An Example of using Keil uVision3 to create Project File of Keil ARM

In this case, we will mention about the proceeding to write program by using C Language Program that is Keil-CARM. It is used to interpret command under Program Text Editor of Keil (Keil uVision3). We only mention about the proceeding to configure Option value for connection commands of interpretation program together by using Keil-CARM through Keil uVision3. For more detailed commands and functions usage for writing program by Keil-CARM, user can learn them by self from User's Manual command of Keil-CARM. We can summarize the proceeding to configure default values of Keil uVision3 for using with Keil-CARM as follows;

1. Open program Keil uVision3 that is a program Text Editor of Keil-CARM, it is used to write C Language Source Code program and the feature of this program is look like in the picture below.

🕎 ?Vision3			
Eile Edit View Project Debug Flash Peripherals Tools SVCS Window Help			
 ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	• • \1 \$ Q <mark>[]</mark> ;	a 🚸 🕅 🕅	
A hundled			
8 M M Build & Command A Find in Files /			
		NUM	OVR R/W

Example of Using Keil uVision3 to Create Project File of Keil-CARM

2.Configure default values to interpret commands of uVision3 and can be used with Program Keil uVision3 and Keil-CARM. Click **Project** → **Components**, Environment, Books ... and then select default value for Compiler from the title Select ARM Development Tools that has 3 modes; Use Keil-CARM Tools, Use GNU Tools and Use ARM Tools. In this case, we must select "Use Keil ARM Tools", and then we must configure position of folder to store default program Keil ARM. Generally, values of is it in "C:\Keil\ARM\" but if we install Keil in other folder, we must change the format of it suitably and corresponding with truly usage as in the picture below.

Components , Environment and Books	X		
Project Components Folders/Extensions Books			
Development Tool Folders			
Use Settings from TOOLS.INI:	C Source: *.c		
Tool Base Folder: C:\Keil\ABM\	C++ Source: *.CPP		
BIN: C:\Keil\ARM\BIN\	Asm Source: *.s*; *.src; *.a*		
INC:	Object: *.obj		
LIB:	Library: ^{*.lib}		
Regfile:	Document: [*] .txt; *.h; *.inc		
Select ARM Development Tools Use RealView RealView Folder: C:\Keil\ARM\BIN\ Use Keil CARM Compiler Use GNU GNU-Tool-Prefix: arm-uclibc- Compiler			
OK Cancel De	efaults Help		

3. Create new Project File by using command Project → New Project and then configure or create position Folder that we want to save new Project File with preferred Project File name. For example, if we create new Project File named DEMO1 and wants to save it into Folder named DEMO1; we can configure position of Folder and Project File name by self. After we configure Project File name in the blank of File Name successfully, click Save to save new Project File as in the picture below.

Create New Proj	ect			? 🔀
Save in: 🗀 DEM	01	-] 🗢 🔁 (* 🎟 •
File name: DEM	401			Save
Save as type: Pro	ect Files (*.uv2)		•	Cancel

After we configure new Project File name and save it completely, Program will wait for user to configure MCU number that is used in the saved Project File. If using with Board "CP-JR ARM7 USB-LPC2148", we must configure MCU number to be LPC2148 from Philips and then select **OK** as in the picture below.

Select Device for Target 'Target 1'	×
CPU Vendor: Philips Device: LPC2148 Toolset: ARM Data base Description: ARM/TDMI-S based high-performance 32-bit RISC Microcontroller with Th LPC2138 LPC2138 LPC2141 LPC2144 LPC214 LPC214 LPC214 LPC214 LPC214 LPC214 LPC214 LPC214 LPC244	
OK Cancel Help	

FTI

After we configure MCU number successfully, in this step, program will wait for user to confirm copy File Startup of Keil and wants to use it with MCU of Philips in new Project File or not. Startup File is the part to configure the default value of operation for MCU; for example, to configure Stack value and to configure value to run for Phase-Lock-Loop before start running follow by our written program. Otherwise, our written program must totally be added these commands into operation of MCU by self.

File Startup of Keil-ARM is Assembly Language File that is configured operation values with development set of Keil, so some configurations and default values are different and it makes Board "CP-JR ARM7 USB-LPC2148" can not be used with File Startup instantly. Therefore, we must modify some default value before using with program Keil-CARM. For interpretation commands, we must modify new File Startup and must set new format that is corresponding with the board's need. In this case, we will recommend selecting "No" to protect Keil uVision3 not copy File Startup of Keil-CARM to use in Project.

?Vision3) 🛛 🕅
?	Copy Philips LPC2100 Startup Code to Project Folder and Add File to Project ?
	Yes No

4. Copy File named "Startup.s" that ETT has already provided in CD-ROM and is saved in Example named "Startup.s", then to place it in the same position folder of new Project File that we created completely.

File "Startup.s" is a file that contains Assembly Language Commands of ARM7 to configure the necessary default value for MCU; for example, to configure Stack value into MCU, to configure Initial Phase-Lock-Loop, to configure value into MAM Function and to configure position Vectors of MCU. For using with Board "CP-JR ARM7 USB-LPC2148", if we Add File "Startup.s" from Keil or Copy this File from other positions, it will be effected on the operation of program in Startup because some operations are different.

5. Configure Option value of Project File by using command Project → Option for Target 'Target 1' and then select Tab of Target to configure value of MCU Target as follows. 5.1 Configure X-TAL to be 12 MHz and then configure Memory internal MCU to be condition of interpretation program of Keil-CARM as in the picture below.

Options for Target 'Target 1'	
Device Target Output Listing C Asm LA Loca Philips LPC2148	te LA Misc Debug Utilities
Operating system: None	 □ Big Endian ☑ Use On-chip ROM (0x0 - 0x7FFFF) ☑ Use On-chip RAM (0x40000000 - 0x40007FFF)
External Memory Start: Size:	Start: Size:
#1: RAM 💌	#4: BAM 💌
#2: RAM -	#5: RAM
₩3: JHAM ▼ J	#6: HAM _
OKCar	ncel Defaults Help

5.2 Output: we must click default values of Create HEX File, configure format of Hex to be HEX-386 and then select OK as in the picture below.

Options for Target 'Target 1'	
Device Target Output Listing C Asm LA Locate LA Misc Debug Utilities	
Select Folder for Objects Name of Executable: DEMO1	
Create Executable: .\DEM01	
Debug Information Browse Information	
Create HEX File HEX Format: HEX-386 Start: End:	
Offset:	
C Create Library: .\DEM01.LIB	
After Make	- I
✓ Beep When Complete	
Run User Program #1: Browse	
Run User Program #2: Browse	
OK Cancel Defaults He	lp

6. Start writing C Language Source Code, click command File → New... and we will get the available are to write Text File. In the first time, we must configure File name to be "Text" follow by the Default as in the picture below.

🛿 øVision3 - [Text1]	
Elle Edit View Project Debug Flash Peripherals Tools SVCS Window Help	_ & ×
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
	~
x modeline	
Build (Command) Find in Files /	
Ready L:1 C:1 NU	M R/W

In this step, it is typing C Language Source Code in the available area under configurations of Keil-CARM and we can write program preferably as in the picture below.

```
/* Examples Program For "CP-JR ARM7 USB-LPC2148"
                                              */
/* Target MCU : Philips ARM7-LPC2148
                                              */
/*
        : X-TAL : 12.00 MHz
                                              */
/*
             : Run Speed 60.00 MHz (With PLL)
                                              */
/*
             : PLL Setup = M(5), P(2)
                                              */
/*
             : VPB Clock = CPU Clock = 60 MHz
                                              */
/* Keil Editor : uVision3 V3.03a
                                              */
/* Compiler : Keil CARM V2.50a
/* Function : Example LED Blink on GPI01[24]
                                              */
                                             */
// Connect P1.24 to LED For Test ON / OFF (Blink)
#include "LPC214x.H"
                                // LPC2148 MPU Register
/* pototype section */
void delay(unsigned long int); // Delay Time Function
int main(void)
{
                              // Set GPIO-1[24] = Output
 IODIR1 = 0 \times 0100000;
 IOSET1 = 0x01000000;
                               // Set GPIO-1[24] Output Pin
 // Loop Test Output GPI01.24
 while(1)
                                // Loop Continue
   IOCLR1 = 0 \times 01000000;
                               // Clear Output GPI01[24]
   delay(1000000);
                                // Display Delay
   IOSET1 = 0x01000000;
                               // Set Output GPI01[24]
   delay(1000000);
                               // Display Delay
 }
}
/************************
/* Delay Time Function */
/* 1-4294967296 */
/************************
void delay(unsigned long int count1)
{
 while(count1 > 0) {count1--;} // Loop Decrease Counter
```

It is an example program of 1 blinking light at GPI01.24.

🕎 DEMO1 - 🕫 Vision3 - [Text1	1	
Eile Edit Yiew Project Debug	Fl <u>a</u> sh Peripherals <u>T</u> ools <u>S</u> VCS <u>W</u> indow <u>H</u> elp	×
12 🛥 🖬 🍠 👗 🖻 名	ユニ 倖存 4 % % % 🙀 📃 🖬 🧰 🔸	← → (12 叠 @, <mark> 国 凤</mark> ⊕ ⊕ 题 题
🗇 🏝 🎬 🔏 🙀 🎊 Targ	et 1 🔄 📩 💼	
Project Workspace 👻 👻	/ * * * * * * * * * * * * * * * * * * *	
Target 1	<pre>/* Examples Program For "CP-JR ARM7 USB-LPC2148" /* Target MCU : Philips ARM7-LPC2148 /* : K-TAL : 12.00 MHz /* : Run Speed 60.00 MHz (With PLL) /* : PLL Setup = N(5),P(2) /* : VPB Clock = CPU Clock = 60.00 MHz /* Keil Editor : uVision3 V3.03a /* Compiler : Keil CARM V2.SOa /* Function : Example LED Blink on GPI01[24] /************************************</pre>	' */ */ */ */ */ */ */ */ */ */ */ */ */ *
	<pre>// Connect P1.24 to LED For Test ON / OFF (Blink) #include "LPC214x.H" /* pototype section */ void delay(unsigned long int); int main(void)</pre>	// LPC2148 MPU Register // Delay Time Function
	<pre>{ // xxxx xxx1 xxxx xxxx xxxx xxxx xx</pre>	<pre>// Set GPIO-1[24] = Output // Set GPIO-1[24] Output Pin(OFF LED) // Loop Continue</pre>
	WITTELT	// LOOD CONCINCE
	E Text1	
put Window		·
	d ∧ Find in Files /	
	Simulation	L:9 C:55 NUM R/W

After typing C Language commands completely, we must save this File and must configure File surname to be ".C". In this case, we recommend to save by using command **File** \rightarrow **Save As**...and then configure File name and File surname as ".main.c" as in the picture below.

Save As	? 🔀
Save in: 🗀 DEMO1	▼ ← € ☆ Ⅲ•
E demo DEMO1 DEMO1 E LPC214x Startup	
File name: main.d	Save
Save as type: All Files (*.*)	Cancel

After save File as ".main.c" completely, we can see color of characters in program are changed follow by the functions such as Comment, Variable and Command. It is an advantage of Keil uVision3 that can extract and display characters follow by their functions, it makes user understand program and read program easily as in the picture below.



7. Add Files into Project File, click command Project → Components, Environment, Books..., select Tab Project Components and then select desired Add File to add into Project File.

In the first time, we must select **Files of type** to be "C Source files (*.c)" and it will display Files name that is C Language Source Code. Click icon of File named "main.c" and then select Add File named "Startup.s" into Project Files that we created.

Then we must configure new File of type to be "ASM Source files(*.s*;*.src;*.a*), it will display File name Startup.s in the blank of File name, so click icon of File "Startup.s" and then select Add File named "Startup.s" into Project Files that we created.

When we command **ADD** both File name "main.c" and "Startup.s" into Project File successfully, we must select Close to end the command Add File and it will display result of operation as in the picture below.

Components, Environment and Bo	oks	
Project Components Folders/Extensions	Books	,
Project Targets: 🖄 🗙 🗲 🗲	<u>G</u> roups: ∑ 🗙 🗲 🗲	<u>Files:</u>
Target 1	Source Group 1	
Set as Current Target	,	Add Files
		Add Liles
	OK Cancel Defau	lts Help





Example of Using Keil uVision3 to Create Project File of Keil-CARM

Add Files to Group 'Source Group 1'	? 🔀
Look in: 🗀 DEMO1 📃 🗲 🗈	📸 🎟 -
Startup	
File name: Startup	Add
Files of type: Asm Source file (*.s*; *.src; *.a*)	Close

\downarrow

Components, Environment and Bo	ooks	×
Project Components Folders/Extensions	s Books	
Project Targets: 📉 🗙 🛧 🗲	· <u>G</u> roups: M ★ ★ ↓ Files: ★ ★ ↓	
Target 1	Source Group 1 main.c Startup.s	
Set as Current Target	Add Files	
	OK Cancel Defaults Help	

After command **Add File** both "main.c" and "Startup.s" into Project File successfully, we can see both Files name are displayed in the blank of Tab of File.

8. Command to interpret the written program, click command Projects → Rebuild all target files and program Keil uVision3 will command program Keil-CARM to interpret commands instantly.

	INADANE-kaobaiN DC 24.49 LICENDEMO41main of			
A DEMOT - WYISIDITS - [C: We	TARM TEAKAC HAILPC 2146_050 DEMOT THAT			
Eile Edit View Project Debu	g 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			<u>_ 8 ×</u>
🎦 😂 🖬 🎒 🐰 🖻 🛍	♀♀ 倖佳 ∧ % % % %	• (4 → (2	🍜 🍳 🔼 💌 🕭 🚸 🕅 题	l
🔹 🏝 🎬 👗 🙀 🔊 Tarç	jet 1 🔄 🛃 🛄			
Project Workspace 👻 🗙	01 /*********************	*****		=
🖃 🔁 Target 1	02 /* Examples Program For "CP-JR	ARM7 USB-LPC2148" */		_
E Group 1	03 /* Target MCU : Philips ARM7-	LPC2148 */		
🛨 📩 main.c	04 /* : X-TAL : 12.00	MHz */		
Startup.s	05 /* : Run Speed 60.	00 MHz (With PLL) */		
	06 /* : PLL Setup = M	(5),P(2) */		
	07 /* : VPB Clock = C	PU Clock = 60.00 MHz */		
	08 /* Kell Editor : uvisions V3.0	3a */		
	10 /* Compiler : Kell CARM V2.	SUA "/		
	10 / "Function : Example LED B	**************************************		
	12 // Connect R1 24 to IED For Te	et ON / OFF (Blink)		
	12 // COMMECC F1.24 CO LED FOI TE	SC ON / OFF (DIIIR)		
	14 #include "LPC214x H"		// LDC2148 MDH Decists	-r
	15		,, mezrio mo kegioee	
	16 /* nototype section */			
	17 void delay(unsigned long int):		// Delay Time Function	n
	18		,, ,, ,, ,, ,, ,, ,, ,, ,, ,, ,, ,, ,,	
	19 int main(void)			
	20 (
	21 IODIR1 = 0x01000000;		// Set GPI0-1[24] = Ou	utput
	22 IOSET1 = 0x01000000;		// Set GPIO-1[24] Outp	out Pin(OFF LED
	23			
	24 // Loop Test Output GPI01.24			
	25 while(1)		// Loop Continue	
	, 26 _ {			
	🖹 main			
compliing main.c	·			
linking Startup	.5			
Program Size: data	=1168 const = 16 code = 444			
creating hey file	from "DFM01"			
"DEMO1" - 0 Error(s). O Warning(s).			
	l) e li el (<u>ل</u> ے
		Simulation	L:20 C:3	NUM R/W

After we command to interpret program successfully and everything is correct without any error (0 Error and 0 Warning), we will get Hex File that has names to be the same as the created Project File name and we can use this Hex File to download into MCU instantly.

An advice to Initial MCU before operation of main program is started

If we want operation of MCU is the most efficient both high speed to collect data of command and operations, we should configure default values into MCU as follows;

- Should configure PLL value to be Processor Clock (cclk) = 60 MHz, in case of using XTAL value to be 12 MHz, must configure value of M(Multiply) = 5 and P(Divide) = 2 and FFCO = 240 MHz
- Should configure **VPB Clock (pclk)** value to equal **cclk** or **60 MHz**
- Should configure MAM Timing value to be 4 Cycle of cclk (MAMTIM = 0x04)
- Should configure MAM Mode value to be Full Enable (MAMCR = 0x02)

There are 2 methods to configure default values above; firstly, writing all Code commands in written program by self and secondly, copies file Startup that has already written, in this case, we can command to Add Startup File into our new created Project file. Additionally, there are 2 methods to check and modify file Startup value; firstly, can modify Code command in file as desired and secondly, configure Startup value from window program of Keil uVision3 by self. In this case, we recommend modifying value from Keil uVision3 because it is quit convenient.

To check Startup File value

Functions of Startup File are containing commands of program for starting operation of MCU before running follow by our written program. Function of Program in Startup file is Initial operation of MCU in the important port first and then jump to run follow by the command in our written C Language Program. If we want to check the Startup value, click Tab of File Startup and then select "Expand All" and we can see the default values that are configured in Startup file as in the picture below.

🕎 LED_BLINK - #Vision3 - [C:	\Keil\ARM\Eakachai\JR-LPC2148\EXAM_KI	IL_CARM_LPC2148\LED_Blink\Star	tup.s]	∎₽×
📄 Eile Edit View Project Debug	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			_ & ×
🎦 😂 🖬 🌒 🕹 🖻 💼 🎼	コ ユ 存 存 え % % % % 🐂	→ 林 本 (y 🚳 🍳 🔼 💌 🗄 🐁 🕅	
🕸 🕮 📇 🙀 🎊 Targe	et 1 💽 🛃 🚍			
Project Workspace 👻 🗙				
🖃 🛅 Target 1	Expand All Collapse All	Help		
E Source Group 1	Option	Value		
thant.c	E-Stack Configuration (Stack Sizes in Bytes)	Value		
	- Undefined Mode	0×0000 0004		
		0x0000 0004		
	Abort Mode	0×0000 0004		
	Fast Interrupt Mode	0x0000 0004		
	- Interrupt Mode	0×0000 0080		
	User/System Mode	0×0000 0400		
	VPBDIV Setup			
		VPB Clock = CPU Clock		
	NCLKDIV: XCLK Pin	XCLK Pin = CPU Clock / 4		
	MSEL DLL Multiplier Selection			
	DSEL: PLL Multiplier Selection	3		
	MAM Setup	.		
	MAM Control	Fully Enabled		
	MAM Timing	4		
	1			
	Text Editor Configuration Wizard			
	🖹 main 📋 Startup			
<pre>x compiling main.c</pre>				
assembling Startup.	s			
linking	1100			
<pre>g rrogram Size: data= E creating hev file f</pre>	*1160 CONST=16 CODE=444 "rom "IFD BIINK"			
"LED_BLINK" - 0 Err	ror(s), 0 Warning(s).			_
Build Command	$d \lambda$ Find in Files /		1	
		Circulation	11	

Stack Configuration (Stack Sizes in Bytes)	
Undefined Mode	0x0000 0004
Supervisor Mode	0x0000 0004
Abort Mode	0x0000 0004
Fast Interrupt Mode	0x0000 0004
Interrupt Mode	0x0000 0080
User/System Mode	0x0000 0400
	✓
VPBDIV: VPB Clock	VPB Clock = CPU Clock
XCLKDIV: XCLK Pin	XCLK Pin = CPU Clock / 4
	✓
MSEL: PLL Multiplier Selection	5
PSEL: PLL Divider Selection	2
MAM Setup	✓
MAM Control	Fully Enabled
MAM Timing	4

Figure Show the method to configure Startup File for LPC2148.

An example of C Code of Keil-CARM for Initial operation of LPC2148

If we want to write program Initial operation MCU by self, we only add code command into the starting point of main Program as in the picture below.

```
// Initial PLL & VPB Clock For CP-JR ARM7 USB-LPC2148
// Start of Initial PLL for Generate Processor Clock
// PLL Configuration Setup
// X-TAL = 12 MHz
// M(Multiply) = 5
// P(Divide) = 2
// Processor Clock(cclk) = M x OSC
11
                          = 5 \times 12 \text{ MHz}
11
                          = 60 \text{ MHz}
// FCCO = cclk x 2 x P
11
       = 60 \times 2 \times P
11
       = 240 MHz
// VPB Clock(pclk) = 60 MHz
// Start of Initial PLL for Generate Processor Clock
PLLCFG \&= 0 \times E0;
                                      // Reset MSEL0:4
PLLCFG |= 0 \times 05;
                                       // MSEL(PLL Multiply) = 5
PLLCFG \&= 0 \times 9F;
                                      // Reset PSEL0:1
                                      // PSEL(PLL Devide) = 2
PLLCFG |= 0x20;
PLLCON \&= 0 \times FC;
                                      // Reset PLLC, PLLE
                                      // PLLE = 1 = Enable PLL
PLLCON |= 0 \times 01;
PLLFEED = 0xAA;
                                      // Start Update PLL Config
PLLFEED = 0 \times 55;
while (!(PLLSTAT & 0x00000400)); // Wait PLL Lock bit
PLLCON |= 0 \times 02;
                                      // PLLC = 1 (Connect PLL Clock)
                                      // Start Update PLL Config
PLLFEED = 0xAA;
PLLFEED = 0 \times 55;
VPBDIV &= 0xFC;
                                      // Reset VPBDIV
VPBDIV |= 0 \times 01;
                                      // VPB Clock(pclk) = cclk
// End of Initial PLL for Generate Processor Clock
// Start of Initial MAM Function
MAMCR = 0 \times 00;
                                      // Disable MAM Function
                                      // MAM Timing = 4 Cycle of cclk
MAMTIM = 0 \times 04;
                                      // Enable MAM = Full Function
MAMCR = 0 \times 02;
// End of Initial MAM Function
// Start of Main Function Here
```

Figure Show the sample Code for initial operation of LPC2148