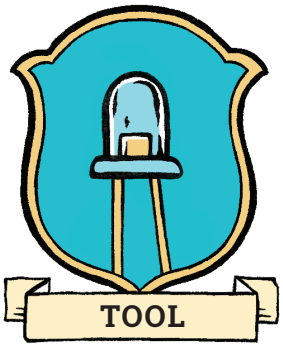
ARDUINO IN 6 CHALLENGES

A collection of sheets to understand how to use Arduino
and to be able to run original projects:
incredible machines, take measurements.
The only limit is your imagination.

Six “Challenge” sheets to discover the essentials of Arduino boards



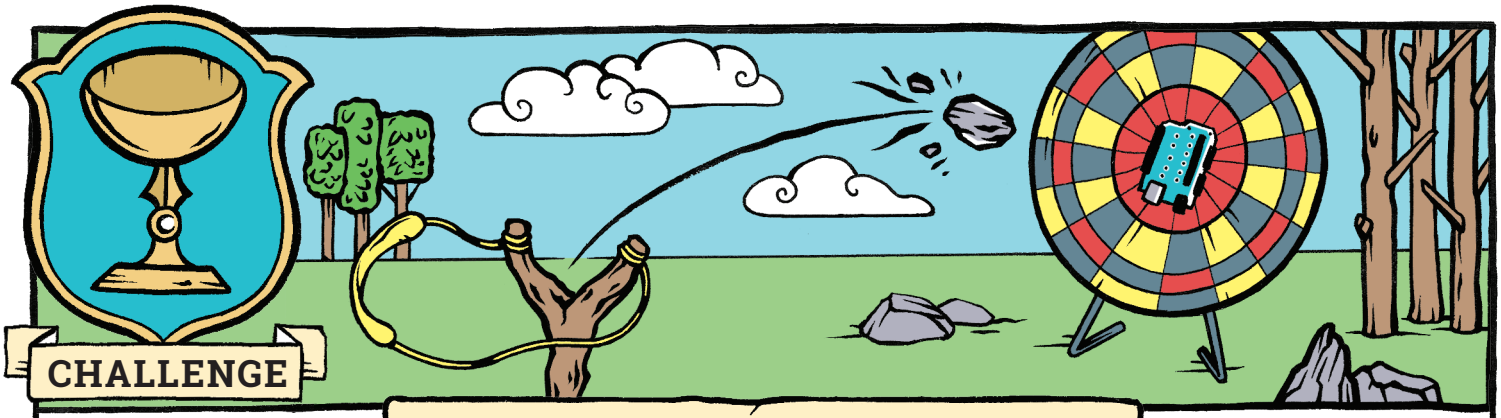
“Tool” sheets to help you get to know the equipment

“Knowledge” sheets to expand on the basic concepts

Development: **Marine Joumard**
in collaboration with **Frédéric Bouquet** and **Julien Bobroff**,
team **La Physique Autrement**, **Université Paris-Sud**

– find it online at: **www.opentp.fr** –

The images of the circuits were created with the open-source software Fritzing



CHALLENGE

TEST YOUR BOARD

Make your board flash to check that everything works!

EASY

NECESSARY HARDWARE

– an Arduino board
– a USB cable
– a computer
(necessary for the Challenges)

ASSOCIATED SHEETS

Arduino board
Arduino IDE software

INSTALL THE ARDUINO SOFTWARE

You need software that allows dialogue between your computer and your Arduino board. For this, go to the reference site **www.arduino.cc**, and click on the tab "software". The Arduino IDE is open and free; install it using the installer that corresponds to your computer.

CONNECT YOUR ARDUINO BOARD

Once the installation is over, use the USB cable to connect your Arduino board to your computer. The board must be recognized by the computer. If it is, the Arduino IDE software should recognize your board. To test this, open the software and open the "Tools" menu. There are two settings to check in this menu, the type of board and the port. The setting "board" must correspond to the board you are using (Uno). The setting "port" must correspond to the port that the board is connected to (for example "COM 11: Arduino/Genuino Uno" if you are using Windows or "\dev\tty. usbmodem … (Arduino Uno)" on an Apple computer. If these two settings are not configured correctly, communication cannot be established.

SEND A PROGRAM TO THE BOARD

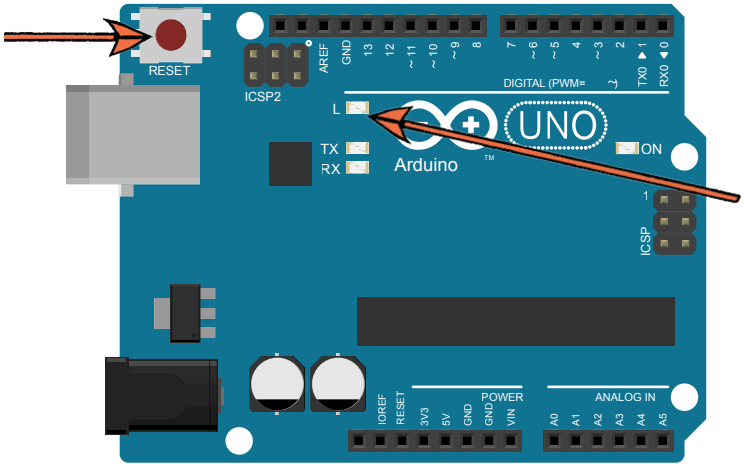
The programs are written on the computer then sent to the Arduino board. To check the connection with your board, you will send a test program. Open the File menu; choose Examples, Basics and the program "Blink".

```
// the setup function runs once when you press reset or power the board
void setup() {

  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
}
```

UPLOAD

Send the program on the Arduino board by clicking on the "Upload" icon of the software. Your computer will transform the program into Arduino understandable instructions and send them to the board through the USB cable. Once installed on your board (this may take several seconds), the program will execute on repeat.

Board reset button. It resets the board and restarts the program that is installed on it.

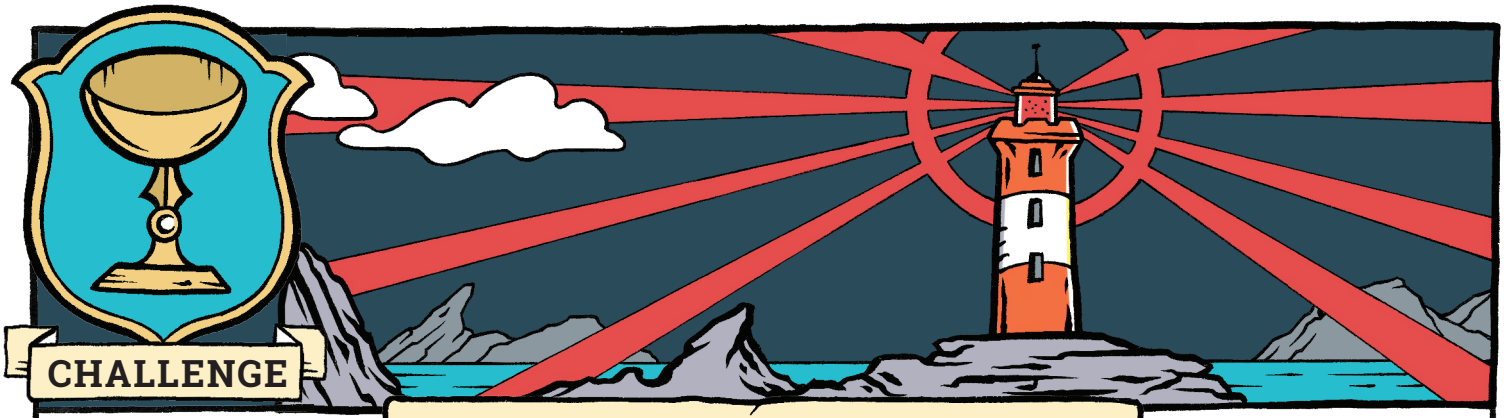


Small test light-emitting diode (LED). This LED is connected to digital port 13 on the majority of Arduino boards.

This program makes the test LED flash. If the LED on your board is flashing: well done! Everything is fine, you have succeeded in communicating with your board.

IN CASE OF PROBLEMS

– Check that you have correctly configured the type of Arduino board and the COM port used by the software (Tools menu);
– Disconnect the board, stop the Arduino software then reconnect the board before restarting the program. If it still doesn't work, restart your computer and begin again from step 1.



CHALLENGE

LIGHT UP A LED

You will control light!

EASY

NECESSARY HARDWARE

- une LED rouge
- une résistance de 220 ohms
- un breadboard
- des petits fils électriques

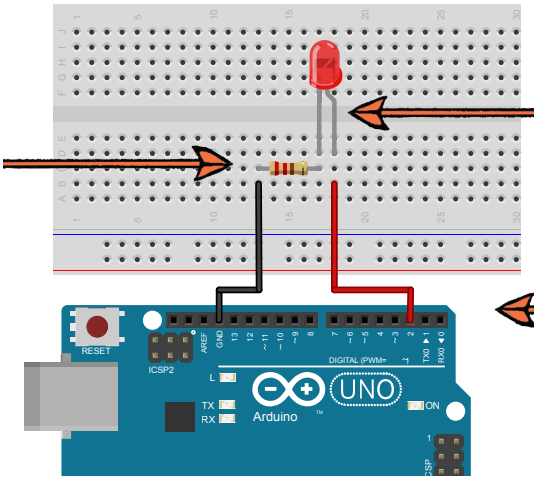
ASSOCIATED SHEETS

Test your board
Digital outputs

COMPLETE THIS ELECTRICAL SETUP

Fully insert the different components and wires into the breadboard to obtain a reliable electrical contact.

This resistor prevents the sending of a too powerful voltage into the LED and blowing it.



Be careful of the direction when connecting the LED: the longest wire is connected with the red wire (towards the +5 V voltage).

The numbered connectors from 0 to 13 are the digital ports: you can force a voltage of 5 V or cancel it on each of the ports.

COPY THIS PROGRAM

HIGH: this means that you send 5 V to the wire connected to port 2.

LOW: this means that you send 0 V.

```
// setup to initialize the board

void setup() {                  // beginning of setup
  pinMode(2, OUTPUT) ;          // initialize digital port 2 as output
}                               // end of setup

// this loop will continue indefinitely
void loop() {                     // start of loop
  digitalWrite(2, HIGH) ;         // switch on LED
  delay(1000) ;                   // wait one second (1000 msec)
  digitalWrite(2, LOW) ;          // switch off LED
  delay(1000) ;                   // wait one second (1000 msec)
}                                 // end of loop
```

The text in grey is a commentary text that is ignored by the computer. It is not necessary to copy it but it allows a better understanding of the program.

A similar program is easily accessible through the software menu (File menu, Examples, Basics, program Blink).

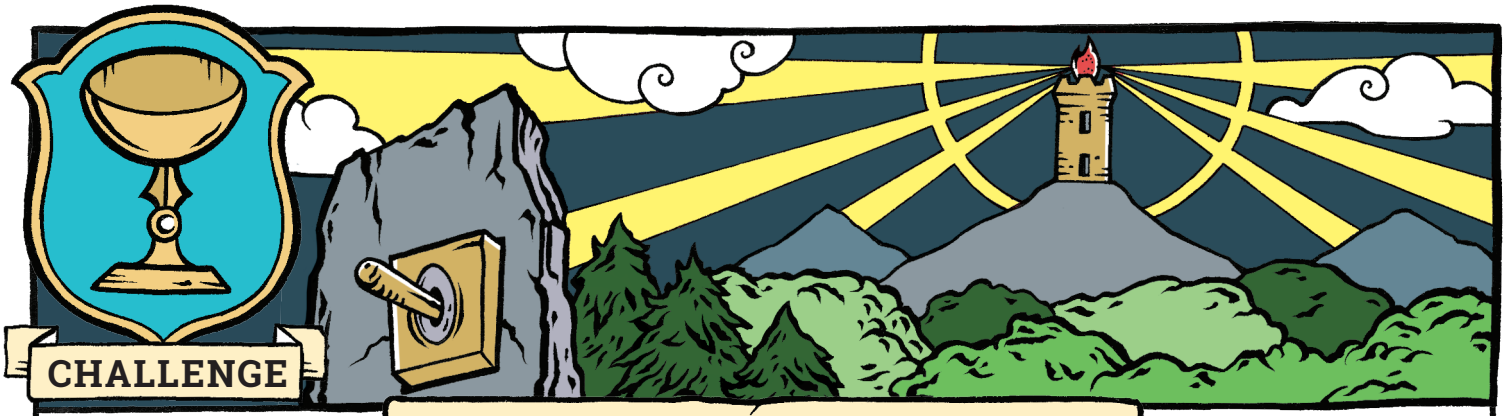
If nothing happens, it means there is a problem! Check your electrical setup and the connection between your computer and the Arduino board.

TAKING IT FURTHER

Try changing the blinking speed. For this, modify the delay() instructions in your program and observe what happens when you upload the new program. Try switching on another LED connected, this time, to port 3.

THE ULTIMATE CHALLENGE!

Modify your program so that your light sends an SOS in Morse code!



CHALLENGE

USE A SWITCH

You will be configuring and flipping a switch!

EASY

NECESSARY HARDWARE

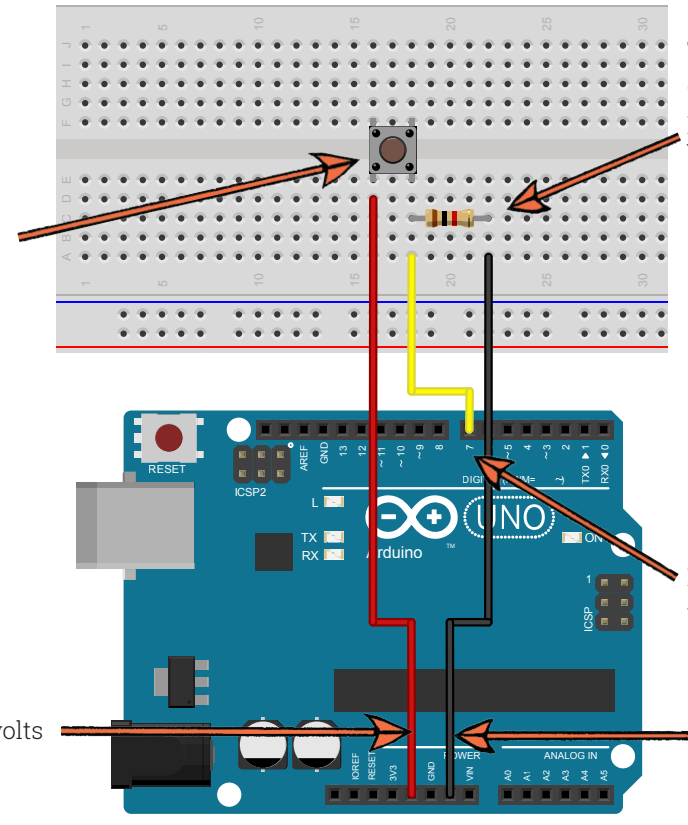
– a switch
– a 1000 ohms resistor
– a breadboard
– small electrical wires
– possibly a multimeter

ASSOCIATED SHEETS

Test your board
Digital inputs

COMPLETE THIS ELECTRICAL SETUP

The switch in the basic Arduino kit has four contacts. The two contacts used in the circuit must have zero resistance when you press the switch and an infinite resistance otherwise. If you have a multimeter, use it to check this. If not, follow the assembly guide below and reverse the switch if it does not work.



When you press the switch, the voltage on the yellow wire is 5 volts. When you don't press it, the voltage is zero because of the resistor that connects the yellow wire to the ground. Without this resistor, the voltage would remain undetermined.

COPY THIS PROGRAM

The digitalRead(7) instruction orders the board to read the voltage of port 7. There are two possible results:
HIGH (5 volts) or LOW (0 volts).

This instruction asks the board to send a message to the computer through the USB cable.

```
// setup to initialize the board
void setup() {         // start of setup
  pinMode(7, INPUT) ; // initialize digital port 7 as input
  Serial.begin(9600) ;// initialize serial communication with computer
}                      // end of setup

// this loop will continue indefinitely
void loop() {                    // start of loop
  if (digitalRead(7) == HIGH) {  // if one measures 5 V on digital input 7
    Serial.println(" ON ! ") ;   // one sends " ON! " to computer
  }
  else {                         // if one does not measure 5 V
    Serial.println(" Off ... ") ; // one sends " Off ... "  to the computer
  }
  delay(100) ;                   // one sends " Off ... "  to the computer
}                                // end of loop
```

A similar program is easily accessible through the software menu (File menu, Examples, Basics, program DigitalReadSerial).

UPLOAD

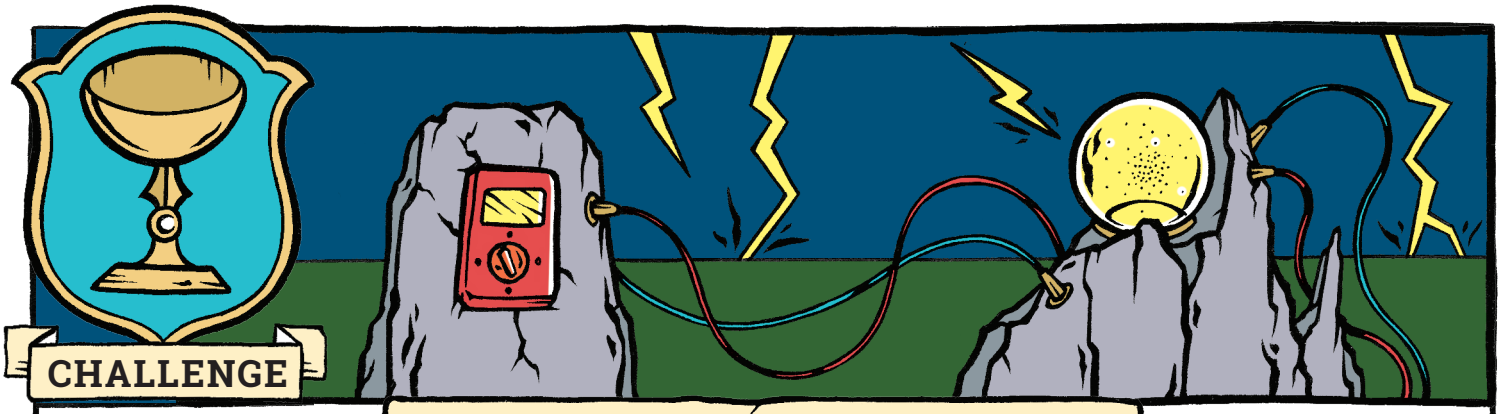
You must read the messages that the board sends to the computer. To do this, use the serial monitor (in the Tools menu of the Arduino software). If strange characters appear, check the connection speed of the serial monitor, which should be the same as the one used to initialize communication in the program (9600 bauds). Now press the switch and watch what happens on the serial monitor.

TAKING IT FURTHER

Swap the black and red wires and observe what happens.

THE ULTIMATE CHALLENGE!

Modify your program so that the test LED on the Arduino board flashes when you press the switch.



MEASURE VOLTAGE

You will measure voltage with your board!

EASY

NECESSARY HARDWARE

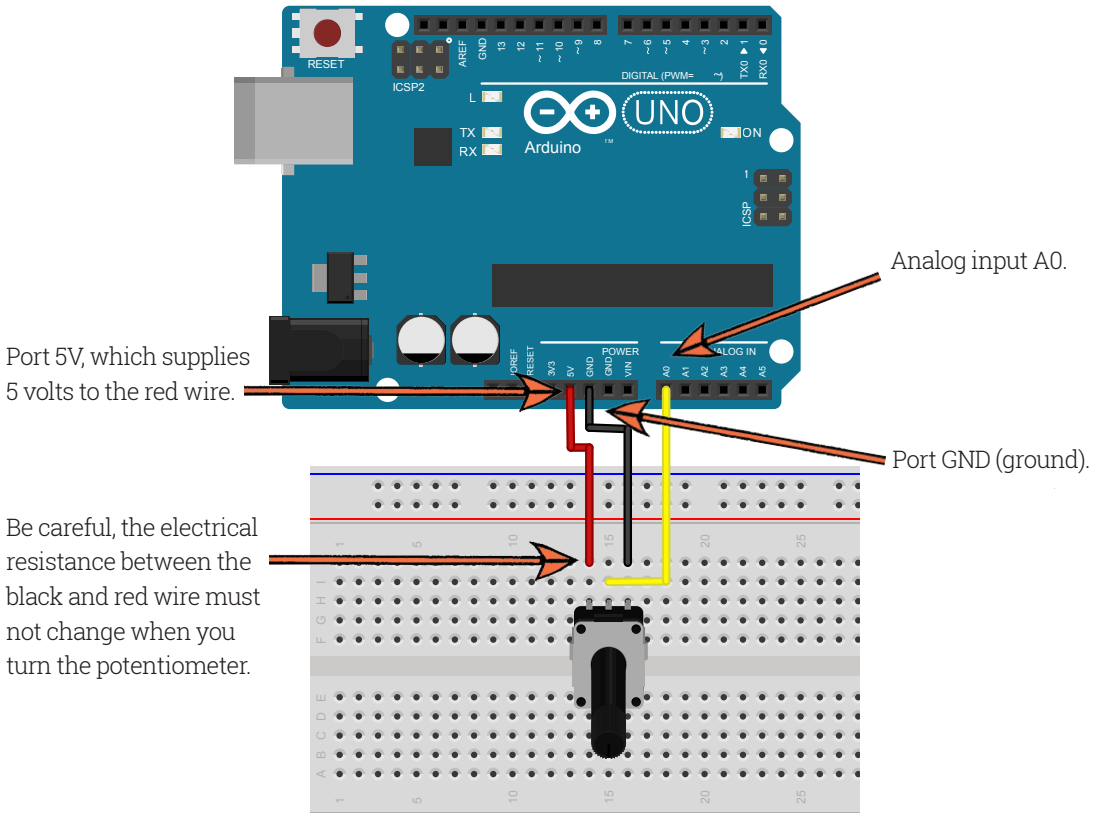
– a potentiometer of 10 000 ohms
– a breadboard
– small electrical wires
– possibly a multimeter

ASSOCIATED SHEETS

Test your board
Analog inputs

COMPLETE THIS ELECTRICAL SETUP

A potentiometer has three contacts. The electrical resistance between two of these contacts does not vary when you turn the potentiometer: it is these two contacts that must be connected to port 5V and to port GND. (If you have a multimeter, you can check where these contacts are located, if not, trust in the setup below.)



COPY THIS PROGRAM

```
int MeasuredVoltage ;              // one defines the MeasuredVoltage variable
                                   // this variable can store values

// setup to initialise the board
void setup() {                     // start of setup
  Serial.begin(9600) ;             // initialize serial communication with the computer
}                                  // end of setup

// this loop will continue indefinitely
void loop() {                                  // beginning of loop
  MeasuredVoltage = analogRead(A0) ;           // one measures the voltage on port A0
                                               // and attributes the MeasuredVoltage variable
  Serial.println(MeasuredVoltage) ;            // one sends the measured value to the computer
  delay(100) ;                                 // wait 0.1 seconds (100 milliseconds)
}                                              // end of loop
```

A variable allows you to store a value in a program. You have to declare them before using them: int is the type of variable (an integer), MeasuredVoltage is the name of the variable.

Short waiting time between two consecutive measurements

This instruction triggers the voltage measurement on port A0

A similar program is easily accessible through the software menu (File menu, Examples, Basics, program AnalogReadSerial).

UPLOAD

The Arduino board will measure the voltage on the A0 port then send the results to the computer. To read it directly through a computer, use the serial monitor (in the Tools menu of the Arduino software). If strange characters appear, check the connection speed of the serial monitor, which must be the same as that used to initialise communication in the program (in this case 9600 bauds).

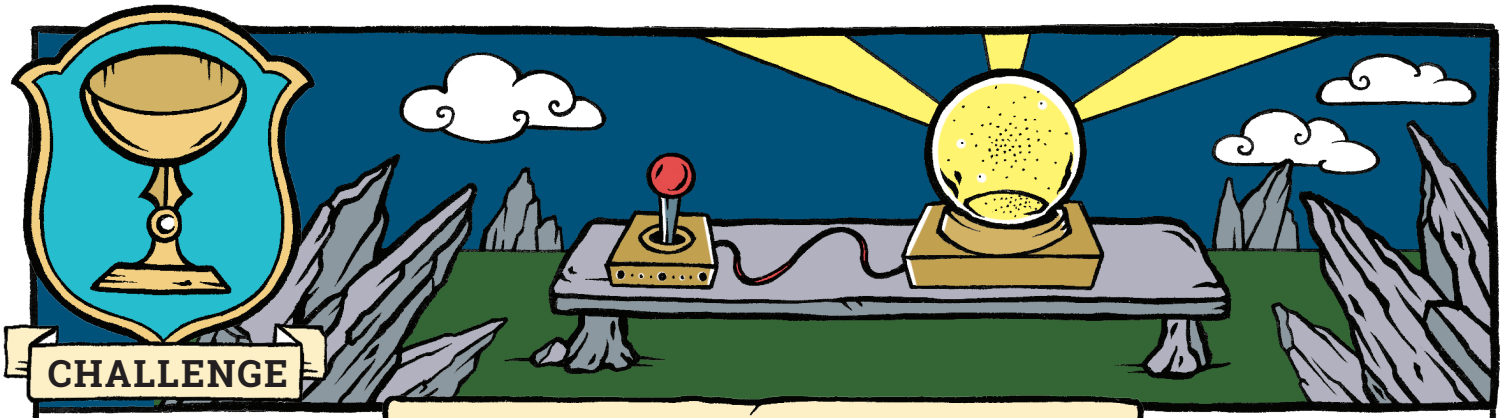
Turn the potentiometer and see what happens to the serial monitor!

TAKING IT FURTHER

Connect the red wire to port 3V3 and see what happens (this port delivers 3.3 volts). Find the relationship that allows you to transform what shows in the serial monitor into volts.

THE ULTIMATE CHALLENGE

Modify your program so that the test LED on the Arduino board lights up when the voltage measured exceeds 3.3 V (you will have to use "if... else....")



CONTROL THE LIGHTS

You will control the intensity of the lights of a LED!

MEDIUM

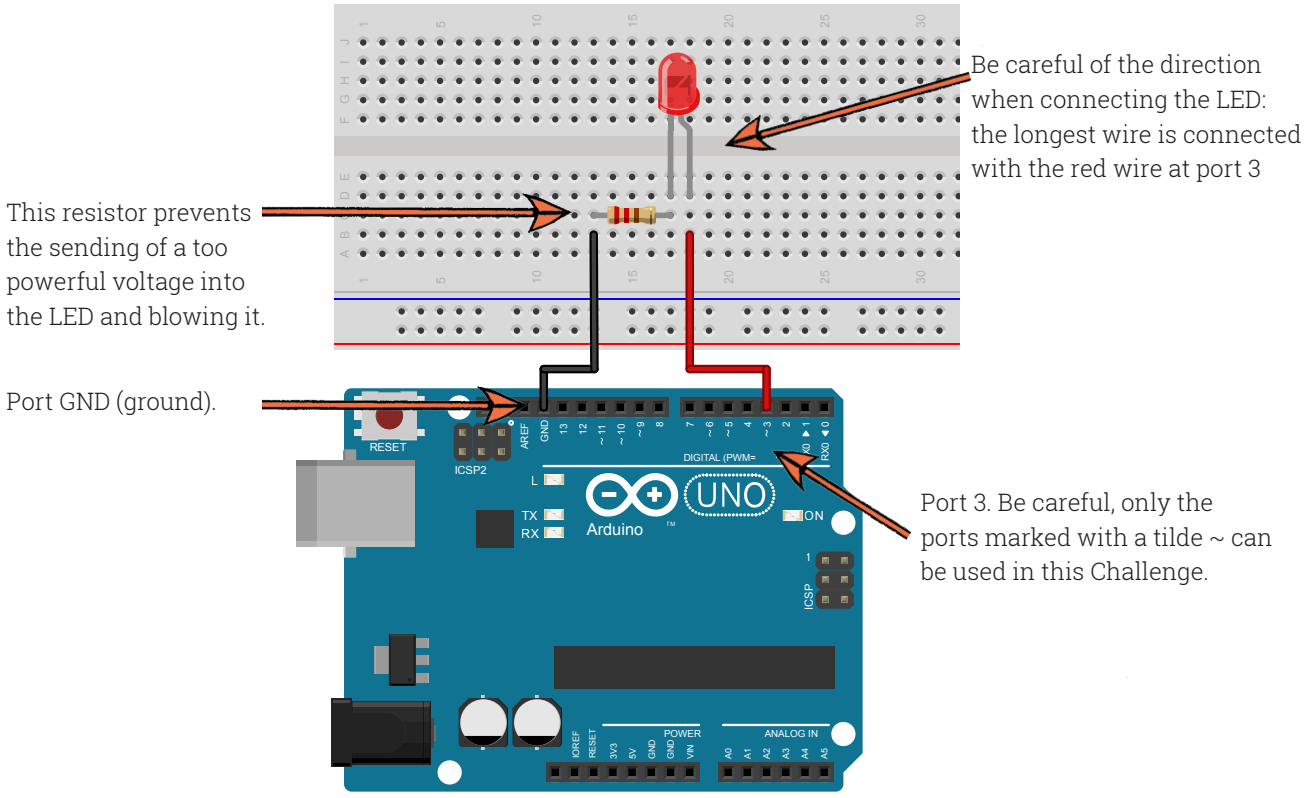
NECESSARY HARDWARE

- a red LED
- a resistor of 220 ohms
- a breadboard
- small electric wires

ASSOCIATED SHEETS

Test your board
PWM outputs

COMPLETE THIS ELECTRICAL SETUP



You can control the average value of the voltage on the digital ports marked with a tilde (see sheet "PWM output").

COPY THIS PROGRAM

LightValue is an integer variable which will store the value of the lighting that you want to apply to the LED. We give it an intial value of 0.

We impose an average voltage on port 3 which is equal to LightValue and can vary from 0 (which corresponds to 0 V) to 255 (which corresponds to 5 V).

```
int LightValue = 0 ;  // one defines the LightValue variable

// setup to initialize the board
void setup() {                    // begin setup
  pinMode(3, OUTPUT);             // one initialises the port 3 as output
}                                 // end of setup

// this loop will continue indefinitely
void loop() {                     // start of loop
  analogWrite(3, LightValue) ;    // one defines the output voltage of port 3
  delay(100) ;                    // wait 0.1 seconds (100 milliseconds)
  LightValue = LightValue + 10 ; // one increases by 10 the value of the variable
  if (LightValue > 255) {         //if one exceeds the maximum authorised value
    LightValue = 0 ;              // then one returns to 0
  }                               // end of  condition
}                                 // end of loop
```

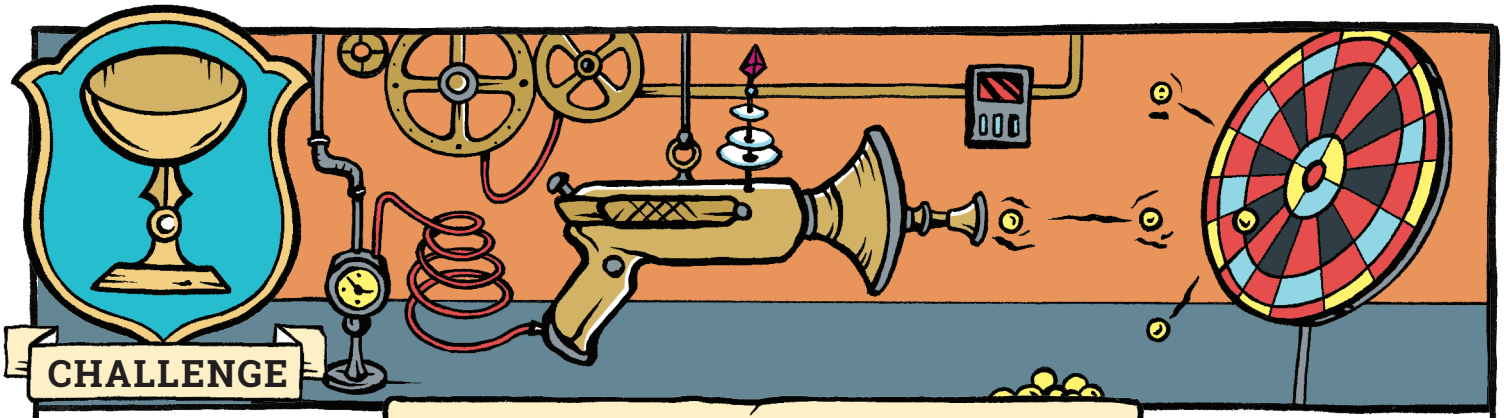
A similar program is easily accessible through the software menu (File menu, Examples, Basics, program Fade).

UPLOAD

The lighting of the LED progressively increases!

THE ULTIMATE CHALLENGE!

If you have already completed the Challenge “measure voltage”, try to modify the program to read the real voltage on port 3 by linking it to port A0, and then sending the results to the serial Monitor. Check to see if you can see the voltage change as you expected (in volts).
If not, modify your program so that the LED gets brighter then dims progressively (and return to the Challenge after having completed the “measure voltage” sheet.)



CREATIVE CHALLENGE

Be creative and original... and have fun!

EASY

NECESSARY HARDWARE

Everything you have at your disposal: Sensors and actuators as well as Lego, magnets, marbles, paper, aluminium foil, balloons, elastic bands, modelling dough, cardboard etc.

ASSOCIATED SHEETS

all

PREPARATION

Divide into teams of between two and four people. Collect your material together on an easily accessible table making sure to separate the tools (sensors and actuators: Hall effect sensor, light sensor, accelerometer, joystick, servomotor, buzzer, etc.) and the material for construction that must be as varied as possible (magnets, modelling dough, kebab skewer, balloon etc.).

THE CHALLENGE

Each team chooses a tool randomly and a piece of construction equipment. The teams then each have a limited amount of time (typically 1.5 hours or 2 hours) to design and build a game with the constraint of using the corresponding tool and accessory. At the end of the Challenge, the teams present their game to each other and have them tested by the other teams.

Try taking photographs or making short films!

VARIATIONS

Easy variation: Each group randomly picks two sensors and two accessories but then chooses to use one of each.
Special Design variation: The objective of the Challenge is no longer to build a game but a useful object, a useless object or perhaps an artistic object!

THE ULTIMATE CHALLENGE!

To add even more constraints to the game, have the teams randomly pick a supplementary constraint among the following: funny / surprising / for drinks / challenge for the user / load and bright / needs dexterity / for a child / without hands / artistic / without eyes…