

AD5669R - Microcontroller No-OS Driver

Supported Devices

• AD5669R

Evaluation Boards

• EVAL-AD5669RSDZ

Overview

The AD5669R device is a low power, octal, 16-bit, buffered voltage-output DACs. It operates from a single 2.7 V to 5.5 V supply and is guaranteed monotonic by design.The AD5669R has an on-chip reference with an internal gain of 2. The user can select the AD5669R with a 1.25 V 5 ppm/°C reference (AD5669R-1), giving a full-scale output range of 2.5 V, or a 2.5 V 5 ppm/°C reference (AD5669R-2 and AD5669R-3), giving a full-scale output range of 5 V. The on-chip reference is off at power-up, allowing the use of an external reference. The internal reference is enabled via a software write. The part incorporates a power-on reset circuit that ensures that the DAC output powers up to 0 V and remains powered up at this level until a valid write takes place. The part contains a power-down feature that reduces the current consumption of the device to 400 nA at 5 V and provides software-selectable output loads while in power-down mode for any or all DAC channels.



The goal of this project (Microcontroller No-OS) is to be able to provide reference projects for lower end processors, which can't run Linux, or aren't running a specific operating system, to help those customers using microcontrollers with ADI parts. Here you can find a generic driver which can be used as a base for any microcontroller platform and also specific drivers for Renesas platforms.

HW Platform(s):

• Renesas Demo Kit for RL78G13 (Renesas)

Driver Description

The driver contains two parts:

- The driver for the AD5669R part, which may be used, without modifications, with any microcontroller.
- The Communication Driver, where the specific communication functions for the desired type of processor and communication protocol have to be implemented. This driver implements the communication with the device and hides the actual details of the communication protocol to the ADI driver.

The Communication Driver has a standard interface, so the AD5669R driver can be used exactly as it is provided.

There are three functions which are called by the AD5669R driver:

- I2C_Init() initializes the communication peripheral.
- I2C_Write() writes data to the device.

• I2C_Read() - reads data from the device.



I2C driver architecture

The following functions are implemented in this version of AD5669R driver:

Function	Description
unsigned char AD5669R_Init(void)	Resets the device and performs several initializations.
void AD5669R_PowerMode(unsigned char pwrMode)	Sets the device in a specific power mode.
void AD5669R_Reset(void)	Resets the device.
void AD5669R_SetInputRegister(unsigned long registerValue)	Writes a 24-bit data-word to the Input Register of the device.
unsigned short AD5669R_GetRegisterValue(void)	Reads the last register written by user.

Downloads

- AD5669R Generic Driver
- AD5669R RL78G13 Driver
- AD5669R RX62N Driver

Renesas RL78G13 Quick Start Guide

This section contains a description of the steps required to run the AD5669R demonstration project on a Renesas RL78G13 platform.

Required Hardware

- Renesas Demo Kit for RL78G13 (Renesas)
- EVAL-AD5669RSDZ(ADI)

Required Software

- IAR Embedded Workbench for Renesas RL78 Kickstart
- Applilet3 for RL78G13

Hardware Setup

An EVAL-AD5669RSDZ has to be interfaced with the Renesas Demonstration Kit (RDK) for RL78G13:

EVAL-AD5669RSDZ	J6	connector	Pin	SCL	(SCL)	→	YRDKRL78G13	J9
connector Pin 1								
EVAL-AD5669RSDZ	J7	connector	Pin	SDA	(SDA)	→	YRDKRL78G13	J9
connector Pin 3								
EVAL-AD5669RSDZ	J8	connector	Pin	\LDAC	(LDAC)	→	YRDKRL78G13	J11
connector Pin 9								
EVAL-AD5669RSDZ	J9	connector	Pin	\CLR	(CLR)	\rightarrow	YRDKRL78G13	J11
connector Pin 10								



Software Setup

With the **Applilet3 for RL78G13** tool the following peripherals have to be configured:

CSI10 (Clocked Serial Interface 10) - For the ST7579 LCD

Choose to generate the Transmit/receive function for the CSI10 and configure the interface with the following settings:

- Transfer mode setting: Single transfer mode
- Data length setting : 8 bits
- Transfer direction setting: MSB
- Specification of data timing: Type 1
- Transfer rate setting Clock mode: Internal clock (master)
- Transfer rate setting Baudrate: 1000000 (bps)

- Interrupt setting Transfer interrupt priority (INTCSI10): Low
- Uncheck the callback functions.

TM00 (Timer 00) - For the DelayMs() function

Configure TM00 as an interval timer:

- Interval timer setting Interval value(16 bits): 1 ms
- Interval timer setting Uncheck Generates INTM00 when counting is started
- Interrupt setting Uncheck End of timer channel 0 count, generate an interrupt (INTM00)

Watchdog Timer

Disable the watchdog timer:

• Choose for the Watchdog timer operation setting: **Unused** option.

Reference Project Overview

The reference project powers-up all DAC channels, displays the content of the power register and starts to generate a saw-tooth wave.



Software Project Setup

This section presents the steps for developing a software application that will run on the **Renesas Demo Kit for RL78G13** for controlling and monitoring the operation of the **ADI** part.

Two software applications have to be used: **Applilet3 for RL78G13** (a tool that automatically generates device drivers for MCU peripheral functions) and **IAR Embedded Workbench for Renesas RL78** (the integrated development environment).

Step 1 - Applilet3 for RL78G13

Run the *Applilet3 for RL78G13* tool and create a new project for *R5F100LE* processor. Select *IAR Compiler* build tool, a project name, a location for the new project and press *OK*.

S Applilet3				
File Peripheral Opt	ions Help		x	
Project tree	Kind of project: Using microcontroller: R5F101, R5F101 R5F101 R5F101 R5F101 R178/G13(F R178/G13(F R178/G13(F R178/G13(F R178/G13(F R178/G13(F R178/G13(F R178/G13(F	Project for RL78/78K0R JE(52pin) LE(64pin) ROM:96KB) ROM:128KB) ROM:192KB) ROM:256KB) ROM:256KB) ROM:256KB)	T	
	Using Build Tools: Project Name: Place:	IAR Compiler ADIEvalBoard D:\		
MCU: Chip:				

• Keep the default **Pin assignment** setting and click **Fix settings**.

Applilet3 for RL78/G13 - ADIEvalB	oard.cgp	X
File Peripheral Options Help File Peripheral Project tree Project tree Port A/D Converter Keal-time Clock Interval Timer Clock Output/Buzzer Outp DMA Controller Voltage Detector	Module File Property Image: Selection of the se	• E
	It's necessary to make a project again to change it. Fix settings	
	Pin Function P17 TI02/T002 P31 TI03/T003 P42 TI04/T004 III W0403029:Please select pin assignment setting.	-
MCU:RL78/G13 Chip:R5F100LE		.::

Now the desired peripherals can be configured and the code can be generated. For example, if the clocked serial interface 10 (*CSI10*) has to be configured, select the *Serial* peripheral, choose for the Channel 2 of Serial Array Unit 0 (*SAU0*) the *CSI10* interface, *Transmit/receive function* option and then go to *CSI10* tab.

C	
Applilet3 for RL78/G13 - ADIEvalBo	ard.cgp* Module* File Property Generate Code SAU0 SAU0 SAU0 SAU1 IICA0 Channel UART0 VART1 CSI00 CSI00 CSI01 CSI01 CSI01 Channel Unused Channel 1 Unused Channel 2 CSI0 Transmit/receive function Channel 3 Unused Channel 3 Unused T
< •	W0403029:Please select pin assignment setting.
MCU:RL78/G13 Chip:R5F100LE	

• To configure the **CSI10** interface for serial transmissions of 8 bits, with MSB first, with the data captured on clock's rising edge, with a frequency of the clock of 1 MHz and the idle state high, the settings from the following image have to be made.

S Applilet3 for RL78/G13 - ADIEval	Goard.cgp*	
File Peripheral Options Help		
Peripheral Clock Generator Port Interrupt Serial A/D Converter Timer Watchdog Timer Real-time Clock Interval Timer Clock Output/Buzzer Outp DMA Controller Voltage Detector	Module File Property Image: Second sec	 Image: Solution of the solution o
	Baudrate	1000000 • (bps) (A
<	W0403029:Please select pin assignment setting. W0403008:Information of valid input value range: 61.034~800 W0403008:Information of valid input value range: 61.034~800	0000 0000
MCU:RL78/G13 Chip:R5F100LE		

• After all the desired peripherals are configured click on the *Generate Code* button and a new workspace and a new project for the *IAR Embedded Workbench* will be generated. After the code was generated close the *Applilet3 for RL78G13* tool.

Step 2 - IAR Embedded Workbench for Renesas RL78

• Run the IAR Embedded Workbench and open the workspace created with the Applilet3 tool.

💥 IAR Embedded Workbench IDE			
File Edit View Project Tools	Wind	w Help	
0 🛩 🖬 🕼 🛋 🕯 🖻	l in	→ 	> 📣 🐘 🛤 👷 🐘 🔈
Workspace	1	x	• • • • • • • • • • • • • • • • • • •
Debug			
Files	8:: B		
🗆 🗇 ADIEvalBoard - Deb	~		
→ ⊕ 🔁 applilet_src			
		-	
		-	
		-	
		-	
		-	
		-	
		-	
		-	
		-	
Ready			14. I

• Copy the files extracted from the zip file into the **user_src** folder, located in the project's folder.



The new source files have to be included into the project. Add in the *user_src* group the files from the corresponding folder (Right click on the group and select *Add - Add Files...*). Because a new *Main* file was included the *r_main.c* file from the *applilet_src* group has to be deleted (Right click on the file and select *Remove*).

IAR Embedded Workbench IDE						
File Edit View Project Tools	Window	Help				
in 🚅 🖬 🗇 😹 i k 🖻 🖻	L ID CI		>	> > 	📣 🏟 🐘 📑	🐘 🔉 🗩 🔈
Workspace	×	1		···	,	• • • • • • • • • •
Debug	•					
Files	2:: B					
E CADIEvalBoard - Deb	· · · ·					
	-					
III H 🕀 🖸 r_cg_cgc.c						
III I -⊞ 🖻 r_cg_cgc_user.c						
r_cg_wdt.c	*					
I I III c] r_cg_wdt_user.c	*					
H I I I I I I I I I I I I I I I I I I I	Options					
	Make					
ADIDriver.c	Compile					
ADIDriver.h	Rebuild	All				
Communication.c	CL					
Communication.h	Clean					
	Stop Bu	ld				
	Add	•				
	Remove					
	Rename					
	Source	Code Control 🔹 🕨				
ADIEvalBoard	File Pro	oerties				
Remove selected item	Set as A	ctive				the second secon

• Now the debugger driver has to be selected from the project's options. Right click on the project name and select **Options**. From the **Debugger** category choose the **TK** Debugger Driver.

File Edit View	Vorkbench IDE Project Tools Windo	w Help	
Workspace Debug Files Fi	Category: General Options C/C++ compiler Assembler Custom Build Build Actions Linker Debugger E1 E20 IECUBE Simulator TK	Setup Extra Options Images Plugins Driver: TK • TK • • Ø Run to: • • main Setup macros • • Device description file • • •	× x
ADIE valBoard		STOOLKIT_DIR\$\CONFIG\DDF\vor5f100le.ddf	

 Now, the project is ready to be compiled and downloaded on the board. Press the F7 key to compile it. Press CTRL + D to download and debug the project.

💥 IAR Embedded Workbench IDE	
File Edit View Project Debug Emulator Tools Window Help	
D 🖆 日 🗿 長 浩 階 信 い い	👍 🐚 👯 🕅 🕭 🚸 🚸
5 • 32 8 3 3 3 X	
🔲 🕅 🖓 避 🤌 🗱 🎻 📕 🥜	
Workspace × Main.c × ×	Disassembly ×
Debug Files ADIEvalBo ADIEvalBo ADIEvalBo ADIEvalBoard	Go to <u>return(1):</u> ADI_Init: 00128 E1 00129 D7 ADI Init(); main: 0012A FD280 <u>while (1)</u> 0012D EFFE <u>CMC = 00 CGC H</u> R_CGC_Create: 0012F CEA00 <u>MSTOP = 1U;</u> 00132 717AA MCM0 = 0U: *
Messages Building configuration: ADIEvalBoard - Debug Lindating build tree	File Li
Configuration is up-to-date.	
Debug Log Build	×
Ready Errors 0, Warnings 0	Ln 80, Col 5

29 Feb 2012 16:01 · Dragos Bogdan

Renesas RX62N Quick Start Guide

This section contains a description of the steps required to run the AD5669R demonstration project on a Renesas RX62N platform.

Required Hardware

- Renesas Demo Kit for RX62N (Renesas)
- EVAL-AD5669REBZ (ADI)

Required Software

• High-performance Embedded Workshop for RX62N family

Hardware Setup

An EVAL-AD5669REBZ board has to be interfaced with the Renesas Demonstration Kit (RDK) for RX62N:

EVAL-AD5669RSDZ]6	connector	Pin	SCL	(SCL)	→	YRDKRX62N	J2	connector
Pin 1									
EVAL-AD5669RSDZ]7	connector	Pin	SDA	(SDA)	→	YRDKRX62N	J2	connector
Pin 3									
EVAL-AD5669RSDZ	J8	connector	Pin	\LDAC	(LDAC)	→	YRDKRX62N	J8	connector
Pin 17									
EVAL-AD5669RSDZ]9	connector	Pin	\CLR	(CLR)	→	YRDKRX62N	J8	connector
Pin 25									



Reference Project Overview

The reference project: The reference project powers-up all DAC channels, displays the content of the power register and starts to generate a saw-tooth wave.



Software Project Setup

This section presents the steps for developing a software application that will run on the **Renesas Demo Kit for RX62N** for controlling and monitoring the operation of the ADI part.

- Run the High-performance Embedded Workshop integrated development environment.
- A window will appear asking to create or open project workspace. Choose "Create a new project workspace" option and press *OK*.
- From "Project Types" option select "Application", name the Workspace and the Project " ADIEvalBoard", select the "RX" CPU family and "Renesas RX Standard" tool chain. Press OK.

New Project Workspace		? X
Projects Project Types Project Typ	Workspace Name: ADIEvalBoard Project Name: ADIEvalBoard Directory: C:\WorkSpace\ADIEvalBoard CPU family: RX Tool chain: Renesas RX Standard	Browse
< Ⅲ ► Properties		
	ОК	Cancel

- A few windows will appear asking to configure the project:
 - In the "Select Target CPU" window, select "RX600" CPU series, "RX62N" CPU Type and press Next.
 - In the "Option Setting" windows keep default settings and press Next.
 - In the "Setting the Content of Files to be generated" window select "None" for the "Generate main() Function" option and press Next.
 - In the "Setting the Standard Library" window press "Disable all" and then Next.
 - In the "Setting the Stack Area" window check the "Use User Stack" option and press Next.
 - In the "Setting the Vector" window keep default settings and press Next.
 - In the "Setting the Target System for Debugging" window choose "RX600 Segger J-Link" target and press Next.
 - In the "Setting the Debugger Options" and "Changing the Files Name to be created" windows keep default settings, press Next and Finish.
- The workspace is created.

ADIEvalBoard - High-performance Embedded Workshop				
File Edit View Project Build Debug Setup Tools Test Window Help				
📗 🗅 😅 🖬 🝘 🛛 🖧 🐘 💼 🕑 📗 🎇 💭 🔽 🖬 🛣 👗 Debug	, 🔽			
ADIEvalBoard ADIEvalBoard C source file d dbsct.c d fresetprg.c sbrk.c vectbl.c Dependencies sbrk.h stacksct.h bypedefine.h vect.h 				
💾 OJ OT AJ AT 😫 ST 🖉 🖻 🖬 🤶				
	*			
< III				
Build ∧ Debug ∧ Find in Files 1 ∧ Find in Files 2 ∧ Macro ∧ Test ∧ Version Control /				
Ready III III III III Default1 desktop				

- The RPDL (Renesas Peripheral Driver Library) has to integrated in the project. Unzip the RPDL files (double-click on the file "*RPDL_RX62N.exe*"). Navigate to where the RPDL files were unpacked and double-click on the "*Copy_RPDL_RX62N.bat*" to start the copy process. Choose the LQFP package, type the full path where the project was created and after the files were copied, press any key to close the window.
- The new source files have to be included in the project. Use the key sequence *Alt*, *P*, *A* to open the " *Add files to project 'ADIEvalBoard'*" window. Double click on the RPDL folder. From the "*Files of type*" drop-down list, select "*C source file* (*.*C*)". Select all of the files and press *Add*.

Look in: 🚺 RPDL	▾ 🗢 🗈 💣 💌		
Name	Date modified	Туре	Si 🔺
Interrupt_ADC_10.c	4/8/2011 11:01 PM	C File	
Interrupt_ADC_12.c	4/8/2011 11:01 PM	C File	
Interrupt_BSC.c	4/8/2011 11:01 PM	C File	
Interrupt_CMT.c	4/8/2011 11:01 PM	C File	E
Interrupt_DMAC.c	4/8/2011 11:01 PM	C File	
Interrupt_EXDMAC.c	4/8/2011 11:01 PM	C File	
Interrupt_IIC.c	4/8/2011 11:01 PM	C File	
Interrupt_INTC.c	4/8/2011 11:01 PM	C File	
Interrupt_MTU2.c	4/8/2011 11:01 PM	C File	
Interrupt_not_RPDL.c	4/8/2011 11:01 PM	C File	
Interrupt_POE.c	4/8/2011 11:01 PM	C File	_
™t + bTC	1/0/0011-11-01-DN4	0.51	Þ
File name: "Interrupt_ADC_10.c" "Interrupt	_ADC_12.c" "Interrupt_BSC.c" "I	nterrupt_CI	Add
Files of type: C source file (* C)		T	Cancel

• To avoid conflicts with standard project files remove the files "*intprg.c*" and "*vecttbl.c*" which are included in the project. Use the key sequence *Alt, P, R* to open the "*Remove Project Files*" window. Select the files, click on Remove and press *OK*.

Remove Project Files		? <mark>×</mark>
Project files: Interrupt_ADC_12.c Interrupt_BSC.c Interrupt_OMAC.c Interrupt_IDMAC.c Interrupt_MTU2.c Interrupt_NTC.c Interrupt_POE.c Interrupt_SCI.c Interrupt_SPI.c Interrupt_WDT.c interrupt_WDT.c interrupt_C sbik.c vectbl.c	[C:\WorkSpace' ▲ [C:\WorkSpace']	OK Cancel <u>Remove</u> Remove <u>A</u> II
<	4	

• Next the new directory has to be included in the project. Use the key sequence Alt, B, R to open the "RX Standard Toolchain" window. Select the C/C++ tab, select "Show entries for: Include file directories" and press Add. Select "Relative to: Project directory", type "RPDL" as sub-directory and press OK.

RX Standard Toolchain		? X
Configuration : Debug All Loaded Projects ADLE valBoard C source file C++ source file	C/C++ Assembly Link/Library Standard Librar Category : Source Show entries for : Include file directories Ctory C	y RTOS ▲ ► Add Insert Remove Move up ove down
4	Options C/C++ : -cpu=rx600 -output=obj="\$(CONFIGDIR)\ \$(FILELEAF).obj" -debug -nologo OK	Cancel

• The library file path has to be added in the project. Select the Link/Library tab, select "Show entries for: Library files" and press Add. Select "Relative to: Project directory", type "RPDL\RX62N_library" as file path and press OK.

Configuration : Debug C/C++ Assembly Link/Library Standard Library RTOS Category : Input C	Standard Toolcham		
Debug Category: All Loaded Projects Show entries for: End C source file	Configuration :	C/C++ Assembly Link/Library Standard Library	y RTOS 🔸 🕨
Options Link/Library :	Debug All Loaded Projects ADIE valBoard ADIE valBoard C source file C++ source file Add library file Relative to : Project directory File path : RPDL\RX62N_librar	Category : Input	Add Insert Remove P Down trol:
Inoprelink rom=D=R,D_I=R_1,D_2=R_2 nomessage - list="\$(CONFIGDIR)\\$(PROJECTNAME).map" -nooptimize -	< Þ	Options Link/Library : noprelink -rom=D=R,D_1=R_1,D_2=R_2 -nomes list="\$(CONFIGDIR)\\$(PROJECTNAME).map" -no	sage -

• Because the "*intprg.c*" file was removed the "*PIntPrg*" specified in option "*start*" has to be removed. Change "*Category*" to "*Section*". Press "*Edit*", select "*PIntPRG*" and press "*Remove*". From this window the address of each section can be also modified. After all the changes are made press OK two times.

RX Standard Toolcha	in					<u> ?</u> X
Configuration :		C/C++	Assembly	Link/Libra	Standard Lib	rary RTOS
Debug	Section Address Se 0x00001000 B_	ction	-		8 ×	Add
	8_ 8_ 8 8 8	1 2 2			Add	<u>M</u> odify <u>R</u> emove
	0xFFFF8000 PR 0xFFFF8100 C_ C_ C_	esetP 1 2		E	New Overlay Remove	Import Export
	C\$ D* P PIr W* 0xFFFFFD0 FI>	* tPRG *		-	Up Dowr Import Export	
	•	-	-	-		nooptimize
					ОК	Cancel

• At this point the files extracted from the zip file located in the "*Software Tools*" section have to be added into the project. Copy all the files from the archive into the project folder.



• Now, the files have to be included in the project. Use the key sequence *Alt*, *P*, *A* to open the "*Add files to project 'ADIEvalBoard'*" window. Navigate into ADI folder. From the "*Files of type*" drop-down list, select "*Project Files*". Select all the copied files and press *Add*.

Add files to project 'ADIEvalBoard'			×		
Look in: 🚺 ADIEvalBoard 💌	← 🗈 💣 📰 -				
Name	Date modified	Туре	Si 🔺		
🜗 Debug	4/24/2012 10:00 AM	File folder			
Debug_RX600_Segger_J-Link	4/24/2012 9:53 AM	File folder			
🔑 Release	4/24/2012 9:53 AM	File folder	E		
🔋 🖟 RPDL	4/24/2012 9:36 AM	File folder			
ADIDriver.c	4/23/2012 5:19 PM	C File			
ADIDriver.h	4/23/2012 5:19 PM	H File			
Communication.c	4/23/2012 5:19 PM	C File			
Communication.h	4/23/2012 5:19 PM	H File			
dbsct.c	4/24/2012 9:15 AM	C File			
intprg.c	4/24/2012 9:15 AM	C File			
iodefine.h	4/24/2012 9:15 AM	H File	-		
File name: "YRDKRX62N.h" "ADIDriver.c" "ADIDriver.h" "Communication.c" "Commun					
Files of type: Project Files Cancel					
✓ <u>R</u> elative Path					

• Now, the project is ready to be built. Press F7. The message after the Build Process is finished has to

be "0 Errors, 0 Warnings". To run the program on the board, you have to download the firmware into the microprocessor's memory.

03 Feb 2012 14:32 · Dragos Bogdan

More information

ask questions about the Microcontroller no-OS Drivers

• Example questions:

- how to configure AD9361 for to obtain the carrier frequency in the transmission channel? by vv_rustamov
- 9361 ADC Capture Function Data Format by kruse
- AD9361 Artix 7 API drivers by Raveendra
- AD9361 Artix-7 drivers by Raveendra
- AD9361 no-OS manual gain control by pharbour

01 Jun 2012 11:21

 $\ensuremath{\mathbb{S}}$ Analog Devices, Inc. All rights reserved. Trademarks and registered trademarks are the property of their respective owners.



www.analog.com