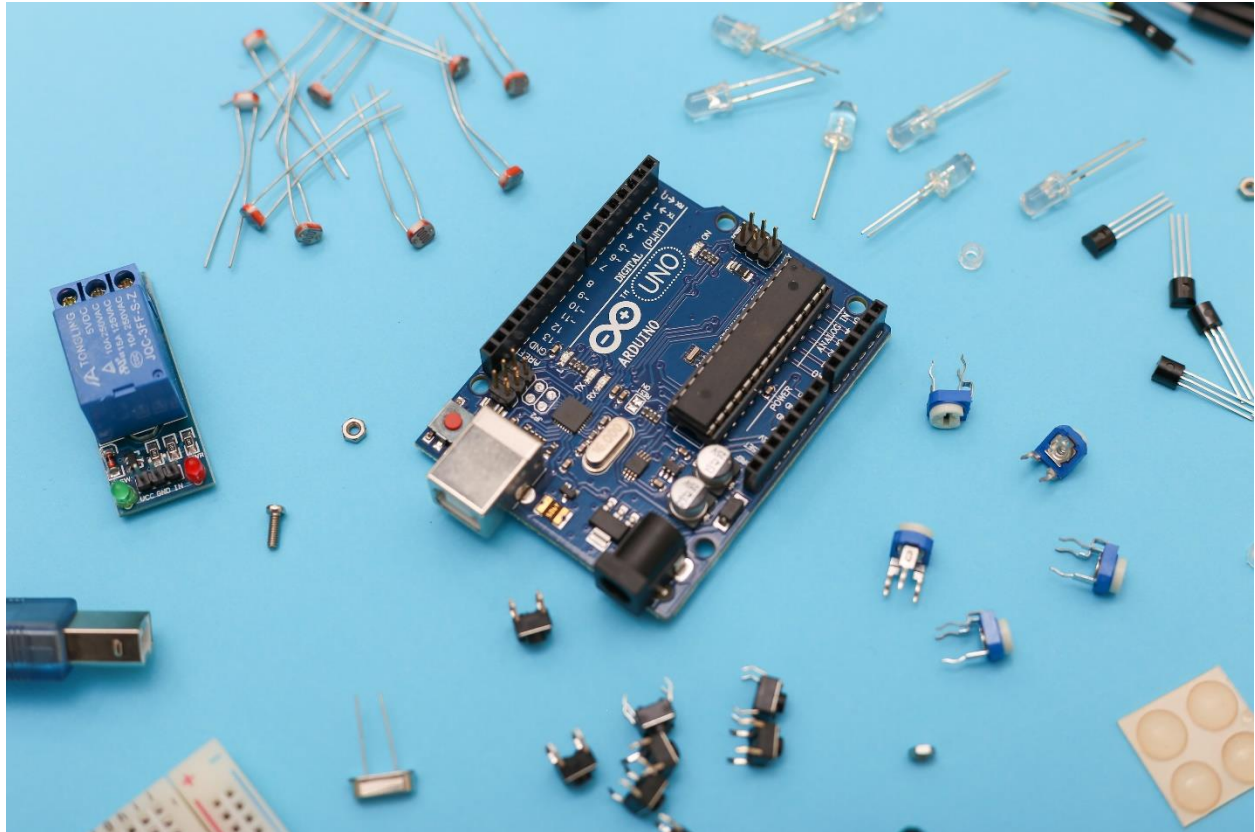


Kinematics exercises with Arduino

Measurement and graphics exercises for preparatory physics lessons



Dorottya Schnider and Mihály Hömöstre

2023

Kinematics exercises with Arduino

Measurement and graphics exercises for preparatory physics lessons

Authors: Dorottya Schnider and Mihály Hömöstre

Produced with the technical support of the Doctoral Student Scholarship Programme of the Cooperative Doctoral Programme of the Ministry of Innovation and Technology, funded by the National Research, Development and Innovation Fund.

The preparation of the curriculum was supported by the Public Education Development Research Programme of the Hungarian Academy of Sciences (project no.: SZKF-7/2022).

Source of the image on the cover: <https://unsplash.com/photos/gavODTHG36Y>, taken by: Sahand Babali

Note to colleagues

Some exercises rely on prior knowledge, and measurement errors can affect the outcome of a measurement. It is recommended that you teach your students the basic concepts through experiments (e.g., processing a computer simulation) or examples that lead to precise results. In the practice phase, the exercises provided by us support the deepening of the acquired knowledge, as well as the development of thinking and competence.

The learning material presented here can be used as a workbook to support practice and, after an introductory experiment to familiarise students with the device and the program code, each measurement/analysis exercise can be published as a stand-alone assignment.

It is recommended that the circuitry is assembled by the students - thus developing their psychomotor skills - and the code is either pre-loaded onto the Arduino or made easily available to the students. The focus is on developing the skills and mindset related to physics, and teaching programming skills is not our primary objective.

The authors

Table of Contents

1. Introduction.....	4
1.1. A brief overview of Arduino	4
1.2. Markings	6
2. Exercises with an ultrasonic distance measuring sensor.....	7
2.1. Preparatory questions and exercises	7
2.2. Arduino exercises - experiments and measurements: distance, uniform motion.....	8
2.2.1. How far is it?.....	8
2.2.2. Distance-time and displacement-time.....	9
2.2.3. The speed of Thomas the Tank Engine	11
2.2.4. The speed of moving.....	15
2.3. Arduino exercises - Experiments and measurements – Varying motion.....	16
2.3.1. Introductory questions	16
2.3.2. Examining variable motion - graph analysis	17
2.3.3. Study of variable motion - data collection and analysis	18
2.3.4. Uniformly varying motion	19
2.3.5. Movement on a slope	22
2.3.6. Freefall	23
2.3.7. *** Bonus measurement:	25
2.3.8. Varying motion - practice exercise	26
2.4. Arduino exercises - experiments and measurements: periodic phenomena.....	28
2.4.1. List other periodic phenomena!	28
2.4.2. Examine the motion of the body vibrating on the spring.....	28
3. Summarising measurement exercise with photoresistors - Exoplanet exploration in the classroom	32
Summary	35
Appendix – Tutorial on how to use the Excel Data Streamer.....	37

1. Introduction

In order to solve the following exercises, you are going to need an Arduino, a laptop, a USB cable and an ultrasonic distance sensor. To carry out the experiments and measurements, you will have to build a circuit in which the sensor is connected to the Arduino and the Arduino is connected to the laptop via a USB cable. The code to operate the ultrasonic rangefinder and read out the data it sends at intervals will be written within the Arduino software installed on the laptop.

1.1. A brief overview of Arduino ¹

Arduino is a platform for electronic projects, consisting of a programmable computer (microcontroller) and a software (IDE = Integrated Development Environment). The software is where you write the code (in a simplified language similar to C++) that is uploaded to the device, which is then used to activate the sensors connected to the Arduino and monitor the time state of the sensors.

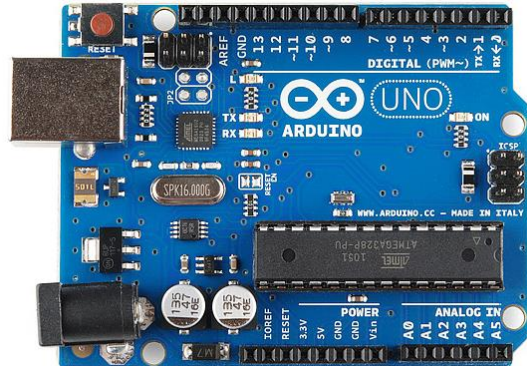


Fig. 1: Arduino board – microcontroller

Source: <https://learn.sparkfun.com/tutorials/what-is-an-arduino/all>

Power supply: The Arduino card is powered by the computer, which is connected to it via a USB cable. Also, the code can only be loaded using a USB connection.

Pins, pin configuration

On the Arduino, there are pins marked with different numbers to which sensors can be connected. This way, we can refer to a particular pin in the program code by the number we want to use to operate that particular sensor.

- Digital outputs: From 0 to 13
- Analogue outputs: From A0 to A5-

For example, analogue connectors can read the signal from an analogue sensor (e.g., temperature sensor, photo resistor) and convert it into a digital signal.

The digital outputs can be used both as digital inputs (to receive a signal - e.g., press a button) and outputs (to output a signal - e.g., flash an LED).

In addition, the pins marked GND and 5V/3V can also be found on the microcontroller.

The GND (ground) provides the grounding, the 5V/3V provides the voltage.

¹ Source: <https://learn.sparkfun.com/tutorials/what-is-an-arduino/all>

Breadboard (panel)²

The electrical circuitry can be built on the breadboard and powered by either a battery or an external power supply, sometimes connected to the Arduino (connected to the laptop via USB cable). The + and - holes indicate the outputs of the circuit. These two pins are used for power connections. The holes in the middle of the breadboard are used to connect electrical components. These holes are vertically aligned in rows, with 5 holes per column, which means that up to 5 devices can be connected per section. The connection of the holes is illustrated by the purple lines drawn in the diagram.

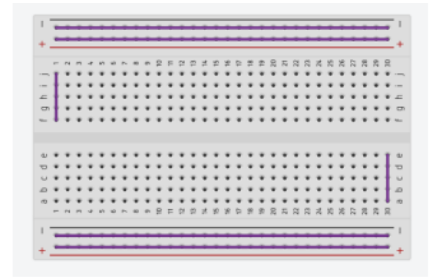


Fig. 2: Breadboard. Source: own editing, Tinkercad. <https://www.tinkercad.com/dashboard>

The software

The Arduino and the sensors connected to it can be operated by entering the appropriate programs (codes)³. The code is written in the software that can be [downloaded for free](#).

The assembled connection is connected to the laptop with a USB cable, the software is opened and the following settings are performed. Select the appropriate motherboard, i.e. the version of the Arduino you are using (found on the device), e.g.

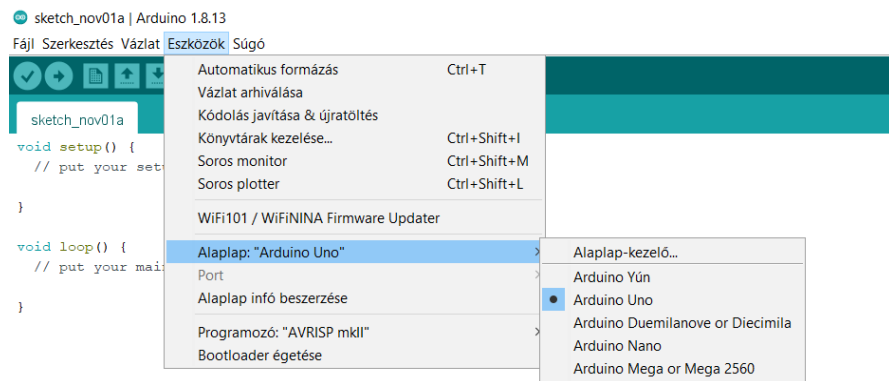


Fig. 3: The default settings of the Arduino software.

UNO or Arduino Uno. Next, the port is configured (click on Ar, next to which the correct Arduino version is indicated in brackets).

After configuration, the code is written - for the present exercises a predefined code is used - and can be checked with the tick. Click on the right arrow next to the tick to upload the code to the motherboard. By opening the small magnifying glass icon (row monitor) in the top right corner, the values read by the sensor are displayed.



Fig. 4: The serial monitor.

Important notice: Arduino is used as a measuring tool for physics lessons, and can be used to carry out a wide range of experiments. The focus is on making and evaluating measurements, no programming skills are required.

² Source: <https://www.seecstudio.com/blog/2020/01/06/how-to-use-a-breadboard-wiring-circuit-and-arduino-interfacing/>

³ More projects and codes are available for example here: <https://funduino.de/> or <https://create.arduino.cc/projecthub>

1.2. Markings

Differentiation is used in the Workbook. The problems the solution of which we recommend to be compulsory are printed in normal format, the more abstract problems are marked with***.

2. Exercises with an ultrasonic distance measuring sensor

For distance measurement, an ultrasonic distance measuring sensor is used. The sensor measures the time between the emitted signal and the return signal and then calculates a distance based on the speed of sound.

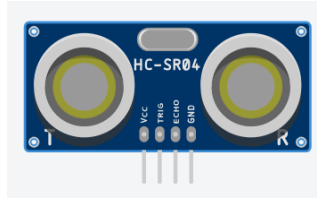


Fig. 5: Ultrasonic distance measuring sensor.
Source: Tinkercad.

2.1. Preparatory questions and exercises

1. How to determine the distance from the sensor to the object being inspected?

- a) The distance between the sensor and the body is Δx , the time from the emission of the signal to its return is Δt , and the speed of sound is constant: $c_h = 340 \frac{m}{s}$.

- b) Describe the principle of measurement in your own words in at least 3 sentences.

- c) What is the relationship between the distance between the sensor and the object? (Write it down in a mathematical form.)

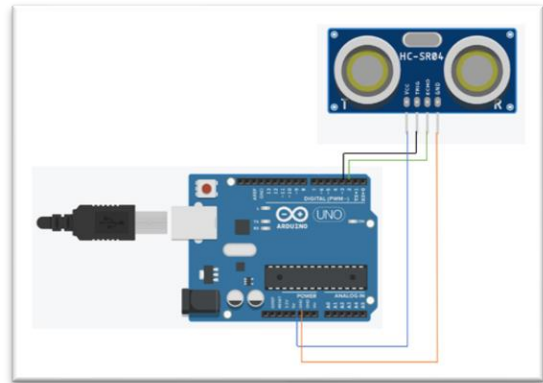
2.2. Arduino exercises - experiments and measurements: distance, uniform motion

2.2.1. How far is it?

Programme an ultrasonic distance measuring sensor for measurement!

Tools

- Arduino + USB cable + laptop
- 4 cables
- Ultrasonic distance measuring apparatus
- Objects to be measured (e.g., phone, pen holder)



Prepare the wiring as shown in the figure: connect the orange cable to the Arduino's "Ground/GND" pin (connector), the echo (green cable) to pin no. 2, the trigger (black cable) to pin no. 3, the "blue" half of the sensor (VCC) to a permanent 5 V connector.

Measure the distance to your phone, pen holder, etc. Write the measured values on the serial monitor. (*Reminder: You can have the serial monitor displayed by clicking on the small magnifying glass icon.*)

In order to solve this exercise, use the following code! The code is also available [here](#) under "Distance measurement with an ultrasonic distance measuring device - the base code".

```
const int trigPin = 3;    // pin 3 is connected to the trigger of the distance
                          // measuring sensor, defined as trigPin
const int echoPin = 2;   // pin 2 is connected to the echo of the distance measuring
                          // sensor, define it as echoPin
long duration;           // variable value, time of sound propagation
float distance;         // the measured distance

void setup() {           // Default settings of the software

  Serial.begin(9600);    // Start of serial communication, max data rate: 9600 bps

  pinMode(trigPin, OUTPUT); // signal output on the trigPin
  pinMode(echoPin, INPUT);  // signal reception on the echoPin
}

void loop() {           // this is where the loop starts

  digitalWrite(trigPin, HIGH); // the trigger is switched on
  delay(500);                  // emits a signal for a given period of time
  // (millisec)
  digitalWrite(trigPin, LOW); // the trigger is switched off after a given period
  // of time
  duration = pulseIn(echoPin, HIGH); // the time it takes for the signal to
  // return
  distance = duration * 0.0340 / 2; // the measured distance

  Serial.println(distance); // the measured values are printed below each other
}
```

2.2.2. Distance-time and displacement-time Tools

In order to solve this exercise, the experimental configuration and code used in the previous exercise is going to be used. This time the data collected by measurement should be sorted and evaluated in Excel, and the [Excel Data Streamer](#) extension should be used to send the data to Excel simultaneously with the measurement. See the *Appendix* for steps on how to use Data Streamer.

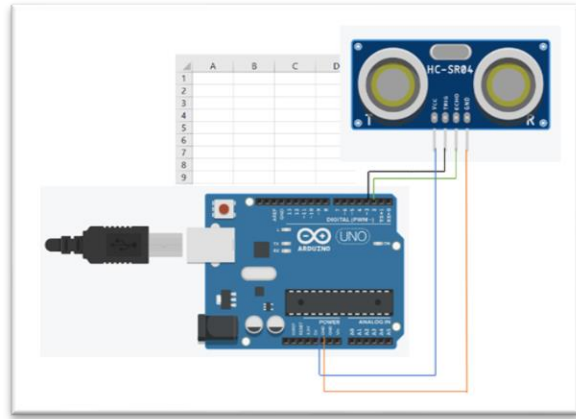


Fig. 7: The instruments used for measurement.
Source: own editing, Tinkercad.

The description of the exercise

Upload the code to your Arduino, connect to the Excel Data Streamer and start scanning the data. Place your phone (pen holder, etc.) in front of the distance measuring device and start recording the data in Excel. At about the same time, start moving the object away from the sensor in a uniform motion, stop it for a moment, and then move it back towards the sensor, also at a constant speed, stopping about 10 cm in front of the sensor. Stop the data recording, save the data file with the .csv extension, then open it and answer the questions based on the measurement data. *Note: The distance value read by the Data Streamer is the distance of the object from the sensor, i.e., its momentary position. The sensor sends the signal at a given interval (every 500 ms according to our code), therefore the same amount of time elapses between any two distance values received.*

Questions and exercises

1. Organise your data in Excel in the following way.

Remember, sometimes data recording could already have been in progress when you started recording, so feel free to delete the initial data (which is irrelevant for the analysis).

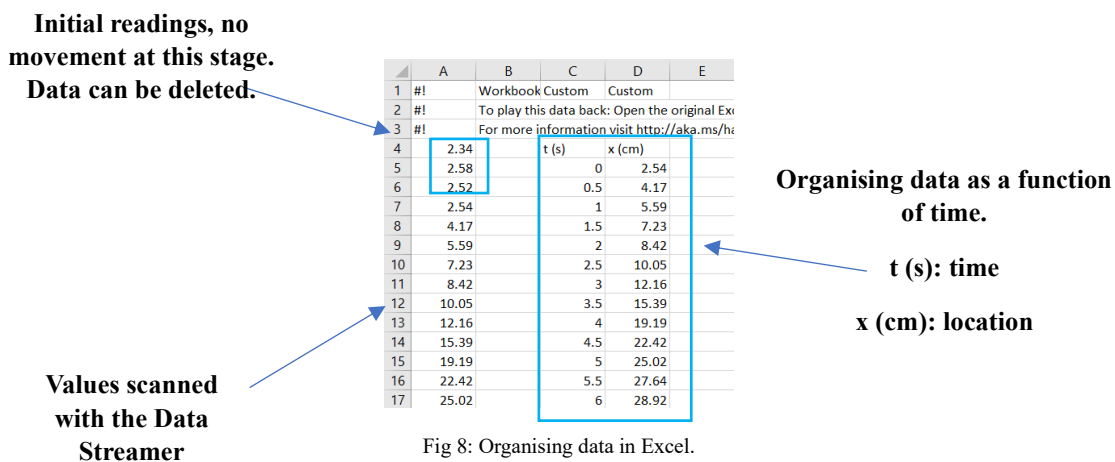


Fig 8: Organising data in Excel.
Source: own editing, Microsoft Excel.

2. Determine the distance s covered by the phone. Show in detail the counting process.

3. Determine the distance the phone has travelled between the start and end points of the test.

4. Plot the data on a place-time graph in Excel.

2.2.3. The speed of Thomas the Tank Engine

Use an ultrasonic distance measuring sensor to investigate and characterise Thomas's movement.

Tools

- Thomas the Tank Engine
- Arduino + USB cable + laptop
- 4 cables
- Ultrasonic distance measuring sensor
- Excel Data Streamer

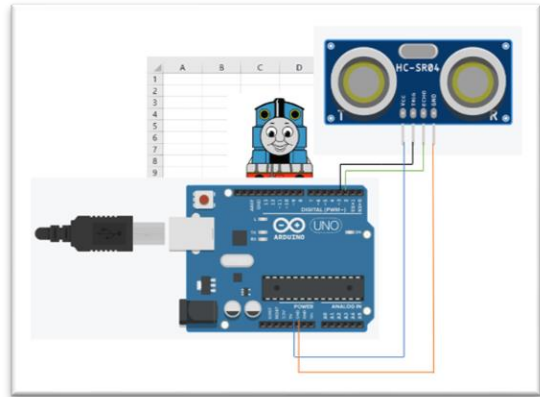


Fig. 9: Instruments used for measurement.
Source: own editing, Tinkercad. Thomas image source: <https://clipart-library.com/thomas-cliparts.html>

Description of the exercise

In order to check Thomas's movement, use the code provided in exercise 2.2.1 and the assembled circuit. Connect the Arduino to the laptop and load the code. You can use the *Data Streamer* described in Exercise 2.2.2 to track the data sent by the sensor. Switch on Thomas, then place it directly in front of the ultrasonic distance measuring sensor, release it, and then stop it at a distance of about 1 m 20 cm from the sensor.

Questions and exercises

The sensor sends a signal at a given interval (every 500 ms according to our code), which means that the same time elapses between any two distance values received.

It is important that you always consider the distance travelled as a function of the time elapsed from the beginning of the movement. Thomas had already been moving at the start of the test, which means that the starting moment $t=0$ is accompanied by a distance x of some distance from the sensor.

Open the saved .csv file and then tabulate the data according to the sample, first digitally in Excel. Use the following notations:

- t**: time elapsed since the start of the movement/actual time instant
- x**: the distance from the starting point (distance travelled in a given time, measured from the starting point) at the current time instant
- s**: distance travelled measured from the start of the movement

$$s = x_{actual} - x_{initial}$$

(Remember: the data plotted in Excel show the distance from the sensor, and all measurements are acquired after a delay of 500 ms.)

You can determine the distance data by subtracting the distance measured at the starting point from the actual distance travelled.

Initial scanned values, no movement at this time. Data can be deleted.

	A	B	C	D	E
1	#!	Workbook Custom	Custom		
2	#!	To play this data back: Open the original Excel w			
3	#!	For more information visit http://aka.ms/hackin			
4		4.46			
5		4.55	t (s)	x (cm)	s (cm)
6		4.57	0	4.57	0
7		15.32	0.5	15.32	10.75
8		29.5	1	29.5	24.93
9		39.64	1.5	39.64	35.07
10		50.15	2	50.15	45.58
11		61.71	2.5	61.71	57.14
12		73.03			
13		83.91			
14		95.54			
15		106.73			
16		117.3			

The values read by the Data Streamer.

The data organised by us - summarised in a table.

Fig. 10: Organising data in Excel.
Source: own editing, Microsoft Excel.

2. Fill in the table according to your measurement data. In a given time from the start of the movement, what distance has Thomas travelled?

t (s)	0.5	1	1.5	2	2.5
s (cm)					

3. Using the recorded data, characterise Thomas' movement. Explain how the distance measured from the sensor varies over time.

4. Fill in the table.

Δt : the time elapsed between the receipt of two pieces of data, the duration of a given section of movement. Δt

Δs : distance travelled in Δt

$\Delta s = s_{final} - s_{initial}$ (in the section under review)

t (s)	0	0.5	1	1.5	2	2.5
Δt (s)	-	0.5	0.5	0.5	0.5	0.5
x (cm)						
s (cm)						
Δs (cm)						
$\frac{\Delta s}{\Delta t}$ ($\frac{cm}{s}$)						

5. What can we conclude from the measurement about Thomas' movement?

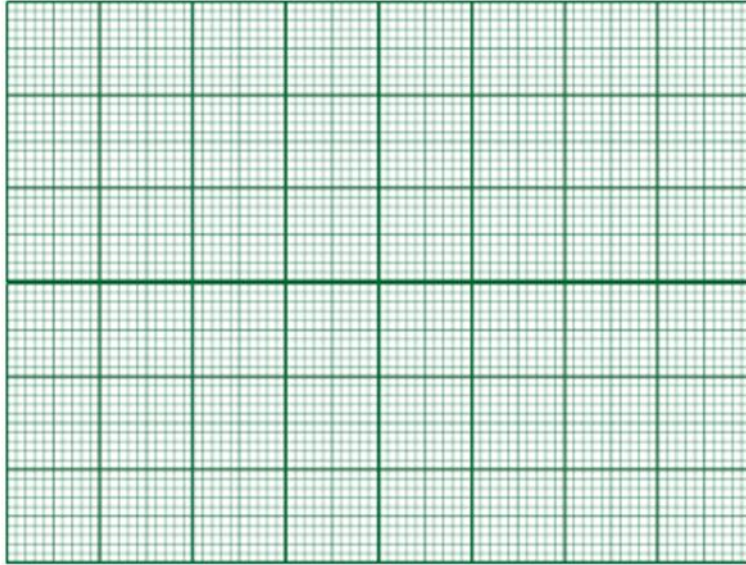
6. Explain any possible differences in $\Delta s/\Delta t$ values.

7. Find the average of the $\Delta s/\Delta t$ values.

8. Express the average speed determined in exercise 4 also in $\frac{m}{s}$ and $\frac{km}{h}$.

9. Stay in Excel and plot the journey data as a function of time (measured minute-by-minute, i.e., from the beginning of the movement) on a graph. For homework, prepare the graph also on graph paper⁴.

⁴ You can download and print your own free graph paper at: <https://www.freeprintablepdf.eu/ru/ru-millimeterpaper-pdf>



10. Select a data point in the graph and right-click on it. Use the "*Add trendline*" menu in Excel to insert a line on the points. Set the "*Set intercept*" to 0, then tick the "*Display equation*" box to display the equation of the line. The coefficient x in the equation gives the velocity of Thomas.

The value calculated by Excel:

11. Answer the questions based on the graph. $\Delta s/\Delta t$)

a) What function is plotted?

b) How does the distance travelled vary as a function of time?

*** Stay with Excel and plot the velocity values ($\Delta s/\Delta t$) as a function of time on a graph.

*** On the basis of the graph, formulate *three true statements*.

*** The area under the speed-time function indicates the distance travelled.

a) What shape is plotted by the velocity-time function?

b) What is the distance travelled?

2.2.4. The speed of moving

Using the previous circuit diagram and code, move your phone steadily (at a constant speed) in front of the ultrasonic distance measuring sensor (move it up to 1 m away from the distance measuring sensor) and then determine the average speed of the movement.

1. Start recording data when your phone is already in motion. Organise the scanned data in a table as in the previous exercise.

2. Draw a distance-time graph. Insert a line onto the data points with the right mouse button, and then display the equation of the line. (Set the y-axis intercept to 0).

3. Using the displayed equation, determine the speed of the movement. Determine the velocity value calculated by Excel also in $\frac{m}{s}$ and $\frac{km}{h}$.

*** Based on the scanned and calculated data, fill in the table below.

$t (s)$	0	0.5	1	1.5	2	2.5	3
$\Delta t (s)$							
$x (cm)$							
$s (cm)$							
$\Delta s (cm)$							
$\frac{\Delta s}{\Delta t} (\frac{cm}{s})$							

*** Use the data to characterise the speed of this movement.

2.3. Arduino exercises - Experiments and measurements – Varying motion

Uniform straight-line motion occurs rarely in nature. Most of the time our movement is variable. In this case, the magnitude and/or direction of the velocity varies. Examples of varying motion include turning, constant speed motion in a circular path, and natural motion.

Consider the following example: a long-distance bus from Nagykanizsa to Budapest arrives in 2 hours and 40 minutes. The distance between the two cities is 210 km. The bus leaves the station, accelerates and then moves at a constant speed for a while. Before the intersection, it slows down, stops, accelerates again, and travels at a constant speed for a while. At the next village it stops again, waits for passengers to get on and off, and so on. Its movement can be divided into many different sections. The speed of the vehicle can vary along each section, so it is useful to introduce the concept of average speed to characterise the sections of movement. The *average speed* can be calculated as the ratio of the length of a given movement section divided by the total time required to complete it.

Average speed: A substitute speed, at which the body, moving at a uniform speed, would cover the same distance in the same amount of time as it would take moving at a variable speed.

$$v_{\text{átlag}} = \frac{s}{t}$$

For the measurements carried out in Chapter 2, Excel was used to calculate the average velocity of the tested objects.

Momentary speed: The speed at which the object is moving at a given moment in time. For example, as indicated by an odometer. v_p

2.3.1. Introductory questions

- What is the average speed of the bus for the whole journey? Please explain your calculations in detail.

- It takes 20 minutes for the bus in the example to reach the neighbouring village of Zalakaros, which is 18 km from Nagykanizsa. What is the average speed on this stretch? Demonstrate your calculation in detail.

2.3.2. Examining variable motion - graph analysis

The following graph was drawn while collecting data with Arduino. A phone was moved in front of the sensor.

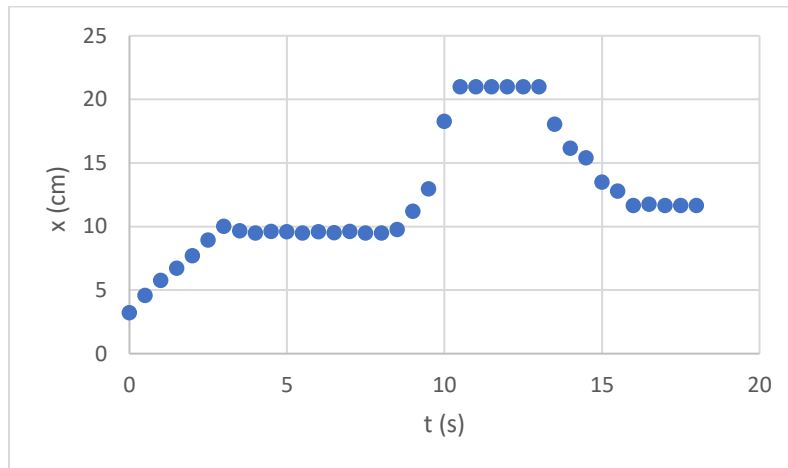


Fig. 11: Analysis of variable motion. Source: own editing.

Questions and exercises

1. Give a name to the graph.
2. How many sections can the movement be divided into?
3. Characterise the movement at the different sections.
4. Using the attached data table, determine the average speed of the phone for each section.

t(s)	0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5
x(cm)	3.23	4.57	5.75	6.73	7.7	8.93	10.01	9.66	9.49	9.61	9.59

5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
9.49	9.59	9.5	9.61	9.49	9.49	9.76	11.19	12.97	18.28

10.5	11	11.5	12	12.5	13	13.5	14	14.5	15
20.98	20.98	20.98	20.98	20.98	20.98	18.05	16.15	15.39	13.48

15.5	16	16.5	17	17.5	18
12.78	11.65	11.75	11.65	11.65	11.65

Detail your calculations.

Section 1:

Section 2:

Section 3:

Section 4:

Section 5:

Section 6:

What is the average speed for the whole journey? To calculate it, use the following correlation:

$$v_{average} = \frac{S_{total}}{t_{total}} =$$

2.3.3. Study of variable motion - data collection and analysis

Now it's your turn! 😊

Tools

- Telephone
- Arduino + USB cable + laptop
- 4 cables and a „breadboard” panel
- Ultrasonic distance measuring sensor
- Graph paper
- Excel and Data Streamer

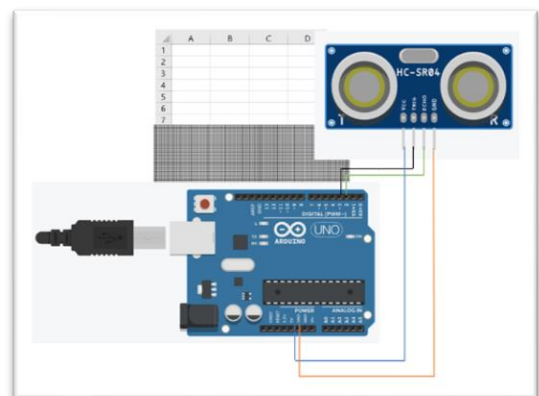


Fig. 12: Instruments used for measurement.
Source: own editing, Tinkercad.

The description of the exercise

Move your phone in front of the distance measuring sensor as shown in the following steps. Display the measurement data using the Excel Data Streamer. The circuit diagram and the code are the same as before.

Read the following steps carefully before starting the measurement.

- Step No. 1: Place your phone directly in front of the distance measuring sensor.
- Step No. 2: Start moving the phone away from the sensor at a steady speed for a few seconds. After starting the movement, start recording data.
- Step No. 3: Stop the phone and leave it still for a short time.
- Step No. 4: Move again and speed up the phone.
- Step No. 5: Stop.
- Step No. 6: Move the phone backwards (towards the sensor), then stop.

Plot your measurement data on a graph and then indicate the average speed for each section.

Also calculate the average speed for the whole journey.

2.3.4. Uniformly varying motion

The movement of a trolley rolling down a 2 m long slope was examined using a video analyser.

The video analyser shows the movement frame by frame. The motion data is shown in the following table.

Frame	Frame No. 1	Frame No. 2	Frame No. 3	Frame No. 4
Δt (s)	1 s	1 s	1 s	1 s
t (s)	1 s	2 s	3 s	4 s
Δs (cm)	10	30	50	70
Average speed $\left(\frac{\text{cm}}{\text{s}}\right)$	10	30	50	70
s (cm)	10	40	90	160
Momentary speed $\left(\frac{\text{cm}}{\text{s}}\right)$	20	40	60	80

Plotting the speed versus elapsed time yields the following graph.

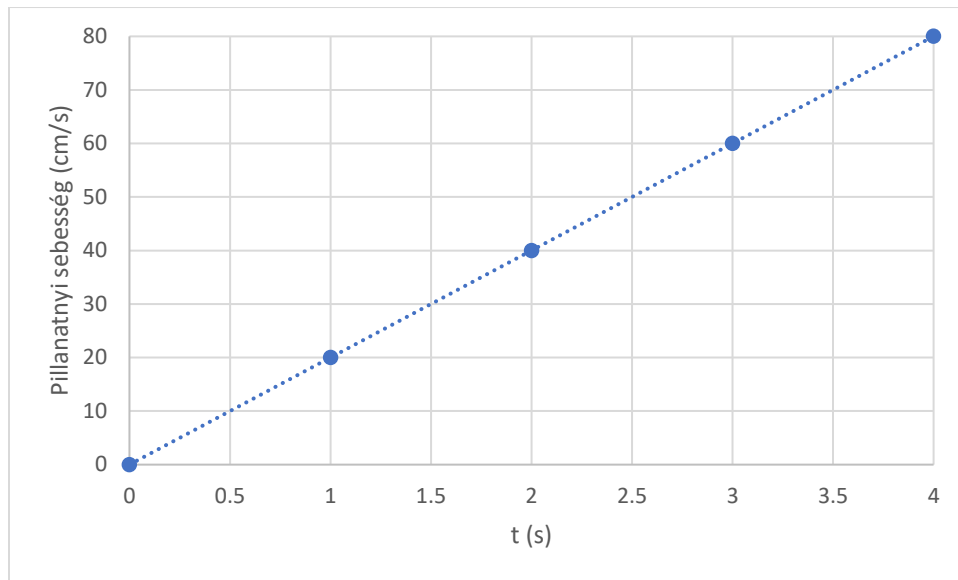


Fig. 13: Moment-by-moment speed-time graph. Source: own editing, Microsoft Excel.

Answer the following questions:

1. What is the relationship between the time elapsed from the start and the achieved velocity?
2. Determine how much the speed changes in 1 s?
3. Express the relationship between the change in velocity (Δv) and the elapsed time (Δt) in mathematical terms.

4. Plotting the distance travelled against the time elapsed produces the following graph.

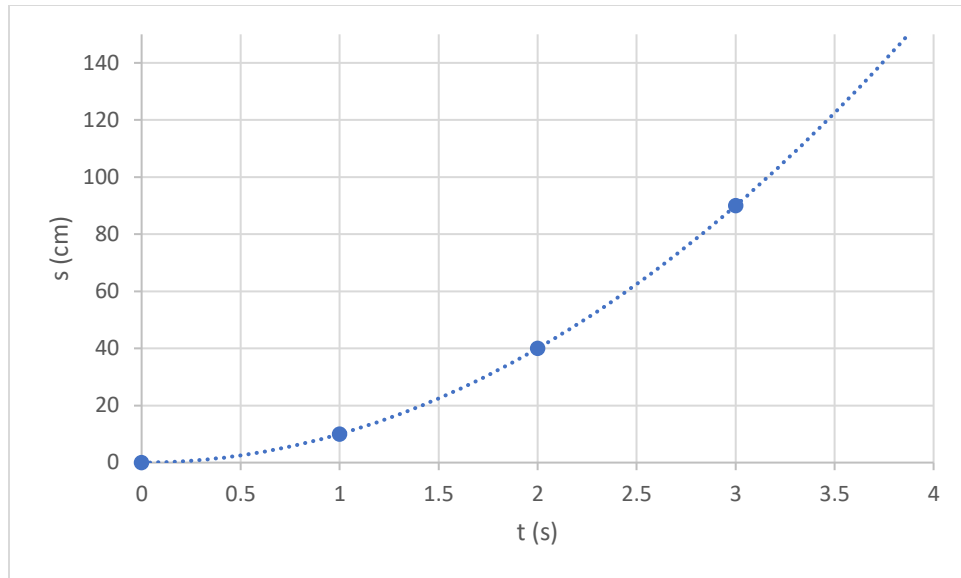
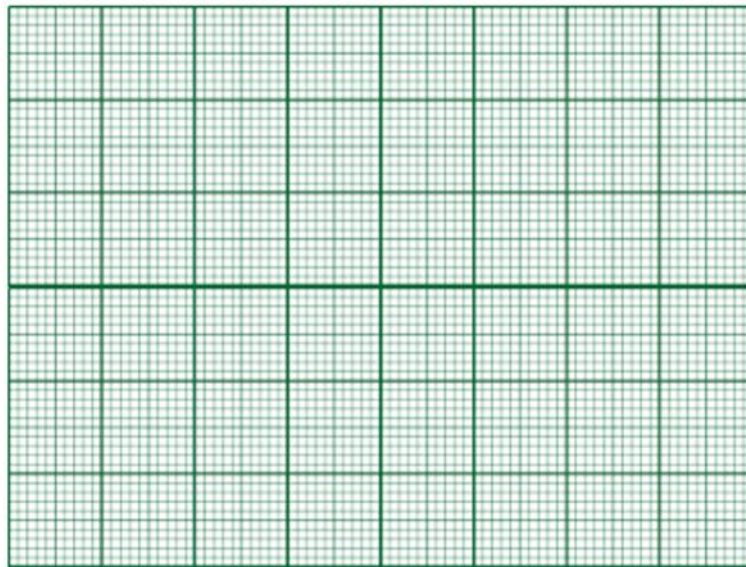


Fig. 14.: Distance-time graph. Source: own editing, Microsoft Excel.

Graph the distance travelled as a function of the time squared - on graph paper and in Excel.



5. Based on the graph, answer the following questions:

- What is the relationship between the distance travelled and the square of the elapsed time?
- What function is plotted?

2.3.5. Movement on a slope

Let's now test the movement of the trolley on a slope using Arduino measurements.

Tools

- Slope (track/board) and trolley
- Arduino + USB cable + laptop
- Cables
- Ultrasonic distance measuring sensor
- Excel Data Streamer

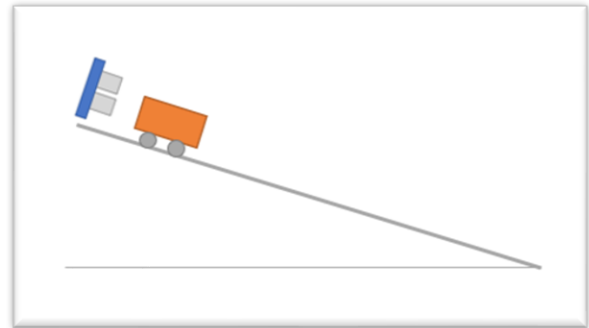


Fig.: Instruments used for measurement.
Source: own editing.

The description of the exercise

Set up the connection as before, using the code you are already familiar with. Adjust the board so that you get a flat slope (the angle of the slope should be subtle).

Place the distance measuring sensor on top of the slope and the trolley directly in front of it. Start the data acquisition with the Data Streamer, and at about the same time as you start the data acquisition, release the trolley (launch it without initial speed) on the slope.

Questions and exercises

1. Organise your data in a table in Excel according to the pattern below:

	A	B	C	D	E	F
1	#!	Workbook Custom	Custom			
2	#!	To play this data back: Open the original Excel workbool				
3	#!	For more information visit http://aka.ms/hackingstem/				
4	2.89		t (s)	x (cm)	s (cm)	
5	9.76		0	2.89	0	
6	23.36		0.5	9.76	2.89	
7	42.31		1	23.36	20.47	
8	67.51		1.5	42.31	39.42	
9	98.5		2	67.51	64.62	
10			2.5	98.5	95.61	

Fig. 16: Organising data in Excel.
Source: own editing, Microsoft Excel.

2. Plot the distance travelled by the trolley as a function of time. Use Excel to plot the graph.

3. True or false?
 - a) The distance travelled is directly proportional to the time needed to cover it.
 - b) The distance travelled is a linear function of time.
 - c) Over a unit of time, the object travels an increasingly greater distance down the slope.

4. Square the time figures in Excel, then plot the distance travelled by the trolley on the slope as a function of time squared.

5. Formulate 3 true statements based on the graph.

2.3.6. Freefall

Freefall is a uniformly varying motion along a straight line. In the case of free-falling objects, air resistance shall be neglected. The velocity of a free-falling object always changes by the same amount per time unit. In 1 s, the velocity change in Hungary is 9.81 m/s. Based on this, determine the acceleration of the free-falling object. The acceleration of a free-falling object is denoted by g and is called gravitational acceleration.

$$g = \frac{\Delta v}{\Delta t} = \quad = \quad \frac{m}{s^2}$$

1. The following graphs of the movement of a free-falling tennis ball were created using a video analyser.
 - a) a) Assign a title to the graphs based on which quantity is displayed as a function of what. (distance-time/ speed-time)

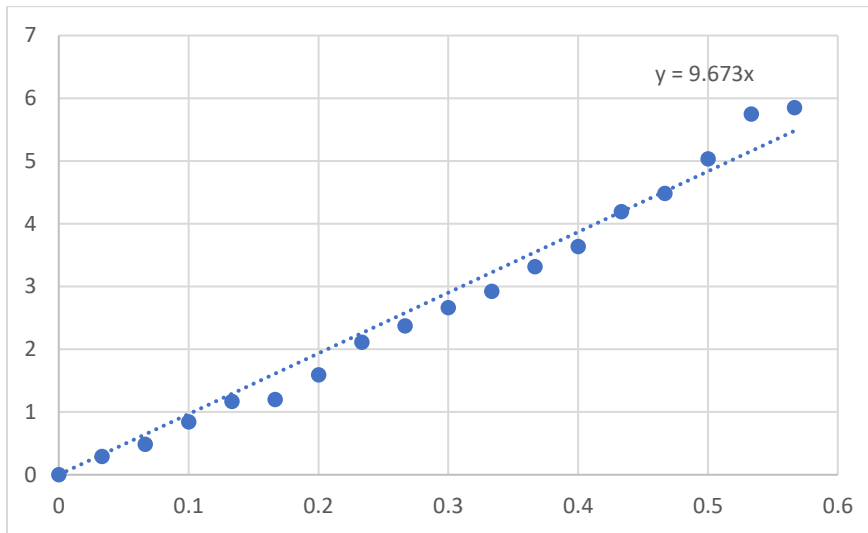
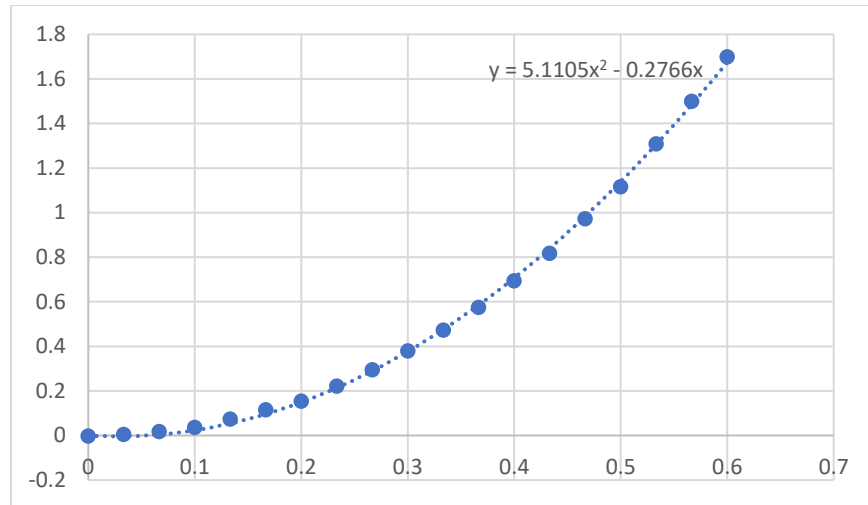


Fig. 17 a) and b): Diagrams of free-falling body motion.
Source: own editing, Microsoft Excel.

- b) Label the axes
- c) The equation for each function is also plotted on the graph. Determine the value of the gravitational acceleration based on the equations.
- d) What causes the possible discrepancy from the already known value of $9,81 \frac{m}{s^2}$?

2.3.7. * Bonus measurement:** One can also use the already well-known code and ultrasonic distance measuring sensor to take measurements in order to check freefall. Design and carry out a measurement to investigate the change in velocity of a free-falling object and determine the value of gravitational acceleration.

2.3.8. Varying motion - practice exercise

The following graph shows the movement of a car from one traffic light to another.

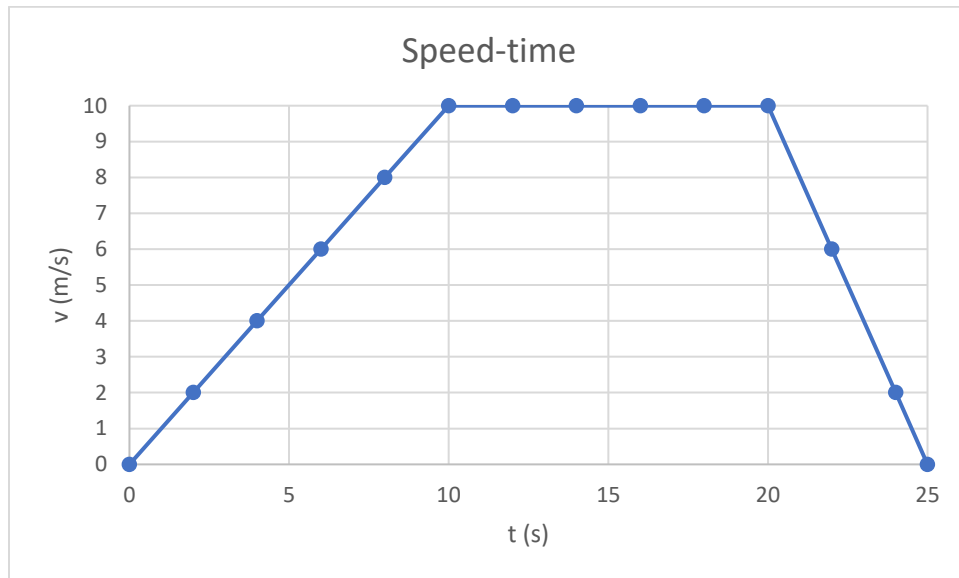
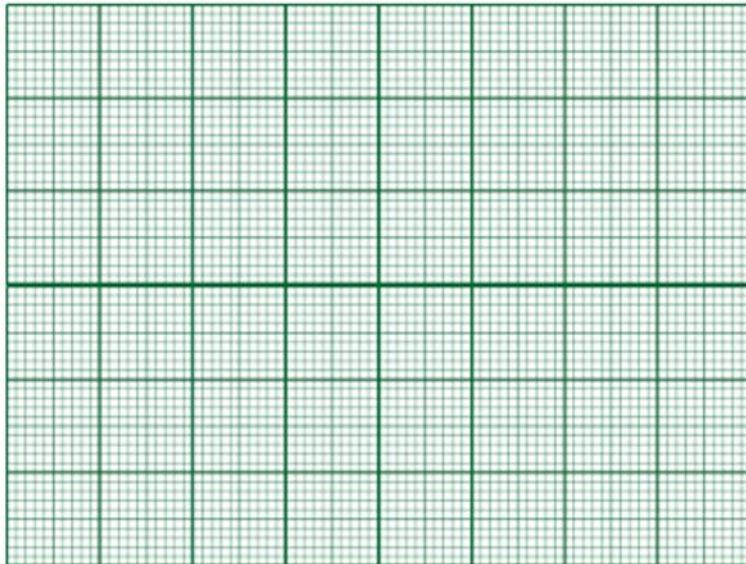


Fig. 18: Speed-time graph. Source: own editing, Microsoft Excel.

- How many sections can the movement of the vehicle be divided into?
- Describe the movement of the vehicle on each section.
- Calculate the distance travelled by the car in the first 10 seconds.

d) Calculate the total distance travelled during the movement.

e) Prepare the acceleration-time graph of the movement.



2.4. Arduino exercises - experiments and measurements: periodic phenomena

Periodic phenomena show a regular recurrence over time.

For example, the changing of the seasons, carousing, etc.

2.4.1. List other periodic phenomena!

-
-
-



Fig. 19: The changing of the seasons. Source: macrovector / Freepik.

2.4.2. Examine the motion of the body vibrating on the spring.

Graph the position of the object suspended on the spring as a function of time. In order to do this, carry out an experiment with Arduino. Set up the experiment according to the diagram. For the test, use an ultrasonic distance measuring sensor, which is operated by the previously used circuit diagram and programme (scroll back 😊). Use Excel Data Streamer to collect the data.

Tools: spring, a weight from the weight series to hang on the spring, tripod, cardboard, glue, laptop, Arduino, cables, panel, ultrasonic distance measuring sensor.

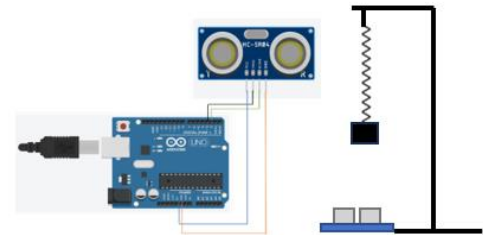


Fig. 20: The experimental arrangement. Source: own editing, Tinkercad.

The description of the exercise

Hang a spring on the stand, choose a small mass of 50 g and glue a cardboard plate to the bottom. Then hang the weight on the spring. The eventually reaches a **position of equilibrium**, at which point the object hung on the spring is at rest (the spring does not vibrate).

Displace the object from its position of equilibrium - at this point the weight hanging on the spring starts to vibrate around a position of equilibrium between the two extreme positions. The distance from the position of equilibrium is called the **deflection** and is usually denoted by y . The maximum displacement of the object is called the **amplitude**. It is denoted by A . At this point the object is at its extreme position.

Use Arduino to collect data. Trace the position of the vibrating object in time.

In order to do this, the code used so far has to be slightly modified. Set the time period between incoming data to 10 ms.

```
const int trigPin = 3;
const int echoPin = 2;
long duration;
float distance;

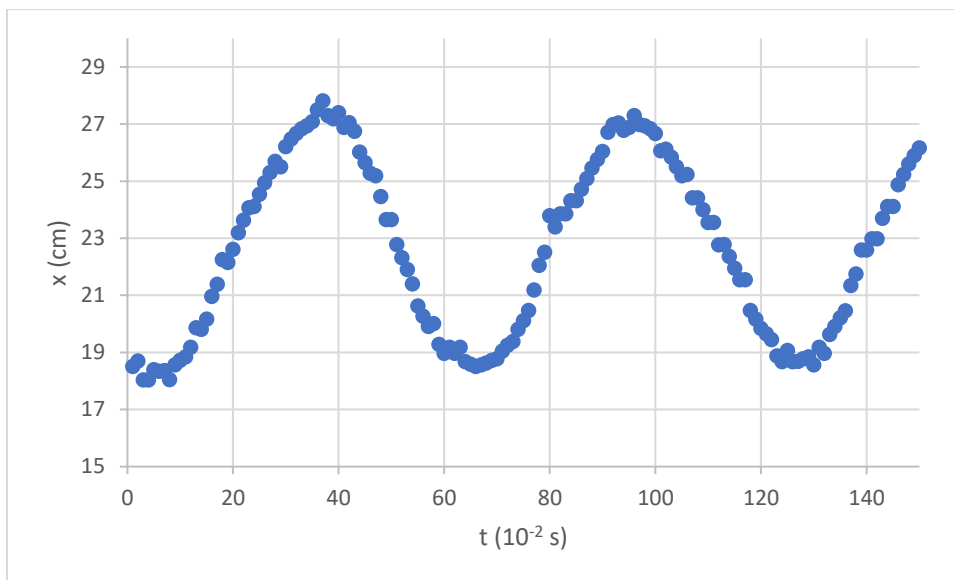
void setup() {
  Serial.begin (9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}
void loop() {

  digitalWrite(trigPin, HIGH);
  delay(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.0340 / 2;

  Serial.println(distance);
}
```

1. Connect the tool to *Excel Data Streamer*. Get the body to vibrate and start recording data. Plot the distance from the sensor as a function of time. Remember, 10 ms elapses between each data input. Delete any data that is out of range.

2. Mark the position of equilibrium and the extreme positions on the graph.



3. What is the amplitude of the vibration?

A=

4. Indicate the time of a full oscillation! ("**Oscillation time**": the time required to make one complete oscillation. This is the time it takes for the oscillating object to move from one of its extreme positions to the other, and then back to its initial extreme position.)

Sign T

Unit of measurement: s

T=

5. **Frequency (number of oscillations)**: number of oscillations per a unit of time, the reciprocal of the oscillation time.

Sign: f, Unit of measurement: 1/s

Determine the frequency of the oscillation.

f=

6. Repeat the previous experiment by hanging 2 small masses of 50 g and then 3 small masses of 50 g on the bottom of the spring. Always stick the cardboard on the lowest mass!

How does the oscillation period and frequency change as the mass of the vibrating object is increased?

7. Design an experiment to investigate damped vibration. What is needed to damp the oscillation, and then gradually stop it?

Tools, plans for the layout of the experiment

8. Carry out the experiment. Plot the motion of the oscillating object on a space-time graph. Express how the amplitude of the oscillation varies as a function of time.

3. Summarising measurement exercise with photoresistors - Exoplanet exploration in the classroom

Exoplanets are planets outside our Solar System. Studying them is an important task of modern astronomy. The aim is to find as many planets as possible in our neighbourhood that exhibit the conditions of habitability - so that interstellar travel can even become a reality in the future.

Researchers use a variety of methods to detect exoplanets, including the so-called *transit method*.

This method involves examining a planet not directly, but via its star. The planet orbits periodically around its star. As it passes its star, the observed brightness of the star decreases.

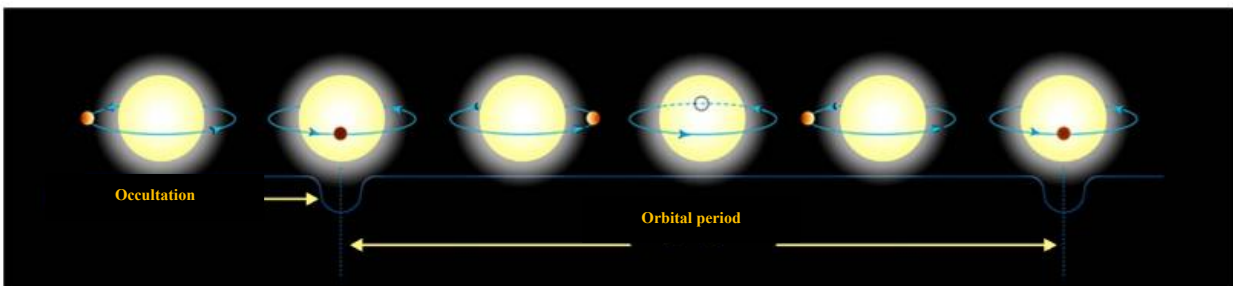


Fig. 21: Brightness reduction due to occultation. Source: http://astro.u-szeged.hu/oktatas/csillagaszat/6_Naprendszer/0108exobolygok/tranzit_periodusa.jpg

Model the transit method in a school setting. This time another sensor is going to be used for the measurement, the *photo-resistor*, which is used to measure the intensity of the light falling on the photo-resistor.



Fig. 22: Photoresistor. Source: <https://www.hestore.hu/>

Tools

- Arduino board and panel
- Laptop
- USB-cable
- Cables
- Photoresistor and 100 Ω resistor
- 15 W LED bulb (E27)
- Condenser lens used in student experiments in a holder
- Cap fixed on a skewer

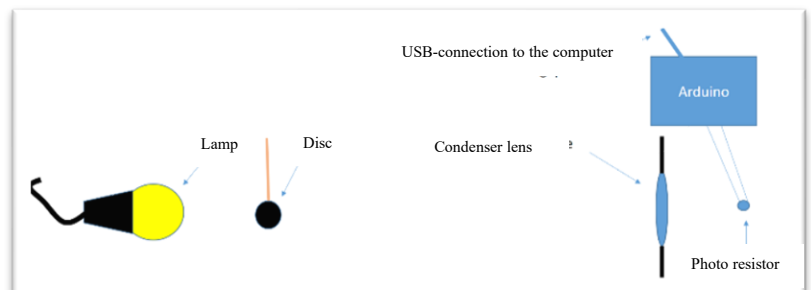


Fig. 23: Experimental arrangement. Source: own editing.

The description of the exercise

Using the attached figures, assemble the electrical connections and then the experimental setup.

The electrical connections

Build the electrical connection as shown in the diagram. Take a 100 Ω resistor and ground it as shown in the diagram, i.e., connect it to the Ground (GND) pin of the Arduino. In series with the resistor, connect a photoresistor, one leg of which is connected to the analogue input of the Arduino "A0" and the other leg to a connector with a permanent 5 V voltage.

Now set up the experiment (the lamp represents the star, the cap models the planet).

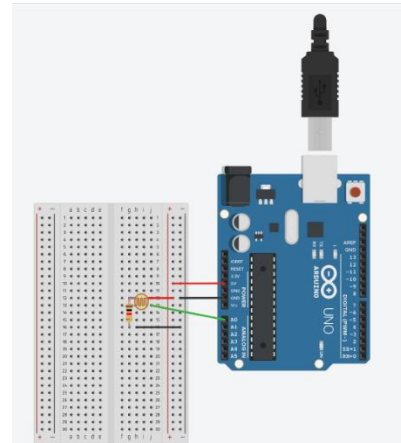


Fig. 24: Experimental arrangement. Source: own editing, Tinkercad.

The lamp should be at least 1 m from the lens! Shape the light source with the condenser lens. Place the photoresistor at the point where the light from the condenser lens is projected in a circle with the smallest cross-section (sharp image).

The intensity of the light is measured by means of the photoresistor.

Use the following code for the measurement.

```
int incoming=A0; // incoming analogue signal on the analogue connector
int sensor data = 0; //set initial value is set to 0

void setup () // this is where the setup of the software starts
{
  Serial.begin (9600); // serial communication, max 9600 bit/s incoming
  data
}

void loop () // loop, executed again and again after a given period of time
{
  sensor data= analogRead (incoming); // the sensor data is going to be the
  scanned value
  Serial.println (sensor data); // values are printed below each other
  delay (100); // 100 ms delay between two pieces of incoming data
}
```

The code is also available [here](#): with the title *Exoplanet - measurement with a photoresistor*.

Questions/exercises

1. Model the transit method. Periodically move the cap attached to the skewer at a constant speed - i.e., repeatedly around the illuminated LED bulb. Measure the light intensity with a photoresistor and study the changes in intensity. Use the *Excel Data Streamer* for the measurement.
2. Determine the relative light intensity. Find the highest intensity value and assign 100% to it, then determine for each piece of data the percentage of maximum light intensity it represents.
3. Prepare a time-relative light intensity table, remembering that the readings are acquired every 100 ms. Plot the data in a graph (relative light intensity versus time).
4. Determine the period time of the "planet".
5. Estimate the speed of the planet.

Summary

What have we learned?

1. The maze⁵ hides the concepts you have learned in the topic. Find them and compile a glossary.

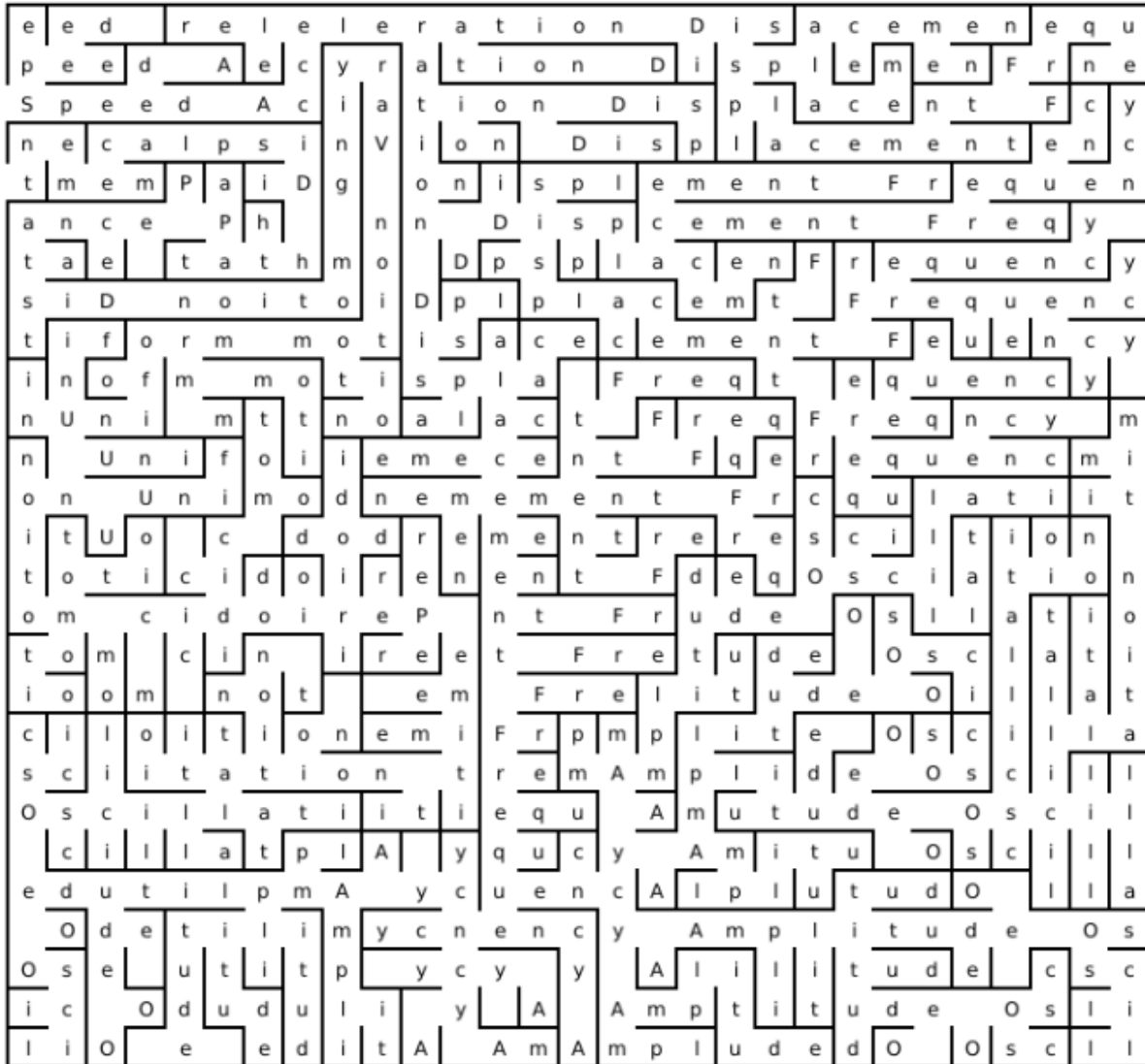


Fig. 25: Summarizing maze. Source: own editing. <https://www.festisite.com/>

Speed Acceleration Displacement Frequency Amplitude Oscillation time Periodic motion Uniform motion Varying motion Distance Path Displacement

⁵ The maze was created on [festisite.com](https://www.festisite.com/). <https://www.festisite.com/text-layout/maze/>

2. The following table shows the movement of a toy racing car. The movement was tested with an ultrasonic distance measuring sensor, measuring the distance from the ultrasonic distance measuring sensor to the race car.

The table shows the distance (x) of the toy car from the sensor as a function of time:

t (s)	0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5
x(cm)	3	4.5	6	7.5	9	10.5	12	10.5	9	7.5	6
s (cm)	0	1.5	3	4.5	6	7.5	9	10.5	12	13.5	15

2.1. Fill in the last row of the table. What was the distance travelled from the beginning of the movement at the given moment in time?

- What was its displacement during the examination?
- What was the distance travelled during the examination?

2.2. The following graph shows the location-time data for the race car:

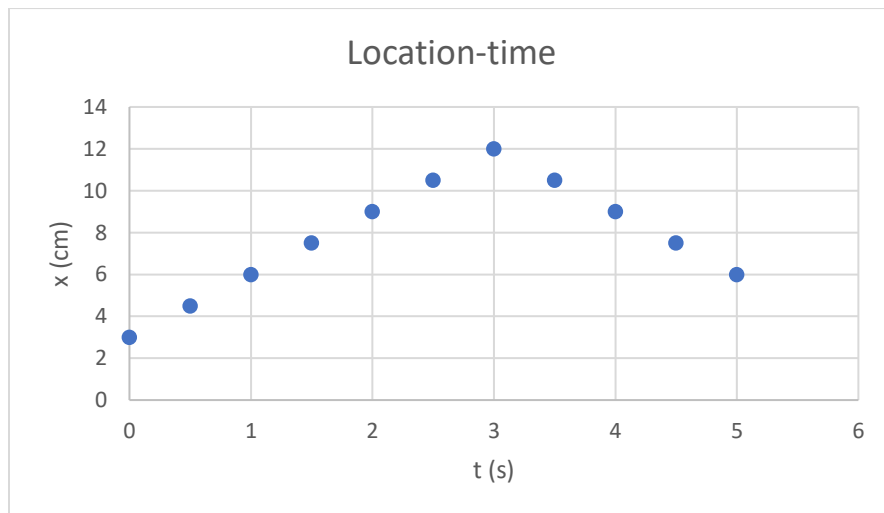
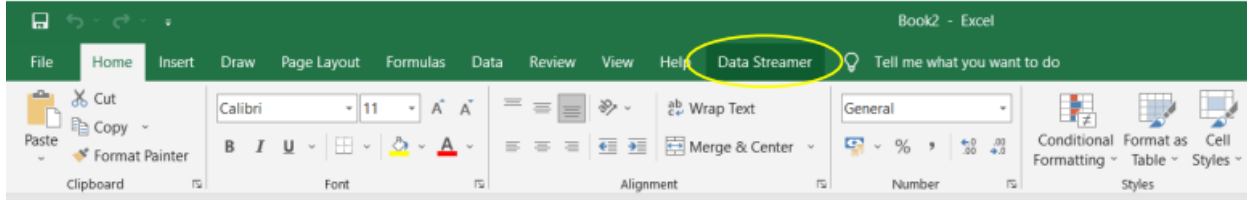


Fig. 26: Location-time graph for practice. Source: own editing, Microsoft Excel.

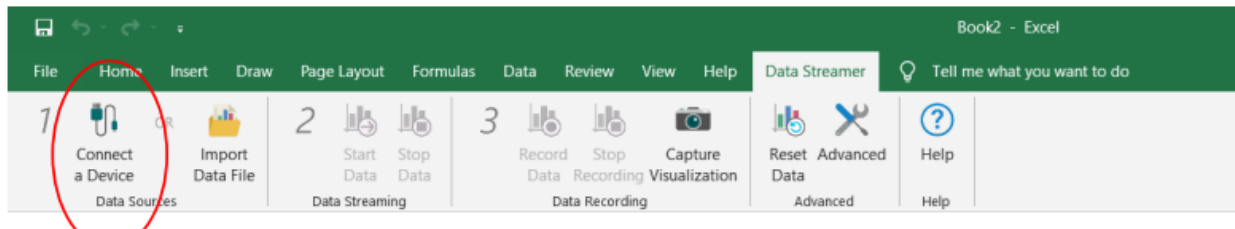
- Plot the associated distance-time values on a graph in Excel.
- Determine the speed of the car in each section and also the average speed for the whole journey.

Appendix – Tutorial on how to use the Excel Data Streamer

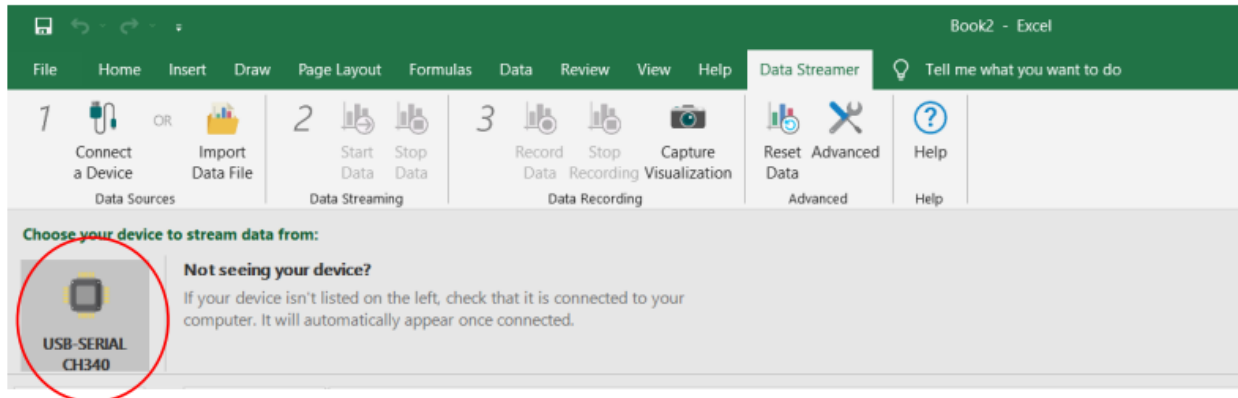
1. On the menu ribbon, click on the Data Streamer menu.



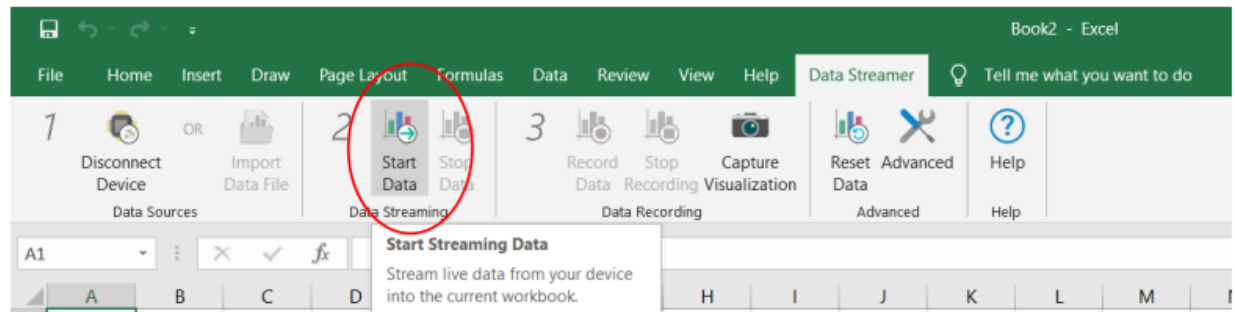
2. Then click on the Connect a Device menu point.



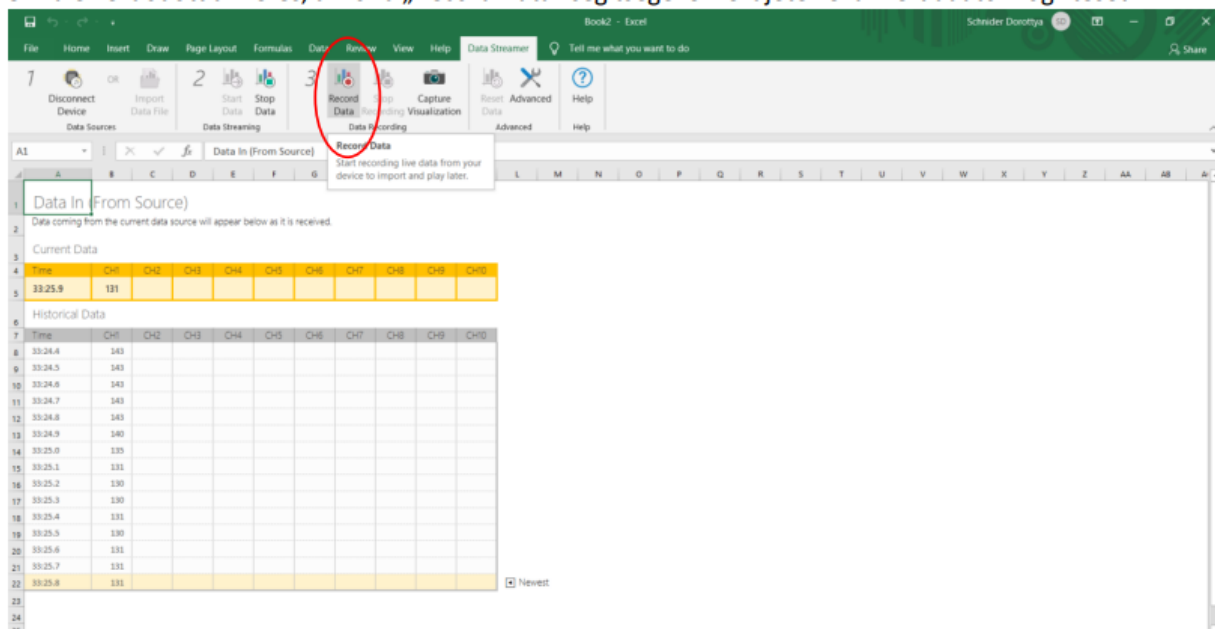
3. Click on the "USB Serial CH340" button.



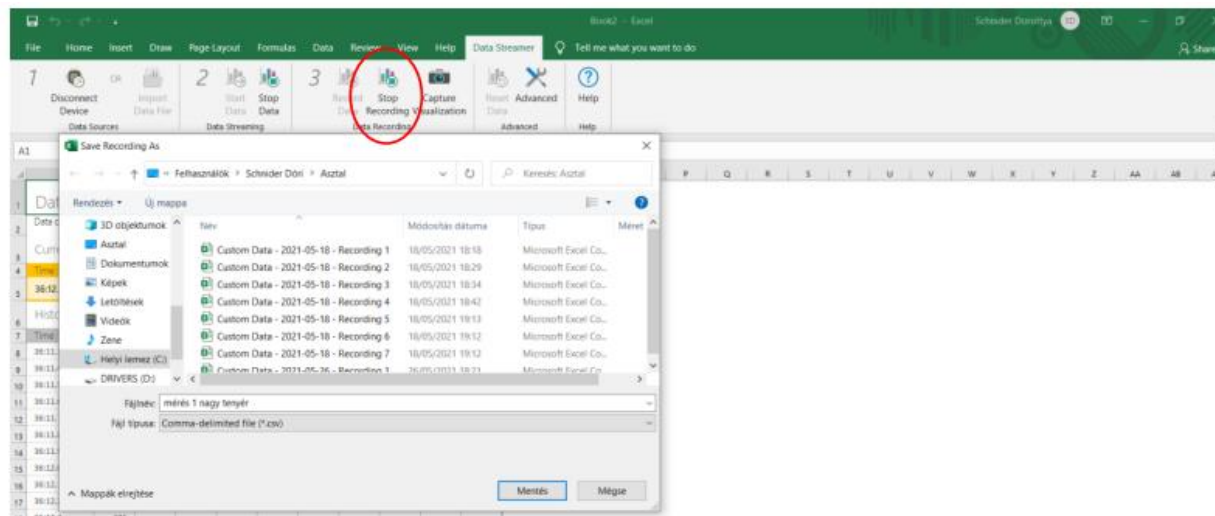
4. Start the real-time data streaming with the "Start data " button.



- Once the measurement has started, click on the "Record data" button to start recording the incoming measurement data.



- Once the measurement is complete, stop the recording by clicking on the "Stop recording" button. Then save the measurement data in the appropriate "filename.csv" format.



- Open the previously saved document "filename.csv" and start evaluating the data according to the instructions.

The Data Streamer can be downloaded from the following website: <https://www.microsoft.com/en-us/download/details.aspx?id=56976>