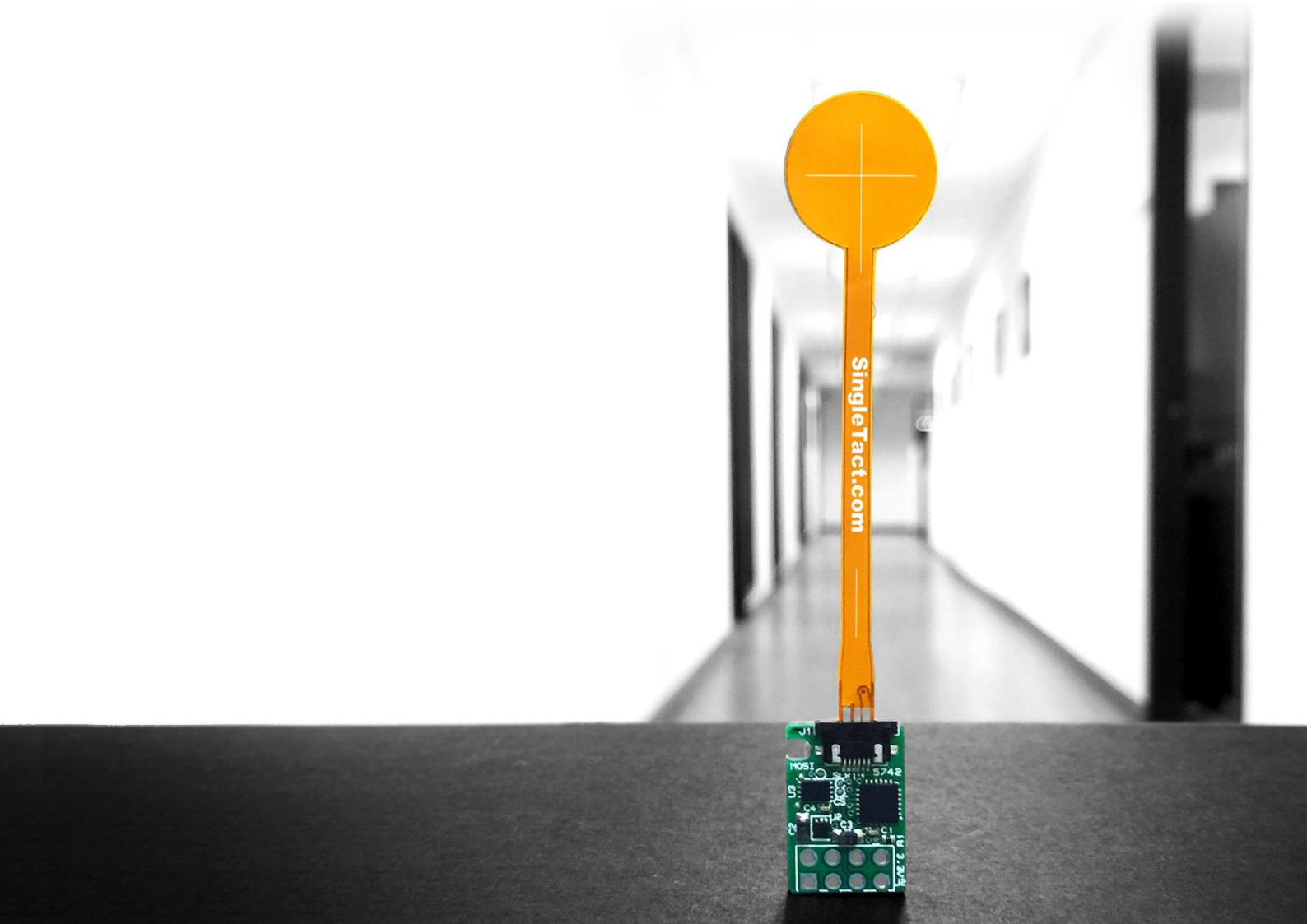
SingleTact 사용자 매뉴얼

MARCH 21, 2017

PRESSURE PROFILE SYSTEMS, INC.



목 차

1. 제품 설명

SingleTact는 단일 센싱 엘리먼트(Sensing Element)의 촉각 압력 센서로, 센서에 전달되는 물리적 압력을 측정하여 0.5 ~ 1.5V 의 아날로그 출력이나 I2C 인터페이스를 기반으로하는 디지털 출력으로 수치화된 압력 데이터를 제공합니다.

본 매뉴얼을 통해, 효율적인 SingleTact 활용을 위한 여러 방법을 간단히 확인할 수 있습니다. Arduino 디지털 인터페이스의 활용과 간단한 C# 기반 PC DAQ 소프트웨어 설정을 위한 여러가지 예제와 API(Application Programing Interface) 소스코드는 SingleTact 홈페이지(www.singletact.com)를 통해 신청(데모 소프트웨어)하거나 다운로드 받으실 수 있습니다.

그림 1. SingleTact 센서와 인터페이스 보드

그림 2. 인터페이스 보드 연결 설정

	용도	시스템 운용	
1.	압력 측정		아날로그 출력, 인디케이터, DAQ 장비, 오실로스코프
2.	평가 / 압력 측정[1]		Arduino, PC 데이터 수집/평가 시스템
3.	사용자 임의[2]		I²C 디지털 출력 인터페이스
4.	사용자 임의[3]		정전용량값을 사용자가 직접 수치화

1) 사용자가 임의로 제작한 소프트웨어 수트에 추가하여 사용해 볼 수 있도록 .NET 라이브러리를 본 매뉴얼 상에 상세히 설명해놓았습니다. .NET API 예시를 참조해 주십시오.

2) 하나의 I2C 버스로 약 100개의 SingleTact 인터페이스 보드를 지원할 수 있습니다. 인터페이스 보드 상의 펌웨어는 사용자의 용도에 맞게 변경/수정될 수 있습니다. 자세한 내용은 Wisetouch 고객 서비스나 PPS 고객서비스를 통해 확인 할 수 있습니다.

3) Wisetouch 고객 서비스나 PPS 고객서비스를 통해 자세히 안내해드립니다.

2. 인터페이스 디자인

2.1. 결선 방법

SingleTact 센서는 녹색 인터페이스 보드상의 FFC 커넥터와 연결됩니다. 인터페이스 보드와 케이블의 결선 방법은 그림 4를 확인해 주십시오.

전기적 파라미터는 표 1를 통해 확인할 수 있습니다.

그림 3. 센서를 인터페이스 보드에 연결

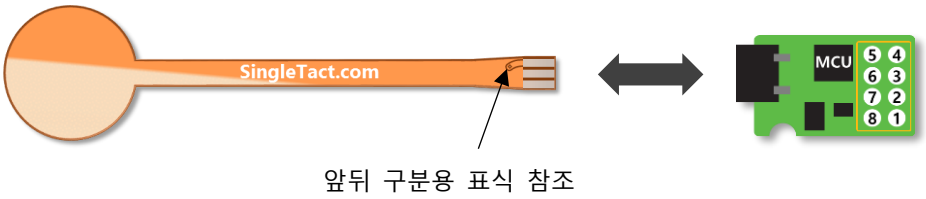


그림 4. 인터페이스 보드 헤더 결선

결선	핀 번호	결선
Reserved	5 4	Reserved
I²C 인터페이스(SDA)	6 3	I²C 인터페이스(SCL)
프레임 동기화	7 2	아날로그 출력
접지(Ground)	8 1	Vcc

표 1. 전기적 파라미터

파라미터	수치
전원 전압, Vcc	3.7 ~ 12V
I²C 응답속도	100KHz 또는 400KHz
I²C bus 레벨	3 ~ 5V
I²C 출력 범위 (센서 데이터)	10-bit (실제 운용시 FSR 출력: 9-bit)
아날로그 출력 범위	0 ~ 2V (실제 운용시 FSR 출력: 0.5 ~ 1.5V)
허용 아날로그 출력 부하	>5KΩ
Frame Sync 레벨	3.3V CMOS 출력
센서 갱신속도 (I²C 또는 아날로그 출력)	>140Hz (설정값에 따름)

2.2. 아날로그 인터페이스

아날로그 출력은 **0 ~ 2V** 범위 내에서 이뤄지며 실제 운용시 그림 5에서 보여지는 바와 같이, **0.5 ~ 1.5V** 출력값 범위를 사용하게 됩니다. 압력이 센서의 압력인지범위(FSR, Full Scale Range) 이상 까지 센서에 전달되면, 센서는 2V까지 출력값을 표시합니다. 이는 센서가 허용하는 압력의 한계를 넘어서고(과압, Over Pressure)있음을 뜻하며, **심한경우 이로인해 센서의 압력 기준점(Baseline)이 바뀔 수 있으므로 즉시 가압을 해제해야 합니다.**

참조: 비가압상태에서 출력이 0.5V 이하인 경우, 센서는 부압(負壓, Negative Pressure)을 표시하는 것으로, 과압(過壓, Over Pressure)으로 인한 기준점 오류나 극적인 온도 상승등으로 인해 센서 장치 내부에 있는 복원물질이 팽창했음을 뜻합니다. 이러한 현상은 영구적인 작동 불량을 일으킬 수 있기 때문에 가급적 피해야 합니다.

참조: 센서의 작동불량을 방지하기 위해, 과압(過壓, Over Pressure)은 센서의 압력인지범위(FSR)의 최대 3배 까지만 허용해야합니다.

그림 5. 아날로그 출력

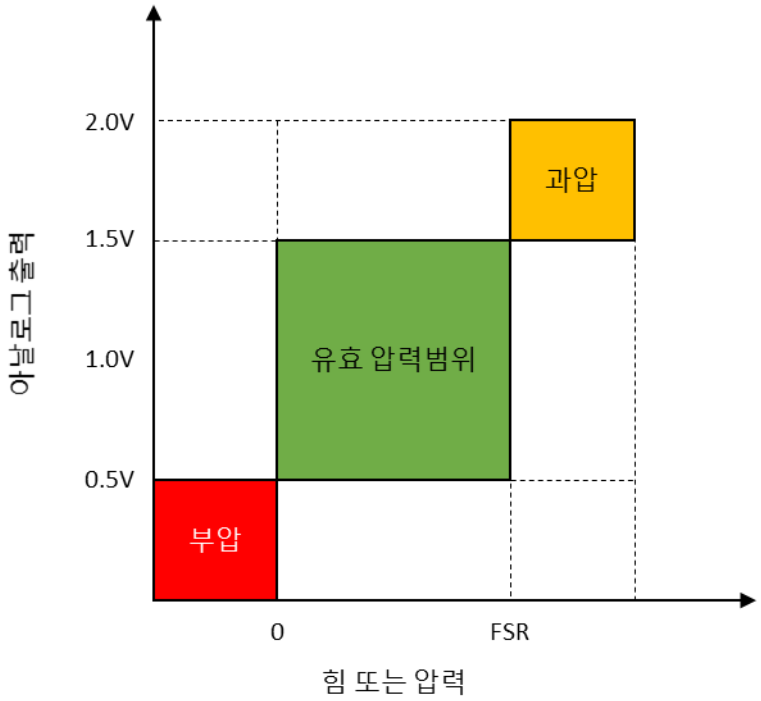


그림 6. DAQ 결선 요구사항

결선	핀 번호	결선
무결선	5 4	무결선
무결선	6 3	무결선
무결선	7 2	아날로그 출력
접지(Ground)	8 1	Vcc

2.3. I²C 인터페이스

SingleTact의 I²C 인터페이스는 7-bit 주소(address) 모드에서 표준클록속도(100 Kbits/s)를 지원합니다. SCL과 SDA라인은 반드시 3V에서 5V 사이의 버스 전압에 맞추어 풀업(pull-up) 해야 합니다. Bus 프로토콜 구현 및 풀업 값 고려사항에 대해서는 I²C 사양을 참조하십시오. 인터페이스 보드는 항상 두 개의 I²C 주소(0X04 및 플래시에서 지정된 주소, 레지스터 주소는 0)에 응답합니다. 제품의 출하시 기본 플래시 주소는 0x04로 설정되어 있습니다. 여러 개의 센서 인터페이스를 구성하는 경우도 단 하나의 I²C 버스에 연결 시킬수 있습니다. 각 센서 인터페이스의 버스 주소는 I²C 인터페이스를 통해 원하는 주소값(4 ~ 127)을 쓰고 I²C 연산으로 주소 0을 등록함으로써 구성 할 수 있습니다. 개별 센서의 I2C 주소 변경 방법은 본 매뉴얼 상의 PC 및 Arduino 예제에서 확인 가능합니다.

참조: 인터페이스 보드는 항상 주소 0x04에 응답하므로, 해당 주소는 SingleTact 전용으로 사용하기 바랍니다. 단 여러 개의 SingleTact 인터페이스가 동일한 I2C 버스에 연결될 경우, 각 SingleTact을 멀티노드버스에 추가하기에 앞서, 각 SingleTact 노드의 설정 가능 주소를 기본값에서 개별적인 설정값으로 변경해야 합니다.

SingleTact 소프트웨어는 192 바이트 레지스터 블록을 기반합니다. – 자세한 내용은 그림 7과 표 2를 참조.

모든 제어 레지스터는 최초 바이트 112에 위치하며 수정시 NVM에 기록됩니다. 이는 한번의 전원 주기(Power Circle) 이후에도 유지됩니다(전원을 껐다 켜도 유지).

센서 결과는 바이트 128에서 바이트 133에 위치합니다. 출하시, 결과는 >140Hz로 업데이트됩니다. (이는 커패시턴스 센서 설정에 따라 다름).

참조: 예약(Reserved)되지 않은 위치로 기록(Write)를 진행하는 경우, 이로인한 상황의 책임은 사용자가 전적으로 부담해야합니다.

그림 7. 레지스터 레이아웃

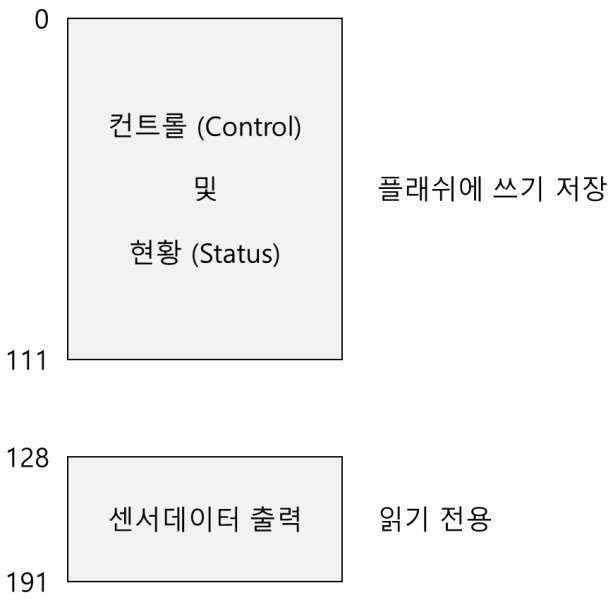


표 2. 레지스터 상세

바이트	설정
0	I2C 주소 (4 ~ 127)
1	유저설정 시리얼 번호 MSB
2	유저설정 시리얼 번호 LSB
3	이미 할당됨 (Reserved)
4	이미 할당됨 (Reserved)
5	정전용량 측정(Accumulator) 기본값 0x04 *1
6	정전용량 측정(Reference Gain) 기본값 0x01 *1
7	이미 할당됨 (Reserved)
8	정전용량 측정(Discharge Time) 기본값 0x03 *1
9	정전용량 측정(Output Current) 기본값 0x00 *1
10	Output digital scaling MSB
11	Output digital scaling LSB
12	엘레먼트 갯수 : 1개로 고정
13	이미 할당됨 (Reserved)
14	Delimiter : 0xFF
15	첫 스캔 엘리먼트 : 0 으로 설정됨
16 ~ 39	이미 할당됨 (Reserved)
40	Delimiter : 0xFF

41	Sensor baseline MSB
42	Sensor baseline LSB
43 ~ 90	이미 할당됨 (Reserved)
91	Delimiter : 0xFF
92 ~ 127	이미 할당됨 (Reserved)
128	Frame index MSB (신규 판독에 따른 증가분)
129	Frame index LSB (신규 판독에 따른 증가분)
130	Sensor Timestamp MSB (증가분 0.1ms) *1
131	Sensor Timestamp MSB (증가분 0.1ms) *1
132	센서 출력 MSB
133	센서 출력 LSB
134 ~ 191	이미 할당됨 (Reserved)

*1 상시 표류(Drift)하기 때문에 대략적인 수치로 적용함

2.4. I2C 운용 방법

I2C SingleTact은 3가지의 I2C 운용(판독, 요청, 기록)을 지원합니다.

2.4.1. I2C 기록 작업

I2C 버스 전송: 슬레이브에 마스터 기록

해당 명령으로 레지스터 블록에 대한 기록를 실행할 수 있으며, 모든 기록은 내부 플레시 메모리를 또한 업데이트합니다. 이 설정은 한 전원 주기(Power Circle) 동안 유지됩니다.

바이트 3 부터 바이트 N-1(N은 패킷 길이)까지의 구간은 연속(Consecutive) 레지스터에 기록 될 데이터를 포함합니다.

데이터는 처음 128 바이트 까지 기록 될 수 있으며, 유효 범위를 벗어나는 기록은 실패하게 됩니다.

보정된 센서 인터페이스의 경우, 별도 구성된 레지스터의 추가 수정이 불가하도록 보호되어 있습니다.

표 3. 기록용 데이터 패킷 포멧(Write Operation Packet Format)

바이트	센서
0	0x02
1	레지스터 블록상의 기록 오프셋
2	기록(Write)에 사용되는 바이트 갯수 (1 ~ 28)
3 ~ (N-1)	기록(Write) 데이터 (1 ~ 28 바이트)
N (최대 31)	0xFF : 패킷 디리미터(Delimiter)의 종료 지점

2.4.2. I2C 판독 요청

I2C 버스 전송 모드: Slave에서 Master 기록로 전환

해당 명령은 다음과 같은 판독에 대한 판독 위치(레지스터 블록 오프셋)와 판독 길이를 설정합니다.

표 4. 판독 요청 데이터 패킷 형식

바이트	센서
0	0x01
1	레지스터 블록상의 판독 오프셋
2	판독(Read)에 사용되는 바이트 갯수 (1 ~ 32)
3	0xFF : 패킷 디리미터(Delimiter)의 종료 지점

2.4.3. I2C 판독 작업

I2C 버스 전송 모드: Slave에서 Master 기록로 전환

슬레이브 전송에서 I2C 마스터 판독를 사용하여 레지스터 세트 및 센서 데이터를 직접 판독할 수 있습니다.

정상 작동시 두 개의 센서 출력 바이트 레지스터를 읽으면 0 ~ 2V 아날로그 출력범위에 해당하는 000 ~ 0x3FF의 10 비트 출력 범위가 반환됩니다.

센서의 인지압력범위(FSR, Full Scale Range)에 대한 기본 출력범위는 x0100에서 0x2FF의 9 비트이며, 이는 아날로그 출력의 0.5V ~ 1.5V에 해당합니다. 보다 큰 범위인 10 비트 출력의 경우, 센서가 장력의 영향을 받을 때, 음의 값을 출력할 수 있도록 합니다.

센서 영점(Baseline)의 올바른 설정을 위해서, 센서에 전원이 공급되기 전에, 센서상

의 추가 가압 요소를 미리 제거하는 것이 좋습니다.

판독 작업 요청이 없는 경우, 판독 위치의 기본값은 128(센서 출력 위치)으로 간주되며, 따라서 연속적인 판독 작업의 경우, 센서 데이터 영역의 기본 32 바이트를 읽게 됩니다.

판독이 판독 요구에 선행되는 경우, 판독 요구에 의해 설정된 레지스터 오프셋 및 판독 길이가 사용됩니다. 레지스터 블록 (주소 0 - 191) 상의 어디에서나 데이터를 판독할 수 있습니다. pg. 유효한 범위를 벗어나는 읽기는 실패합니다. I 2C 슬레이브 읽기 동작은 레지스터 데이터 값을 데이터 패킷의 요청 된 바이트 수 (최대 32 개)까지 반환합니다.

유효 범위를 벗어는 판독은 작동하지 않습니다.

I2C 슬레이브 읽기 동작은 레지스터 데이터 값을 데이터 패킷의 요청 된 바이트 수 (최대 32 개)까지 반환합니다.

참고 : 센서가 장력의 영향을 받고있는 경우, 0x0100 이하의 센서 출력을 표시하며 이는 센서의 내부 구조를 손상시킬 수 있으므로 피해야합니다.

참고 : 센서의 손상을 방지하려면 센서의 과압을 FSR의 3배 이하로 제한해야합니다.

표 5. 슬레이브 데이터 패킷 포멧에서 I2C 마스터 판독

바이트	센서
0 ~ 31 *1	레지스터 데이터 판독 로케이션 - 판독 로세이션 + 31 *1

*1 : 선행 판독 요청 명령으로 판독할 수 있는 바이트 수를 수정 가능

2.5. 신호의 변환(Conversion)

SingleTact의 인터페이스 보드는 16비트의 정밀도로 정전용량방식 센서를 측정합니다. 다음과 같은 공식을 통해 추출된 수치는 10비트 디지털(혹은 2V 아날로그) 신호로 자동 변환되여 출력됩니다.

$$SingleTact\ \ 출력 = \frac{초기\ 정전용량값\ -\ 정전용량\ 변화량}{디지탈\ 조정\ 수치}$$

디지털 조정 수치(Digital scaling value)는 레지스터 위치 10 및 11에 저장된 16비트 값입니다(표 2참조). 정확도를 높이기 위해(기존 센서의 유효 작동 범위에서) 디지털 조정 수치를 0.01 단위로 조정할 수 있습니다. 100은 단위 조정 (100 x 0.01)을 뜻합니다.

디지털 변환기(CDC)에 대한 내부 정전용량값은 정전용량 센서 설정(특히, 누적 횟수)에 따

라 140 ~ 4000 Hz에서 작동합니다.

CDC가 각 측정을 완료할 때,

- 출력 레지스터가 업데이트 됩니다.
- 프레임 인덱스가 1씩 증가합니다.
- 엑티브 하이 펄스(Active High Pulse)가 프레임 동기화 출력핀에서 생성됩니다.
- SingleTact 인터페이스 보드에 의해 타임 스탬프(Time-Stamp)가 생성됩니다(그러나 크리스탈 오실리에이터가 없으므로 대략적인 추정치로만 사용해야함).

프레임 동기화 출력(그림 4. 참조)은 측정이 새로이 갱신될 때마다 높아지는데, 이를통해, I2C 통신채널과 정전용량 센서를 동기화 할 수 있습니다.

다른 방법으로, I2C를 통해 센서를 신속히 폴링(Polling)할 수 있습니다. 실제로 프레임 인덱스는 각 프레임마다 증가기 때문에, 중복되거나 누락 된 데이터 판독 값을 구별하는 데 사용될 수 있습니다.

2.6. 제품의 분류

일반형 센서 제품군은 몇가지 압력범위로 구분되어 제공되고 있습니다.

SingleTact 인터페이스 보드는 일반 SingleTact 센서를 포함한 간단한 커스텀 디자인의 센서와 호환되어 사용할 수 있도록 설계되었습니다. FSR에 맞춰 제작된 각 센서(예를들어, 45N 제품군은 최대 압력 45N에 맞추어 설계됨)와 연결된 인터페이스 보드는, I2C 버스를 통해 구성된 조정인수(Scaling Factor)로 511과 같은 I2C 센서 값과 1.5V의 등가 아날로그 출력을 생성하여 센서가 유효한 범위 내에서 정상적인 작동을 할 수 있도록 합니다..

따라서, 보다 향상된 출력 정확도를 위해 센서의 기본 압력 범위를 사전에 반영한 인터페이스 보드를 함께 제공하고 있습니다. 0 ~ 10N의 압력에 대해 10N으로 FSR 보정된 센서는 0 ~ 511의 선형 I2C 범위와 0.5V ~ 1.5V의 아날로그 신호를 출력합니다.

본 매뉴얼 상에 설명된 I2C 인터페이스 내용과 같이, 여러 센서 인터페이스 보드를 단일 I2C 버스에 연결하는 것으로 다중 센서 솔루션을 구현할 수 있습니다.

3. 문제 해결

3.1. PC에서 Arduino UNO 인식 문제

- USB포트를 통한 통신을 위해서, Arduino UNO의 드라이버 설치를 우선 진행해야합니다
- 상세 순서 : http://www.arduino.cc/guite/homepages

3.2. 설정오류 메세지(Failed to set)

- 핀 결선이 잘못된 경우에 발생합니다. 핀 결선을 확인해 주십시오.

3.3. 아날로그 출력 오류 – 출력값이 0V에 고정

- 케이블과 케이블의 결선 상태를 확인해 주십시오.
- 전원에 문제가 없는지 확인해 주십시오.

3.4. 아날로그 출력 오류 – 출력값이 0.5V에 고정

센서와 인터페이스 보드가 서로 반대로 체결된 경우, 아날로그 출력은 0.49 ~ 0.5V, 디지탈 출력은 영접에 고정(0 카운트)되어 나타납니다.

- 센서와 인터페이스 보드 연결 방향을 확인해 주십시오. 그림 3 참조.

4. 사용 예시

4.1. PC 와 Arduino 연결

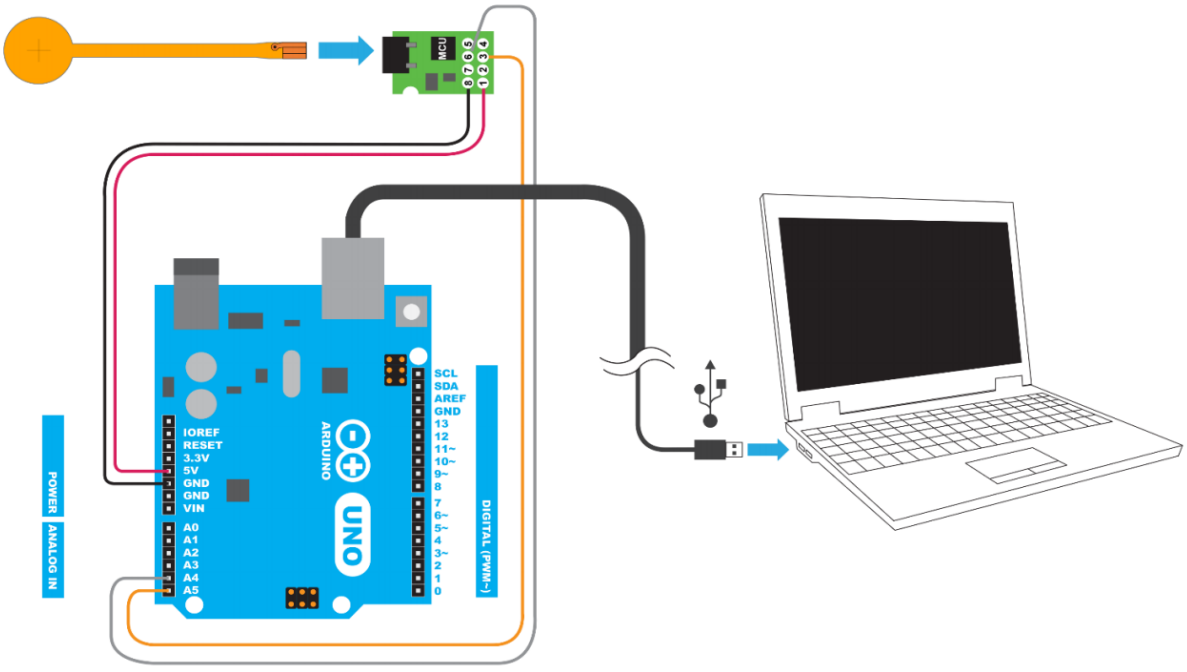
Arduino UNO 보드를 사용하여 SingleTact에 대한 USB 직렬 인터페이스를 구현할 수 있습니다.

Arduino 응용 프로그램(소스) 및 관련 .NET 기반 PC DAQ GUI 응용 프로그램용 코드(Windows용 실행 파일과 소스 포함)는 www.singletact.com을 통해 다운로드 할 수 있습니다.

Arduino 보드가 SingleTact 펌웨어로 프로그래밍되면 PC 응용 프로그램을 실행하여 센서 출력 데이터를 시각적으로 조회 할 수 있습니다.

Arduino 코드는 플래시 (Arduino 용어로 '업로드')에 저장되므로 새 보드에는 한 번만 수행하면됩니다.

그림 8. Arduino and SingleTact 조합



참조: USB 통신은 Arduino 소프트웨어 패키지에서 추가 드라이버 설치가 필요할 수 있습니다. 자세한 내용은 https://www.arduino.cc/en/Guide/Windows#toc4를 참조하십시오.

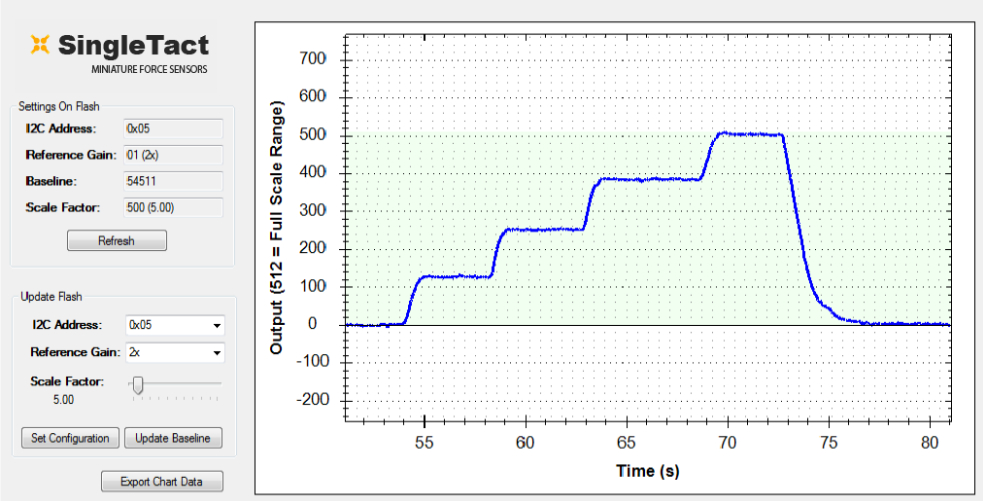
그림 9. SingleTact 과 Arduino UNO 결선

결선	핀 번호	결선
연결 없음	5 4	연결 없음
Arduino UNO A4	6 3	Arduino UNO A5
연결 없음	7 2	연결 없음
Arduino UNO GND	8 1	Arduino UNO 5V

Window GUI 응용프로그램 실행

- SingleTact 프로그램 설치 폴더에서 SingleTact Demo.exe 실행

그림 10. PC DAQ Software DEMO



PC 응용프로그램으로 센서의 I2C 주소를 변경하고 출력 크기 조정을 수정할 수 있습니다. 자세한 내용은 I2C 인터페이스 절을 참조하십시오.

4.2. SingleTact 예제로 Arduino UNO 프로그래밍하기

SingleTact 예제 펌웨어로 Arduino UNO를 프로그래밍하는 방법은 다음과 같습니다.

1. Arduino 소프트웨어를 다운로드하고 설치 : https://www.arduino.cc/en/Main/Software
2. Arduino 펌웨어(ExampleArduinoInterface)를 www.singletact.com에서 다운로드
3. USB 케이블을 사용하여 Arduino를 PC에 연결
4. Arduino IDE 소프트웨어 실행

그림 11. Arduino – PC 연결

USB Interface (Arduino Uno)　　USB Cable

참조 : 추가 드라이버 설치가 필요할 수 있습니다. 자세한 내용은 https://www.arduino.cc/en/Guide/Windows#toc4 를 참조하십시오.

단계별 지침을 따르십시오.

1. 파일 ---> "Open"으로 이동하여 "SingleTactDemo.ino" 파일을 실행합니다.
2. Sketch ---> Include Library ---> .zip Library를 추가하고 "Timer1.zip"을 선택하십시오.
3. 스케치 ---> 검증 / 컴파일
4. Sketch ---> Upload [*1]

* 1 : 업로드시 오류가 발생하면 도구 ---> 포트에서 Arduino가 활성화되어 있는지 확인하십시오.

그림 12. Arduino 내장형 개발 환경

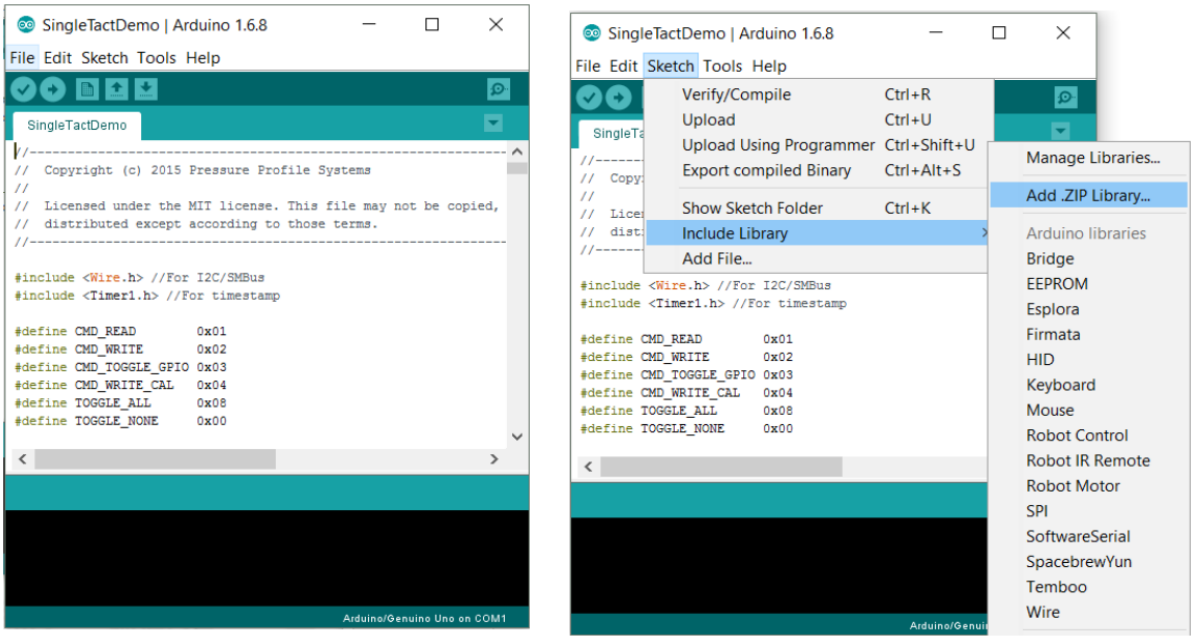
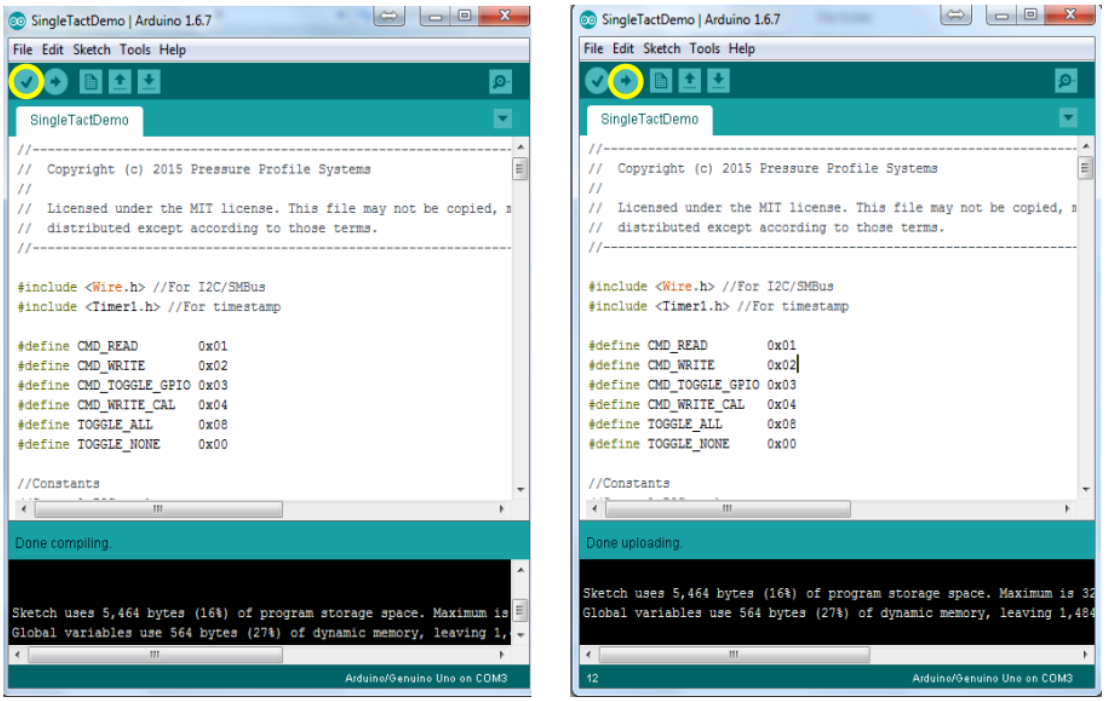


그림 13. SingleTactDemo.ino 파일의 컴파일링과 업로딩



4.3. Arduino 데모 아웃라인

본 섹션의 다이어그램은 이전 섹션에서 설명한 Arduino 데모 기능의 아웃라인입니다. PC와 Arduino 인터페이스가 I2C 인터페이스를 미러링하도록 설정된 경우, 가능한 한 간단하게 Arduino 코드를 유지합니다.

그림 14. Arduino 예제 – 커뮤니케이션 아키텍쳐

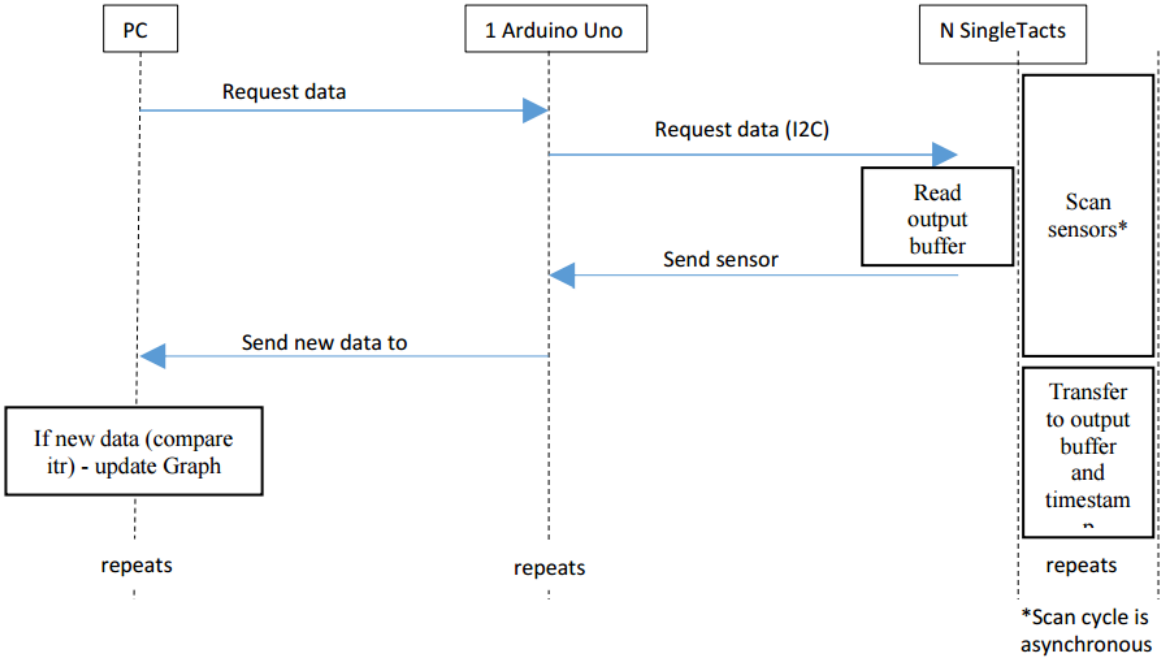
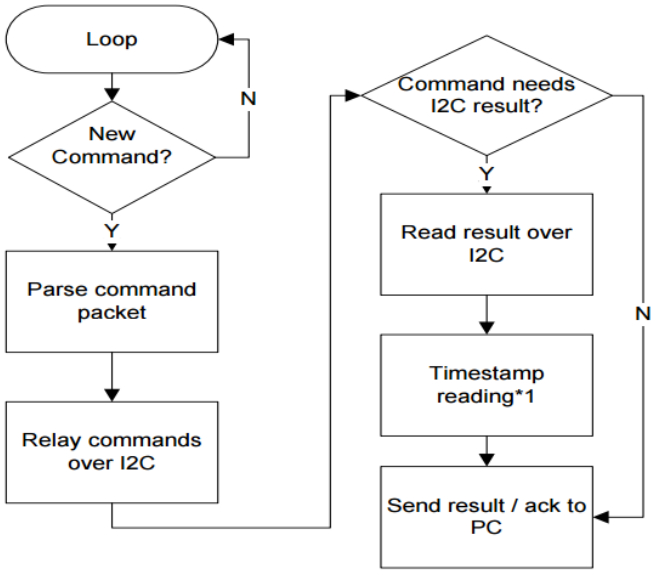


그림 15. Arduino 에플리케이션 순서도



*1: Arduino의 크리스탈 오실리에이터로 SingleTact 인터페이스 보드 보다 더 정확한 타임 스펨프를 생성할 수 있습니다.

호스트에서 Arduino는 가상 RS-232 직렬 장치로 나타납니다. 데이터는 직렬 API를 사용하여 Arduino로 송수신됩니다.

Arduino는 Arduino의 크리스탈 제어 오실레이터를 사용하여 각 패킷의 타임 스탬프를 계산하며, 이는 각 센서의 시간으로 사용될 수 있습니다.

순수 I2C 명령을 미러링하는 시리얼 커멘드(그림 16에서 파란색으로 표시)는 다음 표에 요약되어 있습니다. 헤더(Header) 및 풋터(Footer) 바이트가 추가되어 시리얼 패킷을 쉽게 구분할 수 있습니다. I2C 전송에 타임 아웃을 지정할 수 있습니다.

그림 16. 시리얼 패킷 구조 (Arduino 수신)

바이트	PC -> Arduino
0	Header = 0xFF
1	Header = 0xFF
2	Header = 0xFF
3	Header = 0xFF
4	**센서의 I2C 주소**
5	타임아웃 (100ms 씩 증가)
6	ID (응답시)
7	**읽기 (0x01) 또는 쓰기 (0x02)**
8	**읽기 / 쓰기 위치**

9	**읽기 / 쓰기 -> N 바이트 (최대 32)**
10 -> (10 + N-1)	**쓰기 데이터** **읽기 요청인 경우, 0 바이트**
11 + N	**0xFF – 패킷의 끝을 의미**
12 + N	Footer = 0xFF
13 + N	Footer = 0xFF
14 + N	Footer = 0xFF
15 + N	Footer = 0xFF

그림 17. 시리얼 패킷 구조 (Arduino 전송)

바이트	Arduino -> PC
0	Header = 0xFF
1	Header = 0xFF
2	Header = 0xFF
3	Header = 0xFF
5	타임아웃 초과시, 1
6	ID (응답시)
7	타임스템프 MSB
8	타임스템프
9	타임스템프
10	타임스템프 LSB
11	전송될 N I2C 바이트(최대 32)
12 -> 12 + N	I2C 데이터
13 + N	Footer = 0xFF
14 + N	Footer = 0xFF
15 + N	Footer = 0xFF
16 + N	Footer = 0xFF

4.4. .NET API 예제

본 섹션에서는 PC GUI 응용 프로그램을 작성시 사용되는 .NET API에 대해 자세히 설명합니다. NET 인터페이스 및 데모 응용 프로그램은 www.singletact.com에서 다운로드할 수 있습니다. 편의상 낮은 레벨의 PC 인터페이스는 다음 2가지 .NET 구성 요소로 캡슐화했습니다.

1. ArduinoSingleTactDriver - 기본 Arduino 인터페이스입니다.
2. SingleTact – 고유 I2C 주소를 가진 여러개의 SingleTact가 있을 수 있습니다.

SingleTact 센서 인터페이스:

```
arduinoSingleTactDriver.Initialise(COMport);                    //Arduino 드라이버 시작
singleTact_.I2cAddressForCommunications = 0x04;                 //I2C 주소 설정
singleTact_.Initialise(arduinoSingleTactDriver);                //센서 작동
```

센서 데이터 읽기:

```
SingleTactFrame newFrame = singleTact_.ReadSensorData();        //센서 데이터 가져오기
        if (null != newFrame)                                   //데이터가 있는 경우
        {//Process result}
```

센서 설정을 불러오기:

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

센서 설정값 조정:

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

센서 설정 저장:

```
singleTact_.PullSettingsTordware();
```

5. 추가 참조 링크

SingleTact 홈페이지:

http://www.singletact.com/

I2C 버스 기술사양과 사용자 매뉴얼 (ver 6, Apr 2014):

http://www.nxp.com/documents/user_manual/UM10204.pdf

Arduino 홈페이지:

https://www.arduino.cc/

Microsoft .NET Framework:

https://www.microsoft.com/net

6. 약어설명

API	Application Program Interface
CDC	Capacitance to Digital Converter
DAQ	Data Acquisition
FSR	Full Scale Range
I2C	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

본 매뉴얼은 PPS SingleTact User manual Rev 2.2을 참조하여 제작되었습니다. 자세한 개정 내역은 영문판 매뉴얼을 참조해 주십시오.

1. Rev. 2.2_K_1.0
   2017년 3월 21일: 영문 메뉴얼(Rev 2.2) 기준 번역.