# **Pressure control**

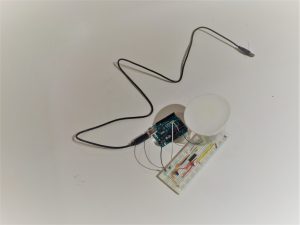
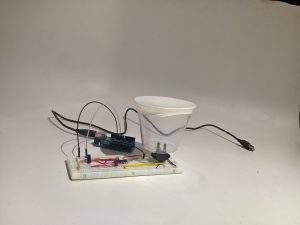
* Posted on [Regina Cantu De Alba](https://itp.nyu.edu/classes/tangible-interaction/author/rcd326/)

* in [sensors](https://itp.nyu.edu/classes/tangible-interaction/category/sensors/)

* on February 26, 2017

* Comments Off on Pressure control

A pressure sensor couldn’t be considered a tangible control within itself, it needs to receive a certain input in order to be able to provide output that can then be harnessed into controlling components.



For this control I will use a project that measures the amount of water in a vessel by means of comparing the vacum and the pressure:

<http://www.practicalarduino.com/projects/water-tank-depth-sensor>

There is a large variety of [pressure sensors](http://www.mouser.com/search/refine.aspx?N=10834212). They vary regarding:

* Its use (The amount of pressure they have to resist)
* The sensibility (a lot of them are used for safety applications so it is fundamental they change any change imediatly)
* The way they are mounted and their size
* The materials they are made of (a lot of them are required to resist corrosion)
* How they are packaged: the amount of nossels they have, if any, how they are mounted, and wether they are dry or wet.

For this control I will be using the MPX2010DP sensor since online documentation shows there is a considerable ease of use with the Arduino microcontroller due to its analog output capability.

“The MPX2010 silicon piezoresistive pressure sensors provide a very accurate and linear voltage output directly proportional to the applied pressure. These sensors house a single monolithic silicon die with the strain gauge and thin film resistor network integrated. The sensor is laser trimmed for precise span, offset calibration and temperature compensation.

Features

* Temperature Compensated over 0°C to +85°C
* Ratiometric to Supply Voltage
* Differential and Gauge Options
* Available in Easy-to-Use Tape & Reel” ()

Other components in the control are:

* LM324 op-amp
* 100nF capacitor
* 1k ohm resistor
* 10k potentiometer
* 1k potentiometer
* 22k ohm resistor
* Arduino Uno

The LM324 op-amp is required to interpret the signal that the change in the piezoresistor generates because of the differential between the scale span and Arduino’s analog read range.

## Link to source for purchase

Both the sensor and the LM324  I got from mouser electronics  , they also provide other pressure sensors in the [MPX2010 line](http://www.mouser.com/ProductDetail/NXP/MPX2010DP/?qs=N2XN0KY4UWUZAyVjnqwQiQ%3D%3D&gclid=CLvDoru0pNICFdlXDQod17oACA), as well as other pressure sensors from different lines.

They have a next day delivery option, but do not guaratee delivery (a problem I  had with them)

## Link to data sheet

**MPX2010**

<http://pdf1.alldatasheet.es/datasheet-pdf/view/86195/MOTOROLA/MPX2010DP.html>

**LM324**

<http://www.ti.com/lit/ds/symlink/lm324.pdf>

<http://pdf.datasheetcatalog.net/datasheet2/9/0oa8seftq8d6peigox0lrx6e9wwy.pdf>

## Description of example uses

The control’s readings need to be perfect in order to understand its possibilities. (ongoing)

## Analysis of strengths or weaknesses

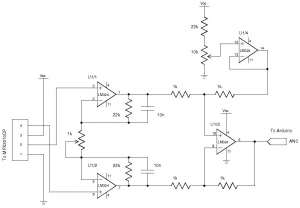
(Of the MPX2010)

Weaknesses

* The sensor I picked is the one with two nozzles because it is a differential pressure sensor[[1]](https://itp.nyu.edu/classes/tangible-interaction/2017/02/26/pressure-control/" \l "_ftn1), this made it really hard to install
* Like any other sensor its readings need to be calibrated to the environment in which it is working, and that proves to be a very tricky task in this sensor due to the circuit that needs to be made with the LM324
* Although most of the web documentation shows it being used to measure the amount of water in a tank, or its depth this is a dry sensor so it must be isolated from the liquid it is measuring, normally through tubing, constraining the fabrication possibilities.

## Example circuit schematic and microcontroller codeto use the control

For the time being these two are the control’s schematic and code.



## Microcontroller code

/\*\*

 \* WaterTankDepthSensor

 \*

 \* Uses an analog input to read the level of water in a tank using a

 \* differential pressure transducer, with the result displayed in serial line

 \*

 \* Copyright 2009 Jonathan Oxer <jon@oxer.com.au>

 \* Copyright 2009 Hugh Blemings <hugh@blemings.org>

 \* http://www.practicalarduino.com/projects/water-tank-depth-sensor

 \*/

int sensorValue      = 0;

int constrainedValue = 0;

int tankLevel        = 0;

int TANK\_SENSOR;

#define TANK\_EMPTY 0

#define TANK\_FULL  1023

void setup() {

  // Enable Serial output and ask WiServer to generate log messages (optional)

  Serial.begin( 9600 );

}

/\*\*

 \* Main program loop. everything is handled by callbacks

 \* in the object itself. Nothing much happens

 \*/

void loop(){

   sensorValue = analogRead(A0);

  constrainedValue = constrain( sensorValue, TANK\_EMPTY, TANK\_FULL );

  tankLevel = map( constrainedValue, TANK\_EMPTY, TANK\_FULL, 0, 100 );

  Serial.print (“Tank Value”);

  Serial.println (sensorValue);

}

# **Any construction drawings you made for laser cutting, CNC, etc.**

(none yet)

# **Citation of all example code or example drawings or documentation that you learned from or used**

(none yet)

[[1]](https://itp.nyu.edu/classes/tangible-interaction/2017/02/26/pressure-control/" \l "_ftnref1) This sensor measures the difference between two pressures, one connected to each side of the sensor. Differential pressure sensors are used to measure many properties, such as pressure drops across [oil filters](https://en.wikipedia.org/wiki/Oil_filter) or [air filters](https://en.wikipedia.org/wiki/Air_filter), fluid levels (by comparing the pressure above and below the liquid) or flow rates (by measuring the change in pressure across a restriction). Technically speaking, most pressure sensors are really differential pressure sensors; for example a gauge pressure sensor is merely a differential pressure sensor in which one side is open to the ambient atmosphere.( https://en.wikipedia.org/wiki/Pressure\_sensor)