Introduction to Soft Robotics

Autumn 2021

Instructors: Ahmad Rafsanjani, Jonas Jørgensen, Xiaofeng Xiong

Tutorial 1: Bendlabs sensor

bendlabs

One Axis

Sensor Specifications

- Dimensions: 100mm x 7.62mm x 1.27mm (3.94in x 0.30in x 0.05in)
- Average Sensitivity: 0.274 pF/°
- Repeatability: 0.18°
- Life Cycle: >1M cycles

Electrical Specifications

- Sensitivity: 0.016° LSB
- Voltage: 1.62 - 3.63V
- Output: I²C

4-channel Logic Level Converter

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

Code Information

Arduino libraries

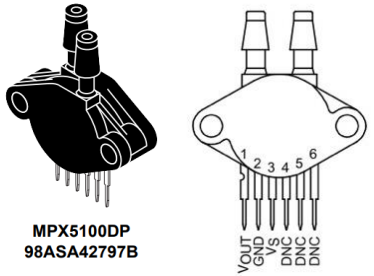
→ Download ads Arduino driver at GitHub Link
→ Copy folder ***ads_driver*** into ***Arduino/Libraries*** folder
→ Open the ***ads.h*** file located at ***ads_driver*** folder
→ Change the command ADS_DFU_CHECK(1) to ADS_DFU_CHECK(0)
→ Save the changes
→ Download ***SoRo_Tutorial_1*** sketch at GitHub 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

Tutorial 2: MPX5100 Integrated Silicon Pressure Sensor

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



MPX5100DP
98ASA42797B

V_{out}	Output Voltage
GND	Ground
V_s	Voltage Supply
DNC	Do not connect

Characteristic	Symbol	Min	Typ	Max	Unit
Pressure range[1] Gauge, differential: MPX5100G/MPXV5100G Absolute: MPX5100AP	P_{OP}	 0 15	 — —	 100 115	kPa
Supply voltage[2]	V_S	4.75	5.0	5.25	V_{DC}
Supply current	I_O	—	7.0	10	mAdc

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

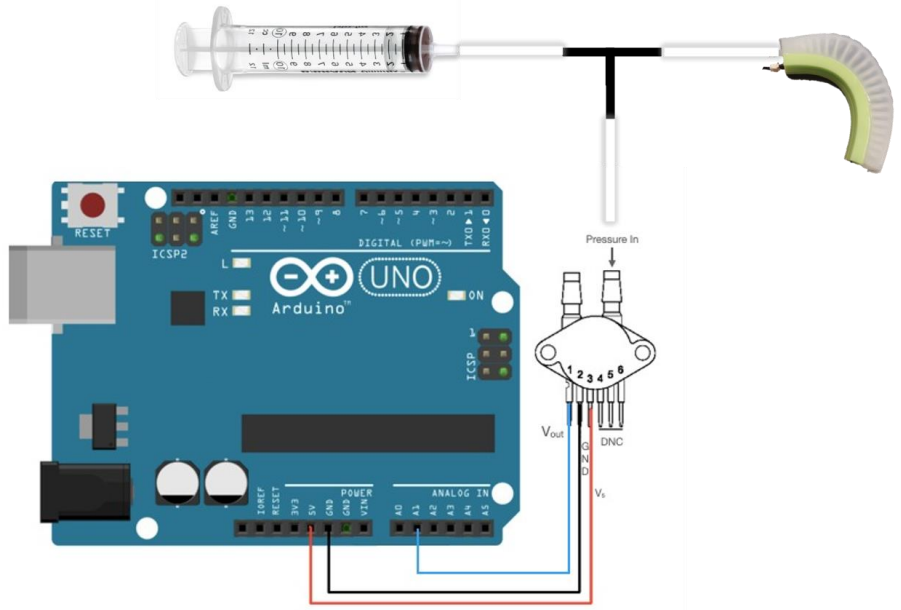


Figure 2. MPX5100DP – Arduino connection diagram

Nominal Transfer Value: $V_{OUT} = V_S (P \times 0.009 + 0.04) \pm (\text{Pressure Error} \times \text{Temp. Mult.} \times 0.009 \times V_S)$

$V_S = 5.0\ V \pm 0.25\ V$

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

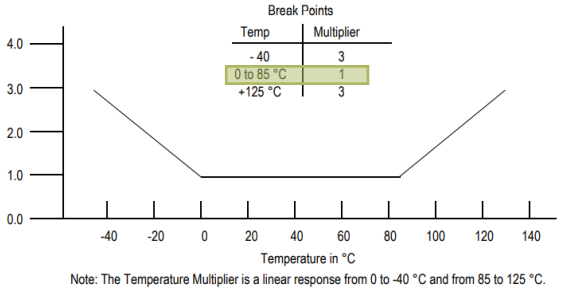


Figure 9. Temperature error multiplier (MPX5100D, MPX5100G, MPXV5100G)

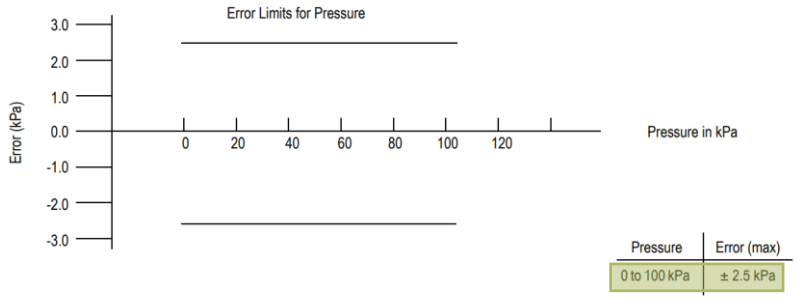


Figure 10. Pressure error band (MPX5100D, MPX5100G, MPXV5100G)

Code Information

Arduino Calibration and plotting

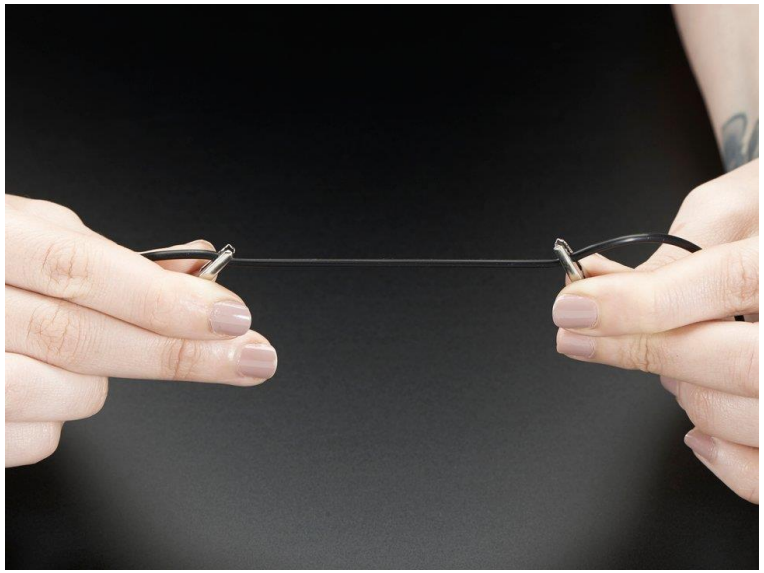
```
//Pressure sensor calibration factors  MPX5100 Series Integrated Silicon Pressure Sensor analog input (0 to 100 kPa)  Vout=Vs(P * 0.009 + 0.04),  Vs=5V = 1024,  P =

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
```

Tutorial 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

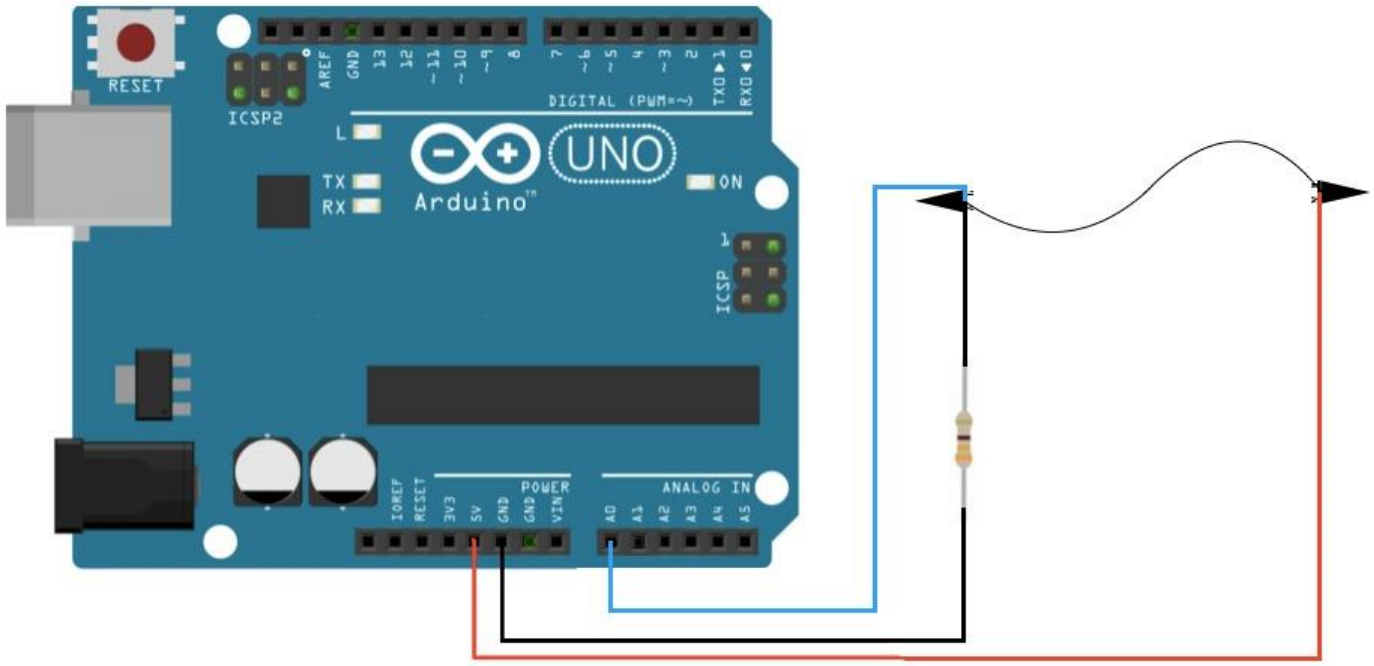


Figure 1. Conductive Rubber Cord Sensor – Arduino connection diagram

Code Information

Arduino Calibration and plotting

```
//Pressure sensor calibration factors

const float SensorOffset2 = 330;  //pressure sensor offset
const float SensorGain2 = 337590;   // pressure sensor proportional relation
```

```
// 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
```

Tutorial 4:
Pumps and valves

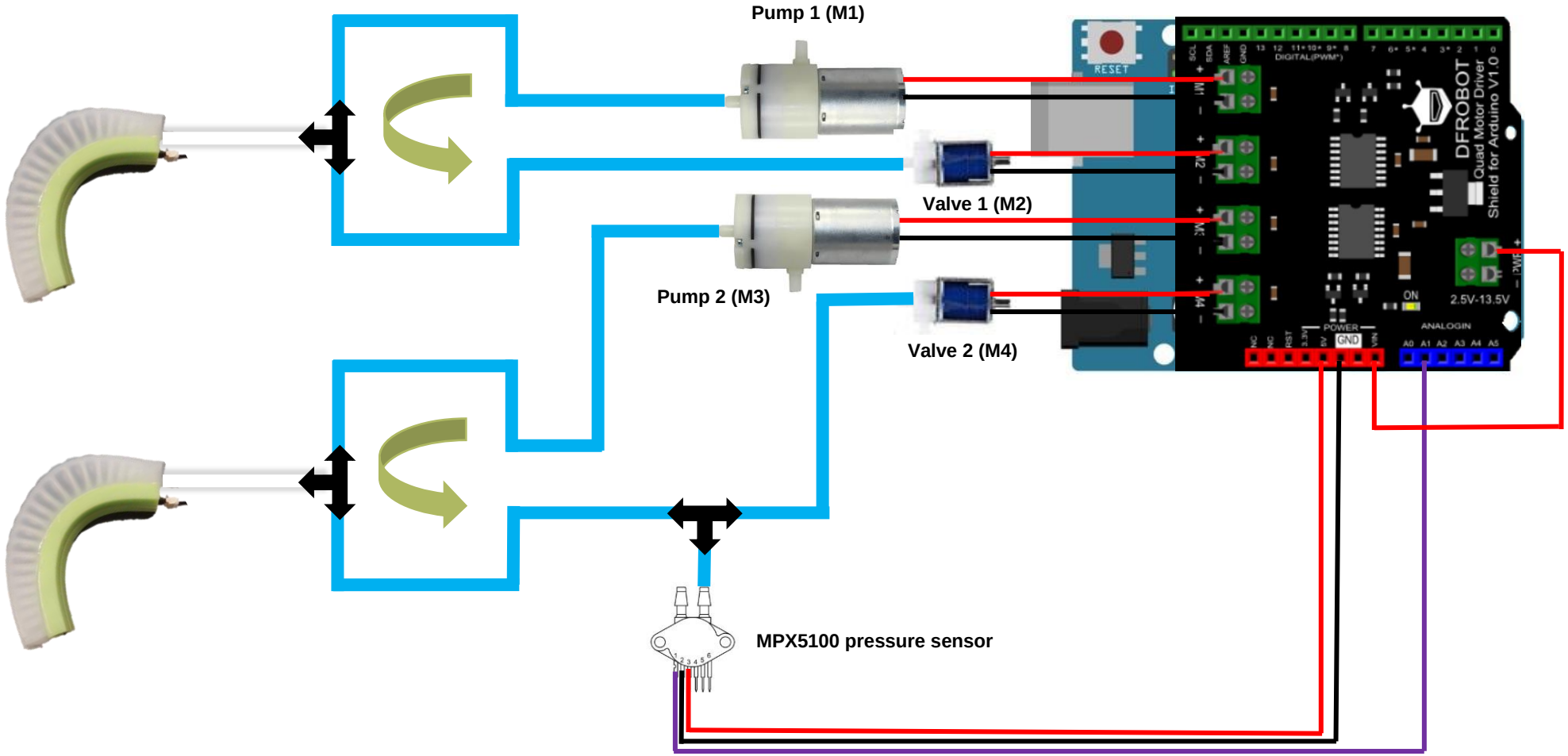


Figure 1. Pneumatic system for two PneuNets and MPX5100 pressure sensor using DFROBOT motor driver for Arduino Uno

Code Information

Arduino Code

→ Download ***Pump_valve_simple_control*** sketch at GitHub Link

Arduino Motor controller functions

```
void motor_1_on(int motorspeed);   // Pump 1 activation motor speed range between 0 - 255   --> 0 - 5 Volts
void motor_2_on(int motorspeed);   // Pump 2 activation motor speed range between 0 - 255   --> 0 - 5 Volts
void motor_1_off(void);            // Pump 1 deactivation  (motor speed == 0)
void motor_2_off(void);            // Pump 2 deactivation  (motor speed == 0)

void valve_1_on(void);             // Valve 1 activated -->  5V
void valve_2_on(void);             // Valve 2 activated -->  5V
void valve_1_off(void);            // Valve 1 deactivated -->  0V
void valve_2_off(void);            // Valve 2 deactivated -->  0V
```

Arduino Time Control:
Sample time = 100 [ms]

```
delay(100);   // defining sample time = 100 miliseconds
```

Code Information

Arduino Time Control:
Sample time = 100 [ms]

```
delay(100);   // defining sample time = 100 miliseconds
```

Example:

```
//Controlling the time of the process
if (timecounter <= 30)      // 3 seconds == 30 * 100 ms
{
  // Introducion air to the PneuNets
     motor_1_on(250);
     valve_1_on();
     timecounter++;
  }
 else if ( timecounter > 30 and timecounter<= 60 )
 {
  motor_1_off();
  valve_1_on();
  timecounter++;
  }
```

```
  else if ( timecounter > 60 and timecounter<= 90)
  {
  motor_1_off();
  valve_1_off();
  timecounter++;
    }
  else if ( timecounter > 90)
  {
    timecounter=0;
    }
```