

LAPIS MCU Introduction Training

LAPIS MCU Workshop Training

© 2017 ROHM Semiconductor (Thailand) Co., Ltd © 2017 LAPIS Semiconductor Co., Ltd. All rights reserved



LAPIS MCU Introduction Training

© 2017 ROHM Semiconductor (Thailand) Co., Ltd © 2017 LAPIS Semiconductor Co., Ltd. All rights reserved

Overview Tooling & S/W for training



Pichet Sriphomma pichet@es.co.th

ROHM GROUP

LAPIS

ES-ICD-V1 Reference board





ES-ICD-V1 Reference board





ES-ICD-V1 Reference board



EVA ML62Q1367 Board



Sensor Shield Board





Sensor Shield Board





Pichet Sriphomma pichet@es.co.th

ROHM GROUP

LAPIS

Connection between ES-ICD-V1 and Sensor Shield.





ROHM GROUP

LAPIS









Pichet Sriphomma pichet@es.co.th

ROHM GROUP

LAPIS





Open Program LEXIDE-U16

When clicked, the following workspace setting dialog box will be output. Set a path to workspace at [Workspace]. After that Click [Launch].







The first window after opening the program

Test - LEXIDE-U16) X
File Edit Navigate Search Project	Run LAPIS Window Help							
📑 🕶 🔙 🐚 💁 🕶 📾 🛷 🕶 🚇 🕶	■「別・別・台々・ウ・						Quick Acces	5 🖻 🔊
Project Explorer 83	E 😒 😜 🗸 🗉 🖬							
Ef Outline Sr Itil Task List An outline is not available.	¢							
		🕘 Tasks 🗯					3 P	
		0 items						
		1 Description	Resource	Path	Location	Type		
0 items selected								



Click the button in the upper right \mathbb{E} to display the [Open Perspective] dialog. Select [C/C++] and Click [OK].

Test - LEXIDE-U16									- 0 ×
File Edit Navigate Search Project	Run LAPIS Window Help								
📑 👻 📓 🕲 👻 🛄 💁 😴 🤌	・								Quick Access 🔡 😰 🔂
Project Explorer 🛛	🕒 🐄 😵 🗢 🗖								- 0
E Outline 🛛 🗐 Task List An outline is not available.	\$ v = -			Open Perspective Image: C/C++ Image: Debug Image: Debug	en Car	x			
		🖲 Tasks 💈	3						🏦 😨 🗸 🗖 🗖
		0 items	B 1.1			D		-	
		1	Description		Kesource	Path	Location	Type	



The Perspective set to [C / C ++].

V Test - LEXIDE-U16								-	x c
File Edit Source Refactor Navig	ate Search Project Run LAPIS	Window Help							
🔁 • 🗑 🕼 🖲 • 🗞 • i 💣 • 🚳	• 🖸 • 🞯 • 📴 🗐 🗑 • 1	🖻 i 🔌 i 🏘 🕶 🔕 🕶 😫 😂 🛷 🕶 i 🖗	• § • \$> \$> \$> • \$					Quick Access	12 12 12
Project Explorer	E 🕸 🖙 🗢 🗖						- 0	🚼 Outline 🕴 🗐 Task List 🛞 Build Tar	jets 🗖 🗖
								An outline is not available.	8, ∧
		🔝 Problems 🕸 🧟 Tasks 🕒 Console 🗔 Prop	perties					무미	
		Description	Resource	Path	Location	Туре			



To create a project1

From the menu, select [File] > [New] > [C/C++ Project] to open the [New C/C++ Project] dialog box.



Select [C Managed Build] and then click [Next]. The [Create C project of selected type] page will be opened.



ROHM GROUP

 ROHII

D2	—	
C Project		->
Create C project of selected type		
Project name: Project_01		
Use default location		
Location: C:\Users\pichet\lexide\Test\Project_01		Browse
Choose file system: default \vee		
Project type: To	olchains:	
✓ ➢ Executable	LAPIS U8/U16 Toolchain	
 ✓ Empty Project ✓ ⇐ Executable (LAPIS) 		
Empty Project Shared Library		
> 🦢 Static Library		
> 🗁 Makefile project		
Show project types and toolchains only if they are	e supported on the platform	
? < Back Next	> Finish	Cancel

Set a project name. Select [Executable (LAPIS)] > [Empty Project] in the [Project type] pane and select [LAPIS U8/U16 Toolchain] in the [Toolchains] pane. Click [Next].

ROHM GROUP

SEMICONDUCTOR

APIS

ROHII

D)	- 🗆 X							
Select Configurations	\rightarrow							
Select platforms and configurations you wish to deploy on								
Project type: Executable (LAPIS)								
Toolchains: LAPIS U8/U16 Toolchain								
Image: Second guardination I	Select all Deselect all							
	Advanced settings							
Use "Advanced settings" button to edit project's properties. Additional configurations can be added after project creation. Use "Manage configurations" buttons either on toolbar or on property pages.								
? < Back Next > Finish	Cancel							

Click [Next].

ROHM GROUP

LAPIS

DX			-		×
Select U8/U16 Device					
Select target device you want	to use				
Target MCUCategory:ML62Q1000 ~Group:ML62Q1300 ~LSI:ML621367 ~					
Toolchain Always use the latest Toolch Version: V2_01_00	nain				
?	Back	Next >	Finish	Cancel	

Click [Next] to open the [Select U8/U16 Device] page.

ROHM GROUP

LAPIS

SEMICONDUCTOR

ROHII



Click [Finish] to exit the project wizard and check that the project has been created.

Y Test - LEXIDE-U16		11. J. J. J.					-	X
File Edit Source Refactor Navigate Search	h Project Run LAPIS	Window Help						
🔁 🔹 📓 📓 👻 📲 🖓 🕶 🗃 🖄 🕶 🙆 🔹	• @ • 📴 🖩 🚇	• 🗉 🔌 🕸 • 🔕 • 🂁 🗁 🛷 •					Quick Access	
Project Explorer 😫							 🗄 Outline 🖾 🗐 Task List 🛞 Build Tar	gets 🗖 🗖
✓ ☑ Project_01 ✓ ☑ Includes							An outline is not available.	§9 ⊽
> B C:/U8Dev/BuildTools/V2_01_00/Inc								
> 🙆 C:/U8Dev/Inc								
✓ ▲ ML621367.ASM								
© \$\$brk_reset								
© \$\$start_up								
\$begin								
☺init_end								
©init_loop								
■ loop by byte								
near_ram_loop								
© _skip								
		🖺 Problems 🖾 🧟 Tasks 📮 Console 🖾 Pr	operties				*	
		0 items	L Martine Contraction	- Twasser		- Day an		
		Description	Resource	Path	Location	Туре		
Project_01								



add a new source file. Right-click on a project folder and select [New] > [Source File].

File	Edit	Source	Refactor	Navigate	Search	Project	R	in LAPIS	Window	Heln			
۵,	New Open I Open I	File Projects 1	from File S	ystem	Alt+Sh	nift+N >		Makefile C/C++ P Project	Project wi	h Existing C	ode		
	Close Close Save Save Save A Save A	AII As			C Ctrl+Sh (Ctrl+Sl	trl+W ift+W Ctrl+S hift+S		Convert Source F Folder Source F Header File from	to a C/C++ older ile File n Template	Project (Ad	ds C/C++ Nature	•)	
8	Move Renam Refrest Convert	ne h rt Line D	elimiters To	D		F2 F5 >		Class Task Other				Ctrl+N	
8 24 24	Print Import Export	t 			(Ctrl+P	_						
	Switch Restart Exit	t Workspa	ace		Alt+	> >							



The [New Source File] dialog box will be opened.

"main.c" is entered to the [Source file]

Select [Default C source template] for [Template].

After that, click [Finish] to add the entered source file to the project.

New Source	e File				×
Source File Create a new	source file.				C
Source folder:	Project_01			Browse	e
Source file:	main.c				
Template:	Default C source template	\sim	C	Configu	ire
?	Finish			Cance	I



Code Training Lapis - Project_01/main.c - LEXIDE-U16 File Edit Source Refactor Navigate Search Project Run LAPIS Window Help ! 🗂 ▾ 🔚 🐚 | እን ▾ 🗞 ▾ 🛗 ! & ▾ ! 🖳 ! & : 🚳 ▾ ! 🚳 ▾ ! 👌 ▾ ! 🐨 ▾ ! 🛠 ▾ () ▾ 💁 ▾ ! 🎾 🗀 🖋 ▾ ! 🗾 💀 🔜 🗐 ! 🗄 ! 🖗 ▼ ! 🖓 ▼ ! !> /> ▼ Project Explorer 🛛 🖻 main.c 🛛 ⊝/* ✓ [™] Project_01 main.c * ✓ ⋒ Includes * > 🕒 C:/U8Dev/BuildTools/V2_01_00/Inc * Created on: 24 n.w. 2563 > 🕒 C:/U8Dev/Inc Author: pichet * > 🖻 main.c */ ✓ 🗟 ML621367.ASM \$\$brk_reset \$\$error \$\$start_up \$begin _init_end □ _init_loop _init_loop2 _loop_by_byte __near_ram_loop _skip <

🖹 Problems 🧟 Tasks 📮 Console 🖾 🔲 Properties

No consoles to display at this time.



Next, create a main function entry in the created "main.c".







Right-click on a project folder and select [Build Project] to start the build process.







When the build succeeds , an ABS file is generated.







ROHM GROUP

LAPIS



😼 DTU8 - main.c		
File Edit View Run	Tool Window Help	Click Pup program
	0 P P 🖺 📰 💷 🕹 🖉 🖬 🖩 🖬 🖬 🖬 🔢 P 4	
📲 main.		
Line B <mark>L C</mark> Addre	-55	
00001	/*	
00002	* main.c	
00003	*	
00004	 Created on: 24 j.%. 2563 Author: picket 	
00006	*/	
00007		
00008	void main(void)	
00009	{	
00010 <u>0:00</u>	30H while(1)	
00011	(
00012	1	
00014 0 0:00E	18H }	
00015		
,		Back to
		Overview
		Overview



S-TERM_v30_setup Double Click File "S-TERM_v30_setup"

InstallShield Wizard	
	Preparing to Install
0	S-TERM v3.0 Setup is preparing the InstallShield Wizard, which will guide you through the program setup process. Please wait.
CONTROL OF	Decompressing: S-TERM v3.0.msi
2	
	Cancel

After that Appear new window







🔀 S-TERM v3.0 - InstallShield Wizard			×
Customer Information			
Please enter your information.			
<u>U</u> ser Name:			
pichet			
Organization:		_	
InstallShield			
	< Back	Next >	Cancel

Click Next >



🟀 S-TERM v3.0 - InstallShield Wizard			×
Ready to Install the Program The wizard is ready to begin installation.			2
Click Install to begin the installation. If you want to review or change any of yo	our installation set	tings, click Back. Click	Cancel to exit
the wizard.			
InstallShield	< Back		Cancel

Click Install



Installing	S-TERM v3.0		4	
The prog	ram features you selected are being installed.			
P	Please wait while the InstallShield Wizard installs S-TERM v3.0. several minutes.	. This mi	ay take	
	Status:			
	Writing system registry values			
actallChield				
nstallShield -				

Wait for Install



Click Finish

© 2017 ROHM Semiconductor (Thailand) Co., Ltd. © 2017 LAPIS Semiconductor Co., Ltd. All rights reserved






Before launching the program S-TERM. Must connect device. Open Bluetooth & other devices





Click Add Bluetooth or other device





Click Bluetooth and select device



S-TERM



Wait for connecting..







Show device connected







Open Program S-TERM



File Mode Option
Comport : 💷 Baudrate : 1200 v Data bt : 8 v Stop bt : 1 v Party : None v Open port Close port
Rs: 0 Reset View : Ascii v New line at : CR Send type : Ascii v Send type : Send type : Ascii v Send type : Sen
Console
For clear console Press key 'F11' in console or press button 'Clear console' >> Clear
Data send : Send Data





💒 S-TERM v3.0
File Mode Option
Comport : COM7 V Baudrate : 1200 V
Rx: 0 Reset Tx: 0 Reset View: Asci
Console

Select Comport

💒 S-TERM v3.0				
File Mode Option				
Comport : COM7 🗸	Baudrate : 120		Data bit: 8	~
Rx: 0 Reset Tx:	0 Re 960 192 384	00 00 200 400	Ascii ~	New line
Console	576 115 128 230 460	600 6200 8000 9400 9800		

Select Baudrate. This workshop select 9600.





Baudrate : 1200	∽ Data bit : 8	✓ Stop bit : 1
0 Reset View :	Ascii 🗸	New line at : CR 🗸
1	Ascii	
	Hex	
	Ascii + Hex	

💒 S-TERM v3.0	
File Mode Option	
🔄 Open port F2	
Close port F3	Baudrate : 1200 V Data
Send file F4	
Save to file Ctrl+S	0 Reset View : Ascii
Exit Ctrl+End	
Console	40

Select view type. This workshop select Ascii.

Click File and select Open port.



Using the application

Step 1 Connect MCU Board with Bluetooth



© 2017 ROHM Semiconductor (Thailand) Co., Ltd. © 2017 LAPIS Semiconductor Co., Ltd. All rights reserved



Using the application

Step 2 Open application and click "Bluetooth Check"





Using the application

Step 3 Choose device and connect





Using the application

Step 4 After connected this app Show button to choose





LAPIS MCU Workshop Training

© 2017 ROHM Semiconductor (Thailand) Co., Ltd © 2017 LAPIS Semiconductor Co., Ltd. All rights reserved















1. GPIO

© 2017 ROHM Semiconductor (Thailand) Co., Ltd. © 2017 LAPIS Semiconductor Co., Ltd. All rights reserved





Open Program LEXIDE-U16

When clicked, the following workspace setting dialog box will be output. Set a path to workspace at [Workspace]. After that Click [Launch].







The first window after opening the program

Test - LEXIDE-U16							- 0 ×
File Edit Navigate Search Project	Run LAPIS Window Help						
📑 🕶 🔙 🐚 💁 🕶 📾 🛷 🕶 🚇 🕶	■創ま剤を行かるものを						Quick Access 🛛 🛃 🔈
Project Explorer SS	😑 😢 😫 🔍 😁 🖬						
E Outline 131 El Tak Lat An outline is not available.	ν ** α						
		Tasks 22					3 D V - D
		0 items					
		1 Description	Resource	Path	Location	Туре	
0 items selected							



Click the button in the upper right \mathbb{E} to display the [Open Perspective] dialog. Select [C/C++] and Click [OK].

🔰 Test - LEXIDE-U16									- 0	\times
File Edit Navigate Search Project Run	LAPIS Window Help									
📑 🕶 🔛 🔞 🕶 💷 💁 🕶 😂 🛷 🕶 🖗	▼ { ▼ \$\$ \$\$ \$\$ \$ ▼ \$\$ \$								Quick Access	🖻 陆
Project Explorer 😫	= 😫 😜 🗢 🗖									- 0
SE Outline ⅔ Task List An outline is not available.				Image: C/C++ Image: C/C+++		×				
		🖲 Tasks 🛛							· · · · · · · · · · · · · · · · · · ·	~ - 8
		0 items			-			-		
		1	Description		Resource	Path	Location	Туре		



The Perspective set to [C / C ++].

Test - LEXIDE-U16								1 × 1
File Edit Source Refactor Naviga	ate Search Project Run LAPIS	Window Help						
🔁 • 🗑 🕼 🛛 • 🗞 • 🗃 • 🚳	• 🖻 • 🞯 • 📴 🗐 🗑 •	🖳 🔌 🌴 🕶 🔕 🕶 😫 🗁 🛷 🗣	• [劉•杨•��				Quick Access	12 10 10
C + D + D + C + C + C + C + C + C + C +		▣ ::	•} ॒ ि • ि • • • • • • • • • •				Quick Access	ten and a second secon
		Problems 2 Tasks Console [O items Description	Properties Resource	Path	Location	Туре	© ∉	



Import Project

DX 1	est -	LEXIDE-U16	5								
File	Ed	t Source	Refactor	Navigate	Search	Project	Run	LAPIS	Window	Help	
: 📬	• 🔒	00.	« •) 🗕 📮 🕅	R 🔯 🔻	🚳 🔻 🖸	- ©	• P	II 🛊	- 0 -	- 💁 -
🔁 P	roje	t Explorer	3								
			□ 🕏	₽ ⊽							
		New						>			
	\geq	Import									
	4	Export									
	\$	Refresh					F	5			

Right-click on project Explorer and select Import.

Import Project

Milmport – 🗆	×
Select Create new projects from an archive file or directory.	5
Select an import wizard:	
type filter text	
 General Archive File Existing Projects into Workspace File System Preferences Projects from Folder or Archive C/C++ Install LAPIS 	~
? < Back Next > Finish Cancel	

Select General and choose Existing Projects into Workspace. Click Next.





Import Project

🔰 Import	—	
Import Projects		
Select a directory to search for existing Eclipse project	ts.	
Select root directory:	~	Browse
O Select archive file:	~	Browse
Projects:		
		Select All
		Deselect All
		Refresh
0 · /		
Search for nested projects		
Copy projects into workspace		
Hide projects that already exist in the workspace		
Working sets		
Add project to working sets		New
Working sets:	\sim	Select
2 A Back Novt	The balls	Consel
S DdCK INPXL3	FINISN	Cancel

LEXIDE up the new window. Click Browse.. at Select root directory. Choose "Chapter 1 GPIO" in Folder window.

Browse For Folder	×
Select root directory of the projects to import	
🕞 Contacts	^
> 🔜 Desktop	
> 🔄 Documents	
> 🖊 Downloads	
> 📙 Favorites	
✓	
🗸 📙 Code Training Lapis	
> 📙 .metadata	
Chapter 1 GPIO	
settings	
> Debug	
source files	
Chapter 2 UART	
Chapter 3 ADC	
Chapter 4 I2C	
	~
Folder: Chapter 1 GPIO	
Make New Folder OK Ca	ncel



Import Project

M Import			—	
Import Projects Select a directory to sea	irch for existing E	clipse projects.		
 Select root directory: Select archive file: Projects: 	C:\Users\pichet	\lexide\Code Train	ing Lapis\Chapter ~ ~	Browse Browse
Chapter 1 GPIO (C	C:\Users\pichet\le	xide\Code Trainin	g Lapis\Chapter 1 GP	Select All Deselect All Refresh
 Options Search for nested pro Copy projects into w Hide projects that all Working sets 	ojects orkspace ready exist in the	workspace	>	
Add project to work	ring sets			New Select
?	< Back	Next >	Finish	Cancel

After choosing Project Click Finish.



Import Project



Then appear the project on Project Explorer.



Check Device

🚺 Code Training Lap	is - LEXIDE-U16
File Edit Source I	Refactor Navigate Search Project Run LAPIS Window He
🖻 🕶 🔚 🕼 💌 🕶	≪ ▼ 🗟 🗃 ▼ 😂 ▼ 🖻 ▼ 🞯 ▼ 🕸 ▼ 🔿 ▼ 🗛 ▼ 🎥 🗁 🔗
Project Explorer ≈ → 55 Chapter 1 GPIO	
> 🔊 Includes > 🗁 Debug	Go Into
 Source file BlinkLE 	Open in New Window Show in Local Terminal
b BlinkLE irq_ter b irq.h c main.c b rdwr_r b reset_c c c reset_c	Copy Ctrl+C Paste Ctrl+V Delete Delete Remove from Context Ctrl+Alt+Shift+Down Source > Move F2
> 🖻 sysdefi 🗎 > 🖻 SysFlag	a Import a Export
 > c time_d > h time_d > c timer0 > h Timer0 > h Timer0 > c VLS.c > h VLS.h 	Clean and Build Project Build Project Clean Project Refresh F5 Close Project Close Unrelated Projects
> 🖻 wdt.c > 🖻 wdt.h > 🗟 ML62136	Build Configurations > ole Build Targets > pIC Index >
	Show in Remote Systems view Run As Debug As Profile As Team Compare With
对	Restore from Local History Run C/C++ Code Analysis Configure
	Properties Alt+Enter

Right-click on a project folder and select [Properties] .



Select Device

Properties for Chapter	i GPIO —	
type filter text	LEXIDE-U16 Settings	↓ ↓ ↓
 > Resource Builders > C/C++ Build Build Variables Environment Logging Settings Tool Chain Editor > C/C++ General LEXIDE-U16 Settings Project References Run/Debug Settings > Task Repository WikiText 	Target MCU Category: ML62Q1300 LS: ML621367 Toolchain Always use the latest Toolchain Version: V2_01_00 Version: V2_01_00	Apply
?	Apply and Close	Cancel

Choose LEXIDE-U16 Settings

Build Project

🔰 Code Training	, Lap	ois - LEXII	DE-U16							
File Edit Sourc	ce	Refactor	Navigate	Search	Project	Run	LAPIS	Window	v Help	
📑 🗕 🖪 👘 🛛 😵	•	% - 🗟	1 💣 🔻 🚳	• 🖻 •	G -	\$ - () - 9	- 🥭	🖨 🛷 👻 🗌	9
 Project Explorer 2 Chapter 1 CDU Include Pebug 	er S GDIC	New Go Into		E	3 4 3 \$*			>		
> 🗁 source > 📓 ML621		Open in Show in	Open in New Window Show in Local Terminal							
		Copy Paste Delete Remove Source Move Rename	from Cont	C	trl+Alt-					
	24 24	Import Export								
		Clean ar	_							
£) **	\$	Build Pro Clean Pr Refresh Close Pro Close Ur	oject roject roject nrelated Pro	ojects				F5		
		Build Co Build Tar Index	onfiguration rgets	ns				> > >		
		Show in Run As Debug A Profile A	Remote Sy As As	ystems vie	w			> > >		
	*	Team Compare Restore Run C/C	e With from Local ++ Code A	l History Inalysis				>	ion	Tas
		Properti	es				Alt+	Enter		
	_								-	

Right-click on a project folder and select [Build Project] to start the build process.



Build Project



When the build succeeds , an ABS file is generated.

© 2017 ROHM Semiconductor (Thailand) Co., Ltd. © 2017 LAPIS Semiconductor Co., Ltd. All rights reserved





CPU

- 16-bit RISC CPU (CPU name: nX-U16/100)
- Instruction system: 16-bit length instruction
- On-chip debug function built-in (supported by LAPIS on-chip debug emulator EASE1000)
- ISP (In-System Programming) function built-in

Operating voltage and temperature

- Operating voltage: VDD = 1.6 to 5.5 V
- Operating temperature: -40 to +105 °C

Clock Generation Circuit

- Low-speed clock
- Internal low-speed RC oscillation: Approx.32.768 kHz
- High-speed clock
- PLL oscillation: 24MHz/16MHz is selectable by code option



Table 1-1 ML62Q1300 Group Product List

Drogram	Dete	Dete	16pin	20pin	24pin	32pin
Program	Dala	Flash	SSOP16	TSSOP20	WQFN24	TQFP32
memory	memory		WQFN16			WQFN32
64Kbyte			—	—	ML62Q1347	ML62Q1367
48Kbyte	4Kbyte		—	—	ML62Q1346	ML62Q1366
32Kbyte			—	—	ML62Q1345	ML62Q1365
32Kbyte		ZKbyle	ML62Q1325	ML62Q1335	—	—
24Kbyte	2Kbyte		ML62Q1324	ML62Q1334	—	—
16Kbyte			ML62Q1323	ML62Q1333	—	_

INTRODUCTION : LAPIS ML62Q1367



	Table 1-4 Main Function List																			
	Pin			Inte	Interrupt Timer			Serial			Analog									
Part number	Total pin-counts	Power pin counts	Reset Input pinInput pin	I/O port	LED drive port (shared with the I/O port)	Internal interrupt [source]	External interrupt [port]	Functional Timer [ch]	16bit General I Timer*1 [ch]	Full-duplex UART or Synchronous serial*2 [ch]	I ² C bus unit (Master/Slave) [ch]	I ² C bus interface (Master only) [ch]	10bit Successive type A/D converter [ch]	Analog comparator [ch]	Analog comparator [input pin]	8bit D/A converter [ch]				
ML62Q1323																				
ML62Q1324	16			12	11								6							
ML62Q1325										23			4							0
ML62Q1333		20 3 1 24 32				20			-							Ŭ				
ML62Q1334	20			16	15															
ML62Q1335			1				R	4		2	1	1		1	2					
ML62Q1345			'				Ŭ	-		~	'	'		'	~					
ML62Q1346	24			20	19								8							
ML62Q1347						25			6							1				
ML62Q1365						20			0							'				
ML62Q1366	32			ĺ					28	27										
ML62Q1367																				

.

-

...

. . .

Pichet Sriphomma pichet@es.co.th

© 2017 ROHM Semiconductor (Thailand) Co., Ltd. © 2017 LAPIS Semiconductor Co., Ltd. All rights reserved P. 67

INTRODUCTION : LAPIS ML62Q1367







1.1 SYSTEM CLOCK



Clock Generation Circuit

Table 6-1 Clocks generated by the clock generation circuit

Clock Name	Symbol	Description
Low-speed clock	LSCLK	Low speed clock for peripherals (32.768kHz)
Simplified RTC clock*1	RTCCLK	Low speed clock for the simplified RTC (32.768kHz)
High-speed clock	HSCLK	High speed clock for peripherals (Max. 24MHz)
System clock	SYSTEMCLK	CPU operating clock (32.768kHz or Max. 24MHz) The maximum frequency depends on the CPU operation mode(See Table 6-2)
Low-speed output clock	OUTLSCLK	Low speed output from a general port (32.768kHz)
High-speed output clock	OUTHSCLK	High speed output from an general port (Max. 12MHz)
WDT clock	WDTCLK	Clock for the watch dog timer (1.024kHz)

*¹ Available on the ML62Q1500 and ML62Q1700 group



Clock Generation Circuit



Pichet Sriphomma pichet@es.co.th

P. 71
Code Options (ML621367.ASM)



Updated at power-on detection reset,

or any other system reset

ML62Q1300	grou	р
-----------	------	---

Program Product name memory space Code			Address									
Product name	memory space size	Code Option area	CODEOP2	CODEOP1	CODEOP0							
ML62Q1323/1333	16K byte	0x0:3FC0 to 0x0:3FFF	0x0:3FD4	0x0:3FD2	0x0:3FD0							
ML62Q1324/1334	24K byte	0x0:5FC0 to 0x0:5FFF	0x0:5FD4	0x0:5FD2	0x0:5FD0							
ML62Q1325/1335/1345 /1365	32K byte	0x0:7FC0 to 0x0:7FFF	0x0:7FD4	0x0:7FD2	0x0:7FD0							
ML62Q1346/1366	48K byte	0x0:BFC0 to 0x0:BFFF	0x0:BFD4	0x0:BFD2	0x0:BFD0							
ML62Q1347/1367	64K byte	0x0:FFC0 to 0x0:FFFF	0x0:FFD4	0x0:FFD2	0x0:FFD0							



Code Options (ML621367.ASM)

26.2.1 Code Options 0 (CODEOP0)

This is the symbol assigned to address in the code option area of the program memory space (different from the special function registers (SFR)).





WDT



Code Options (ML621367.ASM)

Bit No.	Bit symbol name	Description
15 to 13	-	Reserved bits
12	PCERMD	 This bit is used to choose to enable/disable the unused ROM area access reset. See Chapter 29.3.2 "Unused ROM Area Access Reset Function" for the unused ROM area access reset. 0: Unused ROM area access reset disabled 1: Unused ROM area access reset enabled (initial value)
11 to 9	-	Reserved bits
8	REMAPMD	This bit is used to choose to enable/disable the remapping function (software remap or hardware remap) operation.
		See Chapter 2.7 "Remapping Function" for details of the remapping function.
		0: Remapping function operation enabled
		1: Remapping function operation disabled (initial value)
8	REMAPMD	This bit is used to choose to enable/disable the remapping function (software remap or hardware remap) operation.
		See Chapter 2.7 "Remapping Function" for details of the remapping function.
		0: Remapping function operation enabled
		1: Remapping function operation disabled (initial value)
7 to 3	-	Reserved bits
2	WDTNMCK	This bit is used to choose the watchdog timer (WDT) operation clock.
		0: Clock with divided frequency (1.024 kHz) of low-speed oscillation clock (32.768 kHz)
		1: Watchdog timer RC1K oscillation clock (initial value)
		See Chapter 10 "Watchdog Timer" for details of the watchdog timer.
1	WDTSPMD	Set this bit to "0".
0	WDTMD	 This bit is used to choose to enable/disable the watchdog timer (WDT) operation. 0: WDT operation disabled 1: WDT operation enabled (initial value)



Code Options (ML621367.ASM)

26.2.2 Code Options 1 (CODEOP1)

This is the symbol assigned to address in the code option area of the program memory space (different from the special function registers (SFR)).

Address:(See Table 26-1)Access:R/WAccess size:16 bitsInitial value:0xFFFF (factory default setting for products with blank flash memory)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word								COD	EOP1							
byte					-								-			
Bit	-	-	-	-	-	-	-	-	-	-	-	-	PLLMD 1	PLLMD 0	CPUM D1	CPUM D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1



Code Options (ML621367.ASM)

3 2	PLI MD1	These bits are	used to choose the	PIL reference frequency										
0, 2	PLLMD0	00: Do not	use											
		01: Do not	use											
	_		ference frequency="	24 MHz										
		10: T LL TO	ference frequency=	16 MHz (initial value)										
			table shows the rela	ation between the PLL referen	ace frequency and the									
		maximum ope	rating frequency of	CPU and peripheral circuits	ice frequency and the									
		PLL	stating nequency of	Maximum operating from										
		reference		Maximum operating nequ	lency									
		frequency	Peripheral circuit	CPU (wait mode)	CPU (no-wait mode)									
		24MHz	24MHz	24MHz	6MHz									
		16MHz	16MHz	16MHz	8MHz									
		See Chapter :	2 "CPU and Memory	/ Space" and Appendix C "Ins	truction Execution Cycle" fo									
		the CPU oper	ation modes (wait m	node and no-wait mode).										
1, 0	CPUMD1,	These bits are	e used to choose the	e CPU operation mode.										
	CPUMD0	00: Prohib	00: Prohibited to use (wait mode)											
		01: Wait m	01: Wait mode											
		10: Prohib	ited to use (no-wait	mode)										
		11: No-wa	it mode (initial value											

Code Options (ML621367.ASM)

cseg #0 dw	at OffcOh Offffh	; add : 0ff	ress c0h				CODEOP1	CODEOP0
dw dw	0ffffh 0ffffh	; 0ff	c2h			Address	FFD2	FFD0
dw	Øffffh	; 0ff	c6h					
dw	0ffffh	; 0ff	c8h			Value	FFF9	FFF9
dw	0ffffh	; 0ff	cah					
dw	0ffffh	; 0ff	cch					
dw	0ffffh	; 0ff	ceh					
cseg #0	at 0ffd0h	; add	ress					
wt	0fff9h	; 0ff	d0h	;Disable WDT in	STOP mo	de		
cseg #0	at 0ffd2h	; add	ress					
dw	0fff9h	; 0ff	d2h	;PLL=24 MHz and	CPU wai	t mode		
dw	0ffffh	; 0ff	d4h					
dw	0ffffh	; 0ff	d6h					
dw	Øffffh	; 0ff	d8h					
dw	Øffffh	; 0ff	dah					
dw	0ffffh	; 0ff	dch					
dw	0ffffh	; 0ff	deh					

Back to Clock Generation Circuit



Clock Generation Circuit (main.c)

6.2.4 Clock Control Register (FCON)

FCON is a specific function register (SFR) to control the clock generation circuit and choose the system clock.





Clock Generation Circuit (main.c)

6.2.2 High-Speed Clock Mode Register (FHCKMOD)

FHCKMOD is a specific function register (SFR) to choose the oscillation mode of the high-speed clock oscillation circuit (PLL oscillation circuit) and the frequency of high-speed clock.





Clock Generation Circuit (main.c)





Clock Generation Circuit (main.c)





Clock Generation Circuit (main.c)

6.2.4 Clock Control Register (FCON)

FCON is a specific function register (SFR) to control the clock generation circuit and choose the system clock.

void {	InitSystemClock(void) unsigned int i;	Addre Acces Acces Initial	ss: s: s size: value:	0×F00 R/W 8 bits 0×00	6													
	NOSC = 0; // Disable the high spe		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	(/	Word									-			50				
	Set VIS Startun():	Byte					-							FU			510	0510
		Bit	-	-	-	-	-	-	-	-	LPLL	-	-	-	-	-	SC	CLK
H	OSCM0 = 0;	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
S C F	YSC2=0;SYSC1=0;SYSC0=0; UTC2=1;OUTC1=0;OUTC0=0; HUT0=0;	Initial value	0	0	0	0	0	0	0	0	0	0	0	0	D	0	0	0
E	NOSC = 1; //	Bit No.	Bi	t symbo name	bl						Des	scription	n					
j	= 200; // Wait for OSC. is stable.	15 to 8	-		F	Reserve	ed bit											
V	hile(i)	7	LPLI	L	1	his bit PLL h	indicate as the re	es that ead-on	the frec Iv attrib	quency oute.	of the F	PLL osc	illation	is with	in the t	arget e	rror. Th	e
	wat_clear();					0:	The fre	quency d (Initia	of PLI	L oscil	lation is	out of	the tar	get err	or or th	e PLL	oscillat	ion is
5	ELSCLK = 1; // System Clock = HSCLK					1: 1	The free	uency	of PLL	oscilla	ition is v	ithin th	e targe	t error				
-	_asm(" <u>nop</u> ");	6 to 2	-		F	Reserve	ed bit											
-	_asm("nop");	1	ENC	sc	0	nis dit scillati	is usea on circu	ito ena iit.	ible/stai	rt or al	sable/st	op the (oscillati	on of tr	ne nign	-speed	CIOCK	
-	_asm("nop");					0: 1	Disable	Stop th	ne high	-speed	l clock o	scillatio	n (Initia	al value	e)			
– ۱		0	SEL	SCLK	1	1: 1 his bit	is used	to cho	e nign-s ose the	speed syste	m clock os	cillation	i (initia	i value;)			
,		-			v f	Vhen ti ixed to 0: 1	"0" and SCLK	speed the low (Initial)	genera w-speed value)	ation ci d clock	(LSCLI	topped () is ch	(ENOS osen fo	SC bit = or the s	= "0"), t ystem (he SEL clock.	SCLK I	bit is
						1. 1	iigii-sp	000 010	A CHU	Join by	aic 01.	502 10	01000	/ .//				

ROHM GROUP ROHII LAPIS SEMICONDUCTOR

Clock Generation Circuit (main.c)



6.2.4 Clock Control Register (FCON)



1.2 Timer 16-Bit

Timer 16-Bit



Pichet Sriphomma pichet@es.co.th

Timer 16-Bit



Figure 8-3 Repeat Mode Operation Timing in 16-Bit Timer Mode

Pichet Sriphomma pichet@es.co.th



Timer 16-Bit

TMHn Interrupt Frequency = 4 kHz = 0.25 mS

TMHn Interrupt Frequency = $\frac{\text{HSCLK}}{\text{TMH0D+1}}$

```
4 \text{ kHz} = 24 \text{ MHz/TMH0D} + 1
```

TMH0D = 5999



nitial Timer 16-Rit	8.2.4 16	-Bit T	imer	n Mo	de Re	egister	r (TM	HnMO	DD: n	= 0 t	o 7)						
	TMH	InMOI	O(n = 0)	0 to 7) i	is a spe	cific fu	nction r	register	(SFR)	to con	trol the	operati	on mod	le of 10	5-bit tin	ier.	
(timer0.c)	Addr	ess:		0xF320 0xF322 0xF324	(TMH0 (TMH1 (TMH2	MODL/T MODL/T MODL/T	- MH0M MH1M - MH2M	OD), 0 OD), 0 OD), 0	xF321(xF323(xF325(MODH), MODH), MODH),	,					
TH0CS=1; TH0EX=0;				0xF326 0xF328 0xF32A 0xF32C 0xF32C 0xF32E	(TMH3) (TMH4) (TMH5 (TMH6 (TMH7	NODL/T NODL/T MODL/T MODL/ ⁻ MODL/T	-MH3M -MH4M -MH5M -MH5M 	OD), 0 OD), 0 IOD), 0 IOD), 0 IOD), 0	xF327([*] xF329([*] xF32B(xF32D(xF32D(xF32F(ГМН3М ГМН4М ТМН5 ТМН6 ТМН6	MODH) MODH) MODH) MODH) MODH) MODH)	, ,),					
TH0EXS=0; TH0DIV2=0;TH0DIV1=0;TH0DIV0=0;	Acce Acce Initia	ss: ss size l value:	F 6: 6	R/W B/16 bit Dx0000				,-			,						
-		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TH08BM = 0;	Word								TMHn	MOD							
THOOST = 0;	Byte				TMHn	MODH							TMHn	MODL			
THONEG = 0;	Bit	-	-	-	-	-	THn NEG	THn OST	THn 8BM	-	THn DIV2	THn DIV1	THn DIV0	THn EXS	THn EX	-	THn CS
TMH0D = 5999:	R/W	R	R	R	R	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R	R/W
	Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ETM0 = 1; // Timer 0 interrupt	en					clock	choser	h by the	• THnC	S bit.							
THORUN = 1; // Start Counting	1	-			Rese	rved bi	t	-									
	0	ТН	nCS		This 0: 1:	bit is us LSCLK HSCL	ed to o (initia <	choose I value	the tin)	ner clo	ock of th	ne 16-b	oit time	r n.			
	2	TH	nEX		This b 0: 1:	it is use The tim chosen The tim	ed to ch er is co by the er is co	oose ti ounted THnDl ounted	ne cour by the o V2 to 0 by the i	it clock clock c bit. (ir ising e	k (THn(hosen hitial va edge of	CK) of t by the lue) the ext	he 16-b THnCS ternal tr	bit time bit and igger s	r n. d divide ignal de	d by ti etecte	ne ratio d by the



Timer 16-E	Bit	6 to 4	THnDIV2 to	These bits are used to choose frequency dividing ratio for the count clock in the 16-bit timer
(timer0.c)				000: No dividing (initial value) 001: 1/2 of the timer clock
TH0CS=1; TH0EX=0;				010: 1/4 of the timer clock 011: 1/8 of the timer clock 100: 1/16 of the timer clock 101: 1/32 of the timer clock
THOEXS=0;				110: 1/64 of the timer clock 111: 1/128 of the timer clock
TH08BM = 0;	IVI-0, MODIVO -0,	3	THnEXS	This bit is used to choose the external trigger supplied as the count clock of the 16-bit timer n. 0: EXTRG0 (initial value) 1: EXTRG1
THOOST = 0; $THONEG = 0;$		10	THnNEG	 This bit is used to choose the output polarity of timer out (TMHnOUT). 0: Positive logic (initial level is "L") (initial value)
TMH0D = 5999;	// Timon Q into	9	THnOST	This bit is used to choose the operation mode of the 16-bit timer n. 0: Repeat timer mode (initial value)
THORUN = 1;	// Start Counti	8	THn8BM	1: One-shot timer mode This bit is used to choose whether the timer works as one 16-bit timer or two channels of 8-bit timer
			-	0: 16-bit timer mode (initial value) 1: 8-bit timer mode



Timer 16-Bit	8.2.2 10	6-Bit T	imer	n Data	a Reg	gister	(TMF	-InD: ι	n = 0	to 7)							
(timer0.c)	TM regi In ti	HnD (n ster (TM he 8-bit	= 0 to MHnC). timer n	7) is a s node, T	pecific MHnD	function	on regis 0 to 5)	ster (SF is com	R) to s	et the c o TMH:	ompari nCL (n	son val $= 0$ to	ue with 5) and	n the 16 TMHn	5-bit tin DH (n [;]	ier n co = 0 to 5	ounter
TH0CS=1;	con	pared t	o TMH	nCH (n	= 0 to	5).											
TH0EX =0;	Ado	lress:	0xF3 0xF3	300(TMI 304(TMI	H0DL/1 H2DL/1	MHOD), 0xF3), 0xF3	01(TMH 05(TMH	HODH), H2DH),	0xF302 0xF306	2(TMH ⁻ 6(TMH:	1DL/TN 3DL/TN	1H1D), 1H3D),	0xF303 0xF307	3(TMH1 7(TMH3	DH) DH)	
TH0EXS=0;			0xF3 0xF3	308(TM) 30C(TM	H4DL/1 H6DL/ ⁻	TMH4D TMH6D), 0xF3)), 0xF3	09(TMH 30D(TM	44DH), 1H6DH)	0xF30/		5DL/TN 17DL/TI	4H5D), MH7D),	0xF30	3(TMH5 F(TMH	5DH) 7DH)	
TH0DIV2=0;TH0DIV1=0;TH0DIV0=0;	Acc Acc	ess: ess size	R/W e: 8/16	bit			,.	,	,	,	,		,,			,	
TH08BM = 0;	Initi	al value	: OxFF	FF													
TH00ST = 0;		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
THONEG = 0;	Word								TM	HnD							
	Byte				ТМН	InDH							TMH	InDL			
TMH0D = 5999;	Bit	THnD 15	THnD 14	THnD 13	THnD 12	THnD 11	THnD 10	THnD 9	THnD 8	THnD 7	THnD 6	THnD 5	THnD 4	THnD 3	THnD 2	THnD 1	THnl 0
ETM0 = 1; // Timer 0 int	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
THORUN = 1; // Start Count	Initial	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	value																

Pichet Sriphomma pichet@es.co.th



Timer 16-Bit	5.2.3 Int	errup	t Ena	ble R	egiste	er 23	(IE23))							-		-
(timer0.c)	IE23 The l After	is a spe oits are the int	ecific fi unwrit errupt	unction eable w is accep	registe hen the oted, the	r (SFR) e produ e maste) to enal cts do n r interru	ble or ot hav upt ena	disable ve the pe able flag	the int cripher g (MIE	errupt fo al circui) of the	or eacl ts and CPU i	n interru they re is reset	ipt requ turn "0 to "0", 1	iest. " for re howeve	ading. er, the a	applicable
THØCS=1:	each	flag of	1E01 19	s not res	set and	remain	s "1".										
TH0EX=0;	Addr Acce	ess: ss:	0 R	xF022(2/W	(IE2/IE2	23), OxF	023(IE	3)									
THØEXS=0:	Acce Initia	ss size: value:	8 0	/16bit x0000													
THODIV2=0; THODIV1=0; THODIV0=0;		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TH088M - 0:	Word								IE	23							
THOOST - 0:	Byte				IE	3							IE	2			
THONEG = 0;	Bit	ETM1	ETM0	EFTM 1	EFTM 0	EI2C M1	EI2C M0	-	EEXT X	-	ESAD	-	ESIU 01	ESIU 00	EMC S	EDM A	ECBU
TMH0D = 5999;	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R	R/W	R	R/W	R/W	R/W	R/W	R/W
ETM0 = 1; Timer 0 int	Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
THØRUN = 1; // Start Count	14	ETM	0	Т	his bit	control	s to ena	able o	r disable	e the 1	6bit Tim	ner 0 in	nterrupt	(TM0II	NT).		
					0: [Disable	the inte	errupt	(initial v	alue)							
					1: E	Enable	the inte	rrupt									



Timer 16-Bit	8.2.7 16	-Bit T	imer	Start	Regis	ster (1	MHS	TR)									
(timer0.c)	TMF TMF TMF TMF	ISTR is ISTRL ISTRH ISTRI	s a spec is used is used	tific function in the to stant	nction r 16-bit t rt count	egister timer m ing the	(SFR) iode. upper	to conti side 8b	rol to st it count	tart cou ter in the	nting tl ne 8-bit le 8-bit	ie 16-b timer i	it timer node.	n.			
TH0CS=1; TH0EX=0;	TMF	ISTR is	a writ	e-only 850	register	and re	turns a	lways "	0x0000)"for re	ading.		lioue.				
TH0EXS=0; TH0DIV2=0;TH0DIV1=0;TH0DIV0=0;	Acce Acce Initia	ess: ess size I value:	vv : 8/16 0x00	bit 00													
TH08BM = 0; TH00ST = 0;	Word Byte	15	14	13	12 TMHS	11 STRH	10	9	8 TMH	7 STR	6	5	4 TMHS	3 STRL	2	1	0
THONEG = 0;	Bit	TH7H RUN	TH6H RUN	TH5H RUN	TH4H RUN	TH3H RUN	TH2H RUN	TH1H RUN	TH0H RUN	TH7 RUN	TH6 RUN	TH5 RUN	TH4 RUN	TH3 RUN	TH2 RUN	TH1 RUN	TH0 RUN
ETM0 = 1; // Timer 0 int	R/W Initial value	w o	W O	W O	w o	W O	W O	W O	W O	W O	w o	W O	w o	W O	W O	w o	W O
THORUN = 1; tart Count:	In THORUN In the 16bit timer mode, controls the counter of 16bit timer 0 In the 8bit timer mode, controls the lower side 8bit counter of 16bit timer 2 Writing "0": Invalid																
	Writing "1": Start counting																



Timer Interrupt (timer0.c) void Timer0_Interrupt(void) { static unsigned char tick_cnt=0; static unsigned int t_base_cnt=0; //-----//4000 kHz -> 0.25 mS //count to 16 -> (4000 Hz/16) = 250 Hz //250 Hz -> 4 mS if(++tick cnt >= 16) // Count for 250Hz (4000 { tick cnt = 0;Flag._ScanTime=1; } //----- only test -----11 t base cnt++; if(t_base_cnt >= (4000*10)) //if(t base cnt >= 4000) t base cnt = 0;BlinkLED(); }

Timer 16-Bit -> 4 kHz

-> 0.25 mS

Count to 16 -> $\frac{4000\text{Hz}}{16}$ =250Hz Timer Interrupt -> 250 Hz

->4 mS

Real Time Clock (timer0.c)





1.2 GPIO Port

Initial Port (main.c)





Initial Port (main.c)

void Init_Port(void)

{



//P00IE=0;P000E=0;P000D=0;P00PU=0; P01IE=0;P010E=0;P010D=0;P01PU=0; P02IE=0;P020E=0;P020D=0;P02PU=0; P03IE=0;P030E=0;P030D=0;P03PU=0; P04IE=0;P040E=0;P040D=0;P04PU=0; P05IE=0;P050E=0;P050D=0;P05PU=0; P06IE=0;P060E=0;P060D=0;P07PU=0;

17.2.2 Port n Data Register (PnD:n=0 to 9, A, B)

PnD is a special function register (SFR) used to read the level of the port n pin and write output data. The input level of the port n pin can be read by reading PnDI in the input mode. Data written to PnDO in the output mode are output to the port n pin. The PnDO is readable.

Enable or disable the input or output by using the port n mode register.

See Table 17-2 "List of Registers / Bits" to check avaible pins and bits.

Wirte "0" to the bits of PnDO register that have no corresponding pin.

The bits of PnDI register that has no corresponding pin always return "0" for reading.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	rd PnD															
Byte				Pn	DO							Pn	nDI			
Bit	Pn7DO	Pn6DO	Pn5DO	Pn4DO	Pn3DO	Pn2DO	Pn1DO	Pn0DO	Pn7Dl	Pn6DI	Pn5Dl	Pn4DI	Pn3Dl	Pn2Dl	Pn1DI	Pn0DI
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R
Initial	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Bit No.	Bit symbol name	Description
15 to 8	Pn7DO to	This bit is used to set the output level of port n pin.
	Pn0DO	0: Output "L" (initial value)
		1: Output "H"
7 to 0	Pn7DI to	This bit is used to set the input level of port n pin.
	Pn0DI	0: The input level is "L"
		1: The input level is "H" (Initial value)



Initial Port (main.c)

17.2.3 Port n Mode Register 01 (PnMOD01:n=0 to 9, A, B)

void Init Port(void)

ł

//										
// Port 0 Init										
//										
//P00D0=0;	// Output = 0									
P01D0 =0;	// Output = 0									
P02D0=0;	// Output = 0									
P03D0=0;	// Output = 0									
P04D0=0;	// Output = 0									
P05D0	// Output = 0									
P06D0=	// Output = 0									
P07D0	// Output = 0									

//P00IE-0;P000E=0;P000D=0;P00PU=0; - / / **P011E**=0; **P010E**=0; **P010D**=0; **P01PU**=0; P02IE=0; P02OE=0; P02OD=0; P02PU=0; **P03IE**=0; **P030E**=0; **P030D**=0; **P03PU**=0; **P04IE**=0; **P040E**=0; **P040D**=0; **P04PU**=0; P05IE=0; P05OE=0; P05OD=0; P05PU=0; **P06IE**=0; **P060E**=0; **P060D**=0; **P06PU**=0; **P07IE**=0; **P070E**=0; **P070D**=0; **P07PU**=0;

11

11

11

11

11

11

11

//P00MD3=0;P00MD2=0;P00MD1=0;P00MD0=0; P01MD3=0; P01MD2=0; P01MD1=0; P01MD0=0; P02MD3=0; P02MD2=0; P02MD1=0; P02MD0=0; P03MD3=0; P03MD2=0; P03MD1=0; P03MD0=0; P04MD3=0; P04MD2=0; P04MD1=0; P04MD0=0; P05MD3=0; P05MD2=0; P05MD1=0; P05MD0=0; P06MD3=0; P06MD2=0; P06MD1=0; P06MD0=0; **P07MD3**=0; **P07MD2**=0; **P07MD1**=0; **P07MD0**=0; //-----

PnMOD01 is a special function register (SFR) to choose the input/output mode, input/output status, and shared function of Pn0 pin and Pn1 pin.

See Table 17-2 "List of Registers / Bits" to check avaible pins and bits.

Wirte "0" to the bits of PnMOD01 register that have no corresponding pin.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	PnMOD01															
Byte	PnMOD1 PnMOD0															
Bit	Pn1MD 3	Pn1MD 2	Pn1MD 1	Pn1MD 0	Pn10D	Pn1PU	Pn10E	Pn1IE	Pn0MD 3	Pn0MD 2	Pn0MD 1	Pn0MD 0	Pn0OD	Pn0PU	Pn0OE	Pn0IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	*	0	*
* .T1	a initia	1	of DOC	TT and	DOODI	I for th	a Danto	in 1111 .	and ath	on hita	ana "0"					

 \therefore I ne initial value of POOLE and POOPU for the Porto is "1" and other bits are "0".

9	Pn1OE	This bit is used to enable the output of Pn1 pin
		0: Disable the output (initial value)
		1: Enable the output
8	Pn1IE	This bit is used to enable the input of Pn1 pin
		0: Disable the input (initial value)
		1: Enable the input



Initial Port (main.c)

void Init_Port(void)

//-----

{

17.2.3 Port n Mode Register 01 (PnMOD01:n=0 to 9, A, B)

PnMOD01 is a special function register (SFR) to choose the input/output mode, input/output status, and shared function of Pn0 pin and Pn1 pin.

See Table 17-2 "List of Registers / Bits" to check avaible pins and bits.

Wirte "0" to the bits of PnMOD01 register that have no corresponding pin.

//	Port 0 Init		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
//	// 0 / 0	Word								PnM	OD01							
//P00D0=0;	// Output = 0	Byte				PnM	OD1							PnM	OD0			
P02D0=0;	// Output = 0	Dit	Pn1MD	Pn1MD	Pn1MD	Pn1MD	Pn10D	Pn1PI I	Pn10E	Pn1IF	Pn0MD	Pn0MD	Pn0MD	Pn0MD	Pn0OD	Pn0PU	Pn0OE	Pn0IE
P03D0=0;	// Output = 0	Dit	3	2	1	0			11102		3	2	1	0	1 11000	1 1101 0	INCOL	1 11012
P04D0=0;	// Output = 0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
P05D0=0;	// Output	Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	*	0	*
P06D0 =0;	// Output	value												Ū	Ū		•	
P07D0 =0;	// Output	* :11	ne initia	il value	e of POC	ole and	POOPU	for the	e Porto	1S "1" a	and oth	er bits	are "0".					
		-	Pn10	D	Th	is bit is	used c	hoose t	he outp	out type	of Pn1	pin.						
//P001E=0;F	000E=0;0000.0;	P00P0=0; /,			An	LED is	direct	y drive-	able by	enlarg	ing the	current	t when t	the N-c	hannel	open d	rain out	tput
POILE=0;POI	OE = 0; POIDD = 0; PO.	1PU=0; /,			mo	ode is cł	nosen.											
P021E=0,P02	OE 0, P020D=0, P0	2PU=0; /,			Se	e the da	ata she	et for d	etails a	bout th	e curre	nt drive	ability.					
POSIE=0, POS	DE=0, P050D=0, P0	ADU_0; /,				0: CM	1OS ou	tput (in	itial valu	ue)								
P041E=0;P04	OE_0; P040D=0; P04	4PU=0; /,				1: N-c	hanne	lopen	drain ou	utput								
POSTE=0, POS	0E-0.P050D-0.P0	SPU=0, //	Pn1P	U	Т	his bit i	s usec	l to ena	able the	e intern	al pull-	up resi	istor of	Pn1 pi	n.			
P07TE-0.P07	0E-0.P000D-0.P0	7PII-0· //			Т	he inte	rnal pu	ıll-up re	esistor	can be	enable	ed on fo	ollowin	g cond	tions o	f the p	ort.	
10/12-0,10/		// //				The in	put is	enable	d and t	he out	put is d	isableo	d on the	e port				
//P00MD3-0.		-0.00000-0.				The in	put is	enable	d and t	he N-c	hannel	open	drain o	utput is	chose	n on th	e port	
P01MD3=0.P0	1MD2=0.P01MD1=0	• P01MD0=0 •				0: V	Vithout	a pull-	up resi	stor (ir	nitial va	lue)						
P02MD3=0;P0	2MD2=0; P02MD1=0	: P02MD0=0:				1: V	Vith a p	oull-up	resisto	r								
P03MD3=0:P0	3MD2=0: P03MD1=0	: P03MD0=0:																
P04MD3=0; P0	4MD2=0; P04MD1=0	; P04MD0=0;			Т	he con	ditions	of the	port ar	e spec	ified by	Pn1IE	E, Pn1C	DE and	Pn10l	D bit.		
P05MD3=0;P0	5MD2=0; P05MD1=0	; P05MD0=0;				10X:	:	Setting	of Pn1	PU bit	is ena	ole						
P06MD3=0;P0	06MD2=0; P06MD1=0	; P06MD0=0;				111:	:	Setting	of Pn1	PU bit	is ena	ole						
P07MD3=0;P0	7MD2=0; P07MD1=0	; P07MD0=0;				Other	s:	Setting	of Pn1	PU bit	is disa	ble						
//						X: 0 o	r 1 (do	n't care	e)									

Pichet Sriphomma pichet@es.co.th



Initial Port (main.c)

void Init_Port(void)

//-----

{

17.2.3 Port n Mode Register 01 (PnMOD01:n=0 to 9, A, B)

PnMOD01 is a special function register (SFR) to choose the input/output mode, input/output status, and shared function of Pn0 pin and Pn1 pin.

See Table 17-2 "List of Registers / Bits" to check avaible pins and bits.

Wirte "0" to the bits of PnMOD01 register that have no corresponding pin.

//	Port 0 Init		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
//		Word								PnM	OD01							
//P00D0=0;	// Output = 0	Byte				PnM	OD1							PnM	IOD0			
P01D0=0;	// Output = 0		Pn1MD	Pn1MD	Pn1MD	Pn1MD			D 405	D 415	Pn0MD	Pn0MD	Pn0MD	Pn0MD			D 005	D 015
P02D0=0,	// Output = 0	Bit	3	2	1	0	Ph1OD	PhiPU	PhiOE	Phile	3	2	1	0	Phuod	Phopo	Phuoe	Phote
P04D0=0:	// Output = 0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
P05D0=0;	// Output = 0	Initial	-			•	•											
P06D0=0;	// Output = 0	value	0	U	0	0	0	U	0	0	0	0	0	U	0	Ŷ	0	^
P07D0=0;	// Output = 0	* :Tl	ne initia	ıl value	of P00	IE and	P00PU	for the	e Port0	is "1" :	and oth	er bits a	are "0".					
			Pn1M	D3 to	Th	nis bit is	used t	to choc	se the	shared	d functio	on of P	n1 pin.					
//P00IE=0;F	P000E=0;P000D=0;P0	0PU=0; //	Pn1M	D0	Fo	or the de	etails o	f the sl	hared fu	unctior	n, see T	able 1-	7 "ML6	32Q13()0 Grou	ıp Pin l	_ist", Ta	ble 1-8
P01IE=0; P01	LOE=0;P010D=0;P01P	U =0; //	1		"N	1L62Q1	500 Gr	roup Pi	n List" :	and Ta	able 1-9	"ML62	2Q1700) Group	Pin Li	st".		
P02IE=0;P02	20E=0;P020D=0;P02P	U =0; //	1			0000:	Prim	nary fui	nction (initial v	/alue)							
P03IE=0;P03	30E=0; P030D=0; P03P	U =0; //	1			0001:	2 nd f	unctior	า		-							
P04IE=0;P04	40E=0;P040D=0;P04P	U =0; //	1			0010:	3 rd f	unctior	ı									
P05IE=0;P05	50E=0;P050D=0;P05P	U =0; //	1			0011:	4 th f	unction	1									
P06IE=0;P00	50E=0;P060D=0;P06P	U =0; //	1			0100:	5 th f	unction	ì									
P07IE=0;P07	70E=0;P070D=0;P07P	U =0; //	1			0101:	6 th f	unction	1									
						0110:	7 th f	unction	ì									
//P00MD3=0;	;P00MD2=0;P00MD1=0	;P00MD0=0;				0111:	8 th f	unction	ì									
P01MD3=0;P0	01MD2=0;P01MD1=0;P	01MD0 =0;				1XXX:	Do r	not use) (Prima	arv fun	ction)							
P02MD3=0;P0	02MD2=0;P02MD1=0;P	<mark>02MD0</mark> =0;				X: 0 or	1 (don	't care)		,	,							
P03MD3=0;P0	03MD2=0;P03MD1=0;P	03MD0=0;					`	,										
P04MD3=0;P0	04MD2=0;P04MD1=0;P	04MD0=0;																
P05MD3=0;P0	05MD2=0; P05MD1=0; P	05MD0=0;																
P06MD3=0;P0	06MD2=0;P06MD1=0;P	06MD0=0;																
P07MD3=0;P0	07MD2=0; P07MD1=0; P	07MD0=0;																

//-----



2. UART



Import Project



Right-click on project Explorer and select Import.

Chapter 2 UART

Import Project

Mimport – 🗆 🗙	
Select Create new projects from an archive file or directory.	
Select an import wizard:	
type filter text	
 Control Control Contrelation Control Control Control Control Control Control Cont	
? < Back Next > Finish Cancel	

Select General and choose Existing Projects into Workspace. Click Next.



Import Project

M Import					—	
Import Projects Select a directory to sea	rch for existing f	Eclipse pro	jects.			
Select root directory:					~	Browse
○ Select archive file:					~	Browse
Projects:						
						Select All
						Deselect All
						Refresh
Options						
Search for nested pro	ojects orkspace					
Hide projects that alr	eady exist in the	e workspac	e			
Working sets						
Add project to work	ing sets					New
Working sets:					/	Select
?	< Back	Nex	t >	Finish		Cancel

LEXIDE up the new window. Click Browse.. at Select root directory. Choose "Chapter 2 UART" in Folder window.

8	Browse For Folder	×							
	Select root directory of the projects to import								
	> 🔜 Desktop	^							
	> 🖹 Documents								
	> 🖊 Downloads								
	> 📐 Favorites								
	✓								
	🗸 📙 Code Training Lapis								
_	> 📙 .metadata								
	> 📕 Chapter 1 GPIO								
	🗸 📜 Chapter 2 UART								
d	.settings								
t¢	> 📙 Debug								
a	source files								
	> 📕 Chapter 3 ADC								
w	> 📙 Chapter 4 I2C								
	RemoteSystemsTempFiles								
	L								
	Folder: Chapter 2 UART								
	Make New Folder OK Cancel								



Import Project

M Import				—	
Import Projects Select a directory to sea	rch for existing E	clipse projects.			
 Select root directory: Select archive file: 	C:\Users\pichet	\lexide\Code Train	ning Lapis\Cha	ptei ~	Browse Browse
Projects:					
Chapter 2 UART (C	C:\Users\pichet\\	exide\Code Trainii	ng Lapis\Chapt	ter 2 U/ [Select All Deselect All Refresh
Search for nested pro	ojects orkspace ready exist in the	workspace			
Working sets	ing sets				New Select
?	< Back	Next >	Finish		Cancel

After choosing Project Click Finish.



Import Project



Then appear the project on Project Explorer.

Chapter 2 UART



Check Device

🔰 Code Training La	pis - LEXIDE-U16		
File Edit Source	Refactor Navigate Search P	Project Run LAPIS Window	/ H
🖻 🕶 🔚 🕼 🖌 🕶	🔨 🕶 🗟 💌 📮 🔌 📸 🕶	• 🚳 🕶 🖻 🕶 🎯 🕶 🔯 🕶 🖸	•
Project Explorer > 🍰 Chapter 1 GPI 🗸 🔁 Chapter 2	x		
> 🗊 Include > 🗁 Debug	Go Into		
> 🗁 source > 📓 ML621	Open in New Window Show in Local Terminal	>	
	Copy Paste Delete Remove from Context	Ctrl+C Ctrl+V Delete Ctrl+Alt+Shift+Down	
	Move Rename	F2	
29 2	Import Export		
	Clean and Build Project Build Project Clean Project		
8	Refresh Close Project Close Unrelated Projects	F5	
	Build Configurations Build Targets	>	
	Index	>	
	Show in Remote Systems view Run As	>	
	Debug As	>	
	Profile As	>	
	Team Compare With	>	
	Restore from Local History	,	nso
*	Run C/C++ Code Analysis		tim
	Configure	>	
	Properties	Alt+Enter	

Right-click on a project folder and select [Properties] .

Pichet Sriphomma pichet@es.co.th


Choose LEXIDE-U16 Settings

Select Device

N Properties for Chapter	1 GPIO	– D X
type filter text	LEXIDE-U16 Settings	↓ ↓ ↓ ▼
 > Resource Builders > C/C++ Build Build Variables Environment Logging Settings Tool Chain Editor > C/C++ General C/C++ General LEXIDE-U16 Settings Project References Run/Debug Settings > Task Repository WikiText 	Target MCU Category: ML62Q1300 ∨ Group: ML621367 ∨ Toolchain Always use the latest Toolchain Version: V2_01_00 ×	Restore Defaults Apply
?		Apply and Close Cancel

Build Project

🐰 Coo	de Tr	aining Lapis - LEXIDE-U16	
ile E	dit	Source Refactor Navigate Search Project Run LAR	PIS Windov
<u>9</u> –	8	🗟 🗞 🕶 🚳 🕶 📴 💌 🖳 🔞 🕶 🚱 🕶 🚱	- * • C
Proj	ject	Explorer 🛛 🕞 🔄 😜 🗢 🗖 🗖	
) 😂	Chap	oter 1 GPIO	
es (C	New	
>		Go Into	
2		Open in New Window	
>		Show in Local Terminal	>
	P	Copy Ctrl+C	
	Ē	Paste Ctrl+V	
	×	Delete Delete	
	<u>\$</u>	Remove from Context Ctrl+Alt+Shift+Down	
		Move	í
		Rename F2	
	പ്പ	Import	
	4	Export	
		Clean and Build Project	
		Build Project	
		Clean Project	
	<u>8</u>	Ketresh F5 Close Project	
		Close Unrelated Projects	
		Build Configurations	>
		Build Targets	>
		Index	>
		Show in Remote Systems view	-
		Run As	>
		Debug As	>
		Profile As	>
		Ieam Compare With	
		Restore from Local History	sks 📮 Co
	*	Run C/C++ Code Analysis	alay at this
		Configure	>
		Properties Alt+Enter	
	_		

Right-click on a project folder and select [Build Project] to start the build process.



Build Project



When the build succeeds , an ABS file is generated.

© 2017 ROHM Semiconductor (Thailand) Co., Ltd. © 2017 LAPIS Semiconductor Co., Ltd. All rights reserved







UART00_Get_Data









General Description

		ML62Q13	300 group			ML62Q150	0/ML62Q	1700 grou	C
Channel no.	16pin product	20pin product	24pin product	32pin product	48pin product	52pin product	64pin product	80 pin product	100pin product
0	•	•	•	•	•	•	•	•	•
1	•	•	•	•	•	•	•	•	•
2	-	-	-	—	_	-	-	•	•
3	-	-	_	—	-	-	_	•	•
4	-	-	-	_	-	-	-	•	•
5	-	-	-	_	-	-	_	•	•

•: Available -: Unavailable



Features

Serial Communication mode	Operation mode	Features
Synchronous Serial I/O Port	8-bit mode	 Max. 6ch (Both SSIO and UART are unavailable to use in the same channel) Master mode / Slave mode MSB first / LSB first
(3310)	16-bit mode	 8bit / 16bit data length Self-test function using the master and slave modes. For the self-test functions, see Chapter 29 "Safety Function."
UART mode	Half-duplex communication	 5-bit/6-bit/7-bit/8-bit data length Odd parity/even parity/0 parity/1 parity/and no parity One stop bit/Two stop bits Positive logic/negative logic for communication logic MSB first / LSB first Wide range of communication speed 1bps to 4,800bps (Clock frequency is 32.768kHz) 600bps to 3Mbps (Clock frequency is 24MHz) 300bps to 2Mbps(Clock frequency is 16MHz)
	Full-duplex communication	 Built-in baud rate generator for each channel One channel is usable as two channels of half-duplex communication UART Parity error flag, overrun error flag, framing error flag, transmission buffer status flag, reception buffer status flag Self-test function using transmission and reception For the self-test functions, see Chapter 29 "Safety Function."

Table 11-2 Features of the Serial Communication

Configuration



© 2017 ROHM Semiconductor (Thailand) Co., Ltd. © 2017 LAPIS Semiconductor Co., Ltd. All rights reserved



List of Pins

							ML620 gro	Q1300 Dup			ML ML	.62Q15 .62Q17 group	500 700	
Channel no.	Pin name	Sh	ared port	Setting register	Setting value	16pin product	20 pin product	24 pin product	► 32 pin product	48 pin product	52 pin product	64 pin product	80 pin product	100 pin product
		P03	2 nd Func.	P0MOD3	0001_XXXX ^{*1}	•	•	•	•	٠	٠	٠	•	•
_	300_17.00	P13	2 nd Func.	P1MOD3	0001_XXXX ^{*1}	•	•	•	•	•	٠	٠	•	•
		P02	2 nd Func.	P0MOD2	0001_XXXX ^{*2}	•	•	•	•	•	٠	٠	•	•
		P07	3 rd Func.	P0MOD7	0010_XXXX ^{*2}	-	-	-	•	•	•	٠	•	•
_	300_KAD0	P12	2 nd Func.	P1MOD2	0001_XXXX ^{*2}	-	-	•	•	•	•	•	•	•
_		P17	3 rd Func.	P1MOD7	0010_XXXX ^{*2}	•	•	•	•	•	•	•	•	•
0		P03	3 rd Func.	P0MOD3	0010_XXXX ^{*1}	•	•	•	•	•	•	•	•	•
		P10	2 nd Func.	P1MOD0	0001_XXXX ^{*1}	-	-	-	•	•	•	•	•	•
	500_IXDI	P13	3 rd Func.	P1MOD3	0010_XXXX ^{*1}	•	•	•	•	•	•	•	•	•
		P20	2 nd Func.	P2MOD0	0001_XXXX ^{*1}	•	•	•	•	•	•	•	•	•
		P07	2 nd Func.	P0MOD7	0001_XXXX*2	-	-	-	•	•	•	•	•	•
		P17	2 nd Func.	P1MOD7	0001_XXXX ^{*2}	•	•	•	•	•	•	•	•	•
							1							

Combination of UART port

• Full-duplex communication

Innut/outru	it nin	<u> </u>	mbination 1	Buffe	r register				
input/outpu	it pin		mpination 1	SDnBUFL	SDnE	BUFH			
SUn_RX	(D	S	Un_RXD0	Received data	Transm	nit data			
SUn_TX	(D	S	Un_TXD1	(SUn_RXD0)	(SUn_	TXD1)			
						n=0 to 5	5		
Half-duplex c	ommunica	tion							
nout/output pin	Combinat	ion 1	Combination	,	register				
nput/output pin	Combinat		Combination	SDnBL	IFL	5	SDnBUFH		
SUn_RXD	SUn_RX	(D0	SUn_RXD1		eive data	UART	n1 Receive data		
 _							SUN_RXUT)		
SUn_TXD SUn_TXD1			SUn_TXD0	(SUn_T)	Smit data (D0)	(SUn_TXD1)			
							n=0 to E		

Channel 0

n=0 to 5

Channel 1



Transfer Data Format



Figure 11-12 Format of Positive Logic Input/Output (LSB First)

Concept





Concept





Concept







Disable the input (initial value)

Enable the input



0: 1: 0

R/W



S	et Pin (uart0.c)	17.2.3	3 Por	t n Mo	ode Re	gister	r 01 (PnMC	D01:r	n=0 t	o 9, A	, B)						
			PnMOD of Pn0 p See Tab Wirte "0	001 is a oin and le 17-2)" to the	special f Pn1 pin. "List of e bits of	unction Registe PnMOI	n regist ers / Bi D01 reg	er (SFR) ts" to ch gister tha) to cho neck ava at have i	ose the lible pir no corre	input/or is and b espondin	utput m its. ng pin.	iode, inp	out/outp	out statu	s, and s	hared f	inction
voi	d Set_UART_00_Pin(void)		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
{		Word								PnM	DD01							
	// UART00 RXD Pin	Byte				PnM	OD1							PnM	OD0	,		
	P02IE=1;P02OE=0;P02OD=0;P02PU=0; P02MD3=0;P02MD2=0;P02MD1=0;P02MD0=1;	Bit	Pn1MD 3	Pn1MD 2	Pn1MD 1	Pn1MD 0	Pn1OD	Pn1PU	Pn10E	Pn1IE	Pn0MD 3	Pn0MD 2	Pn0MD 1	Pn0MD 0	Pn0OD	Pn0PU	Pn0OE	Pn0IE
	PARTE-0. PARCE-1. PARCO-0. PARDIL-0.	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	P03MD3=0:P03MD2=0, 3MD1=0, 3MD0=1:	Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	*	0	*
}	· · · · · · · · · · · · · · · · · · ·	value * :T	he initia	al value	e of P00	IE and	P00PU	U for the	e Port0	is "1" a	and oth	er bits a	are "0".					
		Pn1	OD	-	This bit	is used	choo	se the d	output t	vpe of	Pn1 pi	n.						
				/	An LED	is dire	ctly dr	ive-able	e by enl	arging	the cu	rrent w	hen the	N-cha	annel o	pen dr	ain out	put
				r	mode is	chose	n.											
					See the	data s	heet fo	or detai	ls abou	t the c	urrent o	trive al	bility.					
					1: N	Join Contraction	nel op	en draii	value) n outpu	t								
		Pn1	PU		This b	it is us	ed to	enable	the int	ernalı	oull-up	resist	or of P	n1 pin				
					The in	ternal	pull-u	p resist	tor can	be en	abled	on follo	owing	conditi	ons of	the po	ort.	
					The	input i	s ena	bled ar	nd the	output	is disa	bled o	on the p	oort				
					The	input i	s ena	bled ar	nd the I	N-char	nnel op	en dra	ain outp	out is c	hosen	i on th	e port	
					0:	Withc	out a p	ull-up	resisto	r (initia	l value	e)						
					1:	With	a pull-	up resi	istor									
					The co	onditio	ns of t	the por	t are s	pecifie	d by P	n1IE, I	Pn1OE	and P	n10D	bit.		
					10X	:	Sett	ing of F	Pn1PU	bit is e	enable							
					111	:	Sett	ing of F	Pn1PU	bit is e	enable							
					Oth	ers:	Sett	ing of F	Pn1PU	bit is o	disable	•						
					X: 0	or 1 ((don't d	care)										_



Set Pin (uart0.c)	17.2.3	8 Port	n Mc	ode Re	egiste	r 01 (l	PnMC	D01:	n=0 t	o 9, A	, B)						
	F C S V	PnMOD01 is a special function register (SFR) to choose the input/output mode, input/output status, and shared function of Pn0 pin and Pn1 pin. See Table 17-2 "List of Registers / Bits" to check avaible pins and bits. Wirte "0" to the bits of PnMOD01 register that have no corresponding pin.												unction			
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<pre>oid Set_UART_00_Pin(void)</pre>	Word								PnM	OD01							
	Byte				PnM	IOD1							Pn№	10D0			
// UARI00 RXD Pin P02IE=1;P02OE=0;P02OD=0;P02PU=0;	Bit	Pn1MD 3	Pn1MD 2	Pn1MD 1	Pn1MD 0	Pn10D	Pn1PU	Pn10E	Pn1IE	Pn0MD 3	Pn0MD 2	Pn0MD 1	Pn0ME 0	Pn0OD	Pn0PU	Pn0OE	Pn0IE
P02MD3=0;P02MD2=0;P02MD1=0;P02MD0=1;	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<pre>P03IE=0;P03OE=1;P03OD=0;P03PU=0; P03MD3=0;P03MD2=0;P03MD1=0;P03MD0=1;</pre>	Initial value * :T1	0 he initia	0 I value	0 e of P00	0 DE and	0 P00Pt	0 J for th	0 e Port0	0 is "1" a	0 and oth	0 er bits	0 are "0".	0	0	*	0	*
	Pn1 Pn1	MD3 to MD0		This bi For the "ML62 000 001 001 010 010 011 011 1XX X: 0	it is use e detail Q1500 0: F 1: 2 0: 3 1: 4 0: 5 1: 6 0: 7 1: 8 (X: E or 1 (c	ed to c ls of th Group Primary nd func th func th func th func th func th func t th func t th func t th func t th func	hoose e share o Pin L function tion tion tion tion tion use (P are)	the sha ed func ist" and on (init	ared fu tion, s d Table ial valu	nction ee Tab ≥ 1-9 "N ie) n)	of Pn1 ble 1-7 ML62Q	pin. "ML62 1700 C	Q1300 Эroup) Group Pin List	Pin L ".	ist", Ta	ble 1-8



Start ,Stop UART (uart0.c)

<pre>void Stop_UART00(void) </pre>	Д Д
1 U00EN = 0; // 5	II
}	Wo
//	Ву
<pre>void Start_UART00(void)</pre>	В
{	R/
unsigned char i;	lnit val
i = SD0BUFL; // F	
UA00STAT = 0xff;	
U00EN = 1; // 5	
}	

11.2.5 Serial Communication Unit n Control Register (SUnCON)

SUnCON is a specific function register (SFR) to control the serial communication unit.

Addro Acce Acce Initial	ess: ss: ss size value:	:	0xF606(0xF627(0xF666(0xF687(R/W 8/16bit 0x0000	SU0CC SU1CC SU3CC SU4CC	ONL/SU ONH), (ONL/SU ONH), (0CON 1xF646 13CON 1xF6A6), 0xF6((SU2CC), 0xF66 (SU5CC	07(SU0 DNL/SU 07(SU3 DNL/SU	CONH) I2CON) CONH) J5CON), 0xF62), 0xF64), 0xF68), 0xF68	26(SU1 47(SU2 36(SU4 A7(SU5	CONL/ CONH CONL/ SCONF	/SU1CC), /SU4CC I)	DN), DN),		
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Nord								SUn	CON							
Byte				SUnC	ONH							SUn	CONL			
Bit	-	-	-	-	_	-	Un1E N	-	Ι	-	-	-	-	-	Un0E N	SnEN
R/W	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R/W	R/W
nitial /alue	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Un	0EN		This • UA 0: 1: • UA 0: 1:	bit is u NRT Fu Stop Star NRT Ha Stop Star	sed to III-dupl the U the U alf-dup the U the U	enable ex moo IARTn IARTn Iex moo IARTn0	e the U de comm comm de) comn) comn	ARTn unicatio unicatio nunication	comm on (Init on tion (In tion	unicati ial valu itial va	on in t ıe) lue)	he UAF	RT mo	de.	

SSIO mode

The Un0EN bit is unwritable.

- 0: Unused
- 1: Unused



void Config_UART_00(void)

unsigned char i;

Set_UART_00_Pin();
Stop_UART00();

//UA00MOD -> UART00 Mode U00I0 = 0; U00CK1=1;U00CK0=0; // Delec U00RSS = 0; // Value U00LG1=0;U00LG0=0; // 8-bit U00PT2=0;U00PT1=0;U00PT0=0; U00STP = 0; // 1 stop bi U00NEG = 0; // Positive U00DIR = 0; // LSB first

```
// 9600 bps Baud Rate Settir
UA00BRT = 0x09c1; // E
UA00BRC = 0x04; // Baud
// 115,200 bps Baud Rate Set
// UA00BRT = 0x00cf; // E
// UA00BRC = 0x02; // Baud
```

```
Set_UART00_RX();
//Set_UART00_TX();
```

}

UAn0MOD is a specific function register (SFR) to set the mode in UARTn0 full-duplex communication mode and half-duplex communication mode.

Address:	0xF60C(UA00MODL/UA00CON), 0xF60D(UA00MODH), 0xF62C(UA10MODL/UA10MOD), 0xF62D(UA10MODH), 0xF64C(UA20MODL/UA20CON), 0xF64D(UA20MODH), 0xF66C(UA30MODL/UA30MOD), 0xF66D(UA30MODH), 0xF68C(UA40MODL/UA40CON), 0xF68D(UA40MODH),
Access: Access size: Initial value:	0xF6AC(UA50MODL/UA50MOD), 0xF6AD(UA50MODH) R/W 8/16bit 0x0000

	15	14	13	12	11	10	9	8	1	6	5	4	3	2	1	0
Word								UAnC	MOD							
Byte				UAn0I	MODH							UAn0I	NODL			
Bit	Un0D IR	Un0N EG	Un0S TP	Un0P T2	Un0P T1	Un0P T0	Un0L G1	Un0L G0	Un0R SS	Ι	Ι	_	_	Un0C K1	Un0C K0	Un0I O
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Un0CK1 to	This bit is used to choose base clock of baud rate generator in UARTn0 full-duplex and
Un0CK0	half-duplex mode.
	00: LSCLK (initial value)
	01: Do not use (LSCLK)
	10: HSCLK
	11: Do not use (HSCLK)
Un0IO	This bit is used to choose the transmission mode or reception mode in UARTn0 full-duplex and half-duplex mode. When the full-duplex communication mode is chosen, this bit is fixed to "1" and the UART performs as the reception mode.
	0: Transmission mode (Initial value)
	1: Reception mode



Un0RSS

void Config_UART_00(void)

unsigned char i;

Set_UART_00_Pin(); Stop_UART00();

```
UA00BRT = 0x09c1; // E
UA00BRC = 0x04; // Baud
// 115,200 bps Baud Rate Set
// UA00BRT = 0x00cf; // E
// UA00BRC = 0x02; // Baud
```

```
Set_UART00_RX();
//Set_UART00_TX();
```

}

UAn0MOD is a specific function register (SFR) to set the mode in UARTn0 full-duplex communication mode and half-duplex communication mode.

Address:	0xF60C(UA00MODL/UA00CON), 0xF60D(UA00MODH), 0xF62C(UA10MODL/UA10MOD), 0xF62D(UA10MODH), 0xF64C(UA20MODL/UA20CON), 0xF64D(UA20MODH)
	0xF66C(UA30MODL/UA30MOD), 0xF66D(UA30MODH), 0xF66C(UA30MODL/UA30MOD), 0xF66D(UA30MODH), 0xF68C(UA40MODL/UA40CON), 0xF68D(UA40MODH),
Access: Access size: Initial value:	0xF6AC(UA50MODL/UA50MOD), 0xF6AD(UA50MODH) R/W 8/16bit 0x0000

	15	14	13	12	11	10	9	8	1	6	5	4	3	2	1	0
Word								UAnC	MOD							
Byte				UAn0I	MODH							UAn0I	MODL			
Bit	Un0D IR	Un0N EG	Un0S TP	Un0P T2	Un0P T1	Un0P T0	Un0L G1	Un0L G0	Un0R SS	Ι	Ι	-	-	Un0C K1	Un0C K0	Un0l O
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W
nitial /alue	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This bit is used to choose sampling timing of the reception data in UARTn0 full-duplex and half-duplex mode.

0: (Values set to UAn0BRTH and UAn0BRTL registers)/2 (initial value)

1: {(Values set to UAn0BRTH and UAn0BRTL registers)/2} -1

Un0LG1 toThis bit is used to choose the communication data length in UARTn0 full-duplex andUn0LG0half-duplex mode.

00: 8-bit length (Initial value)

01: 7-bit length

10: 6-bit length

11: 5-bit length



void Config_UART_00(void)

unsigned char i;

{

}

Set_UART_00_Pin(); Stop_UART00();

//UA00MOD -> UART00	Мос	le
U00IO = 0;	//	Recei
U00CK1=1;U00CK0=0;	//	Selec
U00RSS = 0;	11	Value
U00LG1=0;U00LG0=0;	//	8-bit
U00PT2=0;U00PT1=0;U0	90P1	<mark>0=0;</mark>
U00STP = 0;	L st	top bi
U00NEG = 0;	osi	itive
U00DIR = 0;	SB	first

```
// 9600 bps Baud Rate Settir
UA00BRT = 0x09c1; // E
UA00BRC = 0x04; // Baud
// 115,200 bps Baud Rate Set
// UA00BRT = 0x00cf; // E
// UA00BRC = 0x02; // Baud
```

```
Set_UART00_RX();
//Set_UART00_TX();
```

UAn0MOD is a specific function register (SFR) to set the mode in UARTn0 full-duplex communication mode and half-duplex communication mode.

Addr Acce Acce Initia	Address:0xF60C(UA00MODL/UA00CON), 0xF60D(UA00MODH), 0xF62C(UA10MODL/UA10MOD), 0xF62D(UA10MODH), 0xF64C(UA20MODL/UA20CON), 0xF64D(UA20MODH), 0xF66C(UA30MODL/UA30MOD), 0xF66D(UA30MODH), 0xF68C(UA40MODL/UA40CON), 0xF68D(UA40MODH), 0xF6AC(UA50MODL/UA50MOD), 0xF6AD(UA50MODH)Access:R/WAccess size:8/16bit 0x000															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word								UAn0	MOD							
Byte		UAn0MODH UAn0MODL														
Bit	Un0D IR	Un0N EG	Un0S TP	Un0P T2	Un0P T1	Un0P T0	Un0L G1	Un0L G0	Un0R SS	-	-	-	-	Un0C K1	Un0C K0	Un0l O
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Un0S	STP		This bi	t is us	ed to c	hoose	the st	op bit l	ength	in UAF	RTn0 fเ	ull-dup	lex an	d half-	duplex	mode.
			0:	1 stop	bit (In	iitial va	alue)									
			1:	2 stop	bit											
Un0F	PT2 to		This bi	t is us	ed to c	hoose	the pa	arity bit	in UA	RTn0	full-dup	olex an	d half	-duple:	x mode	Э.
Un0F	РТ0		000	: No	o parity	/ bit (in	nitial va	alue)								
			001	: 00	dd pari	ty										
			010	: No	o parity	/ bit										
			011	: E\	/en pa	rity										

- 100: No parity bit
- 101: Parity bit is fixed to "1"
- 110: No parity bit
- 111: Parity bit is fixed to "0"



void Config_UART_00(void)

unsigned char i;

Set_UART_00_Pin();
Stop_UART00();

//UA00MOD -> UART00	Mod	e
U00IO = 0;	//	Recei
U00CK1=1;U00CK0=0;	11	Selec
U00RSS = 0;	11	Value
U00LG1 =0; U00LG0 =0;	11	8-bit
U00PT2=0;U00PT1=0;U0	90PT	0=0;
U00STP = 0; // 1	l st	op bi
U00NEG = 0;	osi	tive
U00DIR = 0; // l	SB	first

```
// 9600 bps Baud Rate Settir
UA00BRT = 0x09c1; // E
UA00BRC = 0x04; // Baud
// 115,200 bps Baud Rate Set
// UA00BRT = 0x00cf; // E
// UA00BRC = 0x02; // Baud
```

UAn0MOD is a specific function register (SFR) to set the mode in UARTn0 full-duplex communication mode and half-duplex communication mode.

Address:		0xF60C(0xF62C(0xF64C(0xF66C(0xF68C(0xF68C(UA00N UA10N UA20N UA30N UA30N UA30N	NODL/U NODL/U NODL/U NODL/U NODL/U MODL/U	JA00CC JA10MC JA20CC JA30MC JA40CC JA50M	ON), 0) OD), 0; ON), 0) OD), 0; ON), 0) OD), 0	<pre><f60d(\ th="" xf62d(\="" xf64d(\="" xf66d(\="" xf68d(\="" xf6ad(\="" xf6ad(\)="" xf7ad(\)="" xf7ad(\)<=""><th>UA00M UA10M UA20M UA30M UA40M (UA50M</th><th>ODH), ODH), ODH), ODH), ODH), 10DH)</th><th></th><th></th><th></th><th></th></f60d(\></pre>	UA00M UA10M UA20M UA30M UA40M (UA50M	ODH), ODH), ODH), ODH), ODH), 10DH)				
Access: Access size: Initial value:		R/W 8/16bit 0x0000											
15	14	13	12	11	10	q	8	7	6	5	4	3	

Word								UAn0	MOD							
Byte				UAn0I	MODH			UAn0MODL								
Bit	Un0D IR	Un0N EG	Un0S TP	Un0P T2	Un0P T1	Un0P T0	Un0L G1	Un0L G0	Un0R SS	Ι	-	-	_	Un0C K1	Un0C K0	Un0l O
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Un0DIR This bit is used to choose the communication direction in UARTn0 full-duplex and half-duplex mode.

0: LSB first (Initial value)

1: MSB first

1: Negative logic

Un0NEG	This b	it is used to choose logic of the data output in UARTn0 full-duplex and half-duplex
	mode.	
	0:	Positive logic (Initial value)

Set UART00 RX();

```
//Set UART00 TX();
```

}

0

2



Config UART (uart0.c)

void Config_UART_00(void)

unsigned char i;

```
Set_UART_00_Pin();
Stop_UART00();
```

UA00BRC = 0x02; // Baud

11

}

```
UAn0BRT and UAn0BRC can be calculated by the following formulae.
```

UARTn0 Baud Rate Register (UAn0BRT)

 $UAn0BRT = \frac{Base clock frequency (Hz)}{Baud rate (bps) - 1}$

UARTn0 Baud Rate Adjustment Register (UAn0BRC)

UAn0BRC = $\frac{\text{Base clock frequency (Hz) \% Baud rate (bps)} \times 8}{\text{Baud rate (bps)}}$

In Addition you can use table lists the count values for typical baud rates.

Set_UART00_RX();
//Set UART00 TX();



lists the count values for typical baud rates

Pasa alaak	Paud rata	UAn0BRT	UAn0BRC	Actual
Base clock	Daug rate	UAn1BRT	UAn1BRC	baud rate
	1,200bps	0x4E13	0x04	1199.99bps
	2,400bps	0x2709	0x02	2399.99bps
	4,800bps	0x1384	0x01	4799.99bps
Approx. 24 MHz	9,600bps	0x09C1	0x04	9600.23bps
(approx. 23.986176 MHz)	19,200bps	0x04E0	0x02	19200.46bps
	38,400bps	0x026F	0x05	38400.92bps
	57,600bps	0x019F	0x03	57607.14bps
	115,200bps	0x00CF	0x02	115179.71bps

Table 11-7 Count Values for Typical Baud Rates (1/2)



Set UART00 TX ,RX (uart0.c)

11.2.8 UARTn0 Mode Register (UAn0MOD) void Set_UART00_TX(void) UAn0MOD is a specific function register (SFR) to set the mode in UARTn0 full-duplex communiction mode and ł half-duplex communication mode. Stop UART00(); Address: 0xF60C(UA00MODL/UA00CON), 0xF60D(UA00MODH), 00010 = 0;0xF62C(UA10MODL/UA10MOD), 0xF62D(UA10MODH), 0xF64C(UA20MODL/UA20CON), 0xF64D(UA20MODH), **OSIU00** = 0; // C10xF66C(UA30MODL/UA30MOD), 0xF66D(UA30MODH), $\mathbf{ESIU00} = 0;$ Di 0xF68C(UA40MODL/UA40CON), 0xF68D(UA40MODH), 0xF6AC(UA50MODL/UA50MOD), 0xF6AD(UA50MODH) //SUOCON -> Serial Co Access: R/W Access size: 8/16bit Start UART00(); Initial value: 0x0000 } 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 UAn0MOD Word void Set_UART00_RX(void) UAn0MODH UAn0MODL Byte Un0D Un0N Un0S Un0P Un0P Un0P Un0L Un0L Un0R Un0C Un0C Un0I Bit _ IR EG TP T2 T1 T0 G1 G0 SS K0 K1 0 Stop UART00(): R/W R/W R/W R/W R/W R/W R/W R/W R/W R R R R R/W R/W R/W R/W U00I0 = 1;Initial 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 value OSIU00 = 0;C1 ESIU00 = 1;// En Un0IO This bit is used to choose the transmission mode or reception mode in UARTn0 full-duplex and half-duplex mode. When the full-duplex communication mode is chosen, this bit is fixed to "1" and the UART performs as the reception mode. //SUOCON -> Serial Co Transmission mode (Initial value) 0: Start UART00(); 1: Reception mode



Set UART00 TX ,RX (uart0.c)





Set UART00 TX ,RX (uart0.c)

void Set_UART00_TX(void) ł Stop UART00(); **U00I0** = 0; // Th QSIU00 = 0; // Cl ESIU00 = 0; Di //SU0CON -> Serial Co Start UART00(); } void Set_UART00_RX(void) Stop UART00(); **U00I0** = 1; // Th QSIU00 = 0; // C1 **ESIU00** = 1; En //SUOCON -> Serial Co Start UART00();

5.2.3 Interrupt Enable Register 23 (IE23)

IE23 is a specific function register (SFR) to enable or disable the interrupt for each interrupt request. The bits are unwriteable when the products do not have the peripheral circuits and they return "0" for reading. After the interrupt is accepted, the master interrupt enable flag (MIE) of the CPU is reset to "0", however, the applicable each flag of IE01 is not reset and remains "1".

Addre Acces Acces Initial	ess: ss: ss size: value:	0> R/ 8/ 0>	(F022 (/W 16bit (0000	IE2/IE2	23), 0xF	023(IE	3)									
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Vord								IE:	23							
Byte				IE	3							IE	2			
Bit	ETM1	ETM0	EFTM 1	EFTM 0	EI2C M1	EI2C M0	-	EEXT X	-	ESAD	-	ESIU 01	ESIU 00	EMC S	EDM A	ECBU
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R	R/W	R	R/W	R/W	R/W	R/W	R/W
nitial ⁄alue	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ESIL	100		This b 0:	it conti Disab	rols to le the	enable interru	e or dis pt (init	able th ial valu	ie Ser ie)	ial Com	munic	cation ι	unit 00	interru	ıpt (SII	JOOINT

1: Enable the interrupt



Get Command (uart0.c)

```
void UART00_GetData(void)
{
   unsigned char i;
                                                 Read Buffer
    i = SD0BUFL; //Serial communicatior
   if(i=='<')
                                               Check symbol '<'
    {
       if(!Flag. RXD00)
           RXD00 Index = 0;
                                     Set Index Array of UART00 RX Buf = 0
    }
   else if(i=='>')
                                               Check symbol '>'
    Ł
       if(RXD00 Index)
                                             Set Flag RXD00 = 1
           Flag. RXD00 = 1;
           Stop UART00();
        }
    }
    else if(RXD00_Index < RXD00_BUF_NUM)</pre>
                                           Send Data from Buffer to Array
       UART00 RX Buf[RXD00 Index++] = i;
                                           UARTOO RX Buf
}
```



UART00 Send Data (uart0.c)

```
void UART00_SendData(void)
{
                                                                               Ν
                                                                   TXD0
    unsigned int i=0;
                                                                                           index = 0
                                                                  0 == 1
     if(Flag._TXD00)
     {
         i = UART00 TX Buf[TXD00 Index++];
                                                            i = TX_Buffer[index]
         if((TXD00_Index < TXD00_BUF_NUM)&&(i!=0))</pre>
             UART00_Send(i);
                                                                  ſndex ≷
                                                                              Ν
         else
                                                                  50 and
         {
                                                                   i!=0
             Flag. TXD00 = 0;
             Set UART00 RX();
                                                                   Y
                                                                                         TXD00 = 0 and
                                                                  Send i
     }
                                                                                         Set UART RX
    else
         TXD00_Index = 0;
}
```

UART00 Send (uart0.c)



Note : Un0FUL This bit is used to indicate the state of the transmit/receive buffer.





Before launching the program S-TERM. Must connect device. Open Bluetooth & other devices





Click Add Bluetooth or other device





Click Bluetooth and select device



S-TERM



Wait for connecting..






Show device connected



Pichet Sriphomma pichet@es.co.th





Open Program S-TERM

File Mode Option	
Comport : 🚾 Baudrate : 1200 🗸 Data bit : 8 🗸 Stop bit : 1 🗸 Parity : None 🗸 Open port Close port	
Reset Tx: 0 Reset View: Ascii > New line at : CR _ Send type : Ascii > Send type : Ascii _ Show macro	
Console	
For clear console Press key 'F11' in console or press button 'Clear console' >> Cl	ar
Data send : Send Data	
S-TERM v3.0 Disconnected DCD(1) RX(2) TX(3) DTR(4) DSR(6) RTS(7) CTS(8) RI(9) www.silaresearch.com	

Pichet Sriphomma pichet@es.co.th





💒 S-TERM v3.0
File Mode Option
Comport : COM7 V Baudrate : 1200 V
Rx: 0 Reset Tx: 0 Reset View: Asci
Console

Select Comport

💒 S-TERM v3.0					
File Mode Option					
Comport : COM7 🗸 🔁	Baudrate :	1200 1200	~	Data bit :	8 ~
Rx: 0 Reset Tx:	0 Re	4800 9600 19200 38400	As	cii	✓ New lin
Console		57600 115200 128000 230400 460800			

Select Baudrate. This workshop select 9600.





Baudrate : 1200	∠ Data bit : 8	✓ Stop bit : 1
0 Reset View :	Ascii 🗸	New line at : CR 🗸
	Ascii Hex	
	Ascii + Hex	

💒 S-TERM v3.0	
File Mode Option	
🔄 Open port F2	
Close port F3	Baudrate : 1200 V Data
Send file F4	
Save to file Ctrl+S	0 Reset View : Ascii
Exit Ctrl+End	
Console	

Select view type. This workshop select Ascii.

Click File and select Open port.





S-TERM v3.0	- 0 X
File Mode Option	
Comport : COM7 Image: Baudrate : 9600 Data bit : 8 Stop bit : 1 Parity : None Open port	Close port
Rx: 0 Reset View: Ascii v New line at : CR CR Show none charaters Send type: Ascii v New line at : CR CR Send type : Ascii Send type :	CR LF ETX Show macro
Console	
	For clear console Press key 'F11' in console or press button 'Clear console' >> Clear
Data send : Send Data	
S-TERM v3.0 Port open. DCD(1) RX(2) TX(3) DTR(4) DSR(6) RTS(7) CTS(8) RI(9) www.silaresearch.com	



Data send : <start></start>	
S-TERM v3.0 Port open. DCD(1) RX(2) TX(3) DTR(4) DSR(6) RTS(7) CTS(8) RI(9) www.silaresearch.com Image: Comparison of the state	

Print message "<Start>". After that click Send Data.





S-TERM v3.0	n							
Comport : COM7		Baudrate : 9600	✓ Data	a bit : 8 🗸	Stop bit : 1	✓ Parity :	None 🗸	
Rx : 39 Rese	et Tx :	14 Reset	View : Ascii	→ New line	e at : CR 🗸	Show none Scharaters	end type : Ascii	``
	19							
Connected->U9								
Data send : <	rt>					Send D	ata	

After Send message "<Start>" to MCU Board. After that MCU Board return Message "Connected->U00"



3. ADC



Import Project

🚺 Code T	Frain	ing La	pis - LEXIC	DE-U16								_
File Edit	Sc	ource	Refactor	Navigate	Search	Project	Run	LAPIS	Win	dow	Help	
📑 🗝 🖫	G	• 🛞	· 🔨 🗕 🗟	۹ 🗕	🔌 💣	- 🚳 -	c –	6 -	☆ ◄	• •	- 9	• 10
Project	Exp	lorer	8 E	4	~							
> 🎏 Cha	pter	1 GPI	0									
> 😤 Cha	pter	· 2 UAI	RT									
		New						>				
		Go Int	o									
		Open	in New Wi	ndow								
		Show	in Local Te	rminal				>				
Ē		Сору					C	Ctrl+C				
[Paste					(Ctrl+V				
1	₩ 8	Delete	e ve from Co	ntevt	C	`trl+∆lt+	L Shift+l	Down				
		Source	e	intext		an er and e	Jinter	>				
		Move.										
		Renan	ne					F2				
i	2	Impor	t									
12	4	Export	t									
		Clean	and Build	Project								
		Build I	Project									
	5	Clean Dofror	Project									
•	¢	rienes Close	Project					FD				
		Close	Inrelated	Projects								

Right-click on project Explorer and select Import.

Import Project

Mumport – – – ×	
Select Create new projects from an archive file or directory.	
Select an import wizard:	
type filter text	
 Ceneral Archive File Existing Projects into Workspace File System Preferences Projects from Folder or Archive C/C++ Install LAPIS 	
? < Back Next > Finish Cancel]

Select General and choose Existing Projects into Workspace. Click Next.





Import Project

M Import				- 🗆	×
Import Projects					
Select a directory to sea	rch for existing Ecl	ipse projects.			
Select root directory				× Brov	wse
Select archive file:				Brow	NSP
Projects:				DION	¥30
-				Sele	ct All
				Desel	ect All
				Ref	resh
Ortions					
Options	piects				
Copy projects into w	orkspace				
Hide projects that all	ready exist in the w	orkspace			
Working sets					
Add project to work	ing sets			New	
Working sets:				Selec	t
?	< Back	Next >	Finish	Can	cel

LEXIDE up the new window. Click Browse.. at Select root directory. Choose "Chapter 3 ADC" in Folder window.

Browse For Folder		×
Select root directory of the projects to import		
> 🖶 Downloads		^
> 📐 Favorites		
🗸 📙 lexide		
🗸 📙 Code Training Lapis		
> 📜 .metadata		
> 📜 Chapter 1 GPIO		
> 📙 Chapter 2 UART		
✓ ▲ Chapter 3 ADC		
.settings		
> 📙 Debug		
source files		
> 📜 Chapter 4 I2C		
RemoteSystemsTempFiles		
> 📙 Test		
> 📙 workspace		
N 1.1.1		~
Folder: Chapter 3 ADC		
Make New Folder	ОК	Cancel



Import Project

M Import				—		×
Import Projects Select a directory to sea	rch for existing E	clipse projects.				
Select root directory:	C:\Users\pichet	\lexide\Code Train	ing Lapis\Chap	otei ~	Browse	e
Select archive file:				~	Browse	e
Projects:						
Chapter 3 ADC (C	:\Users\pichet\le	xide\Code Training	Lapis\Chapte	r 3 AD	Select	All
				[Deselect	t All
				[Refres	h
<				>		
Options	niects					
Copy projects into w	orkspace					
Hide projects that all	ready exist in the	workspace				
Working sets						
Add project to work	ting sets				New	
Working sets:				\sim	Select	
?	< Back	Next >	Finish		Cance	I

After choosing Project Click Finish.

Import Project



Then appear the project on Project Explorer.



Check Device

🔰 Code Trainin	ig La	pis - LEXIDE-U16		
File Edit Sou	rce	Refactor Navigate Search F	Project Run LAPIS Windo	w Help
📑 🗕 🖪 👘 🕴	§ •	🔨 🕶 🗟 🕶 📮 🔌 👩 🗸	• 🚳 • 🖻 • 🞯 • 🕸 • (D - 🤷 - I
Project Explo > 5 Chapter 1 > 5 Chapter 2 x 5 Chapter 3	GPI GPI UAI	x		
> 🔊 Include		New	>	
> 🗁 Debug		Go Into		
> 🗁 source > 🗟 ML621		Open in New Window Show in Local Terminal	>	
		Сору	Ctrl+C	
	Ē	Paste	Ctrl+V	
	×	Delete	Delete	
		Source	CtrI+AIt+Shitt+Down	
		Rename	F2	
	24 24	Import Export		
	Ł	Clean and Build Project Build Project Clean Project Refresh Close Project Close Unrelated Projects	F5	
		Build Configurations	>	
		Build Targets	>	onsole 🛛 🛛
		Index	>	3 ADC]
		Show in Remote Systems view		
		Run As	>	
		Debug As	>	
		Team	>	
		Compare With	>	
		Restore from Local History		
	*	Run C/C++ Code Analysis		
		Configure	>	
		Properties	Alt+Enter	
	_		<	-

Right-click on a project folder and select [Properties] .

Pichet Sriphomma pichet@es.co.th



Select Device

Properties for Chapter 1 GP	PIO	– 🗆 X
type filter text	EXIDE-U16 Settings	⟨¬ ▼ □⟩ ▼ ▼
 > Resource Builders > C/C++ Build Build Variables Environment Logging Settings Tool Chain Editor > C/C++ General LEXIDE-U16 Settings Project References Run/Debug Settings > Task Repository WikiText 	Target MCU Category: ML62Q1000 Group: ML62Q1300 LS: ML621367 Toolchain Always use the latest Toolchain Version: V2_01_00	Restore Defaults Apply
?		Apply and Close Cancel

Choose LEXIDE-U16 Settings

Pichet Sriphomma pichet@es.co.th





Right-click on a project folder and select [Build Project] to start the build process.





Build Project



When the build succeeds , an ABS file is generated.

© 2017 ROHM Semiconductor (Thailand) Co., Ltd. © 2017 LAPIS Semiconductor Co., Ltd. All rights reserved

Pichet Sriphomma pichet@es.co.th









Pichet Sriphomma pichet@es.co.th







General Description

Channel		ML62Q13	300 group		ML62Q1500 / ML62Q1700 group											
no.	16pin product	20pin product	24pin product	32pin product	48pin product	52pin product	64pin product	80pin product	100pin product							
0	•	•	•	•	•	•	•	•	•							
1	•	•	•	•	•	•	•	•	•							
2	•	•	•	•	•	•	•	•	•							
3	•	•	•	•	•	•	•	•	•							
4	-	•	•	•	•	•	•	•	•							
5	-	•	•	•	•	•	•	•	•							
6	•	•	•	•	•	•	•	•	•							
7	•	•	•	•	•	•	•	•	•							
8	-	-	-	-	•	•	•	•	•							
9	-	-	-	_	•	•	•	•	•							
10	-	-	-	-	•	•	•	•	•							
11	-	-	-	-	•	•	•	•	•							
12	-	-	-	-	-	-	-	•	•							
13	-	-	-	-	-	-	-	•	•							
14	-	-	-	-	-	-	-	•	•							
15	-	-	-	-	-	-	-	•	•							

•: Available -: Unavailable

General Description



General Description

	Channel							ML620 gro	ຊ1300 ວັບp	L		ML62Q1500 ML62Q1700 group						
	Channel no.	Pin name	Shared port	Setting Register	Setting value	16pin product	20pin product	24pin product	32pin product	48pin product	52pin product	64pin product	80pin product	100pin product				
	0	AIN0	P17	8 th Func.	P1MOD7	0111_0000	•	•	•	•	٠	•	•	•	•			
	1	AIN1	P20	8 th Func.	P2MOD0	0111_0000	•	•	•	•	٠	•	٠	•	•			
	2	AIN2	P21	8 th Func.	P2MOD1	0111_0000	•	•	•	•	٠	•	٠	•	•			
.[3	AIN3	P22	8 th Func.	P2MOD2	0111_0000	•	•	•	•	٠	•	•	•	•			
	4	AIN4	P24	8 th Func.	P2MOD4	0111_0000	-	•	•	•	٠	•	•	•	•			
	5	AIN5	P25	8 th Func.	P2MOD5	0111_0000	-	•	•	•	٠	•	•	•	•			
	6	AIN6	P26	8 th Func.	P2MOD6	0111_0000	•	•	•	•	٠	•	•	•	•			
	7	AIN7	P27	8 th Func.	P2MOD7	0111_0000	•	•	•	•	٠	•	•	•	•			
	8	AIN8	P65	8 th Func.	P6MOD5	0111_0000	-	-	-	-	•	•	•	•	•			
	9	AIN9	P66	8 th Func.	P6MOD6	0111_0000	-	-	-	-	•	•	•	•	•			
	10	AIN10	P43	8 th Func.	P4MOD3	0111_0000	-	-	-	-	•	•	•	•	•			



SA-ADC Result Register n

Channel
The conversion result of channel 0 (AIN0)
The conversion result of channel 1 (AIN1)
The conversion result of channel 2 (AIN2)
The conversion result of channel 3 (AIN3)
The conversion result of channel 4 (AIN4)
The conversion result of channel 5 (AIN5)
The conversion result of channel 6 (AIN6)
The conversion result of channel 7 (AIN7)





Concept



Concept





BH1680FVC

This IC can detect the illuminance from 0lx to 50000lx. Supply voltage operates from 2.4V to 5.5V. Supply Current1 (Operate) 75 μ A.

Illuminance detection range [lx]	Gain Mode
~100	H-Gain Mode
~1,000	M-Gain Mode
~50,000	L-Gain Mode

M-Gain mode

$$V_{iout} = 0.61 \times 10^{-6} \times E_v \times R1$$



BD1020HFV



Temperature Sensitivity (V_{SE})

Slope =
$$\frac{1-1.9}{70-(-40)}$$

Slope =
$$-8.1818 \times 10^{-3}$$

Slope
$$\approx -8.2 [mV/C]$$



BD1020HFV

Electrical Characteristics

(Unless otherwise specified, V_{DD}=3.0V, Ta=25°C)

Baramatar	Symbol		Limit		Linit	Conditions
Farameter	Symbol	Min	Тур	Max	Unit	Conditions
		-	-	±1.5		Ta = 30°C
Accuracy	Тасс	-	-	±2.5	°C	Ta = 100°C
		-	-	±2.5		Ta = -30°C
Temperature Sensitivity	V _{SE}	-8.4	-8.2	-8.0	mV/°C	
Supply Current	ls	-	4.0	7.0	μA	
Output Voltage	Vout	1.288	1.300	1.312	V	Ta = 30°C
Output Voltage Line Regulation	ΔV outvdd	-	-	4	mV	V _{DD} = 2.4V to 5.5V
Output Voltage Load Regulation	ΔV outrl	-	-	1	mV	Difference of IOUT: 0µA/0.7µA

BD1020HFV

Output Voltage

Find linear equations :

At $T_a = 30 \text{ C} \ V_{out} = 1.3 \text{ V}$ $y - y_0 = m(x - x_0)$ $V_{out} - V_{out30C} = V_{SE} (T_a - T_{a30C})$ $V_{out} = V_{SE} (T_a - T_{a30C}) + V_{out30C}$

 $V_{out}[mV] = -8.2[mV/C](T_a[C]-30[C])+1300[mV]$



$$V_{out}[mV] = -8.2T_{a}[CmV/C] + 1546[mV]$$

$$T_{a}\left[C\right] = \frac{V_{out}\left[mV\right] - 1546\left[mV\right]}{-8.2\left[mV/C\right]}$$

$$T_{a}[C] = \frac{(-V_{out}[V] \times 1000) + 1546[mV]}{8.2[mV/C]}$$





Set ADC Pin (adc.c)	17.2	.3 Pc	ort n N	lode F	Regist	er 01	(PnM	OD01:	n=0	to 9, A	А, B)						
		PnMC of Pn0 See Ta Wirte	DO1 is pin and able 17- "0" to th	a specia l Pn1 pi 2 "List o he bits c	l function. n. of Regist of PnMO	on regis sters / B DD01 re	iter (SFI its" to c egister ti	R) to ch check av hat have	oose the aible pi no corr	e input/d ins and i respond	output n bits. ling pin.	node, in	put/outŗ	out statu	s, and s	hared fi	nction
void Set_ADC_Pin4(void)										-		-					
		15	14	13	12	11	10	9	8	1	ь	5	4	3	2	1	0
P241E=0;P240E=0;P240D=0;P24P0=0;	Word								PnM	DD01							
r24+105=0,r24+102=1,r24+101=1,r24+100=1,	Byte				PnM	OD1							PnM	OD0			
//	Bit	Pn1MD 3	Pn1MD 2	Pn1MD 1	Pn1MD 0	Pn10D	Pn1PU	Pn10E	Pn1IE	Pn0MD 3	Pn0MD 2	Pn0MD 1	Pn0MD 0	Pn0OD	Pn0PU	Pn0OE	Pn0IE
Void Set_ADC_PinS(Void)	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
P25IE=0;P250E=0;P250D=0;P25PU=0; P25MD3=0;P25MD2=1;P25MD1=1;P25MD0=1;	Initial value * ·T1	0 De initis	0 al value	0 of P00	0 IF and	0	0 L for the	0 e Port0	0 is "1" a	0 and oth	0 er bits :	0 are "0"	0	0	*	0	*
}	Pn1Ml Pn1Ml	D3 to D0	ΤΙ Γ« "Ν	his bit i or the o /L62Q 0000: 0010: 0011: 0100: 0101: 0110: 0111: 1XXX	s used letails 1500 (2 ^{nc} 3 rd 4 th 5 th 6 th 7 th 8 th	d to che of the Group imary f function function function function function function function function function	pose the shared Pin Lis function on on on on on on se (Prin	ne shai d functi st" and n (initia mary fu	red fun ion, se Table I value	ction c e Table 1-9 "M e)	of Pn1 e 1-7 " L62Q1	pin. ML62C 700 G	21300 roup P	Group in List'	Pin Li:	st", Tal	ble 1-8

X: 0 or 1 (don't care)



Initial ADC (adc.c)	23.2.14S	A-AD	C Re	eferen	ce Vo	oltage	e Con	trol R	egiste	er (VF	REFC	ON)					
<pre>void Init_ADC(void)</pre>	VRE opera	FCON ation of	is a sp f the te	ecial fu mperatu	nction ire sens	registe sor,.	r (SFR)) used t	o choos	se the in	nternal	referen	ce volta	ige ope	eration	and co	ntrol the
<pre>{ //Set_ADC_Pin0(); //Set_ADC_Pin1(); //Set_ADC_Pin2();</pre>	Addr Acce Acce Initial	Address: Access: Access size: Initial value:		DxF83A R/W Bbit Dx00	VREF	CON)											
<pre>//Set_ADC_Pin3();</pre>		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<pre>Set_ADC_Pin4();</pre>	Word									-							
<pre>Set_ADC_Pin5();</pre>	Byte					-							VREF	CON			
<pre>//Set_ADC_Pin6(); //Set_ADC_Pin7();</pre>	Bit	-	-	-	-	-	-	-	-	-	-	VREFF 1	VREFP 0	-	-	-	VREFE N
<pre>//Set_ADC_Pin8();</pre>	R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R	R/W
<pre>//Set_ADC_Pin9();</pre>	Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<pre>//Set_ADC_Pin10();</pre>	value	U	U	0	Ū	Ū	U	0	Ū	0	Ū	Ū	U	Ū	0	0	0
<pre>//Set_ADC_Pin11(); // VREECON</pre>		VREFF VREFF	P1, P0	Th	ese bit 00: Vo	s are u oltage i	sed to o nput fro	choose om the	e the re VDD p	ference in (Initia	e voltaç al valu	ge for tl e)	ne A/D	conver	sion.		
VREFEN = 0 : // Pleable	internal				10: Vc	oltage i	enerat	ed by 1	the inte	rnal ref	erence	e voltad	e circui	t (appr	ox.1.5	5V)	
VREFP1=0:VREFP0=0:	ct VDD as				11: Do	o not us	se (Volt	tage in	put fror	n the V	DD pir	ו)				,	
······	<u>-</u>	-		Re	served	l bit											
<pre>// SADMOD SALP = 0; // Single c SACK2=0:SACK1=0:SACK0=0:</pre>	onversion // At PI	VREFE	EN	Th sei the	is bit is nsor. W VREF	used f /hen us EN bit	to enab sing the to "1".	le the intern	operatio al refer	on of in rence v	ternal oltage	referen (appro:	ce volta k. 1.55∖	age and /) or te	d the te mperat	empera ure se	ature nsor, set
// Cloc	k <u>Peroid</u>				0: Di: (In	sable t nitial va	he opei lue)	ration o	of interr	nal refe	rence	voltage	and ter	nperat	ure sei	nsor	
SASHT3=1;SASHT2=0;SASHT1=1; //	SASHT0=0; this sett				1: Er	nable th	ne oper	ation o	ot intern	al refer	ence \	/oltage	and ten	nperati	ure ser	isor	
SAINIT = 1; // Discharg	e sample b																
// SADENØ																	

© 2017 ROHM Semiconductor (Thailand) Co., Ltd. © 2017 LAPIS Semiconductor Co., Ltd. All rights reserved

}

Ir



lai ADC (auc.c)	20.2.0 0			Juen	eyisi			<i>(</i>)									
d Init_ADC(void)	SAD] conve	MOD i erter. T	is a spe The bit	ecial fur symbol	nction i l "rsvd'	register " means	(SFR) a reser	used to ved bit	set the , write	operat "0" to t	ion mod hose bit	le and ts.	operati	ing cloc	ek frequ	ency of	f the A
<pre>//Set_ADC_Pin0(); //Set_ADC_Pin1(); //Set_ADC_Pin2();</pre>	Addre Acces Acces Initial	ess: ss: ss size value:	C F 8 0	0xF828 R/W 8/16bit 0x0000	(SADM	IODL/S <i>i</i>	ADMOE	0), 0xF8	829(SA	DMOD	⊣)						
<pre>//Set ADC Pin3();</pre>		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Set ADC Pin4();	Word								SAD	MOD							
Set_ADC_Pin5();	Byte				SAD	MODH							SADI	MODL			
<pre>//Set_ADC_Pin6(); //Set ADC Pin7();</pre>	Bit	rsvd	rsvd	rsvd	rsvd	rsvd	rsvd	rsvd	SAINIT	SASHT 3	SASHT 2	SASHT 1	SASHT 0	SACK2	SACK1	SACK0	SALP
<pre>//Set_ADC_Pin8();</pre>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<pre>//Set_ADC_Pin9();</pre>	Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<pre>//Set_ADC_Pin10();</pre>	value																
<pre>//Set_ADC_Pin11();</pre>		SACI SACI	K2 to K0	(These SAD_(bits are CLK). S	used t See Cha	o choc apter 2	se the 4.3.2 "/	freque A/D Co	ncy of t nversio	he A/[n Tim	D conve e Settii	ersion on ng" for	operatii the ope	ng clocl erating	k clock,
// VREFCON				C		sion tim 8MI	Hz (Init	ial valu	e ume. Je)								
VREFEN = 0; // Disable in	nternal				001:	4MI	Hz	iai vaic)								
VREFP1=0;VREFP0=0; // Select	t VDD as				010:	2M	Hz										
					011:	1M	Hz										
// SADMOD					100:	0.5	MHz										
SALP = 0; // Single	ersion				101:	Do	not use	9									
// Clock	Peroid				110:	Do 32k	not use Hz	•									
SASHT3=1; SASHT2=0; SASHT1=1; SASHT1	A SHT0=0; this set	SALF	D	-	This bit channe	t is used al or cor	d to cho nsecutiv	oose w vely. T	hether	the A/I version	D conve interva	ersion I time	is perfo in the	ormed consec	once o utive s	nly for e can A/[each D
SAINIT = 1; // Discharge	sample			C	onver: 0:	sion mo Single	ode is s A/D coi	pecifie nversio	ed in the on (Initia	e SADS al value	STM reg e)	gister.					
// SADENØ					1:	Consec	cutive s	can A/	/D conv	rsion							

. -

© 2017 ROHM Semiconductor (Thailand) Co., Ltd. © 2017 LAPIS Semiconductor Co., Ltd. All rights reserved

}

Ir



nit	ial ADC (adc.c) ^{23.2}	2.6 S	A-AD	C M	ode R	egist	er (SA	ADMC) (DC									
void	I Init_ADC(void)	SAD conv	MOD 1 erter. T	s a spe he bit	symbol	rsvd"	register ' means	(SFR) a reser	used to ved bit	set the , write	operat "0" to t	ion mo hose b	de and its.	operati	ng cloc	k frequ	ency of	the A/D
{	<pre>//Set_ADC_Pin0(); //Set_ADC_Pin1(); //Set_ADC_Pin2();</pre>	Addr Acce Acce Initia	ess: ss: ss size l value:	(0xF828 R/W 8/16bit 0x0000	(SADM	ODL/SA	ADMOE	0), 0xF8	329(SA)	DMOD	H)						
	<pre>//Set_ADC_Pin3();</pre>		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	<pre>Set_ADC_Pin4();</pre>	Word								SAD	MOD							
	<pre>Set_ADC_Pin5();</pre>	Byte				SADI	MODH							SAD	MODL			
	<pre>//Set_ADC_Pin6(); //Set_ADC_Pin7();</pre>	Bit	rsvd	rsvd	rsvd	rsvd	rsvd	rsvd	rsvd	SAINIT	SASHT 3	SASHT 2	SASHT	SASHT 0	SACK2	SACK1	SACK0	SALP
	<pre>//Set_ADC_Pin8();</pre>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	<pre>//Set_ADC_Pin9(); //Set_ADC_Pin10();</pre>	Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	<pre>//Set_ADC_Pin11(); // VREFCON VREFEN = 0; // Disable inter VREFP1=0;VREFP0=0; // Select VE // SADMOD</pre>	IT		This bi sample conver 0: 1:	t is use hold sion. Start ti sample Start ti sample	ed to c capaci he A/D e hold he A/D e hold	ontrol v tor on f conve capaci conve capaci	whethe the pre trsion v tor (Ini tor a tor	er or no evious / without tial val after dis	ot to di A/D cc discha ue) scharg	scharg nversi arging ing the	e the e ion, be the ele e elect	electric fore st ectrica rical cl	al char arting t charge narge a	rge ren he nex e accu	nained t SA-A mulate	in the DC d in the	
	SALP = 0; // Single conver	SASH	IT3 to		These	bits ar	e used	to set	the sa	ampling	, time.							
	SACK2=0; SACK1=0; SACK0=0; // /	SASH	1T0		See C	napter	24.3.2	"A/D (Convei	rsion T	ime Se	etting"	for det	tails.				
	<pre>// Clock Per SASHT3=1;SASHT2=0;SASHT1=1;SASHT // this SAINIT = 1; // SADEN0</pre>	roid F0=0; s set nple	= :t }															
}																		



Conversion time

Figure 23-6 shows the operation waveforms when the continuous A/D conversion is performed using channel 0.




Conversion time

Table 23-3	A/D Conversion	time when using	V_{DD} or V_{REF}	pin as	reference voltage
------------	----------------	-----------------	-------------------------------------	--------	-------------------

SADMOD					0			Convers	ion time ^{*1}		
		5AD	NOD		clock count			SAD	_CLK		
		SASH	T[3:0]			32kHz	0.5MHz	1MHz	2MHz	4MHz	8MHz
	0	0	0	0	14	427 µs	28 µs	Prohibited	Drobibitod		
	0	0	0	1	15		30 µs	15 µs	FIONDILEU	Prohibited	Prohibited
	0	0	1	0	16		32 µs	16 µs	8 µs		FIOIIDILEU
	0	0 0 1 1 0 1 0 0	1	17		34 µs	17 µs	8.5 µs	4.25 µs		
	0		0	18		36 µs	18 µs	9 µs	4.5 µs	2.25 µs	
	0	1	0	1 1	19	_	38 µs	19 µs	9.5 µs	4.75 μs	2.375 µs
	0	1	1	0	20		40 µs	20 µs	10 µs	5 µs	2.5 µs
	0	1	1	1	21		42 µs	21 µs	10.5 µs	5.25 µs	2.625 µs
	1	0	0	0	29	Prohibited	58 µs	29 µs	14.5 µs	7.25 µs	3.625 µs
	1	0	0	1	45		90 µs	45 µs	22.5 µs	11.25 µs	5.625 µs
	1	0	1	0	61		122 µs	61 µs	30.5 µs	15.25 µs	7.625 µs
	1	0	1	1	77		154 µs	77 µs	38.5 µs	19.25 µs	9.625 µs
	1	1	0	0	93		186 µs	93 µs	46.5 µs	23.25 µs	11.625 µs
	1	1 1 0 1 1 1 1 0 1	1	109		218 µs	109 µs	54.5 µs	27.25 µs	13.625 µs	
	1		0	125		250 µs	125 µs	62.5 µs	31.25 µs	15.625 µs	
	1	1	1	1	141		282 µs	141 µs	70.5 µs	35.25 µs	17.625 µs

*1: The A/D conversion time does not include discharging time (two clocks of the SAD_CLK) and the clock frequency error.



Sampling time

	240	MOD		Sample/			Sampli	ng time		
	SAD	NOD		hold			SAD	CLK		
	SASH	IT[3:0]		clock count	32kHz	0.5MHz	1MHz	2MHz	4MHz	8MHz
0	0	0	0	1	30 µs	2 µs	Prohibited	Drahibitad		
0	0	0	1	2		4 µs	2 µs	Prohibiled	Prohibited	Drobibited
0	0	1	0	3		6 µs	3 µs	1.5 µs		Prohibited
0	0	1	1	4		8 µs	4 µs	2 µs	1 µs	
0	1	0	0	5		10 µs	5 µs	2.5 µs	1.25 µs	0.625 µs
0	1	0	1	6		12 µs	6 µs	3 µs	1.5 µs	0.75 µs
0	1	1	0	7		14 µs	7 µs	3.5 µs	1.75 µs	0.875 µs
0	1	1	1	8		16 µs	8 µs	4 µs	2 µs	1 µs
1	0	0	0	16	Prohibited	32 µs	16 µs	8 µs	4 µs	2 µs
1	0	0	1	32		64 µs	32 µs	16 µs	8 µs	4 µs
1	0	1	0	48		96 µs	48 µs	24 µs	12 µs	6 µs
1	0	1	1	64		128 µs	64 µs	32 µs	16 µs	8 µs
1	1	0	0	80		160 µs	80 µs	40 µs	20 µs	10 µs
1	1	0	1	96		192 µs	96 µs	48 µs	24 µs	12 µs
1	1	1	0	112		224 µs	112 µs	56 µs	28 µs	14 µs
1	1	1	1	128		256 µs	128 µs	64 µs	32 µs	16 µs

Table 23-5 Sampling time when using V_{DD} or V_{REF} pin as reference voltage



Sampling Time Setting



Sampling time $> 8C_{SAMPLE}R_2$



Scan ADC (adc.c)

```
void ScanADC(void)
{
```

```
static unsigned char index=0;
```

```
//GetADC(0,index);
//GetADC(1,index);
//GetADC(2,index);
//GetADC(3,index);
GetADC(4,index);
GetADC(5,index);
//GetADC(6,index);
//GetADC(6,index);
//GetADC(7,index);
//GetADC(8,index);
//GetADC(9,index);
//GetADC(10,index);
//GetADC(11,index);
```

Receive ADC value from Channel 4,5

}



Get ADC (adc.c)

void GetADC(unsigned char ch, unsigned char id)

unsigned int result;

{

}

```
SADEN0 = ADC_CH[ch];
SARUN = 1;
__asm("nop");
__asm("nop");
__asm("nop");
__asm("nop");
while(SARUN == 1)
__asm("nop");
```

```
result = Get_ADC_Result(ch);
ADC_Buf[ch][id] = result>>6;
ADC_Value[ch] = Average8(ch);
```

23.2.8 SA-ADC Enable Register 0 (SADEN0)

١

SADEN0 is a special function register (SFR) used to choose channels of the A/D converter and enable/disable the conversion.

Addr Acce Acce Initia	ess: ess: ess size l value:		0xF82C R/W 8/16bit 0x0000	(SADE	NOL/SA	DEN0)), 0xF82	2D(SAD	EN0H)							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Vord								SAD	EN0							
Byte				SAD	EN0H							SAD	EN0L			
Bit	SACH1 5	SACH [·] 4	1 SACH1 3	SACH1 2	SACH1 1	SACH1 0	SACH0 9	SACH0 8	SACH0 7	SACH0 6	SACH0 5	SACH0 4	SACH0 3	SACH0 2	SACH0 1	SACH0 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
nitial /alue	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	SACH1	15 to 10	Th en S S S S S S S S S S S S S S S S S S	ese bi able/d ACH00 ACH02 ACH02 ACH02 ACH02 ACH02 ACH02 ACH02 ACH02 ACH02 ACH1	ts are isable isable the isable <tdthe< td=""> isable</tdthe<>	used the con nable con	to cho provers or Disal or Disal	ose ch ion. ble the ble the ble the ble the ble the ble the ble the ble the ble the ble the ble the ble the ble the ble the ble the bl	A/D co A/D co	n (n= nversio nversio nversio nversio nversio nversio nversio nversio nversio nversio nversio nversio nversio nversio nversio nversio nversio	D to 15 on on c on on c	5) of the hanne hanne hanne hanne hanne hanne hanne hanne hanne hanne hanne	ne A/D 0 1 2 3 4 5 6 7 8 1 1 12 12 12 13 14 15	conve	erter a	nd



Get ADC (adc.c)

void GetADC(unsigned char ch, unsigned char id)

unsigned int result;

{

}



```
result = Get_ADC_Result(ch);
ADC_Buf[ch][id] = result>>6;
ADC_Value[ch] = Average8(ch);
```

23.2.7 SA-ADC Control Register (SADCON)

SADCON is a special function register (SFR) used to control the operation of the A/D converter.

Addr Acce Acce Initia	ess: ess: ess size: I value:	0 F 8 0)xF82A(R/W 5/16bit 0x0000	SADC	ONL/SA	ADCON	I), 0xF8	32B(SA	DCON	H)						
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word								SAD	CON							
Byte				SADO	CONH							SADO	CONL			
Bit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SATGE N	SARU N
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SARUN

This bit is used to start or stop the A/D conversion.

Write "1" to this bit to start the A/D conversion, and "0" to stop it.

When "0" is written to SALP bit and the A/D conversion on the largest number of channel is ended, this SARUN bit is automatically reset to "0".

When "1" is written to SALP, the A/D conversion repeats until the SARUN bit is reset to "0" by the software.

- 0: Stop the A/D conversion (Initial value)
- 1: Start the A/D conversion

ROHM GROUP LAPIS ROHIT SEMICONDUCTOR



© 2017 ROHM Semiconductor (Thailand) Co., Ltd. © 2017 LAPIS Semiconductor Co., Ltd. All rights reserved



Get ADC Result (adc.c)

23.2.2 SA-ADC Result Register n (SADRn : n=0 to 15, 16)

SADRn is a special function register (SFR) used to store the SA-ADC conversion results on channels 0 to 15 and channel 16 (temperature sensor).

The A/D conversion result of each channel can be read from SADRn.

unsigned int Get_ADC_Result(unsigned char cm)

```
switch (ch)
{
    case 0: return(SADR0);
    case 1: return(SADR1);
    case 2: return(SADR2);
    case 3: return(SADR3);
    case 4: return(SADR4);
    case 5: return(SADR5);
    case 6: return(SADR5);
    case 7: return(SADR7);
    default: return 0;
```

{

}

Symbol name	Channel
SADR0	The conversion result of channel 0 (AIN0)
SADR1	The conversion result of channel 1 (AIN1)
SADR2	The conversion result of channel 2 (AIN2)
SADR3	The conversion result of channel 3 (AIN3)
SADR4	The conversion result of channel 4 (AIN4)
SADR5	The conversion result of channel 5 (AIN5)
SADR6	The conversion result of channel 6 (AIN6)
SADR7	The conversion result of channel 7 (AIN7)

Acce Acce Initia	ess: ess size l value:	F : 8 0	8 /16bit x0000													
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word								SAI	DRn							
Byte				SAD	RnH							SAD	RnL			
Bit	d15	d14	d13	d12	d11	d10	d9	d8	d7	d6	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



Average8 (adc.c)

```
unsigned int Average8(unsigned char ch)
{
    unsigned char i;
    unsigned int sum=0;
    for(i=0;i<8;i++)
        sum += ADC_Buf[ch][i];
    return(sum /8);
}</pre>
```

Find the average of ADC value all 8 values



BD1020 Read (adc.c)

```
void BD1020_Read_Temp(void){
```

```
unsigned int vin;
float TIN;
```

```
Real_ADC4 = TIN;
```

}

 $T_{a}[C] = \frac{(-V_{out}[V] \times 1000) + 1546[mV]}{8.2[mV/C]}$



```
© 2017 ROHM Semiconductor (Thailand) Co., Ltd.
© 2017 LAPIS Semiconductor Co., Ltd. All rights reserved
```



BH1680 Read (adc.c)

```
void BD1680_Read_ALS(void){
```

unsigned int vin; float EV; float R1 = 11000; float H_const = 6.1; float M_const = 0.61;

// Calculations for ALS - BD1680
// Math: ADC_Voltage = sensorValue * `(VCC/4096)
// H-Gain Mode: Viout = 6.1* (10^(-6)) * EV *R1
// M-Gain Mode: Viout = 0.61* (10^(-6)) * EV *R1
// L-Gain Mode: Viout = 0.061* (10^(-6)) * EV *R1

EV = (vin*((float)Vcc/1024)/(R1*M_const*0.000001));

```
//EV = (((vin*((float)Vcc/4096))/R1)/(H_const*0.000001));
Real_ADC5 = EV;
```

}



$$E_{\rm v} = \frac{V_{\rm iout}}{0.61 \times 10^{-6} \times R1}$$

$$V_{iout} = V_{in} \frac{V_{CC}}{1024}$$



Concept



ST,SL <Hundreds – Thousands><Tens><decimal>



UART00 BD1020 (uart0.c)

```
void UART00_BD1020(void)
```

```
2 ADC4_int = ADC4_float;
3 ADC4_point = (ADC4_float - ADC4_int)*100;
```

```
4 ADC4_Char_High = (ADC4_int & 0xFF00)>>8;
```

```
5 ADC4_Char_Low = ADC4_int & 0x00FF;
```

```
Clear_Buffer();
```

```
UART00_TX_Buf[0] = ADC4_Char_High;
UART00_TX_Buf[1] = ADC4_Char_Low;
UART00 TX Buf[2] = ADC4 point;
```

```
TXD00_Index = 0;
Flag._TXD00 = 1;
Sensor_Index = 1;
Set UART00 TX();
```

```
RXD00_Index = 0;
//Flag._RXD00 = 0;
```

}

ADC Value of BD1020 (Real_ADC4) = 30.24 1. ADC4_float = 30.24

2. $ADC4_int = 30$

Example

3. ADC4_point = (30.24 - 30)*100 = 24

4. ADC4_Char_High = 0x0030 & 0xFF00 = 0x0000

= 0x0000 >> 8 = 0x0000

5. ADC4_Char_Low = 0x0030 & 0x00FF = 0x0030

UART_TX_Buf[0] = 0 UART_TX_Buf[1] = 30 UART_TX_Buf[2] = 24



UART00 BH1680 (uart0.c)

```
void UART00_BH1680(void)
```

```
unsigned int i=0;
  Sensor_Index = 0;
  //*********** Ambient Light Sensor
1 ADC5_float = Real_ADC5;
```

```
2 ADC5 int = ADC5 float;
3 ADC5 point = (ADC5 float - ADC5 int)*100;
```

```
4 ADC5 Char High = (ADC5 int & 0xFF00)>>8;
```

```
5 ADC5_Char_Low = ADC5_int & 0x00FF;
```

```
Clear_Buffer();
```

```
UART00 TX Buf[0] = ADC5 Char High;
UART00 TX Buf[1] = ADC5 Char Low;
UART00 TX Buf[2] = ADC5 point;
TXD00 Index = 0;
Flag. TXD00 = 1;
Sensor Index = 1;
Set UART00 TX();
```

```
RXD00 Index = 0;
//Flag. RXD00 = 0;
```

}

Example

ADC Value of BH1680 (Real ADC5) = 240.68

1. ADC5 float = 240.68

2. ADC5 int = 240

3. ADC5 point = $(240.68 - 240) \times 100 = 68$

4. ADC5 Char High = 0x0240 & 0xFF00 = 0x0200

 $= 0 \times 0200 >> 8 = 0 \times 0002$

5. ADC5 Char Low = 0x0240 & 0x00FF = 0x0040

UART TX Buf[0] = 2UART TX Buf[1] = 40UART TX Buf[2] = 68

Connection ES-ICD-V1 ,Sensor Shield and Module



ROHM GROUP

LAPIS

ROHM



Using the application

Step 1 Connect MCU Board with Bluetooth



© 2017 ROHM Semiconductor (Thailand) Co., Ltd. © 2017 LAPIS Semiconductor Co., Ltd. All rights reserved



Using the application

Step 2 Open application and click "Bluetooth Check"





Using the application

Step 3 Choose device and connect





Using the application

Step 4 After connected this app Show button to choose



Using the application

Step 5 if you choose "Temperature : BD1020" this app show Signal graph Of BD1020 (picture 1) but you choose "Ambient Light : BH1680" this app show signal graph of BH1680 (picture 2)







Import Project



Right-click on project Explorer and select Import.

Import Project

Milmport – 🗆	×
Select Create new projects from an archive file or directory.	5
Select an import wizard:	
type filter text	
 General Archive File Existing Projects into Workspace File System Preferences Projects from Folder or Archive C/C++ Install LAPIS 	~
? < Back Next > Finish Cancel	

Select General and choose Existing Projects into Workspace. Click Next.





Import Project

Import Projects Select a directory to sea	arch for existing Ec	lipse projects.		\square
,				
Select root directory:			~	Browse
O Select archive file:			~	Browse
Projects:				
				Select All
				Deselect All
				Refresh
Options				
Options	ojects			
Options Search for nested pr Copy projects into w Hide projects that a	ojects vorkspace Iready exist in the	workspace		
Options Search for nested pr Copy projects into w Hide projects that a Working sets	ojects vorkspace Iready exist in the t	workspace		
Options Search for nested pr Copy projects into w Hide projects that a Working sets Add project to wor	ojects vorkspace Iready exist in the v	workspace		New
Options Search for nested pr Copy projects into w Hide projects that al Working sets Add project to work Working sets:	ojects vorkspace Iready exist in the v king sets	workspace		New Select
Options Search for nested pr Copy projects into w Hide projects that a Working sets Add project to worl Working sets:	ojects vorkspace Iready exist in the king sets	workspace		New Select
Options Search for nested pr Copy projects into w Hide projects that a Working sets Add project to worf Working sets:	ojects vorkspace Iready exist in the king sets	workspace		New Select

LEXIDE up the new window. Click Browse.. at Select root directory. Choose "Chapter 4 I2C" in Folder window.

E E	Browse For Folder	×
	Select root directory of the projects to import	
ιR		
	> 🖊 Downloads	^
	> 📙 Favorites	
	✓	
	🗸 📙 Code Training Lapis	
	> 📙 .metadata	
	> 📙 Chapter 1 GPIO	
	> 📙 Chapter 2 UART	
	> 📙 Chapter 3 ADC	
	🗸 📙 Chapter 4 I2C	
d	.settings	
to	> 📙 Debug	
a	source files	
-	RemoteSystemsTempFiles	
w	> 📜 Test	
	> 📜 workspace	
		•
	Folder: Chapter 4 I2C	
	Make New Folder OK Cancel	

Import Project

🚺 Import						×
Import Projects Select a directory to sea	rch for existing E	clipse projects.				
 Select root directory: Select archive file: Projects:	C:\Users\pichet	\lexide\Code Train	ing Lapis\Chapteı	×	Browse	
Chapter 4 I2C (C:\	Users\pichet\lexi	de\Code Training	.apis\Chapter 4 I2(Select A Deselect Refres	All All h
 Options Search for nested pro Copy projects into we 	ojects orkspace			>		
Working sets Add project to work Working sets:	ing sets	workspace	~		New	
?	< Back	Next >	Finish		Cancel	

After choosing Project Click Finish.

Import Project



Then appear the project on Project Explorer.



Check Device

🔰 Code	Trair	ning Lap	is - LEXII	DE-U16						
File Edi	t S	ource l	Refactor	Navigat	e Search	Project	t Run	LAPIS	Window	Help
📑 🗕 🖪	0	• 🛞	«S 🗶 🗟	• ۹	🖻 🔌 👔	🕯 🕶 🚳 י	• 🖒 •	ତ -	* • 0	- 9
Projec	t Exp	olorer 🛛	3 0	\$	~ - 6	3				
> 🍰 Ch	apte	r 1 GPIC)							
i > 😂 Chi	apte	r 2 UAR	ſ							
✓ 2 Ch	apie	I S ADC								
> 🔊		New Go Into	2					>		
> 🗁 > 🔝		Open i Show i	n New W n Local T	/indow erminal				>		
		Сору						Ctrl+C		
		Paste						Ctrl+V		
	<u>.</u>	Remov	e from C	ontext		Ctrl+Alt	+Shift+	Down		
		Source						>		
		Move						F2		
	p va	Import						12	-	
	4	Export								
		Clean a	and Build	Project						
		Build P	roject							
	ন্ধ্য	Refrest	project h					F5		
		Close F	Project							
		Close (Jnrelated	l Projects					_	
		Build C	onfigura	tions				>	El Cor	sole ?
		Build I	argets					>	apter 4	12C]
		Show i	n Remot	e Systems	view				-	
		Run As	;	,				>		
		Debug	As					>		
		Profile	As					>		
		Compa	are With							
		Restor	e from Lo	ocal Histo	ry					
	*	Run C/	C++ Coo	le Analysi	S			K,		
		Proper	ties				Alt	+Enter		
		Toper					7.40	21100		

Right-click on a project folder and select [Properties] .



Select Device

LEXIDE-U16 Settings	
Resource Builders C/C++ Build Build Variables Environment Logging Settings Tool Chain Editor C/C++ General LEXIDE-U16 Settings Project References Run/Debug Settings Task Repository WikiText	Restore Defaults Apply

Choose LEXIDE-U16 Settings

Build Project



Right-click on a project folder and select [Build Project] to start the build process.



Build Project



When the build succeeds , an ABS file is generated.







Initial i2c on (i2c.c)

#define #define	SDA_OUT SDA_IN	P22D0 P22DI	
#define #define	SDA_DIR_IN SDA_DIR_OUT	P22IE P220E	
#define #define	SCL_OUT SCL_IN	P23D0 P23DI	
#define	SCI DTR TN	P23TF	
#define	SCL_DIR_OUT	P230E	
void ini	tial_i2c_on(void){		
P22IE	=0; P220E=0; P220D=0; P22Pl	J=1;	1
12510	-0,12302-0,12300-0,12310	-1,	7
SDA_DJ SDA_DJ SCL_DJ	IR_IN=0; IR_OUT=1; IR_OUT=1;		_

17.2.2 Port n Data Register (PnD:n=0 to 9, A, B)

	PnD) is a sp	pecial	functio	n regis	ter (SI	FR) use	ed to re	ead the	level	of the p	port n j	pin and	l write	output	t data.		
	The input level of the port n pin can be read by reading PnDI in the input mode.																	
	Data written to PnDO in the output mode are output to the port n pin.																	
	The PnDO is readable.																	
	Enable or disable the input or output by using the port n mode register.																	
	See Table 17-2 "List of Registers / Bits" to check avaible pins and bits.																	
	Wir	te "0" 1	to the l	oits of	PnDO	registe	er that I	have n	o corre	espond	ing pin	ı.						
	The bits of PnDI register that has no corresponding pin always return "0" for reading.																	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Word	i PnD																	
Byte	PnDO									PnDI								
Bit	Pn7DO	Pn6DO	Pn5DO	Pn4DO	Pn3DO	Pn2DO	Pn1DO	Pn0DO	Pn7DI	Pn6DI	Pn5DI	Pn4DI	Pn3DI	Pn2DI	Pn1DI	Pn0DI		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R		
Initial	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1		
value	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	'		
Bit	Bit symbol Description																	
No.	name																	
15 to 8	Pn7DO to This bit is used to set the output level of port n pin.																	
	Pn0DO 0: Output "L" (initial value)																	
				1:	Output	"H"												
	Pn7DI to This bit is used to set the input level of port n pin.																	
7 to 0	Pn7	DI to		i nis dit	is used	110 361	and mp			ii piii.								
7 to 0	Pn7 Pn0	DI to		0:	The inp	ut level	is "L"		orport	n pin.								
7 to 0	Pn7 Pn0	DI to DI		0: 1:	The inp The inp	ut level ut level	is "L" is "H"	(Initial v	value)	n pin.								

}



Initial i2c on (i2c.c)

PnMOD01 is a special function register (SFR) to choose the input/output mode, input/output status, and shared function of Pn0 pin and Pn1 pin. See Table 17-2 "List of Registers / Bits" to check avaible pins and bits. #define SDA OUT P22D0 Wirte "0" to the bits of PnMOD01 register that have no corresponding pin. #define SDA IN P22DI 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 Word PnMOD01 #define SDA DIR IN P22IE Byte PnMOD1 PnMOD0 #define SDA DIR OUT P220E Pn0MD Pn0MD Pn0MD Pn0MD Pn1MDPn1MDPn1MDPn1MD Pn1OD Pn1PU Pn1OE Pn1IE Pn0ODPn0PUPn0OE Pn0IE Bit 2 0 3 2 0 3 #define SCL OUT P23D0 #define SCL IN P23DT R/W Initial 0 0 0 0 0 0 0 0 Λ 0 0 0 0 0 value * The initial value of P00IE and P00PU for the Port0 is "1" and other bits are "0". P23IE #define SCL DIR IN #define SCL DIR OUT P230E Pn10D This bit is used choose the output type of Pn1 pin. An LED is directly drive-able by enlarging the current when the N-channel open drain output mode is chosen. void initial_i2c_on(void){ See the data sheet for details about the current drive ability. 0: CMOS output (initial value) 1: N-channel open drain output P22IE=0; P220E=0; P220D=0; P22PU=1; P23IE=0; P230E=0; P230D=0; P23PU=1; Pn1PU This bit is used to enable the internal pull-up resistor of Pn1 pin. The internal pull-up resistor can be enabled on following conditions of the port. The input is enabled and the output is disabled on the port SDA DIR IN=0; The input is enabled and the N-channel open drain output is chosen on the port SDA DIR OUT=1; 0: Without a pull-up resistor (initial value) SCL DIR OUT=1; 1: With a pull-up resistor The conditions of the port are specified by Pn1IE, Pn1OE and Pn1OD bit. 10X: Setting of Pn1PU bit is enable 111: Setting of Pn1PU bit is enable Others: Setting of Pn1PU bit is disable X: 0 or 1 (don't care)

17.2.3 Port n Mode Register 01 (PnMOD01:n=0 to 9, A, B)



i2c start (i2c.c)

<pre>void i2c_start(void){</pre>
SDA_OUT=1;
SCL_OUT=1;
i2c_delay();
SDA_OUT=0;
i2c_delay();
SCL_OUT=0;
}

i2c stop (i2c.c)

```
void i2c_stop(void){
    i2c_delay();
    SDA_OUT=0;
    i2c_delay();
    SCL_OUT=1;
    i2c_delay();
    SDA_OUT=1;
}
```






i2c ACK NACK (i2c.c)





KX023

- 1. tri-axis +/-2g, +/-4g or +/-8g accelerometer
- 2. Enhanced integrated Directional Tap/Double-TapTM
- , and Device-orientation Algorithms
- 3. Digital I2C up to 3.4 MHz
- 4. Digital 3-wire and 4-wire SPI up to 10 MHz
- 5. Self-test Function

KX023



Pin	Name
1	IO Vdd
2	NC
3	NC
4	SCLK/SCL
5	GND
6	SDI/SDA
7	SDO/ADDR
8	nCS
9	INT2
10	NC
11	INT1
12	GND
13	TRIG
14	Vdd
15	NC
16	NC



KX023

Static	<u> X/Y/Z OU</u>	itput i	<u> Kespons</u>	e vers	<u>sus Orie</u>	entat	on to E	artn's	<u>s surra</u>	ce (1	<u>g):</u>	
			GSEL ²	1=0, G	SEL0=	0 (± 2	g)					
Position	1		2	2		3		4			6	
										Тор		1
Diagram									Bottom		Тор	
Resolution (bits)	16	8	16	8	16	8	16	8	16	8	16	8
X (counts)	0	0	-16384	-64	0	0	16384	64	0	0	0	0
Y (counts)	-16384	-64	0	0	16384	64	0	0	0	0	0	0
Z (counts)	0	0	0	0	0	0	0	0	16384	64	-16384	-64
X-Polarity	0		-	-			+		0		0	
Y-Polarity	-	-		0		+		0		0		
Z-Polarity	0		0		0		0		+		-	

Static output response = output value×g_ratio = $-16384 \times \frac{2}{32768} = -1$



KX023 I2C Operation

The Slave Address associated with the KX023 is 001111X.

X is determined by the assignment of ADDR (pin 7) to GND or IO_Vdd.

Term	Definition
S	Start Condition
Sr	Repeated Start Condition
SAD	Slave Address
W	Write Bit
R	Read Bit
ACK	Acknowledge
NACK	Not Acknowledge
RA	Register Address
Data	Transmitted/Received Data
Р	Stop Condition



Writing to a KX023 8-bit Register

- 1. Master send Start condition (S) and SAD+W ,and the KX023 acknowledges. The KX023 return ACK.
- 2. After ACK Transmission. An 8-bit Register Address (RA) command is transmitted by the Master. The KX023 return ACK.
- 3. After Master receive ACK from KX023. Master Send DATA to KX023.
- 4. The KX023 Return ACK. ,and Waiting to receive Stop condition (P) from Master.

Sequence 1. The Master is writing one byte to the Slave.

Master	S	SAD + W		RA		DATA		Р
Slave			ACK		ACK		ACK	

Sequence 2. The Master is writing multiple bytes to the Slave.

Master	S	SAD + W		RA		DATA		DATA		Ρ
Slave			ACK		ACK		ACK		ACK	



Reading from a KX023 8-bit Register

- 1. The Master first transmits a start condition (S) and the appropriate Slave Address (SAD) with the LSB set at '0' to write. The KX023 return ACK.
- 2 The KX023 acknowledges and the Master transmits the 8-bit RA of the register it wants to read.
- 3. The KX023 again acknowledges, and the Master transmits a repeated start condition (Sr).
- 4. After the repeated start condition, the Master addresses the KX023 with a '1' in the LSB (SAD+R) to read.
- 5. The KX023 acknowledges and transmits the data from the requested register.
- 6. The Master does not acknowledge (NACK) it received the transmitted data , and transmits a stop condition to end the data transfer.

Sequence 3. The Master is receiving one byte of data from the Slave.

Master	S	SAD + W		RA		Sr	SAD + R			NACK	Ρ
Slave			ACK		ACK			ACK	DATA		

Sequence 4. The Master is receiving multiple bytes of data from the Slave.

Master	S	SAD + W		RA		Sr	SAD + R			ACK		NACK	Ρ
Slave			ACK		ACK			ACK	DATA		DATA		



KX023 Embedded Registers

The KX023 has 57 embedded 8-bit registers that are accessible by the user.

XHPL, XHPH, YHPL, YHPH, ZHPL, ZHPH high pass filter accelerometer output. Data is updated at the ODR frequency determined by OWUF in CNTL3.

Address	Register Name	R/W
00h	XHPL	R
01h	XHPH	R
02h	YHPL	R
03h	YHPH	R
04h	ZHPL	R
05h	ZHPH	R



KX023 Embedded Registers

XOUTL,XOUTH,YOUTL,YOUTH,ZOUTL,ZOUTH accelerometer output . Data is updated at the ODR frequency determined by OSA in ODCNTL.

Address	Register Name	R/W
06h	XOUTL	R
07h	XOUTH	R
08h	YOUTL	R
09h	YOUTH	R
0Ah	ZOUTL	R
0Bh	ZOUTH	R

KX023 Embedded Registers

COTR This register can be used to verify proper integrated circuit functionality. It always has a byte value of 0x55h

WHO_AM_I This register can be used for supplier recognition, as it can be factory written to a known byte value. The default value is 0x15h.

Address	Register Name	R/W
0Ch	COTR	R
0Dh	Kionix Reserved	
0Eh	Kionix Reserved	
0Fh	Who_AM_I	R/W

Initial KX023 (kx023.c)



KX112_Start();



Master	S	SAD + W		RA		Sr	SAD + R			NACK	Ρ
Slave			ACK		ACK			ACK	DATA		



Write KX023 (kx023.c)

void write_kx(unsigned char addr,unsigned char data){

```
unsigned char read;
i2c_start();
i2c_write(kx_Addr);
i2c_write(addr);
i2c_write(data);
i2c_stop();
```

}



Master	S	SAD + W		RA		DATA		Ρ
Slave			ACK		ACK		ACK	



MulRead KX023 (kx023.c)

unsigned char Mulread_kx(unsigned char addr,unsigned char *data,unsigned int size){

unsigned int i,DataLen;

//initial_i2c_on();

DataLen = size-1;

```
i2c_start();
i2c_write(kx_Addr);
i2c_write(addr);
i2c_start();
i2c_write(kx_Addr+1);
for(i=0;i<DataLen;i++){
   rdata[i] =i2c_read();
   i2c_ack_bit();
}
```

```
rdata[DataLen] =i2c_read();
i2c_nack_bit();
i2c_stop();
```

```
for(i=0;i<=size;i++){
    *(data+i) = rdata[i];
}</pre>
```

```
return 0;
// initial_i2c_off();
```





Mulwrite KX023 (kx023.c)

void Mulwrite_kx(unsigned char addr,unsigned char *data, unsigned int size){
 unsigned int i,DataLen;

```
Start I2C
DataLen = size;
//initial_i2c_on();
                                                                               Send Address KX023
for(i=0;i<size;i++){</pre>
  sdata[i] = *(data+i);
}
                                                                                   Send Address
i2c start();
i2c_write(kx_Addr);
i2c write(addr);
                                                                                    Send data
for(i=0;i<DataLen;i++){</pre>
  i2c write(sdata[i]);
                                                                                    Send Stop
}
i2c_stop();
//initial_i2c_off();
```

}

KX023 On Off (kx023.c)

```
void Initial_kx_on(void)
{
     initial_i2c_on();
}
void Initial_kx_off(void)
```

```
{
    initial_i2c_off();
}
```

KX023 On

Pull – up

SDA Enable Output

SCL Enable Output

KX023 Off

Not Pull – up

Read Acceleration (kx023.c)

void Read_Acceleration(void){

```
Mulread kx(0,Acc Read,13);
XHP dec = (Acc Read[1]<<8) Acc Read[0];</pre>
YHP dec = (Acc Read[3]<<8) Acc Read[2];</pre>
ZHP dec = (Acc Read[5]<<8) Acc Read[4];</pre>
XOUT dec = (Acc Read[7]<<8) Acc Read[6];
YOUT dec = (Acc Read[9]<<8) Acc Read[8];
ZOUT dec = (Acc Read[11] << 8) | Acc Read[10];
XHP G = XHP dec*G ratio;
YHP G = YHP dec*G ratio:
ZHP G = ZHP dec*G ratio;
XOUT G = XOUT dec*G ratio;
YOUT G = YOUT dec*G ratio;
ZOUT_G = ZOUT_dec*G_ratio;
//convert signed2unsigned
// XOUT uint = (~((int)XOUT dec))&0x7fff;
// YOUT uint = (~((int)YOUT dec))&0x7fff;
```





Concept



XHP,YHP,ZHP < 0,1 >< decimal ><decimal>



KX023_Float_to_Char (uart0.c)

void KX023_Float_to_Char(void)

{

//*	************* Accelerometer Sensor ************
1	<pre>kx023_XHP_G_float = XOUT_G;</pre>
	<pre>if(kx023_XHP_G_float < 0)</pre>
2 3 4	<pre>kx023_XHP_G_float *= -1; kx023_XHP_G_int = kx023_XHP_G_float; kx023_XHP_G_char = kx023_XHP_G_int 0x80;</pre>
	<pre>} else { kx023_XHP_G_int = kx023_XHP_G_float; kx023_XHP_G_char = kx023_XHP_G_int & 0x7F; }</pre>
5 6 7	<pre>//kx023_XHP_G_float = kx023_XHP_G_float - kx023_XHP_G_int; //kx023_XHP_G_point = kx023_XHP_G_float*10000; kx023_XHP_G_point = (kx023_XHP_G_float - kx023_XHP_G_int)*10000; kx023_XHP_G_Point_Char_High = (kx023_XHP_G_point & 0xFF00)>>8; kx023_XHP_G_Point_Char_Low = kx023_XHP_G_point & 0x00FF;</pre>

Example

Value of KX023 (XOUT_G) = -0.1227

1. XHP_G_float = -0.1227

2. XHP_G_float = 0.1227

3. XHP_G_int = 0x00

4. XHP_G_Char = 0x80

5. XHP_G_point = (0.1227 - 0)*10000 = 1227

6. XHP_G_Point_Char_High = 0x1227 & 0xFF00

= 0x1200

= 0x1200 >> 8 = 0x0012

7. XHP_G_Point_Char_Low = 0x1227 & 0x00FF

 $= 0 \times 0027$



KX023_Float_to_Char (uart0.c)

1 kx023_YHP_G_float = YOUT_G;

Example

Value of KX023 (YOUT G) = 0.5325

1. YHP_G_float = 0.5325

2. YHP_G_int = 0

3. YHP_G_Char = 0x00

4. YHP_G_point = (0.5325 - 0)*10000 = 5325

5. YHP_G_Point_Char_High = 0x5325 & 0xFF00

= 0x5300

= 0x5300 >> 8 = 0x0053

6. YHP_G_Point_Char_Low = 0x5325 & 0x00FF

 $= 0 \times 0025$

KX023_Float_to_Char (uart0.c)

```
Value of KX023 (ZOUT G) = -0.0827
  kx023 ZHP G float = ZOUT G;
                                                                    1. ZHP G float = -0.0827
  if(kx023_ZHP_G_float < 0)</pre>
  {
                                                                    2. ZHP G float = 0.0827
2
3
      kx023_ZHP_G_float *= -1;
      kx023 ZHP G int = kx023 ZHP G float;
      kx023_ZHP_G_char = kx023_ZHP_G_int | 0x80;
                                                                    3. ZHP G int = 0x00
   }
   else
                                                                    4. ZHP G Char = 0x80
   {
       kx023_ZHP_G_int = kx023_ZHP_G_float;
      kx023 ZHP G char = kx023 ZHP G int & 0x7F;
                                                                    5. ZHP G point = (0.0827 - 0)*10000 = 827
   }
5
6
                                                                    6. ZHP G Point Char High = 0x0827 & 0xFF00
      kx023_ZHP_G_point = (kx023_ZHP_G_float - kx023_ZHP_G_int)*10000
      kx023_ZHP_G_Point_Char_High = (kx023_ZHP_G_point & 0xFF00)>>8;
7
      kx023 ZHP G Point Char Low = kx023 ZHP G point & 0x00FF;
                                                                                                = 0 \times 0800
                                                                                                = 0 \times 0800 >> 8 = 0 \times 0008
                                                                    7. ZHP G Point Char Low = 0x0827 & 0x00FF
                                                                                                = 0 \times 0027
```

Example

KX023_Float_to_Char (uart0.c)

```
void UART00_KX023(void)
{
```

```
Sensor Index = 0;
```

```
Clear_Buffer();
```

```
UART00_TX_Buf[0] = kx023_XHP_G_char;
UART00_TX_Buf[1] = kx023_XHP_G_Point_Char_High;
UART00_TX_Buf[2] = kx023_XHP_G_Point_Char_Low;
UART00_TX_Buf[3] = kx023_YHP_G_char;
UART00_TX_Buf[4] = kx023_YHP_G_Point_Char_High;
UART00_TX_Buf[5] = kx023_YHP_G_Point_Char_Low;
UART00_TX_Buf[6] = kx023_ZHP_G_char;
UART00_TX_Buf[6] = kx023_ZHP_G_Point_Char_High;
UART00_TX_Buf[7] = kx023_ZHP_G_Point_Char_Low;
TXD00_TX_Buf[8] = kx023_ZHP_G_Point_Char_Low;
TXD00_Index = 0;
Flag._TXD00 = 1;
Sensor_Index = 2;
Set_UART00_TX();
```

```
RXD00_Index = 0;
//Flag._RXD00 = 0;
```

//}

Example

 $UART00_TX_Buf[0] = 0x80$

 $UART00_TX_Buf[1] = 0x12$

 $UART00_TX_Buf[2] = 0x27$

 $UART00_TX_Buf[3] = 0x00$

 $UART00_TX_Buf[4] = 0x53$

 $UART00_TX_Buf[5] = 0x25$

 $UART00_TX_Buf[6] = 0x80$

 $UART00_TX_Buf[7] = 0x08$

 $UART00_TX_Buf[8] = 0x27$

Connection ES-ICD-V1 ,Sensor Shield and Module









Using the application

Step 1 Connect MCU Board with Bluetooth



© 2017 ROHM Semiconductor (Thailand) Co., Ltd. © 2017 LAPIS Semiconductor Co., Ltd. All rights reserved



Using the application

Step 2 Open application and click "Bluetooth Check"





Using the application

Step 3 Choose device and connect





Step 4 After connected this app Show button to choose. Choose "Accelerometer : KX023"





Using the application

Step 5 After Choose this app show Signal graph of KX023.





|--|



https://www.lapis-semi.com/cgi-bin/MyLAPIS/regi/login.cgi

pichet@es.co.th

ROHM SEMICONDUCTOR

ROHM Semiconductor (Thailand) Co., Ltd.

© 2017 ROHM Semiconductor (Thailand) Co., Ltd.