TelCom Semiconductor, Inc.

# APPLICATION NOTE 56

# USING THE TC643 FOR ON/OFF FAN CONTROL AND FAN FAULT MONITORING

By Pat Maresca

## INTRODUCTION

Intelligent fan control extends fan life, lowers acoustic noise and lowers average fan current. Fan fault monitoring allows, in many cases, the use of less expensive (less reliable) fans. However, many fan control/monitor schemes involve the use of multiple discrete devices, increasing system cost and complexity. This application note discusses a control algorithm to implement intelligent fan on/off control with fan fault detection using a TC643 BDC Fan Driver/ Monitor and control resources provided by the system microcontroller. The algorithm described will work with the most basic of 4-bit (or better) microcontrollers, and is parceled in a small subroutine, callable from the user's system software.

The system is built around TelCom's TC643, a brushless DC fan motor driver with on-board RPM and over-current detection. The on-board driver is capable of driving any 2-wire BDC fan with a maximum voltage of 15V and a maximum operating current of 300 mA. The fan driver is turned on or off by applying a logic high or low to the DRV input. While this application note covers simple on/off control, the DRV input also can be driven by a variable duty cycle (30 Hz to 60 Hz) PWM signal to generate any number of intermediate fan speeds, or to provide continuously variable fan speed control. The TC643 has two outputs: RPM and $\overline{\text{ILMT}}$. The RPM output pulses each time the fan motor commutates. (For example, a four pole fan will cause the RPM output to pulse four times each revolution.) $\overline{\text{ILMT}}$ is driven active when the current passing through the on-board driver exceeds 500 mA, indicating a shorted fan.

## PROCESSOR-BASED INTELLIGENT FAN CONTROL

### Hardware Connections

Figure 1 shows the required hardware configuration for interfacing the TC643 to the microcontroller. Only a pull-up resistor from $\overline{\text{ILMT}}$ to the microcontroller $V_{DD}$, and a 2 nF capacitor are required. As shown, the microcontroller must have three port pins available: an output pin connected to the DRV input of the TC643 (fan on/off control bit); and two input pins, one to monitor the RPM signal from the TC643, the other to monitor the over-current ($\overline{\text{ILMT}}$) output from the TC643. (If so desired, either $\overline{\text{ILMT}}$ or RPM can be connected to interrupt inputs from the microcontroller to eliminate polling. For the purpose of this application note, only polled operation will be discussed.)

### Fan Control Subroutine

The software algorithm described here (which will be referred to as *the Fan Control Subroutine*) requires the

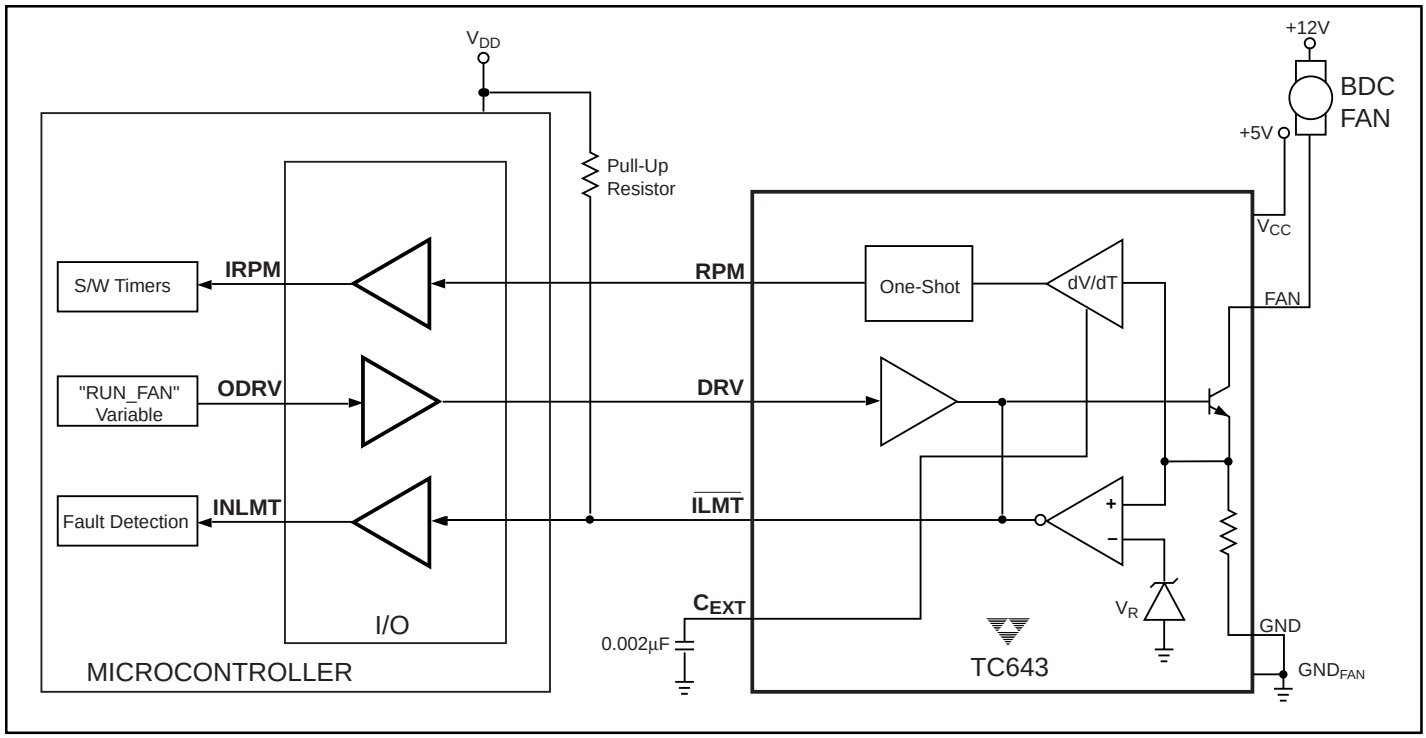


**Figure 1. TC643 Hardware Configuration**

creation of four control/status bits. These bits should be contained in a single processor register or scratchpad memory location, so the fan status and control can be manipulated by the user's system software in a single access. These bits are described below:

- **RUN_FAN**: The user's system software sets this bit to 1 to operate the fan, or to 0 to stop the fan. The Fan Control Subroutine uses this bit to both gate fan operation, and determine if a fan fault condition exists.

- **FAN_FAULT**: The Fan Control Subroutine sets this bit to 1 when RUN_FAN = 1 and RPM pulses from the TC643 are not detected, indicating a stuck or open fan.

- **FAN_OVR_CURRENT**: The Fan Control Subroutine sets this bit to 1 when the $\overline{\text{ILMT}}$ output from the TC643 is driven low, indicating a short-circuited fan.

- **EDGE**: The Fan Control Subroutine uses this bit to ensure that both the rising and falling edges of the incoming RPM pulse have been checked. This bit is cleared upon entry to the Fan Control Subroutine, then set to 1 when the first edge transition is detected. An RPM pulse is recognized valid only when the second (opposite polarity) edge transition is detected *and* EDGE = 1.

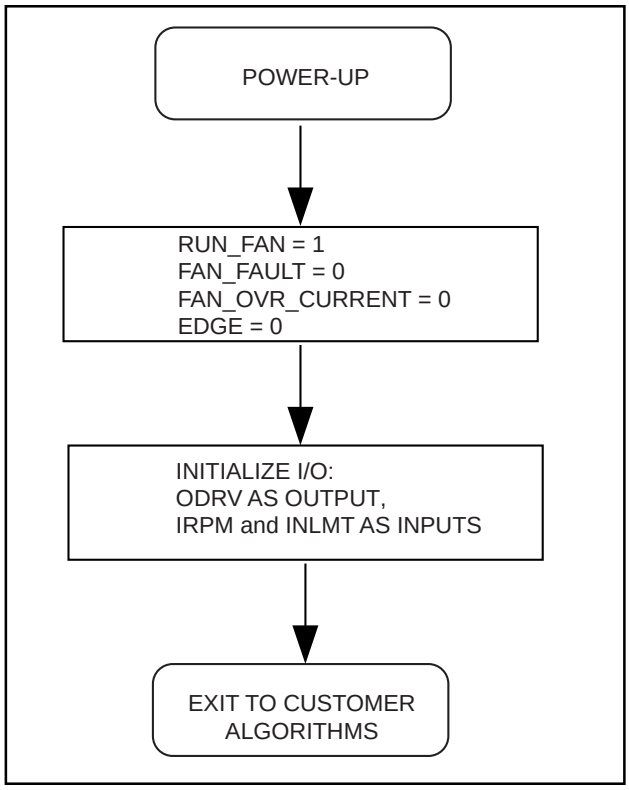


**Figure 2: Power-Up Initialization Flowchart**

It is necessary to initialize these bits as part of the microcontroller's power-up software routine, as shown in Figure 2. Here, RUN_FAN is set to 1 (fan default state = running); FAN_FAULT is reset to 0 (no fault condition present); FAN_OVR_CURRENT = 0 (fan current normal); and EDGE = 0 (no RPM pulse edge transitions detected). The user's software routine is responsible for determining if the fan is required to run. This decision is made with the user's own software algorithm, and is usually based on temperature measurement information provided to the microcontroller. If the fan is required to run, the user's program sets RUN_FAN to 1; if not, the user's program resets the RUN_FAN bit. At some point in the execution of the user's software, the Fan Control Subroutine (Figure 3) is called to perform the fan control and fault detection functions.

Upon entry to the Fan Control Subroutine, the RUN_FAN bit is immediately checked to see if the fan is required to operate. If the fan is not commanded to run, then the microcontroller's fan control bit (ODRV) is reset, ensuring the fan is off, and the subroutine is exited. However, if RUN_FAN = 1, the fan is required to be running, and ODRV is set to 1 as a result. The fan is next checked for an over current condition (i.e. INLMT = 0). If such a condition is detected, ODRV is reset to 0 (stopping the fan); RUN_FAN is reset to 0, and FAN_FAULT and FAN_OVR_CURRENT are set to 1. The subroutine is then exited with a fault condition present, and the fan is stopped. If no over current condition exists, the Fan Control Subroutine next verifies the fan is indeed running.

Fan rotation is verified by first clearing the EDGE bit, then waiting for a fixed time period (determined by a hardware or software timer) for the RPM output from the TC643 to change state. If the timer expires prior to a state change, the fan is not rotating. This results in resetting ODRV to 0, clearing the RUN_FAN bit, setting the FAN_FAULT bit, and exiting the subroutine. If a transition is indeed detected prior

**Table 1.**

| RPM | FREQUENCY | TIMER PERIOD (msec) |
|---|---|---|
| 1000 | 67 | 15.00 |
| 2000 | 133 | 7.50 |
| 3000 | 200 | 5.00 |
| 4000 | 267 | 3.75 |
| 5000 | 333 | 3.00 |
| 6000 | 400 | 2.50 |
| 7000 | 467 | 2.14 |
| 8000 | 533 | 1.88 |
| 9000 | 600 | 1.67 |
| 10000 | 667 | 1.50 |

to the timer expiring, the timer is halted and cleared, and EDGE is set to 1, indicating the first transition has been recognized. The timer is again started in preparation of detecting the second RPM transition. Again, a stuck or open fan is indicated by an expiration of the timer. However, if the second transition is detected, a valid RPM pulse has been sensed, and the subroutine is exited. Table 1 lists suggested timer settings for four pole fans of various speeds.

## SUMMARY

The TC643 provides a low cost, minimum space solution to the problem of intelligent fan control and fault detection, when supervised by the host microcontroller. It works with virtually all 4- and 8-bit microcontrollers, consuming minimual processor resources while providing maximum flexibility.

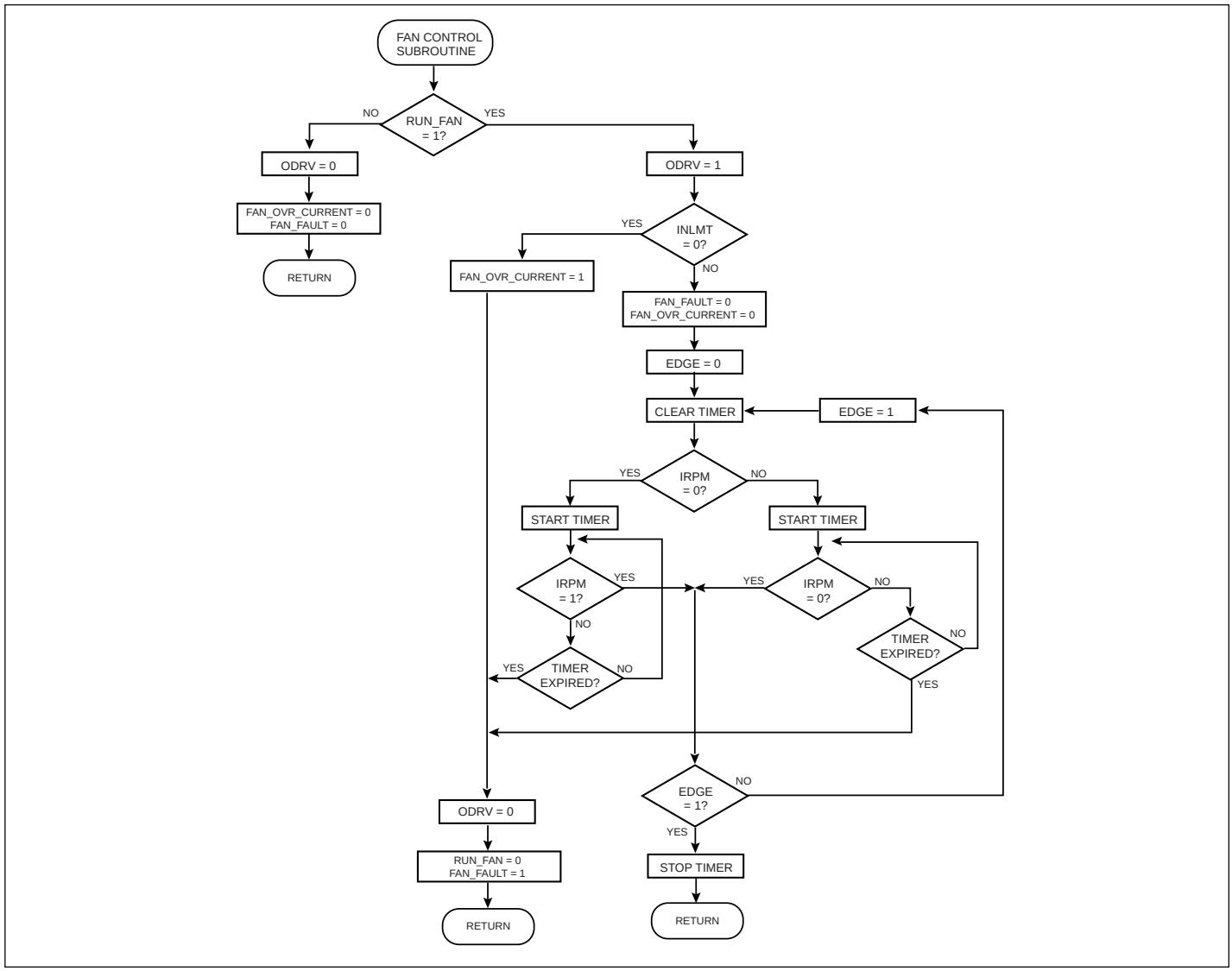


**Figure 3. FanControl Subroutine Flowchart**

**Sales Offices**

**TelCom Semiconductor**
1300 Terra Bella Avenue
P.O. Box 7267
Mountain View, CA 94039-7267
TEL: 650-968-9241
FAX:650-967-1590
E-Mail: liter@c2smtp.telcom-semi.com

**TelCom Semiconductor**
Austin Product Center
9101 Burnet Rd. Suite 214
Austin, TX 78758
TEL: 512-873-7100
FAX: 512-873-8236

**TelCom Semiconductor H.K. Ltd.**
10 Sam Chuk Street, Ground Floor
San Po Kong, Kowloon
Hong Kong
TEL: 852-2324-0122
FAX: 852-2354-9957