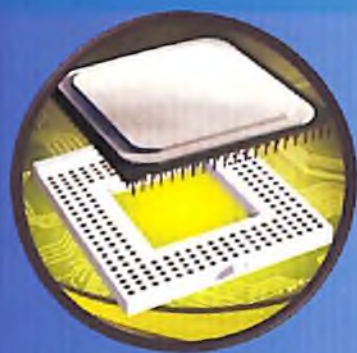


X.YU.ABASKONOVA,
M.B. MIRZAYEVA, S.S. PARSIYEV



MIKROPROTSESSOR

O‘ZBEKISTON RESPUBLIKASI AXBOROT TEXNOLOGIYALARI
VA KOMMUNIKATSIYALARINI RIVOJLANTIRISH VAZIRLIGI

O‘ZBEKISTON RESPUBLIKASI
OLIV VA O‘RTA MAXSUS TA‘LIM VAZIRLIGI

MUHAMMAD AL-XORAZMIY NOMIDAGI TOSHKENT
AXBOROT TEXNOLOGIYALARI UNIVERSITETI

X.YU.ABASXONOVA, M.B. MIRZAYEVA, S.S. PARSIYEV

MIKROPROTSESSOR

O‘zbekiston Respublikasi Oliy va o‘rta maxsus ta‘lim vazirligi
tomonidan o‘quv qo‘llanma sifatida tavsiya etilgan.

Toshkent – 2021

UO‘K: 004.3(075.8)

KBK: 32.973

X.YU.Abasxanova, M.B. Mirzayeva, S.S. Parsiyev. Mikroprotessor. (O‘quv qo‘llanma). – T.: «Nihol print» OK, 2021. – 200b.

ISBN 978–9943–7343-0-2

Telekommunikatsiya sohasi uchun yuqori mutaxassislarni tayyorlash uchun mikroprotessorlarni bilish, dasturiy boshqariladigan mikroprotessorli tizimlarni yaratish, tahlil etish bo‘yicha bilimlar zarurdir. Mavjud davrda hisoblash tizimlari juda tez odimlar bilan rivojlanib kelmoqda. Shuning uchun “Mikroprotessor” fani boshqa fanlarga qaraganda o‘zining tezkor o‘zgarishi bilan farqlanadi. Ushbu o‘quv qo‘llanmada mikrokontroller va uning arxitekturasi, tuzilishi, turlari, tashqi qurilmalar bilan ma’lumot almashishi, dasturlash asoslari, dasturiy va apparat vositalarini yaratishni o‘rganish masalalarini o‘z ichiga olgan. Mazkur fan talabalarga maxsus fanlarni o‘zlashtirishda, keyinchalik ishlab chiqarish, loyihalash va tadqiqot ishlarida kerak bo‘ladigan asosiy negiz tushunchalarini o‘rgatadi.

O‘quv qo‘llanmada “Mikroprotessor” fanini o‘zlashtirish jarayonida talabalar tomonidan o‘zlashtirishi kerak bo‘lgan bilimlar: mikrokontroller arxitekturasi va protessorli yadrosi, buyruqlar tizimi, mikrokontrollerlarni tashqi muhit bilan aloqasini tashkillashtirish; o‘zlashtirgan bilimlarni mikrokontrollerlarni tanlash va dasturiy muhitga moslashtirish, dasturlash, dasturni qurilmaga yuklash, ularni parametrlarini kiritish, tadqiq etish va dasturiy boshqariladigan qurilmalarni ishlatishni; olingan natijalarni sohaga tadbiq etish haqidagi ma’lumotlar keltirilgan.

O‘quv qo‘llanma 5311300 - “Telekommunikatsiya texnologiyalari” (Telekommunikatsiya, teleradioeshittirish, mobil tizimlar) yo‘nalishi bo‘yicha oliy o‘quv yurtlari talabalari, hamda kredit ta’lim tizimi bo‘yicha ta’lim olayotgan talabalarning “Operatsion tizimlar”, “O‘rnatilgan tizimlar” v.b. fanlarini tayanch fani hisoblanadi va shu fanlar bilan mantiqan bog‘langan.

O‘quv qo‘llanma Muhammad al-Xorazmiy nomidagi Toshkent axborot texnologiyalari universiteti imiy-uslubiy kengashining qarori bilan chop etishga tavsiya etildi.

UO‘K: 004.3(075.8)

KBK: 32.973

Taqrizchilar: U.B.Amirsaidov; B.R.Yakubov.

Mas’ul muharrir: A.A.To‘laganov; A.Ganiev.

Musahhihlar: N.X.Rahimova.

ISBN 978–9943–7343-0-2

© «Nihol print» OK nashriyoti, 2021.

KIRISH

Ilm-fan va texnika yutuqlarini keng qoʻllagan holda iqtisodiyot tarmoqlariga, ijtimoiy va boshqa sohalarga zamonaviy innovatsion texnologiyalarni tezkor joriy etish Oʻzbekiston Respublikasi jadal rivojlanishining muhim sharti hisoblanadi.

Jamiyat va davlat hayotining barcha sohalari shiddat bilan rivojlanayotgani islohotlarni mamlakatimizning jahon sivilizatsiyasi yetakchilari qatoriga kirish yoʻlida tez va sifatli ilgarilashini taʼminlaydigan zamonaviy innovatsion gʻoyalar, ishlanmalar va texnologiyalarga asoslangan holda amalga oshirishni taqozo etadi.

Shu bilan birga, oʻtkazilgan tahlil ishlab chiqarishni modernizatsiya, diversifikatsiya qilish, uning hajmini oshirish hamda ichki va tashqi bozorlarda raqobatbardosh mahsulotlar turlarini kengaytirish borasidagi ishlar lozim darajada olib borilmayotganini koʻrsatdi.

Xususan, bu borada koʻplab koʻrsatkichlarning mavjud emasligi va ishlar samarali muvofiqlashtirilmagani sababli mamlakatimiz soʻnggi yillarda nufuzli va obroʻli xalqaro tuzilmalar tomonidan tuziladigan Global innovatsion indeks reytingida ishtirok etmayapti.

Iqtisodiyot va ijtimoiy soha tarmoqlarining ilmiy muassasalar bilan oʻzaro hamkorligi darajasi pastligi, vazirlik va idoralar, shuningdek, mahalliy davlat hokimiyati organlarining innovatsion rivojlanish sohasidagi faoliyati lozim darajada muvofiqlashtirilmayotgani bu boradagi birinchi navbatdagi maqsadlar va vazifalarga erishish imkonini bermayapti.

Mamlakatimizni ijtimoiy-iqtisodiy rivojlantirish boʻyicha ustuvor vazifalarga muvofiq kadrlar tayyorlashning mazmunini tubdan qayta koʻrib chiqish, xalqaro standartlar darajasida oliy maʼlumotli mutaxassislar tayyorlashga zarur shart-sharoitlar yaratish maqsadida Oʻzbekiston Respublikasi Prezidentining 2017-yil 20-apreldagi “Oliy taʼlim tizimini yanada rivojlantirish chora-tadbirlari toʻgʻrisida”gi PQ-2909-son qaroroi qabul qilindi. Mazkur qaror bilan oliy taʼlim darajasini sifat jihatidan oshirish va tubdan takomillashtirish, oliy taʼlim muassasalarining moddiy-texnika bazasini mustahkamlash va modernizatsiya qilish, zamonaviy oʻquv-ilmiy laboratoriyalari, axborot-kommunikatsiya texnologiyalari bilan jihozlash boʻyicha Oliy taʼlim tizimini 2017-2021-yillarda kompleks rivojlantirish dasturi tasdiqlandi.

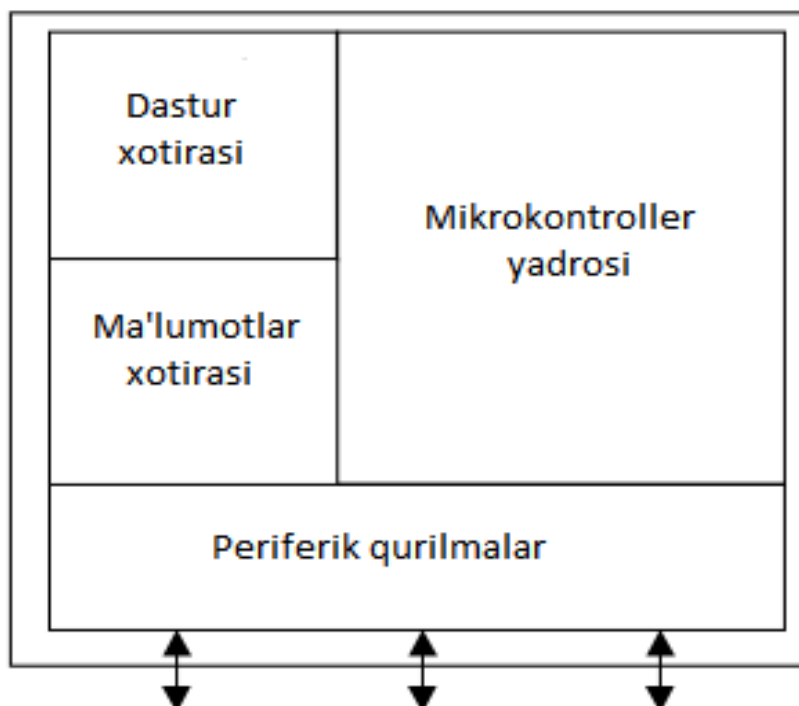
Tasdiqlangan qonunlarni bajarish jarayoni infokommunikatsion texnologiyalarni O‘zbekistonda rivojlanishning tegishli qonunlarni bajarish uchun keng yo‘l ochib berdi. Mavjud davrda O‘zbekistondagi telekommunikatsion aloqa tizimlariga juda katta masshtabdagi ishlarni bajarishga olib kelmoqda. Bu esa aholiga turli telekommunikatsion xizmatlarni yuqori saviyada amalga oshirishga olib kelmoqda.

Yangi texnologiyalar kiritish sharoitida mutaxassislar oldida texnologiya jarayonlarini o‘rnatish tarkibiy qismlarini qo‘llanilishi, zamonaviy texnologiyalar asosida tarmoq yaratish kabi masalalari tadqiqoti dolzarb desa bo‘ladi. Tasdiqlangan qonunlarni bajarish jarayoni infokommunikatsion texnologiyalari O‘zbekistonda rivojlanishning tegishli qonunlarini bajarish uchun keng yo‘l ochib berdi. Bu O‘zbekistondagi telekommunikatsion aloqa tizimlariga juda katta masshtabdagi ishlarni bajarish, aholiga turli telekommunikatsion xizmatlarni yuqori saviyada amalga oshirishning muxim omillaridir.

1. MIKROKONTROLLERLARNI TASHKILLASHTIRISH

1.1. Mikrokontroller arxitekturasi

Mikrokontroller bitta integral sxema ko‘rinishida amalga oshiriladigan hisoblash tizimidan iborat va o‘z ichiga yadro, dastur xotirasi, ma’lumotlar xotirasi, periferiya qurilmalari kabi asosiy bloklarni oladi (1.1- rasm).



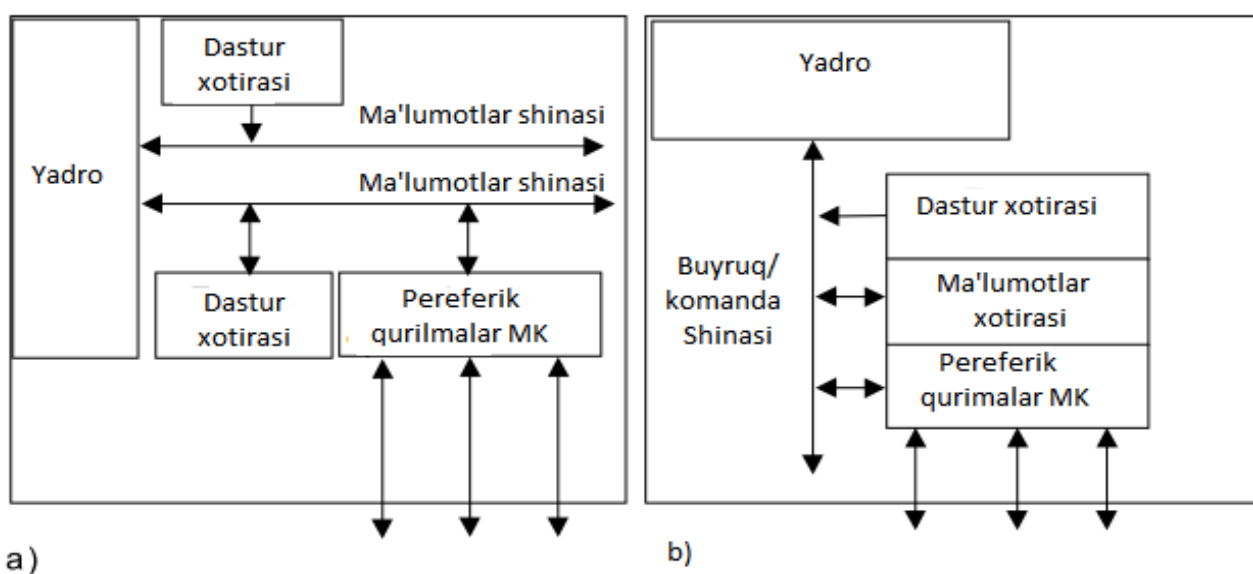
1.1- rasm. Mikrokontrollerning umumlashtirilgan tuzilmaviy sxemasi

Mikrokontroller yadrosi dastur tomonidan beriladigan boshqaruv jarayonini amalga oshiradi. Mikrokontrollerli yadro negizida integral sxemasini ishlab chiqaruvchi zavod tomonidan modulli xotira va periferiya qurilmalarining nomi ro‘yxati bo‘yicha turlicha bo‘lgan, lekin komandalar tizimi va ma’lumotlar almashinuvi sikli bo‘yicha o‘zaro moslashadigan mahsulotlar ishlab chiqariladi. Ushbu belgi bo‘yicha ko‘plab moslashadigan mikrokontroller (MK) mikrokontrollar turkumi deb ataladi.

Dastur xotirasi boshqaruvchi dasturni saqlash uchun mo‘ljallangan. Boshqarish jarayoni uchun zarur bo‘lgan ma’lumotlar ma’lumotlar xotirasida joylashadi.

Periferiy qurilmalari mikrokontrollerning tashqi ob'ektlari va qator boshqaruv funksiyalarini amalga oshiradigan apparat bilan birikishni ta'minlash uchun mo'ljallangan.

Mikrokontrollerlar, boshqa klasslarning hisoblash mashinalari kabi, garvard yoki prinston arxitektura asosida amalga oshiriladi (1.2-rasm). Garvard arxitektura asosida bajariladigan mikrokontrollerlarda dasturlar va ma'lumotlar foydalana olishning turli metodlaridagi mantiqiy bog'liq bo'lmagan xotira bloklarida joylashadi. Prinston arxitektura asosida bajariladigan mikrokontrollerlarda dasturlar va ma'lumotlar xotiraning umumiy blokida joylashishi mumkin, murojaat uchun foydalana olishning yagona metodidan foydalaniladi.



1.2- rasm. MK garvard (a) va prinston (b) arxitekturalari

Namunaviy va kristallga eng ko'p integratsiyalanadigan periferiya qurilmalarining mikrokontrolleriga quyidagi bloklar kiradi:

- mantiqiy signallar ko'rinishida keltirilgan ma'lumotlar almashinuvini amalga oshiradigan kiritish-chiqarish parallel raqamli portlar;
- vaqtli integrallar shakllantirilishini amalga oshiradigan va mantiqiy hodisalarni hisoblashni bajaradigan taymer-hisoblagichlar;
- vaqt bo'yicha bog'liq bo'lgan hodisalarni apparatli qayta ishlash uzellari;
- uzluksiz signallarni chiqarish va kiritishni amalga oshiradigan raqamli-analog va analog-raqamli o'zgartirgichlar;
- taqsimlanadigan tizimlarda ma'lumotlar almashinuvini amalga oshiradigan kiritish-chiqarishning ketma-ket portlari;

- uziluvchi hodisalarga xizmat ko'rsatish bloklari;
- ishlash ishonchligini oshirish vositalari.

MK har bir periferiya uzeli maxsus funksiyalar registri deb ataladigan uzelnig dasturiy qulay bo'lgan konfiguratsion registrida boshqariladigan kodlarni yozish yordamida rostdash imkoniyatiga ega bo'ladi. Rostdash qurilmaning (masalan, taymer va parallel porti razryadlaridagi ma'lumotlarni uzatish yo'nalishlarining razryadliligini va boshqalarni talab etadigan) ishlash rejimini tanlashni amalga oshirish imkonini beradi.

MKda joylashgan periferiya bloklarining tarkibi qurilmaning maqsadli vazifasiga bog'liq bo'ladi va ushbu turkumning mikrokontrollerida amalga oshiriladigan namunaviy vazifalari asosida ishlab chiqaruvchilari tomonidan aniqlanadi.

1.2. Mikrokontroller yadrosi

MK yadrosi tarkibiga protsessor, taktli generator va shina kontrolleri kiradi (1.3- rasm). Protsessor ikkilik kodi va komandalar ketma-ketligidan iborat bo'lgan dasturga muvofiq ushbu jarayonni boshqarish ko'inishida keltirilgan axborotni qayta ishlash jarayonini bevosita amalga oshiradi. Taktli generator MK uzellarida jarayonlar o'tishini sinxronizatsiyalovchi tayanch signallar ketma-ketligini tayanch impulslarning tashqi ketma-ketligi asosida shakllantirishni amalga oshiradi. Shinalar kontrolleri MKda komandlar bajarilishining turli bosqichlarini ichki shina bo'yicha taktlovchi va MK periferiya qurilmalari bilan ma'lumotlar almashinuvini tashkil etish uchun zarur bo'lgan ko'p fazali impulsi ketma-ketlikning tuzilishini amalga oshiradi.

Komandalar berilgan adreslar (katakchalar raqami) bo'yicha komandalar xotirasida joylashadi va o'z ichiga bajariladigan operatsiyalarni tavsiflaydigan boshqariluvchi kodlarni va berilgan operandlar (amallar bajariladigan ma'lumotlar)ni oladi.

Har bir MK komandalar ro'yxati va ularning formatlari xarakterlanadigan komandalarning muayyan tizimiga ega bo'ladi. Komandalar ro'yxati o'z ichiga ushbu MK protsessorida bajarilishi nazarda tutilgan amallar to'plamini oladi. Har qanday MK komandalar ro'yxatida amallarning to'rtta guruhiga ajratish mumkin:

- ma'lumotlar uzatish amallari (MKning boshqa dasturiy qulay elementlar o'rtasida);

- arifmetik amallar («Va», «Yoki», «Yoki» ni istisno qiladigan inversiya, turli siljishlar);

- boshqaruvni uzatish amallari (berilgan adres bo'yicha shubhasiz o'tish, operandlar tengsizligi yoki tenglik sharti bo'yicha o'tish, quyi dasturga o'tish va undan qaytarish va h.k.).

Komandalar formati amallar dasturining navbatdagi qadamida bajariladigan tipni, kirish va chiqish operandlarini, shuningdek dasturning quyidagi qadamda bajarilishi kerak bo'lgan komandalar adresini aniqlash imkonini beradi.

Bajariladigan komandalar tipi amallar kodi (KOP) beriladi.

Operandalarni berish uchun ularni lokalizatsiyalash metodlari (adreslash usullari) qo'llaniladi:

- noaniq bo'lgan: operand undan foydalana olishning bir xilligi bilan bog'liqligi (masalan, uning joylashish imkoniyatiga bog'liqligi) ko'rsatilmaydi;

- bevosita: kirish operand komandalarda (masalan, konstant topshirig'i maqsadida) joylashtiriladi;

- to'g'ri: komandada operand joylashgan ma'lumotlar xotirasidagi adres ko'rsatiladi;

- bilvosita: komandada operand joylashgan ma'lumotlar xotirasidagi katakchalar adresini o'z ichiga olgan ma'lumotlar xotirasidagi katakchalar adresi ko'rsatiladi (masalan, dastur uchastkasining bir necha marta takroranganda ketma-ket joylashgan ma'lumotlardan foydalana olishni tashkil etishda izlanayotgan ma'lumotlar adresini o'zgartirgan holda komandalar operanlining qiymatini o'zgartirish);

- nisbiy: komandada (masalan, noaniq beriladigan) ayrim kattalikka tuzilgan ma'lumotlar xotirasida katakchalar adresi ko'rsatiladi, izlanayotgan operand joylashgan ma'lumotlar xotirasidagi katakchalar adresini beradi (masalan, ma'lumotlar jadvalining elementiga murojaat qilganda jadval boshiga nisbatan siljishi bo'yicha izlanayotgan operandni aniqlash qulaydir).

Quyidagi bajariladigan komandalar adresi komandalarning ushbu vaqtda bajariladigan adresdan keyin keladigan dastur xotirasining adresi kabi noaniq beriladi, bu ko'pgina dasturlarda komandalar ketma-ketligining liniyali uchastkalarga ega bo'lish bilan tushuntiriladi. Sikllar, quyi dasturlar, shartlar bo'yicha vetalenieni va hakazolarni tashkil etishda uning aniq topshirig'i uchun komandalar qo'llaniladi. OPK boshqaruvni uzatishning muayyan amalini kodlaydi.

Ko'plab MK komandalar tizimiga bir, ikki, uch adresli va adressiz komandalarni (bitta operandalar komandasida adreslanadigan miqdori bo'yicha) kiritadi.

MKda komandalarni bajarish protsedurasi quyidagiga olib keladi.

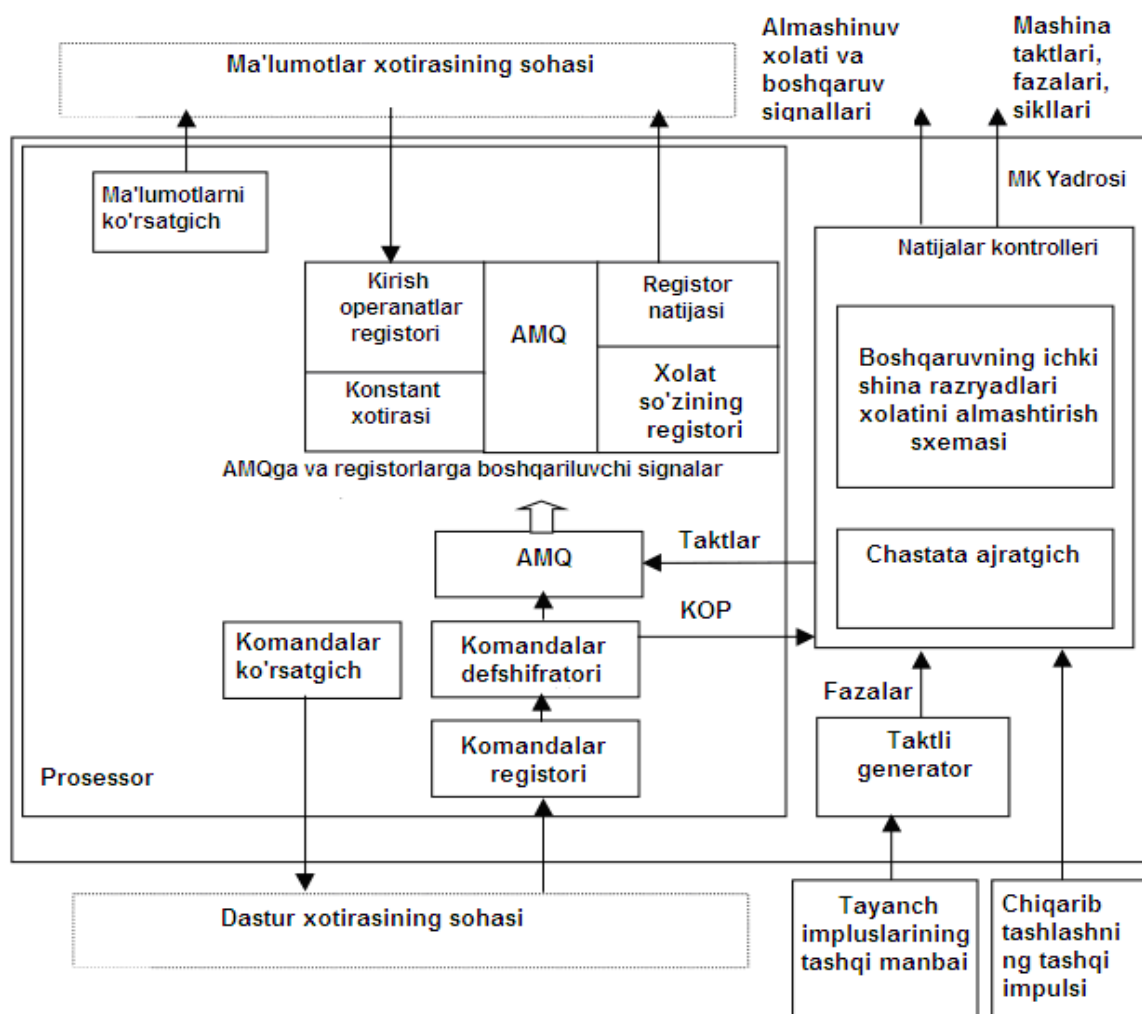
Chiqarib tashlash impulsi amalining tugashi bo'yicha MK yadro registrini intsiyalizatsiyalash amalga oshiriladi. Komandlar ko'rsatkichiga datslabki ishga tushirish adresi kiritiladi.

Komandalar ko'rsatkichidagi adres bo'yicha shina kontroller tomonidan shakllanadigan boshqariluvchi signallar ta'siri ostida dastur xotirasi sohasidan komandalar registriga dastur kontrolleri tomonidan bajariladigan navbatdagi komandalar yuklanadi.

Har qanday komandalar elementar harakatlar (mikroamallar) ketma-ketligini, operandlar amali uchun talab etiladigan miqdorni aniqlash, zarur operandlarni lokalizatsiyalashni aniqlash, ularni chiqarib tashlash, bajariluvchi bloklar uchun harakatlar kodini shakllantirish, amallar bajarilishi tugashini kutish, natijalarni lokalizatsiyalashni aniqlash, natijalarni kiritish, keyingi komandalar va qator boshqa komandalar adresini aniqlashdan iborat. Navbatdagi komandalarni bajarishda amalga oshiriladigan mikroamallarning muayyan ro'yxati uning OPK (opersiya kodi) ni aniqlaydi.

- Ma'lumotlar xotirasining sohasi
- Almashinuv holati va boshqaruvining signallari
- Mashina taktlari, fazalari, sikllari
- Ma'lumotlarni ko'rsatkich
- MK yadrosi
- Kirish operandlar registri
- Natijalar kontrolleri
- Konstant xotirasi
- Arifmetik mantiqiy qurilma (AMQ)
- Holat so'zining registri
- Boshqaruvning ichki shina razryadlari holatini almashtirish sxemasi
- AMQ ga va registrlarga boshqariluvchi signallar
- Taktlar chastota ajratgichi
- Komandalar ko'rsatkichi
- Komandalar deshifratori
- OPK
- Komandalar registri fazalar taktli generator

- Dastur xotirasining sohasi
- Tayanch impulslarning tashqi manbai
- Chiqarib tashlashning tashqi impulse



1.3- rasm. Mikrokontroller yadrosining tuzilmaviy sxemasi

Talab etiladigan mikroamalg protsessor sxemasini rostdash uchun boshqariluvchi signallar ketma-ketligidan foydalaniladi. Komandalar dasturi xotirasidan hisoblab chiqilgan OPK (opersiya kodi) deshifrlanadi va sinxronizatsiyalashning har bir takt bilan ishlab chiqiladigan, boshqariluvchi signallar to'plamining komandalarini ishlab chiqishning ushbu bosqichida zarur bo'ladigan shina kontrolleridan kelib tushadigan mikrodasturli avtomat (MPA)ga kelib tushadi.

Arifmetik va mantiqiy amallar protsessorida siljish, nollash va hokazolarni bajarish arifmetik-mantiqiy qurilma (AMQ) tomonidan ta'minlanadi. Xotira, konstant ma'lumotlarning ikki-o'ntalik tarzda keltirishda to'g'rilovchi kodni, bitlar ustida amallarni AMQ da bajarishda maska kodini ishlab chiqishni, shuningdek konstant kodlarini

berishni ta'minlaydi. Amallarni AMQda bajarilishining tugashi bo'yicha uning natijalar registriga kiritiladi, shuningdek protsessor holatining so'z registriga kiritiladigan va dastur bilan hisoblash va tahlil qilish uchun (masalan, arifmetik jihatdan to'lganligi sababli dasturning boshqa shohiga o'tishni tashkil etish uchun) qulay bo'lgan amallar natijalarining belgilari (to'ldirish, siljish, belgi va h.k.)ni shakllantiradi. Keyin umumiy holatda ma'lumotlar ko'rsatkichiga komandalar natijalarini joylashtirish zarur bo'lgan ma'lumotlar xotirasi katakchalarining adreslari ketma-ket kiritiladi (ushbu adreslar chiqish operandlarni adreslash maydonlaridan chiqarib tashlanadi) va operandalar natijalarning registrlaridan ma'lumotlar ko'rsatkichi tomonidan adreslanadigan ma'lumotlar xotirasi katakchalariga kiritiladi. Shuni ta'kidlash kerakki, qoidaga ko'ra, MK protsessorida operandlarni kiritishda avtoinkrementlash va xotiradan operandlar chiqarib tashlanaganda avtokamaytirish mexanizmlari bilan ta'minlangan ma'lumotlar xotirasida qo'shimcha ko'rsatkichdan foydalaniladi. Xotiradan foydalana olish metodi stekli deb ataladi, ma'lumotlar xotirasida ajratiladigan bunday manipulyatsiyalash uchun soha stek deb ataladi. Stekdan quyi dasturni tashkil etishda, xususan, uzilishni qayta ishlash quyi dasturdan foydalaniladi.

Chiqish natijalari joylashtirilgandan keyin komandalar ko'rsatkichini avtoinkrementlash yuzaga keladi, yoxud bajariladigan komandalarda operandlarning mavjud berilgan maydoni unga kiritiladi. Ikkala holatda komandalar ko'rsatkichida navbatdagi bajarilishi kerak bo'lgan komandalarni o'z ichiga olgan xotira katakchalari adresi bo'ladi va tavsiflangan jarayon takrorlanadi.

Masalan, PLUS opX, opY, opZ ($opZ=opX+opY$) gipotetik komandalarini bajarish ma'lumotlar ko'rsatkichiga opX adresini yuklash, "ma'lumotlar xotirasini o'qish" signalini berish, ma'lumotlar xotirasining tayyorligini kutish, ma'lumotlar xotirasining ma'lumotlar shinasidan AMQning kirish operandining 1-son registriga yuklash, opY uchun shunga o'xshash harakatlarni bajarish (2-son registrga yozish), «qo'shish» kodini AMQga berish, AMQ tayyorligini kutish, opZ adresni ma'lumotlar ko'rsatkichiga yuklash, ma'lumotlarni AMQning kirish operandining 1-son registridan ma'lumotlar xotirasining ma'lumotlar shinalariga berish, «ma'lumotlar xotirasini yozish» signalini berish, ma'lumotlar xotirasi tayyorligini kutish kabi harakatlarini yuzaga keltiradi.

1.3. Mikrokontroller buyruq tizimining xususiyatlari

AVR oilasiga mansub mikrokontrollerlarning komandalar tizimi

Mnemonika	Operandlar	Tavsif	Operatsiya	Bayroqlar	Taktilar soni*
1	2	3	4	5	6
Arifmetik va logik komandalar					
ADD	Rd, Rr	O'tkazish bayrog'ini hisobga olmagan xolda qo'shish	$Rd \leftarrow Rd + Rr$	Z, C, N, V, H, S	1
ADC	Rd, Rr	O'tkazish bayrog'ini hisobga olgan xolda qo'shish	$Rd \leftarrow Rd + Rr + C$	Z, C, N, V, H, S	1
ADIW	Rdl, K6	So'z va konstantani qo'shish	$Rdh:Rdl \leftarrow Rdh:Rdl + K6$	Z, C, N, V, S	2
SUB	Rd, Rr	O'tkazish bayrog'ini hisobga olmagan xolda ayirish	$Rd \leftarrow Rd - Rr$	Z, C, N, V, H, S	1
SUBI	Rd, K8	Konstantani ayirish	$Rd \leftarrow Rd - K8$	Z, C, N, V, H, S	1
SBC	Rd, Rr	O'tkazish bayrog'ini hisobga olgan xolda ayirish	$Rd \leftarrow Rd - Rr - C$	Z, C, N, V, H, S	1
SBCI	Rd, K8	O'tkazish bayrog'ini hisobga olgan xolda konstantani ayirish	$Rd \leftarrow Rd - K8 - C$	Z, C, N, V, H, S	1
SBIW	Rdl, K6	Konstantani so'zdan ayirish	$Rdh:Rdl \leftarrow Rdh:Rdl - K6$	Z, C, N, V, S	2

AND	Rd, Rr	Logik VA ni bajarish	$Rd \leftarrow Rd \cdot Rr$	Z, N, V, S	1
ANDI	Rd, K8	Logik VA ni konstanta bilan bajarish	$Rd \leftarrow Rd \cdot K8$	Z, N, V, S	1
OR	Rd, Rr	Logik Yoki ni bajarish	$Rd \leftarrow Rd \vee Rr$	Z, N, V, S	1
ORI	Rd, K8	Logik Yoki ni konstanta bilan bajarish	$Rd \leftarrow Rd \vee K8$	Z, N, V, S	1
EOR	Rd, Rr	Logik Yoki inkorini bajarish	$Rd \leftarrow Rd \oplus Rr$	Z, N, V, S	1
COM	Rd	Razryad inversiyasi	$Rd \leftarrow \$FF - Rd$	Z, C, N, V, S	1
NEG	Rd	Ishorani o'zgartirish	$Rd \leftarrow \$00 - Rd$	Z, C, N, V, H, S	1
SBR	Rd, K8	Razryadni (razryadlarni) UMR ga joylashtirish	$Rd \leftarrow Rd \vee K8$	Z, C, N, V, S	1
CBR	Rd, K8	Razryadlarni UMR ga tashlash	$Rd \leftarrow Rd \cdot (\$FF - K8)$	Z, C, N, V, S	1
INC	Rd	UMR tarkibini inkrementlash	$Rd \leftarrow Rd + 1$	Z, N, V, S	1
DEC	Rd	UMR tarkibini dekrementlash	$Rd \leftarrow Rd - 1$	Z, N, V, S	1
TST	Rd	Nolga tengligini yoki manfiy qiymatga egaligini tekshirish	$Rd \leftarrow Rd \cdot Rd$	Z, C, N, V, S	1
CLR	Rd	UMR ning hamma razryadini tozalash	$Rd \leftarrow Rd \oplus Rd$	Z, C, N, V, S	1

SER	Rd	UMR ning hamma razryadini o'rnatish	$Rd \leftarrow \$FF$	Net	1
CP	Rd, Rr	Solishtirish	$Rd - Rr$	Z, C, N, V, H, S	1
CPC	Rd, Rr	O'tkazish bayrog'ini inobatga olib solishtirish	$Rd - Rr - C$	Z, C, N, V, H, S	1
CPI	Rd, K8	Konstanta bilan solishtirish	$Rd - K8$	Z, C, N, V, H, S	1
MUL	Rd, Rr	Ikkita UMR (ishorasiz) tarkibini ko'paytirish	$R1:R0 \leftarrow Rd \cdot Rr$	Z, C	2
MULS	Rd, Rr	Ikkita UMR (ishorali) tarkibini ko'paytirish	$R1:R0 \leftarrow Rd \cdot Rr$	Z, C	2
MULSU	Rd, Rr	Ikkita UMR (Rd – ishorali; Rr – ishorasiz) tarkibini ko'paytirish	$R1:R0 \leftarrow Rd \cdot Rr$	Z, C	2
FMUL	Rd, Rr	Ikkita UMR (ishorasiz) tarkibini chapga, 1 razryadga siljish bilan ko'paytirish	$R1:R0 \leftarrow Rd \cdot Rr \ll 1$	Z, C	2
FMULS	Rd, Rr	Ikkita UMR (ishorali) tarkibini chapga, 1 razryadga siljish bilan ko'paytirish	$R1:R0 \leftarrow Rd \cdot Rr \ll 1$	Z, C	2

FMULSU	Rd, Rr	Ikkita UMR (Rd – ishorali; Rr–ishorasiz) tarkibini chapga, 1 razryadga siljish bilan ko‘paytirish	$R1:R0 \leftarrow Rd \cdot Rr \ll 1$	Z, C	2
Tarmoqlanish komandalari					
RJMP	k	Nisbiy o‘tish	$PC \leftarrow PC + k + 1$	Yo‘q	2
IJMP	Net	Bilvosita o‘tish	$PC \leftarrow Z$	Yo‘q	2
EIJMP	Net	Kengaytirilgan nisbiy o‘tish	$PC(15:0) \leftarrow Z,$ $PC(21:16) \leftarrow EIND$	Yo‘q	2
JMP	k	Bilvosita o‘tish	$PC \leftarrow k$	Yo‘q	3
RCALL	k	Bilvosita qism dasturni yuklash	$STACK \leftarrow PC + 1,$ $PC \leftarrow PC + k + 1$	Yo‘q	3
ICALL	Net	Bilvosita qism dasturni yuklash	$STACK \leftarrow PC + 1,$ $PC \leftarrow Z$	Yo‘q	3
EICALL	Net	Kengaytirilgan bilvosita qism dasturni yuklash	$STACK \leftarrow PC + 1,$ $PC(15:0) \leftarrow Z,$ $PC(21:16) \leftarrow EIND$	Yo‘q	3
CALL	k	Qism dasturni chaqirish	$STACK \leftarrow PC + 1,$ $PC \leftarrow k$	Yo‘q	4/5
RET	Net	Qism dasturdan qaytish	$PC \leftarrow STACK$	Yo‘q	4
RETI	Net	Uzilishlarni tashkil qilish qism dasturidan qaytish	$PC \leftarrow STACK$	Yo‘q	4

CPSE	Rd, Rr	Solishtirish, agr teng bo'lsa o'tkazib yuborish	agar $Rd = Rr$, $PC \leftarrow PC + 2$ (yoki 3)	Yo'q	1/2/3
SBRC	Rr, b	O'tkazib yuborish, agar UFRB bit tushirib qoldirilgan	agar $Rr(b) = 0$, $PC \leftarrow PC + 2$ (yoki 3)	Yo'q	1/2/3
SBRS	Rr, b	O'tkazib yuborish, agar UFRB bit o'rnatilgan	agar $Rr(b)=1$, $PC \leftarrow PC + 2$ (yoki 3)	Yo'q	1/2/3
SBIC	I/O, b	O'tkazib yuborish, agar k/ch registrida bit tushirib qoldirilgan.	agar $I/O(b) = 0$, $PC \leftarrow PC + 2$ (yoki 3)	Yo'q	1/2/3
SBIS	I/O, b	O'tkazib yuborish, agar k/ch registrida bit o'rnatilgan.	agar $I/O(b) = 1$, $PC \leftarrow PC + 2$ (yoki 3)	Yo'q	1/2/3
BRBC	s, k	O'tish, agar registr SREG alomati o'rnatilmagan	agar $SREG(s) = 0$, $PC \leftarrow PC + k + 1$	Yo'q	1/2

BRBS	s, k	O'tish, agar registr SREG alomati o'rnatilgan	agar $SREG(s) = 1$, $PC \leftarrow PC + k + 1$	Yo'q	1/2
BREQ	k	O'tish, agar teng bo'lsa	agar $Z = 1$, $PC \leftarrow PC + k + 1$	Yo'q	1/2
BRNE	k	O'tish, agar teng bo'lmasa	agar $Z = 0$, $PC \leftarrow PC + k + 1$	Yo'q	1/2
BRCS	k	O'tish, agar uzatish bayrog'i o'rnatilgan.	agar $C = 1$, $PC \leftarrow PC + k + 1$	Yo'q	1/2
BRCC	k	O'tish, agar uzatish bayrog'i o'rnatilmagan.	agar $C = 0$, $PC \leftarrow PC + k + 1$	Yo'q	1/2

BRSH	k	O'tish, agar katta yoki teng.	agar $C = 0$, $PC \leftarrow PC + k + 1$	Yo'q	1/2
BRLO	k	O'tish, agar kichik	agar $C = 1$, $PC \leftarrow PC + k + 1$	Yo'q	1/2
BRMI	k	O'tish, agar minus	agar $N = 1$, $PC \leftarrow PC + k + 1$	Yo'q	1/2
BRPL	k	O'tish, agar plyus	agar $N = 0$, $PC \leftarrow PC + k + 1$	Yo'q	1/2
BRGE	k	O'tish, agar katta yoki teng (ishorali)	agar $S = 0$, $PC \leftarrow PC + k + 1$	Yo'q	1/2
BRLT	k	O'tish, agar kichik yoki teng (ishorali)	agar $S = 1$, $PC \leftarrow PC + k + 1$	Yo'q	1/2
BRHS	k	O'tish, agar yarim o'tkazish bayrog'i o'rnatilgan	agar $H = 1$, $PC \leftarrow PC + k + 1$	Yo'q	1/2
BRHC	k	O'tish, agar yarim o'tkazish bayrog'i o'rnatilmagan	agar $H = 0$, $PC \leftarrow PC + k + 1$	Yo'q	1/2
BRTS	k	O'tish, agar T bayrog'i o'rnatilgan	agar $T = 1$, $PC \leftarrow PC + k + 1$	Yo'q	1/2
BRTC	k	O'tish, agar yarim o'tkazish bayrog'i o'rnatilmagan	agar $T = 0$, $PC \leftarrow PC + k + 1$	Yo'q	1/2
BRVS	k	O'tish, agar razryadlar setkasini to'lib ketish bayrog'i o'rnatilgan	agar $V = 1$, $PC \leftarrow PC + k + 1$	Yo'q	1/2
BRVC	k	O'tish, agar razryadlar setkasini to'lib ketish bayrog'i o'rnatilmagan	agar $V = 0$, $PC \leftarrow PC + k + 1$	Yo'q	1/2
BRIE	k	O'tish, agar uzilishlarga ruxsat	agar $I = 1$, $PC \leftarrow PC + k + 1$	Yo'q	1/2
BRID	k	O'tish, agar uzilishlarga ruxsat taqiqlangan	agar $I = 0$, $PC \leftarrow PC + k + 1$	Yo'q	1/2

Ma'lumotlarni uzatish buyruqlari					
MOV	Rd, Rr	UFRBga nusxa ko'chirish	$Rd \leftarrow Rr$	Yo'q	1
MOVW	Rd, Rr	UFRBdan nusxa ko'chirish	$Rd+1:Rd \leftarrow Rr+1:Rr$	Yo'q	1
LDI	Rd, K8	UFRBga konstantani yuklash	$Rd \leftarrow K8$	Yo'q	1
LDS	Rd, k	UFRBga OXQ dan yuklash	$Rd \leftarrow (k)$	Yo'q	2
LD	Rd, X	Bilvosita OXQ dan UFRB ga yuklash	$Rd \leftarrow (X)$	Yo'q	2
LD	Rd, X+	OXQ dan bilvosita inkrementli yuklash	$Rd \leftarrow (X),$ $X \leftarrow X + 1$	Yo'q	2
LD	Rd, -X	OXQ dan bilvosita dekrementli yuklash	$X \leftarrow X - 1,$ $Rd \leftarrow (X)$	Yo'q	2
LD	Rd, Y	OXQ dan UFRB ga bilvosita yuklash	$Rd \leftarrow (Y)$	Yo'q	2
LD	Rd, Y+	OXQ dan bilvosta inkrementli yuklash	$Rd \leftarrow (Y),$ $Y \leftarrow Y + 1$	Yo'q	2
LD	Rd, -Y	OXQ dan bilvosita dekrementli yuklash	$Y \leftarrow Y - 1,$ $Rd \leftarrow (Y)$	Yo'q	2
LDD	Rd, Y+q	OXQ ofsetidan bilvosita yuklash	$Rd \leftarrow (Y + q)$	Yo'q	2
LD	Rd, Z	OXQ dan UFRB ga bilvosita yuklash	$Rd \leftarrow (Z)$	Yo'q	2
LD	Rd, Z+	OXQ dan bilvosita inkrementli yuklash	$Rd \leftarrow (Z),$ $Z \leftarrow Z + 1$	Yo'q	2
LD	Rd, -Z	OXQ dan bilvosita dekrementli yuklash	$Z \leftarrow Z - 1,$ $Rd \leftarrow (Z)$	Yo'q	2
LDD	Rd, Z+q	OXQ ofsetidan bilvosita yuklash	$Rd \leftarrow (Z + q)$	Yo'q	2
STS	k, Rr	OXQ ga to'g'ridan to'g'ri saqlash.	$(k) \leftarrow Rr$	Yo'q	2
ST	X, Rr	OXQ ga bilvosita saqlash	$(X) \leftarrow Rr$	Yo'q	2
ST	X+, Rr	OXQ ga bilvosita inkrementli saqlash	$(X) \leftarrow Rr,$ $X \leftarrow X + 1$	Yo'q	2

ST	-X, Rr	OXQ ga bilvosita dekrementli saqlash	$X \leftarrow X - 1,$ $(X) \leftarrow Rr$	Yo'q	2
ST	Y, Rr	Bilvosita OXQ da saqlash	$(Y) \leftarrow Rr$	Yo'q	2
ST	Y+, Rr	Bilvosita inkrementli OXQ da saqlash	$(Y) \leftarrow Rr,$ $Y \leftarrow Y + 1$	Yo'q	2
ST	-Y, Rr	Bilvosita dekrementli OXQ da saqlash	$Y \leftarrow Y - 1,$ $(Y) \leftarrow Rr$	Yo'q	2
STD	Y+q, Rr	OXQ ofsetiga bilvosita saqlash	$(Y + q) \leftarrow Rr$	Yo'q	2
ST	Z, Rr	OXQ da bilvosita saqlash	$(Z) \leftarrow Rr$	Yo'q	2
ST	Z+, Rr	OXQ ga inkrementli saqlash.	$(Z) \leftarrow Rr,$ $Z \leftarrow Z + 1$	Yo'q	2
ST	-Z, Rr	OXQ ga dekrementli saqlash.	$Z \leftarrow Z - 1,$ $(Z) \leftarrow Rr$	Yo'q	2
ST	Z+q,	OXQ ofsetiga bilvosi	$(Z + q) \leftarrow Rr$	Yo'q	2
LPM	Net	Dastur xotirasidan baytni yuklash	$R0 \leftarrow (Z)$	Yo'q	3
LPM	Rd, Z	Dastur xotirasidan baytni yuklash	$Rd \leftarrow (Z)$	Yo'q	3
LPM	Rd, Z+	Dastur xotirasidan baytni inkrementli yuklash	$Rd \leftarrow (Z),$ $Z \leftarrow Z + 1$	Yo'q	3
ELPM	Net	Dastur xotirasidan baytni ommaviy yuklash	$R0 \leftarrow (RAMPZ:Z)$	Yo'q	3
ELPM	Rd, Z	Dastur xotirasidan baytni ommaviy yuklash	$Rd \leftarrow (RAMPZ:Z)$	Yo'q	3
ELPM	Rd, Z+	Dastur xotirasidan inkrementli baytni ommaviy yuklash	$Rd (RAMPZ:Z),$ $Z \leftarrow Z + 1$	Yo'q	3
SPM	Net	Dastur xotirasiga yozish	$(Z) \leftarrow R1:R0$	Yo'q	-
IN	Rd, I/O	K\Ch registridan o'qish	$Rd \leftarrow I/O$	Yo'q	1

OUT	I/O, Rr	K\Ch registriga yozish	$I/O \leftarrow Rr$	Yo'q	1
PUSH	Rr	UFRB tarkibini stekga joylash	$STACK \leftarrow Rr$	Yo'q	2
POP	Rd	Stekdan UFRB ga yuklash	$Rd \leftarrow STACK$	Yo'q	2
Bitlar bilan ishlash buyruqlari					
LSL	Rd	Chapga mantiqiy surish	$Rd(n+1) \leftarrow Rd(n),$ $Rd(0) \leftarrow 0,$ $C \leftarrow Rd(7)$	Z, C, N, V, H, S	1
LSR	Rd	O'nga mantiqiy surish	$Rd(n) \leftarrow Rd(n+1)$ $Rd(7) \leftarrow 0, C \leftarrow 0$	Z, C, N, V, H, S	1
ROL	Rd	Ko'chirish bayrog'i asosida chapga siklik surish	$Rd(0) \leftarrow C,$ $Rd(n+1) \leftarrow Rd(n),$ $C \leftarrow Rd(7)$	Z, C, N, V, H, S	1
ROR	Rd	Ko'chirish bayrog'i asosida o'nga siklik surish	$Rd(7) \leftarrow C,$ $Rd(n) \leftarrow Rd(n+1),$ $C \leftarrow Rd(0)$	Z, C, N, V, S	1
ASR	Rd	O'nga arifmetik surish	$Rd(n) \leftarrow Rd(n+1)$ $n = 0..6$	Z, C, N, V, S	1
SWAP	Rd	Nibblarni (tetrada) o'zaro joyini almashtirish	$Rd(3..0) \leftarrow Rd(7..4),$ $Rd(7..4) \leftarrow Rd(3..0)$	Yo'q	1
BSET	s	Xolatlar registri SREG da bayroq o'rnatish	$SREG(s) \leftarrow 1$	SREG(s)	1
BCLR	s	Xolatlar registri SREG da bayroq o'rnatmaslik	$SREG(s) \leftarrow 0$	SREG(s)	1
SBI	I/O, b	K\Ch registrida razryad o'rnatish	$I/O(b) \leftarrow 1$	Yo'q	2
CBI	I/O, b	K\Ch registrida razryad o'rnatmaslik	$I/O(b) \leftarrow 0$	Yo'q	2
BST	Rr, b	T trigger alomatida UFRB razryadini saqlash	$T \leftarrow Rr(b)$	T	1

BLD	Rd, b	T trigger alomatidan UFRBrazryadini yuklash	$Rd(b) \leftarrow T$	Yo'q	1
SEC	Yo'q	Ko'chirish bayrog'ini o'rnatish	$C \leftarrow 1$	C	1
CLC	Yo'q	Ko'chirish bayrog'ini o'rnatmaslik	$C \leftarrow 0$	C	1
SEN	Yo'q	Manfiy son bayrog'ini o'rnatish	$N \leftarrow 1$	N	1
CLN	Yo'q	Manfiy son bayrog'ini o'rnatmaslik	$N \leftarrow 0$	N	1
SEZ	Yo'q	Nol bayrog'ini o'rnatish	$Z \leftarrow 1$	Z	1
CLZ	Yo'q	Nol bayrog'ini o'rnatmaslik	$Z \leftarrow 0$	Z	1
SEI	Yo'q	Uzilishlarga ruxsat bayrog'ini o'rnatish	$I \leftarrow 1$	I	1
CLI	Yo'q	Uzilishlarga ruxsat bayrog'ini o'rnatmaslik	$I \leftarrow 0$	I	1
SES	Yo'q	Ishorali sonlar bayrog'ini o'rnatish	$S \leftarrow 1$	S	1
CLS	Yo'q	Ishorali sonlar bayrog' o'rnatmaslik	$S \leftarrow 0$	S	1
SEV	Yo'q	To'lib ketish alomat registrini o'rnatish	$V \leftarrow 1$	V	1
CLV	Yo'q	To'lib ketish alomat registrini o'rnatmaslik	$V \leftarrow 0$	V	1
SET	Yo'q	T bayroq registrnini o'rnatish	$T \leftarrow 1$	T	1
CLT	Yo'q	T bayroq registrnini o'rnatmaslik	$T \leftarrow 0$	T	1
SEH	Yo'q	Ichki ko'chirish bayrog'ini o'rnatish	$H \leftarrow 1$	H	1
CLH	Yo'q	Ichki ko'chirish bayrog'ini o'rnatmaslik	$H \leftarrow 0$	H	1

NOP	Yo‘q	Bo‘sh operatsiya	Yo‘q	Yo‘q	1
BREAK	Yo‘q	Qayta tiklash	Mikrokontroller turi	Yo‘q	1
SLEEP	Yo‘q	Manba‘ ta‘minotni kamaytirish	Mikrokontroller turi	Yo‘q	1
WDR	Yo‘q	Qo‘riqchi taymerni o‘rnatmaslik	Mikrokontroller turi	Yo‘q	1

Shartli belgilanishlar:

Rd – rezultatlanadigan (destination) va boshlang‘ich UMR;

Rr – boshlang‘ich (source) RON;

I/O – kiritish-chiqarish registri;

b – UMR yoki kiritish-chiqarish registridagi razryadlar;

s – SREG xolat registridagi razryad (bayroq);

Rdl – R24, R26, R28, R30 registrlari;

X, Y, Z – qisman adreslash uchun ko‘rsatgich-registrlari (X = R27:R26, Y = R29:R28, Z = R31:R30);

RAMPX, RAMPY, RAMPZ – registrlari, X, Y i Z ko‘rsatkich-registrlari bilan bog‘liq va butun ma‘lumotlar xotirasi hajmi bo‘yicha qisman adreslash (ma‘lumotlar xotirasida 64 Kbaytdan oshiq hajmga ega) va dastur xotirasida 64 Kbaytdan oshiq hajmga ega mikrokontrollerlar dastur xotirasida konstantaga kirish yo‘liga ega;

RS – dastur schotchigi (Program Counter);

STACK – qaytish adreslarini va registr tarkibini saqlash uchun stek;

SP – stek ko‘rsatgichi (Stack Pointer);

EIND – dastur hisoblagichiga bog‘liq va dastur xotirasida 64 Kbaytdan oshiq hajmga ega mikrokontrollerlar bo‘yicha butun dastur xotirasi hajmini qisman adreslash;

K6 – konstanta (6 razryadli);

K8 – konstanta (8 razryadli);

k – dastur hisoblagichi uchun konstantani adreslash (hajmi komandaga bog‘liq);

q – qisman adreslashda siljish (6 razryadli).

SREG xolat registrining razryadlari:

C – ko‘chirish bayrog‘i;

Z – nol qiymatining bayrog‘i;

N – manfiy qiymatning bayrog‘i;

V – to‘lish bayrog‘i;

S = N \oplus V – to‘lish belgili nazorat bayrog‘i;

H – yarim ko‘chish bayrog‘i (3- va 4- bayt razryadlari o‘rtasida ko‘chish);

T – o‘tkazish komandalari uchun bayroq;

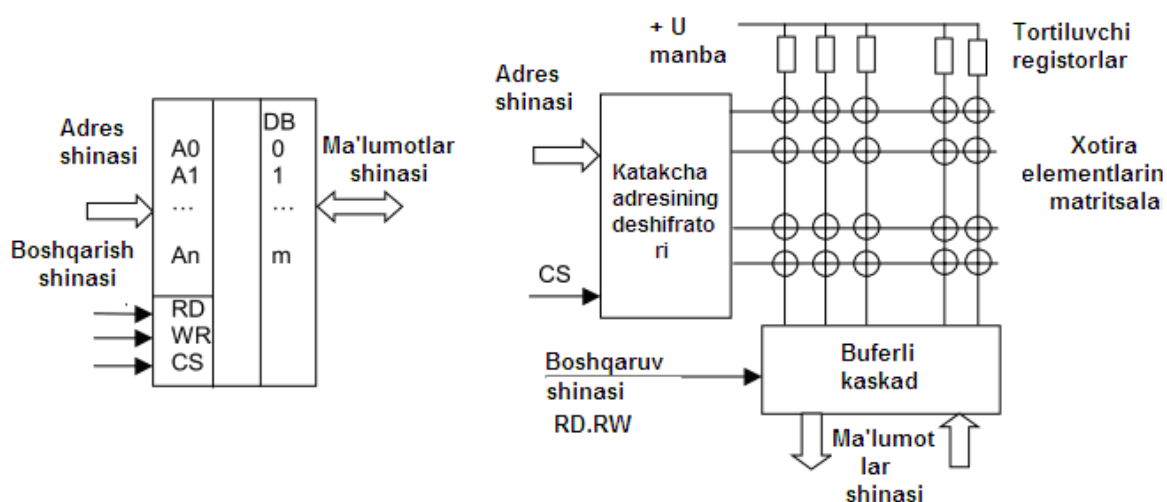
I – uzilishlarga global ruxsat berish (taqiqlash).

1.4. Mikrokontrollerlarda xotirani tashkil etilishi va u bilan ishlash

Mikrokontroller kristallida dasturlar xotirasi va ma’lumotlar xotirasi kabi xotiraning ikkita bloki integratsiyalangan. MK avtonom rejimda ishlashining mo‘ljallanishiga qarab, dastur xotirasi ta’minot kuchlanishi bo‘lmaganda ichidagilarni saqlashi kerak (ya’ni energiyaga bog‘liq bo‘lib hisoblanadi), MKning ichki arxitekturasini soddalashtirish va taktli generatorning chastotalarning keng diapazonida ishlash imkoniyati uchun ma’lumotlar xotirasi statik arxitekturasiga ega bo‘lishi (ya’ni regeneratsiyani talab etmasligi) kerak.

Xotira modulining umumlashtirilgan tuzilmasi 1.4- rasmda keltirilgan. Xotira moduli m-razryadli qatorlarning N ko‘rinishida tashkil etilgan xotira elementlarining matritsalaridan, katakchalar adresining deshifradoridan va buferli kaskaddan iborat bo‘ladi.

- adres shinasi
- ma’lumotlar shinasi
- boshqarish shinasi
- registrlar



1.4- rasm. Xotira modulining umumlashgan tuzilmasi

- katakcha adresining deshifratori
- xotira elementlarining matritsalari

- boshqaruv shinasasi
- buferli kaskad

Xotiraning bunday modulining adres shinalarining razryadliligi $n - \log_2 N$ dan iborat, ma'lumotlar shinasining razryadliligi m dan iborat. Adres kodi ko'rinishdagi katakcha tanlanishini kerak bo'lgan raqam to'g'risidagi axborot o'zining bitta chiqishidagi yuqori mantiqiy darajani generatsiyalashning xotira elementlarining matritsa qatorlaridan birini aktivlashtiradigan deshifratoriga kelib tushadi.

Bunda (kelib tushadigan boshqaruv signallariga bog'liq holda) tanlangan qatorlarning barcha xotira elementlarining mantiqiy darajasi ma'lumotlar shinasiga buferli kuchaytirgich kaskadi orqali kelib tushadi (katakcha holatini o'qish vaziyati), yoxud ma'lumotlar shinasiga tanlangan qatorning xotira elementlariga buferli kuchaytirgich kaskadlari orqali uzatiladi (katakcha holatini yozish vaziyati).

Boshqa qatorlarning xotira elementlarining mantiqiy holati o'zgarmaydi va chiqish mantiqiy darajaga ta'sir etmaydi. Dastur energiyaga bog'liq xotira bo'lib doimiy xotira qurilmalari (DXQ) hisoblanadi. DXQ har bir xotira elementi axborotni DXQga kiritishda (dasturlashtirishda) uzatilishi kerak bo'lgan mantiqiy holatida bo'ladi. DXQda boshqariluvchi dasturni yozishning yo'l quyiladigan sikllar miqdoriga bog'liq bir karrali va ko'p karrali dasturlanadigan moduliga ajratiladi.

1.4.1. Mikrokontroller dasturiy xotirasi

Dastur xotirasining asosiy xususiyati bu uning iste'mol energiyasiga bog'liq bo'lmaganligidir, ya'ni manba bo'lmagan holda ham dasturlarni saqlash imkoniyatidir. MK foydalanuvchilari nuqtai nazaridan dasturlarning energiyaga bog'liq bo'lmagan xotirasini quyidagi turlarga bo'lish mumkun:

-Maskali DXQ turi – mask-ROM. Bu turdagi DXQ yacheykalariga qiymatlarni uning ishlab chiqarilish jarayonida maskalar yordamida kiritiladi va keyinchalik qiymatlarni o'zgartirib yoki qayta dasturlab bo'lmaydi. Shuning uchun bu turdagi xotirali mikrokontrollerlarda dasturlarni uzoq vaqt sinashdan keyingina foydalanish mumkin. Bu xotiraning asosiy kamchiligi yangi shablonlarning to'plamini yaratilishidagi va ularni ishlab chiqarishga tatbiqidagi sarf harajatlarning yuqoriligi. Odatda bu jarayon 2 - 3 oyni tashkil etadi va iqtisodiy nuqtaiy nazardan minglab nusxada ishlab

chiqarilgandagina o'zini oqlashi mumkin. Maskali DXQ axborotlarni saqlashda yuqori ishonchlilikni ta'minlashining sababi, zavod sharoitida dasturlanib so'ng maxsulotning turli nazoratlardan o'tkazilishidir.

Foydalanuvchi tomonidan dasturlanuvchi va ultra binafsha nurlar bilan o'chiruvchi DXQ – EPROM (Erasable Programmable ROM). Bu turdagi DXQ elektr signallari bilan dasturlanadi va ultra binafsha nurlash orqali o'chiriladi. EPROM xotira yacheykasi “suzuvchi” zatvorli metal oksid yarimo'tkazgichli (MOYA) – tranzistorlarda hosil qilingan va ularga zaryad boshqaruvchi zatvordan kerakli elektr signali berilganida o'tadi. Yacheyka qiymatini o'chirish uchun uni ultra binafsha nurlar bilan nurlantiriladi. Bu jarayon bir necha sekunddan bir necha minutgachan davom etishi mumkin. EPROM xotirali MK ko'p marotaba qayta dasturlanishi mumkin va u turdagi xotiralar keramik g'ilofda kvarsli darcha bilan ishlab chiqariladi. Darcha ultra binafsha nurlarni xotira yacheykasiga tushishi uchun mo'ljallanga. Bunday g'ilof ancha qimmat turgani uchun MK narxi ham qimmatlashadi albatta. EPROM xotirali MK narxini arzonlatish uchun xotirani darchasiz ishlab chiqariladi (EPROM bir marotaba dasturlanuvchi versiyasi). - Foydalanuvchi tomonidan bir marotaba dasturlanuvchi DXQ - OTPROM (One – Time Programmable ROM). Bu turdagi xotira, MK narxini arzonlatish uchun 115 xotirani darchasiz ishlab chiqarilgan EPROM versiyasidan iborat. Bu kabi g'iloflarda ishlab chiqarilishi natijasida narxini kamaytirish shunchalik etiborliki, oxirgi vaqtda EPROM versiyalari ko'pincha maskali DXQ xotira turi o'rniga ishlatilmoqda. -Elektr o'chirishli foydalanuvchi tomonidan dasturlanuvchi DXQ – EEPROM (Electrically Erasable Programmable ROM). Bu turdagi DXQ ni EPROM ning yangi avlodi deb hisoblash mumkin, ularda xotira yacheykasi shuningdek elektr signali orqali o'chirish amalga oshiriladi. EEPROM qo'llanilishi MK platasini yechib olmasdan o'chirish va dasturlash imkonini beradi. Shu usulda sozlashni amalga oshirish va dasturiy ta'minotni yangi variantlarini yozish ham mumkin. Bu esa mikrokontrollerli tizimlarni yaratishning boshlang'ich bosqichlarida yoki ularni o'rganish jarayonida, qachonki tizimning ishlamaslik sababini topishga va dasturlar xotirasini o'chirish-dasturlash sikllarini bajarishga ko'p vaqt ketqazilganda juda katta yutuq beradi. EEPROM narxi bo'yicha OTPROM va EPROM larning o'rtasidagi o'rinni egallaydi.

EEPROM xotirasining dasturlash texnologiyasining yacheykalarni baytlab o'chirish va dasturlash imkoniyati mavjud. EEPROM yaqqol

afzalliklariga qaramay MK faqat kam modellarida bunday xotira dasturlarni saqlashga ishlatiladi. Buning sababi, birinchidan EEPROM chegaralangan xotira sig'imga ega. Ikkinchidan, deyarli EEPROM bilan bir vaqtda Flash-DXQ ishlab chiqarildi, ular texnik ko'rsatgichlari jihatidan bir xil bo'lishi bilan bir qatorda, lekin Flash-DXQ narxi ancha arzon. Flash turidagi elektr o'chirishli DXQ - Flash – ROM. Vazifasi jihatidan Flashxotira EEPROM dan kam farq qiladi. Asosiy farqi yozilgan axborotni o'chirish usulida- dir. EEPROM xotirasida har bir yacheyka alohida o'chiriladi, Flash-xotirada esa faqat bloklab o'chirish mumkin. Agarda Flash-xotiraning bitta yacheykasining qiymatini o'zgartirish kerak bo'lib qolsa, butun blokni boshqatan dasturlash kerak bo'ladi. EEPROM xotirasi nisbatan dekoderlovchi sxemalarining soddalashishi, Flash-xotirali MK lar nafaqat bir marotaba dasturlanuvchi DXQ li MK bilan raqobatbardoshlik qilmoqda, maskali DXQ bilan ham raqobat qilmoqda.

1.4.2. Mikrokontroller ma'lumot xotirasi

Xotira ma'lumotlari MK, qoida tariqasida, statik RAM asosida amalga oshiriladi. "Statik" atamasi MK takt chastotasi o'zboshimchalik bilan kichik qiymatlarga tushirilganda (quvvat sarfini kamaytirish uchun) RAM yacheykalarining tarkibi saqlanib qolishini anglatadi.

Ko'pgina MKlarda "ma'lumot saqlash" - USTANDBY kabi parametr mavjud. Ta'minot UDDMIN minimal qabul qilingan darajasidan pastga tushsa, lekin USTANDBY darajasidan yuqori bo'lsa, MK dasturi bajarilmaydi, ammo operativ xotiradagi ma'lumotlar saqlanadi. Ta'minot yuklamasi tiklanganda MKni qayta tiklash va ma'lumotlarni yo'qotmasdan dasturni davom ettirish mumkin bo'ladi. Saqlash kuchlanish darajasi odatda 1 V ni tashkil qiladi, bu kerak bo'lganda MK-ni avtonom manbadan (batareyadan) quvvat olish va ushbu rejimda RAM ma'lumotlarini saqlashga imkon beradi.

MK ma'lumotlarining xotira hajmi odatda kichik, o'nlab va yuzlab baytlarni tashkil qiladi. Ushbu holat MK uchun dasturlarni ishlab chiqishda hisobga olinishi kerak. Shunday qilib, MK dasturlashda, agar iloji bo'lsa, konstantalar o'zgaruvchilar sifatida saqlanmaydi, lekin ROM dasturiga kiritiladi. MK-ning apparat imkoniyatlaridan maksimal darajada foydalaniladi, xususan, taymerlar. Ilovalar katta hajmdagi ma'lumotlardan foydalanmasdan ishlashga e'tibor qaratishlari kerak.

1.5. Mikrokontroller registrlari

MKlar uning manbalarini boshqarish uchun foydalaniladigan registrlar to'plamiga ega. Ushbu registrlar odatda protsessor registrlari (batareyalar, holat registrlari, indeks registrlari), boshqarish registrlari (uzilishlarni boshqarish registrlari, taymer registrlari), ma'lumotlarning kirish /chiqishini ta'minlovchi registrlar (port registrlari, parallel, ketma-ket yoki analog kirish / kirish registrlari). Ushbu registrlarga kirish turli yo'llar bilan amalga oshirilishi mumkin.

RISC protsessoriga ega bo'lgan MKda, barcha registrlar (ko'pincha batareya) aniq belgilangan manzillarda joylashgan. Bu protsessorning ishlashida ko'proq moslashuvchanlikni ta'minlaydi. Muhim masalalardan biri bu ro'yxatga olish kitoblarini MKning manzil maydoniga joylashtirish. Ba'zi MKlarda barcha registrlar va ma'lumotlar xotirasi bitta manzil oralig'ida joylashgan. Bu ma'lumotlar xotirasi registrlar bilan moslashtirilganligini anglatadi. Ushbu yondashuv "MK resurslarini xotirada xaritalash" deb nomlanadi.

Boshqa MKlarda I/O qurilmalarining manzil maydoni umumiy xotira maydonidan ajratilgan. I/O alohida maydoni Garvard arxitekturasi bilan ishlaydigan protsessorlarga ma'lum afzalliklarni beradi, bu esa kirish /chiqish registriga kirishda buyruqlarni o'qish imkoniyatini beradi.

2. MIKROKONTROLLERLARNI TASHQI MUHIT BILAN ALOQASINI TASHKILLASHTIRISH

2.1. Mikrokontrollerlarda axborot kiritish-chiqarish portlarini tashkillashtirish va strukturasi

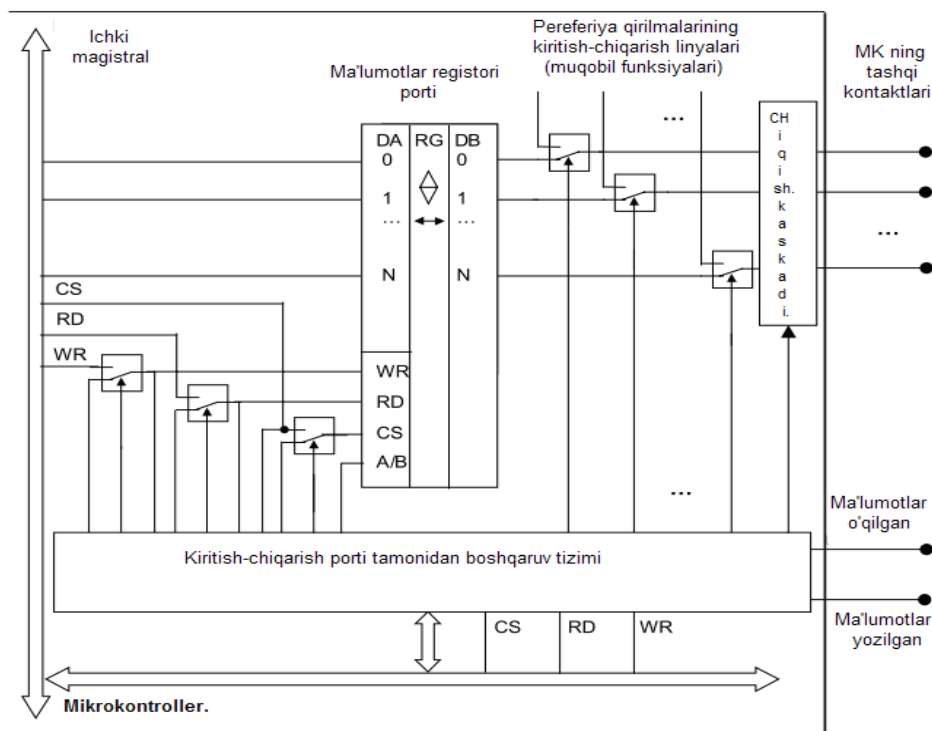
Kiritish-chiqarishning parallel portlari mikrokontroller almashinuvi va MK mikrosxemaning kiritish-chiqarish liniyalari orqali uzatiladigan mantiqiy signallar ko‘rinishda keltirilgan ma’lumotlarning tashqi ob’ekti uchun mo‘ljallangan. Umumiy holatda har bir port bilan ma’lumotlar registri (MKdan ob’ekti chiqariladigan axborotni saqlash uchun yoki MKga ob’ektga kiritiladigan axborotni saqlash uchun), boshqaruv tizimi (portning ishlash rejimlarini berish uchun) va signallarni kuchaytirish va qo‘shish vazifasini hal etuvchi chiqish kaskadiga bog‘liq bo‘ladi. Port tuzilmasi 2.1.- rasmda keltirilgan.

- ichki magistral
- periferiya qurilmalarining kiritish-chiqarish liniyalari
- MKning tashqi kontaktlari
- portning ma’lumotlar registri
- chiqish kaskadi
- ma’lumotlar o‘qilgan
- ma’lumotlar yozilgan
- kiritish-chiqarish porti tomonidan boshqaruv tizimi
- kiritish-chiqarish portlari MK barcha modellarida amalga oshiriladi.

Ma’lumotlar registri o‘z ichiga MKning ichki shinasini bilan axborot registriga kiritish yoki registr holatining ichki shinasiga chiqarish imkonini beradigan yozish va o‘qishni boshqarish liniyalari bilan ikki yo‘nalishli kiritish-chiqarishning N-razryadli registrini oladi. MK yadrosining protsessori va port registri o‘rtasidagi axborot almashinuvi momenti ichki shina orqali yadro shinasining kontrollerini belgilaydi. Almashinuv komandalarining tipiga muvofiq (protsessordan portga yoki portdan protsessorga uzatish) WR yoki RD signallari shakllantiriladi, keyin almashinuv CS boshqaruv signali bilan stroblanadi.

Umumiy holatda, almashinuvning juda murakkab vazifalarini hal etish imkoniyati uchun kiritish-chiqarish portining tuzilmasiga signallar holatini aniqlash va o‘zgartirishning kombinatsiyalanadigan sxemasidan iborat bo‘lgan boshqaruv tizimi, shuningdek ushbu holatni saqlaydigan

registrni yozish va o‘qish uchun dasturiy qulay bo‘lgan registr kiritiladi. Bunday vazifa ikkita: kiritish-chiqarish liniyalarining multiplekslash vazifasi va almashinuvning kengaytirilgan protokollarni qo‘llab-quvvatlash vazifasidir. Ularni batafsil ko‘rib chiqamiz.



2.1.- rasm. Kiritish chiqarishning parallel porti tuzilmasi

Ma'lumotlar registridan tashqari, bir xil va xuddi shunday MK kontaktlari ayrim periferiya qurilmasi (masalan, taymer) ob'ekt bilan (kiritish-chiqarishning muqobil funksiyalari deb ataladigan) axborot almashinuvi uchun talab etilishi mumkin. Bunda qaysi qurilma (porti yoki boshqa periferiya moduli) bilan MK mikrosxemalarning muayyan har bir kontakti mahkamlanganligini aniqlash zarur bo'ladi. Bunday roslash signallar kommutatorining holatini belgilaydigan maxsus kodning boshqaruv tizimining registriga yoziladi (1-8-rasm).

Kommutator kontaktining holatiga muvofiq MK mikrosxemaning har bir tashqi kontakti port registrining kiritish-chiqarish liniyalariga, yoxud boshqa periferiya qurilmasining kiritish-chiqarish liniyalariga fizik jihatdan ulangan bo'ladi. Ma'lumotlarni MKga ob'ektdan uzatishda ob'ekt tomonidan shakllanadigan «ma'lumotlar yozilgan» signali ma'lumotlar registrining CS kirishida boshqaruv tizimi bilan kommutatsiyalanadi, boshqaruv tizimi bilan WR signali shakllanadi. Bundan tashqari, «Ma'lumotlar yozilgan» signali boshqaruv tizimi registrida xotirada saqlanadi. Portning ma'lumotlar registrini o'qishdan

oldin amaliy dastur ob'ektdan yangi ma'lumotlar mavjudligini boshqaruv tizimi registridan so'raydi va ushbu dalil aniqlangandan keyin shina kontrolleri RD va CS signallarini shakllantiradigan ma'lumotlar registrini o'qishni amalga oshiradi.

Boshqaruv tizimi RD signalini «Ma'lumotlar o'qilgan» liniyasiga ma'lumotlarning quyidagi porsiyasini qabul qilishda MK tayyorligi to'g'risida ob'ektga xabar bergan holda kommutatsiyalaydi.

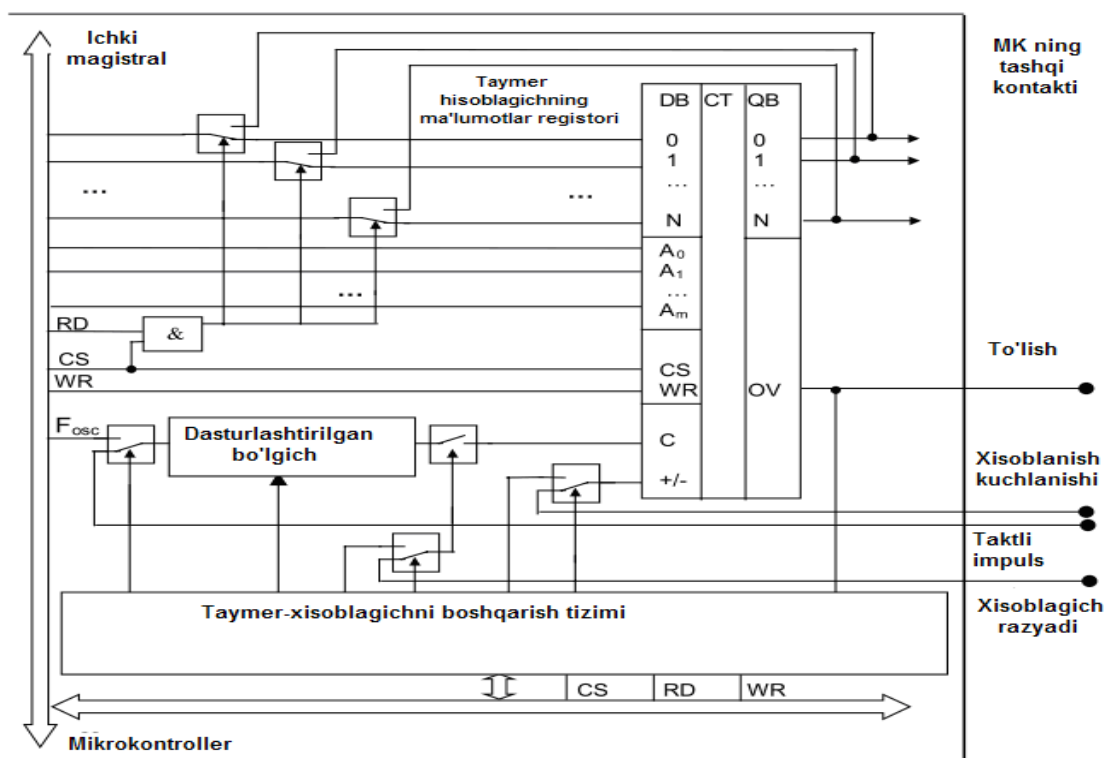
2.2. Taymer-hisoblagichlar

Taymer-hisoblagichlar vaqtli intervallarni shakllantirish va hodisalarni hisoblash uchun mo'ljallangan, bu har qanday vaqt funksiyalari asosida ularni amalga oshirish (tegishli dasturiy ta'minotdan foydalanilganda), shu jumladan real vaqtda boshqarish (ya'ni ob'ektning vaqt bo'yicha masshtabida) imkonini beradi. Taymer-hisoblagichning umumlashtirilgan tuzilmasi 2.2 -rasmda keltirilgan.

- ichki magistral
- taymer-hisoblagichning ma'lumotlar registri
- MKning tashqi kontakti
- dasturlashtirilgan bo'lgich
- hisoblagich kuchlanishi
- taktli impuls
- hisoblagich razryadi
- taymer-hisoblagichni boshqarish tizimi

Taymer-hisoblagich axborotni parallel yuklash va o'qish imkoniyatiga ega sinxron reversiv hisoblagichga asoslanadi. Qoidaga ko'ra, MK hisoblagichining razryadliligi protsessor razryadliligidan oshganligi sababli MK yadrosi va hisoblagichning ma'lumotlar registri qismi o'rtasida axborotni adresli almashinuvidan foydalaniladi.

Ushbu tuziladagi RD va CS signallarni bir vaqtda aktivatsiyalashda MKning ichki shinasida taymer-hisoblagichning chiqish shinasi kommutatsiyalanadi, aks holda, MKning ichki shinasi parallel yuklamaning kirishiga kommutatsiyalanadi. Taymer-hisoblagichda ma'lumotlarni yuklash WR va CS signallarni bir vaqtda aktivatsiyalashda sodir bo'ladi. Boshqaruv tizimi hisob (boshqaruv tizimi registrining dasturiy qulay biti yoki tashqa signal) yo'nalishlarining topshiriq manbaini, shuningdek taktirlash manbai (yadroning tayanch chastotasi yoki tashqi impulslari) va taktli impulslarni bo'lish koeffitsientini aniqlash imkonini beradi.



2.2- rasm. Taymer-hisoblagich tuzilmasi

Hisobning to'liq diapazonini 0 dan 2 gacha aniqlash imkonini beradi. Nominal (maksimal bo'lib hisoblangan) intervallardan farqlanadigan vaqtli intervallarni berish uchun hisoblagichning ($N=T/T$ kvant kattalik bilan kamayishdagi hisobda yoki $N_{max} - N$ ko'payishdagi hisobda) dastlabki dasturiy yuklamasidan foydalaniladi.

0 kodiga erishilganda (kamayishdagi hisobda) yoki N_{max} (ko'payishdagi hisobda) dasturiy vositalar uchun bo'lgani kabi (taymer-hisoblagichni boshqarish tizimi holatining registrini o'qishda), tashqi apparatura uchun, xususan, ob'ekt uchun qulay bo'lgan OV to'lish signali generatsiyalanadi.

Shuni ta'kidlash kerakki, vaqtli intervallarga rioya qilish vazifalarining yechimi dasturiy vositalar ayrim yo'riqnomalar bajariladigan t vaqtini, siklda uning N -karrpali bajarishni tashkil etish oson emasligini bilgan holda, shu bilan birga $T=Nt$ davomiylik bilan kechikishni ta'minlagan holda qo'llanilishi mumkin, biroq, bunday holatda boshqa dasturiy harakatlarni (masalan, axborotni yig'ish, operator bilan ma'lumotlar almashinuvi va h.) bajarish mumkin emas. Bundan tashqari, T kvant qoniqarsiz darajada katta bo'lishi mumkin. Taymer-hisoblagichlar MKning barcha modellarida amalga oshiriladi.

2.3. Jarayonlarga ishlov berish bloklari

Taymer-hisoblagichlar qo'llanilishi bilan erishiladigan vaqt funksiyalarini amalga oshirish imkoniyatlarini kengaytirish uchun MKning ayrim modellarining periferiya qurilmalar tarkibiga taymer-hisoblagichlar bilan bog'liq bo'lgan vaqtli hodisalarga apparatli ishlov berish bloki kiritilgan.

Turli firmalarning MKdagi bunday bloklarning nomi farqlanishi mumkin:

EPA (Event Processors Array – hodisalar protsessorlar massivi), HSIO (High Speed Input/Output unit – tezkor kiritish-chiqarish bloki), RSA (Programmable Counters Array – dasturlanadigan hisoblagichlar massivi) va boshqalar.

Bunday bloklarning asosiy vazifalar quyidagilar hisoblanadi:

- berilgan ko'rastkichli hodisalar (hodisalarni tutib qolish) sodir bo'lgan vaqtni aniqlash. Tutib qolishni dasturiy amalga oshirishning prinsipial qiyinchiligi taymer ma'lumotlarining ko'rsatkichini aniq o'qish imkoniyati mumkin emasligi bo'lib hisoblanadi (chunki taymer razryadliligi ichki shina razryadliligidan oshadi va registr fragmentlarini ketma-ket o'qish protsedurasi o'zgarishlar bilan bajarilishi mumkin);

- vaqtning berilgan momentida berilgan ko'rinishdagi hodisalarni generatsiyalash (vaqtga nisbatan hodisani bog'lash). Bog'lashni dasturiy amalga oshirishning prinsipial qiyinchiligi vaqtning berilgan momentini aniqlash momenti (masalan, taymer ma'lumotlarining registrini to'ldirish bo'yicha) va berilgan mantiqiy hodisalarni generatsiyalash komandalarini bajarish momenti o'rtasidagi vaqtli kechikishning mavjudligi bo'lib hisoblanadi;

- umumiy vaqtli bazadagi keltirilgan ko'rinishlarning ko'p kanalliligini ta'minlab turish. Ko'p kanallilikni dasturiy amalga oshirishning prinsipial qiyinchiligi bir nechta dasturiy modullar tomonidan taymer ko'rsatkichlarini kuzatishning murakkabliligi va samarasizligi bo'lib hisoblanadi.

Ahamiyatli belgilangan qiyinchilik bo'lib, dasturiy ishlov berish o'lchanadigan va generatsiyalanadigan intervallarda ahamiyatli xatoliklar kiritiladigan hodisalar o'rtasidagi vaqt intervallarida hisoblanadi. Mantiqiy signal qiymatining "0" dan "1" ga o'zgarishida va aksida yozish strobi ishlab chiqariladi va ushlab registriga yoziladi. Bayon etilgan amal ushlab voqeasi deb ataladi.

2.4. Uzilishli hodisalarga xizmat ko'rsatish bloklari

Har qanday hisoblash tizimlarining ishlashi (shu jumladan MK asosida), qoidaga ko'ra, ayrim hodisalarga xizmat ko'rsatish nazarda tutiladi. MKda uzilishlarga ishlov berish ham mikroprotessorli tizimlarda uzilishlarga ishlov berishning umumiy tamoyillariga asoslangan holda olib boriladi. Uzilish moduli uzilishga so'rovni olgach ma'lum uzuvchi dasturni bajarishga o'tishni tashkillashtiradi. Uzilishga so'rov tashqi manbalardan ham va shuningdek MK turli ichki modullarida joylashgan ma'nbalardan ham berilishi mumkin. Ularning yuzaga kelish sababalari va joyi turlicha bo'lishi mumkin, biroq ushbu hodisalar bitta umumiy xususiyatga ega bo'ladi, ularning yuzaga kelish vaqti dastur uchun oldindan ma'lum emas. Shu sababli hisoblash tizimida hodisalarni aniqlash dalili bo'yicha ularga xizmat ko'rsatishning ayrim hodisalarining vositalarini nazarda tutish zarur bo'lgan ushbu hodisalarni aniqlash vositalari nazarda tutiladi. (Chunki hodisalarga xizmat ko'rsatishda bajarilayotgan dasturni olib qo'yish («uzish») va xizmat ko'rsatiladigan dasturni bajarish zarur, bunday hodisalar uzilishli hodisalar deb ataladi). Uzilishli hodisalarni aniqlashning oddiy usuli bo'lib dasturiy usul bilan yuzaga kelishini vaqti-vaqti bilan tekshirish hisoblanadi, bunda ikkita ahamiyatli kamchiliklar:

- hisoblash tizimining unumdorligini kamaytirish (dasturning ayrim qismi yuzaga keladigan hodisalar belgisini ortiqcha tanlab olish ostida keltirilgan);

- hodisalar yuzaga kelish vaqti va uni aniqlash vaqti o'rtasidagi kechikish mavjudligi (qator holatlarda, masalan, real vaqt tizimlarida, prinsipial mumkin emas).

Ko'rsatilgan kamchiliklarni oldini olish berilgan uzilishli hodisalarni apparatli aniqlash holatda mumkin, bunda uzilishli hodisalarga xizmat ko'rsatish tizimi yordamida amalga oshiriladi. Uzilishli hodisalarni aniqlashdan tashqari, uzilishli tizimga ulardan ustuvorligini aniqlash uchun hodisalar arbitraji yuklangan. Uzilishli hodisalar ham dasturiy va apparatli tabiatga ega bo'ladi:

- uzilishli hodisalar yuzaga kelishning dasturiy sabablariga mavjud bo'lmagan komandalarni bajarishga urinish (ya'ni KOP MKning

komandalar tizimida mavjud bo'lmagan) yoki yo'l qo'yilmaydigan ma'lumotlar bilan kamandalarni bajarish (masalan, nolga bo'lish holatida) kiradi;

- uzilishli hodisalar yuzaga kelishning apparatli sabablariga periferiya modullarining tayyorligi (masalan, ARO'ga oxirgi o'zgartirish, taymerning to'lishi, chaqiruvni ketma-ket port bo'yicha qabul qilish va h.) va MKga boshqaruv ob'ektidan kelib tushadigan signallarni (masalan, bajariluvchi mexanizmning ishlab ketish datchigidan ikkilik signalni) aktivizatsiyalash.

Uzilishli hodisalar miqdori cheklangan (tashqi uzilishli hodisalar miqdori MKning kiritish-chiqarish liniyalarining sonidan oshmasligi kerak), shuning uchun ular apparatli vositalarni aniqlash bo'lishi mumkin. Aks holda, ularga xizmat ko'rsatish bo'yicha harakatlar MKning maqsadli funksiyalari bilan aniqlanadi va yetarlicha turli xil bo'lishi mumkin, qoiadaga ko'ra, dasturiy vositalar bilan xizmat ko'rsatiladi. Xotiradan to'g'ri foydalana olish rejimida apparatli harakatlarning oldindan berilgan sxemalar bo'yicha bajariladigan ma'lumotlarni jo'natish bilan bog'liq bo'lgan vaziyatlar (masalan, analog-raqamli o'zgartirishning navbatdagi natijasini ma'lumotlar massivlarining berilgan katakchalariga joylashtirish) bundan mustasno (bunday tipga xizmat ko'rsatish moduli PTS - Peripher - tranzaksiyaning periferiya serveri).

Uzilishli hodisalarni aniqlash mohiyati quyidagi harakatlar to'g'ri keladi:

- uzilishli hodisalar dalilini aniqlash;
- unga ta'sir etish zaruriyatini aniqlash;
- unga muayyan ta'sir zarurligini o'rnatish (ya'ni potensial mumkin bo'lgan uzilishli hodisalardan barcha MKning maqsadli funksiyalarini amalga oshirish uchun zarur emas, masalan, ARO'dan uzilishlar talab etilmasligi mumkin);
- hodisalar arbitrajini bajarish (avvalgi uzilishli hodisalarga xizmat ko'rsatish vaqtida ushbu hodisalarning muhimlik bosqichiga muvofiq hodisalarning yangi uzilishlari yuzaga kelganda boshlangan xizmat ko'rsatish davom ettirilishi, yoxud yangisiga xizmat ko'rsatish uchun qayta ulangan hodisani to'xtatish kerak);
- dasturiy uontekst qayta ulanishini bajarish (uzilishli dastur holatini saqlash va adresi uzilishli hodisalarning ("uzilish vektori" yoki "qayta ishlashga kirish nuqtasi" deb ataladigan) tipiga avvaldan bir xil xizmat ko'rsatish dasturiga qayta ulash).

3. BIR KRISTALLI MIKROKONTROLLERLAR

3.1. PIC mikrokontrollerlari tarkibi, parametrlari va tashkillashtirish xususiyatlari

Ushbu MK lar kam energiya sarflanishi bilan farqlanadi. Zamonaviy MK lar foydalanuvchiga juda ko‘p imkoniyatlar yaratib beradi va quyidagi asosiy rejimlar bilan farqlanadi:

- aktiv rejim (run mode) – MK asosiy rejimi. Bu rejimda MK ishchi dasturni qo‘llab, hamma resurslari ishchi xolatda;

- kutish rejimi (Wait mode, idle mode yoki halt mode). Ushbu rejimda markaziy prosessor ishni to‘xtatadi, lekin pereferiya modullari boshqaruv ob’ektidan o‘z nazoratlarini to‘xtatmaydi. Zarur bo‘lganda pereferiya moduli signallari MK larni aktiv rejimga o‘tkazadi. Bunday xolatlar MK larni qayta yuklaganda yoki uzilishlarni tashkil qilganda xosil bo‘ladi. Kutish rejimida MK quvvati PWAIT 5-10 martaga pasayadi.

- To‘xtatish rejimi (stop mode). MK STOP

- Kiritish-chiqarish portlari.

Har bir MK bir qancha kiritish-chiqarish liniyalariga ega bo‘lib, ular ko‘p razryadli (8-razryadli) parallel kiritish chiqarish portlariga ulangan. MK xotirasida har bir kiritish-chiqarish portlari o‘z adreslariga ega. Kiritish-chiqarish portlari ma’lumot registrlariga murojaat komandalari xotira ma’lumotlariga beriladigan komandalar asosida beriladi.

Undan tashqari ko‘pchilik MK larda alohida port razryadlarini tekshirish va ularga murojaat qilish imkoniyatlari ham mavjud. MK qo‘llaniladigan uzellar funksiyalardan foydalanishga qarab quyidagi parallel portlarni ajratish mumkin:

1. Bir yo‘nalishli portlar, ma’lumotlarni faqat kiritish yoki faqat chiqarish uchun;

2. Ikki yo‘nalishli portlar, MK ni initsializatsiya qilish davrida aniqlanadigan kiritish yoki chiqarish;

3. Multipleks portlar. Portlarning alohida liniyalari MK ning birlashtirilgan pereferiya qurilmalari bilan birga, bulardan taymer, ASP, ketma-ket interfeys kontrollerlari ishlatiladi;

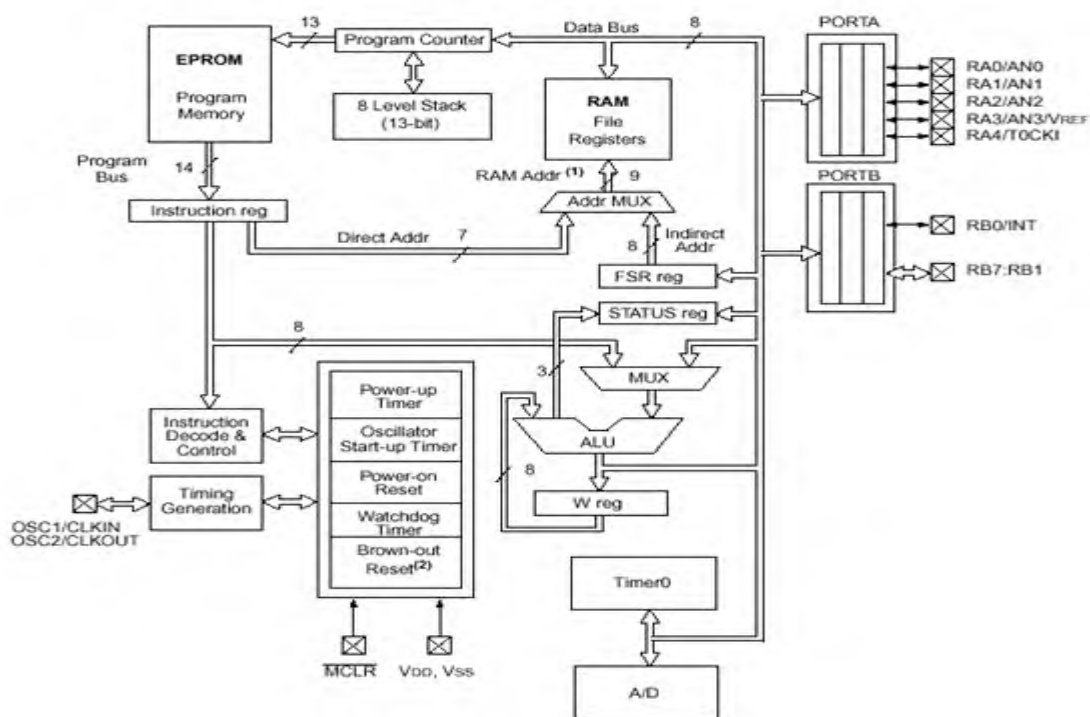
4. Kiritish-chiqarish bufer sxemotexnik boshqaruv dasturlanuvchi portlari.

Portlarning vazifasi vaqtinchalik MK larni ma'lum ob'ekt bilan aloqa o'rnatish uchun foydalaniladi.

Parallel portlar asosida MK va tashqi qurilmalar bilan ma'lumot almashish algoritmi 3 ta tip algoritmgiga ajratiladi:

1. Oddiy dasturiy kiritish-chiqarish;
2. Strobli kiritish-chiqarish;
3. Ma'lumot almashishni to'liq tasdiqlovchi kiritish-chiqarish rejimlari;

Garvard arxitekturasiga asoslangan bo'lib, ma'lumot shinalari va (OZU) ma'lumot xotirasi (kengligi 8-bit), dastur shinalari va dastur xotirasi 14-bit. Bunday arxitektura bit va bayt registr operatsiyalari yuqori tezlikda ishlashga imkon yaratadi [10]. O'tish komandalaridan tashqari hamma komandalar bir siklda bajariladi.



3.1- rasm. MK PICxxxxx struktura sxemasi

MK ikkita bankni tashkil qiladi:

- GPR-umumiy ishlatiladigan
- SFR-maxsus ishlatiladigan

UIRB to'g'ridan to'g'ri va bevosita adreslash usullari asosida foydalaniladi. Maxsus registrlar MK yadrosi va pereferiya modullarini boshqaradi. STATUS-registrida ALQ xolatlar bayroq registri bo'lib, u xolatlar yangilanish sabablarini ko'rsatadi.

Bit 7 IRP: Rezervlangan, 0 ga tenglik xolatini saqlash kerak

Bit 6 RP1: Rezervlangan, 0 ga tenglik xolatini saqlash kerak

Bit 5 RP0: ma'lumotlar xotirasi bankidan tanlash.

1 = Bank 1 (80h – FFh)

0 = Bank 0 (00h – 7Fh)

Bit 4 -TO: storajevoy taymer WDT bayrog'ining to'lib ketganligi

1 = POR qayta yangilanishi natijasida,

SLRWDT yoki SLEEP komandalarining bajarilishi

0 = to'lib ketganligida WDT

Bit 3 -PD: manba'dan uzilish bayrog'i detektori

1 = POR qatydan tiklanishi yoki SLRWDT komandasini bajarilishi

0 = SLEEP komandasini bajarilishi

Bit 2 Z: nolga tenglik bayrog'i

1 = arifmetik yoki logik operatsiyasining nollik natijasi

0 = arifmetik yoki logik operatsiyasining nolga teng emaslik natijasi

Bit 1 DC: o'nlik orttirmani uzatish

1 = kichik yarim baytdan uzatish

0 = kichik yarim baytdan uzatishlar qayd etilmagan

Bit 0 C: orttirmani uzatish

1 = katta bitdan uzatish

0 = katta bitdan uzatish qayd etilmagan

Registr OPTION REG

Registr OPTION REG bitlarni boshqarish tarkibiga ega bo'lib, o'qish va yozish kirishiga ega:

TMR0/WDT-signal bo'lgichlar

GP2/INT-tashqi uzilishlar aktiv fronti

TMR0-taymer

GPIO-kirish rezistori

Registr INTCON. Registr INTCON o'qish va yozish uchun ruxsat etilgan bo'lib, uzilishlarni tashkil qilish, to'lib ketish bayrog'i va bitlari tarkibiga ega. TMR0; GPIO chiqishlarida signal pog'onasini o'zgartirish; GP2/INT tashqi uzilishlar manba'si

Registr PIE1. Registr PIE1 o'qish va yozish uchun ruxsat etilgan bo'lib, uzilishlarni tashkil qilish, to'lib ketish bayrog'i va bitlari tarkibiga ega.

Registr PIR1. Registr PIR1 o'qish va yozish uchun ruxsat etilgan bo'lib, pereferiya modullari uzilishlari modullari tarkibiga ega

Registr PCON.Registr PCON –MK qayta tiklanish sabablari bayroq registri.

Registr PCLATH PCL .13-razryadli komanda xioblagichi registri PC bajarilayotgan instruksiya adresini ko'rsatadi. PC registrida bajarilayotgan xamma operatsiyalar qo'shimcha PCLATH registri orqali bajariladi. Stek. PIC12F629/675 8-pog'onali 13-razryad apparat stekga ega.

Stekdan ma'lumotlarni o'qish va yozish RETURN, RETLW, RETFUE komandalar asosida bajariladi .

Kiritish/chiqarish portlari GPIO.PIC12F629/675 mikrokontrollerlarida 6 ta kiritish/chiqarish port kanallari mavjud. Ba'zi bir kanallar MK ning qo'shimcha pereferiya modullari bilan multipleks bo'lgani uchun kiritish va chiqarish uchun foydalanilmaydi.

Registrlar GPIO va TRISIO. GPIO – 6-razryadli kiritish chiqarish porti. GPIO hamma kanallari TRISIO registriga mos bit yo'nalishlariga ega bo'lib,o'z navbatida kanallarni kiritish yoki chiqarishga moslashtiradi. Ma'lumotlar kerakli bo'limdan o'qib olinadi va taxlili qilingan natija chiqishga uzatiladi.

EEPROM ma'lumot xotirasi. EEPROMdan ma'lumotlarni o'qish va yozish baytma-bayt bajariladi.

EEDATA registri 8-razryadli o'qish va yozishni saqlaydi.EEADR esa EEPROMning xotira adres yacheykalarini saqlaydi [12]. PIC12F675 MKlari 128 bayt ma'lumot xotirasi bo'lib uning diapazoni 00h-7Fh.

PWRT_ manba'ga ulanish taymeri.

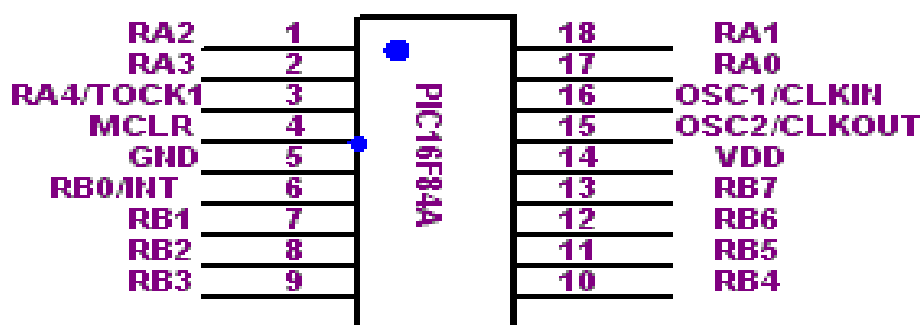
POR VA BOR qayta tiklanish sxemalari signallari asosida 72 ms ga manba'ga ulanish taymerlarini ta'minlash.

OST-generator ishga tushirish taymeri

3.2. PIC mikrokontrollerlari apparat vositalarining strukturasi va imkoniyatlari

Microchip Technology firmasi tomonidan ishlab chiqarilayotgan PIC16FXX integral mikrosxemalar oilasi o'zida RISC arxitekturali sakkiz razryadli mikrokontrollerlardan iborat. Bu mikroprotessorlar oilasi past energiya sarfi, komandalarni bajarish tezligi, past narhi bilan boshqalardan ajraladi. Mikrokontrollerlar o'rnatilgan PZU lardan tuzilgan bo'lib, ular OZU ma'lumotlari va energiyaga bog'liq bo'lgan dasturlarni saqlash uchun mo'ljallangan. Ushbu oilaning eng oddiy vakillaridan biri bu PIC16F84 mikrokontrolleridir. Kontroller KMOP

texnologiyasi bo'yicha tuzilgan bo'lib, ichki 1Kx14 bit electron qayta dasturlovchi PZU (EEPROM) dan iboratdir, bu qurilma bajarilayotgan dastur va 64 baytli energiyaga bog'liq OZU ma'lumotlarining saqlanishi uchun mo'ljallangan. Kontoller bajaradigan komandalar, kengligi 14 bit ikki so'zdan iborat bo'ladi va bir sikl davomida bajariladi (400 ns. Sinxronlashning 10 MgH chastotasida), bunda uzatish boshqaruvining komandalari ikki siklda bajariladi. Siklning davomiyligi taktli chastotaning to'rtta siklidan tashkil topadi. Kontroller 0 dan 10 MgH chastotali diapazonda ishlash qobiliyatini saqlab qoladi. Kontroller 4 manbadan uzilishni ta'minlaydi va ularning sakkiz darajali apparat oqimini ta'minlaydi. 8 bitli taymer/schetchik va 8 bitli dasturlanuvchi boshlang'ich ajratuvchiga ega. Periferik qurilmalarni ulash uchun (datchik, klaviatura, ijrochi mexanizmlar) uchun kontrollerning 13 ta liniyasi bor, ularning har biri dastur jihatidan axborotni kiritish va chiqarishga sozlangan. Bu liniyalarning yuqori quvvar qobiliyati (20mA) bo'lib, bu kirish va chiqish drayverlarining sxemali realizatsiyasini osonlashtiradi. Qurilmalarning ishlab chiqaruvchilari bunday kontrollerlarning asosida assembler yordamida tutib turiladi, bundan tashqari dasturlovchi va dasturli simulyator (PIC Simulator IDE) bilan tutib turiladi, bu esa o'z navbatida dasturlar asozlanishini alamga oshirish va kontrollerning portlariga ulangan tashqi qurilmalarning olib borilishini modellashtiradi.



3.2- rasm. Mikrokontrollerning funktsional sxemasi

3.2- rasmdagi sxemada quyidagi ko'rsatgichlar keltirilgan: OSC1 va OSC2 bo'yicha chiqish nuqtalari kvartslı rezonatorning yoki RC-zanjir, yoki taktli impulslar generatorining ulanishi uchun mo'ljallangan RA0-RA4 chiqishlari PORT A ning razryadlarini hosil qiladi. TOSC1 ham taymer/hisoblagichning kirish nuqtasi kabi ishlatilishi mumkin. RB0-RB7 chiqish nuqtalari B PORT B ning razryadlari hisoblanadi. INT- tashqi ulanishning kirishi. RB4, RB5 signalning o'zgarishi gacha

ulanish kirishi o'rnida ishlatilishi mumkin. RB6 - signalning o'zgarishi gacha ulanish kirishi o'rnida ishlatilishi mumkin yoki kontrollerning dasturlanishida taktli signalning kirishi or'nida ishlatilishi mumkin. RB7 – signalning o'zgarishi bo'yicha uzatishning kirish nuqtasi yoki ma'lumotlarni dasturlash vaqtidagi signal/ MCLR – tashlash signalining kirishi yoki dasturlash vaqtidagi kuchlanish. VDD – energiya manbasining musbat chiqishi. GND - umumiy chiqish

3.3. PIC mikrokontrollerlari buyruqlar tizimi va maxsus funksiyalari

3.1.- Jadval.

Arifmetik va mantiqiy instruksiyalar

Mnemonika	Operanda	Tasviri	Operasiya	Bayroq	Cikl
ADD	<u>Rd</u> , <u>Rr</u>	O'tkazishsiz umumlashtirish	$Rd = Rd + Rr$	Z,C,N,V,H,S	1
ADC	<u>Rd</u> , <u>Rr</u>	O'tkazishli umumlashtirish	$Rd = Rd + Rr + C$	Z,C,N,V,H,S	1
SUB	<u>Rd</u> , <u>Rr</u>	O'tkazishsiz hisoblash	$Rd = Rd - Rr$	Z,C,N,V,H,S	1
SUBI	<u>Rd</u> , <u>K8</u>	Konstantani hisoblash	$Rd = Rd - K8$	Z,C,N,V,H,S	1
SBC	<u>Rd</u> , <u>Rr</u>	O'tkazishli hisoblash	$Rd = Rd - Rr - C$	Z,C,N,V,H,S	1
SBCI	<u>Rd</u> , <u>K8</u>	O'tkazishli konstantani hisoblash	$Rd = Rd - K8 - C$	Z,C,N,V,H,S	1
AND	<u>Rd</u> , <u>Rr</u>	Mantiqiy Va	$Rd = Rd \cdot Rr$	Z,N,V,S	1
ANDI	<u>Rd</u> , <u>K8</u>	Konstantali mantiqiy Va	$Rd = Rd \cdot K8$	Z,N,V,S	1
OR	<u>Rd</u> , <u>Rr</u>	Mantiqiy Yoki	$Rd = Rd \vee Rr$	Z,N,V,S	1
ORI	<u>Rd</u> , <u>K8</u>	Konstantali mantiqiy Yoki	$Rd = Rd \vee K8$	Z,N,V,S	1
EOR	<u>Rd</u> , <u>Rr</u>	Mantiqiy chiqaruvchi Yoki	$Rd = Rd \oplus Rr$	Z,N,V,S	1

COM	<u>Rd</u>	Pobit invertsiyasi	$Rd = FF - Rd$	Z,C,N,V,S	1
NEG	<u>Rd</u>	Belgining o'zgarishi (qo'shimcha kod)	$Rd = \$00 - Rd$	Z,C,N,V,H,S	1
SBR	<u>Rd</u> , <u>K8</u>	Registrda bit ni o'rnatish	$Rd = Rd \vee K8$	Z,C,N,V,S	1
CBR	<u>Rd</u> , <u>K8</u>	Registrda bitlarni olib tashlash	$Rd = Rd (\$FF - K8)$	Z,C,N,V,S	1
INC	<u>Rd</u>	Registr belgisini inkrementasiyasi	$Rd = Rd + 1$	Z,N,V,S	1
DEC	<u>Rd</u>	Registr belgisini diskrementasiyasi	$Rd = Rd - 1$	Z,N,V,S	1
TST	<u>Rd</u>	Nol yoki manfiylikka tekshirish	$Rd = Rd / Rd$	Z,C,N,V,S	1
CLR	<u>Rd</u>	Registrni tozalash	$Rd = 0$	Z,C,N,V,S	1
SER	<u>Rd</u>	Registrni ornatish	$Rd = \$FF$	None	1
ADIW	<u>Rdl</u> , <u>K6</u>	Konstanta va so'zni kiritish	$Rdh:Rdl = Rdh:Rdl + K6$	Z,C,N,V,S	2
SBIW	<u>Rdl</u> , <u>K6</u>	So'zdan konstantani olib tashlash	$Rdh:Rdl = Rdh:Rdl - K6$	Z,C,N,V,S	2
MUL	<u>Rd</u> , <u>Rr</u>	Belgisiz sonlarni ko'paytirish	$R1:R0 = Rd * Rr$	Z,C	2
MULS	<u>Rd</u> , <u>Rr</u>	Belgili sonlarni ko'paytirish	$R1:R0 = Rd * Rr$	Z,C	2
MULSU	<u>Rd</u> , <u>Rr</u>	Belgili son va belgisiz sonni ko'paytirish	$R1:R0 = Rd * Rr$	Z,C	2

FMUL	<u>Rd</u> , <u>Rr</u>	Belgisiz kasrli sonlarni ko'paytirish	$R1:R0 = (Rd * Rr) \ll 1$	Z,C	2
FMULS	<u>Rd</u> , <u>Rr</u>	Belgili sonlarni ko'paytirish	$R1:R0 = (Rd * Rr) \ll 1$	Z,C	2
FMULSU	<u>Rd</u> , <u>Rr</u>	Belgili kasr son va belgisiz sonni ko'paytirish	$R1:R0 = (Rd * Rr) \ll 1$	Z,C	2

Bo'linish instruksiyalari

Mnemonika	Operanda	Tasviri	Operasiya	Bayroq	Sikl
RJMP	k	Nisbiy o'tish	$PC = PC + k + 1$	None	2
IJMP	Yo'q	(Z) ga bilsovita o'tish	$PC = Z$	None	2
EIJMP	Yo'q	(Z) ga bilvosita kengaygan o'tish	$STACK = PC + 1,$ $PC(15:0) = Z,$ $PC(21:16) = EIND$	None	2
JMP	<u>k</u>	O'tish	$PC = k$	None	3
RCALL	<u>k</u>	Dasturchanin g nisbiy chaqiruvi	$STACK = PC + 1,$ $PC = PC + k + 1$	None	3/4*
ICALL	Yo'q	(Z) ning bilvosita chariruvi	$STACK = PC + 1,$ $PC = Z$	None	3/4*
EICALL	Yo'q	(Z) ning bilvosita kengaygan chaqiruvi	$STACK = PC + 1,$ $PC(15:0) = Z,$ $PC(21:16) = EIND$	None	4*
CALL	<u>k</u>	Dasturchanin g chaqiruvi	$STACK = PC + 2,$ $PC = k$	None	4/5*

RET	Yo‘q	Dasturchadan qaytish	PC = STACK	None	4/5*
RETI	Yo‘q t	Uzilishdan qaytish	PC = STACK	I	4/5*
CPSE	<u>Rd</u> , <u>Rr</u>	Teng bo‘lsa solishtirish va o‘tkazish	if (Rd ==Rr) PC = PC 2 or 3	None	1/2/ 3
CP	<u>Rd</u> , <u>Rr</u>	Slishtirish	Rd -Rr	Z,C,N,V,H, S	1
CPC	<u>Rd</u> , <u>Rr</u>	O‘tishli solishtirish	Rd - Rr - C	Z,C,N,V,H, S	1
CPI	<u>Rd</u> , <u>K8</u>	Konstanta bilan solishtirish	Rd - K	Z,C,N,V,H, S	1
SBRC	<u>Rr</u> , <u>b</u>	O‘tkazish agar registrdagi bit toza bo‘lsa	if(Rr(b)==0) PC = PC + 2 or 3	Yo‘q	1/2/ 3
SBRS	<u>Rr</u> , <u>b</u>	O‘tkazish agar registrdagi bit o‘rnatilgan bo‘lsa	if(Rr(b)==1) PC = PC + 2 or 3	Yo‘q	1/2/ 3
SBIC	<u>P</u> , <u>b</u>	O‘tkazish agar portdagi bit toza bo‘lsa	if(I/O(P,b)==0) PC = PC + 2 or 3	Yo‘q	1/2/ 3
SBIS	<u>P</u> , <u>b</u>	O‘tkazish agar portdagi bit o‘rnatilgan bo‘lsa	if(I/O(P,b)==1) PC = PC + 2 or 3	Yo‘q	1/2/ 3
BRBC	<u>s</u> , <u>k</u>	O‘tish agar SREG dagi bayroq toza bo‘lsa	if(SREG(s)==0) PC = PC + k + 1	Yo‘q	1/2

BRBS	<u>s</u> , <u>k</u>	O'tish agar SREG dagi bayroq o'rnatilgan bo'lsa	if(SREG(s)==1) PC = PC + k + 1	Yo'q	1/2
BREQ	<u>k</u>	O'tish agar teng bo'lsa	if(Z==1) PC = PC + k + 1	Yo'q	1/2
BRNE	<u>k</u>	O'tish agar teng bo'lmasa	if(Z==0) PC = PC + k + 1	Yo'q	1/2
BRCS	<u>k</u>	O'tish agar o'tkazish o'rnatilgan bo'lsa	if(C==1) PC = PC + k + 1	Yo'q	1/2
BRCC	<u>k</u>	O'tish agar o'tkazish toza bo'lsa	if(C==0) PC = PC + k + 1	Yo'q	1/2
BRSH	<u>k</u>	O'tish agar tegn yoki yuqori bo'lsa	if(C==0) PC = PC + k + 1	Yo'q	1/2
BRLO	<u>k</u>	Kam bo'lsa o'tish	if(C==1) PC = PC + k + 1	Yo'q	1/2
BRMI	<u>k</u>	Minus bo'lsa o'tish	if(N==1) PC = PC + k + 1	Yo'q	1/2
BRPL	<u>k</u>	Plyus bo'lsa o'tish	if(N==0) PC = PC + k + 1	Yo'q	1/2
BRGE	<u>k</u>	Yuqori yoki teng bo'lsa o'tish (belgili)	if(S==0) PC = PC + k + 1	Yo'q	1/2
BRLT	<u>k</u>	Past bo'lsa o'tish (belgili)	if(S==1) PC = PC + k + 1	Yo'q	1/2
BRHS	<u>k</u>	Agar ichki o'tishni bayrog'i o'rnatilgan bo'lsa	if(H==1) PC = PC + k + 1	Yo'q	1/2

BRHC	<u>k</u>	Agar ichki o'tishni bayrog' toza bo'lsa	if(H==0) PC = PC + k + 1	Yo'q	1/2
BRTS	<u>k</u>	O'tish agar T bayrog'i o'rnatilgan bo'lsa	if(T==1) PC = PC + k + 1	Yo'q	1/2
BRTC	<u>k</u>	O'tish agar T bayrog'i toza bo'lsa	if(T==0) PC = PC + k + 1	Yo'q	1/2
BRVS	<u>k</u>	O'tish agar to'ldirish bayrog'i o'rnatilgan bo'lsa	if(V==1) PC = PC + k + 1	Yo'q	1/2
BRVC	<u>k</u>	O'tish agar to'ldirish bayrog'i toza bo'lsa	if(V==0) PC = PC + k + 1	Yo'q	1/2
BRIE	<u>k</u>	O'tish agar uzishlarga ruhsat bo'lsa	if(I==1) PC = PC + k + 1	Yo'q	1/2
BRID	<u>k</u>	O'tish agar uzishlarlar man qilingan bo'lsa	if(I==0) PC = PC + k + 1	Yo'q	1/2

Ma'lumotlarga operatsion kirish uchun sikllar soni ma'lumotlarning ichki xotirasiga kirish sharti bilan ko'rsatiladi va tashqi OZU bilan ishlaganda korrekt bo'lmaydi. CALL, ICALL, EICALL, RCALL, RET va RETI instruksiyalari uchun uchta sikl qo'shimcha ikkita siklni 16 bitli PC kontrollerlariga qo'shish kerak (128KB dasturlar hotirasi). Xotirasi 128 KB dan ortiq bo'lgan dasturlar uchun beshta sikl va yana har bir kutish uchun uchta sikl qo'shing.

Ma'lumotlarnig uzatish instruktsiyalari

Mnemonika	Operand	Tavsifi	Operatsiya	Flag	Sikl
MOV	<u>Rd</u> , <u>Rr</u>	Registrga nusxalash	$Rd = Rr$	Yo'q	1
MOVW	<u>Rd</u> , <u>Rr</u>	Registerlar juftini nusxalash	$Rd+1:Rd = Rr+1:Rr, r,d$ even	Yo'q	1
LDI	<u>Rd</u> , <u>K8</u>	Konstantani yuklash	$Rd = K$	Yo'q	1
LDS	<u>Rd</u> , <u>k</u>	To'g'ri yuklash	$Rd = (k)$	Yo'q	2*
LD	<u>Rd</u> , <u>X</u>	Bilvosita yuklash	$Rd = (X)$	Yo'q	2*
LD	<u>Rd</u> , <u>X+</u>	Post-inkrementli bilvosita yuklash	$Rd = (X),$ $X=X+1$	Yo'q	2*
LD	<u>Rd</u> , <u>-X</u>	Pre-dekrementli bilvosita yuklash	$X=X-1, Rd =$ (X)	Yo'q	2*
LD	<u>Rd</u> , <u>Y</u>	Bilvosita yuklash	$Rd = (Y)$	Yo'q	2*
LD	<u>Rd</u> , <u>Y+</u>	Post-inkrementli bilvosita yuklash	$Rd = (Y),$ $Y=Y+1$	Yo'q	2*
LD	<u>Rd</u> , <u>-Y</u>	Pre-dekrementli bilvosita yuklash	$Y=Y-1, Rd =$ (Y)	Yo'q	2*
LDD	<u>Rd</u> , <u>Y+q</u>	Egallashli bilvosita yuklash	$Rd = (Y+q)$	Yo'q	2*
LD	<u>Rd</u> , <u>Z</u>	Bilvosita yuklash	$Rd = (Z)$	Yo'q	2*
LD	<u>Rd</u> , <u>Z+</u>	Post-inkrementli bilvosita yuklash	$Rd = (Z),$ $Z=Z+1$	Yo'q	2*
LD	<u>Rd</u> , <u>-Z</u>	Pre-dekrementli bilvosita yuklash	$Z=Z-1, Rd =$ (Z)	Yo'q	2*
LDD	<u>Rd</u> , <u>Z+q</u>	Egallashli bilvosita yuklash	$Rd = (Z+q)$	Yo'q	2*
STS	<u>k</u> , <u>Rr</u>	To'g'ri saqlash	$(k) = Rr$	Yo'q	2*
ST	<u>X</u> , <u>Rr</u>	Bilvosita saqlash	$(X) = Rr$	Yo'q	2*
ST	<u>X+</u> , <u>Rr</u>	Post-inkremetnli bilvosita saqlash	$(X) = Rr,$ $X=X+1$	Yo'q	2*
ST	<u>-X</u> , <u>Rr</u>	Pre-dekrementli bilvosita saqlash	$X=X-1, (X)=Rr$	Yo'q	2*

ST	<u>Y</u> , <u>Rr</u>	Bilvosita saqlash	$(Y) = Rr$	Yo‘q	2*
ST	<u>Y+</u> , <u>Rr</u>	Post-inkremetnli bilvosita saqlash	$(Y) = Rr,$ $Y=Y+1$	Yo‘q	2
ST	<u>-Y</u> , <u>Rr</u>	Pre-dekremetnli bilvosita saqlash	$Y=Y-1, (Y) = Rr$	Yo‘q	2
ST	<u>Y+q</u> , <u>Rr</u>	Egallashli bilvosita saqlash	$(Y+q) = Rr$	Yo‘q	2
ST	<u>Z</u> , <u>Rr</u>	Bilvosita saqlash	$(Z) = Rr$	Yo‘q	2
ST	<u>Z+</u> , <u>Rr</u>	Post-inkremetnli bilvosita saqlash	$(Z) = Rr,$ $Z=Z+1$	Yo‘q	2
ST	<u>-Z</u> , <u>Rr</u>	Pre-dekrementli bilvosita saqlash	$Z=Z-1, (Z) = Rr$	Yo‘q	2
ST	<u>Z+q</u> , <u>Rr</u>	Egallashli bilvosita saqlash	$(Z+q) = Rr$	Yo‘q	2
LPM	Yo‘q	Dasturli hotiradan yuklash	$R0 = (Z)$	Yo‘q	3
LPM	<u>Rd</u> , <u>Z</u>	Dasturli hotiradan yuklash	$Rd = (Z)$	Yo‘q	3
LPM	<u>Rd</u> , <u>Z+</u>	Post-inkrementli dasturli hotiradan yuklash	$Rd = (Z),$ $Z=Z+1$	Yo‘q	3
ELPM	Yo‘q	Dasturli hotiradan kengaygan yuklash	$R0 = (RAMPZ:Z)$	Yo‘q	3
ELPM	<u>Rd</u> , <u>Z</u>	Dasturli hotiradan kengaygan yuklash	$Rd = (RAMPZ:Z)$	Yo‘q	3
ELPM	<u>Rd</u> , <u>Z+</u>	Post-inkrementli dasturli hotiradan kengaygan yuklash	$Rd = (RAMPZ:Z), Z = Z+1$	Yo‘q	3
SPM	Yo‘q	Dasturli hotiradan saqlash	$(Z) = R1:R0$	Yo‘q	-
ESPM	Yo‘q	Dasturli hotiradan kengaygan saqlash	$(RAMPZ:Z) = R1:R0$	Yo‘q	-
IN	<u>Rd</u> , <u>P</u>	Portni o‘qish	$Rd = P$	Yo‘q	1
OUT	<u>P</u> , <u>Rr</u>	Portga saqlash	$P = Rr$	Yo‘q	1

PUSH	<u>Rr</u>	Registrga ma'lumotlarni kiritish	STACK = Rr	Yo'q	2
POP	<u>Rd</u>	Registrdan ma'lumotlarni chiqarish	Rd = STACK	Yo'q	2

Ma'lumotlarga operatsion kirish uchun sikllar soni ma'lumotlarning ichki hotirasiga kirish sharti bilan ko'rsatiladi va tashqi OZU bilan ishlaganda korrekt bo'lmaydi. LD, ST, LDD, STD, LDS, STS, PUSH va POP instruktsiyalari uchun bitta siklni qo'shish kerak.

Bitlar bilan ishlash instruktsiyalari

Mnemonika	Operanda	Tasviri	Operatsiya	Bayroq	Sikl
LSL	<u>Rd</u>	Chapga mantiqiy burilish	$Rd(n+1)=Rd(n)$ $Rd(0)=0,$ $C=Rd(7)$	Z,C,N, V,H,S	1
LSR	<u>Rd</u>	O'ngga mantiqiy burilish	$Rd(n)=Rd(n+1)$ $Rd(7)=0,$ $C=Rd(0)$	Z,C,N,V,S	1
ROL	<u>Rd</u>	C orqali siklli chap tomonga burilish	$Rd(0)=C,$ $Rd(n+1)=Rd(n)$ $C=Rd(7)$	Z,C,N, V,H,S	1
ROR	<u>Rd</u>	C orqali siklli o'ng tomonga burilish	$Rd(7)=C,$ $Rd(n)=Rd(n+1)$ $C=Rd(0)$	Z,C,N,V,S	1
ASR	<u>Rd</u>	O'ng tarafga arifmetik burilish	$Rd(n)=Rd(n+1)$ $n=0,\dots,6$	Z,C,N,V,S	1
SWAP	<u>Rd</u>	Tetradlarning qayta joylashtirish Perestanovka tetrad	$Rd(3..0) =$ $Rd(7..4),$ $Rd(7..4) =$ $Rd(3..0)$	Yo'q	1

BSET	<u>s</u>	Bayroqni oʻrnatish	SREG(s) = 1	SREG(s)	1
BCLR	<u>s</u>	Bayroqni tozalash	SREG(s) = 0	SREG(s)	1
SBI	<u>P,b</u>	Portda bitni oʻrnatish	I/O(P,b) = 1	Yoʻq	2
CBI	<u>P,b</u>	Portdagi bitni tozalash	I/O(P,b) = 0	Yoʻq	2
BST	<u>Rr,b</u>	Registrdan T ga bitni koʻchirish	T = Rr(b)	T	1
BLD	<u>Rd,b</u>	T dan registrga bitni yuklash	Rd(b) = T	Yoʻq	1
SEC	Yoʻq	Koʻchish bayrogʻini oʻrnatish	C = 1	C	1
CLC	Yoʻq	Koʻchish bayrogʻini tozalash	C = 0	C	1
SEN	Yoʻq	Manfiy raqamli bayroqni oʻrnatish	N = 1	N	1
CLN	Yoʻq	Manfiy raqamli bayroqni tozalash	N = 0	N	1
SEZ	Yoʻq	Noldan bayroqni oʻrnatish	Z = 1	Z	1
CLZ	Yoʻq	Bayroqni noldan tozalash	Z = 0	Z	1
SEI	Yoʻq	Uzilishlar bayrogʻini oʻrnatish	I = 1	I	1

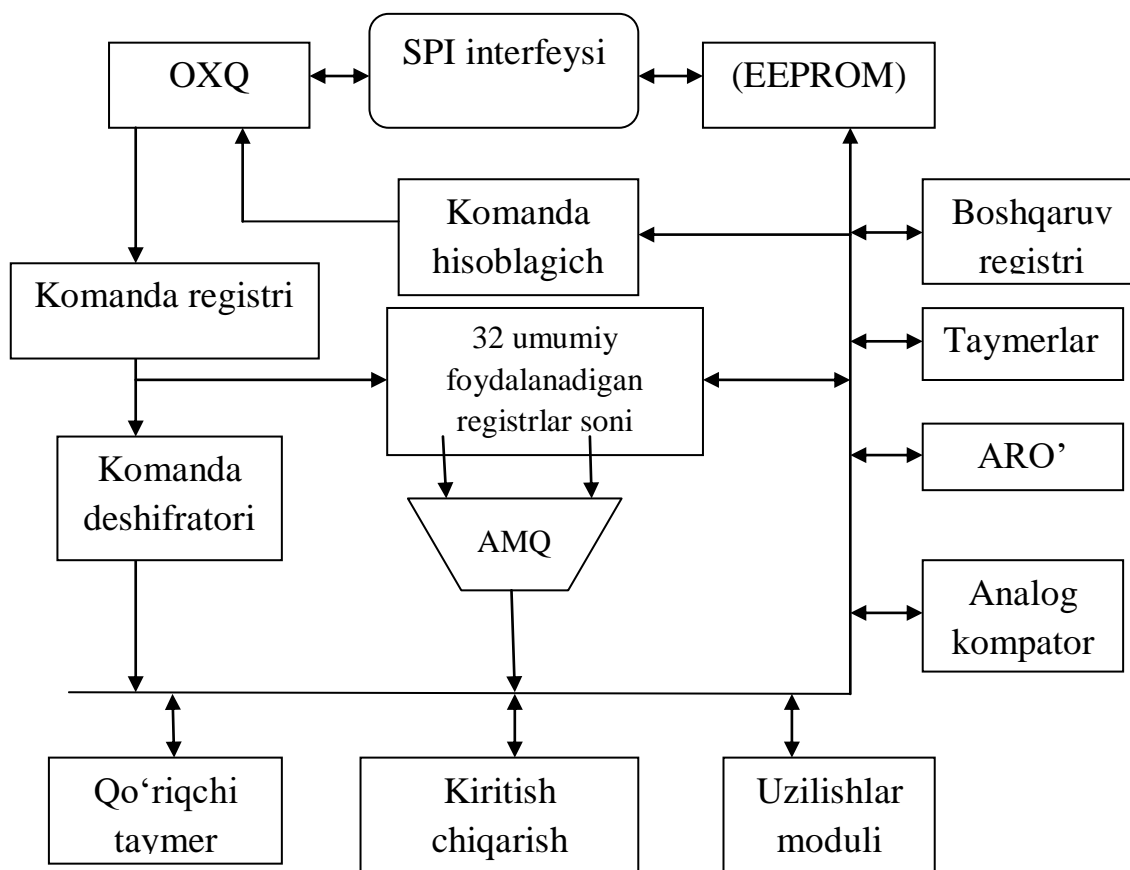
CLI	Yo'q	Uzilishlar bayrogini tozalash	$I = 0$	I	1
SES	Yo'q	Sonli belgili bayroqni o'rnatish	$S = 1$	S	1
CLN	Yo'q	Sonli belgili bayroqni tozalash	$S = 0$	S	1
SEV	Yo'q	To'ldirish bayrog'ini o'rnatish	$V = 1$	V	1
CLV	Yo'q	To'ldirish bayrogini tozalash	$V = 0$	V	1
SET	Yo'q	T bayrog'ini o'rnatish	$T = 1$	T	1
CLT	Yo'q	T bayrog'ini tozalash	$T = 0$	T	1
CLH	Yo'q	Ichki ko'chish bayrogini tozalash	$H = 0$	H	1
NOP	Yo'q	Operatsiya Yo'q	Yo'q	Yo'q	1
SLEEP	Yo'q	Uhlash (energiya kamaytirish)	Instruktsiya tasvirini ko'ring	Yo'q	1

3.4. AVR mikrokontrollerlari tarkibi, parametrlari va tashkillashtirish xususiyatlari

AVR mikrokontrollerlari 1990 yillarning oxirida paydo bo'lgan. Kundan kunga ularning turli xil turlari paydo bo'layapti va keng ommaga taqdim qilinayapti. Hozirgi davrda bir turdagi arxitekturaga ega

mikrokontrollerlar turlari AVR mikrokontrollerlari mavjud bo‘lib, ular quyidagi oilaga mansub:

- Classic AVR (ishlab chiqarishdan olib tashlanyapti);
- Tiny AVR;
- Mega AVR;
- Mega AVR maxsus qo‘llanishlar uchun.



3.3- rasm. AVR-mikrokontrolleri protsessorli yadrosi arxitekturasi

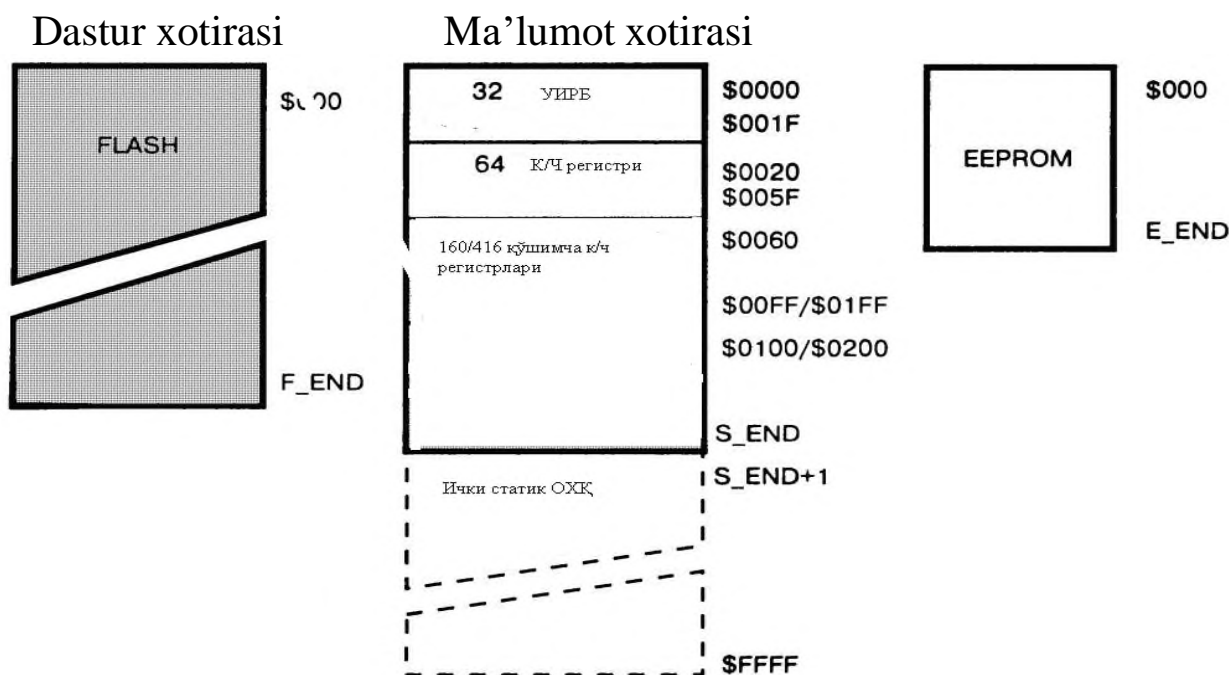
Mikrokontroller oilasi Tiny chegaralangan pereferiyaga ega bo‘lib, katta bo‘lmagan dasturiy va ma‘lumot xotirasiga ega.

Mikrokontrollerlar oilasiga mansub Mega mikrokontrollerlari rivojlangan pereferiyaga ega bo‘lib, shu oilaga mansub bo‘lgan MK lar uchun nisbatan keng ma‘lumot va dasturiy xotiraga ega. Ular mobil telefonlarda, CD-ROM/DVD-ROM uchun foydalaniladi.

AVR-mikrokontrollerlar protsessorli yadrosi (CPU) (1.1-rasm) da ko‘rsatilganidek RISC-arxitekturasi mansub bo‘lib (enhanced RISC), mikrokontrollerlar ish tezligini oshirishda foydalaniladi.

Arifmetik-mantiqiy qurilma (AMQ)da hisob amallari bajariladi, ular bevosita 32 ishchi registrga murojaat etadi. Shuning uchun AMQ bir takt ichida vazifalarni bajaradi.

Undan tashqari har bir komanda bir xotira yacheykasini egallaydi. AVR mikrokontrollerlarida Garvard arxitekturasi qoʻllaniladi. Bunday tashkillashtirish bir vaqt ichida dasturiy xotira va maʼlumot xotirasi bilan ishlash imkoniyatini yaratadi.



3.4– rasm. AVR- Mega mikrokontrollerlari xotira kartasi

Maʼlumot shinalarining boʻlinishi turli xotira uchun turlicha razryadlar va adreslash usullari bilan farqlanadi. Ikki pogʻonali konveyer usuli (bir komanda bajarilayotganda parallel ravishda ikkinchi komanda tanlanib dekoderlanadi) ushbu arxitektura 1 MGs takt chastotada 1 MIPS operatsiyani bajarish imkoniyatini beradi.

AVR Mega oilasiga mansub mikrokontrollerlar xarakteristikasi:

- FLASH-dasturiy xotira sigʻimi 8 dan 256 Kbayt (oʻchirib yozish sikli 10 000);
- operativ xotira (statik OXQ) sigʻimi 512 bayt dan 8 Kbayt;
- maʼlumot xotirasi (EEPROM) sigʻimi 256 bayt dan Kbaytgacha (oʻchirib yozish sikli 100 000);
- oʻqishdan himoyalaniish sikli imkoniyati;
- interfeyslar SPI va JTAG orqali dasturlash imkoniyatining mavjudligi;

- o‘z o‘zini dasturlash imkoniyatning mavjudligi;
- turli usulda sinxronizatsiyalash;
- o‘rnatilgan RS-generator;
- bir qancha kam manba’ talabligi;
- qo‘riqchi taymer mavjudligi;
- dasturiy takt generator chastotasini pasayishi.

Periferiya moduli.

Mikrokontroller AVR Mega 23 dan 86 gacha kiritish chiqarish liniyalariga ega. Undan tashqari pereferiya qurilmalariga ega:

- bir yoki ikkita 8-bitli taymer/hisoblagich;
- bittadan 4 gacha 16-bit taymer/hisoblagich;
- bir- ikki kanalli generatorlar 8-bitli ShIM-signal ;
- uch- va 4 kanalli generatorlar ShIM;
- analog komparator;
- ko‘p kanalli 10-bitli ARO‘ ;
- ketma-ket sinxron interfeys SPI;
- ketma-ket ikki yo‘nalishli interfeys TWI ;
- birdan 4 gacha to‘liq dupleksli universal sinxron/asinxron qabul qilib-uzatgich (USART);

Dastur xotirasi (sig‘imi 8 dan 256 Kbayt) mikrokontroller ish jarayonini boshqarishda foydalaniladigan komandalarni saqlashda foydalaniladi. Dastur xotirasidan ma’lumot xotirasiga baytlarni uzatishda maxsus komanda — LPM foydalaniladi.

3.2- jadval

ATmega8535 mikrokontrolleri kiritish chiqarish registrlari

Nomi	Adres	Funksiya
SREG	\$3F (\$5F)	Xolat registri
SPH	\$3E (\$5E)	Stek ko‘rsatkichi, katta bayt
SPL	\$3D (\$5D)	Stek ko‘rsatkichi, kichik bayt
OCR0	\$3C (\$5C)	Taymer/hisoblagich TO mosligi
GICR	\$3B (\$5B)	Uzilishlarni boshqarish registri
GIFR	\$3A (\$5A)	Uzilishlar umumiy bayroq registri
TIMSK	\$39 (\$59)	Taymer/hisoblagich uzilishlari bayroq registri
TIFR	\$38 (\$58)	Taymer/hisoblagich uzilishlari bayroq registri

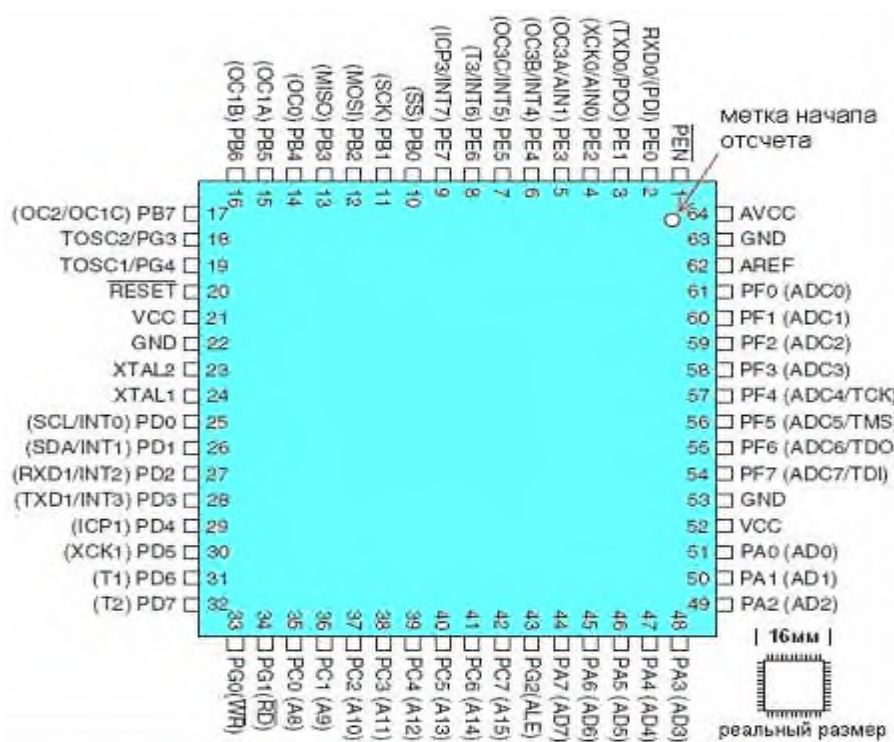
SPMCR	\$37 (\$57)	Dastur xotirasiga operatsiyalarni yozish xolatini boshqarish registri.
TWCR	\$36 (\$56)	Boshqarish registri TWI
MCUCR	\$35 (\$55)	Mikrokontrollerni boshqarish registri
MCUC SR	\$34 (\$54)	Mikroonroller xolati va boshqarish registri
TCCR0	\$33 (\$53)	Taymer/hisoblagichni TO boshqarish registri
TCNT0	\$32 (\$52)	Taymer/hisoblagich TO hisob registri
OSCCAL	\$31 (\$51)	Takt generator kalibrovka registri
SFIOR	\$30 (\$50)	Maxsus registr funksiyasi
TCCR1A	\$2F (\$4F)	Registr A taymer/hisoblagich T1 boshqarish
TCCR1B	\$2E (\$4E)	Registr V taymer/hisoblagich T1 boshqarish
TCNT1H	\$2D (\$4D)	Taymer/hisoblagich T1, katta bayt xisob registri
TCNT1L	\$2C (\$4C)	Taymer/hisoblagich T1, kichik bayt xisob registri
OCR1AH	\$2B (\$4B)	Registr A taymer/hisoblagich T1, katta bayt mosligi
OCR1AL	\$2A (\$4A)	Registr A taymer/hisoblagich T1, katta bayt mosligi
OCR1BH	\$29 (\$49)	Registr V taymer/hisoblagich T1, katta bayt mosligi
OCR1BL	\$28 (\$48)	Registr A taymer/hisoblagich T1, kichik bayt mosligi
ICR1H	\$27 (\$47)	Taymer/hisoblagich T1, katta bayt egallash
ICR1L	\$26 (\$46)	Taymer/hisoblagich T1, kichik bayt egallash
TCCR2	\$25 (\$45)	Taymer/hisoblagich T2 boshqarish registri
TCNT2	\$24 (\$44)	Taymer/hisoblagich T2 xisob registri
OCR2	\$23 (\$43)	Taymer/hisoblagich T2 moslik registri
ASSR	\$22 (\$42)	Asinxron rejim registr xolati
WDTCR	\$21 (\$41)	Qo‘riqchi taymer boshqarish registri
UBRRH	\$20 (\$40)	Uzatish registri USART, katta bayt

UCSRC		Boshqarish registri va xolati USART
EEARH	\$1F (\$3F)	Adres registri EEPROM, katta bayt
EEARL	\$1E (\$3E)	Adres registri EEPROM, kichik bayt
EEDR	\$1D (\$3D)	Ma'lumot registri EEPROM
EECR	\$1C (\$3C)	Boshqarish registri EEPROM
PORTA	\$1B (\$3B)	Port A ma'lumot registri
DDRA	\$1A (\$3A)	Port A ga ma'lumot yo'naltirish registri
PINA	\$19 (\$39)	Port A chiqishlari
PORTB	\$18 (\$38)	Port V ma'lumot registrlari
DDRB	\$17 (\$37)	Port V ma'lumot yo'naltirish registri
PINB	\$16 (\$36)	Port V chiqishlari
PORTC	\$15 (\$35)	S port ma'lumot registri
DDRC	\$14 (\$34)	S port ma'lumot registri yo'naltirish
PINC	\$13 (\$33)	S port chiqishlari
PORTD	\$12 (\$32)	D port ma'lumot registri
DDRD	\$11 (\$31)	D port ma'lumot registri yo'naltirish
PIND	\$10 (\$30)	Port D chiqishlari
SPDR	\$0F (\$2F)	SPI ma'lumot registri
SPSR	\$0E (\$2E)	SPI xolatlar registri
SPCR	\$0D (\$2D)	SPI boshqarish registri
UDR	\$0C (\$2C)	USART ma'lumot registri
UCSRA	\$0B (\$2B)	Registr A xolat va boshqarish USART
UCSRB	\$0A (\$2A)	Registr V xolat va boshqarish USART
UBRR1	\$09 (\$29)	Uzatish tezligi registri USART, kichik bayt
ACSR	\$08 (\$28)	Analog komparator xolati va boshqarish registri
ADMUX	\$07 (\$27)	ASP multipleksorini boshqarish registri
ADCSRA	\$06 (\$26)	Registr A boshqarish va xolati ASP
ADCH	\$05 (\$25)	ASP ma'lumot registri, katta bayt
ADCL	\$04 (\$24)	ASP ma'lumot registri, kichik bayt
TWDR	\$03 (\$23)	Ma'lumot registri TWI
TWAR	\$02 (\$22)	Adres registri TWI
TWSR	\$01 (\$21)	Xolat registri TWI
TWBR	\$00 (\$20)	Uzatish registri TWI

Mikrokontrollerlar oilasiga mansub Mega mikrokontrollerlari rivojlangan pereferiyaga ega bo'lib, shu oilaga mansub bo'lgan MK lar

uchun nisbatan keng ma'lumot va dasturiy xotiraga ega. Ular mobil telefonlarda, CD-ROM/DVD-ROM uchun foydalaniladi.

3.5. ATmega128 mikrokontrollerining strukturasi va imkoniyatlari



3.5- rasm. ATmega128 mikrokontroller chiqishlari belgilanishi.

Mikrokontroller ATmega128 quyidagi funksional bloklar:

- 8-razryadli arifmetik mantiqiy qurilma (AMQ);
- ichki 128 Kbaytli dasturiy flash-xotira;
- 32 umumiy foydalanuvchi registr;
- 4 Kbaytli ichki EEPROM ma'lumot xotirasi;
- 4 Kbaytli ichki OXQ;
- 6 ta parallel 8-razryadli port;
- 4 ta dasturlanuvchi taymera-hisoblagich;
- 10-razryadli 8-kanalli ARO' va analog komparator;
- ketma ket interfeyslar UART0, UART0, TWI va SPI;
- uzilishlar va boshqarish bloki .

Port A (PA7..PA). 8-razryadli ikki yo'nalishli port.

Port B (PB7..PB0). 8-razryadli ikki yo'nalishli port. O'rnatilgan rezistorli. Port V maxsus funksiyalardan foydalanishda ham ishlatiladi.

Port C (PC7..PC0). Port S 8-razryadli chiqish porti. Port C adres shinasini tashkil qilishda foydalaniladi.

Port D (PD7..PD0). 8-razryadli ikki yo‘nalishli port. Port D maxsus funksiyalarni ishga tushirishda foydalaniladi.

Port E (PE7..PE0). 8-razryadli ikki yo‘nalishli port. Port E maxsus funksiyalarda foydalaniladi.

Port F (PF7..PF0). 8-razryadli kirish porti. Kirish portlari analog raqamli o‘zgartirgichlarning analog kirishlari ko‘rinishida foydalaniladi.

#RESET. Qayta yuklash kirishi. 50 ns vaqtni ushlab turish kerak.

XTAL1, XTAL2. Invertirlanuvchi kuchaytirgich takt chastotasini kirish chiqishi.

TOSC1, TOSC2. Taymer hisoblagich generator kuchaytirgichi inverterli kiritish chiarish.

#WR, #RD. Tashqi ma’lumot xotirasi o‘qish yozish strobi.

ALE. Tashqi xotira adresi fiksatsiyasi strobi. AD0-AD7 ma’lumotlarni uzatishda foydalaniladi.

AVCC. Analog raqamli o‘zgartirgichlar manba’ ta’minoti.

AREF. Analog raqamli o‘zgartirgich uchun tayanch kirish.

#PEN. Ketma ket interfeys orqali dasturlanishga ruxsat chiqishlari nterfeysi. Ushbu chiqishda signalning ushlab turilishi qurilma ketma ket kanal orqali dasturlanish rejimiga o‘tadi.

VSS, GND. 8-razryadli ikki yo‘nalishli port. Manba’ ta’minoti va yer.

Mikrokontroller kiritish/chiqarish portlari va xotirani tashkil qilish. Mikrokontroller AVR Garvard arxitekturasiga asoslangan bo‘lib, dasturiy va ma’lumot xotirasi alohida maydonda joylashgan.

AVR arxitekturasining afzalliklari: - Dasturiy xotira sifatida ichki flesh-xotira ishlatiladi. U 16-razryadli yacheykalardan tashkil topgan massiv ko‘rinishida bo‘lib port SPI orqali yuklanadi;

ma’lumot xotirasi 8-razryadni tashkil qiladi. Kichik 32 adres maydonni UFRB tashkil qiladi, 64 k/ch registr adresiniv, ichki OXQ sig‘imi 4096 yacheykani.OXQ 60 Kbaytgacha;

3.2.- Jadval.

ATMega128 xotirasin tashkillashtirish

Dastur xotirasi		Ma’lumot xotirasi	
Amaliy xotira sektori	\$0000	32 registrlar	\$0000-\$001F
		64 registr k/ch	\$0020-\$005F
		160 k/ch kengaytirilgan registr	\$0000-\$00FF

		Ichki YuOXQ (4096x8)	\$0100 \$10FF
		Tashqi YuOXQ (64x8)	\$1100 \$FFFF
Dastur boshlang'ich yuklanish sektori	\$FFFF		

- Ichki EEPROM sig'imi 4 Kbaytgacha bo'lib unga murojaat maxsus k/ch registrlar orqali bo'ladi.

	7	0	Addr.	
	R0		0x00	
	R1		0x01	
	R2		0x02	
	...			
	R13		0x0D	
	R14		0x0E	
	R15		0x0F	
General Purpose Working Registers	R16		0x10	
	R17		0x11	
	...			
	R26		0x1A	X-register Low Byte
	R27		0x1B	X-register High Byte
	R28		0x1C	Y-register Low Byte
	R29		0x1D	Y-register High Byte
	R30		0x1E	Z-register Low Byte
	R31		0x1F	Z-register High Byte

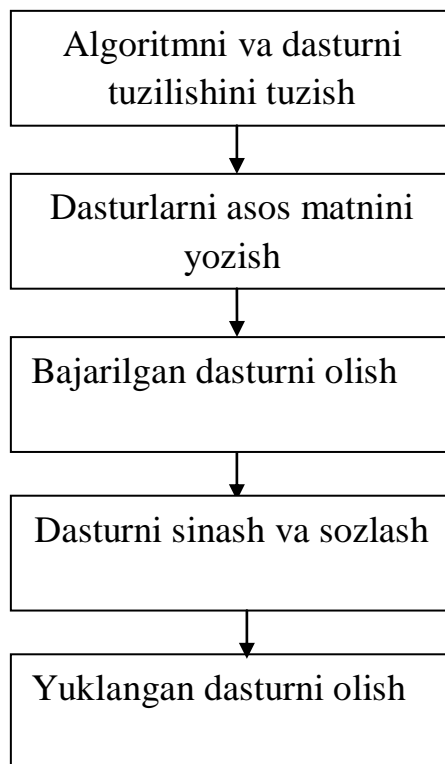
3.6- rasm. ATmega128 mikrokontroller umumiy foydalaniladigan registrlar bloki

3.6. Mikrokontrollerlar asosidagi tizimlarning imkoniyatlarini kengaytirish va sifatini oshirish uchun maxsus funksiyalar

AVR-mikrokontrollerlarining komandalar tizimiga arifmetik va logik operatsiyalar komandalari (instruksiyalari), tarmoqlanish komandalari, dasturning ketma-ketlikda bajarilishini boshqarish komandalari, ma'lumotlarni uzatish komandalari va bitlar bilan bajariladigan operatsiyalar komandalari kiradi. Umumiy komandalar

tizimiga 130 dan ortiq instruksiyalar kiradi. Mikrokontrollerlarning eski modellarida ba'zi bir komandalarni qo'llab bo'lmaydi.

Mikrokontrollerlarni dasturlash. Bir kristalli mikrokontrollerlar asosida dasturiy ta'minotni tuzish jarayoni quyidagi bosqichlarni o'z ichiga oladi (3.7. - rasm):



3.7-rasm. Bir kristalli mikrokontrollerlar asosida dasturiy ta'minotni tuzish jarayoni.

- algoritmni va dasturni tuzilishini (strukturasini) tuzish;
- dasturni asos matnini yozish;
- bajarilgan dasturni olish;
- dasturni sinash va sozlash;
- yuklangan dasturni olish.

Algoritmni va dasturni tuzilishini (strukturasini) tuzish jarayonida vazifani bajarish metodi tanlaniladi va uni amalga oshirish algoritmi tuziladi. Algoritm bu – qoidalar jamlamasi yoki ma'lum bir vazifani bajarilish operatsiyalarini ketma-ketlikda yozilishidir. Algoritmni grafik ko'rinishi - algoritm sxemasidir. Dasturiy ta'minotni yaratish bosqich tarkibi, uni modellarda translyatsiyalash va sozlashi foydalaniladigan tizimli vositalarga sezilarli darajada bog'liq. Hozirgi vaqtda 8-razryadli MK resurslari yuqori darajali dasturlash tillarini quvvatlash uchun

yetarli. Bu tarkibiy dasturlashning barcha afzalliklaridan foydalanish imkoniyatini beradi, alohida translyatsiyalanish modullarini ishlatib dasturiy ta'minotni loyihalashtirish kabi. Bir vaqtning o'zida quyi darajali dasturlash tili assemblerda ham keng miqyosida foydalanilmoqda, ayniqsa kerak bo'lgan vaqt oralig'i nazoratlanishi zarur bo'lganda

Dasturni asos matnini yozish jarayonida tuzilgan algoritmdasturning asos tili ko'rinishida yoziladi (assemblerda yoki yuqori darajali tilda).

Assembler tili – har bir protsessor komandasiga yoki protsessor komandalarining jamlamasiga qisqartirilgan ramziy yozuv (mnemonika) mos keladigan dastur tilidir.

Komandalarni ramziy belgilashni, hamda registr va xotira yacheykalarini adresini, o'zgaruvchilarni, o'zgarmaslarni (konstant) va boshqa dastur elementlarni ishlatish, dasturni tuzish jarayonini osonlashtiradi. Elementlarni ramziy belgilash odatda ularni mazmunini yoritib beradi.

Assembler tili, dasturlanayotgan mikroprotsessorning (mikrokontrollerning) hamma resurslariga yo'l ochib beradi va tezlik jihatdan va xotirada egallanadigan joy bo'yicha effektiv dastur tuzishga imkon beradi. Shuning uchun assembler tilida dasturlash mikroprotsessorning xususiyatini va arxitekturasini bilishni talab etadi.

Assembler tilari har xil turdagi mikroprotsessorlar uchun ajraladi. Bir qator assemblerlarda bitta makrokomandalar (makros) singari qaytariladigan ketma-ketlikdagi komandalarni ro'yxatga olishga yo'l qo'yiladi. Bunday assemblerlarni makroassemblerlar deb atashadi.

Yuqori darajali tillar (S, Paskal, Beysik va boshqalar), assembler singari, mikrokontrollerning hamma resurslariga yo'l ochib beradi, lekin bundan tashqari yaxshi tuzilishga ega dasturlarni tuzishga imkon beradi, dasturchidan xotirani bo'lish tashvishidan holi qiladi va standart operatsiyalarni bajarish uchun ko'p biblioteka funksiyalari to'plamiga ega.

Bajarilgan dasturni olish bosqichida dasturning asoasiy matni maxsus vositalar (translyatorlar, kompilyatorlar, joylashtirgichlar va boshqalar) yordami bilan bajariladigan kodga aylanadi. Translyator (translyator) bu – assembler tilidagi dasturni, protsessor “tushunadigan” mashinaviy kodga o'girish uchun mo'ljallangan dasturdir.

Kompilyator bu – yuqori darajali tilda dastur matnini, ekvivalent mashinaviy kodga o'giradigan dasturdir. Ob'ekt modullaridan

tuzilayotgan dasturni qurish uchun joylashtirgich (linker) qo‘llaniladi. Dasturni asos matnidan, bajarilayotgan dasturni olish jarayonida, ishlatilayotgan dastur tilida sintaksis qoidasi buzilgan statik xatolar yo‘q qilinadi.

Dasturni sinash va sozlash bosqichida izlash, joylashish va unda logik xatolarni bartaraf qilish amalga oshiriladi. Sinash, dasturda xatolarni aniqlash uchun xizmat qiladi va ba’zi bir matnli ma’lumotlarni jamlamasidan foydalanilgan holda bajariladi.

Matnli ma’lumotlar algoritmini hamma tarmoqlarini tekshiruv bilan ta’minlamoqi lozim. Dasturni sinoViddan keyin sozlanishi (debug) lozim. Buning vazifasi xatolarni yo‘qotishdir, ya’ni dasturda xatolik berayotgan joylarni topishdan iborat.

Yuklangan dasturni olish bosqichida sinash va sozlash uchun ishlatilgan ortiqcha parchalardan (qismlardan) dasturni “bo‘shatish” amalga oshiriladi. Bu parchalar dastur hajmini oshiradi va mikroprotsessori tizimini normal ishlashi uchun kerak emas. Keyin olingan yuklash dasturi mikrokontroller xotirasiga kiritiladi.

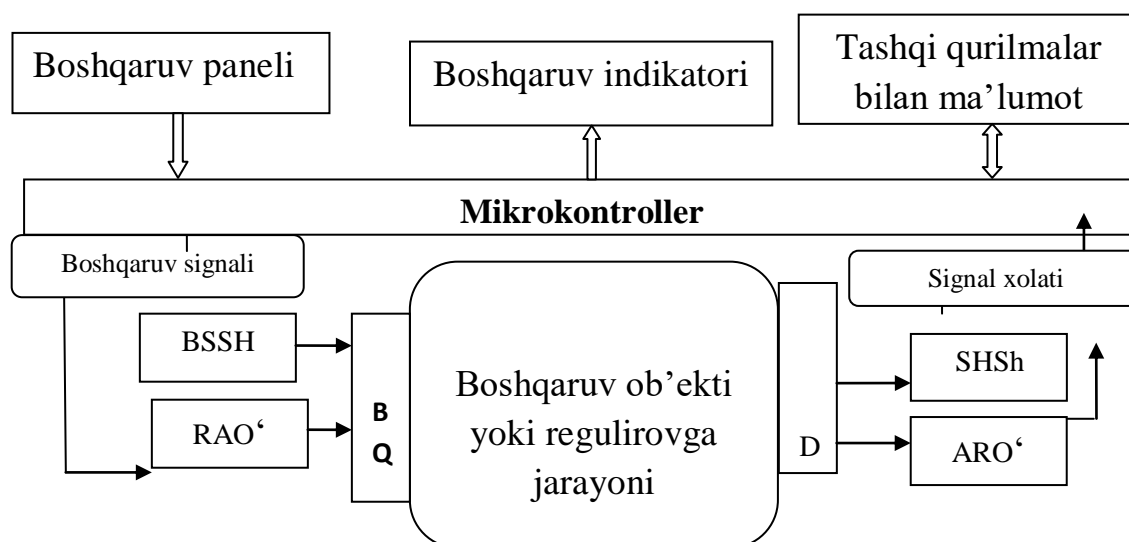
O‘rnatiladigan mikroprotsessornlarni dasturiy ta’minotini (DT) tuzish, maxsus dasturiy va apparat vositalarini ishlatgan holda, shaxsiy kompyuterlarda bajariladi. Dasturiy ta’minot bunday tuzish turi kross-ishlab chiqish deb ataladi. Dasturiy ta’minot ishlab chiqishda va sozlashda qo‘llaniladigan apparat va dasturiy vositalarni jamlamasini, bitta nom bilan – ishlab chiqishni qo‘llab-quvvatlash vositalari deb atashadi.

Dasturiy ta’minotni tekshirish va sozlash uchun dasturiy simulyator deb nomlanuvchi vositadan foydalaniladi, u foydalanuvchiga yaratilgan dasturni MK ning dasturiy-mantiqiy modelida bajarish mikonini beradi.

4. MIKROKONTROLLERLARLAR ASOSIDA QURILMALARINI LOYIHALASHTIRISH

4.1. Mikrokontrollerlarning dasturiy ta'minotini yaratish va apparat ta'minoti bilan integrasiyalash

Mikrokontrollerlar – hisoblash asboblari, qurilmalar va har xil vazifalarda qo'llaniladigan tizimlarda ishlatiladigan mikroprotsektorlarni eng keng sinfini tashkil qiladi. Mikrokontrollerlar bu – texnik ob'ektlarni boshqarish qurilmalarni va texnologik jarayonlarni hosil qilish uchun mo'ljallangan maxsus mikroprotsektoridir [15]. Tuzilishi jihatidan mikrokontrollerlar, kristalida hisoblash tizimining hamma tarkib qismlari: mikroprotsektor, xotira, hamda qo'shimcha funksiyalarni amalga oshirish uchun periferiya qurilmalari joylashtirilgan, katta integral sxemani (KIS) tashkil qiladi.



4.1- rasm. Mikrokontroller asosidagi boshqaruv tizimining tuzilishi: BSSH – boshqaruv signallarini shakillantirgichlar; BQ – bajaruvchi qurilma; D – datchiklar; SHSh – signallar holatini shakillantirgichlar.

Mikrokontrollerli boshqaruv tizimiga mikrokontroller va u bilan boshqaruv ob'ektini ulash (biriktirish) qurilmasi kiradi (4.1- rasm) [13].

Mikrokontrollerlarning hamma elementlari bitta kristalda joylashgani uchun, ularni bir kristalli (bir korpusli) mikro Elektr hisoblash mashinalari yoki bir kistalli mikrokontrollerlar deb ham atashadi. Mikrokontrollerlarni qo'llashdan maqsad – komponentlar sonini qisqartirish, o'lchamini kamaytirish va qurilmani (tizimni) narxini tushirishdir.

Odatda, mikrokontrollerlar RISC-arxitekturasiga (RISC – Reduced Instruction Set Computer), kam xajmli xotiraga, fizik va logik bo‘lingan dastur xotirasi va komandalar tizimini boshqarish uchun mo‘ljallangan ma‘lumot xotirasiga ega. Shunday qilib, mikrokontrollerlar boshqarish masalasini yechish, nazorat, tartibga solish va ma‘lumotlarga dastlabki ishlov berish uchun mo‘ljallangan va shuni aytib o‘tish lozimki, ma‘lumotlarga ishlov berishni qiyin algoritmlarini amalga oshirishda kam samaraga ega.

Mikrokontroller ob‘ekt bo‘yicha holat signallarini davriy so‘rab turadi va joylashtirilgan algoritmgaga muvofiq boshqaruv signallarini ketma-ketligini ishlab chiqaradi. Holat signallari boshqaruv signallarini joriy parametrlarini xarakterlaydi. Ular datchikning (D) chiqish signallarini analog-raqamli o‘zgartirgich (ARO‘) yoki signallar holatini shakillantirgich (SHSh) yordamida o‘zgartirish yo‘li bilan shakillanadi. Mikrokontroller orqali tanlangan Boshqaruv signallari, raqamli-analog o‘zgartirgich (RAO‘) yoki boshqaruv signallarini shakillantirgich (BSSh) yordamida o‘zgartiriladi. Bajaruvchi qurilmaga (BQ) keladigan RAO‘ va BSSh chiqish signallari mos ravishda analog va diskret boshqaruv ta‘siriga ega. Tizimida yana boshqaruv paneli, indikatsiya qurilmasi va tashqi qurilma bilan ma‘lumot almashtirib turish uchun interfeys bo‘lishi mumkin. Ma‘lum tizimning vazifasi va xarakteristikasi qarab yuqorida ko‘rsatitilgan elementlarning ba‘zi biri mavjud bo‘lmasligi mumkin.

Ishlab chiqariladigan mikrokontrollerlarning razryadiga qarab 4 dan 64 bitgacha bo‘ladi. Har xil dasturlarda ishlayotgan va arzon narxi bilan ajralib turadigan 8 razryadli mikrokontrollerlar dunyoda eng ko‘p tarqalganlar mikrokontrollerdir [11]. Atmel firmasining AVR oilasiga mansub mikrokontrollerlar shular sirasidandir.

AVR mikrokontrollerlari bu – 8 razryadli RISC-mikrokontrollerlaridir. Ajralib turuvchi xususiyatlardan biri FLASH dastur xotirasiga, ko‘p periferik qurilmalariga, yuqori hisoblash unimdorligiga, hamda dasturiy ta‘minotini ishlab chiqish uchun kerakli vositalarga egaligidir [20, 21].

Bugungi kunda AVR oilasini tarkibiga 50 dan oshiq har xil qurilmalar kiradi. Ular bir necha gruxlarga bo‘linadi [18, 19].

Universal AVR - mikrokontrollerlari Tiny AVR va Mega AVR gruhlariga kiradi. Tiny AVR (ATtinyXXX) – kam sonli tarmoqlanib chiqish joyiga ega, arzon qurilmadir. Mega AVR (ATmegaXXX) – katta xajmli xotira va tarmoqlanib chiqish joyiga ega, hamda periferik

qurilmalarning maksimal to‘plamiga ega bo‘lgan katta quvvatli AVR-mikrokontrolleridir.

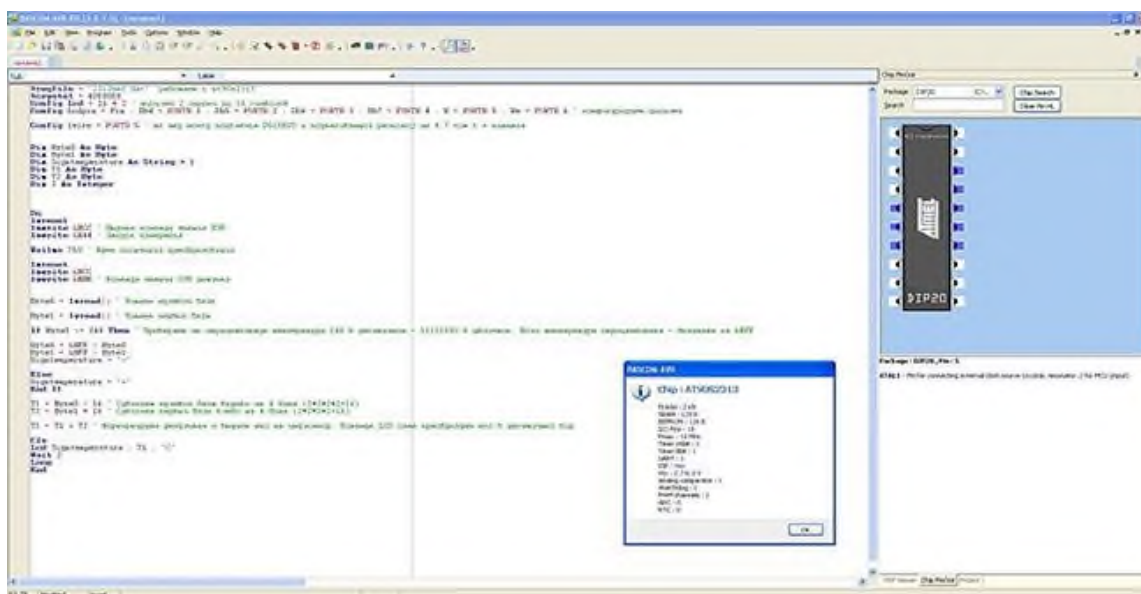
Maxsus AVR-mikrokontrollerlari LCD AVR (ATmega169, ATmega329) guruxga kiruvchi – suyuqkristalli indikatorlar bilan ishlash uchun mikrokontrollerlar;

- USB AVR (AT43USBXXX, AT76C711) – USB interfeysli mikrokontrollerlar; DVD AVR (AT78CXXX) – CD/DVD-uzatmali kontrollerlar;
- RF AVR (AT86RFXXX) – simsiz aloqa tizimlarini qurish uchun mikrokontrollerlar;
- Secure AVR (AT90SCXXXX, AT97SCXXXX) – smart-karta uchun mikrokontroller;
- FPGA AVR (AT94KXXAL) – bitta kristalda tuzilgan dasturlashtiriladigan mantiqiy integral sxemali AVR-mikrokontrollerlari.

4.2. BASCOM-AVR dasturiy muxitida loyixalashtirish

BASCOM-AVR dasturiy muhiti AVR kompaniyasiga tegishli Atmel mikrokontrollerlari dasturiy kodini yaratishga mo‘ljallangan.

8-bitli AVR mikrokontrollerlari RISC arxitekturasiga tegishli mikrokontrollerlar ish tezligi PIC mikrokontrollerlariga nisbatan 4 barobar tez ishlaydi. BASCOM-AVR muhiti dasturiy oynasidan tashqari vertikal ko‘rinishda va to‘g‘ridan to‘g‘ri mikrokontrollerga yuklash muhitlariga ega.



4.1– rasm. BASCOM-AVR muhiti

BASCOM-AVR juda ham foydalanishga oddiy muhit hisoblanadi. Dasturiy paket AVR chiplari asosiy funksiyalarini jamlagan bo‘lib, ulardan hisoblagichlar va taymerlar, analog-raqamli o‘zgartirgichlar va tashqi qurilmalardan tugmachalar, datchiklar, grafik indikatorlar, klaviatura v.b. BASCOM-AVR muhitida yozilgan dasturlar Proteus va maket platalarida sinab ko‘rish mumkin.

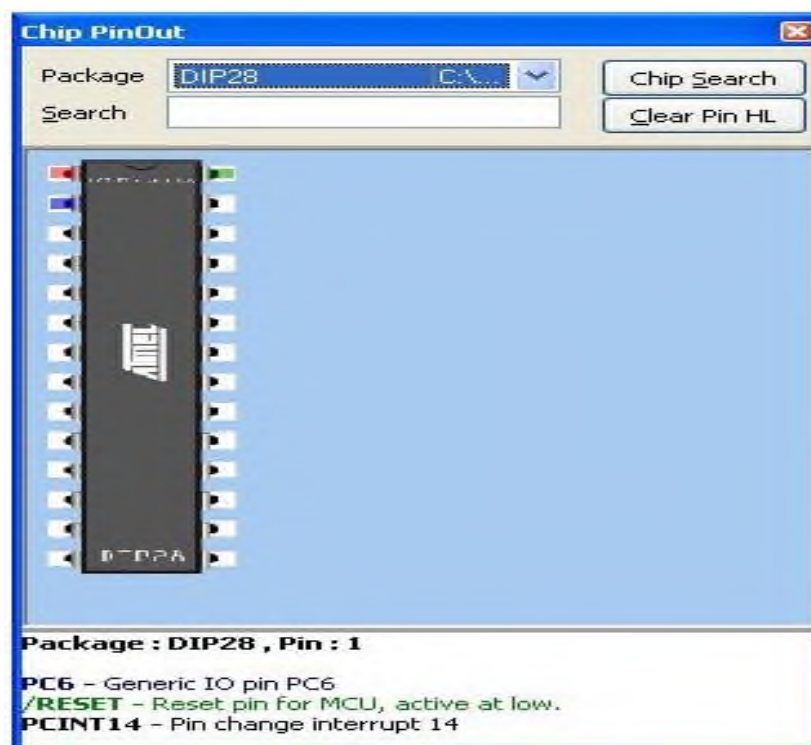
Kiritish chiqarish portlarini nazorat oynasi. Bu yerda aktiv chip REGFILE bilan aniqlanadi.

Sichqonchani oyoqchalari ustiga qo‘yilsa u qizil ranga bo‘yalganini ko‘ramiz. Oynaning pastki qismida uning izoxi ko‘rsatiladi va har bir qator turli rangda aks ettiriladi. Bu esa o‘z navbatida aloqa bir qancha alternativ funksiyalarga ega ekanligini ko‘rsatadi.

Ko‘rsatilgan ko‘k rangli funksiya kiritish chiqarish umumiy chiqishi hisoblanadi.

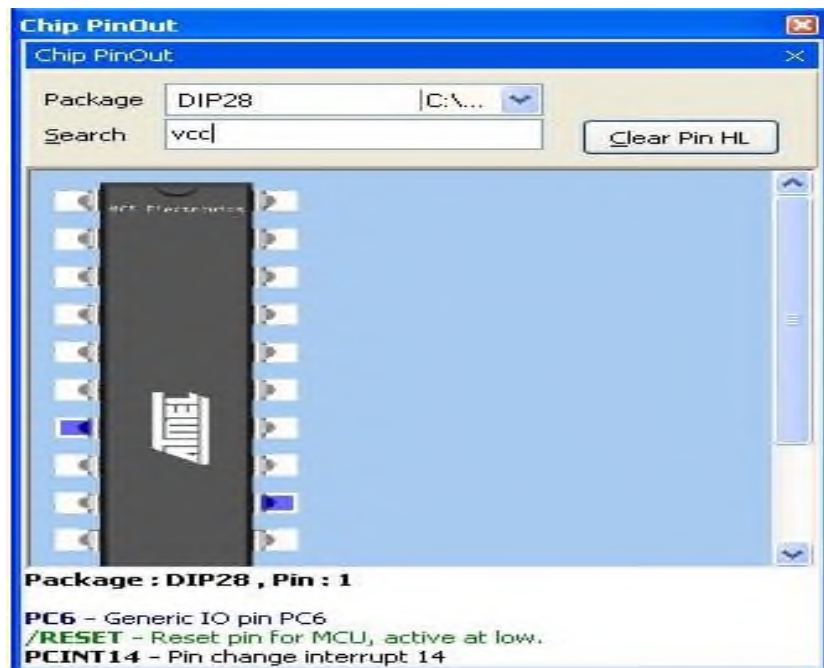
Ko‘rsatilgan yashil rangli funksiya – bu PIN-kod RESET

Ko‘rsatilgan qora rangli funksiya - (прерывание смены PIN) kodni almashishi va uzilishlari.



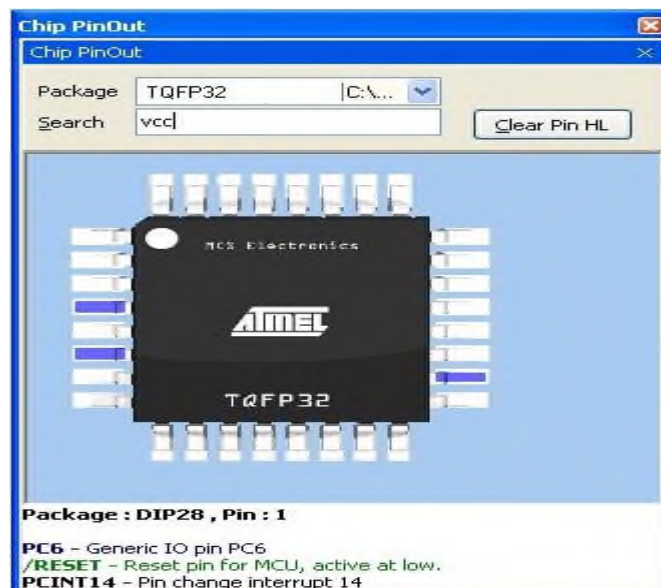
4.2– rasm. Mikrokontroller tanlash

Chiqishlar bitta yoki bir nechta funksiyalarga ega bo‘lishi mumkin, lekin ba’zi bir funksiyalar bir shalikda ishlatilishi mumkin.



4.3– rasm. Mikrokontroller pinlari

«Clear Pin HL» bosilishi bilan hamma ranglarni o‘chirish mumkin.



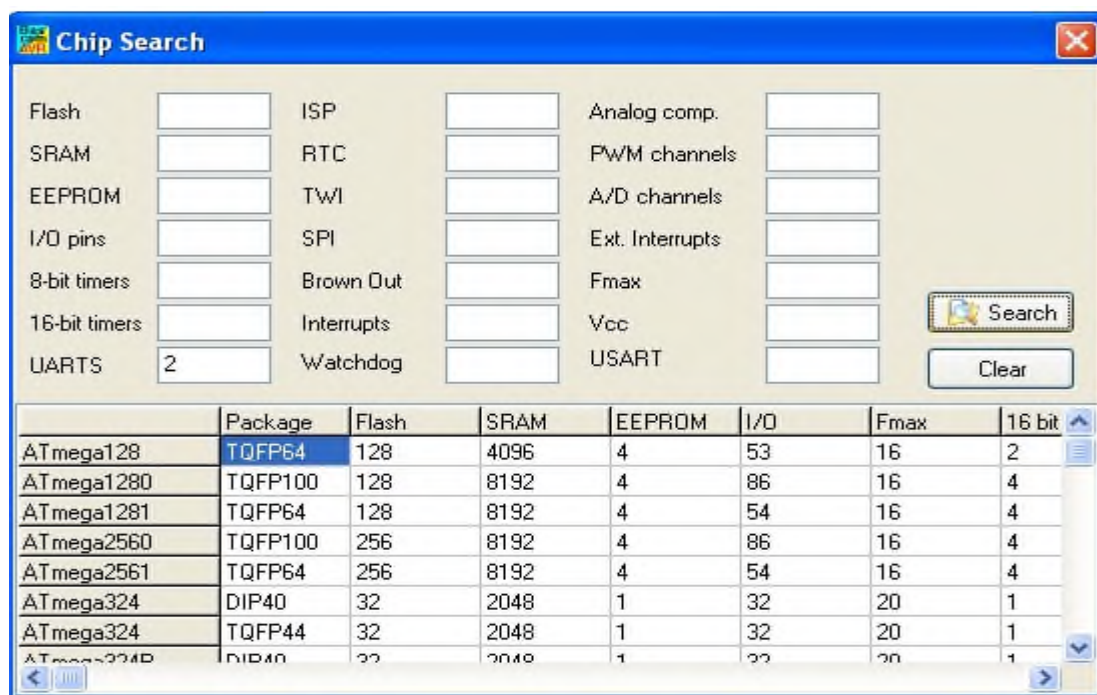
4.4– rasm. Chiqish ranglari

Paket almashtirish bilan chiqish ranglari o‘chirib tashlanadi. Kerakli oyoqchalarga ikki marta bosilishi bilan yashil ranga bo‘yaladi. Aks xolda qizil, ko‘k ranga bo‘yaladi.

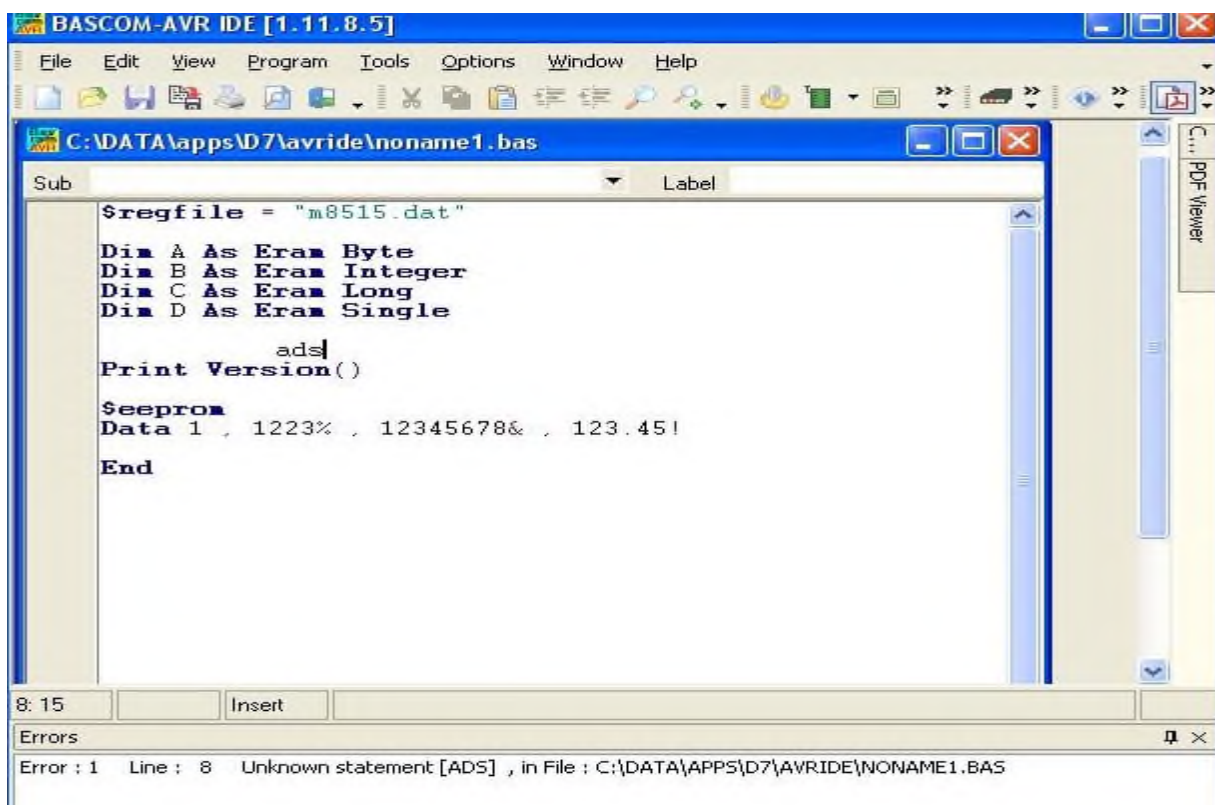


4.5– rasm. ATmega88 qurilmalari

Chipni tanlash uchun, “Chip Search” tugmachasini bosish kerak va uning natijasida quyidagi oyna paydo bo‘ladi:

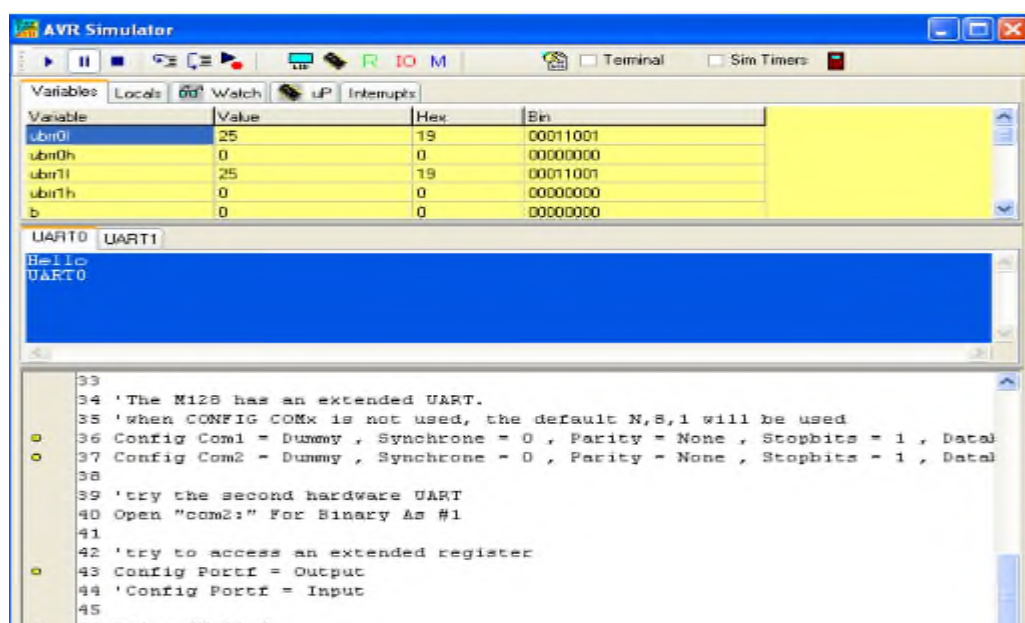


4.6– rasm. Chipni tanlash



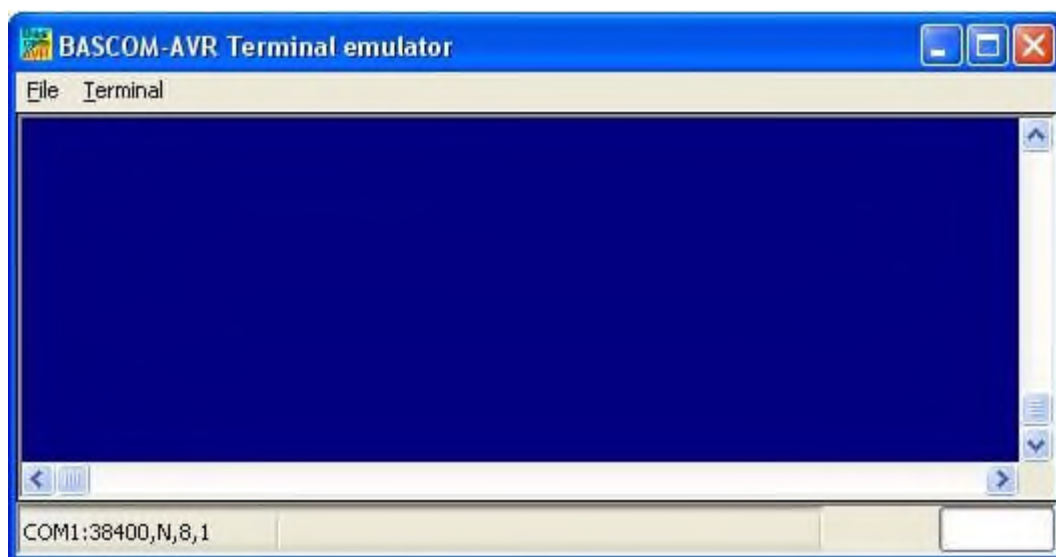
4.7– rasm. Xatoliklar oynasi

Keltirilgan operatsiyadan dastur kompilatsiyasi natijasida sintaksis xatolikga teshiradi va xatoliklar qatori ham ko‘rsatib beriladi. Yaratilgan dasturni modellashtirish kerak. Dasturni AVR Studio yoki boshqa dasturiy muxitda simulyatsiyalash mumkin.



4.8– rasm. Kiritish va chiqarish registrlari

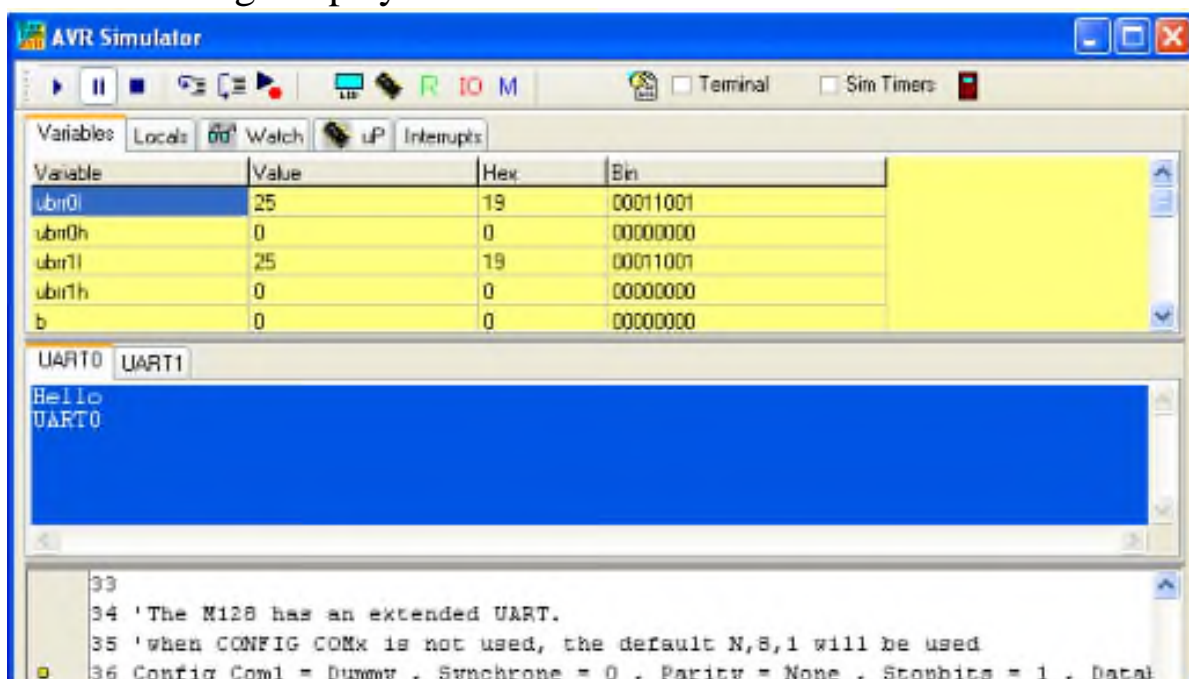
Tashqi qurilmalarni ulash (TERMINAL bo‘limi)



4.9– rasm. Terminal bo‘limi

Dasturda PRINT dan foydalanishda oynada chiqish ma’lumotlari ko‘rinadi.

Dasturdagi INPUT oynada fokusni joylashtirib, qiymatni chiqarish kerak. Keltirilgan opsiya asosida dasturni modellashtirish mumkin.



4.10– rasm. Modellashtirish oynasi

F2 tugmachasini bosilishi bilan TAB Options Simulator o‘rnatilgan simulyatordan foydalaniladi.

Fayl OBJ – bu AVR Studio simulyatoridagi ob’ektidir.

DBG-fayl o‘zgaruvchilar va boshqa ma’lumotlar to‘g‘risidagi ma’lumotni saqlaydi. Mikrokontroller registrlar blokini ko‘rish

Reg	Val
R21	00
R22	08
R23	00
R24	01
R25	00
R26	29
R27	01
R28	80
R29	04
R30	50
R31	07

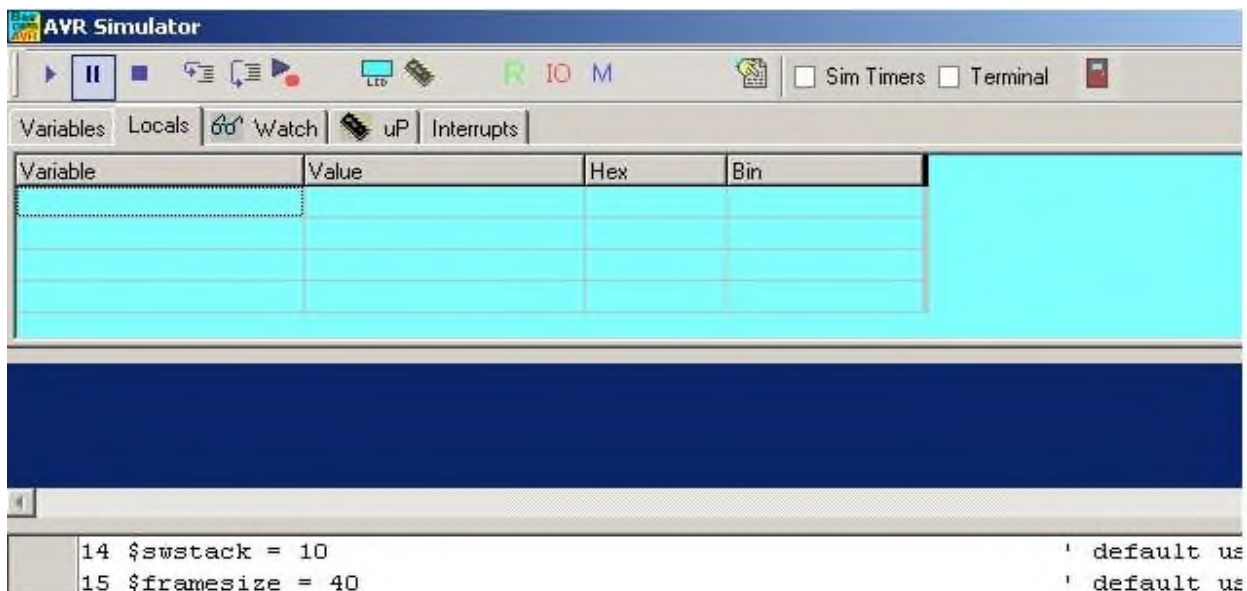
4.11– rasm. Mikrokontroller registrlar bloki

Mikrokontroller xotirasini ko‘rish uchun quyidagi bo‘limdan kirish mumkin.

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
0100	00	00	00	00	00	00	00	00	00	00	00	00	00	08	00	00
0110	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0120	00	00	00	00	00	00	00	00	01	00	63	63	00	00	00	00
0130	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0140	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0150	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0160	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0170	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

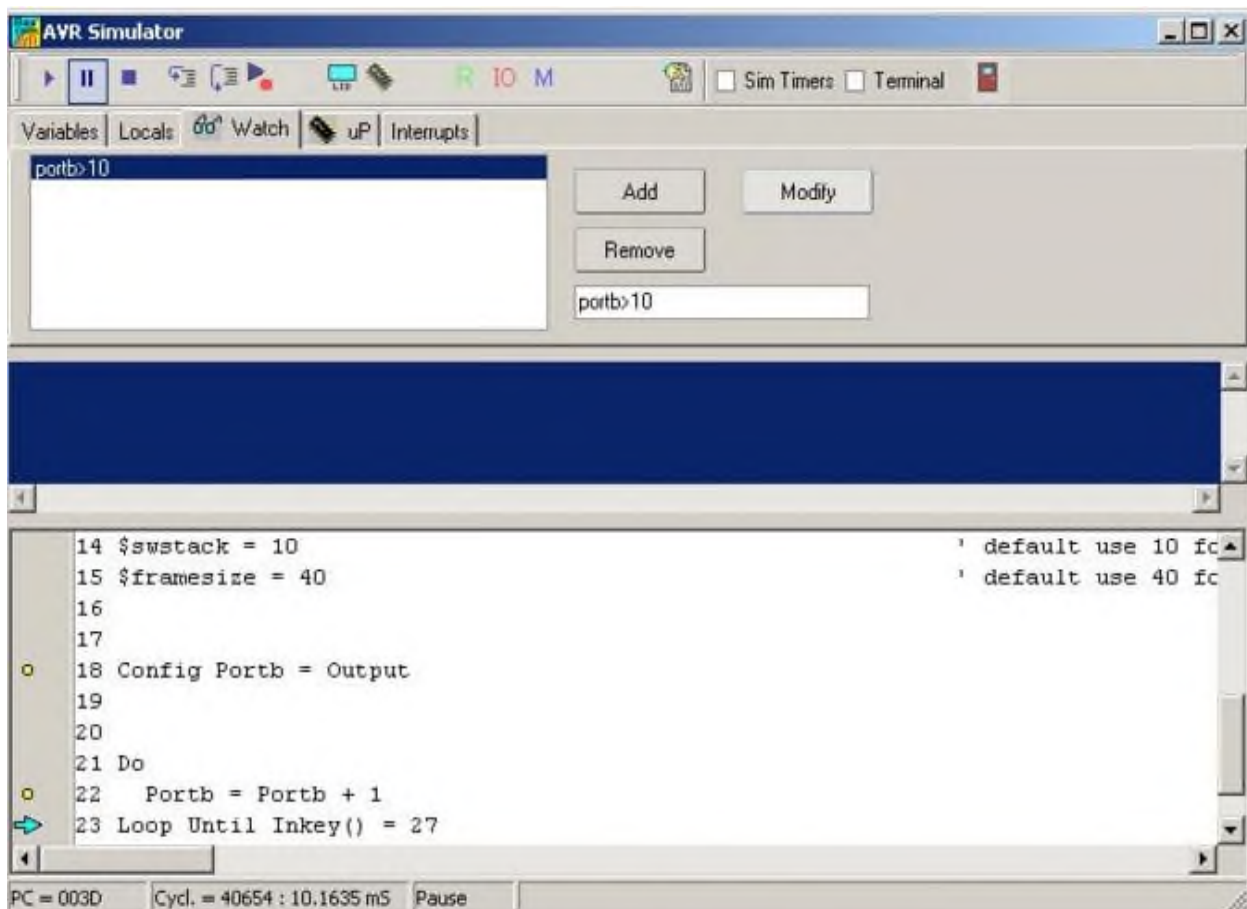
4.12– rasm. EEPROM xotirasi

O‘zgaruvchilarni qo‘shish va o‘zgartirish uchun Variable-column bo‘limidan foydalaniladi, unda o‘zgaruvchilar ro‘yxati ko‘rinadi va kerakli o‘zgaruvchini ajratish mumkin. Massiv o‘zgaruvchilarni ko‘rish uchun indeksli o‘zgaruvchi nomini kiritish kerak. Modellashtirish jarayonida Value, Hex-column yoki Bin-column bo‘limidan o‘zgaruvchilar qiymatini o‘zgartirish mumkin.



4.13– rasm. O‘zgaruvchilar qiymatini kiritish

Modellashtirish jarayonida Watch-TAB tushuntirishlarni kiritish imkoniyatini yaratadi.



4.14– rasm. Watch-TAB oynasi

TAB (SREG) mikroprotssessor status registri qiymatini ko'rsatadi. Dasturlash jarayonida bayroqlarni o'zgartirish va qo'shimchalar kiritish mumkin. Dasturni modellashtirish jarayonida dastur steki, apparat steki, modellashtirish jarayonida minimal va maksimal qiymatlar aks etadi.

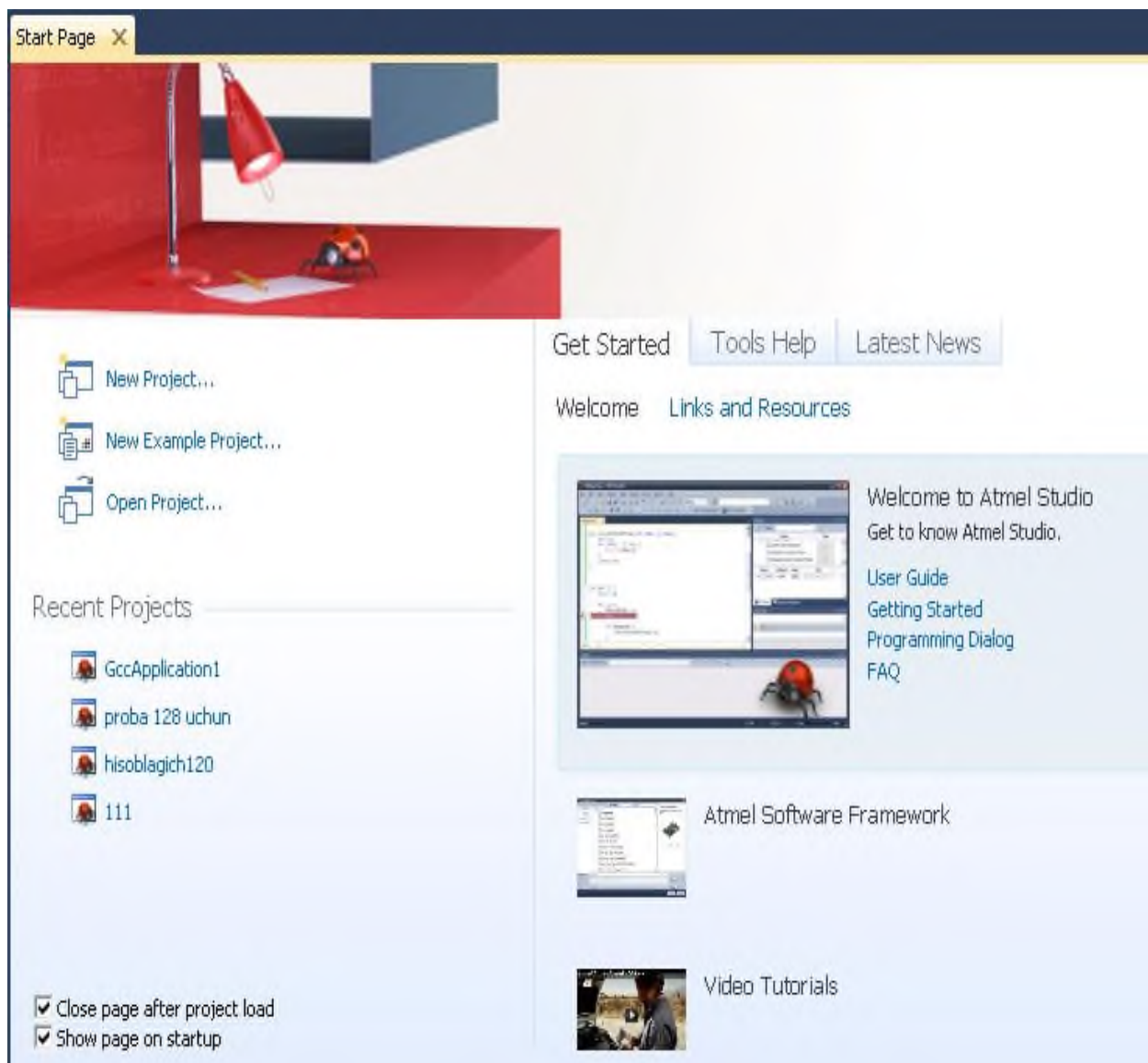
4.3. ATMEL STUDIO 6 dasturiy muxitida loyihalashtirish

Atmel Studio 6 –ARM va AVR mikrokontrollarni dasturlash uchun yagona muhit hisoblanib ularni dasturlash jarayonini bir qancha osonlashtiradi. Atmel Studio 6 –ARM Cortex –M va 8/32 –razryadli mikrokontrollerlarda qo'llanilayapti ushbu muhitning afzalligi bir muhitda ishlab chiqarishning mavjudligidadir



4.15– rasm. Atmel Studio 6 muhitining umumiy ko'rinishi.

Dasturga kirish uchun quyidagi bosqichlar bajariladi.



4.16– rasm. Dastur xosil qilish oynasi

Ushbu oynada quyidagi operatsiyalarni bajarish mumkin:

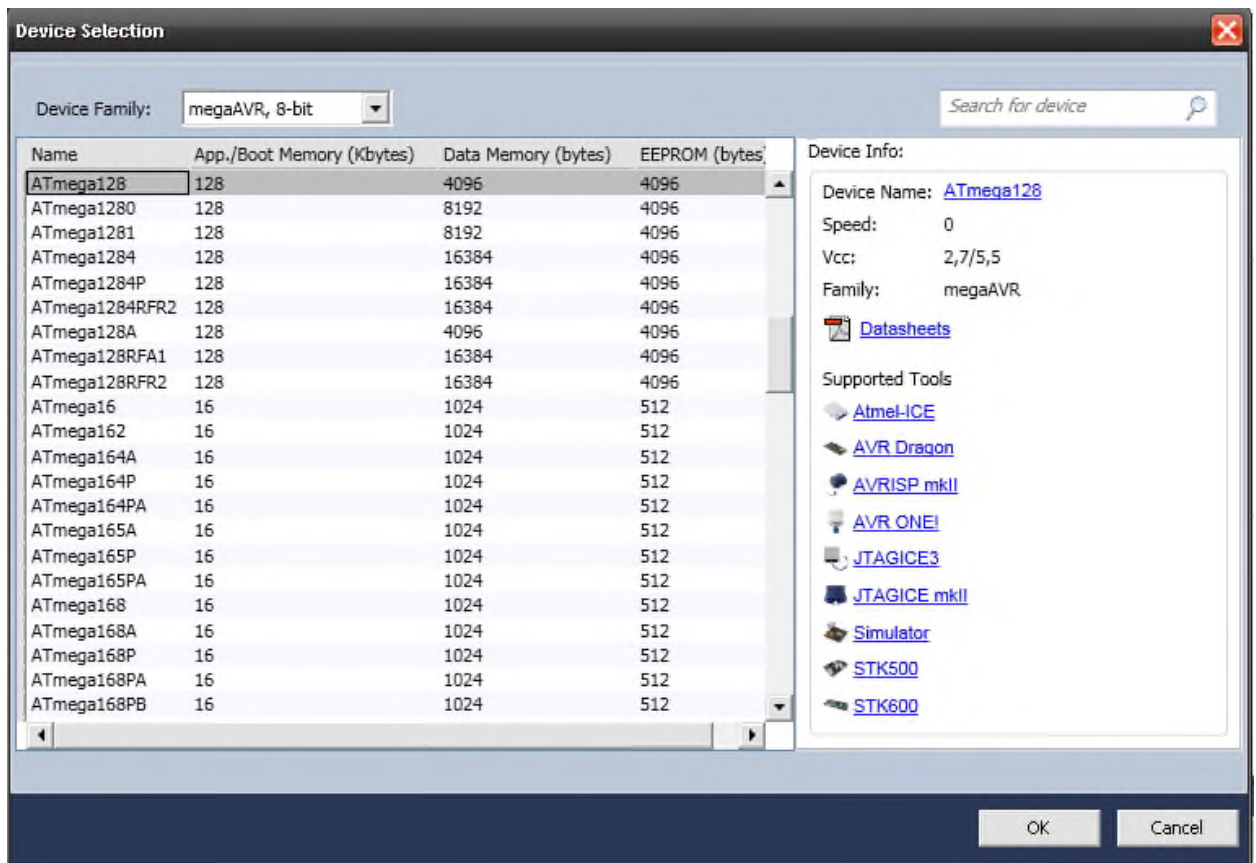
- yangi loyihani tashkil qilish - tugmacha Create New Project;
- faylni ochish .cof yoki .hex - knopka Open;
- ro‘yxatda keltirilgan fayllarni ochish.

Yangi loyihani tashkil qilish. Buning uchun Create New Project (4.17 -rasm)

AVR Simulator va mikrokontroller Atmega128.

Finish tugmachasini bosamiz.

Hamma operatsiyalar menyu bo‘limidan File va Project, MK emulyatori menyu Debug ► Select. Platform and Device.



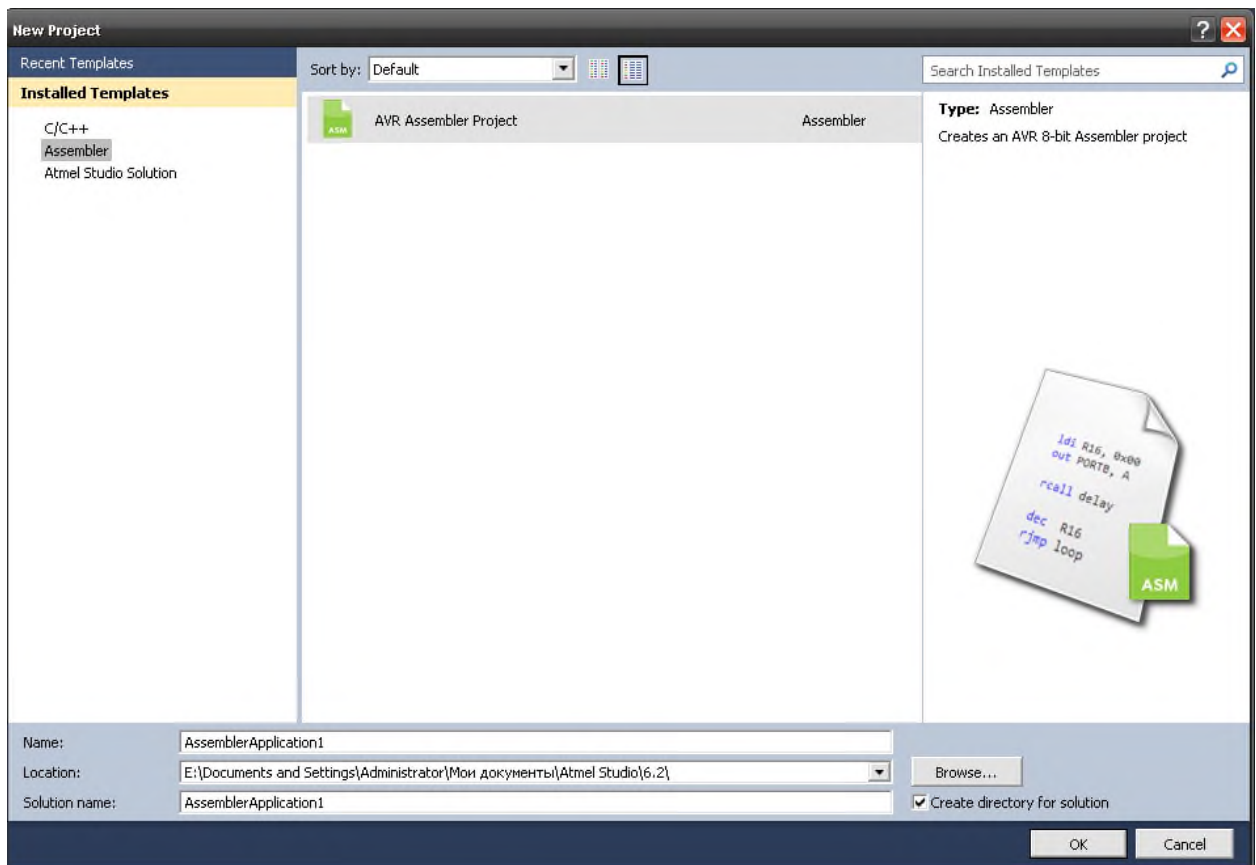
4.17– rasm. Mikrokontroller turini tanlash

Yangi loyiha xosil qilish. Yangi loyiha xosil qilish uchun Project→New Project menyusidan foydalaniladi. Atmel AVR Assembler bo‘limini tanlab loyiha nomini kiritamiz. Masalan Project Name (my_prob_01). Avtomatik xolatda kiritish maydoni to‘ldiriladi Initial file. Location maydonidan D diskdagi yo‘lini ko‘rsatish kerak. Dasturlash platformasini tanlash kerak Debug platform = AVR Simulator;

Dastur strukturasi. Ishning davomi dastur nomi bilan mos bo‘lgan oynada davom etadi. Oynaning asosiy modulida assembler dasturiga mos dastur matni kiritiladi:

Kommentariya – qator bo‘lib (yoki qator qismi), “nuqta vergul bilan boshlanadi - ;”

Derektivalar “tochka” bilan boshlanadi. Shuni ta’kidlash lozimki, assembler qurilmalar aro ma’lumot almashish va shartli, shartsiz o‘tish operatsiyalarini bajaradi. Shuning uchun dasturlash jarayonlarida har bir komandalarga izox beriladi.



4.17– rasm. Dasturlash tilini tanlash

Asosiy modulning oynasiga quyidagi matnni kiritish kerak;
mikrokontroller AT90S2313 seriyasida foydalanish

```
.include "2313def.inc" ; faylni biriktirish
.device AT90S2313; MS uchun komandani tanlash
;registrlar REG shartli nomini kiritish
.def data=r16
.def data1=r17
;o‘zgaruvchilar shartli nomini kiritish SRAM
.dseg
code: .byte 1 ;bir baytli o‘zgaruvchilar
keys: .byte 1 ;tugmacha xolati o‘zgaruvchilari
keys1: .byte 1 ; tugmacha xolati o‘zgaruvchilari
key_up: .byte 1; tugmacha oraliq xolati o‘zgaruvchilari
leds: .byte 1 ; svetodiodlar xolati o‘zgaruvchilari;
;boshlanish flash
.cseg
.org 0 ;qayta yuklash vektori
rjmp ini
;dastur boshlanishi
```

```

ini:
; stekni initsializatsiyalash
ldi data,low(RAMEND)
out spl,data
; ldi data,high(RAMEND)
; out sph,data
;.include "ini_device.asm" ;MK ni o'rnatish
start:
rjmp start
;
; yordamchi modul qism dasturi
.include "delays.asm"
;*****
.exit

```

Mikrokontrollerlarni dasturlashning afzalligi, dastur universal xarakterga ega bo'lmasdan dastur aniq qurilma uchun tuziladi. Shuning uchun direktivalar (.include) mikrokontroller registrlari nomlari bilan ko'rsatilishi kerak (.device). Tizim o'zgaruvchilari fayllarni avtomatik topish uchun Project→Assembler Options→Additional include file yo'nalishi ko'rsatilishi kerak:

C:/Programm Files/Atmel/AVR
Tools/AvrAssembler2/Appnotes/Direktiva (.def) yordamida dasturchi yuqori operativ xotira registrlariga nom berishi mumkin va uning natijasida protsessor xolatlarini aniqlanadi. Barcha dasturiy tillar umumiy yechimga ega bo'lishi mumkin.

Misol uchun:

```

.include "2313def.inc" ;tizim o'zgaruvchilarini biriktirish
.device AT90S2313 ;MS komandalar tizimini aniqlash
konstanta (direktiva .equ).
.equ memo=$378

```

Registrlar bo'limi.

;Registrlar REG ga shartli nom berish

```
.def data=r16
```

```
.def data1=r17
```

O'zgaruvchilar bo'limi

```
.dseg
```

```
code: .byte 1 ;bir baytli o'zgaruvchi
```

```
keys: .byte 1 ;tugmachalar xolati o'zgaruvchilari
```

keys1: .byte 1 ; tugmachalar xolati o'zgaruvchilari
key_up: .byte 1; tugmachalar xolati oraliq o'zgaruvchilari
leds: .byte 1 ;svetodiodlar xolati o'zgaruvchilari

Dastur bo'limi (flash-xotira).

O'z navbatida quyidagi bo'limlarga bo'linadi; uzilishlar, stekni tashkil qilish, mikrokontrollerlarni initsializatsiyalash.

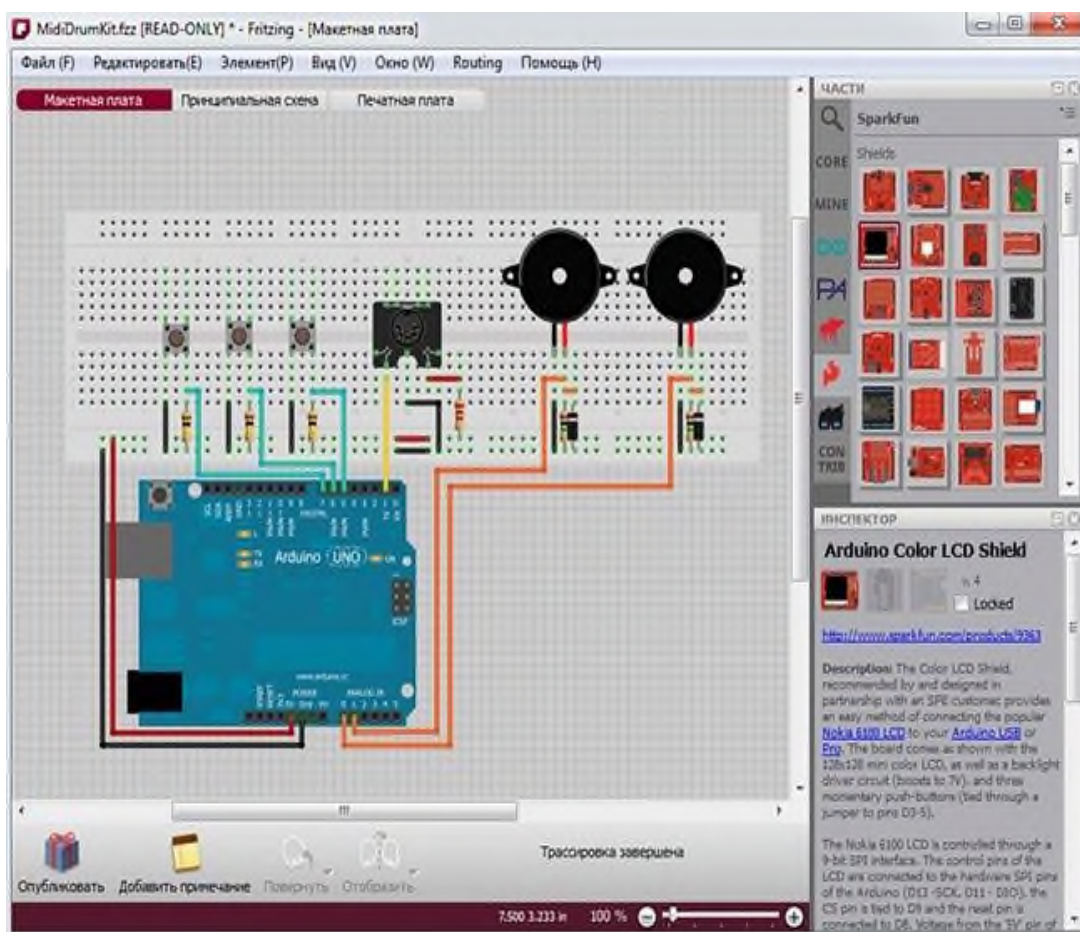
```
.cseg  
.org 0 ; qayta yuklash vektori  
rjmp ini  
;dastur boshlanishi  
ini:  
; stekni initsializatsiyalash  
ldi data, low(RAMEND)  
out spl, data  
; ldi data, high(RAMEND)  
; out sph, data  
;.include "ini_device.asm" ;MK o'rnatish  
start:  
rjmp start  
;  
.include "delays.asm"  
Tugatish  
.exit
```

4.4. Fritzing dasturiy vositalarida sozlash

Fritzing dasturiy to'plami platada sxemalarni sinashda, prinsipial sxemalarni maketli plata ko'rinishga o'tkazishda ishlatiladi.

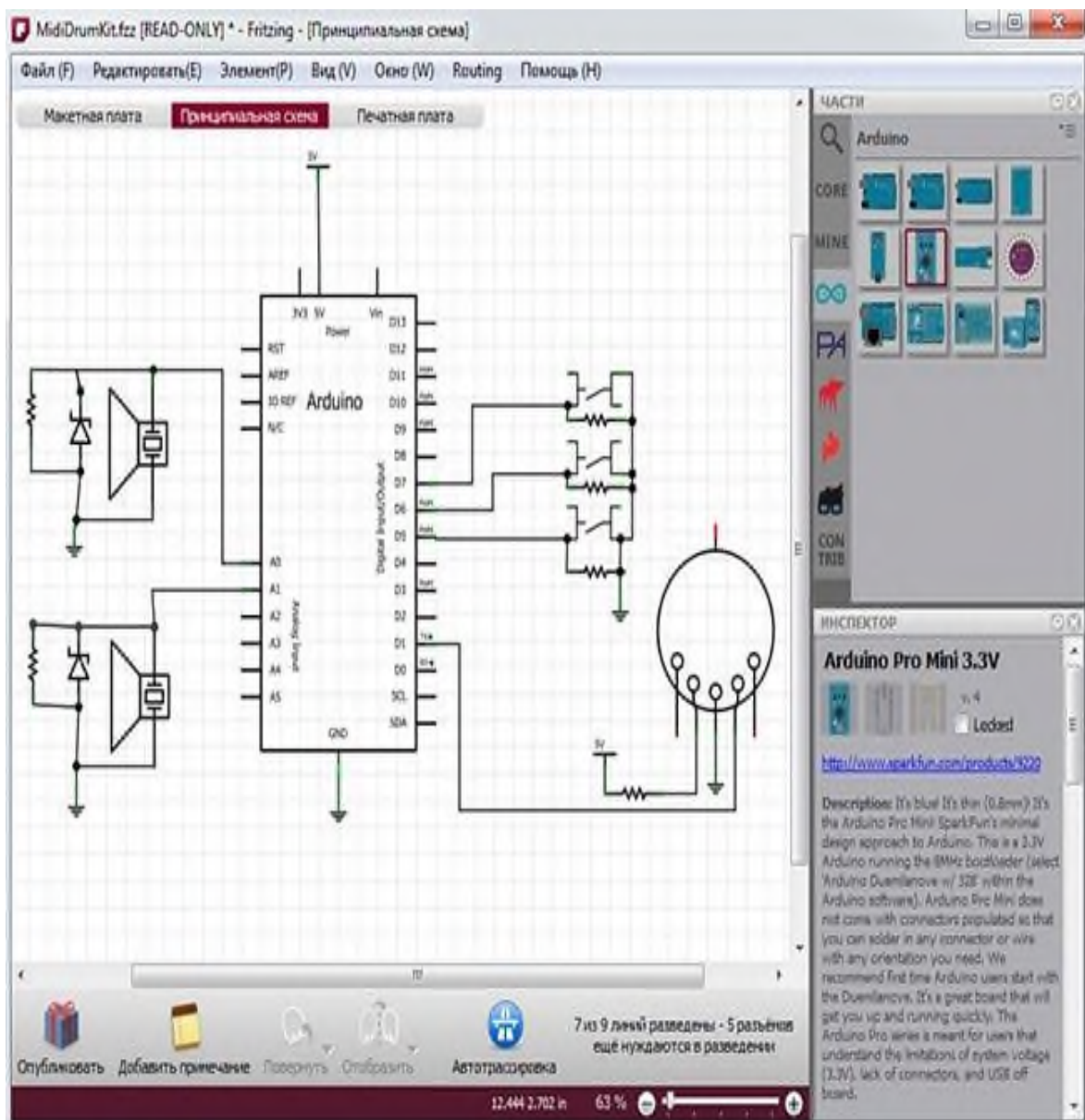
Dasturdan foydalanuvchilar asosan -izlanuvchilar, dizaynerlar, radioxavaskorlar v.b. Fritzing dasturiy to'plami Arduino apparat-dasturiy platformalari uchun yaratilgan bo'lib, uning tarkibi mikrokontroller Atmel AVR, dasturlash uchun radiojixozlar, aloqa interfeysi, dasturiy muhit Processing/Wiring.

Fritzing dasturiy to'plamidagi ish jarayoni tayyor komponentlarni tanlashdan boshlanadi. Komponentlar muxitning ish oynasida ko'rsatilgan. Bu yerda turli xil maketli va montajli platalarni uchratish mumkin (Arduino). Bu yerda analog va raqamli mikrosxemalar, ixtiyoriy radiojixozlar: kondensatorlar, tranzistorlar, rezistorlar, svetodiodlar, batareyka, tugmachalar.



4.18- rasm. Fritzing dasturiy to‘plami oynasi

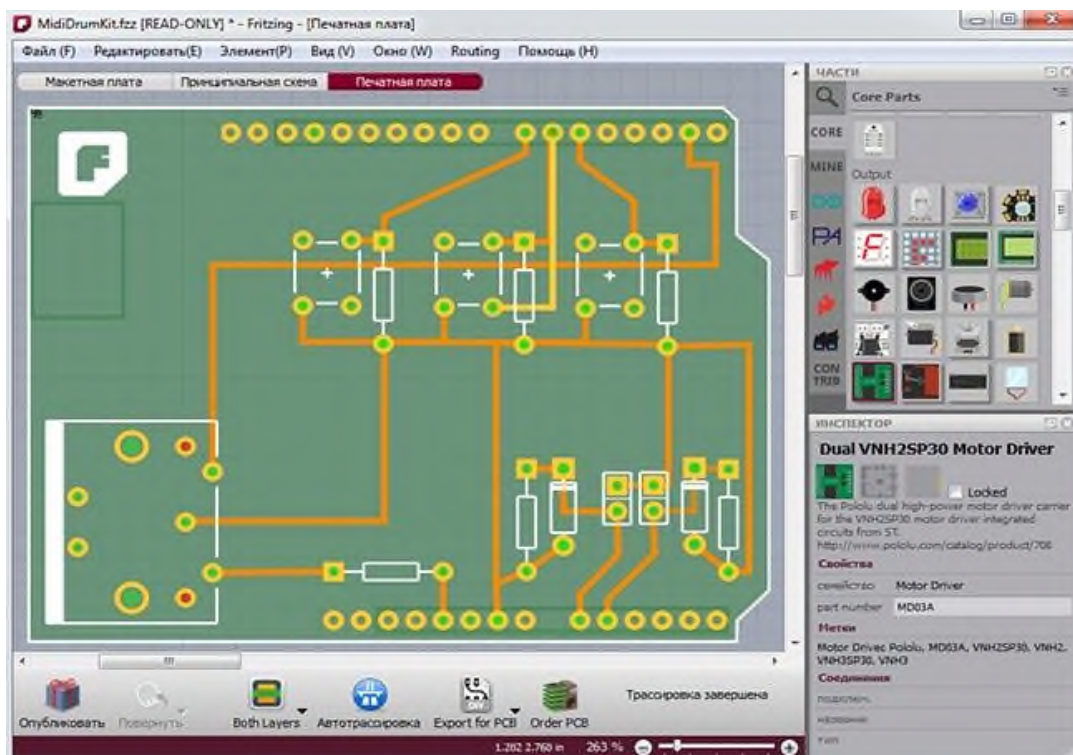
Robototexnika uchun juda ham katta miqdorda qurilmalar mavjud: motorlar, dalnomerlar, dinamiklar, servoprivod, qadamli dvigatellar, LCD va raqamli indikatorlar v.b. O‘z shaxsiy elementlarni xosil qilish va mavjud bazani yangilash imkoniyati mavjud. “Maketnaya plata” oynasida va “Принципиальная схема” oynasida har qanday rasmni ko‘chirish imkoniyati mavjud. Ish natijasini plata ko‘rinishida chop etish va u asosida har qanday qurilmani yaratish imkoniyati mavjud. Fritzing dasturi bu dizaynerlar, rassomlar, tadqiqotchilar va sevimli mashg‘ulotchilarni interfaol elektronika bilan ijodiy ishlash va elektron loyihalarni ishlab chiqishda qo‘llab-quvvatlash uchun ochiq manbali qiziqarli tashabbusdir. Fritzing sizga elektron aylanishlar haqida ko‘proq ma’lumot olishga, loyihalaringizni hujjatlashtirishga va hatto ularni ishlab chiqarishga tayyorlashga yordam beradi.



4.19- rasm. Qurilmaning sxema ko‘rinishi

Yuqorida keltirilgan imkoniyatlardan tashqari Fritzing mustaqil ravishda plata ko‘rinishiga olib kela olmaydi va simulyatsiya qila olmaydi. Rasmlar juda ham sodda va element bazasi kam.

Fritzing dasturiy to‘plami 2009 yilda Potsdam universitetida yaratilgan. “From prototype to product”. Fritzing oddiy ish jarayonini tezlashtiradi va qulay interfeysga ega. Radiotexnika dasturlari havaskor radio ixlosmandlari uchun foydalidir. Va bu ajablanarli emas. Radiotexnika inshootlarini simulyatsiya qilish uchun ushbu dasturga ehtiyoj bor.



4.20- rasm. Qurilmaning plata ko‘rinishi

Uning atrof muhiti quyidagilarni jamlagan: Project View- virtual elektron sxema qurilgan.

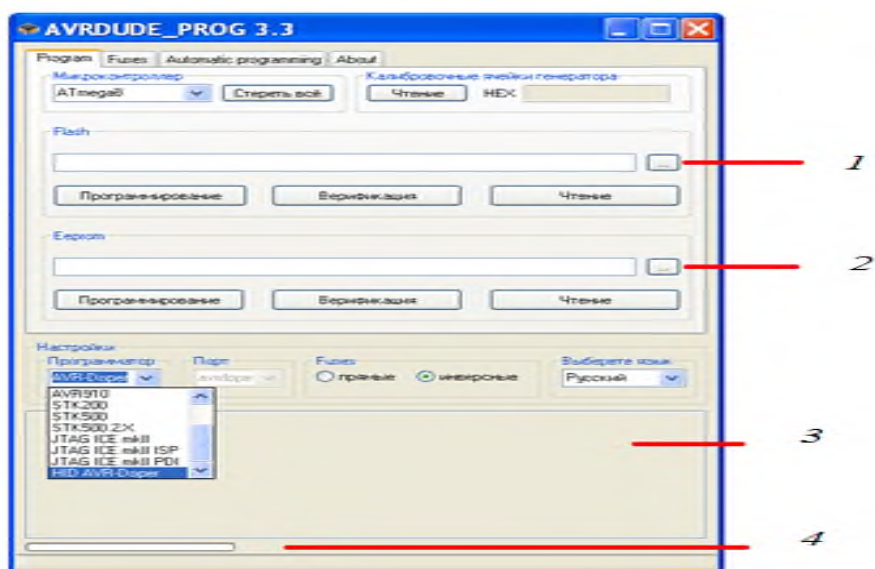
Windows uchun palitra- kutubxona jixozlarini birlashtirish Fritzing dasturiy to‘plami ko‘rinishlari foydalanuvchi tomonidan o‘zgartirilishi mumkin.



4.21 - rasm. Fritzing dasturiy to‘plami

4.5. AVRDUDE dasturiy vositalarida sozlash

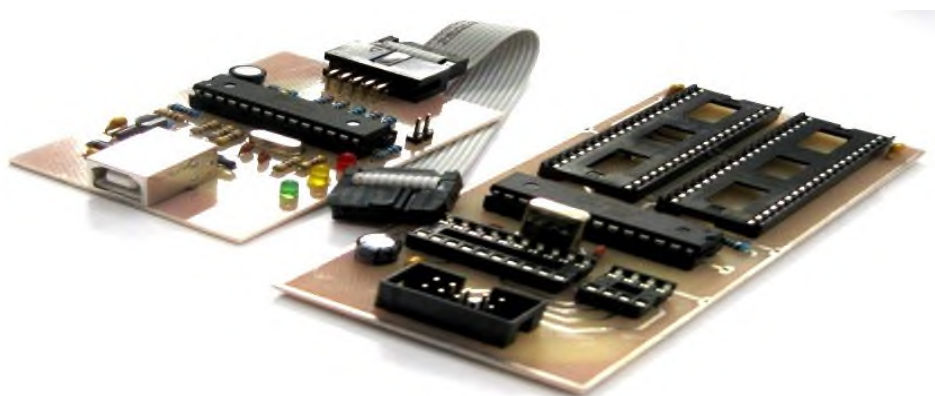
Dasturlarni mikrokontrollerga yuklash uchun AVR DUDE dasturiy muhitidan foydalanildi.



4.22 - rasm. AVR DUDE oynasi

- 1- Mikrokontroller flash xotirasiga yozish uchun hex fayl tanlash
- 2- EEPROM xotirasiga yozishni tanlash
- 3- Konsol oynasi
- 4- Dasturlash jarayoni

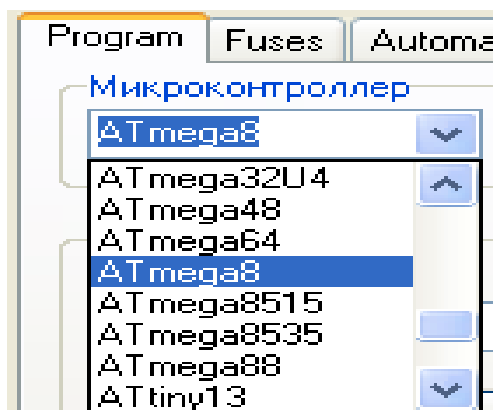
Keltirilgan dasturlash jarayonini amalga oshirish uchun maxsus programmatorlardan foydalaniladi.



4.23- rasm. Programmatorlar

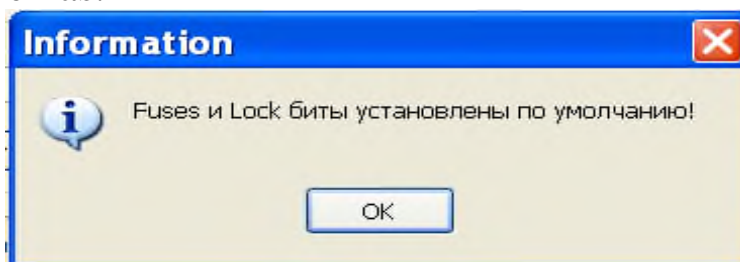
Mikrokontroller mikrosxemasini ulash uchun quyidagi ketma-ketlikni bajarish kerak.

1. “Mikrokontroller” turini tanlash oynasidan kerakli mikrokontrlrlerni tanlanadi.



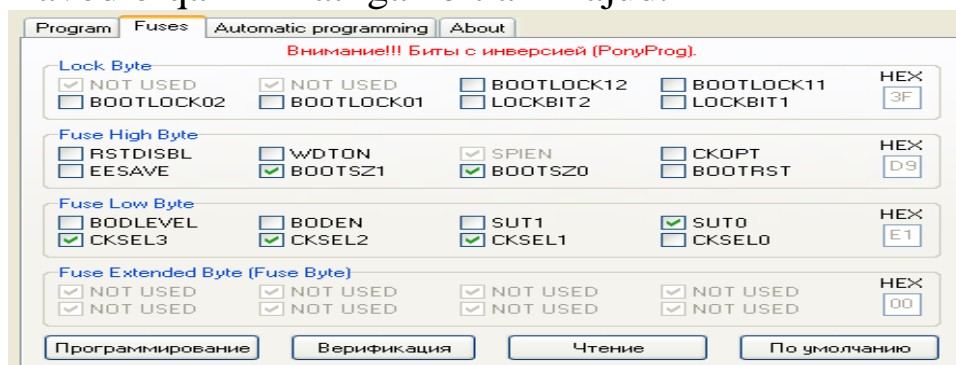
4.24 - rasm. Mikrokontroller turini tanlash

2. “Fuses” va Lock bitlari o‘rnatilgan bo‘lib, ularni o‘zgartirish zarur emas.



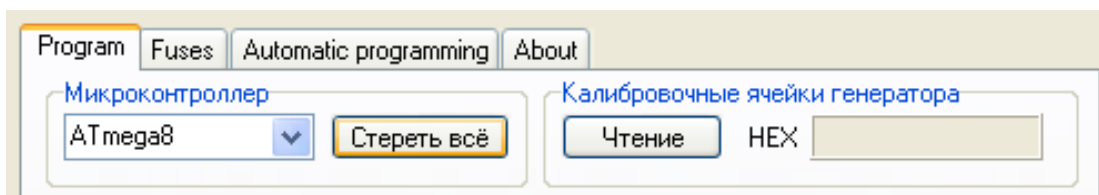
4.25 - rasm. Fuses bitlar

3. “Fuses” oynasida mikrokontrollerlarning ishlab chiqargan zavod orqali o‘rnatilgan bitlari majud.



4.26- rasm. Fuses bitlarni o‘rnatish

Dasturni mikrokontrollerga yozish. Agar mikrokontroller xotirasida qandaydir ma’lumot mavjud bo‘lsa , avval uni o‘chirib tashlash kerak (“Стерет всё”)



4.27 - rasm. Mikrokontrollerga dasturlash oynasi

4.6. Code Vision AVR muxiti va uning imkoniyatlari

Mikrokontrollerlar uchun dastur yozishda ko‘pincha C va assambler dasturlash tilidan foydalaniladi. AVR arxitekturasiga doir bir necha C va assambler dasturlash tilini o‘ziga qamrab olgan bir necha dasturlash muhirlari mavjud: CodeVisionAVR, ImageSraft C, MicroC, AVR bascom, MicroPascal Atmel Studio va boshqalar.

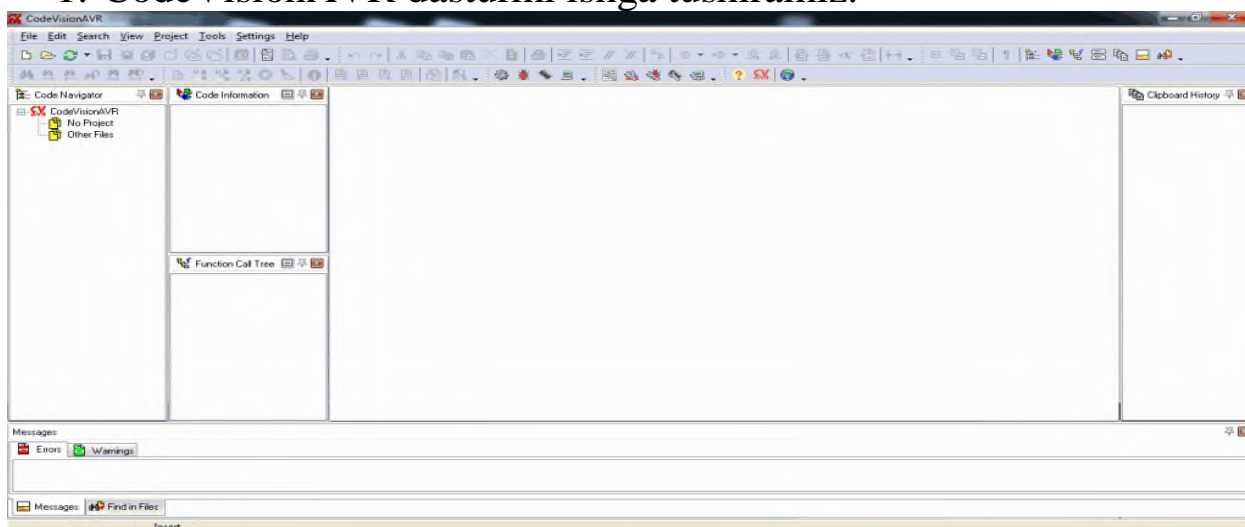
Bundan tashqari, bir qancha kompilyatorlar manba kodni kompilyatsiya keyin mikrokontroller dasturlanishini qo‘llab-quvvatlaydi.

CodeVisionAVR dasturi HP InfoTech kompaniyasi tomonidan yaratilgan dasturiy muhit bo‘lib, C dasturiy tilini o‘zida mujassam etgan.

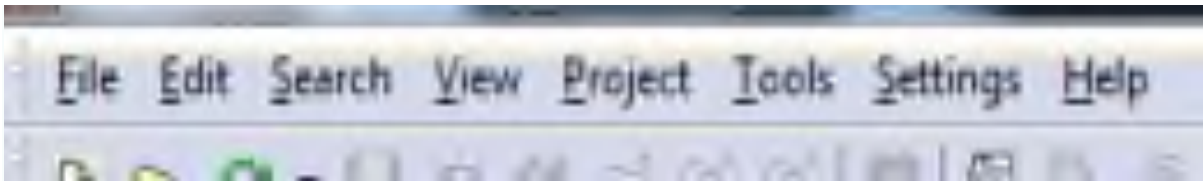
Asosiy afzalliklaridan CodeVisionAVR dasturiy muhiti orqali talaba o‘z bilimini rivojlantirish uchun juda qiyin emas. Muharriri bir vaqtning o‘zida ikkita loyihalar bilan ishlash imkonini beradi va ishlash natijalarini saqlab avtomatik vaqtni belgilash uchun imkon beradi.

CodeVisionAVR dasturi muhitida projektini ishga tushirishni ko‘rib chiqamiz.

1. CodeVisionAVR dasturini ishga tushiramiz.



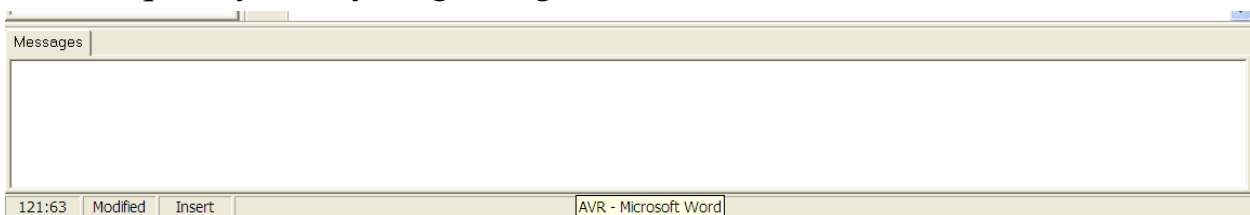
2. Menyular qatori.



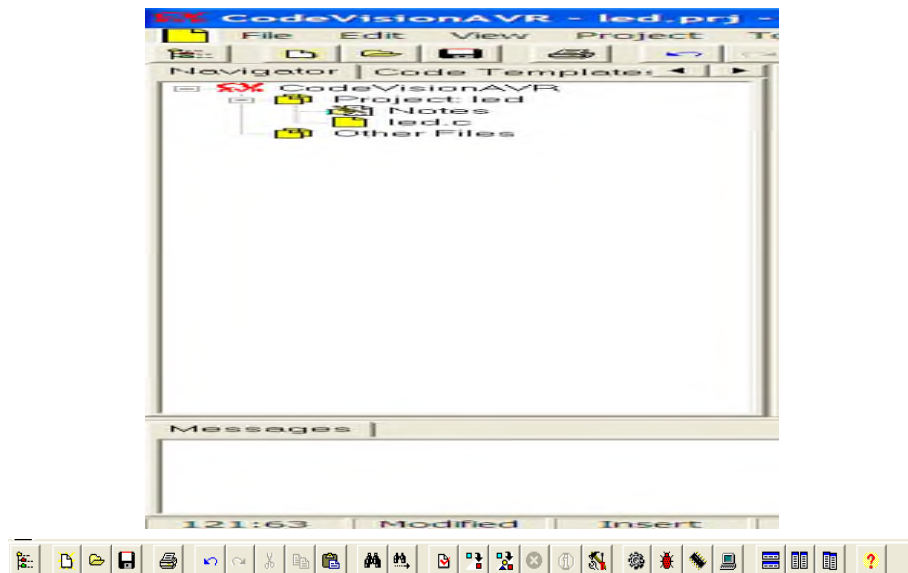
3. Kod kiritiladigan qismi.

```
96 TCCR2=0x00;
97 TCNT2=0x00;
98 OCR2=0x00;
99
100 // External Interrupt(s) initialization
101 // INT0: Off
102 // INT1: Off
103 // INT2: Off
104 MCUCR=0x00;
105 MCUCSR=0x00;
106
107 // Timer(s)/Counter(s) Interrupt(s) initialization
108 TIMSK=0x00;
109
110 // Analog Comparator initialization
111 // Analog Comparator: Off
112 // Analog Comparator Input Capture by Timer/Counter 1: Off
113 ACSR=0x80;
114 SFIOR=0x00;
115
116 while (1)
117 {
118     PORTA=0x01;
119     delay_ms(10);
120     PORTA=0x00;
121     delay_ms(10);
122     // Place your code here
123
124 };
125 }
126
```









4. Ushbu panelda xatolar va ogohlantirishlar ask ettiriladi va qayerda qanday xato yuzaga kelganini ko'rsatadi.



7. Ushbu panelni vazifasi shundan iboratki tuzilgan proyekt nimalardan iboratligini va ularga qanday kirishni ko'rsatadi.



Ushbu panel quyidagilardan iborat:

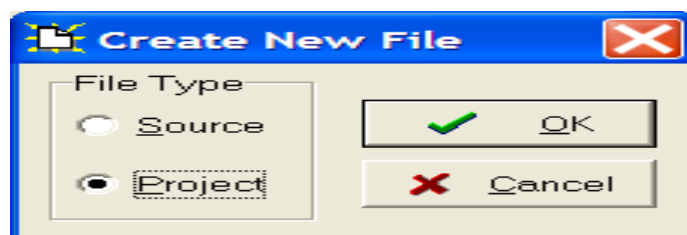
- a. Yangi sahifa ochish. 
- b. Kerakli fayllarni ochish. 
- c. Saqlash. 
- d. Chop etish. 
- e. Hex faylni hosil qilish va kompilyatsiya qilish. 
- f. Tuzatish va o'zgartirishlarni kiritish. 
- g. Kod generatsiya qilish. 
- h. Chipga dastur yozish. 

CodeVisionAVR qismi bo'lgan asosiy modullar quyidagilardan iborat:

- Periferik qurilmalar boshlash uchun elementlari;
- tashqi qurilmalar komponentlar bilan muloqot;
- boshlang'ich kod muharriri;

Misol tariqasida oddiy dasturlash jarayonlari ko'rsatib beriladi.

Sodda dastur nazarida biz oddiygina qilib nur diod yoki svetodiodni o'chirib yoqish dasturini tuzamiza. Buning uchun ushbu dasturlash muhitini ishga tushuramiz, so'ng File→ New bosamiz undan so'ng kichik oyna chiqadi va o'sha oynada *project* ni belgilab *ok* tugmasini bosamiz va

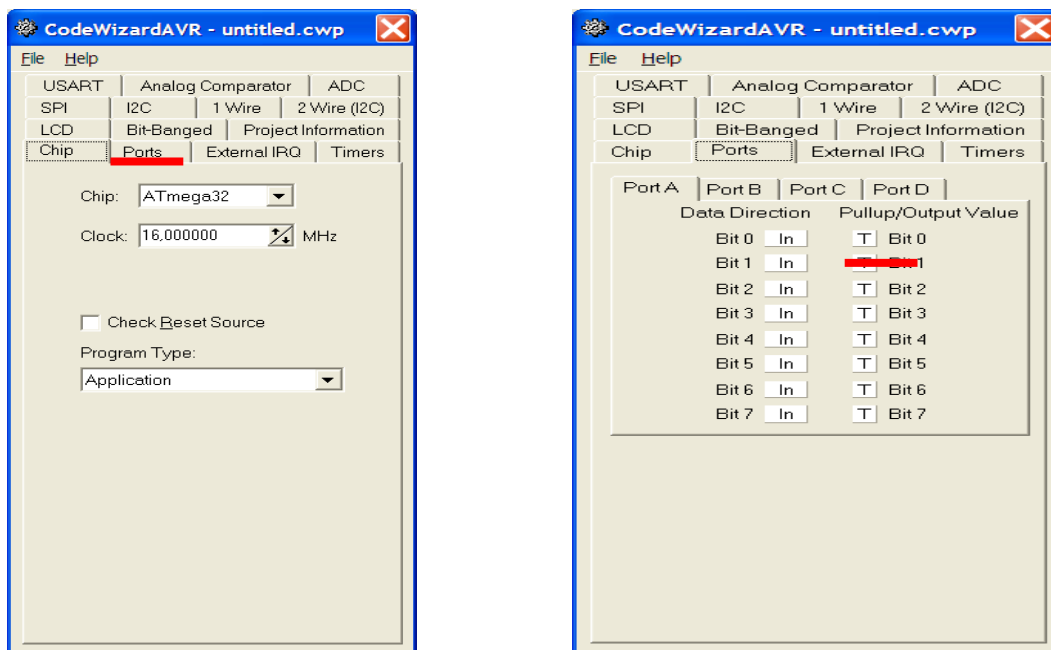


4.28 – rasm. File→ New bo'limi

ushbu dasturlash muhitida kodni generatorini ishga tushurishni so‘raydi unga javoban biz *yes* tugmasini bosamiz.

So‘ng ushbu muhitda qo‘shimcha oyna chiqadi va o‘sha oyna yordamida biz o‘zimizga kerak bo‘lgan MK portlarini qo‘shimcha interfeyslarni, ARO‘ ni hamda boshqa qo‘shimcha pereferiyani sozlashimiz mumkin.

So‘ng u yerdan Ports deb yozilgan menyuga kiramiz va u yerda Mikrakontroller (MK) ning 4 ta portidan birini tanlaymiz.



4.29 -rasm.MK portlarini sozlash jarayonlari

Misol uchun biz PORTA ni 0 bitini ishlatamiz va osha bitni chiqish deb belgilaymiz. PORTA ni chiqish qilib belgilaganimizadan so‘ng kodni generatsiya qilishimiz kerak. Buning uchun File→Generate, *save* and *exit* bosamiz va ushbu dasturimizni 3 bora saqlaymiz. So‘ng bizlarda uzun dastur kodi hosil bo‘ladi:

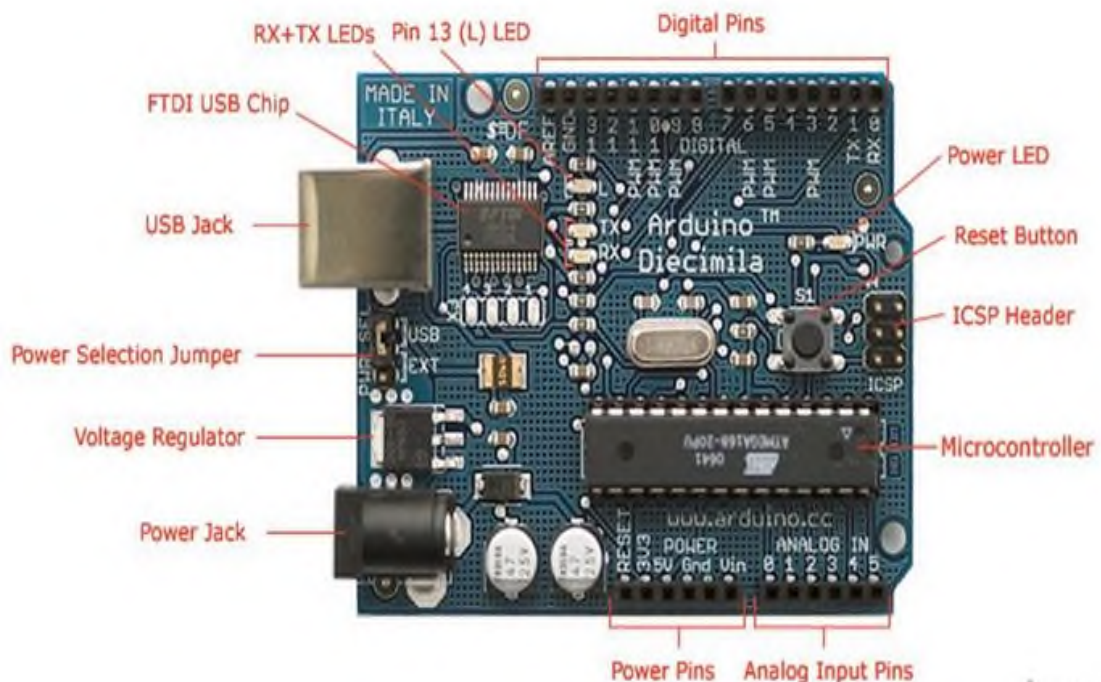
```
#include <mega32.h>
// Declare your global variables here
void main(void)
{
while (1)
{
// Place your code here
};
}
```

Endi biz dasturimizni yozamiz:

```
#include <mega32.h>
#include <delay.h>
// Declare your global variables here
void main(void) // dasturning asosiy qismi
{
PORTA=0x00;// PORTA chiqish
DDRA=0x01;
while (1)
{
// Place your code here
PORTA=0x01; // A portni o - elektrodini yoqamiz
delay_ms(10); // 100 ms kutamiz
PORTA=0x00; // o - elektrodni o'chiramiz
delay_ms(10); // 100 ms kutamiz
};
}
```

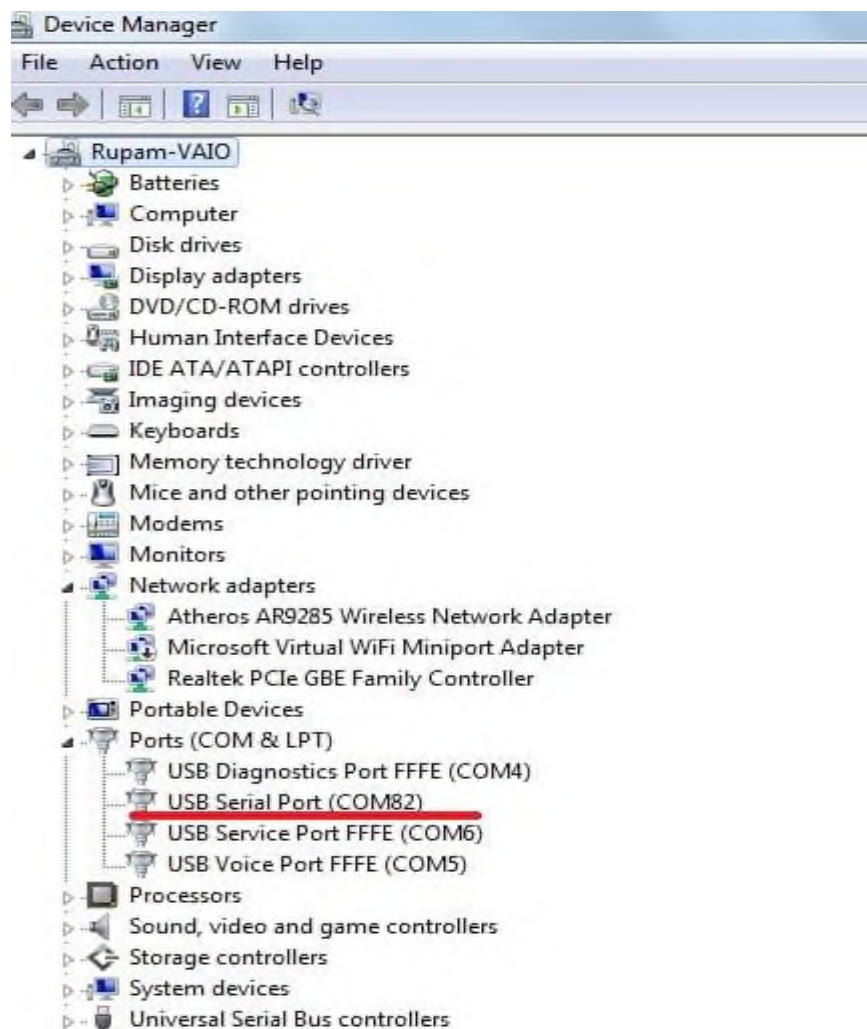
4.7. Arduino imkoniyatlari

Birinchi navbatda Arduino IDE ni yuklab olamiz. Windows muhiti uchun tayyor o'rnatiluvchi dasturni yuklab olishimiz mumkin. Bundan tashqari USB drayverni o'rnatish ham talab qilinadi. Bu juda ham muhim bo'lib, chunki USB drayversiz Arduino IDE ni Arduino platformasiga bog'lab bo'lmaydi.



4.30- rasm. Arduino Diecimilia va Duemilano

Dasturni o‘rnatib bo‘lganimizdan keyin, Arduino platasini USB kabel orqali kompyuterga ulaymiz. Bu xolatda drayverlarni o‘rnatilish jarayoniga guvoh bo‘lishimiz mumkin. Drayver muvaffaqiyatli o‘rnatilib bo‘linganidan so‘ng Device Managerni ochamiz (start->right click on Computer icon->Properties->Device manager).

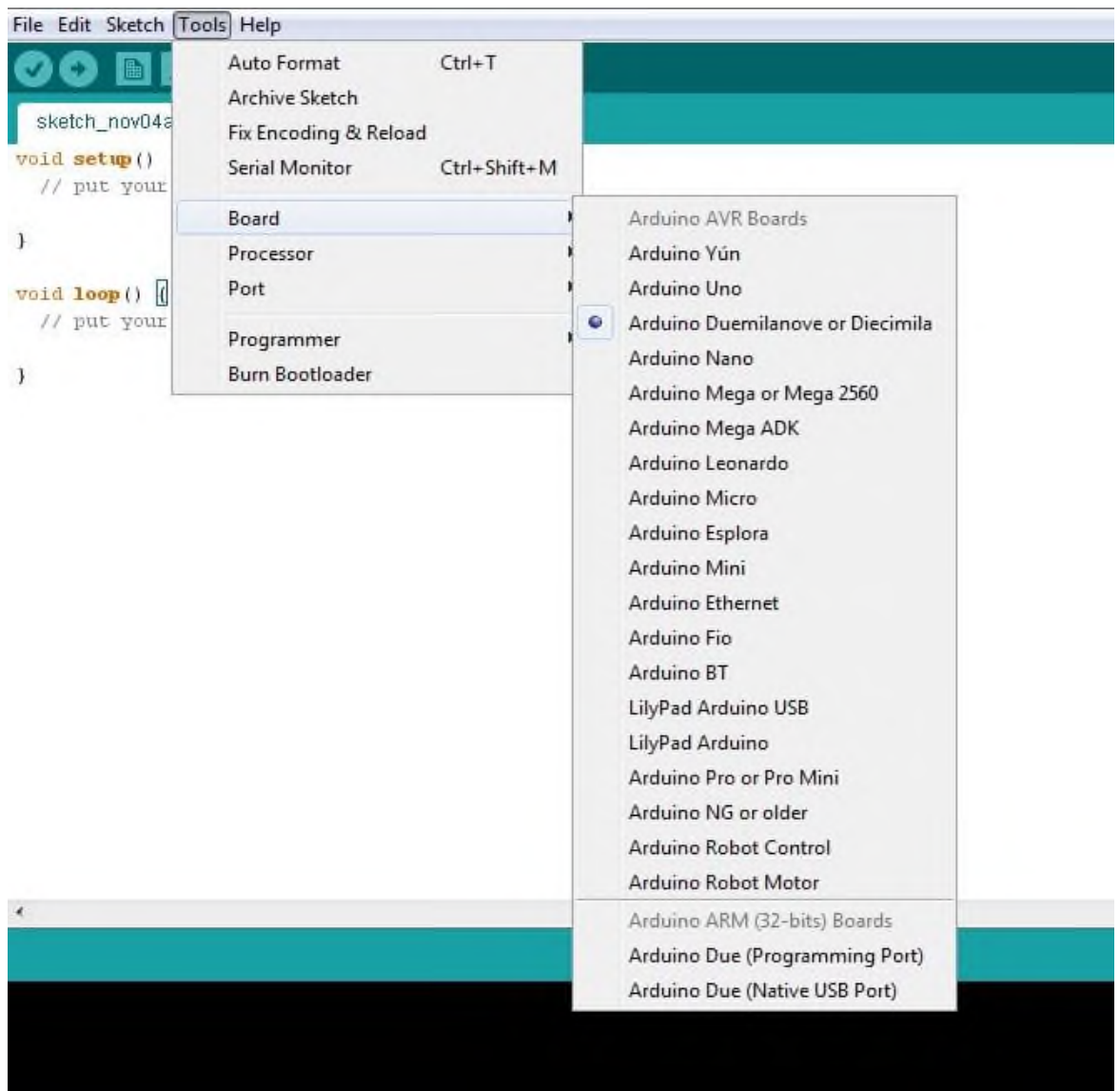


4.31- rasm. Arduino ulanganidan so‘ng Device Managerning ko‘rinishi

Com&LPT bo‘limida USB serial portini ko‘rishingiz mumkin. Agar siz kabelni olsangiz ushbu port g‘oyib bo‘ladi demak ushbu port faqat Arduino platasi kompyuterga ulangan paytda mavjud bo‘ladi.

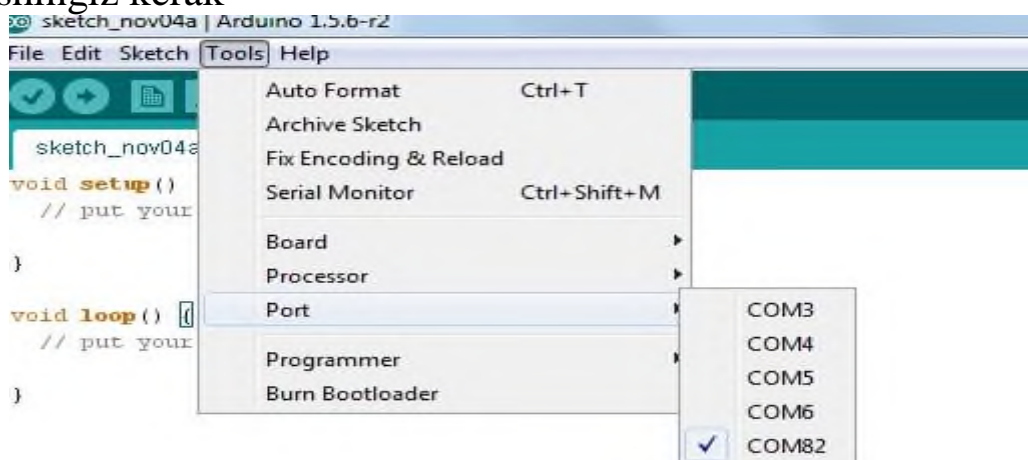
Dastur yozishimizdan oldin biz Arduino IDE bilan platformani o‘zaro bog‘lashimiz kerak.

IDE ochiladi. Tools-> Board bo‘limiga kiriladi



4.32- rasm. Arduino platformasini tanlash

Platforma tanlanganidan so‘ng. Unga mos serial portni tanlashingiz kerak



4.33- rasm. USB Serial portni tanlash

Shu bilan Arduino IDE bilan Arduino platformasi bir-biriga bog‘landi ya’ni dasturiy va texnik qismalar o‘rtasida o‘zaro bog‘lanish o‘rnatiladi.

1. Arduino C dasturlash tili asoslari.

Arduino dasturlash muhiting asosiy strukturasi quyidagicha:

```
void setup() {  
  // put your setup code here, to run once:  
  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  
}
```

setup()- bu bajaruvchi qism bo‘lib, qurilmalarni ulash, uzish va qayta ulash amallarini bajaradi.

loop() – davomylik sikl bo‘lib, dasturning asosiy mantiqiy amallari shu yerda yoziladi.

Arduino Duemilanova/Diecimilia/Uno R3 platformasida 14 ta raqamli pin mavjud. Ular bir-biridan mustaqil shaklda dasturlanishi mumkin. Pin lar OUTPUT (chiqarish) yoki INPUT (kiritish) rejimida bo‘lishi mumkin:

pinMode(PIN_NO,OUTPUT) or pinMode(PIN_NO,INPUT);

Input modeli pinlar uzib ulagichga ulanganida va pin holatni o‘qishi kerek bo‘lganida qo‘llaniladi. Raqamli pinlar faqat 0 va 1 qiymatlarni qabul qiladi. PIN_NO qiymati 0-13 orasida o‘zgarib turishi mumkin.

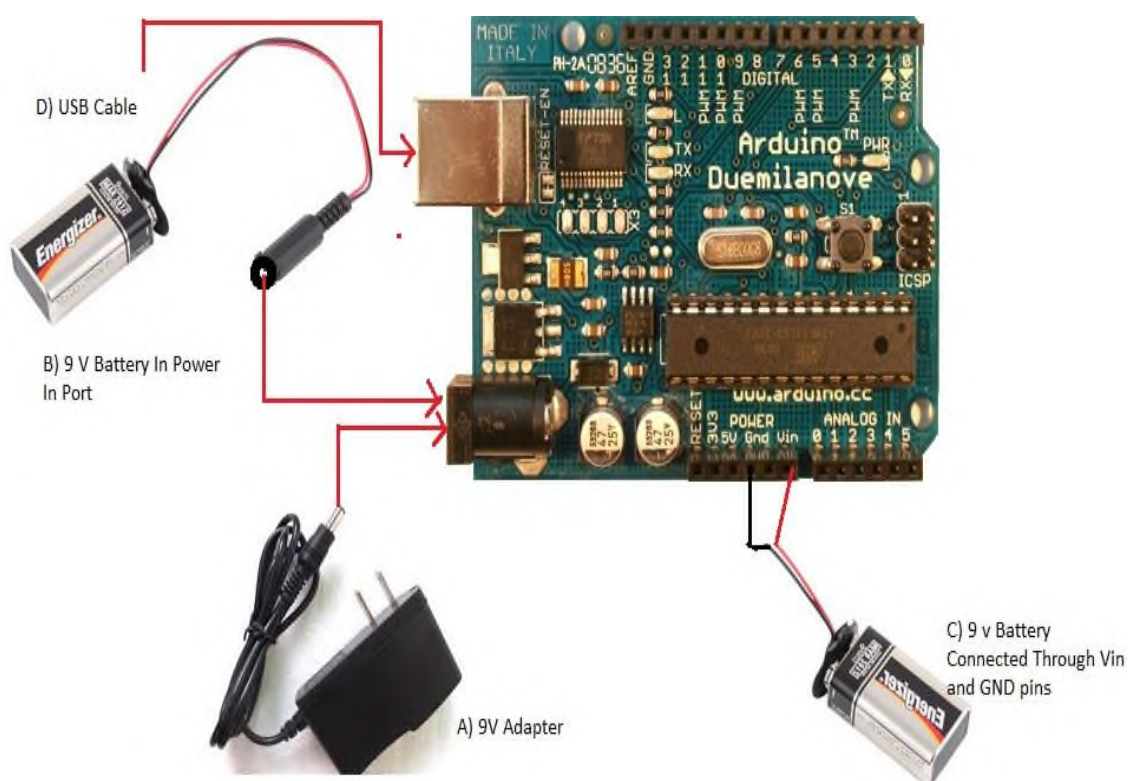
Bir qancha raqamli pinlar PWM sifatida belgilab qo‘yilgan. Ular davriy o‘zgaruvchi impulslar hosil qilish maqsadida qo‘llaniladi. PWM pinida impuls qiymatini o‘zgartirish mumkin. analogWrite(PIN_NO,byteValue) bu yerda byteValue 0 va 255 orasida bolishi mumkin. PWM pinlar DC motorlarni tezligini nazorat qilishga imkon beradi.

Arduino da 6 ta analog pin bo‘lib, ular 10 bitli ADC ga ulangan. Sensorlar ham ushbu pinlarga ulanadi. Arduino kuchlanishni quyidagi ifoda yordamida aniqlaydi: int value=analogRead(ANALOG_PIN_NO)

2.Arduinoda quvvat boshqaruvi.

O'rnatilgan tizimlar motorlar, triggerlar va sensorlarni boshqarish uchun energiya bilan ta'minlanishi kerak. O'rnatilgan tizimlarni qancha quvvat istemol qilishi va qancha quvvat saqlay olishi muhim aspektlari hisoblanadi. Bu bo'limda Arduinoni elektr boshqarishi haqida ma'lumotlar beriladi.

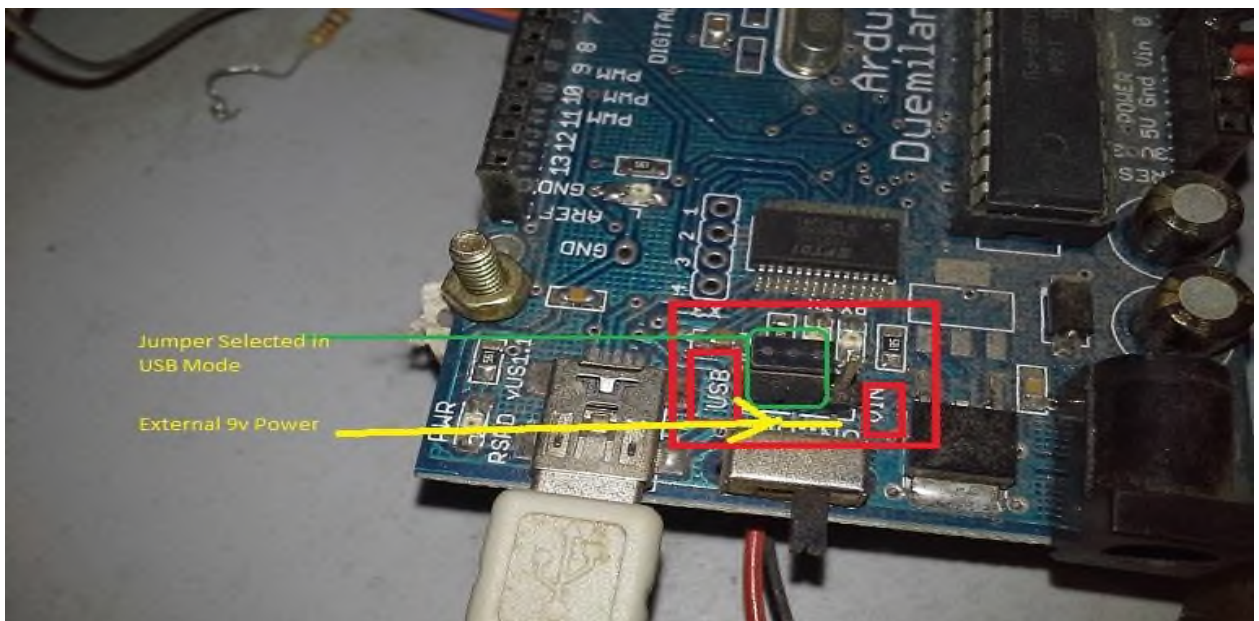
Arduino platformasi boshqa ko'plab platformalar singari 5 V kuchlanishni xosil qilib bera oladi. U USB orqali yoki tashqi elektr manbalari orqali amalda kuchaytirilishi mumkin. Arduino platformasi batareyalar singari tashqi elektr manbalaridan quvvat olishi mumkin, ular 9 V kuchlanishni ta'minlay oladi. Quyidagi rasmda Arduino platasini quvvat bilan ta'minlashning bir qanch yo'llari keltirilgan:



4.34- rasm. Quvvat bilan ta'minlashning bir qancha yo'llari

Arduino platasi 5 V kuchlanishni o'zgartirish uchun maxsus o'zgartirgichga ega. Tashqi quvvat manbalari aynan shu o'zgartirgich yordamida boshqariladi.

Siz 9 V adapterdan foydalanishingiz mumkin. Agar siz platani batareyya orqali energiya bilan ta'minlashni xoxlasangiz, unda maxsus portlarga ulanishingiz kerak. Platada maxsus GND va VIN pinlari bo'lib ular batareyyani plataga ulanishi uchun xizmat qiladi.



4.35- rasm. Jumper orqali energiyani ta'minlanishi

4.35- rasmga e'tibor berib qarasangiz unda RST, 3V, 5V, GND, va VIN pinlari birgalikda guruhlashtirilganini ko'rishingiz mumkin. Bularning barchasini birgalikda quvvat Porti deb ataymiz. 5V va 3V pinlari VCC pinlarini deb ham ataladi ular tashqi qurilmalarga manba bo'lib hisoblanadi. Misol uchun siz LM35 sensorini ishlatmoqchi bo'lsa sizga 5 V energiya kerak bo'ladi. Buni 5V pini ta'minlab beradi.

Arduinoda 14 ta I/O pinlari mavjud. Qachonki yuqori level energiya talab qilinsa, bu pinlar 5 V energiyani ishlab chiqaradi. I/O pinlari 400 mA sigimga ega.

Indikatorlar va Switchlar

LED - Light Emitting Diode – diodlari juda ko'p ishlatiladigan indikator turlaridan biri bo'lib hisoblanadi. U foydalanuvchiga tizim xolati yaxshi ekanligi haqida ma'lumot olish uchun ishlatiladi. LED diodlari turli ranglarda va turli maqsadlarga mo'ljallangan bo'ladi. LED lar asosan 30 mA tok ostida ishlaydi. LED diodlari asosan 1.2 V va 24 V kuchlanish asosida boshqarilishi mumkin.

Arduino platasida LED diodlar 13 raqamli pin orqali boshqariladi. Agar biz biror qurilmani Arduino I/O pinlari orqali boshqarishni xoxlasak, unda bu pinni sozlashimiz kerak.

Misol uchun:

`digitalWrite (13,HIGH).`

Oddiy ochib yonuvchi LED dasturi Arduino muhitida saqlangan bo'lib, uni quyidagicha ochish mumkin: File->Examples->Basics->Blink

```

int led = 13;

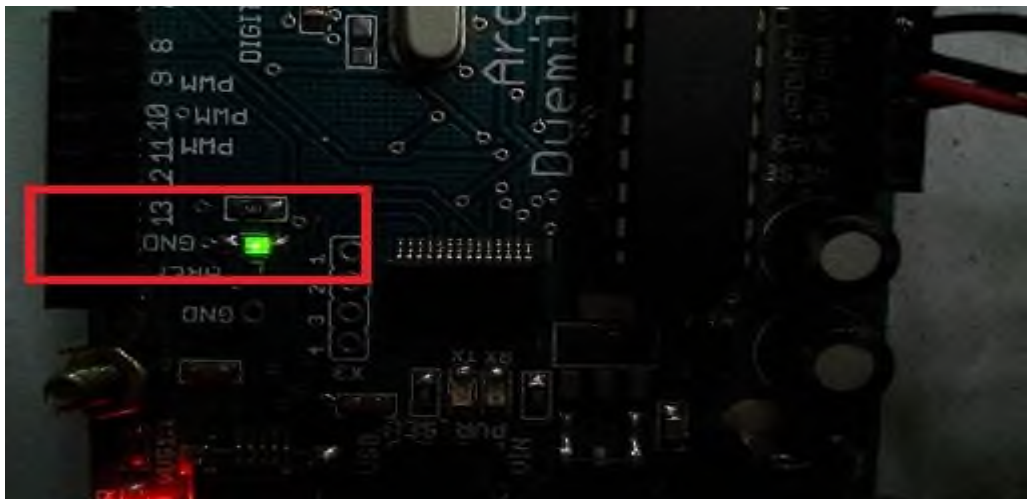
// the setup routine runs once when you press reset:

void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage
  level)
  delay(1000);           // wait for a second
  digitalWrite(led, LOW); // turn the LED off by making the voltage
  LOW
  delay(1000);           // wait for a second
}

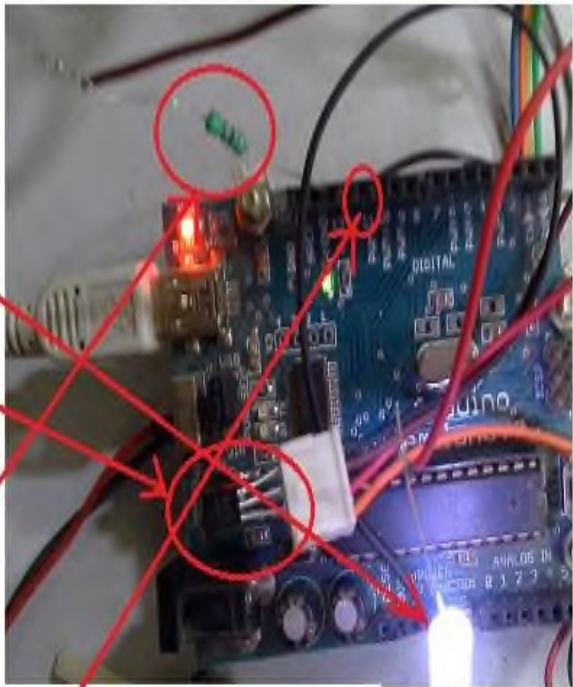
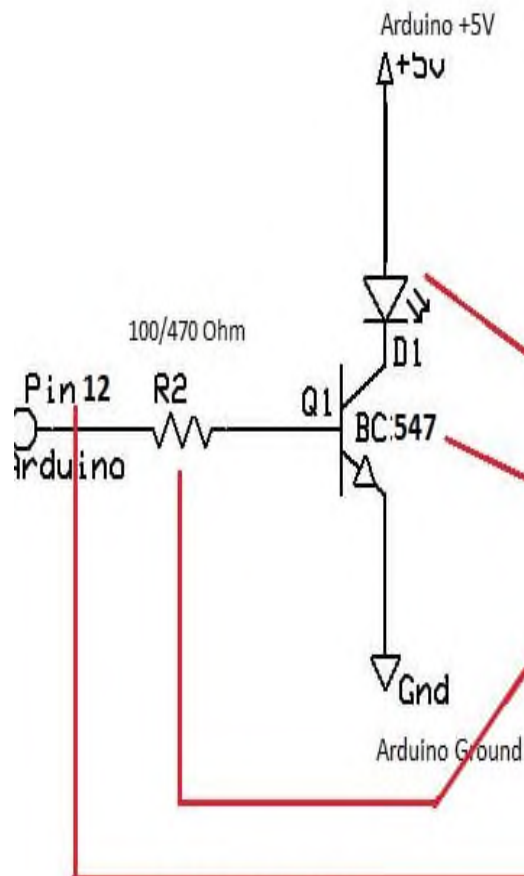
```

Ushbu dasturda LED diode xar 1 sekunda o‘chib yonishi misoli keltirilgan:



4.36- rasm. 13-Pinga bog‘langan LED

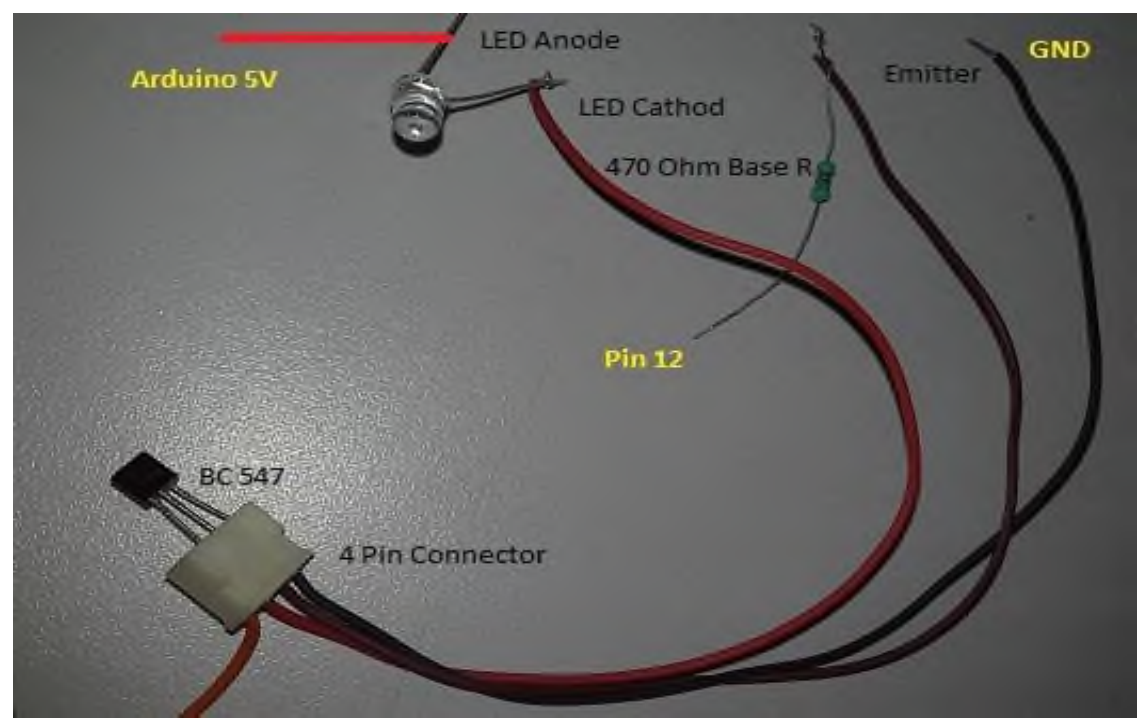
Platformadagi LED ni yonib o‘chishi bu oddiy holat bo‘ladi. Lekin LED tashqi diodi ulangan bo‘lib uni raqamli pinlar bilan boshqarish kerak bo‘lsin.



A) Circuit Diagram

B) Real World Connection

4.37- rasm. Tashqi LED diodini ulanishi



4.38 -rasm. Bog'lanishni yaqinlashgan ko'rinishi

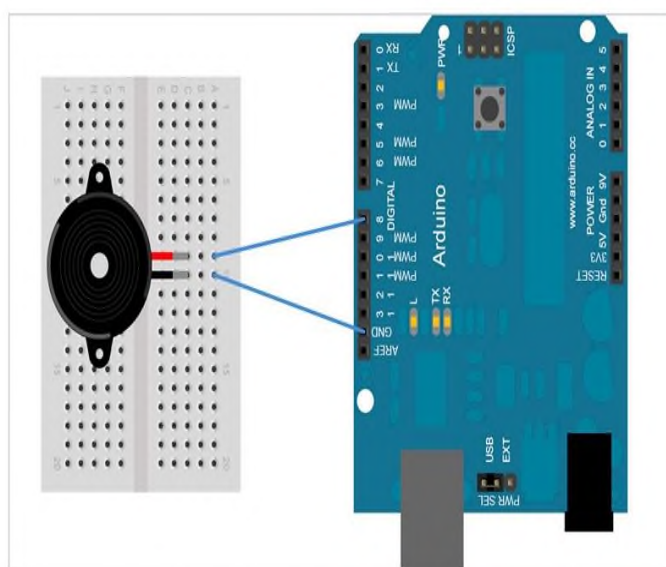
Buzzer bu LED diodlariga o'xshash, chiquvchi komponentdir. Buzzerlarning ko'plab turi mavjud bo'lib, eng ko'p ishlatiladigani Piezo electric buzzerlari hisoblanadi. Lekin Buzzerlar kam elektr ta'minoti talab qiladi. Ular 10 mA tok ostida boshqarilishi mumkin.

Siz buzzerni 8 – pinga ulashingiz mumkin. Agar yuqoridagi kodni buzzer uchun kiritsangiz u ham ishlaydi faqat pin nomerini o'zgartirishni unutmang.

Buzzer da 2 ta pin bo'lib uzun Anod hisoblanib raqamli pinga ulanadi va kalta qismi esa katod hisoblanib yerga ulanadi.

Quyidagi rasmda qizil sim raqamli pinga ulangan sim va ko'k/qora sim esa yerga ulanadi.

SWITCH/Uzib ulagichlar. Switch ya'ni uzib ulagichlarni ko'plab turi mavjud bo'lib, bunga misol qilib tugmachani misol qilishimiz mumkin. Tugmachani bosish orqali LED diodini yoqib o'chirishimiz mumkin.



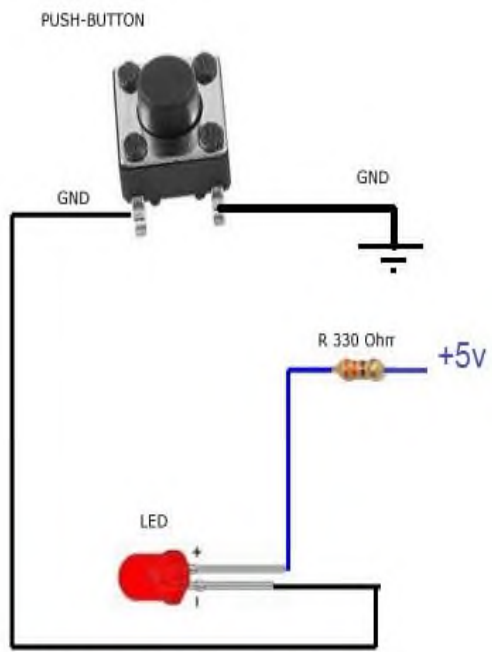
A) Circuit Diagram



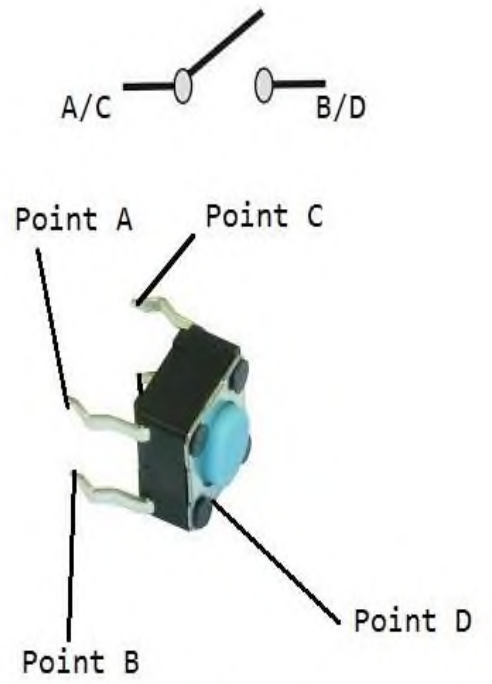
B) Real World Connection

4.38- rasm. Arduino Buzzer bilan

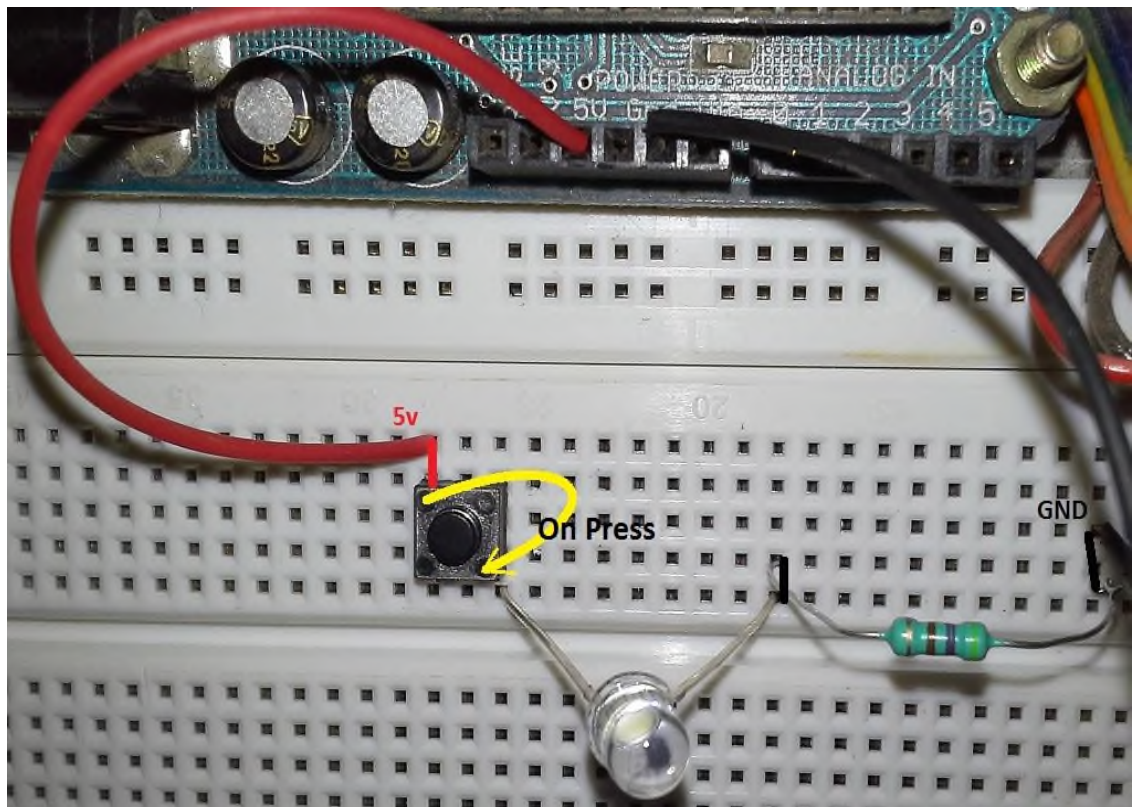
Uzib ulagichlarda to'rta oyoqcha mavjud bo'lib, ular 2 juft qutblardan tashkil topadi.



A) Circuit Diagram



B) Switch Pinout





4.39- rasm. SWITCH/Uzib ulagichlar

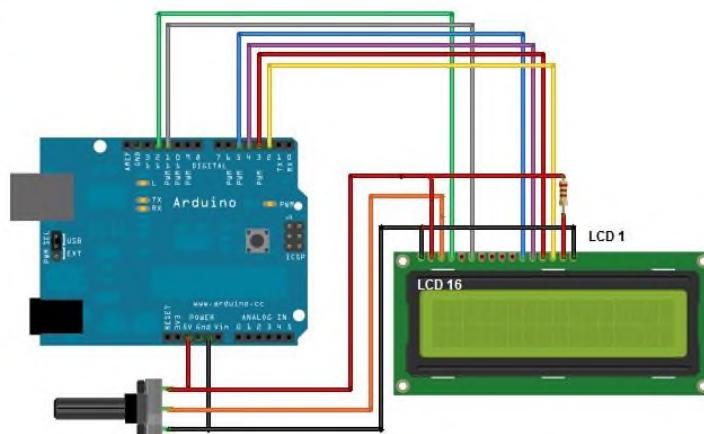
Display Arduinoning eng muhim chiqaruvchi qurilmalaridan biri displaylar bo‘lib, ularning bir necha turlari mavjud. LCD displaylar, Grafik displaylar, Yetti segmentli displeylar va boshqalarni misol keltirishimiz mumkin. Ikki qatorli display eng ko‘p ishlatiladigan turidir.

LCD larga 2 ta VCC kerak: biri oldi frontni yoritish uchun ikkinchisi orqa frontni yoritish uchun. 11, 12, 13 va 14 pinlar Arduinoga ulanishi kerak. LCD displayini boshqarishimiz uchun bizga jami 6 ta pin kerak. VO pin ravshanlikni boshqaradi.

LCD Function	LCD Pin	Arduino
VSS	1	GND
VDD	2	+5V
VO	3	4.7 K resistor
RS	4	Pin 12
RW	5	GND
Enable	6	Pin 11
D4	11	Pin 5
D5	12	Pin 4
D6	13	Pin 3
D7	14	Pin 2
A (Backlight +)	15	+5V
K (Backlight -)	16	GND

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

A) Connection Table



B) Circuit Diagram

4.40- rasm. LCD display

Quyida displayga so‘z chiqarish bo‘yicha oddiy dastur keltirilgan bo‘lib, unda Liquid Crystal kutubxonasiidan foydalanilgan.

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12,11,5,4,3,2); //create the lcd variable
void setup()
{
```

```

pinMode(13,OUTPUT);
digitalWrite(13,HIGH);
lcd.clear();           //clear the LCD during setup
lcd.begin(16,2);      //define the columns (16) and rows (2)
}
void loop() {
  lcd.print("Codeproject.com");
  lcd.setCursor(0,1);
  lcd.print("Rupam, Gulbarga"); //print...
  delay(2000);
  lcd.clear();        //clear LCD, since we are still on 2nd line...
  lcd.home();        //set the cursor the top left
}

```

Sensorlar: Sensorlar kichik qurilmalar bo‘lib ularning asosiy vazifasi fizik parametrlarni elektrik paramatrlarga aylantirib berish uchun ishlatiladi. Sensorlarni bir necha turi mavjud bo‘lib uni quyidagi rasmda ko‘rishimiz mumkin:



4.41- rasm. Sensorlar

Analog/Raqamli converter. Boshqaruv kuchlanishi 5 V bo‘lib, ko‘plab sensorlar aynan 5 V kuchlanish ostida ishlaydi.

Bundan kelib chiqadiki sensorlarni elektr parametrlari analogdir.

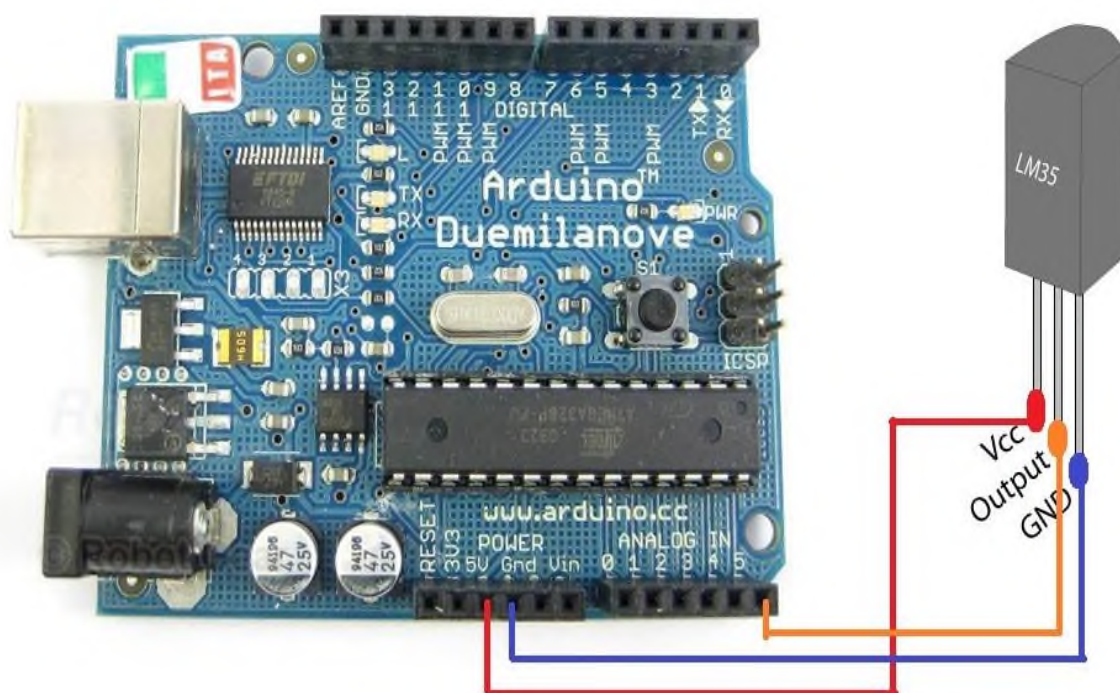
Ammo mikrokontrollerlar asosan raqamli ma’lumotlar ustida amallar bajariladi. Buning uchun Arduinoda maxsus 10 bitli A/D konverterlari mavjud bo‘lib, ular analog ma’lumotlarni raqamli ma’lumotlarga o‘zgartirish uchun qo‘llaniladi. Uni ishlash prinsipi analog ma’lumotni 10 bitli ko‘rinishda kvantlaydi va eng yaqin kvant qiymatini qabul qiladi.

$$\text{Sensor output Voltage} = x * 5 / 1024$$

Temperatura sensorlari

Eng ko‘p tarqalgan temperatura sensorlaridan bir LM 35 bo‘lib, u eng arzon samarali hisoblanadi. U 0-5V kuchlanish ostida ishlab, 0-92 C haroratni o‘lchashga imkon yaratadi.

LM35 da 3 ta pin mavjud bo‘lib, ular mos ravishda yerga, Vccga va Arduinoni analog portiga ulanadi. Quyidagi chizmada qurilma 5V pinga va 5-pinga ulangan.



4.42 -rasm. LM 35 sensori.

Ushbu LM 35 temperaturani kuchlanishga bog‘laydi, ya’ni temperatura taxminan 90 C bo‘lganida chiqish kuchlanishi 1 V atrofida bo‘ladi.

```

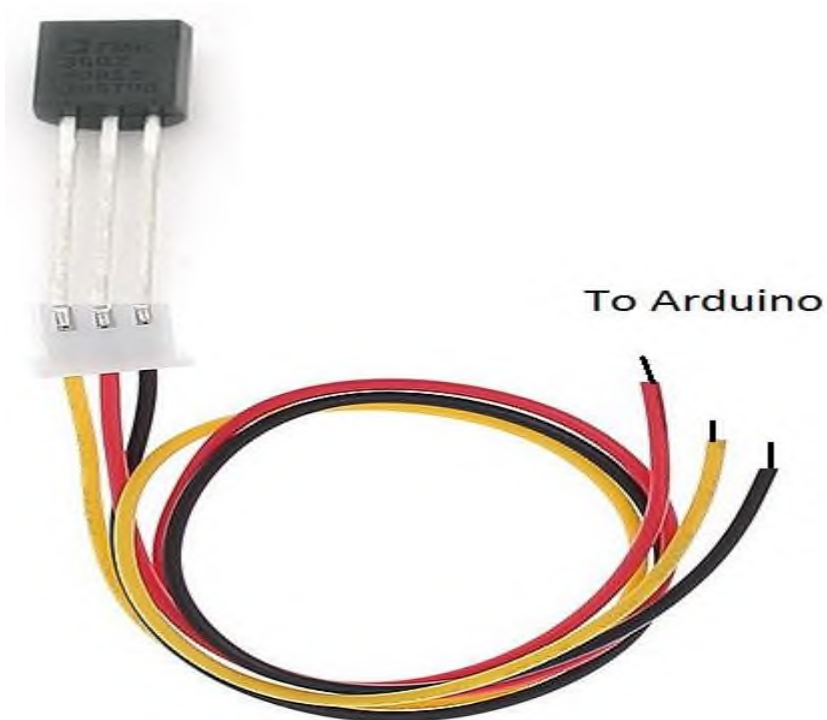
float tempC;
int reading;
int tempPin = 0;

void setup()
{
  analogReference(INTERNAL);
  Serial.begin(9600);
}

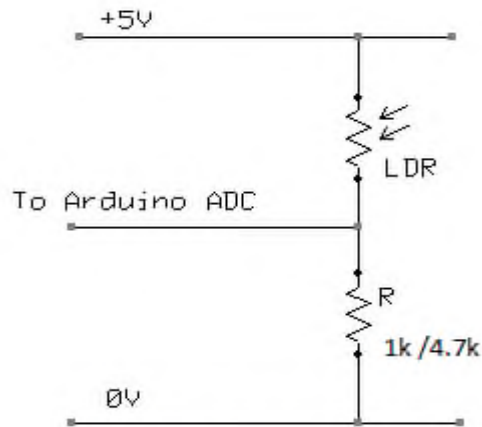
void loop()
{
  reading = analogRead(tempPin);
  tempC = reading / 9.31;
  Serial.print("Temp= ");
  Serial.print(tempC);
  Serial.println(" 'C");
  delay(1000);
}

```

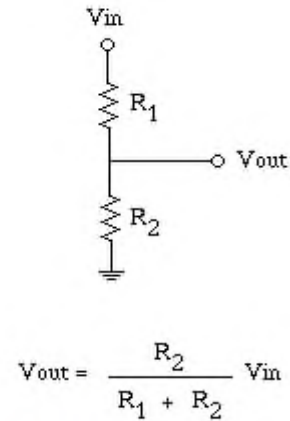
Temperatura sensorini umumiy ko‘rinishi quyidagicha:



4.43- rasm. LDR sensors



a) LDR With Arduino



$$V_{out} = \frac{R_2}{R_1 + R_2} V_{in}$$

b) Voltage Divider

```

float light;
int reading;
int lightPin = 5;

void setup()
{
  Serial.begin(9600);
}
void loop()
{
  reading = analogRead(lightPin);
  light = (float)reading*100/1024.0;
  Serial.print("Light = ");
  Serial.print(light);
  Serial.println(" %");
  Serial.println(reading);
  delay(1000);
}

```

LDR Light Dependent Resistor. Unda yorug‘lik qanchalik ravshan bo‘lsa , qarshilik shuncha ortadi va chiqish kuchlanishi shuncha kichik bo‘ladi va aksincha.

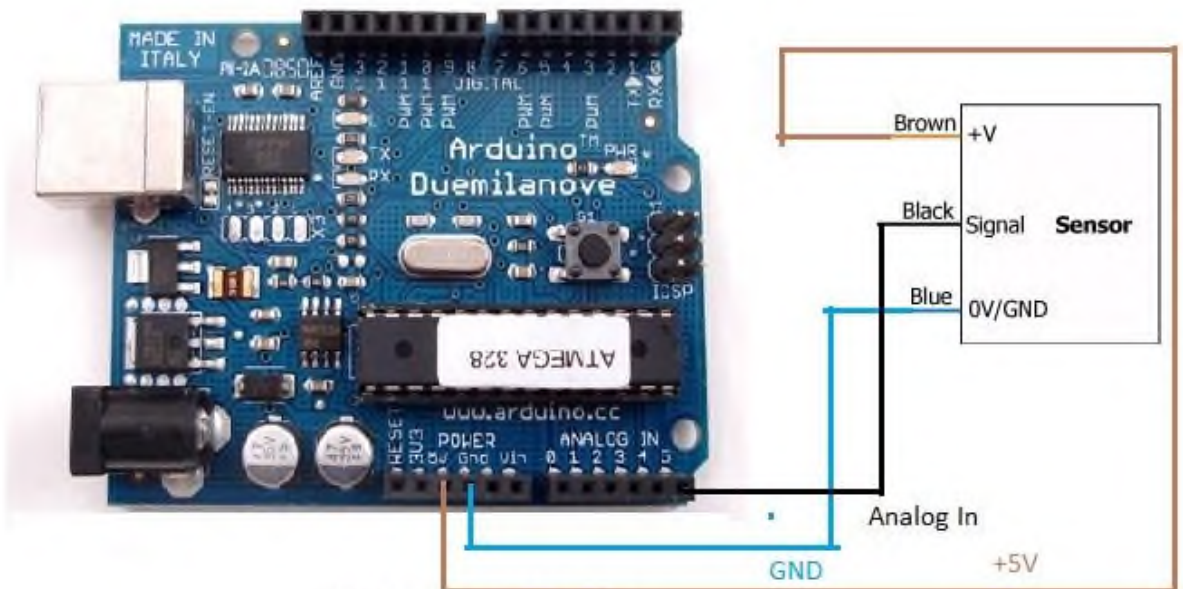
Metal sensor. Metal sensorlar induktiv sensorlar qatoriga kiradi. Bu yerda biz ID18-3008NA rusumidagi metal sensorni ish prinsipini ko‘rib chiqamiz:



A) Top View



B) DC 6-36V PNP NO 3-wire 4mm Cylindrical Inductive Proximity Sensor



C) Overall Circuit Diagram

4.44- rasm. Metal sensor

Ushbu metal sensorlarda uch xil rang bo‘lib, ko‘k yerga ulanish uchun, jigar rang Vcc ga va qora rangli sim esa 5V ga ulanish uchun mo‘ljallangan. Ultrasonic sensor. Ushbu sensor ultratovush orqali obyektgacha bo‘lgan masofani aniqlash uchun xizmat qiladi.

Quyida metal sensorlar ishlatilishiga asoslangan dastur keltirilgan:

```
float metal;
int reading;
int metalPin = 3;

void setup()
```

```

{

Serial.begin(9600);
}

void loop()
{
reading = analogRead(metalPin);
metal = (float)reading*100/1024.0;
Serial.print("Metal in Proximity = ");
Serial.print(metal);
Serial.println(" %");
if(reading>250)
Serial.println("Metal Detected");
delay(1000);
}

```

U uzatuvchi va qabul qiluvchi qismdan tashkil topgan. Uzatuvchi ultra qisqa to‘lqinli impulslar ketma- ketligini uzatadi, agar oldinda biror obyekt bo‘lsa undan to‘lqi ortga qaytadi. Qabul qilgich to‘lqin borib kelish vaqtini hisoblaydi va uni tovush tezligiga ko‘paytirib masofani aniqlaydi.

Motorlar bilan ishlash. Agar siz o‘zingizni shaxsiy robotingizni qurmoqchi bo‘lsa ngiz motorlar eng muhim qurilmalardan biri xisoblanadi. DC motorlar motorlraning eng ko‘p tarqalgan turidir.

```

#include <NewPing.h>
#define TRIGGER_PIN 12 // Arduino pin tied to trigger pin on the
ultrasonic sensor.
#define ECHO_PIN 11 // Arduino pin tied to echo pin on the
ultrasonic sensor.
#define MAX_DISTANCE 200
// Maximum distance we want to ping for (in centimeters). Maximum
sensor distance is rated at 200-300cm.
NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE); //
NewPing setup of pins and maximum distance.
void setup() {
pinMode(13,OUTPUT);

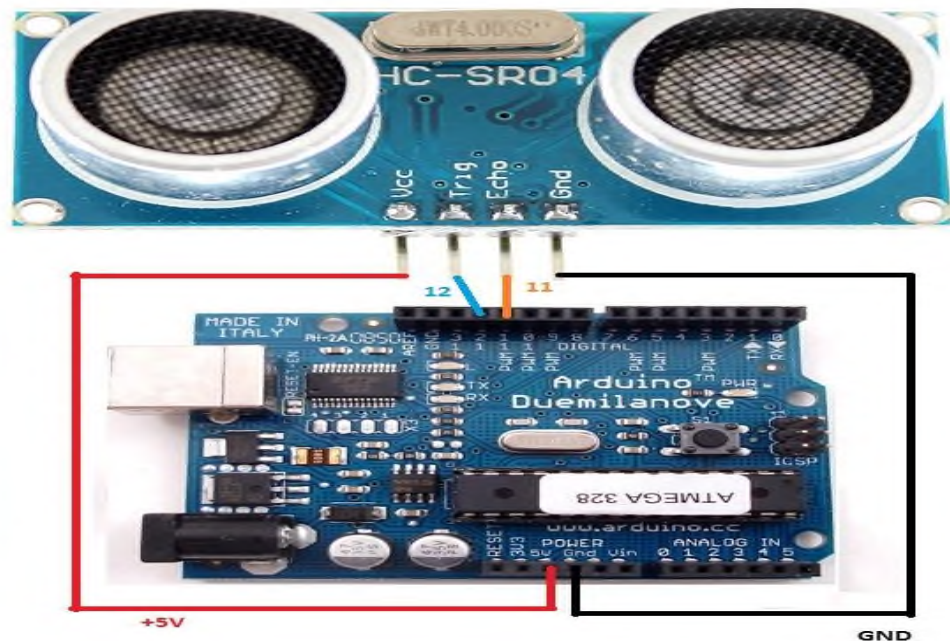
```



```

Serial.begin(19200); // Open serial monitor at 115200 baud to see ping
results.
}
void loop() {
  delay(50);
  // Wait 50ms between pings (about 20 pings/sec). 29ms should be the
  shortest delay between pings.
  unsigned int uS = sonar.ping(); // Send ping, get ping time in
  microseconds (uS).
  Serial.print("Ping: ");
  unsigned int dist=uS / US_ROUNDTRIP_CM;
  if(dist<5 && dist>0) {
    digitalWrite(13,HIGH); }
  else {
    digitalWrite(13,LOW); }
  Serial.print(dist);
  // Convert ping time to distance and print result (0 = outside set distance
  range, no ping echo)
  Serial.println("cm");}

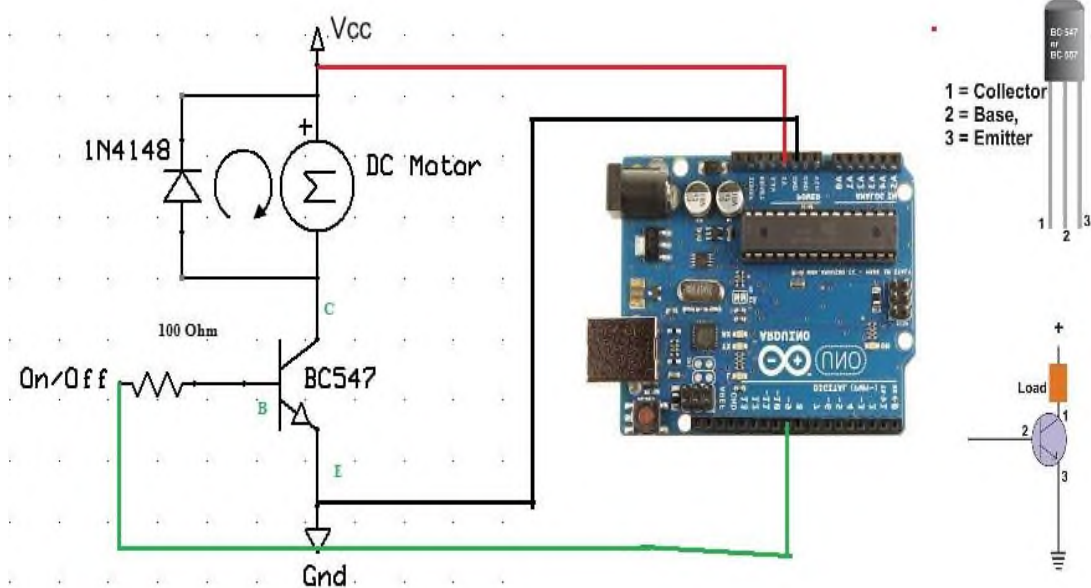
```



4.45- rasm. Ultrasonic sensor

Agar siz motorlarni tezligini nazorat qilmoqchi bo'lsa ngiz unda PWM piniga tranzistor ulashimiz va impulslarni o'zgartirishimiz kerak.

Agarda faqat to'xatib / yurgazish uchun bo'lsa Switchlardan foydalanish tavsiya etiladi.



BOBLAR BO‘YICHA NAZORAT TESTLARI

№	Savollar	JAVOBLAR		
1	Kutish rejimida mikrokontrollerni qaysi moduli ishlashni tugatadi	markaziy protsessor	takt generatori	taymer
2	Chastotani eng katta stabilligini mikrokontrollerni qanday taktlash usuli ta'minlaydi	RC-zanjirini qo'llash bilan	kvarsli rezonator qo'llash bilan	keramik rezonator qo'llash bilan
3	Mikrokontrollerda axborotni uzluksiz kiritish oddiy qurilmasi sifatida nima qo'llaniladi	analog-raqamli o'zgartirgich	kuchlanish komparatori	rezistiv bo'luvchi
5	Mikrokontrollerni (MK) takt generatorini yurg'azishda vaqt to'xtalishi nimaga kerak	generator chastotasini stabillashtirish uchun	MK ishga tushurishda energiya foydalanishni qisqartirish uchun	MK registrlar boshlang'ich xolatga o'tqazish uchun
6	MK tarkibida analog raqamli o'zgartirgichlar qaysi turlari asosan qo'llaniladi	integrallovchi	parallel	ketma ketli yaqinlashtirish
7	MK KMOPni istemol toki generator takt chastotasidan qanday bog'langan	bog'lanmagan	chastotani kvadrat ildiziga proporsional	kvadratik
8	Mikrokontroller (MK) qo'riqlovchi taymeri to'ldirilganlida qanday xolat bo'lishi mumkin?	uzilishni so'rash signalini tashkil etish	energiya foydalanishni qisqartirish rejimiga o'tish	MK o'chirib tashlamoq

9	Mikrokontroller chiqishida sodda raqamli analog o'zgartiruvchi sifatida nima qo'llaniladi	pastgi chastotali filtrli kenglik-impulsli modulyator	operatsion kuchaytirgich	elektron kalit
10	PIC16CXXX o'rta cha darajadagi MK oilasida buyruqlarni qanday formati qo'llanadi?	12 bit	14 bit	16 bit
1	Kutish rejimida mikrokontrollerni qaysi moduli ishlashni tugatadi	markaziy protsessor	takt generatori	taymer
11	RIS – mikrokontrollerda buyruqlar bajaruvchi ikki pogonali konveer nima beradi	takt chastotani ikki marotaba ko'tarish imkoniyati	buyruqlarni bir vaqtda tanlash va bajarish imkoniyatini	ikki buyruqni parallel bajarish imkoniyati
12	RIS- (MK) maxsus funksiyalariregistr lari nima uchun qo'llaniladi	MK ishini boshqarish uchun	MK maxsus buyruqlarini bajarish uchun	oraliq ma'lumotlarni saqlash uchun
13	PIC16F8X guruxi mikrokontrollerlar ni buyruqlar xisolagichlarini kichik baytini ichidagilari qaerda saqlanadi?	STATUS registrida	OPTION registrida	PCLATH registrida
14	PIC –(MK) da qaysi arxitektura qo'llaniladi	fon-neyman RISC- protsessori bilan	fon-neyman CISC- protsessori bilan	garvards RISC- protsessori bilan

15	PIC- mikrokontroller bir buyruqli sikl bajarishga kancha takt egallanadi	1	2	4
16	PIC16F8X guruxi mikrokontrollerlar ni ma'lumotlar xotirasi bankini tanlashdagi axborot qaerda saqlanadi	STATUS registri IRP bitida	STATUS registri RPO bitida	dasturlar xotirasida
17	PIC16F8X guruxi mikrokontrollerlar ni stek ko'rsatgichlari qaerda saqlanadi	foydalanuvchi ga amalga oshirib bo'lmaydi	OPTION registrda	PCLATH registrda
18	RIS- mikrokontrollerda bir buyruqni bajarish qancha siklni egallaydi	1	2	4
19	PIC16F8X guruxi mikrokontrollerlar ni Kakova glubina apparat stekini chuqurligi qanday	4	8	16
20	PIC16F8X guruxi mikrokontrollerlarn i operatsiya natijasi belgilari bitlari qaerda saqlanadi	STATUS registrda	OPTION registrda	taymer/xisoblagi ch registrda
21	PIC16F8X guruxi mikrokontrollerlarn i buyruqlar xisoblagichi razryadligi nimaga teng	11	12	13

22	PIC16F8X seriyali MK larda qayta yuklashni necha ko'rinishi tadbiiq etilgan	1	2	3
23	PIC16F8X seriyali MK larda dasturiy ximoya kodini o'chirish mumkinmi	mumkin emas	mumkin, agar identifikatsion kodni bilsa	MK ma'lumoti va xotira tarkibi bilan birga
24	PIC16F8X seriyali MK lar buyruqlar tizimda foydalanuvchi registrlar sifatida nima qo'llanilishi mumkin	ishchi registr W yoki buyruqda ishlatilayotgan registr.	ixtiyoriy adreslovchi registr	registr INDF
25	PIC16F8X seriyali MK da shartli o'tishni tashkil qiluvchi buyruqlarni aniqlang	Buyruq DECFSZ f, d, INCFSZ f, d, BTFCSC f, b i BTFCSS f, b	Buyruq RETFIE i RETLW k	*Buyruq CALL k i GOTO k
26	Parallel portlar asosida MK va tashqi qurilmalar bilan ma'lumot almashish algoritmi qanday ko'rsatiladi	Strobli va oddiy dasturiy kiritish-chiqarish	Xar bir MK da bir qancha kiritish-chiqarish	Oddiy dasturiy kiritish-chiqarish
	Parallel portlar asosida MK va tashqi qurilmalar bilan ma'lumot almashish algoritmi necha turga ajratiladi	3	5	4
27	DEVICE direktivasi dasturlashda qanday vazifani bajaradi	Mikrokontroller turini belgilaydi	Dastur boshlanishi	Takrorlanuvchi dastur

28	BOR-bu nima	Manba kuchlanishi pasayganda qayta yuklanish taymeri	Manbaga ulanish taymeri	Nazorat taymeri (storajevoy taymer)
29	Berilgan dastur asosida C portdagi ma'lumotlarni ikkilik kodini aniqlang. <i>Ldi temp, 0x0B</i> <i>Out ddr, temp</i>	00001011	00101010	00000011
31	Berilgan dastur asosida C portdagi ma'lumotlarni ikkilik kodini aniqlang. <i>Ldi temp, 0xC6</i> <i>Out ddr, temp</i>	11000110	11010000	01011101
32	Berilgan dastur asosida C portdagi ma'lumotlarni ikkilik kodini aniqlang. <i>Ldi temp, 0x7A</i> <i>Out ddr, temp</i>	01111010	00001111	00010101
33	Berilgan dastur asosida C portdagi ma'lumotlarni ikkilik kodini aniqlang. <i>Ldi temp, 0xBF</i> <i>Out ddr, temp</i>	10111111	00011101	00100010

34	Berilgan dastur asosida C portdagi ma'lumotlarni ikkilik kodini aniqlang. <i>Ldi temp, 0xD5</i> <i>Out ddrc, temp</i>	11010101	00100010	00001001
35	Berilgan dastur asosida C portdagi ma'lumotlarni ikkilik kodini aniqlang. <i>Ldi temp, 0xFF</i> <i>Out ddrc, temp</i>	11111111	10101010	10111110
36	IN buyrug'ining vazifasi	Kiritish buyrug'i	Xotiraga ko'chirish.	Xotiradan o'qish
37	Dasturli uzilishdan qaytishda qanday buyruq qo'llaniladi?	uzilishlardan qaytish buyrug'i	mutlaqo o'tish buyrug'i	uzilishni chaqirish buyrug'i
38	Kutish rejimida mikrokontrollerni qaysi moduli ishlashni tugatadi?	markaziy protsessor	takt generatori	taymer
39	Mikrokontroller asosidagi MPT larining ish jarayonining afzalligi nimada	real vaqt ichida ishlashi	Offlayn rejimida ishlashi	Kutish rejimida ishlashi
40	Mikrokontrollerda axborotni uzluksiz kiritish oddiy qurilmasi sifatida nima qo'llaniladi?	kuchlanish komparatori	Analog-raqamli o'zgartirgich	rezistiv bo'luvchi
41	Mikrokontrollerda qaysi dasturiy tillardan foydalaniladi	Assembler Micropascal C	Assembler va C, Delphi	Assembler va C C#

42	Mikrokontrollerlar da qanday pereferiya qurilmalari mavjud	ARO'	taymer	hisoblagich
43	Mikrokontrollerda iste'mol toki kuchlanish manbaidan qanday bog'langan?	to'g'ri chiziqli	Bog'lanmaydi	qayta proporsional
44	Rivojlangan MK oilasida qaysi razryad keng foydalaniladi.	8 razryadli	16 razryadli	32 razryadli
45	MK da xotira turi EPROM qanday vazifani bajaradi	Elektr signallar asosida dasturlanib, ultrafiolet nur asosida o'chiriladi	Doimiy xotira	Operativ xotira qurilmasi
45	C dasturlash tili qanday pog'onali dasturlash turiga kiradi	o'rta (quyi xam yuqori xam emas)	Quyi	Yukori
47	Mikrokontrollerlar da chastota oshgan sari ularning quvvat iste'mol qilish oshadimi?	Ha	Yo'q	Bog'liq emas
48	MK da qo'riqchi taymerning vazifasi	Dasturiy ta'minot xatoligini aniqlash va protsessorni qayta yuklash uchun foydalaniladi	Xotiraga bir necha marta o'chirib yozish uchun foydalaniladi	Bir marta dasturlanuvchi xotira turi

49	RISC – mikrokontrollerda buyruqlar bajaruvchi ikki pogonali konveer nima yaratadi?	buyruqlarni bir vaqtda tanlash va bajarish imkoniyatini yaratadi	takt chastotani ikki marotaba ko‘tarish imkoniyati	ikki buyruqni parallel bajarish imkoniyati
50	ATMEL kompaniyasi qaysi mikrokontroolerlar ni ishlab chiqaradi	AVR	PIC	STM
51	PIC16F8X guruxi mikrokontrollerlarni buyruqlar xisoblagichlaridagi kichik baytlar qaerda saqlanadi?	PCL registrida	OPTION registrida	PCLATH registrida
52	AVR –MKda qaysi arxitektura qo‘llaniladi?	garvards RISC-protssessori bilan	fon-neyman CISC-protssessori bilan	fon-neyman RISC-protssessori bilan
53	MK yadrosi tarkibiga nimalar kiradi	Protssessor, taktli generator, shina kontrolleri	taktli generator,shina kontrolleri	Arifmetik logik qurilma
54	C++ qaysi dasturlash tili kengaytmasi?	C	Ada	assembler
55	C dasturlash tilida standart va boshqa kutubxonalarni dasturga qo‘shish uchun qanday buyruqdan foydalaniladi?	#include	using	include
56	Mikrokontroller tarkibidpgi ARO‘ nima vazifani bajaradi	Analog signalni raqamli signalga aylantirib beradi	raqamli signalni Analog signalga aylantirib beradi	Signalni diskretlaydi

57	Mikrokontrollr tarkibidpgi RAO' nima vazifani bajaradi	raqamli signalni Analog signalga aylantirib beradi	Analog signalni raqamli signalga aylantirib beradi	Raqamli signallar bilan ishlaydi
58	C dasturlash tilida butun sonlarni e'lon qilish buyrug'i qanday nomlanadi	INT	FLOAT	RD
59	C dasturlash tilida xaqiqiy sonlarni e'lon qilish buyrug'i qanday nomlanadi	FLOAT	INT	RD
60	C dasturlash tilida simvollar bilan ishlash buyrug'i	CHAR	INT	FLOAT
61	C dasturlash tilida musbat sonlar bilan ishlash buyrug'i	UINI	STATIC	PROTEC
62	INT manfiy ishorali sonlarni qabul qiladimi	Ha	Yo'q	Sonlar uchun mo'ljallanmagan
63	Berilgan dastur asosida C portdagi ma'lumotlarni ikkilik kodini aniqlang. <i>Ldi temp, 0x80</i> <i>Out ddrc, temp</i>	10000000	00000001	11001000

64	<p>Berilgan dastur asosida C portdagi ma'lumotlarni ikkilik kodini aniqlang.</p> <p><i>Ldi temp, 0x61</i> <i>Out ddr, temp</i></p>	01100010	10011011	0110 0110
65	<p>Berilgan dastur asosida C portdagi ma'lumotlarni ikkilik kodini aniqlang.</p> <p><i>Ldi temp, 0x28</i> <i>Out ddr, temp</i></p>	00101000	00110111	00100011
66	<p>Berilgan dastur asosida C portdagi ma'lumotlarni ikkilik kodini aniqlang.</p> <p><i>Ldi temp, 0x4E</i> <i>Out ddr, temp</i></p>	01001110	01011011	00000011
67	<p>Berilgan dastur asosida C portdagi ma'lumotlarni ikkilik kodini aniqlang.</p> <p><i>Ldi temp, 0xAA</i> <i>Out ddr, temp</i></p>	10101010	00000011	00101010
68	<p>Berilgan dastur asosida C portdagi ma'lumotlarni ikkilik kodini aniqlang.</p> <p><i>Ldi temp, 0xF6</i> <i>Out ddr, temp</i></p>	11110110	11001001	01110011

69	<p>Berilgan dastur asosida D portdagi ma'lumotlarni ikkilik kodini aniqlang.</p> <p><i>Ldi temp, 0x80</i> <i>Out ddrc, temp</i></p>	10000000	00101010	00000011
70	<p>Berilgan dastur asosida D portdagi ma'lumotlarni ikkilik kodini aniqlang.</p> <p><i>Ldi temp, 0x24</i> <i>Out ddrc, temp</i></p>	00100100	00001110	01000010
71	<p>Berilgan dastur asosida D portdagi ma'lumotlarni ikkilik kodini aniqlang.</p> <p><i>Ldi temp, 0x34</i> <i>Out ddrc, temp</i></p>	00110100	00001011	11000100
72	<p>Berilgan dastur asosida D portdagi ma'lumotlarni ikkilik kodini aniqlang.</p> <p><i>Ldi temp, 0xAD</i> <i>Out ddrc, temp</i></p>	10101101	00001011	00101010
73	<p>Berilgan dastur asosida D portdagi ma'lumotlarni ikkilik kodini aniqlang.</p> <p><i>Ldi temp, 0x16</i> <i>Out ddrc, temp</i></p>	00010111	10010101	11100110

74	Berilgan dastur asosida C portdagi ma'lumotlarni ikkilik kodini aniqlang. <i>Ldi temp, 0x64</i> <i>Out ddrc, temp</i> <i>inc temp</i> <i>out ddrc, temp</i>	01100101	00000111	11100111
75	AVR dasturlash muxitida adreslashning nechta asosiy usuli mavjud	5	3	4
76	R0 registr natijasini aniqlang <i>Ldi R16, 55</i> <i>Ldi R17, 66</i> <i>Mov R0, R17</i>	66	55	121
77	SPH mikrokontroller registrining vazifasi	Stek ko'rsatkichik atta bayt	Stek ko'rsatkichi, kichik bayt	Solishtirish komandasi
78	SPL mikrokontroller registrining vazifasi	Stek ko'rsatkichi kichik bayt	Stek ko'rsatkichika tta bayt	Solishtirish komandasi
79	BYTE qanday buyruqni bajaradi	O'zgaruvchi uchun baytni rezervlaydi	Doimiy baytni belgilaydi	buyruq boshlang'ich adresi
80	CSEG nimani belgilaydi	Kod segmenti	Ma'lumot segmenti	Dastur hisoblagichi
81	INT0 Mikrokontrollerni qanday uzilishini belgilaydi	nol tashqi uzilishlarni so'rovi	Mikrokontroller kirishlari	Mikrokontroller chiqishlari

82	ADC qanday xolatini ko'rsatadi	Analog raqamli o'zgartirish natijalarini yakunlaydi	Analog o'zgartirishni belgilaydi	Raqamli o'zgartirishlarni belgilaydi
83	ANA_COMP qanday vazifani bajaradi	Analogli komparator ishlashini bajaradi.	Mantiqiy ko'paytirish amali	Raqamli sonlarni solishtiradi
84	SPM_RDY qanday xolatini ko'rsatadi	SPM tayyorligi	Dastur boshlanishi	Dastur tamom bo'lishi
85	AND Rd, Rr mnemonikasi nimani bildiradi	ikki UFRB ni «mantiqiy VA» amalini bajaradi	ikki UFRB ni «mantiqiy Yoki» amalini bajaradi	ikki UFRB ni «mantiqiy yo'q» amalini bajaradi
86	NEG Rd mnemonikasi nimani bajaradi	Qo'shimcha kodga o'zgartiradi	Inkor etish kodini bajaradi	Tasdiqlovchi kodni bajaradi
87	SER Rd mnemonikasi nimani bajaradi	UFRB ni barcha bitlarini o'rnatadi	Baytlarni o'rnatadi	Bitlarni o'zgartiradi
88	Berilgan dastur asosida C portdagi ma'lumotlarni ikkilik kodini aniqlang. <i>Ldi temp, 0x80</i> <i>Out ddrc, temp</i>	10000000	00000001	11001000
89	DEC Rd qanday vazifani bajaradi	UFRB ni dekrementlaydi	inkrementlaydi	Chapga suradi
90	LSL Rd qanday vazifani bajaradi	mantiqiy chapga surish	inkrementlaydi	dekrementlaydi
91	MUL Rd, Rr mnemonikasi nimani bajaradi	ishorasiz sonlarni ko'paytiradi	ishorasiz sonlarni bo'ladi	ishorasiz sonlarni qo'shadi

92	ASR Rd qanday vazifani bajaradi	O'nga arifmetik surish	Chapga arifmetik surish	Chapga va o'nga arifmetik surish
93	SEI mnemonikasi qanday bit operatsiyasini bajaradi	Uzilishlarga umumiy ruxsat	Ayirish amali	Uzilishlarga ruxsat bermaydi
94	SEC mnemonikasi qanday bit operatsiyasini bajaradi	Ko'chirish bayrog'ini o'rnatadi	Ko'chirish bayrog'ini o'rnatmaydi	Manfiylik bayrog'ini belgilaydi
95	LDS Rd, k mnemonikasi ma'lumotlarni uzatish qanday buyrug'ini bajaradi	OXQ dan bevosita o'qishni bajaradi	OXQ dan bilvosita o'qishni bajaradi	OXQ dan bevosita o'qishni bajarmaydi
96	PUSH Rr mnemonikasi ma'lumotlarni uzatish qanday buyrug'ini bajaradi	Stekda baytlarni saqlaydi	Stekda baytlarni saqlamaydi	OXQda baytlarni saqlaydi
97	JMP k mnemonikasi qanday boshqaruvni uzatish buyruqlarini bajaradi	Absolyut o'tish	Shartli o'tish	Shartsiz o'tish
98	SR Rd, Rr mnemonikasi qanday boshqaruvni uzatish buyruqlarini bajaradi	UFRB ni solishtiradi	UFRB ni solishtirmaydi	UFRB ni boshqaruv uzatadi

99	SEN mnemonikasi qanday bit operatsiyasini bajaradi	Manfiy ishorali son bayrog'ini o'rnatadi	Manfiy ishorali son bayrog'ini o'rnatmaydi	Musbat ishorali son bayrog'ini o'rnatadi
100	RET Rr mnemonikasi qanday boshqaruvni uzatish buyruqlarini bajaradi	Qism dasturdan	Qism dasturdan qaytmaydi	Qism dasturga murojaat etadi

LABORATORIYA MASHG‘ULOTLARI BO‘YICHA KO‘RSATMA VA TAVSIYALAR

Laboratoriya ishi №1

Mavzu: Ketma-ket interfeysdan foydalanilgan holda Arduinodan arduinoga bog‘lanish

Ishdan maqsad: *serial port*ni ishlash prinsipini o‘rganish va Ketma-ket interfeysdan foydalanilgan holda Arduinodan arduinoga bog‘lanish ko‘nikmalarini xosil qilish.

Kerakli qurilmalar:

- 1.Arduino UNO - 2 dona
- 2.Svetodiodlar
- 3.Bog‘lovchi similar

Nazariy ma‘lumotlar

Serial funksiyasi Arduino bilan kompyuter yoki ketma – ket ma‘lumot almashish interfeysini ta‘minlaydigan boshqa qurilmalar orasida bog‘lanishni hosil qilishga xizmat qiladi. Arduining barcha platalarida hech bo‘laganda bitta keta-ket port mavjud bo‘ladi (UART, Bazida USART deb nomlanadi). Serial ma‘lumot almashish uchun raqamli kiritish/chiqarish 0(RX) va 1 (TX) portlari shuningdek USB portlardan foydalanadi. Serial funksiyalaridan foydalanayotgan vaqtda hech qachon 0 va 1 portlarni boshqa maqsadda foydalanmasilik kerak.

Arduino Mega platasi 3 ta qo‘shimcha ketma-ket portlardan iborat: Serial 1 19(RX) va 18(TX) portlarida, Serial 2 17(RX) va 16(TX) portlarida, Serial 3 15(RX) va 14(TX) portlarida. Ushbu portlardan kompyuterlar bilan bog‘lanish o‘rnatishda foydalanish uchun qo‘shimcha maxsus USB – to – serial adapter kerak bo‘ladi. Chunki ular Mega platasidagi o‘rnatilgan adpterga ulanmagan. Ketma –ket interfeys orqali bog‘lanish o‘rnatish uchun platani TX portiga tashqi qurilmani RX portini, platani RX portiga tashqi qurilmani TX portini ulash lozim.

Ogohlantirish: Ushbu portlarni RS232 portiga to‘g‘ridan to‘g‘ri ulash mumkin emas, plataga zarar yetishi mumkin.

Serial funksiyalari:

- **Serial.begin ();** // bog‘lanishni o‘rnatish
- **Serial.end ();** // bog‘lanishni yopish
- **Serial.available ();** // ketma-ket aloqa interfeysidan o‘qish imkoni mavjud bo‘lgan baytlar soni

- **Serial.read** (); // keta-ket bog‘lanish buferidagi navbatda turgan baytlarni hisoblash
- **Serial.flush** (); // chiquvchi ma’lumotlarni uzatilishini tugashini kutib turish (**Arduino** 1.0+ uchun)
- **Serial.print** (); // ma’lumotlarni ketma-ket port orqali ASCII matn ko‘rinishida uzatish
- **Serial.println** (); // ma’lumotlarni ketma-ket port orqali ASCII matn ko‘rinishida keyingi satrga ko‘chirish belgisi bilan uzatish
- **Serial.write** (); // ushbu funksiya ketma-ket bog‘lanishda ma’lumotalarni ikkilik (binar) ko‘rinishida uzatish
- **Serial.peek** (); // kiruvchi ketma-ket bog‘lanish buferidan keyingi mavjud baytlarni qaytaradi.

Serial.print() ketma-ket port orqali ma’lumotalarni ASCII matn ko‘rinishida uzatadi. Ushbu funksiya turli ma’lumot turlarini qabul qiladi. Butun sonlar ASCII simvollariga muvofiq ravishda chiqariladi. Kasr sonlarni uzatishda butun va kasr qismlarini ifodalash uchun ASCII simvollaridan foydalanadi. Baytlar muvofiq raqamlar simvoli ko‘rinishida uzatiladi. Simvollar va qatorlar shundayligicha uzatiladi.

Misol uchun

- `Serial.print(78)` // "78" ko‘rinishida uzatadi
- `Serial.print(1.23456)` // "1.23" ko‘rinishida uzatadi
- `Serial.print(byte(78))` // "N" ko‘rinishida uzatadi (chunki ASCII jadvalida "N" 78 tartib raqamida joylashgan)
- `Serial.print('N')` // "N" ko‘rinishida uzatadi
- `Serial.print("Hello world.")` // "Hello world." ko‘rinishida uzatadi

Ikkinchi parameter yordamida sonlar uchun sanoq sistemasini o‘zgartirishimiz mumkin. BYTE berilgan raqamga mos ASCII dagi simvolni chiqaradi. Bundan tashqari BIN (ikkilik ko‘rinishida), OCT (sakkizlik ko‘rinishida), DEC (o‘nlik ko‘rinishida) va HEX (o‘n oltilik ko‘rinishida) parametrlari ham mavjud. Kasr sonlarni chiqarishda ikkinchi parameter nuqtadan keyingi kasr qism sonlarni necha xona bo‘lishini belgilaydi.

Misol uchun

- `Serial.print(78, BYTE)` // "N" chiqaradi
- `Serial.print(78, BIN)` // "1001110" chiqaradi
- `Serial.print(78, OCT)` // "116" chiqaradi

- `Serial.print(78, DEC) // "78" chiqaradi`
- `Serial.print(78, HEX) // "4E" chiqaradi`
- `Serial.println(1.23456, 0) //"1" chiqaradi`
- `Serial.println(1.23456, 2) // "1.23" chiqaradi`
- `Serial.println(1.23456, 4) // "1.2346" chiqaradi`

Sintaksis

- `Serial.print(val)`
`Serial.print(val, format)`

Parametrlar

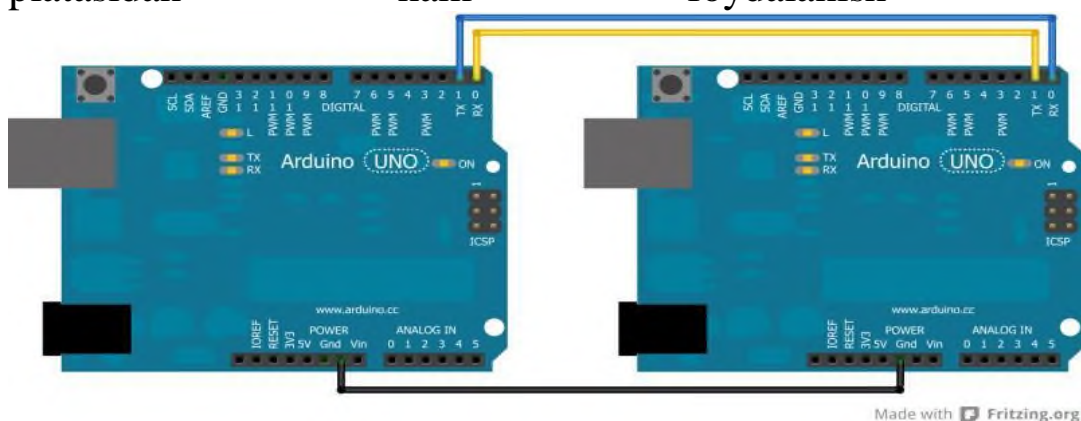
- **val**: ketma-ket bog‘lanishdan uzatiladigan ma’lumot
- **format**: sanoq sistemasi asosi yoki kasr sonni nuqtadan keying belgilar soni

Ketma-ket interfeysdan foydalanilgan holda Arduinodan arduinoga bog‘lanish

Arduino 2 platasidan ma’lumot almashish imkoniyati uning asosiy xususiyatlaridan biri hisoblanadi. Misol uchun bitta Arduino platasi motorni boshqarsa, ikkinchisi sensorlarga ulanadi va boshqaruv signallarini birinchi mikrokontrollerga uzatadi. Ikkita Arduino platasida ma’lumot almashish ketma – ket interfeys (serial) yordamida amalga oshiriladi. Buning uchun RX va TX kontaktlari ishlatiladi.

Ulanish sxemasi

Quyidagi rasmda ikkita Arduino platasini ulanish sxemasi ko‘rsatilgan. Ushbu chizmada ikkita Arduino UNO platasi keltirilgan, agar RX va TX muvofiq kontaktlaridan foydalanilsa, Arduino Mega platasidan ham foydalanish mumkin.



5.1- rasm. Ikkita Arduino UNO platasini ulanish sxemasi

Dasturlash xususiyatlari

Serial interfeysdan foydalanilgan vaqtda, ma'lumotlar baytlarda uzatiladi. Ushbu baytlar ikkinchi Arduino tomonidan bittalab hisobga olinadi. Agar biz simvollar yuborsak, ularni baytga aylantirishda muammo bo'lmaydi. Agarda simvol va baytlarni birdaniga yuborsak, simvol va raqamlar baytda bir xil qiymatni qabul qilishi mumkin. Bu esa ma'lumotni qayta o'qishda muammolar tug'diradi. Bu muammodan qutilshning birdan bir yo'li simvol va raqamlarni bir vaqtda yubormaslik.

Oddiy sketch

Ikkita Arduinoda ma'lumot uzatishni sozlash uchun [Physical Pixel](#) misolidan foydalanishimiz mumkin. Physical Pixel ni yuklash, ushbu misolni Arduino IDE papkasidan topishimiz mumkin: File >> Examples >> Communication.

Ikkinchi Arduino platasiga quyidagi sketchni yuklang:

```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  Serial.print('H');  
  delay(500);  
  Serial.print('L');  
  delay(500);  
}
```

Kod ishlashni boshlasa, Arduinoning 13-pinidagi svetodiod 0.5 HZ chastota bilan yonib o'chadi.

Sketchni ishlashiga yanada ishonch hosil qilish uchun kutish qiymatini (delay) o'zgartirib tekshirib ko'rish mumkin.

“H” sivoli svetodiodning yonishini ta'minlasa, “L” svetodiodni o'chishi uchun javobgar bo'ladi. Boshqa amallar bajarish uchun ushbu ro'yhatni kengaytirish mumkin.

Lekin ushbu kod yetarlicha moslashuvchan emas va qiyinroq va kompleks masalalarni yechish imkoniyatiga ega bo'lmasligi mumkin.

Kengaytirilgan sketch

Arduinodan boshqa Arduinoga ma'lumot uzatish uchun sketchni boshqa usuli bor. Bunda uzatiladigan barcha ma'lumotlar simvolga aylantiriladi. Ikkinchi Arduino platasi ham kelayotgan ma'lumotlarni simvol ko'rinishida qabul qiladi.

Ma'lumot uzatuvchi Arduino uchun sketch. Ushbu sketch ma'lumot uzatuvchi, simvollarni baytga aylantiruvchi va agar zarur bo'lsa raqamni simvolga aylantiruvchi kontroller uchun. Quyida ushbu sketchga misol keltirilgan:

```
const int ledPin = 13; // LED ulangan pin
byte incomingByte;      // kiruvchi ma'lumotlarni o'qish uchun
o'zgaruvchi
void setup() {
  Serial.begin(9600);
  pinMode(ledPin, OUTPUT);
}
void loop() {
  if (Serial.available() > 0) {
    incomingByte = Serial.read();
    Serial.print(incomingByte);
    if (incomingByte == 'H') {
      digitalWrite(ledPin, HIGH);
    }
    if (incomingByte == 'L') {
      digitalWrite(ledPin, LOW);
    }
  }
}
```

Topshiriq: Arduinodan Arduinoga ketma-ket interfeys orqali o'z shaxsiy ma'lumotlaringizni (F.I.SH guruh raqami) uzating.

Nazorat savollari:

1. Arduinoda serial port nima maqsadda ishlatiladi?
2. Serial port funksiyalarini sanang?
3. RX va TX farqlarini sanang?
4. Qaysi Serial funksiyasi ketma-ket bog'lanishda ma'lumotlarni ASCII matn bilan keyingi qatorga o'tish belgisi bilan uzatiladi?

Laboratoriya ishi №2

Arduino uchun RS232 portini ulash

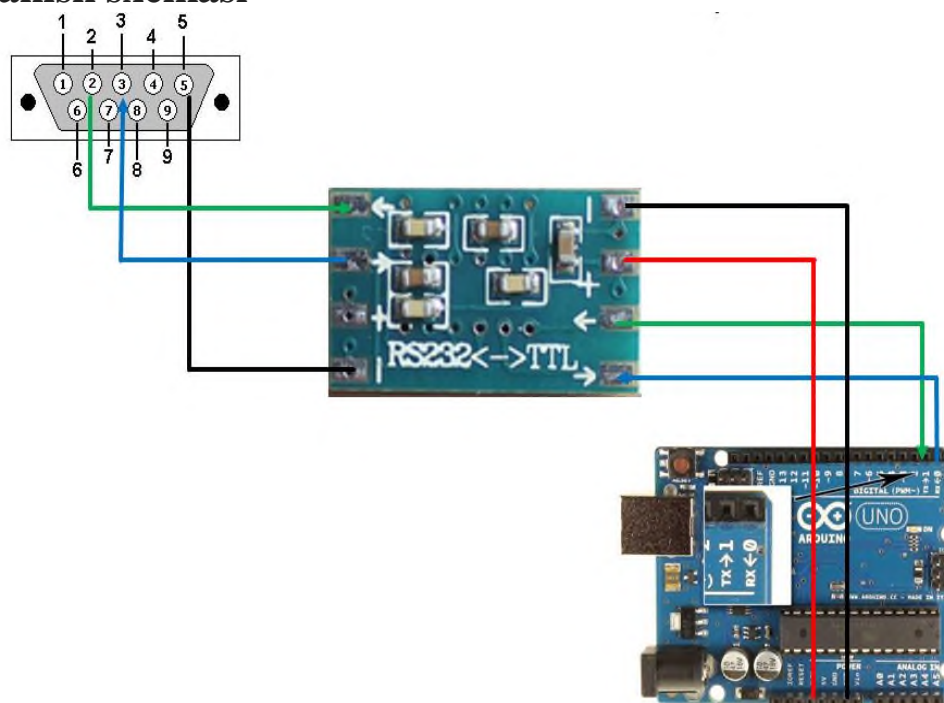
Ishdan maqsad: Ketma-ket RS232 portini o‘rganish va Arduinoga ulashni jarayonini o‘rganish.

Kerak qurilmalar:

1. Kontroller Arduino UNO R3
2. RS 232 seriyali port
3. Ulash uchun similar

Nazariy ma’lumotlar

Ulanish sxemasi



5.2- rasm. Arduino uchun RS232 portini ulash

Seriali Port (Ketma-ket port, COM-port) interfeysi standarti RS-232, shaxsiy kompyuterlar bilan jihozlangan.

Bunda parallel portlardan farqli o‘laroq ma’lumotlar bir ozroq, izchillik bit tomonidan shu port orqali uzatiladi, chunki Port o‘z nomi bilan "keyingi" deb nomlangan. Ba’zida kompyuter interfeyslari (masalan, Ethernet, FireWire va USB) deb ham ataladi axborot izchil yo‘l almashishdan, foydalanishiga qaramay "serial" port RS-232 standarti tayinlangan edi.

ASCII-jadval

Misol uchun, ASCII kodli karta yuklash uchun jadval ko'rsatadi. ASCII da o'nlik raqamlar, lotin va milliy alifbo, tinish va nazorat belgilari ishlatiladi kodlash uchun.

```
int symbol = 33;
void setup() {
  Serial.begin(9600);
  Serial.println("ASCII Table ~ Character Map");}
void loop() {
  Serial.write(symbol);
  Serial.print(", dec: ");
  Serial.print(symbol);
  Serial.print(", hex: ");
  Serial.print(symbol, HEX);
  Serial.print(", oct: ");
  Serial.print(symbol, OCT);
  Serial.print(", bin: ");
  Serial.println(symbol, BIN);
  if(symbol == 126) {
    while(true)
    {
      continue;
    }
  }
  symbol++;
}
```

O'zgaruvchi belgisi kodni saqlaydi. Jadval 33 dan boshlanadi va 126 tugaydi, shuning uchun original o'zgaruvchan belgi sifatida 33 uchun o'rnatiladi.

UART port ish boshlash uchun Serial.begin () funksiyasi hisoblanadi. Uning yagona qobiliyati - bu tezlik bo'ladi.

Asynchronous Transfer Protokol darajasi sifatida oldindan uzatish va qabul qiluvchi tomonida kelishish kerak. Bu misolda, 9600 s darajasi.

Funksiyalari tomonidan ishlatiladigan port uchun qiymatni yozish uchun:

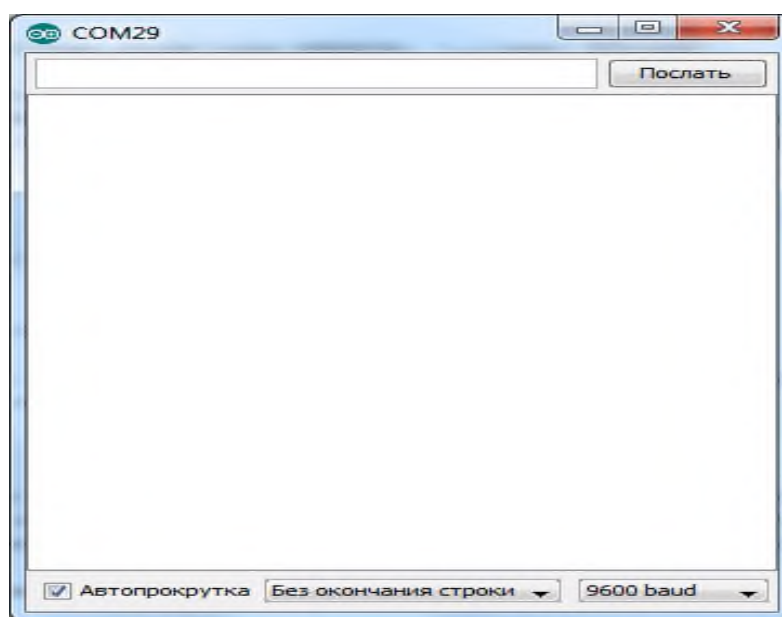
Uch funksiyali funksiya tomonidan ishlatiladigan port uchun qiymatni yozish:

1.Serial.write () - ikkilik shaklda portga ma'lumotlar yozadi.

2.Serial.print () ko'p ma'nolarga ega bo'lishi mumkin, lekin ular inson o'qib ma'lumot shaklida ko'rsatish uchun ishlatiladi. Terminal dasturini namoyishi bu ma'lumotlarni o'zgartirgan holda - axborot uzatish uchun, bir parametr sifatida belgilangandek ta'kidladilar.

Agar hisob bir tizimda bir qiymatni aks ettirish uchun zarur bo'lsa, bu rasmiy so'z qo'shish kerak bo'ladi: bin ikkilik, OKT - sakkiz qirrali, DEC - o'nlik, HEX - hexadecimal.

Misol uchun, Serial.print (25, HEX). Serial.println () va Serial.print () bir xil lekin, print kelgan joyidan davom etaversa println keyingi qatorga o'tkazadi.

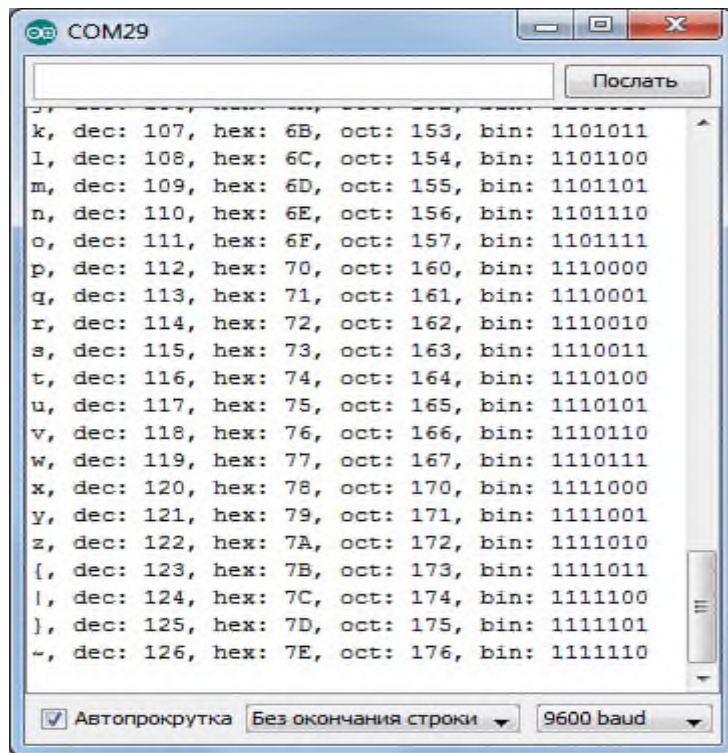


5.3- rasm. Menyudan tanlash

Sinash uchun dastur kompyuteringizdan COM-port terminalda dasturi orqali bor ma'lumotlarni oladi. Uni chaqirish uchun, menyu Vositalar-> Port monitor tanlanadi. Bu qism juda oddiy:

Monitoring porti:

Endi tiklash (перезагрузки) tugmasini bosing. МК qayta tiklanganda ASCII jadvali namoyon:



5.3- rasm. Jadval ASCII

Bu yerda kod qismi eslatma:

```
if(symbol == 126){
    while(true) {
        continue;
    }
}
```

Bu dasturning ijrosini to‘xtatadi. Agar uni bataraf qilib bo‘lsa – jadval abadiy ko‘rsatiladi. Bilimlarni mustahkamlash uchun nomi ketma-ket bo‘lgan port yuborish sekundiga bir marta bo‘ladi va bu sikl doim bir yo‘nalishda harakat qiladi.

Kompyuterlardan buyuruqlar yuborish

Agar bu ishni oldin siz qanday qilib qilish kerakligini bilmogchi bo‘lsa ngiz COM-port haqida ma’lumot olishingiz kerak. Birinchidan barcha kommunikatsiya xotira buferi orqali sodir bo‘ladi. Bunda siz PC qurilmaga signal, belgi yuborganizda, ma’lumotlar xotiraning ba’zi maxsus bo‘limiga joylashtiriladi. Qurilma tayyor holatda deyilganda bir martra – bu bufer ma’lumotlarini o‘qishi hisoblanadi.

Agar buffer boshqaruvi Serial.available () funksiyasini bersa, ushbu funksiya driktev baytning sonini qaytaradi. Bu bayt yordamida serial.read () funksiyasidan foydalanishingiz kerak. Bir misol bilan bu funksiyalarning ishini ko‘rib chiqaylik:

```

int val = 0;
void setup(){
    Serial.begin(9600);
}
void loop(){
if (Serial.available ()>0) ;
    val = Serial.read();
    Serial.print("I received: ");
    Serial.write(val);
    Serial.println();
    }
}

```

Mikrokontroller kodi, ochiq COM-port monitor xotirasiga o'rnatilganidan so'ng, bir belgi kiriting va Enter tugmasini bosing. Ma'lumotlar sohasidagi "Men qabul qilgan X", o'rniga X belgilar kiritilishi qayerda ekanligini ko'rasiz.

Dastur cheksiz magistral ko'chadan aylanadi. Port baytda Serial.available () funksiya yozilgan bir paytda, holati Serial.available ()>0 bo'ladi. Keying Serial.read() funksiyasi shu bilan tozalanadi, ya'ni 1 qiymatini oladi. Shundan so'ng siz allaqachon ma'lim xususiyatlardan foydalangan holda arduinoga o'rnatilgan monitor COM-portdan foyalanishda ayrim cheklovlar hambor.

Mahsulotining COM-portdan foydalanishda ayrim cheklovlar ham bor. Mahsulotining COM-portiga kartadan ma'lumotlar yuborganda har qanday shaklda ajratish mumkin. Agar karta xususiyatini kompyuterdan yuborsangiz, ASCII jadvalga ko'ra sodir bo'ladi. Bu sizga, bunday COM-port orqali xarakterdagi "1", deb (ya'ni "49") ikkilik "00110001" yuborilgan degan ma'noni anglatadi. Ko'p bo'lmagan kodni o'zgartirish bu bayonot tekshirishidir:

```

int val = 0;
void setup(){
    Serial.begin(9600); }
void loop(){
if (Serial.available ()>0) ;
    val = Serial.read();
    Serial.print("I received: ");

```

```

        Serial.println(val, BIN);
    }
}

```

Yuklashdan so‘ng, siz “1” yuborib u uchun monitorda “Men olganman: 110001” javobini ko‘rasiz. Siz boshqa belgilar bilan zaryadni oladigan chiqish formatini va ko‘rinishini o‘zgartirishingiz mumkin.

COM-port orqali qurilma Controli:

Shubhasiz, kompyuterdan buyuruqlar har qanday mikrokontroller vazifalari tomonidan nazorat qilinishi mumkin. LED ishlashini nazorat dasturini yuklab oling:

```

int val = 0;
void setup(){
    Serial.begin(9600);}
void loop(){
if (Serial.available ()>0) ;
    val = Serial.read();
if (val=='H') digitalWrite (13,HIGH);
if (val=='L') digitalWrite (13,LOW);
    }
}

```

Agar siz COM-portda yuborganingizda 'H' belgisining LED yotitishi 13 ni ishlab chiqarish belgisi bo‘lib oladi va sizga 'L' LED yuborib, o‘chirmoqchi bo‘ladi. COM-port ma’lumotlarni olgandan keyin siz magistral halqa sharoitida aniqlash amalga oshirilishi mumkin, turli harakatlarni amalga oshirish uchun magistral liniya ichida dasturiy ta’minot bo‘lishi kerak.

Misol uchun:

```

int val = '0';
void setup(){
    Serial.begin(9600);
}
void loop(){
if (Serial.available ()>0) ;{
    if (val = '1');{
        digitalWrite (13,HIGH); delay (100);
    }
}

```

```

    digitalWrite (13,LOW); delay (100);
}
if (val = '0');
    digitalWrite (13,HIGH); delay (500);
    digitalWrite (13,LOW); delay (500);
}
}

```

Monitor port qiymatini “1” LED 5 hz bo‘ladi, bir chastotada band bo‘ladi. Agar “0” jo‘natsangiz 1 hz bo‘lib chastota o‘zgaradi.

Topshiriq

№	LED'lar soni	Kechikish sek
1	1	200
2	2	100
3	6	500
4	4	2000
5	3	300
6	2	450
7	1	1000
8	4	850
9	2	600
10	6	520
11	4	430
12	1	1000
13	4	670
14	3	780
15	1	1000
16	2	1500
17	3	440
18	4	250
19	2	500
20	1	200

Laboratoriya ishi №3

Mavzu: O'rnatilgan aloqa tizimlarida Bluetooth interfeysi orqali ma'lumot uzatish va qabul qilishni loyihalashtirish.

Ishdan maqsad: Arduino misolida o'rnatilgan aloqa tizimlarida Bluetooth interfeysi orqali ma'lumot uzatish va qabul qilishni loyihalashtirishni o'rganish.

Kerakli qurilmalar:

- Arduino Uno R3
- Ulash simlari komplekti
- HC-06 Bluetooth
- Android bazasidagi telefon (Bluetooth interfeysi bilan)
- Bluetooth client dasturiy ta'minoti

Nazariy ma'lumotlar

Ushbu loyiha sizning telefon apparatingiz orqali masofadan turib boshqarish va ma'lumotlarni uzatib/qabul qilishga mo'ljallangan.



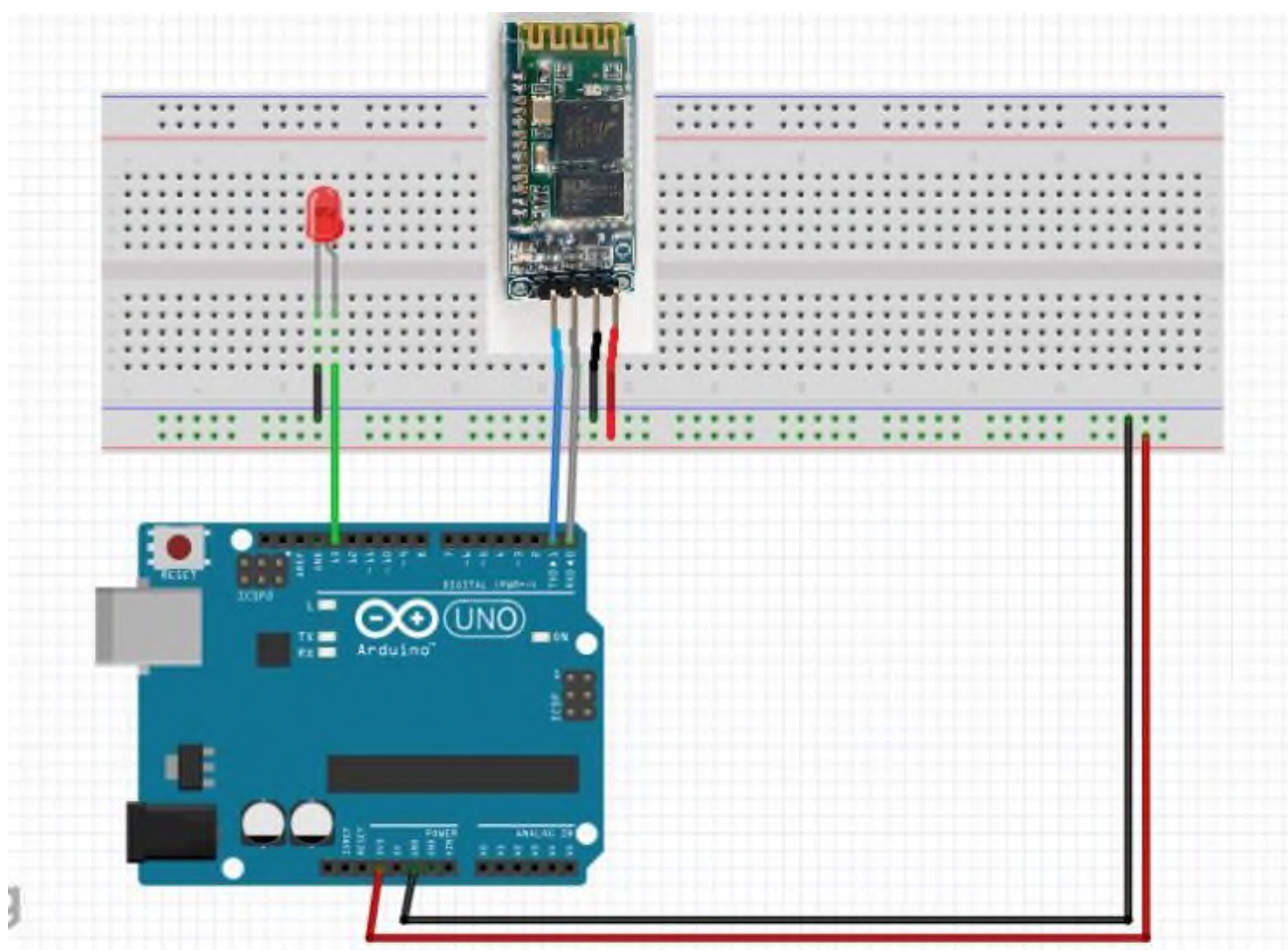
5.4- rasm. Ma'lumotlarni qabul qilish va uzatish

Ma'lumot almashishning eng mashhur usullaridan biri bu Bluetooth.

Bugun biz oddiy misol yordamida Arduinoga Bluetooth modulni qanday qilib ulanishini va Telefon orqali qanday qilib masofadan boshqarilishini ko'rib chiqamiz. Bluetooth modulni Arduino platasiga

ulash maxsus (PAPA-MAMA) simlari orqali juda oson amalga oshadi.

Arduino	Bluetooth
Pin 1 (TX)	RXD
Pin 0 (RX)	TXD
GND	GND
5V	VCC



5.4- rasm. Bluetooth interfeysi orqali ma'lumot uzatish

Quyidagilarni ulanishiga diqqatingizni qarating: $TX \rightarrow RXD$, $RX > TXD$.

Endi sinov kodini yozib ko'ramiz: Yuklash chog'ida Bluetooth modul Arduino platasidan uzilgan bo'lishi kerak. Agar ulangan

bo'lsa dastur yozilmaydi chunki Bluetooth modul bilan aloqa bitta port bilan bo'lib qoladi. RX va TX, USB uchun.

```
int val;
int LED = 13;
void setup()
{
  Serial.begin(9600);
  pinMode(LED, OUTPUT);
  digitalWrite(LED, HIGH);
}
void loop()
{
  if (Serial.available())
  {
    val = Serial.read();
    // При символе "1" включаем светодиод
    if (val == '1')
    {
      digitalWrite(LED, HIGH);
    }
    // При символе "0" выключаем светодиод
    if ( val == '0')
    {
      digitalWrite(LED, LOW);
    }
  }
}
```

Dastur kodini yuklab bo'lgandan keyin Bluetooth modulni Arduinoga ulash mumkin.

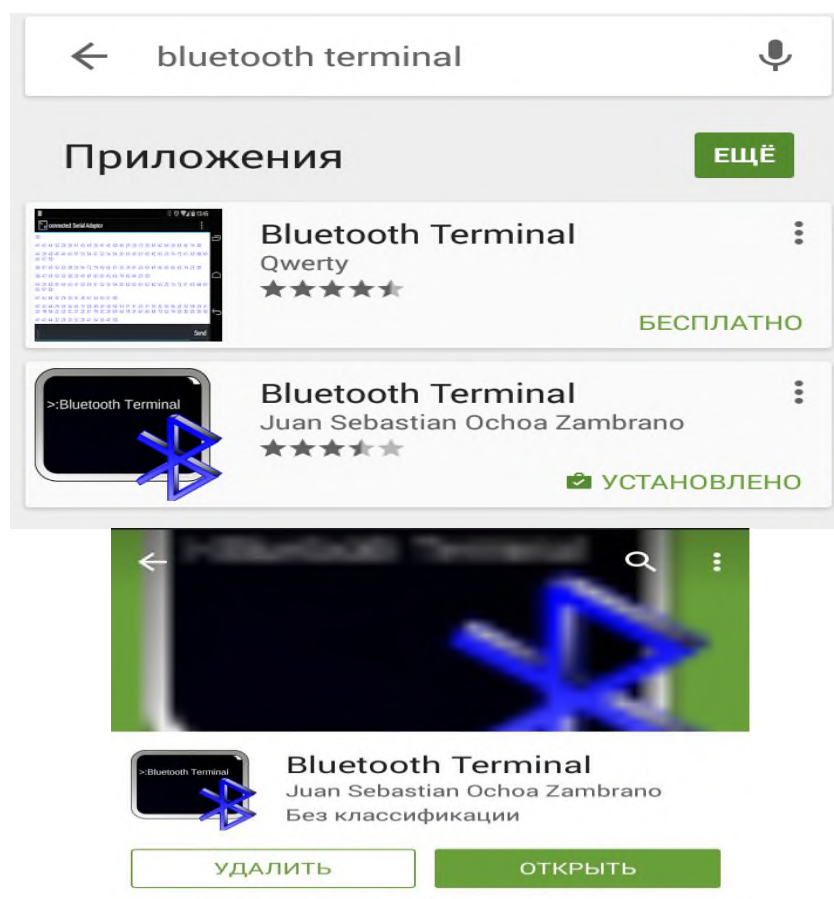
Keyingi qadam Bluetooth ni telefonga ulash. 1.Telefondan Bluetooth ni yoqamiz va yangi qurilmani qidiramiz

2.Topilgan ro'yhatda "HC-06" qurilmani nomi chiqadi va unga ulanamiz

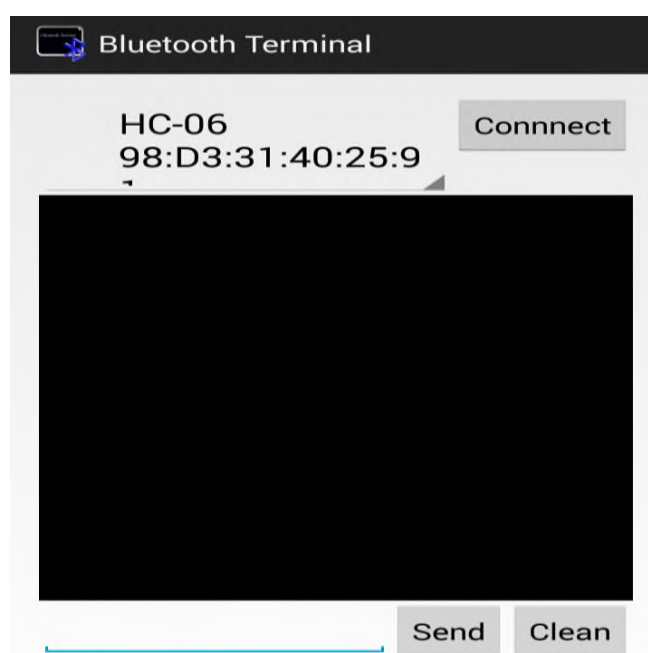
3.Telefon Pin-Code so'raydi. Shunda "1234" yoki "0000" ni teramiz.

4.URRRRRRAAAA!!!. Qurilma ulandi.

Endi sizning telefon qurilmangizga bluetooth terminal ni yuklab olish kerak. Biz Android platformada qarab ko'ramiz.



Biz har xil bluetooth terminallarni sozlashimiz mumkin va ular faqat dizayni bilan farq qiladi vazifasi esa bir xil. Bundan keyin biz Bluetooth modulni ishga tushiramiz.



Qoʻllash uchun namuna: Bluetooth modul Arduino platasiga Bluetoothni oʻrnatish uchun ishlatiladi. Ish rejimi Passiv. Ishlash

uchun shunchaki boshqa qurilmani qidiramiz va unga ulanamiz xuddi NS-06 kabi. Bazaviy pini 1234.

Arduinoga ulagandan keyin ushbu Bluetooth port rolini o'ynaydi, biz ketma- ket portdan junatamiz va bu jarayonda Arduino jo'natib qabul qiluvchi port orqali muloqot qiladi.

Pinlari:

STATE – Svetodioddan signal qabul qilish.

RXD – kontakt ma'lumot qabul qilish uchun. Biz undan ma'lumot yuboramiz.

TXD – kontakt ma'lumot uzatish uchun.

GND – yer (minus “zemlya”)

VCC – Ta'minot 3.3 v

EN – Ta'minot yoqish\o'chirish, agar bir bersak modul yonadi. Agar modulga bir junatsak o'chadi.

Sozlash: Bazaviy sozlanmalarini o'zgartirish uchun nomi, pini, modulni perexodnik orqali PC ga ulash kerak va PC orqali AT buyrug'i bilan sozlamalarni o'zgartirish mumkin. Bu vaziyatda biz uni ko'rmaymiz shu sababli bizga perexodnik kerak bo'lmaydi.

O'zgartirish uchun buyruqlar ketma-ketligi:

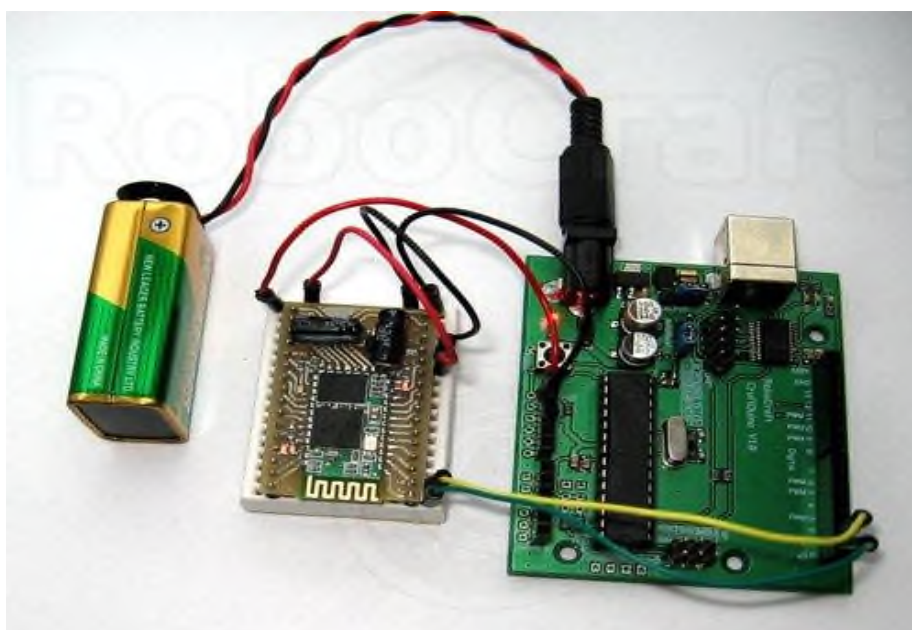
AT - Qaytadi OK

AT+VERSION - Versiyasini ko'rish

AT+NAMEyyy – Qurilmani nomini o'zgartirish.

AT+PINxxxx – boshqa pin berish, bazaviy pin 1234

AT+BAUDx – tezlikni o'rnatish.



Keltirilgan misolda telefondan “1”ni 13 portdan junatyapmiz va svetodiod yonyapti “0”ni junatganimizda uchadi. Svetodiod 1sekund intervali bilan yonib o‘chadi.

```
enum { LED_PIN = 13 };
enum LedState
{ LED_ON, LED_OFF, LED_BLINK };
LedState led_state;
void setup(){
  led_state = LED_OFF;
  pinMode(LED_PIN, OUTPUT);
  pinMode(3, OUTPUT);
  Serial.begin(9600);
}
void loop(){
  if (Serial.available())
  {
    char command = Serial.read();
    switch (command)
    {
      case '1':
        led_state = LED_ON;
        Serial.println("Led On Cool");
        break;
      case '0':
        led_state = LED_OFF;
        Serial.println("Led Off Fall");
        break;
      case '*':
        led_state = LED_BLINK;
        Serial.println("Led Blink ");
        break;
      default:
        {
          for (int i = 0; i < 5; ++i)
          {
            digitalWrite(LED_PIN, HIGH);
            delay(50);
            digitalWrite(LED_PIN, LOW);
```

```

    delay(50);
  }
  Serial.println("Fail ....");
}
}
}
switch (led_state)
{
  case LED_ON:
    digitalWrite(LED_PIN, HIGH); break;
  case LED_OFF:
    digitalWrite(LED_PIN, LOW); break;
  case LED_BLINK:
    {
      static unsigned long start_millis = 0;
      if (millis() - start_millis >= 300)
      {
        start_millis = millis();
        digitalWrite(LED_PIN, !digitalRead(LED_PIN));
      }
    }
}
}
}
}

```

Laboratoriya ishi №4-5 O‘rnatilgan Ethernet tarmoqlarida ma’lumot uzatish

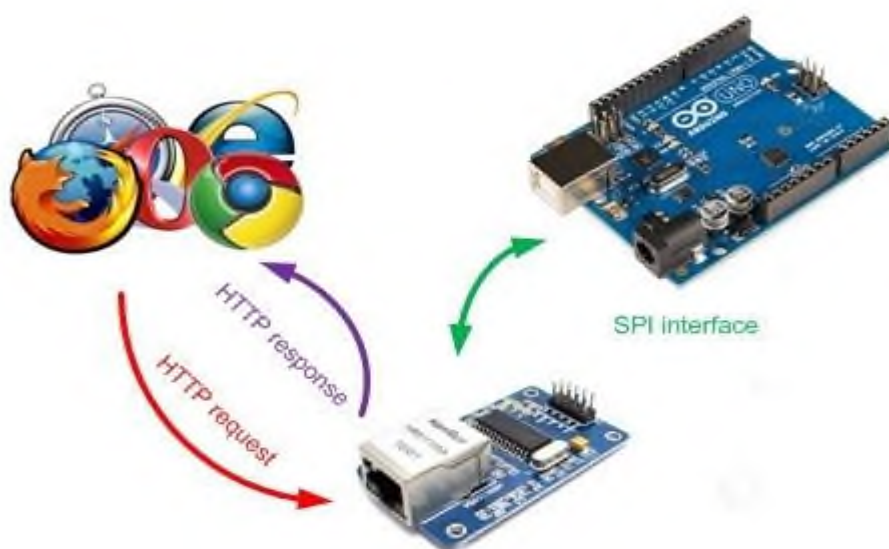
Ishdan maqsad: Ethernet modulini o‘rganish, uni ishlash prinsipi va Ethernetni aloqaning har xil yo‘nalishlarida sinab ko‘rish

Kerakli qurilmalar:

1. Ethernet moduli
2. Arduino platasi
3. kabellar
4. svetodiotlar
5. potensiometr
6. rezistor
7. 10BASE-T, 100BASE-T(x) i 1000BASE-T(x) standartidagi Ethernet tarmoqlarida ishlatiladigan 8P8C tipidagi UTP kabellari.

Nazariy ma'lumot

Arduino uchun ENC28J60 Ethernet modul - bu modul elektr priborlarni internet orqali boshqarish uchun mo'ljallangan qurilmalarni yig'ish va sayt sahifasida datchiklardan kelayotgan ma'lumotlar ko'rinib turish uchun mo'ljallangan. Sayt orqali priborlarni boshqarish va komandalarni bajarilganini tekshirish mumkin. Masalan: qadamli dvigael yordamida videokameralarni yon tomonga burishni va PK ekranida videosyomka ko'rinib turishini ishlab chiqish mumkin. Modul TCP/IP local tarmog'ida va internet tarmog'ida ishlaydi. ENC28J60 moduli imkoniyatlari va mikrokontroller dasturidan foydalangan holda Web-server yaratishadi. Modul MKni SPI interfeysi orqali TCP/IP tarmog'i bilan ulaydi. ENC28J60ga bog'langan Avtomatika qurilmalarini boshqarish internet orqali bir xil vaqtda yoki ketma-ket bit nechta



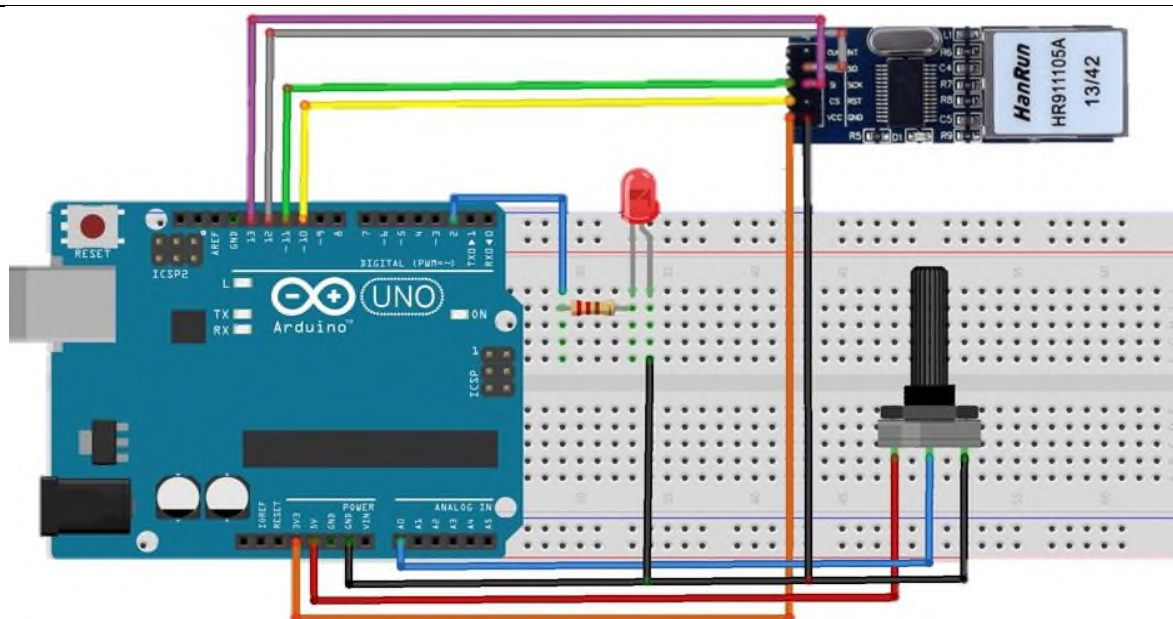
operator komyuterlari yoki mobil qurilmalari orqali amalga oshiriladi. ENC28J60 Ethernet modul Arduino uchun nafaqat avtomatika qurilmalarining bir qismi sifatida ishlatiladi, balki TCP/IP tarmog'i va Web-serverning tuzilishini o'rganishda ko'rinarli misol sifatida qo'llaniladi. Ko'rinuvchi komponentlar - ENC28J60 mikroshema. Platada TCP/IP tarmoq kabellari uchun razedkalar joylashtirilgan rozedka transformatorlarni TCP/IP tarmoqlardan modulning galvanichesiy yechimlarini ta'minlash va 2 ta yonib turuvchi svetodiod qaysiki ma'lumot almashayotganidan dalolat beruvchi. 25 MGs li kvarsniy rezonatorga ega. Vilka 10 kontaktli MK ga ulanadi. Quvvat ko'rsatgich. Android uchun ENC28J60 Ethernet modul Raspberry Pi yoki boshqa MK qurilmasi rejimida ishlashi mumkin. U MK bilan

yaxshi bog‘lanadi va bir xil urevendagi kuchlanishni oladi. MK dan faqat paketlarni uzatish va SPI interfeysi orqali kuchlanish qabul qilish talab etiladi.

Xarakteristika:

- kuchlanish 3,14 – 3,45 V
- tok kuchi , mA maksimalniy 250 normal 170
- MAC va 10 Base-T PHY ning birlashishi - polniy yoki poludupeks
- 10 Base-T portini qutblanishni avtomatik tanlovchi va korreksiya bilan quvvatlash
- kontrollangan summani avtomatik generatsiya qilish
- hatolik pakelarini avtomaik tashlab yuborish
- oshibka paketlarni qaytadan yuborish programma funksiyasi
- Padding va CRC geneatsiya qiladigan programma funksiyasi
- hatolik paketlarni filtrlovchi programma funksiyasi
- SPI takt chastotasi- 20 MGs gacha
- Plata razmeri: 55 x 36 mm

ENC28J60 modul	Arduino Uno
VCC	3.3V Pin
GND	GND Pin
SS (CS)	Pin 10
MOSI (SI)	Pin 11
MISO (SO)	Pin 12
SCK	Pin 13



4.1- rasm. Ethernet ENC28J60 va Arduino orasidagi bog‘lanish

Qayd etish joizki bu ulanishlarda aniq Arduino Uno yoki ATmega328 (Arduino Mini Pro, Arduino Nano va h.k.) uchun ENC28J60 moduli SPI interfeyslari ishlatiladi, shunday ekan agarda siz Mega lar bila ishlangiz unga mos SPI kontaktlarni tanlashingiz kerak.

Misol uchun: Ethernet ENC28J60 modul uchun programma. Siz bu modulni programma qilyotganizda **Arduino IDEda Ethercard bibliotekasidan** yuklab olish va qabul qilish uchun kerak bo‘ladi Bibliotekadan so‘ng biz keying dasturimizni yozamiz:

```
#include <EtherCard.h>
static byte mymac[] = {0xDD,0xDD,0xDD,0x00,0x01,0x05};
static byte myip[] = {192,168,1,177};
byte Ethernet::buffer[700];
const int ledPin = 2;
char* StatusLed="OFF";
void setup () {
  Serial.begin(9600);
  Serial.println("Laboratory Module ENC28J60");
  if (!ether.begin(sizeof Ethernet::buffer, mymac, 10))
  Serial.println( "Ne udalos poluchit dostup k kontrolleru Ethernet ");
  else
  Serial.println("Ethernet Controller initsializirovan ");
  if (!ether.staticSetup(myip))
  Serial.println("Ne udalos ustanovit IP - adres ");
  Serial.println();
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW); }
static word homePage() {
  BufferFiller bfill = ether.tcpOffset();
  bfill.emit_p(PSTR("HTTP/1.0 200 OK\r\n"
    "Content-Type: text/html\r\nPragma: no-cache\r\nRefresh: 5\r\n\r\n"
    "<html><head><title>Naylamp Mechatronics</title></head>"
    "<body>"
    "<div style='text-align:center;'>"
    "<h1> Laboratory Module ENC28J60</h1>"
    "Delay Encoder : $L Milliseconds"
    "<br /><br />STATUS LED: $$<br />"
    "<a href='/?status=ON'><input type='button' value='ON'></a>"
    "<a href='/?status=OFF'><input type='button' value='OFF'></a>"
  ));
}
```

```

    "<br /><br />Encoder: $D (MAX 1024)"
    "<br /><br />"
    "</body></html>" ),
millis()/1000,StatusLed,analogRead(0));
    return bfill.position();}
void loop() {
    word len = ether.packetReceive();
    word pos = ether.packetLoop(len);
    if(pos) {
if(strstr((char *)Ethernet::buffer + pos, "GET /?status=ON") != 0) {
    Serial.println("Comando ON recibido");
    digitalWrite(ledPin, HIGH);
    StatusLed = "ON"; }
if(strstr((char *)Ethernet::buffer + pos, "GET /?status=OFF") != 0) {
    Serial.println("Comando OFF recibido");
    digitalWrite(ledPin, LOW);
    StatusLed= "OFF"; }
    ether.httpServerReply(homePage()); }
}

```

Asosiy kod:

```

staticeskii bayt mymac [] = {0xdd, 0xdd, 0xdd,
0x00,0x01,0x05};

```

```

staticeskii bayt MyIP [] = {192,168,1,177};

```

Biz bu yerda o‘zimizning MAC va IP adreslarini o‘rnatib va xohlagan qiymatni yoza olamiz ammo tarmoqqa ulanishda qurilmalarda bir xil MAC yoki IP bo‘lib qolmasligi kerak. **статического слова домашней страницы ()** funksiyasida biz o‘zimizning HTML ko‘rinishidagi vebsaytimizni yozamiz:

```

статическое слово домашней страницы () {
    BufferFiller ether.tcpOffset bfill = ();
    bfill.emit_p (PSTR ( "HTTP / 1.0 200 OK\r\n"
    "Content-Type: Текст / html\r\nPragma: без cachernRefresh: 5\r\n"
    "<HTML> <HEAD> <название> Naylamp Мехатроника </
title> </ голова>" " <тело> "
    " <DIV стил = \"выравнивания текста: центр;\"> "
    " <h1> Модул тестирования ENC28J60 </ h1> "

```



```

" Время истекло: $ L секунды "
" <br /> <br /> Государство LED: S $ <br /> "
" <a href= \"/?status=ON\"> <входной тип = \"кнопка\" значение
= \"на\"> </a> "
" <A HREF = \"/? состояние = OFF \"> <входной тип = \"
\"значение =\" OFF кнопка \"> </a>\"
" <br /> <br /> Потенциометр: $ D (1024)\"
" <br /> <br /> "
" <a href='http://www.naylampmechatronics.com/'>
www.naylampmechatronics.com </a> "
" </ body> </ html> "
милли секунд () / 1000, StatusLed, analogRead (0));
вернут bfill. положение ();}

```

E'tibor qarating o'zaruvchi ma'lumotlar yuborilganda \$ L \$ S \$ D markerlari ishlatiladi undan keyin uzun va tog'ridan- to'g'ri ma'umotlar uzatiladi. Biz browser ham o'rnatdik negaki har 5 sekunda stranitsa yangilanib tursin, bu "Refresh: 5" bilan bajariladi, sizga malumot-doimiy o'zgarissni ko'rsatuvchi qonun kerak. Yangilanish vaqti ma'lumotlar tipiga bog'iq. Keyingi solishtirish, agarda ma'lumotlar brauzerdan GET metodi yordaida jo'natilgan bo'lsa.

```

если (strstr (( символ *) Ethernet :: буфер + поз, "GET /? статус
= ON" ) != 0) {
Серийный . Println ( "ПОЛУЧИЛ по команде" );
digitalWrite (ledPin, ВЫСОКИЙ );
StatusLed = "ON" ; }

```



Topshiriq

№	Encoder uchun port	LED uchun portlar	Svetodiodlar soni
1	A1	2	1
2	A2,A4	3,8	2
3	A3,A5,A1,A2,A0,A4	2-8	6
4	A1,A3,A4,A5	2,4,6,7	4
5	A2,A0,A3	6,7,8	3
6	A4,A0	2,9	2
7	A0	2	1
8	A1-A4	3-7	4
9	A0,A5	8,9	2
10	A5,A4,A2,A1,A0,A3	3-9	6
11	A0-A3	4-8	4
12	A0	2	1
13	A5,A3,A0,A2	2-6	4
14	A3,A4,A5	7,8,9	3
15	A0	2	1
16	A4,A3	5,7	2
17	A5,A0,A1	4,5,6	3
18	A2,A3,A4,A5	1,5,6,8	4
19	A1,A3	8,9	2
20	A5	5	1

Nazorat savollari

1. Ethernet nima?
2. Ethernet ishlash prinsipi va ishlatilish sohasi?
3. Ethernet modulining xarakteristikasi?
4. Arduino va Ethernet ulanish sxemasi?

Laboratoriya ishi №6, №7

O‘rnatilgan ZigBee tarmoqlarda ma’lumotlar uzatish

Ishdan maqsad: ZigBee modulini o‘rganish, ishlash prinsipi va turli aloqa tarmoqlarida ZigBee foydalanish.

Kerakli qurilmalar:

- Arduino Platasi 2ta
- XBee 2 ta
- XBee shield 2 ta
- Kabel USB 1 ta
- 8 batareyali ko‘rinishda Arduino elektr ta’minoti AA.
- Rangli diodlar 2ta
- Simlar
-

Nazariy ma’lumotlar

ZigBee - yuqori darajadagi tarmoq protokollari spetsifikatsiyasi- ilova darajasi APS (eng. application support sublayer) va tarmoq darajasi NWK- pastki darajadagi xizmatlardan foydalanish – fizik darajada Mac dan foydalanish. ZigBee spetsifikatsiyasiga ko‘ra u ilovalarga yo‘naltirilgan, yuqori bo‘lmagan tezlikda ma’lumotlarni xavfsiz yuborish.




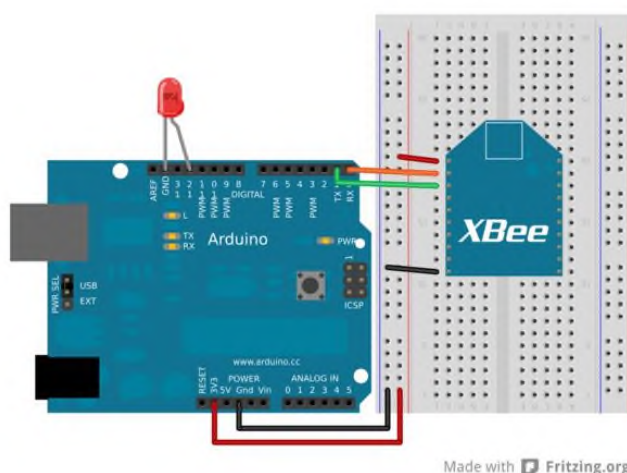
ZigBee qurilmasi

Italiya kompaniyasi Telit tomonidan ishlab chiqilgan ZigBee standart moduli, bu chipning TI dan CC2430 standartida tuzilgan.

Ushbu modul $21 \times 14,2 \times 2,2$ mm o‘lchamlarda tashkil etiladi.

Telit kompaniya moduli

Modul	Tuzilishi	Rasm
ZE50	1 mW chiqish quvvati; -92 Dbm befarqligi; 70 m gacha oralig'ida.	
ZE60	100 mW chiqish quvvati (ichki kuchaytirgichi); -98 DBm befarqligi; qadar 800 m diapazoni.	
ZE51	2,5 mW chiqish quvvati; -97 DBm befarqligi; takomillashtirilgan RF-qismi; 1000 m gacha oralig'ida.	
ZE61	100 mW chiqish quvvati (ichki kuchaytirgichi); -99 DBm befarqligi; takomillashtirilgan RF-qismi; up 4000 m oralig'i.	

**Misol**

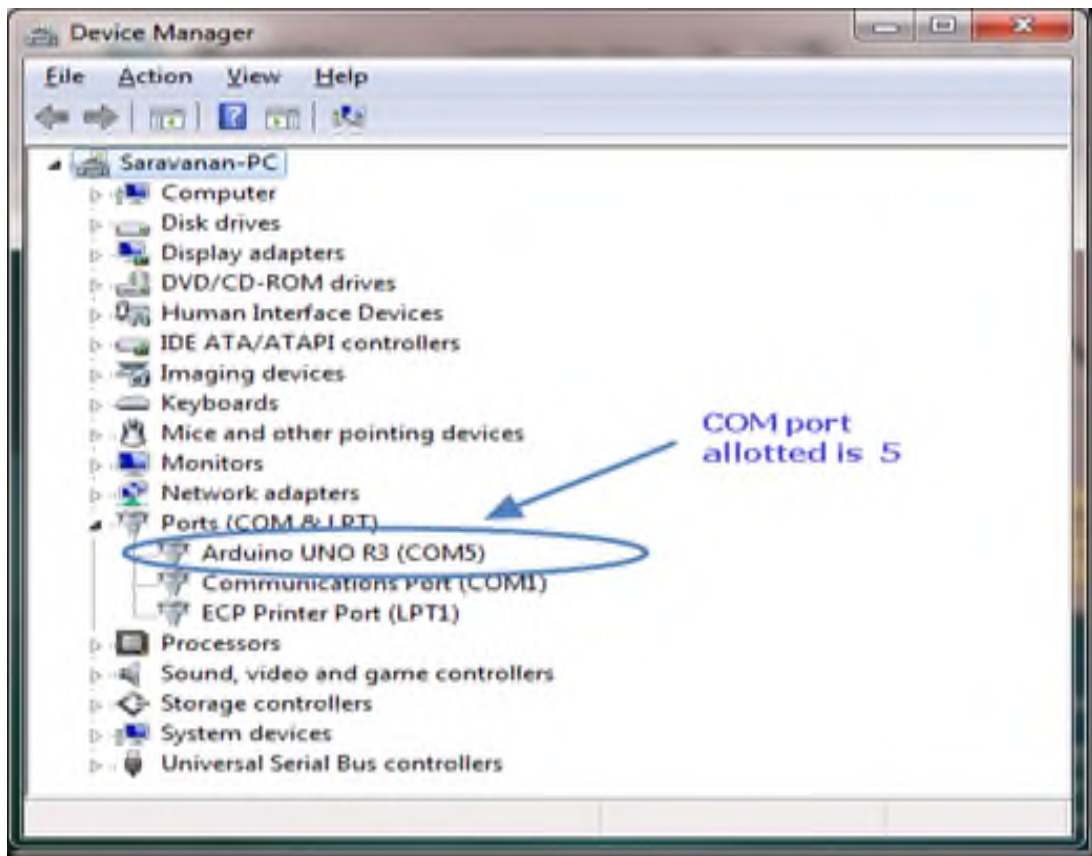
2 x Arduino UNO doska

2 x 1 XBee Series Radio

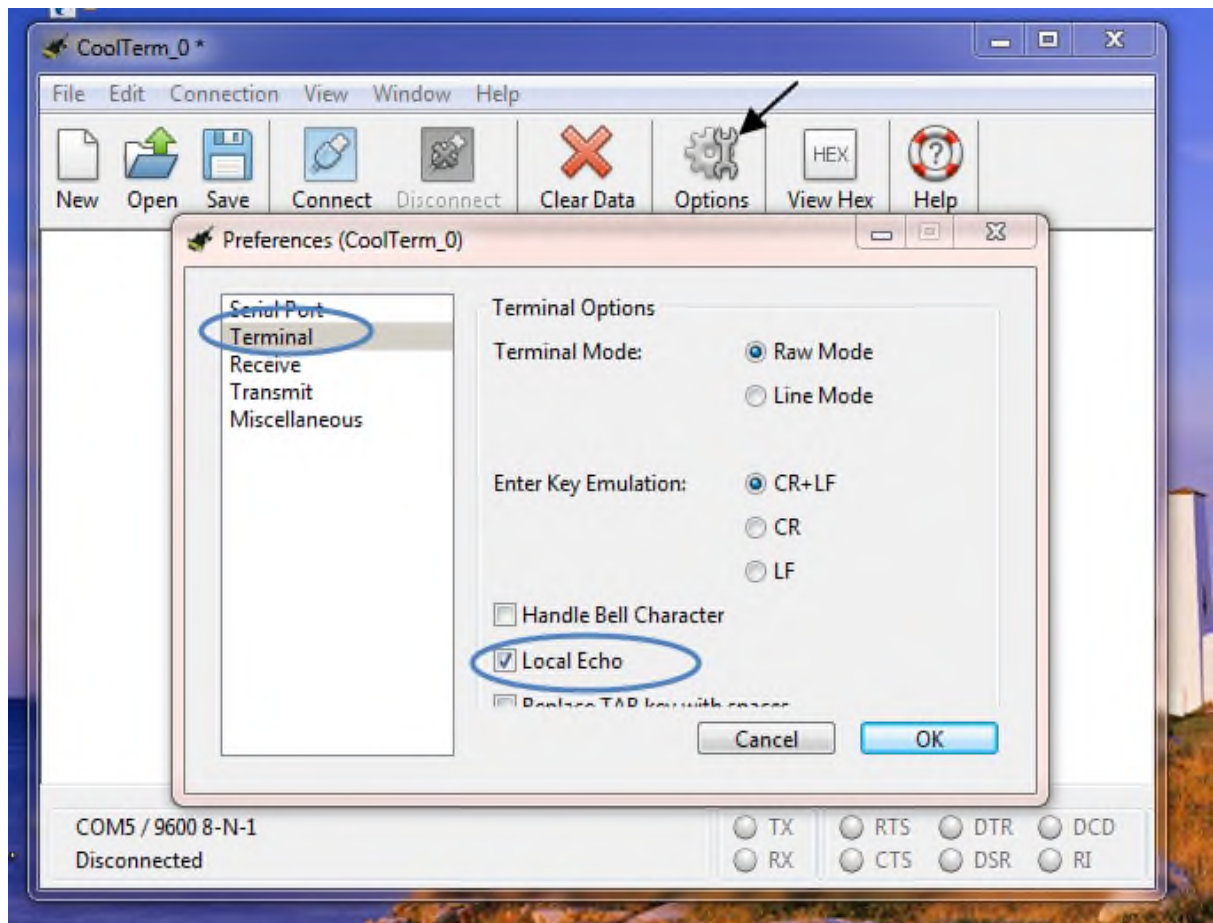
XBee uchun 2 x Breakout Board& Kabel, USB-kabel AB, LED, mini zummer.

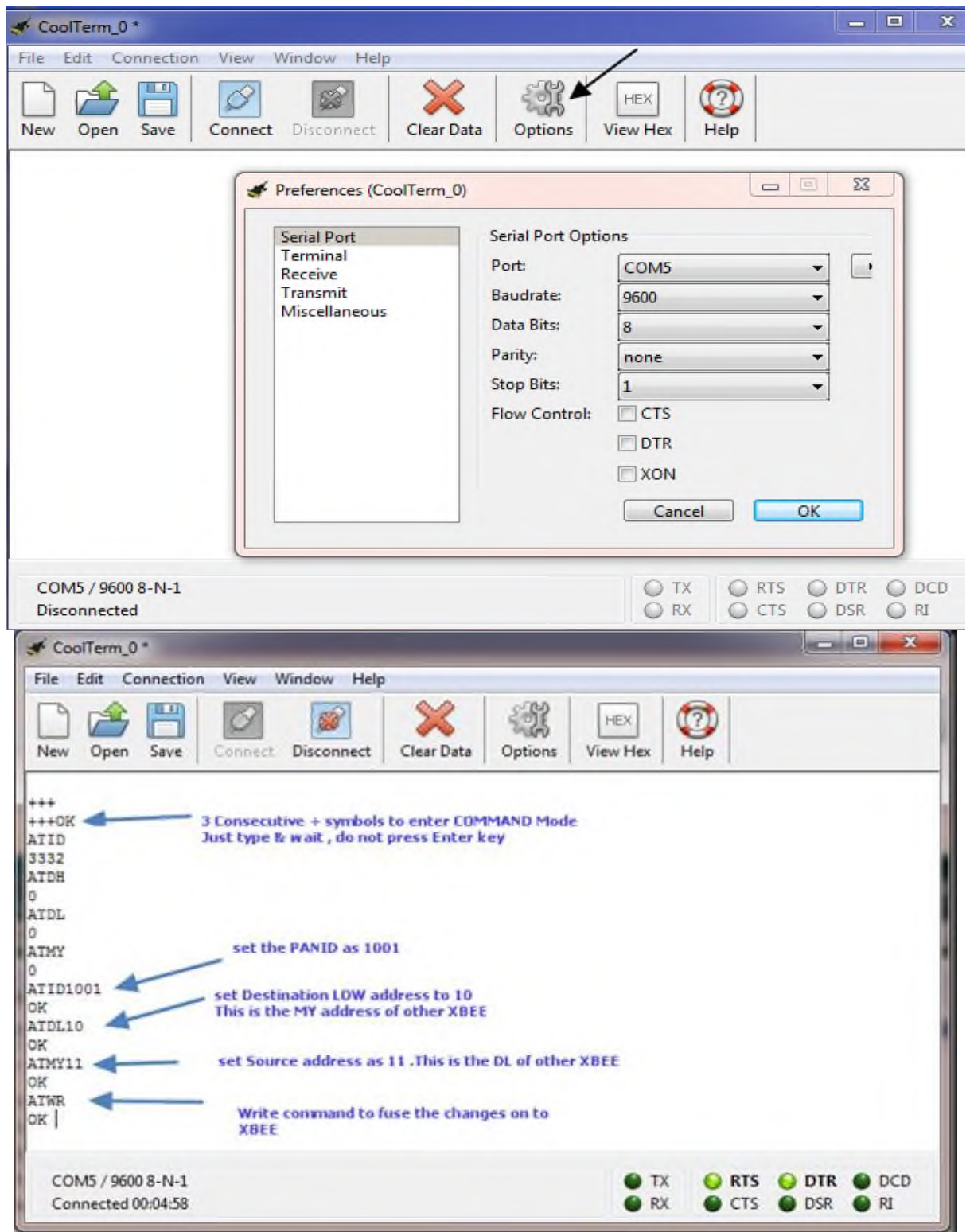
Birinchi bo'lib 1ta Xbee Radio seriasini konfiguratsiya qilamiz. Arduino boardning bittasidan foydalanamiz.

COM-port sizning kompyuteringiz qurilma menejeri orqali o'zgartiring chunku u Arduino Boarddan himoyalangan bo'ladi.



So'ng rasmdagidek o'zgartiring





Ikkinchi konfiguratsiya Xbee

Quyida ko'rsatilgandek, sozlash tanlang:

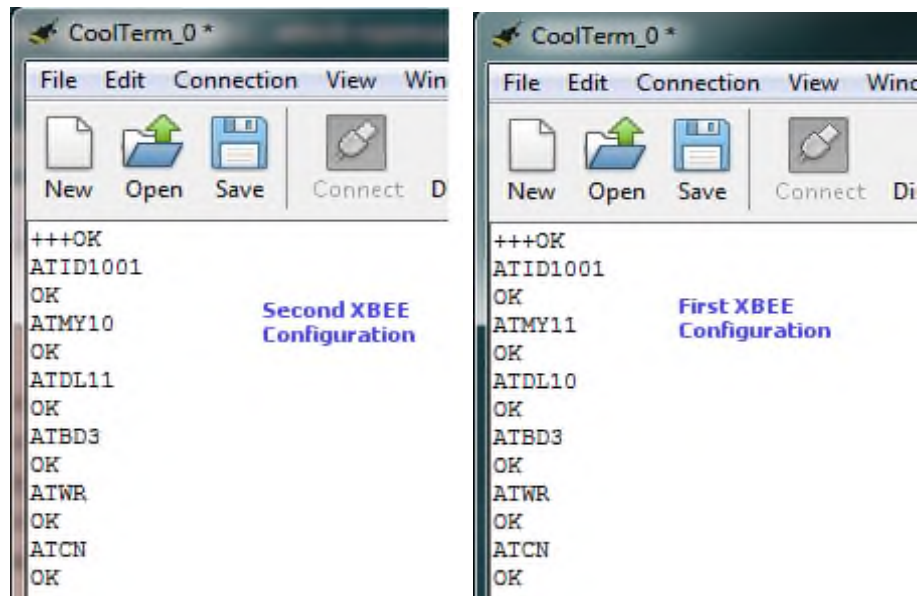
ATID1001 - birinchi XBee sifatida PAN ID Shu.

ATDL11 - joy Low birinchi My XBee manzili bo'lgan, 11 bo'ladi.

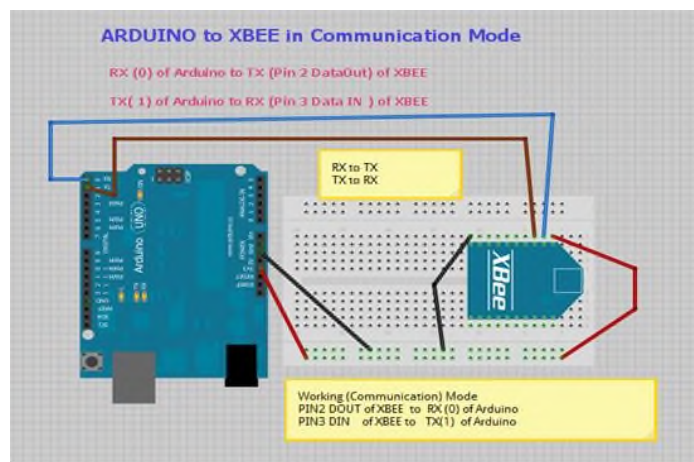
ATMY10 - Source URL sozlanishni u 10.

ATBD3 - oʻrnatish tezligi 3 (9600)

ATWR - oʻzgarishlarni roʻyxatdan uchun buyruq yozing.



XBEEs Arduino Wireless ulanishni amalga oshiramiz.



Joʻnatish Side Code ARDUINO "A"

-----/ * Bu dastur ketma-ket port boshlab va kapital "K" va soʻngra kutishlar ramzi yuboradi ismlar va «O qisqa vaqt ichida PIN 13 LED yoqish uchun */

```
Char readChar;
```

```
Void() {
```

```
pinMode (13, OUTPUT);
```

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

```
delay (500);
```

```
Serial.print ( 'K ');
```

```
delay (500); }
```

```

Loop() {
if (Serial.available ()> 0) {
    readChar = Serial.read ();
    if (readChar == 'O') {
        digitalWrite (13, HIGH);
        delay(5000);
        digitalWrite (13, LOW);    }
    }
}

```

Qabul qiluvchi tomon kodi Arduino "B"

/* Ushbu kod ovozli signal ketma-ket port va kommutator orqali kapital belgi "K" kutmoqda

1,5 uchun 11 PIN ulangan secs. Keyin u ketma-ket port orqali ko'rsatuvchi "O" yuboradi

Arduino A jo'natuvchi tomonida LED chiroqlar */

Int alarm = 11;

Void(){

pinMode (alarm, OUTPUT);

Serial.begin (9600); }

Loop(){

if (Serial.available ()> 0) {

if(Serial.read () == 'K') {

digitalWrite (alarm, HIGH);

delay (3000);

digitalWrite (alarm, LOW);

delay (1000);Serial.print ("O ");

}

}

```

sketch_feb26a5 | Arduino 1.0.3
File Edit Sketch Tools Help
sketch_feb26a5
char readChar;
void setup()
{
  pinMode(13,OUTPUT);
  Serial.begin(9600);
  delay(500);
  Serial.print('K');
  delay(500);
}
void loop()
{
  if (Serial.available() >0){
    readChar=Serial.read();
    if(readChar=='O'){
      digitalWrite(13,HIGH);
      delay(5000);
      digitalWrite(13,LOW);
    }
  }
}
Done uploading.
Binary sketch size: 2,430 bytes (of a 32,256 byte maximum)

```



```

sketch_feb26a 5
int Buzzer=11;
void setup()
{
  pinMode(Buzzer, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  if (Serial.available()>0) {
    if (Serial.read()=='K') {
      digitalWrite(Buzzer,HIGH);
      delay(3000);
      digitalWrite(Buzzer,LOW);
      delay(1000);
      Serial.print('0');
    }
  }
}

```

Done uploading.
Binary sketch size: 2,416 bytes (of a 32,256 byte max)

Topshiriq

No	LED uchun port	Svetadiodlar soni	Kutish vaqti msek
1	2	1	200
2	3,8	2	100
3	2-8	6	500
4	2,4,6,7	4	2000
5	6,7,8	3	300
6	2,9	2	450
7	2	1	1000
8	3-7	4	850
9	8,9	2	600
10	3-9	6	520
11	4-8	4	430
12	2	1	1000
13	2-6	4	670
14	7,8,9	3	780
15	2	1	1000
16	5,7	2	1500
17	4,5,6	3	440
18	1,5,6,8	4	250
19	8,9	2	500
20	5	1	200

Nazorat savollari:

1. ZigBee nima?
2. Zigbee tamoyili?
3. ZigBee ning ulanish sxemasi?

Laboratoriya ishi №10, №11

Arduino platasiga SIM800L GSM modulini ulash

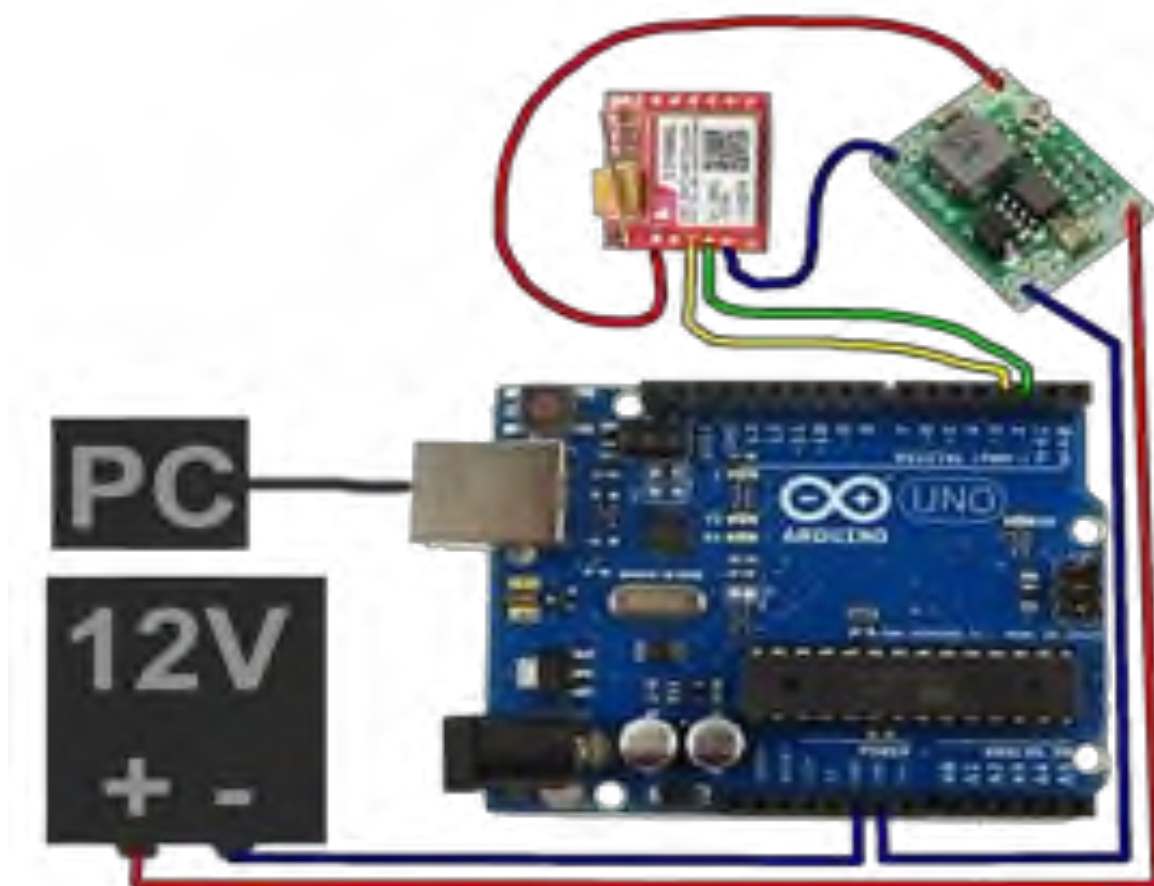
Ishdan maqsad: GSM moduli haqida ko'nikmga ega bo'lish Arduino platasiga bilan bo'lanishini o'rganish.

Kerakli qurilmalar:

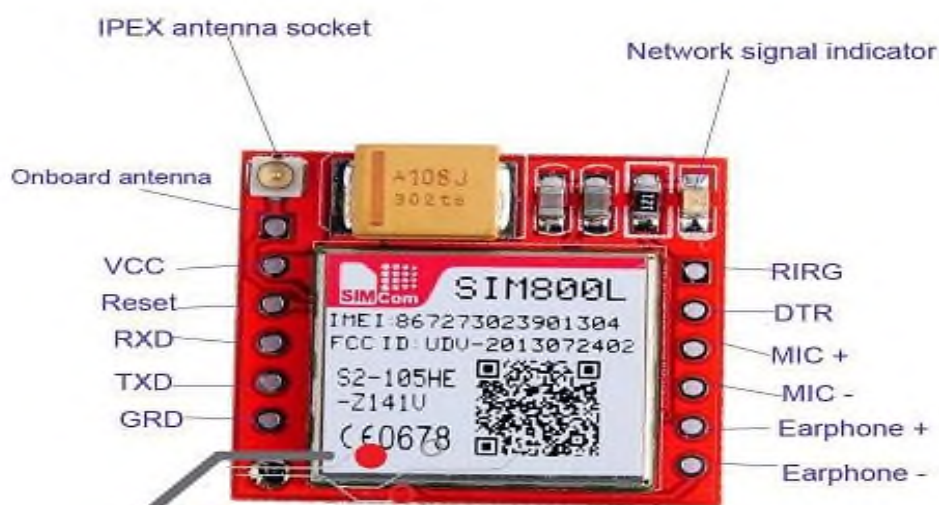
1. Arduino UNO R3 kontrolleri va USB kabellari
2. SIM800L GSM moduli (elektr manbasi $3.7\text{ V} < U < 4.2\text{ V}$)
3. Kuchlanishni kamaytiruvchi o'zgartirgich (DC-DC step-downconverter)
4. 12 V li batareya yoki ixtiyoriy elektr manbasi 6 V dan 20 V gacha
5. Ulovchi similar

Nazariy ma'lumot

GSM SIM800L modulining Arduino ga ulanish sxemasi:



GSM SIM800L modulining tuzilishi



Kuchlanishni o'zgartirgich orqali GSM modulga 12 V li batareyadan elektr kuchlanishi beramiz:

1. 12 V li batareyaning “manfiy” qutbi Arduino platasining GND siga ulanadi, GND esa kuchlanish o'zgartirgichning “kirish manfiy” qutbiga ulanadi.

2. 12 V li batareyaning “musbat” qutbi esa kuchlanish o'zgartirgichning “kirish musbat” qutbiga ulanadi.

Shuni alohida ta'kidlash lozimki, modul ishlashi uchun Arduino dan 5 V ni ulash xavfli bo'lishi mumkin. Shuning uchun uni sezgirlikni teksirish uchun avvaliga 3.3 V manba beramiz. Bu esa modullar o'rtasida bog'lanish hosil qilish uchun yetarli.

Bog'lanish o'rnatilgandan so'ng kuchlanish o'zgartirgich orqali $3.7 \text{ V} < U < 4.2 \text{ V}$ diapazondagi ixtiyoriy kuchlanishni chiqishga berishimiz mumkin.

Kuchlanish o'zgartirgichning chiqish qutblari GSM modul bilan o'zaro aloqani saqlaydi.

GSM SIM800L modulining TX va RX portlari Arduino platasining mos ravishda 2 va 3 oyoqchalariga ulanadi.

Agar biz Arduino platasiga bir vaqtning o'zida bir nechta GSM modulini ulamoqchi bo'lsa k, bimalol Arduino platasining boshqa oyoqchalarini ishlatishimiz mumkin va buni SoftwareSerial dasturi orqali belgilab qo'yamiz.

Bir nechta AT buyruqlari

1. AT + CMGF = 1 – SMS xabarni matn rejimini o'rnatish.
2. AT + IFC = 1, 1 – ma'lumot uzatishning dasturiy nazoratini o'rnatadi.

3. AT + CPBS = "SM" - xuddi shu telefon kitobidagi SIM kartaga kirishni ochadi.
4. AT + CNMI = 1,2,2,1,0 – yangi xabar kelganligi haqidagi signalni yoqish, kelgan yangi xabarning keying formati + CMT: “ < telefon raqami > ”, “”, “<kun, soat>” , keying qatordan xabarning birinchi belgisi boradi.

Sketch – dastur

```
#include <SoftwareSerial.h>
SoftwareSerial mySerial(2, 3); // RX, TX
void setup() {
  Serial.begin(19200); // Skorost porta dlya svyazi Arduino s kompyuterm
  Serial.println("Goodnightmoon!");
  mySerial.begin(19200);
  // Skorost porta dlya svyazi Arduino s GSM modulem
  mySerial.println("AT");
}
void loop() {
  if (mySerial.available())
    Serial.write(mySerial.read());
  if (Serial.available())
    mySerial.write(Serial.read());
}
```

SMS jo‘natish protsedurasi

```
void sms(String text, String phone) {
  Serial.println("SMS send started");
  mySerial.println("AT+CMGS=\"" + phone + "\"");
  delay(1000);
  mySerial.print(text);
  delay(300);
  mySerial.print((char)26);
  delay(300);
  Serial.println("SMS send finish");
  delay(3000);
}
```

Protsedura sketch dastur oxiriga qo‘shiladi va asosiy siklga quyidagicha murojaat qiladi:

sms(String("sms xabar matni"), String("+telefon raqami"));

SMS xabar jo‘natishni faqat muhim hodisalar uchun yoki sikl oddiy bo‘lmagan xollarda tavsiya qilamiz.

Tekshirish:

Port monitori orqali elektr manbasini ulaymiz, tezligini 19200 ga belgilaymiz va albatta “yangi qator” holatini belgilaymiz.

ATI buyrig‘ini kiritamiz va Enter tugmasini bosamiz. Keyin moduldan axborotni olishimiz mumkin.

Topshiriqlar

№	LED uchun portlar	Svetodiodlar soni	Kechikish vaqti, ms
1	2	1	200
2	3,8	2	100
3	2-8	6	500
4	2,4,6,7	4	2000
5	6,7,8	3	300
6	2,9	2	450
7	2	1	1000
8	3-7	4	850
9	8,9	2	600
10	3-9	6	520
11	4-8	4	430
12	2	1	1000
13	2-6	4	670
14	7,8,9	3	780
15	2	1	1000
16	5,7	2	1500
17	4,5,6	3	440
18	1,5,6,8	4	250
19	8,9	2	500
20	5	1	200

Nazorat savollari

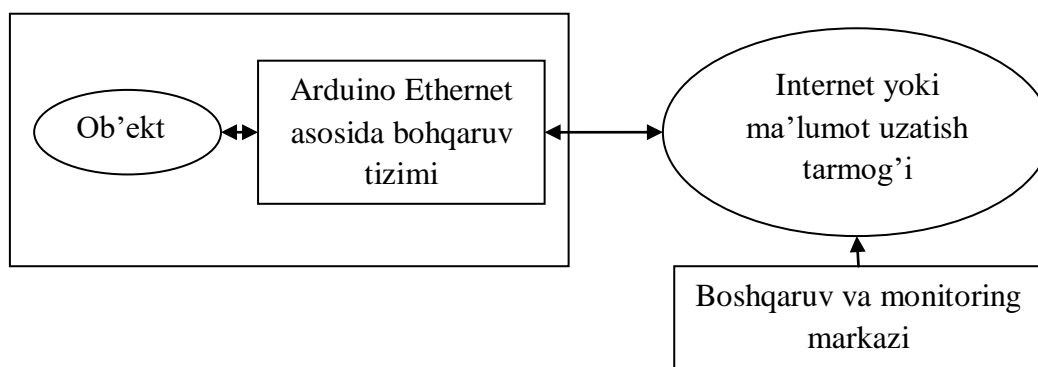
1. GSM SIM800L modulini Arduino ga ulash sxemasini chizing va tushuntiring.
2. GSM modul tuzulishini tushuntiring.
3. Qo‘llanilish sohalari haqida ma’lumot bering.

Laboratoriya ishi № 12

Mavzu: O‘rnatiladigan ma’lumot uzatish va boshqarish tizimini loyihalash

Ishdan maqsad – Embedded Arduino asosida ma’lumot uzatish tamoyillarini o‘rganish va boshqaruv tizimlarini loyihalash.

Narsalar interneti (IoT) konsepsiyasiga asosan boshqariluvchi ob’ekt internetga ulanish imkoniyatiga ega bo‘lishi zarur. Shu sababli Arduino Ethernet dasturlanuvchi apparat ta’minotidan foydalanamiz. Bu holatda ob’ektni boshqarish tizimini quyidagi ko‘rinishda tasvirlaymiz.



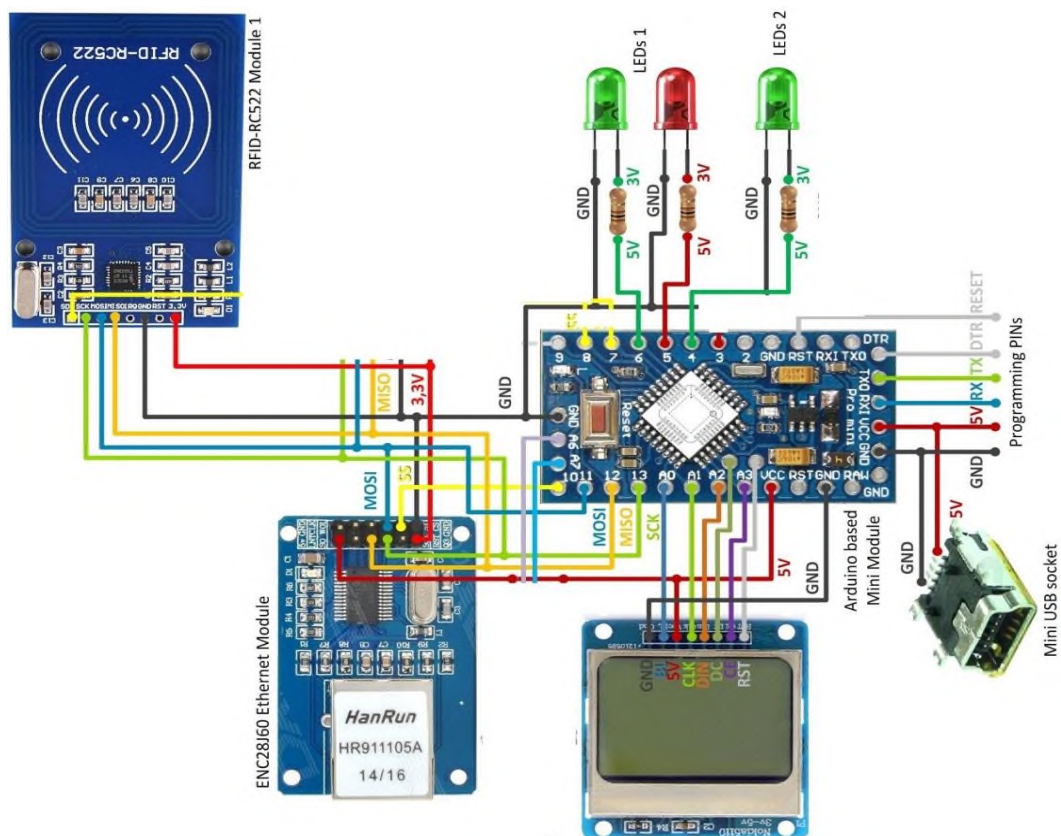
12.1-rasm. Arduino Ethernet boshqaruv tizimi

Ushbu ishlab chiqilgan boshqaruv tizimi modelini real ob’ektlar misolida tadqiq qilamiz.

Arduino apparat platformasi asosida boshqaruv tizimini yaratish va ishlash samaradorligi

Arduino asosida asosan kichik o‘lchamli yoki ob’ektlarni nazorat qilish va boshqarish amalga oshiriladi. Arduino tizimi bilan ishlash uchun turli vazifalarni bajaruvchi datchiklar, sensorlar, boshqaruv motorlari, harakatlanuvchi qurilmalar mavjud. Ushbu qurilmalar yordamida turli ob’ektlarni boshqarirish va nazorat qilish mumkin.

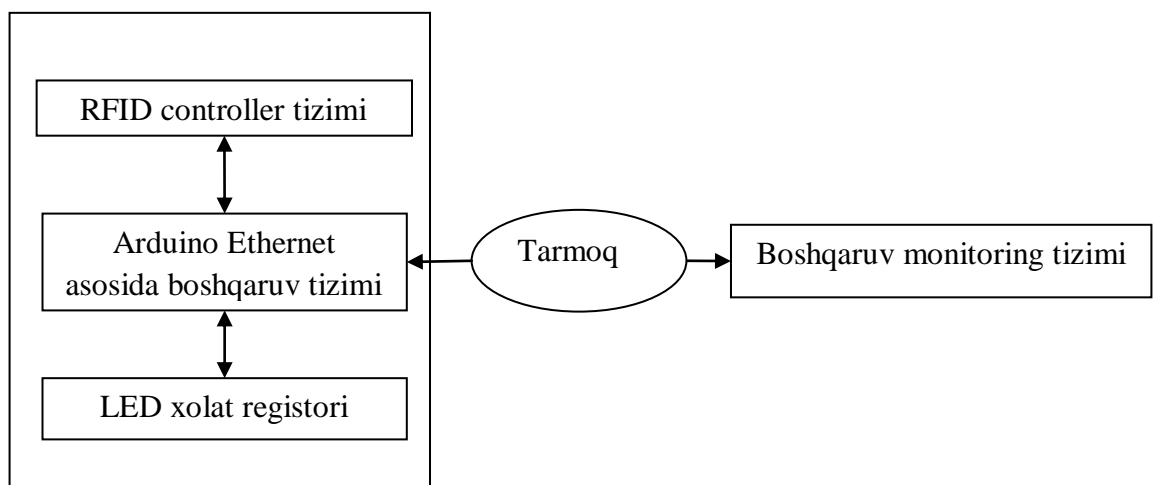
Dissertatsiya ishining maqsadi Arduino asosida boshqaruv tizimini ishlab chiqish va tadqiq qilish bo‘lganligi sababli, biz quyida maketi keltirilgan tizimni ishlab chiqdik va uning ustida tadqiqotlar olib bordik (12.2- rasm).



12.2- rasm. Arduino asosidagi yaratilgan tizim maketi

Ushbu tizim RFID kirishni boshqarish tizimi bo‘lib, bugungi kunda turli muassasa va ob‘ektlarga kirishni boshqarish uchun qo‘llaniladi.

Tadqiqot uchun yaratilgan tizim quyidagi rasmda keltirilgan ko‘rinishda tuzilgan, ya’ni, RFID qurilmasi, Arduino Ethernet apparat ta’minoti, holatlar indikatorlari va boshqaruv markazi(12.3-rasm).



12.3- rasm. Tadqiqot uchun yaratilgan tizim tuzilishi

Ushbu algoritm asosida RFID tizimi uchun dasturiy ta'minot ishlab chiqilgan. Dastur kodi keltiramiz:

Arduino dasturi kodi

RFID ni o'qish

```
#include <SPI.h>
```

```
#include <rfid.h>
```

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

```
#define SS_PIN 10
```

```
#define RST_PIN 9
```

```
RFID rfid(SS_PIN, RST_PIN);
```

```
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);
```

```
// Setup variables:
```

```
int serNum0;
```

```
int serNum1;
```

```
int serNum2;
```

```
int serNum3;
```

```
int serNum4;
```

```
void setup()
```

```
{
```

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

```
lcd.begin(16, 2);
```

```
SPI.begin();
```

```
rfid.init();
```

```
}
```

```
void loop()
```

```
{
```

```
if (rfid.isCard()) {
```

```
if (rfid.readCardSerial()) {
```

```
if (rfid.serNum[0] != serNum0
```

```
&& rfid.serNum[1] != serNum1
```

```
&& rfid.serNum[2] != serNum2
```

```
&& rfid.serNum[3] != serNum3
```

```
&& rfid.serNum[4] != serNum4
```

```
) {
```

```
/* With a new cardnumber, show it. */
```

```
Serial.println(" ");
```

```
Serial.println("Card found");
```

```
serNum0 = rfid.serNum[0];
```

```
serNum1 = rfid.serNum[1];
```



```

        serNum2 = rfid.serNum[2];
        serNum3 = rfid.serNum[3];
        serNum4 = rfid.serNum[4];
        //Serial.println(" ");
        Serial.println("Cardnumber:");
        Serial.print("Dec: ");
        Serial.print(rfid.serNum[0],DEC);
        Serial.print(", ");
        Serial.print(rfid.serNum[1],DEC);
        Serial.print(", ");
        Serial.print(rfid.serNum[2],DEC);
        Serial.print(", ");
        Serial.print(rfid.serNum[3],DEC);
        Serial.print(", ");
        Serial.print(rfid.serNum[4],DEC);
        Serial.println(" ");
    }
else {
    /* If we have the same ID, just write a dot. */
    Serial.print(".");
}
}
}
}
rfid.halt();
}
Ob'ektni boshqarish
#include <SoftwareSerial.h>
#include <Stepper_28BYJ.h>
int gRxPin = 10;
int gTxPin = 11;
SoftwareSerial BTSerial
(gRxPin, gTxPin);
#define steps 4078
Stepper_28BYJ stepper(steps, 2,3,4,5);
char input;
void setup()
{
    Serial.begin(9600);
    stepper.setSpeed(13);

```

```

BTSerial.begin(38400);
//Serial.println(">>start<<");
delay(300);
}
void loop() {
  if (BTSerial.available())
  {
    input = BTSerial.read();
    Serial.write(BTSerial.read());
    if(input == '1')
    {
      Serial.println("on");
      stepper.step(1000);
    }
    if(input == '0')
    {
      Serial.println("off");
      stepper.step(-1000);    }
    }
  }
}

```

Endi yaratilgan ushbu tizimni tadqiq qilishga o‘tamiz.

Avvalo, ushbu tizimni tadqiq qilishdan maqsad, Arduino asosidagi tarmoqli boshqaruv jarayonini baholash hisoblanadi. Shu sababli, yaratilgan tizim ustida bir qator eksprementlar o‘tkazamiz va quyidagilarni aniqlaymiz:

- lokal, simli va simsiz tarmoq orqali RFID identifikatsiya uchun ketadigan vaqt;
- tarmoqli boshqaruv tizimlarida kechikish
- loqa kanalining boshqarish jarayoniga ta’siri: Ethernet vs WiFi asosida;
- TCP asosida boshqaruv vaqti;
- UDP asosida boshqaruv vaqti;
- Tizimning ishlash samaradorligini analitik baholash.

Tajrivba №1.

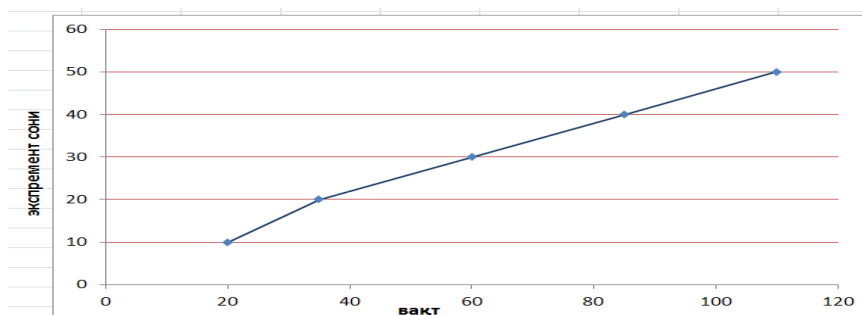
Birinchi eksprement asosida RFID identifikatsiya vaqtini hisoblaymiz. Bir necha marta RFID qurilmasiga qilinganda tarmoq orqali kirishni boshqarish uchun ketadigan vaqtni hisoblaymiz.

O'tkazilgan eksperimentga ko'ra, quyidagi natija olingan (13.1-rasm).

Tajribalar soni	Vaqt (sekund)
10	20
20	35
30	60
40	85
50	110

13.2-rasm. RFID identifikatsiya soni va unga ketadigan vaqt.

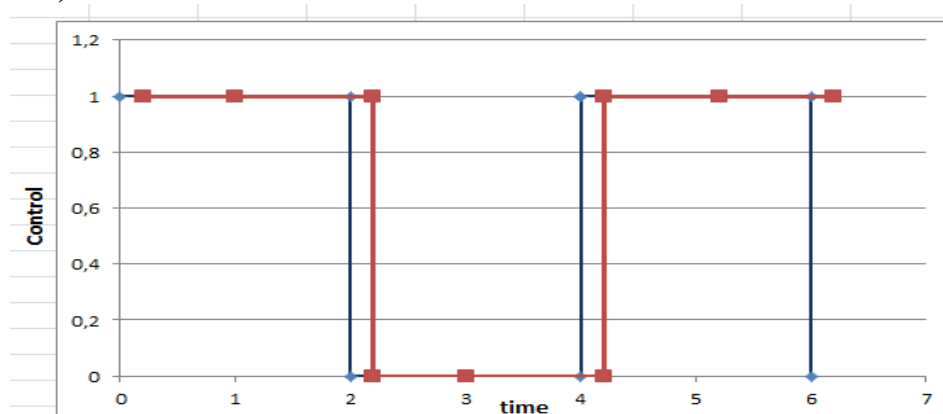
RFID identifikatsiya sonining identifikatsiya uchun ketadigan vaqtga bog'lanishi quyidagi ko'rinishda bo'ladi (13.3-rasm).



13.3- rasm. RFID identifikatsiya sonining identifikatsiya uchun ketadigan vaqtga bog'lanishi.

Demak, identifikatsiya vaqtining identifikatsiya soniga nohiziqli bog'langan. Umumiy holatda oladigan bo'lsa k, o'lchangan vaqt sensor orqali o'qib olingan ID raqamini serverda tekshirib, mavjud bo'lsa, ob'ektga kirishga ruxsat bergungacha bo'lgan jarayonni o'z ichiga oladi.

ID raqamini tekshirish va ob'ektni harakatlantirishni lokal va tarmoq orqali boshqarishda olingan natijalar quyidagi rasmda keltirilgan (13.4- rasm).

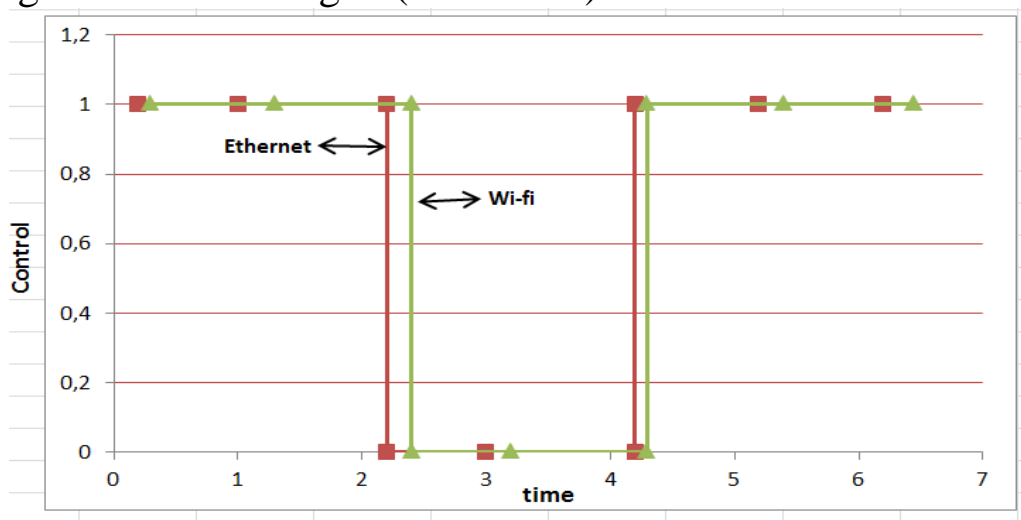


13.4- rasm. Lokal va tarmoqli boshqaruv vaqti.

Ko'p adabiyotlarda boshqaruv vaqti sifatida lokal boshqaruv vaqti olinadi, ya'ni, ob'ektning bevosita o'zidan yoki uning yoniga o'rnatilgan kompyuterli boshqaruv tizimidan foydalanish tushuniladi. Shu sababli, tarmoqli boshqarish kechikishli boshqaruv hisoblanadi. Bunda tizimning tarmoq orqali boshqarishda yuzaga keladigan kechikishlar hisoblanadi.

Tajriba №2.

Ob'ektni tarmoq orqali boshqarish uchun simli va simsiz texnologiyalar qo'llanishi mumkin. Biz tuzilish jihatidan bir-biriga yaqin bo'lgan ikki texnologiya Ethernet va WiFi tarmoqlarida yuqoridagi eksprementni o'tkazdik. Simli va simsiz aloqa kanali boshqarish jarayoniga turlicha ta'sir ko'rsatadi. Olingan natijalar quyidagi rasmda keltirilgan (13.5-rasm).



13.5- rasm. Ethernet va WiFi tarmoqlarida ob'ektni boshqarishda yuzaga keladigan kechikish

Eksprement uchun 10 Mbit/s tezlikli Arduino Ethernet va WiFi modullaridan foydalanildi. Olingan tadqiqotlardan ko'rinadiki, simsiz tarmoq orqali boshqarishda kechikish qiymati yuqori bo'ladi. Chunki, simsiz tarmoq muhitiga tashqi ta'sirlar yuqori bo'ladi. Eksprement asosida TCP va UDP protokollarining ob'ektni boshqarishga ta'siri aniqlangan.

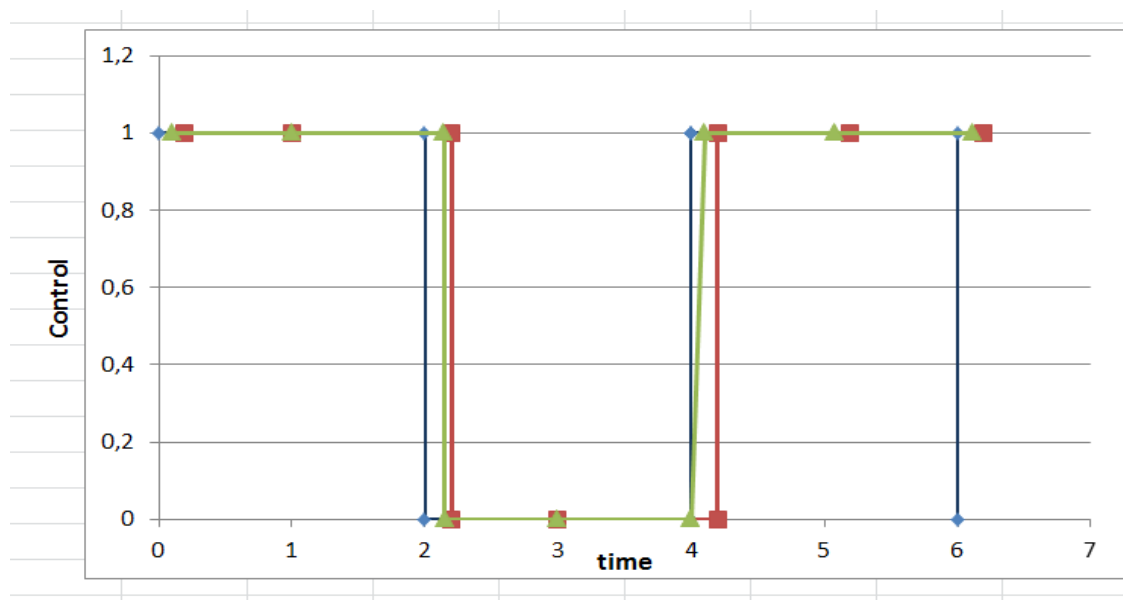
Olingan tadqiqot natijalari shuni ko'rsatadiki, tarmoqli boshqaruv jarayonida UDP protokoli asosida ishlovchi tizimdan foydalanish kechikishning kam bo'lishiga olib keladi.

O'tkazilgan eksprementlar asosida asosan boshqarishda

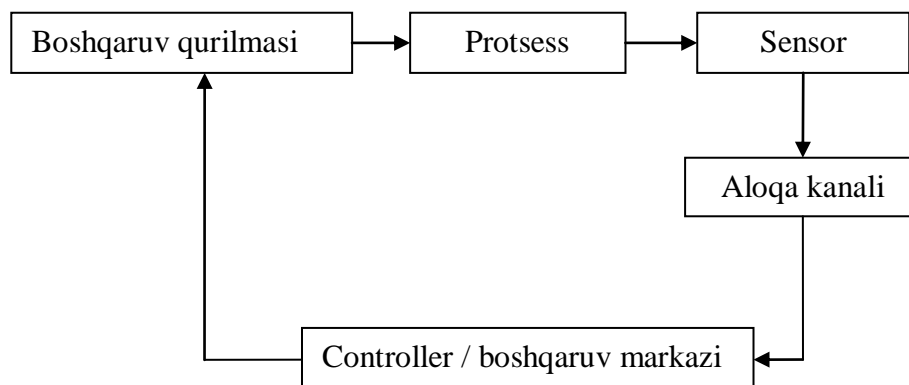
bo‘ladigan kechikishlar va ularga tarmoq texnologiyalarining ta’siri ko‘rib chiqilgan.

Tarmoqli boshqaruv jarayonini baholash uchun tarmoqli boshqaruv tizimlari nazariyasi elementlaridan foydalaniladi. Ushbu nazariya asosida tarmoqli boshqaruv jarayonini hisoblash va baholash mumkin.

Biz UDP va TCP asosida boshqarishni baholash uchun tarmoqli boshqaruv tizimlari nazariyasi elementlaridan foydalandik. Ushbu nazariya asosida boshqaruv sifati, samaradorligi va boshqaruv jarayonida yuzaga keladigan kechikishlarni aniqlash mumkin. Tarmoqli boshqaruv tizimlari nazariyasi quyidagi modelga asosan tavsiflanadi.



13.6- rasm. Lokal, UDP va TCP asosida boshqaruv tizimlari eksprement natijalari.



13.7- rasm. Tarmoqli boshqaruv tizimlari nazariyasi elementlari

Ushbu nazariya asosida tarmoqli boshqaruv tizimlarida yuzaga keladigan kechikishni baholaymiz.

Tarmoqli boshqaruv tizimlarida umumiy kechikish – D_k :

$$D_k = D_k + D_b$$

Kontrollerga ob'ektdagi sensor ma'lumotini uzatish vaqti – D_k :

$$D_k = T_s + T_c + T_p + T_{qc} + T_{con}$$

Kontroller orqali ob'ektni boshqarish vaqti – D_b :

$$D_b = T_{bq} + T_c + T_p + T_t.$$

Hisoblashlar uchun boshqaruvda ishtirok etuvchi elementlarning standart ishlash vaqt qiymatlari olingan:

- $T_s = 0,2 - 0,3s$;
- $T_c = 0,3s$;
- $T_p = 0,05-0,1s$ oralig'ida, 10 Mbit/s kanal uchun olingan;
- $T_{qc} = 0,1-0,5s$;
- $T_{con} = 0,016s$;
- $T_{bq} = 0,01-0,02s$;
- $T_t = 0,1-1,3s$.

Avval TSR protokoli asosida boshqaruv tizimida bo'ladigan kechikishlarni hisoblaymiz. Ma'lumki, TSR protokoli asosida ishlovchi tarmoqlarda ma'lumot almashish dastlab ulanish orqali amalga oshiriladi shu sababli, tarmoqli boshqaruv tizimi modelini quyidagi ko'rinishda tasvirlaymiz. TCP asosida tarmoqli boshqaruv quyidagi formulalar asosida aniqlanadi.

TCP orqali sensordan ob'ekt holati ma'lumotini qabul qilishdagi kechikish:

$$D_k = T_s + T_c + T_p + T_{qc} + T_{con}.$$

Bu yerda:

- D_k –kontrollerga sensor ma'lumotini uzatish vaqti;
- T_s –sensor ishlash vaqti;
- T_c – ulanish vaqti;
- T_p – tarmoqda uzatish vaqti;
- T_{qc} –kontroller yoki boshqaruv tizimida qabul qilish va kutish vaqti;
- T_{con} - kontroller yoki boshqaruv tizimida ma'lumotni qayta ishlash vaqti.

TCP asosida kontroller orqali boshqarish vaqti:

$$D_b = T_{bq} + T_c + T_p + T_t.$$

Bu yerda:

- D_b –kontroller orqali boshqarish vaqti;

- T_c – ulanish vaqti (ulangan bo‘lsa, hisobga olinmaydi);
- T_p – tarmoqda uzatish vaqti;
- T_{bq} – boshqaruv qurilmasi ishlash vaqti;
- T_t – ob‘ektga ta’sir vaqti.

Hisoblashlar bo‘yicha TSR da boshqaruv vaqti - $D_{uk_TCP} = 2,235$ sekundga teng. Endi UDP asosida kechikish vaqtini hisoblaymiz.

Sensordan ob‘ekt holati ma’lumotini UDP asosida qabul qilishdagi kechikish:

$$D_k = T_s + T_p + T_{qc} + T_{con},$$

formula asosida hisoblanadi.

Bu yerda:

- D_k – kontrollerga sensor ma’lumotini uzatish vaqti;
- T_s – sensor ishlash vaqti;
- T_p – tarmoqda uzatish vaqti;
- T_{qc} – kontroller yoki boshqaruv tizimida qabul qilish va kutish vaqti;
- T_{con} - kontroller yoki boshqaruv tizimida ma’lumotni qayta ishlash vaqti.
- UDP asosida kontroller orqali boshqarish vaqti quyidagiga teng:

$$D_b = T_{bq} + T_p + T_t$$

Bu yerda:

- D_b – kontroller orqali boshqarish vaqti;
- T_p – tarmoqda uzatish vaqti;
- T_{bq} – boshqaruv qurilmasi ishlash vaqti;
- T_t – ob‘ektga ta’sir vaqti.

Hisoblash bo‘yicha UDR da boshqaruv vaqti - $D_{uk_UDP} = 2,062$ sekundga teng. Hisoblashlardan ko‘rinadiki, UDR asosida tarmoqli boshqaruv tizimlarini loyihalash samaralidir. Chunki UDR protokolida tarmoqda ma’lumot uzatish ulanishsiz amalga oshiriladi va bunda ulanishga ketadigan vaqt hisobga olinmaydi. Shuningdek, tarmoqli boshqaruv nazariyasi asosida tarmoqda ob‘ektlarni boshqarishda bo‘ladigan kechikishlar aniqlandi. UDP va TCP asosida ishlovchi tizimlarda ob‘ektlarni boshqarishda yuzaga kelishi mumkin bo‘lgan kechikishlar hisoblandi va boshqaruv tizimlarini loyihalashda UDP dan foydalanish samarali ekanligi isbotlandi.

Talabalar uchun variantlar

Talabalar o'zlarining jurnalidagi tartib raqamlari bo'yicha belgilangan Arduino platasi chiqishlariga svetodiodni ulab, TCP va UDP asosida ularni boshqaradilar va holatni displeyda tasvirleydilar.

T/r	Pin raqami	T/r	Pin raqami
1.	1-2-4	16.	1-3-5
2.	2-5-6	17.	7-9-0
3.	3-6-9	18.	13-2-4
4.	2-4-11	19.	2-4-6
5.	0-10-4	20.	8-10-12
6.	8-9-10	21.	1-2-3
7.	10-1-8	22.	4-5-6
8.	2-9-4	23.	7-8-9
9.	3-8-5	24.	10-11-12
10.	4-7-9	25.	13-3-2
11.	5-6-13	26.	4-3-1
12.	2-12-2	27.	6-5-4
13.	3-13-1	28.	11-0-4
14.	4-12-5	29.	12-11-13
15.	5-0-2	30.	1-13-0

Laboratoriya mashg'ulotlari №14, №15.
O'rnatilgan tizimlarni loyihalash vositalari bilan tanishish va ularda ishlashni o'rganish. Sodda o'rnatilgan tizimlarni loyihalash

Ishdan maqsad: Sozlash platasi asosida mikrokontrollerni bitta elektrodini ma'lum vaqt oralig'ida yoqib o'chirish.

Nazariy ma'lumotlar

MK(mikrokontroller) – mikroprotessor tizimining soddarog'i bo'lib, unda hamma yoki tizimni ko'pchilik tugunlari bitta mikrosxema shaklida qilingan.

AVR – Atmel firmasining sakkiz bitli mikrokontrollerlari 1996 yilda yaratilgan.

AVR ni qisqartmasini bir qancha kengaytmasi mavjud. Kimdir Advanced Virtual RISC (Kengaytirilgan virtual RISC) deb takidlaydi, boshqalar AVRni yaratgan ikki insonni ismini bosh harflar olingan deyishadi Alf Egil Bogen Vegard Wollan RISC.

AVR mikrokontrollerlari Garvard arxitekturasiga ega (dastur va ma'lumotlar hotirasi alohida bo'ladi) buyriqlar tizimi RISC g'oyasiga yaqin. AVR protessori 32 ta 8 – bitdan iborat umumiy foydalanishga mo'ljallangan registirlarga ega, registirli fayllarga birlashtirilgan.

Ko'pchilik buyruqlar 1 ta yacheyka xotirani egalaydi(16 bit).

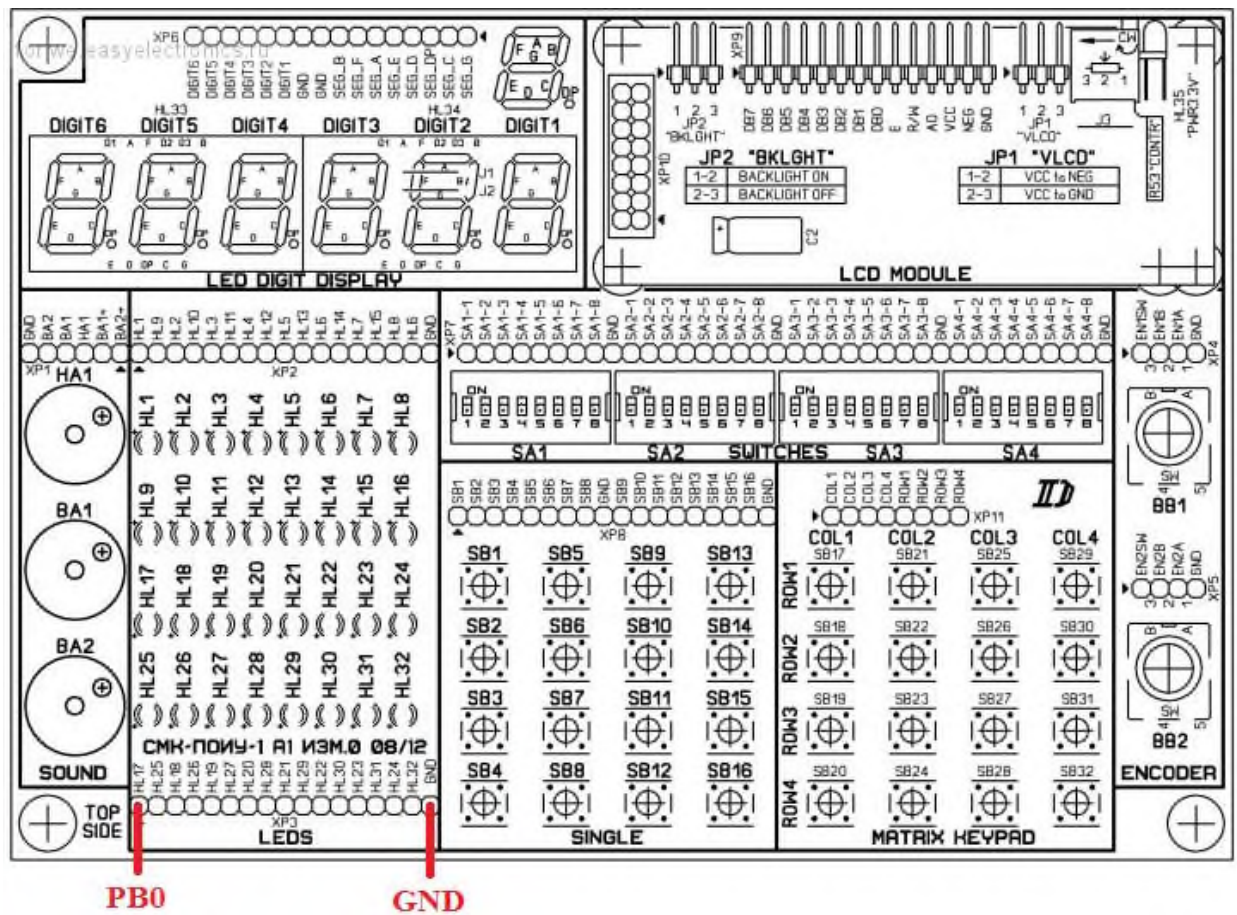
Ko'pchilik buyriqlar 1 davrda bajariladi.

Ularni quyidagi guruhlariga ajratish mumkin:

AVR mikrokontrollerlarini bir nechta guruhlariga ajratish mumkin:

- Mantiqiy operatsiyalar;
- Mantiqiy va surish operatsiyasi;
- Bitlar bilan ishlash;
- Uzatish operatsiyalari;
- Boshqarishni uzatish operatsiyasi;
- Tizimni boshqarish operatsiyasi.

Periferiya qurilmalarini boshqarish ma'lumot adres maydoni asosida boshqariladi. Qulaylik uchun qisqartirilgan IN/OUT buyruqlardan foydalaniladi.



(XCK/T0) PB0	1	40	PA0 (ADC0)
(T1) PB1	2	39	PA1 (ADC1)
(INT2/AIN0) PB2	3	38	PA2 (ADC2)
(OC0/AIN1) PB3	4	37	PA3 (ADC3)
(SS) PB4	5	36	PA4 (ADC4)
(MOSI) PB5	6	35	PA5 (ADC5)
(MISO) PB6	7	34	PA6 (ADC6)
(SCK) PB7	8	33	PA7 (ADC7)
RESET	9	32	AREF
VCC	10	31	GND
GND	11	30	AVCC
XTAL2	12	29	PC7 (TOSC2)
XTAL1	13	28	PC6 (TOSC1)
(RXD) PD0	14	27	PC5 (TDI)
(TXD) PD1	15	26	PC4 (TDO)
(INT0) PD2	16	25	PC3 (TMS)
(INT1) PD3	17	24	PC2 (TCK)
(OC1B) PD4	18	23	PC1 (SDA)
(OC1A) PD5	19	22	PC0 (SCL)
(ICP1) PD6	20	21	PD7 (OC2)

1.1- rasm. O'quv platasining umumiy ko'rinishi.

```
#include<avr/io.h>
```

```
//Sarlavhali faylni ulash (Mkni registrlarini fizikmanzillarini ro'yhati)
```

```
#include<util/delay.h> // Sarlavhali faylni ulash
```

```
(_delay_ms() va _delay_us())funksiyalarini yozilmasi)
```

```

intmain(void)
//Asosiy funktsiya (dasturboshi)
{
DDRB= 0xFF; // PORTBni chiqish deb e'lon qilamiz
PORTB=0x00; //B portda 0[V] deb o'rnatamiz
while(1) // Uzliksiz davr

{
PORTB = 0x01; //PORTB0ga 5[V]berilsin
_delay_ms(50); //PORTB0 50 ms yoqib turilsin
PORTB = 0x00; //PORTB0ga 0[V]berilsin
_delay_ms(50); //PORTB0 50 ms o'chiq turilsin
}
}

```

Nazorat savollari

1. Mikrokontroller deb nimaga aytiladi?
2. AVR qisqartma so'zining ma'nosini ayta olasizmi?
3. AVR mikrokontrollerlari qaysi firma tomonidan ishlab chiqariladi?
4. AVR mikrokontrollerlarini qanday guruhlarga ajratish mumkin?
5. AVR mikrokontrollerlari qanday arxitekturaga asoslangan?

Variantlar

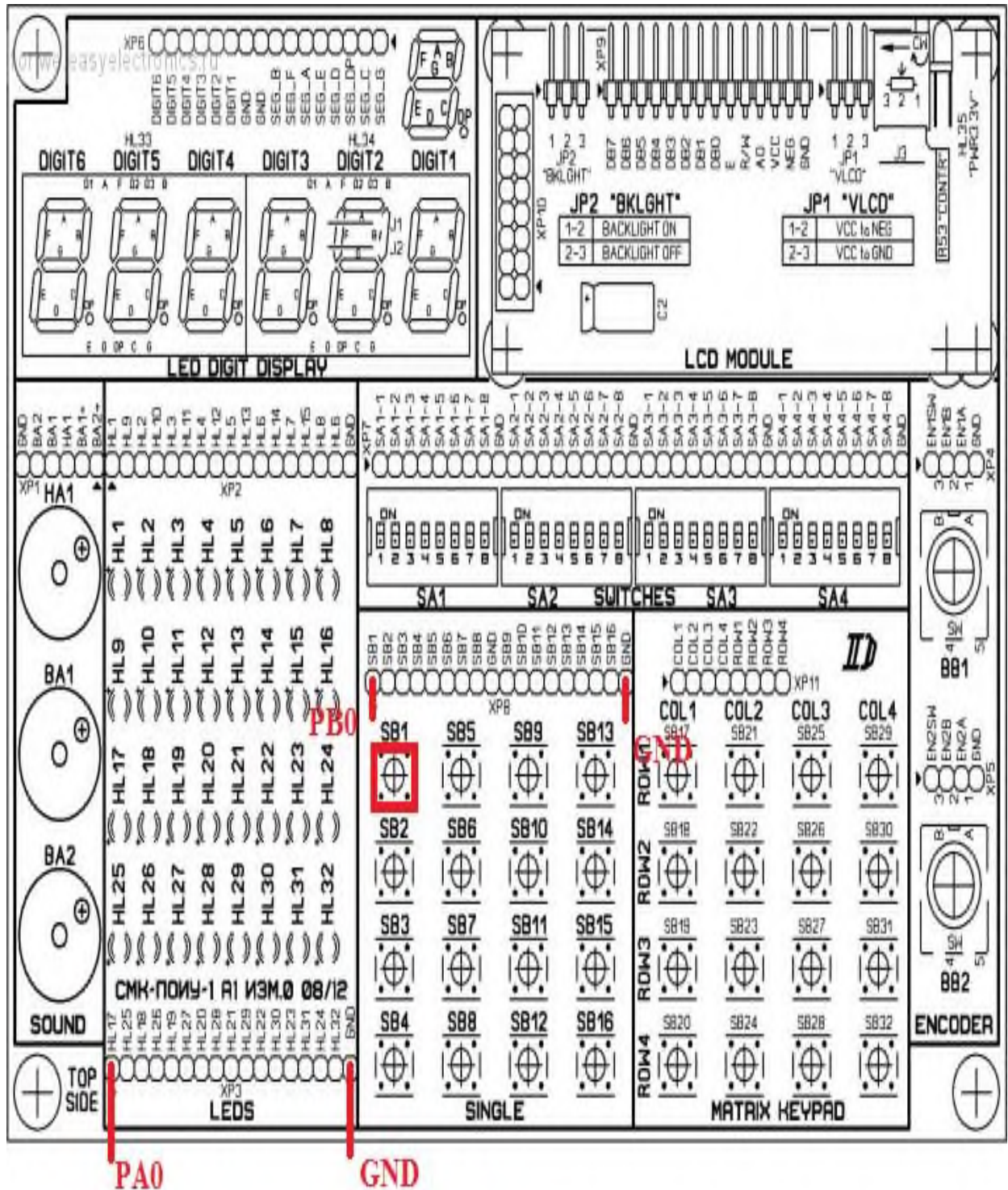
№	Kuchlanish	Yorug'lik diodlari soni va rangi	Qarshiliklar soni yorug'lik diodlariga mos ravishda	Kalitlar soni
1	9 volt	Green led 4x	Resistor 4x	SPST 2x
2	9 volt	Red led 2x, Green	Resistor 4x	SPST 1x
3	10.5 volt	Yellow led 3x	Resistor 3x	SPST 1x
4	9 volt	Red led 5x	Resistor 5x	SPST 1x
5	11 volt	Green led 3x	Resistor 3x	SPST 2x
6	9 volt	Red led 1x, Green	Resistor 2x	SPST 1x
7	10.5 volt	Red led 2x, Green	Resistor 4x	SPST 1x

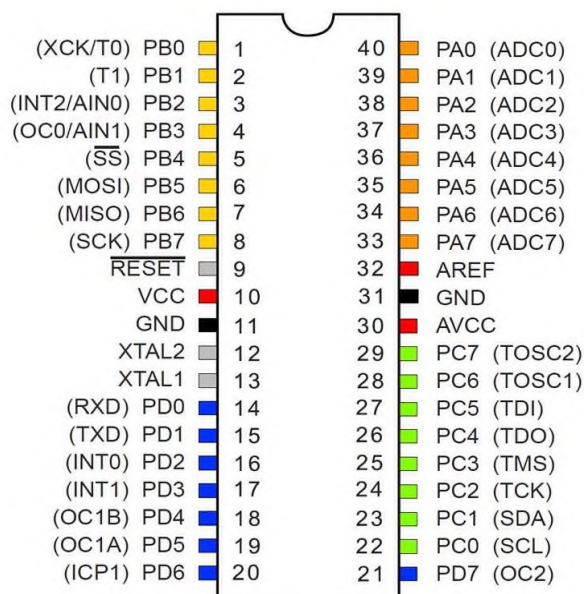
8	9 volt	Yellow led 5x	Resistor 5x	SPST 2x
9	11 volt	Green led 3x	Resistor 3x	SPST 1x
10	9 volt	Red led 5x	Resistor 5x	SPST 1x
11	8 volt	Yellow led 5x	Resistor 5x	SPST 2x
12	9 volt	Yellow led 3x	Resistor 3x	SPST 1x
13	8 volt	Green led 4x	Resistor 4x	SPST 1x
14	12 volt	Red led 1x, Green	Resistor 2x	SPST 1x
15	11 volt	Yellow led 5x	Resistor 5x	SPST 2x
16	12 volt	Green led 3x	Resistor 3x	SPST 1x
17	9.6 volt	Green led 6x	Resistor 6x	SPST 1x
18	12 volt	Red led 2x, Green	Resistor 4x	SPST 1x
19	10.5 volt	Green led 6x	Resistor 6x	SPST 2x
20	8 volt	Yellow led 3x	Resistor 3x	SPST 1x
21	11 volt	Red led 5x	Resistor 5x	SPST 1x
22	12 volt	Green led 3x	Resistor 3x	SPST 1x
23	11.1 volt	Yellow led 5x	Resistor 5x	SPST 2x
24	10.5 volt	Red led 5x	Resistor 5x	SPST 1x
25	11.1 volt	Green led 3x	Resistor 3x	SPST 1x
26	12 volt	Red led 1x, Green	Resistor 2x	SPST 1x
27	9.6 volt	Red led 5x	Resistor 5x	SPST 2x
28	8 volt	Green led 4x	Resistor 4x	SPST 1x
29	11 volt	Yellow led 3x	Resistor 3x	SPST 1x
30	9.6 volt	Red led 2x, Green led 2x	Resistor 4x	SPST 2x

Laboratoriya mashg'ulotlari №16, №17

Mavzu: Sodda o'rnatilgan tizimlarni loyihalash

Ishdan maqsad: Sozlash platasi asosida mikrokontrollerni tugmacha va svetodioldniga mantiqiy nol berish orqali mikrokontrollerni boshqa elektrodini yoqish.

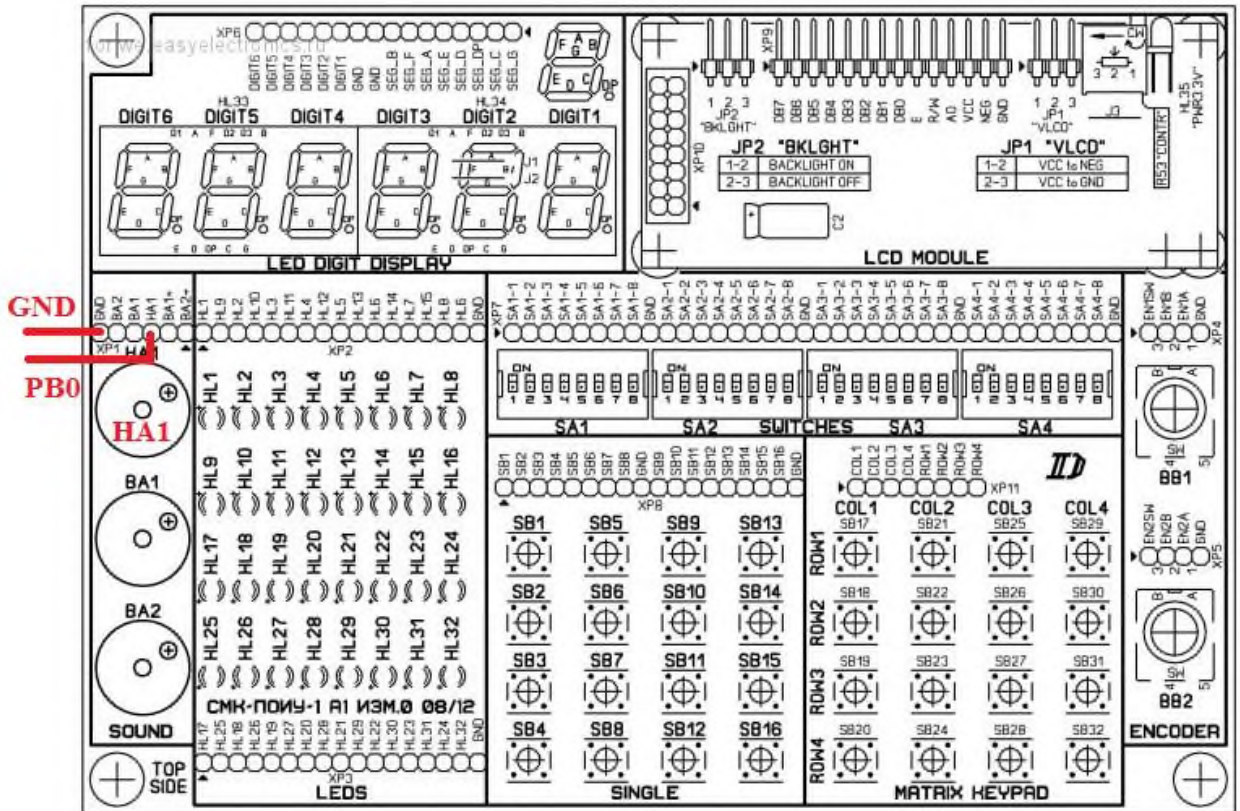




2.1- rasm. Atmega 16 mikrokontrolleri

```
#include<avr/io.h> //Sarlavhali faylni ulash (MKni registirlarini fizik
manzillarini ro‘yxati)
#include<util/delay.h> // Sarlavhali faylni ulash (_delay_ms() va
_delay_us())funksiyalarini yozilmasi)
intmain(void) //Asosiy funktsiya (dastur boshi)
{
DDRA= 0xFF; // PORTAni chiqish deb e’lon qilamiz
PORTA=0x00; //A portda 0[V] deb o’rnatamiz
DDRB= 0x00; // PORTBni kirish deb e’lonqilamiz
PORTB=0xFF; //B portda1[V] deb o’rnatamiz
while(1) // Uzliksiz davr
{
PORTA = ~ PORTB; //~ beligirazyadliinvertirlash
}
}
```

Sozlash platasi asosida mikrokontrollerni bitta elektrodi orqali tovushni hosil qilish.



(XCK/T0) PB0	1	40	PA0 (ADC0)
(T1) PB1	2	39	PA1 (ADC1)
(INT2/AIN0) PB2	3	38	PA2 (ADC2)
(OC0/AIN1) PB3	4	37	PA3 (ADC3)
(\overline{SS}) PB4	5	36	PA4 (ADC4)
(MOSI) PB5	6	35	PA5 (ADC5)
(MISO) PB6	7	34	PA6 (ADC6)
(SCK) PB7	8	33	PA7 (ADC7)
RESET	9	32	AREF
VCC	10	31	GND
GND	11	30	AVCC
XTAL2	12	29	PC7 (TOSC2)
XTAL1	13	28	PC6 (TOSC1)
(RXD) PD0	14	27	PC5 (TDI)
(TXD) PD1	15	26	PC4 (TDO)
(INT0) PD2	16	25	PC3 (TMS)
(INT1) PD3	17	24	PC2 (TCK)
(OC1B) PD4	18	23	PC1 (SDA)
(OC1A) PD5	19	22	PC0 (SCL)
(ICP1) PD6	20	21	PD7 (OC2)

2.2- rasm. Ovoz signallarini ulash jarayonlari.

#include<avr/io.h> //Sarlavhali faylni ulash
(MKnregistrlarinifizikmanzillariniro'yhati)

```

#include<util/delay.h> // Sarlavhali faylni ulash (_delay_ms() va
_delay_us())funksiyalarini yozilmasi)
intmain(void) //Asosiyfunksiya (dasturboshi)
{
DDRB= 0xFF; // PORTBni chiqish deb e'lonqilamiz
PORTB=0x00; //Bportda 0[V] deb o'rnatamiz
while(1) // Uzliksiz davr
{
PORTB = 0x01; //PORTB0ga 5[V]berilsin
_delay_ms(50); //PORTB0 50 ms yoqib turilsin
PORTB = 0x00; //PORTB0ga 0[V]berilsin
_delay_ms(50); //PORTB0 50 ms o'chiq turilsin
}
}

```

Nazorat savollari

1. Intmain(void) buyrug'i nima vazifani bajaradi?
2. Portlarni e'lon qilish uchun qaysi buyruqdan foydalaniladi?
3. Delay buyrug'i qanday vazifani bajaradi?
4. PORTB = 0x00 ushbu buyruqqa izoh bering.

Topshiriq

1. Jurnaldaagi tartib raqamingizga mos bo'lgan elektrodni yoqing va u jurnaldaagi tartib raqamingizga mos ravishda(shuncha sekund vaqt oralig'ida) bir marta yonib-o'chishini ko'rsatib bering.

2. Jurnaldaagi tartib raqamingizga mos bo'lgan elektrodni yoqing va u jurnaldaagi tartib raqamingizga mos ravishda(shuncha marta) yonib-o'chishini ko'rsatib bering.

Laboratoriya mashg'ulotlari № 18, №19
Mavzu: Embedded Arduino board. LCD displey asosida tizimlarni loyihalash

Ishdan maqsad. Arduino UNO yordamida LCD displey asosida tizimlarni loyihalash.

Arduino IDE muhitida LCD-displeya Hitachi HD44780 mavjud. LCD displey tiniqligi boshqaruv kirishiga berilayotgan kuchlanish o'lchamiga bog'liq. Kuchlanish 0.5-1 V oarlig'ida bo'lishi kerak, lekin muhit darajasiga ham bog'liqdir. 5-jadvalda LCD-Arduino pinouti keltirilgan

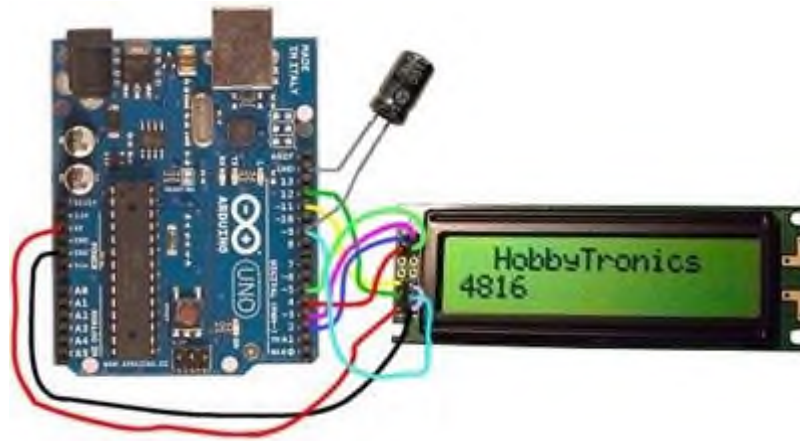
5.1. -jadval.
LCD-Arduino pinout

LCD Pin	Simvol	Naznachenie	Arduino Pin
1	Vss	umumiy (0 V)	umumiy (0 V)
2	Vdd	Pitanie (4.5 – 5.5 V)	+5V
3	Vo	Uprav.kontrastnostyu	9
4	RS	H/L register select signal	12
5	R/W	H/L read/write signal	Umumiy (0 V)
6	E	H/L enablesignal	11
11	DB4	H/L data bus for 4-bit mode	5
12	DB5	H/L data bus for 4--bit mode	4
13	DB6	H/L data bus for 4-bit mode	3
14	DB7	H/L data bus for 4-bit mode	2

Liquid Crystal kutubxonasi barcha LCD Hitachi HD44780 bilan ishlaydi.

Ulanish sxemasi:

- * LCD RS pin raqamli chiqish 12
- * LCD Enable pin raqamli chiqish 11
- * LCD R/W umumiy
- * LCD VO pin (pin 3) ShIMga-chiqish 9



5.1 -rasm. Ulanish jarayoni

- * LCD D4 pin raqamli chiqish 5
- * LCD D5 pin raqamli chiqish 4
- * LCD D6 pin raqamli chiqish 3
- * LCD D7 pin raqamli chiqish 2
- */

```
#include      // kutubxonani ulaymiz
```

```
Liquid Crystalled (12, 11, 5, 4, 3, 2);      // initsializatsiyalash
```

```
Void setup()
```

```
{
pinMode(9, OUTPUT);
analogWrite(9, 50); // ShIM chiqishga o‘rnatish
lcd.begin(16, 2); // qator va ustunlar miqdorini o‘rnatish
lcd.print(" HobbyTronics");//ma'lumotlarni LCD ga uzatish }
voidloop() {
// kursorni v 0-chi ustunga, 1 qator ( 0 dan boshlanadi)
lcd.setCursor(0, 1);
lcd.print(millis()/1000); // qayta yuklashdan so‘ng sekund chop etish
}
```

Nazorat savollari

1. LCD displey tiniqligi nimaga bog‘liq?
2. LCD Hitachi HD44780 qaysi kutubxona bilan ishlaydi?
3. LCD-Arduino pinauti haqida nimalar bilasiz?
4. Arduino muhiti qanday muhit?

Topshiriqlar

1. "Smart uy" loyihasiga asosan maxsus qurilma tayyorlanmoqda. Bu uskumaning vazifasi yoritish chiroqlari ish vaqtini hisoblash orqali elektr energiya iste'molini aniqlashdan iborat. Qurilma algoritmiga asosan takt tugma bosilganda elektr chiroq yonishi va mikrokontroller schetchigi bittaga oshishi kerak. Tugmaning navbatdagi bosilishi chiroqni o'chiradi. Yuqorida bayon qilingan algoritm asosida mikrokontroller uchun **bosh sikl**dastur fragmentini tuzing. Inson qo'li titrashi hisobiga tugma bosilganda ulab-uzishlar soni oshib ketadi. Dasturda bu effekt mavjud emas deb hisoblash mumkin.

Namunaviy javob:

```
1.Bosh siklda davriy ravishda tugma holatini tekshirish.
ATmega16A, 16MHz

main(){
int schetchik=0;
int tugma_status=1; // 1=tugma bosilmangan, 0=tugma bosilgan
...

DDRD=0b00000010; //PORTD.1- chiqish rejimida,chiroq
uchun
//PORTD.2- kirish rejimida,tugma uchun

PORTD=0b00000100; //PORTD.1 - chiroq dastlabki holatda
o'chgan
//PORTD.2 - kirish porti manbaga ichki rezistor
// orqali ulangan

...
while ()
{
if (PORTD.2==0) //agar tugma bosilgan bo'lsa
{
if (tugma_status==1)
{
tugma_status==0;
schetchik++; // schetchik qiymatini ishiramiz
if (PORTD.1==0xff) {PORTD.1=0;} // chiroqni
o'chiramiz
else PORTD.1=0xff; //yoki yoqamiz
```

```

}
    else
    {
        //tugma bosilgan holatcha turibdi
    }
}
else tugma_status=1; //tugma bosilmagan
};

```

2. "Smart uy" loyahasiga asosan maxsus qurilma tayyorlanmoqda. Bu uskumaning vazifasi yoritish chiroqlari ish vaqtini hisoblash orqali elektr energiya iste'molini aniqlashdan iborat. Qurilma algoritmiga asosan takt tugma bosilganda elektr chiroq yonishi va mikrokontroller schetchigi bittaga oshishi kerak. Tugmaning navbatdagi bosilishi chiroqni o'chiradi. Yuqorida bayon qilingan algoritm asosida mikrokontroller uchun **taymer orqali** tugma holatini tekshirish dastur fragmentini tuzing. Inson qo'li titrashi hisobiga tugma bosilganda ulab-uzishlar soni oshib ketadi. Dasturda bu effekt mavjud emas deb hisoblash mumkin.

Namunaviy javob:

```

2. Timer orqali har 100 ms da tugma holatini tekshirish.
ATmega16A, 16MHz
...
//Global o'zgaruvchilar
int schetchik=0;
int tugma_status=1; // 1=tugma bosilmangan, 0=tugma bosilgan
...
//Timer 1 bo'yicha uzilishga ishlov berish funksiyasi
interrupt [TIM1_OVF] void timer1_ovf_isr(void)
{
    TCNT1H=0x9E58 >> 8; // Timer1ni boshlang'ich holatga
    o'tkazish
    TCNT1L=0x9E58 & 0xff;
    if (PORTD.2==0) //agar tugma bosilgan bo'lsa
    {
        if (tugma_status==1)
        {
            tugma_status==0;
            schetchik++; // schetchik qiymatini ishiramiz
            PORTD.1=(PORTD.1==0xff)?(0):(0xff);

```

```

        // chiroqni yoqamiz yoki o'chiramiz
    }
    else
    {
        //tugma bosilgan holatcha turibdi
    }
} else tugma_status=1; //tugma bosilmagan
}
...
main(){
...
DDRD=0b00000010; //PORTD.1- chiqish rejimida,chiroq
uchun
//PORTD.2- kirish rejimida,tugma uchun

PORTD=0b00000100; //PORTD.1 - chiroq dastlabki holatda
o'chgan
//PORTD.2 - kirish porti manbaga ichki rezistor
// orqali ulangan

...
// Timer1: 250 kHz, Uzilish= har 100ms da, hisoblagich to'lganda
TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) |
(0<<COM1B0) |
(0<<WGM11) | (0<<WGM10);
TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) |
(0<<WGM12) |
(0<<CS12) | (1<<CS11) | (1<<CS10);
TCNT1H=0x9E;
TCNT1L=0x58;
TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A)
|
(0<<OCIE1B) | (1<<TOIE1) | (0<<OCIE0) | (0<<TOIE0);
#asm("sei") // uzilishlarga global ruxsat berish
while ()
{
    /* tugma holati tekshirilguncha
    qurilma boshqa funksiyalarini bajarishi mumkin
    */ zahirada 100 msek vaqt bor
};

```

3."Smart uy" loyihasiga asosan maxsus qurilma tayyorlanmoqda. Bu uskumaning vazifasi yoritish chiroqlari ish vaqtini hisoblash orqali elektr energiya iste'molini aniqlashdan iborat. Qurilma algoritmiga asosan takt tugma bosilganda elektr chiroq yonishi va mikrokontroller schetchigi bittaga oshishi kerak. Tugmaning navbatdagi bosilishi chiroqni o'chiradi. Yuqorida bayon qilingan algoritm asosida mikrokontroller uchun **tashqi uzilish manbasi (external IRQ)** yordamida ishlovchi dastur fragmentini tuzing. Inson qo'li titrashi hisobiga tugma bosilganda ulab-uzishlar soni oshib ketadi. Dasturda bu effekt mavjud emas deb hisoblash mumkin.

Namunaviy javob:

```

3. "Any change" IRQ orqali tugma holatini tekshirish.
ATmega16A, 16MHz
...
//Global o'zgaruvchilar
int schetchik=0;
int tugma_status=1; // 1=tugma bosilmangan, 0=tugma bosilgan
...
/* PORTD.2 ga berilgan signal mantiqiy o'zgarganda (0->1 yoki
1->0)
uzulishlarga ishlov berish funksiyasi ishga tushadi. */
interrupt [EXT_INT0] void ext_int0_isr(void)
{
    if (PORTD.2==0) //agar tugma bosilgan bo'lsa
    {
        if (tugma_status==1)
        {
            tugma_status==0;
            schetchik++; // schetchik qiymatini ishiramiz
            PORTD^=(1<PORTD.2); // chiroqni yoqamiz yoki
o'chiramiz
        }
        else
        {
            //tugma bosilgan holatcha turibdi
        }
    } else tugma_status=1; //tugma bosilmagan
}
...

```

```

main(){
...
DDRD=0b00000010;    //PORTD.1- chiqish rejimida,chiroq
uchun
                //PORTD.2- kirish rejimida,tugma uchun
PORTD=0b00000100; //PORTD.1 - chiroq dastlabki holatda
o'chgan
                //PORTD.2 - kirish porti manbaga ichki rezistor
                //      orqali ulangan
// IRQ, INT0 : Any change,
GICR|=(0<<INT1) | (1<<INT0) | (0<<INT2);
MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (1<<ISC00);
MCUCSR=(0<<ISC2);
GIFR=(0<<INTF1) | (1<<INTF0) | (0<<INTF2);
#asm("sei") // uzilishlarga global ruxsat berish
while ()
{
    /* tugma holati tekshirilguncha
    qurilma boshqa funksiyalarini bajarishi mumkin
    */ zahirada ancha vaqt bo'ladi
};

```

4. Elektronika va sxematexnika fanlari uchun turli shakldagi signallarni hosil qiluvchi generator tayyorlash vazifasi qo'yildi. Generator bir vaqtda 3ta chiqish kanali orqali signallarni hosil qilishi kerak. Maksimal generatsiyalanadigan signal chastotasi 1Mhz. Birinchi kanalda sinusoidal funksiyaga o'xshash bo'lgan signallarni generatsiyalash dasturini AVR ATmega oilasiga mansub bo'lgan mikrokontroller asosida tuzing.

5. Elektronika va sxematexnika fanlari uchun turli shakldagi signallarni hosil qiluvchi generator tayyorlash vazifasi qo'yildi. Generator bir vaqtda 3ta chiqish kanali orqali signallarni hosil qilishi kerak. Maksimal generatsiyalanadigan signal chastotasi 1Mhz. Birinchi kanalda $y=\sin(x)$, ikkinchi kanalda $y=\cos(x)$ funksiyaga o'xshash bo'lgan signallarni generatsiyalash dasturini AVR ATmega oilasiga mansub bo'lgan mikrokontroller asosida tuzing.

6. Elektronika va sxematexnika fanlari uchun turli shakldagi signallarni hosil qiluvchi generator tayyorlash vazifasi qo'yildi. Generator bir vaqtda 3ta chiqish kanali orqali signallarni hosil qilishi

kerak. Maksimal generatsiyalanadigan signal chastotasi 1Mhz. Birinchi kanalda $y=\sin(x)$, ikkinchi kanalda $y=\cos(x)$ funksiyaga o'xshash va uchunchi kanalda to'ldirish koeffisienti $D=40\%$ bo'lgan impuls signallarni generatsiyalash dasturini AVR ATmega oilasiga mansub bo'lgan mikrokontroller asosida tuzing.

7. Klimitni nazorat qilish tizimini yaratish ustida ish olib borilmoqda. Qurilma havo harorati va namligini o'lchasi, temperature 50°C dan ko'tarilganda yoki absolyut namlik 75% dan oshganda signal beruvchi chiroq yonishi kerak. Havo temperaturasini aniqlash uchun T-tipidagi termoparadan foydalanish zarurati tug'ildi. Bir nechta passiv komponentlar va mikrokontroller asosida datchikdan foydalanib havo haroratini o'lchash sxemasi va dasturini tuzing.

8. Klimitni nazorat qilish tizimini yaratish ustida ish olib borilmoqda. Qurilma havo harorati va namligini o'lchasi, agar temperature 50°C dan ko'tarilganda yoki absolyut namlik 75% dan oshganda signal beruvchi chiroq yonishi kerak. Havoning nisbiy namligini aniqlash uchun sig'imli passiv namlik datchigidan foydalanish zarurati tug'ildi. Bir nechta passiv komponentlar va mikrokontroller asosida datchikdan foydalanib havoning namligini o'lchash sxemasi va dasturini tuzing.

9. Klimitni nazorat qilish tizimini yaratish ustida ish olib borilmoqda. Qurilma havo harorati va namligini o'lchasi, agar temperature 50°C dan ko'tarilganda yoki absolyut namlik 75% dan oshganda signal beruvchi chiroq yonishi kerak. Havo temperaturasini aniqlash uchun T-tipidagi termoparadan, havoning nisbiy namligini aniqlash uchun sig'imli passiv namlik datchigidan foydalanish zarurati tug'ildi. Bir nechta passiv komponentlar va mikrokontroller asosida datchikdan foydalanib havo harorati, namligini o'lchash va signalizatsiya uchun sxema va dastur tuzing.

Har birida 8tdan nuqtaga ega bo'lgan segmentli indikatorlar orqali 5 ta simvulli axborotni ko'rsatish talab etilmoqda. AVR ATmega oilasiga mansub bo'lgan mikrokontroller asosida qurilma uchun dastur tuzing.

Laboratoriya mashg'ulotlari №20, №21
Mavzu: O'rnatilgan ma'lumot uzatish tizimini loyihalash
(Interfeyslar USART va SPI)

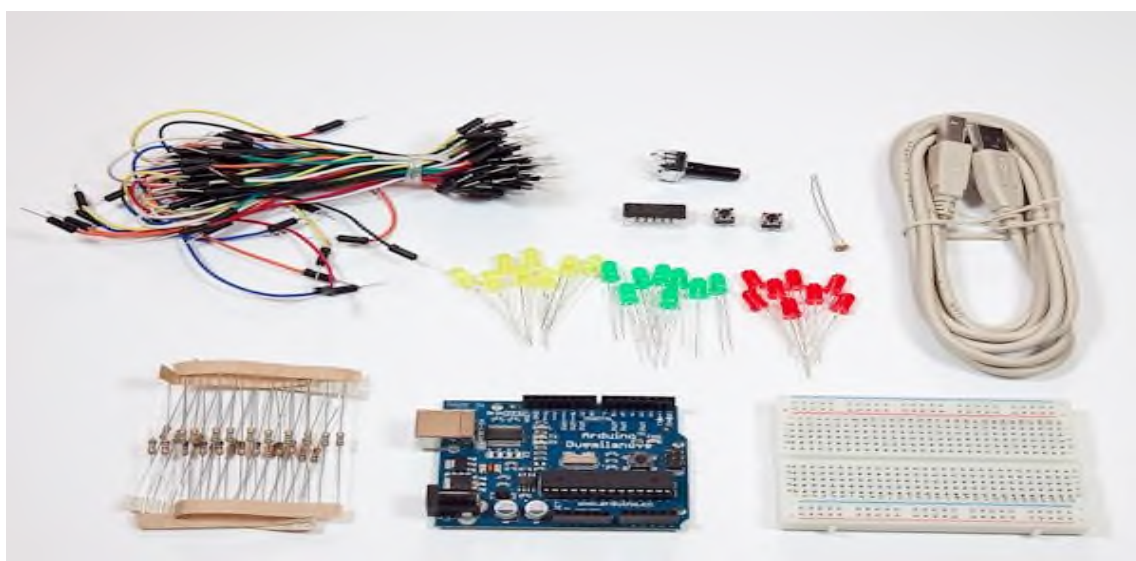
Ishdan maqsad: Kompyuter SOM porti asosida Arduino ga ulangan qurilmalarni boshqarish.

Arduino — apparat hisoblash platformasi hisoblanib, asosiy komponentlari oddiy kiritish/chiqarish platasi va Processing/Wiring dasturiy tilni qo'llaydigan dasturiy muhit hisoblanadi. Arduino avtonom interaktiv ob'ektlarni yaratishda to'g'ridan to'g'ri kompyuterga ulangan holda dasturlash jarayonini amalga oshirish mumkin bo'lgan qurilma hisoblanadi (masalan, Macromedia Flash, Processing, Max/MSP, Pure Data, SuperCollider). Mikrokontroller Atmega 328 arzon va qo'llanishga qulay hisoblanadi.

Arduino — elektron konstruktor va foydalanuvchi uchun elektron qurilmalarni yaratish mumkin bo'lgan platforma hisoblanadi. Platforma jahonda qulayligi va dasturiy tillarining oddiyligi uchun keng qo'llaniladi. Qurilma programmatorsiz USB orqali dasturlanadi.

Platadagi mikrokontroller Wiring dasturiy tilga asoslangan Arduino tilida dasturlanadi.

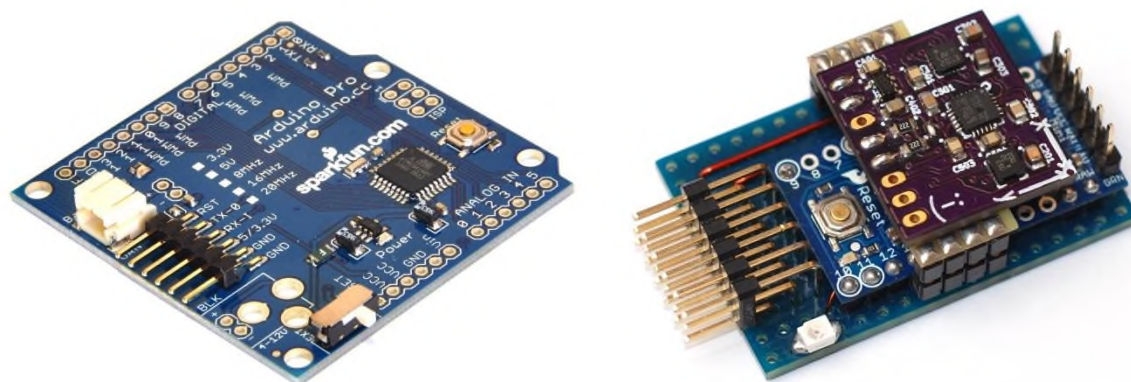
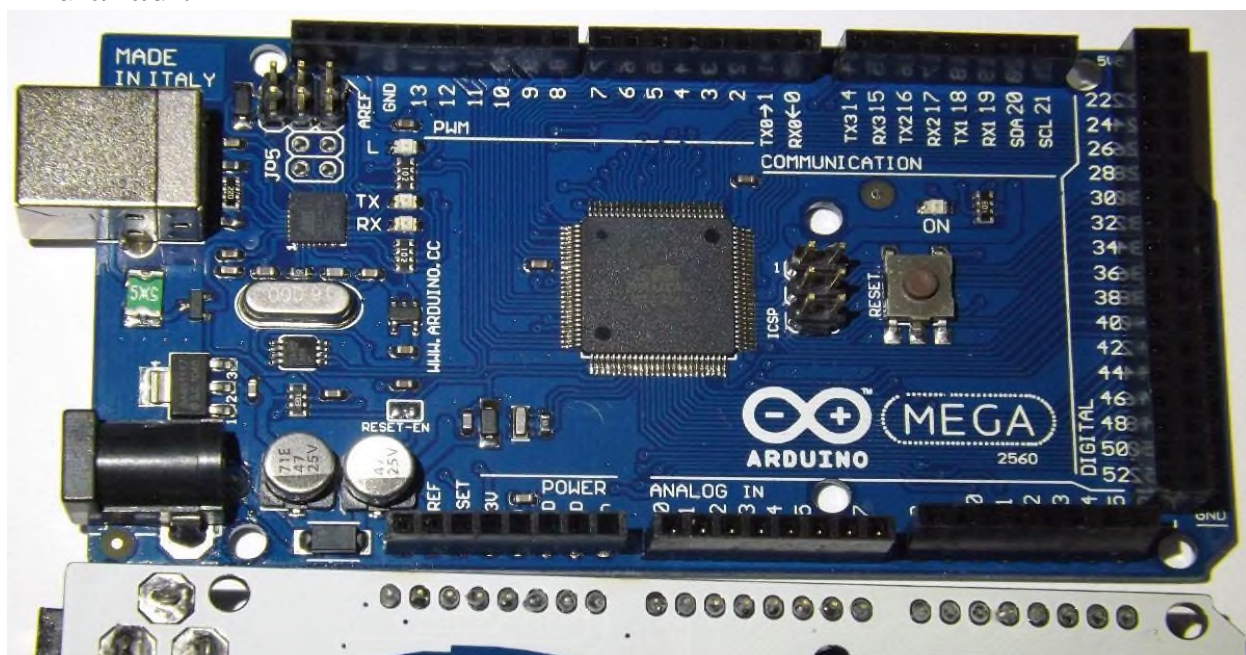
Arduino platformasi apparat qismi Arduino platformasining bir necha versiyalari mavjud bo'lib, ulardan Leonardo AT mega 32u4 mikrokontrolleri bazasi asosida yaratilgan. Uno esa Atmel AT mega 328 mikrokontrolleri asosida yaratilgan.



7.1- rasm. Arduino platformasi

Arduino platformasi versiyalari:

- **Due** — yangi plata ARM mikroprotsessori asosida 32bit Cortex-M3 ARM SAM3U4E.
- **Leonardo** — Arduino yangi platforma ATmega 32u4 mikrokontroller asosida .
- **Yun** - yangi plata, o‘rnatilgan WiFi ni qo‘llaydi ATmega 32u4 and the Atheros AR9331 asosida.
- **Micro** — AT mega32u4 asosidagi yangi loyiha .
- **Uno** —Arduino USB ko‘p foydalaniladigan versiya. Uni o‘zida standart port USB saqlaydi. Platforma o‘z navbatida kengaytirilgan platalar va foydalanuvchi turli funksiyalari bilan to‘ldirilishi mumkin.
- **Nano** — kompakt platforma hisoblanib, maket ko‘rinishida foydalaniladi. Nano kompyuterga USB Mini-B kabeli asosida ulanadi.



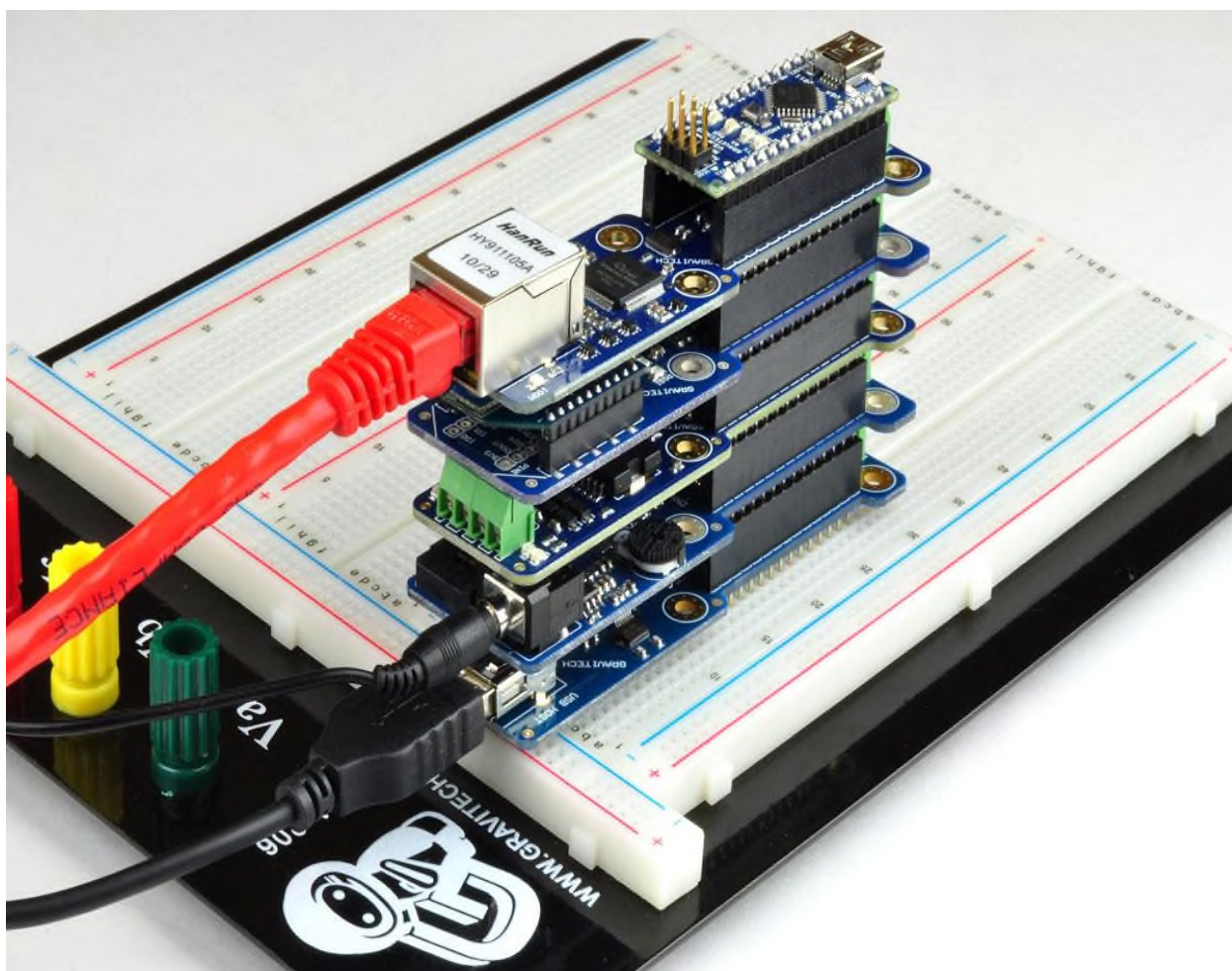
7.2.- rasm. Arduino platasi umumiy ko‘rinishi

Kengaytirilgan platalar asosan turli qurilmalarni boshqarishda va ma'lumotlar olishda foydalaniladi:

- Plata WiFi - 802.11 b/g simsiz tarmoq standarti bilan ulanishda foydalaniladi.

Plata Xbee Shield - Maxstream Xbee Zigbee modullari asosida bir necha Arduino bilan 35 metr radiusdagi maydonni va 90 metr maydondan tashqari simsiz aloqani ta'minlaydi.

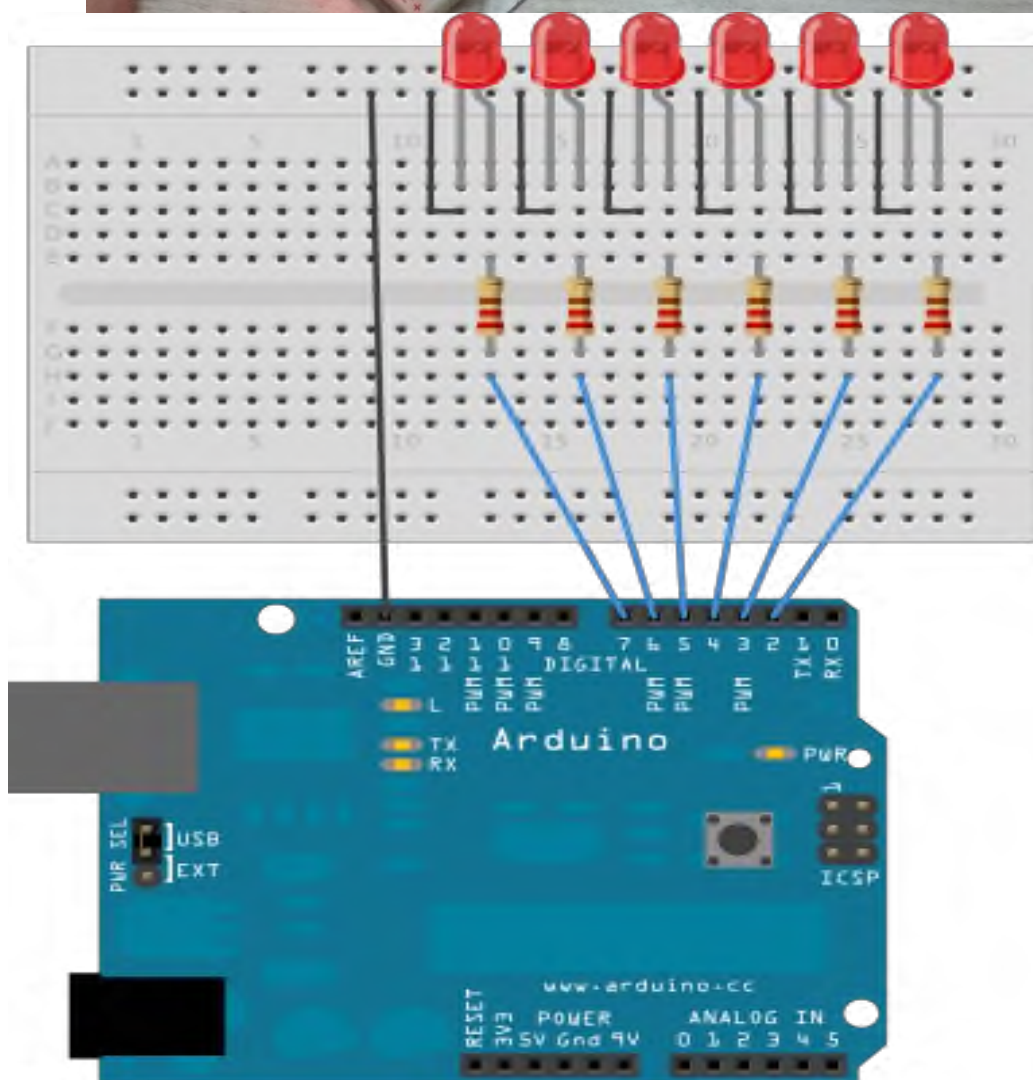
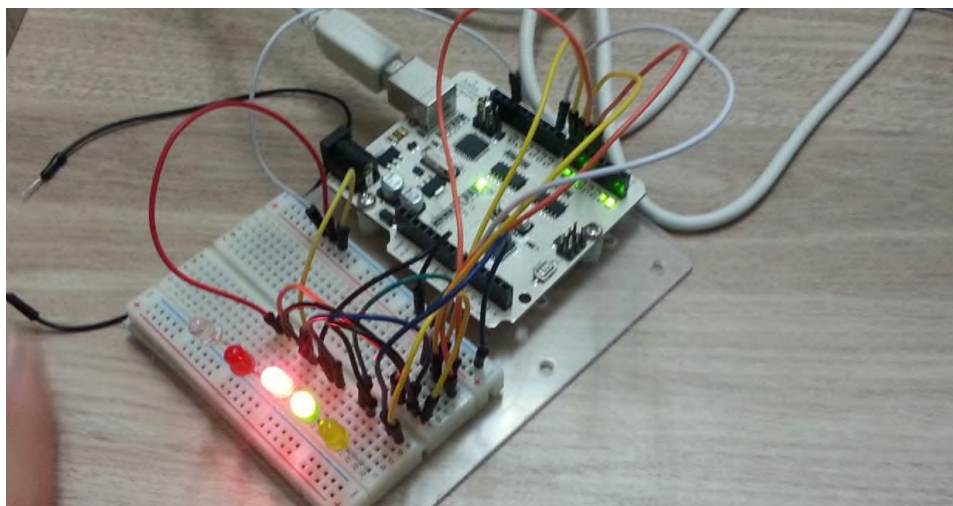
- Plata Motor Shield – doimiy tok dvigatellarini boshqarish va datchiklar holatini o'qishda foydalaniladi.
- Plata Ethernet Shield – internetga ulanishni ta'minlaydi.



8.1- rasm. Plata Ethernet Shield

Keltiriladigan masalalarda svetodiodlarni boshqarishda murakkab dasturlash jarayonlari orqali ko'rish mumkin.

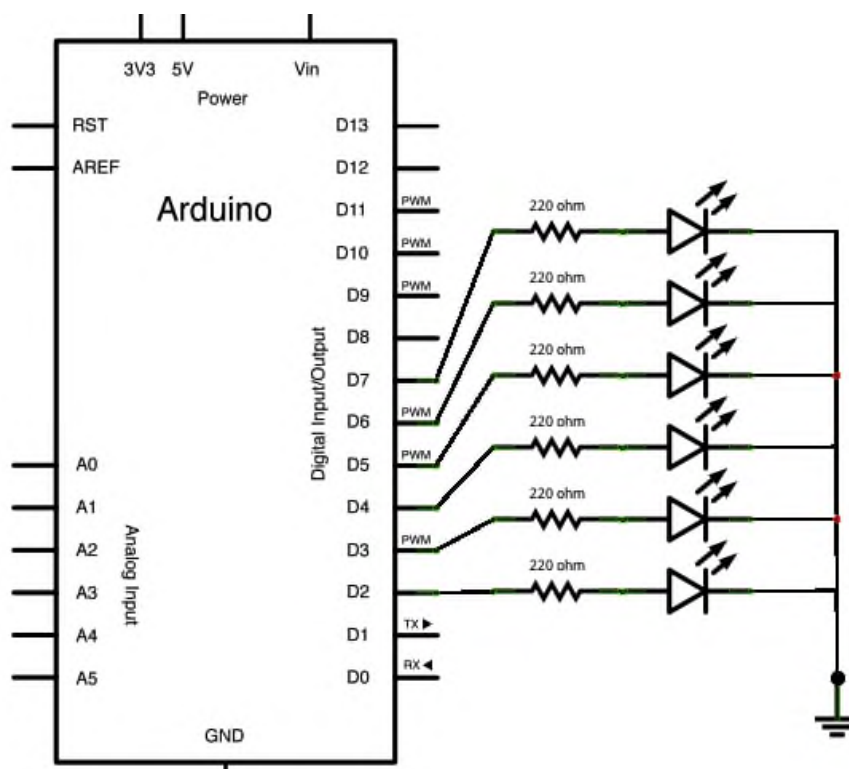
Ishni bajarish uchun 5 ta svetodiod (turli rangli) va 5 rezistor (rasm 7.1).



8.2- rasm. Arduino ga ulangan qurilmalarni boshqarish

Keltirilgan sxemada 6 svetodioldlar raqamli pinlarga 2-7 ulangan. Ushbu ulanish pin (digitalWrite) dastur asosida kuchlanish beriladi va tok rezistor asosida oqib o'tib svetodiold anod (+) ga uzatiladi. Elektrik

tok svetodiod yorug'lik energiyasiga aylanadi (fotonlar), shuning uchun katod (—) yerga (Ground) ulanishi kerak.



8.2- rasm. Arduinoga ulangan qurilmalarni boshqarish prinsipial sxemasi.

Mavjud 6 svetodioddan, 1- chini pin №2 ga (yoki pin №4 ga), 2- chi svetodiod pin №3 ga (yoki pin №5 ga), 3- chi pin №4 ga (yoki №6 ga), 4- chi pin №5 ga (yoki pin №7 ga), 5- chi pin №6 ga (yoki pin №8 ga) va 6- chi svetodiod pin №7 ga.

Qurilmani dasturiy ta'minoti Arduino IDE muhitida bajariladi va Servis menyusida Monitor porti mavjud (Ctrl+Shift+M).

COM-port asosida ulangan Arduinoga kompyuter orqali ixtiyoriy simvollarni uzatish mumkin. Uzatish natijasida ba'zi bir harakatlarni boshqarish mumkin, masalan «1» raqami svetodiodni yoqish va «0» raqami svetodiodni o'chirish vazifasini bajarladi.

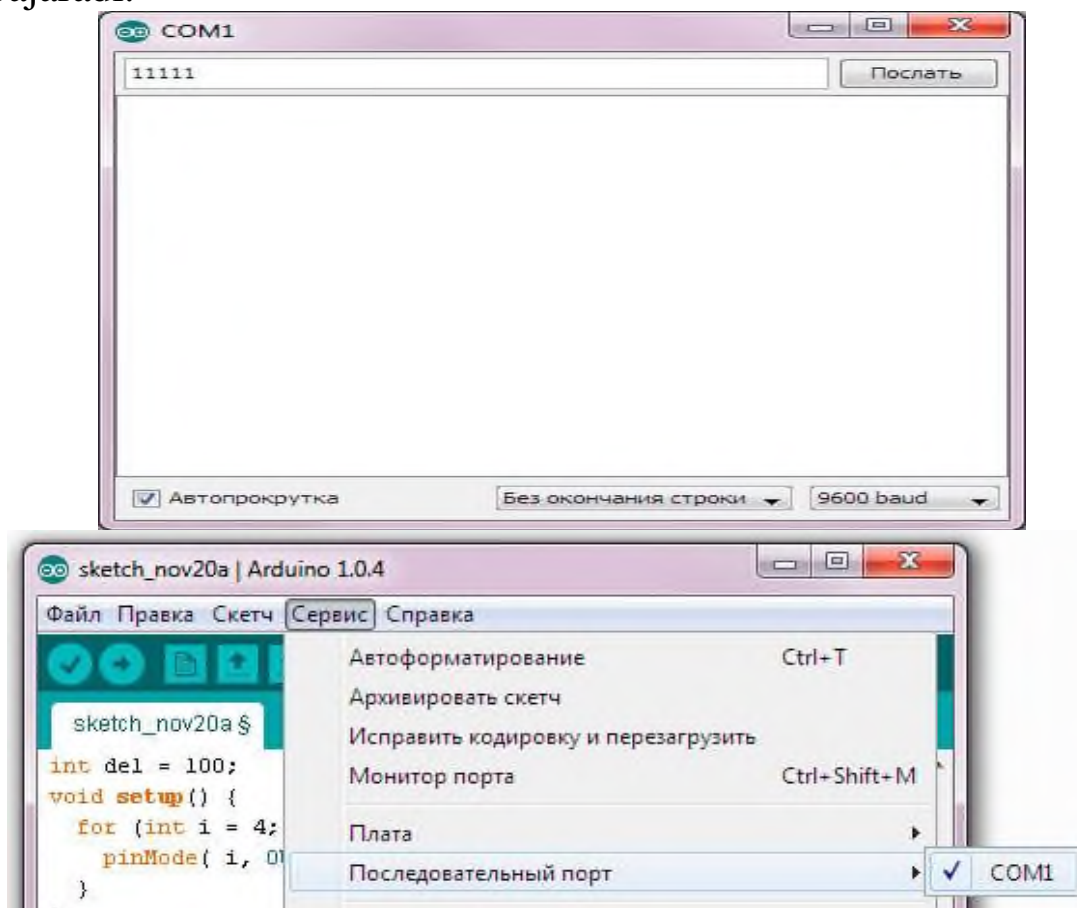
Uzatilgan simvollar to'plami svetodioldarni boshqaradi.

Monitor COM-porta orqali Arduino IDE ga uzatilgan masalan «10110» qator svetodiodlar №1, 3 va 4 yoqadi va svetodiodlar №2, №5 (pozitsiyaga nisbatan) o'chiradi.

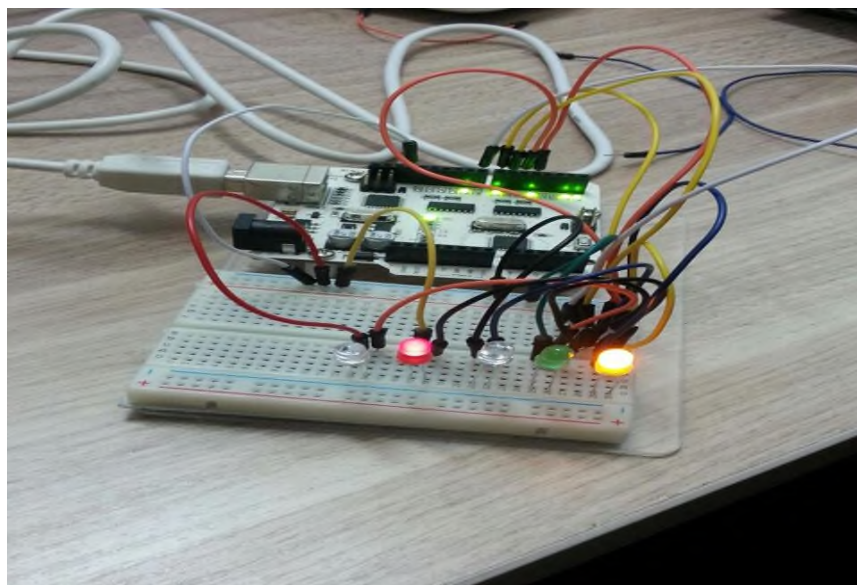
Dastur qismida klass Si – Serial dan foydalaniladi:

- **Serial.begin** – portdan (COM-porta) foydalanishga tayyorlik;
- **Serial.print** – ma'lumotlarni Arduino dan kompyuterga uzatish va Monitor portda aks ettirish;

- **Serial.read** – Arduino yordamida ma'lumotlarni kompyuterdan o'qish (masalan: «10110» qatorni o'qish). Serial funksiyasi to'plami Arduino bilan kompyuter orasidagi aloqachi vazifasini bajaradi.



8.3- rasm. Monitor porti

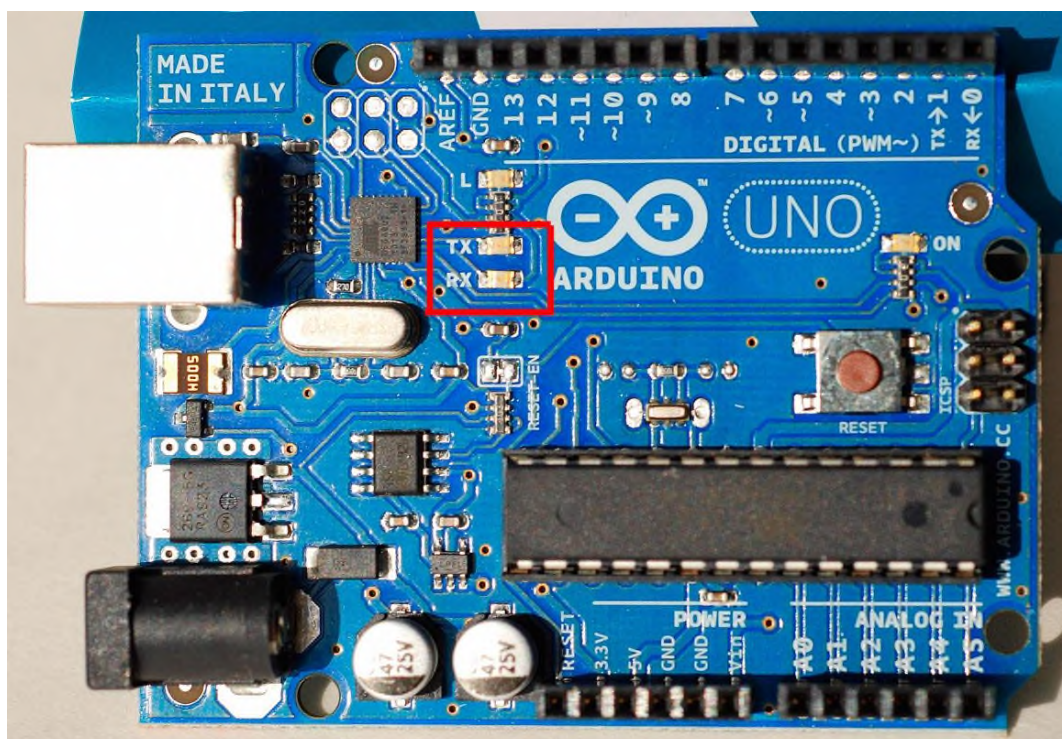


8.4-rasm. Arduino platasiga ulanish

Barcha Arduino platalari hech bo'lmaganda bitta ketma-ket portga ega port UART yoki USART. Serial ma'lumot almashish uchun raqamli portlardan 0 (RX) va 1 (TX), USB foydalaniladi.

Plata Arduino Mega o'zida 3-ta ketma ket portlarni saqlaydi:

Serial1 port 19 (RX) va 18 (TX), Serial2 port 17 (RX) va 16 (TX), Serial3 port 15 (RX) va 14 (TX). Keltirilgan portlarni kompyuter bilan aloqa uchun foydalanishda qo'shimcha adapterlar USB-to-serial ishlatiladi, chunki Mega platasi o'rnatilgan tizimiga ulanmagan. Ketma-ket interfeys orqali tashqi qurilmalar bilan aloqa o'rnatish uchun qurilmadagi TX portni tashqi qurilma RX porti bilan ulang.



8.5- rasm. Portlarni ulash jarayonlari

Nazorat savollari

1. Arduino muhitida qurilmalar bilan ishlayotganda katod nimaga ulanishi kerak?
2. **Serial.begin** buyrug'i nima vazifa bajaradi?
3. **Serial.print** buyrug'i nima vazifa bajaradi?
4. **Serial.read** buyrug'i nima vazifa bajaradi?
5. Svetodioldlarni nimalar boshqaradi?

Topshiriqlar

1. Haroratni monitoring qilish uchun kun davomida termodatchikdan 24 marotaba ma'lumot qabul qilib oldindi va mikrokontroller xotirasida massiv shaklida saqlanmoqda. Berilgan ma'lumotlarni haroratning kamayish tartibida sortirovkalab boshqa massivga saqlovchi dastur fragmentini tuzing.

2. Haroratni monitoring qilish uchun kun davomida termodatchikdan 24 marotaba ma'lumot qabul qilib oldindi va mikrokontroller xotirasida massiv shaklida saqlanmoqda. Ma'lumotlar havo harorati va qabul qilingan vaqt (soat) dan iborat. Berilgan ma'lumotlarni haroratning oshishi tartibida sortirovkalab boshqa massivga saqlovchi dastur fragmentini tuzing.

3. Haroratni monitoring qilish uchun kun davomida termodatchikdan 24 marotaba ma'lumot qabul qilib oldindi va mikrokontroller xotirasida massiv shaklida saqlanmoqda. Ma'lumotlar havo harorati va qabul qilingan vaqt (soat:min:sek) dan iborat. Berilgan ma'lumotlarni haroratning oshishi tartibida sortirovkalab boshqa massivga saqlovchi dastur fragmentini tuzing.

4. Ilmiy-taqdiqot laboratoriyasidagi maxsus kabinada turli gazlardan iborat muhit hosil qilingan. Tarkibini barqaror saqlab turish maqsadida o'rtanilgan tizim ishlab chiqish zarur. Kabina ichida elektron qurilmalardan foydalanish imkoni mavjud emas. Kabinaga kirish eshigi ochilib 5 sekund ichida yopilmasa gazlar tarkibi o'zgarishi ni hisobga olib belgilangan vaqtdan oshganda signalizatsiya ishga tushishi zarur bo'lgan dasturni bosh siklda tuzing.

5. Ilmiy-taqdiqot laboratoriyasidagi maxsus kabinada turli gazlardan iborat muhit hosil qilingan. Tarkibini barqaror saqlab turish maqsadida o'rtanilgan tizim ishlab chiqish zarur. Kabina ichida elektron qurilmalardan foydalanish imkoni mavjud emas. Kabinaga kirish eshigi ochilib 5 sekund ichida yopilmasa gazlar tarkibi o'zgarishi ni hisobga olib belgilangan vaqtdan oshganda signalizatsiya ishga tushishi zarur bo'lgan dasturni taymer asosida tuzing.

6. Ilmiy-taqdiqot laboratoriyasidagi maxsus kabinada turli gazlardan iborat muhit hosil qilingan. Tarkibini barqaror saqlab turish maqsadida o'rtanilgan tizim ishlab chiqish zarur. Kabina ichida elektron qurilmalardan foydalanish imkoni mavjud emas. Kabinaga kirish eshigi ochilib 5 sekund ichida yopilmasa gazlar tarkibi o'zgarishi ni hisobga

olib belgilangan vaqtdan oshganda signalizatsiya ishga tushishi zarur bo'lgan dasturni tashqi uzilishlar (external IRQ) asosida tuzing.

7. Aqlli ventilyator loyihasini amalga oshirish uchun maket tayyorlanmoqda. Kichiq quvvatli elektromotorni turli tezlikda boshqarish dasturini bosh siklda AVR ATmega oilasiga mansub bo'lgan mikrokontroller asosida tuzing.

8. Aqlli ventilyator loyihasini amalga oshirish uchun maket tayyorlanmoqda. Kichiq quvvatli elektromotorni turli tezlikda boshqarish dasturini taymer yordamida AVR ATmega oilasiga mansub bo'lgan mikrokontroller asosida tuzing.

9. Aqlli ventilyator loyihasini amalga oshirish uchun maket tayyorlanmoqda. Qurilma DS18B20 termodatchigi va kichiq quvvatli elektromotordan iborat. Harorat 25°C dan 35°C gacha o'zgarganda elektromotor aylanish tezligini 50% dan 100% gacha o'zgartiradigan dasturini AVR ATmega oilasiga mansub bo'lgan mikrokontroller asosida tuzing.

10. Signalizatsiya tizimini ishlab chiqarish masalasi qo'yildi. Unga ko'ra maxsus xona eshigiga o'rnatiladigan signalizatsion qurilma eshik ochilgach o'rnatilgan vaqtdan oshib ketganda yorug'lik indikatsiyasini amalga oshirish kerak. Eshik ochilib yopilishini aniqlash maqsadida gerikondan foydalanilmoqda. AVR ATmega oilasiga mansub bo'lgan mikrokontroller asosida dastur tuzing.

11. Signalizatsiya tizimini ishlab chiqarish masalasi qo'yildi. Unga ko'ra maxsus xona eshigiga o'rnatiladigan signalizatsion qurilma eshik ochilgach o'rnatilgan vaqtdan oshib ketganda yorug'lik va tovush indikatsiyasini amalga oshirish kerak. Eshik ochilib yopilishini aniqlash maqsadida gerikondan foydalanilmoqda. AVR ATmega oilasiga mansub bo'lgan mikrokontroller asosida dastur tuzing.

12. Signalizatsiya tizimini ishlab chiqarish masalasi qo'yildi. Unga ko'ra maxsus xona eshigiga o'rnatiladigan signalizatsion qurilma eshik ochilgach o'rnatilgan vaqtdan oshib ketganda yorug'lik va tovush indikatsiyasini amalga oshirish kerak. Eshik ochilib yopilishini aniqlash maqsadida gerikondan foydalanilmoqda. Signalizatsiya ishga tushish vaqti potensiometr yordamida o'rnatiladi. AVR ATmega oilasiga mansub bo'lgan mikrokontroller asosida dastur tuzing.

13. Kichik ilmiy laboratoriya uchun magnit maydon kuchlanganligini aniqlovchi katta aniqlikka ega bo'lmagan qurilma tayyorlashga ehtiyoj tug'ildi. Buning uchun olimlar bir qancha

gerkonlardan foydalanishni maqul ko‘rishdi. Ushbu qurilmaning ishlash algoritmini tuzing va dastur fragmentini yozing.

14. Kichik ilmiy laboratoriya uchun magnit maydon kuchlanganligini aniqlovchi katta aniqlikka ega bo‘lmagan qurilma tayyorlashga ehtiyoj tug‘ildi. Buning uchun olimlar bir qancha gerkonlardan foydalanishni maqul ko‘rishdi. Olingan natijalarni yorug‘lik diodlaridan iborat bo‘lgan lineyka asosida tasvirlashmoqchi. Ushbu qurilmaning ishlash algoritmini tuzing va dastur fragmentini yozing.

15. Kichik ilmiy laboratoriya uchun magnit maydon kuchlanganligini aniqlovchi katta aniqlikka ega bo‘lmagan qurilma tayyorlashga ehtiyoj tug‘ildi. Buning uchun olimlar bir qancha gerkonlardan foydalanishni maqul ko‘rishdi. Olingan natijalarni yorug‘lik diodlaridan iborat bo‘lgan lineyka asosida tasvirlab, UASRT orqali kompyuterga uzatishmoqchi. Ushbu qurilmaning ishlash algoritmini tuzing va dastur fragmentini yozing.

16. Magnit maydon kuchlanganligi asosida ishlovchi gerkon yordamida maxsus yashik ochilganligi yoki yopiq ekanligini aniqlovchi dastur fragmentini AVR ATmega oilasiga mansub bo‘lgan mikrokontroller asosida tuzing.

17. Magnit maydon kuchlanganligi asosida ishlovchi gerkon yordamida maxsus yashik ochilganligi yoki yopiq ekanligini aniqlovchi va bunga mos ravishda yorug‘lik indikatsiyalarini namoyon qiluvchi dastur fragmentini AVR ATmega oilasiga mansub bo‘lgan mikrokontroller asosida tuzing.

18. Magnit maydon kuchlanganligi asosida ishlovchi gerkon yordamida maxsus yashik ochilganligi yoki yopiq ekanligini aniqlovchi va bunga mos ravishda yorug‘lik va tovush indikatsiyalarini namoyon qiluvchi dastur fragmentini AVR ATmega oilasiga mansub bo‘lgan mikrokontroller asosida tuzing.

19. Aqlli signalizatsiya loyihasida ishga tushish vaqtini yorug‘lik diodlarida vizual tasvirlash masalasi qo‘yildi. Unga ko‘ra signalizatsiya hisob vaqti boshlanib toki ishga tushguncha yorug‘lik diodlari o‘z rangini qoradan oqqacha asta sekinlik bilan o‘zgartirishi kerak. Buning uchun RGB yorug‘lik diodlaridan foydalaniladi. Ishga tushish vaqti dasturlash jarayonida o‘rnatiladi va qurilma ish davomida o‘zgarmay qoladi. AVR ATmega oilasiga mansub bo‘lgan mikrokontroller asosida dastur fragmentini tuzing.

20. Aqlli signalizatsiya loyihasida ishga tushish vaqtini yorug'lik diodlarida vizual tasvirlash masalasi qo'yildi. Unga ko'ra signalizatsiya hisob vaqti boshlanib toki ishga tushguncha yorug'lik diodlari o'z rangini qoradan oqqacha asta sekinlik bilan o'zgartirishi kerak. Buning uchun RGB yorug'lik diodlaridan foydalaniladi. Ishga tushish vaqti qurilma ish davomida o'zgarishi mumkin. AVR ATmega oilasiga mansub bo'lgan mikrokontroller asosida dastur fragmentini tuzing.

21. Aqlli signalizatsiya loyihasida ishga tushish vaqtini yorug'lik diodlarida vizual tasvirlash masalasi qo'yildi. Unga ko'ra signalizatsiya hisob vaqti boshlanib toki ishga tushguncha yorug'lik diodlari o'z rangini dastlab qoradan oqqacha, so'ng sariqqa, keyin qizilga asta sekinlik bilan o'zgartirishi kerak. Buning uchun RGB yorug'lik diodlaridan foydalaniladi. Ishga tushish vaqti qurilma ish davomida o'zgarishi mumkin. AVR ATmega oilasiga mansub bo'lgan mikrokontroller asosida dastur fragmentini tuzing.

22. Aqlli klavyatura loyihasi uchun maxsus buyruqlarni kompyuterga uzatuvchi tugmalardan foydalanish ko'zda tutilmoqda. Bunda bir tugma bosilganda "a" dastur, ikkinchi tugma bosilganda "b" dastur ishga tushishi kerak. AVR ATmega oilasiga mansub bo'lgan mikrokontroller asosida dastur fragmentini tuzing.

GLOSSARIY

ALU (Arithmetic Logic Unit)- Arifmetik mantiqiy qurilma
APS (Application Program Section)- Amaliy dasturlash bo'limi
ASSR (Asynchronous Mode Status Register)- Taymer asinxron rejimi
CISC (Complicated Instruction Set)- Rivojlangan arxitektura buyruglar tizimi
CLR (Clear)- Tozalash
DDRA (Data Direction Register)- Port A yo'naltiruvchi registr
DDRB (Data Direction Register)- Port B yo'naltiruvchi registr
DDRC (Data Direction Register)- Port C yo'naltiruvchi registr
DDRD (Data Direction Register)- Port D yo'naltiruvchi registr
EEARH (EEPROM Address Register) - EEPROM adres registry katta bayt
EEARL EEPROM (Address Register) - EEPROM adres registry kichik bayt
EECR EEPROM (Control Register) EEPROM boshqaruv registri
EEDR EEPROM (Data Register)- EEPROM ma'lumot registri
EEPROM (Electrically EPROM)- Elektr o'chirishli xotira
EPROM (Erasable Programmable ROM)- Qayta dasturlanuvchi xotira
FB (Fusebit)- Bit-predoxranitel
Flash (Flash memory)- Flesh xotira
GIFR (General Interrupt Flag Register)- uzilishlar bayroq registri
GIMSK (General Interrupt Mask Register)-Uzilishlar niqob registri
GPR (General Purpose Registers)-Umumiy foydalanuvchi registr
IN (Input)- Kirish
INC (Increment)- 1 ga oshirish
IOR (Input Output Registers)- Kiritish chiqarish registri
IR (Instruction register)- Prosessor yadrosi qo'llanmasi registri
LB (Lockbit)- Blokirovka biti
MCU (Microprocessor Core Unit)- Prosessorli yadro
MCUCR MCU (general Control Register)- Prosessorli yadro umumiy boshqaruv registri
MCUSR MCU(general Status Register)- Prosessorli yadro umumiy status registri
MISO (Master In Slave Out)- Boshqaruv kirishli- yordamchi chiqish SPI interfeysi
MOSI (Master Out Slave In)- Boshqaruv hiqish - yordamchi c kirishli SPI

OC (Output Compare)- Chiqish solishtirgichi
 OSCCAL (Oscillator Calibration Register)- Kalibrovka registry rezonatori
 PINA (Input Pins, Port A)- A port kirish registri
 PINB (Input Pins, Port B)- B port kirish registri
 PINC (Input Pins, Port C)- C port kirish registri
 PIND (Input Pins, Port D)- D port kirish registri
 POR (Power-on Reset) – Manba Yo‘qligida qayta yuklash
 PORTA (Data Register, Port A) - A port ma’lumot registri
 PORTB (Data Register, Port B)- B port ma’lumot registri
 PORTC (Data Register, Port C)- C port ma’lumot registri
 PORTD (Data Register, Port D)- D port ma’lumot registri
 RET (Return)- Qaytish
 RISC (Reduced Instruction Set Computer)- Qisqartirilgan komandalar tizimi
 ROM (Read Only Memory)- Doimiy xotira
 SP (StackPointer)- Stek ko‘rsatkichi
 SPH (Stack Pointer High)- Katta bayt,stek ko‘rsatkichi
 SPI (Serial Peripheral Interface)- Ketma ket periferiya interfeysi
 SPL (Stack Pointer Low)- Kichik bayt,stek ko‘rsatkichi
 SRAM (Static Random Access Memory)- Statik operativ xotira
 SREG (Status register)- Xolatlar registri
 TCNT1H (Timer/Counter1 High-byte)- Taymer/hisoblagich 1, kata bayt
 TCNT1L (Timer/Counter1 Low-byte)- Taymer/hisoblagich 1, kichik bayt
 TIFR (Timer/Counter Interrupt Flag)- Uzilishlarni tashkil qilish taymer hisoblagichi
 UART (Universal Asynchronous Receiver)- Universal qabul qilib uzatgich
 WDT (Watchdog Timer)- Qo‘riqchi taymer

FOYDALANILGAN ADABIYOTLAR

1. O‘zbekiston Respublikasi Prezidentining "O‘zbekiston Respublikasini yanada rivojlantirish bo‘yicha harakatlar strategiyasi to‘g‘risida"gi farmoni. (2017-yil 7-fevral, PF-4947-son).
2. O‘zbekiston Respublikasi Prezidentining "Oliy ta‘lim tizimini yanada rivojlantirish chora-tadbirlari to‘g‘risida"gi qarori (2017 yil 20 aprel, PQ-2909-son).
3. O‘zbekiston Respublikasi Prezidentining "Oliy ma‘lumotli mutaxassislar tayyorlash sifatini oshirishda iqtisodiyot sohalari va tarmoqlarining ishtirokini yanada kengaytirish chora-tadbirlari to‘g‘risida"gi qarori (2017 yil 27 iyul, PQ-3151-son).
4. Yunusov J.Yu., Abasxanova X.Yu. Raqamli qurilmalar va mikroprosessor tizimlari .O‘quv qo‘llanma. Itisod , Toshkent 2010yil, .-256 b.
5. Abasxanova X.Yu., Amirsaidov U.B. Mikroprotsesszorlar. Oliy o‘quv yurtlari uchun o‘quv qo‘llanma. Toshkent 2017 yil. -350 b.
6. М.Б.Лебедев CodeVision AVR Пособие для начинающих .М-Додека-XXI.2010. -592 с.
7. Ю.В. Новиков. Основы микропроцессорной техники. Учебное пособие. Москва: ИНТУИТ, 2012. - 357 с.
8. Жмакин А.П. Архитектура ЭВМ.-СПб.: БХВ-Петербург, 2014. -320 с.
9. Преснухин Л.Н. Микропроцессоры: В 3 кн. Кн. 2: Средства сопряжения. Контролирующие и управляющие системы: Учеб.для техн. Вузов /В.Д. Вернер, Н.В. Воробев, А.В. Горячев и др.; Под ред. Л.Н. Преснухина. - Мн.: Выш.шк., 2017.-303 с
10. Зарубин А.А. Микропроцессорное программное управления. Архитектура IХА. Методические рекомендации к практическим занятиям. СПбГУТ.- СПб, 2004.
11. Белов А.В. Создаем устройства на микроконтроллерах.- СПб.: Наука и техника, 2007.- 304 с
12. Хартов В.Я. Микроконтроллеры AVR. Практикум для начинающих.- М.: МГТУ им. Н.Э.Баумана, 2007.- 240 с
13. Василев А.Е. Микроконтроллеры. Разработка встраиваемых приложений: Учеб. Пособие. СПб: СПбГПУ, 2003.- 210 с.
14. Ramesh Chopra "Microcontroller Based Projects" EFY

- Enterprises Pvt Ltd. 2010
15. Ю.Ревич. Практическое программирование микроконтроллеров AtmelAVR на языке ассемблера
 16. Брей.Б. Применение микроконтроллеров PIC18. Архитектура, программирование и построение интерфейсов с применением С и ассемблера.Пер.с англ.-К: "МК пресс",Спб Корона-Век .2008г. -576 с.
 17. А.А. Зубарев. Ассемблер для микроконтроллеров AVR.Учебное пособие.-Омск: Изд-воСиБАДИ, 2007. - 112 с.
 18. А.В.Евстифеев. Микроконтроллеры AVR семейств Tiny М-Додека-XXI.2008/-560с.
 19. М.С.Голубцов. Микроконтроллеры AVR:от простого к сложному.М.:Солон-Пресс 2017.-288 с.
 20. А.В.Евстифеев. Микроконтроллеры AVR семейств Tiny М-Додека-XXI.2018. -560 с.
 21. Б.Брей. Применение микроконтроллеров PIC. Архитектура,программирование и построение интерфейсов с применением С и ассемблера .Пер с англис.МК:.Пресс., СпбКОРОНА ВЕК 2008.-576 с.

Axborot resurs manbalari

<http://www.elkutubhona.narod.uz>

<http://ziyonet.uz>

<http://www.intuit.ru/studies/courses/3/3/info>

MUNDARIJA

KIRISH	3
1. Mikrokontrollerlarni tashkillashtirish	
1.1 Mikrokontroller arxitekturasi	5
1.2 Mikrokontroller protsessorli yadrosi.....	7
1.3 Mikrokontroller buyruq tizimining xususiyatlari	12
1.4 Mikrokontrollerlarda xotirani tashkil etilishi va u bilan ishlash	23
1.4.1 Mikrokontroller dasturiy xotirasi	24
1.4.2 Mikrokontroller ma'lumot xotirasi	26
1.5 Mikrokontroller registrlari	27
2. Mikrokontrollerlarni tashqi muhit bilan aloqasini tashkillashtirish	
2.1. Mikrokontrollerlarda axborot kiritish-chiqarish portlarini tashkillashtirish va strukturasi.....	28
2.2. Taymer-hisoblagichlar.....	30
2.3. Jarayonlarga ishlov berish bloklari	32
2.4. Uzilishli hodisalarga xizmat ko'rsatish bloklari.....	33
3. Bir kristalli mikrokontrollerlar	
3.1. PIC mikrokontrollerlari tarkibi, parametrlari va tashkillashtirish xususiyatlari.....	35
3.2. PIC mikrokontrollerlari apparat vositalarining strukturasi va imkoniyatlari.....	38
3.3. PIC mikrokontrollerlari buyruqlar tizimi va maxsus funksiyalari.....	40
3.4. AVR mikrokontrollerlari tarkibi, parametrlari va tashkillashtirish xususiyatlari.....	50
3.5. ATMEga128 mikrokontrollerining strukturasi va imkoniyatlari	56
3.6 Mikrokontrollerlar asosidagi tizimlarning imkoniyatlarini kengaytirish va sifatini oshirish uchun maxsus funksiyalar ...	58
4. Mikrokontrollerlarlar asosida qurilmalarini loyihalashtirish	
4.1. Mikrokontrollerlarning dasturiy ta'minotini yaratish va apparat ta'minoti bilan integrasiyalash	62
4.2. BASCOM-AVR dasturiy muxitida loyihalashtirish.....	64
4.3. ATMEL STUDIO 6 dasturiy muxitida loyihalashtirish	72
4.4. Fritzing dasturiy vositalarida sozlash.....	77
4.5. AVRDUDE dasturiy vositalarida sozlash	81
4.6. Code Vision AVR muxiti va uning imkoniyatlari.....	83
4.7. Arduino imkoniyatlari	87
BOBLAR BO'YICHA NAZORAT TESTLARI	106
5. Laboratoriya mashg'ulotlari bo'yicha ko'rsatma va tavsiyalar	121
GLOSSARIY	195
FOYDALANILGAN ADABIYOTLAR.....	197

**X.YU.ABASXONOVA,
M.B. MIRZAYEVA,
S.S. PARSIYEV**

MIKROPROTSESSOR

(O‘quv qo‘llanma)

Toshkent – «NIHOL PRINT» OK – 2021

Muharrir: A.Tog‘ayev
Tex. muharrir: F.Tog‘ayeva
Musavvir: B.Esanov
Musahhiha: O.Muxammadiyeva
Kompyuterda
sahifalovchi: G.Tog‘ayeva

9323



№ 7439-765f-47f1-7ea1-a683-4648-1314.
Bosishga ruxsat etildi: 4.06.2021. Bichimi 60x841 /16.
Shartli bosma tabog‘i 12,75. Nashr bosma tabog‘i 12,5.
Adadi 100. Buyurtma № 64.

«Nihol print» Ok da chop etildi.
Toshkent sh., M. Ashrafiy ko‘chasi, 99/101.