

Leader of Microcontroller Technology Global Top Smart MCU Innovation Company

CodeGen8 (8-Bit MCU Code Generator)

User Manual

Version 1.0.0 Jun. 28, 2019

For additional information or inquiry, please contact ABOV Semiconductor or visit its website at <u>www.abov.co.kr</u>.



Revision History

Date	Version	Description
Jun. 28, 2019	1.0.0	Document created.



Contents

Chapter 1.	Getting started
	1.1. System requirements 4 1.1.1. Software requirements 4
	1.1.2. Hardware requirements
	1.2. Software installation
Chapter 2.	Structure of CodeGen8 folders
	2.1. Structure of CodeGen8 project folder
	2.2. Structure of CodeGen8 library folder
Chapter 3.	Using CodeGen8
	3.1. Features of CodeGen8
	3.1.1. Detailed features
	3.1.2. Getting started with CodeGen8
	3.2. Menu descriptions 16 3.2.1. File 16 3.2.2. View 20 3.2.3. Help 26
	3.2.4. Toolbar
	3.3. Control panes. 26 3.3.1. File View pane 27 3.3.2. Properties pane. 28 3.3.3. Output pane 30
	3.4. Child windows
	3.4.1. Source program view. 32 3.4.2. Package view. 33
Chapter 4.	Output files
	4.1. Header files. 35 4.1.1. Device header 35 4.1.2. func_def.h 36
	4.2. Source files. 37 4.2.1. startup.a51 37 4.2.2. main.c 38 4.2.3. interrupt.c. 40 4.2.4. peripheral.c. 41



Chapter 1. Getting started

1.1 System requirements

This section describes the hardware and software system requirements.

1.1.1 Software requirements

This software is compatible with the following operating systems. Both 32-bit and 64-bit versions are available.

- Microsoft Windows NT
- Microsoft Windows 2000
- Microsoft Windows XP
- Microsoft Windows Vista
- Microsoft Windows 7
- Microsoft Windows 8 & 8.1
- Microsoft Windows 10

Disk space

For the full installation of the software, up to 20 MB of hard disk space is required.

1.1.2 Hardware requirements

This software can be run on a basic PC and does not require powerful specifications. The following are minimum hardware requirements for installing and running this software.

Pentium PC

Performance is based on the following factor:

Processor performance

1.2 Software installation

The installer can be downloaded from the ABOV website. You should ideally have the latest version of the software because ABOV constantly adds newer devices and features. The installer package can be executed on any Microsoft operating system.

Perform the following:

1. Execute the installer.



2. Once the warning message below pops up, click **Yes**.

😯 Use	r Account Control	
۲	Do you want unknown put	to allow the following program from an blisher to make changes to this computer?
	Program name: Publisher: File origin:	Unknown Hard drive on this computer
🕑 s	how details	Yes No
		Change when thes potifications appear

3. You will see the following dialog box. Click Next.





4. Select I accept the items of the license agreement on the License agreement dialog box and click Next.



5. Enter the username and company name and click **Next**.





6. Select **Complete** and click **Next**.

SetupUSB_Dr - InstallShi	eld Wizard 🛛 🕅
Setup Type Select the setup type to install.	
	Please select a setup type. Complete Image: Com
InstallShield	< Back Next > Cancel

7. Click Install.





8. Wait until all the application components are installed.



9. Click **Finish** to complete the installation.





Chapter 2. Structure of CodeGen8 folders

2.1 Structure of CodeGen8 project folder

CodeGen8 includes CodeGen8 project files in the Project folder, which is located in the CodeGen8 installation folder. All CodeGen8 project files are either $\ \ vpp$ or $\ v$. VPD files:

• *. VPP files contain basic information (device name, package type, pin count, etc); for example,

```
ABOV-CodeGen8-MC9x A1.03 20130419
MC96F6332 SOP 28
```

• *. VPD files contain the target device's peripheral property settings; for example,

```
B_isSingle 0
PORT001 -1 0 0 0 0
BUZ 0 2659.574463
:
:
EXTINT10 0 0 0
WDT 0 0 120
```

```
Caution: Do not modify \*.VPP or \*.VPD files.
```

Folder assignment:



2.2 Structure of CodeGen8 library folder

CodeGen8 includes each device's sample files in the device's specific library folder under the common Library folder, which is located in the CodeGen8 installation folder. These device-specific library folders contain the following:

- Device package definition file
- Device header file
- Sample source program files



CodeGen8 generates a standard basic source program. To see details, refer to the sample source files.

Caution: Do not delete or modify any of the files in the Library folders.

Folder assignment:





Chapter 3. Using CodeGen8

3.1 Features of CodeGen8

CodeGen8 supports the M8051-based MC9x/A9x series developed by ABOV Semiconductor Co., Ltd.

ABOV MC9x series code generat	or				
MC9656332:28 SOR (project)	fila = ddd \/DD \				Ontion
File View	MC05F6432 h	-	Parkana		Properties
					Device property
	2 // device : MC96F6432(44 pir	n)	Device : MC96F6332, Package : 28SOP		2⊨ 4↓ □ 4
🖨 🥁 Source Files	3 // version : Verl.0 4 // date : 2011-03-30				System clock
	5 // compiler : Keil-C	E	vee L		I/O ports
	6 // author : Ma, Nakwon		XOTT / P50 a 27 P00	803	😑 P0:
	7 //		XTN/P51 2 25 P01	730	#27 : P00/EC3
	<pre>9 #include <intrins.h></intrins.h></pre>		PT NF (PC0 / PTNT9 / D52 4	an0/aup =	
Header Files	10		PERMO / TOO / SYTN / P53 5	SPG26/A	#25 : P02/AN0/AVREF/EINT0/T4O/P
Tune def h	11 #ifndef _MC96F6432_		PINTO / SVIT / 25 4	SEC25/A	#24 : P03/SEG26/AN1/EINT1/PWM4AB
Charry Otherr	12 #define _MC96F6432_		DECEMPTO / DEC 2 005	SEG23/ R	#23 : P04/SEG25/AN2/EINT2/PWM4BA
D Packane	14 sfr P0 = 0x80;	// PO Data E	MTSO0/SCT.0/Pwp0/TC2/P40 4	SEG21/A	#22 : P05/SEG24/AN3/EINT3/PWM4BB
Package	15 sbit P00 = 0x80;		MISCO/SCLO/REDO/LCS/P40 P	SEG23/M	#21 : P06/SEG23/AN4/EINT4/PWM4CA
	16 sbit P01 = 0x81;		NOSTO/ SDRO/ 1RD0/ DC2/ PHT P	CDC1C/A	#20 : P07/SEG22/AN5/EINT5/PWM4CB
	17 SDit P02 = 0x82; 18 shit P03 = 0x83;		SCR0/V1C1/P42 P P12/	SEG16/A	
	19 sbit P04 = 0x84;		SEG2/COM4/P33 P11/	SEGID/A	
	20 sbit P05 = 0x85;		3E03/CON3/P32 P	SEG14/A	
	21 sbit P06 = 0x86;		SEG4/COM6/P31 P20/	SEGIS/A	
	22 SDIC PU/ = UX8/;		3E63/COR//E30 [4] [21/	SEGIZ/A	
	24 sfr P0IO = 0xA1;	// PO Direct			A/D convertor
	25 sfr POOD = 0x91;	// PO Open-c			Basic interval timer
	26 sfr POPU = 0xAC;	// PO Pull-t			Buzzer
	27 SIT PODB = 0xDL; 28	// Po Debour			Ext. interrupt
	29 sfr P1 = 0x88;	// P1 Data F			I2C interface
	30 sbit P10 = 0x88;				LCD driver
	31 sbit P11 = 0x89;				SPI interface
	32 SDIT P12 = UX8A; 33 Sbit P13 = 0x8B;				Timer/Capture/PWM
	34 sbit P14 = 0x8C;				Timer0
	35 sbit P15 = 0x8D;				Timer1
	36 sbit P16 = 0x8E;				Use or not Not use
	38 SDIC FI/ = 0x0F;				Interrupt disable
	39 sfr P1IO = 0xB1;	// P1 Direct			Function Timer
	40 sfr P10D = 0x92;	// P1 Open-c			input source Clock
	41 SIT PIPU = 0xAD; 42 sfr P15DB = 0xDF.	// P1 Pull-t		-	Period (ms) 1.000000
	<	>> 11/20 Det	< III	•	Frequency (Hz) 1000.000000
					Timer2
	Output				Timer3
	// No warning message found				
					System clock
1	H 4 P H Warning N.C /				-
Ready					CAP NUM SCRL

CodeGen8 screenshot

3.1.1 Detailed features

This software application aims at helping both beginners and veterans dramatically save time spent on 8-bit MCU programming. It provides the following functions:

- Generates C source files for the target device simultaneously when the user sets a peripheral.
- Supports the KEIL project format by:
 - Generating $\ \ v2$ project files. They support KEIL uVision 4 and 5.
 - Generating each device's header file (e.g. MC95FG308.h).
 - Generating the startup.a51 assembly code file.
 - Generating a single source file (main.c) or multiple source files (main.c, interrupt.c, and peripheral.c).



- Displays the following:
 - Device header file(s)
 - Device source file(s)
 - Package view
- Provides a peripheral setting pane.
- Provides a package view that shows pin assignment and allows the user to set port functions.
- Manages CodeGen8 project files automatically.
- Generated source file contents are changed as soon as you click for device setting changes.

3.1.2 Getting started with CodeGen8

Perform the following:

1. Execute CodeGen8 to open the following dialog box; among the controls, only **device name**, **package type**, and **Cancel** will be active.

ound
e MC95FG208 ▼ e 16 SOP ▼ e



2. Select a target device.

ABOV basic C source generator : Project selec	tion Dialog
LAST project	No project file found
NEW project	device name MC95FG208 ▼ MC95FG208 package type MC95FG308 MC96FG332 project name MC96F6432
Load a project	
Cancel	date

3. Select a device package.

ABOV basic C source generator : Project sele	No project file fou	nd
NEW project	device name package type project name	MC96F6332 28 SOP 28 SOP 32 QFP 32 SOP 32 SOP
Load a project		
Cancel	date	



4. Type the project name, and the **New project** button will be enabled.

LAST project	No project file found
NEW project	device name MC96F6332 package type 32 QFP project name myNewProject
Load a project	
A1.03 (2013. 4.19)	
Cancel	date

5. Click on **New project**.



6. If you have created more than one CodeGen8 project, the dialog box will list them as shown below:

LAST project	ddd.VPP MC96F6332 - 28 SOP
NEW project	device name MC95FG208 package type 16 SOP project name
Load a project	ddd.vpp 6332_28sop.VPP 6432_44.VPP 308_32QFN.VPP 308_32QFN.VPP
	308_32sop.VPP 208_20TS.VPP 308_32LQ.VPP 208_16.VPP
Cancel	MC96F6332 - 28 SOP

- LAST project: Load the last project.
- NEW project: Create a new project.
- Load a project: Load a project from the list of previous projects.





7. On the main screen below, configure your device.

Ent your gep N R G C C C T	menu		Cellers
FirVing T & X	MORECVED antichard a w	Parkane v mint	Posteries
Source structure	<pre>/// recipionsi recipionsized /// recipionsi recipionsized // recipionsi recipionsized // recipionsized</pre>	Device : MC95F630S, Package : 32QFN www.ucurumwooodura www.ucuru	Beliese property Image: Second Se

3.2 Menu descriptions

3.2.1 File

The File menu includes the following items:





New or Load Project

This menu item closes the currently open CodeGen8 project and prompts the user to create a new project or load a project from the list of previous projects.

LAST protect	ddd.VPP
LAST project	MC96F6332 - 28 SOP
	device name MC95FG208
NEW project	package type 16 SOP
	project name
Load a project	ddd.VPP 6332_28sop.VPP 6432_44.VPP
	308_32QFN.VPP 308_28TSSOP.VPP 308_32sop.VPP 208_20TS.VPP 308_32LQ.VPP 208_16.VPP
A1.03 (2013, 4,19)	

Multi -> Single source / Single -> Multi source

Single-source project:

CodeGen8 generates a single source file named main.c. This file contains the main function routine, interrupt vector functions, and peripheral functions. It is automatically closed when a multisource file (e.g. interrupt.c) is opened.

Multisource project:

CodeGen8 generates main.c, interrupt.c, and peripheral.c source files.

- The main.c file contains the main function routine only.
- The interrupt.c file contains interrupt vector functions.
- The peripheral.c file contains each peripheral's initializing functions.

You can see the file structure in the File View pane.



Generate source file(s)

CodeGen8 saves the following files:

- Header files
 - Device header file, for example, MC95FG308.h
 - Function definition file, for example, func_def.h
- Source files
 - Assembly file that includes the initialization code, for example, startup.a51
 - C source program file, for example, main.c, interrupt.c, or peripheral.c
- KEIL project files
 - It contains the target device, file management, etc., for example, projectName.uv2.

If there is a warning appearing because of errors in the device settings, it prompts to choose whether to continue or not. You can see the warning in the **Output** pane.

MC9x_CGen	X
There are 2 warnings that you have to correc Continue anyway ? (Y/	ct. 'N)
Yes(Y) No	(N)



Call KEIL compiler

This menu item saves the current header file(s), source file(s), and KEIL project file, and calls the KEIL compiler as shown below:

🕎 ddd - ≩lision3 - [C:\Temp\n	ain.c]	
Eile Edit ⊻iew Project De	bug Fl <u>a</u> sh Pe <u>r</u> ipherals <u>T</u> ools <u>S</u> VCS <u>W</u> indow <u>H</u> elp	_ 8 ×
🎦 😂 🖬 🎒 🐰 🛍 🛍	으 으 孝 孝 ゟ % % % % <mark>@ccl_read_sub • 碘 种 ← → 蚀 음 @, [国国 신 송 </mark>	y 🚛
🕸 🏝 🎬 👗 🛱 🌾 Tarı	et 1 🔹 🛃 🚍	
Project Workspace 👻 🗙	42 // internal RC clock (1.000000MHz)	
🖃 📲 Target 1	43 // Nothing to do for the default clock	_
🗄 📇 Source Group 1	44 }	
startup.a51	45	
🖃 🔛 main.c	46 void port_init()	
🛄 mc96f6432		
intrins.h	48 // initialize ports	
	43 FOLO = OXFF; // direction	
	51 POOD = 0x00. // puttip	
	52 PODB = 0x00: // open dram	
	53 P0 = 0x00; // port intial value	
	54	
	55 P1IC = 0xFF; // direction	
	56 P1PU = 0x00; // pullup	
	57 P10D = 0x00; // open drain	
	58 P15DB = 0x00; // debounce : P54, 52, 17, 16, 12, 11	
	59 P1 = 0x00; // port intial value	
	60	
	61 P2IC = 0xFF; // direction	
	$\begin{array}{cccc} 62 & P2PU = 0 \times 00; & // pullup \\ \end{array}$	
	$p_{3} = p_{2} = p_{3} = p_{3$	-
→ III →		•
	🖹 main,c	
×		▲
op		
Mir.		
	<u> </u>	
Build Comman	d 👌 Find in Files /	•
For Help, press F1	Simulation L:1 C:1	NUN //

If there is a warning appearing because of errors in the device settings, it prompts to choose whether to continue or not. You can see the warning in the **Output** pane.

Print...

It prints the current child window. The child windows include the view of each source code (header, C) and a package view.



Print Preview

This feature allows you to see exactly how your pages will look when they are printed out. Using this function, you can save paper right up until final printing. **Print Preview** is easy to use, and you can print directly from this screen by clicking on the **Print** button or printer icon.

<u> </u>	1	//	×		
	2	11	device	: MC96F6432(44 pin	u) Print P 4 🛄 🔍 🔍 Close
	3	11	version	: Ver1.0	
	4	11	date	: 2011-03-30	
	5	11	compiler	: Keil-C	
	6	11	author	: Ma, Nakwon	1.0
	7	//=====			2 // davias · (20070.02(% pin) 2 // vandas · (Vad.0 4 // data · 1202.4740
	8				2.17 samples : Fall-C 2.17 sollar : No, Bitron 7.29
	9	#include	<pre><intrins.< pre=""></intrins.<></pre>	h>	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
	10				11 (Calor Jonana) 13 (Calor Jonana) 13
	11	#ifndef	_MC96F	6432_	14 cb 80 - 540 // 70 bbb Xepidere 13 cb 20 80 - 540 / 14 cb 20 80 - 540 /
	12	#define	_MC96F	6432_	17 402 802 - 043) 18 402 802 - 043) 18 402 802 - 044)
	13				10 402 102 - 045 11 402 105 - 045 13 402 107 - 045
	14	sfr	PO	= 0x80;	// 23 34 dor 2522 - 544/c // PD Disedim Refere 32 do 2522 - 545/c // PD Disedim Refere
	15	sbit	5 P00	= 0x80;	11 da 1507 - 040 ///12 hill up fail de Indelan Balle - Baller 17 da 1503 - 040 //12 hill up fail de Indelan Balle - Baller
	16	sbit	5 P01	= 0x81;	31 db E - 543 //71 bds Tagidae 20 db Z EO - 543 427 EO - 545
	17	sbit	E P02	= 0x82;	
			Norma	al view	Print Preview

Print Setup...

This command displays the standard **Print Setup** dialog. You can choose a printer and select other options.

Exit

This button helps quit from CodeGen8 immediately.

3.2.2 View

The View menu controls the display of the CodeGen8 software frame and child windows.



Add vertical TAB group

It provides the functionality for tab control. The tab control displays a dockable window with flat or threedimensional tabs at its top or bottom. The tabs can display text and images and can change colors when active.



Toolbars and Dockable Windows

This menu item shows or hides different child views.

View	v Help		
	Add vertical TAB group		
	Toolbars and Docking Windows	~	Standard
~	Status Bar	~	Properties
~	aption Bar	~	File View
	Application Look	~	Output
			Customize

Customize

It allows users to change the debugger software GUI environment to suit their preferences regarding **Commands**, **Toolbars**, **Keyboard**, **Menu**, and **Options**.

The **Commands** tab provides choices to modify the composition of each menu's sub-items.

Customize	×
Commands Toolbars Keyboard	Menu Options
Categories: File View Help New Menu All Commands	Comman <u>d</u> s: N New or Load Project Single/Multi files Generate source file(s) C Call KEIL compiler Print, Print Preview Print Setup,
Description:	 Close





The **Toolbars** tab allows you to switch between toolbar styles.

Customize	×
Commands Toolbars Keyboard Menu Op	tions
Toolbars:	
Menu Bar	<u>H</u> eset
	Reset <u>A</u> ll
	<u>N</u> ew
	Rena <u>m</u> e,
	<u>D</u> elete
	Show text labels
	Close

The **Keyboard** tab allows you to define shortcut keys. You can reset them or restore them to the default settings.

Customize		×
Commands Toolbars Key	board Menu Options	
<u>C</u> ategory: File ▼ C <u>o</u> mmands: Call KEIL compiler Exit Concentrate courses file(o	Set Accelerator <u>f</u> or: Default C <u>u</u> rrent Keys:	<u>A</u> ssign
Print Preview Print Setup	Press <u>N</u> ew Shortcut Key:	<u>R</u> emove Re <u>s</u> et All
Description: Call external KEIL C compiler		
		Close



The Menu tab is to define the menu style.

Customize	
Application Frame Menus: Show Menus for: Default Menu	Context Menus: Select <u>c</u> ontext menu:
Default application menu, Appears when no documents are open,	Hint: select the context menu, change the page to 'Commands' and drag the toolbar buttons into the menu window,
<u>M</u> enu animations: <u>None</u> ▼ ✓ Menu s <u>h</u> adows	
	Close

The **Options** tab provides settings regarding toolbar tip display, icon size, etc.

Customize	x
Commands Toolbars Keyboard Menu Options	
Toolbar	
Show ScreenTips on toolbars	
Show shortcut <u>k</u> eys in ScreenTips	
Large Icons	
Personalized Menus and	
Menus show recently used commands first	
Show full menus after a short delav	
<u>R</u> eset my usage data	
Clos	e]



Status Bar

It turns on or off the Status bar, which displays information on the current state of CodeGen8.

	I	•	۲	H	74
Ready					

Caption Bar

It turns on or off the Caption bar, which displays the device name, package type, and project name.

[:] N	E		С		@ ₌							
i	MC96	F633	2 : 28	SOP	(project	t file =	ddd	.VPP)				
File V	iew				– п ×		M	06F64	37 h	v)		



Application Look

It changes CodeGen8's GUI look at once.



GUI selection



Different GUI styles



3.2.3 Help

It shows the CodeGen8 version and copyright, and the programmer's name.

Help							
About N	MC9x_CGen						
About MC9	0x_CGen	\times					
Code Generator V1.058.00 C Copyright(C) 2018.12.19							
	Programmed by Seungduk Ha						
	OK						

3.2.4 Toolbar

This provides an easy and single-click access to the most frequently used commands.



3.3 Control panes

CodeGen8 provides three control panes for developers. The information displayed on those panes include project status, device settings, and warnings. All panes except the child windows are dockable.

What is docking

Docking is manipulating a window to align it with the edge of another window or to move it into another window.



e Xiem Helb					
MC96P6332:28 SOP (project)	file = ddd.VPP)				Options
View • # X	MC96P6432.h x	Package x	•	Properties	
	1 //	A	*	Device property	
😋 ddd < Multi>	2 // device : MC96F6432(44 pir	Device : MC96F6332,	Package : 28SOF	20 24 (3) 5	
😓 🥁 Source Files	3 // Version I Verl.0			System clock	
startup.a51	5 // compiler : Weil-C			Clock source	External
- 🕤 main.c	6 // author : Ma, Nakwon	V35		Ext. Freq. (MHz)	8,000000
- interrupt.c	7 //	xout/950 P	- 밑테	(i) 1/0 ports	
peripheral.c		XIN/951	[[]	A/D convertor	
🔆 🦢 Header Files	<pre>9 #include cintring.hb</pre>	BLAK/ECO/EINT8/P52	[] [] [] [] [] [] [] [] [] [] [] [] [] [(i) Radic Internal times	
— MC96F6432.h	11 Alfodef MC96F6432	PMM00/T00/SXIN/P53		to basic interval timer	
func_def.h	12 #define MC96F6432	EINT10/SXOUT/P54		🗄 Buzzer	AL
🖯 🇁 Others	13	RESETB/P55] Dr	Use or not	Not use
Package	14 sfr PO = 0x80;	// MISOO/SCL0/RxD0/LC3/P40		Frequency (Hz)	61.035156
	15 SD15 P00 = 0x803	MOSI0/SDA0/Tx00/1C2/P41		Ext. interrupt	
	10 8010 PV1 = 0001; 17 shir 200 = 0x01;	SCK0/VLC1/P42		I2C interface	Property
	10 sbit P03 = 0x13;	5E52/COM4/211	1 5	LCD driver	roperty
File View	19 sbit P04 = 0x84;	ench/ower/pth	H H H	B SPI Interface	
The view	20 sbit 905 = 0x85;	ence (court inth in		Imer/Capture/PWM	pane
	21 sbit 906 = Cx86;	SEGN/COMP/PS1E	80	I Timer0	1 C C C C C C C C C C C C C C C C C C C
pane	22 sbit P07 = 0x87;	SE05/COR7/P30 [5		Timer1	
	24 efr 2010 = 0x31+	11		I Timer2	
	25 sfr 9000 = 0x91;	<i>ïi</i>		(i) Timer3	
	26 sfr 9090 = 0xAC;	11		C Thread	
	27 sfr 9008 = 0x0E;	11		C. Fangers	A1-1
	28			Use or not	Pilot use
	29 SIF F1 = Cx00;	//		interrupt	disable
	31 sbit P11 = Cwt9;			Function	Timer
	32 sbit \$12 = 0xtA;			Period (ms)	1.000000
	33 sbit P13 = Cw18;			Frequency (Hz)	1000.000000
	34 sbit P14 = Cx8C;			UART interface	
	30 8015 P15 = 00803	*	*	Watch-dog timer	
	4	> <		Watch timer	
				Use or not	Not use
	Output		¥ a ×	Interrupt	disable
	> OSC : You have to select XIN port	_		Period	Fwck/128
	// Warning count = 2	Output pane		Clock source	enal
	and the second sec			one on another no or cold	

3.3.1 File View pane

It shows the file components of the current project. Single-source projects generate main.c, while multisource projects generate another two C files (interrupt.c and peripheral.c). The rest files (startup.a51, func_def.h, and device header) are common. You can open any file by double-clicking on the file name.





3.3.2 Properties pane

It shows the peripherals of the current project. This is the most important component of CodeGen8.

- This lists specifications of each peripheral.
- The peripheral settings are viewable and editable.
- If you change any of the settings, the source program is generated automatically.
- Some peripherals support only one function; an attempt to enable multiple functions pops up a warning message.
- If a peripheral setting conflicts with other peripherals, the **Output** pane displays it.



Properties	▼ Ф	x		
Device property		-		
₽ <u>2</u> ↓				
System clock				
Clock source	External			
Ext. Freq. (MHz)	8.000000			
A/D convertor				
Basic interval timer				
Buzzer				
Use or not	Not use			
Frequency (Hz)	61.035156			
🗄 Ext. interrupt				
I2C interface				
LCD driver				
SPI interface				
Timer/Capture/PWM		Ξ		
Timer0				
Timer1				
Timer2				
Timer?				

Case 1

Suppose that the system clock is 4 MHz, and you want to use UART at 9600 bps. But a baud rate of exactly 9600 bps cannot be rendered at 4 MHz. CodeGen8 calculates and displays the nearest bps and generates the source program.

UA	RT interface	
	UART0	
	Use or not	Use
	Rx interrupt	disable
	Tx interrupt	disable
	Operation	Async. UART
	Baudrate	9600
	Parity	Nothing
	Data bit count	8
	Stop bit count	1
+	UART1	

Your input: 9600 bps

-	UA	RT interface	
	-	UART0	
		Use or not	Use
		Rx interrupt	disable
		Tx interrupt	disable
		Operation	Async. UART
		Baudrate	9615.384766
		Parity	Nothing
		Data bit count	8
		Stop bit count	1
	+	UART1	

CodeGen8 calculation: 9615.38 bps

Case 2

Suppose that you are using MC95FG308, whose Timer0 and Timer1 are 8-bit timers. If you want to use Timer0 as a 16-bit timer, Timer1 must be extended to the upper 8-bit timer of Timer0. This means, if Timer0 is set to 16 bits, you do not need to set Timer1. CodeGen8 disables editing of Timer1.



🗆 Tir	ner/Capture/PWM			🗆 Tir	ner/Capture/PWM		
=	Timer0			-	Timer0		
	Use or not	Use 🔹	c		Use or not	Use	
	Interrupt	disable			Interrupt	disable	
	Function	Timer			Function	Timer	
	bit count	8bit			bit count	16bit ·	•
	input source	Clock			input source	Clock	
	Period (ms)	1.000000			Period (ms)	1.000000	
	Frequency (Hz)	1000.000000			Frequency (Hz)	1000.000000	
=	Timer1			-	Timer1		
	Use or not	Use			Use or not	Not use	
	Interrupt	enable			Interrupt	enable	
	Function	PWM			Function	PWM	
	input source	Clock			input source	Clock	
	Period (ms)	1.000000			Period (ms)	1.000000	
	Frequency (Hz)	1000.000000			Frequency (Hz)	1000.000000	
	PWM polarity	Low first			PWM polarity	Low first	
	PWM duty A (%)	66.000000			PWM duty A (%)	66.000000	
	PWM duty B (%)	33.299999			PWM duty B (%)	33.299999	
	PWM duty C (%)	70.000000			PWM duty C (%)	70.000000	

Timer0 and Timer1 are separated

Timer0 and Timer1 are merged

You do not need to know all the peripheral specifications because it is straightforward to set all the peripherals. If you want to make more detailed settings, then you need to understand all the peripheral specifications.

3.3.3 Output pane

It shows warning messages regarding peripheral settings or conflicts between them.

Output	•	ņ	×
> OSC : You have to select XIN port > OSC : You have to select XOUT port			
// Warning count = 2			
K () N \Warning \N,C \			

You must clear the warning messages by changing the device's peripheral settings. Else, it generates a C source program omitting some peripheral settings.



3.4 Child windows

The child windows operate differently from the previously mentioned panes. They include a source file view, virtual source program view, and device package view. They all are simpler views and do not provide editing functions.

N 🛅 🖨 C 🛛 🕀 🔍 📜				
ile Yiew Help				
MC96P6332:28 SOP (proje	1 file = ddd.VPP)			Options
e View 💌 🕷 🗙	MC96/6432.h x	Package x	Properties	-
	1 //	*	Device property	
🔄 ddd <multi></multi>	2 // device : MC96F6432(44 pin)	Device : MC96F6332, Package : 28S0F	20 24 (3) 5	
Source Files	4 // date : 2011-03-30		System clock	
startup.a51	5 // compiler : Keil-C	1944 1 1944	Clock source	External
- 🖸 main.c	6 // author : Ma, Sakwon	Val C	Ext. Freq. (MHz)	8.000000
- C interrupt.c	? //		I I/O ports	
peripheral.c	Alaciada cintrina ba		A/D convertor	
🕀 🤐 Header Files	10	BLOK/ECU/EINTO/PS2	Basic interval timer	
- MC96F6432.h	11 #1fndef _MC96F6432_	P98600/T00/5013/P53	Burrer	
func_def.h	12 #define _HC96F6432_	EIST10/3000T/P54	Use or not	Not use
Others	13 14 sfr 20 s (m10)	RESETB/PS5 C	Energy (Hr)	61.035156
Package	15 shit P00 = 0x807	M1900/SCL0/RxD0/LC3/P40	Ext Internet	41477174
	16 _ sbit P01 = 0x81;	MOSIO/SDA0/TxD0/LC2/P41 P	© DC interface	
	17 204LLG6 - 0x12:	SCK0/VLC1/P42	 ICD delver 	
	10 #D1C P03 = 0x83;	5852/C084/P33 14 14 P	© CEU anver	
	View	5853/cox6/932 🕘 🔛 P	0 SPI Internace	
	21 abit 906 = 0x16;	SEG4/COM6/P31 🗃 🔛 P:	U Timet/Capture/PWM	
	22 WINCOW 6x87	SE05/COM7/930 🖛 💷 9:	Timer0	
	23 WINGOW		18 Timer1	
	24 sfr P010 = 0xA1; //		III Timer2	
	25 SIT PUOD = UX913 //	Package	it Timer3	
	27 sfr P008 = 0x0E; //	rackage	Immer4	
	28	view	Use or not	Not use
	29 sfr P1 = 0x88; //	VIEW	Interrupt	disable
	30 SD11 P10 = Cx88;	window	Function	Timer
	32 mbit P12 = 0m12;	window	Period (ms)	1.000000
	33 abit P13 = Cx18;		Frequency (Hz)	1000.000000
	34 sbit P14 = 0x8C;		UART interface	
	35 sbit P15 = 0x8D;		Watch-dog timer	
		x >	Watch timer	
			Use or not	Not use
	Output	¥ 8 ×	Interrupt	disable
	 OSC : You have to select XIN port OSC : You have to select XOUT port 		Period	Fwck/128
	// Warning count = 2		Clock source	
			Une of anternal KL or Extern	Autor Contraction of



3.4.1 Source program view

It shows text files or virtually generated source code files with line numbers.

```
MC95FG308.h
              peripheral.c x
                                                Ŧ
32
    void UART init()
                                                Å.
33 {
        // initialize UART interface
34
35
        // UARTO : ASync. 9615bps N 8 1
       UCTRL2 = 0x02; // activate UART0
36
                           // Async/Sync, bit
        UCTRL1 = 0x06;
37
        UCTRL2 |= 0xA4; // interrupt, spee
//UCTRL2 |= 0x10; // enable line whe
38
        UCTRL2 |= 0xA4;
       //UCTRL2 |= 0x00; // stop b=:
0x33; // baud rate
0x33; // baud rate
39
40
41
42
        IEN1 |= 0x01;
                            // enable UART int
43 }
44
45
   void clock init()
46
    {
        // internal RC clock (4.000000MHz)
47
48
        cli();
49
       IEN3 |= 0x10;
                            // Enable BIT inte
                                                Ξ
50
      sei();
51
52
      BCCR = 0x05;
                            // 16msec BIT
       SCCR = 0xA0;
                            // Internal clock
53
54
      PCON = 0x03;
                            // STOP1 mode entr
        _nop_();
55
56
       _nop_();
57
        _nop_();
        _nop_();
58
59
         nop_();
        IEN3 &= ~(0x10); // Disable BIT int
60
61 }
62
63 void port_init()
64 {
65
        // initialize ports
        // 6:P10 i
66
        P0T0 = 0x00:
                             // direction
67
      III
                                              Þ
```

It does not support file editing but allows the user to copy the text to the clipboard. Tab size displayed in the view is fixed at 4. C or assembly keywords and comments are colored for an enhanced visibility.





3.4.2 Package view

Package x • P05/AN5/AG P04/AN4/AC P03/RxD0/M P02/TXD0/M ۸ P33/INT7 P32/INT6 P31/INT5 P30/INT4 32 31 30 29 28 27 26 25 ACOUT/AN6/T00/SCL/P06 24 23] 22] 21] 20] 19] 18] AN7/EC0/SDA/P07 2 AN14/(SCL)/P25 3 Ξ (SDA) / P26 4 VDD 5 PWM1AA/T10/INT0/P10 PWM1AB/INT1/P11 7 17 PWM1BA/BUZ/INT2/P12 9 10 11 12 13 14 15 16 PWM1BB/P13 PWM1CA/P14 AN8/PWM1CB/P15 SS1/P34 ACK1/P35 AN10/TxD1/P36 AN11/RxD1/P37 AN9/PWM3/T30/INT3/P16 Pin list : 1/30 used, 2 power pins Pin# Function in/out Comment P10 6 input Ш

It shows the current target device's package shape and its pin assignment.

You can easily read the status of each pin based on its color:

Pin color	Meaning	
Red	This pin is a power source pin.	
Blue	This pin is a ground pin.	
White	This pin is not assigned a function yet.	
	This pin is assigned a specific function.	
Yellow	Assigned function is to change the text color to red.	
	Assigned pin list is displayed under the package shape.	



You can set a port function both in the **Properties** pane and from this window. By double-clicking a pin, you can edit port properties using the following dialog box:

Pin6 : P10/I	NT0/T10/PWM1AA		x
Function Direction	P10 ▼ input ▼	Pullu Ope Deb	ip register n drain ounce
Comment			
		ОК	Cancel



Chapter 4. Output files

4.1 Header files

CodeGen8 generates two header files as shown below. They are not affected by whether they belong to single-source or multisource projects.





4.1.1 Device header

The device header file contains the peripheral definitions. This is the real text file located in the device's Library folder. You do not need to modify this file when you are working in a KEIL environment.



The following is an example of MC95FG308.h:

```
1 #include <intrins.h>
  2
 3 #ifndef _MC95FG308_
  4 #define _MC95FG308_
  5
 6 //=====
      // PORT Control Register
 7
 8 //=====
 9 // PORTO

      P0
      = 0x80;

      P00
      = 0x80;

      P01
      = 0x81;

      P02
      = 0x82;

      P03
      = 0x83;

      P04
      = 0x84;

      P05
      = 0x82;

10 sfr
                                                                     // P0 Data Register
11
         sbit
            sbit PO1
sbit PO2
12
13
            sbit
14
          sbit PO3
           sbit P04
sbit P05
15
16
                                            = 0x85;
           sbit P06
                                          = 0x86;
17
           sbit
                          P07
                                            = 0x87:
18
                                          = 0x80;
                          PODA
19 sfr
                                                                 // PO Data Register
                                         = 0x80;
         sbit
                          PODAO
20
                          PODA1
           sbit
                                            = 0x81:
21
                                           = 0x82;
22
            sbit
                          PODA2
          sbit PODA3
                                           = 0x83;
23
          sbit
                          PODA4
                                             = 0x84;
24
                                            = 0x85;
25
            sbit
                          PODA5
          sbit PODA6
                                          = 0x86;
26

      sbit
      PUDAo
      - 0x80,

      sbit
      PODA7
      = 0x87;

      r
      POIO
      = 0x89;
      // PO Direction Register

      efine
      POPU
      * (volatile unsigned char xdata *) 0x2F00
      // PO Pull-up Register

      efine
      POOD
      * (volatile unsigned char xdata *) 0x2F0C
      // PO Open Drain Register

      efine
      PODB
      * (volatile unsigned char xdata *) 0x2F18
      // PO DEBOUNCE Register

27
28 sfr
29 #define
30 #define
31 #define
32 // PORT1
       sfr P1 = 0x88;
sbit P10 = 0x88;
sbit P11 = 0x89;
33 sfr
                                                                      // P1 Data Register
34
35
```

4.1.2 func_def.h

This file contains function definitions. This is a virtual text file that does not exist in any folder. The function name, function body, and comments are assigned by CodeGen8 automatically. You can recognize each function's operation easily with the function name and comments. You do not need to modify this file when you are working in a KEIL environment.

The following is an example of MC95FG308.h:



4.2 Source files

These files contain the skeleton of the real source code. The function name, function body, and comments are assigned by CodeGen8 automatically.



Example: Multisource project

4.2.1 startup.a51

It contains device memory and stack initialization code. This is a real file located in the device's Library folder. You do not need to modify this file when you are working in a KEIL environment.



The following is an example of startup.a51.

```
83 ?C C51STARTUP
                     SEGMENT
                               CODE
    2STACK
                     SEGMENT
 84
                               IDATA
 85
                     RSEG
                             ?STACK
 86
 87
                     DS
                             1
 88
 89 EXTRN CODE (?C START)
 90
 91
                     PUBLIC ?C STARTUP
 92
 93
                     CSEG
                             AΤ
                                      0
 94
     ?C STARTUP:
                     LJMP
                             STARTUP1
 95
 96
                     RSEG
                             ?C C51STARTUP
 97
 98
     STARTUP1:
 99
100
     IF IDATALEN <> 0
101
                     MOV
                             RO, #IDATALEN - 1
102
                     CLR
                             Α
103
    IDATALOOP:
                     MOV
                             @RO,A
104
                     DJNZ
                             R0, IDATALOOP
     ENDIF
105
106
     IF XDATALEN <> 0
107
108
                     MOV
                             DPTR, #XDATASTART
109
                     MOV
                             R7, #LOW (XDATALEN)
110
       IF (LOW (XDATALEN)) <> 0
111
                             R6,#(HIGH (XDATALEN)) +1
                     MOV
112
      ELSE
113
                     MOV
                             R6, #HIGH (XDATALEN)
114
       ENDIF
115
                     CLR
                             А
                             @DPTR,A
116 XDATALOOP:
                     MOVX
```

4.2.2 main.c

The main.c file is the most important in C programming. Every C program starts with the main function and ends with a null statement. The following are the characteristics of the main function:

- Any C program can have only one main function.
- This function is called by the reset vector.
- In MiCOM programming, the main function must not be terminated to prevent malfunction.



CodeGen8 generates the main.c file very clearly and simply. It contains initialization codes only. CodeGen8 generates slightly different main.c files in single-source and multisource projects. The main.c file in a single-source project contains the main function, interrupt vector functions, and initialization functions.

```
11 void main()
12 {
13
     cli();
                   // disable INT. during peripheral setting
     port_init();
                   // initialize ports
14
    port_init();
clock_init();
ExINT_init();
sei();
                   // initialize operation clock
15
                   // initialize external interrupt
16
17
     sei();
                    // enable INT.
18
    // TODO: add your main code here
19
20
     while(1);
21
22 }
23
25 // interrupt routines
26 //-----
27
28 void INT_BIT() interrupt 22
29 {
    // BIT interrupt
30
    // TODO: add your code here
31
32 }
33
34 //-----
35 // peripheral setting routines
36 //-----
37
38 void ExINT init()
39 {
40
     // initialize external interrupt
41
    EIEDGE = 0x00; // level / edge
42
    EIPOLA = 0x00;
                   // polarity
43
    EIBOTH = 0x00; // both polarity
44
    EIFLAG = 0x00;
                   // clear all flags
```

Example: Single-source project

The main.c file in a multisource project contains the main function routine only.



You will see that the file contents are neatly separated.

```
_____
 2
    // Main program routine
 3
    // - Device name : MC95FG308
 4
    // - Package type : 32QFN
 5
    6
 7
    // - Generated : Fri, Apr 26, 2013 (13:07:51)
    #include "MC95FG308.h"
 8
                  "func_def.h"
 9 #include
10
11 void main()
       cli(); // disable INT. during press
port_init(); // initialize ports
clock_init(); // initialize operation clock
BIT_init(); // initialize Basic interval t
BUZ_init(); // initialize Buzzer
ExINT_init(); // initialize External interry
Timer0_init(); // initialize Timer0
WaRT init(); // initialize UART interface
// crable INT.
12
    {
13
                                  // disable INT. during peripheral setting
14
15
16
                                  // initialize Basic interval timer
17
18
                                  // initialize external interrupt
19
20
21
22
23
         // TODO: add your main code here
24
25
         while(1);
26 }
27
```

Example: Multisource project

You can switch between a single- and a multisource project at any time.

4.2.3 interrupt.c

This file is optionally generated when you select a multisource project. This file contains the interrupt vector functions only. Vector functions are automatically sorted by their interrupt numbers. Each vector contains a comment that shows the interrupt source.





You will see that the file contents are neatly separated.

```
1 //=====
                        _____
2
   // interrupt routines
3
   //-----
4
           "MC95FG308.h"
5 #include
6 #include "func_def.h"
7
8 void INT_USARTO_Rx() interrupt 6
9
   {
      // USARTO Rx interrupt
10
      // TODO: add your code here
11
12 }
13
14 void INT_I2C() interrupt 9
15 {
      // I2C interrupt
16
17
      // TODO: add your code here
18 }
19
20 void INT_BIT() interrupt 22
21 {
22
      // BIT interrupt
23
      // TODO: add your code here
24 }
25
```

Example: interrupt.c

4.2.4 peripheral.c

This file is optionally generated when you select a multisource project. This file contains MCU peripheral functions only. All functions are automatically sorted by their names. Each function contains the initialization code and comments.



You will see that the file contents are neatly separated.

```
21 void BUZ_init()
22 {
23
       // initialize Buzzer
24
       // Frequency (Hz) = 61.035156
       BUZCR = 0x04; // clock source
BUZDR = 0xFF; // count value
25
26
27
   }
28
29 void ExINT init()
30 {
31
        // initialize external interrupt
       EIEDGE = 0x00; // level / edge
32
       EIPOLA = 0x00;
33
                           // polarity
       EIBOTH = 0x00;
                           // both polarity
34
35
       EIFLAG = 0x00;
                           // clear all flags
       IEN0 |= 0x00;
                           // INT. 3,2,1,0
36
37
       IEN4 |= 0x00;
                           // INT. 4,5
                          // INT. 7,6
38
       IEN5 |= 0x00;
       EIENAB = 0x00;
                          // enable INT pin
39
40 }
41
42 void I2C_init()
43 {
       // initialize I2C interface
44
45
       // Master : Frequency(Hz) = 3906.250000
       // High duty = 49.902344(%)
46
       I2CSCLLR = 0x80; // low count
47
       I2CSCLHR = 0x7F;
                           // high count
48
49
       I2CMR = 0x7C;
                          // setting
50
       IEN1 |= 0x80;
                          // Enable I2C interrupt
51 }
```

Example: peripheral.c