BOSCH CISS Integration

Operation Manual

Pintor Ortiz, Walter Frank
Sr Development Engineer
Advanced Robotic Applications,
Advanced Remanufacturing and Technology Center (ARTC)
walterfrank@artc.a-star.edu.sg

Contents

1 Overview

1.1 About

This package offers end-users a simplified way to connect and stream data from the BOSCH CISS sensor. The opportunity to develop a solution that allows information extraction, was raised from the continuous work in several in-house applications, dedicated to enhance the connectivity and digitization processes in the robotics team and the company (ARTC). This development aims at enabling more technology core developments and applications, in specific areas such as navigation and perception. BOSCH provides a wide variety of solutions for customers but in specific the CISS sensor, is dedicated for robust activities. The selection of this sensor was purely due to the industrial capacity and thus, the recognition that this sensor can have a deeper impact in more sectors for research.

Figure 1: BOSCH CISS sensor - 8 Sensors in 1

The are four parts in this software package for easier end-user adaptation and adoption:

1. The first is by BOSCH (OEM), as they have provided an example in the main website. This example was taken as the main template for the overall development and it is written in Python 2.7.

2. The second portion is the conversion to Python 3.5. This modification aims at ensuring a wider public adoption in case future developments are carried out in higher versions of Python.

3. The third part is dedicated to the integration of an industrial protocol with the data streaming provided in the Python 3 code. This simplifies the connectivity with PLCs.

4. And the final portion is dedicated to the integration with the ROS environment. Aimed at enhancing the localization for the navigation capabilities.

1.2 Revision History

S.No.	Version Number	Latest
1	1.0.0	Yes

Table 1: Revision History of BOSCH CISS sensor

1.3 General Note

This is an open application and end-users are encouraged to utilize it, debug it and provide feedback to enhance the usage and utilization.

1.4 General Information

As indicated in the official website, the CISS is *"a multi-sensor device detecting acceleration and vibration as well as environmental conditions. The robust housing and the small outline makes it perfectly suitable for industrial retrofit applications such as condition monitoring and predictive maintenance. Configuring the device enables the customer to address a broad variety of use cases by interpreting the sensor data by smart algorithms"*

Sensor	Measurement Range	Accuracy
Accelerometer	± 2, 4, 8, 16 g (14 bit resolution)	± 50 mg
Gyroscope	± 2000 °/s	± 1°/s
Magnetometer	±1300 T (X,Y-Axis); ±2500 T (Z-Axis)	0.06 x M ± 25 T
Temperature	-20 ºC – 80 °C	max. ±2 °C + 3% T °C
Humidity	20 – 90% (non-condensing)	max. ±7% at +20 °C max. ±10% at -20°C
Pressure	300 - 1100 hPa	± 1.5 hPa
Light	0 - 2112800 Lux	± 15 %

Table 2: Description of Data Extraction

The CISS sensor is comprised of 8 individual sensors. Specifications of each of them can be seen in the description of data extraction, in table 2

The sensor has two methods to connect. It can be either using a USB cable (wired) or/and through Bluetooth (wireless). For this development, USB is utilized.

For Windows users, the COM port selected by default is 81. (This information can be modified in the sensor.ini file that comes by default in the package)

For Linux users, the port utilized is in /dev/ttyACM0 by default. (Users must run "sudo chmod 666 /dev/ttyACMx" in order to provide rights for applications to access this port - x changes depending on the system's numbering)

As an example, the sensor.ini file is recommended to have the following format:

```
[sensorcfg]
sensorid = CISS_1_244B2207551B8950
acc_stream = true
acc_event = false
acc_threshold = 0
acc_range = 16
gyr_stream = true
gyr_event = false
gyr_threshold = 0
mag_stream = true
mag_event = false
mag_threshold = 0
period_inert_us = 100000
env_stream = true
env_event = false
temp_threshold = 0
hum_threshold = 0
pres_threshold = 0
light_stream = true
light_event = false
light_threshold = 0
noise_event = false
noise_threshold = 0
period_env_us = 1000000
port = /dev/ttyACM0
```

Figure 2: sensor.ini file - Recommended parameters

The current frequencies utilized (100 Hz - inertial sensors || 1 Hz - environmental sensors) are sufficient to ensure correct operation for all sensors. The CISS sensor has also the capability to operate the accelerometer at higher frequencies (2 KHz). However, operating under such circumstances, would mean that the rest of the sensors have to be turned off. This approach will not be seen in this package.

1.5 Prerequisites

In general terms, the following libraries are required to start streaming data:

- Serial
- Signal
- Config
- Parser
- Csv
- Time
- Os
- Rospy
- Roslib
- tf
- pyModbusTCP

2 Operation

The complete development is located in the repository: `https://github.com/FrankP89/CISS_BOSCH_INTEGRATION`

Running Python 2 to export CISS data

1. Generate permissions to run in a particular COM port.
2. Execute ”python CissUsbConnectord.py”

Running Python 3 to export CISS data

1. Generate permissions to run in a particular COM port.
2. Execute ”python3 CissUsbForPython3.py”

Running ModBUS protocol to export CISS data

1. Generate permissions to run in a particular COM port.
2. Ensure a local server is running in parallel. (Option removed from code to give flexibility to end-users but can be activated back on)
3. Execute ”python3 CissUsbForPython3.py”

When extracting information through Modbus TCP, the port is by default 502 and the IP address is the Hostname. In the future, users will be able to define the IP address manually.

Variable	Register number(s)
Accelerometer X	100-101
Accelerometer Y	102-103
Accelerometer Z	103-105
Gyroscope X	106-107
Gyroscope Y	108-109
Gyroscope Z	110-111
Magnetometer X	112-113
Magnetometer Y	114-115
Magnetometer Z	116-117
Temperature	118-119
Humidity	120-121
Pressure	122-123
Light	124-125

Table 3: Modbus TCP registers

Running ROS Node to export CISS data

1. Generate permissions to run in a particular COM port.
2. Access branch ciss_ros_wrapper and clone
3. Build using catkin_make.
4. Source using "source /devel/setup.bash"
5. To visualize the sensor in a virtual environment, execute "roslaunch ciss_imu display.launch"
6. Retrieve the information using the correct topic. Check the "rostopic list" to determine the required sensor.

Sensor	Final Units
Accelerometer	m/s
Gyroscope	°/s
Magnetometer	T
Temperature	ºC
Humidity	% (non-condensing)
Pressure	hPa
Light	Lux %

Table 4: RViZ units required

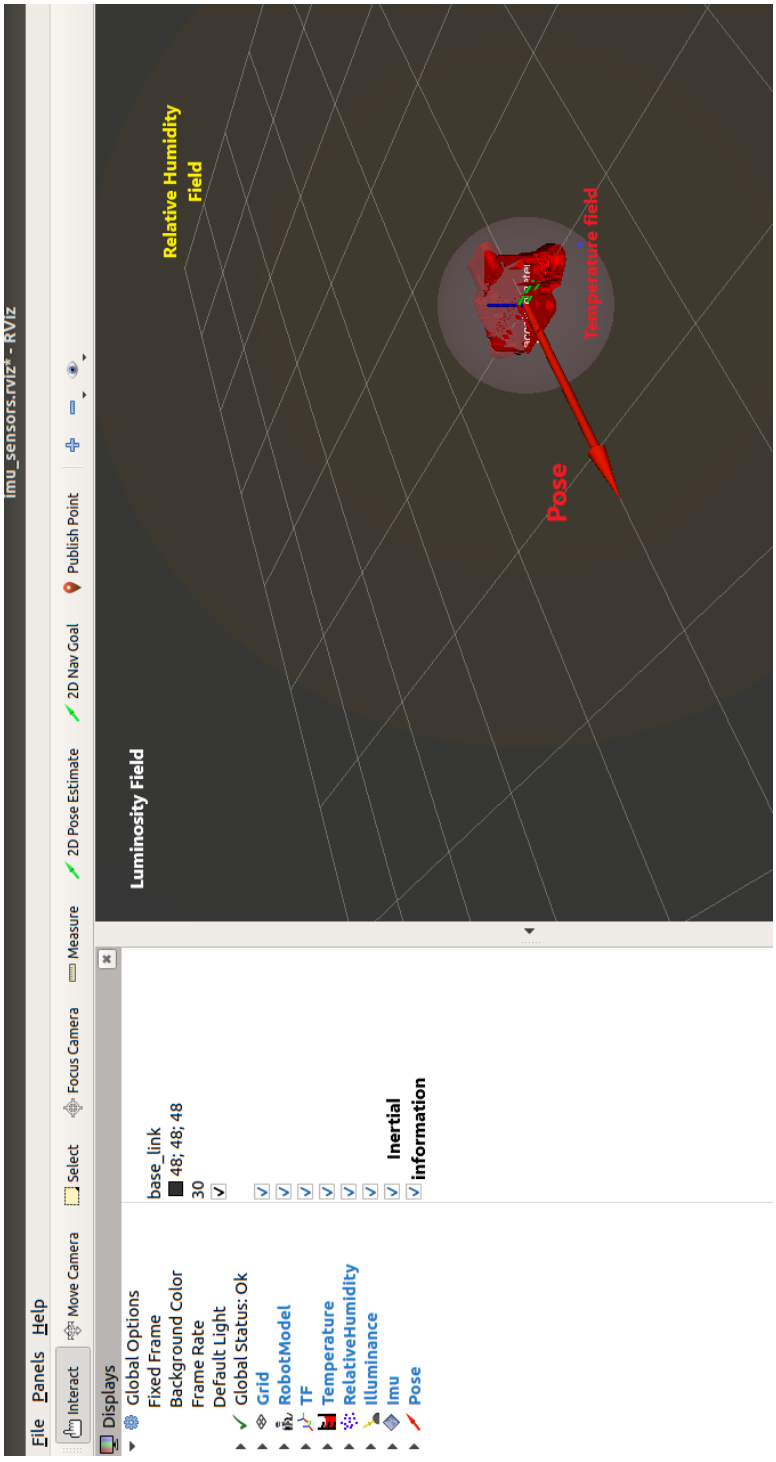

Figure 3: RViz - Visualization of sensor in virtual environments