

# ARDUINO TARAQQIYOTI

Ma'ruza #7

## **ARDUINODA USB VA KETMA-KET KOMMUNIKATSIYA (ALOQA)**

### **Reja:**

1. Arduining ketma-ket kommunikatsiyasi imkoniyatlari bilan tanishish
2. Arduinoni tinglash
  - print komandasidan foydalanish
  - Maxsus belgilarni ishlatish
  - Ma'lumotlar turlarini almashtirish
3. Arduino bilan muloqot qilish
  - Kompyuter yoki boshqa ket-ket kommunikatsiyasi mavjud qurilmalardan ma'lumot olish

Foydalanilgan adabiyotlar

Boshlash uchun siz ketma - ket va USB o'rtasidagi farqni tushunishingiz kerak. Yoshingizga qarab, siz ketma-ket (yoki texnik jihatdan RS-232) portlarni eslay olmasligingiz mumkin, chunki ular asosan USB bilan almashtirilgan. Quyidagi rasmda standart ketma-ket port nimaga o'xshashligi ko'rsatilgan.

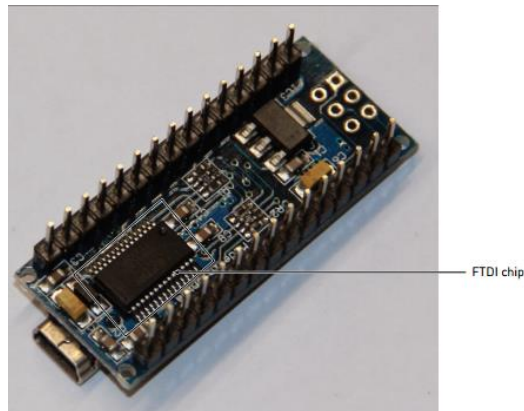


Asl Arduino platalari ketma-ket port bilan jihozlangan bo'lib, siz ularni kompyuterga 9-pinli ketma-ket kabel orqali ulaysiz. Hozirgi kunda kamdan-kam kompyuterlarda bunday portlar mavjud, garchi siz USB portlaridan DB-9 (9-pinli ulagich turi) ketma-ket portlarini yaratish uchun adapterlardan foydalanish mumkin. ATmega328P kabi mikrokontrollerlarda bitta apparat ketma-ket port mavjud. U 0 va 1 raqamli pinlarda kirish mumkin bo'lgan uzatish (TX) va qabul qilish (RX) pinlarini o'z ichiga oladi, "Arduinoda miltillash" darsida aytilganidek, Arduino yuklash qurilmasi bilan jihozlangan. Bu sizga ushbu ketma -ket interfeys orqali dasturlash imkonini beradi. Buni yengillashtirish uchun bu pinlar "multipleksli" (ya'ni ular bir nechta funktsiyaga ulangan); ular bilvosita USB kabelingizning uzatish va qabul qilish liniyalariga ulanadi. Biroq, ketma -ket va USB to'g'ridan -to'g'ri mos kelmaydi, shuning uchun ikkalasini bog'lash uchun ikkita usuldan biri ishlatiladi. Birinchi variant - bu ikkinchisining konvertatsiyasini osonlashtirish uchun ikkilamchi integral mikrosxemadan (IC) foydalanish (Arduino platasida yoki undan tashqarida). Bu Uno-da mavjud bo'lgan interfeys turi, bu yerda vositachi IC USB-ketma-ket aloqani osonlashtiradi. Ikkinchi variant - USB boshqaruvi o'rnatilgan mikrokontrollerni tanlash (masalan, Arduino Leonardning 32U4 MCU).

### **Arduino platalarida ichki yoki tashqi FTDI USB-dan ketma-ket konvertorli**

Yuqorida aytib o'tilganidek, ko'plab Arduino platalari (va Arduino klonlari) USB-ni ketma-ket o'zgartirishni osonlashtirish uchun ikkilamchi integral sxemadan foydalanadilar. "FTDI" chipi faqat bitta funktsiyaga ega bo'lgan mashhur chipdir: ketma - ket va USBga aylantirish. Agar sizning kompyuteringiz FTDI chipiga ulangan bo'lsa, u

sizning kompyuteringizda "virtual ketma -ket port" sifatida paydo bo'ladi, unga siz xuddi kompyuteringizga ulangan DB9 porti kabi kirishingiz mumkin. Quyidagi rasmda o'rnatilgan FTDI chipidan foydalaniladigan Arduino Nano-ning pastki qismi ko'rsatilgan.



Agar siz ishlash uchun USB orqali kompyuterga ulanishni talab qilmaydigan loyihani ishlab chiqayotgan bo'lsangiz, olinadigan FTDI dasturchisi bilan platadan foydalanish juda yaxshi. Agar siz bir nechta qurilmalar yasangiz, bu xarajatlarni kamaytiradi va tayyor mahsulotning umumiy hajmini kamaytiradi.

Quyida FTDI chipini ishlatadigan Arduino platalari ro'yxati keltirilgan. E'tibor bering, yangi Arduino platalarida endi FTDI chipi ishlatilmaydi, shuning uchun ularning ko'pchiligi to'xtatildi. Ular to'liqligicha bu yerda keltirilgan:

- ■ Arduino Nano
- ■ Arduino Ekstremal
- ■ Arduino NG
- ■ Arduino Diecimila
- ■ Arduino Duemilanove
- ■ Arduino Mega (original)

Quyidagi tashqi FTDI programmatoridan foydalanadigan Arduino platalari ro'yxati:

- ■ Arduino Pro
- ■ Arduino Pro Mini
- ■ LilyPad Arduino

- ■ Arduino Fio
- ■ Arduino Mini
- ■ Arduino Ethernet

Arduino platalari Arduino Uno USB-dan seriyali konvertatsiyani boshqarish uchun FTDI chipidan boshqa integratsiyalashgan sxemadan foydalanishni joriy qilgan birinchi plataidir. Funktsional jihatdan, u bir xil texnik farqlar bilan bir xil ishlaydi. Pastdagi rasmda Arduino Unoning 8U2 ketma-ket konvertori ko'rsatilgan (hozirda yangi versiyalarda 16U2). Farqlarning qisqacha ro'yxati quyida keltirilgan :

- ■ Birinchidan, Windows operatsion tizimida USB-dan seriyali konvertatsiya qilishning yangi yechimiga ega bo'lgan platalar maxsus drayverini o'rnatishni talab qiladi. Ushbu drayver Arduino IDE bilan birga yuklab olinadi. (OS X yoki Linux uchun drayver kerak emas.)

- ■ Ikkinchidan, konvertatsiya qilish uchun ushbu ikkinchi mikrokontrollerdan (MCU) foydalanish, Arduinoning maxsus ishlab chiqaruvchi identifikatori va mahsulot identifikatori plata ulanganda asosiy kompyuterga xabar berishga imkon berdi. FTDI-ga asoslangan plata kompyuterga ulanganida, u oddiy USB-seriyali qurilma sifatida namoyon bo'ldi. Arduino FTDI bo'lmagan konvertor IC (ATMega 8U2, erta Arduino Unos bo'lsa, hozir 16U2) ishlatilganda, u kompyuterga Arduino sifatida aniqlanadi.



Quyida USBdan seriyali konvertatsiya qilish uchun ikkinchi darajali MCU ni ishlatadigan Arduino platalari ro'yxati keltirilgan:

- ■ Arduino Uno

- ■ Arduino Mega 2560
- ■ Arduino Mega ADK (2560 ga asoslangan)
- ■ Arduino Due (shuningdek dasturlashtirilishi mumkin) to'g'ridan - to'g'ri)

### **Yagona USB qo'llanadigan mikrokontrollerga ega Arduino platalari**

Arduino Leonardo faqat bitta chipga ega bo'lib, u foydalanuvchi tomonidan dasturlashtiriladigan MCU va USB interfeysi vazifasini bajaradi. Leonardo (va shunga o'xshash Arduino platalari) ATmega 32U4 mikrokontrolleridan foydalanadi, bu to'g'ridan - to'g'ri USB aloqasi uchun o'rnatilgan chip. Bu xususiyat bir qancha yangi xususiyatlar va yaxshilanishlarga olib keladi.

Birinchidan, plata narxi kamayadi, chunki uning qismlari kamroq bo'ladi va platalarni ishlab chiqarish uchun zavod dasturlash bosqichidan kam talab qilinadi. Ikkinchidan, platani ketma -ket portdan (klaviatura, sichqoncha yoki joystik kabi) boshqa USB qurilmalarini bog'lash uchun osonroq ishlatish mumkin. Uchinchidan, ATMegadagi bitta oddiy USART porti USB -dasturchi bilan ko'paytirilmaydi, shuning uchun asosiy kompyuter va ikkilamchi ketma -ket qurilmalar bilan aloqa (masalan, GPS birligi) bir vaqtning o'zida sodir bo'lishi mumkin.

Quyida bitta USB-quvvatlaydigan MCU ishlatadigan Arduino platalari ro'yxati keltirilgan:

- ■ Arduino Due (ikkinchi darajali MCU orqali ham dasturlash mumkin)
- ■ LilyPad Arduino USB
- ■ Arduino Esplora
- ■ Arduino Leonardo
- ■ Arduino Micro

### **USB-xost imkoniyatlariga ega Arduino platalari**

Ba'zi Arduino platalari USB qurilmalariga xost sifatida ulanishi mumkin, bu sizga an'anaviy USB qurilmalarini (klaviatura, sichqonlar, Android telefonlari) Arduino-ga ulash imkonini beradi. Tabiiyki, siz ulayotgan qurilmani qo'llab -quvvatlash uchun tegishli drayverlar bo'lishi kerak. Masalan, siz veb - kamerani Arduino Duega ulay olmaysiz. Due hozirda klaviatura yoki sichqonchani boshqarish uchun USB portiga ulashga imkon beradigan USB xost-klassini qo'llab-quvvatlaydi. Arduino Mega ADK Arduino va Android qurilmasi o'rtasidagi aloqani osonlashtirish uchun Android Open Accessory Protocol (AOA) dan foydalanadi. Bu birinchi navbatda Android qurilmasida ishlaydigan ilovadan Arduino I/O ni boshqarish uchun ishlatiladi. USB xost imkoniyatlariga ega bo'lgan ikkita Arduino platasi-Arduino Due va Arduino Mega ADK (Mega 2560 asosida).

## **ARDUINO ni tinglash**

Arduinoda qilishingiz mumkin bo'lgan eng asosiy ketma -ket funktsiya - kompyuterning ketma -ket terminaliga chop etish. Bu darsda siz funktsionallikni chuqurroq o'rganasiz, keyinroq siz terminalga chop etish o'rniga, siz yuborgan ma'lumotlarga javob beradigan ba'zi ish stoli ilovalarini yaratishingiz mumkin. Bu jarayon barcha Arduinolar uchun bir xil.

## **Print funksiyasidan foydalanish**

Terminalga ma'lumotlarni chop etish uchun siz faqat uchta funksiyadan foydalanishingiz kerak:

- ■ `Serial.begin(baud_rate)`
- ■ `Serial.print("Xabar")`
- ■ `Serial.println("Xabar")`

bu yerda `baud_rate` va " Xabar " bu siz ko'rsatgan o'zgaruvchilar.

`Serial.begin ()` dastur boshida bir marta chaqirilishi kerak, muloqot o'rnatish uchun. Buni qilganingizdan so'ng , ketma -ket portga ma'lumotlarni yozish uchun `Serial.print ()` va `Serial.println()` funktsiyalaridan erkin foydalanishingiz mumkin. Ikkalasi orasidagi yagona farq shundaki, `Serial.println()` satr oxiriga karetk qaytishini qo'shadi (shunday qilib keyingi bosilgan narsa keyingi satrda paydo bo'ladi). Ushbu funktsiyani sinab ko'rish uchun quyidagi rasmda ko'rsatilganidek, Arduinodagi A0 piniga ulangan potentsiometrli oddiy sxemani sim bilan ulang.

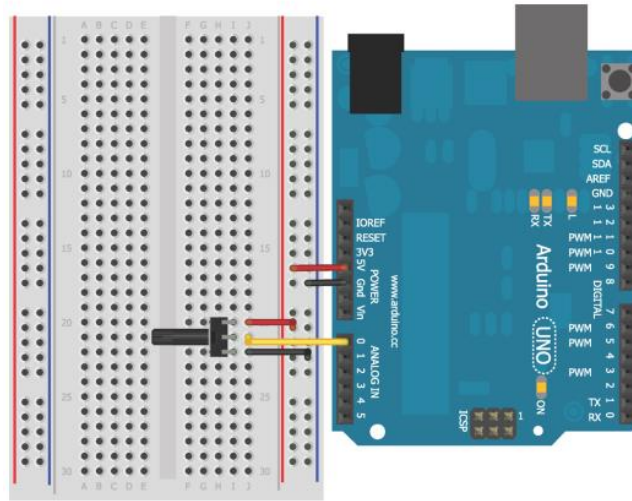


Image created with Fritzing.

Potensiometrni ulaganingizdan so'ng, quyida ko'rsatilgan oddiy dasturni yuklang, u potensiometrning qiymatini o'qiydi va uni ham haqiqiy, ham foiz qiymatini chop etadi.

Potensiometrli ketma-ket chop etish dasturi

```
const int POT=0; //Potensiometr analog pin Oda
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  int val = analogRead(POT); //Potensiometerni o'qish
  int per = map(val, 0, 1023, 0, 100);
  Serial.print("Analog Reading: ");
  Serial.print(val);
  Serial.print(" Percentage: ");
  Serial.print(per);
  Serial.println("%");
  delay(1000); //1 sekund kutish va yana qayta takrorlsh
}
```

Serial.print() va Serial.println() komandalarining kombinatsiyasidan foydalangan holda, bu kod sekundiga bir marta haqiqiy va foiz qiymatlarini chop etadi. E'tibor bering, biz faqat oxirgi satrda Serial.println () dan foydalangan holda, har bir oldingi uzatish bir qatorda qoladi.

Arduino IDEdan ketma - ket monitorni oching va Arduino eskizida ko'rsatilgan qiymatga mos keladigan tezlik 9600 ga o'rnatilganligiga ishonch hosil qiling. Potensiyometrni aylantirganda qiymatlar soniyada bir marta chop etilishini ko'rishingiz kerak.

## **MAXSUS BELGILARDAN FOYDALANISH**

Siz chop etayotgan seriyali ma'lumotlar formatini o'zgartirish imkonini beruvchi turli "maxsus belgilar"ni ketma-ketlik orqali ham uzatishingiz mumkin. Siz bu maxsus belgilarni chiziq chizig'idan (\) keyin buyruq belgisi bilan ko'rsatasiz. Ushbu maxsus belgilarning xilma-xilligi bor, lekin ikkita eng katta qiziqish yorliq va yangi qator belgilardir. Yorliq belgisini kiritish uchun satrga \t qo'shing. Yangi qatorni kiritish uchun qatorga \n qo'shing. Agar siz yangi satrni Serial.println () funksiyasi o'rniga oxiriga emas, balki satr boshiga kiritishni xohlasangiz, bu ayniqsa foydali bo'ladi. Agar biror sababga ko'ra, siz aslida \n yoki \t ni satrda chop etmoqchi bo'lsangiz, buni mos ravishda \\n yoki \\t bosib chiqarish mumkin. Quyidagi dastur bu ma'lumotlarni jadval formatida ko'rsatish uchun ushbu maxsus belgilardan foydalanish uchun oldingi kodning o'zgarishi.

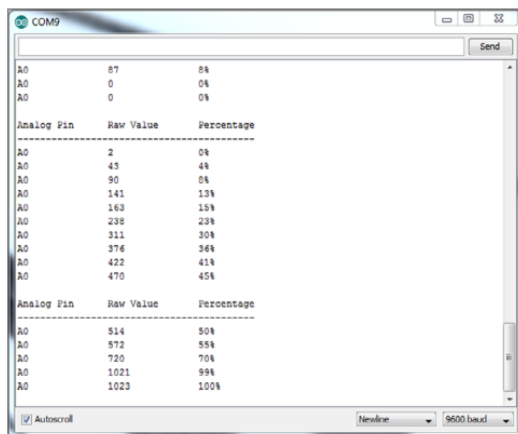
```
const int POT=0;
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  Serial.println("\nAnalog Pin\tRaw Value\tPercentage");
  Serial.println("-----");
  for (int i = 0; i < 10; i++)
  {
    int val = analogRead(POT);
    int per = map(val, 0, 1023, 0, 100);
    Serial.print("A0\t");
    Serial.print(val);
    Serial.print("\t");
    Serial.print(per);
```

```

Serial.println("%");
delay(1000);
}
}

```

Potentsiometrni aylantirganda, bu dasturning chiqishi quyidagi rasmda ko'rsatilgan natijalarga o'xshash bo'lishi kerak.



## MA'LUMOTLAR TURINI O'ZGARTIRISH

Serial.print () va Serial.println () Agar kutuyapmiz formatda ma'lumotlarni olib chop kelganda vazifalari juda aqlli. Biroq, sizda ma'lumotlarni turli formatlarda, jumladan, o'n oltilik, sakkizlik va ikkilik formatlarda chiqarish imkoniyatlari mavjud. O'nli kodli ASCII standart formatdir. Serial.print () va Serial.println () funksiyalari ixtiyoriy ikkinchi dalillar deb bosma format bor. Pastdagi jadvalda bir xil ma'lumotlarni turli formatlarda chop etishingiz va ularning seriyali terminalda qanday ko'rinishi misollari keltirilgan.

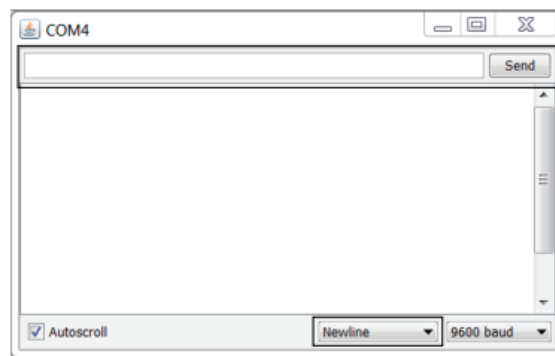
### Ketma-ket ma'lumotlar turi variantlari

Ma'lumot turi	Kodga misollar	Ketma-ket chiqishi
O'nlik	Serial.println(23);	23
O'n oltilik	Serial.println(23, HEX);	17

Sakkizlik	Serial.println(23, OCT);	27
Ikkilik	Serial.println(23, BIN);	00010111

## ARDUINO bilan gaplashish

Arduino bilan muloqot faqat bitta yo'nalishda ketayotganidan nima foyda? Endi siz Arduino ma'lumotlarini kompyuteringizga qanday yuborishini tushunganingizdan so'ng, keling, kompyuterdan Arduinoga buyruqlarni qanday yuborishni muhokama qilaylik. Ehtimol, siz allaqachon Arduino IDE seriyali monitorining yuqori qismida matn kiritish maydoni, pastda esa ochiladigan menyu borligini payqagandirsiz. Quyidagi rasmda ikkalasi ham ko'rsatilgan.



Birinchidan, ochiladigan ro'yxat Newline-ga o'rnatilganligiga ishonch hosil qiling. Ochiladigan menyu sizning buyruqlaringizni Arduinoga yuborganingizda oxiriga nima qo'shilishini aniqlaydi. Keyingi bo'limlardagi misollar sizda "yangi qator" tanlangan deb taxmin qilinadi, u ketma -ket monitor oynasining yuqori qismidagi matn kiritish maydonidan yuborilgan har bir narsaning oxiriga \n qo'shiladi.

Boshqa terminal dasturlaridan farqli o'laroq, Arduino IDE ketma - ket monitor Enter tugmachasini yoki Yuborish tugmachasini bosganingizda bir vaqtning o'zida butun buyruqlar qatorini yuboradi (siz ko'rsatgan tezlikda).

## KOMPYUTERDAN VA BOSHQA SERIAL QURILMALARDAN MALUMOTNI O'QISH

Arduino IDE ketma - ket monitorini qo'lda buyruqlarni Arduinoga yuborishdan

boshlaysiz. Bu ishlagandan so'ng, siz bir vaqtning o'zida bir nechta buyruq qiymatlarini qanday yuborishni va buyruqlarni yuborish uchun oddiy grafik interfeysni yaratishni o'rganasiz.

Eslatib o'tamiz, Arduino ketma - ket portida bufer mavjud. Boshqacha qilib aytganda, siz bir vaqtning o'zida bir nechta bayt ma'lumotni yuborishingiz mumkin va Arduino ularni navbatga qo'yadi va eskizingiz mazmuniga qarab tartibda qayta ishlaydi. Ma'lumotni uzatish vaqtidan ko'ra tezroq yuborish haqida qayg'urishingiz shart emas, lekin ma'lumotni buferdan oshib ketishi va ma'lumot yo'qolishi haqida yuborishingiz kerak.

## **ARDUINOGA KIRISH MALUMOTLARINI ECHO QILISHINI AYTISH**

Siz qila oladigan eng oddiy narsa - bu Arduinoga siz yuborgan hamma narsani qaytarib berishni buyurish . Buni amalga oshirish uchun Arduino asosan ketma -ket kirish buferini kuzatishi va olgan har qanday belgini chop etishi kerak. Buning uchun siz Serial obyektidan ikkita yangi buyruqni bajarishingiz kerak :

■ Serial.available () - hozirda Arduino kiruvchi ketma -ket buferida saqlangan belgilar sonini (yoki baytlarini) qaytaradi. Qachonki u noldan oshsa, siz belgilarni o'qiysiz va ularni kompyuterga qaytarasiz.

■ Serial.read () - buferda mavjud bo'lgan keyingi belgini o'qiydi va qaytaradi.

E'tibor bering, Serial.read() ga har bir qo'ng'iroq faqat 1 baytni qaytaradi, shuning uchun Serial.available() noldan kattaroq qiymatni qaytarguncha uni ishga tushirishingiz kerak. Har safar Serial.read () baytni ushlab qolsa, bu bayt ham buferdan chiqariladi, shuning uchun keyingi bayt o'qishga tayyor bo'ladi. Ushbu ma'lumotga ega bo'lgan holda, siz Arduinoga quyidagidagi echo dasturini yozishingiz va yuklashingiz mumkin.

```
char data;
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  if (Serial.available() > 0)
  {
    data = Serial.read();
    Serial.print(data);
```

```
}  
}
```

Seriya monitorini ishga tushiring va matn kiritish maydoniga xohlagan narsani kiriting. Yuborish tugmachasini bosganingizda, siz yozgan narsalar qaytariladi va ketma - ket monitorda aks etadi. Siz allaqachon har bir buyruq oxiriga "yangi qator" qo'shishni tanlagansiz, bu har bir javob yangi satrda bo'lishini ta'minlaydi. Shuning uchun oldingi eskizda `Serial.println ()` o'rniga `Serial.print ()` ishlatiladi.

### **Foydalanilgan adabiyotlar**

- Jeremy Blum, “Exploring arduino: Tools and techniques for engineering wizardry”, 2013
- Electronics: A Systems Approach (6th edition), Neil Storey, Pearson Education UK, 2017
- Arduino.cc veb sayti ma'lumotlari