



Spolufinancováno
z programu Evropské unie
Erasmus+

Inquiry-based Learning and Teaching Across Europe

2018-1-CZ01-KA201-048113

Using Arduino in STEM Education - Biology

Area of education/subject:	STEM, Arduino
Title of activity:	Using Arduino in STEM Education - Biology
Jaroslav Trnka	trnka@gyoa.cz



Why Arduino?

Arduino is one of the current most famous *microcontrollers*. Its popularity is based upon several advantages:

- 1) It is very cheap – affordable for your school as well as for individual students in their private activity.
- 2) It is complex enough to construct very diverse or quite complex devices. It mediates an experience of a free invention.
- 3) Yet it is simple enough to be comprehended even by children in lower secondary.
- 4) It has a broad community support.

Course of activity

Expected implementation duration: 45 minutes (depends on the extent of activities).

Use of the tool (forms and methods): group work in coding, heuristic conversation, practical experiments

Before the activity, basic one-time preparations are needed, like buying the stuff (Arduino, Pulse sensor, Servo, female to female jumpers), Arduino IDE installation or the sensor adaptation (more in the text below).

Information sources

<https://www.arduino.cc/>

<https://pulsesensor.com/>

https://docs.google.com/document/d/1d8EwDcXH1AZpIpEnrET28EBgStrbkbp_pxjQZcNRAkl/edit#

Equipment

Arduino UNO Rev3 (original or some of its clones): <https://store.arduino.cc/arduino-uno-rev3>

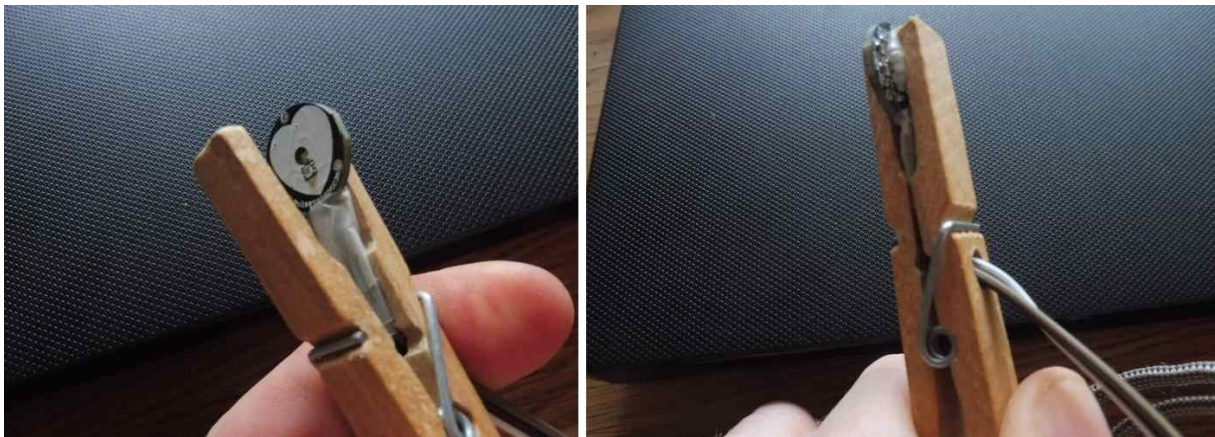
Any servo motor, for example 9G SG90 Micro Servo: <https://www.ebay.com/itm/9G-SG90-Micro-Servo-Motor-RC-Robot-Helicopter-Airplane-Remote-Control-Car-Boat/353369786610?hash=item52467b08f2:g:On0AAOSwTbRd0kWh>

Pulse sensor: <https://pulsesensor.com/>

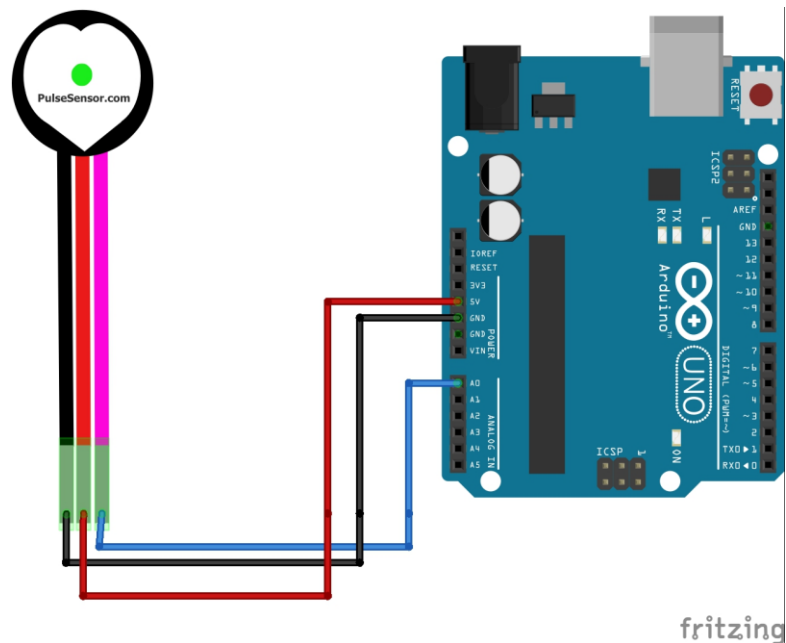


1. Adaptation of the sensor

Sensor is often provided with the belt for fixing to a finger. Yet I have a better experience with clipping the sensor on the earlobe. (Some looking for the right place is necessary.) If you don't have an original earclip, you can use for example a clothes peg – a hot glue will help:)



2. A connection of the pulse sensor to your Arduino:





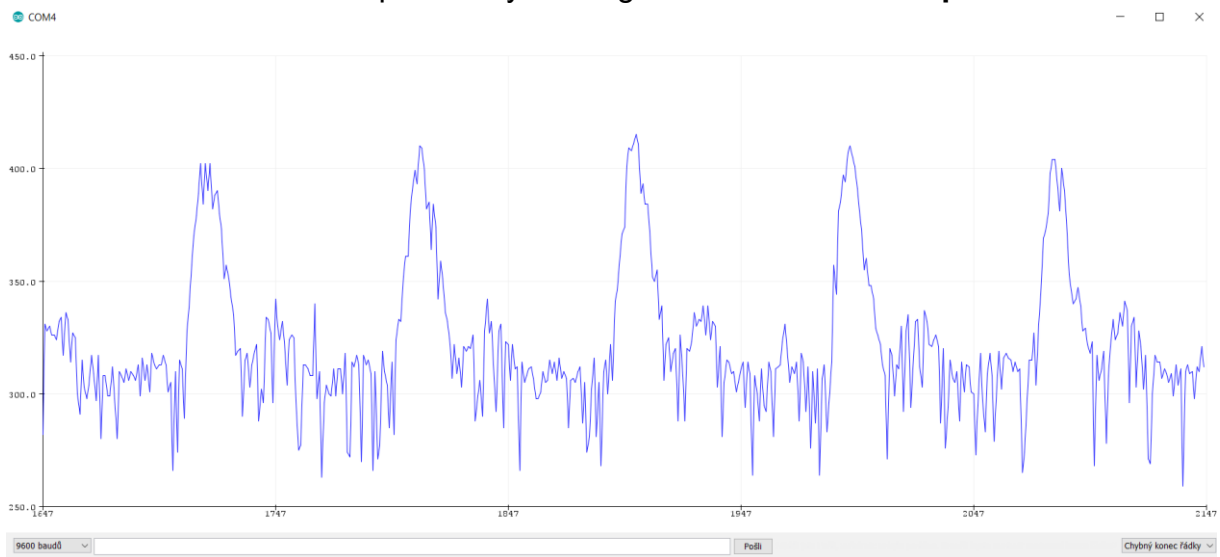
3. Writing a basic code. (The text after “//” is just a comment – Arduino doesn’t read it.)

```
int Signal;      //Defining a variable called “Signal” as integer

void setup() {   //Entrance setting
  Serial.begin(9600);
  Signal = 0;
}

void loop(){     //Main loop - continuing on and on and on...
  Signal = analogRead(A0); //Signal variable is filled up with a value from A0 input
  Serial.println(Signal);  //Signal variable is printed
  delay(10);             //Waiting 10 miliseconds before continuing
}                       //Returning to the beginning of the main loop
```

4. After basic introduction of code writing, you can look on the result through your Arduino connected to computer – by clicking on **Tools** and **Serial plotter**.



Suggested activities:



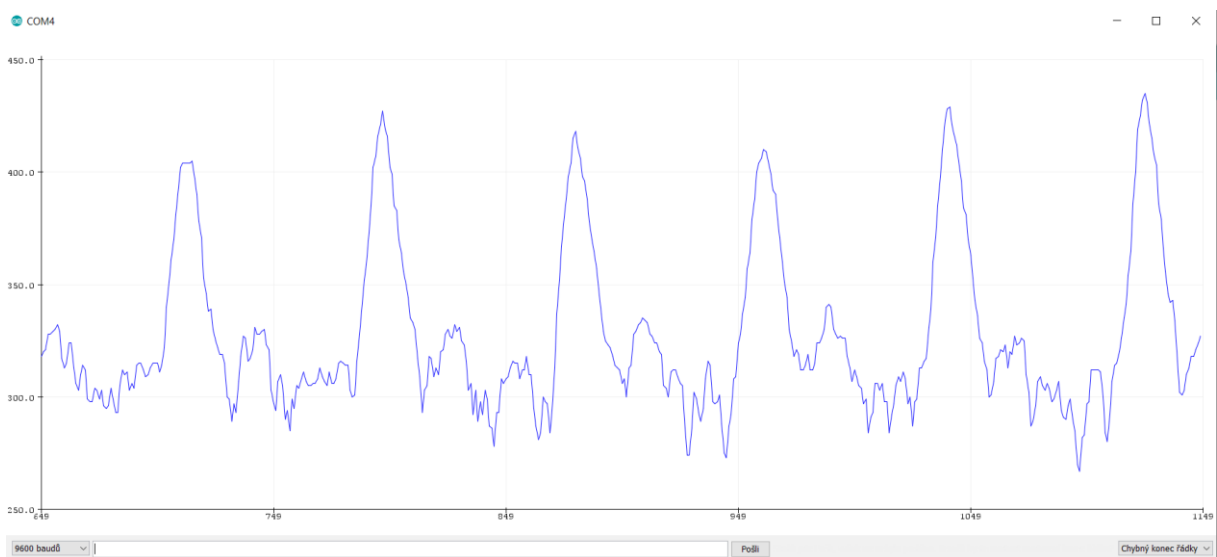
5. Smoothing the signal. Usually the signal is very rough, so you may want students to rewrite the code in order to compute an average of previous values. One of the possible simple solutions could look like this.

```
int Signal;    //Defining a variable called "Signal" as integer
int previous;
int preprevious;
int average;

void setup() {    //Entrance setting
  Serial.begin(9600);
  previous = 0;
  preprevious = 0;
  average = 0;
}

void loop(){      //Main loop - continuing on and on and on...
  Signal = analogRead(A0);    //Signal variable is filled up with a value from A0 input
  average = (Signal + previous + preprevious)/3; //Formula for computing an average
  Serial.println(average);    //Average variable is printed
  preprevious = previous;
  previous = Signal;
  delay(10);                //Waiting 10 miliseconds before continuing
}                            //Returning to the beginning of the main loop
```

Then your result could look like this:





6. Using the signal from the sensor for moving a servo:

```
#include <Servo.h> //Connecting a Servo Library
Servo myservo; // create servo object to control a servo

int Signal; //Defining a variable called "Signal" as integer
int previous;
int preprevious;
int average;
int servoposition; //Defining a variable for servoposition

void setup() { //Entrance setting
  myservo.attach(9); // attaches the servo on pin 9 to the servo object
  Serial.begin(9600);
  previous = 0;
  preprevious = 0;
  average = 0;
}

void loop(){ //Main loop - continuing on and on and on...
  Signal = analogRead(0); //Variable Signal is filled up with the value af AO input
  average = (Signal + previous + preprevious)/3;//A formula for computing an average
  Serial.println(average); //Printing average
  servoposition = map(average, 0, 1023, 0, 180); //mapping values of "average" on the
  extent of possible values of "servoposition"
  myservo.write(servoposition); // sets the servo position
  preprevious = previous;
  previous = Signal;
  delay(10); //Waiting 10 miliseconds before continuing
} //Returning to the beginning of the main loop
```

7. Possible following activities:

- a) Trying to identify the nature of the signal recorded by pulse sensor (Pressure? Electricity?....)
- b) Comparing the signal from two parallel sensors on distant body parts
- c) Comparing the signal from the pulse sensor with a signal from EKG sensor – computing the speed of a blood pressure wave.
- d) Writing a code for computing a heartbeat rate.