Poglavlje 15

Alat za generisanje dokumentacije

15.1 Uvod

Dokumentacija često predstavlja stavku koja se zanemaruje prilikom rada na nekom projektu. U nekim specijalnim situacijama, ukoliko postoje određeni ograničavajući faktori (na primer vreme), ovo se može smatrati i opravdanim. Ipak, u opštem slučaju, dokumentovanje projekta treba da bude stvar koja se *de facto* podrazumeva. Razlog za to je sledeći.

Naime, dokumentacija prati razvoj svakog projekta i, ujedno, povećava njegov kvalitet. Dobro napisana dokumentacija pomaže prilikom, kako razvoja, tako i održavanja nekog projekta ili već gotovog proizvoda. Prema nekim procenana, održavanje nekog softverskog proizvoda oduzima i do *tri* puta više vremena od samog razvoja datog proizvoda, u okviru njegovog životnog veka. U tom procesu održavanja, čitanje već napisanog koda predstavlja komponentu koja programeru oduzima najviše vremena.

Pored toga, još jedna važna uloga dokumentacije jeste i u *prenosu znanja*. Naime, broj različitih ljudi koji trenutno rade na nekom projektu je varijabilan. Za nove programere je potrebno da, u što je moguće kraćem vremenskom roku, dobiju potpunu sliku o samom projektu. Dobra dokumentacija značajno ubrzava ovu proceduru.

Prilikom rada na softverskim projektima, pod dokumentacijom se najčešće podrazumeva dokumentacija koda. Prilikom dokumentovanja delova koda, generalna preporuka jeste dati odgovor na naredna tri pitanja.

- Koja je uloga datog dela koda?
- Kako dati deo koda izvršava tu ulogu?
- Kako neko može iskoristiti dati deo koda na nekom drugom mestu?

Kako bi se sama procedura dokumentovanja olakšala i ubrzala, u praksi postoje različiti alati za automatsko generisanje dokumentacije. Jedan od takvih alata jeste i *Doxygen*, čije objašnjenje sledi u nastavku.

15.2 Kreiranje dokumentacije korišćenjem alata Doxygen

Doxygen predstavlja alat otvorenog koda koji omogućava automatizovano generisanje projektne dokumentacije. Dokumentacija, pri tome, mora biti sadržana unutar samog izvornog koda, u obliku komentara koji poštuju format koji će biti opisan kasnije u tekstu.

Alat je moguće besplatno preuzeti sa zvanične *Doxygen* stranice, putem sledećeg linka:

https://www.doxygen.nl/index.html

Sam alat je moguće koristiti na dva različita načina:

- iz komandne linije
- pomoću Doxywizard-a

Rad sa *Doxygen* alatom iz komandne linije

Standardna procedura generisanja dokumentacije upotrebom alata iz komandne linije je sledeća:

- 1. Potrebno je dokumentovati izvorni kod u skladu sa odgovarajućim pravilima.
- 2. Potrebno je generisati konfiguracionu datoteku pozivom alata sa odgovarajućom opcijom, na sledeći način:

gde parametar <konfiguraciona datoteka> predstavlja naziv datoteke koja će biti generisana.

3. Potrebno je modifikovati parametre unutar konfiguracione datoteke u skladu sa projektom za koji se generiše dokumentacija. Detaljno objašnjenje svih parametara je moguće pronaći na narednom linku:

https://www.doxygen.nl/manual/config.html

4. Konačno, potrebno je generisati dokumentaciju korišćenjem alata, na sledeći način:

gde parametar <konfiguraciona datoteka> predstavlja naziv prethodno modifikovane konfiguracione datoteke.

Korisna opcija jeste i opcija --help, koja omogućava kratko objašnjenje načina na koje se alat može koristiti iz komandne linije.

Rad sa Doxygen alatom korišćenjem Doxywizard-a

Drugi način rada sa *Doxygen* alatom podrazumeva upotrebu ugrađenog *Doxywiz-ard*-a. Ovaj pristup je značajno lakši za korišćenje od prethodnog, te se, u praksi, uglavnom i češće koristi. Navigacijom kroz prozore *wizard*-a, na jednostavan način je moguće popuniti sve neophodne parametre koji se tiču trenutnog projekta, kao što su naziv projekta, ulazni direktorijum u okviru kog je smešten dokumentovani izvorni kod, izlazni format i drugo.

Na narednim slikama, prikazani su osnovni delovi Doxywizard-a.

Broxygen GUI frontend	- L X	Br Doxygen GUI frontend		- L	1 X
File Settings Help		File Settings Help			
Specify the working directory from which doxygen will ru	·	Specify the working directory from which doxygen will run			
1	Select			9	elect
Configure doxygen using the Wizard and/or Expert tab,	then switch to the Run tab to generate the documentation	Configure doxygen using the Wizard and/or Expert tab, t	en switch to the Run tab to generate the documentation		
Wizard Expert Run		Wizard Expert Run			
ropics Project Mode Output Diagrams	Provide some information about the project you are documenting Project name: My Project Project sympose: Project version or ds:	ropics Project Mode Output Diagrams	Select the desired extraction mode: © Documented entities only All Drates Include cross-referenced source code in the output Select programming language to optimize the results for		
	Project logo: Select No Project logo selected.		Optimize for C++ output Optimize for C++/OLI output Optimize for C++/OLI output Optimize for Java or C# output		
	Specify the directory to scan for source code Source code directory: Select Select		Optimize for C or PHP output Optimize for Fertran output Optimize for VHDL output Optimize for VHDL output		
	Specify the directory where doxygen should put the generated documentation Destination directory: Select		C openize for succe surput		
	Previous Next		Previous		Next

(a) Karakteristike projekta



൙ Doxygen GUI frontend	-	0 ×	Poxygen GUI frontend	-		×
File Settings Help			File Settings Help			
Specify the working directory from which doxygen will ru- Configure doxygen using the Wizard and/or Expert tab, Wizard Expert Run	In the next the Run tab to generate the documentation	Select	Specify the working directory from which dorugen will no Configure dorugen using the Water of and/or Expert table, then within to the Run table to generate the document Water of Expert Run	ston	Select.	
Project	Select the output format(s) to generate		Specify additional command line options for running doxygen			
Output	plain HTML		Run doxygen Status: not running	Condensed	Save log	
Diagrams	with neighbor panel orgenese HTML (Jmm) With search function Wards and function Wards and function wards function wards function wards function wards function		Denne TER, sonool Ourpud produced by derrygen	Show configuration		
	ar intermediate format for PSP ar intermediate format for PSP ar intermediate format for PSF both Cest Format (PTP) 394 Cochook Previous	Next				
		d.				

(c) Izbor izlaznog formata

(d) Run meni

Slika 15.1: Izgled Doxywizard-a

Detaljno objašnjenje Doxywizard-a je moguće pronaći na narednom linku:

https://www.doxygen.nl/manual/doxywizard_usage.html

Rad sa Doxygen alatom u okviru razvojnog okruženja Eclipse

Pored navedena dva, standardna načina, razvojno okruženje Eclipse, takođe podržava rad sa ovim alatom preko odgovarajućeg dodatka (eng. *plugin*) zvanog *eclox*.

Instalacija eclox dodatka vrši se izborom opcije $Help \rightarrow Install New Software$ u okviru Eclipse okruženja. Po otvaranju prozora, u polju Work with potrebno je upisati sledeći link:

http://anb0s.github.io/eclox

i kliknuti *Add.* Po iskakanju prozora *Add Repository*, u polju *Name* dodeljuje se ime eclox dodatku, na primer, *Doxygen Plugin.* Ovo je prikazano na slici 15.2.



Slika 15.2: Prozor Add Repository

Nakon toga, potrebno je selektovati najnoviju dostupnu verziju *eclox* dodatka i *Eclox Doxygen.* Ovo je ilustrovano na slici 15.3. Nakon toga, potrebno je ispratiti preostale korake do završetka instalacije.

Carl Install					×
Available Software Check the items that you wish to install.					
Work with: Doxygen Plugin - http://anb0s.github.io/eclox		~	Add	Man	ige
type filter text				Selec	t All
Name > □ 000 eclos 0.10 > □ 000 Eclos 0.11 > □ 000 Eclos 0.12 > □ 000 Eclos 0.13 □ 000 Eclos 0.13 <th>Version</th> <th></th> <th></th> <th>Desele</th> <th>ct All</th>	Version			Desele	ct All
Show only the latest versions of available software Group items by category Show only software applicable to target environment Contact all update sites during install to find required software	Hide items that are already installed What is <u>already installed</u> ?				2
0	< Back Next >		Finish	Cano	el

Slika 15.3: Izbor dodatka u okviru menija za izbor dodataka

Nakon instalacije eclox dodatka, potrebno je podesiti putanju do izvršne verzije Doxygen-a, pomoću opcije Window \rightarrow Preferences \rightarrow Doxygen, kao na slici 15.4.

Preferences				_		×
type filter text	Doxygen			¢	• =	₩ 000
> General > AVR	Choose among avai	lable doxyge	en versions, the one	you would l	like to us	e.
 ChangeLog Docker Docygen Misc Help Install/Update Library Hover Mylyn Oomph Remote Development S RM 	_ @ unknown	Default Bundled			Re	dit
 Run/Debug SWTChart Extensions Terminal TextMate Tracing Validation Version Control (Team) XML 			Restore [Defaults	Appl	у
? 🏜 🖆 🖲	1		Apply and 0	lose	Cancel	

Slika 15.4: Podešavanje putanje do izvršne verzije Doxygen-a

Kako bi bilo omogućeno korišćenje
 Doxygen-a, u okviru projekta potrebno je dodati konfiguracionu dato
teku, pomoću opcije $New \rightarrow Other \rightarrow Other \rightarrow Doxyfile.$ Ovo je ilustrovano na slici 15.5.

Select a wizard	-		×
Select a wizard			2
Wizards:			
>			^
> & Valgrind > & XML > & Other			~
? < Back Next > Finish		Cance	ł

Slika 15.5: Dodavanje konfiguracione datoteke

Nakon toga je potrebno podesiti putanju do direktorijuma u kojem će biti kreirane *Doxygen* datoteke. Takođe, potrebno je uneti ime *.doxyfile* datoteke, kao na slici 15.6.

		- v
Wew Doxygen Configuration		ц х
Doxygen Configuration		
Creates a new Doxygen configuration file.		@
Enter or select the parent folder:		
test_project		
☆ (> ↓)		
> 😂 test_project		
Doxyfile name: test project.doxyfile		
Advanced >>		
A Back Next > Fire	nish	Cancel
i i i i i i i i i i i i i i i i i i i		

Slika 15.6: Podešavanje putanje i unos imena .doxyfile datoteke

Dvoklikom na ime .doxyfile datoteke, otvara se prozor u kojem se definišu opcije kao što su naziv projekta, ulazni direktorijum (koji sadrži datoteke koje će biti skenirane), izlazni direktorijum koji će sadržati generisanu dokumentaciju, izlazni format i drugo. Konfiguracioni prozor Doxygen datoteke je prikazan na slici 15.7.

🗋 test_project.doxyfile 🛛	
Basic	
Project	Output Formats
Name: PLS7Shield	⊡ HTML
Version or Identifier	O plain HTML
Version of identifier: 1.0	with trames and navigation tree
Input directories:	with search function (requires PHP enabled server)
	Add LaTeX
	as itermediate format for hypedlinked PDF
	as itermediate format for PDF
	Up as itermediate format for PostScript
	Man Pages
	Down Rich Text Format
Scan recursively	
Output Directory:	Browse Diagrams to Generate
	○ No diagrams
Mode	Use built-in diagram generator
Select the desired extraction mode:	O Use dot tool from the GraphViz package to generate:
documented entities only	✓ class diagrams
\bigcirc all entities	Collaboration diagrams
Include cross-referenced source code in the output	✓ include dependency graph
Ontimize results for:	v overall class bierarchy
○ C++	call graphs
⊖ Java	
. © C	
○ C#	
Basic Advanced	

Slika 15.7: Konfiguracioni prozor .doxyfile datoteke

Dokumentacija se generiše pomoću opcije *Build Documentation* u izabranom formatu i nalazi se u odgovarajućem izlaznom folderu. U nastavku je objašnjen način pisanja komentara prilagođenih ovom alatu.

15.3 Stil pisanja komentara za Doxygen

Prilikom generisanja dokumentacije, *Doxygen* skenira izvorne .h i .c datoteke u ulaznom direktorijumu i pri tome traži komentare koji su propisno formatirani. Na primer, zaglavlje datoteke može imati format prikazan na narednom listingu.

```
/**
* @file ime_datoteke.c
* @brief Dpis datoteke
* @author Ime Prezime
* @date DD-MM-YYYY
* @version 1.0
*/
```

Listing 15.1: Validan format dokumentovanja zaglavlja

U okviru zaglavlja, svaka linija komentara počinje karakterom '*'. Polja se označavaju ključnim rečima kojima prethodi karakter '@' (ili '\'), nakon čega se navodi njihov opis. Listu svih dostupnih ključnih reči je moguće pronaći na sledećem linku.

https://www.doxygen.nl/manual/commands.html

Njihovo objašnjenje daleko prevazilazi okvire ovog praktikuma, te će na ovom mestu biti izostavljeno. Svakako, zainteresovanom čitaocu se preporučuje da ih detaljnije prouči.

Ukoliko korisnik unese "/**" u redu koji prethodi prototipu funkcije i nakon toga pritisne taster *Enter*, Doxygen će automatski generisati blok za opis funkcije:

```
/**
* Oparam str
* Oparam n
* Oreturn
*/
int16_t myFunction(int8_t* str, int16_t n)
```

Listing 15.2: Validan format dokumentovanja funkcije

Polja **Oparam** sadrže opise ulaznih argumenata, dok polje **Oreturn** opis povratne vrednosti.

Doxygen uzima u obzir i jednolinijske komentare koji počinju sa tri kose crte – "///", kao što je i prikazano na listingu 15.3.

/// Primer jednolinijskog komentara
int16_t var;

Listing 15.3: Validan format linijskog komentara

U nastavku je, u okviru Primera 15.3.1, dat izvorni kod prilagođen generisanju dokumentacije pomoću *Doxygen*-a.

Primer 15.3.1. Napisati program koji implementira treperenje ugrađene diode na Arduino platformi. Izvorni kod programa je potrebno dokumentovati u skladu sa prethodno opisanim pravilima.

Rešenje:

```
/**
* @file main.c
* Cbrief Aplikacija koja implementira ogranicen broj treptanja
    diode
* Cauthor Ime Prezime
* @date 14-03-2021
* Quersion 1.0
*/
#include <avr/io.h>
#include <avr/interrupt.h>
#include <stdint.h>
/// Makro za podesavanje visoke vrednosti signala na pinu
#define HIGH 1
/// Makro za podesavanje niske vrednosti signala na pinu
#define LOW 0
/// Makro za podesavanje izlaznog smera pina
#define OUTPUT 1
/// Makro za podesavanje ulaznog smera pina
#define INPUT 0
/// Makro za selektovanje porta B
#define PORT_B 0
/// Makro za selektovanje porta C
#define PORT_C 1
/// Makro za selektovanje porta D
#define PORT_D 2
/// Makro za selektovanje pina na koji je povezana dioda
#define DIODE_PIN 5
/// Promenljiva koja skladisti broj milisekundi proteklih od
   pokretanja aplikacije
volatile uint32_t ms = 0;
```

```
/**
* pinPulse – Funkcija koja implementiran podizanje i spustanje
    vrednosti na pinu
* odgovarajucom brzinom
* Cparam port – ulaz tipa uint8 t – Port na kojem je potrebno
   implementirati funkcionalnost
* Cparam pin – ulaz tipa uint8_t – Pin na kojem je potrebno
   implementirati funkcionalnost
* @param period - ulaz tipa uint32_t - Perioda promene
   vrednosti na pinu
* @return Nema povratnu vrednost
*/
void pinPulse(uint8_t port, uint8_t pin, uint32_t period);
/**
* pinSetValue – Funkcija koja postavlja vrednost na pinu
* Oparam port – ulaz tipa uint8_t – Port na kojem je pin ciju
   vrednost potrebno postaviti
* @param pin - ulaz tipa uint8_t - Pin ciju je vrednost
   potrebno postaviti
* @param value – ulaz tipa uint8_t – Vrednost koju je potrebno
    postaviti na pin
* @return Nema povratnu vrednost
*/
void pinSetValue(uint8_t port, uint8_t pin, uint8_t value);
/**
* pinInit – Funkcija koja implementiran inicijalizaciju pina
* @param port – ulaz tipa uint8_t – Port na kojem je pin koji
   je potrebno inicijalizovati
* Cparam pin – ulaz tipa uint8_t – Pin koji je potrebno
   inicijalizovati
* Cparam direction – ulaz tipa uint8 t – Smer prema kojem je
   potrebno inicijalizovati pin
* @return Nema povratnu vrednost
*/
void pinInit(uint8_t port, uint8_t pin, uint8_t direction);
/**
* timerODelayMs - Funkcija koja implementira pauzu u broju
   milisekundi koji je prosledjen
* kao parametar
* @param delay_length - ulaz tipa uint32_t - Duzina pauze u
   milisekundama
* @return Nema povratnu vrednost
*/
void timerODelayMs(uint32_t delay_length);
/**
* timerOMillis - Funkcija koja, na bezbedan nacin, vraca kao
```

```
povratnu vrednost broj milisekundi proteklih od pocetka
   merenja vremena
* @return Broj milisekundi proteklih od pocetka merenja
   vremena
*/
uint32 t timerOMillis();
/**
* timer0Init - Funkcija koja inicijalizuje timer 0 tako da
   pravi prekide
* svake milisekunde
* @return Nema povratnu vrednost
*/
void timerOInit();
/**
* main – funkcija koja implementiran glavni deo aplikacije
* @return Nema povratnu vrednost
*/
int16_t main()
{
   uint32_t period = 1000; // Period jednog treptaja
   uint8_t repetitions = 5; // Broj treptaja
   int16_t i;
   // Inicijalizacija
   pinInit(PORT_B, DIODE_PIN, OUTPUT);
   timerOInit();
   // Glavna petlja
   while (1)
   {
       // Treptanje
       for (i = 0; i < repetitions; i++)</pre>
           pinPulse(PORT_B, DIODE_PIN, period);
       // Kraj
       while (1)
       ;
   }
   return 0;
}
void pinPulse(uint8_t port, uint8_t pin, uint32_t period)
{
// Poluperioda u kojoj pin ima visoku vrednost
```

```
pinSetValue(port, pin, HIGH);
   timerODelayMs(period/2);
   // Poluperioda u kojoj pin ima nisku vrednost
   pinSetValue(port, pin, LOW);
   timerODelayMs(period/2);
}
void pinSetValue(uint8_t port, uint8_t pin, uint8_t value)
ł
   // Postavljanje vrednosti pina
   switch(port)
   ſ
       case PORT B:
           if (value == HIGH)
              PORTB |= 1 << pin;</pre>
           else
              PORTB &= ~(1 << pin);</pre>
       break;
       case PORT_C:
           if (value == HIGH)
              PORTC |= 1 << pin;</pre>
           else
              PORTC &= ~(1 << pin);</pre>
       break;
       case PORT D:
           if (value == HIGH)
              PORTD |= 1 << pin;</pre>
           else
              PORTD &= ~(1 << pin);</pre>
       break;
       default:
       break;
   }
}
void pinInit(uint8_t port, uint8_t pin, uint8_t direction)
{
   // Inicijalizacija smera pina
   switch (port)
   {
       case PORT_B:
           if (direction == OUTPUT)
```

```
DDRB |= 1 << pin;
         else
            DDRB &= ~(1 << pin);
      break;
      case PORT C:
         if (direction == OUTPUT)
            DDRC |= 1 << pin;
         else
            DDRC &= \sim (1 << pin);
      break;
      case PORT_D:
         if (direction == OUTPUT)
            DDRD |= 1 << pin;
         else
            DDRD &= ~(1 << pin);
      break;
      default:
      break;
   }
}
void timerODelayMs(uint32_t delay_length)
{
   // trenutak pocevsi od kog se racuna pauza
   uint32_t t0 = timer0Millis();
   // implementacija pauze
   while(timerOMillis() - t0 < delay_length)</pre>
   ;
}
uint32_t timerOMillis()
{
   uint32_t tmp;
   cli();
               // Zabrana prekida
   tmp = ms;
               // Ocitavanje vremena
   sei();
               // Dozvola prekida
   return tmp;
}
void timer0Init()
```

```
{
   TCCROA = 0x02; // tajmer 0: CTC mod
   // Provera frekvencije takta prilikom kompilacije
   #if F_CPU > 20000000
   #error "Frekvencija..takta..mora..biti..manja..od..20MHz!"
   #endif
   // Inicijalizacija promenljivih za preskaler i periodu
       tajmera O
   uint32_t n = F_CPU / 1000;
   uint8_t clksel = 1;
   // Odredjivanje vrednosti preskalera i periode tajmera O
   while (n > 255)
   ſ
       n /= 8;
       clksel++;
   }
   TCCROB = clksel; // tajmer 0: Podesavanje preskalera
   OCROA = (uint8_t)(n & Oxff) - 1; // tajmer 0: Podesavanje
       periode
   TIMSKO = 0x02; // tajmer 0: Dozvola prekida
   sei(); // Globalna dozvola prekida
}
/**
* ISR - prekidna rutina tajmera 0 u modu CTC
*/
ISR(TIMER0_COMPA_vect)
{
   // Inkrementovanje broja milisekundi koje su prosle od
       pokretanja aplikacije
   ms++;
}
```

Listing 15.4: Izvorni kod sa Doxygen komentarima

Generisanoj dokumentaciji, u HTML formatu, je moguće pristupi otvaranjem *index.html* stranice, u proizvoljnom internet pretraživaču. Izgled ove stranice za prethodni primer je dat na slikama 15.8, 15.9 i 15.10.



Slika 15.8: Stranica Index.html

Trepere Program koji im	enje d	Ode 0.1 eperenje ugrađene diode na Arduino p	latformi	
Main Page	Files +			Qr Search
File List				
Here is a list of	of all docum	ented files with brief descriptions:		
🖹 main.c	Aplikacija	i koja implementira ogranicen broj	treptanja diode	
				Generated by

Slika 15.9: Lista izvornih datoteka

	erenje d	liode 0.1	Í
Main Pa	ige Files •		Qr Search
main.	c File Ref	erence	Macros Functions Variables
Aplikacija	i koja implemer	ntira ogranicen broj treptanja diode. More	
#include	e ≺avr/io.h≻		
<pre>#include</pre>	e <avr inter<="" td=""><td>rupt.h></td><td></td></avr>	rupt.h>	
#include	<pre><stdint.h></stdint.h></pre>		
Macro	s		
#define	HIGH 1 Makro za pod	desavanje visoke vrednosti signala na pinu.	
#define	LOW 0 Makro za pod	lesavanje niske vrednosti signala na pinu.	
#define	OUTPUT 1 Makro za pod	Jesavanje izlaznog smera pina.	
#define	INPUT 0 Makro za pod	lesavanje ulaznog smera pina.	
#define	PORT_B 0 Makro za sele	ektovanje porta B.	
#define	PORT_C 1 Makro za sele	ektovanje porta C.	
#define	PORT_D 2 Makro za sele	ektovanje porta D.	
#define	DIODE_PIN Makro za sele	5 ektovanje pina na koji je povezana dioda.	
Functi	ons		
void	pinPulse (ui	int8_t port, uint8_t pin, uint32_t period)	
void	pin SetValue	e (uint8_t port, uint8_t pin, uint8_t value)	
void	pinInit (uint8	3_t port, uint8_t pin, uint8_t direction)	
void	timer0Delay	Ms (uint32_t delay_length)	
uint32_t	timer0Millis	0	
void	timer0Init ()		
int16_t	main ()		

Slika 15.10: Dokumentacija datoteke main.c 1

15.4 Zadatak za vežbu

Zadatak 15.4.1. Dokumentovati, u skladu sa *Doxygen* pravilima, pet projekata (uključujući i korišćene korisničke biblioteke), realizovanih u prethodnim poglavljima, po izboru. Upotrebom alata *Doxygen* izvršiti generisanje dokumentacije, na naredna tri načina:

- (a) upotrebom alata iz komandne linije;
- (b) upotrebom *Doxywizard*-a;
- (c) upotrebom eclox dodatka u okviru razvojnog okruženja Eclipse.