## A TUTORIAL FOR PIC C COMPILER AND ISIS SIMULATION

Aim of this tutorial is to show how to use PIC C COMPILER and ISIS simulator. Tutorial covers an object that writing a code to light a LED on 0th bit of PORTB. In this way, PORTB in PIC16F877A should be an output port. After showing how to open a project and how to write a CCS C code in PIC C COMPILER at part one, an ISIS simulation will be done at part two.

## **1. Introduction to PIC C COMPILER**

1) Open PIC C COMPILER. Icon of the editor can be seen in Figure-1.





2) From the "Project" choose "PIC Wizard". Click "PIC Wizard" in the Figure-2 window.



3) Open new file "CCS C TUTORIAL" and write file name "hello\_world" in the Figure-3 window and click SAVE.

🌋 Farklı Kaydet	t			×
Konum:	CCS C TUTO	RIAL ~	G 🤌 📂 🛄 🗸	
Hızlı erişim	Ad	^ Aramanızla eşleşen öğ	Değiştirme tarihi je yok.	Tür
Masaüstü				
Kitaplıklar				
Bu biigisayar Ağ				
	<			>
	Dosya adı:	hello_world	~	Kaydet
	Kayıt türü:	.PJT Project file only	~	İptal

Figure-3

Select the device. In our case choose "PIC16F877A". Select the fuses. In our case choose "High speed Osc (>4mhz)" and click "OK".

PIC Wizard		×
File		
Project Name: C:\l	Users\görkem\Desktop\CCS C TUTORIAL\hello_ <del>w</del> orld.pjt	
General - Communications SPI and LCD - Timers - PCH Timers - Analog - Other - Interrupts - Intr Uscillator Config - Header Files - CAN BUS - LCD options - MOD BUS - BOOT LOADER	Options       Code         General       Function Generation            • Opening brace on the following line        Opening brace on the same line            Device:       PIC16F877A       Oscilator Frequency:       20.000.000            Lenable Integrated Chip Debugging (ICD)        Restart WDT during calls to DELAY             Use 16 bit pointers for Full RAM use        One fuse per line with comments             Fuses           High speed Osc (> 4mhz)             Device: Up Timer           Code protected from reads             Debug mode for use with ICD             Reset when brownout detected             Low Voltage Programming on B3(PIC16) or B5(PIC18)	~
	Ok Cancel	Help

Figure-4

5) A default is opened as it seen in the **Figure-5**.

🛸 PCW	
	Project Edit Search Options Compile View Tools Debug Document UserToolbar
Project	PIC Wizard     Create     Open All Files Close Project     Find Text in     Project     Project Options
hello	p_world.c
1 2 3 4 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18	<pre>#include "C:\CCS C TUTORIAL\hello_world.h"  void main()  setup_adc_ports(NO_ANALOGS); setup_adc(ADC_OFF); setup_spi(SPI_SS_DISABLED); //setup_timer_0(RTCC_INTERNAL RTCC_DIV_1);); setup_timer_1(T1_DISABLED); setup_timer_2(T2_DISABLED,0,1); setup_comparator(NC_NC_NC_NC); setup_vref(FALSE); // TODO: USER CODE!!</pre>

Figure-5

6) Aim of the project is to light the LED on the 0<sup>th</sup> bit of the PORTB. A sample code is written in **Figure-6**.



Figure-6

7) From the "Compile" choose the "Build All".



Figure-7

8) "Build" is completed and there is no error, Figure-8.



**Figure-8** 

## 2. Simulate your assembly code in ISIS

1) Open ISIS, our simple project should be light the LED on 0th bit PORTB. In this scope, we need to add related components to the schematic project and construct the simulation circuit. In the **Figure-9**, a blank schematic page can be seen.



**Figure-9** 

2) Circuit elements can be added from the ISIS libraries. At first click "Component Mode" icon, **Figure-10** step 1, then click "P" icon, **Figure-10** step 2, to open library window.



Figure-10

**3**) Type PIC16F877A, LED-RED, RES step by step then double click for each one. **Figure-11** shows the operation.

💓 led_tutorial - Prote	us 8 Professional - Schematic Capture		- 0	$\times$
File Edit View Too	i) Design Graph Debug Library Template System Help Generation Color Color Help Color Color Nation Temperation Action Victor Color C			
Schematic Cap	THE X B PCB LANOT X Source Code X			
	The Devices		2	×
â 📥	tal Fra Derives	Diotoconta Duraleuro		^
÷ 📍	Keyworgs: Besuits (2) 19877 Desire Librery: Description	VSM DLL Model IPIC16L		
	Match Whole Words? PICIAE677 PICAECR0 PICIE Microscontroller (8kB code, 3688 deta, 2568 EPROM, Parts A-E, 2xCCP, PSP, 3xTimets, MSSP, USART, 8x10-bit ADC)	14 OSC1/CL/4N 14 OSC2/CL/4DUT	RBO/INT 30 RB1 34	
H LED-RED	Show only parts with models? DECE66977A PICMCR0_PICE6Microscontroller (6kB code, 368B date, 256B EPROM_Purts A-E_2ACMP_2ACOP_PSP_3/Timers_MSSP_USART_B/10-bit ADC)	2 3 RAD/AND 3 RA1/ANI	RB2 36 RB3/POM 37 RB4 56	
1 RES	_energions	4 RA2/AN2/VREF/C 5 RA3/AN3/VREF+ 9 RA4/DDF/M110//	REF R95 29 R95/F9C 40	
<b>□</b> ₽	Microprocessor ICs	RA5/AN4/SS/20U	CO/T10S07104 15	
¥¥		9 RE1/AND/WR 10 RE2/AN7/CS	RC2/CCP1 17 RC2/CCP1 18 RC3/SCK/SCL 22	
9		MCLR/Vpp/THV	RC6/SD0 25 RC6/TX/CK 26	
<u>^</u>			RC7/RX0T R00/PSP0 19	
			RD1/PSP1 21 RD2/PSP2 22 RD3/PSP3 22	
			RD4/PSP4 20 RD5/PSP6 20 RD5/PSP6 29	
			R07/PSP7 30	
00		PCB Preview:		
A S		0.6	in	
+			<b>⊢</b> +≞	
C			40	
2				
+	She-category.			
\$		1.9i		
	Monufacturer:			
		DIL40		~
		ОК	Cancel	
	0 No Mersoon Root sheet 1			

Figure-11

**4)** As seen in the **Figure-12**, circuit for the simulation of assembly code in the previous part is constructed. GND object has been taken from "Terminals Mode" part. It is important to mention that, OSC pins and MCLR pin are left floating. In the simulation, there is no need to add them however, they should be also used while constructing on breadboard or PCB.

13     OSC1/CLKIN     RB0/INT     33     INT       14     OSC2/CLKOUT     RB1     35     220       2     RA0/AN0     RB3/PGM     37       3     RA1/AN1     RB4     35       4     RA1/AN1     RB4       5     RA3/AN3/VREF-/CVREF     RB5       5     RA3/AN3/VREF+/CVREF     RB6       6     RA4/TOCKI/C10UT     RB7/PGD       7     RA6/AN4/S5/C2OUT     RC0/T10S0/T1CKI       8     RE0/AN5/RD     RC1/T10SIVCCP2       9     RE1/AN6/WR     RC2/CCP1       1     RCLR/NFR     RC2/CP1       1     RCLR/NFR     RC2/CP2       20     25       20     26       20     27       21     20       22		U1		R1	
20	13 14 2 3 4 5 6 7 7 7 8 9 10 10	OT OSC1/CLKIN OSC2/CLKOUT RA0/AN0 R RA1/AN1 RA2/AN2/VREF-/CVREF RA3/AN3/VREF+ F RA4/T0CK/IC1OUT F RA5/AN4/SS/C2OUT RE0/AN5/RD RC1/T1O RE0/AN5/RD RC1/T1O RE0/AN5/RD RC1/T1O RE1/AN6/WR R RE2/AN7/CS RC3/ MCLR/Vpp/THV F RC4/ MCLR/Vpp/THV F RC4 RC4 RC4 RC4 RC4 RC4 RC4 RC4	RB0/INT         33           RB1         34           RB2         35           RB4         37           RB4         37           RB4         38           RB6/PGC         40           RB7/PGD         40           IO/T1CKI         15           IS0/CCP2         17           SCX/SCL         18           SOU/SCL         18           SOU/SCL         23           SOU/SCL         25           SOT/RX/DT         20           D0/PSP0         19           D1/PSP1         21           D2/PSP2         22           D3/PSP3         27           D4/PSP4         28	R1 220	D1 LED-RED
RD6/PSP8 20 PD7/PSP7 30		RI	D6/PSP6 29 07/PSP7 30		

Figure-12

**5**)Circuit is constructed, now we need to load our ".hex" file to the PIC and run the simulation. By double clicking on the PIC16F84A IC on ISIS, a new window will be opened. From this window, choose your ".hex" file from the directory that you have created in part one. In Figure-13, ".hex" file is added and clock frequency of the IC is set to 20MHz.After setting these, press "OK" button.

Edit Component			? ×
Part <u>R</u> eference: Part <u>V</u> alue: <u>E</u> lement:	U1 PIC16F877A	Hidden: 🗌 Hidden: 🗌	OK Help
PCB Package: Program File: Processor Clock Frequency: Program Configuration Word: Advanced Properties: Pendomico Program Momon /2	DIL40   CCS C TUTORIAL\hello_world.cof  20MHz  0x3FFB	Hide All Hide All Hide All Hide All Hide All V	Hidden Pins Edit Firmware Cancel
Other Properties:	Attach hierarchy module Hide common pins Edit all properties as text		

Figure-13

6) Program is set up on IC. Final step is to run the simulation. Click play button and see the result, **Figure-14**.



Figure-14

Prepared by A.ÇağrıArlı