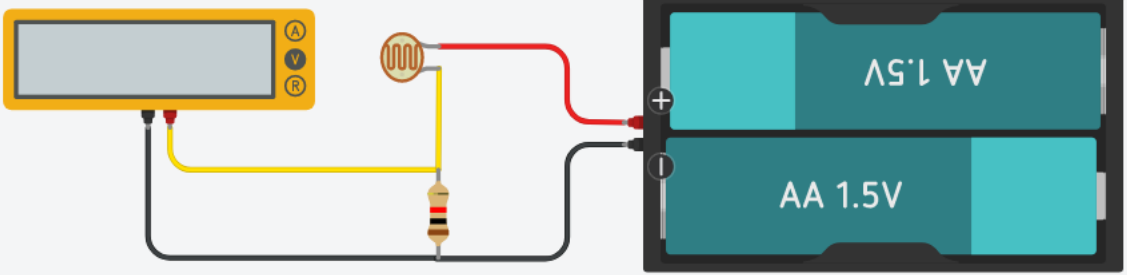


# Пупинов изазов – Фото сензор

*<Ово је рејрезенијивни пример изазова који се решавају на финалном ијесју ијакмичења „Пуинов изазов“. Изазови ијокривају различиие обласии као шиио су физика, механика, елекїроїтехника, елекїроника, ијроїрамирање хардвера и софївера.>*

Тема овог Пупиновог изазова је детектовање и мерење интензитета светлости. Потребно је дизајнирати шему којом се детектује интензитет светлости, а онда приказати резултате мерења на графику. Кроз 6 проблемских изазова из различитих области је потребно доћи до финалног решења.

Изазов 1	Редна веза отпорника	Област: Физика/електротехника
<p>Потребно је дизајнирати електричну шему у којој су два редно везана отпорника везана на извор напајања од 3V. Један од отпорника је фото-отпорник коме се смањује отпорност ако на њега пада јача светлост. Потребно је поставити мерач напона којим ће се измерити напон на неком од отпорника тако да је измерени напон већи ако фото-отпорник детектује више светлости.</p>		
<p><b>Решење</b></p> <p>Када фото-отпорник детектује јачу светлост, његова отпорност се смањује а самим тим се смањује и напон на њему. Ако напон на њему смањује, онда напон на другом отпорнику повећава зато што је збир напона на два отпорника увек 3V. То значи да се напон на обичном отпорнику повећава када на фото-отпорник пада више светлости, тако да је потребно мерити напон на обичном отпорнику.</p> <p>Једна од могућих шема је приказана на слици:</p>		
 <p>The diagram illustrates a circuit for measuring the voltage across a resistor. A multimeter is connected in parallel with a resistor. This resistor is in series with a photoresistor. The entire circuit is powered by a 3V battery source, which is represented by two AA 1.5V cells connected in series. The positive terminal of the battery is connected to the photoresistor, and the negative terminal is connected to the resistor. The multimeter's positive lead is connected to the node between the photoresistor and the resistor, and its negative lead is connected to the negative terminal of the battery.</p>		
<p>Извор напона су две батерије од по 1.5V. На њихове +/- прикључке су везана два отпорника. Мултиметар је конфигуриран да мери напон на једном отпорнику тако да је негативни прикључак повезан на негативни крај батерије, а позитивни прикључак на тачку у којој се повезују отпорници. На овај начин је мултиметар повезан паралелно отпорнику и мери напон на њему.</p> <p><b>Напомена:</b> За максимални број бодова је потребно да такмичари објасне зашто је одабрана оваква конфигурација, зашто су одабрали да мере напон на одабраном отпорнику а не на другом и како интензитет светлости утиче на мерени напон.</p>		
Изазов 2	Фото отпорник	Област: Физика/електротехника
<p>Фото-отпорник има већу отпорност ако на њега пада мање светлости а мању отпорност ако на њега пада више светлости. Када нема светлости, отпорност фото-отпорника је 180kΩ. Када је осветљен максималном количином светлости отпорност му је 506Ω.</p> <p>Ако је фото-отпорник редно везан са отпорником који има отпорност 20kΩ и ако су заједно везани на извор напајања од 3V, колики је напон на обичном отпорнику измерен у случају да је фото-отпорник неосветљен, односно максимално осветљен?</p>		

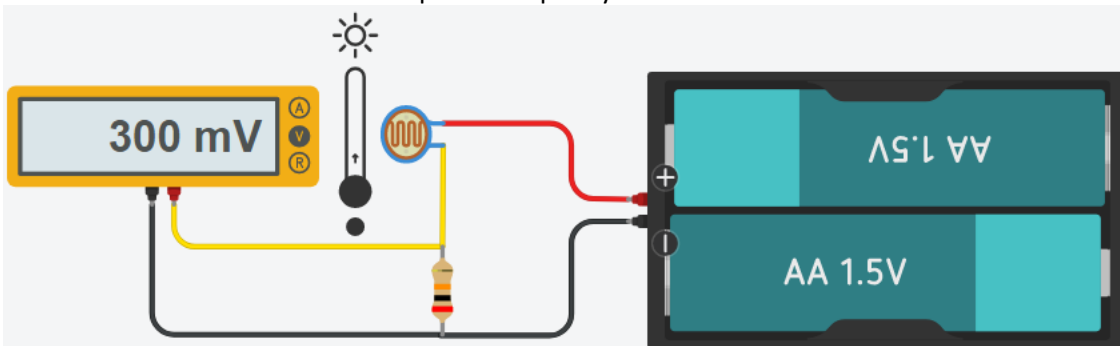
### Решење

Променама вредности осветљења на фото-отпорнику може се измерити напон на уређају ако се користи симулатор, или се може користити формула  $V = \frac{R}{R+R_f} * E$ , где је  $R_f$  фото-отпорник.

Минимум осветљења	300mV
Максимум осветљења	2.9V

Када на фото-отпорник пада минимална количина светлости, његова отпорност је максимална а самим тим и напон на њему је максималан. Када је напон на њему максималан, тада је напон који се мери најмањи (300mV). Када на фото-отпорник пада највише светлости, његова отпорност је максимална а самим тим и напон на њему је максималан. Измерени напон на обичном отпорнику је максималан. На овај начин показујемо да вредност измереног напона сразмерна количини светлости.

У случају да се користи симулатор, вредност напона се може прочитати на уређају за мерење напона тако што се мења осветљење на фото-отпорнику.



**Напомена:** За максимални број бодова је потребно да такмичари прикажу формулу или покажу да знају како се мењају односи напона на отпорницима. Признају се резултати чак и ако у опсегу од 20% грешке ако су коректно објашњени.

### Изазов 3

Дизајн *Arduino* фото сензора

**Област:** Електротехника

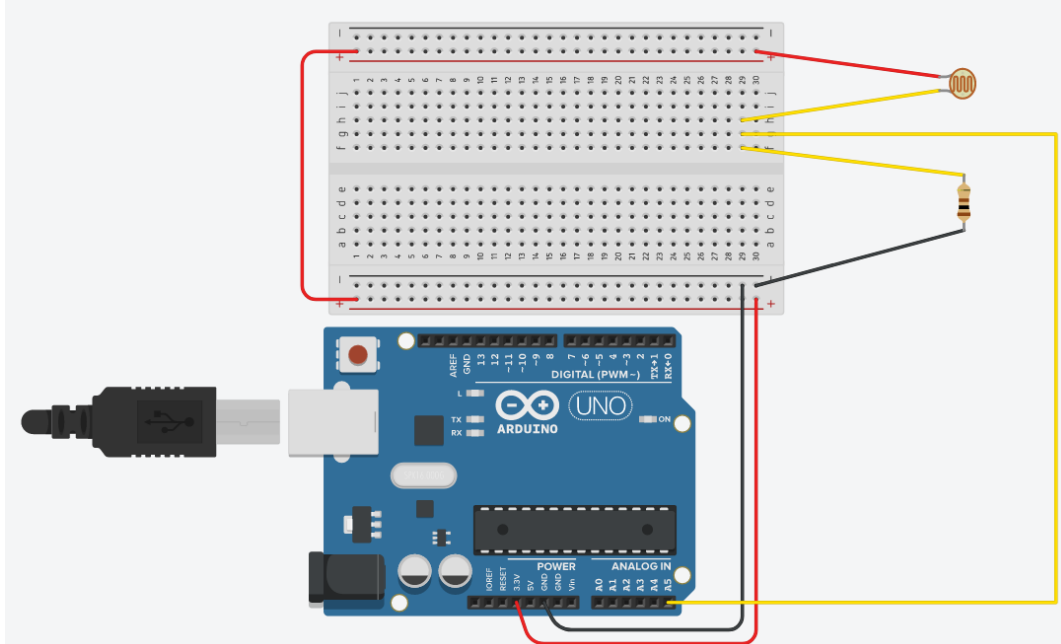
Потребно је дизајнирати шему којом се помоћу *Arduino* уређаја мери интензитет светлости. Потребно је користити уграђено напајање на *Arduino* уређају и мерити напон помоћу аналогних пинова на *Arduino* уређају.

### Решење

Дизајн се може добити променом оригиналног дизајна шеме у коме се уместо батерије и уређаја за мерење напона користи *Arduino* уређај.

- Уместо батерије користи уграђено напајање од 3.3V на *Arduino* плочи.
- Уместо уређаја за мерење напона, користи улазни аналогни пин на *Arduino* плочи.

Дизајн је приказан на слици:



Улаз А5 је замена за улаз за мерење напона у претходном примеру. То значи конфигурација *Arduino* уређаја може да мери интензитет светлости коју региструје фото-сензор преко пина А5. **Напомена:** Признаје се било какав начин повезивања који је еквивалентан шеми која је дизајнирана у претходном задатку.

**Изазов 4** | *Arduino* фото сензор | **Област:** Програмирање хардвера

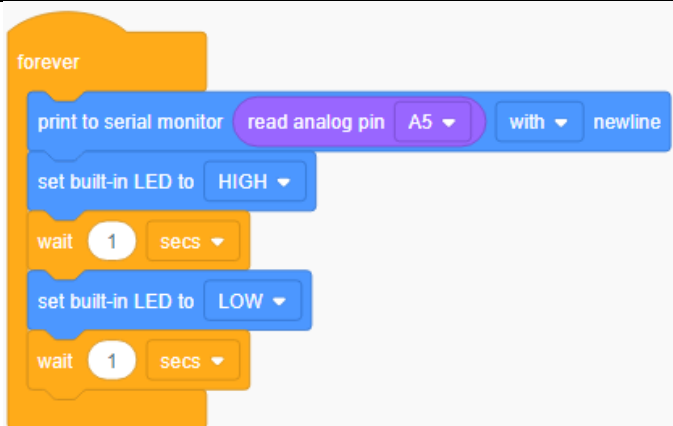
Потребно је имплементирати код (визуелно или на С језику) којим се програмира *Arduino* уређај за мерење количине светлости. Код мора да уради следеће операције:

- Пали и гаси уграђену LED-диоду на *Arduino* уређају.
- Чита вредност са пина којим се мери напон и ту вредност шаље ка серијском порту.

### Решење

Визуелни и програмски код којим се имплементира оваква функционалност су приказани у примеру:

*Arduino Scratch* решење



*Arduino C* решење

```
void setup()
{
  pinMode(A5, INPUT);
  Serial.begin(9600);
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop()
{
  Serial.println(analogRead(A5));
  digitalWrite(LED_BUILTIN, HIGH);
  delay(1000); // Сачекај 1 секунд
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000); // Сачекај 1 секунд
}
```

**Напомена:** За максимални број бодова је потребно да такмичари прикажу на серијском порту вредности које се шаљу док се мења интензитет светлости на фото-отпорнику, као и да покажу да схватају шта ради код. Задатак се признаје и без C кода у случају да валидно ради и са *Scratch* имплементацијом.

**Изазов 5** Цртање графика

**Област:** Програмирање софтвера

Потребно је написати програмски код за *Processing* библиотеку којим се исцртава график. Програм треба да прати Y-координату миша и исцртава је на графику сваке секунде. У случају да је позиција миша ван опсега цртања треба је мапирати на опсег (0...height)

Да бисмо нацртали вредности које се мере на *Arduino* уређају, потребно је да декларишемо једну променљиву која представља позицију на x-оси где треба нацртати измерену вредност:

```
int xPos = 1; // хоризонтална позиција графика
float value = 0; // вертикална вредност која ће бити нацртана
```

Прво је потребно иницијализовати апликацију у функцији *setup()*:

```
void setup () {
  size(400, 300);
  background(0);
  stroke(155);
}
```

300 представља максималну висину апликације (height=300). Функција *draw()* из *Processing* библиотеке треба да нацрта тачку на позицији (xPos, height-value). Ако је x-позиција дошла до краја x-осе треба да је врати на почетак како би се опет исцртавала са леве стране. Ако x-позиција није дошла до краја прелази се на следећу позицију по x-оси и одреди следећу вредност коју треба нацртати.

```
void draw () {
  // Нацртај тачку:
  point(xPos, height - value);

  // Ако је x-позиција дошла до краја x-осе треба да је врати на почетак:
  if (xPos >= width) {
    xPos = 0;
    background(0);
  } else {
    // Помери се за 1 по x-оси како би се нацртала следећа вредност:
    xPos = xPos+1;
    // Одреди следећу вредност коју треба нацртати:
    value = mouseY%400;
  }

  delay(1000); // Сачекај секунд пре следећег исцртавања
}
```

**Напомена:** За максимални број бодова је потребно да такмичари прикажу како се мења график док померају позицију миша. Грешке у коду (нпр. коришћење < уместо <=) којима се не покривају неки случајеви доносе негативне поене.

**Изазов 6**    Приказивање резултата мерења    **Област:** Програмирање софтвера

Потребно је имплементирати код у *Processing* окружењу којим се читају вредности измереног напона са *Arduino* уређаја и те вредности се приказују на графику. Програмски код треба да сваке секунде прочита вредност коју шаље *Arduino* уређај путем серијског порта и ту вредност нацрта на линијском графику.

### Решење

Потребно је укључити библиотеке за рад са *Arduino* уређајем и декларисати серијски порт путем ког ће *Arduino* уређај слати податке:

```
import processing.serial.*;
```

```
Serial arduinoPort;    // Серијски порт путем кога Arduino уређај шаље податке
```

Да бисмо нацртали вредности које се мере на *Arduino* уређају, потребно је да декларишемо једну променљиву која представља позицију на x-оси где треба нацртати измерену вредност:

```
int xPos = 1;    // хоризонтална позиција графика
```

```
float value = 0;    // вертикална вредност која ће бити нацртана
```

Прво је потребно иницијализовати апликацију у функцији *setup()*:

```
void setup () {  
  size(400, 300);  
  background(0);  
  stroke(155);  
  arduinoPort = new Serial(this, Serial.list()[0], 9600);
```

```
  // Не позивај serialEvent() све док не добијеш карактер који представља нову линију:
```

```
  arduinoPort.bufferUntil('\n');
```

```
}
```

Потребно је написати функцију која ће реаговати на поруке које *Arduino* уређај шаље путем серијског порта.

```
void serialEvent (Serial arduinoPort) {  
  // читај поруку коју шаље Arduino до првог новог реда:  
  String data = arduinoPort.readStringUntil('\n');
```

```
  if (data != null) {  
    // претвори текст у број и мапирај га на величину прозора:  
    value = float(data);  
    value = map(value, 0, 1023, 0, height);  
  }  
}
```

Функција *draw()* из *Processing* библиотеке треба да нацрта тачку на позицији (xPos, height-value). Ако је x-позиција дошла до краја x-осе треба да је врати на почетак како би се опет исцртавала са леве стране. Ако x-позиција није дошла до краја прелази се на следећу позицију по x-оси.

```
void draw () {  
  // Нацртај тачку:  
  point(xPos, height - value);  
  
  // Ако је x-позиција дошла до краја x-осе, треба да је врати на почетак:  
  if (xPos >= width) {  
    xPos = 0;  
    background(0);  
  } else {  
    // Помери се за 1 по x-оси како би се нацртала следећа вредност:  
    xPos = xPos+1;  
  }  
}
```

**Напомена:** За максимални број бодова је потребно да такмичари прикажу како се мења график док се мења интензитет светлости на фото-отпорнику. Грешке у коду (нпр. коришћење < уместо <=) којима се не покривају неки случајеви доносе негативне поене. Квалитет, елегантност и једноставност програмског кода могу да утичу на број бодова.