

# Дигитални изазов – Мерење брзине

Тема овог Дигиталног изазова је **детектовање растојања и мерење брзине**. Потребно је пројектовати хардверски уређај којим се детектује растојање објекта од сензора и његова брзина. Кроз 6 проблемских изазова из различитих области је потребно доћи до финалног решења.

	Тема	Поени (тачно решење)	Додатни поени	
			Брзина	Квалитет
1	Физика	3	1..5	
2	Физика	3	1..5	7
3	Дизајн уређаја ( <i>Arduino</i> , <a href="#">tinkercad</a> )	10	1..5	5
4	Дизајн и симулација уређаја ( <i>Arduino</i> , <a href="#">tinkercad</a> )	30	1..5	1..10
5	Програмирање софтвера ( <i>Processing</i> )	30	1..5	1..12
6	Интеграција хардвера и софтвера ( <i>Arduino</i> , <i>Processing</i> )	50	1..5	1..10
	Укупно бодова:	126	6..30	15...44
		Максимални број поена:		200

У наставку су поставке задатака, решења и критеријуми за бодовање.

Изазов 1	Детектовање растојања	Област: Физика
<p>Сонар емитује звучни сигнал који путује кроз ваздух брзином 340m/s, одбија се од тела које је детектовао и враћа назад. Микрофон сонара детектује да се сигнал вратио. Ако је сонар детектовао послати сигнал после 10 милисекунди од слања, колико је детектовано тело удаљено од сонара?</p>		
<p><b>Решење</b> Звучни талас путује до тела које је на растојању <math>d</math> за <math>\Delta t/2=5</math> милисекунди, и прелази растојање <math>d=\Delta t \cdot u/2</math>. Растојање објекта од сонара је <math>340\text{m/s} \cdot 0.01\text{s}/2=1.7\text{m}</math>.</p>		
<p><b>Критеријум оцењивања</b></p> <ol style="list-style-type: none"> <li>1. Тачно решење задатка доноси 3 поена</li> <li>2. Првих пет тимова који доставе задатак имејлом добијају 5,4,3,2,1 поена у редоследу пристизања решења.</li> </ol> <p><b>Максимални број поена: 3+5=8</b></p>		
Изазов 2	Детектовање брзине	Област: Физика
<p>Сонар сваке секунде емитује звучни сигнал који путује кроз ваздух брзином 340m/s, одбија се од тела које је детектовао и враћа назад ка сонару. Микрофон сонара детектује одбијени сигнал. Ако је после првог емитовања сонар детектовао да се сигнал вратио за 10 милисекунди, а после другог емитовања да се вратио после 8 милисекунди, којом брзином се креће детектовани објекат?</p>		
<p><b>Решење</b> Пошто је друго време краће, објекат се креће ка сонару. Први звучни талас путује до тела које је на удаљености <math>d_1</math> за <math>\Delta t_1/2=5\text{ms}</math> и прелази растојање <math>d_1=\Delta t_1 \cdot u/2</math>. Објекат је детектован на растојању <math>d_1 = \Delta t_1 \cdot u/2 = 1.7\text{m}</math>. Други звучни талас путује до тела које је на дистанци <math>d_2</math> за <math>\Delta t_2/2=4\text{ms}</math> и прелази растојање <math>d_2=\Delta t_2 \cdot u/2</math>. Објекат је детектован на растојању <math>d_2 = \Delta t_2 \cdot u/2 = 1.36\text{m}</math>.</p>		

**Приближно решење:** Први талас је детектовао објекат на  $d_1 = 1.7\text{m}$ , а други талас који је емитован следеће секунде га је детектовао на  $d_2 = 1.36\text{m}$ . Објекат је прешао растојање  $d_1 - d_2 = 0.34\text{m}$  док су емитована два таласа у размаку од  $\Delta t = 1\text{s}$ , тако да му је брзина  $\frac{d_1 - d_2}{\Delta t} = \frac{0.34\text{m}}{1\text{sec}} = 0.34\text{m/s}$ .

**Прецизно решење:** Нека је први талас емитован у 00:00.000. Талас путује  $\Delta t_1/2 = 5\text{ms}$  до објекта, прелази  $d_1 = \Delta t_1 * u/2 = 1.7\text{m}$ , погађа га у  $t_1 = 00:00.005$  и одбија се. Други талас се емитује у 00:01.000, путује до објекта после  $\Delta t_2/2 = 4\text{ms}$  у тренутку  $t_2 = 00:01.004$ , прелази растојање  $d_2 = \Delta t_2 * u/2 = 1.36\text{m}$ , одбија се и враћа назад у 00:01.008. Разлика путева је  $d_1 - d_2 = 0.34\text{m}$ , али је разлика времена за  $\Delta t = 00:01.004 - 00:00.005 = 0.999\text{s}$ . Пошто је објекат прешао 0.34m за 0.999s, тачна брзина објекта је  $0.34\text{m}/0.999\text{s} = 0.\overline{340}\text{m/s}$ .

#### Критеријум оцењивања

1. Приближно решење задатка доноси 3 поена.
2. Првих пет тимова који доставе задатак имејлом добијају 5,4,3,2,1 поена у редоследу пристизања решења.
3. Прецизно решење задатка доноси додатних 7 поена (додају се на поене).

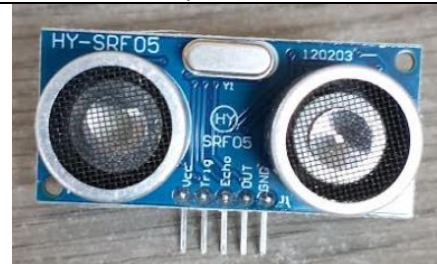
**Максимални број поена: 3+5+7=15**

#### Изазов 3

Дизајн *Arduino* мерача растојања

**Област:** Електротехника

Потребно је дизајнирати шему којом се помоћу *Arduino* уређаја и ултразвучног сензора мери удаљеност или брзина објекта. Ултразвучни сензор растојања има улазе за напајање од 5V (Vcc), уземљење (GND), улаз којим се окида слање сигнала (TRIG) и излаз којим се јавља да је сигнал враћен (ECHO).

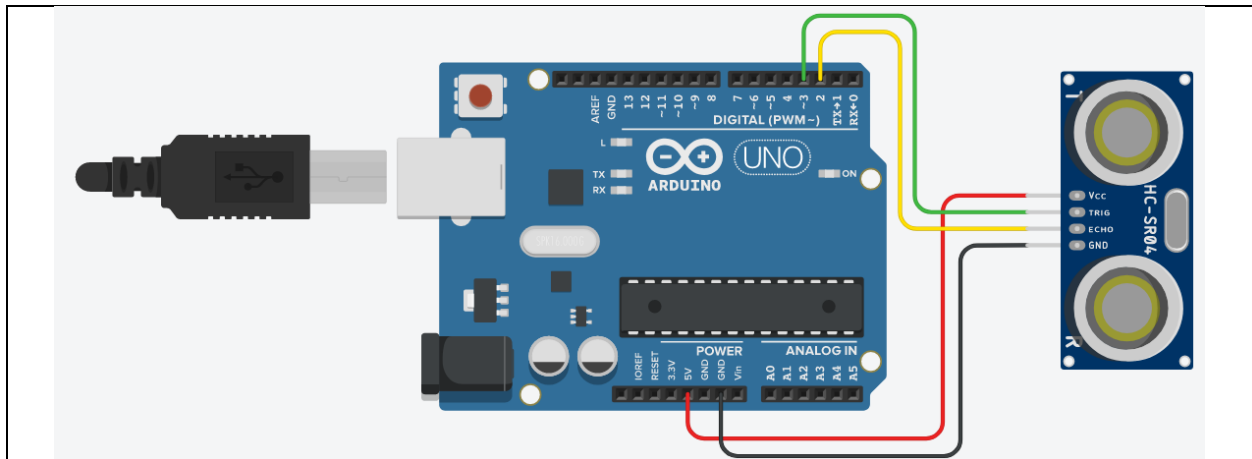


Шему је потребно дизајнирати у [tinkercad](https://www.tinkercad.com) веб апликацији (у дизајну се може користити и други сличан тип сензора који постоји у [tinkercad](https://www.tinkercad.com) веб апликацији, као на пример HY-SRF04 који има Trig/Echo прикључке) и покренути симулацију. За додатне поене пожељно је користити протоборд (енгл. *protoboard*, *breadboard*) на дизајну зато што ће се користити и у физичкој имплементацији.

#### Решење

Сензор се повезује на *Arduino* уређај тако да се напајање сензора повеже са *Arduino* излазним пином од 5V, а GND улаз сензора на GND излаз *Arduino* уређаја. Пинови TRIG и ECHO се повезују на дигиталне пине *Arduino* уређаја како би се слао и читавао сигнал.

Дизајн (без протоборда) је приказан на слици:



Излазом A3 (зелена) се „окида“ сензор да пошаље нови сигнал, а улазом A2 (жута) се детектује да је примљен повратни сигнал.

#### Критеријум оцењивања

1. Тачан дизајн доноси 10 поена. Признаје се било какав начин повезивања који је еквивалентан о оваквој шеми чак и без прото-борда.
2. Првих пет тимова који доставе задатак имејлом добијају 5,4,3,2,1 поена у редоследу пристизања решења.
3. Дизајн са протобордом доноси 5 додатних поена.

**Максимални број поена: 10+5+5=20**

**Изазов 4** | Мерење брзине *Arduino* уређајем

**Област:** Програмирање хардвера

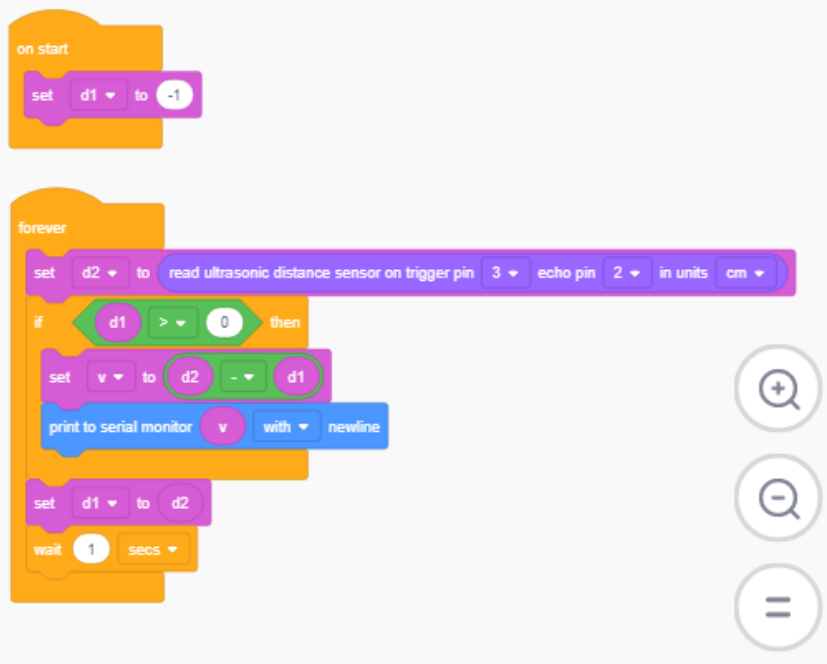
Потребно је имплементирати код (визуелно или на C језику) у [tinkercad](https://www.tinkercad.com) веб симулатору којим се програмира *Arduino* уређај за мерење брзине објекта ког детектује. Код мора да уради следеће операције:

- Сваке секунде шаље инструкцију сензору да пошаље нови сигнал.
- Мери време до повратног сигнала и рачуна брзину апроксимативном формулом.
- Шаље брзину преко серијског порта.

Потребно је послати слику алгоритма или програмски код и покренути симулатор којим се доказује да уређај детектује брзину.

#### Решење

Визуелни код којим се имплементира оваква функционалност је приказан у примеру:



У алгоритму се сваке секунде чита вредност раздаљине са ултразвучног сензора тако што се шаље сигнал за окидање преко пина А3, а чита одговор преко пина А2. Брзина се рачуна као разлика растојања пошто не мора да се дели са временом које је 1. Пре следеће итерације се претходно измерено растојање ( $d_1$ ) поставља на вредност тренутно измереног растојања ( $d_2$ ) како би било спремно за следеће мерење. У следећој итерацији ће се очитати ново растојање  $d_2$  и од њега ће се одузети тренутно које је сада уписано у  $d_1$ .

Функција `readUltrasonicDistance(triggerPin, echoPin)` која чита удаљеност објекта од сензора је приказана у примеру:

```
long readUltrasonicDistance(int triggerPin, int echoPin)
{
  pinMode(triggerPin, OUTPUT); // ресетовање окидача
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  // Поставља пин окидача на вредност 1 на 10 микросекунди
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);
  pinMode(echoPin, INPUT);
  // Чита echo пин и враћа измерено време у микросекундама
  return pulseIn(echoPin, HIGH);
}
```

**Напомена:** За максимални број бодова је потребно да такмичари прикажу на серијском порту вредности брзине која се детектује док се објекат креће ка сензору, као и да покажу да схватају како ради код.

#### Критеријум оцењивања

1. Тачно решење које приказује у симулатору брзину доноси 30 поена.

2. Првих пет тимова који доставе задатак имејлом добијају 5,4,3,2,1 поена у редоследу пристизања решења.
3. На квалитет кода и решења се може добити још 10 поена.

**Максимални број поена: 30+5+10=35**

**Изазов 5** | Цртање графика брзине | **Област:** Програмирање софтвера

Потребно је написати програмски код за *Processing* библиотеку којим се симулира цртање графика промене брзине објекта.

Потребно је нацртати координатни систем са стрелицама које представљају координатне осе, баждарити осе вредностима (нпр. 10, 20, 30,...height на y осе) како би се виделе вредности.

Промена брзине објекта се симулира стрелицама ↓ и ↑ на тастатури. Програм треба да повећава/умањује брзину ако је притиснута стрелица на горе или на доле (на тастатури) и та вредност треба да се прикаже тако да наставља линија у виду линијског или сличног дијаграма. У случају да је симулирана брзина ван опсега цртања треба је мапирати на опсег (0...height). Када се исцрта крајња десна вредност, вредност x осе се „ресетује“ нулу и нове вредности се исцртавају из почетка координата тако да пребришу претходне вредности.



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:
if ( keyPressed ) {
  if ( keyCode == UP )
    value = value + 1;
  else if ( keyCode == DOWN )
    value = value - 1;
}
value = value % height;
}

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

```

#### Критеријум оцењивања

1. Тачно решење које приказује брзину доноси 20 поена са осама које не морају да имају стрелице или обележене вредности.
2. Првих пет тимова који доставе задатак имејлом добијају 5,4,3,2,1 поена у редоследу пристизања решења.
3. Решење на коме су приказане вредности на x и y осама доноси додатних 5 поена.
4. За квалитет кода (одсуство багова, исправно брисање претходне вредности, код који не брише вредности оса) се добија између 1 и 7 поена.

**Максимални број поена: 30+5+5+7=47**

<b>Изазов 6</b>	Приказивање резултата мерења	<b>Област:</b> Интеграција хардвера и софтвера
-----------------	------------------------------	--

Потребно је интегрисати физички *Arduino* уређај који користи сензор удаљености са апликацијом на основу дизајна шеме којом се мери брзина објекта.

1. Потребно је имплементирати електронску шему у којој је уређај повезан са сензором удаљености и имплементирати програмски код којим се мери брзина објекта и шаље на серијски порт, у складу са дизајном и симулатором.
2. Потребно је имплементирати код у *Processing* окружењу којим се читају вредности брзине коју шаље *Arduino* уређај 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 декларирасмо 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); // Некад ово неће бити 0-ти порт!!!

  // 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;
  }
}
```

#### Критеријум оцењивања

1. Тачно решење које приказује брзину доноси 50 поена, чак и ако нема обележене вредности на осама.
2. Првих пет тимова који доставе задатак имејлом добијају 5,4,3,2,1 поена у редоследу пристизања решења.

3. За квалитет решења се добија до 10 додатних поена.

**Максимални број поена:  $50+5+10=65$**

**Напомена** – комисија задржава право да зада 1-2 џокер задатака у зависности од брзине израде регуларних задатака.