

**O'ZBEKISTON RESPUBLIKASI OLIY TA'LIM, FAN VA
INNOVATSIYALAR VAZIRLIGI**

CHIRCHIQ DAVLAT PEDOGOGIKA UNIVERSITETI

Sultanov R.O., Matyakubov K.K.

ROBOTOTEXNIKA ASOSLARI

Amaliy mashg'ulot uchun

O'quv qo'llanma

Chirchiq-2023

UUK:

KBK:

Sultanov R.O., Matyakubov K.K. / Robototexnika asoslari / O'quv qo'llanma.-
Chirchiq, "Zebo Prints", 2023 – 124 b.

Ushbu o'quv qo'llanma Robototexnika asoslari fanining asosiy tushunchalari, zamonaviy dunyoda robotlarning o'rni, robototexnik qurilmalar, svetodiodlar, oddiy motorlar, servo motorlar, driver motorlari bilan ishlash, mikrokontrollerlarni o'rganish, display qurilmalari, arduinov va uni kompyuter bilan bog'lash, Arduino asosida harakanlanuvchi platformalar, sensor yordamida harakanlanuvchi mashina, ovoz yordamida harakatlanuvchi mashina, masofadan turib boshqariladigan mashina yig'ish va dasturlash, Arduino asosida yig'ilgan mehanizmga dasuriy ta'minot o'rnatish va ulardan amaliyotda foydalanish ko'nikmalarini hosil qilishga qaratilgan.

В этом руководстве рассматриваются основные понятия науки о робототехнике, роль роботов в современном мире, роботизированные устройства, работа со светодиодами, простые двигатели, серводвигатели, драйверы двигателей, изучение микроконтроллеров, устройств отображения, ардуино и подключение его к компьютеру, направлены навыки сборки и программирования мобильных платформ на базе Arduino, машины с сенсорным управлением, машины с голосовым управлением, машины с дистанционным управлением, установки программного обеспечения и практического использования собранного механизма на базе Arduino..

This guide covers the basic concepts of the science of robotics, the role of robots in the modern world, robotic devices, working with LEDs, simple motors, servo motors, motor drivers, the study of microcontrollers, display devices, arduino and connecting it to a computer, the skills of assembling and programming mobile Arduino-based platforms, touch-controlled machines, voice-controlled machines, remote-controlled machines, software installation and practical use of the assembled Arduino-based mechanism.

Taqrizchilar: p.f.f.d(phd), TVCHDPU, **Xurramov A.J.**

t.f.n., GFU, **Jo'rayeva N.V.**

O'zbekiston Respublikasi Oliy ta'lim, fan va innovatsiyalar vazirligi Chirchiq davlat pedagogika universiteti kengashining 20__ yil ____-sonli qaroriga asosan 5110100 – Matematika va informatika ta'lim yo'nalishlari bo'yicha tahsil olayotgan talabalar va professor o'qituvchilar uchun o'quv qo'llanma sifatida nashr qilishga tavsiya etilgan.

ISBN:

© Sultanov R.O. va b., 2023

©«Zebo Prints», 2023

KIRISH

Ushbu o'quv qo'llanma amaliy va virtual robototexnikaga bag'ishlanadi. O'quv qo'llanmada bugungi kundagi zamonaviy ta'lim yo'nalishi "Robototexnika asoslari" fanini moddiy texnik bazasi mavjud bo'lmagan ta'lim muassasalarida amaliy va virtual konstruktor dasturlari va simulyatsiya vositalari orqali o'qitish masalalari bayon etilgan. O'quv qo'llanmada Arduino, Lego Digital Designer, 3D TinkerCAD, Bricklink Studio 2.0 virtual konstruktorlik dasturlari imkoniyatlari keltirilgan. O'quv qo'llanmadan Oliy ta'lim muassasalarining 5110100 – Matematika va informatika yo'nalishi talabalariga "Robototexnika asoslari" fanini o'qitishda yaxshi samara beradi. Bu to'plam yosh o'quvchilar bilan birga robototexnikaga qiziqadigan barcha bolalar uchun birdek qiziqarli loyihalarni yaratish uchun keraklidir. Bugungi kunda robototexnika bo'yicha ko'plab xalqaro musobaqalar o'tkazib kelinmoqdaki, ularda farzandlarimiz muvaffaqiyatli ishtirok etib, yuqori o'rinlarni egallay olishi uchun robototexnikani kichik maktab yoshidan boshlab o'rgatish juda muhim. Hozirgi vaqtda sotuvda robotlarni qurish uchun ko'plab tayyor to'plamlarni topish mumkin yoki kerakli komponentlarni alohida sotib olish va o'zingizning dizayningizni yig'ishingiz mumkin. Odatda, bunday to'plamlarni narxi qimmat bo'lib, chekka hududdagi o'quvchilarning bunday to'plamlarni sotib olishlari biroz mushkul. Bu borada onlayn so'rovnoma o'tkazganimizda, aksariyat yoshlar robototexnikaga qiziqishini bildirgan va afsuski, ko'pchilik bu sohani o'rgana olmayotganligiga sabab sifatida jihozlar yetishmasligini ko'rsatgan. Qo'lingizdagi ushbu kitob esa bu muammoni hal etishda sizga yordam beradi. Mazkur kitobda bugungi kundagi zamonaviy ta'lim yo'nalishi "Robototexnika"ni Lego Digital Designer, TinkerCAD, Scratch, Arduino, 3D TinkerCAD kabi virtual simulatsiya dasturlari orqali o'rgatish masalalari bayon etilgan.

Bu qo'llanma orqali o'quvchilar haqiqatan ham tasavvuridagi loyihalarni virtual dasturiy vositalar yordamida yaratishi va o'z imkoniyatlarini namoyish etishlari mumkin. Natijada bola maishiy texnika jihozlarining ishlash prinsiplarini tahlil eta oladi hamda robototexnika va unga tegishli fanlarni yaxshiroq tushuna boshlaydi. Bunda dasturda loyihalarni amalga oshirish qadam-baqadam sodda

tilda bayon etilgan. Dasturlar shunday tanlanganki, murakkablik asta-sekin o'sib boradi va biz boshlang'ich yosh tadqiqotchi bajara oladigan oddiy vazifalardan tortib, hatto tajribali robotistlarni ham qiziqtiradigan juda hayratlanarli vazifalargacha bo'lgan jarayonlarni kitobda muxtasar bayon etganmiz. Bola virtual ilovalar yordamida mustaqil ravishda o'zining robotini yaratadi. Albatta, biz bu jarayonni tushunarli va qiziqarli tarzda bayon qilishga harakat qilamiz. O'quv qo'llanmadan 5110100-Matematika va informatika talim yo'nalishi talabalari bilan bir qatorda barcha umumta'lim maktabi o'quvchilari va "Robototexnika"ga qiziquvchi o'quvchilar, bundan tashqari informatika hamda texnologiya fani o'qituvchilari, "Barkamol avlod" markazlarining "Robototexnika" to'garagi rahbarlari ham foydalanishlari mumkin.

Robototexnika va uning qurilmalari bilan tanishish

«Robototexnika - bu og'ir, zerikarli va xavfli ishlarni bajarishda odamni almashtirish uchun murakkab texnologik jarayonlar va operatsiyalarni, shu jumladan, deterministik bo'lmagan sharoitlarda olib boriladigan ishlarni avtomatlashtirish uchun mo'ljallangan robotlar va robot tizimlarini yaratishga yo'naltirilgan fan. Kasbiy faoliyat ob'ekti robotlar va ishlab chiqarish uchun ham, noishlab chiqarish maqsadlar uchun robotik tizimlar, shuningdek, bunday tizimlarni boshqarish, ularning dizayni va ishlashini boshqarish uchun zarur dasturiy ta'minot va algoritmik yordamdir. "

Robot so'zining etimologik kelib chiqishi Chexiya tilidan ko'proq yoki kam emas. Xususan, ikki atamaning birlashmasida: *robota* "majburiy mehnat" sifatida belgilanishi mumkin va ichida *rabota* "servitut" bilan sinonim. Xuddi shu tarzda, shuni ta'kidlash kerakki, unga ko'proq yoki kamroq murojaat qilingan birinchi marta 1920 yilda yozuvchi Karel Kapekning "Rossum universal robotlari" nomli asarida bo'lgan.

Robototexnika fani dizayn va **texnika** ishlab chiqarish va foydalanish bilan shug'ullanadigan **robotlar** hisoblanadi. Robot, boshqa tomondan, **mashina** ob'ektlar bilan o'zaro aloqa qilish va ma'lum bir tarzda odam yoki hayvon xatti-harakatlarini taqlid qilish uchun dasturlashtirilishi mumkin.

Kompyuter elektronika mexanika va **muhandislik fanlari** ular faqat robototexnika sohalarida **birlashadigan** ba'zi fanlardir. Robototexnikaning asosiy maqsadi - bu avtomatik ravishda ishlaydigan va odamlar uchun qiyin yoki imkonsiz ishlarni bajaradigan asboblarni qurish.

Robototexnika - bu murakkab texnologik jarayonlarni avtomatlashtirish uchun mo'ljallangan robotlar va robot tizimlarini yaratishga yo'naltirilgan fan va texnika sohasi qachon odamni almashtirish uchun operatsiyalar, shu jumladan, deterministik bo'lmagan sharoitda amalga oshiriladigan operatsiyalar og'ir, zerikarli va xavfli ishlarni bajarishdir.

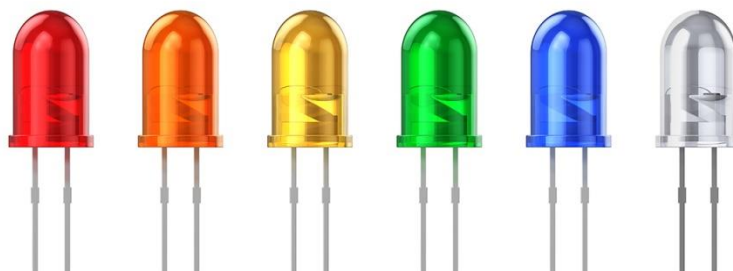
Insonni ilm-fan va texnikaning yangi yo'nalishlarini rivojlantirishga majbur qilishning asosiy sababi bu tashqi muhit

bilan o'zaro aloqada o'z hayotini va imkoniyatlarini yaxshilash uchun unga uzoq vaqt davomida xos bo'lgan istakdir. Bunday o'zaro ta'sirning tabiati juda xilma-xildir va bo'lishi mumkin nafaqat qiyin, balki jismonan ham chidab bo'lmas. Shuning uchun, qadim zamonlardan beri odamlar orzu qilgan ularning ishini engillashtirish, sifat va samaradorlikni oshirish maqsadida turli xil asbob-uskunalar, mexanizmlar, mashinalarni ixtiro qilish.

Kasbiy faoliyat ob'ekti robotlar va sanoat va noishlab chiqarish maqsadlar uchun robotik tizimlar, shuningdek, bunday tizimlarni boshqarish, ularning dizayni va ishlashini boshqarish uchun zarur dasturiy ta'minotdir.

Hozirgi vaqtda robototexnika irodasi va chegarasi bilan rivojlanib, robot xirurgiyasi kabi bir qator fanlarning rivojlanishiga olib keldi. Bunday holda, u insonning sog'lig'ini yaxshilash uchun aniq maqsadga ega va buning uchun juda aniqlik talab qiladigan juda murakkab jarrohlik aralashuvlarni amalga oshiradi. Shunday qilib, robotlar yordamida inson tomonidan qilinadigan xavfni yo'q qilish mumkin.

Yorituvchi diodlar (LED) - har qanday elektron do'konida mavjud bo'lgan rangli chiroqlar zamonaviy jamiyatda keng tarqalgan. Ular stereolarda, avtomobil panellarida va mikroto'lqinli pechlardagi ko'rsatkich chiroqlari, soatlardagi raqamli displeylar, raqamli soatlar va hisoblagichlarda uchraydi. LED'lar shuningdek, masofadan boshqarish pultlari kabi qisqa masofali optik signal uzatish uchun telekommunikatsiya dasturlarida qo'l keladi.



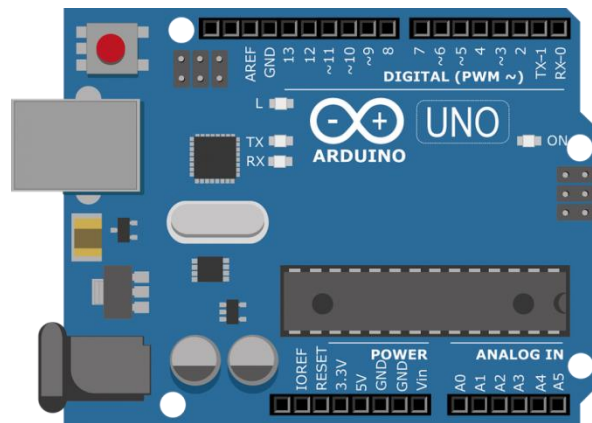
2.3-rasm. Led chiroqlar.

Chiroqlar, aslida, energiya manbaiga bog'langan simlardir. Ular nurni chiqaradi va uning issiqlik energiyasini nur shaklida o'chiradi. Boshqa tomondan, LED, issiqlik ishlab chiqarish o'rniga, elektron qo'zg'alish orqali yorug'lik chiqaradi. Ular "yoqilganda",

elektronlar yuqori elektron zichlikli hududdan past elektron zichlikka ega bo'lgan hududga o'tishadi. Elektronlarning bu harakati nurning chiqishi bilan birga keladi. Qatlamlar orasidagi chegara bo'ylab o'tib ketadigan elektronlar qanchalik ko'proq bo'lsa, yorug'roq bo'ladi.

Arduino - bu unchalik katta bo'lmagan plata bo'lib o'zining protsessori (mikrokontrolleri) va xotirasiga ega bo'lgan qurilma hisoblanadi.

Arduino robototexnika va elektronikiga qiziquvchi va izlanuvchi yoshlarga juda qo'l keladi chunki bu qurilmada kichik va katta bo'lgan dastur, algoritmlar yaratgan holda xar hil qurilmalar, robotlar va boshqa qiziq amaliyotlarni bajarsa bo'ladi. Boshqacha qilib aytganda, Arduino dasturiy va texnik qismlarni birlashtirib beruvchi qurilmadir.



2.6-rasm Arduino platasi

Servo Motor - bu chiqish miliga ega bo'lgan kichik qurilma. Servo kodlangan signalni yuborish orqali ushbu o'qni ma'lum burchakli holatlarga joylashtirish mumkin. Kodlangan signal kirish chizig'ida mavjud ekan, servo milning burchak holatini saqlab qoladi. Agar kodlangan signal o'zgarsa, milning burchak holati o'zgaradi. Amalda servolardan radio boshqariladigan samolyotlarda liftlar va rullar kabi boshqarish sirtlarini joylashtirish uchun foydalaniladi. Ular, shuningdek, radio boshqariladigan avtomobillarda, qo'g'irchoqlarda va, albatta, robotlarda qo'llaniladi.



2.4-rasm servo mator



2.5-rasm servo mator ichki ko'rinishi

Servolar robototexnika sohasida juda foydali. Dvigatellar kichik, o'rnatilgan boshqaruv sxemasiga ega va ularning o'lchamlari uchun juda kuchli. Futaba S-148 kabi standart servo o'lchamlari uchun kuchli bo'lgan 42 oz / dyuym momentga ega.

Bundan tashqari, u mexanik yukga mutanosib quvvat oladi. Engil yuklangan servo, shuning uchun ko'p energiya sarf qilmaydi.

Joystik - bu ma'lumotni odamdan kompyuterga yoki mikrokontrollerga qulay tarzda uzatadigan qurilmalardan biridir. Joystiklar robotlar, mobil platformalar va boshqa mexanizmlarning harakatini boshqarish uchun ishlatiladi.

Ikki tomonlama joystik moduli (2.1-rasm) ikki erkinlik darajasiga ega, bu ikki o'zaro perpendikulyar o'qi bilan sharning bo'g'imiga mahkamlangan tutqich.



2.1-Rasm. Joystik.

Tugma qiyshayganda, har 10 kes potentsiometrning har birining harakatlanuvchi kontaktlari aylanadi, bu X va Y o'qlarining holatini aniqlaydi. Har bir potentsiometrning o'rta kontagi

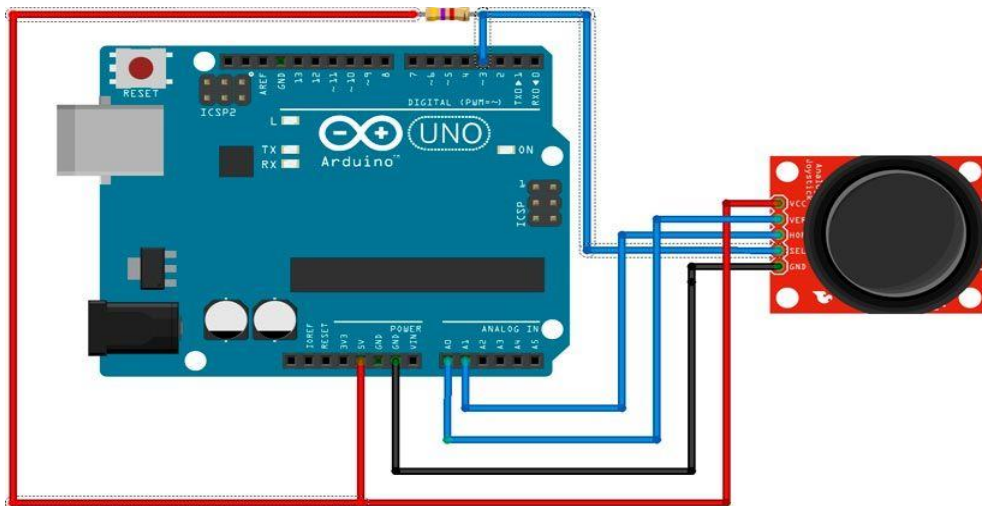
ulagichning VRX va VRY pinlariga chiqariladi va tashqi qismlar elektr ta'minoti va erga ulangan. Shuningdek, joystick tak tugmachasi bilan jihozlangan bo'lib, u dastani vertikal bosilganda paydo bo'ladi; o'qishlar SW kontaktidan olinadi. Chiqarilganda joystick asl markaz holatiga qaytadi.

Texnik xususiyatlari

- Ta'minot kuchlanishi: nominal 3,0 ... 5,5 V;
- Chiqish signali: raqamli (tugma) va analog (X va Y o'qlari);
- Olchamlari: 26mm x 40mm x 22mm.

Arduino platasiga ulanish

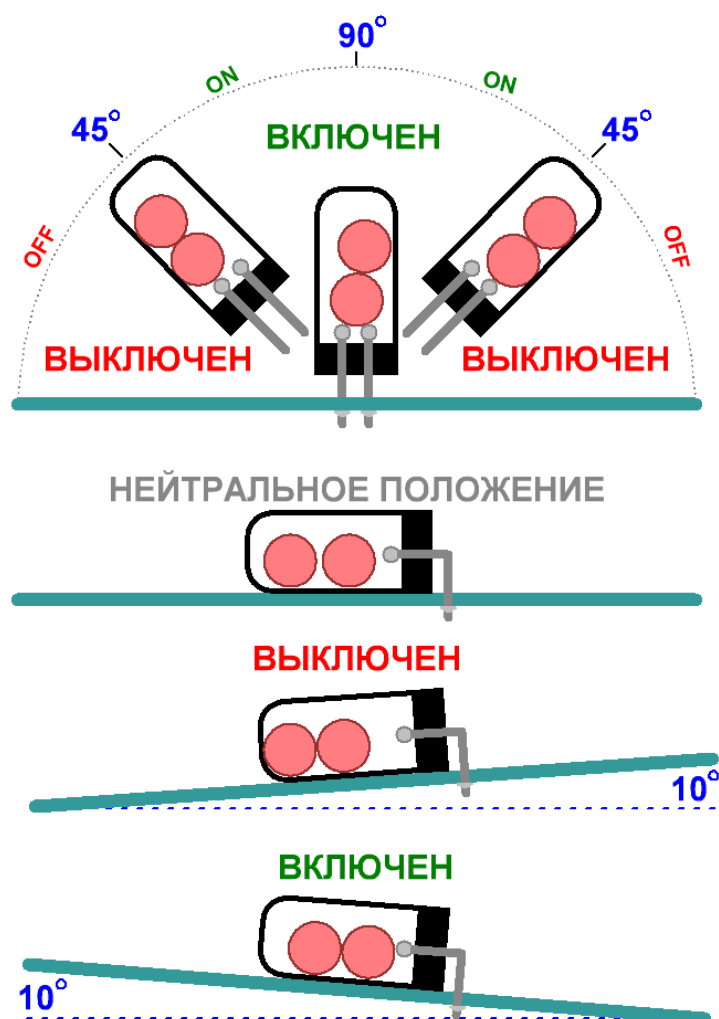
Joystick modulini Arduino platasiga ulash uchun biz Arduining ikkita analog va bitta raqamli pinlaridan foydalanamiz, shuningdek Arduino platasidan GND va + 5V joystick kontaktlarini quvvat bilan ta'minlaymiz. Ulanish diagrammasi 2.2-rasmda keltirilgan.



2.2-Rasm. Joystick modulini Arduino platasiga ulash sxemasi.

Joystickdan ma'lumot olishning eskizini yozamiz. X va Y o'qlari bo'ylab potansiyometrlardan olingan ma'lumotlar 0 dan 1023 gacha qiymatlarni qabul qilishi mumkin. Joystickning sobit pozitsiyasi har bir potentsiometr uchun 511 qiymatiga to'g'ri keladi. Arduining 3-kirish qismidagi tugmani bosganingizda 0 paydo bo'ladi, shovqinni oldini olish uchun tugma chiqishi +5 V ga ko'tarilishi kerak. Biz ma'lumotlarni ketma-ket portga chiqaramiz.

SW-520D egilish va tebranish sensori moduli- bu robototexnik qurilmaning muvozanatini ushlab turishda ishlatiladi.

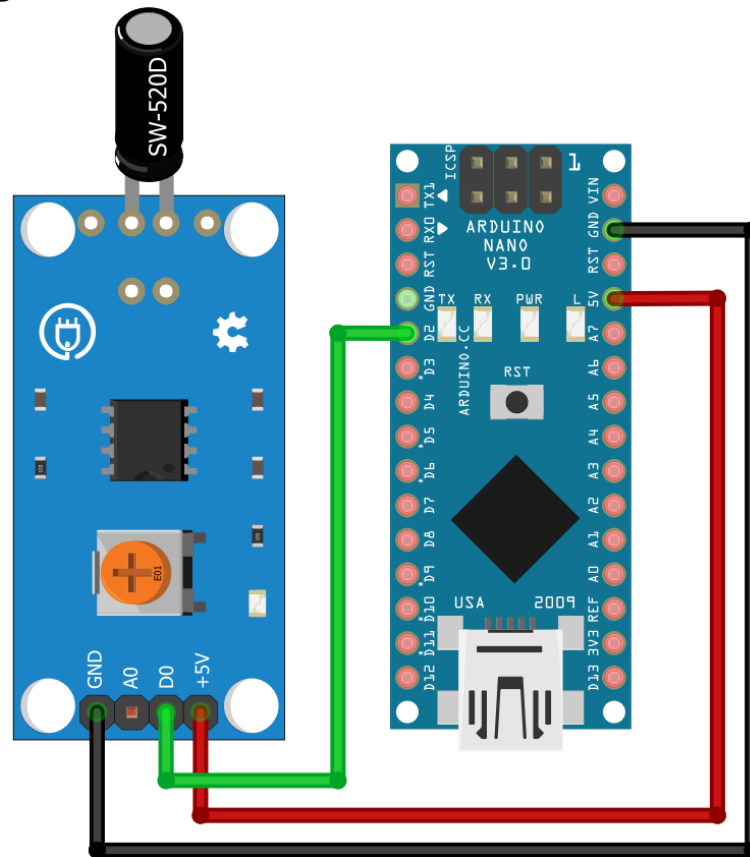


Texnik xususiyatlari

- Ishning vertikal burchagi: **45** ;
- **10** gorizontall ishlash burchagi ;
- Maksimal elektrod oqimi: **300 mA** ;
- Ish kuchlanishi: **3.3-5V** ;
- Komparatorning chiqish oqimi: **15mA dan ortiq** ;
- Chiqish turi: **raqamli** ;
- Ochiq aloqa qarshiligi: **10 Mohm** dan ortiq ;
- Yopiq aloqa qarshiligi: **30 ohmdan** kam ;
- **Taxta** kattaligi: **32x14mm** ;
- **Sensorli** modul o'lchami: **45x14 mm** .
-

Arduinoga ulanish

SW-520D modulini Arduinoga ulash uchun sizga faqat uchta sim kerak bo'ladi, ulardan ikkitasi quvvat beradi, uchinchisi signalni mikrokontrollarga uzatadi



SW-520D modulining Arduinoga ulanish sxemasi

D2 kirishidagi mantiqiy darajani o'qish orqali sensor holatining o'zgarishini aniqlash mumkin, ammo murakkab dasturlarda mikrokontroller boshqa vazifalar bilan band bo'lishi va natijada qisqa muddatli tetikni o'tkazib yuborishi mumkin. Shuning uchun tashqi uzilishlardan foydalanish maqsadga muvofiqdir. Arduino Nano va shunga o'xshash narsalarda D2 va D3 pinlari mavjud.

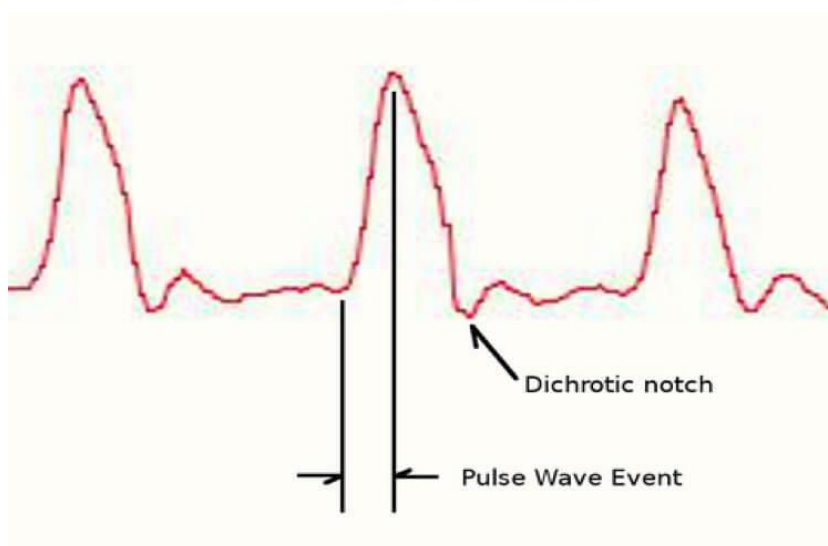
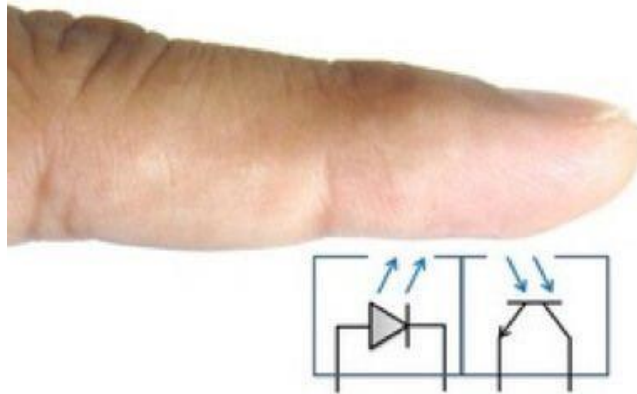
Arduino yurak urish tezligi sensori

Pulse - bu yurakning qisqarishi paytida paydo bo'ladigan qon tomirlari devorlarining ritmik tebranishlari. Yurak-qon tomir kasalliklarini aniqlash uchun yurak urish tezligini o'lchash juda muhimdir. Tananing ortiqcha yuklanishiga yo'l qo'ymaslik uchun, ayniqsa, sport paytida yurak urish tezligining o'zgarishini kuzatib borish muhimdir. Yurak urishining tushunarli parametrlaridan biri bu yurak urish tezligi. Bir daqiqada urish bilan o'lchanadi.



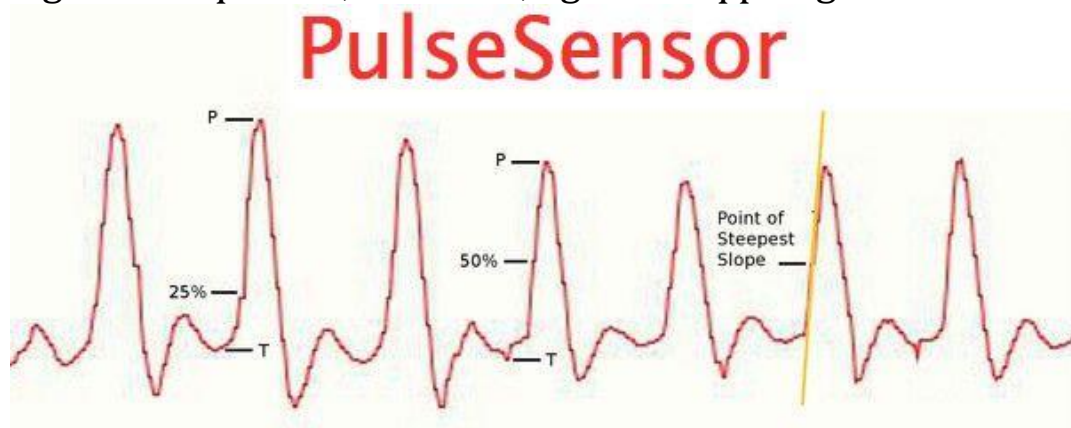
Yurak urishi sensori

Bu fotopletizmografiya usuliga asoslangan analog datchik - qon tomirlari orqali qon oqimining o'zgarishiga qarab, o'lchov o'tkaziladigan sohada (masalan, barmoq yoki quloq pufagi) qon hajmining optik zichligi o'zgarishi. yurak siklining fazasida. Sensor yorug'lik manbai (yashil LED) va fotodetektorni (2-rasm) o'z ichiga oladi, uning kuchlanishi yurak pulsatsiyalari paytida qon hajmiga qarab o'zgaradi. Ushbu grafik (fotopletismogramma yoki PPG diagrammasi) shaklda ko'rsatilgan shaklga ega.



Fotopletizmogramma

Impuls sensori analog signalni kuchaytiradi va uni datchik besleme zo'riqishining o'rtacha nuqtasiga nisbatan normalizatsiya qiladi ($V / 2$). Yurak urish tezligi sensori yorug'lik intensivligining nisbatan o'zgarishiga ta'sir qiladi. Agar sensorga tushadigan yorug'lik miqdori doimiy bo'lib qolsa, signal ADC diapazonining o'rta nuqtasi yonida qoladi. Agar o'rganishning yuqori intensivligi qayd etilsa, u holda signal egri chizig'i yuqoriga ko'tariladi, intensivligi kamroq bo'lsa, aksincha, egri chiziq pastga tushadi.



Puls urishini ro'yxatdan o'tkazish

Biz yurak urish tezligini o'lchash uchun yurak urish tezligi sensori yordamida signal puls boshlanganda to'lqin amplitudasining 50% qiymatiga ega bo'lganda grafadagi nuqtalar orasidagi intervalni o'rnatamiz.

Sensorning texnik xususiyatlari

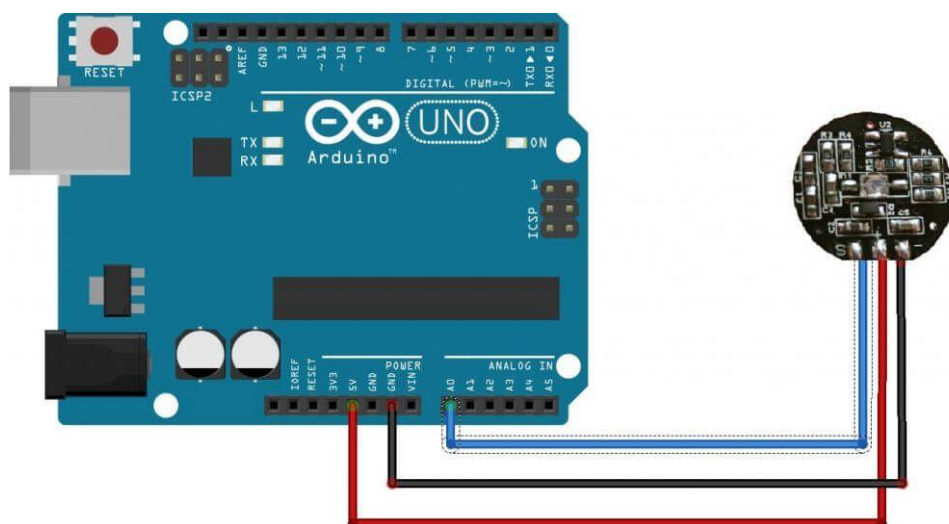
- Besleme kuchlanishi - 5 V;
- Iste'mol oqimi - 4 mA;

Arduinoga ulanish

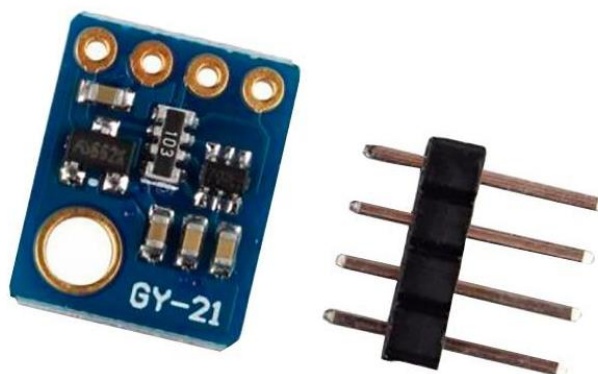
Sensorning uchta uchi bor:

- **VCC - 5V ;**
- **GND - tuproq;**
- **S - analog chiqish.**

Puls sensorini Arudino platasiga ulash uchun datchikning S pimini Arduino analog kirishiga ulashingiz kerak



Arduino platasiga yurak urish tezligi sezgichini ulash **Harorat va namlik sensori HTU-21 (GY-21)**



SHT21 sensoriga asoslangan GY21 namlik va harorat sensori moduli harorat va namlikni o'lchash uchun yuqori aniqlikdagi moduldir.

SHT21 sensoriga asoslangan GY21 namlik va harorat sensori moduli harorat va namlikni o'lchash uchun yuqori aniqlikdagi moduldir. Uning sinfida juda kam xatolar mavjud: haroratni o'lchashda - 0,4%, namlik esa 2% . Bunday past noaniqlik va universal I2C interfeysi bilan ushbu modul sanoat binolarida harorat va namlikni o'lchash uchun javob beradi.

GY-21 texnik xususiyatlari

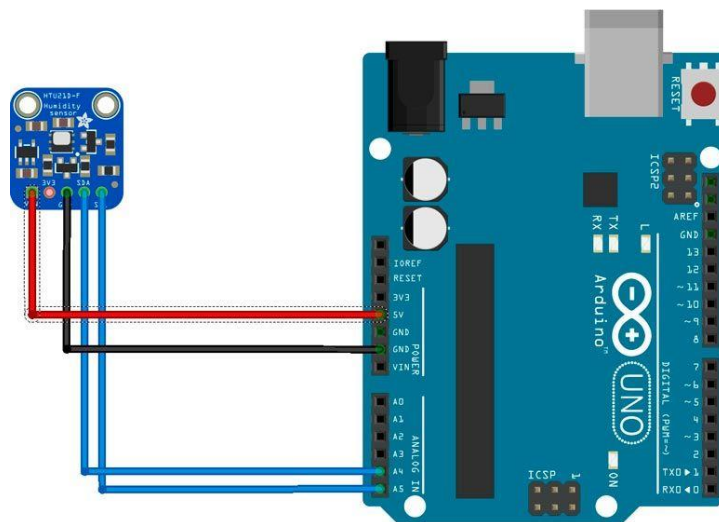
- Sensor: **SHT21**
- Interfeys: **I2C**
- Sensorning ish kuchlanishining diapazoni: **1,9 - 3,6 V**
- Modul besleme zo'riqishida: **5 V**

- O'lchash rejimidagi oqim sarfi: **300 mA**
- Kutishdagi oqim sarfi: **0,15 mA**
- Namlik: ish oralig'i: **0 dan 100% gacha** . O'lchov aniqligi: **± 3% (maksimal), 0-80%**
- Haroratning ishlash oralig'i **-40 ° C dan +125 gacha** . O'lchov aniqligi: **± 0,4 ° C (maksimal), -10 dan 85 ° C gacha**
- Zavodni kalibrlash
- O'rnatilgan batareyani tushirish detektori (etkazib berish kuchlanishi 2,25 V dan pastga tushsa, bayroqchani o'rnatadi)
- Datchiklarni o'z-o'zini diagnostika qilish uchun o'rnatilgan isitgich

GY-21 ni Arduino platasiga ulash

Sensor I2C protokoli ostida ishlaydi. Biz modulni jadvalga muvofiq Arduino platasiga ulaymiz.

GY21	Arduino
VIN	+ 5 B
GND	GND
SCL	A5
SDA	A4



Nazorat uchun savollar:

1. Robototexnika qurilmalarini sanab bering.
2. Harorat va namlik sensori qanday qurilma?
3. Iyostik qurilmasining xususiyatlarni tushuntirib bering.
4. Tebranish sensori plataga qanday ulanadi?

Arduino dasturi bilan tanishish

Ushbu IDE bizga kod yozish uchun matn muharriri, xabarlar maydoni, matn konsoli, umumiy funksiyalar uchun tugmalar bilan asboblari paneli va menyular to'plamini o'z ichiga olgan integratsiyalashgan ishlab chiqish muhitini taklif qiladi. Dasturlarni yuklash va ular bilan bog'lanish uchun dastur Arduino platasiga ulanishi mumkin.

Arduino IDE bilan yozilgan dasturlarga eskizlar deyiladi. Ushbu eskizlar IDE matn muharriri yordamida yoziladi va .ino fayli kengaytmasi bilan saqlanadi. **Arduino Integrated Development Environment (IDE) platformalararo dastur hisoblanadi** (*Gnu / Linux, Windows va macOS*). U Arduino-mos platalarga dasturlar yozish va yuklash uchun ishlatiladi, lekin uchinchi tomon yadrolari yordamida boshqa sotuvchilarning ishlab chiqish kengashlari bilan ham foydalanish mumkin.

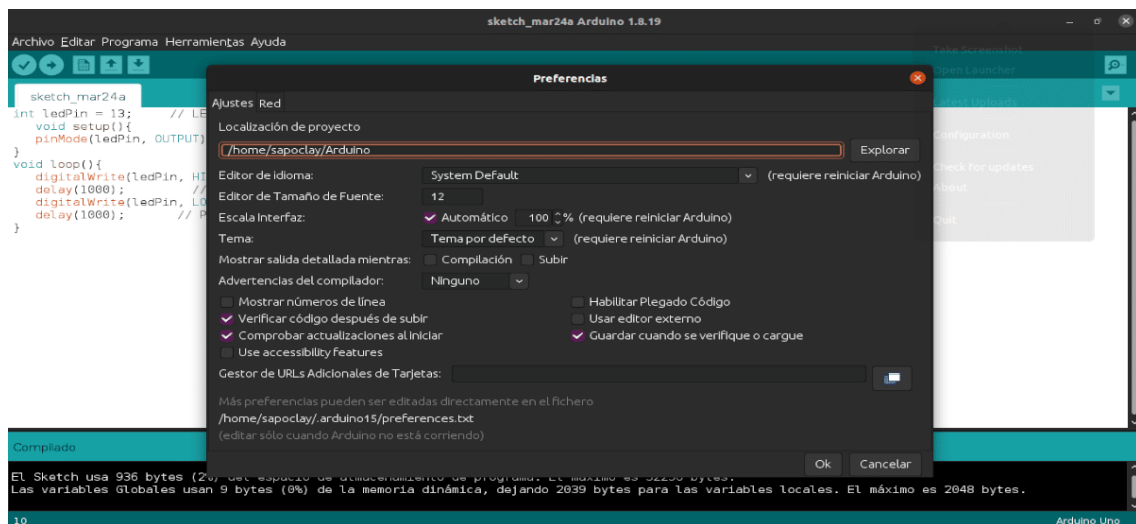
Tahrirlovchi interfeysi sodda va toza va unda biz matnni kesish/joylashtirish va qidirish/almashtirish funksiyalarini topishimiz mumkin. Xabar maydoni saqlash va eksport qilishda ma'lumot beradi, shuningdek, xatolarni ko'rsatadi. Konsol to'liq xato xabarlarini va boshqa ma'lumotlarni o'z ichiga olgan IDE matn chiqishini ko'rsatadi. Oynaning pastki o'ng burchagida ulangan plata va ketma-ket port haqida ma'lumot ko'rsatiladi. **Asboblari panelidagi tugmalar yordamida biz dasturlarni tekshirish va yuklash, eskizlar yaratish, ochish va saqlashimiz mumkin.** Tahrirlash funksiyalarini menyu satrida topish mumkin.



Arduino IDE maxsus kod tuzilmasi qoidalaridan foydalangan holda C va C++ tillarini qo'llab-quvvatlaydi. Shuningdek, u dasturiy ta'minot kutubxonasini taqdim etadi Simlarni ulash loyihasi, bu ko'plab umumiy kiritish-chiqarish protseduralarini ta'minlaydi.



Foydalanuvchi tomonidan yozilgan kod faqat ikkita asosiy funktsiyani talab qiladi, eskizni va dasturning asosiy tsiklini boshlash uchun. Ular kompilyatsiya qilingan va asosiy() dastur stub bilan bog'langan GNU asboblarni zanjiri, bu ham kiritilgan.



Arduino IDE-ni Ubuntu 22.04 yoki 20.04-ga o'rnatish

Arduino IDE ni o'rnatish imkoniyatlaridan biri (1.8.15 versiya) Ubuntu da mavjud **paketidan foydalaning snap Snapcraft-da mavjud**. SNAPD allaqachon yoqilgan va Ubuntu operatsion

tizimlarida foydalanishga tayyor, shuning uchun biz terminalda (Ctrl+Alt+T) terish orqali dasturni o'rnatishimiz kerak bo'ladi:

Arduino ochiq manbali apparat va dasturiy ta'minot kompaniyasi, loyiha va foydalanuvchilar hamjamiyati bo'lib, raqamli qurilmalarni qurish uchun bitta platali mikrokontrollerlar va mikrokontrollerlar to'plamlarini loyihalashtiradi va ishlab chiqaradi.

Arduino orqali nimalar qilish mumkin?

Arduino yordamida siz raqamli soatlar va harorat sensorlari kabi oddiy qurilmalardan tortib, robotlar va uyni avtomatlashtirish tizimlari kabi murakkabroq loyihalargacha turli xil loyihalarni yaratishingiz mumkin. Arduino motorlar, chiroqlar va boshqa elektron komponentlarni boshqarish uchun ishlatilishi mumkin. Bundan tashqari, interaktiv badiiy instalyasiyalar, musiqa asboblari va hatto 3D printerlarni yaratish uchun ham foydalanish mumkin. Arduino - bu elektronika, dasturlash va jismoniy hisoblashni o'rganish uchun ajoyib platforma.

Arduino - bu foydalanish uchun qulay apparat va dasturiy ta'minotga asoslangan ochiq manbali elektronika platformasi. Arduino platalari kirish ma'lumotlarini o'qiy oladi - sensordagi yorug'likni aniqlash, barmoq izini aniqlash yoki buyruqlarni qabul qilish va uni chiqishga aylantirish - dvigatelni faollashtirish, LED lambani yoqish, biror narsani onlayn nashr qilish ham mumkin. Doskadagi mikrokontrollerga ko'rsatmalar to'plamini yuborish orqali siz o'zingizning taxtangizga nima qilish kerakligini aytishingiz mumkin. Buning uchun siz Arduino dasturlash tilidan(Wring ya'ni mikrokontrollerlar uchun ochiq kodli dasturlash tizimi)ga asoslangan va Processing(Processing - bu moslashuvchan dasturiy ta'minot eskiz kitobi va kodlashni o'rganish uchun til)ga asoslangan Arduino Software (IDE)(Arduino uchun dastur yoza oladigan muhit)dan foydalanasiz.

Arduino nomi Italiyaning Ivrea shahridagi bardan olingan bo'lib, u yerda loyiha asoschilarining bir qismi uchrashgan. Bar 1002

yildan 1014 yilgacha Ivrea marti va Italiya qiroli margravi bo'lgan Ivrealik Arduin sharafiga nomlangan.

Yillar davomida Arduino kundalik narsalardan tortib murakkab ilmiy asboblargacha bo'lgan minglab loyihalarning miyasi bo'ldi. Butun dunyo bo'ylab ishlab chiqaruvchilar hamjamiyati - talabalar, havaskorlar, rassomlar, dasturchilar va professionallar - bu ochiq manba platformasi atrofida to'planishdi, ularning hissasi yangi boshlanuvchilar va mutaxassislar uchun katta yordam bo'lishi mumkin bo'lgan aql bovar qilmaydigan miqdordagi mavjud bilimlarni qo'shdi.

Arduino Ivrea Interaction Design Institutida elektronika va dasturlash bo'yicha ma'lumotga ega bo'lmagan talabalarga mo'ljallangan tezkor prototiplash uchun oson vosita sifatida tug'ilgan. Kengroq hamjamiyatga yetib borishi bilanoq, Arduino platasi yangi ehtiyojlar va muammolarga moslashish uchun o'zgarib boshladi va o'z taklifini oddiy 8 bitli platalardan IoT(buyumlar interneti) ilovalari, taqiladigan, 3D bosib chiqarish va o'rnatilgan muhitlar uchun mahsulotlarga ajratdi.

Arduino platalari haqida:

Arduino platalari ochiq manbali mikrokontroller platalari bo'lib, ulardan foydalanish va dasturlash oson bo'lishi uchun mo'ljallangan. Ular interaktiv elektron loyihalarni yaratish uchun ishlatiladi va havaskorlar, ishlab chiqaruvchilar va talabalar orasida mashhurdir. Arduino platalarida mikrokontroller, kirish/chiqish pinlari va modelga qarab turli xil komponentlar mavjud. Arduino platalari Arduino dasturlash tili va Arduino IDE yordamida dasturlashtirilishi mumkin va motorlar, LEDlar, sensorlar va boshqa elektron komponentlarni boshqarish uchun ishlatilishi mumkin.

Arduino Uno - bu ATmega328P asosidagi mikrokontroller platasi. Unda 14 ta raqamli kirish/chiqish pinlari (shundan 6 tasi PWM chiqishi sifatida ishlatilishi mumkin), 6 ta analog kirish, 16 MGts chastotali kvarts kristalli, USB ulanishi, quvvat uyasi, ICSP sarlavhasi va tiklash tugmasi mavjud. U mikrokontrollerni qo'llab-

quvvatlash uchun zarur bo'lgan hamma narsani o'z ichiga oladi. Arduino Uno butun Arduino oilasining eng ko'p ishlatiladigan platasidir. Arduino Uno Arduino kontrollerlar oilasidagi boshqa "aka"lariga nisbatan ancha ixcham, qulay, tez, oddiy va albatta arzonroq hisoblanadi

Nima uchun Arduino?

Oddiy va qulay foydalanuvchi tajribasi tufayli Arduino minglab turli loyihalar va ilovalarda qo'llanilgan. Arduino dasturi yangi boshlanuvchilar uchun qulay, ammo ilg'or foydalanuvchilar uchun etarlicha moslashuvchan. U Mac, Windows va Linuxda ishlaydi. O'qituvchilar va talabalar undan arzon ilmiy asboblarni yaratish, kimyo va fizika tamoyillarini isbotlash yoki dasturlash va robototexnika bilan ishlashni boshlash uchun foydalanadilar. Dizaynerlar va arxitektorlar interaktiv prototiplarni yaratadilar, musiqachilar va rassomlar undan o'rnatish va yangi musiqa asboblari bilan tajriba o'tkazish uchun foydalanadilar. Albatta, ishlab chiqaruvchilar undan Maker Yarmarkasida namoyish etilgan ko'plab loyihalarni qurish uchun foydalanadilar. Arduino yangi narsalarni o'rganish uchun asosiy vositadir. Har bir inson - bolalar, havaskorlar, rassomlar, dasturchilar - to'plamning bosqichma-bosqich ko'rsatmalariga amal qilgan holda o'ylashni boshlashi mumkin,

Jismoniy hisoblash uchun ko'plab boshqa mikrokontrollerlar va mikrokontroller platformalari mavjud. Parallax Basic Stamp, Netmedia-ning BX-24, Phidgets, MIT Handyboard va boshqalar shu kabi funksiyalarni taklif etadi. Ushbu vositalarning barchasi mikrokontroller dasturlashning tartibsiz tafsilotlarini oladi va uni ishlatish uchun qulay paketga o'radi. Arduino shuningdek, mikrokontrollerlar bilan ishlash jarayonini soddalashtiradi, lekin u o'qituvchilar, talabalar va qiziqqan havaskorlar uchun boshqa tizimlarga nisbatan bir qator afzalliklarni taqdim etadi:

Arzon:

Arduino platalari boshqa mikrokontroller platformalariga nisbatan nisbatan arzon. Arduino modulining eng arzon versiyasini qo'lda yig'ish mumkin va hatto oldindan o'rnatilgan Arduino modullarining narxi \$50 dan kam.

Kross platforma :

Arduino dasturiy ta'minoti (IDE) Windows, Macintosh OSX va Linux operatsion tizimlarida ishlaydi. Lekin ko'pgina mikrokontroller tizimlari Windows bilan cheklangan.

Oddiy va tushunarli dasturlash muhiti :

Arduino Dasturiy ta'minoti (IDE) yangi boshlanuvchilar uchun foydalanish uchun qulay, ammo ilg'or foydalanuvchilar uchun ham foydalanishi uchun etarlicha moslashuvchan. O'qituvchilar uchun bu qulay tarzda Processing dasturlash muhitiga asoslangan, shuning uchun o'sha muhitda dasturlashni o'rganayotgan talabalar Arduino IDE qanday ishlashi bilan tanish bo'ladi.

Ochiq kodli va kengaytiriladigan dasturiy ta'minot:

Arduino dasturiy ta'minoti tajribali dasturchilar tomonidan kengaytirilishi mumkin bo'lgan ochiq manbali vositalar sifatida nashr etilgan. Tilni C++ kutubxonalari orqali kengaytirish mumkin va texnik tafsilotlarni tushunmoqchi bo'lgan odamlar Arduino-dan u asoslangan AVR C dasturlash tiliga sakrashlari mumkin. Xuddi shunday, agar xohlasangiz, AVR-C kodini to'g'ridan-to'g'ri Arduino dasturlaringizga qo'shishingiz mumkin.

Ochiq manba va kengaytiriladigan apparat :

Arduino platalarining rejalari Creative Commons litsenziyasi ostida nashr etiladi, shuning uchun tajribali sxema dizaynerlari modulning o'z versiyasini yaratishi, uni kengaytirishi va yaxshilashi mumkin. Hatto nisbatan tajribasiz foydalanuvchilar ham modulning

qanday ishlashini tushunish va pulni tejash uchun panel versiyasini yaratishi mumkin.

Arduino loyihalari uchun maxsus elektronikasi kengaytirish endi har qachongidan ham osonroq. Ushbu kompaniya boshqa qo'shimchalar yordamida raqamli va interaktiv qurilmalarni yaratish uchun apparat to'lovlarini kengaytirmoqda va o'zgartirmoqda. Va vinolar ma'lum bir koddan loyiha sifatida kengaytirilayotganligi uchun barcha kredit.

Xo'sh, siz allaqachon ushbu kichik va oddiy kompyuterlarga tayanadigan imkoniyatlarning samarali g'alabalarining bo'g'uvchi dunyosiga tashlanganingiz uchun, siz o'zingizning imkoniyatlaringizdan foydalanishingiz kerak. Arduino IDE(Kengaytirishning o'rtasida).

Mikrokontroller platalarini dasturlashni joriy kod bilan boshlang

Biz ushbu kengashlarni dasturlashni so'raydigan IDE haqida gapiramiz va bu erda biz Arduino uskunasi tegishli hamma narsani bilamiz, siz bunday dasturda bilishingiz mumkin. Bu kodni yozish va kompilyatsiya qilish uchun muharrir. Yaxshi so'z, shuningdek, oddiy yordamchilar doirasidan tashqariga chiqadigan turli xil koristuvachevi zasobi-ni targ'ib qilayotganlardir, shuning uchun siz ushbu to'lovlar bilan noldan kurashishni boshlashingiz mumkin: shuningdek, turli funktsiyalar va qo'shimchalar uchun butts va kodlar bilan ta'minlangan.

Von shunday yaratilganki, agar bir lahza bo'lsa, siz maxsus bilimsiz g'alaba qozonishni boshlashingiz mumkin edi. Barcha maqsadlar va maqsadlar uchun YouTube'da turli loyihalardan minglab video darsliklar mavjud bo'lib, ular mo''jizaviy tarzda tushuntirilgan.

Katlanadigan vino tizimlari original dasturiy ta'minot - Arduino yordamini izlaydi. U xuddi shu nom ostida chiqarilgan haydovchi platasi bilan o'zaro bog'lanish uchun yaratilgan bo'lib, ikkita komponentdan iborat bo'lishi mumkin: tarqatishning o'rtasi (ibtidoiy kod muharriri, kompilyator, proshivkani miltillovchi modul bilan) va plata (USB bilan birga). kabel). Arduino uchun oddiy

interfeysda yuqori panelda "Fayl", "Tahrirlash", "Sketch", "Xizmat", "Finish" tugmalari mavjud.

Individual xususiyatlar:

- dastur-nazoratchining smut tan olinishi - robotlashtirilgan binolarni yaratish va hatto keyingi nazorat;
- dasturiy ta'minot va yoga asosida topilgan robotlar veb-dasturchilar uchun xizmatda mavjud;
- Arduino bir-biriga zid bo'lmagan dasturlarsiz (masalan, Pur Data, Macromedia Flash, Super Collider) shaxsiy plaginning dasturiy ta'minotiga osongina ulanishi mumkin;
- Qo'shimcha uskunarlar sotib olish uchun PZ kompyuteriga o'rnatilgan - USB kabeli uchun to'lov;
- dasturlarni yozishda uni o'zimning xohishim bilan C ++ dasturlash bo'yicha odamlar ijro etadilar, proshivka yaratish parchalari faqat ortiqcha belgilarda ko'rsatiladi;
- kompyuterda o'rnatilgan boshqa dasturlar, shu jumladan Windowsning joriy versiyasida qo'llab-quvvatlanadigan barcha dasturlar bilan o'zaro ishlash oson;
- Arduino arsenalida - kompilyator, dasturiy ta'minotdagi kechirimlar va noaniqliklarni sinash, aniqlash va tuzatish vositalari, amaliy, ixcham grafik qobiq.

Soat ostida modellashtirish boy galuzeylarning asosi edi. Uzoq vaqt davomida aeronavtika va aviatsiya sohasida ajoyib simulyatsiyalar amalga oshirildi. Bugungi Arduino simulyatorlari bizga yangi boshlanuvchilar va professional dizaynerlarga g'oyalarni dasturlash va sinab ko'rishni o'rganishga imkon beradi, bir vaqtning o'zida o'z tiyinlariga energiya sarflashdan qo'rqmaydi.

Arduino simulyatorlari dizayn va sxema asoslarini o'rganmoqchi bo'lgan dasturchilar va dizaynerlar uchun ajoyib platformalardir. Bunday dasturlarning muvaffaqiyati sizga yordam beradi, siz o'rganishga umid qilishingiz mumkin, giyohvandlarga zarar etkazishdan qo'rqmang. Bundan tashqari, elektr jihozlari bilan

bog'liq muammolarni ularning qanday ishlashini bilmagan holda ayblashi mumkin bo'lgan talabalar ushbu simulyatorlardan yordam so'rash orqali ko'plab nuanslarni tushunishlari mumkin. Tse o'sha soatda sizga ko'p tiyinlarni ayamaydi.

Arduino simulyatorlarining yana bir katta yutug'i shundaki, vinolar maqtoqlar qatorini ko'paytirmoqda, shuning uchun koristuvach aniq biladi, de va qandaydir vino qatorida, yoki u bunday ishlamaydi. Simulyatorlar turli shakllarda qo'llaniladi va asosiy operatsion tizimlar - Windows, Linux va Mac OS bilan muvofiqligi uchun kengaytiriladi. Shunday qilib, sizning kompyuteringiz ekotizimlari uchun yaratilgan ajoyib Arduino simulyatori haqida bilish uchun biz eng mashhur dasturlarning ro'yxatini tuzdik.

PaulWare tomonidan Arduino Simulator

Nomidan farqli o'laroq, ushbu Arduino simulyatori Pol nomi ostida sotuvchi tomonidan yaratilgan. Ochiq kodli simulyator va muxlislarning bir qismini yig'ib, o'z g'oyalarini qo'shish va ular haqida qo'l san'atlarini yaratish, simulyatorni qanday mag'lub etish uchun bir soat vaqt ajrating. Bu xarajatsiz mahsulot Windows ekotizimlari uchun muhim va u yangi kelganlar uchun yetarlicha yordam beradi.

Asosiy komponentlar, yaki vín zabezpechuê loyihangizni qo'llab-quvvatlash uchun yengil qisqa soatlik vimikach, matritsali klaviatura 4 dan 4 gacha, RK-displayli matritsali klaviatura 4 dan 4 gacha, aylanadigan o'zgartirgich va boshqalar. YouTube videosi Arduino simulyatorini ishga tushirish uchun yetarlicha ma'lumot beradi.

Yangisi uchun Arduino quruvchisi forumiga maxsus bo'lim ham o'tkazildi, unda siz ushbu dizayn sxemasini yangilash haqida ko'proq ma'lumot olish uchun ishtirokchi bo'lishingiz mumkin.

Nazorat uchun savollar:

- 1.Arduino dasturi qanday dastur?
- 2.Arduino dasturi qaysi dasturlash tillarini qo'llab quvvatlaydi?
- 3.Arduinoning qanday turlari mavjud?
- 4.Arduinoning qanday xususiyatlari bor?

Bir nechta svetodiodlar va paykasiz maket platalari bilan ishlash asoslari.

1. Ishning maqsadi

Arduinoning barcha o'quv qo'llanmalari va yoritgichlari LEDning miltillashi misolidan boshlanadi. Buning ikkita sababi bor: bunday loyihalar minimal dasturiy ta'minotni talab qiladi va ular elektron sxemani yig'masdan ham ishga tushirilishi mumkin va bu har qanday Arduino platasida LED mavjud.

1. Laboratoriya ishining vazifasi

2. Arduino platasida LED chirog'ini yig'ing.

3. Arduino platasida LED chirog'ini belgilangan vaqtda uchib yonishni dasturlash

4. Laboratoriya ishning maqsadi va mazmuni

Tadqiqot ob'ekti - Arduino Uno, Nano yoki Mega-karta, 13-pinga ulangan ichki LED asosida yaratilgan belgilangan vaqtda uchib yonishni modeli. Vazifa LED chiroqni dasturlash muhiti xususiyatlari bilan tanishishdan iborat.

Zarur jihozlar

1. Arduino Uno platasi

2. Platani kompyuterga bog'lovchi USB kabel

3. Arduino taxtasi(Breadboard)

4. LED yoritkich (*eslatma yoritkich uzun oyiqchasi +musab va kalta oyiqchasi –manfiy*)

5. 0.22 vt. Rezistor

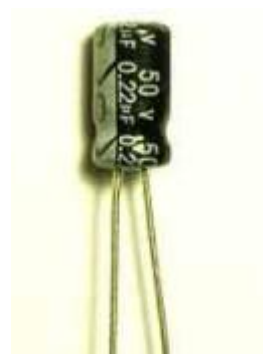
6. Arduino kabellari 2 dona (-va+ uchun)

5. Uskunalar, texnik vositalar va vositalarning tavsifi

Arduino Uno, Nano yoki Mega-karta, 13-pinga ulangan ichki LED mikrokontroller tomonidan boshqariladi.

Arduino loyihasi har doim elektron zanjir, ba'zi bir-biriga bog'langan apparat va mexanik qurilmalar, energiya tizimlari va barcha bu betartiblikni boshqaruvchi dasturlarning birikmasidan iborat.

Arduino Uno va LED ko'rinishi 1.1-rasmda keltirilgan.



1.1-pacm – Arduino Uno, rezistor va LED

Biz Arduinoni kompyuterga ulaymiz, taxtaning hayotga kirishiga va yuklash chiroqlari bilan yonib-o'chishiga ishonch hosil qilamiz. Ko'plab taxtalarda "miltillovchi" eskiz allaqachon mikrokontrollaga yozilgan, shuning uchun LED yoqilgandan so'ng darhol miltillay boshlashi mumkin.

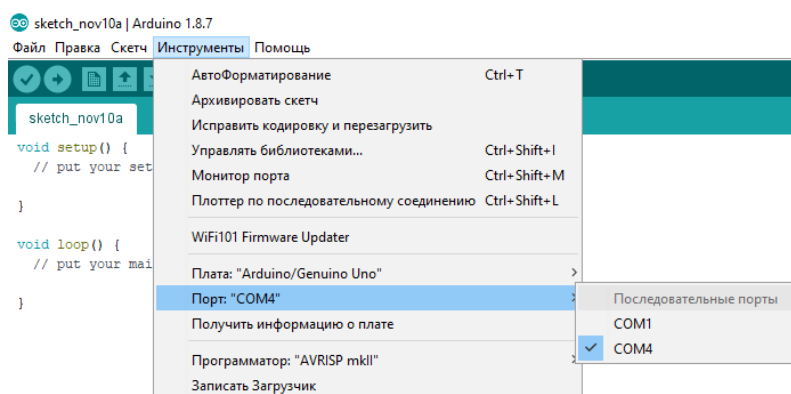
Bajarish tartibi

1. LEDni taxtaga joylashtiramiz
2. Rezestorni Ledga moslab (+musbat tomonga)

Arduinoda dasturlash

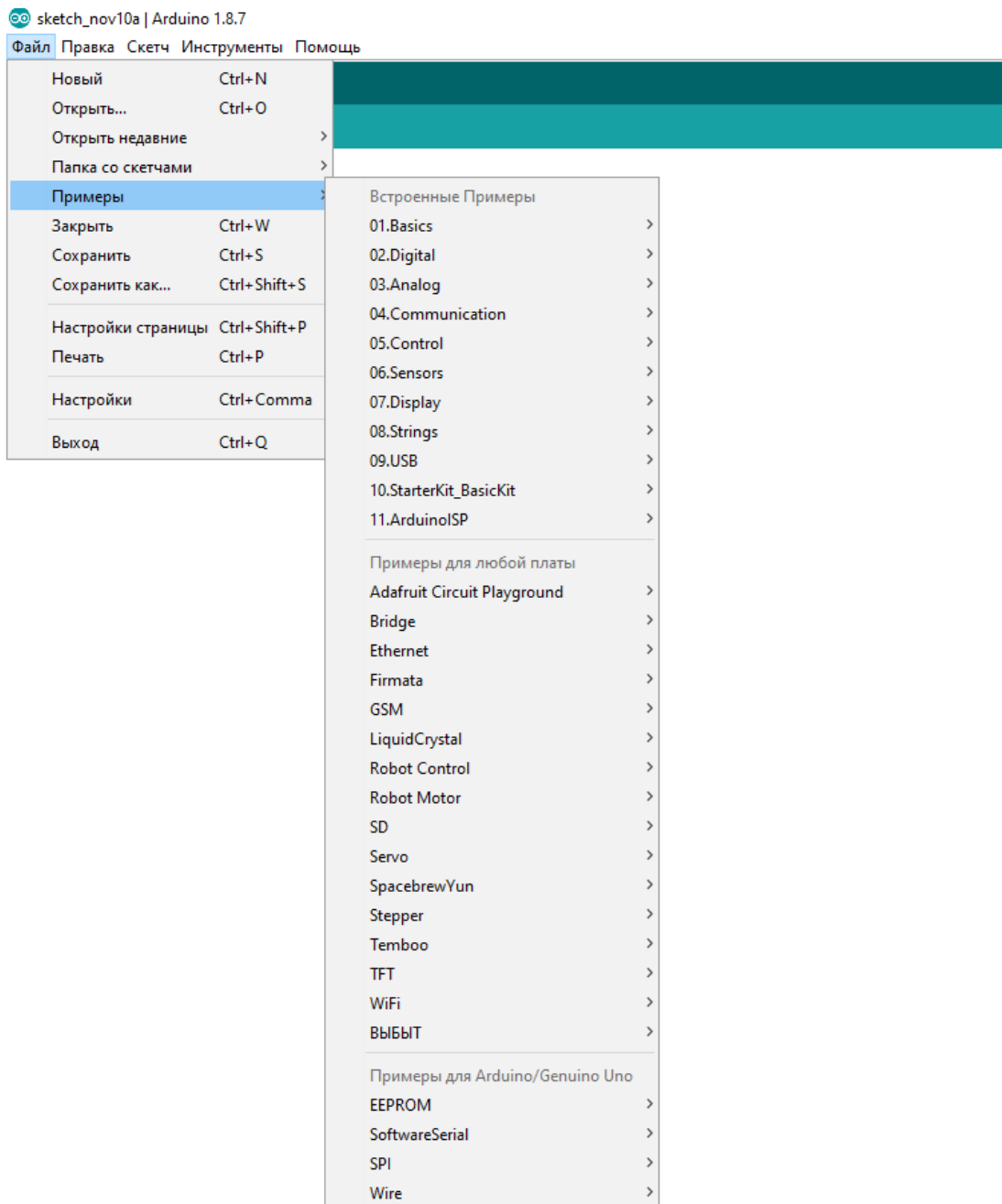
Agar sizning taxtangizda yuklab olingan mayoq chizmasi bo'lmasa, muammo bo'lmaydi. Arduino dasturlash muhitida mavjud bo'lgan tayyor misolni osongina yuklab olishingiz mumkin.

Arduino IDE dasturini oching, to'g'ri port tanlanganligiga ishonch hosil qiling.



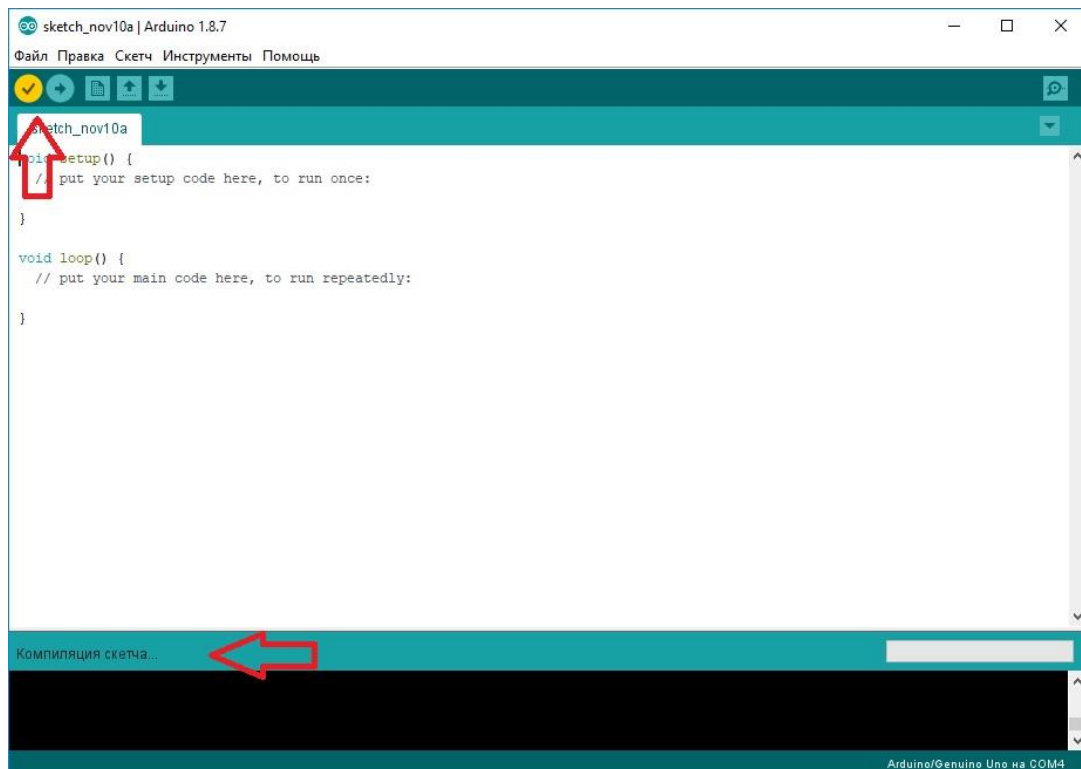
Arduino portini tekshirish - maksimal raqam bilan portni tanlang.

Keyin tayyor Blink eskizini oching - bu o'rnatilgan misollar ro'yxatida. Fayl menyusini oching, "Namunalar" kichik bandini, keyin "Asosiy" ni toping va "Blink" faylini tanlang.

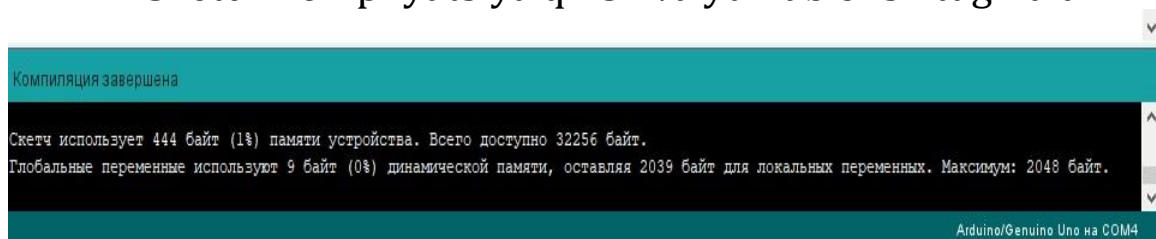


Arduino IDE-da Blink misolini ochish

Ochiq oynada dasturning manba kodi (eskiz) aks etadi, uni boshqaruvchiga yuklashingiz kerak bo'ladi. Buning uchun o'q bilan tugmani bosish kifoya.



Sketch kompilyatsiya qilish va yuklab olish tugmalari.



Arduino IDE ma'lumotlari - Yuklash tugallandi

Biz biroz kutamiz (quyida yuklab olish jarayonini kuzatishingiz mumkin) - barchasi shu. Kengash yana bir nechta LED bilan miltillaydi, so'ngra LEDlardan biri o'lchov aylanishini yoqish va o'chirishni boshlaydi.

6 Hisobotning mazmuni va dizayniga qo'yiladigan talablar

1. Ishning nomi va maqsadi.
2. Dasturlarning skrinshotlari.
3. Xavfsizlik savollariga javoblar.

Arduino platformasida svetafor yaratish.

1. Ishning maqsadi

Arduinoning barcha o'quv qo'llanmalari va yoritgichlari LEDning miltillashi misolidan boshlanadi. Buning ikkita sababi bor: bunday loyihalar minimal dasturiy ta'minotni talab qiladi va ular elektron sxemani yig'masdan ham ishga tushirilishi mumkin va har qanday Arduino platasida LED chiroqlari mavjud bo'ladi.

2. Laboratoriya ishining vazifasi

1. Arduino platasida LED chiroqlarini yig'ing.
2. Arduino platasida LED chiroqlarini belgilangan vaqtda o'chib yonishini dasturlash

3. Zarur jihozlar

1. Arduino Uno platasi
2. Platani kompyuterga bog'lovchi USB kabel
3. Arduino taxtasi(Breadboard)
4. LED yoritkich (*eslatma yoritkich uzun o'yoqchasi +musab va kalta o'yoqchasi –manfiy*)
5. 0.22 vt. Rezistor
6. Arduino kabellari 2 dona (-va+ uchun)

1. Uskunalar, texnik vositalar va vositalarning tavsifi

Arduino Uno, Nano yoki Mega-karta, pinlarga ulangan ichki LED mikrokontrollerlar tomonidan boshqariladi.

Arduino loyihasi har doim elektron zanjir, bir-biriga bog'langan apparat va mexanik qurilmalar, energiya tizimlari va barcha betartiblikni boshqaruvchi dasturlarning birikmasidan iborat.

Biz Arduinoni kompyuterga ulaymiz, plataning ulanishi va yuklash chiroqlari bilan yonib-o'chishiga ishonch hosil qilamiz. Ko'plab platalarda "miltillovchi" eskiz allaqachon mikrokontrollaga yozilgan, shuning uchun LED yoqilgandan so'ng darhol miltillay boshlashi mumkin.

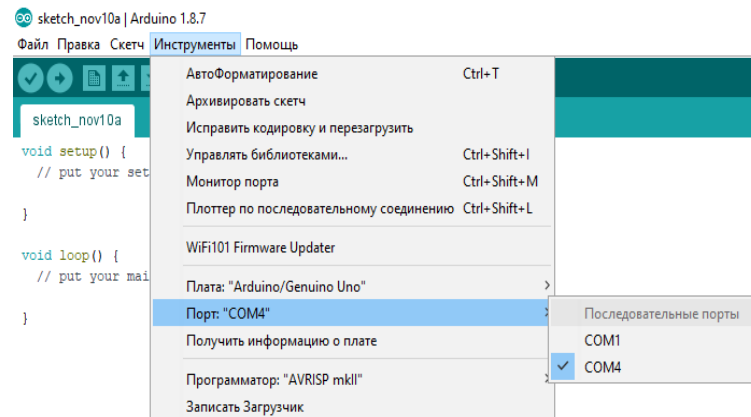
1. Bajarish tartibi

2. LED larni plataga joylashtiramiz
3. Rezistorni Ledga moslab (*+musbat tomonga*) ulaymiz
1. Platani kompyuterga ulab, Arduinoda tuzilgan dasturni plataga yuklaymiz

Arduinoda dasturlash

Agar sizning platangizda yuklab olingan mayoq chizmasi bo'lmasa, muammo bo'lmaydi. Arduino dasturlash muhitida mavjud bo'lgan tayyor misolni osongina yuklab olishingiz mumkin.

Arduino IDE dasturini oching, to'g'ri port tanlanganligiga ishonch hosil qiling.

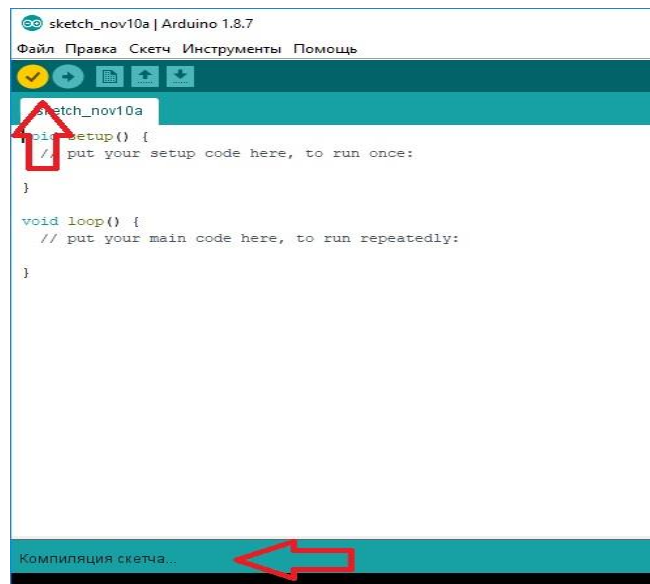


1-rasm. Arduino platasini ulanish porti

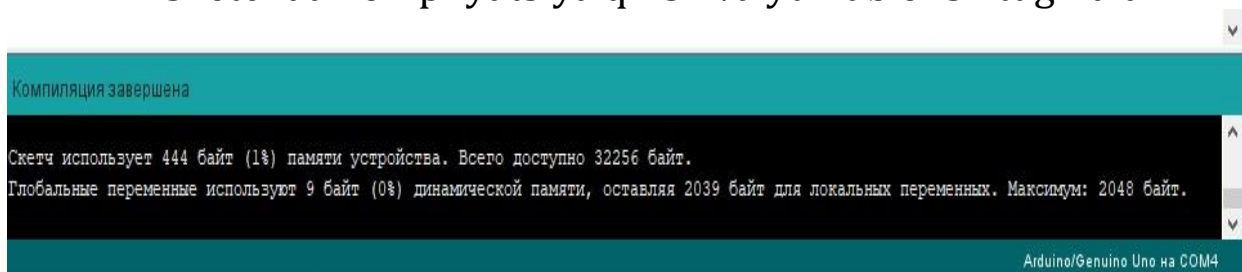
Arduino portini tekshirish - maksimal raqam bilan portni tanlang.

Keyin tayyor Blink eskizini oching - bu o'rnatilgan misollar ro'yxatida. Fayl menyusini oching, "Namunalar" kichik bandini, keyin "Asosiy" ni toping va "Blink" faylini tanlang.

Ochiq oynada dasturning manba kodi (eskiz) aks etadi, uni boshqaruvchiga yuklashingiz kerak bo'ladi. Buning uchun quyidagi tugmani bosish kifoya.

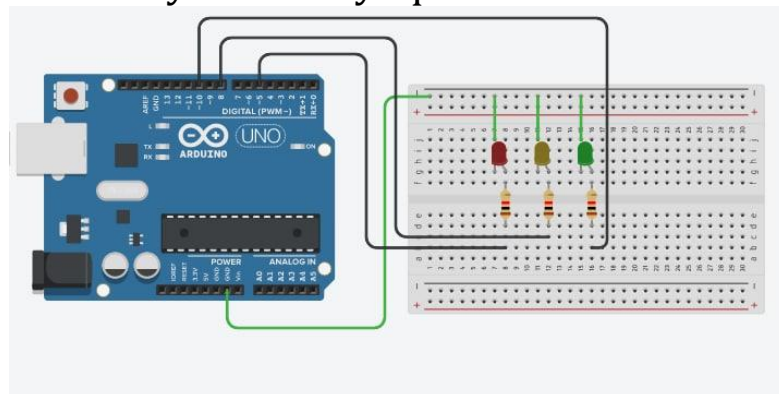


2-rasm. Dasturni kompilyatsiya qilish.
Sketchda kompilyatsiya qilish va yuklab olish tugmalari.



3-rasm. Kompilyatsiya jarayonining tugashi.

Arduino IDE ma'lumotlari - Yuklash tugallandi Biz biroz kutamiz (quyida yuklab olish jarayonini kuzatishingiz mumkin) - barchasi shu. Kengash yana bir nechta LED bilan miltillaydi, so'ngra LEDlardan biri o'lchov aylanishini yoqish va o'chirishni boshlaydi.



4-rasm (arduino platasiga ulash sxemasi)

Dasturdagi kodi.

```

void setup()
{
    pinMode(5,OUTPUT);
    pinMode(8,OUTPUT);
    pinMode(10,OUTPUT);
}
void loop()
{
    digitalWrite(5,HIGH);
    digitalWrite(8,LOW);
    digitalWrite(10,LOW);
    delay(10000);
    digitalWrite(5,LOW);
    digitalWrite(8,HIGH);
    digitalWrite(10,LOW);
    delay(500);
    digitalWrite(5,LOW);
    digitalWrite(8,LOW);
    digitalWrite(10,HIGH);
    delay(10000);
    digitalWrite(5,LOW);
    digitalWrite(8,LOW);
    digitalWrite(10,HIGH);
    delay(100);
}

```

Nazorat uchun savollar:

1. LED chiroqlari nima maqsadda ishlatiladi?
2. Paykasiz maketda LED chiroqlar qanday ulanadi?
3. Arduino dasturi orqali LED chiroqlariga qanday tartibda buyruqlar berish mumkin?
4. Kompilyatsiya jarayoni qanday jarayon hisoblanadi?

Drayver motori orqali oddiy motor bilan ishlash

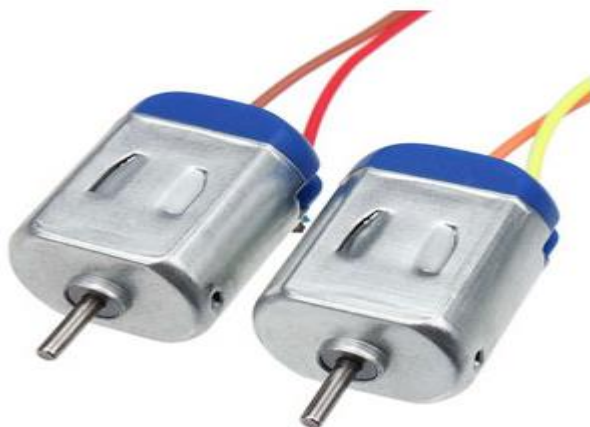
Ishdan maqsad : Drayver L293D orqali oddiy motorni ishlatish uchun uni Arduino ga ulab qurilma yasash.

Nazariy qism :

Motorlar ko'pincha Arduino loyihalarida qo'llaniladi. Ular g'ildiraklarni boshqaradi, burama pervanellarni aylantiradi, sanoat robotlarining qo'llarini aylantiradi va 3D-printerni tashiydi.

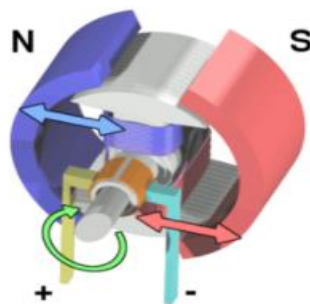
Har bir yangi robototexnika muhandisi dvigatelni mikrokontrollerga ulash muammosiga duch keladi. Dvigatelda faqat ikkita pin bor va u dvigatelni Arduinoning raqamli pimlariga ulab, so'ngra dastur orqali yoqish va o'chirish mumkin. Ko'pincha har xil o'yinchoqlarda ishlatiladigan kichik dvigatel ham ishlash uchun 200 mA dan 1 Ampergacha bo'lgan oqimni talab qiladi. Va Arduino-ning raqamli chiqishi bizga faqat 20 mA quvvat berishi mumkin.

Odatda pervanel bilan starter to'plamiga kiritilgan doimiy dvigatel bilan ishlaylik.



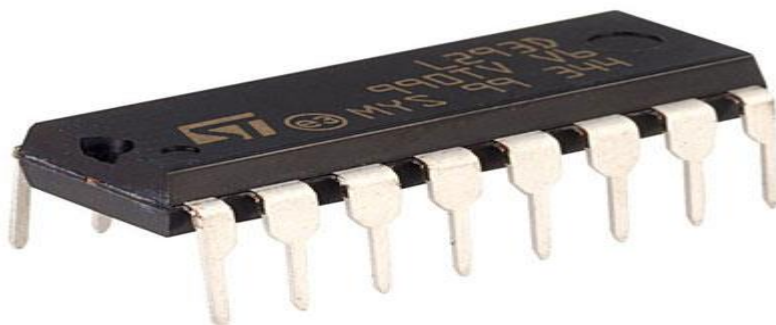
1-rasm. DC motor.

Ikki kutupli rotorli (armatura) va doimiy magnitlangan statorli elektr motorining ishi. "N" va "S" magnitlarning ichki o'qi yuzlaridagi qutblanishlarni belgilaydi; tashqi yuzlari qarama-qarshi qutblarga ega. + Va - belgilari armatura bobinlariga tok etkazib beradigan komutatorga doimiy tokning qaerda qo'llanilishini ko'rsatadi



2-rasm. DC motor ishlash tartibi.

Har xil turdagi motorlarni boshqarishga imkon beradigan ko'plab tayyor mikrosxemalar mavjud. Biz L293D drayverini ko'rib chiqamiz.



L293D

Mikrosxem ikkita H ko'prikdan iborat bo'lib, demak siz bir vaqtning o'zida ikkita motorni boshqarishingiz mumkin. Har bir ko'prikk to'rtta himoya diodasi va qizib ketishdan himoya bilan jihozlangan. L293D dvigatelga etkazishi mumkin bo'lgan maksimal oqim 1,2A ni tashkil qiladi. Ish oqimi - 600mA. Maksimal kuchlanish 36 V ni tashkil qiladi.

L293D 16 pinli DIP paketiga ega. Pinout quyida. Pinslar soat sohasi farqli o'laroq sanaladi va mikrosxemalar pog'onasidan boshlanadi.

L293D			
1	EN1	16	+V
2	IN1	15	IN4
3	OUT1	14	OUT4
4	0V	13	0V
5	0V	12	0V
6	OUT2	11	OUT3
7	IN2	10	IN3
8	+Vmotor	9	EN2

2-rasm L293D pinlari .

- + V - mikrosxemali elektr ta'minoti, 5V;
- + Vmotor - 36V gacha bo'lgan dvigatellar uchun quvvat manbai;
- 0V - tuproq;
- En1, En2 - H ko'priklarini yoqish / o'chirish pinlari;
- In1, In2 - birinchi H ko'prigining boshqarish chiqishlari;
- Out1, Out2 - birinchi motorni ulash uchun pinlar;
- In3, In4 - ikkinchi H ko'prigining boshqarish chiqishlari;
- Out3, Out4 - ikkinchi motorni ulash uchun pinlar.

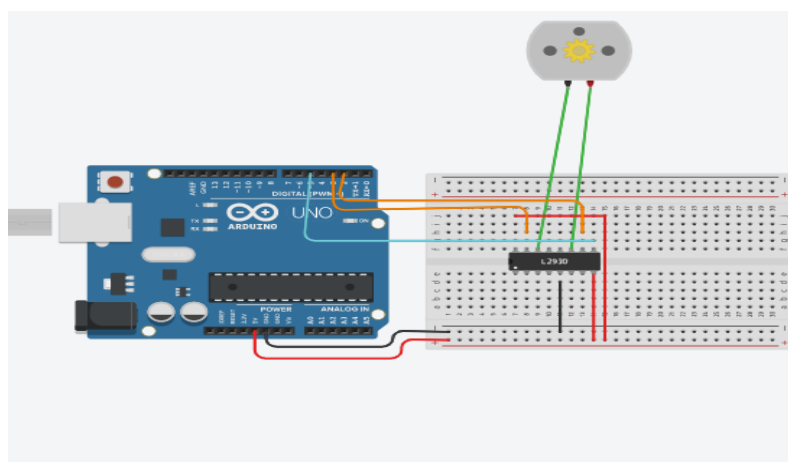
En1 va En2 pinlari ko'priklarni ochish yoki yopish uchun ishlatiladi. Agar biz 0 ga En-ni bersak, mos keladigan ko'prik to'liq o'chiriladi va dvigatel aylanishni to'xtatadi. Ushbu signallar biz uchun PWM signalidan foydalangan holda dvigatelning surilishini boshqarish uchun foydalidir.

Ishni bajarish:

Kerakli qurilmalar:

1. Arduino Uno.
2. Breadbord (paykasiz maket)
3. Drayver L293D
4. DC motor.

Ulanish tartibi:



3-rasm. Ulanish jarayoni.

Har soniyada yo'nalishni o'zgartirib, dvigatelni aylantirish uchun eskiz. AnalogWrite () vosita quvvatini boshqarish uchun PWM signalidan foydalanadi. Biz haydovchiga dvigatelni maksimal tezlikda aylantirishni aytamiz, bu PWM signaliga 255 ta mos keladi. Shuni ta'kidlash kerakki, PWM signalini ikki baravarga kamaytirish tezlikning ikki barobariga olib kelmaydi.

```
const int in3 = 2;
const int in4 = 3;
const int en2 = 5;

void setup() {
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);
  pinMode(en2, OUTPUT);

  analogWrite(en2, 255);
}

void loop() {
  digitalWrite(in3, LOW);
  digitalWrite(in4, HIGH);
  delay(1000);
  digitalWrite(in3, HIGH);
  digitalWrite(in4, LOW);
  delay(1000);
}
```

Dasturni murakkablashtiramiz. Yo'nalishdan tashqari, biz kuchni ham o'zgartiramiz.

```
const int in3 = 2;
const int in4 = 3;
const int en2 = 5;

void setup() {
```

```

    pinMode(in3, OUTPUT);
    pinMode(in4, OUTPUT);
    pinMode(en2, OUTPUT);
}

void loop() {
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);

    analogWrite(en2, 150);
    delay(2000);
    analogWrite(en2, 255);
    delay(2000);

    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);

    analogWrite(en2, 150);
    delay(2000);
    analogWrite(en2, 255);
    delay(2000);
}

```

Bu kod ishlaganda natija yaxshi yakun topmaydi. Birinchidan, vosita past tezlikda aylanadi, keyin maksimal tezlikka etadi va hamma narsani teskari yo'nalishda takrorlaydi. Shuning uchun standart tezlik 255 berishimiz kerak.

Nazorat uchun savollar:

1. Drayver motori qanday qurilma?
2. DC motori arduinoga qanday ulanadi?
3. Drayver motorini ishlatish uchun qanday dastur tuzish kerak?
4. Drayver motor loyihasini yig'ish uchun qanday qurilmalar kerak?

Servomotor bilan ishlash. Bosqichli motor bilan ishlash.

Servo dvigatelning ish printsipti nisbatan sodda, ammo uning ishi yanada samaralidir. Servo pallasida vosita bloki o'rnatilgan va odatda vites bilan jihozlangan moslashuvchan mil ishlatiladi. Elektr uzatish motorni boshqaradi, shuningdek, yelka harakati miqdorini aniqlaydi. Servo dvigatelning ichki sozlanishi juda oddiy: kichik DC vosita, boshqaruv davri va potentsiometrdan iborat bo'ladi. DC vosita boshqaruv g'ildiragiga vites orqali ulangan. Dvigatel aylanayotganda potentsiometrning qarshiligi o'zgaradi va boshqaruv pallasi harakat va yo'nalishni aniq sozlashi mumkin.

Mil to'g'ri (ideal) holatda bo'lganda, vosita quvvat berishni to'xtatadi. Agar mil belgilangan maqsadda to'xtamasa, vosita to'g'ri yo'nalishga kirguncha ishlaydi. Nishon pozitsiyasi elektr pulsi yordamida signal chizig'i orqali uzatiladi. Shuning uchun, vosita tezligi haqiqiy va ideal holatga mutanosibdir. Dvigatel kerakli joyga yaqinlashganda, vosita asta-sekin aylana boshlaydi, lekin vosita eng uzoqqa burilsa, tezlik juda katta bo'ladi. Boshqacha aytganda, servo motorlar faqat vazifalarni iloji boricha tezroq bajarishlari kerak, shuning uchun ular yuqori samarali uskunalar bilan ta'minlanadi.

Odatda servo dvigatellarning ikki turi mavjud: AC va DC motorlari. DC servo motorlar kichik dasturlar uchun juda mos keladi, ammo katta oqimlarga bardosh bera olmaydi. Shu bilan birga, AC servo motorlari yuqori oqimlarni yengishi mumkin va sanoat mashinalarida keng qo'llaniladi. Narx haqida gap ketganda, DC motorlari AC servolariga qaraganda arzonroq, shuning uchun ular tez-tez ishlatiladi. Bundan tashqari, DC motorlari doimiy aylanish uchun maxsus ishlab chiqilgan bo'lib, bu ularni robot harakati uchun ideal holga keltiradi.

O'zining noyob xususiyatlari bilan servo motorlar juda ko'p turli xil dasturlarda ishlatilishi mumkin. Radio orqali boshqariladigan samolyotlarda ular robotlar, liftlar, ruderlar va boshqalar kabi boshqaruv sirtlarini topish uchun ishlatiladi. Onlayn ishlab chiqarishda servo motorlar yuqori takrorlash chastotalarida qo'llaniladi va aniq ishlashni talab qiladi.

Servo dvigatel (servo motor) servo tizimdagi mexanik qismlarning ishlashini boshqaradigan va yordamchi vosita bilvosita

uzatish moslamasi bo'lgan dvigatelni anglatadi.

Servo vosita tezlikni boshqarishi mumkin va pozitsiyaning aniqligi juda aniq va boshqaruv ob'ektini boshqarish uchun kuchlanish signalini moment va tezlikka o'zgartirishi mumkin. Servo dvigatelning rotor tezligi kirish signali tomonidan boshqariladi va tezda reaksiyaga kirishishi mumkin. Avtomatik boshqaruv tizimida aktuator sifatida ishlatiladi va kichik elektromexanik vaqt doimiy, yuqori chiziqli, boshlang'ich kuchlanish va boshqalar xususiyatlariga ega va qabul qilingan elektr signalini qabul qilishi mumkin.

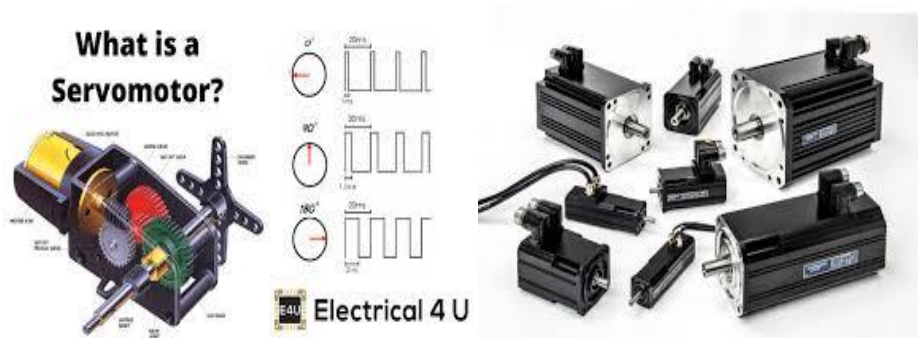
Birinchidan, servo vosita chiqish milining avtomatik oldinga va teskari yo'nalishda aylanishini qanday ta'minlash kerakligini ko'rib chiqamiz.

Shundan so'ng, biz qo'shimcha ravishda kontaktlarning zanglashiga olib boradigan potentsiometrni kiritamiz.

Kerakli tugunlar

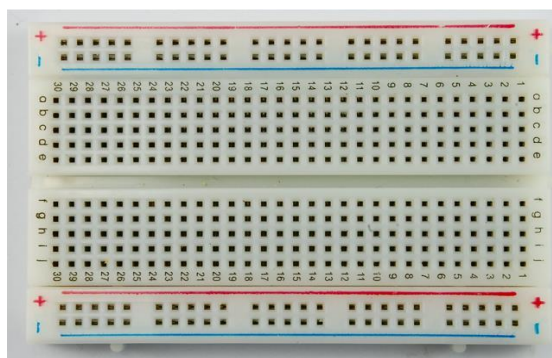
Servo drayverni boshqarish texnikasini o'zlashtirish uchun sizga kerak bo'ladi:

1 ta servomotor

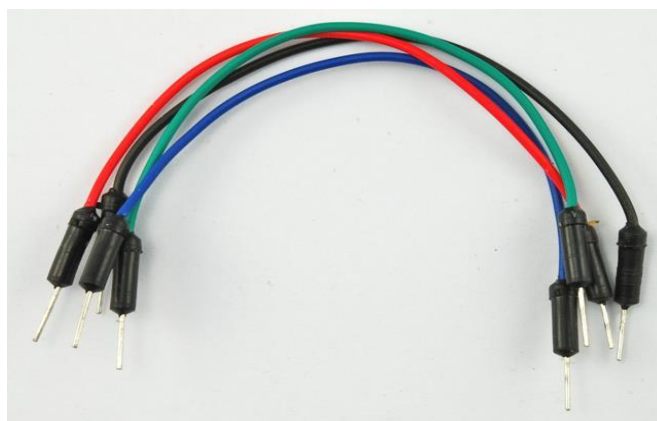


1-rasm servo motor

1 o'zgaruvchan qarshilik (potentsiometr) 10 kOm



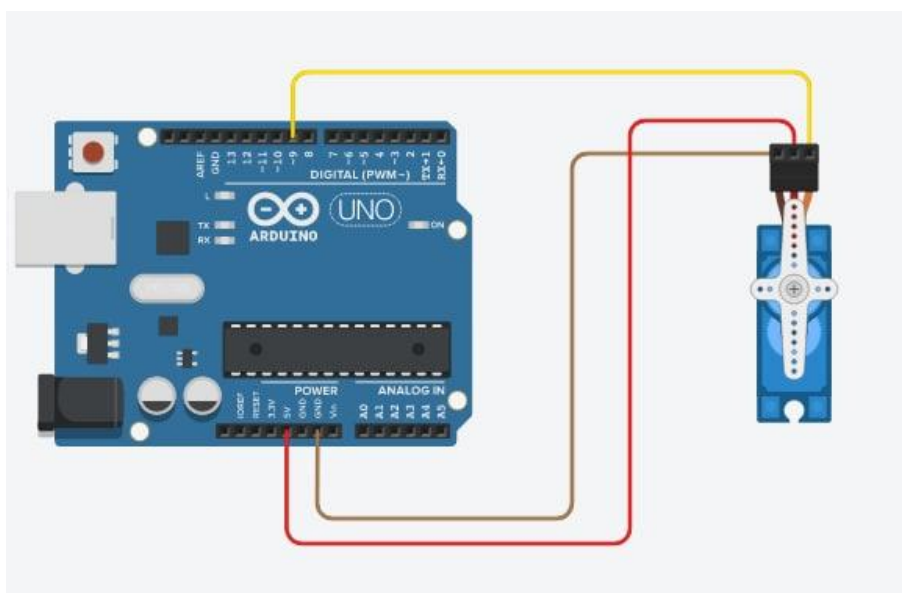
2-rasm Montaj platasi



3-rasm Father-father konnektorlari

Servomotorda 3 ta aloqa mavjud. Iplarning rangi ishlab chiqaruvchiga qarab farq qilishi mumkin, ammo qizil har doim 5V pin bo'lib, GND pin qora yoki jigarrang bo'lishi mumkin. Qolgan uchinchi aloqa servo motorning holatini boshqarish uchun ishlatiladigan signaldir. Odatda sariq rangda bo'ladi. Ushbu pinni Arduino-dagi 9-raqamli pin bilan bog'laymiz.

Servolarning pinlarida ulagichlar mavjud, ularning ichiga ulagichlarni (simlarni) o'rnatib, montaj platasiga va keyin Arduinoga ulashingiz mumkin.



4-rasm Servo matorni Arduinoga ulanishi

Dastur kodi

```
svetofor_2 §  
#include <Servo.h>  
Servo servo;  
void setup()  
{  
  servo.attach(9);  
}  
void loop()  
{  
  servo.write(0);  
  delay(1000);  
  servo.write(150);  
  delay(1000);  
}  
|
```

Nazorat uchun savollar:

1. Servomotor qanday qurilma?
2. Servoning qanday turlari bor?
3. Servomotor Arduinoga qanday ulanadi?
4. Loyihani ishga tushirish uchun qanday dastur tuzish kerak?

Displey qurilmasi. Tasvirlarni aks ettirish

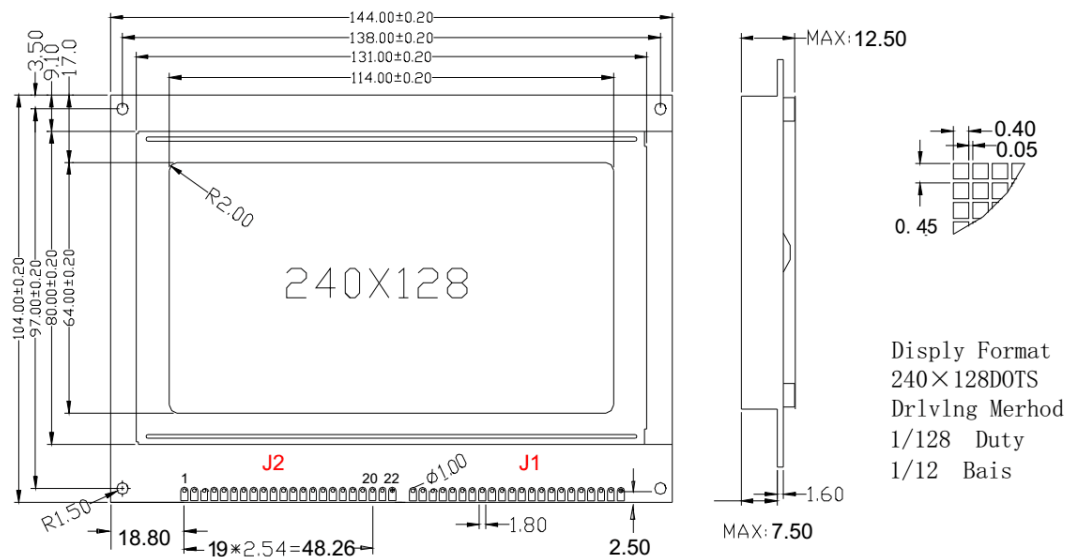
Arduino asosida prototiplarni yaratishda monitorlar har doim ham video ma'lumotni chiqarish uchun talab qilinmaydi. Ishchi diagrammalar, grafik va ramziy ma'lumotlarni ko'rsatadigan etarlicha ekranlar mavjud. Ushbu maqsadlar uchun passiv matritsa asosida ishlab chiqarilgan Arduino STN-LCD displey ishlatiladi. Ekrandagi tasvir kuchlanishning bosqichma-bosqich ta'minlanishi tufayli qurilgan, buning natijasida piksellar shaffof bo'lib, tasvir hosil bo'ladi.

STN LCD display qurilmasining tavsifi:

LCD turi va rangi	STN-sariq-yashil, STN-ko'k, FSTN	UNIT
Ko'rish burchagi	6 O'clock	
Ishlash harorati	-20 °C ---- 70 °C	
Saqlash harorati	-30 °C ---- 80 °C	
IC va paketlarni boshqarish	T6963C , COB	
Og'irligi	170 (B / L bilan)	g
Yoritish	LED	

Toki = 25 °C		TYP		MAX	UNIT
	IN				
Besleme oqimi orqa yoritishni o'z ichiga olmaydi	0		12		mA
VDD	-0.3	3.3	5.0	5.5	V
VLCD-V0 (LCD)	0		-6.8		V
Yoritish	Yon orqa	0	3.0	5.0	V
LE	yorug'lik	0		75	mA

D	Pastki orqa yorug'lik					V
						mA

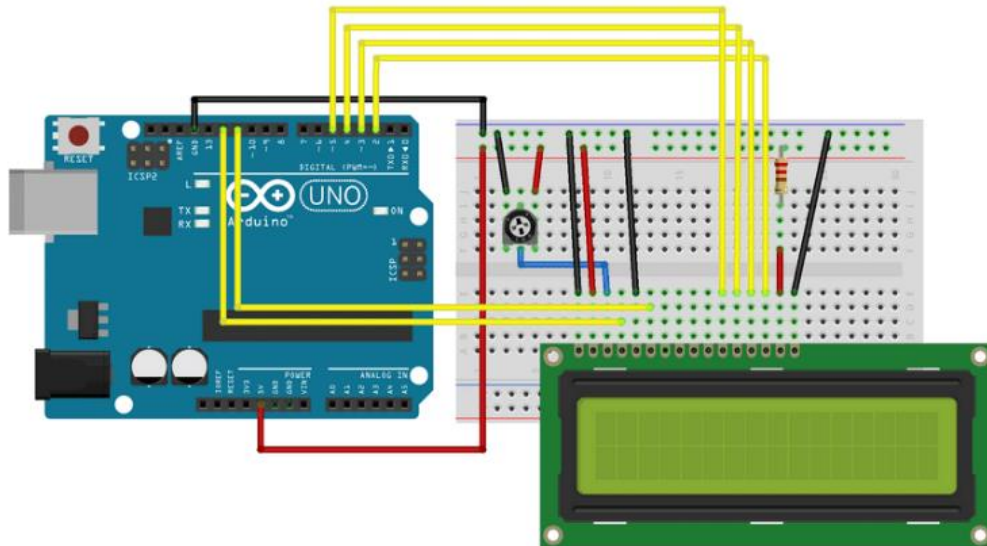


STN-LCD Arduino displeylarining asosiy afzalliklari orasida:

- past narx;
- yaxshi kontrast;
- barqaror ish;
- kam quvvat iste'moli.

Monitorlar kichik o'lchamlarga ega, chiqish ranglarini tanlash cheklangan. Uskunalar texnik ma'lumotlarni, asboblarni o'qishlarini ko'rsatish uchun ishlatiladi. Onlayn do'konimizda Arduino STN-LCD

displayni sotib olishni taklif qilamiz. Bizning katalogimizda turli o'lchamdagi, diagonal o'lchamdagi grafik va ramziy modellar mavjud. Maslahatchilar komponentlarni tanlashda yordam beradi, ishlab chiqilgan prototiplarning xususiyatlarini va ularning ishlash shartlarini hisobga oladi.



```
#include <LiquidCrystal.h>
```

```
// initialize the library with the numbers of the interface pins  
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
```

```
void setup() {  
  // set up the LCD's number of columns and rows:  
  lcd.begin(16, 2);  
  // Print a message to the LCD.  
  lcd.print("hello, world!");  
}  
  
void loop() {  
  // set the cursor to column 0, line 1
```

```
0): // (note: line 1 is the second row, since counting begins with  
    lcd.setCursor(0, 1);  
    // print the number of seconds since reset:  
    lcd.print(millis() / 1000);  
}
```

Nazorat uchun savollar:

1. Display qurilmasi nima vazifa bajaradi?
2. Bu qurilma qanday plataga yig'iladi?
3. Bu qurilmada nechta port ishlatiladi?
4. Displayni ishga tushiruvchi dastur kodlarini tahlil qiling.

Mikrokontrollerlarni o'rganish

Mikrokontroller (mikrokontroller birligi uchun MCU)-bu bitta metall oksidi-yarimo'tkazgich (MOS) vlsi integral mikrosxemasi (IC) chipidagi kichik kompyuter. Mikrokontroller bir yoki bir nechtasini o'z ichiga oladi CPU (protssessor yadrolari) xotira va dasturlashtiriladigan kirish/chiqish tashqi qurilmalari bilan birga. Ferroelektrik operativ xotira shaklida dastur xotira, na flash yoki OTP ROM ham tez-tez yonga kiritilgan, RAM, shuningdek, bir kichik miqdori. Mikrokontrollerlar ishlatilgan mikroprotssessorlardan farqli o'laroq, o'rnatilgan dasturlar uchun mo'ljallangan shaxsiy kompyuterlar yoki turli xil diskret chiplardan iborat boshqa umumiy maqsadli dasturlar.

Zamonaviy terminologiyada mikrokontroller chipdagi tizimga (SoC) o'xshash, ammo unchalik murakkab emas. SoC tashqi mikrokontroller chiplarini anakart komponentlari sifatida ulashi mumkin, ammo SoC odatda rivojlangan tashqi qurilmalarni birlashtiradi grafik ishlov berish birligi (GPU) va simsiz interfeys tekshiruvi uning ichki mikrokontroller birligi davrlari sifatida.

Mikrokontrollerlar - bu qurilmalarda, implantlarda, masofadan boshqarishda, ofislarda, uskunalarda, elektr jihozlarida, avtomatik o'chirgichlarda va boshqa qurilmalarda avtomobil dvigatellari kabi avtomatik boshqariladigan qurilmalar va qurilmalar. Alohida mikroprotssessor, xotira va kirish / chiqish dizayni bilan mikrokontrollerlar hajmi va narxini kamaytirish orqali ko'proq uskunar va jarayonni boshqarishni tejaydi. Raqamli bo'lmagan elektron qurilmalar uchun zarur bo'lgan analog komponentlarni birlashtirgan aralash signal mikrokontrollerlari keng tarqalgan. Narsalar Interneti kontekstida mikrokontrollerlar ma'lumotlarni to'plash, sezish va jismoniy dunyoni yo'ldan uzoqlashtirish uchun iqtisodiy va mashhur vositadir.

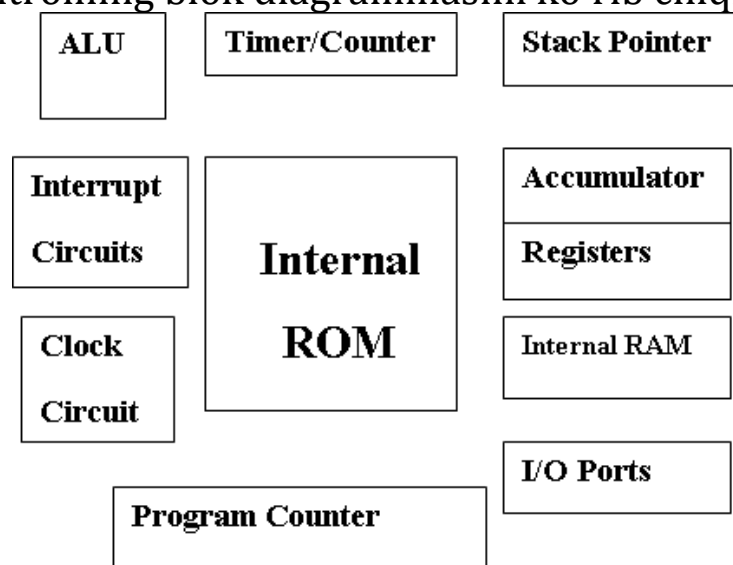
Ba'zi mikrokontrollerlar to'rt bitli so'zlardan foydalanishi va kam 4 kHz uchun 4 kHz gacha bo'lgan chastotalarda ishlashi mumkin (bitta raqamli millivat yoki mikroto'lqinli pechlar). Ular, odatda, tugmachani bosish yoki boshqa uzilish kabi hodisani kutish paytida funktsionallikni saqlab qolish qobiliyatiga ega; uxlash paytida quvvat sarfi (CPU soati va aksariyat tashqi qurilmalar

o'chirilgan) shunchaki nanovatt bo'lishi mumkin, bu ularning aksariyatini batareyaning uzoq muddatli dasturlari uchun juda mos keladi. Boshqa mikrokontrollerlar ishlash uchun muhim rollarga xizmat qilishi mumkin, bu erda ular ko'proq raqamli signal protsessori (DSP) kabi harakat qilishlari kerak, soat tezligi va quvvat sarfi yuqori.

Mikrokontroller - mikroprotsessorlarning rivojlangan versiyasi. U chip markaziy protsessorida (protsessor), faqat o'qish uchun xotira (ROM), tasodifiy kirish xotirasi (RAM), kirish / chiqish blokida, uzilishlar tekshirgichida va boshqalarda mavjud.

Shuning uchun mikrokontroller o'rnatilgan tizim ichida yuqori tezlikda signalni qayta ishlash uchun ishlatiladi. U ichki tizimni loyihalashda ishlatiladigan asosiy komponent sifatida ishlaydi.

Mikrokontrolning blok diagrammasini ko'rib chiqing:

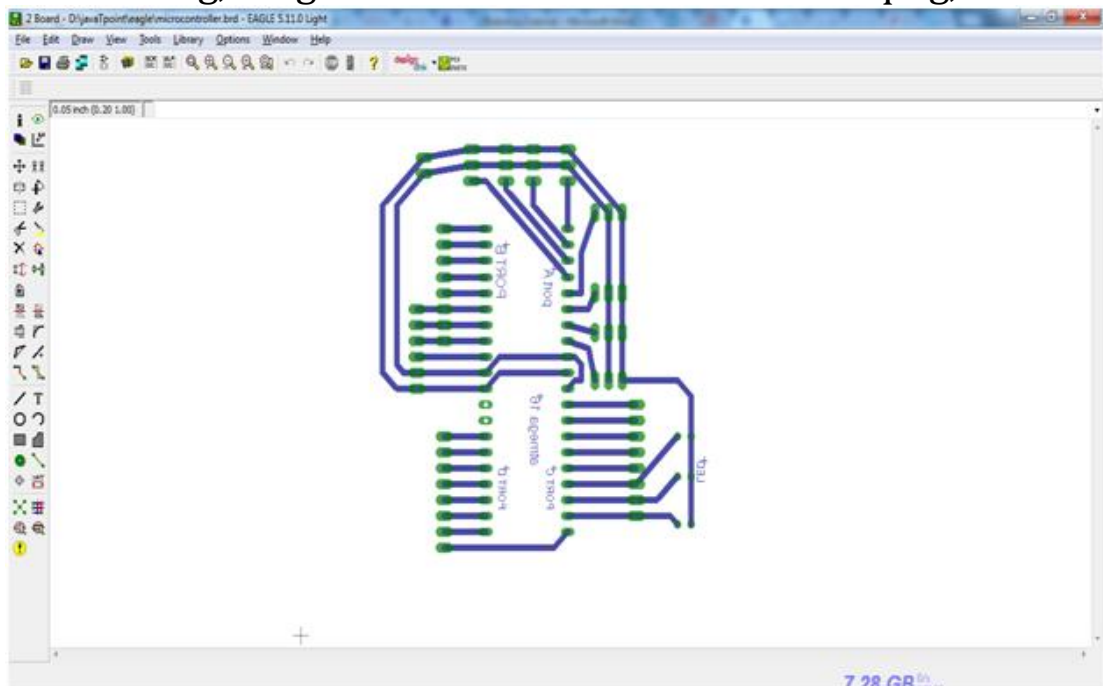


Mikrokontrollerning asosiy tarkibiy qismlari

- **Arifmetik va mantiqiy birlik (ALU)** - arifmetik va mantiqiy ishlarni bajarish uchun ishlatiladigan mikrokontroller ichidagi ALU. U registrda saqlangan ma'lumotlarga mantiqiy amalni bajaradi.
- **Akkumulyator** - akkumulyator - bu oraliq arifmetik va mantiqiy operatsiya ma'lumotlari saqlanadigan registr.
- **Ish registrlari - registrlar** - bu mikrokontroller ichidagi ma'lumotlarni turli manzil joylarida saqlash uchun ishlatiladigan saqlash moslamasi.

- **Dastur hisoblagichi** - dastur hisoblagichi mikrokontroller ichida bajarilgan dastur sonini hisoblash uchun ishlatiladi.
- **Stek ko'rsatkichi** - Stack ko'rsatkichi ma'lum manzilga ko'rsatkich vazifasini bajaradi. Bu protsessor tomonidan tayyorlangan dasturning so'nggi so'rovi manzilini stek ichida saqlash uchun ishlatiladigan registr.
- **Soat davri** - Soat davri mikrokontroller uchun mos yozuvlar signali sifatida zarur bo'lgan soat impulsini yaratish uchun ishlatiladi.
- **Uzilish davri** - uzilish davri, mikrokontroller tomonidan birinchi navbatda birinchi navbatda xizmat qilishi kerak bo'lgan ustuvor jarayon talab qilinganda uzilish signalini hosil qilish uchun ishlatiladi.
- **Ichki ROM** - Ichki ROM faqat o'rnatilgan tizimda ma'lumotni saqlash uchun ishlatiladigan xotirani o'qiydi. Bu ko'rsatma va ma'lumotlarni mikrokontroller ichida saqlash uchun asosiy xotira vazifasini bajaradi.
- **I / U portlari** - I / U portlari sensor, klaviatura va boshqalar kabi kirish moslamalarini kirish portlari bilan va LCD, buzzer va boshqalar kabi chiqish moslamalarini mikrokontrollerda mavjud bo'lgan chiqish portlari bilan ulash uchun ishlatiladi.

Mikrokontroller va sarlavha terminallari orasidagi marshrutni tugatgandan so'ng, tenglikni taxtasi tartibini ko'rib chiqing,



Taxta tartibi tugatgandan so'ng, u PDF-ni yaratish uchun ishlaydi. Tuzilgan PDF-dan yuqoridagi maketning nusxasini Over Head Projektor varag'iga (OHP varaqlari deb nomlanuvchi) yoki PCB kartasini loyihalash uchun mum qog'ozidan olish uchun foydalaniladi.

Nazorat uchun savollar:

1. Mikrokontrellar qanday qurilma?
2. Ular nima vazifa bajaradi?
3. Mikrokontrellar nimalardan tashkil topadi?

Arduino va kompyuter bilan aloqa usullari.

Ishdan maqsad: Arduino haqida tushunchaga va amaliy ko'nikmaga ega bo'lish va uni kompyuterga bog'lash usullarini o'rganish.

Nazariy ma'lumotlar

Arduino - bu o'ziga xos protsessor va xotiraga ega bo'lgan kichik plata. Arduino - bu professional bo'lmagan foydalanuvchilarga mo'ljallangan oddiy avtomatlashtirish va robototexnika tizimlarini yaratish uchun mo'ljallangan apparat va dasturiy ta'minot brendidir.

Dasturiy ta'minot qismi dasturlarni yozish, kompilyatsiya qilish va dasturiy ta'minot uchun bepul dasturiy qobiqdan (IDE) iborat. Uskuna qismi rasmiy ishlab chiqaruvchilar tomonidan ham, uchinchi tomon ishlab chiqaruvchilari tomonidan sotiladigan oldindan yig'ilgan bosma elektron platalar to'plamidir. To'liq ochiq tizim arxitekturasini sizga Arduino mahsulot qatoriga bimalol nusxa ko'chirish yoki qo'shish imkoniyatini beradi.

Arduinodan avtomatlashtirilgan ob'ektlarni yaratish yoki standart simli va simsiz interfeyslar orqali kompyuterda dasturiy ta'minotga ulanish uchun foydalanish mumkin.

Arduino kontseptsiyasi uy-joy yoki o'rnatish tuzilishini o'z ichiga olmaydi. Ishlab chiquvchi mustaqil ravishda yoki uchinchi tomon kompaniyalari yordamida platalarni o'rnatish va mexanik himoya usulini tanlaydi. Uchinchi tomon sotuvchilari, shuningdek, Arduino platalari bilan ishlashga mo'ljallangan robot elektromexanika to'plamlarini ham ishlab chiqaradilar. Mustaqil ishlab chiqaruvchilar shuningdek Arduino -ga ko'proq yoki kamroq mos keladigan har xil turdagi sensorlar va aktuatorlarning keng turlarini ishlab chiqaradilar.



Arduino uchun mikrokontrollerlar oldindan tikilgan bootloader (inglizcha) mavjudligi bilan ajralib turadi. bootloader). Ushbu bootloader yordamida foydalanuvchi an'anaviy individual apparat dasturchilaridan foydalanmasdan o'z dasturini mikrokontrollerga yuklaydi. Bootloader kompyuterga USB interfeysi (agar u kartada bo'lsa) yoki alohida UART-USB adapteri orqali ulanadi. Bootloader yordami Arduino IDE -ga o'rnatilgan va bir marta bosish bilan amalga oshiriladi.

Bootloader yoki mikrokontrollerni bootloader -dan sotib olmagan holda, ishlab chiquvchilar bootloader ni mikrokontrollerga o'z-o'zidan yoqish imkoniyatini beradi. Buni amalga oshirish uchun Arduino IDE -da bir nechta mashhur arzon dasturchilarni qo'llab-quvvatlaydi va ko'pchilik Arduino platalarida ichki dasturlash uchun ulagich mavjud (AVR, JTAG yoki SWD[en] ARM uchun ICSP). "#Mikrokontroller dasturini yuklab olish " bo'limiga o'ting» Asta-sekin, ARM protsessorlari platalar qatorida paydo bo'ldi. Dastlab AT91SAM3X8E klassik konstruktsion kengashda ("Due"). Keyinchalik SAMD21 (Cortex-M0, 48 MHz, 256k Flash, 32k RAM) tekshirgichi bilan jihozlangan DIP dizaynida Arduino «MKR» platalari paydo bo'ldi. 2020 dan beri ARM Cortex-M7 (STM32H747 @ 480 MGts) bilan «Portenta» modullari bir xil "MKR" dizaynida paydo bo'ldi.

ESP8266

Uchinchi tomon ishlab chiquvchilari mashhur ESP8266 Wi-Fi mikrokontrolerini Arduino -ga ko'chirishdi. Endi kompilyatsiya va Wi-Fi tarmog'i qo'llab-quvvatlash bilan birga, Arduino IDE to'g'ridan to'g'ri tutashishi mumkin, ularning eskizlari va Wi-Fi qo'llab-quvvatlash bilan ESP8266 uchun proshivka qilinadi.

Intel x86 Uchinchi tomon ishlab chiqaruvchilari bilan hamkorlik doirasida Arduino IDE ba'zi Intel x86 apparatlarini qo'llab-quvvatlaydi. Intel Galileo (ingliz tili.) rus. Intel Quark X1000 400 MGts protsessor, Intel Edison (ingliz tili.) rus. Arduino 101[10] - Intel x86 arxitekturasida bo'yicha Arduino -mos keluvchi platalar. Platalar Arduino atrof-muhit platalari bilan mexanik va elektrga mos keladi. Kengashlar o'z Linux operatsion tizimida ishlaydi, uning ustida Arduino sketchlarini yuklab olish va ijro etish imkonini beruvchi dastur ishlaydi.

Dasturlash tili.

Arduino dasturlash tili - bu standart C++ (GNU Compiler Collection kompilyatorlari oilasining kompilyatori ishlatiladi), bu ba'zi funktsiyalar bilan yangi boshlanuvchilar uchun birinchi ish dasturini yozishni osonlashtiradi.



Dasturchi Arduino tomonidan yozilgan dasturlarga eskizlar (yoki ba'zan eskizlar — ingliz tilidan tarjima qilinganda sketch deyiladi) va *.ino kengaytmali fayllarda saqlanadi. Ushbu fayllar kompilyatsiya oldin Arduino preprocessor tomonidan

qayta ishlanadi. Bundan tashqari, standart C++ fayllarini loyihaga yaratish va ulash mumkin.

C++ dagi `main()` funksiyasini Arduino preprocessor yaratadi, joylashtirish uchun zarur bo'lgan "chernovoy" harakatlar.

Dasturchi Arduino uchun ikkita majburiy `setup()` va `loop()` yozishi kerak. Birinchisi boshida chaqiriladi, ikkinchisi esa cheksiz siklda amalga oshiriladi.

Dastur matnida (`sketch`) dasturchi tomonidan ishlatiladigan standart kutubxonalarning header fayllarini kiritish shart emas. Ushbu header fayllari loyiha konfiguratsiyasiga muvofiq Arduino protsessorini qo'shadi. Biroq, maxsus kutubxonalar ko'rsatilishi kerak.

Arduino IDE loyihasi menejeri kutubxonalarni qo'shish uchun nostandart mexanizmga ega. Standart C++ da manba matnlari shaklidagi kutubxonalar IDE ish katalogidagi maxsus papkaga qo'shiladi. Shu bilan birga, kutubxona nomi IDE menyusidagi kutubxonalar ro'yxatiga qo'shiladi. Dasturchi kerakli kutubxonalarni aniqlaydi va ularni kompilyatsiya ro'yxatiga kiritadi.

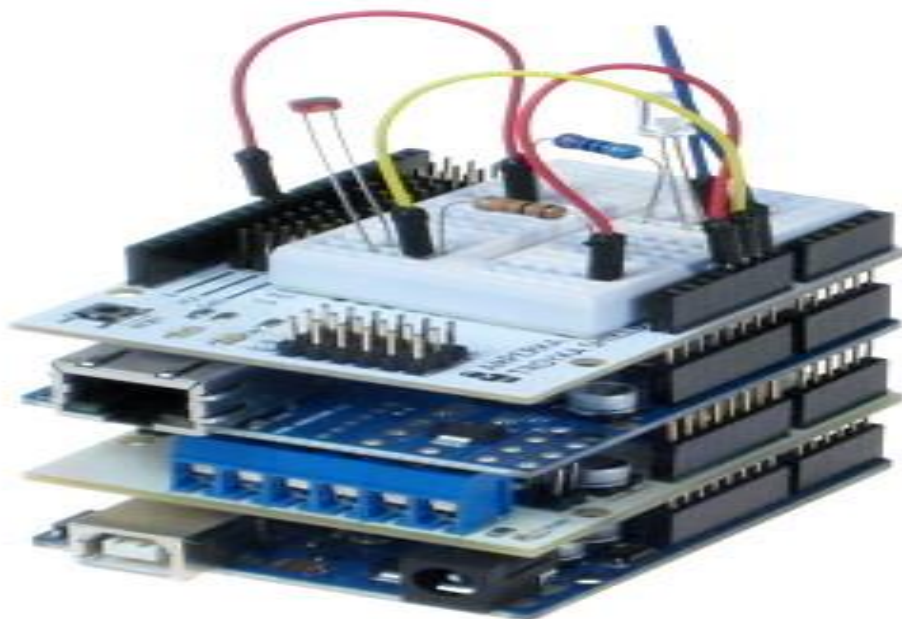
Arduino IDE hech qanday kompilyator sozlamalarini taklif qilmaydi va boshqa sozlamalarni kamaytiradi, bu esa yangi boshlovchilar uchun ishlashni osonlashtiradi va muammolar xavfini kamaytiradi.

Shunday qilib, eng oddiy dasturning to'liq matni (`Sketch`) svetodiod ("pin") Arduino ning 13 soniyasiga, 2 soniyasiga (yarim oraliq, ya'ni 1 soniya svetodiod yonadi, yarim oraliq yoqilmaydi).

```
void setup () {  
    pinMode (13, OUTPUT); // 13 portni chiqish porti sifatida  
    belgilash  
}  
void loop () {
```

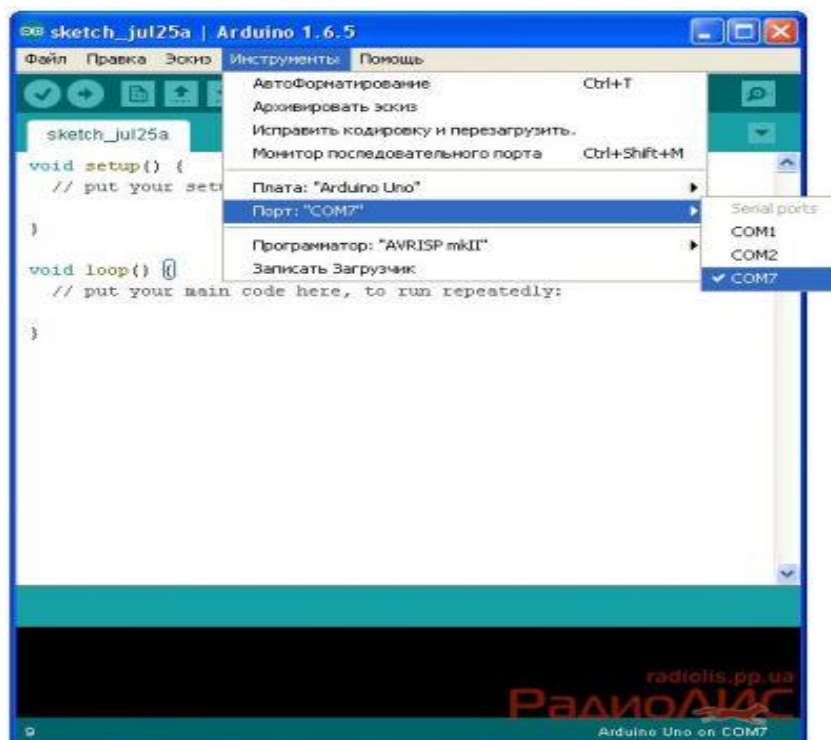
```
digitalWrite (13, HIGH); // 13 portni "1" holatiga o'rnatish,  
svetodiod yonadi  
delay (1000); // 1000 millisekundga kechikish  
digitalWrite (13, LOW); // 13 portni "0" holatiga o'rnatish,  
svetodiod o'chadi  
delay (1000); // 1000 millisekundga kechikish  
}
```

Buterburut usuli.



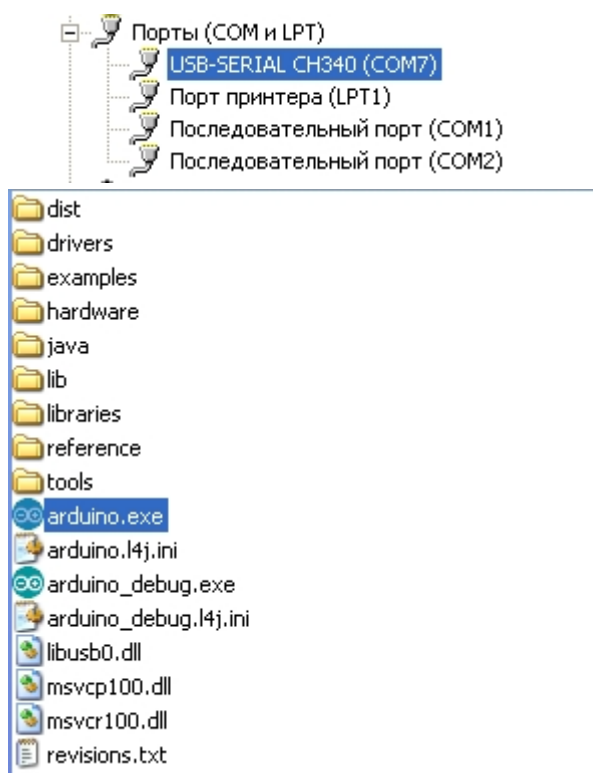
Arduino IDE-ni o'rnatishda asl Arduino platalari bilan ishlash uchun zarur bo'lgan drayverlar avtomatik ravishda o'rnatilishi kerak. Ammo hiyla shundaki, Xitoy taxtalarida arzonroq USB interfeysi kontrolleri mavjud, u bilan ishlash uchun maxsus haydovchi kerak. Xitoylik tekshirgich bundan ham yomon emas, shunchaki arzonroq \u003d)

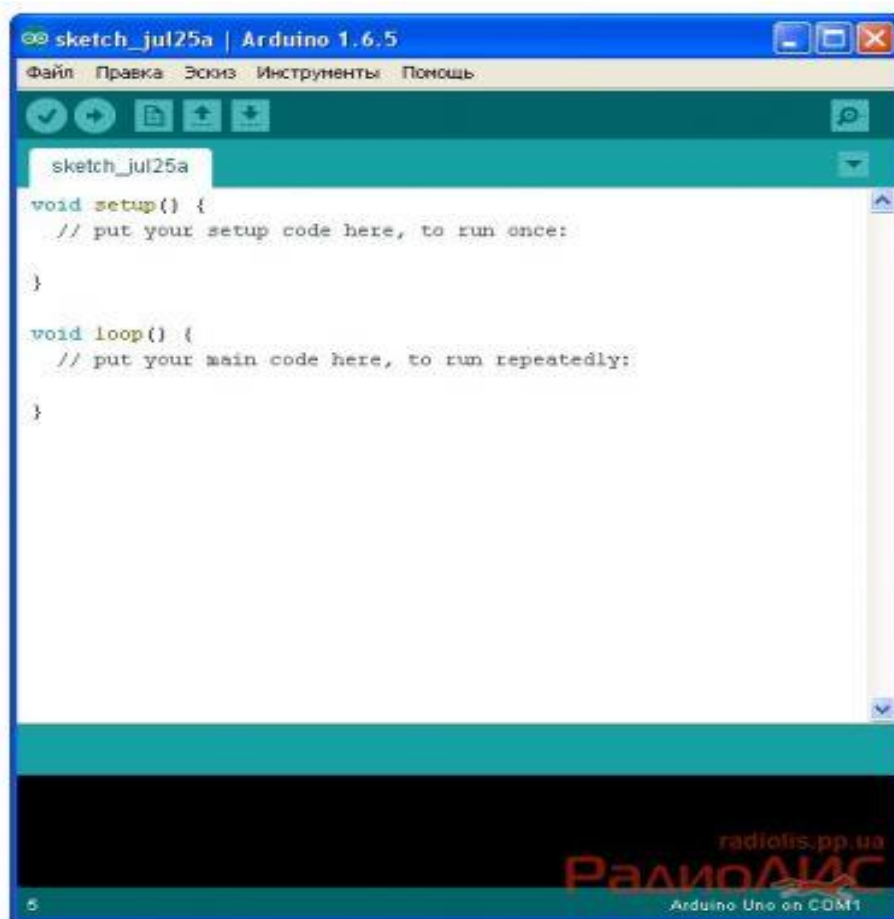
Agar biron sababga ko'ra siz Arduinodan drayverlarni o'rnatmagan bo'lsangiz, ularni dastur papkasidan qo'lda



o'rnatishingiz mumkin.

Siz Windows uchun CH341 drayverini yuklab olishingiz yoki o'zingiz google-ga o'rnatishingiz mumkin.

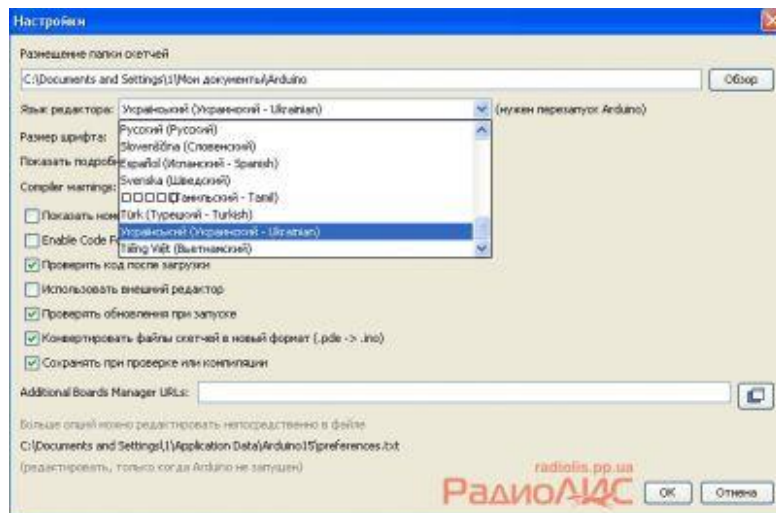




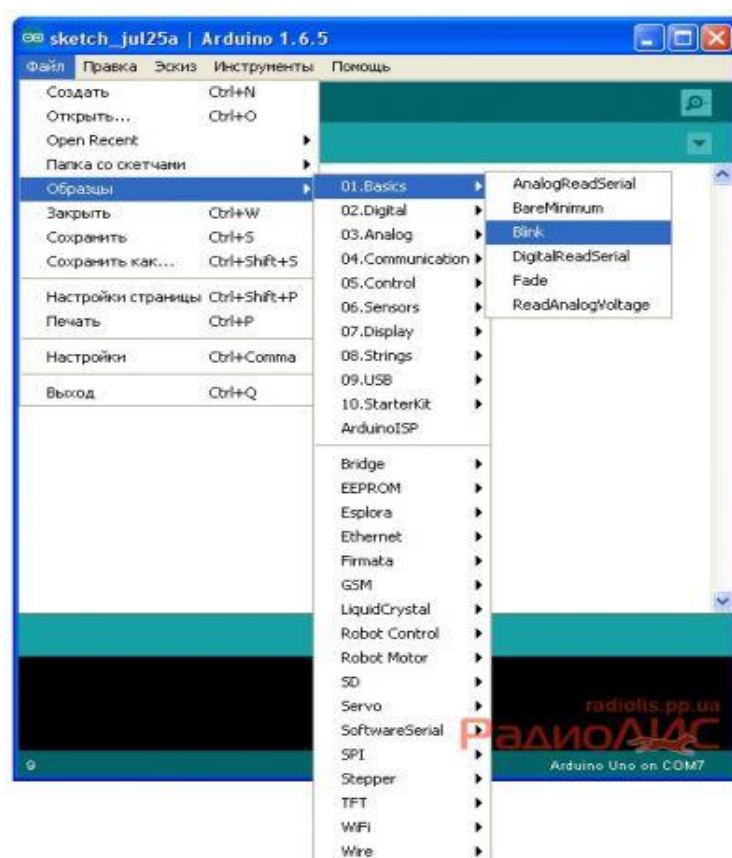
Arduino dasturining ish muhiti



Arduino UNO platasi



Arduino dasturini kompyuter bilan bog'lash



Arduino dasturi bazasidagi tayyor loyihalarni ishga tushirish

Nazorat uchun savollar:

1. Arduino va kompyuter aloqasi qanday amalga oshiriladi?
2. Tayyor loyihalarni qanday ishga tushirish mumkin?
3. Yig'ilgan loyihalar Arduinoga qanday ulanadi?

Arduino asosidagi harakatlanuvchi platforma.

Ishdan maqsad: Arduino asosida harakatlanuvchi platformani o'rganish va unda ko'nikma hosil qilish.

Nazariy ma'lumot

Mikrokontroller	<u>ATmega</u> <u>328P</u>
Ishlash kuchlanishi	5V
Kirish kuchlanishi (tavsiya etiladi)	7-12V
Kirish kuchlanishi (chegara)	6-20V
Raqamli I / U pinlari	14 (shundan 6 tasi PWM chiqishini ta'minlaydi)
PWM raqamli I / U pinlari	6
\Analog kirish pinlari	6
I / U piniga doimiy oqim	20 mA
3.3V pin uchun doimiy oqim	50 mA
Flash xotira	32 KB (ATmega328P), undan 0,5 KB yuklovchi tomonidan ishlatiladi
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Soat tezligi	16 MGts
LED_BUILTIN	13
Uzunlik	68,6 mm
Kengligi	53,4 mm
Og'irligi	25 g

L293D divigatel haydovchisi



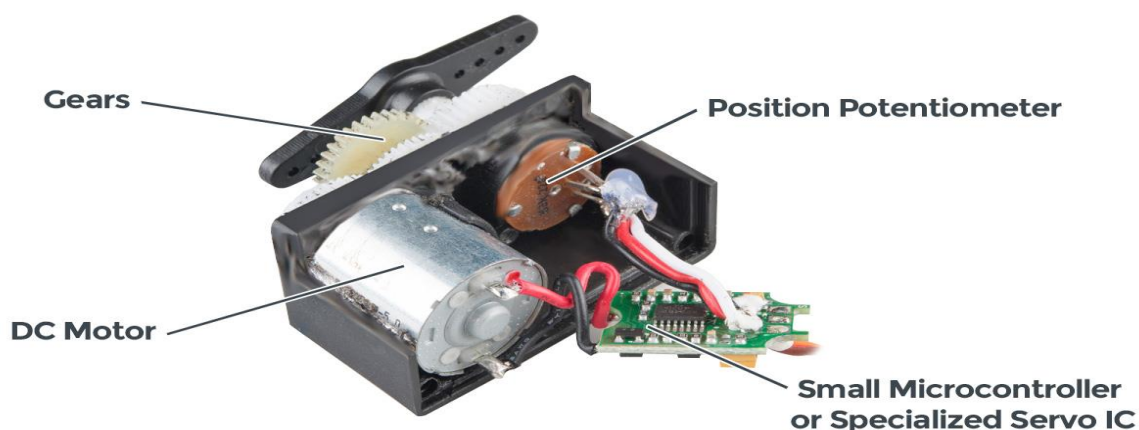
L293D chipining xususiyatlari

- vosita ta'minot kuchlanishi (V_s) - 4,5 ... 36V
 - mikroxemalarni etkazib berish kuchlanishi (V_{ss}) - 5V
 - ruxsat etilgan yuk oqimi - 600mA (har bir kanal uchun)
 - yuqori (maksimal) chiqish oqimi - 1,2A (har bir kanal uchun)
 - mantiqiy "0" kirish kuchlanishi - 1,5 V gacha
 - mantiqiy "1" kirish kuchlanishi - 2,3 ... 7V
 - 5 kHz gacha tezlikni almashtirish.
- qizib ketishdan himoya qilish

Servo haqida

Servomehanizm (servo) ko'pchilik taxmin qilganidan ancha uzoqroq bo'lgan bir nechta turli xil mashinalarni nazarda tutishi mumkin. Aslida, servo - bu teskari aloqa elementi o'rnatilgan har qanday dvigatel bilan ishlaydigan tizim. Servolar og'ir texnikadan tortib, avtomashinalarda boshqarish, robototexnika va turli xil elektronikalarda mavjud.

Agar siz sevimli mashg'ulotlariga mo'ljallangan servo motorni ochsangiz, deyarli har doim uchta yadroli komponentni topasiz: doimiy dvigatel, boshqaruvchi sxemasi va potansiyometr yoki shunga o'xshash qayta aloqa mexanizmi. Dvigatelning tezligi va momentini oshirish uchun doimiy dvigatel vites qutisi va chiqish valiga biriktirilgan. DC motor chiqish milini harakatga keltiradi. Nazoratchi sxemasi tekshirgich tomonidan yuborilgan signallarni izohlaydi va potansiyometr chiqish milining holatini kuzatib borish uchun boshqaruvchi sxemasi uchun teskari aloqa vazifasini bajaradi. Sevimli mashg'ulotlarning deyarli barcha servolari servoni quvvatlantirish va boshqarish uchun standart uch pinli, 0,1 dyuymli ulagichga ega. Ranglarni kodlash markalar orasida farq qilishi mumkin, ammo pinlar deyarli bir xil tartibda. Birgalikda siz faqat uchta sim yordamida quvvat milini yo'nalishini, tezligini va holatini boshqarishingiz mumkin.

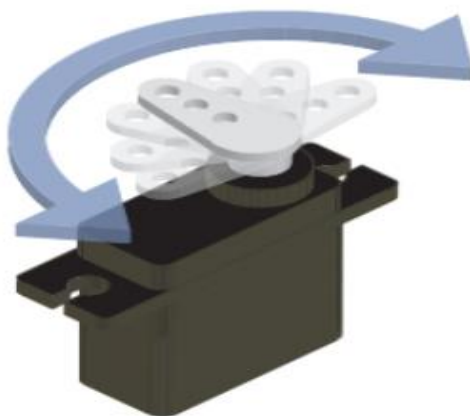


3-rasm. *Standart Servo ichki tuzulishi.*

Servoni harakatlanish yoyi bo'ylab bir holatga o'tkazish yoki doimiy aylanishli servolarda dvigatelning tezligi va yo'nalishi bo'yicha boshqarish uchun servo talqin qilishi uchun aniq vaqt signalini yuborishi kerak. Odatda, sevimli mashg'ulot servolari 20 daqiqada bir puls ko'rishni kutishadi va bu signalning kengligi pozitsiyani belgilaydi. Ushbu kenglik odatda bir va ikki millisekund orasida. Ushbu turdagi signallarni boshqarish tez-tez PWM deb qisqartirilgan Pulse Width Modulation deb nomlanadi. Servo kontroller odatda servo uchun boshqaruv signalini o'rnatish uchun joystick, potansiyometr yoki sensorning teskari aloqasi kabi boshqa tarkibiy qismlardan kirishni oladigan maxsus jihoz bo'limi bo'ladi. Boshqarishning boshqa variantlari orasida ushbu signalni to'g'ridan-to'g'ri servoga yuborish uchun mikrokontrollerdagi PWM-ga mos keladigan pinlardan foydalanish kiradi.

Servoingizning kattaligi va tork chiqishiga qarab, kirish voltaji o'zgaradi, lekin sevimli mashg'ulot servolarining ko'pi sizning afzal ko'rgan mikrokontrolleringiz yoki batareyangiz zanjiridan 5V bilan yaxshi ishlaydi. Kuchlanishdan ko'ra muhimroq - bu harakatlanayotganda va biriktirilgan yuk bilan servoning tortishi mumkin bo'lgan oqim chizig'i. Bo'shatish paytida odatdagi sevimli mashg'ulot servolari 10mA dan kam tortishi mumkin, ammo yuk ostida bo'lgan katta servolar Amperdan va undan ko'pini tortishi mumkin. Elektr ta'minotingiz mos voltaj diapazoniga ega ekanligiga ishonch hosil qilish uchun siz ishlatmoqchi bo'lgan servoning texnik xususiyatlarini tekshirish va yukni biriktirilgan holda servoni siljitish uchun etarli oqimni etkazib berish muhimdir.

Standart yoki "yopiq tsikli" servo 90 yoki 180 daraja harakatlanish diapazoniga ega bo'ladi, ba'zilar belgilangan diapazondan biroz kattaroq yoki kichikroq bo'ladi, shuning uchun siz o'zingizning loyihangizda ishlatishdan oldin motorning xususiyatlarini tekshiring. 1. Standart servolar boshqaruv pultidagi simob ustidagi harakatlanish kamoniidagi holatini kuzatishi uchun tekshiruvchiga teskari aloqani ta'minlaydi. Bu sizga servoni boshqaruvchidan to'g'ri impuls uzunligi bilan aniq joylarga ko'chirishga imkon beradi.



4-rasm. *Standart Servo.*

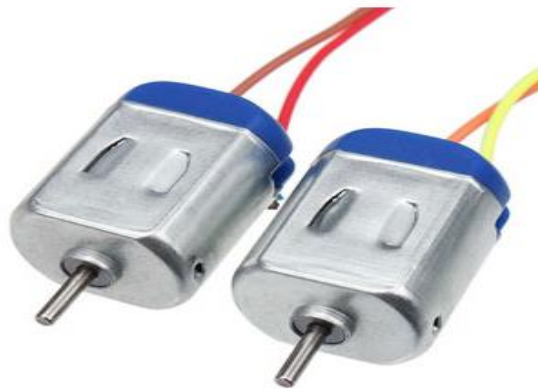
Doimiy aylanish yoki "ochiq halqa" servolari odatdagi servo singari ishlamaydi. Boshqarish signali faqat yo'nalishni va tezlikni emas, balki holatni boshqaradi. pozitsiyani boshqarish uchun hech qanday teskari aloqa yo'q, shuning uchun ular aylanish kamoniidagi aniq nuqtalar o'rtasida harakatlanishni talab qiladigan ilovalar uchun tavsiya etilmaydi

DC motor

Motorlar ko'pincha Arduino loyihalarida qo'llaniladi. Ular g'ildiraklarni boshqaradi, burama pervanellarni aylantiradi, sanoat robotlarining qo'llarini aylantiradi va 3D-printerni tashiydi.

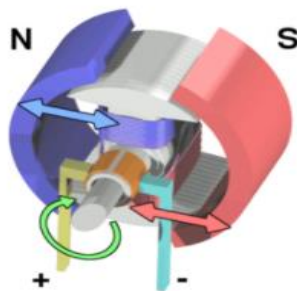
Har bir yangi robototexnika muhandisi dvigatelni mikrokontrollerga ulash muammosiga duch keladi. Dvigatelda faqat ikkita pin bor va u dvigatelni Arduinoning raqamli pimlariga ulab, so'ngra dastur orqali yoqish va o'chirish mumkin. Ko'pincha har xil o'yinchoqlarda ishlatiladigan kichik dvigatel ham ishlash uchun 200

mA dan 1 Ampergacha bo'lgan oqimni talab qiladi. Va Arduino-ning raqamli chiqishi bizga faqat 20 mA quvvat berishi mumkin. Odatda pervanel bilan starter to'plamiga kiritilgan doimiy dvigatel bilan ishlaylik.



5-rasm. DC motor.

Ikki kutupli rotorli (armatura) va doimiy magnitlangan statorli elektr motorining ishi. "N" va "S" magnitlarning ichki o'qi yuzlaridagi qutblanishlarni belgilaydi; tashqi yuzlari qarama-qarshi qutblarga ega. + Va - belgilari armatura bobinlariga tok etkazib beradigan komutatorga doimiy tokning qaerda qo'llanilishini ko'rsatadi



6-rasm. DC motor ishlash tartibi.

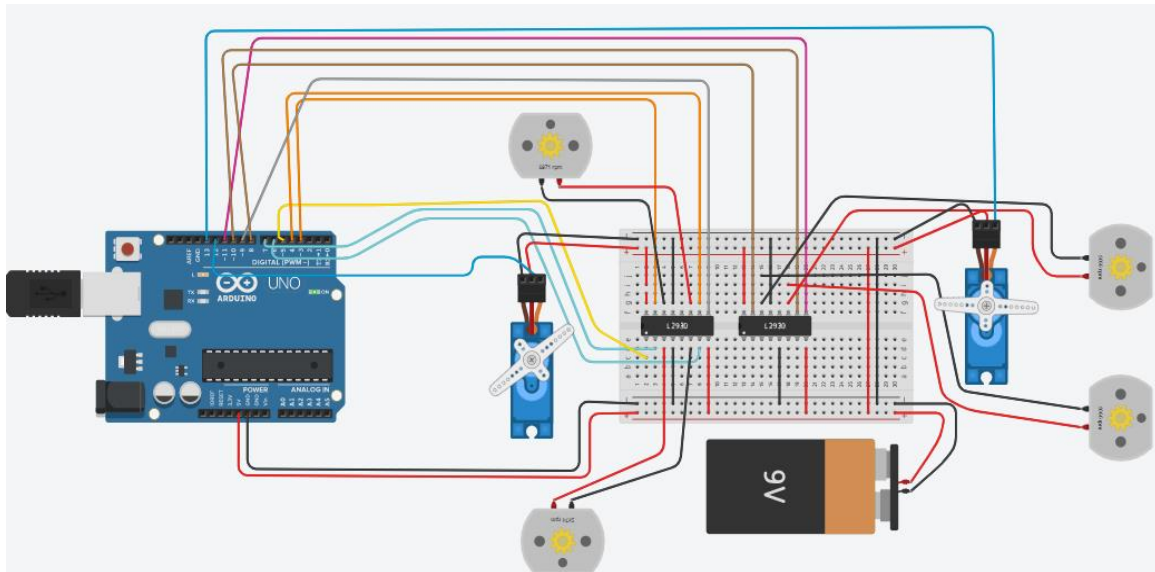
Har xil turdagi motorlarni boshqarishga imkon beradigan ko'plab tayyor mikrosxemalar mavjud. Biz L293D drayverini ko'rib chiqamiz.

Ishni bajarish:

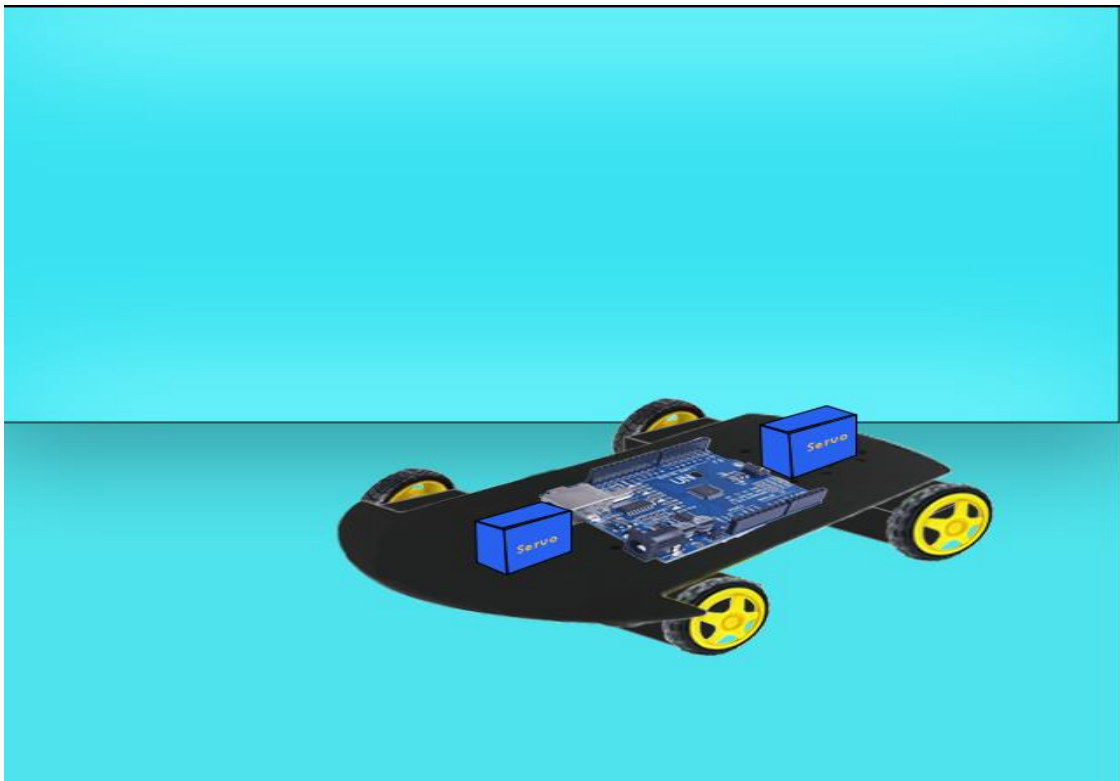
Kerakli qurilmalar:

1. Arduino Uno.
2. Breadbord (paykasiz maket)
3. 2 ta Drayver L293D

4. 4 ta DC motor.
5. 2 ta Servo



7-rasm. Ulanish sxemasi



8-rasm. Transport taxminiy korinishi.

Dastur kodi

```
#include <Servo.h>
int hozirdagi_gradus;
```

```
int ung=135;
int chap=45;
int utish;
int En1=9;
int in1=4;
int in2=3;
int En2=5;
int in3=7;
int in4=6;
int En3=11;
int in5=10;
int in6=8;
Servo servo1,servo2;
void setup()
{
  servo1.attach(12);
  servo2.attach(13);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in5, OUTPUT);
  pinMode(in6, OUTPUT);
  pinMode(En1, OUTPUT);
  pinMode(En2, OUTPUT);
  pinMode(En3, OUTPUT);
```

```

servo1.write(90);
servo2.write(90);
delay(550);
  digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, LOW);
  digitalWrite(in5, LOW);
digitalWrite(in6, LOW);
  analogWrite(En1, 255);
analogWrite(En2, 255);
  analogWrite(En3, 255);
}
void loop()
{
  oldi_servo1_bilan_ung_burulish();
  oldi_servo1_bilan_chap_burulish();
  orqa_servo2_bilan_ung_burulish();
  orqa_servo2_bilan_chap_burulish();
}
void togriga_Harakat(){
  servo1.write(90);
  servo2.write(90);
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
  digitalWrite(in3, HIGH);

```

```

digitalWrite(in4, LOW);
digitalWrite(in5, HIGH);
digitalWrite(in6, LOW);
analogWrite(En1,255);
analogWrite(En2, 255);
analogWrite(En3, 255);
delay(1000);
}
void oldi_servo1_bilan_ung_burulish(){
    analogWrite(En1,150);
    analogWrite(En2, 150);
    analogWrite(En3, 150);
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
    digitalWrite(in5, HIGH);
    digitalWrite(in6, LOW);
    hozirdagi_gradus=servo1.read();
    for(int p =hozirdagi_gradus+1 ; p<ung; p++ ){
        servo1.write(p);
        delay(7);
    }
    delay(500);
    togriga_Harakat();
}

```

```

void oldi_servo1_bilan_chap_burulish(){
    analogWrite(En1,150);
    analogWrite(En2, 150);
    analogWrite(En3, 150);
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
    digitalWrite(in5, HIGH);
    digitalWrite(in6, LOW);
    hozirdagi_gradus=servo1.read();
    for(int p =hozirdagi_gradus-1 ; p>=chap; p-- ){
        servo1.write(p);
        delay(7);
    }
    delay(500);
    togriga_Harakat();
}

void orqa_servo2_bilan_ung_burulish(){
    analogWrite(En1,100);
    analogWrite(En2, 100);
    analogWrite(En3, 150);
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);

```

```

digitalWrite(in5, HIGH);
digitalWrite(in6, LOW);
hozirdagi_gradus=servo2.read();
for(int p =hozirdagi_gradus-1 ; p>=chap; p-- ){
    servo2.write(p);
    delay(7);
}
delay(500);
togriga_Harakat();
}

void orqa_servo2_bilan_chap_burulish(){
    analogWrite(En1,100);
    analogWrite(En2, 100);
    analogWrite(En3, 150);
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
    digitalWrite(in5, HIGH);
    digitalWrite(in6, LOW);
    hozirdagi_gradus=servo2.read();
    for(int p =hozirdagi_gradus+1 ; p<ung; p++){
        servo2.write(p);
        delay(7); }
    delay(500);
    togriga_Harakat(); }

```

Robototexnik qurilmalarni dasturlash.

Ishdan maqsad: Robototexnik qurilmalarni arduino UNO yordamida dasturlashning amaliy ko'nikmalarini hosil qilish.

Nazariy ma'lumotlar

Bugungi kunda biz PIR sensorini (harakatni) Arduino ga ulash loyihasini tahlil qilamiz va sensorni ishga tushirganda elektron pochtani avtomatik ravishda yuborishni tashkil qilamiz. Arduino bu loyihaning qalbi — IQ sensori o'qishlarini o'qiydi va harakat aniqlanganda kompyuterni USB port orqali elektron pochta orqali yuboradi. Kompyuterga keladigan signalni qayta ishlash Python dasturi yordamida amalga oshiriladi.

Modelni yig'ish uchun ehtiyot qismlar ro'yxati

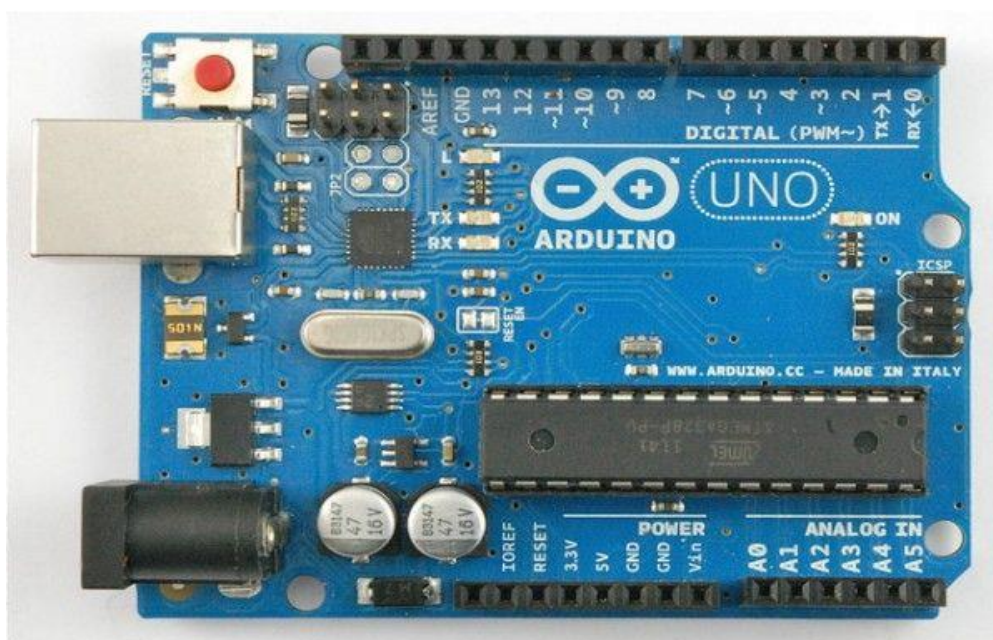
Ushbu darsda tasvirlangan loyihani yig'ish uchun sizga quyidagi qismlar kerak bo'ladi:

Arduino UNO yoki analog (Arduino ni qanday tanlash haqida batafsil ma'lumot);

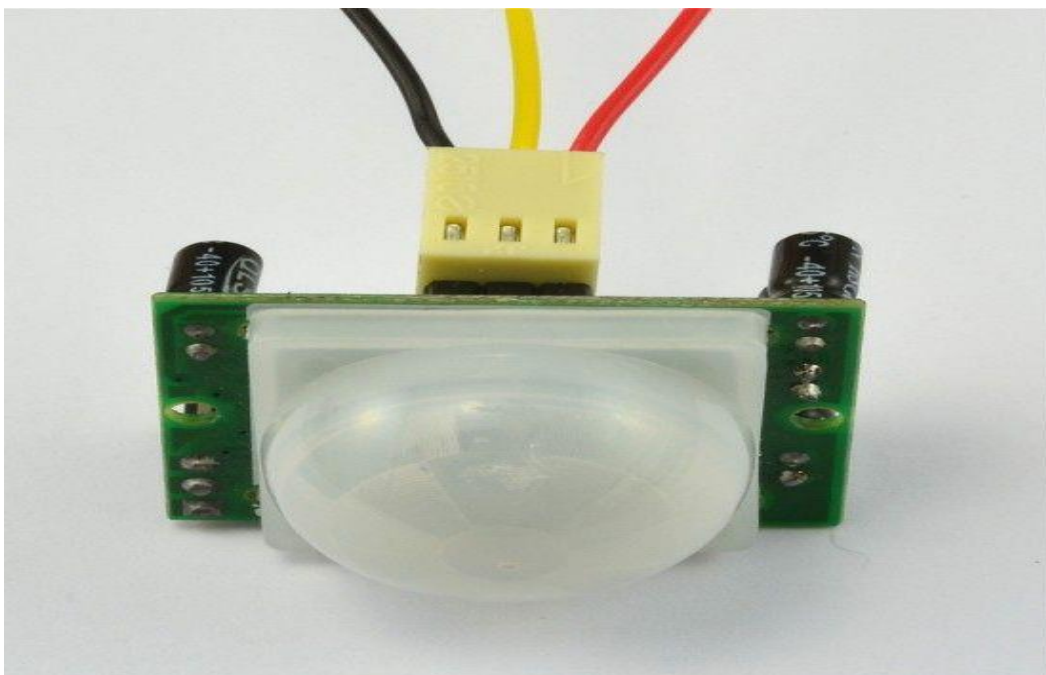
PIR-sensor (\$2 uchun mos);

breadboard (\$2,4 uchun sotib olish mumkin);

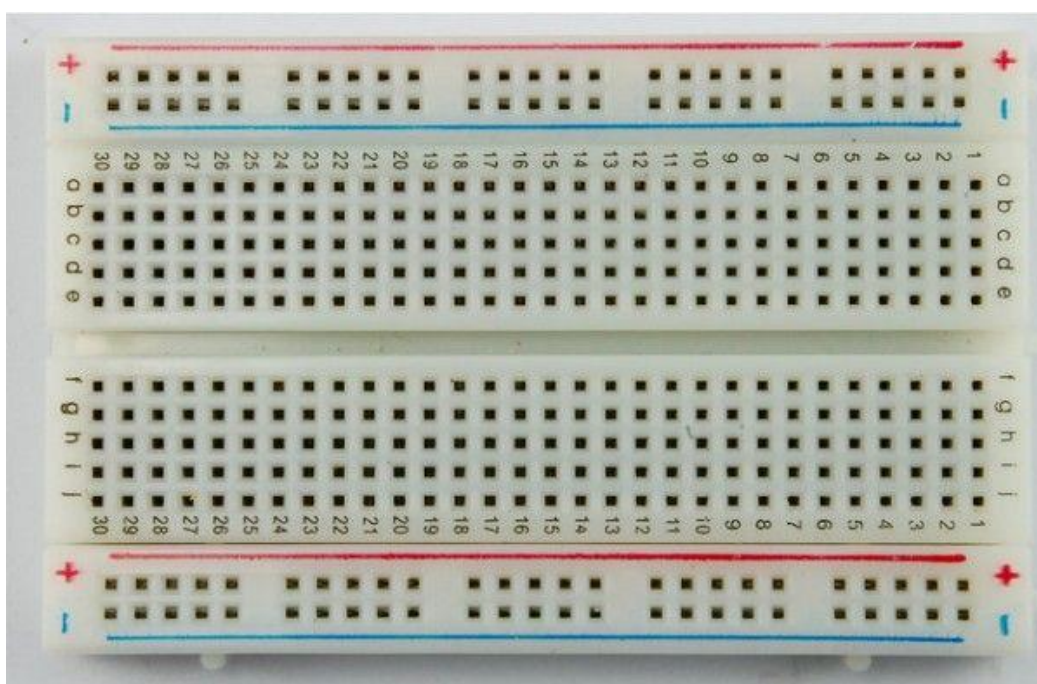
Father-father simlari (siz katta hajm bilan bunday to'plamni sotib olishingiz mumkin).



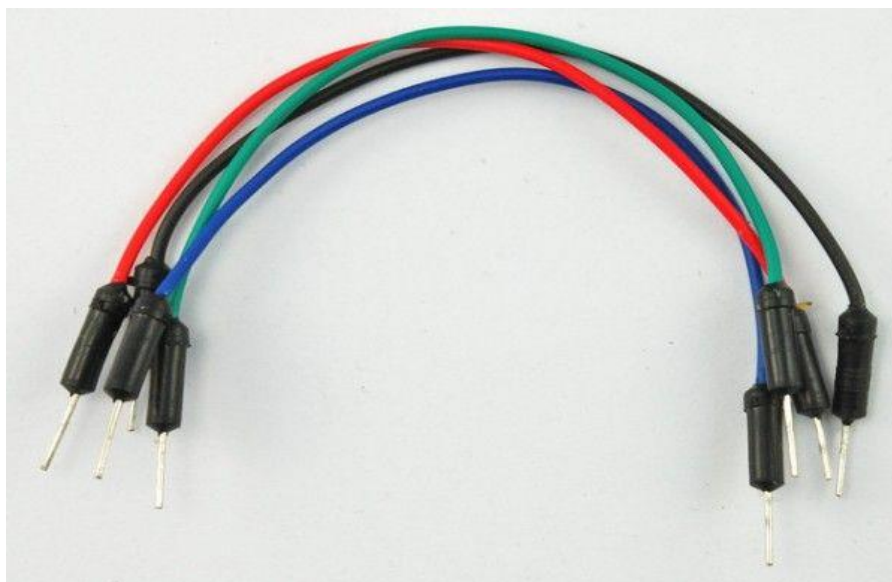
Arduino UNO



PIR – datchik



Buterburot



Father – father simlari

Arduino harakatni aniqlashda USB Serial aloqasi orqali xabar yuboradi. Biroq, sensorning har bir ishlashi bilan elektron pochta xabarini yuborsangiz, juda ko'p sonli elektron pochta xabarlarini olishingiz mumkin. Shuning uchun, agar o'tgan signaldan juda oz vaqt o'tgan bo'lsa, biz boshqa xabarni yuboramiz.

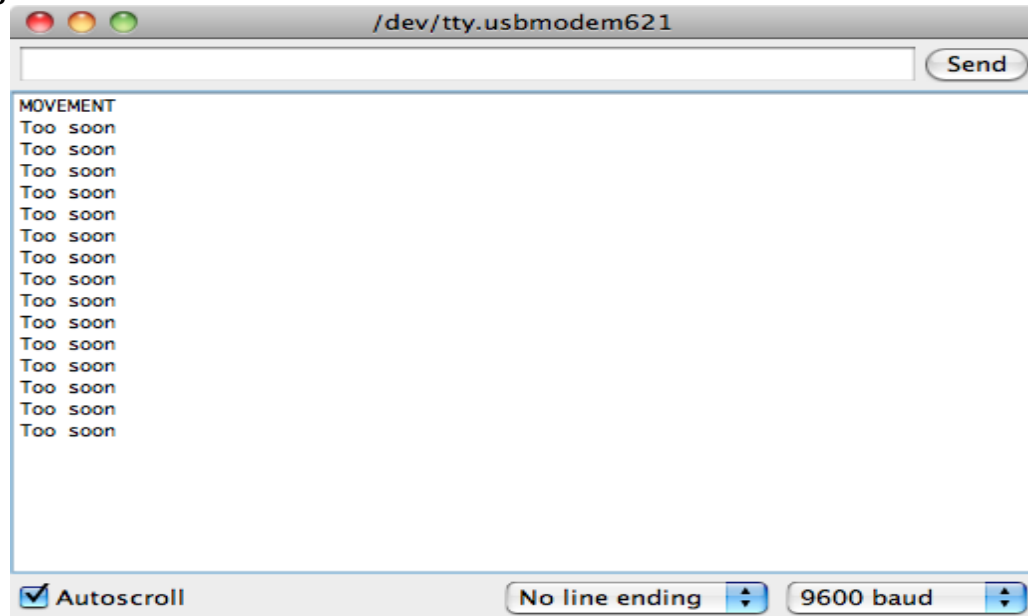
Dastur kodi:

```
int pirPin = 7;
int minSecsBetweenEmails = 60; // 1 min
long lastSend = -minSecsBetweenEmails * 1000;
void setup()
{
  pinMode(pirPin, INPUT);
  Serial.begin(9600);
}
void loop()
{
  long now = millis();
  if (digitalRead(pirPin) == HIGH)
  {
    if (now > (lastSend + minSecsBetweenEmails * 1000))
    {
      Serial.println("MOVEMENT"); lastSend = now;
    }
  }
}
```

```

else
{
  Serial.println("Too soon"); }
}
delay(500);
}

```



Arduino uchun sketch va sensorli tugma.

Sketch sensorli tugma bosilganda ketma-ket monitor oynasida ma'lumotlarni namoyish qiladi.

// Arduino bilan sensorli tugma modulidan foydalanish bo'yicha ko'rsatmalar

// arduino-diy.com saytida to'liq ko'rsatmalar

// qachonki Sig Output, high bo'lsa, таҭ sensorni bosish

#define ctsPin 2 // sensor datchik uchun pin

int ledPin = 13; // svetodiod uchun pin

void setup() {

Serial.begin(9600);

pinMode(ledPin, OUTPUT);

pinMode(ctsPin, INPUT);

}

void loop() {

int ctsValue = digitalRead(ctsPin);

if (ctsValue == HIGH){

digitalWrite(ledPin, HIGH);

Serial.println("TOUCHED");

```

    } else{
    digitalWrite(ledPin,LOW);
    Serial.println("not touched");
    }
    delay(500);
    }

```

Sensorli ekran modelimizda yuqorida keltirilgan signallar mos ravishda 6, 5, 4, 3 va 2-sonli pinlarga ulangan.

Shunday qilib, bizning displeyimiz uchun dasturga quyidagi satrlarni kiritishimiz kerak:

```

#include <UTouch.h>
UTouch myTouch(6,5,4,3,2);

```

Keyinchalik, bo'sh joyni voidsetup() tartibida displey va sensorli ekranni sozlashingiz kerak:

```

void setup() {
myGLCD.InitLCD();
myTouch.InitTouch();
}

```

InitTouch buyrug'i, asosiy kutubxonaning InitLCD buyrug'i kabi, displeyning vertikal yoki gorizontal yo'nalishini o'rnatadi. Standart yo'nalish gorizontal. Mana, endi siz bizning sensorli ekranimiz bilan ishlashingiz mumkin. Sensorli ekranning ishlashini tekshirishimiz mumkin bo'lgan juda oddiy misolni ko'rib chiqaylik. Keling, quyidagi dasturni tuzamiz va uni boshqaruvchining xotirasiga yuklaymiz:

```

#include <UTFT.h>
UTFT myGLCD(CTE32HR,25,26,27,28);
#include <UTouch.h>
UTouch myTouch(6,5,4,3,2);
void setup() {
myGLCD.InitLCD();
myTouch.InitTouch();
myGLCD.clrScr();
myGLCD.fillScr(255,0,0);
}
void loop() {
if (myTouch.dataAvailable()) {

```

```

myGLCD.fillScr(0,255,0);
}else {myGLCD.fillScr(255,0,0);}
}

```

Hammasi bo'lib, 1 dan 4 gacha raqamlar bilan kodlangan to'rtta aniqlik darajasi aniqlanadi:

```

PREC_LOW – quyi aniqlik;
PREC_MEDIUM – o'rta aniqlik;
PREC_HI – yuqori aniqlik;
PREC_EXTREME – juda yuqori aniqlik.

```

Aniqlik darajalari o'rtasidagi farqni UTouch kutubxonasi bilan ta'minlangan UTouch_QuickDraw misolida tasvirlash mumkin. Ushbu dastur sensorli ekran yordamida ekranda chizish imkonini beradi:

```

#include <UTFT.h>
UTFT myGLCD(CTE32HR,25,26,27,28);
#include <UTouch.h>
UTouch myTouch(6,5,4,3,2);
void setup() {
  myGLCD.InitLCD();
  myGLCD.clrScr();
  myTouch.InitTouch(PORTRAIT);
  myTouch.setPrecision(PREC_MEDIUM);
} void loop() {
  long x, y;
  while (myTouch.dataAvailable() == true) {
    myTouch.read();
    x = myTouch.getX();
    y = myTouch.getY();
    if ((x!=-1) and (y!=-1)) {
      myGLCD.drawPixel (x, y);
    } } }

```

Display rangining o'zgarishiga olib kelgan va teginish davomida saqlanib qolgan dataAvailable funksiyasiga qo'ng'iroq mavjud. Yana bir misolni ko'rib chiqamiz - teginish vaqtini millisekundlarda o'lchash:

```

#include <UTFT.h>

```

```

UTFT myGLCD(CTE32HR,25,26,27,28);
extern uint8_t BigFont[];
#include <UTouch.h>
UTouch myTouch(6,5,4,3,2);
void setup() {
  myGLCD.InitLCD();
  myGLCD.setFont(BigFont);
  myTouch.InitTouch();
  myGLCD.clrScr();
}
void loop() {
  unsigned long timeOld;
  unsigned long timeNew;
  if (myTouch.dataAvailable() == true){
    timeOld=millis();
    while (myTouch.dataAvailable() == true) {
      timeNew=millis();
    }
    myGLCD.clrScr();
    myGLCD.printNumI(timeNew-timeOld, 100, 100);
  }
}

```

Sensorli ekranga tegganda quyidagi misol, sensorli nuqta koordinatalarini piksel va xom formatda ko'rsatadi:

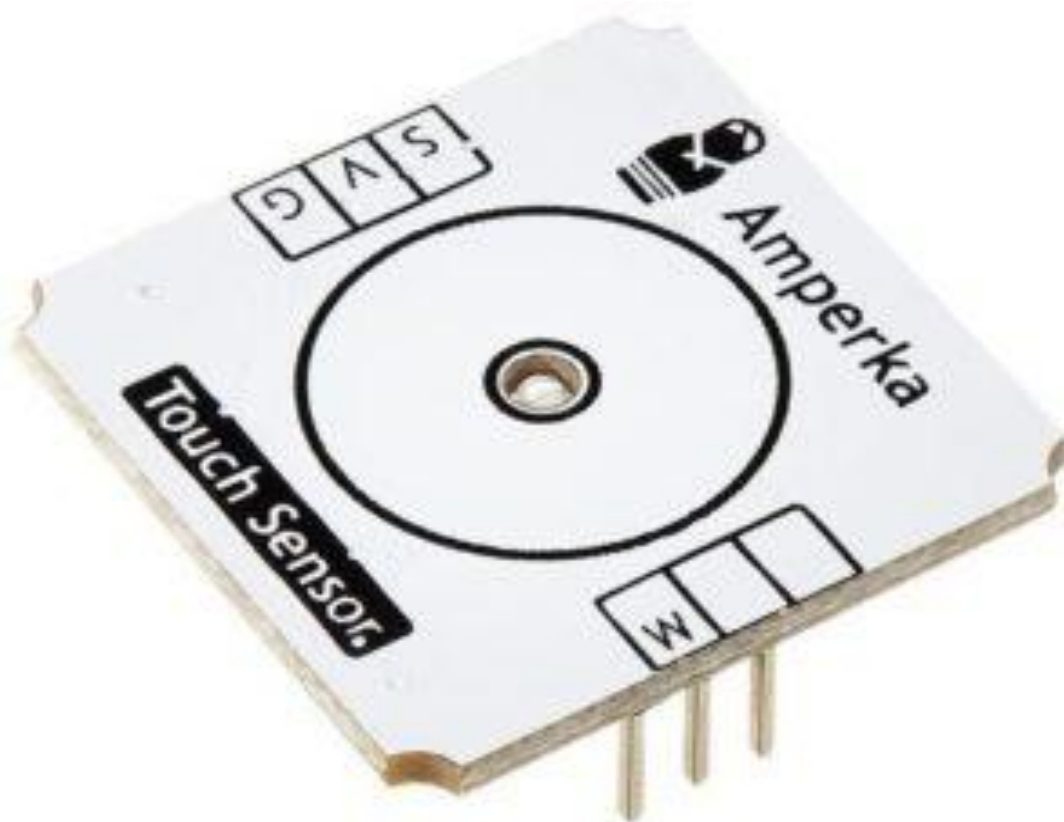
```

#include <UTFT.h>
UTFT myGLCD(CTE32HR,25,26,27,28);
extern uint8_t BigFont[];
#include <UTouch.h>
UTouch myTouch(6,5,4,3,2);
void setup() {
  myGLCD.InitLCD();
  myGLCD.setFont(BigFont);
  myTouch.InitTouch();
  myGLCD.clrScr();
}
void loop() {
  int X_RAW;

```

```
int Y_RAW;  
int X;  
int Y;  
if (myTouch.dataAvailable() == true){  
    myTouch.read();  
    X_RAW=myTouch.TP_X;  
    Y_RAW=myTouch.TP_Y;  
    X=myTouch.getX();  
    Y=myTouch.getY();  
    myGLCD.clrScr();  
    myGLCD.print("X", 100,80);  
    myGLCD.print("Y", 200,80);  
    myGLCD.print("PIX", 20,100);  
    myGLCD.print("RAW", 20,120);  
    myGLCD.printNumI(X, 100, 100);  
    myGLCD.printNumI(Y, 200, 100);  
    myGLCD.printNumI(X_RAW, 100, 120);  
    myGLCD.printNumI(Y_RAW, 200, 120);  
}  
}
```





Endi vazifani murakkablashtiraylik: tugmachani bosish svetodiodning ishlash rejimini o'zgartiradi.

```
1.  const int buttonPin = 7; // tugmachaning signal
    portiga ulangan portning qiymatlarini o'rnating
2.  int count = 0; // Ishlash rejimini tanlash uchun
    o'zgaruvchi
3.  void setup() {
4.    pinMode(LED_BUILTIN, OUTPUT); // Svetodiodning
    to'g'ri javob berish uchun buyruq
5.    pinMode(buttonPin, INPUT); // O'qish uchun portni
    ochamiz
6.  }
7.  void loop() {
8.    if(digitalRead(buttonPin)){ // Agar tugma bosilsa ...
9.      count = count + 1; // Tugmacha rejimini o'zgartiramiz
10.   if(count > 2){ // Agar count bo'lingan bo'lsa, boshidan
        hisoblashni boshlang
11.     count = 0;
12.   }
```



```

13. while(digitalRead(buttonPin)){ // Foydalanuvchiga
tugmachani qo'yib yuborishni kutadigan bo'sh sikl
14. }
15. }
16. if(count == 0) { // tugmani almashtirish uchun 3
rejim:
17. digitalWrite(LED_BUILTIN, LOW); // 1: O'chirilgan
svetodiod
18. } else if(count == 1) {
19. digitalWrite(LED_BUILTIN, HIGH); // 2: Yoqilgan
20. } else {
21. digitalWrite(LED_BUILTIN, HIGH); // 3: Miltillovchi
22. delay(100);
23. digitalWrite(LED_BUILTIN, LOW);
24. delay(100);
25. }
26. }

```

Datchiklarni dasturlash

Jyostik qurilmasini dasturlash

```
const int xPin = A1;
const int yPin = A0;
const int buttonPin = 2;

int xPosition = 0;
int yPosition = 0;
int buttonState = 0;

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

  pinMode(xPin, INPUT);
  pinMode(yPin, INPUT);

  pinMode(buttonPin, INPUT_PULLUP);
}

void loop() {
  xPosition = analogRead(xPin);
  yPosition = analogRead(yPin);
  buttonState = digitalRead(buttonPin);
  Serial.print("X: ");
  Serial.print(xPosition);
  Serial.print(" | Y: ");
```

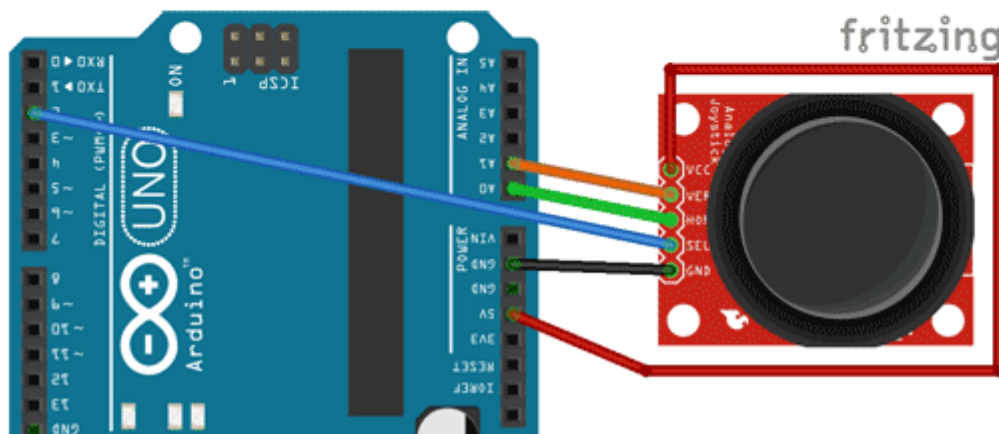
```

Serial.print(yPosition);
Serial.print(" | Button: ");
Serial.println(buttonState);
delay(100);
}

```

KY-023 | Arduino

GND	GND
+5V	5V
VRx	A
VRy	A
SW	D



Ultrasonic Sensorni ishlatish

Arduino kodi - NewPing kutubxonasidan foydalanish

Ultrasonik sensorni ishga tushirish va qabul qilingan signal pulsining kengligini qo'lda o'lchash o'rniga, biz maxsus kutubxonadan foydalanamiz. Ularning bir nechitasi mavjud, eng ko'p qirrali " NewPing " deb ataladi.

```

// Include NewPing Library
#include "NewPing.h"

```

```

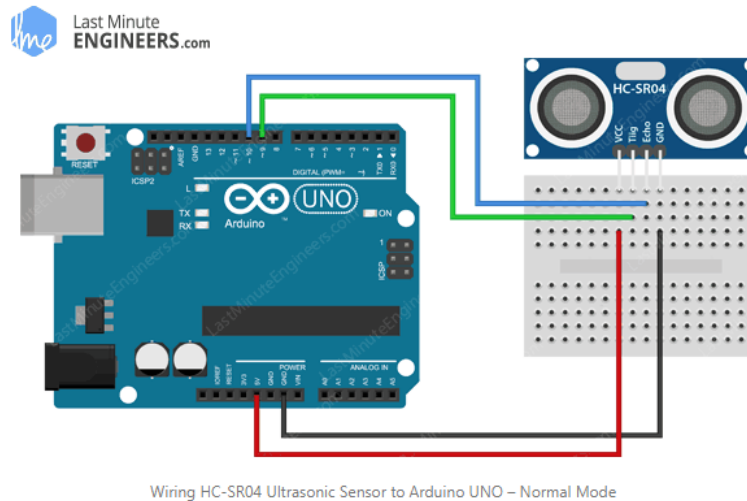
// Hook up HC-SR04 with Trig to Arduino Pin 9, Echo to
Arduino pin 10
#define TRIGGER_PIN 9
#define ECHO_PIN 10
// Maximum distance we want to ping for (in centimeters).
#define MAX_DISTANCE 400
// NewPing setup of pins and maximum distance.
NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);
float duration, distance;
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  // Send ping, get distance in cm
  distance = sonar.ping_cm();
  // Send results to Serial Monitor
  Serial.print("Distance = ");
  if (distance >= 400 || distance <= 2)
  {
    Serial.println("Out of range");
  }
  else
  {
    Serial.print(distance);

```

```

    Serial.println(" cm");
}
delay(50);
}

```



Ultrasonic sensorni Displey qurilmasi bilan bog'lab dasturlash

```
// includes the LiquidCrystal Library
```

```
#include <LiquidCrystal.h>
```

```
// includes the LcdBarGraph Library
```

```
#include <LcdBarGraph.h>
```

```
// Maximum distance we want to ping for (in centimeters).
```

```
#define max_distance 200
```

```
// Creates an LCD object. Parameters: (rs, enable, d4, d5, d6,
d7)
```

```
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
```

LcdBarGraph lbg(&lcd, 16, 0, 1); // Creates an LCD Bargraph object.

```
const int trigPin = 9;
const int echoPin = 10;
long duration;
int distance;
```

```
void setup()
{
  lcd.begin(16,2); // Initializes the interface to the LCD screen

  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}
```

```
void loop()
{
  // Write a pulse to the HC-SR04 Trigger Pin
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
```

```

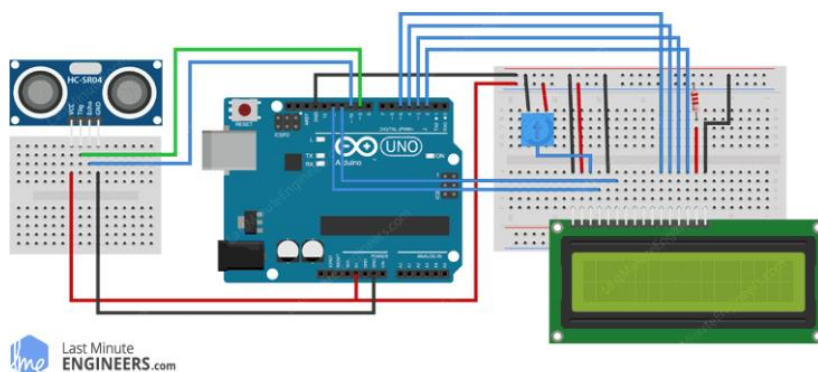
// Measure the response from the HC-SR04 Echo Pin
duration = pulseIn(echoPin, HIGH);

// Determine distance from duration
// Use 343 metres per second as speed of sound
distance= duration*0.034/2;

// Prints "Distance: <value>" on the first line of the LCD
lcd.setCursor(0,0);
lcd.print("Distance: ");
lcd.print(distance);
lcd.print(" cm");

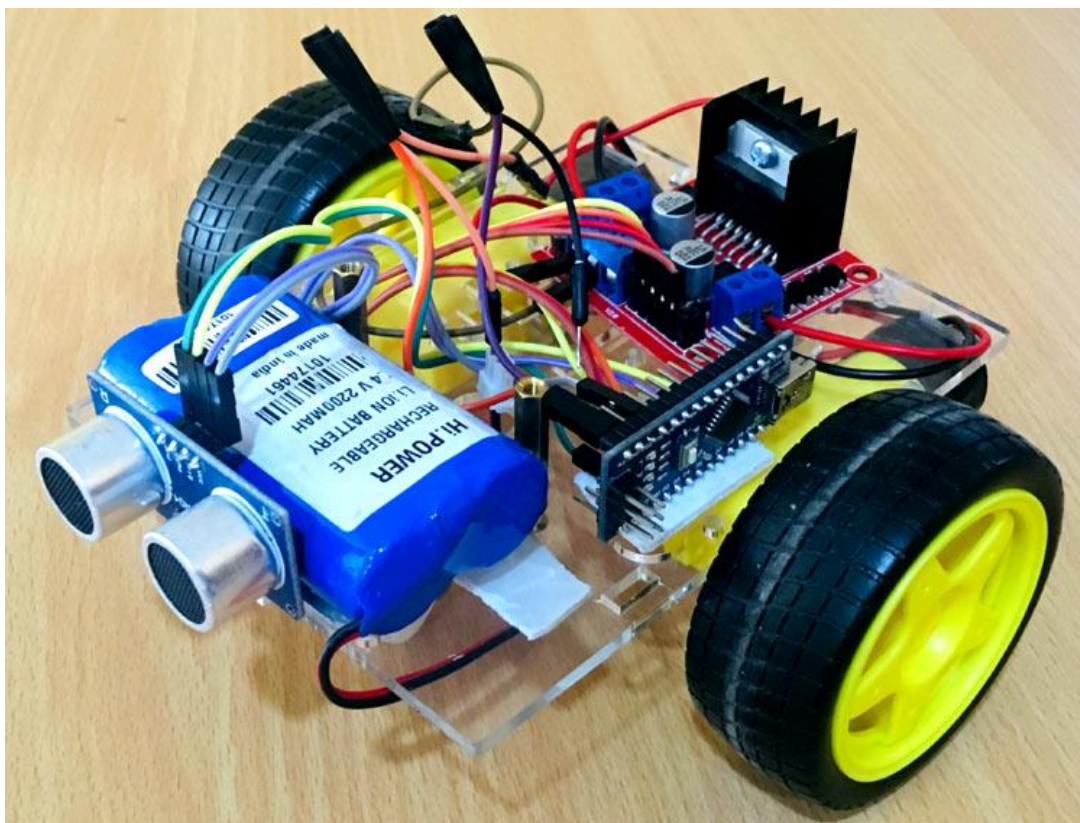
// Draws bargraph on the second line of the LCD
lcd.setCursor(0,1);
lbg.drawValue(distance, max_distance);
delay(500);
}

```



HC-SR04 ultratovush sensori va 16x2 LCD displeyni Arduino UNO ga ulash

Sensor yordamida harakatlanadigan robot mashina yig'ish

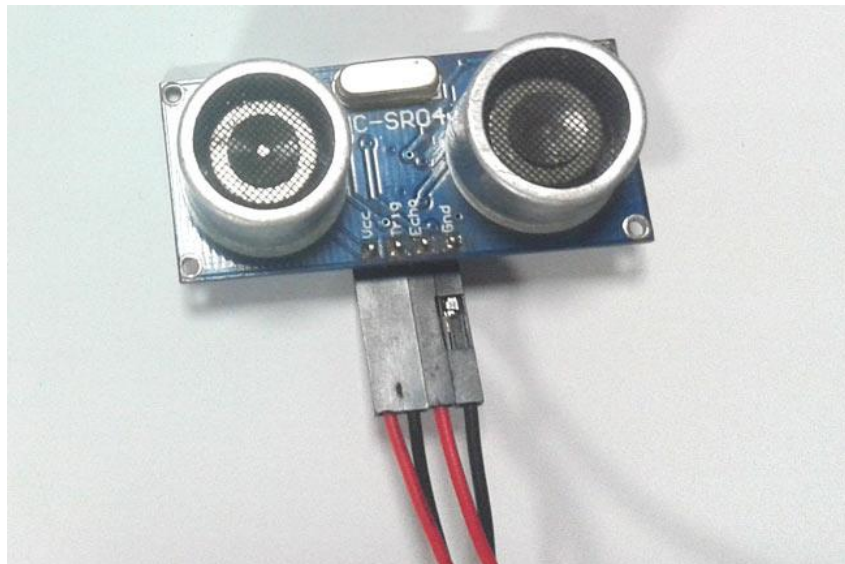


To'siqlardan qochish roboti - bu aqlli qurilma bo'lib, u oldidagi to'siqni avtomatik ravishda sezadi va o'zini boshqa tomonga burib, ulardan qocha oladi. Ushbu dizayn robotga to'qnashuvlardan qochib, noma'lum muhitda harakat qilish imkonini beradi, bu har qanday avtonom mobil robot uchun asosiy talabdir. To'siqlardan qochish robotining qo'llanilishi cheklanmagan va u hozirda ko'pgina harbiy tashkilotlarda qo'llaniladi, bu hech qanday askarlar bajara olmaydigan ko'plab xavfli ishlarni bajarishga yordam beradi.

Bu loyihada ultratovush sensori va Arduino yordamida to'siqlardan qochish robotini quramiz . Bu yerda ultratovush sensori robot va to'siq o'rtasidagi masofani hisoblash orqali yo'ldagi to'siqlarni sezish uchun ishlatiladi. Agar robot biron bir to'siqni topsa, u yo'nalishini o'zgartiradi va harakatni davom ettiradi.

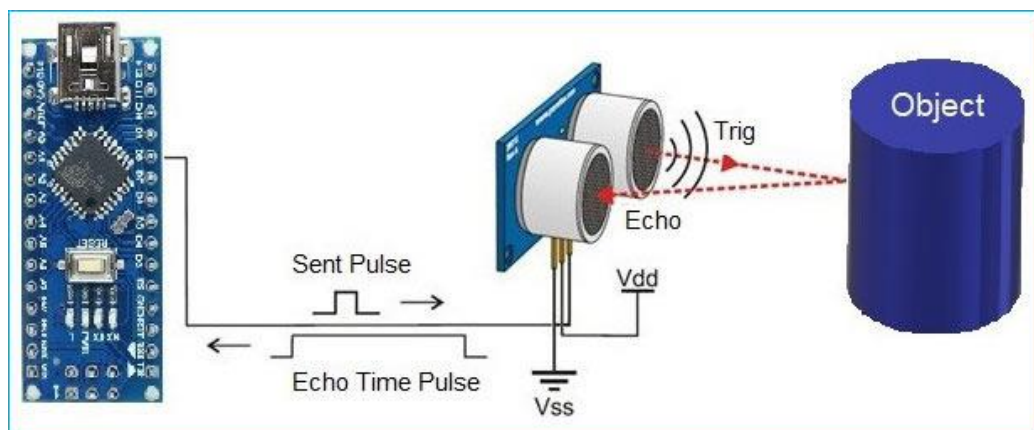
Robotni qurishdan oldin, ultratovush sensori qanday ishlashini tushunish kerak, chunki bu sensor to'siqni aniqlashda muhim rol o'ynaydi. Ultrasonik sensorning ishlashining asosiy printsipi

sensorning ultratovush nurlarini uzatish va ultratovush nurlarini sirtga urgandan keyin qabul qilish uchun sarflagan vaqtini qayd etishdir. Keyinchalik masofa formula yordamida hisoblab chiqiladi. Ushbu loyihada keng tarqalgan **HC-SR04 ultratovush sensori** qo'llaniladi. Ushbu sensordan foydalanish uchun yuqorida tavsiflangan shunga o'xshash yondashuv qo'llaniladi.



Shunday qilib, HC-SR04 trig pin kamida 10 us uchun baland qilingan. Ovozli nur har biri 40 KHz 8 ta impuls bilan uzatiladi.

Keyin signal sirtga tushadi va orqaga qaytib keladi va HC-SR04 qabul qiluvchining Echo piniga tushadi. Echo pin allaqachon yuqori yuborish vaqtida yuqori qilgan edi.



Nurning orqaga qaytish vaqti o'zgaruvchida saqlanadi va quyidagi kabi tegishli hisoblar yordamida masofaga aylantiriladi.

Biz ultratovush sensori haqida ko'proq ma'lumot olish uchun ultratovush sensori bilan bog'liq boshqa loyihalarni tekshirish uchun ko'plab loyihalarda foydalandik.

Ushbu to'siqdan qochib qutuladigan robotning tarkibiy qismlarini osongina topish mumkin. Shassi qilish uchun har qanday o'yinchoq shassisidan foydalanish mumkin yoki buyurtma asosida tayyorlanishi mumkin.

Quyidagi komponentlar talab qilinadi

1. Arduino NANO yoki Uno (har qanday versiya)
2. HC-SR04 ultratovush sensori
3. LM298N Dvigatel haydovchi moduli
4. 5V DC motorlar
5. Batareya
6. G'ildiraklar
7. Shassi
8. Jumper simlari

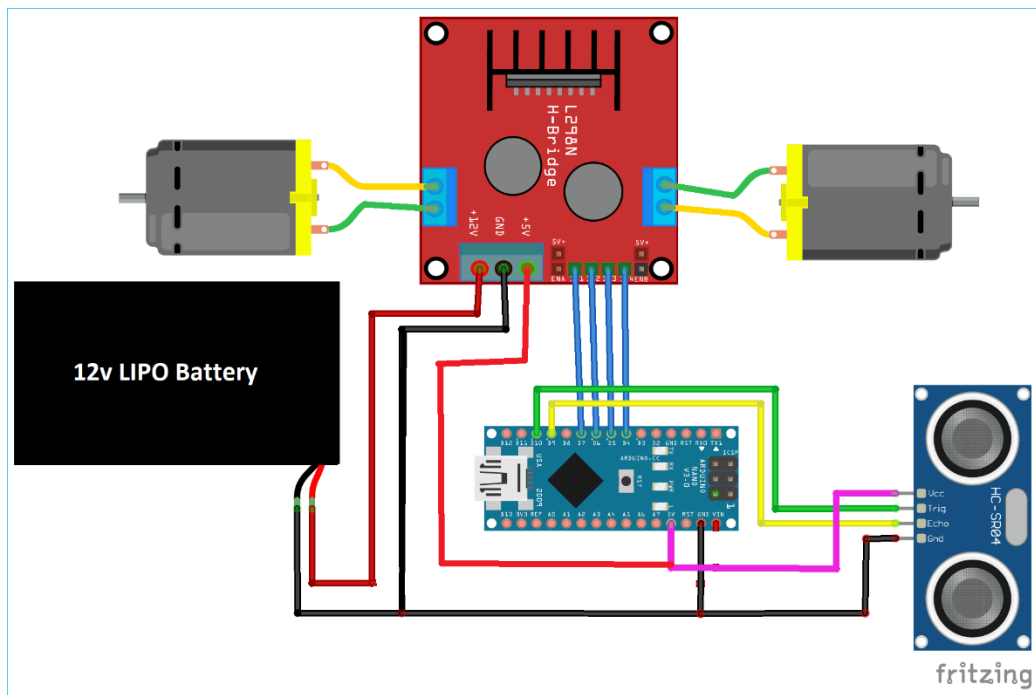
Elektr sxemasi

Ushbu loyihaning to'liq sxemasi quyida keltirilgan, chunki siz ko'rib turganingizdek, u Arduino nanodan foydalanadi. **Ammo biz Arduino UNO-dan foydalanib**, bir xil sxema (xuddi shu pinout bo'yicha) va kod bilan to'siqlardan qochadigan robot qurishimiz mumkin .

Sxema tayyor bo'lgach, biz quyida ko'rsatilganidek, sxemani robot shassisining tepasiga yig'ish orqali **mashinani chetlab o'tish uchun to'siqni** qurishimiz kerak .

Arduino yordamida to'siqlardan qochish roboti – kodlarini ko'rib chiqamiz

Dastur HC-SR04 modulini o'rnatishni va dvigatel yo'nalishini mos ravishda siljitish uchun signallarni Motor Pinlarga chiqarishni o'z ichiga oladi. Ushbu loyihada kutubxonalardan foydalanilmaydi.

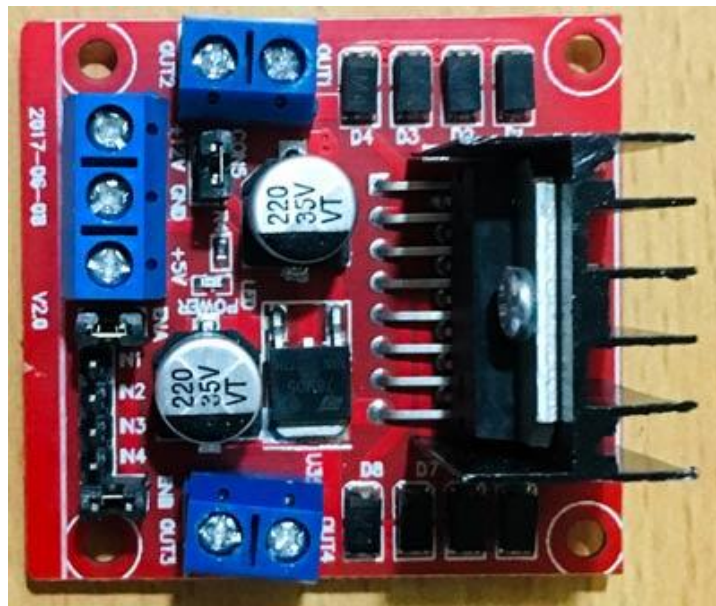


Avval dasturda **HC-SR04 trig va echo pinini aniqlang** . Ushbu loyihada trig pin GPIO9 ga ulangan va echo pin Arduino NANO ning GPIO10 ga ulangan.

```
int trigPin = 9; // HC-SR04 trig pin
int echoPin = 10; // HC-SR04 aks-sadosi
```

LM298N Dvigatel drayveri modulini kiritish uchun pinlarni aniqlang. LM298N-da unga ulangan dvigatel yo'nalishini boshqarish uchun ishlatiladigan 4 ta ma'lumot kiritish pinlari mavjud.

```
int revleft4 = 4; //Chap dvigatelning teskari harakati
int fwdleft5 = 5; //Chap dvigatelning ForWarD harakati
int revright6 = 6; //O'ng motorning teskari harakati
int fwdright7 = 7; //O'ng dvigatelning Oldinga
harakatlanishi
```



Setup() funksiyasida foydalanilgan **GPIO pinlarining ma'lumotlar yo'nalishini aniqlang**. To'rtta Dvigatel pinlari va Trig pinlari OUTPUT sifatida o'rnatiladi va Echo Pin Kirish sifatida o'rnatiladi.

```
pinMode (revleft4, OUTPUT);
// Dvigatel pinlarini chiqish pinMode (fwdleft5,
OUTPUT)
sifatida o'rnatish ; pinMode (revright6, OUTPUT);
pinMode (fwdright7, OUTPUT);
pinMode (trigPin, OUTPUT);
// trig pinini chiqish pinMode (echoPin, INPUT) sifatida
o'rnatish ; // aks ettirilgan to'lqinlarni olish uchun echo
pinini kirish sifatida o'rnatish
```

Loop() funksiyasida HC -SR04 dan masofani oling va masofaga qarab vosita yo'nalishini siljiting. Masofa robot oldida kelayotgan ob'ekt masofasini ko'rsatadi. Masofa 10 usgacha bo'lgan ultratovush nurini yorib, 10 usdan keyin qabul qilish orqali olinadi. Ultrasonik sensor va Arduino yordamida masofani o'lchash haqida ko'proq bilish uchun havolaga o'ting.

```
digitalWrite (trigPin, LOW);  
delayMicroseconds(2);  
digitalWrite (trigPin, HIGH); // to'lqinlarni 10 us  
delayMicroseconds(10) uchun yuborish;  
davomiyligi = pulseIn(echoPin, HIGH); // aks ettirilgan  
to'lqinlarni qabul qilish  
masofa = davomiyligi / 58,2; // masofadagi kechikishga  
aylantirish  
(10);
```

Agar masofa belgilangan masofadan kattaroq bo'lsa, uning yo'lida to'siq yo'q va u oldinga siljiydi.

```
agar (masofa > 19)  
{  
digitalWrite(fwdright7, HIGH); // oldinga siljiting  
digitalWrite(revright6, LOW);  
digitalWrite(fwdleft5, HIGH);  
digitalWrite (revleft4, LOW);  
}
```

To'siqdan qochish uchun masofa belgilangan masofadan kamroq bo'lsa, oldinda qandaydir to'siq borligini anglatadi. Shunday qilib, bu holatda robot bir muncha vaqt to'xtaydi va orqaga qarab harakat qiladi, keyin yana bir muddat to'xtaydi va keyin boshqa yo'nalishga aylanadi.

```
agar (masofa < 18)  
{
```

```

    digitalWrite(fwdright7, LOW); //Digital yozishni
to'xtatish
    (revright6, LOW);
    digitalWrite (chapga 5, LOW);
    digitalWrite (revleft4, LOW);
    kechikish (500);
    digitalWrite (fwdright7, LOW); //movebackward
    digitalWrite(revright6, HIGH);
    digitalWrite (chapga 5, LOW);
    digitalWrite (revleft4, HIGH);
    kechikish (500);
    digitalWrite (fwdright7, LOW); //Digital yozishni
to'xtatish
    (revright6, LOW);
    digitalWrite (chapga 5, LOW);
    digitalWrite (revleft4, LOW);
    kechikish (100);
    digitalWrite(fwdright7, HIGH);
    digitalWrite (revright6, LOW);
    digitalWrite (revleft4, LOW);
    digitalWrite (chapga 5, LOW);
    kechikish (500);
}

```

Shunday qilib, robot hech qayerda qolib ketmasdan, yo'lidagi to'siqlardan qochishi mumkin. **To'liq kod va videoni quyidatoping**

```
/* Ultrasonik sensor va Arduino NANO yordamida robotdan  
to'siqlardan qochish
```

```
* O'chirish to'plami ( www.circuitdigest.com )
```

```
*/
```

```
int trigPin = 9; // HC-SR04 trig pin
```

```
int echoPin = 10; // HC-SR04 aks-sadosi
```

```
int revleft4 = 4; //Chap dvigatelning teskari harakati
```

```
int fwdleft5 = 5; //Chap dvigatelning ForWarD harakati
```

```
int revright6 = 6; //O'ng motorning teskari harakati
```

```
int fwdright7 = 7; //O'ng dvigatelning Oldinga  
harakatlanishi
```

```
uzoq davom etish, masofa;
```

```
bekor o'rnatish () {
```

```
    kechikish (tasodifiy (500,2000)); // tasodifiy vaqt uchun  
kechikish
```

```
    Serial.begin(9600);
```

```
    pinMode (revleft4, OUTPUT);
```

```
    // Dvigatel pinlarini chiqish pinMode (fwdleft5, OUTPUT)  
sifatida o'rnatish ;
```

```

pinMode (revright6, OUTPUT);

pinMode (fwdright7, OUTPUT);


pinMode (trigPin, OUTPUT);

// trig pinini chiqish pinMode (echoPin, INPUT) sifatida
o'rnating ; // aks ettirilgan to'lqinlarni olish uchun aks-sado
pinini kirish sifatida o'rnating

}

void loop() {

digitalWrite (trigPin, LOW);

delayMicroseconds(2);

digitalWrite (trigPin, HIGH); // to'lqinlarni 10 us

delayMicroseconds(10) uchun yuborish;

davomiyligi = pulseIn(echoPin, HIGH); // aks ettirilgan
to'lqinlarni qabul qilish

masofa = davomiyligi / 58,2; // masofadagi kechikishga
aylantirish

(10);

// Agar robotingiz to'g'ri harakat qilmasangiz, pin
raqamlarini o'zgartiring,

agar (masofa > 19)

```



```

{

digitalWrite(fwdright7, HIGH); // oldinga siljiting

digitalWrite(revrigh6, LOW);

digitalWrite(fwdleft5, HIGH);

digitalWrite (revleft4, LOW);

}

agar (masofa < 18)

{

digitalWrite(fwdright7, LOW); //Digital yozishni to'xtatish

(revrigh6, LOW);

digitalWrite (chapga 5, LOW);

digitalWrite (revleft4, LOW);

kechikish (500);

digitalWrite (fwdright7, LOW); //movebackward

digitalWrite(revrigh6, HIGH);

digitalWrite (chapga 5, LOW);

digitalWrite (revleft4, HIGH);

kechikish (500);

```

```
digitalWrite (fwdright7, LOW); //Digital yozishni to'xtatish  
  
(revright6, LOW);  
  
digitalWrite (chapga 5, LOW);  
  
digitalWrite (revleft4, LOW);  
  
kechikish (100);  
  
digitalWrite(fwdright7, HIGH);  
  
digitalWrite (revright6, LOW);  
  
digitalWrite (revleft4, LOW);  
  
digitalWrite (chapga 5, LOW);
```



```
    kechikish (500);  
  
}  
  
}
```

Shu QR-kod skaner qilish orqali siz ushbu loyihaning amaliyotda bajarish videosini ko'rishingiz mumkin.

Scratch dasturida o'yin yaratish texnologiyasi.

Scratch – bu asosan kodni o'rganishda yordam beradigan 8-16 yoshdagi bolalarga mo'ljallangan blokli vizual dasturlash tili va veb-sayti. Sayt foydalanuvchilari Internetda blokga o'xshash interfeys yordamida loyihalar yaratishlari mumkin. Xizmat MIT Media Lab tomonidan ishlab chiqilgan, 70 dan ortiq tillarga tarjima qilingan va dunyoning aksariyat qismlarida qo'llaniladi. Scratch maktabdan keyingi markazlarda, maktablarda va kollejlarda, shuningdek boshqa jamoat bilimlari muassasalarida o'qitiladi va qo'llaniladi.

SCRATCHning maqsad va vazifalari

SCRATCH dasturida dasturlash asoslarini o'rganish bolalar uchun o'zini namoyon qilish va ijod qilish uchun yangi imkoniyatlar ochadi va ularga g'oyalarni gavdalantirish imkonini beradi.

Loyiha maqsadlari:

- bolalarda mantiqiy va tanqidiy fikrlashni rivojlantirish;
- tasavvurni, shuningdek, murakkab muammolarni hal qilish qobiliyatini rivojlantirish,
- SCRATCH misolida dasturlashning umumiy tamoyillari haqida tasavvur hosil qilish;
- SCRATCH asosida asosiy dasturlash ko'nikmalarini shakllantirish;

SCRATCH dasturlash muhiti bolalarga o'zlarining interaktiv loyihalarini yaratishga imkon beradi: o'yinlar, multfilmlar, o'quv tizimlari, komikslar va boshqalar.

O'quvchilar SCRATCHda loyihalar yaratishda, ular 21-asrning ko'plab ko'nikmalarini o'rganadilar:

- ijodiy fikrlash,
- mavzu bo'yicha muloqot,
- tizimli tahlil;
- texnologiyadan bimalol foydalanish;
- samarali o'zaro ta'sir;
- loyihalash,
- doimiy ta'lim.

Bu muhitning asosiy afzalliklaridan biri shundaki, u erkin tarqatuvdagi dasturiy mahsulot, har qanday ta'lim muassasasi dasturni Internetdan yuklab olib, bevosita yangi dasturlash muhitida o'qish va ishlashni boshlashi mumkin.

Scratch qiziqarli va dasturlashni o'rganish oson bo'lishi uchun yaratilgan. Unda bloklarga asoslangan dasturlash yordamida interaktiv hikoyalar, o'yinlar, badiiy, simulyatsiya va boshqalarni yaratish uchun vositalar mavjud. Scratchda o'zining bo'yoq muharriri va ovoz muharriri o'rnatilgan.

Foydalanuvchilar Scratchda bloklar palitrasidan bloklarni sudrab olib, boshqotirmalar kabi jumboq singari dasturlash orqali dasturlashadi. Bir nechta bloklarning tuzilmalari skript deb nomlanadi. Ushbu dasturlash usuli (bloklar bilan qurilish kodi) "sudrab olib tashlash dasturlash" deb nomlanadi.

Scratch dasturlash muhitida ham LEGO konstruktori kabi dasturning kodi rangli bloklardan hosil qilinadi (yig'iladi). Konstruktor shaklidagi bunday bloklarning soddaligi dasturlashni qiziqarli o'yin shaklida o'rganish imkonini beradi.

Scratch Tarixi

Scratch 2003 yilda ishlab chiqila boshlandi va 2007 yilda ommaga taqdim etildi. Scratch 1, birinchi versiyasi, Scratch 4 ga juda o'xshash edi. O'sha paytda faqat oflayn muharrir mavjud edi; veb-sayt loyihalarni yuklash va o'ynash mumkin bo'lgan kichik blog edi. Scratch o'sishi bilan veb-sayt kengaytirildi.

Scratch 1, Scratch 2, Scratch 3 va Scratch 4 chiqdi. Shu nuqtada Scratch-ning millionlab foydalanuvchilari, loyihalari va ko'plab yangi funktsiyalari mavjud edi. Scratch 2 yilda 2013, veb-sayt va foydalanuvchi interfeysi o'zgartirildi. Scratch o'sishda davom etdi, 30 million foydalanuvchi va loyihaga erishdi. Scratch 3 2019 yilda chiqarilgan.

Shiori

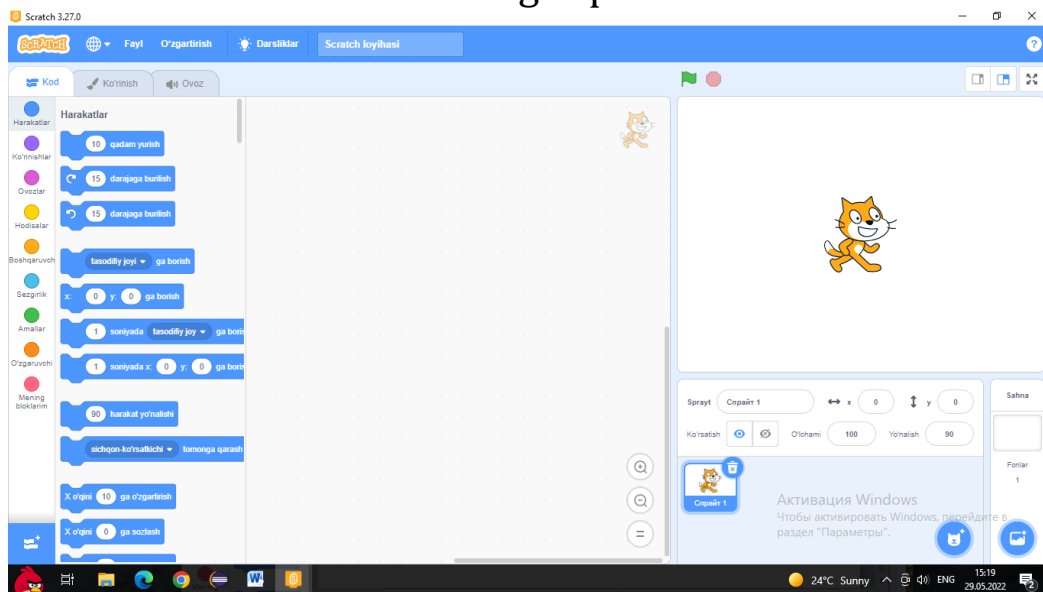
Scratch shiori - "Tasavvur qiling, dasturlashing, ulashing". Bu loyihani yaratishning asosiy printsiptiga amal qiladi: g'oya paydo bo'ladi ("Tasavvur qiling"), g'oyani Scratchda dasturlashtirasiz ("Dastur"), so'ngra jamoada nashr etamiz ("Bahamlashish").

Kompyuterda biror masalani yechish va bundan to'g'ri natija olish uchun unga beriladigan buyruq va ko'rsatmalar ketma-ketligini to'g'ri belgilash lozim.

Bunday jarayonga algoritmlash jarayoni deyiladi. Demak, Algoritm ijrochisi — inson kabi, dastgohlar(stanoklar), mashinalar, robotlar, dronlar bo'lishi mumkin.



Scratch logotipi



1. Boshqarish panellari – Ribbon (tasmali panel) ko'rinishida joylashgan bo'lib, oq rangdagi bo'lim faol panel hisoblanadi. Ularni sichqoncha yordamida oson boshqarish mumkin.

Code (dastur kodi) – asosiy ish muhiti. Bu muhitda bloklar yordamida dastur yaratiladi va ishga tushiriladi.

Costumes (kostyumlar bo'limi) – bu muhitda dastur kutubxonasidagi kostyumlarni almashtirish, yoki yangi kostyumni hosil qilish va internet orqali yuklab olish vazifalari bajariladi.

Sounds (musiqa bo'limi) – dasturda ishtirok etayotgan qahramonlarga ovoz berish yoki musiqa tovushiga harakatlantirish vazifalari bajariladi.

2. Skriptlar (bloklar) paneli.

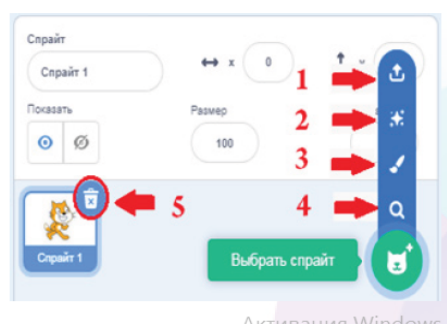
3. Markaziy panel. Algoritm asosida bloklar taxlanib skript (dastur) hosil qilinadi.

4. Stage (sahna). Dastur ishga tushirilgach, spraytlar harakatini kuzatish mumkin bo'lgan ishchi oyna.
5. Rangli bloklar paneli.
6. Bloklarni(skriptlarni) kattalashtirish, kichiklashtirish va o'zgarmas o'lchamga keltirish uskunolari.
7. Spraytlarni (qahramonlarni) boshqarish paneli.
8. Spraytlarni tanlash.
9. Fon tanlash.
10. Dasturni ishga tushirish va to'xtatish tugmachalari.

Spraytlarni qo'shish va olib tashlash

Dastur ishga tushirilganda sahnada standart ob'ekt (sprayt) — mushuk aks etib turadi. Yangi spraytni qo'shish uchun quyidagi tugmachalardan  foydalaniladi:

- 1 – Kompyuter xotirasidan spraytni yuklash.
- 2 – Syurpriz (dasturning o'zi tanlagan tasodifiy sprayt).
- 3 – Dasturning o'z grafik muharririda yangi sprayt chizish.
- 4 – Dasturning spraytlar kutubxonasidan tanlash.
- 5 – Spraytni olib tashlash.



Bloklarni(skriptlarni) tahlash oynasi

Markaziy oynachaning bo'sh qismiga 2 – ustunda joylashgan buyruq bloklarini algoritm bo'yicha sudrab o'tkaziladi va taxlanadi. Bu jarayon pazl yig'ish jarayoniga o'xshaydi, shu sababli Skript (процедура) deb qaraladi.

Command	Команды	Buyruqlar
		Harakat bloklari Tashqi ko'rinish (kostyumlar) Ovoz, musiqa bloklari Jarayonlar bloki Boshqarish bloklari Sensor bloklari Operatorlar O'zgaruvchi bloklar Qo'shimcha bloklar



Yashil bayroqcha — Skript (dastur)ni ishga tushirish tugmasi.

Pushti oltiburchak — namoyishni ixtiyoriy vaqtda to'xtatish tugmachasi.

Sahnaning yuqori o'ng qismida joylashgan. Skriptlarni ishga tushirishning boshqa yo'li. Bu blok skriptning eng birinchi qatorida joylashadi. Yashil bayroqchani 2 marta sichqoncha tugmachasi bilan bosilsa barcha buyruqlar ishga tushadi.

Multfilm o'zi nima?

Uning to'liq atamasi – multiplikatsion film (lotin. Multiplikatio (ko'paytirish) va ing.film (plenka) kinoteatr, teleko'rsatuv va kompyuter ekranlarida ko'rishga mo'ljallangan film.

Multfilmlar yaratilish texnologiyasida bir necha san'at yo'nalishlari jamlangan: rasm, harakat, tovush, musiqa va rejissura, ya'ni multfilm ssenariysi. Ssenariydagi voqealar ketma-ketligi ham dasturlash jarayoniga o'xshash bo'ladi.

Scratch dasturida sodda multfilmlardan, murakkab o'yin dasturlari yaratish mumkinligini aytib o'tgan edik.

SCRATCH o'rnatilgan kutubxonaga ega bo'lib, undan siz o'zingizning loyihalaringsizga tayyor spraytlarni tanlashingiz va qo'shishingiz mumkin. Ulardan ba'zilar jonlantirilgan va ovoz effektlariga ega.

Loyihangizning ishlab chiqish bosqichlari

O'z o'yiningizni yoki multfilmingizni **yaratish uchun asosiy harakatlar ketma-ketligi quyidagicha:**

1. SCRATCHni ishga tushiramiz va sahnaga birinchi spraytni qo'shamiz. Siz loyihani nomlashingiz mumkin, ammo bu shart emas.

2. Obyekt yoki personajning tashqi ko'rinishini sozlaymiz (kimdan boshlaganingizga qarab). Kostyum chizamiz yoki tayyorlardan tanlaymiz.

3. Skriptni bloklardan yig'amiz. Chap tomonda palitradan kerakli guruhni qidiramiz va kerakli shartlarni, operatorlarni, amallarni birin-ketin kod maydoniga tortamiz.

4. Yangi spraytlar va skriptlarni qo'shamiz, ular o'rtasida almashing va g'oyamizni amalga oshirgunimizcha ularni yaxshilaymiz.

5. Vaqti-vaqti bilan loyihani ishga tushiramiz va uning ishini sinab ko'ramiz. O'zgartirishlar kiritamiz, yana o'ynash yoki multfilmni tomosha qilishga urinamiz.

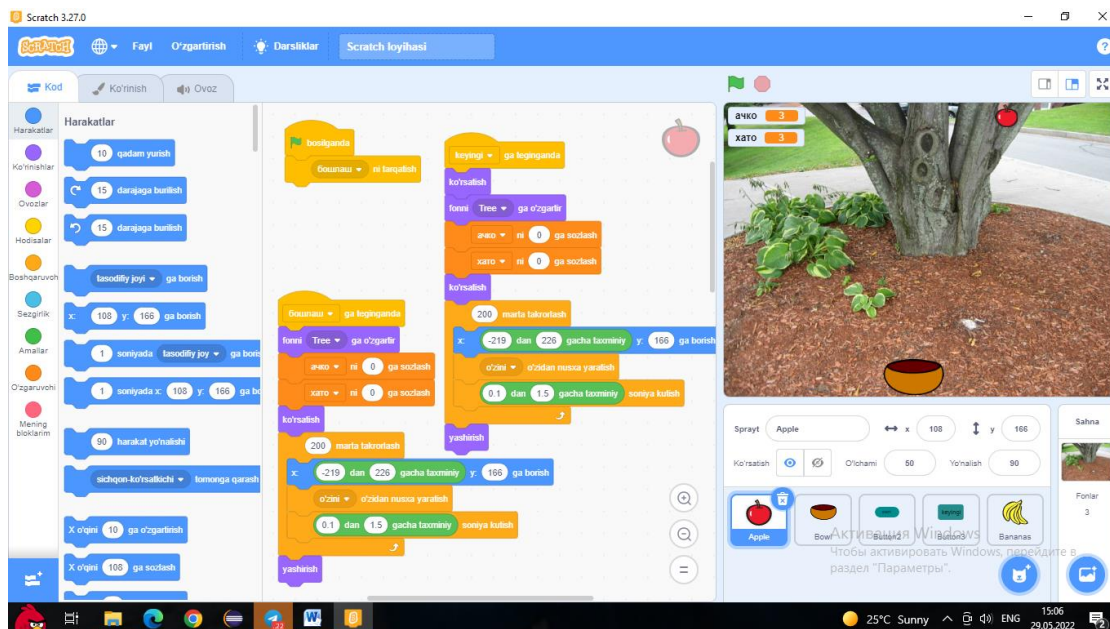
Tugallangan loyiha nashr etilishi va uni boshqalarga o'ynashga berish mumkin.

Bajarish texnologiyasi:

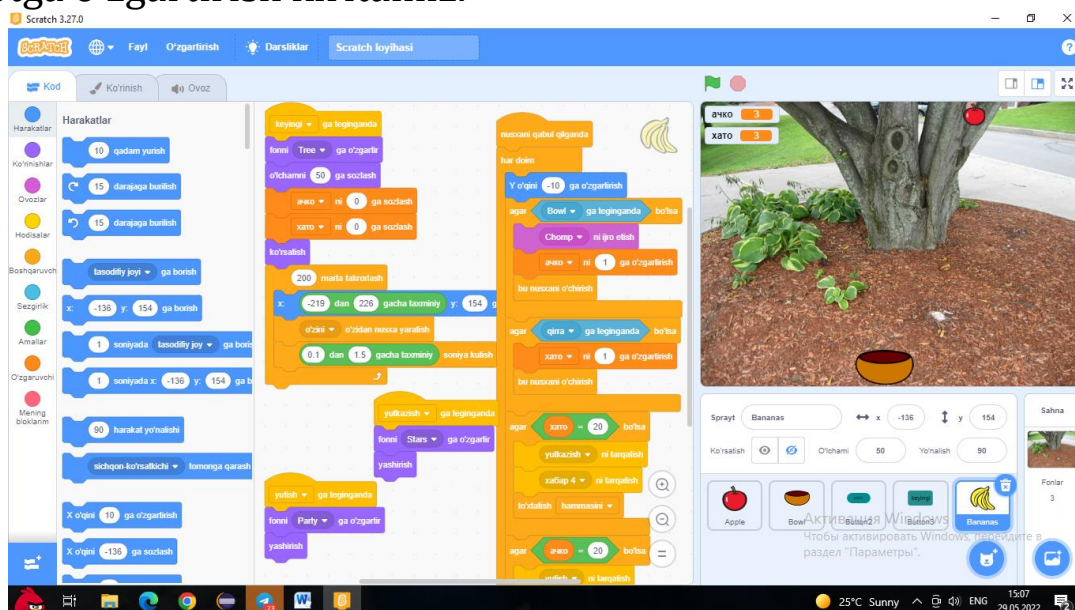
[illegible]

1-Mevalar sprayti o'lchami 50 % va $x = -74$, $y = 166$ pozitsiyasini belgilaymiz.

2-Paqircha sprayt o'lchami 150% va $x = 60$, $y = -144$ pozitsiyasini o'rnatamiz.



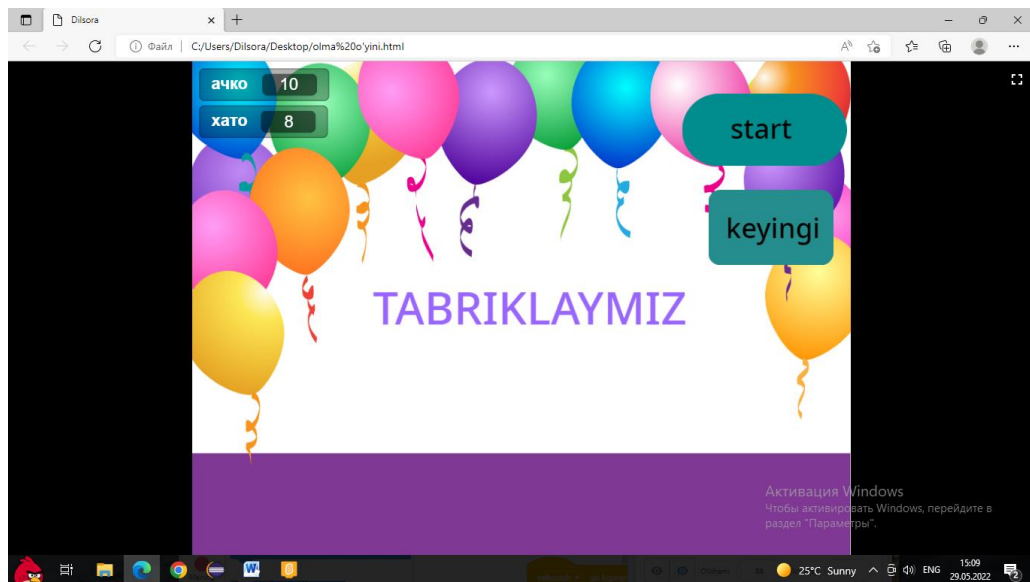
Loyihamizga Ovoz - Takrorlash - musiqa tanlaymiz va skriptga o'zgartirish kiritamiz.



Qo'shimcha yozilgan skriptni ixtiyoriy sprayt kodiga qo'shish mumkin.

Dasturni ishga tushirib o'yinni tomosha qilamiz. Dasturni to'xtatish uchun  tugmachasini bosamiz.

Dasturni saqlab qolish uchun Fayl - kompyuteringizga saqlang – ko'rsatmasini bajaramiz.



SCRATCH dasturi yordamida interaktiv loyihalar yaratish.

Interaktivlik, shartlar va o'zgaruvchilar

Interfaollik deganda biz turli muhitlarga tegishli obyektlar o'rtasidagi o'zaro ta'sir o'tkazish imkoniyatini tushunamiz. Masalan, agar biz yozgan dasturda Scratchdan mushuk Scratchdan sichqonchani ushlasa va u bunga reaksiya qilsa va qochib ketsa, bu interaktivlik deb hisoblanmaydi. Mushuk ham, sichqon ham bir xil muhitning obyekti. Ammo agar Scratchdan olingan mushuk haqiqiy odamning harakatlariga (masalan, foydalanuvchi tomonidan biron bir tugmachani bosish) ta'sir ko'rsatsa, bu allaqachon interaktivlik, chunki narsalar turli xil muhitlarga tegishli.

Scratch ish sohaida dasturchi mushukning xatti-harakatlarini nazorat qila olishiga ishonch hosil qilish talab qilinadi. Agar siz bunday skriptni yozsangiz, u ishga tushirilgandan so'ng sichqonchani chap tugmachasini bosib tursangiz, obyekt ko'rsatgichni ta'qib qiladi.



Sichqoncha buyrug'i bosilganmi? Buyruqlar guruhlangan sensorlar bo'limida joylashgan bo'lib, ularning maqsadi obyekt muhitida belgilangan o'zgarishlarning mavjudligini tekshirishdir. Misol uchun, x sichqonchasi buyrug'i x o'qi bo'ylab sichqonchani kursori qayerda joylashganligini kuzatadi. Nazorat tuzilishi har doim, agar har doim buyrug'i aylanish jarayoniga o'xshash bo'lsa, bu

dizayn siklni shart bilan birlashtiradi. Bunday sikl faqat shart to'g'ri bo'lganda amalga oshiriladi.

Ushbu ssenariyni tuzing. Obyekt xususiyatlariga kattalikda "faqat chapdan o'ngga burilish" tugmachasini kiritish tavsiya etiladi. Keyin voqea joyiga yana bir obyekt qo'shing, birinchi obyektning skriptini nusxasini oling va ikkinchisiga o'tkazing. Natijada, sichqoncha yurgizgandan keyin ikkita obyekt ish sohada "yurishi" kerak.

Biz dasturni quyidagicha murakkablashtiramiz. Obyektlar bir-biridan uzoqlashmasin, lekin bir-biridan uzoqroq bo'lsin: birinchisi, kursordan keyin, ikkinchisi esa ishlaydi. "Faoliyat"ni obyektlar o'rtasida almashtirish bo'sh joy satridan foydalanib amalga oshishiga imkon beradi. Bunday dasturni qanday dasturlash mumkin? Agar biron bir dasturlashning juda muhim tarkibiy qismi ya'ni o'zgaruvchilar haqida bilmasangiz, bu savolga javob berish juda qiyin bo'ladi. O'zgaruvchilar har qanday ma'lumotlar saqlanadigan yacheykalardir (raqam, satr va boshqalar). Bundan tashqari, dasturni bajarish paytida yacheykalar tarkibi o'zgarishi mumkin.

O'zgaruvchilar har qanday ma'lumotlar saqlanadigan kattaliklardir (raqam, satr va boshqalar). Bundan tashqari, dasturni bajarish paytida kattaliklar tarkibi o'zgarishi mumkin. Biror nom ostida o'zgaruvchi katak yarating. Buni amalga oshirish uchun tugmachalarni bosing, so'ng Scratch buyruqlari ichidan O'zgaruvchi yaratish tugmachasini bosing, ochilgan muloqot oynasida uning ismini kiriting va OKni bosing. Shundan so'ng biz o'zgaruvchining o'zi va uni o'zgartirish uchun bir nechta buyruqlarga ega bo'lamiz. Bizga ushbu o'zgaruvchi kerak, shunda uning qiymatiga qarab, birinchi obyekt yoki ikkinchisi harakatlanadi. Agar yuruvchi=0 bo'lsa, unda birinchi obyekt siljiydi va agar o'zgaruvchi teng bo'lsa, ikkinchisi. O'zgaruvchining qiymatini o'zgartirish bo'shliqni bosganingizda yuz beradi. Dasturning ushbu qismi ikkala obyektga ham tegishli bo'lganligi sababli, ssenariyni sahna obyektiga qo'yib qo'yish o'rinli bo'ladi (garchi bu shart emas). Mana senariy:



Bo'sh probel o'rniga siz birinchi skript buyrug'ining ochiladigan ro'yxatida boshqa kalitni tanlashingiz mumkin. Quyidagi murakkab agar - yoki konstruktsiya amalga oshadi. Bu quyidagicha ishlaydi: agar qisman shart to'g'ri bo'lsa, u holda o'rnatilgan buyruqlar bajariladi, agar bo'lmasa, yoki keyin o'rnatilgan buyruqlar bajariladi. Ikkala tarmoq ham birgalikda bajarilmaydi, ular bir-birini istisno qiladi.

Endi agar bayonotidagi shartni ko'rib chiqing. Yashil buyruqlar operatorlarga murojaat qilishadi. Operatorlar  bu ma'lumotlar bilan biror amallarni bajaradigan buyruqlar. Masalan, ular matematik operatsiyalarni bajaradilar yoki raqamlarni bir-biri bilan taqqoslaydilar. Bunday holda, tenglikni taqqoslash operatori tanlanadi. U "=" belgisining chap tomonidagi ifoda o'ngdagi ifoda bilan tengligini tekshiradi. Agar "ha" bo'lsa, keyin kod ishlaydi, agar "yo'q" bo'lsa, ishlamaydi. Bizning holatda, ishlaydigan o'zgaruvchining qiymati 0 bilan taqqoslanadi. Biz o'zgaruvchiga hech narsa bermaymiz, lekin odatdagidek 0 qiymati berilgan, shuning uchun biz bunga majbur bo'lmaymiz. Bundan tashqari, agar to'satdan o'zgaruvchi nolga teng bo'lmasa, unda yoki tarmog'i faollashadi. Ushbu skript qanday vazifani bajaradi? Shuni korib chiqamiz. Agar o'zgaruvchining qiymati 0 bo'lsa, u 1 ga o'zgaradi. Aks holda, u 0 ga o'zgaradi. Bu tugmani bosib chiqadi bo'sh joy o'zgaruvchan o'zgaruvchining qiymatini o'zgartiradi. Biroq, dasturni amalga oshirishda hech qanday o'zgarishlarni ko'rmaymiz. Haqiqatan ham, obyektlar hali ham "bilishmaydi", chunki o'zgaruvchining qiymati ularning harakatlariga bog'liq. Yuqoridagi skriptni yarating va birinchi obyektga o'ting.

O'zgaruvchining qiymatini tekshirishni tashkil qilish kerak. Buning uchun oddiy ko'rsatmalar juda yaxshi. Bu erda birinchi va ikkinchi obyektlar uchun olingan stsenariylar



Ularning orasidagi farq faqat o'zgaruvchining qiymati taqqoslanadigan holda bo'ladi: nol yoki bir bilan. pni bosganingizda uning qiymati o'zgaradi. Natijada, dastur ishlayotganida, biz "faol" obyektни o'zgartirishimiz mumkin. Dasturni yakunlang va u qanday ishlashini ko'ring. Interfaollik bilan o'ynashni davom ettirishingiz mumkin. Muayyan tugmachalar bosilganda obyektlar tashqi ko'rinishini o'zgartirishi mumkin. Misol uchun, shunday:



Siz imkoniyatlari harakat qilib ko'rishingiz mumkin. Dasturni Scratch muhitida tuzing, unda foydalanuvchi obyektни klaviaturada o'q bilan boshqarishi mumkin va boshqa tugmachalarni bosganda, obyekt "turli fikrlarni" o'ylaydi.

Scratchda chizish. Sichqoncha bilan chizish Scratch dasturlash muhitida siz Turtle muhitida ham chizishingiz mumkin. Biroq Scratch juda ko'p funksional vosita bo'lib, siz ish sohada tasvirlarni namoyish qilmaydigan dasturlarni yozishingiz mumkin. Scratchda masalan, foydalanuvchi dasturni amalga oshirish jarayonida (ya'ni, ishga tushirish tugmasini bosgandan so'ng) ish sohasi tasvirni yaratganda, interaktiv chizilgan dasturni dasturlashingiz mumkin. Turtleda dasturning foydalanuvchi bilan o'zaro aloqasini faqat dialog oynalari orqali tashkil qilish mumkin edi. Sichqoncha va klaviatura Scratchda ishlatilishi mumkin. Ushbu imkoniyatlarning vizual tasviri uchun biz Scratch kutubxonasidan chizilgan qalam obyektini (qalamni chizish) import qilamiz, u obyektlar katalogida (obyektlar). Ushbu obyekt tayyor skript bilan paydo bo'ladi:



Mushukning sahnasidan bering va chizilgan qalam qo'shing. Dasturni ishga tushiring va bu qalam nima ekanligini bilib olasizmi? Dasturni amalga oshirishni to'xtating. Endi qalam skriptini batafsil tahlil qilamiz. Dastur ishga tushirilgandan so'ng, qalamning qalinligi qalam o'lchamini belgilash buyrug'i tufayli 5 nuqtaga ko'tariladi. Qalam tugmasini bosganingizda ko'rsatiladigan buyruqlar guruhida. Boshqacha qilib aytganda, siz mutlaqo biron bir obyekt (mushuk, odam, to'p va boshqalarni) chizish uchun foydalanishingiz mumkin. Keyinchalik, skriptda sikl har doim ishlatiladi, shuning uchun dastur ishlayotgan paytda doimiy ravishda amalga oshiriladi. Har doim quyidagicha amalga oshiriladi: agar sichqonchani chap tugmasi

bosilsa, qalam ko'rsatgich joylashgan joyga o'tadi va qalam tushadi; agar tugma bosilmasa, qalam ko'tariladi. Qalam ish soha ustiga tushganda, uning harakati izlari qoladi, ya'ni sichqonchani harakatga keltirib, biz qalam bilan harakat qilamiz va uni pastga tushirilgan holatda qalam bilan bog'laymiz. Natijada, dasturni amalga oshirishda sichqonchani tugmasi bosilganda qalam chiziladi va bosilganda chizilmaydi. Sichqoncha kursori joylashgan joyda chizilgan. Bundan tashqari, dasturni ishga tushirishda ish sohani tozalaydigan obyekt sahnasi uchun skriptni qo'shing: (yashil bayroq) → Clear.

Klaviatura yordamida chizish

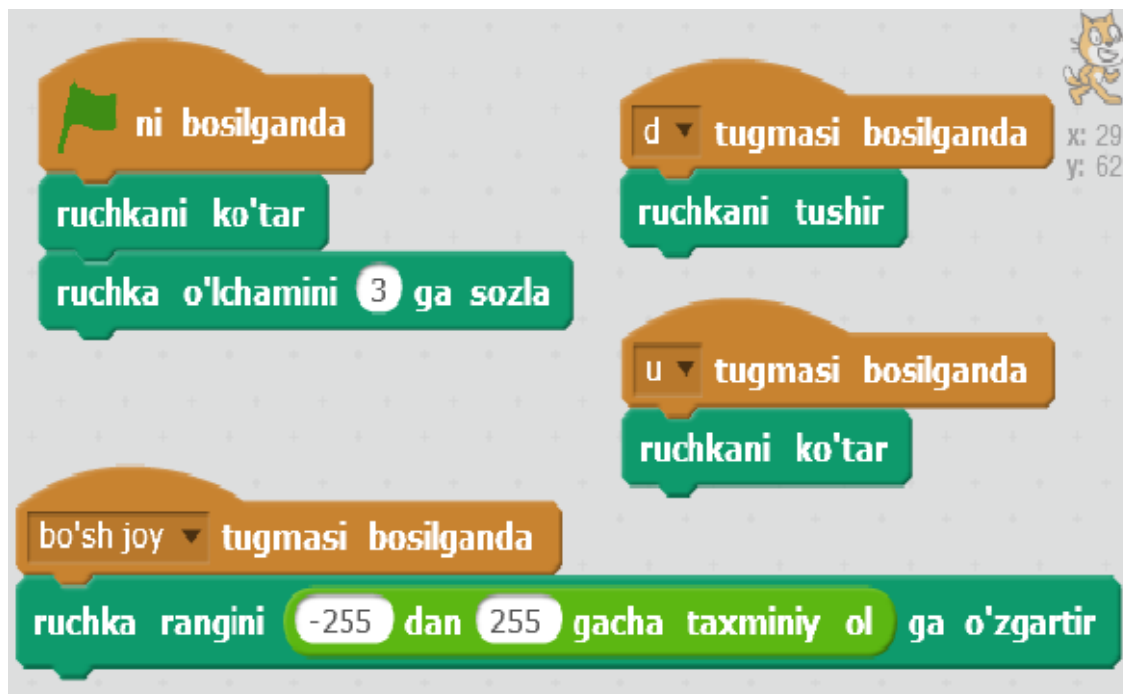
Endi biz "chizish" skriptini o'zingiz yaratamiz. Buning uchun sahnaga yana bir qalam qo'shing, mavjud dasturni olib tashlang va uni qalam deb ataymiz. Biz uni klaviatura yordamida boshqaramiz.



Bunday skriptlar qalamni klaviaturadagi o'qlar yordamida ish sohasi bo'ylab harakatlanishga imkon beradi. Turtleni eslang: toshbaqa yuqoriga qaraganida, uning yo'nalishi 0, o'ng, keyin 90 va boshqalar edi. Skriptlarni tuzing va qalam ish sohada aylanmasligi uchun obyektning xususiyat yacheykasida "qaytmaslik" rejimiga o'tishni unutmang.

Agar siz endi dasturni ishga tushirsangiz va qalamni o'qlar bilan boshqarsangiz, biz ish sohasi qanday harakat qilayotganini ko'rib chiqamiz. Biroq, hech qanday iz qoldirmaydi, chunki qalam yo'q edi. Ushbu kamchilikni tuzatish uchun siz qalam uchun yana bir

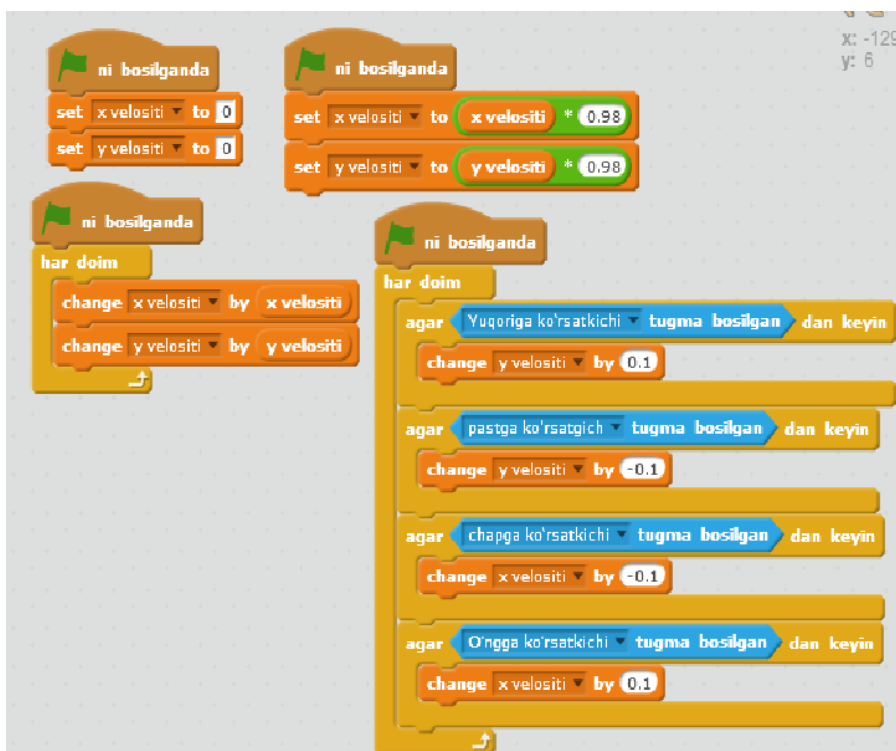
kichik skriptni qo'shishingiz kerak: (yashil bayroq) ustiga bosilganda → qalamni tushiring. Buni sinab ko'ring, lekin bu skript endi tuzatilishini yodda tuting. Endi qalam nafaqat harakat qilish, balki chizish ham kerak. Bir nechta kichik stsenariylarni qo'shib, qalam uchun dasturni murakkablashtiramiz:



Endi dastur ishga tushirilganda, qalam har doim ko'tarilishi kerak. U tugmachasini bosganingizda qalam yana ko'tariladi. Bularning barchasi tufayli qalamni ish sohaga tortmasdan ko'chirishimiz mumkin, keyin qalamni tushirish va chizish kerak bo'lsa. Bizning qalamimiz qalam rangini belgilash uchun buyruqlar kombinatsiyasidan foydalangan holda tasodifiy rangga bo'yalgan bo'lishi mumkin.

Ushbu dastur qanday ishlashini tushunish uchun fizikani biroz tushunishingiz kerak. Biz gorizontal ravishda (masalan, to'p) obyektini itarganimizda, odatda, bir muncha vaqt o'tgach to'xtaydi. Bu tuproq ishqalanish tufayli sodir bo'ladi. Agar biz doimo to'pni itarsak, uning harakat tezligi asta-sekin ortadi. Obyekt uchun dastur friction_marble bu jismoniy qonunlarni simulyatsiya qiladi. O'qni qanchalik ko'p bossak, to'p qanchalik tez rivojlanadi. Ta'sir to'xtatilganda uning tezligi asta-sekin kamayadi, undan keyin to'p to'xtaydi. Bu dasturda nima bilan erishiladi? Dasturda ikkita o'zgaruvchi x velocity va y velocity mavjud, ularning qiymatlari

dasturni ishga tushirgandan so'ng darhol nolga teng. Bu o'zgaruvchilar qiymatlari doimiy (har doim) ta'sir, qancha birliklari obyekt o'rnini o'zgaradi. OK tugmalari bosilganda, o'zgaruvchilar qiymatlari o'zgaradi. Muayyan o'q qanchalik ko'p bosilsa, o'zgaruvchining qiymati noldan farq qiladi. Natijada, x va y o'zgaruvchan birliklar soni ham oshadi. Bu obyektning harakatlanish tezligini oshirish ta'sirini yaratadi. Boshqa tomondan, ta'sir bo'lmasa (kalitlar bosilmasa), o'zgaruvchining qiymati asta-sekin nolga yaqinlashadi. Bunga 0 dan 1gacha bo'lgan kasr sonini ko'paytirish orqali erishiladi (bizning holatlarimizda o'zgaruvchilar 0.98 bilan ko'paytiriladi).



Scratch dasturida spraytlarni o'zgartirish dastur kodi

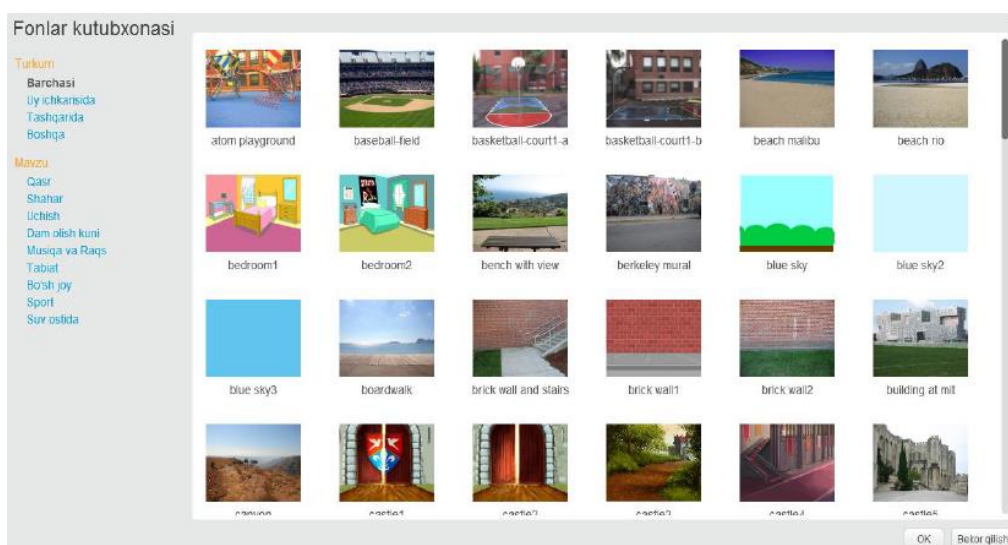
Orqa fon o'zgarishi

Tasavvur qilinga, foydalanuvchi tomonidan boshqariladigan qahramon xonadan xonaga ko'chirilishi mumkin bo'lgan o'yinni yaratamiz. Shu bilan birga, ma'lum bir xonadan qahramon faqat bir yoki bir nechtasiga tushishi mumkin, ammo barchasi hammasi emas. Misol uchun, xonalarning joylashuvi sxemasi shunday bo'lsin:



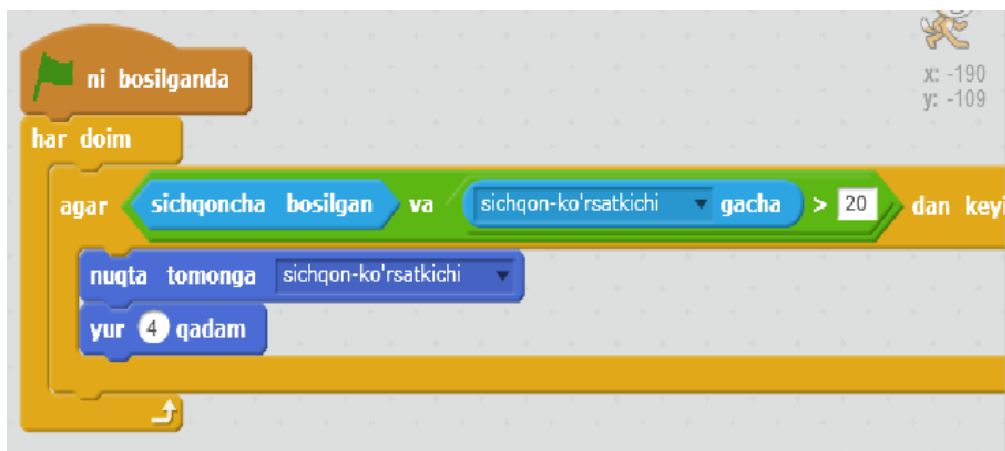
Scratch dasturida xona uchun fon tanlash

1-xonasidan 2-xonasiga kirishingiz mumkin. Ikkinchi xonadan birinchi va uchinchi va hokazolarga o'tish mumkin. Keyinchalik o'tish amalga oshiriladigan voqea qahramonning sahnaning tegishli chekkasida joylashgan joyi bo'ladi. Shunday qilib, birinchi xona uchun bu o'ng tomon bo'ladi. Scratch dasturlash muhitida sahnaning foniga uning nomi (Agar dastur kodi sahna obyekt uchun tuzilgan bo'lsa) yoki fon raqami (boshqa obyektlar uchun) murojaat qilinadi. Bizning dasturimizda fonni to'g'ri tartibga solish va ularni nomlash muhimdir.



Scratch dasturida fonlar kutubxonasi

Keling, ba'zi obyektlarni qahramon sifatida tanlaymiz va uni harakatga keltiramiz. Obyektning boshqariladigan harakati sichqoncha va klaviatura yordamida amalga oshirilishi mumkin. Agar sichqonchani tanlasangiz, kod shunday bo'lishi mumkin:



Bu yerda, foydalanuvchi sichqonchani bosganda, obyekt kursga qarab harakat qiladi. Shuni esda tutingki, Scratch ish sohasining o'lchami 480x360 pikselga teng va koordinata tizimining boshlang'ich nuqtasi markazda joylashgan. Shuning uchun o'ng tomonda $x=qiymati\ 240$, chap $x=-240$. Yuqori tomoni: $y=180$; pastki chet: $y=-180$. Unga ruxsat bering obyektning mos keladigan koordinatasi 200 yoki 150 moduliga teng bo'lsa, fon o'zgaradi. Bu boshqaruvning qulayligi bilan bog'liq. Obyektni 1 xonadan (laboratoriya) 2 xonaga ko'chirish algoritmi (yotoqxonasi) quyidagi kabi bo'lishi mumkin.

Qahramon uchun:

- Obyektning x koordinatasi 200 ga teng bo'lganda, u tegishli xabarni yuboradi (masalan, "laboratoriyadan yotoqxonaga").
- X koordinatasining qiymati teskari tomonga o'zgaradi va y bir xil bo'ladi. Bu boshqa xonaga kirish ta'sirini yaratadi.

Sahna uchun:

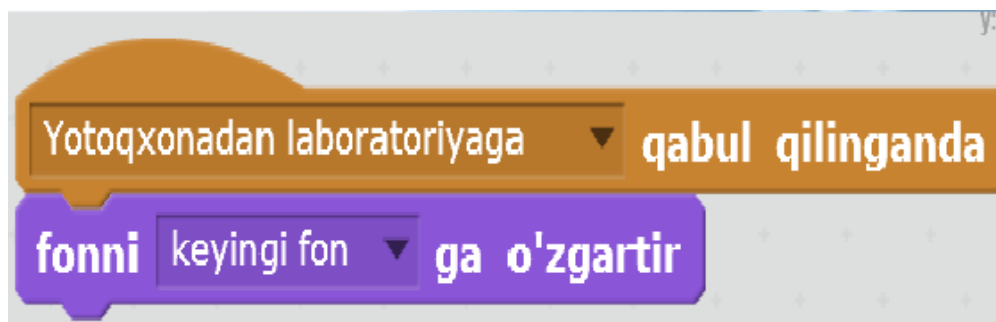
- Sahna uchun qabul qilingan xabarga qarab, postga mos keladigan fon o'rnatiladi.

Obyekt uchun dastur kodi:



Scratch dasturida xona mushuk harakatini dasturi

Sahna uchun dastur kodi:



Boshqa xonalarga o'tish ssenariylari qadriyatlar bundan mustasno. Xonadan xonaga o'tishning sakkizta kombinatsiyasi bo'lishi kerak. Ushbu o'tishlarni o'zingiz dasturlang. Yordam uchun quyidagi jadvaldan foydalaning.

Mustaqil bajarish uchun topshiriqlar.

1. Scratch muhitida foydalanuvchi mushukni oshirish yoki kamaytirish uchun qancha foiz berishini so'rashi mumkin bo'lgan dastur tuzing. Keyin Ish soha ustiga obyekt hajmini o'zgartirish edi.

2. Dasturni Scratchga yozing. Foydalanuvchining rasmning yon tomonlari sonini va tomonlar orasidagi burchakni so'rab, so'ngra Ish soha ustiga olingan rasmni torting. Maslahat. Ikkinchi vazifada siz ikkita o'zgaruvchini ishlatishingiz kerak.

FOYDALANILGAN ADABIYOTLAR

Asosiy adabiyotlar

1. Филиппов В.И. Методика использования робототехники для формирования универсальных учебных действий у обучающихся во внеурочной деятельности по информатике. М.: 2020. № С. 67-105
2. Патаракин Е. Д. Учимся готовить в среде Скретч. М.: Просвещение, 2012
3. Справочник учителя. Упражнения по робототехнике. Образовательная теория робототехники и практическое руководство, разработанное MiniRobots, S.L, 2011. http://ozgsch4.edumsko.ru/images/users-files/ozgsch4/robot/moway_spravochnik_prepodavatelya.pdf
4. Шапошникова С. Введение в Scratch. Цикл уроков по программированию для детей // Лаборатория юного линуксоида. <http://younglinux.info/>
5. Moway. Minirobots // Minirobots S.L. <http://mowayrobot.com>
6. Филиппов В.И. Методика использования робототехники для формирования универсальных учебных действий у обучающихся во внеурочной деятельности по информатике. М.: 2020. № С. 67-105
7. Филиппов В.И. Методика использования робототехники для формирования универсальных учебных действий у обучающихся во внеурочной деятельности по информатике Дис.канд.пед.наук: М. 2009 № С 454.
8. Sultanov, R.O., (2021). O'qitishning raqamli texnologiyasi masalalari. Academic research in educational sciences, 2(CSPI conference 3), 804-807
9. Sultanov, R., Xalmetova, M.(2021). Ikki g'ildirakli transport robotlari harakatini dasturlash. Academic Research in Educational Sciences, 2(2), 108-114.
10. Sultanov, R.O., (2023). Talabalarning loyihalash faoliyatini tashkil etishda Arduino platformasida robototexnik uskunalardan foydalanish. Kasb-hunar ta'limi, 3-son, 175-183.

11. Sultanov, R.O., Xurramov A.J., (2023). Robototexnika kursini takomillashtirish (kelajak kasblari bo'yicha mutaxassislar tayyorlash uchun). Kasb-hunar ta'limi, 2-son, 40-43.

12. Хуррамов, А.Ж., Ражабов, О.Т., Султонов, Р.О.,(2021). Ta'lim jarayonida animatsiya va kompyuter grafikasidan foydalanish. Academic research in educational sciences, 2(11), 1382-1388.

Qo'shimcha adabiyotlar

1. Mirziyoyev Sh.M. Buyuk kelajagimizni mard va olijanob xalqimiz bilan birga quramiz. 2017.
2. Mirziyoyev Sh.M. Qonun ustuvorligi va inson manfaatlarini ta'minlash – yurt taraqqiëti va xalq farovonligining garovi. 2017.
3. Mirziyoyev Sh.M. Erkin va farovon, demokratik O'zbekiston davlatini birgalikda barpo etamiz. 2017.
4. Usmonov J.T., Xujaqulov T.A. Ma'lumotlar bazasini boshqarish tizimi// o'quv qo'llanma. - T. : Aloqachi, 2018. – 96 b.
5. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL и JavaScript. – СПб.: Питер, 2011. –496 с.
6. Макфарланд Д. Новая большая книга CSS – СПб.: Питер, 2016. –720 с.
7. Гольцман В. MySQL 5.0. Библиотека программиста. Питер; Санкт-Петербург; 2010.

Internet saytlari

1. Lego Duplo // URL: <https://www.lego.com/ru-ru/themes/duplo/>
2. Учись учиться // URL: <https://education.lego.com/ru-ru/support/learntolearn/>
3. LEGO Education WeDo 2.0 // URL:<https://education.lego.com/ru-ru/product/wedo-2>
4. <https://www.tinkercad.com/>
5. <https://www.lego.com/>
6. <https://www.arduino.cc/>
7. LEGO MINDSTORMS Education EV3 // URL: <https://education.lego.com/ru-ru/product/mindstorms-ev3>

MUNDARIJA:

KIRISH	2
ROBOTOTEXNIKA VA UNING QURILMALARI BILAN TANISHISH.5	
ARDUINO DASTURI BILAN TANISHISH	16
BIR NECHTA SVETODIODLAR VA PAYKASIZ MAKET PLATALARI	
BILAN ISHLASH ASOSLARI.	26
ARDUINO PLATFORMASIDA SVETAFOR YARATISH.	30
DRAYVER MOTORI ORQALI ODDIY MOTOR BILAN ISHLASH	34
SERVOMOTOR BILAN ISHLASH. BOSQICHLI MOTOR BILAN	
ISHLASH.	39
DISPLEY QURILMASI. TASVIRLARNI AKS ETTIRISH.	43
MIKROKONTROLLERLARNI O'RGANISH	47
ARDUINO VA KOMPYUTER BILAN ALOQA USULLARI.	51
ARDUINO ASOSIDAGI HARAKATLANUVCHI PLATFORMA.	59
ROBOTOTEXNIK QURILMALARNI DASTURLASH.	71
DATCHIKLARNI DASTURLASH	81
SENSOR YORDAMIDA HARAKATLANADIGAN ROBOT MASHINA	
YIG'ISH.	87
SCRATCH DASTURIDA O'YIN YARATISH TEXNOLOGIYASI.	98
SCRATCH DASTURI YORDAMIDA INTERAKTIV LOYIHALAR	
YARATISH. INTERAKTIVLIK, SHARTLAR VA O'ZGARUVCHILAR	
.....	106
FOYDALANILGAN ADABIYOTLAR.	118

Sultanov R.O., Matyakubov K.K.

ROBOTOTEXNIKA ASOSLARI

Amaliy mashg'ulot uchun

O'quv qo'llanma

Muharrir:	X. Taxirov
Tehnik muharrir:	S. Melikuziva
Musahhih:	M. Yunusova
Sahifalovchi:	A.Ziyamuhamedov

Nashriyot litsenziya № 2044, 25.08.2020 й

Bichimi 60x84¹/₁₆. "Cambria" garniturası, kegli 16.

Offset bosma usulida bosildi. Shartli bosma tabog'i 7,75. Adadi
100 dona. Buyurtma № 508

Zebo prints MCHJda chop etildi.

Manzil: Toshkent shahar, Yashnobod tumani,
22-harbiy shaharcha

QAYDLAR UCHUN

QAYDLAR UCHUN