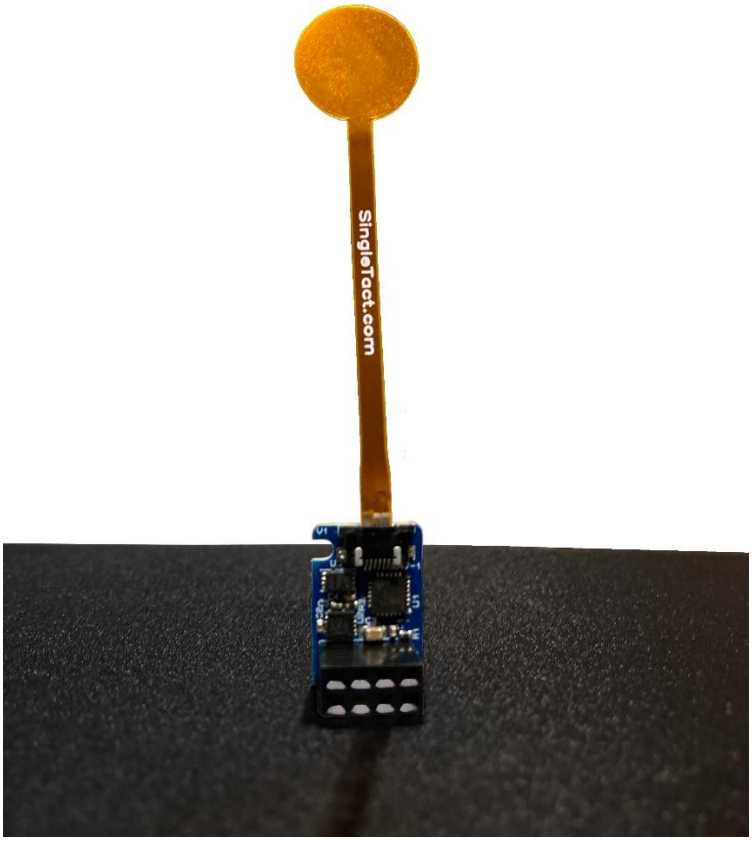
USER MANUAL

EXPERIENCE
INCREDIBLE
PERFORMANCE



V2.9

CONTENTS

1 INTRODUCTION

1.1 SingleTact Pressure Sensors

SingleTact is a single element tactile pressure sensor that accurately and reliably quantifies applied force, that can be combined with either: a simple interface board offering a 0 to 2 V analog output for immediate Data Acquisition (DAQ) integration and an I2C based interface for integration into embedded systems, or a direct to PC USB solution for the simplest way to get started with tactile pressure sensors.

Calibrated sensors are available with the USB turnkey system, and the general purpose interface board. More economical standard sensors are available for the general purpose interface board only.

For tricky designs were more sensor length is needed, a 150 mm tail extender is available compatible with all current black colored SingleTact interfaces. This is not compatible with our legacy green colored electronics.

This document provides all the information necessary to interface with the SingleTact including a sample Arduino digital interface and simple C# PC DAQ software (see EXAMPLE USE CASE)

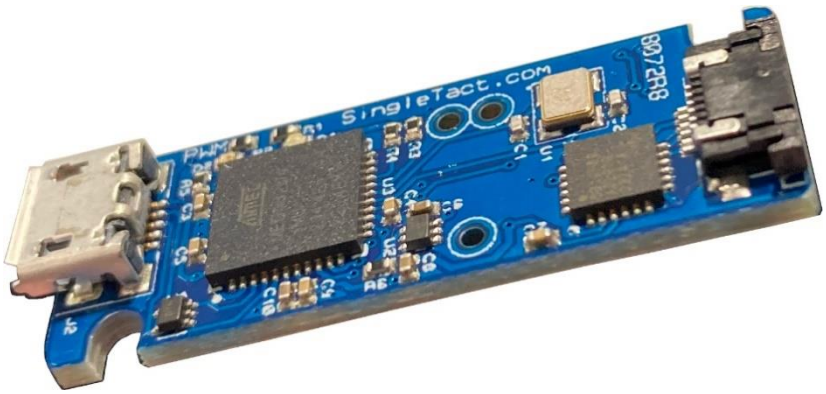
All demo and API source code is open source and can be downloaded from: www.singletact.com.

1.2 USB Turnkey Solution

The SingleTact USB Turnkey solution is our easiest way to get started with SingleTact pressure sensors. Shipped calibrated as standard and comprising all of the components required to connect your SingleTact sensors to a PC via USB, this system lets you get started with your pressure sensing application without worrying about wiring, circuitry, or DAQ systems.

The USB Turnkey system is designed to interface the SingleTact sensors directly with our open source demo software. Multiple USB systems can be used together on a single PC and displayed on our demo app, enabling prototyping of complex systems.

Figure 1 SingleTact USB Turnkey System



1.3 SingleTact System Interface Electronics

The SingleTact system interface board is the most common method for obtaining pressure information from our SingleTact pressure sensors.

Featuring both an analog and digital I²C interface in a compact form factor, the system interface board allows simple integration of SingleTact pressure sensors into embedded systems, analog controllers, and custom DAQ systems to fulfil any application.

Open source software is provided to use the system interface electronics with an Arduino Uno, and other platforms, to display data on the demo app, act as a stand-alone controller, and handle multiple parallel sensors.

Figure 2 SingleTact Sensor and Interface Board

Figure 3 Use Case Configurations

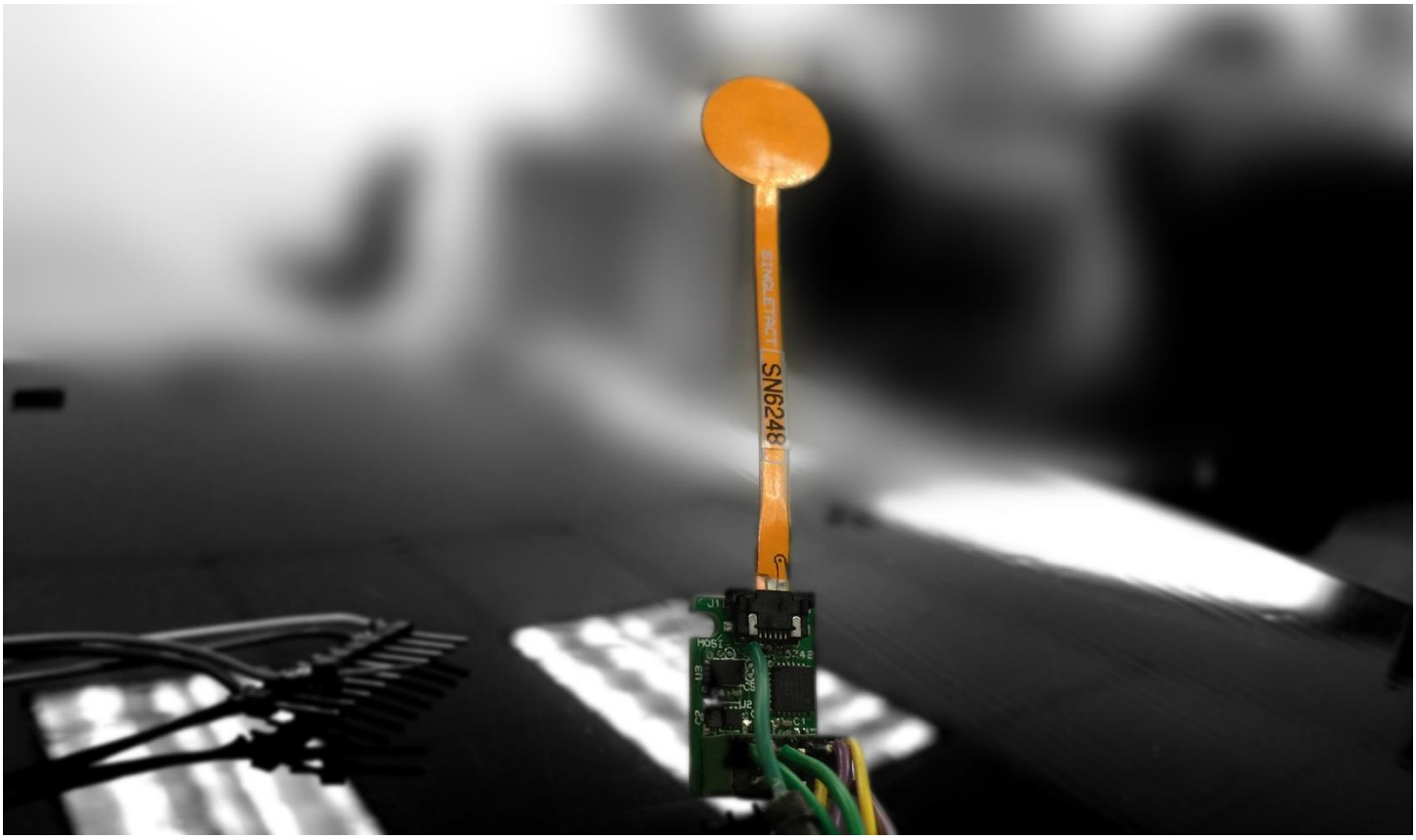
	Use Case	System Overview	
1	Force Measurement	Analog Measurement	Analog Out. Connect to multitier, oscilloscope or data acquisition card
2	Evaluation/ Force Measurement*1	I²C UNO	Arduino and PC data acquisition evaluation software
3	Customer Integration*2	I²C User Hardware	I²C Digital Output. Interface to user electronics
4	Customer Integration*3	User Hardware	User reads capacitance using their own circuitry

*1 – In addition to the data acquisition example, a .NET library is available to download for simple integration into a user's own software suite. See Example .NET API.

*2 – Supports over 100 SingleTact interface boards on a single I²C bus. The interface board firmware can be modified to fit user's specific use cases – if required please contact PPS to discuss this option.

*3 – PPS maybe able to assist with this – use the contact links at http://www.singletact.com/contact/.

2 INTERFACE DESIGN



2.1 Connectivity

2.1.1 Using System Interface Electronics

The sensor is plugged into the FFC connector on the interface board (**with the sensor connector pads facing upward**). The connections are outlined in Figure 5. Refer to the SingleTact quick start guide for connection details and procedure. Electrical parameters are outlined in Table 1.

Figure 4 Sensor Assembly

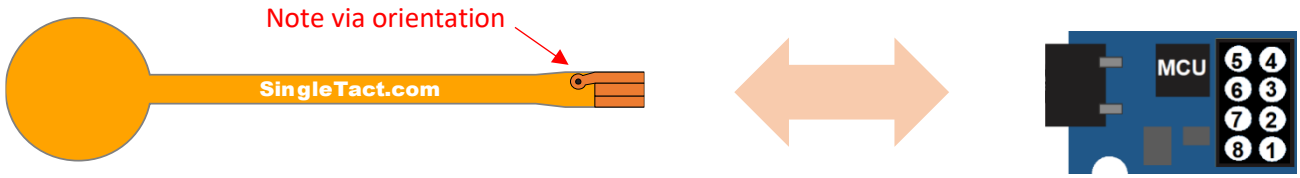


Figure 5 Interface board header connections

CONNECTION	PIN NUMBER	CONNECTION
Reserved	5 4	Reserved
I²C Interface (SDA)	6 3	I²C Interface (SCL)
Frame Sync	7 2	Analog Out
Ground	8 1	Vcc

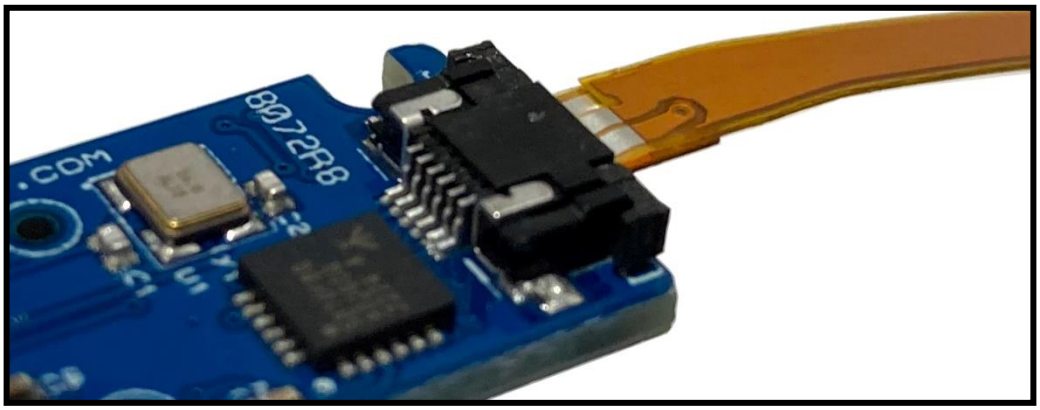
Table 1 Electrical parameters

Parameter	**Value**
Supply Voltage, Vcc	3.7 – 5 V
I²C clock frequency	100 KHz or 400 KHz
I²C bus level	3 – 5 V
I²C output range (sensor data)	10-bit (Operational FSR output 9-bit)
Analog output range	0 – 2 V (Operational FSR output 0.5 – 1.5V)
Permitted analog output load	>5 KΩ
Frame Sync level	3.3 V CMOS output
Sensor update rate (I²C or analog)	Up to 120 Hz (dependent on settings)

2.1.2 Using USB Turnkey System

The sensor is plugged into the FFC connector on the interface board (**with the sensor connector pads facing upward**) in exactly the same manner as with the system interface board, as shown in Figure 6. Refer to the USB quick start guide for connection details and procedure.

Figure 6 USB System Sensor Assembly



The SingleTact USB Turnkey solution comes calibrated as standard, with a PWM controlled LED to indicate applied pressure.

The USB system is designed to be easy to use and implement on a PC, as such, it does not feature an analog or a user serviceable I^2C output. The data is transmitted as a 10 bit number (9 for valid data) similarly to the SingleTact system interface electronics.

The USB system uses a USB mini B cable to connect to the PC. No additional wiring is required.

Only the number of USB ports available on your PC or laptop limits the maximum number of USB systems that can be used. Both USB 2.0 and USB 3.0 are supported by this system, although there will be no performance change between protocols.

2.1.3 Using a Tail Extender

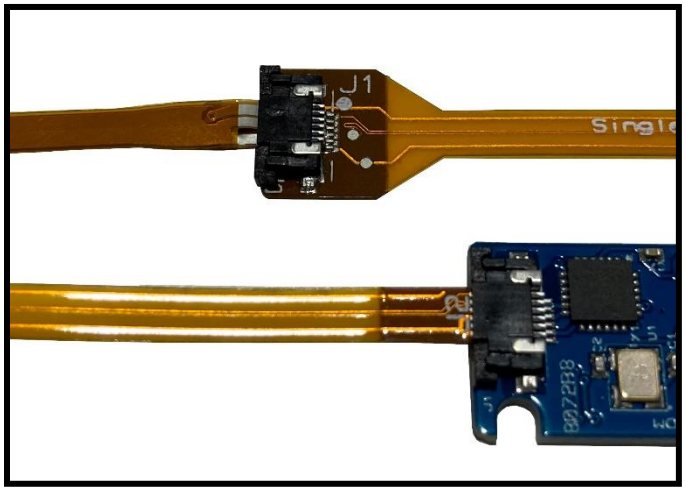
We understand that sometimes the 50 mm tail on the sensor is just not long enough for certain applications. As such, we offer a 150 mm (6”) tail extender to give your pressure sensing applications a longer reach.

The tail extender sits between the SingleTact pressure sensor and the Interface module of your choice, whether that be the System Interface Electronics or the USB turnkey solution.

NOTE: The tail extender is not backwards compatible with our, now obsolete, green system interface electronics. Only the current black interface modules, or USB systems, can be used with a tail extender. Refer to the SingleTact product list for details.

The sensors are connected to the tail extender, and the tail extender to the electronics modules as shown in Figure 7. The tail extender uses the same FFC socket as the electronics modules and the procedure for fixing is identical to that of the sensor to the electronics. Refer to the relevant module quick start guide for detailed procedure.

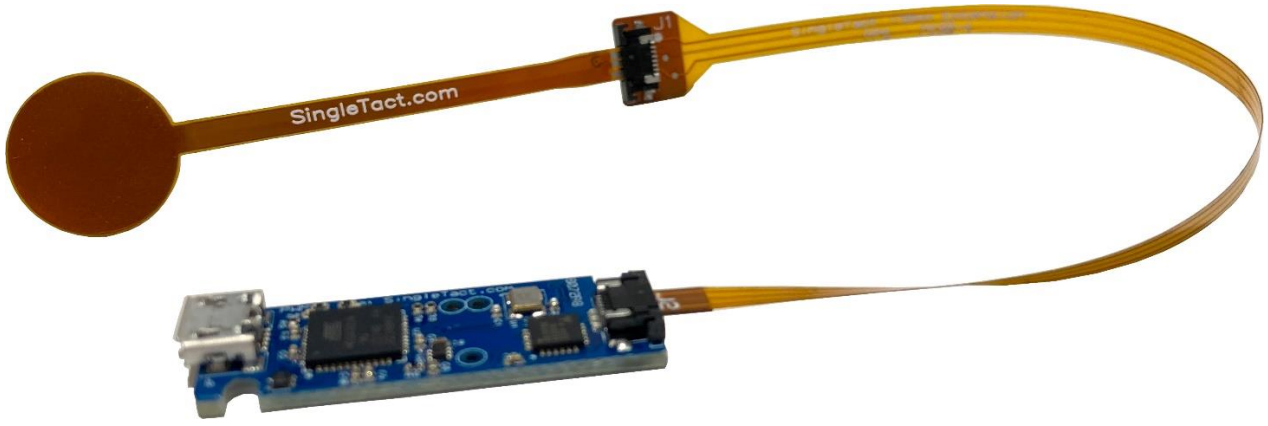
Figure 7 Tail Extender Connection



The tail extender can be bent, but not kinked, and twisted to contour to your particular needs. Care should be taken to avoid injecting interference into the central signal traces.

It is not advised to touch the tail extender during operation, or to use an earthed or grounded shield in close proximity to the tail extender as this can invalidate the results from the sensor.

Figure 8 Tail Extender Connection Example with USB System



2.2 Using the System Interface Analog Output

The analog output swings from **0** to **2 V**, with the valid working output ranging from **0.5 V** to **1.5 V** as shown in Figure 9.

As pressure increases beyond the full scale range (FSR) the output will increase to **2 V** and then saturate.

The sensor should be unloaded at power on to allow the sensor's baseline to be registered correctly.

NOTE: An output below **0.5 V** may indicate negative pressures, which occur when the sensing area is under tension. This should be avoided since it can damage the internal structure of the sensor.

NOTE: Sensor over pressure should be limited to less than 3x FSR to avoid damaging the sensor.

Figure 9 Analog Output

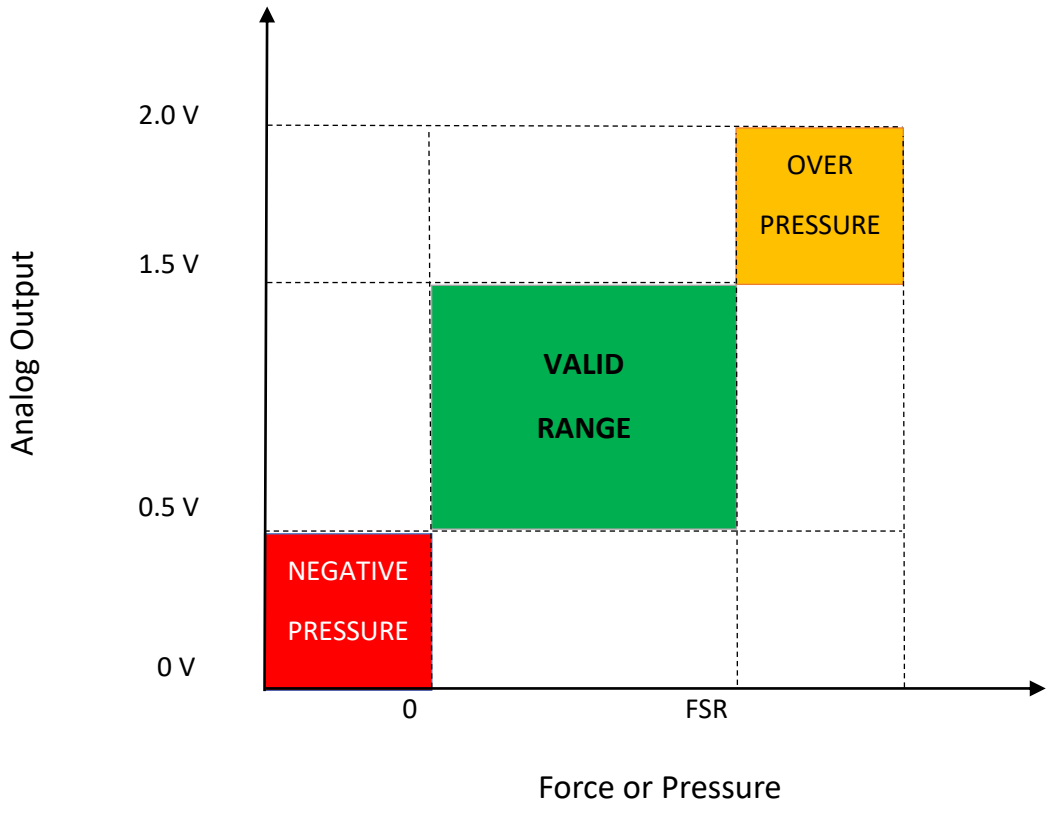


Figure 10 DAQ Connection Requirements

CONNECTION	PIN NUMBER	CONNECTION
No Connect	5 4	No Connect
No Connect	6 3	No Connect
No Connect	7 2	Analog Out
Ground	8 1	Vcc

2.3 Using the System Interface I²C Output

The SingleTact I²C interface supports the standard (**100 Kbits/s**) clock rate in **7-bit** address mode.

The SCL and SDA lines must be pulled up to the bus voltage which can be between **3 V** and **5 V**. Please refer to the I2C specification for bus protocol implementation & pull-up value considerations.

The interface board will always respond to two I²C addresses: **0x04** ***and*** the address specified in flash (register address 0). As shipped the default flash address is also **0x04.**

Multiple sensor interfaces may be connected to a single I²C bus. The bus address of individual sensor interfaces can be configured by writing desired address value (4 to 127) via the I²C interface to register address 0 with an I2C Write Operation. Change of individual sensor I²C addresses is supported by the PC and Arduino Example.

NOTE: As the interface board will always respond to address 0x04 then this address must be considered reserved for SingleTact. Where multiple SingleTact interfaces are to be connected to the same I²C bus then address 0x04 must be considered invalid and in this use case the configurable address of all connected SingleTact nodes must be individually changed from the default value before each SingleTact is added to the multi-node bus.

The SingleTact software architecture is based on a **192 byte** register block – see Figure 11 and Table 2 for details.

All control registers are located in first **112** bytes and get written to NVM when modified (and are therefore persistent after a power cycle). Configuration registers on Calibrated sensors interfaces are protected from modification.

The sensor results are available from bytes **128 to 133.** As shipped, results are updated at 50-120 Hz (this is dependent on capacitance sensor settings).

NOTE: It is the Users responsibility not to write to any Reserved locations.

Figure 11 Register Layout

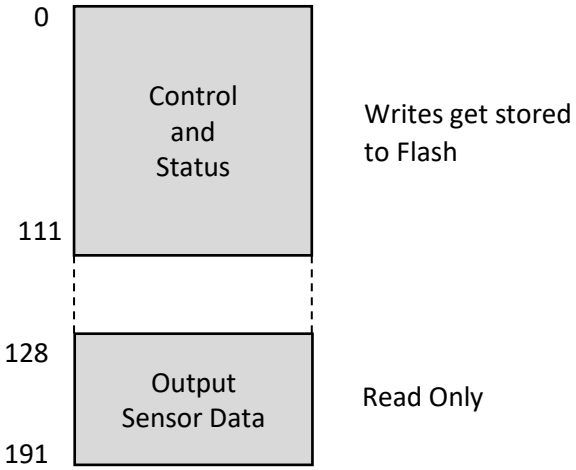


Table 2 Register Details

BYTE	SETTING
0	I^2C Address (4-127)
1	User configurable serial number MSB
2	User configurable serial number LSB
3	Reserved
4	Reserved
5	Capacitive Sense (Accumulator) Default 0x05 *1
6	Capacitive Sense (Reference Gain) Default 0x01 *1
7	Firmware Reversion
8	Capacitive Sense (Discharge Time) Default 0x03 *1

BYTE	SETTING
9	Capacitive Sense (Output Current) Default 0x00 *1
10	Output digital scaling MSB
11	Output digital scaling LSB
12	Number of elements, must be 1
13	Calibrated Sensor Identification. Calibrated sensor is 1, not calibrated is 0.
14	Delimiter – leave as 0xFF
15	First element to scan, set to 0
16-39	Reserved
40	Delimiter – leave as 0xFF
41	Sensor baseline MSB
42	Sensor baseline LSB
43-90	Reserved
91	Delimiter – leave as 0xFF
92-127	Reserved
128	Frame index MSB (increments on each new reading)
129	Frame index LSB (increments on each new reading)
130	Sensor Timestamp MSB (0.1ms increments) *1
131	Sensor Timestamp LSB (0.1ms increments) *1

BYTE	SETTING
132	Sensor output MSB
133	Sensor output LSB
134 - > 191	Reserved

**1 Should only be used as a coarse estimate as it is subject to drift.*

2.4 I²C Operations List

I²C SingleTact supports three I²C operations: Write, Read Request and Read.

2.4.1 I²C Write Operation

I²C bus Transfer: *Master Write to Slave*.

This command writes values to the register block. All writes also update the internal flash memory so settings are persistent through a power cycle.

Bytes **3** to **N-1** (where N is the packet length) contain the data to be written to consecutive registers.

Data may be written to the first **128** bytes (excluding reserved locations).

Writing outside of the valid range will fail.

Configuration registers on Calibrated sensors interfaces are protected from modification.

Table 3 Write Operation Data Packet Format

BYTE	TO SENSOR
0	0x02
1	Write offset in register block
2	Number of bytes to write (1 – 28)
3 -> (N-1)	Data to write (1 to 28 bytes)
N (max 31)	0xFF – end of packet delimiter

2.4.2 I²C Read Request Operation

I²C bus Transfer: *Master Write to Slave*.

This command sets the read location (register block offset) and read length for a following Read operation.

Table 4 Read Request Operation Data Packet Format

BYTE	TO SENSOR
0	0x01
1	Read offset in register block
2	Number or bytes to read (1 – 32)
3	0xFF – end of packet delimiter

2.4.3 I²C Read Operation

I²C bus Transfer: *Master Read from Slave*.

An I²C *Master Read from Slave* transfer can be used to directly read the register set and sensor data.

In normal operation a read of the two Sensor Output byte registers returns a 10-bit output range from 0000 to 0x3FF, corresponding to the 0 to 2 V analog output range.

The functional sensor FSR output range is 9-bits from x0100 to 0x2FF, corresponding to the 0.5 V to 1.5V analog output range. The larger 10-bit total range allows for detection of negative values when the sensor is under tension and some level of over pressure detection.

The sensor should be unloaded at power on to allow the sensor's baseline to be registered correctly.

Where a Read operation is not preceded by a Read Request operation the read location defaults to **128** (the sensor output location) and consecutive reads will therefore simply read the default 32 bytes of the sensor data region.

Where a Read operating is preceded by a Read Request operation then the register offset and read length as set by the Read Request will be used.

Data can be read from anywhere in the register block (addresses **0 – 191**).

Reading outside of the valid range will fail.

I²C slave Read operations simply return the register data values up to the number of requested bytes (32 max) in the data packet.

NOTE: A sensor output reading below **0x0100** may indicate negative pressures, which occur when the sensing area is under tension. This should be avoided since it can damage the internal structure of the sensor.

NOTE: Sensor over pressure should be limited to less than 3x FSR to avoid damaging the sensor.

Table 5 I²C Master Read from Slave Data Packet Format

BYTE	FROM SENSOR
0 - 31*	Register Data Read Location-Read Location+31*

* number of bytes read can be modified by a preceding read-request command.

2.5 Using the USB Turnkey Interface

The SingleTact USB Turnkey System is designed to be easy to use, and interface directly to a PC. As such, there are no registers, no wiring, and no hassle involved in using your SingleTact system in this way.

Follow the USB quick start guide to connect the sensor, with or without a tail extender, to the USB board and you are ready to start recording real data on our open source demo app.

To ensure your system is operating correctly before you use the demo app, or to troubleshoot issues, there is a PWM controlled LED on the board that responds linearly to pressure applied to the sensor.

There is no other user addressable output on the USB system.

The USB system will uses an ATMega32u4 as the USB controller, and can be differentiated from a SingleTact interface electronics module working through a similar microcontroller (as described in our typical use case) by a preceding header packet. The USB system will have a header packet of 4 bytes of 0xAA, the user implementation will typically be 4 bytes of 0xFF. The SingleTact demo app will detect this and inhibit user changes to be made to the calibrated USB system.

The demo app communicates with the USB system in exactly the same way as for a user implemented system, described in the demo app description.

Although the number of USB ports only limits the number of USB systems that can be used on a single PC, the configuration of ports on your PC can reduce the sensor sample rate from its nominal value due to USB bottlenecking. Additionally, when using large numbers of USB sensors, using a powered USB extender can improve sensor startup time.

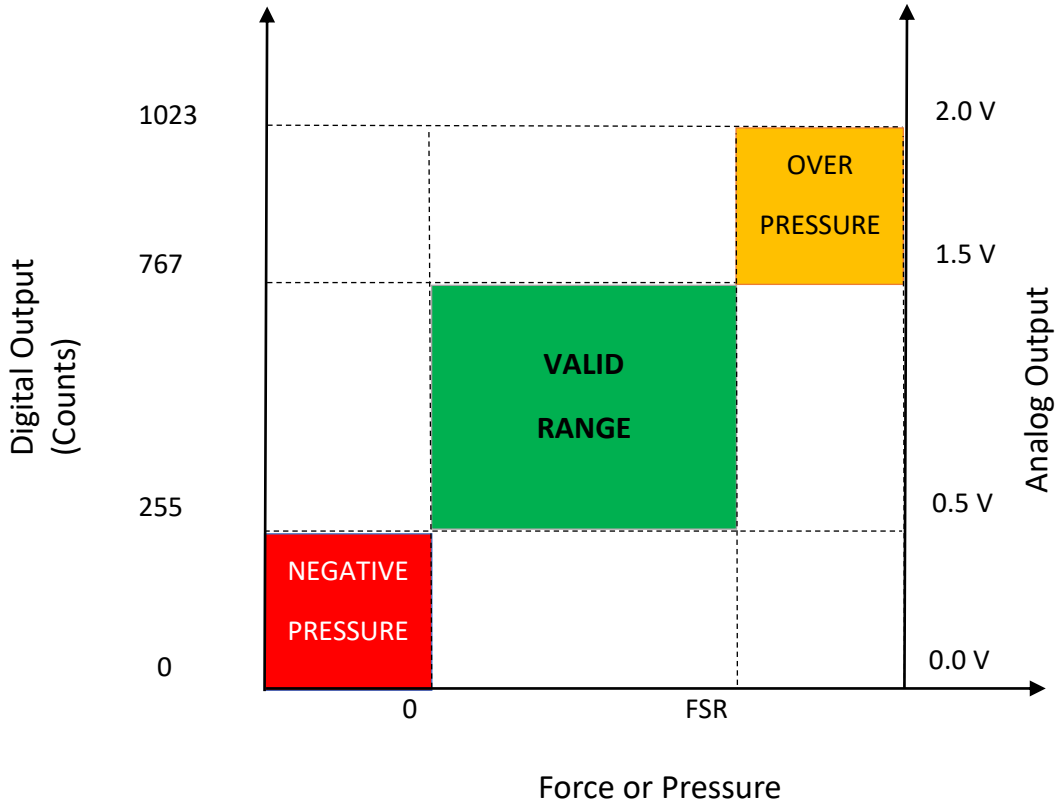
The output is 10 bit precision, with 9 bits used for useful valid data.

2.6 Load/Pressure Conversion Details

The SingleTact electronics interface, and USB system, measures the capacitive pressure sensor with **16-bit** precision. This is then scaled to a **10-bit** digital (0 V-**2 V** analog, for system interface module only) output using the following calculation:

$$\boldsymbol{SingleTact\ Output} = \frac{\boldsymbol{Raw\ capacitance - Baseline\ capacitance}}{\boldsymbol{Digital\ scaling\ value}} + \mathbf{255}$$

Figure 12 Digital Output



The Digital output from the System Interface electronics, or the USB system, is converted into applied load using the following equation:

$$\boldsymbol{Load\ (N)} = \frac{\boldsymbol{Digital\ Output\ (Counts) - Baseline\ Output(Counts)}}{\mathbf{512}\ \boldsymbol{(Counts)}} * \boldsymbol{Sensor\ Rating\ (N)}$$

The Analog output from the System Interface electronics is converted into applied load using the following equation:

$$\boldsymbol{Load\ (N)} = \frac{\boldsymbol{Analog\ Output\ (V) - Baseline\ Output(V)}}{\mathbf{1}\ \boldsymbol{(V)}} * \boldsymbol{Sensor\ Rating\ (N)}$$

Converting this load to pressure is accomplished by dividing the applied load by the sensor area (assuming the load is applied evenly over the full sensor head).

The digital scaling value is a 16-bit value stored at register locations 10 and 11 (see Table 2). For increased precision (within a given sensor's valid operating range) the digital scaling value can be adjusted in **0.01** increments using the demo app. A value of **100** represents unity scaling (100 x 0.01).

The internal capacitance to digital converter (CDC) operates at **140** to **4000** Hz depending on the capacitance sensor settings (in particular the number of accumulations).

Each time the CDC completes a measurement:

- the output register gets updated
- the frame index increases by one
- an active high pulse is produced on the frame synchronization output pin
- a timestamp is generated by SingleTact interface board (however as there is no crystal oscillator this should only be used as a coarse estimate).

The frame synchronization output (Frame Sync, Figure 5) goes high each time a new measurement is available. This can be used to synchronize the I^2C communications channel to the capacitance sensor.

Alternatively, the sensor might be polled as quickly as possible over I^2C. Since the frame index increments with each frame it can be used to identify duplicate or missing data readings.

2.7 Product Categories

Standard sensors will perform typically to their specified force range, as specified in the datasheet.

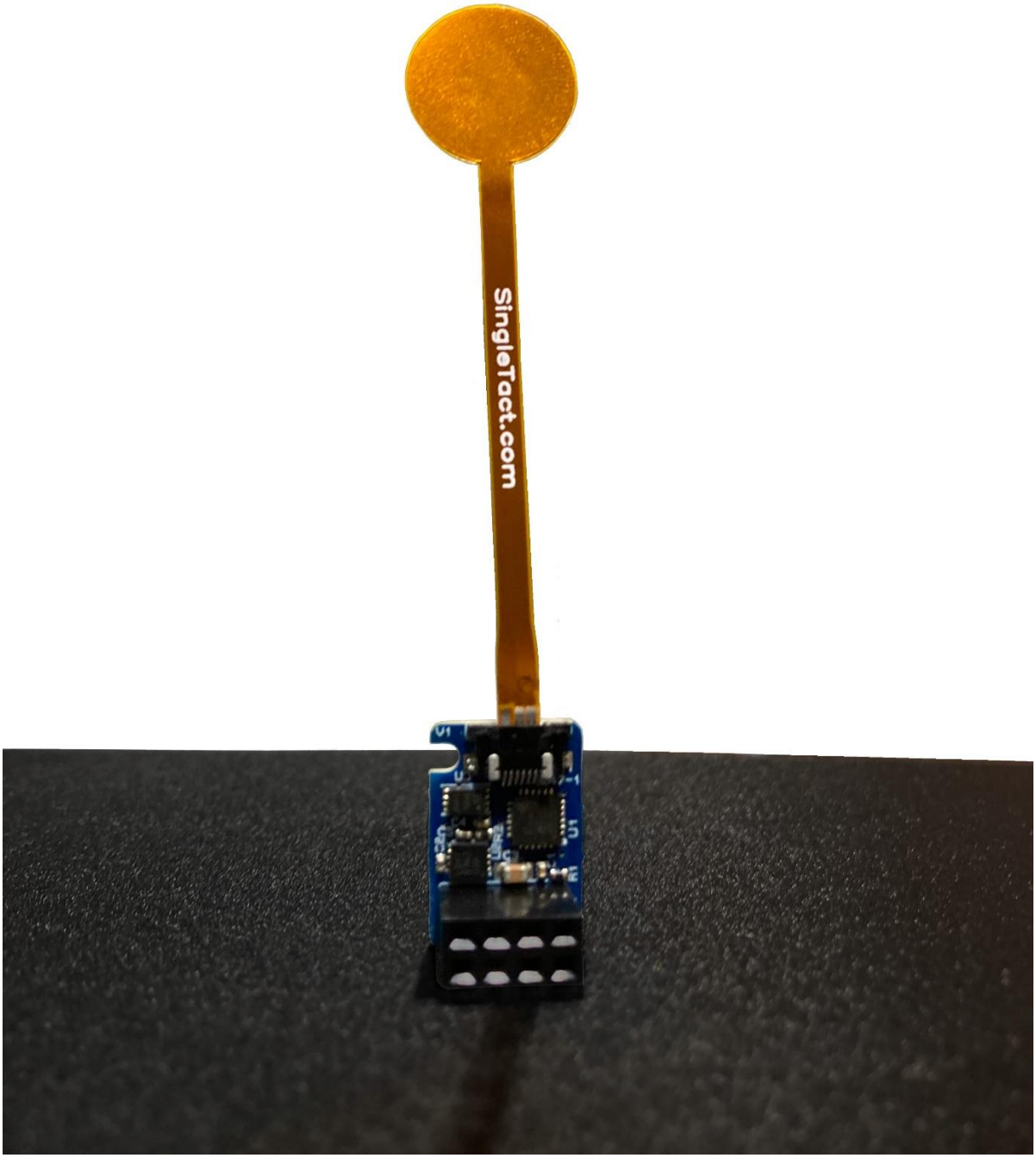
The separately available, generic electronics interface from PPS is suitable for use with the Standard sensors. With a standard sensor fitted and loaded to its specified maximum force (e.g. 45 N for a 45N sensor) the scaling factor may be configured via the I^2C bus to generate an I^2C sensor value equal to 511 and equivalent analog output of 1.5 V to ensure the sensor is operating within its valid range.

Calibrated sensors offer improved accuracy and linearity over the Standard sensors and come as a matched sensor plus electronics interface board or USB system providing a pre-configured, calibrated linear output for the specified sensor range. e.g. for an input force of 0 to 10 N a 10N calibrated sensor will output a linear I^2C range of 0 to 511 and equivalent analog output of 0.5 V to 1.5 V.

To maintain calibration, calibrated electronics + sensor combinations as shipped should be maintained as matched pairs.

To implement multiple sensor user implemented solutions, multiple sensor interface boards can be connected to a single I^2C bus as described in the I2C Interface section.

3 Using the SingleTact Demo App



3.1 Demo App Description

We have provided a SingleTact Demo App, a GUI for displaying and saving SingleTact sensor data on a PC or laptop. This is available from our websites download section.

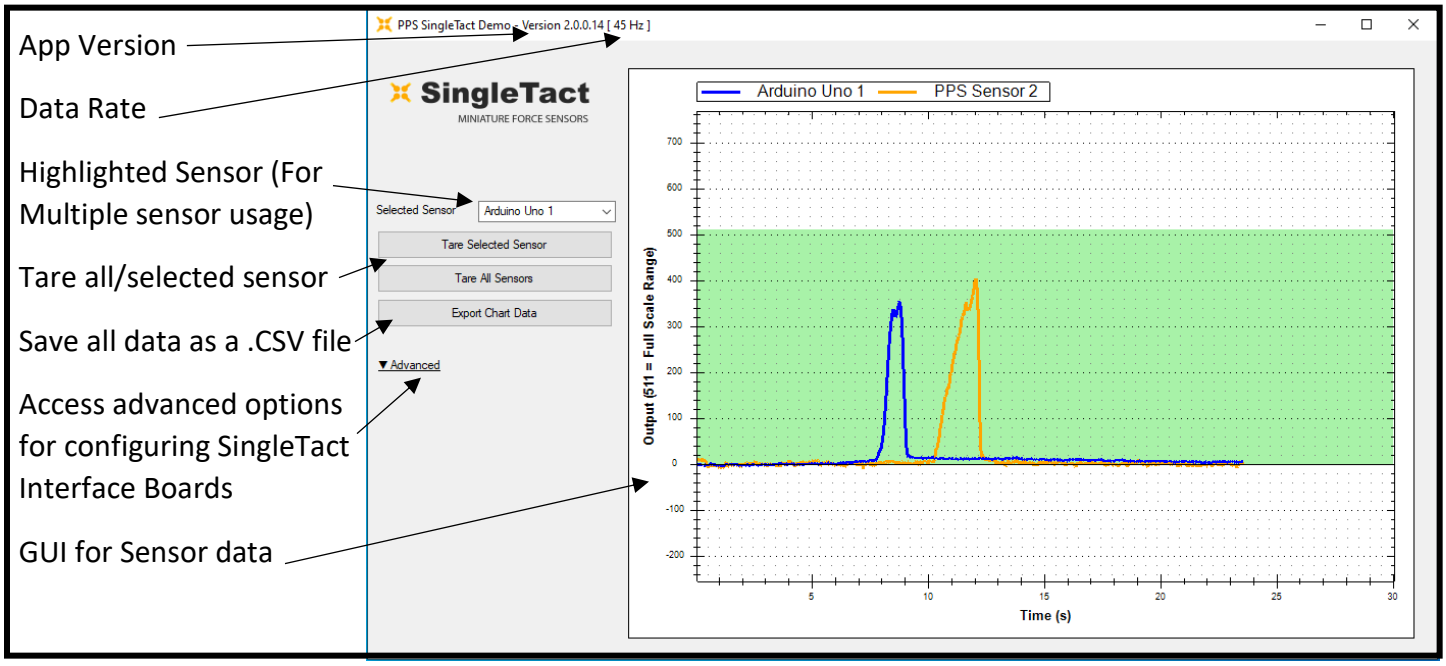
The software is available as a standalone executable for rapid setup, a Microsoft Store app (for those who prefer that platform), and as open source C# code for developers to make changes to as desired.

This section details the features and core operation of the SingleTact Demo App, and best practices for getting the most out of your Demo App.

The key features of the Demo App, shown in Figure 13, are:

- GUI for single or multiple SingleTact sensors, showing sensor data in real time
- Export sensor data and time series as .CSV for external analysis
- Differentiation of SingleTact USB system from System Interface Electronics, and differentiation of calibrated and non-calibrated SingleTact systems
- Automatic sensor legend (USB Systems are identified as PPS Sensor, Arduino based systems are identified as Arduino)
- Configuration options for SingleTact Interface Modules including the ability to change I^2C address and scale factors
- Baseline adjustments can be applied singly or globally by selecting the desired sensor(s)
- Safety features to stop accidental corruption of pre-calibrated sensor systems

Figure 13 - SingleTact Demo App General Features



3.2 Streaming and Saving Data

Streaming data on the SingleTact Demo App is simple. By following the QuickStart guide for your respective SingleTact system, and connecting your system appropriately, the SingleTact system can be connected to a PC or laptop via USB.

The demo app can be opened once a SingleTact sensor system is connected to the PC.

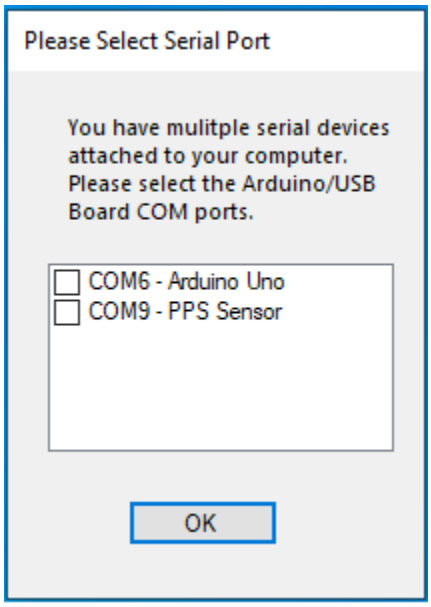
The demo app will automatically detect Arduino based systems (calibrated or otherwise) and SingleTact USB systems and immediately begin streaming data from all selected available sensors.

NOTE: The SingleTact demo app does not support multiple sensors on a single I^2C bus. The Demo app will support multiple USB systems, and/or multiple single sensor Arduino based systems whether calibrated or not.

When starting the demo app, if multiple sensors (regardless of type) are connected then you will have the option of selecting which sensors you wish to interface with. This is shown in Figure 14. SingleTact USB systems will be listed as PPS Sensors. Sensors connected via an Arduino will be listed as such. If a single SingleTact sensor system is used, then the demo app will start immediately.

NOTE: If unrelated Arduino based sensors are connected to the PC when the demo app is run, the user should select the COM port relating to the SingleTact sensor rather than that for other sensors.

Figure 14 - Startup selection of multiple sensor types



The demo app will then connect to the specified sensors and begin displaying data on the GUI. The GUI will display data from all selected sensors and allow the sensor data to be saved as a .CSV file, using the 'Export chart data' option on the GUI.

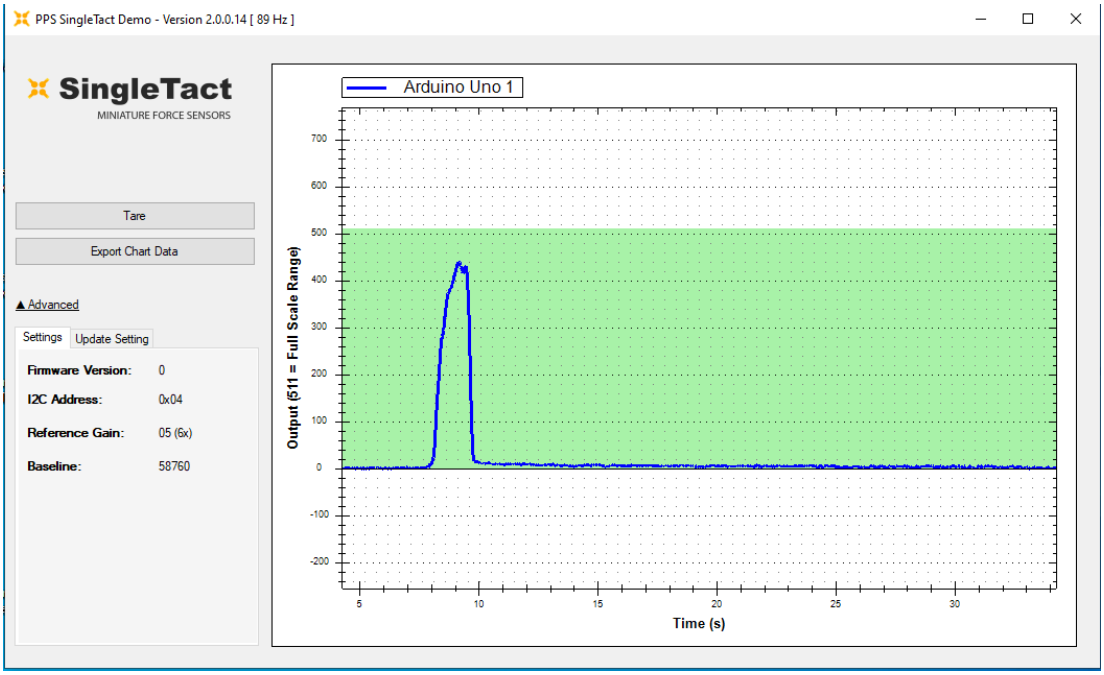
The maximum buffer length for recording data is 2 hours, plenty for the majority of user applications.

Data recording from sensors is interrupted during export to .CSV.

Advanced settings of the connected sensors can be viewed under the ‘Advanced Settings’ tab, shown in Figure 15. These settings include viewing the firmware version, I²C address (if applicable), and sensor parameters.

The scale on the GUI is a value between 0-511, where 511 corresponds to the FSR of the sensor being used.

Figure 15 - Advanced settings available on the GUI



3.3 Interfacing and Configuring the SingleTact Sensors

There are several additional options available on the SingleTact Demo App for configuring the SingleTact interface modules, allowing the I²C addresses to be configured as well as the scale factor for non-calibrated sensors. Additionally, the sensor offset can be managed using the ‘TARE’ option, operable on selected sensors or on all available sensors.

Taring, the act of removing the sensor offset (caused by drift or preloading), can be performed on a selected sensor or on all sensors simultaneously using the respective buttons on the GUI. This is shown in Figure 16.

Before performing a series of pressure measurements, it is advised to Tare all sensors prior to commencement to ensure accuracy.

The sensor configuring settings can be updated using the ‘Advanced Update’ option, shown in Figure 17. The I²C address of the SingleTact Interface Boards can be changed (for use in parallel sensor systems) using the dropdown selection provided.

Other configuration settings such as the scale factor and reference gain can be similarly updated.

Figure 16 - Sensor Taring

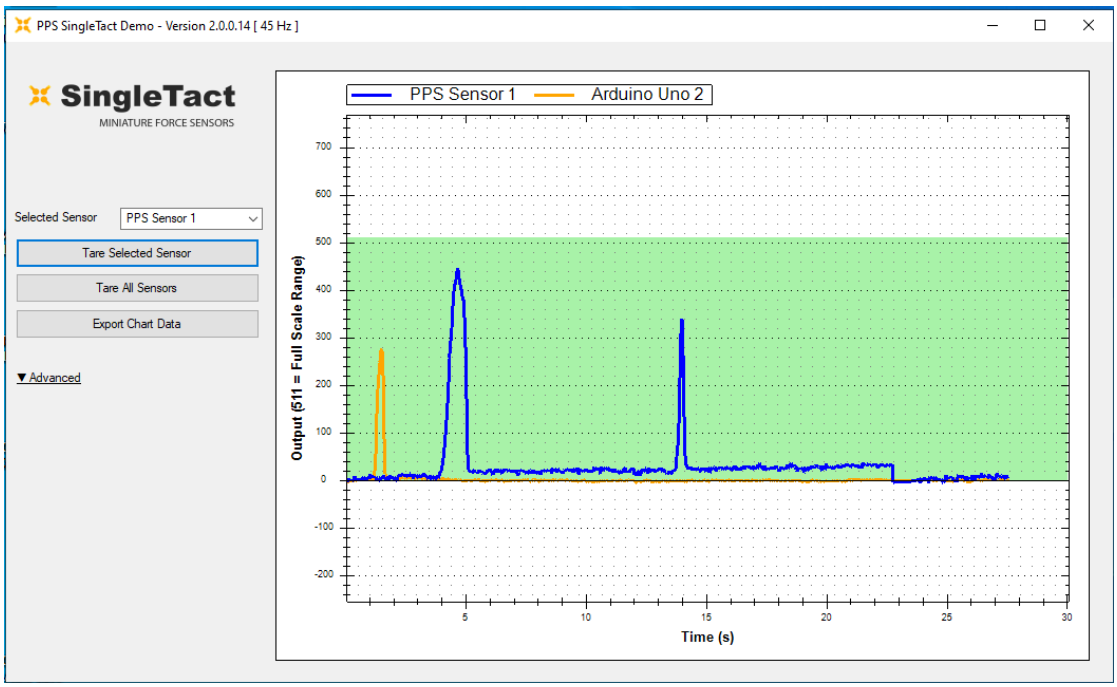
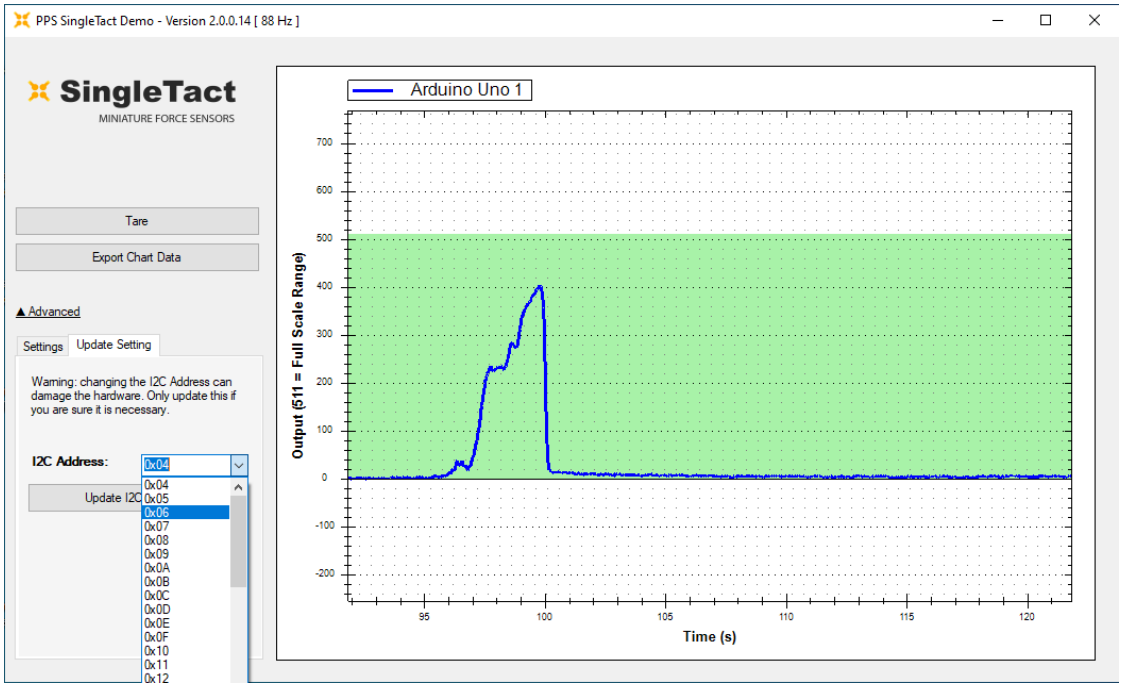
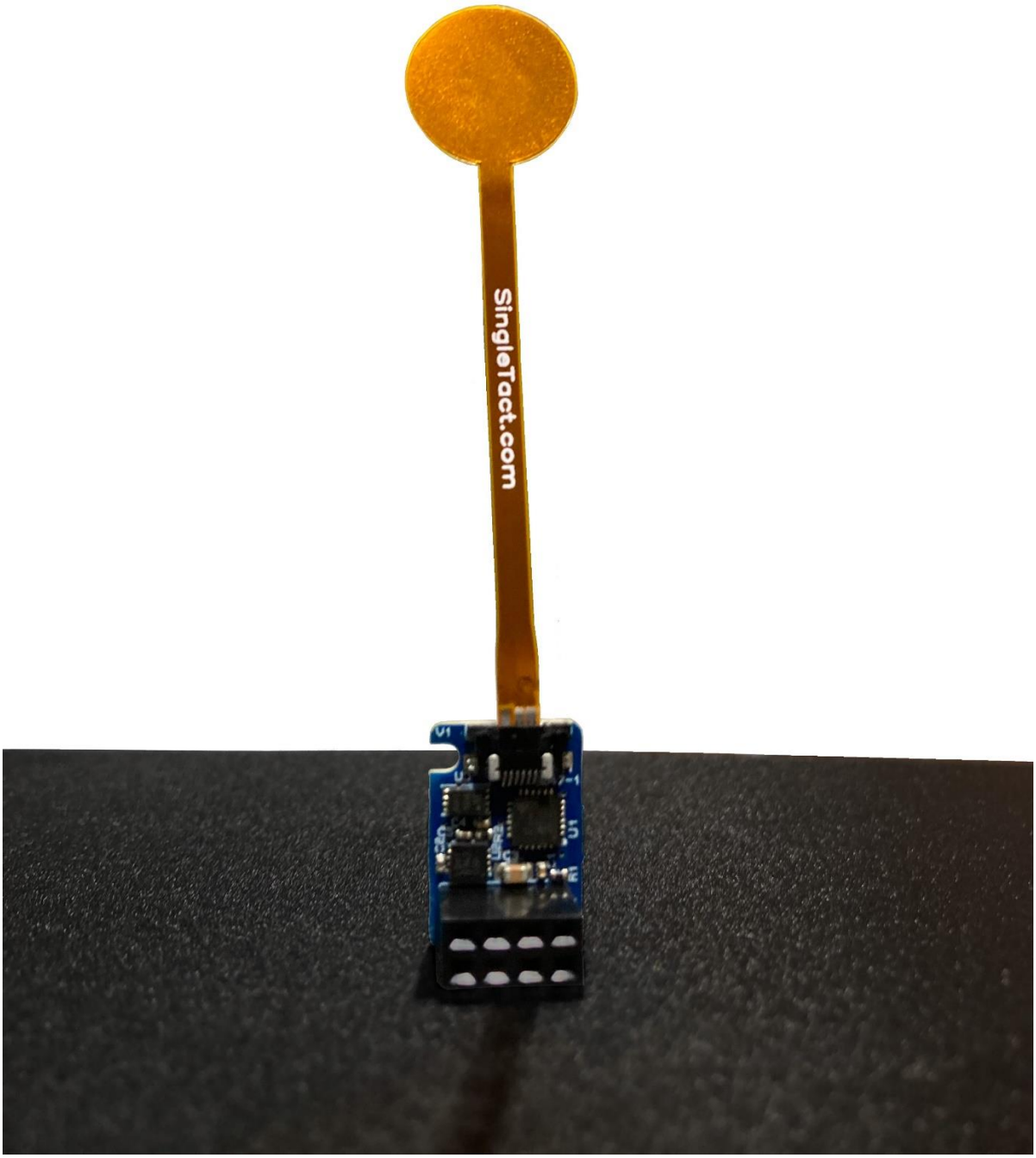


Figure 17 - Updating sensor settings



NOTE: Making changes using the advanced options should only be performed if necessary, and by a competent person. Mistakes made in configuring a sensor manually can damage the SingleTact sensor electronics.

4 TROUBLESHOOTING SingleTact



4.1 Arduino UNO not detected by PC.

Arduino UNO requires installation of a driver to communicate over the USB port.

Follow the step-by-step instruction from https://www.arduino.cc/en/Guide/HomePage.

4.2 Invalid setting error on PC (Interface Electronics)

Likely reason:

- Faulty pin connection on I^2C bus lines (ensure communication lines are made securely)
- USB port already in use (Close other applications using Arduino port)

4.3 Invalid setting error on PC (USB System)

Likely reason:

- USB port already in use (Close other applications using Arduino port)
- USB driver not found (automatically install required driver)

4.4 Persistent Invalid setting error on PC (All systems)

Likely reason:

- Computer compatibility issues (Run software in compatibility mode)
- Computer compatibility issues (Try using the demo app version available on the Microsoft store if issue is with the PC executable, or vice versa)
- Sensor electronics issue (Ensure you have not written data onto a reserved data register, contact SingleTact support)

4.5 No Analog output (remains at 0 V).

Check wire connections and ensure that you are powering the sensor module.

Likely reasons:

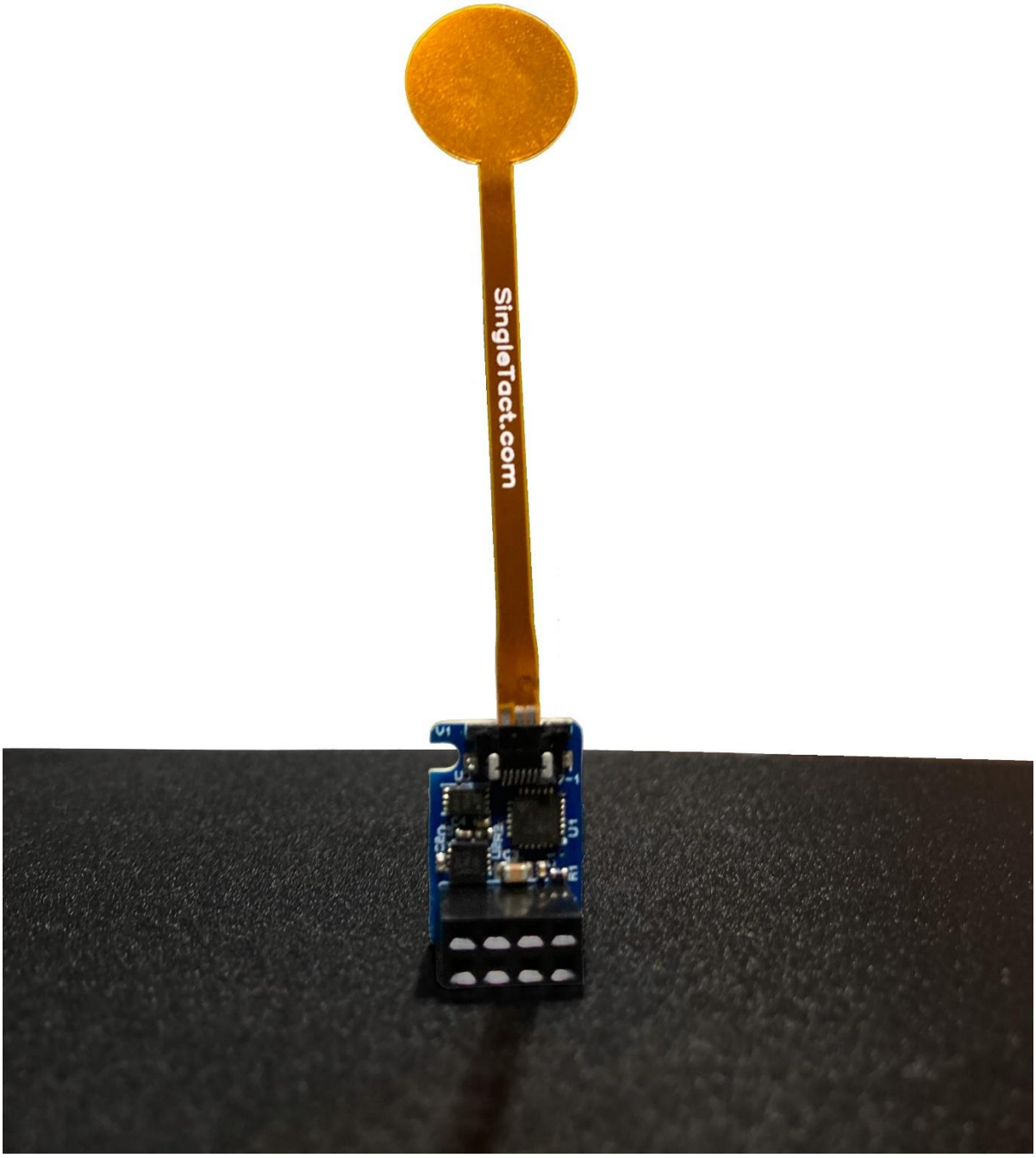
- Power, Ground or output connection in the wrong place.
- Power supply is off (Vcc below 3.7 V)

4.6 Analog output stays at 0.5 V.

Likely reason:

- Possible sensor fault.
  - Check sensor orientation (see Figure 4).
  - If in error analog output will stay at 0.49 - 0.5 V.
  - Digital output will remain on the baseline (0 counts).
- Testing high rating sensor at low loads (If sensor is 100 N rated, test it at 100 N rather than a gentle press to make voltage change detectable)

5 EXAMPLE USE CASE



5.1 PC and Arduino Example

An Arduino UNO board can be used to implement a USB serial interface to SingleTact.

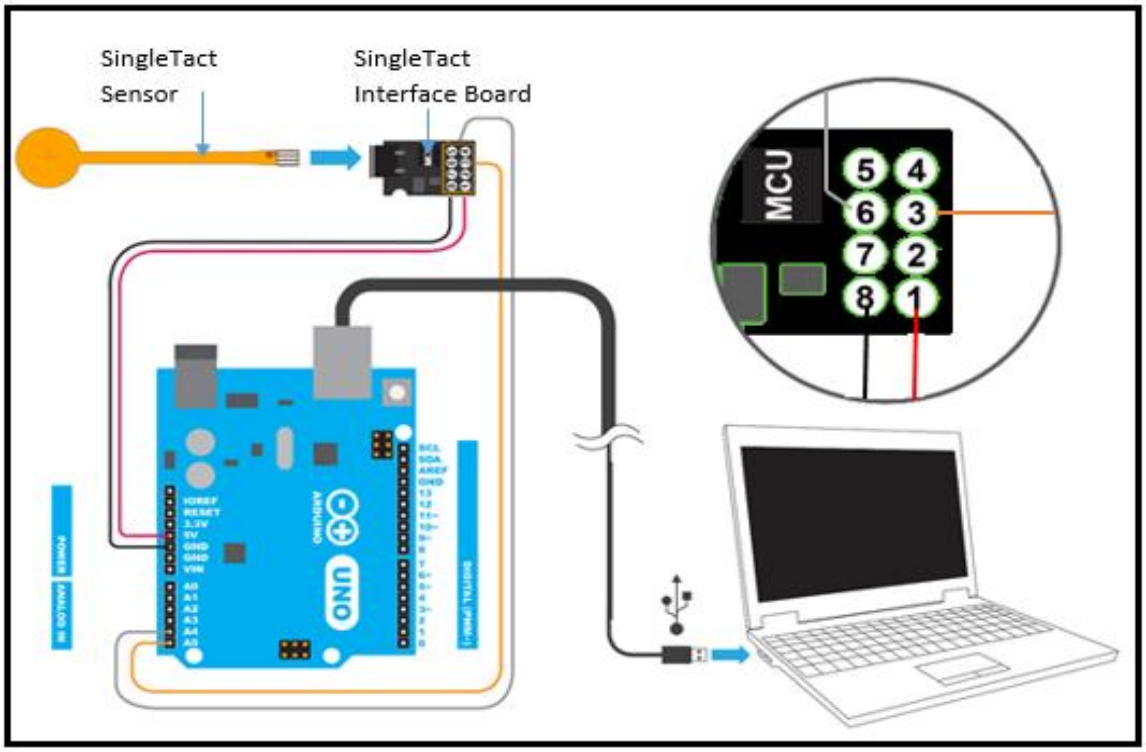
The code for an Arduino application (source) and an associated .NET based PC DAQ GUI application (both Windows executable and source) can be downloaded via www.singletact.com.

Once the Arduino board has been programmed with the SingleTact firmware (see Programming the Arduino UNO with SingleTact Example) the PC application can be run to visually observe the sensor results.

As the Arduino code is stored in flash the programming (or 'upload' in Arduino terms, only needs to be done once for a new board.

Refer to the relevant quick start guide for more details if required.

Figure 18 Arduino and SingleTact Assembly



Note: USB communication may need additional driver installation from Arduino software package. See https://www.arduino.cc/en/Guide/Windows#toc4 for further information.

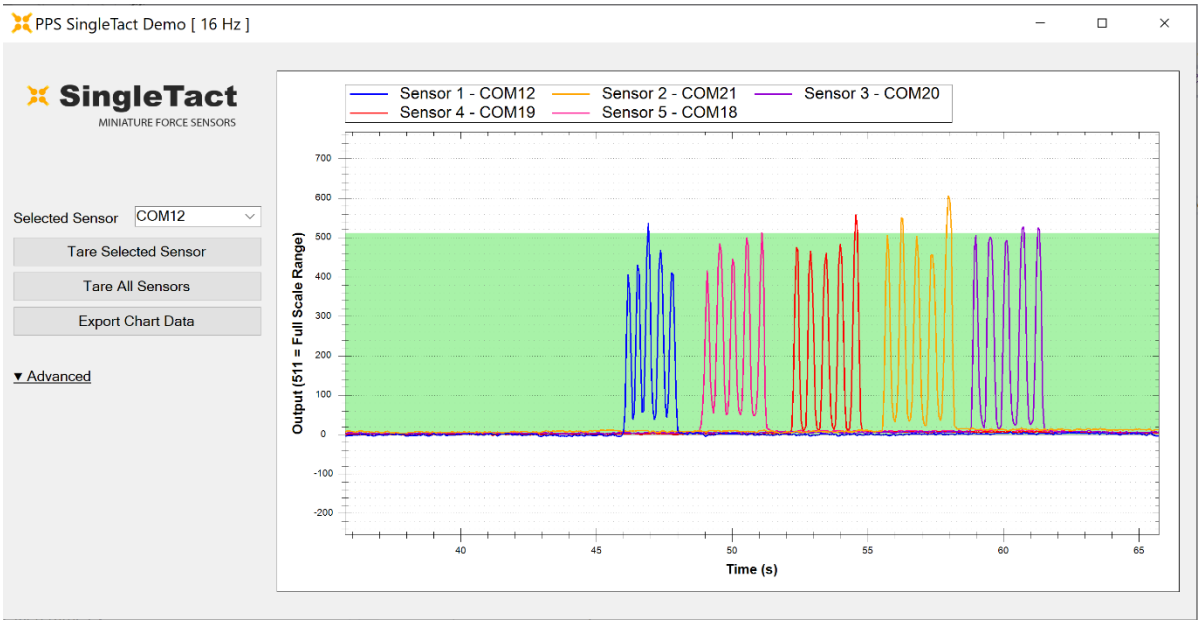
Figure 19 SingleTact and Arduino UNO Connection

CONNECTION	PIN NUMBER	CONNECTION
No Connect	5 4	No Connect
Arduino UNO pin A4	6 3	Arduino UNO pin A5
No Connect	7 2	No Connect
Arduino UNO GND pin	8 1	Arduino UNO 5 V pin

To run the Windows GUI application:

- Open the PCExecutable folder.
- Run SingleTact Demo.exe to bring up the demonstration application.

Figure 20 Demo of PC DAQ software



***Note**: Reference Gain will automatically change depending on the sensor size.

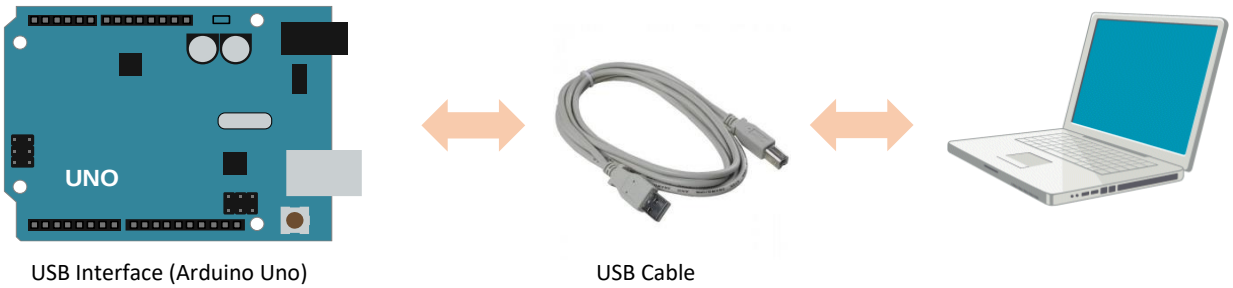
The PC application can be used to change the sensor's I^2C address and to modify its output scaling (SingleTact system interface module only). For more information on these settings please refer to the Using the System Interface I2C section.

5.2 Programming the Arduino UNO with SingleTact Example

This process outlines how to program the Arduino UNO with SingleTact example firmware.

1. Download and install the Arduino Software from: https://www.arduino.cc/en/Main/Software
2. Download the Arduino firmware (ExampleArduinoInterface) from: www.singletact.com
3. Connect the Arduino to the PC using the supplied USB cable.
4. Open the Arduino IDE software:

Figure 21 Arduino - PC connection



Note: USB communication may need additional driver installation from Arduino software package. See https://www.arduino.cc/en/Guide/Windows#toc4 for further information.

Follow the step-by-step instructions.

1. Go to File --->Open and open “SingleTactDemo.ino”
2. Go to Sketch --->Include Library --->Add .zip Library and select “Timer1.zip”
3. Go to Sketch --->Verify/Compile.
4. Go to Sketch --- > Upload*.

***Note:** If you receive an error on Upload make sure the Arduino is selected under Tools --->Port.

Figure 22 Arduino integrated development environment

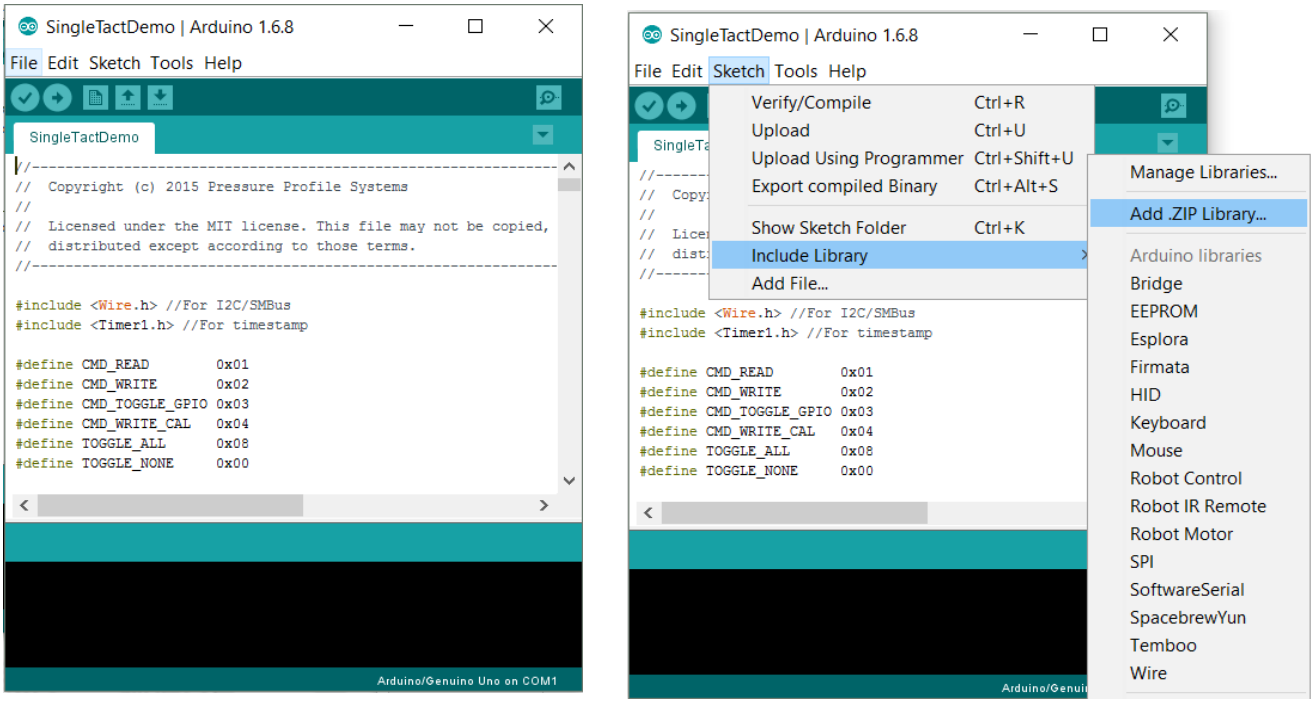
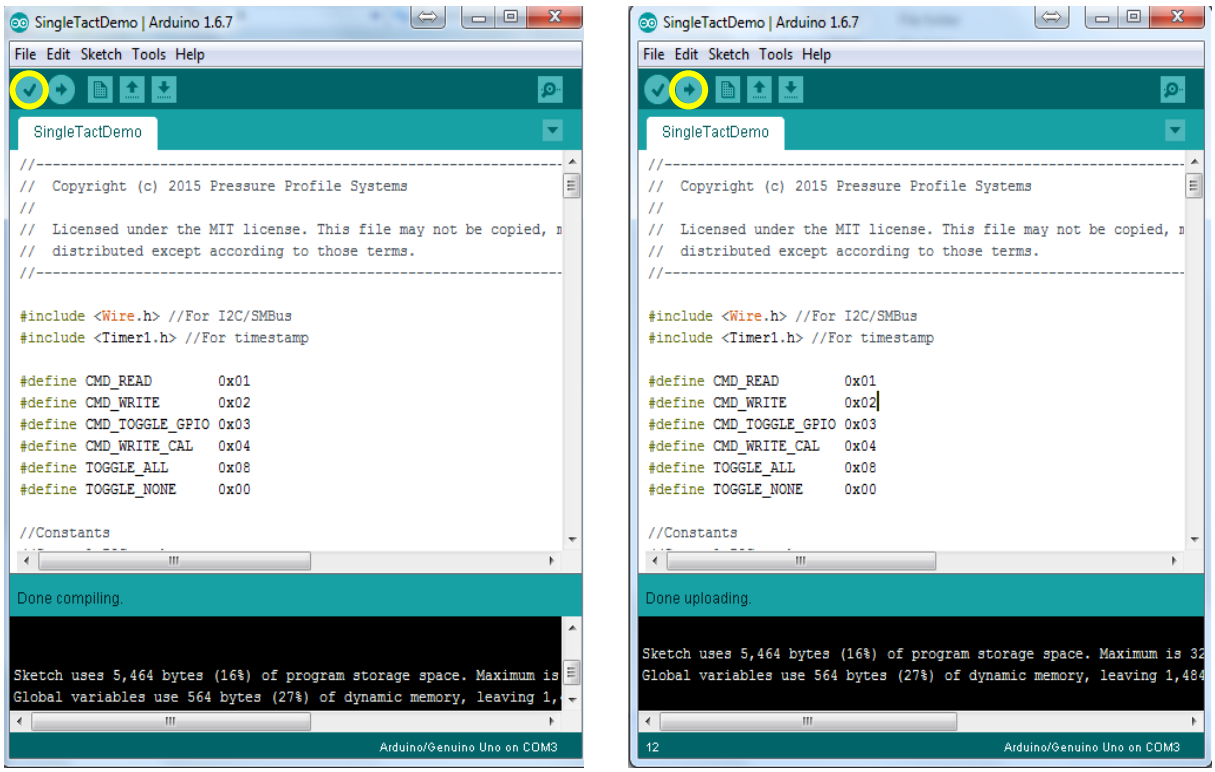


Figure 23 Compiling and uploading the SingleTactDemo.ino file



5.3 Arduino Demo Outline

The diagrams in this section provide an outline of the Arduino demo functionality as described in the previous section. In this case the PC to Arduino interface was setup to mirror the I²C interface, keeping the Arduino code as simple as possible.

Figure 24 Arduino example - communication architecture

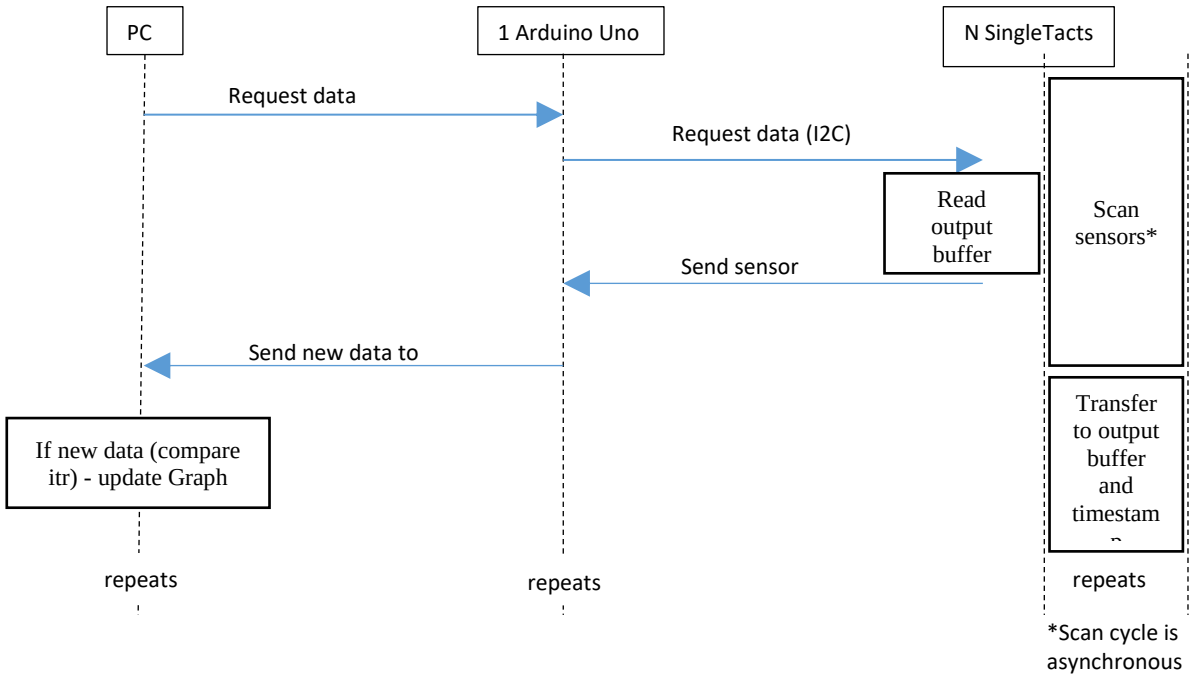
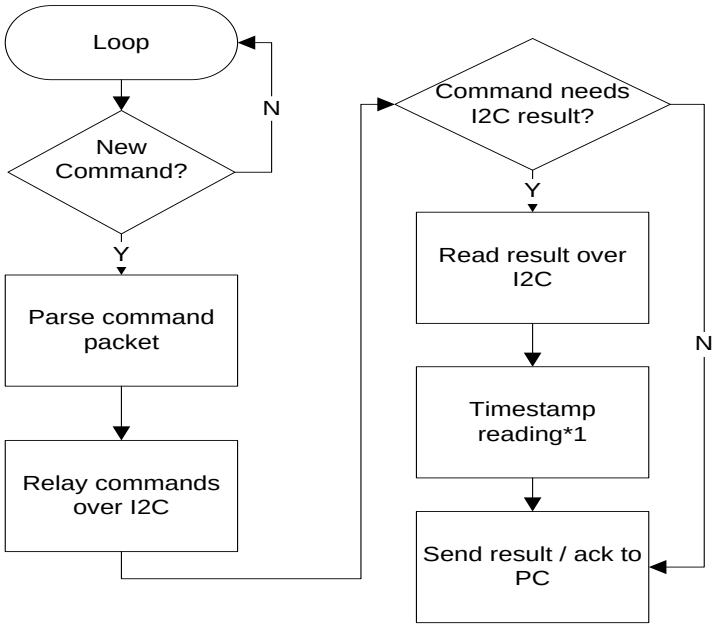


Figure 25 Arduino Application flow



(*1 Note: The Arduino contains a crystal oscillator so it is able to produce a more accurate time stamp than the SingleTact interface board.)

On the host, the Arduino appears as a virtual RS-232 serial device. Data is sent to/from the Arduino using a serial API, such as the one available in .NET.

The Arduino calculates a timestamp for each packet using the Arduino's crystal controlled oscillator. This can be used as the time for each sensor.

The serial commands, which mirror the raw I^2C commands (as shown in blue in Figure 26), are outlined in the following tables. Header and footer bytes are added to easily delimit serial packets. A timeout can be specified for I^2C transfers.

Figure 26 Serial packet structure (sent to Arduino)

BYTE	FROM PC TO ARDUINO
0	Header = 0xFF
1	Header = 0xFF
2	Header = 0xFF
3	Header = 0xFF
4	I^2C address of sensor
5	Timeout (in 100ms increments)
6	ID (echoed in reply)
7	Read (0x01) or Write (0x02)
8	Read/ write location
9	N bytes to read/ write (max 32)
10-> (10 + N-1)	Data to write 0 bytes for read request
11 + N	0xFF – signifies end of packet
12 + N	Footer = 0xFF
13 + N	Footer = 0xFF
14 + N	Footer = 0xFF
15 + N	Footer = 0xFF

Figure 27 Serial packet structure (sent from Arduino)

BYTE	FROM ARDUINO TO PC
0	Header = 0xFF
1	Header = 0xFF
2	Header = 0xFF
3	Header = 0xFF
5	1 if timeout exceeded
6	ID (echoed transmit ID)
7	Timestamp MSB
8	Timestamp
9	Timestamp
10	Timestamp LSB
11	N I^2C bytes to be sent (max 32)
12 -> 12+N	I^2C data
13+N	Footer = 0xFE
14+N	Footer = 0xFE
15+N	Footer = 0xFE
16+N	Footer = 0xFE

5.4 Example .NET API

This section provides some detail on the .NET API used to construct the PC GUI application.

Download the .NET Interface and demo application from www.singletact.com.

For convenience the low level PC interface is encapsulated in two .NET components.

1. ArduinoSingleTactDriver – The basic Arduino interface. The user must create one of these.
2. SingleTact – There can be multiple SingleTacts each with their own I^2C address.

Creating a SingleTact interface is as simple as:

```
arduinoSingleTactDriver.Initialise(COMport);          //Start Arduino driver
singleTact_.I2cAddressForCommunications = 0x04;       //Set I2C address
singleTact_.Initialise(arduinoSingleTactDriver);      //Start sensor
```

The sensor is read using the following:

```
SingleTactFrame newFrame = singleTact_.ReadSensorData(); //Get sensor data
   if (null != newFrame)                                  //If we have data
   {//Process result}
```

Settings can be pulled from the sensor using:

```
singleTact_.PullSettingsFromHardware();
```

and sent to the sensor using:

```
singleTact_.PushSettingsToHardware();
```

NOTE: The sensor settings can be modified using commands such as:

```
singleTact_.Settings.ReferenceGain = ###
```

6 Resources

SingleTact homepage
http://www.singletact.com/

I2C-bus specification and user manual VERSION 6, April 2014
http://www.nxp.com/documents/user_manual/UM10204.pdf

Arduino home
https://www.arduino.cc/

Microsoft .NET Framework
https://www.microsoft.com/net

7 Product Compatibility List

Product	Additions Notes	Status	Compatibility
Green SingleTact Interface	Including Calibrated versions (with respective sensor)	Obsolete	Sensors only (calibrated sensor only, if calibrated)
Black SingleTact Interface	Including Calibrated versions (with respective sensor)	Obsolete	Sensors + tail extender (calibrated sensor only, if calibrated)
Blue SingleTact Interface	Including Calibrated versions (with respective sensor)	Active	Sensors + tail extender (calibrated sensor only, if calibrated)
USB Turnkey System	Flexible Flat Cable (connector)	Active	Calibrated sensor only + tail extender
Tail Extender	150mm	Active	All except green interface modules
S8-1N	8mm diameter, 1N FSR	Active	All
S8-10N	8mm diameter, 10N FSR	Active	All

S8-100N	8mm diameter, 100N FSR	Active	All
Product	**Additions Notes**	**Status**	**Compatibility**
S15-4.5N	15mm diameter, 4.5N FSR	Active	All
S15-45N	15mm diameter, 45N FSR	Active	All
S15-450N	15mm diameter, 450N FSR	Active	All

8 Glossary

API	Application Program Interface
CDC	Capacitance to Digital Converter
DAQ	Data Acquisition
FFC	Flexible Flat Cable (connector)
FSR	Full Scale Range
I^2C	Inter IC bus
IDE	Integrated Development Environment
LSB	Least Significant Byte
MSB	Most Significant Byte
.NET	A Microsoft .NET software framework
NVM	Non-Volatile Memory
RS-232	A serial communications standard
PWM	Pulse Width Modulation
LED	Light Emitting Diode

9 Revision History

Revision 2.0

1) Removed Section 3 Updating the Interface Board.
2) Updated Table 1 accessibility to interface board design detail.

Revision 2.1

1) Added Revision History.
2) Section 1: Referenced Calibrated and Uncalibrated product options.
3) Figure 2: Fixed the link in footnote 3.
4) Table 1: Added I^2C Sensor Output range.
5) Table 2: Corrected the addresses of the following parameters:

40	Delimiter – leave as 0xFF
41	Sensor baseline MSB
42	Sensor baseline LSB

Were 39, 40, & 41 respectively.

6) Section 2.3 Removed reference to I^2C high speed mode. Updated detail on multiple sensor interfaces sharing a single I^2C bus
7) Section 2.4.3: Added detail on I^2C Sensor Output data values.
8) Section 2.5: corrected output scale resolution from 12-bit to 10-bit and clarified operating output values.
9) Added Section 2.6 Product Categories.
10) Figure 10: Removed Set Reference Gain control from the GUI image (the Reference Gain Setting is automatic in the current interface board design).

Revision 2.2

1) Fixed Table 1 to show correct maximum supply voltage (12V rather than 5V as stated)
2) Updated Copyright year to 2017

Revision 2.3

1) Clarified 255 digital offset and added Figure 8 to demonstrate.
2) Add manual version to page 1.

Revision 2.4

1) Added product notes for USB turnkey solution and tail extender (150mm)
2) Clarified distinction between USB board and system interface modules
3) Added component compatibility table.
4) Added load conversion equations.
5) Updated imagery for new black electronics and DAQ.
6) Renamed SingleTact electronics to System Interface Module (to differentiate from USB system).

7) Expanded Troubleshooting guide

Revision 2.5

1) Table 2: Corrected the addresses of the following parameters:

7	Firmware Reversion
13	Calibrated Sensor Identification. Calibrated sensor is 1, not calibrated is 0.

Revision 2.6

1) Added section 3 ‘Using the SingleTact Demo App’ detailing use and features of Demo App

Revision 2.9

1) Corrected electrical parameters table to be up 120 Hz refresh rate
2) Updated figures and diagrams to include the new blue boards
3) Added blue board to the compatibility table