PNC FORCE SENSOR MATERIAL

RESISTANCE DATASHEET

Version 1.0, February 2025

Manufacturer: NanoSen GmbH

Address: Technologie-Campus 1, 09126 Chemnitz, Germany

Contact: info@nanosen.de | www.nanosen.de

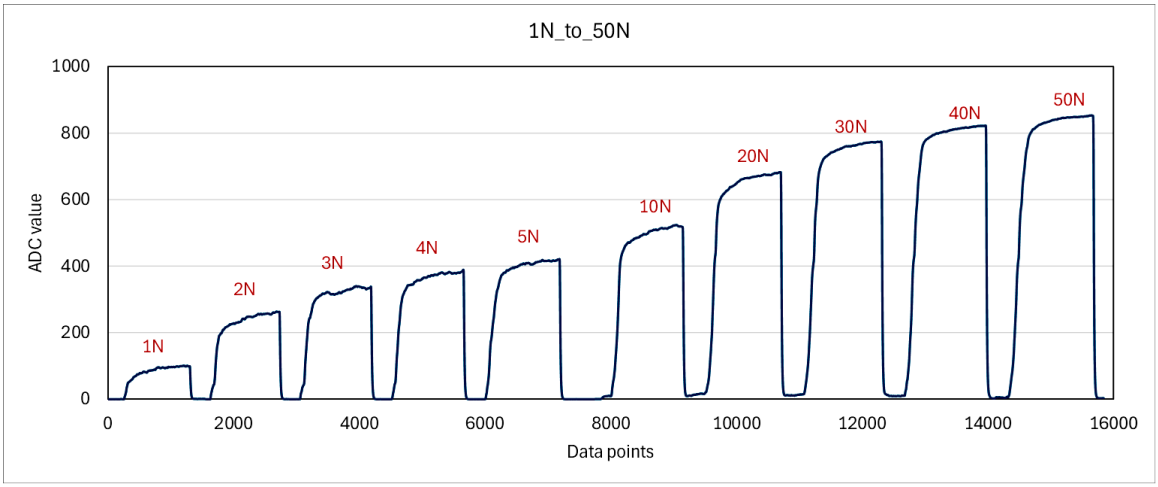
Typical characteristics of a Ø20 mm* PNC force sensor:

Property	Value
Measurement device	Voltage divider with Arduino Nano Every
Reference resistor	10 kΩ
ADC range	0 - 1024
Sensor material	20 mm diameter PNC material
Sensor material version	V1.37
Electrode	Interdigitated screen-printed silver
Actuation force	1 N
Minimum measurable force	5 N
Maximum measurable force	50 N
Drift (Constant force, 5 minutes)	5N: 27%; 10N: 19%; 50N: 9%
Single part repeatability	2%

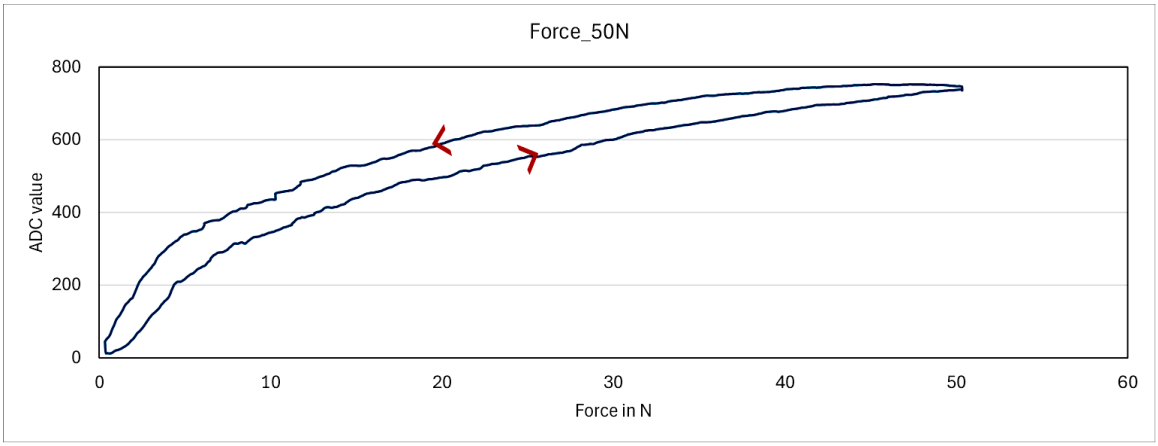
* The values provided correspond to the 20 mm diameter PNC force sensor tested under the described testing setup. Deviations may occur if the sensor is measured differently or integrated into other systems.

Typical performance of a Ø20 mm PNC force sensor:

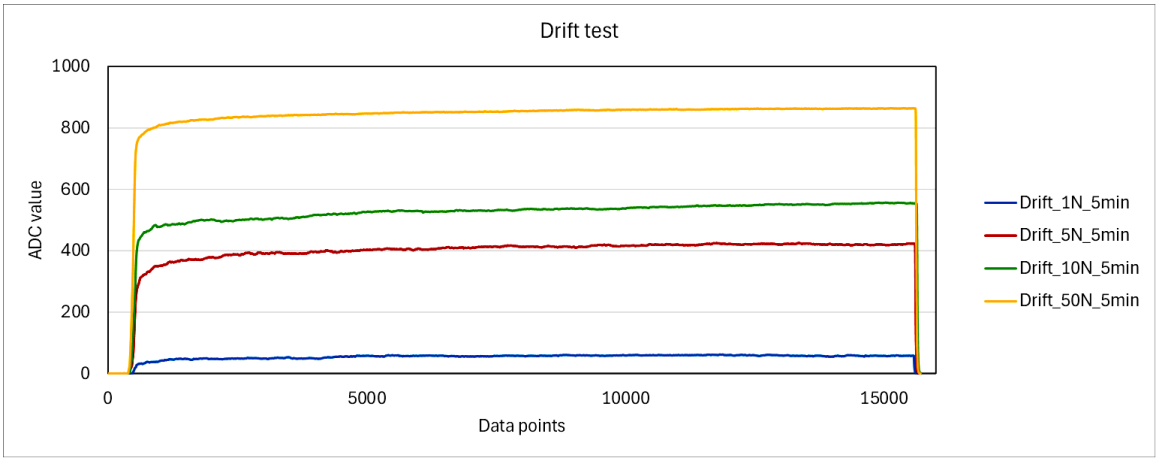
Loading steps from 1N to 50N:



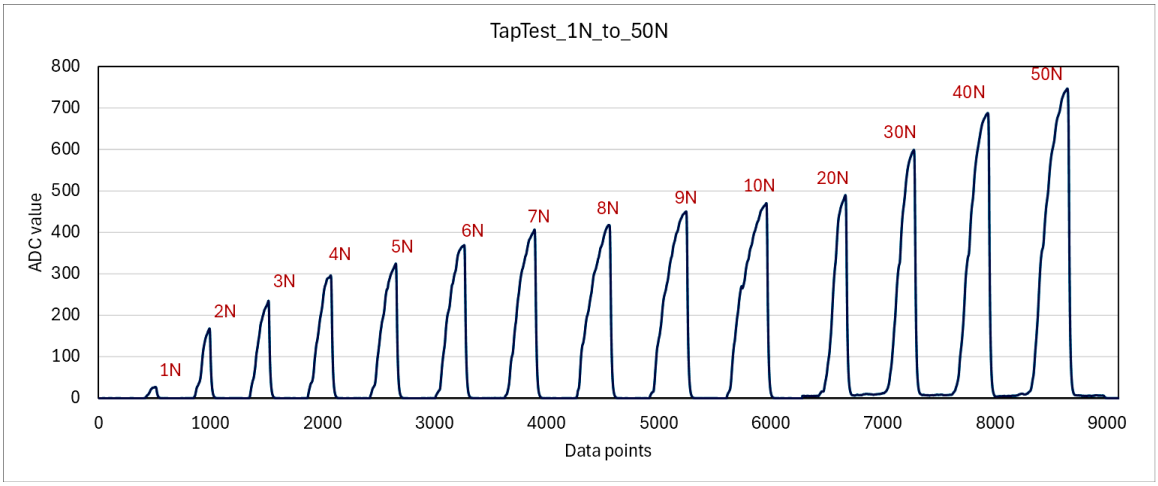
Typical force response curve:



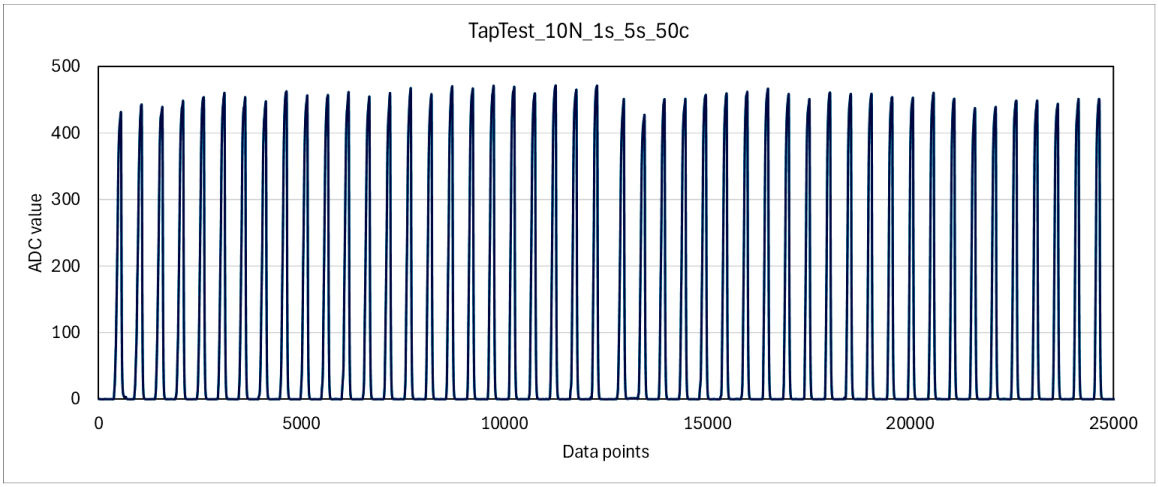
Drift with constant force:



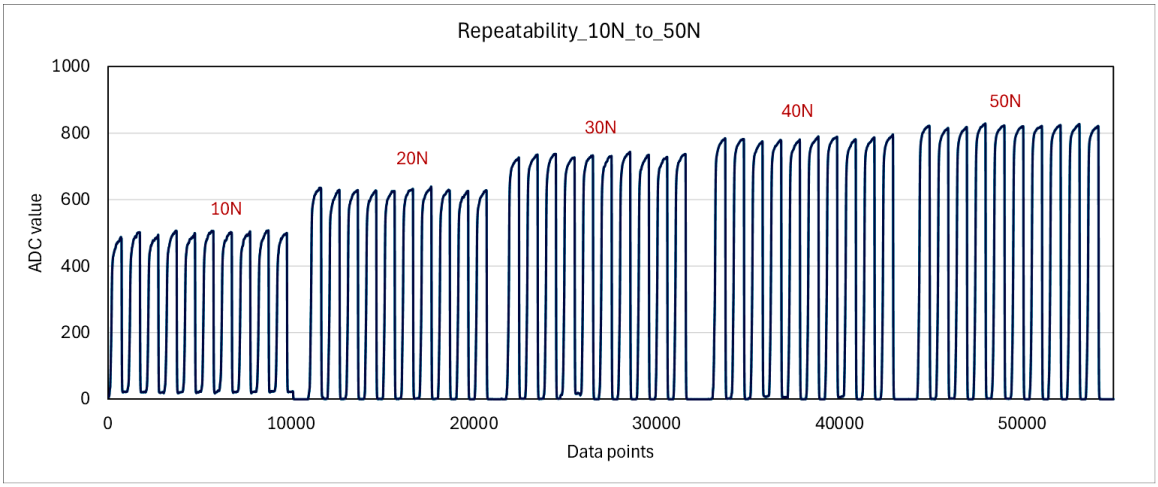
Tap test from 1N to 50N:



Tap repeatability 10N, 50 cycles:



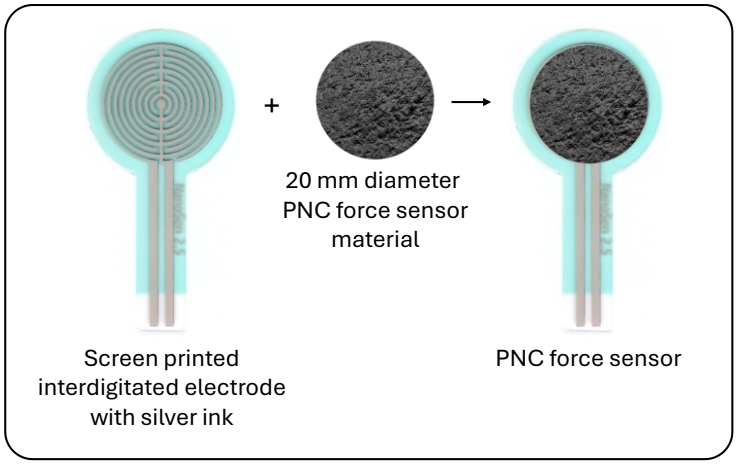
Repeatability test 10N to 50N:

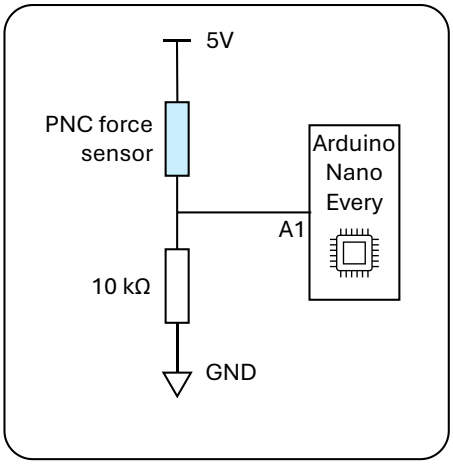


Testing setup:

- The PNC force sensor was evaluated using a Zwick-Roell "Zwicki Z2.5TN" Universal Testing Machine (UTM) equipped with a 1kN load cell.
- To ensure accurate measurements, the sensor was securely mounted in the UTM.
- A 20 mm diameter aluminium force concentrator cylinder was used to distribute force evenly across the sensor's surface. The cylinder was electrically insulated from the sensor to prevent measurement interference.
- The sensor was connected to an Arduino Nano Every in a voltage divider configuration.
- It was set up as a pull-up (+5V) with a 10kΩ reference resistor acting as a pull-down to GND.

Information on the PNC force sensor, schematic and program used:






```ino
const float alpha = 0.1;  // Smoothing factor
const int numSamples = 5; // Number of samples
int readings[numSamples]; // Array to store readings
int readIndex = 0;
long total = 0;
float smaValue = 0;
float emaValue = 0;
unsigned long lastSampleTime = 0;
const int sampleInterval = 20; // 50 SPS -> 1000ms / 50 = 20ms

void setup() {
    Serial.begin(115200);
    for (int i = 0; i < numSamples; i++) { // Initialize buffer
        readings[i] = analogRead(A1);
        total += readings[i];
    }
    smaValue = total / numSamples;
    emaValue = smaValue; // Initialize with first averaged reading
}
void loop() {
    if (millis() - lastSampleTime >= sampleInterval) {
        lastSampleTime = millis(); // Update last sample time
        total -= readings[readIndex]; // Remove oldest reading from total
        readings[readIndex] = analogRead(A1); // Read new value
        total += readings[readIndex];
        readIndex = (readIndex + 1) % numSamples; // Update index for circular buffer
        smaValue = total / numSamples; // Compute average
        emaValue = (alpha * smaValue) + ((1 - alpha) * emaValue); // Apply smoothing
        Serial.println(emaValue);// Print smoothed value
    }
}
```

Document history

Version	Date	Description of changes	Updated by
V1.0	February 2025	Initial Release	NanoSen