Investigation of the Rotational Motion of the Fidget Spinner by Means of Technical Informatics

SOMOGYI Anikó, KELEMEN András, MELLÁR János, MINGESZ Róbert

Abstract. The boundaries between science, engineering and computer sciences seem to have been blurred in the 21st century. Microcontroller programming plays a role in informatics education, while modern physics teaching integrates the potential of computer simulations and computer-controlled measurement experiments. In order to demonstrate the latter, we present an experiment to precisely identify the dynamic causes of a kinematic phenomenon solved by means of technical computer science. To investigate the role of air resistance, we developed a vacuum chamber in which the signals of sensors were processed by the NI myDAQ multifunctional device and the popular Arduino UNO. A wide range of ways is presented to control data acquisition and numerical analysis of a pulsed signal produced by a photogate, as well as graphical representation and evaluation of derived time-dependent signals. Controlling the two devices due to their characteristics and hosting of measured data by a computer require different development tools from the user (LabVIEW, C/C++). The results and methods of the study can be applied in physics and computer science education at secondary school level.

Keywords: interdisciplinary education, technical computer sciences, Arduino, LabVIEW, Excel Data Streamer

1.   Introduction

In the 21st century computer sciences, engineering and science in general seem to have blurred boundaries.

Microcontroller programming plays a role in IT education. A good example of this is the spread of micro:bit in the UK [1], [2], which is programmable in Python and JavaScript or the inexpensive C/C++ based Arduino. The number of publications that use these boards has increased dramatically in the last ten years [3], [4]. At the same time, modern physics teaching integrates the potential of computer simulations and computer-controlled measurement experiments. The NI myDAQ multifunctional device - programmable in LabVIEW and developed for school measurement - is also gaining popularity. Members of the Hungarian Academy of Sciences' Research Group of Technical Informatics Methodology (among them, some authors of this article) are also working on the renewal of computer and natural sciences education, which is the basis of technical education in Hungarian secondary schools [5], [6]. Generally speaking, researchers see opportunities in software-controlled measurements to increase the popularity of fields of sciences mentioned above, as well as filling skill shortages connected to technical sciences by youth education [7].

2.   Impact of Information Technology in teaching the Scientific Method

The Scientific Method basically always starts with gaining human experience and collecting information (cognition, experiment, measurement). This is followed by the evaluation of data, the search for connections and their systemisation. Based on these, one can formulate general regularities and make a mathematical model (e.g. set up ratio pairs, equations, formulas). The model makes it possible to derive new statements, but these must always be compared to experiments [8]. Throughout human history, the development of measuring instruments has always led to the development of natural sciences [9]. Naturally, the advancement of computing technologies opens

up new doors to this day, as it can make measurement, evaluation, systematization and modelling more accurate, faster and more efficient in the detailed process [10]. We consider it important to introduce this modern role of informatics to our students starting in early childhood and to provide them with the means of inquiry-based learning. Combination of information technology and science can play a role in motivation: experiments embedded in information technology, as well as the enrichment of otherwise less popular natural sciences with IT content, have a motivating and attention-attracting effect. Instead of general principles, methods and procedures, we would like to present a computer-assisted examination of a specific physical phenomenon to illustrate how technical computer sciences can provide possibilities for interdisciplinary education of physics and computer sciences [11].

Problem

In international literature on physics education fidget spinners are increasingly featured. By spinning the well-known game, the angular velocity of its high-speed rotation decreases over time due to braking effects [12], [13]. On several online forums, video sharing portals it is suggested (or declared) that the effects of air resistance or rather friction dominate [14]. When this easy-to-understand physical phenomenon is examined by the modern digital metering, the question in discussion is realized not to be an easy one. However, our computer-driven experiment will help uncover the cause-effect relationships behind the phenomenon.

Objective

During the presentation of our examination, we try to highlight the applied IT methods and ideas which we hope will turn out to be applicable at various levels of education.

3. 'Hardware' used in the measurement

When looking at the kinematic characteristics of the rotational motion (e.g. angular rotation, angular velocity), namely answering the question "How does it move?", it is sufficient to connect a simple photogate to a sound card [15]. However, if aiming to answer the question "Why does it move that way?", i.e. the exploration of dynamic causes, then the two types of braking effects on the spin must be distinguished during the measurement. The idea came naturally, that air resistance could technically be eliminated by a control experiment performed in a vacuum.
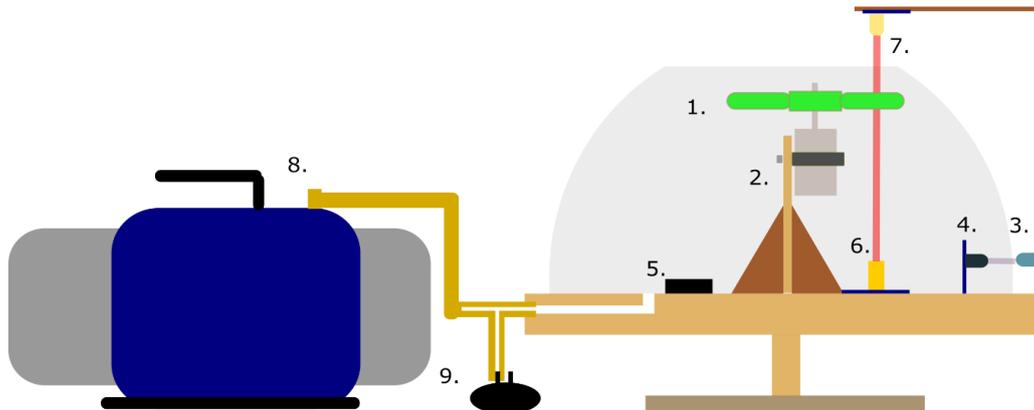
3.1. Measurement setup

An experimental setup was needed in which the movement of the fidget spinner could be examined at normal atmospheric pressure and also makes it possible to suck out the air around the toy and to repeat our experiment under changed conditions.

The fidget spinner was placed in a self-made vacuum chamber. As a lid, a salad bowl was used, while the bases was a wooden board. A ring – fitting to the mouth of the bowl – was cut from a silicone baking sheet and was stuck by neutral silicone sealant to the base. Obviously, it is not possible to manually spin the fidget spinner in a vacuum, so a DC motor (1 - 3 V, max. 6600 ± 10% RPM) remote-controlled by infrared transmitter and receiver was used for this purpose.

The motion detection photocell was constructed by placing an illuminating laser diode under the fidget spinner, and a phototransistor detecting the light beam and its interruptions above the glass. The air was sucked out of the chamber by a high-performance vacuum pump, and the inner air

pressure was monitored by a pressure meter. A schematic diagram of this self-designed vacuum chamber and a photo of the experimental setup are shown in Figure 1 a-b, respectively.

Figure 1.a: Structure of the measurement setup. Legend: 1. fidget spinner, 2. DC motor mounted on a stand, 3. infrared transmitter circuit, 4. infrared receiver circuit, 5. batteries 6. laser diode module 7. phototransistor circuit (6 - 7. combined: photogate), 8. vacuum pump, 9. pressure sensor[1]
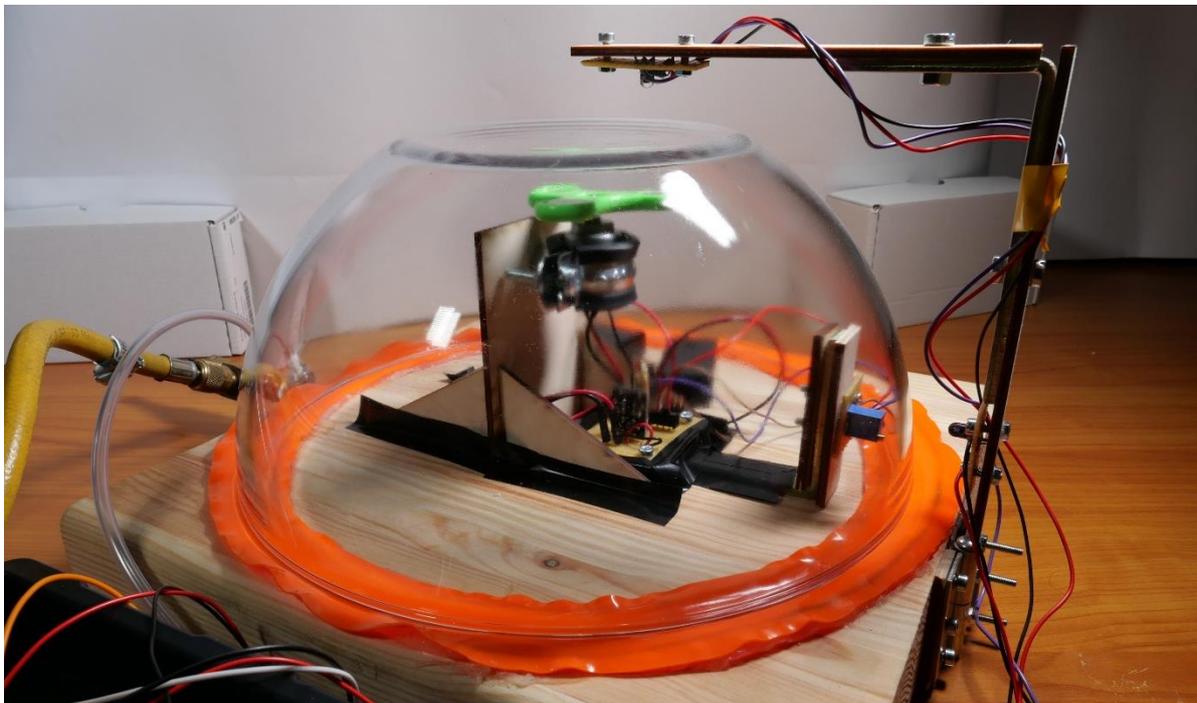
Figure 1.b: Photo of the measurement setup

3.2. Electronic implementation of measurement tasks

We also designed custom electronics for our measurement setup. Tasks to be solved were

a) Power supply to the motor that drives the fidget spinner
b) Power supply to the laser diode and the transistor switching the motor

---

[1] Figure was made in Inkscape.

c) Control of the IR transmitter and receiver opening the transistor
d) Detecting the signal of the photogate
e) Measuring the pressure.

3.2.1. Sensors and converters

In the process of digital signal processing, sensors that convert physical signals into electrical signals and devices that convert analogue electrical signals into computer-readable digital signals are indispensable. Their roles are illustrated in Figure 2 [16].
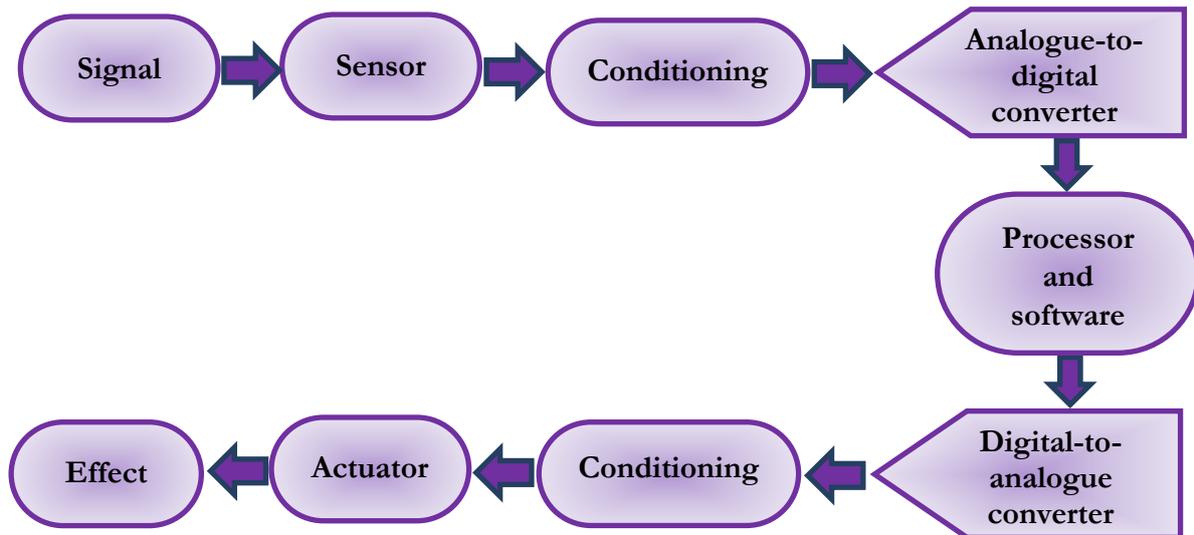

Figure 2: Process of digital signal processing

During the measurement, an MPX 4250DP type pressure sensor was connected to the vacuum hose with valve. . This is a so-called differential, calibrated sensor that tracks the pressure in the vacuum chamber in comparison to the outside air pressure [17]. As mentioned above, for motion detection the self-assembled photogate consisting of a KY-008 laser module and a phototransistor was applied [18], [19].

The sensors and circuits performing the listed measurement tasks were connected to two types of measurement devices (myDAQ, Arduino UNO), which play an analogue-to-digital converter role among others in our measurement process.
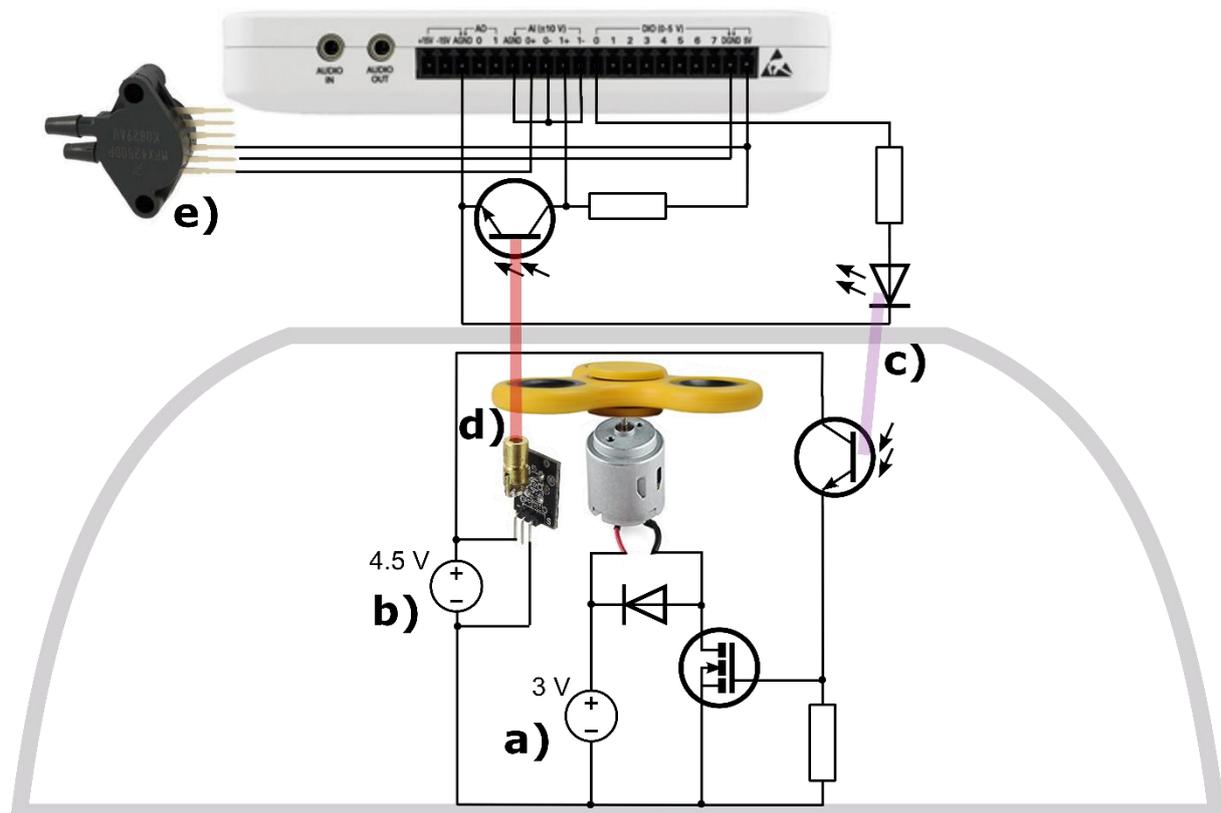
3.2.2. Signal measurement with a myDAQ

The measurement was performed with a multifunctional device called myDAQ (DAQ stands for Data Acquisition), which was developed by National Instruments directly for school laboratory measurements. On the device two analogue input (AI) and two analogue output (AO), eight digital input / output (DIO), one audio input and one output connector, and digital multimeter (DMM) connectors can be found. Power to the instrument is provided via USB port, just like the bidirectional communication between the instrument and the computer controlling data logging and transferring to the measured data [20].

In the first version of our measurement, the photogate signal was measured on the AI1 analogue channel at a sampling frequency of 100,000 samples per second. As one of the arms of the toy passes through the photogate, the measured voltage changes from ~ 0V to ~ 5V. When the angular

velocity of the motor-driven fidget spinner is 110 $\frac{rad}{s}$, a voltage switch occurs in every 19 ms, and due to the deceleration, growingly longer periods of time can be measured. As a result of the measurement, an analogue rectangular signal characterized by an increasing pulse width is obtained.

The pressure was measured on the AI0 analogue channel of the myDAQ at a sample rate of 1000 samples per second. The advantages of measuring an analogue signal are that we can see the waveform, we constantly sample it, and the hardware controls reliable timing. Circuits constructed to perform the previously listed measurement tasks (a-e) and connected to the myDAQ are shown in Figure 3.

Figure 3: Wiring diagram of the circuits connected to the NI myDAQ device. In the lower and upper half of the figure, the circuit inside and outside the vacuum chamber is shown, respectively[2]

3.2.3. Signal measurement with an Arduino

The NI myDAQ is specifically designed for student experimentation, also considered very user-friendly in many ways, however, it is relatively expensive to acquire. Although, to our knowledge, myDAQ is available in a growing number of Hungarian secondary schools through various competitions and grants, we present a modified version of our measurement using an Arduino as well. The popular, reasonably priced, open-source and easily obtainable platform could, in

principle, be applied to measure *an analogue signal* in the same way as the myDAQ. However, the maximum sample rate is far below that of the device already described, less than 10,000 samples per second. Therefore, in our measurement with Arduino, the signal of the gate was switched to a digital channel and observed when the signal jumped from a logical low to a logical high, that is, when an arm of the spinner intersected the light path. Figure 4 illustrates how circuits connected to an Arduino UNO board can be assembled using a breadboard, e.g. for testing purposes. (The measurement tasks are indicated as in Figure 3.)
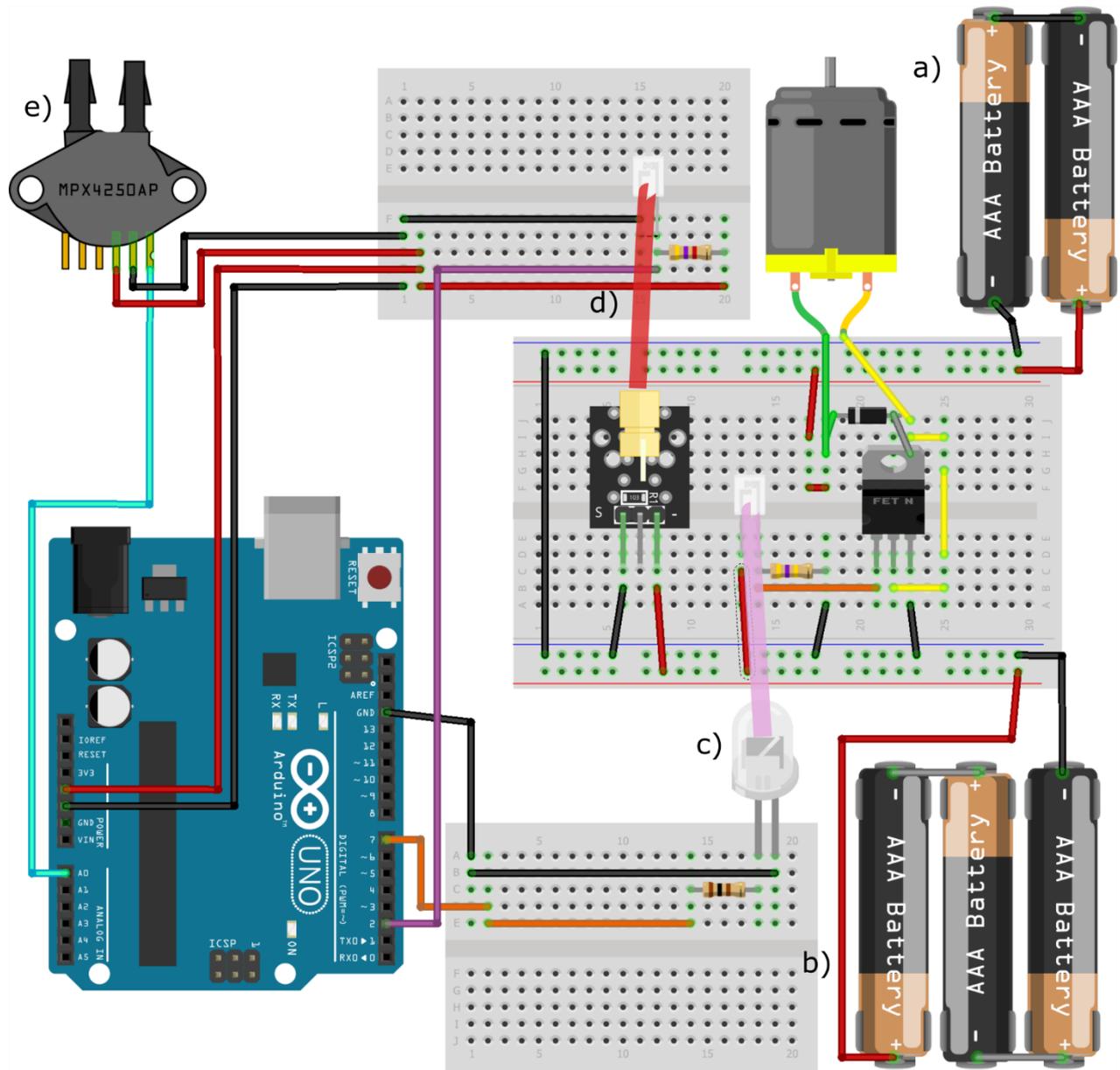
Figure 4: Building of circuits connected to an Arduino UNO on breadboards[3]

---

[3] Figure was made in Fritzing.

4. 'Software' used in the measurement

Programming the two devices described above is quite a different. MyDAQ is programmable in the LabVIEW graphical programming language, while Arduino is programmed in a C/C++ -based environment or even in Python or other text-based programming languages. The applied software solutions for these two devices are also radically different because of the difference between evaluation of analogue and digital signals, measured by the myDAQ and Arduino, respectively, based on the arguments above.

4.1.   Programming LabVIEW for the myDAQ

4.1.1. LabVIEW graphical programming environment

LabVIEW has been mainly developed for controlling industrial computer-driven measurements and for processing and evaluating measurement results. Programs written in LabVIEW are commonly referred to as virtual instruments (.vi), the two main parts of which are the Front Panel resembling a dashboard and the Block Diagram containing the program code. In the Front Panel, control fields (e.g. buttons, drop-down fields) and indicator fields for the user (e.g. numerical or string fields, graphs) can be displayed. In the Block Diagram, graphical objects can be placed that represent built-in functions, connected to each other by data lines, and arranged into different program structures. An important principle of the language is the so-called Data Flow: an instruction is executed in the program when it has received the required signals for all its inputs. In this way, the commands can be chain-linked in order to determine their execution order [21].

The NI-DAQmx driver is usually used to program the DAQ devices of National Instruments for configuration, data measurement and generation tasks according to our needs. The function library of the driver includes, among other things, creating, reading and writing a virtual channel, as well as starting, stopping and timing a task, etc. [22]

4.1.2. Solution of the producer-consumer problem in LabVIEW

The essence of the classical *producer-consumer synchronisation problem* i.e. two processes, the producer creating objects and putting them into a fix-sized *buffer*, and the consumer extracting and using data from the buffer are handled by the program. When the buffer is empty or full of data, one of the processes must wait. In our analogue measurement with the myDAQ the producer takes samples at determined intervals (in case of 100,000 samples/s the period is 1 μs) and places this data in a buffer for storing 100,000 samples.

A queue data structure is used to implement the first in first out (FIFO) principle, that is, whichever data comes first, leaves first. Before running the producer, the queue is created, and the producer fills it with 10,000 voltage values per cycle. Tasks of the consumer process in our virtual instrument are to retrieve the measured voltage signal from the queue, to return time interval between data points, then to arrange data series into arrays and finally to write it into a data file. Due to this solution, real-time measurement at equal intervals is not interrupted by the storage and processing of data.
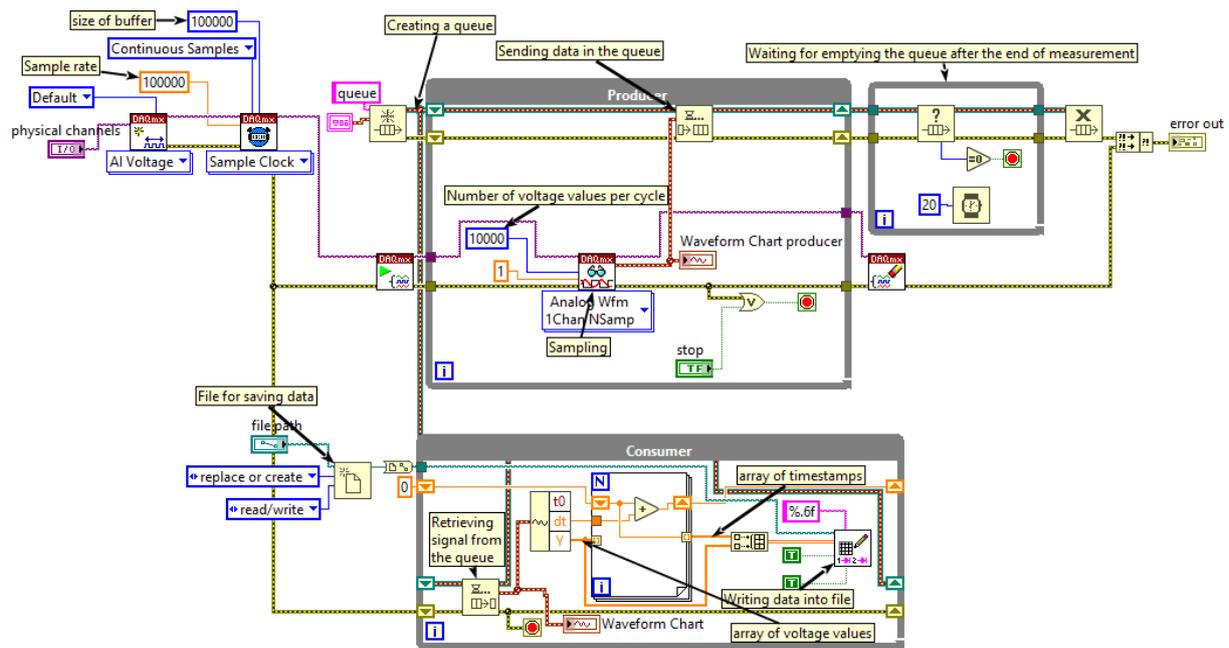
Figure 5: Block diagram of the LabVIEW virtual instrument for measuring the signal of the photogate

4.1.3. Postprocessing or live data streaming in LabVIEW

After completing the measurement driven by the virtual instrument described above, the measured data separated by tabulator characters are stored in a .txt file. The data can then be processed in a simple spreadsheet software or in LabVIEW, too. The first task is to perform a level crossing detection in the rectangle signal, that is, to find timestamps when the photogate signal has a jump and to collect them in an array. Between these periods, the fidget spinner moves 120°, and from its increasing rotation angle we also build an array with a for loop. This pair of arrays arranged in a cluster, i.e. rotation angle as a function of time can be plotted on an XY graph. It is also possible to visualize the results of more measurements on the same graph.

About physical changes it could be more informative to look at the speed of change. Let α_i and t_i denote the i-th element in the arrays of rotation angle (in radians) and timestamp, respectively.

$$\omega_i \approx \frac{\alpha_{i+1} - \alpha_i}{t_{i+1} - t_i}$$

Here, in fact, we calculate the value of the difference quotient, also called the forward numeric derivative. By performing the operations on the elements of arrays, a shift register can also be used to form an array of angular velocity and after clustering these values can be plotted as a function of time, illustrated in Figure 7. A great advantage of LabVIEW is that our graphical code of data postprocessing could be transformed and transported into the consumer process, and this way a near real-time graphical data representation would be implemented. This live data stream can be an excellent tool for demonstrational experiments in a classroom.

4.2. Programming embedded software for Arduino

Two software programs need to be discussed when using the Arduino platform. First, there is a code running on the microcontroller, written in C/C++. Data are transferred to the PC via USB serial port using a simple protocol: traverses of the fidget spinner in front of the phototransistor

are continuously detected and the timestamp (in the form of integers) of these events are sent. Each message is closed with a comma and a line break. Second, there are several ways to receive data on a host computer. We present three simple options for this in the rest of the chapter.

4.2.1. Implementation of data logging using an Arduino

While in case of the myDAQ, measurement is controlled by the LabVIEW software running on a computer, we need to implement an embedded software when using an Arduino, though. The purpose of this software is to measure the incoming signals and then send the received data to the computer through serial port. For the implementation, we searched for software tools that are well known and compatible with all Arduino boards (so we did not apply direct writing and reading of registers). There is no way to sample analogue signals as quickly on an Arduino as with a myDAQ. However, the signal we want to measure is digital: the fidget spinner either intersects the path of light or not. Because of this, it can be directly measured through a digital input. Yet, care must be taken to ensure that the signal levels measured at the output of the phototransistor are within the voltage range sensed by the Arduino.

When precise and accurate time measurement is required, the best way is to use interrupts [23]. The Arduino can receive external interrupts where the interrupt servicing routine can be set to be called for the rising edge (switching from 0 to 1). To query the time within the routine the micros() function can be used which returns the elapsed time from the start of the microcontroller in microseconds (µs) as a 32-bit number. This will be used as a timestamp to track events. The actual resolution of the timestamp may vary depending on the specific platform, with 4 µs for the currently used Arduino UNO. In addition, there may be some uncertainty in the time when exactly the interrupt routine would be executed but overall, the error will be less than 10 µs. This way the Arduino can be better than the previously used myDAQ. In addition, there may be some uncertainty in the time when exactly the interrupt routine would be executed, and the slew rate of the rising edge can also introduce timing errors. Depending on the rise time of the photogate signal, the error can be less than 10 µs this way the Arduino can be better than the previously used myDAQ.

Although we talk about a single-core system, because of the use of interrupts as a kind of time-sharing system, we have multiple processes running. Therefore, secure data transfer between processes is critical. First, the variables involved must be declared as volatile, so they will certainly not be cached in any register. Second, since a timestamp is 32 bit long, that is 4 bytes, its copy is not an atomic operation. Therefore, interrupts are disabled in the main program (loop) for the duration of the copy. Finally, data are transferred to the PC via USB serial port at a 115200 bit/s baud rate.

Program code

```
// Digital input for time measurement
#define INT_PIN 2
// Detected: Rising edge
#define TYPE RISING

volatile unsigned long timeIsr; // temporary variable to safely pass the timestamp value
unsigned long timeTmp; // temporary variable to store timestamp value temporarily
unsigned long timePrev = 0; // previous value of the timestamp
unsigned int timeOutCounter = 0; // timeout counter

/* Initialization, after starting the microcontroller, it runs once */
void setup() {
  pinMode(INT_PIN, INPUT); // Configuring digital input for time measurement
```

```
  attachInterrupt(digitalPinToInterrupt(INT_PIN), isr, TYPE); // Assigning interrupt to
digital input
  Serial.begin(115200); // Setting up Baud Rate of Serial port
  while (!Serial) {  ;  } // Waiting for connection (in case of USB devices with native USB
ports)

}

/* Main loop, repeated as an infinite loop. */
void loop() {
  noInterrupts(); // Denying interrupts in order to pass the timestamp value safely
  timeTmp = timeIsr; // Copying timestamp in the temporary variable
  interrupts(); // Allowing interrupts again
  if (timePrev != timeTmp) { // When the previous timestamp is not equal to the current one:
there is an event!
    timePrev = timeTmp; // Refreshing stored timestamp
    Serial.print(timeTmp); // Sending the timestamp
    Serial.println(","); // Sending the delimiter
  }

  delay(1); // Waiting for a little time
 }

/* Interrupt service routine. */
void isr() {
    timeIsr = micros(); // reading timestamp
 }
```

The above code is relatively simple, but it only performs time measurement. If a real-time value of the pressure is also required, the program must be supplemented with the measurement and transmission of the analogue signal. If you want to measure pressure even when the fidget spinner is not spinning, the data transmission protocol must be more complex. The Arduino is capable of controlling the motor by switching on or off a digital output depending on the control message received from the computer. The description of implementing these two tasks is beyond the scope of the article, but the code is available in the attachments of the article [24].

4.2.2. Receiving data from an Arduino using Excel Data Streamer

One of the development teams of Microsoft Garage created a free add-in for Microsoft Excel designed directly for educational purposes and for student measurements [25], [26]. Data Streamer can be downloaded from Microsoft's website or enabled in Office 365 among COM extensions. For the former, the add-in will appear in the header of the Excel file as 'Data Streamer'. The application should be customized according to our requirements. On the 'Connect a Device' tab, the appropriate communication port or device needed to be selected, in this case an Arduino UNO. Worksheets in Excel for Data in, Data Out, and Settings will be then automatically displayed. In the latter, you can, among others, set the time interval between incoming data to 10 ms-10 s, the number of channels, and choose whether the most recent data would be written in the last or first cell of the displayed data column. The default number of data rows that can be displayed is 25, which in the current version of Data Streamer can be increased to a maximum of 500. Then, in Advanced settings, the communication rate (baud rate) must be set in the same way as it is used in the measurement when its value is different from the default value of 9600 (in our case 115200).

The measurement can be started with the Start Data function. As a result, the data will be displayed in the Data In worksheet. The data can be stored by pressing the Start Recording and then the Stop Recording button, after which a pop-up window is displayed making it possible to save the data in CSV format. It can later be imported into a spreadsheet program for further evaluation. It may be useful for example to make the measured data available to the students and let them perform

numerical derivation (calculate the rate of change) as an individual task. Of course, when real-time evaluation of the measured data for classroom demonstration purposes is required, a standard cell reference on a new worksheet could be a useful tool. The problem, however, is that (in the current version of the add-in) the number of displayed data rows cannot be increased to more than 500, so when more useful pieces of data come in through the channels, finding another way to collect data is necessary.

We provided an obvious solution to the problem with the help of an Excel VBA macro. This short program monitors the cell in which the most recent data is displayed and when its value changes, exports the content of this cell and writes it at the bottom of a specified column. Thus, the collected timestamps are listed in chronological order in the corresponding column of our worksheet used for evaluation.

Excel VBA macro:

```
Private old As Long
Private Sub Worksheet_Change(ByVal Target As Range)
   Dim KeyCells As Range

   Set KeyCells = Range("B5")

If Not Application.Intersect(KeyCells, Range(Target.Address)) _
        Is Nothing Then

      v = Range("B5").Value Worksheets("Data In").Range("B5").Value
      If old <> v Then
         LstRw = ActiveCell.Row + 1
         Range("F" & LstRw).Select            Cells(LstRw, "F") = v
         old = v
      End If
End If
End Sub
```

Our Arduino measurement program, presented earlier, reads the event timestamps in microseconds. It is advisable to work in seconds and to fix t = 0 at the time the first data arrives (the first element of column A) by the formula

```
IF(A3=0;" ";(A3-$A$2)*10^(-6))
```

Excel can be pre-set to calculate the angle of rotation in another column of the evaluation worksheet by the formula

```
IF(A1=0;" ";(ROW(D1)-1)*2*PI()/3))
```

If a timestamp has already appeared in a given row, we add $\frac{2\pi}{3}$ (=120 °) to the value from the previous row.

A column is also devoted to numerical derivation, i.e. angular velocity calculation. By the

```
IF($A1=0;" ";2*PI()/3/($B2-$B1))
```

formula the first element of the column can be calculated.

Obviously, in addition to the current received timestamp, an undefined field will appear in the angular velocity column, which will be interpreted only when the next timestamp is received, this way the current value of the angular velocity is always determined in a step disadvantage.

By selecting the appropriate columns to create an XY-diagram that would be updated according to the available data, the rotation angle and the angular velocity over time can be monitored in real-time.

4.2.3. Receiving data from an Arduino using a LabVIEW program

The data sent by the Arduino board through serial port can also be received and processed by LabVIEW. The advantage of LabVIEW is that the program can be easily supplemented with further data processing. The created relatively simple LabVIEW program code with detailed comments is shown in Figure 6. The program constantly monitors the serial port and in case of message arrival tries to interpret it. If the message format is valid, it stores the data in an array (the timestamps of events and the time difference between events). The stored data can be displayed on a graph or saved to a file. The saved data can later be processed by any program. This program is available as a LabVIEW source code and as an executable that can be run without the LabVIEW development environment, as well. Extra functions mentioned before (continuous pressure measurement, motor control) can also be implemented in LabVIEW, the source code of which is attached to the article.
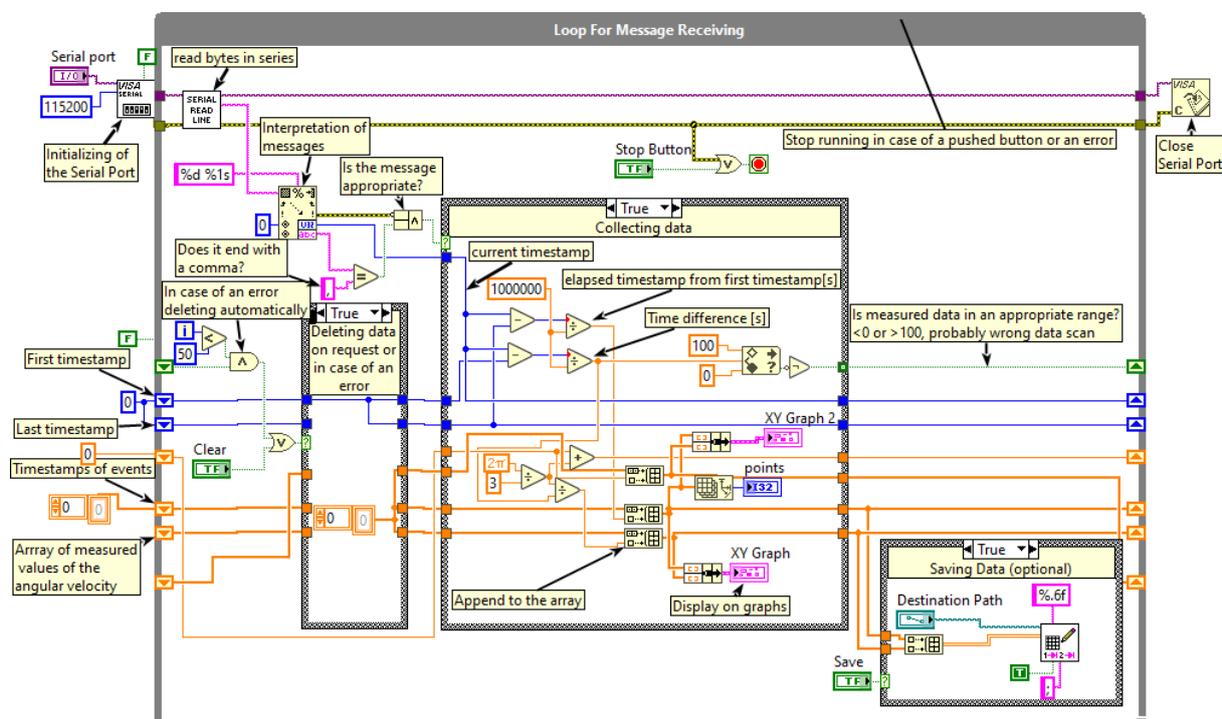

Figure 6: Reading data received through Serial Port in LabVIEW

4.2.4. Receiving data from an Arduino using a Visual C based program

Beside the two ways described in the previous parts of the article, with the aim of processing measurement data on a host PC coming from an Arduino, a self-developed program was developed in Visual C.

It receives, evaluates and stores CSV data in real-time coming from an Arduino board. In offline mode, it is capable of simulating the phenomenon and analysing it time after time.

5. Results and discussion

Based on our measurement results, our question in the problem identification might finally be answered. The graphs in Figure 7 display the rotation angle and the angular velocity of the fidget spinner as a function of time. The results of five measurements at normal atmospheric pressure and five control measurements at 17% vacuum (17 kPa) are compared. Starting points of the data series were aligned with each other so that the decrease from the 110 $\frac{rad}{s}$ angular velocity was plotted from the start time of t = 0 s.
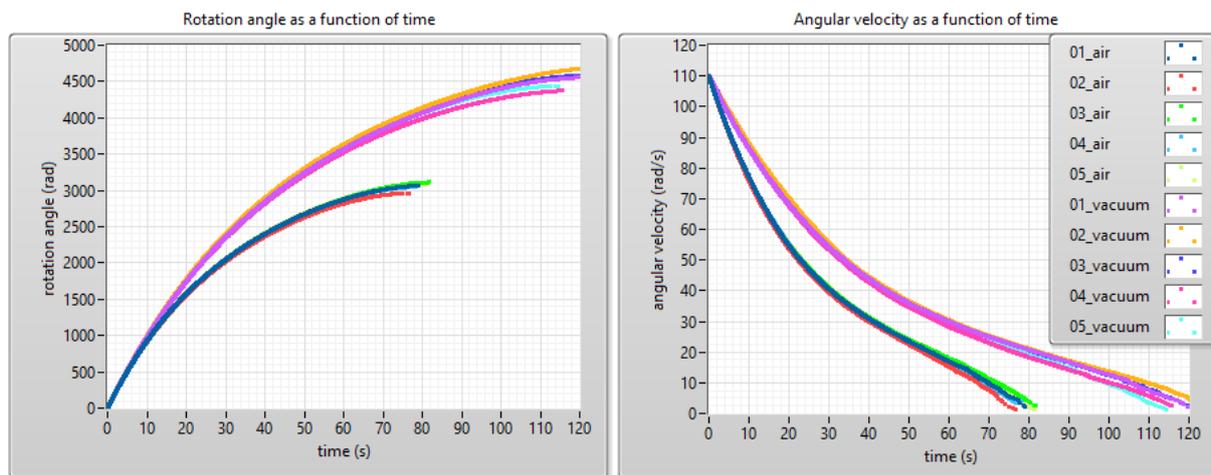

Figure 7: Results of our measurement in 17% vacuum and in normal air pressure

Comparing the graphs, it can be seen that starting from the same speed, free rotation takes about 40 seconds longer under vacuum. This suggests that air drag was involved in braking the movement of the toy. At the same time, there is a significant braking effect in vacuum, which is caused by the time-dependent friction of our bearing. The nonlinear decrease in angular velocity, even in vacuum, may indicate that friction is velocity-dependent, which may be a surprising statement for those who are aware of high school physics. Further measurements are needed to get around the problem correctly (see Chapter 6). Measuring the moment of inertia of the fidget spinner and examining the change in angular acceleration over time would reveal the supposed speed dependence of the friction torque, but other factors, such as the possible temporal change in operating temperature, may also be partly responsible.

6. Future research steps

In our paper we have presented a possible measurement method for solving wireless signal transmission with an optical signal, either for controlling circuits in a vacuum or for measuring the signals there. Our goal is to improve our system so that it is suitable for school use, that is why we need to take into consideration more issues that may arise.

The first is that in some schools vacuum chambers and pumps are already available, so only a pressure sensor is needed to be connected to the equipment. However, if the connectors do not meet current standards, the task becomes complicated. It may therefore be desirable to use an absolute pressure gauge (instead of a differential one) and to place it inside the vacuum chamber. Another possible aggravating circumstance is the effect of the chamber wall on the optical signal. Thick glass material partially absorbs the laser light, often with light scattering, and when the light path is not perpendicular to the glass interface, the glass refracts the beam of light. All these make

it difficult to detect the laser light and set the phototransistor properly. This problem can also be solved by placing a photogate inside the chamber.

However, if this is the case, it must be solved for both sensors to transmit their signal outside the vacuum chamber. Data transfer through electrical wires is not advisable as it would be required to prevent airflow at more places in the chamber. It is also a possible choice to place a microcontroller in the vacuum chamber so that the pressure sensor and photocell signal can be measured locally and even the motor circuit could directly be switched. Obviously, there is a need to choose a device that can wirelessly transmit measured data, such as an Arduino-compatible ESP32 chip capable of Bluetooth and Wi-Fi communication [27] .

We also plan to further validate the results. The bearing of the game can be replaced by one with known catalogue information, e.g. by an SKF bearing so that the radial and axial forces acting on the bearing during movement could be determined [28]. With this information available, according to a bearing model developed by SKF, the parameters of a given type of bearing can be used to calculate the frictional torque acting on a bearing as a function of its rotational speed. When the angular acceleration β (the second derivative of the rotation angle) from the remeasured kinematic properties of the modified fidget spinner would be determined, and the mass moment of inertia (θ) describing the mass distribution would be measured, their product, the total braking torque ($\sum M$) could be given as a function of speed according to the basic equation of rotational motion ($\sum M = \theta \cdot \beta$). This would allow comparison of theoretical and measured values.

7. Educational experiences

Previously three competitive essays were made at the Radnóti Miklós Experimental High School in Szeged, Hungary to take part in a competition arranged by Eötvös Loránd Physical Society and National Instruments Hungary Kft. with the aim of encouraging physics teachers and students to use and program a myDAQ device in LabVIEW [29]. The core of this study was also included in the project Repetition is the Mother of all Learning - Generation and Evaluation of Periodic Signals by Anikó Somogyi, Kornél Kovács and Katalin Balog. The authors of this study hold a weekly myDAQ course at school and a twice-weekly programming course. The presented measurement was carried out with a simpler experimental setup in smaller (three persons), then larger (ten persons) groups, and was used as a classroom demonstration experiment as well. Some of the members of the Research Group of Technical Informatics Methodology are high school teachers themselves. The team has published a number of tested measurements developed for Edaq530 and Arduino over the past decade [6], [15], [23], [30], [31], [32].

8. Conclusions

Technical informatics can create a natural connection between informatics and physics. In our experience, it is worth adopting measurements in the classroom as it is useful in teaching programming beyond its motivational role. In this article, we have provided effective solutions for performing a specific measurement and evaluating the data, which can be utilized for other measurements as well. In order to achieve this, the source codes are attached to the study.

Acknowledgement

Bibliography

1. *Micro:bit Educational Foundation.* https://microbit.org/ (Last retrieved 04/22/2020).
2. Y. Rogers, V. Shum, N. Marquardt, S. Lechelt, R. Johnson, H. Baker, M. Davies, *From the BBC micro to micro:bit and beyond: a British innovation.* Interactions, (2017) 24. 74–77. DOI: 10.1145/3029601.
3. *Arduino - Home.* https://www.arduino.cc/ (Last retrieved 11/04/2019).
4. M. El-Abd, *A Review of Embedded Systems Education in the Arduino Age: Lessons Learned and Future Directions.* International Journal of Engineering Pedagogy (IJEP), (2017) 7. 79. DOI: 10.3991/ijep.v7i2.6845.
5. *MISZAK.* http://www.inf.u-szeged.hu/miszak/ (Last retrieved 11/04/2019).
6. G. Makan, D. Antal, R. Mingesz, Z. Gingl, J. Mellár, G. Vadai, K. Kopasz, *Interdisciplinary Technical Exercises for Informatics Teacher Students.* Central-European Journal of New Technologies in Research, Education and Practice, (2019) 1. 21–34. DOI: 10.36427/CEJNTREP.1.1.382.
7. Z. Gingl, J. Mellár, T. Szepe, G. Makan, R. Mingesz, G. Vadai, K. Kopasz, *Universal Arduino-based experimenting system to support teaching of natural sciences.* Journal of Physics: Conference Series, (2019) 1287. 012052. DOI: 10.1088/1742-6596/1287/1/012052.
8. A. Helmenstine, *Steps of the Scientific Method.* Science Notes and Projects, (2014). https://sciencenotes.org/steps-scientific-method/ (Last retrieved 04/22/2020).
9. G. Gooday, *Precision Measurement and the Genesis of Physics Teaching Laboratories in Victorian Britain.* The British Journal for the History of Science, (1990) 23. 25–51. http://www.jstor.org/stable/4026801.
10. H. Gerola, R.E. Gomory, *Computers in Science and Technology: Early Indications.* Science, (1984) 225. 11–18. DOI: 10.1126/science.225.4657.11.
11. H. Zieris, H. Gerstberger, W. Müller, *Using Arduino-Based Experiments to Integrate Computer Science Education and Natural Science.* (2015).
12. L. Mesquita, G. Brockington, P.A. de Almeida, M.E. Truyol, L.A. Testoni, P.F.F. Sousa, *Using a fidget spinner to teach physics.* Physics Education, (2018) 53. 045024. DOI: 10.1088/1361-6552/aac69f.
13. R. Tanel, M. Büyükdede, A. Çoban, *Using fidget spinners in teaching some physics concepts.* Physics Education, (2019) 54. 035012. DOI: 10.1088/1361-6552/ab0808.
14. D. Macisaac, *Fidget spinner physics sites.* The Physics Teacher, (2017) 55. DOI: 10.1119/1.4999745.
15. Z. Gingl, R. Mingesz, P. Makra, J. Mellár, *Review of sound card photogates.* European Journal of Physics, (2011) 32. 905–924. DOI: 10.1088/0143-0807/32/4/006.
16. *Digital Signal Processing System - an overview | ScienceDirect Topics.* https://www.sciencedirect.com/topics/engineering/digital-signal-processing-system (Last retrieved 04/22/2020).
17. *MPX4250DP20 Datasheet, PDF - Alldatasheet.* https://www.alldatasheet.com/ (Last retrieved 11/05/2019).
18. *KY-008 Laser Transmitter Module - ArduinoModulesInfo.* https://arduinomodules.info/ky-008-laser-transmitter-module/ (Last retrieved 11/05/2019).

19. *Phototransistor - an overview | ScienceDirect Topics.*
https://www.sciencedirect.com/topics/engineering/phototransistor
(Last retrieved 04/22/2020).
20. *NI myDAQ User Guide - National Instruments.* http://www.ni.com/pdf/manuals/373060g.pdf.
21. *Learn LabVIEW: intro to graphical programming in NI LabVIEW - National Instruments.*
https://www.ni.com/getting-started/labview-basics/ (Last retrieved 11/05/2019).
22. R. Mingesz, Z. Gingl, Measurement and data acquisition. http://eta.bibl.u-szeged.hu/899/1/2011_0104_szte_4_measurement_supplement.pdf.
23. Z. Gingl, G. Makan, J. Mellar, G. Vadai, R. Mingesz, *Phonocardiography and Photoplethysmography With Simple Arduino Setups to Support Interdisciplinary STEM Education.* IEEE Access, (2019) 7. 88970–88985. DOI: 10.1109/ACCESS.2019.2926519.
24. *Fidget spinner – attachment files | MISZAK.*
http://www.inf.u-szeged.hu/miszak/en/fidgetspinner_cejntrep/.
25. *Education | DataStreamer.*
https://www.microsoft.com/en-us/education/hackingstem/datastreamer
(Last retrieved 04/22/2020).
26. A. Teiermayer, *Improving students' skills in physics and computer science using BBC Micro:bit.* Physics Education, (2019) 54. 065021. DOI: 10.1088/1361-6552/ab4561.
27. *ESP32 Datasheet.*
https://www.alldatasheet.com/datasheet-pdf/pdf/1148023/ESPRESSIF/ESP32.html
(Last retrieved 04/22/2020).
28. *Using a friction model as an engineering tool.* Evolution Online,.
https://evolution.skf.com/us/using-a-friction-model-as-an-engineering-tool-2/
(Last retrieved 04/22/2020).
29. *MyDAQ Pályázat Fizikatanároknak* (In Hungarian). http://sukjaro.eu/ELFT-NI-palyazat/
(Last retrieved 11/05/2019).
30. Z. Gingl, K. Kopasz, *A high-resolution stopwatch for cents.* Physics Education, (2011) 46. 430–432. https://doi.org/10.1088/0031-9120/46/4/009.
31. K. Kopasz, P. Makra, Z. Gingl, *Edaq530: a transparent, open-end and open-source measurement solution in natural science education.* European Journal of Physics, (2011) 32. 491–504.
DOI: 10.1088/0143-0807/32/2/020.
32. Z. Gingl, K. Kopasz, G. Makan, R.Z. Mingesz, J.Z. Mellár, T. Szépe, G. Vadai, *Műszaki informatikai megoldások a modern középiskolai oktatásban* (in Hungarian), in: szte, Zamárdi, 2017 Terjedelem: 10 p. DOI: 10.6084/m9.figshare.6554567 (Last retrieved 11/04/2019).

Authors

SOMOGYI Anikó
University of Szeged, Faculty of Science and
Informatics, Department of Technical
Informatics, Radnóti Miklós Experimental
High School of Szeged, Hungary,
e-mail: somogyia@inf.u-szeged.hu

KELEMEN András
University of Szeged, Juhász Gyula Faculty of
Education, Department of Applied
Information Technology, Radnóti Miklós
Experimental High School of Szeged,
Hungary,
e-mail: kelemen.andras.felix@szte.hu

MINGESZ Róbert
University of Szeged, Faculty of Science and
Informatics, Department of Technical
Informatics, Hungary,
e-mail: mingesz@inf.u-szeged.hu

MELLÁR János
University of Szeged, Faculty of Science and
Informatics, Department of Technical
Informatics, Hungary,
e-mail: mellar@inf.u-szeged.hu

About this document

License