

# ARDUINO TARAQQIYOTI

Ma'ruza #8

## **Shift Registrlari**

Reja:

1. Shift (siljish) registrlarini tushunish
2. Parallel va ketma -ket ma'lumotlarni yuborish
3. 74HC595 shift registr bilan ishlash
  - a. Shift registrining pin funksiyalarini tushunish
  - b. Shift registrining qanday ishlashini tushunish
4. Arduinodan ketma-ket ma'lumotlarni o'tkazish
5. Ikkilik va o'nlik formatlarni konvertatsiya qilish
6. Yorug'lik animatsiyalarini siljish registri yordamida boshqarish
7. "Yengil chavandoz" loyihasini yaratish

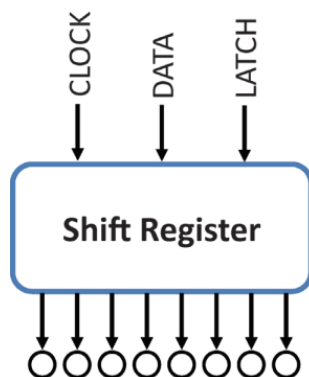
Xulosa

Foydalanilgan adabiyotlar

## Shift (siljish) registrlarini tushunish

Shift registri - ketma -ket bitlar oqimini qabul qiladigan va bir vaqtning o'zida parallel kirish -chiqish pinlariga bu bitlarning qiymatlarini chiqaradigan qurilma.

Ko'pincha, ular ko'p sonli LEDlarni boshqarish uchun ishlatiladi, masalan, yetti segmentli displeylarda yoki LED matritsalarida. Arduinoda shift registrni ishlatishdan oldin, ketma-ket va parallel siljish registriga kirish va chiqishlar ko'rsatilgan quyidagi rasmdagi diagramani ko'rib chiqing. Dars davomida ushbu sxemaning o'zgarishi har xil kirishlar chiqishga qanday ta'sir qilishini ko'rsatiladi.



Sakkizta doira shift registrining sakkizta chiqishiga ulangan LEDlarni ifodalaydi. Uchta kirish - bu registrni Arduino -ga ulaydigan ketma -ket aloqa liniyalari.

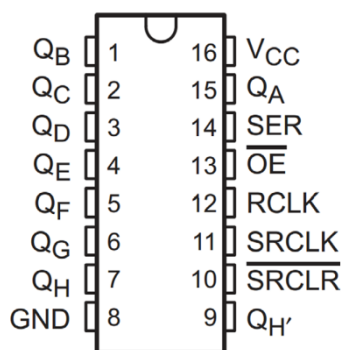
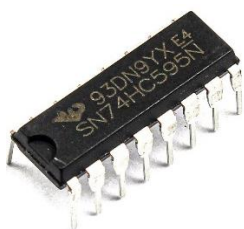
## Parallel va ketma -ket ma'lumotlarni yuborish

Ma'lumotlarning bir nechta bitlarini yuborishning ikki yo'li mavjud. Eslatib o'tamiz, Arduino, barcha mikrokontrollerlar kabi, raqamli; u faqat 1 va 0 sonlarni tushunadi. Shunday qilib, agar siz sakkizta LEDni raqamli boshqarish uchun yetarli ma'lumotga ega bo'lishni xohlasangiz (ularning har biri yoqilgan yoki o'chirilgan bo'lsa), siz jami 8 bit ma'lumotni uzatish usulini topishingiz kerak. Oldingi boblarda siz buni *digitalWrite ()* va *analogWrite ()* yordamida parallel ravishda qildingiz. Parallel ma'lumotlarni uzatish misoli uchun, sakkizta raqamli chiqish bilan sakkizta LEDni yoqish kerak edi deylik; Barcha bitlar bir vaqtning o'zida mustaqil kirish -chiqish pinlarida uzatiladi. "USB va ketma -ket aloqa" - u bir vaqtning o'zida 1 bitli ma'lumotlarni uzatadi. Shift registrleri ma'lumotlarni uzatishning ketma -ket va parallel usullarini osongina o'zgartirish imkonini beradi. Bu sinf ketma-ket parallel siljish registrlariga, ba'zan ketma-ket kirish, parallel chiqish (SIPO) siljish registrlariga qaratilgan. Ushbu qulay qurilmalar yordamida siz bir necha bayt ma'lumotni ketma -ket "CLOCK"ga

kiritishingiz va ularni parallel ravishda registrdan chiqarishingiz mumkin. Siz shuningdek shift registrlarni bir -biriga zanjirlashingiz va shu tariqa faqat uchta Arduino kirish - chiqish pinidan yuzlab raqamli chiqishni boshqarishingiz mumkin.

## 74HC595 shift registr bilan ishlash

Siz foydalanishingiz mumkin bo'lgan shift registr - bu 74HC595 shift registri. Quyidagi rasmlarda uning tashqi ko'rinishi, sxemadagi belgilanishi va pinlarini ko'rib chiqishingiz mumkin.



### Shift registrining pin funksiyalarini tushunish

Quyida shift registri pin funksiyalarining tasnifi keltirilgan :

- ■ Q<sub>H</sub> dan Q<sub>A</sub>gacha pinlari siljish registrdan sakkizta parallel chiqishni ko'rsatadi (Yuqoridagi rasmda ko'rsatilgan doiralarga ulangan).
- ■ VCC 5V ga ulanadi.
- ■ GND Arduino bilan umumiy minus qutbga ulanadi.
- ■ SER pin tepadagi rasmda DATA kiritilishi bilan ko'rsatilgan. Bu parallel chiqishlar qiymatlarini o'rnatish uchun 8 ta ketma - ket bit qiymatlari bilan ta'minlanadigan pin.
- ■ SRCLK pini tepadagi rasmdagi CLOCK pini bilan ifodalangan. Har safar bu pin ko'tarilganda, shift registrdagi qiymatlar 1 bitga siljiydi. Ma'lumotlar piniga yuborilgan barcha ma'lumotlarni yig'ish sakkiz marta uriladi.
- ■ RCLK pin tepadagi rasmda LATCH pin bilan ifodalangan. Registr soati pin sifatida ham tanilgan, LATCH pin yaqinda o'tkazilgan ketma-ket qiymatlarni bir vaqtning o'zida parallel chiqishlarga "majbur qilish" uchun ishlatiladi. Ushbu pin sizga ma'lumotlarni chipga ketma-ket o'tkazish imkonini beradi va barcha qiymatlar bir vaqtning o'zida parallel chiqishlarda ko'rsatiladi.

Siz bu misollarda SRCLR yoki OE pinlarini ishlatmaysiz, lekin siz ularni loyihangiz uchun ishlatishni xohlashingiz mumkin, shuning uchun ular nima qilayotganini tushunishga arziydi. OE chiqish imkoniyatini anglatadi. Pin nomi ustidagi chiziq uning faol pastligini ko'rsatadi. Boshqacha qilib aytganda, pin pastda bo'lsa, chiqish yoqiladi. Qachonki u baland ushlab turilsa, chiqish o'chiriladi. Ushbu misollarda, bu pin to'g'ridan -to'g'ri yerga ulanadi, shuning uchun parallel chiqish har doim yoqiladi. Siz bir vaqtning o'zida barcha LEDlarni yoqish yoki o'chirish uchun Arduinoning kirish - chiqish piniga ulashingiz mumkin. SRCLR pin - bu ketma -ket aniq pin. Pastga tortilganda, u shift registrining tarkibini bo'shatadi. Ushbu bobdagi maqsadlaringiz uchun, shift registri qiymatlari o'chirilmasligi uchun uni to'g'ridan -to'g'ri 5V ga bog'lab qo'yasiz.

### **Shift registrining qanday ishlashini tushunish**

Shift registri sinxron qurilma; u faqat soat (CLOCK) signalining ko'tarilgan chekkasida harakat qiladi. Har safar soat signali pastdan yuqoriga o'tganda, hozirda sakkizta chiqish registrlarida saqlangan barcha qiymatlar bir pozitsiyaga siljiydi. Bir vaqtning o'zida DATA kiritishidagi qiymat birinchi holatga o'tkaziladi. Buni sakkiz marta bajarish bilan joriy qiymatlar siljiydi va yangi qiymatlar registrga o'tkaziladi. LATCH pini bu tsikl oxirida yangi o'zgartirilgan qiymatlarni chiqishlarda paydo bo'lishi uchun baland qilib o'rnatiladi. Quyidagi rasmda ko'rsatilgan sxema ushbu dastur oqimini yanada aniqroq ko'rsatadi. Masalan, har bir boshqa LEDni ON holatiga (QA, QC, QE, QG) o'rnatmoqchisiz. Ikkilik shaklda ko'rsatilgan, siz o'tish registridagi parallel pinlarning chiqishi shunday ko'rinishini ta'minlashingiz kerak: 10101010.



## Arduinodan ketma-ket ma'lumotlarni o'tkazish

Endi siz sahna ortida nima sodir bo'layotganini tushunganingizdan so'ng, o'zgartirish registrini shu tarzda boshqarish uchun Arduino kodini yozishingiz mumkin. Oldingi barcha tajribalaringiz kabi, siz ma'lumotlarni IC registriga o'tkazish uchun Arduino IDE-ga o'rnatilgan qulay funksiyadan foydalanasiz. *ShiftOut* () funksiyasidan foydalanib, 8 bitli ma'lumotlarni ixtiyoriy kirish -chiqish piniga osongina ko'chirish mumkin. U to'rtta parametрни qabul qiladi:

- ■ Ma'lumotlar pin raqami
- ■ Soat (CLOCK) pin raqami
- ■ Bit tartibi
- ■ O'tish qiymati

Agar, oldingi bo'limda tasvirlangan o'zgaruvchan naqshni o'zgartirmoqchi bo'lsangiz, *shiftOut* () funksiyasidan quyidagicha foydalanish mumkin:

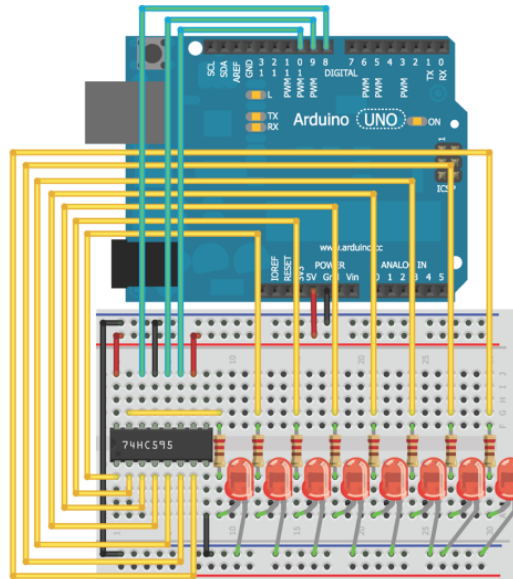
*shiftOut (DATA, CLOCK, MSBFIRST, B10101010);*

*DATA* va *CLOCK* o'zgarmas kattaliklar shu liniyalari uchun pin raqamlariga belgilanadi. *MSBFIRST* birinchi navbatda eng muhim bit yuborilishini bildiradi (yuborish uchun ikkilik raqamga qaraganda eng chap bit). Ma'lumotni ikkilik ma'lumotlarning o'ng tomonidan bitlarni uzatishdan boshlanadigan *LSBFIRST* sozlamalari bilan yuborishingiz mumkin. Yakuniy parametr - bu yuboriladigan raqam. Raqamdan oldin *B* harfini qo'yib, siz Arduino IDEga quyidagi raqamlarni kasrli son sifatida emas, balki ikkilik qiymat sifatida talqin qilishni aytasiz.

Keyinchalik, siz oldingi bo'limlarda o'rgangan tizimning fizik versiyasini yaratasiz. Birinchidan, siz shift registrini Arduinoga ulashingiz kerak::

- ■ DATA pin 8. PIN ulanadi
- ■ LATCH pin 9. PIN ulanadi
- ■ CLOCK pin 10. PIN ulanadi

LED laringiz bilan tok oqimini cheklovchi rezistorlardan foydalanishni unutmang. Zanjirni o'rnatish uchun quyidagi rasmda ko'rsatilgan diagramaga murojaat qiling.



Shift registrlari qanday ishlashini va shiftOut () funksiyasini tushunishingizdan foydalanib, quyidagi kodni birlashtirilgan LEDlarga muqobil LED naqshini yozish uchun ishlatishingiz mumkin.

```
const int SER =8;
const int LATCH =9;
const int CLK =10;
void setup()
{

pinMode(SER, OUTPUT);
pinMode(LATCH, OUTPUT);
pinMode(CLK, OUTPUT);
digitalWrite(LATCH, LOW);
shiftOut(SER, CLK, MSBFIRST, B10101010);
digitalWrite(LATCH, HIGH);
}
void loop()
{

}
```

Shift registri qiymatlarni ushlab turishi uchun ularni sozlashda faqat bir marta yuborish kerak; keyin siz ularni boshqa narsaga o'zgartirmaguningizcha bu qiymatlarda qoladilar. Ushbu dastur tepadagi rasmda grafik tarzda ko'rsatilgan bir xil qadamlarni bajaradi. LATCH pini past o'rnatilgan, 8 bit ma'lumotlar shiftOut () funktsiyasidan foydalangan holda siljiydi, so'ng LATCH pin yana baland qilib o'rnatiladi, shunda siljigan qiymatlar IC registrining parallel chiqish pinlarida chiqariladi.

### Ikkilik va o'nlik formatlarni konvertatsiya qilish

Yuqoridagi dasturda LED holati to'g'risidagi ma'lumotlar raqamlarning ikkilik qatori sifatida yozilgan. Ushbu satr qaysi LEDlar yoqilishi va o'chirilishini tasavvur qilishga yordam beradi. Shu bilan birga, base2 (ikkilik) va base10 (o'nlik) tizimlar o'rtasida konvertatsiya qilish orqali siz naqshni o'nlik qiymat sifatida yozishingiz mumkin. Ikkilik sonning har bir biti (eng o'ng yoki eng ahamiyatli bitdan boshlab) ortib borayotgan darajani ifodalaydi. Avvalo, pastdagi rasmda ko'rsatilgan ikkilik sonni tegishli o'nlik kasrga aylantirish bosqichlarini ko'rib chiqing.

$$\begin{array}{cccccccc}
 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
 2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\
 1 \times 128 + 0 \times 64 + 1 \times 32 + 0 \times 16 + 1 \times 8 + 0 \times 4 + 1 \times 2 + 0 \times 1 = 170
 \end{array}$$

Har bir bitning ikkilik qiymati 2-sonli ortish darajasini ifodalaydi. Bu sonda 7, 5, 3 va 1 bitlar yuqori. Shunday qilib, kasrli ekvivalentni topish uchun siz  $2^7$ ,  $2^5$ ,  $2^3$  va  $2^1$  ni qo'shasiz. Olingan o'nlik qiymat 170. Bu qiymatni avval sanab o'tilgan kodga almashtirish orqali o'zingizga ekvivalent ekanligini isbotlashingiz mumkin. ShiftOut () qatorini ushbu versiya bilan almashtiring:

```
shiftOut (SER, CLK, MSBFIRST, 170);
```

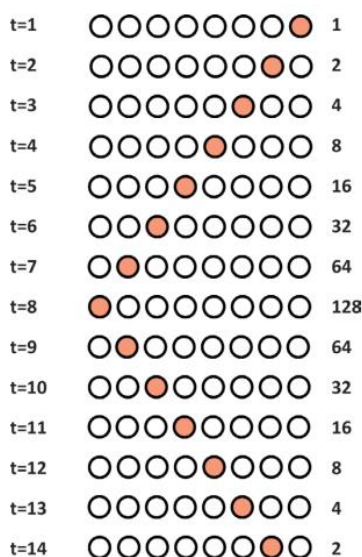
Siz ikkilik belgini ishlatganingizdagi natijani ko'rishingiz kerak.

## Yorug'lik animatsiyalarini siljish registri yordamida boshqarish

Oldingi misolda siz shift registri bilan statik displey qurgansiz. Biroq, ehtimol siz LED -larda yanada dinamik ma'lumotlarni ko'rsatishni xohlaysiz. Keyingi ikkita misolda siz yorug'lik effektini va fizik chiziqli grafikni boshqarish uchun shift registrdan foydalanasiz.

### "Yengil chavandoz" loyihasini yaratish

Yengil chavandoz - bu aniq effekt, shuning uchun LEDlar bir -birlarini oldinga va orqaga quvayotganga o'xshaydi. Ilgari ishlatgan sxemadan foydalanishda davom eting. *ShiftOut* () funksiyasi juda tez ishlaydi va siz uning yordamida shift registrini soniyasiga bir necha ming marta yangilashingiz mumkin. Shu sababli, siz dinamik yoritish animatsiyasini yaratish uchun shift registr chiqishlarini tezda yangilashingiz mumkin. Bu yerda siz har bir LEDni navbat bilan chapdan o'ngga, keyin o'ngdan chapga yondirasiz. Siz birinchi navbatda har bir animatsiya holatini aniqlab olmoqchisiz, shunda siz ularni osongina aylanib o'tishingiz mumkin. Har bir vaqt bosqichida hozirda yoritilgan LED o'chadi va keyingi chiroq yonadi. Chiroqlar oxiriga yetganda, xuddi shu narsa teskarisida sodir bo'ladi. Quyidagi rasmdagi vaqt diagrammasi har bir qadam uchun chiroqlar qanday ko'rinishini va shu LEDni yoqish uchun zarur bo'lgan o'nlik qiymatini ko'rsatadi.



Ilgari o'rganganlaringizni eslab, har bir yorug'lik bosqichining ikkilik qiymatlarini o'nlik kasrlarga aylantiring, ularni osongina aylanib o'tish mumkin. *For loop*idan foydalanib, siz ushbu qiymatlarning har bir qatorini aylanib o'tishingiz va ularni birma - bir siljish registriga o'tkazishingiz mumkin. Quyidagi kod aynan shunday qiladi.

```
const int SER =8;
const int LATCH =9;
const int CLK =10;

int seq[14] = {1,2,4,8,16,32,64,128,64,32,16,8,4,2};
void setup()
{

pinMode(SER, OUTPUT);
pinMode(LATCH, OUTPUT);
pinMode(CLK, OUTPUT);
}
void loop()
{
for (int i = 0; i < 14; i++)
{
digitalWrite(LATCH, LOW);
shiftOut(SER, CLK, MSBFIRST, seq[i]);
digitalWrite(LATCH, HIGH);
delay(100);
}
}
```

Kechikish funksiyasi ichidagi qiymatni sozlash orqali siz animatsiya tezligini o'zgartirishingiz mumkin. Har xil naqshlar ketma - ketligini yaratish uchun *seq* massivining qiymatlarini o'zgartirishga harakat qiling.

## **Xulosa**

Ushbu darsda quyidagilar haqida bilib oldingiz:

- ■ Shift registrlarining qanday ishlashi
- ■ Ketma - ket va parallel ma'lumotlar uzatish turlari orasidagi farqlar
- ■ O'nlik va ikkilik ma'lumotlarni o'rtasidagi farqlar
- ■ Shift registrlari yordamida animatsiyalar qilish

## **Foydalanilgan adabiyotlar**

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