XELA Robotics Tactile Sensor - Instruction Manual (uSPa 44)

August 2023

1 General Limitations for using the Sensors

- Applying too high forces or pressures will destroy the sensor module and will void the warranty. Never apply more than 20 N z-axis force (force applied perpendicular to the sensor's surface) to one sensor cell (the XR1922 has 4 sensor cells, for example). Regarding x-axis and y-axis, the sum of the shear forces has to stay below 15 kPa.

  Furthermore, apply forces only to the sensor surface, not to the sides of the sensor module.

- As our sensor uses magnetic field changes induced by the skin deformation as its sensing principle, other magnetic fields (including the earth magnetic field), nearby magnets, or nearby ferromagnetic materials can influence the sensor measurements. Also crosstalk between two of our sensor modules is possible. Please confirm within the inspection period (1 month from receipt of the product) with your application if those influences are prohibitory for your application. To counteract those influences, please also consider that a reference sensor could be used.

- Never bend the sensor modules. When you install the sensor modules on your robot, glue them to a sturdy and flat surface with thin double-sided sticky tape. Make sure to provide flat support to the whole backside of the sensor module.

2 Requirements

To collect data from the sensors, a PC with a USB 2.0 port is required. Both a Windows or Linux PC can be used, as described in the "Software Manual". However, for the simple procedure to check if your sensors are connected correctly to your PC, as described in this manual, a Windows PC is used. The software described in this manual was tested on Windows 7 and Windows 10.

Our sensors work with various CAN-USB converters, as described on our webpage (https://xela.lat-d5.com/canusb). However, we recommend to use the CAN-USB/2 from VSCOM for Linux or from ESD for both Windows and Linux to achieve a higher measurement frequency and for best compatibility.

This manual (in particular Section 5.2 and 6.1) is based on the software for the CAN-USB/2 from ESD.

The following CAN-USB devices are supported and tested.

- ESD CAN-USB/2 (bus: *esd* in Windows or *socketcan* in Linux)
- VScom USB-CAN Plus (bus: *slcan*, Linux/ROS only)
- PEAK USB-CAN (bus: *pcan*, default channel: CAN_USBBUS1, Linux/ROS only)
- CANable and CANable Pro (bus: *socketcan*, Linux/ROS only, with candlelight firmware) (Recommended only for advanced users knowing CAN DSUB-9 pinout)

Warning

Even though other CAN devices might work, misconnection can break the sensor.

Please always use CAN devices with standardized DSUB-9 connector

The software described in the "Software Manual" is based on Python. However, other programming languages can be used both for the server (which reads out the sensor data from the CAN bus) and the client (which uses the sensor data). It is straightforward to use other programming languages for the client, as they only have to connect to the server. For the server, while we only provide the server in Python, other environments can be used, as long as they are compatible with the used CAN-USB converter. For example:

- ESD provides API for .NET, C#, Python, VC, Visual Basic, BC, LabVIEW, Linux, etc. Please refer to ESD website for more information.
- VSCOM provides API for C/C++, C#, VB.NET, Delphi and LabVIEW. Please refer to VSCOM website for more information.

3 Hardware Introduction

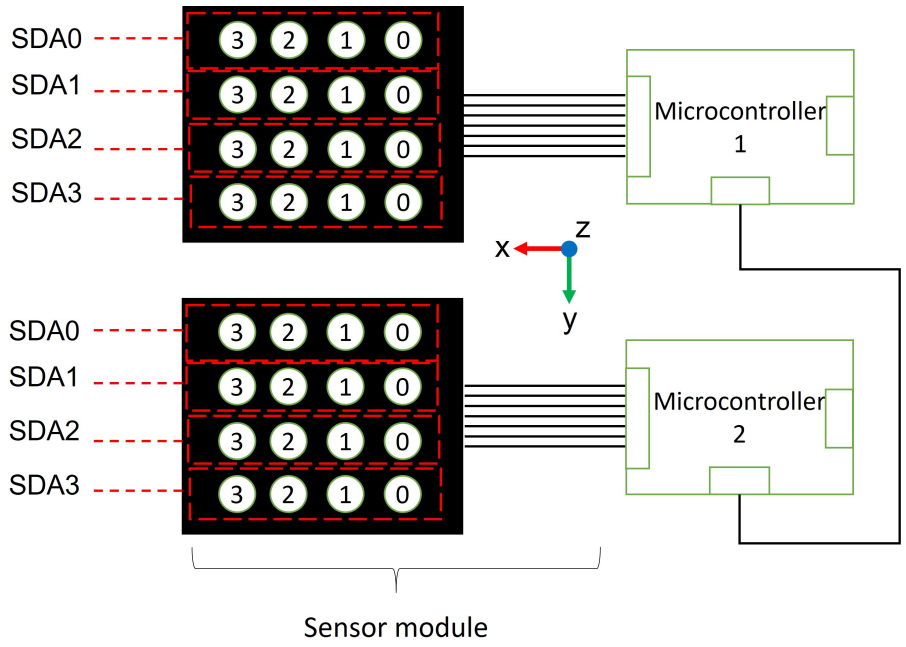

Figure 1: The connection between Sensor Module, the microncontroller, and their respective SDA numbers and taxel numbers. The measurement axis is also shown here.

3.1 Sensor Module

Each Sensor Module has 16 sensing points (taxels) in total. Each taxel measures 3-axis skin deformation. The sampling rate is 125 Hz. Each measurement has 16-bit (8 Most Significant Bit/ MSB and 8 Least Significant Bit/ LSB) resolution per axis. Please see Figure.1 for the taxels' number, their position, and their respective SDA.

3.2 Microcontroller

The pre-programmed microcontroller is used to start the communication, configuring, and collecting the data of the Sensor Module. The microcontroller can be connected to the Sensor Module through its 8-pin port.

On the microcontroller, there are two 4-pin ports (VDD, D+, D-, GND). One of those ports is for the communication between the microcontroller and the CAN/USB converter. The other one is for a daisy-chain communication between microcontrollers through a CAN protocol. These ports are interchangeable. Several microcontrollers with Sensor Modules can be daisy-chained.

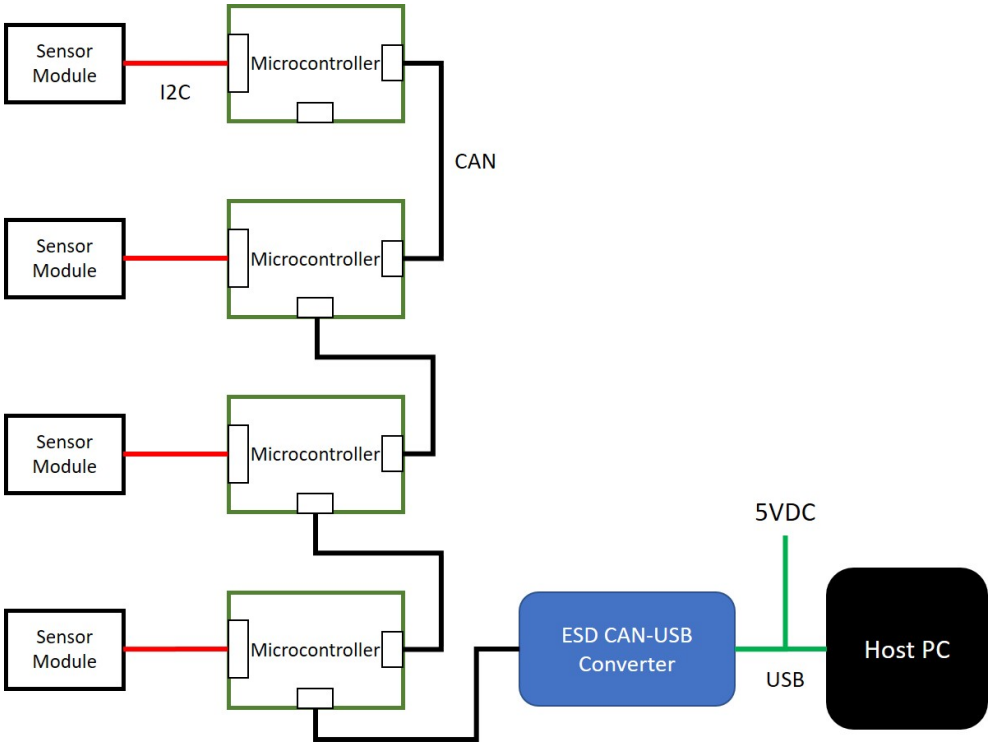

Figure 2: Connection diagram.

3.3 ESD CAN/USB Interface

This device interfaces a PC with the microcontroller. It is connected to the PC with a serial bus (USB). This device was developed by ESD and can be purchased separately from https://esd.eu/en/products/can-usb2. The driver is also available from the given link.

3.4 CAN to DSUB9 cable

This cable connects the microcontroller (4-pin connector) to the ESD CAN/USB converter (DSUB9 connector). The 4 pin wires are for transmitting data to the ESD CAN/USB interface.

4 Setup & Installation

4.1 Connecting the hardware

1. Plug the 8-pin wires of one Sensor Module's into a microcontroller.
2. Connect the 4-pin connector of the CAN/DSUB9 cable to 1 of the 2 4-pin port of the microcontroller.
3. Connect the DSUB9 connector of the CAN/DSUB9 cable to the CAN/USB Interface.
4. Plug the USB cable of the CAN/USB Interface into any of the USB ports of your PC.
5. Plug the USB power cable of the CAN/DSUB9 cable into a PC or USB wall adapter (5V). The power indicator of the microcontroller should be on.

4.2 Driver and Libraries Installation

4.2.1 ESD CAN/USB driver

After plugging in the USB cable, open the device manager from the control panel. In the USB section, make sure that the device is detected as an unknown device. Right click the unknown device and specify the driver location to the CAN USB Driver folder (...\CAN USB Driver). The driver can be downloaded

from the ESD website or can be found inside the installation CD. If it is successful, the unknown device should turn into "CAN Interface - CAN USB/2" as in Fig. 3.

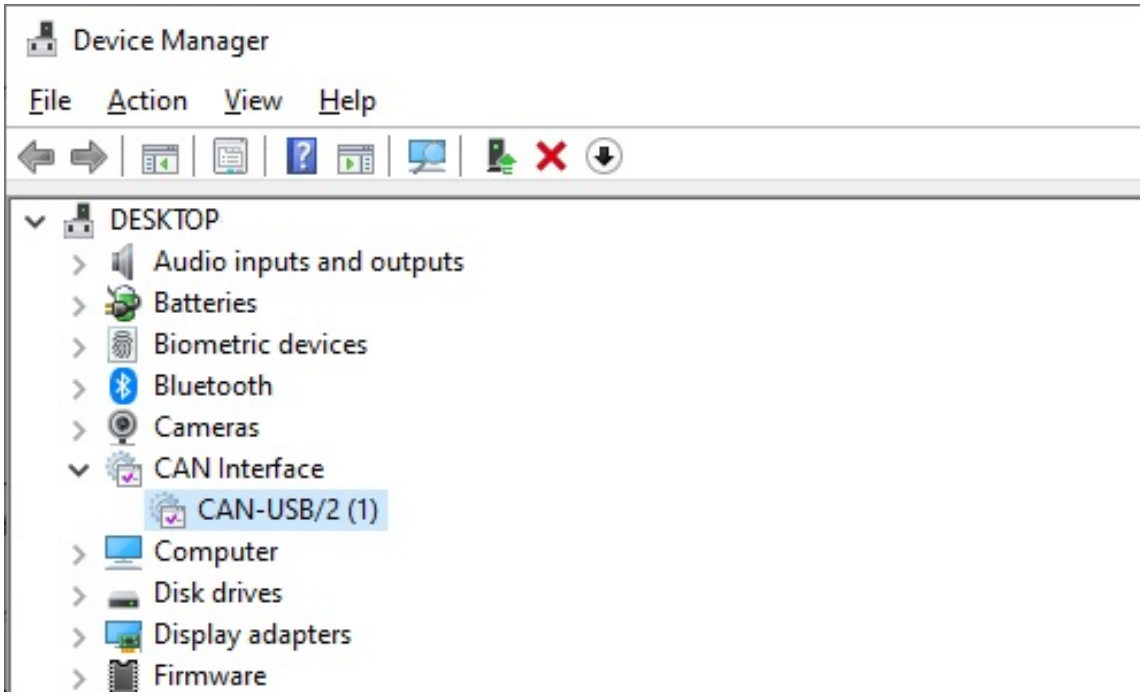

Figure 3: Properly installed CAN-USB device.

4.2.2 CAN SDK

Run CAN_SDK.exe from ...\CAN USB driver\CAN_SDK and follow the instructions. This will install the ESD CAN/USB libraries and sample programs required for the next step.

5 Explanation of CAN ID and CAN message

5.1 CANreal

Here we use CANreal application provided by ESD to make the explanation easy to understand. Run the CANreal application. The application is installed as part of SDK and can be found by default at C:\Program Files\ESD\CAN\SDK\bin32\CANreal. Configure it as in Fig. 4.

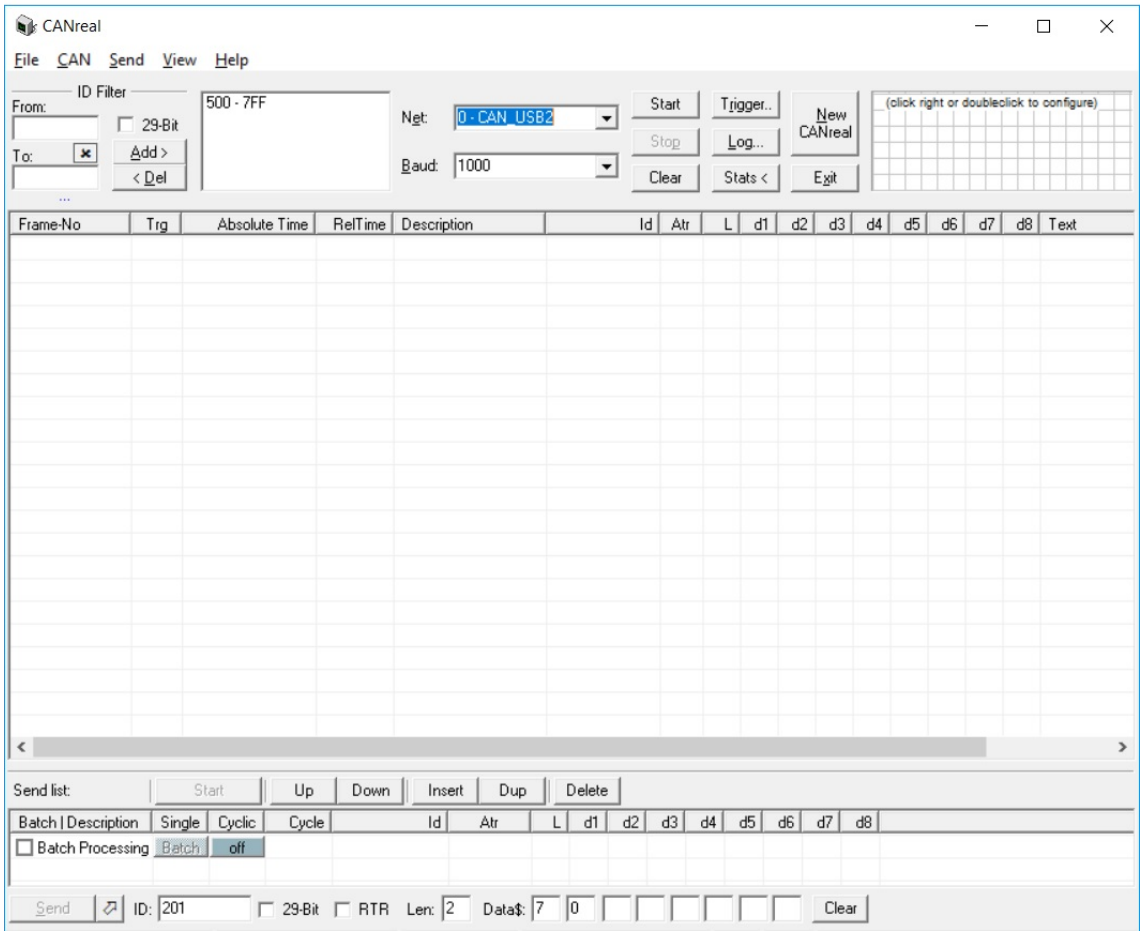

Figure 4: CANreal configuration.

Select a detected CAN/USB device by choosing it from the "Net" drop-down menu. If there is nothing that can be selected, the device may not be plugged or the driver is not installed properly. Configure the "Baud" to 1000 then click "Start". A Successful connection will lead to an incoming CAN Message as in Fig. 5. At this point, your PC is ready to run our sample code.

5.2 Incoming CAN ID structure

Each incoming CAN message comes with its ID. The ID represents the number of microcontroller and the taxel of the Sensor Module connecting to that microcontroller. The meaning of the ID is as shown in Table 1.

Bit	10	9	8	7	6	5	4	3	2	1	0
Group	SDA number			Taxel number				Microcontroller ID			

Table 1: Incoming CAN ID Structure

- "Microcontroller ID" is pre-defined. The number can be found on each microcontroller.
- "SDA number" is defined as in Figure 1.
- "Taxel Number" is defined as in Figure 1.

Therefore, the incoming CAN IDs of the taxels on the Sensor Module which is connected to the microcontroller **ID1**, are as follow.

- SDA 0 Taxel 0 - 3 : 0x001 - 0x031

- SDA 1 Taxel 0 - 3 : 0x101 - 0x131
- SDA 2 Taxel 0 - 3 : 0x201 - 0x231
- SDA 3 Taxel 0 - 3 : 0x301 - 0x331

For the Sensor Module that is connected to the microcontroller **ID2**, the incoming CAN IDs of the taxels are as follow.

- SDA 0 Taxel 0 - 3 : 0x002 - 0x032
- SDA 1 Taxel 0 - 3 : 0x102 - 0x132
- SDA 2 Taxel 0 - 3 : 0x202 - 0x232
- SDA 3 Taxel 0 - 3 : 0x302 - 0x332

5.3 Incoming CAN Message structure

Each incoming CAN message contain 8-byte data. The data compose of 3-axis components of contact measurement. The structure of the data is as follows.

- 1st byte : Not used
- 2nd byte : X-axis MSB
- 3rd byte : X-axis LSB
- 4th byte : Y-axis MSB
- 5th byte : Y-axis LSB
- 6th byte : Z-axis MSB
- 7th byte : Z-axis LSB
- 8th byte : Not used

By combining the MSB and LSB part of each axis, the 16-bit measurement can be acquired.

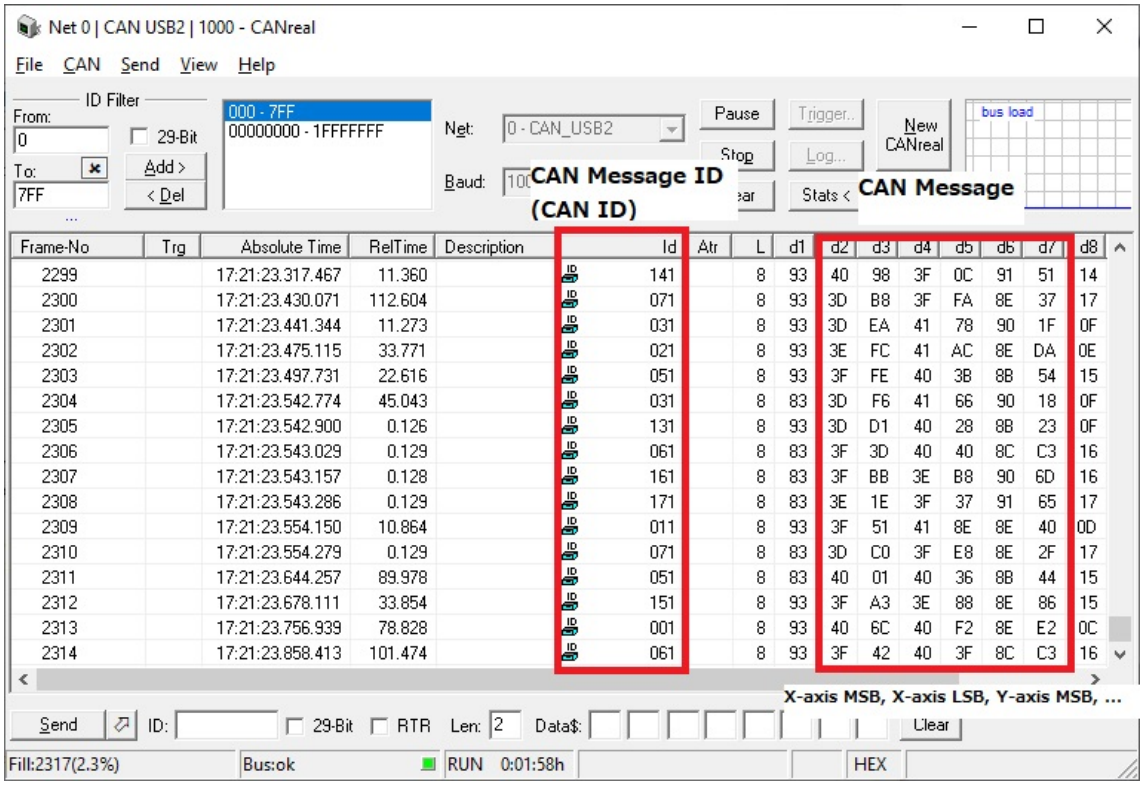

Figure 5: Incoming CAN message.