Maker Portal

Arduino Pitot Tube Wind Speed and Airspeed Indicator - Theory and Experiments

8 De Fevereiro De 2019 · Joshua Hrisko

Pitot Tube in the foreground and the velocity plot in the background

Named after its french creator, Henri Pitot, the pitot tube is a device used to approximate the speed of vehicles traveling by air and water. An in-depth article on NASA's website is dedicated to pitot tubes (also called pitot-static tubes, Prandtl tubes), where it cites the primary application as airspeed indicator on aircraft. For more information on design and limitations of the instrument, I recommend perusing that page. For this tutorial, only the basic theory is explored using Bernoulli's equation and a practical application. An inexpensive pitot tube and a digital differential pressure sensor

De volta ao topo

Maker Portal

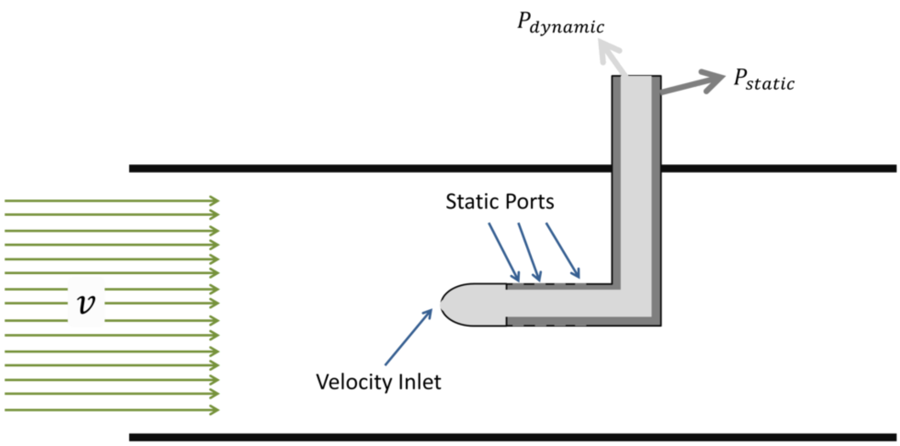
Arduino board.

Pitot Tube and Bernoulli's Equation

For fluid mechanics problems involving incompressible, frictionless flow; Bernoulli's principle is a valid place to start. Bernoulli's principle can be written as a pressure balance between two sections:

$$\frac{1}{2}\rho_a v_a^2 + P_a + \rho_a g h_a = \frac{1}{2}\rho_b v_b^2 + P_b + \rho_b g h_b$$

where ρ, v, P, h are the fluid density, velocity, pressure, and height of the flow section, respectively. The subscripts a,b represent two differing sections of the flow section. In our case, we need the stagnation pressure, which is the pressure due to the static pressure and the pressure incident on the bend in the pitot tube. A setup of this can be seen below:



Assuming no change due to gravity and no velocity at the stagnation point, we can see how the pitot tube can measure both velocity and static pressure:

Additionally, we have a separate section that allows only static pressure to be measured via ports perpendicular to the flow direction (static ports). We can now measure the stagnation pressure and the static pressure separately:

$$P_{stagnation} = P_{static} + \frac{1}{2}\rho v^2$$
$$P_{static} = P_{static}$$

subtracting the two:

$$P_{stagnation} - P_{static} = P_{static} + \frac{1}{2}\rho v^2 - P_{static}$$

and if we solve for velocity as a function of the pressure differential:

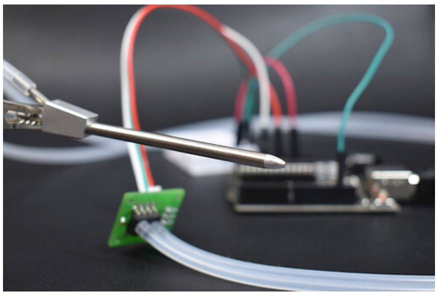
$$v = \sqrt{\frac{2(P_{stagnation} - P_{static})}{\rho}}$$

and this is the classic result, which gives us velocity as a function of the fluid density and two pressure measurements using the static ported pressure and the pitot tube's stagnation pressure! From here, I will introduce methods for finding the pressure differential from an inexpensive pitot tube and pressure sensor using Arduino.

Parts List and Wiring

Este experimento consiste em uma combinação de tubo pitot e sensor diferencial de pressão e uma placa Arduino. O método de teste do tubo pitot fica a critério do usuário - por exemplo, usarei um pequeno ventilador CC e um tubo de 0,5 polegada para medir a ou velocidade do carro. Abaixo está uma lista de peças necessárias e específicas para este experimento:

1. Tubo Pitot e MLXV7002DP - US$ 32,99 [Amazon] ou US$ 48,89 [Amazon]
   - NOTA: estes parecem estar indisponíveis no momento, mas os usuários podem comprar um sensor de pressão diferencial semelhante e um tubo de pitot em nossa loja: Sensor de velocidade do tubo de Pitot para Arduino e Raspberry Pi - $ 45,00 [Nossa loja]
2. Arduino Uno - $ 13,00 [Nossa Loja]
3. Ventilador 5V DC - $ 7,98 (4 unidades) [Amazon]
4. Tubulação de 0,5 polegadas - $ 12,39 (10 pés) [Amazon]
5. Fios de jumper (macho para macho) - $ 0,45 (3pcs) [Nossa loja]



Sensor de velocidade do tubo de Pitot para Arduino e Raspberry Pi

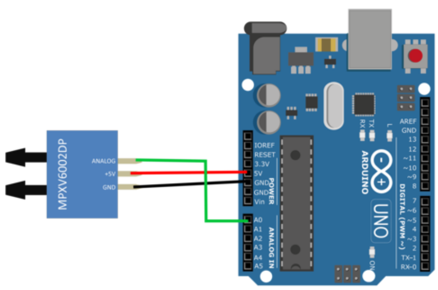
$ 45,00

Maker Portal

Tábua de pão

A fiação do Arduino e do sensor de pressão é simples: conectamos a saída analógica do sensor MPXV7002DP a uma das entradas analógicas da placa Arduino.

MPXV7002DP Fiação para Medição de Pressão Diferencial

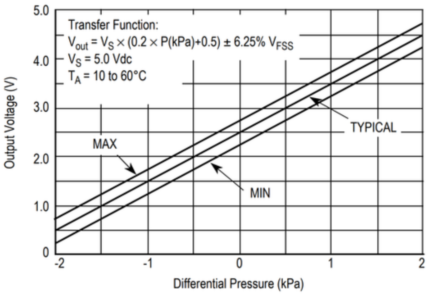


Como o MPXV7002DP lida com o cálculo do diferencial de pressão e o tubo de pitot lida com a medição da pressão de estagnação e pressão estática - muito do trabalho a ser feito é no cálculo da velocidade, não na fiação elétrica. Portanto, a parte eletrônica deste tutorial é bastante simples. Nas próximas seções, a conversão da leitura de tensão para o diferencial de pressão e, subsequentemente, para a velocidade é abordada.

MakerPortal

Folha de Dados MPXV7002DP Conversão de Tensão para Pressão

O MPXV7002DP é um sensor que relaciona o diferencial de pressão entre suas duas entradas. Na folha de dados , o fabricante fornece a relação de calibração entre a tensão de saída e a diferença de pressão. Este gráfico é mostrado abaixo:



Relação tensão x pressão para representar o diferencial de pressão em função da tensão. Este gráfico será essencial para relacionar a tensão lida pelo pino analógico do Arduino de volta ao diferencial de pressão entre a pressão de estagnação e a pressão estática.

Algumas notas sobre o enredo acima:

1. Os limites de pressão não mudam apesar da tensão de alimentação - o que significa que podemos usar 3,3 V ou 5,0 V
2. A pressão máxima e mínima colocam os limites de velocidade em torno de 65 m/s
3. A alimentação bruta do Arduino 5.0V é bastante barulhenta, o que resulta em alto erro

Como a função de transferência não está correta, usarei um método de inclinação de ponto para derivar novamente a função de transferência e, em seguida, plotar novamente abaixo. Para começar, usamos a equação de inclinação linear:

$$y = mx + b$$

where m, b are the slope and y-intercept of the line, respectively. And y is the voltage reading, and x is the pressure difference between the

Maker Portal

$$\frac{V_s}{2} = b$$

$$V_s = m(2.5) + \frac{V_s}{2}$$

$$m = \frac{V_s}{5}$$

which can be written in equation form as:

$$y = \frac{V_s}{5}x + \frac{V_s}{2}$$

This is the relation for converting pressure to voltage, and if we solve for x , we can get the equation for going from voltage to pressure:

$$x = \frac{5y}{V_s} - \frac{5}{2}$$

MPXV7002DP Conversion Table

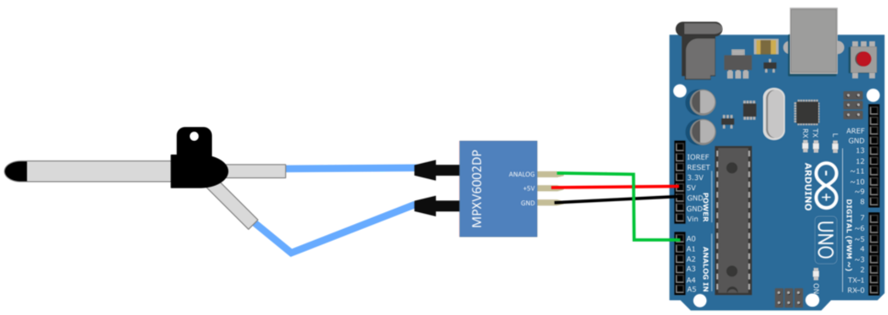
Pressure [kPa] to Voltage [Volts]	Voltage [Volts] to Pressure [Pa]
$V_r = \frac{V_s}{5}\Delta P + \frac{V_s}{2}$	$\Delta P = 1000 \cdot \left(\frac{5 \cdot V_r}{V_s} - \frac{5}{2}\right)$

The table above is a summary ofthe conversion between pressure and voltage and vice-versa. Above, V_r is the voltage reading from the either kPa or Pa depending on the conversion direction. When going from pressure to voltage, we want to use kPa to match the plot units. When going from voltage to pressure, we convert to Pa so that we can match the units of the pitot tube equation.

NOTE: WHEN CONVERTING FROM VOLTAGE TO PRESSURE, BE SURE TO MULTIPLE BY 1000 TO GO FROM THE UNITS OF THE PRESSURE DIFFERENTIAL (KILOPASCAL) TO S.I. UNITS (PASCALS).

Pitot Tube and MPXV7002DP Velocity Approximation

At this stage, the user must connect their pitot tube to the pressure differential sensor. Be sure that the tubes are not bend or pinched in any area, otherwise results may be misleading due to friction or turbulence. Below is a drawing of the pitot tube connected to the MPXV7002DP, which is subsequently wired to the Arduino board:



Wiring the pitot tube to the MPXV7002DP sensor, which is wired to the Arduino uno board.

Maker Portal

voltage by the MPXV7002DP sensor. We can do this by citing the velocity equation:

$$v = \sqrt{\frac{2(P_{stagnation} - P_{static})}{\rho}}$$

and the pressure differential from the voltage readout:

$$\Delta P = 1000 \cdot \left(\frac{5 \cdot V_r}{V_s} - \frac{5}{2} \right)$$

and recognizing that $\Delta P = P_{stagnation} - P_{static}$, we can make the following relation between our input parameters, our voltage readout, and the resulting velocity approximated by the pitot tube:

$$v = \sqrt{\frac{2000 \cdot \left(\frac{5 \cdot V_r}{V_s} - \frac{5}{2} \right)}{\rho}}$$

One final note: don't forget the conversion from bits to voltage:

$$V_r = \frac{(\text{ADC Value} \cdot V_s)}{2^{10} - 1}$$

Finally, one unified equation for converting from ADC values to velocity can be written as:

$$v = \sqrt{\frac{10000 \cdot \left(\frac{R}{2^{10}-1} - 0.5 \right)}{\rho}}$$

where R is the reading from the Arduino analog pin, ρ is the density of the fluid (air in this case), and $(2^{10}$-1) is the 10-bit resolution of the

Arduino Code Velocity Implementation

Once the pitot tube and MPXV7002DP are connected and wired to the Arduino board, the user should take some basic analog readings to test the system and its response. You can do this by blowing into the pitot tube or using a fan. The first thing to note is the baseline value. It should be 1023/2 = 511.5, however, this value is not an integer - so we can blindly assume either 511 or 512 are the expected baseline values. Now, in my case, the baseline was a bit higher (534), so I implemented an offset average in the setup portion of the Arduino code. This also means that when starting up the Arduino, there should be no flow such that the code can configure the baseline.

The full Arduino code is shown below, also with an averaging loop which is meant to stabilize the output value (though there is still a lot of noise in the velocity measurements):

```
//Routine for calculating the velocity from
//a pitot tube and MPXV7002DP pressure differential sensor

float V_0 = 5.0; // supply voltage to the pressure sensor
float rho = 1.204; // density of air

// parameters for averaging and offset
int offset = 0;
```

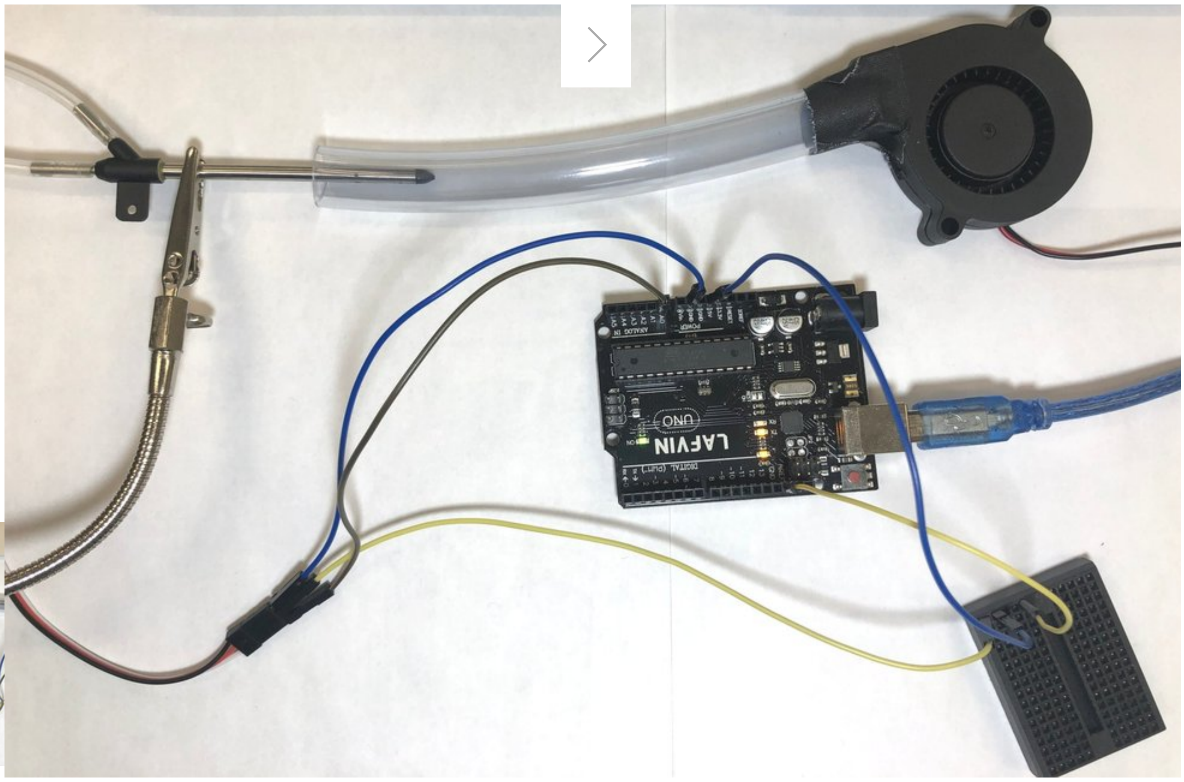
MakerPortal

```
// setup and calculate offset
void setup() {
  Serial.begin(9600);
  for (int ii=0;ii<offset_size;ii++){
    offset += analogRead(A0)-(1023/2);
  }
  offset /= offset_size;
}

void loop() {
  float adc_avg = 0; float veloc = 0.0;

// average a few ADC readings for stability
  for (int ii=0;ii<veloc_mean_size;ii++){
    adc_avg+= analogRead(A0)-offset;
  }
  adc_avg/=veloc_mean_size;

  // make sure if the ADC reads below 512, then we equate it
  if (adc_avg>512-zero_span and adc_avg<512+zero_span){
  } else{
    if (adc_avg<512){
      veloc = -sqrt((-10000.0*((adc_avg/1023.0)-0.5))/rho);
    } else{
      veloc = sqrt((10000.0*((adc_avg/1023.0)-0.5))/rho);
    }
  }
  Serial.println(veloc); // print velocity
  delay(10); // delay for stability
}
```

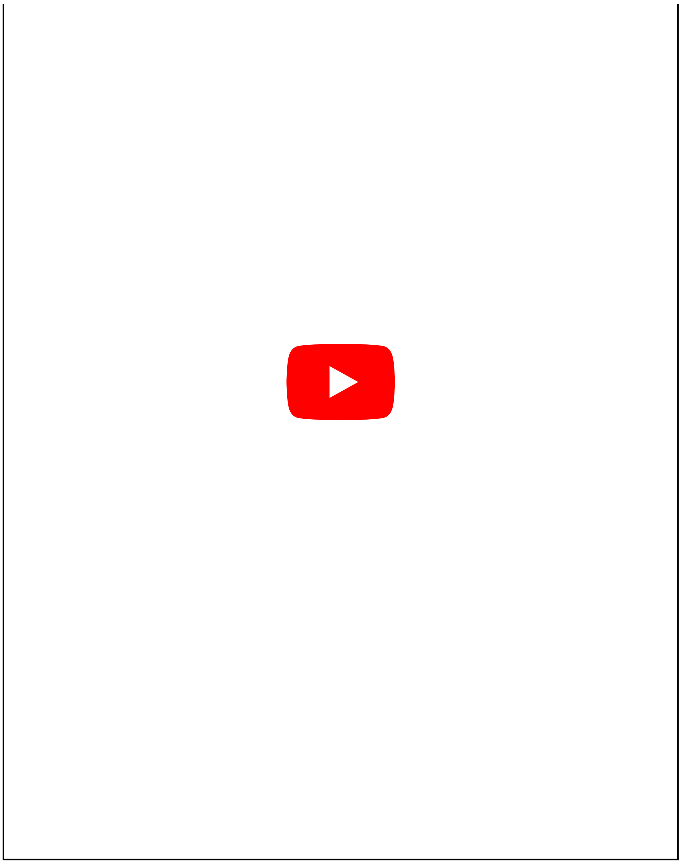
Pitot Tube Experiment with Pipe Flow

As stated earlier, this experiment uses a 0.5 inch cylindrical pipe and a 5V blower fan (which I ran at 12V) to analyze the volumetric flow rate through the pipe using a pitot tube. This will allow us to both test the pitot tube and quantify the efficiency of a cooling fan. Below is a series of photos of the setup:

MakerPortal



Below is a video of the pitot tube measuring velocity in real-time. The pitot tube measures the velocity around 14.5 m/s within the tube.



Now that we know the velocity within the tube, we calculate the volumetric flow rate of the fan and compare it to the manufacturer's value. To start, we can assume that the velocity is uniform due to the shortness of the pipe, which means the velocity profile hasn't developed in the tube (in the presence of friction). So we measured the velocity at the center of the tube and can invoke conservation of energy:

$$Q = Av$$

as stated before, the inner diameter of the acrylic pipe is 0.5 in. (0.0127 m) so we can quantify the volumetric flow rate:

$$Q = \pi \left(\frac{0.0127}{2} \right)^2 \cdot v$$

and previously we found the velocity in the tube to be about 14.5 m/s

And to compare with the manufacturer's value, we convert to CFM (cubic feet per minute):

$$\frac{\mathrm{m}^3}{\mathrm{s}} \times \frac{60\mathrm{s}}{\mathrm{min}} \times \frac{\mathrm{ft}^3}{(0.3048\ \mathrm{m})^3} = 2118.88\ \mathrm{CFM}$$

This leads to our actual volumetric flow rate:

$$Q \approx 3.9\ \mathrm{CFM}$$

and this is quite a satisfying result, because the manufacturer cites the maximum flow rate as 5 CFM, which is not too far off. Not to mention the fact that the areas between the outlet of the fan and the pipe aren't identical, the tube is slightly bent, and our measurements are quite noisy.

Conclusion

Tip of pitot tube used in this experiment

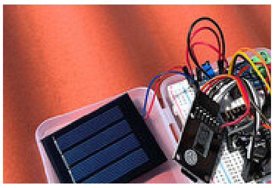
The pitot tube is an instrument widely used to measure the speed of a flowing fluid. I started the tutorial by deriving the pitot tube velocity approximation from Bernoulli's principle. I also introduced the conversion from voltage to pressure differential from the MPXV7002DP sensor, and the subsequent conversion from pressure differential to flow velocity.

efficiency (we measured 15.6 CFM, the datasheet cites 20 CFM). This tutorial was meant to introduce the user to the pitot tube by using a real-world example, while also exploring a classic engineering experiment that utilizes classic fluid mechanics techniques for solving problems.

See More in Engineering:

Haptic Feedback Joystick with Arduino

Aug 9, 2021

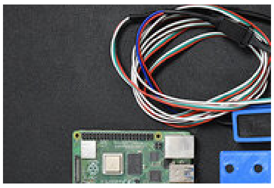


Caracterização de Painéis Solares e Experimentos com Arduino

18 de junho de 2021

Projeto e Fabricação de PCB com NextPCB

27 de março de 2021

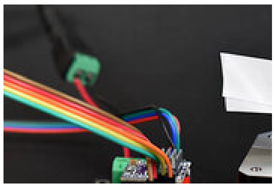


Medição de água com o WaWiCo USB Kit e Raspberry Pi

16 de março de 2021

Ouvindo seus tubos com um microfone MEMS e Raspberry Pi

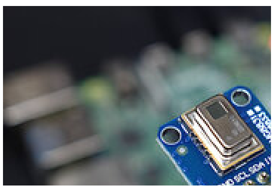
12 de fevereiro de 2021



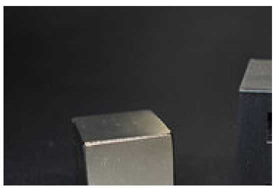
Controle de motor de passo Raspberry Pi com NEMA 17

25 de janeiro de 2021

Detecção de distância com o TF-Luna LiDAR e Raspberry Pi



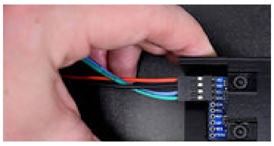
Análise de câmera térmica com Raspberry Pi (AMG8833)



Calibração de um magnetômetro com Raspberry Pi

11 de janeiro de 2021

Maker Portal



Calibração de giroscópio e acelerômetro com Raspberry Pi

3 de janeiro de 2021

COMMENTS (15)

Newest First

Preview Post Comment…

3 years ago · 0 Likes

Hi Joshua, great Job!

3 years ago · 1 Like

Hi Joshua, could you explain this excerpt to me:

```
if (adc_avg> 512-zero_span and adc_avg <512 + zero_span) {
} else {
if (adc_avg <512) {
veloc = -sqrt ((- - 10000.0 * (((adc_avg / 1023.0) -0.5)) / rho);
} else {
veloc = sqrt ((10000.0 * ((adc_avg / 1023.0) -0.5)) / rho);
}
}
```

Maker Portal

Thankful...

Joshua Hrisko 3 years ago · 1 Like

The limits around 512 are just for the ADC on the Arduino board (10-bit ADC - 1024 points. Since the pressure sensor has 512 at its center (0-511 negative velocities, 512-1023 positive velocities), the if statement above just prevents non-physical jitter around the zero velocity (jumping from negative to positive unnecessarily). Thus, we also have the two if statements beyond the first: which just make the velocity negative or positive (without dealing with imaginary numbers).

I hope this helps!

-Josh

Marcos Vinicius 3 years ago · 0 Likes

Hi Joshua, how do I extract the average speed? And why is the program reading negative speeds?

Joshua Hrisko 3 years ago · 0 Likes

Marcos,

The speed is already averaged in the code. Also, if you have negative values with flows into the pitot tube, it's likely that the tubes need to be reversed from the static and stagnation ports to the MPXV7002DP.

elza 3 years ago · 0 Likes

I have tried swapping ports but there is still a negative curve, what is the solution

I'm not sure how that would happen. What are the amplitudes of your signal? Do you have a positive pressure and a lower positive pressure? You need a differential for this sensor to work properly

Thorsten 3 years ago · 0 Likes

Hi Joshua, whatever i try and no matter how many measurements i use to calibrate the offset: i always end up with a non-zero baseline of the measured speed. It's -2 / -3 m/s most time, sometimes +2 or +3 m/s.
Of course, i could set zero_span, but then i lose those samples, which might be relevant for what i would like to measure.

4 years ago · 0 Likes

Hi! thank you for your very detailed explanation about this experiment! Can you explain to me how did you get Vs/2 = b? And how did you get 2.5 at this formula Vs=m(2.5)+Vs/2?

Joshua Hrisko 4 years ago · 0 Likes

This comes from the datasheet of the MPXV7002DP and the idea that the sensor can read negative and positive pressures, all with positive voltages.

David 4 years ago · 0 Likes

Hello ! Thank you for this very proffessional work !
Did you try to measure the maximum/minimum speed of the pitot tube ?

Joshua Hrisko 4 years ago · 0 Likes

David, I never tested the maximum in terms of calibration, but the maximum reading (by blowing into the tube) is just below 5V,

Matija Kranjcec 4 years ago · 1 Like

Nice! I added a 7805 regulator with capacitors, 9V Batteries, a python script for datalogging and used a True Airspee Multiplex 85423 pitot tube instead. I've tested it in a automotive wind tunel and compared it to the readings of a professional hot wire anemometer where the Arduino was only 0.2 m/s off!

3 years ago · 0 Likes

Hi Matija, do you have the wiring diagram for this? Please send to artrepublik.ph@gmail.com

Thanks in advance!

Joshua Hrisko 4 years ago · 0 Likes

Matija, that's fantastic. Good job.

ANTERIOR

Análise e experimentos de alto-falantes: Parte I

20 De Fevereiro De 2019

PRÓXIMO

Teoria, calibração e experimento do termistor do Arduino

15 De Janeiro De 2019