# Distance sensor

**Engineer: Carlos Agorreta**

**GP2Y0A21YK0F**

**Distance Measuring Sensor Unit**
**Measuring distance: 10 to 80 cm**
**Analog output type**


Reference document: gp2y0a21yk_e-3493 (3).pdf

**Description:**

**GP2Y0A21YK0F** is a distance measuring sensor unit, composed of an integrated combination of PSD (position sensitive detector) , IRED (infrared emitting diode) and signal processing circuit. The variety of the reflectivity of the object, the environmental temperature and the operating duration are not influenced easily to the distance detection because of adopting the triangulation method. This device outputs the voltage corresponding to the detection distance. So this sensor can also be used as a proximity sensor.

**Features:**

1. Distance measuring range : 10 to 80 cm
2. Analog output type
3. Package size : 29.5×13×13.5 mm
4. Consumption current : Typ. 30 mA
5. Supply voltage : 4.5 to 5.5 V

## ■Absolute Maximum Ratings

(Ta=25℃,Vcc=5V)

| Parameter | Symbol | Rating | Unit |
|---|---|---|---|
| Supply voltage | $V_{CC}$ | -0.3 to +7 | V |
| Output terminal voltage | $V_O$ | -0.3 to $V_{CC}$+0.3 | V |
| Operating temperature | $T_{opr}$ | -10 to +60 | ℃ |
| Storage temperature | $T_{stg}$ | -40 to +70 | ℃ |

## ■Electro-optical Characteristics

(Ta=25℃,Vcc=5V)

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Average supply current | $I_{CC}$ | L=80cm (Note 1) | — | 30 | 40 | mA |
| Distance measuring | ΔL | (Note 1) | 10 | — | 80 | cm |
| Output voltage | $V_O$ | L=80cm (Note 1) | 0.25 | 0.4 | 0.55 | V |
| Output voltage differential | $\Delta V_O$ | Output voltage difference between L=10cm and L=80cm (Note 1) | 1.65 | 1.9 | 2.15 | V |

* L : Distance to reflective object

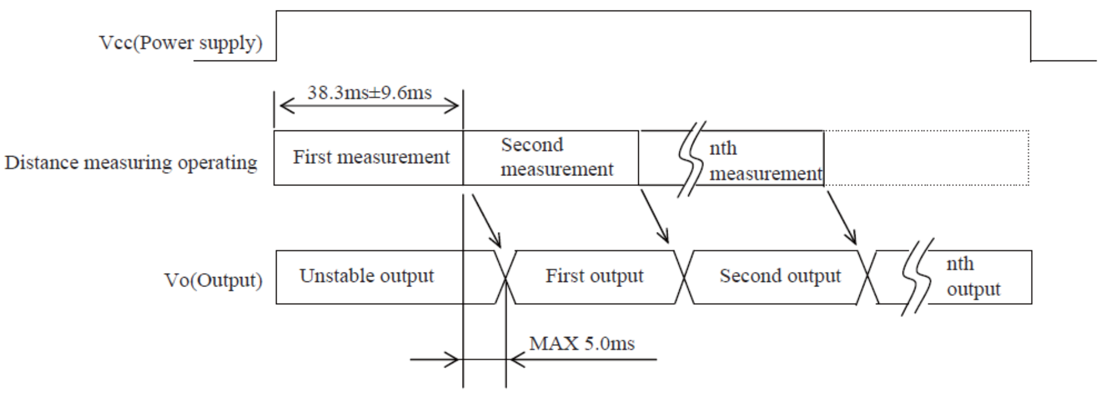
Note 1 : Using reflective object : White paper (Made by Kodak Co., Ltd. gray cards R-27・white face, reflectance; 90%)

## ■Recommended operating conditions

| Parameter | Symbol | Rating | Unit |
|---|---|---|---|
| Supply voltage | $V_{CC}$ | 4.5 to 5.5 | V |

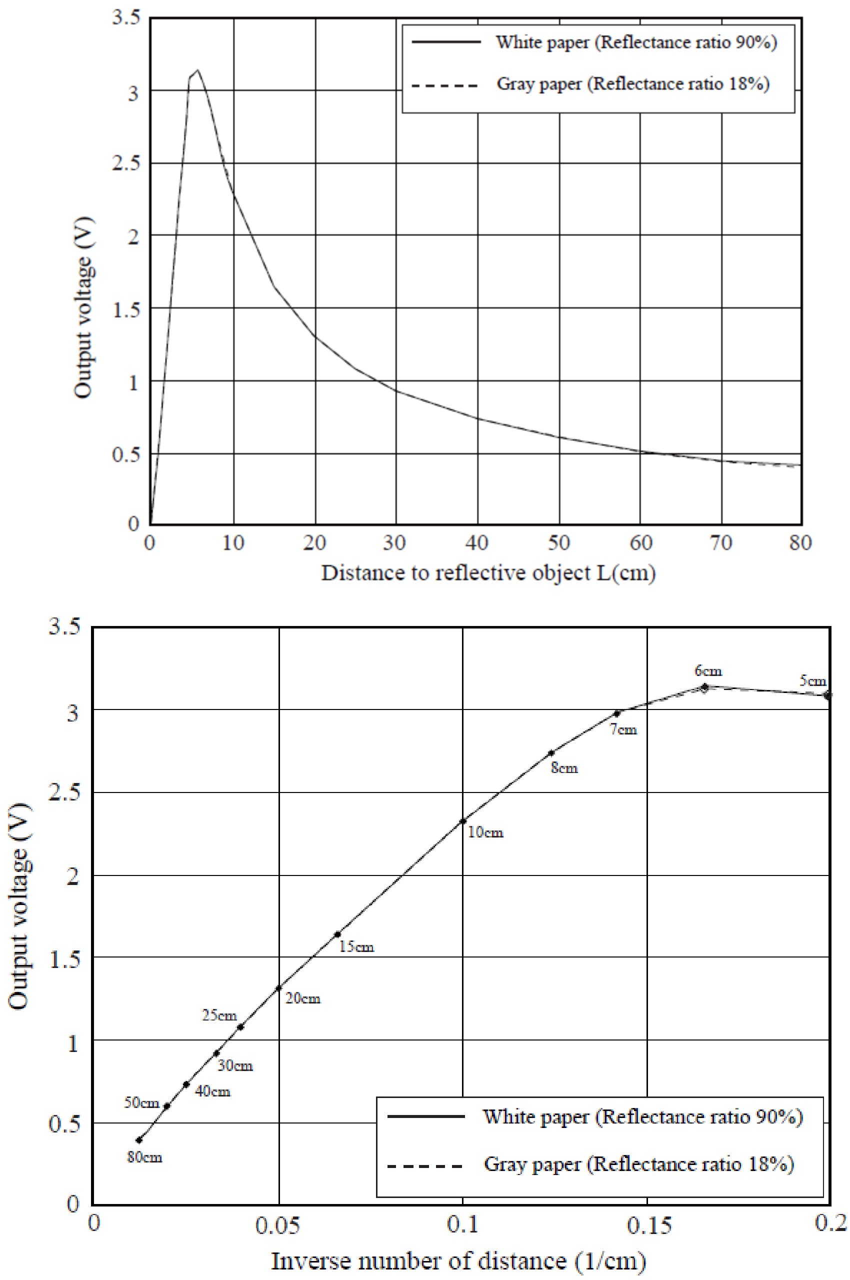
We Will consider the worst case scenario: a 40 mA consumption.

**Fig. 1 Timing chart**




When Reading the values, we have to take into account that we need to wait 55 ms t oread the first stable measurement. This doesn't take into account the delay of the analog to digital conversion. So we cannot expect to work with this sensor with a higher response time than 100 ms (security margin). Which we don't need for our purposes.

**Fig. 2 Example of distance measuring characteristics(output)**




In this graph we can see that if we use the inverse number of distance, we have a quasi-linear behaviour in the rango of 80 cm to 7 cm.
We Will test the sensors in order to obtain a similar graph of our device.
For this test, we Will use a White paper and place the sensor into 8 different distance (80, 70, 60, … , 10). Then we Will test 10 cm to 15 cm with 1 cm increment to test the precisión of the device. We Will do these tests twice to test its accuracy (repeteability).

To obtain a more linear behaviour, a linearization circuit may be applied.

**Maintenance:**
The optics must be kept clean. The lens shouldn’t be washed.
The optical filter which has the most efficient transmittance at the emitting wavelength range of LED for this product (λ = 870 ± 70nm).

● **Advice for the power supply**

• In order to stabilize power supply line, we recommend to insert a by-pass capacitor of 10μF or more between Vcc and GND near this product.

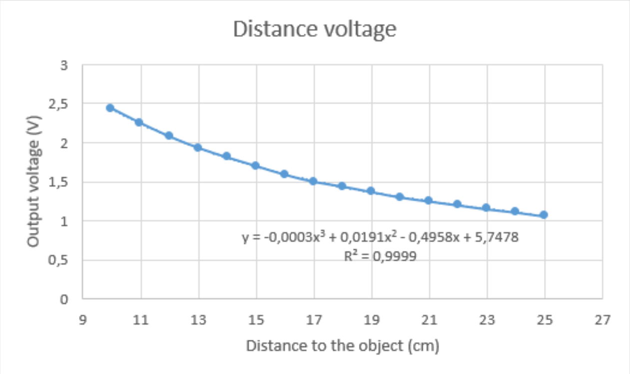
**Wiring:**

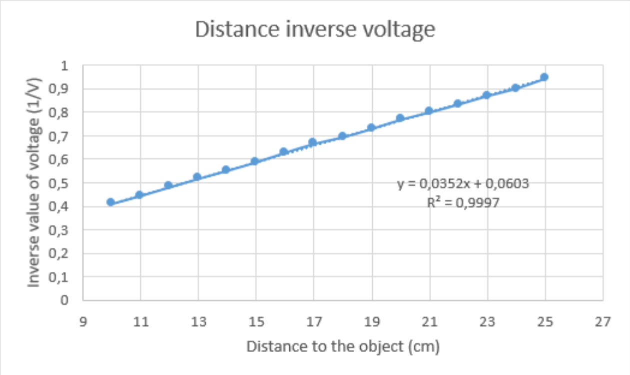
- Red: Power supply (5V)
- Black: Ground
- Yellow: Analog signal (to be measured in an analog pin with the ADC).
- 10 uF capacitor between Power supply and ground to ensure a stable power supply.

**TEST**

I have performed a test for a rango of 10 cm to 25 cm.

| Distance (cm) | Voltage (V) | Inverse V |
|---|---|---|
| 10 | 2,43 | 0,411523 |
| 11 | 2,245 | 0,445434 |
| 12 | 2,075 | 0,481928 |
| 13 | 1,93 | 0,518135 |
| 14 | 1,81 | 0,552486 |
| 15 | 1,7 | 0,588235 |
| 16 | 1,59 | 0,628931 |
| 17 | 1,5 | 0,666667 |
| 18 | 1,44 | 0,694444 |
| 19 | 1,37 | 0,729927 |
| 20 | 1,3 | 0,769231 |
| 21 | 1,25 | 0,8 |
| 22 | 1,2 | 0,833333 |
| 23 | 1,15 | 0,869565 |
| 24 | 1,11 | 0,900901 |
| 25 | 1,06 | 0,943396 |



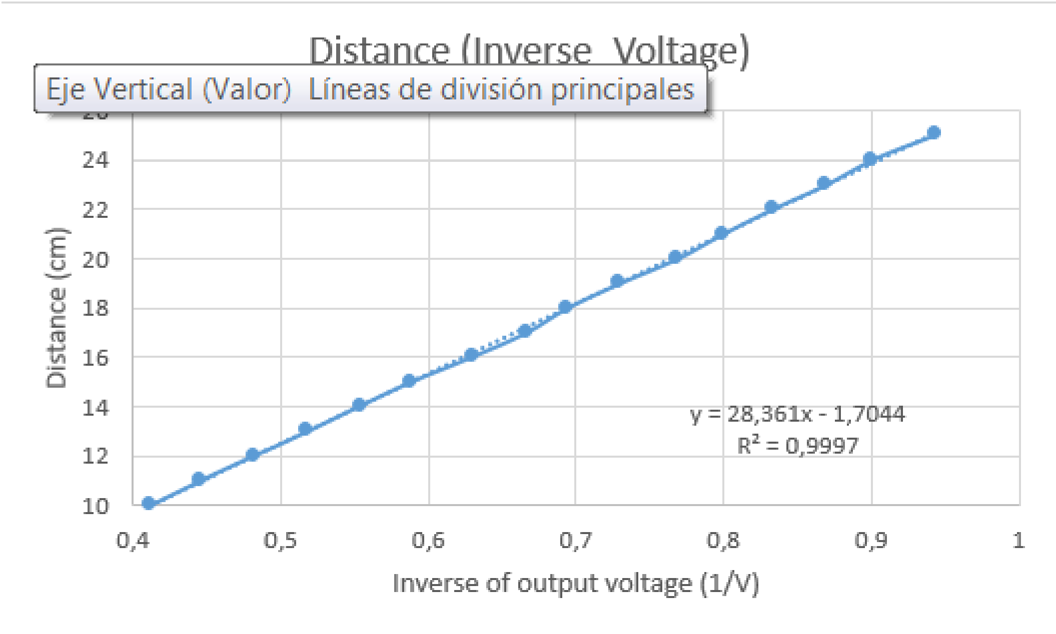




We can find a third degree polynomial approximation for the given rang (10 to 25 cm) with R^2 = 0.9999.

If we compute the inverse value of the voltage as it's recommended in the datasheet, we find a linear approximation with an R^2 = 0.9997. It's much easier for a microcontroller to use a linear function rather than a third-degree function.

If we plot the data with the inverse voltage as the independent value and distance as a dependent value of the inverse voltage, we obtain the function that we Will apply to Arduino to obtain the desired distance.

$$Distance(inverse\ voltage) = inverse\ voltage \cdot 28.361 - 1.7044$$




## Using two distance sensors to measure angle:

Idea:

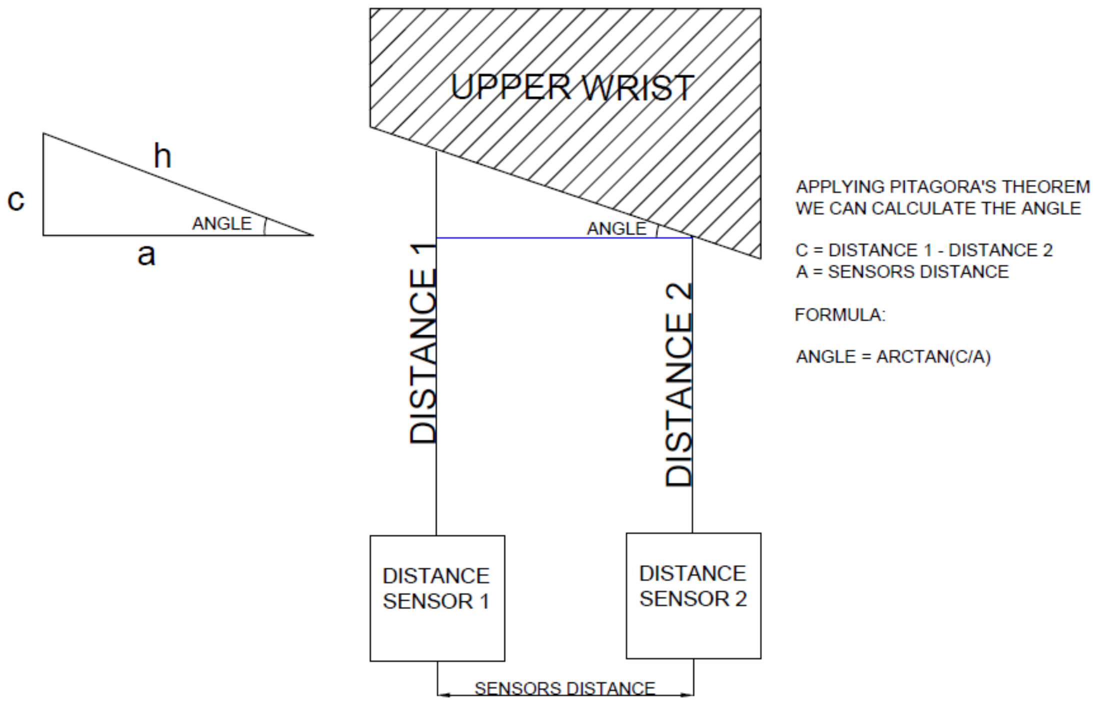
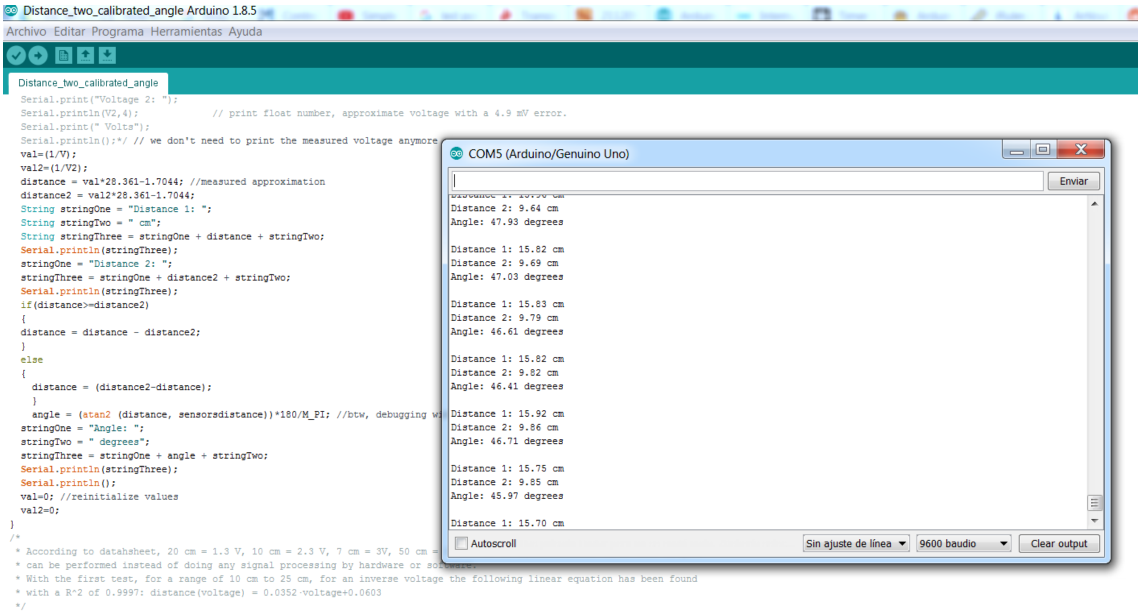



*Figure 1 Schematic of angle measurement. Graph made with Autocad*

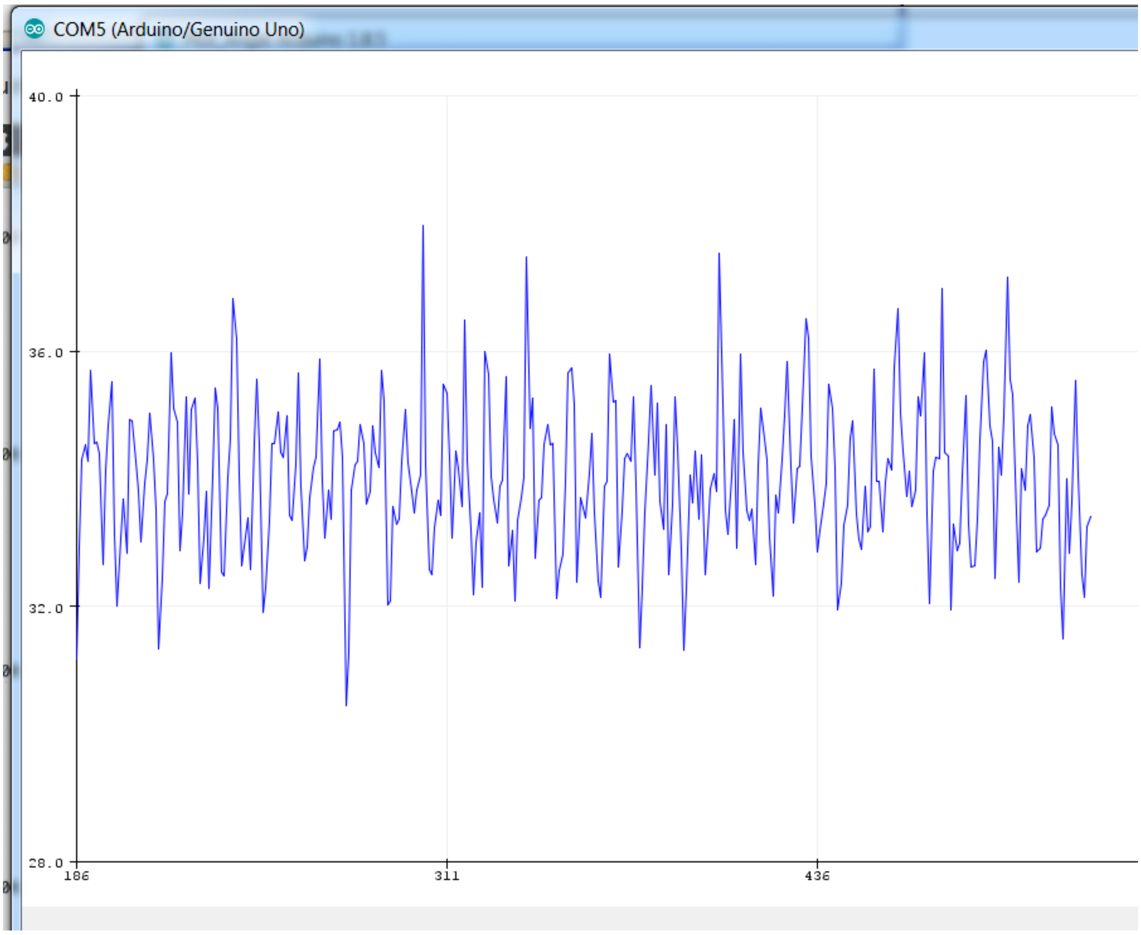
Given a certain distance between the sensors, considering that the distance from the sensor until the inclinated upper wrist placed in each symmetric half can be measured without any trouble (mechanical or for the type of material of the wrist and the sensor sensitivity and precision) we can measure the tilt angle with applying the pitagora's theorem and it's derived trigonometric equivalences.

Arduino testing with the two sensors. Placed at 13 cm and 21 cm. As we can see, placement of the sensor is key to its resolution, just as measurement.

A program has been developed for arduino to test this technique. An averaged value of 10 samplings for the distance has been applied.




We can check this variability with the Arduino built-in plotter for a fixed angle of 35º we can see that the variability is too high.



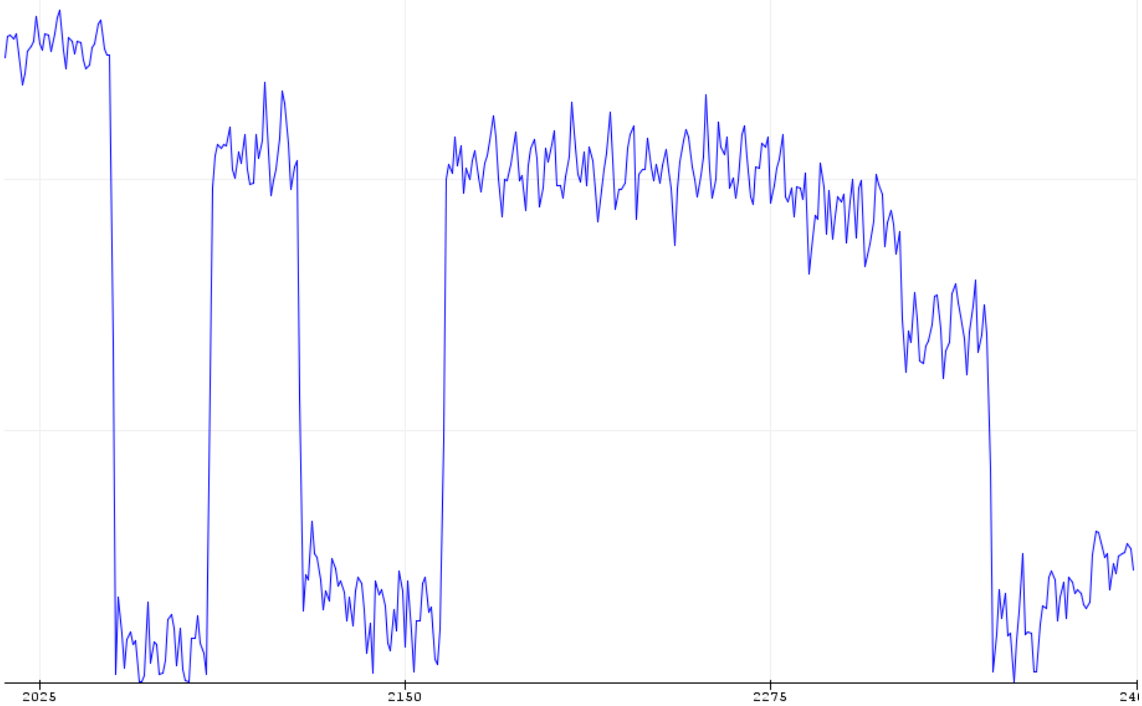

An average value of the angle should be made, but still the averaged angle wouldn't have the required accuracy, repeteability or standard deviation.

It will be required to upgrade the testing conditions of the experiment, since the rulers used and the measurement table, as the sensors' positioning might not be placed good enough (a calibration of the sensors when the team arrives to china will be key for the good operation of the device, just like when a company like ASML sell's its high precision machines to another company, the machine needs to be calibrated).

We also have to take into account limitations of the arduino's ADC and the internal Arduino UNO's computation capabilities, that are limited to a 10-bit ADC and a 16 MHz single core microcontroller. All this can be upgraded with a computer like the MH2.

However, as we can see, this angle measurement method with the two distance sensors is good enough to detect movements with a 5 degree to 10 degree precision.

GRAPH OF ANGLE.

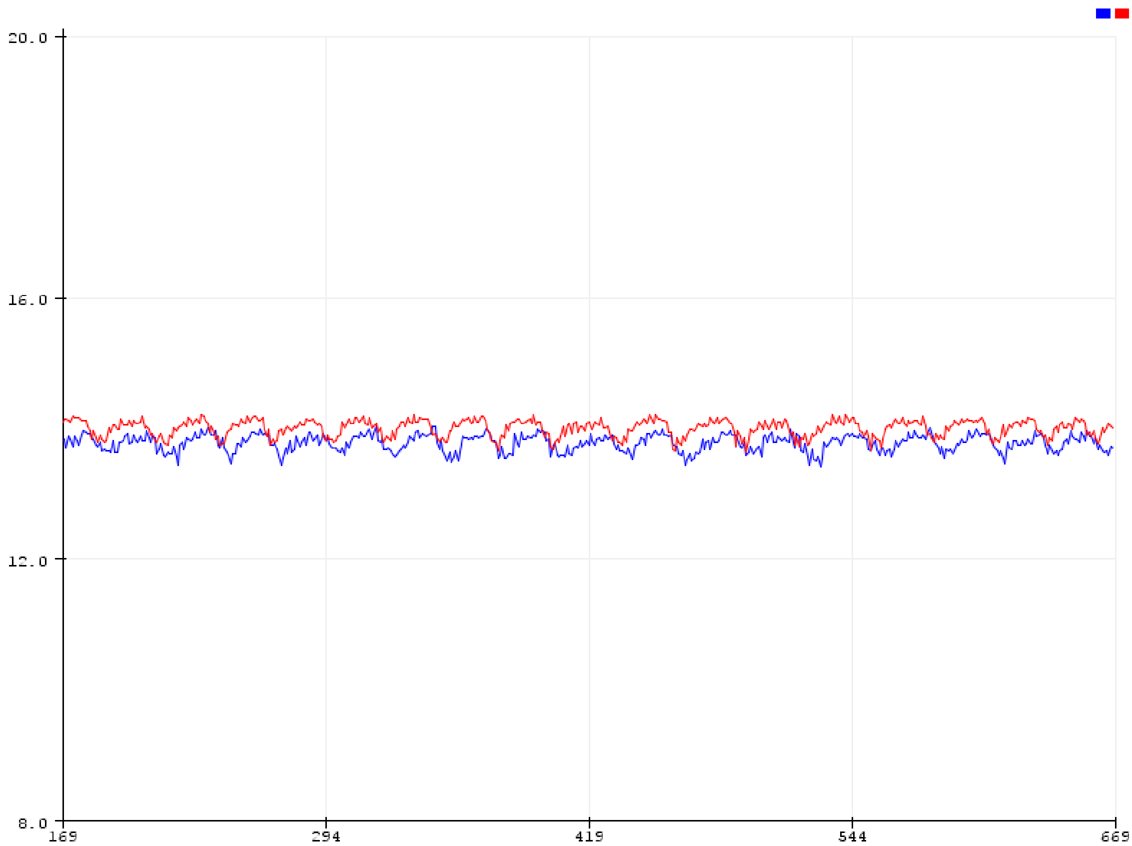



## USING THE DISTANCE SENSOR TO JUST MEASURE DISTANCE

Because the angle measurement is an indirect angle, the results weren't convincing for our purposes. But what about the sensor distances?

The results are good enough to use the sensors as a distance sensor. These sensors can be used to give distances to objects to the robotic arm

GRAPH OF THE DISTANCE SENSORS WITH AN OBJECT TO DISTANCE OF 14 CM. EACH SIGNAL CORRESPOND TO EACH DISTANCE SENSOR




Now if we place the object of an (approximate) 45º with sensor 1 at a distance of 18.5 cm (therefore sensor 2 should be at an approximate distance of:

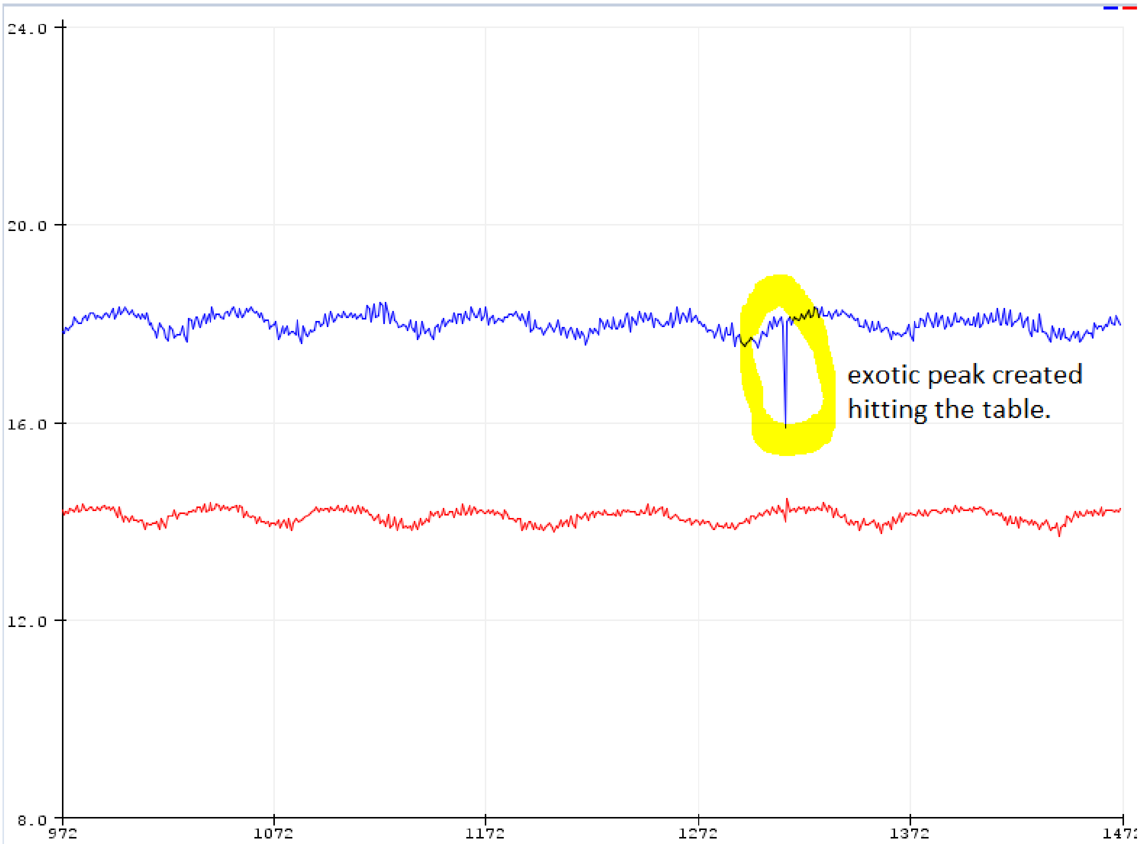
Calculations:

Considering a distance between the sensors Δd of 5.71 cm (approximation, probably it hasn't been properly determined, since the datasheet of the sensor doesn't provide any help for interacting two sensors at the same time to measure position).

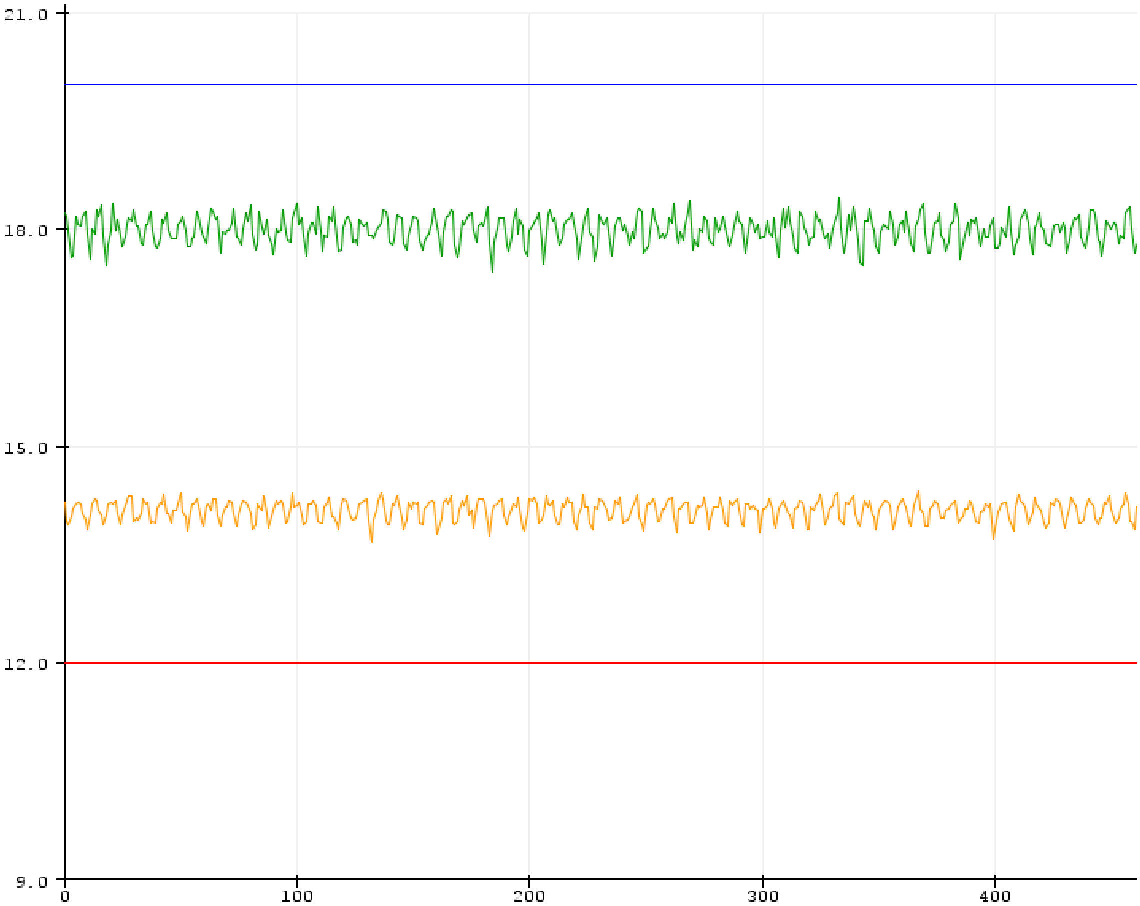
$$c = (distance2 - distance1) = 18.5\ cm - distance\ 1 = \Delta d$$

$$\tan(45º) = \frac{\Delta d}{5.71\ cm} \rightarrow \Delta d = 5.71\ cm$$

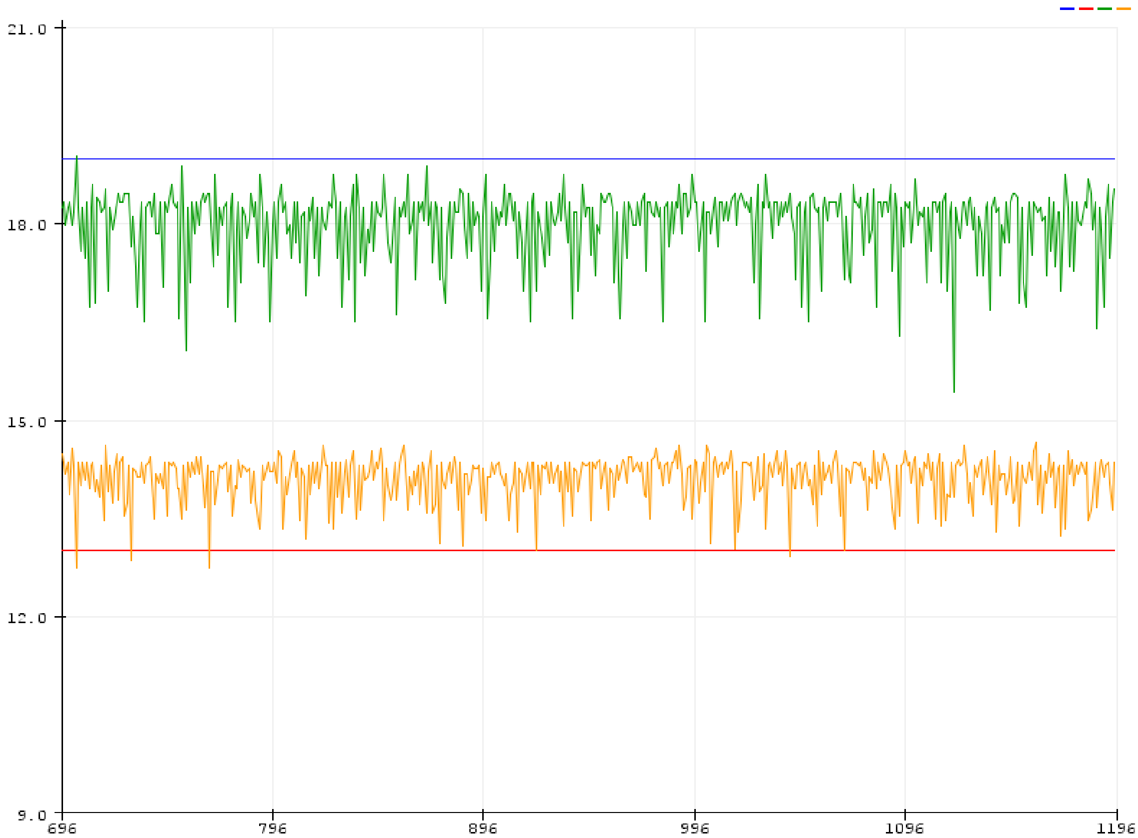
$$distance\ 1 = 18.5cm - 5.71cm = 12.79\ cm$$




Plot with the average and delay time



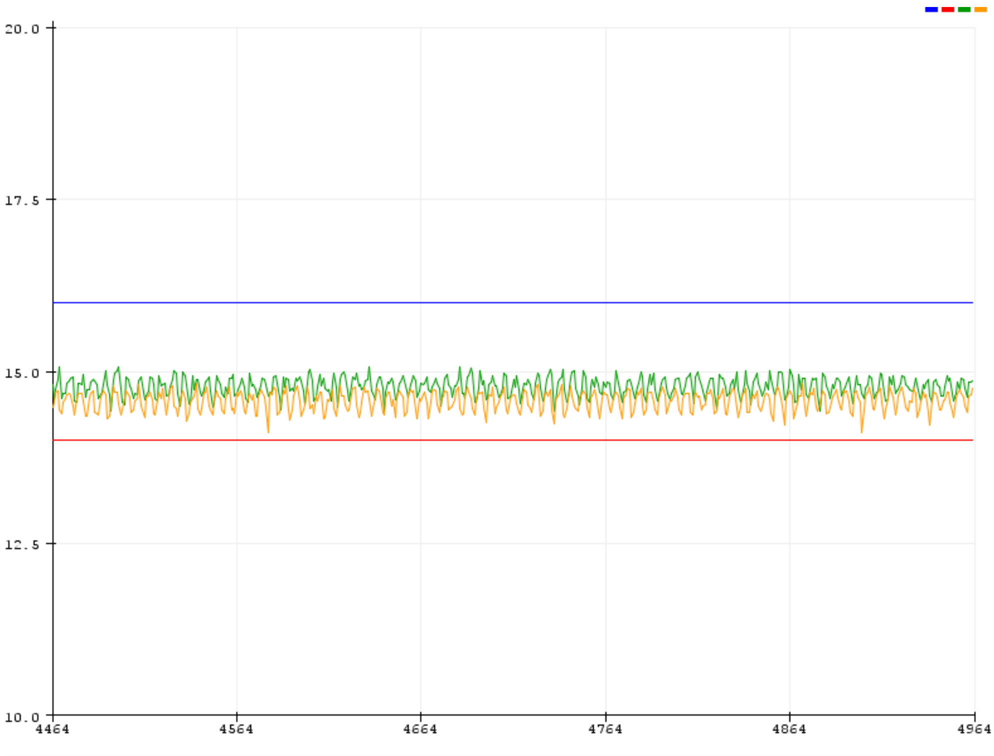
Plot without the average and a reduced delay time of 100 ms




As we can see, without the averaging the quality of the measure declines.

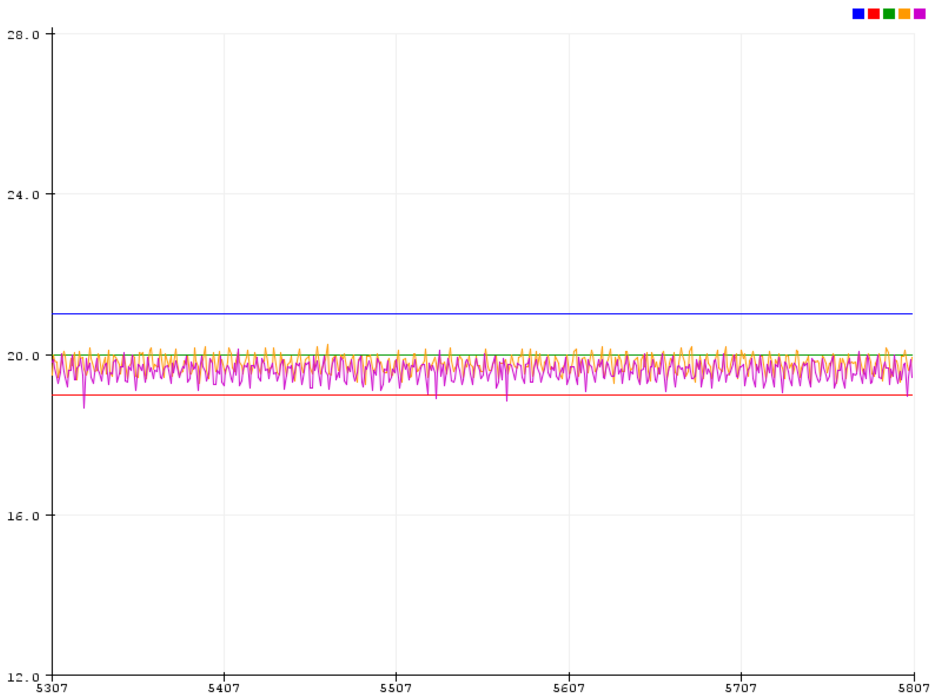
Sensor data at a distance of 15 cm without averaging at approximately a flat angle (0 degrees)

Sensor data at a distance of 15 cm with averaging of 10 samples with a 50 ms delay at approximately a flat angle (0 degrees)



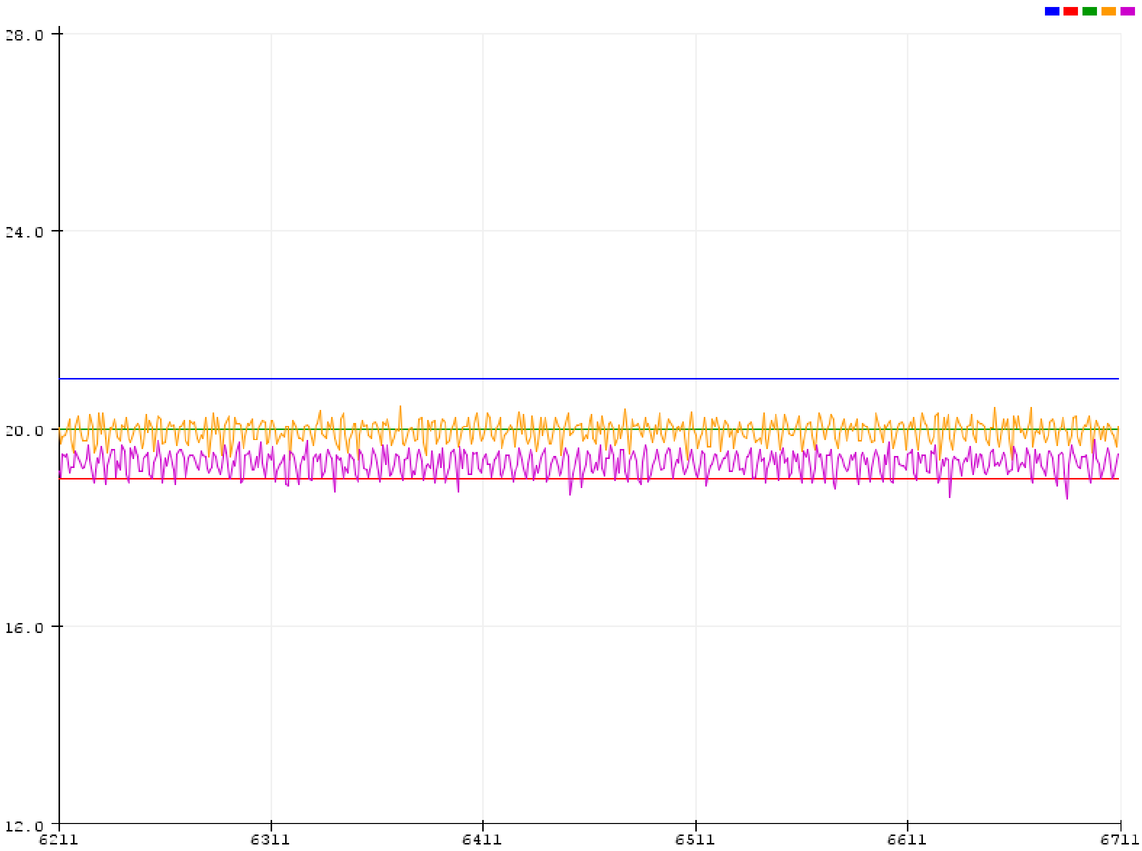

As we can see, variability has been reduced, and the obtained number is much closer to the desired value.

Now for the same measurement at a distance of 20 cm with the averaging:




The measurement is good enough with an approximate margin of 0.5 cm (5 mm).

Now, if we change the measured object from the black box to the brown box, results also differ:




As we can see, sensor 1 has a more accurate measurement of the distance than sensor 2. This is because of the positioning and installation of sensor 2, which has been a little bit worse because it has been added later and hasn't been calibrated that well.