

**O'zbekiston Respublikasi Oliy va o'rta ta'lim vazirligi**

**Namangan muhandislik – texnologiya instituti**

**“Texnologik jarayonlarni avtomatlashirish va boshqarish va informatsion texnologiyalar” kafedrası**



**“Mexatronika”**

**Fanidan amaliyot mashg'ulotlarni bajarish**

**bo'yicha**

**USLUBIY QO'LLANMA**

**5321700 – Texnologik jarayonlarni boshqarishning axborot kommunikatsiya tizimlari yo'nalishi talabalari uchun**

Namangan-2021

Ushbu uslubiy ko'rsatma "Mexatronika" faning O'zbekiston Respublikasi Oliy va o'rta maxsus ta'lim vazirligi 20\_\_ yil \_\_\_\_ dagi \_\_\_-sonli buyrug'i bilan tasdiqlangan namunaviy va Namangan muhandizlik-texnologiya instituti kengashi tomonidan tasdiqlangan ishchi o'quv dasturlari asosida tuzildi.

Ko'rsatma 5311000 – "Texnologik jarayonlar va ishlab chiqarishni avtomatlashtirish va boshqarish (to'qimachilik, yengil va paxta sanoati) yo'nalishi. Talabalari uchun mo'jalangan bo'lib, ko'rsatmada xar bir amaliyot ishi qisqacha nazariy ma'lumotlar, ishlarini bajarish tartibi va mavzuga oid savollar keltirilgan.

Uslubiy ko'rsatma – TJABIT kaferdasining "\_\_\_\_" \_\_\_\_\_ 201\_ yil №\_\_\_\_ sonli yig'ilishida muhokama qilindi va tasdiqlandi.

Tuzuvchilar:

Qodirov D.T. – NamMTI, TJABIT kafedrasida kata o'qtuvchi.

Ismanov M.A. – NamMTI, TJABIT kafedrasida o'qituvchisi.

Taqrizchilar:

Abdulazizov B.T. – Nam.DU, fizika kafedrasida mudiri, fizika – matematika fanlari bo'yicha falsafa doktori

Turaqulov A.A. – NamMTI, TJABIT kafedrasida dotsenti, fizika-matematika fanlari nomzodi

Uslubiy ko'rsatma TJABIT kaferdasining "\_\_\_\_" \_\_\_\_\_ 201\_ yil №\_\_\_\_ sonli yig'ilishida muhokamadan o'tgan va fakultet kengashida muhakama qilish uchun tavsiya etilgan.

Uslubiy ko'rsatma "avtomatika va energetika" fakulteti o'quv-uslubiy kengashining 20\_\_ yil "\_\_\_" \_\_\_\_\_ №\_\_\_\_-sonli yig'ilishida muhokamadan o'tgan va institute o'quv-uslubiy kengashida muhakama qilish uchun tavsiya etilgan.

NamMTI o'quv – uslubiy kengashida ishlatishga tavsiya etiladi.

<<\_\_\_\_>> \_\_\_\_\_20\_\_ yil Bayonnoma № \_\_\_\_\_

## Mundarija

<b>Amaliy mashg`ulot.</b> Mexatronika elektr yuritmalarning kinematik sxemani loyihalash va hisoblash.....	
<b>Amaliy mashg`ulot.</b> Mexatronika gidravlik yuritmalarini hisoblash.....	
<b>Amaliy mashg`ulot.</b> “Arduino Uno” qurilmasini ishga tushirish.....	
<b>Amaliy mashg`ulot.</b> “Fritzing” dasturida elektron sxemalarni qurish.....	
<b>Amaliy mashg`ulot.</b> “Virtulbreadboard” dasturida elektron sxemalarni qurish.....	
<b>Amaliy mashg`ulot.</b> “Virtulbreadboard” dasturida “Arduino” kontrollerini qo‘llash.....	
<b>Amaliy mashg`ulot.</b> “Virtulbreadboard” dasturi yordamida “Arduino” kontrolleriga yorug‘lik diodlarini ulash.....	
<b>Amaliy mashg`ulot.</b> “Virtulbreadboard” dasturi yordamida LED ekranlarini ishga tushirish.....	
<b>Amaliy mashg`ulot.</b> “Virtulbreadboard” dasturi yordamida temperaturani boshqarish.....	

## **Amaliy ish**

### **Mexatronika elektr yuritmalarning kinematik sxemani loyihalash va hisoblash.**

*Mashg'ulot maqsadi:* Talabalarda mexatronika elektr yuritmalari uchun kerakli kinematik sxemani loyihalash amaliy ko'nimalarni shakllantirish.

Elektryuritmalarning kinematik sxemasini qurish mexatronika tizim loyihalashning mas'ul bosqichi hisoblanadi. Kinematik sxema mexatronika tizimning mexanika qismi konstruksiyasini, shuningdek uning ishlatish sifatini belgilaydi.

*Kinematik sxema* mexatronika tizim mexanika qismining hamma zveno va mexanizmlarini bog'lanish tartibini tekislik yoki fazoviy tasvirda ko'rsatadi va energiya oqimining taqsimlanishi haqida, mexanika qismi elementlarining kinematik aloqasi haqida, asosiy zvenolarni bir-biriga nisbatan joylashish holati haqida ma'lumot beradi. U tizimning kinematik va kuchini hisoblash uchun asosiy hujjat hisoblanadi.

Sxema masshtabga amal qilmasdan bajariladi. Sxemaning hamma elementlari shartli grafik belgilar bilan yoki tashqi ko'rinishi soddalashtirilgan holda belgilanadi.

Kinematik sxemada yurutma dvigatelning quvvati, dvigatel valining va mashina barcha vallarining aylanma tezligi, shkvlar diametrlari, tasma uzunligi va turi, tishli g'ildirak, yulduzcha va xrapovik tishlari soni, tishli uzatma moduli, zanjirli uzatma qadami, ishchi organlar yurishi soni va qiymati ko'rsatiladi. Vallar rim raqamlari bilan nomerlanadi. Mexanika qismning hamma elementlari sxemada arab raqamida tartib bilan nomerlanadi, chapdan o'nga yoki o'ngdan chapga va chizmaning bo'sh joyida matn bilan izoh beriladi, masalan jadvallar yordamida (ГОСТ 2.702—69).

#### **Umumiy uzatish sonini aniqlash va uni bosqichlarga taqsimlash**

Uzatmalarning asosiy kinematik xarakteristikasi bo'lib uzatishlar soni hisoblanadi:

$$u_{12} = \frac{\omega_1}{\omega_2} = \frac{n_1}{n_2} = \frac{M_{a2}}{M_{a1}\eta} \quad (5.1)$$

bu yerda  $\omega_1$  va  $\omega_2$  - yetaklovchi va yetaklanovchi vallarning burchak tezligi,  $s^{-1}$ ;  $n_1$  va  $n_2$  - yetaklovchi va yetaklanovchi vallarning aylanish chastotasi, ayl/min;  $M_{a1}$  va  $M_{a2}$  - yetaklovchi va yetaklanovchi vallardagi aylantirovchi momentlar, N·m;  $\eta$  - yurutma FIK.

Umumiy uzatish soni

$$u_{um} = n_{dv} \cdot n_{io} \quad (5.2)$$

bu yerda  $n_{dv}$  - elektrodvigatelning aylanishlar chastotasi, ayl/min ( $s^{-1}$ );  $n_{io}$  - ishchi organning aylanishlar chastotasi, ayl/min ( $s^{-1}$ ).

Keyin  $u_{um}$  ni alohida bosqichlarga bo'ladilar. Buning ko'p yechimi mavjud, lekin biz optimal variantini tanlab olishimiz kerak.

Umumiy uzatishlar sonini alohida bosqichlarga bo'lishda uzatishning umumiy soni har bir boshqich uzatish soni ko'paytmasiga teng bo'ladi:

$$u_{um} = u_1 u_2 u_3 \times \dots \times u_n \quad (5.3)$$

Agar umumiy uzatishlar soni aniq bo'lsa, boshqa  $n-1$  boshqichlar uzatishlar soni beriladi va oxirgi bosqich uzatishlar soni kattaliklari aniqlanadi:

$$u_n = u_{um} / (u_1 u_2 u_3 \times \dots \times u_{n-1}) \quad (5.4)$$

Umumiy uzatishlar sonini aniqlash umumiy me'zoni yo'q. Ammo quyidagilar tavsiya etiladi:

- qo'llanmalarda ko'rsatgan uzatishlar sonlarining o'rtacha qiymatlari tanlab olinadi;
- birinchi navbatda reduktor uchun, keyin zanjirli, tasmali va tishli uzatmalar uchun uzatishlar soni aniqlab olish kerak;
- silindrik va silindrik-konus reduktorlarda birinchi tezyurar bosqichlari uchun uzatishlar soni kattaroq tanlanadi (3 dan 5 gacha), keyingi sekin ishlaydigan bosqich uchun avvalgisidan 30-40 % kamroq olinadi;
- chervyakli reduktorning FIK ko'p bo'lishi uchun ko'p zaxodli chervyak qo'llaniladi.

Umumiy uzatishlar sonini alohida boshqichlarga bo'lish uchun 5.1-jadvaldagi ma'lumotlardan foydalanish mumkin:

**5.1-jadval. Har xil uzatmalarning uzatishlar soni qiymatlari**

Uzatmalar	Uzatishlar soni qiymati	
	o'rtachasi	Yuqorisi
Reduktorgagi tishli uzatma:		
silindrik g'ildirakli	3-6	10
konus g'ildirakli	2-3	6
Ochiq tishli	3-7	15
Chervyakli:		
yopiq	10-40	80
ochiq	15-60	120
Zanjirli	2-5	6
Friksion	2-4	5
Tasmali:		
yassi tasmali ochiq	2-4	6
yassi tasmali tortish rolikli	4-6	8
ponasimon tasmali	2-4	6

Keyingi hisoblashda tishli yoki yulduzchali uzatma tishlari soni, shkivlar diametri va h.k. aniqlanadi, bunda umumiy kinematik hisoblashda aniqlangan va qabul qilingan uzatishlar soni orasida tafovut bo'lishi mumkin. Umumiy uzatishlar sonini o'zgartirmaslik uchun bu tafovut yurutmaga kiradigan ba'zi uzatmalar sonini o'zgartirish bilan yoqotiladi. Odatda hisob-kitoblarni o'zgartirish uchun ochiq tishli va zanjirli uzatmalar uzatishlar soni o'zgartiriladi.

Uzatishlar sonini o'zgartirishda quyidagilar e'tiborga olinadi: tishli, chervyakli yoki zanjirli uzatmalar soni butun sonlar (tishlar sonlari) nisbatiga teng bo'ladi; tasmali





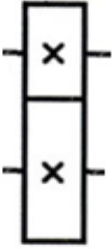

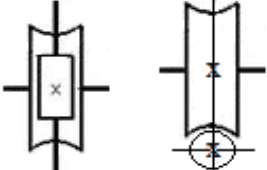

uzatma uchun – sirpanishni hisobga olgan holda shikivlar diametri standart qatoridagi qiymatlar nisbatiga teng. Shuning uchun yurutmaning haqiqiy uzatishlar soni hamma vaqt hisoblangan uzatishlar soniga teng bo'lmaydi. Chiqayotgan valning haqiqiy aylanish chastotasi deyarli hamma vaqt hisoblangan aylanish chastotasidan oz-moz farq qiladi. Bu farqni quyidagi formula bilan aniqlash mumkin:


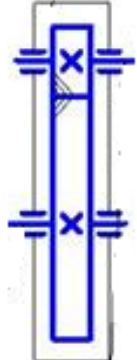
$$\Delta n = ((n - n_h) / n) \cdot 100 \% \quad (5.5)$$

bu yerda  $n$ - berilgan aylanish chastotasi, ayl/min ( $s^{-1}$ );  $n_h$  - haqiqiy aylanish chastotasi, ayl/min ( $s^{-1}$ ).

Faqat etaklanuvchi valning aylanish chastotasidagi xatolik  $\pm 1\%$  bo'lsa, hisoblash qoniqarli hisoblanadi.

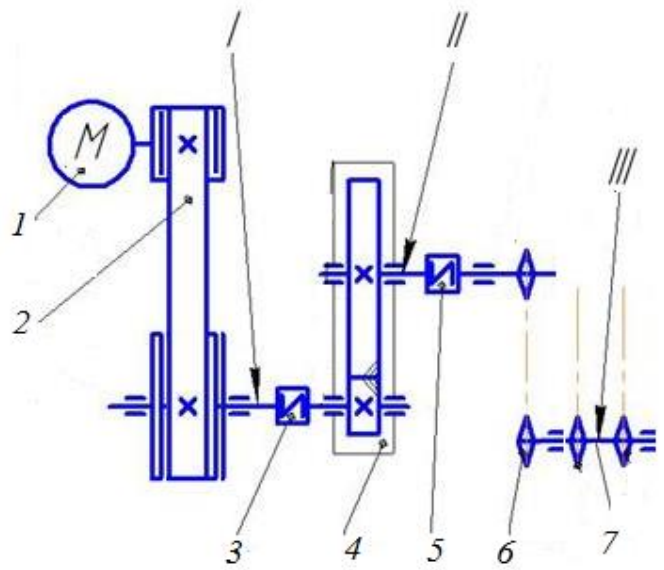
### 5.2-jadval. Kinematik sxemadagi elementlarning shartli belgisi

	Elektrdvigatel
	Mufta
	Val
	Podshipnik, tayanch
	Silindrik tishli uzatma
	Konussimon tishli uzatma
	Chervyakli uzatma
	Zanjirli uzatma

	Tasmali uzatma
	Bir boshqichli silindrik qiyshiq tishli reduktor (yopiq tishli uzatma)

***Ishni bajarish uchun namuna:***

*Mexatronik modulning quyidagi kinematik sxemasi berilgan:*



Modul tarkibiga quyidagilar kiradi: valining aylanishlar soni  $n_{dv}=1400$  ayl/min bo`lgan elektrodvigatel 1, tasmali uzatma 2, bir boshqichli silindrik qiyshiq tishli reduktor 4, zanjirli uzatma 6 va valining aylanishlar soni  $n_{r.o.}=12$  ayl/min bo`lgan ishchi organ.

I va II vallardagi aylanishlar sonini aniqlash kerak bo`ladi.

1. Kinematik sxemadagi umumiy uzatishlar sonini (5.2) formula bilan aniqlaymiz:

$$u_{um} = \frac{n_{dv}}{n_{i.o.}} = \frac{1400}{12} = 116,67$$

2. (5.3) formulaga asosan umumiy uzatishlar sonini alohida bosqichlarga bo'lishda uzatishning umumiy soni har bir boshqich uzatish soni ko'paytmasiga teng bo'ladi. Bizning holatda 3 ta uzatishlar mavjud (tasmali uzatma, bir bosqichli reduktor va zanjirli uzatma), shuning uchun:

$$u_{um} = u_1 \cdot u_2 \cdot u_3$$

bu yerda  $u_1$ ;  $u_2$  va  $u_3$ - tasmali uzatma, reduktor va zanjirli uzatma uzatishlari soni.

5.1-jadvalga asosan reduktor va zanjirli uzatma uchun uzatishlarni sonini belgilaymiz, tasmali uzatmani (5.4) formuladan hisoblaymiz.

$$u_2 = 6; u_3 = 5,$$

$$\text{Unda } u_1 = u_{um} / (u_2 \cdot u_3) = 116,67 / (6 \cdot 5) = 3,9$$

Kinematik sxemaning haqiqiy uzatishlar soni:

$$u'_{um} = u_1 \cdot u_2 \cdot u_3 = 6 \cdot 5 \cdot 3,9 = 117$$

Ishchi organning haqiqiy aylanishlar soni:

$$n_h = n_{dv} / u'_{um} = 1400 / 117 = 11,966 \text{ ayl/min}$$

Ishchi organ valining berilgan va hisoblangan aylanishlar soni orasidagi xatolikni (5.5) formula yordamida aniqlaymiz:

$$\Delta n = ((n - n_h) / n) \cdot 100 \% = ((12 - 11,966) / 12) \cdot 100 \% = 0,28\%$$

### Ishni bajarish tartibi

1. Talabalar o'qituvchi tomonidan berilgan ma'lumotlar asosida 1 ta mexatronik yuritma uchun kinematik sxema qurib, uzatishlar soni hisoblanadi.

2. Talaba bajargan ishlari bo'yicha o'qituvchiga yozma hisobot tayyorlab topshiradi.

### Takrorlash uchun savollar:

1. Kinematik sxema qaysi maqsadda quriladi?
2. Kinematik sxemada qanaqa parametrlar ko'rsatiladi?
3. Uzatishlar soni qanday hisoblanadi?
4. Umumiy uzatishlar sonini aniqlashda qaysi tavsiyalarga e'tibor berish kerak?

### Amaliy ish

#### Mexatronika gidravlik yuritmalarini hisoblash.

*Mashg'ulot maqsadi:* Talabalarda mexatronik elektr yuritmalari uchun kerakli quvvatni aniqlash amaliy ko'nimalarni shakllantirish.

### Nazariy ma'lumotlar

Energetik hisoblashda mexatronik elektr yuritmasi uchun kerakli quvvat hisoblanib, dvigatelning quvvati aniqlanadi va konkret dvigatel tanlanadi.

Yuritma uchun kerakli quvvat quyidagi formuladan aniqlanadi:



$$N = N_{sol} \Pi, \quad (6.1)$$

bu yerda  $N_{sol}$ — solishtirma quvvat, kVt·soat/kg;  $\Pi$  — mashina ish unumdorligi, kg/soat.

Solishtirma quvvat taqribiy kattalik va hisoblashda amaliyotdan olingan qiymati qo'llaniladi.

Quvvatni aniqlash uchun navbatdagi formulalar tavsiya etiladi:  
ilgarilama harakatda

$$N = P v \cdot 10^{-3} \quad (6.2)$$

aylanma harakatda

$$N = M \omega \cdot 10^{-3} \quad (6.3)$$

bu yerda  $P$ -foydali qarshiliklarning ta'sir qiluvchi kuchi (tortish kuchi), N;  $v$  — bajaruvchi organ tezligi, m/s;  $M$  – yetaklanuvchi valdagi aylanish momenti, N·m;  $\omega$ - yetaklanuvchi valning burchak tezligi, rad/s.

Transportlovchi qurilmalar yuritmasi uchun kerak bo'lgan quvvat material qaysi yo'nalishda: vertikal, gorizontal yoki qiyalik bo'yicha harakatlanishiga bo'g'liq. Umumiy ko'rinishda:

$$N = \Pi L w g/1000 \pm \Pi H g/1000 \quad (6.4)$$

bu yerda:  $\Pi$  — transportlovchi qurilmaning ish unumdorligi, kg/s;  $L$  — tashish uzunligi, m;  $N$  — ko'tarish balandligi, m;  $w$  — harakatga qarshilik koeffitsiyenti.

Bu formulani qiya burchak ostida yuk tashish uchun ham ishlatiladi. Agar vertikal tashish kerak bo'lsa, unda formuladagi  $L = N$  bo'ladi. Agar gorizontal tashish kerak bo'lsa, unda formuladagi  $H = 0$  bo'ladi

Yuritmaning quvvatini quyidagi formula bilan ham aniqlab bo'ladi.

$$N = F_0 v \cdot 10^{-3} \quad (6.5)$$

bu yerda:  $F_0$  — tortish kuchi, N;  $v$  — tortish elementining tezligi, m/s.

Elektrodvigatelning nominal quvvatini aniqlash formulasi:

$$N_{dv} = N / \eta \quad (6.6)$$

bu yerda:  $N$  - mashina yurutmasi uchun kerak bo'lgan quvvat, kVt;  $\eta$ - yurutmaning umumiy FIK.

Yuritmaning umumiy foydali ish koeffitsenti uzatmalarning xususiy FIKlari ko'paytmasiga teng:

$$\eta = \eta_1 \eta_2 \eta_3 \times \dots \times \eta_n \quad (6.7)$$

Bu formula faqat quvvatni ketma-ket uzatishda to'g'ri bo'ladi. Yurutmaning istalgan vali tayanadigan podshipniklar parallel ishlaydi, va bitta val podshipniklaridagi yo'qotilgan kuchni hisobga oladigan FIK (podshipniklar soni qancha bo'lishidan qat'iy nazar), bitta kattalik bilan ko'rsatiladi.

Tayanchlardagi ishqalanishda sarflanadigan kuchlar koeffisientlari quyidagicha: bir juft dumalash podshipniklari uchun  $\eta = 0,99-0,995$ ; bir juft sirpanish podshipniklari uchun moylash sharoitlariga qarab  $\eta=0,98-0,99$ .

Har xil uzatmalarning o'rtacha FIK qiymatlari quyidagi jadvalda keltirilgan (podshipniklardagi yo'qotishlar hisobga olinmagan):

Elektrodvigatelning nominal qiymatini aniqlangan keyin, berilgan ish sharoiti uchun qaysi tipdagi dvigatel kerakligi aniqlab olinadi va kataloglardan hisob-kitoblarimizga yaqin dvigatel tip-o'lchami tanlanadi. Bunda, ko'pincha, hisoblangan quvvatdan yuqori quvvatli dvigatel tanlanadi.

### Har xil uzatmalarning o'rtacha FIK qiymatlari

Uzatma	Moy vannasida	Ochiq
Tishli	0,96-0,98	0,94
Chervyakli, kirish sonlari uchun:		
z=1	0,7	0,5
z=2	0,75	0,6
z=3	0,85	--
Zanjirli	0,95-0,97	0,9
Friktsion	0,90-0,95	0,7-0,88
Tasmali	--	0,95-0,96

Kataloglarda quvvati bir xil, ammo burchak tezligi har xil bo'lgan bir necha tipdagi elektrodvigatellar boladi. Ular ichidan burchak tezligi kinematik sxemaga to'g'ri keladigan elektrodvigatelni tanlashimiz kerak. Bunda quyidagini hisobga olish kerak: elektrodvigatel vali qancha tez ayladigan bo'lsa uning o'lchami, massasi va narxi kichikroq; dvigatel vali sekin aylanadigan bo'lsa, shuncha umumiy uzatishlar soni kichik bo'ladi.

### Ishni bajarish tartibi

1. Talabalar o'qituvchi tomonidan berilgan ma'lumotlar asosida 1 ta mexatronik yuritma uchun kerakli quvvatni hisoblab, elektrodvigatelni tanlab oladi.

2. Talaba bajargan ishlari bo'yicha o'qituvchiga yozma hisobot tayyorlab topshiradi.

### Takrorlash uchun savollar:

1. Mexatronik yuritmalarni energetik hisoblashda qanaqa ishlar bajariladi?
2. Ilgarilanma va aylanma harakatda yuritma quvvati qanday aniqlanadi?
3. Elektrodvigatelning nominal quvvati qanday hisoblanadi?
4. Yuritmaning umumiy F.I.K. hisoblash formulasi?

## amaliy mashg'ulot.

### “Arduino Uno” qurilmasini ishga tushirish.

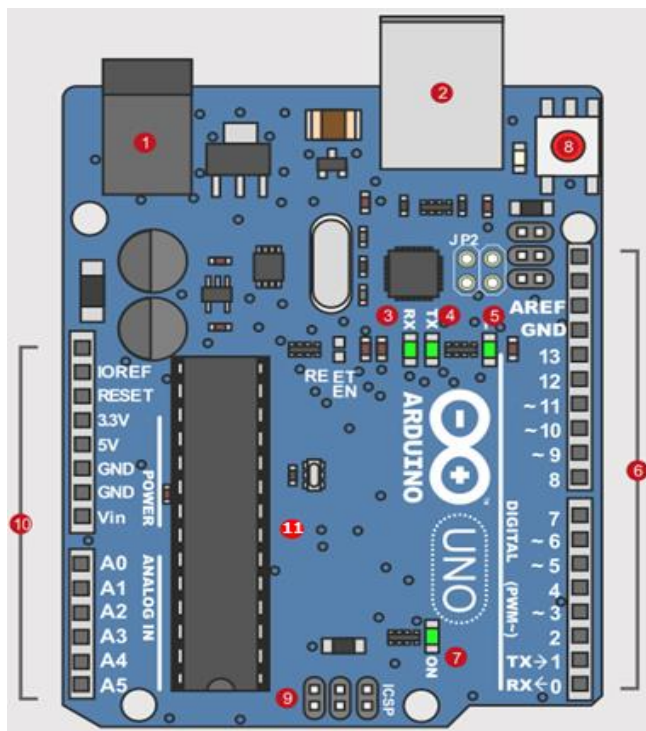
#### Nazariy ma'lumotlar: Arduino nima?

Yarim avtomatik tizimlar, web-server, o'z-o'zini boshqaruvchi mustaqil tizimlar (avtopilot) ni yaratish uchun, Arduino IDE dasturlash tili yordamida mikrokontrollerni boshqaruv tizimini loyihalash uchun vositadir. Datchiklar yordamida analog va raqamli signallarni qayta ishlash imkoniyatiga ega bo'lamiz. Qisqacha qilib aytganda barcha avtomatik boshqaruv tizimlarini “miya” sini yaratamiz.

#### Arduino bilan tanishish.

Barcha Arduino platformalari quyidagi bo'laklardan tarkib topgan. 1- rasm

1. Elektr manbasini ulash uchun port. 9 – 12 volt kuchlanish oralig'i uchun
2. Komyuter bilan bog'lanish uchun va Dasturlash va ma'lumotlarni uzatish uchun USB-iterfeys.
3. Rx svetadiodi.
4. Tx svetadiodi.
5. Indikator. Platforma ishlayotganda u aktiv holatda bo'ladi.
6. Aref, ground (yer yoki faza nol), digital (raqamli), Rx, Tx portlari
7. Manba indikator. Platformaga manba ulangan bo'lsa u aktiv holatda bo'ladi.
8. Qayta yuklash uchun tugmacha (reset). Yuklangan dastur uchun ham amal qiladi.
9. Dasturlash uchun port ICSP (In Circuit serial programming). Programmator yordamida dasturlash uchun.
10. Analog In (analogli), Power In (kiruvchi signal), power out (chiquvchi signal), Ground (yer yoki faza nol), reset (qayta yuklash) portlari.
11. Atmel mikrokontrolleri.



1-rasm. Arduino platformasi ko'rinishi

## **Texnik parametrlari**

Mikrokontroller-ATmega328

Ishchi kuchlanish-5 volt

Kiruvchi kuchlanish-7-12 volt

Raqamli kirish-chiqish portlari-14 dona

Analogli kirish portlari-6 dona

O'zgarimas tok kirish-chiqish uchun-40 mA

O'zgarimas tok analogli kirish uchun-50mA

Flesh-xotira-32 kilobayt bundan 0.5 kilobayt hajmi yuklovchi qismiga to'g'ri keladi.

Operativ xotira-2 kilobayt

EEPROM xotira-1 kilobayt

Taktik chastota-16 MegaGers

Bundan tashqari Arduino platformalarning bir necha xil turlari mavjud bo'lib, ular loyihani bajarish turiga va yo'nalishiga qarab tanlash mumkin. Arduino Uno platformasiga ATmega328 mikrokontrolleri o'rnatilgan bo'lib, u 14 ta raqamli portlardan iborat va xotira registrlari ham boshqalarga nisbatan hajm bo'yicha o'xshash emas. Arduino Leonarda platformasida esa 32U4 kontrolleri mavjud bo'lib, u bilan loyihaning kichikroq qismlarida foydalanishga mo'ljallangan. Masalan: joystick bilan ishlashda yoki bo'lmasa klaviatura ulanganda va boshqa turli masalalarda. Arduino Mega platformasiga esa ATmega 2560 mikrokontrolleri o'rnatilgan bo'lib 54 ta raqamli kiritish va chiqarish portlari mavjud. Taktoviy chastotasi 84 MegaGers SAM3X8E ARM Cortex-M3 Atmel protsettori negizida 32 razryadli Arduino DUE platformalari ham mavjudki, ularda yuqori aniqlikdagi raqamli-analogli konvertrlar mavjuddir.

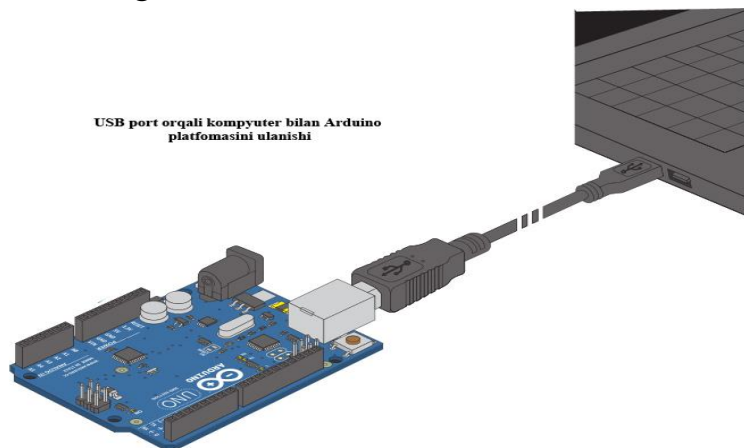
## **Elektr manba**

Platforma aktiv holatda bo'lishi uchun elektr manba bilan ta'minlangan bo'lishi zarur. Mikrokontrollerga dastur yozish vaqtida platforma USB port orqali kompyuterga ulangan bo'lsa elektr manba sifatida kompyuterni olishimiz mumkin. Ushbu port orqali 5v kuchlanishli elektr tokini olishimiz mumkin. 5v kuchlanish Arduino platformasi va uning elementlari uchun yetarlidir. Platforma mustaqil ishlayotgan paytda yoki albatta elektr manbasiga ulanadi (DC port yoki  $V_{in}$ ). 1-rasm.1. Ushbu port orqali 9 – 12 v (yoki 6-20 v) kuchlanishli elektr tokini ulashimiz mumkin. Platformada stabilizator mavjud bo'lib, u 0-5 v hamda 3.3 v kuchlanishlarni ajratib beradi. Platformadagi mantiqiy amallar 0-5 v kuchlanish ostida, qo'shimcha ulangan periferik qurilmalar esa 3.3v kuchlanish ostida ishlaydi.

## **Dasturlash interfeysi**

Oddiy dasturlar Arduino platformasiga C (si) yoki Assambler tilida programmator yordamida ICSP porti orqali yoziladi. 1-rasm.9. Dastur Arduino platformasi xotirasiga dastur kodining yuklovchi qismi orqali SPI periferik interfeysi yordamida yuklanadi.

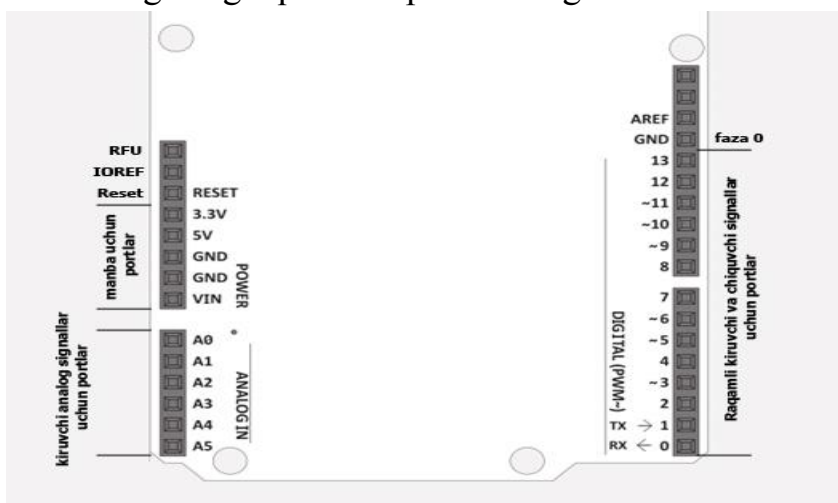
Arduino platformasi yoqilganda birinchi yuklovchi aktivlashadi hamda u dasturni ishga tushirib beradi. USB port orqali ham dastur yozish mumkin va bu holat foydalanuvchiga qulayliklar tug'diradi. 1-rasm.2. USB port Arduino platformasining zamonaviy versiyalarida platformaning ichida joylashtirilgan holatda bo'ladi. Oldingi versiyalarida esa USB port uchun qo'shimcha maxsus mikrosxemalar kerak bo'lar edi. Axborot texnologiyalari rivojlangani sari ishlab chiqaruvchilar ham bozor sharoiti va talabga mos mahsulotlar yaratmoqdalar. Mikrokontrollerga dastur yozish hozirgi vaqtda juda oson va sodda ko'rinishga keldi. 2-rasm.



2-rasm.

### Analog va raqamli kiritish-chiqarish portlari

Avtomatlashtirilgan tizimlarni loyihalash uchun albatta analog yoki raqamli signallar bilan ishlanadi. Ushbu signallar orqali ma'lumotlar qayta ishlanadi. Arduino platformasida analog va raqamli signallar bilan ishlash uchun maxsus portlar mavjud. Qo'shimcha periferik qurilmalarni ushbu portlar orqali platformaga ulaymiz. Portlar analog signal qabul qiluvchi, chiqaruvchi, raqamli signal qabul qiluvchi yoki chiqaruvchi va funksional kabi turlar ajratiladi. 14 ta raqamli portlar dasturlash yordamida kiritish va chiqarish parametrlarini o'zgartirishimiz mumkin. 3-rasm. Arduino platformalarining turiga qarab bu portlar o'zgarishi mumkin.



3-rasm.

## **Mikrokontrollerlarni dasturlash.**

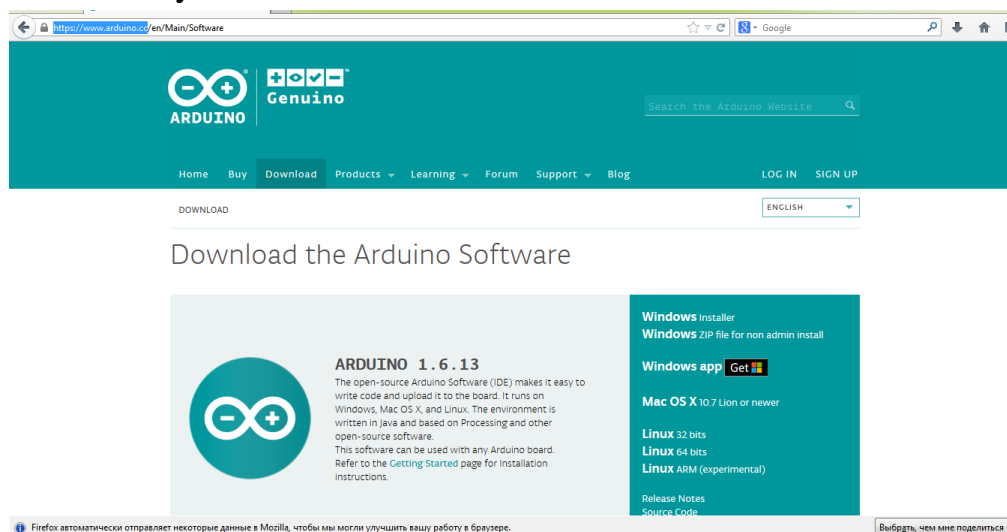
Mikrokontrollerlarga dastur yozishda turli hil dasturlash tillaridan foydalaniladi.

1. Mashina tili
2. Assembler
3. Yuqori darajadagi tillar

Mashina tili yordamida dastur yozish juda qiyin va dasturchidan maxsus bilim shuningdek vaqt kerak bo'ladi. Dasturlar to'g'ridan to'g'ri ikkilik tilida yoziladi. Lekin shuni qayd etish kerakki mashina tilida yozilgan dastur boshqa tillarga qaraganda programmaning vaqt bo'yicha bajarilishini hamda xotiraga uni talab bo'yicha joylashtirishni optimallashtirib beradi. Assamblerdan foydalanilganda dasturchi bilan mikrokontrollerni Assambler ajratib turadi. Assambler aslida assembler tilidan mashina tiliga dasturni o'tkazish translyatsiya qiluvchi vosita hisoblanadi. Mashina tiliga qaraganda osonligi bilan mashina tilidan farq qiladi. Yuqori darajali tillarga ALGOL, FORTRAN, KOBOL, BEYSIK, KARAT, DELPHI, C, C++ kabi dasturlash tillari kiradi. Dastur yozayotganda mashina tiliga ko'proq imkoniyatga ega, chunki mashina tilida xotira registrlarini yacheykalari juda samarali ishlatiladi. Mashina tilining kamchiligi shundaki murakkab dasturlar tuzish qiyin. Chunki dastur bevosita ikkilik kodida yoziladi va kompyuterga kiritiladi. Ikkilik kodida bajariladigan ammallarni eslab qolish esa mumkin emas. Shuning uchun yuqori darajadagi tillardan keng foydalanib kelinmoqda. Bularda dastur yozish tez bajariladi, lekin kamchiligi qo'shimcha xotira kerak ekanligida.

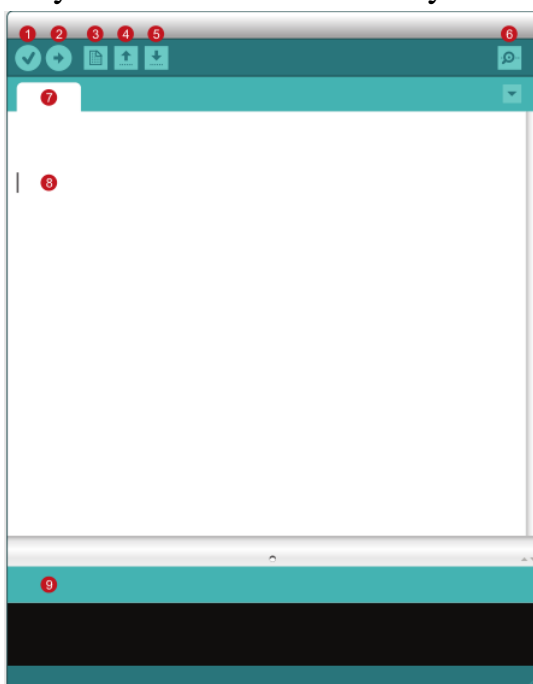
## **Arduino IDE muhitini kompyuterga o'rnatish.**

Arduino IDE dasturlash muhiti yordamida mikrokontrollerga dastur yozish mumkin. Ushbu muhit soddaligi bilan bashqa muhitlardan farq qiladi. Ushbu dasturni kompyuterga o'rnatish uchun <https://www.arduino.cc> internet sahifadan operatsion tizimga mos dasturni yuklab olish zarur. 4-rasm.



4-rasm

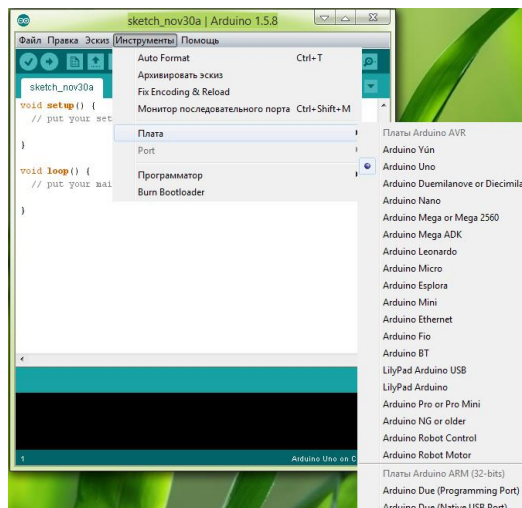
Dasturni ko'chirib olgandan so'ng uni kompyuterga o'rnatamiz va bizda quyidagicha dasturlash tili muhiti oynasi paydo bo'ladi. (Windows operatsion tizim uchun yuklandi). 5-rasm. Drayverlar avtomatik tarzda yuklanadi.



5-rasm. Arduino IDE dasturlash muhiti oynasi

- |   |              |   |
|---|--------------|---|
| 1 | verify       | Dastur yozilgandan so'ng tekshirish kamchiliklarni bartaraf etish uchun va dasturni test qilish uchun ishlatiladi |
| 2 | Upload       | Arduino mikrokontrolleriga dasturni yozish. Albatta platforma kompyuterga ulangan holatda bo'lishi zarur          |
| 3 | New          | Yangi dastur yaratish   |
| 4 | Open         | Xotiradan tayyor dasturni chaqirib olish  |
| 5 | save         | Loyihani saqlash  |
| 6 | Serial motor | Kiruvchi yoki chiquvchi signallarni ekranda ko'rsatish oynasi   |
| 7 | Sketch name  | Dastur nomi   |
| 8 | Code Area    | Dastur yozish oynasi  |
| 9 | Message Area | Xabarlar oynasi   |

Dastur ishga tushgandan so'ng ayrim parametrlarni o'zgartirish kerak bo'ladi. Qanday platforma bilan ishlayotgan bo'lsangiz shunga mos ravishda quyidagilardan birni tanlash kerak bo'ladi. 6-rasm. Eng so'nggi dasturlash muhitida bunday vazifani bajarish shart emas, chunki dasturning o'zi qanday platforma kompyuterga ulanganini mustaqil ravishda aniqlaydi.

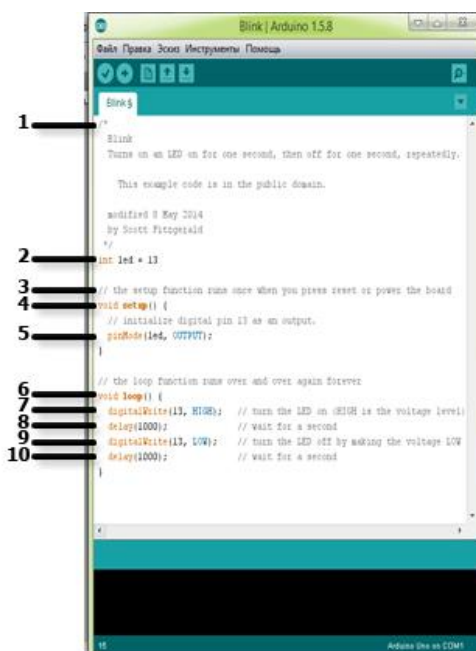


6-rasm.

### Amaliy qisim:

### Qisqa bir dastur tahlili

Ushbu muhitda dastur tuzaylik. Dastur matnini tahlil qilish orqali ayrim savollarga javob izlaymiz. 7-rasm.



7-rasm.

- 1- Ko'p qatorli izoh. Dastur matnini yoddan chiqarmaslik va tushunarli bo'lishi uchun izohlardan foydalanamiz. Ko'p qatorli izoh boshlanishida /\* belgisi bilan boshlanadi va tugashida \*/ belgilari bilan tugaydi. Kata dasturlar yozilganda izohlardan foydalanish tavsiya etiladi.
- 2- O'zgaruvchini e'lon qilish. O'zgaruvchi-bu xotira yacheykasi va u o'zida ma'lumot saqlaydi. O'zgaruvchilarni turlari ko'p. yuqoridagi misolimizda int (butun) tipli o'zgaruvchi e'lon qilinmoqda. Butun tipli led o'zgaruvchi 13 soniga o'zlashtirilmoqda. Bu yerda 13 soni Arduino platformasi 13 sonli raqamli porti nomi. Dasturning istalgan qismida led o'zgaruvchisi orqali 13 portga murojaat etishimiz yoki ushbu portni boshqarishimiz mumkin. Ushbu tip xotiradan 2 bayt joy egallaydi. O'zgarish oralig'i (-32 768 dan 32 767 gacha).



Umumiy formulasi:

Int o'zgaruvchi nomi = o'zgarishi tayinlangan qiymat;

Int x

X = -32,768;

X = x-1; // x ning qiymati 32,767 ga teng bo'ladi.

X = 32,767;

X = x+1; // x ning qiymati -32,768 ga teng bo'ladi.

Bundan tashqari o'zgaruvchilarning float(haqiqiy), bool ( mantiqiy), char (belgili) tiplari ham mavjud.

Float tipi verguli suriladigan ya'ni haqiqiy sonlar to'plamini o'z ichiga oladi. - 3.4028235E+38 do 3.4028235E+38 diapazonni o'z ichiga oladi. Xotiradan bayt joy egallaydi.

Float tipi double tipi bilan bir xil aniqlikka ega va ular verguldan keyin 6-7 belgi aniqlikkacha ishlaydi.

Misol:

Float maktab;

Float komp = 1.117;

Umumiy formulasi:

Float o'zgaruvchi nomi = o'zgarishi tayinlangan qiymat;

Int x;

Int y;

Float z;

X = 1;

Y = x/2; //y ning qiymati 0 ga teng.

Z = (float)x/2.0 //z ning qiymati 0.5 ga teng

Char belgili tipi xotiradan 1 bayt joy egallaydi. Uni e'lon qilishda qo'shtirnoq ichiga olib beriladi, masalan: 'A'. belgi xotirada sonni saqlaydi, ASCII jadvaliga asosan.

Masalan:

Char maktab = 'A';

Char maktab = 65; /\* ekvivalent o'zgaruvchilar, chunki A belgisi \*/ASCII=65 ga teng

Boolean mantiqiy tipi esa xotiradan 1 bayt joy egallaydi va faqat true (rost) yoki false (yolg'on) qiymat qabul qiladi.

- 3- Qisqa satrli izoh. U // belgisi yozilgandan so'ng kiritiladi va kompilyator bu belgi o'qishi bilan keying satrga o'tib ketadi va bu belgi kompilyatsiya qilinmaydi.
- 4- Funksiya – setup(). Dasturni ishga tushirish uchun kompilyatorga xabar jo'natadi. Har bir dasturda ushbu funksiya mavjud bo'lishi kerak. funksiya dastur kodining

bir qismidir. O'zgaruvchilar hamda qo'shimcha kutubxonalar setup() funksiyasi ichida e'lon qilinadi.

Masalan:

```
Int buttonpin = 3;
Void setup(){
Serial.begin(9600);
pinMode(buttonPin, INPUT);
}
Void loop(){
//...
}
```

- 5- pinMode(led,OUTPUT) portdan chiqish rejimini o'rnatish. Uni ikki parametri mavjud. Birinchi parameter portni nomerini aniqlamoqda. Misolda uni led nomli o'zgaruvchi bilan 13 ga tenglagan edik. Signalni 13- portdan chiqish rejimi ishga tushirilmoqda. Ikkinchi parameter INPUT(kirish) yoki OUTPUT(chiqish) bo'lishi mumkin.
- 6- Loop() ushbu funksiya albatta dasturda bo'lishi zarur. Bu sikl operatori.
- 7- digitalWrite kirish-chiqish signallari holatini belgilab beradi. Uni ham ikki parametri mavjud bo'lin birinchisi portni raqamini belgilaydi, ikkinchisi esa signal holatini aniqlaydi. Ikkinchi parameter HIGH (5v) yoki LOW(0v) bo'lishi mumkin. Agar pinMode() uchun OUTPUT (chiqish) holati belgilangan bo'lsa, u holda digitalWrite() ni ikkinchi parametric HIGH yoki Low qiymat qabul qiladi. Agar pinMode() uchun Input holati belgilangan bo'lsa, u holda Arduino platformasi avtomatik ravishda 20 kiloOm qiymatga ega bo'lgan qarshilikni ishga tushuradi.
- 8- Delay() funksiyasi faqat bitta argument qabul qiladi. Delay(1000) 1 sekund mobaynida kutib turmoq ya'ni keying komandaga o'tish uchun bir sekund kutib turadi.
- 9- Svetadiod LOW holatiga o'tmoqda.
- 10- Bir sekund kutib turiladi va yana siklga o'tib jarayon takrorlanadi. Demak ushbu dastur ishga tushgandan so'ng svetadiod 5 volt kuchlanish oladi, bir sekund vaqtdan so'ng nol volt kuchlanish oladi va shu holatda sikl bajarilaveradi. Dasturlash uchun minimal tushunchalar berib o'tildi. Qolgan funksiya va operatorlar bilan keying darslarda aytib o'tamiz.

### **amaliy mashg'ulot.**

#### **“Fritzing” dasturida elektron sxemalarni qurish.**

##### **Nazariy qisim:**

Fritzing - bu Dizaynerlar, rassomlar va jismoniy hisoblash va prototiplashga qiziqish bildirgan har bir kishi uchun [elektron Dizayn Avtomatlashirish](#) dasturi.

Fritzingning maqsadi jismoniy hisoblash loyihalarini hujjatlashtirish va almashish, bosilgan elektron platalar (PCB) va elektronikani o'qitish uchun sxemalarni ishlab chiqarish uchun qulay vositalarni taqdim etishdir.

Agar siz mustahkam prototiplarni yaratmoqchi bo'lsangiz, o'z g'oyalaringizni seriyali ishlab chiqarishga va g'ayrat bilan rivojlanayotgan hamjamiyatga qo'shilishni istasangiz, bu boshlash uchun joy.

## Amaliy qism:

### Mundarija:

Fritzing foydalanuvchi interfeysi

Asosiy ish oqimi

PCB ishlab chiqarish

Fritzing hamjamiyati

### Fritzing foydalanuvchi interfeysi

Fritzing tez va oson ish jarayoni uchun foydalanuvchilarga qulay interfeysni taqdim etadi.

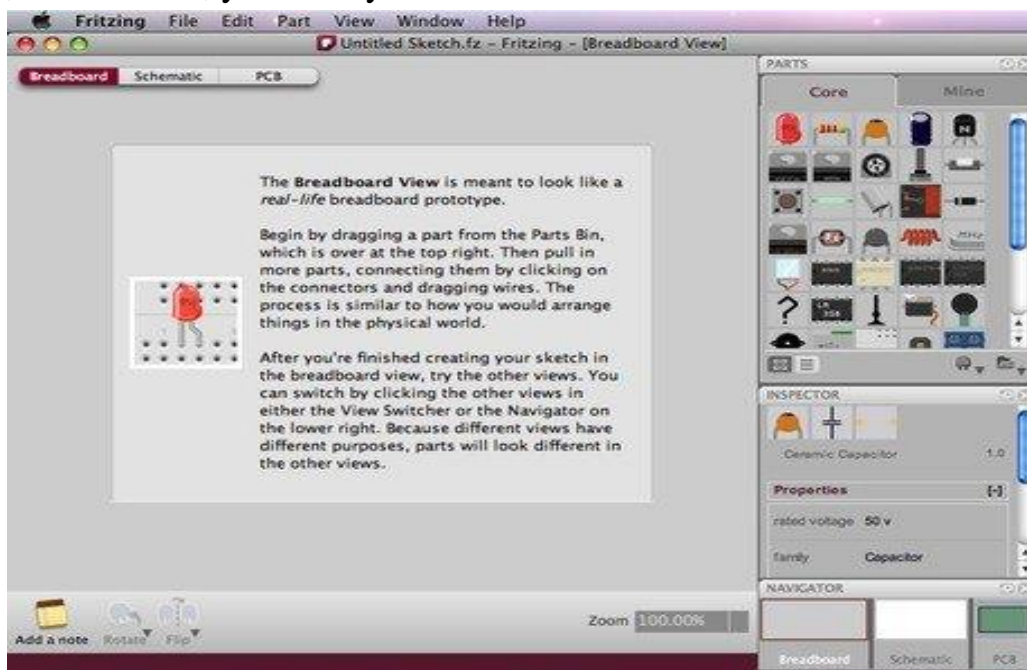
Uning atrof-muhit bo'limlari quyidagilarni o'z ichiga oladi:

Project View - bu virtual elektron sxemasi qurilgan va taxtada, sxematik yoki pcb ko'rinishida tahrir qilingan.

Palitra Windows - qism kutubxonasi, qism inspektori, tarixni bekor qilish va navigatorni o'z ichiga oladi.

Part Creator - bu Fritzing uchun qismlarni o'zgartirish yoki yangi qismlarni yaratish uchun vosita (Part Creator-ni ochish uchun, asosiy menyuda Part> New ni tanlang).

Fritzing Environment-ni foydalanuvchi ehtiyojlar va imtiyozlarga muvofiq qayta tashkil qilishi mumkin. Turli bo'limlar va Palitra Windows-ning o'lchamini o'zgartirish, ko'chirish, birlashtirish, yashirish yoki suzuvchi sifatida ko'rsatish mumkin.



## Asosiy ish oqimi

Oddiy Fritzing ish oqimi quyidagi bosqichlardan iborat:

**Haqiqiy kontaktlarning zanglashiga olib borish** - Fritzingda uni qaytadan qurishdan oldin, avvalo haqiqiy dunyoda zanjir hosil qilib, uni sinab ko'rish juda muhimdir.

**Fritzing-da kontaktlarning zanglashiga olib borish** - bu qism kutubxonasidan Project View-ga tortish va ulanish orqali amalga oshiriladi. Agar siz kutubxonada biron bir qismni topa olmasangiz, Fritzing sizga Part Creator yaratuvchisidan foydalanib o'z qisimingizni yaratishga imkon beradi. Shu bilan bir qatorda, siz yangi qism va uning ulagichlarini tezda aniqlashga imkon beradigan sirli qismdan foydalanishingiz mumkin (belgi qusish belgisiga o'xshaydi -?). Qismlar non paneli va simlar yordamida ulanadi. Simlar qism ulagichini bosish va tortish orqali darhol yaratiladi.

**Zanjirni tahrirlash** - tortish / tushirish, nusxalash va yopish, bir nechta tanlov, aylantirish, tarixni bekor qilish va boshqalar - bu Fritzing atrof-muhit bo'limlari va menyu panelida birlashtirilgan funktsiyalar. Bükme nuqtalarini yaratish orqali siz simni bükishingiz mumkin. Ularni simdan tortib oling. Sizning eskizingizga Eslatmalar (pastki satr menyusi) yoki Yorliqlarni (qismni o'ng tugmasini bosib) qo'shishingiz mumkin.

**Bir qismning xususiyatlarini o'zgartirish** - qismlarning xususiyatlarini Palitra Windows bo'limidagi qism inspektorida topish va o'zgartirish mumkin. Loyihani ko'rish qismida qism tanlanganida, tegishli qism to'g'risidagi ma'lumotlar Part inspektorida ko'rsatiladi.

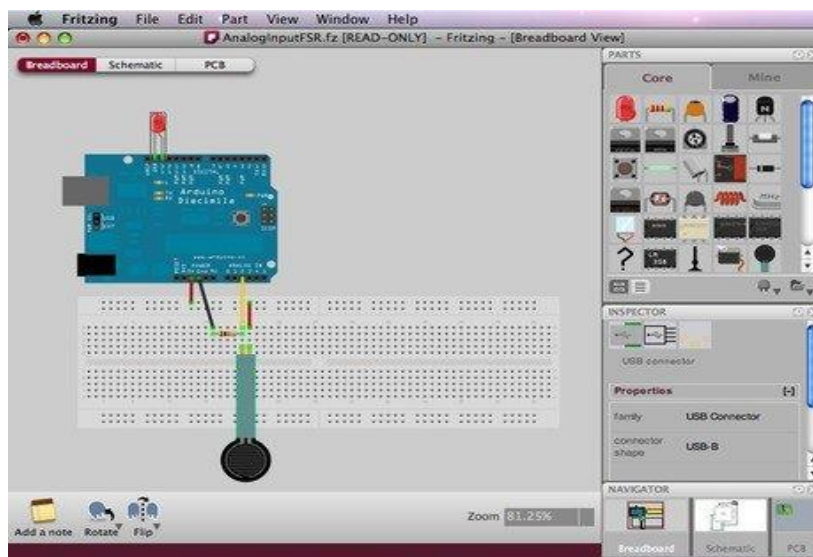
**Loyiha ko'rinishi o'rtasida almashish** - Navigator palitrasi yoki View almashtirgichidan foydalanib, siz panel, sxematik va pcb ko'rinishlarini almashtirishingiz mumkin. Ushbu qarashlarning har biri loyihaning asosiy ish muhiti sifatida ishlatilishi mumkin va istalgan vaqtda tanlanishi mumkin.

**PCB-ni loyihalash** - kontaktlarning zanglashiga olib borishi chizig'i tayyor bo'lganda, siz PCB ko'rinishida PCB tartibini yaratishingiz mumkin. Fritzing avtomatik marshrutlash va turli xil eksport formatlariga ega.

**Loyihani hujjatlashtirish** - loyihalashtirish va menyu satrida Eksport funktsiyasidan foydalanib hujjatlashtirish mumkin.

Bu erda [qanday qilib kontaktlarning zanglashiga olib borilishi](#) va [tenglikni joylashtirish tartibini loyihalash bo'yicha](#) bosqichma-bosqich qo'llanmalar mavjud .

Keyinchalik rivojlangan foydalanuvchilar [maxsus qismlarni qanday yaratish bo'yicha](#) qo'llanmani tekshirishlari kerak .



### Ishni bajarish tartibi

3. Talabalar o`qituvchi tomonidan berilgan ma`lumotlar asosida 1 ta mexatronik yuritma “fritzing” sxema qurib, uzatishlar soni hisoblanadi.

4. Talaba bajargan ishlari bo`yicha o`qituvchiga yozma hisobot tayyorlab topshiradi.

Nazariy savollar:

1. “Fritzing” dasturi nima ?

2. Dasturda asosan nima ish qilinadi?

### Amaliy mashg`ulot

#### “Virtulbreadboard” dasturida elektron sxemalarni qurish.

*Mashg`ulot maqsadi:* Talabalarda mexatronika moduli uchun oddiy elektron sxemalarni qurish **amaliy** ko`nimalarni shakllantirish.

Bu amaliy mashg`ulotda talabalar juda oddiy elektron sxemani yaratishni o`rganishadi. Bu sxema yorug`lik chiqaruvchi diod *LEDni (Light Emitting Diode)* yoqishni amalga oshiradi.

*Talabalar quyidagilar haqida ma`lumotga ega bo`lishadi:*

- rezistorlar haqida;
- svetodiodlar (yoruhlik diodlari);

*Talabalar quyidagilarni o`rganadi:*

- elektr sxemani o`qish tartibini;
- maketda elektr sxemani qurish tartibini.

### Qisqa nazariy ma`lumotlar

**Rezistor** (ingl. *resistor*, lotinchada *resisto* – qarshilik ko`rsataman) – elektrik zanjirning passiv elementi bo`lib, elektr qarshilikni aniq va o`zgaruvchan qiymatiga ega, tok kuchini chiziqli ravishda kuchlanishga aylantirishga va kuchlanishni tok kuchiga aylantirish, tokni cheklash, elektr energiyani yutish va hokazolar uchun mo`ljallangan.



- a) Evropa va Rossiyada qabul qilingan belgilanishi  
 b) AQSH qabul qilingan belgilanishi

Rezistorlarni ketma-ket ulaganda ularning qarshiliklari qiymati yig'indisi olinadi  
 $R = R_1 + R_2 + R_3 + \dots$

Rezistorlarni paralell ulaganda esa teskari qarshiliklar yig'iladi (ya'ni  $1/R$  umumiy qarshilik  $1/R_i$  har bir qarshilik o'tkazuvchanligi yig'indisidan iborat bo'ladi)  
 $\frac{1}{R} = \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3} + \dots$

**Svetodiód** (yorug'lik diodi) yoki **yorug'lik tarqatuvchi diod** (*angl. light-emitting diode, LED*) – elektron-tirqishli o'tishlarga ega yarim o'tkazgichli element bo'lib, unda to'g'ri yo'nalishda elektr tokini o'tkazilishi paytida optik nurlanish paydo bo'ladi.



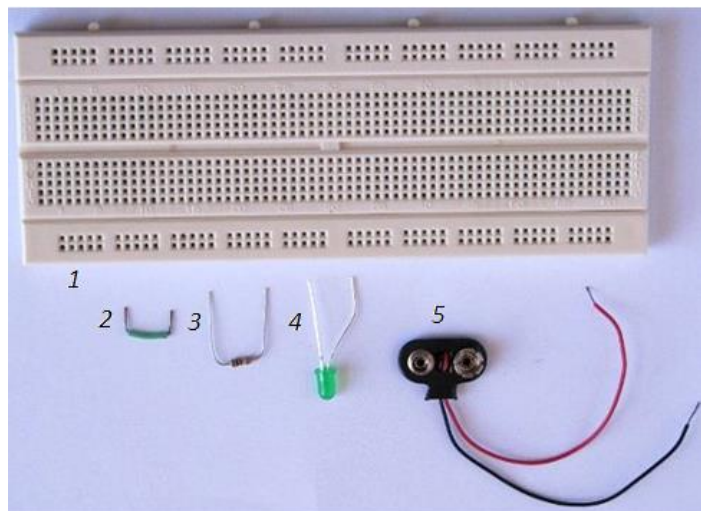
Element – Svetodiód

Ixtirochilar - Oleg Losev (1927),  
 Nik Xolon'yak (1962)  
 Belgilanishi



### Sxema komponentlari:

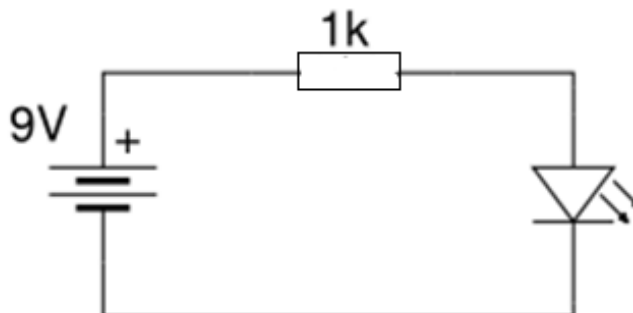
Sxemani tashkil etish uchun quyidagilar kerak bo'ladi (**1.1-rasm**):



1. Maket (Breadboard);
2. Maketda ulash uchun o`tkazgich simlari (Wire Link);
3.  $1\text{ k}\Omega$  qarshilikli rezistor;
4.  $5\text{ mm}$ li LED;
5. Batareya qisqichi;
6.  $9\text{V}$ li batareya yoki boshqa elektr energiya manbai.

### Printsiptial sxemani o`qish.

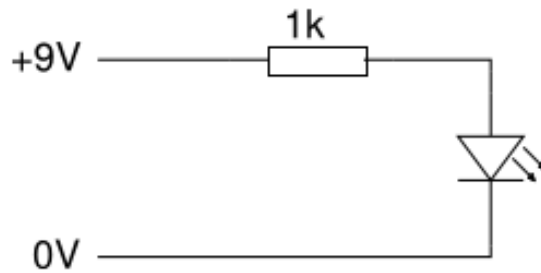
LEDni ulash printsiptial sxemasi (smexatik diagramma sifatida ham ma`lum) 1.2-rasmda ko`rsatilgan:



**1.2-rasm. LEDni ulash printsiptial sxemasi**

Bu sxema bo`yicha (batareyadan soat strelkasi bo`yicha) akkumlyatorli batareyaning musbat chiqishini (qora ulash o`tkazgichi)  $1\text{ k}\Omega$  rezistorga ulaymiz. Rezistorning boshqa uchini esa yorug`lik diodning anod uchiga ulaymiz. Yorug`lik diodning katod uchini esa akkumlyatorli batareyaning manfiy ulagichiga (qizil ulash o`tkazgichi) ulaymiz.

Ko`pincha batareya yoki boshqa elektr manbani sxemada ko`rsatilmaydi. Bunday alternativ sxema qanday qilib kuchlanish zanjirda bog`lanishini ko`rsatib beradi. 1.3-rasmda alternativ sxemani ko`rishimiz mumkin:

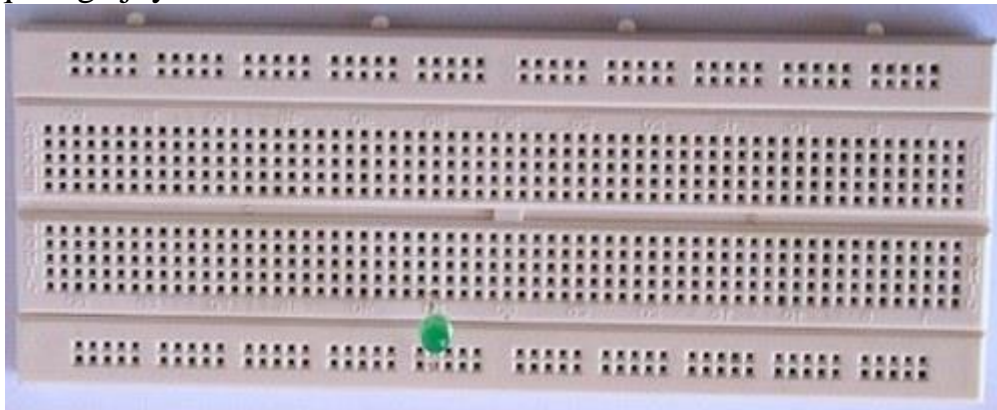


**1.3-rasm. LEDni ulash sxemasi**

### Zanjirni qurish

#### **1-qadam: Indikator (svetodiod) ni maketda joylashtirish.**

Yorug'lik diodning uzun (anod) o'tkazgichini maketning quyi relsiga joylashtiramiz, boshqa qo'rg'oshin o'tkazgichini esa rasmda ko'rsatilganidek maketning asosiy qismiga joylashtiramiz:

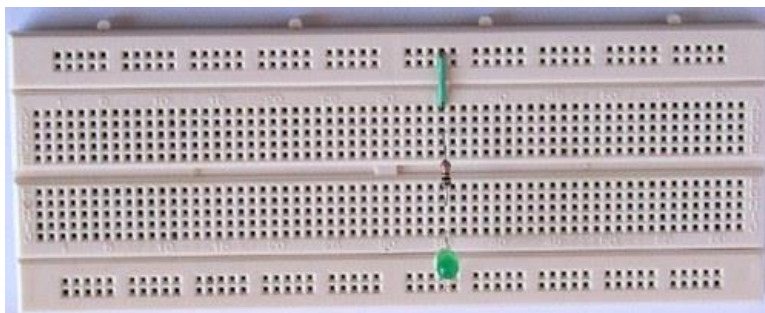


#### **2-qadam: Rezistorni maketga joylashtirish.**

Rezistor o'tkazgichlarini quyi rasmda ko'rsatilganidek qatlab qo'ying. Rezistorning bir uchini yorug'lik diodning katod chiqishi ustida to'g'ri chizig'ida ulang, qo'rg'oshinli uchini esa maketning o'rta kanalidan pastga joylashtiring. Bu LED katodini rezistorning bir uchiga ulash imkonini beradi. Platada rezistor atrofida qanday yo'l bo'lishi muhim emas.

#### **3-qadam: Maketga o'tkazgich ulagichni joylashtirish.**

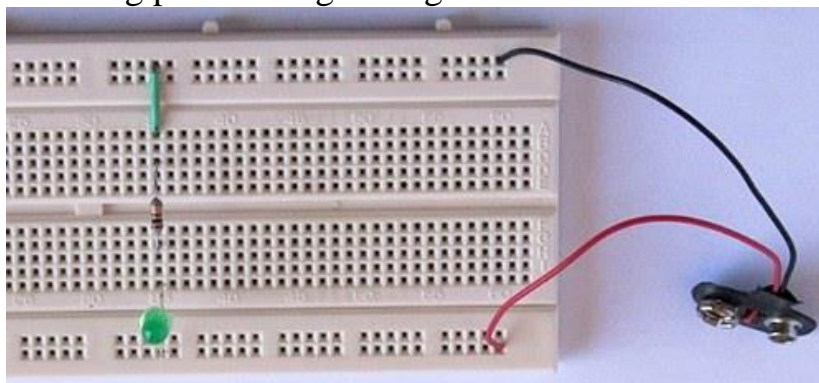
O'tkazgich ulagichni bir uchini rezistorning uchi ustidagi chiziqda joylashtiring bir uchi esa maketning yuqori relsida joylashgan bo'lishi kerak.





#### **4-qadam: Batareya qisqichlarini maketga joylashtirish.**

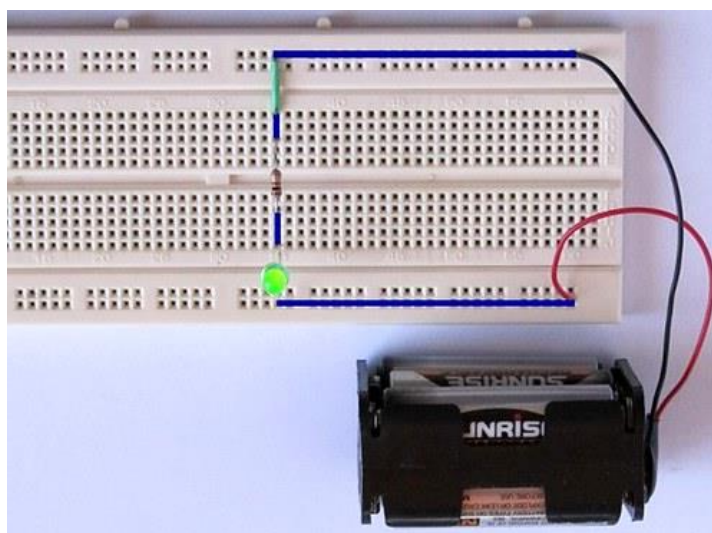
Batareya qora qisqichi ulagichini maketning yuqori relsiga ulang. Batareya qizil qisqichi ulagichini maketning pastki relsiga ulang.



#### **5-qadam: Batareyani batareya qisqichiga ulash.**

Va nihoyat zanjirga manba berish uchun va yorug'lik diodni yoqish uchun batareyani batareya qisqichiga joylashtiring.

Quyidagi rasm bu amaliy mashg'ulotimizda qurilgan sxemani ko'rsatib berib, unda maketda ulanish chiziqlarini ko'rsatadi. Bu chiziqlar ko'k rangda keltirilgan.



#### **Ishni bajarish tartibi**

1. Talabalar o'qituvchi tomonidan berilgan ma'lumotlar asosida 1 ta maketda mexatronik modul yig'adi.
2. Talaba bajargan ishlari bo'yicha o'qituvchiga yozma hisobot tayyorlab topshiradi.

#### **Takrorlash uchun savollar:**

1. Ishning maqsadi nima?
2. Rezistor va svetodiod haqida tushuncha bering.
3. Yig'ladigan sxema komponentlarini sanab o'ting.
4. Prinsipial sxemani chizib tushuntiring.

## Amaliy mashg'ulot “Virtulbreadboard” dasturida “Arduino” kontrollerini qo'llash.

*Mashg'ulotning maqsadi:* Talabalarga “*Arduino Uno*” qurilmasi bilan ishlash amaliy ko'nikmalarni hosil qilish.

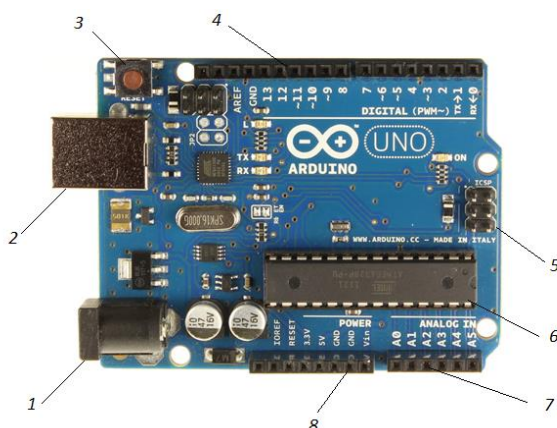
Mashg'ulot jarayonida talablar quyidagilarni o'rganishadi:

- “*Arduino Uno*”ni o'rnatish va unda dasturlash usulini;
- “*Arduino Uno*” bilan interfeys orqali o'zaro bog'lanishni.

### Qisqa nazariy ma'lumotlar

*Arduino* — noprofessional foydalanuvchilar uchun mo'ljallangan oddiy avtomatika va robototexnika tizimlarini qurish apparat-dasturiy vositalarning *savdo markasi* hisoblanadi. Uning dasturiy ta'minoti bepul dasturiy qobiq (IDE)dan iborat bo'lib, dasturlarni yaratish va apparaturani dasturlash uchun mo'ljallangan. *Arduino* ning apparat ta'minoti pechatlab o'rnatilgan plata bo'lib, rasmiy ishlab chiquvchi va boshqa ishlab chiquvchilar tomonidan sotiladi.

*Arduino Uno* (2.1-rasm) – *ATmega328* mikrokontrolleri asosida ishlangan qurilma hisoblanadi. Uning tarkibida mikrokontroller bilan ishlash uchun zarur barcha tarkibiy qismlar mavjud.



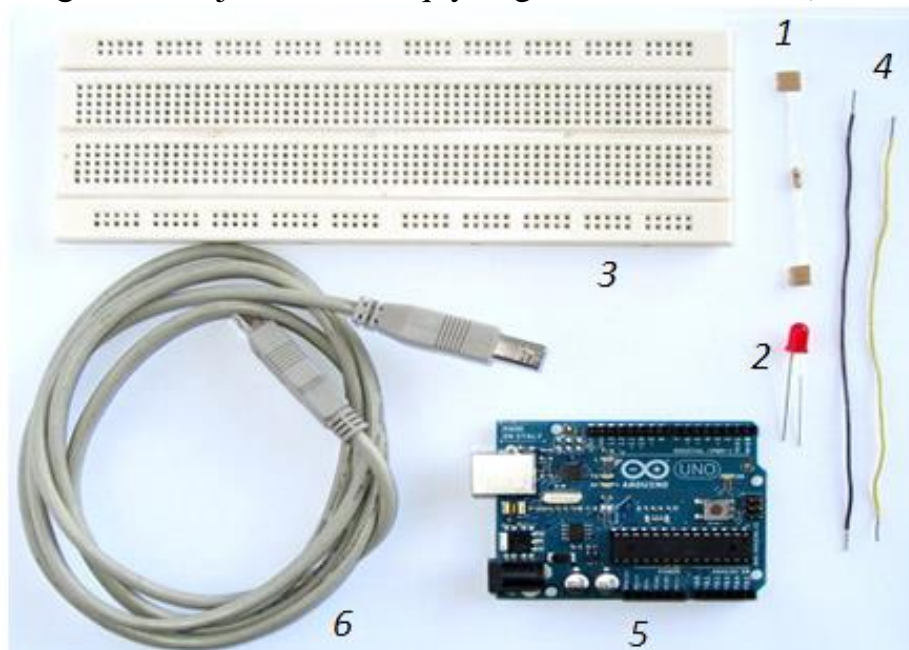
**2.1-rasm. *Arduino Uno* qurilmasi**

- 1-Elektr manba ulanish porti;
- 2-*USB* interfeys;
- 3-Tashlab yuborish tugmasi;
- 4-14 ta raqamli kirish/chiqish portlari, ulardan 6 tasi *KIM* (keng impul'sli modulyatsiya)-chiqish porti sifatida ishlatilishi mumkin;
- 5-Ichki sxemalarni dasturlash uchun ulanish (*ICSP*);
- 6-16 *MGts* li kvartslı rezonator;
- 7-6 ta analogli kirish porti;
- 8- *GND* - заземление.

Qurilma bilan ishlashdan oldin uni *AC/DC*-adapteri yoki elektr batareyaga manbasiga yoki *USB*-kabel orqali kompyuterga ulash zarur.

## Ishni bajarish uchun komponentlar

Amaliy mashg'ulotni bajarish uchun quyidagilar kerak bo'ladi (2.2-rasm):

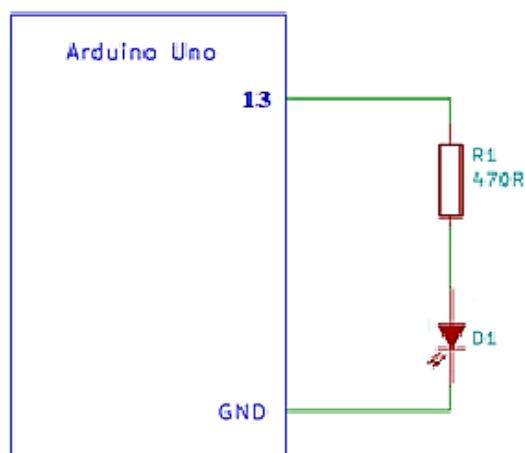


**2.2-rasm:**

1. 470 Om qarshilikka ega rezistor- (elektr sxemada *R1* belgili);
2. LED –(elektr sxemada *D1* belgili);
3. Maket (breadboard);
4. Maket platalasi uchun o'tkazgichlar;
5. “Arduino Uno” qurilmasi;
6. USB standart kabeli.

### Arduino Uno qurilmasiga LEDni ulash printsiptial elektr sxemasi

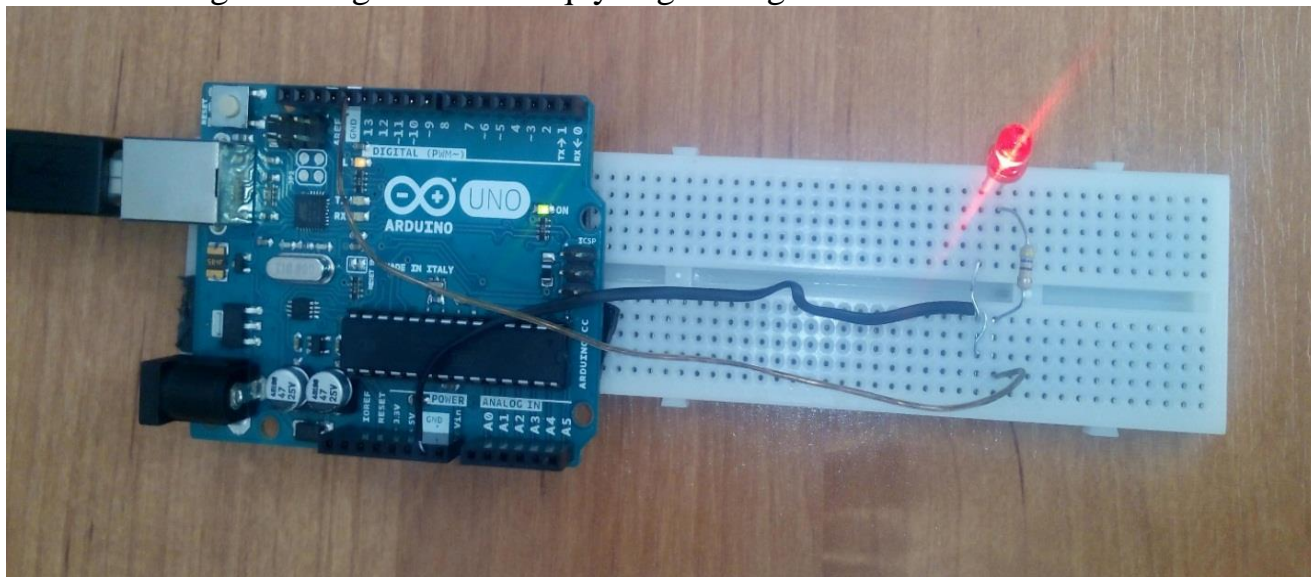
Biz birinchi elektron sxemani tashkil etamiz va uni “Arduino Uno” platasiga ulaymiz.



**2.3-rasm. Arduino Uno qurilmasiga LEDni ulash printsiptial elektr sxemasi**

*LED* katodi rezistor bilan bog'lanadi va boshqa chiqishi esa *Arduino* platasining *GND* kontakti bilan ulanishi kerak. So'ngra *Arduino* platasining 13-raqamli chiqishi bilan printsiptial sxemadagi rezistorning anod tomoni ulanishi zarur, ya'ni rezistorning ikkinchi uchi.

Sizning sxemangiz taxminan quyidagi rasimga o'xshash bo'lishi kerak:



### **“Arduino” dasturiy ta'minotini o'rnatish**

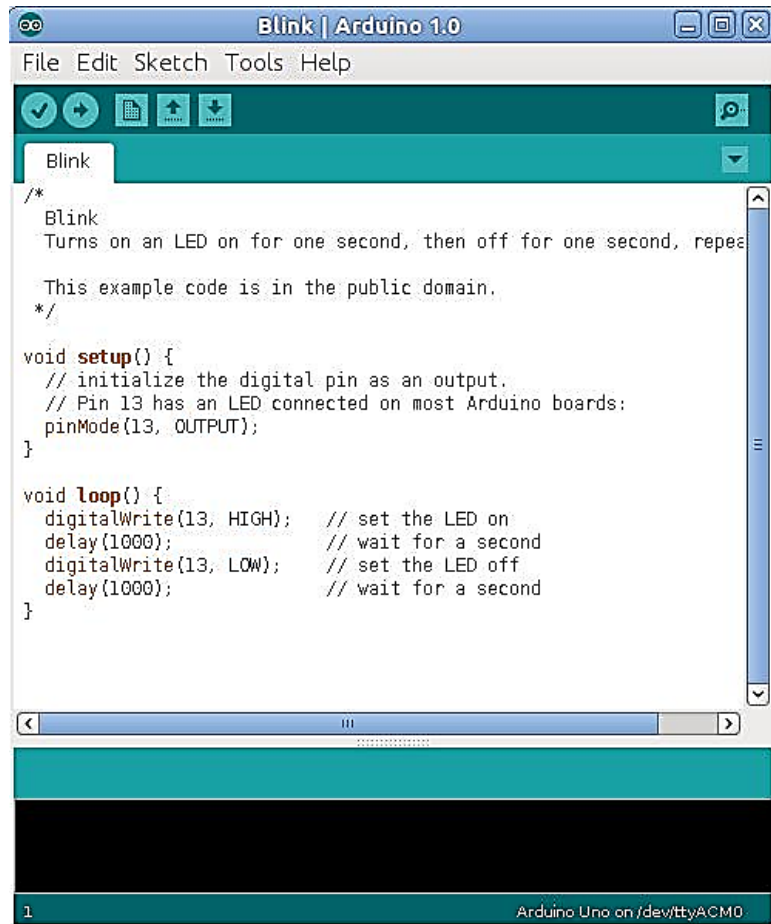
Biz kompyuterimizga *Arduino IDE* dasturiy ta'minotini o'rnatishimiz zarur.

### **“Arduino” da dasturlash**

Bu amaliy mashg'ulotimizda biz tayyor dasturdan foydalanamiz, qaysiki bu dastur *Arduino IDE* dasturlash tizimida kiritilgan (o'zi mavjud) bo'ladi. Bu tayyor dastur elektrik sxemadagi *LED* lampochkani o'chirish/yoqish buyrug'ini amalga oshiradi va lampochkani o'chiradi/yoqadi.

**Arduino** platani kompyuterga ulaymiz va *Arduino IDE* dasturlash tizimida quyidagi mavjud tayyor dasturni ochamiz:

1. *USB* kabelni *Arduino USB* portiga ulaymiz va boshqa uchini esa kompyuterning *USB* portiga ulaymiz (bu kompyuterda *IDE Arduino* dasturiy ta'minoti o'rnatilgan bo'lishi kerak).
2. *IDE Arduino* dasturlash tizimini ishga tushiramiz.
3. Dasturlash tizimida *Arduino* platasi uchun mos portni tanlaganingizga iqror bo'lamiz.
4. Dasturlash tizimining eng yuqori asosiy menyusida quyidagi buyruqni tanlaymiz: “*Fayl* → *Primeri* → *1.Basics* → *Blink*”
5. 2.4-rasmda ko'rsatilganidek tayyor dastur kodli yangi oyna paydo bo'ladi



## 2.4-rasm.Blink tayyor dasturning IDE Arduino dagi oynasi

Plataga *Arduino* dasturini yuklaymiz:

1. Dasturni *Arduino* yuklash uchun asosiy instrumentlar panelidagi *Upload* tugmani tanlaymiz (2.5-rasmda qizil chiziq bilan belgilangan).



## 2.5-rasm. Yuklash tugmasi

2. Dastur plataga yuklanishi zarur va so`ngra ishlashni boshlashi kerak. Dastur ishini boshlaganda siz *LED* lampochkasini yonish/o`chishini ko`rishingiz mumkin.

### Ishni bajarish tartibi

1.Talabalar o`qituvchi tomonidan berilgan ma`lumotlar asosida “*Arduino Uno*” qurilmasini ishga tushirib beradi.

2.Talaba bajargan ishlari bo`yicha o`qituvchiga yozma hisobot tayyorlab topshiradi.

## Takrorlash uchun savollar:

1. *Arduino* savdo markasiga tushuncha bering.
2. *Arduino Uno* platasining tuzilishini aytib bering.
3. Ishni bajarish uchun komponentlarni ayting.
4. *Arduino Uno* qurilmasiga *LED*ni ulash printsiptial elektr sxemasini chizib tushuntiring.
5. Plataga *Arduino* dasturini yuklash tartibini aytib bering.

## Amaliy mashg'ulot

### “Virtulbreadboard” dasturi yordamida “Arduino” kontrolleriga yorug'lik diodlarini ulash.

*Mashg'ulotning maqsadi:* Talabalarda **Arduino** ga yorug'lik diodlarni ulash va ularning yonish ketma-ketligini boshqarish ko'nikmalarini shakllantirish.

Amaliy mashg'ulot jarayonida talabalar quyidagilarni bilib olishadi:

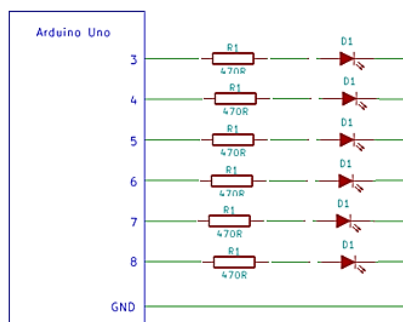
- Yorug'lik diodlarni ulash uchun **Arduino** dastur tuzishni;
- Yorug'lik diodlarni yonish/o'chishini ketma-ketligini boshqarish uchun **Arduino** interfeysi bilan o'zaro muloqot qilishni.

## Ishni bajarish uchun kerakli jihozlar:

1. Rezistor 470 Om - belgilanishi *R1*;
2. *LED* – belgilanishi *D1*;
3. Maket platasi;
4. Maket platasi uchun o'tkazgichlar;
5. *Arduino* platasi;
6. *USB* standart kabeli.

## Yorug'lik diodlarni *Arduinoga* ulash printsiptial elektr sxemasi

Printsiptial elektr sxemasi 3.1-rasmda ko'rsatilgan. Bu juda oddiy bo'lib, olti dona yorug'lik diodi *Arduino* bilan 3-raqamli portdan to 8 – portgacha chiqish portlari orqali ulangan.



3.1-rasm. Yorug'lik diodlarni *Arduinoga* ulash printsiptial elektr sxemasi

## Zanjirni qurish

Svetodiodlarni bir-biriga yaqin holatda maketga shunday joylashtiringki anod (uzun oyoqlari) maketning chap tomonida joylashgan bo`lsin (maketni vertikal joylashuviga ko`ra), katod oyoqchalari esa maketning o`ng tomonida joylashgan bo`lsin.

470 Om rezistorlarni bir tomonini svetodiodning anod tomonida ulash zarur, ikkinchi tomonini esa o`tkazgichlar yordamida 2-raqamli portdan boshlab 8-raqamli portgacha Arduino ga ulab chiqing.

Zanjirni ulash ishi tugagach Arduino platasini *USB* kabel yordamida kompyuterga ulang.

## Dasturni yuklash

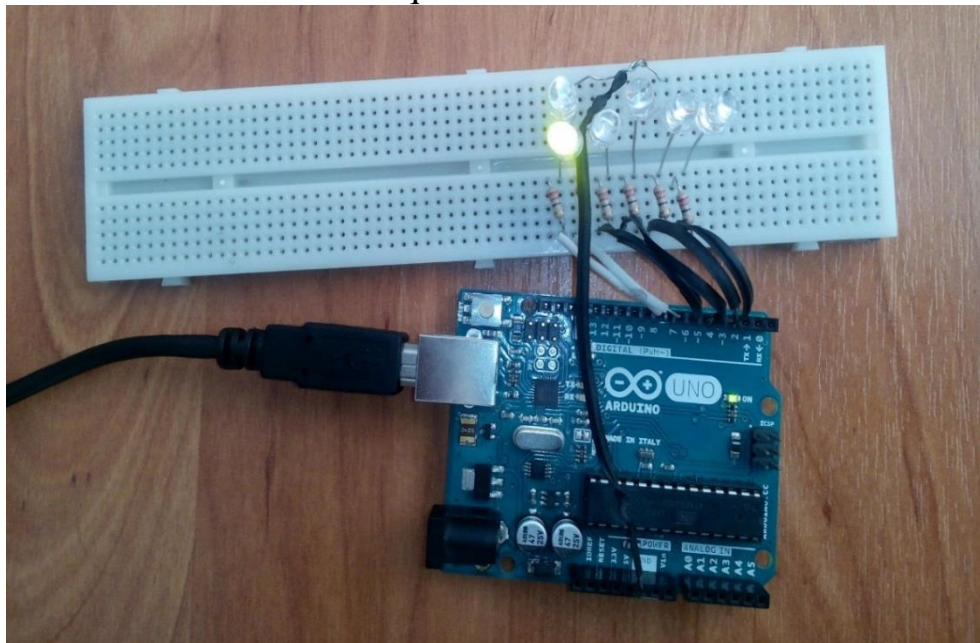
Yorug`lik diodlarni ketma-ket boshqarish dasturi *Knight Rider* birlamchi kodi quyida keltirilgan. Ushbu kodni nusxalang va *Arduino IDE* ga joylashtiring.

```
/q
Knight Rider

Knight rider display on 6 LEDs
q/
void setup() {
  // set up pins 2 to 8 as outputs
  for (int i = 2; i < 8; i++) {
    pinMode(i, OUTPUT);
  }
}
// function to switch all LEDs off
void allLEDsOff(void)
{
  for (int i = 2; i < 8; i++) {
    digitalWrite(i, LOW);
  }
}
void loop() {
  // move on LED to the right
  for (int i = 2; i < 8; i++) {
    allLEDsOff();
    digitalWrite(i, HIGH);
    delay(200);
  }
  // move on LED to the left
  for (int i = 7; i > 1; i--) {
    allLEDsOff();
    digitalWrite(i, HIGH);
    delay(200);
  }
}
```

Dasturni *Arduino* ga yuklang va agar tuzilgan zanjir to`g`ri bo`lsa ushbu dastur ishini boshlaydi va yorug`lik diodlari ketma-ket yonib-o`chishni boshlaydi.

Shuni ham aytish mumkinki, dastur kodini o`zgartirish orqali yorug`lik diodlarni yonib-o`chish ketma-ketliklarini boshqarish mumkin.



**3.2-rasm. Yorug`lik diodlari ketma-ket yonib-o`chishi  
Ishni bajarish tartibi**

1. Talabalar o`qituvchi tomonidan berilgan ma`lumotlar asosida 1 ta “Arduino Uno”ga yorug`lik diodlarini ulab yig`adi.

2. Talaba bajargan ishlari bo`yicha o`qituvchiga yozma hisobot tayyorlab topshiradi.

### **Takrorlash uchun savollar:**

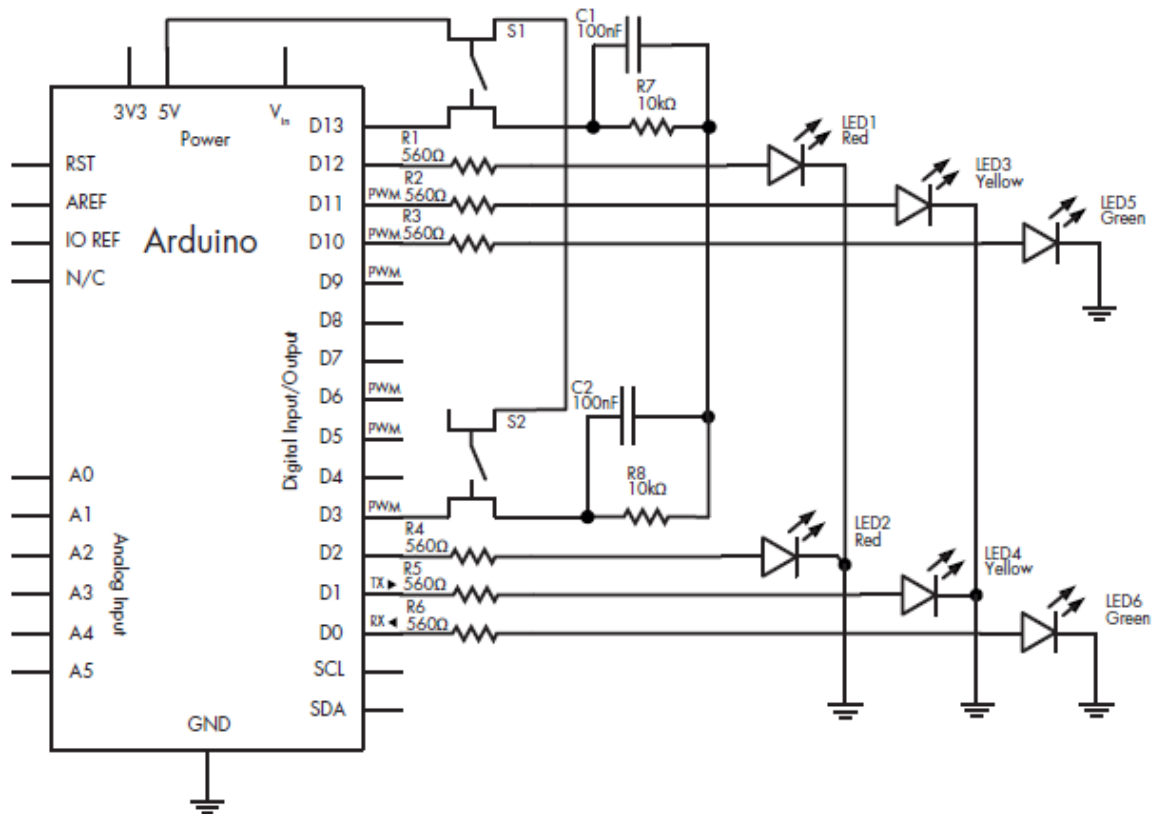
1. Mashg`ulotning maqsadini ayting.
2. Ishni bajarish uchun kerakli jihozlarni sanab o`ting.
3. Yorug`lik diodlarni Arduinoga ulash printsiplial elektr sxemasi zanjirini yig`ishni tushuntiring.
4. Dasturni yuklash tartibini tushuntiring.

### **amaliy mashg`ulot.**

#### **“Virtualbreadboard” dasturi yordamida LED ekranlarini ishga tushirish.**

Biz olti LEDni nazorat qilamiz va ikkita tugmachani kiritishimiz sababli, dizayn juda qiyin bo`lmaydi. Shakl quyidagi sxema ko`rsatilgan loyihimiz uchun.





```
// Svetoforni boshqarish dasturi
// define the pins that the buttons and lights are connected to:
#define westButton 7
#define eastButton 13
#define westRed 2
#define westYellow 1
#define westGreen 0
#define eastRed 12
#define eastYellow 11
#define eastGreen 10
#define yellowBlinkTime 500 // 0.5 seconds for yellow light blink
Building Blocks 77
v boolean trafficWest = true; // west = true, east = false
w int flowTime = 10000; // amount of time to let traffic flow
x int changeDelay = 2000; // amount of time between color changes
void setup()
{
// setup digital I/O pins
pinMode(westButton, INPUT);
pinMode(eastButton, INPUT);
pinMode(westRed, OUTPUT);
pinMode(westYellow, OUTPUT);
pinMode(westGreen, OUTPUT);
pinMode(eastRed, OUTPUT);
pinMode(eastYellow, OUTPUT);
```

```

pinMode(eastGreen, OUTPUT);
// set initial state for lights - west side is green first
digitalWrite(westRed, LOW);
digitalWrite(westYellow, LOW);
digitalWrite(westGreen, HIGH);
digitalWrite(eastRed, HIGH);
digitalWrite(eastYellow, LOW);
digitalWrite(eastGreen, LOW);
}
void loop()
{
if ( digitalRead(westButton) == HIGH ) // request west>east traffic flow
{
if ( trafficWest != true )
// only continue if traffic flowing in the opposite (east) direction
{
trafficWest = true; // change traffic flow flag to west>east
delay(flowTime); // give time for traffic to flow
digitalWrite(eastGreen, LOW); // change east-facing lights from green
// to yellow to red
digitalWrite(eastYellow, HIGH);
delay(changeDelay);
digitalWrite(eastYellow, LOW);
digitalWrite(eastRed, HIGH);
delay(changeDelay);
for ( int a = 0; a < 5; a++ ) // blink yellow light
{
digitalWrite(westYellow, LOW);
delay(yellowBlinkTime);
digitalWrite(westYellow, HIGH);
delay(yellowBlinkTime);
}
digitalWrite(westYellow, LOW);
digitalWrite(westRed, LOW); // change west-facing lights from red to green
digitalWrite(westGreen, HIGH);
}
}
}

```

## 78 Chapter 4

```

if ( digitalRead(eastButton) == HIGH ) // request east>west traffic flow
{
if ( trafficWest == true )
// only continue if traffic flow is in the opposite (west) direction
{
trafficWest = false; // change traffic flow flag to east>west
delay(flowTime); // give time for traffic to flow
}
}
}

```

```

digitalWrite(westGreen, LOW);
// change west lights from green to yellow to red
digitalWrite(westYellow, HIGH);
delay(changeDelay);
digitalWrite(westYellow, LOW);
digitalWrite(westRed, HIGH);
delay(changeDelay);
for ( int a = 0 ; a < 5 ; a++ ) // blink yellow light
{
digitalWrite(eastYellow, LOW);
delay(yellowBlinkTime);
digitalWrite(eastYellow, HIGH);
delay(yellowBlinkTime);
}
digitalWrite(eastYellow, LOW);
digitalWrite(eastRed, LOW); // change east-facing lights from red to green
digitalWrite(eastGreen, HIGH);
}
}
}

```

Yuqoridagi dasturni Arduino platformasiga yuklaganimizda sevetofor qanday qilib dastur orqali boshqarilishini ko'rishimiz mumkin.

## **Amaliy mashg'ulot**

### **“Virtulbreadboard” dasturi yordamida temperaturani boshqarish.**

*Mashg'ulot maqsadi:* Talabalarda «Arduino Uno» mikrokontroller platasi va harorat datchigi yordamida haroratni o'lchash va boshqarish ko'nikmalarini shakllantirish.

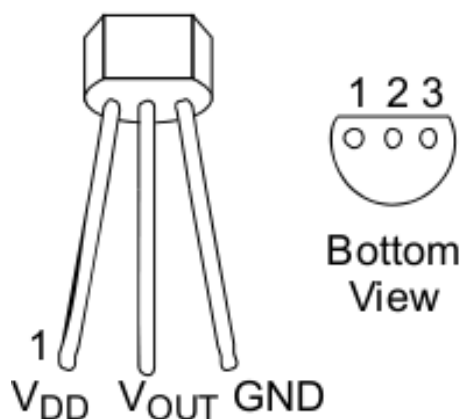
*Amaliy mashg'ulotni bajarish davomida quyidagilar o'rganiladi:*

- MCP 9700 harorat datchigining tuzilishi va ishlash printsipini;
- datchikni mikrokontrollerga ulashni;
- datchikni ishlatuvchi dasturni tuzishni;
- datchik yordamida haroratni o'lchashni va boshqarishni.

### **Qisqa nazariy ma'lumotlar**

MCP 9700 harorat datchigi (4.1-rasm) hech qanday qo'shimcha qurilmalarni ulamasdan bevosita haroratni o'lchash imkonini beruvchi datchik hisoblanadi. Datchik quyidagi xarakteristikalariga ega (4.1-jadval):

**3-Pin TO-92  
MCP9700/9701  
Only**



**4.1.-rasm. MCP 9700 datchigi**

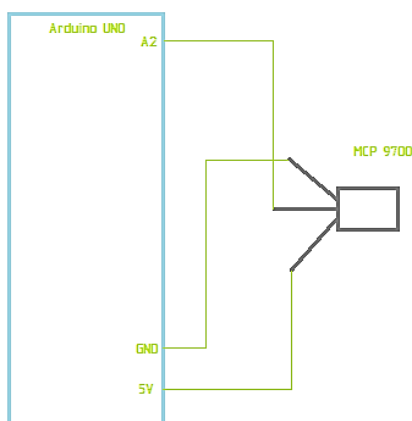
**4.1-jadval. MCP 9700 datchik xarakteristikasi**

<b>Xarakteristika nomi</b>	<b>Qiymati</b>
Kuchlanishga nisbatan haroratning o`zgarishi	10mV/°C
0°C dan +70°C gacha bo`lgan oraliqda asbobning aniqligi	± 4 °C
-40°C dan +150°C gacha bo`lgan oraliqda asbobning aniqligi	-4°C/+6°C
Iste`mol kuchlanishi	2.3 V dan 5.5 V gacha
O`lchash chegarasi	-40°C ÷ +150°C

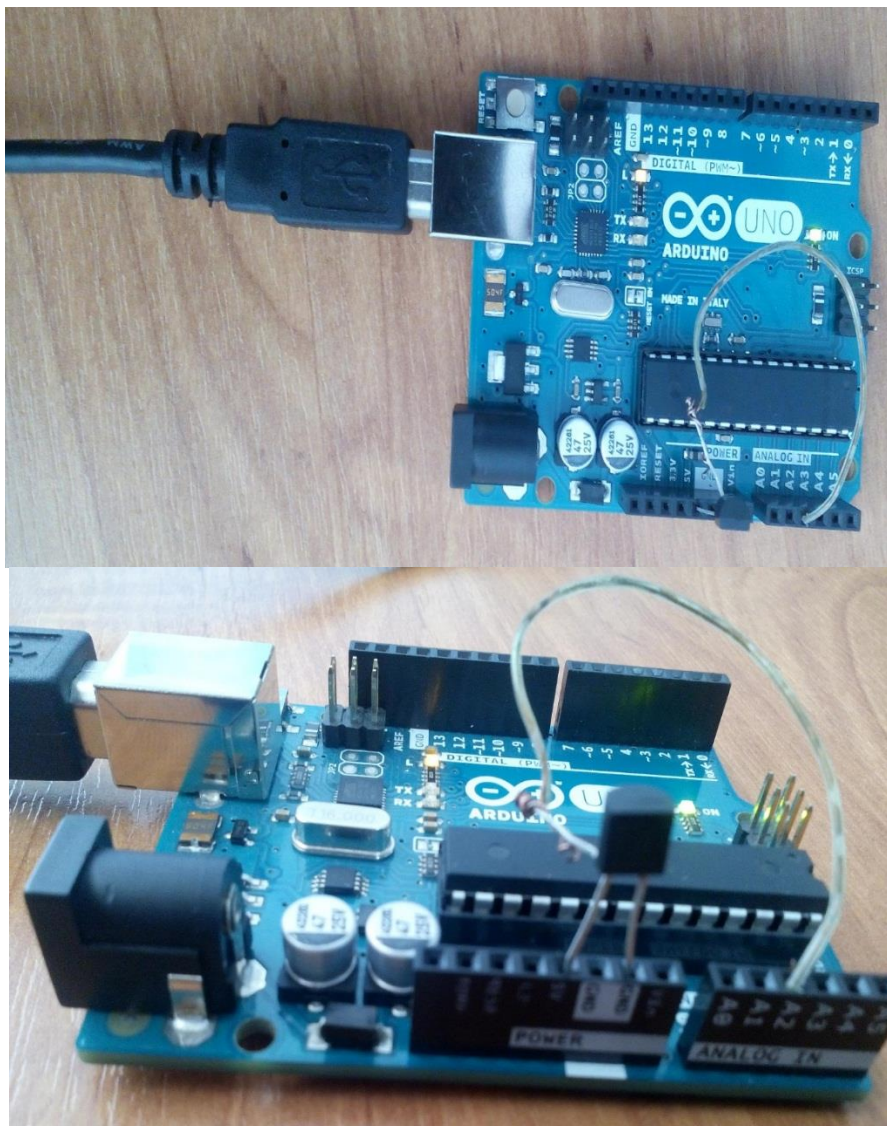
**Ishni bajarish tartibi:** MCP 9700 datchigi 3 ta kontaktdan iborat bo`lib, ularning har biri mikrokontrollerning tegishli nuqtalariga ulanishi kerak:

- Datchikning birinchi kontakti iste`mol kuchlanishi beriladigan nuqtaga ulanadi (3.5 V yoki 5 V);
- Ikkinchi kontakti analogli chiqish nuqtasiga ulanadi (A0...A5);
- Uchinchi kontakt esa *GND* (ground(zazemlenie)) nuqtasiga ulanadi.

Ulanish sxemasi quyida keltirilgan



#### 4.2-rasm. “Arduino Uno” va MCP 9700 datchigi ulash sxemasi



4.3-rasm. “Arduino Uno” va MCP 9700 datchigini ulash fotosuratlarini

Endi datchikni ishlatadigan mikrokontroller dasturini yozamiz:

```
float temp;
```

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

```
void loop () {  
  temp = analogRead(2)*5/1024.0;  
  temp = temp - 0.5;  
  temp = temp / 0.01;  
  Serial.println(temp);  
  delay(500);  
};
```

Bu dastur datchikdan keladigan signalni qabul qiladi va biz harorat haqida tasavvurga ega bo'lishimiz uchun uni raqamlarga aylantirib *Arduino* dasturining maxsus oynasiga ko'rsatadi (*Ctrl+Shift+M*). Endi tushunarliroq bo'lishi uchun dasturni bosqichma – bosqich ko'rib chiqamiz:

*Shuni esda tutish kerakki, siz temperaturani o'lchash datchigini 2.7V dan to 5.5V gacha manbaga ulab o'lchash imkoniyatiga egasiz. Bizning misolda sizga 5V manba portiga datchigni ulab temperaturani o'lchashni ko'rib o'tamiz, lekin shuni inobatga olingki siz 3.3V manbaga datchikni ulab ham hisoblash imkoniga egasiz. Agar siz *Arduino* platasining 5V portiga datchikni to'g'ridan-to'g'ri ulab ishlatadigan bo'lsangiz, siz quyidagi formuladan foydalanib analog port orqali o'qilayotgan 10 bitli ma'lumotni aniqlashingiz mumkin:*

$$\text{Voltage at pin in milliVolts} = (\text{reading from ADC}) * (5000/1024)$$

*Ushbu formula 0-1023 bo'lgan qiymatni 0-5000mV (= 5V) kuchlanish qiymatiga aylantirib beradi.*

*Agar siz *Arduino* da 3.3V portni ishlatadigan bo'lsangiz, siz quyidagi formuladan foydalanishingiz mumkin:*

$$\text{Voltage at pin in milliVolts} = (\text{reading from ADC}) * (3300/1024)$$

*Ushbu formula 0-1023 bo'lgan sonli qiymatnin 0-3300mV (= 3.3V) bo'lgan kuchlanish qiymatiga aylantirib beradi.*

*Keyinchalik, millivoltni temperaturaga aylantirish kerak, bunda quyidagi formuladan foydalanamiz:*

$$\text{Centigrade temperature} = [(\text{analog voltage in mV}) - 0,5] / 0,01$$

1. `temp = analogRead(2)/5/1024.0;` - Analog kirish sonli qiymat signalini kuchlanish signaliga aylantiradi.

2. `temp = temp - 0.5;` va `temp = temp / 0.01;` - esa millivolt qiymatini temperatura qiymatiga aylantirib beradi.

### **Ishni bajarish tartibi**

1. Talabalar o'qituvchi tomonidan berilgan ma'lumotlar asosida "*Arduino Uno*" yordamida atrof-muhit temperaturasini o'lchaydi.

2. Talaba bajargan ishlari bo'yicha o'qituvchiga yozma hisobot tayyorlab topshiradi.

### **Takrorlash uchun savollar:**

1. Amaliy mashg'ulotni bajarish davomida nimalar o'rganiladi?
2. *MCP 9700* harorat datchigi haqida ma'lumot bering.
3. Ishni bajarish tartibini ayting.
4. Datchikni ishlatadigan mikrokontroller uchun dasturni tushuntiring.



