

**ARDUINO DASTURIDA DASTURLASH ALGORITMINI
TUZISHDA RAQAMLI VA ANALOG CHIQISH**

Turayeva Shaxlo Norboy qizi

Nizomiy nomidagi Toshkent Davlat Pedagogika Universiteti dotsenti

To'rasheva Maftunabonu Muxammadtoir qizi

Nizomiy nomidagi Toshkent Davlat Pedagogika Universiteti magistri

Annaotatsiya: "Raqamli" va "analog" chiqishlar — bu ikki turdagi signal chiqish formatlaridir, ularning har biri o'ziga xos xususiyatlarga ega. Raqamli chiqish — bu signallarni faqat ikkita qiymatga (0 yoki 1) asoslangan holda uzatadigan chiqish turidir. **Analog chiqish** (Analog Output) nomidan ko'rinib turibdiki, analog chiqish mikrokontroller pinida kuchlanish darajasini hosil qiladi. Bir ma'noda, bu analog kirish bilan taqqoslaganda, bu mutlaqo teskari jarayon, chunki bu safar raqamli qiymat yana analog kuchlanish signaliga aylantirilmoqda.

Аннотация: "Цифровой" и "аналоговый" выходы - это два типа сигнальных выходных форматов, каждый из которых имеет свои особенности. Цифровой выход - это тип выхода, который передает сигналы только на основе двух значений (0 или 1). Как следует из названия аналоговый выход (Analog Output), аналоговый выход генерирует уровень напряжения на pin микроконтроллера. В некотором смысле это совершенно противоположный процесс по сравнению с аналоговым входом, поскольку на этот раз цифровое значение снова преобразуется в сигнал аналогового напряжения.

Annotation: "Digital" and "analog" outputs are two types of signal output formats, each of which has its own characteristics. A digital output is a type of output that transmits signals based on only two values (0 or 1). As can be seen from the name Analog Output, the analog output generates a voltage level at the microcontroller pin. In a sense, this is a completely opposite process compared to the

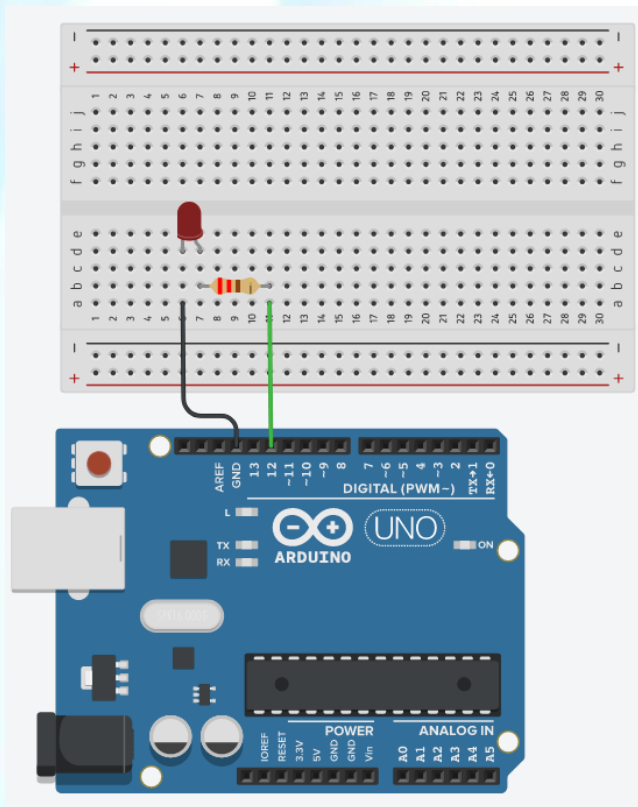


analog input, since this time the digital value is again converted into an analog voltage signal.

Kalit soʻzlar: *Raqamli chiqish (Digital Output), Analog chiqish (Analog Output), Arduino, dasturlash, maket platasi, svetodiod.*

Kirish. Raqamli chiqish (Digital Output)da ishlaydigan qurilmalarga qoʻyidagilar misol keltirishimiz mumkin. Arduino raqamli chiqishi yordamida LEDni (yorugʻlik chiqaradigan diod) qanday miltillashni bilib olaylik. Biz LEDni Arduino Unoga ulaymiz va LEDni yoqish va oʻchirish uchun oddiy dastur tuzamiz. Bu erda Tinkercad sxemalarida siz namunaviy sxemani oʻrganishingiz va uning yonida oʻzingizning algoritmingizni qurishingiz mumkin. LED miltillashini tomosha qilish uchun "simulyatsiyani boshlash" tugmasini bosing. Sxemalaringizni sinab koʻrish uchun simulyatordan istalgan vaqtda foydalanishingiz mumkin.

Asosiy qisim: Komponentlar paneliga qaytib, Arduino Uno platasini topib ish stoliga quyamiz. Simni komponentga yoki pinga ulash uchun bir marta bosiladi va boshqa uchini ulash uchun yana bosiladi. Rezistorni LEDning ikkala tomoniga ulab olamiz. Agar siz rezistoringizni LED anodiga ulagan boʻlsangiz (ijobiy, uzunroq), rezistorning boshqa oyoqini Arduino raqamli pin 12 ga ulang. Agar siz rezistoringizni LED katodiga (salbiy, qisqaroq oyoq) ulagan boʻlsangiz, rezistorning boshqa oyoqini Arduinoning GND piniga ulang.



1-rasm. Led va Arduinoning ulanish sxemasi

Eng oddiy narsadan boshlaylik. Aytaylik, Arduino dasturi boshlanganda LEDni yoqilishida ushbu kod bizga so‘zlashni va raqamli pinlar qanday ishlashini tushunishga yordam beradi.

```
sketch_dec19a | Arduino 1.8.15
Файл Правка Скетч Инструменты Помощь

sketch_dec19a.g
1 #define LED_PIN 12
2
3 void setup()
4 {
5   pinMode(LED_PIN, OUTPUT);
6   digitalWrite(LED_PIN, HIGH);
7 }
8
9 void loop() {}
```

Ushbu kod juda qisqa, bu yerda 12 uyachaga ulangan led yonadi. Keling tahlil qilaylik.

#define LED_PIN 12

Birinchidan, bu eng yaxshi amaliyot, biz pin raqamini aniqlaymiz. Shunday qilib, biz ushbu raqamli pin-kodni ishlatishimiz kerak bo‘lganda, biz kodli raqam



o'rniga LED_PIN yozishimiz mumkin. Kelajakda, agar siz boshqa raqamli pin-kodni ishlatishingiz kerak bo'lsa (masalan 11), unda siz bu erda raqamni o'zgartirishingiz kerak va u sizning dasturingizning hamma joyida yangilanadi.

void setup()

{ pinMode(LED_PIN, OUTPUT);

Dastur boshida bir marta bajariladigan **void setup()** funksiyasiga kiramiz. Raqamli pin-kodni ishlatishdan oldin biz rejimni o'rnatishimiz kerak. Asosan bizda 2 rejim mavjud: chiqish (agar siz komponentni boshqarishni xohlasangiz) yoki kiritish (agar siz komponentdan ba'zi ma'lumotlarni o'qishni istasangiz). Bu erda biz LEDni boshqarishni xohlaymiz, shuning uchun biz chiqishni tanlaymiz. Buning uchun biz 2 parametrli **pinMode()** funksiyasidan foydalanamiz:

Pin raqami: bu erda biz ilgari aniqlagan **LED_PIN** -dan foydalanamiz.

Rejim: chiqish rejimi uchun biz **OUTPUT** (barchasi katta harflarda) dan foydalanishimiz kerak.

Ushbu algoritmi bajarilgandan so'ng, raqamli pin 12 chiqish sifatida o'rnatiladi va biz LEDni boshqarishimiz mumkin

digitalWrite(LED_PIN, HIGH); }

Endi LEDni boshqarish juda oddiy. Biz 2 parametrli **digitalWrite()** funksiyasidan foydalanishimiz kerak:

Pin raqami: biz belgilangan **LED_PIN** dan foydalanamiz.

Bu yerda 2 ta variant bor. LEDni yoqish uchun **HIGH** va uni o'chirish uchun **LOW**.

Shunday qilib, ushbu amaliyotdan keyin LED yoqiladi.

void loop() {}

Void setup () funksiyasi tugagandan so'ng, biz **void loop()** ga kiramiz va Arduinoni o'chirmaguningizcha, bu funksiya qayta-qayta bajariladi.

Bu erda biz **void loop()** da hech narsa yozmadik, chunki biz faqat LEDni yoqishni xohlaymiz va tamom.

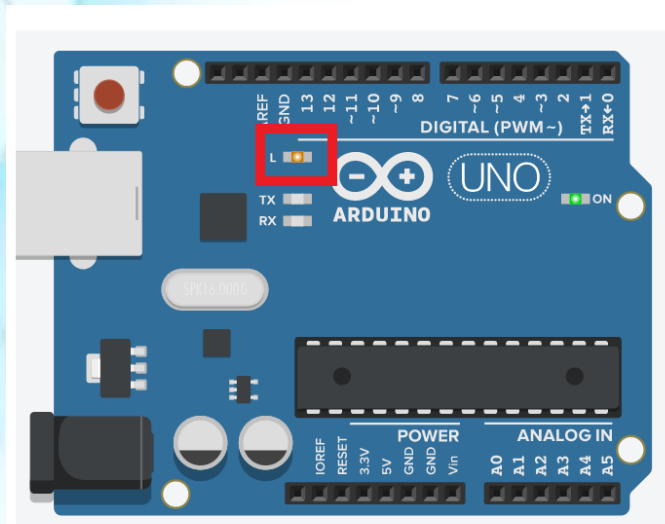
Agar siz sxemani qurishni xohlamasangiz yoki undan ham oddiyroq narsa bilan sinab ko'rmoqchi bo'lsangiz, Arduinoda raqamli pin 13 da **lehimlangan** o'rnatilgan LEDdan foydalanishingiz mumkin.

Siz ushbu LED uchun oldindan belgilangan LED_BUILTIN - dan foydalanishingiz mumkin.

```
sketch_dec19a | Arduino 1.8.15
Файл Правка Скетч Инструменты Помощь

sketch_dec19a.g
1 void setup()
2 {
3   pinMode(LED_BUILTIN, OUTPUT);
4   digitalWrite(LED_BUILTIN, HIGH);
5 }
6
7 void loop() {}
```

Va siz o'rnatilgan LED yoqilganligini ko'rasiz. Eslatma: LEDning joylashuvi Arduino platangiz turiga qarab farq qilishi mumkin.



2-rasm. Arduinoda o'rnatilgan *lehimlangan* led chiroqchasining joylashuvi. Endi bizda LEDni sozlash sxemasi va kodi bor, keling, biroz qiziqroq narsa qilaylik.

Keling, LEDni miltillatamiz,

1. LED yonadi biroz muddat,
2. LEDni o'chiring biroz muddatga,



Va 1 ga qayting, bu holat biz arduinoni o'chirmagunimizgacha davriy davom etsin

Buning uchun kod (raqamli pin 12 ga ulangan LED yordamida).

```
sketch_dec19a.g
1 #define LED_PIN 12
2
3 void setup()
4 {
5   pinMode(LED_PIN, OUTPUT);
6 }
7
8 void loop()
9 {
10  digitalWrite(LED_PIN, HIGH);
11  delay(500);
12  digitalWrite(LED_PIN, LOW);
13  delay(500);
14 }
```

Keling, ushbu kodni satr bo'yicha ajratamiz.

#define LED_PIN 12

Bu avvalgidek. Biz aniq kod yaratamiz, shunda biz pin raqamini keyinchalik kodda pin uyachasini raqamlamasdan ishlatishimiz mumkin.

```
void setup()
{
pinMode(LED_PIN, OUTPUT);
}
```

void setup()da biz pin chiqishi uchun rejimni ishga tushiramiz. E'tibor bering!!! bu biz bu erda qiladigan yagona narsa.

```
void loop()
{
digitalWrite(LED_PIN, HIGH);
delay(500);
digitalWrite(LED_PIN, LOW);
delay(500);
}
```

void loop(), biz blink mexanizmini band. Avval biz LEDni **digitalWrite ()** va **HIGH** bilan yoqamiz, keyin **delay()** yordamida kutamiz. **delay()** funksiyasi dasturni ma'lum vaqt davomida (millisekundlarda) bloklaydi. Shunday qilib, bu erda biz dasturni 500 millisekund yoki 0,5 soniya davomida bloklaymiz. Shundan so'ng, biz LEDni **digitalWrite()** va **LOW** bilan o'chiramiz va **delay()** yordamida yana 0,5 soniya kutamiz. Shu tarzda, biz "LED yoqilgan" + "LED o'chirilgan"ning to'liq sikliga egamiz. Void loop () dan chiqqanimizdan so'ng, u yana chaqiriladi va shuning uchun LED yana yoqiladi va hokazo va hokazo. Ushbu miltillovchi misol ehtimol, eng keng tarqalgan misollardan biri, ammo Arduino bilan ishlashni boshlaganingizda juda foydali. Agar siz sxema va kodni to'g'ri tushunsangiz, siz allaqachon katta yutuqlarga erishgansiz.

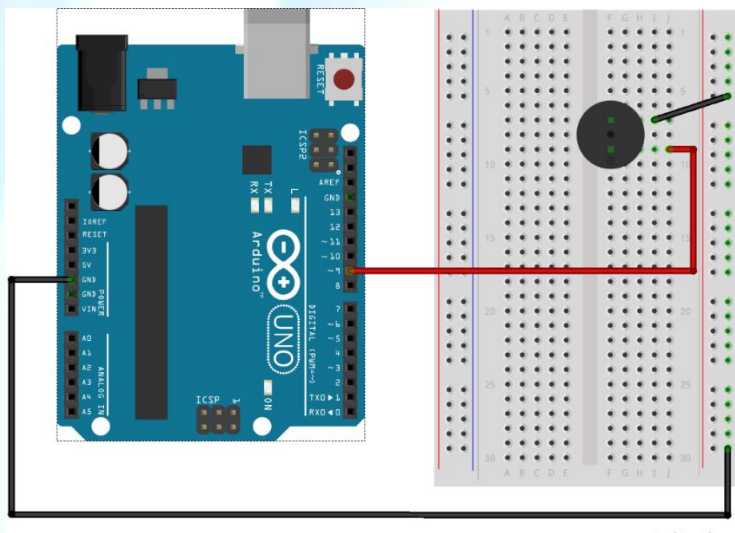
Arduino uno yordamida zummer modulidan (piezo karnay) foydalaning

Ushbu loyiha xavfsizlik uchun zummerlar yordamida signal va signallarga asoslangan.



3-rasm. Zummer.

Arduino bilan zummer yoki piezo karnayidan qanday foydalanishni o'rganasiz. Zummerlarni signal qurilmalarida, kompyuterlarda, taymerlarda va sichqonchani bosish yoki tugmachani bosish kabi foydalanuvchi kiritishini tasdiqlashda topish mumkin. Shuningdek, siz **tone()** va **noTone()** funksiyasidan qanday foydalanishni o'rganasiz.



4-rasm. Arduino bilan zummerning ulanish sxemasi.

Ulanishlar juda oson, yuqoridagi rasmga qarang maket platasi elektron sxemasi.

Qanday ishlaydi? Bu oddiy, ohang(buzzer, 1000) pin uchun 1khz tovush signal yuboradi 9, kechikish(1000) bir soniya va noTone uchun dasturini to'xtatib turish (buzzer) signal ovozi to'xtaydi. loop () muntazamligi bu ishni qayta-qayta qisqa signal tovushini chiqaradi (siz ohang (pin, chastota, davomiylik) funksiyasidan ham foydalanishingiz mumkin)

```

sketch_dec20a | Arduino 1.8.15
Файл Правка Скетч Инструменты Помощь

sketch_dec20a
1 const int zummer = 9; //arduino pin uchun zummer 9
2 void setup(){
3   pinMode(zummer, OUTPUT); // Chiqish sifatida zummer pin 9-ni o'rnatish
4 }
5 void loop(){
6   tone(zummer, 1000); // 1khz tovush signalini yuboring...
7   delay(1000);         // ...1 soniya uchun
8   noTone(zummer);      // Ovozni to'xtatish...
9   delay(1000);         // ...1 sekund uchun
10 }
  
```

Natjalar: Sensor birligi oddiy sensor yoki ba'zi qo'shimcha komponentlar bo'lishi mumkin. Agar sezgir qurilma mikrokontroller bilan ishlasa, quyidagi bloklar talab qilinadi. Raqamli chiqish yuqori aniqlik, chidamlilik va uzoq masofaga uzatishda afzalroq bo'lsa, **analog chiqish** ko'proq o'ziga xos va oddiy ilovalar uchun mos keladi, ammo uzoq masofaga uzatishda buzilishlar yoki shovqinlar ta'sir qilishi mumkin. Arduino-da kodning bir qismining



davomiyligini qanday hisoblash bilan tanishib chiqamiz. Dasturlashda vaqt/davomiylik tushunchasi juda muhim, ayniqsa Arduino kabi oʻrnatilgan tizimlarda. Juda koʻp vaqt talab qiladigan kodni yozish va uni bilmaslik sizning umumiy ilovangizda jiddiy muammolarga olib kelishi mumkin. Ushbu rejani oʻrganishdan maqsad digitalWrite() funksiyasining davomiyligini oʻlchash boʻladi, bundan tashqari Arduino dasturlarida kerakli kodni oʻlchash uchun foydalanishingiz mumkin boʻlgan shablonni tuzamiz.

FOYDALANILGAN ADABIYOTLAR

1. X.N.Nazarov —Robototexnika asoslaril. TDTU., Toshkent:, 2005 y.
2. “Робототехника для детей и родителей” С.А. Филиппов, Санкт-Петербург “Наука” 2010. - 195 с. 12
3. Бокселл Дж. Б78 Изучаем Arduino. 65 проектов своими руками. — СПб.: Питер, 2017. . — 400 с.
4. С. Монк Програмируем Arduino. Профессиональная работа со скетчами . — СПб.: Питер, 2017.

Axborot manbaalari

1. <http://www.robotics.uc.edu>
2. <http://www.robotics.utexas.edu/rrg>
3. <http://www.lib.tpu.ru/fulltext2/m/2012/m85.pdf>