

## **№2. AMALIY MASHG'ULOT**

### **AQLLI SHAHARNI LOYIHALASHDA FOYDALANILADIGAN DASTURIY VOSITALARNI O'RGANISH.**

**Ish maqsadi :** Arduino platformasi qurilmasi bilan tanishish Uno , Wiring tilida dastur tuzilishi bilan . Arduino uchun dasturlar yozishni o'rganing . Arduino IDE ishlab chiqish muhiti bilan tanishing , ishni boshlashdan oldin dasturiy ta'minotni qanday sozlashni o'rganing. Tizimlarni dasturlash uchun zarur bo'lgan bilimlarni o'rganish Arduino C tilida.

#### **№2.1. Amaliy mashg'ulot Arduino IDE dasturini o'rnatish, platani kompyuterga ulash**

**Mashg'ulotning maqsadi :** Arduino IDE ishlab chiqish muhiti bilan tanishish uchun ishni boshlashdan oldin dasturiy ta'minotni sozlashni o'rganing. Doskani kompyuterga ulash va birinchi dasturni yuklab olishni o'rganish.

#### **Qisqacha nazariya**

**Arduino IDE** - bu Windows, MacOS va Linux uchun integratsiyalashgan ishlab chiqish muhiti bo'lib, u C va C++ tillarida ishlab chiqilgan bo'lib, Arduino -mos platalarga, shuningdek, boshqa ishlab chiqaruvchilarning platalariga dasturlar yaratish va yuklash uchun mo'ljallangan .

Dasturiy ta'minotni o'rnatish va Arduino UNO R3 kontrollerini kompyuterga ulash uchun sizga kerak bo'ladi:

- boshqaruv paneli;
- USB kabeli (odatda kiritilgan)
- Internetga ulangan Windows shaxsiy kompyuteri.

Plata kompyuterning USB portidan quvvatlanishi mumkin, shuning uchun tashqi quvvat manbai talab qilinmaydi.



#### **Arduino IDE Integrated Development Environment dasturini o'rnatish .**

Avvalo, dasturning so'nggi versiyasini yuklab olishingiz kerak. ZIP arxivini rasmiy Arduino qo'llab-quvvatlash saytidan yuklab olishingiz mumkin. [ushbu havolada](https://www.arduino.cc/en/software) ( <https://www.arduino.cc/en/software> ) . Kerakli operatsion tizim bilan qatorni tanlashingiz kerak - **Windows ZIP fayli** .

Atrof-muhit uchun manba kodi GNU General Public License 2-versiyasi ostida chiqarilgan. Maxsus kod tuzilmasi qoidalaridan foydalangan holda C va C++ tillarini qo'llab-quvvatlaydi.

## Download the Arduino Software



### ARDUINO 1.6.8

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board. Refer to the [Getting Started](#) page for installation instructions.

**Windows** Installer  
[Windows ZIP file for non admin install](#)

**Mac OS X** 10.7 Lion or newer

**Linux** 32 bits  
**Linux** 64 bits

[Release Notes](#)  
[Source Code](#)  
[Checksums](#)

Papka yarating, masalan , Arduino va unga zip faylni chiqarib oling.

Имя	Тип	Размер	Дата
[..]		<Папка>	28.03.2016 19:38
[dist]		<Папка>	28.03.2016 19:38
[drivers]		<Папка>	28.03.2016 19:38
[examples]		<Папка>	28.03.2016 19:38
[hardware]		<Папка>	28.03.2016 19:38
[java]		<Папка>	28.03.2016 19:38
[lib]		<Папка>	28.03.2016 19:38
[libraries]		<Папка>	28.03.2016 19:38
[reference]		<Папка>	28.03.2016 19:38
[tools]		<Папка>	28.03.2016 19:38
[tools-builder]		<Папка>	28.03.2016 19:38
arduino	exe	402 432	09.03.2016 16:11
arduino-builder	exe	3 857 920	09.03.2016 16:11
arduino.l4j	ini	20	09.03.2016 16:11
arduino_debug	exe	399 872	09.03.2016 16:11
arduino_debug.l4j	ini	20	09.03.2016 16:11
libusb0	dll	43 520	09.03.2016 16:11
msvcpr100	dll	421 200	09.03.2016 16:11
msvcr100	dll	770 384	09.03.2016 16:11
revisions	txt	74 967	09.03.2016 16:11

### Arduino platasini ulash .

USB kabeli yordamida platani kompyuteringizga ulang. LED (yorlig'i ON) yonishi kerak, bu plataga quvvat etkazib berilayotganligini ko'rsatadi.

### Haydovchi o'rnatish.

Men USB-UART-dan ko'prik sifatida foydalanadigan Arduino UNO R3 platalarini bilaman

- ATmega16 U2 chipi (asl versiya)
- chip CH340G (Xitoy kloni).

Ushbu variantlar uchun drayverlarni o'rnatish jarayonlari boshqacha.

### ATmega16U2 interfeysi konvertori bilan ARDUINO UNO uchun drayverlarni o'rnatish.

Platani kompyuterga ulagandan so'ng, Windows o'zi drayverni o'rnatish jarayonini boshlaydi. Biroz vaqt o'tgach, muvaffaqiyatsiz urinish xabari paydo bo'ladi.

Drayv qo'lda o'rnatilishi kerak. Buni amalga oshirish uchun **Boshlash -> Boshqaruv paneli -> Tizim -> Qurilma menejeriga o'ting .**

**va LPT) bo'limida** ogohlantirish sariq belgisi bo'lgan **Arduino UNO qurilmasi** bo'lishi kerak .

Belgini o'ng tugmasini bosing.

Tanlang **Haydovchi yangilash**.

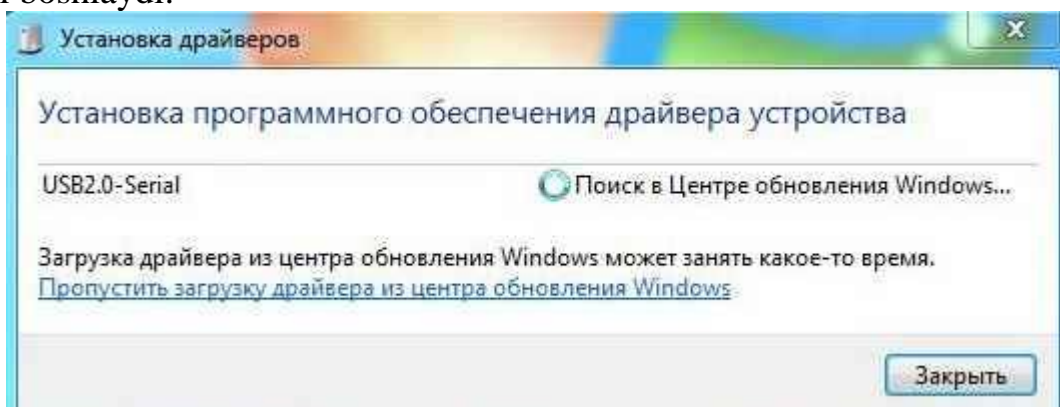
Uzoqroqda **Ushbu kompyuterda drayverlarni qidiring**.

Haydovchining joylashishini qo'lda belgilang. **ArduinoUNO.inf** fayli arxiv ochilgan papkaning **Drivers katalogida** joylashgan.

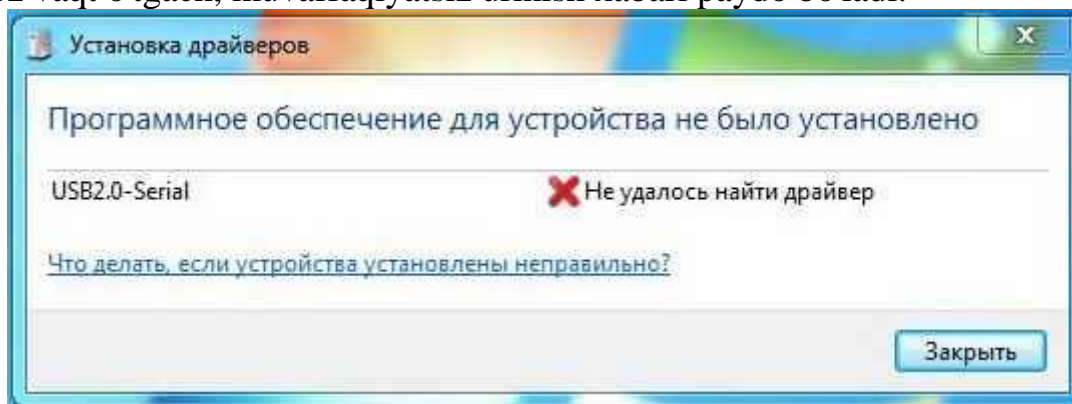
Portlar (COM va LPT) bo'limida yangi, virtual MAQOMOTI paydo bo'ladi. Uning raqamini eslab qolish kerak.

**CH340G interfeysli konvertorli ARDUINO UNO uchun drayverni o'rnatish (Xitoy kloni).**

Platani kompyuterga ulagandan so'ng, Windows o'zi drayverni o'rnatish jarayonini boshlaydi.



Biroz vaqt o'tgach, muvaffaqiyatsiz urinish xabari paydo bo'ladi.



Drayver qo'lda o'rnatilishi kerak. Buni amalga oshirish uchun **Boshlash -\u003e Boshqaruv paneli -\u003e Tizim -\u003e Qurilma menejeriga o'ting**.

Yangi **USB2.0-seriyali qurilma** sariq ogohlantirish belgisi bilan paydo bo'ldi.

O'rnatish faylini ishga tushiramiz ch341ser.exe.

Arduino IDE ko'plab umumiy kirish va chiqish tartiblarini ta'minlovchi Wiring loyihasidan dasturiy ta'minot kutubxonasini taqdim etadi.

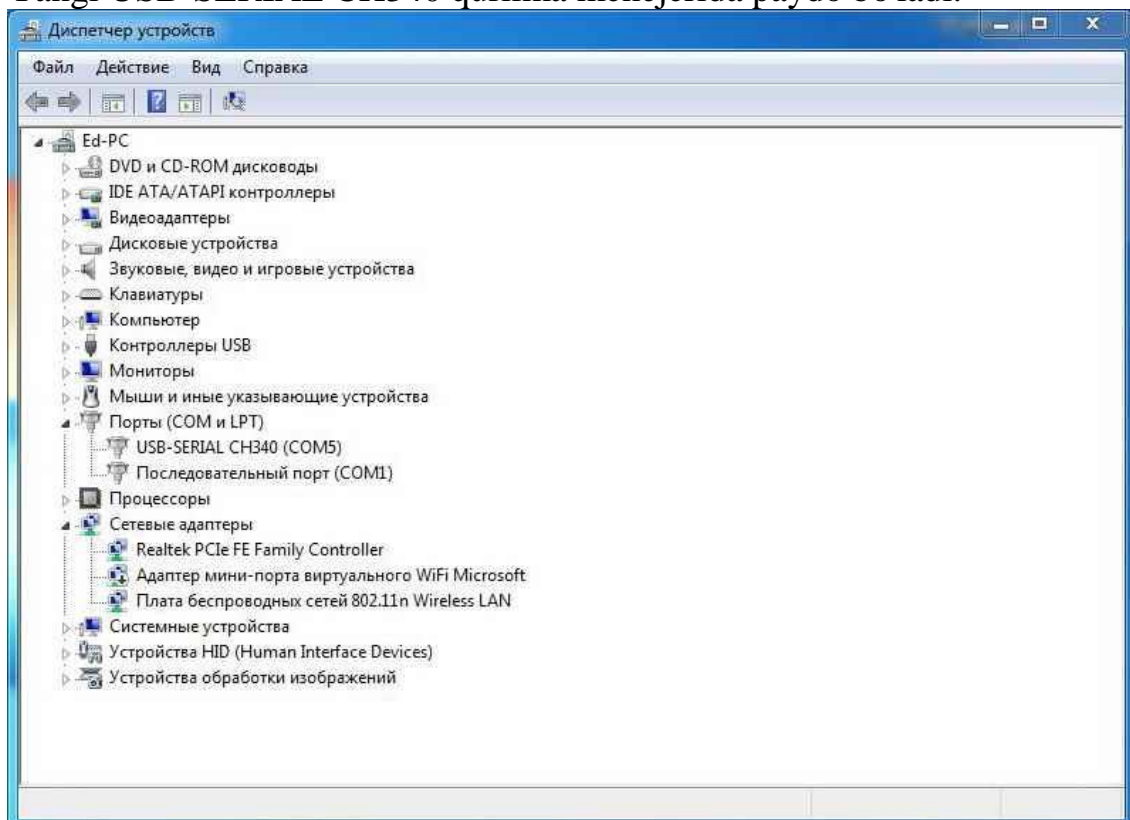


**INSTALL** ni tanlang .

Muvaffaqiyatli o'rnatish haqida xabar kutmoqdamiz.



Yangi USB-SERIAL CH340 qurilma menejerida paydo bo'ladi.



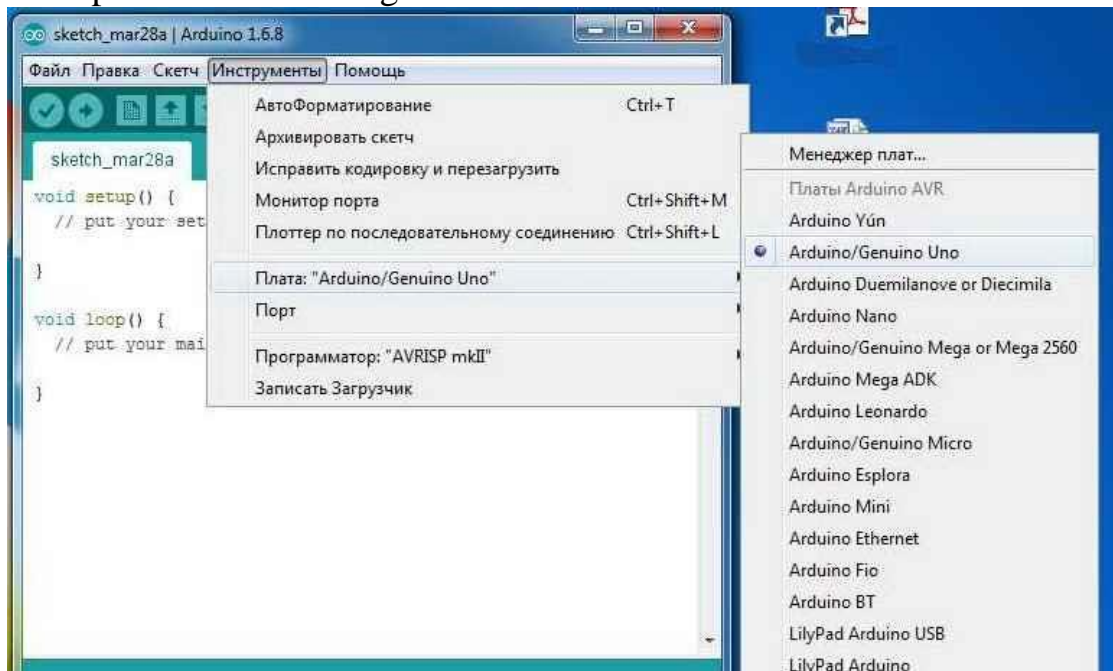
COM port raqamini eslab qolishingiz kerak.

**Arduino IDE integratsiyalashgan rivojlanish muhitini ishga tushirish .**

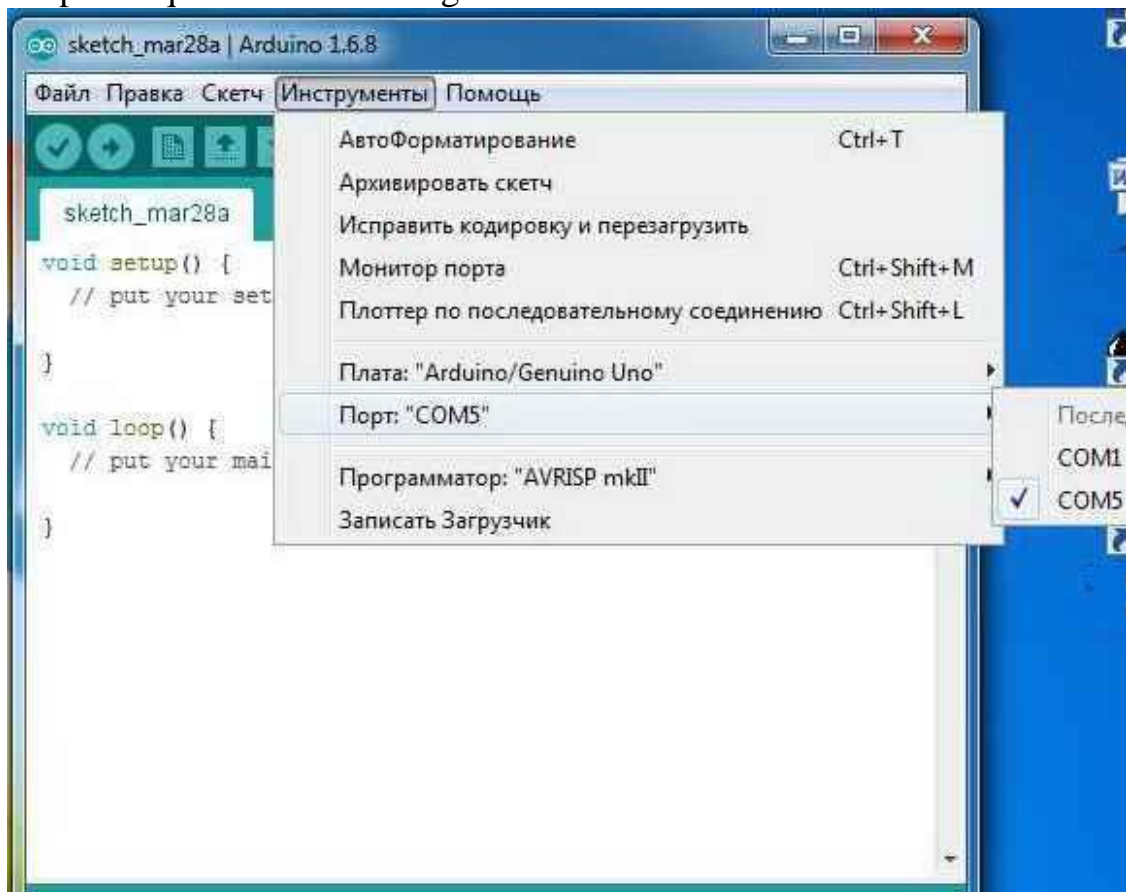


arduino.exe faylini ishga tushiring.

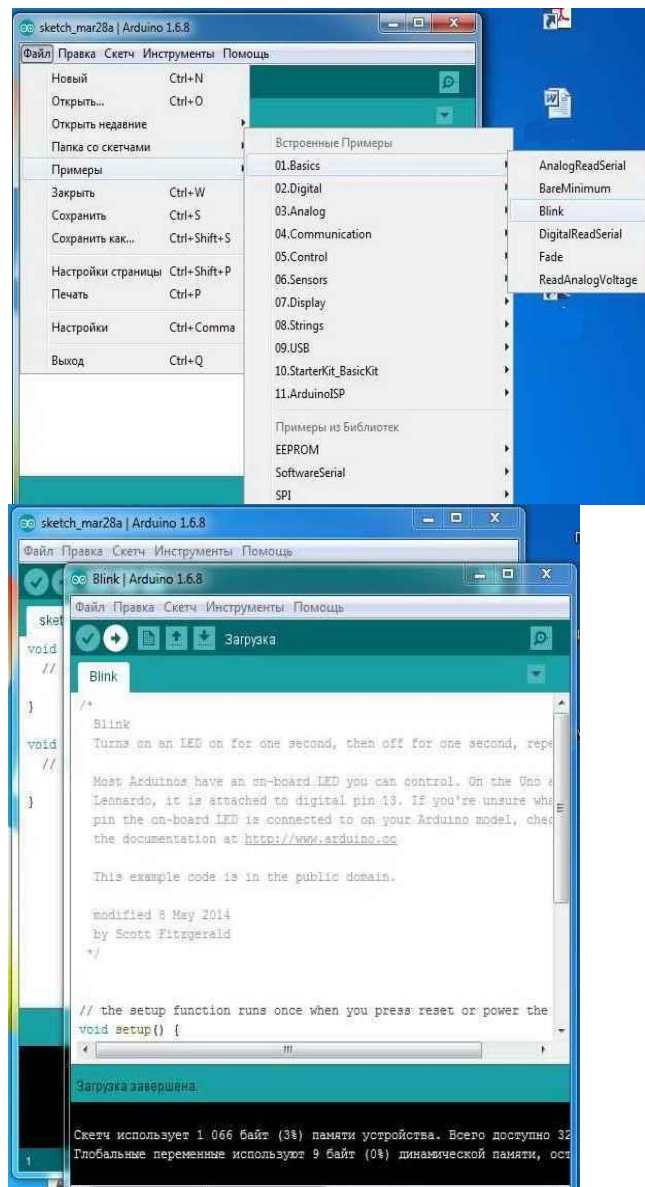
Arduino platasi turini tanlang : **Asboblar -> Plata -> Arduino UNO .**



COM port raqamini ko'rsatishingiz kerak: **Asboblar -> Port .**



Tizimning ishlashini tekshirish uchun siz birinchi eskizni - miltillovchi LEDni ishga tushirishingiz mumkin. Buning uchun: **File -> Examples -> 01.Basics -> Blink**



**Yuklab olish tugmasini bosing .**

Biz dasturning yuklanishini kutamiz va L harfi bilan belgilangan doskadagi LED soniyada bir marta miltillay boshlaydi. Shunday qilib, ular hamma narsani to'g'ri qilishdi.

Foydalanuvchi tomonidan yozilgan kod eskizni va asosiy dastur tsiklini ishga tushirish uchun faqat ikkita asosiy funksiyani talab qiladi, ular kompilyatsiya qilinadi va asosiy () dastur stub bilan IDE taqsimotiga kiritilgan GNU asboblari zanjiri bilan bajariladigan tsikl dasturiga bog'lanadi. Avrdude dasturidan bajariladigan kodni plata proshivkasidagi yuklovchi dasturi tomonidan Arduino platasiga yuklangan o'n oltilik kodlangan matn fayliga aylantirish uchun foydalanadi.

Arduino -ning mashhurligi bilan boshqa ishlab chiqaruvchilar dasturiy ta'minot platformasi sifatida rasmiy Arduino mikrokontroller liniyasi tomonidan qo'llab-quvvatlanmaydigan boshqa mikrokontrollerlarga eskizlarni yaratishi va yuklashi mumkin bo'lgan maxsus kompilyatorlar va ochiq manbali vositalarni (yadrolarni) joriy qila boshladilar .

2019-yil oktabr oyida Arduino Foundation nosozliklarni tuzatish va boshqa ilg'or funksiyalar bilan yangi Arduino Pro IDE -ga erta kirishni ta'minlay boshladi .

### Hisobotda quyidagilar bo'lishi kerak:

1. Amaliyot nomlari va ishning maqsadi.
2. Breadboard va tadqiqot tushunchasi.
3. Tadqiqotda foydalaniladigan texnik vositalarning xarakteristikalar.
4. Dastur kodini va kompilyatsiya skrinshotlarini yozish.
5. Bajirilgan vazifa bo'yicha foto / video hisobot.
6. Nazorat savollariga javob yozing (qo'lda).

## №2.2. Amaliy mashg'ulot C tilida Arduino dasturlash asoslari .

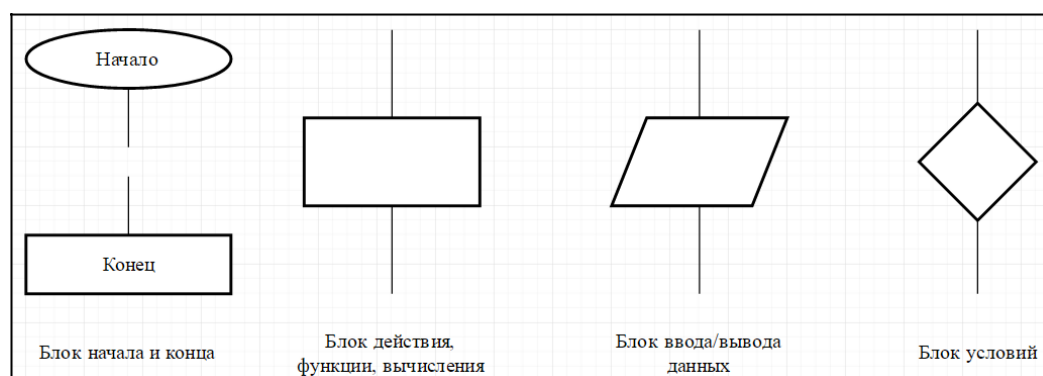
**Mashg'ulotning maqsadi :** Tanish Windows OS ostida Arduino tizimi bilan ishlash uchun dasturiy ta'minotni qanday o'rnatish kerak . Arduino tizimlarini C tilida dasturlash uchun zarur bo'lgan bilimlarni o'rganish .

### Qisqacha nazariya

*Arduino* ( *Arduino* ) apparatli hisoblash platformasi bo'lib, uning asosiy komponentlari kirish-chiqarish platasi va ishlab chiqish muhiti hisoblanadi.

*Algoritm* - bu har qanday muammolarni hal qilish imkonini beruvchi mantiqiy tartibda amal qiladigan ketma-ket harakatlar/buyruqlar to'plami .

Algoritmni grafik tasvirlash uchun blok diagrammalar deb ataladigan *diagrammalar* qo'llaniladi. *Oqim sxemalari* algoritmning grafik ko'rinishi bo'lib , geometrik shakllar to'plamidan iborat. Har bir rasmda ma'lum bir operatsiya yoki harakat tasvirlangan (2.1-rasm) .



2.1-rasm. Algoritmning grafik tasviri

Algoritmlar bir nechta turlarga bo'linadi: chiziqli, siklik , tarmoqli .

*Arduino IDE* ishlab chiqish muhiti - dasturlash tilining asosi  
В bu muhitda eng mashhur dasturlash tillaridan biridir .

### Arduino dasturining tuzilishi .

Arduino dasturining tuzilishi juda oddiy va minimal versiyada ikki qismdan iborat void setup () va void loop r ().

```
void setup ( ) {  
  // dastur ishga tushganda kod bir marta bajariladi  
}  
void loop ( ) {  
  // asosiy kod, tsiklda ishlaydi  
}
```

Setup ( ) funksiyasi kontroller yoqilganda yoki qayta o'rnatilganda bir marta bajariladi . Odatda, o'zgaruvchilar va registrnlarning dastlabki sozlamalari unda sodir bo'ladi. Funktsiya dasturda mavjud bo'lishi kerak, hatto unda hech narsa bo'lmasa ham.

O'rnatish ( ) tugallangandan so'ng , boshqaruv loop () funktsiyasiga o'tadi. U tanasida (jingalak qavslar orasiga) yozilgan buyruqlarni cheksiz tsiklda bajaradi. Aslida, bu buyruqlar boshqaruvchining barcha algoritmik harakatlarini bajaradi.

### **C tilining asl sintaksisi qoidalari.**

**; nuqta- vergul Ifodalar** o'zboshimchalik bilan ko'p bo'shliqlarni, qatorlarni o'z ichiga olishi **mumkin** . Ifodaning oxirining belgisi " nuqtali vergul" belgisidir.

$z = x + y$   $z = x + y ;$

**{ } jingalak qavslar** funktsiya yoki ifoda blokini belgilaydi. Masalan, o'rnatish ( ) va tsikl () funktsiyalarida .

**/\* ... \*/ izoh bloki** , yopilishi kerak.

**/\* bu izoh bloki \*/**

**// bitta qatorli izoh** , yopish shart emas, satr oxirigacha amal qiladi.

**// bu bitta sharh qatori**

### **O'zgaruvchilar va ma'lumotlar turlari.**

O'zgaruvchi - bu xotiradagi ma'lumotlarni saqlaydigan joy. Dastur oraliq hisoblash ma'lumotlarini saqlash uchun o'zgaruvchilardan foydalanadi. Hisob-kitoblar uchun turli formatdagi ma'lumotlar, turli bit chuqurliklari ishlatilishi mumkin, shuning uchun C tilidagi o'zgaruvchilar quyidagi turlarga ega.

Ma'lumotlar turi	Bit chuqurligi, bit	Raqam diapazoni
mantiqiy	8	rost , yolg'on
char	8	-128 ... 127
imzosiz char	8	0 ... 255
bayt	8	0 ... 255
int	16	-32768 ... 32767
imzosiz int	16	0 ... 65535
so'z	16	0 ... 65535
uzoq	32	-2147483648 ... 2147483647
imzosiz uzoq	32	0 ... 4294967295



qisqa	16	-32768 ... 32767
suzmoq	32	-3.4028235+38 ... 28235+38

Ma'lumotlar turlari talab qilinadigan hisoblash aniqligi, ma'lumotlar formatlari va boshqalar asosida tanlanadi. Masalan, 100 tagacha hisoblovchi hisoblagich uchun uzun turini tanlash shart emas. Bu ishlaydi, lekin operatsiya ko'proq ma'lumot va dastur xotirasini oladi, ko'proq vaqt talab etadi.

Chiziqli algoritmli dasturlar (2.3-jadval). Dasturlarning taqqoslanadigan qismlari qalin kursiv bilan yozilgan.

2.1-jadval - Turbo Paskal bilan solishtirganda Arduino IDE chiziqli dastur misollari

Arduino IDE	blok diagrammasi	TurboPaskal _
<pre>int a; int b; bekor sozlash () {   Serial.begin (9600); } bekor halqa ( ) {   <b>b=5;</b>   <b>a=b*10;</b>   Serial.println (a);   kechikish ( 5000); }</pre>	<pre> graph TD     Start([Начало]) --&gt; Init[a,b]     Init --&gt; Assign[b=5]     Assign --&gt; Calc[a=b*10]     Calc --&gt; Output[/Вывод a/]     Output --&gt; End([Конец])   </pre>	<pre>PROGRAM summasi ; _ CRT dan foydalanish; VAR a,b : butun son ; <b>BOSHLANISHI</b> CLRSCR; <b>b:= 5;</b> <b>a:= b*10;</b> WRITELN(a); <b>OXIRI.</b></pre>

### O'zgaruvchilarni e'lon qilish.

Ma'lumotlar turi ko'rsatilgan, keyin o'zgaruvchining nomi.

int x; // int tipidagi x nomli o'zgaruvchini e'lon qilish

float widthBox; // float tipidagi widthBox nomli o'zgaruvchini e'lon qilish  
oldin e'lon qilinishi kerak .

O'zgaruvchini dasturning istalgan qismida e'lon qilish mumkin, ammo bu dasturning qaysi bloklari undan foydalanishi mumkinligiga bog'liq. Bular. o'zgaruvchilar doiraga ega.

- void setup funksiyasidan oldin dastur boshida e'lon qilingan o'zgaruvchilar o'rnatish ( ) global hisoblanadi va dasturning istalgan joyida mavjud.

- Mahalliy o'zgaruvchilar funksiyalar yoki bloklar ichida e'lon qilinadi, masalan, for tsikli va faqat e'lon qilingan bloklar ichida ishlatilishi mumkin. Xuddi shu nomga ega, ammo turli xil qamrovli bir nechta o'zgaruvchilar mumkin.

```
int mode; // o'zgaruvchi barcha funktsiyalar uchun mavjud
void setup ( ) {
// bo'sh blok, boshlang'ich sozlamalar talab qilinmaydi
}
void loop() {
long count; // argumentlar countsoni faqat loop funksiyasida mavjud()
for (int i = 0; i < 10;) // argumentlar i faqat tsikl ichida mavjud
{
i++;
} }
```

Agar o'zgaruvchini e'lon qilsangiz, uning boshlang'ich qiymatini belgilashingiz mumkin (proinitializatsiya qilish).

```
int x = 0; // boshlang'ich qiymati bilan o'zgarmaydigan x e'lon 0
char d = 'a'; // D o'zgaruvchisi " a " belgisi kodiga teng bo'lgan boshlang'ich qiymat bilan e'lon qilinadi.
```

Turli xil ma'lumotlar turlari bilan arifmetik operatsiyalarda ma'lumotlar turlarini avtomatik ravishda o'zgartiradi. Lekin har doim aniq konvertatsiya qilish yaxshiroqdir.

```
int x; // o'zgaruvchan int
char y; // o'zgaruvchan char
int z; // o'zgaruvchan int
z = x + (int) y; // o'zgaruvchan y aniq int aylanadi
```

#### **Arifmetik operatsiyalar.**

=	tayinlash
+	qo'shish
-	olib tashlash
*	mahsulot
/	bo'linish
%	bo'linishning qolgan qismi

#### **Munosabatlar operatsiyalari.**

==	teng
!=	teng emas
<	kamroq
>	ko'proq
<=	kamroq yoki teng
>=	ko'proq yoki teng

#### **Mantiqiy operatsiyalar.**

&	mantiqiy va
//	mantiqiy yoki
!	mantiqiy emas

#### **Ko'rsatkichlar bo'yicha operatsiyalar.**

*	bilvosita manzil
&	o'zgaruvchan manzil olish

#### **Bit operatsiyalari.**

&	va
	yoki
^	istisno yoki
~	inversion
<<	chapga
o'tish >>	o'ng tomonga o'tish

#### **Aralash tayinlash operatsiyalari.**

++	+ 1 o'zgaruvchiga
--	- 1 o'zgaruvchiga
+=	qo'shish
-=	olib tashlash
*=	ko'paytirish
/=	bo'linish
%=	bo'linish qoldig'i
&=	bit va
/=	bit yoki

#### **Variantlarni tanlash, dasturni boshqarish.**

**Operator IF** qavs ichida holatini tekshiradi va qavs ichida keyingi ifoda yoki blok bajaradi, shart to'g'ri bo'lsa.

agar (x == 5) // x=5 bo'lsa, u holda z=0 amalga oshiriladi  
z=0;

agar (x >> 5) // x > 5 bo'lsa, u holda blok z=0, y=8 amalga oshiriladi;>  
{ z=0; y=8; }

**IF ... ELSE** siz ikki variantlari o'rtasida tanlov qilish imkonini beradi.

agar (x >> 5) // x > 5 bo'lsa, u holda blok z=0, y=8 amalga oshiriladi;>  
{

```

    z=0;
    y=8;
}
else // aks holda, bu birlik amalga oshiriladi
{
    z=0;
    y=0;
}

```

**Boshqa IF**-bir nechta tanlov qilish imkonini beradi

agar (x >> 5) // x > 5 bo'lsa, u holda blok z=0, y=8 amalga oshiriladi;>

```

{
    z=0;
    y=8;
}

```

boshqa if (x >> 20) // x > 20 bo'lsa, ushbu birlik amalga oshiriladi>

```

{
}

```

else // aks holda, bu birlik amalga oshiriladi

```

{
    z=0;
    y=0;
}

```

**Switch CASE**-bir nechta tanlov. Agar o'zgaruvchilar solishtirish imkonini beradi (masalan, bu x) bir necha sobit bilan (misol 5 va 10) va bir blok bajarish, qaysi o'zgaruvchilar sobit teng.

```

switch (x) {
case 5 :
    // kod x = 5 bo'lsa amalga oshiriladi
    break;
case 10 :
    // kod x = 10 bo'lsa amalga oshiriladi
    break;
default :
    // kodi если nehar qanday oldingi qiymati mos emas, agar amalga oshiriladi
    break;
}

```

**Davr uchun.** Dizayn sizga ma'lum miqdordagi yinelemeler bilan tsikllarni tashkil qilish imkonini beradi. Sinopsis shunday ko'rinadi:

```

for (tsikl boshlanishidan oldin harakat;
    tsiklni davom ettirish holati;
    har bir iteratsiya oxirida harakat ) {
    // tana kodi aylanishi
}

```

100 yinelemelerinden bir tsikl namunasi.

for (i=0; i < 100; i++) // boshlang'ich qiymati 0, end 99, qadam 1

```
{
    sum = sum + I;
}
```

**Davr WHILE.** Operator sizga qurilish bilan tsikllarni tashkil qilish imkonini beradi:

```
while (ifoda )
{
    // tana kodi aylanishi
}
```

Davr qavslar ichida ifoda haqiqiy qadar amalga oshiriladi. 10 yinelemelerinde tsiklning namunasi.

```
x = 0;
while ( x < 10 )
{
    // tana kodi aylanishi
    x++;
}
```

**DO WHILE**-chiqish sharti bilan aylanishi.

```
do
{
    // tana kodi aylanishi
} while (ifoda );
```

Tsikl ifoda haqiqiy bo'lsa-da amalga oshiriladi.

**Break**-ko'chadan chiqish operatori. Ko'chadan bajarilishini to'xtatish uchun ishlatiladi for,, while, do while.

```
x = 0;
while ( x < 10 )
{
    agar (z > 20) > break; // agar z > 20, keyin ko'chadan chiqish>
    // tana kodi aylanishi
    x++;
}
```

**GOTO**-shartsiz o'tish operatori.

goto metka1; // metka1 ga o'tish

.....

metka1:

**CONTINUE** - operatorlarning tsikl tanasining oxirigacha o'tishi.

```
x = 0;
while ( x < 10 )
{
    // tana kodi aylanishi
    agar (z > 20 ) > continue; // agar z > 20, keyin tananing boshiga qaytish>
    // tana kodi aylanishi
    x++;}
```

**Massivlar.**



Array xotira maydoni bo'lib, unda bir nechta o'zgaruvchilar ketma-ket saqlanadi. Shunday qilib, qator e'lon qilinadi.

```
int ages[10]; // bir qator 10 o'zgaruvchilar turi int
```

```
float weight[100]; // float turi 100 o'zgaruvchilar bir qatorfloat
```

Agar tillo e'lon boshlash mumkin bo'lsa:

```
int ages[10] = { 23, 54, 34, 24, 45, 56, 23, 23, 27, 28};
```

Quyidagicha o'zgaruvchilar, tillo murojaat:

```
x = ages[5]; // x qator 5 element qiymati beriladi.
```

```
ages[9] = 32; // 9 qator element belgilangan qiymati 32
```

Tillo elementlarini raqamlash har doim noldan.

### **Vazifalar.**

Vazifalar turli xil ma'lumotlar bilan bir xil ishlarni bajarishga imkon beradi.

Funktsiya mavjud:

- unga chaqirilgan ism;
- vajlari-ma'lumotlar, qaysi funktsiya hisoblash uchun foydalanadi;
- funktsiya tomonidan qaytarilgan ma'lumotlar turi.

Описывается пользовательская функция вне функций Setup() va loop () funktsiyalaridan tashqari maxsus funktsiya tavsiflanadi.

```
void setup() {
```

```
// dastur boshlanganda kod bir marta amalga oshiriladi
```

```
}
```

```
void loop() {
```

```
// asosiy kod, ko'chadan amalga oshiriladi
```

```
}
```

```
// functionName nomi bilan maxsus xususiyati e'lonfunctionName
```

```
type functionName( type argument1, type argument1, ... , type argument)
```

```
{
```

```
// tana vazifasi
```

```
return();
```

```
}
```

Funktsiyaning namunasi, ikki dalilning kvadratchalarining yig'indisini hisoblash.

```
int sumQwadr (int x, int y)
```

```
{
```

```
return( x* x + y*y);
```

```
}
```

Funktsiyani chaqirish shunday bo'ladi:

```
d= 2; b= 3;
```

```
z = sumQwadr(d,b); // v z D va b o'zgaruvchilari kvadratlarining yig'indisi bo'ladi
```

Vazifalar o'rnatilgan, maxsus, ulangan.

Juda qisqa, ammo bu ma'lumotlar Arduino tizimlari uchun C dasturlarini yozishni boshlash uchun etarli bo'lishi kerakАрдуино.

O'ylaymanki, agar siz ushbu mashg'ulotni birinchi marta o'qiyotgan bo'lsangiz, ushbu bo'limni o'tkazib yubormaslik va undan keyin nima qilish kerakligi haqida qaytib kelish kerak.

**C tilida dasturlarni ishlab chiqish bo'yicha tavsiyalar.**

Dasturlarning tashqi dizayni asosiy maqsadi dasturlarning okunabilirliğini yaxshilash, rasmiy xatolar sonini kamaytirishdir. Shuning uchun, bu maqsadga erishish uchun barcha tavsiyalarni xavfsiz tarzda buzishingiz mumkin.

### **C tilidagi ismlar.**

Ma'lumotlar turlarini ifodalovchi ismlar aralash registrda yozilishi kerak. Ismning birinchi harfi katta (katta harf) bo'lishi kerak.

Signal, TimeCount

O'zgaruvchilar aralash registrda nomlar bilan yozilishi kerak, birinchi harf kichik harf (kichik harf).

signal, timeCount

Sobit yuqori ro'yxatdan yozilgan bo'lishi kerak. Ajratuvchi sifatida pastki chiziq.

MAX\_TEMP, RED

Usullari va vazifalari aralash registrda yozilgan fe'llar, kichik harfdagi birinchi harf deb nomlanishi kerak.

getTime, setTime

**Vazifa:** Arduino IDE rivojlanish muhitini o'rnating, ishga tushirishdan oldin jihozni sozlang.

### **Hisobotning mazmuni.**

Hisobotda quyidagilar bo'lishi kerak:

1. Amaliyot nomlari va ishning maqsadi.
3. Tadqiqotda ishlatiladigan apparatning xususiyatlari.
4. Kompilyatsiya kodi va ekran tasvirlarini yozish.
5. Bajarilgan topshiriq bo'yicha foto / video hisobot.
6. Test savollariga javob yozing (qo'lda)

### **Nazorat savollari**

1. Arduino IDE nima?
2. Algoritm nima?
3. Qaysi algoritmlarni bilasiz?
4. Oqim sxemasi tushunchasiga ta'rif berasizmi?
5. Nima uchun "for" deb ataladi?
6. Turbo Paskal dasturlash Tili " kursidan qanday ma'lumotlarni Turbo Pascalslaysiz?
7. - Har bir inson uchun foydali ma'lumotlar
8. Nima uchun "void loop" deb ataladi?
9. Nima uchun "qachon" deb ataladi?
10. Nima uchun "while" deb ataladi?

