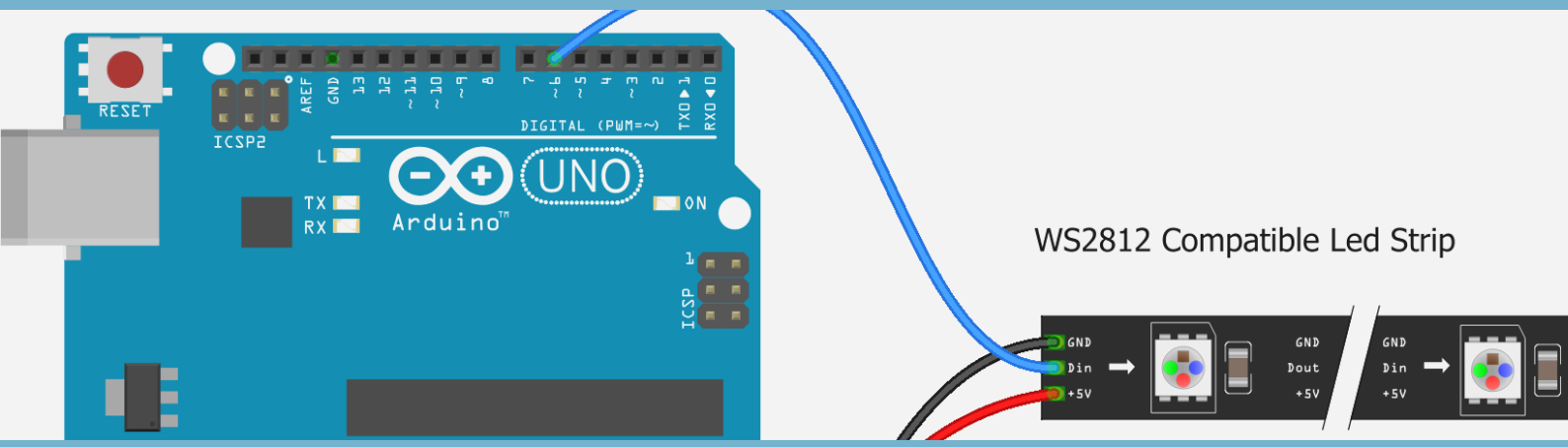
JAUNRADES NAMS JUNDA

LED GAISMAS VIRKNE

autors UĢIS PEKŠA

2020



MĒRĶIS

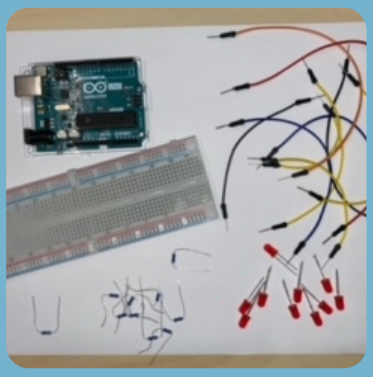
Iemācīt izmantot programmas Arduino IDE. Ciklu un nosacījumu operatorus veidojot LED gaismas virkni.

UZDEVUMI

- Izveidot un savienot detaļas atbilstoši shēmai, lai izveidotos 10 LED diožu virkne.
- Uzprogrammēt programmu, kura nodrošinās LED diožu darbību virknē.

NEPIECIEŠMIE MATERIĀLI

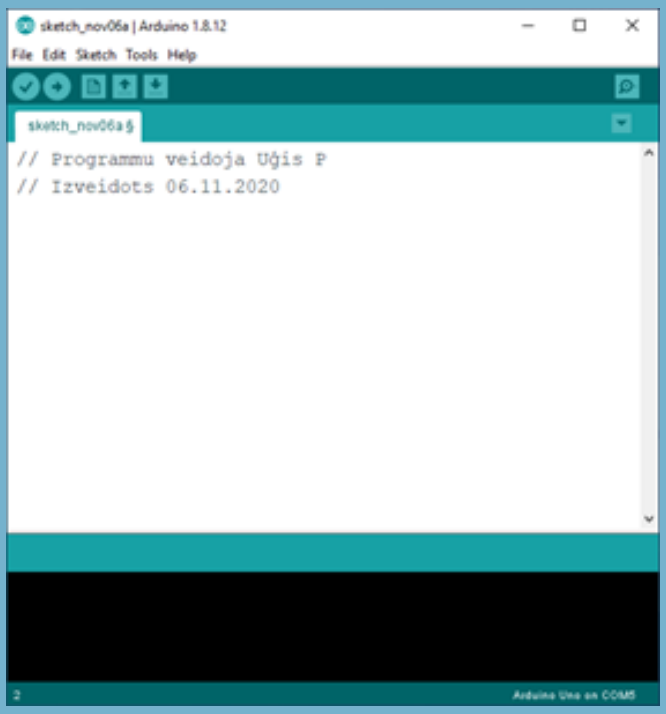
1. Arduino UNO pamatplate(der arī citas arduino pamatplates)
2. Detaļu savienojumu pamatplate (bredboard)
3. 10 gab. LED diodes, jebkādā krāsā
4. 10 gab 220 omu pretestības;
5. Vadi savienošanai;
6. Dators ar instalētu bezmaksas Arduino IDLE programmatūru;



MĒRĶAUDITORIJA

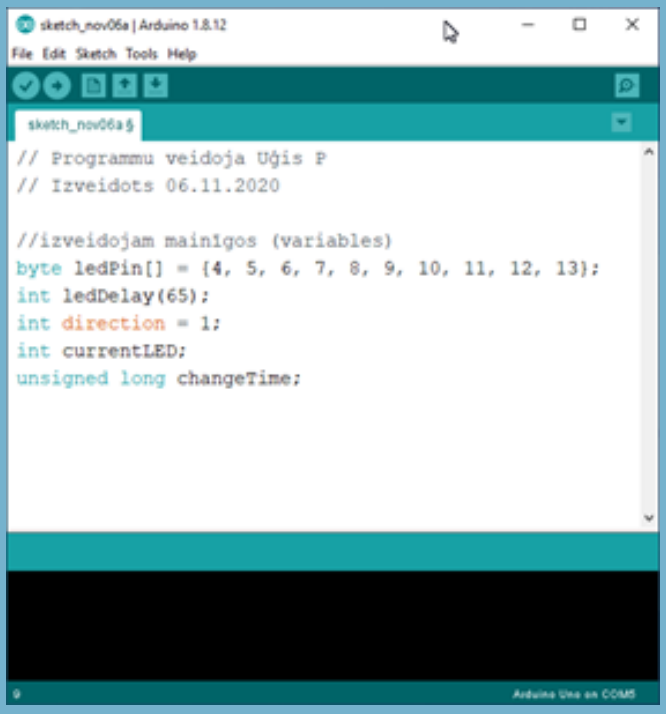
6. – 12. klašu skolēni

1. solis: Sāksim ar kodu



Atveriet savu Arduino IDE. Izdzēšam visu tekstu, kas jau ir ierakstīts, jo mēs izveidosim savu. Skripta augšdaļā ievietojiet savu vārdu un datumu un komentējiet tos, rindas sākumā ievadot //

2. solis: izveidojam mainīgos



Lai vēlāk programmas laikā mums pašiem būtu vieglāk, vislabāk būs, ja izveidosim mainīgos skaitļiem vai vienumiem, ar kuriem mēs visā programmas laikā turpmāk strādāsim.

byte ledPin[] = {4, 5, 6, 7, 8, 9, 10, 11, 12, 13};

Tiek izveidots masīvs. (saraksts ar noteiktām vērtībām tajā), kur mūsu 10 diodes tiks izvietotas uz slēguma dēlīša (bredboard). Kad sāksim savienot projektā fiziski detaļas un diodes tad katra diode būs savienota ar konkrēto Arduino izeju.

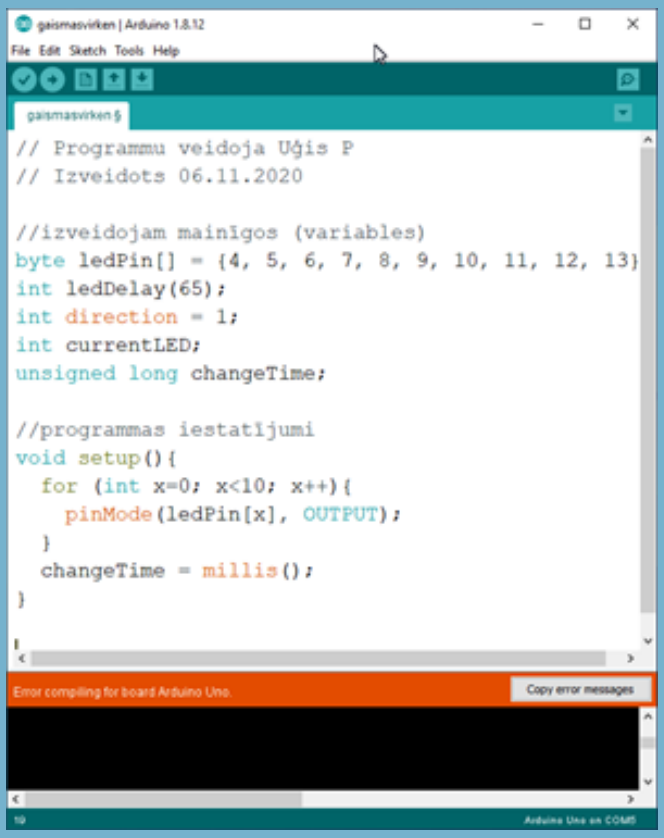
int ledDelay (65);
Tas izraisīs 65 milisekunžu kavēšanos.

int Direction = 1;
Tā mēs mainīsim, kādā virzienā virzīsies mūsu gaismas.

int currentLED;
Mēs veidojam mainīgo, kas pašlaik ir tukšs. Mēs piešķirsim tai vērtību vēlāk programmā.

unsigned long changeTime;
Šis ir vēl viens mainīgais, kuru mēs pašlaik atstājam tukšu. Arī šo mainīgo izmantosim vēlāk programmā.

3. solis: programmas sākuma iestatījumi



Izveidosim funkciju, kas izpildās tikai vienu reizi. Funkcija izpildās tajā momentā kad programma tiek palaista pirmo reizi un viss. Apskatīsim kodu pa soļiem.

void setup () {
Tas izsauc funkcijas ar nosaukumu setup (funkcija ir rezervēta Arduino IDE) un palaiž to vienreiz.

for (int x = 0; x <10; x ++) {
šī ir cikls for. Vispirms mēs inicializējam veselu skaitli x un iestatām to uz 0, pēc tam sakām, lai tas darbojas tik ilgi, kamēr x ir mazāks par 10, katrā nākamajā cikla darbības solī mēs x palielinām pa 1.

pinMode (ledPin [x], OUTPUT);
Veids kā mēs iestatām un pasakām kura Arduino izeja mums darbosies.

}
Tas aizver for ciklu.

changeTime = milis ();
Tas iestata mainīgo timeTime uz funkcijas milis () vērtību. millis () ir iebūvēta funkcija, kas atgriež laika periodu kopš programmas darbības sākuma.

}
Tas aizver funkciju setup.

4. Solis: izveidojam funkciju kas pārslēdz LED diodes

```
//programmas iestatījumi
void setup(){
  for (int x=0; x<10; x++){
    pinMode(ledPin[x], OUTPUT);
  }
  changeTime = millis();
}

//funkcija LED diožu pārslēgšanai
void changeLED() {
  for (int x=0; x<10; x++){
    digitalWrite(ledPin[x], LOW);
  }
  digitalWrite(ledPin[currentLED], HIGH);
  currentLED += direction;
  if (currentLED == 9){
    direction = -1;
  }
   if (currentLED == 0){
    direction = 1;
  }
}
```

Un mūsu izveidotās funkcijas paskaidrojums

void changeLED () {
Šis ir mūsu funkcijas sākums, ko mēs sauksim par changeLED ().

for (int x = 0; x <10; x ++) {

Tas ir tieši tas pats cikls, ko mēs rakstījām iepriekš. Mēs ņemam veselu skaitli x un iestatām to uz 0 un palaižam ciklu, katru reizi pievienojot vienu, līdz mēs sasniedzam 10.

digitalWrite (ledPin [x], LOW);

Ar šo rindiņu mēs vienkārši izslēdzam Led diodi.

}

Tas aizver for ciklu.

digitalWrite (ledPin [currentLED, HIGH); :

Tas iestatīs LED pašreizējā LED pozīcijā (atcerieties, ka currentLED patiešām apzīmē skaitli, kā mēs to inicializējām sākumā) un ieslēgs LED diodi.

currentLED + = direction;

Šo rindu varētu uzrakstīt arī kā currentLED = currentLED + direction;.
Tas ņem pašreizējā LED saglabāto kārtas numuru un pievieno tam virziena vērtību. Tas ir nozīmīgs solis, jo tieši tas ļauj mūsu gaismai pāriet no viena LED uz nākamo.

if (currentLED == 9) {:

Šī ir nosacījuma operators. Tas salīdzinās currentLED vērtību ar 9, un, ja tie ir līdzvērtīgi (==), tad darbosies šāds kods. Ja tas tā nav, tad programma neizpildīsies.

direction = -1;

Kad gaisma sasniedz pēdējo LED diodi, šis kods liek tai iestatīt virziena vērtību -1. Kad changeLED () funkcija ar ciklu darbosies vēlreiz, tā piespiedīs gaismas diodes gaismu virzīties pa kreisi.

}

Tas aizver for ciklu.

if (currentLED == 0) {

Šis ir vēlviens nosacījuma operators, taču šoreiz tas darbosies tikai tad, ja currentLED vērtība ir ekvivalents 0.

direction = 1;

Tas ir līdzīgs iepriekšējam ciklam, taču šoreiz tas iestatīs virzienu, kas ir vienāds ar 1. Kad changeLED () atkal tiek palaists, gaisma sāks kustēties pa labi.

}

Tas aizver for for ciklu.

}

Tas aizver funkciju changeLED ().

5. solis: izveidojam funkciju kas liks programmai strādāt nepārtraukti

```
    direction = -1;
  }
   if (currentLED == 0){
    direction = 1;
  }
}

//Programmas pamata funkcija
void loop(){
  if ((millis()-changeTime)>ledDelay) {
    changeLED();
    changeTime=millis();
    }
}
```

Mēs gandrīz esam pabeiguši šī projekta kodēšanas daļu! Atliek tikai pievienot funkciju, kas liks darboties visam iepriekš rakstītajam kodam.

void loop () {:

tiek sākta funkcija, ko sauc par ciklu. loop () ir iebūvēta funkcija, un tāpēc tā ir iepriekš ieprogrammēta automātiskai darbībai.

if ((millis () - changeTime)> ledDelay) {

pārbaudiet, vai kopš pēdējās LED diodes maiņas ir bijusi pauze vismaz ledDelay (mūsu gadījumā 65) milisekundes vērtībā. Ja tas tā ir, tad turpinās ar šādu kodu.

changeLED ();

Šeit mēs izsaucam savu funkciju changeLED ().

changeTime = millis ();

Tas maina changeTime vērtību par pašreizējo vērtību millis (). Tas palīdz nodrošināt, ka programma vienmēr gaida ledDelay milisekundes pirms gaismas pārslēgšanas uz nākamo LED.

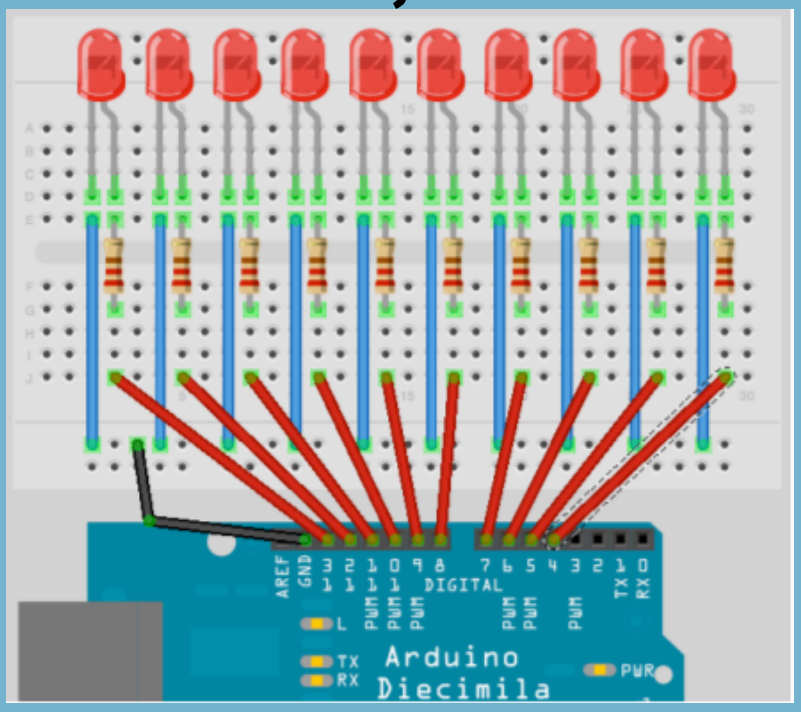
}

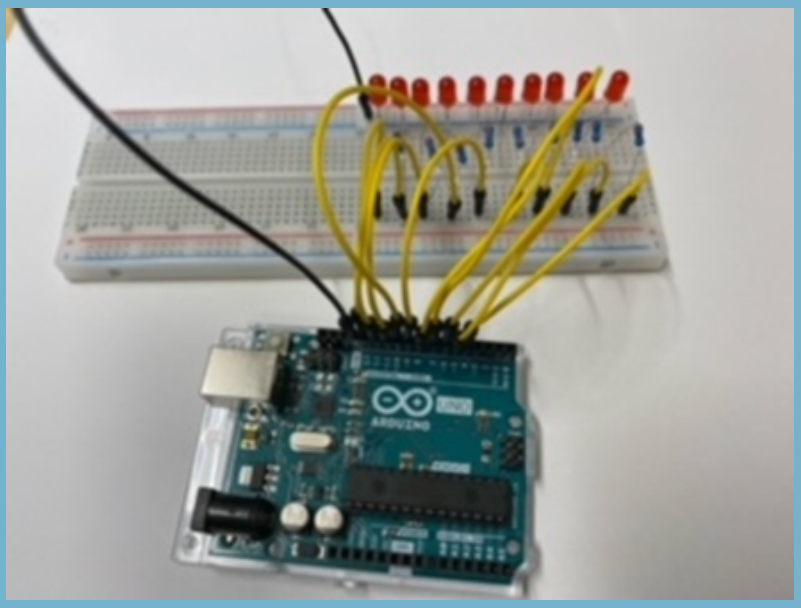
Tas aizver for ciklu.

}

Tas aizver funkciju loop ().

6. solis: izveidojam shēmu





7. solis: lejuplādējam mūsu kodu Arduino Uno pamatplatē un pārbaudām tās darbību