International Journal for Innovation Education and Research

ISSN: 2411-2933

Measuring interference with ultrasound and Arduino – How to build an automated ultrasonic scanner

Fanni Vitkóczi;Károly Piláth

Abstract

In this paper we describe the construction and application of a simple and inexpensive measurement tool that can be used in the improvement of the demonstration of wave-phenomena, and can help studensts to understand interference in physics education. Studying or even redesigning of the algorithm controlling the measurement and recording the data can give students an insight into an algorithmic thinking process and helps in implementing programming skills obtained in other classes.

Keyword: Data Streamer; Young's experiment; diffraction; physics; education;

Published Date: 7/1/2020 **Page.60-75** **Vol 8 No 07 2020**

DOI: https://doi.org/10.31686/ijier.vol8.iss7.2447

Measuring interference with ultrasound and Arduino – How to build an automated ultrasonic scanner

Fanni Vitkóczi
Eötvös Loránd University
Hungary

Károly Piláth
Eötvös Lorand University Trefort Ágoston (Secondary) Training School
Hungary

Abstract

In this paper we describe the construction and application of a simple and inexpensive measurement tool that can be used in the improvement of the demonstration of wave-phenomena, and can help studensts to understand interference in physics education. Studying or even redesigning of the algorithm controlling the measurement and recording the data can give students an insight into an algorithmic thinking process and helps in implementing programming skills obtained in other classes.

Keywords: Data Streamer; Young's experiment; diffraction; physics; education;

1. Introduction

In many physics teachers oppinion, physics education needs some refreshment. One likely way to make physics lessons more interesting, and make studenst to be interested in physics and science is showing new technologies and develpoment in experiments and using new, innovative devices [1].

The interference patterns of light can be demonstrated using a simple screen, but the behaviour of surface waves is easier to observe, creating them on a sufficiently large water surface. However a special and expensive device, a wave tank is needed to perform such experiments. Examples for the experimental application of ultrasonic waves in the 40 kHz region have already been described in the international literature [2]–[4]. These papers were published shortly after the appearance of the popular and low-cost sensors we also use. One of the first articles presenting basic demonstrational experiments with ultrasonic waves was published in 2003 [5]. PHYWE Systeme GmbH &Co has also begun to produce experimental equipment to study the topic, which also shows the popularity of these new educational methods. In this paper we are showing a way to look into the attributes of ultrasonic waves that are getting more and more widespread in many common areas of life (e.g., medical diagnostics, and vehicles).

We designed a very useful educational equipment made from low-cost components and repurposed parts of outdated electronic devices. Based on this paper, it can be reproduced by anyone having basic electronic knowledge. We tried to develop an automated measuring system capable of spectacularly and obviously demonstrating the interference shown by ultrasonic waves. The source of the ultrasonic wave was made from the transmitter of a distance measurement ultrasonic sensor. These sources are based on piezoelectricity, the resonance frequency is 40 kHz, thus the wavelength of the ultrasonic wave in air is approximately 8.5 mm. This wavelength makes it possible to observe the intensity of the forming wave front by moving a detector (microphone as an ultrasonic receiver) along a line. Measuring the sign of the moving microphone, we get a graph about the intensity of soundwaves depending on the location. To scan the wave front, we used the machinery of an old, disused document scanner.

2. The structure of the device

The stand of the microphone was attached to the table of the scanner, moved by a toothed belt. The microphone moves along a path of approx. 20 cm. Before attaching the microphone, do not forget to gently remove the glass over the machinery. Discard the useless cards and sensors. The microphone moved by the scanner is measuring the intensity of soundwaves depending on the location. The toothed belt mechanism is run by a T13119635 type unipolar stepper motor. At first we have to solve the control of the motor. We can use any other type old, cast-off scanner, because most of the flatbed types have similar operating principles. If you do not feel like recycling an old document scanner, it can be also used a HPV4 MINI V Linear Actuator Set [6] (Fig. 1.), which is a ready-to-use stepper motor.

Figure 1 HPV4 MINI V Linear Actuator Set [6]

During the development of the measurement tool we intended to build in low-cost ingredients, that can easily be obtained from webshops like Ebay or Aliexpress. The measuring tool is controlled by an Arduino microcontroller, shown in a block diagram in Fig. 2. This controller sends a set of instructions at D2-D5 digital output to the driver board (ULN2003) [7], [8] of the stepper motor (EPOCH T1319635). Furthermore this card completes every other task during the measurement, thus it manages the alternating current (voltage) in the circuit of the digital signal generator (AD9850) [9], [10] of the ultrasonic transmitter.

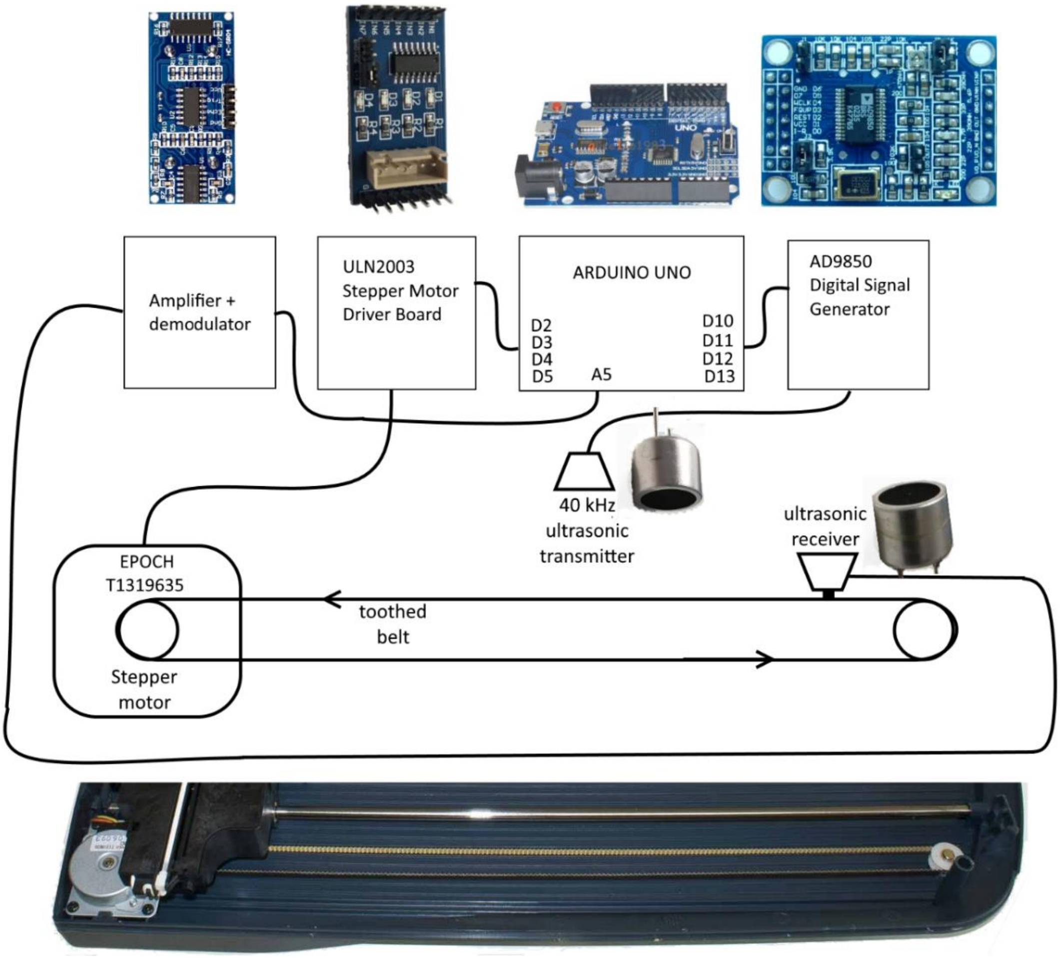

Figure 2 The block diagram of the self-devised automated ultrasonic scanner

A short section of the source code configures the Arduino card via D10-D13 outputs at the beginning of the measurement. On the output, a 40 kHz sinusoidal voltage will be generated, which is directly connected to the ultrasonic transmitter. To measure the intensity of ultrasonic waves, we used a modified ultrasonic distance measuring sensor, type HC-SR04. This device was developed to work with Arduino or other microcontrollers and works on 40 kHz frequency. The sensor is pictured in Fig. 3.

Figure 3 A low-cost ultrasonic distance measuring sensor [11]

The conception of the HC-SR04 sensor is shown in the circuit draw [12] in Figure 4. It is very simple to solder out the T (transmitter) signed "speaker" from this device, and we can modify the other, receiver side to create a suitable detector. In Figure 3. It can be seen, that the R (Receiver) microphone is connected to a U2D preamplifier, which was created from the 2. operation amplifier of an LM324 integrated circuit. The preamplifier is set to 6 times amplification. The increased sign of the microphone is connected to a 40 kHz bandpass filter, which is followed by another 10 times amplifier device (U2B). The fourth operation amplifier (U2A) of the integrated circuit is a comparator - we do not need this at all. We would like to use the remaining part of the circuit to measure the intensity of ultrasound, thus we have to add some more elements. The extended circuit was shown in Figure. 4. This figure shows the added parts in blue border, which is a simple single-phase rectifier. The main part is an OA 1182 type germanium diode, but a silicon diode (e. g.: 1N4148) is also suitable. The rectified signal is integrated by a C9-R16 filter, and measured by the Arduino connected to A5 analogue input. This voltage is commensurate (proportional) with the intensity of the detected ultrasounds. The PCB of the modified hardware could be redesigned, but the modification of the original one is so simple that it is not worth it to design and manufacture a new PCB when only needing a few examples. It is the most useful to begin with the transformation of the original PCB by removing the ‘speaker’ marked with T. Let us store the speaker, because later this must be connected to the analogue output of the sound generating card. The amplified signal of the microphone can be taken from the 7th pin (U2B output) of the integrated circuit LM324. Thus, the anode of the diode must be soldered here. Afterwards, the rest of the elements must be connected. The circuit operates from a 5V power supply, so a connector ending in two pins must be prepared from two cables that can be used to connect our modified circuit to the GND and 5V connector pair of the Arduino.

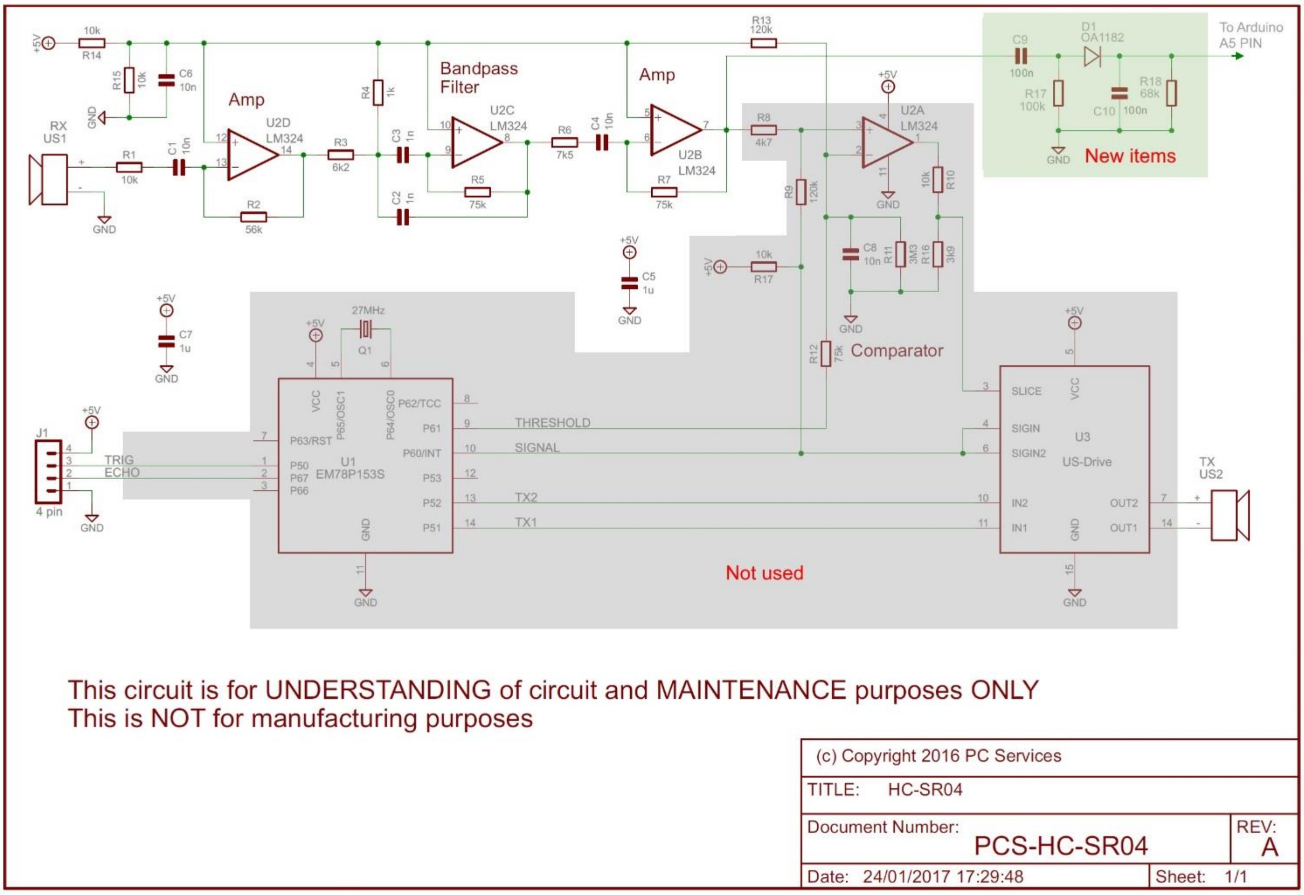

Figure 4 The circuit draw of the conception of the HC-SR04 sensor [12] and an added simple single-phase rectifier

The pins of the ultrasonic speaker (previously removed from the distance measuring sensor) must be connected to the signal generator controlled by the Arduino using a pair of light and flexible cables. At the end of the conversion the receiver sensor (the microphone marked with R, serving the detection of ultrasounds) must also be removed and connected to its original connection points at the demodulator using a flexible piece of shielded cable. This step is necessary to make the microphone mobile. Both the signal generator and the microphone need to get attached to a stand (for example made out of a ruler), with the microphone being fixed to the moving part of the scanner. They need to be placed at the same height. The circuit draw and the schematic of the wiring of the hardware is shown in Figure. 5 and 6.

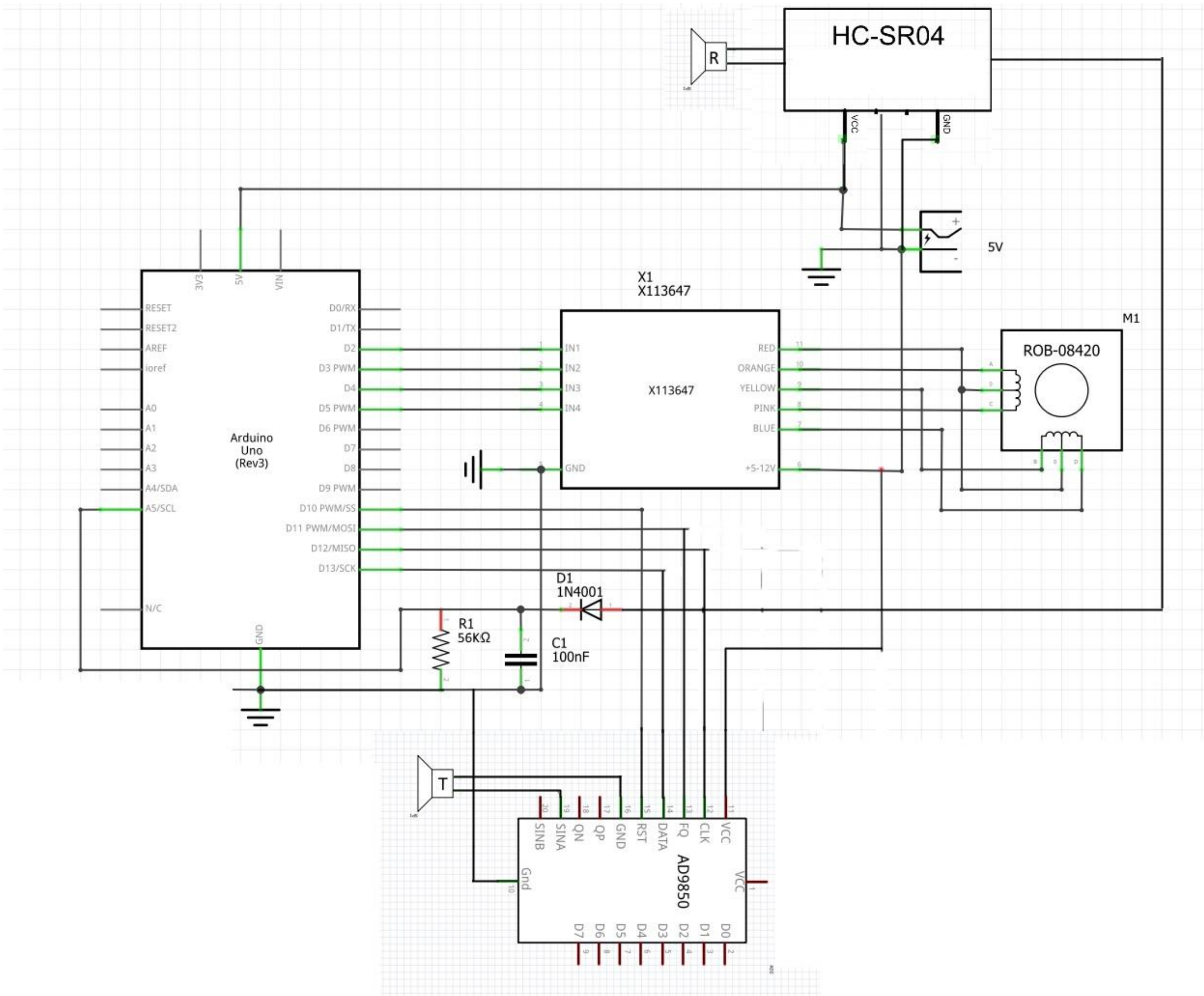

Figure 5 The circuit draw of the hardware

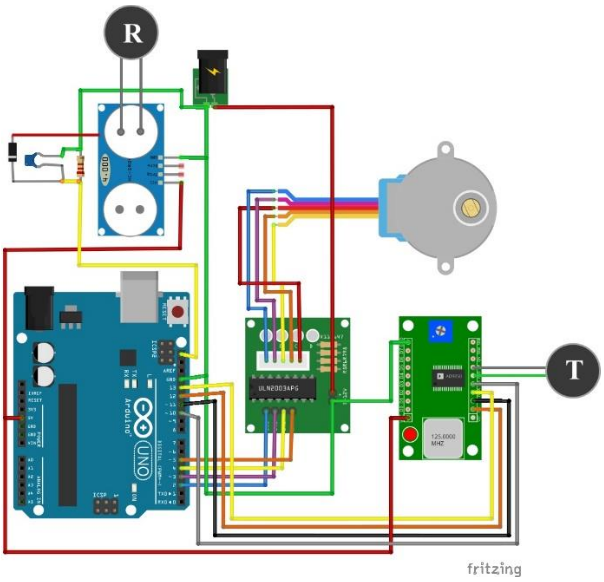

Figure 6 The schematic of the wiring of the hardware

3. The operational principle of the device

Before presenting the code, the operational steps of the apparatus need to be discussed. On switching on the Arduino, the program initializes the important variables and starts operating the hardware. At the beginning of a new measurement it is necessary to record the voltage level measured on the output of the demodulator card corresponding to a zero sound intensity. So, the signal given to the output of the sound generator is switched off, then the signal level of ‘silence’ is measured and stored. After recording this value called ‘Blank’, the sound of the wave source is switched on. Afterwards, moving of the microphone through the wave-front. Sound intensity is measured over and over after specified step intervals. The value of ‘Blank’ is then subtracted and the result is sent to the COM port used to communicate between the Arduino and the PC. The number of steps can be used to calculate when the microphone reaches the end of its course. Then, it is moved back to its original place at the highest possible pace. The microphone will wait for the initiation of a new scanning process in this position.

The program and the detailed explanation of the Arduino-code can be found in the appendix.

4. Collecting measurement data, and results

Measurement data sent to the COM port are collected in an Excel spreadsheet using a Microsoft Data Streamer [13] unit. In the spreadsheet intensity values are plotted against horizontal coordinates. The latter values are obtained by converting step numbers. As the ultrasonic receiver travels 20 cm for every 1000 steps of the motor, one step corresponds to 0.02 cm. Thus performing the measurement in five-step intervals provides a resolution of 1 mm. Figure 7. shows the results of a measurement performed with the apparatus. In this case the obstacle was placed L = 67 mm away perpendicularly to the middle of the path of the receiver. Then, an obstacle of 16 cm height and 14.1 mm diameter was placed into the wave-space. Results are shown in the blue curve. The distance of two adjacent peaks are x = 46 mm, the wavelength is $\lambda = 8.5$ mm. The diameter of the obstacle can be calculated by the following formula [14]:

$$d = \frac{\lambda}{\sin \alpha},$$

where α can be calculated from the peak distance and the obstacle-detector distance:

$$\tan \alpha = \frac{x}{L}$$

As the distance of the obstacle and the detector is too short, we can not use the $\tan \alpha = \sin \alpha$ approximation as usual.

The calculated diameter of the obstacle is 15.0 mm, which convincingly demonstrate the applicability of our measurement system.

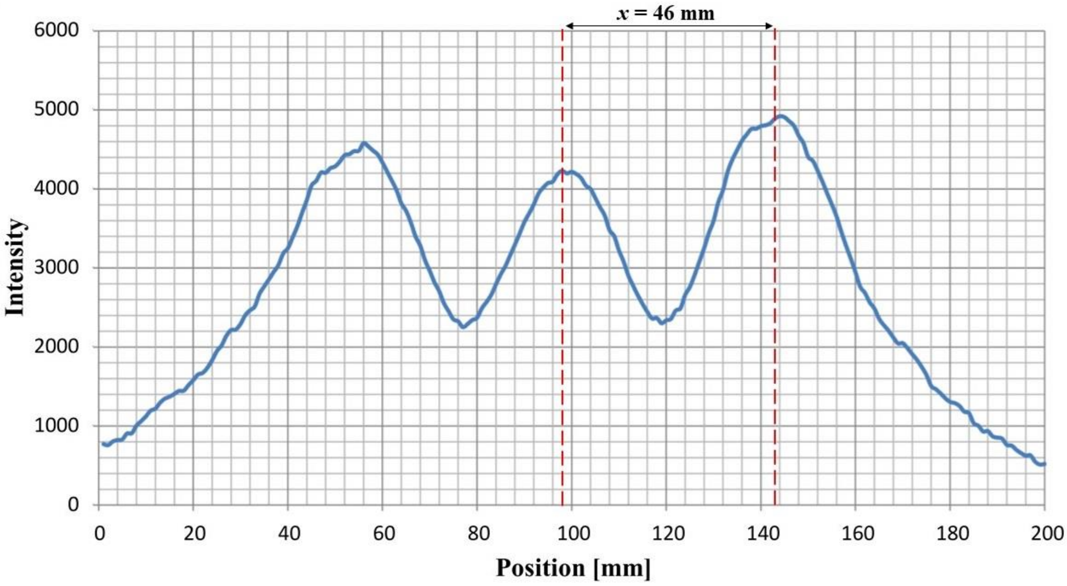

Figure 7 The results of the measurement performed with the apparatus. (The obstacle was 14.1 mm diameter, 16 cm height cylindrycal tube 67 mm away to the scanning path)

Figure 8. shows the diffraction pattern obtained in the original, optical variant of the experiment. A human hair was used as the obstacle resulting in a seemingly similar pattern to the one plotted in Figure 7.

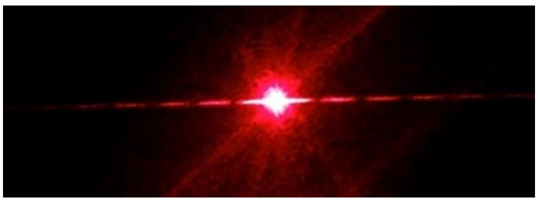

Figure 8 The diffraction pattern of hair obtained in the original, optical variant of the experiment

Figure 9. shows a photo of the complete and functioning experimental setup.

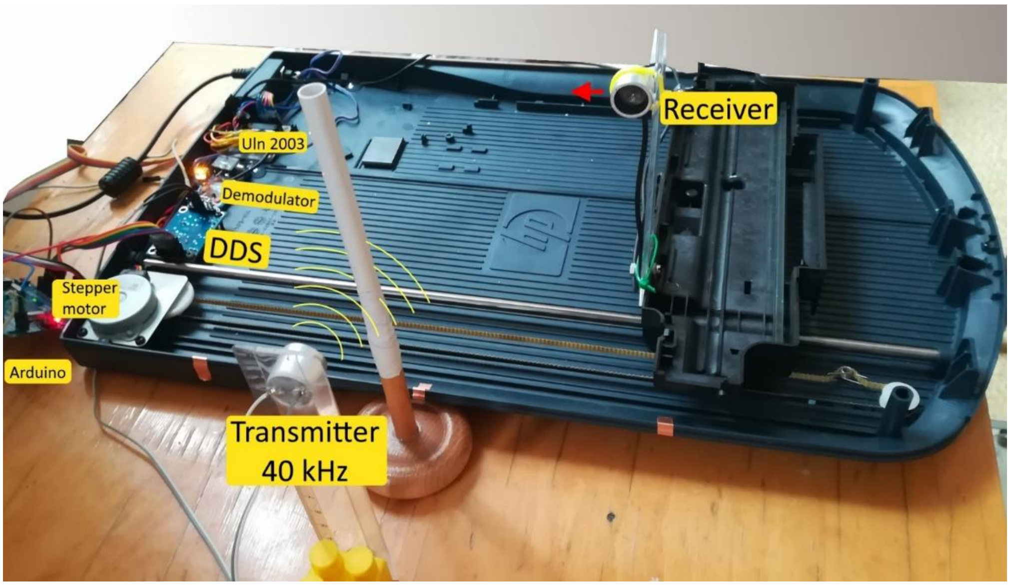

Figure 9 The structure of the automated ultrasonic scanner

5. Further experiments with the device

The automated ultrasonic scanner described in the current paper may also be useful in the conversion of other famous experiments on the interference of light to their sound wave analogies. These can be than simply and spectacularly presented to students. For example using a double slit placed appropriately with the slits being 1 cm away from each other the double slit experiment of Young can be presented convincingly. Placing more slits next to each other evenly would afford a similar diffraction pattern to that of an optical grid. The Lloyd’s mirror experimnet can also be presented by the experimental setup with small alterations. Basic wave-phenomenon may also be studied with it, like refraction and reflection.

6. Research results

We have already put the device to the touch of 30 students in a few experiments during physics lessons, and made a questionnaire with them about their general physics knowledge and impression about the new experiment. According to the answers, the students showed high interest in the experiments presented by the device. The inquire of the group about physics was medium low, the average scores based on the answers was 3.6 (1 means ‘very interested in physics’,…, 6 means ‘not at all’). In contrast, the points of the answers regarding the presented apparatus implied that it could help students’ involvement in physics. The average score based on the interest in the device was 2.3, which is better than the previous result about the usual attitude toward physics.

Figure 10 shows the correspondence between the physics knowledge of the students and their opinion, how much they understood the experiment and the operational principle of the device. However, the average score of physics knowledge was regrettably high, 3.9 (1 means ‘very good at physics’,…, 6 means ‘not good at physics’), the understanding score was 2.3, which is much better result.

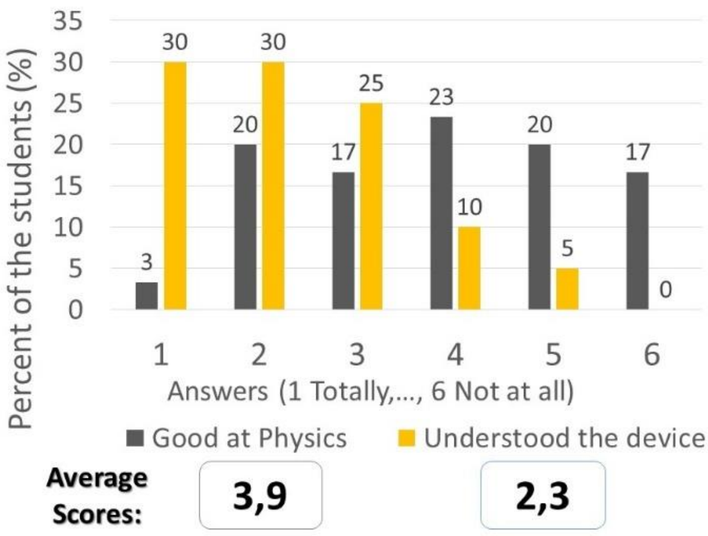

Figure 10 Average score of students based on the answers about usual knowledge of physics and understanding the operation of the device

7. Conclusion

An example on an inexpensive yet robust device is presented for the use in student experiments in physics-classes. The Arduino-automated interferometer is based on repurposed computer-peripheries and is used in the larger-scale demonstration of wave-interference experiments such as the diffraction of waves around an obstacle or the well-known two-slit experiment. Because of using ultrasound instead of light (commonly used in current educational experiments) the objects having a comparable size to the wavelength are larger, making the experiments less sensitive and also more easily perceptible. As programming of the Arduino and the collection and evaluation of the experimental data can be done also by students, a connection between multiple subjects can be established, supporting interdisciplinary education. According to the results of a questionnaire involving 30 students, the device greatly increased their interest in science and modern technologies and improve their attitude about learning physics.

8. Acknowledgement

The research financed by the Content Pedagogy Research Program of the Hungarian Academy of Sciences and HAS Research Group of Technical Informatics Methodology.

8. References

[1] J. Batista Mota *et al.*, “Construction of a low-cost mobile embedded system for computer numerical control and educational purpose,” *Int. J. Innov. Educ. Res.*, vol. 7, no. 11, pp. 1235–1247, Nov. 2019, doi: 10.31686/ijier.vol7.iss11.1996.

[2] F. Bouquet, J. Bobroff, M. Fuchs-Gallezot, and L. Maurines, “Project-based physics labs using low-

cost open-source hardware," *Am. J. Phys.*, vol. 85, no. 3, pp. 216–222, Jan. 2016, doi: 10.1119/1.4972043.

[3] M. Y. Stoychitch, "Generate stepper motor linear speed profile in real time," in *IOP Conference Series: Materials Science and Engineering*, 2018, vol. 294, no. 1, doi: 10.1088/1757-899X/294/1/012055.

[4] G. Organtini, "Arduino as a tool for physics experiments," in *Journal of Physics: Conference Series*, 2018, vol. 1076, no. 1, doi: 10.1088/1742-6596/1076/1/012026.

[5] M. Se-Yuen, "Wave experiments using low-cost 40 kHz ultrasonic transducers," *Phys. Educ.*, vol. 38, no. 5, pp. 441–446, 2003, doi: 10.1088/0031-9120/38/5/310.

[6] Ebay, "HANPOSE HPV4 Linear Guide Set Openbuilds Mini V Linear Actuator 100-500mm | eBay." https://www.ebay.com/itm/HANPOSE-HPV4-Linear-Guide-Set-Openbuilds-Mini-V-Linear-Actuator-100-500mm/233254401492?hash=item364f0bb5d4:m:mFs3cp6PXL-CFsjOL6KB4Uw (accessed May 23, 2020).

[7] Micro Robotics, "ULN2003 Stepper Driver - Micro Robotics." https://www.robotics.org.za/ULN2003-MOD (accessed May 23, 2020).

[8] Electronicos Caldas, "4 Phase ULN2003 Stepper Motor Driver PCB." https://www.electronicoscaldas.com/datasheet/ULN2003A-PCB.pdf (accessed May 23, 2020).

[9] ND6T, "SIGNAL GENERATOR," no. (MAY 29, 1975), 1975, Accessed: May 23, 2020. [Online]. Available: http://www.nd6t.com/test/Easy Signal Generator.htm.

[10] Analog Devices, "CMOS, 125 MHz Complete DDS Synthesizer." https://www.analog.com/media/en/technical-documentation/data-sheets/ad9850.pdf (accessed May 23, 2020).

[11] Makerfabs, "HC-SR04 Ultrasonic Range Measurement Module." https://www.makerfabs.com/hc-sr04-ultrasonic-range-measurement-module.html (accessed May 23, 2020).

[12] PC Services, "HC-SR04 circuit." http://www.pcserviceselectronics.co.uk/arduino/Ultrasonic/HC-SR04-cct.pdf (accessed May 23, 2020).

[13] Microsoft, "Data-Streamer - Office Support." https://support.office.com/en-us/article/data-streamer-c90aebcf-3d44-47ab-a068-549a0b9edfc6 (accessed May 23, 2020).

[14] LaserClassroom, “Measure Human Hair with a LASER!” https://edu.workbencheducation.com/cwists/preview/6199-measure-human-hair-with-a-laserx (accessed May 23, 2020).

Appendix

In this table you can find the Arduino code of the device with explanation. I you feel like rebuilding the device, fell free to use the code in the 2nd column.

	Arduino code	Explanation
1	//**********Stepper motor	
2	int STP1 = 2;//Wiring for stepper motor (blue)	The program (line 2.-5.) defines the digital outputs to send data to the stepper motor. E.g.: **int STP2 = 3** means, the 2nd wire (purple) of the stepper motor is connected to the D3 output of the Arduino (ULN2003 driver board was inserted)
3	int STP2 = 3;//Wiring for stepper motor (purple)	
4	int STP3 = 4;//Wiring for stepper motor (yellow)	
5	int STP4 = 5;//Wiring for stepper motor (orange)	
6	int Speed = 2;//The speed of the stepper motor (1=high, 20=slow)	With the variable “Speed” we can identify the time between two steps of the stepper motor.
7	int N=1000;//The number of steps	In variable N we can store the number of all of the steps along the 20 cm long path.
8	//**********AD9850 DDS signal card	
9	int CLK = 12; // pin 12: AD9850 clock pin	In line 9. -12. We can declare the outputs to control the AD 9850 Analog Signal Generator [9], [10].
10	int FQ = 11; // pin 11: freq update pin	
11	int DAT = 13; // pin 13: D7 DATA pin	
12	int RST = 10; // pin 10: RESET	
13	//**********	
14	String C="";//Start command	Declaration of the variable to start the program.
15	int LEDPin = 13;//Indicator lamp	The indicator lamp shows us, when a program is running on the Arduino.
16	int Ain_Pin = A5;//ADC input	
17	int Act_Step=0;//Current number of steps	
18	int Blank=0;//Demodulator voltage without signal	
19	int U_in=0;//Demodulator voltage	
20	int Nm;//Number of measurements	

21	int Ns;//Number of steps without measure	
22	void setup() {	
23	pinMode(FQ, OUTPUT);	Program in line 23.-31 is running once during every measurement. We can setup which pins can be used as an output. E.g.: **pinMode(STP2, OUTPUT).**
24	pinMode(CLK, OUTPUT);	
25	pinMode(DAT, OUTPUT);	
26	pinMode(RST, OUTPUT);	
27	pinMode(STP1, OUTPUT);	
28	pinMode(STP2, OUTPUT);	
29	pinMode(STP3, OUTPUT);	
30	pinMode(STP4, OUTPUT);	
31	pinMode(LEDPin, OUTPUT);	
32	Nm=60;	We can specify the important constants. **Nm=60** is the number of act of measuring. **Ns=5** means that we do measurement in every 5^{th} steps.
33	Ns=5;	
34	DDS_pulse(RST);	In line 34.-36. we set the Analog Signal Generator to default settings [10].
35	DDS_pulse(CLK);	
36	DDS_pulse(FQ);	
37	analogReference(EXTERNAL);//	Between 0V and 5V we can use the voltage AREF pin of Arduino as a voltage reference which make the measurement more accurate.
38	Blank=0;//Measuring a blank signal	Calling subroutin **meas_b()** (line 90. – 98.) we can measure the level of the signal when the transmitter is tuneless.
39	meas_b();	
40	Serial.begin(9600); // Serial comm setup	Than we can set the boud_rate value of the serial communication.
41	DDS_Output (40000);//Ultrasound signal turns on 40 kHz	
42	C="S";//Ready for first scan	
43	}	
44		
45		
46	//**********Start scan	
47	void loop() {	
48	if (Serial.available() > 0) {	
49	String cln=Serial.readString() ;//Waiting a command from Excel ("S" command = start scanning)	To start the measurement, write an “S” the appointed cell of the Excel.
50	cln.trim();	
51	cln=cln.substring(0,1);	

52	if (cln=="S"){	The scanning process will start as soon as the starter command ("S") arrived.
53	meas_b();//Measuring a blank signal	
54	C="S"; //Ready for a new scan	
55	DDS_Output (40000);//Ultrasound signal turns on 40 kHz	
56	}	
57	}	
58	if (C=="S"){	
59	Act_Step=0;	The program in line 59 resets the actual stepnumber.
60	digitalWrite(LEDPin, HIGH);	The indicator lamp flares up.
61	for (int I = 0; I < N; I++) {	The cycle will be repeated until the stepnumber reaches N (= 1000).
62	Forward();//Just one step forward	
63	U_in =0;	Reset the previous value of voltage.
64	Act_Step++;	The program adds one to the actual stepnumber, than decides, whether it is necessary to do a measurement. If the stepnumber is divisible by Ns (5 in this case), a measuring is done (else, it is not). This is useful to decrease measurement time. The program collects datas 60 times in a single position.
65	int x=0;	
66	x = Act_Step % Ns;//Modulo	
67	if(x==0){	
68	for (int J = 0; J < Nm; J++) {	
69	U_in = U_in + analogRead(Ain_Pin);//Step and measure	Save the new value of voltage.
70	}	(The speed of steps can be affected by the **delay(Speed)** function as well.)
71	U_in = U_in - Blank;	After measuring the voltage in Ain_Pin, we have to substract the value of Blank voltage to eliminate the background noise.
72	String ln="";	We clear variable **String ln="".**
73	ln=ln + String(Act_Step) + "," + String(U_in);//Print a line for Excel	We concatenate the measured values and the number of step to this string variable.
74	Serial.println(ln);	In line 72. the program sends the value of the previous string to the com port
75	}	Data was sent by Arduino with the command **Serial.println(ln)** are collected by the Excel by the help of the free Data Streamer module [13] The data separated by comma are automatically placed in coulmns.
76	}	
77	STP_off();	The program turns off the stepper motor and the indictor

78	delay(100);	lamp, than moves the microphone back to the original position by calling the right subroutine. Then, the device turns off permanently.
79	digitalWrite(LEDPin, LOW);	
80	for (int I = 0; I < N; I++) {	
81	Backwards();	
82	}	
83	}	
84	C="X";	
85	STP_off();	
86	}	
87		
88		
89	//**********	
90	void meas_b() {	It is measuring the intensity of the background noise.
91	DDS_Output (40);//Ultrasound signal turns off 40 Hz	We send 40 Hz signal to the output of the signal generator. In case of our microphone – which is optimised to 40 000 Hz – this sound is practically acts like silence.
92	U_in =0;	We reset the value of voltage.
93	delay(500);	
94	for (int J = 0; J < Nm; J++) {	We measure and read the value of the input voltage. We repeat this measurement step Nm=60 times like in case of active measuring.
95	U_in = U_in + analogRead(Ain_Pin);//Average of demodulator voltage without signal	
96	}	
97	Blank=U_in;	We store the value of background noise intensity in the variable Blank.
98	}	
99	//**********	
100	void DDS_pulse(int pin) { digitalWrite(pin, HIGH); digitalWrite(pin, LOW); }	
101		
102	void DDS_Output(float frequency) {	
103	long freq = frequency * pow (2,32) /125000000; // Reference clock	
104	for (int b=0; b<4; b++, freq >>= 8) shiftOut(DAT,CLK,LSBFIRST,freq);	
105	shiftOut(DAT,CLK,LSBFIRST,0);	
106	DDS_pulse(FQ);	

107	}	
108	//**********	
109	void Forward () { //stepper motor one step forward	The code of the stepper motor: https://coeleveld.com/arduino-stepper-uln2003a/ (Example 3, 4-step sequence), [7]
110	{digitalWrite(STP1, HIGH);digitalWrite(STP2, HIGH);digitalWrite(STP3, LOW);digitalWrite(STP4, LOW);delay(Speed);}	
111	{digitalWrite(STP1, LOW);digitalWrite(STP2, HIGH);digitalWrite(STP3, HIGH);digitalWrite(STP4, LOW);delay(Speed);}	
112	{digitalWrite(STP1, LOW);digitalWrite(STP2, LOW);digitalWrite(STP3, HIGH);digitalWrite(STP4, HIGH);delay(Speed);}	
113	{digitalWrite(STP1, HIGH);digitalWrite(STP2, LOW);digitalWrite(STP3, LOW);digitalWrite(STP4, HIGH);delay(Speed);}	
114	}	
115	void Backwards () { //Stepper motor one step backwards	The code of the stepper motor to reset the original position. The steps are the same in case of stepping forward, but in reverse order. Moving backwards can be fast, thus we can decrease the time between steps (Speed).
116	{digitalWrite(STP1, HIGH);digitalWrite(STP2, LOW);digitalWrite(STP3, LOW);digitalWrite(STP4, HIGH);delay(Speed);}	
117	{digitalWrite(STP1, LOW);digitalWrite(STP2, LOW);digitalWrite(STP3,	

	HIGH);digitalWrite(STP4, HIGH);delay(Speed);}	
118	{digitalWrite(STP1, LOW);digitalWrite(STP2, HIGH);digitalWrite(STP3, HIGH);digitalWrite(STP4, LOW);delay(Speed);}	
119	{digitalWrite(STP1, HIGH);digitalWrite(STP2, HIGH);digitalWrite(STP3, LOW);digitalWrite(STP4, LOW);delay(Speed);}	
120	}	
121	void STP_off() { //stepper motor turn off	
122	{digitalWrite(STP1, LOW);digitalWrite(STP2, LOW);digitalWrite(STP3, LOW);digitalWrite(STP4, LOW);delay(Speed);}	
123	}	