1. Introduction

O

The I²C bus consists of two bidirectional signals (clock and data) shared by a Master and one or more Slaves, where each Slave has a bus-unique address. In this application, we display force measurements from multiple SingleTact sensors connected as Slaves on an I²C bus controlled by an Arduino.



Figure 1: I²C connection between Arduino(master) and SingleTacts(slaves).

You can read more about I²C in the <u>specification document</u> or on <u>Wikipedia</u>.

2. Getting Started

Before you start, you will need:

- 1. Two or more SingleTact sensors
- 2. Two or more SingleTact electronics modules
- 3. Twelve jumper wires
- 4. An Arduino UNO
- 5. Arduino Software (IDE). If your PC doesn't recognise the Arduino board, additional board drivers need to be installed https://www.arduino.cc/en/Guide/ArduinoUno#toc3)
- 6. USB cable
- 7. Computer

3. Configuring the I²C address

Connect the SingleTact sensor with the Arduino UNO (one sensor at a time).

Connect the Arduino to the PC using the USB cable.

Download the PC DAQ binary from: https://github.com/SingleTact/PCExecutable

Open \rightarrow PCExecutable folder \rightarrow run SingleTact Demo.exe.

Choose different I²C Address for each sensor and address should be 0x05 and above.

(Note: addresses 0x00 - 0x03 are reserved and 0x04 is default address).

Click on Set Configuration.

Repeat steps 4-7 for all sensors.

ARDUINO WITH MULTIPLE SINGLETACTS

O



MINIATURE FORCE SENSORS

A. Multi-sensor l²C setup with Arduino UNO Connect the sensors into the connector on the green interface boards, connect Arduino and all interface boards on a common l²C bus using master-multiple slave l²C configuration as shown in the figure.



5. Running the Arduino code

- 1. The Standalone Arduino and Multi SingleTacts code can be downloaded from: https://github.com/SingleTact/StandaloneArduino
- 2. Open the Arduino IDE software.
- 3. Go to File \rightarrow Open \rightarrow open StandaloneArduino folder \rightarrow run SingleTactMultiSensorDemo.ino
- 4. Modify I²C addresses (*inside "void loop(*)*", as shown in the Figure 3*) according to the previous modification has done in Section- 3, step-6.
- 5. Go to Sketch \rightarrow Verify/Compile.
- 6. Go to Sketch \rightarrow Upload
- 7. Click on Tools \rightarrow Serial Monitor.
- 8. Remember to set the baud rate at 57600.

ARDUINO WITH MULTIPLE SINGLETACTS



```
0-
      void loop()
      ſ
          byte i2cAddress; // Slave address (SingleTact), default 0x04
          short data;
              /* Note: No sensor should be addressed with default 0x04 value */
          // Reading data from sensor 1
                                                       Change it according to your
          i2cAddress = 0x06;
                                                              preferences
          data = readDataFromSensor(i2cAddress);
          Serial.print("I2C Sensor 1 Data:");
          Serial.print(data);
          Serial.print("\n");
          delay(100); // Change this if you are getting values too quickly
          // Reading data from sensor 2
          i2cAddress = 0x08;
          data = readDataFromSensor(i2cAddress);
          Serial.print("I2C Sensor 2 Data:");
          Serial.print(data);
          Serial.print("\n");
          delay(100); // Change this if you are getting values too quickly
      }
```

6. Troubleshooting

While running, the SingleTact Demo.exe may receive an error message indicating "invalid setting".

- 1. Check the pin configuration properly.
- 2. Follow the steps mentioned in the User Manual Section 4.2 (Programming the Arduino UNO with SingleTact Example).

Note: For more detailed information please check the User Manual on the SingleTact website https://www.singletact.com/SingleTact_Manual.pdf.

Figure 3: Part of Arduino code for Multi Singletact interface.

ARDUINO WITH MULTIPLE SINGLETACTS

0



Figure 4: Arduino and Multi SingleTact circuit diagram.

SingleTactMultiSensorDemo Arduino 1.8.1	COM3 (Arduino/Genuino Uno) –	>	~
File Edit Sketch Tools Help		Ser	nd
	I2C Sensor 1 Data:461		^
	I2C Sensor 2 Data:265		
SingleTactMultiSensorDemo	I2C Sensor 1 Data:521		
void loop()	I2C Sensor 2 Data:263		
	I2C Sensor 1 Data:533		
bute i2caddress. // Slave address (SingleTart) default 0x04	I2C Sensor 2 Data:263		
short data:	I2C Sensor 1 Data:575		
	I2C Sensor 2 Data:267		
/* Note: No sensor should be addressed with default 0x04 value */	I2C Sensor 1 Data:604		
// Reading data from sensor 1	I2C Sensor 2 Data:262		
i2chdreas = 0x06.	I2C Sensor 1 Data:614		
data = readDataFromSensor(i2cAddress):	I2C Sensor 2 Data:268		
Serial.print("T2C Sensor 1 Data:"):	I2C Sensor 1 Data:619		
Serial.print(data):	I2C Sensor 2 Data:267		
Serial.print("\n"):	I2C Sensor 1 Data:624		
delay(100): // Change this if you are getting values too guickly	I2C Sensor 2 Data:266		
(, , , ,	I2C Sensor 1 Data:646		
// Reading data from sensor 2	I2C Sensor 2 Data:267		
i2clddreas = 0x08:	I2C Sensor 1 Data:663		
data = readDataFromSensor(i2cAddress):	I2C Sensor 2 Data:268		
Serial.print("T2C Sensor 2 Data:"):	I2C Sensor 1 Data:662		
Serial.print(data);	I2C Sensor 2 Data:272		
Serial.print("\n"):	I2C Sensor 1 Data:654		
delay(100); // Change this if you are getting values too guickly	I2C Sensor 2 Data:268		
}	I2C Sensor 1 Data:646		
	I2C Sensor 2 Data:267		
	I2C Sensor 1 Data:650		
short readDataFromSensor(short address)	I2C Sensor 2 Data:266		
{	I2C Sensor 1 Data:648		
byte i2cPacketLength = 6;//i2c packet length. Just need 6 bytes from each slave	I2C Sensor 2 Data:264		
byte outgoing12CBuffer[3];//outgoing array buffer	I2C Sensor 1 Data:645		
byte incomingI2CBuffer[6];//incoming array buffer	I2C Sensor 2 Data:261		
	I2C Sensor 1 Data:643		
outgoing12CBuffer[0] = 0x01;//12c read command	I2C Sensor 2 Data:266		
outgoingI2CBuffer[1] = 128;//Slave data offset	I2C Sensor 1 Data:647		
outgoingI2CBuffer[2] = i2cPacketLength;//require 6 bytes	I2C Sensor 2 Data:269		
	I2C Sensor 1 Data:640		
	I2C Sensor 2 Data:268		
Done uploading.	I2C Sensor 1 Data:630		
	I2C Sensor 2 Data:267		
Sketch uses 4050 bytes (12%) of program storage space. Maximum is 32256 bytes.	I2C Sensor 1 Data:622		
Global variables use 562 bytes (27%) of dynamic memory, leaving 1486 bytes for local variables. Maximum is	I2C Sensor 2 Data:264		
	I2C Sensor 1 Data:639		
<	I2C Sensor 2 Data:268		~
	No line ending V 576	00 baud	~

Figure 5: Example of serial port output.