

13. Ponavljanje stringova, funkcija, bitwise operacija

STRINGOVI

U Arduinu imenom „string“ nazivaju se dvije stvari:

1. niz znakova (karaktera, „char“) - kao u C jeziku
2. arduinov String koji je objekt

U nastavku će kroz zadatke biti detaljnije objašnjeni svaki do njih i kada je koji praktičan.

STRING KAO NIZ KARAKTERA

String je niz („array“) karaktera („char“) koji još na kraju ima specijalan znak 0 ("null terminated string").

PRIMJER 1.

- a) Prepišite sljedeći kod i promotrite što se dogodi. Promotrite koliko je dugačak stvoreni niz na početku. Primijeti kako se pristupa pojedinim elementima niza.

```
void setup() {  
    char my_str[6]; // an array big enough for a 5 character string  
    Serial.begin(9600);  
    my_str[0] = 'H';  
    my_str[1] = 'e';  
    my_str[2] = 'l';  
    my_str[3] = 'l';  
    my_str[4] = 'o';  
    my_str[5] = 0;    Serial.println(my_str);  
}  
  
void loop() {}
```

- b) Ako sada prepišete sljedeći kod, postoji li razlika u ispisanome?

```
void setup() {  
    char my_str[] = "Hello";  
    Serial.begin(9600);  
    Serial.println(my_str);  
}  
  
void loop() {}
```

Je li sada bilo potrebno zadati duljinu niza na početku? Kompajler je samostalno odredio duljinu niza te dodao 0 na kraj tog niza.

ZADATAK 1. Po uzoru na 1b) primjer, napravi string „vrijeme“ koji će na početku glasiti „Danas je sunčano“, te ga ispiši. Nakon toga u kodu dodaj liniju u kojoj mijenjaš 8 element tog niza i postavljaš ga na 0. Ponovno ispiši niz. Za kraj promijeni karaktere na pozicijama od 9 do 13 tako da piše umjesto „sunčano“ riječ „toplo“. Ponovno ispiši niz.

Da li je potrebno promijeniti 8 element? Da li je potrebno dodati nešto iza slova 'o' u „toplo“?

Ovdje smo mogli vidjeti da funkcija „println“ ispisuje samo do prvog pojavljivanja znaka 0.

Da li stavljanja 0 znaka ispred nekih drugih karaktera skraćuje niz? Testiraj to koristeći funkcije „strlen“ i „sizeof“.

Npr. `len1=strlen(my_str); Serial.println(len1);`

`len2=sizeof(my_str); Serial.println(len2);`

Koja je razlika ove dvije funkcije. Napiši što nam koja vraća:

STRLEN vraća _____

SIZEOF vraća _____

ZADATAK 2. Kopiraj postojeći string „my_str“ u novi string „new_str“ koristeći funkciju „strcpy“ koja je definirana `strcpy(new_string, current_string);` Ispisivanjem provjeri je si li dobio opet isti niz.

ZADATAK 3. Na postojeći string „my_str“ dodaj novi string „ i proljetno!“ koristeći funkciju „strcat“ koja je definirana kao `strcat(current_string, string_to_add_to_end);` Ispisivanjem provjeri je si li dobio niz „Danas je toplo i proljetno!“.

STRING KAO OBJEKT

Objekt je konstrukt koji istovremeno ima svojstva varijable ali i funkcija, odnosno na varijabli koja je definirana kao taj objekt definirane su metode (funkcije) koje se mogu vršiti na njoj.

U sljedećem primjeru obraditi će se nekoliko takvih metoda (funkcija) koje su definirane na tipu String.

PRIMJER 2. Prepiši sljedeći kod, te zatim promotri što si dobio ispisano na Serial monitoru. Razjasni što radi koja funkcija. U sljedećim zadacima primjenjivati će se neke od tih funkcija.

```
void setup() {
  String my_str = "This is my string.";
  Serial.begin(9600);

  Serial.println(my_str);

  // change the string to upper-case
  my_str.toUpperCase();
  Serial.println(my_str);

  // overwrite the string
  my_str = "My new string.";
  Serial.println(my_str);

  // replace a word in the string
  my_str.replace("string", "Arduino sketch");
  Serial.println(my_str);

  // get the length of the string
  Serial.print("String length is: ");
  Serial.println(my_str.length());
}

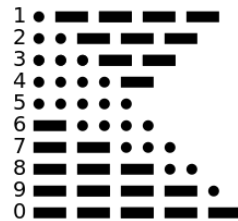
void loop() { }
```

ZADATAK 4. Spoji 2 LED-ice različite boje na Arduino, crvenu i plavu. Preko *Serial Monitor-a* javi korisniku da upiše koju LED-icu želi upaliti/ugasiti. Dakle, korisnik upisuje „naredba boja“ (pr. „ugasi crvena“, „upali plava“). Naredbe mogu biti samo „upali“ i „ugasi“, a boja može biti samo „crvena“ i „plava“. Prema takvom korisnikovom unosu mijenjaj stanje LED-ica.

Korisne funkcije bi ti mogle biti: `substring()`, `indexOf()` itd. Da bi saznao kako ih koristiti googlaj njihovo ime i „arduino“.

ZADATAK 5. Modificiraj prethodni program tako da ne ovisi o malim/velikim slovima, tj. ako korisnik napiše „Ugasi Crvena“ ili „UGASI crvena“, rezultat je jednak kao da je napisao „ugasi crvena“. Korisne bi mogle biti funkcije `toUpperCase()`, `toLowerCase()` i `compareTo()`.

ZADATAK 6. Napravi program kojem ćeš preko Serial Monitora unijeti jednoznamenkasti broj, a zatim će program ispisati broj koji smo unijeli, ali kodiran Morseovim kodom.



Savjet je da sve znakove Morseovog koda staviš u jedan niz, a zatim ovisno o upisanom broju ispišeš samo član niza na toj poziciji.

Pomoć za definiranje niza stringova (primjer): `String morse[3] = { "prvi string", "drugi string", "treći string" };`

PISANJE FUNKCIJA

Ukoliko želimo pojednostaviti kod i napraviti ga preglednijim, onda je korisno za znati kako napisati svoju funkciju koja će nešto raditi. Potrebno je prvo definirati tu funkciju. Sintaksa glasi:

Tip_varijable_koju_funkcija_vraca (npr. String, Int, Float, ili void ako ništa ne vraća) Ime_funkcije (tip_parametra (npr. String, Int itd) Ime_parametra,){
Tijelo_funkcije...

}

Npr. `String to_morse(int broj){`
 `String morse_string=...;`
 `return morse_string;`

`}`

A zatim tu funkciju treba i pozvati negdje unutar programa. Poziva se samo s imenom i parametrima koje joj predajemo.

Pokušaj to sad primijeniti na sljedeći zadatak.

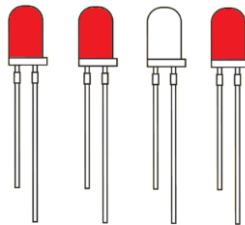
ZADATAK 7. Modificiraj prethodni zadatak, tako da možeš upisati višeznamenkasti broj. Napiši funkciju „to_morse“ koja će jednoznamenkasti broj (koji joj se preda kao parametar) pretvoriti u String. U glavnom programu potrebno je uneseni String preko Serial monitora rastavljati na jednoznamenkaste brojeve te ispisivati vraćene Stringove.

BITWISE OPERACIJE

Zajedno će se objasniti na ploči ☺ a zatim naučeno primjeniti na sljedeći zadatak.

ZADATAK 8. Spoji 4 LED-ice na Arduino. Njih ćemo koristiti za binarni prikaz broja. Na *Serial Monitor* je potrebno javiti korisniku da unese jedan broj koji želi pretvoriti u binarni zapis. Pošto imamo četiri LED-ice na raspolaganju, neka taj broj bude u rasponu od 0 do 15.

Ako korisnik unese broj izvan raspona, na *Serial Monitor* se ispisuje poruka korisniku da ponovno upiše broj iz zadanog raspona. Uneseni broj pretvori u binarni (koristi operacije nad bitovima) te zatim prikaži na LED-icama (1 svijetli, 0 ne svijetli).



PRIMJER IZLAZA ZA BROJ 13

Za operacije nad bitovima googlaj „bitwise operators arduino“, a posebno „bitwise AND“ i „bitshift LEFT“. Koristan ti može biti i „Bit Math Tutorial“ (prva stvar koju dobiješ kad to upišeš u google).

ZADATAK 9. ZA BRZE: Modificiraj prethodni program tako da se pretvaranje broja u stanja 4 ledice radi u funkciji `toLedSignals` koja je definirana na način:

```
void toLedSignals (int num, int* led1, int* led2, int* led3,
int* led4) {}
```

Unutar funkcije stanja `led1`, `led2` itd mijenjaš pristupajući im koristeći pointere:

```
*led1=...;
```

Fukciju `toLedSignals` iz loopa ćeš pozvati predajući adrese `led1`, `led2`, `led3`, i `led 4` na način:

```
toLedSignals (inputNum, &led1, &led2, &led3, &led4);
```

ZADATAK 9. ZA BRZE: Napravi program sličan programu iz 8 zadatka, ali sada ti unosi niz Morseovih znakova ('.' i '-'), koje ćeš odvajati razmakom ' '. Program mora dekodirati broj poslan Morseovim znakovima i ispisati koji je to bio broj. Također napravi da cijeli proces dekodiranja radi funkcija koju ćeš napisati npr. `int MorseToNum (String inputString){}`.