



CODE VISION AVR DASTURIDA MIKROKONTROLLER

Imamnazarov Erkin¹, Ikromov Islomjon², Hasanov Xojiakbar³

¹Namangan muhandislik qurilish instituti katta o'qituvchi, ^{2,3} talaba

<https://doi.org/10.5281/zenodo.6584450>

MAQOLA TARIXI

Qabul qilindi: 10-may 2022

Ma'qullandi: 14- may 2022

Chop etildi: 24- may 2022

KALIT SO'ZLAR

Mikrokontroller, dasturlash, AVR, mikroprotessor, dasturiy ta'minot

ANNOTATSIYA

Ushbu maqolada zamonaviy dasturlash tillaridan biri – C haqida umumiy va foydalanish uchun zarur bo'lgan ma'lumotlar, hamda, Code vision AVR dasturida mikrokontrollerlar oilasiga mansub bir necha turdagi mikrokontrollerlar xususida so'z yuritiladi

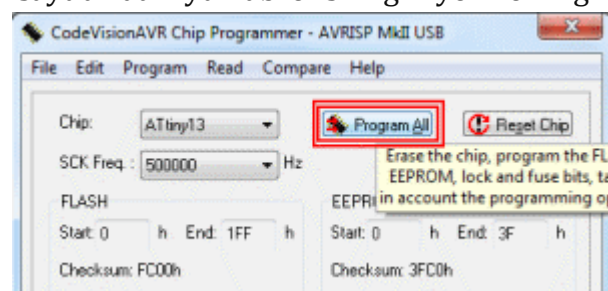
Mikrokontroller (MCU) - Elektron qurilmalarni boshqarish uchun mo'ljallangan mikrohiy hisoblanadi. Oddiy mikrokontroller protessorning funktsiyalarini birlashtiradi va periferik asboblari: RAM va ROM bo'lishi mumkin. Shaxsiy kompyuterlarda qo'llanilgan an'anaviy protessorlar o'rniga bitta chipdan foydalanish, shaxsiy kompyuterlarda qo'llanilganidek, mikrokontrollerlar asosida yaratilgan qurilmalarni quvvat sarfini va qurilmalarni sezilarli darajada kamaytiradi. Mikroprotessor tizimi bu bir yoki bir nechta qurilmadan iborat funktsional to'ldirilgan mahsulot, asosan mikroprotessor va / yoki mikrokontrollerdir.

Mikrokontrollerlarni dasturlash bilan birinchi bo'lib shug'ullangan va ilgari C tili bilan ko'pchilikka tanish bo'lmaganligi uchun kichik kirish ma'lumotlar yozish hozirgi kunda muhim hisoblanadi. CodeVisionAVR bilan ishlash haqida umumiy tasavvurga ega bo'lish uchun

batafsil izlanish va shug'ullanish talab etiladi.

Shuningdek, biz qanday harakatlar qilsak ham, oxir-oqibat, barchasi mikrokontrollerning dasturiy ta'minotiga to'g'ri keladi. Mikrodestur jarayonining o'zi quyidagicha: ma'lum bir dastur yordamida proshivka fayli tanlanadi, parametrlar tanlanadi, tugma bosiladi va proshivka to'g'ridan-to'g'ri yonadi, bu aslida nusxadir. Musiqa yoki hujjatlarni kompyuterdan USB flesh-diskga nusxalash kabi jarayonning fizikasi ham bir xil.

Mikrodesturning o'zi kengaytmaga ega. hex va mikrokontroller uchun tushunarli bo'lgan birlik va nol ko'rinishidagi ko'rsatmalar to'plamidir. Mikrodesturni qayerdan olsam bo'ladi? Siz uni elektronika saytlaridan yuklab olishingiz yoki o'zingiz





yo'zishingiz mumkin. Siz uni ishlab chiqish muhiti deb ataladigan maxsus dasturlarda yo'zishingiz mumkin. Menga eng yaxshi ma'lum bo'lgan AVR Studio, IAR, CodeVision, WinAVR ... Bu muhitlarning qaysi biri yaxshiroq yoki yomonroq ekanligini aytish mumkin emas, ularning har biri o'ziga xosdir. Aytishimiz mumkinki, bu dasturlar asosan qulayligi, dasturlash tili va narxi bilan farqlanadi. Ushbu saytda faqat CodeVision ko'rib chiqiladi.

Muhitni aniqlashdan so'ng, endi proshivka yo'zish jarayoni bilan shug'ullanish kerak bo'ladi. CodeVision da birinchi navbatda loyiha yaratishin uchun kerak. U kod ustasi yordamida yoki bo'sh holda yaratilishi mumkin. Har qanday holatda, ishlatiladigan mikrokontroller turini tanlashingiz va uning chastotasini belgilashingiz kerak. Sehgardan foydalanganda sizdan dastlabki sozlamalarni tanlash va sozlamalar bilan manba kodini yaratish so'raladi. Keyinchalik, ushbu kodni tahrirlashingiz mumkin bo'lgan oyna paydo bo'ladi. Garchi siz o'z manba kodingizni bloknotga yo'zishingiz va keyin uni sozlamalarda loyihaga biriktirishingiz mumkin.

Manba kodi fayli dasturlash tilidagi buyruqlar to'plamidir, CodeVision ning vazifasi bu buyruqlarni ikkilik kodga tarjima qilish, sizning vazifangiz esa ushbu manba kodini yo'zishdir. CodeVision C tilini tushunadi, manba kodi fayllari ".c" kengaytmasiga ega. Ammo CodeVision - da C tilida ishlatilmaydigan ba'zi konstruktsiyalar mavjud, ular uchun ko'p dasturchilar uni yoqtirmaydi va foydalaniladigan til C o'xshash deb ataladi. Biroq, bu jiddiy loyihalar yo'zishga to'sqinlik qilmaydi. Ko'pgina misollar, kod generatori, katta kutubxonalar to'plami CodeVision -ga katta plyus beradi. Yagona salbiy tomoni

shundaki, u pullik, garchi cheklangan kodli bepul versiyalar mavjud.

Manba kodida ishlatiladigan mikrokontrollör turi va asosiy funksiyasi bo'lgan sarlavha bo'lishi kerak. Masalan, ATtiny13 ishlatiladi

```
#include <tiny13.h>
void main(void)
{
};
```

Asosiy funktsiyadan oldin siz kerakli kutubxonalarni ulashingiz, global o'zgaruvchilarni, konstantalarni, sozlamalarni e'lon qilishingiz mumkin. Kutubxona - bu alohida fayl bo'lib, odatda ".h" kengaytmali, oldindan yozilgan kodga ega. Ba'zi loyihalarda bizga ushbu kod kerak bo'lishi mumkin, ammo boshqalarda bizga kerak emas. Misol uchun, bitta loyihada biz LCD -dan foydalanamiz display, ikkinchisida esa biz foydalanmaymiz. Siz kutubxonani "alcd.h" LCD display bilan ishlash uchun ulashingiz mumkin, masalan:

```
#include <mega8.h>
#include <alcd.h>
void main(void)
{
};
```

O'zgaruvchilar xotiraning ba'zi qiymatlarini qo'yish mumkin bo'lgan sohalardir. Misol uchun, biz ikkita raqam qo'shdik, natijani kelajakda ishlatish uchun biror joyda saqlash kerak. Avval siz o'zgaruvchini e'lon qilishingiz kerak, ya'ni. unga xotira ajrating, masalan: int i=0; bular. biz i o'zgaruvchini e'lon qildik va unga 0 qiymatini qo'ydik, int o'zgaruvchining turi yoki boshqacha aytganda, ajratilgan xotira hajmini bildiradi. Har bir o'zgaruvchi turi faqat ma'lum bir qiymat oralig'ini saqlashi mumkin. Misol uchun, int -32768 dan 32767 gacha raqamlar yozilishi mumkin. Agar kasr qismi



bo'lgan raqamlardan foydalanish kerak bo'lsa, u holda o'zgaruvchi float sifatida e'lon qilinishi kerak, belgilar uchun char turidan foydalaning.

bit, _bit 0 или 1

char от -128 до 127

unsigned char от 0 до 255

int от -32768 до 32767

unsigned int от 0 до 65535

long int от -2147483648 до 2147483647

unsigned long int от 0 до 4294967295

float от $\pm 1.175e-38$ до $\pm 3.402e38$

Foydalanuvchi manba kodining istalgan qismida sharh yozishingiz mumkin, bu dasturning ishlashiga hech qanday ta'sir ko'rsatmaydi, lekin u yozilgan kodga eslatma qo'yishga yordam beradi. Ikki

*/*Основные математические операции:*/*

int **i**=0; *//объявляем переменную i и присваиваем ей значение 0*

//Сложение:

i = 2+2; *//после выполнения данного выражения переменная i будет равна 4*

//Вычитание:

i = 2-2; *//после выполнения данного выражения переменная i будет равна 0*

//Умножение:

i = 2*2; *//после выполнения данного выражения переменная i будет равна 4*

//Деление:

i = 2/2; *//после выполнения данного выражения переменная i будет равна 1*

Ko'pincha dastur shartlarga qarab bir kod qismidan ikkinchisiga o'tishi kerak, buning

if(**i**>3) *//если i больше 3, то присвоить i значение 0*

{

i=0;

}

*/*если i меньше 3, то перейти к коду следующему после тела условия, т.е. после скобок {}*/*

Shuningdek, if else bilan birgalikda ishlatilishi mumkin - aks holda

if(**i**<3) *//если i меньше 3, то присвоить i значение 0*

{

i=0;

}

else

{

i=5; *//иначе, т.е. если i больше 3, присвоить значение 5*

Asosiy funktsiya ichida asosiy dastur allaqachon ishlamoqda. Funktsiya bajarilgandan so'ng, dastur to'xtaydi, shuning uchun ular doimo bir xil dasturni aylantiradigan cheksiz while tsiklini yaratadilar.

void **main**(**void**)

{

while (1)

{

};

};

chiziqli chiziqqa izoh berishingiz mumkin
// shundan keyin kompilyator butun qatorni yoki bir nechta satrlarni */**/* e'tiborsiz qoldiradi, masalan:

uchun shartli if () operatsiyalari mavjud, masalan:



}

Taqqoslash operatori "==" ham mavjud, uni
"=" tayinlash bilan aralashtirib yubormaslik

```
if(i==3)//если i равно 3, присвоить i значение 0
{
i=0;
}
if(i!=5) //если i не равно 5, присвоить i значение 0
{
i=0;
}
```

kerak. Teskari amal "!=" ga teng emas,
deylik

Ba'zi hollarda, murakkabroq narsalarg –
funktsiyalar asosida masalaga yechim
topish muhim hisoblanadi. Misol uchun,
masala yechuvchi shaxsdada bir necha
marta takrorlanadigan kod bo'lagi bor.
Bundan tashqari, bu kod juda katta hajmga
ega. Uni har safar yozish noqulay

```
void main(void)
{
if(PIND.0==0) //проверяем нажата ли кнопка на PD0
{
if(i==0) //если i==0 включить PB0
{
PORTB.0=1;
}
if(i==5) // если i==5 включить PB1
{
PORTB.1=1;
}
}
...
if(PIND.3==0)// выполняем тоже самое, при проверке кнопки PD3
{
if(i==0)
{
PORTB.0=1;
}
if(i==5)
{
PORTB.1=1;
}
}
```

hisoblanadi. Masalan, dasturda i
o'zgaruvchisi qandaydir tarzda
o'zgartiriladi, D portining 0 va 3 tugmalari
bosilganda xuddi shu kod bajariladi, u i
o'zgaruvchining qiymatiga qarab B
portining pinlarini yoqadi.



```
}
```

Umuman olganda, kod unchalik katta emas, lekin u bir necha marta kattaroq bo'lishi mumkin, shuning uchun o'z funktsiyangizni

```
void i_check()
```

```
{
    if(i==0)
    {
        PORTB.0=1;
    }
    if(i==5)
    {
        PORTB.1=1;
    }
}
```

```
void i_check()
```

```
{
    if(i==0)
    {
        PORTB.0=1;
    }
    if(i==5)
    {
        PORTB.1=1;
    }
}
```

```
void main(void)
```

```
{
    if(PIND.0==0) //проверяем нажата ли кнопка на PD0
    {
        i_check();
    }
    ...
    if(PIND.3==0)
    {
        i_check();
    }
}
```

Kod i_check () qatoriga yetganda; keyin u funksiya ichiga o'tadi va ichidagi kodni bajaradi. Qabul qiling, kod yanada ixcham va aniqroq, ya'ni. funktsiyalari bir xil kodni faqat bitta satr bilan almashtirishga yordam beradi. E'tibor bering, funksiya asosiy koddan tashqarida e'lon qilinadi, ya'ni.

yaratish ancha qulayroq bo'ladi. Misol uchun:

```
}
}
```

void funksiya hech narsani qaytarmasligini bildiradi, ko'proq quyida i_check () - bu bizning funktsiyamizning nomi, siz uni xohlaganingizcha chaqirishingiz mumkin, men uni shunday chaqirdim - i. Endi biz kodimizni qayta yozishimiz mumkin:

asosiy funksiyadan oldin . Siz aytishingiz mumkin, ha, nima uchun bu menga kerak, lekin darslarni o'rganayotganda siz tez-tez funktsiyalarga duch kelasiz, masalan, LCD displeyni tozalash lcd_ clear () - funksiya hech qanday parametrlarni olmaydi va hech narsani qaytarmaydi, lekin ekranni



tozalaydi. . Ba'zan bu funksiya deyarli har bir boshqa satrda ishlatiladi, shuning uchun kodni tejash aniq.

Funktsiyadan qiymat qabul qilganda foydalanish qiziqroq, masalan, c o'zgaruvchisi va int tipidagi ikkita qiymatni qabul qiluvchi sum funksiyasi mavjud . Asosiy dastur ushbu funktsiyani bajarganda, argumentlar allaqachon qavs ichida bo'ladi, shuning uchun "a" ikkiga teng bo'ladi va "b" 1 ga teng bo'ladi. Funktsiya bajariladi va "c" 3 ga teng bo'ladi. .

```
int c=0;
```

```
void sum(int a, int b)
{
    c=a+b;
```

```
}
void main(void)
{
    sum(2,1);
}
```

Tez-tez uchraydigan bunday funktsiyalardan biri lcd_gotoxy (0,0) LCD displeyidagi kursorni tarjima qilishdir; Aytgancha, bu argumentlarni ham oladi - x va y koordinatalari.

Funktsiyadan foydalanishning yana bir usuli, u qiymatni qaytarganda, endi u bekor bo'lmaydi , keling, ikkita raqamni qo'shish funktsiyasining oldingi misolini yaxshilaymiz:

```
int c=0;
```

```
int sum(int a, int b)
{
    return a+b;
}
void main(void)
{
    c=sum(2,1);
```

```
}
```

Endi massivlarga o'tamiz. Massiv bog'langan o'zgaruvchilardir. Misol uchun, sizda bir nechta nuqtali sinuslar jadvali mavjud, siz o'zgaruvchilarni yaratmaysiz int sinus1=0; int sinus2=1; va hokazo. Buning uchun massiv ishlatiladi. Masalan, siz uchta elementdan iborat massivni yaratishingiz mumkin:

```
int sinus [3]={0,1,5};
```

```
char hello[6]={'п','р','и','в','е','т'};
```

Biz harflar massivini yaratdik, endi ular ko'rsatilishi kerak, bitta belgini ko'rsatish funktsiyasi, aytaylik, nol, shunday ko'rinadi - lcd_putchar (0); bular. salomlashish bilan massivni ko'rsatish uchun barcha harflarni navbat bilan chiqarishingiz kerak bo'ladi.

```
lcd_putchar(hello[0]);
lcd_putchar(hello[1]);
lcd_putchar(hello[2]);
```

va boshqalar

Bu juda mashaqqatli bo'lib chiqdi, bu erda sikllar yordamga keladi. Masalan, while tsikli

```
int i=0;
while(i<6)
{
    lcd_putchar(hello[i]);
    i++;
}
```

Bu dastur qismi ancha ixchamroq ko'rinadi, buni anglatadi: i oltidan kichik bo'lsa, massivning har bir elementini navbat bilan chop eting. Bular. ichidagi kod qavs ichidagi shart to'g'ri bo'lmaguncha bajariladi. Masalan, while loop dan "nostandart" usuldan ham foydalanishingiz mumkin

```
while(PINB.0!=0)
{
}
```

Tugma bosilmasa-da, hech narsa qilmang - bo'sh tsiklni boshqaring.



For tsiklining yana bir versiyasi

```
int i;  
for(i=0;i<6;i++)  
{  
    lcd_putchar(hello[i]);  
}
```

while ga to'liq o'xshaydi, faqat i=0 boshlang'ich sharti qo'shiladi va shart har bir i++ siklida bajariladi. Loop ichidagi kod imkon qadar soddalashtirilgan.

Dasturni yozgandan so'ng, manba kodi tuziladi va agar xato bo'lmasa, siz loyiha papkasida kerakli proshivkani olasiz. Endi

siz mikrokontrollerni miltillatishingiz va qurilmaning ishlashidan bahramand bo'lishingiz mumkin.

Mikrodasturingizdagi tsikllar, massivlar va funktsiyalardan darhol foydalanishga urinmang. Foydalanuvchining asosiy vazifasi proshivkaning ishlashini ta'minlashdir, shuning uchun buni siz uchun osonroq qiling, kod hajmiga e'tibor bermang. Shunday vaqt keladiki, biz shunchaki ishchi kodni yozishni emas, balki uni chiroyli va ixcham yozishni xohlaysiz. Shunda C tilining murakkab va ilg'or jihatlarini o'rganish mumkin bo'ladi.

Foydalanilgan adabiyotlar:

1. R.McRoberts Published 2009 //by Earthshine Design. Design: Mike McRoberts.
2. Маесимо Банци Arduino для начинающих волшебников /М. Банци.—М.: Рид Групп, 2012. — 128 с.
3. Тухтасинов, М. Т., & Имамназаров, Э. Д. (2018). Алгоритмы распознавания лица на основе оператора локально направленные шаблоны (ЛНШ). Экономика и социум, (5), 1654-1661.
4. Жакбаров, О. О., Имамназаров, Э. Д., & Кодиров, З. З. (2015). Создание пакета прикладных программ для оптимального управления процессом фильтрации для разработки газовых месторождений. Молодой ученый, (9), 226-230.
5. Imamnazarov, E. (2020). The use of educational and practical games in the formation of the independent work in the personnel skills. ACADEMICIA: An International Multidisciplinary Research Journal, 10(4), 123-126.
6. Imamnazarov, E. D., & Orifjonov, U. (2019). Farewell to educational institutions and educational practices. Экономика и социум, (6), 36-38.
7. Ж.Юнусов, Х.Ю.Абасханова "Рақамли қурилмалар ва микропроцессор тизимлари". Ўқув қўлланма. Тошкент. 2010. 256 б
8. Имамназаров, Э., & Дехканов, А. (2021). Использование и роль информационных технологий в нашей жизни. Экономика и социум, (6-1), 633-635.
9. Справочник по промышленной робототехнике: 1-китоб. Н. Ноф таҳрири остида: Инглизчадан Д.Ф.Миронов ва б. Таржимаси –М.: Машиностроение, 1989.- 480б. 14
10. Аҳтамзода Деҳнавий "Ишлаб чиқаришни роботлаштириш: афзалликлар ва нуқсонлар" <https://ictnews.uz/uz/08/01/2019/> robotlashtirish