

# Introduction to Soft Robotics

**Autumn 2022**

**Instructors: Ahmad Rafsanjani, Jonas Jørgensen**

# Part 1: Arduino 101

# Arduino Uno and Serial communication



## Arduino IDE 1.8.19

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

Refer to the [Getting Started](#) page for Installation instructions.

### SOURCE CODE

Active development of the Arduino software is [hosted by GitHub](#). See the instructions for [building the code](#). Latest release source code archives are available [here](#). The archives are PGP-signed so they can be verified using [this](#) gpg key.

### DOWNLOAD OPTIONS

**Windows** Win 7 and newer

**Windows** ZIP file

**Windows app** Win 8.1 or 10 [Get](#)

**Linux** 32 bits

**Linux** 64 bits

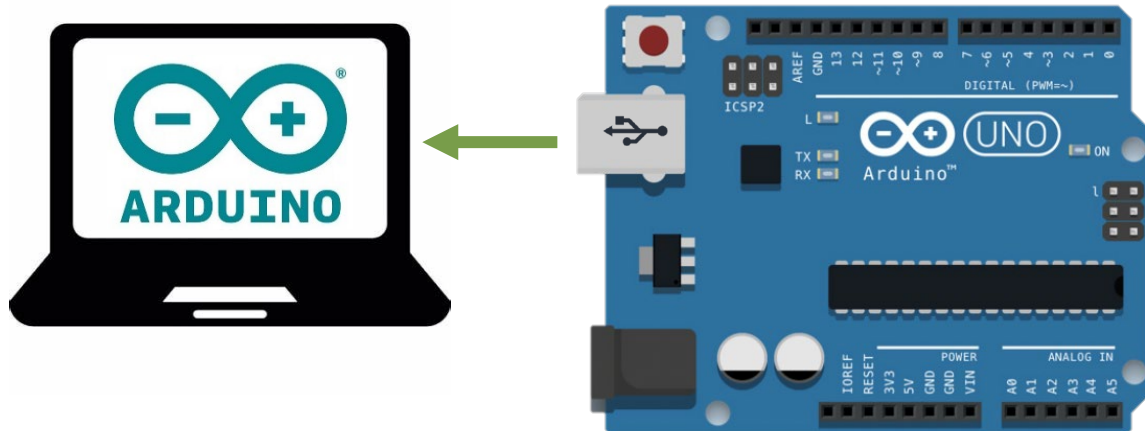
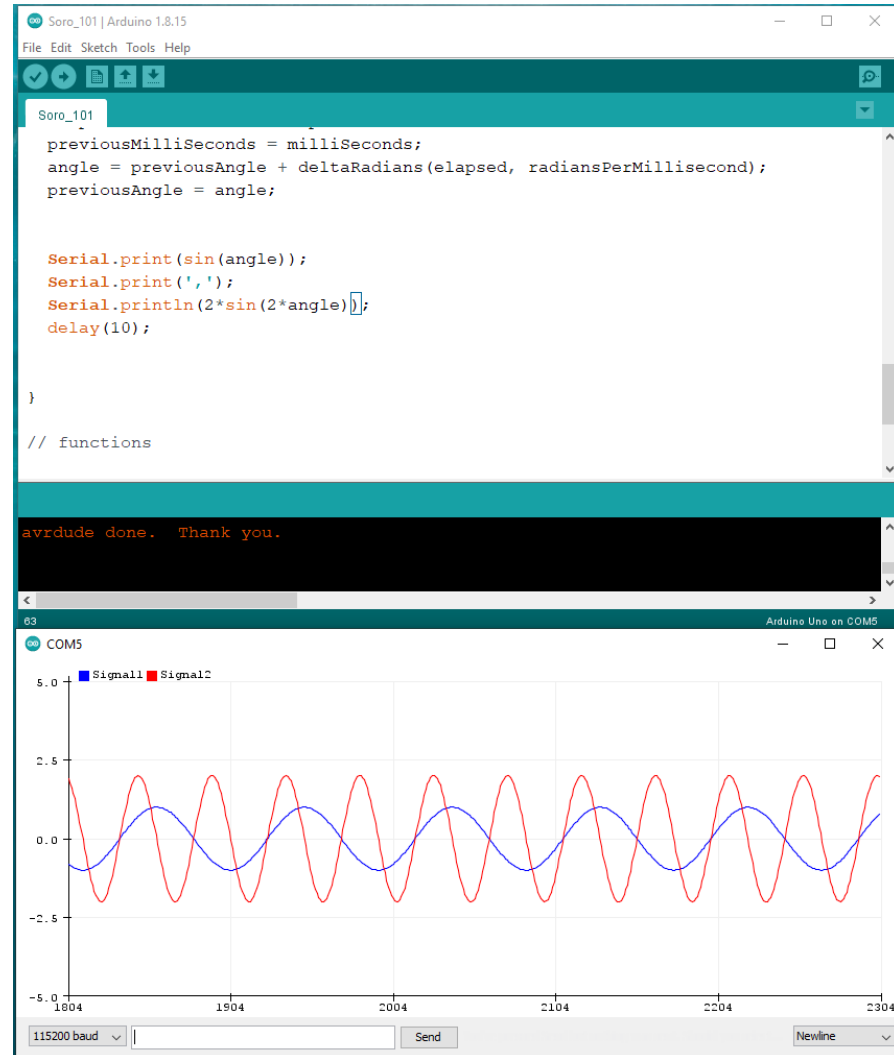
**Linux** ARM 32 bits

**Linux** ARM 64 bits

**Mac OS X** 10.10 or newer

[Release Notes](#)

[Checksums \(sha512\)](#)



SoRo\_101 QR code

[GitHub Link](#)

# Part 2: MPX5100 Integrated Silicon Pressure Sensor

# MPX5100, 0 to 100 kPa, Differential, Gauge, and Absolute, Integrated, Pressure Sensor

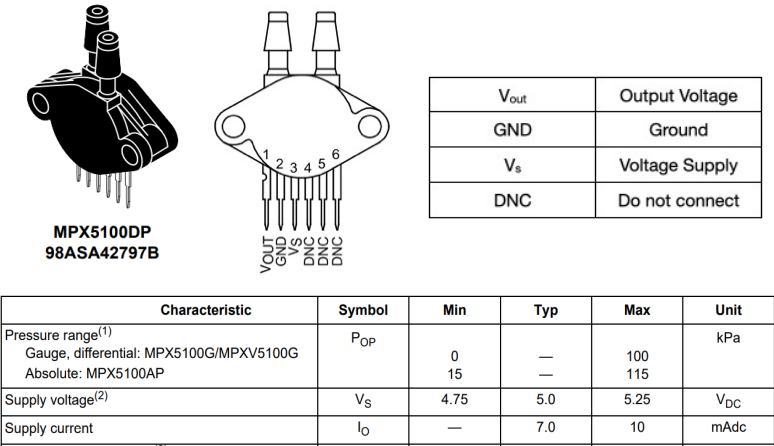


Figure 1. MPX5100DP Pinout (top view), Pin functions, and mechanical and electrical specifications

Nominal Transfer Value:

$$V_{OUT} = V_S (P \times 0.009 + 0.04)$$
$$\pm (\text{Pressure Error} \times \text{Temp. Mult.} \times 0.009 \times V_S)$$
$$V_S = 5.0 \text{ V} \pm 0.25 \text{ V}$$

Figure 8. Transfer function (MPX5100D, MPX5100G, MPXV5100G)

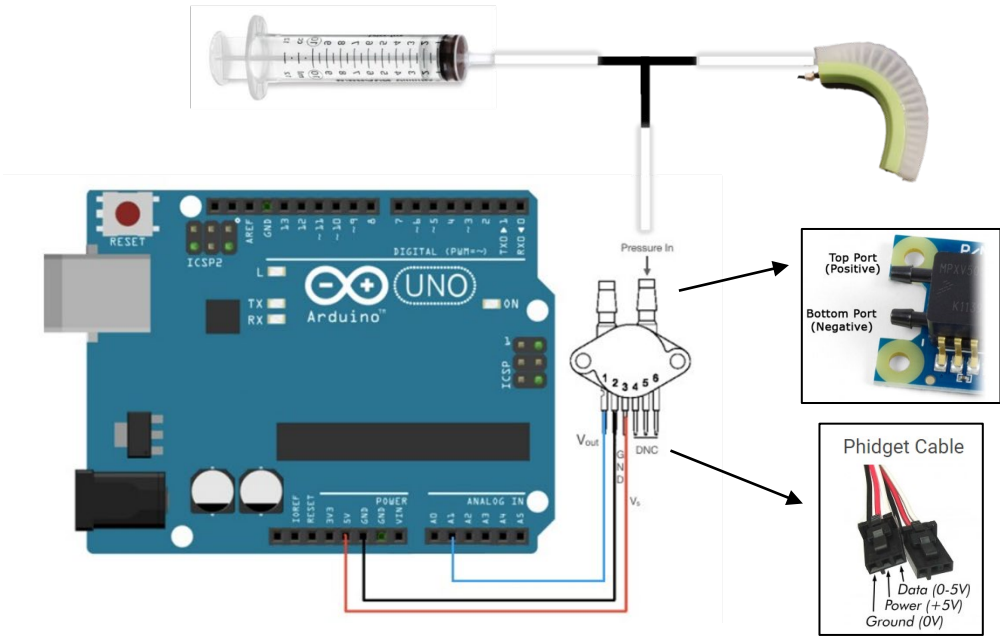


Figure 2. MPX5100DP – Arduino connection diagram

# Code Information

## Arduino libraries

- Download ads Arduino driver at [GitHub Link](#)
- Copy folder **ads\_driver** into **Arduino/Libraries** folder
- Open the **ads.h** file located at **ads\_driver** folder
- Change the command **ADS\_DFU\_CHECK(1)** to **ADS\_DFU\_CHECK(0)**
- Save the changes
- Download **SoRo\_Tutorial\_1** sketch at [GitHub Link](#)

## Arduino Calibration

- Compile and Upload **SoRo\_Tutorial\_1** sketch at Arduino Uno Device
- Open the serial monitor
- Put the Bendlabs sensor in the 0° position
- Type **0** and press **Enter**
- Put the Bendlabs sensor in the 90° position
- Type **9** and press **Enter**



Ads\_driver QR code



SoRo\_Tutorial\_1 QR code

# Code Information

## Arduino Calibration and plotting

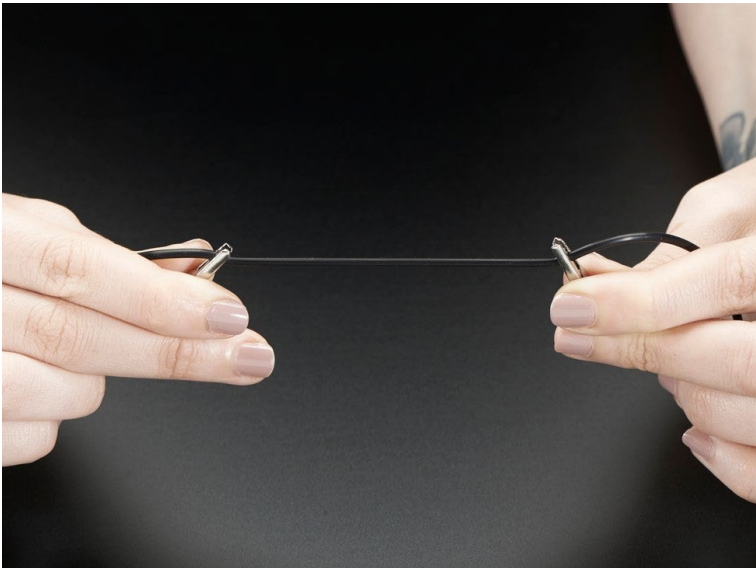
```
//Pressure sensor calibration factors  MPX5100 Series Integrated Silicon Pressure Sensor analog input (0 to 100 kPa)  Vout=Vs(P * 0.009 + 0.04),  Vs=5V = 1024,  P =  
  
const float SensorOffset = 4.44;  //pressure sensor offset  
const float SensorGain = 0.109;  // pressure sensor proportional relation
```

```
// read the input on analog pin 1:  
float pressure_sensorValue = (analogRead(PRESSURE_SENSOR)*SensorGain-SensorOffset); //Do maths for calibration
```

```
Serial.print(sample[0]);    // Angle data  
Serial.print(",");  
Serial.print(pressure_sensorValue);    // pressure data in kpa  
Serial.print(",");  
Serial.println(resistance_sensorValue);  // Stretch data
```

# Part 3: Conductive Rubber Cord Sensor





TECHNICAL DETAILS

Length: approximately 1 meter = 39 inches  
Diameter: 2mm  
Resistance: 350-400 ohms per inch / 140 - 160 ohms per centimeter

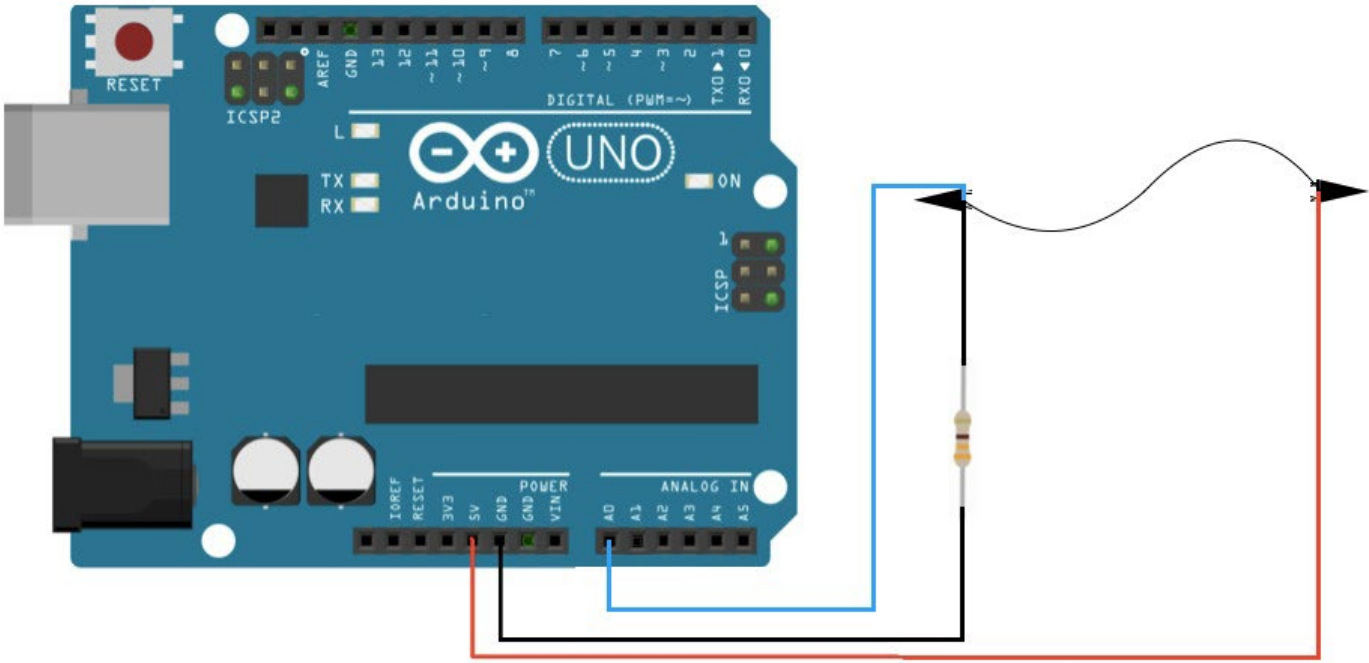


Figure 1. Conductive Rubber Cord Sensor – Arduino connection diagram

# Code Information

## Arduino Calibration and plotting

```
//Resistor sensor calibration factors  
const float SensorOffset2 = 330; //resistance offset  
const float SensorGain2 = 337590; // resistance gain
```

```
// read the input on analog pin 0:  
float resistance_sensorValue = (SensorGain2/analogRead(RUBBER_SENSOR)-SensorOffset2); //Do maths for calibration
```

```
Serial.print(sample[0]); // Angle data  
Serial.print(",");  
Serial.print(pressure_sensorValue); // pressure data in kpa  
Serial.print(",");  
Serial.println(resistance_sensorValue); // Stretch data
```

# Part 4: Bendlabs sensor



# bendlabs

## One Axis

Sensor Specifications	Electrical Specifications
<ul style="list-style-type: none"><li>- Dimensions: 100mm x 7.62mm x 1.27mm (3.94in x 0.30in x 0.05in)</li><li>- Average Sensitivity: 0.274 pF/°</li><li>- Repeatability: 0.18°</li><li>- Life Cycle: &gt;1M cycles</li></ul>	<ul style="list-style-type: none"><li>- Sensitivity: 0.016° LSB</li><li>- Voltage: 1.62 - 3.63V</li><li>- Output: I<sup>2</sup>C</li></ul>

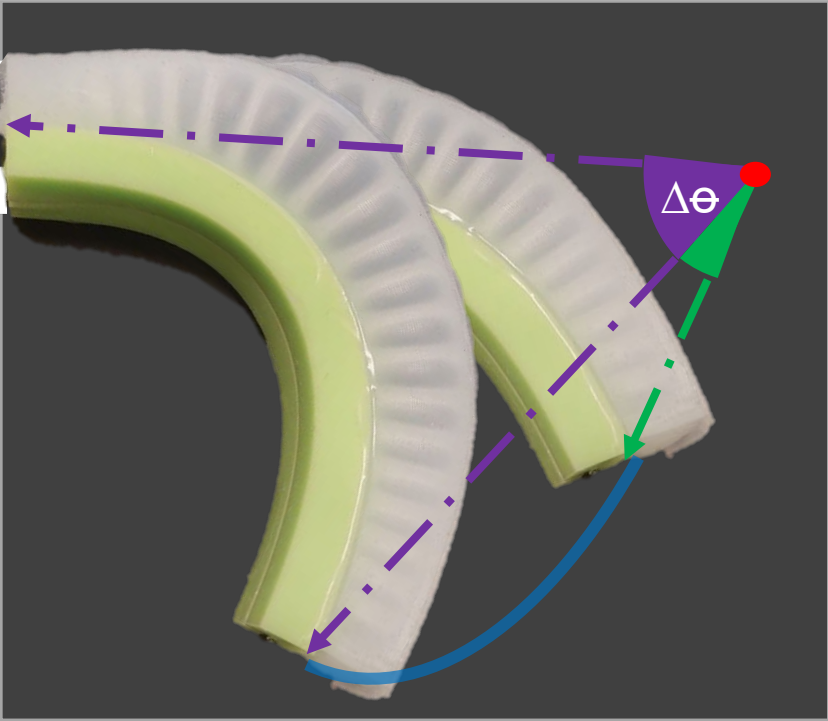
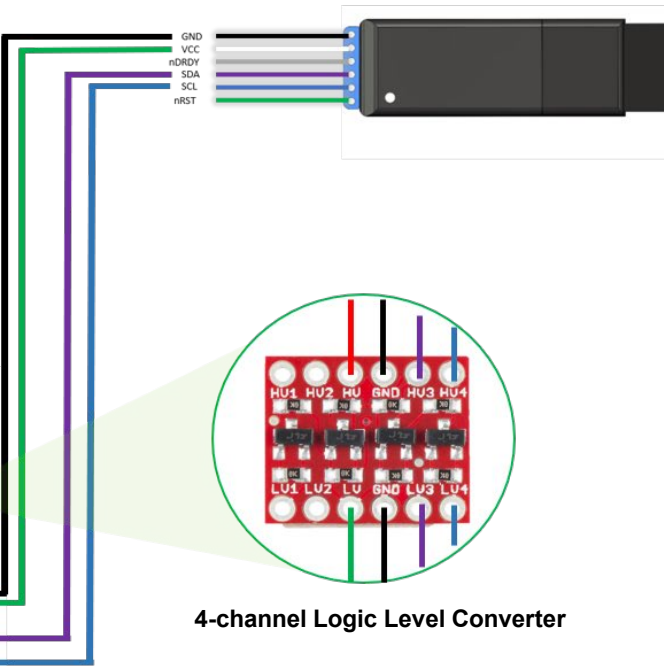
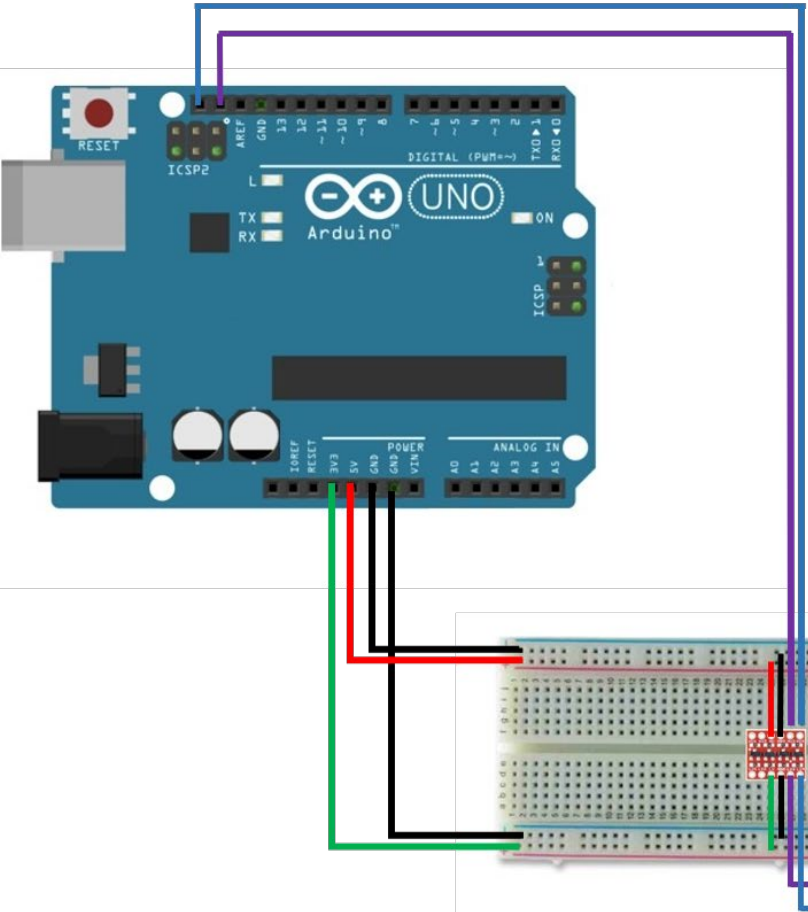


Figure 1. Bendlabs sensor – Logic Level Converter – Arduino connection diagram

# Code Information

## Arduino Calibration and plotting

```
//BendSensor functions for dataprocessing

void ads_data_callback(float * sample);           // Bendlabs sensor information callback
void deadzone_filter(float * sample);             // Bendlabs sensor Deadzone filter
void signal_filter(float * sample);               // Bendlabs signal filter
void parse_com_port(void);                       // I2C communication decoding
```

```
//Defining variables for bendlabs sensor data processing
static float sample[2];
uint8_t data_type;

// Read data from the one axis ads sensor
int ret_val = ads_read_polled(sample, &data_type);

// Check if read was successful

if(data_type == ADS_SAMPLE)
{
    // Low pass IIR filter
    signal_filter(sample);

    // Deadzone filter
    deadzone_filter(sample);
}
```

```
case '0':
    // Take first calibration point at zero degrees
    ads_calibrate(ADS_CALIBRATE_FIRST, 0);
    break;
case '9':
    // Take second calibration point at ninety degrees
    ads_calibrate(ADS_CALIBRATE_SECOND, 90);
    break;
```

# Part 5: System integration

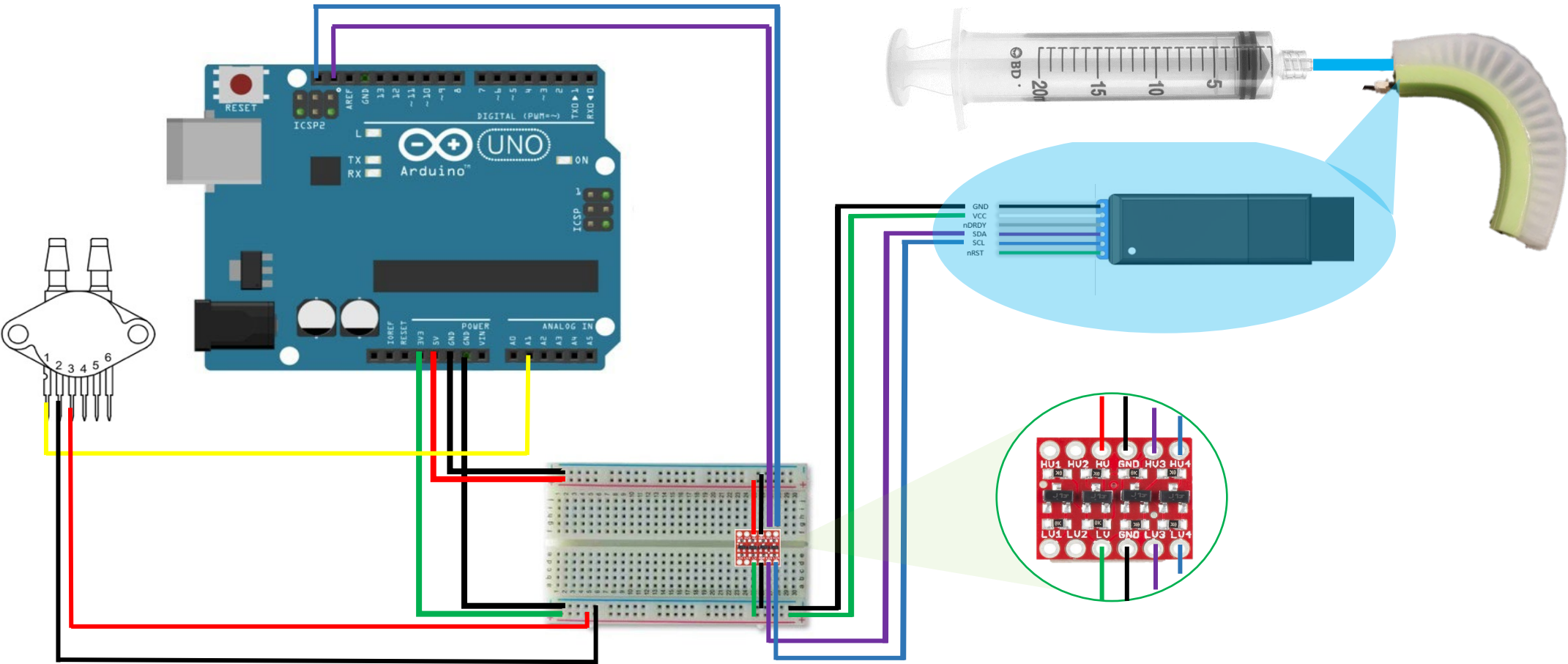


Figure 1. Connection Scheme for a Bendlabs and MPX5100 pressure sensor for Arduino Uno