RENESAS APPLICATION NOTE

RL78 Family

RL78 8-bit DAC Dithering

R01AN3795EU0100
Rev.1.00
Aug 10, 2017

Introduction

Some RL78 MCUs have a basic (unbuffered) 8-bit resolution Digital-to-Analog Convertor (DAC). This app note describes a method to increase the 8-bit DAC resolution to 12-bit effective resolution using a DAC amplitude dithering method.

Target Device

RL78/G14 MCUs having 8-bit DAC and fSUB = 32768HZ oscillator. **Note:** Other *RL78 MCUs having 8-bit DAC, 10-bit ADC, and DTC (Data Transfer Control), such as RL78/G11, G1F, F14, and F15 may also be used, but should be tested thoroughly to assure compatibility.*

Contents

1. Overview

Typically MCUs having 10-bit or 12-bit DAC function are more expensive that those with 8-bit DAC. Therefore, it may be desirable to expand the 8-bit DAC resolution to achieve more than 256 possible DC voltage output levels. For example, in LED lighting or analog control systems that need a logarithmic type transfer function, a total of 256 linear levels may not be adequate, since the control needs to have finer resolution steps at lower drive levels. By modulating or dithering the DAC output between 2 consecutive 8-bit code output levels and smoothing the output by RC low-pass filter, a variable duty cycle can achieve more than 256 DC output levels from an 8-bit DAC.

This application note provides a framework to implement the DAC dithering method to increase the resolution of effective DAC output levels. It is up to the user to perform a full evaluation to measure DC accuracy, and response time versus DAC dithering update frequency and RC low-pass filter characteristics. The DC output accuracy and performance depends on the user's Hardware implementation, including measures to minimize residual noise level in their board design.

2. DAC Dithering method

The dithering method is accomplished by modulating the DAC output between 2 adjacent hex values in a periodic manner, automatically, by using DTC and ELC functions. For example, the simplest DAC dithering method could be toggling between a DAC output code of 0x80 (128decimal) and 0x81 (129 decimal) with a 50%/50% ratio. This waveform would look like a square ware with a peak-to-peak value of 1 LSB out of 256 full-scale values available with 8-bit DAC output. If this square wave is filtered by a sufficiently designed low pass filter, an equivalent DC value of 128.5 out of full-scale 255 amplitude levels above 0Volts could be obtained, simulating approximately 9 bits of resolution. The low pass filter is used to reduce the dithering pattern ripple to less than 1 LSB at the new effective resolution. **Note:** *Figure 1 example shows a 200KΩ series resistor and 2.2nF filter capacitor for reducing ripple in a repeating 320uSEC DAC dither pattern, but RC filter values could be modified as required if using different DAC dithering timing.*

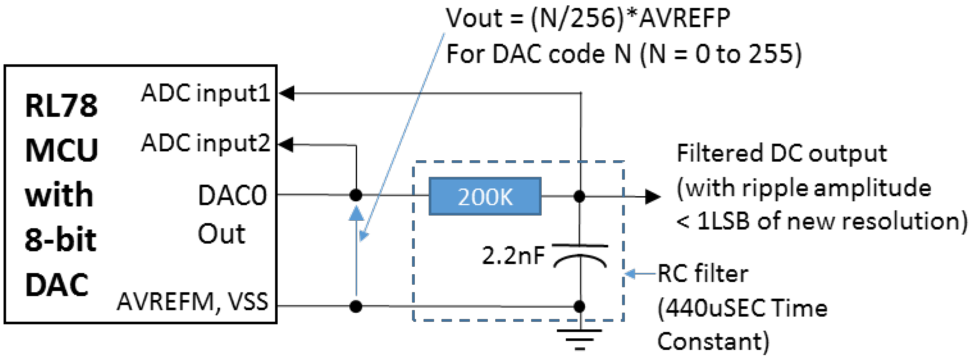
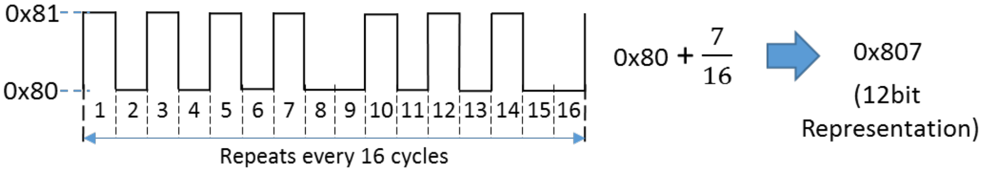


Figure 1: Low Pass RC filtering of dither DAC Output

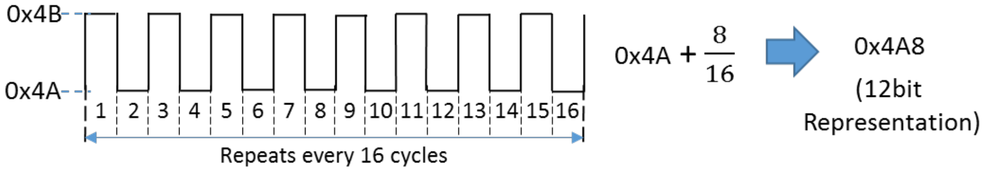
2.1 8-bit Dithering over 16 DAC output cycles

By modulating or dithering the DAC output values over a longer repeat period, with different dither patterns, more equivalent resolution bits can be obtained. This App note shows a method to create 4 additional bits of resolution, by outputting DAC output patterns that repeat every 16 cycles. With the 16 cycle pattern smoothed out by an RC low pass filter, it is capable of producing one of 15 intermediate DC voltage values between two consecutive 8-bit DAC values. Thus, the effective DC output resolution can be approximately 12 bits, with some limitations (described later).

(a) **Example 1: DAC output dithered between codes 0x80 and 0x81 (pattern for 7/16)**



(b) **Example 2: DAC output dithered between codes 0x4A and 0x4B (pattern for 8/16)**



The Data Transfer Control (DTC) and Event Link Controller (ELC) functions are used to automatically update the DAC output for 16 states of Dithering pattern, without CPU intervention needed until the last of 16 cycles, when the DTC is re-set for next 16-state operation. The RC filter with 440uSEC Time constant shown in Figure 1 is used to implement the DAC dithering Application Note Software project. With a DAC update cycle time of 20uSEC, and the dither pattern repeating every 320uSEC, a single stage RC filter with time constant of 440uSEC can reduce the ripple to less than 1LSB amplitude at the new, higher effective resolution, targeted at 12 bits. As shown in the **Figure 1** block diagram, the ADC input1 can be used to monitor the filtered DAC dithering, and apply a correction (closed-loop method) for better DC output accuracy, within the absolute accuracy of the ADC conversion measurements. By using the ADC Input2, the DAC output absolute errors before dithering can also be checked relative to the better 10-bit ADC accuracy. (see section 7.4)

2.2 DAC dithering control options:

(a) **Using open-loop DAC dithering WITHOUT ADC monitoring:**

This method will create monotonic DC steps, with almost 12-bit resolution (limitations discussed later). However, the RL78 8-bit DAC has an overall error including DAC AINL (Analog Integral Non-Linearity) of +/-2.5 LSB out of 256 output values, which is close to +/-1% error relative to the full-scale analog range. Therefore the absolute value of the 12-bit effective (dithered DAC) uncompensated DC output may also have an error up to +/-2.5LSB relative to 8-bit resolution, and may not be linear over the full range of possible values. The open-loop method may be more useful when an absolute DC voltage level is not required, but DC level feedback is done by a human interacting with potentiometer adjustment or up-down pushbutton control. ***Tradeoffs:*** Using the open-loop DAC dither mode will settle to the target DC value more quickly (usually within 16mSEC) but will have larger absolute offset error compared to using the closed-loop method with ADC monitoring and feedback.

(b) **Using closed-loop DAC dithering WITH ADC monitoring and feedback:**

The RL78 10-bit ADC maximum overall error is +/-3.5LSB (when using AVREFP and AVREFM full-scale analog voltage reference), which is slightly better than +/-1LSB error at the DAC 8-bit resolution. Therefore, it is possible to somewhat correct for both the intrinsic DAC +/-2.5 LSB max error and error from the DAC impedance loading. By calibrating the individual RL78 MCU ADCs on each board in the factory, even better accuracy might be obtained for the ADC conversion results, and therefore to the filtered DAC dithered DC output.

By summing and averaging 16 consecutive 10-bit ADC conversions, it is possible to measure the 12 bits effective resolution of the filtered DAC dithering DC output. The ADC measurement result is dependent on the ripple amplitude of the filtered DAC dithering output, and MCU/System noise riding on the PC board power, grounds and Analog reference voltage, which may not be truly random. Therefore, it’s inevitable that each averaged/scaled sum of 16 consecutive 10-bit ADC conversion of filtered DAC dithered values will still vary by 1-2 LSB (or possibly more) out of 12-bit effective value. So there will be some random amplitude jitter on the corrected DAC dithered 12-bit effective resolution when using ADC averaged measurements and feedback.

However, even with the amplitude jitter, the absolute DC output for 12-bit effective resolution can still correct for DAC output overall error, to the accuracy of the RL78’s 10-bit ADC which is almost 3 times more accurate than the DAC. The app note Software project takes 256 ADC samples (16384HZ sample rate) and updates the Dither factor by +1, or -1 (for correction) or zero (if no offset is detected), every 256 ADC samples which is at a 64HZ (15.625mSEC) update rate. This simple correction method is often referred to as “bang-bang” feedback.

Tradeoffs: the closed-loop DAC Dithering method provides more DC accuracy, but takes longer (typically 32mSEC to 128mSEC) to settle to the target 12-bit effective DC value.

3. RL78 MCU resources used

3.1 MCU Hardware

(1) 8-bit DAC

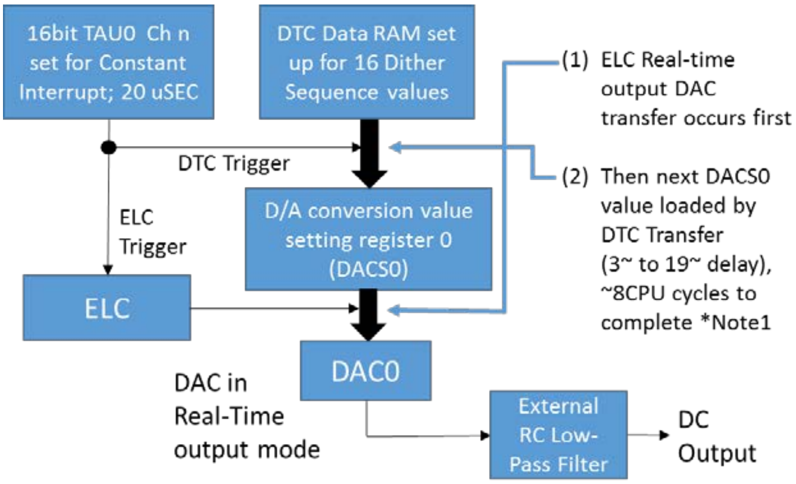
(2) DTC (Data Transfer Control) operation to automatically load sequential 16 state dither pattern into DAC buffer DACS0. Takes about 8 CPU/System clock cycles per DTC transfer, with no CPU operation needed until after 16 DTC cycles to reset for next 16 DTC transfers.

(3) ELC used to trigger DAC buffer contents to DAC output at a constant rate (20uSEC updates)

(4) 16bit High speed timer TAU0 to create 20uSEC update rate, simultaneously triggering the next DAC buffer loading action, and the constant update to actual DAC output by ELC.

(5) 10bit ADC conversions at 16384HZ rate (every 61 uSEC), using fSUB = 32768HZ clock with ADC Hardware Trigger wait mode from 12-bit Interval Timer

(6) Optional UART1 output to capture live 12-bit averages of DAC dithering.


16bit TAU0 Ch n set for Constant Interrupt; 20 uSEC
DTC Data RAM set up for 16 Dither Sequence values
(1) ELC Real-time output DAC transfer occurs first
DTC Trigger
(2) Then next DACS0 value loaded by DTC Transfer (3~ to 19~ delay), ~8CPU cycles to complete *Note1
ELC Trigger
D/A conversion value setting register 0 (DACS0)
ELC
DAC0
DAC in Real-Time output mode
External RC Low-Pass Filter
DC Output

(*Note1: After 16 DTC data transfers (320uSEC rate, DTC reset to repeat in TAU0 timer ISR)

Figure 2: Automatic DAC Dithering using TAU0 timer, DTC, ELC and 8-bit DAC

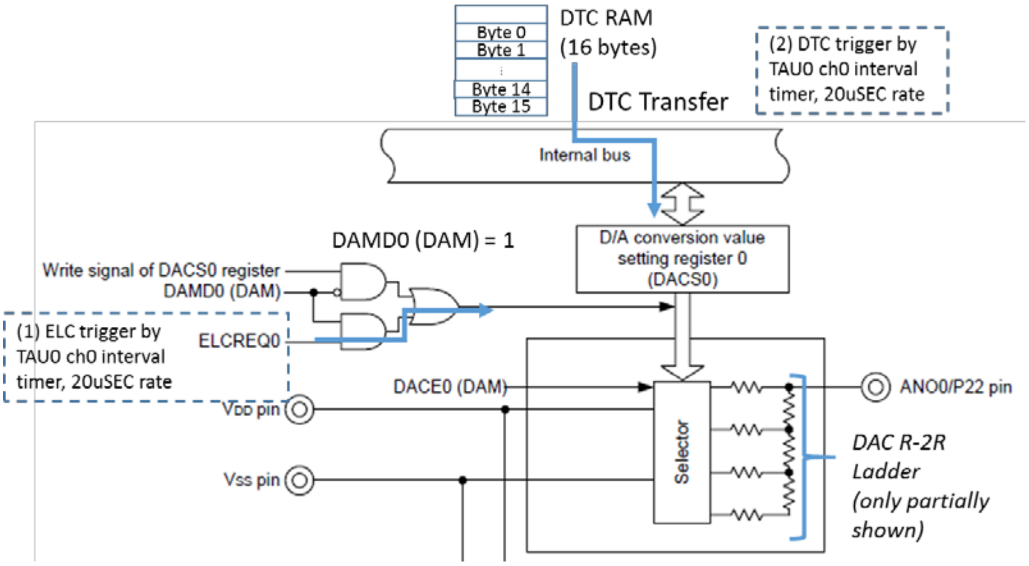

Byte 0
Byte 1
Byte 14
Byte 15
DTC RAM (16 bytes)
DTC Transfer
(2) DTC trigger by TAU0 ch0 interval timer, 20uSEC rate
Internal bus
DAMD0 (DAM) = 1
D/A conversion value setting register 0 (DACS0)
Write signal of DACS0 register
DAMD0 (DAM)
(1) ELC trigger by TAU0 ch0 interval timer, 20uSEC rate
ELCREQ0
DACE0 (DAM)
VDD pin
Vss pin
Selector
ANO0/P22 pin
DAC R-2R Ladder (only partially shown)

Figure 3: DTC data transfers with ELC trigger to DAC Real-time output

3.2 MCU CPU processing

Since the DAC output is being dithered, some CPU processing overhead is required:

(1) DTC updates every 20uSEC; DTC uses 8 CPU cycles out of every 640 cycles when using 32MHZ CPU clock rate. The update rate could be adjusted for faster or slower updates if desired, as long as the RC low-pass filter time constant is adjusted accordingly.

(2) DTC state reload (TAU0 channel0) interrupt every 16 DTC cycles, at 320uSEC rate.

(3) 10-bit ADC conversion completion interrupt processing every 61 uSEC (16384HZ rate). 10-bit ADC conversions occur at 16384HZ rate using fSUB = 32768HZ clock in ADC Hardware trigger wait mode. However, the ADC HW triggered conversions themselves can operate in parallel with Software operation with no CPU intervention until ADC completion interrupt occurs. ADC conversions are an optional method to monitor RC low-pass filter output from DAC dithering pattern, and correct the DC value by modifying the dither pattern.

(4) CPU processing the sum of 16 consecutive ADC conversion every 967.7uSEC (1024HZ rate)

(5) Optional UART1 interrupts at 86.81uSEC per ASCII Character at 115.2Kbaud. This facilitates real-time monitoring of 16 consecutive averaged 10-bit ADC conversions of DAC dithering output, in a testing mode.

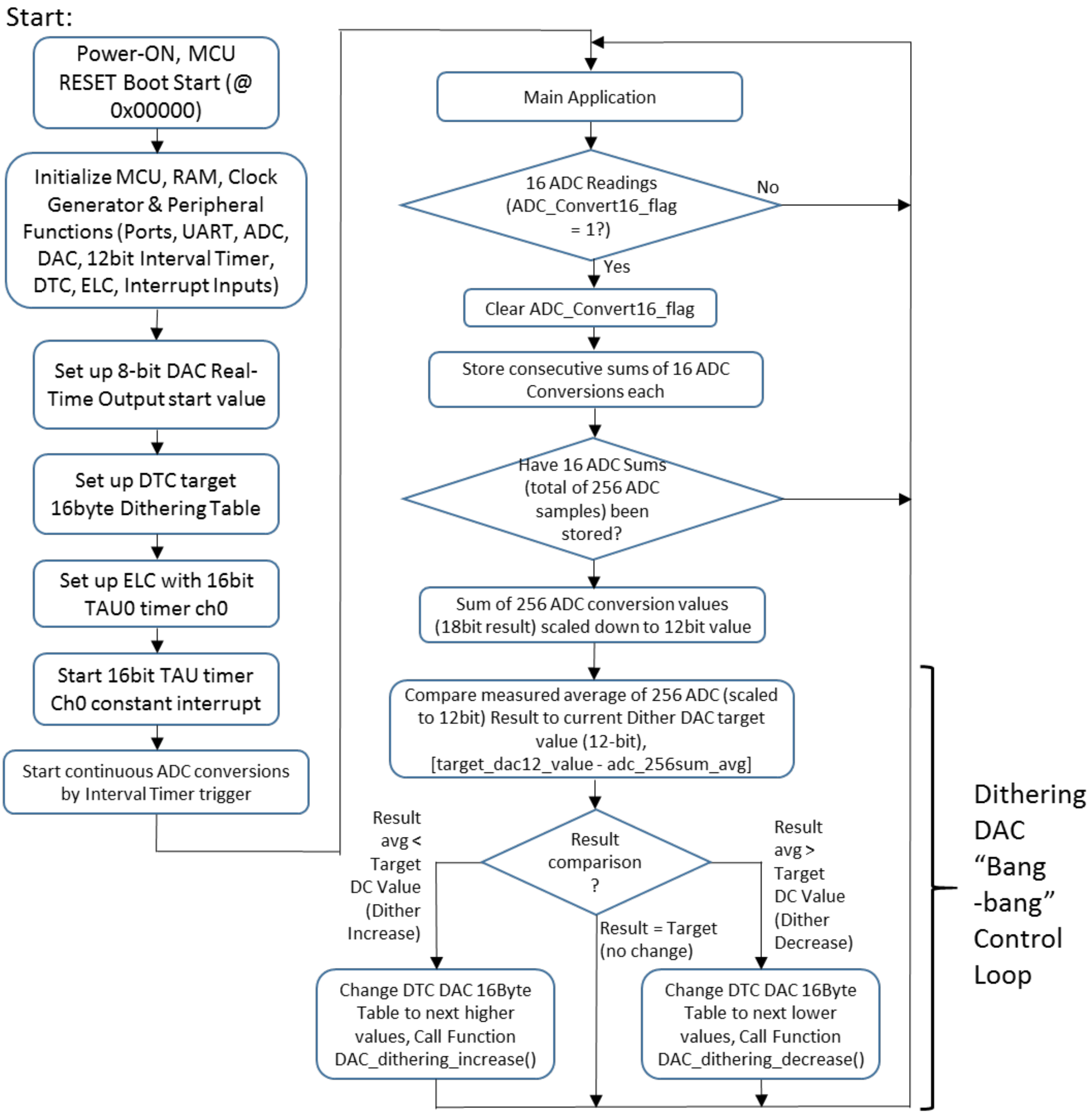
4. Software Environment

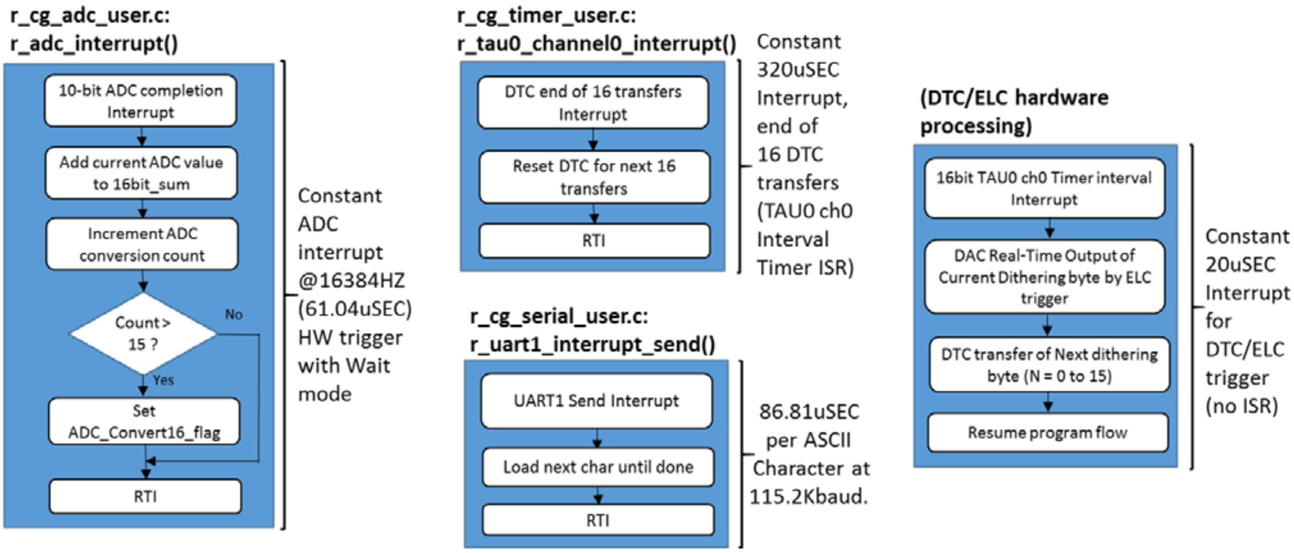
The sample Software project accompanying this application note uses the SW environment as follows:

IDE	RL78 Compiler	Compiler Version
Renesas e2studio v 5.2.1.010	CCRL	v1.04

Table 1: Sample Software project environment

5. DAC Dithering (High-level Flowchart and Interrupt processing)


Start:
Power-ON, MCU RESET Boot Start (@ 0x00000)
Initialize MCU, RAM, Clock Generator & Peripheral Functions (Ports, UART, ADC, DAC, 12bit Interval Timer, DTC, ELC, Interrupt Inputs)
Set up 8-bit DAC Real-Time Output start value
Set up DTC target 16byte Dithering Table
Set up ELC with 16bit TAU0 timer ch0
Start 16bit TAU timer Ch0 constant interrupt
Start continuous ADC conversions by Interval Timer trigger
Main Application
16 ADC Readings (ADC_Convert16_flag = 1?)
No
Yes
Clear ADC_Convert16_flag
Store consecutive sums of 16 ADC Conversions each
Have 16 ADC Sums (total of 256 ADC samples) been stored?
Sum of 256 ADC conversion values (18bit result) scaled down to 12bit value
Compare measured average of 256 ADC (scaled to 12bit) Result to current Dither DAC target value (12-bit), [target_dac12_value - adc_256sum_avg]
Result comparison ?
Result avg < Target DC Value (Dither Increase)
Result = Target (no change)
Result avg > Target DC Value (Dither Decrease)
Change DTC DAC 16Byte Table to next higher values, Call Function DAC_dithering_increase()
Change DTC DAC 16Byte Table to next lower values, Call Function DAC_dithering_decrease()
Dithering DAC “Bang -bang” Control Loop


r_cg_adc_user.c:
r_adc_interrupt()
10-bit ADC completion Interrupt
Add current ADC value to 16bit_sum
Increment ADC conversion count
Count > 15 ?
No
Yes
Set ADC_Convert16_flag
RTI
Constant ADC interrupt @16384HZ (61.04uSEC) HW trigger with Wait mode
r_cg_timer_user.c:
r_tau0_channel0_interrupt()
DTC end of 16 transfers Interrupt
Reset DTC for next 16 transfers
RTI
Constant 320uSEC Interrupt, end of 16 DTC transfers (TAU0 ch0 Interval Timer ISR)
r_cg_serial_user.c:
r_uart1_interrupt_send()
UART1 Send Interrupt
Load next char until done
RTI
86.81uSEC per ASCII Character at 115.2Kbaud.
(DTC/ELC hardware processing)
16bit TAU0 ch0 Timer interval Interrupt
DAC Real-Time Output of Current Dithering byte by ELC trigger
DTC transfer of Next dithering byte (N = 0 to 15)
Resume program flow
Constant 20uSEC Interrupt for DTC/ELC trigger (no ISR)

6. DAC dithering method details

6.1 DAC Dither Adder Sequence table

This DAC dither adder sequence table was designed to distribute the zeroes and ones pattern for highest repetition frequency, and thus minimize the ripple amplitude when filtered by a simple low pass RC filter. A zero means the base DAC value is used and a one means the base DAC + 1 value is used. Therefore, the Dither factor is the total duty cycle of N/16, where N = 0 to 15.

	Dither sequence (Dither factor)															
DTC State	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
3	0	0	0	0	0	0	1	1	1	0	0	1	1	1	1	1
4	0	0	0	0	0	1	0	0	0	1	1	0	1	1	1	1
5	0	0	0	0	1	0	0	1	1	0	1	1	0	1	1	1
6	0	0	0	1	0	0	1	0	0	1	0	1	1	0	1	1
7	0	0	0	0	0	1	0	1	1	0	1	0	1	1	1	1
8	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
9	0	0	1	0	1	0	1	0	1	1	0	1	0	1	0	1
10	0	0	0	0	0	1	0	1	0	0	1	0	1	1	1	1
11	0	0	0	1	0	0	1	0	1	1	0	1	1	0	1	1
12	0	0	0	0	0	0	0	1	0	0	1	1	1	1	1	1
13	0	0	0	0	1	1	0	0	1	1	1	0	0	1	1	1
14	0	0	0	0	0	0	1	1	0	0	0	1	1	1	1	1
15	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
16	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
DTC Dither sequence	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

🟨 Base DAC value

🟩 Base DAC value + 1

Table 2: Dithering adder sequence table

Each vertical set of 16 adder values is for creating a new DTC sequence by adding to the base DAC value.

Examples:

- DTC Adder Sequence 0 has all zeroes (no dithering)
- DTC Adder Sequence 7 has 9 zeroes and 7 ones (7/16 dithering)
- DTC Adder Sequence 15 has one zero and 15 ones (15/16 dithering)

6.2 DTC Dithering adder example

Example to create new DTC Dithering values:

If base DAC value is 0x80, and adder is for 7/16 dithering (DTC Adder Sequence 7). Effective 12-bit target value = 0x807

DTC state	DAC Base Value		Dithering Adder Values		New DTC Sequence
1	0x80	+	1	=	0x81
2	0x80	+	0	=	0x80
3	0x80	+	1	=	0x81
4	0x80	+	0	=	0x80
5	0x80	+	1	=	0x81
6	0x80	+	0	=	0x80
7	0x80	+	1	=	0x81
8	0x80	+	0	=	0x80
9	0x80	+	0	=	0x80
10	0x80	+	1	=	0x81
11	0x80	+	0	=	0x80
12	0x80	+	1	=	0x81
13	0x80	+	0	=	0x80
14	0x80	+	1	=	0x81
15	0x80	+	0	=	0x80
16	0x80	+	0	=	0x80

Table 3: Dithering adder example for target 12bit value 0x807

6.3 Tradeoffs for DAC dither update rate versus CPU overhead

When implementing the DAC dithering method shown in this app note, it may be desirable to change the dithering rate, which may also require modifying the low-pass filter RC time constant. The tradeoff is that a slow dithering update rate, and longer RC filter time constant, will have a longer response time when changing the target DAC dithered 12-bit effective resolution value, especially when the target DC output value delta change is large. For example, changing from 10% to 90% of the DAC analog full scale reference (AVREFP-AVREFM) will take settling time that is proportional to the RC low pass filter time constant (approximately 2.5 to 3 RC time constants). However, a faster DAC dither rate will, on average, require more CPU/System clock cycles for DTC operation, ADC conversion complete interrupts and CPU processing overhead. Therefore, the user will need to determine a balance between optimum DC Slew rate performance and System/CPU processing time overhead.

6.4 Measuring DAC dithering ripple at RC low-pass filter output

The app note software project implements a 20uSEC DAC dither update rate that repeats every 16 periods (320uSEC total period) to create close to 12-bit effective resolution. The RC low pass filter should be designed to limit ripple to less than 1LSB at 12-bit effective resolution, which means less than 1/16 of the DAC dithering amplitude (DAC dithering is done at AVREFP/256 level, with 1LSB peak-to-peak amplitude), and probably closer to ½ LSB at 12-bit effective resolution (1/32 of AVREP/256, 1LSB amplitude). This app note sample project uses 200K series resistor and 2.2nF capacitor for a 440uSEC RC time constant. Using the 440uSEC RC time constant achieves about 0.69 LSB typical ripple (relative to 12-bit effective resolution) when the dither sequence is 7/16, which is worst case. The best case is about 0.36 LSB typical ripple (relative to 12-bit effective resolution) when the dither sequence is 8/16.

It is difficult to measure the dithering ripple amplitude after RC filtering for these reasons:

(1) The ripple amplitude for 12bit effective DAC resolution is typically less than 0.75mV-1.25mV depending on VDD value. In Table 2 example below, VDD values of 3.072Volts and 5.12 Volts (similar to typical 3Volt and 5Volt power supplies) are shown to make the steps easy to represent. At these 0.75-1.25millivolts levels (or less) the System/MCU noise levels may be as high the DAC dithering ripple value.

	DAC/ ADC Reso- lution	Each LSB step, when VDD =	
Using RL78:		**3.072 Volts**	**5.12 Volts**
DAC	8bit	12 mV	20 mV
ADC	10bit	3 mV	5 mV
DAC dithering	12bit	0.75 mV	1.25 mV

Table 4: DAC/ADC resolutions

(2) Additionally, the high output impedance of the 8-bit DAC (5K-39KΩ) and example 200KΩ series resistance of RC low pass filter would seem to require using a standard X10 (10MΩ input impedance) oscilloscope probe, since a 1MΩ probe would load down a 200K source impedance significantly. However, the actual ripple amplitude into the scope input after a X10 probe 10:1 voltage reduction will typically be too low (possibly less than 100microvolts pk-pk), and unreadable on most standard oscilloscopes.

(3) As an alternative, if a 1MΩ scope probe is used, then the probe impedance significantly loads down the RC low-pass filter output amplitude, almost 17% when using series R of 200K. Also, since the RC filtering reduces the DAC dithered ripple amplitude to less than 1LSB at 12bits effective resolution, the digital sampling of the storage scope will have problems capturing the ripple waveform accurately, since most digital scopes typically sample with 10-bit resolution for analog signals.

The easiest way to measure relative DAC dither ripple amplitude at the RC low-pass filter output is to use maximum DAC output amplitude of 0x00 to 0xFF peak-peak pulsing and measure with a X10 (10MΩ input impedance) scope probe. The ripple amplitude reduction of the RC filter output fed with full 0volt to (255/256)*AVREFP pk-pk voltage swings will be the same as 1LSB (relative to 8-bit resolution) pulse waveform attenuation on the RC filter output. Then, the filtered DAC dithering waveform is larger by a factor of 255 and the resulting filtered ripple amplitude is approximately 60mV-100mV pk-pk (or greater, depending on AVREP, VDD) and much easier to measure. A dither pattern of 7/16 in Figure 4 is the same as 7/16 duty cycle.

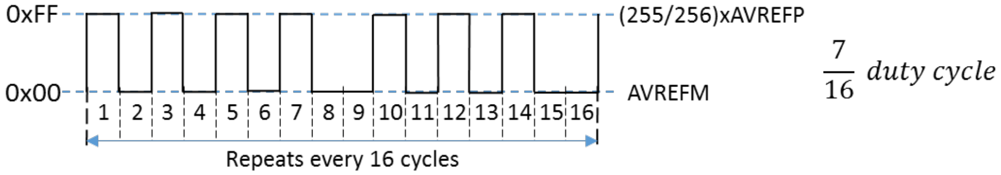


Figure 4: Method to measure RC low-pass filter output ripple

This method is included in the Application note Software Project as a test mode to facilitate Dithering ripple amplitude measurement. (when `dac_dither_full_mag_test_flag = 1;` in r_main.c)

6.5 Typical DAC dithering ripple amplitude from RC low filter

Using the section 6.4 ripple measuring method, the following tables shows approximate measured RC low pass ripple amplitudes for each of the 15 possible dithering patterns. This table was generated from empirical data, using a 320uSEC repeating period DAC dithering pattern and RC low-pass filter with time constant = 440uSEC. If the User desires to implement a different DAC dithering repeat period and change the RC low-pass filter time constant, it may be useful to re-measure the RC filter output ripple to confirm filter performance.

From a frequency domain standpoint, in Table 5 it can be seen the largest ripple amplitude is for the lowest Dither pattern repeat frequency of 16 cycles, and has the lowest ripple amplitude at the highest Dither pattern repeat frequency of 2 cycles. That is because the RC low-pass filter attenuates increasingly higher frequencies more.

Dither Sequence Number N (duty = N/16)	Dither Sequence Pattern repeats every N cycles of DTC transfer	Ripple Amplitude relative to 1LSB of 8bit DAC	Ripple Percentage of 1LSB at 12bit effective resolution
1, 15	16	0.0416	67%
2, 14	8	0.0387	62%
3, 13	16	0.0420	67%
4, 12	4	0.0337	54%
5, 11	16	0.0422	68%
6, 10	8	0.0389	62%
7, 9	16	0.0434	69%
8	2	0.0227	36%

Table 5: Relative ripple amplitude
(based on 320uSEC dither repeat period, 440uSEC RC low-pass filter time constant)

The ripple amplitude listed in Table 5 is the fraction relative to two adjacent DAC codes peak-peak amplitude (example; 0x80 and 0x81) normalized to a value of one. When this is extrapolated to a 1LSB pk-pk amplitude of 8-bit DAC, the ripple percentage becomes approximately 36%-69% of 1LSB at 12-bit resolution.

As can be seen in Table 5, the minimum ripple is for Dithering sequence N = 8, where the DAC output looks like a square wave between 2 adjacent DAC values (example: 0x80 and 0x81). The worst case ripple amplitude occurs for dither sequence N = 7 and N = 9, where the ripple amplitude is almost twice that of N = 8. It should be noted that the ripple amplitude is identical for Dither sequence N and 16-N, where N = 1 to 15. For example, the ripple amplitude for Dither sequences 3 and 13 is the same, and for Dither sequences 7 and 9 is the same. That is because the Dither Sequence pattern is just inverted between sequence 3 and 13, and between sequence 7 and 9, and therefore the ripple magnitude is the same amplitude but inverted for inverted dither pattern sequences.

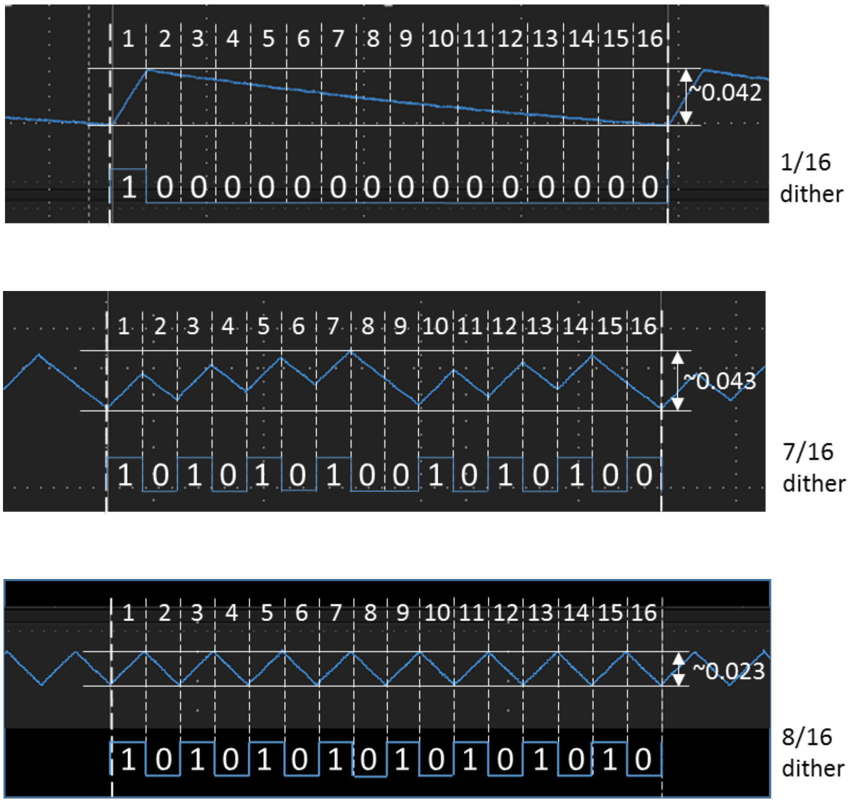


Figure 5: Typical RC low-pass filter ripple waveforms

(ripple output listed as a fraction of input normalized to 1.0)

6.6 Understanding RC low pass filter output loading

This app note sample project uses a 200KΩ series resistance and C = 2.2nF RC filter to avoid loading the unbuffered 8-bit DAC too much. There is a tradeoff on RC filter impedance. If the RC filter series resistance is too low, approaching the typical output DAC impedance, R = 5K-10KΩ, then there will be a large voltage drop due to loading the unbuffered DAC output too much. The DAC output impedance R is not specified in RL78 MCU electrical specifications section, but is estimated to be 5K-10KΩ typically, and possibly as high as 39KΩ worst case.

However, if the series R of the RC filter is too large, then there will be larger measurement errors when using a 10MΩ input impedance of a typical scope probe or DC voltmeter (10MΩ to 11MΩ typical for instrumentation quality voltmeter). Also, as impedance of the RC filter goes higher, the external system noise may affect the desired DAC-like 12-bit effective DC output value more. For example, an R = 200KΩ and C = 2.2nF gives a 440uSEC time constant compared to R = 2MΩ and C = 220pF where the RC time constant is still 440uSEC, but the filter impedance is 10x higher for the latter values.

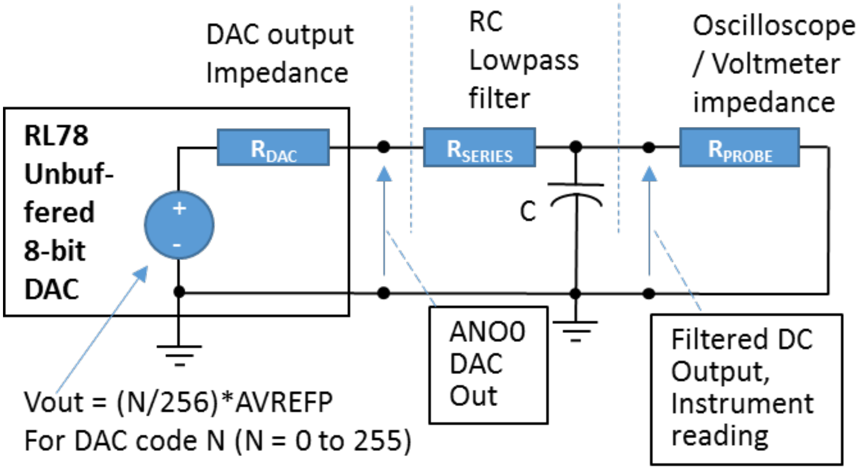

DAC output Impedance
RC Lowpass filter
Oscilloscope / Voltmeter impedance
RL78 Unbuf-fered 8-bit DAC
RDAC
RSERIES
RPROBE
+
-
C
ANO0 DAC Out
Filtered DC Output, Instrument reading
Vout = (N/256)*AVREFP
For DAC code N (N = 0 to 255)

Figure 6: Impedance Model of DAC output with RC Low-pass filter and instrument loading

In **Figure 6 (**Impedance model**)** above, loading both the ANO0 DAC output and RC Low-pass filter output with x10 (10MΩ) scope probe or 10MΩ input voltmeter will affect the results somewhat. The total DAC load will still be greater than 5MΩ (10MΩ in parallel with 10.2MΩ), greater than the RL78 minimum 8-bit DAC loading resistance of 4MΩ minimum for guaranteed specification of DAC overall error = +/-2.5% maximum.

However, if using a X1 (1MΩ) scope probe on either the ANO0 DAC output or the (filtered) RC Low-pass filter DC output, there will be significant errors. For example, if a 1MΩ Scope probe is placed only on the RC low pass filter output (no dithering, with R_{DAC} = 10KΩ and R_{SERIES} = 200KΩ), the output will result in approximately:

$$RC\ filtered\ output = R_{PROBE} / (R_{DAC} + R_{SERIES} + R_{PROBE}) * \left(\frac{N}{256}\right) * AVREFP$$

$$= 1M\Omega / (0.01M\Omega + 0.2M\Omega + 1M\Omega) * \left(\frac{N}{256}\right) * AVREFP$$

$$= 0.826 * \left(\frac{N}{256}\right) * AVREFP$$

Conclusion: Using a 1MΩ oscilloscope probe or voltmeter impedance on RC filter output will reduce the DC level by (100% - 82.6%) = ~17.4%

Alternatively, if a 10MΩ probe is used;

$$RC\ filtered\ output = 10M\Omega / (0.01M\Omega + 0.2M\Omega + 10M\Omega) * \left(\frac{N}{256}\right) * AVREFP$$

$$= 0.979 * \left(\frac{N}{256}\right) * AVREFP$$

Conclusion: Using a 10MΩ oscilloscope probe or voltmeter impedance on RC filter output will reduce the DC level by (100% - 97.9%) = ~2.1%. The RL78 specification for minimum resistance 8-bit DAC loading is 4MΩ, to meet the max Overall 8-bit DAC error of +/-2.5LSB but there can still be up to -2.1% level drop between no loading and 10MΩ DC loading.

In actuality, almost any DC loading will lower the RC filtered DC output by some percentage. The closed loop correction mode can somewhat help overcome the DC loading drop by increasing the DAC dither level but cannot compensate for all loads at higher 12-bit effective codes. The open loop DAC dithering method will definitely exhibit lowered DC output at the RC filter when loading resistance is decreased (smaller R value). In empirical tests, even a 10MΩload on the combination R= 200KΩ and C = 2.2nF, RC filter caused the filter output ADC reading to droop to 0xF9F code reading when the Target DC value was set to 0xFEF. However, setting the Target DC value to 0xF9F, the closed loop method as able to compensate to an actual average of 0xF9F. (typical example, not worst case.) The ADC input impedance is extremely high (maybe >100MΩ) and typically won't affect the RC filter loading for R = 200KΩ.

The result is that almost any normal test instruments with 10MΩ input impedance will adversely affect the DAC dithering results relative to a very high (50MΩ-100MΩ) impedance buffer amplifier or MCU ADC input.

6.7 DAC Dithering output vs ripple output observations

If 10MΩ scope probes are attached to both the ANO0 DAC output and the RC filter output, the observed waveforms will be approximately as shown in **Figure 7.**

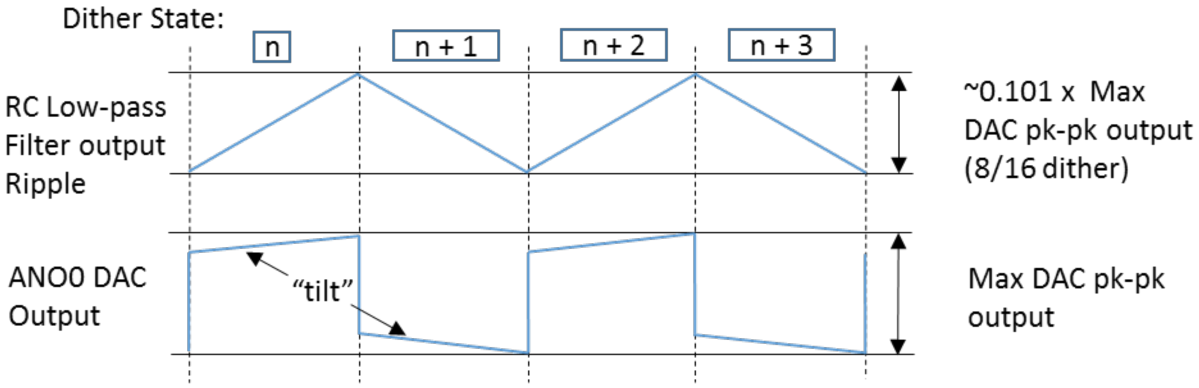

Dither State:
n
n + 1
n + 2
n + 3
RC Low-pass Filter output Ripple
~0.101 x Max DAC pk-pk output (8/16 dither)
ANO0 DAC Output
"tilt"
Max DAC pk-pk output

Figure 7: Example of measuring RC Low-pass filter output at max DAC pk-pk amplitude

Notes:

- *Figure 7. Waveforms not to scale, and "tilt" amplitude is exaggerated to show effect*
- *Max DAC pk-pk output is for 0x00 to 0xFF amplitude = (255/256)*(AVREFP-AVREFM), although the actual amplitude will be reduced by about 2.1% due to 10Megohm scope probe loading*

The "tilt" on ANO0 DAC output (slightly exaggerated here) is normal due to the internal DAC output impedance 5KΩ-10KΩ, acting as a voltage divider with the 200KΩ RC filter series resistor, and the tilt shows up as the RC filter capacitor is charging and discharging. When AVREFP = VDD = 5Volts, a DAC output impedance of 5KΩ-10KΩ can cause a tilt "difference" value of approximately 61millivolts (with 5KΩ DAC impedance) to 119millivolts (with10K DAC impedance) and could theoretically be higher if DAC output impedance (R_{DAC}) = 39KΩ worst case.

If RC filter output ripple measurements are made, either the AVREFP*(255/256) value can be used as a reference approximation, or the max DAC peak to peak amplitude seen on ANO0 DAC output can be used.

7. Attributes of D/A Convertors (DAC) using R-2R ladder

7.1 R-2R resistor ladder used in RL78 8-bit DAC

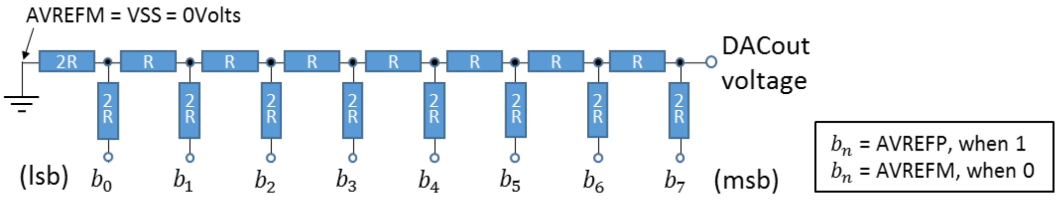


Figure 8: R-2R ladder configuration

On a D/A convertor R-2R resistor ladder, the output impedance is R. The RL78 DAC is unbuffered, meaning that if R is in the range of 5KΩ to 39KΩ, any significant loading can cause the output to "droop", causing a DC error. In fact the RL78 electrical specification for 8-bit DAC shows a minimum loading resistance of 4MΩ to still meet the worst case DAC output voltage of +/-2.5 LSB maximum overall error.

7.2 Ideal equation for 8-bit DAC output voltage

Using the RL78 full scale Analog reference voltages of AVREFP and AVREFM minimizes noise coupling for both the DAC and ADC units, and gives the optimum accuracy for 10-bit ADC conversion (+/-3.5LSB max overall error). So it is recommended to use the AVREFP/AVREFM voltage reference for the DAC dither project. Then, for the 8-bit DAC, there are 256 possible voltage outputs, but only 255 steps above the 0x00 DAC output value. Since the full-scale

Analog reference value is AVREFP-AVREFM, but maximum code is 0xFF (255 decimal), the maximum DAC output is 255/256*AVREFP (when AVREFM = 0Volts). This can be seen in the following equations (assuming no loading on the unbuffered DAC output) in the ideal case with no other DAC output errors:

$$\text{DACout voltage} = \left(\frac{b_7}{2}+\frac{b_6}{4}+\frac{b_5}{8}+\frac{b_4}{16}+\frac{b_3}{32}+\frac{b_2}{64}+\frac{b_1}{128}+\frac{b_0}{256}\right) * (AVREFP - AVREFM)$$

When AVREFM = VSS = 0Volts, and DAC code is 0xFF;

$$\text{DACout voltage} = \left(\frac{1}{2}+\frac{1}{4}+\frac{1}{8}+\frac{1}{16}+\frac{1}{32}+\frac{1}{64}+\frac{1}{128}+\frac{1}{256}\right) * AVREFP = \frac{255}{256}AVREFP$$

Actually this is the case for all DACs (and successive approximation type ADCs) that use the R-2R resistor ladders, that the maximum DAC output voltage (or maximum ADC input voltage) is always 1 LSB below the full-scale analog voltage, assuming the ideal case with no other errors but in reality there are DAC errors due to slight AVREFP and AVREFM voltage differences from one 2R tap to another, as well as minute differences in actual R resistance values!

Range	ratio of AVREFP	8bit DAC Output Code		10bit ADC conversion code		ratio of AVREFP	Range
		Hex value	Decimal	Hex value	Decimal		
Not Valid	256/256	0x100	256	0x1000	1024	1024/1024	Not Valid
				0x3FF	1023	1023/1024	Valid Range
				0x3FE	1022	1022/1024	
				3FD	1021	1021/1024	
Valid Range	255/256	0xFF	255	0x3FC	1020	1020/1024	
				0x3FB	1019	1019/1024	
				0x3FA	1018	1018/1024	
				0x3F9	1017	1017/1024	
	254/256	0xFE	254	0x3F8	1016	1016/1024	
				0x3F7	1015	1015/1024	
				0x3F6	1014	1014/1024	
				0x3F5	1013	1013/1024	
	.	.	.	.	.	.	
	.	.	.	.	.	.	
	.	.	.	.	.	.	
	1/256	0x01	1	0x007	7	7/1024	
				0x006	6	6/1024	
				0x005	5	5/1024	
				0x004	4	4/1024	
	0/256	0x00	0	0x003	3	3/1024	
				0x00	2	2/1024	
				0x00	1	1/1024	
				0x00	0	0/1024	

Table 6: Ideal DAC/ADC max values as ratio of AVREFP

(ideal values, not including real-world DAC and ADC accuracy errors)

7.3 Limitation of Dynamic range using DAC Dithering method

There are 256 possible output values of the 8-bit DAC, but the dithering requires toggling between 2 adjacent DAC output values. Using a 16 state dithering sequence there are only 15 different dithering sequences between any 2 DAC output states. As can be seen, the Dither adder sequence 0 (in Table 2.) is all zeroes which means no dithering is done.

Otherwise, dithering can only be applied on 255 combinations of 2 adjacent DAC codes, so there are only 4081 possible values:

Maximum range of Dithered "12-bit" output values =

256 (DAC codes 0x00 to 0xFF, no dithering) + 255 x 15 dithering sequences

= 4081 values,

0-4080 decimal (0x000 to 0xFF0)

So when using the DAC Dithering method, the effective 12-bit resolution target setting cannot exceed the 12-bit code of 0xFF0, which is the same maximum of the 8-bit DAC output code.

Note: *Although the DAC dithering output setting is limited to a max target value of 0x0FF0, it is possible that ADC conversion values of 0xFF1 to 0xFFF may still occur, due to DC offsets/errors in both the DAC output and ADC conversion.*

7.4 RL78 8-bit DAC accuracy relative to 10-bit ADC accuracy

The RL78 8-bit DAC electrical specification only specifies one error value; Overall error of +2.5LSB max (for 4MΩ loading or greater), which includes AINL (Analog Integral Non-Linearity) error. In reality, the DAC Overall error includes Zero-scale, Full-scale and Differential Non-Linearity (DNL) error. What this means is the 8-bit DAC output absolute error can change somewhat from one DAC code to another, throughout the range of the DAC output code values. The +/-2.5LSB max error for 8-bit DAC is equivalent to +/-10LSB variance range for 10bit ADC conversions. Since the RL78 10-bit ADC overall error is +/-3.5LSB max (when using ADC full-scale analog reference of AVREFP and AVREFM connections), the 10-bit ADC has almost 2.9X better accuracy than the 8-bit DAC output accuracy. However, the 10-bit ADC accuracy of +/-3.5LSB (at 10-bit resolution) overall error means that the overall error in monitoring the 12bit effective Dithered DAC (RC filtered) output average value could be +/-14LSB (4x 3.5) or more relative to the ideal 12bit equivalent, effective DC value from DAC dithering.

As can be seen in this graph (Figure 9.), 8-bit DAC output varies from code-to-code when measured by averaged 10-bit ADC conversions. This graph shows a general trend for the DAC output to be high (> +1LSB at 10bit resolution) for codes of 0 to 48 decimal, and to be low (< -1LSB at 10bit ADC resolution) for output codes of about 242 decimal and higher. This graph is just measured on one RL78/G14 MCU, and may not represent a typical RL78/G14 MCU. On one hand, this data seems to indicate the 8-bit DAC output accuracy can be tighter than the DAC Overall error maximum specification, but still has an error range of +2LSB to -1.5LSB at 10-bit ADC resolution. However, this graph does not address the possible inherent 10bit ADC conversion error of +/-3.5LSB (10bit resolution).

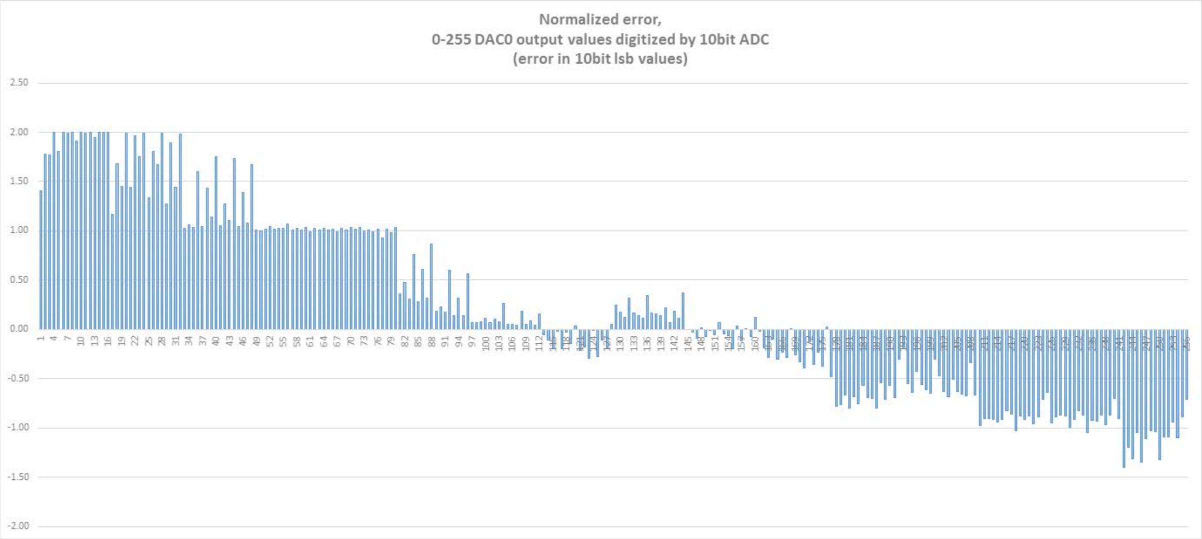


Figure 9: Typical 8-bit DAC output measured by 10-bit ADC
(each point averaged from 256 consecutive ADC conversions per DAC code)

8. Typical monitoring results of DAC dithering Output

The app note Software Project includes an optional UART1 output (provided on RSKRL78G14 board DB-9 connector) to provide real-time monitoring of the DAC dithering output, both before and after RC filtering. UART1 is set for 115.2Kbps, and each 10bit character takes about 86.8uSEC. Each hex value of 3 ASCII characters (plus comma or LF character) in section 8.1 and 8.2 is output at 1024HZ (976.6uSEC) rate and takes about 347uSEC for 4characters. The UART1 print function was intended for troubleshooting and accuracy measurement, but is not needed to implement the DAC dithering open-loop or closed-loop functions.

8.1 UART output for RC filtered DAC dithering, open-loop mode

When variables `enable_print = 1` and `g_closed_loop = 0` in r_main.c, UART1 will output the sum of 16 consecutive ADC conversion, scaled down to a 12bit Hex value, with no correction on the DAC dithering. The first (single) hex value is the target 12-bit effective dithered DAC DC value. Each line thereafter represents 16 averaged/scaled RC filtered output values (each value is sum of 16 ADC 10-bit conversions). So each line occurs at 64HZ rate (16384HZ sample rate/256 total samples). In this example output, the `target_dac12_value = 0xFE6`. After about 16mSEC startup time, the dithered DAC RC filtered output settles to 0xFE2, which is an offset of about -4 steps below the target 0xFE6 value in 12bit effective DAC resolution.

```
FE6
FFF,FFF,FFF,FFF,FFF,FFF,FFF,FFF,FFF,FFF,FFF,FFF,FFF,FFF,FFF,FFF
FFF,FE2,FE2,FE2,FE2,FE2,FE2,FE2,FE2,FE2,FE2,FE2,FE2,FE2,FE2,FE2
FE2,FE2,FE2,FE2,FE2,FE2,FE2,FE2,FE2,FE2,FE2,FE2,FE2,FE2,FE2,FE2
```

(…continues indefinitely)

Figure 10.: UART1 print monitoring in Open-loop mode

8.2 UART output for RC filtered DAC dithering, closed-loop mode

When variables `enable_print = 1` and `g_closed_loop` = 1 in r_main.c, UART1 will output the sum of 16 consecutive ADC conversion, scaled down to a 12bit Hex value, with the DAC Dithering value corrected once on each line (15.625mSEC correction rate). The data output is the same format as described in section 8.1 above. In this example, again the `target_dac12_value = 0xFE6`. However due to the closed-loop ADC measurement and correction, the dithered DAC RC filtered output settles to the target 0xFE6 value in 12bit effective DAC resolution but takes about 7 periods of 15.625mSEC correction to achieve stable output.

```
FE6
FFF,FFF,FFF,FFF,FFF,FFF,FFF,FFF,FFF,FFF,FFF,FFF,FFF,FFF,FFF,FFF
FFF,FE2,FE2,FE2,FE3,FE2,FE2,FE2,FE2,FE3,FE2,FE2,FE2,FE2,FE3,FE2
FE2,FE2,FE2,FE2,FE2,FE2,FE3,FE3,FE3,FE3,FE3,FE3,FE3,FE3,FE3,FE3
FE3,FE3,FE3,FE3,FE3,FE3,FE3,FE4,FE4,FE4,FE4,FE4,FE4,FE4,FE4,FE4
FE5,FE5,FE5,FE5,FE5,FE5,FE5,FE5,FE5,FE5,FE5,FE5,FE5,FE5,FE5,FE5
FE5,FE5,FE5,FE5,FE6,FE6,FE6,FE6,FE6,FE6,FE6,FE6,FE6,FE6,FE6,FE6
FE6,FE6,FE7,FE7,FE6,FE7,FE6,FE7,FE7,FE7,FE7,FE7,FE7,FE7,FE7,FE7
FE6,FE6,FE6,FE6,FE6,FE6,FE6,FE6,FE6,FE6,FE6,FE6,FE6,FE6,FE6,FE6
```

(…continues indefinitely)

Figure 11.: UART1 print monitoring in Closed-loop mode

8.3 UART output for measuring DAC output offset errors

When `g_dacout_test_flag = 1` in r_main.c, and function `dacout_test_to_adc_input()` is run, the 10bit ADC measures each of 8-bit DAC codes 0x00 to 0xFF, for 256 conversions per DAC output code. Each line evaluates a single 8-bit DAC output code, starting with 0x00 and ending with `0xFF.` Each 4 character value corresponds to the sum of 16 consecutive 10-bit ADC conversions, so each value can contain up to 14bit results, not scaled down. Possible output range is 0x0000 to 0x3FFC max (0 to 16380 decimal).

From this example below, it is seen the 8-bit DAC code of 0xFF, when measured 256 times has a 14bit average value of 0x3FB0, or about 0x3FB at 10-bit resolution, which is about 1LSB (at 10-bit resolution) below the predicted max value for 0xFF DAC output with 10-bit ADC conversion. (Reference Table 6.)

```
0013,0015,0015,0012,0013,0012,0013,0010,0013,0010,0012,0012,0012,0013,0012,0013
005E,005D,005C,005D,005B,005A,005C,005D,005A,005E,005B,005D,005D,005B,005D,005B
0099,009B,0099,009B,0097,009A,009A,009C,009B,009B,009A,0097,009D,0099,0098,009B
           :
           .
3EF0,3EF0,3EF0,3EEE,3EEE,3EF1,3EEE,3EEE,3EF0,3EF0,3EF7,3EF1,3EEC,3EF0,3EF0,3EF0
3F2A,3F2E,3F2D,3F2E,3F2A,3F2E,3F35,3F2F,3F37,3F2B,3F35,3F2B,3F2E,3F2D,3F2E,3F2F
3F6A,3F70,3F70,3F70,3F70,3F70,3F70,3F70,3F70,3F6F,3F71,3F70,3F70,3F70,3F70,3F70
3FB0,3FAF,3FAF,3FB6,3FB1,3FB5,3FB0,3FB2,3FAF,3FB4,3FB0,3FB0,3FB1,3FB0,3FB0,3FB0
```

(completes after 256 total lines)

Figure 12.: UART1 print monitoring for all 256 DAC codes out to ADC input

The Figure 9. Plot in section 7.4 was generated from `dacout_test_to_adc_input()` UART1 data output similar to the above data in Figure 12, averaged and scaled down to 10bit equivalent resolution.

9. DAC Dithering increase/decrease, closed-loop algorithm

In the DAC Dithering close-loop mode, the sum of the last 256 ADC conversions (10bits x 256 = 18bit result) is scaled down and rounded off to a 12bit effective result every 15.625mSEC. Then the 12bit actual (averaged) result is compared to the User settable 12-bit target_dac12_value. If target and measured values are equal, then no DAC dithering correction is made. If the actual measured value is less than the target value, the dithering factor is increased by one bit value in 12-bit resolution. If the actual measured value is greater than the target value, the dithering factor is decreased by one bit value (12-bit resolution).

Care is taken not to produce a Dithering base value and factor that is out of range. Depending on the previous DAC output Base Value, either the Base Value, the Dithering factor, or both values will need to be updated into the 16byte RAM table transferred to DAC by DTC data transfers.

9.1 DAC Dithering Increase

For Dithering increases, a special case occurs when the previous DAC Dithering value is 0xFF0, and no increase is possible, since Dithering can only occur between DAC output codes 0xFE and 0xFF to produce 12bit DAC effective codes of 0xFE0 to 0xFEF. A constant DAC output code of 0xFF represents 0xFF0 in 12-bit resolution and no further increase is possible. Of course, there is typically some lowered DC offset when DAC output is 0xFF code and the 12bit effective DAC value is slightly less than 0xFF0.

9.2 DAC Dithering Decrease

For Dithering decreases, a limit case occurs when the previous DAC Dithering value is 0x000, and no decrease is possible, since Dithering can only occur between DAC output codes 0x00 and 0x01 to produce 12bit DAC effective codes of 0x000 to 0x00F. A constant PWM output code of 0x00 represents 0x000 in 12-bit resolution and no further decrease is possible.

10. HW implementation for Sample SW project

The application note Sample SW project was developed on an RSKRL78G14 board, populated with 100pin RL78/G14 (device R5F104PJ). This board layout was designed to maximize access to all the MCU pins, and provide a preliminary prototyping platform for customer RL78/G14 designs. However, the printed circuit board layout may not be optimum for lowest Analog noise riding on the RL78 power supply and ground lines. Nevertheless, the 10-bit ADC specification show the lowest overall ADC conversion errors (+/-3.5LSB at 10-bit resolution) when the AVREFP and AVREFM lines are used to provide the Full-scale analog voltage reference. This specification assumes ideal conditions, with no additional noise being fed into AVREFP/AVREFM lines.

In a customer-designed board, care should be taken to minimize ground loops and power supply line impedance to the Analog voltage reference pins. For the RSKRL78G14 board, connector J4 is the shortest, most convenient place to wire up the DAC and ADC lines on G14 MCU ports P2 and P15. DAC and ADC analog reference pins AVREFP and AVREFM are connected to VDD and VSS/Ground respectively, by connecting to connector J12.

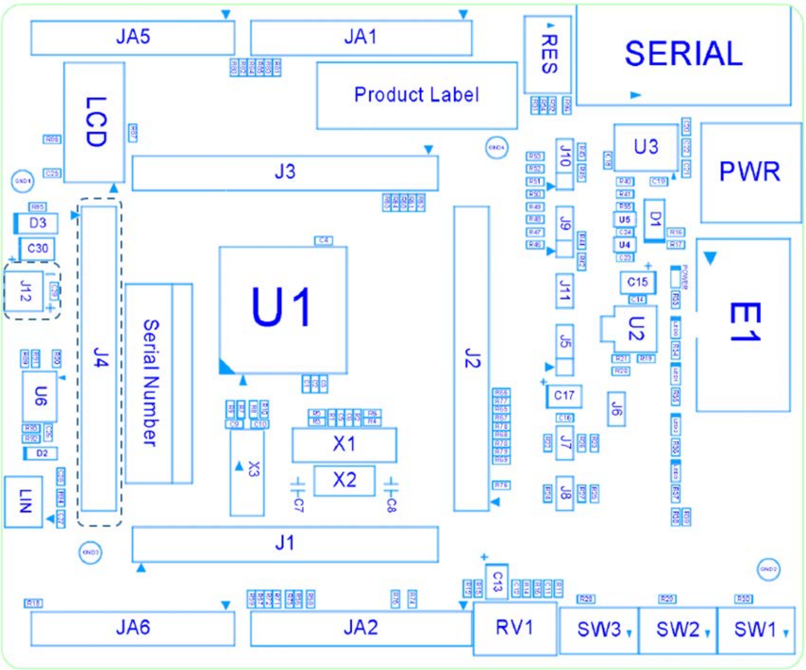
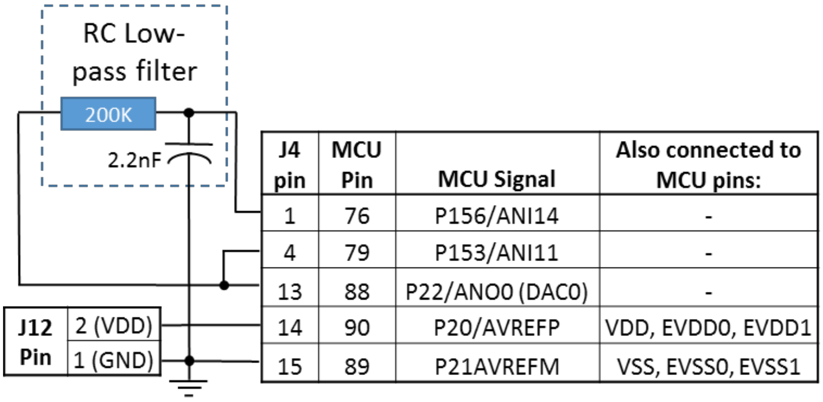


Figure 13: Standard RSKRL78G14 board (part #R0K50104PC000BE)

In wiring diagram Figure 14, ANI11 ADC input is used for directly checking the DAC output errors (without DAC dithering), and ANI14 ADC input used for checking/correcting the DAC dithering Low-pass filtered output value.



J4 pin	MCU Pin	MCU Signal	Also connected to MCU pins:
1	76	P156/ANI14	-
4	79	P153/ANI11	-
13	88	P22/ANO0 (DAC0)	-
14	90	P20/AVREFP	VDD, EVDD0, EVDD1
15	89	P21AVREFM	VSS, EVSS0, EVSS1

Figure 14: Recommended wiring for DAC dithering Sample SW project using RSKRL78G14 board

11. Documents for Reference

User’s Manual:
RL78/G14 User’s Manual Manual: Hardware Rev.3.30 (R01UH0200EJ0330)
RL78 Family User’s Manual: Software Rev.2.00 (R01US0015EJ0200)
(The latest versions of the documents are available on the Renesas Electronics Website.)

Technical Updates/Technical Brochures
(The latest versions of the documents are available on the Renesas Electronics Website.)

Website and Support

Renesas Electronics Website
http://www.renesas.com/

Inquiries
http://www.renesas.com/contact/

All trademarks and registered trademarks are the property of their respective owners.

Revision History

		Description	
Rev.	Date	Page	Summary
1.00	10Aug2017		First edition

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Handling of Unused Pins

   Handle unused pins in accordance with the directions given under Handling of Unused Pins in the manual.

   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment when power is supplied.

   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.
   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

   Before changing from one product to another, i.e. to a product with a different part number, confirm that the change will not lead to problems.

   — The characteristics of Microprocessing unit or Microcontroller unit products in the same group but having a different part number may differ in terms of the internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other disputes involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawing, chart, program, algorithm, application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics products.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots etc.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (space and undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. When using the Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat radiation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions or failure or accident arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please ensure to implement safety measures to guard them against the possibility of bodily injury, injury or damage caused by fire, and social damage in the event of failure or malfunction of Renesas Electronics products, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures by your own responsibility as warranty for your products/system. Because the evaluation of microcomputer software alone is very difficult and not practical, please evaluate the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please investigate applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive carefully and sufficiently and use Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall not use Renesas Electronics products or technologies for (1) any purpose relating to the development, design, manufacture, use, stockpiling, etc., of weapons of mass destruction, such as nuclear weapons, chemical weapons, or biological weapons, or missiles (including unmanned aerial vehicles (UAVs)) for delivering such weapons, (2) any purpose relating to the development, design, manufacture, or use of conventional weapons, or (3) any other purpose of disturbing international peace and security, and you shall not sell, export, lease, transfer, or release Renesas Electronics products or technologies to any third party whether directly or indirectly with knowledge or reason to know that the third party or any other party will engage in the activities described above. When exporting, selling, transferring, etc., Renesas Electronics products or technologies, you shall comply with any applicable export control laws and regulations promulgated and administered by the governments of the countries asserting jurisdiction over the parties or transactions.
10. Please acknowledge and agree that you shall bear all the losses and damages which are incurred from the misuse or violation of the terms and conditions described in this document, including this notice, and hold Renesas Electronics harmless, if such misuse or violation results from your resale or making Renesas Electronics products available any third party.
11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.3.0-1 November 2016)

RENESAS

SALES OFFICES

Renesas Electronics Corporation

http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HAL II Stage, Indiranagar, Bangalore, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 6.0