

OMA LIKUMS. DIODES. C++ DATU TIPI.

1.1 Praktiskais darbs #1 (Blink.ino)

Vajadzīgās detaļas: Arduino kontrolieris, USB vads (adapteris no USB-A uz USB-B).

1. Palaist Arduino IDE. Atvērt izvēlni **File > Examples > 01.Basics > Blink**. Saglabāt Blink.ino starp saviem failiem ar **File > Save As**. Piemēram, C:\Users\MansVards\Arduino\Blink\Blink.ino (skripts jāliek direktoriņā ar to pašu nosaukumu).
2. Izvilkt no kastītes kontrolieri un USB vadu. Vienu USB galu pievienot datoram, otru galu pievienot Arduino platei. Pārliecināties, ka uz plates iedegas diode **ON**.
3. Uzstādīt kontroliera tipu un portu. Arduino IDE atvērt **Tools > Board > Arduino AVR Boards > Arduino Uno** un arī **Tools > Port > usb.serial** (vai **Tools > Port > Com**, ja nav minēts USB).
4. Kompilēt programmiņu ar **Sketch > Verify/Compile**.
5. Nosūtīt programmiņu uz Arduino: **Sketch > Upload**. Pārliecināties, ka lampiņa **L** (cita diode uz Arduino plates) sāk mirkšķināt, ik pēc sekundes ieslēdzoties vai izslēdzoties.

```
void setup() {  
  pinMode(LED_BUILTIN, OUTPUT); // uzstāda 13.kontaktu kā izvadi  
}  
  
, kamēr Arduino ir ieslēgts  
void loop() { // Šis cikls atkārtojas, kamēr Arduino ir ieslēgts  
  digitalWrite(LED_BUILTIN, HIGH); // ieslēdz LED ar spriegumu HIGH uz 13. kontakta  
  delay(1000); // gaida 1 sekundi,  
  digitalWrite(LED_BUILTIN, LOW); // izslēdz LED ar spriegumu LOW uz 13.kontakta  
  delay(1000); // gaida 1 sekundi.  
}
```

6. LED_BUILTIN vienāds ar 13 (jo iebūvēto diodi iedez 13.kontakts uz plates). Pārliecinieties par to, LED_BUILTIN visur aizstājot ar skaitli 13 un nosūtot programmu uz Arduino vēlreiz.
- **Pārbaudīt maksimālo spriegumu/strāvu:** Nepārsniedgt atļautos voltus/ampērus. LED slēgt pie 5 V tikai kopā ar rezistoru. Nesaslēgt Arduino kontaktus uz īso.
 - **Lietot iezemējumu:** Katra ierīce ir savienota ar Arduino **GND**.
 - **Atslēgt USB, ja maina ķēdi:** Ķēdi jāpārbauda pirms pieslēdz strāvu.
 - **Tīras rokas:** Rokām jābūt sausām un uz darba galda neturēt ēdamo. (Pēc nodarbības, pēc pieskaršanās metāla kontaktiem un lodējumiem nomazgāt rokas.)

1.2 Elektrotehnika Arduino un LED saslēgšanai

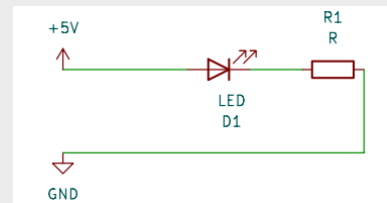
Lai droši lietotu LED ar Arduino, jāizmanto rezistors. Tas neļauj LED caurlaist pārāk lielu strāvu, kas to var sabojāt.

Oma likums:

- **Spriegums (U):** Līdzīgi kā ūdens spiediens caurulē vai ūdenskrituma augstums.
- **Strāvas stiprums (I):** Līdzīgi kā ūdens plūsmas ātrums.
- **Pretestība (R):** Pretojas strāvas plūsmai, līdzīgi kā caurules sašaurinājums ūdenim.

Oma likums: $I = \frac{U}{R}$. Maza pretestība nozīmē ļoti lielu strāvas stiprumu (*īsaslēgums*).

Virknēs slēgums: Ja ierīces saslēdz virknē, tad to kopējā pretestība ir visu ieslēgto pretestību summa: $R = R_1 + R_2 + \dots$. Viena un tā pati strāva plūst caur visām ierīcēm, bet sprieguma kritums uz katras ierīces ir proporcionāls tās pretestībai. Iznēmums ir diode, kurai (no garākās kājiņas uz īsāko) pretestība ir niecīga, bet sprieguma kritums ir aptuveni 2 volti.



Pretestības aprēķināšana:

Ja pieslēdz LED ar 2 V sprieguma kritumu un maksimālo strāvu 20 mA (20 miliampēri jeb 0.020 A) Arduino platei ar 5 V spriegumu, no strāvas avota sprieguma jāatņem LED sprieguma kritums. Uz rezistoru paliek spriegums $5\text{ V} - 2\text{ V} = 3\text{ V}$. Izmantojam Oma likumu $R = \frac{U}{I}$ un atrodam $R = \frac{3\text{ V}}{0.020\text{ A}} = 150\ \Omega$.

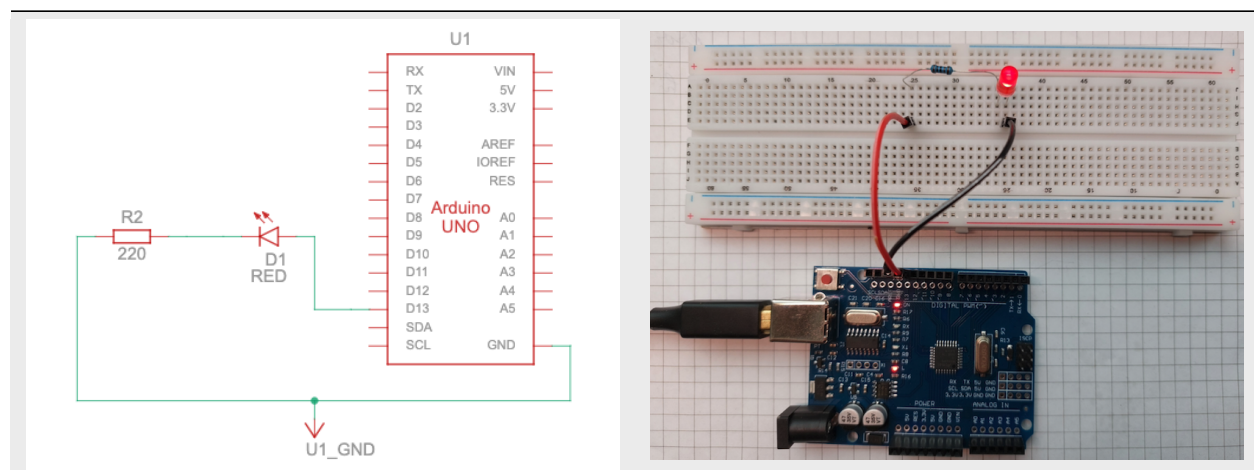
Var izmantot arī 220 Ω rezistoru no komplekta, kas vēl drusku samazinās strāvas stiprumu. Šādā kādē LED droši darbojas kopā ar Arduino!

1.3 Praktiskais darbs #2: Ārēja LED diode (Blink.ino)

Vajadzīgās detaļas:

Arduino kontrolieris, USB vads, maketēšanas plate (*breadboard*), krāsaina LED diode, 220 Ω rezistors, divi savienotājspieņi M-M (ar adatiņām abos galos).

Šajā darbā scenārijs joprojām ir `Blink.ino`, bet pievienosim ārēju diodi.



1. Atvienot Arduino kontrolieri no USB vada. Saslēgt attēlā redzamo ķēdi. LED garāko kājiņu likt tuvāk plusam, īsāko – tuvāk mīnusam. (**Uzmanību:** Pārliedzinieties, ka diode un rezistors ir saslēgti virknē, lai nepārsniegtu maksimālo strāvas stiprumu uz ārējās LED.)
2. Pievienot Arduino kontrolieri USB vadam. Pārliedzināties, ka tagad mirkšķinās gan mazā diode **L** uz Arduino plates, gan arī F5 gaismas diode mūsu ķēdē.

1.4 Praktiskais darbs #3: Potenciometrs (ReadingValues.ino)

Vajadzīgās detaļas:

Arduino kontrolieris, USB vads, potenciometrs, maketēšanas plate, trīs vadiņi M-F (ar adatiņu vienā galā un kontaktligzdu otrā), divi vadiņi M-M (ar adatiņām abos galos).

1. Atvienot vadus no Arduino uz ārējo gaismas diodi LED.
2. Arduino **GND** savieno ar maketēšanas plates “-”, Arduino **5V** savieno ar maketēšanas plates “+”. Potenciometra **GND** savieno ar maketēšanas plates “-”; Potenciometra **VCC** savieno ar maketēšanas plates “+”; Potenciometra **OTA** savieno ar Arduino **A0**.
2. Izveidot jaunu skriptu, saglabāt to kā ReadingValues.ino

```
const int pinAnalog = A0;

void setup(void) {
  Serial.begin (9600);
}

void loop(void) {
  int valInt = analogRead(pinAnalog);
  double valU = (5. / 1023) * valInt;
  String separator = " spriegums:";
  Serial.print(valInt);
  Serial.print(" ");
  Serial.print(A0);
  Serial.print(separator);
  Serial.println(valU , 2);
  delay(100);
}
```

3. Kompilēt un nosūtīt šo skriptu uz Arduino. Novērot strauju **RX** (seriālā porta) lampiņas mirkšķināšanu.
4. No Arduino IDE atvērt **Tools > Serial Monitor**. Pārliedzināties, ka uz datora ekrāna parādās skaitļi (skaitļi no 0 līdz 1023 un spriegums no 0.0 līdz 5.0 voltiem). Pārvietot potenciometra regulatoru, novērot vērtību maiņu.
5. Atvērt **Tools > Serial Plotter**. Pārvietot regulatoru un novērot grafika izmaiņas.
6. Izmainīt Arduino skriptā $(5. / 1023) * valInt$ skaitli 5. ar 5 (bez punkta beigās).
7. No jauna atvērt **Tools > Serial Monitor** un novērot tur redzamās sprieguma vērtības. Kādi skaitļi tur tagad parādās? Vai šīs vērtības ir pareizas?

Table 1: C++ datu tipi

const int	const int pinAnalog = A0;	Vesels konstants skaitlis (piemēram A0=14)
int	int valInt = analogRead(pinAnalog);	Potenciometra vesels stāvoklis no 0 līdz 1023
double	double valU = (5. / 1023) * valInt;	Reāls skaitlis (spriegums U volts)
String	String separator = " spriegums:";	Strings jeb burtu virknīte

1.5 Uzdevumi

Kas ir frekvence:

Frekvence raksturo, cik bieži notiek kāda darbība. Frekvenci mēra hercos. Piemēram, 50 Hz (50 hercu) frekvence nozīmē, ka kaut kas atkārtojas 50 reizes sekundē.

Piemērs:

Atradīsim frekvenci LED mirkšķināšanai piemērā `Blink.ino`. Tā kā pilns periods starp divām LED iemirgošanās reizēm ir $1 + 1 = 2$ sekundes, tad frekvence mirkšķināšanai ir $\frac{1}{2\text{s}} = 0.5$ Hz jeb 0.5 herci.

1.uzdevums:

Daži cilvēki ir jutīgi pret signāllampīņu mirkšķināšanu noteiktās frekvencēs (*Photosensitive Epilepsy*). Īpaši problemātiskas mēdz būt 15 līdz 20 hercu frekvences. Kā jāizvēlas abi gaidīšanas laiki izsaukumos `delay(millisecods)`, lai iegūtu šādu nekomfortablu diodes mirkšķināšanas frekvenci? Izmainiet Arduino skriptu `Blink.ino` no 2.praktiskā darba, lai krāsainā diode radītu šo frekvenci.

2.uzdevums:

Izveidot kēdi, kur gaismas diode sāk mirkšķināt tikai tad, kad potenciometrs ir aizbīdīts pāri pusei. To var pārbaudīt divos veidos:

- Skaitlis, ko nolasa no analogās ieejas A0 (`int valInt = analogRead(pinAnalog)`) ir virs 512 (t.i. pārsniedz $1023/2$ jeb pusi no maksimālās vērtības).
- *Potenciālu starpība* starp GND un OTA kontaktiem ir vismaz 2.5 V.

Jums var noderēt “if” operators. Piemēram,

```
if (valInt >= 512) {
    // mirkšķina diodi
}
else {
    // izslēdz diodi
}
```

Pabeigt šo skriptu. Saslēgt kēdi tā, lai potenciometra kontakts **OTA** rakstītu uz Arduino **A0**. Pieslēgt arī LED diodi (virknē ar rezistoru!) starp GND un D13 kontaktiem. Faktiski – kombinēt praktiskos darbus #2 un #3.

3.uzdevums:

Vai ir iespējama tāda ķēde un Arduino skripts, kas ar potenciometru maina uz diodes padoto spriegumu no 0 V līdz 2 V?

Atbilde:

Potenciometrs var mainīt spriegumu atbilstoši regulatora stāvoklim un gaismas diode turpina spīdēt arī tad, ja sprieguma kritums tajā ir mazāks nekā 2 volti. Elektriskā ķēde, kas var pakāpeniski samazināt gaismas diodes spožumu, dota zīmējumā. Ievērojam, ka šī shēma izmanto tikai **GND** un **5V** kontaktus (Arduino plate ir parasts strāvas avots). Tāpēc LED spožumu var regulēt neatkarīgi no skripta.

Ievērojam, ka ķēdē bez potenciometra (mainīgas pretestības) ir arī 220 V fiksētā pretestība, lai pie jebkura potenciometra stāvokļa LED nepārsniegtu maksimālo strāvas stiprumu. (Potenciometra izejā spriegums var sasniegt 5 V.)

