თემა 3. ელექტრონიკის ძირითადი ციფრული ხელსაწყოები

მეცადინეობა 2

არითმეტიკულ-ლოგიკური კვანძი და მიკროკონტროლერები

წინა მეცადინეობებზე განვიხილეთ პრაქტიკულად ყველა ელექტრული კვანძი, რომელიც თანამედროვე ციფრულ ელექტრონიკაში მუშაობს. ყველა ეს ხელსაწყო მარტივ ლოგიკურ ელემენტებზეა აწყობილი. არის ციფრულ ელექტრონიკაში რამდენიმე კვანძი, რომელიც არითმეტიკულ ოპერაციებს ასრულებს. ყველა არითმეტიკული ოპერაციის საფუძველში დევს შეკრების ოპერაცია. რადგან ციფრულ ელექტრონიკაში ორობით რიცხვებს ვიყენებთ, უნდა განვსაზღვროთ თუ რას წარმოადგენს ორობითი შეკრების ოპერაცია. განვიხილოთ უმარტივესი შემთხვევა – ორი ერთბიტიანი რიცხვის შეკრება:

- თუ რიცხვებია 0 და 0 – შეკრების შედეგი 0;
- თუ რიცხვებია 0 და 1 – შეკრების შედეგი 1;
- თუ რიცხვებია 1 და 0 – შეკრების შედეგი 1;
- თუ რიცხვებია 1 და 1 – შეკრების შედეგი 10, მიმდინარე თანრიგი შეივსო და შემდეგ, უფრო მაღალ თანრიგში, გაჩნდა 1.

უკანასკნელი შედეგი მიუთითებს, რომ იმ თანრიგში, რომელსაც რიცხვები მიეკუთნებოდა არის 0, და აუცილებელია მაღალი თანრიგის შემოყვანა. აჯამვის შედეგი შემდეგ თანრიგში უნდა აისახოს.

განვიხილოთ ორი რიცხვი 1100 და 1010, და მათი შეკრების ოპერაცია ცხრილის სახით.

A	**1**	**1**	**0**	**0**
B	**1**	**0**	**1**	**0**
S	**0**	**1**	**1**	**0**
C	**1**	**0**	**0**	**0**

ცხრილის პირველ ორ სტრიქონში წარმოდგენილია რიცხვები, მესამეში მათი ჯამი, ხოლო მეოთხეში იმის "ნიშანი", რომ შემოსაღებია კიდევ ერთი თანრიგი, და მასში 1 უნდა ჩაიწეროს. ეს ბოლო სტრიქონი ე.წ. "გადატანის" აუცილებლობას აღნიშნავს. ამ სტრიქონში ჩვენ 1-ის სახით ვიმახსოვრებთ, რომ თანრიგი შეივსო, და შემდეგ, უფრო მაღალ თანრიგს, მის ბიტებთან ერთად, უნდა გადაეცეს შესაკრებათ კიდევ ერთი ერთიანი. ცხრილი მიიღებს სახეს:

A	**0**	**1**	**1**	**0**	**0**
B	**0**	**1**	**0**	**1**	**0**
S	**1**	**0**	**1**	**1**	**0**
C	**0**	**1**	**0**	**0**	**0**

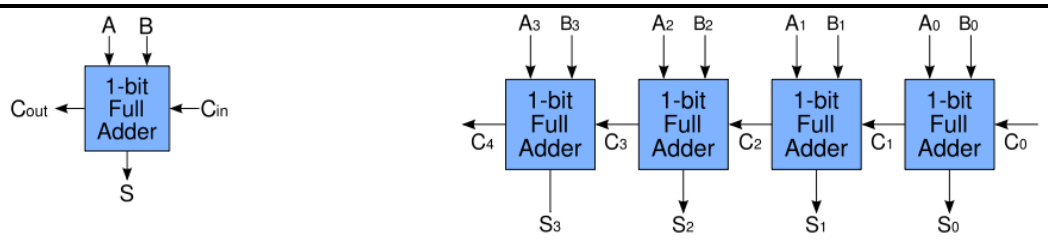
, ანუ ზოგადად -

A		A_n	.	A_i	.	A_3	A_2	A_1	A_0
B		B_n	.	B_i	.	B_3	B_2	B_1	B_0
S		S_n	.	S_i	.	S_3	S_2	S_1	S_0
C		C_n	.	C_i	.	C_3	C_2	C_1	C_0

A_i და B_i შესაკრები რიცხვების i–ური თანრიგის ციფრებია, S_i – მათი ჯამია, ხოლო C_i i+1 თანრიგში გადატანის მაჩვენებელია.

ნახ. 3.11-ზე სქემატურად წარმოდგენილია 1 ბიტში მიმდინარე შეკრების ოპერაციის მონაწილე წევრები და მრავალი ბიტის მქონე რიცხვების აჯამვის თანმიმდევრობა, რომელიც ხორციელდება ზოგადი შემთხვევის ცხრილის შესაბამისად.

ნახ. 3.11



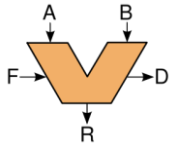
ა) 1 ბიტში მიმდინარე შეკრების ოპერაციის წევრები და სათანადო ელექტრული ყჯრედი, რომელის ასრულებს მოქმედებას ტექსტში ხსენებული წესის შესაბამისად.

ბ) მრავალი ბიტის მქონე რიცხვების აჯამვის თანმიმდევრობა

აჯამვის ოპერაციის ორგანიზაცია ხდება ლოგიკური ელემენტების საშუალებით.

აჯამვა და სხვა არითმეტიკული ოპერაციების განხორციელება ციფრულ ელექტრონულ ხელსაწყოებში ხდება ე.წ. **არითმეტიკულ-ლოგიკური** კვანძის (**ალკ**) საშუალებით. სქემატურად ეს კვანძი ნაჩვენებია ნახ. 3.12-ზე. ამ გრაფიკული გამოსახულებიდან ნათელია, რომ განმარტებით, ეს კვანძი აწარმოებს ოპერაციას ორ შესავალ რიცხვზე (A და B) სათანადო ბრძანების შესაბამისად (F). ოპერაციის შედეგად მიიღება შედეგი (R) და განისაზღვრება მისი სტატუსი (D) მასთან რაიმე სხვა ოპერაციის შესრულებისათვის ამ კვანძის გარეთ. არითმეტიკულ-ლოგიკური კვანძის ცნება – ზოგადი ცნებაა. კვანძი შეიძლება სხვადასხვა სირთულის იყოს, შესაძლებელია ერთნაერი კვანძებისაგან შესდგებოდეს და პარალელურად ასრულებდეს სხვადასხვა ფუნქციებს. ზოგადად, კვანძის მუშაობის ციკლი მოსული ბრძანების "გააზრებით" იწყება. შემდეგ მოყვება ბრძანების შესრულება. დასრულებისას, ახალი ბრძანების მიღება-გააზრება და ა.შ.

ნახ. 3.12



არითმეტიკულ-ლოგიკური კვანძი

არითმეტიკულ-ლოგიკური კვანძი – ალკ, ყველა თანამედროვე სათვლელი მოწყობილობის განუყოფელი ნაწილია. ასეთი კვანძის კონცეფცია 1945 წელს შეიმუშავა მათემატიკოსმა ჯონ ფონ ნეიმანმა.

ალკ-ს ცნებასთან ერთად ჩვენ ავტომატურად შემოვიღეთ **ბრძანების ცნება** და ბრძანებების ერთობლიობის, ანუ **პროგრამის ცნება.**

ამ მომენტიდან გაგვაჩნია სრული ზოგადი ცოდნა იმისათვის, რომ ვიმსჯელოთ ე.წ. მიკროპროცესორის, ან მიკროკონტროლერის, ანდა უფრო რთული მოწყობილობის – პროცესორის აგებულებაზე. **ალკ**-ს დანიშნულების ლოგიკა კარნახობს შემდეგი მსჯელობის თანმიმდევრობას:

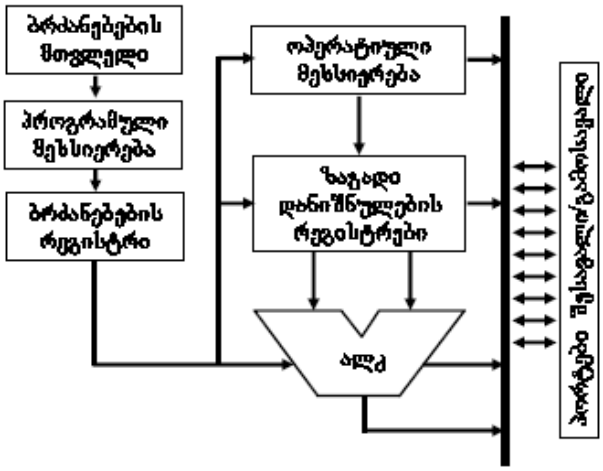
- მუშაობისას ალკ იღებს ინფორმაციულ ნაკადს ორი ტიპის რიცხვების, ორი ნაკადის სახით: ბრძანებების კოდები და დასამუშავებელი რიცხვები. ბრძანებები, თავის მხრივ, კოდირებულია რიცხვების სახით და ალკ-ს "ესმის", თუ რა ოპერაცია უნდა განახორციელოს ამა თუ იმ ბრძანების მიღებისას. ზოგადად, ბრძანება უნდა შეიცავდეს ინფორმაციას იმის შესახებ, თუ საიდან უნდა აიღოს ალკ-მ რიცხვები, რა ოპერაცია უნდა შეასრულოს და სად განათავსოს შედეგი. ალკ საიდანღაც უნდა იღებდეს

ბრძანებებს და რიცხვებს. ნათელია, ამისათვის საჭიროა მეხსიერება – მეხსიერება რომელიც მოემსახურება ბრძანებებს, ანუ პროგრამას, და მეხსიერება დასამუშავებელი რიცხვებისათვის.

- დაცული უნდა იყოს პროგრამის ბრძანებების თანმიმდევრობა, ანუ რომელიდაც კვანძი უნდა აკონტროლებდეს და იცავდეს ამ თანმიმდევრობას.
- ოპერაციების შესრულების შედეგი უნდა გადაიგზავნებოდეს სათანადო დანიშნულებით.

ამ ზოგადი მოსაზრებებიდან გამომდინარე, ისახება უმარტივესი **მიკროკონტროლერის**, ან **მიკროპროცესორის** სტრუქტურა. ეს მიკროსქემაა, რომლის შეიცავს ყველაფერს, რაც საჭიროა დამოუკიდებელი მუშაობისათვის რიცხვით მონაცემებთან. ოდნავ უფრო რთული მიკროკონტროლერები შეიცავენ კვანძებს ანალოგური ინფორმაციის ციფრულში და ციფრულის ანალოგურში გარდასაქმნელად, ე.წ. ტაიმერებს და სხვა ფუნქციონალურ კვანძებს. მიკროკონტროლერის გამარტივებული ბლოკ სქემა წარმოდგენილია ნახ. 3.13-ზე.

ნახ. 3.13



მიკროკონტროლერის გამარტივებული ბლოკ სქემა. მიკროკონტროლერი გამოიყენება სხვადასხვა მოწყობილობების მართვისათვის და მათი ურთიერთქმედების და ერთობლივი ფუნქციონირების ორგანიზაციისათვის.

მიკროკონტროლერის პირველი ნიმუში დაპატენტებული იყო 1971 წელს – მიკროსქემის ერთ კრისტალზე განთავსდა ამ ბლოკ სქემაზე მოცემული კვანძები.

სხვადასხვა კორპუსებში აწყობილი სხვადასხვა დანიშნულების მიკროკონტროლერები

ოპერაციის შესასრულებლად, ალკ იღებს რიცხვებს რეგისტრებიდან, რეგისტრებშიც ანთავსებს შედეგს. **ზოგადი დანიშნულების რეგისტრების კვანძი** შეიცავს სათანადო რაოდენობის რეგისტრებს ამ მიზნების უზრუნველსაყოფად. მუშაობის დროს რეგისტრების შერჩევა და მათთან ურთიერთქმედების თანმიმდევრობა განისაზღვრება პროგრამით. პროგრამამ ყოველთვის უნდა "დაასახელოს" - მიუთითოს ის რეგისტრები, რომლებთან უნდა განხორციელდეს მუშაობა ყოველ ეტაპზე. უმეტეს შემთხვევაში მიკროკონტროლერები იყენებენ 8-ბიტიან რეგისტრებს და მუშაობენ 8-ბიტიან ორობით რიცხვებთან. 8-ბიტიანი ორობითი რიცხვი 1 ბაიტის სახელს ატარებს.

რეგისტრების რაოდენობა შემოსაზღვრულია – ისინი უზრუნველყოფენ ალკ-ს ფუნქციონირების უშუალო მოთხოვნილებებს. მიკროკონტროლერის

სრულყოფილი მუშაობისათვის ეს არ არის საკმარისი, ამიტომ კონტროლერი შეიცავს **ოპერატიულ მეხსიერებას.** აქ ინახება ინფორმაცია, რომელიც იქმნება კონტროლერის მუშაობის პროცესში, ან შემოდის გარედან, და უზრუნველყოფს მისი მუშაობის სრულ ალგორითმს და ფუნქციის შესრულებას.

ბუნებრივია შეკითხვა - სად და როგორ ინახება კონტროლერის პროგრამა, და როგორ არის ორგანიზებული პროგრამის ურთიერთქმედება ალკ-სთან. ამისათვის გამოიყენება ბრძანებების რეგისტრი, პროგრამული მეხსიერება და ბრძანებების მთვლელი. ბრძანებების რეგისტრს თანმიმდევრულად მიეწოდებიან პროგრამის ბრძანებები. ზოგადათ, ეს რეგისტრი ემსახურება არა მარტო ალკ-ს – შესაძლებელია კონტროლერში სხვა კვანძებიც იყოს. ალკ მუდმივად "იხედება" ამ რეგისტრში, "არკვევს" არის თუ არა მისკენ მიმართული ბრძანება. თუ ასეთი ბრძანება გამოჩნდა, ალკ მას ასრულებს.

ბრძანებების რეგისტრში ბრძაბებები ხვდებიან პროგრამული მეხსიერებიდან. მიკროკონტროლერებში ეს კვანძი **მუდმივი მეხსიერების კვანძის** სახელითაა ცნობილი. ამ მეხსიერების ყოველი მისამართით აღნიშნულ უჯრედში ერთი ბრძანებაა ჩაწერილი. ერთად ეს ბრძანებები ქმნიან პროგრამას და ჩაწერილები არიან იმ თანმიმდევრობით, რომელიც გათვალისწინებულია მიკროკონტროლერის მუშაობის ალგორითმში. მეხსიერებიდან ბრძანების "ამოსაღებად" საჭიროა მიუთითოთ მას იმ უჯრედის მისამართი, სადაც ჩაწერილია სათანადო ბრძანება. კონტროლერის პროგრამირების დროს, ბრძანებები იწერება უჯრედებში იმ თანმიმდევრობით, რა თანმიმდევრობით უნდა სრულდებოდეს პროგრამა. პროგრამის ასამოქმედებლად საკმარისია ვცვალოთ ბრძანებების მთვლელის მნიშვნელობა. თუ მთვლელის მნიშვნელობა 1-ით იცვლება, პროგრამა შემდეგი ბრძანების შესრულებაზე გადადის.

როგორც ზემოდ აღინიშნა, მიკროკონტროლერები გამოიყენება სხვადასხვა მოწყობილობების სამართავად, ამიტომ კონტროლერს აუცილებლად უნდა გააჩნდეს ინფორმაციის გარეთ გამოყვანის და მიღების საშუალება. ეს ფუნქცია უზრუნველყოფილია ე.წ. **პორტებით**. პორტი წარმოადგენს ერთბიტიანი არხების ერთობლიობას, რომელთა ცალცალკე აწყობა შესაძლებელია როგორც ინფორმაციის მიღებაზე, ასევე გადაცემაზე. პორტის არხების აწყობა – მათი ფუნქციონალური დანიშნულების დაპროგრამება - ხორციელდება კონტროლერის ამოცანიდან გამომდინარე. კონტროლერის დაპროგრამება, როგორც წესი, იწყება პორტების აწყობით.

კონტროლერების დაპროგრამების პროცესი უზრუნველყოფილია პროგრამირების ენით, რომელიც ცნობილია **Assembler**–ის სახელით. ამ ენაზე იქმნება საწყისი პროგრამა. ეს უმარტივესი, უმარტივეს ფუნქციონალურ ელემენტებზე დაყვანილი პროგრამაა, რომელიც მუშაობს ე.წ. **მანქანურ კოდებთან.** კონტროლერისათვის უშუალოდ გასაგებია მხოლოდ ეს ენა, რომელიც უშუალოდ ორგანიზებულია კონტროლერის ელექტრონიკის კონსტრუქციაში. **Assembler**–ზე მომზადებული პროგრამა გარდაიქმნება მანქანურ კოდებში ე.წ. კომპილატორის საშუალებით. ეს პროგრამაა, რომელიც ამარტივებს პროგრამირების პროცესს – თავისებური "თარჯიმანია", რომელსაც გადაყავს **Assembler**–ის სიტყვებით და სიმბოლოებით აღნიშნული ბრძანებები მანქანური კოდების თანმიმდევრობაში.

დღესდღეობით, მიკროკონტროლერების დაპროგრამება ხორციელდება კომპიუტერის საშუალებით. მზადდება პროგრამა **Assembler**–ზე, ხდება მისი კომპილირება მანქანურ კოდებში და დამატებითი მოწყობილობის, **პროგრამატორის** საშუალებით, მისი ჩაწერა კონტროლერის პროგრამულ მეხსიერებაში. **პროგრამატორი**, ერთისმხვრივ, უერთდება კომპიუტერის პორტებს, მეორესმხვრივ კონტროლერს. პროგრამატორი უზრუნველყოფს ჩაწერის პროცედურის ტექნოლოგიურ ელექტრულ რეჟიმს.

პირველი მიკროკონტროლერების მახასიათებლები

პირველი მიკროკონტროლერი გაჩნდა 1971 წელს. კონტროლერი TMS1000 დაამზადა ფირმა Texas Instruments. მისი ოპერატიული მეხსიერება შეიცავდა 32 ბაიტს (ბაიტი 8-ბიტიანი ორობითი რიცხვია), პროგრამული მეხსიერება 1 კილობაიტს.

1976 წელს Intel –მა გამოუშვა კონტროლერი ცნობილი სახელით 8048. კონტროლერი შეიცავდა 64 ბაიტიან ოპერატიულ მეხსიერებას, 1 კილობაიტიან პროგრამულ მეხსიერებას, ორ ტაიმერს – მოწყობილობას რომელიც ზომავს განსაზღვრული დროის ინტერვალს, საათის გენერატორს და 27 ბიტიან პორტს (ან 27 ერთბიტიან პორტს, რაც ფუნქციონალურად ერთი და იგივეა). საათით აღჭურვილი კონტროლერი "მიბმულია" რეალურ დროსთან. მას შეუძლია დანიშნულ დროს დაიწყოს ან შეწყვიტოს რაიმე ფუნქცია, ტაიმერის საშუალებით კი გადაზომოს პროგრამით გათვალისწინებული ინტერვალი და, ისევ დაიწყოს ან შეაჩეროს რაიმე ქმედება. მაგალითად, დანიშნულ დროს ჩართოს რაიმე აგრეგატი და ამუშაოს იგი ტექნოლოგიური პროცესით განსაზღვრული დროის განმავლობაში.

თანამედროვე მიკროკონტროლერების მახასიათებლები

1996 წელს ფირმა Atmel-მა დაიწყო კონტროლერების საკმაოდ განვითარებული ოჯახის წარმოება. ამ ფირმის კონტროლერები დღეს საკმაოდ პოპულარულია, ერთგვარ სტანდარტსაც კი წარმოადგენენ კონტროლერებს შორის. რამდენიმე კონტროლერის მახასიათებელი, მაგალითისათვის, მოყვანილია ცხრილში.

Flash ROM – პროგრამულუ მეხსიერება (კილობაიტი);
RAM – მონაცემთა მეხსიერება (ბაიტი);
Max I/O – პორტების რაოდენობა;
F. max – მუშაობის მაქსიმალური სიხშირე;
Timer(s) – ტაიმერების/მთვლელების რაოდენობა;
RTC – რეალური დროის სისტემა (საათი);
AD – ანალოგურ/ციფრული გარდამქმნელების არხების რაოდენობა.

Device	*Flash ROM*	*RAM*	*Max I/O*	*F.max MHz*	*16-bit timer*	*8-bit timer*	*RTC*	*AD*
ATmega128	128	4096	53	16	2	2	Yes	8
ATmega1280	128	8192	86	16	4	2	Yes	16
ATmega1281	128	8192	51	16	4	2	Yes	16
ATmega16	16	1024	32	16	1	2	Yes	8
ATmega162	16	1024	35	16	2	2	Yes	-
ATmega164	16	1024	32	20	1	2	Yes	8
ATmega165	16	1024	54	16	1	2	Yes	8

ATmega168	16	1024	23	20	1	2	Yes	8
ATmega169	16	1024	53	16	1	2	Yes	8
ATmega2560	256	8192	86	16	4	2	Yes	16
ATmega2561	256	8192	51	16	4	2	Yes	16
ATmega32	32	2048	32	16	1	2	Yes	8
ATmega324	32	2048	32	20	1	2	Yes	8
ATmega325	32	2048	53	16	1	2	Yes	8
ATmega3250	32	2048	68	16	1	2	Yes	8
ATmega329	32	2048	53	16	1	2	Yes	8
ATmega3290	32	2048	68	16	1	2	Yes	8
ATmega406	40	2048	18	1	1	1	Yes	

ყველა კონტროლერი აღჭურვილია ერთი ან რამდენიმე სხვადასხვა ტიპის საკომუნიკაციო ხაზით, რომელიც უზრუნველყოფს მის ურთიერთქმედებას გარე მოწყობილობებთან.

მიკროკონტროლერის გამოყენების მაგალითები

მიკროკონტროლერების ფართო დანერგვამ შეცვალა თანამედროვე ტექნიკის სახე. მრავალი, მათ შორის ფართო მოხმარების, საგანი გახდა ავტომატიზირებული და აღჭურვილი ფუნქციონირების სხვადასხვა რეჟიმით. ამ ელექტრონული ხელსაწყოების ასეთმა ფართო გავრცელებამ შეცვალა მრავალი საინჟინრო მიმართულების სახე. კონტროლერების გასაცნობად, შესასწავლად და გამოსაყენებლად მრავალი სახელმძღვანელო გამოიცა, უამრავი ვებ საიტი იძლევა ინფორმაციას და რეკომენდაციებს. მიჩნეულია, რომ კონტროლერების გამოყენების ტექნოლოგიის ასათვისებლად ერთერთი საუკეთესოა ATmega8 კონტროლერი. მისი ტექნიკური მაჩვენებლები მოყვანილია ცხრილში.

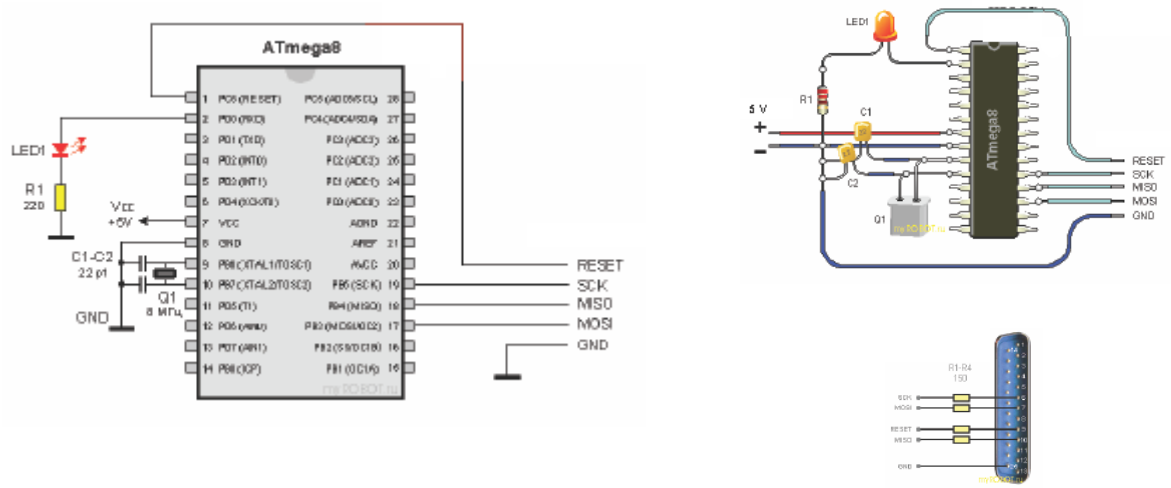
Device	*Flash ROM*	*RAM*	*Max I/O*	*F.max MHz*	*16-bit timer*	*8-bit timer*	*RTC*	*AD*
ATmega8	8	1024	23	16	1	2	Yes	8

ნახ 3.14 –ზე წარმოდგენილია ამ კონტროლერზე აწყობილი უმარტივესი ხელსაწყო – შუქგამომსხივებელი დიოდის პროგრამულად მართვის ბლოკი. ამ ხელსაწყოთი შესაძლებელია ნათების ნებისმიერი რეჟიმის ორგანიზაცია: სხვადასხვა სიხშირით ციმციმი, ცვალებადი სიხშირით ციმციმი, განსაზღვრულ დროს ჩართვა და გამორთვა, განსაზღვრული დროის ინტერვალებით ჩართვა, და სხვა, პრაქტიკულად ნებისმიერი რეჟიმის ორგანიზაცია. შესაძლებელია რამდენიმე შუქდიოდის ჩართვა და ე.წ. შუქგირლიანდების ორგანიზაცია.

ნახ 3.14 –ზე წარმოდგენილი სქემის ელემენტებია:
შუქდიოდი LED 1,
დამცავი რეზისტორო R1,
კონდენსატორები C1, C2,
კვარცის რეზონატორი Q1.

კონდენსატორები და კვარცის რეზონატორი წარმოადგენენ მიკროკონტროლერის სტაბილური სიხშირის გენერატორის გარე კვანძს. ეს სიხშირე გამოიყენება კონტროლერის საათისა და ტაიმერების მუშაობაში. რეზონატორი წარმოადგენს ელექტრულ მოწყობილობას – კონდენსატორს, რომლის ელექტროდებს შორის მოთავსებულია კვარცის კრისტალი. გენერატორში, საკუთარ სიხშირეხე აღზნებული კრისტალი, საკმაოდ ზუსტად ინარჩუნებს ამ სიხშირეს. ეს თავისებური ელექტრული ქანქარაა ელექტრონული საათისათვის.

ნახ 3.14

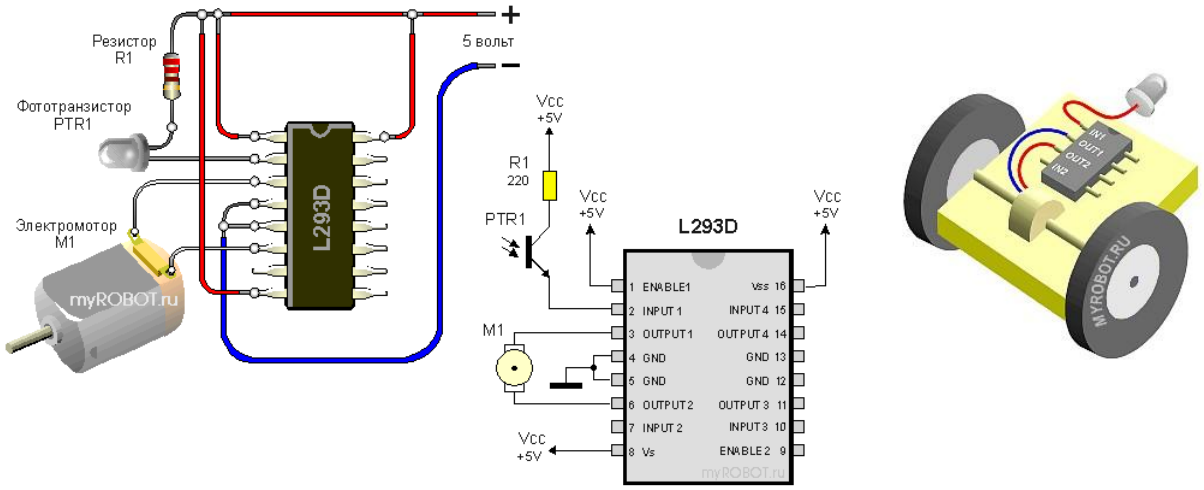


ა) ხელსაწყოს პრინციპული სქემა. მარჯვნივ ნაჩვენებია პორტები, რომელთა გამოყენებით ხერხდება კომპიუტერთან შეერთება და დაპროგრამება.

ბ) ხელსაწყოს სამონტაჟო სქემა და პროგრამირებისათვის კომპიუტერთან მიერთების ორგანიზაცია – მოითხოვს მხოლოდ დამცავ რეზისტორებს კომპიუტერის პორტთან შესაერთებლად

ნახაზიდან ნათელია, თუ რამდენად მარტივია ხელსაწყოს ტექნიკური ნაწილი, შეერთებების და დამატებითი დეტალების რაოდენობა საკმაოდ მცირეა. პრაქტიკულად მთელი ელექტრონიკა მოთავსებულია მიკროჩიპის შიგნით. მთელი ტექნოლოგიური აქცენტი გადატანილია კონტროლერის პროგრამირებაზე. კონტროლერი შეგვიძლია განვიხილოთ როგორც შავი ყუთი და მივყვეთ მისი გამოყენების ინსტრუქციას.

ნახ 3.15



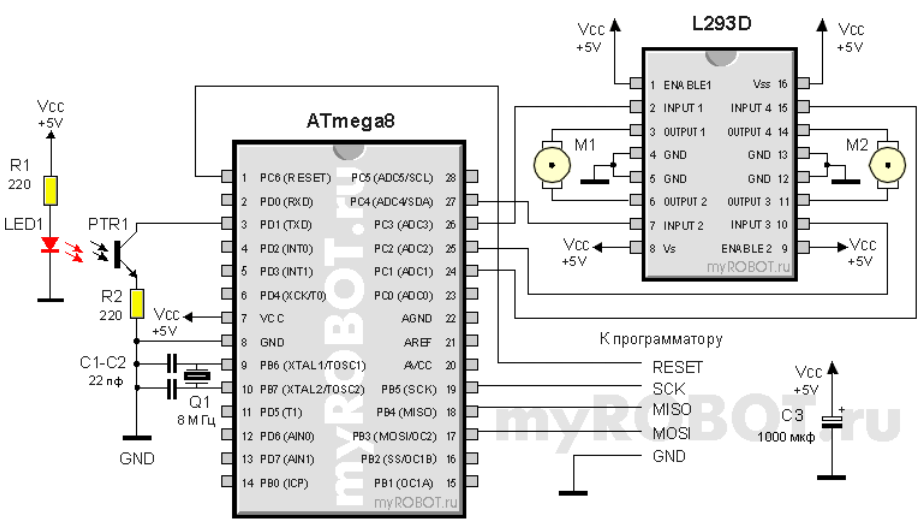
ა) ხელსაწყოს სამონტაჟო და პრინციპული სქემა. გამოყენებულია მიკროძრავის მიკროკონტროლერი. კომპონენტები: რეზისტორი, შუქდიოდი და ელექტრიძრავა.

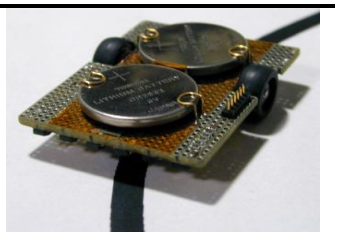
ბ) მინი რობოტი

კონტროლერების გამოყენების თვალსაზრისით ასეთი გამარტივება მათ მიმართ დიდ ინტერესს იწვევს და ტექნიკური მოდელირების და შემოქმედების ერთერთი მნიშვნელოვანი წყარო ხდება. განსაკუთრებით პოპულარული გახდა მარტივი მოძრავი ავტონომიური რობოტების აწყობა და სხვადასხვა რეჟიმით პროგრამირება. ქვემოდ მოყვანილია რამდენიმე მაგალითი საიტიდან myrobot.ru.

ეს კონსტრუქციები ადვილად ასაწყობია და მიკროკონტროლერების ტექნოლოგიის ასათვისებლად მეტად სასარგებლოა. პირველი რობოტი რეაგირებს სინათლეზე. პროგრამის მიხედვით შესაძლებელია დაიწყოს ან შეაჩეროს მოძრაობა გარე შუქსიგნალის მოქმედების შედეგად, ან უფრო რთული მოძრაობის პროგრამა შეასრულოს: სიგნალის შემდეგ მოძრაობა დაიწყოს ან შეაჩეროს პროგრამულად განსაზღვრული დროის ინტერვალის შემდეგ, დათვალოს სინათლის სასიგნალო იმპულსები და მოახდინოს მათზე რეაგირება პროგრამით გათვალისწინებული წესით და ა.შ. ასეთი კონსტრუქცია წარმოდგენილია ნახ 3.15 –ზე.

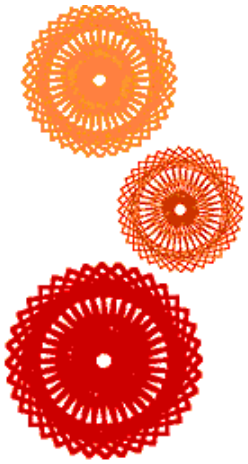
ნახ 3.16





ა) ხელსაწყოს სამონტაჟო და პრინციპული სქემა. კომპონენტები: შუქგამომსხივებელი და შუქმიმღები დიოდები, რეზისტორები, კონდენსატორები, კვარცის რეზონატორი, მიკროძრავები.

ბ) მინი რობოტის კონსტრუქცია და რობოტი-მხატვარი



გ) ნახატების ნიმუშები

კიდევ ერთი კონსტრუქცია წარმოდგენილია ნახ. 3.16-ზე. ეს რობოტი აღჭურვილია ორი ძრავით. მას შეუძლია უფრო რთული მოძრაობის შესრულება. ტექნიკურად სქემა ისევ საკმაოდ მარტივია, მაგრამ ფუნქციონალურად მისი შესაძლებლობანი მკვეთრად გაზრდილია. შესაძლებელია ეს რობოტი

ისეთნაირად დავაპროგრამოდ, რომ მოძრაობისას მიყვეს ხაზს. ასეთი კონსტრუქციები საკმაოდ პოპულარულია. მათ იყენებენ სახატავად. ამ შემთხვევაში ნახატის სახე განისაზღვრება პროგრამით. ზოგი ასეთი რობო-ნახატი გამოფენილია თანამედროვე სამხატვრო გალერეაში.

ციფრული ელექტრონიკის კომპონენტების მახასიათებლები

აქამდე ჩვენ ვიხილავდით ელექტრულ სიგნალებს იდეალიზირებულად, მხოლოდ 1 და 0 ლოგიკური დონეების სახით. რეალურ ციფრულ ელექტრონიკაში ამ და სხვა მნიშვნელობებს ენიჭება სავსებით კონკრეტული ფიზიკური შინაარსი, რომელიც წარმოდგენილია ელექტრონული ხელსაწყოების ტექნიკური პარამეტრების სახით. რეალურ ელექტრონულ ხელსაწყოებთან მუშაობისას საჭიროა ამ პარამეტრების ცოდნა. განვიხილოთ ეს პარამეტრები:

1. **ხელსაწყოს კვების ძაბვა**. ყველა მიკროსქემა მუშაობს გარკვეულ მისთვის განკუთვნილ ძაბვაზე. ნაკლები ძაბვის მოდებისას ის შესაძლოა ვერ "ჩაირთოს" ან სრულად ვერ იფუნქციონეროს, ზედმეტი ძაბვის მოდებისას კი "გადაიწვას" – რომელიმე ნაწილის მწყობრიდან გამოსვლის გამო შეწყდეს ფუნქციონირება.
2. **შესავალი ლოგიკური 0-ს და 1-ს დონეები.** 0-სა და 1-ის შესაბამისი მნიშვნელობებისთვის საზოგადოდ მიუთითებენ ძაბვათა დიაპაზონებს, მაგალითად: 0 – არაუმეტეს 2 ვოლტისაა, 1 – არანაკლებ 3 ვოლტისაა. ასეთ შემთხვევაში 2.5 ვოლტის მიწოდებისას გაურკვეველი იქნება შედეგი.
3. **გამოსავალი ლოგიკური 0-ს და 1-ს დონეები.** საზოგადოდ გამოსავალ მონაცემებში 0-სა და 1-ის დონეები შეიძლება მნიშვნელოვნად განსხვავდებოდეს, მაგრამ ყოველთვის გამოსავალი 0-ის დონე არ აღემატება შესავალი 0-ის დონეს, და გამოსავალი 1-ის დონე არაა შესავალი 1-ის დონეზე დაბალი.
4. **დენი, რომელსაც მოიხმარს მოწყობილობა.** საზოგადოდ ცდილობენ ეკონომიური ენერგომოხმარების კონტროლერების შერჩევას.
5. **შესავალი დენი.** ჩვეულებრივ მიუთითებენ მაქსიმალურად შესაძლებელ მნიშვნელობას.
6. **გამოსავალი დენი.** ჩვეულებრივ მიუთითებენ მაქსიმალურად შესაძლებელ მნიშვნელობას. მის მნიშვნელობაზეა დამოკიდებული თუ რა სიმძლავრის ხელსაწყოს მიერთებაა შესაძლებელი გამოსავალზე.
7. **იმპულსის ხანგრძლივობა.** ჩვეულებრივ მიუთითებენ მინიმალურ შესაძლებელ მნიშვნელობას. ის აღწერს თუ რა უმცირესი ხანგრძლივობის იმპულსის აღქმა და დამუშავება შეუძლია მიკროკონტროლერს.
8. **მაქსიმალური სამუშაო სიხშირე.** ეს მახასიათებელი მეტად მნიშვნელოვანია ის აღწერს ხელსაწყოს მუშაობის სიჩქარეს ანუ რა სიხშირით ხორციელდება თითოეული ოპერაცია ხელსაწყოში. ცხადია შედარებით მცირე სიხშირეზე მომუშავე (შედარებით ნელ) ძველ (ადრე გამოშვებულ) მიკროკონტროლერებს არ შეუძლიათ სრულფასოვნად იმუშაონ უფრო მაღალსიხშირეზე მომუშავე (ახალ, შედარებით ჩქარ) თანამედროვე მიკროპროცესორებთან.
9. **0-1 და 1-0 გადართვის დრო.** ჩვეულებრივ მიუთითებს იმ მინიმალურ დროს რომლის განმავლობაშიც სიგნალი შეიძლება შეიცვალოს თავისი დიაპაზონის 10%-დან 90%-მდე ან პირიქით. ცხადია ეს მახასიათებელი მჭიდრო კავშირშია ხელსაწყოს სამუშაო სიხშირესთან.
10. **სიგნალის გავრცელების დაყოვნება.** უმეტეს შემთხვევაში მიკროკონტროლერი შედგება რამდენიმე კასკადისგან, რომლებიც მყისიერად არ რეაგირებენ შემოსულ სიგნალზე. ამიტომ სიგნალის გავრცელების დაყოვნების დრო წარმოადგენს მის მერ ყოველი კასკადის გავლისას დახარჯულ დროთა ჯამს.