Application Note

PWM Control for PC Fans

AN-CM-248

Abstract

This Application Note details how Renesas Electronics built a fully-featured 12 V PC fan PWM controller with GreenPAK™ programmable mixed-signal ASICs. The project involves rotary encoding, PWM control, PCB design, and C# application programming.

This document is complimented by design files, which are listed in the References section.

Contents

Figures

Tables

1 Terms and Definitions

ASIC	Application-specific integrated circuit
CCW	Counterclockwise
CW	Clockwise
I^2C	Inter-integrated circuit
PCB	Printed circuit board
PWM	Pulse width modulation

2 References

For related documents and software, please visit:

https://www.dialog-semiconductor.com/products/greenpak.

Download our free GreenPAK™ Designer software [1] to open the .gp files [2] and view the proposed circuit design. Use the GreenPAK development tools [3] to freeze the design into your own customized IC in a matter of minutes. Renesas Electronics provides a complete library of application notes [3] featuring design examples as well as explanations of features and blocks within the IC.

[1] GreenPAK Designer Software, Software Download and User Guide, Renesas Electronics

[2] AN-CM-248 PWM Controller for PC Fans.gp, GreenPAK Design File, Renesas Electronics

[3] GreenPAK Development Tools, GreenPAK Development Tools Webpage, Renesas Electronics

[4] GreenPAK Application Notes, GreenPAK Application Notes Webpage, Renesas Electronics

[5] SLG46108, https://www.dialog-semiconductor.com/products/slg46108, Renesas Electronics

[6] SLG46826, https://www.dialog-semiconductor.com/products/slg46826, Renesas Electronics

[7] AN-1101 Unclocked Quadrature Decoder, https://www.dialog-semiconductor.com/greenpak-application-notes?combine=1101, Renesas Electronics

[8] DMP3085LSD-13DICT-ND, https://www.digikey.com/product-detail/en/diodes-incorporated/DMP3085LSD-13/DMP3085LSD-13DICT-ND/4251594, Diodes Incorporated

[9] QS5K2CT-ND, https://www.digikey.com/product-detail/en/rohm-semiconductor/QS5K2TR/QS5K2CT-ND/1144089, Rohm Semiconductor

3 Introduction

This document describes building a fully-featured 12 V PC fan PWM controller. The design can control up to 16 3-pin computer fans. The design uses a pair of GreenPAK programmable mixed-signal ASICs to control each fan's duty cycle. It also includes two ways to change the speed of the fan:

a. with a quadrature/rotary encoder

with a Windows application built in C# that communicates with the GreenPAK ICs through I²C.

3.1 System Block Diagram

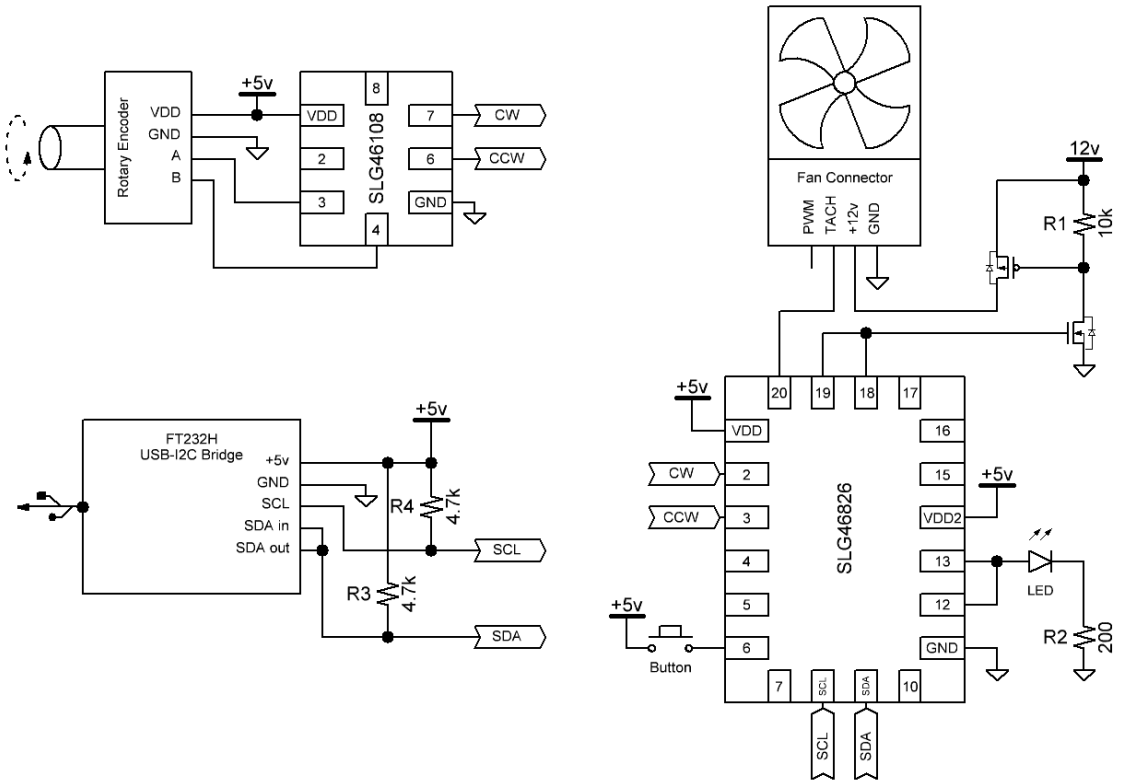

Figure 1. System Block Diagram

4 SLG46108 Rotary Decoder Design

A rotary encoder is used to increase or decrease the duty cycle of the fans manually. This device outputs pulses on its Channel A and Channel B outputs that are 90 ° apart. See AN-1101: Unclocked Quadrature Decoder [7] for more information about how a rotary encoder works.

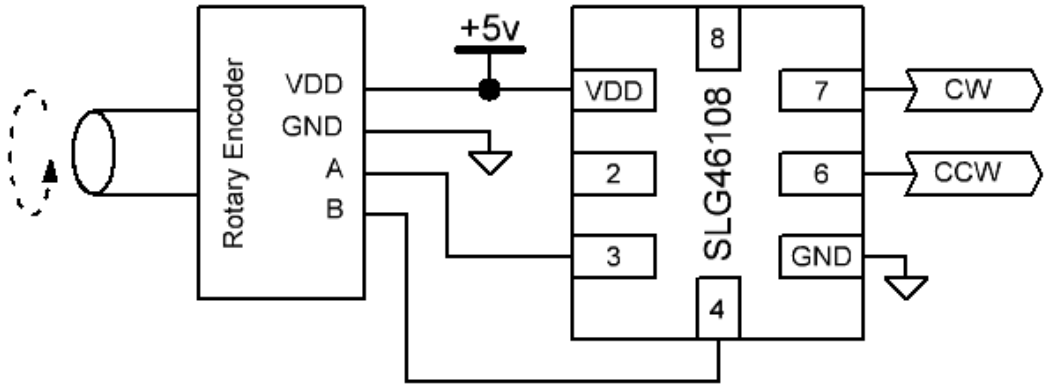

Figure 2: Rotary Encoder Block Diagram

A clocked rotary decoder can be created using a GreenPAK SLG46108 to process the Channel A and Channel B signals and output them as counterclockwise (CCW) and clockwise (CW) pulses.

When Channel A leads Channel B, the design outputs a short pulse on CW. When Channel B leads Channel A, it outputs a short pulse on CCW.

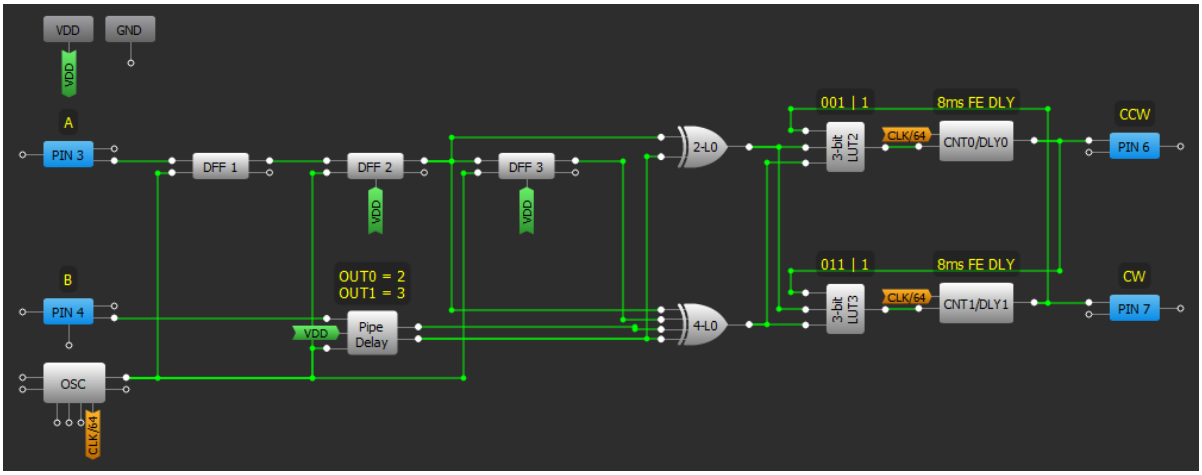

Figure 3. GreenPAK SLG46108 Rotary Decoder Design

Three DFFs synchronize the Channel A input with the clock. Similarly, the pipe delay with OUT0 set to two DFFs and OUT1 set to three DFFs create the same functionality for channel B.

To create CW and CCW outputs use a few LUTs, for more information about this standard rotary decoder design, visit this website.

The GreenPAK Rotary Decoder will receive input pulses A and B and output the CW and CCW pulses as shown in Figure 4.

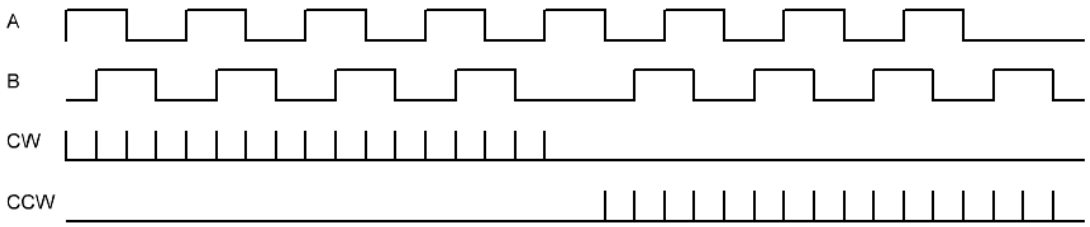

Figure 4. Timing Diagram of Rotary Signals

The circuitry after the XOR gates ensures that there will never be a CW pulse and CCW pulse at the same time, allowing for any error with the rotary encoder. The 8 ms falling edge delay on the CW and CCW signals force them to stay high for 8 ms plus one clock cycle, which is necessary for the downstream SLG46826 GreenPAKs.

5 SLG46826 Fan Controller Design

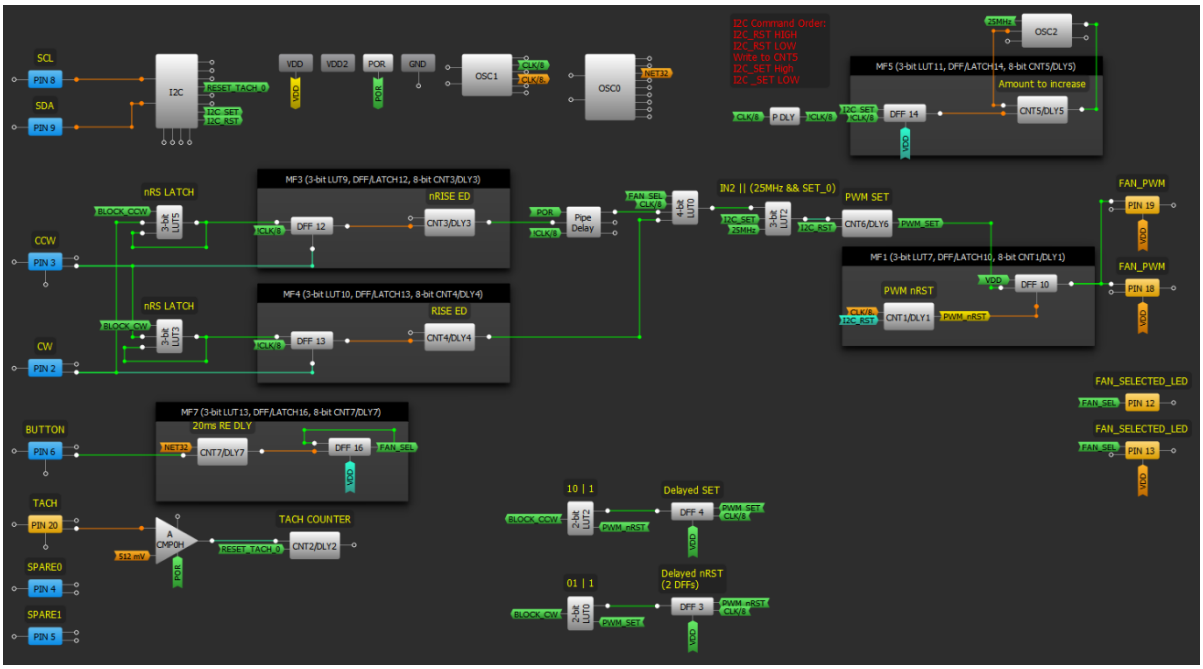

Figure 5. GreenPAK SLG46826 Fan Controller Design

5.1 PWM Generation with Offset Counters

A pair of offset counters with the same period are used to generate the PWM signal. The first counter sets a DFF, and the second resets it, creating a consistent duty cycle PWM signal as shown in Figure 6 and Figure 7.

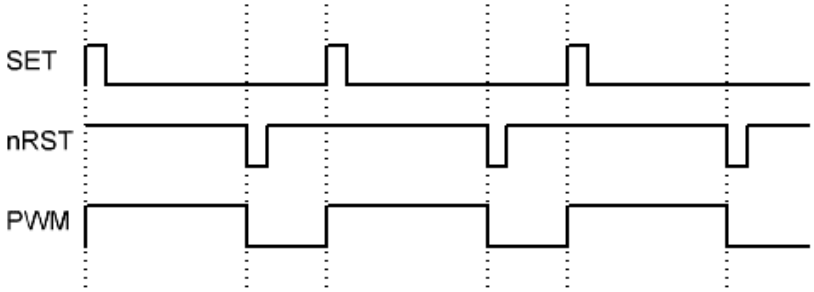

Figure 6. PWM Generation Timing Diagram

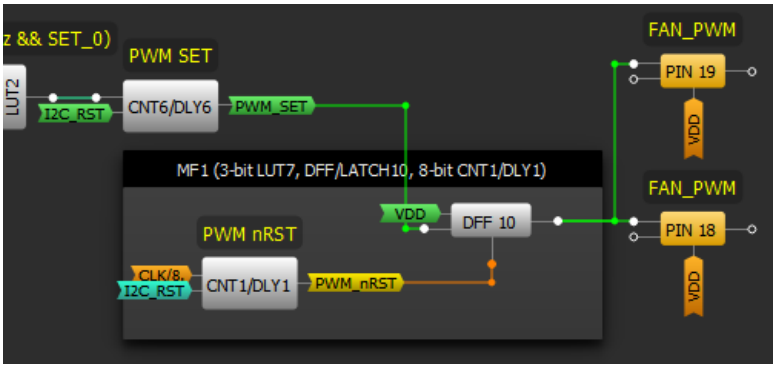

Figure 7. PWM Generation with Offset Counters

CNT6 sets DFF10 and the inverted output of CNT1 resets DFF10. Pins 18 and 19 are used to output the PWM signal to external circuitry.

5.2 Duty Cycle Control with Clock Injection and Clock Skipping

The fan controller receives the CW and CCW signals as inputs from the rotary decoder and uses them to either increase or decrease the PWM signal that controls the fan speed. This is achieved with several digital logic components.

The duty cycle needs to increase when a CW pulse is received. This is done by injecting an extra clock pulse into the CNT6 block, causing it to output one clock cycle earlier than it otherwise would have. This process is shown in Figure 8.

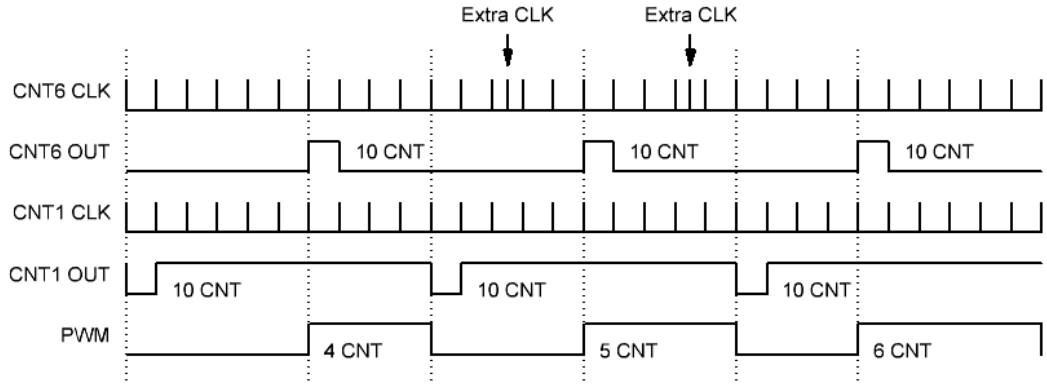

Figure 8. Clock Pulse Injection

CNT1 is still getting clocked at a constant rate, but CNT6 has a couple of extra clocks injected. Every time there is an extra clock to the counter, it shifts its output one clock period to the left.

Conversely, to decrease the duty cycle, skip a clock pulse for CNT6 as shown in Figure 9. CNT1 is still getting clocked at a constant rate, and there are skipped clock pulses for CNT6, where the counter did not get clocked when it was supposed to. This way the output of CNT6 is pushed to the right by one clock period at a time, shortening the output PWM duty cycle.

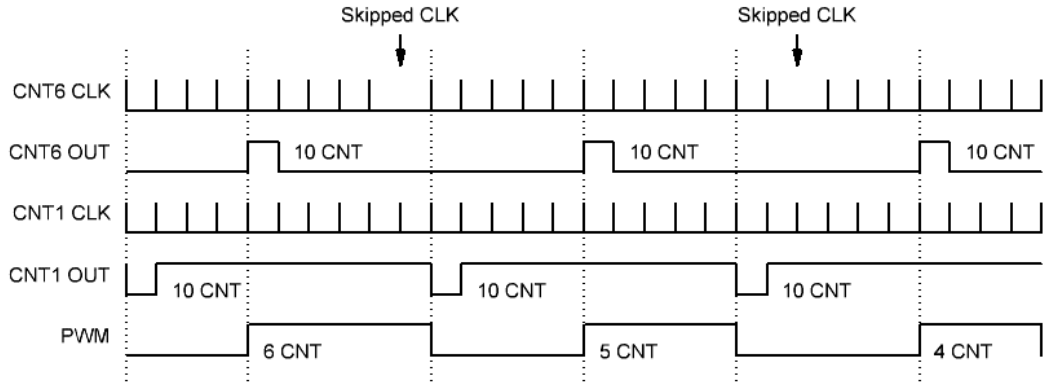

Figure 9. Clock Pulse Skipping

The clock injecting and clock skipping functionality is performed with use of some digital logic elements within the GreenPAK. A pair of multifunction blocks are used to create a pair of latch/edge detector combos. 4-bit LUT0 is used to mux between the general clock signal (CLK/8) and the clock injecting or clock skipping signals. This functionality is described in more detail in Section 5.2.2.

5.2.1 BUTTON Input

The BUTTON input is debounced for 20 ms, then used to toggle a latch that determines whether this particular chip is selected. If it is selected, then the 4-bit LUT passes the clock skipping or injection signals. If the chip is not selected, then the 4-bit LUT simply passes the CLK/8 signal.

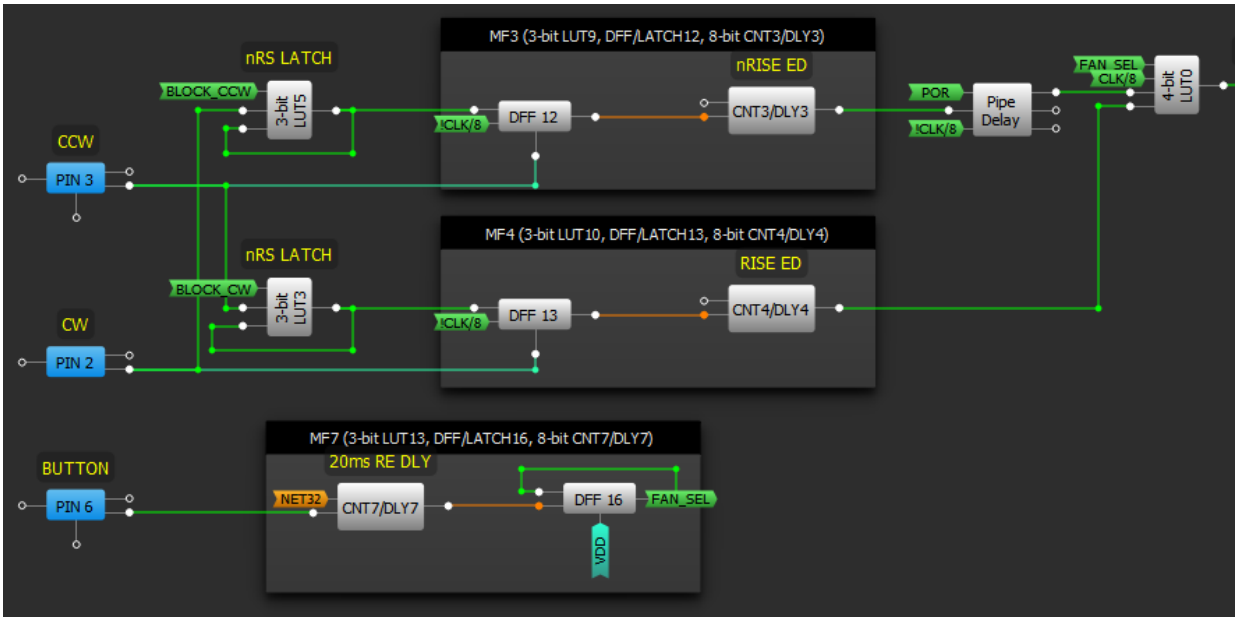

Figure 10. Clock Skipping and Clock Injecting

5.2.2 Preventing Duty Cycle Rollover

The RS latches 3-bit LUT5 and 3-bit LUT3 are used to make sure that you cannot inject or skip so many clocks that the offset counters roll over. This is to avoid the system reaching 100 % duty cycle and then rolling over to a 1 % duty cycle if it receives another injected clock.

The RS latches prevent this from happening by latching the inputs to the multifunction blocks when the system is one clock cycle away from rolling over. A pair of DFFs delay the PWM_SET and PWM_nRST signals by one clock period as shown in Figure 11.

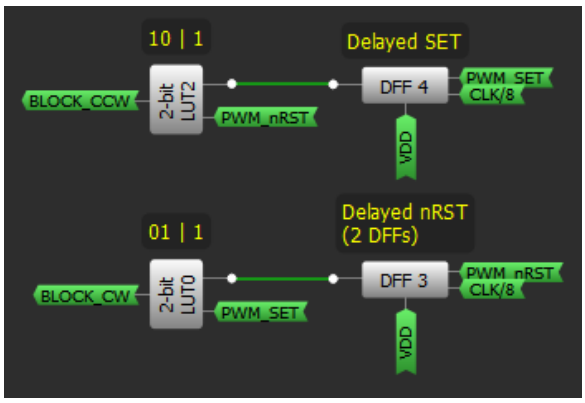

Figure 11. BLOCK_CW and BLOCK_CCW

A pair of LUTs are used to create the necessary logic. If the duty cycle is so low that the delayed PWM_SET signal occurs at the same time as the PWM_nRST signal, a further decrease in the duty cycle will cause a rollover.

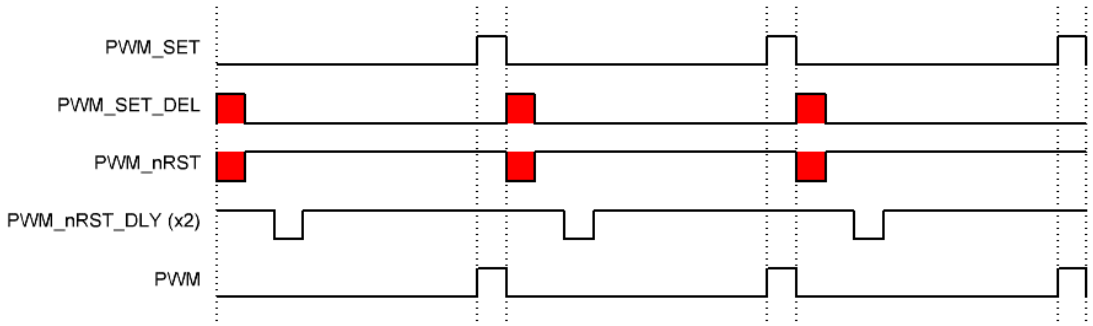

Figure 12. Minimum Duty Cycle Rollover Case

Similarly, if approaching maximum duty cycle, such that the delayed PWM_nRST signal occurs at the same time as the PWM_SET signal, it is necessary to avoid any further increase to the duty cycle. In this instance, delay the nRST signal by two clock cycles to ensure that the system does not roll over from 99 % to 1 %.

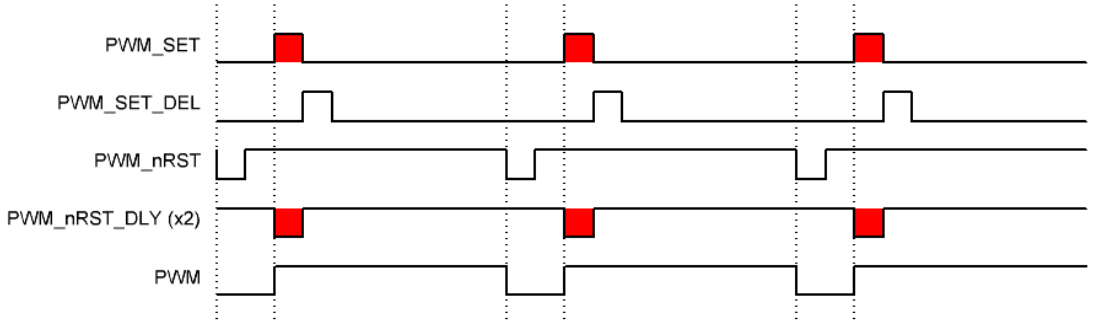

Figure 13. Maximum Duty Cycle Rollover Case

5.3 Duty Cycle Control with I²C

This design incorporates another way to control the duty cycle other than clock skipping/clock injecting. An external microcontroller can be used to write I²C commands to the GreenPAK to set the duty cycle.

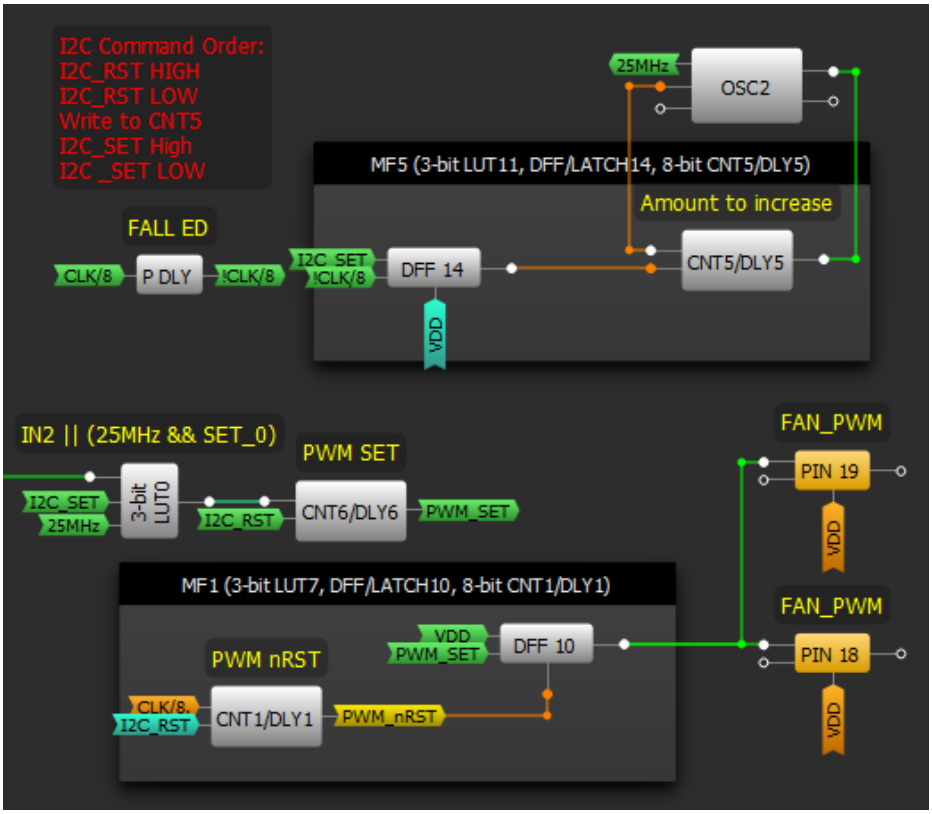

Figure 14. I²C Control of Duty Cycle

Controlling the duty cycle over I²C requires the controller to perform a specific command sequence. These commands are shown in order in Table 1. An "x" indicates a bit that should not change, "[" indicates a START bit, and "]" indicates a STOP bit.

Table 1: Duty Cycle I²C Commands

Desired Function	I²C Command (with slave address of 0001)	Purpose
Set I2C_RST HIGH	[0x10 0x7A xxxx xxx1]	Reset CNT6 and CNT1 at the same time so that their outputs are synchronized
Set I2C_RST LOW	[0x10 0x7A xxxx xxx0]	Release the reset on CNT6 and CNT1 so that they start counting
Write to CNT5	[0x10 0xBC (0-255)]	Load the OneShot pulse with the desired amount to increase the duty cycle
Set I2C_SET HIGH	[0x10 0x7A xxxx xx1x]	Start the CNT5 OneShot pulse on the next falling edge of CLK/8
Set I2C_SET LOW	[0x10 0x7A xxxx xx0x]	Reset the I2C_SET signal to LOW

The PDLY block generates a short active high pulse on the falling edge of the CLK/8 signal, which is called !CLK/8. That signal is used to clock DFF14 at a steady frequency. When I2C_SET goes high asynchronously, the next rising edge of !CLK/8 causes DFF14 to output HIGH, which triggers the CNT5 OneShot. The OneShot runs for the number of clock cycles that the user wrote as specified in the "Write to CNT5" I²C command in Table 1. In this case, it is 10 clock cycles. The OneShot allows

the 25 MHz oscillator to run for exactly its duration and no longer, so that 3-bit LUT0 receives the number of clock cycles that were written to CNT5.

Figure 15 shows these signals, where the red clocks are the ones that are sent to 3-bit LUT0, which passes them into CNT6 (the PWM_SET counter), thus creating the offset for the duty cycle generation.

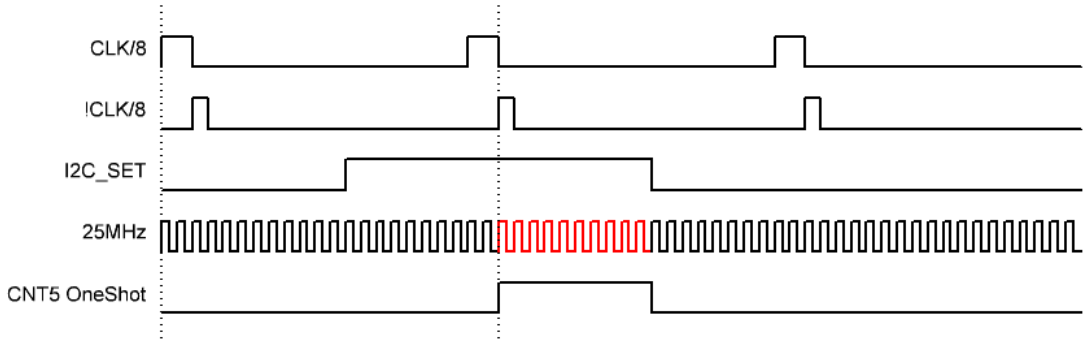

Figure 15. Loading the Duty Cycle with I²C (frequencies are not to scale)

5.4 Tachometer Reading

If desired, the user can read the tachometer value over I²C to track how quickly the fan is turning by reading the CNT2 value. CNT2 is incremented every time ACMP0H has a rising edge, and can be asynchronously reset with an I²C command. Note that this is an optional feature, and the threshold of ACMP0H will need to be tweaked according to the specifications of the particular fan being used.

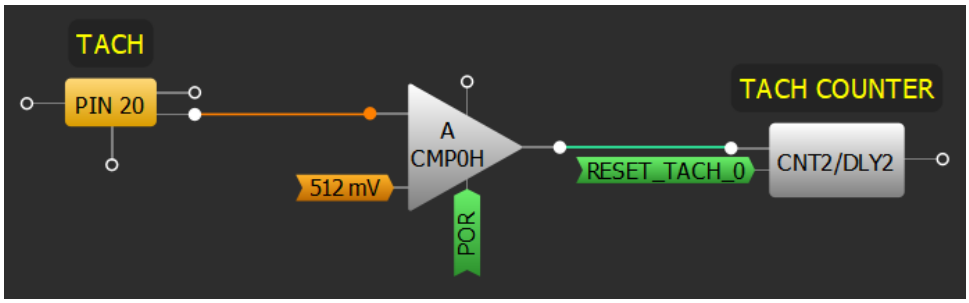

Figure 16. Tachometer Section

Table 2: Tachometer I²C Commands

Desired Function	I²C Command (with slave address of 0001)	Purpose
Read TACH COUNTER	[0x10 0x7E [0x11 r	Read the value in CNT2
Reset TACH COUNTER	[0x10 0x7A xxx1 xxxx]	Reset the value in CNT2
Un-reset TACH Counter	[0x10 0x7A xxx0 xxxx]	Release the reset on CNT2 so that it starts counting

6 External Circuit Design

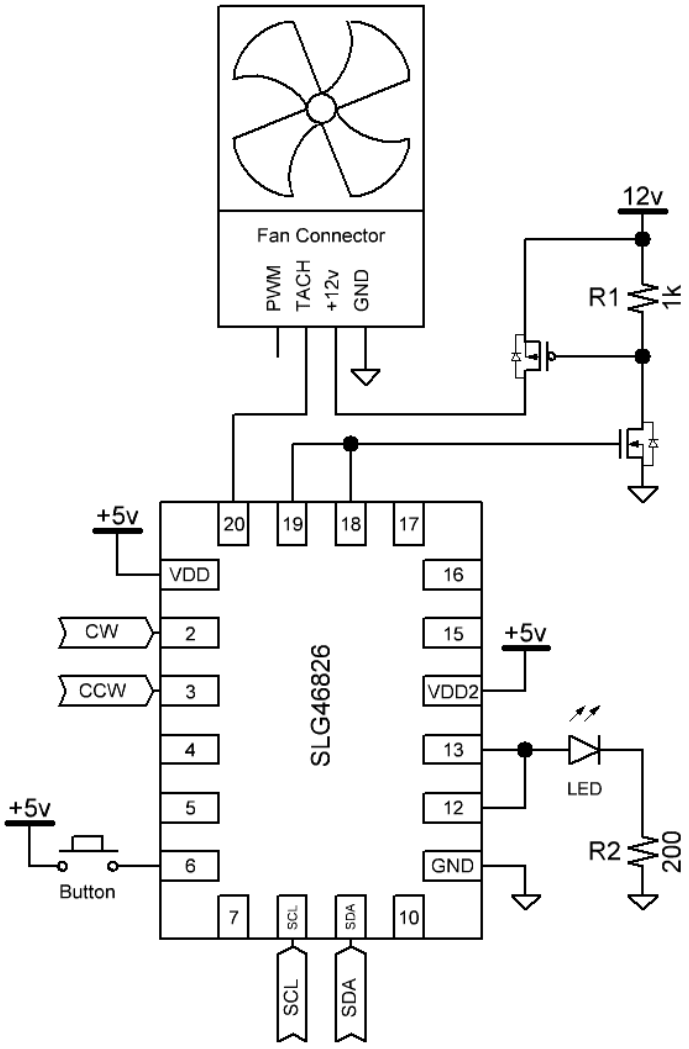

Figure 17. Fan Controller Block Diagram

The external circuit is fairly simple. There is a pushbutton connected to Pin6 of the GreenPAK to toggle whether this particular device is selected for rotary control, and an LED connected to Pin12 and Pin13 to indicate when the device is selected.

Since the fan runs off 12 V, a pair of FETs to control its switching is required. The GreenPAK's Pin18 and Pin19 drive an nFET. When the nFET is turned on, it pulls the gate of the pFET LOW, which connects the fan to +12 V. When the nFET is turned off, the gate of the PFET is pulled up by the 1 kΩ resistor, which disconnects the fan from +12 V.

7 PCB Design

To prototype the design a couple of PCBs were assembled. The PCB on the left is the "Fan Controller," which houses the rotary encoder, 12 V jack, SLG46108 GreenPAK, and connectors for the FT232H USB to I²C breakout board. The two PCBs on the right are "Fan Boards," which contain the SLG46826 GreenPAKs, pushbuttons, switches, LEDs, and fan headers.

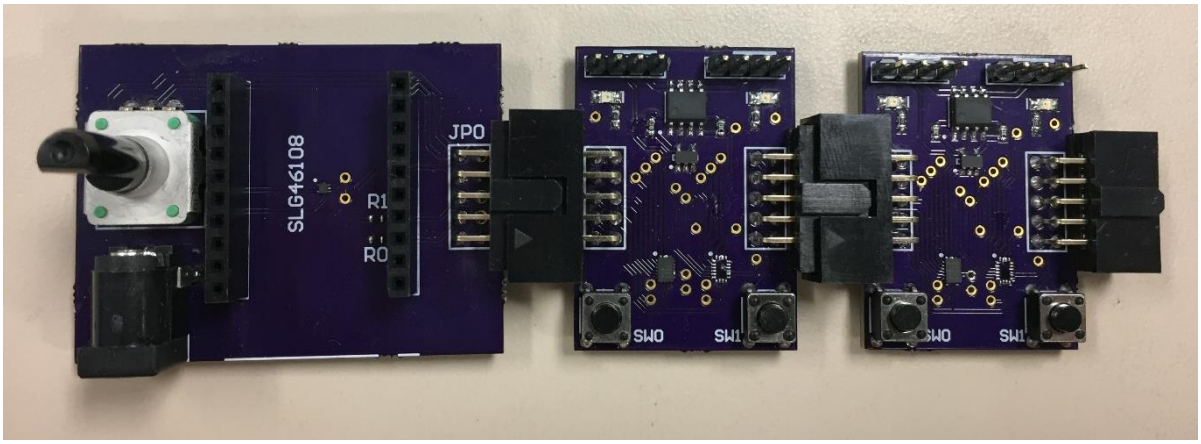

Figure 18. PCBs and Connectors

Each Fan Board has a shrouded male header on the left side and a female header on the right side so that they can be daisy-chained together. Each Fan Board can be populated with resources to independently control two fans.

8 C# Application

A C# application was written to interface with the Fan Boards through the FT232H USB-I²C bridge. This application can be used to adjust the frequency of each fan with I²C commands that are generated by the application.

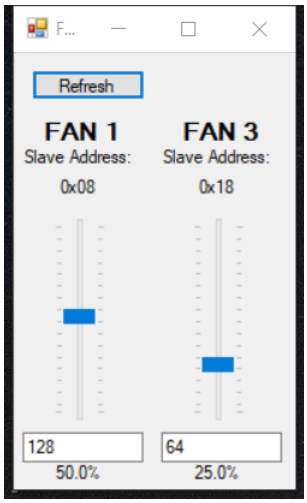

Figure 19. C# Application GUI

The application pings all 16 I²C addresses once per second and populates the GUI with the slave addresses that are present. In this example Fan 1 (slave address 0001) and Fan 3 (slave address 0011) connected to the board. Adjustments to the duty cycle of each fan individually can be made by moving the slider bar or by typing a value from 0-256 in the textbox underneath the slider bar.

9 Conclusions

Using this design it is possible to independently control up to 16 fans (since there are 16 possible I^2C slave addresses) either with a rotary encoder or with a C# application. It has been demonstrated how to generate a PWM signal with a pair of offset counters, and how to increase and decrease the duty cycle of that signal without rollover.

Revision History

Revision	Date	Description
1.0	03-Aug-2018	Initial Version

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers skilled in the art designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only for development of an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising out of your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Rev.1.0 Mar 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.