



Force Torque Sensor FTS-LWA

User Guide

User Guide Force Torque Sensor FTS-LWA

Version 2014/06

SWPC V04-022, HWE ADW V13, SWMC V0x51 0x14, HWM V04, DOC V05

© Commonplace Robotics GmbH, 2014

Commonplace Robotics GmbH

Osterfeldstr. 1

D-49326 Melle

Germany

++49 5429 / 374983-4

info@cpr-robots.com

www.cpr-robots.com



The CE sign confirms that this product meets the requirements of the directive 2004/108/EC (EMC) and 2002/95/EC (RoHS). The according documentation is deposited at the manufacturer.



Help to save our environment! When reaching the end of lifetime, do not throw the device into the garbage, but bring it to a public recycling place.

Contents

- 1. Safety Instructions..... 5
- 2. Introduction 6
 - 2.1 Product..... 6
 - 2.2 Specifications..... 7
- 3. Interfaces..... 8
 - 3.1 Dimensions and Orientation..... 8
 - 3.2 Electrical Connectors 8
- 4. CAN Protocol..... 9
- 5. Software..... 12
 - 5.1 Driver Installation for the Peak System PCAN USB Adapter 12
 - 5.2 Test Software SensorViz 12
- 6. C++ Sample Code 14
- 7. Troubleshooting 15
 - 7.1 Error Codes 15
 - 7.2 Service Contact 15

1. Safety Instructions



- The sensor must not be used for safety critical or medical applications.
- The sensor transmits an error code with overload and error information. To prevent damages the application must react with error strategies if these codes are set.
- The sensor has been designed for indoor use. It has to be protected of humidity and dust.
- The sealed areas of the sensor must not be opened. When opening the sealed screws the calibration and warranty is void.
- It is necessary to backup important data before the installation of the test software.

2. Introduction

2.1 Product

The FTS-LWA force torque sensor is a small and compact sensor with integrated electronics. It is developed to be used with the Schunk LWA robots:

- Integrated electronics, no need for external A/D cards
- Integrated Schunk Adapter to directly integrate into the LWA
- Communication using the second integrated CAN bus
- 8 signal looped through the sensor to avoid external wiring



Pic. 1: The FTS-LWA force torque sensor

A windows test program and C++ sample code is provided with the sensor.

The sensor specific calibration matrix can be found written on the quality management sheet, in electronic form in the configuration file of the test software SensorViz.

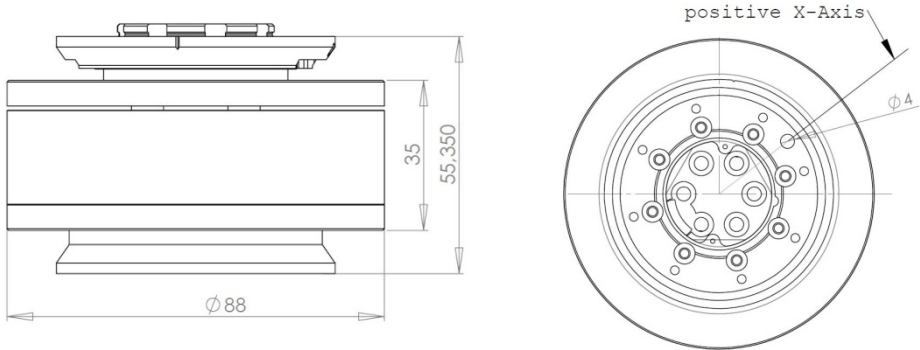
2.2 Specifications

Functional	
Force measurement	In x, y and z +/- 300 N
Torque measurement	Around x, y and z +/- 15 Nm
Allowed overload	2 x
Principle	Strain gage measurements
Resolution	16 bit
Communication Frequency	up to 100 Hz
Mechanical	
Dimensions	55.35 mm height, 88 mm diameter
Weight	0.7 kg
Flanges	Schunk FWS115 (FWK and FWA)
Electrical	
Operating voltage	24 V, < 0,3 A
Range	21,6 to 26,4V
Interface	CAN bus with 500 kBit/s
Connector	Schunk FWS115
Environment	
Ambient Temperature	Living- / working areas: 10 - 35° C

3. Interfaces

3.1 Dimensions and Orientation

The sensor coordinate system is a right handed rectangular system. The X axis is oriented through the 4 mm hole in the flange adapter. The Z axis goes up.



Pic. 1: Sensor Drawings (form above and side, shown without the FWS fixture on the upper adapter)

The sensor is equipped with Schunk FWS115 adapter on both sides:

- ➔ FWK on the lower side (to be attached to the robot)
- ➔ FWA on the upper side (for attaching further equipment)

3.2 Electrical Connectors

The sensor is connected via the Schunk FWS115 adapter. The used pins are:

- Power supply via GND and 24V
- CAN via CAN_L_SDH and CAN_H_SDH. A termination resistor is integrated

The 8 signals 24V, GND, CAN_L_SDH, CAN_H_SDH, RxD, TxD, CAN_L_ERB, CAN_H_ERB and the two power lines +Um and -Um are connected to the two adapter pcbs through the sensor body. The power lines Um provide a 0.5 mm² wire diameter.

The sensor needs a power supply of 24V DC at < 300 mA.

4. CAN Protocol

The communication with the sensor is based on the CAN protocol with standard frame. Factory settings for CAN ID and baud rate are:

- Baud rate: 500 kBit/s
- CAN ID: 0x50 for data requests, 0x51 and 0x52 for answers.

It is important to send the messages with the right length; otherwise the sensor will ignore them.

Measurements:

A measurement has to be requested by the robot controller with a CAN message, then the sensor answers with two CAN messages containing the measurements. Sensor values can be requested with up to 100 Hz.

	CAN ID	Data
Request	0x50	0x07 timeStamp
Answer part 1	0x51	Byte 1: error / timeStamp Byte 2: channel 1 HighByte Byte 3: channel 1 LowByte Byte 4: channel 2 HighByte Byte 5: channel 2 LowByte Byte 6: channel 3 HighByte Byte 7: channel 3 LowByte Byte 8: temperature HighByte
Answer part 2	0x52	Byte 1: error / timeStamp Byte 2: channel 4 HighByte Byte 3: channel 4 LowByte Byte 4: channel 5 HighByte Byte 5: channel 5 LowByte Byte 6: channel 6 HighByte Byte 7: channel 6 LowByte Byte 8: temperature LowByte

The timestamp byte in the request message is repeated in the answer and allows to detect communication gaps. It should lie in between 0 and 63. In the sample program

it is just a counter being incremented with modulo 64. Byte 1 of the answers contains the following information:

Bit	Contained Information
1 – 6	timeStamp repeated
7	0: no error 1: overload
8	0: no error 1: system error



A value of byte 1 higher than 63 or 0x3F means that the sensor is in an error condition! The application has to react on this!
The measured strain gage values are not changed.

The transmitted measurements are averaged raw measurements of the strain gages in the sensor. The forces and torques can be gathered with a matrix multiplication with the supplied calibration matrix provided with the sensor. The sensor values have a range of 2 bytes, the idle position is calibrated to approx. 32000.

The temperature value can be transformed into a °C value using a sensor specific formula given in the QS sheet provided with the sensor.

Information Requests:

	CAN ID	Data
Request Working Hours	0x50	0x03 0x54
Answer	0x51	Byte 1: 0x54 Byte 2: working hours HighByte Byte 3: working hours LowByte Byte 4: working minutes Byte 5: working seconds Byte 6: firmware version primary Byte 7: firmware version secondary Byte 8: 0x00

Request Serial Number	0x50	0x03 0x55
Answer	0x51	Byte 1: 0x55 Byte 2: product type HighByte Byte 3: product type LowByte Byte 4: serial number HighByte Byte 5: serial number LowByte Byte 6: firmware version primary Byte 7: firmware version secondary Byte 8: 0x00

Configuration:

Changes of the CAN ID and baud rate take effect after the next restart of the sensor. There is no answer from the sensor.

	CAN ID	Data
Set CAN ID	0x50	0x02 0x01 newID
Set CAN baud rate	0x50	0x02 0x02 brValue brValue = 0x20: 125 kbit/s brValue = 0x30: 250 kbit/s brValue = 0x40: 500 kbit/s brValue = 0x50: 1 Mbit/s

5. Software

The sensor is provided with the test software SensorViz. In the following sections we provide information on the installation of the necessary driver and the software.

5.1 Driver Installation for the Peak System PCAN USB Adapter

The SensorViz software can be used with the PCAN USB adapter from Peak Systeme, or with a CAN2USB adapter from Commonplace Robotics. The adapter can be purchased separately.

The .ini-File determines which adapter to use, the PCAN-USB is preset.

Please follow the Peak installation instructions to install the necessary driver.

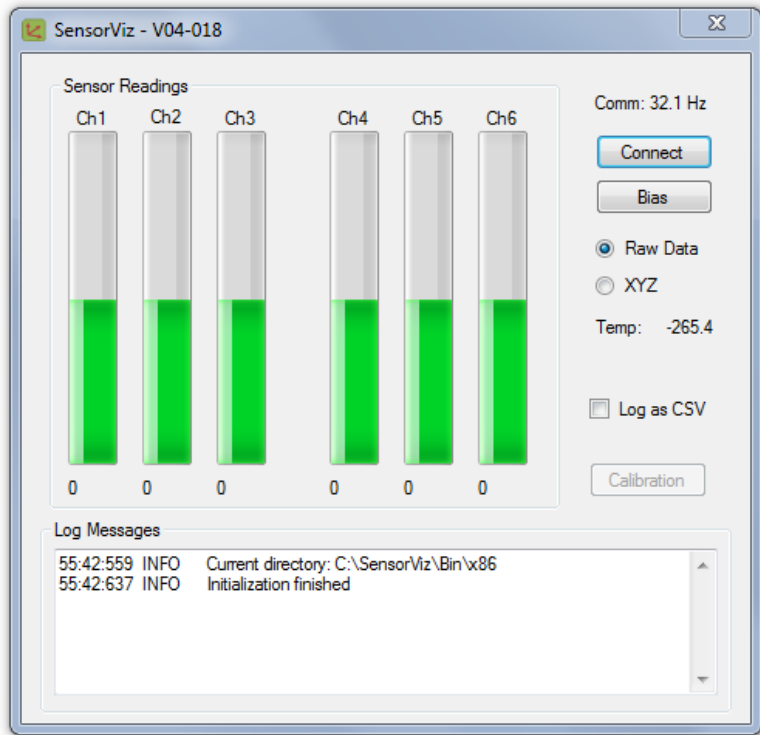
5.2 Test Software SensorViz

SensorViz is software that allows to:

- Connect with the sensor
- Show the current raw sensor readings
- Set the bias of the sensor
- Load the calibration matrix and show XYZABC measurements
- Log the measurements into a .csv file

The “Connect” button tries to open a connection to the CAN adapter. When the adapter is not found a message is written to the log window.

When connected successfully, new measurements are requested with approx 30 Hz. These measurements are shown as raw channel measurements, or as forces and torques when XYZ has been checked.



Pic. 2: Sensor test software user interface

The check box “Log CSV” allows to log the sensor values into a file. The name of the file is

forceValues_hhmmss.csv

hhmmss stands for hours, minutes and seconds. The values are separated by a space.

The software is initialized with the configuration file “Configuration.xml”. The following entries are relevant:

Adapter to use:

```
<BusConfiguration Type="PCANUSB" />
```

Decimal CAN ID and cycle time:

```
<Communication CANID="80" ComCycleTime="30" />
```

Furthermore the configuration file contains the calibration matrix:

```
<FTSTransferMatrix SerialNumber="24210294">  
  <RowTX Col1="-0.05644" Col2="0.30843" Col3="-0.00130"  
    Col4="-0.01268" Col5="-0.00048" Col6="-0.28078" />  
  ...  
</FTSTransferMatrix />
```

6. C++ Sample Code

The installation CD contains the folder \CPPSample. Here you can find a small sample program that

- connects with the Commonplace Robotics USB2CAN bridge,
- reads the sensor measurements,
- does the matrix multiplication and
- shows the applied forces.

The software has been developed on Ubuntu Linux and uses boost and ncurses libraries. The source code contains extended comments.

The calibration matrix in the sample code is a generic one and has to be exchange with the sensor specific one.

7. Troubleshooting

7.1 Error Codes

The board provides two means of error indications, via the inner LED and via the CAN bus. Furthermore the USB2CAN adapter provides information.

7.1.1 *Sensor LED*

Off:	No power
Slow blinking (1 Hz):	Sensor is running idle
On:	Sensor is communicating

7.1.2 *CAN-bus*

See section "CAN Protocol" for error codes in the measurement answers:

Byte 1, bit 6:	0: no error	1: overload
Byte 1, bit 7:	0: no error	1: system error

7.2 Service Contact

If your problem is afterwards still not solved, we are happy to provide assistance:

- Mail: info@cpr-robots.com
Please always add serial number and a description of the problem.

