

Examples of physics lessons supported by the **ARDUINO** platform

PETR KOLÁŘ

FACULTY OF MATHEMATICS AND PHYSICS
CHARLES UNIVERSITY
CZECH REPUBLIC



non-research contribution

**teaching experience
from grammar school**

CONTENTS

- MY MOTIVATION WHY TO USE ARDUINO
- EXAMPLES HOW TO IMPLEMENT ARDUINO INTO PHYSICS LESSONS

MY MOTIVATION WHY TO USE ARDUINO

PHYSICS CURRICULUM RESEARCH RESULTS:

- RESEARCH AMONG: LEADING PHYSICISTS,
OTHER SCIENTISTS,
PHYSICS TEACHER EDUCATORS,
AND PHYSICS TEACHERS
- RESULTS: HUMANISTIC ORIENTATION (student-centered),
USE OF ICT, CONTEXT, ETC.

STATE POLICY

- ALL STUDENTS AT GRAMMAR SCHOOLS SHOULD BE ABLE TO PROGRAM
- MORE ICT LESSONS/LESS NATURAL SCIENCES AND MATH LESSONS

LET'S PROGRAM DURING PHYSICS LESSONS/DO PHYSICS DURING ICT LESSONS

(The wolf ate and the goat remained whole)

ADVANTAGES OF ARDUINO

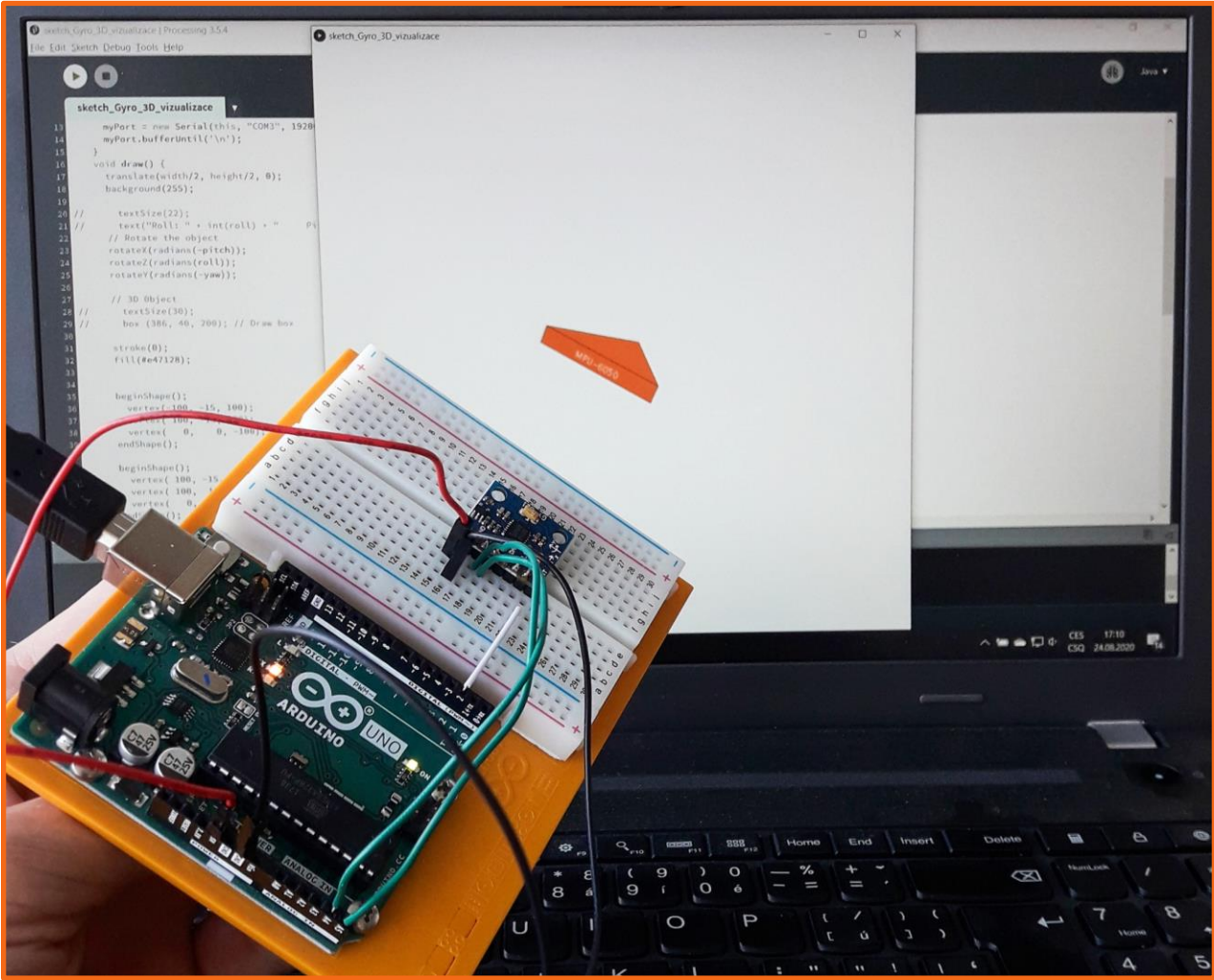
- CHEAP/NOT VERY EXPENSIVE
- WORLD-WIDE PLATFORM WITH MANY USERS
- ArduSat
- MANY SENSORS USABLE IN PHYSICS AND EASY TO PROGRAM
- STUDENTS' INTEREST AND MOTIVATION

EXAMPLES HOW TO IMPLEMENT ARDUINO INTO PHYSICS LESSONS

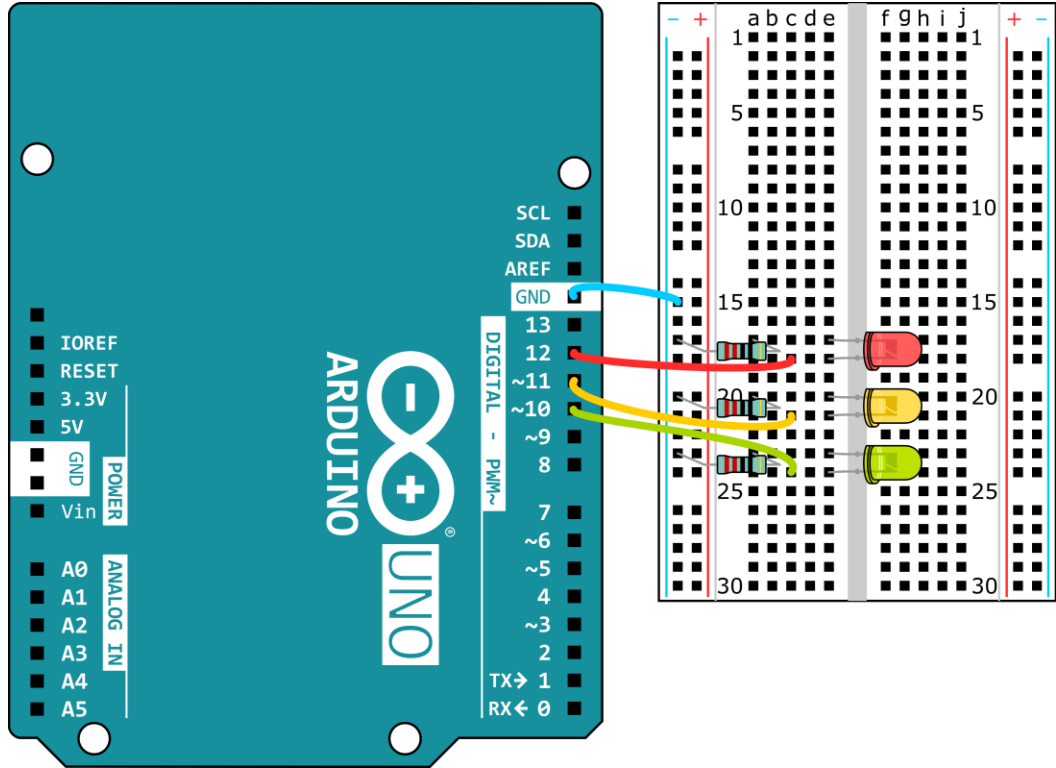
LIST OF EXAMPLES:

- TRAFFIC LIGHTS
- SPEED & DISTANCE
- TEMPERATURE
- VOLTAGE
- LIGHT INTENSITY
- BACKGROUND RADIATION

ACCELEROMETER AND GYROSCOPE



TRAFFIC LIGHTS:



TRAFFIC LIGHTS | Arduino IDE 2.0.3 | Arduino UNO Rev3

```
1 #define RED 12
2 #define YELLOW 11
3 #define GREEN 10
4
5 void setup() {
6   pinMode(RED, OUTPUT);
7   pinMode(YELLOW, OUTPUT);
8   pinMode(GREEN, OUTPUT);
9 }
10
11 void loop() {
12   digitalWrite(RED, HIGH);
13   delay(10000);
14
15   digitalWrite(YELLOW, HIGH);
16   delay(1000);
17
18   digitalWrite(RED, LOW);
19   digitalWrite(YELLOW, LOW);
20   digitalWrite(GREEN, HIGH);
21   delay(10000);
22
23   digitalWrite(GREEN, LOW);
24   digitalWrite(YELLOW, HIGH);
25   delay(2000);
26
27   digitalWrite(YELLOW, LOW);
28 }
29
```

SPEED OF SOUND:

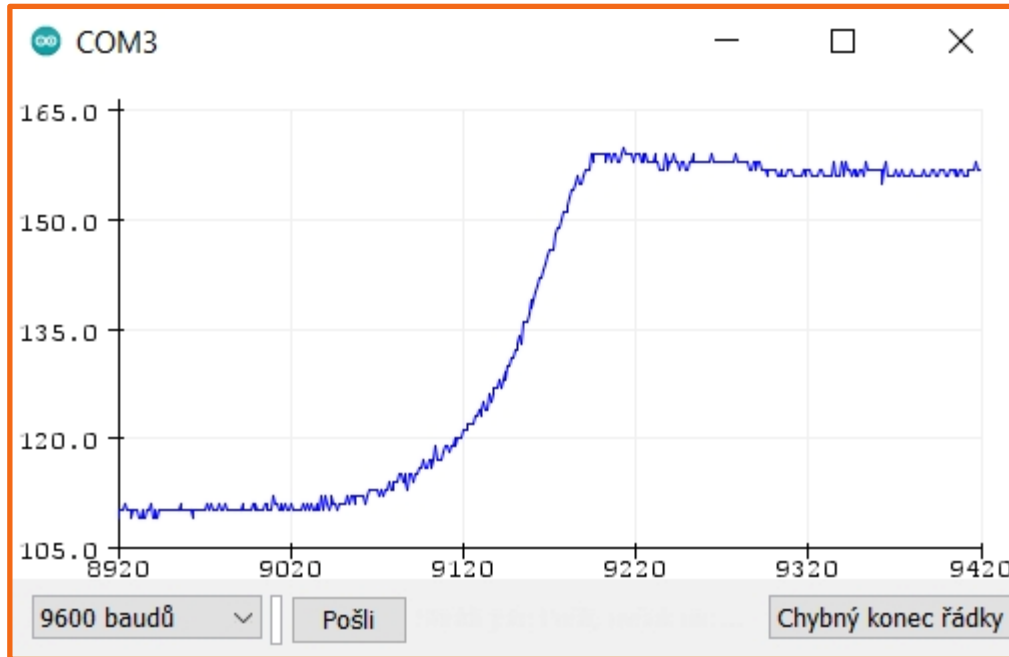
- SONAR: HC-SR04
(1.73 USD)
- Alternatives:
 - Vernier Go!Motion
(381.67 USD)
 - PASCO PS-2103A
(249.66 USD)



```
1 #define TRIG 4
2 #define ECHO 6
3
4 long response; // time (microseconds)
5 float speed;
6
7 float distance = 0.755; // (meters)
8
9 void setup() {
10   Serial.begin(9600);
11
12   pinMode(TRIG, OUTPUT);
13   pinMode(ECHO, INPUT);
14 }
15
16 void loop() {
17   digitalWrite(TRIG, LOW);
18   delayMicroseconds(2);
19   digitalWrite(TRIG, HIGH);
20   delayMicroseconds(5);
21   digitalWrite(TRIG, LOW);
22
23   response = pulseIn(ECHO, HIGH);
24
25   speed = distance/response*2000000; // (meters/seconds)
26                                     // *2.237 (mph)
27   Serial.print("Speed of sound: ");
28   Serial.print(speed);
29   Serial.println(" m/s");
30   delay(1000);
31 }
32
```

DISTANCE:

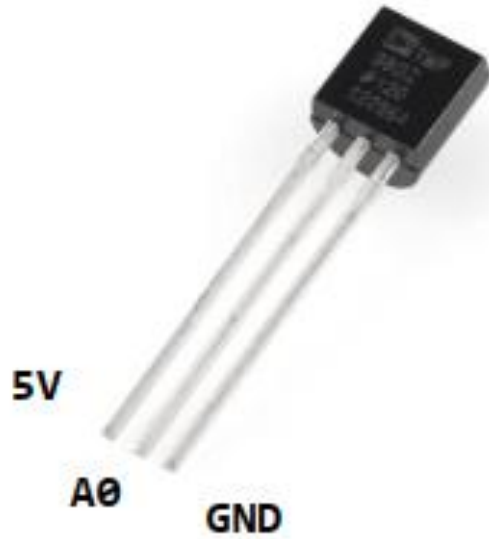
- GRAPHS:



- PARKING SENSOR

DISTANCE | Arduino IDE 2.0.3 | Arduino UNO Rev3

```
1 #define TRIG 4
2 #define ECHO 5
3
4 unsigned long time;
5 float duration = 10; // s
6 float interval = 0;
7 float constant = 58.309; // us/cm
8 long response, distance;
9
10 void setup() {
11   pinMode(TRIG, OUTPUT);
12   pinMode(ECHO, INPUT);
13
14   Serial.begin(9600);
15 }
16
17 void loop() {
18   while(time <= duration*1000){
19     digitalWrite(TRIG, LOW);
20     delayMicroseconds(2);
21     digitalWrite(TRIG, HIGH);
22     delayMicroseconds(5);
23     digitalWrite(TRIG, LOW);
24
25     response = pulseIn(ECHO, HIGH);
26
27     distance = response / constant;
28
29     time = millis();
30
31     if(time > interval){
32       Serial.println(distance);
33       interval += duration*1000/500;
34     }
35   }
36 }
37
```

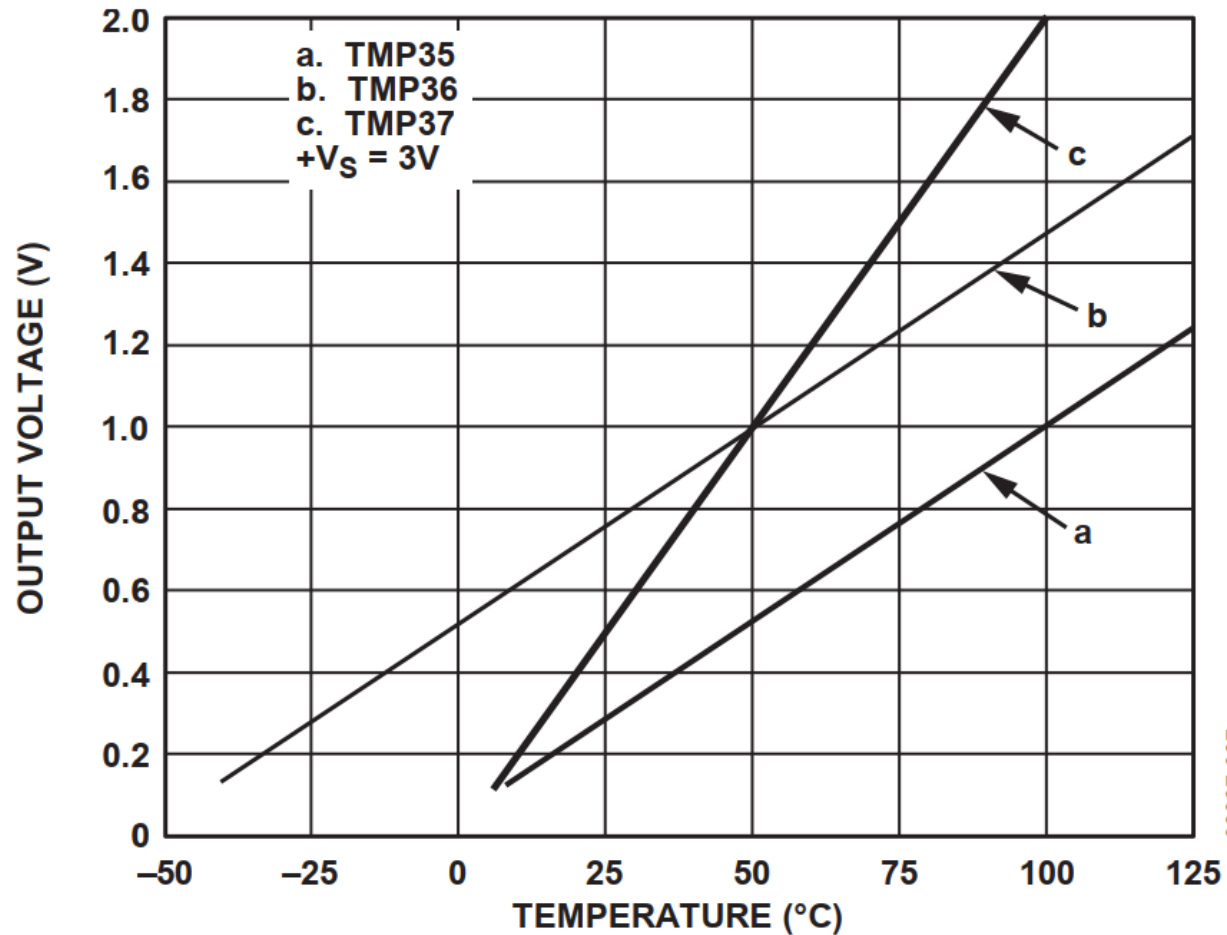



TEMPERATURE | Arduino IDE 2.0.3 | Arduino UNO Rev3

```
1 void setup() {  
2   Serial.begin(9600);  
3 }  
4  
5 void loop() {  
6   int sensorValue = analogRead(A0);  
7  
8   float voltage = sensorValue*5.0/1024.0;  
9  
10  float temperature = 100*voltage - 50;  
11  
12  //Serial.print("Temperature: ");  
13  Serial.println(temperature);  
14  //Serial.println(" °C");  
15  
16  delay(1000);  
17 }  
18
```

• EQUATION: $\text{temperature} = 100 * \text{voltage} - 50$

• TMP36 DATASHEET:



$$T = -50 \rightarrow V = 0$$

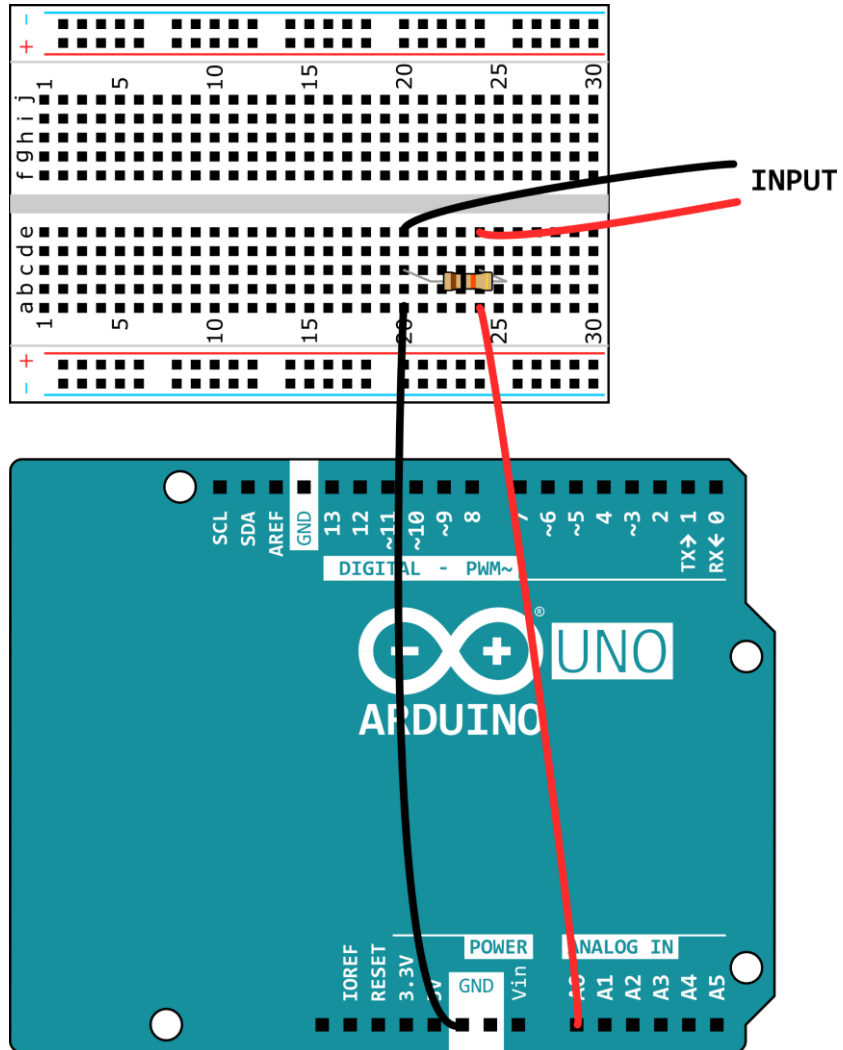
$$T = 50 \rightarrow V = 1$$

$$V = T/100 + 1/2$$

$$T = 100 * V - 50$$

00337-007

VOLTAGE:

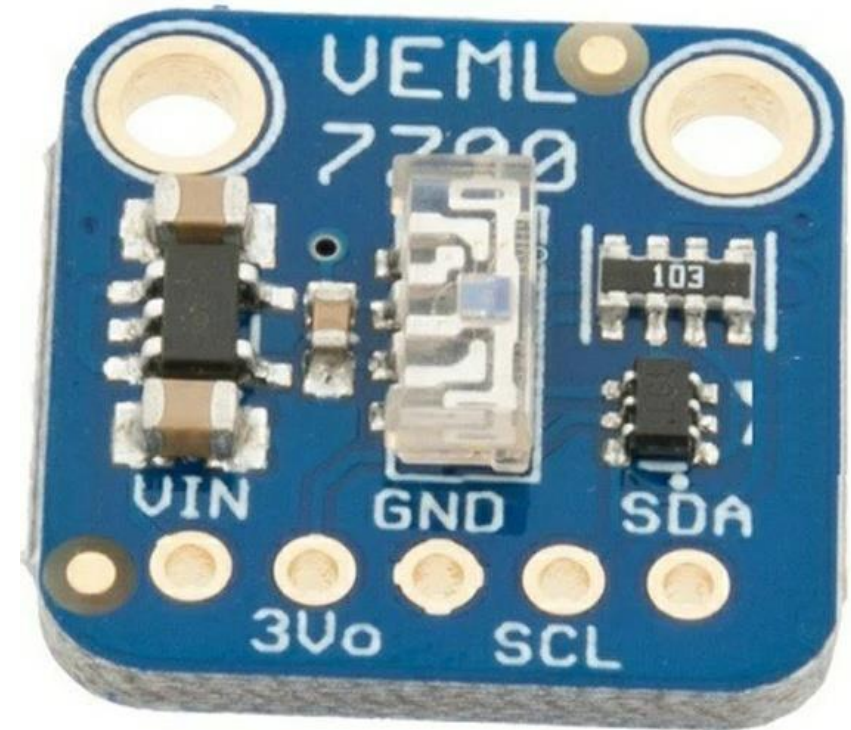


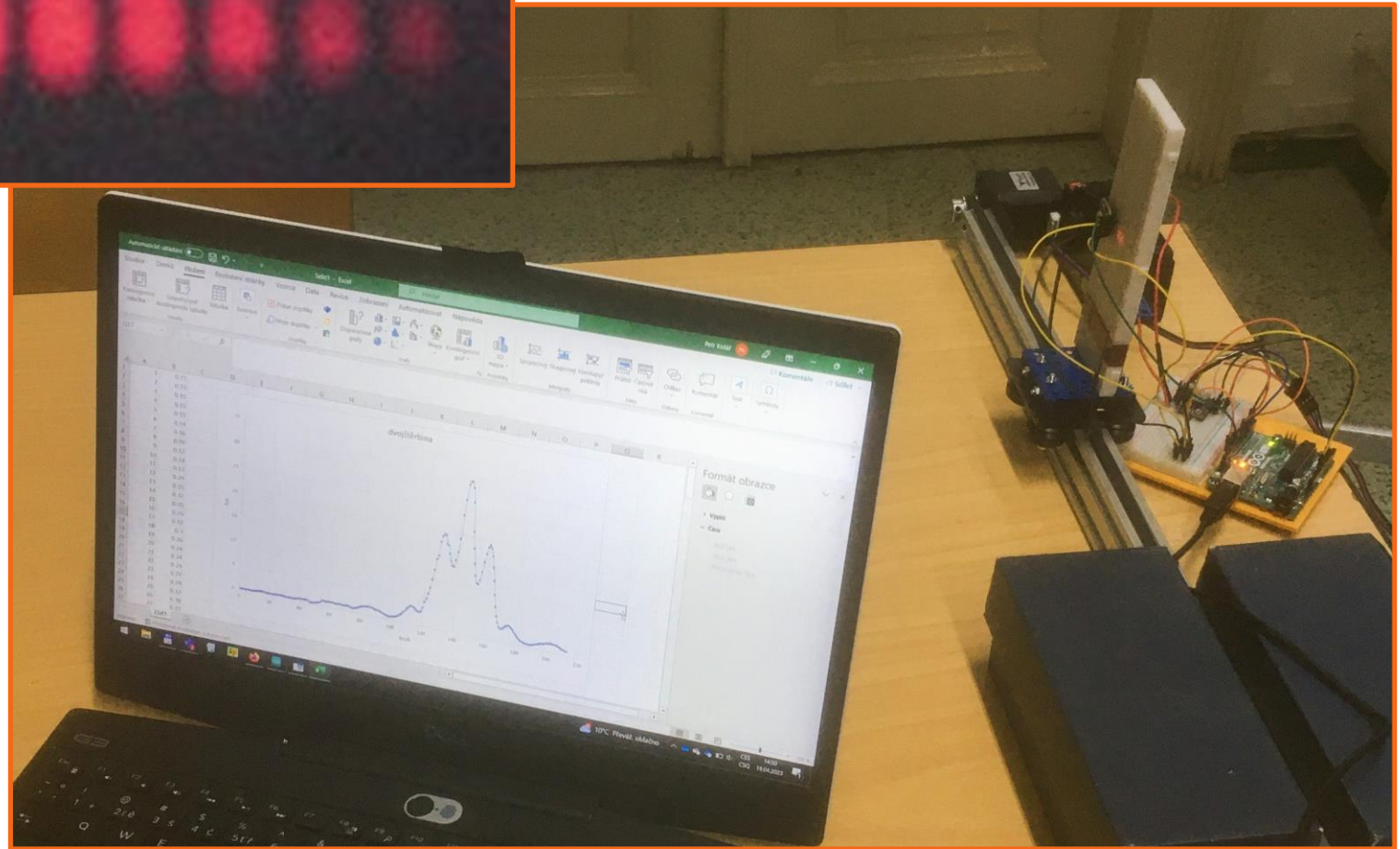
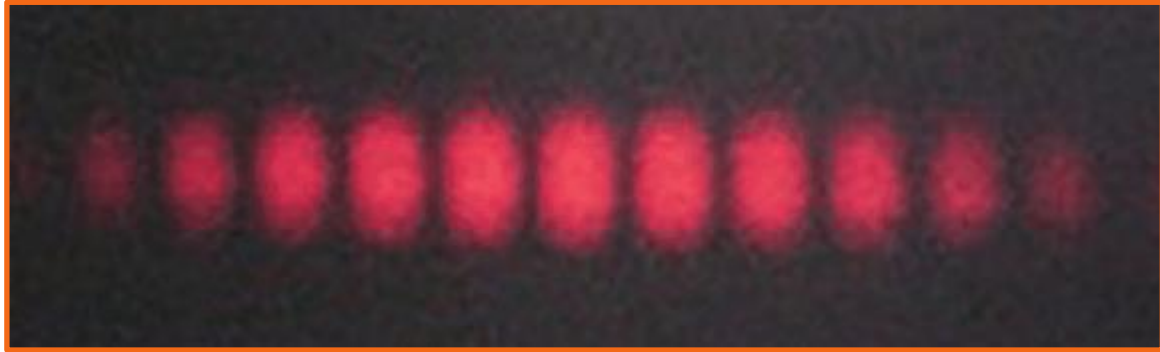
VOLTAGE | Arduino IDE 2.0.3 | Arduino UNO Rev3

```
1 void setup() {  
2   Serial.begin(9600);  
3 }  
4  
5 void loop() {  
6   int sensorValue = analogRead(A0);  
7  
8   float voltage = sensorValue*5.0/1024.0;  
9  
10  Serial.print("Voltage: ");  
11  Serial.print(voltage);  
12  Serial.println(" V");  
13  
14  delay(1000);  
15 }  
16
```


LIGHT INTENSITY:

- **SENSOR: VEML7700**
(16.73 USD)
- **Alternatives:**
 - Vernier Light Sensor**
(188.66 USD)
 - PASCO PS-2168**
(109.96 USD)







BACKGROUND RADIATION:

- **SENSOR: CAJOE RadiationD v1.1**
(90.80 USD)

- **Alternatives:**

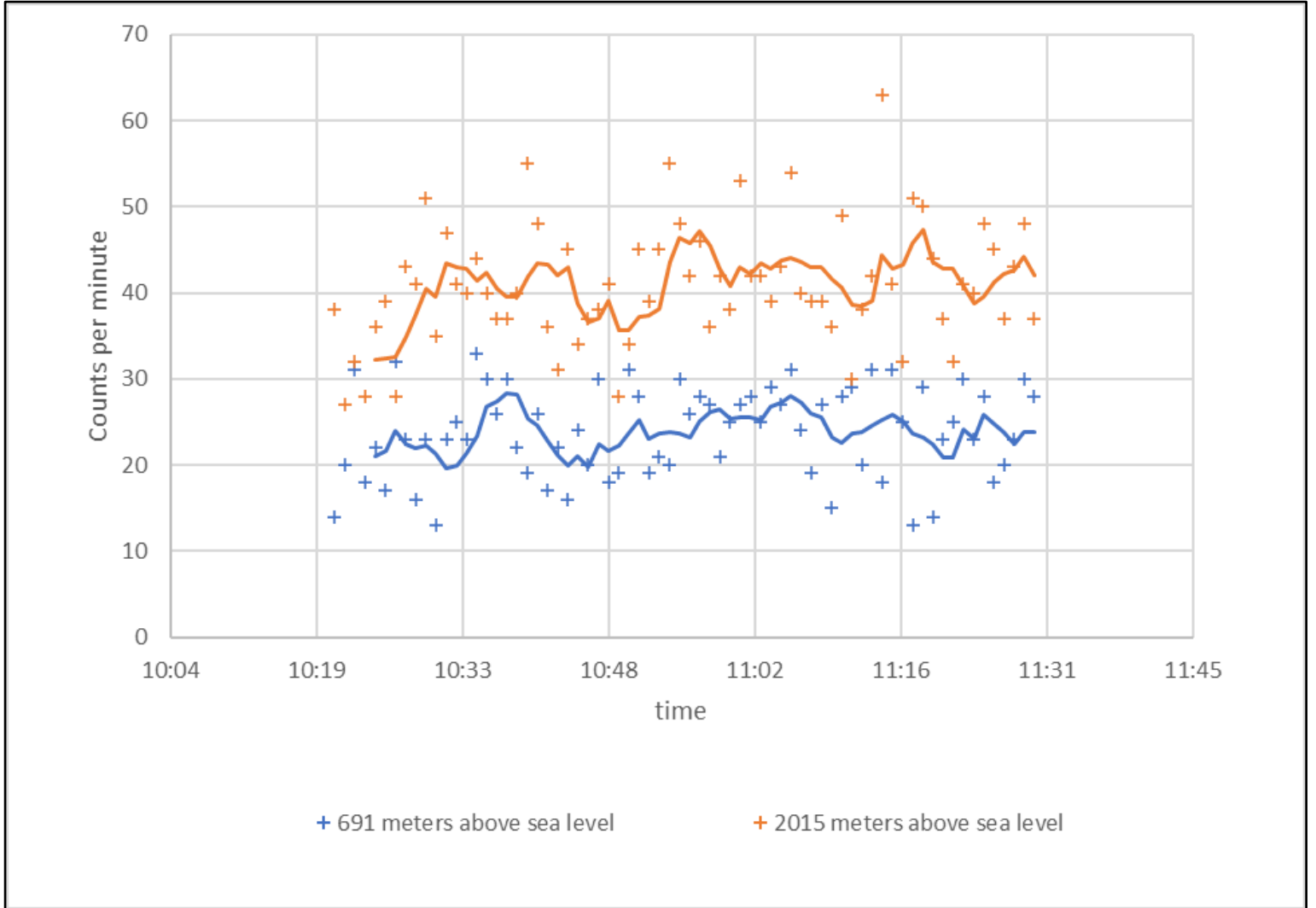
Vernier Radiation Monitor
(544.10 USD)

PASCO PS-3238
(544.37 USD)

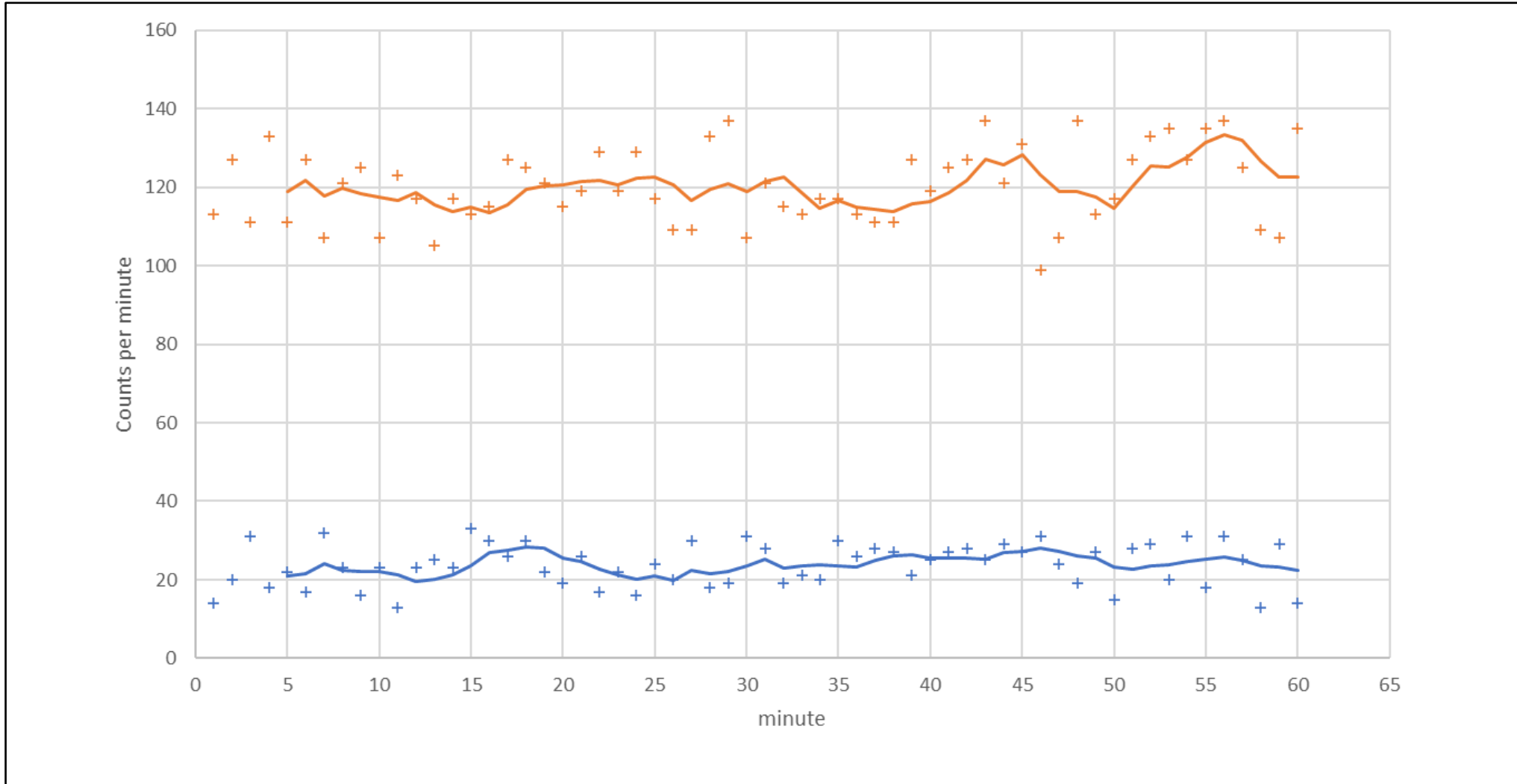


```
1 unsigned long TIME = 0;
2 unsigned long previousTIME = 0;
3 unsigned long COUNTS = 0;
4 unsigned long MINUTE = 0;
5
6 void setup() {
7   Serial.begin(9600);
8   pinMode(2, INPUT);
9   attachInterrupt(digitalPinToInterrupt(2), event, FALLING);
10 }
11
12 void loop() {
13   TIME = millis();
14
15   if (TIME - previousTIME >= 10000) {
16     previousTIME = TIME;
17     MINUTE++;
18     Serial.println(COUNTS);
19     COUNTS = 0;
20   }
21 }
22
23 void event() {
24   COUNTS++;
25 }
26
```

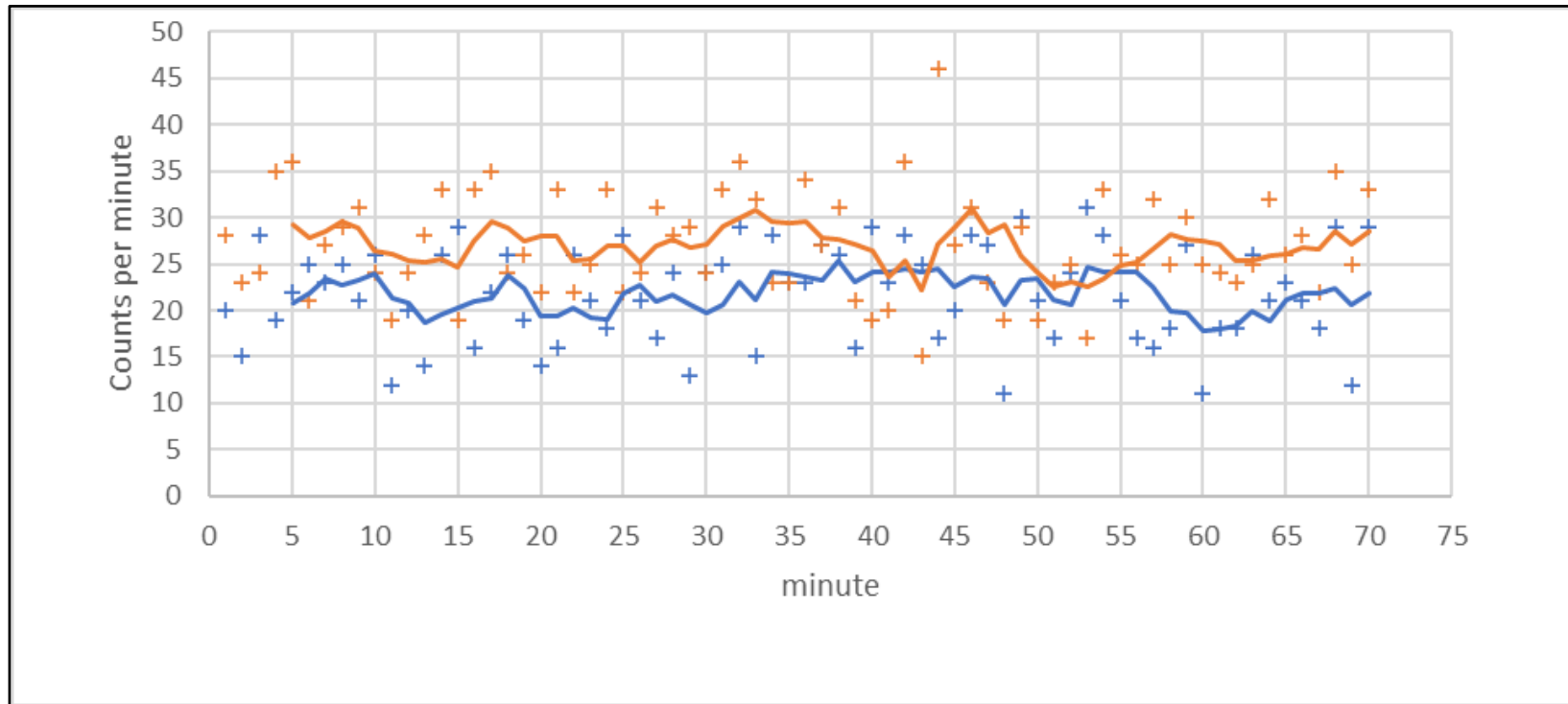
• BACKGROUND RADIATION CHANGE WITH ELEVATION



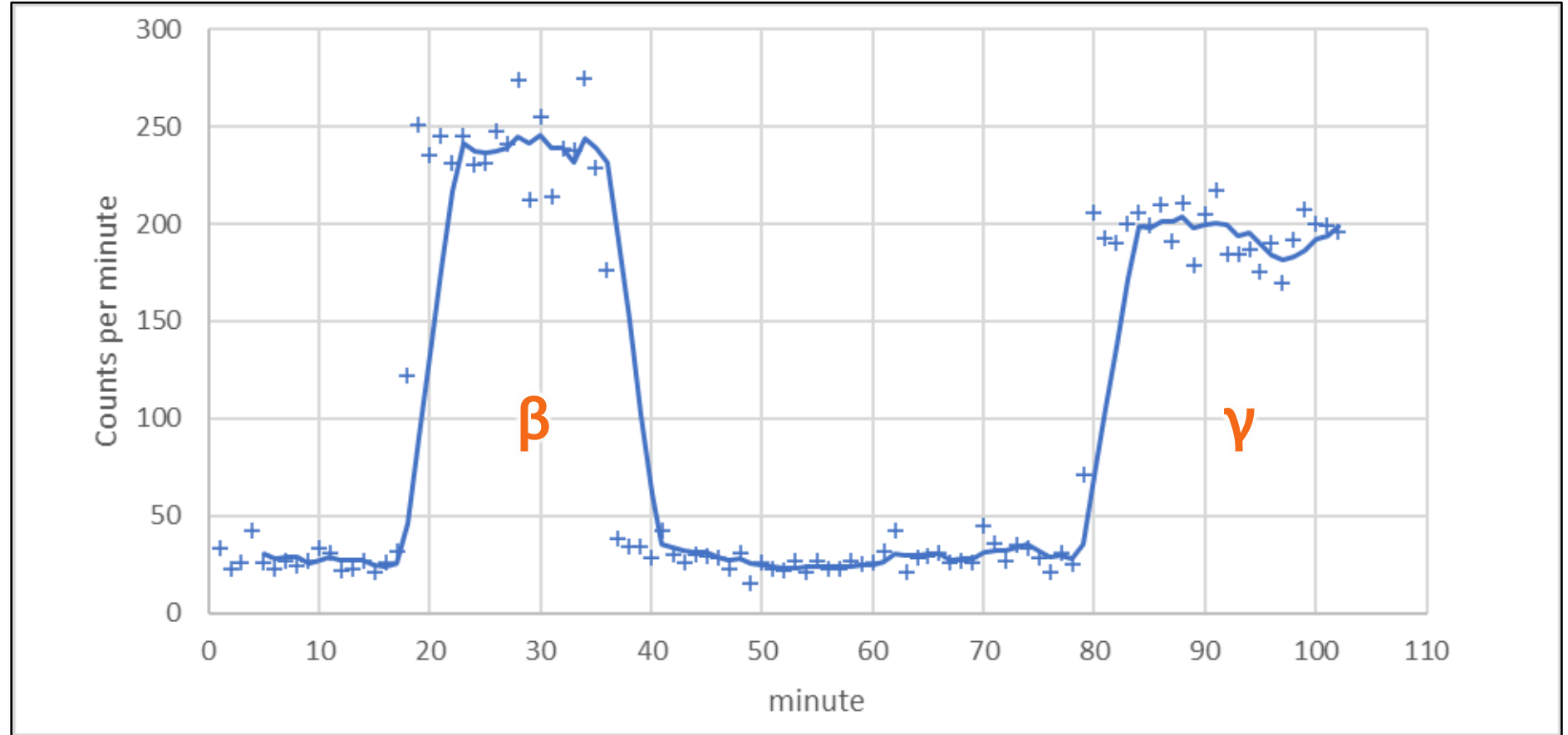
- **BACKGROUND RADIATION CHANGE WITH ELEVATION - flight**



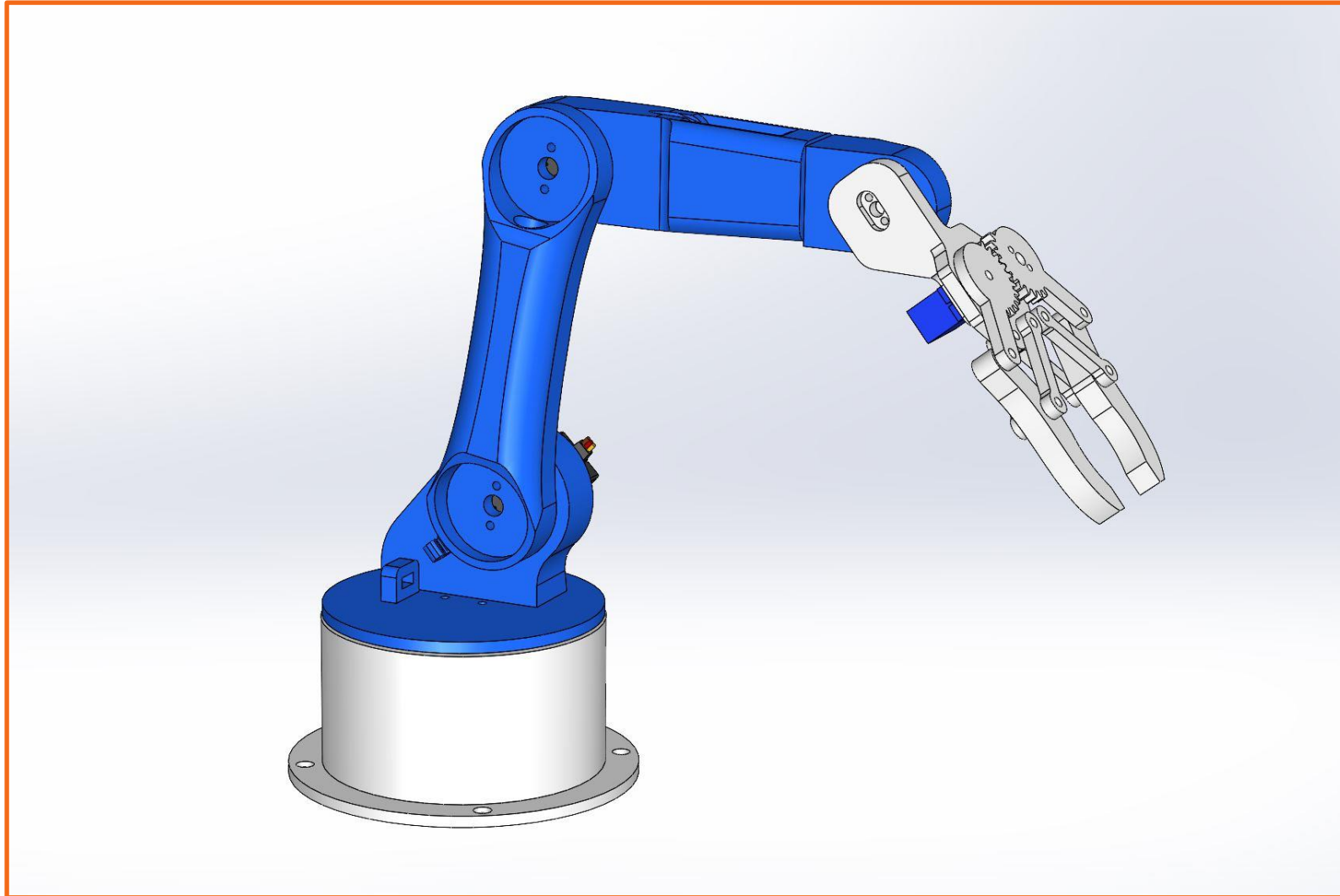
- **BACKGROUND RADIATION CHANGE WITH LOCATION (uranium ore underground)**



• SCHOOL RADIATION SOURCES:



STUDENT PROJECT – ROBOTIC ARM:



THANK YOU

petr.kolar@matfyz.cuni.cz