ABOV CodeGen32

(Code Generator)

USER GUIDE

Release V1.000.00

@ANU BncdFdm21 ' Bncd Fdmdq`snq (TRDQ FTHCD

Qdkd`rd hmenql`shnm

Udqrhnm	C`sd	Bg`mfd
V1.000.00	July 2016	First release

Oqnoqhds`qx mnshbd

The product described in this document is subject to continuous developments and improvements. All particulars of the product and its use contained in this document are given by ABOV Semiconductor Co.,Ltd. in good faith.

However, all warranties implied or expressed, including but not limited to implied warranties or merchantability, or fitness for purpose, are excluded.

This document is intended only to assist the reader in the use of the product. ABOV Semiconductor Co.,Ltd. shall not be liable for any loss or damage arising from the use of any information in this document, or any error or omission such information, any incorrect use of the product.

Cnbtldms bnmehcdmsh`khsx rs`str

This document is Open Access. This document has no restriction on distribution.

Oqnctbs rs`str

The information in this document is Final (information on a developed product).

@ANU vda `ccqdrr

http://www.abov.co.kr

Contents

ABOV CodeGen32 (Code Generator) USER GUIDE

Chapter 1
Getting Started

This chapter describes:

- **System requirements**
- **Setup package installation**

0-0□□ □□

This section described the hardware and software system requirements.

0-0-0 □□

You can install and run this S/W on one of the following operation systems.

- MS-Windows NT
- MS-Windows 2000
- MS-Windows XP
- MS-Windows Vista
- MS-Windows7
- MS-Windows8
- MS-Windows10

Chrj ro`bd

If you wish to carry out a full installation of the software, up to 100MB of hard disk space is required.

0-0-1 □□

This S/W is not so picky to install or run.

The following are the minimum recommended hardware requirements for installing and running this S/W.

- Pentium PC

Performance is based on following factors:

- Processor performance

0-1□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

You can download the install program from our website (http://www.abov.co.kr).

You had better to keep newest software because we add new devices and newer features continuously.

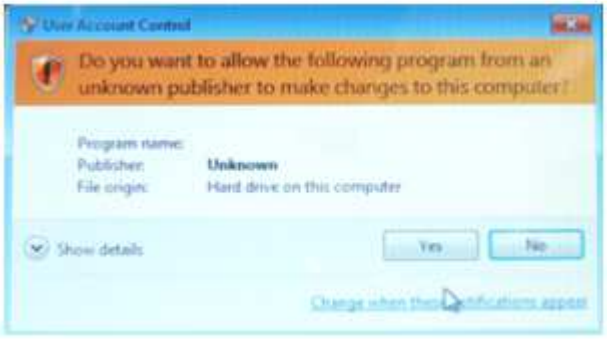
We provide install package which can be installed on any of Microsoft Operating system.

0-1-0 □□□

Execute setup program.

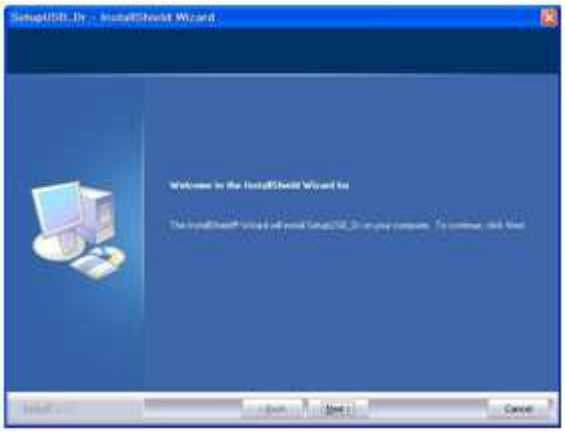
You can see the warning message as below.

Click "Yes" button.



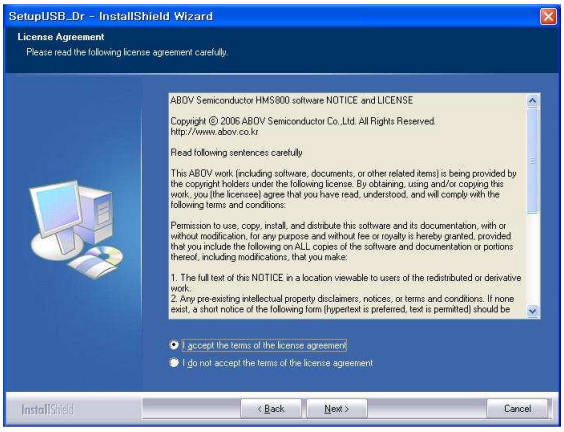
And then you can see following dialog box.

Click "Next" button.



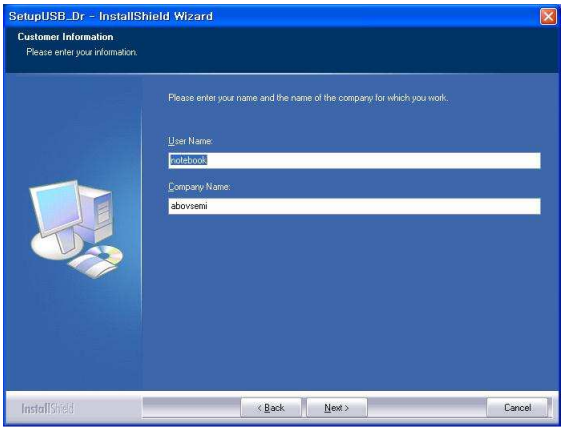
When the license agreement dialog box is appeared, select "I accept the items of the license agreement".

Click the "Next" button.



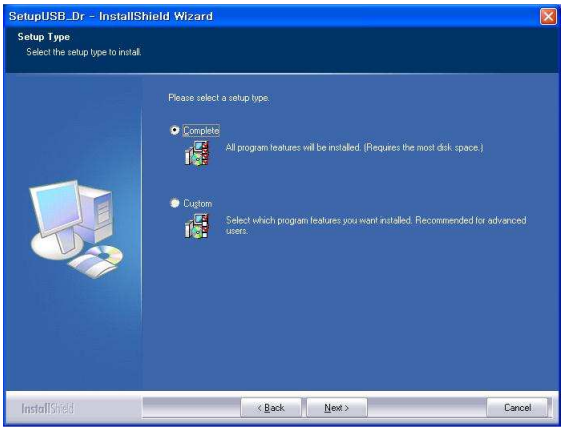
Fill the user name and company name.

Click the "Next" button.

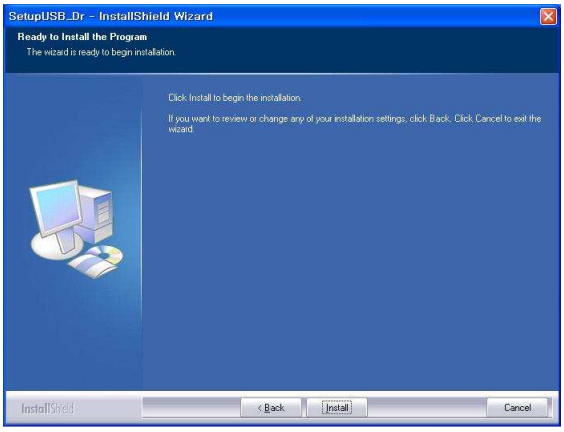


Select "Complete".

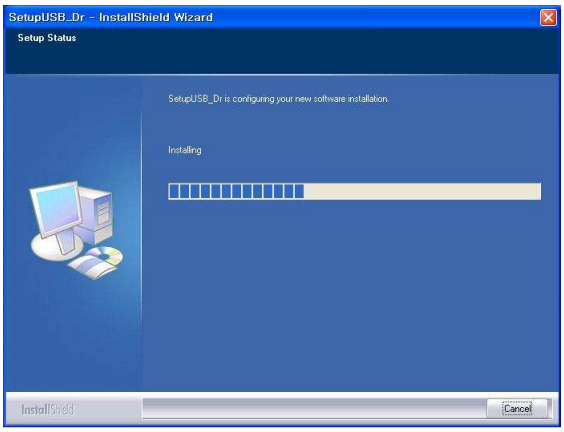
Click the "Next" button.



Click “Install” button.

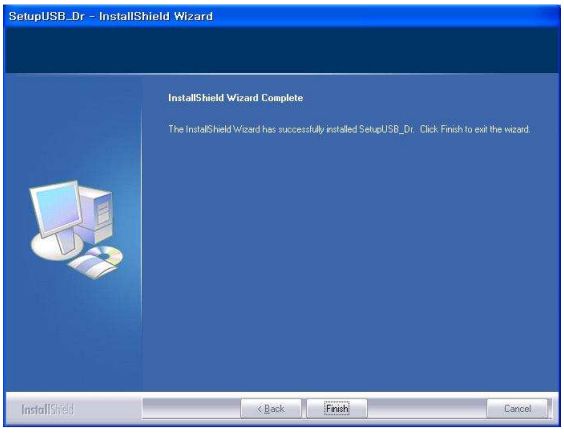


Wait until it installs all of the program components.



Installation is completed.

Click “Finish” button.



Chapter 2
CodeGen32 S/W folder structure

This chapter describes:

- CodeGen32 project folder structure
- CodeGen32 library folder structure

1-0 BncdFdm21 oqnidbs enkcdq rsqtbstqd

CodeGen32 S/W maintains each CodeGen32 project files within “Project” folder

“Project” folder is located in CodeGen32 installed folder.

Each CodeGen32 project files are composed with *.VPR and *.VDA files.

*.VPR file contains basic information (device name, package type, pin count, etc)

Ex)

ABOV-CodeGen32-A3x V1.000.00 20160422

AC33M8128 MQFP 80

*.VDA file contains target device’s peripheral property settings.

Ex)

*Clock 1

*GPIO-A 1

*GPIO-B 1:

:

u02 000000.....00000000

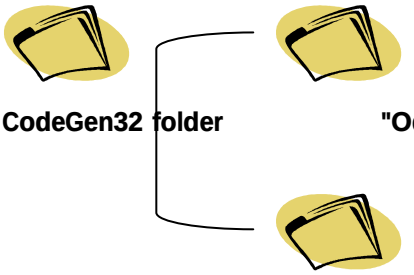
u03 000000.....00000000

w00 040000.....CA000000

Warning :

Do not modify *.VPR or *.VDA file.

Folder assignment



1-1 BncdFdm21 khaq`qx enkcdq rsqtbstqd

CodeGen32 S/W maintains each device's sample files within "Library" folder

"Library" folder is located in CodeGen32 installed folder.

"Library" folder includes each device's library folders.

Device library folder contains followings

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

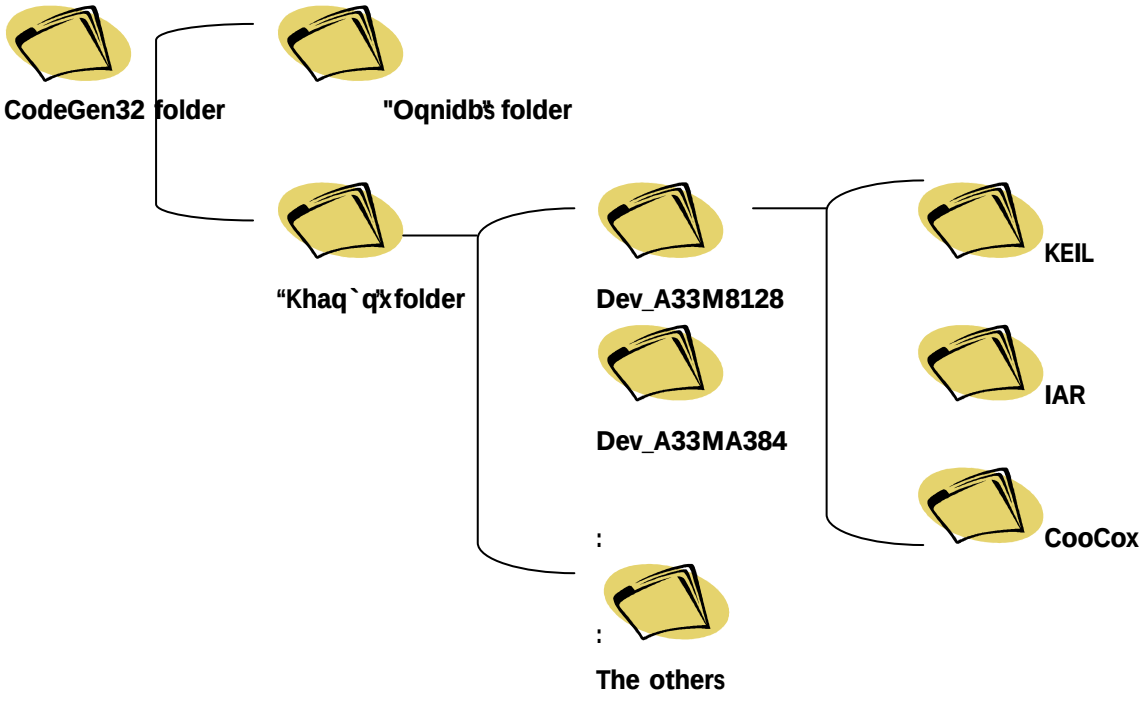
CodeGen32 S/W generate standard basic source program.

If you want to see more details, you can refer the sample source files.

Warning :

Do not delete or modify any of files in library folder.

Folder assignment



Chapter 3
Using CodeGen32 S/W

This chapter describes:

- CodeGen32 S/W feature
- Menu usage
- Control panes
- Child windows

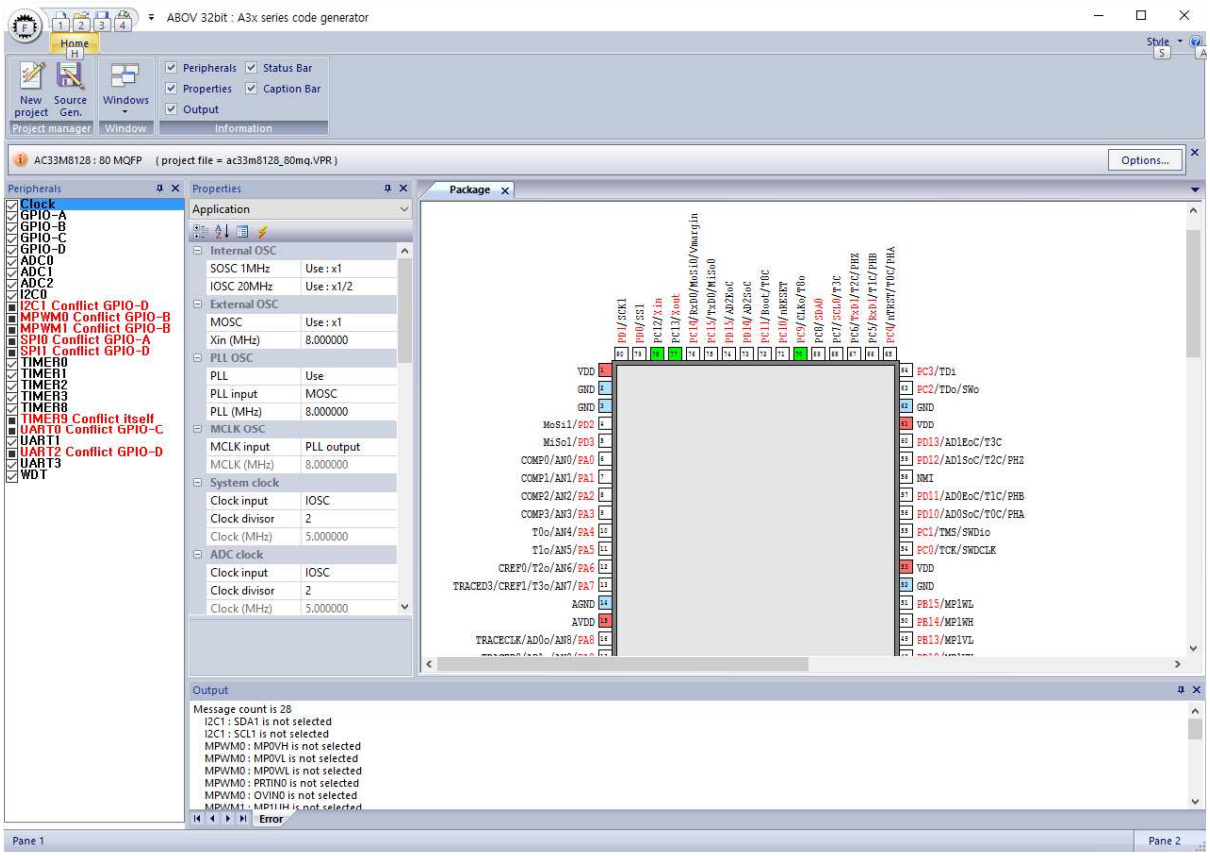
2-0 BncdFdm21 R.V ed`stqd

It makes or loads new project.

CodeGen32 S/W supports ABOV A3x device series.

Target device will be added continuously.

Ex) CodeGen32 S/W screen shot



2-0-0 Cds`hkdc ed`stqd

It supports A3x series of ABOV Semiconductor Co.,Ltd.

Followings are supported.

- **It generates C source files and libraries for target device automatically.**
- **It supports KEIL project format. (Year 2016)**
  - **Generate *.uvproj project file.**
  - **Generate each device's header files.**
  - **Create and copy library folder and its files.**
  - **Generate C source files.**
    "main.c", "init.c", "peri.c"
- **The other compiler project format is planed too.**
  - **Ex : IAR, CooCox, etc.**
- **It displays followings.**
  - **Peripherals list**
  - **Properties**
  - **Package view**
  - **Output (Error messages)**
- **Package view shows pin assignment and you can set port function.**
- **Check and display each peripheral's confliction as soon as you changed peripheral setting.**
- **It manages CodeGen32 project files automatically.**

2-0-1 Rs`qs rnesv`qd

If you start CodeGen32 S/W, you can see a following dialog box.

Most of controls are disabled.

Enabled controls are “Series name”, “Device name”, “Package type”, “Cancel” only

S/W usage is very simple.

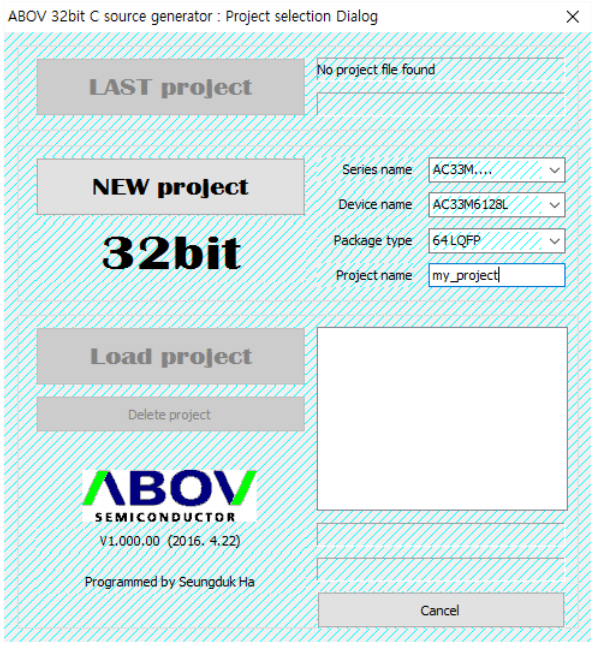
Select target series first

Select target device as following figure.

Select device package as following figure.

Type your project name.

“New project” button will be enabled automatically.



Click “New project” button.

If you made more than 1 CodeGen32 project(s), you can see as following dialog box.

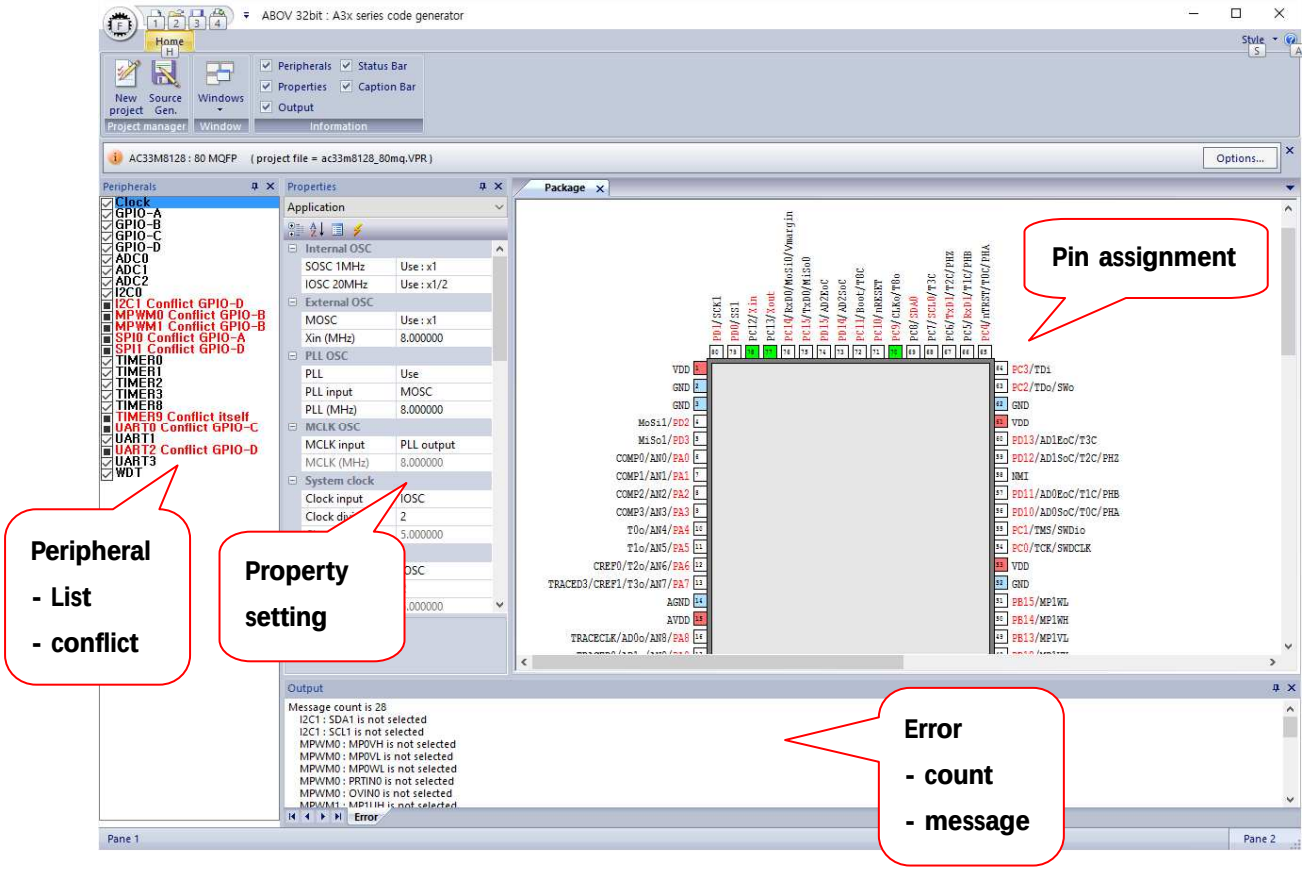
In upper case,

- **You can load the last project by clicking "LAST project" button.**
- **You can make new project.**
- **You can load a project within previous projects.**

Delete project

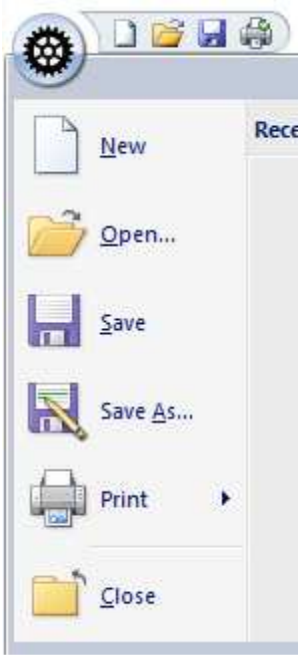
- **You can delete un-using project.**

Target project is selected, the main CodeGen32 S/W will be executed as following.



2-1 A`rhb ldmt tr`fd

It is familiar basic functions to all of S/W users.



2-2 Qhaanm ldmt tr`fd

In computer interface design, a ribbon is a graphical control element in the form of a set of toolbars placed on several tabs.

CodeGen32 S/W adopted very simple and easy user interface.

There are only two function buttons and few check buttons only.

2-2-0 Mdv oqnidbs

It makes or loads new project.

And ask new project or load a project within previous projects.

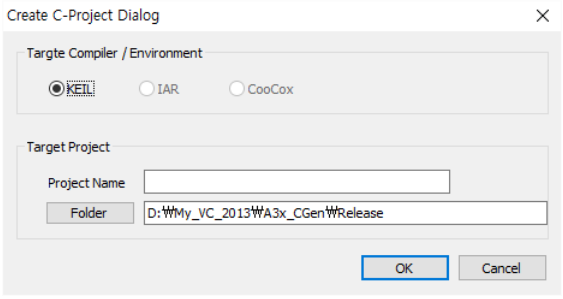
If you make new project or select another project, it closes current project and start selected project immediately.

Current project data will be saved automatically.

2-2-1 Rntqbd Fdm-

CodeGen32 S/W generates source file (header and C) and copy library folder and its contents to selected project folder.

Now, it support KEIL compiler project only.



If there is warning within the device setting, it asks continue or not.

If the warnings are not cleared, generated source file omit some settings.

You can see the warning within Output pane.

Rntqbd ehkd fdmdq`shnm

CodeGen32 S/W generates following files.

- Header files
  - init.h : it includes library header files and define something
  - peri.h : it defines peripheral initial functions
- Source files
  - init.c : it calls peripheral initialization functions.
  - main.c : it includes main(void) function only.
  - peri.c : It includes peripheral initial functions.
- KEIL project file
  - KEIL_proj.uvproj : It contains target device, file management, etc

Khaq`qx bnox

CodeGen32 S/W saves following files.

- Driver files
- Sample files
- Documents

Mnshehb`shnm

If there is same named file exist, it asks overwrite or not.

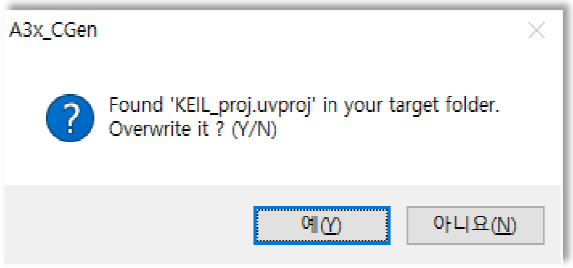
If you select file overwrite, it copy file into the folder, and it does not care the target files were modified by yourself or not.

We recommend to selecting another folder, not your current working project folder.

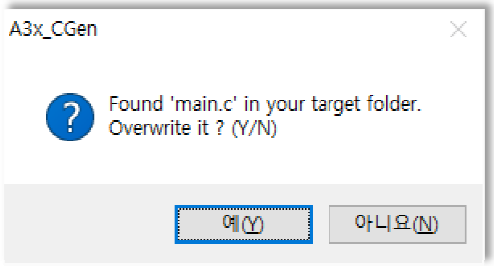
CodeGen32 does not check and does not care some files are exist or not.

- **“Init.c”, “peri.h”, “peri.c”**

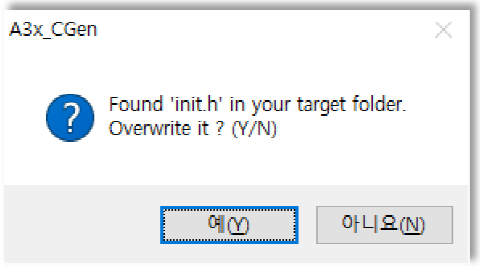
Ex) If project file found.



Ex) If main.c file found.



Ex) If init.h file found.

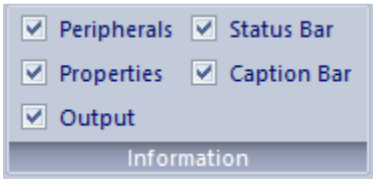


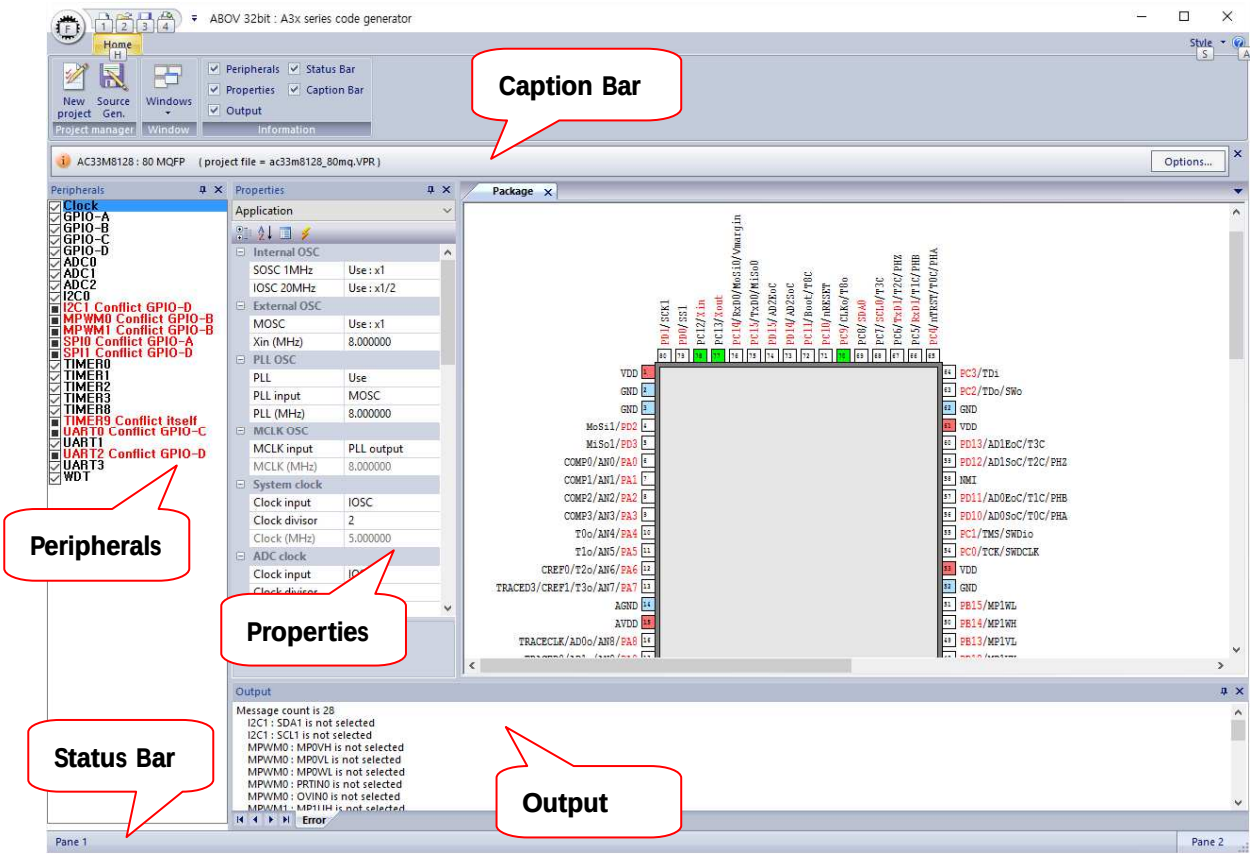
2-2-2 Vhmcnvr

It controls child windows selection or arrangements, etc.

2-2-3 Hmenql`shnm

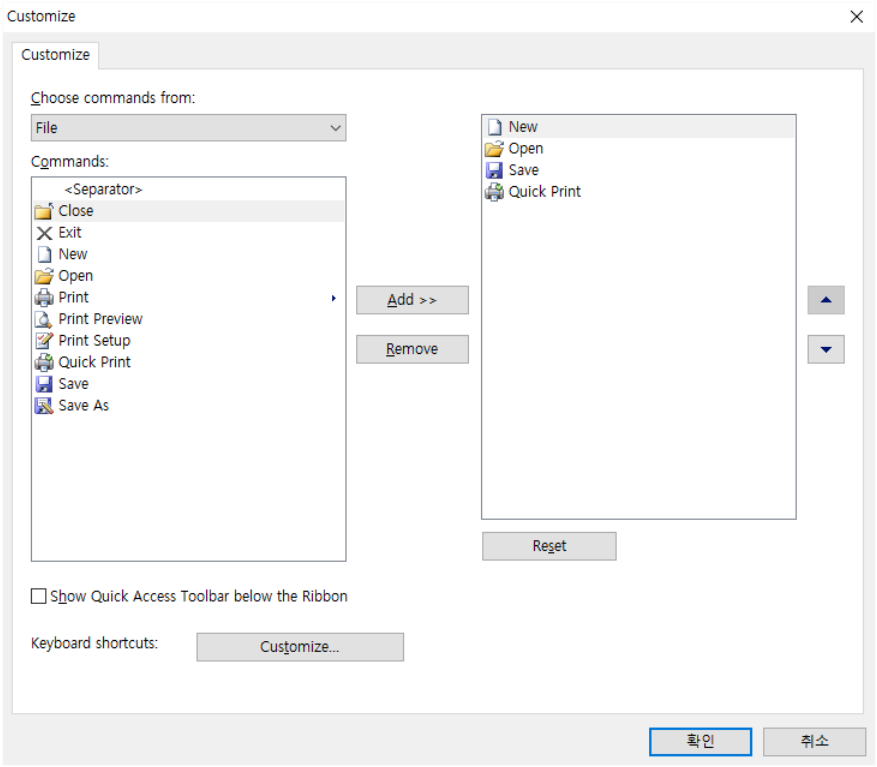
It controls show or hide each information display pane.





B`oshnm a`q . Noshnm

It offers to modify menu.



2-3 Bnmsqnk o`mdr

CodeGen32 S/W provides 3 control panes to developer.

It shows device's peripherals, peripheral setting, Error message.

All panes support docking feature.

Docking means that the moving pane will be placed each pane's border or move into the other pane, etc.

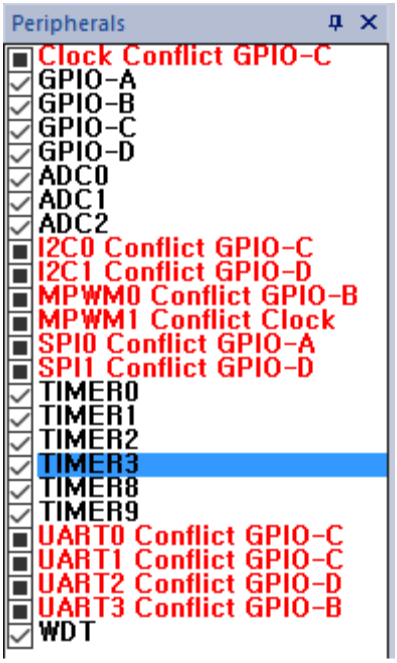
2-3-0 Odqhogdq`kr o`md

It shows file peripherals of current device.

It shows which peripherals are used or not.

And more, it shows peripheral confliction if exist.

Ex) Screen shot



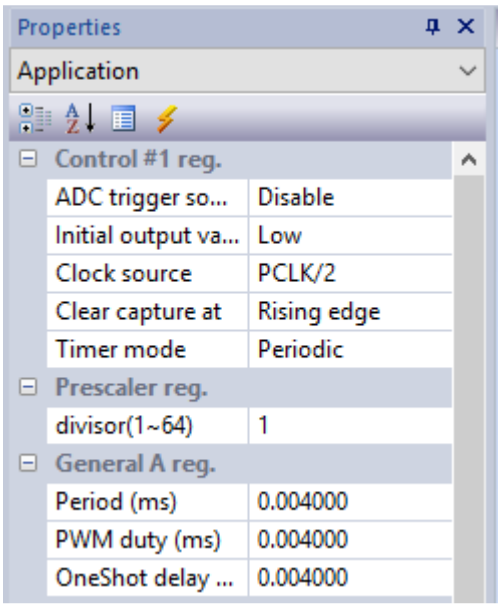
2-3-1 Oqnodqshdr o`md

It shows device setting of current selected peripheral.

This is the most important component of CodeGen32 S/W.

- **The contents are closely related to each device's specification.**
- **You can see or edit the peripheral settings.**
- **If you change any of setting, it calculates some data and shows it automatically.**
- **If device setting is conflict with the other peripheral, Peripheral pane and output pane shows it.**

Ex) screen shot

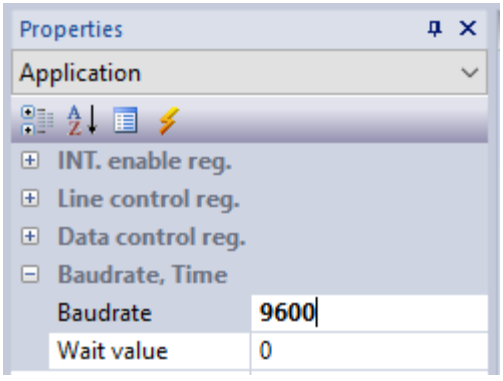


➢ **Example :**

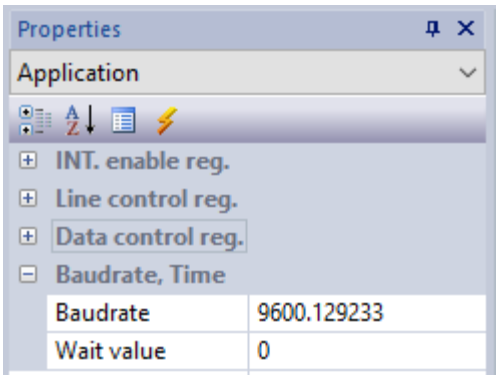
Let's assume that the system clock is 66MHz, and you want to use UART with 9600bps.

You will find out the exact 9600bps is not possible from 66MHz.

CodeGen32 S/W calculates the nearest bps, and then shows it and generate source program.



Your input : 9600bps



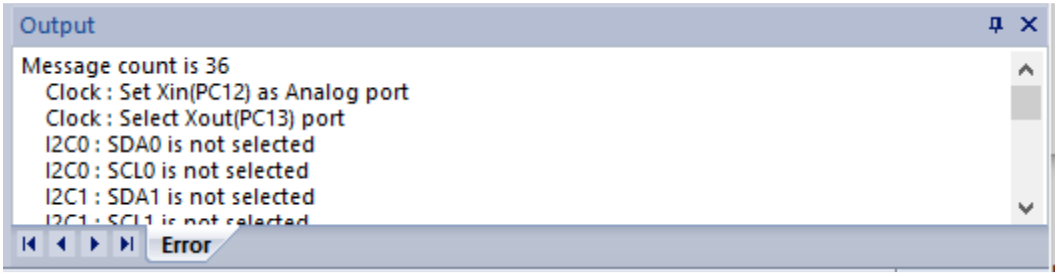
CodeGen32 calculation : 9600.129bps @66MHz

You do not need to know whole device specification, because of it is very simple to setting all of peripherals.

Anyway, if you want to more sophisticated setting, you had better to understand the device's detailed specification.

2-3-2 Ntsots o`md

It shows Error messages of device peripheral setting or confliction.

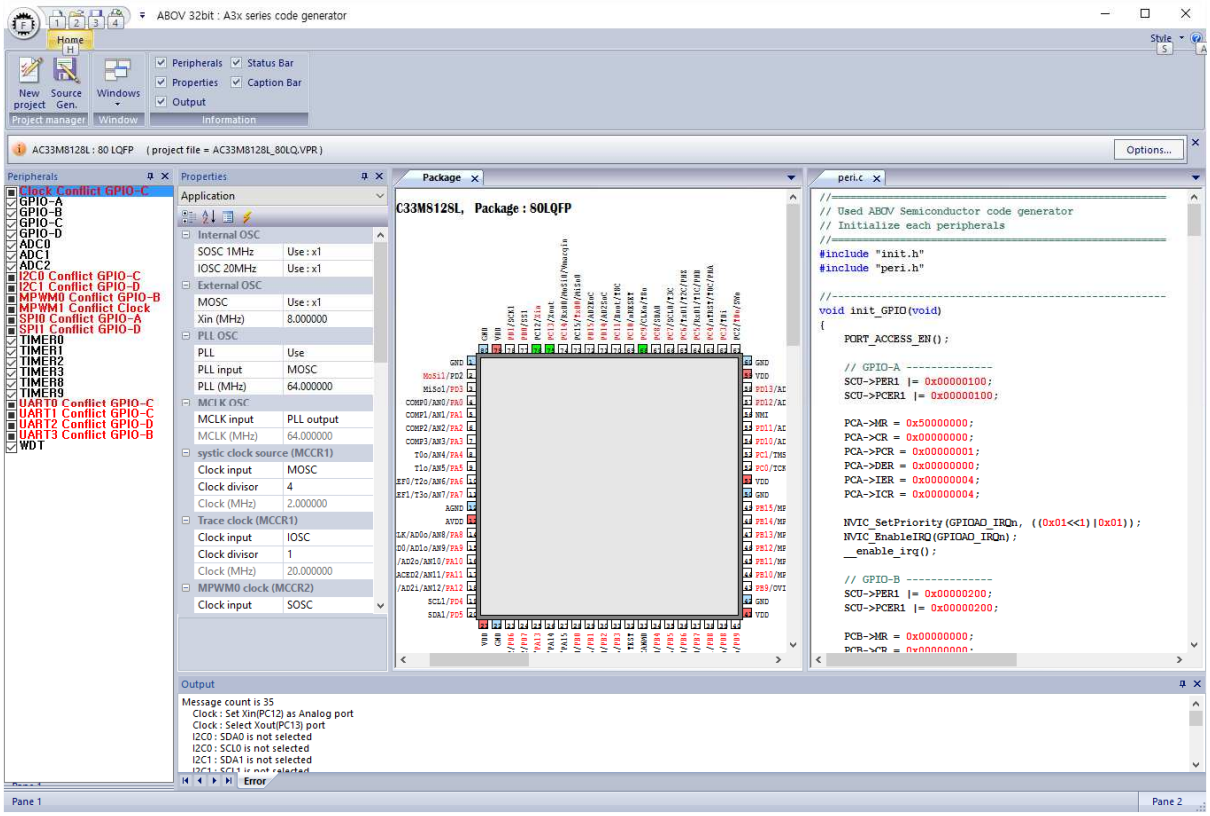


You must clear the warning messages by changing the device peripheral setting.
Or not, generated C source program omit some peripheral settings.
And more, compiled code execution will not be worked correctly.

2-4 Bghkc vhmcnvr

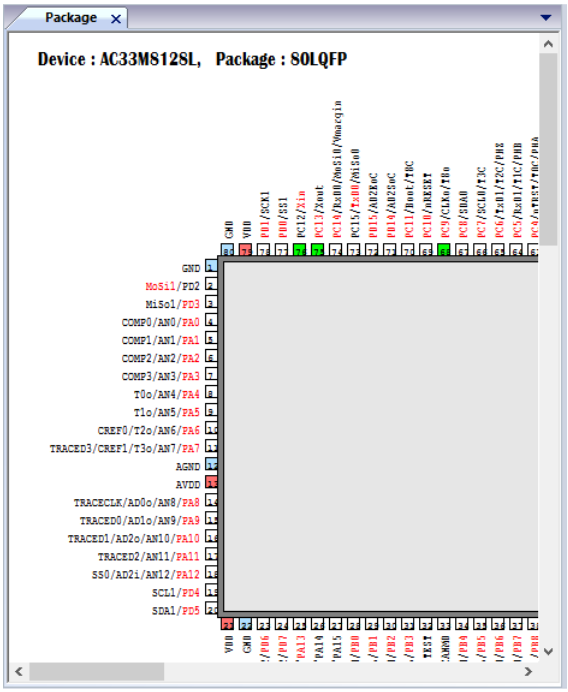
Child windows are differ from each pane controls.

It shows device package viewer or text file



2-4-0 O`bj`fd uhdvdq

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

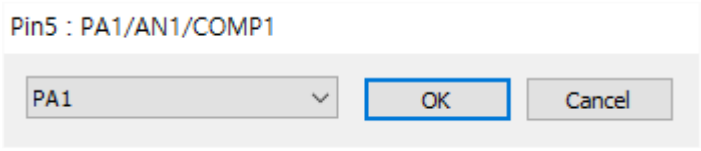


You can read easily with each pin status with its color.

Pin color	Meaning
Red	It is a power source pin.
Blue	It is a ground pin.
White	Port, except power (Vcc, Gnd)
Green	It is associated pin with the selected peripheral.

You can set a port function within property pane and this window also.

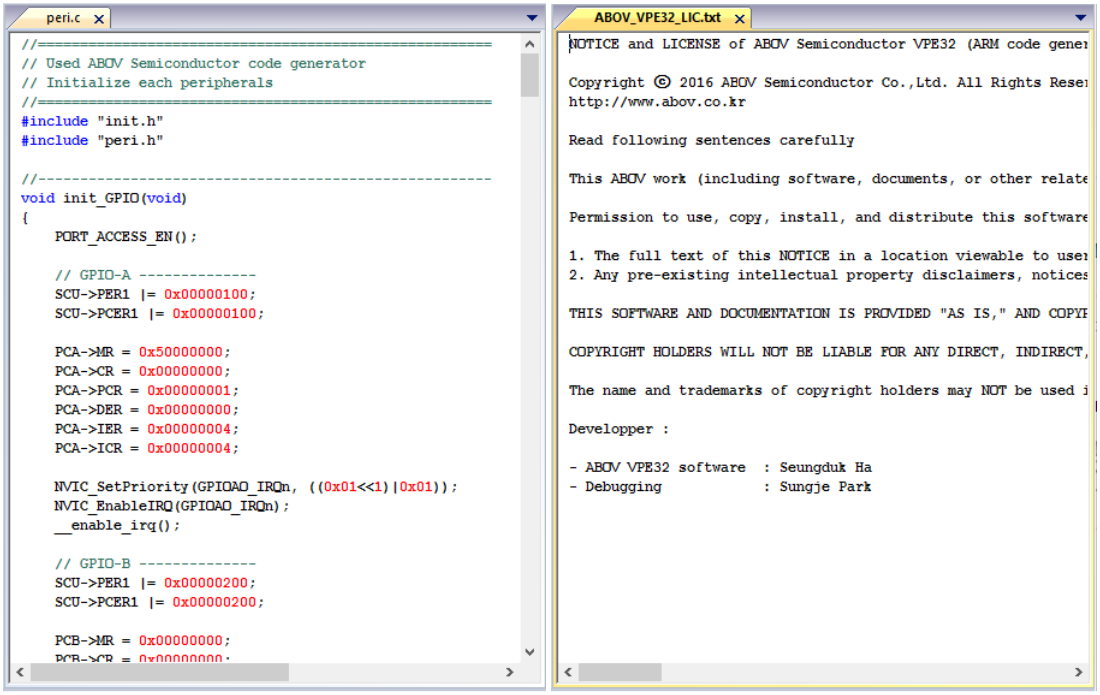
If you double click a pin, you can edit port function with following dialog box.



2-4-1 Sdws ehkd uhdvdq

It shows text file.

If file extension name is 'c' or 'h', it shows with colored text to increase readability.



File editing is not supported.

Its displaying TAB size is fixed to 4.

Chapter 4
Output files

This chapter describes:

- **CodeGen32 generating files**
- **CodeGen32 coping library**

3-0 Gd`cdq ehkdr

CodeGen32 S/W generates header files as following.

3-0-0 hmhs-g

It contains its peripheral header files and some definitions.

You do not need to modify this file when you are working on KEIL environment.

Ex) Following is init.h of AC33M8128.

```
//=====================================================
// Used ABOV Semiconductor code generator
// Device name : AC33M8128
//=====================================================
#include "AC33Mx128.h"
#include "system_AC33Mx128.h"
#include "aa_types.h"
#include "ac33mx128_adc.h"
#include "ac33mx128_afe.h"
#include "ac33mx128_dmac.h"
#include "ac33mx128_gpio.h"
#include "ac33mx128_i2c.h"
#include "ac33mx128_libcfg.h"
#include "ac33mx128_mpwm.h"
#include "ac33mx128_pcu.h"
#include "ac33mx128_pwr.h"
#include "ac33mx128_scu.h"
#include "ac33mx128_spi.h"
#include "ac33mx128_timer.h"
#include "ac33mx128_uart.h"
#include "ac33mx128_wdt.h"
#include "debug_frmwrk.h"
#include "slib.h"

#define _ADC
#define _AFE
#define _DMAC
#define _GPIO
#define _I2C
#define _MPWM
#define _SPI
#define _TIMER
#define _UART
#define _WDT

void init(void);
```

3-0-1 odqh-g

It defines initializing functions.

Function name, function body, comment are assigned by CodeGen32 S/W automatically.

You can recognize easily each function's operation with function name and comments.

You do not need to modify this file when you are working on KEIL environment.

Ex) Following is peri.h of AC33M8128.

```
//=====================================================
// Used ABOV Semiconductor code generator
// Define initialize function of each peripherals
//=====================================================

void init_GPIO(void);
void GPIOAO_IRQHandler(void);
void GPIOCE_IRQHandler(void);
void init_clock(void);
void init_ADC_0(void);
void init_ADC_1(void);
void init_ADC_2(void);
void init_I2C_0(void);
void init_I2C_1(void);
void init_MPWM_0(void);
void init_MPWM_1(void);
void init_SPI_0(void);
void init_SPI_1(void);
void init_TIMER_0(void);
void init_TIMER_1(void);
void init_TIMER_2(void);
void init_TIMER_3(void);
void init_TIMER_8(void);
void init_TIMER_9(void);
void init_UART_0(void);
void init_UART_1(void);
void init_UART_2(void);
void init_UART_3(void);
void init_WDT(void);
```

3-1 Rntqbd ehkdr

CodeGen32 S/W generates source files as following.

3-1-0 hmhs-b

It contains peripheral initialization code.

You do not need to modify this file when you are working on KEIL environment.

Ex) Following is init.c of AC33M8128.

```
//=====================================================
// Used ABOV Semiconductor code generator
// Basic initialize function
//=====================================================
#include "init.h"
#include "peri.h"

void init(void)
{
    init_GPIO();
    init_clock();
    init_ADC_0();
    init_ADC_1();
    init_ADC_2();
    init_I2C_0();
    init_I2C_1();
    init_MPWM_0();
    init_MPWM_1();
    init_SPI_0();
    init_SPI_1();
    init_TIMER_0();
    init_TIMER_1();
    init_TIMER_2();
    init_TIMER_3();
    init_TIMER_8();
    init_TIMER_9();
    init_UART_0();
    init_UART_1();
    init_UART_2();
    init_UART_3();
    init_WDT();
}
```

3-1-1 l`hm-b

As you know, this file is the most important in programming C.

Every c program starts with a main function and end with null statement.

Properties of main function:

- **Any c program can have only one main function.**
- **Generally in MCU programming, main program must not be terminated to prevent malfunction.**

CodeGen32 S/W generates main.c very clearly and simply.

It contains initialization codes only.

You can modify main.c as you want.

If you want “main.c” not be overwritten by CodeGen32 S/W code generation, you have to care code overwriting dialogboxs

Ex) Following is main.c of AC33M8128.

```
//=====================================================
// Used ABOV Semiconductor's code generator
// Device name : AC33M8128
//=====================================================
#include "init.h"

int main()
{
    WDT->CON = 0;    // disable watch-dog timer
    init();          // initialize selected peripherals here

    while(1) {
        // TOTO : Fill your code
    };
    return 0;
}
```

3-1-2 odqh-b

This file contains all you selected peripheral setting.

You do not need to modify this file when you are working on KEIL environment.

Ex) Following is peri.c of AC33M8128.

```
//=====================================================
// Used ABOV Semiconductor code generator
// Initialize each peripherals
//=====================================================
#include "init.h"
#include "peri.h"

//-----------------------------------------------------
void init_GPIO(void)
{
    PORT_ACCESS_EN();

    // GPIO-A --------------
    SCU->PER1 |= 0x00000100;
    SCU->PCER1 |= 0x00000100;

    PCA->MR = 0x50000000;
    PCA->CR = 0x00000000;
    PCA->PCR = 0x00000001;
    PCA->DER = 0x00000000;
    PCA->IER = 0x00000004;
    PCA->ICR = 0x00000004;

    NVIC_SetPriority(GPIOA0_IRQn, ((0x01<<1)|0x01));
    NVIC_EnableIRQ(GPIOA0_IRQn);
    __enable_irq();

    // GPIO-B --------------
    SCU->PER1 |= 0x00000200;
    SCU->PCER1 |= 0x00000200;
```

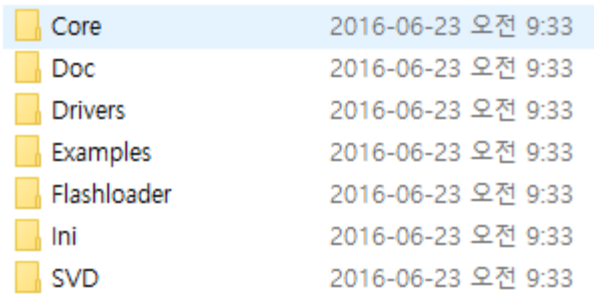
3-2 Khaq`qx

CodeGen32 S/W copies library folder and related files.

It contains header, source files and example programs, documents too.

Current CodeGen32 S/W supports KEIL library only.

Do not modify or rename or delete library files.



End of document.