SCENARIO

Title	CAR ACCELERATION	
Summary	This Activity consists of calculating the speed of acceleration of an object in a given time. Therefore, visually, the student helps them to pose the problem and better understand what is being asked.	
Author/s	AIJU	

DIDACTIC OBJECTIVES

- Teach math in a different and attractive way.
- Teach how to calculate times based on speed and distance.

Physics **X** Mathematics **X** Information Technology □ Robotics □ Programming □

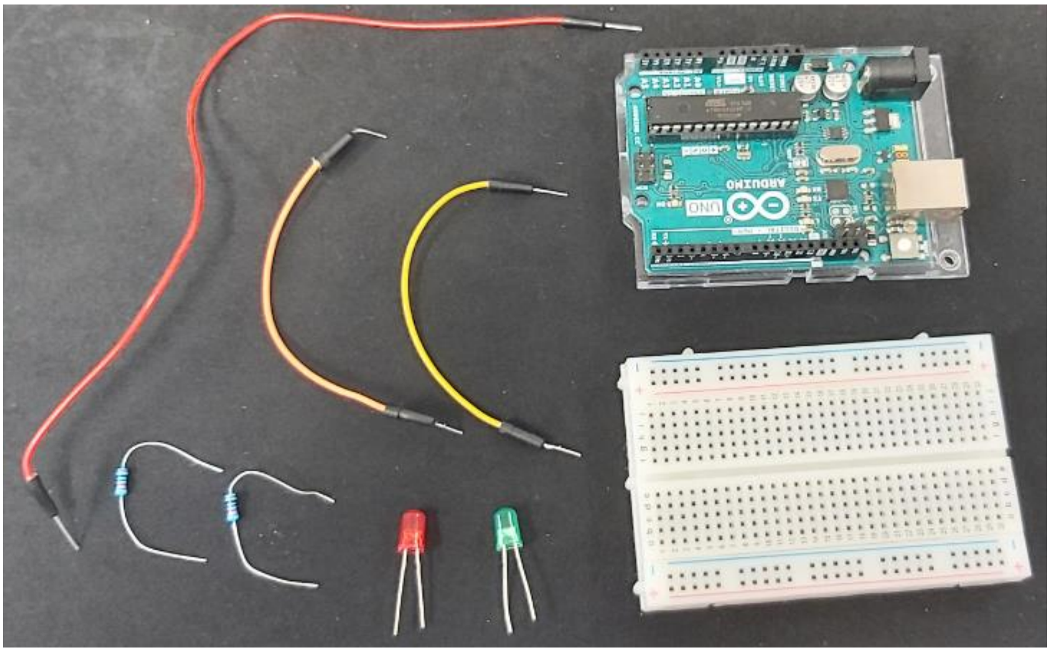
Education Level: 10-12 years □ 12-14 years **X**

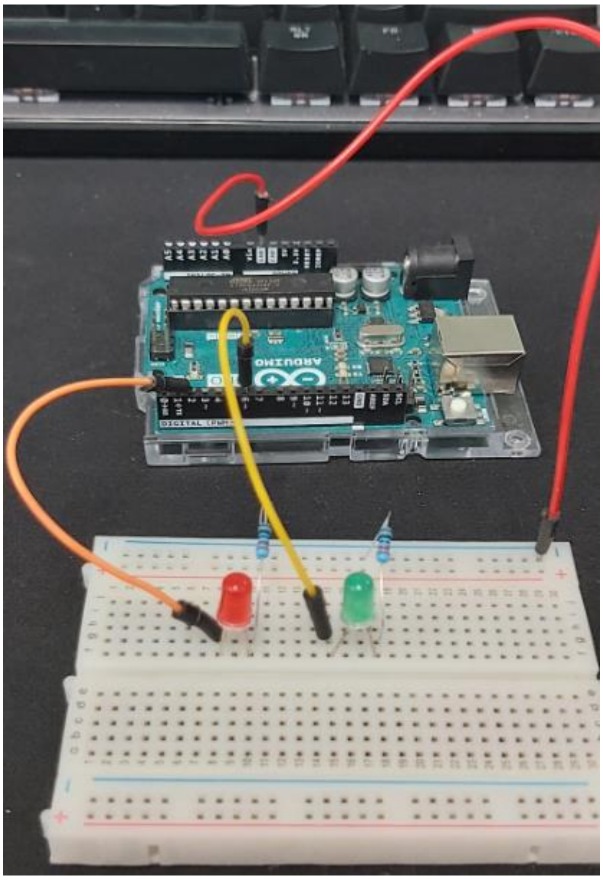
PROBLEM STATEMENT

Some students have problems understanding the concept of what the mathematical problem asks, so through a visual example it is intended to facilitate learning and understanding.

BOM (Bill Of Materials needed)

- Arduino Device
- Board
- (x2) Leds (Red and Green)
- (x3) Cables
- (x2) Resistors

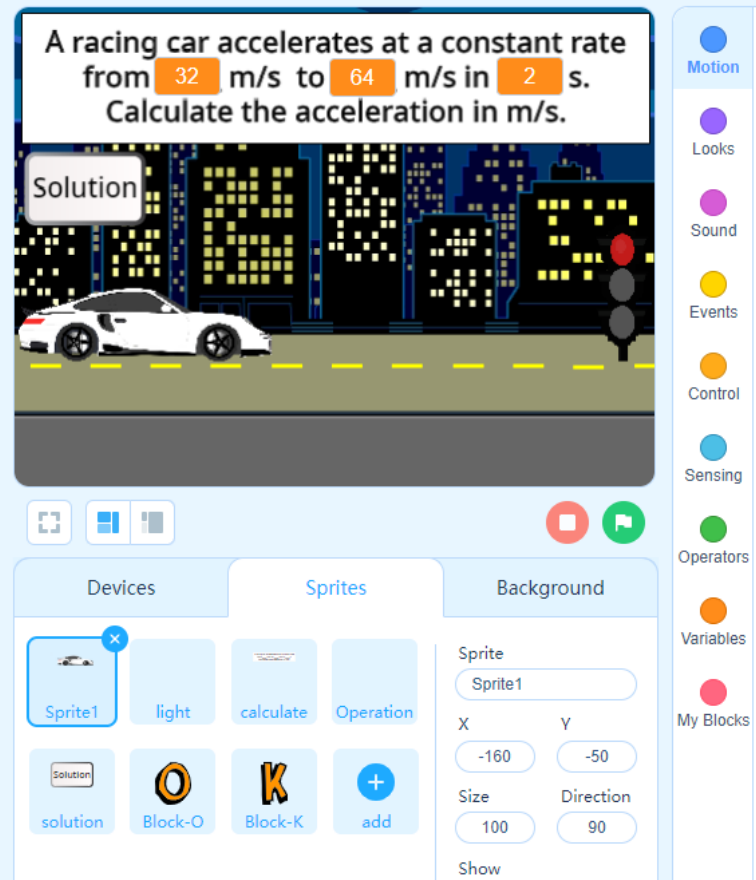




ACTIVITY DESCRIPTION

For the development of the activity, we will use software that allows us to unify the game developed in Scratch with the use of the Arduino board. In this case, we have used the mBlock software: (https://mblock.makeblock.com/en-us/)

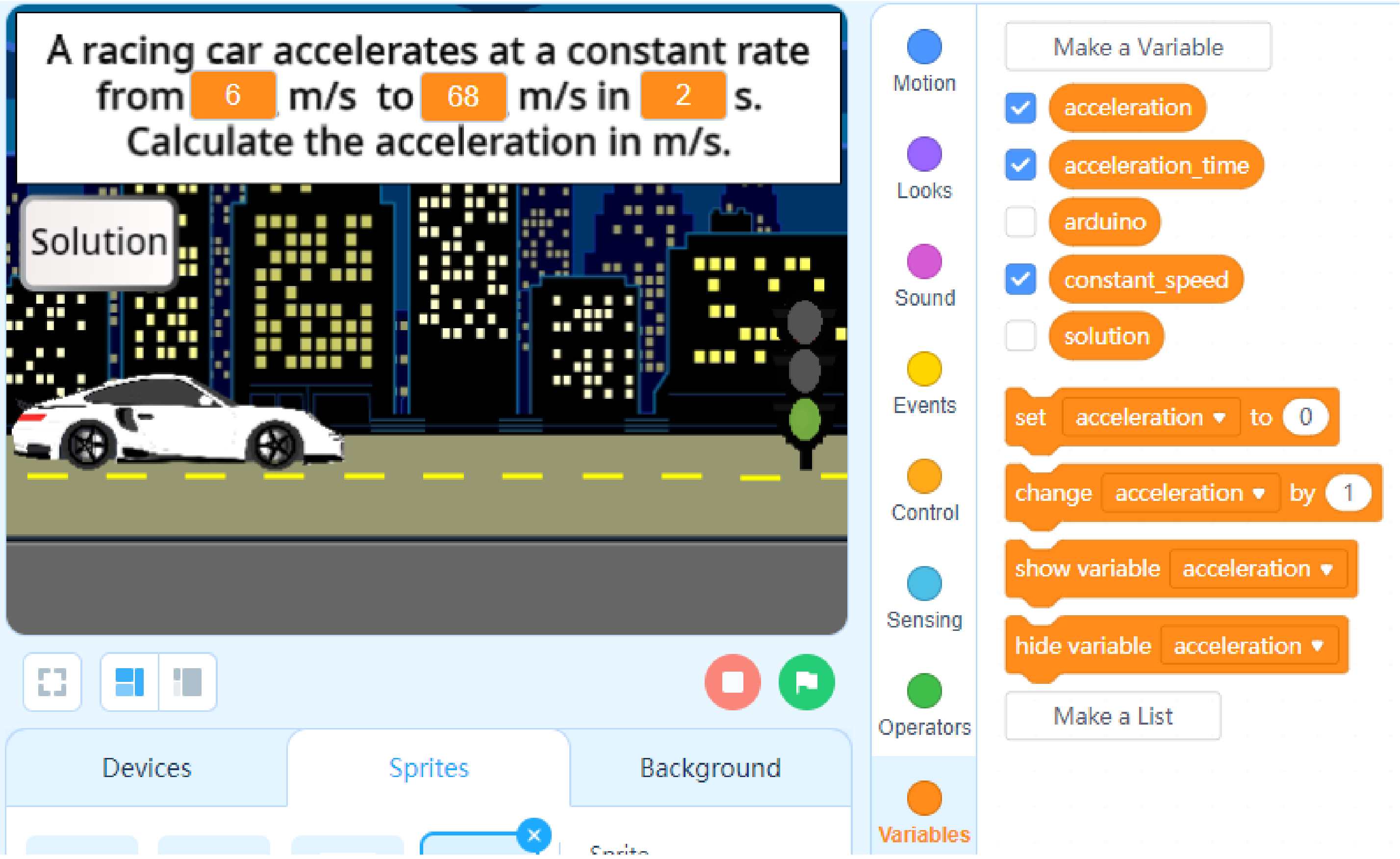
First of all, we will make the graphic composition of the activity:



A racing car accelerates at a constant rate from 32 m/s to 64 m/s in 2 s. Calculate the acceleration in m/s.

Sprite: Sprite1
X: -160 Y: -50
Size: 100 Direction: 90
Show

We add the “acceleration”, “acceleration_time” and “constant_speed” variables for the problem statement:



A racing car accelerates at a constant rate from 6 m/s to 68 m/s in 2 s. Calculate the acceleration in m/s.

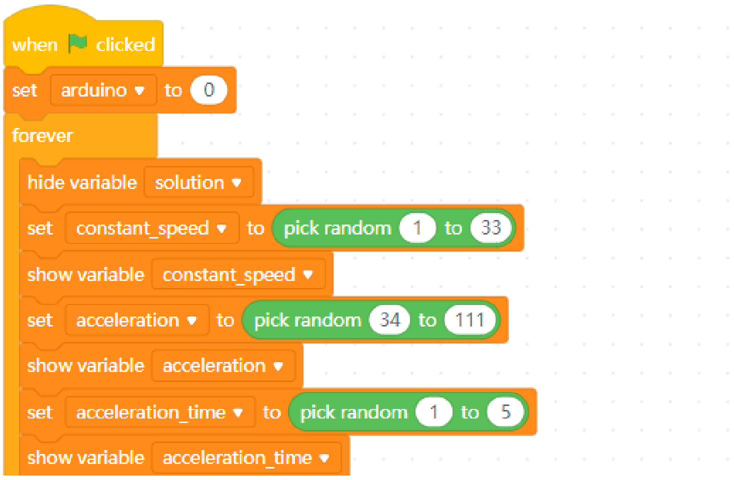
☑ acceleration
☑ acceleration_time
□ arduino
☑ constant_speed
□ solution

set acceleration ▾ to 0
change acceleration ▾ by 1
show variable acceleration ▾
hide variable acceleration ▾

In addition, we will create the "Solution" variable, which will be the one that calculates the solution to the problem, and the "Arduino" variable, which will be in charge of sending to the Arduino board when and that the corresponding LED lights up.

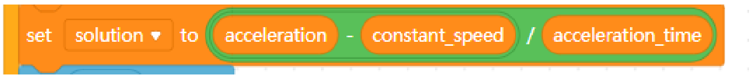
Once we have the graphic composition and the variables created, we will start with the programming:

1. We will start by setting the visual background that we want to appear while the mathematical/physics problem is being posed and we will set the “Arduino” variable to 0, so that the LEDs are off. In addition, for the “acceleration”, “acceleration_time” and “constant_speed” variables, random values will be created, so that whenever the Activity starts, different values come out:



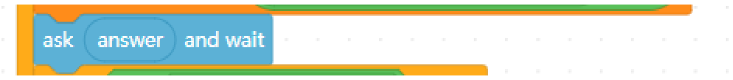
```
when ⚑ clicked
set arduino ▾ to 0
forever
    hide variable solution ▾
    set constant_speed ▾ to pick random 1 to 33
    show variable constant_speed ▾
    set acceleration ▾ to pick random 34 to 111
    show variable acceleration ▾
    set acceleration_time ▾ to pick random 1 to 5
    show variable acceleration_time ▾
```

2. Then, we apply the formula that will calculate the problem:

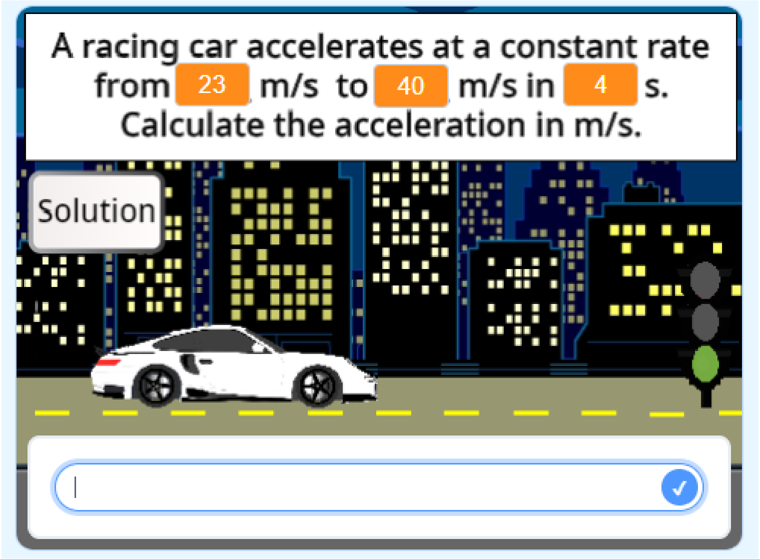


```
set solution ▾ to ((acceleration) - (constant_speed)) / (acceleration_time)
```

3. Next, the system wait for the user to write the result:

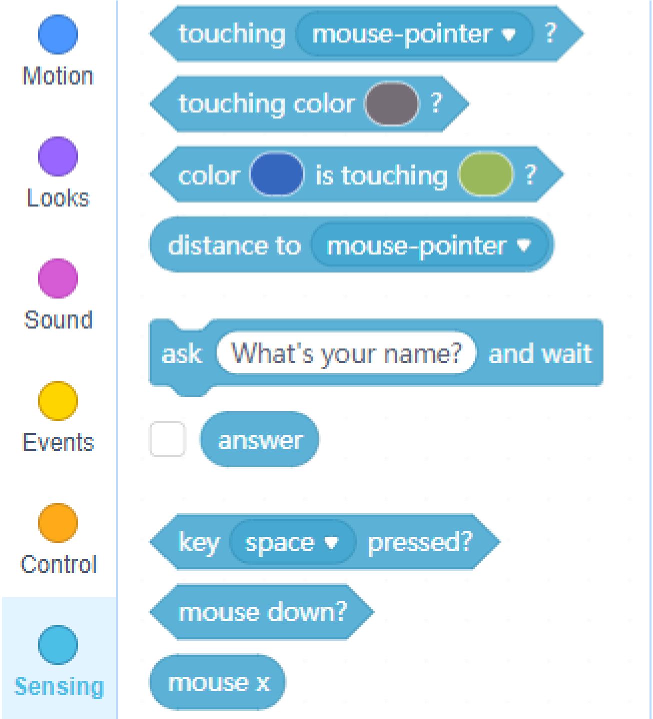


```
ask answer and wait
```

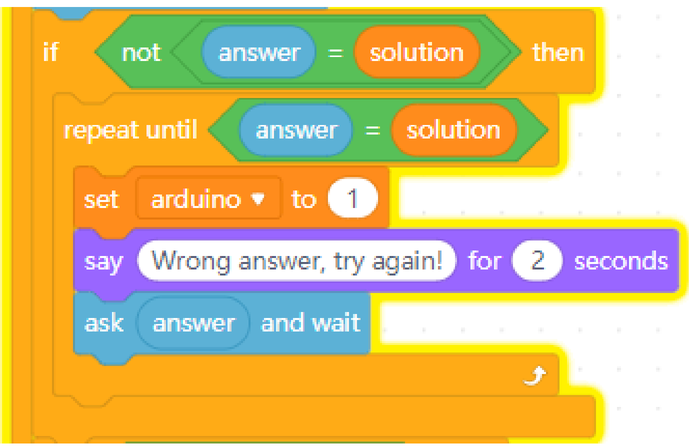


A racing car accelerates at a constant rate from 23 m/s to 40 m/s in 4 s. Calculate the acceleration in m/s.

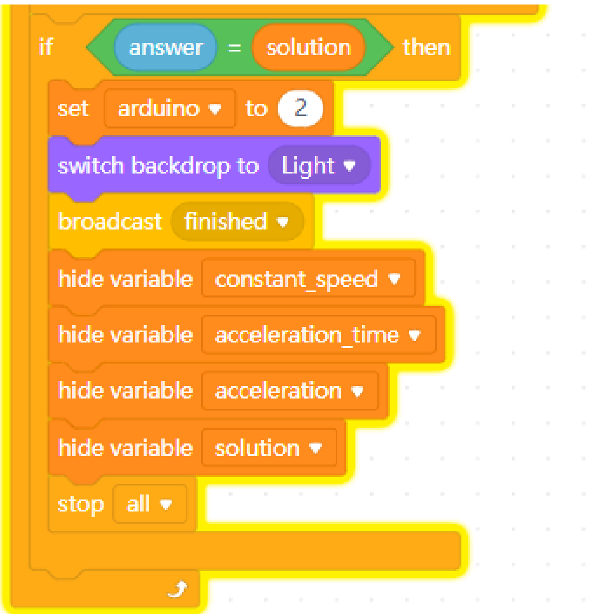
4. The result indicated by the user will be saved in a variable called “answer”, which will be created in the “Sensing” section:



5. Once the answer has been inserted, the programming will check to verify the result:

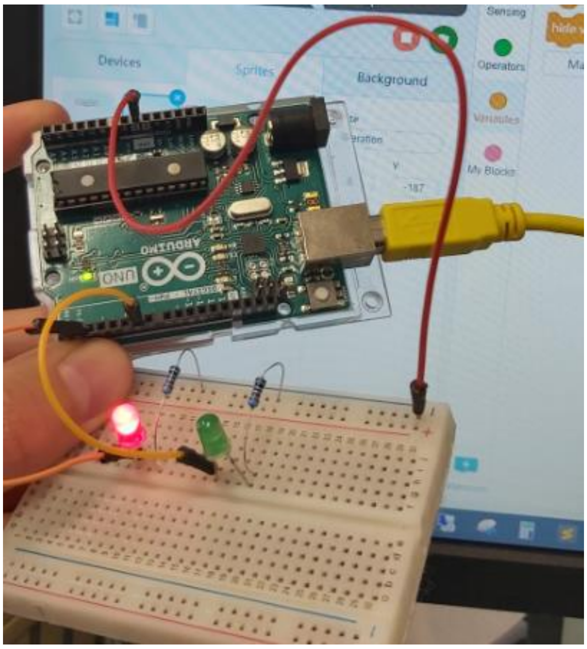


```
if not answer = solution then
    repeat until answer = solution
        set arduino ▾ to 1
        say Wrong answer, try again! for 2 seconds
        ask answer and wait
```

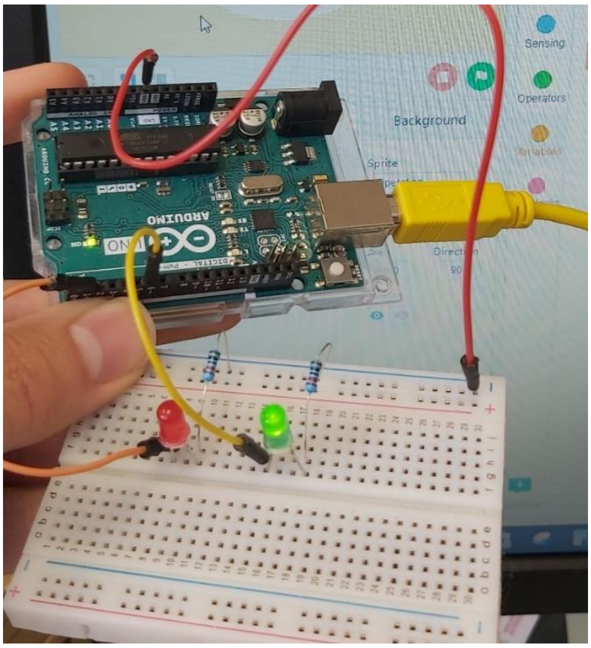


```
if answer = solution then
    set arduino ▾ to 2
    switch backdrop to Light ▾
    broadcast finished ▾
    hide variable constant_speed ▾
    hide variable acceleration_time ▾
    hide variable acceleration ▾
    hide variable solution ▾
    stop all ▾
```

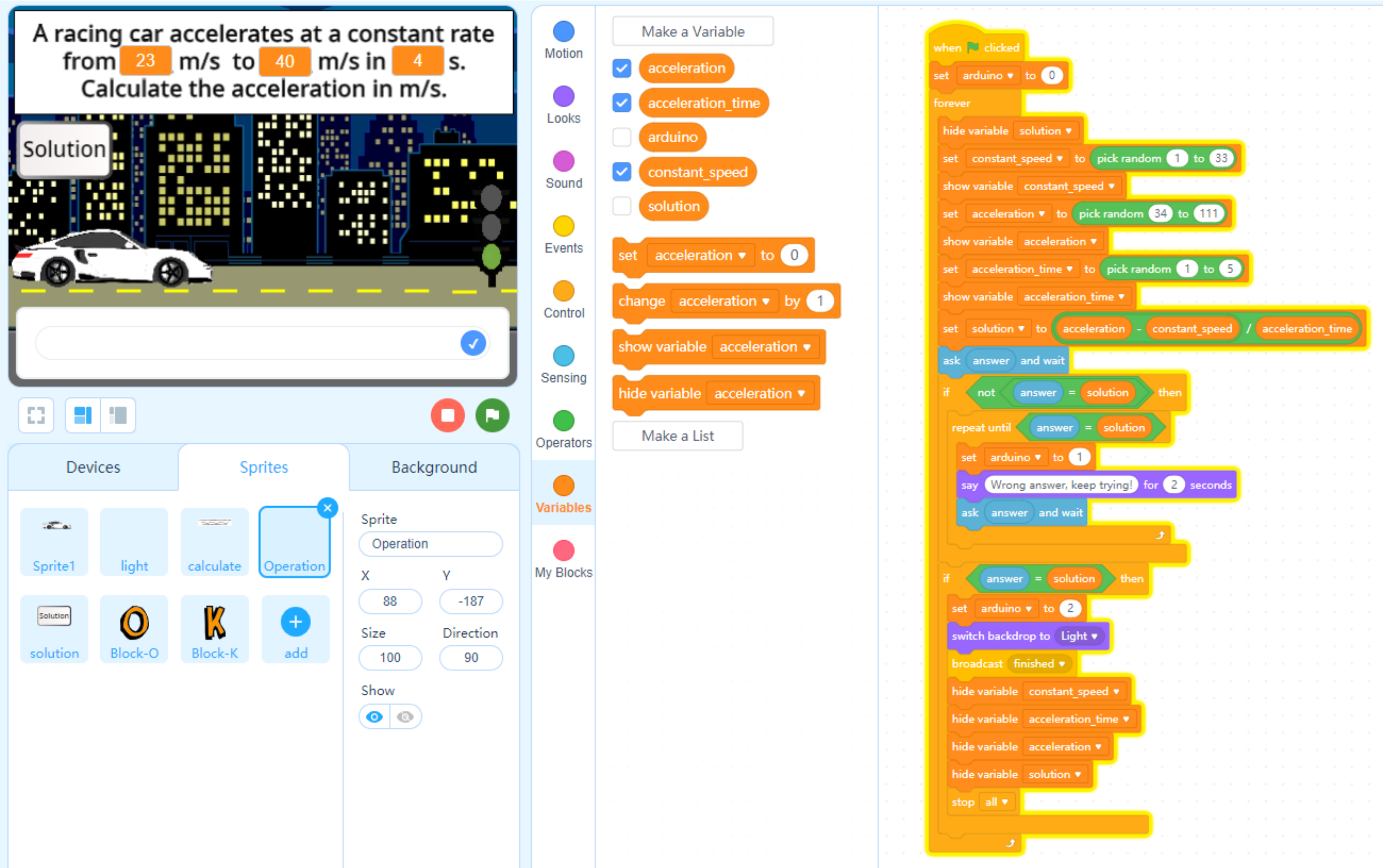
6. If the entered result is wrong, the game will display "Wrong answer, try again!" and will prompt the user to retype the answer. In this block the Red LED of the Arduino will light:



7. If, on the contrary, the user types the correct answer, the Green LED will light up and the variables will be hidden and the background will change:



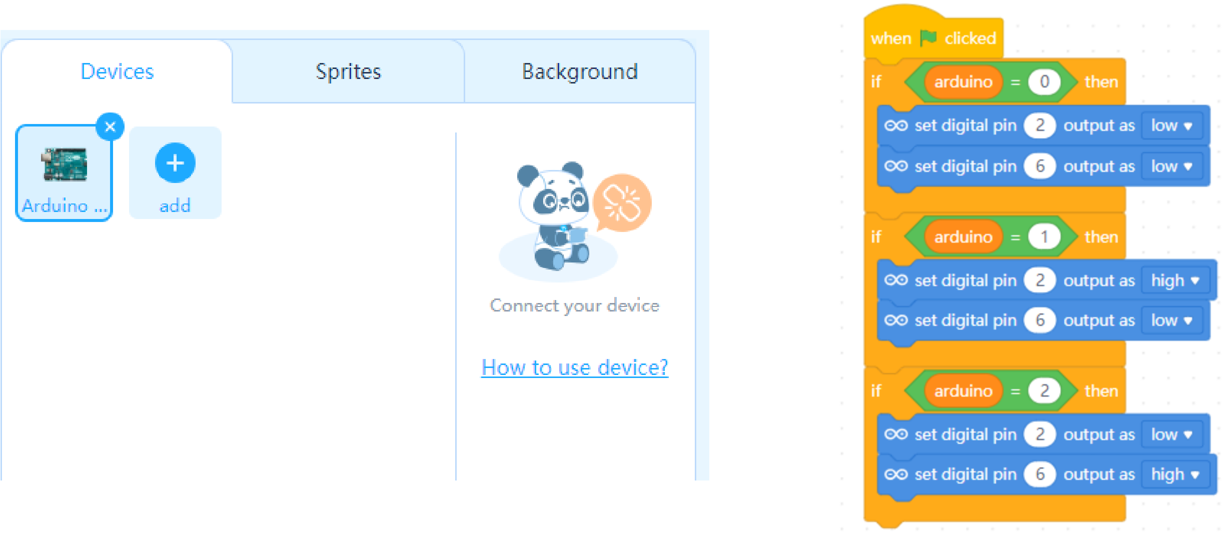
8. The main Code would be as follows:



A racing car accelerates at a constant rate from 23 m/s to 40 m/s in 4 s. Calculate the acceleration in m/s.

```
when ⚑ clicked
set arduino to 0
forever
    hide variable solution
    set constant_speed to pick random 1 to 33
    show variable constant_speed
    set acceleration to pick random 34 to 111
    show variable acceleration
    set acceleration_time to pick random 1 to 5
    show variable acceleration_time
    set solution to acceleration - constant_speed / acceleration_time
    ask answer and wait
    if not answer = solution then
        repeat until answer = solution
            set arduino to 1
            say Wrong answer, keep trying! for 2 seconds
            ask answer and wait
    if answer = solution then
        set arduino to 2
        switch backdrop to Light
        broadcast finished
        hide variable constant_speed
        hide variable acceleration_time
        hide variable acceleration
        hide variable solution
        stop all
```

9. The code that would be in charge of lighting the LEDs on the Arduino board would be as follows:



```
when ⚑ clicked
if arduino = 0 then
    set digital pin 2 output as low
    set digital pin 6 output as low
if arduino = 1 then
    set digital pin 2 output as high
    set digital pin 6 output as low
if arduino = 2 then
    set digital pin 2 output as low
    set digital pin 6 output as high
```

STUDENTS’ EVALUATION
The way to evaluate the students would be for them to demonstrate on paper how they have developed the exercise and to use the program / game to check the solutions.
SCALABILITY
Regarding the concept of scalability, the complexity could be increased by adding problems that need more parameters, such as a third component in which they have to be found, or determining the speed of an element so that they meet in a certain second.