contactile

3D Force Button Sensor
ROS
(Beta v1.0)

Installation and Operation Manual

Document #: 3DFROS_B1.0_MAN_JAN22

January, 2022

Foreword

Information contained in this document is the property of Contactile Pty Ltd. and shall not be reproduced in whole or in part without prior written approval of Contactile Pty Ltd. The information herein is subject to change without notice and should not be construed as a commitment on Contactile Pty Ltd. This manual is periodically revised to reflect and incorporate changes made to the 3D Force Button Sensor Development Kit.

Contactile Pty Ltd assumes no responsibility for any errors or omissions in this document. Users' critical evaluation is welcome to assist in the preparation of future documentation.

Conditions of Sale

Contactile's conditions of sale apply to all products sold by Contactile to the Distributor under this Agreement. The conditions of sale that apply are provided on the USB flash drive shipped with the product in the folder ‘LEGAL’ in the root directory.

End User Licence Agreement

Contactile's end user license agreement applies to all software and algorithms included with the products sold by Contactile. The end user license agreement that applies is provided on the USB flash drive shipped with the product in the folder ‘LEGAL’ in the root directory.

Compliance

The devices are sold as is.

The devices are specifically designed solely for the purposes of research and development only made available on a business-to-business basis.

The devices are not for resale.

Table of Contents

1 Introduction

The 3D Force Button Sensor Development Kit (Beta v1.0) is a system of (up to) five 3D Force Button Sensors per adapter, (up to) two Adaptors, and a Controller. Each 3D Force Button Sensor can measure 3D force. The Controller supplies power for (up to) two Adaptors and coordinates the simultaneous data acquisition from up to ten 3D Force Button Sensors. The Development Kit is shipped with visualisation software and (optional) C++ libraries for Windows and Linux environments and a ROS node for developing software control algorithms using the sensor signals.

The main components of the 3D Force Button Sensor Development Kit (Beta v1.0) are shown in Figure 1.1, connected to a laptop running the visualisation software.

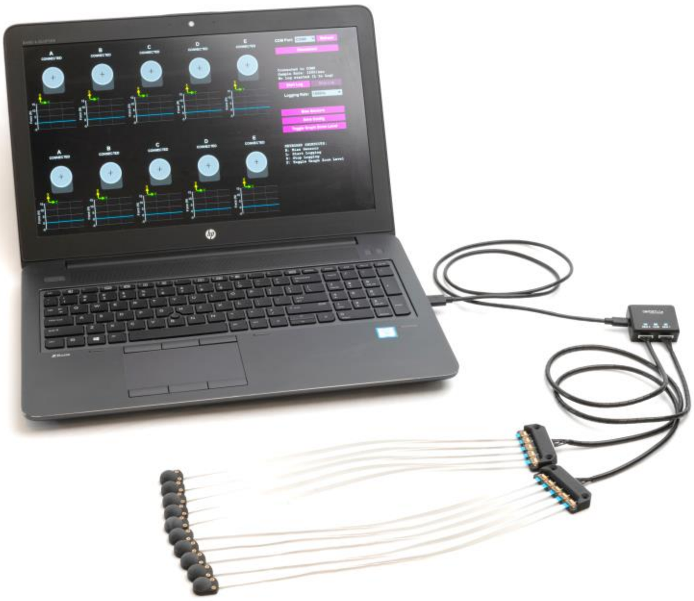

Figure 1.1 – The 3D Force Button Sensor Development Kit (Beta v1.0). Laptop not included.

This document is an installation and operation manual for the ROS Node which was provided on the Contactile USB flash drive that was shipped with the 3D Force Button Development Kit.

2 Safety

2.1 General

The customer should verify that the maximum loads and moments expected during operation fall within the sensing range of the sensor as outside this range, sensor reading accuracy is not guaranteed (refer to Document #3DFDK_B1.0_MAN_JAN22). Particular attention should be paid to dynamic loads caused by robot acceleration and deceleration if the sensors are mounted on robotic equipment. These forces can be many multiples of the value of static forces in high acceleration or deceleration situations.

2.2 Explanation of warnings

The warnings included here are specific to the product(s) covered by this manual. It is expected that the user heed all warnings from the manufacturers of other components used in the installation.

⚠ Danger indicates that a situation could result in potentially serious injury or damage to equipment.

⚠ Caution indicates that a situation could result in damage to the product and/or the other system components.

2.3 Precautions

⚠ **DANGER**: Do not attempt to disassemble the sensor. This could damage the sensor and will invalidate the calibration.

⚠ **DANGER**: Do not attempt to drill, tap, machine, or otherwise modify the sensor casing. This could damage the sensor and will void invalidated the calibration.

⚠ **DANGER**: Do not use the sensor on abrasive surfaces or surfaces with sharp points/edges. This could damage the silicone surface of the sensor.

⚠ **CAUTION**: Sensors may exhibit a small offset in readings when exposed to intense light sources.

⚠ **CAUTION**: Exceptionally strong and changing electromagnetic fields, such as those produced by magnetic resonance imaging (MRI) machines, constitute a possible source of interference with the operation of the sensor and Controller.

⚠ **CAUTION**: Temperature variations can cause drift in sensor readings. Some temperature compensation is performed. However, bias removal in software prior to operation is necessary, and it is recommended that biasing is performed each time the sensor is known to be unloaded.

3 Getting started

This section contains instructions for setting up and using 3D Force Button Sensor C++ Library for WINDOWS (Beta v1.0). It is recommended that first time users first read the preceding Safety section, then read through this section to get more familiar with the system.

3.1 Hardware installation

The C++ Library is used with the 3D Force Button Sensor Development Kit (Beta v1.0). The Controller should be connected to the 3D Force Button Sensors, then the Controller should be connected via the micro USB port on the Controller to a PC running WINDOWS before you can use the C++ Library. For more information about connecting the sensors and powering on the Controller, refer to Document #3DFDK_B1.0_MAN_JAN22.

3.2 Minimum requirements

The ROS node has been tested on a HP EliteBook with the following specifications

- CPU: Intel Core i7 -4600U
- RAM: 16 GB RAM
- Operating System: Ubuntu 20.04.2.0 LTS
- ROS installation: noetic

3.3 ROS node location

The *ROS* node is provided on the Contactile USB flash drive that was shipped with the development kit in the folder named SOFTWARE/*ROS*. The ROS node is the *buttonsensor_ros_v1* subfolder. The files in the *ROS/buttonsensor_ros_v1* folder are summarised in Table 3.1.

Table 3.1 – *Files in ROS/buttonsensor_ros_v1* folder

Sub Folder	File Name	File Description
.	CMakeLists.txt	Directives and instructions describing the project's source files and targets
.	package.xml	The package manifest
.	README.md	Readme file
include	buttonsensor_ros_node.hpp	The header file for the node
launch	buttonsensor.launch	A launch file
lib	libPTSDK.a PTSDKListener.h PTSDKSensor.h PTSDKPillar.h PTSDKConstants.h	The C++ library and associated header files
msg	ButtonSensorState.msg	Message definitions of the ButtonSensorState message
src	buttonsensor_ros_node.cpp	The cpp file for the node
srv	BiasRequest.srv	Service definition of the BiasRequest service

4 Package summary

This package implements a driver for the 3D Force Button Sensor Development Kit (Beta v1.0).

Maintainer status: maintained
Maintainer: Heba Khamis (Contactile) <heba DOT khamis AT contractile DOT com>
Author: Heba Khamis
License: LGPLv3

The buttonsensor_ros_v1 package provides a ROS interface that publishes data from ten 3D Force Button Sensors connected to a Controller that is connected over USB. The package allows the following:

1. Connecting to the controller, sampling the sensors at 100, 500 or 1000 Hz and publishing the data
2. Biasing the sensors
3. Starting and stopping slip detection on the Controller

5 Installing the package

To use this package you should have ROS up and running. To install the package copy the buttonsensor_ros_v1 folder into the *src* folder of your catkin workspace.

6 Starting the node

The buttonsensor.launch launch file is provided to configure the COM port connection, get data from the sensors and publish the data.

7 buttonsensor_ros_node

ROS-Node for connecting to the controller and publishing sensor data.

7.1 Parameters

Table 7.1 – Parameters of the buttonsensor_ros_v1 node

Name	Type	Description
hub_id	int	Identifier for the Controller
n_sensors	int	Number of sensors connected to the Controller. Should be 10.
com_port	string	Name of the COM port. Usually /dev/ttyACM0.
baud_rate	int	Baud rate for the serial connection to the Controller. Should be 9600.
parity	int	Parity of the serial connection to the Controller. Should be 0.
byte_size	int	Size of a byte. Should be 8.
log_file_rate	int	Log file rate of the Controller. Options: 100, 500 or 1000 (Hz)

7.2 Subscribed topics

/hub_0/sensor_0 (See SensorState message)

- Publish sensor data from Sensor 0 connected to the Controller (SEN0 – A)

/hub_0/sensor_1 (See SensorState message)

- Publish sensor data from Sensor 1 connected to the Controller (SEN0 – B)

…

/hub_0/sensor_4 (See SensorState message)

- Publish sensor data from Sensor 4 connected to the Controller (SEN0 – E)

/hub_0/sensor_5 (See SensorState message)

- Publish sensor data from Sensor 5 connected to the Controller (SEN1 – A)

/hub_0/sensor_6 (See SensorState message)

- Publish sensor data from Sensor 6 connected to the Controller (SEN1 – B)

…

/hub_0/sensor_9 (See SensorState message)

- Publish sensor data from Sensor 9 connected to the Controller (SEN1 – E)

7.3 Messages

7.3.1 ButtonSensorState message

Table 7.2 – Variables in the ButtonSensorState message

Parameter	Description
Header header	
int64 tus	The time in µs on the Controller of this sample
float32 gfX	The global X-force of the sensor
float32 gfY	The global Y-force of the sensor
float32 gfZ	The global Z-force of the sensor

7.4 Services

Table 7.3 – Services of the buttonsensor_ros_v1 node

Name	Description
SendBiasRequest	▪ Biasing refers to removing any offset in the sensor readings when the sensors are unloaded. ▪ It is recommended that the user performs a bias each time the sensors are known to be unloaded. ▪ Ensure that the sensor has been unloaded for at least one second before performing a bias to ensure that the bias calculation does not include hysteresis effects. A bias operation can take up to two seconds. Ensure that the sensor remains unloaded throughout this time.

7.5 Log file

7.5.1 Overview

In the buttonsensor_ros_v1/src/buttonsensor_ros_node.cpp file (line 3) a PTSDKListener object (listener_) is initialised with a boolean argument that specifies whether to log data to a .csv file or not. The argument can be changed to true (to enable logging) or false (to disable logging), remembering that the node should be remade by a call to the catkin_make utility.

7.5.2 Log file location

If logging is enabled, then a .csv log file will be generated and saved in the location /home/.ros/Logs.

NB: The .ros folder is a hidden location – make sure that you can view hidden files.

7.5.3 Log file name

The name of the log file that is generated is LOG_YYYY_MM_DD_hh_mm_ss.csv where:

- YYYY is the four digit year,
- MM is the two digit month,
- DD is the two digit day,
- hh is the two digit hour,
- mm is the two digit minute and
- ss is the two digit second,

from the system clock at the time that the log file was created.

7.5.4 Log file format

The log file is saved as comma-separated values (CSV) in ASCII text format. See 3DFDK_B1.0_MAN_JAN22.pdf for information on the order of the values and a description.