
4	15 Hz (default)
5	5 Hz
6	1.5 Hz

In order to clear the sensors offset, the sensor can be zeroed by setting the ZERO byte to 255 (decimal). It is restored to the original values by setting it to 0.

If the device has been set to zero and needs to be re-zeroed at a later time, first send 0 as the ZERO byte, wait at least 2ms and send 255 again.

Example:

In order to set 1000 Hz update rate, 500Hz cut-off frequency and cancel the offset (by zeroing) the 16 bytes to be sent should be the following:

```
170  0  50  3  1  1  255   1  224
```

> Please note that the configuration will reset on power reset.

After sending any packet to the device (for example, the Configuration packet), the device acknowledges by replying with the content of the error register (80). Therefore, the user can monitor whether the command was received or not.

If there were no errors, 0 is sent as the content of the error register:

170 0 80 1 **0** 0 251

The last two bytes are the checksum of this packet.

5 Status

The description of the Status word is the following:

Status Word

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

ERROR		**OVERLOAD**						**Multiple sensor selection**	
DAQ type	**Sensor type**	**Fx**	**Fy**	**Fz**	**Tx**	**Ty**	**Tz**	**Single/Multiple**	**Sensor number**
000 = No error 001 = DAQ error 010 = Communication error 011 = Reserved 100 = Reserved 101 = Reserved 110 = Reserved 111 = Reserved	000 = No error 001 = The sensor has not been detected 010 = Sensor failure 100 = Temperature error 011 = Reserved 101 = Reserved 110 = Reserved 111 = Reserved	0 = The axis has not been overloaded 1 = The axis has been overloaded			These are not used in case of force only sensors			0 = Only a single sensor has error (or no error) 1 = Multiple sensors have error /This case, only the last sensor number has been indicated/	000 = No sensor has error 001 = Sensor #1 010 = Sensor #2 011 = Sensor #3 100 = Sensor #4 101 = Reserved 110 = Reserved 111 = Reserved

Example: a decimal 514 equal to 0b0000001000000010 would imply an overload condition of axis Fx of the sensor #2 (channel 2).

6 CAN ID Reconfiguration

The CAN address can be changed any time via the following two standard frames:

CAN standard frames										
IDs (11bit)	Length (bytes)	Frame number	BYTES							
			0	1	2	3	4	5	6	7
Sensor_ID_Rx	8	1	Header (170,0,60,8)				Sensor_ID_Rx		Sensor_ID_Tx	
	6	2	EEPROM ('S','A','V','E')				Checksum			

New IDs can be given by assigning `Sensor_ID_Rx` and `Sensor_ID_Tx` (that can be identical if there is no available CAN address) as UINT16 variables.

To store the new IDs to the EEPROM of the DAQ, the EEPROM field of the packet must contain the 'S','A','V','E' ASCII characters (83,65,86,69), otherwise the new ID will not be saved.

After the new IDs are stored to the EEPROM, a power cycling (power off and power on) is required for the change to take effect.

The Checksum (UINT16) needs to be updated according to the value of the `Sensor_ID_Rx` and `Sensor_ID_Tx`:

```
Checksum=170+0+60+8+HighByte(Sensor.ID.Rx)+LowByte(Sensor.ID.R
x) + HighByte(Sensor.ID.Tx)+LowByte(Sensor.ID.Tx)+83+65+86+69
```

Example

To store a new ID of 0x103 for Tx and 0x104 for Rx the following (14) bytes have to be sent:

170 000 060 008 001 004 001 003

083 065 086 069 **002 038**

All are decimal numbers, the leading zeros are only shown for visualization.

7 Digitalizer (DAQ)

OnRobot DAQ has a standard USB Mini-B plug that is used to operate the DAQ in USB mode. The External Data Interface (8 pin) is used to communicate via CAN or UART.

The SPI interface (shares pins with the UART) is obsolete.

The External Data Interface is the 8 pin terminal block (highlighted in gray):

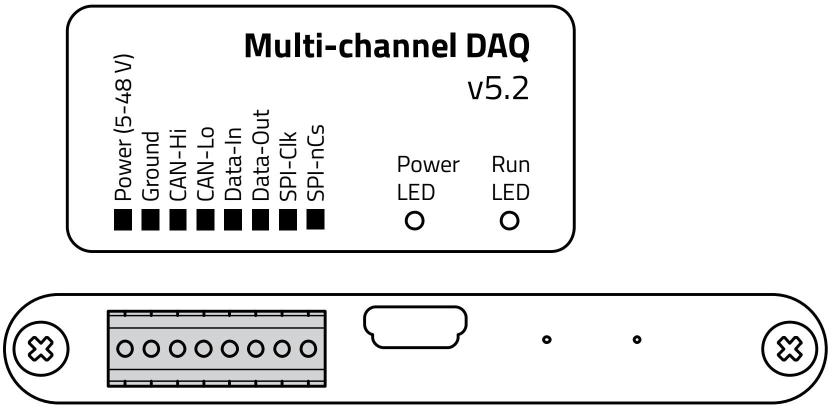


Table 2: Pin names and descriptions

Name	Description
POWER	This is the power supply pin for the DAQ if it is not powered via USB. The power supply voltage can be 5V-48V.
GROUND	The power and data signal ground pin.
CAN-HI	CAN bus High pin
CAN-LO	CAN bus Low pin
DATA-IN	In UART mode it is the RX pin. In SPI mode it is the Data input pin.
DATA-OUT	In UART mode it is the TX pin. In SPI mode it is the Data output pin.
SPI-CLK	In SPI mode it is the clock pin.
SPI-nCS	In SPI mode it is the Chip Select (active-low).

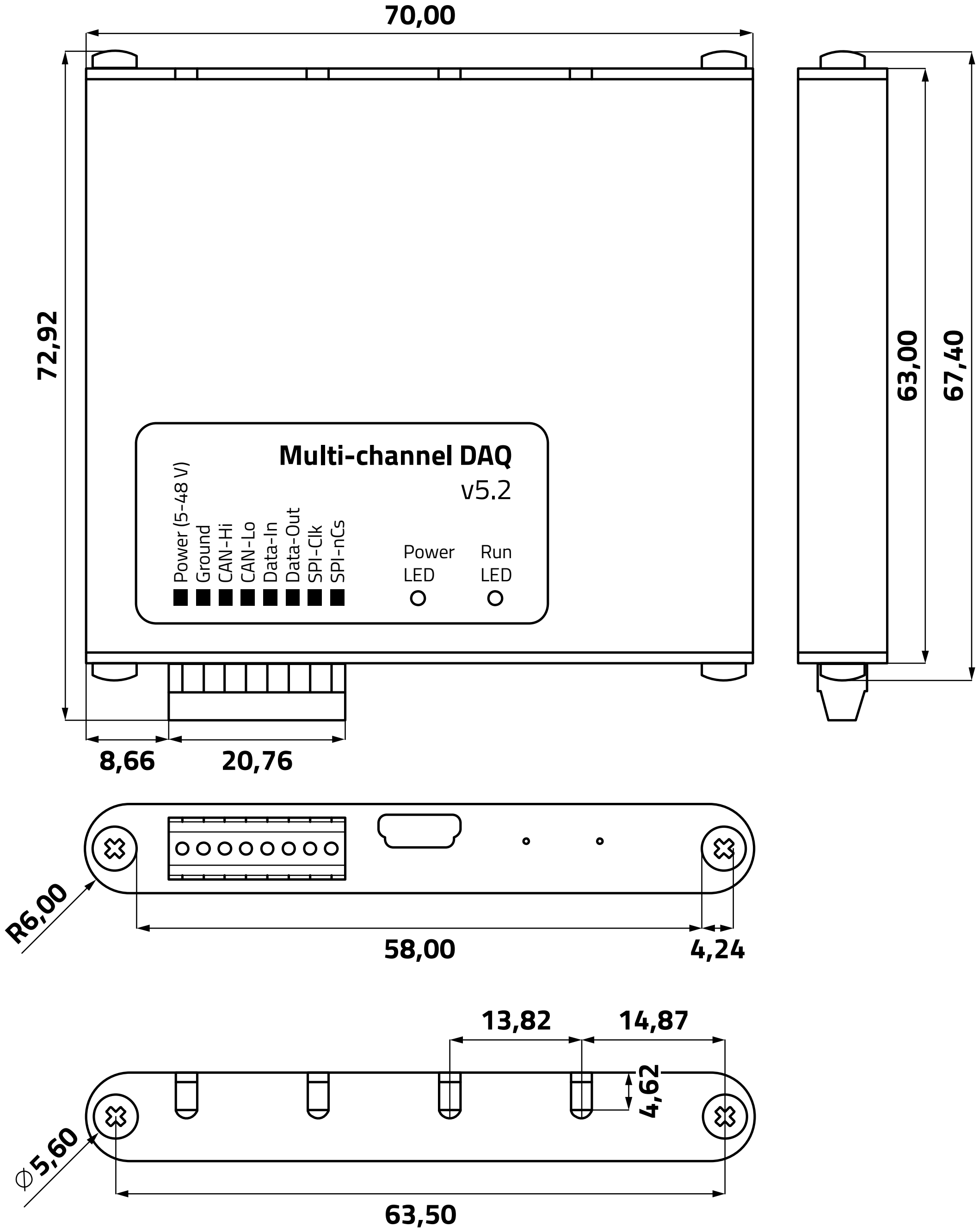
> The Rx and Tx signals are TTL 3.3V and cannot tolerate any voltage under 0.6V or above 3.6V.
>
> Any voltage outside these boundaries can cause damage to the pins. (Compared to the RS-232 that can have +/-12V signal voltage level.)

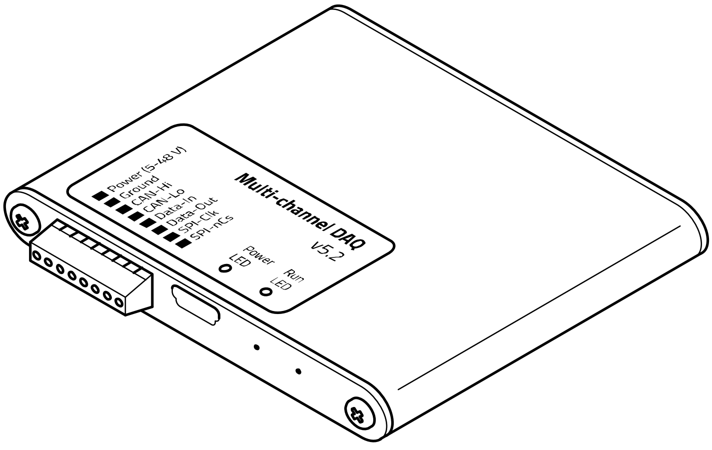
The UART needs the following configuration:

Baud rate	1 000 000
Stop bit	1
Parity	none
Data bits	8
Flow control	none

8 Mechanical Drawing of the DAQ

All dimensions are in mm.




Power (5-48 V)
Ground
CAN-Hi
CAN-Lo
Data-In
Data-Out
SPI-Clk
SPI-nCs
Multi-channel DAQ
v5.2
Power LED
Run LED

9 List of Acronyms

Acronym	Expansion
DHCP	Dynamic Host Configuration Protocol
DIP	dual in-line package
F/T	Force/Torque
ID	Identifier
IP	Internet Protocol
IT	Information technology
MAC	media access control
PC	Personal Computer
RPY	Roll-Pitch-Yaw
SP	Starting Position
SW	software
TCP	Tool Center Point
UR	Universal Robots
URCap	Universal Robots Capabilities
USB	Universal Serial Bus
UTP	unshielded twisted pair

10 Appendix

10.1 Editions

Edition	Comment
Edition 2	New Look&Feel Document restructured. List of Acronyms added. Appendix added. Target audience added. Intended use added. Copyright, Trademark, contact information, original language information added.