РЕГИОНАЛНИ ЦЕНАТР ЗА ТАЛЕНТЕ ВРАЊЕ

КРЕИРАЊЕ ПРОИЗВОЉНОГ ЗНАКА ЗА LCD ДИСПЛЕЈ КОРИШЋЕЊЕМ MIKROPASCAL-А ЗА PIC КОНТРОЛЕРЕ

CREATE CUSTOM CHARACTER FOR LCD DISPLAY USING MIKROPASCAL FOR PIC CONTROLLER

Аутори:

Давид Миленковић, ученик 8. разреда ОШ "Синиша Јаиић", Власотинце и Миљан Маринковић, ученик 8. разреда ОШ "Свети Сава", Гложане

Ментор:

Новица Стојилковић, дипл. енг. ел. самостални стручни сарадник шри ФДВ "Х. Црниловић- Кица"

Власотинце 2015

КРЕИРАЊЕ ПРОИЗВОЉНОГ ЗНАКА ЗА LCD ДИСПЛЕЈ КОРИШЋЕЊЕМ MIKROPASCAL-А ЗА PIC КОНТРОЛЕРЕ

CREATE CUSTOM CHARACTER FOR LCD DISPLAY USING MIKROPASCAL FOR PIC CONTROLLER

Аутори:

Давид Миленковић, ученик 8. разреда ОШ "Синиша Јаиић", Власотинце и Миљан Маринковић, ученик 8. разреда ОШ "Свети Сава", Гложане

Ментор: Новица Стојилковић, дипл. енг. ел. самостални стручни сарадник при ФДВ "Х. Црниловић- Кица"

РЕЗИМЕ

Овим се радом жели показати како се може искористити MicroPascal компајлер и његов LCD Custom Character за креирање произвољних карактера. За проверу функционалнисти направљених карактера коришћена је развојна плоча EasyPic v7 произвођача "Микроелектроника" који је и произвођач компајлера MicroPascal. За приказ знакова на развојној плочи коришћен је LCD 2x16 дисплеј. Коришћени контролер је PIC контролер произвођача микрочип.

Кључне речи:: развојна плоча, компајлер, LCD, контролер, произвољни знак, изворни код.

SUMMARY

The paper aims to show how it can be used MicroPascal compiler and its LCD Custom Character for creation of arbitrary character. To check the functionality at-made character development board was used EasyPIC 7 manufacturers "Microelectronics", which is the manufacturer of the compiler MicroPascal. To display characters on the development board was used 2x16 LCD display. Used controller is a PIC controller microchip manufacturers.

Keywords :: development board, compiler, LCD controller, custom character, the source code.

УВОД

Када се помене програмирање најчешће се мисли на прављење програма за PC рачунаре и прављење web апликација. Такво мишљење јавности пресликава се и у радовима за смотру талената при чему су већина радова намењена за PC рачунаре. Међутим у практичном животу човек се среће са многим уређајима који су такође програмирани али су програми прављени за микроконтролере а они су ништа друго него мали компјутери са тачно дефинисаним могућностима.

Овим се радом жели показати шта је све потребно за програмирање микроконтролера и то PIC контролера произвођача "микрочип" (mikrochip).

Компајлер који је коришћен је mikroPascal произвођача "МикроЕлектроника". Основа mikroPascal-а је програмски језик Pascal тако да ће Вам уколико познајете Pascal то бити од велике помоћи. Вредно је напоменути да због специфичности намена микроконтролера овај компајлер има и своје специфичности.

За проверу изворних кодова коришћена је развојна плоча EasyPIC v7 такође произвођача "МикроЕлектроника".

Поред тога овим радом ће бити показано како је могуће креирати произвољни знак прављењем програма за LCD дисплеј 2x16 (дисплеј у 2 реда са по 16 знака) коришћењем MicroPascal-а као и примена тако направљеног знака.

АЛАТИ И МЕТОДЕ РАДА

За израду овог рада коришћени су следећи алати.

- РС рачунар са компајлером mikroPascal PRO for PIC и софтвером за програмирање PIC кола mikroProg Suite For PIC. Најбоље је скинути са сајта микроелектронике најновију верзију јер можете са њом програмирати и нове серије PIC контролера (387 типа контролера). Ова скинута верзија је демо верзија и све што постоји код овог компајлера ће радити а ограничење се огледа у томе да када треба да преведете написани програм он ће превести само ако је величина програма мања од 2KB. За овај рад биће коришћен контролер PIC 16f1847 који је новије серије тако да ће морати да буде коришћена новија верзија компајлера и програматора. За креирање произвољног знака биће коришћен LCD Custom Character. Он се налази у компајлеру и то на Tools->LCD Custom Character.
- За проверу напсаних кодова биће коришћена развојна плоча EasyPIC v7 којом уједно можете и да програмирате PIC контролере. На слици 1 је приказана развојна плоча EasyPIC v7.



Слика 1, развојна плоча Easy PIC v7 Picture 1, development board Easy PIC v7

- LCD дисплеј 2x16 са плавим позадинским осветлењем на коме ће бити приказан резултат израде произвољних знакова. Види слику 2.



Слика 2, LCD дисплеј 2x16 Picture 2, LCD display 2x16

Коришћење LCD Custom Character

LCD Custom Character се налази у саставу компајлера mikroPascal PRO for РІС произвођача "МикроЕлектроника".

	/olini_znaci.mpas			Ę
	program Proizvoljni_znaci;			
-				
-	<pre>var LCD_RS : sbit at RB4_bit;</pre>			
	<pre>var LCD_EN : sbit at RB5_bit;</pre>			
	<pre>var LCD_D4 : sbit at RBO_bit;</pre>			
	<pre>var LCD_D5 : sbit at RB1_bit;</pre>			
•	<pre>var LCD_D6 : sbit at RB2_bit;</pre>			
	<pre>var LCD_D7 : sbit at RB3_bit;</pre>			
10	<pre>var LCD_RS_Direction : sbit at TRISB4_b</pre>	it;		
•	<pre>var LCD_EN_Direction : sbit at TRISB5_b</pre>	it;		
	<pre>var LCD_D4_Direction : sbit at TRISB0_b</pre>	it;		
1	var LCD_D5_Direction : sbit at TRISB1_b	.1t;		
•	<pre>var LCD_D6_Direction : sbit at TRISB2_b</pre>	it;		
-	<pre>var LCD_D7_Direction : shit at TRISB3_b</pre>	it;		
-				
	var blink:boolean;			
	(Declarations section)			
20	const characteri: array[0] or byte -	(4,14,21,21,21,14,4,0))		
	procedure znak_1(pos_row, pos_char: byc	el; // znak P u cirilici		
				18
-				
Mess	ages 🔛 Quick Converter			
, Mess V Error	ages 🔛 Quick Converter s 🗸 Warnings 🗸 Hints			
, Mess V Error	rages 20 Quick Converter s V Warnings V Hints Message No. Messag	je Text	Unit	
Mess V Error	ages 🔐 Quick.Converter 🛛	pe Text	Unit	
, Mess V Error ine	rages 🔛 Quick Converter 's 🗸 Warnings 🗸 Hinks Message No. Messag	je Text	Unit	
, Mess V Error ine	agas 😰 Qudk Converter s 🗸 Wanings V Hints Message No. Messag	pe Text	Unit	
, Mess V Error ine	espes 🔝 Quick Conventer s V Wannings V Hots Message No. Message	jø Text	Unit	

Слика 3, изглед компајлера Picture 3, the appearance of the compiler

100 0	1 R2 R9 m 1 d9 d 25	mE Programmer	E11 4	N N I 🖬 16 // 🤊 🖬 🕞 .	1024v768		a a 🔅	
40° - 12		MixeDrog Dower Options	CHILADAR		1 Eline avion			
Proizvo	lini_znaci.mpas	Dark you Manazaw	Same					41
	program Proiz var LCD_RS: var L	Active Comment Editor Agit Over Agit Over UpPRONE Editor Egot Ode To HTML UpPRONE Editor Egot Ode To HTML UpD Terminal UpP Terminal UpPRONE Societar UpP Terminal UppEND Bootbooder UppEND Bootb	Ctri+At+C Ctri+At+I Ctri+At+I					
•	procedure Znak_1	pos_rov, pos_cha	r: Byte);	// znak F u cirilici				
								>
 Messar Errors 	pes 🔜 Quick Converter	¥ Hints						
Line	Message No.		Message Text			Unit		





Слика 5, изглед креираног ћиричичног знака "И" са кодом Picture 5, appearance created ćiričičnog symbol "И" with the code

Креирање знака је једноставно. Одаберите тип дисплеја (има могућност за два дисплеја и то 5х7 и 5х10). Најчешће се користи 5х7 па одаберите тај. Помоћу миша креирајте произвољни знак. У **CGRAM address** увећајте поље **Char** за 1. После креирања одаберите језик за који ће Вам програм направити код и у овом случају је mikroPascal Pro. Креирање кода радите тако што кликнете на GenerateCode. Потом кликните на дугме Copy Code to ClipBoard да пренесете код у ClipBoard. Потом у компајлер пренесите код са ctrl+V (Paste). Уколико креирате више произвољних знакова промените назив константе и процедуре како не би дошло до забуне.

ДИСКУСИЈА

program Proizvoljni_znaci;

var LCD_RS : sbit at RB4_bit; // дефинисање пинова за ЛЦД дисплеј var LCD_EN : sbit at RB5_bit; var LCD_D4 : sbit at RB0_bit; var LCD_D5 : sbit at RB1_bit; var LCD_D6 : sbit at RB2_bit; var LCD_D7 : sbit at RB3_bit;

var LCD_RS_Direction : sbit at TRISB4_bit; var LCD_EN_Direction : sbit at TRISB5_bit; var LCD_D4_Direction : sbit at TRISB0_bit; var LCD_D5_Direction : sbit at TRISB1_bit; var LCD_D6_Direction : sbit at TRISB2_bit; var LCD_D7_Direction : sbit at TRISB3_bit;

var blink:boolean;

```
const character1: array[0..7] of byte = (4, 14, 21, 21, 21, 14, 4, 0);
procedure Znak 1(pos row, pos char: Byte); // знак \Phi у ћирилици
var i: Byte;
begin
Lcd Cmd(64);
for i := 0 to 7 do Lcd_Chr_CP(character1[i]);
Lcd Cmd( LCD RETURN HOME);
Lcd_Chr(pos_row, pos_char, 0);
end;
const character2: array[0..7] of byte = (14, 10, 10, 10, 10, 10, 31, 17);
procedure Znak 2(pos row, pos char: Byte); // знак Д у ћирилици
var i: Byte;
begin
Lcd Cmd(80);
for i := 0 to 7 do Lcd_Chr_CP(character2[i]);
Lcd Cmd( LCD RETURN HOME);
Lcd_Chr(pos_row, pos_char, 2);
end;
```

const character3: array[0..7] of byte = (3,5,5,5,5,21,29,0);

```
procedure Znak_3(pos_row, pos_char: Byte);
                                             // знак Л у ћирилици
var i: Byte;
begin
 Lcd_Cmd(72);
 for i := 0 to 7 do Lcd Chr CP(character3[i]);
 Lcd_Cmd(_LCD_RETURN_HOME);
 Lcd_Chr(pos_row, pos_char, 1);
end;
const character4: array[0..7] of byte = (17, 17, 19, 21, 25, 17, 17, 0);
procedure Znak_4(pos_row, pos_char: Byte); // знак И ћирилица
var i: Byte;
begin
 Lcd_Cmd(88);
 for i := 0 to 7 do Lcd_Chr_CP(character4[i]);
 Lcd_Cmd(_LCD_RETURN_HOME);
 Lcd Chr(pos row, pos char, 3);
end;
const character5: array[0..7] of byte = (18, 18, 18, 18, 18, 18, 18, 11, 1);
procedure Znak_5(pos_row, pos_char: Byte); // знак Ц ћирилица
var i: Byte;
begin
 Lcd_Cmd(96);
 for i := 0 to 7 do Lcd_Chr_CP(character5[i]);
 Lcd_Cmd(_LCD_RETURN_HOME);
 Lcd_Chr(pos_row, pos_char, 4);
end;
begin
 ANSELA := 0x00; // порт A је дигитални
 ANSELB := 0x00; // порт Б је дигитални
 blink:=true;
 TRISA := 0x00; // порт А је излазни
 PORTA := 0;
                // вредност порта А је 0
                    // порт Б је излазни
 TRISB := 0x00;
                // порт Б има вредност 0
 PORTB := 0;
 LCD Init ();
                // иницирање дисплеја
 LCD_Cmd (_LCD_TURN_ON); // дозвола рада дисплеју
 LCD Cmd ( LCD CLEAR); // брисање екрана ЛЦД
 LCD_Cmd (_LCD_CURSOR_OFF); // искљућивање курсора
while true do begin
 if blink then begin
 Znak_1(1, 7);
 Znak_2(1, 8);
```

Lcd_chr(1,9,'B'); Lcd_chr(2,4,'B'); Znak_3(2, 5); Lcd_Out(2,6,'ACOT'); Znak_4(2, 10); Lcd_chr(2,11,'H'); Znak_5(2, 12); Lcd_chr(2,13,'E'); end else LCD_Cmd (_LCD_CLEAR); Delay_ms(1000); blink:=not blink; end; end.

Резултат овог кода је да ће на LCD дисплеју блинкати натпис и то у првом реду "ФДВ" а у другом "ВЛАСОТИНЦЕ". Блинкање је са приказом натписа од 1 секунде=1000ms и без натписа у трајању од 1 секунде=1000ms. Ово је омогућено бесконачном WHILE петљом и наредбом Delay_ms(1000).



Слика 6, проказ "ФДВ ВЛАСОТИНЦЕ " на ЛЦД дисплеју на развојној плочи Picture 6, denounced "ФДВ ВЛАСОТИНЦЕ" on the LCD display on the development board

Објашњење константе character1, character2, character3, character4, character5

const character4: array[0..7] of byte = (17,17,19,21,25,17,17,0);

Обратите пажњу на бројеве у заградама. Приметићете да има 8 броја и њихов редослед је битан. Наиме први број је за први ред знака, други број је за други ред знака, трећи број је трећи ред знака и тако до осмог реда који је осми ред знака. Вредност бројева се добија тако што се бинарни број за задати ред преведе у декадни број.

Ево како се добија број за први ред ћириличног слова "И".



Слика 7, ћиричично слово "И" Picture 7, Cyrillic letter "И"

Бинарни запис: 10001. (црни квадратић је 1 а жути квадратић је 0) Превођењем броја у декадни систем имамо. $1 \cdot 2^{5 \cdot 1} + 0 \cdot 2^{4 \cdot 1} + 0 \cdot 2^{3 \cdot 1} + 0 \cdot 2^{2 \cdot 1} + 1 \cdot 2^{1 \cdot 1} = 2^4 + 2^0 = 16 + 1 = 17$

Што и јесте први број. Поступак је исти и за остале бројеве.

"Штедљива" примена

Како је ово време у коме треба да се штеди па ево и једне "штедљиве" примене креирања произвољног знака. Наиме када бисмо правили приказ на дисплеју класичним начином онда бисмо "°С" (степен целзијус) морали да прикажемо као 2 знака. То може довести до случаја да све што је потребно да прикажемо на дисплеју не може стати. Уколико направимо произвољни знак који приказује "°С" онда бисмо у том случају могли да прикажемо све што нам је потребно јер бисмо заузели место само за један знак. Тиме смо избегли случај да купујемо дисплеј са већим бројем знакова а самим тим бисмо штедели новац јер су дисплеји са већим бројем знакова скупљи. Ево кодова за тај знак.

const character: array[0..7] of byte = (16,6,9,8,8,9,6,0); procedure CustomChar(pos_row, pos_char: Byte); var i: Byte; begin Lcd_Cmd(64); for i := 0 to 7 do Lcd_Chr_CP(character[i]); Lcd_Cmd(_LCD_RETURN_HOME); Lcd_Chr(pos_row, pos_char, 0); end;

F	F	F	F
H	=	F	
H			
H			

Слика 8, графички приказ °C Picture 8, graphical representation °C



Слика 9, Практична примена једипственог знака степен целзијус ($^{\circ}C$) Picture 9, Practical application of unique characters degrees Celsius ($^{\circ}C$)

ЗАКЉУЧАК

Овим се радом показује како је могуће креирати произвољне знакове. То може да буде искоришћено за креирање знака који одговара неком специфичном случају. Међутим могуће је креирати и једниствен знак који би био приказ за два знака како је наведено у раду. Тиме можете уштедети и новац и време за прављење неког уређаја који је контролисан са РІС колима.

ЛИТЕРАТУРА

[1.] http://www.mikroe.com/easypic/

[2.] http://www.mikroe.com/mikropascal/pic/