Low-cost ultrasound measurement system for accurate detection of container utilization rate

J.Grönman, P. Rantanen, M.Saari, P.Sillberg and J.Vihervaara
Tampere University of Technology/Pori Department, Pori, Finland
jere.gronman@tut.fi

Abstract - **The increase in the availability of low-cost components has made it possible to design and implement new and innovative devices for the Internet of Things (IoT). Today, cost-effective and open solutions can be created to replace previous expensive and proprietary measurement systems. This study presents the prototype of an ultrasound measurement system. The purpose of the system was to make the surveillance process of waste management containers more effective and reliable. The system was built using commonly available sensor components in combination with a microcontroller, and the collected data was stored and analyzed in a cloud service. The configuration of the ultrasound measurement system is introduced in this study. The measurement results obtained from the tests performed in the actual operating environment are also presented. Based on these tests, the accuracy of the system was observed to be well-suited for the use case. The average systematic error of the results was -2.0 mm and the average of random error was within 0.9 mm.**

I. INTRODUCTION

The Internet of Things (IoT) is the expansion of Internet services, which connects everyday physical objects to a network. This connection between network and physical objects makes it possible to access remote sensor data and to control the physical world from a distance. The first mention of the term ‘IoT’ is said to have come from Kevin Ashton in 1999. A survey of the areas covered by the IoT was made by Atzori, Iera and Morabito [1].

Sensor networks are a highly researched application area in the field of the IoT. A key cost and resource factor in the development of IoT network sensor solutions is prototype implementation.

This study originated from the need for a waste management company to obtain information on the accurate utilization rate of its containers. Previously, the staff needed to check the utilization rate by visiting the site. The aim was to develop this process so that the surface level of the containers could be monitored via a cloud service.

When mapping the different measurement methods, the ultrasound technique was found to be more suitable for measuring the surface level than, for example, laser or infrared techniques. The use of ultrasound was considered more practical because the company prefers a system that does not require a contact or counterpart inside the waste container.

Ultrasound-based sensors are commonly used in the industrial, medical, and service sectors. Ultrasound sensors work above the human hearing range. The most common way to form and detect ultrasound waves is to utilize the piezoelectric phenomenon of some crystalline material, such as quartz. [2] When measuring distance, the ultrasound sensor transmits longitudinal sound waves toward the target by the pulse-echo method. The beam of the sound wave is conical and only objects within it are detected.

Distance measurements with an ultrasound sensor are based on transducer technology. The time spent by the ultrasound pulse traveling from its source to the measurement object and back to the sensor is directly proportional to the distance of the object. Temperature is the most important of the environmental factors that affect the propagation of ultrasound waves. A one-degree rise in temperature increases the sound speed by 0.607 meters per second. In warm air, the measurement result is smaller than in cold air. [3]

The aim of the study was to create a reliable ultrasound measurement system prototype based on open source and low-cost components. The prototype was expected to be able to measure within a margin of error (10 mm) given by the company. The design of the prototype largely takes advantage of the solutions created by the open source community and the literature in the field. The study does not take a stand on the cloud service, the enclosure of the measuring system, nor its installation in the test location.

Earlier studies are presented in Section II. Section III includes the practical part of the study, which consists of the requirements, design and implementation of the ultrasound measurement system and the system model, components and their features, and the connections between them. Section IV analyzes the measurement results produced by the system. Section V includes discussion and suggestions for future research on the topic. Section VI summarizes the study.

II. BACKGROUND

In the scope of this paper, ‘Sensor networks’ refer to distributed autonomous sensors that are used to monitor the physical environment, e.g., temperature or pressure. Sensor networks are a widely studied area. For example, the use of embedded Linux for sensor networks has been proposed [4] and a simple model of a sensor network has been introduced earlier [5]. There has also been research on long-range wireless sensor networks with geolocation tracking [6] and on a low energy algorithm for a sensor network [7]. Sensor networks have several development possibilities such as the one introduced in the Fog Gateway [8] study.

Earlier research by other authors related to this topic was searched mainly using Google Scholar, Scienceport [9] and Researchgate network services. The objective was to find an ultrasound sensor suitable for use with the Arduino Uno microcontroller and the Raspberry Pi 3 computer. The HC-SR04 ultrasound sensor [10] was chosen because it had references from several sources [11] – [13]. In this study, the HC-SR04 was used to measure distance, as in the following papers.

As examined by [11], the operation of an ultrasound measurement system within a PVC pipe. They found that the diameter of the pipe (23 centimeters) was sufficient for the ultrasound sensor to function without a tube. In their article they were unable to say how measurements were carried out (measurements, program code for the measurement, whether there was temperature compensation or not). They measured distances between 5 and 285 cm. The largest systematic error was -4 cm at distances of 283 cm and 285 cm. The measurement results are reported to centimeter accuracy because, according to them, the AT89S51 microcontroller is unable to handle decimals. When examining the results, many of the measured distances appear to be too large in comparison to the actual distance.

In [12], an ultrasound measurement system was assembled to take the temperature into account in the distance measurements. They have calculated the sound rate with using an adiabatic constant and gas constant. In their measurements they observed that a temperature of + 25 ° C in the test conditions increased the error. They also provided results to centimeter accuracy. They made six measurements between 50 and 250 cm. The biggest systematic error was +2 cm at distances between 75 cm and 200 cm.

An ultrasound measurement system was created in [13], and measurements that are more accurate were performed in laboratory conditions using the X-Y table. They measured 20 to 120 mm at intervals of 5 mm to millimeter accuracy, each distance being measured ten times. The largest systematic error was -8.9 mm at a distance of 80 mm. They performed uncertainty calculations as in this study. The maximum random error was ± 1.52 mm at a distance of 70 mm. In the measurement results, the random error was ± 0.64 mm on average.

These papers were selected because they contain measurement results that can be used as a basis for comparison when estimating the accuracy of the system presented in this study. However, in [11] and [12] the measurement results were expressed in centimeters, so it was uncertain whether the results were accurate enough. In addition, it was not clear how the temperature was taken into consideration. Thus, in this study it was relevant to perform tests to millimeter accuracy and with temperature compensation. On the other hand, according to [13] the ultrasound sensor was too inaccurate for commercial use as an electronic measurement device, but accurate enough for obstacle detection.

III. MEASUREMENT SYSTEM

A. Requirements

The purpose of this study was to implement a system of low-cost components and to explore new technologies. Ultrasound technology was chosen because it is a non-destructive method and no counterpart is needed in the measuring target. The material of the measurement object can be almost anything [14]. Ultrasound sensors are relatively inexpensive, compact and low-power. The components of the system, in addition to the ultrasound sensor, are intended for experimental use, but in some cases are suitable for replacing more expensive options. Fig. 1 shows the measurement system process.

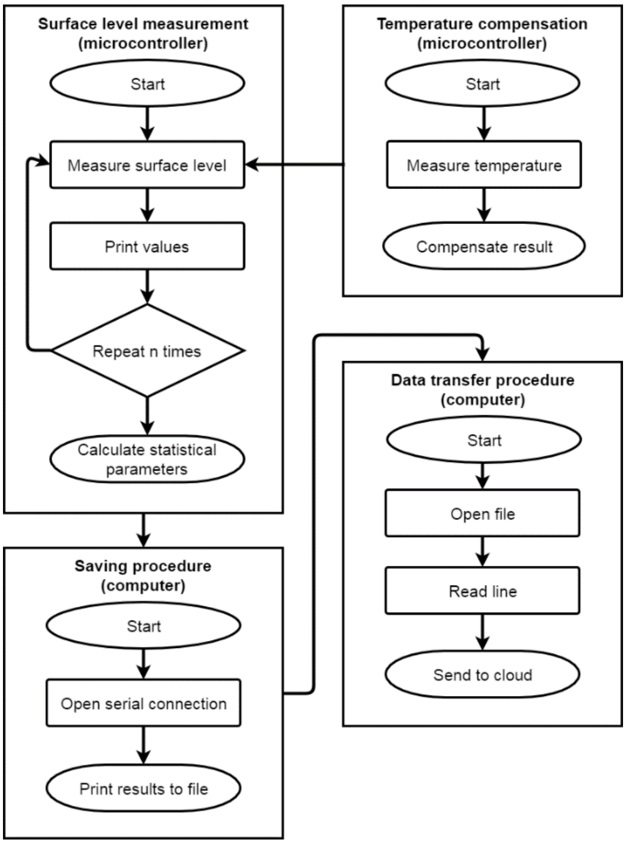

Figure 1. Process chart of the measurement system

When the prototype is started, the ultrasound sensor begins to measure the distance to the target at certain predefined intervals. The *microcontroller* in the system processes the measurement results obtained from the ultrasound sensor, taking into account the temperature received from the temperature sensor. This is repeated as often as it has been defined, after which the statistical results are calculated from the measurement results. The *computer* reads the values printed by the *microcontroller* via the serial port connection. Then it saves the received values to a file. After saving, the line in the file is read and sent to the cloud service.

B. Design

The measurement system consists of a microcontroller, a computer, an ultrasound sensor, a temperature sensor, and a 3G / 4G wireless modem. Based on earlier studies [11] – [13] the following components were selected: the Arduino Uno microcontroller, the Raspberry Pi 3 computer, the HC-SR04 ultrasound sensor, the BME280 sensor unit [15] and the Huawei E5377 wireless 3G / 4G modem [16]. The choice of system components was also influenced by their availability in the IT field, their affordability, low power consumption, physical size and customizability. Fig. 2 shows a deployment chart of the measurement system.

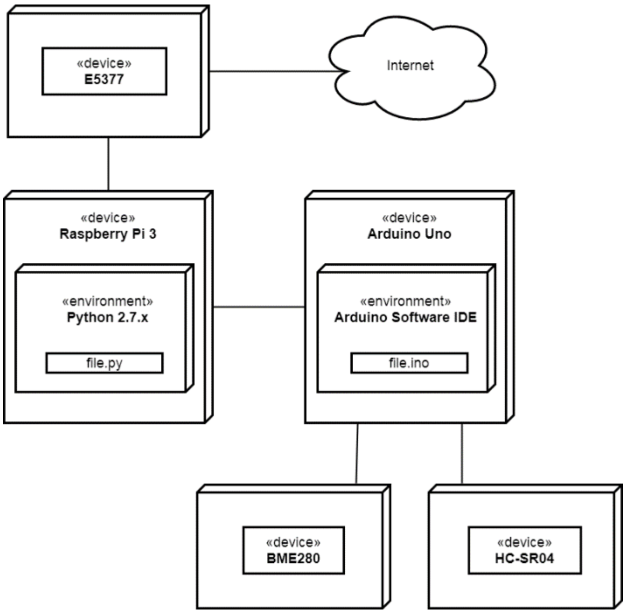

Figure 2. Deployment chart of the measurement system

The primary reason for choosing Raspberry Pi 3 was cost-effectiveness. Any other device capable of Internet connectivity could also replace Raspberry Pi 3. The Arduino Uno could also be directly connected to the Internet by using Ethernet or Wireless shield.

C. Implementation

The components of the measuring system are easily available in online stores, e.g., Amazon.com. In this study, the components were purchased from Finnish online stores. The assembly of the system was started by connecting the HC-SR04 ultrasound sensor and the BME280 sensor unit to the Arduino Uno. The ultrasound sensor's IC power supply pin (VCC) and ground (GND) pins were connected to the Arduino Uno's 5V pin and GND pin. Trig and Echo can be connected to any of the Arduino Uno's digital pins (2-13). The BME280 sensor unit's Vin pin and GND were connected to the Arduino Uno's 3.3V pin and GND pin. SCL and SDA pins can be connected to any of the Arduino Uno's analog pins (A0-A5). Table II shows the connections between the sensors and Arduino Uno.

TABLE I. CONNECTIONS BETWEEN SENSORS AND ARDUINO UNO

HC-SR04 ultrasound sensor	**Arduino Uno**
VCC	5 V
Trig	Digital 10
Echo	Digital 8
GND	GND
BME280 temperature sensor	
VIN	3.3 V
GND	GND
SCL	Analog A4
SDA	Analog A5

Arduino Uno is connected to Raspberry Pi 3 using a USB cable. There is a wireless WLAN connection between the Raspberry Pi 3 and the Huawei E5377 modem. Fig. 3 shows the system connections.

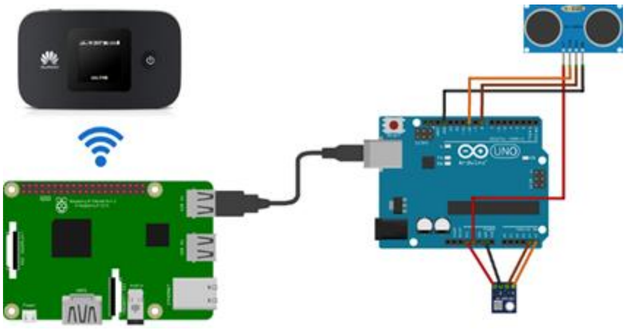

Figure 3. Connections

The Raspberry Pi 3 is equipped with a desktop version of the Raspbian operating system. After installation, it is recommended to upgrade the Raspbian operating system and package repositories instance to a python-serial library [17] that allows serial connection with Arduino Uno. Additionally, Arduino IDE development environment and the necessary NewPing [18], BME280 [19] and QuickStats [20] libraries were installed.

The NewPing library has drivers for many different ultrasound sensors, including HC-SR04. The drivers include the program code that controls the sensor operation. The code is freely available and editable, but its own implementations must be kept separate from the original version management branch.

BME280 is a software library for the BME280 sensor unit manufactured by Bosch. It reads temperature, humidity and pressure values as well as height and dew point. The code is compliant with the GNU GPL (General Public License), which entitles users to access, copy, modify and distribute programs and their source code.

The QuickStats library is a collection of functions used to calculate statistical key figures. It includes functions for calculating the mean, minimum, maximum, standard deviation, mean error, variation coefficient, median, and mode of the sample.

Python code is used to read, save, and send data and is executed on a Raspberry Pi 3 computer. The above tasks are divided into two Python files. The first one reads the serial port connection and saves the measurement results to the file. Then the second one reads the line of information that it sends to the cloud service.

IV. EXPERIMENTS AND ANALYSIS

The system test was performed in a warehouse building, where the values given by the system were compared to the values provided by using a tape measure. The accuracy of the measuring tape is 1 millimeter. Other devices used in the measurement were a forklift truck and a line laser. The ultrasound sensor was attached to the forklift and the level was set horizontally. The measurement was carried out at a temperature of 5.4 to 5.6 ° C and the humidity remained between 77.2 and 79.8 %. Samples were taken at 48 different distances between 18.0 and 3791.0 mm, and one sample included 90 measurements. Measurements 1 to 10 were done at intervals of approx. 10 millimeters and the remaining 11 to 48 at about 100 millimeters. The measurement results were stored on a Raspberry Pi 3 computer via a serial port connection.

A. *Uncertainty analysis*

Systematic and random errors were calculated from the measurement results using a spreadsheet program. The systematic error was calculated by subtracting the mean of the measured values from the actual distance.

$$\pm\sqrt{\left(\frac{s}{\sqrt{n}}\right)^2+\left(\frac{a}{2\sqrt{3}}\right)^2}*k, \qquad (1)$$

The random error was calculated using an expanded uncertainty (1), where s is the standard deviation of the measurement results and is divided by the square root of the number of measurements. The measuring tape has an accuracy of 1.0 (a) mm and is uniform, so it was divided by the square root of three. The square root was multiplied by two because the aim was to obtain a total range. The standard uncertainties were raised to the power of 2 and the sum of the square root was calculated. This was multiplied by the coverage factor obtained from the Welch-Satterthwaite equation. A coverage factor was needed to extend the 68% probability in this case to 95.45% probability. With this probability and the number of degrees of freedom (89), a coefficient of 2.0285 was obtained. [21]

Using the statistical key figures in uncertainty calculations shows that the average systematic error of the results was -2.0 mm and the average of random error was ±0.9 mm. The largest systematic error is -8.3 mm at a distance of 1089.0 mm. The smallest systematic error was 0.1 mm at a distance of 3191.0 mm.

B. *Unequal variance t-test*

Unequal variance t-test, also known as the Welch's t-test, was used to analyze the measurement results to examine the means of the two independent groups. Welch's t-test is used when variances are unequal and there is an assumption of normality. It is an adaptation of Student's t-test assumption, where variances are equal and two groups have normal distributions. [22]

In this study, the groups were the actual distances and the measured distances. To determine the mean deviations of the groups, the ttest2 function of the MathWorks MATLAB R2016b numerical computing program was used. The measurement results were recorded in mat files of 48 samples. Each of them contained information about the actual distance (x) and 90 measurements (y). The null hypothesis H0 was used: mean deviations are less than the value of the *mm_maara* variable. As an alternative hypothesis, H1 was used: mean deviations are more than the *mm_maara* variable value. The parameters x, y, risk level (*Alpha*), type of alternative hypothesis (*Tail*), and variance type (*Vartype*) were passed to the ttest2 function.

In this case, *Alpha* can be either 0.05, 0.01, or 0.001. An alternative hypothesis can be used for the value *both*, which means that the differences between the mean values of the groups in each direction are explained. The variance type was *unequal* because the variance of the groups differs from one another. Fig. 4 includes a MATLAB code for evaluating mean difference variables compared to a set variable representing the measured error.

```
alfa = 0.05
mm_maara=3.0;
for i = 1:48
   load (['y_' num2str(i) '.mat'], 'x', 'y')
   temp_x = x * ones(1,90);
   if x>=mean(y)
      temp_x=temp_x-mm_maara;
   else
      y=y-mm_maara;
   end
[hyp_equal(i),p_equal(i)]=test2(temp_x,y,'Alpha',alfa,'Tail','both,
'Vartype','unequal');
end
hyp_equal
sum(hyp_equal)
```

Figure 4. MATLAB code for evaluating mean difference variables compared to a set variable representing the measured error.

The MATLAB code in Fig.4 was run at risk levels of *0.05*, *0.01*, *0.001* and measurement error values of 3 to 10 millimeters. Table 2 includes probabilities at a particular *risk level* and *measurement error*. The values of the risk level columns indicate how many samples are accepted by the null hypothesis, and the parentheses indicate the relative proportion of 48 samples.

TABLE II. THE PROBABILITY OF MEASUREMENT ERRORS AT DIFFERENT RISK LEVELS BETWEEN 3 AND 10 MM

Measurement error	**Risk level**		
	0.05	*0.01*	*0.001*
3,0	36 (75.0 %)	33 (68.8 %)	31 (64.6 %)
4,0	38 (79.2 %)	34 (70.8 %)	34 (70.8 %)
5,0	45 (93.8 %)	43 (89.6 %)	41 (85.4 %)
6,0	44 (91.7 %)	43 (89.6 %)	43 (89.6 %)
7,0	45 (93.8 %)	45 (93.8 %)	44 (91.7 %)
8,0	46 (95.8 %)	46 (95.8 %)	46 (95.8 %)
9,0	48 (100 %)	46 (95.8 %)	46 (95.8 %)
10,0	48 (100 %)	48 (100 %)	48 (100 %)

From the results of the Welch t-test, it can be seen that the accuracy of 3.0 mm provided by the ultrasound sensor manufacturer is best achieved at a risk level of *0.05* 75.0% of the distances measured in the cases. If it is desired that the measurement error could be met at a risk level of *0.05* within the tolerance allowed at all measurement distances, the tolerance should be increased to 9.0 mm. Given the original aim of this study, it can be stated that the measurement system produces sufficiently accurate results.

V. DISCUSSION AND FUTURE RESEARCH

This paper presents a low-cost ultrasound measurement system, which utilizes an Arduino Uno microcontroller, HC-SR04 ultrasound sensor, BME280 sensor unit, and Raspberry Pi 3 computer. The aim was to test the accuracy of the results that the system produces. It can be said that results were accurate enough for the use case.

Although the measurement system provided statistical key figures, they are not presented in this paper. The statistical results as well as the mathematical theory of uncertainty analysis can be examined in previous research [23] related to this paper.

Another issue beyond the scope of this paper is energy efficiency. The components chosen for the example system have low power consumption, but extensive measurements regarding the power usage of the implemented system were not performed.

As a future study, measurements could be made using multiple sensors, different surface materials, shapes and temperatures. It is appropriate to perform a time series analysis of sensors to gain more information about their power usage. It is also recommended to carry out measurements over a longer term to test the durability of the system components under different conditions.

VI. CONCLUSION

In this study, a prototype of an ultrasound measurement system was created. It met its goal of staying within a margin of error of 10 millimeters when measuring the utilization rate of waste containers. The result of the study is that it is possible to implement a reliable and accurate ultrasound measurement system with low-cost components.

REFERENCES

[1] L. Atzori, A. Iera and G. Morabito. "The Internet of Things: A survey," Computer Networks, 54(15), 2010, pp. 2787–2805.

[2] I. Ihara. "Ultrasonic Sensing: Fundamentals and its Applications to Nondestructive Evaluation. Sensors: Advancements in Modeling, Design Issues, Fabrication and Practical Applications," Ed. Subhas Chandra Mukhopadhyay and Yueh-Min Huang, Springer-Verlag Berlin Heidelberg, 2008.

[3] B. Nalini, V.Nandhini, E. Kavitha and R. Chandralekha, "Implementation of Temperature Compensation Technique with Ultrasonic Ranging for Obstacle Identification," International Journal for Research and Development in Engineering (IJRDE), Special Issue, 2014, pp. 230-234.

[4] M. Saari, P. Sillberg, P. Rantanen, J. Soini, and H. Fukai, "Data collector service - practical approach with embedded Linux," 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2015 pp. 1037–1041.

[5] M. Saari, A. Muzaffar, P. Sillberg, P. Rantanen and J. Soini, "Embedded Linux controlled sensor network," 39th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO, 2016

[6] A. Muzaffar and W. Yan, "Long-Range Wireless Sensor Networks for Geo-location Tracking : Design and Evaluation," In the 18th International Electronics Symposium (IES), Bali, Indonesia, 29-30th September 2016.

[7] A. Muzaffar, M. Saari, P. Sillberg, P. Rantanen, J. Soini and T. Kuroda, "Low-energy algorithm for self-controlled Wireless Sensor Nodes," International Conference on Wireless Networks and Mobile Communications (WINCOM), 2016 pp. 42–46. IEEE.

[8] A. Muzaffar bin Baharudin, M. Saari, P. Sillberg, P. Rantanen, J. Soini, and W. Yan, "Portable Fog Gateways for Resilient Sensors Data Aggregation" in Internetless Environment in International Symposium of the 11 th SSMS and the 5 th RCND, 2017.

[9] Scienceport, "Information Technology: Articles and databases," http://scienceport.tut.fi/information-technology Retvieved January 15, 2018.

[10] HC-SR04, "Ultrasonic Ranging Module HC-SR04," http://www.micropik.com/PDF/HCSR04.pdf Retvieved January 15, 2018.

[11] M. Saraswati, E. Kuantama and P. Mardjoko, "Design and Construction of Water Level Measurement System Accessible through SMS," UKSim/AMSS Europe-an Symposium on Computer Modelling and Simulation (EMS), November 14-16, Malta, 2012.

[12] M. Suleiman, G.I. Saidu, M.I Ilyasu, O.A Adeboye and M. Hamza, "Ultrasonic Fluid Level Measuring Device," International Journal of Research in Science, Vol. 1, No.1, 2015, pp. 27-31.

[13] A.M Nakatani, A.V Guimarães and V.M Neto, "Medição com Sensor Ultrassônico HC-SR04," Universidade Tecnológica Federal do Paraná – UTFPR, Departa-mento de Electrônica, Curitiba, Brazil, 2014.

[14] N. Karlsson and O. Pedersen, "Ultrasonic Displacement Sensors. Measurement, Instrumentation, and Sensors Handbook, Second Edition: Spatial, Mechanical, Thermal, and Radiation Measurement," Ed. John G. Webster & Halit Eren, CRC Press, 2014.

[15] Bosch, "Final Datasheet BME280 Environmental sensor," https://cdn-shop.adafruit.com/datasheets/BST-BME280_DS001-10.pdf Retvieved January 15, 2018.

[16] Huawei, "Huawei E5377 Specifications," https://consumer.huawei.com/en/mobile-broadband/e5377/specs/ Retvieved January 15, 2018.

[17] PySerial, "Python-serial library for Raspberry Pi," https://github.com/pyserial/pyserial Retvieved January 15, 2018.

[18] T. Eckel, "NewPing Arduino Library for Arduino," https://bitbucket.org/teckel12/arduino-new-ping/wiki/Home Retrieved January 15, 2018.

[19] T. Glenn, "BME280 Arduino Library," http://github.com/finitespace/BME280 Retvieved January 15, 2018.

[20] D. Dubins, "QuickStats - Descriptive Statistics for Arduino Float Arrays," https://github.com/dndubins/QuickStats Retvieved January 15, 2018.

[21] GUM, "Evaluation of measurement data — Guide to the expression of uncertainty in measurement," https://www.bipm.org/utils/common/documents/jcgm/JCGM_100_2008_E.pdf Retvieved January 15, 2018.

[22] G.D Ruxton, "The unequal variance t-test is an underused alternative to Student's t-test and the Mann–Whitney U test," Behavioral Ecology, Volume 17, Issue 4, 1 July 2006, Pages 688–690.

[23] J. Grönman, "Ultraäänimittausjärjestelmän toteutus. Diplomityö - Master's thesis" https://dspace.cc.tut.fi/dpub/handle/123456789/24601 Retvieved January 15, 2018