Speed Regulating PWM Fan Motor Controller

Author: Yu-Han Sun
Date: August 15, 2016

Introduction

The GreenPAK device can be used in high power applications by controlling a FET to drive a motor i.e. a PWM DC-DC buck converter. In this app note, we will show how to design a programmable closed-loop speed-regulating PWM fan controller using RPM feedback and I2C. The average percent error is less than 2.5%, and the min and max were less than 15%. These can be improved by increasing resolution at the cost of increased settling time.

Application

The three-wires of a 3-wire fan are VDD, GND and a tachometer OUT.

The motor which turns the fan is brushless, and uses about 0.2A at 5.0V. In this circuit, the motor speed of the fan is proportional to the input power, which is pulse modulated by a PFET to the power supply. The PFET gate is controlled by the GreenPAK PWM output, PIN18, as shown in Figure 1. The PWM output is determined by the Hall Effect RPM feedback.

A Hall Effect Sensor is positioned between two coils in the fan, and is the tachometer feedback out of the fan into the GreenPAK. Each edge of the Hall Effect Sensor signifies one rotation. By measuring the time between two rotations, we can calculate the speed at which the motor is rotating and adjust the power to the motor accordingly.

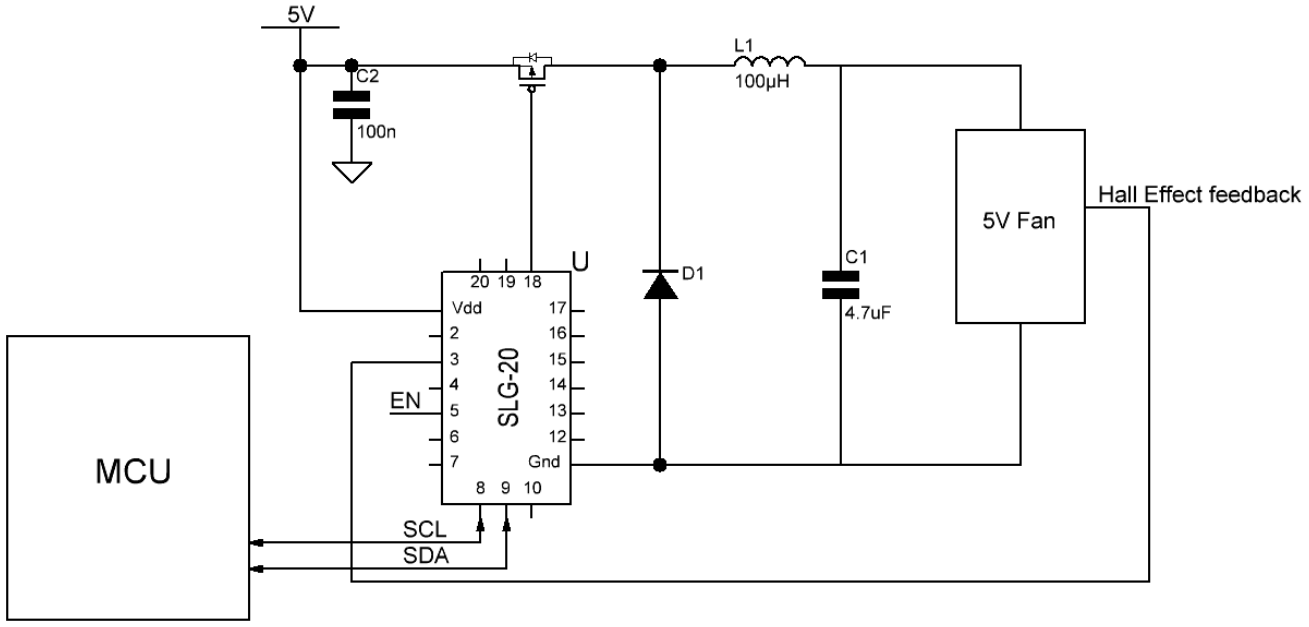


Figure 1. System Level View

As shown in Figure 1, the supply is at 5V, which is within GPAK's operating limit. The circuit below is a DC-DC buck converter using one FET on the high side, a clamping diode and a LC filter.

GPAK Design, the ASM states

This design uses all eight states of the ASM. The states are labeled as in Figure 2. The initial state is the Reset state. In the Reset state, the PWM output is forced low.

Due to inertia, a fan does not start moving until we overcome static friction. That minimum duty cycle is typically 30%.

When the ASM is enabled, we transition to an 'Overdrive' state, where the output is allowed to be driven at maximum power, 100% duty cycle, for 3 seconds. This initial kick is meant to overcome the static friction, allowing the motor to start spinning.

From there, we arbitrarily choose to enter the PWM Up state. If the RPM is too high, the GreenPAK will self-correct and move to PWM Down. The state machine continues to toggle between PWM Down and PWM Up. As we reach a steady state, the toggling will become more and more stable.

In the event that the RPM desired is too low or too high, the GreenPAK will eventually reach either 100% or 0% duty cycle. To prevent the PWM from wrapping around, the output is gated at a 'Stop High' or 'Stop Low' state. In these states, the PWM is forced high and low.

'Buffer Low' and 'Buffer High' are intermediary states to prevent unwanted ASM state transitions. The state machine uses edge detectors for 'Stop Low' and 'Stop High'.

If the PWM should reverse directions at this point, the very first period will incur an edge. If there was no Buffers state, we would look from one Stop state to another Stop state immediately. Therefore, the Buffer state is an intermediary state to prevent looping.

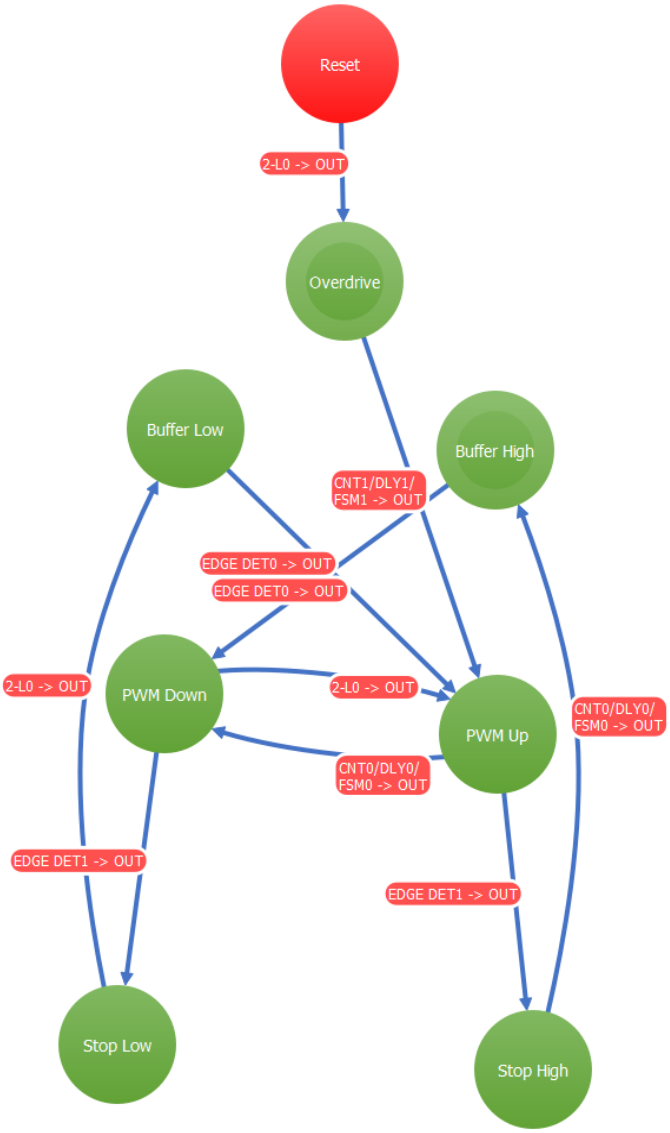


Figure 2. ASM State Machine

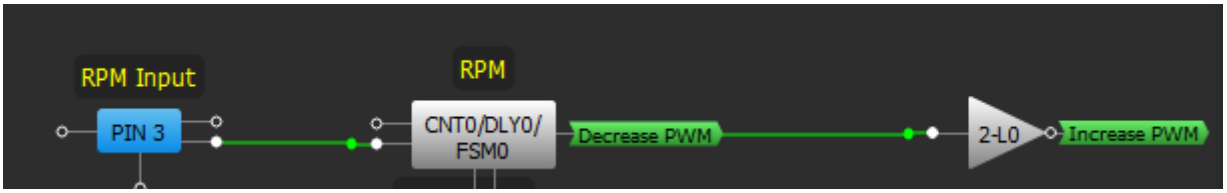


Figure 3. RPM Input to Frequency Detector FSM0

GPAK Design, the ASM inputs

The main inputs to the ASM are 'Decrease PWM' and 'Increase PWM', two inverted signals. They signify when to increase or decrease the PWM width as shown in Figure 4. 'Increase PWM' transitions the ASM from PWM Down to PWM Up and 'Decrease PWM' transitions the ASM from PWM Up to PWM Down.

FSM0 is the RPM frequency counter. In frequency detect mode, the time between two input edges is measured and compared to the counter data. If the length is longer, the speed is too slow. If the length is shorter, the speed is too fast. The output of FSM0 regulates the DC-DC buck to pump more or less power to the fan circuit.

FSM0 is configured as a both edge frequency detector because the tachometer used in this application, US1881, is a latching hall effect sensor. Meaning each rotation toggles the tachometer output. Therefore each edge, whether falling or rising, represents one 360 degree rotation. For this example, the target RPM has a period of 40ms.

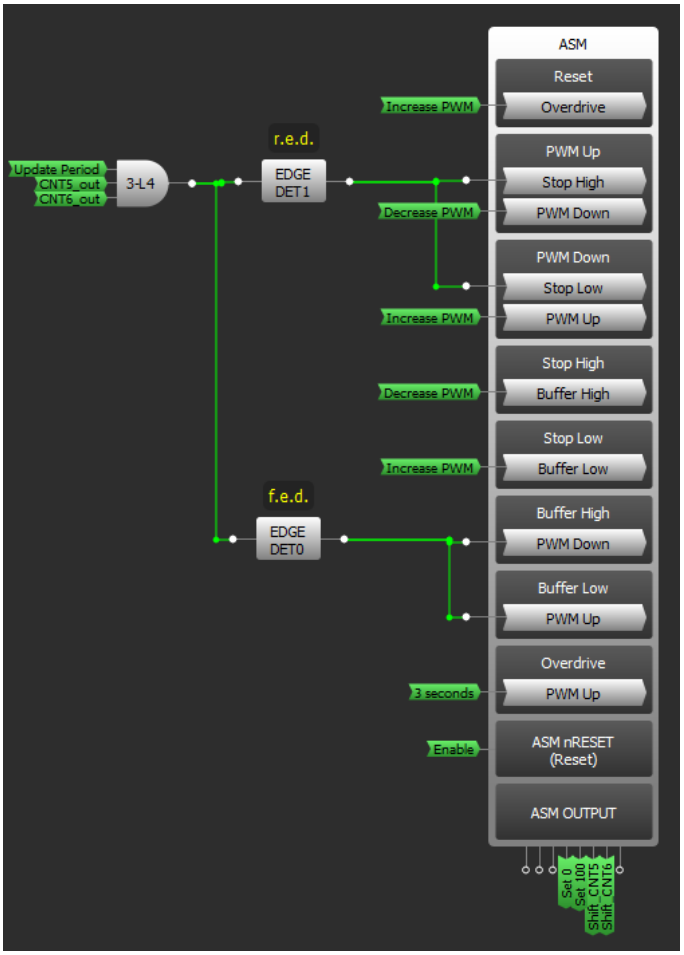


Figure 4. ASM Inputs

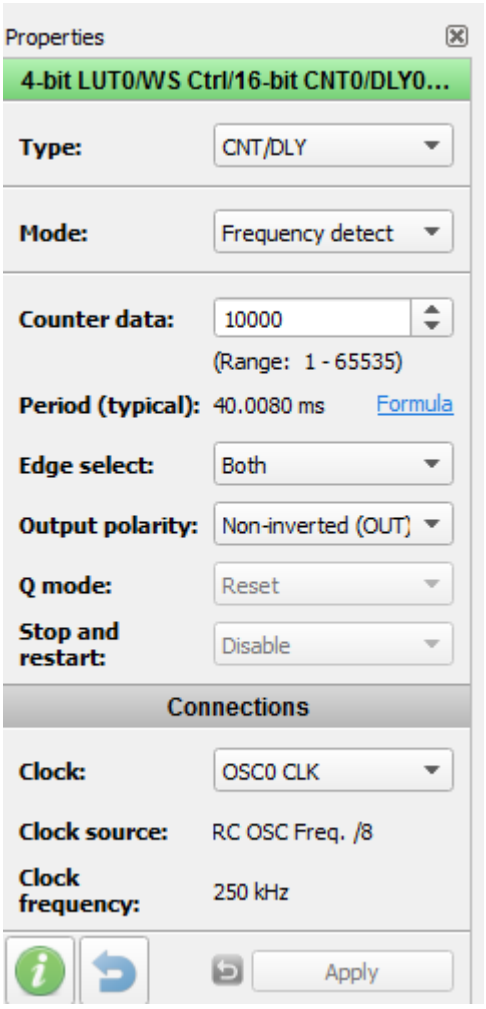


Figure 5. FSM0 Configuration

The input to the edge detectors is an AND of the Set and Reset signals. By checking for an overlap, we are able to know when the duty cycle has reached 100% or 0%. The edge detect outputs then transition the ASM to 'Stop High' and 'Stop Low'.

Since the very first PWM output Set and Reset signals overlap by default, we include intermediary states to prevent the states from transitioning incorrectly.

Shifting Set and Reset counters

The PWM duty cycle is controlled by a Set and Reset signal which come from the output of the CNT5/DLY5 and CNT6/DLY6. The settings are identical as seen in Figure 6.

In order to increase and decrease the PWM, we shift the Set and Reset signals relative to each other to achieve a shorter or longer duty cycle. Refer to Figure 8 timing diagram. Depending on the current state, one of two CNT/DLY blocks will gain an extra clock, shifting the output to the left, earlier in time. If Set shifts, the PWM increases. If Reset shifts, the PWM decreases. This is controlled by the signals 'Reset' and 'Set' as seen in Figure 7.

The extra clock occurs every 50 periods, allowing time for the mechanical world to catch up with the electrical signals. This is the update period, set by CNT4/DLY4. Every update period, the pipe delay and 2-bit LUT2 generates a pulse into the 3-bit LUT0 and 3-bit LUT1 IN0 inputs. If XORed with the normal clock source, we will generate an extra pulse.

GPAK Design, the ASM outputs

Only four ASM outputs are used. 'Shift_CNT6' and 'Shift_CNT5' go into the SR Counter logic and control the shifting. They are used to select which CNT gets an extra clock, representing the current direction that the PWM is going in.

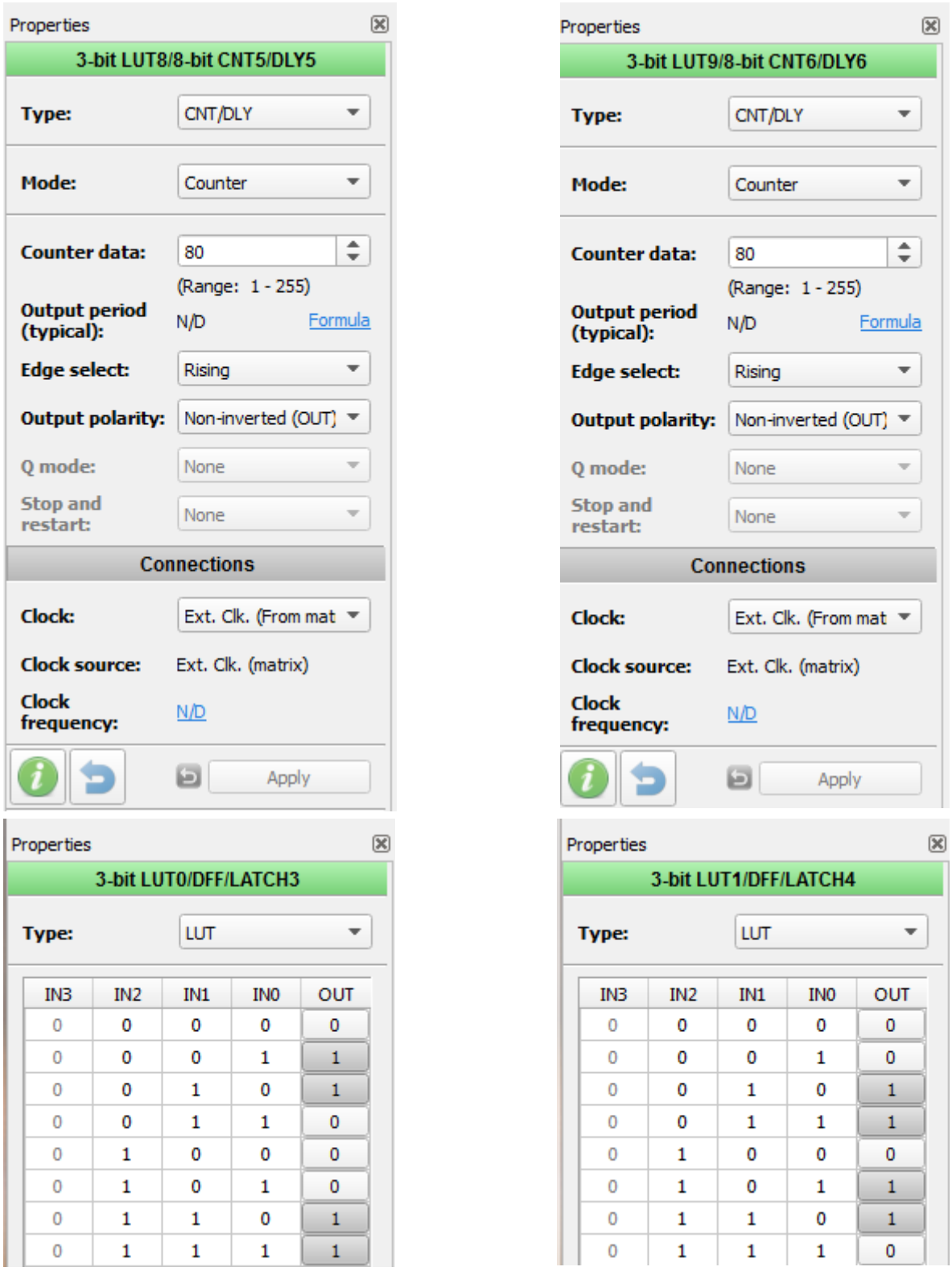


Figure 6. Set and Reset for PWM output

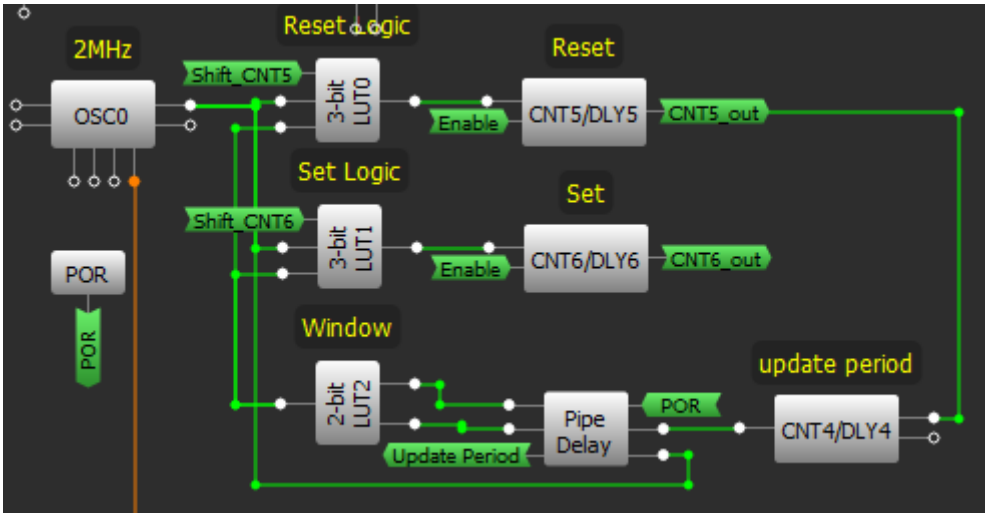


Figure 7. Set and Reset for PWM output

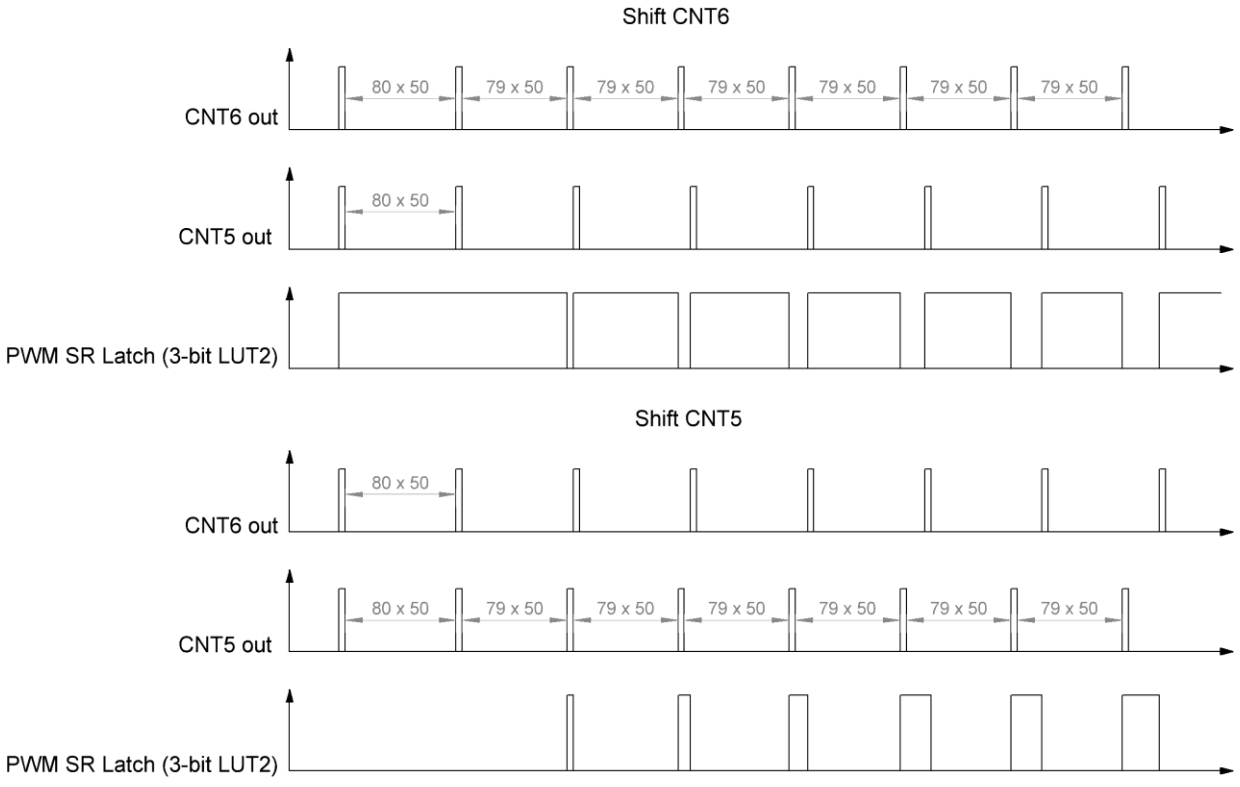
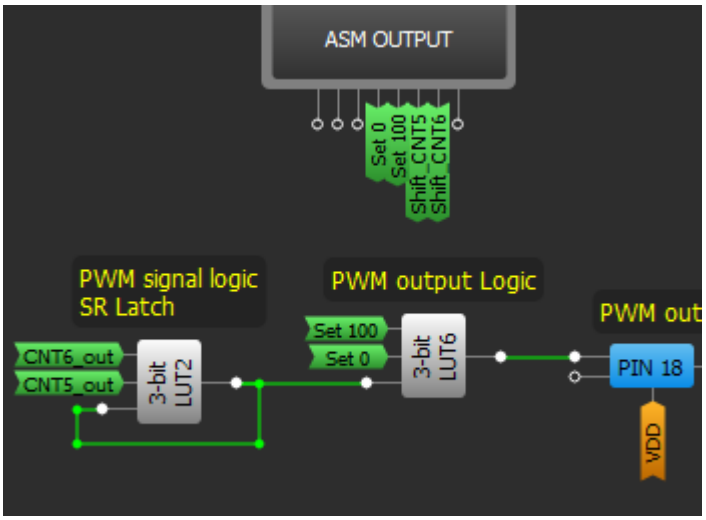


Figure 8. Set and Reset for PWM output

RAM

State name	Connection Matrix Output RAM							
	OUT7	OUT6	OUT5	Stop 100	Stop 0	Shift_C...	Shift_C...	Initial
Reset	0	0	0	0	1	1	0	0
PWM Up	0	0	0	0	0	1	1	0
PWM Down	0	0	0	0	0	0	0	0
Stop High	0	0	0	1	0	1	0	0
Stop Low	0	0	0	0	1	1	0	0
Buffer Hig..	0	0	0	1	0	0	0	0
Buffer Low	0	0	0	0	1	1	1	0
Overdrive	0	0	0	1	0	0	0	1



3-bit LUT2/DFF/LATCH5

Type: LUT

IN3	IN2	IN1	IN0	OUT
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0

3-bit LUT6/8-bit CNT3/DLY3

Type: LUT

IN3	IN2	IN1	IN0	OUT
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1

Figure 9. ASM Outputs and the Output configuration

As seen in the ASM Output Table of Figure 9, Reset and Set are both High if we are counting up and both Low when counting down. To stop extra clock counting, Reset and Set are opposites. Reset is high and Set is Low. This makes sure that they both choose the regular clock source.

'Force 0' and 'Force 100' indicate when the PWM logic has reached 100% and 0% and masks out the SR Latch. 'Force 0 forces the PFET closed for Stop Low, Buffer Low, and Reset states. 'Force 100 forces the PFET open for Stop High, Buffer High and Overdrive states.

3-bit LUT2 is the SR Latch. If CNT6_out is logic 1, the latch is set. If CNT5_out is logic 1, the Latch is reset. 3-bit LUT6 gates the SR Latch. If Set High is high, the output is forced high. If Set low is high, the output is forced low. Both signals will never be high at the same time. If both signals are low, use the inverse of the 3-bit LUT2.

I2C

To change the desired RPM, use I2C to write the counter value of FSM0. The word address is located at address 0xC5 and 0xC6.

Refer to the example in Table 1 below. If we want to write to FSM0 the counter value 0xDD, the command would be:

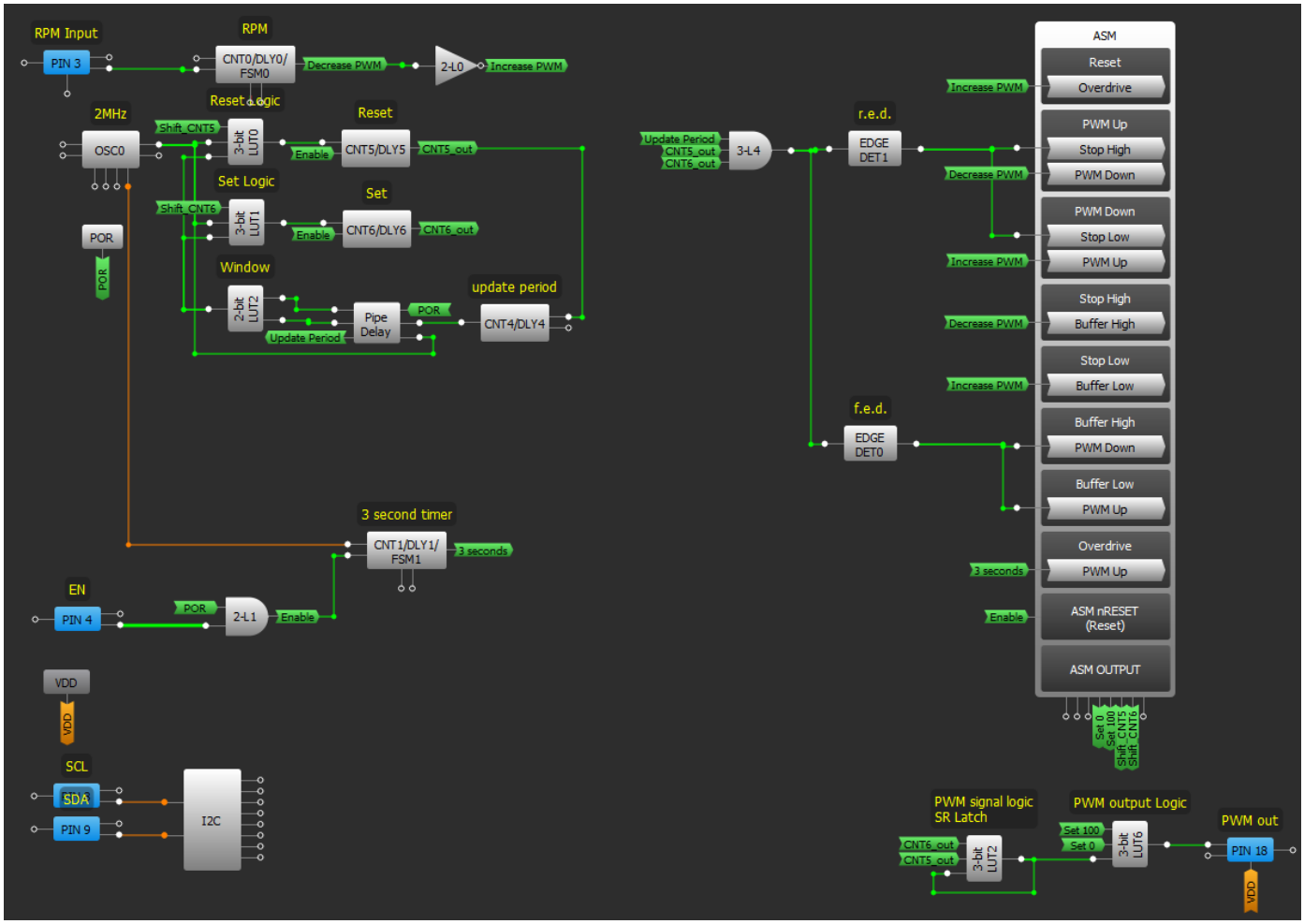


Figure 10. GP Design

[0xSA 0xC5 0xE0 0x2E] for example 1

[0xSA 0xC5 0x40 0x1F] for example 2

Where SA is the slave address.

To convert from RPM to counter data, use the equation below:

$$counter\ data = \frac{OSC}{RPM}$$

The counter data depends on the GreenPAK internal OSC. In this design, the OSC should ideally be 2MHz/8 = 250kHz. To get an RPM frequency of 25Hz, the period must be 40ms, which corresponds to a counter data of 9998.

Because the Actual OSC was measured to be 253kHz, we end up with 25.3Hz instead of 25Hz.

The percent error was 2.4%, 1.0% and .5% from the above three examples. The longer the device is running, the better the average percent error. Some error is due to OSC trim, which in our case was measured to be 253kHz instead of 250kHz. The PWM will take some time to ramp up or down until it reaches a steady state. After reaching steady state, the state machine will flip flop between the 'PWM Up' and 'PWM Down' which causes the PWM output to jitter around the average.

Word Address	Current RPM/FSM0 counter value	Example 1	Example 2
0xC5 0xC6	0x10 0x27	0xE0 0x2E	0x40 0x1F
Decimal	9998	11998	7998
Period	40ms	48ms	32ms
Frequency/ RPM	**25 Hz**	**20.8 Hz**	**31.25Hz**
Actual Frequency	**24.4 Hz**	**21.0 Hz**	**31.4 Hz**
Min	**23.2 Hz**	**18.9 Hz**	**28.6 Hz**
Max	**27.8 Hz**	**22.8 Hz**	**34.4 Hz**
St. Dev.	**478 mHz**	**385 mHz**	**520 mHz**

Table 1. RPM I2C write example

This causes the min and max frequency to jump around the average depending on the PWM resolution. In the above examples, the percent error of the mins and max frequencies were at worst, 15%. To decrease this error, we can increase the CNT5 and CNT6 counter value such that each increment would be a smaller PWM width, allowing for better frequency control but slower settling time.

Functionality Waveforms

The function below was taken after many seconds, when the PWM output has reached a steady state.

In steady state, the state machine is switching from PWM Up to PWM Down rhythmically.

We can see that the RPM input has a frequency 11.4Hz, or 88ms. However, since the Hall effect sensor is a latching device, every edge is being interpreted as a revolution. FSM0 is configured to detect both edges. Therefore, the actual RPM frequency is twice at 22.8Hz, or 45ms. This is right around our 40ms target, which was set by the counter value in FSM0.

Channel 1 (yellow) – PIN#18 (PWM Out)

Channel 2 (light blue) – PIN#3 (RPM)

Channel 3 (magenta) – Fan Input Voltage

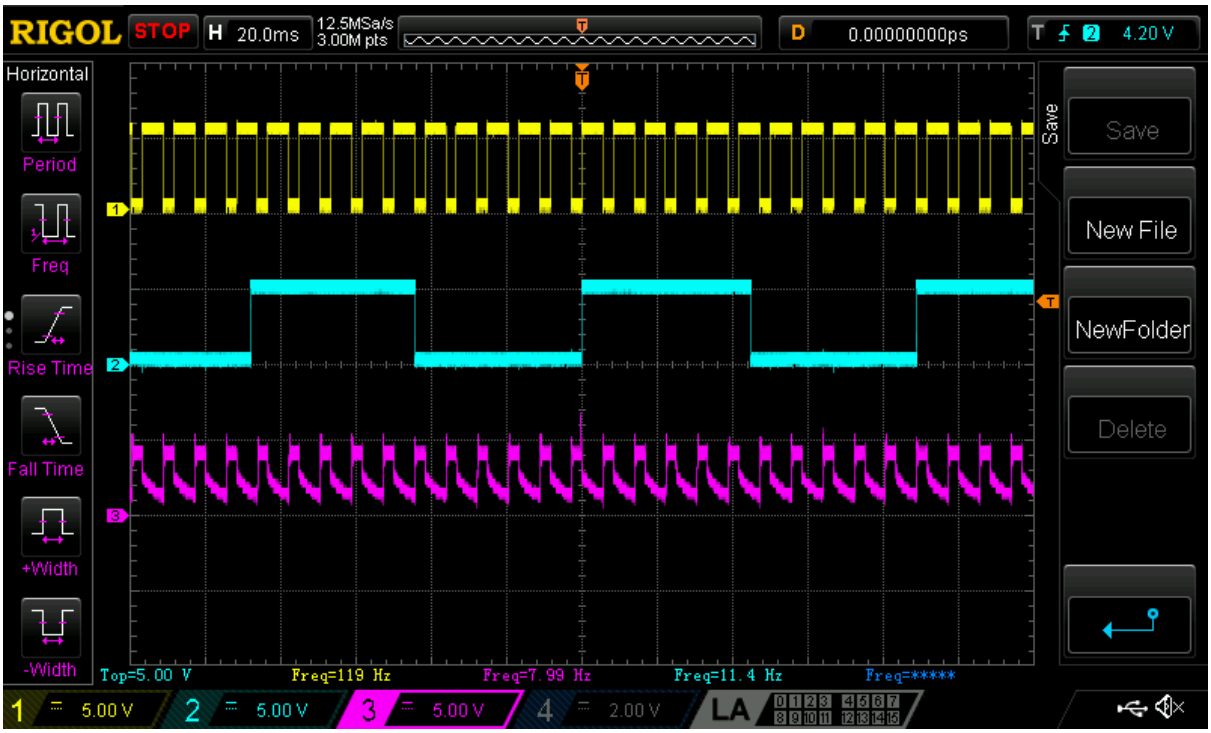


Figure 11. PWM Fan waveform

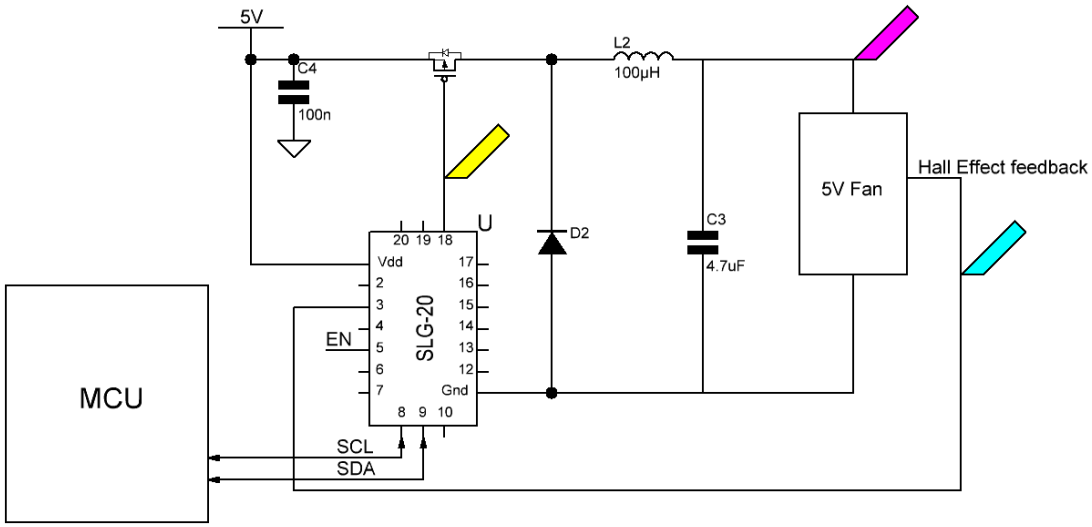


Figure 12. PWM Fan circuit

Figure 13 shows the functionality when the FSM0 counter data is set to 12000. We can see the Hall effect frequency is 9.42Hz, which implies an RPM of 18.84 Hz.

Figure 14 shows the functionality when the FSM0 counter data is set to 8000. We can see the Hall effect frequency is 14.8Hz, which implies an RPM of 29.6 Hz.

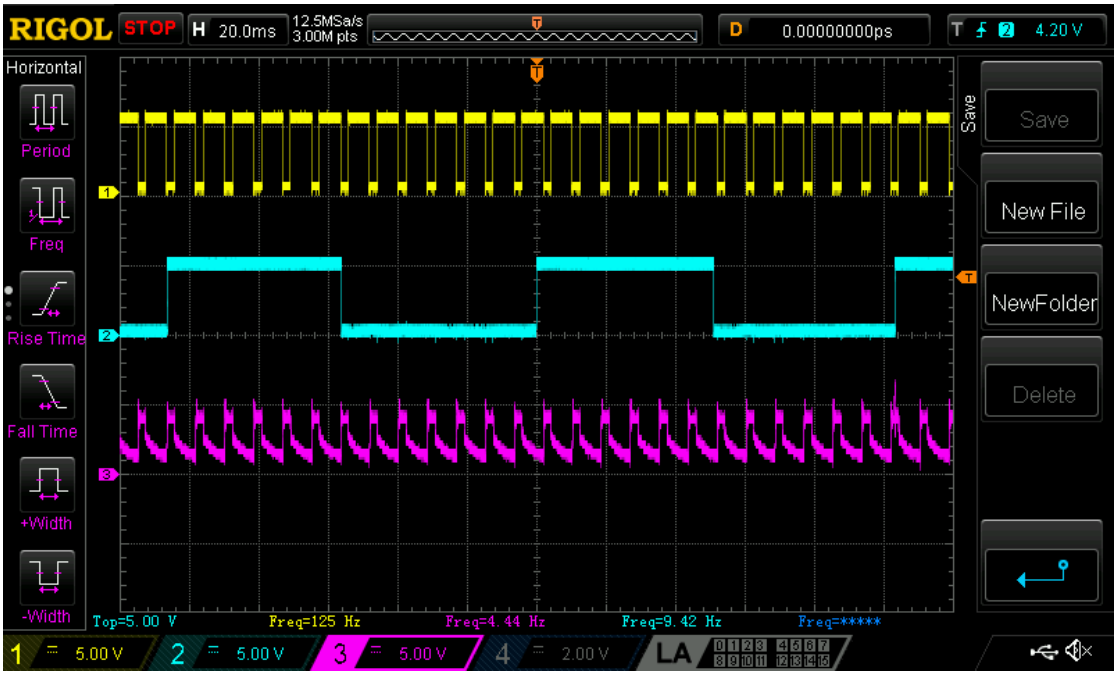


Figure 13. PWM Fan waveform Example 1

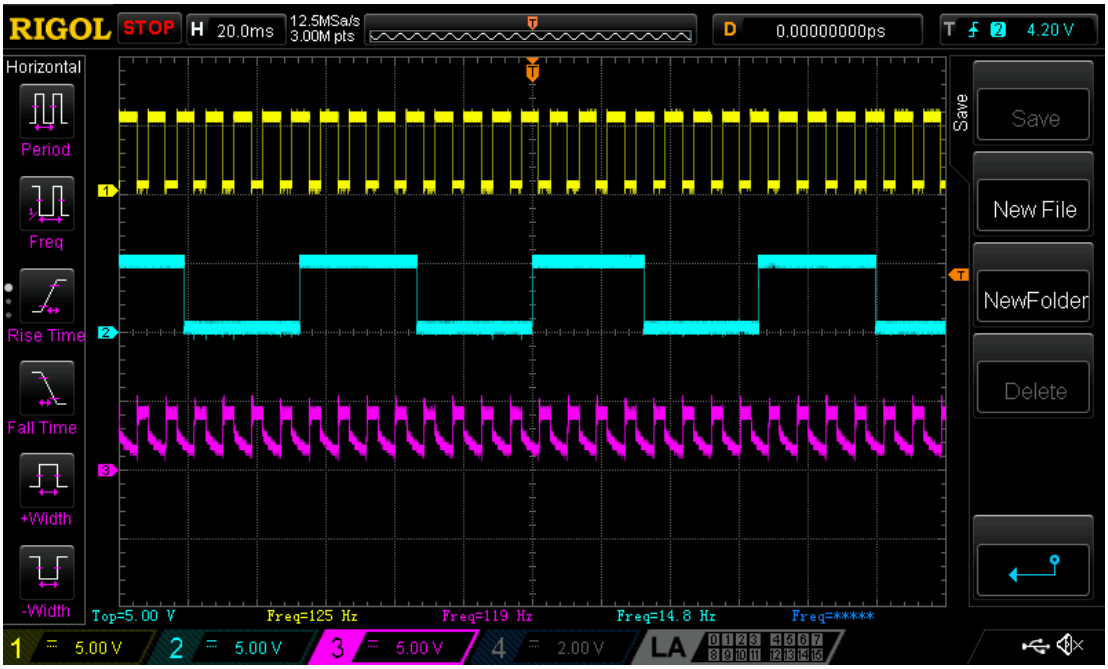


Figure 14. PWM Fan waveform Example 2

Circuit

Below is a breadboard prototype of the PWM Fan. The breadboard on the left has the inductor, PFET, diode and capacitors. The GPAK IC is in the development tool evaluation board.

The breadboard on the right connects the GPAK external connectors to the Hall Effect Sensor, which has to be powered continuously.

In the first image below, the fan is stopped because the PWM circuit is Disabled. In the second image, the Fan is running because the PWM circuit is Enabled.

Conclusion

The Speed Regulating PWM Fan Motor Controller is an example of how the ASM can be used in a feedback loop to control an external motor. We use the frequency detector function to easily create a watchdog on the Hall effect input. Then, we use I2C to change the reference frequency. The entire digital control logic is implemented in the PAK device such that, due to use of the ASM, it can be easily tweaked.

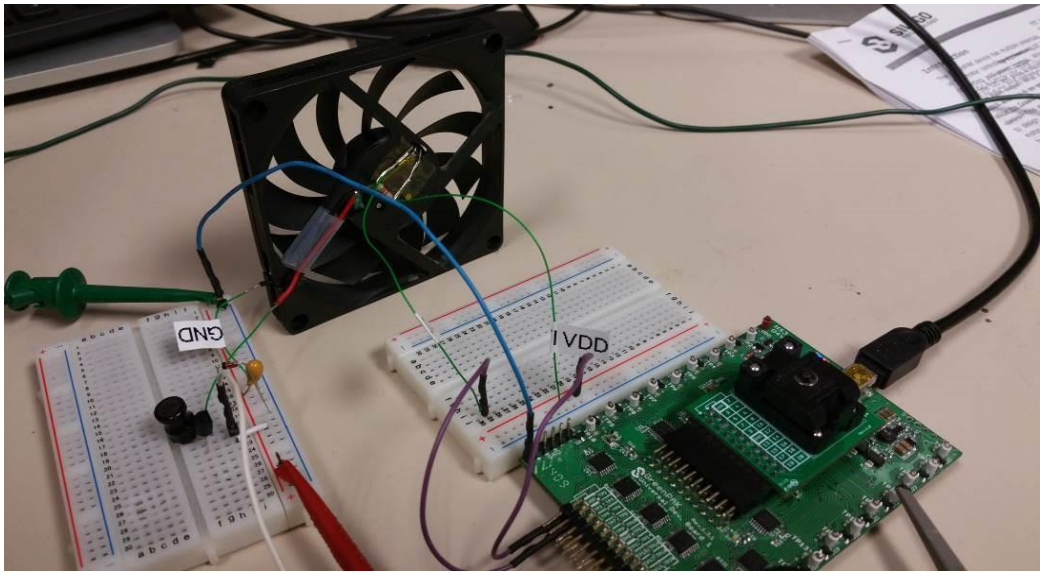


Figure 15. Fan Disabled

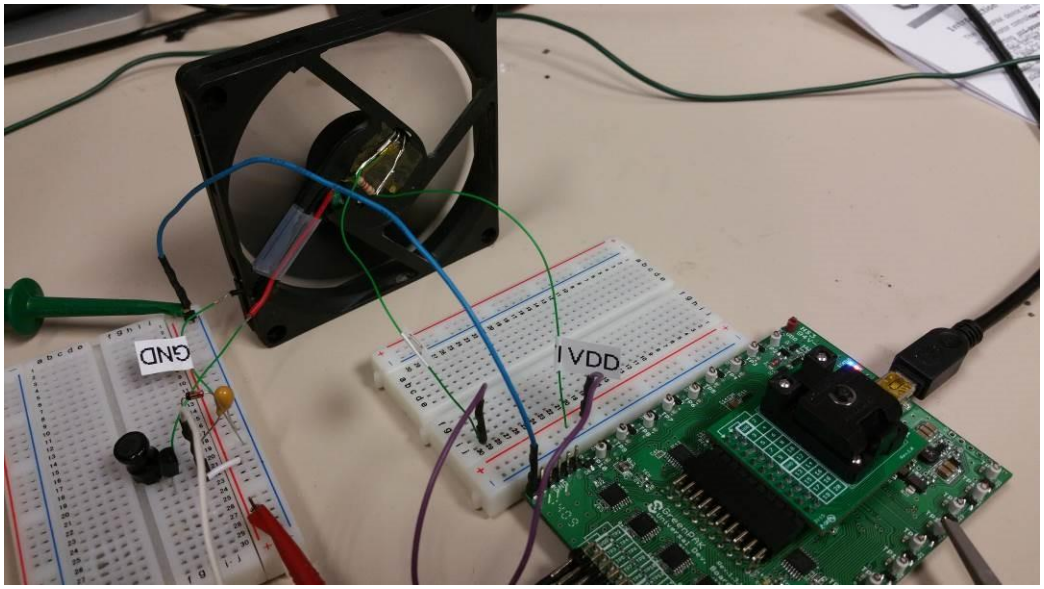


Figure 16. Fan Enabled

About the Author

Name: Yu-Han Sun

Background: Yu-Han is currently an Applications Engineer at Silego Technologies working primarily with the GreenPAK products lines.

Contact: **appnotes@silego.com**

Document History

Document Title: Speed Regulating PWM Fan Motor Controller

Document Number: AN-1117

Revision	Orig. of Change	Submission Date	Description of Change
A	Yu-Han Sun	08/15/2016	New application note

Worldwide Sales and Design Support

Silego Technology maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the sales person closest to you, visit us at **Sales Representatives and Distributors.**

About Silego Technology

Silego Technology, Inc. is a fabless semiconductor company headquartered in Santa Clara, California, with operations in Taiwan, and additional design/technology centers in China, Korea and Ukraine.

SILEGO TECHNOLOGY

Silego Technology Inc.
1515 Wyatt Drive
Santa Clara, CA 95054

Phone: 408-327-8800
Fax: 408-988-3800
Website: www.silego.com