

Motor Tachometer Speed Calculation Using Hardware Timer Capture Feature

AN2087

Author: Arnold Motley

Associated Project: Yes

Associated Part Family: CY8C24x23A, CY8C27x43, CY8C29x66

[GET FREE SAMPLES HERE](#)

Software Version: PSoC Designer™ 4.4

Associated Application Notes: None

Application Note Abstract

Motor speed sensing is a necessary and critical part of μ processor based fan control. This application note explains how to utilize the unique PSoC® architecture to capture, filter, and store the period of a fan tachometer signal.

Introduction

The PSoC MCU captures and measures tachometer (tach) signals from fan and other types of motors. This is critical when performing automatic speed control and speed error detection and correction. The PSoC digital blocks provide a hardware timer capture, which is a special feature used for the accurate measurement of motor tach signals.

Motor Background

Motors come in a wide variety and have different features and capabilities. For this design, a voltage controlled variable speed motor with a tach output feature is selected. The tach output comes from an open drain connection, so a 1 k Ω resistor to V_{CC} is necessary.

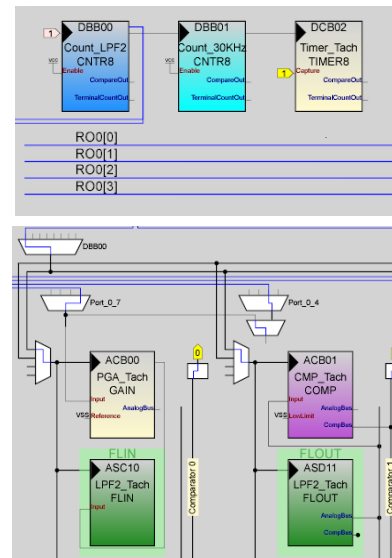
The motor has 2 poles, which means that two square wave pulses (periods) are generated for one mechanical revolution of the motor. The tach period has a duty cycle of 50%. The motor statistics follow:

- Operating Voltage: 7vdc - 12vdc
- Operating Speed: 3900 RPM at 7 vdc
6800 RPM at 12 vdc

Hardware Module Setup

Converting the analog tach signal from the fan into a digital signal that a timer can use requires several steps. Figure 1 shows the hardware blocks used to implement the design. First, a Programmable Gain Amplifier (PGA) is used in the PSoC to provide a connection block. From the PGA the signal goes to a Low Pass Filter (LPF) to remove the high frequency components that are common on motor tach signal lines.

Figure 1. Tachometer User Module Placement



The LPF is configured for 1 kHz cutoff frequency. The 600 kHz column clock required for the LPF is generated by the Count_LPF2 user module. The output of this counter also feeds the Count_30 kHz counter, which generates a 30 kHz clock for the capture timer.

The last component in the analog stage of the design is a comparator. The two purposes of a comparator follow:

1. It "conditions" the tach signal when it leaves the filter. This is done by decreasing the leading edge rise time of the tach signal. In this design, the tach output of the comparator has a rise time of about 6-7 μ s.
2. The comparator also provides the hardware connection to the timer. The comparator output bus feature is enabled in the comparator block for this purpose.

The hardware capture feature of the timer works as follows:

When a positive tach pulse is sensed on the capture line of the timer, an interrupt is generated. Upon entering the timer Interrupt Service Routine (ISR), the timer is stopped and then a count variable "PulseCount" is checked to see if this is the first of two pulses. If it is the first pulse, the counter variable is incremented, the timer period is loaded with a value of FFh, and the timer is restarted in preparation for the next pulse.

If it is the second pulse, the timer compare register is read and the value (the tach period) is moved to the "PulseData" variable. A value of 1 is added to the "PulseData" variable in case an overflow condition is detected. In an overflow condition, the variable returned from the timer compare register is FFh and the compare test indicates that the overflow did not occur. Adding 1 to "PulseData" makes the value 0, and now the test correctly indicates an overflow condition.

The compare value in the ISR is set to a value of 20h, which correlates to a timer period window of 7.5 ms. This is calculated as follows:

- Upper time limit = (input Clk period) * (Timer period count - value set in ISR)
- Upper time limit = (1/30 kHz) * (256 - 32)
- Upper time limit = 7.5 ms

According to the previous calculations, the window for capturing tach pulses is 7.5 ms with a resolution of 33 μ s or 1 over the timer clock frequency of 30 kHz.

Setting up the timer correctly is very important to ensure that the pulses are captured accurately and an overflow condition is detected properly. In the design, the hardware timer compare value is also set to 20h. At a setting of 20h, PSoC indicates an overflow condition at a fan speed of 3720 RPM. This trip point is 4.6% below the minimum operating fan speed of 3900 RPM at 7vdc. The trip point setting is variable and may be adjusted by changing the compare value in the timer module and the ISR.

When this state is sensed, a flag may be set to indicate a low fan speed condition. This allows the design to add support for a complete fan failure and slow fan conditions. This same capability allows variable speed error correction. For example, if a slow fan condition is detected, instead of just turning on the backup fan at full bore, an algorithm is added to the firmware to allow the variable speed operation of the backup fan.

Firmware Support

If "PulseData" is sampled less than or equal to 20h, this indicates that an overflow condition has occurred. Now, in the ISR an overflow variable "Overflow" is set to 1 and the timer is started. A variable "NewData" is set to 1 to indicate that a new tach signal pair is received. The "PulseCount" variable is also reset to 0 and a `reti` is done. This code detects a low fan speed condition.

Note This code does not detect the complete failure of the fan because there would be no tach signal to generate the

capture interrupt. Moreover, if the time between the tach signals exceeds the total period of the timer, the firmware cannot detect this condition. For such conditions, a 16-bit timer is used to increase the measurement range or a timeout mechanism may be implemented by using another timer.

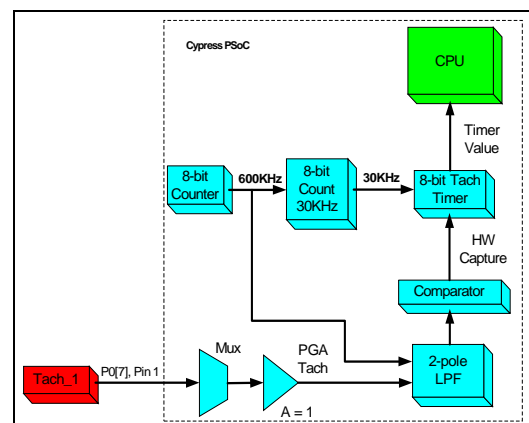
The main body of code sits in a loop until the variable "NewData" is set. When "NewData" is set, the code goes into a switch statement to check if the data is valid or not. If the data is valid, the variable "PulseWidth" is set with the variable "PulseData." "PulseWidth" holds the tach data until a new data point is captured.

If the data is invalid then "PulseWidth" is set with a value of 0. "Overflow" and "NewData" are also set to 0 in preparation for the next capture. The data stored in "PulseWidth" could be sent to a monitoring device for further processing.

Block Diagram

The block diagram in Figure 2 shows the PSoC hardware user modules that are used in the design. Pin 1 in the following figure is configured as a standard analog input pin. The tach signal from the fan is a square-wave pulse train from 0 – 5 vdc, so the gain on the PGA is set at unity ($A=1$).

Figure 2. Tachometer Block Placement



The 2 pole LPF has an $f_c = 1$ kHz, and the comparator switching level is set at 2.5 vdc. This level is chosen to limit noise affects, yet it allows enough margin for proper design operation. With the PSoC A_Mux module, a multichannel sensing scheme is easily implemented.

Summary

The PSoC μ controller is used to control motor operation. It incorporates many functions normally done with external hardware, including the following:

- Speed Sensing
- Error Detection
- Correction

This application note demonstrates straightforward PSoC device implementation of capturing and measuring tachometer signals.

About the Author

Name: Arnold Motley

Title: Cypress Customer Design Center Engineer

Contact: iam@cypress.com

In March of 2007, Cypress recataloged all of its Application Notes using a new documentation number and revision code. This new documentation number and revision code (001-xxxxx, beginning with rev. **), located in the footer of the document, will be used in all subsequent revisions.

PSoC is a registered trademark of Cypress Semiconductor Corp. "Programmable System-on-Chip," PSoC Designer and PSoC Express are trademarks of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
Phone: 408-943-2600
Fax: 408-943-4730
<http://www.cypress.com/>

© Cypress Semiconductor Corporation, 2002-2007. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

This Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.