

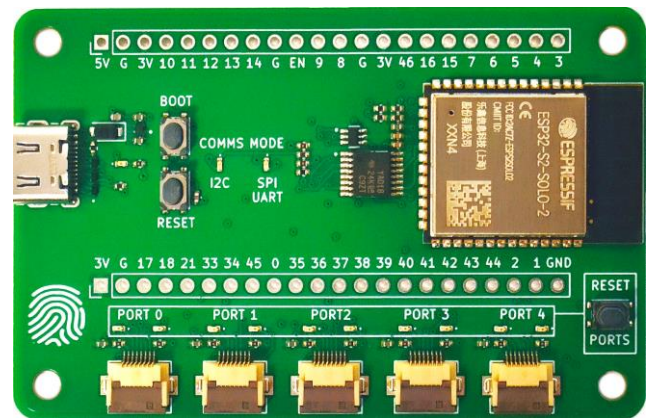
Product Overview

The DEV001 Sensor Development Board (devboard) is designed for quick connection, data visualization and logging, and firmware and software development for Contactile's I2C/SPI/UART sensors.

The development board can interface up to five Contactile I2C/SPI/UART sensors simultaneously, via 8-way 0.5 mm pitch FPC cables, and has a USB-C connection for connecting to a Windows/Linux PC or other USB host device.

The development board is pre-loaded with firmware program for streaming data to the supplied visualisation software, C++ software library, ROS and ROS2 nodes. This software enables data visualization, data logging and sensor-based control through software integration.

The development board is also supplied with a C/C++ firmware library and example firmware for custom firmware development.



Applications

- Quick start with Contactile sensors
- Custom firmware development for interfacing with Contactile sensors

Features

- I²C and SPI communication with Contactile sensors
- Dimensions 74 x 47 x 1.6 mm
- Mounting via 4x M3 holes
- Access PC software incl. Visualisation Software, C++ Library, ROS Node and ROS2 Node

Contents

1	Quick Start	3
2	Board Description	3
3	Firmware	3
3.1	Commands	4
3.1.1	help, h	4
3.1.2	bias, b [port].....	4
3.1.3	hub, mux [i2c/spi].....	4
3.1.4	info, i	4
3.1.5	mode, m [i2c/spi].....	4
3.1.6	reset, r [port]	4
3.1.7	scan	4
3.1.8	stream, s [action] [port] -d [data flags] -f [frequency] -p [print format]	4
3.2	Uploading firmware	5
4	Resources	6

1 Quick Start

Plug in one or more sensors, connect a USB-C cable, open a serial terminal, and type *help* for a list of commands. Commands are terminated with a newline character.

Start streaming 3D forces

bias

stream start -d force -f 1000 -p ascii

Stop streaming

stream stop

2 Board Description

The DEV001 Sensor Development Board (devboard) is a five-port *hub* built around an ESP32-S2-SOLO, designed to aid development with Contactile 3D force sensors. Each port is equipped with an *FH12-8S-0.5SH(55)* FPC/FFC connector, with a red LED attached to the RESET pin, and a green LED attached to the INT pin.

To accommodate multiple communication protocols (I²C/SPI/UART) without the need for rewiring, peripheral communication signals are routed through two analog switches (*TS3A5018PWR*, *SN74LVC1G3157DCKR*), which allow the communication mode to be selected by setting the value of pin GPIO45. The active communication mode is identified by one of two blue LEDs on the board, labelled under COMMS MODE.

The devboard includes a USB-C connector for both UART serial communication (up to 12Mbit/s) and device programming. The devboard can be placed into Device Firmware Upgrade (DFU) mode by holding the BOOT button (GPIO0) while toggling the RESET button (EN).

A switch labelled RESET PORTS is connected to GPIO38.

Almost all of the pins on the ESP32-S2-SOLO package are broken out to one of the two header rows on the board. For further details, the pinout and schematic are available for download [here](#).

3 Firmware

The DEV001 Sensor Development Board ships with default firmware that turns the devboard into a 5-port *communication hub* for Contactile sensors, and abstracts much of the low-level code required to retrieve force data. The firmware is available as a [Platform IO project](#). If you wish to modify the default application, but maintain *hub* functionality, include *SensorHub.h* in your project, and call *SensorHub::update()* periodically from your custom application, usually once during the main loop.

The firmware configures the integrated USB peripheral as a USB Communication Device Class (CDC), which will appear as a serial COM port when plugged into a USB port. A basic, human-readable serial command interface is included, to configure and retrieve data from connected Contactile sensors. All commands should be terminated by a newline character '\n'. Type 'help' or 'h' to print the available commands.

3.1 Commands

The *port* argument is optional for all commands. If *port* is not provided, the command will be applied to *all* ports.

3.1.1 **help, h**

Print a help message.

3.1.2 **bias, b** [port]

Remove the bias for the sensor connected to the given port.

3.1.3 **hub, mux** [i2c/spi]

Configure the analog switches on the Sensor Development Board to communicate via either the I²C or SPI peripheral interface.

3.1.4 **info, i**

Print information about connected sensors and the Sensor Development Board.

3.1.5 **mode, m** [i2c/spi]

Send a command to all connected sensors to set their communication mode.

3.1.6 **reset, r** [port]

Reset the sensor on the given port.

3.1.7 **scan**

Scan the I²C bus for connected devices. This is a diagnostic tool.

3.1.8 **stream, s** [action] [port] -d [data flags] -f [frequency] -p [print format]

Configure sensor data stream contents and frequency, and start/stop streaming data via the USB serial port. All arguments are optional.

Valid arguments

[action]:

start - start streaming

stop - stop streaming

[data flags] (determines which data are returned by a sensor):

force - force data
all - all data (force and temperature)
fx - force x
fy - force y
fz - force z
temp - temperature

Data flags should be provided as a comma-separated list, without spaces.

[frequency] (Hz):

25 – 1000

[print format]:

binary - binary data
ascii - human-readable characters (may limit data acquisition frequency)
none - stream without printing

Example 1: Stream forces from all ports at 1kHz

stream start -d force -f 1000 -p binary

Example 2: Print human-readable force data from port 0 at 100Hz:

stream start 0 -d force -f 100 -p ascii

Example 3: Stop streaming all ports

stream stop

3.2 Uploading firmware

The ESP32-S2 can be placed into Device Firmware Upgrade (DFU) mode by holding the BOOT button (GPIO0) while toggling the RESET button (EN). However, the default [firmware Platform IO project](#) includes a board configuration file (*boards\esp32-s2-solo-2.json*) which configures the host to enter DFU mode automatically, when using Platform IO tools.

Note: the ESP32-S2 is not able to properly reset itself after flashing new firmware. Therefore, the RESET button must be pressed manually after a firmware update.

For more details regarding application upload, see the [ESP32-S2 guide by Espressif](#).

4 Resources

1. Sensor Development Board Platform IO firmware project: <https://github.com/contactile/dev001.git>
2. 3DFBS datasheet <https://contactile.com/manuals/#resources>
3. C3DFBS library <https://github.com/contactile/c3dfbs>.