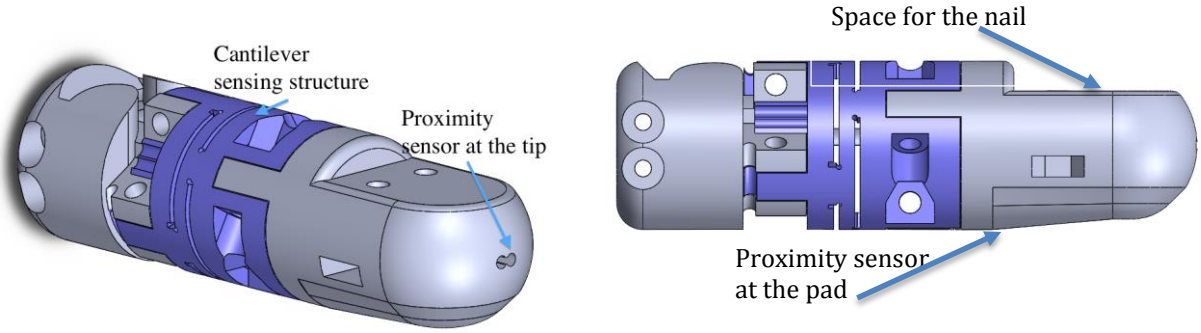
Contents

Fingertip sensors

General information

Each finger is equipped with one three-dimensional tactile and two proximity sensors, as shown on figure below. The cantilever sensing structure enables the possibility to perceive normal force (Fz) and two lateral moments (Mx and My). The proximity sensors are located at the pad and at the tip (distal) of the fingertip.



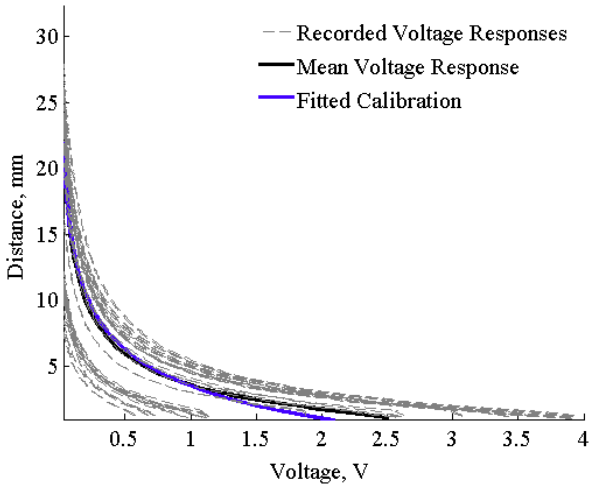
Sensors work using the principle of light intensity modulation. Light to voltage conversion is performed using KEYENCE convertors. Each convertor is specifically calibrated for specific sensing element. Do not adjust the settings of the device, unless it is performed to remove the saturation, or to bias to zero.

KEYENCE convertors are numbered from 1 to 15 to correspond to specific sensing element (five for each finger).

Do not apply excessive moments or pressure to the cantilever structure, as it might result to breakage of a material.

Proximity sensors

In the current version, readings of the proximity sensor depend on the environmental lightening and reflective properties of an object. Current calibration uses the mean response across range of various objects. That calibration measures the distance up to 18 mm. Currently the distance estimation can be used as a relative estimate. (In future versions of the software we will include an improved distance estimation that uses several calibration curves).



Hardware connection and data acquisition

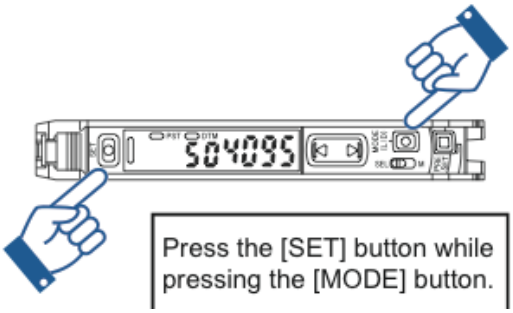
To read voltage signals form KEYENCE convertors, Arduino board is used, employing 15 analog inputs. Connections required for the system of fingertip sensors are as follows:

1. USB port for Arduino board (A to B USB cable);
2. 12 V power supply (X A) - brown cable from KEYENCE convertors;
3. Ground connection – blue cable from KEYENCE convertors;
4. 5 V output – orange cable from KEYENCE convertors to Arduino board;

Instructions after installation of the sensors

After installation of the fingertips sensors to the robotic system the following steps should be performed:

- Check if all KEYENCE devices reflect zeros (**red cipher**). Make sure no object is in the vicinity of the proximity sensors and tactile elements are not pressed.
- In case some of the KEYENCE display other red numbers then zero biasing should be performed.
- Press SET and MODE buttons at the same time. This is done to remove possible **saturation**.



- Then **set the value to zero** by pressing left arrow and PRESET button. Make sure no pressure is applied to sensor, as it influences the calibration.

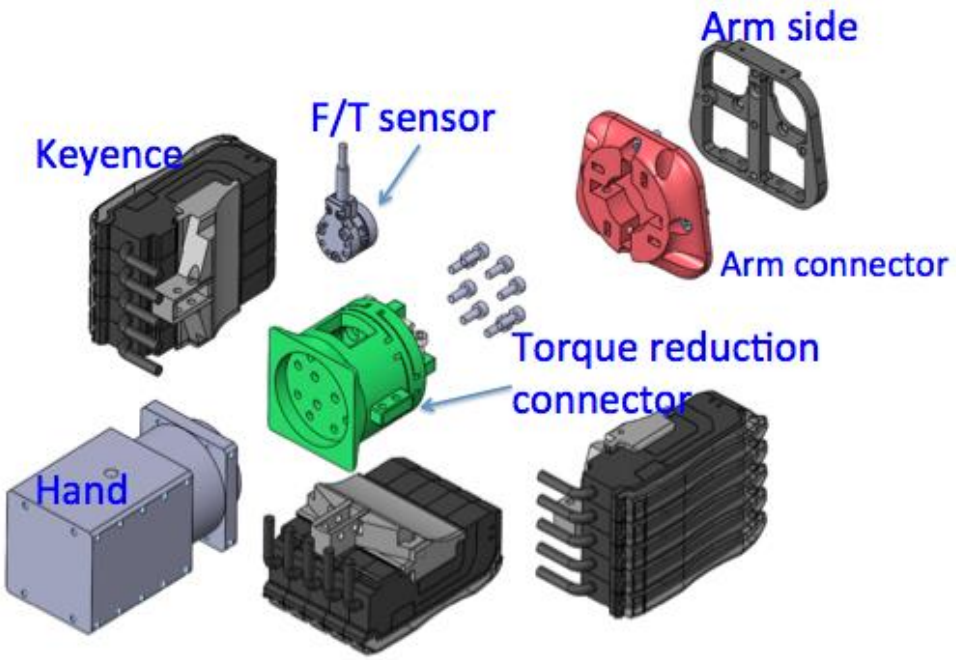
Press briefly at the same time.

The above-mentioned steps can be performed also after an extensive use of sensors to remove saturation.

Force and torque sensors (FT17)

Important information

This sensor is a sensitive instrument and should always be handled with care. Never exceed pure forces of 100N and Torques of 0.5Nm on any axis. Be extremely careful with the loads applied at the end of the mounted tool. Sensor is mounted inside a protective connector, as shown on the figure below (In the next configuration that will be provided soon, KEYENCE will be moved to Robotino). Do not use the sensor with the damaged connector. When mounting a connector at the sensor output flange make sure that no torque is transferred by the screw tightening action between the flange and the sensor base. If the screw tightening loads exceed the sensor specifications permanent damage may occur.



A method to achieve safe mounting of the tool to the output flange is to first mount the tool on the sensor flange while the sensor base is still unmounted (Not grounded). First insert and lightly tighten the mounting screws to hold the tool in place. Next hold the sensor-tool assembly by the output flange or by the tool and tighten the screws while the sensor base is ungrounded and completely free. Never hold the tool by the base while tightening the screws of the output flange.Then place the sensor base at the grounding assembly or robot and insert and tighten the base mounting screws.

Never tamper with the sensor body screws as this will void the calibration and can damage the sensor permanently.

For the details of setting up the sensor, including the default IP mode, please refer to *FT_SMDB_B0.pdf*.

Hardware connection and data acquisition

Sensor requires Ethernet connection and 5 V power supply.

Software

Sensor drivers are located in GitHub server in the folder *squirrel_project/squirrel_driver/squirrel_sensing_node/*.

The sensor drivers are composed of two parts, the ROS node (node.cpp/.h) and the hardware drivers (sensing_drivers.cpp/.h). To start the ROS node it is required to give the following command:

```
rosrun squirrel_sensing_node sensing
```

Assuming that an instance of roscore is running in background.

Before running the ROS node, the user shall make sure that right port used for the communication with the Arduino is referenced in the code. The easiest way to check which serial port is used by the arduino is to run the Arduino IDE and check which serial port is assigned by the operative system. Most of the time it would be `/dev/ttyACM0` but occasionally this might not be the case. In this case, it would be required to change the line 23 of *main.cpp*:

```
SensingNode sensing(name,"/dev/ttyACM0");
```

To reflect the right serial port. Indicate the wrong serial port of the Arduino will fail the initialization of the ROS node.

Additionally, it is advised to have the F/T sensor is connected to the setup before starting the ROS node. Not doing so will crash the ROS node at start-up, however, it is possible to exclude the F/T sensor after recompiling the code as described in the F/T sensor subsection.

Fingertip sensors

The fingertip sensors use an Arduino Mega microcontroller to perform Analog Digital Conversion (ADC) of the readings. The code (Arduino sketchbook) used to perform the ADC operation is stored in folder *squirrel_project/squirrel_driver/daq_arduino*. To flash the arduino with the code it is reccomended to use the functionalities of the Arduino IDE.
The force and proximity readings are published on topic */fingertips.* The topic publishes 15 values. The first 12 readings correspond to the tactile sensors while the last 3 readings correspond to the proximity sensors.

The sensor driver uses file *tactile_calibration.ini* to calibrate the tactile sensors. If, for any reason, the file is not found at start up, default values of 1 are assumed and a warning message will be printed. This will most likely cause wrong readings from the sensors.

F/T sensor

To execute the ROS node using the F/T sensor the default IP address should be setup.

The topic that broadcasts data from the sensors is */wrist*. The topic broadcasts 7 values: force on the X, Y and Z axes, torque on the X, Y, and Z axes and a timestamp. The force and torque values are all unbiased.

If the F/T sensor is not connected when the ROS node is started, the node will crash. To prevent this to happen, if the F/T sensor is not connected, it is possible to recompile the code to exclude the use of the F/T sensor. In order to do so, line 10 of node.h:

```
#define _FT17_AVAIL
```

Should be commented and the code should be recompiled afterwards.