



# Ievads Iot izstrādē

3.Pogas un pirmie sensori

Pasākumu finansē: Eiropas Jūrlietu un zivsaimniecības fonds, projekts "Viedo ciemu attīstība piekrastes teritorijās". Nr. 19-00-F043.0443-00001.  
VRG "Partnerība laukiem un jūrai", sadarbībā ar vadošo partneri VRG "Liepājas rajona partnerība"



NACIONĀLAIS  
ATTĪSTĪBAS  
PLĀNS 2020

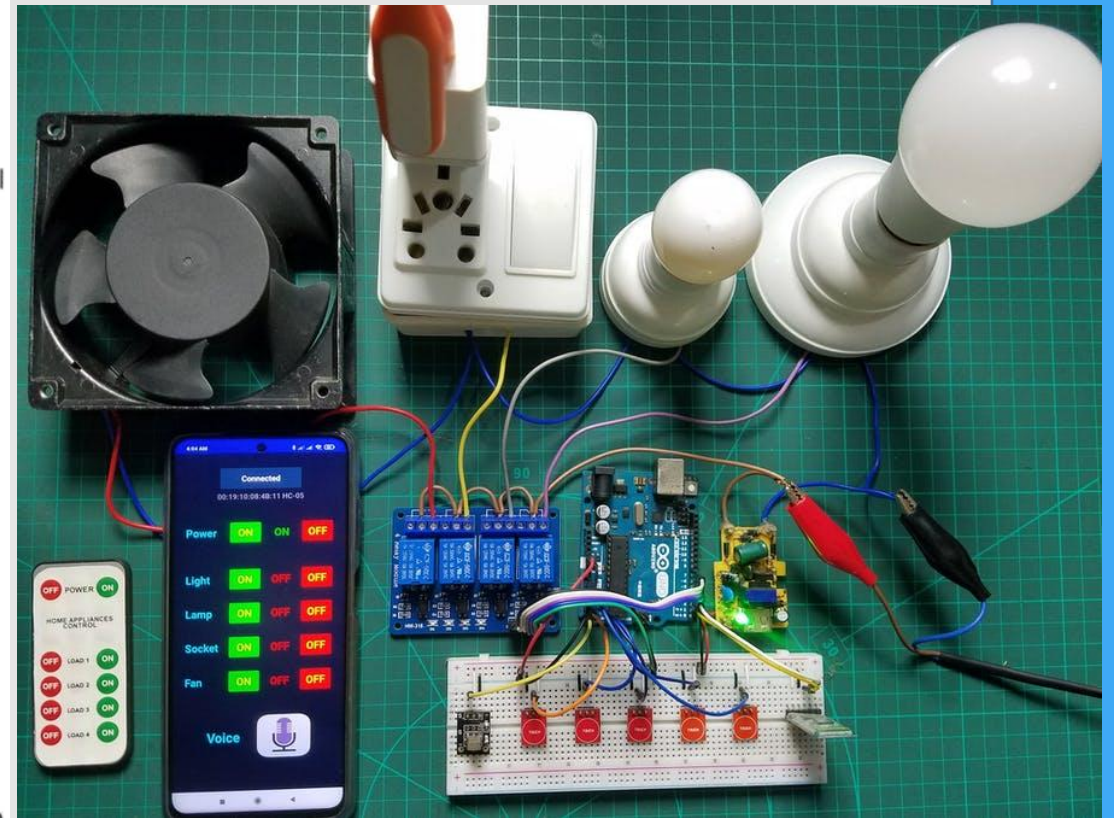
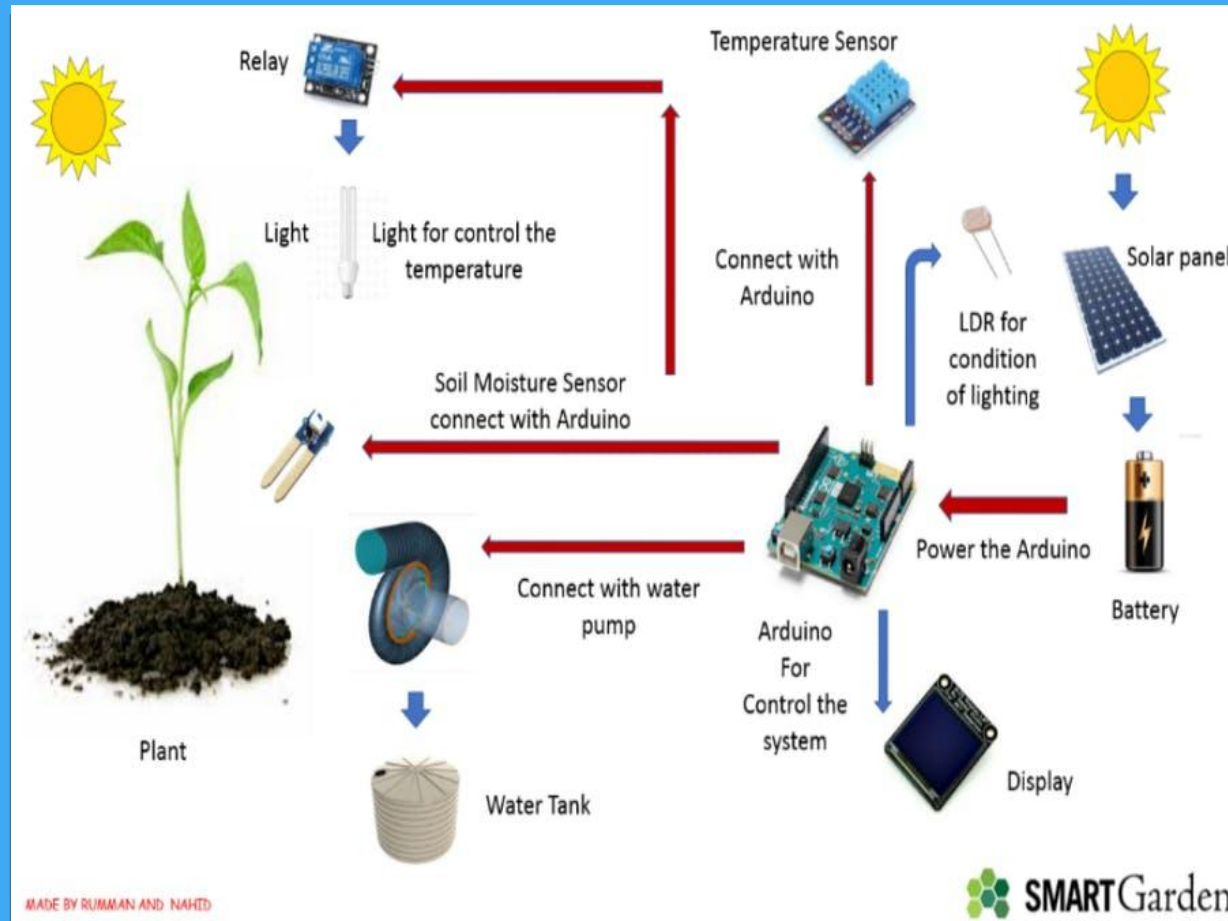


EIROPAS SAVIENĪBA  
Eiropas Jūrlietu un  
zivsaimniecības fonds



Atbalsta Zemkopības ministrija un Lauku atbalsta dienests

# Kam nepieciešami sensori?



# Pienācis laiks izmantot ievadi ar pogām un sensoriem.

Kā bija minēts pirmajā lekcijā, izvadi var nolasīt datus un sūtīt datus. Tomēr, lai pateiktu mikrokontrolierim, ka mēs gribam sūtīt pogas vai sensora stāvokli.

Izmanto:

```
pinMode(poga_vai_sensors, INPUT);
```

# Rezultāts

```
//Koda šablons
//Lekcija: XXXXXX

//Vieta definētiem izvadiem -----//
#define POGA 12

//Vieta globāliem mainīgajiem -----//

void setup() { // FUNKCIJAS SĀKUMS (Atceramies kods šajā daļā izpildas vienu reizi pēc Arduino ieslēgšanās)

//Vieta izvadu inicializācijai -----//
pinMode(POGA, INPUT);

//Vieta kodam, kas izpildas vienu reizi -----//

} // FUNKCIJAS BEIGAS
```

# Tas nav viss

- Mums ir nepieciešams nolasīt un glabāt vērtību, lai varētu veikt dažādas operācijas. Priekš tā ir nepieciešams ievietot to mainīgajā.

Lai mēs nolasītu digitālos datus mēs izmantojam:

```
stavoklis = digitalRead(poga_vai_sensors);
```

Lai mēs nolasītu analogus datus mēs izmantojam:

```
stavoklis = analogRead(poga_vai_sensors);
```

# Kā rezultāts

```
//Lekcija: XXXXXX

//Vieta definētiem izvadiem -----//
#define POGA 12

//Vieta globāliem mainīgajiem -----//
int pogas_stavoklis = 0;

void setup() { // FUNKCIJAS SĀKUMS (Atceramies kods šajā daļā izpildas vienu reizi pēc Arduino ieslēgšanās)

//Vieta izvadu inicializācijai -----//
pinMode(POGA, INPUT);

//Vieta kodam, kas izpildas vienu reizi -----//

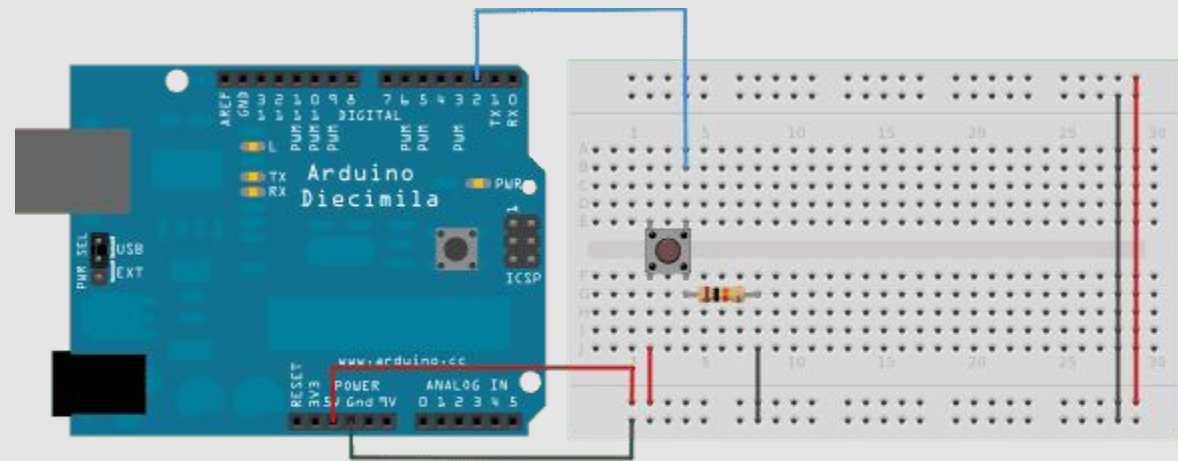
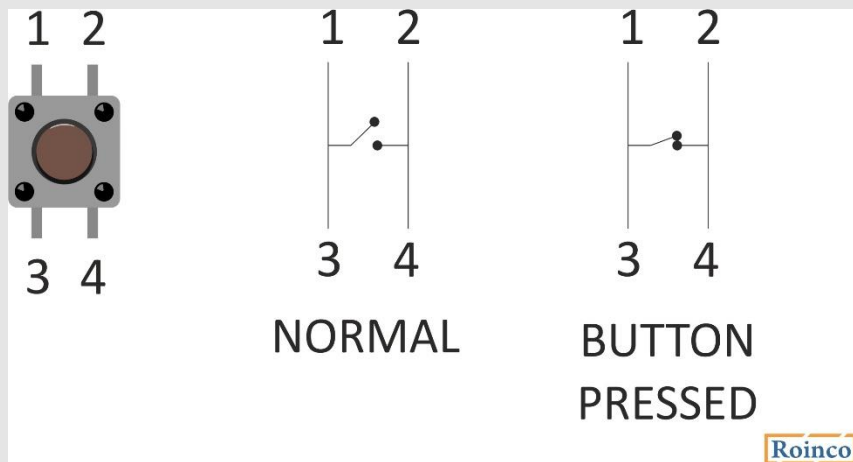
} // FUNKCIJAS BEIGAS

void loop() { // FUNKCIJAS SĀKUMS (Kods šajā daļā atkārtosies mūžīgi)
//Vieta kodam, kas atkārtojas -----//

pogas_stavoklis = digitalRead(POGA);

} // FUNKCIJAS BEIGAS
```

# Pogas pieslēgšanas shēma



# 1.uzdevums

Nolasīsim pogas vērtību un izvadīsim seriāla monitorā.



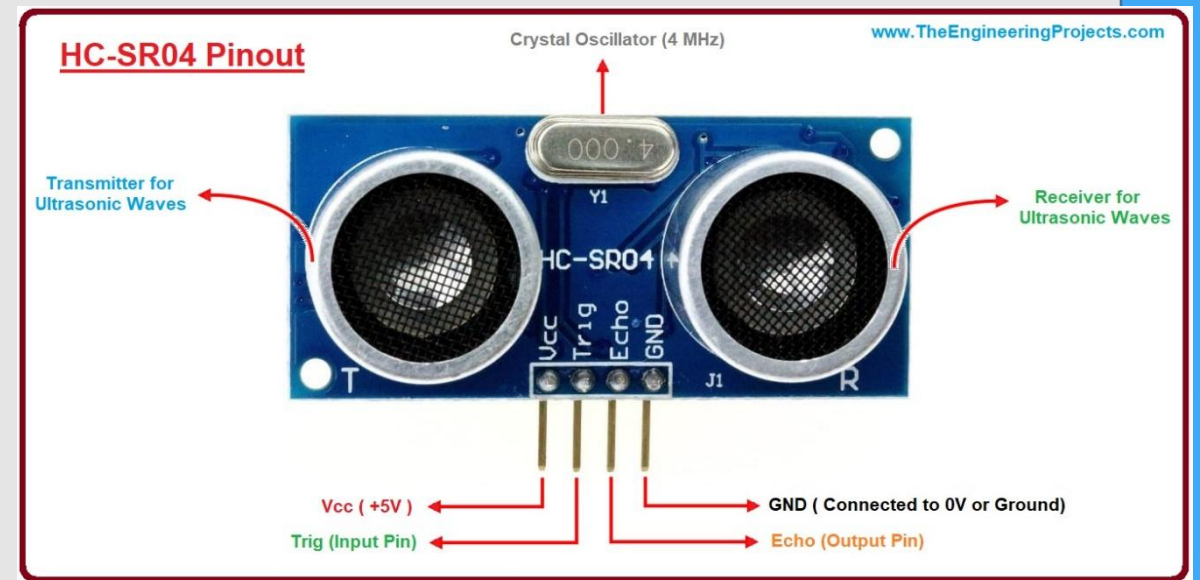
# Beidzot nonācām līdz pirmajam sensoram

- Iepazīstaties: Attāluma sensors HC-04

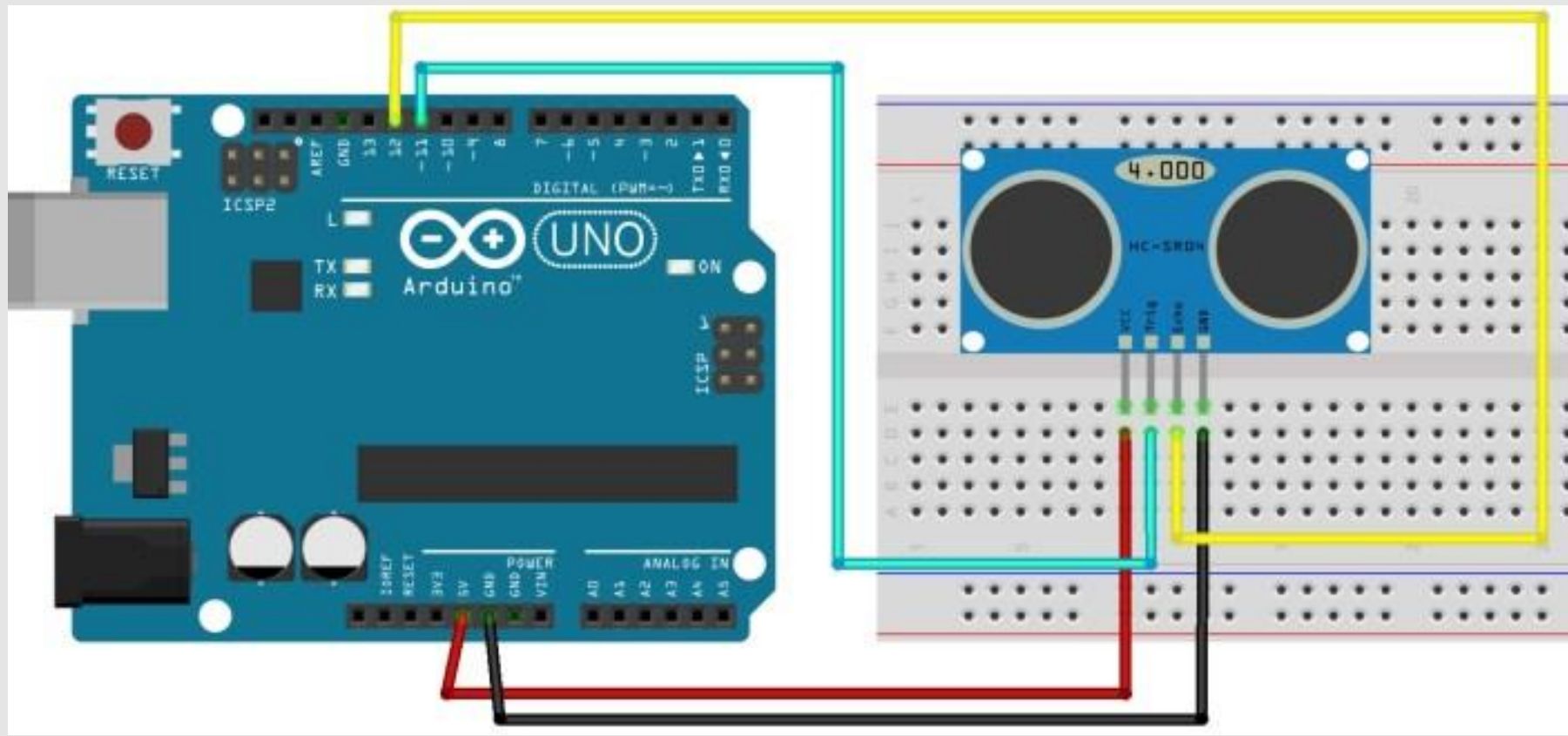


# Sensora darbība

Lai darbinātu sensoru ar Arduino jāizprot, kam jābūt pieslēgtam, kas saņem signālu vai, kas sūta signālu. Ja viss būs veiksmīgi saņemsim attālumu no priekšmeta līdz sensoram.



# Mēģināsim pieslēgt sensoru



# Programmēsim Arduino

Kā jau minēju mums nepieciešams uzrādīt izvadus, kuri mums nepieciešami un mainīgie, kuros varam glabāt vērtības.

```
//Vieta definētiem izvadiem -----//  
#define trigPin 11  
#define echoPin 12  
  
//Vieta globāliem mainīgajiem -----//  
float laiks = 0.0;  
float cm = 0.0;
```

# Atceramies pateikt kas sūta, kas saņem.

Arduino nepieciešams zināt, kuri izvadi izmantosies tāpēc iestatām:

```
void setup() { // FUNKCIJAS SĀKUMS (Atceramies kods šajā daļā izpildas vienu reizi pēc Arduino ieslēgšanās)

//Vieta izvadu inicializācijai -----//
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);

//Vieta kodam, kas izpildas vienu reizi -----//

} // FUNKCIJAS BEIGAS
```

# Skaņas signāla sūtīšana

`delayMicroseconds(1000) == delay(1)`

```
void loop() { // FUNKCIJAS SĀKUMS (Kods šajā daļā atkārtosies mūžīgi)
//Vieta kodam, kas atkārtojas -----//

digitalWrite(trigPin, LOW);
delayMicroseconds(5);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);

} // FUNKCIJAS BEIGAS
```

# Un saņemam atstaroto signālu

Atstaroto signālu var salīdzināt ar skrējēju, kas pieskaroties sienai skrien atpakaļ un līdz tas sasniedzis finišu, Arduino apstādina taimeri un iegūst laiku.

```
void loop() { // FUNKCIJAS SĀKUMS (Kods šajā daļā atkārtosies mūžīgi)
//Vieta kodam, kas atkārtojas -----//

digitalWrite(trigPin, LOW);
delayMicroseconds(5);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);

laiks = pulseIn(echoPin, HIGH);

} // FUNKCIJAS BEIGAS
```

# Kāds ir skaņas ātrums?

Attālums = (veiktā ceļa laiks / 2) \* skaņas ātrums



## 2.Uzdevums

Izvadam uz seriālā monitora attālumu mērvienības ik pēc 0.5 sekundēm

# 3.Uzdevums

Pieslēgt sensoru un RGB gaismas diodi atkarībā no attāluma spīdēs viena no trīs krāsām.

