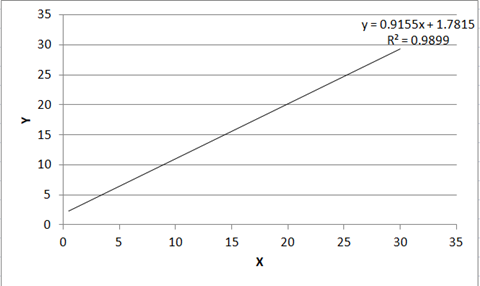
# Measured data and analysis

**(1)Linearity**

Data:

|  |  |  |
| --- | --- | --- |
| X=Move the object | Y=Serial output [Distance(cm)] | Error(%) |
| 0.0 | 0 | INF |
| 0.5 | 4.15 | 730 |
| 1.0 | 3.43 | 243 |
| 1.5 | 3.07 | 104.7 |
| 2.0 | 4.01 | 100.5 |
| 4.0 | 4.10 | 2.5 |
| 6.0 | 6.05 | 0.83 |
| 8.0 | 8.66 | 8.25 |
| 12.0 | 12.04 | 0.33 |
| 18.0 | 18.14 | 0.78 |
| 24.0 | 24.15 | 0.625 |
| 30.0 | 29.76 | 0.8 |



Summary：It is nonlinerity between X and Y when distance below to 4cm, because the distance is too close, so the sound basically does not need the propagation speed, but the sensor is based on the influence of sound speed, which leads to the non-linear relationship between X and y. At 4 to 30 centimeters, the function is linear

**(2)Sensitivity**

Data:

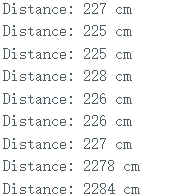
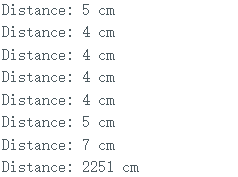
|  |  |  |
| --- | --- | --- |
| X=Move the object | Y=Serial output [Distance(cm)] | Error(%) |
| 0.0 | 2352 | Infinity |
| 0.5 | 4.15 | 730 |
| 1.0 | 3.43 | 243 |
| 1.5 | 3.07 | 104.7 |
| 2.0 | 4.01 | 100.5 |
| 4.0 | 4.10 | 2.5 |
| 6.0 | 6.05 | 0.83 |
| 8.0 | 8.66 | 8.25 |
| 12.0 | 12.04 | 0.33 |
| 18.0 | 18.14 | 0.78 |
| 24.0 | 24.15 | 0.625 |
| 30.0 | 29.76 | 0.8 |

**Summary:** Due to the limited range of the equipment, the sensitivity value is within the acceptable error range within a reasonable range.

**(3)Directivity（measuring range）**

Data:

|  |  |
| --- | --- |
| X=Move the object | Y=Serial output [Distance(cm)] |
| 0 | 2352 |
| 1 | 2450 |
| 2 | 2261 |
| 4 | 4.10 |
| 5 | 5.08 |
| 10 | 10.06 |
| 50 | 50.09 |
| 150 | 152.71 |
| 200 | 219.58 |
| 220 | 223.47 |
| 240 | 2479 |



**Summary:** Through measurement, we know that after 0 to 4 cm and 200 cm, the value is obviously wrong, so we know that range is 4-200 cm. The measuring range is 196cm.

**(4)working temperature （Stability）**

**Summary:** After a long time of testing, we found that it has high stability and can work for a long time.

**(5)resolving power**

**Methods:** duration = pulseIn(echoPin, HIGH);

distance = duration \* sound\_speed / 2;

**Summary:** yes, pulseln()=2distance/sound\_speed