Science Friction

<Author> Ilia Mestvirishvili
<Author> David Shapakidze

<Info>

<Keywords> friction force, stopping distance, anti-lock braking system (ABS), app programming, data acquisition

<Disciplines> physics, computer science, mathematics

<Age level of the students> 14+

<Hardware> Arduino[1], servo, motor, Bluetooth module, motor shield, photogate

<Language> Arduino programming environment[2], AppInventor[3], Snap4Arduino[4], Blockly[5]

<Programming level> easy, medium

<Summary>

An investigation of the factors affecting the force of friction can be turned into an interesting and entertaining experiment by building a low-cost Bluetooth controlled car with a simple braking system. This will enable the students to observe real data such as the car's speed (absolute value of velocity) before applying the brakes, its stopping distance, and how the mass of a car and the type of surface affect friction force. The students will then conduct experiments to investigate the relationship between the factors affecting the stopping distance with sufficient precision to check their own hypotheses or those suggested by the teacher.

<Conceptual introduction>

Friction is a very important force in everyday life and is taught in physics at both middle school and high school level. However, traditional experiments related to the topic of friction are limited and not much fun. This project will turn the exploration of friction into an exciting group project which involves:

1. building and fine-tuning a car
2. programming an Arduino[1] microcontroller to measure instantaneous speed and stopping distance
3. programming a mobile phone using AppInventor[3] to send, receive and display real data on the phone screen

<What the students/teachers do>

Since these three tasks can initially be done simultaneously, we recommend that the teacher divides his/her class into groups of two to three students, who then work on the tasks separately, but come together to discuss and revise their work.

When friction is introduced in a particular curriculum, the teachers could offer the unit about friction in conjunction with this project, which will motivate the students and improve their understanding of the related theoretical concepts. The best way to start the project would be to pose the following questions to your students and give them time to brainstorm and come up with their own ideas, predictions or hypotheses:

- What is the relationship between the speed of a car and the stopping distance? (Answers might vary of course, i.e. 'the stopping distance is proportional to the speed before applying the brakes', or some might 'remember' from physics that the stopping distance is actually proportional to the square of the speed.)
- How would increasing the mass of a car affect the stopping distance, provided that everything else is unaffected? (A popular answer is that increasing the mass should increase the stopping distance.)
- How should we apply the brakes to stop a car as quickly as possible? (Possible answers: the best way is to stop the wheels completely; if we make them rotate in the opposite direction to the motion, this will stop the car more quickly; etc.)
- If both the front and rear brakes are identical, will they stop the car at the same time? (The students can reflect on their own experience with bicycles.)
- Any other questions that might come from the teacher or the students.

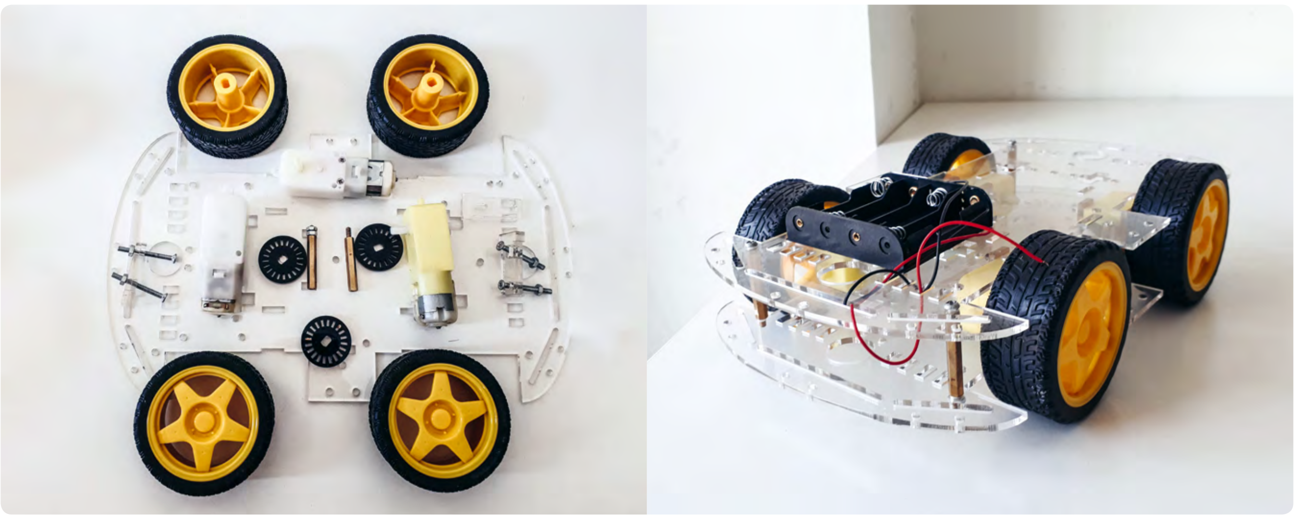
After the students have written down their initial ideas, the next step will be to think about how to build a simple car and the data that they will require to check and further develop their initial ideas. The teacher can facilitate this process and suggest that the students build a car that is able to collect and send relevant data to a phone, which in turn can control the car as well as receive and display the data.

Based on their interests, skills and preferences, at this stage of the project students can split into the previously mentioned groups if desired. However, just one group of students can do all these tasks as well. The next steps of the project development will be the same for either scenario.

<Building a chassis and mounting the electrical components>

This approach involves building a car from the low-cost and readily available Arduino[1] chassis kit depicted in 📷1. Teachers and students are encouraged to try different approaches to design and implementation, for example, different ways to collect and send data, and control the car remotely as well as different software and languages to write the necessary code.

After assembling the car and deciding where to mount the Arduino controller and the motor, the students will need to think of different ways to measure the speed of the car. The recommended approach is to brainstorm and give the students an



1: A complete chassis kit

opportunity to propose some ideas of their own. With appropriate help from the teacher, the best and easiest way to do this is to use a photogate to count the rotation rate of a free rear wheel. In this way, the students will be able to measure both the instantaneous velocity and the distance travelled. This can be done by just using the materials in the Arduino 4 wheel chassis kit, or developed separately if the teacher prefers to do so.

Some maths will be required here to transform the number of blocking events counted by the photogate into speed or distance. The kit includes a disc with 22 holes in it and a wheel with a diameter of 5.1±0.1 cm. It is not difficult to calculate that 1 pulse from the photogate, which means a wheel rotated 1/22 of a full rotation, corresponds to a distance of $d = 0.72$ cm. At the same time, the photogate measures and sends a time interval t in milliseconds between consecutive pulses. Instantaneous speed can be calculated by dividing 0.72 cm by this time interval.

The following steps can be used, irrespective of whether the students are working in different groups or only in one. A single group would work through all the steps one after the other, while different groups would split the three tasks.

<Arduino programming>

The Arduino programming group will work on the coding using the following approach:

1. Define the actions and consequently the methods or functions to collect and send the required data via Bluetooth.
2. Write and test each method separately.
3. Put everything together.

Beginners could start with TinkerCad[6], which allows online Arduino circuit design and simulation, thus preventing burnouts and short circuit problems at the prototyping stage.

The following sections outline each part in more detail:

1. The required actions are: starting and stopping a motor, applying and releasing brakes, measuring the distance, measuring the speed, sending and receiving the data via Bluetooth.
2. The crucial part here is to write a code to measure the speed and distance travelled during the same experiment. Both of them use pulses from a photogate and are activated when '2' is received from the phone app via Bluetooth:
   - To measure the distance, there is a counter which starts counting pulses sent to the Arduino from a freely rotating rear wheel after the brakes are applied to the front wheels.
   - The time intervals between pulses are used to measure the instantaneous speed. A rear wheel turns 0.72 cm during one pulse, so this needs to be divided by the time interval between pulses.
   - The functionality of an anti-lock braking system (ABS) could be implemented by applying the brakes and releasing them once for a variety of time intervals between 50 and 200 ms (optimised experimentally), which in most cases leads to a shorter stopping distance.
3. When you put together a program for Arduino, you need to ensure that everything happens in one big loop. Therefore, if the program is interrupted at any particular step, it will affect all the following steps.

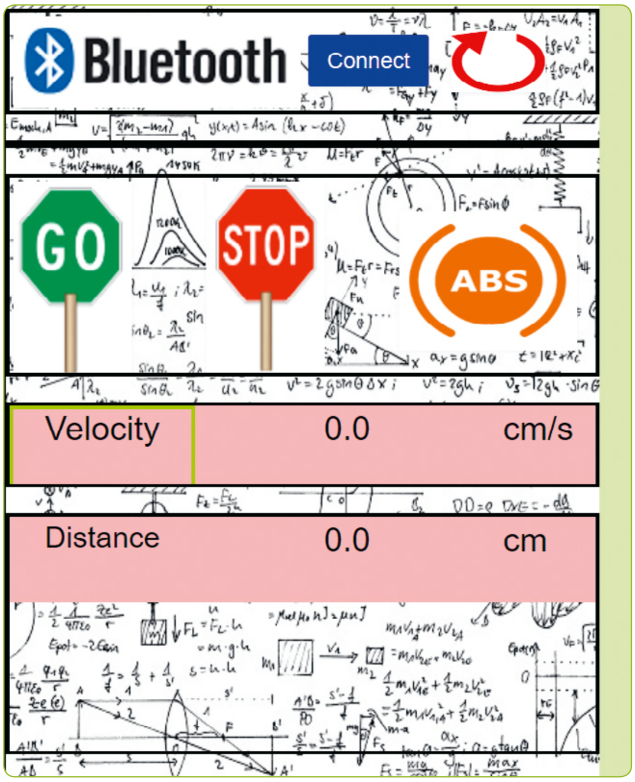
The sample code and references to other sources for each of these functionalities are available online[7] but, with a little help from their teachers, the students should be encouraged to try and write their own code.

<Android programming>

The Android programming group can explore AppInventor[3] and think of ways to display the data on the screen (user interface, UI). The students will decide where and how to arrange the buttons to control the car as well as the panels and labels to display the data on a screen (📷2). The code of the AppInventor program is provided online[7] and has the following functionalities:

1. Pushing the **START** button sends ’1‘ to the Arduino[1] via Bluetooth and starts the motor in the car.
2. Pushing the **STOP** button sends ’2‘ to the Arduino via Bluetooth, which stops the motor and then applies the brakes.
3. Pushing the **ABS** button sends ’3‘ to the Arduino via Bluetooth, which stops the motor and then applies the brakes at regular intervals (simulating an ABS feature).
4. After pushing the **STOP** button or the **ABS** button, the data received about the instantaneous speed before stopping and the distance that the car has travelled after applying the brakes, i.e. the stopping distance, will be displayed in two corresponding panels with the ’speed‘ and ’distance‘ labels respectively.
5. The **RESET** button sends ’0‘ to the Arduino via Bluetooth, clears the speed and distance data, and then resets the Arduino.

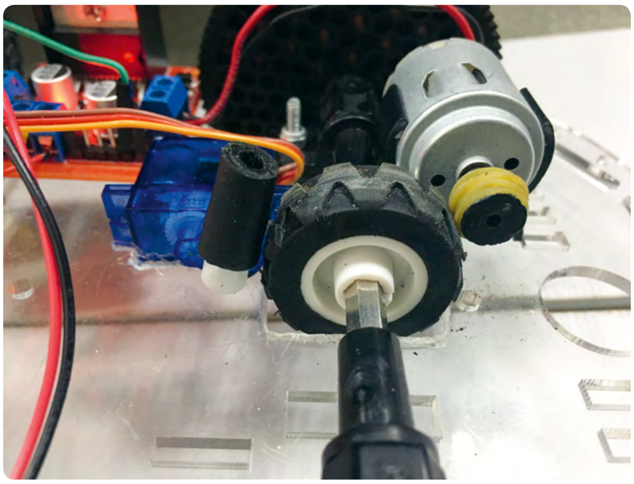
We recommend that you use the provided code as a reference for teachers and give the students the opportunity to explore AppInventor[3] and write their own code based on the above functionalities.



📷 2: User interface of the app

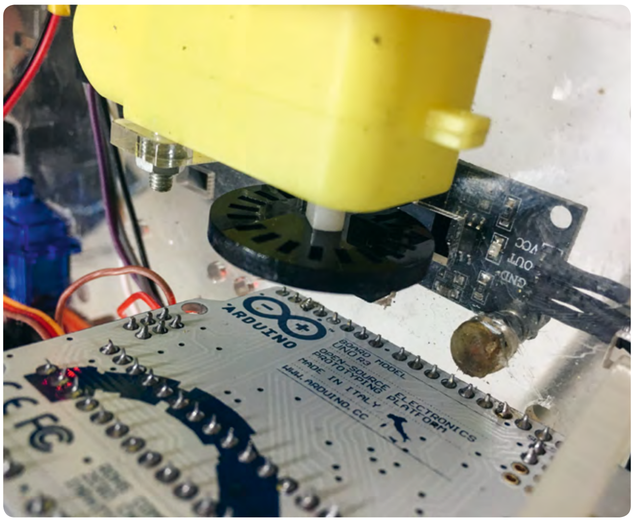
<Car building>

The car building group will be asked to think of appropriate locations and ways to attach components like the motor, photogate, servo, batteries, Bluetooth module, motor shield and, finally, the Arduino board itself. When the servo turns, it is important that it pushes a handle with a rubber tube firmly to the rotating disk, which is mounted on the front wheel axis (📷3).



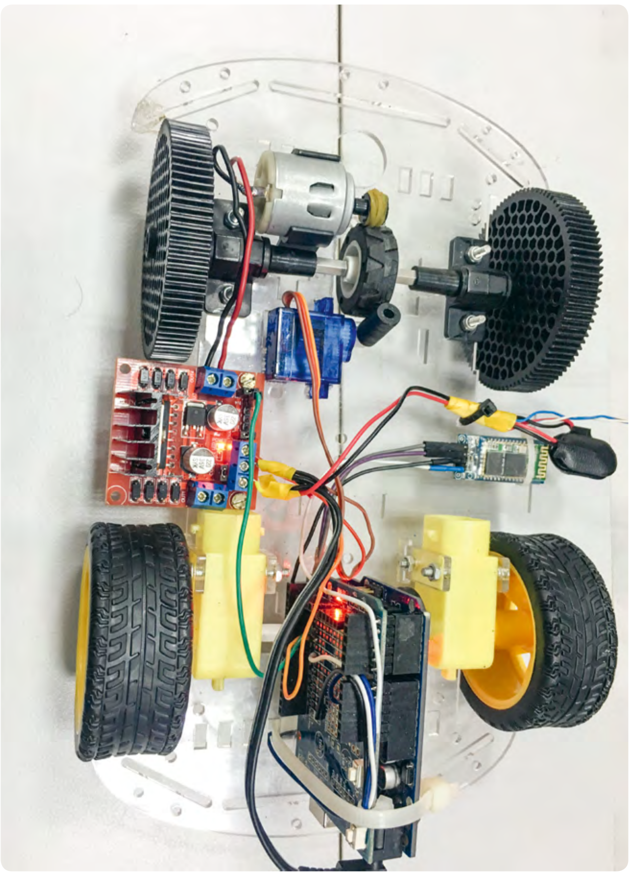
📷 3: A close-up of the brake system

It can provide enough force to stop the wheels immediately. The position of the photogate is equally important. Please ensure that it counts all the pulses correctly – the recommended photogate Arduino sensor has a built-in LED, which blinks when something enters the photogate gap. For this reason, please make sure that the photogate counts the rotations correctly when the rear wheels rotate (📷4).



📷 4: The photogate

Also, please ensure that the rear wheels rotate as freely as possible at all times. Remember that we calculate speed and distance using the free rotation of the rear wheels. Finally, the car should look similar to the example depicted in 📷5 once it has been built.



5: The assembled car

We recommend that you keep detailed guidelines as a reference and give students the opportunity to brainstorm and implement their own solutions.

<Conclusion>

This project is a fun way for students to learn core physics concepts like force of kinetic and static friction, and, at the same time, relatively modern technology like ABS, where physics, electronics, programming and design come together to explore factors affecting the stopping distance of a car. It is always a challenge to interpret and analyse real experimental data. It involves key concepts such as uncertainty, validity, reproducibility and visualisation. The project allows the students to experience and understand force of friction in a hands-on, thought-provoking lesson.

<Cooperation activity>

This project offers great potential for collaboration, since its three independent elements – the design of the car, the coding of the Arduino[1] and the coding with AppInventor[3] – could be further developed and improved. All the cooperating partners would benefit from each other's contribution to any of these components.

Another option for cooperation could be a competition between school teams based on who could make a car with the same wheels and the same mass stop more quickly, provided that the 'road' surface and speed before stopping are the same.

Many thanks to our Greek colleagues: Astrinos Tsoutsoudakis for providing important suggestions about the physics involved in this project and Georgios Georgoulakis for his extremely useful coding tips. We would also like to acknowledge the comprehensive feedback and support of Jörg Gutschank, which made this project more interesting and reproducible.

<References>

[1] www.arduino.cc
[2] www.arduino.cc/en/Guide/Environment
[3] http://appinventor.mit.edu
[4] http://snap4arduino.rocks/
[5] https://developers.google.com/blockly/
[6] www.tinkercad.com/circuits
[7] All additional materials are available at www.science-on-stage.de/coding-materials.

<Imprint>

<Taken from>
Coding in STEM Education
www.science-on-stage.eu/coding

<Published by>
Science on Stage Deutschland e.V.
Am Borsigturm 15
13507 Berlin, Germany

<Revision and Translation>
Translation-Probst AG

<Design>
WEBERSUPIRAN.berlin

<Illustration>
Rupert Tacke, Tricom Kommunikation und Verlag GmbH

<Credits>
The authors have checked all aspects of copyright for the images and texts used in this publication to the best of their knowledge.

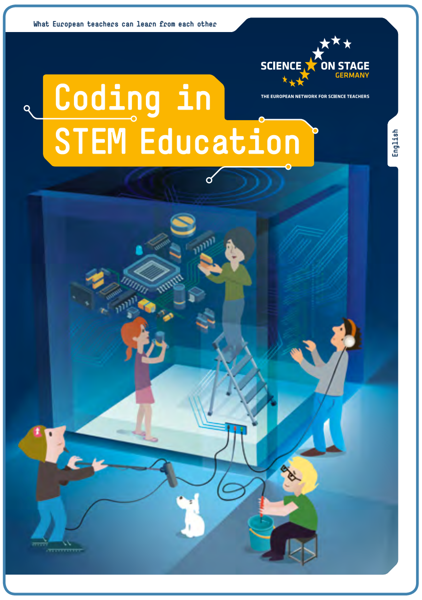
<Please order from>
www.science-on-stage.de
info@science-on-stage.de

<ISBN PDF>
978-3-942524-58-2

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License: https://creativecommons.org/licenses/by-sa/4.0/.

CC BY SA

First edition published in 2019
© Science on Stage Deutschland e.V.



Science on Stage – The European Network for Science Teachers

… is a network of and for science, technology, engineering and mathematics (STEM) teachers of all school levels.
… provides a European platform for the exchange of teaching ideas.
… highlights the importance of science and technology in schools and among the public.

The main supporter of Science on Stage is the Federation of German Employers' Association in the Metal and Electrical Engineering Industries (GESAMTMETALL) with its initiative think ING.

Join in – find your country on
www.science-on-stage.eu
www.facebook.com/scienceonstageeurope
www.twitter.com/ScienceOnStage

Subscribe for our newsletter
www.science-on-stage.eu/newsletter

A project by
SCIENCE ON STAGE GERMANY

Main supporter of Science on Stage Germany
think ING.
Die Initiative für Ingenieurnachwuchs

Proudly supported by
SAP